Technische Universität München
Fakultät für Elektrotechnik und Informationstechnik
Lehrstuhl für Steuerungs- und Regelungstechnik

Model-Free Robust-Adaptive Controller Design and Identification for Robot Manipulators

Rameez Hayat

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Prof. Dr.-Ing. Andreas Jossen

Prüfer der Dissertation:

1. Prof. Dr.-Ing. (Univ. Tokio) Martin Buss

2. Prof. Dr.-Ing. Harald Aschemann

Die Dissertation wurde am 21.01.2019 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 06.06.2019 angenommen.

Foreword

This thesis summarizes my research carried out over the last few years at the Institute of Automatic Control Engineering (LSR), Technical University of Munich (TUM). This work was supported in part by the TUM, and in part by the Higher Education Commission of Pakistan through the Faculty Development Program “UESTP". I feel very blessed for having an inspiring and professional environment throughout my Ph.D. at LSR. First of all, I would like to thank my supervisor Prof. Martin Buss for his continuous support and the freedom. I am also very grateful to my co-supervisor Dr. Marion Leibold for her support and feedback during my research. I want to thank them for giving me the independence in pursuing different research ideas and providing me the state-of-the-art hardware. I would also like to appreciate the support of Dr. Dirk Wollherr.

Furthermore, I want to thank Prof. Harald Aschemann for his constructive feedback on this work. His insightful comments and recommendations were valuable contributions to my dissertation and are highly appreciated.

I would like to thank my colleagues at LSR for being very supportive and helpful during my stay, especially the SCHRINE group: Stefan Kersting, Sotiris Apostolopoulos, Philine Donner, Stefan Friedrich, Fangzhou Liu, Alexander Pekarovskiy, Markus Schill and Michaela Semmler. The weekly discussions were constructive in improving my research work and confidence. I am also thankful to the secretary Larissa Schmid of LSR for the administrative support. I also want to appreciate Mr. Jaschik, Mr. Weilbach, Mr. Lowitz, Mr. Weber, Mr. Ott and Mr. Stöber for their assistance and help related to hardware.

I am very thankful to my parents, brother, sister and wife for their constant support and valuable advice in all these years. Without my family support, I would not have been able to finish.

Munich, January 2019 Rameez Hayat

Abstract

Robotic manipulators are extensively used in industries, such as automobile, laser-cutting, assembly lines and printed circuit boards. They have the ability to perform a variety of tasks with low cost, high precision, fast production and working in a hazardous environment. A lot of research has been conducted to improve the performance and precision of robotic manipulators. Some major issues in controlling a robot manipulator are the dynamical complexity, the strong coupling between the various joints, input saturation, peaking effect and controller tuning. In addition, modeling of manipulators, especially for higher degree-of-freedom robots is a difficult task and prone to uncertainties. This work presents a novel robust-adaptive control design that does not require system parameters for control. Furthermore, implementation issues present in experiments are addressed for performance enhancement. The other distinctive features of the proposed controller are model-free with easy controller tuning, performance analysis in terms of transient and steady-state, robustness, optimality, the inclusion of input constraints and suppressing peaking effect.

The model-free controller consists of two feedback loops: the inner loop evaluates the robot dynamics to linearize the system and the outer loop is a simple proportional-derivative control. The feedback linearization is performed without using system parameters, instead the estimated model starts from zero and reaches the real system model in finite time. The controller has the same structure as computed-torque control except that the inner loop is model-free. Thus it can be called as an adaptive inverse-dynamics control. Two-players zero-sum using the $\mathcal{H}_\infty$ technique is used to find the stability and robustness of the proposed controller. Quantitative performance analysis is also formulated in terms of steady-state joint errors using the input-to-state-stability approach. The controller guarantees that the state error at steady-state always remains in predefined limits. The parameters of the controller are evaluated using the performance analysis and the input-to-state-stability. Inverse-optimal approach is utilized to find the optimal cost for the proposed controller.

The proposed controller is further improved by addressing the implementation issues, such as peaking and input saturation. Since the proposed controller guarantees a predefined accuracy at steady-state, the unknown time-varying input saturation can be indirectly sensed using the error values. A trajectory-scaling formulation is introduced to modify the desired trajectory such that the performance criterion is satisfied. This dissertation also addresses the issue of peaking effect by using two techniques: the first method uses a variable gain for the controller and the second method adds an extra term in the controller to improve the transient response.

The system is controlled using few tuning parameters along with an intuitive procedure to find the controller parameters. Based on the proposed adaptive controller and the persistent of excitation trajectory, parameters of the system are identified using the least squares technique. The proposed adaptive controller is validated using two, three and seven degree-of-freedom robots. For system parameters identification, two and three degree-of-freedom manipulators are used that result in accurate estimation.

Zusammenfassung

Roboter Manipulatoren sind kaum mehr wegzudenken aus Industrieapplikationen, wie dem Automobilbereich, Laserschneiden, Fertigungslinien und Platinen. Sie ermöglichen es Arbeiten mit niedrigen Kosten, hoher Präzision, schnell und in gefährlichen Umgebungen durchzuführen. Viele Forschungsvorhaben beschäftigten sich damit, die Leistungsfähigkeit und Präzision von Robotern zu verbessern. Einige der großen Schwierigkeiten der Roboterregelung sind die dynamische Komplexität, die starke Verkopplung zwischen Gelenken, Eingangsbeschränkungen, Peaking-Effekt und die Reglereinstellung. Außerdem ist die Modellierung von Manipulatoren eine schwierige Aufgabe und anfällig für Unsicherheiten, vor allem für Roboter mit hohen Freiheitsgraden. Diese Arbeit stellt ein neues Design für robuste, adaptive Regler dar, welches keine Systemparameter zur Regelung benötigt. Weiterhin werden Möglichkeiten zur Leistungssteigerung bei der Implementierung für Experimente aufgezeigt. Weitere Merkmale des vorgestellten Reglers sind die Modellfreiheit, eine einfache Reglereinstellung, eine Leistungsanalyse bzgl. des Einschwingvorgangs und des eingeschwungenen Vorgangs, Robustheit, Optimalität, Berücksichtigung von Eingangsnebenbedingungen und eine Unterdrückung des Peaking-Effekt.

Der modellfreie Regler besteht aus zwei Rückkopplungsschleifen, einer inneren Schleife, die die Roboterdynamik evaluiert, um das System zu linearisieren und einer äußeren Schleife, die einen proportional-ableitenden Regler darstellt. Die Feedback-Linearisierung geschieht ohne Systemparamter zu nutzen, stattdessen startet das geschätzte Modell von Null und erreicht das reale Modell in endlicher Zeit. Der Regler hat die gleiche Struktur wie die Drehmomentenregelung, außer dass die innere Schleife modellfrei ist. Deshalb kann man den Regler als adaptiven Inversdynamikregler bezeichnen. Ein Zwei-Personen-Nullsummenspiel wird zusammen mit der $\mathcal{H}_\infty$ Technik genutzt, um Stabilität und Robustheit zu zeigen. Eine quantitative Leistungsanalyse mit Hilfe eines Ansatzes zur Eingangszustandsstabilität wird ebenso durchgeführt, in Bezug auf den Gelenkfehler im eingeschwungenen Zustand. Der Regler garantiert, dass sich der Zustandsfehler im eingeschwungenen Zustand in einem vorher festgelegten Bereich befindet. Die Reglerparameter werden mit Hilfe der Leistungsanalyse und der Eingangszustandsstabilität ausgewertet. Eine inversoptimale Methode wird genutzt, um die optimalen Kosten für den vorgestellten Regler zu finden.

Der vorgeschlagene Regler wird weiter verbessert, indem Implementierungsprobleme, wie Peaking und Eingangssättigung, behandelt werden. Da der vorgestellte Regler eine vorher definierte Genauigkeit im eingeschwungenen Zustand garantiert, kann die unbekannte zeitvariante Eingangssättigung indirekt durch die Fehlerwerte festgestellt werden. Es wird ein trajektorienskalierender Ansatz vorgestellt, um die gewünschte Trajektorie so zu verändern, dass die Leistungskriterien erfüllt werden. Im Rahmen der Dissertation wird auch der Peaking-Effekt behandelt, indem zwei Techniken angewandt werden, zum einen eine Methode mit variabler Reglerverstärkung und zum anderen eine Methode, die einen zusätzlichen Term in den Regler einfügt, um das Einschwingverhalten zu verbessern.

Das System wird mit wenigen Einstellparametern geregelt. Das Vorgehen, um die Reglerparameter zu finden, ist intuitiv. Basierend auf dem vorgeschlagenen adaptiven Regler und der [permanenten Anregung] werden die Systemparameter mit der Methode der kleinsten Quadrate identifiziert. Der vorgeschlagene Regler wird mit Robotern mit zwei, drei und sieben Freiheitsgraden validiert. Roboter mit zwei und drei Freiheitsgraden werden zur Parameteridentifikation genutzt, was zu genauen Schätzungen führt.

List of Publications

Publications resulting from the work performed during the doctoral research at Technical University of Munich with first authorship and co-authorship:

- R. Hayat, M. Leibold, and M. Buss, “Robust-adaptive controller design for robot manipulators using the $\mathcal{H}_\infty$ approach,” *IEEE Access*, vol. 6, pp. 51 626 – 51 639, 2018

- ——, “Addressing control implementation issues in robotic systems using adaptive control,” *Robotica*, pp. 1–14, 2019

- R. Hayat and M. Buss, “Model identification for robot manipulators using regressor-free adaptive control,” *IEEE 11th International Conference on Control (UKACC)*, pp. 1–7, 2016

- Y. Sun, Z. Zhang, M. Leibold, R. Hayat, D. Wollherr, and M. Buss, “Protective control for robot manipulator by sliding mode based disturbance reconstruction approach,” *IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 1015–1022, 2017

Contents

List of Figures **ix**

List of Tables **xi**

Notation **xiii**

1 Introduction **1**

1.1 Challenges in Control of Manipulators 2

1.2 State of the Art 3

1.2.1 Model-Based Control 4

1.2.2 Model-Free Control 6

1.3 Main Contribution and Thesis Outline 8

2 Adaptive Control **13**

2.1 Euler-Lagrange (E-L) Representation 14

2.2 Computed-Torque 14

2.3 Regressor-Based Adaptive Control 15

2.3.1 Slotine and Li's Adaptive Control 17

2.4 Regressor-Free Adaptive Control 18

2.5 Proposed Model-Free Robust-Adaptive Control 19

2.6 Results 21

2.6.1 Simulation Results 23

2.6.2 Experimental Results 24

2.7 System Identification 27

2.8 Summary 29

3 Stability, Optimality and Performance Analysis **31**

3.1 Proposed Controller 32

3.2 Robustness Analysis 34

3.3 Stability 37

3.3.1 Derivation of Proposed Controller 37

3.3.2 Evaluation of the Matrix $\boldsymbol{R}$ 39

3.4 Performance Analysis and Stability 40

3.4.1 Evaluation of Controller parameters 41

3.4.2 Discussion on $\hat{\boldsymbol{\phi}} \to \boldsymbol{\phi}$ 42

3.5 Results 43

3.5.1 Simulation Results . . . 44
3.5.2 Experimental Results . . . 48
3.5.3 Effects of Order and Gain of the Controller . . . 48
3.5.4 Experimental Results for 7-DoF Robot . . . 51
3.6 Summary . . . 51

4 Performance Enhancement of Adaptive Controller 53
4.1 Contribution . . . 55
4.1.1 Robustness and Stability . . . 55
4.1.2 Modified Desired Trajectory . . . 58
4.1.3 Improve Transient Response . . . 60
4.1.4 Easy Tuning for the Controller . . . 62
4.2 Results . . . 63
4.2.1 Simulation Results . . . 63
4.2.2 Experimental Results . . . 66
4.3 Summary . . . 67

5 Conclusion and Future Directions 69
5.1 Discussion . . . 69
5.2 Outlook . . . 72

A Mathematical Background 73

B Proofs and Codes 77

C Details of the 3-DoF Manipulator 81

Bibliography 87

List of Figures

1.1 Feedback Linearization. 10

2.1 Computed-torque in the absence of disturbances. 15
2.2 Two-DoF serial robot used in experiments. 22
2.3 Joint angles for the desired circular trajectory. 22
2.4 Results of regressor-based adaptive control: this technique requires inverse of inertia matrix ($\hat{\boldsymbol{M}}$). Tracking of the desired trajectory is sensitive to any change in controller parameters. 23
2.5 Results of Slotine and Li's adaptive control. As clear from Fig. (a), the estimated model dynamics do not follow the original model. Reason is that Lyapunov function only shows the tracking error to be a decreasing function. 24
2.6 Results of FAT-based adaptive control. Approximation of system parameters by a combination of some orthogonal bases has high computational cost. . . 25
2.7 Results of proposed adaptive control. Taking a very low time constant for the update law will confirm the modeling error to approach zero. This technique requires only one parameter for the adaptation of system dynamics. 25
2.8 Error ($\tilde{\boldsymbol{\phi}}$) comparison of all the four controllers for the first joint angle of 2-DoF robot. As shown, the error for the FAT, regressor and Slotine/Li's controllers are not zero. 26
2.9 Experimental results: position of end effectors starting from $0.5\widehat{x} + 0\widehat{y}$ m. . . 26

3.1 Robust-adaptive control. 32
3.2 Adaptive control from the perspective of $\mathcal{H}_\infty$ optimal control. 35
3.3 3-DoF (left) and 7-DoF (right) robot manipulators used in experiments. . . 43
3.4 Top view of the 3-DoF robot manipulator along with the desired trajectory. The lengths are: $l_1 = 0.30\ m$, $l_2 = 0.24\ m$ and $l_3 = 0.34\ m$. 45
3.5 Simulation results: trajectory tracking in Cartesian space. The end-effector starts from (0.88 m,0) in the x-y coordinate system. 45
3.6 Simulation results: (a), (b) and (c) represent the three joint angles with respect to time. (d), (e) and (f) represent the three joint torques with respect to time. Dashed line: $m = 1$, solid line: $m = 2$. 46
3.7 Simulation results for joint errors: (a) $m = 1$ (b) $m = 2$. Red line: q_1, green line: q_2, blue line: q_3. These figures show that the controller parameters satisfy the performance criteria of (3.32) and (3.33). 46
3.8 Difference between $\hat{\boldsymbol{\phi}}$ and $\boldsymbol{\phi}$. Dashed line: $m = 1$, solid line: $m = 2$. 47

3.9 Experimental results: (a), (b) and (c) represent the three joint angles for $m = 1$ with respect to time. (d), (e) and (f) represent the three joint angles for $m = 2$. Dashed line: q, solid line: q_{d}. 48

3.10 Experimental results: trajectory tracking in Cartesian space. Dashed line: $m = 1$, solid line: $m = 2$. The end-effector starts from (0.88 m,0) in the x-y coordinate system. 49

3.11 Experimental results for joint errors: (a) $m = 1$ (b) $m = 2$. Red line: q_1, green line: q_2, blue line: q_3. These figures show that the controller parameters satisfy the performance criteria of (3.32) and (3.33). 49

3.12 First order approximation with different a_0. 50

3.13 Various orders of control law with same convergence rate. 50

3.14 Errors in the joint angles: – PD control, – robust control [5] and – proposed control. 50

4.1 Adaptive control with speed adjustment. 55

4.2 User-defined specification for $f(\boldsymbol{e})$. 59

4.3 Specifications for the gain scheduling used in the results. 62

4.4 Simulation: Tracking without input constraints. (a) Position of the end-effector in Cartesian coordinates, (b) joint errors. 63

4.5 Joint angles for the desired circular trajectory. 65

4.6 Simulation: Trajectory in the presence of input constraints. (a) Without scaling, (b) with scaling. 65

4.7 Simulation: Comparison of time t and θ. (a) t and θ, (b) q and q_d. 65

4.8 Simulation: (a) Without gain adjustment. (b) Gain adjustment with $a = 20 - 80$ using (4.27). 66

4.9 Experiment: (a) Trajectory of the end-effector, (b) shows that the performance limitation is satisfied by the blue trajectory in (a). (c) represents the error in the joint angles for the green trajectory in (a). 66

4.10 Experiments: (a) The desired trajectory took 1.2 seconds for one complete rotation for both trajectories. (b) shows that the performance limitation is satisfied by the blue trajectory in (a). (c) represents the difference between θ (red) and t (blue). 67

C.1 Details of the 3-DoF manipulator used in the dissertation. 81

C.2 Dimensions of L_1. 83

C.3 Dimensions of $\mathrm{L}_2, \mathrm{L}_3, \mathrm{L}_4$ and L_5. 83

C.4 Dimensions of l_1. 84

C.5 Dimensions of l_2. 84

C.6 Dimensions of l_3. 84

C.7 Dimensions of $\mathrm{M}_1, \mathrm{M}_2$ and M_3. 85

C.8 Dimensions of $\mathrm{G}_1, \mathrm{G}_2$ and G_3. 85

C.9 Dimensions of N_1 and N_2. 85

List of Tables

1.1 Comparison: (-) means not completely true/known. Q-P stands for Quantitative Performance. 9

2.1 Calculated, simulation and experimental estimation of parameters for 2-DoF manipulator. 28
2.2 Parameter estimation for 3-DoF. 28

3.1 Controller parameters: (a) ignoring $-\boldsymbol{x}^T\boldsymbol{S}\boldsymbol{B}\boldsymbol{B}^T\boldsymbol{S}\boldsymbol{x}$, (b) including $-\boldsymbol{x}^T\boldsymbol{S}\boldsymbol{B}\boldsymbol{B}^T\boldsymbol{S}\boldsymbol{x}$, (c) values used in controller. 44
3.2 Error in the joint angles (rad). 50

4.1 3-DoF planar robot dynamical equations. 64

C.1 Details of L_1. 83
C.2 Details of $\mathrm{L}_2, \mathrm{L}_3, \mathrm{L}_4$ and L_5. 83
C.3 Details of l_1. 84
C.4 Details of l_2. 84
C.5 Details of l_3. 84
C.6 Details of $\mathrm{M}_1, \mathrm{M}_2$ and M_3. 85
C.7 Details of $\mathrm{G}_1, \mathrm{G}_2$ and G_3. 85
C.8 Details of N_1 and N_2. 85

Notation

Acronyms and Abbreviations

DoF	degree-of-freedom
E-L	Euler-Lagrange
FAT	function approximation technique
GARE	game algebraic Riccati equation
HJI	Hamilton-Jacobi-Isaacs
ISS	input-to-state stable
LPV	linear parameter-varying modeling
LS	least squares
MPC	model-predictive control
PD	proportional-derivative
PI	performance index
PID	proportional-integral-derivative
Q-P	quantitative performance
SMC	sliding-mode control

Mathematical Conventions

Scalar Constants

n_{r}	gears ratio
k_{t}	torque constant

i_{a}	armature current
$m_{(\cdot)}$	scalar parameters of inertia matrix
$n_{(\cdot)}$	scalar parameters of Coriolis, centrifugal and gravity
ω	positive constant
$a_{\mathrm{l}}^i, b_{\mathrm{l}}^i$	weights of l^{th} harmonic for the reference trajectory of the i^{th} joint angle
a_{i}, b_0	positive constants used in proposed controller
t_0	initial time
h	positive constant
$\sigma_{(\cdot)}$	positive constants
$a_{\max}$	maximum acceptable value of control parameters
$a_{\min}$	minimum acceptable value of control parameters
K_1	positive constant
K_2	positive constant
$\alpha_{(\cdot)}$	positive constants
$\gamma_{(\cdot)}$	positive constants
N	total number of sampled data

Scalar variables

$a(\boldsymbol{x})$	function of states
$\lambda_{k_{\max}}$	maximum eigenvalue of $\boldsymbol{K}^T\boldsymbol{K}$
$\lambda_{\min}$	minimum eigenvalue of $\boldsymbol{Q} + (2a - \tilde{K})\boldsymbol{S}\boldsymbol{B}\boldsymbol{B}^T\boldsymbol{S}$
θ	function of state error and time
$f(\boldsymbol{e})$	trajectory down-scaling function
$g(\theta, t)$	trajectory up-scaling function
$V(\cdot)$	Lyapunov function
m	order of the proposed adaptive controller

Vectors, Matrices and Functions

$\boldsymbol{M}(\cdot) \in \mathbb{R}^{n\times n}$ inertial matrix

$\boldsymbol{C}(\cdot) \in \mathbb{R}^{n\times n}$ Coriolis and centrifugal matrix

$\boldsymbol{G}(\cdot) \in \mathbb{R}^{n\times 1}$ gravity vector

$\boldsymbol{F}(\cdot) \in \mathbb{R}^{n\times n}$ viscous friction matrix

$\boldsymbol{\tau} \in \mathbb{R}^{n\times 1}$ total torque

$\boldsymbol{\tau}_{\mathrm{d}} \in \mathbb{R}^{n\times 1}$ disturbance

$\boldsymbol{\tau}_{\mathrm{in}} \in \mathbb{R}^{n\times 1}$ input torque vector

$\boldsymbol{N}(\cdot) \in \mathbb{R}^{n\times 1}$ Coriolis, centrifugal and gravity vector

$\boldsymbol{v}_{\mathrm{PD}} \in \mathbb{R}^{n\times 1}$ computed-torque augmented variable vector

$\boldsymbol{v}_{\mathrm{SL}} \in \mathbb{R}^{n\times 1}$ Slotine and Li's controller variable vector

$\boldsymbol{f}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{n\times 1}$ general system representation

$\boldsymbol{q} \in \mathbb{R}^{n\times 1}$ joint angle vector

$\boldsymbol{q}_{\mathrm{d}} \in \mathbb{R}^{n\times 1}$ desired angle vector

$\boldsymbol{K}_{\mathrm{p}} \in \mathbb{R}^{n\times n}$ proportional gain

$\boldsymbol{K}_{\mathrm{d}} \in \mathbb{R}^{n\times n}$ derivative gain

$\boldsymbol{K}_{\mathrm{D}} \in \mathbb{R}^{n\times n}$ Slotine and Li controller gain

$\boldsymbol{e} \in \mathbb{R}^{n\times 1}$ joint error vector

$\boldsymbol{Y}(\cdot) \in \mathbb{R}^{n\times p}$ regressor matrix, where p depends on number of parameters

$\boldsymbol{p} \in \mathbb{R}^{p\times 1}$ system parameters vector

$\boldsymbol{A'} \in \mathbb{R}^{2n\times 2n}$ transition matrix for regressor-based controller

$\boldsymbol{B'} \in \mathbb{R}^{2n\times n}$ input matrix for regressor-based controller

$\boldsymbol{x'} \in \mathbb{R}^{2n\times 1}$ states for regressor-based controller

$\boldsymbol{x}_C \in \mathbb{R}^{3\times 1}$ Cartesian coordinate vector

$\boldsymbol{\phi} \in \mathbb{R}^{n\times 1}$ vector representing system

$\hat{\boldsymbol{\phi}} \in \mathbb{R}^{n\times 1}$ estimation of system vector

$\tilde{\boldsymbol{\phi}} \in \mathbb{R}^{n\times 1}$ error in estimation of system vector

$\boldsymbol{A} \in \mathbb{R}^{(m+2)n\times(m+2)n}$ transition matrix

$\boldsymbol{L} \in \mathbb{R}^{n\times(m+1)n}$	part of transition matrix
$\boldsymbol{B} \in \mathbb{R}^{(m+2)n\times n}$	input matrix
$\boldsymbol{x} \in \mathbb{R}^{(m+2)n\times 1}$	state vector
$\boldsymbol{u} \in \mathbb{R}^{n\times 1}$	system input
$\boldsymbol{d} \in \mathbb{R}^{n\times 1}$	disturbance input
$\bar{\boldsymbol{x}} \in \mathbb{R}^{2n\times 1}$	state vector
$\boldsymbol{z} \in \mathbb{R}^{mn\times 1}$	state vector
$\bar{\boldsymbol{A}} \in \mathbb{R}^{2n\times 2n}$	transition matrix
$\boldsymbol{H} \in \mathbb{R}^{mn\times mn}$	transition matrix
$\bar{\boldsymbol{B}} \in \mathbb{R}^{2n\times mn}$	input matrix
$\boldsymbol{I}_n \in \mathbb{R}^{n\times n}$	identity matrix
$\boldsymbol{s} \in \mathbb{R}^{n\times 1}$	vector
$\boldsymbol{J}(\boldsymbol{q}) \in \mathbb{R}^{2\times 2}$	Jacobian Matrix
$\boldsymbol{\zeta} \in \mathbb{R}^{n}$	constant vector
$\triangle \in \mathbb{R}^{n\times n}$	positive definite matrix
$\boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^{n}$	nonlinear affine system
$\boldsymbol{g}(\boldsymbol{x}) \in \mathbb{R}^{n\times r}$	input function
$\boldsymbol{k}(\boldsymbol{x}) \in \mathbb{R}^{n\times p}$	disturbance function
$\boldsymbol{R} \in \mathbb{R}^{n\times n}$	positive definite matrix of proper dimensions
$\tilde{\boldsymbol{K}} \in \mathbb{R}^{n\times n}$	positive definite matrix
$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{d}) \in \mathbb{R}^{n}$	nonlinear system
$\overline{\boldsymbol{\tau}} \in \mathbb{R}^{n\times 1}$	torque saturation
$\boldsymbol{x}_{\max_{(\cdot)}}$	maximum allowed error vector at steady-state
$\boldsymbol{P}_{(\cdot)}$	positive definite matrices
$\boldsymbol{Q}$	positive definite matrix
$\boldsymbol{\mu}$	kinematic parameter vector
$\boldsymbol{Z}_M$	basis function approximation of inertia matrix
$\boldsymbol{Z}_F$	basis function approximation of friction matrix

$\boldsymbol{Z}_N$	basis function approximation of $\boldsymbol{N}$ matrix
$\boldsymbol{W}_M^T$	weights for basis function approximation of inertia matrix
$\boldsymbol{W}_F^T$	weights for basis basis function approximation of friction matrix
$\boldsymbol{W}_N^T$	weights for basis basis function approximation of $\boldsymbol{N}$ matrix
$\boldsymbol{Q}_M$	cost function for inertial matrix
$\boldsymbol{Q}_F$	cost function for friction matrix
$\boldsymbol{Q}_N$	cost function for Coriolis, centrifugal and gravity vector
∇V	partial derivative of V with respect to $\boldsymbol{x}$
$\boldsymbol{\mathcal{Y}}$	least squares regressor matrix with N sampled data
$\overline{\boldsymbol{B}}$	least squares input matrix with N sampled data

Derivatives and Integrals

$\frac{\mathrm{d}x}{\mathrm{dt}}$	total derivative
$\frac{\mathrm{d}^m x}{\mathrm{dt}^m}$	mth order total derivative
$\int x\mathrm{dt}$	integral of x with respect to t
$\int_m \boldsymbol{e}\mathrm{dt}^m$	mth integral of $\boldsymbol{e}$

More Conventions

$\mathcal{K}_\infty$	a class $\mathcal{K}_\infty$ is a function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, which is continuous, unbounded, increasing and satisfies $\alpha(0) = 0$
$\mathcal{KL}$	a class $\mathcal{KL}$ is a function $\beta : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, such that $\beta(.,t) \in \mathcal{K}_\infty$ for all t and $\beta(r,t) \searrow 0$ as $t \to \infty$, where r is a constant
$\|\| \cdot \|\|_\infty$	infinity norm
$\| \cdot \|$	vector norm 1
$\triangleq$	by definition “equal to”
$\min_{\boldsymbol{u}}$	minimize $\boldsymbol{u}$
$\max_{\boldsymbol{d}}$	maximize $\boldsymbol{d}$
$(\hat{\cdot})$	estimation of $(\cdot)$

$(\tilde{\cdot})$	difference between real and estimated value of $(\cdot)$
$\|\| \cdot \|\|$	Euclidean norm
$\|\boldsymbol{x}_{\mathrm{perf}}(2)\|$	maximum allowed error at steady-state
$\|\boldsymbol{e}_{\mathrm{perf}}\|$	maximum allowed error at steady-state

1

Introduction

In industries, the demand for increased productivity, uniform quality end-product and low manufacturing cost have paved the way for high-performance robots. Recent advancements in the field of computer-based automation and reliable feedback sensing have made it possible to automate industries [6]. Earlier robots, also known as 'Hard Automation', were designed for mass production of specific products. However, even a slight change in the product design required new hardware manufacturing setup, thus increasing the production cost. With the help of computer-controlled robots, it is now possible to re-program a robot to perform a variety of tasks [7]. This also led to the use of robots in many non-industrial fields, such as hospitals and homes.

A robot manipulator is a combination of revolving and sliding joints, controlled electronically to manipulate objects. These robot manipulators have gained overwhelming attention in the past few decades. Controlling a robotic manipulator, especially at high speed, is a challenging task because of the strong coupling between the joints, complex nonlinear dynamics, unknown input saturation and uncertainties [8, 9]. The presence of uncertainties, unmodeled dynamics and disturbances make the control operation difficult for high-performance requirements, such as assembly, machining and laser cutting. In literature, a vast variety of control approaches are available to control a robot, such as robust control [5, 10–12], adaptive control [3, 13–22], neural network control [23–26], observer-based control [27, 28], orbital stabilization based control [29, 30], model predictive control (MPC) [31–35], GPI control [36], active disturbance rejection control [37] and sliding mode control [4, 38, 39]. Some of these control schemes are model-free but the lack of system information limits their performance.

In literature, model-based control schemes are extensively used to control robot manipulators. Thus, prior information about the system model is required to implement such controllers [40, 41]. The number of system parameters expands with the increase of degree-of-freedom (DoF) and consequently, the parameter identification becomes a difficult process. These parameters are also highly sensitive to factors, such as, load and temperature variations, which make the control process extremely challenging. Due to the significant difference between the experimental and simulation results, a simplified model approach for model-based control results in a deteriorating performance. Another issue with model-based methods is the need for a trade-off between the system's performance and robustness because of uncertainties and unmodeled dynamics.

Unlike robust control with a fixed control law, adaptive control has the ability to adapt to changes in the system. The main focus of this dissertation is the design and implementation of a novel model-free adaptive controller. The basic idea is related to reinforcement learning, where the controller gives an optimal result without knowing the system dynamics parameters. In fact, the system dynamics are estimated online by using a simple differential equation that only requires the state information. A number of mathematical tools, such as input-to-state-stability and inverse optimal control have been used to prove the stability, optimality and robustness of the proposed control method.

In Section 1.1, an overview of the challenges associated with controlling a robot manipulator is presented. We explored the literature for the solution of identified challenges and compared the related studies in Section 1.2. The state-of-the-art control methods are explained in two categories for better understanding: model-base and model-free. Finally, a short overview of the main contributions of our work is given in Section 1.3.

1.1 Challenges in Control of Manipulators

A manipulator dynamics consist of inertia, Coriolis, centrifugal, gravity and friction forces. Actuation in robots is usually performed by current controlled motors. In the following, the main challenges in the control of robot manipulators are summarized:

Coupled Dynamical Model

The robot dynamics are often not accurately known, especially for 6- or 7-DoF robots with a large number of system parameters. These parameters are dependent on the masses, lengths and frictions of different joints. Inertia, gravity, Coriolis and centrifugal forces are the reasons for the coupling effect. At low speeds, these coupling can be ignored and a decoupled controller can be utilized to achieve high performance. A model-based control method usually relies on the dynamics of the controller and hence, the performance of many state-of-the-art control schemes are dependent on the estimation of the system dynamics.

Complex Friction Model

In general, a linear friction model is used in literature for control design [42, 43]. The real friction model is complex and highly nonlinear, e.g., static friction, Stribeck friction and Coulomb's friction [7, 44, 45]. The effect of these friction models is often dominating and thus can not be ignored. Furthermore, the friction parameters are sensitive to disturbances, such as temperature and orientation of the robot manipulator. Makkar *et al.* proposed a continuously differentiable friction model that estimates the nonlinear friction model [46]. The state-of-the-art control approaches usually use only the viscous friction model because of its linear behavior.

Torque Saturation

The motors and the actuator dynamics have a significant impact on the performance of the system. Input saturation in robotics is state-dependent because of back-electromotive force and it is challenging to predict the precise time-varying input saturation. This saturation affects the performance of the system in terms of joint errors. Another problem is the first-order behavior of the motor drives that force the current/torque to reach the desired value in finite time and not instantaneously. Fortunately, this phenomenon is useful in adaptive control to suppress the high-frequency noise caused by the encoders as will be explained later.

Uncertainties and Unmodel Dynamics

As discussed, most of the friction models are not considered during controller design due to their complexity. Load and temperature variations also affect the parameters of the system. There are also uncertainties in the input provided by the motor drives. Furthermore, because of the high turn ratios, small inertial values of the gears and rotors are amplified by the square of the turn ratios and hence play a vital role in the system dynamics.

Controller Parameters

In general, a controller requires the states, desired states and system information along with some constant controller parameters that significantly affect the performance. It is often not feasible to tune the controller parameters during operation because of safety. Usually, the control tuning is a laborious and troublesome task in controlling a robot manipulator. Additionally, the appropriate controller parameters acquired from simulation results are often not applicable in experiments due to uncertainties and unmodeled dynamics.

Physical Safety

Uncertain transient phase in robot manipulators is an implementation issue, especially when using an adaptive controller. The unpredictable yet stable behavior during this phase is caused because of the challenges mentioned above. A sudden overshoot, also called peaking effect, can sometimes lead to physical danger for users or it can damage the robot.

Performance

Apart from the challenges mentioned above, there are other controlling concerns, such as optimality and guaranteed performance. Most of the controllers only guarantee the stability of the closed-loop system and lack optimality.

Unlike linear systems with known dynamics, steady and transient-state performance analysis for robotic systems are difficult because of the uncertainties and unmodeled dynamics [47, 48]. Such performance analysis are also not covered in literature because of the challenges previously mentioned.

To address these challenges and concerns, some state-of-the-art control methods are discussed in the next section.

1.2 State of the Art

For robot manipulators, high-speed performance and accuracy are the basic requirements in industrial applications. Model-based control approaches usually deliver high performance in terms of user-defined steady-state and transient-state response. Unfortunately, an accurate model of the system is required to achieve high performance. On the other hand, model-free controllers are usually robust, at the same time, compromising the system performance. In this chapter, a number of state-of-the-art control approaches are analyzed on the basis of the following performance criteria:

- Model-based/model-free
- Optimality
- Tuning of controller parameters
- Quantitative performance analysis

The control design techniques for robots are divided into two main classes: (1) model-based and (2) model-free. The state-of-the-art controllers for robots are (but not limited to):

1.2.1 Model-Based Control

In literature, the majority of the control techniques for manipulators are model-based. These techniques show acceptable performance for low DoF robots. For a 6- or 7-DoF manipulator, the system uncertainties deteriorate the performance of these controllers. Following are the model-based control approaches available in the literature, categorized according to their design methods for easy understanding.

Lyapunov-Based Control

The Lyapunov-based control methods use the Lyapunov stability equation to design controllers. The focus of this approach is to find a controller that keeps the derivative of Lyapunov equation negative, thus guaranteeing the stability. However, the approach does not consider any performance criterion.

The dynamics of a manipulator is generally represented by Euler-Lagrange (E-L) equations, as explained in Chapter 2. Since the structure of the system is known and only the constant parameters of the system are prone to uncertainties, such as masses and lengths of joints, the E-L equations can be alternatively represented by a known regressor matrix and an unknown parameter vector. A large number of adaptive control methods use this linear-in-parameters representation along with Lyapunov stability technique to design the controller [20, 49–57]. The Lyapunov equation is utilized to estimate the unknown vector such that the system is stable [49]. The basic idea is to cancel cross terms in the Lyapunov stability equation. The drawback is the slow convergence rate that makes the transient response long and unpredictable. Besides, the regressor matrix has too many parameters, which complicates the matrix evaluation for a high DoF robot.

The Lyapunov-based controllers, such as Slotine and Li's adaptive control perform well for a low DoF robot [49]. Finding suitable controller gains is also a drawback of the Lyapunov-based controller. As a result, implementation on higher DoF robots requires much trial and error to achieve the desired performance.

Robust Control

Robust control design deals with disturbances and uncertainties provided that they are within some bounds. Sharma *et al.* proposed a predictive-based control method for E-L systems, where the controller can deal with long input delays [58]. Fortunately, recent robotic systems usually do not have severe delay issues. A robust-adaptive controller is proposed in [59] that is much similar to Slotine and Li's adaptive control [49]. The advantage of the controller is fast parameter convergence for better transient response; thus, high gains are not required for convergence. Nonetheless, the controller requires many tuning parameters and like Lyapunov-based controller, there is a problem of scalability to higher order systems.

An inverse-optimal proportional-integral-derivative (PID) controller based on $\mathcal{H}_\infty$ optimality is proposed by Chung *et al.* [5,10], which uses the Hamilton-Jacobi-Isaacs (HJI) equations to design the controller. The controller is easy to implement with few tuning parameters, however, an estimate of the system dynamics is required for the control implementation. Similarly, a finite-time $\mathcal{H}_\infty$ control is implemented in [60] that ensures a quantitative performance in terms of stability. Tuning and model information are the main concerns in the implementation of the controller. Furthermore, a linear parameter-varying modeling (LPV) and identification method is proposed by Hashemi *et al.* [61]. Principle component analysis is used to obtain the LPV models with tighter parameter sets and $\mathcal{H}_\infty$ technique is used for the controller tuning. The primary challenge in the LPV approach is scheduling parameters for higher order systems. Another drawback is a large number of system parameters, especially for a 6- or 7-DoF manipulator.

Orbital Stable Control

An optimal control for periodic trajectories is proposed by Shiriaev *et al.* [30], which transforms the robotic system dynamics into a lower dimension system. The controller also considers the input constraints and can be implemented on under-actuated systems. Unfortunately, the proposed method does not show the performance in the presence of disturbances. Furthermore, the time-varying input saturation is assumed to be known, which in reality is difficult to be evaluated because of the challenges explained in Section 1.1.

A similar control technique is introduced by Pchelkin *et al.* [29], which is based on transverse dynamics that ensures the states to remain in a vicinity around the desired trajectory. The controller relies on the estimated model of the system and also assumes that the input saturation is available.

Model Predictive Control

Model predictive control is a numerical optimization technique that evaluates the optimal input for the next time-step by considering a finite horizon. The advantage of MPC is that the controller gives an optimal solution with input constraints embedded in the controller [62]. Hedjar *et al.* proposed a nonlinear MPC method using Taylor approximation [33], where an integral action was introduced into the loop to enhance the robustness of the controller in the presence of disturbances and uncertainties.

It has been shown in [3,31] that the computation cost of MPC is much higher for systems that usually requires 1 kHz sampling-time, such as manipulators. MPCs are more helpful for path planning, mobile robots and chemical industries, etc.

Neural Network Control

Neural network is a universal approximation technique for nonlinear systems. It has been used for identification and control in robotics. In [23,24], a nominal system model, similar to inverse-dynamics control, is used to control a robot. The unknown mismatch between nominal and real model is approximated by using radial basis functions neural network. $\mathcal{H}_\infty$ is then used to validate the robustness of the control method. The advantage is an improvement upon the conventional inverse-dynamics control, where no compensation is provided for the mismatch.

1.2.2 Model-Free Control

The model-based adaptive control design depends on apriori knowledge of the system dynamics. As a result, if the modeling errors become large, the system will exhibit poor transient and steady-state performance. To improve the tracking response, the design requires high controller gains in order to minimize the modeling uncertainties. This requirement of high controller gains makes the model-based control design even more complicated in real-time application as they may cause high-frequency chattering. In contrast, model-free controllers are, in general, easy to implement and do not rely on parameter variations. A comparison between basic model-free and model-based controllers is given in [63]. Following are some model-free control methods:

Adaptive Control

The Lyapunov based controllers require a difficult task of finding the regressor matrix. This matrix can have hundreds of parameters depending on the DoF of robot manipulators. To avoid the use of system model, function approximation technique (FAT) approximates the system dynamics with linear orthogonal basis functions [13, 14, 64]. For an accurate approximation of the system, a large number of basis functions are required, which increase the computation time of the FAT-based control. Also, the nonlinear dynamics of a manipulator is approximated by linear functions.

Furthermore, for the estimation of the system dynamics, two robust-adaptive laws are proposed by Safaei *et al.* [65]. The Lyapunov method is used in this technique to validate the stability of the system. A significant issue in the controller is the scalability of the controller to higher order manipulators. Both adaptive controllers exhibit the same drawbacks of Lyapunov-based controller design.

Sliding-Mode Control (SMC)

SMC consists of two steps: first, a sliding surface is designed with some predefined specification, and then a control law is selected that will force the system states towards the surface. SMC keeps the states in the close neighborhood of the sliding surface. An advantage of SMC is the model-free and robust behavior [38, 39, 66–69].

Sun *et al.* [4] proposed a sliding mode observer to estimate the external and internal disturbances, and a SMC that deals with small disturbances. In addition, a protective control strategy is used when there is strong disturbances, e.g., external faults or collisions. The focus is more on the disturbances rather than the performance of the controller. A switching sliding-mode control method is proposed in [39] with multiple Lyapunov functions for stability.

The drawback is that SMC usually have chattering effects that is often not acceptable in robots. Similarly, another issue is the implementation on high DoF manipulators. To the best of our knowledge, most of the SMC schemes use two or three DoF manipulators for validation.

Fuzzy and Neural Network Control

A discrete fuzzy estimator approach is proposed by Fateh *et al.* [70, 71]. An adaptive fuzzy cerebellar model articulation controller is proposed by Guan *et al.* [72]. A contouring control scheme is introduced in [73], where the controller considers task space for better contour tracking. Radial basis functions neural networks is used to approximate the system dynamics. The above controllers have no steady-state or transient phase performance analysis. A model-free discrete fuzzy control is introduced in [70] that uses fuzzy estimator to estimate the system parameters. The main concept is to decouple the system equations into multiple single-input-single-output systems. The decoupling will deteriorate the performance further as we increase the order of the system. Similarly, a multi-dimensional Taylor network inverse control method was proposed by Zhang and Yan [74] that has a low computational cost.

Robust Control

Bechlioulis *et al.* [75] proposed a robust controller that ensures a prescribed steady-state and transient-state performance. The controller is only applied on a 2-DoF robot by using linear filters. An issue with the controller is the decoupled joints approach, which affects the performance of the feedback system. Furthermore, a PID controller with a delay compensator is proposed by Alibeji *et al.* [76]. The compensator estimates the delay present in the input of an E-L system. A drawback of the controller is the number of controller tuning parameters, which limits the scalability to higher order systems.

Fractional-Order Control

Compared to classical PID control, which has three tuning parameters, the fraction-order control method consists of five turning parameters and hence providing more flexibility to design a controller [77].

A fractional-order PID controller for manipulators is designed in [78]. Due to the PID nature, the controller is easy to implement, however, there is no consideration of disturbance or uncertainties in the system. Another fractional-order control based on back-stepping control method is introduced by Nikdel *et al.* [79, 80]. Implementation on higher order systems and tuning are the major issues with the controller.

Intelligent PID control

Based on the operating region of a system and model variations, the classical PID controllers require re-tuning for better performance. In model-free intelligent controllers, a single PID is sufficient because of online estimation of the system dynamics. A generalized intelligent PID (iPID) controller with straightforward tuning is proposed by Fliess *et al.* [81–83]. The basic idea is the same as inverse-dynamics control, where the system dynamics are canceled and eventually, a linear error model is obtained. The system dynamics are approximated by piecewise constant functions using operational calculus.

So far, the iPID controller lacks any quantitative performance analysis. It will be shown in Chapter 3 that the proposed controller uses the same idea of the intelligent PID controller.

A major difference lies in the evaluation of the system dynamics for linearization. Moreover, the proposed controller is validated by optimality and performance analysis.

The control schemes discussed above are summarized in Table 1.1 with the following features for comparison: (a) quantitative performance (Q-P), (b) easy tuning, (c) optimality and (d) model-free.

1.3 Main Contribution and Thesis Outline

Adaptive control methods are extensively used in control systems because of their less dependence on the system dynamics. In addition, internal and external disturbances can be considered as part of unknown dynamics as long as they are bounded. However, the classical adaptive controllers are not usually optimal and do not provide performance analysis. Recently, optimal adaptive control using reinforcement learning is utilized to control robot manipulators. An advantage of this method is that the controller does not require system dynamics [86–89]. This dissertation proposed an adaptive controller that has the following features:

- **Model-free:** The only assumption is that the inertia, Coriolis and centrifugal, gravity and friction matrices are bounded.
- **Optimal:** Inverse optimal control method is used to show the optimality of the controller.
- **Robust:** Since the controller is model-free, it is robust to any bounded disturbances or parameter variations.
- **Quantitative performance analysis:** The steady-state behavior of the system satisfies predefined performance criterion. The criterion is to keep the steady-state errors below a predefined value, which is 0.01 rad in this dissertation.
- **Easy tuning:** Tuning is intuitive and straightforward because of few controller parameters. A procedure is also formulated to find the control parameters.
- **Improved transient response:** The transient response is qualitatively improved using some augmentations in the proposed controller. Peaking is also reduced by using variable controller gains.
- **Include input saturation:** The unknown input saturation is incorporated in the control design with the help of quantitative performance analysis. Since the controller keeps the steady-state errors below 0.01 rad, this information is used to deal with the torque saturation.

As discussed earlier, the dynamics of a robot is complex and coupled, and finding the robot parameters is a complicated task. The underlying motivation is to find a controller that does not require a dynamical model with comparable performance as a model-based control. The controller requires only the states information, i.e., joint velocities, joint angles, and integral of joint angles, which is readily available from the encoder sensors. Hence, the controller is independent of the estimated model or regressor matrix. Based on intuition, simple linear differential equations are proposed to identify system dynamics during run-time.

Table 1.1: Comparison: (-) means not completely true/known. Q-P stands for Quantitative Performance.

Approach	**Reference(s)**	**Q-P**	**Easy tuning**	**Optimal**	**Model free**
Lyapunov-based	Slotine *et al.* [49]	✗	✗	✗	✗
Lyapunov-based	Arteaga *et al.* [59]	✓	✗	✗	✗
Lyapunov-based	Kai *et al.* [13]	✗	✗	✗	✓
Riccati inequality	Bascetta *et al.* [51]	✗	-	✗	✗
Lyapunov-based two controllers	Wang [54]	-	✗	✗	✗
Regressor-based	Pagilla *et al.* [20]	✗	✗	✗	✗
Computed-torque with modified outer-loop	Peng *et al.* [52]	-	✓	✗	✗
Lyapunov-based with an extra compensated term	Na *et al.* [53]	-	✗	✗	✗
Robust control using PID-type delay compensator	Alibeji *et al.* [76]	✗	✗	✗	✓
Sliding-mode	Yang *et al.* [66]	-	✗	✗	✓
Sliding-mode	Yu *et al.* [23, 24, 39, 84]	✗	✓	✗	✗
Sliding-mode	Sun *et al.* [4]	✗	✗	✗	✓
Radial basis function networks	Wang *et al.* [73]	✗	-	✗	✓
Finite-time	Liu *et al.* [60]	✓	✗	✓	✗
Orbital stability	Pchelkin *et al.* [29]	✓	✗	-	✗
PID control using HJI	Chung *et al.* [5, 10]	✓	✗	✓	✓
Inverse control	Zhang and Yan [74]	✗	✗	✗	✓
Fuzzy control	Fateh *et al.* [70]	✗	✗	✗	✓
Robust-adaptive control	Safaei *et al.* [65]	✗	✗	✓	✓
Intelligent PID control	Fliess *et al.* [81–83]	✗	✓	✗	✓
Robust control	Bechlioulis *et al.* [75, 85]	✓	✗	✗	✓
Inverse optimal control using $\mathcal{H}_\infty$	Proposed [1–3]	✓	✓	✓	✓

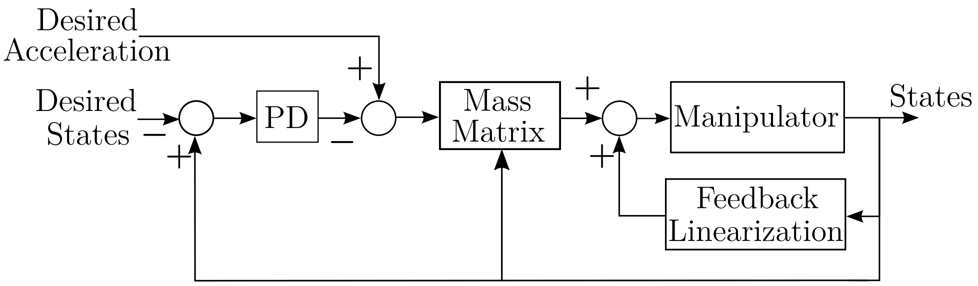


Figure 1.1: Feedback Linearization.

To understand the motivation behind the proposed controller, a brief explanation for computed-torque is first given in Chapter 2. Computed-torque is a control method that consists of two loops: the inner loop is the feedback linearization loop, which strives to cancel the nonlinear terms in the system model provided that the system dynamics are known. The outer loop is a simple proportional-derivative control that is applied on a linearized system. The proposed adaptive controller uses the same analogy of two feedback loops as shown in Fig. 1.1. The difference lies in the inner loop with the assumption that the system dynamics are unavailable. The adaptive inner loop tries to estimate the system dynamics using ordinary differential equations. In summary, the controller is designed using intuition of adaptive feedback linearization or adaptive computed-torque.

The optimality and robustness of the controller is validated by using the inverse-optimal control technique. The procedure is *"to find the optimal cost function given the proposed controller"*. The manipulator is represented as an $\mathcal{L}_2$-gain problem and HJI formulation is used to find the optimal cost function. The stability and parameters of the controller are evaluated using the input-to-state-stability (ISS) approach [90–92]. ISS is also used to reduce the effect of external disturbance. An important aspect of the proposed controller is the quantitative performance analysis. The controller guarantees that the steady-state error for the joint angles never exceeds a predefined maximum value. The dissertation also proposes techniques to incorporate input saturation and improve the transient response of the controller.

We have compared the proposed control method with twelve state-of-the-art controllers, where experimental evaluation is performed on five of these controllers [1, 3]. Simulations and experiments for some of the controllers were not feasible because of qualitative issues that can't be shown in comparison, e.g., how difficult it is to tune the controller or whether a controller is highly dependent on nominal model. The rest of the dissertation is divided as follows.

Chapter 2 starts with an introduction of computed-torque and some state-of-the-art adaptive control schemes. Next, an overview of the proposed adaptive control and system identification using least squares approach based on the adaptive control are given. Finally, a comparison of the proposed controller with a few state-of-the-art controllers is presented.

The design of a model-free adaptive control is explained in detail in **Chapter 3**. Robustness and stability of the controller are also proved using the $\mathcal{H}_\infty$ approach. We further devised a quantitative performance analysis for the controller in terms of the state error. Simulation and experiments are also performed to validate the proposed adaptive controller.

Chapter 4 is dedicated to the performance improvement methods and addressing the implementation issues. To improve the transient response, which is often the bottleneck of an adaptive controller, a few techniques are discussed. A simplified first-order adaptive controller is discussed for easy implementation without compromising the performance. The desired trajectory is also modified using a feedback loop to incorporate the input saturation. Finally, these enhancements in the adaptive controller are validated using experiments and simulations on a 3-DoF robot manipulator. In the end, conclusion and future work are summarized in **Chapter 5**.

2

Adaptive Control

After many decades of research, the control design for a robot manipulator is still a contemporary field of research [10, 13, 14, 64, 93]. The difficulty associated with robot control is due to the complexity of the robot dynamics and the strong coupling between the joints [8, 9]. Thus, high performance is often not achieved using a simple computed-torque technique due to the challenges mentioned in the previous chapter. For instance, only linear friction models are used in many control designs [42, 43], and as a consequence, many traits of friction are ignored, such as Coulomb and static friction. In addition, there are many reasons for parameter uncertainties, e.g., lumped parameter models or the effect of temperature variation on friction. The state-of-the-art control techniques for robots usually lack one or many of the following: optimality, robustness, easy-tuning, quantitative performance analysis, inclusion of input saturation and implementation on higher-order robots. Some state-of-the-art adaptive control schemes are discussed in this chapter that will give a motivation for the proposed controller.

The objective of this dissertation is to design and implement a simple robust-adaptive controller with few tuning parameters. Furthermore, the controller also addresses the challenges mentioned in Chapter 1. Similar to computed-torque, the proposed control system has two feedback loops: the inner loop is the adaptive part that performs the linearization. After the transient response, the outer loop performs like a proportional-derivative (PD) control for the linear system. The controller can alternatively be viewed as an intelligent proportional-integral-derivative (iPID) controller, where a single PID controller with fixed parameters is sufficient for any fully actuated manipulator [81, 83].

The control design has two main perspectives. The first objective is based on intuition, where simple state-dependent differential equations are used to achieve feedback linearization. The linearization does not require any system knowledge. In fact, the process starts with zero system information and during the execution of the controller, it tries to follow the real system trajectory. One advantage is that the system dynamics in such a case will also include all the unmodeled dynamics and external or internal disturbances, as long as they are bounded. Another advantage is that the controller only requires joint errors, joint velocities and their integrals for feedback loops. The second perspective of the controller is the mathematical validation of the control method, which is thoroughly explained in Chapter 3. It has been shown that the control design yields robustness and optimality with quantitative performance analysis.

The manipulator model used in the rest of the dissertation is introduced in Section 2.1.

The implementation of various adaptive controllers on robot manipulators is extensively available in literature and a complete review of all the controllers is not possible in the dissertation. In this chapter, a basic understanding of adaptive controllers is explained, starting with the model-reference inverse-dynamics control in section 2.2. Next, the regressor-based and regressor-free adaptive controllers are discussed in Section 2.3 and 2.4, respectively. In Section 2.5, the proposed controller is introduced along with the motivation, which is based on intuition. Since the inner loop of the proposed controller performs as a feedback linearization, closed-loop identification can be applied on the system data. For experimental validation of the proposed controller, a 2-degree-of-freedom (DoF) manipulator is used, which is described in Section 2.6. The section also shows a comparison with the state-of-the-art control schemes. The identification process using the least squares (LS) is explained in Section 2.7. For identification, 2- and 3-DoF manipulators are used in simulations and experiments.

2.1 Euler-Lagrange (E-L) Representation

The dynamics of a robot manipulator, in general, is represented by the E-L equations. For an n-link robot manipulator, the dynamics can be represented as

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

where $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is a symmetric positive definite inertia matrix, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{n\times n}$ is a matrix of centrifugal and Coriolis terms, $\boldsymbol{G} \in \mathbb{R}^{n\times 1}$ contains the gravitational terms acting on the robot, $\boldsymbol{F} \in \mathbb{R}^{n\times n}$ is a diagonal matrix representing approximate values of viscous friction, $\boldsymbol{q} \in \mathbb{R}^{n\times 1}$ is a vector of joint angles and $\boldsymbol{\tau} \in \mathbb{R}^{n\times 1}$ is a vector of input torques applied at each joint. In the presence of external disturbances, $\boldsymbol{\tau} = \boldsymbol{\tau}_{\mathrm{d}} + \boldsymbol{\tau}_{\mathrm{in}}$, where $\boldsymbol{\tau}_{\mathrm{d}}$ is the disturbance term and $\boldsymbol{\tau}_{\mathrm{in}}$ is the input torque. Throughout the dissertation, this representation of system dynamics is used. Following are the assumptions for the control system:

Assumption 1. *The desired trajectory $\boldsymbol{q}_{\mathrm{d}}$ is selected such that $\boldsymbol{q}_{\mathrm{d}}, \dot{\boldsymbol{q}}_{\mathrm{d}}$ and $\ddot{\boldsymbol{q}}_{\mathrm{d}} \in \mathcal{L}_\infty$.*

Assumption 2. *If $\boldsymbol{q}_{\mathrm{d}}$ and $\dot{\boldsymbol{q}}_{\mathrm{d}} \in \mathcal{L}_\infty$, then $\boldsymbol{M}(\boldsymbol{q}), \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}), \boldsymbol{G}(\boldsymbol{q})$ and $\boldsymbol{F}\dot{\boldsymbol{q}}$ are bounded.*

Assumption 3. *The disturbance term $\boldsymbol{\tau}_{\mathrm{d}}$ is bounded, i.e., $\boldsymbol{\tau}_{\mathrm{d}} \in \mathcal{L}_\infty$.*

Remark 1. *Only viscous friction, which is a linear model is used in simulation results. Coulomb and Stribeck frictions are not considered in the dynamical model, as they are usually considered as part of external disturbance. A continuous friction model is proposed in [46] to easily incorporate the complex friction characteristics.*

2.2 Computed-Torque

Computed-torque is a feedback linearization technique that is easy to implement and ensures stability in the presence of system mismatch and disturbances. The concept is to find an input torque for the system (2.1)

$$\boldsymbol{\tau}_{\mathrm{in}} = \boldsymbol{f}(\boldsymbol{q}, \dot{\boldsymbol{q}}), \tag{2.2}$$

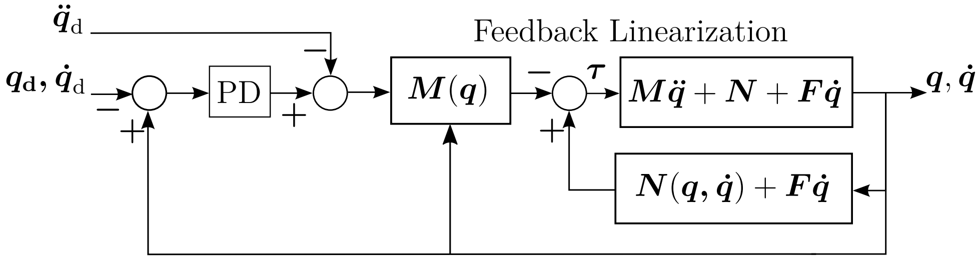


Figure 2.1: Computed-torque in the absence of disturbances.

such that the nonlinear dynamics of the manipulators are canceled by the inverse-dynamics of the controller. Such kind of feedback linearization might not be possible because of the challenges outlined in Chapter 1. The basic structure of the computed-torque control is shown in Fig. 2.1. The inner loop cancels the nonlinearities in the system and the outer loop is a simple PD control.

The linearization can only be achieved for manipulators assuming that the external disturbance is zero and the system dynamics are exactly known. Let

$$\boldsymbol{\tau}_{\text{in}} = \boldsymbol{M}\ddot{\boldsymbol{v}}_{\text{PD}} + \boldsymbol{N}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{F}\dot{\boldsymbol{q}}, \tag{2.3}$$

where $\boldsymbol{N}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q})$. By using (2.3), a new input $\ddot{\boldsymbol{v}}_{\text{PD}}$ is introduced, which makes the overall system decoupled and linear. Next for a smooth performance with desired transient and steady-state, $\ddot{\boldsymbol{v}}_{\text{PD}}$ must be selected such that the feedback system dynamics become a second-order linear differential equation. Let

$$\ddot{\boldsymbol{v}}_{\text{PD}} = \ddot{\boldsymbol{q}}_{\text{d}} - \boldsymbol{K}_{\text{d}}\dot{\boldsymbol{e}} - \boldsymbol{K}_{\text{p}}\boldsymbol{e}, \tag{2.4}$$

where $\boldsymbol{q}_{\text{d}}$ is the desired trajectory, $\boldsymbol{e} = \boldsymbol{q} - \boldsymbol{q}_{\text{d}}$, and $\boldsymbol{K}_{\text{d}}$ and $\boldsymbol{K}_{\text{p}} \in \mathbb{R}^{n\times n}$ are PD gains [8]. Substituting (2.3) and (2.4) into (2.1), the following error dynamics are achieved:

$$\ddot{\boldsymbol{e}} + \boldsymbol{K}_{\text{d}}\dot{\boldsymbol{e}} + \boldsymbol{K}_{\text{p}}\boldsymbol{e} = \boldsymbol{0}. \tag{2.5}$$

Since the motors are controlled by current, $\boldsymbol{\tau}_{\text{in}}$ can be replaced by

$$\boldsymbol{n}_r\boldsymbol{k}_\tau\boldsymbol{i}_a = \boldsymbol{M}(\ddot{\boldsymbol{q}}_{\text{d}} - \boldsymbol{K}_{\text{d}}\dot{\boldsymbol{e}} - \boldsymbol{K}_{\text{p}}\boldsymbol{e}) + \boldsymbol{N} + \boldsymbol{F}\dot{\boldsymbol{q}}, \tag{2.6}$$

where $\boldsymbol{k}_\tau$ is the torque constant, $\boldsymbol{i}_a$ is the current vector and $\boldsymbol{n}_r$ is a vector that represents the gear ratio for individual joints.

As we assume, if there are no unmodeled dynamics, the desired performance in terms of steady-state error, percent overshoot etc., can be achieved by using the pole-placement technique [94]. If there is any mismatch, which is inevitable in experiments, the right-hand side of (2.5) will be non-zero. Although the stability can still be assured, nevertheless, the performance will be deteriorated.

2.3 Regressor-Based Adaptive Control

As the system dynamics are not entirely known, a proper feedback linearization like computed-torque is not possible. Therefore, the new input torque with an estimated model can be represented as

$$\boldsymbol{\tau}_{\text{in}} = \hat{\boldsymbol{M}}(\ddot{\boldsymbol{q}}_{\text{d}} - \boldsymbol{K}_{\text{d}}\dot{\boldsymbol{e}} - \boldsymbol{K}_{\text{p}}\boldsymbol{e}) + \hat{\boldsymbol{N}} + \hat{\boldsymbol{F}}\dot{\boldsymbol{q}}, \tag{2.7}$$

where $(\hat{\cdot})$ represents the estimate of the corresponding matrices or vectors. By using the estimated model (2.7), the system dynamics are described by

$$\ddot{\boldsymbol{e}} + \boldsymbol{K}_{\mathrm{d}}\dot{\boldsymbol{e}} + \boldsymbol{K}_{\mathrm{p}}\boldsymbol{e} = -\hat{\boldsymbol{M}}^{\boldsymbol{-1}}(\tilde{\boldsymbol{M}}\ddot{\boldsymbol{q}} + \tilde{\boldsymbol{N}} + \tilde{\boldsymbol{F}}\dot{\boldsymbol{q}}), \tag{2.8}$$

where

$$\tilde{\boldsymbol{M}} = \boldsymbol{M} - \hat{\boldsymbol{M}},$$

$$\tilde{\boldsymbol{N}} = \boldsymbol{N} - \hat{\boldsymbol{N}}$$

and

$$\tilde{\boldsymbol{F}} = \boldsymbol{F} - \hat{\boldsymbol{F}}.$$

In this section, the disturbance $\boldsymbol{\tau}_{\mathrm{d}}$ is assumed to be zero. In the presence of any mismatch, the above equation (2.8) can still give a stable result using the Lyapunov stability equation. The major issue is the poor performance of the system and depending on the mismatch, the performance deteriorates.

Many of the state-of-the-art control schemes use the regressor approach to deal with uncertainties. The structure of the system is known from the E-L equations. Ignoring the external disturbances and unmodeled dynamics, the uncertainties lie only in the constant parameters of the system. A regressor-based control converts system dynamics into a known regressor matrix and an unknown parameter vector. The parameter vector usually consists of masses, lengths and friction constants of each joint. For the system equation (2.1), the regressor form is

$$\boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\boldsymbol{p} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{N}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{F}\dot{\boldsymbol{q}}, \tag{2.9}$$

where $\boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ be a nonlinear, known regression matrix with proper dimensions and $\boldsymbol{p}$ is the unknown constant parameter vector. An example of a 2-degree-of-freedom (DoF) manipulator is given in Appendix B1.

Using the regressor matrix (2.9) with $\hat{\boldsymbol{p}}$ as estimation of $\boldsymbol{p}$ in (2.7), and let $\tilde{\boldsymbol{p}} = \boldsymbol{p} - \hat{\boldsymbol{p}}$ is the estimation error, then the error dynamics become

$$\ddot{\boldsymbol{e}} + \boldsymbol{K}_{\mathrm{d}}\dot{\boldsymbol{e}} + \boldsymbol{K}_{\mathrm{p}}\boldsymbol{e} = -\hat{\boldsymbol{M}}^{-1}\boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\tilde{\boldsymbol{p}}. \tag{2.10}$$

To get a stable system i-e., $\boldsymbol{e} \to \boldsymbol{0}$, a suitable value for the estimated parameter vector $\hat{\boldsymbol{p}}$ must be selected. Let $\boldsymbol{x'} = [\boldsymbol{e}^T \ \dot{\boldsymbol{e}}^T]^{\boldsymbol{T}}$ represents the state error, then (2.10) can be written as

$$\dot{\boldsymbol{x}}' = \boldsymbol{A'x'} - \boldsymbol{B'}\hat{\boldsymbol{M}}^{-1}\boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\tilde{\boldsymbol{p}}, \tag{2.11}$$

where

$$\boldsymbol{A'} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I}_n \\ -\boldsymbol{K}_{\mathrm{p}} & -\boldsymbol{K}_{\mathrm{d}} \end{bmatrix} \in \mathbb{R}^{2n \times 2n}, \boldsymbol{B'} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{I}_n \end{bmatrix} \in \mathbb{R}^{2n \times n}$$

and $\boldsymbol{I}_n$ is the identity matrix. Consider the following Lyapunov function

$$V(\boldsymbol{x'}, \tilde{\boldsymbol{p}}) = \frac{1}{2}\boldsymbol{x'}^T\boldsymbol{P}_1\boldsymbol{x'} + \frac{1}{2}\tilde{\boldsymbol{p}}^T\boldsymbol{P}_2\tilde{\boldsymbol{p}}, \tag{2.12}$$

where $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ are symmetric, positive definite matrices. Taking the derivative of (2.12) along (2.11) yields

$$\dot{V} = -\frac{1}{2}\boldsymbol{x'}^T\boldsymbol{Q'x'} - \tilde{\boldsymbol{p}}^T[(\hat{\boldsymbol{M}}^{-1}\boldsymbol{Y})^T\boldsymbol{B'}^T\boldsymbol{P}_1\boldsymbol{x'} + \boldsymbol{P}_2\dot{\hat{\boldsymbol{p}}}]. \tag{2.13}$$

Since $\boldsymbol{A'}$ is a Hurwitz matrix [31], there exists a symmetric positive definite matrix $\boldsymbol{Q'}$ satisfying $\boldsymbol{A'}^T\boldsymbol{P}_1 + \boldsymbol{P}_1\boldsymbol{A'} = -\boldsymbol{Q'}$. In order to cancel the possibly non-negative terms, the update law for $\hat{\boldsymbol{p}}$ must be selected as

$$\dot{\hat{\boldsymbol{p}}} = -\boldsymbol{P}_2^{-1}(\hat{\boldsymbol{M}}^{-1}\boldsymbol{Y})^T\boldsymbol{B'}^T\boldsymbol{P}_1\boldsymbol{x'}. \tag{2.14}$$

For an asymptotic stable system, the function $\dot{V}$ must be negative. Substituting $\dot{\hat{\boldsymbol{p}}}$ in (2.13), the Lyapunov function derivative becomes $\dot{V} = -\frac{1}{2}\boldsymbol{x'}^T\boldsymbol{Q'}\boldsymbol{x'} \leqslant 0$, which ensures asymptotic tracking of the reference trajectory.

2.3.1 Slotine and Li's Adaptive Control

The main problem with the regressor technique presented above is that the inverse of the inertia matrix is often not feasible and in some cases results into a singular matrix. Similarly, the use of joint-acceleration in the feedback loop results in noisy output. Although nowadays, the joint encoders have a high-resolution, however, the second derivative of the joint angle still results in noisy data because of high sampling time. To address the above issues, Slotine *et al.* [49] proposed a modification to the regressor technique.

The Coriolis and centrifugal matrix $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$ can be determined to make $\dot{\boldsymbol{M}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$ a skew-symmetric [9]. Let $\boldsymbol{s} = \dot{\boldsymbol{e}} + \triangle\boldsymbol{e}$, where $\triangle \in \mathbb{R}^{n\times n}$ is a diagonal matrix with positive eigenvalues, the system dynamics in (2.1) transform into

$$\boldsymbol{M}\dot{\boldsymbol{s}} + \boldsymbol{N} + \boldsymbol{F}\boldsymbol{s} + \boldsymbol{M}\dot{\boldsymbol{v}}_{\mathrm{SL}} + \boldsymbol{F}\boldsymbol{v}_{\mathrm{SL}} = \boldsymbol{\tau}_{\mathrm{in}} \tag{2.15}$$

and $\boldsymbol{v}_{\mathrm{SL}} = \dot{\boldsymbol{q}}_{\mathrm{d}} - \triangle\boldsymbol{e}$. Taking the torque $\boldsymbol{\tau}_{\mathrm{in}}$ as

$$\boldsymbol{\tau}_{\mathrm{in}} = \hat{\boldsymbol{M}}\dot{\boldsymbol{v}}_{\mathrm{SL}} + \hat{\boldsymbol{F}}\boldsymbol{v}_{\mathrm{SL}} + \hat{\boldsymbol{N}} - \boldsymbol{K}_{\mathrm{D}}\boldsymbol{s}, \tag{2.16}$$

where $\boldsymbol{K}_{\mathrm{D}}$ is a positive definite matrix, and (2.15) takes the form

$$\boldsymbol{M}\dot{\boldsymbol{s}} + \boldsymbol{F}\boldsymbol{s} + \boldsymbol{K}_{\mathrm{D}}\boldsymbol{s} = -\tilde{\boldsymbol{M}}\dot{\boldsymbol{v}}_{\mathrm{SL}} - \tilde{\boldsymbol{F}}\boldsymbol{v}_{\mathrm{SL}} - \tilde{\boldsymbol{N}}. \tag{2.17}$$

Replacing the right-hand side of (2.17) with the regressor form, we get

$$\boldsymbol{M}\dot{\boldsymbol{s}} + \boldsymbol{F}\boldsymbol{s} + \boldsymbol{K}_{\mathrm{D}}\boldsymbol{s} = -\boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{v}_{\mathrm{SL}},\dot{\boldsymbol{v}}_{\mathrm{SL}})\tilde{\boldsymbol{p}}. \tag{2.18}$$

Since $\boldsymbol{M}$ is a positive definite matrix, the Lyapunov function can be chosen as

$$V(\boldsymbol{x'},\tilde{\boldsymbol{p}}) = \frac{1}{2}\boldsymbol{s}^T\boldsymbol{M}\boldsymbol{s} + \frac{1}{2}\tilde{\boldsymbol{p}}^T\boldsymbol{P}_2\tilde{\boldsymbol{p}}. \tag{2.19}$$

The derivative of the cost function

$$\dot{V} = -\boldsymbol{s}^T\boldsymbol{K}_{\mathrm{D}}\boldsymbol{s} - \tilde{\boldsymbol{p}}^T\Big(\boldsymbol{P}_2\dot{\hat{\boldsymbol{p}}} + \boldsymbol{Y}^T\boldsymbol{s}\Big). \tag{2.20}$$

will be negative if $\hat{\boldsymbol{p}}$ is formulated such that the possible positive terms are canceled:

$$\dot{\hat{\boldsymbol{p}}} = -\boldsymbol{P}_2^{-1}\boldsymbol{Y}^T(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{v}_{\mathrm{SL}},\dot{\boldsymbol{v}}_{\mathrm{SL}})\boldsymbol{s}. \tag{2.21}$$

An extension of the above adaptive control is presented by Ciliz and Narendra [95], where an indirect approach is used to separate identification and control. For that purpose, multiple models switching control was used. Although the controller improves the transient performance, the joint acceleration requirement in the feedback is the main drawback.

2.4 Regressor-Free Adaptive Control

The regressor matrix requires an estimated model of the system and also the evaluation of the regressor matrix is difficult, especially for a higher DoF robot. To avoid the use of the regressor matrix and to make the adaptive control model-free, Huang *et al.* [64] proposed a function approximation technique (FAT)-based adaptive control. This approach is based on approximation of system dynamics by linear orthogonal basis functions. The matrices $\boldsymbol{M}$, $\boldsymbol{N}$ and $\boldsymbol{F}$ are approximated by weighted sums of some orthogonal linear functions

$$\boldsymbol{M} = \boldsymbol{W}_M^T \boldsymbol{Z}_M, \quad \boldsymbol{F} = \boldsymbol{W}_F^T \boldsymbol{Z}_F, \quad \boldsymbol{N} = \boldsymbol{W}_N^T \boldsymbol{Z}_N,$$

where $\boldsymbol{W}_M \in \mathbb{R}^{n^2\beta_M \times n}$, $\boldsymbol{W}_F \in \mathbb{R}^{n^2\beta_F \times n}$ and $\boldsymbol{W}_N \in \mathbb{R}^{n\beta_N \times n}$ are weighting matrices and $\boldsymbol{Z}_M \in \mathbb{R}^{n^2\beta_M \times n}$, $\boldsymbol{Z}_F \in \mathbb{R}^{n^2\beta_F \times n}$ and $\boldsymbol{Z}_N \in \mathbb{R}^{n\beta_N \times n}$ are matrices of the basis functions. The number $\beta_{(.)}$ represents the number of basis functions used for approximation. The input in (2.16) is modified accordingly as

$$\boldsymbol{\tau}_{\text{in}} = \hat{\boldsymbol{W}}_M^T \boldsymbol{Z}_M \dot{\boldsymbol{v}}_{\text{SL}} + \hat{\boldsymbol{W}}_F^T \boldsymbol{Z}_F \boldsymbol{v}_{\text{SL}} + \hat{\boldsymbol{W}}_N^T \boldsymbol{Z}_N - \boldsymbol{K}_{\text{D}} \boldsymbol{s}. \tag{2.22}$$

Based on the Lyapunov function with $\tilde{\boldsymbol{W}}_{(.)} = \boldsymbol{W}_{(.)} - \hat{\boldsymbol{W}}_{(.)}$,

$$V(\boldsymbol{s}, \tilde{\boldsymbol{W}}_M, \tilde{\boldsymbol{W}}_N, \tilde{\boldsymbol{W}}_F) = \frac{1}{2}\boldsymbol{s}^T \boldsymbol{M} \boldsymbol{s} + \frac{1}{2}\text{tr}(\tilde{\boldsymbol{W}}_M^T \boldsymbol{Q}_M \tilde{\boldsymbol{W}}_M + \tilde{\boldsymbol{W}}_F^T \boldsymbol{Q}_F \tilde{\boldsymbol{W}}_F + \tilde{\boldsymbol{W}}_N^T \boldsymbol{Q}_N \tilde{\boldsymbol{W}}_N),$$

the update laws are derived as [11]

$$\begin{aligned}
\dot{\hat{\boldsymbol{W}}}_M &= -\boldsymbol{Q}_M^{-1} \boldsymbol{Z}_M \dot{\boldsymbol{v}}_{\text{SL}} \boldsymbol{s}^T, \\
\dot{\hat{\boldsymbol{W}}}_F &= -\boldsymbol{Q}_F^{-1} \boldsymbol{Z}_F \boldsymbol{v}_{\text{SL}} \boldsymbol{s}^T, \\
\dot{\hat{\boldsymbol{W}}}_N &= -\boldsymbol{Q}_N^{-1} \boldsymbol{Z}_N \boldsymbol{s}^T.
\end{aligned} \tag{2.23}$$

To further improve the control law, Kai *et al.* [13] replaced $\boldsymbol{M^{-1}Y(q, \dot{q}, \ddot{q})p}$ by $\boldsymbol{\psi}$, where $\boldsymbol{\psi} = \boldsymbol{W}^T \boldsymbol{Z}$. Following the same procedure for finding the control law by analyzing the Lyapunov function, we obtain

$$\dot{\hat{\boldsymbol{W}}} = -\boldsymbol{Q}^{-1} \boldsymbol{Z} \boldsymbol{x}^T \boldsymbol{P} \boldsymbol{B}, \tag{2.24}$$

where $\boldsymbol{P} \in \mathbb{R}^{2n \times 2n}$ is a positive definite matrix. The FAT method can be a linear combination of cosine and sine functions with proper weights. The regressor-free approach does not require a system model, so the uncertainties and external disturbances can be considered as part of the basis functions.

In theory, the controller should perform robustly for bounded disturbances. A drawback of the approach is that it requires a high computational cost because of large number of orthogonal basis functions. Another major drawback in the Lyapunov based control is the tuning of the controller parameters and it would require trial and error to find feasible controller parameters.

2.5 Proposed Model-Free Robust-Adaptive Control

The adaptive controllers explained so far were designed using the Lyapunov stability technique. Lyapunov-based controllers make sure that the joint error approaches zero, however, there is no assurance of $\hat{\boldsymbol{p}}$ to approach $\boldsymbol{p}$. These control approaches only focus on the stability of the system and the performance is compromised. In this section, we give a brief introduction to the proposed adaptive controller.

Let $\boldsymbol{\phi} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{N}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{F}\dot{\boldsymbol{q}} - \ddot{\boldsymbol{q}} - \boldsymbol{\tau_d}$, where $\boldsymbol{\tau_d}$ represents disturbances and unmodeled parameters. Then the input torque is selected as [13]

$$\boldsymbol{\tau}_{\text{in}} = \hat{\boldsymbol{\phi}} + \ddot{\boldsymbol{q}}_{\text{d}} - \boldsymbol{K}_{\text{d}}\dot{\boldsymbol{e}} - \boldsymbol{K}_{\text{p}}\boldsymbol{e}. \tag{2.25}$$

Using input as (2.25) for the robotic system (2.1), the error dynamics become

$$\ddot{\boldsymbol{e}} + \boldsymbol{K}_{\text{d}}\dot{\boldsymbol{e}} + \boldsymbol{K}_{\text{p}}\boldsymbol{e} = \hat{\boldsymbol{\phi}} - \boldsymbol{\phi}, \tag{2.26}$$

where $\hat{\boldsymbol{\phi}}$ is the estimate of $\boldsymbol{\phi}$. Next step is to find the input $\hat{\boldsymbol{\phi}}$ for the error dynamics (2.26).

Remark 2. *The error dynamics* (2.26) *is an alternate representation of the dynamics in* (2.8)*.*

Remark 3. *Similar to the intelligent PD (iPD) controller [83], the proposed input torque* (2.25) *has two important terms along with the desired joint acceleration. The first part is the system estimation vector* $\hat{\boldsymbol{\phi}}$ *and the second is the PD controller* $-\boldsymbol{K}_{\text{d}}\dot{\boldsymbol{e}} - \boldsymbol{K}_{\text{p}}\boldsymbol{e}$*. However, we have used a different approach for the approximation of the system dynamics* $\boldsymbol{\phi}$*. The advantage of an iPD controller is that a single controller gain is sufficient for any bounded system parameters variations or disturbances. The reason is that after linearization using the online system estimation, all systems end up into same linear error dynamics of* (2.5)*.*

Theorem 1 (Adaptive law)**.** *Let* $\tilde{\boldsymbol{\phi}} = \hat{\boldsymbol{\phi}} - \boldsymbol{\phi}$ *and*

$$\frac{\mathrm{d}^m\hat{\boldsymbol{\phi}}}{\mathrm{dt}^m} = -\sum_{i=0}^{m-1} a_{\text{i}} \frac{\mathrm{d}^i\tilde{\boldsymbol{\phi}}}{\mathrm{dt}^i}, \tag{2.27}$$

with the assumption that $\frac{\mathrm{d}^m\boldsymbol{\phi}}{\mathrm{dt}^m} = 0$*, then the control law* (2.25) *leads to asymptotic stability of the system* (2.1) *with respect to joint and estimation error.*

Proof. The stability of the proposed adaptive law is proven by using the Lyapunov equation. Using (2.27) and the assumption $\frac{\mathrm{d}^m\boldsymbol{\phi}}{\mathrm{dt}^m} = 0$, we get

$$\sum_{i=0}^{m} a_{\text{i}} \frac{\mathrm{d}^i\tilde{\boldsymbol{\phi}}}{\mathrm{dt}^i} = \boldsymbol{0}, \tag{2.28}$$

where $a_{\text{m}} = 1$. By selecting proper values of a_{i} (see Chapter 3), it is clear that $\tilde{\boldsymbol{\phi}}$ will approach zero according to the above equation providing that all the poles of (2.28) are in the left half of the s-plane (Laplace [94, 96]). The next step is to prove the stability of the controller using the above equation.

Equation (2.28) can be represented in matrix form as

$$\dot{\boldsymbol{z}} = \boldsymbol{H}\boldsymbol{z},$$

where

$$\boldsymbol{z} = [\tilde{\boldsymbol{\phi}}^T \quad \frac{\mathrm{d}\tilde{\boldsymbol{\phi}}^T}{\mathrm{d}t} \quad \dots \quad \frac{\mathrm{d}^{m-1}\tilde{\boldsymbol{\phi}}^T}{\mathrm{d}t^{m-1}}]^T \in \mathbb{R}^{mn\times 1}$$

and $\boldsymbol{H} \in \mathbb{R}^{mn\times mn}$ is the Jordan canonical form of (2.28). Similarly the error dynamics (2.26) can be expressed in the state space form as

$$\dot{\bar{\boldsymbol{x}}} = \bar{\boldsymbol{A}}\bar{\boldsymbol{x}} + \bar{\boldsymbol{B}}\boldsymbol{z},$$

where

$$\bar{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I}_n \\ -\boldsymbol{K}_\mathrm{p} & -\boldsymbol{K}_\mathrm{d} \end{bmatrix} \in \mathbb{R}^{2n\times 2n},$$

$$\bar{\boldsymbol{B}} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{I}_n & \boldsymbol{0} & \dots & \boldsymbol{0} \end{bmatrix} \in \mathbb{R}^{2n\times mn}$$

and

$$\bar{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{e}^T & \dot{\boldsymbol{e}}^T \end{bmatrix}^T \in \mathbb{R}^{2n\times 1}.$$

Let

$$V(\bar{\boldsymbol{x}}, \boldsymbol{z}) = \frac{1}{2}\bar{\boldsymbol{x}}^T\boldsymbol{P}_1\bar{\boldsymbol{x}} + \frac{1}{2}\boldsymbol{z}^T\boldsymbol{P}_2\boldsymbol{z},$$

be a Lyapunov candidate, where $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ are symmetric, positive definite matrices with proper dimensions. The derivative of the above Lyapunov function is

$$\dot{V}(\bar{\boldsymbol{x}}, \boldsymbol{z}) = -\frac{1}{2}\bar{\boldsymbol{x}}^T\boldsymbol{Q}_1\bar{\boldsymbol{x}} - \frac{1}{2}\boldsymbol{z}^T\boldsymbol{Q}_2\boldsymbol{z} + \boldsymbol{z}^T\bar{\boldsymbol{B}}^T\boldsymbol{P}_1\bar{\boldsymbol{x}}. \tag{2.29}$$

As $\bar{\boldsymbol{A}}$ and $\boldsymbol{H}$ are Hurwitz matrices, $\bar{\boldsymbol{A}}^T\boldsymbol{P}_1 + \boldsymbol{P}_1\bar{\boldsymbol{A}} = -\boldsymbol{Q}_1$ and $\boldsymbol{H}^T\boldsymbol{P}_2 + \boldsymbol{P}_2\boldsymbol{H} = -\boldsymbol{Q}_2$ with positive definite matrices $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$. It is shown in the Appendix B2 that (2.29) is indeed negative. From the above equation, it is evident that both the joint errors and system mismatches are decreasing and the equilibrium point is $\bar{\boldsymbol{x}}, \tilde{\boldsymbol{\phi}} = \boldsymbol{0}$. □

Remark 4. *For $m = 1$ and $m = 2$, the control laws are found by combining* (2.26) *and* (2.27)*:*

$$m = 1: \qquad \hat{\boldsymbol{\phi}} = -a_0\Big(\dot{\boldsymbol{e}} + \boldsymbol{K}_\mathrm{d}\boldsymbol{e} + \boldsymbol{K}_\mathrm{p}\int \boldsymbol{e}\ \mathrm{d}t\Big), \tag{2.30}$$

$$\begin{aligned} m = 2: \qquad \hat{\boldsymbol{\phi}} = &-a_1(\dot{\boldsymbol{e}} + \boldsymbol{K}_\mathrm{d}\boldsymbol{e} + \boldsymbol{K}_\mathrm{p}\int \boldsymbol{e}\ \mathrm{d}t) - \\ &a_0(\boldsymbol{e} + \boldsymbol{K}_\mathrm{d}\int \boldsymbol{e}\ \mathrm{d}t + \boldsymbol{K}_\mathrm{p}\iint \boldsymbol{e}\ \mathrm{d}t^2). \end{aligned} \tag{2.31}$$

A thorough stability proof is given in the next chapter. The main motivation in selecting the control law as in (2.27) is to ensure that $\tilde{\boldsymbol{\phi}}$ converges to zero. The intuition for using such a controller is given in Appendix B2.

Remark 5. *The motivation behind Theorem 1 is not to prove stability as it is thoroughly proved in the next chapter. Here, only an intuitive understanding of the proposed controller is given, which is, "for high values of* a_i*, the mismatch will eventually approach to zero. Hence, resulting in a feedback linearized or iPD controller" as discussed in Section 2.6.*

Remark 6. *The assumption in Theorem 1, is not essential for the stability of the system with the proposed controller. The motivation is to show that both the state error and system mismatch approach to zero for better performance.*

The first-order controller $m = 1$ can be alternatively represented as a classical PID controller using the linear quadratic regulator technique. Let

$$\boldsymbol{\tau}_\mathrm{in} = \hat{\boldsymbol{\phi}} + \ddot{\boldsymbol{q}}_\mathrm{d}. \tag{2.32}$$

Using (2.32) for the robotic system (2.1), the error dynamics are

$$\ddot{\boldsymbol{e}} = \hat{\boldsymbol{\phi}} - \boldsymbol{\phi}. \tag{2.33}$$

Remark 7. *From the viewpoint of the pole placement technique [96], the two terms* $-\boldsymbol{K}_\mathrm{d}\dot{\boldsymbol{e}} - \boldsymbol{K}_\mathrm{p}\boldsymbol{e}$ *from* (2.25) *will now be part of input* $\hat{\boldsymbol{\phi}}$*.*

Let

$$\begin{bmatrix} \boldsymbol{e} \\ \dot{\boldsymbol{e}} \\ \ddot{\boldsymbol{e}} \end{bmatrix} = \begin{bmatrix} 0 & \boldsymbol{I} & 0 \\ 0 & 0 & \boldsymbol{I} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \int \boldsymbol{e} \\ \boldsymbol{e} \\ \dot{\boldsymbol{e}} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \boldsymbol{I} \end{bmatrix} \hat{\boldsymbol{\phi}} - \begin{bmatrix} 0 \\ 0 \\ \boldsymbol{I} \end{bmatrix} \boldsymbol{\phi},$$

which can be written as

$$\dot{\boldsymbol{x}}'' = \boldsymbol{A}''\boldsymbol{x}'' + \boldsymbol{B}''\boldsymbol{u} + \boldsymbol{B}''\boldsymbol{d}. \tag{2.34}$$

By using the optimal control (Appendix A2), the input is

$$\boldsymbol{u} = -\boldsymbol{K}''\boldsymbol{x}'', \quad \boldsymbol{K}'' = [\boldsymbol{k}_1'' \quad \boldsymbol{k}_2'' \quad \boldsymbol{k}_3'']. \tag{2.35}$$

Equating (2.30) and (2.35) along with their corresponding $\boldsymbol{\tau_{in}}$,

$$\boldsymbol{k}_1'' = a_0\boldsymbol{K}_\mathrm{p},$$
$$\boldsymbol{k}_2'' = a_0\boldsymbol{K}_\mathrm{d} + \boldsymbol{K}_\mathrm{p},$$
$$\boldsymbol{k}_3'' = a_0\boldsymbol{I} + \boldsymbol{K}_\mathrm{d}.$$

Remark 8. *A similar analogy can be used to find the optimal controls for the second and higher-order controllers. Although, the above proof simplifies the stability of the adaptive controller but to get the insight on the performance analysis,* (2.25) *is used in the rest of the dissertation.*

2.6 Results

In this section, we have shown the simulation and experimental results of the proposed adaptive controller on a 2-DoF robot manipulator (Fig. 2.2 and Appendix B1). A comparison with other control techniques is also included in Section 2.6.1 and 2.6.2. The results are already published in [3]. The desired trajectory is a circle in Cartesian coordinates and the corresponding joint-space trajectory is shown in Fig. 2.3.

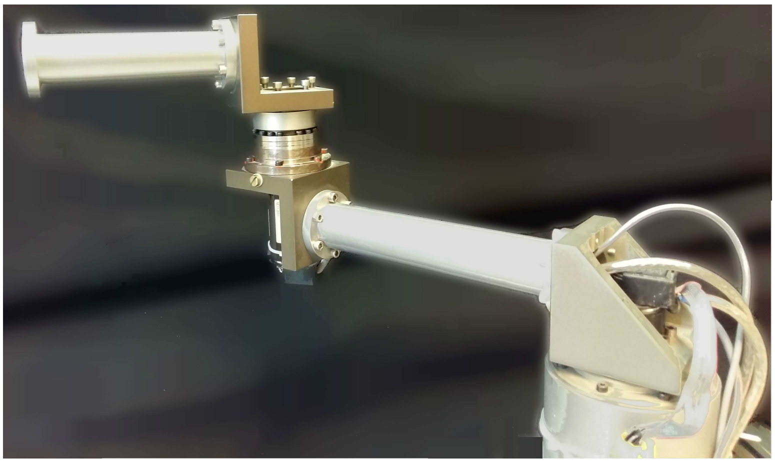


Figure 2.2: Two-DoF serial robot used in experiments.

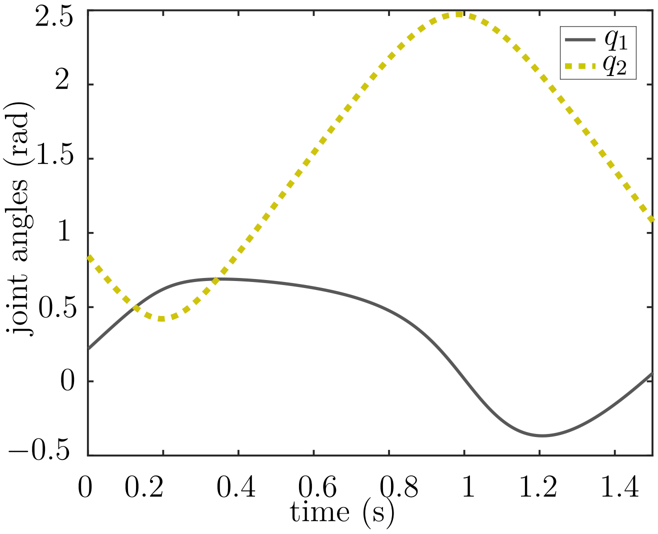
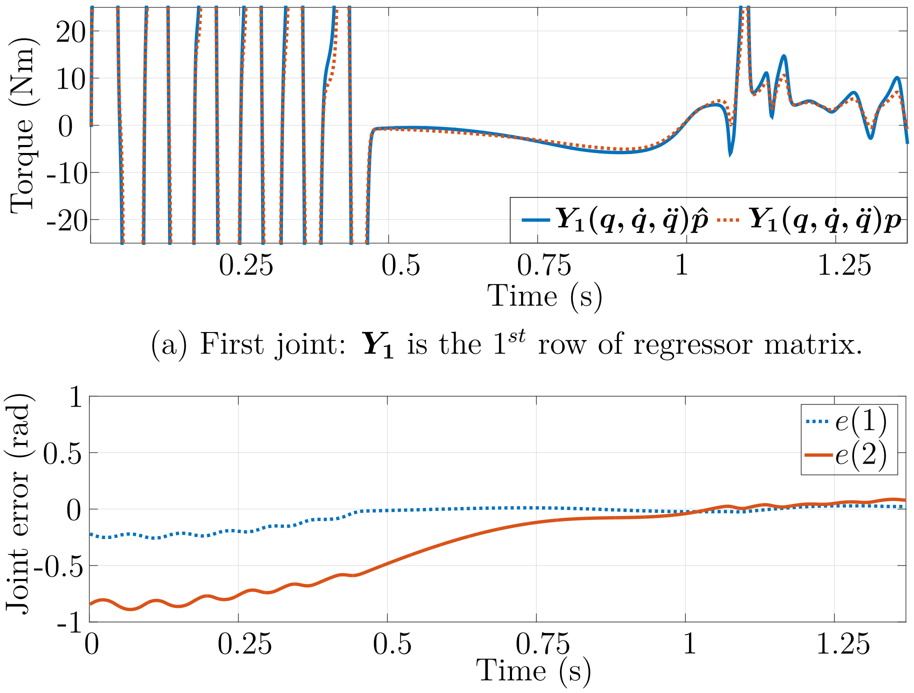


Figure 2.3: Joint angles for the desired circular trajectory.



(a) First joint: $\boldsymbol{Y_1}$ is the 1^{st} row of regressor matrix.

(b) Error in joint angles.

Figure 2.4: Results of regressor-based adaptive control: this technique requires inverse of inertia matrix ($\hat{\boldsymbol{M}}$). Tracking of the desired trajectory is sensitive to any change in controller parameters.

2.6.1 Simulation Results

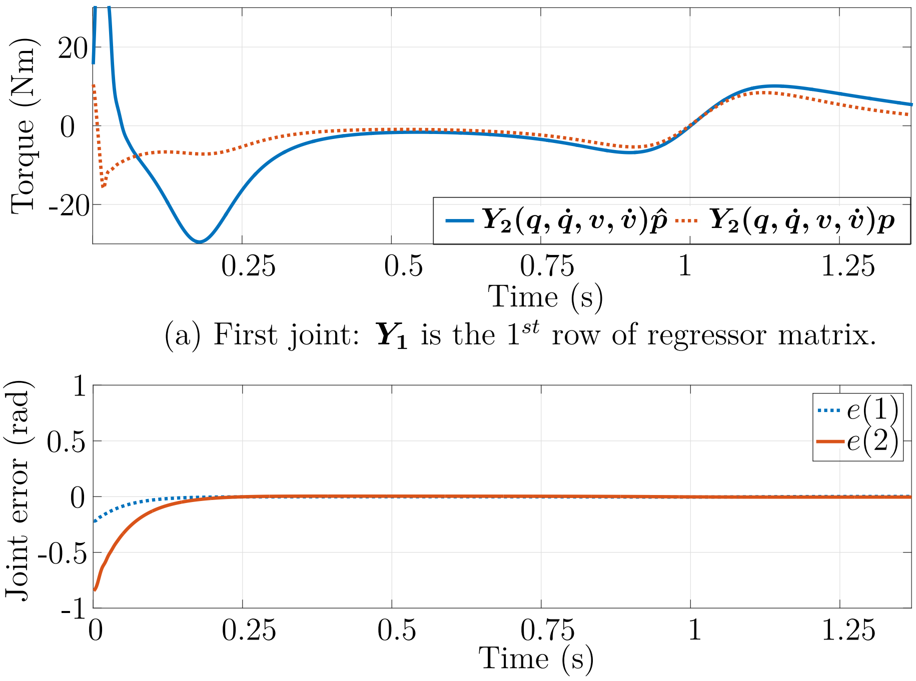
For the regressor-based method of Section 2.3, the inverse of the inertial matrix limits the overall performance of the controller. The matrices in (2.12) are chosen as $\boldsymbol{P}_2 = \boldsymbol{I}$ and

$$\boldsymbol{P}_1 = \begin{bmatrix} 6.05 & 0 & 0.008 & 0 \\ 0 & 2.13 & 0 & 0.0025 \\ 0.008 & 0 & 0.01 & 0 \\ 0 & 0.0025 & 0 & 0.01 \end{bmatrix}.$$

Fig. 2.4-a represents $\boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\boldsymbol{p}$ and $\boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\hat{\boldsymbol{p}}$ while their difference is shown in Fig. 2.8. An upper bound on the joint torques should be applied to avoid singularity problems. Although Fig. 2.4-b shows that the joint angle tries to follow the desired trajectory to satisfy the Lyapunov criteria, there is no guarantee on $\boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\tilde{\boldsymbol{p}}$ to approach zero (see Fig. 2.8). The approach by Slotine *et al.* in Fig. 2.5 shows an improved performance as compared to the simple regression because the inverse of the inertia matrix is not required. However, there are various design parameters in this approach, which require trial and error to adjust the controller. From Section 2.3.1, $\boldsymbol{P}_2 = \boldsymbol{I}$ and

$$\triangle = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}, \quad \boldsymbol{K}_{\mathrm{D}} = \begin{bmatrix} 200 & 0 \\ 0 & 150 \end{bmatrix}.$$

The function approximation technique (FAT)-based control from Section 2.4 in Fig. 2.6 has the advantage of using linear approximated models of the system. However, this technique



(a) First joint: $\boldsymbol{Y_1}$ is the 1^{st} row of regressor matrix.

(b) Error in joint angles.

Figure 2.5: Results of Slotine and Li's adaptive control. As clear from Fig. (a), the estimated model dynamics do not follow the original model. Reason is that Lyapunov function only shows the tracking error to be a decreasing function.

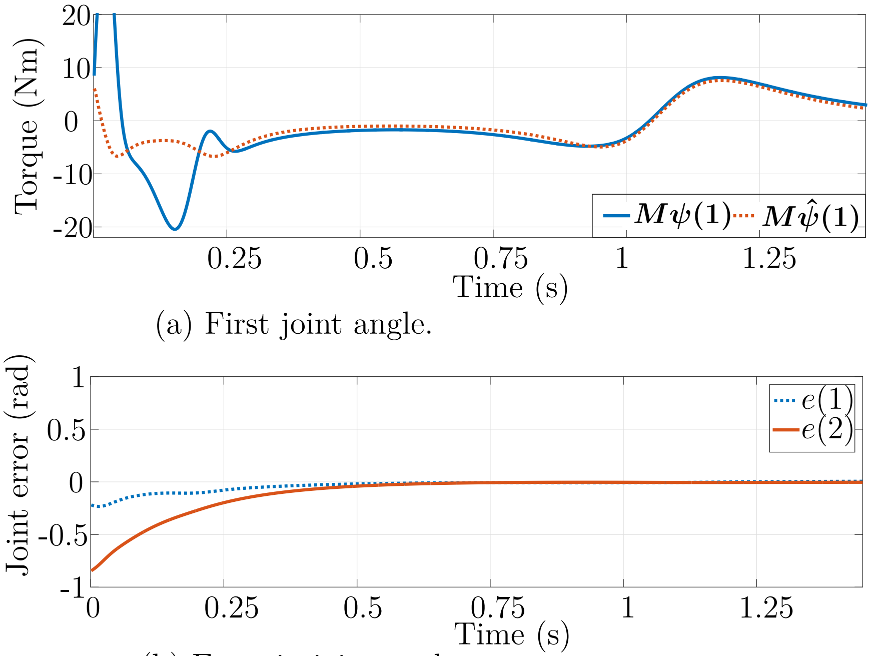
uses large matrices for the approximation, thus increasing the complexity of the controller. Also, some information is lost because of linearization. The design parameters of the FAT-based approach are taken from [13].

As shown in Fig. 2.8, $\boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\tilde{\boldsymbol{p}}$, $\boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{v}_{\mathrm{SL}}, \dot{\boldsymbol{v}}_{\mathrm{SL}})\tilde{\boldsymbol{p}}$ and $\boldsymbol{W}^T_{(.)}\boldsymbol{Z}_{(.)}$ for regressor-based, Slotine and FAT-based approaches, respectively, never converge to zero. For this reason, these approaches cannot be used to identify the system coefficients.

In the proposed adaptive method (Fig. 2.7), if one can ignore values at higher derivatives of the original model $\boldsymbol{\phi}$, the error $\tilde{\boldsymbol{\phi}}$ converges to zero and also evident from Fig. 2.8. Considering a critically damped system, a_i can be easily evaluated using the pole-placement technique. $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ are taken as identity matrices. The results in this section only show one aspect for comparison, which is the approximation of the system dynamics. In the next chapter, we showed that the proposed controller is optimal, easy to tune and has a quantitative performance analysis.

2.6.2 Experimental Results

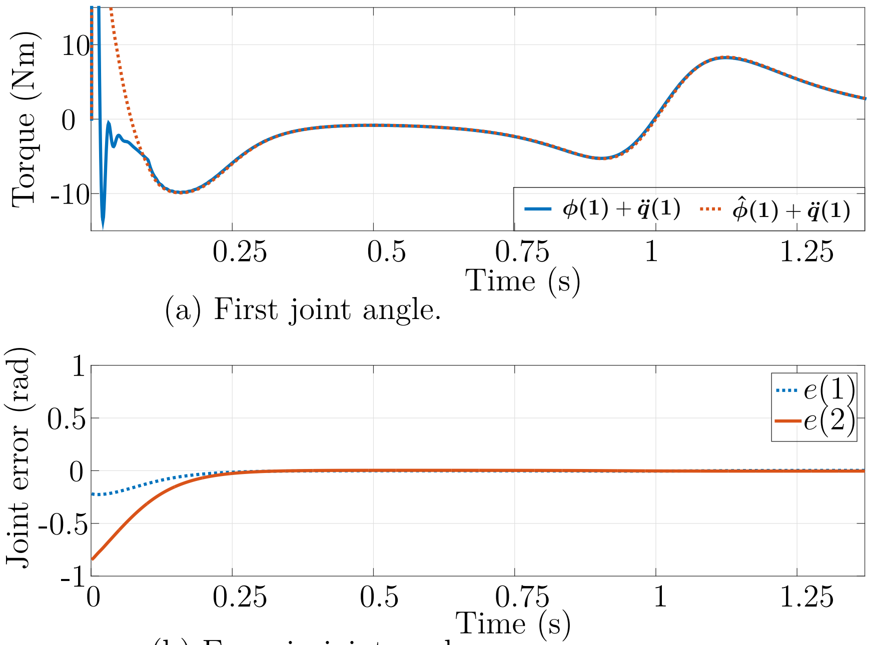
The experimental results for all of the controllers are shown in Fig. 2.9. $m = 1$ is considered for experiments to avoid high-frequency noise. A low-pass filter is also used to remove noise in the joint acceleration. Apart from the PD gains $\boldsymbol{K}_{\mathrm{d}}$ and $\boldsymbol{K}_{\mathrm{p}}$, the proposed technique requires only one variable a to adjust the adaptive controller. However, a large value of a gives higher torques, which introduces high-frequency oscillation. In experiments $a = 80$, thus ensuring that the performance of the PD control is not disturbed by the convergence



(a) First joint angle.

(b) Error in joint angles.

Figure 2.6: Results of FAT-based adaptive control. Approximation of system parameters by a combination of some orthogonal bases has high computational cost.



(a) First joint angle.

(b) Error in joint angles.

Figure 2.7: Results of proposed adaptive control. Taking a very low time constant for the update law will confirm the modeling error to approach zero. This technique requires only one parameter for the adaptation of system dynamics.

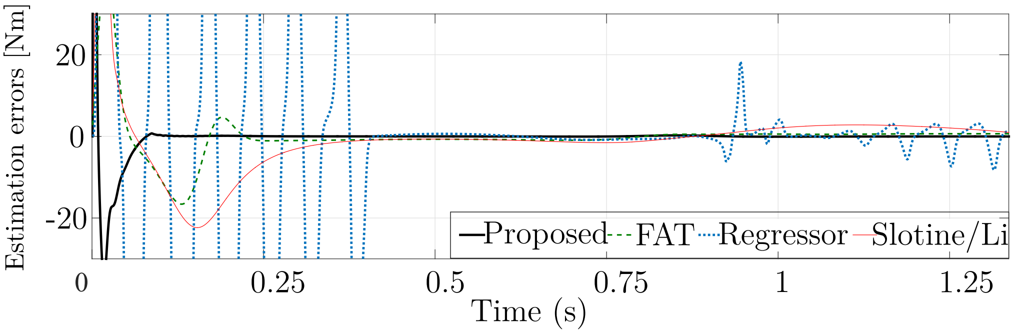


Figure 2.8: Error ($\tilde{\boldsymbol{\phi}}$) comparison of all the four controllers for the first joint angle of 2-DoF robot. As shown, the error for the FAT, regressor and Slotine/Li's controllers are not zero.

of system error. The values for the PD gain are

$$\boldsymbol{K}_{\mathrm{d}} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}, \quad \boldsymbol{K}_{\mathrm{p}} = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}$$

The poles for the PD gain are at -10, while a is selected eight times faster than the poles of the PD control. Because of the reason that the mismatch tends to zero in the proposed adaptive controller, the end-effector follows the trajectory satisfying equation (2.5) as clear from Fig. 2.9. Note that the proposed adaptive control technique also deals with disturbances because $\boldsymbol{\tau}_d$ is also considered as part of the unknown system dynamics.

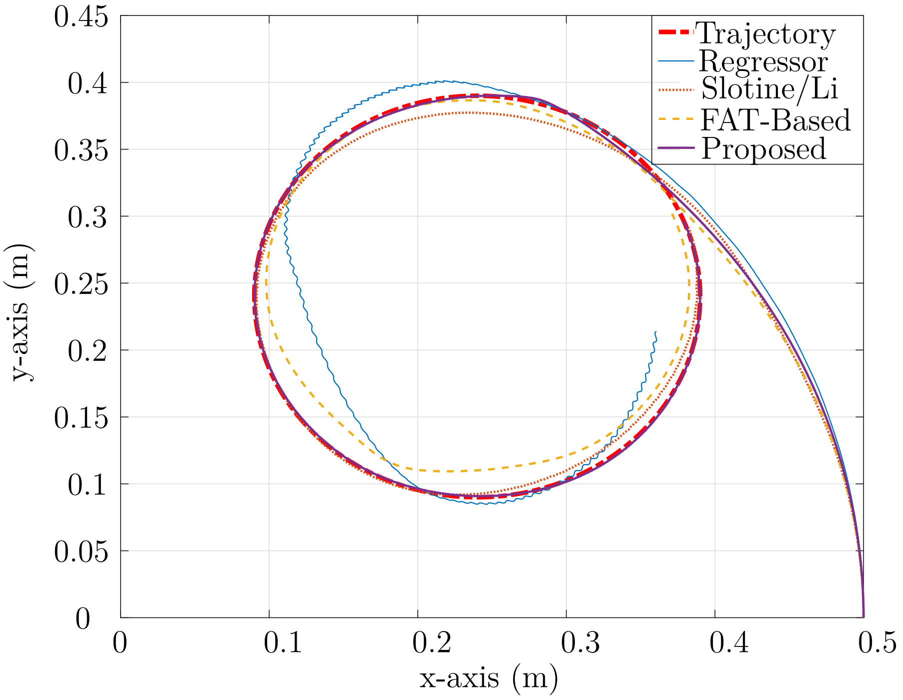


Figure 2.9: Experimental results: position of end effectors starting from $0.5\widehat{x} + 0\widehat{y}$ m.

2.7 System Identification

System parameter identification has long been a major interest in industrial applications. The identification of dynamical model improves the performance of a model-based control. It is also beneficial for a realistic simulation of a control method on a system. Computing the system dynamics using computer-aided design methods are not often accurate. Many researchers have contributed to the identification of robotic systems [3,97–105]. An overview of the offline and online identification methods is presented in [106]. In [107], a few set membership estimation techniques are implemented to identify the parameters of the system. A sequential procedure is proposed in [108], where the parameters are identified sequentially, in order of friction, gravity and rest of the parameters.

Remark 9. *The proposed adaptive controller is model-free and hence the system identification is not required for the controller. The identified values could be only useful for finding the initial value of* $\hat{\boldsymbol{\phi}}$*, which can improve the transient response.*

It has been shown in [109] that a simple model with a focus on the most influential parameters of the system gives better simulation results. The adaptive controllers usually use model-reference control with parametrized model and the parameters are referred to as the adaptive weights. Due to lack of persistence of excitation, these parameters often do not approach real parameters [110]. The structure of robot manipulators can be evaluated using the Euler-Lagrange equation and the uncertainties lie only in the constant coefficients of the system parameters. Using the proposed controller, $\boldsymbol{z} \to \boldsymbol{0}$, we eventually get

$$\ddot{\boldsymbol{e}} + \boldsymbol{K}_{\mathrm{d}}\dot{\boldsymbol{e}} + \boldsymbol{K}_{\mathrm{p}}\boldsymbol{e} = \boldsymbol{0}.$$

To find the unknown parameters, take the values of $\hat{\boldsymbol{\phi}}$ whenever the above equation is true at each time step (since $\boldsymbol{\phi} = \hat{\boldsymbol{\phi}}$ at that sample). Once $\boldsymbol{\phi}$ is known, the LS technique can be used to determine the system parameters. The essential requirement for identifying the system model using LS is by having a proper excitation trajectory. There are techniques that assure the identification is valid without the use of persistence of excitation. For that reason, the following criterion is required to be fulfilled [111]:

$$\int_{t_0}^{t_f} \boldsymbol{R}(t)\boldsymbol{R}(t)^T \mathrm{d}t > \gamma \boldsymbol{I}, \tag{2.36}$$

where $\boldsymbol{R}(t)$ is the regressor matrix, t_0 and t_f are the initial time and final time, respectively and γ is a positive number. Because of the increasing computational power, concurrent learning approaches gained more attention [112], where recorded data and current data are used to identify the system parameters online. To identify model parameters experimentally, it is necessary to have position, velocity and acceleration at each joint to get a rich regressor matrix. Once a large number of data points for $\hat{\boldsymbol{\phi}}$ is available, which in our experiments are nearly 5000 data points, LS can be applied to estimate the model parameters [9]:

$$\begin{aligned} \boldsymbol{\mathcal{Y}}\hat{\boldsymbol{p}} &= \overline{\boldsymbol{B}}, \\ \hat{\boldsymbol{p}} &= (\boldsymbol{\mathcal{Y}}^T\boldsymbol{\mathcal{Y}})^{-1}\boldsymbol{\mathcal{Y}}^T\overline{\boldsymbol{B}}, \end{aligned} \tag{2.37}$$

where, $\boldsymbol{\mathcal{Y}} = [\boldsymbol{Y}_1^T\ \boldsymbol{Y}_2^T\ ...\ \boldsymbol{Y}_N^T]^T$, $\overline{\boldsymbol{B}} = [\hat{\boldsymbol{\phi}}_1^T\ \hat{\boldsymbol{\phi}}_2^T\ ...\ \hat{\boldsymbol{\phi}}_N^T]^T$, and N is the total number of sampled data points. The columns of the matrix $\boldsymbol{\mathcal{Y}}$ should be linearly independent for LS to accurately approximate the parameters.

Table 2.1: Calculated, simulation and experimental estimation of parameters for 2-DoF manipulator.

Var.	**Equivalent**	**Cal.**	**Sim.**	**Exp.**
m_{11}	$\frac{1}{3}m_1l_1^2 + \frac{1}{3}m_2l_2^2 + m_2l_1^2$	0.442	0.442	0.590
$m_{11}^{'}$	$m_2l_1l_2$	0.028	0.027	0.027
m_{12}	$\frac{1}{3}m_2l_2^2$	0.009	0.008	0.006
$m_{12}^{'}$	$\frac{1}{2}m_2l_1l_2$	0.014	0.015	0.010
n_{11}	$\frac{1}{2}m_2l_1l_2$	0.014	0.014	0.216
n_{12}	$m_2l_1l_2$	0.030	0.028	0.041
f_{11}	$-$	0.001	0.004	0.001
m_{21}	$\frac{1}{3}m_2l_2^2$	0.009	0.009	0.005
$m_{21}^{'}$	$\frac{1}{2}m_2l_1l_2$	0.014	0.015	0.009
m_{22}	$\frac{1}{3}m_2l_2^2$	0.222	0.222	0.282
n_{21}	$\frac{1}{2}m_2l_1l_2$	0.014	0.013	0.282
f_{22}	$-$	0.001	0.006	0.001

Table 2.2: Parameter estimation for 3-DoF.

Var.	**Cal.**	**Est.**	**Var.**	**Cal.**	**Est.**	**Var.**	**Cal.**	**Est.**
m_{11}	0.7089	0.7003	$m_{21}^{''}$	0.0457	0.0414	n_{11}	0.1636	0.6585
$m_{11}^{'}$	0.0457	0.0375	$m_{21}^{'''}$	0.2470	0.2427	n_{12}	0.0837	0.0501
$m_{11}^{''}$	0.0457	0.0510	m_{22}	0.2493	0.2490	n_{13}	0.0228	0.0112
$m_{11}^{'''}$	0.4940	0.5012	$m_{22}^{'}$	0.0457	0.0437	n_{14}	0.0228	0.0244
m_{12}	0.2493	0.2444	m_{23}	0.0130	0.0126	n_{15}	0.0837	0.0384
$m_{12}^{'}$	0.0228	0.0203	$m_{23}^{'}$	0.0228	0.0225	n_{16}	0.0228	0.0226
$m_{12}^{''}$	0.0457	0.0503	n_{21}	0.2470	0.2478	n_{17}	0.0228	0.0233
$m_{12}^{'''}$	0.2470	0.2462	n_{22}	0.0228	0.0248	f_{11}	0.0010	0.0017
m_{13}	0.0130	0.0120	n_{23}	0.0228	0.0214	m_{21}	0.2493	0.2510
$m_{13}^{'}$	0.0228	0.0236	n_{24}	0.0228	0.0219	$m_{21}^{'}$	0.0228	0.0264
$m_{13}^{''}$	0.0228	0.0199	f_{22}	0.0010	0.0020	m_{31}	0.0130	0.0133
$m_{31}^{''}$	0.0228	0.0223	$m_{31}^{'}$	0.0228	0.0224	m_{33}	0.0130	0.0141
$m_{32}^{'}$	0.0228	0.0228	m_{32}	0.0130	0.129	n_{31}	0.0228	0.0232
f_{33}	0.0010	0.0050	n_{32}	0.0228	0.0228	-	-	-

Using the LS approximation with an adequate excitation trajectory, the unknown parameters of 2-DoF manipulator are estimated as shown in Table 2.1. The table consists of three sets of parameters: (1) calculated parameters using the geometry of the manipulator, (2) identified parameters using the simulation data with calculated parameters used as the system model and (3) identified parameters using the experimental data. The simulations for parameter estimations produce good results, nevertheless, because in experiments, the real model is not known, we can only assume that the experimentally identified parameters must be close to the calculated values. Moreover, the identification process can be improved by adding torque boundedness and unmodeled parameters in terms of disturbances. Both in simulations and experiments, $\boldsymbol{\tau}_d$ is ignored for the identification process.

Following the same procedure for the 3-DoF manipulator, the system parameters are shown in Table 2.2. The white box model of the 3-DoF manipulator is

$$\boldsymbol{M} = \begin{bmatrix} m_{11} + m'_{11}c_3 + m''_{11}c_{23} + m'''_{11}c_2 & m'''_{12} + m'_{12}c_{23} + m''_{12}c_3 + m_{12}c_2 & m_{13} + m'_{13}c_3 + m''_{13}c_{23} \\ m'''_{21} + m'_{21}c_{23} + m''_{21}c_3 + m_{21}c_2 & m_{22} + m'_{22}c_3 & m_{23} + m'_{23}c_3 \\ m_{31} + m'_{31}c_3 + m''_{31}c_{23} & m_{32} + m'_{32}c_3 & m_{33} \end{bmatrix}, \tag{2.38}$$

$$\boldsymbol{N} = \begin{bmatrix} n_{11}\left(\left(\dot{q}_1^2 - \left(\dot{q}_1 + \dot{q}_2\right)^2\right)s_2\right) - n_{12}\left(\left(\dot{q}_1 + \dot{q}_2\right)^2 s_2\right) + n_{13}\left(\left(\dot{q}_1 + \dot{q}_2\right)^2 s_3\right) - n_{14}\left(\left(\dot{q}_1 + \dot{q}_2 + \dot{q}_3\right)^2 s_3\right) + n_{15}\left(\dot{q}_1^2 s_2\right) - \\ n_{16}\left(\left(\dot{q}_1 + \dot{q}_2 + \dot{q}_3\right)^2 s_{23}\right) + n_{17}\left(\dot{q}_1^2 s_{23}\right) \\ n_{21}\left(\dot{q}_1^2 s_2\right) + n_{22}\left(\dot{q}_1^2 s_{23}\right) + n_{23}\left(\left(\dot{q}_1 + \dot{q}_2\right)^2 s_3\right) - n_{24}\left(\left(\dot{q}_1 + \dot{q}_2 + \dot{q}_3\right)^2 s_3\right) \\ n_{31}\left(\dot{q}_1^2 s_{23}\right) + n_{32}\left(\left(\dot{q}_1 + \dot{q}_2\right)^2 s_3\right) \end{bmatrix}, \tag{2.39}$$

$$\boldsymbol{N} = \begin{bmatrix} f_{11} & 0 & 0 \\ 0 & f_{22} & 0 \\ 0 & 0 & f_{33} \end{bmatrix}, \tag{2.40}$$

where c and s represent cosine and sine, respectively. The MATLAB code for the parameters identification of the 3-DoF manipulator is given in Appendix B4.

2.8 Summary

A well-known state-of-the-art control design technique known as computed-torque is briefly explained that uses feedback linearization. By using such a controller, a simple proportional-derivative (PD) controller can be utilized to achieve desired tracking performance. However, computed-torque is a model-based controller and its performance depend on the estimated system parameters.

The dynamics of a robot is given by the E-L equations. If uncertainties in the system dynamics appear only in the constant coefficients, e.g. masses or lengths of joints, we can reformulate the E-L equations into a state-dependent regressor matrix and an unknown constant coefficient vector. Many of the existing adaptive controllers are Lyapunov-based and take advantage of this fact, and Lyapunov's descent property is then used to find the unknown coefficient vector. Calculation of the regressor matrix, especially for a high degree-of-freedom manipulator, is a complicated task if there are many system parameters. The function approximation technique approximates the system dynamics by linear orthogonal basis functions to avoid the evaluation of the regressor matrix. This technique relies on intensive computation because of a large number of basis-functions required to replace the system dynamics. It also sacrifices the non-linear structure of the robot's dynamical equations by approximating it with linear basis-functions. The above mentioned control schemes focus only on the stability of the feedback system. These control methods lack optimality, quantitative performance, easy tuning, the inclusion of input constraints and implementation on high DoF manipulators.

The proposed adaptive control technique guarantees convergence of any mismatch in the system by adjusting a single controller parameter for a first order controller. Once the mismatch is eliminated, there is an accurate feedback linearization and a simple PD control will achieve the desired performance. Using the adaptive control law, the model parameters are approximated applying LS approximation. After the identification of system parameters, the adaptive control can be replaced by a simple PD control.

Compared to existing algorithms involving regressor-based approaches or FAT, the presented algorithm requires less computational cost. Furthermore, design parameters are few and more easily tuned in the proposed algorithm. Both simulation and experimental results show better and more precise tracking performance as compared to existing adaptive methods.

Open problem

This chapter explains a novel control method with the aim of removing both the state errors and system mismatch for better performance. Since the controller is designed on the basis of intuition and not mathematical proofs, a detailed stability and optimality proofs are not given. In the next chapter, we have shown that the proposed control method is $\mathcal{H}_\infty$ optimal and guarantees a predefined steady-state accuracy. Furthermore, a systematic procedure is also devised to evaluate the parameters of the controller. For fast removal of mismatch, a higher-order controller is required, however, that will bring high-frequency noise. A modification in the feedback system is formulated in Chapter 4 to apply high-order controller without deteriorating the performance because of the noise.

3

Stability, Optimality and Performance Analysis

In this chapter, the stability and robust optimality of the proposed adaptive controller is elaborated with mathematical validations. In addition, a performance analysis of the adaptive controller in terms of steady-state error is also explained using input-to-state stability (ISS). The controller assures that the joint errors at steady-state do not exceed a predefined maximum limit.

Like an inverse-dynamical control, the proposed controller has two feedback loops as shown in Fig. 3.1: the inner loop is adaptive and estimates the system dynamics for cancellation of the nonlinearities and the outer loop is a proportional-derivative (PD) control. A simple linear differential equation is formulated to evaluate the adaptive part. An advantage of the proposed controller is that it neither requires an estimated system model nor a regressor matrix and acts like a model-free adaptive control. Moreover, the controller only uses the joint velocities, joint angles and integrals of joint angles to calculate the control input.

The control design considers two aspects: first, the $\mathcal{H}_\infty$ approach and ISS are used to ensure the stability, robustness and a prescribed quantitative performance. Second, the adaptation of the estimated system dynamics towards the real dynamics is ensured without any initial knowledge about the actual system. The estimation of system dynamics is essential in the control because it is then used to linearize the overall feedback system, hence winding up as an adaptive computed-torque method. The robotic system is represented as an $\mathcal{L}_2$-gain problem and using the Hamilton-Jacobi-Isaacs (HJI) equations, we proved that the controller is robust optimal with predefined steady-state. An inverse-optimal control method is used to find the matrices for the HJI equations, i.e., "given a controller, evaluation of the $\mathcal{L}_2$-gain that satisfies the robustness". Thus, we get a simple controller with only a few controller parameters. The effect of external disturbances is significantly reduced using the ISS. It is also the tool used in this chapter to derive a general form of the adaptive control. Another advantage of using ISS is that the parameters of the controller can be determined depending on desired performance specifications.

In summary, the state-of-the-art robust/adaptive control schemes have one or more of the following limitations: no optimality, high computational cost, dependence on system parameters, no quantitative performance analysis and challenging control's tuning. This chapter addresses the issues mentioned along with stability, robustness and straightforward implementation. However, the primary focus is the performance and relatively simple implementation of the controller, especially for high degree-of-freedom (DoF) manipulators.

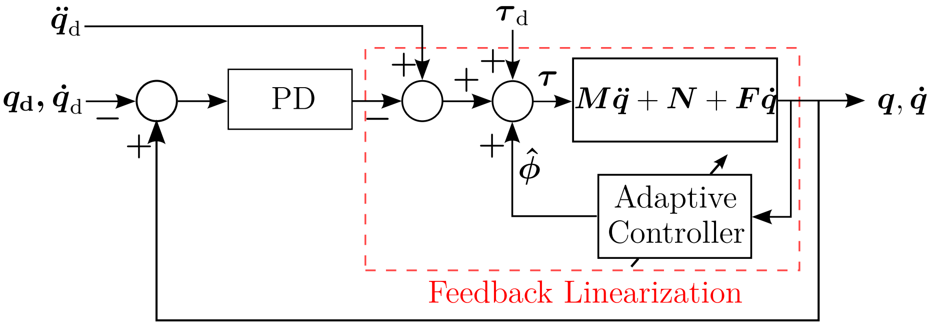

$\ddot{\boldsymbol{q}}_{\mathrm{d}}$
$\boldsymbol{\tau}_{\mathrm{d}}$
$\boldsymbol{q}_{\mathrm{d}}, \dot{\boldsymbol{q}}_{\mathrm{d}}$
PD
$\boldsymbol{\tau}$
$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{N} + \boldsymbol{F}\dot{\boldsymbol{q}}$
$\boldsymbol{q}, \dot{\boldsymbol{q}}$
$\hat{\boldsymbol{\phi}}$
Adaptive Controller
Feedback Linearization
+
−

Figure 3.1: Robust-adaptive control.

The performance of the proposed controller is validated by experiments using a 3-DoF and a 7-DoF robotic manipulator. Further improvements relating to the transient response and input saturation are described qualitatively in the next chapter.

The remainder of the chapter is organized as follows: the main contribution of the dissertation, which is a novel robust-adaptive control, is summarized in Section 3.1. The robustness shown by the $\mathcal{H}_\infty$ inverse-optimal control is discussed in Section 3.2. Section 3.3 investigates ISS of the proposed controller. Performance and evaluation of controller parameters is presented in Section 3.4. In Section 3.5, simulation and experimental results for a 3-DoF and a 7-DoF robotic arm are shown. In the end, conclusions are drawn in Section 3.6.

3.1 Proposed Controller

The system dynamics used for the proposed adaptive controller is given in (2.1) with the assumptions introduced in Section 2.1.

Second-Order Adaptive Controller

The basic idea of the proposed controller is that a single vector $\boldsymbol{\phi}$ is used to represent the system dynamics. The major concern is evaluation of the estimated vector $\hat{\boldsymbol{\phi}}$, which is later used to calculate the input torque. The estimated system dynamics $\hat{\boldsymbol{\phi}}$ starts from zero and converges to the real system dynamics, which is used for feedback linearization. Hence the controller can be seen as an adaptive computed-torque method.

Let $\boldsymbol{\phi}$ summarizes the system dynamics

$$\boldsymbol{\phi} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{N}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{F}\dot{\boldsymbol{q}} - \ddot{\boldsymbol{q}} - \boldsymbol{\tau}_{\mathrm{d}}, \tag{3.1}$$

where $\boldsymbol{\tau}_{\mathrm{d}}$ represents external disturbances and unmodeled dynamics [3, 13].

If the input torque is chosen as follows,

$$\boldsymbol{\tau}_{\mathrm{in}} = \hat{\boldsymbol{\phi}} + \ddot{\boldsymbol{q}}_{\mathrm{d}} - \boldsymbol{K}_{\mathrm{d}}\dot{\boldsymbol{e}} - \boldsymbol{K}_{\mathrm{p}}\boldsymbol{e} \tag{3.2}$$

with the assumption that the desired trajectory is twice differentiable, then by using (2.1), (3.1) and (3.2), we obtain the error dynamics

$$\ddot{\boldsymbol{e}} + \boldsymbol{K}_{\mathrm{d}}\dot{\boldsymbol{e}} + \boldsymbol{K}_{\mathrm{p}}\boldsymbol{e} = \hat{\boldsymbol{\phi}} - \boldsymbol{\phi}. \tag{3.3}$$

To find the input torque $\boldsymbol{\tau}_{\text{in}}$, we need to evaluate $\hat{\boldsymbol{\phi}}$, which will be shown as a function of only the states of the system (Section 3.2). It is obvious here that if $\hat{\boldsymbol{\phi}} = \boldsymbol{\phi}$, we end up with (2.5), which is an ideal response. However, $\boldsymbol{\phi}$, which includes all the system dynamics along with disturbances in a single column (vector), is not known. The following Theorem provides the control law for the estimation of $\hat{\boldsymbol{\phi}}$.

Theorem 2 (Adaptation)**.** *Let $\tilde{\boldsymbol{\phi}} = \hat{\boldsymbol{\phi}} - \boldsymbol{\phi}$ and*

$$\frac{\mathrm{d}^m \hat{\boldsymbol{\phi}}}{\mathrm{d}t^m} = -\sum_{i=0}^{m-1} a_i \frac{\mathrm{d}^i \tilde{\boldsymbol{\phi}}}{\mathrm{d}t^i}, \tag{3.4}$$

then the control law (3.2) *leads to stability of the joint error $\boldsymbol{e} \approx \mathbf{0}$ and estimation error $\tilde{\boldsymbol{\phi}} \approx \mathbf{0}$ for the system* (2.1).

Proof. The stability of the proposed adaptive law is proved in Section 3.4. □

The update law (3.4) does not require a regression matrix and can be implemented with low computational cost. The input torque $\boldsymbol{\tau}_{\text{in}}$ is a function of only the state of the system (2.1), which is explained in Lemma 1 and also shown in (2.30) and (2.31). The constant coefficients a_i decide the rate of convergence of $\hat{\boldsymbol{\phi}}$ towards $\boldsymbol{\phi}$, explained in Section 3.4.

Fig. 3.1 shows a block diagram representation of the proposed controller. The adaptive control is applied in the inner loop that linearizes the overall feedback system. The reason for defining the controller as adaptive computed-torque is that, after $\hat{\boldsymbol{\phi}}$ converges to $\boldsymbol{\phi}$, the controller acts as a feedback linearization method or intelligent PD controller [83].

Remark 10 (High-frequency problem)**.** *Adaptive control, in general, suffers from high-frequency noise during the transient response. Fortunately, the motor drives and the power supplies act like low pass filters, and suppress the effect of noise.*

Remark 11 (Noise)**.** *The only purpose of including $\ddot{\boldsymbol{q}}$ in* (3.1) *is to avoid the estimation of joint-acceleration when evaluating input torques.*

Remark 12 (Approximation of $\boldsymbol{\phi}$)**.** *For a better estimation of $\boldsymbol{\phi}$, it is desired to select a large m. However, the system will require large torques during the transient, which will be shown in the results section (3.5). These high torques result in peaking which can affect the performance, see [113]. For that reason, we have determined that $m = 2$ along with proper controller gains is sufficient to achieve the desired performance.*

Remark 13 (Tuning Parameters)**.** *The only tuning parameters are $a_i, \boldsymbol{K}_{\text{p}}$ and $\boldsymbol{K}_{\text{d}}$ and it will be shown in Section 3.4 on how to select these controller parameters while satisfying the desired performance.*

Remark 14. *In our previous work [3], a term $\boldsymbol{f}(\boldsymbol{e}, \dot{\boldsymbol{e}})$ was included on the right-hand side of* (3.4) *that was necessary for the stability proof. Fortunately, here a modified proof is presented that does not need this extra term.*

Remark 15. *The proposed adaptive controller uses the same concept of intelligent PID (iPID) controller proposed by Fliess et al. [81–83]. The controller parameters do not require a re-calibration if there are variations in the system.*
The iPID controller estimates the system dynamics $\boldsymbol{\phi}$ with piece-wise constants which are then used in the identification of system dynamics. In the proposed controller, the system is approximated by constants for first-order controller, ramps for 2nd-order controller and so on. Moreover, we have shown the optimality, and the quantitative performance analysis of the proposed controller.

3.2 Robustness Analysis

The proposed adaptive control method eventually constitutes an $\mathcal{H}_\infty$ robust-optimal control. In this section, we will show that there exist matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ such that the system (2.1) along with the control law (3.4) satisfies the $\mathcal{L}_2$-gain condition (A.3) in Appendix A1. From the $\mathcal{H}_\infty$ perspective, the main idea is that we consider the system shown in Fig. 3.1 as a linear system with all the nonlinearities, disturbances and unknown parameters as external disturbances as shown in Fig. 3.2. Some important mathematical techniques, used in this chapter, are discussed in Appendix A.

Robustness Proof

An inverse-optimal control method is used to prove the robustness of the proposed controller. So, the objective of this section is: *given the control law* (3.4)*, find the matrices* $\boldsymbol{S}$ *and* $\boldsymbol{Q}$ *such that the game-algebraic Riccati equation (GARE)* (A.8) *is satisfied.*

The state-space form of the closed-loop equation for the Euler-Lagrange representation (3.3) can be written in the form of (A.7) as

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\hat{\boldsymbol{\phi}} - \boldsymbol{B}\boldsymbol{\phi}. \tag{3.5}$$

Based on the value of m in (3.4), the matrices are

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{0}_{(m+1)n\times n} & \boldsymbol{I}_{(m+1)n\times(m+1)n} \\ \boldsymbol{0}_{n\times n} & \boldsymbol{L}_{n\times(m+1)n} \end{bmatrix} \in \mathbb{R}^{(m+2)n\times(m+2)n},$$

$$\boldsymbol{L} = \begin{bmatrix} \boldsymbol{0}_{n\times(m-1)n} & -\boldsymbol{K}_\mathrm{p} - \boldsymbol{K}_\mathrm{d} \end{bmatrix} \in \mathbb{R}^{n\times(m+1)n},$$

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{0}_{n\times(m+1)n} & \boldsymbol{I}_{n\times n} \end{bmatrix}^T \in \mathbb{R}^{(m+2)n\times n},$$

$$\boldsymbol{x} = [\int_m \boldsymbol{e}^T \mathrm{d}t^m \quad \int_{m-1} \boldsymbol{e}^T \mathrm{d}t^{m-1} \quad . \;\; . \;\; . \quad \boldsymbol{e}^T \quad \dot{\boldsymbol{e}}^T]^T,$$

where $\int_m$ represents the m^{th} integral with respect to time. Since $\hat{\boldsymbol{\phi}}$ determines the input torque according to (3.2), we can consider $-\boldsymbol{\phi}$ as disturbance '$\boldsymbol{d}$'. In the remainder of this chapter, we will use the conventional input symbol '$\boldsymbol{u}$' and disturbance '$\boldsymbol{d}$' variables interchangeably for '$\hat{\boldsymbol{\phi}}$' and '$-\boldsymbol{\phi}$', respectively.

The following Lemma describes a different representation of (3.4) that will make it possible to apply the HJI equation on our system.

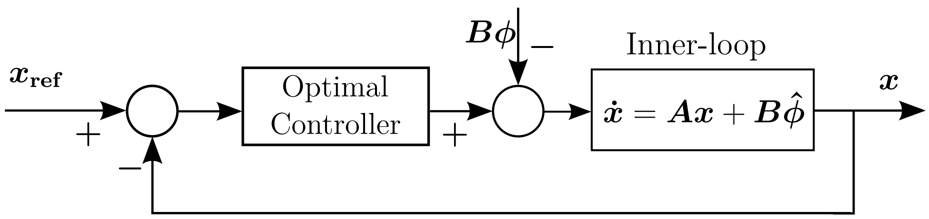

$\boldsymbol{B\phi}$
$-$
Inner-loop
$\boldsymbol{x}_{\mathbf{ref}}$
$+$
$-$
Optimal Controller
$+$
$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\hat{\boldsymbol{\phi}}$
$\boldsymbol{x}$

Figure 3.2: Adaptive control from the perspective of $\mathcal{H}_\infty$ optimal control.

Lemma 1 (Alternative form for (3.4))**.** *Let*

$$\hat{\boldsymbol{\phi}} = -\sum_{i=1}^{m+2}\Big(a_{i-1}\boldsymbol{K}_\mathrm{p} + a_{i-2}\boldsymbol{K}_\mathrm{d} + a_{i-3}\Big)\boldsymbol{x}[i] \triangleq -\boldsymbol{K}\boldsymbol{x}, \tag{3.6}$$

where $\boldsymbol{x}[i]$ *is the* i^{th} *element of the state vector, with the condition*

$$\textit{E1: } a_i = 0 \quad \forall \quad i < 0, \quad i > m - 1.$$

Then (3.6) *is equivalent to* (3.4)*.*

Proof. Equation (3.6) can be rearranged by taking advantage of condition E1 to get the following form:

$$\hat{\boldsymbol{\phi}} = -\sum_{i=1}^{m+2}\Big(a_{i-1}\boldsymbol{K}_\mathrm{p}\boldsymbol{x}[i] + a_{i-1}\boldsymbol{K}_\mathrm{d}\boldsymbol{x}[i+1] + a_{i-1}\boldsymbol{x}[i+2]\Big),$$

$$\hat{\boldsymbol{\phi}} = -\sum_{i=1}^{m} a_{i-1}\Big(\boldsymbol{K}_\mathrm{p}\boldsymbol{x}[i] + \boldsymbol{K}_\mathrm{d}\boldsymbol{x}[i+1] + \boldsymbol{x}[i+2]\Big),$$

Integrating (3.3), we get

$$\int_{m-i+1} \tilde{\boldsymbol{\phi}}\mathrm{d}t^{m-i+1} = \boldsymbol{K}_\mathrm{p}\boldsymbol{x}[i] + \boldsymbol{K}_\mathrm{d}\boldsymbol{x}[i+1] + \boldsymbol{x}[i+2]. \tag{3.7}$$

Taking the m^{th} derivative, we get the control law (3.4). □

Remark 16. *For second-order (*$m = 2$*), the controller* (3.6) *can be written as*

$$\frac{\mathrm{d}^2\hat{\boldsymbol{\phi}}}{\mathrm{d}t^2} = -a_1\frac{\mathrm{d}}{\mathrm{d}t}(\ddot{\boldsymbol{e}} + \boldsymbol{K}_\mathrm{d}\dot{\boldsymbol{e}} + \boldsymbol{K}_\mathrm{p}\boldsymbol{e}) - a_0(\ddot{\boldsymbol{e}} + \boldsymbol{K}_\mathrm{d}\dot{\boldsymbol{e}} + \boldsymbol{K}_\mathrm{p}\boldsymbol{e}). \tag{3.9}$$

It is evident from the example $m = 1$ in (2.30) and $m = 2$ in (2.31) that $\hat{\boldsymbol{\phi}}$ can be written as a state feedback $\hat{\boldsymbol{\phi}} = -\boldsymbol{K}\boldsymbol{x}$ with appropriate choice of $\boldsymbol{K}$. The general case is treated in Lemma 1. This will allow us a comparison of (A.9) and (3.6). The next Theorem gives the optimality of the proposed adaptive control method for the appropriate choice of $\boldsymbol{R}$ and $\boldsymbol{Q}$. Here, only the matrices $\boldsymbol{S}$ and $\boldsymbol{Q}$ are evaluated using inverse-optimal control.

Remark 17. *The matrix $\boldsymbol{R}$ is chosen as*

$$\boldsymbol{R}^{-1} = \alpha\tilde{\boldsymbol{K}} + \boldsymbol{I}/\gamma^2 \triangleq a_{m-1}\boldsymbol{I}, \quad \alpha > \boldsymbol{0}, \tilde{\boldsymbol{K}} > \boldsymbol{0}, \tag{3.10}$$

which is proved by ISS (Section 3.4). Note that $\alpha, \tilde{\boldsymbol{K}}$ and γ are intermediate terms and they are only used to prove the optimality. In the above equation, only a_{m-1} is required for the tuning of the proposed controller.

For proving the robust optimality, $m = 2$ is assumed. The reason is that there is no generalized procedure to proof the robustness for all m. But fortunately, the proof can be extended for any value of m. Another intuition for $m = 2$ is explained in Section 3.5.3.

Theorem 3 (Robust-optimal controller $m = 2$)**.** *For a system* (3.5)*, the proposed adaptive controller* (2.31) *or* (3.9) *gives an optimal solution that satisfies the GARE* (A.8) *under the following conditions:*

- *A1: a_0, a_1, $\boldsymbol{K}_\mathrm{p}$ and $\boldsymbol{K}_\mathrm{d} > \boldsymbol{0}$*
- *A2: $\boldsymbol{K}_\mathrm{d}^2 > 2\boldsymbol{K}_\mathrm{p}$*
- *A3: $a_1\tilde{\boldsymbol{K}} > 2a_0\boldsymbol{I}$*

Proof. If the above controller (3.9) is optimal, that means it satisfies the $\mathcal{L}_2$-gain condition of (A.3), there exist $\boldsymbol{Q}$, $\boldsymbol{R}$ and $\boldsymbol{S}$ symmetric, positive definite satisfying (A.8).

To find the values of $\boldsymbol{Q}$ and $\boldsymbol{S}$, we follow the steps below:

- assume $\boldsymbol{Q}$ to be a diagonal matrix
- find the last column/row of matrix $\boldsymbol{S}$ using $-\boldsymbol{K}\boldsymbol{x} = -\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{S}\boldsymbol{x}$
- find the remaining entries of $\boldsymbol{S}$ and diagonal elements of $\boldsymbol{Q}$ using (A.8), see (3.8)

From the $\boldsymbol{Q}$ matrix (3.8), required to be positive definite, we can deduce conditions A1-A3. □

There exists more than one solution for $\boldsymbol{Q}$ and $\boldsymbol{S}$ that fulfills the positive definiteness criterion but using the form (3.6) and (3.10), after some mathematical manipulation, we

$$\boldsymbol{Q} = \begin{bmatrix} \tilde{\boldsymbol{K}}b_0^2\boldsymbol{K}_\mathrm{p}^2 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{K}}b_0^2(\boldsymbol{K}_\mathrm{d}^2-2\boldsymbol{K}_\mathrm{p})+\boldsymbol{K}_\mathrm{p}^2(\tilde{\boldsymbol{K}}-2b_0\boldsymbol{I}) & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \tilde{\boldsymbol{K}}b_0^2+(\tilde{\boldsymbol{K}}-2b_0\boldsymbol{I})(\boldsymbol{K}_\mathrm{d}^2-2\boldsymbol{K}_\mathrm{p}) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \tilde{\boldsymbol{K}}-2b_0\boldsymbol{I} \end{bmatrix}, \; b_0 = \frac{a_0}{a_1}, \; \boldsymbol{S} =$$

$$\begin{bmatrix} b_0\tilde{\boldsymbol{K}}\boldsymbol{K}_\mathrm{p}^2+b_0^2\tilde{\boldsymbol{K}}\boldsymbol{K}_\mathrm{p}\boldsymbol{K}_\mathrm{d} & b_0\boldsymbol{K}_\mathrm{p}^2+b_0^2\tilde{\boldsymbol{K}}\boldsymbol{K}_\mathrm{p}+b_0\tilde{\boldsymbol{K}}\boldsymbol{K}_\mathrm{p}\boldsymbol{K}_\mathrm{d} & b_0\tilde{\boldsymbol{K}}\boldsymbol{K}_\mathrm{p}+b_0\boldsymbol{K}_\mathrm{p}\boldsymbol{K}_\mathrm{d} & b_0\boldsymbol{K}_\mathrm{p} \\ b_0\boldsymbol{K}_\mathrm{p}^2+b_0^2\tilde{\boldsymbol{K}}\boldsymbol{K}_\mathrm{p}+b_0\tilde{\boldsymbol{K}}\boldsymbol{K}_\mathrm{p}\boldsymbol{K}_\mathrm{d} & b_0^2\tilde{\boldsymbol{K}}\boldsymbol{K}_\mathrm{d}+b_0\tilde{\boldsymbol{K}}\boldsymbol{K}_\mathrm{d}^2+\tilde{\boldsymbol{K}}\boldsymbol{K}_\mathrm{p}\boldsymbol{K}_\mathrm{d}+\boldsymbol{K}_\mathrm{p}^2 & b_0(\boldsymbol{K}_\mathrm{d}^2+\tilde{\boldsymbol{K}}\boldsymbol{K}_\mathrm{d}-\boldsymbol{K}_\mathrm{p})+\tilde{\boldsymbol{K}}\boldsymbol{K}_\mathrm{p}+\boldsymbol{K}_\mathrm{p}\boldsymbol{K}_\mathrm{d} & \boldsymbol{K}_\mathrm{p}+b_0\boldsymbol{K}_\mathrm{d} \\ b_0\tilde{\boldsymbol{K}}\boldsymbol{K}_\mathrm{p}+b_0\boldsymbol{K}_\mathrm{p}\boldsymbol{K}_\mathrm{d} & b_0\boldsymbol{K}_\mathrm{d}^2+\tilde{\boldsymbol{K}}\boldsymbol{K}_\mathrm{p}+b_0\tilde{\boldsymbol{K}}\boldsymbol{K}_\mathrm{d}+\boldsymbol{K}_\mathrm{p}\boldsymbol{K}_\mathrm{d}-b_0\boldsymbol{K}_\mathrm{p} & \boldsymbol{K}_\mathrm{d}^2+b_0\tilde{\boldsymbol{K}}+\tilde{\boldsymbol{K}}\boldsymbol{K}_\mathrm{d} & \boldsymbol{K}_\mathrm{d}+b_0\boldsymbol{I} \\ b_0\boldsymbol{K}_\mathrm{p} & \boldsymbol{K}_\mathrm{p}+b_0\boldsymbol{K}_\mathrm{d} & \boldsymbol{K}_\mathrm{d}+b_0\boldsymbol{I} & \boldsymbol{I} \end{bmatrix} \tag{3.8}$$

can evaluate the matrices (3.8). To facilitate the analysis, we assume $\alpha = 1$ as explained in Section 3.3.1.

Since the control law satisfies the Hamilton-Jacobi-Bellman equation, we can conclude that the $\mathcal{L}_2$-gain is always less than γ. From the definition of the $\mathcal{L}_2$-gain problem, it is also clear that the system remains stable for $\boldsymbol{\phi} \in \mathcal{L}_2[0,\infty)$ [114]. In the results section, it will be shown that condition A2 is a sufficient but not necessary for stability. For $m = 2$, $\boldsymbol{K}$ can be written as

$$\boldsymbol{K} = a_1[b_0\boldsymbol{K}_{\mathrm{p}} \quad \boldsymbol{K}_{\mathrm{p}} + b_0\boldsymbol{K}_{\mathrm{d}} \quad \boldsymbol{K}_{\mathrm{d}} + b_0\boldsymbol{I} \quad \boldsymbol{I}], \tag{3.11}$$

where $b_0 = a_0/a_1$. The reason for writing the a_1 out of $\boldsymbol{B}^T\boldsymbol{S}\boldsymbol{x}$ is that $\boldsymbol{R}^{-1} = a_1\boldsymbol{I}$, see Section 3.3. Disturbance attenuation depends on the value of γ that is included in the $\boldsymbol{R}^{-1}$ matrix.

Remark 18 (Special case)**.** *For a first order approximation, that is $m = 1$, the robust-adaptive controller proposed in this chapter is similar to the $\mathcal{H}_\infty$ inverse-optimal Control [5, 10]. The $\boldsymbol{Q}$ matrix in this case is*

$$\boldsymbol{Q} = \begin{bmatrix} \tilde{\boldsymbol{K}}\boldsymbol{K}_{\mathrm{p}}^2 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{K}}(\boldsymbol{K}_{\mathrm{d}}^2 - 2\boldsymbol{K}_{\mathrm{p}}) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \tilde{\boldsymbol{K}} \end{bmatrix}. \tag{3.12}$$

The conditions for this $\boldsymbol{Q}$ matrix are

- B1: a_0, $\boldsymbol{K}_{\mathrm{p}}$ and $\boldsymbol{K}_{\mathrm{d}} > 0$
- B2: $\boldsymbol{K}_{\mathrm{d}}^2 > 2\boldsymbol{K}_{\mathrm{p}}$

If we let $a_0\boldsymbol{I} = \boldsymbol{R}^{-1}$ where $\boldsymbol{R}^{-1}$ is given by equation (3.10), then the control law is represented as (2.30), which is equivalent to the controller proposed by Chung *et al.* [5, 115, 116]. Of course, $\hat{\boldsymbol{\phi}} \neq \boldsymbol{\tau}_{\mathrm{in}}$, so the controllers are not exactly equal. Because of that, the performance analysis is different, and we can show that the mismatch approaches zero. Similar to that $\mathcal{H}_\infty$ controller, the proposed adaptive control method can be applied on a robot manipulator without any knowledge about the system dynamics.

3.3 Stability

So far, we have introduced a novel adaptive control method and showed its robustness using the $\mathcal{H}_\infty$ and inverse-optimal approach. In this section, we use ISS to derive the proposed controller (3.4) along with stability. In addition, the matrix $\boldsymbol{R}$ (3.10), which we have used in Theorem 3 is also derived using the theory of two-player zero-sum optimal control.

3.3.1 Derivation of Proposed Controller

To obtain the control law, the basic idea here is to consider $\boldsymbol{u} = \hat{\boldsymbol{\phi}}$ and use equation (A.15) as explained in the next Theorem.

Theorem 4 (Derivation)**.** *For the control law*

$$\hat{\boldsymbol{\phi}} = -\Big(\alpha\tilde{\boldsymbol{K}} + \frac{\rho^{-1}(|\boldsymbol{x}|)}{|\boldsymbol{x}_\triangle|}\Big)\boldsymbol{x}_\triangle, \quad \tilde{\boldsymbol{K}} > \boldsymbol{0}, \tag{3.13}$$

the system (3.5) *is ISS, where* $\rho \in \mathcal{K}_\infty$, $\alpha > 1/2$ *and* $\boldsymbol{x}_\triangle$ *is*

$$b_0\boldsymbol{K}_\mathrm{p}\boldsymbol{x}[1] + (b_0\boldsymbol{K}_\mathrm{d} + \boldsymbol{K}_\mathrm{p})\boldsymbol{x}[2] + (b_0\boldsymbol{I} + \boldsymbol{K}_\mathrm{d})\boldsymbol{x}[3] + \boldsymbol{x}[4],$$

which is equal to $\boldsymbol{B}^T\boldsymbol{S}\boldsymbol{x}$. *Also there exists an ISS-Lyapunov function*

$$V(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{S}\boldsymbol{x} \tag{3.14}$$

with $\boldsymbol{S}$ *from* (3.8) *and if conditions A1-A3 are satisfied.*

Proof. For the stability of (3.5), the derivative of the Lyapunov function has to satisfy the following condition:

$$\dot{V}(\boldsymbol{x}) = \nabla V\boldsymbol{A}\boldsymbol{x} + \nabla V\boldsymbol{B}\hat{\boldsymbol{\phi}} - \nabla V\boldsymbol{B}\boldsymbol{\phi} \leq 0. \tag{3.15}$$

$\nabla V\boldsymbol{A}\boldsymbol{x}$ from (3.15) can be rewritten as

$$\begin{aligned}
\nabla V\boldsymbol{A}\boldsymbol{x} &= \frac{1}{2}\boldsymbol{x}^T(\boldsymbol{A}^T\boldsymbol{S} + \boldsymbol{S}\boldsymbol{A})\boldsymbol{x},\\
&= \frac{1}{2}\Big[\tilde{\boldsymbol{K}}\boldsymbol{x}_\triangle^T\boldsymbol{x}_\triangle - \boldsymbol{x}^T[4](\tilde{\boldsymbol{K}} - 2b_0\boldsymbol{I})\boldsymbol{x}[4]\\
&\quad - \boldsymbol{x}^T[2]\Big(\tilde{\boldsymbol{K}}b_0^2(\boldsymbol{K}_\mathrm{d}^2 - 2\boldsymbol{K}_\mathrm{p}) + \boldsymbol{K}_\mathrm{p}^2(\tilde{\boldsymbol{K}} - 2b_0\boldsymbol{I})\Big)\boldsymbol{x}[2]\\
&\quad - \boldsymbol{x}^T[3]\Big((\tilde{\boldsymbol{K}} - 2b_0\boldsymbol{I})(\boldsymbol{K}_\mathrm{d}^2 - 2\boldsymbol{K}_\mathrm{p}) + \tilde{\boldsymbol{K}}b_0^2\Big)\boldsymbol{x}[3]\Big].
\end{aligned} \tag{3.16}$$

and

$$\nabla V_x\boldsymbol{B} = \boldsymbol{x}^T\boldsymbol{S}\boldsymbol{B} \triangleq \boldsymbol{x}_\triangle^T, \tag{3.17}$$

$\dot{V} < 0$ is thus satisfied if

$$\begin{aligned}
&\frac{1}{2}\tilde{\boldsymbol{K}}\boldsymbol{x}_\triangle^T\boldsymbol{x}_\triangle + \boldsymbol{x}_\triangle\hat{\boldsymbol{\phi}} - \boldsymbol{x}_\triangle\boldsymbol{\phi} \leq\\
&\qquad \frac{1}{2}\Big[\boldsymbol{x}^T[2]\Big(\tilde{\boldsymbol{K}}b_0^2(\boldsymbol{K}_\mathrm{d}^2 - 2\boldsymbol{K}_\mathrm{p}) + \boldsymbol{K}_\mathrm{p}^2(\tilde{\boldsymbol{K}} - 2b_0\boldsymbol{I})\Big)\boldsymbol{x}[2]\\
&\qquad + \boldsymbol{x}^T[3]\Big((\tilde{\boldsymbol{K}} - 2b_0\boldsymbol{I})(\boldsymbol{K}_\mathrm{d}^2 - 2\boldsymbol{K}_\mathrm{p}) + \tilde{\boldsymbol{K}}b_0^2\Big)\boldsymbol{x}[3]\\
&\qquad + \boldsymbol{x}^T[4](\tilde{\boldsymbol{K}} - 2b_0\boldsymbol{I})\boldsymbol{x}[4]\Big].
\end{aligned} \tag{3.18}$$

From the above equation, if conditions A1-A3 from Theorem 3 are satisfied, then the right-hand side of (3.18) is positive. To prove that the Lyapunov function (3.15) is an ISS-Lyapunov function, the left-hand side of (3.18) must be at least negative semi-definite. Taking advantage of (A.15), we get

$$\frac{\tilde{\boldsymbol{K}}}{2}\boldsymbol{x}_\triangle^T\boldsymbol{x}_\triangle + \boldsymbol{x}_\triangle\hat{\boldsymbol{\phi}} - \boldsymbol{x}_\triangle\boldsymbol{\phi} \leq \frac{\tilde{\boldsymbol{K}}}{2}\boldsymbol{x}_\triangle^T\boldsymbol{x}_\triangle + \boldsymbol{x}_\triangle\hat{\boldsymbol{\phi}} + |\boldsymbol{x}_\triangle|\rho^{-1}(|\boldsymbol{x}|). \tag{3.19}$$

To make sure that the above equation is negative definite or semi-definite, $\hat{\boldsymbol{\phi}}$ is chosen as

$$\hat{\boldsymbol{\phi}} = -\Big(\alpha\tilde{\boldsymbol{K}} + \frac{\rho^{-1}(|\boldsymbol{x}|)}{|\boldsymbol{x}_\triangle|}\boldsymbol{I}\Big)\boldsymbol{x}_\triangle. \tag{3.20}$$

□

Remark 19 (Special case [5]). *For $m = 1$, we get a similar solution as suggested by Chung et al. [5].*

3.3.2 Evaluation of the Matrix $\boldsymbol{R}$

It is now obvious, how $\boldsymbol{R}$ has to be chosen. Comparing (A.9) and (3.13), we get

$$\boldsymbol{R} = (\alpha\tilde{\boldsymbol{K}} + \frac{\rho^{-1}(|\boldsymbol{x}|)}{|\boldsymbol{x}_\triangle|}\boldsymbol{I})^{-1}. \tag{3.21}$$

To show the $\mathcal{H}_\infty$ optimality for the proposed controller, we choose the following performance index (PI) [114]:

$$J = \int_0^t \boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u} - \gamma^2\boldsymbol{d}^T\boldsymbol{d}\ \mathrm{d}\tau \tag{3.22}$$

To proof the optimality of the robust-adaptive controller, we use inverse-optimal control as explained in [92]. For $m = 1$, the inverse-optimality is also proved in [5, 10]. The above performance equation as illustrated in Section 3.2 requires the HJI (A.8) equation to be solved.

Theorem 5 (Optimality [5]). *For a control law* (3.20)*, applied to a robot manipulator* (2.1)*, we get an optimal solution if the following condition is satisfied*

$$\frac{\rho^{-1}(|\boldsymbol{x}|)}{|\boldsymbol{x}_\triangle|} = \frac{1}{\gamma^2},$$

where without loss of generality $\alpha = 1$ is considered for simplicity.

Proof. It is already proved under conditions A1-A3 that matrix $\boldsymbol{Q}$ (3.8) is positive definite. Condition 2 is evident because we select $\boldsymbol{R}^{-1}$ such that $1/\gamma^2$ cancels out the disturbance term. The HJI equations can be used in the performance equation (3.22) as

$$J(\boldsymbol{x}) = -\lim_{t\to\infty}\int_0^t \boldsymbol{x}^T(\boldsymbol{A}^T\boldsymbol{S} + \boldsymbol{S}\boldsymbol{A} - \boldsymbol{S}\boldsymbol{B}\tilde{\boldsymbol{K}}\boldsymbol{B}^T\boldsymbol{S})\boldsymbol{x}\ \mathrm{d}\tau + \int_0^t \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u} - \gamma^2\boldsymbol{d}^T\boldsymbol{d}\ \mathrm{d}\tau,$$

$$\begin{aligned} &= -\int_0^\infty \boldsymbol{x}^T(\boldsymbol{A}^T\boldsymbol{S} + \boldsymbol{S}\boldsymbol{A})\boldsymbol{x} + 2\boldsymbol{x}^T\boldsymbol{S}\boldsymbol{B}\boldsymbol{u}\ \mathrm{d}\tau \\ &+ \int_0^\infty \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u} + 2\boldsymbol{x}^T\boldsymbol{S}\boldsymbol{B}\boldsymbol{u} + \boldsymbol{x}^T\boldsymbol{S}\boldsymbol{B}\tilde{\boldsymbol{K}}\boldsymbol{B}^T\boldsymbol{S}\boldsymbol{x}\ \mathrm{d}\tau \\ &+ \int_0^\infty -\gamma^2\boldsymbol{d}^T\boldsymbol{d} + 2\boldsymbol{x}^T\boldsymbol{S}\boldsymbol{B}\boldsymbol{d} - 2\boldsymbol{x}^T\boldsymbol{S}\boldsymbol{B}\boldsymbol{d}\ \mathrm{d}\tau \end{aligned}$$

$$
\begin{aligned}
&= -\int_0^\infty 2V_{\boldsymbol{x}}\boldsymbol{A}\boldsymbol{x} + 2V_{\boldsymbol{x}}\boldsymbol{B}\boldsymbol{u} + 2V_{\boldsymbol{x}}\boldsymbol{B}\boldsymbol{d}\ \mathrm{d}\tau \\
&+ \int_0^\infty (\boldsymbol{u} + \boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{S}\boldsymbol{x})^T\boldsymbol{R}(\boldsymbol{u} + \boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{S}\boldsymbol{x})\ \mathrm{d}\tau \\
&- \gamma^2\int_0^\infty (\boldsymbol{d} - \frac{1}{\gamma^2}\boldsymbol{B}^T\boldsymbol{S}\boldsymbol{x})^T(\boldsymbol{d} - \frac{1}{\gamma^2}\boldsymbol{B}^T\boldsymbol{S}\boldsymbol{x})\ \mathrm{d}\tau, \\
&- \int_0^\infty (\frac{\rho^{-1}(|\boldsymbol{x}|)}{|\boldsymbol{x}_\triangle|} - \frac{1}{\gamma^2})\boldsymbol{x}^T\boldsymbol{S}\boldsymbol{B}\boldsymbol{B}^T\boldsymbol{S}\boldsymbol{x}\ \mathrm{d}\tau, \\
&= -2V(\boldsymbol{x}(0)) + \\
&- \int_0^\infty (\boldsymbol{u} + \boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{S}\boldsymbol{x})^T\boldsymbol{R}(\boldsymbol{u} + \boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{S}\boldsymbol{x})\ \mathrm{d}\tau \\
&- \gamma^2\int_0^\infty (\boldsymbol{d} - \frac{1}{\gamma^2}\boldsymbol{B}^T\boldsymbol{S}\boldsymbol{x})^T(\boldsymbol{d} - \frac{1}{\gamma^2}\boldsymbol{B}^T\boldsymbol{S}\boldsymbol{x})\ \mathrm{d}\tau, \\
&- \int_0^\infty (\frac{\rho^{-1}(|\boldsymbol{x}|)}{|\boldsymbol{x}_\triangle|} - \frac{1}{\gamma^2})|\boldsymbol{x}_\triangle|^2\ \mathrm{d}\tau.
\end{aligned}
$$

In the last step, we assume that the final state fulfills $V(\boldsymbol{x}(\infty)) = 0$. Considering $\boldsymbol{u}$ to be equal to (A.9), we get the optimal solution with respect to the PI mentioned above. Note that

$$\boldsymbol{d} = \frac{1}{\gamma^2}\boldsymbol{B}^T\boldsymbol{S}\boldsymbol{x}$$

is the worst case disturbance as explained in [114]. To achieve the minimum cost function, the following should be satisfied:

$$\frac{\rho^{-1}(|\boldsymbol{x}|)}{|\boldsymbol{x}_\triangle|} = \frac{1}{\gamma^2}.$$

Hence it is proved that

$$\boldsymbol{R}^{-1} = \tilde{\boldsymbol{K}} + \frac{\rho^{-1}(|\boldsymbol{x}|)}{|\boldsymbol{x}_\triangle|}\boldsymbol{I} = \tilde{\boldsymbol{K}} + \frac{\boldsymbol{I}}{\gamma^2}$$

gives the optimal solution. By considering $\boldsymbol{R}^{-1} = a_{m-1}\boldsymbol{I}$, we get to the proposed controller from Theorem 2. □

3.4 Performance Analysis and Stability

In this section, the primary focus is to discuss and quantify the performance of the controller. Then based on a performance criterion, the parameters of the proposed controller are evaluated. The performance criterion in this dissertation is an upper bound on the absolute values of the joint errors at steady-state.

Theorem 6 (Performance). *For the control law*

$$\hat{\boldsymbol{\phi}} = \left(\tilde{\boldsymbol{K}} + \frac{\boldsymbol{I}}{\gamma^2}\right)\boldsymbol{x}_\triangle, \quad \boldsymbol{x}_\triangle = \boldsymbol{B}^T\boldsymbol{S}\boldsymbol{x} \tag{3.23}$$

with the conditions given in Theorem 3, the performance limitation is given as

$$|\boldsymbol{x}|_{\mathrm{P,L}} \leq \frac{2|\boldsymbol{S}\boldsymbol{B}|}{\lambda_{\min}}|\boldsymbol{\phi}_{\max}|, \tag{3.24}$$

where

$$\lambda_{min} = \lambda_{min}\Big(\boldsymbol{Q} + (2a_{m-1} - \tilde{\boldsymbol{K}})\boldsymbol{S}\boldsymbol{B}\boldsymbol{B}^T\boldsymbol{S}\Big),$$

and $|\boldsymbol{x}|_{\mathrm{P,L}}$ *is the maximum allowed error in steady-state.*

Proof. The derivative of the Lyapunov function (3.14) is given as

$$\dot{V} = \frac{1}{2}\boldsymbol{x}^T(\boldsymbol{A}^T\boldsymbol{S} + \boldsymbol{S}\boldsymbol{A})\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{S}\boldsymbol{B}\tilde{\boldsymbol{\phi}}. \tag{3.25}$$

The above equation can also be written as

$$\dot{V} = \frac{1}{2}\boldsymbol{x}^T(\boldsymbol{A}^T\boldsymbol{S} + \boldsymbol{S}\boldsymbol{A} - \boldsymbol{S}\boldsymbol{B}\tilde{\boldsymbol{K}}\boldsymbol{B}^T\boldsymbol{S})\boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^T\boldsymbol{S}\boldsymbol{B}\tilde{\boldsymbol{K}}\boldsymbol{B}^T\boldsymbol{S}\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{S}\boldsymbol{B}\tilde{\boldsymbol{\phi}},$$

Applying (3.23), we get

$$\tilde{\boldsymbol{\phi}} = -\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{S}\boldsymbol{x} - \int_m \frac{\mathrm{d}^m\boldsymbol{\phi}}{\mathrm{d}t^m}.$$

For the sake of simplicity, let $\tilde{K}$ be a scalar quantity. The derivative of the Lyapunov function is

$$\dot{V} = -\frac{1}{2}\boldsymbol{x}^T(\boldsymbol{Q} + (2a_{m-1} - \tilde{K})\boldsymbol{S}\boldsymbol{B}\boldsymbol{B}^T\boldsymbol{S})\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{S}\boldsymbol{B}\int_m \frac{\mathrm{d}^m\boldsymbol{\phi}}{\mathrm{d}t^m}, \tag{3.26}$$

where $a_{m-1}\boldsymbol{I} = \boldsymbol{R}^{-1}$. To find the performance limitation of the controller in terms of errors, (3.26) can be written as

$$\dot{V} \leq -\frac{1}{2}\lambda_{min}\boldsymbol{x}^T\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{S}\boldsymbol{B}\boldsymbol{\phi}. \tag{3.27}$$

According to that, $\dot{V} < 0$ is possible if and only if

$$|\boldsymbol{x}| \leq \frac{2|\boldsymbol{S}\boldsymbol{B}|}{\lambda_{min}}|\boldsymbol{\phi}_{\max}|.$$

This also proves Theorem 2. □

Remark 20. *The disturbance term* $\boldsymbol{\phi}$ *is a function of* $\boldsymbol{q}, \dot{\boldsymbol{q}}$ *and* $\ddot{\boldsymbol{q}}$*. It is relatively easy to find the estimate of the maximum value of* $|\boldsymbol{\phi}|$*, so to ensure that the joint errors never exceed a predefined value. For that reason, the above equation must be satisfied. From equation* (3.27)*, it can also be concluded that the proposed control method is indeed ISS.*

3.4.1 Evaluation of Controller parameters

After discussing the performance limitation, the next step is to use (3.24) for the identification of the controller parameters. Based on the previous Theorem, we can find the minimum permissible values for the controller parameters $\boldsymbol{K}_{\mathrm{p}}, \boldsymbol{K}_{\mathrm{d}}$ and a_i. For $m = 1$, it is possible to find the parameters analytically, however, for $m \geq 2$, there is no analytical solution, but the parameters can be determined using numerical techniques for solving systems of nonlinear equations, such as the Newton-Raphson method. We start with $m = 1$,

$$\dot{V} \leq -\frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{S}\boldsymbol{B}\boldsymbol{\phi} < 0.$$

To find an analytical solution for the parameters, the term $-(2a_{m-1} - \tilde{K})\boldsymbol{x}^T\boldsymbol{SBB}^T\boldsymbol{Sx}$ is ignored because $\dot{V} < 0$ if $a_{m-1} > \tilde{K}/2$. To satisfy the inequality, the following equations must hold:

$$\frac{1}{2}\boldsymbol{Q}|\boldsymbol{x}| > \boldsymbol{SB}|\boldsymbol{\phi}_{\max}| \tag{3.28}$$

$$\frac{1}{2}\begin{bmatrix} \tilde{\boldsymbol{K}}\boldsymbol{K}_{\mathrm{p}}^2 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{K}}(\boldsymbol{K}_{\mathrm{d}}^2 - 2\boldsymbol{K}_{\mathrm{p}}) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \tilde{\boldsymbol{K}} \end{bmatrix}\boldsymbol{x} > \begin{bmatrix} \boldsymbol{K}_{\mathrm{p}} \\ \boldsymbol{K}_{\mathrm{d}} \\ \boldsymbol{I} \end{bmatrix}|\boldsymbol{\phi}_{\max}|.$$

Solving the above system of equations, we get

$$\tilde{\boldsymbol{K}}(i,i) > \frac{2|\boldsymbol{\phi}_{\max}|}{\dot{\boldsymbol{e}}(i)},$$

$$\boldsymbol{K}_{\mathrm{p}}(i,i) > \frac{2|\boldsymbol{\phi}_{\max}|}{\tilde{\boldsymbol{K}}(i,i)\int \boldsymbol{e}(i)\,\mathrm{d}t},$$

$$\boldsymbol{K}_{\mathrm{d}}(i,i) > \sqrt{2\boldsymbol{K}_{\mathrm{p}}(i,i) + \left(\frac{|\boldsymbol{\phi}_{\max}|}{\tilde{\boldsymbol{K}}(i,i)\boldsymbol{e}(i)}\right)^2} - \frac{|\boldsymbol{\phi}_{\max}|}{\tilde{\boldsymbol{K}}(i,i)\boldsymbol{e}(i)},$$

where $\tilde{\boldsymbol{K}}(i,i), \boldsymbol{K}_{\mathrm{p}}(i,i)$ and $\boldsymbol{K}_{\mathrm{d}}(i,i)$ are scalars representing the i^{th} diagonal elements of the corresponding matrices. Similarly, $\boldsymbol{e}(i)$ is the i^{th} element of the vector $\boldsymbol{e}$. Hence, for a given scenario, the controller parameters can be easily identified by specifying the maximum acceptable $\boldsymbol{e}, \dot{\boldsymbol{e}}$ and $\int \boldsymbol{e}\mathrm{d}t$. As mentioned earlier, $-(2a_{m-1} - \tilde{K})\boldsymbol{x}^T\boldsymbol{SBB}^T\boldsymbol{Sx}$ is ignored to find the controller parameters. This term can be used to find the minimum values of parameters that will satisfy stability criteria for the system. However, there is no analytical solution in that case. The same procedure can be used to find the controller parameters for $m \geq 2$. More about the parameters of the adaptive control is presented in the next section.

3.4.2 Discussion on $\hat{\boldsymbol{\phi}} \to \boldsymbol{\phi}$

As discussed, for an intelligent PD controller, the estimated model must approach the real model. As a result, we get a linear error dynamics with predefined performance as (2.5). The performance is then set by using proper values for $\boldsymbol{K}_{\mathrm{p}}$ and $\boldsymbol{K}_{\mathrm{d}}$.

The system dynamical vector $\boldsymbol{\phi}$ is approximated with piecewise constant functions for the first-order controller. This kind of approximation is common in mathematical analysis [83]. The proposed first-order differential equation is

$$\frac{\mathrm{d}\tilde{\boldsymbol{\phi}}}{\mathrm{d}t} + a_0\tilde{\boldsymbol{\phi}} = \boldsymbol{0}. \tag{3.29}$$

Since $\dot{\phi}$ is assumed to be zero for a short time interval, (3.29) is equivalent to (2.30) and $\tilde{\boldsymbol{\phi}}$ will eventually approach zero.

Similarly, the system vector $\boldsymbol{\phi}$ can be approximated by a ramp function for a small time interval

$$\boldsymbol{\phi}(t_o + h) \approx \boldsymbol{\phi}(t_o) + h\dot{\boldsymbol{\phi}}(t_o), \tag{3.30}$$

where $h > 0$ and we consider the case when the controller order $m = 2$, the control law (3.4) can be written as

$$\frac{\mathrm{d}^2\tilde{\boldsymbol{\phi}}(t_o)}{\mathrm{d}t^2} + a_1\frac{\mathrm{d}\tilde{\boldsymbol{\phi}}(t_o)}{\mathrm{d}t} + a_0\tilde{\boldsymbol{\phi}}(t_o) = \boldsymbol{0}. \tag{3.31}$$

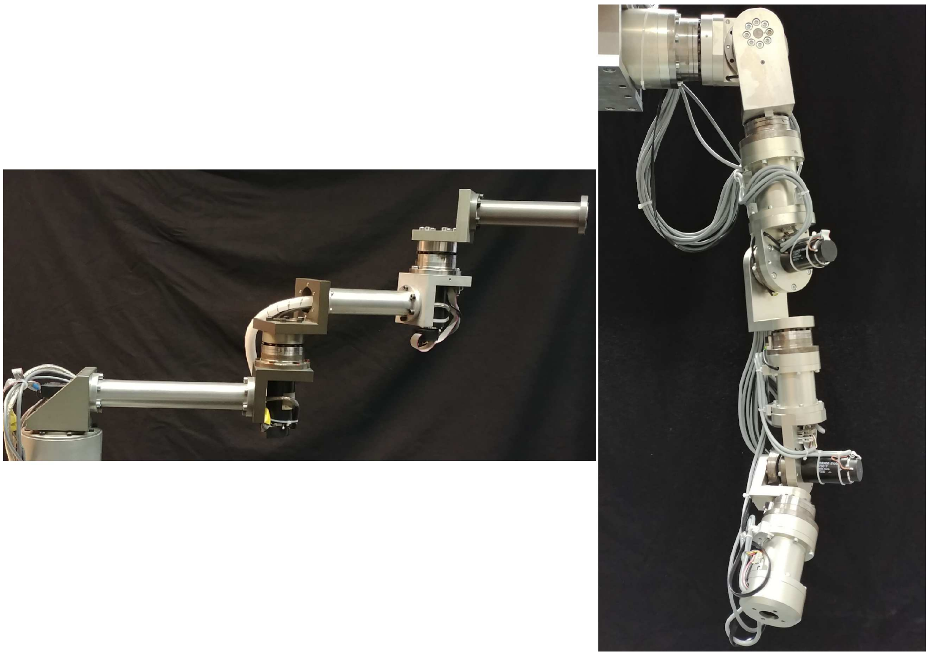


Figure 3.3: 3-DoF (left) and 7-DoF (right) robot manipulators used in experiments.

From basic root-locus technique [94], the values for a_0 and a_1 can be chosen such that $\tilde{\boldsymbol{\phi}} \to \mathbf{0}$. A similar analogy can be used for the third-order controller, where the Taylor approximation will consider even the second-derivative of $\boldsymbol{\phi}$ and hence the convergence will be faster. The condition (3.30) can be satisfied if the gains a_i are high enough to converge during the time h. We have shown the convergence of the estimated model towards the real model in the simulation results.

3.5 Results

A 3-DoF and a 7-DoF robot manipulator (see Fig. 3.3) is considered to validate the proposed robust-adaptive controller. MATLAB Simulink with a sampling rate of 1 ms is used to implement the controller. The minimum allowed values for the controller are calculated using equation (3.28) and thus satisfy the performance criteria. In finding the controller parameters, ignoring the term $-\boldsymbol{x}^T\boldsymbol{S}\boldsymbol{B}\boldsymbol{B}^T\boldsymbol{S}\boldsymbol{x}$ will still provide us with the optimal values. Although to find the absolute minimum parameters, this term should be included and the Newton-Raphson method can be used to find the minimum values for acceptable controller parameters. An easy and intuitive way to find the controller parameters is to follow the steps mentioned below:

1. Find the minimum values for the controller parameters using (3.24) that satisfy the condition for the maximum allowed joint-error at steady-state.

2. Find the values of $\boldsymbol{K}_\mathrm{p}$ and $\boldsymbol{K}_\mathrm{d}$ using a performance criterion like rise-time, settling-time, etc.

3. Take a_i such that the gain of (3.31) is 3-5 times higher than (2.5).

Table 3.1: Controller parameters: (a) ignoring $-\boldsymbol{x}^T\boldsymbol{SBB}^T\boldsymbol{Sx}$, (b) including $-\boldsymbol{x}^T\boldsymbol{SBB}^T\boldsymbol{Sx}$, (c) values used in controller.

Var.	$m = 1$			$m = 2$		
	(A)	(B)	(C)	(A)	(B)	(C)
$\tilde{K}$	222	3.3	68.9	174.7	63.4	68.9
K_p	200	78	400	57	100	400
K_d	32.3	0.0	40	39.9	20	40
b_0	-	-	-	0	20	69.6

The next two subsections present the simulation and experimental results for the 3-DoF manipulator. Because of the horizontal setup, the planar robot dynamics has no gravity term. Section 3.5.3 discusses the effect of the order and gain of the controller. The results for the 7-DoF manipulator are shown in Section 3.5.4.

3.5.1 Simulation Results

The desired trajectory for the end-effector of the manipulator is shown in Fig. 3.4. It was chosen heuristically and is in the workspace of the manipulator. The end-effector starts on the x-axis at 0.88 m from the base of the robot and it takes 6.17 s to complete one rotation. In this study, $m = 1$ and $m = 2$ is investigated. The threshold values for the state errors are

$$\boldsymbol{x}_{\max_1} = [10^{-3} \;\; 10^{-2} \;\; 10^{-1}]^T, \tag{3.32}$$

$$\boldsymbol{x}_{\max_2} = [10^{-4} \;\; 10^{-3} \;\; 10^{-2} \;\; 10^{-1}]^T, \tag{3.33}$$

where $\boldsymbol{x}_{\max_1}$ and $\boldsymbol{x}_{\max_2}$ are the maximum allowed errors in steady-state for $m = 1$ and $m = 2$, respectively.

The performance of the robot manipulator is shown in Fig. 3.5. The controller for $m = 2$ suffers from peaking and as a consequence, the trajectory of the end-effector shows a larger overshoot during the transient phase, see Fig. 3.5 and 3.6. Under the controller parameters mentioned in Table 3.1, the maximum permissible error after the transient response does not exceed (3.32) and (3.33), see Fig. 3.7. Because the selected values for the controller parameters are higher than the required values for the specified performance, the maximum error $\boldsymbol{e}_{\max}$ is much smaller than the threshold value, which is 0.01 radians in our case ($\boldsymbol{x}_{\max_1}[2] = \boldsymbol{e}$ and $\boldsymbol{x}_{\max_2}[3] = \boldsymbol{e}$).

Fig. 3.8 shows the mismatch between the real and approximated model. Considering the same convergence rate for $m = 1$ and $m = 2$, the mismatch $\tilde{\boldsymbol{\phi}}$ is getting smaller as we increase the order of the controller. The obvious reason is that a higher order Taylor approximation gives better results, see Section 3.4.2. However, the limitation of taking a higher order approximation is the peaking effect during the transient. One way of dealing with this problem is to start with $m = 1$ and as the mismatch approaches zero, we shift to a higher order approximation, because after the transient response, the torques for any order m are equal.

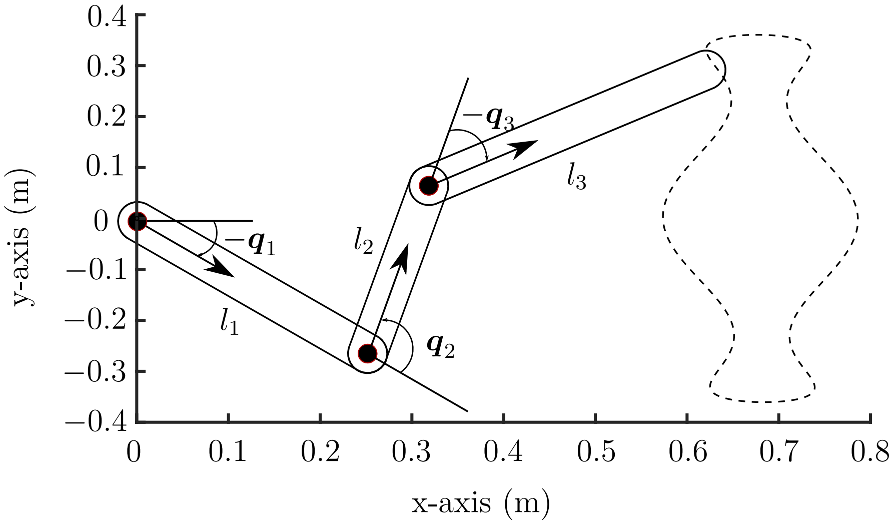


Figure 3.4: Top view of the 3-DoF robot manipulator along with the desired trajectory. The lengths are: $l_1 = 0.30\ m$, $l_2 = 0.24\ m$ and $l_3 = 0.34\ m$.

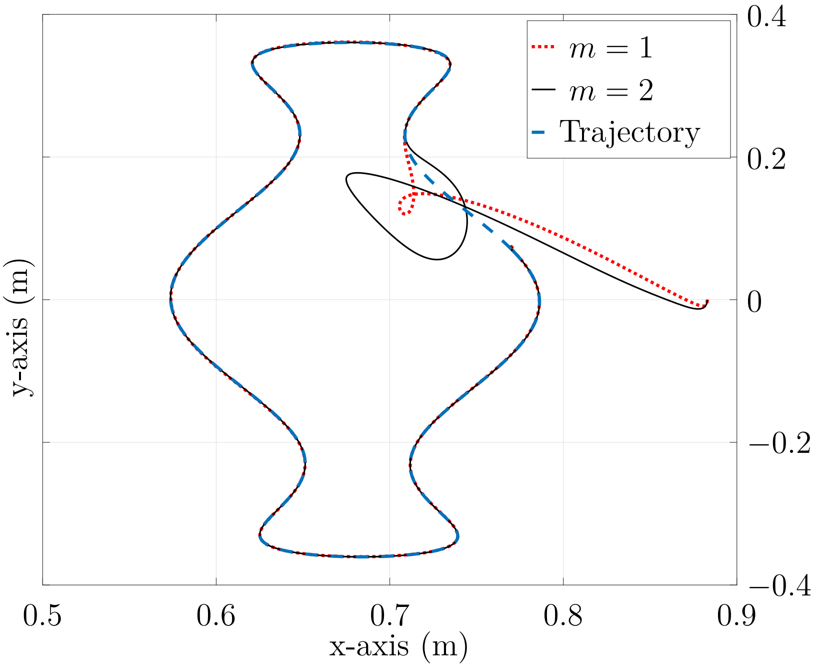


Figure 3.5: Simulation results: trajectory tracking in Cartesian space. The end-effector starts from (0.88 m,0) in the x-y coordinate system.

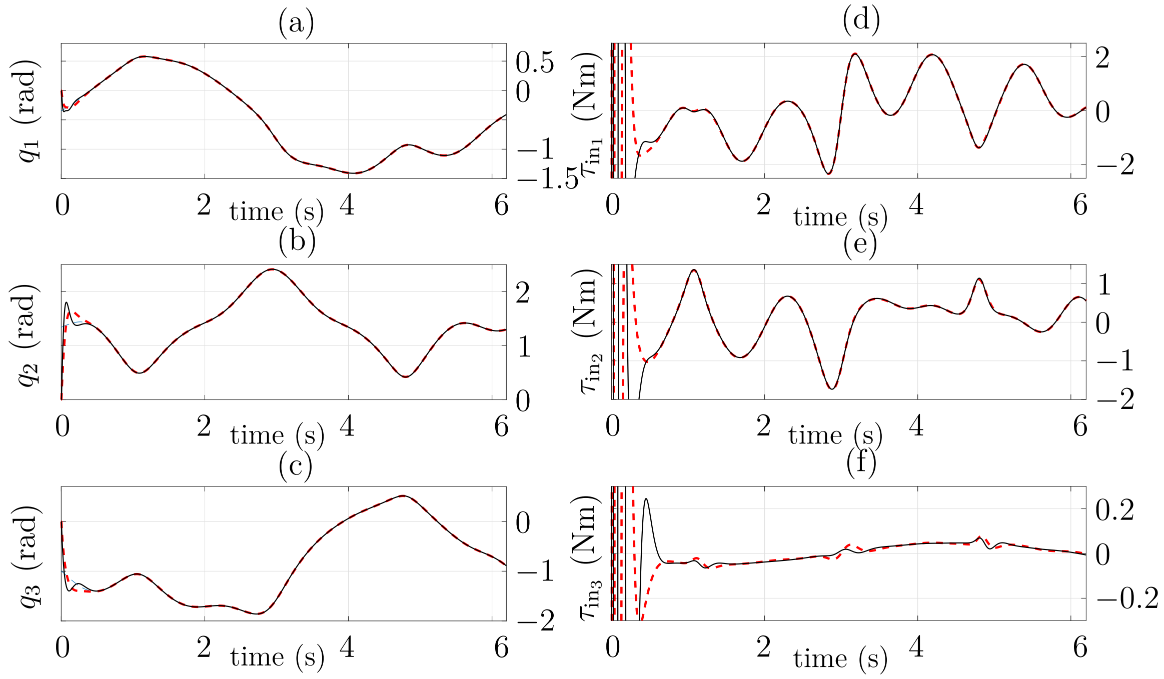


Figure 3.6: Simulation results: (a), (b) and (c) represent the three joint angles with respect to time. (d), (e) and (f) represent the three joint torques with respect to time. Dashed line: $m = 1$, solid line: $m = 2$.

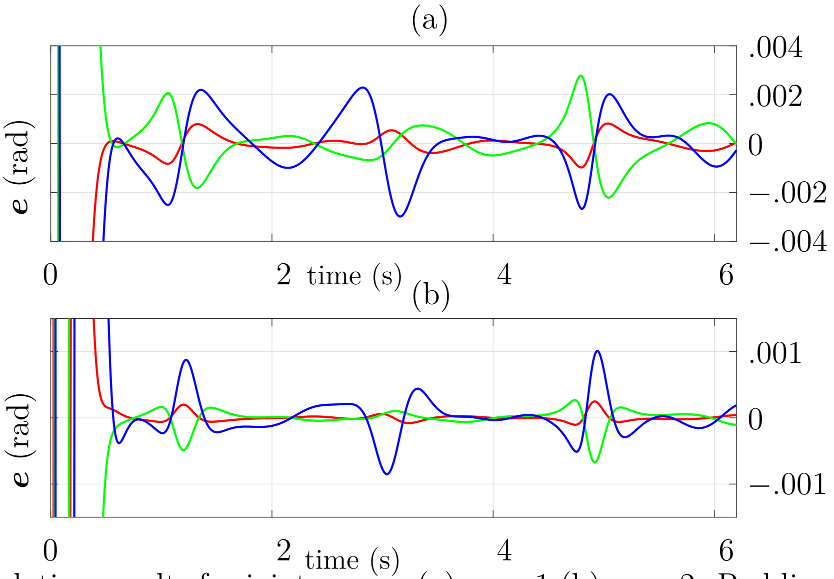


Figure 3.7: Simulation results for joint errors: (a) $m = 1$ (b) $m = 2$. Red line: q_1, green line: q_2, blue line: q_3. These figures show that the controller parameters satisfy the performance criteria of (3.32) and (3.33).

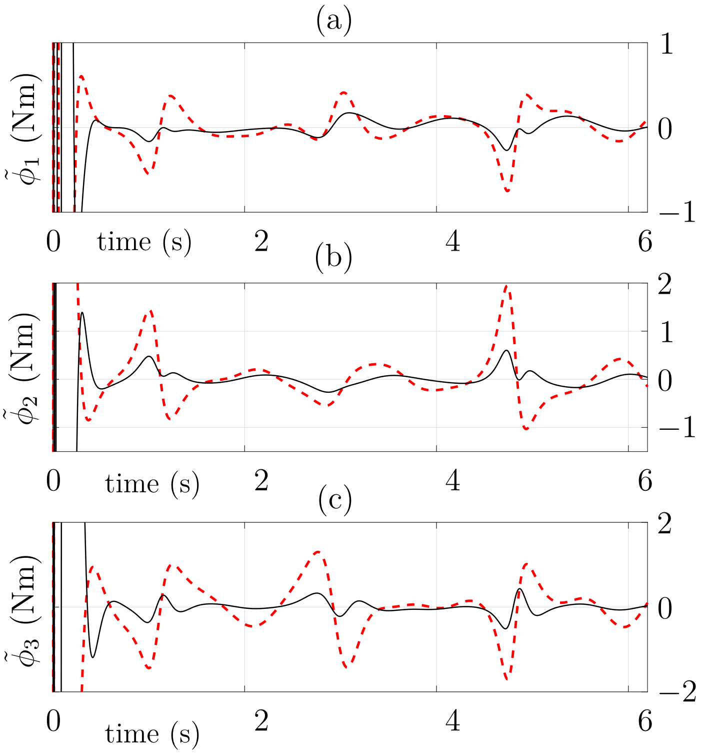


Figure 3.8: Difference between $\hat{\boldsymbol{\phi}}$ and $\boldsymbol{\phi}$. Dashed line: $m = 1$, solid line: $m = 2$.

3.5.2 Experimental Results

In simulations, we did not consider input saturation, disturbances and unmodeled parameters. Fortunately, they do not deteriorate the performance during steady-state because these disturbances can be considered as part of $\boldsymbol{\phi}$, which is estimated by the controller during runtime. In experiments, the joint space result for the 3-DoF robot manipulator (Fig. 3.3) is shown in Fig. 3.9, and the trajectory tracking is shown in Fig. 3.10. The long transient response is because of the torque-saturation; nevertheless, the performance criterion during steady-state is satisfied for both simulations and experiments. The errors in the joint angles are shown in Fig. 3.11, which satisfy the performance criteria (3.32) and (3.33). In the experiments, a comparison between real and approximated dynamics cannot be shown because we do not know the exact values of the actual system dynamics $\boldsymbol{\phi}$. As long as $\boldsymbol{\phi}$ is bounded, $\tilde{\boldsymbol{\phi}}$ will approach zero even in experiments.

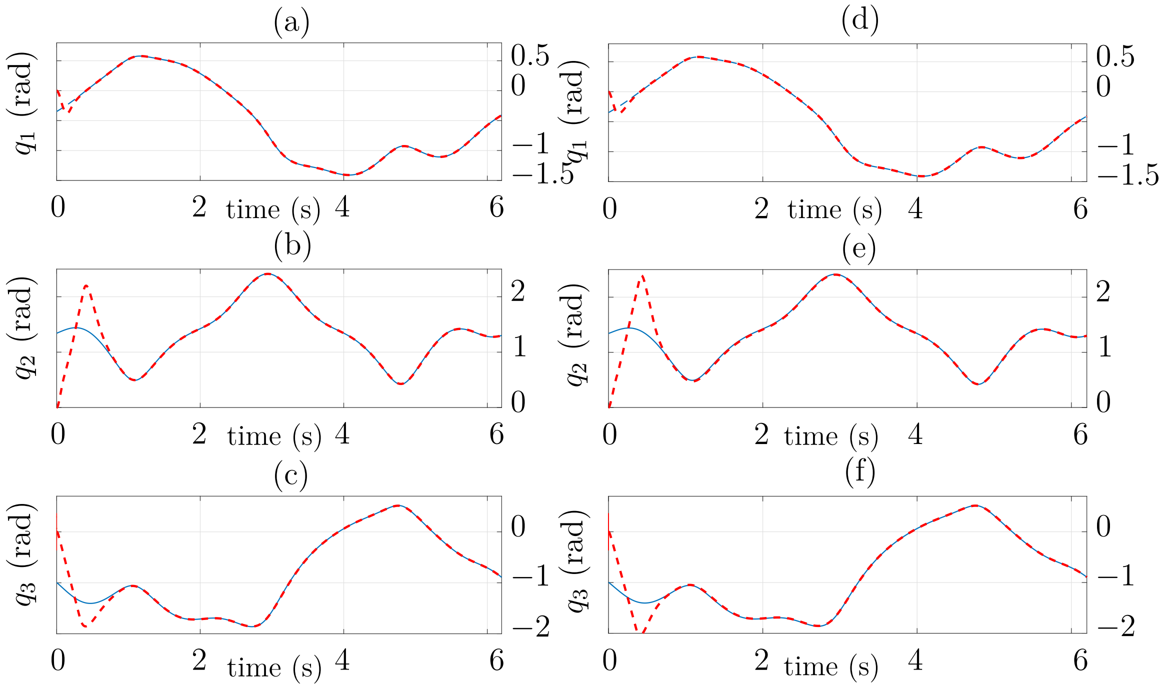


Figure 3.9: Experimental results: (a), (b) and (c) represent the three joint angles for $m = 1$ with respect to time. (d), (e) and (f) represent the three joint angles for $m = 2$. Dashed line: q, solid line: q_d.

3.5.3 Effects of Order and Gain of the Controller

For the proposed adaptive controller, the system dynamics $\boldsymbol{\phi}$ is approximated by a Taylor series of order $m - 1$. Thus, for a first order approximation, the approximated system dynamics $\hat{\boldsymbol{\phi}}$ can only approach the true dynamics if $\boldsymbol{\phi}$ is constant. However, $\boldsymbol{\phi}$ depends on joint angles, velocities and acceleration, so unless the desired trajectory is slowly moving or

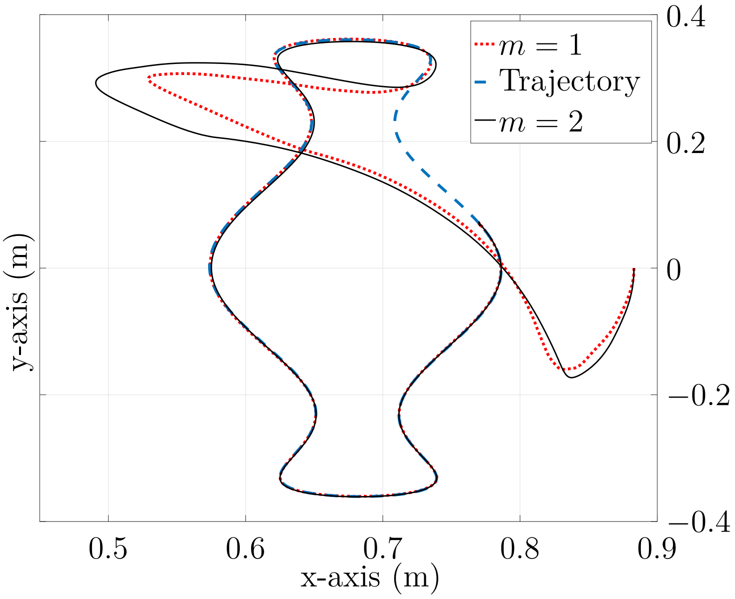


Figure 3.10: Experimental results: trajectory tracking in Cartesian space. Dashed line: $m = 1$, solid line: $m = 2$. The end-effector starts from (0.88 m,0) in the x-y coordinate system.

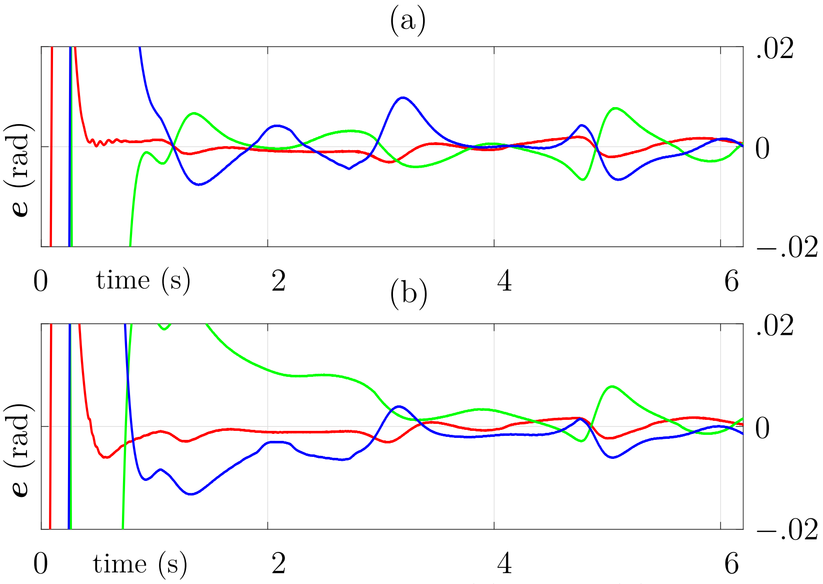


Figure 3.11: Experimental results for joint errors: (a) $m = 1$ (b) $m = 2$. Red line: q_1, green line: q_2, blue line: q_3. These figures show that the controller parameters satisfy the performance criteria of (3.32) and (3.33).

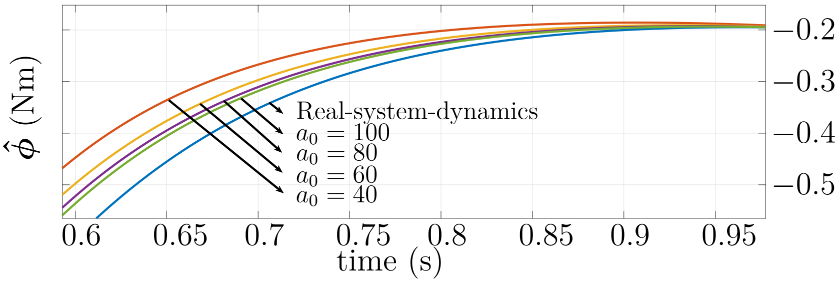


Figure 3.12: First order approximation with different a_0.

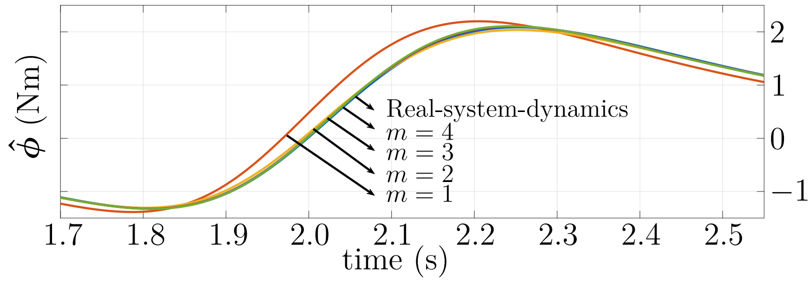


Figure 3.13: Various orders of control law with same convergence rate.

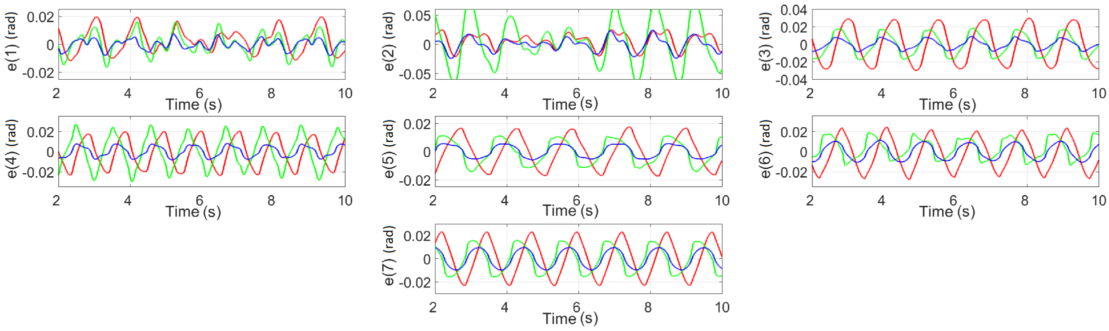


Figure 3.14: Errors in the joint angles: – PD control, – robust control [5] and – proposed control.

Table 3.2: Error in the joint angles (rad).

joint	PD	Chung *et al.*	proposed
1	0.798×10^{-4}	0.537×10^{-4}	0.123×10^{-4}
2	1.910×10^{-4}	14.40×10^{-4}	1.240×10^{-4}
3	4.230×10^{-4}	1.440×10^{-4}	0.269×10^{-4}
4	2.050×10^{-4}	2.550×10^{-4}	0.276×10^{-4}
5	1.220×10^{-4}	0.731×10^{-4}	0.182×10^{-4}
6	2.660×10^{-4}	1.090×10^{-4}	0.548×10^{-4}
7	2.350×10^{-4}	1.260×10^{-4}	0.519×10^{-4}

the gain a_0 is high, there will always be a slight difference between $\hat{\boldsymbol{\phi}}$ and $\boldsymbol{\phi}$, as shown in Fig. 3.12.

By increasing the gain a_0, the estimated dynamics will converge faster, but more torque is required if the error is high. Also, if $\boldsymbol{\phi}$ is approaching a constant value, the estimated value approaches the real system dynamics because the system acts like a Type 1 system [94]. However, if $\boldsymbol{\phi}$ is varying with time, the first-order will never reach the real system dynamics as shown in Fig. 3.13.

To sum up, a first order approximation has no error for a constant $\boldsymbol{\phi}$, a second order has no error for a ramp $\boldsymbol{\phi}$, a third order approximation has no error for a hyperbolic $\boldsymbol{\phi}$ and so on. However, by selecting a sufficiently high gain a_i along with a second order approximation is quantitatively suitable to approximate the system dynamics. Fig. 3.13 shows the response of various orders for the same convergence rate.

The next point of interest is the effect of the order of the controller on the input torque. Once the error $\tilde{\boldsymbol{\phi}}$ approaches zero, then the torques will be equal for any order because the input torque is equal to

$$\boldsymbol{\tau}_{\text{in}} = \hat{\boldsymbol{\phi}} + \ddot{\boldsymbol{q}}_{\text{d}} - \boldsymbol{K}_{\text{d}}\dot{\boldsymbol{e}} - \boldsymbol{K}_{\text{p}}\boldsymbol{e}. \tag{3.34}$$

As long as $\hat{\boldsymbol{\phi}} \approx \boldsymbol{\phi}$, the torques for different m will remain equal.

3.5.4 Experimental Results for 7-DoF Robot

The proposed adaptive controller is also used to control a 7-DoF manipulator, see Fig. 3.3. The motivation is to show that the performance criterion of (3.24) is satisfied for a 7 DoF robot. We performed experiments under the assumption that the system parameters are not known, and for that reason, we compared our proposed controller with a couple of model-free control methods. The performance in terms of the integral square error for the proposed controller is shown along with a simple PD and robust controller proposed by Chung *et al.* [5]. A more detailed comparison is given in [3]. The values of $\boldsymbol{K}_{\text{p}}$ and $\boldsymbol{K}_{\text{d}}$ are the same for the proposed and PD control for the experiment. For the robust control [5], a trial and error method is chosen to tune the controller parameters because of the assumption that the system parameters are not available. The integral square error

$$I(i) = \frac{1}{T_2 - T_1}\int_{T_1}^{T_2} e(i)^2 \mathrm{d}t, \quad i = 1, 2, .., 7 \tag{3.35}$$

for the three control methods are shown in Table 3.2. Fig. 3.14 shows the error for each joint angle for sinusoidal references of different frequency and amplitude. During the steady-state, the integral square error for the proposed controller is much smaller compared to the other control schemes. The order of the controller is $m = 1$, and the error reduction can be improved by taking higher order controller, however, the transient response will be worse. The gain of parameters of the adaptive controller can be reduced during the transient phase for improved response.

3.6 Summary

The main contribution of this chapter is to implement an adaptive controller that satisfies a predefined performance with few tuning parameters. Apart from the PD parameters, the total number of control parameters are equal to the order of the controller. Ideally, a higher-order controller converges quickly, however, introducing noise in the system states. It has been shown that a second-order controller is sufficient to achieve the desired performance. For bounded disturbance, the controller is shown to be robust using the HJI equations. An inverse-optimal control method is utilized to evaluate an optimal cost function for the proposed controller. The control law is derived by using input-to-state-stability analysis that also ensures the stability of the system. The parameters of the proposed controller are identified using a quantitative performance analysis, which put an upper bound on the absolute values of the joint errors. The adaptive controller also ensures the removal of mismatch between real and estimated system model during feedback linearization to get better performance in terms of joint errors. Once the mismatch is removed, a PD control can perform well to get the desired performance.

Open problem

There are two major implementation issues that are usually not considered in many state-of-the-art control schemes: (a) unknown state-dependent input saturation and (b) peaking during transient-phase. Further improvements in the performance of the controller is proposed in the next chapter.

4

Performance Enhancement of Adaptive Controller

Almost all of the state-of-the-art trajectory tracking control methods do not consider input saturation, although saturation decisively affect the performance of the control system in terms of state errors. The input constraints are usually state dependent and uncertain, which makes it difficult to include them in the control loop. One way of considering input constraints is to adjust the speed of the reference path such that the required inputs are within the allowed range. Such approaches are called time-optimal path tracking [117]. Bobrow *et al.* [118] showed different techniques to find the minimum or optimal-time for the tracking of the end-effector while satisfying the input constraints. The optimal-time for the reference trajectory is usually evaluated from the inverse-dynamics of the manipulator, which allows finding the maximum allowed joint-acceleration under the permitted torques. The speed of the trajectory is adjusted according to the maximum joint-acceleration. However, these techniques require information about the input constraints and the system model for the inverse-dynamics [119–121].

Usually one of the design objectives is that the end-effector follows the desired trajectory even if time is compromised. An approach to include the input constraints by using time scaling is proposed by Shiriaev *et al.* [29, 122–124]. This method focuses on the orbital stability of the desired path. The technique converts the system dynamics into a transverse dynamics using a moving Poincare section [125]. Any trajectory starting in the vicinity of the desired periodic solution is forced to follow the desired trajectory. In orbital stability, the primary objective is to ensure that the closed-loop system converges to an orbit of the desired trajectory rather than considering the stability as a function of time. But again, the design of such a controller requires an estimated model and information about the time-varying input constraints. Furthermore, for constant input constraints, Lewis *et al.* [126] proposed a technique that uses online $\mathcal{H}_\infty$ control. This technique uses a quasi-norm to transform the constrained control problem to an unconstrained problem. Ahanda *et al.* [127] proposed an adaptive back-stepping approach, applied on a manipulator using a support vector machine. The controller is robust to system uncertainties and external disturbances. The above-mentioned techniques require that the input constraints are known. Unfortunately, input constraints are not easy to estimate because of many reasons. For example, torque saturation varies with joint velocity because of back electromotive force in the motors, the motors have a maximum voltage rating that limits the maximum current, the drives have gain uncertainty, and also there is always some offset.

The next issue in control implementation is to ensure safety during the transient phase. In theory, it is feasible to use a high gain adaptive control for fast convergence. But this leads to peaking effects [113, 128] because of the large initial state error and high controller gain. Furthermore, there is high-frequency noise, which is partially suppressed by the intrinsic property of motor drives that act as low pass filters. One of the remedies is to choose a trade-off between fast response and high performance by selecting mid-range values for the controller parameters. Another critical issue is the tuning of the controller parameters. Tuning is a complicated task [1,49], and in some cases, it is not desirable to tune the controller by trial and error method. A detailed discussion about controllers tuning is given in Hayat *et al.* [1].

The objective of this chapter is the use of the model-free robust-adaptive controller that (a) gives an improved transient and steady-state response by incorporating unknown input saturation, (b) reduces the peaking effect during the transient phase, (c) provides robustness of the system under parameter variations and (d) allows intuitive tuning of the controller parameters. A second-order robust-adaptive controller was proposed in Chapter 3. As a first contribution, we present a simpler first-order version of this robust-adaptive controller with robust optimality and stability proofs. The advantage of the first-order controller is the ease of implementation and avoiding high-frequency noise. Two-players zero-sum game theory [129] is used to form a performance index (PI) that must be minimized and maximized with respect to inputs and disturbances, respectively. $\mathcal{H}_\infty$ optimality is used to perform the analysis, where an inverse-optimal approach is applied to find the PI for the given adaptive control.

This adaptive control allows for performance analysis in terms of the state errors. This means, in steady-state, the error only exceeds the maximum allowed value if there is input saturation. The second contribution of this chapter is the introduction of a trajectory-scaling formulation that modifies the desired trajectory depending on the input saturation. The desired trajectory is altered only when the motor drives are unable to generate the required torques for the nominal trajectory. The advantage of the proposed method is that no constraints information is needed and only states are used in the feedback loop. The proposed trajectory scaling is only applicable for control techniques that offer quantitative performance at steady-state.

For an improved transient response, and to avoid peaking without compromising the performance of the system, two techniques are presented in this chapter. The first procedure adds an extra term to the controller that is a filtered value of the system states [130]. The second method uses the well-known gain scheduling technique to improve the transient response. An analytical expression is introduced that modifies the controller gains to avoid peaking during transient phase by using only the state error.

Finally, the adaptive control [3] parameters can be evaluated with little information requirements, e.g., rise time, settling time, maximum allowed steady-state error, etc [94]. In general, most of the state-of-the-art control schemes have no tuning mechanism [1]. Tuning by trial and error might not be a feasible solution for experiments because of safety issues. The fourth contribution is the evaluation of the controller parameters for the adaptive control using performance limitations.

The rest of the chapter is organized as follows: Section 4.1 covers the contributions of the chapter. The validation of the proposed techniques using simulations and experiments is presented in Section 4.2. Conclusions are drawn in Section 4.3.

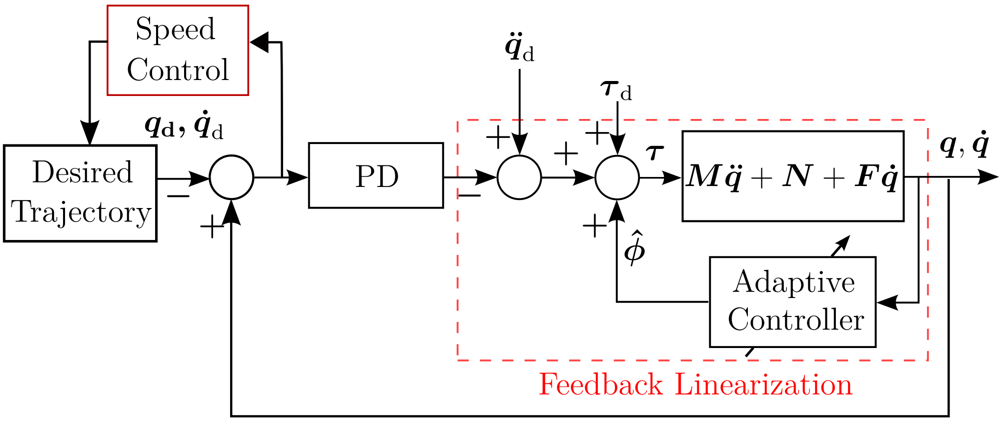


Figure 4.1: Adaptive control with speed adjustment.

4.1 Contribution

The contributions of the chapter are presented in four parts:

- Robustness and stability of the first-order adaptive control (2.30), see Section 4.1.1
- Compensation for unknown input saturation to improve the steady-state response, see Section 4.1.2
- Improvement of transient response, see Section 4.1.3
- Controller parameters tuning, see Section 4.1.4

As shown in Fig 4.1, the proposed controller has three feedback loops. The inner-loop in the ‘Feedback Linearization’ removes the mismatch in the estimation of ϕ. By considering a high value for the controller gain ‘a’, the mismatch will vanish quickly [3]. Once the vector $\hat{\phi} \to \phi$, the outer-loop will achieve the desired specification by tuning the Proportional-Derivative (PD) gains K_d and K_p.

To follow the desired trajectory, the controller might require inputs that are higher than the maximum allowed torques. Unfortunately, it is not easy to identify the bounds for input torques, so a third loop is devised ‘Speed Control’ that will modify the speed of the desired trajectory whenever the end-effector moves away from the predefined vicinity of the desired path. Finally, an intuitive approach is given to find the controller parameters after finding the minimum allowed values.

4.1.1 Robustness and Stability

Before we proof robustness of the control system (3.3), (2.30), we start with a brief introduction of game-algebraic Riccati (GARE) and HJI equations using the $\mathcal{H}_\infty$ optimality approach. Let a system be given in the state-space form as

$$\dot{x} = Ax + Bu + Dd, \tag{4.1}$$

where $\boldsymbol{x}, \boldsymbol{A}, \boldsymbol{u}$ and $\boldsymbol{d}$ are the states, transition matrix, input and disturbance, respectively. The $\mathcal{H}_\infty$ PI is

$$V(\boldsymbol{x}) = \min_{\boldsymbol{u}} \max_{\boldsymbol{d}} \int_{\tau}^{\infty} (\boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{u}^T \boldsymbol{R} \boldsymbol{u} - \gamma^2 ||\boldsymbol{d}||^2) dt. \tag{4.2}$$

The matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ are positive definite and $\gamma > 0$ is the sensitivity of the system with respect to the disturbance. The above optimal min-max control problem is also called two-players zero-sum problem, where one player ($\boldsymbol{u}$) tries to minimize the cost function, and the other player ($\boldsymbol{d}$) tries to maximize it [114]. The input to the system (4.1) that results in an optimal solution is

$$\boldsymbol{u}^* = -\boldsymbol{R}^{-1} \boldsymbol{B}^T \boldsymbol{S} \boldsymbol{x} \triangleq -\boldsymbol{K} \boldsymbol{x}. \tag{4.3}$$

Considering the worst-case disturbance, the GARE for the system is

$$\boldsymbol{A}^T \boldsymbol{S} + \boldsymbol{S} \boldsymbol{A} - \boldsymbol{S} \boldsymbol{B} \boldsymbol{R}^{-1} \boldsymbol{B}^T \boldsymbol{S} + 1/\gamma^2 \boldsymbol{S} \boldsymbol{D} \boldsymbol{D}^T \boldsymbol{S} + \boldsymbol{Q} = \boldsymbol{0}. \tag{4.4}$$

In the following, we use an inverse optimal approach to determine weighting matrices $\boldsymbol{Q}$, $\boldsymbol{S}$ and $\boldsymbol{R}$ given the controller (2.30) for (3.3).

Theorem 7. ***Robustness***: *The controller* (2.30) *for the error dynamics* (3.3) *is a robust* $\mathcal{H}_\infty$ *control with the PI* (4.2), *if*

$$\boldsymbol{Q} = \begin{bmatrix} \tilde{\boldsymbol{K}} \boldsymbol{K}_{\mathrm{p}}^2 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{K}}(\boldsymbol{K}_{\mathrm{d}}^2 - 2\boldsymbol{K}_{\mathrm{p}}) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \tilde{\boldsymbol{K}} \end{bmatrix}, \qquad \tilde{\boldsymbol{K}} > \boldsymbol{0}. \tag{4.5}$$

Proof. To rewrite (3.3) and (2.30) in the form (4.1) and (4.3), let $\boldsymbol{u} = \hat{\boldsymbol{\phi}}$, $\boldsymbol{d} = -\boldsymbol{\phi}$, $\boldsymbol{D} = \boldsymbol{B}$ and $\boldsymbol{x} = [\int \boldsymbol{e}^T dt \quad \boldsymbol{e}^T \quad \dot{\boldsymbol{e}}^T]^T \in \mathbb{R}^{3n \times 1}$. As a consequence,

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{0} & -\boldsymbol{K}_{\mathrm{p}} & -\boldsymbol{K}_{\mathrm{d}} \end{bmatrix} \in \mathbb{R}^{3n \times 3n}, \qquad \boldsymbol{B} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{I} \end{bmatrix} \in \mathbb{R}^{3n \times 1}. \tag{4.6}$$

The control input (2.30) can be written as

$$\hat{\boldsymbol{\phi}} = -a \begin{bmatrix} \boldsymbol{K}_{\mathrm{p}} & \boldsymbol{K}_{\mathrm{d}} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \int \boldsymbol{e} dt \\ \boldsymbol{e} \\ \dot{\boldsymbol{e}} \end{bmatrix} \triangleq -\boldsymbol{K} \boldsymbol{x}. \tag{4.7}$$

From $\boldsymbol{K}$ defined in (4.7), using the inverse optimal control technique, the matrices $\boldsymbol{S}$ and $\boldsymbol{Q}$ can be evaluated by (4.3) and (4.4) (Appendix B3). With the assumption, that $\boldsymbol{Q}$ is a diagonal matrix, we obtain (4.5) and

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{K}_{\mathrm{p}}^2 + \tilde{\boldsymbol{K}} \boldsymbol{K}_{\mathrm{p}} \boldsymbol{K}_{\mathrm{d}} & \tilde{\boldsymbol{K}} \boldsymbol{K}_{\mathrm{p}} + \boldsymbol{K}_{\mathrm{p}} \boldsymbol{K}_{\mathrm{d}} & \boldsymbol{K}_{\mathrm{p}} \\ \tilde{\boldsymbol{K}} \boldsymbol{K}_{\mathrm{p}} + \boldsymbol{K}_{\mathrm{p}} \boldsymbol{K}_{\mathrm{d}} & \boldsymbol{K}_{\mathrm{d}}^2 + \tilde{\boldsymbol{K}} \boldsymbol{K}_{\mathrm{d}} & \boldsymbol{K}_{\mathrm{d}} \\ \boldsymbol{K}_{\mathrm{p}} & \boldsymbol{K}_{\mathrm{d}} & \boldsymbol{I} \end{bmatrix}. \tag{4.8}$$

□

Remark 21. *The matrix*

$$\boldsymbol{R}^{-1} = \alpha\tilde{\boldsymbol{K}} + \boldsymbol{I}/\gamma^2 \triangleq a\boldsymbol{I}, \quad \alpha, \tilde{\boldsymbol{K}} > \boldsymbol{0} \tag{4.9}$$

is identified using the input-to-state stability (ISS) approach [90, 131]. The procedure for finding $\boldsymbol{R}$ *is similar as in Chapter 3.*

Remark 22. *The requirement that* $\boldsymbol{Q}$ *and* $\boldsymbol{R}$ *are positive definite, delivers the following conditions for* $\tilde{\boldsymbol{K}}, \boldsymbol{K}_\mathrm{p}$, *and* $\boldsymbol{K}_\mathrm{d}$*:*

- $\tilde{\boldsymbol{K}}, \boldsymbol{K}_\mathrm{p}$, *and* $\boldsymbol{K}_\mathrm{d} \geq \boldsymbol{0}$
- $\boldsymbol{K}_\mathrm{d}^2 \geq 2\boldsymbol{K}_\mathrm{p}$

Theorem 7 shows that the adaptive control along with the matrices given by $\boldsymbol{Q}$ and $\boldsymbol{R}$ leads to a robust-optimal solution. Theorem 8 explains how to choose the controller parameters that will satisfy the maximum allowed joint errors at steady-state. For the quantitative performance analysis, we define the maximum error $|\boldsymbol{x}_\mathrm{perf}|$ that is allowed in the steady-state. $\tilde{\boldsymbol{K}}$ is assumed to be scalar for the sake of simplicity.

Theorem 8. *For the control law* (2.30) *with the conditions given in Remark 22, the performance limitation* $|\boldsymbol{x}_\mathrm{perf}|$ *is provided as [1]*

$$|\boldsymbol{x}_\mathrm{perf}| \leq \frac{2|\boldsymbol{SB}|}{\lambda_\mathrm{min}}|\boldsymbol{\phi}_\mathrm{max}|, \tag{4.10a}$$

where

$$\lambda_\mathrm{min} = \lambda_\mathrm{min}\Big(\boldsymbol{Q} + (2a - \tilde{K})\boldsymbol{SBB}^T\boldsymbol{S}\Big). \tag{4.10b}$$

Proof. Let the Lyapunov function for the GARE along with worst-case disturbance (4.4) be given by

$$V(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{Sx}. \tag{4.11}$$

Using (4.1), the first derivative of (4.11) delivers

$$\dot{V} = \frac{1}{2}\boldsymbol{x}^T(\boldsymbol{A}^T\boldsymbol{S} + \boldsymbol{SA})\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{SB}\hat{\boldsymbol{\phi}} - \boldsymbol{x}^T\boldsymbol{SB}\boldsymbol{\phi}. \tag{4.12}$$

Using (4.4) and (4.9) with scalar $\tilde{K}$ we obtain

$$\dot{V} = -\frac{1}{2}\boldsymbol{x}^T(\boldsymbol{Q} + (2a - \tilde{K})\boldsymbol{SBB}^T\boldsymbol{S})\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{SB}\boldsymbol{\phi}. \tag{4.13}$$

The first term is negative if $2a > \tilde{K}$, which is obvious from (4.9). With (4.10b), we get

$$\dot{V} \leq -\frac{1}{2}\lambda_\mathrm{min}\boldsymbol{x}^T\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{SB}\boldsymbol{\phi}. \tag{4.14}$$

Using that $\dot{V} < 0$, we obtain a performance limitation

$$|\boldsymbol{x}_\mathrm{perf}| \leq \frac{2|\boldsymbol{SB}|}{\lambda_\mathrm{min}}|\boldsymbol{\phi}_\mathrm{max}|.$$

□

To summarize, the state vector $|\boldsymbol{x}|$ will decrease until $|\boldsymbol{x}| \leq |\boldsymbol{x}_{\text{perf}}|$ because of the performance limitation explained in Theorem 8.

Theorem 8 illustrates that the adaptive control achieves stability using the GARE if the performance limitation of (4.10a) is considered. As long as $|\boldsymbol{x}_{\text{P,L}}| > 2\frac{|\boldsymbol{SB}|}{\lambda_{\min}}|\boldsymbol{\phi}_{\max}|$ the derivative of the Lyapunov function will be negative. Next, we proposed a remedy for unknown input saturation. An HJI-based control with input saturation is proposed by Abu-Khalaf [132, 133]. The input constraints are encoded as quasi-norm than enables using $\boldsymbol{\mathcal{L}}_2$ gain technique. The controller also formulates a procedure to solve the HJI equation by using a sequence of linear equations. A quadratic programming implementation of a control Lyapunov function based control is proposed in [134]. However, the input constraints must be known for both the feedback controllers.

4.1.2 Modified Desired Trajectory

The joint errors will not exceed the maximum allowed error $|\boldsymbol{x}_{\text{perf}}(2)| = |\boldsymbol{e}_{\text{perf}}|$ during steady-state if the controller parameters are selected according to (4.10a). However, (4.10a) might not be satisfied because the controller does not consider input constraints,

$$|\boldsymbol{\tau}_{\text{in}}| < |\overline{\boldsymbol{\tau}}|.$$

For the trajectory scaling, we selected the state error $\boldsymbol{e}$ as a design feature, since the proposed adaptive control offers quantitative performance analysis. The desired trajectory is modified whenever $\boldsymbol{e}$ exceeds a maximum allowed value at steady-state. $|\boldsymbol{x}_{\text{perf}}|$ is a function of the system dynamics $\boldsymbol{\phi}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ and input torques are dependent on the estimate of $\boldsymbol{\phi}$. So, to deal with the unknown saturation, the speed of the desired trajectory should be compromised whenever the error at steady-state exceeds the maximum allowed $|\boldsymbol{e}_{\text{perf}}|$. Next, we explain how to modify the desired trajectory by making the path a function of a variable θ rather than time [29, 117]. Intuitively, the trajectory is tracked slower when the motor drives cannot deliver the required current/torque. At times when the torque requirements are low, the speed of the trajectory can be increased to achieve the task in a prescribed span of time. A scalar differential equation is proposed to perform a variable speed scaling for the desired trajectory.

Trajectory scaling: Given a desired trajectory $x_d(\theta)$, where $\theta = t$ only when the torque required for generating the desired motion does not saturate $|\bar{\tau}|$. Otherwise,

$$\frac{d\theta}{dt} = \underbrace{0.5 - f(\boldsymbol{e})}_{\dot{\theta}_{\text{des}}} + g(\theta, t), \qquad \theta(t_0) = 0, \tag{4.15}$$

where

$$f(\boldsymbol{e}) = 0.5\tanh(K_1(|\boldsymbol{e}(t)|_\infty - x_{\text{off}})),$$
$$g(\theta, t) = K_2(t - \theta),$$

where $|\boldsymbol{e}(t)|_\infty = |\boldsymbol{x}(2)_{\text{perf}}|_\infty$ is a scalar representing the maximum joint error, K_1 and K_2 are positive gains and x_{off} is an offset. The function $f(\boldsymbol{e})$ can be derived from Fig. 4.2 using curve fitting technique. From the above differential equation, there are three conditions, these are

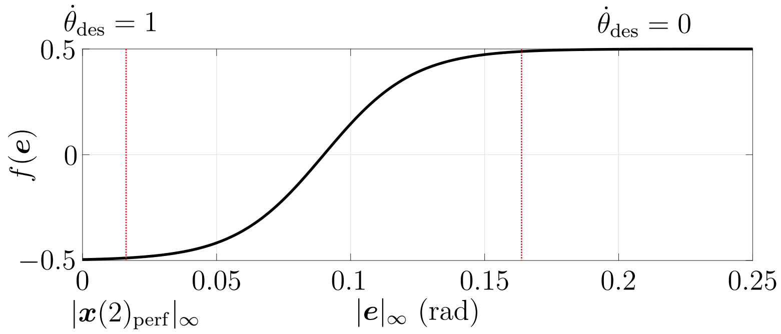


Figure 4.2: User-defined specification for $f(\boldsymbol{e})$.

1. if $|\boldsymbol{e}(t)|_\infty \leq |\boldsymbol{x}(2)_{\rm perf}|_\infty$ for all t, then $\theta = t$ because $f(\boldsymbol{e}) = -0.5$ and $g(\theta, t) = 0$
2. if $|\boldsymbol{e}(t)|_\infty > |\boldsymbol{x}(2)_{\rm perf}|_\infty$, then $\theta < t$ because $-0.5 < f(\boldsymbol{e}) \leq 0.5$, hence decreasing the speed of trajectory
3. if $|\boldsymbol{e}(t)|_\infty \leq |\boldsymbol{x}(2)_{\rm perf}|_\infty$ and $\theta(t) < t$, then

$$\theta(t) = t + \theta(t_0)e^{-K_2 t},$$

where t_0 is the time instant where $|\boldsymbol{e}(t)|_\infty > |\boldsymbol{x}(2)_{\rm perf}|_\infty \to |\boldsymbol{e}(t)|_\infty \leq |\boldsymbol{x}(2)_{\rm perf}|_\infty$.

The function $f(\boldsymbol{e})$ returns an output in the range of -0.5 to 0.5. If the error in the joint angle is above the allowed maximum error during the steady-state, θ will slow down according to the magnitude of the error. Once the trajectory is slowed down enough that the adaptive control can follow the desired trajectory, $f(\boldsymbol{e}) \to -0.5$. The term $g(\theta, t)$ always tries to reduce the error between θ and t. Depending on the value of the gain K_2, θ will move faster to approach t. The functions $f(\boldsymbol{e})$ and $g(\theta, t)$ have conflicting objectives. The former tries to slow down the trajectory and the latter tries to increase the speed of trajectory. The domination of $f(\boldsymbol{e})$ depends on the speed of the desired trajectory. If $\boldsymbol{\tau}_{\rm in} > |\bar{\tau}|$ for the most part of the trajectory tracking, then $g(\theta, t)$ will not have enough time to force $\theta \to t$, and one complete circular rotation would take more time compared to the original trajectory. Although the trajectory modification improves both the transient and steady-state performance, however, if the starting position of the end-effector is far away from the desired trajectory, there will be peaking.

The state-of-the-art methods use time-optimal control methods [117,118]. However, these techniques require knowledge about the input saturation.

Remark 23. *The function $f(\boldsymbol{e})$ can be selected differently, e.g., a ramp function or a step function. The only conditions are that $f(\boldsymbol{e}) = -0.5$ if $|e|_\infty \leq |\boldsymbol{x}_{\rm perf}(2)|_\infty$ and $f(\boldsymbol{e}) = 0.5$ if $|e|_\infty >> |\boldsymbol{x}_{\rm perf}(2)|_\infty$. Equation* (4.15) *can be customized and velocities of the joint-errors can also be included in defining the trajectory modification.*

Remark 24. *The trajectory scaling method devised in this section is not specific to the adaptive control* (2.30) *because the method only modifies the reference signal and not the controller.*

4.1.3 Improve Transient Response

The trajectory scaling explained in the previous subsection improves both the transient and steady-state performance as it will be evident in Section 4.2. But there might be a peaking effect if the initial conditions for the joint angles are not in the region of attraction, i.e., $|\boldsymbol{e}(0)|_\infty >> |\boldsymbol{x}(2)_{\text{perf}}|_\infty$.

To improve the performance in terms of joint-errors, we will first discuss the cause of the peaking effect. For the proposed adaptive controller, a high gain a for the control equation (2.30) is required to remove the mismatch as quickly as possible and thus have a system with performance similar to (2.5). To show the necessity of high controller gains, we use the Bellman equation [114]

$$\dot{V} + \frac{1}{2}(\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u} - \gamma^2\boldsymbol{d}^T\boldsymbol{d}) = 0. \tag{4.16}$$

Using (4.3), we can rewrite the Bellman equation in ISS form as

$$\dot{V} \leq -\frac{1}{2}\lambda_{Q_k}\boldsymbol{x}^T\boldsymbol{x} + \frac{1}{2}\gamma^2\boldsymbol{d}^T\boldsymbol{d} := -\gamma_1||\boldsymbol{x}||^2 + \gamma_2||\boldsymbol{d}||^2 \quad \gamma_1, \gamma_2 > 0, \tag{4.17}$$

where λ_{Q_k} is the minimum eigenvalue of the matrix $\boldsymbol{Q} + \boldsymbol{K}^T\boldsymbol{K}$. From the above equation, we can also set the maximum allowed steady-state error as explained in Section 4.1.1. Integrating the previous equation,

$$||\boldsymbol{x}||_2^2 \leq \frac{2}{\lambda_{Q_k}}V(0) + \frac{\gamma^2}{\lambda_{Q_k}}||\boldsymbol{d}||_2^2, \tag{4.18}$$

where $-\int_0^\infty \dot{V}\mathrm{d}t = -V(\infty) + V(0) \leq V(0)$. Under the assumption that, at any time the cost function is given by

$$V(0) = \frac{1}{2}\boldsymbol{x}^T(0)\boldsymbol{S}\boldsymbol{x}(0),$$

the $\mathcal{L}_2$ norm of states is given by

$$||\boldsymbol{x}||_2^2 \leq \frac{\lambda_S}{\lambda_{Q_k}}||\boldsymbol{x}(0)||^2 + \frac{\gamma^2}{\lambda_{Q_k}}||\boldsymbol{d}||_2^2, \tag{4.19}$$

where λ_S is the maximum eigenvalue of $\boldsymbol{S}$. From the above result, it is evident that the joint-error decreases as we increase the controller gains λ_{Q_k}. Also, the offset in the steady-state reduces with higher controller gain.

We have shown quantitatively that high gains are essential to achieving better performance regarding tracking error. Next, we will present the adverse effect of high gains, which is also called peaking effect in literature. High gains introduce high-frequency oscillations during the transient phase, which might make the system unstable. Some of this influence is reduced naturally because of the input saturation. Also, the motor drives have a first-order response that limits high-frequency torque. The oscillations in transient response can be reduced by using low gains [130]:

$$\dot{\boldsymbol{u}} = -\boldsymbol{K}\dot{\boldsymbol{x}}. \tag{4.20}$$

After some computation, the $\mathcal{L}_2$ norm of input $\boldsymbol{u}$ is given by

$$||\dot{\boldsymbol{u}}||_2^2 \leq \lambda_{k_{\max}}||\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\tilde{\boldsymbol{\phi}}||_2^2, \tag{4.21}$$

where $\lambda_{k_{\max}}$ is the maximum eigenvalue of $\boldsymbol{K}^T\boldsymbol{K}$. From (4.20), we conclude that the oscillations, which deteriorate the transient response dependents on the controller gain $\boldsymbol{K}$, joint-errors and the mismatch. Of course, once the system reaches steady-state and the mismatch $\tilde{\boldsymbol{\phi}} \to \mathbf{0}$, the effect of oscillations is minimal and can be ignored.

Remark 25. *To summarize, we need a low controller gain during the transient response because the initial conditions are usually unknown and can be anywhere in the workspace of the robot end-effector.*

Next, a couple of techniques for the improvement of the transient response are presented.

Procedure 1

This procedure is inspired from Na *et al.* [130], where a model-reference adaptive control is used. An extra term is added to the input torque, which depends on the filtered values of the states.

Let $\boldsymbol{x}_\mathrm{f}$ is the filtered values of states,

$$k\dot{\boldsymbol{x}}_\mathrm{f} + \boldsymbol{x}_\mathrm{f} = \boldsymbol{x} \tag{4.22}$$

and

$$\boldsymbol{E} = \boldsymbol{A}\boldsymbol{x}_\mathrm{f} - \frac{\boldsymbol{x} - \boldsymbol{x}_\mathrm{f}}{k} = -\boldsymbol{B}(\tilde{\boldsymbol{\phi}}_f). \tag{4.23}$$

By adding the term $\boldsymbol{E}$ in the input torque, the state space representation of (3.3) is given by

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\tilde{\boldsymbol{\phi}} + \boldsymbol{E}. \tag{4.24}$$

The above equation can be written as

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}(\tilde{\boldsymbol{\phi}} - \tilde{\boldsymbol{\phi}}_f). \tag{4.25}$$

Considering the new system dynamics, the oscillations in the systems can be related to

$$||\dot{\boldsymbol{u}}||_2^2 \leq \lambda_{k_{\max}}||\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}(\tilde{\boldsymbol{\phi}} - \tilde{\boldsymbol{\phi}}_f)||_2^2. \tag{4.26}$$

As it is clear that $\dot{\boldsymbol{u}}$ is reduced because of the factor $\tilde{\boldsymbol{\phi}} - \tilde{\boldsymbol{\phi}}_f$. The value of k is selected as low as possible such as to avoid oscillations.

Procedure 2

To reduce the oscillation, one method is to use small values of gains a during the transient response. For that purpose, after selecting minimum and maximum values of a, the gains for the controller can be adjusted as follows:

Let

$$a(|e|_\infty) = a_{\max} - \frac{a_{\max} - a_{\min}}{1 + \sigma_1^{(|e|_\infty - \sigma_2)}}, \tag{4.27}$$

the values for σ_1 and σ_2 can be easily identified using Fig. 4.3.

Remark 26. *The minimum value for a can be acquired from* (4.10a) *and the maximum value is evaluated using the procedure given in Section 4.1.4. The gain scheduling* (4.27)

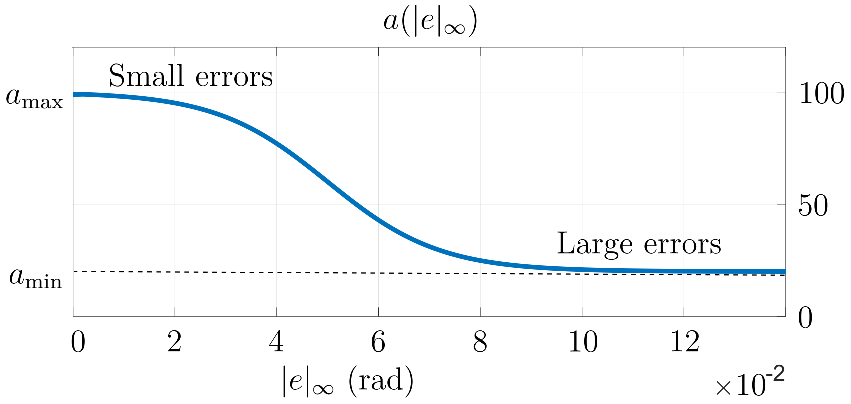


Figure 4.3: Specifications for the gain scheduling used in the results.

is not a unique function and the only objective was to get a smooth $a(|e|_\infty)$*. Furthermore, we can also use variable* $\boldsymbol{K}_\mathrm{p}$ *and* $\boldsymbol{K}_\mathrm{d}$ *as long as the conditions in Remark 22 and* (4.10a) *are satisfied.*

Theorem 9. *The system* (3.3) *and control law* (2.30) *along with the modified gain* $a(|\boldsymbol{e}|_\infty)$ *of* (4.27)*, result in a stable closed-loop system.*

Proof. The stability of the modified controller is easily verified using (4.16). The control vector $\boldsymbol{K}$ is now a function of the states, but there is an upper and lower bound because of $a_{\max}$ and $a_{\min}$. We get

$$\dot{V} = -\frac{1}{2}\boldsymbol{x}^T\Big(\boldsymbol{Q} + \boldsymbol{K}^T(\boldsymbol{e})\boldsymbol{K}(\boldsymbol{e})\Big)\boldsymbol{x} + \frac{1}{2}\gamma^2\boldsymbol{d}^T\boldsymbol{d}. \tag{4.28}$$

Because $\boldsymbol{K}(\boldsymbol{e})$ is bounded, $\dot{V}$ is always negative with respect to the states if condition (4.10a) is satisfied for all $a(|e|_\infty)$ [135]. The characteristic equation for the feedback system (3.3) represented in the state-space form (4.6) along with the inputs (4.7) and (4.27) is

$$(\boldsymbol{s} + a(|e|_\infty)\boldsymbol{I})(\boldsymbol{s}^2 + \boldsymbol{K}_\mathrm{d}\boldsymbol{s} + \boldsymbol{K}_\mathrm{p}) = \boldsymbol{0}. \tag{4.29}$$

The poles of the closed-loop system are always negative as long as $a(|e|_\infty)$ is positive.

Another way to prove the stability is by considering $\tilde{\boldsymbol{K}}$ in (4.5). The matrix $\boldsymbol{Q}$ will always be positive definite if $\tilde{\boldsymbol{K}} > 0$. The relation between $\tilde{\boldsymbol{K}}$ and a is given in Remark 21. □

The controller gain $\boldsymbol{K}$ becomes a constant as $\boldsymbol{x}$ approaches to zero. More details about the state-dependent controller gains are available in references [136, 137].

4.1.4 Easy Tuning for the Controller

This subsection explains a systematic and intuitive strategy to find the parameters for the controller (2.30).

1. Evaluate the minimum values for the controller parameters using (4.10a) that satisfy the condition for the allowed maximum joint-error at steady-state.

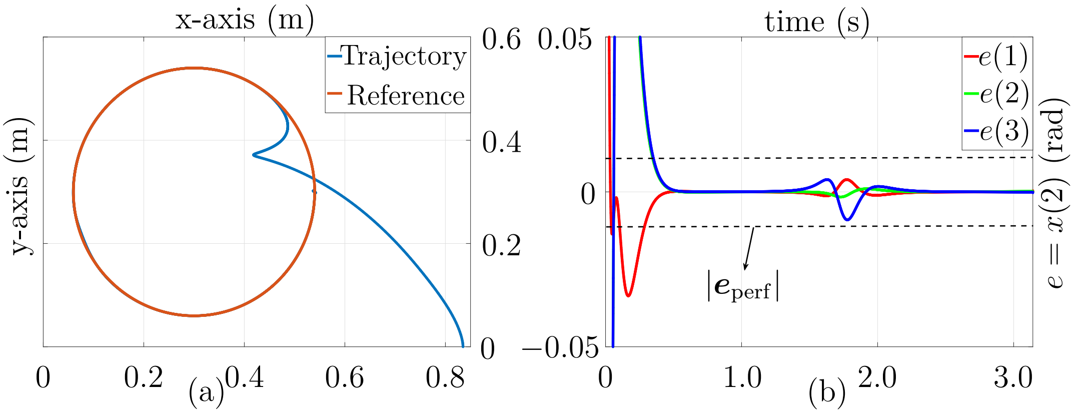


Figure 4.4: Simulation: Tracking without input constraints. (a) Position of the end-effector in Cartesian coordinates, (b) joint errors.

2. Find the values of $\boldsymbol{K}_\mathrm{p}$ and $\boldsymbol{K}_\mathrm{d}$ satisfying a performance criterion such as settling time, rise time, etc.

3. Choose a such that the value of the pole of [3]

$$\dot{\tilde{\boldsymbol{\phi}}} + a\tilde{\boldsymbol{\phi}} = \mathbf{0}$$

is 3-5 times higher than the poles of (2.5).

The intuition for the above procedure is that the PD gains will perform in a deterministic way by choosing a proper rise time, settling time, etc. The PD control will perform linearly only if the mismatch is removed. For that reason, the pole location for the convergence of mismatch to zero (a) must be faster than the PD gains ($\boldsymbol{K}_\mathrm{p}$ and $\boldsymbol{K}_\mathrm{d}$). Finally, to compensate for the peaking effect during the transient phase, a is selected as a variable with upper and lower bounds, see Section 4.1.2.

4.2 Results

A 3-degree-of-freedom (DoF) planar robot is used to validate the proposed controller in both simulations and experiments. For experiments, 10 Amp Maxon drives along with 150 Watts motors are used (Fig. 3.3). The desired end-effector trajectory $\boldsymbol{q}_d$ is a circle in the first quadrant of the x-y plane with radius 0.24 m and center $0.3\hat{x}$ m + $0.3\hat{y}$ m and the corresponding joint-space trajectory is shown in Fig. 4.5. The values for the controller gains are $\boldsymbol{K}_\mathrm{p} = 40\boldsymbol{I}$, $\boldsymbol{K}_\mathrm{d} = 400\boldsymbol{I}$, $a = 80$ and $|\boldsymbol{e}_{\mathrm{perf}}|_\infty = 0.01$ radians. The end-effector starts at $\hat{x}$ 0.83 m. The dynamical equations of the 3-DoF robot manipulator used in experiments are calculated from the real robot, see Table 4.1 and Appendix C. Since the adaptive control is model-free, the dynamics are only used for simulations.

4.2.1 Simulation Results

The simulation results are divided into three parts. The objective of the first result is to show that using (4.10a), the joint-error at steady-state is always less than $|\boldsymbol{e}_{\mathrm{perf}}|_\infty = 0.01$ rad/s.

Table 4.1: 3-DoF planar robot dynamical equations.

Var.	**Equivalent**
$M(1,1)$	$1.04 + 0.08\cos(q_2 + q_3) + 0.34\cos(q_2) + 0.05\cos(q_3)$
$M(1,2)$	$0.43 + 0.04\cos(q_2 + q_3) + 0.17\cos(q_2) + 0.05\cos(q_3) = M(2,1)$
$M(1,3)$	$0.17 + 0.04\cos(q_2 + q_3) + 0.023\cos(q_3) = M(3,1)$
$M(2,2)$	$0.43 + 0.05\cos(q_3)$
$M(2,3)$	$0.17 + 0.03\cos(q_3) = M(3,2)$
$M(3,3)$	0.17
$N(1,1)$	$-0.17\sin(q_2)\dot{q}_2^2 - 0.04\sin(q_2 + q_3)\dot{q}_2^2 - 0.03\sin(q_3)\dot{q}_3^2 - 0.04\sin(q_2 + q_3)\dot{q}_3^2$ $-0.34\sin(q_2)\dot{q}_1\dot{q}_2 - 0.08\sin(q_2 + q_3)\dot{q}_1\dot{q}_2 - 0.05\sin(q_3)\dot{q}_1\dot{q}_3$ $-0.08\sin(q_2 + q_3)\dot{q}_1\dot{q}_3 - 0.05\sin(q_3)\dot{q}_2\dot{q}_3 - 0.08\sin(q_2 + q_3)\dot{q}_2\dot{q}_3$
$N(2,1)$	$0.17\sin(q_2)\dot{q}_1^2 + 0.04\sin(q_2 + q_3)\dot{q}_1^2 - 0.03\sin(q_3)\dot{q}_3^2 - 0.05\sin(q_3)\dot{q}_1\dot{q}_3$ $-0.05\sin(q_3)\dot{q}_2\dot{q}_3$
$N(3,1)$	$0.028\sin(q_3)\dot{q}_1^2 + 0.03\sin(q_3)\dot{q}_2^2 + 0.04\sin(q_2 + q_3)\dot{q}_1^2 + 0.05\sin(q_3)\dot{q}_1\dot{q}_2$
$F(i,j)$	$2.6 \times 10^4 \;\; \forall \; i = j$
$F(i,j)$	$0 \;\; \forall \; i \neq j$

Fig. 4.4 shows that the error at steady-state is much lower than 0.01 rad/s for all time during steady-state. For one complete rotation of the end-effector, the desired trajectory requires 3.14 seconds. Here the assumption is that the torque requirements for the desired trajectory are lower than the saturation.

The second set of simulations shows, how the torque saturation affects the trajectory tracking. We have increased the speed of the desired path until input saturation is reached, and as a result, the manipulator could not follow the trajectory for most of the desired trajectory as shown in Fig. 4.6-a. The desired trajectory span was 1.62 seconds for one complete rotation. After applying the path scaling, the trajectory slowed down in the region where the input torque requirement is higher than the saturation Fig. 4.6-b. The modified desired trajectory makes sure that the error is much lower compared to the previous results. The parameters for the trajectory modification (4.15) are evaluated using Fig. 4.2.

Fig. 4.7 shows that the parameter θ, which ideally is equal to time, slowed down during the first 1000 ms because of the function $f(\boldsymbol{e})$. In the last half of the desired trajectory, the function $g(\theta, t)$ reduces the difference between θ and t. At the end of one complete circle, the total time is equal for both the trajectories in Fig. 4.6, as evident from Fig. 4.7.

The third result provides the improvement of the transient response by using the two techniques of Section 4.1.3. Fig. 4.8 shows the effect of variable gain to improve the transient response. The maximum and minimum values for the parameter $a = 20 - 80$ as in Fig. 4.3.

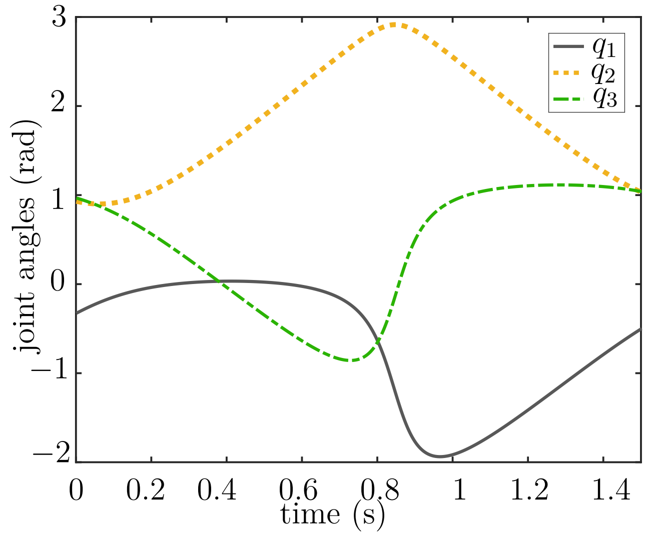
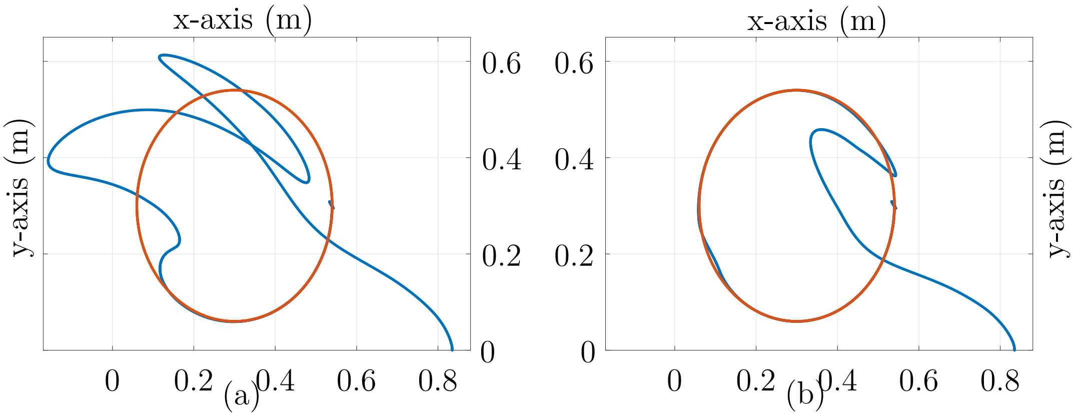


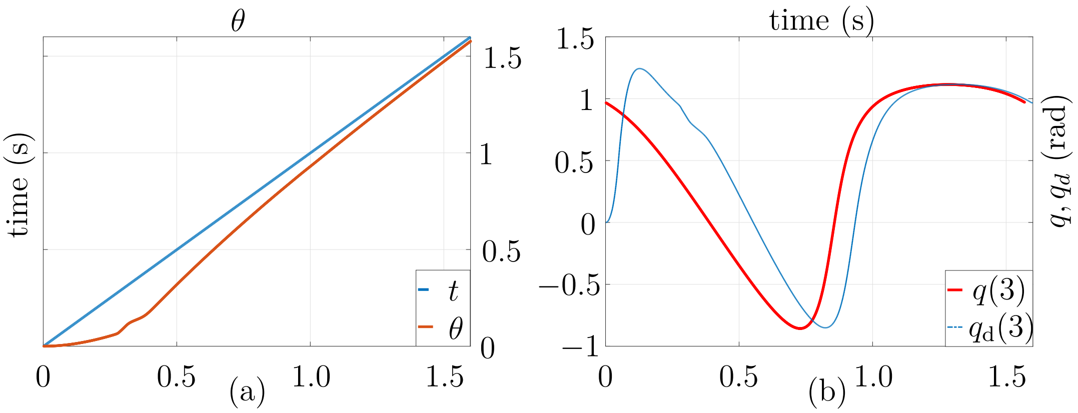
Figure 4.5: Joint angles for the desired circular trajectory.



(a)

(b)

Figure 4.6: Simulation: Trajectory in the presence of input constraints. (a) Without scaling, (b) with scaling.



(a)

(b)

Figure 4.7: Simulation: Comparison of time t and θ. (a) t and θ, (b) q and q_d.

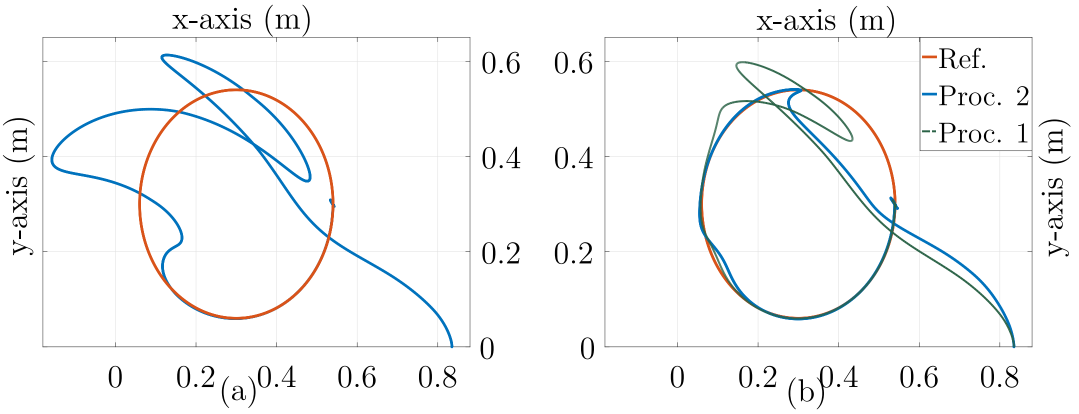


Figure 4.8: Simulation: (a) Without gain adjustment. (b) Gain adjustment with $a = 20 - 80$ using (4.27).

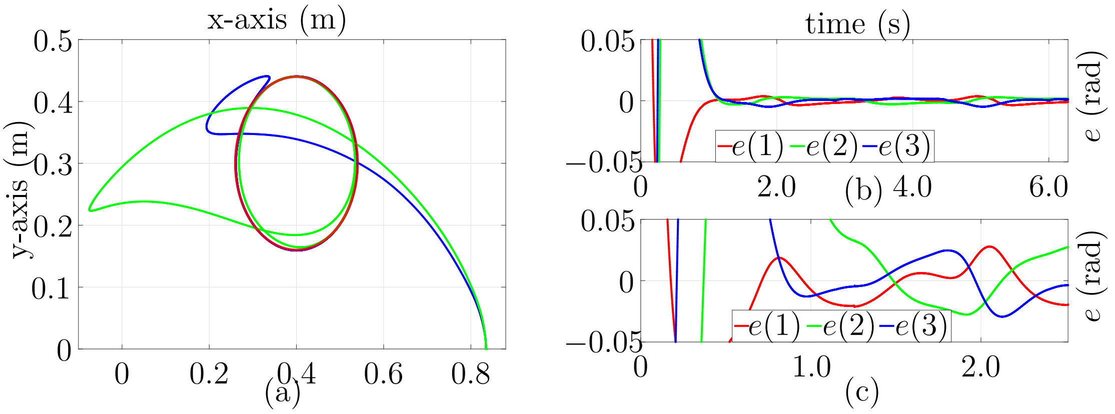


Figure 4.9: Experiment: (a) Trajectory of the end-effector, (b) shows that the performance limitation is satisfied by the blue trajectory in (a). (c) represents the error in the joint angles for the green trajectory in (a).

4.2.2 Experimental Results

The adaptive control includes all the uncertainties and disturbance in the vector $\boldsymbol{\phi}$, and as long as $\boldsymbol{\phi}$ is bounded, the controller will force the system to follow the desired trajectory with an accuracy of 0.01 radians. Fig. 4.9 shows the results of two experiments with the same desired trajectory but with different speeds. The first experiment, Fig. 4.9 (a-blue and b) satisfies the $|\boldsymbol{e}_{\text{perf}}|_\infty = 0.01$ bound because the torques for the desired trajectory are less than the saturation. This condition is not satisfied if the torque saturation is lower than the required torque as evident in Fig. 4.9 (a-green and c). The desired trajectory took 3.1/1.2 seconds for one complete rotation for blue/green trajectories. To make sure that the end-effector follows the orbit of the desired trajectory, the second experiment is improved by modifying the desired trajectory. Fig. 4.10 (a-blue and b) shows that the performance limitations are now satisfied but the total time required for the circular trajectory is higher than the original time Fig. 4.9 (c). The transient response is further improved by using the variable gain technique from Section 4.1 as shown in Fig. 4.9 (a-green).

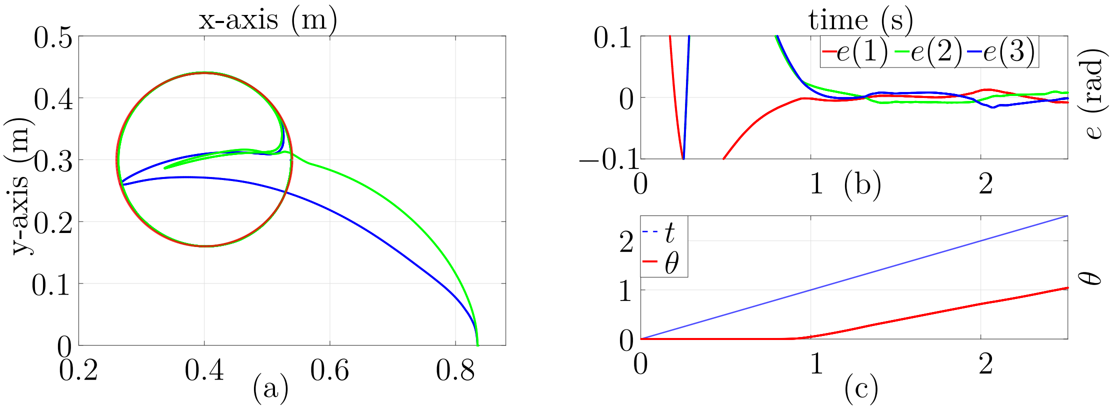


Figure 4.10: Experiments: (a) The desired trajectory took 1.2 seconds for one complete rotation for both trajectories. (b) shows that the performance limitation is satisfied by the blue trajectory in (a). (c) represents the difference between θ (red) and t (blue).

4.3 Summary

Robust optimality of the first-order adaptive controller is evaluated using $\mathcal{H}_\infty$ technique. The stability is proved by using the inverse-optimal method, where the matrices for the optimized cost function are identified using the adaptive control. To compensate for the unknown torque saturation, a solution is proposed that modifies the speed of the desired trajectory such that the error in the joint angles remains within certain bounds at steady-state. For the transient response improvement, two methods are discussed that can be used to improve the high-frequency noise and peaking effect during the transient phase. And lastly, an intuitive procedure for the identification of the minimum allowed controller parameters is formalized. This chapter is aimed at the practical implementation of the proposed controller, we improved the following:

- **Consideration of unknown input saturation:** The previous chapter assumed no input saturation, which affects both steady-state and transient-state performance. In this chapter, a systematic procedure is proposed to incorporate the unknown input saturation in the feedback loop and reduces its effect in both transient and steady state error. Usually, in state-of-the-art controllers, the assumption is that the input saturation is known as discussed in the introduction section of the manuscript.

- **Improving transient-response:** The original controller also has the peaking effect during the transient response, since, adaptive controllers require some time to estimate the system dynamics. This peaking effect is not acceptable in many cases because of physical and safety constraints on robot manipulators. Two procedures are proposed in this chapter to reduce the peaking.

- **Simplified controller:** Chapter 3 proposed a second-order controller with robust optimality proof. In theory, a higher order controller will give better approximation of the system. Because of input saturation and the time that the motor drives take to produce high currents during transient phase is limited, which makes the higher-order proposed adaptive controller noisy at high speeds. To avoid the noise, this chapter proposed that a first-order controller can also perform better provided that the controller gains are selected properly. The robust optimality of the first-order controller is also proved, while Chapter 3 only has a proof for the second-order controller.

- **Easy-tuning:** The proposed adaptive controller can be tuned easily because of very few parameters. The first-order adaptive controller is a model-free control method with only three tuning parameters. One parameter is used to remove the mismatch between the estimated and the real system dynamics. The other two parameters are used to achieve the desired response. An intuitive approach is used to find the control parameters.

Open problem

The proposed adaptive controller in its current form is only applicable on a fully actuated system. Further modification is required to implement the controller on an under-actuated system.

5

Conclusion and Future Directions

A reliable control technique is important for robotic manipulators, especially with the increasing demand for robots. With the advent of powerful computers and latest control techniques, the use of robots has extended from industries to different fields of life. In the last few decades, many control methods had been proposed but there is still much improvements required in the performance of those control schemes. The major challenges are highly coupled nonlinear dynamics, unmodeled dynamics, parameters uncertainties, variable load, input saturation and complex friction model. In this dissertation, a robust-adaptive control method is proposed with guaranteed quantitative performance and easy tuning. Implementation issues, such as input saturation and peaking are also addressed to enhance the performance of the controller.

5.1 Discussion

For low operational speed, decoupled control techniques can be used to independently control the joints of a robot by considering the coupling effect as a disturbance. However, at high speed, the influence of nonlinear coupling among the joints is inevitable and must be included in control design [50]. Similarly, a proportional-derivative (PD) controller can be applied on a planar robot that is not influenced by gravity. The stability proofs of such controllers are available in literature [8]. In addition, for a system that involves gravity, a PD control with gravity compensation gives a stable result. The derivative of a Lyapunov function for such a controller will be negative as long as the joint velocity is not zero. Any mismatch in the gravity terms will affect the system state errors and performance. Inverse-dynamics control, also known as computed-torque, takes advantage of the full system's knowledge. It is based on feedback linearization, where all the nonlinear terms in the system are canceled using a feedback loop. A PD control is then used to achieve the desired performance. However, computed-torque is a model-based control method and its performance depends on the estimated system parameters.

In robot manipulators, the uncertainties in the inertia and the Coriolis matrices and the unmodeled friction models limit the performance of computed-torque in terms of joint errors. Adaptive controllers are extensively used in such situations, where the system dynamics have uncertainties and disturbances. The aim of an adaptive or robust controller is to maintain the desired performance in the presence of uncertainties, unmodeled dynamics and external or internal disturbances.

Robust-Adaptive Control

A model-free robust-adaptive control is proposed for robot manipulators. The adaptive control considers the system dynamics and disturbances as an unknown vector and a feedback loop is used to estimate them without any system information. A two-player zero-sum approach is utilized by using the game-algebraic-Riccati equation to prove the optimality of the controller. Input-to-state-stability is used for finding the appropriate controller parameters values. An advantage of the proposed method is the quantitative performance analysis that is achieved by using the robust optimal approach. This means that the state errors will not exceed a predefined maximum value during the steady-state, thus guaranteeing a certain amount of accuracy. The main features of the proposed controller are

- **Model-free:** Finding parameters of a robot dynamics is a difficult task because of uncertainties, unmodeled dynamics and load variation. Furthermore, for a higher-order manipulator, the number of system parameters is large, thus prone to large mismatch. The proposed adaptive controller is model-free and the system dynamics are estimated online using simple differential equations. The controller considers the system dynamics, internal and external disturbance as part of an unknown vector, and using a simple differential equation, that vector is estimated. A feedback linearization is then utilized to cancel the nonlinearities and eventually results in a linear system with a proportional-derivative controller in the second feedback loop.

- **Optimal:** Inverse-optimal control using the $\mathcal{H}_\infty$ method is used to show the robust optimality of the controller. Unlike conventional optimal control, where the controller is designed using a cost function, the proposed controller uses the inverse approach, *"finding the cost function using the given controller."*

- **Robust:** Two-player zero-sum approach is used to prove the robustness of the controller. It has been shown that any bounded disturbance will not affect the quantitative performance of the feedback control system.

- **Quantitative steady-state performance:** The steady-state behavior of the system satisfies predefined performance criterion. The state errors is guaranteed to ensure a used-defined specification of maximum errors during the steady-state. $\mathcal{H}_\infty$ method is used to identify the minimum values of the controller parameters such that the joint errors are bounded at steady-state. For example, if the maximum allowed steady-state error is 10 milli-radians, the controller will make sure to keep the joint angles under the predefined bounds.

- **Easy tuning:** There are only three tuning parameters for the first-order and four parameters for a second-order proposed adaptive controller. An intuitive approach is proposed to find proper control values to satisfy the performance criterion.

- **Intelligent controller:** The proposed adaptive controller is also shown as an intelligent controller, similar to the intelligent PID controller [83]. An advantage of such a controller is the universal control parameters for any system variations or disturbances.

Improvements

We introduced a robust adaptive controller in Chapter 2 and 3, which addresses unmodeled dynamics, system uncertainties, optimality, easy tuning and quantitative performance analysis in terms of steady-state error. The state-of-the-art controller do not consider all the above features in the controller design. To further improve the performance of the proposed controller, some important implementation issues are also solved by augmenting the controller. The two major issues in the control system, which is often ignored in literature are the input saturation and the peaking effect.

Input saturation is a highly nonlinear phenomenon and in robotics, it is not easy to find the saturation values. These constraints are time-varying and depend on many parameters, such as maximum voltage and current rating of the motors and the back electromotive force. Since the proposed control method guarantees a predefined accuracy, which means that the steady-state error is always less than a maximum allowed value, the error will exceed the limit only if the desired input is not achievable because of input saturation. A feedback loop is used to modify the input trajectory by scaling it down such that the maximum allowed state errors are below the defined threshold.

In theory, high gains are used in adaptive control for better reference tracking. The drawback is that the transient response has high-frequency noise because of high gains and an overshoot, which is also called peaking effect [113]. Nonlinearities and large input torques during the transient phase is the reason for the peaking effect. In robotics, this phenomenon is usually avoided because of the safety precautions. The high-frequency noise up to some extent is suppressed by the natural property of motor drives, which act as a low-pass filter. However, some unmodeled dynamics can be excited because of high gains and it can cause instability in the system. A trade-off between high performance and noise suppression can be one of a solution for the above-mentioned problem by selected mid-range gains for the controller. In adaptive control, peaking is a major issue, which occurs during the transient phase. Peaking in general is caused by the difference between initial states and desired states. For robots, this phenomenon is often not acceptable because of safety reasons for both humans and robots. Two techniques are used to suppress the peaking effect. The first technique is a variable controller gain approach, where a low-gain controller is used and once the transient response is over, the nominal gains are used. The second method uses an extra term in the controller to compensate for the peaking effect. Following are the improvements in the proposed controller:

- **Qualitative transient-state performance:** The major issues during transient-state are peaking and input saturation. The proposed adaptive controller is slightly modified to improve the transient-state. However, this improvement cannot be quantized because of random initial states and unknown input saturation.

- **Include input saturation:** Most of the state-of-the-art control schemes do not consider the input saturation or assume that the input saturation is known. However, finding the input saturation is not possible because of uncertainties, actuators unmodeled dynamics and back electromotive force in the motors. In this dissertation, the unknown time-varying input saturation is incorporated in control design using the quantitative performance analysis. When the state errors exceed the maximum allowed error at steady-state, the trajectory is scaled down to give more time for the end-effector to follow the desired trajectory.

System Identification

The dynamics of robot manipulators can be represented in regressor form, which is also called linear-in-parameters systems. Although the proposed adaptive controller does not require system dynamics, it is still useful to improve the transient response by incorporating the system information during initial states. The input-output data from the robotic system with the proposed adaptive controller is used to find the parameters of the system. For this purpose, the system dynamics represented in the Euler-Lagrange format is first transformed into a known regressor matrix and an unknown parameter vector. Since the new system is linear in parameters, the least-squares technique is used to identify the system dynamics parameters. The parameters of a two- and three-degree-of-freedom robots are identified using this method.

5.2 Outlook

The proposed adaptive controller introduced in this dissertation is applicable only on fully actuated systems. An under-actuated system has fewer actuators than the degree-of-freedom to be controlled. Some systems intrinsically have under-actuation, such as legged robots and swimming/flying robots. Others are designed with under-actuation to reduce the cost or some practical purposes e.g., satellites with two thrusters, flexible-joints manipulators, etc. In the last couple of decades, many control schemes have been developed for under-actuated robots. Using the proposed control method for under-actuated robotic systems is an open problem.

A

Mathematical Background

The essential mathematical tools that are used in the proposed controller are explained in this appendix. The advantage of the $\mathcal{H}_\infty$ technique and ISS is the incorporation of the quantitative performance and optimality in the control design.

A1 $\mathcal{H}_\infty$ Control

Consider a nonlinear control affine system

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{k}(\boldsymbol{x})\boldsymbol{d}, \tag{A.1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^n$, $\boldsymbol{u} \in \mathbb{R}^r$, $\boldsymbol{g}(\boldsymbol{x}) \in \mathbb{R}^{n\times r}$, $\boldsymbol{d} \in \mathbb{R}^p$ and $\boldsymbol{k}(\boldsymbol{x}) \in \mathbb{R}^{n\times p}$ are the state, transition, input and disturbance vectors/functions, respectively. We can shift the state variables so that $\boldsymbol{f}(\boldsymbol{0}) = \boldsymbol{0}$, such that $\boldsymbol{x} = \boldsymbol{0}$ is an equilibrium point.

According to two-player zero-sum games [129], one can get optimal control values for $\boldsymbol{u}$ and $\boldsymbol{d}$ that will minimize the cost function mentioned below. A cost function associated with this system can be expressed as [11, 114]

$$\begin{aligned} V(\boldsymbol{x}) &= \min_{\boldsymbol{u}} \max_{\boldsymbol{d}} \int_\tau^\infty l(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{d})\mathrm{d}t \\ &= \min_{\boldsymbol{u}} \max_{\boldsymbol{d}} \int_\tau^\infty (\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u} - \gamma^2\boldsymbol{d}^T\boldsymbol{d})\mathrm{d}t, \end{aligned} \tag{A.2}$$

where $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{R} \in \mathbb{R}^{r\times r}$ are positive definite matrices.

For a system to be robust, the $\mathcal{L}_2$-gain from disturbance to the performance index is less than or equal to some positive constant γ^2:

$$\frac{\int_\tau^\infty (\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u})\mathrm{d}t}{\int_\tau^\infty ||\boldsymbol{d}||^2\mathrm{d}t} \leq \gamma^2. \tag{A.3}$$

There also exists a smallest value $\gamma^* > 0$ such that (A.3) is satisfied for all $\gamma > \gamma^*$.

The primary objective is to find values of $\boldsymbol{u}$ and $\boldsymbol{d}$, that will respectively, minimize and maximize the cost function. Taking the derivative of the cost function, we get the Bellman equation

$$\dot{V} + l(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{d}(t)) = 0,$$

$$\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u} - \gamma^2\boldsymbol{d}^T\boldsymbol{d} + (\frac{\partial V}{\partial \boldsymbol{x}})^T(\boldsymbol{f} + \boldsymbol{g}\boldsymbol{u} + \boldsymbol{k}\boldsymbol{d}) = 0, \tag{A.4}$$

which is also written as

$$H(\boldsymbol{x}, \frac{\partial V}{\partial \boldsymbol{x}}, \boldsymbol{u}, \boldsymbol{d}) = 0. \tag{A.5}$$

To find a saddle of the Hamiltonian equation (A.5), we set

$$\frac{\partial H}{\partial \boldsymbol{u}} = 0, \quad \frac{\partial H}{\partial \boldsymbol{d}} = 0.$$

We get the equations for $\boldsymbol{u}(t)$ and $\boldsymbol{d}(t)$, which is given by

$$\boldsymbol{u}^* = -\frac{\boldsymbol{R}^{-1}}{2}\boldsymbol{g}^T\nabla V^*, \quad \boldsymbol{d}^* = \frac{1}{2\gamma^2}\boldsymbol{k}^T\nabla V^*, \tag{A.6}$$

the asterisk represents optimal values and $\nabla V = \partial V/\partial \boldsymbol{x}$. The cost function associated with these optimized inputs yields the HJI equation, which is given by

$$\boldsymbol{Q} + \nabla V^T\boldsymbol{f}(\boldsymbol{x}) - \frac{1}{4}\nabla V^T(\boldsymbol{x})\boldsymbol{g}(\boldsymbol{x})\boldsymbol{R}^{-1}\boldsymbol{g}^T(\boldsymbol{x})\nabla V(\boldsymbol{x}) + \frac{1}{4\gamma^2}\nabla V^T(\boldsymbol{x})\boldsymbol{k}\boldsymbol{k}^T\nabla V(\boldsymbol{x}) = \boldsymbol{0}.$$

Since disturbance cannot be controlled, therefore, the objective of the $\mathcal{L}_2$-gain problem is to find the control law $\boldsymbol{u}$ such that, for a Lipschitz system, and all $\boldsymbol{d}(t) \in \mathcal{L}_2[0,\infty)$, the $\mathcal{L}_2$-gain of (A.3) is satisfied.

A2 Game-Algebraic Riccati Equation (GARE)

Solving the HJI equation for a nonlinear system is a tedious job. Fortunately, if a robot manipulator is expressed in the following form:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{D}\boldsymbol{d}, \tag{A.7}$$

which is the linear case of (A.1), the HJI equation associated with the $\mathcal{L}_2$-gain (A.3) reduces to the GARE

$$\boldsymbol{A}^T\boldsymbol{S} + \boldsymbol{S}\boldsymbol{A} - \boldsymbol{S}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{S} + 1/\gamma^2\boldsymbol{S}\boldsymbol{D}\boldsymbol{D}^T\boldsymbol{S} + \boldsymbol{Q} = \boldsymbol{0}. \tag{A.8}$$

Calculating $\boldsymbol{S}$ from (A.8), leads the equation (A.6) to reduce to a static state feedback form

$$\boldsymbol{u}^* = -\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{S}\boldsymbol{x} \triangleq -\boldsymbol{K}\boldsymbol{x}. \tag{A.9}$$

Remark 27. *It can be shown that the input choice* (A.9) *yields an asymptotically stable system* (A.7)*, provided that* $\boldsymbol{d} \in \mathcal{L}_2[0,\infty)$ *[114]. The disturbance input given in* (A.6) *is called the worst-case disturbance.*

A3 Input-to-State-Stability

For a linear system, the bounded-input-bounded-output stability is not affected by inputs or disturbances provided that they are bounded. However, for nonlinear systems, an internally stable system can become unstable if certain inputs are applied. Because of that, *Sontag* proposed a definition called input-to-state-stability that also considers inputs to find the stability of a nonlinear system [90,91,131,138,139]. Since disturbances and model uncertainties can be considered as part of input disturbance, ISS can also be used to evaluate the stability and robustness of a system.

For a general nonlinear control system

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{d}), \tag{A.10}$$

where $\boldsymbol{x}$ and $\boldsymbol{d}$ are the state and disturbance vectors, respectively, the ISS stability is defined as

$$|\boldsymbol{x}(t)| \leq \beta(|\boldsymbol{x}(0)|, t) + \gamma(||\boldsymbol{d}||_\infty), \quad \forall t \geq 0, \tag{A.11}$$

where $\beta \in \mathcal{KL}$, $\gamma \in \mathcal{K}_\infty$ [131] and $\boldsymbol{x}(0)$ is the initial state. A class $\mathcal{K}_\infty$ is a function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, which is continuous, unbounded, increasing and satisfies $\alpha(0) = 0$. A class $\mathcal{KL}$ is a function $\beta : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, such that $\beta(., t) \in \mathcal{K}_\infty$ for all t and $\beta(r, t) \searrow 0$ as $t \to \infty$, where r is a constant.

In case of a linear system, we get

$$\beta(t) = |\boldsymbol{x}(0)| \; ||\boldsymbol{e}^{\boldsymbol{A}t}||, \quad \gamma = ||\boldsymbol{B}|| \int_0^\infty ||\boldsymbol{e}^{\boldsymbol{A}t}|| \mathrm{d}\tau. \tag{A.12}$$

A result that will later be used to find the proposed control law is given in [131]: "*A system is ISS if and only if it admits a smooth ISS-Lyapunov function.*" This suggests that there exists a positive definite function $V(\boldsymbol{x}) \in \mathbb{R}$ and $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$ such that

$$\alpha_1(|\boldsymbol{x}|) \leq V(\boldsymbol{x}) \leq \alpha_2(|\boldsymbol{x}|). \tag{A.13}$$

ISS stability of (A.10) can then be concluded from

$$\dot{V}(\boldsymbol{x}) \leq -\gamma_1(|\boldsymbol{x}|) + \gamma_2(|\boldsymbol{d}|), \;\; \forall \; \boldsymbol{x}, \boldsymbol{d}, \tag{A.14}$$

where $\gamma_{(.)} \in \mathcal{K}_\infty$. Further, let there exist $V(\boldsymbol{x})$ such that the following condition is satisfied for all $\boldsymbol{x}$ and $\boldsymbol{u}$ [92]:

$$|\boldsymbol{x}| \geq \rho(|\boldsymbol{d}|) \Rightarrow \dot{V}(\boldsymbol{x}) \leq -\gamma_3(|\boldsymbol{x}|), \tag{A.15}$$

where γ_3 and $\rho \in \mathcal{K}_\infty$. In this case, not only is the system ISS but also asymptotically stable. The last result is important, since it is used to find the controller parameters that satisfy the predefined performance specifications.

B

Proofs and Codes

B1 Regressor form

For a 2-DoF manipulator, consider the white-box model

$$\begin{bmatrix} m_{11}+m_{11}^{'}\cos q_2 & m_{12}+m_{12}^{'}\cos q_2 \\ m_{21}+m_{21}^{'}\cos q_2 & m_{22} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} -n_{11}\dot{q}_2^2\sin q_2 - n_{12}\dot{q}_1\dot{q}_2\sin q_2 \\ n_{21}\dot{q}_1^2\sin q_2 \end{bmatrix}.$$

The above model can be represented in the regressor form as

$$\begin{bmatrix} \ddot{q}_1 & \ddot{q}_1\cos q_2 & \ddot{q}_2 & \ddot{q}_2\cos q_2 & \dot{q}_2^2\sin q_2 & \dot{q}_1\dot{q}_2\sin q_2 & 0 & 0 \\ 0 & 0 & \ddot{q}_1 & \ddot{q}_1\cos q_2 & 0 & 0 & \ddot{q}_2 & \dot{q}_1^2\sin q_2 \end{bmatrix} \times$$

$$\begin{bmatrix} m_{11} & m_{11}^{'} & m_{12} & m_{12}^{'} & -n_{11} & -n_{12} & m_{22} & n_{21} \end{bmatrix}^{\mathrm{T}}$$

The regressor matrix is not unique and can be optimized by removing any dependent variables. Constructing the regressor matrix for high DoF manipulator is difficult because of large number of system parameters.

B2 Proof of $\dot{V}(\boldsymbol{x}) < 0$

There can be many inverse-dynamics solution for the control law suggested in Chapter 2. The matrices $\boldsymbol{Q}_{(.)}$ can be selected as diagonal with positive real values on its diagonal to achieve the positive definiteness. Taking advantage of this fact, it can be proved that $\dot{V}(\bar{\boldsymbol{x}}, \boldsymbol{z}) < 0$ in (2.29), To avoid mathematical complications, let

$$\boldsymbol{Q}_1 = diag(\boldsymbol{q}_{\mathrm{ii}}),\ \forall\ \boldsymbol{q}_{\mathrm{ii}} > 0,$$

$$\boldsymbol{Q}_2 = diag(\boldsymbol{r}_{\mathrm{ii}}),\ \forall\ \boldsymbol{r}_{\mathrm{ii}} > 0.$$

are matrices of proper dimensions, then equation (2.29) takes the form

$$\dot{V} = -\boldsymbol{q}_{11}||\boldsymbol{e}||^2 - \boldsymbol{q}_{22}||\dot{\boldsymbol{e}}||^2 - \boldsymbol{r}_{11}||\tilde{\boldsymbol{\phi}}||^2 + 2\boldsymbol{p}_{21}\tilde{\boldsymbol{\phi}}^T\boldsymbol{e} + 2\boldsymbol{p}_{22}\tilde{\boldsymbol{\phi}}^T\dot{\boldsymbol{e}} - l_1, \tag{B.1}$$

where $l_1 > 0$ consists of all derivatives of $\tilde{\boldsymbol{\phi}}$. We assumed $\boldsymbol{p}_{(.)}$, $\boldsymbol{q}_{(.)}$, $\boldsymbol{K}_{\mathrm{d}}$ and $\boldsymbol{K}_{\mathrm{p}}$ to be scalar to make the proof simple. If $m > 1$, $\boldsymbol{p}_{21}$ and $\boldsymbol{p}_{22}$ are the elements of matrix $\boldsymbol{P}_1$ and can be evaluated from $\bar{\boldsymbol{A}}^T\boldsymbol{P}_1 + \boldsymbol{P}_1\bar{\boldsymbol{A}} = \boldsymbol{Q}_1$:

$$p_{21} = \frac{q_{11}}{2K_\mathrm{p}}$$

$$p_{22} = \frac{q_{22}}{2K_\mathrm{d}} + \frac{q_{11}}{2K_\mathrm{p}K_\mathrm{d}}.$$

Completing squares for (B.1), we get

$$\dot{V} \leq -||\sqrt{q_{11}}\boldsymbol{e} - \sqrt{\alpha}\tilde{\boldsymbol{\phi}}||^2 - ||\sqrt{q_{22}}\dot{\boldsymbol{e}} - \sqrt{\beta}\tilde{\boldsymbol{\phi}}||^2 - l_2 - l_1, \tag{B.2}$$

where l_2 consists of residuals and is positive if the following conditions are satisfied:

$$\alpha = \frac{q_{11}}{4K_\mathrm{p}^2},$$

$$\beta = \frac{1}{4q_{22}}\Big(\frac{q_{22}}{K_\mathrm{d}} + \frac{q_{11}}{K_\mathrm{p}K_\mathrm{d}}\Big)^2,$$

where

$$|\boldsymbol{r}_{11}| \geq \alpha + \beta.$$

Since matrices $\boldsymbol{Q_1}$ and $\boldsymbol{Q_2}$ are independent of each other, the above conditions can be easily satisfied. Intuitively it is important that the gains of $\tilde{\boldsymbol{\phi}}$, $\boldsymbol{r}_\mathrm{ii}$ should be greater than some fraction of the tracking gains $\boldsymbol{q}_\mathrm{ii}$. Otherwise the controller is not able to make the estimated system dynamics converge to the real dynamics. Hence, it completes the proof which is

$$\dot{V} < 0.$$

Remark 28. *The control law proposed in this dissertation uses Lyapunov's equation only for the proof of stability and not for the design of the control parameters.*

Remark 29 (Assumption)**.** *To evaluate $\hat{\boldsymbol{\phi}}$, which is required for the input torque, the real system dynamics $\boldsymbol{\phi}$ is approximated by a $(m-1)^{th}$ order Taylor series and m can be selected depending on the maximum frequencies of the desired joint angles. We have shown in Chapters 3 and 4 that the approximation of $\boldsymbol{\phi}$ using a Taylor series expansion will not affect the stability of the overall feedback system.*

B3 Evaluation of $\boldsymbol{S}$ and $\boldsymbol{Q}$

An easy way to identify the matrices is first to consider $\boldsymbol{Q}$ =diag(q_{11}, q_{22}, q_{33}). From (4.3), (4.7) and (4.9), $\boldsymbol{R}^{-1}, \boldsymbol{B}^T$ and $\boldsymbol{K}$ are known, which can be used to find

$$\boldsymbol{S} = \begin{bmatrix} s_{11} & s_{12} & \boldsymbol{K}_\mathrm{p} \\ s_{21} & s_{22} & \boldsymbol{K}_\mathrm{d} \\ \boldsymbol{K}_\mathrm{p} & \boldsymbol{K}_\mathrm{d} & \boldsymbol{I} \end{bmatrix}, \tag{B.3}$$

where $s_{12} = s_{21}$. Using all of the above matrices and vectors, (4.4) can be utilized to find the unknown terms of matrix $\boldsymbol{S}$:

$$s_{11} = \boldsymbol{K}_\mathrm{p}^2 + \tilde{\boldsymbol{K}}\boldsymbol{K}_\mathrm{p}\boldsymbol{K}_\mathrm{d},$$

$$s_{12} = \tilde{\boldsymbol{K}}\boldsymbol{K}_\mathrm{p} + \boldsymbol{K}_\mathrm{p}\boldsymbol{K}_\mathrm{d},$$

$$s_{22} = \boldsymbol{K}_\mathrm{d}^2 + \tilde{\boldsymbol{K}}\boldsymbol{K}_\mathrm{d}.$$

Once the matrix $\boldsymbol{S}$ is identified, matrix $\boldsymbol{Q}$ is readily available from (4.4).

B4 MATLAB Code for 3-DoF Manipulator Identification

Unlike the regressor form of the 2-DoF robot (Appendix B1), the regressor form for 3-DoF can not be expressed on paper. Therefore, a MATLAB code is given with individual elements of the regressor matrix.

```
for $i = 1 : \mathrm{length}(Q1(1,:)) - 1$                          % Length of total trajectory
% Q1(1,:) is the desired trajectory for joint 1
% Joint 1 identification,                                           % $\boldsymbol{Y}$ is the regressor matrix
    if abs $(X_s(1,i)) < 0.001$                                     % $X_s = \ddot{\boldsymbol{e}} + \boldsymbol{K}_\mathrm{d}\dot{\boldsymbol{e}} + \boldsymbol{K}_\mathrm{p}\boldsymbol{e}$
    $\boldsymbol{Y}_1(i,1) = \mathrm{qdd}_s(1,i);$                  % $i$ represents sampling time (ms)
    $\boldsymbol{Y}_1(i,2) = \cos(\mathrm{q}_s(2,i) + \mathrm{q}_s(3,i)) * \mathrm{qdd}_s(1,i);$   % $\mathrm{qdd}_s = \ddot{q}$, $\mathrm{qd}_s = \dot{q}$, $\mathrm{q}_s = q$
    $\boldsymbol{Y}_1(i,3) = \cos(\mathrm{q}_s(3,i)) * \mathrm{qdd}_s(1,i);$   %
    $\boldsymbol{Y}_1(i,4) = \cos(\mathrm{q}_s(2,i)) * \mathrm{qdd}_s(1,i);$   %
    $\boldsymbol{Y}_1(i,5) = \mathrm{qdd}_s(2,i);$                  %
    $\boldsymbol{Y}_1(i,6) = \cos(\mathrm{q}_s(2,i) + \mathrm{q}_s(3,i)) * \mathrm{qdd}_s(2,i);$   %
    $\boldsymbol{Y}_1(i,7) = \cos(\mathrm{q}_s(3,i)) * \mathrm{qdd}_s(2,i);$   %
    $\boldsymbol{Y}_1(i,8) = \cos(\mathrm{q}_s(2,i)) * \mathrm{qdd}_s(2,i);$   %
    $\boldsymbol{Y}_1(i,9) = \mathrm{qdd}_s(3,i);$                  %
    $\boldsymbol{Y}_1(i,10) = \cos(\mathrm{q}_s(2,i) + \mathrm{q}_s(3,i)) * \mathrm{qdd}_s(3,i);$   %
    $\boldsymbol{Y}_1(i,11) = \cos(\mathrm{q}_s(3,i)) * \mathrm{qdd}_s(3,i);$   %
    $\boldsymbol{Y}_1(i,12) = \sin(\mathrm{q}_s(2,i)) * (\mathrm{qd}_s(1,i)^2$   %
                    $-(\mathrm{qd}_s(1,i) + qd_s(2,i))^2);$
    $\boldsymbol{Y}_1(i,13) = \sin(\mathrm{q}_s(2,i)) * \mathrm{qd}_s(1,i)^2;$   %
    $\boldsymbol{Y}_1(i,14) = \sin(\mathrm{q}_s(2,i) +$            %
                    $\mathrm{q}_s(3,i)) * \mathrm{qd}_s(1,i)^2;$
    $\boldsymbol{Y}_1(i,15) = \sin(\mathrm{q}_s(3,i)) * (\mathrm{qd}_s(1,i) +$   %
                    $\mathrm{qd}_s(2,i))^2;$
    $\boldsymbol{Y}_1(i,16) = -\sin(\mathrm{q}_s(2,i)) * (\mathrm{qd}_s(1,i) +$   %
                    $\mathrm{qd}_s(2,i))^2;$
    $\boldsymbol{Y}_1(i,17) = -\sin(\mathrm{q}_s(3,i)) * (\mathrm{qd}_s(1,i) +$   %
                    $\mathrm{qd}_s(2,i) + \mathrm{qd}_s(3,i))^2;$
    $\boldsymbol{Y}_1(i,18) = -\sin(\mathrm{q}_s(2,i) + \mathrm{q}_s(3,i)) *$   %
                    $(\mathrm{qd}_s(1,i) + \mathrm{qd}_s(2,i) + \mathrm{qd}_s(3,i))^2;$
    $\boldsymbol{Y}_1(i,19) = \mathrm{qd}_s(1,i);$                  %
    $B1(i) = \mathrm{Uhats}(1,i);$                                  % $\mathrm{Uhats} = \hat{\boldsymbol{\phi}}$
    end                                                             %
% Joint 2 identification
    if abs $(Xs(2,i)) < 0.01$                                       %
    $\boldsymbol{Y}_2(i,1) = \mathrm{qdd}_s(1,i);$                  %
    $\boldsymbol{Y}_2(i,2) = \cos(\mathrm{q}_s(2,i) + \mathrm{q}_s(3,i)) * \mathrm{qdd}_s(1,i);$   %
    $\boldsymbol{Y}_2(i,3) = \cos(\mathrm{q}_s(3,i)) * \mathrm{qdd}_s(1,i);$   %
    $\boldsymbol{Y}_2(i,4) = \cos(\mathrm{q}_s(2,i)) * \mathrm{qdd}_s(1,i);$   %
    $\boldsymbol{Y}_2(i,5) = \mathrm{qdd}_s(2,i);$                  %
    $\boldsymbol{Y}_2(i,6) = \cos(\mathrm{q}_s(3,i)) * \mathrm{qdd}_s(2,i);$                      %
    $\boldsymbol{Y}_2(i,7) = \mathrm{qdd}_s(3,i);$                                   %
    $\boldsymbol{Y}_2(i,8) = \cos(\mathrm{q}_s(3,i)) * \mathrm{qdd}_s(3,i);$                      %
    $\boldsymbol{Y}_2(i,9) = \sin(\mathrm{q}_s(2,i)) * \mathrm{qd}_s(1,i)^2;$                     %
    $\boldsymbol{Y}_2(i,10) = \sin(\mathrm{q}_s(2,i) + \mathrm{q}_s(3,i)) * \mathrm{qd}_s(1,i)^2;$         %
    $\boldsymbol{Y}_2(i,11) = \sin(\mathrm{q}_s(3,i)) * (\mathrm{qd}_s(1,i) + \mathrm{qd}_s(2,i))^2;$     %
    $\boldsymbol{Y}_2(i,12) = -\sin(\mathrm{q}_s(3,i)) * (\mathrm{qd}_s(1,i)+$                  %
                  $\mathrm{qd}_s(2,i) + \mathrm{qd}_s(3,i))^2;$
    $\boldsymbol{Y}_2(i,13) = \mathrm{qd}_s(2,i);$                                   %
    $B2(i) = \mathrm{Uhats}(2,i);$                                     %
    end                                                  %
  % Joint 3 identification
    if  abs $(Xs(3,i)) < 0.01$                                 %
    $\boldsymbol{Y}_3(i,1) = \mathrm{qdd}_s(1,i);$                                   %
    $\boldsymbol{Y}_3(i,2) = \cos(\mathrm{q}_s(2,i) + \mathrm{q}_s(3,i)) * \mathrm{qdd}_s(1,i);$          %
    $\boldsymbol{Y}_3(i,3) = \cos(\mathrm{q}_s(3,i)) * \mathrm{qdd}_s(1,i);$                      %
    $\boldsymbol{Y}_3(i,4) = \mathrm{qdd}_s(2,i);$                                   %
    $\boldsymbol{Y}_3(i,5) = \cos(\mathrm{q}_s(3,i)) * \mathrm{qdd}_s(2,i);$                      %
    $\boldsymbol{Y}_3(i,6) = \mathrm{qdd}_s(3,i);$                                   %
    $\boldsymbol{Y}_3(i,7) = \sin(\mathrm{q}_s(2,i) + \mathrm{q}_s(3,i)) * \mathrm{qd}_s(1,i)^2;$          %
    $\boldsymbol{Y}_3(i,8) = \sin(\mathrm{q}_s(3,i)) * (\mathrm{qd}_s(1,i) + \mathrm{qd}_s(2,i))^2;$      %
    $\boldsymbol{Y}_3(i,9) = \mathrm{qd}_s(3,i);$                                    %
    $B3(i) = \mathrm{Uhats}(3,i);$                                     %
    end                                                  %
    $i = i + 1;$                                             %
  end                                                    %
  % $P1$ consists of parameters of joint 1
    $P1 = \mathrm{inv}(\boldsymbol{Y}_1' * \boldsymbol{Y}_1) * \boldsymbol{Y}_1' * B1'$                           %
  % $P2$ consists of parameters of joint 2
    $P2 = \mathrm{inv}(\boldsymbol{Y}_2' * \boldsymbol{Y}_2) * \boldsymbol{Y}_2' * B2'$                           %
  % $P3$ consists of parameters of joint 3
    $P3 = \mathrm{inv}(\boldsymbol{Y}_3' * \boldsymbol{Y}_3) * \boldsymbol{Y}_3' * B3'$                           %
```

C

Details of the 3-DoF Manipulator

The inertia of each component for the 3-DoF manipulator (Fig. C.1) is calculated using basic mathematical tools. Apart from motors and gears, the rest of the components are made of aluminum with density 2700 kg/km^3. Maxon motors are used along with harmonic drives with turn ratios of 100 : 1. The motor torques are controlled by PWM amplifiers, operating in current control mode with the reference given by a voltage from the D/A converter output of the I/O board (Sensoray 626). The position of each joint is measured by an optic pulse incremental encoder on the motor shaft and then processed by a quadrature encoder on the I/O board.

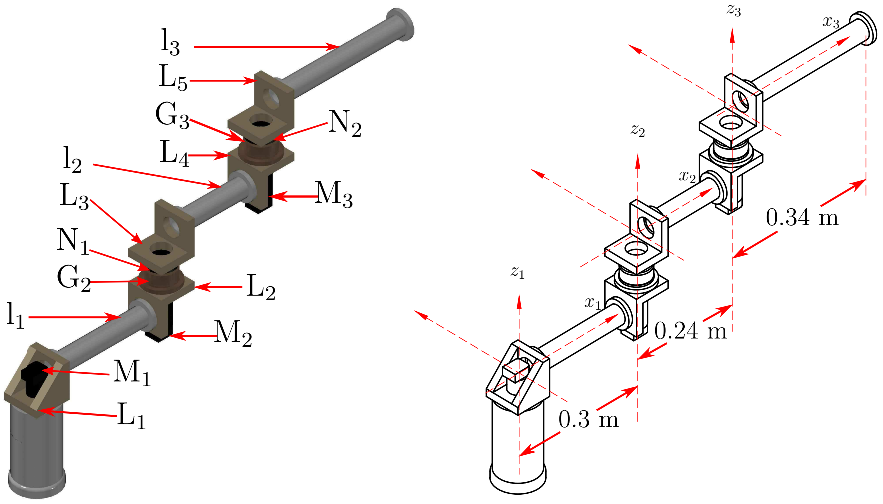


Figure C.1: Details of the 3-DoF manipulator used in the dissertation.

C1 Forward Kinematics

The location of the third link with respect to the base is given by

$$\boldsymbol{T}_0^3 = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{p} \\ \boldsymbol{0} & 1 \end{bmatrix}, \tag{C.1}$$

where $\boldsymbol{T}_0^3$ is computed using the link homogeneous matrices as in [8]:

$$\boldsymbol{T}_0^3 = \boldsymbol{A}_0^1 \boldsymbol{A}_1^2 \boldsymbol{A}_2^3. \tag{C.2}$$

The matrices $\boldsymbol{A}_{(\cdot)}^{(\cdot)}$ are given as

$$\boldsymbol{A}_0^1 = \begin{bmatrix} \cos(q_1) & -\sin(q_1) & 0 & 0 \\ \sin(q_1) & \cos(q_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{A}_1^2 = \begin{bmatrix} \cos(q_2) & -\sin(q_2) & 0 & l_1 \\ \sin(q_2) & \cos(q_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\boldsymbol{A}_2^3 = \begin{bmatrix} \cos(q_3) & -\sin(q_3) & 0 & l_2 \\ \sin(q_3) & \cos(q_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{C.3}$$

C2 Mass Properties

Following are the inertia matrices:

$$\boldsymbol{J} = \begin{bmatrix} \int x^2 \mathrm{dm} & \int xy \mathrm{dm} & \int xz \mathrm{dm} & \int x \mathrm{dm} \\ \int yx \mathrm{dm} & \int y^2 \mathrm{dm} & \int yz \mathrm{dm} & \int y \mathrm{dm} \\ \int zx \mathrm{dm} & \int zy \mathrm{dm} & \int z^2 \mathrm{dm} & \int z \mathrm{dm} \\ \int x \mathrm{dm} & \int y \mathrm{dm} & \int z \mathrm{dm} & \int \mathrm{dm} \end{bmatrix} = \begin{bmatrix} J_{xx} & J_{xy} & J_{xz} & J_x \\ J_{yx} & J_{yy} & J_{yz} & J_y \\ J_{zx} & J_{zy} & J_{zz} & J_z \\ J_x & J_y & J_z & J_m \end{bmatrix} \tag{C.4}$$

$$\boldsymbol{I} = \begin{bmatrix} \int (y^2 + z^2) \mathrm{dm} & -\int xy \mathrm{dm} & -\int xz \mathrm{dm} \\ -\int yx \mathrm{dm} & \int (x^2 + z^2) \mathrm{dm} & -\int yz \mathrm{dm} \\ -\int zx \mathrm{dm} & -\int zy \mathrm{dm} & \int (x^2 + y^2) \mathrm{dm} \end{bmatrix} \tag{C.5}$$

The Euler-Lagrange (E-L) representation of the manipulator is acquired using the procedure given in [140]. We used matrix $\boldsymbol{J}$ and the conventional inertia matrix $\boldsymbol{I}$ can be evaluated from the matrix $\boldsymbol{J}$. In the following figures, all the dimensions are in meters. Also, the inertia of each component is taken with respect to its frame of reference.

Table C.1: Details of L$_1$.

Link	1	L$_1$
Mass (Kg)		0.485
[kgm]	J_x	0.0058
	J_y	0
	J_z	-0.0058
Inertia [kgm^2]	J_{xx}	0.0004
	J_{yy}	0.0004
	J_{zz}	0.0003
	J_{xy}	0
	J_{xz}	-0.0015
	J_{yz}	0

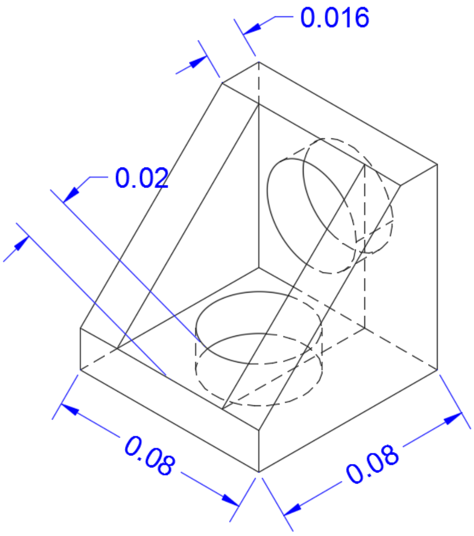


Figure C.2: Dimensions of L$_1$.

Table C.2: Details of L$_2$, L$_3$, L$_4$ and L$_5$.

Link	$1, 2, 3$	L$_2$ and L$_4$	L$_3$ and L$_5$
Mass (Kg)		0.395	0.395
[kgm]	J_x	0.1132	0.0090
	J_y	0	0
	J_z	0.009	-0.0090
Inertia [kgm^2]	J_{xx}	0.0327	0.0004
	J_{yy}	0.0003	0.0003
	J_{zz}	0.0004	0.0004
	J_{xy}	0	0
	J_{xz}	0.0028	-0.0001
	J_{yz}	0	0

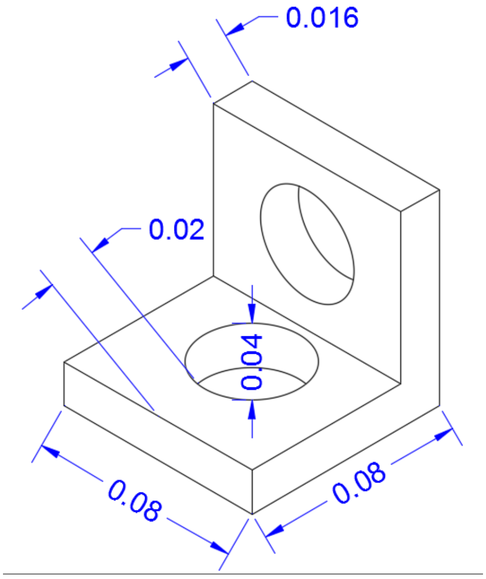


Figure C.3: Dimensions of L$_2$, L$_3$, L$_4$ and L$_5$.

Table C.3: Details of l_1.

Link	1	l_1
Mass (Kg)		0.398
[kgm]	J_x	0.0583
	J_y	0
	J_z	0
Inertia [kgm^2]	J_{xx}	0.0109
	J_{yy}	0.0001
	J_{zz}	0.0001
	J_{xy}	0
	J_{xz}	0
	J_{yz}	0

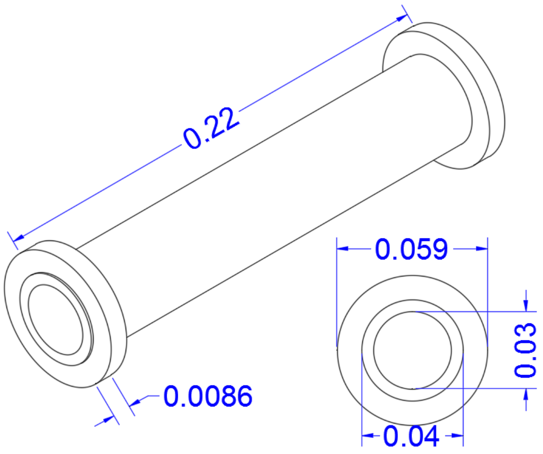


Figure C.4: Dimensions of l_1.

Table C.4: Details of l_2.

Link	2	l_2
Mass (Kg)		0.3028
[kgm]	J_x	0.0387
	J_y	0
	J_z	0
Inertia [kgm^2]	J_{xx}	0.0058
	J_{yy}	0.0001
	J_{zz}	0.0001
	J_{xy}	0
	J_{xz}	0
	J_{yz}	0

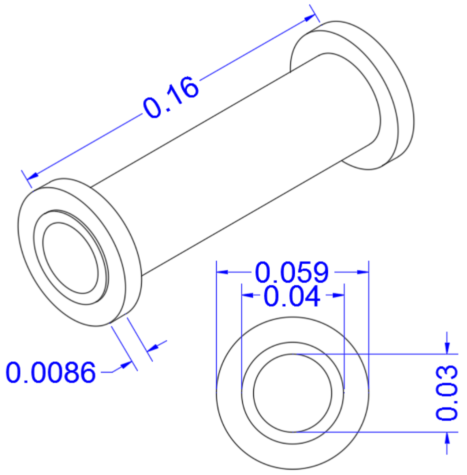


Figure C.5: Dimensions of l_2.

Table C.5: Details of l_3.

Link	3	l_3
Mass (Kg)		0.5015
[kgm]	J_x	0.0951
	J_y	0
	J_z	0
Inertia [kgm^2]	J_{xx}	0.0227
	J_{yy}	0.0001
	J_{zz}	0.0001
	J_{xy}	0
	J_{xz}	0
	J_{yz}	0

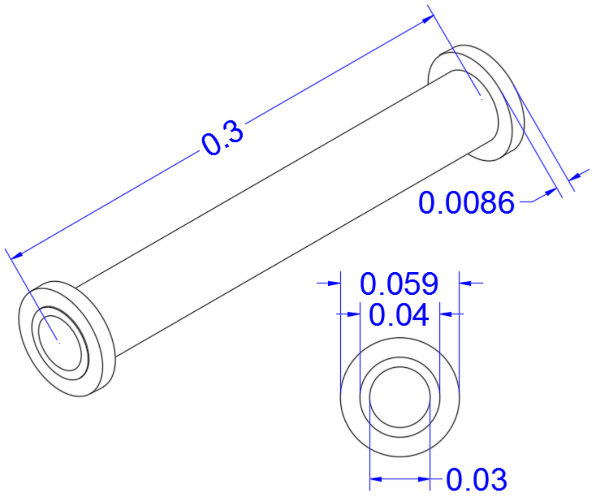


Figure C.6: Dimensions of l_3.

Table C.6: Details of M_1, M_2 and M_3.

Link	$1, 2, 3$	M_1, M_2 and M_3
Mass (Kg)		0.28
[kgm]	J_x	0.084
	J_y	0
	J_z	0
Inertia [kgm^2]	J_{xx}	0.0252
	J_{yy}	0
	J_{zz}	0
	J_{xy}	0
	J_{xz}	0
	J_{yz}	0

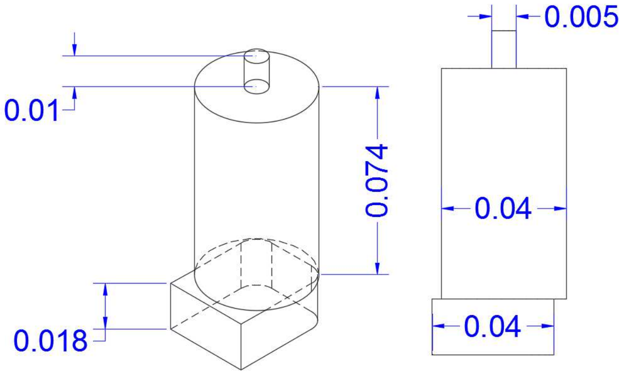


Figure C.7: Dimensions of M_1, M_2 and M_3.

Table C.7: Details of G_1, G_2 and G_3.

Link	$1, 2, 3$	G_1, G_2 and G_3
Mass (Kg)		0.65
[kgm]	J_x	0.195
	J_y	0
	J_z	0.0432
Inertia [kgm^2]	J_{xx}	0.0585
	J_{yy}	0
	J_{zz}	0.0029
	J_{xy}	0
	J_{xz}	0.013
	J_{yz}	0

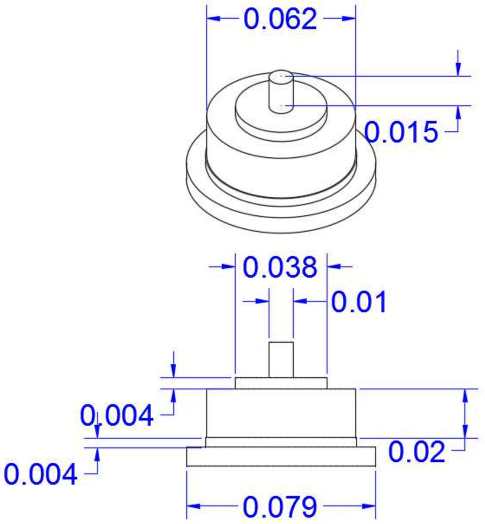


Figure C.8: Dimensions of G_1, G_2 and G_3.

Table C.8: Details of N_1 and N_2.

Link	$2, 3$	N_1 and N_2
Mass (Kg)		0.1113
[kgm]	J_x	0
	J_y	0
	J_z	-0.0002
Inertia [kgm^2]	J_{xx}	0
	J_{yy}	0
	J_{zz}	0.0003
	J_{xy}	0
	J_{xz}	0
	J_{yz}	0

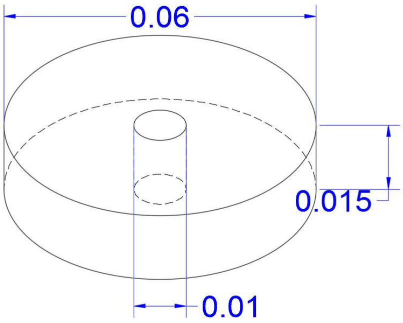


Figure C.9: Dimensions of N_1 and N_2.

Bibliography

[1] R. Hayat, M. Leibold, and M. Buss, "Robust-adaptive controller design for robot manipulators using the $\mathcal{H}_\infty$ approach," *IEEE Access*, vol. 6, pp. 51 626 – 51 639, 2018.

[2] ——, "Addressing control implementation issues in robotic systems using adaptive control," *Robotica*, pp. 1–14, 2019.

[3] R. Hayat and M. Buss, "Model identification for robot manipulators using regressor-free adaptive control," *IEEE 11th International Conference on Control (UKACC)*, pp. 1–7, 2016.

[4] Y. Sun, Z. Zhang, M. Leibold, R. Hayat, D. Wollherr, and M. Buss, "Protective control for robot manipulator by sliding mode based disturbance reconstruction approach," *IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 1015–1022, 2017.

[5] Y. Choi, W. K. Chung, and I. H. Suh, "Performance and $\mathcal{H}_\infty$ optimality of PID trajectory tracking controller for Lagrangian systems," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 857–869, 2001.

[6] J. Wallén, *The history of the industrial robot.* Linköping University Electronic Press, 2008.

[7] R. J. Schilling, *Fundamentals of robotics: analysis and control.* Prentice Hall New Jersey, 1990.

[8] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control.* Wiley New York, 2006.

[9] W. Khalil and E. Dombre, *Modeling, identification and control of robots.* Butterworth-Heinemann, 2004.

[10] M. J. Kim, Y. Choi, and W. K. Chung, "Bringing nonlinear $\mathcal{H}_\infty$ optimality to robot controllers," *IEEE Transactions on Robotics*, vol. 31, no. 3, pp. 682–698, 2015.

[11] J. Park and W. K. Chung, "Analytic nonlinear $\mathcal{H}_\infty$ inverse-optimal control for Euler-Lagrange system," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 6, pp. 847–854, 2000.

[12] A. Del Prete and N. Mansard, "Robustness to joint-torque-tracking errors in task-space inverse dynamics," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1091–1105, 2016.

[13] C. Y. Kai and A. C. Huang, “A regressor-free adaptive controller for robot manipulators without Slotine and Li’s modification,” *Robotica*, vol. 31, no. 07, pp. 1051–1058, 2013.

[14] M. C. Chien and A. C. Huang, “Adaptive impedance controller design for flexible-joint electrically-driven robots without computation of the regressor matrix,” *Robotica*, vol. 30, no. 01, pp. 133–144, 2012.

[15] Y. Yanling, “Model free adaptive control for robotic manipulator trajectory tracking,” *The Open Automation and Control Systems Journal*, vol. 7, no. 1, 2015.

[16] A. Izadbakhsh and M. M. Fateh, “A model-free robust control approach for robot manipulator,” *International Journal of Electrical and Computer Engineering*, vol. 25, p. 30, 2007.

[17] L. Cai and X. Tang, “A model-free decentralized control for robot manipulators,” in *International Conference on Robotics and Automation*, vol. 4. IEEE, 1997, pp. 3106–3111.

[18] C. M. Hackl, A. G. Hofmann, and R. M. Kennel, “Funnel control in mechatronics: An overview,” in *Conference on Decision and Control and European Control Conference (CDC-ECC)*. IEEE, 2011, pp. 8000–8007.

[19] C. M. Hackl, “High-gain adaptive position control,” *International Journal of Control*, vol. 84, no. 10, pp. 1695–1716, 2011.

[20] R. P. Pagilla and M. Tomizuka, “An adaptive output feedback controller for robot arms: stability and experiments,” *Automatica*, vol. 37, no. 7, pp. 983–995, 2001.

[21] J. Na, M. N. Mahyuddin, G. Herrmann, X. Ren, and P. Barber, “Robust adaptive finite-time parameter estimation and control for robotic systems,” *International Journal of Robust and Nonlinear Control*, vol. 25, no. 16, pp. 3045–3071, 2015.

[22] J. Na, G. Herrmann, X. Ren, M. N. Mahyuddin, and P. Barber, “Robust adaptive finite-time parameter estimation and control of nonlinear systems,” in *International Symposium on Intelligent Control (ISIC)*. IEEE, 2011, pp. 1014–1019.

[23] L. Yu, S. Fei, L. Sun, J. Huang, and G. Yang, “Design of robust adaptive neural switching controller for robotic manipulators with uncertainty and disturbances,” *Journal of Intelligent & Robotic Systems*, vol. 77, no. 3-4, pp. 571–581, 2015.

[24] L. Yu and S. Fei, “Robustly stable switching neural control of robotic manipulators using average dwell-time approach,” *Transactions of the Institute of Measurement and Control*, vol. 36, no. 6, pp. 789–796, 2014.

[25] W. He, H. Huang, and S. S. Ge, “Adaptive neural network control of a robotic manipulator with time-varying output constraints,” *IEEE Transactions on Cybernetics*, vol. 47, no. 10, pp. 3136–3147, 2017.

[26] M. Li, Y. Li, S. S. Ge, and T. H. Lee, “Adaptive control of robotic manipulators with unified motion constraints,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 1, pp. 184–194, 2017.

[27] B. Xiao, S. Yin, O. Kaynak, and H. Gao, "Observer-based control for robotic manipulations with uncertain kinematics and dynamics," *IEEE 14th International Workshop on Advanced Motion Control (AMC)*, pp. 282–288, 2016.

[28] J. Qin, F. Léonard, and G. Abba, "Real-time trajectory compensation in robotic friction stir welding using state estimators," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 6, pp. 2207–2214, 2016.

[29] S. S. Pchelkin, A. S. Shiriaev, A. Robertsson, L. B. Freidovich, S. A. Kolyubin, L. V. Paramonov, and S. V. Gusev, "On orbital stabilization for industrial manipulators: Case study in evaluating performances of modified PD+ and inverse dynamics controllers," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 1, pp. 101–117, 2017.

[30] A. Shiriaev, J. W. Perram, and C. Canudas-de Wit, "Constructive tool for orbital stabilization of underactuated nonlinear systems: Virtual constraints approach," *IEEE Transactions on Automatic Control*, vol. 50, no. 8, pp. 1164–1176, 2005.

[31] R. Hedjar and P. Boucher, "Nonlinear receding-horizon control of rigid link robot manipulators," *International Journal of Advanced Robotic Systems*, vol. 2, no. 1, pp. 15–24, 2005.

[32] P. Poignet and M. Gautier, "Nonlinear model predictive control of a robot manipulator," in *International Workshop on Advanced Motion Control.* IEEE, 2000, pp. 401–406.

[33] R. Hedjar, R. Toumi, P. Boucher, and D. Dumur, "Finite horizon nonlinear predictive control by the Taylor approximation: application to robot tracking trajectory," *International Journal of Applied Mathematics and Computer Science*, vol. 15, pp. 527–540, 2005.

[34] A. Rojas-Moreno and R. Valdivia-Mallqui, "Embedded position control system of a manipulator using a robust nonlinear predictive control," in *International Conference on Advanced Robotics (ICAR).* IEEE, 2013, pp. 1–6.

[35] A. Vivas and V. Mosquera, "Predictive functional control of a PUMA robot," in *ACSE Conference Proceedings*, 2005.

[36] M. A. Arteaga-Pérez and A. Gutiérrez-Giles, "On the GPI approach with unknown inertia matrix in robot manipulators," *International Journal of Control*, vol. 87, no. 4, pp. 844–860, 2014.

[37] J. Han, "From PID to active disturbance rejection control," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 3, pp. 900–906, 2009.

[38] A. Ferrara and G. P. Incremona, "Design of an integral suboptimal second-order sliding mode controller for the robust motion control of robot manipulators," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 6, pp. 2316–2325, 2015.

[39] L. Yu, J. Huang, and S. Fei, "Sliding mode switching control of manipulators based on disturbance observer," *Circuits, Systems, and Signal Processing*, vol. 36, no. 6, pp. 2574–2585, 2017.

[40] J. J. Craig, *Introduction to robotics: mechanics and control.* Pearson Prentice Hall Upper Saddle River, 2005, vol. 3.

[41] M. I. Ullah, S. A. Ajwad, R. U. Islam, U. Iqbal, and J. Iqbal, "Modeling and computed torque control of a 6-degree of freedom robotic arm," *International Conference on Robotics and Emerging Allied Technologies in Engineering (iCREATE)*, pp. 133–138, 2014.

[42] A. C. Bittencourt and S. Gunnarsson, "Static friction in a robot joint-modeling and identification of load and temperature effects," *Journal of Dynamic Systems, Measurement, and Control*, vol. 134, no. 5, 2012.

[43] C. C. De Wit, H. Olsson, K. J. Astrom, and P. Lischinsky, "A new model for control of systems with friction," *IEEE Transactions on Automatic Control*, vol. 40, no. 3, pp. 419–425, 1995.

[44] K. J. Astrom and C. Canudas-De-Wit, "Revisiting the lugre friction model," *IEEE Control Systems*, vol. 28, no. 6, pp. 101–114, 2008.

[45] C. Makkar, G. Hu, W. G. Sawyer, and W. E. Dixon, "Lyapunov-based tracking control in the presence of uncertain nonlinear parameterizable friction," *IEEE Transactions on Automatic Control*, vol. 52, no. 10, pp. 1988–1994, 2007.

[46] C. Makkar, W. E. Dixon, W. G. Sawyer, and G. Hu, "A new continuously differentiable friction model for control systems design," in *ASME International Conference on Advanced Intelligent Mechatronics.* IEEE, 2005, pp. 600–605.

[47] H. G. Sage, M. F. De Mathelin, and E. Ostertag, "Robust control of robot manipulators: A survey," *International Journal of Control*, vol. 72, no. 16, pp. 1498–1522, 1999.

[48] T. Brogårdh, "Robot control overview: An industrial perspective," *Modeling, Identification and Control*, vol. 30, no. 3, p. 167, 2009.

[49] J. J. Slotine and W. Li, "On the adaptive control of robot manipulators," *The International Journal of Robotics Research*, vol. 6, no. 3, pp. 49–59, 1987.

[50] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control.* Springer Science & Business Media, 2009.

[51] L. Bascetta and P. Rocco, "Revising the robust-control design for rigid robot manipulators," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 180–187, 2010.

[52] W. Peng, Z. Lin, and J. Su, "Computed torque control-based composite nonlinear feedback controller for robot manipulators with bounded torques," *IET Control Theory & Applications*, vol. 3, no. 6, pp. 701–711, 2009.

[53] J. Na, G. Herrmann, and K. Zhang, “Improving transient performance of adaptive control via a modified reference model and novel adaptation,” *International Journal of Robust and Nonlinear Control*, vol. 27, no. 8, pp. 1351–1372, 2017.

[54] H. Wang, “Adaptive control of robot manipulators with uncertain kinematics and dynamics,” *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 948–954, 2017.

[55] P. M. Patre, W. MacKunis, K. Dupree, and W. E. Dixon, “Modular adaptive control of uncertain euler–lagrange systems with additive disturbances,” *IEEE Transactions on Automatic Control*, vol. 56, no. 1, pp. 155–160, 2011.

[56] E. Kharisov, N. Hovakimyan, and K. J. Åström, “Comparison of architectures and robustness of model reference adaptive controllers and adaptive controllers,” *International Journal of Adaptive Control and Signal Processing*, vol. 28, no. 7-8, pp. 633–663, 2014.

[57] P. M. Patre, W. MacKunis, C. Makkar, and W. E. Dixon, “Asymptotic tracking for systems with structured and unstructured uncertainties,” *Transactions on Control Systems Technology*, vol. 16, no. 2, pp. 373–379, 2008.

[58] N. Sharma, S. Bhasin, Q. Wang, and W. E. Dixon, “Predictor-based control for an uncertain Euler–Lagrange system with input delay,” *Automatica*, vol. 47, no. 11, pp. 2332–2342, 2011.

[59] M. A. Arteaga and Y. Tang, “Adaptive control of robots with an improved transient performance,” *IEEE Transactions on Automatic Control*, vol. 47, no. 7, pp. 1198–1202, 2002.

[60] H. Liu, X. Tian, G. Wang, and T. Zhang, “Finite-time $\mathcal{H}_\infty$ control for high-precision tracking in robotic manipulators using backstepping control,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 9, pp. 5501–5513, 2016.

[61] S. M. Hashemi, H. S. Abbas, and H. Werner, “Low-complexity linear parameter-varying modeling and control of a robotic manipulator,” *Control Engineering Practice*, vol. 20, no. 3, pp. 248–257, 2012.

[62] J. A. Rossiter, *Model-based predictive control: a practical approach.* CRC press, first edition, 2003.

[63] S. Islam, P. X. Liu, and A. El Saddik, “Experimental comparison of model-based and model-free output feedback control system for robot manipulators,” in *International Conference on Autonomous and Intelligent Systems.* Springer, 2011, pp. 177–188.

[64] A. C. Huang, S. C. Wu, and W. F. Ting, “A FAT-based adaptive controller for robot manipulators without regressor matrix: theory and experiments,” *Robotica*, vol. 24, no. 02, pp. 205–210, 2006.

[65] A. Safaei, Y. C. Koo, and M. N. Mahyuddin, “Adaptive model-free control for robotic manipulators,” in *IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*, 2017, pp. 7–12.

[66] Z.-J. 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, 2012.

[67] V. Utkin, J. Guldner, and J. Shi, *Sliding mode control in electro-mechanical systems.* CRC press, 2009.

[68] D. Nojavanzadeh and M. Badamchizadeh, "Adaptive fractional-order non-singular fast terminal sliding mode control for robot manipulators," *IET Control Theory & Applications*, vol. 10, no. 13, pp. 1565–1572, 2016.

[69] B. Xiao, S. Yin, and O. Kaynak, "Tracking control of robotic manipulators with uncertain kinematics and dynamics," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 10, pp. 6439–6449, 2016.

[70] M. M. Fateh, S. Azargoshasb, and S. Khorashadizadeh, "Model-free discrete control for robot manipulators using a fuzzy estimator," *COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, vol. 33, no. 3, pp. 1051–1067, 2014.

[71] M. M. Fateh and S. Azargoshasb, "Discrete time robust control of robot manipulators in the task space using adaptive fuzzy estimator," *Journal of AI and Data Mining*, vol. 3, no. 1, pp. 113–120, 2015.

[72] J. Guan, C.-M. Lin, G.-L. Ji, L.-W. Qian, and Y.-M. Zheng, "Robust adaptive tracking control for manipulators based on a tsk fuzzy cerebellar model articulation controller," *IEEE Access*, vol. 6, pp. 1670–1679, 2018.

[73] L. Wang, T. Chai, and C. Yang, "Neural-network-based contouring control for robotic manipulators in operational space," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 4, pp. 1073–1080, 2012.

[74] C. Zhang and H.-S. Yan, "Inverse control of multi-dimensional Taylor network for permanent magnet synchronous motor," *COMPEL-The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, vol. 36, no. 6, pp. 1676–1689, 2017.

[75] C. P. Bechlioulis, M. V. Liarokapis, and K. J. Kyriakopoulos, "Robust model free control of robotic manipulators with prescribed transient and steady state performance," in *RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014, pp. 41–46.

[76] N. Alibeji and N. Sharma, "A PID-type robust input delay compensation method for uncertain euler–lagrange systems," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 6, pp. 2235–2242, 2017.

[77] M. Dulău, A. Gligor, and T. M. Dulău, "Fractional order controllers versus integer order controllers," *Procedia Engineering*, vol. 181, pp. 538–545, 2017.

[78] A. Dumlu and K. Erenturk, “Trajectory tracking control for a 3-dof parallel manipulator using fractional-order PI$^\lambda$D$^\mu$ control,” *IEEE Transactions on Industrial Electronics*, vol. 61, no. 7, pp. 3417–3426, 2014.

[79] N. Nikdel, M. Badamchizadeh, V. Azimirad, and M. A. Nazari, “Fractional-order adaptive backstepping control of robotic manipulators in the presence of model uncertainties and external disturbances,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 10, pp. 6249–6256, 2016.

[80] N. Nikdel and M. A. Badamchizadeh, “Fractional-order adaptive backstepping control of a class of uncertain systems with external disturbances,” *International Journal of Control*, pp. 1–10, 2017.

[81] M. Fliess and C. Join, “Intelligent PID controllers,” in *16th Mediterranean Conference on Control and Automation.* IEEE, 2008, pp. 326–331.

[82] M. Fliess and S. Riachy, “Revisiting some practical issues in the implementation of model-free control,” *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 8589–8594, 2011.

[83] M. Fliess and C. Join, “Model-free control,” *International Journal of Control*, vol. 86, no. 12, pp. 2228–2252, 2013.

[84] L. Yu, S. Fei, and X. Li, “Robust adaptive neural tracking control for a class of switched affine nonlinear systems,” *Neurocomputing*, vol. 73, no. 10-12, pp. 2274–2279, 2010.

[85] C. P. Bechlioulis, Z. Doulgeri, and G. A. R., “Guaranteeing prescribed performance and contact maintenance via an approximation free robot force/position controller,” *Automatica*, vol. 48, no. 2, pp. 360–365, 2012.

[86] F. L. Lewis and D. Vrabie, “Reinforcement learning and adaptive dynamic programming for feedback control,” *IEEE Circuits and Systems Magazine*, vol. 9, no. 3, 2009.

[87] D. Vrabie, K. G. Vamvoudakis, and F. L. Lewis, *Optimal adaptive control and differential games by reinforcement learning principles.* IET, 2013, vol. 2.

[88] S. G. Khan, G. Herrmann, F. L. Lewis, T. Pipe, and C. Melhuish, “Reinforcement learning and optimal adaptive control: An overview and implementation examples,” *Annual Reviews in Control*, vol. 36, no. 1, pp. 42–59, 2012.

[89] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, “Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers,” *IEEE Control Systems*, vol. 32, no. 6, pp. 76–105, 2012.

[90] E. D. Sontag and Y. Wang, “On characterizations of the input-to-state stability property,” *Systems & Control Letters*, vol. 24, no. 5, pp. 351–359, 1995.

[91] E. D. Sontag, “Smooth stabilization implies coprime factorization,” *IEEE Transactions on Automatic Control*, vol. 34, no. 4, pp. 435–443, 1989.

[92] M. Krstic and Z. H. Li, “Inverse optimal design of input-to-state stabilizing nonlinear controllers,” *IEEE Transactions on Automatic Control*, vol. 43, no. 3, pp. 336–350, 1998.

[93] C. Kim and K. Lee, "Robust control of robot manipulators using dynamic compensators under parametric uncertainty," *International Journal of Innovative Computing, Information and Control*, vol. 7, no. 7, pp. 4129–4137, 2011.

[94] K. Ogata, "Modern control engineering," *Prentice Hall, 5th edition*, p. 55, 2009.

[95] M. K. Ciliz and K. S. Narendra, "Adaptive control of robotic manipulators using multiple models and switching," *The International Journal of Robotics Research*, vol. 15, no. 6, pp. 592–610, 1996.

[96] N. S. Nise, *Control systems engineering.* John Wiley & Sons, fifth edition, 2007.

[97] C. D. Sousa and R. Cortesão, "Physical feasibility of robot base inertial parameter identification: A linear matrix inequality approach," *The International Journal of Robotics Research*, vol. 33, no. 6, pp. 931–944, 2014.

[98] J. Swevers, C. Ganseman, D. B. Tükel, J. De Schutter, and H. Van Brussel, "Optimal robot excitation and identification," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 5, pp. 730–740, 1997.

[99] D. Kostic, B. De Jager, M. Steinbuch, and R. Hensen, "Modeling and identification for high-performance robot control: An RRR-robotic arm case study," *Transactions on Control Systems Technology*, vol. 12, no. 6, pp. 904–919, 2004.

[100] E. Wernholt and S. Moberg, "Nonlinear gray-box identification using local models applied to industrial robots," *Automatica*, vol. 47, no. 4, pp. 650–660, 2011.

[101] H. Mayeda, K. Yoshida, and K. Osuka, "Base parameters of manipulator dynamic models," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 3, pp. 312–321, 1990.

[102] N. Ramdani and P. Poignet, "Robust dynamic experimental identification of robots with set membership uncertainty," *IEEE/ASME Transactions on Mechatronics*, vol. 10, no. 2, pp. 253–256, 2005.

[103] G. Chen, H. Wang, and Z. Lin, "Determination of the identifiable parameters in robot calibration based on the poe formula," *IEEE Transactions on Robotics*, vol. 30, no. 5, pp. 1066–1077, 2014.

[104] A. Janot, P. O. Vandanjon, and M. Gautier, "A revised Durbin-Wu-Hausman test for industrial robot identification," *Control Engineering Practice*, vol. 48, pp. 52–62, 2016.

[105] J. Jin and N. Gans, "Parameter identification for industrial robots with a fast and robust trajectory design approach," *Robotics and Computer-Integrated Manufacturing*, vol. 31, pp. 21–29, 2015.

[106] J. Wu, J. Wang, and Z. You, "An overview of dynamic parameter identification of robots," *Robotics and Computer-Integrated Manufacturing*, vol. 26, no. 5, pp. 414–419, 2010.

[107] N. Ramdani and P. Poignet, “Robust dynamic experimental identification of robots with set membership uncertainty,” *ASME Transactions on Mechatronics*, vol. 10, no. 2, pp. 253–256, 2005.

[108] Z. Qin, L. Baron, and L. Birglen, “A new approach to the dynamic parameter identification of robotic manipulators,” *Robotica*, vol. 28, no. 4, pp. 539–547, 2010.

[109] M. Díaz-Rodríguez, V. Mata, Á. Valera, and Á. Page, “A methodology for dynamic parameters identification of 3-dof parallel robots in terms of relevant parameters,” *Mechanism and Machine Theory*, vol. 45, no. 9, pp. 1337–1356, 2010.

[110] K. J. Åström and B. Wittenmark, *Adaptive control.* Courier Corporation, 2013.

[111] G. Chowdhary and E. Johnson, “Concurrent learning for convergence in adaptive control without persistency of excitation,” in *Conference on Decision and Control (CDC).* IEEE, 2010, pp. 3674–3679.

[112] G. Chowdhary, M. Mühlegg, and E. Johnson, “Exponential parameter and tracking error convergence guarantees for adaptive controllers without persistency of excitation,” *International Journal of Control*, vol. 87, no. 8, pp. 1583–1603, 2014.

[113] H. K. Khalil, *Nonlinear systems.* Prentice-Hall, New Jersey, 1996.

[114] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal control.* Wiley, 2012, vol. 3.

[115] Y. Choi and W. K. Chung, “On the optimality and performance of PID controller for robotic manipulators,” in *ICRA*, 2001, pp. 1142–1148.

[116] M. J. Kim, S. Park, and W. K. Chung, “Nonlinear robust internal loop compensator for robust control of robotic manipulators,” in *International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2012, pp. 2742–2748.

[117] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, “Time-optimal path tracking for robots: A convex optimization approach,” *IEEE Transactions on Automatic Control*, vol. 54, no. 10, pp. 2318–2327, 2009.

[118] J. E. Bobrow, S. Dubowsky, and J. S. Gibson, “Time-optimal control of robotic manipulators along specified paths,” *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 3–17, 1985.

[119] J. M. Hollerbach, “Dynamic scaling of manipulator trajectories,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 106, no. 1, pp. 102–106, 1984.

[120] K. Shin and N. McKay, “A dynamic programming approach to trajectory planning of robotic manipulators,” *IEEE Transactions on Automatic Control*, vol. 31, no. 6, pp. 491–500, 1986.

[121] ——, “Minimum-time control of robotic manipulators with geometric path constraints,” *IEEE Transactions on Automatic Control*, vol. 30, no. 6, pp. 531–541, 1985.

[122] A. S. Shiriaev, L. B. Freidovich, and S. V. Gusev, “Transverse linearization for mechanical systems with several passive degrees of freedom with applications to orbital stabilization,” in *American Control Conference.* IEEE, 2009, pp. 3039–3044.

[123] U. Mettin, “Applications of the virtual holonomic constraints approach: Analysis of human motor patterns and passive walking gaits,” Ph.D. dissertation, Umeå university, 2008.

[124] M. La Hera and P. Xavier, “Contributions to motion planning and orbital stabilization: case studies: Furuta pendulum swing up, inertia wheel oscillations and biped robot walking,” Ph.D. dissertation, Umeå university, 2008.

[125] A. S. Shiriaev, L. B. Freidovich, and I. R. Manchester, “Can we make a robot ballerina perform a pirouette? Orbital stabilization of periodic motions of underactuated mechanical systems,” *Annual Reviews in Control*, vol. 32, no. 2, pp. 200–211, 2008.

[126] H. A. Modares, F. L. Lewis, and M.-B. N. Sistani, “Online solution of nonquadratic two-player zero-sum games arising in the h-infinity control of constrained input systems,” *International Journal of Adaptive Control and Signal Processing*, vol. 28, no. 3-5, pp. 232–254, 2014.

[127] J. J.-B. M. Ahanda, J. B. Mbede, A. Melingui, and B. Essimbi, “Robust adaptive control for robot manipulators: Support vector regression-based command filtered adaptive backstepping approach,” *Robotica*, pp. 1–19, 2017.

[128] T. E. Gibson, A. M. Annaswamy, and E. Lavretsky, “On adaptive control with closed-loop reference models: transients, oscillations, and peaking,” *IEEE Access*, vol. 1, pp. 703–717, 2013.

[129] T. Basar and G. J. Olsder, *Dynamic noncooperative game theory.* Siam, 1999, vol. 23.

[130] J. Na, G. Herrmann, and K. Zhang, “Improving transient performance of adaptive control via a modified reference model and novel adaptation,” *International Journal of Robust and Nonlinear Control*, vol. 27, no. 8, pp. 1351–1372, 2017.

[131] E. D. Sontag, “Input to state stability: Basic concepts and results,” *Nonlinear and Optimal Control Theory*, pp. 163–220, 2008.

[132] M. Abu-Khalaf, F. L. Lewis, and J. Huang, “Hamilton-jacobi-isaacs formulation for constrained input nonlinear systems,” *Decision and Control*, vol. 5, pp. 5034–5040, 2004.

[133] F. L. Lewis, M. Abu-Khalaf, and J. Huang, “Hamilton-jacobi-isaacs formulation for constrained input systems: Neural network solution,” *IFAC Proceedings*, vol. 37, no. 21, pp. 01–06, 2004.

[134] K. Galloway, K. Sreenath, A. D. Ames, and J. W. Grizzle, “Torque saturation in bipedal robotic walking through control lyapunov function-based quadratic programs,” *IEEE Access*, vol. 3, pp. 323–332, 2015.

[135] T. Cimen, “State-dependent riccati equation (SDRE) control: A survey,” *IFAC Proceedings*, vol. 41, no. 2, pp. 3761–3775, 2008.

[136] J. R. Cloutier, C. N. D’Souza, and C. P. Mracek, “Nonlinear regulation and nonlinear h-infinity control via the state-dependent riccati equation technique: Part 1, theory,” in *Proceedings of the First International Conference on Nonlinear Problems in Aviation and Aerospace.* Embry-Riddle Aeronautical Univ. Press Daytona Beach, FL, 1996, pp. 117–130.

[137] X. Xu, J. J. Zhu, and P. Zhang, “The optimal solution of a non-convex state-dependent LQR problem and its applications,” *PloS one*, vol. 9, no. 4, p. e94925, 2014.

[138] E. D. Sontag and Y. Wang, “New characterizations of input-to-state stability,” *IEEE Transactions on Automatic Control*, vol. 41, no. 9, pp. 1283–1294, 1996.

[139] E. D. Sontag, “Further facts about input to state stabilization,” *IEEE Transactions on Automatic Control*, vol. 35, no. 4, pp. 473–476, 1990.

[140] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee, *Robotics: control, sensing, vision, and intelligence.* McGraw-Hill, Inc. New York, USA, 1987.

[141] S. R. Munasinghe, M. Nakamura, S. Aoki, S. Goto, and N. Kyura, “High speed precise control of robot arms with assigned speed under torque constraint by trajectory generation in joint co-ordinates,” *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 2, pp. 854–859, 1999.

[142] V. Mhase, R. Sudarshan, O. Pardeshi, and P. V. Suryawanshi, “Integrated speed-position tracking with trajectory generation and synchronization for 2-axis DC motion control,” *International Journal of Engineering Research and Development*, vol. 1, pp. 61–66, 2012.

[143] S. H. Chong and K. Sato, “Practical and robust control for precision positioning systems,” *IEEE International Conference on Mechatronics (ICM)*, pp. 961–966, 2011.

[144] H. Wang, “Adaptive control of robot manipulators with uncertain kinematics and dynamics.” *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 948–954, 2017.

[145] P. R. Pagilla and M. Tomizuka, “An adaptive output feedback controller for robot arms: stability and experiments,” *Automatica*, vol. 37, no. 7, pp. 983–995, 2001.

[146] B. S. Chen, T. S. Lee, and J. H. Feng, “A nonlinear $\mathcal{H}_\infty$ control design in robotic systems under parameter perturbation and external disturbance,” *International Journal of Control*, vol. 59, no. 2, pp. 439–461, 1994.

[147] A. J. Van der Schaft, “L_2-gain analysis of nonlinear systems and nonlinear state-feedback $\mathcal{H}_\infty$ control,” *IEEE Transactions on Automatic Control*, vol. 37, no. 6, pp. 770–784, 1992.

[148] S. Yin, X. Li, H. Gao, and O. Kaynak, “Data-based techniques focused on modern industry: An overview,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 1, pp. 657–667, 2015.

[149] M. Jin, S. H. Kang, P. H. Chang, and J. Lee, “Robust control of robot manipulators using inclusive and enhanced time delay control,” *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 5, pp. 2141–2152, 2017.

[150] M. N. Mahyuddin, S. G. Khan, and G. Herrmann, “A novel robust adaptive control algorithm with finite-time online parameter estimation of a humanoid robot arm,” *Robotics and Autonomous Systems*, vol. 62, no. 3, pp. 294–305, 2014.

[151] D. Zhang and B. Wei, “Design, analysis and modelling of a hybrid controller for serial robotic manipulators,” *Robotica*, vol. 35, no. 9, pp. 1888–1905, 2017.

[152] T. Yucelen and A. J. Calise, “Derivative-free model reference adaptive control,” *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 4, pp. 933–950, 2011.

[153] H. C. Ferreira, P. H. Rocha, and R. M. Sales, “Nonlinear H_∞ control and the Hamilton-Jacobi-Isaacs equation,” in *Proceedings of the 17th IFAC World Congress, Seoul, South Korea*, vol. 17, no. 1, 2008, p. 188.

[154] T. Dierks and S. Jagannathan, “Optimal control of affine nonlinear continuous-time systems,” in *American Control Conference (ACC), 2010.* IEEE, 2010, pp. 1568–1573.

[155] X. Yang, D. Liu, and D. Wang, “Reinforcement learning for adaptive optimal control of unknown continuous-time nonlinear systems with input constraints,” *International Journal of Control*, vol. 87, no. 3, pp. 553–566, 2014.