

**Direct Adaptive Control Methodologies for Flexible-Joint
Space Manipulators with Uncertainties and Modeling Errors**

by
Steve Ulrich, M.A.Sc., P.Eng.

A thesis submitted to the Faculty of Graduate and Postdoctoral Affairs
in partial fulfillment of the requirements for the degree of

**Doctor of Philosophy
in
Aerospace Engineering**

Carleton University
Ottawa, Ontario

© 2012
Steve Ulrich



Library and Archives
Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 978-0-494-94537-7

Our file Notre référence

ISBN: 978-0-494-94537-7

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

This work is dedicated to my family; to my son Jakob, to my mother Denise and late father Guy, and to my wife Judith.

Abstract

This work addresses the direct adaptive trajectory tracking control problem associated with lightweight space robotic manipulators that exhibit elastic vibrations in their joints, and which are subject to parametric uncertainties and modeling errors. Unlike existing adaptive control methodologies, the proposed flexible-joint control techniques do not require identification of unknown parameters, or mathematical models of the system to be controlled. The direct adaptive controllers developed in this work are based on the model reference adaptive control approach, and manage modeling errors and parametric uncertainties by time-varying the controller gains using new adaptation mechanisms, thereby reducing the errors between an ideal model and the actual robot system. More specifically, new decentralized adaptation mechanisms derived from the simple adaptive control technique and fuzzy logic control theory are considered in this work.

Numerical simulations compare the performance of the adaptive controllers with a nonadaptive and a conventional model-based controller, in the context of $12.6 \text{ m} \times 12.6 \text{ m}$ square trajectory tracking. To validate the robustness of the controllers to modeling errors, a new dynamics formulation that includes several nonlinear effects usually neglected in flexible-joint dynamics models is proposed. Results obtained with the adaptive methodologies demonstrate an increased robustness to both uncertainties in joint stiffness coefficients and dynamics modeling errors, as well as highly improved tracking performance compared with the nonadaptive and model-based strategies.

Finally, this work considers the partial state feedback problem related to flexible-joint space robotic manipulators equipped only with sensors that provide noisy measurements of motor positions and velocities. An extended Kalman filter-based estimation strategy is developed to estimate all state variables in real-time. The state estimation filter is combined with an adaptive composite controller, to provide a closed-loop adaptive partial state feedback control scheme for flexible-joint manipulators. In addition to noise, measurement bias is a detrimental characteristic associated with motor encoders and tachometers. Therefore, this work also demonstrates that

the state observation approach developed herein can be modified to compensate for unknown sensor biases.

Acknowledgements

My deepest gratitude goes to my advisor, Professor Jurek Z. Sasiadek, not only for his constant support and trust throughout this project, but also for encouraging me to participate in the aerospace robotics and control systems community, for presenting papers on my behalf at various conferences, and for writing several recommendation letters. I have fond memories of our many conversations we have had while sharing a cup of green tea/dry cappuccino. I will always be grateful for the influence he had, and will have, on my career, both as a mentor and friend.

I would like to thank our collaborator, Dr. Itzhak Barkana from Barkana Consulting (Israel), co-author of *Direct Adaptive Control Algorithms: Theory and Applications*, whose technical assistance was invaluable throughout this project. I am extremely fortunate to have Dr. Barkana as a mentor.

I would also like to express my appreciation to those who contributed their time and effort to serve on my examining board: Dr. Vladimir Polotski from Neptec Design Group, Professor Dan Neculescu from the University of Ottawa, as well as Distinguished Research Professor John A. Goldak, Professor Jie Liu, and Professor Abhijit Sarkar from Carleton University.

On a personal level, I want to thank my family. I wish my father had lived to see me complete my graduate work. Many thanks to my mother who gave me the greatest gift of all: the love of reading and learning. Thanks also to my son, Jakob, who I am so proud of, and who taught me unconditional love. Finally, a very special thank you to my wife for her patience, support and encouragement during my studies, but most importantly, for all her love.

This research was financially supported in part by Carleton University, through the 2011 J.Y. and E.W. Wong Research Award in Mechanical/Aerospace Engineering, the Dean of Graduate Studies Academic Excellence Scholarship, and the Graduate Scholarship. The Canadian Space Agency through the Space Awareness and Learning Program, the Canadian Armed Forces under the Honorary Colonel S. B. Lerner Memorial Educational Bursary and the Individual Learning Program, and the

Natural Sciences and Engineering Research Council of Canada under the Alexander Graham Bell Canada Graduate Scholarship CGS D3-374291-2009 also provided financial support. Professor Sasiadek's research grants, and teaching assistantships from the Department of Mechanical and Aerospace Engineering are also gratefully acknowledged.

Table of Contents

Abstract	iii
Acknowledgements	v
List of Tables	xi
List of Figures	xii
Chapter 1 Introduction	1
1.1 Motivation	1
1.2 Problem Statement	3
1.3 Previous Work	5
1.4 Thesis Objectives	13
1.5 Task-Space Benchmark Trajectory and Performance Evaluation . . .	15
1.6 Organization	17
Chapter 2 Fuzzy Logic-Based Adaptive Control	20
2.1 Introduction	20
2.2 Rigid-Joint Dynamics	21
2.3 Nonadaptive Transpose Jacobian Control Development	24
2.4 Fuzzy Adaptive Control Development	25
2.5 Simulation Results	29
2.6 Summary	30
Chapter 3 A New Class of SAC-Based Adaptive Controllers	34
3.1 Introduction	34
3.2 System and Definitions	38
3.3 Control Objective	40
3.4 Decentralized Simple Adaptive Control Development	41

3.4.1	Error Dynamics	44
3.4.2	Stability Analysis	45
3.5	Decentralized Modified Simple Adaptive Control Development	49
3.5.1	Error Dynamics	50
3.5.2	Stability Analysis	50
3.6	Application Examples	53
3.6.1	Square Manipulator Systems	53
3.6.2	Nonsquare Manipulator Systems	56
3.7	Summary	62
Chapter 4 Direct Adaptive Composite Control Methodologies		68
4.1	Introduction	68
4.2	Dynamics Modeling	71
4.2.1	Linear Joint Stiffness Dynamics	72
4.2.2	Nonlinear Joint Stiffness Dynamics	75
4.3	Control Objective	78
4.4	Composite Control Strategies Development	78
4.4.1	Nonadaptive Composite Control	82
4.4.2	Fuzzy Logic-Based Adaptive Composite Control	84
4.4.3	DSAC-Based Composite Control	95
4.4.4	DMSAC-Based Composite Control	97
4.5	Simulation Results	101
4.5.1	Robustness to Parametric Uncertainties	114
4.5.2	Robustness to Modeling Errors	124
4.5.3	Comparison to a Model-Based Flexible-Joint Controller	134
4.6	Summary	138
Chapter 5 Partial State Feedback Adaptive Composite Control		139
5.1	Introduction	139
5.2	Flexible-Joint Manipulator Estimation	142
5.2.1	Linear Joint Stiffness Dynamics	143

5.2.2	Nonlinear Joint Stiffness Dynamics	144
5.2.3	Measurement Model	145
5.2.4	Observability Analysis	146
5.3	Bias Estimation	150
5.4	Simulation Results	152
5.5	Summary	157
Chapter 6	Conclusion	161
6.1	Thesis Summary	161
6.2	Significance of Work	164
6.3	Recommendations for Future Work	166
Appendix A	Rigid Kinematics and Dynamics Equations	170
A.1	Kinematics Equations	170
A.1.1	Forward Kinematics	170
A.1.2	Velocity Kinematics - The Jacobian	172
A.1.3	Inverse Kinematics	174
A.2	Dynamics Equations	175
Appendix B	Fuzzy Logic MATRIXx Code	181
B.1	Initialization File	181
B.2	Adaptation Mechanism Function	183
Appendix C	DSAC/DMSAC for Square Flexible-Joint Manipulators	187
C.1	Control Objective	187
C.2	DSAC-Based Composite Control	188
C.3	DMSAC-Based Composite Control	188
C.4	Simulation Results	188
Appendix D	Nonlinear Backstepping Control	205
D.1	Nonlinear Backstepping Control Development	205
D.2	Stability Analysis	209

Appendix E	Extended Kalman Filter Theory	215
Appendix F	Extended Kalman Filter MATRIXx Code	219
F.1	State Estimation Function	219
F.2	State Derivative Calculation Function	224
Bibliography		226
Biographical Sketch		244

List of Tables

2.1	Fuzzy logic system rule table	28
4.1	Fuzzy logic system rule table for $\tilde{\lambda}_1$ and $\tilde{\lambda}_2$	92
4.2	Fuzzy logic system rule table for $\tilde{\lambda}_3$ and $\tilde{\lambda}_4$	93
4.3	Nonminimal linear joint stiffness manipulator characteristics . . .	101
4.4	Nonlinear joint stiffness manipulator characteristics	125

List of Figures

1.1	Typical robotic manipulation tasks (pictures courtesy of NASA). (a) Space Shuttle Remote Manipulator System used as a platform for an extravehicular activity to repair the Hubble Space Telescope. (b) Space Station Remote Manipulator System servicing the International Space Station.	3
1.2	Desired end-effector square trajectory in the task space, illustrating the motion of the two-link manipulator in a counterclockwise direction starting from rest at the lower-right-hand corner.	16
1.3	Desired end-effector position and velocity as a function of time illustrating abrupt changes in direction at $t = 15$ sec., 30 sec. and 45 sec.	16
2.1	Schematic of the two-link robot manipulator	23
2.2	Membership functions for normalized input variables	27
2.3	Membership functions for normalized output variables	27
2.4	Trajectory tracking results obtained with the nonadaptive transpose Jacobian controller applied to the rigid-joint manipulator. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$	31
2.5	Trajectory tracking errors ($y_e(t) = y_d(t) - y(t)$) obtained with the nonadaptive transpose Jacobian controller applied to the rigid-joint manipulator.	31
2.6	Trajectory tracking results obtained with the fuzzy adaptive controller applied to the rigid-joint manipulator. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$	32

2.7	Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the fuzzy adaptive controller applied to the rigid-joint manipulator.	32
2.8	Adaptation history of the fuzzy controller gains $K_p(t)$ and $K_d(t)$ when applied to the rigid-joint manipulator.	33
3.1	Trajectory tracking results obtained with the DMSAC controller applied to the square rigid-joint manipulator. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$.	57
3.2	Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the DMSAC controller applied to the square rigid-joint manipulator.	57
3.3	Adaptation history of the DMSAC controller gains $K_e(t)$ when applied to the square rigid-joint manipulator.	58
3.4	Trajectory tracking results obtained with the DSAC controller applied to the square rigid-joint manipulator. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$.	59
3.5	Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the DSAC controller applied to the square rigid-joint manipulator.	59
3.6	Adaptation history of the DSAC controller gains $K_e(t)$ when applied to the square rigid-joint manipulator.	60
3.7	Adaptation history of the DSAC controller gains $K_{x_p}(t)$ and $K_{x_d}(t)$ when applied to the square rigid-joint manipulator. . .	61
3.8	Adaptation history of the DSAC controller gains $K_{u_p}(t)$ and $K_{u_d}(t)$ when applied to the square rigid-joint manipulator. . .	61

3.9	Trajectory tracking results obtained with the DMSAC controller applied to the nonsquare rigid-joint manipulator. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$	63
3.10	Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the DMSAC controller applied to the nonsquare rigid-joint manipulator.	63
3.11	Adaptation history of the DMSAC controller gains $K_{e_p}(t)$ and $K_{e_d}(t)$ when applied to the nonsquare rigid-joint manipulator.	64
3.12	Trajectory tracking results obtained with the DSAC controller applied to the nonsquare rigid-joint manipulator. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$	65
3.13	Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the DSAC controller applied to the nonsquare rigid-joint manipulator.	65
3.14	Adaptation history of the DSAC controller gains $K_{e_p}(t)$ and $K_{e_d}(t)$ when applied to the nonsquare rigid-joint manipulator.	66
3.15	Adaptation history of the DSAC controller gains $K_{x_p}(t)$ and $K_{x_d}(t)$ when applied to the nonsquare rigid-joint manipulator.	66
3.16	Adaptation history of the DSAC controller gains $K_{u_p}(t)$ and $K_{u_d}(t)$ when applied to the nonsquare rigid-joint manipulator.	67
4.1	Flexible joint modeling	73
4.2	Block diagram representation of the composite control design methodology applied to flexible-joint manipulators, where τ_s controls the rigid dynamics and τ_f improves damping of the elastic vibrations at the joints.	79

4.3	Block diagram representation of the nonadaptive transpose Jacobian composite controller.	83
4.4	Block diagram representation of the fuzzy adaptive composite controller.	86
4.5	Computation details of the fuzzy adaptive control gain $K_e(t)$	87
4.6	Membership functions for normalized input variables u_1, u_2, u_3 and u_4	89
4.7	Membership functions for normalized output variables λ_1 and λ_2	90
4.8	Membership functions for normalized output variables λ_3 and λ_4	90
4.9	Adaptation of λ_1 and λ_2 (a) Three-dimensional nonlinear adaptation surface (b) Contour map of the adaptation surface	96
4.10	Nonlinear adaptation curve for λ_3 and λ_4	97
4.11	Block diagram representation of the DSAC composite controller.	99
4.12	Block diagram representation of the DMSAC composite controller.	100
4.13	Trajectory tracking results obtained with the nonadaptive transpose Jacobian composite controller applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m). The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$	107
4.14	Trajectory tracking errors ($y_e(t) = y_d(t) - y(t)$) obtained with the nonadaptive transpose Jacobian composite controller applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m).	107
4.15	Trajectory tracking results obtained with the fuzzy adaptive composite controller applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m). The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$	108

4.16	Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the fuzzy adaptive composite controller applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m).	108
4.17	Adaptation history of the fuzzy controller gains $K_p(t)$ and $K_d(t)$ when applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m).	109
4.18	Trajectory tracking results obtained with the DSAC composite controller applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m). The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$	110
4.19	Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the DSAC composite controller applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m).	110
4.20	Adaptation history of the DSAC composite controller gains $K_{e_p}(t)$ and $K_{e_d}(t)$ when applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m).	111
4.21	Adaptation history of the DSAC composite controller gains $K_{x_p}(t)$ and $K_{x_d}(t)$ when applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m).	111
4.22	Adaptation history of the DSAC composite controller gains $K_{u_p}(t)$ and $K_{u_d}(t)$ when applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m).	112
4.23	Trajectory tracking results obtained with the DMSAC composite controller applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m). The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$	113

4.24	Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the DMSAC composite controller applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m).	113
4.25	Adaptation history of the DMSAC composite controller gains $K_{e_p}(t)$ and $K_{e_d}(t)$ when applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m).	114
4.26	Trajectory tracking results obtained with the nonadaptive transpose Jacobian composite controller applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m). The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$.	117
4.27	Trajectory tracking errors ($y_e(t) = y_d(t) - y(t)$) obtained with the nonadaptive transpose Jacobian composite controller applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m).	117
4.28	Trajectory tracking results obtained with the fuzzy adaptive composite controller applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m). The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$	118
4.29	Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the fuzzy adaptive composite controller applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m).	118
4.30	Adaptation history of the fuzzy controller gains $K_p(t)$ and $K_d(t)$ when applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m).	119

4.31	Trajectory tracking results obtained with the DSAC composite controller applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m). The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$	120
4.32	Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the DSAC composite controller applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m).	120
4.33	Adaptation history of the DSAC composite controller gains $K_{e_p}(t)$ and $K_{e_d}(t)$ when applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m).	121
4.34	Adaptation history of the DSAC composite controller gains $K_{x_p}(t)$ and $K_{x_d}(t)$ when applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m).	121
4.35	Adaptation history of the DSAC composite controller gains $K_{u_p}(t)$ and $K_{u_d}(t)$ when applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m).	122
4.36	Trajectory tracking results obtained with the DMSAC composite controller applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m). The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$	123
4.37	Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the DMSAC composite controller applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m).	123
4.38	Adaptation history of the DMSAC composite controller gains $K_{e_p}(t)$ and $K_{e_d}(t)$ when applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m).	124

4.39	Joint stiffness and friction characteristics for the nonlinear joint stiffness dynamics model (a) Nonlinear joint stiffness curve with soft windup effect in the region around 0 deg (b) Continuous and symmetric friction curve that includes static, Stribeck, viscous, and Coulombic effects	126
4.40	Trajectory tracking results obtained with the nonadaptive transpose Jacobian composite controller applied to the nonlinear joint stiffness manipulator. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$	127
4.41	Trajectory tracking errors ($y_e(t) = y_d(t) - y(t)$) obtained with the nonadaptive transpose Jacobian composite controller applied to the nonlinear joint stiffness manipulator.	127
4.42	Trajectory tracking results obtained with the fuzzy adaptive composite controller applied to the nonlinear joint stiffness manipulator. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$	128
4.43	Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the fuzzy adaptive composite controller applied to the nonlinear joint stiffness manipulator.	128
4.44	Adaptation history of the fuzzy controller gains $K_p(t)$ and $K_d(t)$ when applied to the nonlinear joint stiffness manipulator. . . .	129
4.45	Trajectory tracking results obtained with the DSAC composite controller applied to the nonlinear joint stiffness manipulator. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$	130

4.46	Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the DSAC composite controller applied to the nonlinear joint stiffness manipulator.	130
4.47	Adaptation history of the DSAC composite controller gains $K_{e_p}(t)$ and $K_{e_d}(t)$ when applied to the nonlinear joint stiffness manipulator.	131
4.48	Adaptation history of the DSAC composite controller gains $K_{x_p}(t)$ and $K_{x_d}(t)$ when applied to the nonlinear joint stiffness manipulator.	131
4.49	Adaptation history of the DSAC composite controller gains $K_{u_p}(t)$ and $K_{u_d}(t)$ when applied to the nonlinear joint stiffness manipulator.	132
4.50	Trajectory tracking results obtained with the DMSAC composite controller applied to the nonlinear joint stiffness manipulator. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$	133
4.51	Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the DMSAC composite controller applied to the nonlinear joint stiffness manipulator.	133
4.52	Adaptation history of the DMSAC composite controller gains $K_{e_p}(t)$ and $K_{e_d}(t)$ when applied to the nonlinear joint stiffness manipulator.	134
4.53	Trajectory tracking results obtained with the nonlinear backstepping controller applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m). The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$	137

4.54	Trajectory tracking errors ($y_e(t) = y_d(t) - y(t)$) obtained with the nonlinear backstepping controller applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m).	137
5.1	Block diagram representation of the adaptive partial state feedback composite control strategy.	153
5.2	Trajectory tracking results obtained with the linear joint stiffness-based EKF / DMSAC composite controller combination applied to the linear joint stiffness manipulator. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$	155
5.3	Estimation results obtained with the linear joint stiffness-based EKF / DMSAC composite controller combination applied to the linear joint stiffness manipulator. The red lines correspond to the true link variables $q(t), \dot{q}(t)$, and the black lines correspond to the estimated link variables $\hat{q}(t), \dot{\hat{q}}(t)$	156
5.4	Estimation errors obtained with the linear joint stiffness-based EKF / DMSAC composite controller combination applied to the linear joint stiffness manipulator.	156
5.5	Trajectory tracking results obtained with the nonlinear joint stiffness-based EKF / DMSAC composite controller combination applied to the nonlinear joint stiffness manipulator. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$	157
5.6	Estimation results obtained with the nonlinear joint stiffness-based EKF / DMSAC composite controller combination applied to the nonlinear joint stiffness manipulator. The red lines correspond to the true link variables $q(t), \dot{q}(t)$, and the black lines correspond to the estimated link variables $\hat{q}(t), \dot{\hat{q}}(t)$	158

5.7	Estimation errors obtained with the nonlinear joint stiffness-based EKF / DMSAC composite controller combination applied to the nonlinear joint stiffness manipulator.	158
5.8	Trajectory tracking results obtained with the nonlinear joint stiffness-based EKF / DMSAC composite controller combination applied to the nonlinear joint stiffness manipulator. The measurements are corrupted by biases of $b_e = 2$ deg and $b_t = 2$ deg/s, but without considering them in the EKF estimator. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$	159
5.9	Trajectory tracking results obtained with the nonlinear joint stiffness-based EKF / DMSAC composite controller combination applied to the nonlinear joint stiffness manipulator. The measurements are corrupted by biases of $b_e = 2$ deg and $b_t = 2$ deg/s, which are considered in the EKF estimator. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$	160
A.1	Two-link revolute joint robot manipulator	173
C.1	Trajectory tracking results obtained with the DMSAC composite controller applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m) modeled as a square system. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$	190
C.2	Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the DMSAC composite controller applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m) modeled as a square system.	190

C.3	Adaptation history of the DMSAC composite controller gains $K_{e_p}(t)$ and $K_{e_d}(t)$ when applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m) modeled as a square system.	191
C.4	Trajectory tracking results obtained with the DSAC composite controller applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m) modeled as a square system. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$	192
C.5	Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the DSAC composite controller applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m) modeled as a square system.	192
C.6	Adaptation history of the DSAC composite controller gains $K_{e_p}(t)$ and $K_{e_d}(t)$ when applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m) modeled as a square system.	193
C.7	Adaptation history of the DSAC composite controller gains $K_{x_p}(t)$ and $K_{x_d}(t)$ when applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m) modeled as a square system.	194
C.8	Adaptation history of the DSAC composite controller gains $K_{u_p}(t)$ and $K_{u_d}(t)$ when applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m) modeled as a square system.	194
C.9	Trajectory tracking results obtained with the DMSAC composite controller applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m) modeled as a square system. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$	195

C.10	Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the DMSAC composite controller applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m) modeled as a square system.	195
C.11	Adaptation history of the DMSAC composite controller gains $K_{e_p}(t)$ and $K_{e_d}(t)$ when applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m) modeled as a square system.	196
C.12	Trajectory tracking results obtained with the DSAC composite controller applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m) modeled as a square system. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$. 197	197
C.13	Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the DSAC composite controller applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m) modeled as a square system.	197
C.14	Adaptation history of the DSAC composite controller gains $K_{e_p}(t)$ and $K_{e_d}(t)$ when applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m) modeled as a square system.	198
C.15	Adaptation history of the DSAC composite controller gains $K_{x_p}(t)$ and $K_{x_d}(t)$ when applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m) modeled as a square system.	199
C.16	Adaptation history of the DSAC composite controller gains $K_{u_p}(t)$ and $K_{u_d}(t)$ when applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m) modeled as a square system.	199

C.17	Trajectory tracking results obtained with the DMSAC composite controller applied to the nonlinear joint stiffness manipulator modeled as a square system. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$	200
C.18	Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the DMSAC composite controller applied to the nonlinear joint stiffness manipulator modeled as a square system.	200
C.19	Adaptation history of the DMSAC composite controller gains $K_{e_p}(t)$ and $K_{e_d}(t)$ when applied to the nonlinear joint stiffness manipulator modeled as a square system.	201
C.20	Trajectory tracking results obtained with the DSAC composite controller applied to the nonlinear joint stiffness manipulator modeled as a square system. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$	202
C.21	Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the DSAC composite controller applied to the nonlinear joint stiffness manipulator modeled as a square system.	202
C.22	Adaptation history of the DSAC composite controller gains $K_{e_p}(t)$ and $K_{e_d}(t)$ when applied to the nonlinear joint stiffness manipulator modeled as a square system.	203
C.23	Adaptation history of the DSAC composite controller gains $K_{x_p}(t)$ and $K_{x_d}(t)$ when applied to the nonlinear joint stiffness manipulator modeled as a square system.	204
C.24	Adaptation history of the DSAC composite controller gains $K_{u_p}(t)$ and $K_{u_d}(t)$ when applied to the nonlinear joint stiffness manipulator modeled as a square system.	204
E.1	Implementation details of EKF estimator.	218

Chapter 1

Introduction

1.1 Motivation

Ever since the first deployment of the Shuttle Remote Manipulator System (SRMS)—the 15.2 meter long, six degrees of freedom (DOF) robotic manipulator known as *Canadarm*—from the cargo bay of the Space Shuttle Columbia in 1981, robotic systems have been used on most space missions, and are also employed on the International Space Station (ISS). Due to the advances in space robotics, some manipulators can move even more freely than human arms, and are therefore well suited to support, or even replace, astronauts for accomplishing precise, complex or risky manoeuvres. For example, with the addition of the two-arm robot Special Purpose Dexterous Manipulator (SPDM) to the Space Station Remote Manipulator System (SSRMS), fine assembly tasks that were previously performed by astronauts during spacewalks are now carried out by highly advanced robotic systems [1]. In addition, robotic manipulators are well suited to execute highly repetitive tasks that would be too time-consuming and expensive if performed by astronauts. Also, space robots require much less infrastructure than humans (e.g. life support systems), which makes them important to the overall ability to operate in space [2]. For these reasons, robotic manipulators are attractive for several types of space missions, particularly on-orbit servicing missions. In fact, the National Aeronautics and Space Administration (NASA) realized the importance of on-orbit servicing to protect their assets in space as early as the 1980s. On-orbit servicing refers to the maintenance of space systems in orbit, including the maintenance, repair, assembly, refueling and/or upgrade of spacecraft after their deployment. An exhaustive list of possible applications of robotic on-orbit servicing can be found in [3] and [4].

However, to date, typical manipulation tasks have been restricted to the assembly of space structures or satellite servicing missions, such as maintenance of the Hubble

Space Telescope (HST). The HST mission, a unique example of an unmanned spacecraft designed to be regularly serviced, highlights the significant benefits of developing an on-orbit servicing infrastructure for space systems. One of the most remarkable HST missions was the first servicing mission, intended to correct an optical flaw in the primary mirror of the telescope that was causing blurred observations. The telescope would have been inoperable without this repair. The HST is also an example of the benefits possible with space robotics. Many instruments were replaced to take advantage of new technologies, providing the scientific community with an up-to-date telescope at less cost than if the spacecraft had to be replaced.

Besides the HST, most of the space robotic technology has been driven by the ISS program requirements. The SRMS was the first general-purpose robotic manipulator designed specifically for use in space; it has been on over 70 shuttle flights and retrieved more than a dozen satellites [5]. The SRMS was also the first robotic manipulator to demonstrate the principle of robotic servicing. In addition to the SRMS, a number of cargo-handling manipulators designed to operate on the ISS have been developed by international space agencies. The main ISS robotic system, known as the ISS Mobile Servicing System, comprises the SSRMS, the SPDM and the Mobile Remote Servicer Base System, which was jointly developed by Canada and NASA and acts as a movable platform for the SSRMS and SPDM. The Japanese Experiment Module Remote Manipulator System (JEMRMS), built by the Japan Aerospace Exploration Agency, is a robotic manipulator system for supporting experiments conducted on the Japanese Experiment Module (JEM) Exposed Facility. The European Robotic Arm (ERA), developed by the European Space Agency, is an 11 meter manipulator with seven DOF and two links, which is attached to the Russian segment of the ISS. Originally designed for the deployment and retrieval of large payloads, these cargo-handling robotic systems have proven their versatility, from serving as platforms for extravehicular activities (EVAs) to performing inspection operations.

Unfortunately, the manned on-orbit servicing currently performed on the ISS is prohibitively expensive, and is not a viable architecture for future space missions. The control of these cargo-handling robotic arms is not yet fully autonomous; human assistance is still required to accurately position the end-effector of the manipulator.

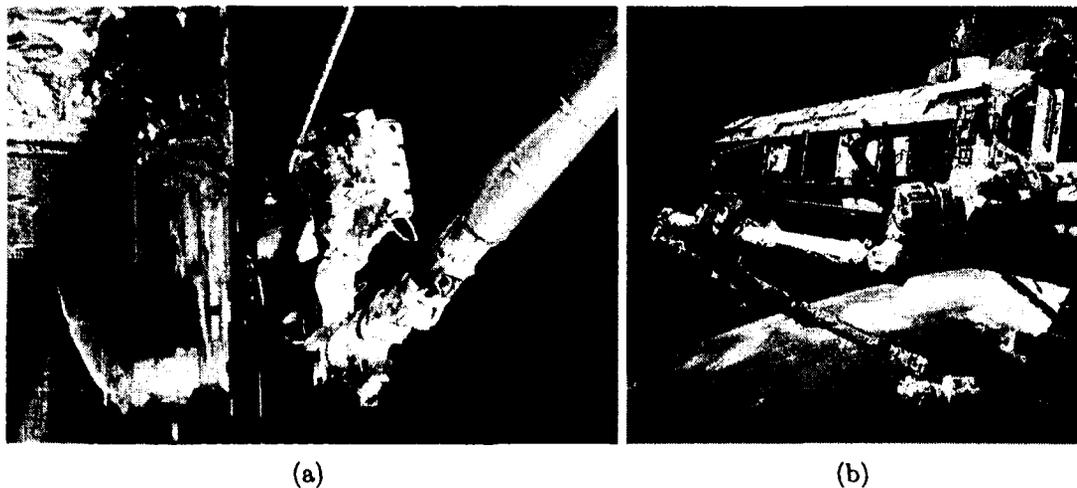


Figure 1.1: Typical robotic manipulation tasks (pictures courtesy of NASA). (a) Space Shuttle Remote Manipulator System used as a platform for an extravehicular activity to repair the Hubble Space Telescope. (b) Space Station Remote Manipulator System servicing the International Space Station.

To date, control of space robots is either achieved locally from a control console, or remotely by ground operators on Earth, which can involve significant time delays [2]. For example, operational control of ERA is performed by astronauts, either through an EVA man-machine interface, or via an intravehicular activity man-machine interface when the operator is inside the Russian Service Module. The JEMRMS control console is located inside the JEM Pressurized Module, and provides either manual or semi-autonomous control through pre-programmed trajectories [6]. In fact, astronauts require specialized training to operate the robotic arms with hand controllers or keyboards, and the procedure often results in overcorrections until the end-effector of the robotic arm is correctly positioned. The accuracy of the positioning of the end-effector is thus dependent on the ability and dexterity of the astronauts. As explained in [6], such extended duration manipulations are delicate operations, and can take more than seven hours.

1.2 Problem Statement

The growing research interest in space robotic systems capable of accurately performing manipulation tasks within an acceptable execution time without human assistance

has led to an increased demand for lightweight robot arms with load-to-weight ratios comparable to that of the human arm. As stated by Reintsema *et al.* [7], ongoing development of space robotics technology must focus on intelligent, sensor-controlled lightweight robots. One well-known problem of such lightweight space manipulators in space applications is the increased structural flexibility, and various control strategies have been proposed in the literature to minimize the resulting vibrations [8–15]. In addition to link flexibility, the benefits of lightweight space robotic manipulators are gained at the expense of higher elasticity in the joints, which is a major drawback of large manipulators [16]. Indeed, joint flexibility is often considered more important than link flexibility, at least in the operational range of space robots [17], and in most cases, this is the limiting factor to achieving satisfactory performance [18]. As a result, accelerating and stopping the arm produces large vibrations, making precise positioning of the end-effector very difficult. Although current space robots with planetary gear trains, such as the SSRMS, can exhibit considerable joint elasticity effects [19, 20], these flexible behaviors are particularly important in lightweight manipulators equipped with harmonic drives [7], such as the Lightweight Robot III developed by the German Aerospace Center (DLR). Harmonic drive joint mechanisms are becoming increasingly popular for space applications, due to their low weight, compactness, high torque capabilities, wide operating temperature range and reliable repeatability. However, the flexibility effects of this type of gear mechanism in the joints of robotic manipulators are significant enough to make real-time operation challenging, particularly when accurate end-effector positioning is required [21]. Joint flexibility effects are also significant for manipulators where elastic bands, or motor with long shafts are used. Neglecting joint elasticity effects in control system design can also lead to instability [22]. The impacts of joint flexibility on the control system performance of space robotic manipulators were highlighted by Cetinkunt and Book [23]. Also, the additional flexible dynamics introduce two more state variables for each joint, which means the complete description of the dynamics model (including joint elasticity) requires four variables for each joint: position and velocity of the motor, and position and velocity of the link. Therefore, since the flexible-joint

dynamics do not have an independent control input for each DOF, most control algorithms designed for rigid-joint robots do not extend to the flexible-joint case, as explained by Ghorbel *et al.* [24]. Elastic vibration of the joints, coupled with their large rotations and nonlinear dynamics, can also induce structural flexibility of the links, as discussed by Sasiadek [25]. The flexible-joint problem is further aggravated in the case of multi-link robots with several joints. For these reasons, control systems design for flexible-joint space manipulators is a difficult and challenging problem.

Thus, achieving accurate trajectory tracking motion and dampening elastic vibrations represent the most important control problems for this class of manipulators. The performance requirement for this problem is typically very high. For example, in some applications it may be necessary to quickly move the end-effector of the manipulator to a new position. This must be done in the least possible time, while keeping the elastic motion and accompanying trajectory tracking errors within acceptable limits. Consequently, the success of future robotic on-orbit servicing missions is strongly dependent on the design and implementation of advanced flexible-joint control strategies, which can compensate for weakly damped elasticity in the robot joints and achieve high performance motion control [26]. The general objective of this work is to address the flexible-joint control problem for space robotic manipulators.

1.3 Previous Work

The section presents a detailed survey of past and present research related to the problem of flexible-joint control. The aim of this section is to lay a foundation for the remainder of the thesis by identifying what has been done, in order to propose innovative control methodologies to fill the knowledge gap identified in the published literature.

Detailed literature surveys in specific areas of flexible-joint control, i.e. the backstepping control approach and the global tracking control approach, have been presented by Bridges *et al.* [27] and Brogliato *et al.* [28], respectively. Wide survey papers have also been published, namely by Troch and Kopacek [29], Spong [30], and more recently by Ozgoli and Taghirad [31]. While the first two are now outdated, the most recent one contains many errors and inaccuracies.

Early Control Approaches

The basic problem of trajectory tracking of robot manipulators with elastic joints is not new. Indeed, ever since the first clear demonstration in 1984 by Sweet and Good [16] that robot manipulators were exhibiting flexible effects in their joints, new control strategies developed specifically for this new class of robot systems started to emerge. One of the first studies on flexible-joint control was published in 1986 by Tomei *et al.* [32], in which an adaptive controller based on an approximate linearized model of the flexible-joint dynamics was proposed. With this linear dynamics approximation, the Coriolis and Centrifugal forces were namely neglected. Using a similar linearized dynamics representation, Kuntze [33] designed a predictive controller, but the performance was not satisfying. By this time, most of the research efforts were focused on linear control methodologies. In 1986, for the first time, a nonlinear control method was considered by Marino and Spong [34] for a single-link flexible-joint manipulator. In their work, the authors proved that a single-link flexible-joint robotic manipulator under static nonlinear state feedback acts like a controllable linear system.

Proportional-Integral-Derivative Control

Due to their simplicity and ease of implementation, proportional-derivative (PD) controllers, which consist of a generic control loop feedback mechanism, have been widely used. Tomei [35] proposed a simple PD regulator for flexible-joint robots, similar to that used for rigid robots. Simulation results for a regulation problem about a reference position are provided. An adaptive version of this simple PD regulator was derived by Tomei [36]. For the sake of simplicity, actuators dynamics and friction were neglected in the control algorithms in those two PD regulation approaches. An extension of the PD regulator for robot manipulator considering simultaneously joint flexibility, actuators dynamics as well as friction was presented by Lozano *et al.* [37]. A PD controller with on-line gravity compensation was proposed for regulation tasks by De Luca *et al.* [38]. One of the features of this controller is to allow more flexibility in the tuning of the proportional gains, as compared with the original controller proposed by Tomei [35].

Integral Manifold

The concept of integral manifold control is similar to the idea of singular perturbation-based control [39], in which a fast control component to damp the fast dynamics is added to a slow control component. The main difference between the two approaches is that with the integral manifold technique, the slow control term contains additional *corrective* terms. For this reason, the resulting integral manifold-based control laws are often referred to as *corrective* control laws. Early references which dealt with flexible-joint control approaches in the context of integral manifolds include the work of Khorasani and Spong [40], Khorasani and Kokotovic [41], and Spong *et al.* [22,42]. The most important deficiency of the original integral manifold approach is its lack of robustness of parametric uncertainties, as explained by Ghorbel *et al.* [24]. For this reason, an extension to an adaptive integral manifold technique has been proposed by Ghorbel and Spong [43].

Feedback Linearization

Feedback linearization is one of the most powerful approach for nonlinear control system design. Essentially, the objective of this approach is to transform a nonlinear dynamic system into a (fully or partly) linear model, so that linear control techniques can be applied. Feedback linearization of flexible-joint robots was first investigated by Cesareo and Marino [44] by using an algorithm which, given the configuration (spatial geometry, mass distribution, structural parameters), allows the symbolic generation of the equations of motion and the analysis of their controllability properties. Then, in 1987, Spong [45] introduced the well-known linear joint stiffness dynamics model and showed that it is globally feedback linearizable, and can be reduced to that of the standard rigid-joint robot as the joint stiffness values reach infinity; properties which naturally led to the early development of several feedback linearization control approaches, namely by Nicosia and Tomeo [46] and De Luca [47]. Lin and Yuan [48] considered an external linearization scheme combining a linear quadratic regulator (LQR) controller with a servo compensator to actively damp out the elastic vibrations and yield a complete closed-loop system that is robust to parametric uncertainties. De Luca [49] and De Luca and Lucibello [50] suggested dynamic state feedback control techniques for a particular class of flexible-joint robots for which conventional static state feedback approaches failed to achieve exact linearization and input output

decoupling. Recently, a static state feedback linearization approach was applied by Ge *et al.* [51] to a class of flexible-joint robots to guarantee global asymptotic stability of the closed-loop system. Furthermore, linearization with static and dynamic state feedbacks was also investigated by De Luca [52] for robots with mixed flexible/rigid joints. More recently, De Luca *et al.* [53] extended the classic feedback linearization problem to include the effects of joint damping in the dynamics representation. Palli *et al.* [54] considered feedback linearization for robotic manipulators with variable joint stiffness.

Optimal Control

Optimal control, which consists in the process of determining the control input and state trajectories for a dynamic system over a period of time to minimise a performance index, has also generated interest among the flexible-joint control community. An optimal nonlinear position tracking controller for a two-link flexible-joint manipulator was proposed by Lahdhiri and ElMaraghy [55]. The controller was designed using the concept of feedback linearization and linear quadratic Gaussian (LQG). In their work, a new approach for computing the state estimate was presented and measurement noise was considered. Recently, Consolini *et al.* [56] proposed a linear programming algorithm to compute the globally optimal minimum-time control for rest-to-rest constrained transitions with flexible-joints robots. Taking into account both input and output constraints, the optimal bang-bang control was computed by discretizing a continuous-time joint model and by solving a sequence of linear programming feasibility problems. Another class of controllers which belongs to optimal control is predictive control. In 2008, Merabet and Gu [57], applied a predictive control method with a disturbance observer to a single-link flexible-joint robot. The controller was shown to be robust with respect to modeling errors and to give good tracking performance despite external disturbances.

Adaptive Control

Among the different flexible-joint control techniques available in the literature, adaptive control methods have generated the greatest interest. Mrad and Ahmad [58] proposed an adaptive control scheme which was applicable to manipulators with low

joint stiffness coefficients. The suggested scheme was derived from a candidate Lyapunov function to guarantee asymptotic stability, and bounded parameter errors. Similarly, Lozano and Brogliato [59] presented an adaptive control scheme for which asymptotic stability was ensured regardless of the joint flexibility value, i.e. the results were not restricted to weak joint elasticity. Furthermore, the values of the joint stiffness coefficients were not assumed to be known. Joint position and velocity tracking errors were shown to asymptotically converge to zero. Similarly, Yuan and Stepanenko [60] derived an adaptive control law taking into account uncertainties in the inertia parameters. The asymptotic stability of the adaptive controller was established in the Lyapunov sense. Nicosia and Tomei [61] proposed an adaptive trajectory tracking control law for a single-link robot with one revolute elastic joint whose parameters were assumed to be unknown. Recently, Kim and Lee [62] derived a tracking controller for an uncertain parameter robot system without using link and actuator velocity measurements. Specifically, the controller used joint stiffness matrix estimates to overcome the overparametrization problem, that is, the number of parameter estimates becomes the same as the number of unknown parameters. An adaptive output feedback controller based on a backstepping design was developed by Yim [63]. The parameters of the system were assumed to be unknown and only link and motor position measurements were used for the synthesis of the controller. Fu and Cheng [64] presented a Lyapunov-based adaptive control technique framework for flexible-joint robot with time-varying uniformly bounded disturbances, where the disturbances were characterized by a combination of unknown and known constants. In 2008, a study on the regulation problem of flexible-joint robot with uncertain kinematics was proposed by Liu *et al.* [65]. Their proposed adaptive regulation scheme considered kinematic uncertainties as well as uncertainties in actuator dynamic parameters, such as the constant joint stiffness coefficients. Simulation results were provided to confirm the effectiveness of the proposed adaptive regulator. However, the adaptive control scheme required a nonlinear observer which complexified greatly the design process.

Robust Control

Besides adaptive control techniques, several robust control techniques have been proposed in the literature to cope with uncertainties, such as the robust controller derived by Tomei [66] for elastic-joint robots with uncertain kinematic and dynamic parameters and subject to unknown disturbances. Similarly, Dawson *et al.* [67] presented a robust tracking controller that achieves global uniform ultimate bounded tracking despite the presence of bounded disturbances and parameter uncertainties. Similarly, Jain and Khorrami [68] proposed a robust controller for a flexible-joint robot with both parametric and dynamic uncertainties. However, although rejection of any bounded disturbances entering the system was demonstrated, asymptotic stability could not be guaranteed. Elmaraghy *et al.* [69] presented the development of a robust linear controller for a realistic flexible-joint robot which took into account friction, nonlinear spring force characteristics and measurement noise. The robust controller was designed using LQG techniques and L1 and L2 gains. In their work, extensive experimental tests were performed using several trajectories. The obtained results demonstrated that the proposed model/controller provided good tracking and regulation performance. More recently, a robust controller was designed by Taghirad and Rahimi [70] using the singular perturbation-based dynamics representation of a flexible-joint robot, under which the system acts on a two time-scale. In their work, a simple PD controller was used to stabilize the fast dynamics while a robust controller designed with the quantitative feedback theory was used in addition to an integral manifold corrective term to stabilize the slow dynamics. Taghirad and Shaterian [71] designed a nonlinear controller which combined an H_∞ control term with an inverse dynamics control term. Simulation results for single and multiple-joint manipulators showed that the proposed controller achieved satisfactory regulation and tracking performance. Moghaddam and Goldenberg [72] analyzed nonlinear properties of harmonic drives that cause joint flexibility based on previous experiments. The nonlinear dynamics effects were then used in a robust H_∞ -based control design process. The design procedure to compensate for the nonlinear effects included a link-level motion control and an actuator-level torque control. Yim *et al.* [73, 74] proposed a robust controller using a recursive design method which contained two steps. First, a fictitious H_∞ robust controller for the link dynamics was designed as if the link dynamics

could be controlled independently of the motor dynamics. Second, a robust controller was designed recursively by using the Lyapunov second method. The resulting robust controller was applied to a two-link flexible-joint robot. Similarly, a robust H_∞ control method which satisfies the L2 gain property was proposed by Lee *et al.* [75]. Also, Yeon and Park [76] presented the design of a robust control scheme which comprised two parts: a model-based computed torque control part, and robust control part to maintain the tracking performance using an H_∞ controller. The resulting robust controller was applied in numerical simulations to a six DOF flexible-joint robot manipulator.

Nonlinear Control

Many researchers have proposed various nonlinear control design techniques for flexible-joint robot manipulators. Some early nonlinear control techniques include a nonlinear dynamic feedback developed by De Luca *et al.* [77] and a feedback linearization controller based on diffeomorphic coordinate transformations by Khorasani [78]. A nonlinear backstepping approach was proposed by Oh and Lee [79]. However, the drawback of the backstepping control technique is that the closed-loop system becomes easily unstable if the system is subject to parametric uncertainties. To overcome this limitation, an adaptive backstepping method using tuning function was developed by Macnab *et al.* [80]. In their work, the performance was assessed against a simple PD controller in tracking a 15 cm \times 15 cm square trajectory. Lee *et al.* [81] proposed a robust backstepping control method for robot manipulators with flexible joints which used the nonlinear H_∞ and the saturation control method. In each design step, the robust input was designed by satisfying the L2-gain property. An advantage of this approach is that it does not require any acceleration measurements. A nonlinear regulator for flexible-joint robots which achieved constant torque disturbance rejection was derived by Abouelsoud [82]. The design of the regulator used the nonlinear integrator backstepping technique. Lyapunov-based nonlinear controllers have also been considered in the past years. Nicosia and Tomei [83] investigated a nonlinear control law which guarantees global convergence either for regulation about a constant reference position or for tracking a desired trajectory. The proposed nonlinear control law was applied to a single-link robot. In addition, it was shown how the approach could

be extended to cases where the robot parameters are not known. Brogliato *et al.* [84] experimentally implemented seven different schemes on a two-link planar manipulator: a PD controller, a rigid-joint adaptive controller, a singular perturbation-based algorithm, and four nonlinear controllers derived from the nonlinear backstepping and energy-shaping design methodologies. One of the main conclusions of this study was that the complex structure of certain nonlinear controllers for flexible-joint robots (e.g. nonlinear backstepping and energy-shaping schemes) may be an obstacle to their closed-loop behaviour enhancement. The experimental results clearly showed that the inputs of such schemes chatter and result in control input magnitudes larger than the simpler controllers. Moreover, the authors did not succeed in finding feedback gains to stabilize the nonlinear backstepping controllers. According to the authors, this is due to a too-large input magnitude that hampers stabilization. This confirms the observations made by Brogliato *et al.* [28] and showed that a slight modification of the backstepping controller gains has a significant influence on the obtained performance. Brogliato and Rey [85] performed further experimentations on a linear but highly flexible manipulator system made of two pulleys linked by a spring. This second experiment completed the previous one by Brogliato *et al.* [84] in the sense that the manipulator used in the first experiment was nonlinear but with higher joint stiffness, whereas the system in the second experiment was linear but with highly flexible joints.

Iterative Control

Another class of control technique which was successfully used to improve the tracking accuracy for flexible-joint robot manipulators is the iterative learning control methodology. For accurate tracking control performance, iterative learning control methods are namely useful to overcome the lack of knowledge on robot dynamics and joint flexibility. The design of iterative control schemes is based on the fact that most robots perform repetitive tasks. This is done by using an iterative learning law as the manipulation task is repeated over time. Wang [86] used such learning laws to guarantee the convergence of the link trajectory. The learning law was computed off-line using link position, velocity and acceleration tracking errors. Also, Fu and Sinha [87] proposed a learning law based on neural network, which was shown to

reduce the tracking errors after a few repetitions.

1.4 Thesis Objectives

Despite the considerable research that has been devoted to the dynamics and control of flexible-joint robots, the topic remains an important area of contemporary research, as noted by Albu-Schäffer and Hirzinger [88], and Ozgoli and Taghirad [31], and several constraints and limitations have yet to be overcome.

Most existing flexible-joint control strategies reported in the literature and reviewed in the previous section (e.g. nonlinear feedback control, feedback linearization, optimal control and robust control) are model-based techniques, and have reasonably good tracking performance only when substantial knowledge of the plant mathematical model and its parameters is available. Consequently, if significant or unpredictable plant parameter variations arise as a result of joint mechanism degradations, for example, or if there are modeling errors due to complex flexible dynamics behaviors, model-based control approaches could perform inadequately.

Moreover, it is common practice to specify the desired end-effector trajectory in the task space (i.e. in terms of Cartesian coordinates) particularly when the desired trajectory requires precise control of end-effector motion. In this case, joint-space control schemes are usually not suitable, though most existing flexible-joint controllers are based on joint-space schemes. Joint-based control schemes refer to controllers in which the feedback is based on joint angular positions and velocities. With task-space control schemes, however, the actual Cartesian description of the end-effector, which may be directly measured or computed from joint positions and velocities by means of the direct kinematic equations, is compared with the desired Cartesian trajectory in order to form errors in the task space. Thus, task-space control schemes are those driven by Cartesian position and velocity errors, rather than joint position or velocity errors. In addition to rendering improved trajectory tracking performance, the main practical advantage of Cartesian-based controllers is that the solution of the inverse kinematics and inverse Jacobian, required to transform the desired end-effector position and velocity into joint position and velocity, is removed. One of the most popular task-space controllers is the transpose Jacobian rigid-joint control law,

developed by Craig [89]. However, selecting the control gains is a problem with this rigid-joint controller, and tuning involves a tedious trial-and-error procedure. It was also observed that, depending on the desired Cartesian trajectory and the actual manipulator parameters, retuning of the gains is required; a severe drawback of this controller, and unacceptable in practical applications.

These challenges, constraints and limitations of existing control methodologies motivate research on task-space *robust* flexible-joint controllers; that is, Cartesian position and velocity errors-driven controllers that maintain good trajectory tracking performance regardless of modeling errors and parametric uncertainties. To achieve this, model-independent, direct adaptive control methodologies—in which the controller parameters are adapted in real-time without requiring identification of unknown plant parameters or mathematical models of the system to be controlled—could be employed. More specifically, new controller parameters direct adaptation mechanisms based on fuzzy logic [90], as well as derived from the simple adaptive control (SAC) theory [91], are considered in this work.

Another deficiency of existing control schemes for flexible-joint manipulators is that they have been validated in numerical simulations using the common Spong model [45]. Considered the centerpiece of most work in the area of flexible-joint control, this basic dynamics formulation assumes that each joint is modeled as a linear torsional spring of constant stiffness. However, experimental studies [37, 92–97] have revealed that joint mechanisms are far more complex than linear springs, and that they exhibit highly nonlinear effects. Thus, the development of a new comprehensive flexible-joint dynamics model to better capture nonlinear effects observed in experiments is investigated in this work.

Finally, most advanced trajectory tracking flexible-joint control laws are full state feedback control techniques. In other words, they require knowledge of all state variables (i.e. link positions and velocities, and motor positions and velocities). This is difficult to achieve in lightweight flexible-joint space manipulators, since link positions and velocities are typically not measured. Hence, there is a need to develop innovative strategies that reconstruct all state variables based only on noisy and biased measurements from motor encoders and tachometers in flexible-joint manipulator

systems. Such a partial state feedback control scheme is researched in this work.

To summarize, this work addresses the direct adaptive trajectory-tracking control problem associated with flexible-joint space manipulators with parametric uncertainties and modeling errors. In addition, dynamics modeling and partial state feedback control issues will be explored.

1.5 Task-Space Benchmark Trajectory and Performance Evaluation

The modeling, simulation and performance assessment of all estimation and control strategies developed in this work will be done via numerical simulations, using MATRIXx, National Instruments' software suite for model-based control design. To validate the performance of the proposed control systems, the desired end-effector trajectory is a $12.6 \text{ m} \times 12.6 \text{ m}$ square trajectory, which is tracked in 60 seconds with a constant velocity (i.e. no deceleration at the corners) in a counter-clockwise direction, starting from rest at the lower-right-hand corner. The shoulder joint of the manipulator coincides with its fixed (i.e. stabilized) base, located at the center of the square trajectory. Compared with other types of Cartesian trajectories, such as straight lines or circles, square trajectories represent greater control challenges, as each corner of the square entails an abrupt change in direction. This makes square trajectories ideal for studying the control of transient vibrations in flexible systems; in the literature they are commonly used to assess the trajectory-tracking performance of robotic manipulators [12, 15, 80, 98, 99]. The speed of the commanded square trajectory (15 seconds for each side of the square) is fast enough to render the nonlinearities and flexibility effects significant. As a comparison, in Carusone *et al.* [11], a $75 \text{ cm} \times 75 \text{ cm}$ square trajectory is tracked in 40 seconds, and in Macnab *et al.* [80], a $15 \text{ cm} \times 15 \text{ cm}$ square trajectory is tracked in 60 seconds. For the reasons outlined in the previous section, this desired trajectory will be specified directly in terms of end-effector Cartesian position and velocity (i.e. in the task space). The resulting square trajectory in the task space, and related time histories of the desired end-effector position and velocity, are depicted in Figs. 1.2 and 1.3 respectively.

For a square trajectory as specified in the task space, the end-effector of the manipulator system must reach each corner, and redirect itself in an orthogonal direction

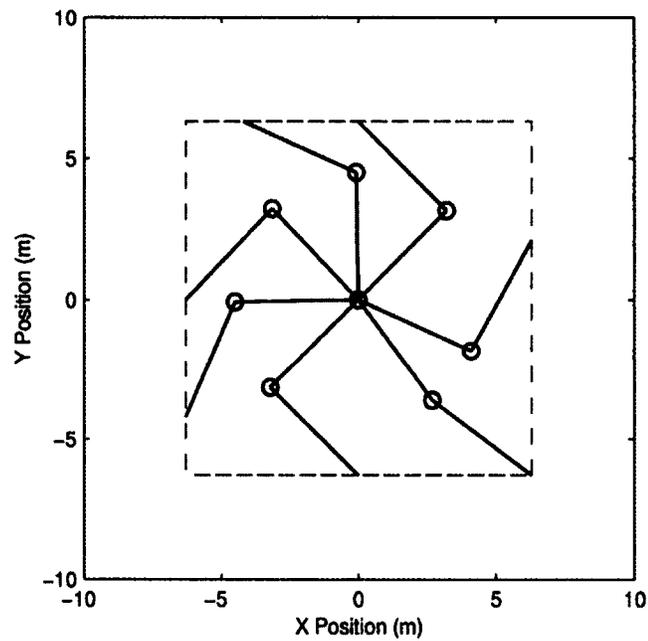


Figure 1.2: Desired end-effector square trajectory in the task space, illustrating the motion of the two-link manipulator in a counterclockwise direction starting from rest at the lower-right-hand corner.

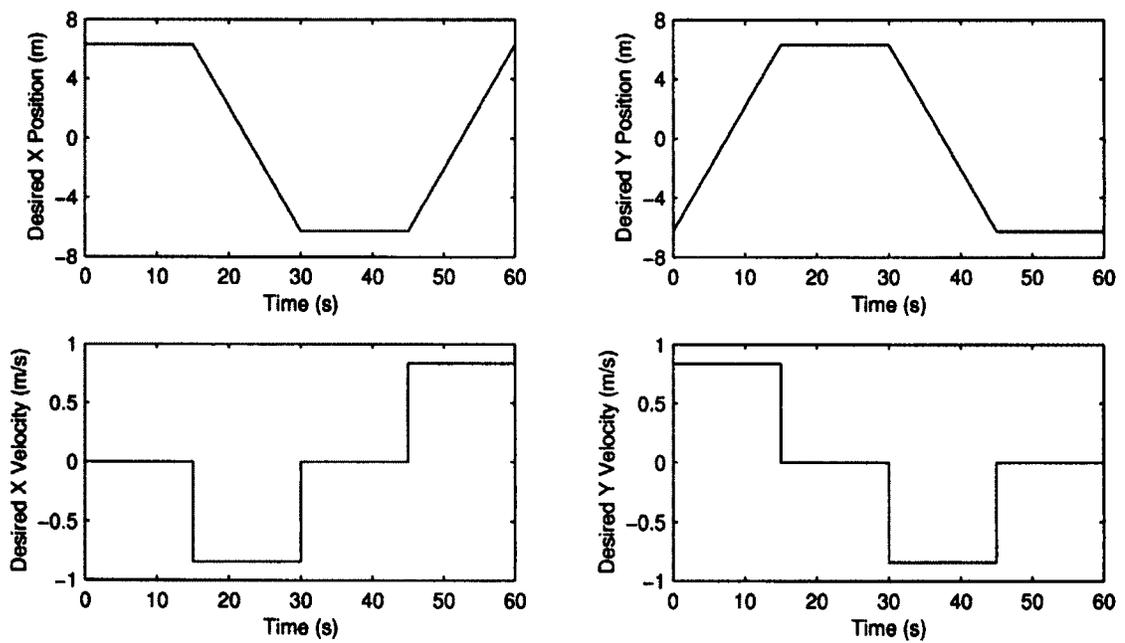


Figure 1.3: Desired end-effector position and velocity as a function of time illustrating abrupt changes in direction at $t = 15$ sec., 30 sec. and 45 sec.

with a minimum *positioning overshoot*, which is defined as the position deviation observed in the task space. Indeed, overshooting the trajectory in the task space could lead to collisions and severe operational consequences; while attempting to grasp a spinning satellite, for example. For this reason, all control methodologies developed in this work will be assessed by their ability to provide minimum end-effector positioning overshoots at the direction changes, and minimum settling time to a steady state along each side, while maintaining stability throughout the entire trajectory.

1.6 Organization

The remainder of this work is organized as follows:

To establish the feasibility of a direct adaptive flexible-joint control scheme based on fuzzy logic, Chapter 2: *Fuzzy Logic-Based Direct Adaptive Control*, presents the development of a fuzzy logic-based adaptive controller for the task-space trajectory tracking problem by the end-effector of a rigid-joint manipulator. The control strategy is based on a new direct adaptive control scheme, in which the controller gains are adapted in real-time according to a fuzzy logic system, such that the tracking errors between a reference model and the actual robot system outputs are brought to zero. Numerical simulations compare the trajectory tracking performance of the fuzzy-logic-based adaptive controller with a nonadaptive transponse Jacobian controller.

Based on the recent finding of sufficient conditions that allow nonlinear and non-stationary systems to become stable and strictly passive via static or dynamic output feedback, Chapter 3: *A New Class of SAC-Based Adaptive Controllers*, presents the theoretical development of a new class of direct adaptive controllers that are applicable to nonlinear systems. The proposed control methodology uses a innovative decentralized adaptation mechanism, developed upon the SAC technique. The resulting direct adaptive control technique is referred to as *decentralized simple adaptive control* (DSAC). A simplification of this new control law, referred to as *decentralized modified simple adaptive control* (DMSAC), is also presented. Furthermore, how these new control techniques can be modified to be applicable to practical situations, where the trajectory-tracking errors cannot reach zero for some reasons, will be illustrated. The formal Lyapunov proof of stability is established in all cases. To illustrate the

new adaptive control approaches, numerical simulation studies for a robotic trajectory tracking problem are presented.

Building on the new adaptive control methodologies presented in Chapters 2 and 3, Chapter 4: *Direct Adaptive Composite Control Methodologies*, addresses the problem of adaptive trajectory tracking control of space manipulators that exhibit elastic vibrations in their joints and are subject to parametric uncertainties and modeling errors. First, it presents a comprehensive study of rigid-joint and linear flexible-joint stiffness dynamics models, in order to propose a new dynamic formulation that includes nonlinear effects, such as soft-windup and time-varying joint stiffness. Secondly, it develops innovative fuzzy-logic and SAC-based adaptive composite control schemes for tracking the end-effector of a two-link flexible-joint manipulator. All new control schemes consist of a direct model reference adaptive system, designed to stabilize the rigid manipulator dynamics, and a linear correction term to improve damping of vibrations at the joints. A Tychonov-based stability analysis is used to analyze the closed-loop stability of the adaptive composite control approaches. Numerical simulations compare the performance of the adaptive composite controllers with a nonadaptive control technique, and with a conventional model-based nonlinear backstepping control strategy.

In Chapter 5: *Partial State Feedback Adaptive Composite Control*, an extended Kalman filter (EKF) strategy to estimate all state variables from noisy and biased measurements in a flexible-joint manipulator system modeled with the linear flexible-joint stiffness dynamics model is provided. An extension for the new nonlinear flexible-joint stiffness dynamics formulation is then presented. In both cases, a nonlinear observability analysis based on Lie derivatives is provided, to establish the theoretical applicability of the observation schemes. The state estimates are coupled to one of the direct adaptive composite controllers, to provide a partial state-feedback adaptive composite control strategy. Numerical simulations are used to demonstrate the approach.

Finally, Chapter 6: *Conclusion*, provides an overview of the significant contributions this work makes in various areas related to the field of direct adaptive control

theory with applications to rigid and flexible-joint space manipulator systems. Suggestions for future research directions are also proposed.

Chapter 2

Fuzzy Logic-Based Adaptive Control

To establish the feasibility of a direct adaptive flexible-joint control scheme derived from the fuzzy logic theory, this chapter presents the development of a new fuzzy-logic based adaptive controller for tracking a square trajectory by the end-effector of a rigid-joint manipulator. The control strategy is based on a new direct adaptive control scheme in which the controller gains are adapted in real-time according to a fuzzy logic system, such that the tracking errors between a reference model and the actual manipulator system outputs are brought to zero. Numerical simulations compare the trajectory tracking performance of the fuzzy logic-based adaptive controller with a nonadaptive transpose Jacobian controller.

2.1 Introduction

Over the years, different fuzzy logic methods have been proposed to control rigid-robot manipulators, and these have provided various results. One of the earliest fuzzy control approaches for rigid-joint robots, in which a fuzzy logic system tuned the gains of a PID controller was developed by de Silva [100]. The performance of this controller was evaluated while tracking a square trajectory. However, although satisfactory results were obtained, the square trajectory was specified to reduce the desired end-effector velocity to zero at each corner of the trajectory, which facilitated the tracking task. Yi and Chung [101] investigated the robustness of a fuzzy logic controller for a two-link robot manipulator with uncertainties. In their work, the performance was validated by numerical simulations while tracking a circle trajectory. Green and Sasiadek [102] developed a strategy using two fuzzy controllers to substitute for the dynamics equations of a two-link robot tracking a $12.6 \text{ m} \times 12.6 \text{ m}$ square trajectory. Compared to a nonadaptive controller, end-effector transient vibrations at direction switches were significantly reduced with the fuzzy control strategy. A control

scheme comprising a sectorial fuzzy control term and a full nonlinear robot dynamics control term was developed by Santibanez *et al.* [103]. Experimental comparison of the proposed fuzzy logic control strategy with both a conventional fixed-gain and a variable-gain control developed by Llama *et al.* [104] was presented. More recently, Ham and Johnson [105] proposed a fuzzy logic controller designed to guarantee good tracking performance, in which each fuzzy set included a nonlinear robust control term designed using Lyapunov's direct method. The resulting controller was compared with a standard nonlinear robust controller in simulations for a manipulator with friction and disturbances, and the results demonstrated slightly improved tracking performance. Islam and Liu [106] designed a controller by combining a fuzzy system with a robust adaptive control algorithm. In their work, the fuzzy system was used to approximate a certainty equivalent-based optimal controller, while the robust adaptive algorithm was used to cope with uncertainties due to external disturbances and modeling errors. Simulation results demonstrated the feasibility of this robust fuzzy approach.

The objective of this chapter is to propose an innovative direct adaptation mechanism for tuning the gains of a transpose Jacobian controller using a fuzzy logic adaptation system. The adaptive fuzzy control scheme seeks to bring the trajectory tracking errors between the reference model outputs and the plant outputs to zero. In contrast with earlier strategies, this new direct adaptive fuzzy control approach does not require the identification of plant parameters, does not rely on any mathematical models of the manipulator system dynamics, and is simple to implement, thus decreasing the computational resources required for implementation.

2.2 Rigid-Joint Dynamics

A typical closed form of the nonlinear dynamics of a multilink robot with rigid joints is derived in terms of kinetic and potential energies stored in the system by the Euler-Lagrange formulation [18, 107–110]. Given an independent set of generalized coordinates, where n is the number of DOF of the system, the total kinetic and potential energies stored in the rigid n -DOF system, denoted by $T_r, U_r \in \mathbb{R}$, respectively, are defined by the Lagrangian

$$L(q_i, \dot{q}_i) = T_r - U_r; \quad i = 1, \dots, n \quad (2.1)$$

For a robot subjected to a generalized force $\tau_i \in \mathbb{R}$ acting on generalized coordinates $q_i \in \mathbb{R}$, the nonlinear dynamics equations of motion are given by

$$\tau_i = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i}; \quad i = 1, \dots, n \quad (2.2)$$

In the case of a rigid manipulator, the only source of potential energy is gravity. However, for space robot applications, the potential energy of the links U_r is omitted. Assuming that the kinetic energy of the manipulator is a quadratic function of the link angular rates, T_r is obtained as follows

$$T_r = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n M_{ij} \dot{q}_i \dot{q}_j = \frac{1}{2} \dot{q}^T M(q) \dot{q} \quad (2.3)$$

where $M_{ij} \in \mathbb{R}$ represents the components of the symmetric positive-definite link inertia matrix denoted by $M(q) \in \mathbb{R}^{n \times n}$, and $q \in \mathbb{R}^n$ represents the link angle vector.

The Lagrangian function of the rigid robot is

$$L = \frac{1}{2} \dot{q}^T M(q) \dot{q} \quad (2.4)$$

Applying the partial derivatives (2.2) to the Lagrangian yields the following Euler-Lagrange rigid-joint dynamics equations of a n -DOF space-based rigid-joint manipulator

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} \quad (2.5)$$

where $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ represents the centrifugal-Coriolis matrix, and $\tau \in \mathbb{R}^n$ denotes the control input vector. For the two-link robot shown in Fig. 2.1, $M(q) \in \mathbb{R}^{2 \times 2}$ and $C(q, \dot{q}) \in \mathbb{R}^{2 \times 2}$ are given by [18, 108]

$$M(q) = \begin{bmatrix} m_1 l_{c_1}^2 + m_2(l_1^2 + l_{c_2}^2 + 2l_1 l_{c_2} \cos q_2) + I_1 + I_2 & m_2(l_{c_2}^2 + l_1 l_{c_2} \cos q_2) + I_2 \\ m_2(l_{c_2}^2 + l_1 l_{c_2} \cos q_2) + I_2 & m_2 l_{c_2}^2 + I_2 \end{bmatrix} \quad (2.6)$$

$$C(q, \dot{q}) = -m_2 l_1 l_{c_2} \sin q_2 \begin{bmatrix} \dot{q}_2 & \dot{q}_1 + \dot{q}_2 \\ -\dot{q}_1 & 0 \end{bmatrix} \quad (2.7)$$

where, $\forall i = 1, 2$, $l_i \in \mathbb{R}$ denotes the length of link i , $l_{c_i} \in \mathbb{R}$ denotes the distance from the previous joint to the center of gravity of link i , and $m_i \in \mathbb{R}$ denotes the mass of link i . The moment of inertia of link i about an axis perpendicular to the xy plane passing through the center of gravity of link i , denoted by $I_i \in \mathbb{R}$, is given by

$$I_i = \frac{m_i l_i^2}{12} \quad (2.8)$$

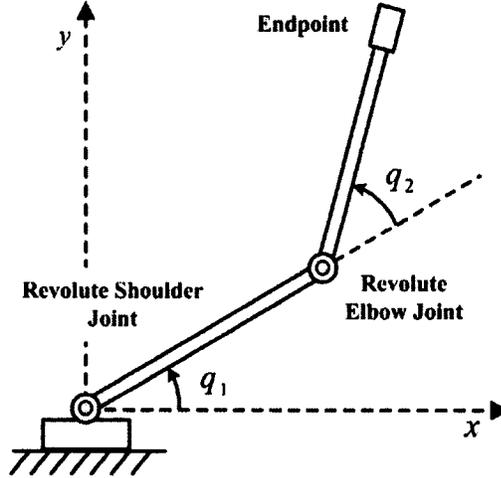


Figure 2.1: Schematic of the two-link robot manipulator

The detailed derivation of the dynamics equations is presented in Appendix A.

The physical characteristics of the robot are based on previous studies on intelligent control of space robots by Banerjee and Singhose [12], and Green and Sasiadek [15, 111], and are given by $m_1 = m_2 = 1.5075$ kg, $l_1 = l_2 = 4.5$ m, and $l_{c_1} = l_{c_2} = 2.25$ m. With these values, $M(q)$ and $C(q, \dot{q})$ are given by

$$M(q) = \begin{bmatrix} 50.8781 + 30.5269 \cos q_2 & 10.1756 + 15.2634 \cos q_2 \\ 10.1756 + 15.2634 \cos q_2 & 10.1756 \end{bmatrix} \quad (2.9)$$

$$C(q, \dot{q}) = -15.2634 \sin q_2 \begin{bmatrix} \dot{q}_2 & \dot{q}_1 + \dot{q}_2 \\ -\dot{q}_1 & 0 \end{bmatrix} \quad (2.10)$$

2.3 Nonadaptive Transpose Jacobian Control Development

As a benchmark control strategy, the following intuitive task-based nonadaptive transpose Jacobian control law developed by Craig [89] is considered

$$\tau = J^T(q) \left\{ K_p \begin{bmatrix} y_{e1} \\ y_{e2} \end{bmatrix} + K_d \begin{bmatrix} y_{e3} \\ y_{e4} \end{bmatrix} \right\} \quad (2.11)$$

where $J(q) \in \mathbb{R}^{2 \times 2}$ denotes the robot Jacobian matrix, $K_p, K_d \in \mathbb{R}^{2 \times 2}$ represent the proportional and derivative control gains respectively, and $y_{e_i} \in \mathbb{R} \forall i = 1, \dots, 4$ denotes the component of the trajectory tracking error vector $y_e \in \mathbb{R}^4$ that is defined between the desired end-effector Cartesian position and velocity vectors, denoted by $x_{r_d} \in \mathbb{R}^2$ and $\dot{x}_{r_d} \in \mathbb{R}^2$, and the actual end-effector position and velocity vectors, denoted by $x_r \in \mathbb{R}^2$ and $\dot{x}_r \in \mathbb{R}^2$, as follows

$$y_e \triangleq y_d - y = \begin{bmatrix} x_{r_d} \\ \dot{x}_{r_d} \end{bmatrix} - \begin{bmatrix} x_r \\ \dot{x}_r \end{bmatrix} \quad (2.12)$$

The control torque actuating each joint obtained with the control law (2.11) feeds into the inverse flexible dynamics equations to give an angular acceleration vector, which is double integrated to obtain link positions and velocities. Link positions transform into x_r using the direct kinematics relations given by

$$x_r = \begin{bmatrix} l_1 \cos q_1 + l_2 \cos (q_1 + q_2) \\ l_1 \sin q_1 + l_2 \sin (q_1 + q_2) \end{bmatrix} \quad (2.13)$$

and the end-effector velocity \dot{x}_r is obtained by time-differentiating (2.13), as follows

$$\dot{x}_r = \begin{bmatrix} -l_1 \sin(q_1) \dot{q}_1 - l_2 \sin(q_1 + q_2) (\dot{q}_1 + \dot{q}_2) \\ l_1 \cos(q_1) \dot{q}_1 + l_2 \cos(q_1 + q_2) (\dot{q}_1 + \dot{q}_2) \end{bmatrix} \quad (2.14)$$

2.4 Fuzzy Adaptive Control Development

The proposed fuzzy adaptive control scheme automatically calculate the gains of the transpose Jacobian control law in (2.11) as a function of the tracking errors between the actual plant output vector $y \in \mathbb{R}^4$ and the output vector of a reference model $y_m \in \mathbb{R}^4$. Thus, the control objective consists in minimizing the output tracking error vector denoted by $e_y \in \mathbb{R}^4$ and defined as follows

$$e_y \triangleq y_m - y = \begin{bmatrix} x_{r_m} \\ \dot{x}_{r_m} \end{bmatrix} - \begin{bmatrix} x_r \\ \dot{x}_r \end{bmatrix} \quad (2.15)$$

where the reference model defines the ideal response to the desired square trajectory. In this work, the reference model is designed to incorporate the desired input-output plant behavior, which is expressed in terms of the ideal damping ratio $\zeta \in \mathbb{R}$ and the undamped natural frequency $\omega_n \in \mathbb{R}$, as follows

$$A_m = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\omega_n^2 & 0 & -2\zeta\omega_n & 0 \\ 0 & -\omega_n^2 & 0 & -2\zeta\omega_n \end{bmatrix} \quad B_m = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\omega_n^2 & 0 & 0 & 0 \\ 0 & -\omega_n^2 & 0 & 0 \end{bmatrix} \quad (2.16)$$

$$C_m = I_4 \quad (2.17)$$

and where the reference model state vector x_m and input vector u_m are given by

$$x_m = \begin{bmatrix} x_{r_m} \\ \dot{x}_{r_m} \end{bmatrix} \in \mathbb{R}^4, \quad u_m = \begin{bmatrix} x_{r_d} \\ \dot{x}_{r_d} \end{bmatrix} \in \mathbb{R}^4$$

Adopting a decentralized adaptive fuzzy control strategy, the proportional and

derivative control gains of the transpose Jacobian controller, K_p and K_d , are respectively replaced by diagonal time-varying control gains, as follows

$$K_p(t) = \text{diag} \left[h_1 \lambda_1(t), h_2 \lambda_2(t) \right] \quad (2.18)$$

$$K_d(t) = \text{diag} \left[h_3 \lambda_3(t), h_4 \lambda_4(t) \right] \quad (2.19)$$

As a result, the direct adaptive fuzzy control law is given by

$$\tau = J^T(q) \left\{ \begin{bmatrix} h_1 \lambda_1(t) & 0 \\ 0 & h_2 \lambda_2(t) \end{bmatrix} \begin{bmatrix} e_{y_1} \\ e_{y_2} \end{bmatrix} + \begin{bmatrix} h_3 \lambda_3(t) & 0 \\ 0 & h_4 \lambda_4(t) \end{bmatrix} \begin{bmatrix} e_{y_3} \\ e_{y_4} \end{bmatrix} \right\} \quad (2.20)$$

Four normalized fuzzy logic systems (FLS) are used, each dedicated to one diagonal element of the control gains. Each FLS takes one component of the tracking error vector e_y through scaling gains as an input, and outputs its corresponding parameter $\lambda_i \in \mathbb{R} \forall i = 1, \dots, 4$. Figures 2.2 and 2.3 show the FLS membership functions for normalized input and output variables respectively. Universes of discourse range from -2 to 2 m for e_{y_1} and e_{y_2} , from -10 to 10 m/s for e_{y_3} and e_{y_4} , and from 0 to 1 for λ_i . The input scaling gains are selected as $g_1 = g_2 = 0.5$ and $g_3 = g_4 = 0.1$ causing the left-most membership function to saturate (i.e. to peak) at -1 and the right-most at +1 for both the input and output universe of discourse, hence resulting in normalized FLS. Output scaling gains $h_i \in \mathbb{R} \forall i = 1, \dots, 4$ multiply their corresponding output variable λ_i , in order to modify their base widths and provide greater tracking accuracy toward asymptotically zero position errors as the scaling gain increases. Increasing the output scaling gains actually has the effect of spreading the output membership functions, thereby making the meaning of their associated linguistics quantify larger numbers, as noted by Passino and Yurkovich [112].

Each normalized FLS is a Mamdani type, and their single input variable is designed with nine Gaussian membership functions, while all output variables are designed with five Gaussian membership functions. Similar to Green and Sasiadek [15], verbal descriptors for *positive maximum* (PM), *positive very high* (PVH), *positive*

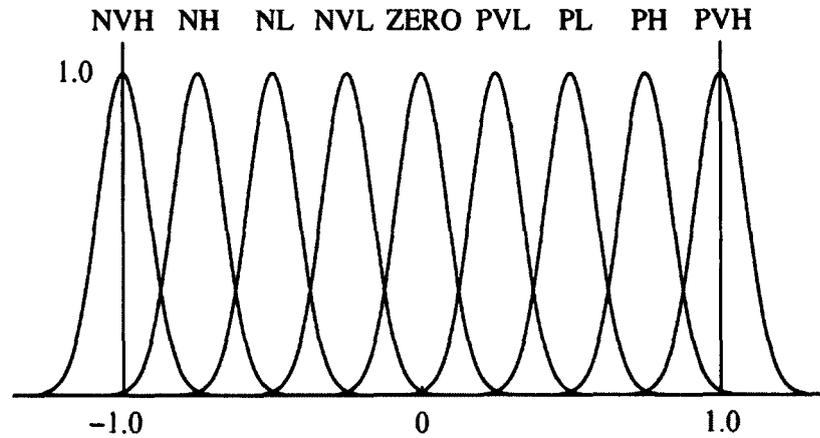


Figure 2.2: Membership functions for normalized input variables

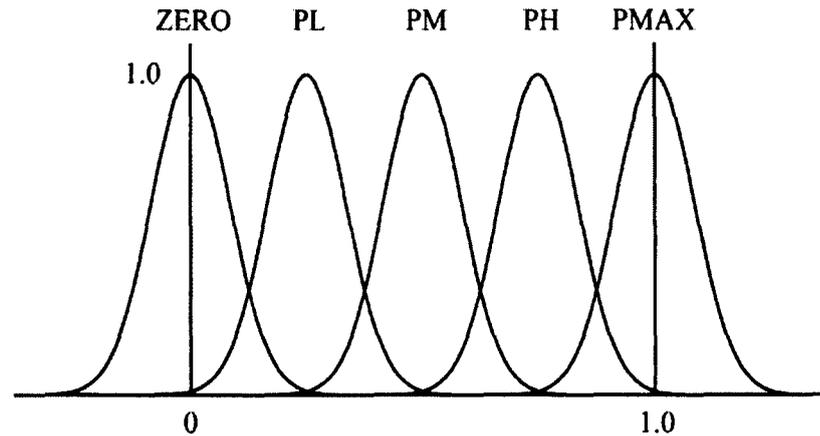


Figure 2.3: Membership functions for normalized output variables

high (PH), *positive medium* (PM), *positive low* (PL), *positive very low* (PVL), *zero* (ZERO), *negative very low* (NVL), *negative low* (NL), *negative high* (NH), and *negative very high* (NVH) are used in the generation of the fuzzy rules of the form

IF e_{y_1} **is** PVL **THEN** λ_1 **is** PVL

IF e_{y_2} **is** NH **THEN** λ_2 **is** PH

All four FLS are designed intuitively, so that as the magnitude of a given tracking error varies positively or negatively, the control parameter λ_i related to that tracking

e_{y_i}	NVH	NH	NL	NVL	ZERO	PVL	PL	PH	PVH
λ_i	NVH	NH	NL	NVL	ZERO	PVL	PL	PH	PVH

Table 2.1: Fuzzy logic system rule table

error varies according to the magnitude of the tracking error. The value of λ_i is maintained ≥ 0 , ranging from ZERO for zero tracking error to PVH for the largest tracking error, either positive or negative, resulting in the symmetric fuzzy rule given in Table 2.1.

Within each FLS, the implication, aggregation and defuzzification methods are MIN, GEOMETRIC, and CENTROID, respectively. Fuzzification simply determines the degree of membership $0 \leq \mu \leq 1$ associated with their input crisp datum value. The MIN implication operator determines the minimum value of μ causing the truncation of the output membership function. The CENTROID method defuzzifies an implicated fuzzy membership curve (i.e. converts it to a crisp output value) by calculating its center of area (COA), which results in a μ value corresponding to a crisp value for λ_i , as follows

$$\text{COA} = \frac{\int \mu(\lambda_i) \lambda_i d\lambda_i}{\int \mu(\lambda_i) d\lambda_i} \quad (2.21)$$

Each crisp datum variable appears in all rules. Hence, each rule assigns a defuzzified value to the crisp datum λ_i . The GEOMETRIC aggregation method [112] determines the final crisp value from the results of all the rules.

As explained by Green [113], a general framework for the stability analysis of fuzzy logic control systems has not yet been developed. So far, several methods have been proposed in attempts to analyze the stability of fuzzy logic control methodologies, including include hyperstability, Lyapunov, bifurcation, interval matrix, linguistic, graphical, parameter plane and rule firing trajectory observation [112, 114, 115]. Among these, the Lyapunov method is the most highly developed with respect to general wide acceptance. However, this method is only applicable to Takagi-Sugeno type fuzzy inference systems, in which the consequents are represented by singletons or first-order polynomials as opposed to membership functions. Thus, this stability analysis is not well suited for the proposed application. In addition, most of the other

methods involve several restrictions to general use, and are most often only suited for linearized plants. In view of the above discussion, the stability of the proposed fuzzy adaptive control scheme is assessed in closed-loop numerical simulations.

2.5 Simulation Results

Modeling and simulation of both control strategies were evaluated in numerical simulations. For the sake of simplicity, to demonstrate the applicability and performance of the proposed fuzzy logic-based control approach, the MATRIXx *fuzzy logic block* was used to implement the adaptive fuzzy controller within a standardized block-diagram control-logic structure [116]. The nonadaptive control gains were tuned in numerical simulations, to provide optimal tracking results. For the specified square trajectory, the control gains were set to

$$K_p = 67.24I_2 \quad K_d = 10.59I_2$$

Typically, achieving good results with a fuzzy controller with suitable output scaling gains requires several iterations in numerical simulations. Within the context of this work, the best tracking results were obtained with the following scaling gains

$$h_1 = h_2 = 300 \quad h_3 = h_4 = 525$$

Figures 2.4 and 2.6 show the results of end-effector tracking trajectories. For the nonadaptive transpose Jacobian control scheme shown in Fig. 2.4, pronounced overshoots of 0.29 m occur at each direction switch due to large trajectory tracking errors, as illustrated in Fig. 2.5. However, the trajectory for the direct adaptive fuzzy control strategy shown in Fig. 2.6 exhibits minimal positioning overshoots of 0.10 m at each corner, and rapidly settles to a steady state with small tracking errors (see Fig. 2.7). These results show that using the proposed FLS adaptation mechanism greatly improves the tracking accuracy, particularly at each corner of the trajectory, and provides smaller tracking errors, resulting in tracking that is close to a straight

line along each side of the square trajectory.

As shown in Fig. 2.8, proportional control gain components $K_{p_{11}}(t)$ and $K_{p_{22}}(t)$ are relatively low, varying from 3.9 to 4.8 along the each side of the trajectory, and peaking at 7.8 at each direction change. As shown in this figure, $K_{p_{11}}(t)$ reaches a maximum only at the second direction change ($t=30$ sec.), since this control gain is associated with the input variable e_{y_1} which must be stabilized particularly at the second direction change, when the overshoot occurs about the x -direction. At the first and third direction changes ($t=15$ sec. and $t=45$ sec.) the output control gain $K_{p_{11}}(t)$ nevertheless fluctuates rapidly, thereby providing additional stabilization and damping. A similar behavior can be observed for output control gain $K_{p_{22}}(t)$ associated with input variable e_{y_2} , which peaks at the first and third direction change and provides additional damping at the second direction switch.

Similarly, as illustrated in Fig. 2.8, derivative control gain components $K_{d_{11}}(t)$ and $K_{d_{22}}(t)$ remain stable at a value of 8.4 along each side of the square trajectory and increase sharply to 26.8 at each direction change, which provides good velocity tracking control at each direction change.

2.6 Summary

As a first step toward the development of a fuzzy logic-based direct adaptive control methodology for flexible-joint manipulators, this chapter presented an effective fuzzy adaptive rigid-joint control strategy. The direct adaptation scheme consists of fuzzy adaptation of the nonadaptive transpose Jacobian control gain parameters as a function of the tracking errors between the reference model outputs and the actual plant outputs. The numerical simulation results presented in this chapter not only establish the feasibility and applicability of the approach, but also demonstrate greatly improved trajectory tracking results compared to the nonadaptive transpose Jacobian control law.

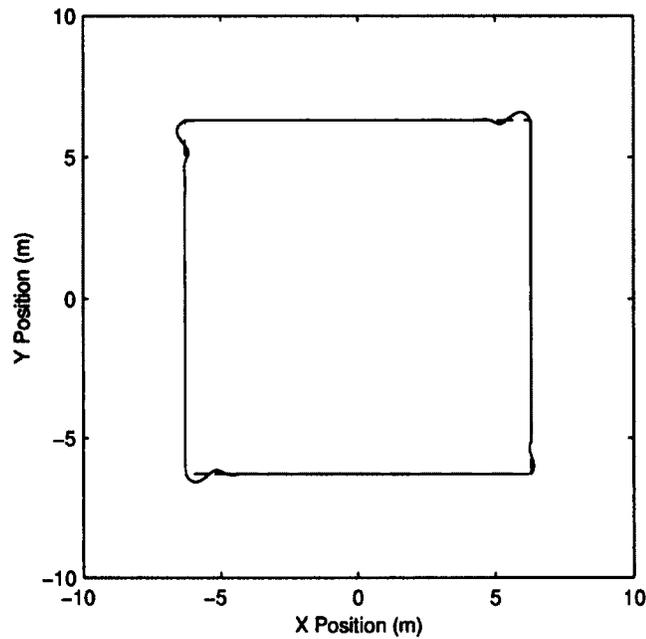


Figure 2.4: Trajectory tracking results obtained with the nonadaptive transpose Jacobian controller applied to the rigid-joint manipulator. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$.

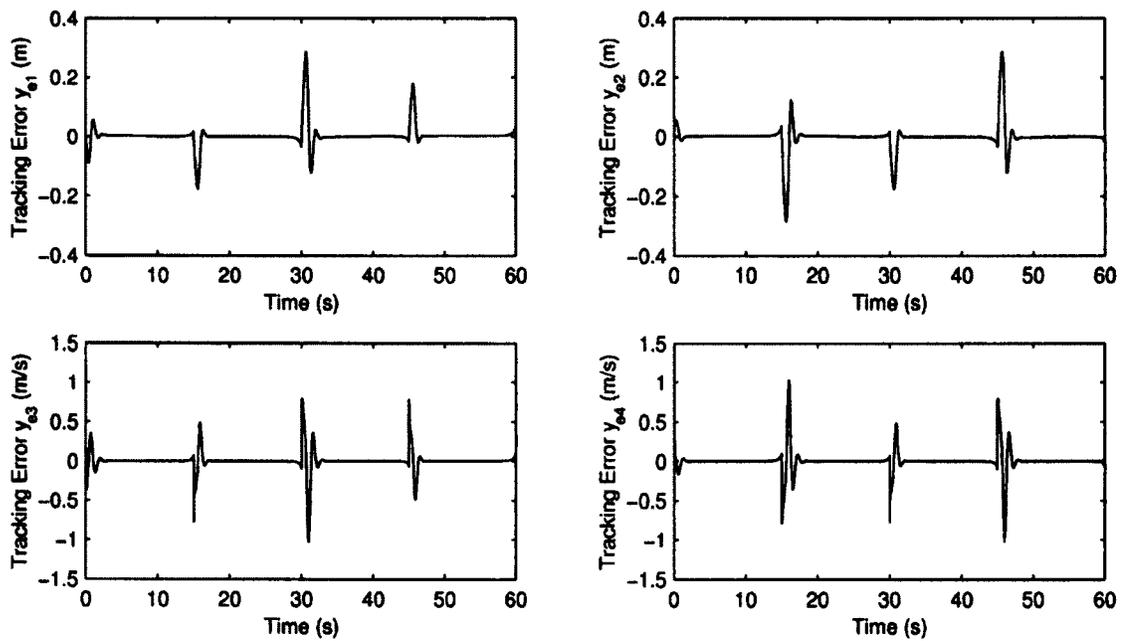


Figure 2.5: Trajectory tracking errors ($y_e(t) = y_d(t) - y(t)$) obtained with the nonadaptive transpose Jacobian controller applied to the rigid-joint manipulator.

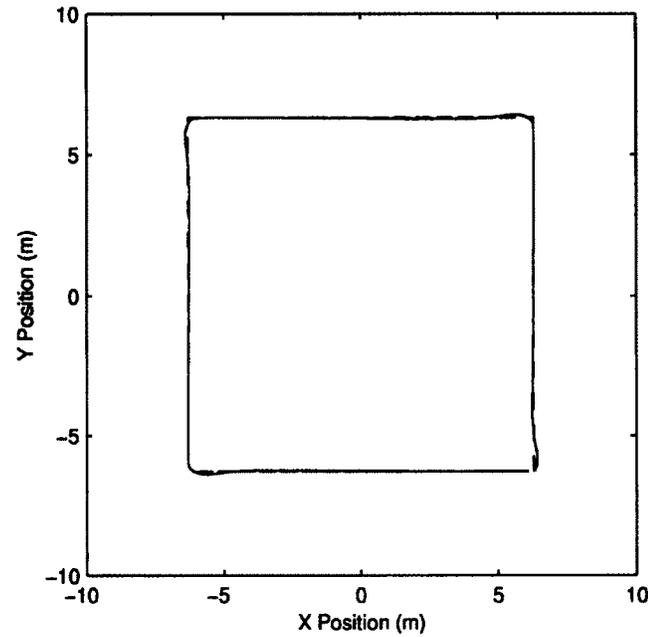


Figure 2.6: Trajectory tracking results obtained with the fuzzy adaptive controller applied to the rigid-joint manipulator. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$.

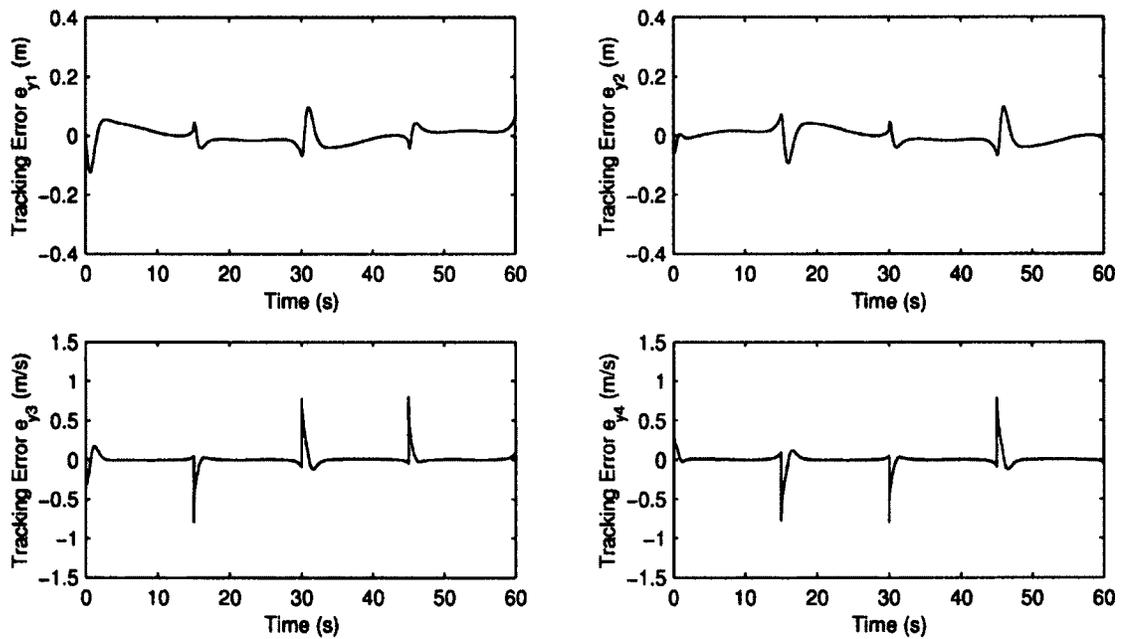


Figure 2.7: Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the fuzzy adaptive controller applied to the rigid-joint manipulator.

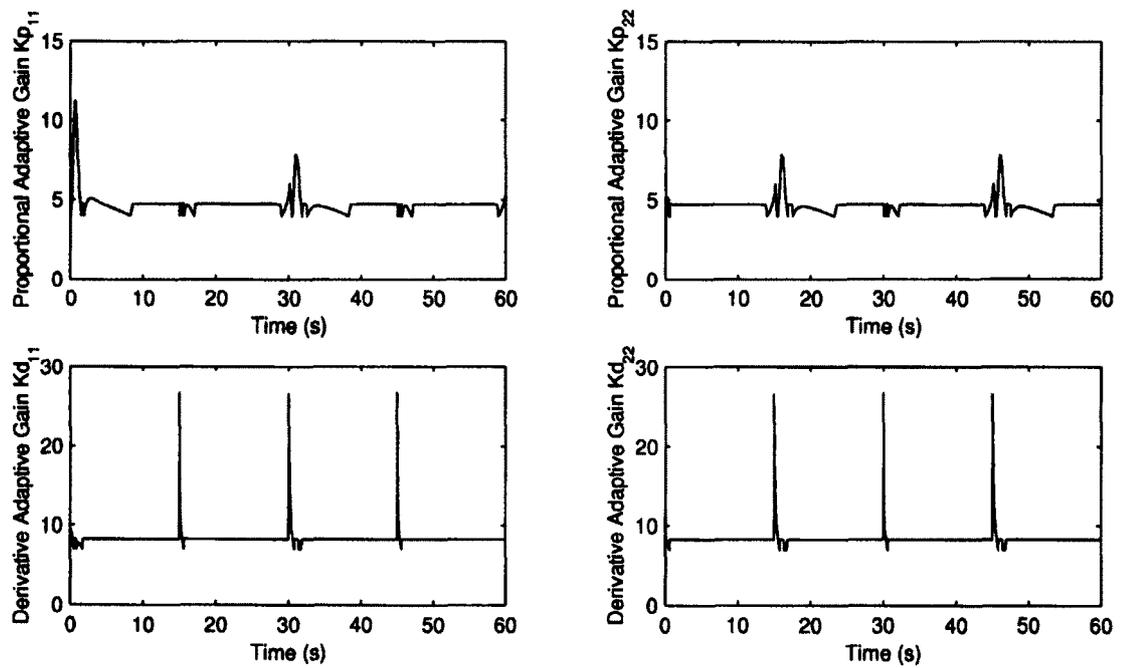


Figure 2.8: Adaptation history of the fuzzy controller gains $K_p(t)$ and $K_d(t)$ when applied to the rigid-joint manipulator.

Chapter 3

A New Class of SAC-Based Adaptive Controllers

Recently, the passivity results for linear time-invariant systems were successfully extended to nonlinear and nonstationary systems, thus guaranteeing stability of adaptive control of nonlinear square systems. Based on this theoretical breakthrough, this chapter presents the development of a new class of direct adaptive controllers, which employ a new decentralized adaptation law mechanism that is developed from the simple adaptive control technique. The resulting direct adaptive control methodology is referred to as *decentralized simple adaptive control* (DSAC). A simplification of this new control algorithm, referred to as *decentralized modified simple adaptive control* (DMSAC), is also presented. In addition, how to modify both control methodologies can be modified to avoid divergence in practical situations, where the trajectory tracking errors cannot reach zero, is presented. The formal Lyapunov proof of stability is established in all cases. As well, a numerical simulation study for a trajectory tracking problem by a rigid-joint manipulator is presented to illustrate the new adaptive control approaches.

3.1 Introduction

A well-known result in control theory that plays an important role in guaranteeing stability in adaptive control is the notion of passivity, which requires the plant to be strictly passive (SP). For linear time-invariant (LTI) systems, this stability condition is equivalent to requiring the input-output transfer function to be strictly positive real (SPR). However, as most real-world systems are not inherently SP, this condition can be mitigated for LTI systems for which any constant output feedback gain (unknown and not required for implementation) could render the (fictitious) closed-loop system SP. Such systems, that are only separated from strict passivity by a constant output feedback, have been referred to as almost strictly passive (ASP), and their transfer

function as almost strictly positive real (ASPR) (see Barkana and Kaufman [117]). Many studies have attempted to clearly define what classes of systems satisfy the ASP conditions. Although some early results were obtained for both SISO and MIMO systems by Fradkov [118,119], these basic conditions had been considered difficult to satisfy and remained unclear. It was finally shown by Barkana [120,121] and Weiss *et al.* [122] that the ASP conditions required to guarantee stability with adaptive control are equivalent to requiring an LTI system with state-space realization $\{A, B, C\}$ to be ASP, or in other words, to be strictly minimum phase and with the product CB positive definite symmetric (PDS). Furthermore, it was demonstrated by Huang *et al.* [123] that if a system cannot be made SP via constant output feedback, no dynamic output feedback can render it SP.

Over the years, a variety of direct model reference adaptive control (MRAC) techniques have been developed to address the problem of time-varying the gains of a controller so that the plant closed-loop characteristics match those defined by a reference model [124]. In particular, this approach found many successful applications to robot control. This general interest in MRAC schemes is mainly due to the fact that adaptive control theory is well suited to robotic manipulators whose dynamics is highly nonlinear. Classical adaptation mechanisms for MRAC systems include the gradient method (also known as the MIT rule), the Lyapunov second direct method and the Popov hyperstability theory [125].

Dubowsky and DesForges [126] introduced the first MRAC technique for robotic manipulators, by using the gradient method as the adaptation law. Then, Takegaki and Arimoto [127], and Craig *et al.* [128] presented an MRAC strategy for manipulators using a Lyapunov method. An alternative to Lyapunov methodologies, a Popov hyperstability-based MRAC approach for a nonlinear industrial robot system, was developed by Nicosia and Tomei [129]. Horowitz and Tomizuka [130] used the hyperstability theory to derive an MRAC scheme for linearizing and decoupling the nonlinear manipulator dynamics. Although not an MRAC technique, the algorithm developed by Slotine and Li [131], referred to as the SLI control law, is one of the most popular adaptive control methods for manipulator systems. However, the SLI technique is not a *direct adaptive control* scheme, in that the controller gains are kept

constant. Instead, this *indirect adaptive control* law depends on unknown manipulator parameters, which are estimated (adapted) in real-time.

The drawback associated with classical MRAC adaptation mechanisms is their considerable computational burdens. When applied to robotic control, this shortcoming is often due to the computation of the Euler-Lagrange dynamics formulation, which prevents the use of such MRAC schemes for space robots. As with any space applications, available on-board computing resources are highly limited. The computational load problem inherent to classical MRAC techniques is further aggravated in real-time applications. The SLI control scheme also suffers from computational complexity, largely due to the parameter estimation process. Moreover, most MRAC methods are based on the assumption that prior knowledge of the unknown plant is available, and/or requires the plant to be of the same order as the reference model.

To mitigate these limitations, the SAC approach was developed by Sobel *et al.* [132], Barkana *et al.* [133] and Barkana and Kaufman [134]. Using the ASP results, the stability of the SAC technique for square LTI systems was rigorously established by Kaufman *et al.* [91]. This direct adaptive output feedback method is based on the command generator tracker methodology developed by Broussard and O'Brien [135], and requires the plant to track the reference model that is the ideal representation of the plant only as far as its outputs represent the desired output behavior of the plant. For this reason, this MRAC-based approach has been successfully applied to the control of a number of large-scale systems, without requiring large-order adaptive controllers. Sometimes, for practicality, the SAC algorithm is robustified by adopting Ioannou and Kokotovic's idea [136,137], in which a sigma-term, or forgetting factor, is introduced in the algorithm to avoid divergence of the time-varying control gains. The stability of the adaptive algorithm with the sigma-term for LTI systems was also demonstrated in [91].

However, although the complexity of SAC technique is greatly reduced when compared with standard model following techniques, it still presents a design complexity issue due to the large number of parameters and coefficients to be selected. In fact, the calculation of the control input involves a stabilizing output feedback control gain and two feedforward control gains, each calculated as the summation of a proportional

and an integral control gain component. To mitigate this design complexity, the modified simple adaptive control (MSAC) idea proposed by Ulrich and de Lafontaine [138] exploited the concept that only the stabilizing output feedback gain is absolutely necessary to guarantee the stability of the closed-loop system. In other words, with MSAC, the feedforward control gains are ignored.

In spite of successful implementations of SAC/MSAC, the conditions required to guarantee the stability of these adaptive algorithms for nonlinear systems remained unclear for a long time. This is why, until recently, the SAC/MSAC adaptive algorithms were designed *ad hoc* and validated by simulations. Then, in 2010, the ASPR results for LTI systems were successfully extended to nonlinear and nonstationary systems by Barkana [139], which ensured the stability of nonstationary control applied to nonlinear square systems. In addition, this work demonstrated the stability and applicability of a reduced SAC method that used only the integral component of the time-varying control gains.

Based on this theoretical breakthrough, this chapter presents the theoretical development of two novel SAC-based control schemes for nonlinear systems: (1) a *decentralized simple adaptive control* (DSAC) methodology, and (2) a *decentralized modified simple adaptive control* (DMSAC) methodology. In these two control techniques, only the diagonal of the time-varying gain matrices are considered. This way, the computational requirements of the new approaches are decreased in comparison with SAC/MSAC, thus facilitating real-time implementation. Compared to centralized control approaches, the computational efficiency advantage of decentralized control techniques make them attractive for applications in complex dynamical systems, such as nonlinear multilink space robot manipulators. The stability of the DSAC/DMSAC for general nonlinear square ASP systems is also formally established in the sense of Lyapunov. In addition, this chapter provides illustrative examples to compare the tracking performance of the proposed DSAC and DMSAC approaches for square and nonsquare rigid-joint manipulator systems.

3.2 System and Definitions

Consider a class of $m \times m$ nonlinear square systems described by the following formulation

$$\dot{x}(t) = A(x, t)x(t) + B(x, t)u(t) \quad (3.1)$$

$$y(t) = Cx(t) \quad (3.2)$$

where

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^n, \quad u = \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix} \in \mathbb{R}^m, \quad y = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} \in \mathbb{R}^m$$

are the states, inputs and outputs, respectively. The following definitions and theorem applicable to the nonlinear square system (3.1)-(3.2) will be exploited in the subsequent development. Refer to the work of Barkana [139] for more details.

Definition 3.1: Uniformly Strictly Minimum-Phase

Any nonlinear systems $\{A(x, t), B(x, t), C\}$ with the square state-space realization (3.1)-(3.2) is uniformly strictly minimum-phase if its zero dynamics is uniformly stable, or in other words, if there exist two matrices $M(x, t)$ and $N(x, t)$ satisfying the following relations

$$CM = 0 \quad (3.3)$$

$$NB = 0 \quad (3.4)$$

$$NM = I_m \quad (3.5)$$

such that the resulting zero dynamics given by

$$\dot{z} = (\dot{N} + NA)Mz \quad (3.6)$$

is uniformly asymptotically stable.

Definition 3.2: Strict Passivity in Time-Domain

Any nonlinear systems $\{A(x, t), B(x, t), C\}$ with the square state-space realization (3.1)-(3.2) is strictly passive (SP) if there exists two positive definite symmetric (PDS) matrices $P(x, t)$ and $Q(x, t)$ such that the following two conditions are simultaneously satisfied

$$\dot{P} + PA + A^T P = -Q \quad (3.7)$$

$$PB = C^T \quad (3.8)$$

The Lyapunov differential equation (3.7) shows that an SP system is uniformly asymptotically stable, whereas the second relation (3.8) shows that

$$B^T P B = B^T C^T = (CB)^T = CB \quad (3.9)$$

which implies that the product CB is PDS.

As most real-world systems are not inherently SP, a class of almost strictly passive (ASP) systems can be defined by Definition 3.3.

Definition 3.3: Almost Strict Passivity in Time-Domain

Any nonlinear systems $\{A(x, t), B(x, t), C\}$ with the square state-space realization (3.1)-(3.2) is ASP if there exists two PDS matrices $P(x, t)$ and $Q(x, t)$ and a constant output feedback gain \tilde{K}_e , such that the closed-loop system

$$\dot{x}(t) = \left[A(x, t) - B(x, t)\tilde{K}_e C \right] x(t) \quad (3.10)$$

$$y(t) = Cx(t) \quad (3.11)$$

simultaneously satisfies the following relations

$$\dot{P} + P(A - B\tilde{K}_e C) + (A - B\tilde{K}_e C)^T P = -Q \quad (3.12)$$

$$PB = C^T \quad (3.13)$$

Theorem 3.1: Any uniformly strictly minimum-phase nonlinear system $\{A(x, t), B(x, t), C\}$ with the square state-space realization (3.1)-(3.2), and with the product $CB(x, t)$ being PDS, is ASP.

3.3 Control Objective

The control objective is to design decentralized SAC-based controllers which ensure that the nonlinear square system tracks the output vector $y_m(t)$ of the following (not necessarily square) reference model

$$\dot{x}_m(t) = A_m x_m(t) + B_m u(t) \quad (3.14)$$

$$y_m(t) = C_m x_m(t) \quad (3.15)$$

where

$$x_m = \begin{bmatrix} x_{m_1} \\ \vdots \\ x_{m_{n_m}} \end{bmatrix} \in \mathbb{R}^{n_m}, \quad u_m = \begin{bmatrix} u_{m_1} \\ \vdots \\ u_{m_{p_m}} \end{bmatrix} \in \mathbb{R}^{p_m}, \quad y_m = \begin{bmatrix} y_{m_1} \\ \vdots \\ y_{m_m} \end{bmatrix} \in \mathbb{R}^m$$

are the reference model states, inputs and outputs, respectively. To quantify this control objective, an output tracking error, denoted by $e_y(t) \in \mathbb{R}^m$, is defined as

$$e_y \triangleq y_m - y \quad (3.16)$$

When the system tracks the reference model perfectly (i.e. $y_m = y^* = Cx^*$), it moves along a bounded ideal state trajectory, denoted by $x^*(t) \in \mathbb{R}^n$. To facilitate the

subsequent analysis, a state error, denoted by $e_x(t) \in \mathbb{R}^n$, is defined as

$$e_x \triangleq x^* - x \quad (3.17)$$

Thus, (3.16) can be rewritten as

$$e_y = Cx^* - Cx = Ce_x \quad (3.18)$$

Assumption 3.1: Both the order and the number of inputs of the reference model, n_m and p_m , are multiples of m , and thus satisfy the following relationships

$$n_m = k_n m \quad (3.19)$$

$$p_m = k_p m \quad (3.20)$$

where $k_n, k_p \in \mathbb{R}$ are positive scalars.

3.4 Decentralized Simple Adaptive Control Development

The standard SAC algorithm is adopted [91]

$$u = K_e(t)e_y + K_x(t)x_m + K_u(t)u_m \quad (3.21)$$

where $K_e(t) \in \mathbb{R}^{m \times m}$ is the time-varying stabilizing control gain matrix, and $K_x(t) \in \mathbb{R}^{m \times n_m}$ and $K_u(t) \in \mathbb{R}^{m \times m}$ are time-varying feedforward control gain matrices that contribute to maintaining the stability of the controlled system, and to bringing the output tracking error to zero. Each control gain matrix is calculated as the summation of a proportional and an integral component, as follows:

$$K_e(t) = K_{P_e}(t) + K_{I_e}(t) \quad (3.22)$$

$$K_x(t) = K_{P_x}(t) + K_{I_x}(t) \quad (3.23)$$

$$K_u(t) = K_{P_u}(t) + K_{I_u}(t) \quad (3.24)$$

where only the integral adaptive control terms are absolutely necessary to guarantee the stability of the direct adaptive control system. However, also including the proportional adaptive control terms increases the rate of convergence of the adaptive system toward perfect tracking. This is demonstrated in Section 3.4.2.

Proposing a DSAC adaptation mechanism, the proportional and the integral components of the stabilizing control gain in (3.22), $K_{P_e}(t), K_{I_e}(t) \in \mathbb{R}^{m \times m}$, are both updated by the output tracking error, which results in the following adaptive law

$$K_{P_e}(t) = \text{diag} \{e_y e_y^T\} \Gamma_{P_e} \quad (3.25)$$

$$\dot{K}_{I_e}(t) = \text{diag} \{e_y e_y^T\} \Gamma_{I_e} \quad (3.26)$$

where $\text{diag} \{A\}$ denotes the diagonalization operation on the square matrix $A \in \mathbb{R}^{n \times n}$ whose elements are denoted $a_{i,j}$, as follows

$$\text{diag} \{A\} = \begin{bmatrix} a_{1,1} & 0 & \cdots & 0 \\ 0 & a_{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{n,n} \end{bmatrix} \quad (3.27)$$

The components of the feedforward control gain matrices $K_{P_x}(t), K_{I_x}(t) \in \mathbb{R}^{m \times nm}$ and $K_{P_u}(t), K_{I_u}(t) \in \mathbb{R}^{m \times pm}$ are updated as follows

$$K_{P_x}(t) = R^T \text{diag} \{R e_y x_m^T\} \Gamma_{P_x} \quad (3.28)$$

$$\dot{K}_{I_x}(t) = R^T \text{diag} \{R e_y x_m^T\} \Gamma_{I_x} \quad (3.29)$$

$$K_{P_u}(t) = T^T \text{diag} \{T e_y u_m^T\} \Gamma_{P_u} \quad (3.30)$$

$$\dot{K}_{I_u}(t) = T^T \text{diag} \{ T e_y u_m^T \} \Gamma_{I_u} \quad (3.31)$$

with

$$R = \begin{bmatrix} I_m \\ I_m \\ \vdots \\ I_m \end{bmatrix} \in \mathbb{R}^{n_m \times m}, \quad T = \begin{bmatrix} I_m \\ I_m \\ \vdots \\ I_m \end{bmatrix} \in \mathbb{R}^{p_m \times m} \quad (3.32)$$

and where $\Gamma_{P_e}, \Gamma_{I_e} \in \mathbb{R}^{m \times m}$, $\Gamma_{P_x}, \Gamma_{I_x} \in \mathbb{R}^{n_m \times n_m}$, and $\Gamma_{P_u}, \Gamma_{I_u} \in \mathbb{R}^{p_m \times p_m}$ are constant diagonal matrices that control the rate of adaptation.

The adaptive algorithm can be rewritten in the following concise form

$$u = K(t)r(t) \quad (3.33)$$

where $K(t) \in \mathbb{R}^{m \times (m+n_m+p_m)}$ and $r(t) \in \mathbb{R}^{m+n_m+p_m}$ are respectively defined as:

$$K(t) \triangleq \begin{bmatrix} K_e(t) & K_x(t) & K_u(t) \end{bmatrix} = K_P(t) + K_I(t) \quad (3.34)$$

$$r \triangleq \begin{bmatrix} e_y^T & x_m^T & u_m^T \end{bmatrix}^T \quad (3.35)$$

With this representation, the total proportional and integral adaptive control gains, denoted by $K_P(t), K_I(t) \in \mathbb{R}^{m \times (m+n_m+p_m)}$, are updated as follows:

$$K_P = S^T \text{diag} \{ S e_y r^T \} \Gamma_P \quad (3.36)$$

$$\dot{K}_I = S^T \text{diag} \{ S e_y r^T \} \Gamma_I \quad (3.37)$$

where $\Gamma_P, \Gamma_I \in \mathbb{R}^{(m+n_m+p_m) \times (m+n_m+p_m)}$, and the scaling matrix S is given by

$$S = \begin{bmatrix} I_m \\ I_m \\ \vdots \\ I_m \end{bmatrix} \in \mathbb{R}^{(m+n_m+p_m) \times m} \quad (3.38)$$

3.4.1 Error Dynamics

The time derivative of (3.17) is

$$\dot{e}_x = \dot{x}^* - \dot{x} = A^*x^* + B^*u^* - Ax - Bu \quad (3.39)$$

Adding and subtracting Ax^* to (3.39), and rearranging gives

$$\dot{e}_x = Ae_x + (A^* - A)x^* + B^*u^* - Bu \quad (3.40)$$

Adding and subtracting Bu^* to (3.40), results in

$$\dot{e}_x = Ae_x + (A^* - A)x^* + B(u^* - u) + (B^* - B)u^* \quad (3.41)$$

Adding and subtracting $B\tilde{K}_e e_y$ to (3.41), and substituting e_y from (3.18) in the first term of the right-hand side of (3.41), yields

$$\dot{e}_x = (A - B\tilde{K}_e C) e_x + (A^* - A)x^* + B(u^* - u) + (B^* - B)u^* + B\tilde{K}_e e_y \quad (3.42)$$

Noting that the tracking error e_y along the ideal trajectory is zero, the ideal control input vector u^* is given by

$$u^* = \tilde{K}_x x_m + \tilde{K}_u u_m \quad (3.43)$$

Thus, substituting (3.33) and (3.43) into (3.42) gives

$$\dot{e}_x = \left(A - B\tilde{K}_e C \right) e_x + (A^* - A) x^* + B\tilde{K}_x x_m + B\tilde{K}_u u_m - BKr + (B^* - B) u^* + B\tilde{K}_e e_y \quad (3.44)$$

Equation(3.44) can be rewritten as

$$\dot{e}_x = \left(A - B\tilde{K}_e C \right) e_x + (A^* - A) x^* + (B^* - B) u^* - B \left(K - \tilde{K} \right) r \quad (3.45)$$

with $\tilde{K} \in \mathbb{R}^{m \times (m+n_m+p_m)}$ defined as

$$\tilde{K} \triangleq \begin{bmatrix} \tilde{K}_e & \tilde{K}_x & \tilde{K}_u \end{bmatrix} \quad (3.46)$$

Finally, substituting K from (3.35) yields

$$\dot{e}_x = \left(A - B\tilde{K}_e C \right) e_x + (A^* - A) x^* + (B^* - B) u^* - BK_P r - B \left(K_I - \tilde{K} \right) r \quad (3.47)$$

3.4.2 Stability Analysis

Theorem 3.2: The adaptive control law given by (3.21) with DSAC adaptation mechanism (3.22)-(3.31) ensures that all system signals are bounded under closed-loop operation, and results in asymptotic convergence of the state and output tracking errors, in the sense that

$$\|e_y\| \rightarrow 0 \quad \text{and} \quad \|e_x\| \rightarrow 0 \quad \text{as} \quad t \rightarrow \infty$$

where $\|\cdot\|$ denotes the standard Euclidean norm of a vector.

Proof: Let $V \in \mathbb{R}$ be a continuously differentiable positive-definite symmetric function given by

$$V = e_x^T P e_x + \text{tr} \left[\left(K_I(t) - \tilde{K} \right) \Gamma_I^{-1} \left(K_I(t) - \tilde{K} \right)^T \right] \quad (3.48)$$

The time-derivative of (3.48) is obtained as

$$\begin{aligned}\dot{V} = & \dot{e}_x^T P e_x + e_x^T \dot{P} e_x + e_x^T P \dot{e}_x + \text{tr} \left[\dot{K}_I(t) \Gamma_I^{-1} \left(K_I(t) - \tilde{K} \right)^T \right] \\ & + \text{tr} \left[\left(K_I(t) - \tilde{K} \right) \Gamma_I^{-1} \dot{K}_I^T(t) \right]\end{aligned}\quad (3.49)$$

Substituting e_y from (3.18), $K_P(t)$ from (3.36), $\dot{K}_I(t)$ from (3.37) and \dot{e}_x from (3.47) into (3.49) gives

$$\begin{aligned}\dot{V} = & e_x^T \left[\dot{P} + P \left(A - B \tilde{K}_e C \right) + \left(A - B \tilde{K}_e C \right)^T P \right] e_x \\ & - 2e_x^T P B S^T \text{diag} \{ S C e_x r^T \} \Gamma_P r \\ & - r^T \left[K_I(t) - \tilde{K} \right]^T B^T P e_x - e_x^T P B \left[K_I(t) - \tilde{K} \right] r \\ & + [(A^* - A) x^* + (B^* - B) u^*]^T P e_x + e_x^T P [(A^* - A) x^* + (B^* - B) u^*] \\ & + \text{tr} \left[S^T \text{diag} \{ S C e_x r^T \} \Gamma_I \Gamma_I^{-1} \left(K_I(t) - \tilde{K} \right)^T \right] \\ & + \text{tr} \left[\left(K_I(t) - \tilde{K} \right) \Gamma_I^{-1} \Gamma_I \text{diag} \{ S C e_x r^T \} S \right]\end{aligned}\quad (3.50)$$

Using the ASP conditions (3.12) and (3.13), the expression in (3.50) can be simplified as

$$\begin{aligned}\dot{V} = & -e_x^T Q e_x \\ & - 2e_x^T C^T S^T \text{diag} \{ S C e_x r^T \} \Gamma_P r \\ & - r^T \left[K_I(t) - \tilde{K} \right]^T C e_x - e_x^T C^T \left[K_I(t) - \tilde{K} \right] r \\ & + [(A^* - A) x^* + (B^* - B) u^*]^T P e_x + e_x^T P [(A^* - A) x^* + (B^* - B) u^*] \\ & + \text{tr} \left[S^T \text{diag} \{ S C e_x r^T \} \left(K_I(t) - \tilde{K} \right)^T \right] \\ & + \text{tr} \left[\left(K_I(t) - \tilde{K} \right) \text{diag} \{ S C e_x r^T \} S \right]\end{aligned}\quad (3.51)$$

Due to the diagonal forms of the results inside the trace functions, the following terms cancel one another

$$\text{tr} \left[S^T \text{diag} \{ S C e_x r^T \} (K_I(t) - \tilde{K})^T \right] - r^T [K_I(t) - \tilde{K}]^T C e_x = 0 \quad (3.52)$$

and similarly

$$\text{tr} \left[(K_I(t) - \tilde{K}) \text{diag} \{ S C e_x r^T \} S \right] - e_x^T C^T [K_I(t) - \tilde{K}] r = 0 \quad (3.53)$$

Thus, (3.51) can be simplified to

$$\begin{aligned} \dot{V} = & -e_x^T Q e_x - 2e_x^T C^T S^T \text{diag} \{ S C e_x r^T \} \Gamma_P r \\ & + [(A^* - A) x^* + (B^* - B) u^*]^T P e_x \\ & + e_x^T P [(A^* - A) x^* + (B^* - B) u^*] \end{aligned} \quad (3.54)$$

Assuming that the system parameters vary slowly compared to the control dynamics, such that

$$A^* = A \quad \text{and} \quad B^* = B \quad (3.55)$$

allows (3.54) to be simplified to

$$\dot{V} = -e_x^T Q e_x - 2e_x^T C^T S^T \text{diag} \{ S C e_x r^T \} \Gamma_P r \quad (3.56)$$

The Lyapunov derivative \dot{V} in (3.56) is uniformly negative definite with respect to e_x , but only negative semidefinite with respect to the entire state space $[e_x, K(t)]$. A direct result of the Lyapunov stability theory is that all dynamic values are bounded. Also, according to LaSalle's invariance principle [91, p. 43], all states and adaptive gains are bounded, and all system trajectories ultimately end in the domain defined by $\dot{V} \equiv 0$. Since \dot{V} is negative definite in e_x , it implies that the system ultimately ends with $e_x \equiv 0$, which proves that the adaptive control system demonstrates asymptotic convergence of the states and output errors, and boundedness of the adaptive gains.

Remark 3.1. Compared to the stability results obtained by Barkana [139], the additional negative term in (3.54) introduced by considering $K_P(t)$ in the DSAC algorithm contributes to the negativity of the Lyapunov derivative function and thus improves the rate of asymptotic convergence of the states and output tracking errors.

Remark 3.2. Based on Ioannou and Kokotovic's work [136, 137], to robustify the DSAC algorithm, forgetting terms can be introduced in the adaptation law to avoid divergence of the integral adaptive control gains in cases where the tracking error would not reach zero. With this adjustment, the time-varying integral control gain matrices are obtained as follows

$$\dot{K}_{I_e}(t) = \text{diag} \{e_y e_y^T\} \Gamma_{I_e} - \sigma_e K_{I_e}(t) \quad (3.57)$$

$$\dot{K}_{I_x}(t) = R^T (\text{diag} \{R e_y x_m^T\} \Gamma_{I_x} - \text{diag} \{\sigma_x R K_{I_x}(t)\}) \quad (3.58)$$

$$\dot{K}_{I_u}(t) = T^T (\text{diag} \{T e_y u_m^T\} \Gamma_{I_u} - \text{diag} \{\sigma_u T K_{I_u}(t)\}) \quad (3.59)$$

and similarly,

$$\dot{K}_I(t) = S^T (\text{diag} \{S e_y r^T\} \Gamma_I - \text{diag} \{\sigma_I S K_I(t)\}) \quad (3.60)$$

where $\sigma_e \in \mathbb{R}^{(m \times m)}$, $\sigma_x \in \mathbb{R}^{(n_m \times n_m)}$, $\sigma_u \in \mathbb{R}^{(p_m \times p_m)}$, and $\sigma_I \in \mathbb{R}^{(m+n_m+p_m) \times (m+n_m+p_m)}$ denote the forgetting coefficient matrices. With this modification to the DSAC algorithm, the Lyapunov derivative function becomes

$$\begin{aligned} \dot{V} = & -e_x^T Q e_x - 2e_x^T C^T S^T \text{diag} \{S C e_x r^T\} \Gamma_{Pr} \\ & - 2\text{tr} \left[S^T \text{diag} \{\sigma_I S K_I(t)\} \Gamma_I^{-1} \left(K_I(t) - \tilde{K} \right)^T \right] \end{aligned} \quad (3.61)$$

Thus, according to Lyapunov-Lasalle theorem, the application of the DSAC algorithm with the σ -terms results in bounded error tracking. Note that, although it affects the proof of stability, the use of the DSAC control law with this adjustment is preferable in most applications. Indeed, without the σ -terms the integral adaptive gains are

allowed to increase for as long as there is a tracking error. When the integral gains reach certain values, they have a stabilizing effect on the system and the tracking error begins to decrease. However, if the tracking error does not reach zero for some reasons, the integral gains will continue to increase and eventually diverge. On the other hand, with the forgetting coefficient matrices the integral gains increase as required (e.g. due to large tracking errors), and decrease when large gains are no longer necessary. In fact, with the σ -terms the integral gains are obtained as a first-order filtering of the tracking errors, and cannot diverge unless the tracking errors diverge.

Remark 3.3. In the general case given by (3.54), it can be shown that the term

$$(A^* - A)x^* + (B^* - B)u^* \quad (3.62)$$

is bounded. Nevertheless, (3.62) affects the proof of stability, and the tracking errors converge to the final magnitude of (3.62). However, it is clear that the Lyapunov derivative (3.54) is negative semidefinite for large e_x , which guarantees that the system is stable with respect to boundedness.

□

3.5 Decentralized Modified Simple Adaptive Control Development

To further decrease the number of operations required to implement the SAC algorithm in real time, Ulrich and de Lafontaine [138] proposed the idea of using a modified version of the SAC algorithm, referred to as the modified simple adaptive control (MSAC) law. The MSAC algorithm is obtained by retaining only the error-related adaptive gains $K_{P_e}(t)$ and $K_{I_e}(t)$. In fact, as mentioned by Barkana *et al.* [117], only the stabilizing control gain matrix $K_e(t)$ is absolutely required for the stability of the adaptive system. However, Ulrich and de Lafontaine [138] did not investigate the stability of the MSAC approach in their work, as the required ASP theorem for nonlinear systems had not yet been developed.

Adopting a MSAC control approach yields the following adaptive law

$$u = K_e(t)e_y = [K_{P_e}(t) + K_{I_e}(t)]e_y \quad (3.63)$$

where $K_{P_e}(t)$ and $K_{I_e}(t)$ are adapted with the decentralized adaptation law (3.25)-(3.26).

3.5.1 Error Dynamics

With the proposed DMSAC approach, since the tracking error e_y along the ideal trajectory is zero, the ideal control input vector u^* is also zero. Thus, the time derivative of (3.17) is

$$\dot{e}_x = \dot{x}^* - \dot{x} = A^*x^* - Ax - Bu \quad (3.64)$$

Adding and subtracting Ax^* to (3.64) and rearranging gives

$$\dot{e}_x = Ae_x + (A^* - A)x^* - Bu \quad (3.65)$$

Adding and subtracting $B\tilde{K}_e e_y$ to (3.65), and substituting e_y from (3.18), yields

$$\dot{e}_x = (A - B\tilde{K}_e C) e_x + (A^* - A)x^* - B(u - \tilde{K}_e e_y) \quad (3.66)$$

Substituting (3.63) into (3.66) gives

$$\dot{e}_x = (A - B\tilde{K}_e C) e_x + (A^* - A)x^* - B(K_e(t) - \tilde{K}_e) e_y \quad (3.67)$$

Finally, substituting $K_e(t)$ from (3.22) yields

$$\dot{e}_x = (A - B\tilde{K}_e C) e_x + (A^* - A)x^* - BK_{P_e} e_y - B(K_{I_e} - \tilde{K}_e) e_y \quad (3.68)$$

3.5.2 Stability Analysis

Theorem 3.3: The DMSAC algorithm given by (3.63) with adaptation mechanism (3.25)-(3.26) ensures that all system signals are bounded under closed-loop operation, and results in asymptotic convergence of the state and output tracking errors in the sense that

$$\|e_y\| \rightarrow 0 \text{ and } \|e_x\| \rightarrow 0 \text{ as } t \rightarrow \infty$$

Proof: Let $V \in \mathbb{R}$ be a continuously differentiable positive-definite symmetric function given by

$$V = e_x^T P e_x + \text{tr} \left[\left(K_{I_e}(t) - \tilde{K}_e \right) \Gamma_{I_e}^{-1} \left(K_{I_e}(t) - \tilde{K}_e \right)^T \right] \quad (3.69)$$

The time-derivative of (3.69) is obtained as

$$\begin{aligned} \dot{V} = & \dot{e}_x^T P e_x + e_x^T \dot{P} e_x + e_x^T P \dot{e}_x + \text{tr} \left[\dot{K}_{I_e}(t) \Gamma_{I_e}^{-1} \left(K_{I_e}(t) - \tilde{K}_e \right)^T \right] \\ & + \text{tr} \left[\left(K_{I_e}(t) - \tilde{K}_e \right) \Gamma_{I_e}^{-1} \dot{K}_{I_e}^T(t) \right] \end{aligned} \quad (3.70)$$

Substituting e_y from (3.18), $K_{P_e}(t)$ from (3.25), $\dot{K}_{I_e}(t)$ from (3.26) and \dot{e}_x from (3.68) into (3.70) gives

$$\begin{aligned} \dot{V} = & e_x^T \left[\dot{P} + P \left(A - B \tilde{K}_e C \right) + \left(A - B \tilde{K}_e C \right)^T P \right] e_x \\ & - 2e_x^T P B \text{diag} \{ C e_x e_x^T C^T \} \Gamma_{P_e} C e_x \\ & - e_x^T C^T \left[K_{I_e} - \tilde{K}_e \right]^T B^T P e_x - e_x^T P B \left[K_{I_e} - \tilde{K}_e \right] C e_x \\ & + \left[(A^* - A) x^* \right]^T P e_x + e_x^T P (A^* - A) x^* \\ & + \text{tr} \left[\text{diag} \{ C e_x e_x^T C^T \} \Gamma_{I_e} \Gamma_{I_e}^{-1} \left(K_{I_e}(t) - \tilde{K}_e \right)^T \right] \\ & + \text{tr} \left[\left(K_{I_e}(t) - \tilde{K}_e \right) \Gamma_{I_e}^{-1} \Gamma_{I_e} \text{diag} \{ C e_x e_x^T C^T \} \right] \end{aligned} \quad (3.71)$$

Using the ASP conditions (3.12) and (3.13), the expression in (3.71) can be simplified

as

$$\begin{aligned}
\dot{V} = & -e_x^T Q e_x \\
& - 2e_x^T C^T \text{diag} \{C e_x e_x^T C^T\} \Gamma_{P_e} C e_x \\
& - e_x^T C^T [K_{I_e} - \tilde{K}_e]^T C e_x - e_x^T C^T [K_{I_e} - \tilde{K}_e] C e_x \\
& + [(A^* - A) x^*]^T P e_x + e_x^T P (A^* - A) x^* \\
& + \text{tr} \left[\text{diag} \{C e_x e_x^T C^T\} \Gamma_{I_e} \Gamma_{I_e}^{-1} (K_{I_e}(t) - \tilde{K}_e)^T \right] \\
& + \text{tr} \left[(K_{I_e}(t) - \tilde{K}_e) \Gamma_{I_e}^{-1} \Gamma_{I_e} \text{diag} \{C e_x e_x^T C^T\} \right] \tag{3.72}
\end{aligned}$$

Due to the diagonal forms of the results inside the trace functions, the following terms cancel each other

$$\text{tr} \left[\text{diag} \{C e_x e_x^T C^T\} \Gamma_{I_e} \Gamma_{I_e}^{-1} (K_{I_e}(t) - \tilde{K}_e)^T \right] - e_x^T C^T [K_{I_e} - \tilde{K}_e]^T C e_x = 0 \tag{3.73}$$

and similarly

$$\text{tr} \left[(K_{I_e}(t) - \tilde{K}_e) \Gamma_{I_e}^{-1} \Gamma_{I_e} \text{diag} \{C e_x e_x^T C^T\} \right] - e_x^T C^T [K_{I_e} - \tilde{K}_e] C e_x = 0 \tag{3.74}$$

Thus, (3.72) can be simplified to

$$\begin{aligned}
\dot{V} = & -e_x^T Q e_x \\
& - 2e_x^T C^T \text{diag} \{C e_x e_x^T C^T\} \Gamma_{P_e} C e_x \\
& + [(A^* - A) x^*]^T P e_x + e_x^T P (A^* - A) x^* \tag{3.75}
\end{aligned}$$

As before, assuming that the system parameters vary slowly in comparison with the control dynamics results in

$$\dot{V} = -e_x^T Q e_x - 2e_x^T C^T \text{diag} \{C e_x e_x^T C^T\} \Gamma_{P_e} C e_x \tag{3.76}$$

The Lyapunov derivative \dot{V} in (3.76) is uniformly negative definite with respect to e_x , but only negative semidefinite with respect to the entire state space $[e_x, K(t)]$. Therefore, as before, the adaptive control system demonstrates asymptotic convergence of the states and output errors, and boundedness of the adaptive gains.

Remark 3.4. *By considering σ_e , the time-varying integral control gain matrix is obtained as (3.57). With this modification to the DMSAC algorithm, the Lyapunov derivative function becomes*

$$\begin{aligned} \dot{V} = & -e_x^T Q e_x - 2e_x^T C^T \text{diag} \{ C e_x e_x^T C^T \} \Gamma_{P_e} C e_x \\ & - 2\text{tr} \left[\sigma_e K_{I_e}(t) \Gamma_{I_e}^{-1} \left(K_{I_e}(t) - \tilde{K}_e \right)^T \right] \end{aligned} \quad (3.77)$$

and the same conclusions stated in Remark 3.2 can be drawn.

Remark 3.5. *Regarding the general case given by (3.75), the same conclusions stated in Remark 3.3 can be drawn.*

□

3.6 Application Examples

In this section, the applicability of both new control schemes to square and nonsquare systems is demonstrated.

3.6.1 Square Manipulator Systems

To facilitate the following demonstration, let the nonlinear rigid-joint dynamics of a space manipulator (2.5) be rewritten in the task space as follows

$$\Lambda(q)\ddot{x}_r(t) + \Pi(q, \dot{q})\dot{x}_r(t) = F(t) \quad (3.78)$$

where $\Lambda(q), \Pi(q, \dot{q}) \in \mathbb{R}^{2 \times 2}$ and $F(t) \in \mathbb{R}^2$ denote the PDS pseudo-inertia matrix, the centripetal-Coriolis matrix in task space, and the control force vector, which are respectively defined as

$$\Lambda(q) = J^{-T}(q)M(q)J^{-1}(q) \quad (3.79)$$

$$\Pi(q, \dot{q}) = J^{-T}C(q, \dot{q})J^{-1}(q) + \Lambda(q)J(q)\dot{J}^{-1}(q) \quad (3.80)$$

$$F(t) = J^{-T}(q)\tau(t) \quad (3.81)$$

where $\dot{J}^{-1}(q)$ is defined as

$$\dot{J}^{-1}(q) = \frac{d}{dt} \{J^{-1}(q)\} \quad (3.82)$$

The nonlinear system dynamics given by (3.78) can be expressed in a standard state-space representation with

$$A(q, \dot{q}) = \begin{bmatrix} 0 & I_2 \\ 0 & -\Lambda^{-1}(q)\Pi(q, \dot{q}) \end{bmatrix} \quad B(q) = \begin{bmatrix} 0 \\ \Lambda^{-1}(q) \end{bmatrix} \quad (3.83)$$

Defining the scaled-position-plus-velocity output matrix as

$$C \triangleq \begin{bmatrix} \alpha I_2 & I_2 \end{bmatrix} \quad (3.84)$$

where $\alpha \in \mathbb{R}$ is a known scaling factor related to the sensors, and the state vector is given by

$$x = \begin{bmatrix} x_r \\ \dot{x}_r \end{bmatrix} \quad (3.85)$$

It is easy to see the product CB is PDS, as follows

$$CB(q) = \begin{bmatrix} \alpha I_2 & I_2 \end{bmatrix} \begin{bmatrix} 0 \\ \Lambda^{-1}(q) \end{bmatrix} = \Lambda^{-1}(q) > 0 \quad (3.86)$$

Moreover, a simple selection of matrices that satisfies (3.3)-(3.5) is

$$M = \begin{bmatrix} I_2 \\ -\alpha I_2 \end{bmatrix} \quad N = \begin{bmatrix} I_2 & 0 \end{bmatrix} \quad (3.87)$$

Computing

$$A_z = NA(q, \dot{q})M = -I_2 \quad (3.88)$$

and thus

$$\dot{z} = A_z z = -z \quad (3.89)$$

which shows that the zero dynamics is stable and the nonlinear dynamics is minimum-phase. This demonstrates that a two-link rigid-joint manipulator system is ASP.

Similar to the depiction in Chapter 2, the reference model was designed to incorporate the desired input-output plant behavior, and aside from the scaling parameter which is assumed to be known, the reference model is not based on any modeling of the plant. The matrices A_m and B_m are designed in terms of the ideal damping ratio ζ and undamped natural frequency ω_n as in (2.16), but with output matrix defined as

$$C_m \triangleq \begin{bmatrix} \alpha I_2 & I_2 \end{bmatrix} \quad (3.90)$$

The following control gains and parameters were used

$$\begin{aligned} \Gamma_{P_e} &= 15I_2 & \Gamma_{I_e} &= 30I_2 & \Gamma_{P_x} &= \Gamma_{P_u} = 10I_4 & \Gamma_{I_x} &= \Gamma_{I_u} = 15I_4 \\ \sigma_e &= 0.018I_2 & \sigma_x &= \sigma_u = 0.5I_4 \end{aligned}$$

The integral structure of the integral time-varying gains is computed online via a standard Tustin algorithm. All integral control gains were initialized to zero, and the reference model parameters were selected as $\zeta = 0.9$, $\omega_n = 10$ rad/s and $\alpha = 2.5$. Two simulation experiments were conducted, the first without (DMSAC), and the second with (DSAC) the time-varying control gain matrices $K_x(t)$ and $K_u(t)$.

The trajectory tracking results obtained with the DMSAC controller, and the DSAC controller are depicted in Fig. 3.1 and 3.4, respectively. The positioning overshoots achieved with the DMSAC controller are 0.142 m, 0.115 m and 0.101 m for the first, second and third direction change respectively. Comparison to trajectory tracking results of zero overshoot for the proposed DSAC controller indicates that the DSAC yields improved tracking performance. This is also demonstrated in Figs. 3.2 and 3.5, where an increased damping of the tracking errors with smaller settling times are obtained with the DSAC strategy. However, this increase in performance comes at the expense of greater complexity in the controller structure. The successive increase in tracking performance along each side of the trajectory for the DMSAC strategy is explained by analyzing the adaptation history of the control gains depicted in Fig. 3.3, which shows that the gains are increasing after each direction change, and thereby providing improved tracking results. It must be noted that this particular behavior is mainly due to the specific control parameters selected herein, and that a different behavior could be obtained with different parameters. On the other hand, the time-varying control gains for the DSAC algorithm shown in Figs. 3.6 to 3.8 do not exhibit such behavior, thus providing similar tracking performance at each corner of the trajectory.

3.6.2 Nonsquare Manipulator Systems

As Theorem 3.1 is only applicable to square systems, the stability of the decentralized adaptive control algorithms developed in this chapter can only be guaranteed for these systems. Nevertheless, in this section the performance and the applicability of the proposed DSAC and DMSAC control methodologies to nonsquare systems is established in numerical simulations. To this end, consider the nonsquare robotic manipulator described by (2.5), with output vector given by

$$y = \begin{bmatrix} x_r \\ \dot{x}_r \end{bmatrix} \quad (3.91)$$

and with the output of the reference model given by

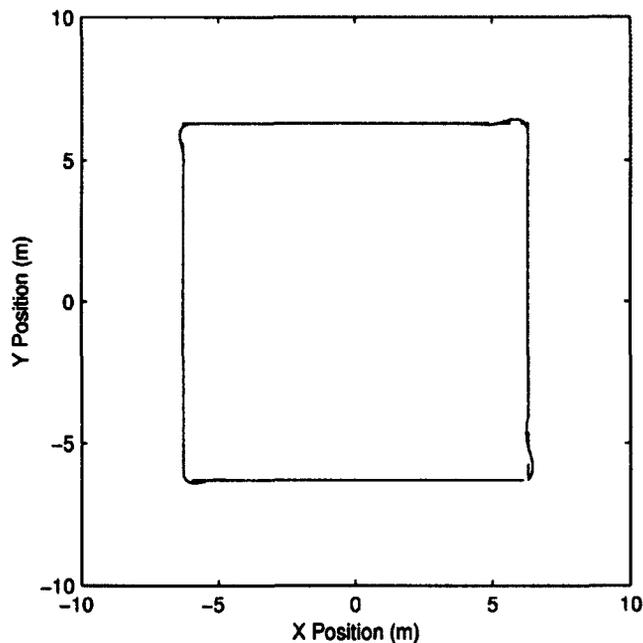


Figure 3.1: Trajectory tracking results obtained with the DMSAC controller applied to the square rigid-joint manipulator. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$.

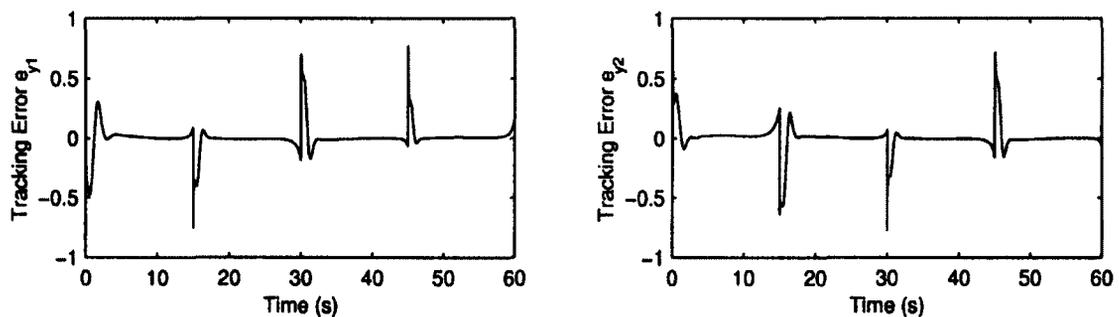


Figure 3.2: Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the DMSAC controller applied to the square rigid-joint manipulator.

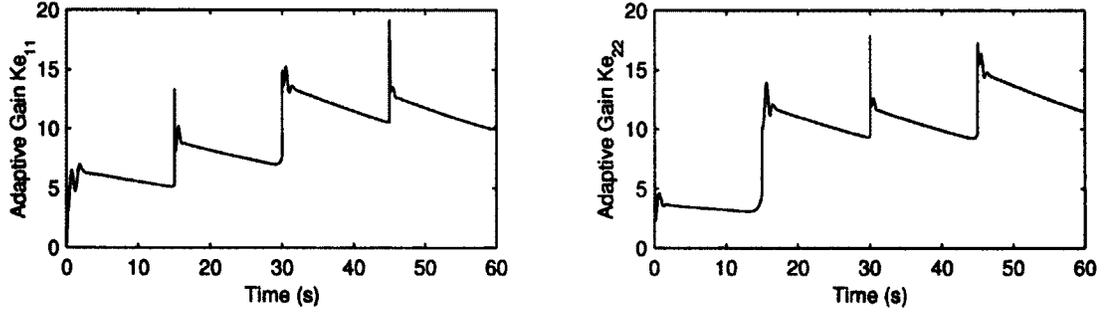


Figure 3.3: Adaptation history of the DMSAC controller gains $K_e(t)$ when applied to the square rigid-joint manipulator.

$$y_m = \begin{bmatrix} x_{r_m} \\ \dot{x}_{r_m} \end{bmatrix} \quad (3.92)$$

The dimensions of the time-varying gain matrices are adapted accordingly

$$K_{P_e}(t) = W^T \text{diag} \{e_y e_y^T\} \Gamma_{P_e} \quad (3.93)$$

$$\dot{K}_{I_e}(t) = W^T [\text{diag} \{e_y e_y^T\} \Gamma_{I_e} - \text{diag} \{\sigma_e W K_{I_e}\}] \quad (3.94)$$

$$K_{P_x}(t) = W^T \text{diag} \{e_y x_m^T\} \Gamma_{P_m} \quad (3.95)$$

$$\dot{K}_{I_x}(t) = W^T [\text{diag} \{e_y x_m^T\} \Gamma_{I_m} - \text{diag} \{\sigma_x W K_{I_m}\}] \quad (3.96)$$

$$K_{P_u}(t) = W^T \text{diag} \{e_y u_m^T\} \Gamma_{P_u} \quad (3.97)$$

$$\dot{K}_{I_u}(t) = W^T [\text{diag} \{e_y u_m^T\} \Gamma_{I_u} - \text{diag} \{\sigma_u W K_{I_u}\}] \quad (3.98)$$

where the scaling matrix W is

$$W = \begin{bmatrix} I_2 & I_2 \end{bmatrix}^T \quad (3.99)$$

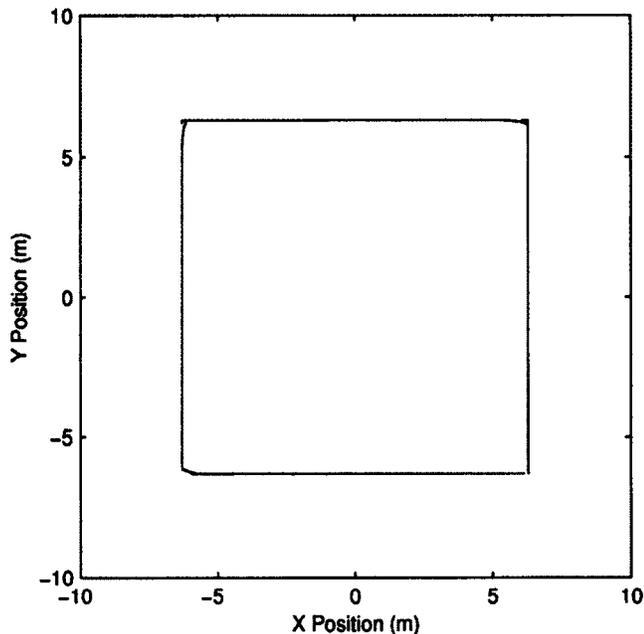


Figure 3.4: Trajectory tracking results obtained with the DSAC controller applied to the square rigid-joint manipulator. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$.

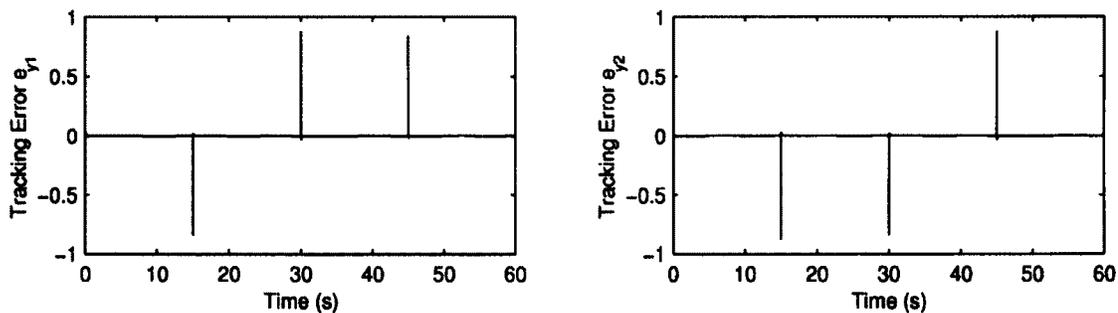


Figure 3.5: Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the DSAC controller applied to the square rigid-joint manipulator.

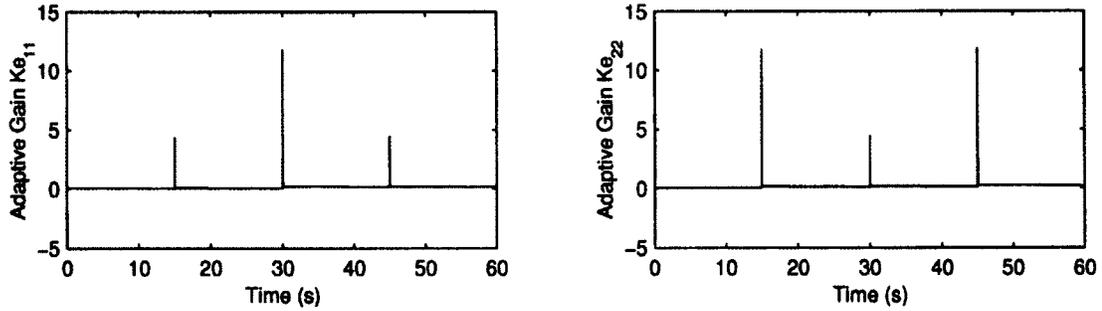


Figure 3.6: Adaptation history of the DSAC controller gains $K_e(t)$ when applied to the square rigid-joint manipulator.

As done previously, two simulation experiments were conducted: the first without (DMSAC) and the second with (DSAC) the time-varying control gain matrices $K_x(t)$ and $K_u(t)$. In the first case, the gains for the controller that resulted in the best tracking performance were determined as follows

$$\begin{aligned}\Gamma_{P_e} &= \text{diag} [120 \quad 120 \quad 15 \quad 15] \\ \Gamma_{I_e} &= \text{diag} [180 \quad 180 \quad 30 \quad 30] \\ \sigma_e &= 0.018I_4\end{aligned}$$

In the second case, the following adaptation parameters and coefficients were selected to provide satisfactory tracking performance

$$\begin{aligned}\Gamma_{P_e} = \Gamma_{I_e} &= \text{diag} [1500 \quad 1500 \quad 125 \quad 125] \\ \Gamma_{P_u} = \Gamma_{I_u} = \Gamma_{P_x} = \Gamma_{I_x} &= 100I_4 \\ \sigma_e = 0.018I_4 \quad \sigma_x = \sigma_u &= 0.5I_4\end{aligned}$$

As illustrated in Figs. 3.9 and 3.12, the DSAC approach provided better tracking performance than the DMSAC methodology: 0.001 m, 0.002 m, and 0.002 m for the first, second and third direction switch respectively, compared to 0.151 m, 0.129 m and 0.104 m at the first, second and third corner, respectively. This is also demonstrated

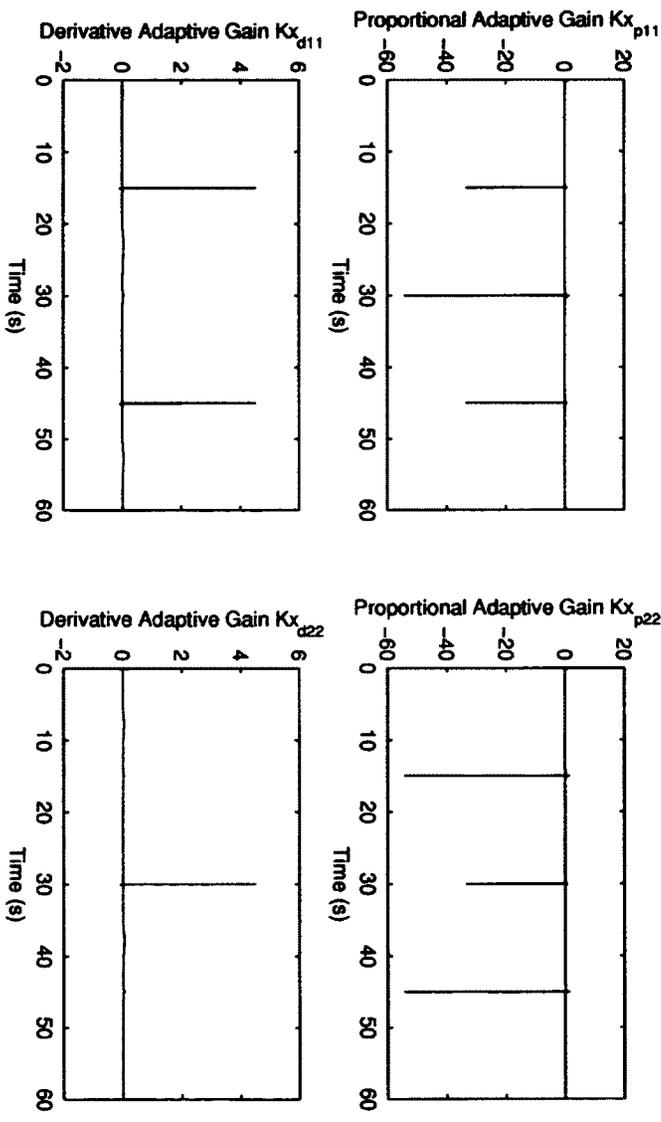


Figure 3.7: Adaptation history of the DSAC controller gains $K_{x_p}(t)$ and $K_{x_d}(t)$ when applied to the square rigid-joint manipulator.

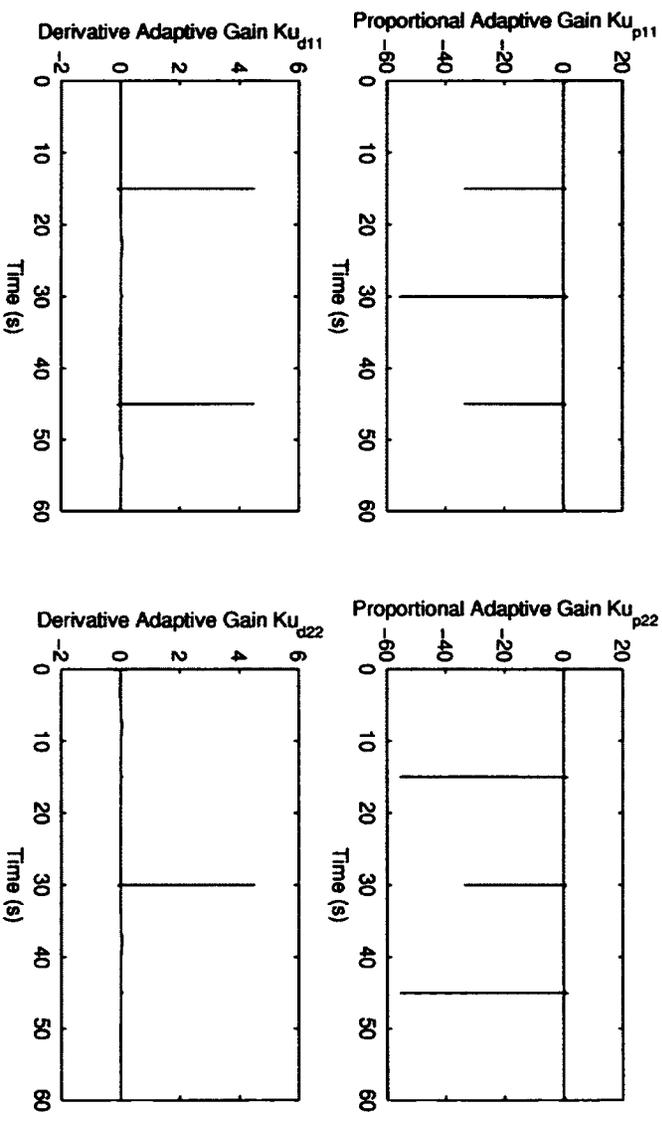


Figure 3.8: Adaptation history of the DSAC controller gains $K_{u_p}(t)$ and $K_{u_d}(t)$ when applied to the square rigid-joint manipulator.

by the smaller trajectory tracking errors obtained with the DSAC algorithm (see Figs. 3.10 and 3.13). However, as previously, this increase in performance comes at the expense of greater controller structure complexity. The time-varying control gains behave correctly in both control strategies, as shown in Figs. 3.11, and 3.14 to 3.16; they increase and decrease in accordance with the specific tracking situation, with sudden increases at each direction switch to reduce the positioning overshoots.

3.7 Summary

This chapter introduced the concept of almost strict passivity for nonlinear and non-stationary systems, based on which two new control methodologies, the *decentralized simple adaptive control* (DSAC) and the *decentralized modified simple adaptive control* (DMSAC), were proposed. The DMSAC methodology represents a simplification of the DSAC method, in that the feedforward control gain matrices are ignored. Both decentralized adaptive control methodologies consider only the diagonal elements of the control gain matrices. As a result, compared to the existing SAC scheme for nonlinear systems, the number of control parameters is reduced and the efficiency of the calculations is greatly improved. In addition, the Lyapunov stability analysis revealed that the rate of asymptotic convergence of the states and output tracking errors is improved, compared with the recently-developed reduced SAC control methodology for nonlinear systems. Further investigation was carried out in order to modify both methodologies to avoid divergence of the integral control gains in situations where perfect tracking cannot be achieved. Although stability can only be rigorously guaranteed for square systems, the applicability of both control schemes to square and nonsquare systems was validated in numerical simulations applied to rigid-joint manipulators. As anticipated, the injection of knowledge about the reference model in the DSAC control structure leads to improved trajectory tracking results compared with DMSAC. With the fuzzy logic-based adaptive controller presented in the previous chapter, these newly developed adaptive control methodologies form the basis for investigation of direct adaptive control schemes for flexible-joint manipulators.

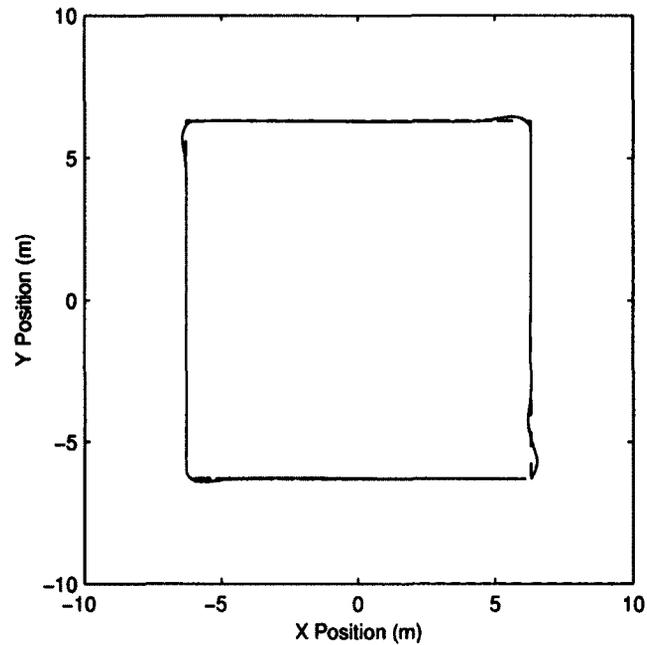


Figure 3.9: Trajectory tracking results obtained with the DMSAC controller applied to the nonsquare rigid-joint manipulator. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$.

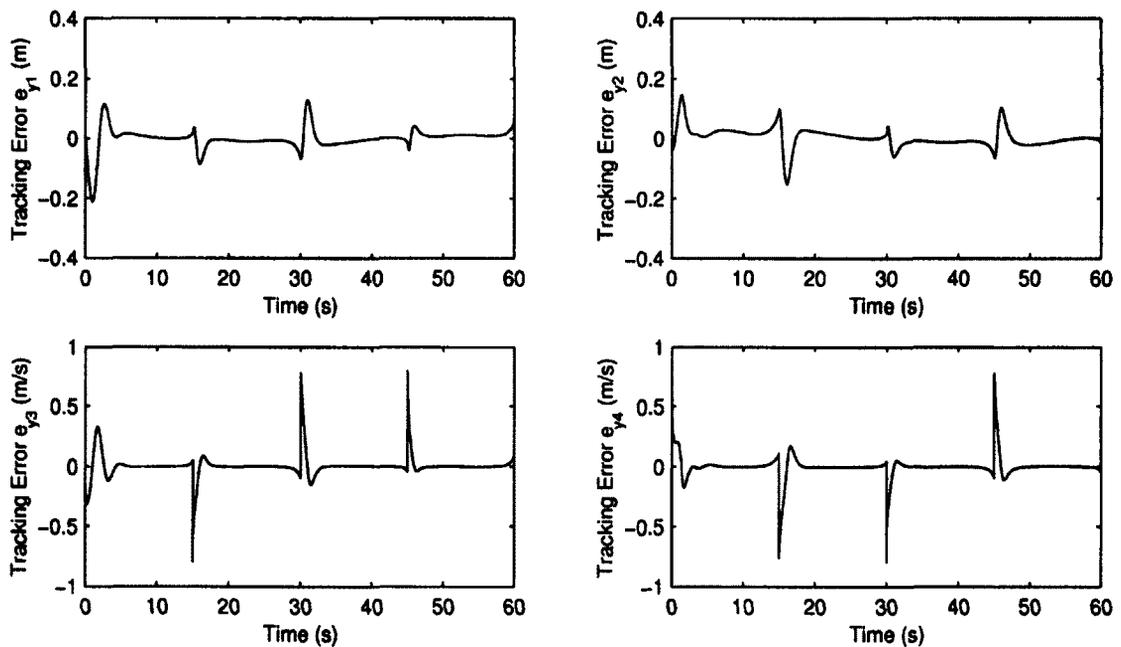


Figure 3.10: Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the DMSAC controller applied to the nonsquare rigid-joint manipulator.

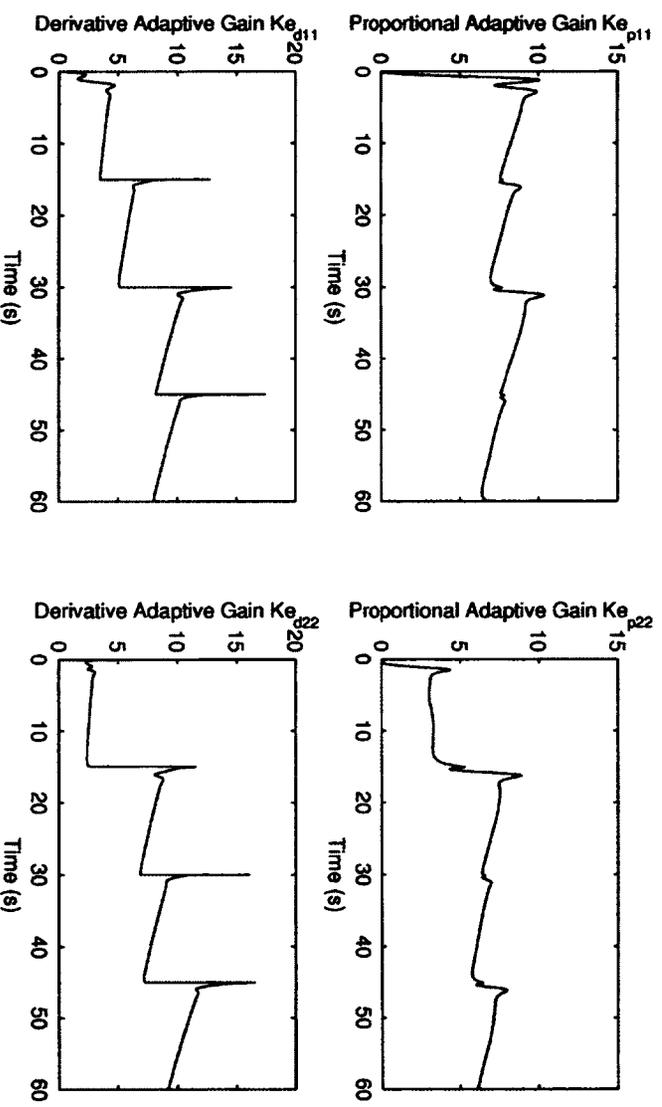


Figure 3.11: Adaptation history of the DMSAC controller gains $K_{e_p}(t)$ and $K_{e_d}(t)$ when applied to the nonsquare rigid-joint manipulator.

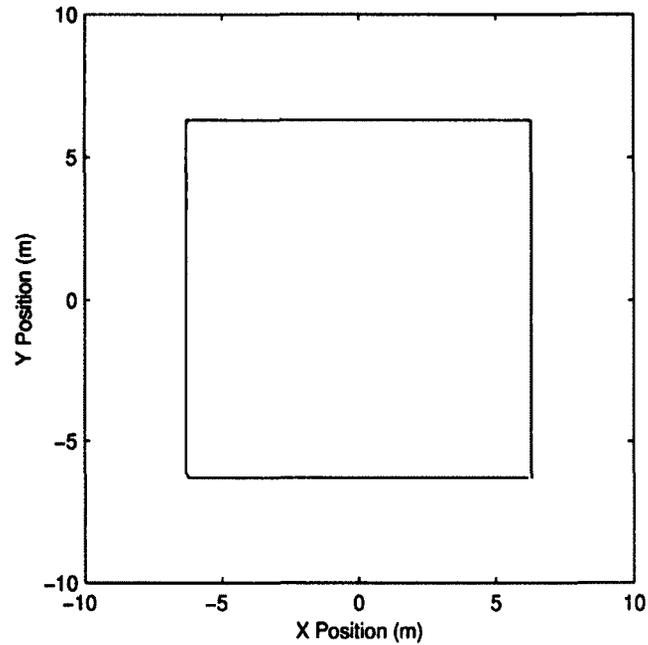


Figure 3.12: Trajectory tracking results obtained with the DSAC controller applied to the nonsquare rigid-joint manipulator. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$.

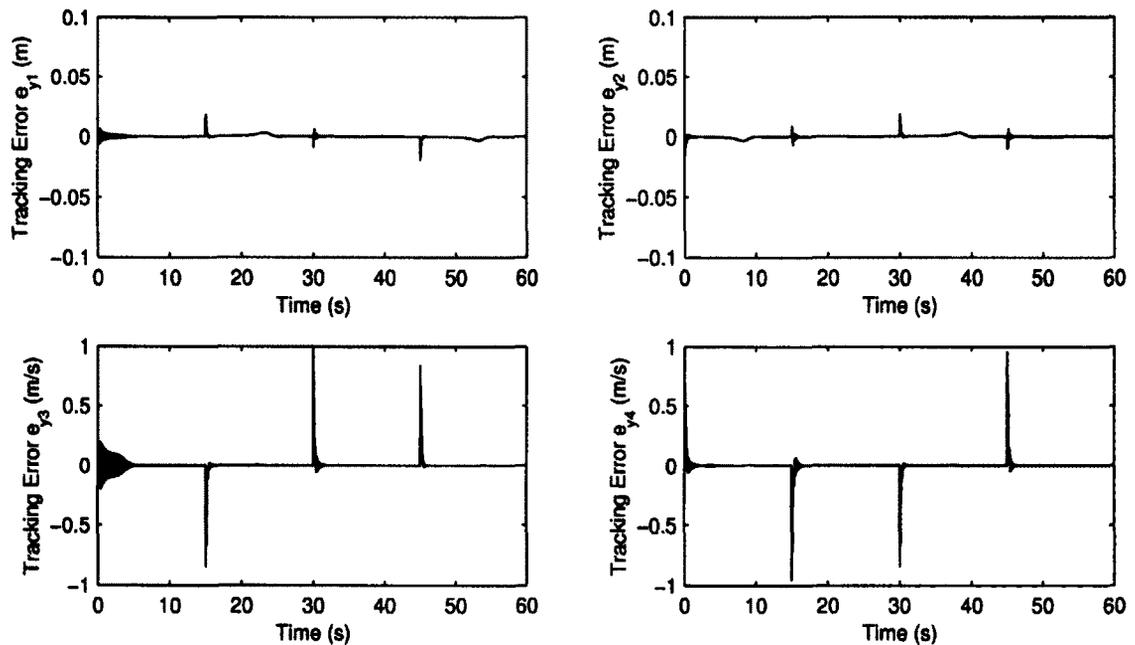


Figure 3.13: Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the DSAC controller applied to the nonsquare rigid-joint manipulator.

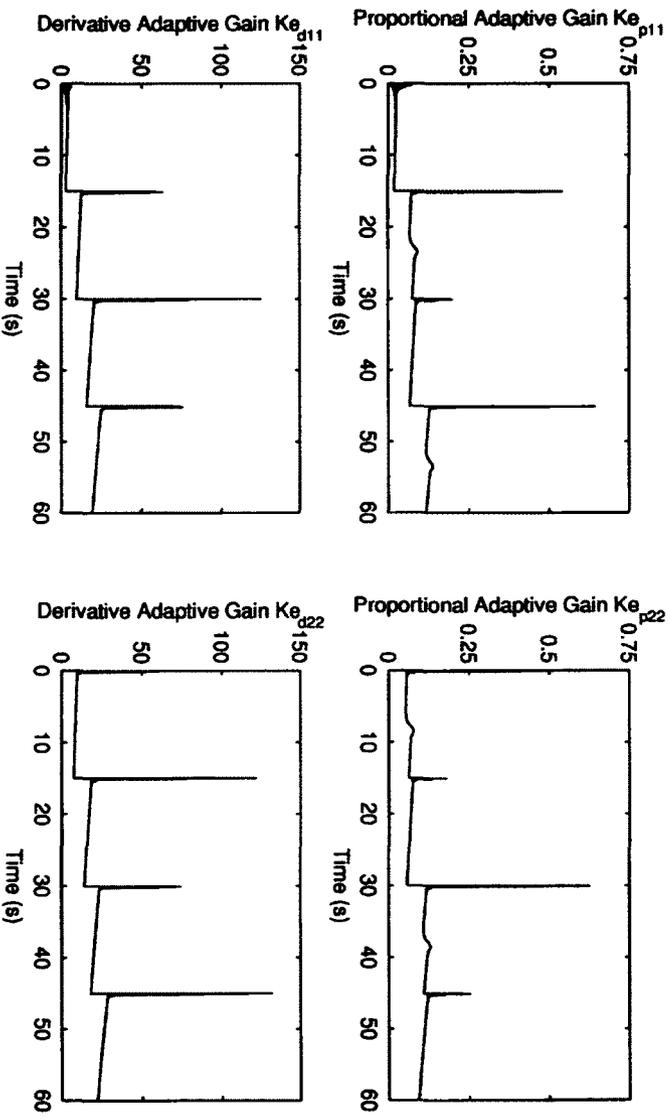


Figure 3.14: Adaptation history of the DSAC controller gains $K_{e_p}(t)$ and $K_{e_d}(t)$ when applied to the nonsquare rigid-joint manipulator.

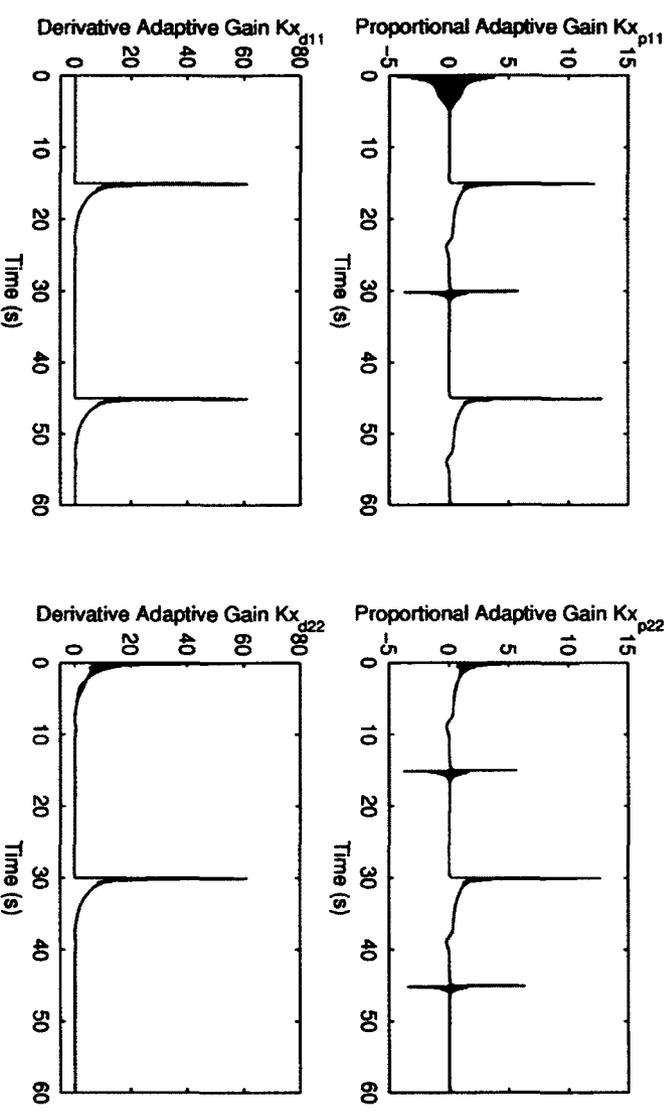


Figure 3.15: Adaptation history of the DSAC controller gains $K_{x_p}(t)$ and $K_{x_d}(t)$ when applied to the nonsquare rigid-joint manipulator.

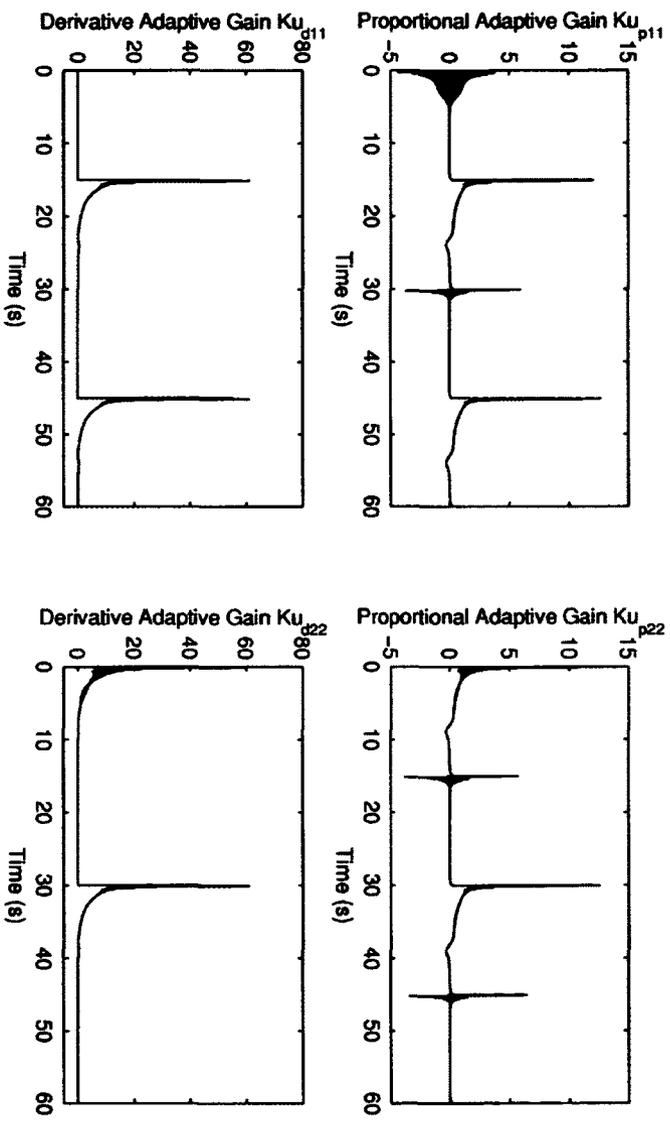


Figure 3.16: Adaptation history of the DSAC controller gains $K_{u_p}(t)$ and $K_{u_d}(t)$ when applied to the nonsquare rigid-joint manipulator.

Chapter 4

Direct Adaptive Composite Control Methodologies

This chapter addresses the problem of adaptive trajectory tracking control of space manipulators that exhibit elastic vibrations in their joints, and are subject to parametric uncertainties and modeling errors. It first presents a comprehensive study of the rigid and linear joint stiffness models, in order to propose a new dynamics formulation that includes nonlinear effects observed in experiments such as soft-windup and time-varying joint stiffness. Then, using a singular perturbation-based approach, it develops novel fuzzy logic and SAC-based adaptive composite control schemes for tracking the end-effector of a two-link flexible-joint space manipulator system. The new control schemes consist of a direct model reference adaptive system designed to stabilize the rigid manipulator dynamics, and a linear correction term to improve damping of vibrations at the joints. Numerical simulations compare the performance of the adaptive controllers with both their nonadaptive counterpart and a conventional nonlinear model-based flexible-joint control strategy.

4.1 Introduction

Several fuzzy logic techniques have been used to address the trajectory tracking control problem associated with flexible-joint manipulators. Fuzzy logic controllers are known to render good tracking performance in situations where mathematical modeling of the plant is uncertain; with flexible-joint manipulators for example. Goulet *et al.* [140] applied a complex multi-layer approach, with a conventional control bottom layer, a preprocess layer and an intelligent fuzzy logic top layer, to a two-link flexible-joint manipulator composed of revolute and prismatic joints. Ahmad *et al.* [141] used a fuzzy control system designed with triangular membership functions, to vary the voltage of the robot actuator in response to the end-effector position error and change-of-errors for a linearized state-space model of a single-link flexible-joint manipulator.

The performance of this fuzzy control scheme, combined with an input shaping technique, was evaluated in numerical simulations in a set-point control scenario. Park and Cho [142] designed a fuzzy model reference adaptive control strategy and applied it to a single-link elastic-joint manipulator that was mathematically modeled by a Takagi-Sugeno (TS) fuzzy model. The design of this controller was based on the TS dynamics model using a parallel distributed compensation technique. Poor performance occurs when tracking a sine-wave trajectory described in the joint space, as demonstrated by the ± 2 degree oscillating tracking error along the desired trajectory. Weiming *et al.* [143] proposed a fuzzy proportional-integral (PI) controller, with the link position and velocity errors chosen as inputs, and an incremental control input torque vector selected as the controller output. The strategy included an inner-loop system that controlled the variation of the incremental control gain, which significantly increased the complexity of the control scheme, largely due to the real-time integration of various signals.

In addition to conventional fuzzy logic controllers, a number of studies addressing the problem of adaptive trajectory control of flexible-joint manipulators are based on the singular perturbation approach [144], in which a flexible-joint robot exhibits a *two time-scale* behavior. With this approach, the flexible system can be separated into two subsystems which describe a fast and a slow dynamics. Notably, the singular perturbation theory allows the formulation of stability conclusions based on the stability properties of the separated subsystems, similar to the procedure of the analysis of cascaded control systems in the linear case. Using this dynamics representation, *composite controllers* consisting of a slow control term and a fast control term can be developed. In particular, several adaptive composite controllers consisting of a slow adaptive control law designed on the basis of the rigid-joint robot model, and a simple fast control law designed to dampen the elastic oscillations at the joints, were proposed by Spong [145, 146], Ghorbel *et al.* [24] and Chang and Daniel [147]. Ott *et al.* [148] verified the well-known Slotine and Li adaptive composite controller experimentally, and though good tracking results were achieved, the proposed strategy relied on joint torque measurements, suggesting the use of additional sensors. Huang *et al.* [149] presented an adaptive composite controller for a flexible-joint robot with

uncertain parameters and constrained motion, and though their tracking results were also good, the controller used force feedback measurements. Neural network adaptive composite control laws have also been proposed for flexible-joint manipulators. Subudhi and Morris [150] developed an approach in which a neural network replaced the inverse dynamics control for the slow dynamics, in order to cope with model uncertainties. Cao and de Silva [151] applied neural network-based adaptive composite control to a flexible manipulator which included both revolute and deployable links.

These existing adaptive composite control methodologies cope with parametric uncertainties by identifying the unknown robot parameters used explicitly in their control algorithm, an approach known as indirect adaptive control. For example, [151] used neural networks to approximate the unknown manipulator's inertia, and the Coriolis matrices used in their control law. One adverse consequence of such identification procedures is the increased computational burden associated with real-time computation of unknown parameters. This drawback could rule out the use of such adaptive controllers for space operations, where available computational resources are limited. In addition, these indirect adaptive controllers rely on an exact structure of the dynamics model, and could perform inadequately if the structure is different than the nominal one, due to modeling errors.

The key original contribution of this chapter is the design of direct adaptive control methodologies for flexible-joint robot manipulators. These novel control strategies employ diverse automatic controller tuning mechanisms, enabling them to adapt to different system conditions. Unlike existing adaptive flexible-joint composite controllers, the proposed designs do not require identification of unknown parameters, or mathematical models of the system to be controlled. Specifically, the controllers described in this chapter are based on the developments presented in Chapters 2 and 3. As a result, the new composite control methodologies deal with modeling errors and parameter uncertainties by time-varying the controller gains using a fuzzy logic system, the DSAC algorithm, and the DMSAC adaptation mechanism, thereby reducing the errors between an ideal model and the actual robot system. In addition, the proposed direct adaptive composite control schemes are simple to implement, and

render effective trajectory-tracking performance regardless of large parametric uncertainties and dynamics modeling errors. Moreover, unlike other flexible-joint control methodologies, the proposed adaptive controllers are independent of acceleration and torque signals.

Another deficiency of existing control schemes for flexible-joint manipulators is that they have been validated in numerical simulations using the classic Spong model [45]. Considered the foundation of nearly all work in the area of flexible-joint control, this basic dynamics formulation assumes that each joint is modeled as a linear torsional spring of constant stiffness. However, studies [37, 92–97] have revealed that joint mechanisms are far more complex than linear springs, and that they exhibit highly nonlinear effects. Therefore, in order to better capture the nonlinear effects observed in experiments, a new comprehensive dynamics model that simultaneously considers friction, nonlinear stiffness, soft-windup and inertial cross-coupling is proposed in this paper. Both models will be used in numerical simulations to validate the performance of the adaptive controllers.

This chapter is organized as follows: Section 4.2 presents the derivation of the nonlinear joint stiffness dynamics model, Section 4.3 describes the control objective and Section 4.4 presents the detailed development of the direct adaptive composite methodologies. Finally, the simulation studies in Section 4.5 demonstrate the performance of the proposed adaptive control systems, compared with a nonadaptive controller and a nonlinear model-based controller, in the presence of parametric uncertainties and modeling errors.

4.2 Dynamics Modeling

This section first reviews the linear joint stiffness equations of motion for a two-link flexible-joint manipulator, after which the details of the derivation of the proposed nonlinear joint dynamics formulation are provided. The linear joint stiffness model will be used for the design and preliminary validation of the adaptive controller performance, and the nonlinear joint stiffness model will be used to assess the robustness of the proposed flexible-joint adaptive composite control strategies to modeling errors.

As with previous studies on the control of flexible space manipulators [8–12, 14,

15, 99, 111], a stabilized platform (base) is assumed in this work. Also, gravitational potential energy is omitted for space robot applications.

4.2.1 Linear Joint Stiffness Dynamics

As with the rigid-joint model, the well-known dynamics equations proposed by Spong [45] for robots with links elastically coupled to the actuators (i.e. the rotors of DC motors) are derived in terms of kinetic and potential energies stored in the system by the Euler-Lagrange formulation. In this model, each joint is modeled as a linear torsional spring of constant stiffness, and the resulting dynamics of flexible-joint manipulators consists of two second-order differential equations. Due to the introduction of elastic coupling between the motor shafts and the links, the dynamics model has n -DOF link dynamics and n -DOF motor dynamics. The i th flexible joint, in which the rotor is directly coupled to its link, is shown schematically in Fig. 4.1. Referring to this figure, let $q_m \in \mathbb{R}^n$ be the vector denoting the angular displacements of the motor shaft angles, where the elastic joint vibration vector is defined as $q - q_m$. The joint flexibility effect is accounted for by augmenting the kinetic energy of the rigid-joint model with the kinetic energy of each rotor, which is assumed to be due to its own rotation only, and by considering the elastic potential energy of the flexible joints. The kinetic energy of the rotors $T_e \in \mathbb{R}$, and the elastic potential energy $U_e \in \mathbb{R}$ are respectively given by

$$T_e = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n J_{m_{ij}} \dot{q}_{m_i} \dot{q}_{m_j} = \frac{1}{2} \dot{q}_m^T J_m \dot{q}_m \quad (4.1)$$

$$U_e = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n k_{ij} (q_{m_i} - q_{m_j})(q_{m_i} - q_{m_j}) = \frac{1}{2} (q - q_m)^T k (q - q_m) \quad (4.2)$$

where $J_m \in \mathbb{R}^{n \times n}$ denotes the positive-definite motor inertia matrix and the stiffness of the flexible joints is modeled by $k \in \mathbb{R}^{n \times n}$, the diagonal positive-definite stiffness matrix of the joints. By combining elastic terms with rigid-dynamics equations, the following dynamics equations of a flexible-joint robot with revolute joints actuated directly by DC motors are obtained

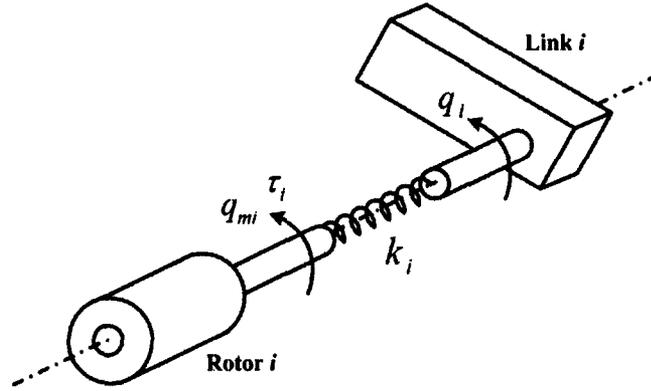


Figure 4.1: Flexible joint modeling

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} - k(q_m - q) = 0 \quad (4.3)$$

$$J_m\ddot{q}_m + k(q_m - q) = \tau \quad (4.4)$$

In such a dynamics model, the link dynamics (4.3) and the actuator dynamics (4.4) are only coupled by the elastic torque term $k(q_m - q)$.

Property 4.1. The linear joint stiffness dynamics model is a direct extension of the rigid-joint model, since the rigid model can be recovered from (4.3) and (4.4), when the joint stiffness tends to infinity. Indeed, assuming $k \rightarrow \infty$ implies $q = q_m$ and $\dot{q} = \dot{q}_m$.

Property 4.2. The linear joint stiffness robot dynamics can be expressed in a standard singular perturbation form [146], as shown in the following development. Let k be defined as

$$k \triangleq \frac{k_1}{\epsilon^2} \quad (4.5)$$

with $k_1 \in \mathbb{R}^{2 \times 2}$, and where $\epsilon \in \mathbb{R}$ is a small positive scalar. Also, let $L \in \mathbb{R}^2$ define the torque vector transmitted through the flexible joints as

$$L \triangleq k(q_m - q) = \frac{k_1}{\epsilon^2}(q_m - q) \quad (4.6)$$

This allows the dynamics equations (4.3) and (4.4) to be rewritten as a singular

perturbation system, as follows

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} = L \quad (4.7)$$

$$\epsilon^2 J_m \ddot{L} = k_1 (\tau - J_m \ddot{q} - L) \quad (4.8)$$

where L and \dot{L} are the fast variables, and where the link variables q and \dot{q} are the slow variables. Solving for \ddot{q} in (4.7), substituting in (4.8), and multiplying the resulting expression by J_m^{-1} gives

$$\epsilon^2 \ddot{L} = k_1 \{ J_m^{-1} \tau + M^{-1}(q)C(q, \dot{q})\dot{q} - (J_m^{-1} + M^{-1}(q)) L \} \quad (4.9)$$

Using the singular perturbation theory [152], the system (4.7) and (4.9) can be approximated by a quasi-steady state (slow) subsystem, and a boundary-layer (fast) subsystem. Letting $\epsilon \rightarrow 0$, the joint stiffness tends to infinity, which is equivalent to assuming that the transients due to the fast modes are instantaneous. With $\epsilon \rightarrow 0$ in (4.8) yields

$$L_s = \tau_s - J_m \ddot{q}_s \quad (4.10)$$

where the subscript s stands for slow variables defined in the slow time scale t , i.e. at $\epsilon \rightarrow 0$. Substituting (4.10) into (4.7) at $\epsilon \rightarrow 0$ results in the following quasi-steady state subsystem

$$(M(q_s) + J_m) \ddot{q}_s + C(q_s, \dot{q}_s)\dot{q}_s = \tau_s \quad (4.11)$$

Note that (4.11) has the form of the rigid-joint manipulator dynamics model, with the inertia matrix equal to $M(q_s) + J_m$.

Defining $\eta = L - L_s$ and $\tau_f = \tau - \tau_s$, the boundary-layer subsystem is obtained by expressing (4.9) in the fast time scale t' , and setting $\epsilon = 0$, as follows

$$\frac{d^2 L}{dt'^2} = k_1 \{ J_m^{-1} \tau + M^{-1}(q)C(q, \dot{q})\dot{q} - (J_m^{-1} + M^{-1}(q)) L \} \quad (4.12)$$

where the slow variables, q and \dot{q} , are considered frozen during the fast transients.

The steady-state of (4.12) with $\tau_f = 0$ is given by

$$0 = k_1 \{ J_m^{-1} \tau_s + M^{-1}(q) C(q, \dot{q}) \dot{q} - (J_m^{-1} + M^{-1}(q)) L_s \} \quad (4.13)$$

The boundary-layer subsystem, which is obtained by subtracting (4.13) from (4.12), is given by

$$\frac{d^2 \eta}{dt^2} = k_1 \{ J_m^{-1} \tau_f - (J_m^{-1} + M^{-1}(q)) \eta \} \quad (4.14)$$

Note that since the variable q is considered constant in (4.14), the boundary-layer subsystem of the flexible-joint manipulator is linear and time-invariant (LTI).

4.2.2 Nonlinear Joint Stiffness Dynamics

Since the development of the linear joint stiffness model, several experiments have been conducted to increase understanding of flexible effects in the joints of robotic manipulators, as accurate modeling is key to the successful design of advanced flexible-joint control laws. The most relevant nonlinear effects are related to nonlinear joint stiffness. To replicate experimental joint stiffness curves, most researchers [92, 94] recommended approximating the stiffness torque using a nonlinear cubic function, while others suggested a simple piecewise linear stiffness curve approximation [96]. A recent experiment by Li *et al.* [97] confirmed that the nonlinear cubic model better represents nonlinear stiffness effects. Therefore, in (4.3) and (4.4) the term $k(q_m - q)$ is replaced by a time-varying nonlinear cubic function [97]

$$k(q, q_m)(q_m - q) = a_1 \begin{bmatrix} (q_{m1} - q_1)^3 \\ (q_{m2} - q_2)^3 \end{bmatrix} + a_2(q_m - q) \quad (4.15)$$

Another important characteristic related to nonlinear joint stiffness was demonstrated in experiments by Kircanski and Goldenberg [95]. They observed that the torque-torsion characteristic was deformed toward the torque axis in the region from 0 to 1 Nm. In this region, the stiffness is lower due to the soft-windup effect, which can be mathematically modeled as a saddle-shaped function, as follows [95]

$$K_{sw}(q, q_m) = -k_{sw} \begin{bmatrix} e^{-a_{sw}(q_{m1}-q_1)^2} & 0 \\ 0 & e^{-a_{sw}(q_{m2}-q_2)^2} \end{bmatrix} \quad (4.16)$$

where $k_{sw} \in \mathbb{R}^{n \times n}$ and $a_{sw} \in \mathbb{R}$ denote a diagonal positive-definite matrix and a parameter defining the soft-windup function respectively.

In addition to nonlinear stiffness effects, it was determined that friction could not be neglected and must be modeled appropriately [153]. In the literature, several models of friction in joint mechanisms are available, most developed to reproduce friction curves observed in experiments. In this work, friction is assumed to have the following form as in [154, 155]

$$f(\dot{q}) = \gamma_1[\tanh(\gamma_2\dot{q}) - \tanh(\gamma_3\dot{q})] + \gamma_4 \tanh(\gamma_5\dot{q}) + \gamma_6\dot{q} \quad (4.17)$$

where $\gamma_i \in \mathbb{R} \forall i = 1, \dots, 6$ denotes positive constants defining different friction components. The continuous friction model in (4.17) has the following properties: 1) it is symmetric about the origin; 2) it has a static coefficient of friction; 3) it exhibits the Stribeck effect, where the friction coefficient decreases from the static coefficient of friction with increasing slip velocity near the origin; 4) it includes a viscous dissipation term; and 5) it has a Coulombic friction coefficient in the absence of viscous dissipation. To a good approximation, the static friction coefficient is given by $\gamma_1 + \gamma_4$, and the Stribeck effect is captured by $\tanh(\gamma_2\dot{q}) - \tanh(\gamma_3\dot{q})$. The Coulombic friction coefficient is given by $\gamma_4 \tanh(\gamma_5\dot{q})$, and the viscous dissipation is given by $\gamma_6\dot{q}$. Compared with other experimental-based friction models, this formulation offers two advantages. First, it efficiently captures major nonlinear friction effects observed in experiments without involving discontinuous or piecewise continuous functions; most friction models are either discontinuous or only piecewise continuous, which can be problematic in the development of real-time estimation and control systems. Second, this friction model was successfully validated by Lightcap [156], and Lightcap and Banks [157] in experimental work that required accurate estimation and control in a flexible-joint robot system, indicating that friction effects in flexible-joint mechanisms can be efficiently described by the continuous and symmetric model given by (4.17). Other experimental evidence that symmetric friction models are appropriate

for flexible-joint manipulator applications includes the work of ElMaraghy *et al.* [69].

Another extension to the linear joint stiffness model can be made by removing the simplifying assumption of Spong's model regarding the kinetic energy of the rotors. In reality, the kinetic energy of the second rotor is also influenced by the rotation of the first link \dot{q}_1 , that is,

$$T_{e_2} = \frac{1}{2} J_{m_2} (\dot{q}_1^2 + \dot{q}_{m_2}^2) \quad (4.18)$$

In view of the above, the following nonlinear joint stiffness dynamics model for flexible-joint manipulator systems, which overcomes the limitations of the model given in (4.3) and (4.4) by incorporating prior experimental knowledge, is obtained herein as follows:

$$M(q)\ddot{q} + S\ddot{q}_m + C(q, \dot{q})\dot{q} + f(\dot{q}) - k(q, q_m)(q_m - q) = 0 \quad (4.19)$$

$$S^T \ddot{q} + J_m \ddot{q}_m + k(q, q_m)(q_m - q) = \tau \quad (4.20)$$

where the strictly upper triangular matrix $S \in \mathbb{R}^{n \times n}$ represents the inertial couplings between motor and link accelerations introduced by the dependency of the second rotor's kinetic energy on the first link's angular velocity. The last term on the left side of (4.19) and (4.20) can be rewritten as

$$k(q, q_m)(q_m - q) = a_1 \begin{bmatrix} (q_{m_1} - q_1)^3 \\ (q_{m_2} - q_2)^3 \end{bmatrix} + a_2(q_m - q) - k_{sw} \begin{bmatrix} e^{a_{sw}(q_{m_1} - q_1)^2} & 0 \\ 0 & e^{a_{sw}(q_{m_2} - q_2)^2} \end{bmatrix} \quad (4.21)$$

Similar comprehensive dynamics models have been used by others in the literature to validate flexible-joint control systems, including: Moghaddam and Goldenberg [72] to validate an H_∞ controller, and ElMaraghy *et al.* [69] to assess the performance of a linear robust controller designed with linear-quadratic-Gaussian/loop transfer-recovery techniques. However, the first disregards the inertial coupling and assumes only static and viscous friction terms, and the second neglects the soft-windup effect and uses a simplified friction model, which is linear in its parameters.

4.3 Control Objective

The control objective consists in designing direct adaptive composite controllers which ensure that the output vector $y(t) \in \mathbb{R}^4$ of the flexible-joint manipulator system tracks the time-varying reference model output $y_m(t) \in \mathbb{R}^4$, despite uncertainties and modeling errors in the dynamics model. To quantify this objective, a tracking error, denoted $e_y(t) \in \mathbb{R}^4$, is defined as

$$e_y \triangleq y_m - y \quad (4.22)$$

where the reference model defines the ideal response to the desired square trajectory. As in previous chapters, the reference model is expressed in terms of the ideal damping ratio ζ , and undamped natural frequency ω_n .

4.4 Composite Control Strategies Development

The following proposed control strategies all use the composite control design methodology. As shown in Fig. 4.2, this methodology consists of separating the control input into a slow and a fast control term, as follows

$$\tau = \tau_s(q, \dot{q}) + \tau_f(\eta, \dot{\eta}) \quad (4.23)$$

where the slow control term τ_s stabilizes the quasi-steady state subsystem, and the fast control term τ_f is applied to the boundary-layer subsystem. Applied to flexible-joint manipulators, composite controllers typically consist of a fast control torque that adds damping to the elastic vibrations at the joints, and a slow control torque which controls the rigid dynamics. As mentioned previously, the quasi-steady state subsystem has the form of an equivalent rigid-joint manipulator dynamics model. Thus, any controllers for rigid-joint robots may be considered for τ_s , providing they satisfy some considerations discussed below.

An important theorem for composite control systems, known as Tychonov's theorem [144], is presented in the following. This theorem is based on equation (4.11)

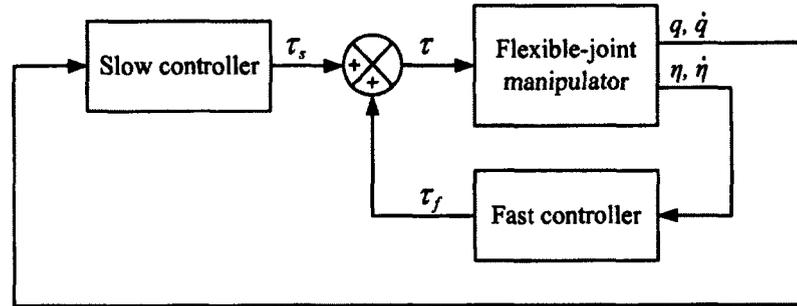


Figure 4.2: Block diagram representation of the composite control design methodology applied to flexible-joint manipulators, where τ_s controls the rigid dynamics and τ_f improves damping of the elastic vibrations at the joints.

and (4.14) and will be useful for justifying the flexible-joint controller design methodologies.

Theorem 4.1. If the quasi-steady state subsystem (4.11) has a unique solution q_s defined on an interval $t \in [0, t_1]$, and if the boundary-layer subsystem (4.14) is exponentially stable, uniformly in (t, q_s) , then, there exists ϵ^* such that $\forall \epsilon \in [0, \epsilon^*]$, the solution q, L of the exact system satisfies

$$L = L_s + \eta + O(\epsilon) \quad (4.24)$$

$$q = q_s + O(\epsilon) \quad (4.25)$$

Similar conclusions are also applicable to \dot{L} , and \dot{q} .

Remark 4.1. To clarify the above theorem, the composite control objective consists of designing: (1) a fast control term that guarantees the exponential stability of the closed-loop boundary-layer subsystem, and (2) a slow control term that guarantees convergence of the tracking error for the closed-loop quasi-steady state subsystem. This ensures that the tracking error will not deviate more than of the order $O(\epsilon)$ from its quasi-steady state on a finite time interval. Thus, the smaller ϵ is, the greater the tracking accuracy. From (4.5), it can be concluded that better tracking results are achieved when the joint stiffness matrix k is greater; in other words, when the boundary-layer is considerably faster than the quasi-steady state subsystem.

Remark 4.2. *Tychonov's theorem above holds only for a finite time interval. Although an extension of this theorem to the infinite time interval [144] does exist, and may be applied in an attempt to show asymptotic stability for sufficiently small ϵ , it requires exponential stability of the quasi-steady state subsystem.*

□

By virtue of Tychonov's theorem defined above, the control torque separation approach results in two important practical design considerations [158]. First, to ensure that the quasi-steady state subsystem has a unique solution, τ_s must only be a function of the state of the quasi-steady state model, and must not contain a feedback of the control torque τ . The rigid-joint adaptive fuzzy control law and the DSAC and DMSAC control algorithms satisfy this design consideration, thus providing adequate candidate control laws for use as slow control terms in a composite control scheme for flexible-joint manipulators. In addition, when applied to square systems it was demonstrated in Chapter 3 that the DSAC and DMSAC control schemes achieve asymptotic stability, thereby guaranteeing convergence of the tracking errors. While the closed-loop stability of the rigid-joint fuzzy adaptive control methodology developed in Chapter 2 cannot be formally guaranteed, the applicability of a composite control scheme using a similar fuzzy logic adaptation system as the slow control term is nevertheless presented in the following subsection.

The second design consideration imposed by Tychonov's theorem implies that the fast control torque τ_f must be designed such that it does not affect the quasi-steady state model. To do so, τ_f must meet the following condition

$$\tau_f|_{\epsilon \rightarrow 0} = 0 \quad (4.26)$$

Recalling that the boundary-layer subsystem is LTI, a simple linear correction of the form

$$\tau_f = K_v (\dot{q} - \dot{q}_m) \quad (4.27)$$

is selected as the fast control term, where $K_v \in \mathbb{R}^{2 \times 2}$ is a constant diagonal control gain that provides additional damping of the elastic vibrations at the joints, and is

defined as

$$K_v \triangleq \frac{k_2}{\epsilon} \quad (4.28)$$

where $k_2 \in \mathbb{R}^{2 \times 2}$. Solving for $(\dot{q} - \dot{q}_m)$ in (4.6) and substituting the result, as well as (4.28), into (4.27), yields

$$\tau_f = -\epsilon k_2 k_1^{-1} \dot{L} \quad (4.29)$$

Since L_s is considered constant in the boundary-layer subsystem, (4.29) becomes

$$\tau_f = -\epsilon k_2 k_1^{-1} \dot{\eta} \quad (4.30)$$

which clearly demonstrates that the chosen τ_f does not affect the quasi-steady state subsystem, is equal to 0 when $\epsilon \rightarrow 0$, and thus qualifies as a fast control term.

In addition, a fast control term of the form (4.27) yields an exponentially stable closed-loop boundary-layer subsystem, as demonstrated by the following theorem [146].

Theorem 4.2. The linear control law given by (4.27) results in an exponentially stable closed-loop boundary-layer subsystem.

Proof: Substituting (4.30) into (4.14) yields

$$\frac{d^2 \eta}{dt'^2} + J_m^{-1} k_2 \frac{d\eta}{dt'} + k_1 (J_m^{-1} + M^{-1}(q)) \eta = 0 \quad (4.31)$$

Equation (4.31) represents the closed-loop fast subsystem. Since k_1 , k_2 , and J_m are positive definite diagonal matrices, and $M^{-1}(q)$ is uniformly positive definite in q , the following positive definite Lyapunov function is considered

$$V_f = \frac{1}{2} \eta'^T \eta' + \frac{1}{2} \eta^T k_1 (J_m^{-1} + M^{-1}(q)) \eta \quad (4.32)$$

where η' denotes $d\eta/dt'$. The fast time scale derivative of (4.32) is

$$\frac{dV_f}{dt'} = \eta'^T \eta'' + \eta'^T k_1 (J_m^{-1} + M^{-1}(q)) \eta^T \quad (4.33)$$

Evaluating (4.33) along (4.31) results in

$$\frac{dV_f}{dt'} = -\eta^T J_m^{-1} k_2 \eta' \leq 0 \quad (4.34)$$

Global asymptotic stability of the closed-loop boundary-layer subsystem then follows from the Lyapunov-LaSalle theorem. The stability is also exponential for frozen q , since the system is linear and is uniform due to the uniform positive definitiveness of $M(q)$.

□

4.4.1 Nonadaptive Composite Control

The following nonadaptive composite control strategy using the intuitive transpose Jacobian nonadaptive control law presented in Section 2.3 as the slow control term is used as a benchmark flexible-joint controller.

$$\tau = J^T(q)K_e y_e + K_v(\dot{q} - \dot{q}_m) \quad (4.35)$$

where $y_e \in \mathbb{R}^4$ represents the tracking error defined previously by (2.12). In (4.35), $K_e \in \mathbb{R}^{2 \times 4}$ can be defined as

$$K_e \triangleq \begin{bmatrix} K_p & K_d \end{bmatrix} \quad (4.36)$$

where $K_p, K_d \in \mathbb{R}^{2 \times 2}$ are the constant proportional and derivative diagonal control gain matrix, respectively. A block diagram depiction of (4.35) is shown in Fig. 4.3.

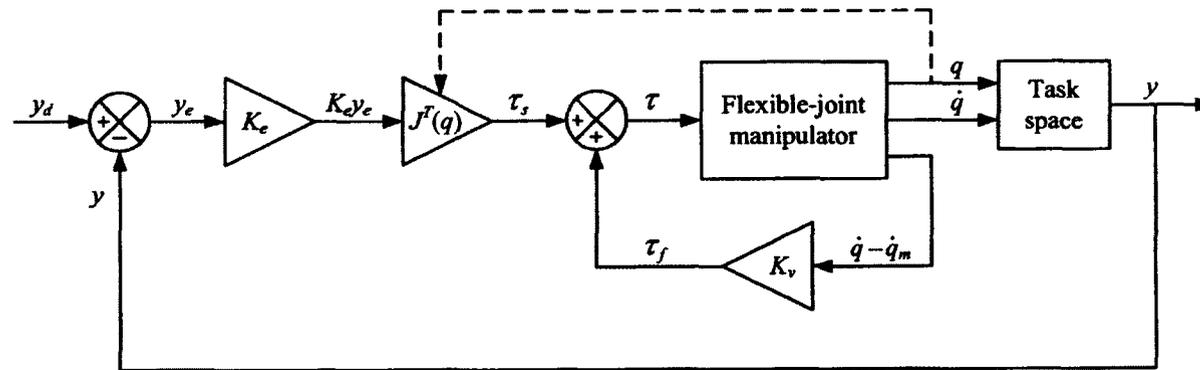


Figure 4.3: Block diagram representation of the nonadaptive transpose Jacobian composite controller.

4.4.2 Fuzzy Logic-Based Adaptive Composite Control

Adaptive control is achieved by replacing the constant control gain K_e in (4.36) with a time-varying control gain, which results in the following adaptive control law

$$\tau = J^T(q)K_e(t)e_y + K_v(\dot{q} - \dot{q}_m) \quad (4.37)$$

where, $K_e(t) \in \mathbb{R}^{2 \times 4}$ can be defined in terms of a time-varying proportional $K_p(t) \in \mathbb{R}^{2 \times 2}$ and a derivative control gain $K_d(t) \in \mathbb{R}^{2 \times 2}$, as follows

$$K_e(t) \triangleq \begin{bmatrix} K_p(t) & K_d(t) \end{bmatrix} \quad (4.38)$$

Adaptation of these two control gains is achieved via a fuzzy logic mechanism, which updates the gains as a function of the tracking errors between the actual plant outputs and the reference model outputs. Adopting a decentralized approach, $K_p(t)$ and $K_d(t)$ are updated as follows

$$K_p(t) = \text{diag} \left[\lambda_1(t)h_1, \lambda_2(t)h_2 \right] \quad (4.39)$$

$$K_d(t) = \text{diag} \left[\lambda_3(t)h_3, \lambda_4(t)h_4 \right] \quad (4.40)$$

where the time-varying control parameters $\lambda_q \in \mathbb{R} \forall q = 1, \dots, 4$ represent the outputs of a normalized fuzzy logic system (FLS). As shown in (4.39) and (4.40), each λ_q is dedicated to the control of a single element of the tracking error vector along a single axis, resulting in a decentralized approach. More specifically, the proportional-based control parameters λ_1 and λ_2 are applied to the position tracking errors e_{y_1} and e_{y_2} respectively, whereas the derivative-based control parameters λ_3 and λ_4 are applied to the velocity tracking errors e_{y_3} and e_{y_4} respectively. Whenever possible, the adaptation mechanism of each control parameter is based on its associated tracking error and its time-derivative. Thus, since no acceleration measurements are assumed, the adaptation mechanisms of λ_3 and λ_4 are based only on e_{y_3} and e_{y_4} respectively. Figures 4.4 and 4.5 respectively illustrate the block scheme implementation of the direct fuzzy adaptive system, and the details of the control gains computation are

described in the following subsections. In contrast to Chapter 2, the fuzzy adaptation mechanism presented in this chapter was implemented in a script file for MATRIXx, without using a pre-programmed *fuzzy logic block*. For this reason, a more detailed treatment of the fuzzy logic adaptation mechanism, including the relevant equations, is provided in the following. The details of the simulation code to calculate the time-varying parameters λ_q can be found in Appendix B.

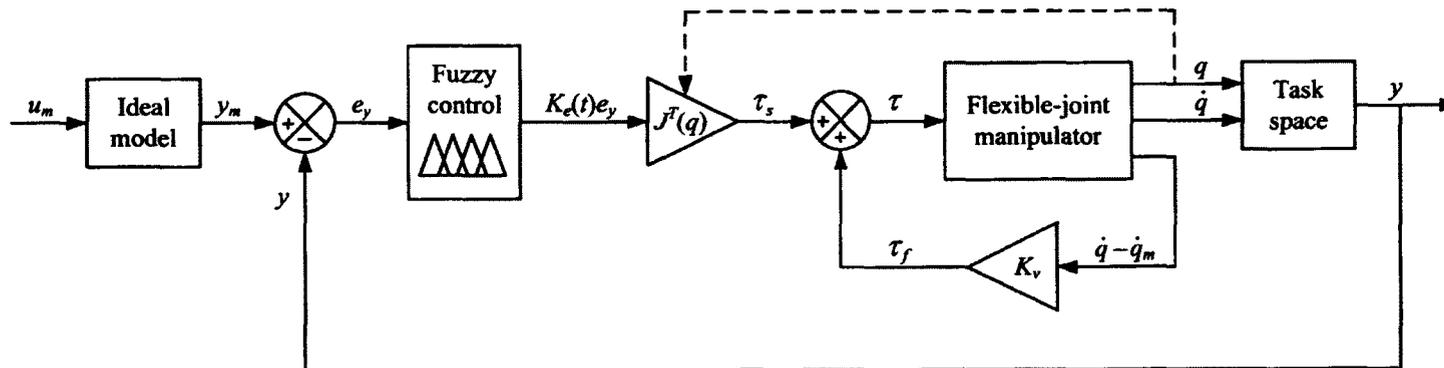


Figure 4.4: Block diagram representation of the fuzzy adaptive composite controller.

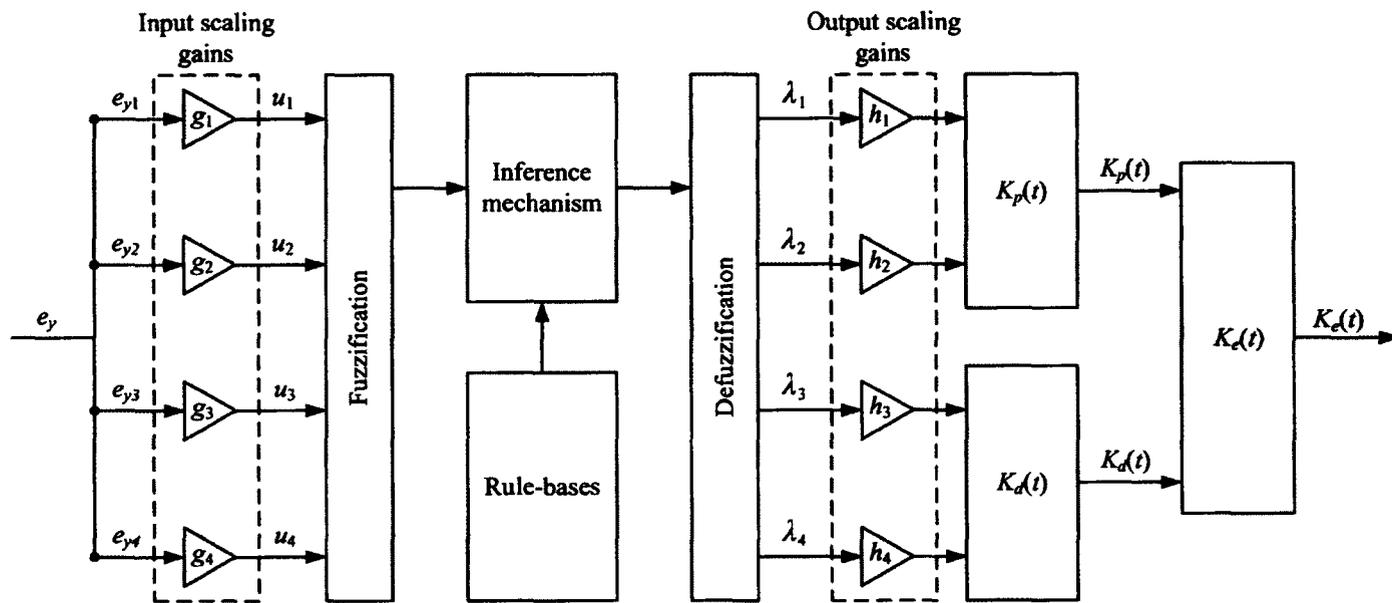


Figure 4.5: Computation details of the fuzzy adaptive control gain $K_e(t)$.

Fuzzification

The universes of discourse are $\pm 2/3$ m for e_{y_1} and e_{y_2} , ± 10 m/s for e_{y_3} and e_{y_4} , and 0 to 1 for λ_q . The input scaling gains $g_i \in \mathbb{R}$, are selected as $g_1 = g_2 = 1.5$ and $g_3 = g_4 = 0.1$, such that the universe of discourse U_i for the normalized crisp inputs $u_i \in \mathbb{R} \forall i = 1, \dots, 4$, is defined in $U_i = [-1 \ 1]$. Constant linguistic variables denoted by \tilde{u}_i and $\tilde{\lambda}_q$, are used to describe the time-varying crisp inputs u_i and outputs λ_q , respectively. Here, $\tilde{u}_1 = \textit{position tracking error along the x-axis}$, $\tilde{u}_2 = \textit{position tracking error along the y-axis}$, $\tilde{u}_3 = \textit{velocity tracking error along the x-axis}$, $\tilde{u}_4 = \textit{velocity tracking error along the y-axis}$, $\tilde{\lambda}_1 = \textit{proportional control parameter along the x-axis}$, $\tilde{\lambda}_2 = \textit{derivative control parameter along the y-axis}$, $\tilde{\lambda}_3 = \textit{derivative control parameter along the x-axis}$, and $\tilde{\lambda}_4 = \textit{derivative control parameter along the y-axis}$. Linguistic variable \tilde{u}_i take on elements from the set of nine linguistic values denoted by $\tilde{A}_i = \{ \tilde{A}_i^j : i = 1, \dots, 4 : j = 1, \dots, 9 \}$, where \tilde{A}_i^j denotes the j^{th} linguistic value. Similarly, $\tilde{B}_i^j \forall i = 1, 2$ denotes the j^{th} linguistic value for linguistic variables $\tilde{\lambda}_1$ and $\tilde{\lambda}_2$, both of which take on elements from the set of nine linguistic values denoted by $\tilde{B}_{1,2} = \{ \tilde{B}_i^j : i = 1, 2 : j = 1, \dots, 9 \}$, and $\tilde{B}_i^j \forall i = 3, 4$ denotes the j^{th} linguistic value for linguistic variables $\tilde{\lambda}_3$ and $\tilde{\lambda}_4$, both of which take on elements from the set of five linguistic values denoted by $\tilde{B}_{3,4} = \{ \tilde{B}_i^j : i = 3, 4 : j = 1, \dots, 5 \}$. Similar to Green and Sasiadek [15], linguistic values are defined as *negative very high (NVH)*, *negative high (NH)*, *negative low (NL)*, *negative very low (NVL)*, *zero (ZERO)*, *positive very low (PVVL)*, *positive very low (PVL)*, *positive low (PL)*, *positive medium (PM)*, *positive high (PH)*, *positive very high (PVH)*, *positive very very high (PVVH)* and *positive maximum (PMAX)*.

Through fuzzification, crisp input u_i are first mapped into fuzzy sets denoted by A_i^j , and defined as

$$A_i^j = \left\{ \left(u_i, \mu_{A_i^j}(u_i) \right) : u_i \in U_i \right\} \quad i = 1, \dots, 4 \quad j = 1, \dots, 9 \quad (4.41)$$

where the membership function $\mu_{A_i^j}(u_i)$ associated with \tilde{A}_i^j maps U_i to $[0 \ 1]$, in order to determine the degree of certainty that u_i with linguistic description \tilde{u}_i may be classified linguistically as \tilde{A}_i^j . All input u_i and output variables λ_q are designed with

Gaussian membership functions. Notably, unlike triangular or trapezoidal functions, Gaussian membership functions render non-zero degrees of membership over the entire universe of discourse. Moreover, a recent work by Green and Sasiadek [99] established heuristically that Gaussian functions result in improved tracking results. In this work, input Gaussian membership functions are modeled as

$$\mu_{A_i^j}(u_i) = e^{-\frac{1}{2} \left(\frac{u_i - c_{A_i^j}}{\sigma_{A_i^j}} \right)^2} \quad i = 1, \dots, 4 \quad j = 1, \dots, 9 \quad (4.42)$$

where $c_{A_i^j}, \sigma_{A_i^j} \in \mathbb{R}$ represent the center and width (shape) of the functions, respectively. Output Gaussian functions $\mu_{B_i^j}(\lambda_q)$ are also described mathematically with the same model. The resulting membership functions for input variables are shown in Fig. 4.6, and the membership functions for output variables in Figs. 4.7 and 4.8. Note that these membership functions are defined for normalized input and output variables, suggesting that the left-most membership functions saturate (peak) at -1, and the right-most at +1.

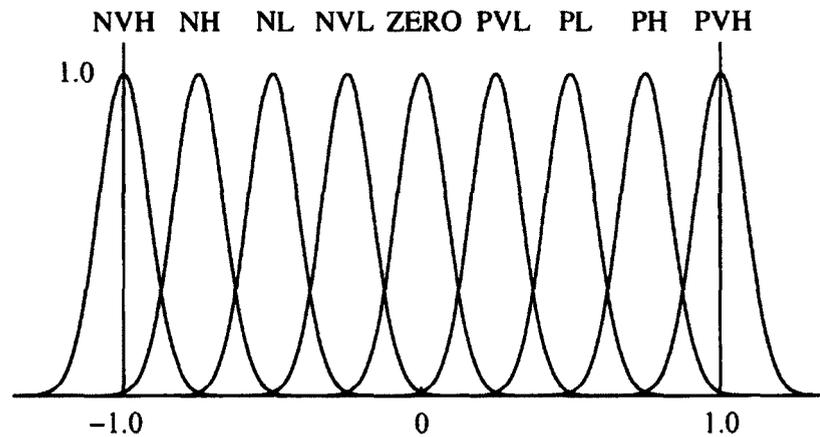


Figure 4.6: Membership functions for normalized input variables u_1, u_2, u_3 and u_4

Rule-Bases

The previously-defined linguistic values are also used to generate linguistic rules of the form

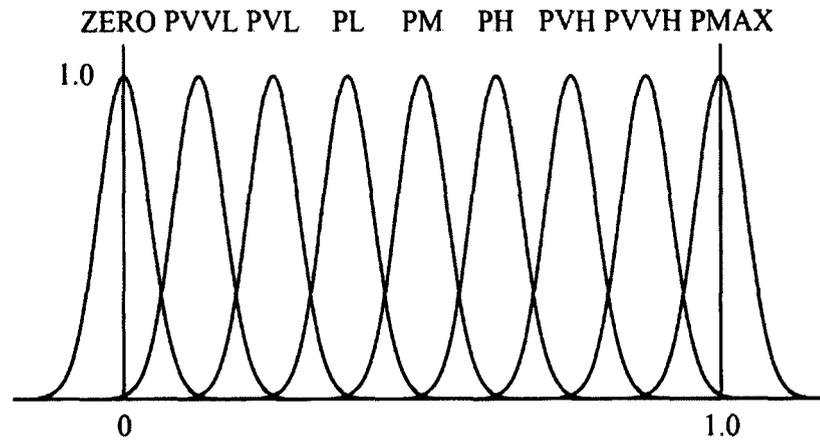


Figure 4.7: Membership functions for normalized output variables λ_1 and λ_2

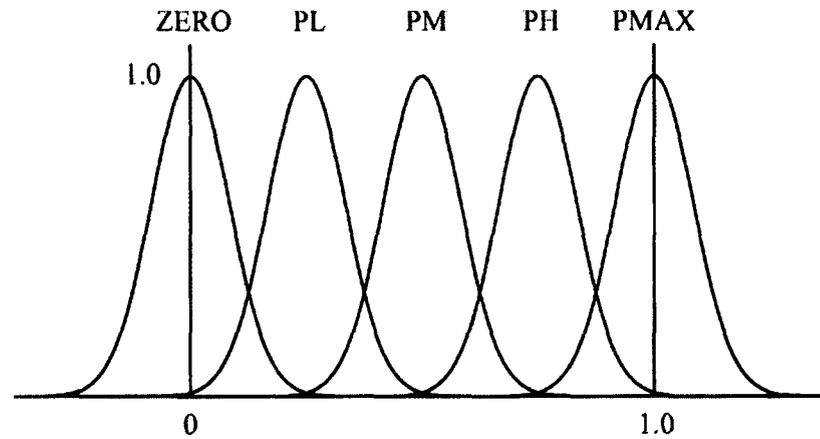


Figure 4.8: Membership functions for normalized output variables λ_3 and λ_4

IF \tilde{u}_1 is NH AND \tilde{u}_3 is PL THEN $\tilde{\lambda}_1$ is PH

IF \tilde{u}_2 is PVL AND \tilde{u}_4 is NL THEN $\tilde{\lambda}_2$ is PL

IF \tilde{u}_3 is PVH THEN $\tilde{\lambda}_3$ is PMAX

IF \tilde{u}_4 is NL THEN $\tilde{\lambda}_4$ is PM

The rules are designed intuitively; as the magnitude of a given tracking error along an axis varies positively or negatively, the associated control parameter λ_q varies

accordingly. The values of λ_q are maintained at ≥ 0 , ranging from ZERO for zero tracking errors to PMAX for the largest tracking errors, either positive or negative. As mentioned, the adaptation of λ_1 and λ_2 is also based on the magnitude of the velocity tracking errors, and varies accordingly. The resulting symmetric rule matrices are given in Tables 4.1 and 4.2.

		$\{\tilde{u}_1, \tilde{u}_2\}$								
		$\{\tilde{\lambda}_1, \tilde{\lambda}_2\}$	NVH	NH	NL	NVL	ZERO	PVL	PL	PH
$\{\tilde{u}_3, \tilde{u}_4\}$	NVH	PMAX	PMAX	PVH	PH	PM	PH	PVH	PMAX	PMAX
	NH	PMAX	PMAX	PH	PM	PL	PM	PH	PMAX	PMAX
	NL	PVH	PH	PM	PL	PVL	PL	PM	PH	PVH
	NVL	PH	PM	PL	PVL	PVVL	PVL	PL	PM	PH
	ZERO	PM	PL	PVL	PVVL	ZERO	PVVL	PVL	PL	PM
	PVL	PH	PM	PL	PVL	PVVL	PVL	PL	PM	PH
	PL	PVH	PH	PM	PL	PVL	PL	PM	PH	PVH
	PH	PMAX	PMAX	PH	PM	PL	PM	PH	PMAX	PMAX
	PVH	PMAX	PMAX	PVH	PH	PM	PH	PVH	PMAX	PMAX

Table 4.1: Fuzzy logic system rule table for $\tilde{\lambda}_1$ and $\tilde{\lambda}_2$

$\{\tilde{u}_3, \tilde{u}_4\}$	NVH	NH	NL	NVL	ZERO	PVL	PL	PH	PVH
$\{\tilde{\lambda}_3, \tilde{\lambda}_4\}$	PMAX	PH	PM	PL	ZERO	PL	PM	PH	PMAX

Table 4.2: Fuzzy logic system rule table for $\tilde{\lambda}_3$ and $\tilde{\lambda}_4$

Inference Mechanism

For rules involving two premise terms (i.e. rules with a consequent of either $\tilde{\lambda}_1$ or $\tilde{\lambda}_2$), the degree of certainty to which the i^{th} rule applies is quantified with the MIN operator, as follows

$$\mu_i(u_1, u_3) = \min \left\{ \mu_{A_1^j}(u_1), \mu_{A_3^k}(u_3) \right\} \quad i = 1, \dots, 81 \quad j = 1, \dots, 9 \quad k = 1, \dots, 9 \quad (4.43)$$

$$\mu_i(u_2, u_4) = \min \left\{ \mu_{A_2^j}(u_2), \mu_{A_4^k}(u_4) \right\} \quad i = 1, \dots, 81 \quad j = 1, \dots, 9 \quad k = 1, \dots, 9 \quad (4.44)$$

Otherwise, for rules with a consequent of either $\tilde{\lambda}_3$ or $\tilde{\lambda}_4$, the degree of certainty to which the i^{th} rule applies is simply obtained as

$$\mu_i(u_3) = \mu_{A_3^i}(u_3) \quad i = 1, \dots, 9 \quad (4.45)$$

$$\mu_i(u_4) = \mu_{A_4^i}(u_4) \quad i = 1, \dots, 9 \quad (4.46)$$

Next, the membership function $\mu_{B_q^i}(\lambda_q) \forall q = 1, \dots, 4$ of the implied fuzzy set B_q^i for the i^{th} rule is computed with the MIN operator, as follows

$$\mu_{B_1^i}(\lambda_1) = \min \left\{ \mu_i(u_1, u_3), \mu_{B_1^p}(\lambda_1) \right\} \quad i = 1, \dots, 81 \quad (4.47)$$

$$\mu_{B_2^i}(\lambda_2) = \min \left\{ \mu_i(u_2, u_4), \mu_{B_2^p}(\lambda_2) \right\} \quad i = 1, \dots, 81 \quad (4.48)$$

$$\mu_{B_3^i}(\lambda_3) = \min \{ \mu_i(u_3), \mu_{B_3^p}(\lambda_3) \} \quad i = 1, \dots, 9 \quad (4.49)$$

$$\mu_{B_4^i}(\lambda_4) = \min \{ \mu_i(u_4), \mu_{B_4^p}(\lambda_4) \} \quad i = 1, \dots, 9 \quad (4.50)$$

where the implied fuzzy set B_q^i specifies the certainty level at which the q^{th} control parameter should be a specific crisp value, taking only rule i into consideration. The resulting membership function $\mu_{B_q^i}(\lambda_q)$ is a function of λ_q , and the MIN operator truncates the top of the $\mu_{B_q^p}(\lambda_q)$ membership function associated with the output fuzzy set B_q^p of the i^{th} rule. Since the MIN operator works on each point of a given output membership curve, the final shape of the implied membership function could change, depending on the number of points used to generate the membership functions. In this work, each output membership function was modeled with 100 points, evenly distributed across the output universe of discourse.

As an alternative to using implied fuzzy sets to perform the inference step, the overall implied fuzzy set with membership function $\mu_{B_q}(\lambda_q)$, which represents the conclusion reached by all rules simultaneously, can be computed; as it was in Chapter 2 and in [15, 99]. However, as stated by Passino and Yurkovich [112], the use of the overall implied fuzzy set is not recommended for real-time applications, due to the increased computation complexity of the overall implied fuzzy set itself, and of the defuzzification step, which is usually based on the center of area (COA) method.

Defuzzification

The center of gravity (COG) defuzzification method calculates a crisp control parameter $\lambda_q \forall q = 1, \dots, 4$ based on the area and center-of-area of all implied fuzzy sets. With this method, the crisp value for the control parameter λ_q is given by

$$\lambda_q = \frac{\sum_{i=1}^n b_i^q \int \mu_{B_q^i}(\lambda_q) d\lambda_q}{\sum_{i=1}^n \int \mu_{B_q^i}(\lambda_q) d\lambda_q} \quad q = 1, \dots, 4 \quad (4.51)$$

where $n \in \mathbb{R}$ denotes the number of rules (81 for λ_1 and λ_2 , and 9 for λ_3 and λ_4), and

$b_i^q \in \mathbb{R}$ denotes the center of area of the membership function of B_q^p associated with the implied fuzzy set B_q^i for the i^{th} rule.

Finally, output scaling gains $h_q \in \mathbb{R} \forall q = 1, \dots, 4$ multiply their corresponding control parameter λ_q , thereby modifying the base widths and providing greater tracking accuracy as the scaling gains increase. Increasing output scaling gains spreads out the output membership functions, and makes the meaning of their associated linguistics quantify larger numbers [112].

Nonlinear Control Surfaces

The resulting nonlinear adaptation surface and nonlinear adaptation curve that concisely represent all the information in the fuzzy adaptation mechanism are shown in Figs. 4.9 and 4.10 respectively. Note that decreasing the input scaling gains would rescale the axes and decrease the slope of the adaptation surface and adaptation curve, thus reducing the controller's overall sensitivity to small tracking errors. Conversely, setting the input scaling gains too high would cause the membership functions to saturate at low values, resulting in oscillations or chattering. The ripple effects observed in Figs. 4.9 and 4.10 are created by the interpolation between the different rules. Ultimately, tuning the scaling gains, selecting the membership functions, and defining the rules modifies the shape of the adaptation surface and adaptation curve, which could affect the closed-loop behavior.

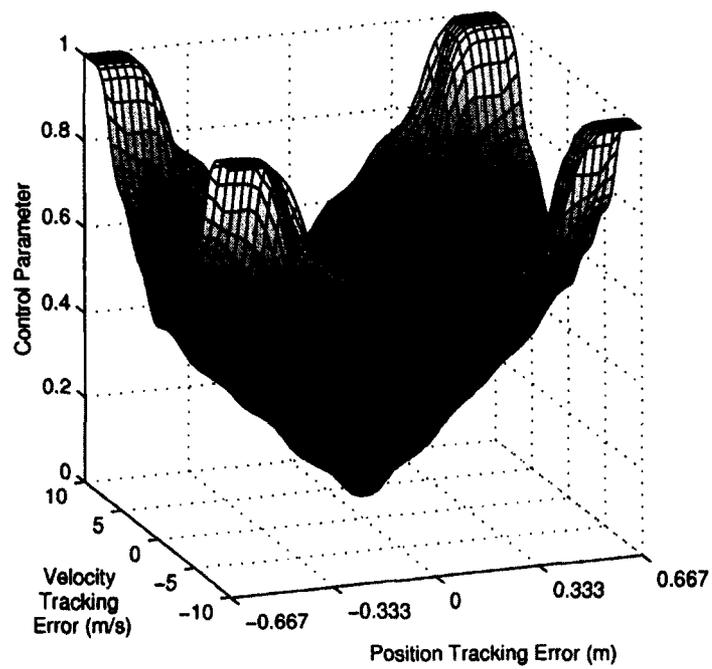
4.4.3 DSAC-Based Composite Control

Adopting the DSAC control law developed in Section 3.5 as the slow control term to stabilize the quasi-steady-state subsystem yields the following direct adaptive composite control law

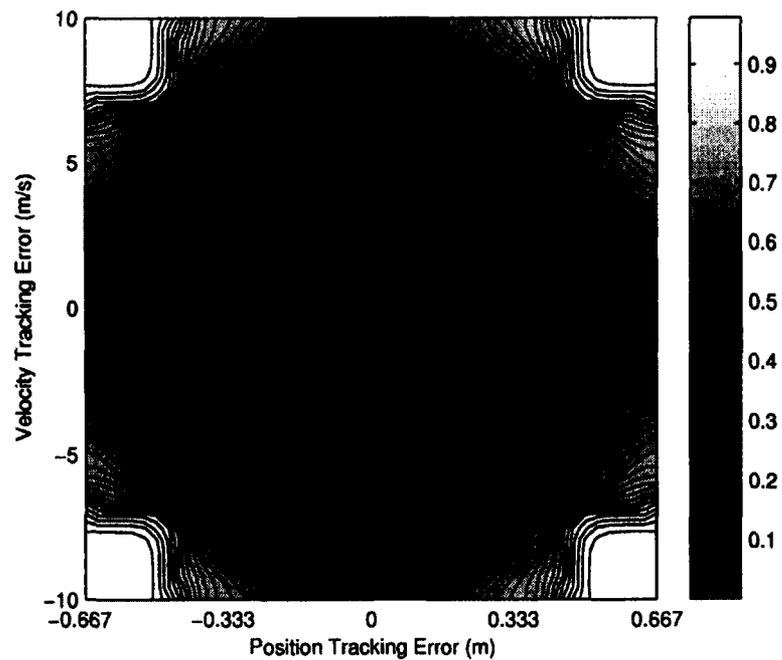
$$\tau = J^T(q) [K_e(t)e_y + K_x(t)x_m + K_u(t)u_m] + K_v(\dot{q} - \dot{q}_m) \quad (4.52)$$

where the time-varying control gains can be defined as

$$K_e(t) \triangleq \begin{bmatrix} K_{e_p}(t) & K_{e_d}(t) \end{bmatrix} = K_{P_e}(t) + K_{I_e}(t) \quad (4.53)$$



(a)



(b)

Figure 4.9: Adaptation of λ_1 and λ_2 (a) Three-dimensional nonlinear adaptation surface (b) Contour map of the adaptation surface

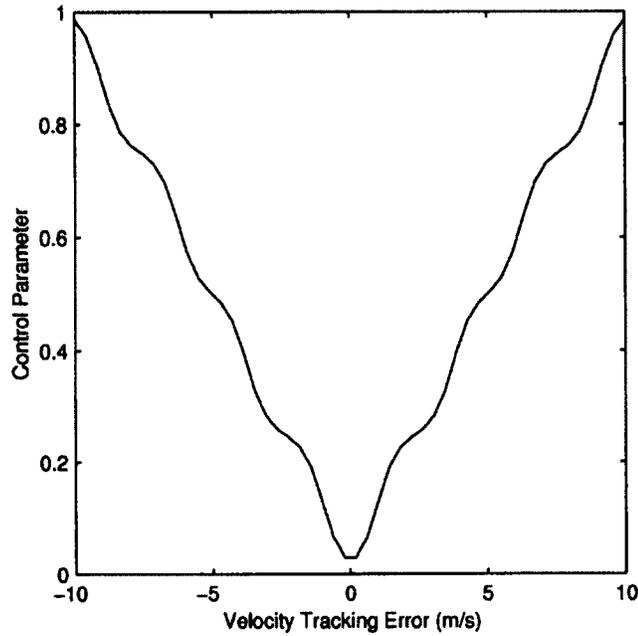


Figure 4.10: Nonlinear adaptation curve for λ_3 and λ_3

$$K_x(t) \triangleq \begin{bmatrix} K_{x_p}(t) & K_{x_d}(t) \end{bmatrix} = K_{P_x}(t) + K_{I_x}(t) \quad (4.54)$$

$$K_u(t) \triangleq \begin{bmatrix} K_{u_p}(t) & K_{u_d}(t) \end{bmatrix} = K_{P_u}(t) + K_{I_u}(t) \quad (4.55)$$

The adaptation mechanism of the time-varying control gains $K_{P_e}(t)$, $K_{I_e}(t)$, $K_{P_x}(t)$, $K_{I_x}(t)$, $K_{P_u}(t)$ and $K_{I_u}(t)$ is given by (3.93)-(3.98). The corresponding composite controller block diagram is shown in Fig. 4.11.

4.4.4 DMSAC-Based Composite Control

In this section, adaptive control is achieved by using the DMSAC control law (3.63) as the slow control term; in other words, by considering only the output feedback control gain $K_e(t)$ in (4.52) and disregarding the feedforward control gains $K_x(t)$ and $K_u(t)$, as follows

$$\tau = J^T(q)K_e(t)e_y + K_v(\dot{q} - \dot{q}_m) \quad (4.56)$$

where, for convenience, the time-varying control gain $K_e(t)$ can be defined as in (4.53), with $K_{Pe}(t), K_{Ie}(t)$ being updated as (3.93) and (3.94). The resulting block diagram is shown in Fig. 4.12.

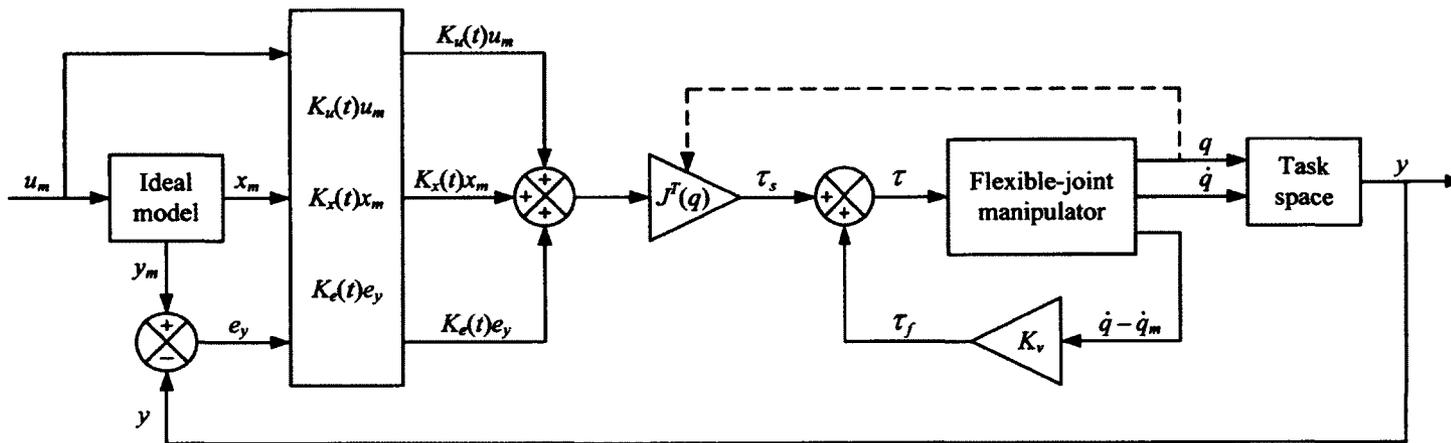


Figure 4.11: Block diagram representation of the DSAC composite controller.

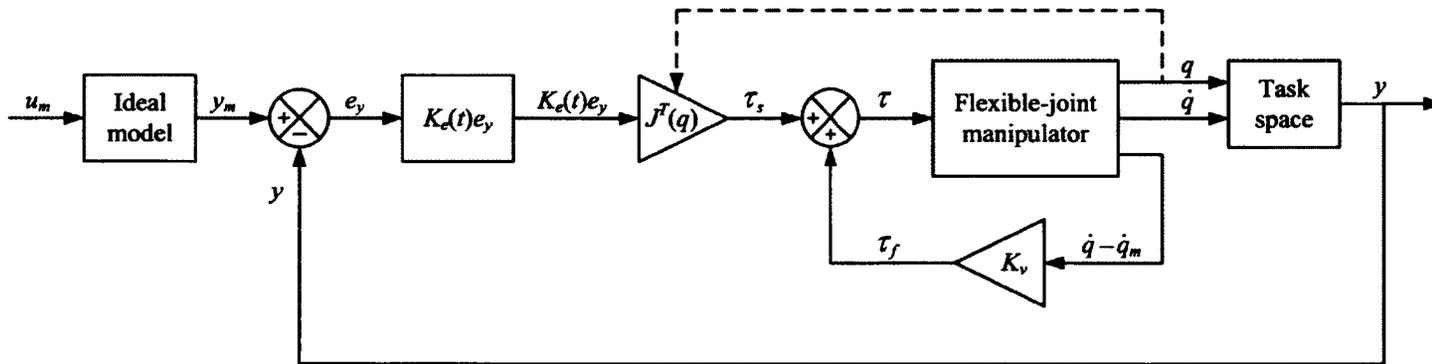


Figure 4.12: Block diagram representation of the DMSAC composite controller.

Parameter	Value	Unit
m_1	1.5075	kg
m_2	1.5075	kg
l_1	4.50	m
l_2	4.50	m
l_{c_1}	2.25	m
l_{c_2}	2.25	m
J_m	I_2	kg·m ²
k	$500I_2$	N·m/rad

Table 4.3: Nonnominal linear joint stiffness manipulator characteristics

4.5 Simulation Results

This section presents the numerical simulations used for validating and comparing the different composite controllers developed in this chapter. To simplify comparison with the other control strategies, the DSAC and DMSAC composite control schemes will only be applied to the nonsquare flexible-joint manipulator. The results of these two control laws for a square manipulator are presented and discussed in Appendix C. To validate the nominal trajectory tracking performance, the control strategies were applied to the linear joint stiffness robot manipulator given by (4.3) and (4.4). Although Readman [159] suggests joint stiffness values on the order of 10^4 N·m/rad for a harmonic drive joint, and 10^3 N·m/rad for a flexible coupling joint, the joint parameters are based on Cao and de Silva [151], and are representative of manipulators with obvious flexible effects in their joints. The parameters of the two-link flexible-joint robot are summarized in Table 4.3.

The integral structure of the integral adaptive control gains in (3.57)-(3.59) is computed online using a standard Tustin algorithm. In addition, all integral control gains were initialized to zero. The reference model parameters selected were $\omega_n = 10$ rad/s and $\zeta = 0.9$. The control gains of the nonadaptive transpose Jacobian composite control law (4.35) were selected as

$$K_p = 7I_2 \quad K_d = 10I_2 \quad K_v = 35I_2$$

The fuzzy adaptive composite controller (4.37) was designed with the following output scaling parameters and control gain matrix

$$h_1 = h_2 = 100 \quad h_3 = h_4 = 270 \quad K_v = 35I_2$$

The parameters, gains, and coefficients of the DSAC composite control algorithm given by (4.52) were determined as follows

$$\begin{aligned} \Gamma_{P_e} &= \text{diag} [80 \quad 80 \quad 50 \quad 50] \\ \Gamma_{I_e} &= \text{diag} [120 \quad 120 \quad 150 \quad 150] \\ \Gamma_{P_x} = \Gamma_{P_u} &= \text{diag} [0.1 \quad 0.1 \quad 2 \quad 2] \\ \Gamma_{I_x} = \Gamma_{I_u} &= \text{diag} [0.2 \quad 0.2 \quad 2 \quad 2] \\ \sigma_e &= \text{diag} [0.75 \quad 0.75 \quad 0.15 \quad 0.15] \\ \sigma_x = \sigma_u &= \text{diag} [0.9 \quad 0.9 \quad 0.4 \quad 0.4] \\ K_v &= 120I_2 \end{aligned}$$

Finally, for the DMSAC composite controller (4.56), the control parameters, gains and coefficients were selected as follows

$$\begin{aligned} \Gamma_{P_e} = \Gamma_{I_e} &= \text{diag} [150 \quad 150 \quad 25 \quad 25] \\ \sigma_e &= \text{diag} [0.008 \quad 0.008 \quad 0.023 \quad 0.023] \\ K_v &= 35I_2 \end{aligned}$$

In all cases, the control gains, parameters and coefficients were selected to provide satisfactory tracking performance along each side of the 12.6 m \times 12.6 m square trajectory, with acceptable transient response at the corners when applied to the two-link flexible-joint robot modeled with the nominal linear joint stiffness dynamics

representation described by (4.3) and (4.4), and with physical characteristics listed in Table 4.3. Note that the values of the selected σ -coefficients are relatively small, since they are only to prevent the integral adaptive gains from reaching excessively high values, or diverging in time. Figures 4.13 to 4.25 show the trajectory-tracking results for the nominal linear joint stiffness manipulator system described above, with all four composite control strategies .

As shown in Fig. 4.13, when the nonadaptive transpose Jacobian composite control strategy is used, pronounced positioning overshoots of 0.553 m occur at each direction change, due to the large trajectory tracking errors illustrated in Fig. 4.14. Improved results were obtained with the fuzzy adaptive composite scheme, as shown in Fig. 4.15, where the trajectory exhibits minimal overshoots of 0.089 m at each direction change, with rapid settling to a steady-state along each side of the trajectory. This improvement in performance is also demonstrated in Fig. 4.16 by lower magnitude of trajectory tracking errors, particularly with the position-related tracking errors (i.e. e_{y_1} and e_{y_2}). The corresponding time-varying control gains $K_p(t)$ and $K_d(t)$, shown in Fig. 4.17, are relatively low in magnitude and almost constant between two direction changes, and they increase sharply when the end-effector reaches each corner of the square trajectory, thus adapting the control law to reduce tracking errors and positioning overshoots. It is also important to note that as observed in this figure, the adaptation rates (i.e. the rates of change of the adaptive gains) at each direction switch are large. These high adaptation rates ensure that the required gains are provided at the correct time, with peaks occurring only at the corners of the square trajectory. Fast adaptation rates allow the controller to use low gains, and to increase them only when necessary.

Figure 4.17 also shows that the adaptive gains reached higher peak magnitudes than the constant gain values selected for the nonadaptive transpose Jacobian control strategy. As a result, the control torque magnitudes and the rates associated with the fuzzy adaptive control strategy are higher than those related to the nonadaptive transpose Jacobian control scheme. Therefore, when tuning the fuzzy adaptive controller for use in practical applications with sudden increases in tracking errors, the peak magnitudes in adaptive gains must not result in unrealistic control torques

that could not be implemented by the joint mechanism hardware selected. The same consideration must be taken into account when tuning the nonadaptive controller as well. To allow better tracking performance over the complete trajectory, the constant gains of the nonadaptive controller, K_p and K_d would have to be increased to sufficiently high values. Switching direction at high constant gains, however, would be more demanding in terms of required torque amplitudes and torque rates, and could also result in torque profiles that could not be handled by typical joint mechanisms, such as harmonic drives or planetary gear drives. The maximum control torque peak magnitudes (not shown in the figures) obtained are 52.9 N·m and 30.2 N·m for joints 1 and 2 for the nonadaptive transpose Jacobian controller, compared with 142.5 N·m and 81.4 N·m for joints 1 and 2 for the fuzzy adaptive controller. These peak control efforts occur initially as a single impulse to increase the Cartesian velocity of the end-effector along the y-axis from rest to 0.84 m/s. Thereafter, the control torque magnitude reaches a maximum level at each direction change of 18.0 N·m and 10.5 N·m for joints 1 and 2 for the nonadaptive transpose Jacobian strategy, compared with 46.2 N·m and 30.5 N·m for both joints for the fuzzy adaptive strategy. This demonstrates that the improved tracking accuracy provided by the fuzzy adaptive strategy is obtained at the cost of greater control torque efforts. Nevertheless, both nonadaptive and fuzzy adaptive control torque profiles could be implemented by a typical lightweight flexible-joint mechanism (HFUC-2A, Harmonic Drive AG).

Again from Fig. 4.17, it is clear that both components of the proportional and derivative time-varying control gain matrices are bounded by lower limits, given by 8.05 and 5.67 respectively. Indeed, after each subsequent trajectory direction switch, the tracking errors tend toward zero, reaching the minimum points on the nonlinear adaptation surface and adaptation curve, which correspond to 0.0805 for λ_1 and λ_2 , and 0.0210 for λ_3 and λ_4 . The multiplication of these minimum points by their associated output scaling gains, h_i , corresponds to the lower bounds of the time-varying control gains observed in Fig. 4.17.

Significant improvements over the nonadaptive transpose Jacobian controller and the fuzzy adaptive controller were achieved with the DSAC composite control strategy. This is illustrated in Fig. 4.18, where positioning overshoots of 0.031 m, 0.030 m and

0.030 m for the first, second, and third direction changes respectively are obtained, respectively, and in Fig. 4.19 where trajectory tracking errors of lower magnitudes are obtained. Moreover, tracking with the DSAC composite controller is closer to a straight line along each side of the trajectory. Hence, the response obtained with the DSAC composite control approach is much closer to a square than that of non-adaptive and fuzzy adaptive responses. Figures 4.20 to 4.22 demonstrate that the control gains perform correctly and similarly to those obtained with the fuzzy adaptive control scheme; that is, they increase and decrease in accordance with the specific tracking situation, with sudden increases at each direction switch so that an improved transient response is achieved, compared to the nonadaptive transpose Jacobian control methodology. Among the control gain adaptation histories provided in Figs. 4.20 to 4.22, Fig. 4.20, which illustrates the adaptation of the stabilizing control gains $K_{e_p}(t)$ and $K_{e_d}(t)$ as a function of time, provides a good depiction of the convergence of the trajectory tracking errors, since both gains are shown to asymptotically converge to zero between each direction change. Indeed, recall that $K_{e_p}(t)$ and $K_{e_d}(t)$ are adapted only as function of the trajectory tracking error $e_y(t)$. Similar behavior can be observed in Figs. 4.21 and 4.22 for $K_{x_p}(t)$ and $K_{x_d}(t)$, and $K_{u_p}(t)$ and $K_{u_d}(t)$ respectively, where a convergence to zero is shown. Note that the behavior of the time-varying control gains observed in these last two figures is almost identical, since the signals x_m and u_m are similar and the DMSAC composite controller was designed with $\Gamma_{P_x} = \Gamma_{P_u}$, $\Gamma_{I_x} = \Gamma_{I_u}$, and $\sigma_x = \sigma_u$.

Finally, the trajectory for the DMSAC-based composite controller shown in Fig. 4.23 has positioning overshoots of 0.107 m, 0.104 m and 0.082 m for the first, second and third direction changes respectively. In addition, as with the other direct adaptive composite methods developed in this chapter, Fig. 4.25 demonstrates that the control gains behave correctly (i.e. according to the tracking errors with sudden increases at each corner). The successive increase in tracking performance at each corner of the trajectory, shown in Fig. 4.24, is explained by analyzing the adaptation history of the stabilizing adaptive control gains provided in Fig. 4.25. As discussed earlier, the integral term of both $K_{e_p}(t)$ and $K_{e_d}(t)$ is obtained from a first-order filtering of the tracking error. Therefore, once the tracking errors start to decrease

as a function of time the integral terms also decrease, due to the contribution of their respective previous values. The decrease in the integral terms occurs at a rate proportional to the σ -term. Thus, the greater these forgetting coefficients are, the faster the adaptive gains will decrease between two direction changes. With the DMSAC composite controller, the two coefficients selected are relatively small; hence, the rate of decrease of the adaptive gains is also small, and it does not fully compensate for the sudden increase of the adaptive gains at each direction change. This results in slightly higher values of $K_{e_p}(t)$ and $K_{e_d}(t)$ at each direction change, and successive improvement of the tracking performance at each corner and along each side of the trajectory. These results highlight the importance of the using the σ -term in the DSAC and DMSAC-based algorithm in a way that avoids instability due to excessively high values of the adaptive control gains, particularly for the present application, in which sudden increases in tracking errors occur periodically. Although the DMSAC composite strategy provides better tracking results compared to both the nonadaptive transpose Jacobian and fuzzy adaptive control methodologies, the results are not as good as the those achieved with the DSAC composite algorithm. As anticipated, these results clearly suggest that injecting information about the reference model into the controller structure through the use of the adaptive feedforward control gains $K_x(t)$ and $K_u(t)$ improves the tracking performance.

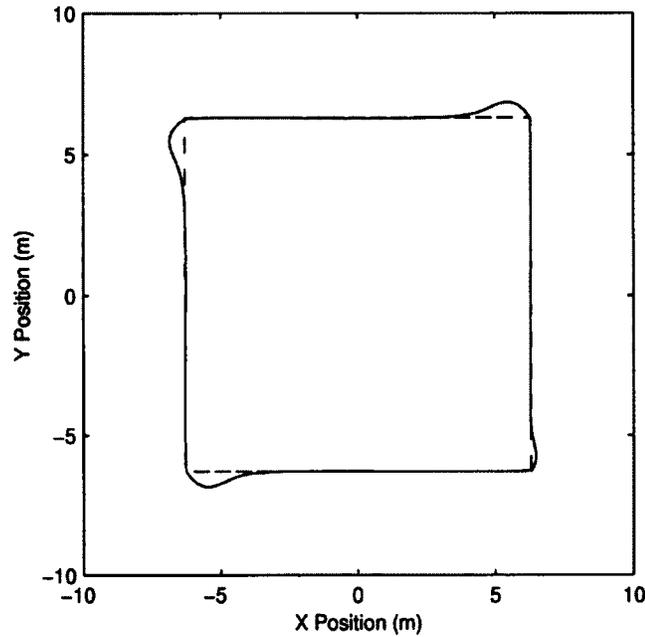


Figure 4.13: Trajectory tracking results obtained with the nonadaptive transpose Jacobian composite controller applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m). The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$.

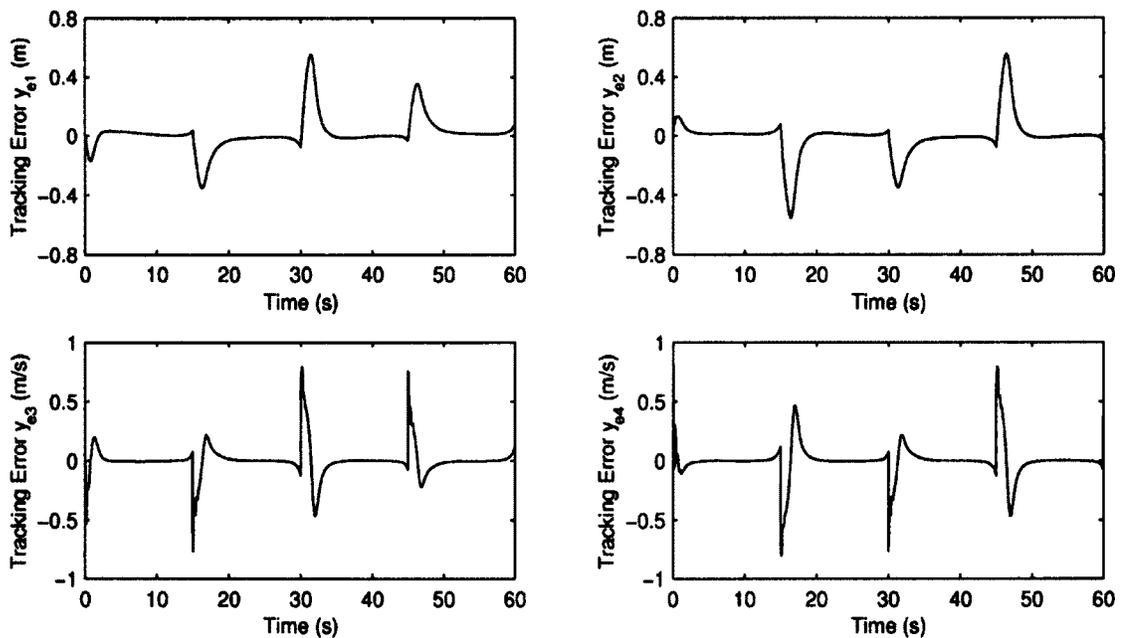


Figure 4.14: Trajectory tracking errors ($y_e(t) = y_d(t) - y(t)$) obtained with the nonadaptive transpose Jacobian composite controller applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m).

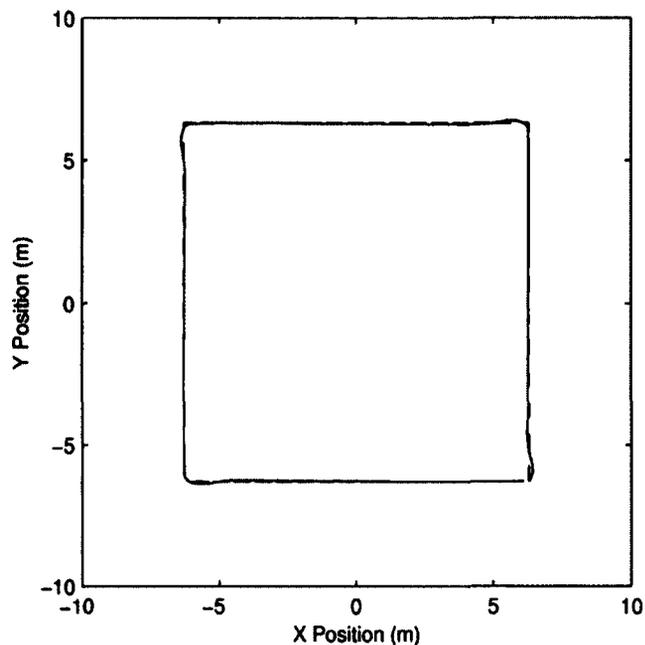


Figure 4.15: Trajectory tracking results obtained with the fuzzy adaptive composite controller applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m). The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$.

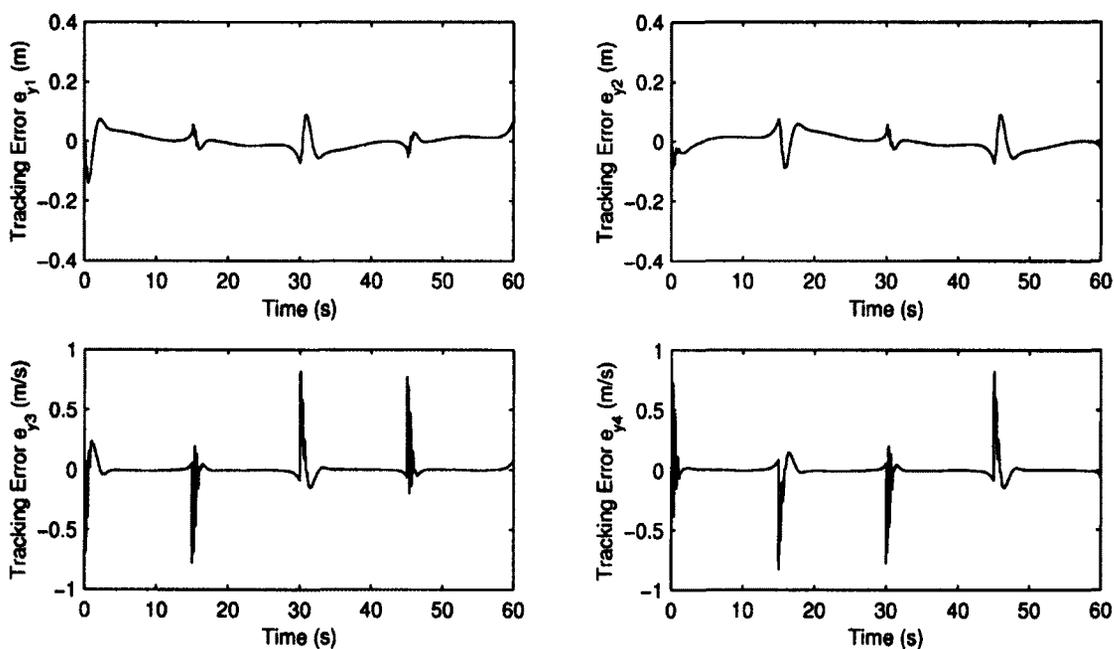


Figure 4.16: Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the fuzzy adaptive composite controller applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m).

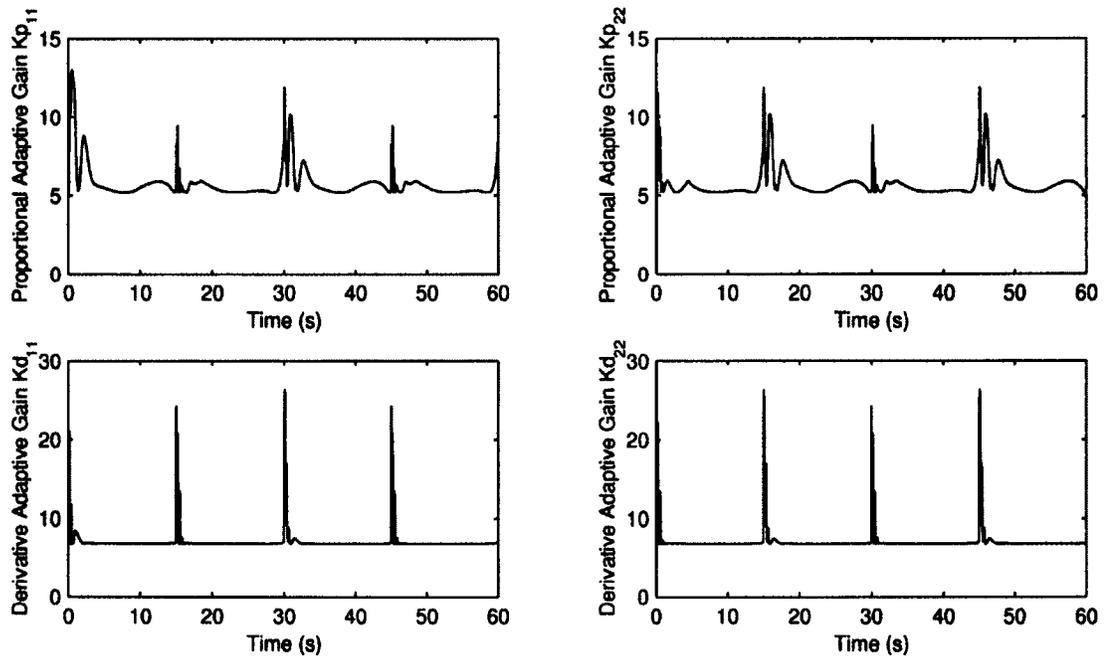


Figure 4.17: Adaptation history of the fuzzy controller gains $K_p(t)$ and $K_d(t)$ when applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m).

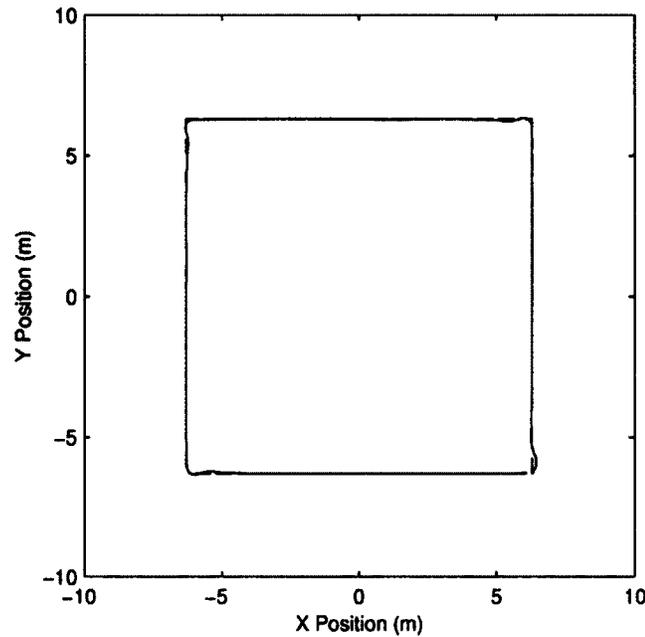


Figure 4.18: Trajectory tracking results obtained with the DSAC composite controller applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m). The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$.

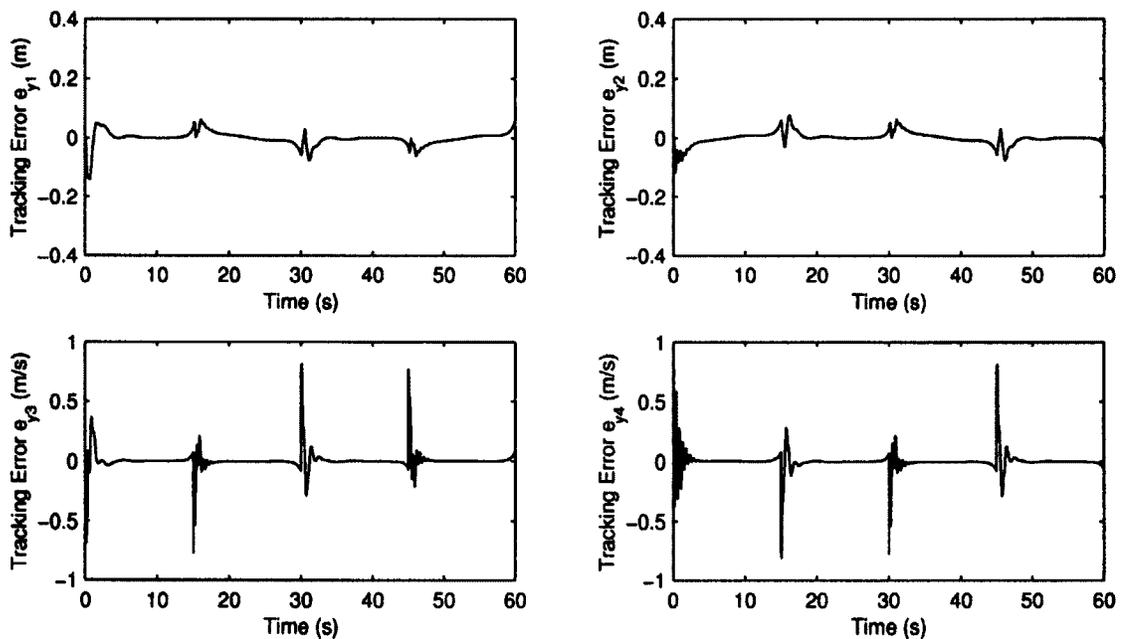


Figure 4.19: Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the DSAC composite controller applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m).

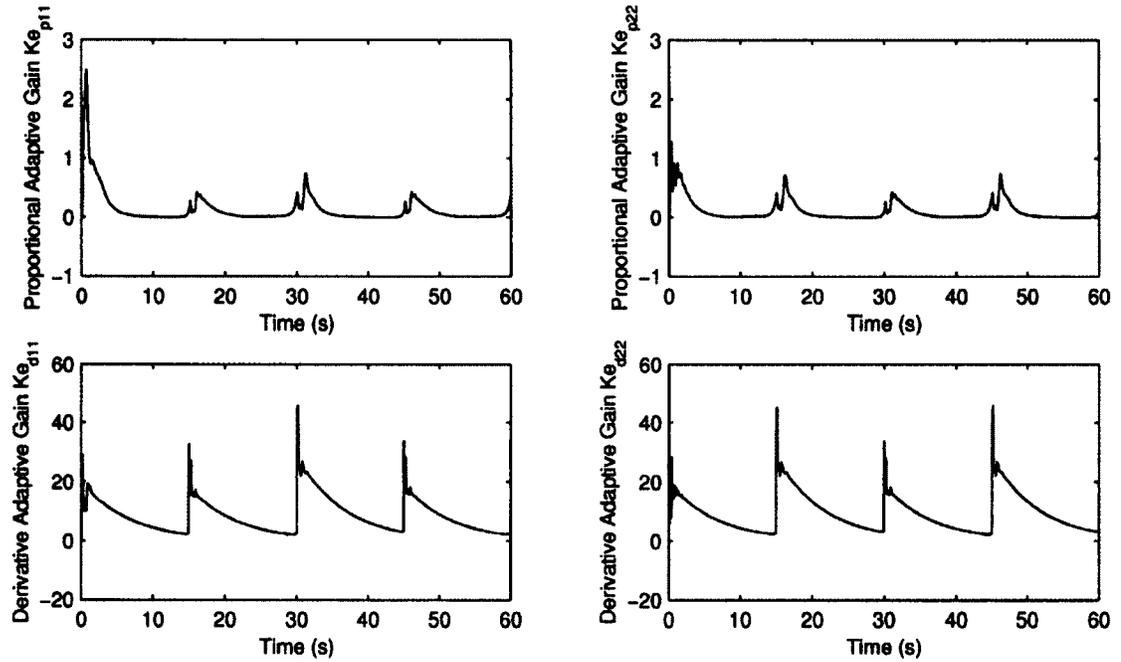


Figure 4.20: Adaptation history of the DSAC composite controller gains $K_{ep}(t)$ and $K_{ed}(t)$ when applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m).

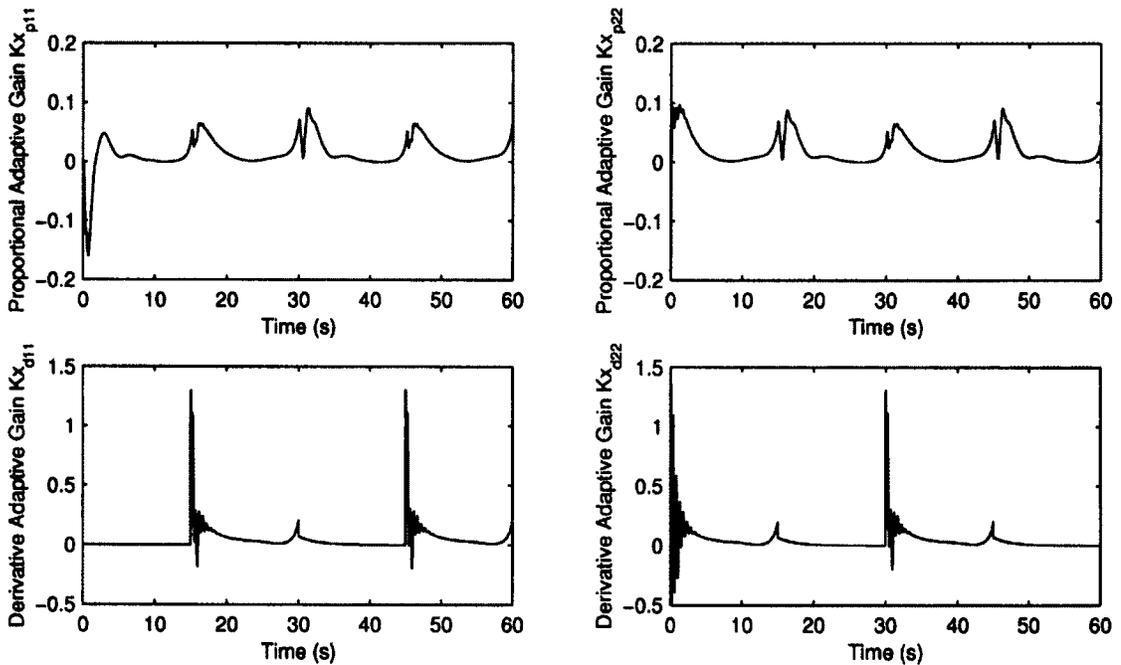


Figure 4.21: Adaptation history of the DSAC composite controller gains $K_{xp}(t)$ and $K_{xd}(t)$ when applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m).

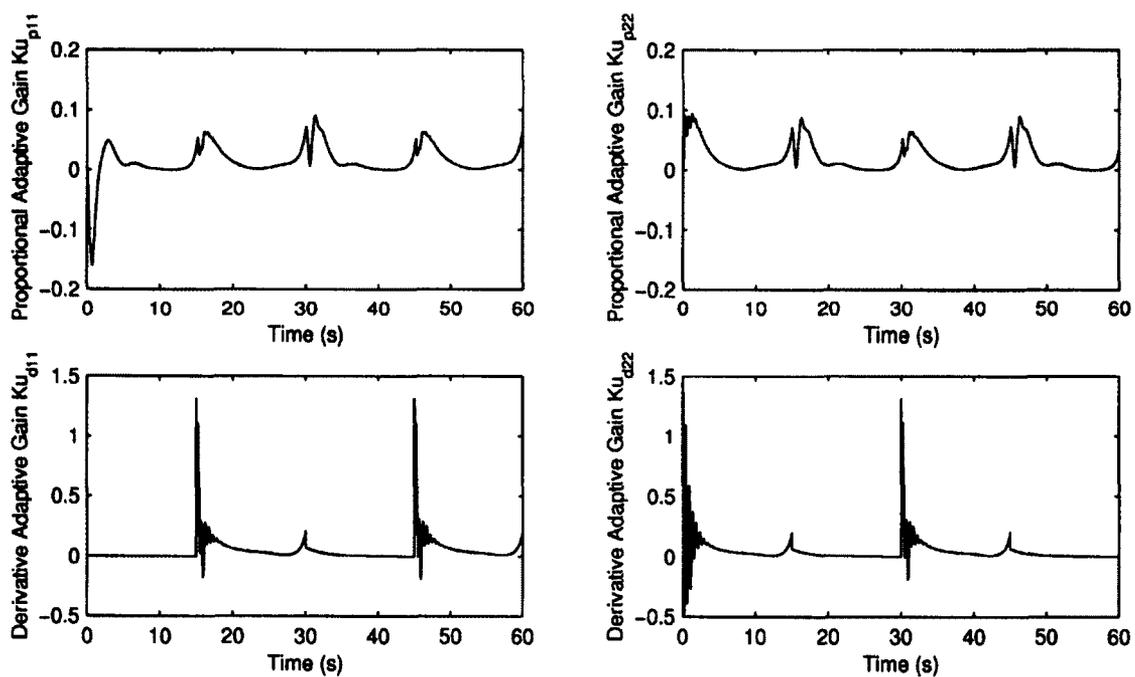


Figure 4.22: Adaptation history of the DSAC composite controller gains $K_{u_p}(t)$ and $K_{u_d}(t)$ when applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m).

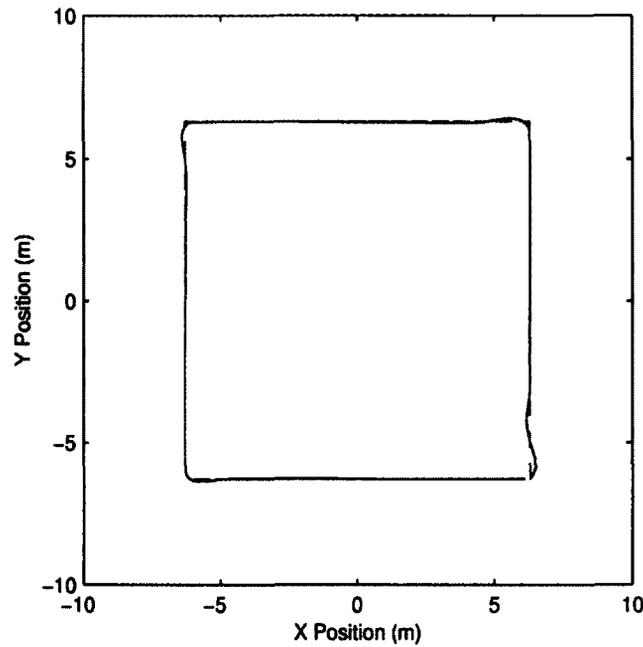


Figure 4.23: Trajectory tracking results obtained with the DMSAC composite controller applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m). The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$.

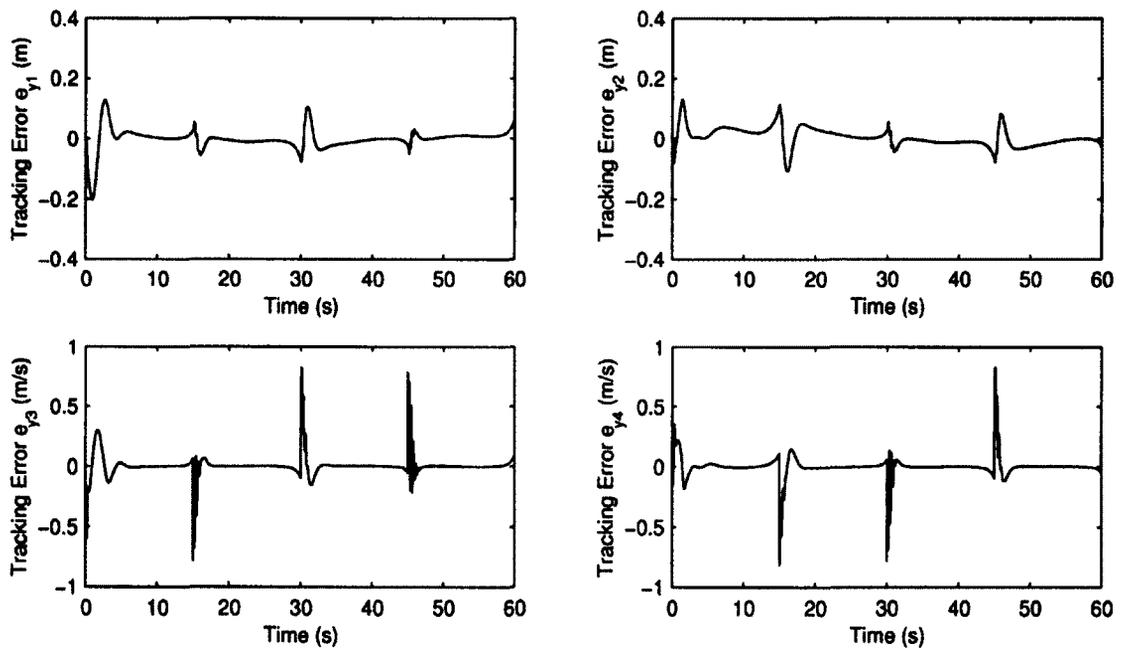


Figure 4.24: Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the DMSAC composite controller applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m).

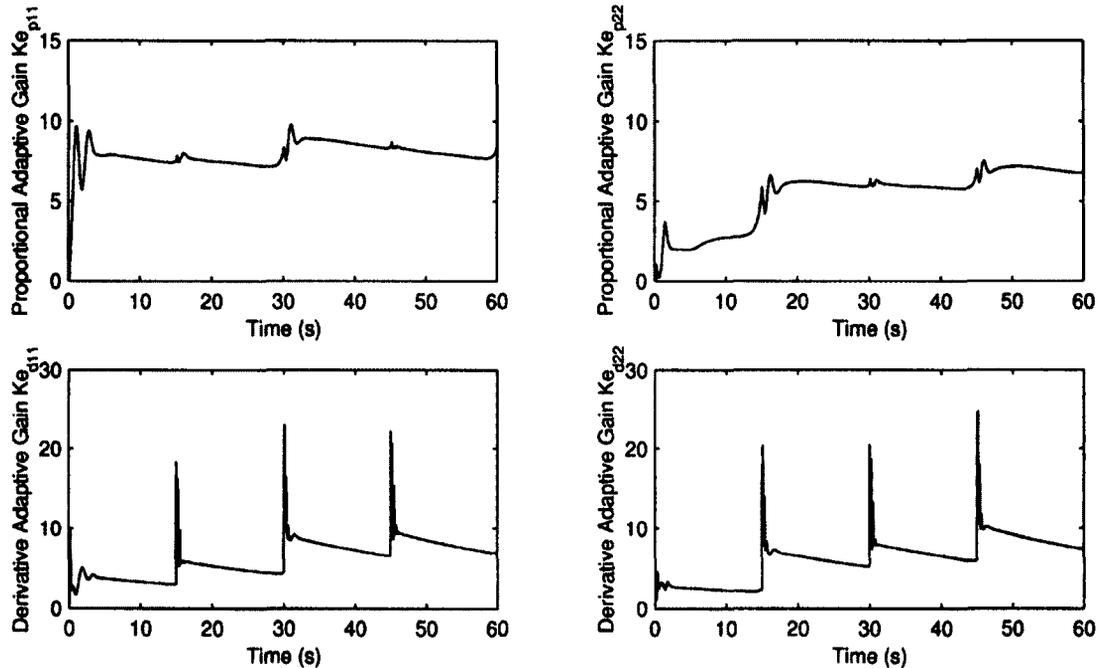


Figure 4.25: Adaptation history of the DMSAC composite controller gains $K_{ep}(t)$ and $K_{ed}(t)$ when applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m).

4.5.1 Robustness to Parametric Uncertainties

Numerical simulations using a manipulator with joints of significantly lower stiffness were performed to assess the robustness of the controllers to parametric uncertainties in the system. The same composite controllers previously tuned with the nominal robot manipulator were applied to a robot with a joint stiffness matrix of $k = 200I_2$ N·m/rad, representing an uncertainty of 60%. The obtained results are presented in Figs. 4.26 to 4.38. As shown in Fig. 4.26, the tracking performance of the nonadaptive transpose Jacobian control strategy is further aggravated, and a positioning overshoot of 0.637 m occurs at each direction change. Trajectory tracking errors are also higher than those obtained for the nominal linear joint stiffness robot system, as illustrated in Fig. 4.27.

The end-effector trajectory for the fuzzy adaptive controller shown in Fig. 4.28 exhibits slightly greater positioning overshoots than those of the nominal case: 0.108 m at each direction change. In response to the higher flexibilities in the manipulator system, oscillations in the time-varying control gains, illustrated in Fig. 4.30, are

greater than in the nominal case. This provides efficient damping of the elastic vibrations while keeping the tracking errors as low as possible, as shown in Fig. 4.29. These findings demonstrate that compared with the nonadaptive transpose Jacobian control scheme, the fuzzy adaptive algorithm is not significantly sensitive to uncertainties in the system parameters.

When validated under parametric uncertainties, the DSAC composite control strategy achieved smaller positioning overshoots than all other: 0.080 m, 0.083 m and 0.083 m for the first, second, and third direction change, as well as smaller tracking errors as shown demonstrated in Figs. 4.31 and 4.32. While the DSAC-based composite control methodology achieved superior tracking performance in terms of peak positioning overshoot magnitudes, the settling times between two direction changes are greater, due to a decreased damping of the oscillations in the trajectory tracking error signals, as illustrated in Figs. 4.32 to 4.35. Despite the increased oscillations, the control gains behave correctly and stabilize the tracking errors, as indicated by the convergence to zero of the stabilizing adaptive control gain $K_e(t)$ in Fig. 4.33. The oscillating behavior can be explained by the larger control inputs associated with the DSAC strategy, due to the introduction of additional feedforward control terms in the control structure. Indeed, these terms increase the overall control effort required to improve the tracking performance. However, this makes the controller more sensitive to sudden changes in the desired trajectory, which in turn results in greater oscillations and settling times when applied to a manipulator with excessively low joint stiffness coefficients, as demonstrated by the results obtained.

Finally, the trajectory for the DMSAC composite controller shown in Fig. 4.36 exhibits overshoots similar to those of the nominal case: 0.095 m, 0.112 m and 0.094 m for the first, second and third direction changes respectively. In addition, as shown in Fig. 4.37, the trajectory tracking errors are minimized, despite the increased flexibility in the joints. Similar to the other direct adaptive composite control methodologies proposed in this chapter, greater adaptation rates occur in the time-varying control gains, illustrated in Fig. 4.38, in order to provide the correct gain values at the precise time required to reduce the tracking errors as much as possible. These results illustrate the robustness of the DMSAC controller, and demonstrate that once the

DMSAC algorithm is correctly designed under nominal conditions, the results are not overly sensitive to uncertainties in the system parameters.

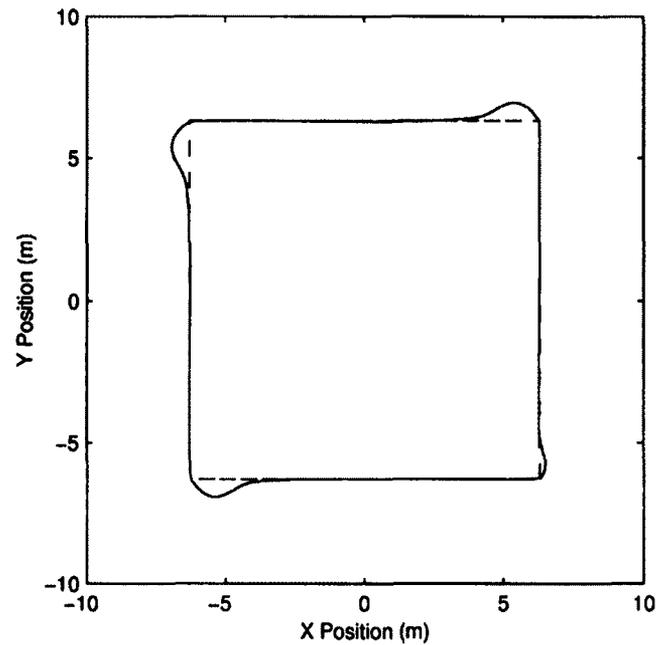


Figure 4.26: Trajectory tracking results obtained with the nonadaptive transpose Jacobian composite controller applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m). The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$.

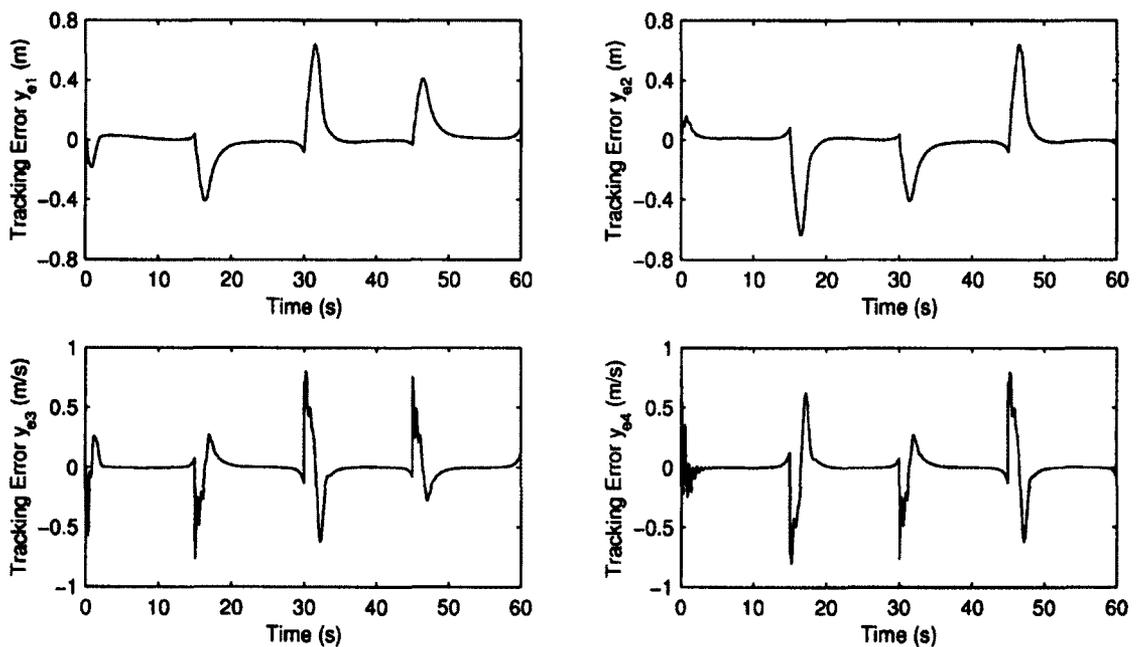


Figure 4.27: Trajectory tracking errors ($y_e(t) = y_d(t) - y(t)$) obtained with the nonadaptive transpose Jacobian composite controller applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m).

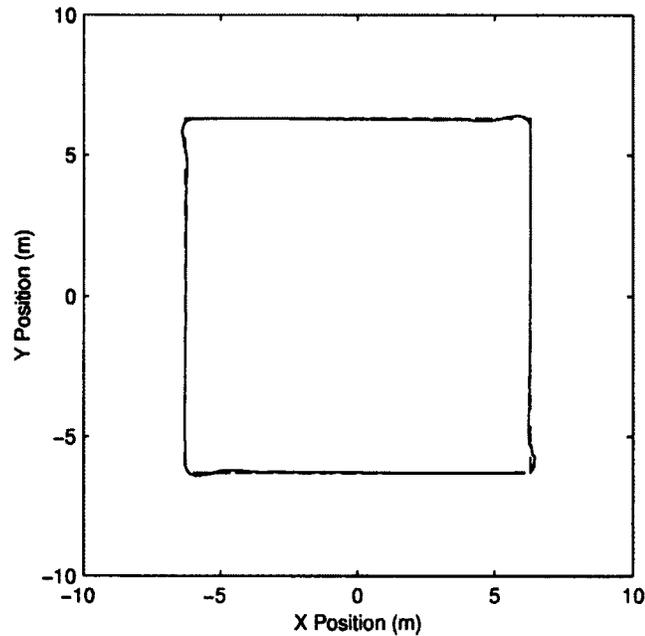


Figure 4.28: Trajectory tracking results obtained with the fuzzy adaptive composite controller applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m). The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$.

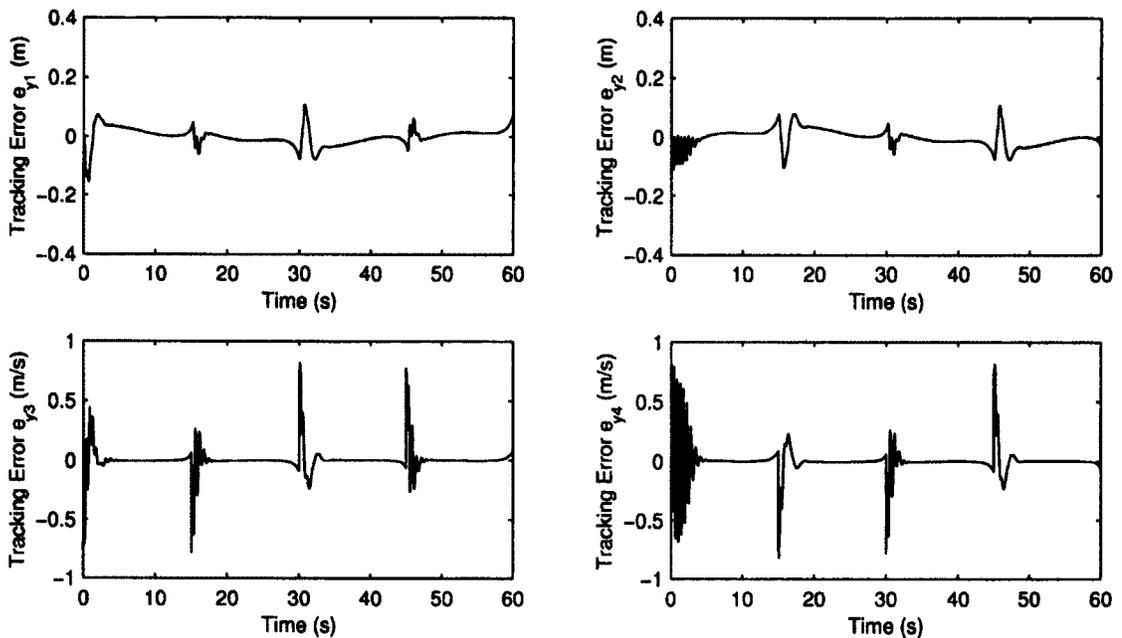


Figure 4.29: Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the fuzzy adaptive composite controller applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m).

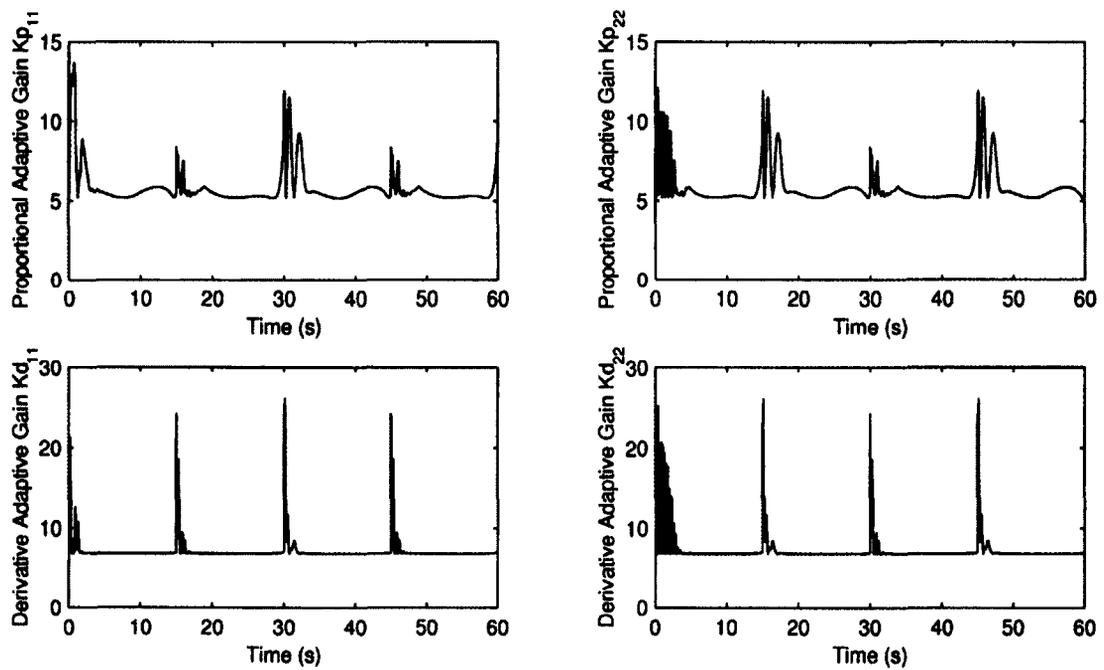


Figure 4.30: Adaptation history of the fuzzy controller gains $K_p(t)$ and $K_d(t)$ when applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m).

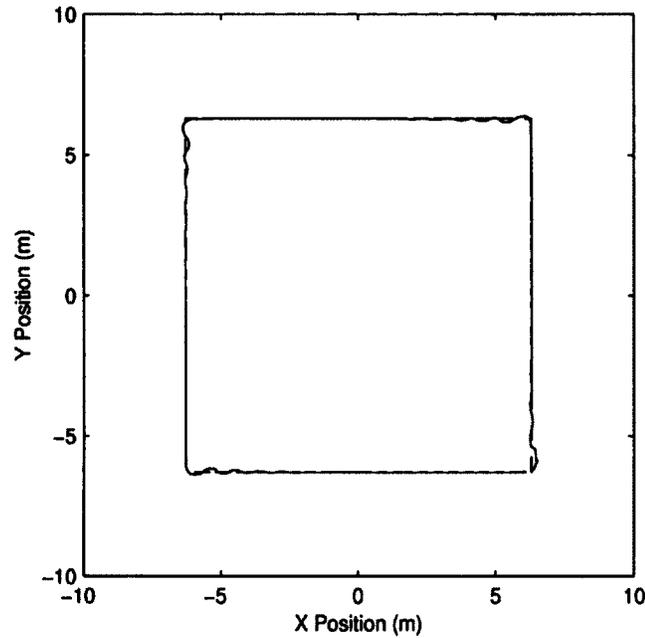


Figure 4.31: Trajectory tracking results obtained with the DSAC composite controller applied to the uncertain linear joint stiffness manipulator ($k = 200I_2 \text{ N}\cdot\text{m}$). The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$.

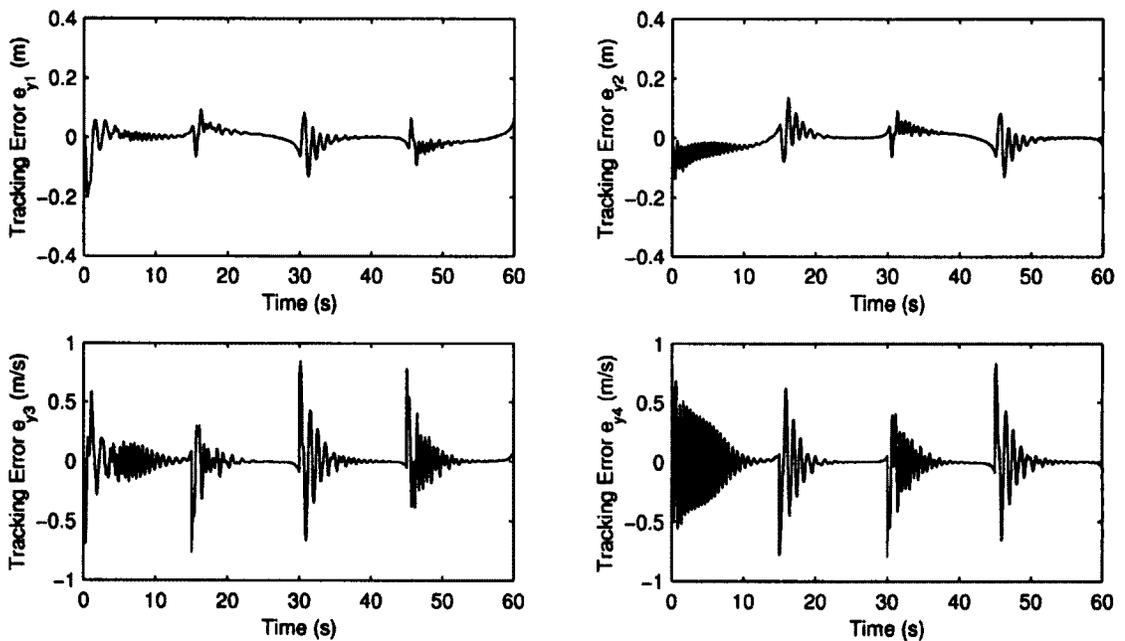


Figure 4.32: Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the DSAC composite controller applied to the uncertain linear joint stiffness manipulator ($k = 200I_2 \text{ N}\cdot\text{m}$).

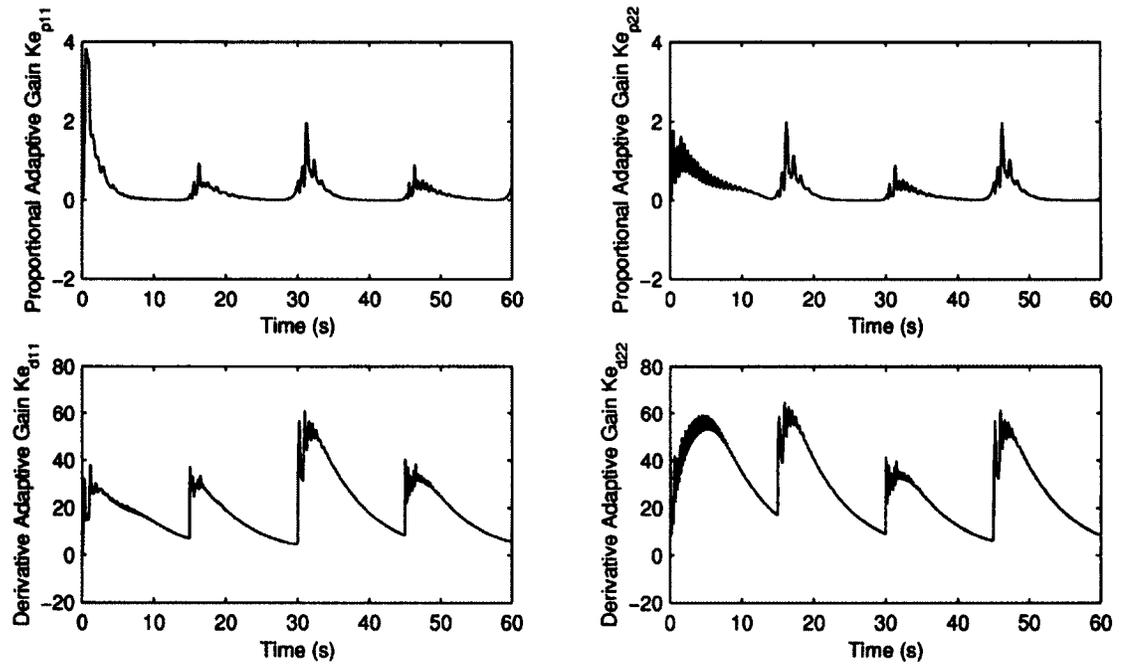


Figure 4.33: Adaptation history of the DSAC composite controller gains $K_{e_p}(t)$ and $K_{e_d}(t)$ when applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m).

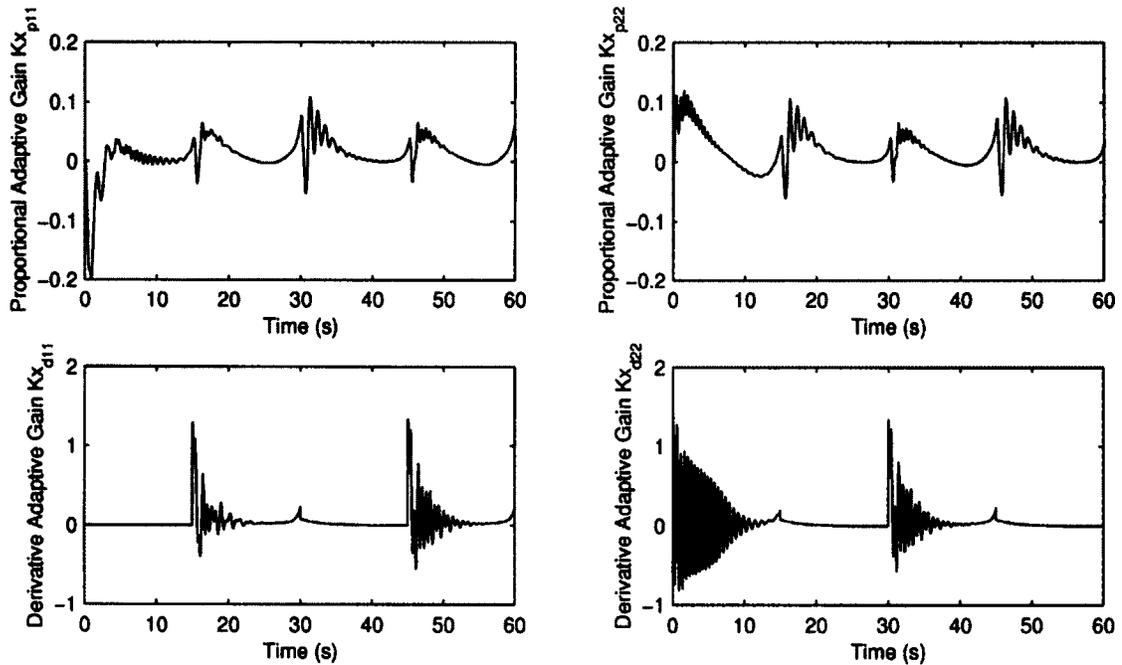


Figure 4.34: Adaptation history of the DSAC composite controller gains $K_{x_p}(t)$ and $K_{x_d}(t)$ when applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m).

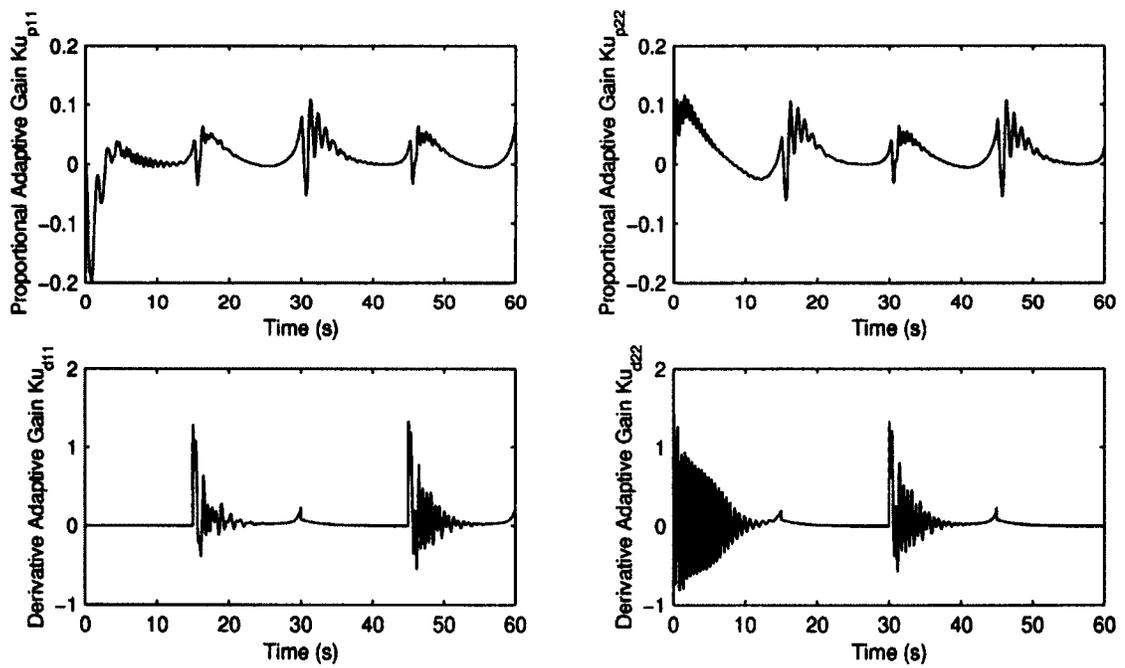


Figure 4.35: Adaptation history of the DSAC composite controller gains $K_{u_p}(t)$ and $K_{u_d}(t)$ when applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m).

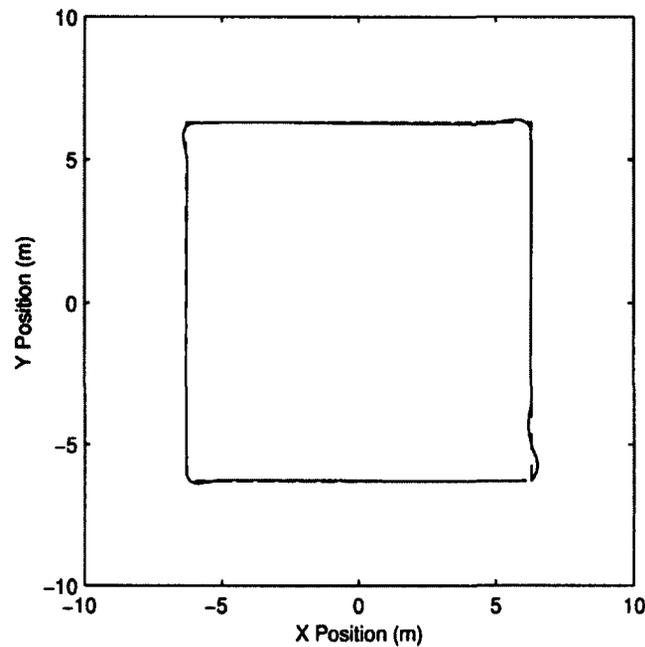


Figure 4.36: Trajectory tracking results obtained with the DMSAC composite controller applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m). The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$.

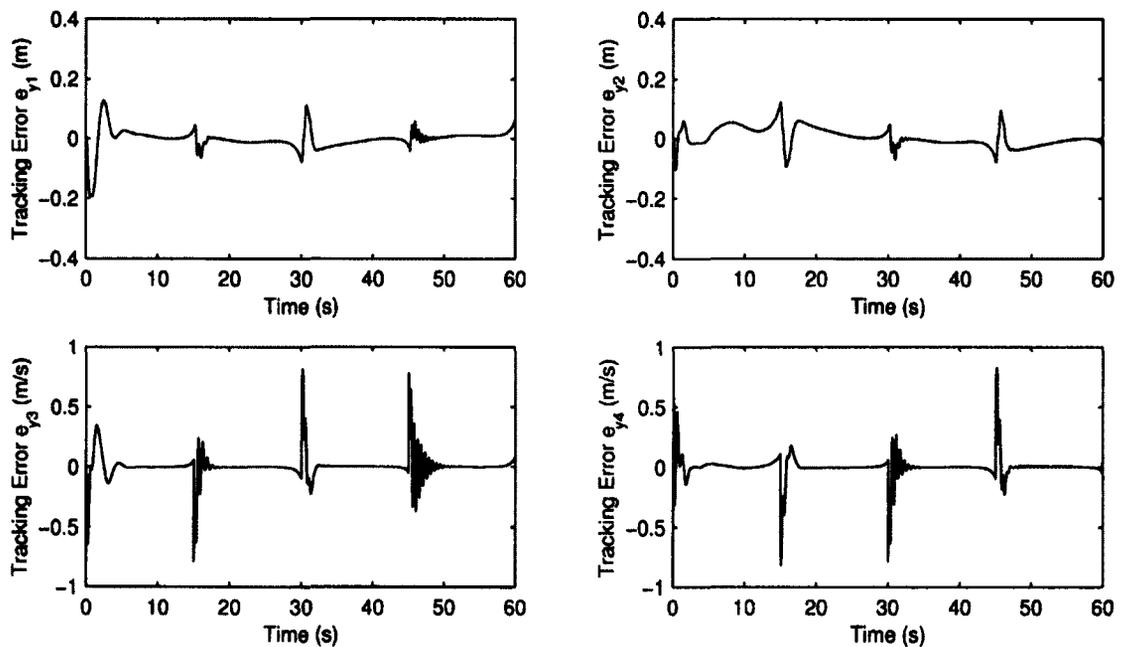


Figure 4.37: Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the DMSAC composite controller applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m).

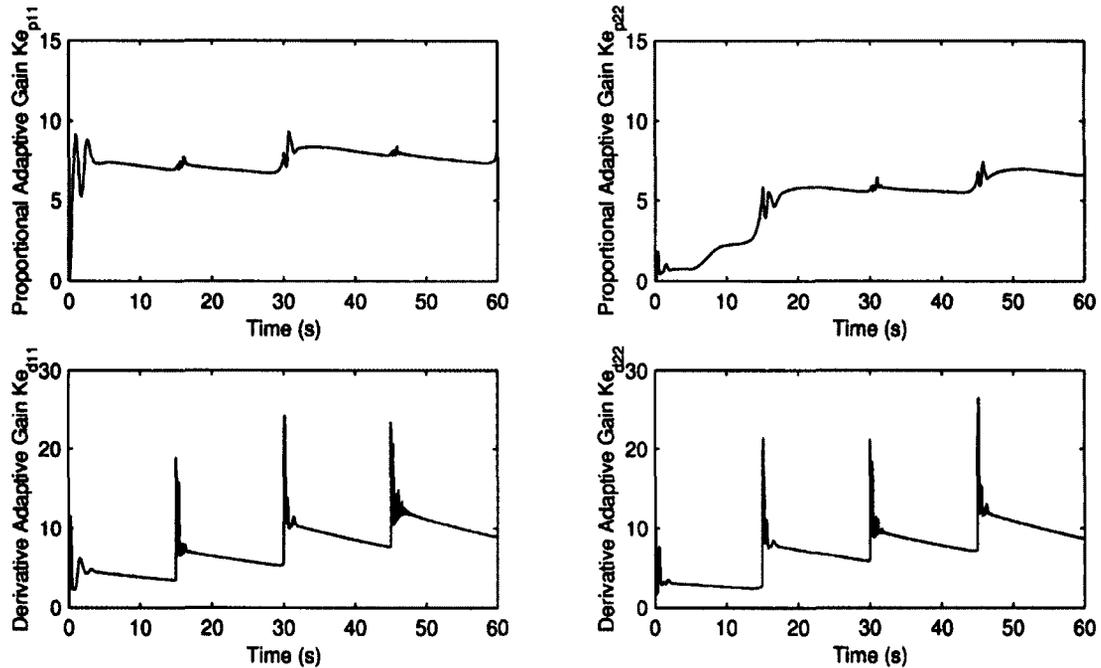


Figure 4.38: Adaptation history of the DMSAC composite controller gains $K_{e_p}(t)$ and $K_{e_d}(t)$ when applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m).

4.5.2 Robustness to Modeling Errors

As mentioned previously, for most practical applications the linear joint stiffness model is not satisfactory, as several unknown or unpredictable nonlinear effects can affect the actual end-effector motion. For this reason, it is important to determine the controller's capability to handle unmodeled dynamics. To achieve this, the composite controllers designed for the nominal robot manipulator were applied without retuning the control parameters and scaling gains to the more realistic nonlinear joint stiffness model described in Section 4.2.2. The parameters for the nonlinear joint stiffness dynamics model are presented in Table 4.4. With these parameters, the nonlinear joint stiffness and friction curves are shown in Figs. 4.39(a) and 4.39(b) respectively, where the joint torsion is defined as $q_m - q$.

Similar to the previous cases, all direct adaptive control composite methodologies achieved smaller overshoots at each direction change, compared with the nonadaptive transpose Jacobian controller. This is demonstrated in Fig. 4.40 where a positioning overshoot of 0.207 m occurs at each corner of the trajectory for the nonadaptive

Parameter	Value	Unit
m_1	1.5075	kg
m_2	1.5075	kg
l_1	4.50	m
l_2	4.50	m
l_{c_1}	2.25	m
l_{c_2}	2.25	m
J_m	I_2	kg·m ²
a_1	$500I_2$	N·m/rad
a_2	$500I_2$	N·m/rad
k_{sw}	$10I_2$	–
a_{sw}	3000	–
γ_1	0.5	–
γ_2	150	–
γ_3	50	–
γ_4	2	–
γ_5	100	–
γ_6	0.5	–
S	$\begin{bmatrix} 0 & J_{m_2} \\ 0 & 0 \end{bmatrix}$	kg·m ²

Table 4.4: Nonlinear joint stiffness manipulator characteristics

transpose Jacobian strategy, compared to 0.134 m for the fuzzy adaptive controller (see Fig. 4.42). As illustrated in Figs. 4.45 and 4.50, further improvements were obtained with both the DSAC and DMSAC control methodologies, with overshoots of 0.077 m at each directing change for the DSAC controller, and overshoots of 0.111 m, 0.092 m and 0.081 m for the first, second, and third direction changes respectively for the DMSAC controller. These diminished positioning overshoots are due to smaller trajectory tracking errors, as shown in Figs. 4.41, 4.43, 4.46, and 4.51, for the non-adaptive transpose Jacobian, fuzzy adaptive, DSAC, and DMSAC control schemes, respectively.

It is also notable that the response obtained with the nonadaptive control strategy failed to converge to a straight line along each side of the square trajectory, as depicted in Fig. 4.40. This is because the nonlinear effects introduced in the dynamics model would have required larger gains to stabilize the end-effector correctly along the square trajectory. Indeed, as shown in Figs. 4.44, 4.47 to 4.49, and 4.52, all adaptive gains

reached greater magnitudes than previously attained, as required to minimize the tracking errors. This suggests that the proposed direct adaptive control schemes cope with modeling errors as expected: by adjusting their control gains to maintain similar tracking performance to that achieved with the nominal linear joint stiffness model.

As discussed earlier, all controllers were tuned in numerical simulations with the nominal linear joint stiffness model. Hence, these results clearly indicate that the direct adaptive control methodologies are more robust with respect to modeling errors, and demonstrate the ability of the adaptive controllers to provide acceptable and constant performance without modification of the control parameters, coefficients or gains.

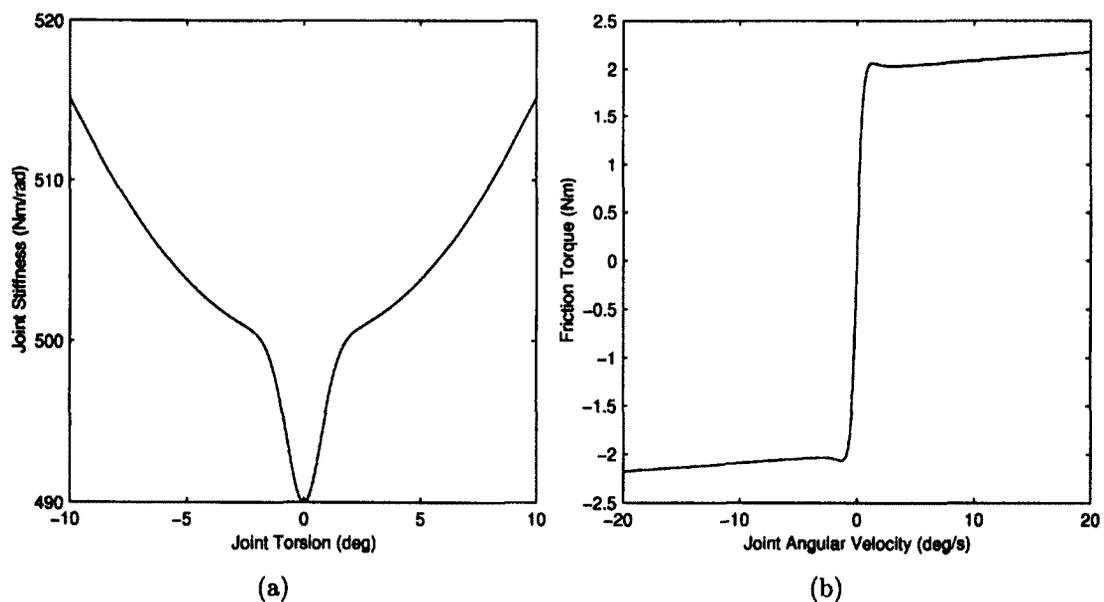


Figure 4.39: Joint stiffness and friction characteristics for the nonlinear joint stiffness dynamics model (a) Nonlinear joint stiffness curve with soft windup effect in the region around 0 deg (b) Continuous and symmetric friction curve that includes static, Stribeck, viscous, and Coulombic effects

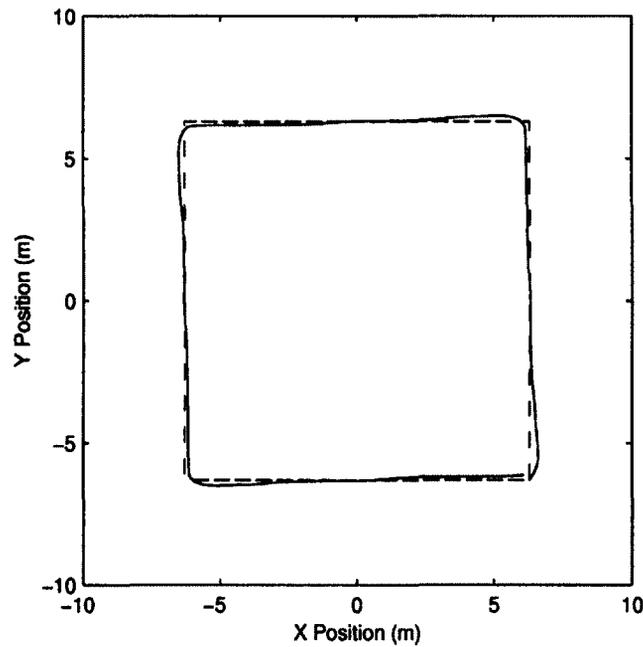


Figure 4.40: Trajectory tracking results obtained with the nonadaptive transpose Jacobian composite controller applied to the nonlinear joint stiffness manipulator. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$.

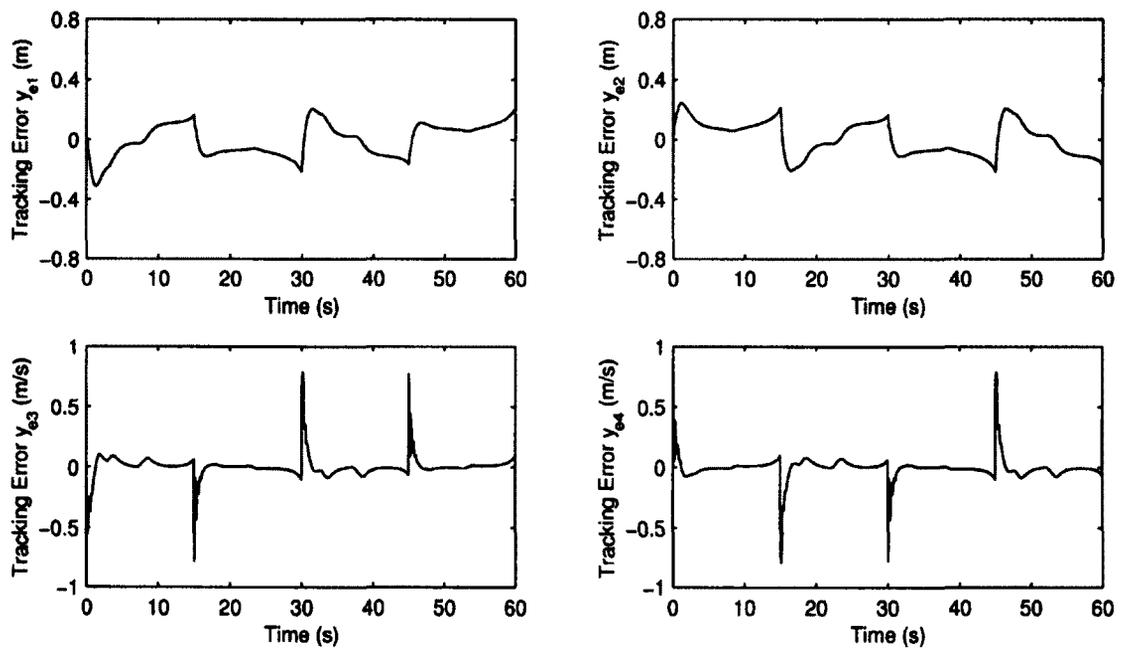


Figure 4.41: Trajectory tracking errors ($y_e(t) = y_d(t) - y(t)$) obtained with the nonadaptive transpose Jacobian composite controller applied to the nonlinear joint stiffness manipulator.

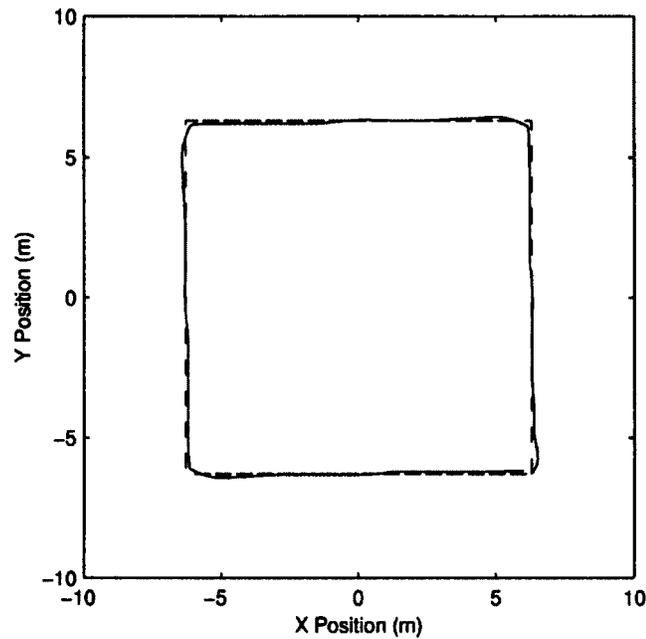


Figure 4.42: Trajectory tracking results obtained with the fuzzy adaptive composite controller applied to the nonlinear joint stiffness manipulator. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$.

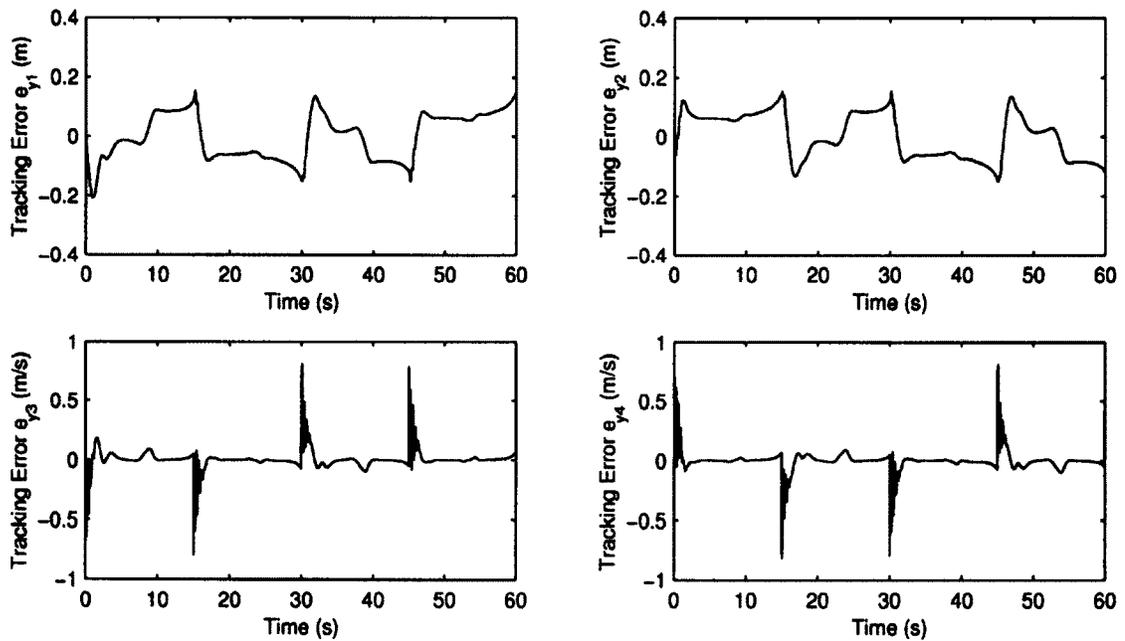


Figure 4.43: Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the fuzzy adaptive composite controller applied to the nonlinear joint stiffness manipulator.

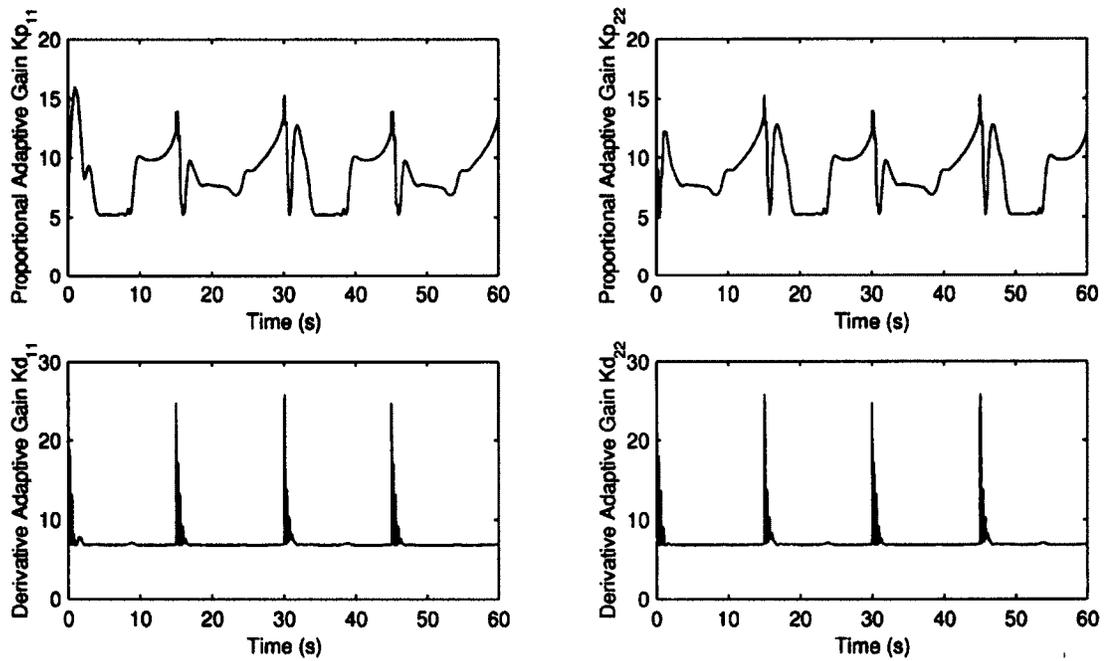


Figure 4.44: Adaptation history of the fuzzy controller gains $K_p(t)$ and $K_d(t)$ when applied to the nonlinear joint stiffness manipulator.

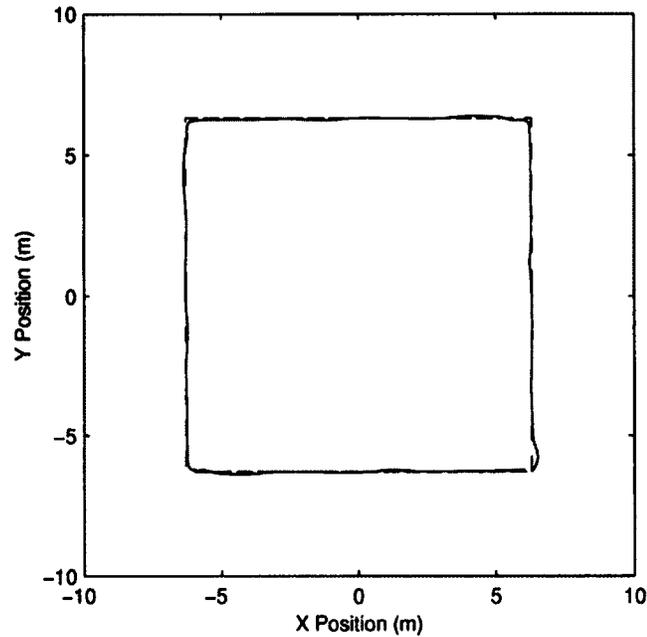


Figure 4.45: Trajectory tracking results obtained with the DSAC composite controller applied to the nonlinear joint stiffness manipulator. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$.

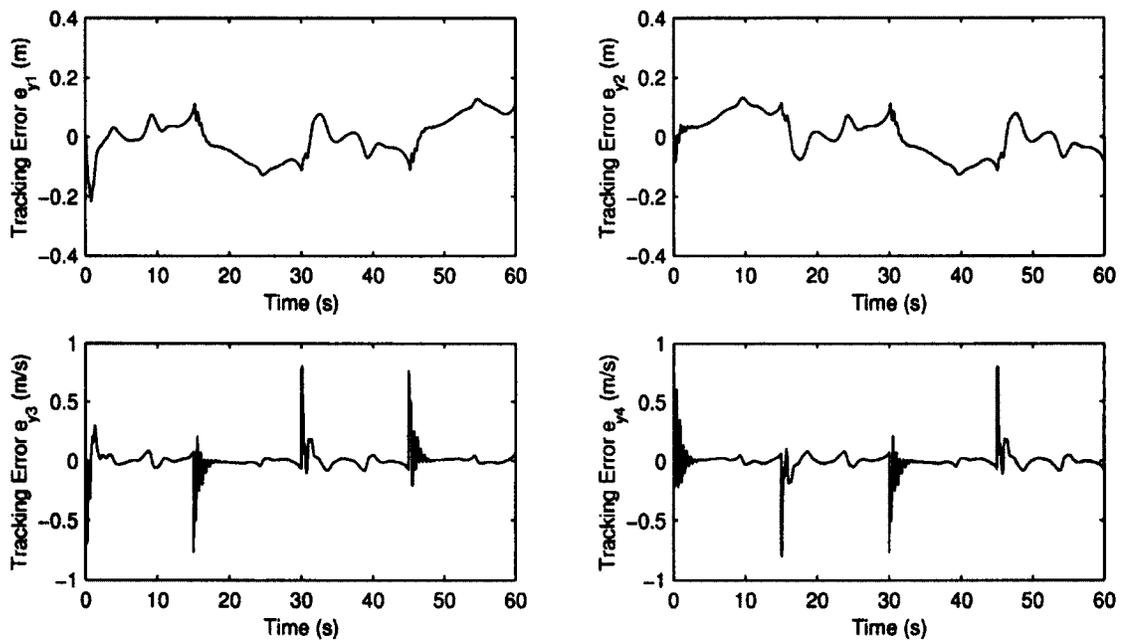


Figure 4.46: Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the DSAC composite controller applied to the nonlinear joint stiffness manipulator.

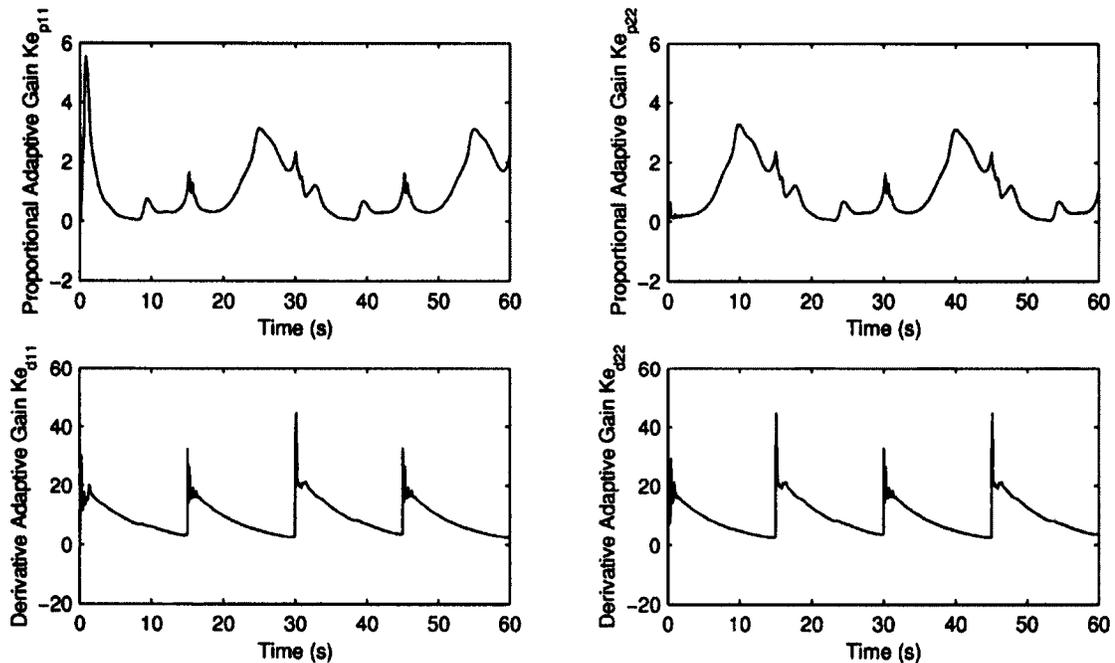


Figure 4.47: Adaptation history of the DSAC composite controller gains $K_{e_p}(t)$ and $K_{e_d}(t)$ when applied to the nonlinear joint stiffness manipulator.

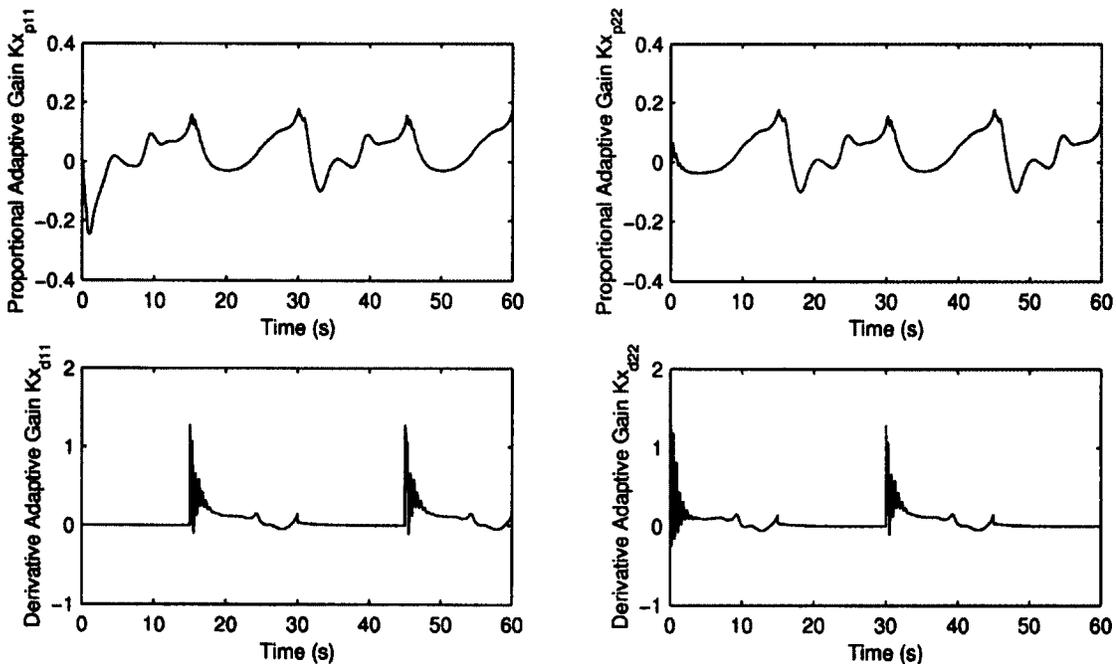


Figure 4.48: Adaptation history of the DSAC composite controller gains $K_{x_p}(t)$ and $K_{x_d}(t)$ when applied to the nonlinear joint stiffness manipulator.

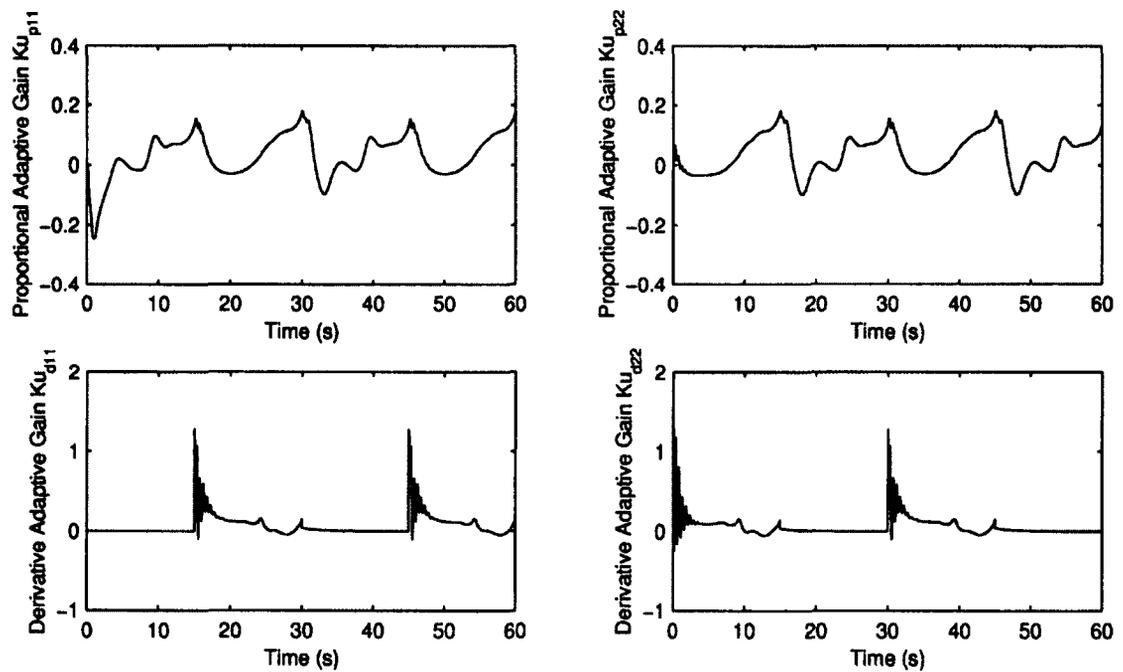


Figure 4.49: Adaptation history of the DSAC composite controller gains $K_{u_p}(t)$ and $K_{u_d}(t)$ when applied to the nonlinear joint stiffness manipulator.

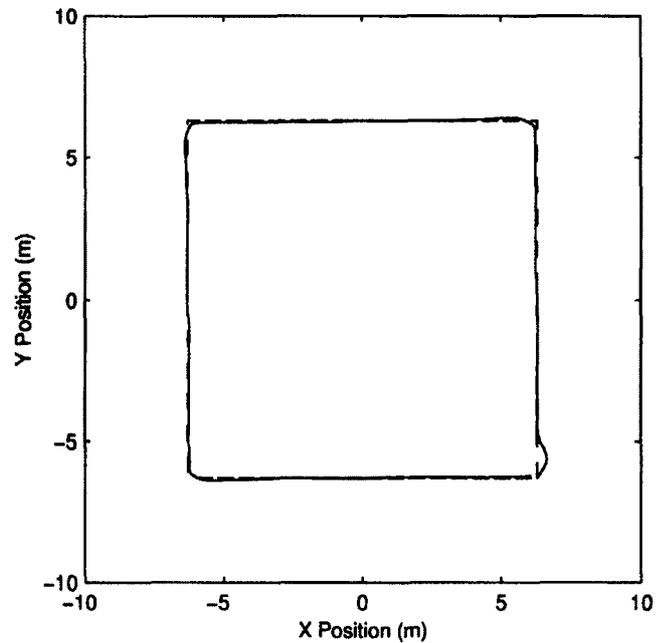


Figure 4.50: Trajectory tracking results obtained with the DMSAC composite controller applied to the nonlinear joint stiffness manipulator. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$.

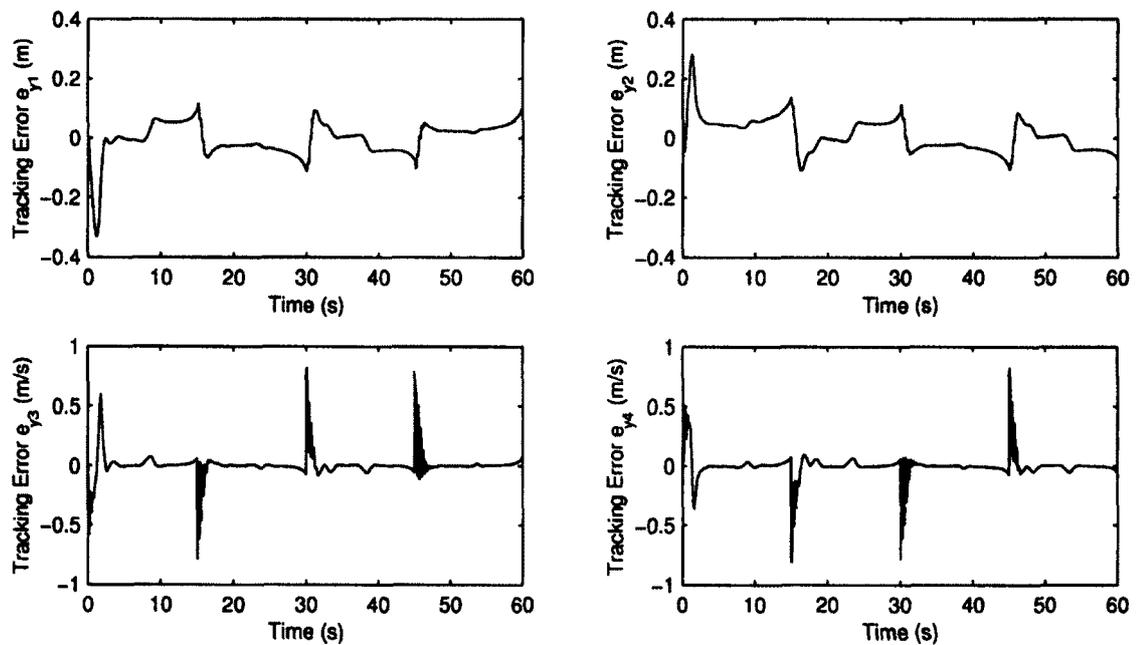


Figure 4.51: Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the DMSAC composite controller applied to the nonlinear joint stiffness manipulator.

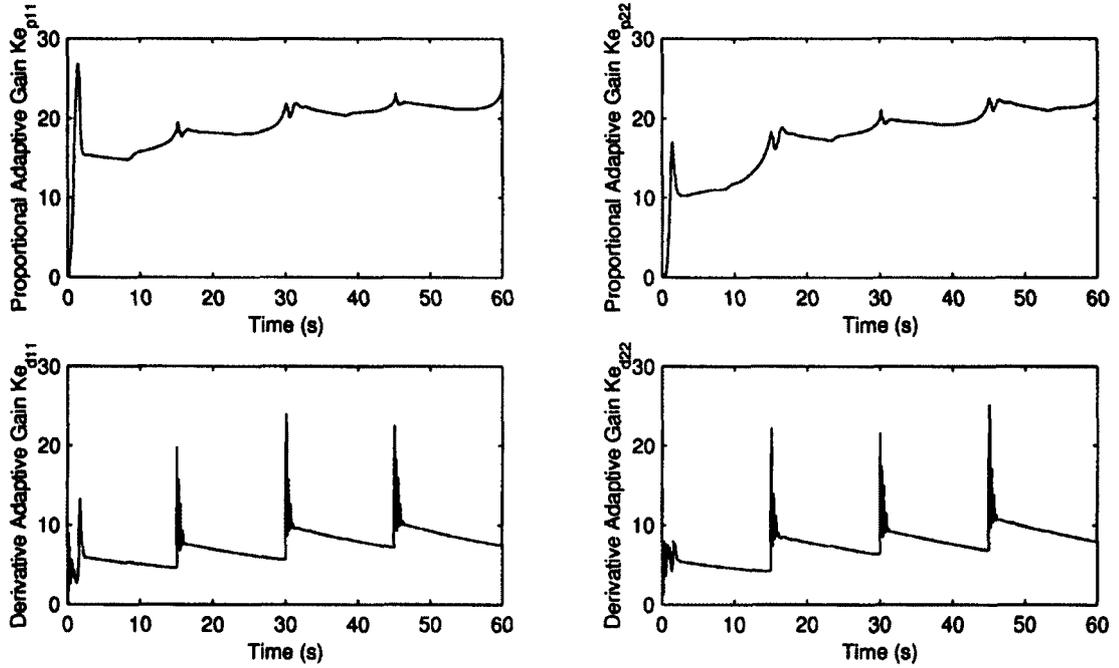


Figure 4.52: Adaptation history of the DMSAC composite controller gains $K_{e_p}(t)$ and $K_{e_d}(t)$ when applied to the nonlinear joint stiffness manipulator.

4.5.3 Comparison to a Model-Based Flexible-Joint Controller

For completeness, and to demonstrate the benefits of using a model-independent direct adaptive control approach, the composite controllers developed in this chapter are compared to a conventional model-based controller. For this purpose, the nonlinear backstepping controller proposed by Brogliato *et al.* [28] is used. The nonlinear backstepping control law is given by [28]

$$\tau = J_m [\ddot{q}_{m_d} - 2(\dot{q}_m - \dot{q}_{m_d}) - 2(q_m - q_{m_d}) - (\dot{s} + s)] + k(q_m - q) \quad (4.57)$$

where $s \in \mathbb{R}^n$ is given by

$$s = (\dot{q} - \dot{q}_d) + \Lambda (q - q_d) \quad (4.58)$$

with $\Lambda \in \mathbb{R}^{n \times n}$ denoting a constant gain matrix. In (4.57), q_{m_d} , \dot{q}_{m_d} , $\ddot{q}_{m_d} \in \mathbb{R}^n$ denote

the desired motor angular position, velocity and acceleration vectors respectively, which are given by

$$q_{md} = k^{-1}\tau_{SLI} + q \quad (4.59)$$

$$\dot{q}_{md} = k^{-1}\dot{\tau}_{SLI} + \dot{q} \quad (4.60)$$

$$\ddot{q}_{md} = k^{-1}\ddot{\tau}_{SLI} + \ddot{q} \quad (4.61)$$

In (4.59)-(4.61), the term τ_{SLI} denotes the well-known adaptive control law for rigid-joint robots proposed by Slotine and Li [160]. More details on the nonlinear backstepping controller, including the proof of stability, can be found in Appendix D.

As previously, the control gains of this control law were selected in numerical simulations, to provide good trajectory tracking results when applied to the nominal linear joint stiffness manipulator system. The backstepping control gains were determined as

$$K_p = 2I_2 \quad \Lambda = I_2$$

The results obtained with this control strategy show rapid settling to a straight line along each side of the trajectory and no overshoots at the corners, as illustrated in Figs. 4.53 and 4.54. However, the high trajectory tracking accuracy obtained for the nominal manipulator dynamics is achieved at the expense of greater complexity in the controller structure, which involves physical parameters of the system in a highly nonlinear way, as shown in Appendix D. This problem is typical with model-based controllers. With this specific controller, the increased complexity is largely due to the real-time analytical calculation of the inverse of the robot inertia matrix $M^{-1}(q)$, and the time-derivative of the inertia and the Coriolis/centrifugal matrix, $\dot{M}(q)$ and $\dot{C}(q, \dot{q})$ respectively.

When applied to the manipulator system with uncertainties in its joint stiffness matrix ($k = 200I_2$ N·m/rad), the response of the controller was unstable. This is due to the dependence of the backstepping control algorithm on exact knowledge of k ,

the joint stiffness matrix, as observed in (4.59)-(4.61). As a result, any discrepancies between the stiffness coefficient values used in the control law and the actual values causes poor tracking performance, or even closed-loop instability. This divergence occurred even when tested with a manipulator with a stiffness matrix of $k = 400I_2$ N·m/rad, which represents a significantly smaller deviation from the nominal stiffness matrix ($k = 500I_2$ N·m/rad).

The same unstable behavior was also observed when the controller was applied to the nonlinear joint stiffness model. Indeed, the development of the backstepping control law in (4.57) is based on exact knowledge of the dynamics model structure. Any variations in the actual dynamics or system structure have a detrimental effect on the tracking performance, and can potentially lead to an unstable closed-loop response, as suggested by the results obtained. The results support those obtained by Brogliato *et al.* [28], and illustrate the lack of robustness to uncertainties and modeling errors of conventional model-based flexible-joint controllers.

In addition, this model-based controller may not be the best for practical use, since the very high control torque magnitudes initially required to increase the end-effector velocity from rest to 0.84 m/s along the y -axis (538 N·m and 9.30×10^3 N·m for joints 1 and 2 respectively) cannot be achieved by typical joint mechanisms. Finally, another drawback of this joint-based controller is the real-time calculation of the robot inverse kinematics required to transform the desired end-effector trajectory into joint positions and velocities.

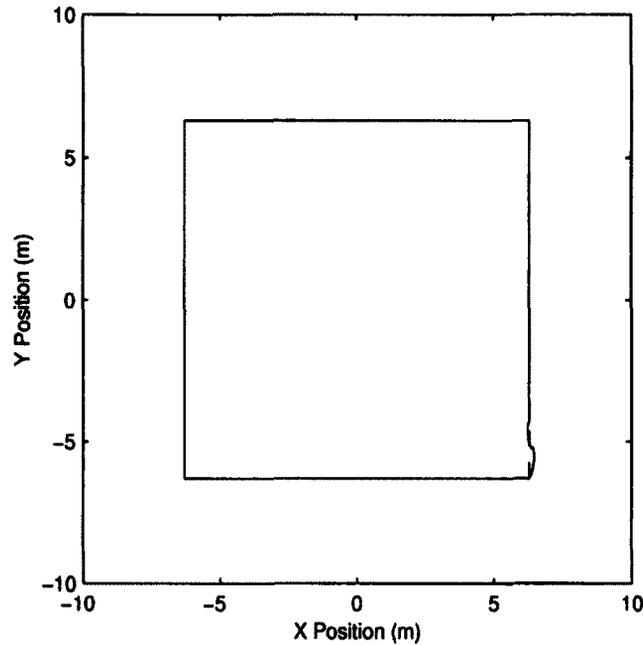


Figure 4.53: Trajectory tracking results obtained with the nonlinear backstepping controller applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m). The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$.

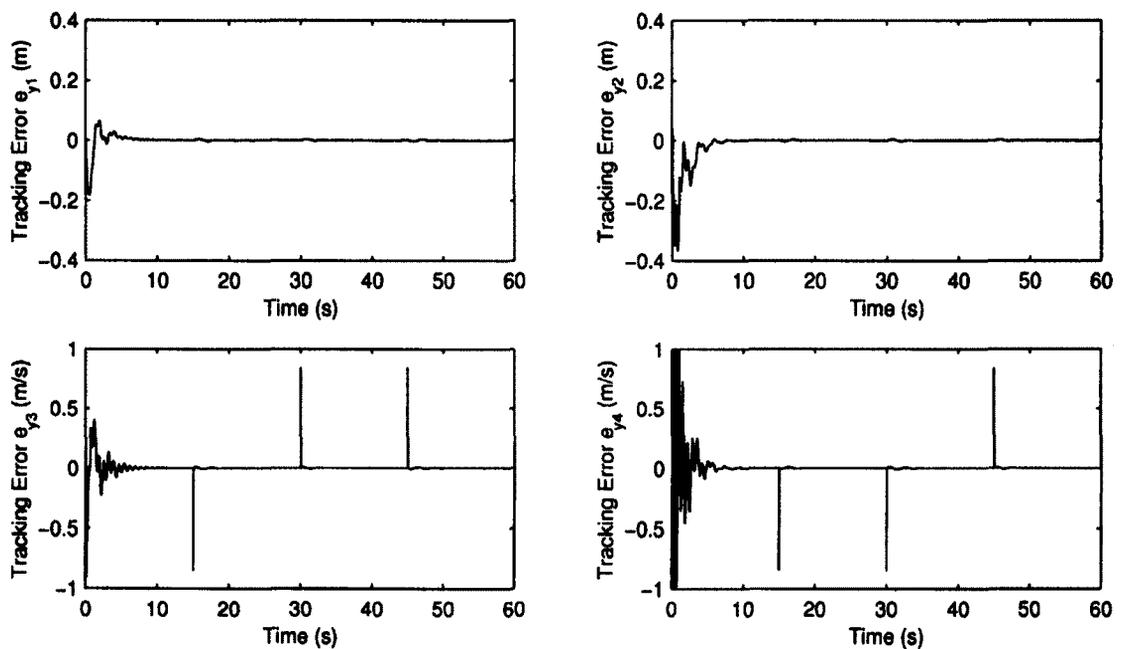


Figure 4.54: Trajectory tracking errors ($y_e(t) = y_d(t) - y(t)$) obtained with the nonlinear backstepping controller applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m).

4.6 Summary

To overcome the limitations of the usual linear joint stiffness dynamics model, this chapter developed a new nonlinear joint stiffness formulation. This proposed nonlinear dynamics model for flexible-joint manipulators takes several factors into account that are disregarded neglected in most existing flexible-joint models: soft-windup, friction, inertial coupling and nonlinear and nonstationary characteristics of joint elasticity. To the best of our knowledge, this is the first time such a realistic flexible-joint model has been presented. This newly developed model formed the basis for the investigation of the controller robustness to modeling errors in this work. Subsequently, novel direct adaptive composite controllers were developed, in which a transpose Jacobian slow control term is added to a fast-control term designed to dampen the elastic vibrations at the joints of a flexible-joint manipulator system. The control gain adaptation mechanisms of the proposed control methodologies are based on a normalized fuzzy logic system, the DSAC algorithm and the DMSAC algorithm. These adaptive systems calculate time-varying control parameters in response to tracking errors between the actual system outputs and the outputs of a reference model, then adapt the control gains to minimize trajectory tracking errors. The applicability and effectiveness of the adaptive controller methodologies under adverse conditions, such as parametric uncertainties and modeling errors, were demonstrated in numerical simulations with both the classic linear joint dynamics model and the derived nonlinear joint stiffness model. Simulation results established that the proposed direct composite control methodologies effectively manage significant uncertainties and modeling errors in the manipulator system, by time-varying the control parameters accordingly. The developed adaptive strategies achieved better tracking results than both a nonadaptive model-independent controller and a conventional nonlinear model-based control scheme. The findings clearly indicate that the newly-developed direct adaptive control strategies provide better tracking results than conventional model-based control techniques or nonadaptive model-independent methodologies, regardless of the presence of uncertainties and modeling errors in the manipulator system.

Chapter 5

Partial State Feedback Adaptive Composite Control

Most advanced trajectory tracking control laws for robot manipulators require knowledge of all the state variables. For lightweight flexible-joint space manipulators this is difficult to achieve, since link positions and velocities are typically not measured. This chapter first develops an extended Kalman filter (EKF) strategy to estimate all state variables from noisy and biased measurements in a flexible-joint manipulator system modeled with the linear joint stiffness dynamics model. Secondly, an extension of the EKF for the new nonlinear joint stiffness dynamics formulation is presented. The state estimates are coupled to the DMSAC composite controller, to provide a partial state feedback control scheme. Numerical simulations are used to demonstrate the trajectory tracking performance of the resulting partial state feedback strategy for both dynamics representations.

5.1 Introduction

The main drawback of most existing flexible-joint controllers is that their implementation requires measurements of all the state variables (i.e. link position q , link velocity \dot{q} , motor position q_m and motor velocity \dot{q}_m). Although some advanced space robot systems have access to measurements providing knowledge of joint elasticity effects, such as the DLR's Lightweight Robot III that is equipped with joint torque and link position sensors [161], typical robot manipulators are instrumented to measure only motor variables, q_m and \dot{q}_m , with an encoder and a tachometer on each motor axis of the manipulator [156]. In practice, measurement devices (or sensors) add extra weight and cost to the system. On the other hand, any velocity measurements provided through numerical differentiation are prone to contamination from noise, which degrades the controller performance.

To reduce the number of sensors required for state variable measurements, partial state feedback (i.e. output feedback) control algorithms have been suggested in the literature. These output feedback controllers are usually designed based on reconstructing the missing signals from the available physical measurements. Oh and Lee [162] proposed an output feedback controller that makes the link positions track the reference signal semiglobally exponentially fast while keeping all the remaining signals bounded. The controller was based on the integrator backstepping method where two observers are used to substitute link and actuator velocity measurements. A nonlinear tracking controller based on singularity perturbed systems and sliding-mode techniques, that could ensure that link positions approached the reference signal semiglobally asymptotically based on link position measurements only was provided by De Leon-Morales *et al.* [163]. Lim *et al.* [164] proposed an adaptive semiglobal tracking controller, assuming link positions and motor positions were available from measurements. Based on the work Loria [165] on a single-link rigid-joint robot, Dixon *et al.* [166] addressed the global output feedback tracking control problem of flexible-joint robots. More specifically, they proposed an output feedback controller that ensures global asymptotic link position tracking, and only requires link position measurements. In their approach, hyperbolic trigonometric functions were introduced in both the observer-based controller and the stability analysis, to automatically enlarge the domain of attraction while keeping the controller and observer gains constant. A drawback of this strategy is that the resulting control law contains terms that increase exponentially, thus yielding high control input signals to the system. Nicosia *et al.* [167] proposed a procedure to construct approximate nonlinear observers using an approach to derive observers based on the geometric nonlinear control theory, was combined with an approximation technique. Selecting the link coordinates and their time derivatives as the system outputs, Tomei [168] proposed a nonlinear observer which reconstructs all state variables. Chatlatanagulchai *et al.* [169] designed a neural network observer to determine link and motor positions and velocities, and combined it with a robust controller. Similarly, Abdollahi *et al.* [170] presented a stable neural network-based observer for flexible-joint robots. Nicosia and Tomei [171] proposed a

controller that requires only link positions, with the other state variables provided by a nonlinear observer. A global asymptotic link position tracking controller that requires only link and motor position measurements was proposed by Dixon *et al.* [172]. Specifically, the authors used a nonlinear link velocity filter to eliminate link velocity measurements, while a set of linear filters was employed to eliminate the need for motor velocity measurements. Kim and Lee designed an adaptive controller based on link and actuator position measurements only, with link and motor velocity filters used to estimate the unknown velocity terms [173]. Using link and motor positions, Melhem and Wang [174] proposed a controller that ensures the link positions globally track the desired trajectory, while keeping all the remaining signals bounded. Their controller used a partial state feedback linearization technique combined with the integrator backstepping control design method, with a filter and an observer were used to remove the requirement of link and actuator velocity measurements. A major problem with all the above methods is that link angular position measurements are required in the observation process.

As well as using nonlinear observers, another way to estimate in the state variables in real-time that not available through measurements is to apply the Kalman filter (KF) theory. Goudreau and Schwartz [175] developed an extended Kalman filter (EKF) to estimate joint positions and velocities based on joint measurements and control torques for a direct-drive two-link rigid-joint manipulator. Lertpiriyasuwat and Burg [176] proposed a KF that combines end-effector position measurements obtained from a laser tracker sensor with link position measurements, to estimate end-effector positions and orientations in an industrial robot. However, in these studies the KF/EKF designs are based on a rigid-joint dynamic representation.

Timcenko and Kircanski [177] developed a linear KF to estimate the control torque in a flexible-joint robot, and used it in a feedforward/feedback controller scheme. Hollars and Cannon [178] used a constant-gain EKF (CGEKF) to estimate the state variables of a planar two-link robot arm with rigid links and flexibility in its joints, for use by a state feedback control law. A significant advantage of the CGEKF approach is that the computational load is much less than that of the EKF. However, as stated in [179], the stability of such a system must be carefully evaluated, since the problem is

inherently nonlinear. In 2010, Lightcap and Banks [157] presented an EKF to estimate link and motor positions/velocities based on motor measurements. Although their approach did not use directly link position measurements, a real-time knowledge of link positions was provided by sets of retro-reflective markers positioned on the links; an uncommon sensor for robotic manipulators, particularly those operating in space. A major problem with these Kalman filter-based methodologies for flexible-joint robots is that, despite the fact that experimental studies have shown that flexible gears are much more complex than a linear spring [92], their designs and simulation validations were based on the classical dynamic representation proposed by Spong [45], which models each joint as a linear torsional spring of constant stiffness.

The first contribution of this chapter is the design of two EKFs that estimate link and motor positions/velocities for a flexible-joint space robotic manipulator: (1) the first EKF is designed for a robot modeled with the linear joint stiffness model, and (2) the second EKF is proposed for a space manipulator modeled with the novel nonlinear joint stiffness dynamics formulation proposed in Chapter 4. While the second EKF expands on previous work by incorporating nonlinear flexible effects in the process model, such as nonlinear joint stiffness and soft-windup, both EKFs are innovative in the sense that their applicability to a manipulator equipped only with motor encoders and tachometers is demonstrated.

In addition, the proposed nonlinear estimation schemes are modified to consider measurement biases. Finally, the state variable estimates are coupled to the DMSAC-based composite control methodology developed in Chapter 4 to provide a closed-loop partial state feedback trajectory tracking strategy that requires only motor position and velocity measurements. For completeness, a brief review of the EKF theory is provided in Appendix E.

5.2 Partial State Feedback Objective

The partial state feedback control objective consists in designing EKF-based state estimation techniques which, when combined with the DMSAC-based composite controller, ensure that the estimated output vector $\hat{y}(t) \in \mathbb{R}^4$ of the flexible-joint manipulator system tracks the time-varying reference model output $y_m(t) \in \mathbb{R}^4$, considering

that noisy and biased measurements are provided by motor encoders and tachometers. To quantify this objective, a tracking error, denoted $\tilde{e}_y(t) \in \mathbb{R}^4$, is defined as

$$\tilde{e}_y \triangleq y_m - \hat{y} \quad (5.1)$$

where the estimated output vector is calculated as a function of the estimated link positions and velocities, denoted by $\hat{q}, \dot{\hat{q}} \in \mathbb{R}^2$, respectively.

5.3 Flexible-Joint Manipulator Estimation

Let define the state vector, denoted by $x \in \mathbb{R}^8$, as

$$x \triangleq [q^T \quad \dot{q}^T \quad q_m^T \quad \dot{q}_m^T]^T \quad (5.2)$$

such that $\dot{x} = f(x, u)$, and the partial derivatives of the robot dynamics with respect to the states are

$$F = \frac{\partial f(x, u)}{\partial x} = \begin{bmatrix} 0 & I_2 & 0 & 0 \\ F_{21} & F_{22} & F_{23} & 0 \\ 0 & 0 & 0 & I_2 \\ F_{41} & 0 & F_{43} & 0 \end{bmatrix} \quad (5.3)$$

5.3.1 Linear Joint Stiffness Dynamics

For the linear joint stiffness dynamics model given by (4.3) and (4.4), $f(x, u)$ is given by

$$f(x, u) = \begin{bmatrix} \dot{q} \\ M^{-1}(q) [k(q_m - q) - C(q, \dot{q})\dot{q}] \\ \dot{q}_m \\ J_m^{-1} [\tau - k(q_m - q)] \end{bmatrix} \quad (5.4)$$

and the partial derivatives in (5.3) are given by

$$F_{21} = -M^{-1}(q) \left[\frac{\partial M(q)}{\partial q} \dot{q} + \frac{\partial C(q, \dot{q})}{\partial q} \dot{q} + k \right] \quad (5.5)$$

$$F_{22} = -M^{-1}(q) \left[\frac{\partial C(q, \dot{q})}{\partial \dot{q}} \dot{q} + C(q, \dot{q}) \right] \quad (5.6)$$

$$F_{23} = M^{-1}(q)k \quad (5.7)$$

$$F_{41} = J_m^{-1}k \quad (5.8)$$

$$F_{43} = -F_{41} \quad (5.9)$$

In (5.5), the partial derivative of the robot inertia matrix and Coriolis matrix with respect to the link positions is given by

$$\frac{\partial M(q)}{\partial q} \ddot{q} = -m_2 l_1 l_{c_2} \sin q_2 \begin{bmatrix} 0 & 2\ddot{q}_1 + \ddot{q}_2 \\ 0 & \ddot{q}_1 \end{bmatrix} \quad (5.10)$$

$$\frac{\partial C(q, \dot{q})}{\partial q} \dot{q} = -m_2 l_1 l_{c_2} \cos q_2 \begin{bmatrix} 0 & \dot{q}_1 \dot{q}_2 + \dot{q}_2 (\dot{q}_1 + \dot{q}_2) \\ 0 & -\dot{q}_1^2 \end{bmatrix} \quad (5.11)$$

and the partial derivative in (5.6) necessary for the computation of F_{22} is given by

$$\frac{\partial C(q, \dot{q})}{\partial \dot{q}} \dot{q} = -m_2 l_1 l_{c_2} \sin q_2 \begin{bmatrix} \dot{q}_2 & (\dot{q}_1 + \dot{q}_2) \\ -\dot{q}_1 & 0 \end{bmatrix} \quad (5.12)$$

5.3.2 Nonlinear Joint Stiffness Dynamics

Considering the nonlinear joint stiffness model given by (4.19) and (4.20), $f(x, u)$ is

$$f(x, u) = \begin{bmatrix} \dot{q} \\ M^{-1}(q) [k(q, q_m)(q_m - q) - f(\dot{q}) - C(q, \dot{q})\dot{q} - S\ddot{q}_m] \\ \dot{q}_m \\ J_m^{-1} [\tau - k(q, q_m)(q_m - q) - S^T \ddot{q}] \end{bmatrix} \quad (5.13)$$

and the partial derivatives of the robot dynamics are

$$F_{21} = -M^{-1}(q) \left[\frac{\partial M(q)}{\partial q} \ddot{q} + \frac{\partial C(q, \dot{q})}{\partial q} \dot{q} - \frac{\partial k(q, q_m)}{\partial q} (q_m - q) + k(q, q_m) \right] \quad (5.14)$$

$$F_{22} = -M^{-1}(q) \left[\frac{\partial C(q, \dot{q})}{\partial \dot{q}} \dot{q} + \frac{\partial f(\dot{q})}{\partial \dot{q}} + C(q, \dot{q}) \right] \quad (5.15)$$

$$F_{23} = M^{-1}(q) \left[\frac{\partial k(q, q_m)}{\partial q_m} (q_m - q) + k(q, q_m) \right] \quad (5.16)$$

$$F_{41} = -J_m^{-1} \left[\frac{\partial k(q, q_m)}{\partial q} (q_m - q) - k(q, q_m) \right] \quad (5.17)$$

$$F_{43} = -J_m^{-1} \left[\frac{\partial k(q, q_m)}{\partial q_m} (q_m - q) + k(q, q_m) \right] \quad (5.18)$$

where

$$\frac{\partial k(q_m - q)}{\partial q} (q_m - q) = -\frac{\partial k(q_m - q)}{\partial q_m} (q_m - q) = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} \quad (5.19)$$

$$\frac{\partial f(\dot{q})}{\partial \dot{q}} = \begin{bmatrix} \frac{\partial f(\dot{q})}{\partial \dot{q}_1} & \frac{\partial f(\dot{q})}{\partial \dot{q}_2} \end{bmatrix} = \begin{bmatrix} c & 0 \\ 0 & d \end{bmatrix} \quad (5.20)$$

with

$$a = -2(q_{m1} - q_1)^2 \left[a_{111} + k_{sw11} a_{sw} e^{-a_{sw}(q_{m1} - q_1)^2} \right] \quad (5.21)$$

$$b = -2(q_{m2} - q_2)^2 \left[a_{122} + k_{sw22} a_{sw} e^{-a_{sw}(q_{m2} - q_2)^2} \right] \quad (5.22)$$

$$c = \gamma_1 \left[\gamma_2 \operatorname{sech}^2(\gamma_2 \dot{q}_1) - \gamma_3 \operatorname{sech}^2(\gamma_3 \dot{q}_1) + \gamma_4 \gamma_5 \operatorname{sech}^2(\gamma_5 \dot{q}_1) + \gamma_6 \right] \quad (5.23)$$

$$d = \gamma_1 \left[\gamma_2 \operatorname{sech}^2(\gamma_2 \dot{q}_2) - \gamma_3 \operatorname{sech}^2(\gamma_3 \dot{q}_2) + \gamma_4 \gamma_5 \operatorname{sech}^2(\gamma_5 \dot{q}_2) + \gamma_6 \right] \quad (5.24)$$

5.3.3 Measurement Model

By considering that the only available measurements are provided by an encoder and tachometer on the motor sides, the measurement model denoted by $h(x) \in \mathbb{R}^4$ can be defined as

$$h(x) \triangleq \begin{bmatrix} q_m^T & \dot{q}_m^T \end{bmatrix}^T \quad (5.25)$$

The linearization of this measurement model for the robot dynamics is thus given by

$$H = \frac{\partial h(x)}{\partial x} = \begin{bmatrix} 0 & 0 & I_2 & 0 \\ 0 & 0 & 0 & I_2 \end{bmatrix} \quad (5.26)$$

5.3.4 Observability Analysis

In control theory, the concept of observability refers to a measure of how well internal states of a system can be inferred by knowledge of its external outputs. A system is observable if, for any possible sequence of state and control input vectors, the actual state can be determined in finite time using only the outputs. In other words, the outputs contain all sufficient information to completely determine the behaviour of the system. Thus, a system is not observable when the values of some of its actual states cannot be determined through output measurements. A suitable method to determine the observability of a system is the observability matrix rank condition. For nonlinear systems with the state-space realization

$$\dot{x} = f(x, u) \quad (5.27)$$

$$y = h(x) \quad (5.28)$$

the rank condition is tied to the Lie derivative of the measurement functions $h(x)$ along the process function $f(x, u)$, denoted by $(\mathcal{L}_f h)(x)$, and defined as [180]

$$(\mathcal{L}_f h)(x) \triangleq \nabla h(x) f(x, u) \quad (5.29)$$

where ∇ denotes the gradient operator. By definition, the zeroth-order Lie derivative denoted by $(\mathcal{L}_f^0 h)(x)$ is

$$(\mathcal{L}_f^0 h)(x) \triangleq h(x) \quad (5.30)$$

such that the gradient of $(\mathcal{L}_f^0 h)(x)$ is given by

$$\nabla(\mathcal{L}_f^0 h)(x) = H \quad (5.31)$$

The higher-order Lie derivatives are defined as

$$(\mathcal{L}_f^k h)(x) \triangleq \nabla(\mathcal{L}_f^{k-1} h)(x) f(x, u); \quad \forall k > 0 \quad (5.32)$$

Defining the observation space \mathcal{O} as the space containing all Lie derivatives of $h(x)$ up to order $N - 1$, where $N \in \mathbb{R}$ is the order of the system, the observability matrix denoted O is obtained as the gradient of \mathcal{O} , as follows

$$O = \begin{bmatrix} \nabla(\mathcal{L}_f^0 h)(x) \\ \vdots \\ \nabla(\mathcal{L}_f^{N-1} h)(x) \end{bmatrix} \quad (5.33)$$

The number of columns of O is equal to $n \in \mathbb{R}$, the dimension of the state vector x , while the number of rows is determined by m the dimension of the measurement model $h(x)$ and by N , and is equal to $m \cdot N$. However, in the present case, recalling that the dynamics model consists in two second-order differential equations, the order of the system is $2N$, with $N = 2$. As a result, Lie derivatives up to the third order will be required.

Formally, the nonlinear system with state-space realization (5.27) and (5.28) is said to be locally observable at $x_0 \in \mathbb{R}^n$ if there exists a neighborhood $U_0 \subset \mathbb{R}^n$ of x_0 in which every state $x \neq x_0$ is distinguishable from x_0 . In other words, there must exist a unique mapping between each observation and respective state. The sufficient condition for the nonlinear system to be locally observable in the neighborhood U_0 containing the origin is

$$\text{rank}(O) = n \quad (5.34)$$

In other words, O must be full rank. Note that (5.34) is a necessary and sufficient condition for observability in nonlinear systems.

Considering the state vector given by (5.2) and measurement model defined in (5.25) and (5.26), the first Lie derivative and its gradient are respectively obtained as follows

$$\begin{aligned} (\mathcal{L}_f^1 h)(x) &= Hf(x, u) \\ &= \begin{bmatrix} 0 & 0 & I_2 & 0 \\ 0 & 0 & 0 & I_2 \end{bmatrix} \begin{bmatrix} \dot{q} \\ \ddot{q} \\ \dot{q}_m \\ \ddot{q}_m \end{bmatrix} \\ &= \begin{bmatrix} \dot{q}_m \\ \ddot{q}_m \end{bmatrix} \end{aligned} \quad (5.35)$$

$$\nabla(\mathcal{L}_f^1 h)(x) = \begin{bmatrix} 0 & 0 & 0 & I_2 \\ F_{41} & 0 & F_{43} & 0 \end{bmatrix} \quad (5.36)$$

The second Lie derivative and its gradient are given as

$$\begin{aligned}
(\mathcal{L}_f^2 h)(x) &= \nabla(\mathcal{L}_f h)(x) f(x, u) \\
&= \begin{bmatrix} 0 & 0 & 0 & I_2 \\ F_{41} & 0 & F_{43} & 0 \end{bmatrix} \begin{bmatrix} \dot{q} \\ \ddot{q} \\ \dot{q}_m \\ \ddot{q}_m \end{bmatrix} \\
&= \begin{bmatrix} \ddot{q}_m \\ F_{41} \dot{q} + F_{43} \dot{q}_m \end{bmatrix} \tag{5.37}
\end{aligned}$$

$$\nabla(\mathcal{L}_f^2 h)(x) = \begin{bmatrix} F_{41} & 0 & F_{43} & 0 \\ 0 & F_{41} & 0 & F_{43} \end{bmatrix} \tag{5.38}$$

Finally, the third Lie derivative and its gradient are obtained as

$$\begin{aligned}
(\mathcal{L}_f^3 h)(x) &= \nabla(\mathcal{L}_f^2 h)(x) f(x, u) \\
&= \begin{bmatrix} F_{41} & 0 & F_{43} & 0 \\ 0 & F_{41} & 0 & F_{43} \end{bmatrix} \begin{bmatrix} \dot{q} \\ \ddot{q} \\ \dot{q}_m \\ \ddot{q}_m \end{bmatrix} \\
&= \begin{bmatrix} F_{41} \dot{q} + F_{43} \dot{q}_m \\ F_{41} \ddot{q} + F_{43} \ddot{q}_m \end{bmatrix} \tag{5.39}
\end{aligned}$$

$$\nabla(\mathcal{L}_f^3 h)(x) = \begin{bmatrix} 0 & F_{41} & 0 & F_{43} \\ F_{41} F_{21} + F_{43} F_{41} & F_{41} F_{22} & F_{41} F_{23} + F_{43} F_{43} & 0 \end{bmatrix} \tag{5.40}$$

This leads to the observability matrix

$$O = \begin{bmatrix} 0 & 0 & I_2 & 0 \\ 0 & 0 & 0 & I_2 \\ 0 & 0 & 0 & I_2 \\ F_{41} & 0 & F_{43} & 0 \\ F_{41} & 0 & F_{43} & 0 \\ 0 & F_{41} & 0 & F_{43} \\ 0 & F_{41} & 0 & F_{43} \\ F_{41}F_{21} + F_{43}F_{41} & F_{41}F_{22} & F_{41}F_{23} + F_{43}F_{43} & 0 \end{bmatrix} \quad (5.41)$$

For a matrix where the number of rows is greater than the number of columns, the maximum rank is equal to the number of columns. In (5.41), O has $m \cdot 2N = 16$ rows and $n = 8$ columns. Thus, the maximum rank is 8, the number of state variables in the EKF. Upon inspection, all columns in O are linearly independent. The matrix is then said to be full rank. In other words, its rank is the maximum rank, i.e. 8. Note that the matrices F_{41} and F_{43} are linearly dependent upon each other, but this does not affect the rank of the observability matrix.

5.4 Bias Estimation

One of the underlying assumptions of the EKF formulation is that the measurement noise is zero-mean Gaussian, with a standard deviation σ . However, measurements are commonly corrupted by biases as well, hence the need for on-board calibration. In this section, the bias-free extended Kalman filter (EKF) estimation strategy developed in Section 5.3 is modified to compensate for measurement biases in real-time. The proposed strategy is to augment the state vector with an additional state representing the unknown measurement biases. Let the augmented state vector denoted by $x_{aug} \in \mathbb{R}^{8+m}$ be defined as

$$x_{aug} \triangleq \begin{bmatrix} x^T & b^T \end{bmatrix}^T \quad (5.42)$$

where $b \in \mathbb{R}^m$ denotes the measurement bias vector. Typically, since there is no *a priori* information about b , it is assumed that

$$\dot{b} = 0 \quad (5.43)$$

which indicates that the biases are varying slowly with respect to the plant dynamics. Because of this dynamics model, the propagation of b is obtained as

$$\hat{b}_{k+1}^- = \hat{b}_k \quad (5.44)$$

where \hat{b}_{k+1}^- , $\hat{b}_k \in \mathbb{R}^m$ respectively denote the propagated measurement bias vector and the estimate of the measurement bias vector. For the two-link robot manipulator application, it is assumed that the motor angular positions and velocities are obtained from the motor encoder and tachometer noisy and biased measurements, thus resulting in the following augmented state vector and measurement model

$$x_{aug} \triangleq \begin{bmatrix} q^T & \dot{q}^T & q_m^T & \dot{q}_m^T & b_e^T & b_t^T \end{bmatrix}^T \in \mathbb{R}^{12} \quad (5.45)$$

$$h(x_{aug}) \triangleq \begin{bmatrix} (q_m + b_e)^T & (\dot{q}_m + b_t)^T \end{bmatrix}^T \in \mathbb{R}^4 \quad (5.46)$$

where $b_e, b_t \in \mathbb{R}^2$ denote the encoder and tachometer bias vector, respectively. Hence, the partial derivatives of the robot dynamics with respect to x_{aug} is given by

$$F_{aug} = \frac{\partial f(x_{aug}, u)}{\partial x} = \begin{bmatrix} 0 & I_2 & 0 & 0 & 0 & 0 \\ F_{21} & F_{22} & F_{23} & 0 & 0 & 0 \\ 0 & 0 & 0 & I_2 & 0 & 0 \\ F_{41} & 0 & F_{43} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.47)$$

and the partial derivatives of $h(x_{aug})$ with respect to x_{aug} can be written as

$$H_{aug} = \frac{\partial h(x_{aug})}{\partial x_{aug}} = \begin{bmatrix} 0 & 0 & I_2 & 0 & I_2 & 0 \\ 0 & 0 & 0 & I_2 & 0 & I_2 \end{bmatrix} \quad (5.48)$$

The observability matrix, taking into account x_{aug} and h_{aug} , becomes

$$O = \begin{bmatrix} 0 & 0 & I_2 & 0 & I_2 & 0 \\ 0 & 0 & 0 & I_2 & 0 & I_2 \\ 0 & 0 & 0 & I_2 & 0 & 0 \\ F_{41} & 0 & F_{43} & 0 & 0 & 0 \\ F_{41} & 0 & F_{43} & 0 & 0 & 0 \\ 0 & F_{41} & 0 & F_{43} & 0 & 0 \\ 0 & F_{41} & 0 & F_{43} & 0 & 0 \\ F_{41}F_{21} + F_{43}F_{41} & F_{41}F_{22} & F_{41}F_{23} + F_{43}F_{43} & 0 & 0 & 0 \end{bmatrix} \quad (5.49)$$

The observability matrix O given by (5.49) has $m \cdot 2N = 16$ rows and $n = 12$ columns. Thus, the maximum rank is 12, the number of state variables in the EKF. As previously, all columns in O are linearly independent and O is full rank. As a result, the observability rank condition is satisfied, a necessary and sufficient condition for observability in nonlinear systems.

5.5 Simulation Results

In this section, the linear joint-based EKF is applied to the linear joint stiffness dynamics (4.3) and (4.4), and the nonlinear joint-based EKF is applied to the nonlinear joint stiffness dynamics (4.19) and (4.20). The sample MATRIXx programming code that implements the nonlinear joint stiffness-based EKF is provided in Appendix F. To achieve a complete closed-loop adaptive partial state-feedback strategy, and to demonstrate the applicability of both estimation schemes, the DMSAC control methodology presented in Chapter 4.4.4 is used, as follows

$$\tau = J^T(\hat{q})K_e(t)\tilde{e}_y + K_v(\dot{\hat{q}} - \dot{\hat{q}}_m) \quad (5.50)$$

A block diagram realization of the complete estimation and control strategy is shown in Fig. 5.1.

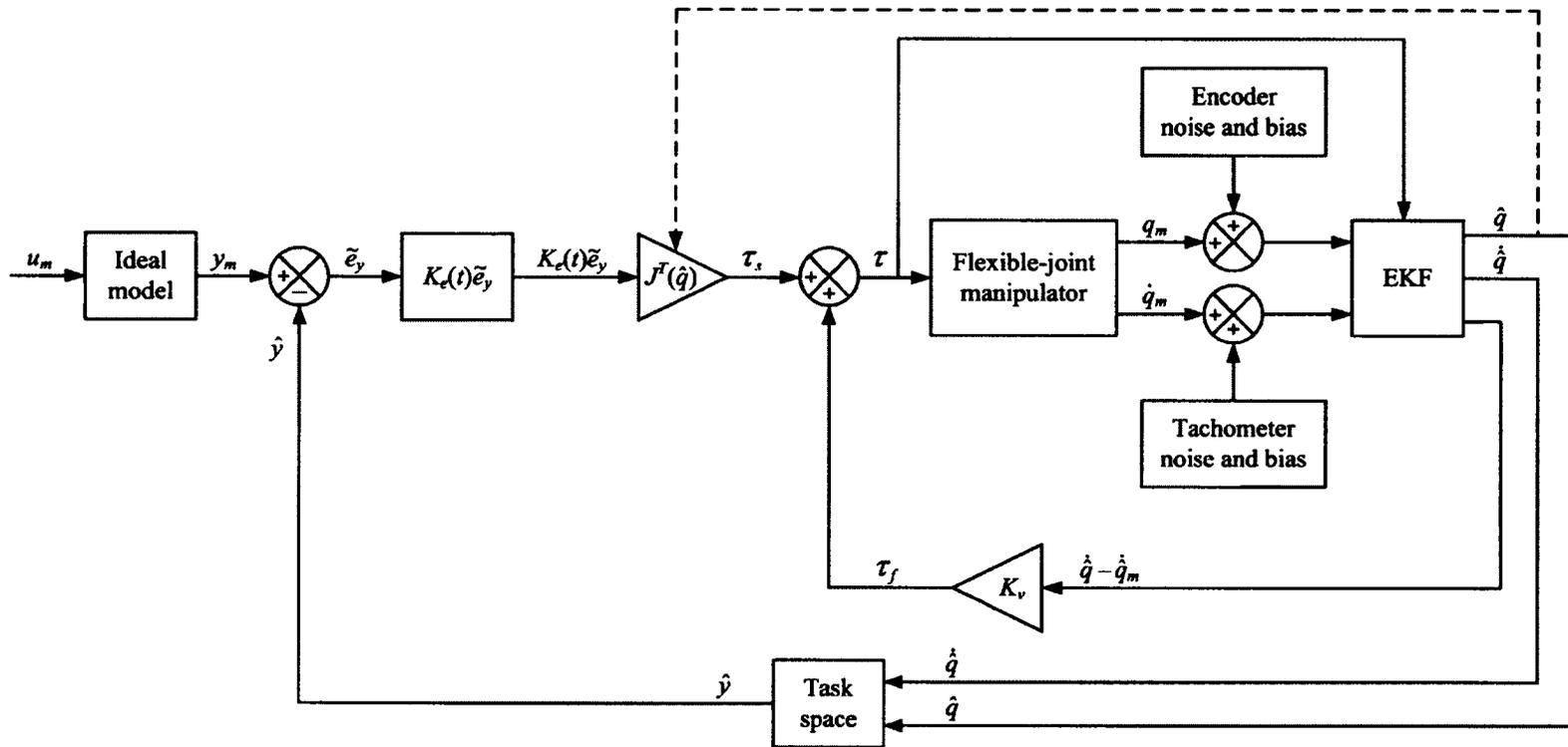


Figure 5.1: Block diagram representation of the adaptive partial state feedback composite control strategy.

In the first scenario, random zero-mean (bias-free) Gaussian noise with standard deviations of 1 deg and 1 deg/s were added to the measurement of each motor position and velocity respectively. The following covariance matrices were selected to optimise the estimation performance in numerical simulations

$$P_0 = 10^{-3}I_8 \quad Q_k = 10^{-6}I_8 \quad R_k = \pi/180I_4$$

Figures 5.4 and 5.7 illustrate the state estimation errors for both estimators. From these results, it is evident that both EKF's provide good estimation accuracy, since the estimation errors remain small despite large measurement noises. Figures 5.2 and 5.5 show the results of tracking end-effector trajectories with both EKF-DMSAC combinations. As shown in Fig. 5.2, the linear joint-based strategy exhibits minimal overshoots at each corner of the trajectory, while the nonlinear joint-based strategy yields slightly degraded performance, as illustrated in Fig. 5.5. Though both EKF's yield similar estimation accuracy, this slight decrease in trajectory tracking performance for the nonlinear joint stiffness dynamics is due to the addition of the highly nonlinear effects and friction torques, which make the trajectory tracking control task more difficult. Finally, it must be noted that when the linear joint-based EKF is applied to the nonlinear joint dynamic model the estimation results diverge, resulting in an unstable trajectory. This demonstrates the benefits of using the nonlinear joint-based EKF in a practical application, where the dynamics of the manipulator is likely to be affected by highly nonlinear effects, such as inertial cross-coupling and friction.

In the second scenario, in addition to the measurement noise, biases of 2 deg and 2 deg/s were added to the motor variable measurements without considering them in the EKF estimator. For consistency, in this scenario, only the nonlinear joint stiffness-based EKF was considered. The tuning of the EKF covariance matrices was not modified. The trajectory tracking results obtained with the bias-free EKF-DMSAC combination are highly aggravated, as the control feedback loop is relying on biased estimated variables. Indeed, with the bias-free EKF, no compensation of unknown measurement biases is performed. This is shown in Fig. 5.8, where the robot end-effector trajectory fails to adequately follow the desired square trajectory.

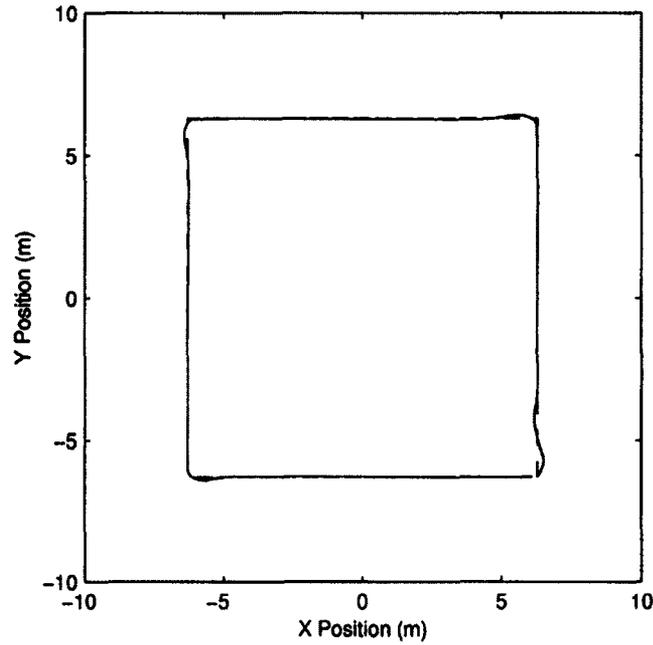


Figure 5.2: Trajectory tracking results obtained with the linear joint stiffness-based EKF / DMSAC composite controller combination applied to the linear joint stiffness manipulator. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$.

Finally, in the third scenario measurement biases were considered in the EKF estimator. Since it is assumed that there is no *a priori* knowledge of the unknown measurement biases, the EKF bias estimator was initialized with $\hat{b}_0 = 0$. The process noise covariance matrix was selected as follows

$$P_{0_{aug}} = 10^{-3}I_{12} \quad Q_{k_{aug}} = \begin{bmatrix} 10^{-6}I_8 & & \\ & 10^{-12}I_2 & \\ & & 10^{-6}I_2 \end{bmatrix}$$

In this case, calibrating the measurement biases in real-time (i.e. by augmenting the state vector with the additional states representing the unknown biases) provides results similar to those obtained with zero-mean Gaussian noise measurements, as shown in Fig. 5.9. Thus, these results give an indication of the increased robustness to measurement biases provided by the bias EKF estimator strategy.

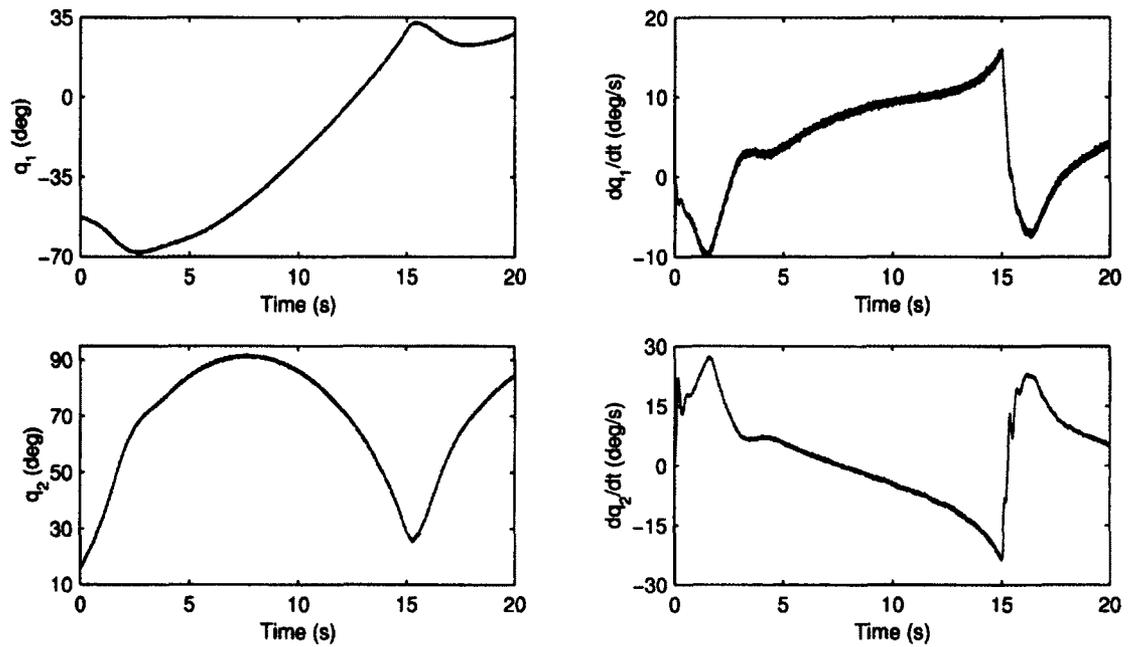


Figure 5.3: Estimation results obtained with the linear joint stiffness-based EKF / DMSAC composite controller combination applied to the linear joint stiffness manipulator. The red lines correspond to the true link variables $q(t), \dot{q}(t)$, and the black lines correspond to the estimated link variables $\hat{q}(t), \hat{\dot{q}}(t)$.

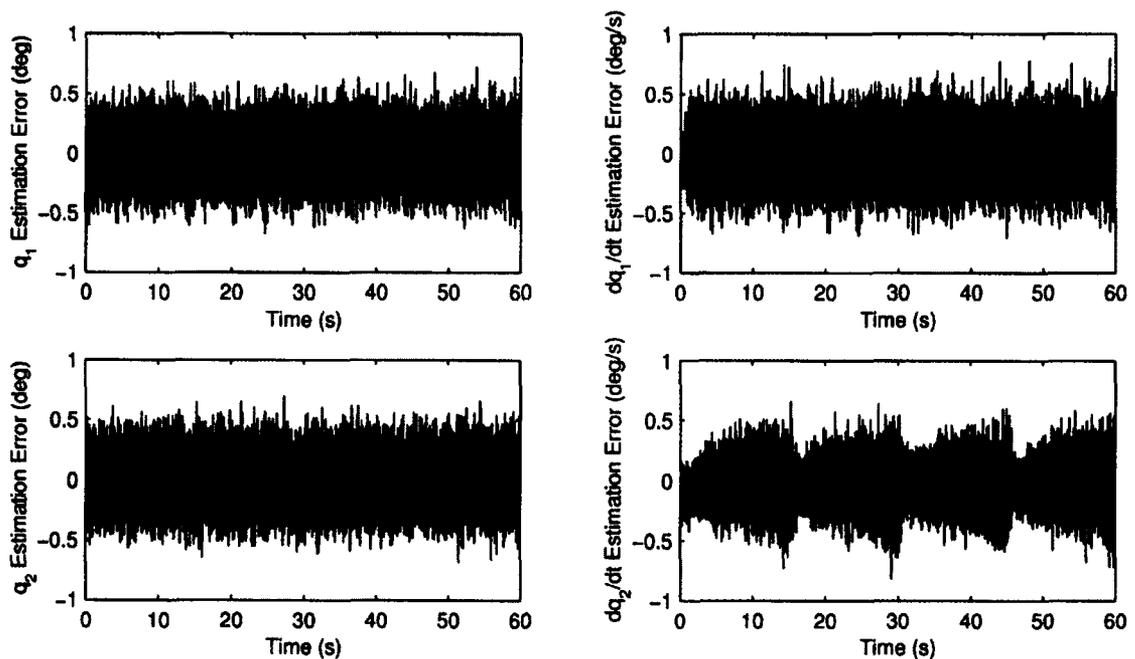


Figure 5.4: Estimation errors obtained with the linear joint stiffness-based EKF / DMSAC composite controller combination applied to the linear joint stiffness manipulator.

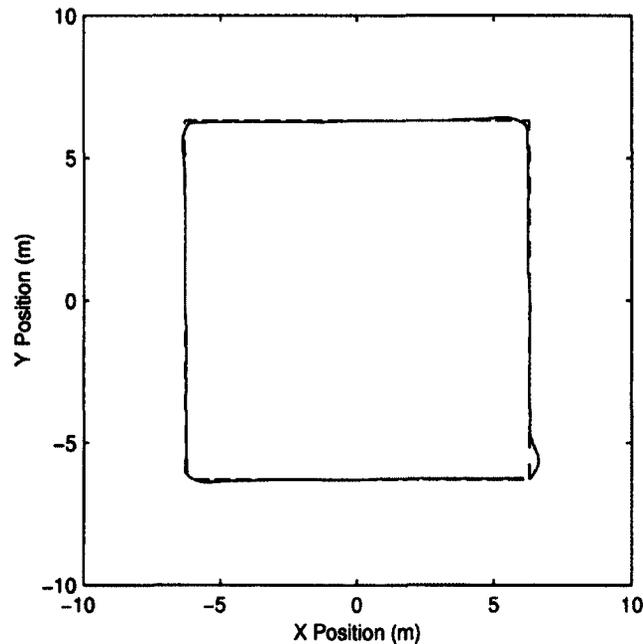


Figure 5.5: Trajectory tracking results obtained with the nonlinear joint stiffness-based EKF / DMSAC composite controller combination applied to the nonlinear joint stiffness manipulator. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$.

5.6 Summary

In this chapter, a partial state feedback composite control scheme that requires only motor position and velocity measurements was proposed. The control scheme combines the DMSAC composite controller with an extended Kalman filter (EKF) estimator developed for a flexible-joint space manipulator, modeled with both the well-established linear joint stiffness dynamics model and the novel nonlinear joint stiffness dynamics model proposed in this work. Numerical simulation results were presented to demonstrate the effectiveness of the proposed new partial state feedback control schemes. In addition, by augmenting the EKF state vector with the unknown biases, a simple strategy to calibrate the measurement biases in real-time was provided.

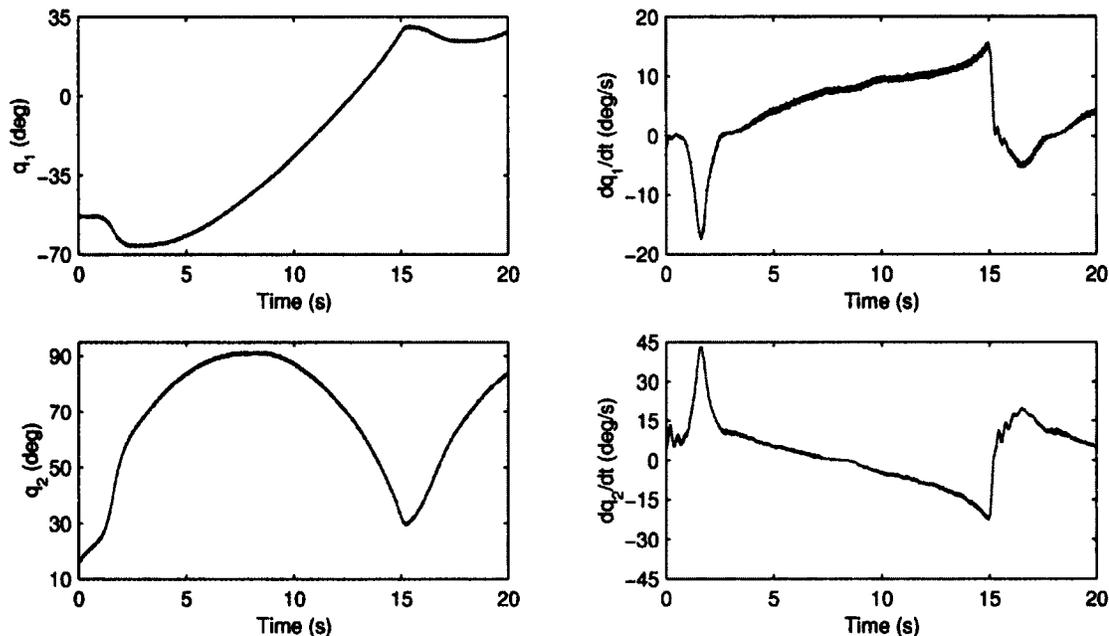


Figure 5.6: Estimation results obtained with the nonlinear joint stiffness-based EKF / DMSAC composite controller combination applied to the nonlinear joint stiffness manipulator. The red lines correspond to the true link variables $q(t), \dot{q}(t)$, and the black lines correspond to the estimated link variables $\hat{q}(t), \hat{\dot{q}}(t)$.

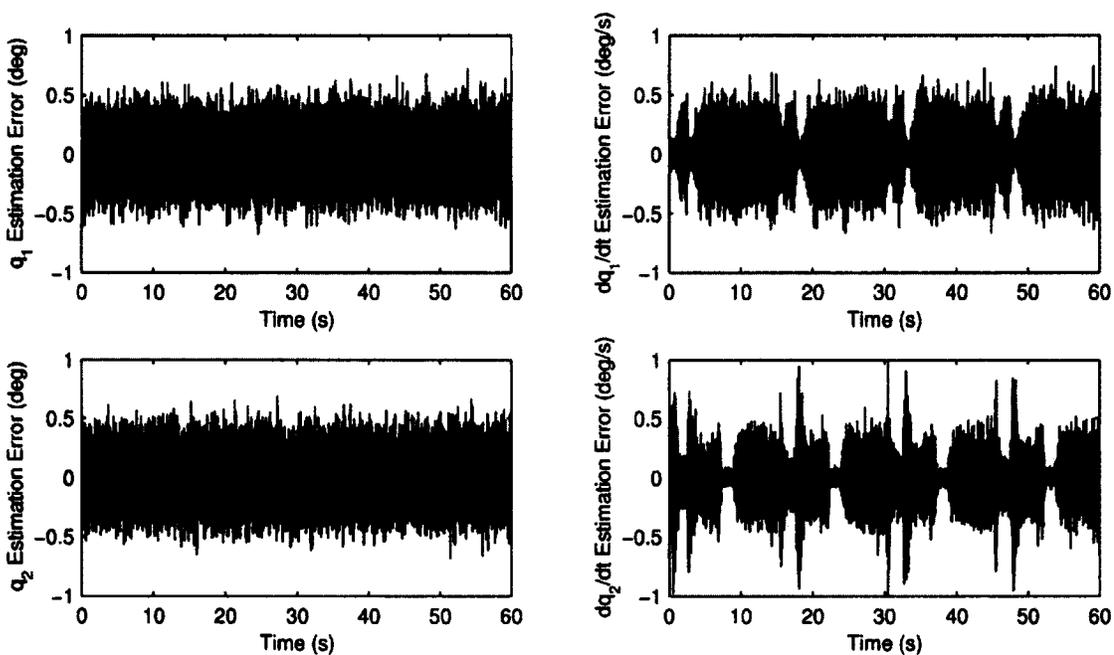


Figure 5.7: Estimation errors obtained with the nonlinear joint stiffness-based EKF / DMSAC composite controller combination applied to the nonlinear joint stiffness manipulator.

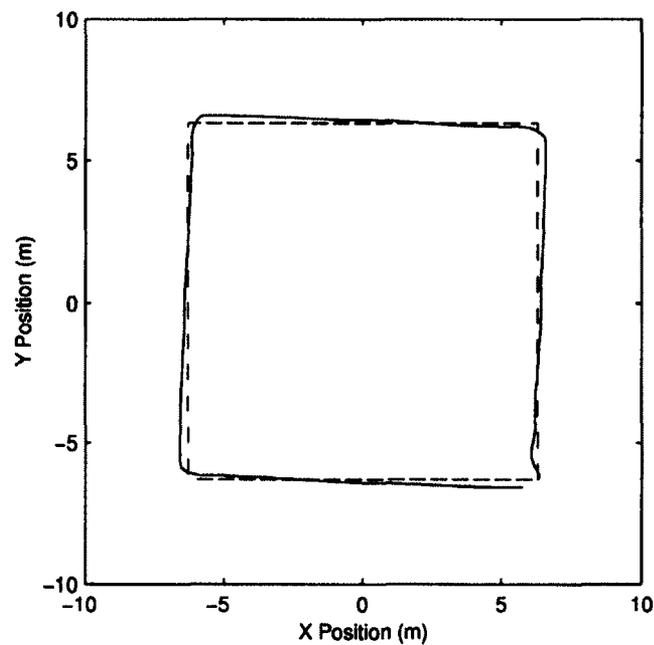


Figure 5.8: Trajectory tracking results obtained with the nonlinear joint stiffness-based EKF / DMSAC composite controller combination applied to the nonlinear joint stiffness manipulator. The measurements are corrupted by biases of $b_e = 2$ deg and $b_t = 2$ deg/s, but without considering them in the EKF estimator. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$.

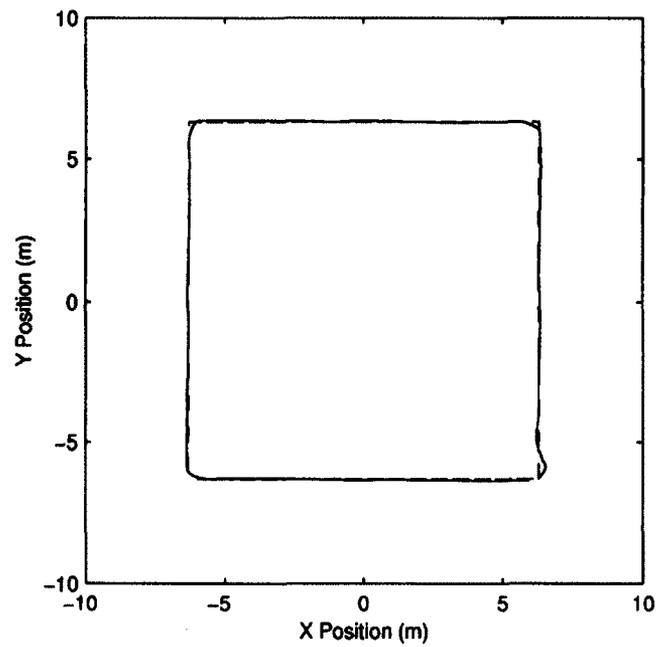


Figure 5.9: Trajectory tracking results obtained with the nonlinear joint stiffness-based EKF / DMSAC composite controller combination applied to the nonlinear joint stiffness manipulator. The measurements are corrupted by biases of $b_e = 2$ deg and $b_t = 2$ deg/s, which are considered in the EKF estimator. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$.

Chapter 6

Conclusion

In this chapter, the general objective of this research project is recalled and the significant contributions this work made to the field of rigid-joint and flexible-joint direct adaptive control techniques are outlined. A list of publications in which these contributions are documented is provided. Finally, a discussion on suggestions for future research directions is presented.

6.1 Thesis Summary

The literature review on the control of flexible-joint manipulators identified that most existing control strategies are model-based techniques, and thus achieve reasonably good tracking performance only when substantial knowledge of the plant mathematical model and its parameters is available. Consequently, existing control approaches could perform inadequately under adverse conditions, such as plant uncertainties or modeling errors. The objective of this work was to develop model-independent, task-space direct adaptive control methodologies, in which controller parameters adapt in real-time without requiring identification of unknown plant parameters or mathematical models of the system to be controlled.

This work first presented theoretical development of a new rigid-joint direct adaptive fuzzy controller, in which the control gains of a transpose Jacobian control law were adapted using a fuzzy logic system in response to tracking errors between a reference model and the actual robot system outputs. Numerical simulations in the context of 12.6 m \times 12.6 m square trajectory tracking by the end-effector of a rigid-joint manipulator modeled with the Euler-Lagrange dynamics formulation compared the performance of the fuzzy-logic-based adaptive controller with a nonadaptive transpose Jacobian controller. Results established that the fuzzy adaptive controller

yields improved trajectory tracking performance when compared with the nonadaptive controller, particularly at each corner of the trajectory where the nonlinearities in the system are significant.

Based on a recent theoretical breakthrough in defining the almost strictly passive conditions for nonlinear and nonstationary systems, this work presented the theoretical development of two novel direct adaptive control schemes applicable to nonlinear systems with adaptation mechanisms derived from the simple adaptive control (SAC) theory: (1) the *decentralized simple adaptive control* (DSAC) methodology, and (2) the *decentralized modified simple adaptive control* (DMSAC) methodology. Similar to the direct adaptive fuzzy control law, these two decentralized control techniques only consider the diagonal of the time-varying gain matrices, thereby increasing the computational efficiency in comparison to centralized algorithms. The stability of both the DSAC and DMSAC control methodologies was rigorously established in the sense of Lyapunov for any nonlinear square systems satisfying the recently-developed almost strictly passive conditions. The two new approaches were first applied in numerical simulations for a rigid-joint manipulator modeled as a square system, then the DSAC/DMSAC control algorithms were applied to the nonsquare rigid-joint manipulator. Although the stability cannot be established in the latter case, simulation results demonstrated the applicability of the adaptive techniques for nonsquare systems. In both cases, the results obtained when tracking the square trajectory showed improved tracking performance for the DSAC approach compared with the DMSAC algorithm, though at the expense of greater complexity in the controller structure.

The fuzzy logic-based and SAC-based direct adaptive control techniques for rigid-joint manipulators were then extended to flexible-joint manipulators by applying the singular perturbation-based theory. Using this theory, the control input torque actuating the robot motors was separated into a slow control term designed to stabilize the rigid-joint dynamics, and a fast control term designed to improve damping of the elastic vibrations at the joints. Invoking Tychonov's theorem rigorously established the analytical stability of the resulting composite controllers. Extensive numerical simulations testing the trajectory tracking performance of the developed adaptive control methodologies were performed on a flexible-joint robot manipulator following

the square trajectory. Despite abrupt changes in direction that caused vibrations in the joints, the proposed adaptive control schemes all exhibited good trajectory tracking performance, with minimal overshoots when applied to the nominal linear joint stiffness dynamics model ($k = 500I_2$ N·m). To validate the robustness to parametric uncertainties, the composite controllers tuned for the nominal manipulator were applied to a robot with significantly lower joint stiffness ($k = 200I_2$ N·m). The adaptive control gains responded accordingly to the higher flexibilities and provided improved damping of the elastic vibrations, while minimizing the tracking errors, compared to a nonadaptive composite control strategy. These results clearly demonstrated that the proposed adaptive composite controllers were more robust to large uncertainties in the systems. Furthermore, the same composite controllers tuned for the nominal manipulator were applied to a manipulator modeled with the nonlinear joint stiffness formulation derived in this work. This new dynamics formulation includes nonlinear effects that were observed in experimental studies, but were disregarded in most existing flexible-joint dynamics models. The effects considered in this nonlinear joint stiffness model are: nonlinear and nonstationary joint stiffness matrix, joint friction, soft-windup effect and dynamics coupling between joint and motor accelerations. The motivation for testing the adaptive controllers with this new dynamics formulation was that complex and unpredictable flexible dynamics behaviors can arise, and the controllers must be robust enough to handle dynamics effects that are not included in the basic model used for tuning the controllers. Simulation results demonstrated significantly improved trajectory tracking when applied to the nonlinear joint stiffness model, and robustness to modeling errors when compared to the nonadaptive control strategy.

Finally, this work addressed the partial state feedback problem associated with the limited number of measurements provided by noisy and biased sensors. Most existing flexible-joint control techniques, including all direct adaptive control methodologies developed in this work, require full state feedback. However, since typical robot manipulators are only equipped with motor position and velocity sensors, the remaining state variables (i.e. link positions and velocities) had to be estimated. To address this issue, two extended Kalman filters were developed to estimate all state variables

in real-time; one for the classic linear joint stiffness model and one for the newly-developed nonlinear joint stiffness model. Observability analyses were carried out for both estimation schemes, and numerical simulations demonstrated good estimation and control results when the state estimates were used by the DMSAC composite controller to track the square trajectory. In addition, it was shown that both estimation strategies could be successfully extended to cases with measurements that were also corrupted by biases, as demonstrated by the simulation results.

6.2 Significance of Work

The significance of this research work is evidenced by the publications and presentations in control engineering-related journals and conferences, as listed below:

Journal Articles

Ulrich, S., and Sasiadek, J. Z., "Direct Fuzzy Adaptive Control of a Manipulator with Elastic Joints," *Journal of Guidance, Control, and Dynamics*, accepted for publication, to appear. [181]

Ulrich, S., Sasiadek, J. Z., and Barkana, I., "Modeling and Direct Adaptive Control of a Flexible-Joint Manipulator," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 1, 2012, pp. 25-39. [182]

Ulrich, S. and Sasiadek, J. Z., "Trajectory Tracking Control of Flexible-Joint Space Manipulators," *Canadian Aeronautics and Space Journal*, Vol. 58, No. 1, 2012, pp. 47-59. [183]

Book Chapter

Ulrich, S. and Sasiadek, J. Z., "Real-Time Estimation and Adaptive Control of Flexible Joint Space Manipulators," *Robot Motion and Control*, edited by K. Kozlowski, Vol. 442 of *Series on Lecture Notes in Control and Information Sciences*, Springer-Verlag, 2012, ISBN 978-1-4471-2342-2, pp.215-224. [184]

Conference Proceedings

Ulrich, S., Sasiadek, J. Z., and Barkana, I. On a New Class of Direct Adaptive Output Feedback Controllers for Nonlinear Square Systems. *51st IEEE Conference on Decision and Control*, Maui, Hawaii, 10 - 13 December, 2012, accepted for presentation. [185]

Ulrich, S., and Sasiadek, J. Z.. Composite Simple Adaptive Control for Flexible-Joint Space Manipulators with Uncertainties and Modeling Errors. *ASME Dynamic Systems and Control Conference*, Fort Lauderdale, Florida, 17 - 19 October, 2012, accepted for presentation. [186]

Ulrich, S. and Sasiadek, J. Z., "Direct Fuzzy Adaptive Control of a Manipulator with Elastic Joints," *AIAA Guidance, Navigation and Control Conference*, Minneapolis, MN, 13-16 August, 2012. [187]

Ulrich, S., Sasiadek, J. Z., and Barkana, I. Decentralized Simple Adaptive Control for Nonsquare Euler-Lagrange Systems. *American Control Conference*, Montreal, Quebec, 27 - 29 June, 2012. [188]

Ulrich, S. and Sasiadek, J. Z., "Methods of Trajectory Tracking for Flexible Joint Space Manipulators," *18th IFAC World Congress*, Milan, Italy, 28 August - 2 September, 2011, pp. 10307-10312. [189]

Ulrich, S. and Sasiadek, J. Z., "Direct Adaptive Fuzzy Control for a Two-Link Space Robot," *16th IEEE/IFAC International Conference on Methods and Models in Automation and Robotics*, Miedzydroje, Poland, 22-25 August, 2011, pp. 290-295 [190]

Ulrich, S. and Sasiadek, J. Z., "Control Strategies for Flexible Joint Manipulators," *AIAA Guidance, Navigation, and Control Conference*, Portland, OR, 8-11 August, 2011. [191]

Ulrich, S. and Sasiadek, J. Z., "Extended Kalman Filtering for Flexible Joint Space Robot Control," *American Control Conference*, San Francisco, CA, 29 June-1 July, 2011, pp. 1021-1026. [192]

Ulrich, S. and Sasiadek, J. Z., "Real-Time Estimation and Adaptive Control of Flexible Joint Space Manipulators," *8th International Workshop on Robot Motion and Control*, Bukowy Dworek, Poland, 15-17 June, 2011. [193]

Ulrich, S. and Sasiadek, J. Z., "Direct Model Reference Adaptive Control of a Flexible Joint Robot," *AIAA Guidance, Navigation and Control Conference and Exhibit*, Toronto, Canada, 2-5 August, 2010. [194]

Ulrich, S., and Sasiadek, J. Z. Modified Simple Adaptive Control for a Two-Link Space Robot, *American Control Conference*, Baltimore, Maryland, 30 June-2 July, 2010, pp. 3654-3659. [195]

Ulrich, S. and Sasiadek, J. Z., "Control of Rigid and Flexible Joint Space Manipulators," *15th CASI Conference on Astronautics*, Toronto, Canada, 4-6 May, 2010. [196]

Ulrich, S. and Sasiadek, J. Z., "Autonomous Control for Flexible Joint Space Robotic Manipulators," *60th International Astronautical Congress*, Daejeon, Korea 12-16 October, 2009. [197]

6.3 Recommendations for Future Work

The main contribution of this work is the pioneer theoretical study on the development and evaluation of two new SAC-based control methodologies for both rigid-joint and flexible-joint space manipulators: the *decentralized simple adaptive control* (DSAC) and the *decentralized modified simple adaptive control* (DMSAC). As such, this work can be viewed as establishing the foundation for more detailed theoretical studies to develop better understanding of this new class of direct adaptive control methodologies. Additionally, many other novel contributions related to the control of rigid-joint and flexible-joint space robot manipulators have been made in this work, including fuzzy logic-based adaptive control, dynamics modeling and nonlinear state estimation, and these should lead to further studies in these areas. Some suggestions are provided herein to guide future efforts.

Almost Strictly Passive Conditions for Nonsquare Systems: Although the applicability of the DSAC and DMSAC control methodologies developed in this work to nonsquare systems was demonstrated through extensive numerical simulations, the extension of the almost strictly passive conditions recently developed by Barkana [139] would be required to guarantee stability of this new class of adaptive controllers to such systems. However, those conditions do not yet exist, even for the linear time-invariant (LTI) case, although inroads to solving this problem for LTI systems have been made by Fradkov [198].

Exponentially Stable SAC-Based Quasi-Steady State Subsystem: As noted in Chapter 4, Tychonov's theorem to guarantee the stability of adaptive composite controllers holds only for a finite time interval. Though there is an extension of this theorem to the infinite time interval (see [144]) which could be applied in an attempt to show asymptotic stability for sufficiently small ϵ , this would require exponential stability of the quasi-steady state subsystem. Future efforts could focus on modifying the SAC-based adaptive controllers to yield an exponentially stable quasi-steady state closed-loop subsystem for flexible-joint manipulators.

Fuzzy Logic Type-2 Control: The fuzzy logic system designed to stabilize the quasi steady-state in Chapter 4 was based on the type-1 fuzzy logic control theory, a control methodology known for its ability to compensate for structured and unstructured uncertainties. However, a new type-2 fuzzy logic control theory to specifically yield improved performance under larger degrees of uncertainties has been developed by Mendel [199], Mendel and John [200], and Hagrass [201]. This theory is also suitable for nonstationary systems subject to modeling errors, such as robot manipulators. One of the first attempts to design and implement a type-2 fuzzy logic control architecture for flexible-joint manipulator systems took place in 2008, and can be found in [202] where the authors proposed a type-2 fuzzy controller strategy based on a sliding mode control approach. Their controller is also compared to its type-1 counterpart, and numerical simulations illustrated the superiority of type-2 controller in compensating for large uncertainties. More recently, in 2011, Biglarbegian *et al.* [203] presented an interval type-2 Takagi-Sugeno-Kang fuzzy logic controller for a rigid-joint manipulator with parametric uncertainties. Trajectory tracking results show

that for some configurations, the fuzzy controller can outperform well-known linear and nonlinear controllers. Based on these two studies, development of a type-2 fuzzy logic-based adaptive composite controller for flexible-joint manipulators should be pursued.

Applications to Free-Floating and Free-Flying Space Manipulators: As in previous studies on the control of flexible space manipulators [8–12,14,15,99,111], the new controllers developed in this work were applied to a space manipulator attached to a full controlling base, with both its position and orientation controlled through jet thrusts and reaction wheels to compensate for the manipulator motions. However, a particular problem with space robots is how to handle the dynamics coupling between the robot manipulator and its base. In reality, the position and orientation of the base spacecraft fluctuates due to the manipulator motion, and vice versa. Space robots attached to a moving base can be classified into two categories [204]: (1) free-floating, where neither the position nor the orientation of the base is controlled to save energy, and (2) free-flying, where only the orientation of the base is controlled through reaction wheels to maintain a communication link with the ground and to generate electrical power from the solar panels. In either case, the issue of coordination or isolation between the base and a flexible-joint manipulator dynamics should be investigated.

Unscented Kalman Filter for State Estimation: The state estimators used to determine the full state vector in real-time were derived using the extended Kalman filter (EKF) theory. As discussed in Chapter 5, the EKF propagates the estimated states through an exact, nonlinear dynamics model, while the state covariance matrix is propagated through a linearized approximation. This linear approximation could lead to a variety of complications and limitations [205]. For example, the linearization is prohibitive in terms of computation, since it requires determining the exponential of a matrix. Another example, is that since the Kalman theory guarantees the stability and performance of the filter only for linear systems driven by ideal Gaussian noise processes, the inherent nonlinearities of nonlinear systems imply the need for intensive simulation validations, to fine-tune the filter parameters to the actual system. As a consequence, the filter behaves adequately only for the range of conditions it was

tuned for, which leads to poor robustness to non-nominal conditions and parameter variations. Thus, stability of the EKF is only ensured for small deviations from the reference conditions. This was observed in this work through the unstable behavior (divergence) in numerical simulations when the linear joint stiffness-based EKF was applied to the nonlinear joint stiffness dynamics model. A new class of Kalman-based state estimation techniques, known as the unscented Kalman filter (UKF), was developed by Julier and Uhlmann [206]. This technique uses the so-called unscented transform (UT) derived from the field of statistics for handling random variables transformed by nonlinear functions. The UT allows the optimal computation of the exact first and second-order statistical moments (i.e. mean and covariance) of a nonlinear function of a random variable, making it far more accurate than conventional linearization. The UKF merges the UT within the Kalman filter to avoid issues related to linearization. The estimated state variables, with a few well-chosen neighbouring points, known as sigma points, are propagated through the exact nonlinear dynamics. A specific weighting function of the computed results provides the mean and the state covariance prior to the measurement update. In view of the above, future efforts could focus on the development of state estimation techniques, using the UKF theory for flexible-joint manipulators.

Appendix A

Rigid Kinematics and Dynamics Equations

This appendix presents the detailed development of the kinematics and dynamics equations of a two-link robot manipulator.

A.1 Kinematics Equations

In this section, the forward and inverse kinematics equations are presented.

A.1.1 Forward Kinematics

The forward kinematics problem is concerned with the determination of the end-effector position and orientation as a function of the individual joint angles. To perform the kinematic analysis, a reference frame $o_i x_i y_i z_i$ is attached to link i . For the two-link planar robot with two revolute joints illustrated in Fig. A.1, the frame $o_0 x_0 y_0 z_0$ is referred to as the inertial frame and is attached to the base, the frame $o_1 x_1 y_1 z_1$ is attached to the first link, at the elbow joint, and the frame $o_2 x_2 y_2 z_2$ is attached to the second link, at the end-effector. The position and orientation of $o_i x_i y_i z_i$ with respect to $o_{i-1} x_{i-1} y_{i-1} z_{i-1}$ can be obtained with the time-varying transformation matrix $A_i \in \mathbb{R}^{4 \times 4}$, which varies as the configuration of the robot is changed. Assuming that all joints are either revolute or prismatic, the matrix A_i is obtained as a function of the joint variable $q_i \in \mathbb{R}$, as follows [18,107–109]

$$A_i \equiv A_i(q_i) \tag{A.1}$$

where the transformation A_i , is determined using the Denavit-Hartenberg, or DH convention. With this convention, A_i is given by a product of four basic transformations

$$\begin{aligned}
A_i &= \text{ROT}_{z,q_i} \text{TRANS}_{z,d_i} \text{TRANS}_{x,l_i} \text{ROT}_{x,\alpha_i} \\
&= \begin{bmatrix} \cos q_i & -\sin q_i & 0 & 0 \\ \sin q_i & \cos q_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & l_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \cos q_i & -\sin q_i \cos \alpha_i & \sin q_i \sin \alpha_i & l_i \cos q_i \\ \sin q_i & \cos q_i \cos \alpha_i & -\cos q_i \sin \alpha_i & l_i \sin q_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.2}
\end{aligned}$$

where $\forall i = 1, 2$, the parameters $l_i, \alpha_i, d_i, q_i \in \mathbb{R}$ denote the length of link i , twist angle of link i , offset distance of link i , and angular position of link i , respectively. Neglecting the twist and offset for the two-link planar robot manipulator shown in Fig. A.1, the only remaining parameters of interest are l_i and q_i . From A.2, the A matrices are then given by

$$A_1 = \begin{bmatrix} \cos q_1 & -\sin q_1 & 0 & l_1 \cos q_1 \\ \sin q_1 & \cos q_1 & 0 & l_1 \sin q_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.3}$$

$$A_2 = \begin{bmatrix} \cos q_2 & -\sin q_2 & 0 & l_2 \cos q_2 \\ \sin q_2 & \cos q_2 & 0 & l_2 \sin q_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.4}$$

The position and orientation of $o_i x_i y_i z_i$ with respect to $o_j x_j y_j z_j$ is then obtained with the homogeneous transformation matrix $C_{ij} \in \mathbb{R}^{4 \times 4}$ given by

$$C_{ij} = \begin{cases} A_{i+1}A_{i+2} \dots A_{j-1}A_j & \text{if } i < j \\ I_4 & \text{if } i = j \\ C_{ji}^T & \text{if } i > j \end{cases} \quad (\text{A.5})$$

Therefore, using (A.3) and (A.4), the rotation matrices are given by

$$C_{10} = A_1 \quad (\text{A.6})$$

$$C_{20} = A_1A_2 = \begin{bmatrix} \cos(q_1 + q_2) & -\sin(q_1 + q_2) & 0 & l_1 \cos q_1 + l_2 \cos(q_1 + q_2) \\ \sin(q_1 + q_2) & \cos(q_1 + q_2) & 0 & l_1 \sin q_1 + l_2 \sin(q_1 + q_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.7})$$

The first two entries of the last column of C_{20} represent the x and y components of the reference frame $o_2x_2y_2z_2$ attached to the end-effector with respect in the base frame, which are given by

$$x_2 = l_1 \cos q_1 + l_2 \cos(q_1 + q_2) \quad (\text{A.8})$$

$$y_2 = l_1 \sin q_1 + l_2 \sin(q_1 + q_2) \quad (\text{A.9})$$

Equations (A.8) and (A.9) are called the forward kinematics equations.

A.1.2 Velocity Kinematics - The Jacobian

The velocity kinematics can be conveniently formulated in differential form by deriving a suitable Jacobian that relates the linear and angular velocity of the end-effector, denoted by $v_2, \omega_2 \in \mathbb{R}^2$, respectively, to the joint velocity $\dot{q} \in \mathbb{R}^2$, as follows [18]

$$v_2 = J_v(q)\dot{q} \quad (\text{A.10})$$

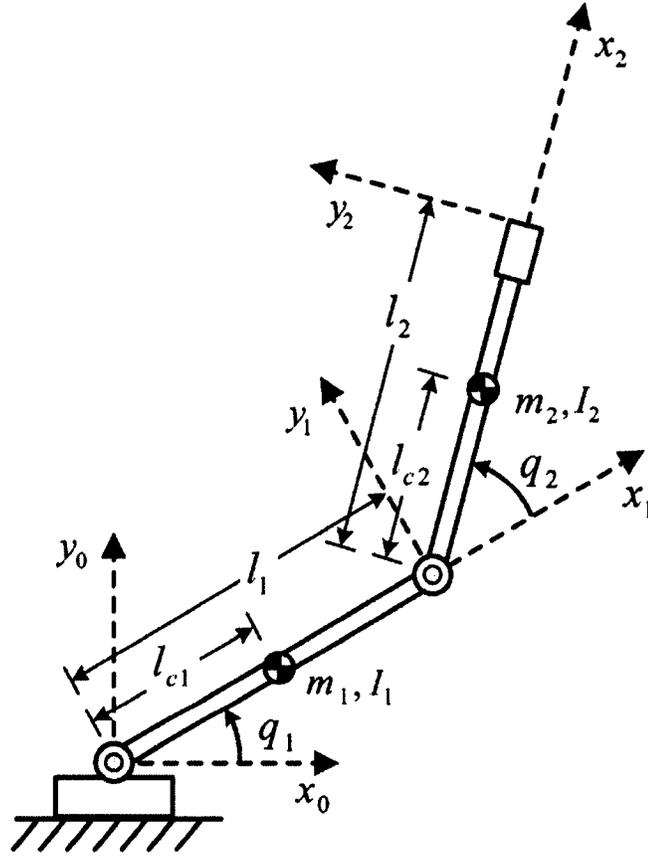


Figure A.1: Two-link revolute joint robot manipulator

$$\omega_2 = J_\omega(q)\dot{q} \quad (\text{A.11})$$

where $J_v(q), J_\omega(q) \in \mathbb{R}^{3 \times 2}$. Equations (A.10) and (A.11) can be grouped together as

$$\begin{bmatrix} v_{2x} \\ v_{2y} \\ \omega_{2z} \\ \omega_{2x} \\ \omega_{2y} \\ \omega_{2z} \end{bmatrix} = J(q) \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \quad (\text{A.12})$$

In the above expression, $J(q) \in \mathbb{R}^{6 \times 2}$ denotes the Jacobian matrix, which is given by

$$J(q) = \begin{bmatrix} -l_1 \sin q_1 - l_2 \sin(q_1 + q_2) & -l_2 \sin(q_1 + q_2) \\ l_1 \cos q_1 + l_2 \cos(q_1 + q_2) & l_2 \cos(q_1 + q_2) \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \quad (\text{A.13})$$

The first two rows of (A.13) gives the linear velocity of the end-effector relative to the base, which is obtained by time-differentiating (A.8) and (A.9), as follows

$$\dot{x}_2 = -l_1 \sin q_1 \dot{q}_1 - l_2 \sin(q_1 + q_2)(\dot{q}_1 + \dot{q}_2) \quad (\text{A.14})$$

$$\dot{y}_2 = l_1 \cos q_1 \dot{q}_1 + l_2 \cos(q_1 + q_2)(\dot{q}_1 + \dot{q}_2) \quad (\text{A.15})$$

The third row of the Jacobian is the linear velocity of the end-effector in the direction of z_0 , which is indeed zero for a planar two-link manipulator. The last three rows of the Jacobian represent the angular velocity of the end-effector, which is simply a rotation about the z_0 axis at a rate of $\dot{q}_1 + \dot{q}_2$. In the present case, the orientation of the end-effector is of no concern and the Jacobian matrix, denoted by $J(q) \in \mathbb{R}^{2 \times 2}$, reduces to the first two rows.

A.1.3 Inverse Kinematics

Inverse kinematics consists of the determination of the joint angles given the coordinates of the end-effector frame $o_2x_2y_2z_2$. Using a simple geometrical approach and the law of cosines it can be shown that for the two-link planar robot shown in Fig. A.1, the joint angle q_2 is given by

$$\cos q_2 = \frac{x_2^2 + y_2^2 - l_1^2 - l_2^2}{2l_1l_2} = D \quad (\text{A.16})$$

However, a better way to determine q_2 is based on the fact that since $\cos q_2 = D$, then

$$\sin q_2 = \pm\sqrt{1 - D^2} \quad (\text{A.17})$$

Hence, q_2 can be found by

$$q_2 = \tan^{-1} \frac{\pm\sqrt{1 - D^2}}{D} \quad (\text{A.18})$$

where the negative and positive sign in (A.18) represent the elbow-up and elbow-down configuration, respectively. The angle q_1 is given as a function of q_2 , as follows

$$q_1 = \tan^{-1} \left(\frac{y_2}{x_2} \right) - \tan^{-1} \left(\frac{l_2 \sin q_2}{l_1 + l_2 \cos q_2} \right) \quad (\text{A.19})$$

The determination of the joint velocities from the end-effector velocities is performed by using the inverse Jacobian

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = J^{-1}(q) \begin{bmatrix} \dot{x}_2 \\ \dot{y}_2 \end{bmatrix} \quad (\text{A.20})$$

where the inverse of the Jacobian is given by

$$J^{-1}(q) = \frac{1}{l_1 l_2 \sin q_2} \begin{bmatrix} l_2 \cos(q_1 + q_2) & l_2 \sin(q_1 + q_2) \\ -l_1 \cos q_1 - l_2 \cos(q_1 + q_2) & -l_1 \sin q_1 - l_2 \sin(q_1 + q_2) \end{bmatrix} \quad (\text{A.21})$$

The determinant of the Jacobian is equal to $l_1 l_2 \sin q_2$. Hence, the Jacobian does not have an inverse when $q_2 = 0$ or $q_2 = \pi$, in which case the manipulator is said to be in a singular configuration.

A.2 Dynamics Equations

For the two-link planar robot manipulator with two revolute joints shown in Fig. A.1 where, $\forall i = 1, 2$, $m_i \in \mathbb{R}$ denotes the mass of link i , $l_i \in \mathbb{R}$ denotes the length of link i , $l_{ci} \in \mathbb{R}$ denotes the distance from the previous joint to the center of mass of link i , and $I_i \in \mathbb{R}$ denotes the moment of inertia of link i about an axis perpendicular to the xy plane passing through the center of gravity of link i , the dynamics equations of motion

are derived with the Euler-Lagrange formulation which is defined as a function of the kinetic $T_r \in \mathbb{R}$, and potential energy $U_r \in \mathbb{R}$ of the system as follows [18, 107–110]

$$L = T_r - U_r \quad (\text{A.22})$$

The kinetic and potential energies are defined in terms of a set of so-called generalized coordinates (q_1, \dots, q_n) (which also denote the joint angles) where n is the number of DOF of the system. The equations of motions of a two-link rigid-joint robot subjected to torques $\tau \in \mathbb{R}^2$ is derived according to the following equation

$$\tau_i = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i}; \quad i = 1, 2 \quad (\text{A.23})$$

The kinetic energy is assumed to be a quadratic function of the link position vector q of the form

$$T_r = \frac{1}{2} \sum_{i=0}^n \sum_{j=0}^n M_{ij} \dot{q}_i \dot{q}_j; \quad i = 1, 2 \quad j = 1, 2 \quad n = 2 \quad (\text{A.24})$$

where M_{ij} are the components of the robot inertia matrix denoted $M(q) \in \mathbb{R}^{2 \times 2}$ which is symmetric and positive definite. In the case of rigid dynamics, the only source of potential energy is gravity. The potential energy of the i th link is computed by assuming that the mass of the entire link is concentrated at its center of gravity. Assuming that the potential energy is independent of \dot{q} , the total potential energy of the two-link robot is therefore

$$U_r(q) = \sum_{i=0}^n m_i g^T r_{ci}; \quad i = 1, 2 \quad n = 2 \quad (\text{A.25})$$

where $g \in \mathbb{R}^2$ is the vector giving the direction of the gravity in the inertial frame and the vector $r_{ci} \in \mathbb{R}$ gives the coordinates of the center of gravity of the i th link.

The Lagrangian is given by

$$L = \frac{1}{2} \sum_{i=0}^n \sum_{j=0}^n M_{ij} \dot{q}_i \dot{q}_j - \sum_{i=0}^n m_i g^T r_{ci} \quad (\text{A.26})$$

For space robot applications, the gravitational force g is disregarded, and the Lagrangian is reduced to

$$L = \frac{1}{2} \sum_{i=0}^n \sum_{j=0}^n M_{ij} \dot{q}_i \dot{q}_j \quad (\text{A.27})$$

The partial derivative of the Lagrangian with respect to the k th joint position is given by

$$\frac{\partial L}{\partial q_k} = \frac{1}{2} \sum_{i=0}^n \sum_{j=0}^n \frac{\partial M_{ij}}{\partial q_k} \dot{q}_i \dot{q}_j \quad (\text{A.28})$$

Similarly, the partial derivatives of the Lagrangian with respect to the k th joint velocity is given by

$$\frac{\partial L}{\partial \dot{q}_k} = \sum_{j=0}^n M_{kj} \dot{q}_j \quad (\text{A.29})$$

and therefore

$$\begin{aligned} \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_k} &= \sum_{j=0}^n M_{ij} \ddot{q}_j + \sum_{j=0}^n \frac{d}{dt} M_{kj} \dot{q}_j \\ &= \sum_{j=0}^n M_{ij} \ddot{q}_j + \sum_{i=0}^n \sum_{j=0}^n \frac{\partial M_{kj}}{\partial q_i} \dot{q}_i \dot{q}_j \end{aligned} \quad (\text{A.30})$$

Thus, for each joint, the Euler-Lagrange equations are given by

$$\tau_k = \sum_{j=0}^n M_{ij} \ddot{q}_j + \sum_{i=0}^n \sum_{j=0}^n \left\{ \frac{\partial M_{kj}}{\partial q_i} - \frac{1}{2} \frac{\partial M_{ij}}{\partial q_k} \right\} \dot{q}_i \dot{q}_j \quad (\text{A.31})$$

It can be shown that

$$\begin{aligned} \sum_{i=0}^n \sum_{j=0}^n \left\{ \frac{\partial M_{kj}}{\partial q_i} - \frac{1}{2} \frac{\partial M_{ij}}{\partial q_k} \right\} \dot{q}_i \dot{q}_j &= \sum_{i=0}^n \sum_{j=0}^n \frac{1}{2} \left\{ \frac{\partial M_{kj}}{\partial q_i} + \frac{\partial M_{ki}}{\partial q_j} - \frac{\partial M_{ij}}{\partial q_k} \right\} \dot{q}_i \dot{q}_j \\ &= \sum_{i=0}^n \sum_{j=0}^n c_{ijk} \dot{q}_i \dot{q}_j \end{aligned} \quad (\text{A.32})$$

where the terms c_{ijk} are known as the Christoffel symbols. The Euler-Lagrange equations of motion can then be written as

$$\tau_k = \sum_{j=0}^n M_{ij} \ddot{q}_j + \sum_{i=0}^n \sum_{j=0}^n c_{ijk} \dot{q}_i \dot{q}_j; \quad k = 1, \dots, n \quad (\text{A.33})$$

In the above expression, the terms of the type \dot{q}_i^2 are called centrifugal while the terms of the type $\dot{q}_i \dot{q}_j$ are called Coriolis terms. Equation (A.33) is often written in matrix form as

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} \quad (\text{A.34})$$

where the k,j th element of the centrifugal/Coriolis matrix denoted $C(q, \dot{q}) \in \mathbb{R}^{2 \times 2}$ is defined as

$$c_{jk} = \sum_{i=1}^n c_{ijk} \dot{q}_i \quad (\text{A.35})$$

For a two-link planar robot manipulator, the inertia matrix $M(q)$ is derived in terms of linear and angular velocity which are function of the Jacobian matrix and the derivatives of the joint variables. Using the Jacobian matrix expressions defined earlier, the linear velocity of link 1 and 2 center of gravity represented by $v_{c_1}, v_{c_2} \in \mathbb{R}$, respectively, are given by

$$\begin{aligned} v_{c_1} &= J_{vc_1}(q)\dot{q} \\ \begin{bmatrix} \dot{x}_{c_1} \\ \dot{y}_{c_1} \end{bmatrix} &= J_{vc_1}(q) \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \end{aligned} \quad (\text{A.36})$$

$$\begin{aligned} v_{c_2} &= J_{vc_2}(q)\dot{q} \\ \begin{bmatrix} \dot{x}_{c_2} \\ \dot{y}_{c_2} \end{bmatrix} &= J_{vc_2}(q) \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \end{aligned} \quad (\text{A.37})$$

where

$$J_{vc_1}(q) = \begin{bmatrix} -l_{c_1} \sin(q_1) & 0 \\ l_{c_1} \cos(q_1) & 0 \\ 0 & 0 \end{bmatrix} \quad (\text{A.38})$$

and

$$J_{vc_2}(q) = \begin{bmatrix} -l_1 \sin(q_1) - l_{c_2} \sin(q_1 + q_2) & -l_{c_2} \sin(q_1 + q_2) \\ l_1 \cos(q_1) + l_{c_2} \cos(q_1 + q_2) & l_{c_2} \cos(q_1 + q_2) \\ 0 & 0 \end{bmatrix} \quad (\text{A.39})$$

Hence, the translational contribution to the total kinetic energy, $T_{rv} \in \mathbb{R}$, is

$$\begin{aligned} T_{rv} &= \frac{1}{2} m_1 v_{c_1}^T v_{c_1} + \frac{1}{2} m_2 v_{c_2}^T v_{c_2} \\ &= \frac{1}{2} \dot{q} \{ m_1 J_{vc_1}^T J_{vc_1} + m_2 J_{vc_2}^T J_{vc_2} \} \dot{q} \end{aligned} \quad (\text{A.40})$$

Considering that the angular velocity of the i th link, ω_i , is aligned with the z -axis of each joint coordinate frame, the rotational kinetic energy, $T_{r\omega} \in \mathbb{R}$, is simply $I_i \omega_i^2$. Hence the rotational contribution to the total kinetic energy is

$$T_{r\omega} = \frac{1}{2} \dot{q}^T \left\{ I_1 \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + I_2 \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \right\} \dot{q} \quad (\text{A.41})$$

where

$$I_i = \frac{m_i l_i^2}{12}; \quad i = 1, 2 \quad (\text{A.42})$$

The inertia matrix $M(q)$ is obtained by adding the two matrices in (A.40) and (A.41), as follows

$$M(q) \equiv \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} = m_1 J_{vc_1}^T J_{vc_1} + m_2 J_{vc_2}^T J_{vc_2} + \begin{bmatrix} I_1 + I_2 & I_2 \\ I_2 & I_2 \end{bmatrix} \quad (\text{A.43})$$

Expanding the previous equation, and using standard trigonometric identities $\cos^2(\theta) +$

$\sin^2(\theta) = 1$, $\cos(\alpha) \cos(\beta) + \sin(\alpha) \sin(\beta) = \cos(\alpha - \beta)$ yields

$$\begin{aligned} M_{11} &= m_1 l_{c_1}^2 + m_2(l_1^2 + l_{c_2}^2 + 2l_1 l_{c_2} \cos q_2) + I_1 + I_2 \\ M_{12} &= M_{21} = m_2(l_{c_2}^2 + l_1 l_{c_2} \cos q_2) + I_2 \\ M_{22} &= m_2 l_{c_2}^2 + I_2 \end{aligned} \quad (\text{A.44})$$

With the above expressions, the Christoffel symbols can be derived using the definition given in (A.32) as follows

$$c_{111} = \frac{1}{2} \frac{\partial M_{11}}{\partial q_1} = 0 \quad (\text{A.45})$$

$$c_{121} = c_{211} = \frac{1}{2} \frac{\partial M_{11}}{\partial q_2} = -m_2 l_1 l_{c_2} \sin q_2 = h \quad (\text{A.46})$$

$$c_{221} = \frac{\partial M_{12}}{\partial q_2} - \frac{1}{2} \frac{\partial M_{22}}{\partial q_1} = h \quad (\text{A.47})$$

$$c_{112} = \frac{\partial M_{21}}{\partial q_1} - \frac{1}{2} \frac{\partial M_{11}}{\partial q_2} = -h \quad (\text{A.48})$$

$$c_{122} = c_{212} = \frac{1}{2} \frac{\partial M_{22}}{\partial q_1} = 0 \quad (\text{A.49})$$

$$c_{222} = \frac{1}{2} \frac{\partial M_{22}}{\partial q_2} = 0 \quad (\text{A.50})$$

With the definition of the centrifugal/Coriolis matrix given in (A.35), the matrix $C(q, \dot{q})$ is

$$C(q, \dot{q}) = -m_2 l_1 l_{c_2} \sin q_2 \begin{bmatrix} \dot{q}_2 & \dot{q}_1 + \dot{q}_2 \\ -\dot{q}_1 & 0 \end{bmatrix} \quad (\text{A.51})$$

Appendix B

Fuzzy Logic MATRIXx Code

This appendix presents the MATRIXx programming code of the fuzzy logic-based adaptation mechanism developed in Chapter 4 to calculate the time-varying control parameters $\lambda_q \in \mathbb{R} \forall q = 1, \dots, 4$.

B.1 Initialization File

```
1  #=====
   # Initialization of the adaptive fuzzy controller
   # Written by: Steve Ulrich
   # Copyright(c) 2011 by Steve Ulrich
   #=====

   If !exist(CTL_FUZ, {partition}) then
       NEW PARTITION CTL_FUZ
   endif
10  set partition CTL_FUZ

   #-----
   # Control gain and input/output scaling gains
   # -----

   Kv = 35*eye(2,2);

   g0 = 1.5;
20  g1 = 1.5;
   g2 = 0.1;
   g3 = 0.1;

   h0 = 100;
   h1 = 100;
   h2 = 270;
   h3 = 270;

30  #-----
   # Rules
   # -----

   rules = [9, 9, 7, 6, 5, 6, 7, 9, 9;
            9, 9, 6, 5, 4, 5, 6, 9, 9;
            7, 6, 5, 4, 3, 4, 5, 6, 7;
            6, 5, 4, 3, 2, 3, 4, 5, 6;
            5, 4, 3, 2, 1, 2, 3, 4, 5;
            6, 5, 4, 3, 2, 3, 4, 5, 6;
40  7, 6, 5, 4, 3, 4, 5, 6, 7;
            9, 9, 6, 5, 4, 5, 6, 9, 9;
            9, 9, 7, 6, 5, 6, 7, 9, 9];
```

```

rulesVEL = [5,4,3,2,1,2,3,4,5];

#-----
# Centers of output membership functions
#-----
50 # [ZERO, PVVL, PVL, PL, PMED, PH, PVH, PVVH, PMAX]
   c = [0, 0.125, 0.25, 0.35, 0.5, 0.625, 0.75, 0.875, 1];
   # [ZERO, PL, PMED, PH, PMAX]
   cVEL = [0, 0.25, 0.5, 0.75, 1];

#-----
# Gaussian membership function definitions
#-----
60
   nbpts = 100;
   X = [0:(1/nbpts):1];
   outmf = zeros(9,nbpts+1);
   for i=1:9
       outmf(i,1:nbpts+1) = EXP(-0.5*((X-c(i))/(1/24))^2);
   endfor
70
   outmfVEL = zeros(5,nbpts+1);
   for i=1:5
       outmfVEL(i,1:nbpts+1) = EXP(-0.5*((X-cVEL(i))/(1/12))^2);
   endfor

#-----
# Initilization of output fuzzy sets to zero
#-----
80
# 1 = Implied Fuzzy Sets
# 2 = Overall (Aggregated) Fuzzy Set

   flag = 1
   if (flag==1)then
       impmf = zeros(1,nbpts+1);
       impmfVEL = impmf;
90   aggmf = 0;
   else
       impmf = zeros(81,nbpts+1);
       impmfVEL = zeros(9,nbpts+1);
       aggmf = zeros(1,nbpts+1);
   endif
   set partition main;

```

B.2 Adaptation Mechanism Function

```

1  #=====
   # Fuzzy adaptation mechanism of control parameter lambda_1
   # or lambda_2
   # Written by: Steve Ulrich
   # Copyright(c) 2011 by Steve Ulrich
   #=====

function [lambda] = TYPE1(t,u)

10 #-----
   # Input vector assignation
   #-----
   x1 = u(1); # pos_x_err or pos_y_err
   x2 = u(2); # vel_x_err or vel_y_err

   #-----
   # Parameters initialization
   #-----
20  rules = CTL_FUZ.rules;
   nbpts = CTL_FUZ.nbpts;
   X = CTL_FUZ.X;
   uA = 0;
   outmf = CTL_FUZ.outmf;
   impmf = CTL_FUZ.implemf;
   aggmf = CTL_FUZ.aggmf;
   c = CTL_FUZ.c;
   flag = CTL_FUZ.flag;

30 #-----
   # Fuzzification
   #-----
   u_x1 = zeros(1,9);
   u_x2 = zeros(1,9);

   u_x1(1) = EXP(-0.5*((x1+1)/(1/12))^2);      # NVH
   u_x1(2) = EXP(-0.5*((x1+0.75)/(1/12))^2);  # NH
   u_x1(3) = EXP(-0.5*((x1+0.5)/(1/12))^2);   # NL
40  u_x1(4) = EXP(-0.5*((x1+0.25)/(1/12))^2); # NVL
   u_x1(5) = EXP(-0.5*((x1)/(1/12))^2);       # ZERO
   u_x1(6) = EXP(-0.5*((x1-0.25)/(1/12))^2);  # PVL
   u_x1(7) = EXP(-0.5*((x1-0.5)/(1/12))^2);   # PL
   u_x1(8) = EXP(-0.5*((x1-0.75)/(1/12))^2);  # PH
   u_x1(9) = EXP(-0.5*((x1-1)/(1/12))^2);     # PVH

   u_x2(1) = EXP(-0.5*((x2+1)/(1/12))^2);     # NVH
   u_x2(2) = EXP(-0.5*((x2+0.75)/(1/12))^2);  # NH
   u_x2(3) = EXP(-0.5*((x2+0.5)/(1/12))^2);   # NL
50  u_x2(4) = EXP(-0.5*((x2+0.25)/(1/12))^2); # NVL
   u_x2(5) = EXP(-0.5*((x2)/(1/12))^2);       # ZERO
   u_x2(6) = EXP(-0.5*((x2-0.25)/(1/12))^2);  # PVL
   u_x2(7) = EXP(-0.5*((x2-0.5)/(1/12))^2);   # PL
   u_x2(8) = EXP(-0.5*((x2-0.75)/(1/12))^2);  # PH
   u_x2(9) = EXP(-0.5*((x2-1)/(1/12))^2);     # PVH

   #-----
   # Fuzzy inference mechanism - implied fuzzy sets
   #-----
60 #-----

```

```

if (flag==1) then
    area = 0;
    num = 0;
    den = 0;
    for i=1:9
        for j=1:9
70      # Antecedent Composition (MIN)
          uA = min(u_x2(i),u_x1(j));
          # Implication (MIN - Mamdani)
          impmf = min(outmf(rules(i,j),:),uA);
          # Find area under all implied fuzzy sets
          area = sum((1/nbpts)*(impmf(1:nbpts) + impmf(2:nbpts+1))/2);
          # Defuzzification (COG Numerator)
80      num = num + c(rules(i,j))*area;
          # Defuzzification (COG Denominator)
          den = den + area;
        endfor
    endfor

    # Crisp Output
    lambda = num/den;
90    else

#-----
# Fuzzy inference mechanism - aggregated implied fuzzy set
#-----
    for i=1:9
        for j=1:9
100     # Antecedent Composition (MIN)
          uA = min(u_x2(i),u_x1(j));
          # Implication (MIN - Mamdani)
          impmf((i-1)*9+j,1:nbpts+1) = min(outmf(rules(i,j),:),uA);
        endfor
    endfor

    # Aggregation (MAX)
110   for i=1:nbpts+1
          aggmf(1,i)=max(impmf(:,i))
        endfor
    Xaggmf = X.*aggmf;

    # Defuzzification (COA Numerator)
    num = sum((1/nbpts)*(Xaggmf(1:nbpts) + Xaggmf(2:nbpts+1))/2);
    # Defuzzification (COA Denominator)
120   den = sum((1/nbpts)*(aggmf(1:nbpts) + aggmf(2:nbpts+1))/2);
    # Crisp Output
    lambda = num(1,1,nbpts+1)/den(1,1,nbpts+1);
endif

```

```

endfunction

1  #=====
# Fuzzy adaptation mechanism of control parameter lambda_3
# or lambda_4
# Written by: Steve Ulrich
# Copyright(c) 2011 by Steve Ulrich
#=====

function [lambda] = TYPE1VEL(t,u)

10 #-----
# Input vector assignation
#-----
    x1 = u(1); # vel_x_err or vel_y_err

#-----
# Parameters initialization
#-----
20 rulesVEL = CTL_FUZ.rulesVEL;
   nbpts = CTL_FUZ.nbpts;
   X = CTL_FUZ.X;
   uA = 0;
   outmfVEL = CTL_FUZ.outmfVEL;
   impmfVEL = CTL_FUZ.impmfVEL;
   aggmf = CTL_FUZ.aggmf;
   cVEL = CTL_FUZ.cVEL;
   flag = CTL_FUZ.flag;

30 #-----
# Fuzzification
#-----
    u_x1 = zeros(1,9);
    u_x1(1) = EXP(-0.5*((x1+1)/(1/12))^2);      # NVH
    u_x1(2) = EXP(-0.5*((x1+0.75)/(1/12))^2);  # NH
    u_x1(3) = EXP(-0.5*((x1+0.5)/(1/12))^2);  # NL
    u_x1(4) = EXP(-0.5*((x1+0.25)/(1/12))^2); # NVL
    u_x1(5) = EXP(-0.5*(x1)/(1/12))^2);       # ZERO
40 u_x1(6) = EXP(-0.5*((x1-0.25)/(1/12))^2);  # PVL
    u_x1(7) = EXP(-0.5*((x1-0.5)/(1/12))^2);  # PL
    u_x1(8) = EXP(-0.5*((x1-0.75)/(1/12))^2); # PH
    u_x1(9) = EXP(-0.5*((x1-1)/(1/12))^2);    # PVH

#-----
# Fuzzy inference mechanism - implied fuzzy sets
#-----
50 if (flag==1) then
    area = 0;
    num = 0;
    den = 0;

    for i=1:9
        # Implication (MIN - Mamdani)
        impmfVEL = min(outmfVEL(rulesVEL(i),:),u_x1(i));

```

```

60    # Find area under all implied fuzzy sets
      area = sum((1/nbpts)*(impmfVEL(1:nbpts) + impmfVEL(2:nbpts+1))/2);
      # Defuzzification (COG Numerator)
      num = num + cVEL(rulesVEL(i))*area;
      # Defuzzification (COG Denominator)
      den = den + area;
70    endfor
      # Crisp Output
      lambda = num/den;
      else

#-----
# Fuzzy inference mechanism - aggregated implied fuzzy set
#-----
80    # Implication (MIN - Mamdani)
      for i=1:9
        impmfVEL(i,1:nbpts+1) = min(outmfVEL(rulesVEL(i,:),u_x1(i)));
      endfor

      # Aggregation (MAX)
      for i=1:nbpts+1
        aggmf(1,i) = max(impmfVEL(:,i));
      endfor
90    Xaggmf = X.*aggmf;

      # Defuzzification (COA Numerator)
      num = sum((1/nbpts)*(Xaggmf(1:nbpts) + Xaggmf(2:nbpts+1))/2);
      # Defuzzification (COA Denominator)
      den = sum((1/nbpts)*(aggmf(1:nbpts) + aggmf(2:nbpts+1))/2);
      # Crisp Output
100   lambda = num(1,1,nbpts+1)/den(1,1,nbpts+1);
      endif
      endfunction

```

Appendix C

DSAC/DMSAC for Square Flexible-Joint Manipulators

This appendix presents and briefly discusses the simulation results obtained with the DSAC/DMSAC composite control methodologies for a flexible-joint manipulator modeled as a square system. Although similar results to those presented in Chapter 4 are obtained, the present chapter is distinctly different, as no formal analytical guarantee of closed-loop stability could be obtained in Chapter 4 (recall that the closed-loop stability of the DMSAC and DSAC control methodologies can only be guaranteed for square manipulators).

C.1 Control Objective

The control objective consists in designing a DSAC and a DMSAC composite controllers which ensure that the output vector of the flexible-joint manipulator system denoted by $y(t) \in \mathbb{R}^2$ tracks the time-varying reference model output $y_m(t) \in \mathbb{R}^2$, regardless of uncertainties and modeling errors. Let define $y(t)$ and $y_m(t)$ as

$$y \triangleq \alpha x_r + \dot{x}_r \quad (\text{C.1})$$

$$y_m \triangleq \alpha x_{r_m} + \dot{x}_{r_m} \quad (\text{C.2})$$

such that a tracking error, denoted by $e_y(t) \in \mathbb{R}^2$, is defined as

$$e_y \triangleq y_m - y = \left[\alpha (x_{r_m} - x_r) + (\dot{x}_{r_m} - \dot{x}_r) \right] \quad (\text{C.3})$$

where the reference model defines the ideal response to the desired square trajectory, and is expressed in terms of the ideal damping ratio ζ , and undamped natural frequency ω_n . The position scaling factor α was arbitrarily selected as 0.5.

C.2 DSAC-Based Composite Control

The following DSAC composite control law presented in (4.52) is used

$$\tau = J^T(q) [K_e(t)e_y + K_x(t)x_m + K_u(t)u_m] + K_v(\dot{q} - \dot{q}_m) \quad (\text{C.4})$$

where the time-varying control gains are defined as in (4.53)-(4.55), and where the adaptation mechanism is defined in (3.25), (3.26) and (3.28)-(3.31).

C.3 DMSAC-Based Composite Control

The DMSAC composite controller given in (4.52) is considered

$$\tau = J^T(q)K_e(t)e_y + K_v(\dot{q} - \dot{q}_m) \quad (\text{C.5})$$

where the time-varying control gain $K_e(t)$ can be defined as in (4.53), with $K_{P_e}(t)$, $K_{I_e}(t)$ updated as (3.25) and (3.26).

C.4 Simulation Results

This section presents the numerical simulations used to validate and compare the DSAC and DMSAC composite controllers for a flexible-joint manipulator modeled a square system. The parameters, gains and coefficients of the DSAC composite control algorithm were determined as follows

$$\begin{aligned} \Gamma_{P_e} &= 15I_2 & \Gamma_{I_e} &= 50I_2 & \Gamma_{P_x} = \Gamma_{P_u} &= 0.1I_4 & \Gamma_{I_x} = \Gamma_{I_u} &= 0.2I_4 \\ \sigma_e &= 0.1I_2 \\ \sigma_x = \sigma_u &= \text{diag} \left[0.9 \quad 0.9 \quad 0.4 \quad 0.4 \right] \\ K_v &= 120I_2 \end{aligned}$$

and the control parameters, gains and coefficients of the DMSAC composite controller were selected as follows

$$\Gamma_{P_e} = 15I_2 \quad \Gamma_{I_e} = 35I_2 \quad \sigma_e = 0.03I_2 \quad K_v = 35I_2$$

As previously, the control parameters listed above were selected in numerical simulations to provide good trajectory tracking performance for the flexible-joint manipulator modeled with the nominal linear joint stiffness dynamics representation.

Results for the DMSAC and the DSAC control laws are shown in Figs. C.1 to C.3, and in Figs. C.4 to C.8, respectively. As in Chapter 4, the DSAC composite control strategy provides improved tracking results, when compared to the DMSAC composite controller. This is demonstrated in Figs. C.1 and C.4, where the DMSAC control strategy exhibits greater positioning overshoots; 0.117 m, 0.112 m and 0.104 m, for the DMSAC control law, in comparison with 0.082 m, 0.079 m and 0.079 m for the DSAC control law, at the first, second and third direction change, respectively.

To validate the robustness to parametric uncertainties, the same DMSAC and DSAC composite controllers tuned for the nominal robot manipulator were applied to the uncertain linear joint stiffness robot model ($k = 200I_2$ N·m/rad). The results are provided in Figs. C.9 to C.16. Both composite control methodologies achieved similar tracking performance in terms positioning errors; 0.124 m, 0.126 m and 0.121 m for the DMSAC strategy, compared with 0.127 m, 0.129 m and 0.123 m for the DSAC strategy.. However, with the DSAC approach, the settling time between two direction changes is greater, as shown in Figs. C.13 to C.16. This endorses the results obtained in Chapter 4 with the nonsquare manipulator model, and indicates that the DSAC composite controller is more sensitive to sudden changes in the desired trajectory when applied to a manipulator with excessively low joint stiffness coefficients.

Finally, both adaptive control schemes were applied to the nonlinear joint stiffness dynamics model. Similar to Chapter 4, both adaptive control strategies provide a stable closed-loop behavior, yet the trajectory obtained with the DSAC control law demonstrates improved tracking results, as shown in Figs. C.17 and C.20

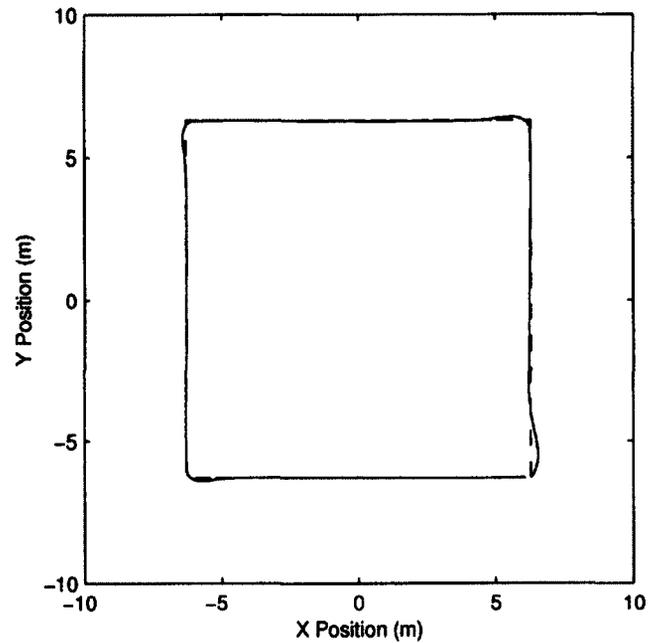


Figure C.1: Trajectory tracking results obtained with the DMSAC composite controller applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m) modeled as a square system. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$.

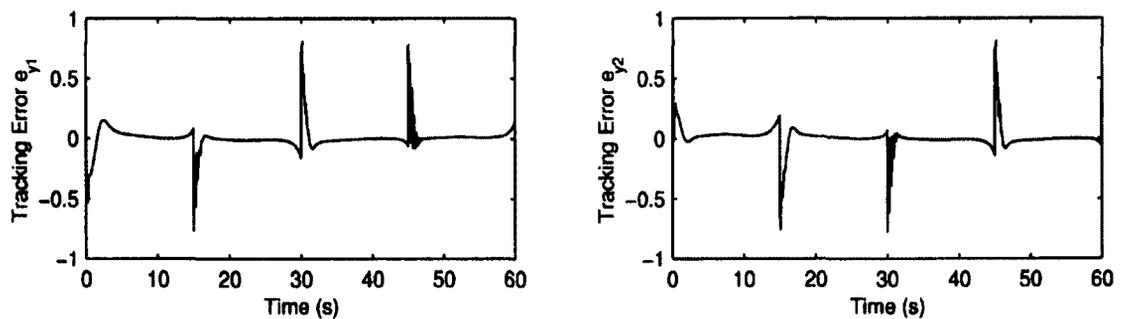


Figure C.2: Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the DMSAC composite controller applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m) modeled as a square system.

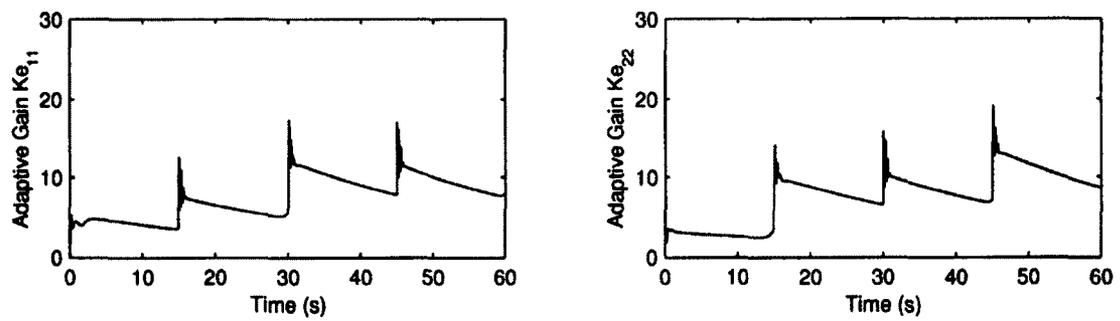


Figure C.3: Adaptation history of the DMSAC composite controller gains $K_{e_p}(t)$ and $K_{e_d}(t)$ when applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m) modeled as a square system.

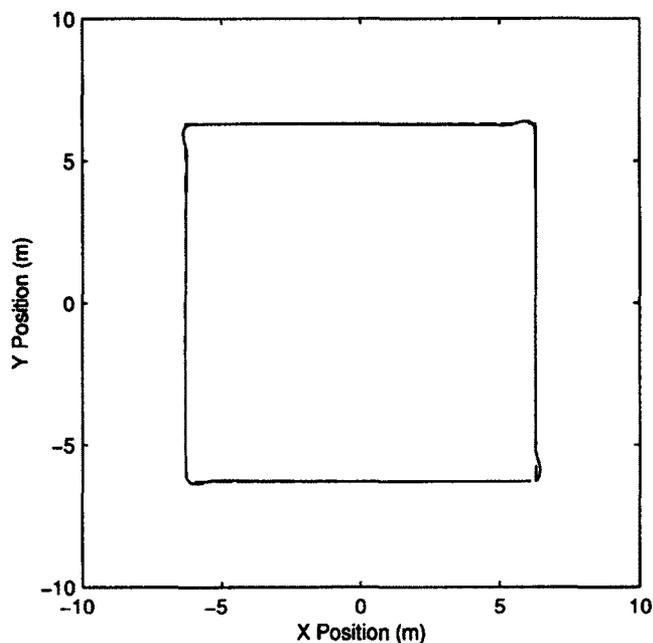


Figure C.4: Trajectory tracking results obtained with the DSAC composite controller applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m) modeled as a square system. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$.

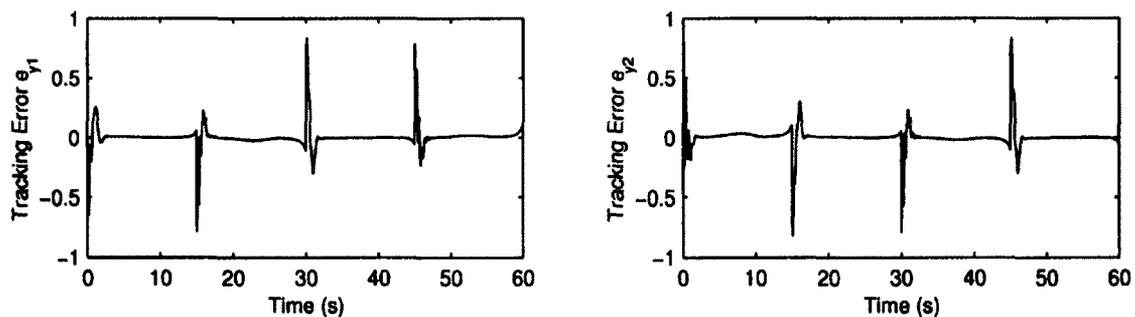


Figure C.5: Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the DSAC composite controller applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m) modeled as a square system.

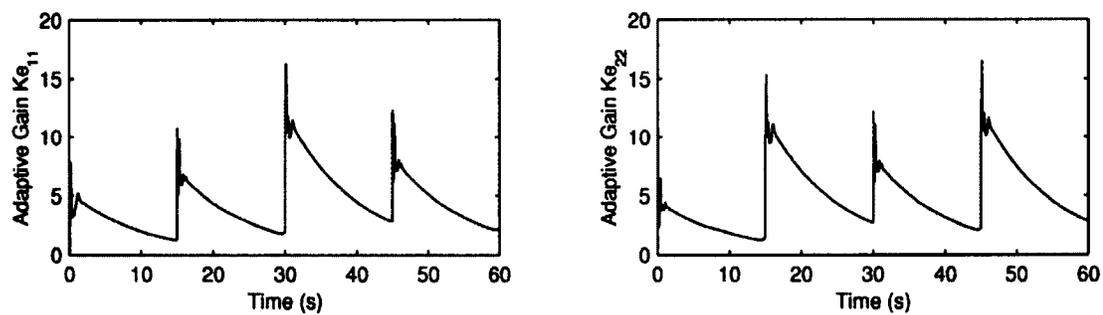


Figure C.6: Adaptation history of the DSAC composite controller gains $K_{e_p}(t)$ and $K_{e_d}(t)$ when applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m) modeled as a square system.

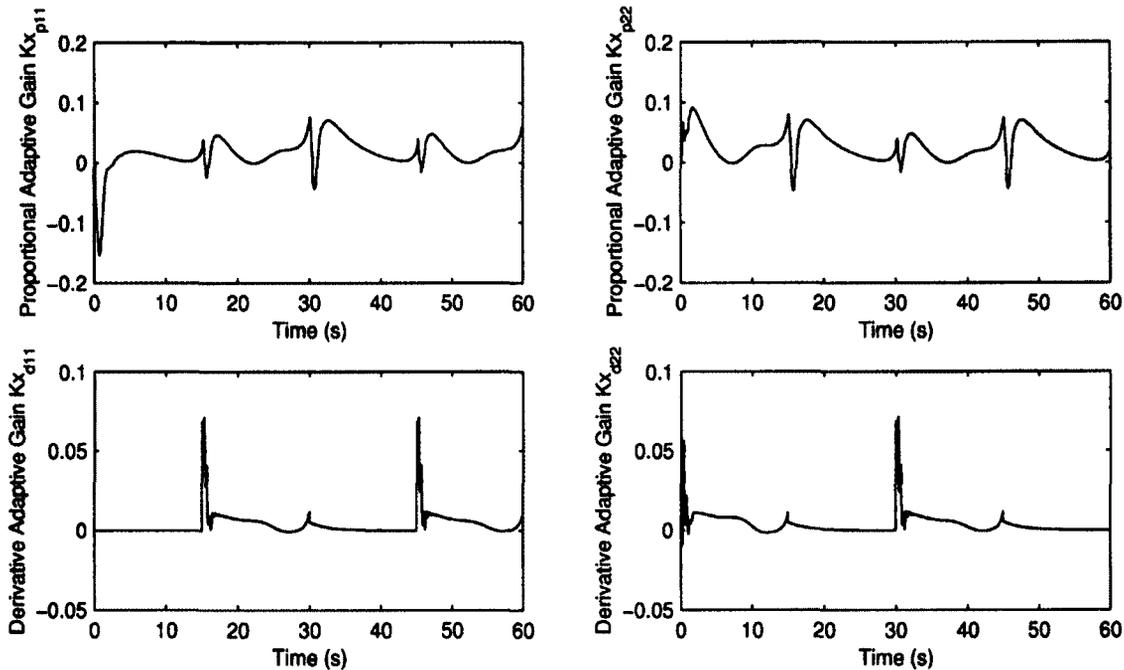


Figure C.7: Adaptation history of the DSAC composite controller gains $K_{x_p}(t)$ and $K_{x_d}(t)$ when applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m) modeled as a square system.

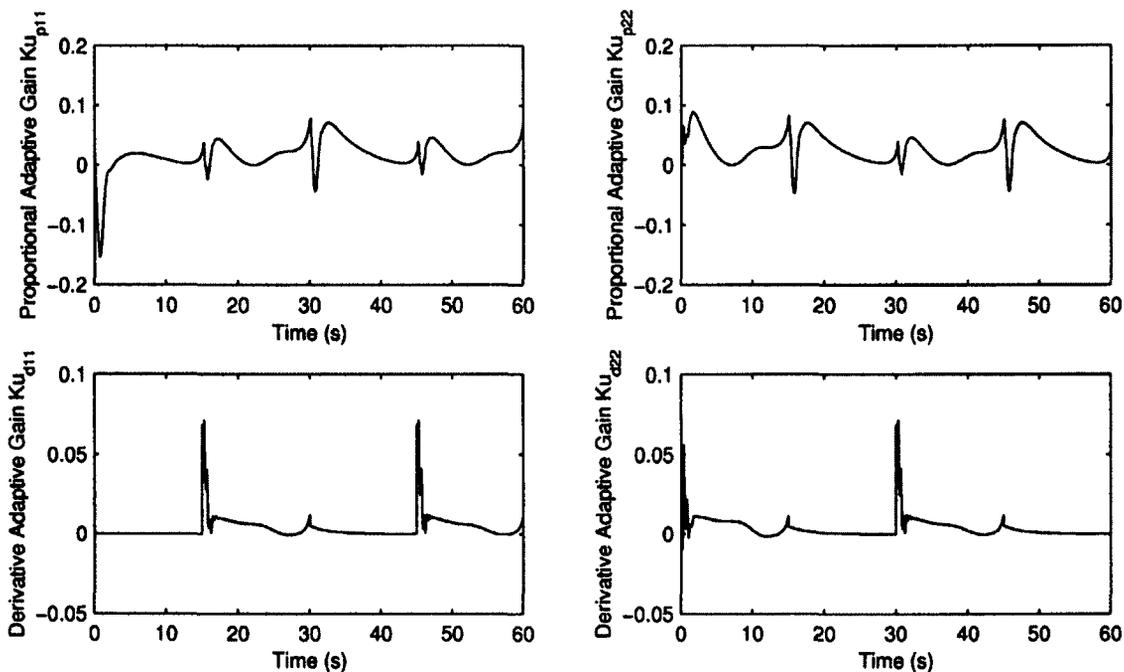


Figure C.8: Adaptation history of the DSAC composite controller gains $K_{u_p}(t)$ and $K_{u_d}(t)$ when applied to the nominal linear joint stiffness manipulator ($k = 500I_2$ N·m) modeled as a square system.

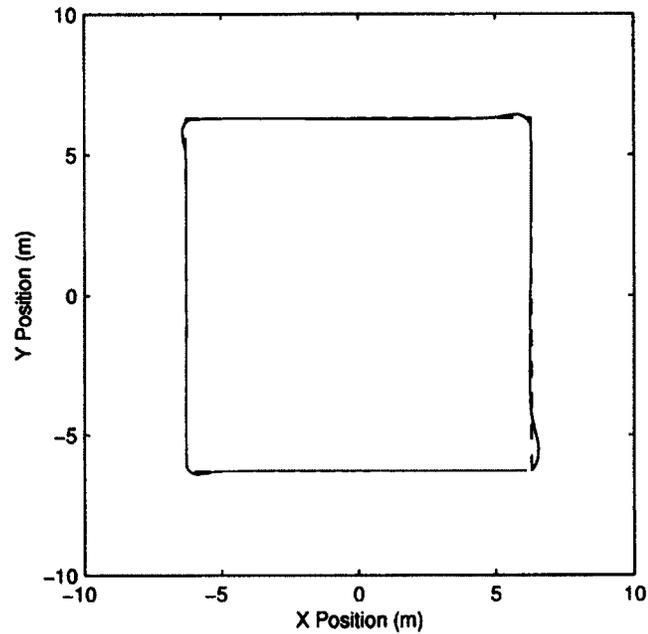


Figure C.9: Trajectory tracking results obtained with the DMSAC composite controller applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m) modeled as a square system. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$.

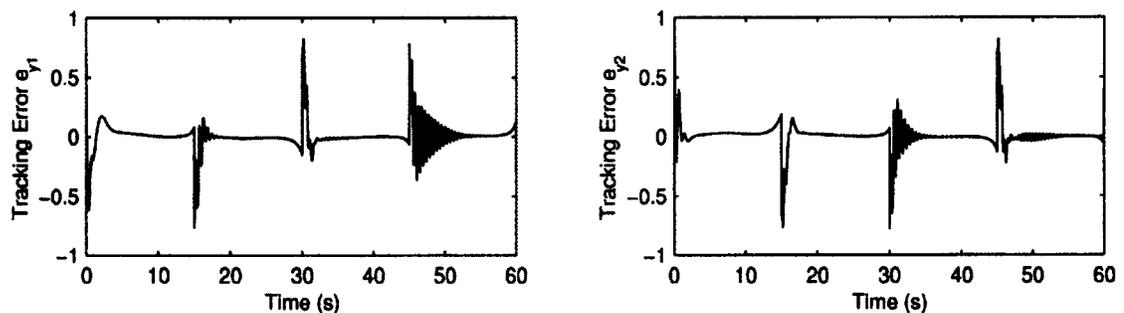


Figure C.10: Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the DMSAC composite controller applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m) modeled as a square system.

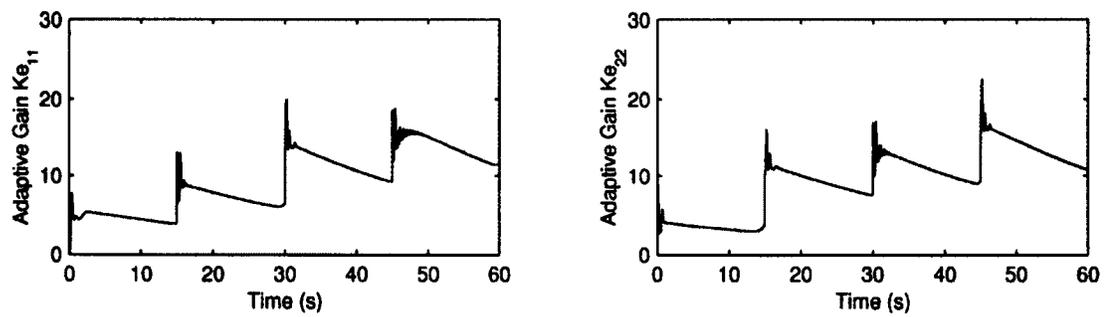


Figure C.11: Adaptation history of the DMSAC composite controller gains $K_{e_p}(t)$ and $K_{e_d}(t)$ when applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m) modeled as a square system.

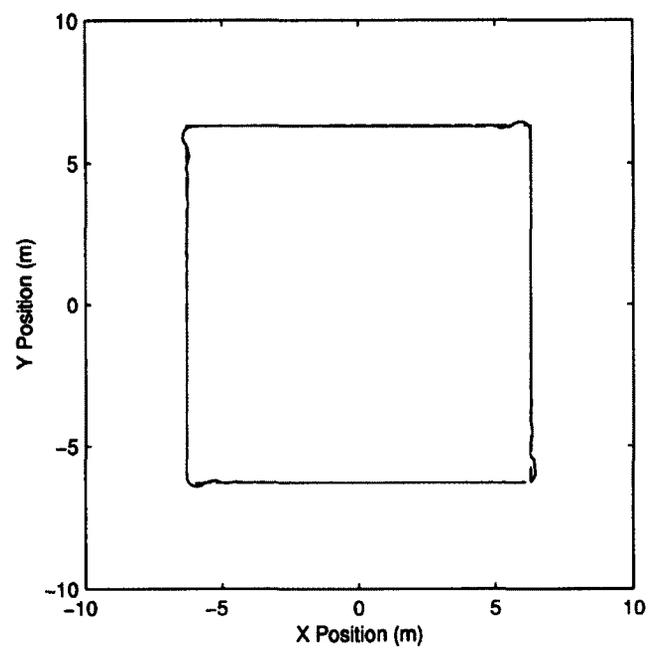


Figure C.12: Trajectory tracking results obtained with the DSAC composite controller applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m) modeled as a square system. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$.

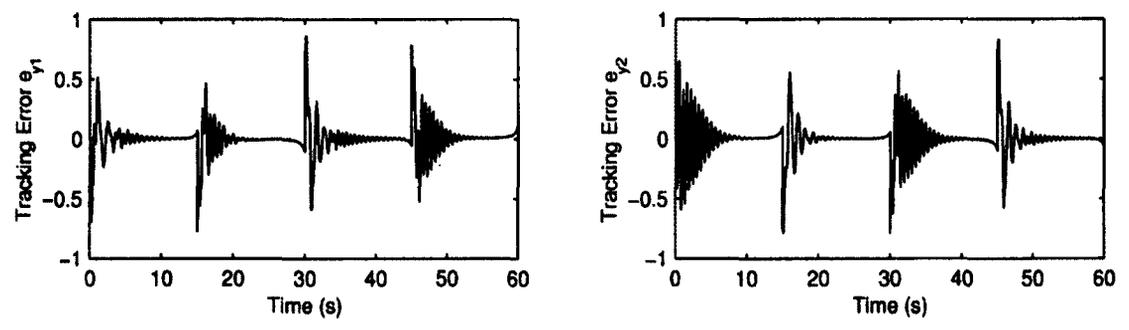


Figure C.13: Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the DSAC composite controller applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m) modeled as a square system.

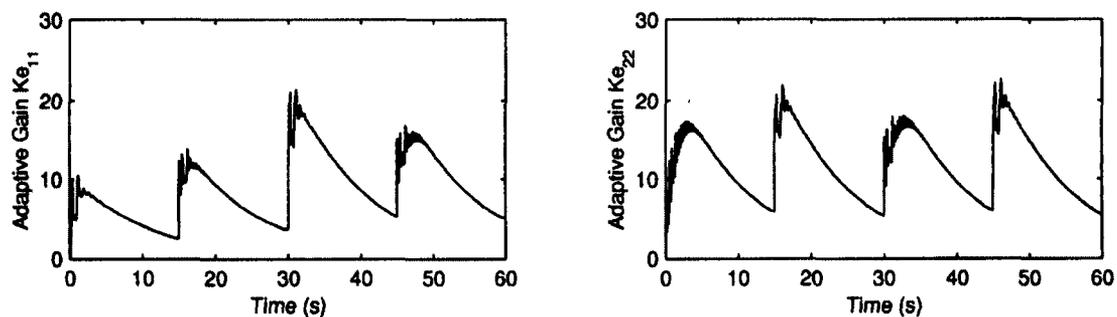


Figure C.14: Adaptation history of the DSAC composite controller gains $K_{e_p}(t)$ and $K_{e_d}(t)$ when applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m) modeled as a square system.

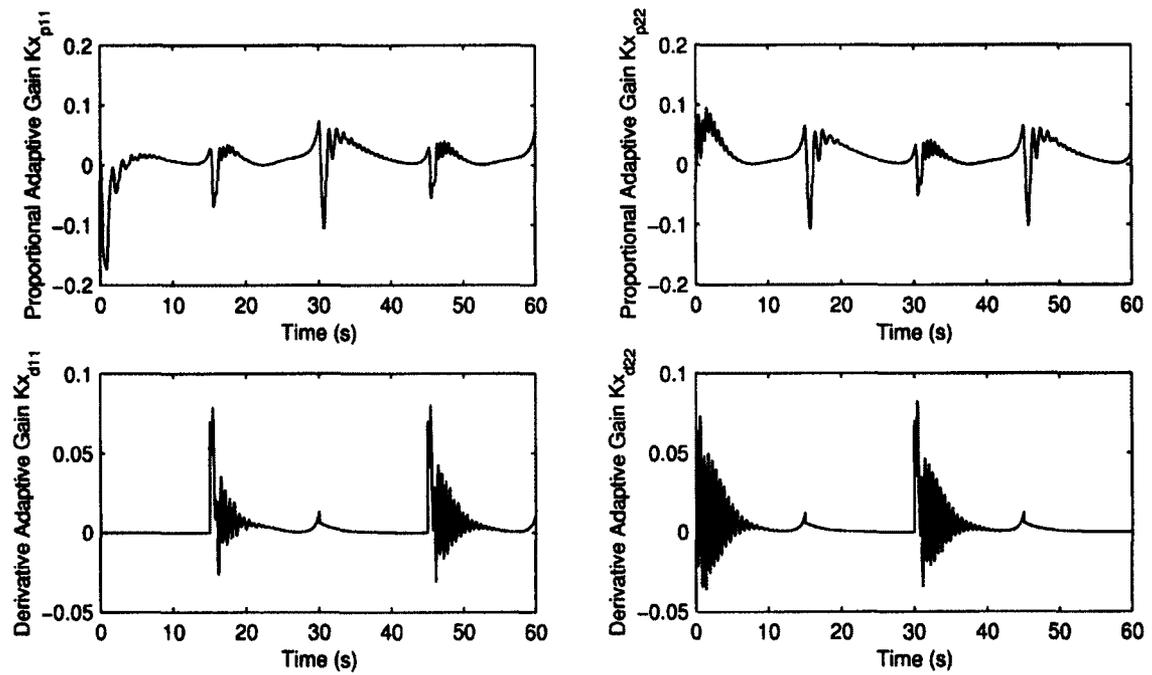


Figure C.15: Adaptation history of the DSAC composite controller gains $K_{x_p}(t)$ and $K_{x_d}(t)$ when applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m) modeled as a square system.

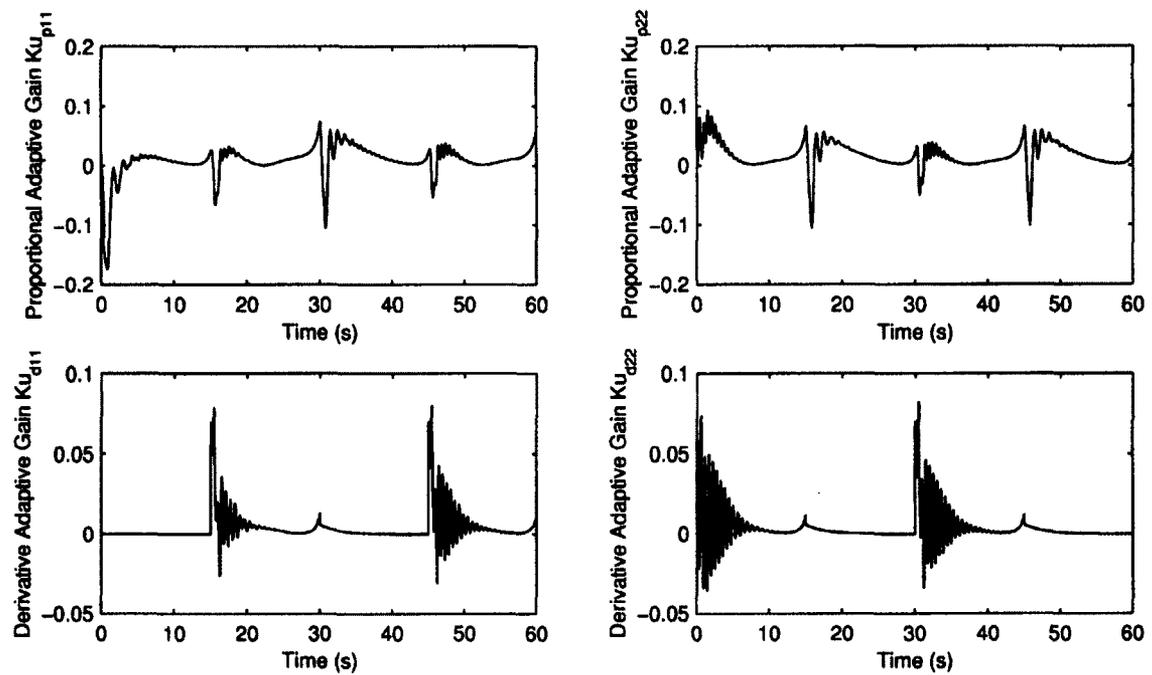


Figure C.16: Adaptation history of the DSAC composite controller gains $K_{u_p}(t)$ and $K_{u_d}(t)$ when applied to the uncertain linear joint stiffness manipulator ($k = 200I_2$ N·m) modeled as a square system.

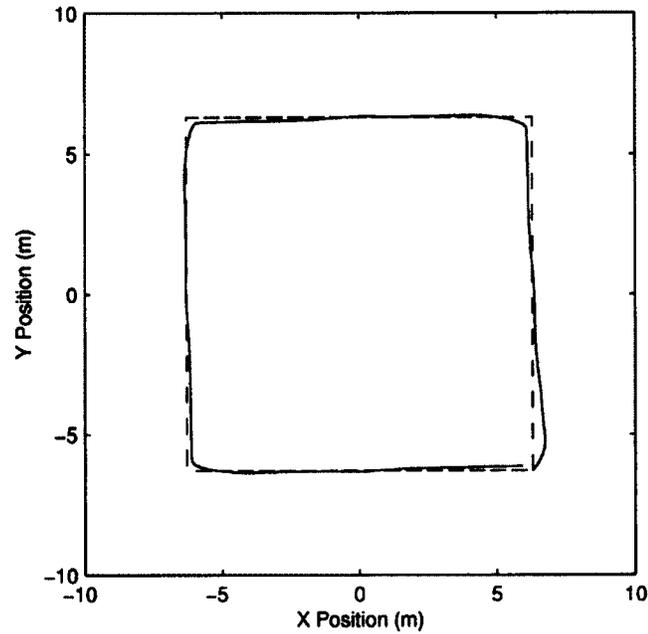


Figure C.17: Trajectory tracking results obtained with the DMSAC composite controller applied to the nonlinear joint stiffness manipulator modeled as a square system. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$.

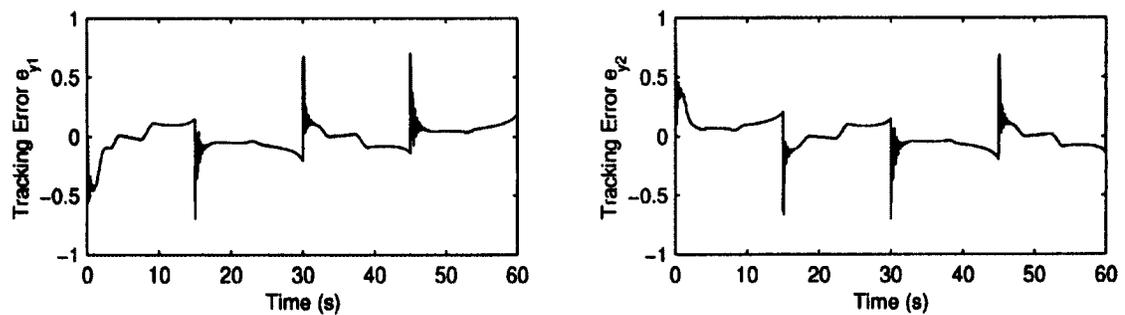


Figure C.18: Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the DMSAC composite controller applied to the nonlinear joint stiffness manipulator modeled as a square system.

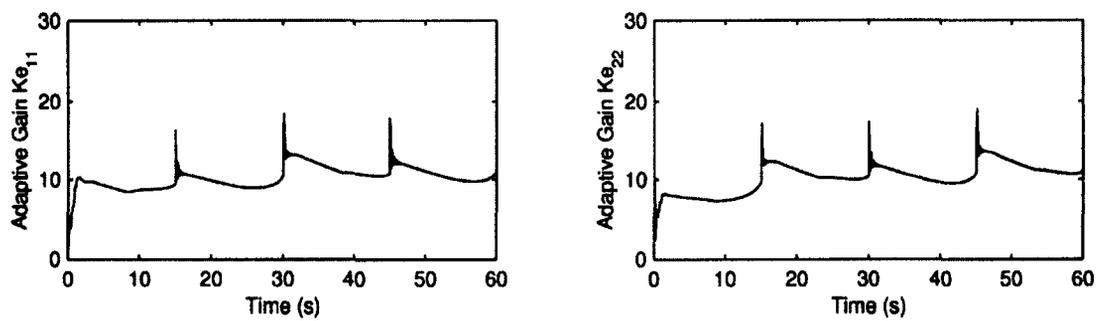


Figure C.19: Adaptation history of the DMSAC composite controller gains $K_{e_p}(t)$ and $K_{e_d}(t)$ when applied to the nonlinear joint stiffness manipulator modeled as a square system.

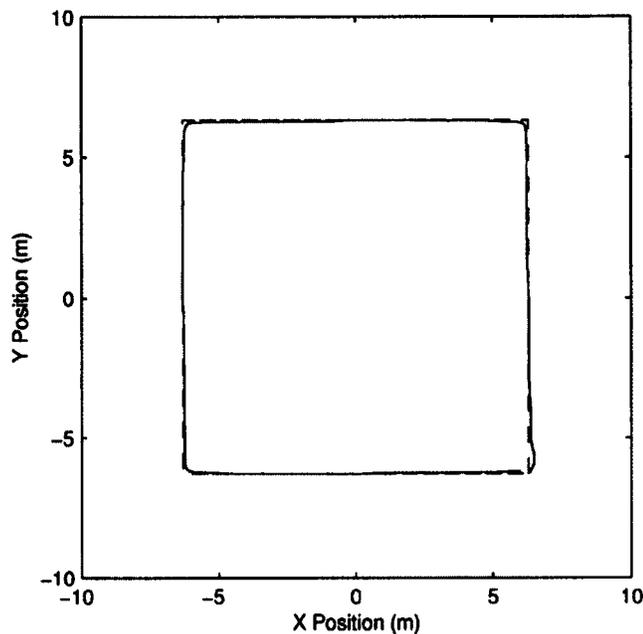


Figure C.20: Trajectory tracking results obtained with the DSAC composite controller applied to the nonlinear joint stiffness manipulator modeled as a square system. The dashed line corresponds to the desired end-effector position $x_{r_d}(t)$, and the solid line corresponds to the actual end-effector position $x_r(t)$.

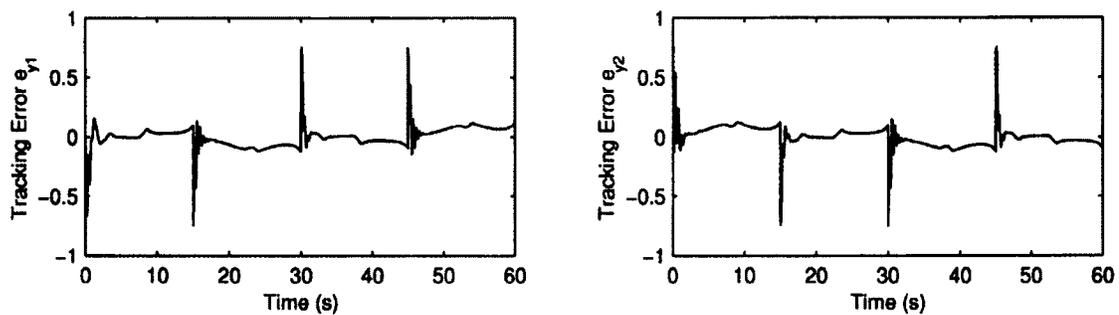


Figure C.21: Trajectory tracking errors ($e_y(t) = y_m(t) - y(t)$) obtained with the DSAC composite controller applied to the nonlinear joint stiffness manipulator modeled as a square system.

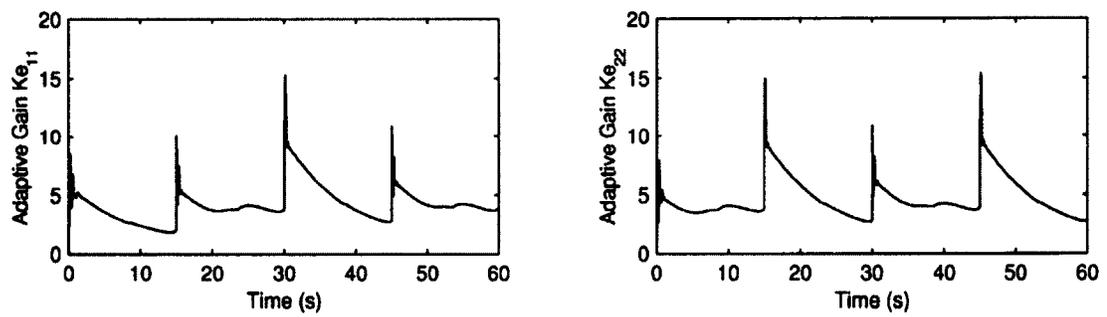


Figure C.22: Adaptation history of the DSAC composite controller gains $K_{e_p}(t)$ and $K_{e_d}(t)$ when applied to the nonlinear joint stiffness manipulator modeled as a square system.

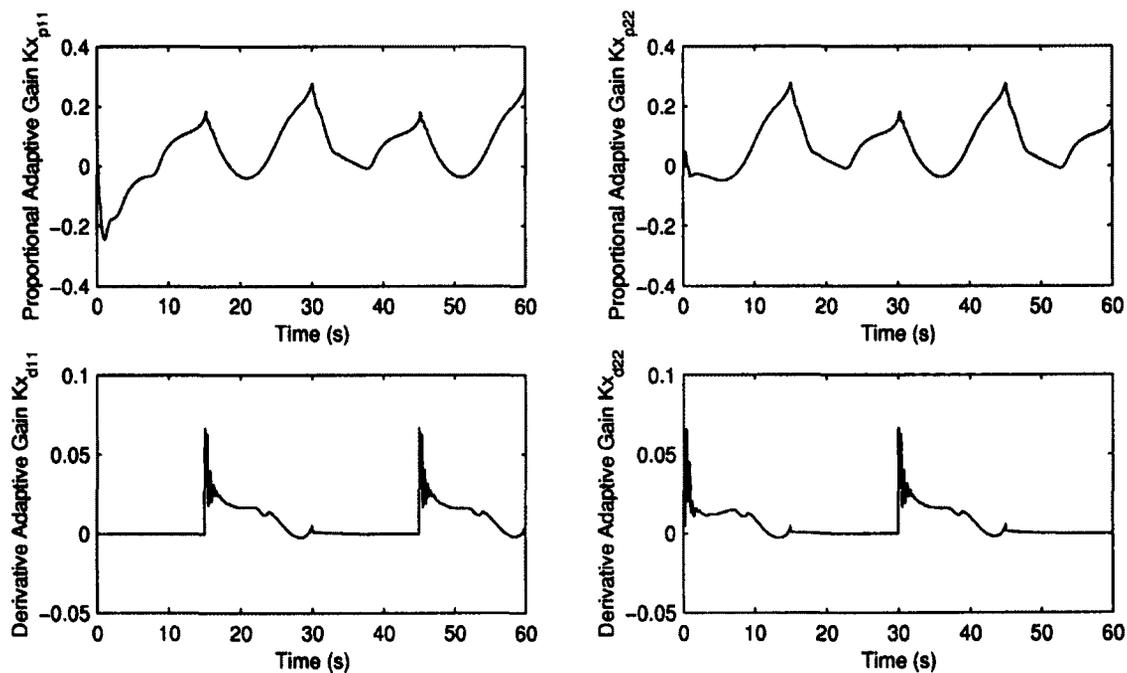


Figure C.23: Adaptation history of the DSAC composite controller gains $K_{x_p}(t)$ and $K_{x_d}(t)$ when applied to the nonlinear joint stiffness manipulator modeled as a square system.

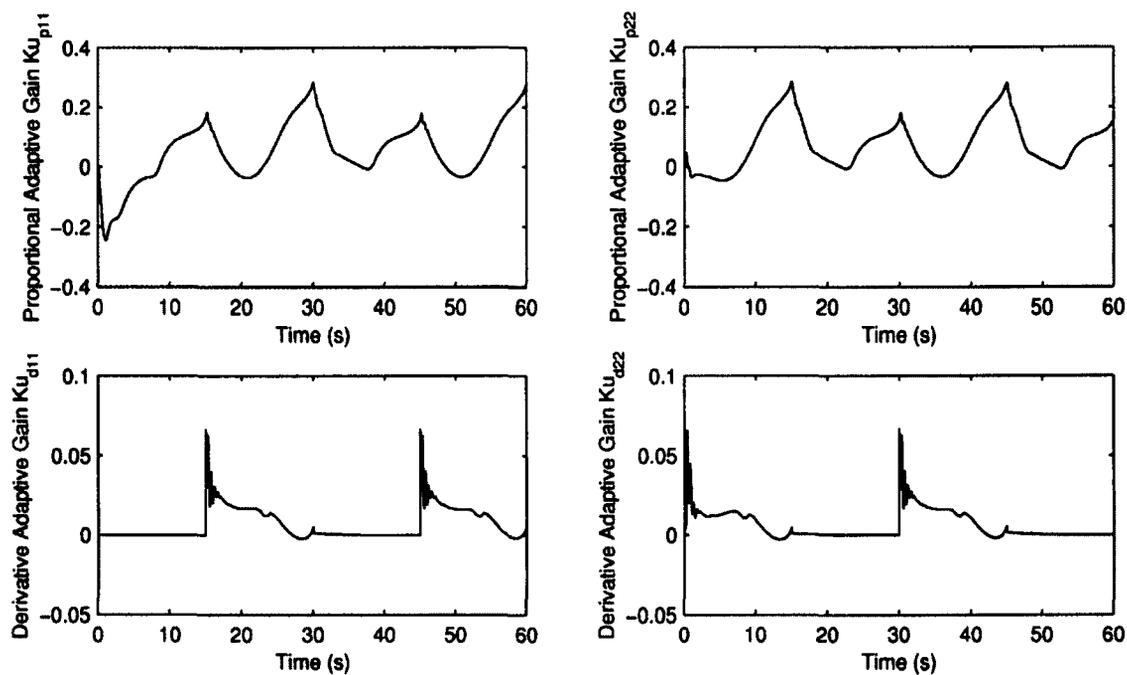


Figure C.24: Adaptation history of the DSAC composite controller gains $K_{u_p}(t)$ and $K_{u_d}(t)$ when applied to the nonlinear joint stiffness manipulator modeled as a square system.

Appendix D

Nonlinear Backstepping Control

This appendix presents the development of the nonlinear flexible-joint controller based on the backstepping control theory, as proposed by Brogliato *et al.* [28].

D.1 Nonlinear Backstepping Control Development

Brogliato *et al.* [28] present three nonlinear design techniques for global stabilization of flexible-joint manipulators, among them, the so-called nonlinear backstepping control methodology. The backstepping control scheme uses the cascade decomposition property of the model combined with the integrator augmentation stabilization of Kokotovic and Sussmann [207]. To develop their nonlinear control law, Brogliato *et al.* [28] have adopted the backstepping technique as a step-by-step methodology, in order to highlight that it is a systematic control design procedure. Several variations are possible at some stages of the backstepping design. This could lead to a controller with better tracking performance, although this may not seem possible to obtain other controllers than the one proposed without modifying the initial triangular structure, hence resulting in an antisymmetric system. Another alternative could have been to select $q_m - q$ as the hypothetical input, as proposed by Nicosia and Tomei [61]. However, the controller obtained with q_m represents a more general case than the one proposed with $q_m - q$. The resulting nonlinear backstepping controller is given by [28]

$$\tau = J_m [\ddot{q}_{m_d} - 2(\dot{q}_m - \dot{q}_{m_d}) - 2(q_m - q_{m_d}) - (\dot{s} + s)] + k(q_m - q) \quad (\text{D.1})$$

where $s \in \mathbb{R}^n$ is given by

$$s = (\dot{q} - \dot{q}_d) + \Lambda (q - q_d) \quad (\text{D.2})$$

with $\Lambda \in \mathbb{R}^{n \times n}$ denoting a constant gain matrix. In (D.1), $q_{m_d}, \dot{q}_{m_d}, \ddot{q}_{m_d} \in \mathbb{R}^n$ denotes respectively the desired motor angular position, velocity and acceleration vectors, and are given by

$$q_{m_d} = k^{-1} \tau_{SLI} + q \quad (\text{D.3})$$

$$\dot{q}_{m_d} = k^{-1} \dot{\tau}_{SLI} + \dot{q} \quad (\text{D.4})$$

$$\ddot{q}_{m_d} = k^{-1} \ddot{\tau}_{SLI} + \ddot{q} \quad (\text{D.5})$$

In (D.3)-(D.5), the term τ_{SLI} denotes the well-known adaptive control law proposed by Slotine and Li [160] for rigid-joint robots. This rigid-joint adaptive control law consists of a PD feedback part and a full dynamics feedforward compensation part. Although the controller gains and parameters are kept constant, the adaptive property of this control scheme comes from the fact that the algorithm relies on the real-time estimation of manipulator and payload parameters. For this reason, this adaptive controller is referred to as an indirect adaptive control law, as opposed to a direct adaptive control law that does not rely on identification or estimation process. Compared to most robot control algorithms in the literature, this indirect adaptive control strategy does not require the measurement or estimation of joint accelerations nor inversion of the identified inertia matrix. Over the years, mainly for these practical advantages, this algorithm has become one of the most widely used robot control law, and it is now often referred to as the Slotine and Li (SLI) algorithm. Omitting the gravitational effects for space applications, the indirect adaptive control law is given by

$$\tau_{SLI} = \hat{M}(q)\ddot{q}_r + \hat{C}(q, \dot{q})\dot{q}_r - K_d s \quad (\text{D.6})$$

where $K_d \in \mathbb{R}^{n \times n}$ is a constant control gain matrix, and where $\hat{M}(q), \hat{C}(q, \dot{q}) \in \mathbb{R}^{n \times n}$ denote respectively the estimated inertia and the centrifugal/Coriolis matrix. To

eliminate any steady-state tracking errors, the desired link position vector $q_d \in \mathbb{R}^n$ is replaced by the *virtual* link trajectory vector $q_r \in \mathbb{R}^n$ given by

$$q_r = q_d - \Lambda \int_0^t (q - q_d) dt \quad (\text{D.7})$$

The idea is to force the steady-state tracking errors to lie on the sliding surface defined by $s = 0$. Accordingly, the time-derivatives of (D.7) in (D.6) are given by

$$\dot{q}_r = \dot{q}_d - \Lambda (q - q_d) \quad (\text{D.8})$$

$$\ddot{q}_r = \ddot{q}_d - \Lambda (\dot{q} - \dot{q}_d) \quad (\text{D.9})$$

Note that the SLI control law (D.6) does not contain a K_p term since the position tracking error $(q - q_d)$ is already included in (D.8). The unknown manipulator parameters are estimated online using the following adaptation law

$$\dot{\hat{a}} = -\Gamma^{-1} \Upsilon^T (q, \dot{q}, q_r, \dot{q}_r) s \quad (\text{D.10})$$

where $a \in \mathbb{R}^m$ denotes the vector containing m unknown manipulator parameters, and $\hat{a} \in \mathbb{R}^m$ is its estimate, $\Gamma \in \mathbb{R}^{m \times m}$ is a symmetric positive definite matrix, usually diagonal, and $\Upsilon \in \mathbb{R}^{n \times m}$ is a matrix, whose definition can be found in Slotine and Li [160].

Practically, the matrices $\hat{M}(q)$ and $\hat{C}(q, \dot{q})$ may be updated at a low rate, whereas a high update rate must be used for \dot{q}_r , \ddot{q}_r and s , since typically, the error terms vary much faster than the system matrices. Further, the adaptation gain matrix Υ , whose calculation is naturally coupled to the dynamics computation, can also be updated at the slow rate.

The time-derivatives of τ_{SLI} in (D.4) and (D.5) are given by

$$\dot{\tau}_{SLI} = M(q) \ddot{q}_r + \left[\dot{M}(q) + C(q, \dot{q}) \right] \dot{q}_r + \dot{C}(q, \dot{q}) \dot{q}_r - K_d \dot{s} \quad (\text{D.11})$$

$$\begin{aligned} \ddot{\tau}_{SLI} = & M(q)\ddot{\ddot{q}}_r + [2\dot{M}(q) + C(q, \dot{q})] \ddot{\ddot{q}}_r + [\ddot{M}(q) + 2\dot{C}(q, \dot{q})] \ddot{q}_r \quad (D.12) \\ & + \ddot{C}(q, \dot{q})\dot{q}_r - K_d\ddot{s} \end{aligned}$$

where

$$\ddot{\ddot{q}}_r = \ddot{\ddot{q}}_d + \Lambda(\ddot{q}_d - \ddot{q}) \quad (D.13)$$

$$\ddot{\ddot{q}}_r = \ddot{\ddot{q}}_d + \Lambda(\ddot{q}_d - \ddot{q}) \quad (D.14)$$

$$\ddot{s} = -(\ddot{q}_d - \ddot{q}) - \Lambda(\ddot{q}_d - \ddot{q}) = \ddot{q} - \ddot{\ddot{q}}_r \quad (D.15)$$

In (D.13) and (D.15), the i, j th element of the space robot inertia and Coriolis/centrifugal matrix first and second derivatives are given by

$$\dot{M}_{11} = -30.5269 \sin(q_2)\dot{q}_2 \quad (D.16)$$

$$\dot{M}_{r12} = \dot{M}_{21} = -15.2634 \sin(q_2)\dot{q}_2 \quad (D.17)$$

$$\dot{M}_{r22} = 0 \quad (D.18)$$

$$\ddot{M}_{r11} = -30.5269 [\cos(q_2)\dot{q}_2^2 + \sin(q_2)\ddot{q}_2] \quad (D.19)$$

$$\ddot{M}_{r12} = \ddot{M}_{21} = -15.2634 [\cos(q_2)\dot{q}_2^2 + \sin(q_2)\ddot{q}_2] \quad (D.20)$$

$$\ddot{M}_{r22} = 0 \quad (D.21)$$

$$\dot{C}_{11} = -15.2634 [\cos(q_2)\dot{q}_2^2 + \sin(q_2)\ddot{q}_2] \quad (\text{D.22})$$

$$\dot{C}_{21} = 15.2634 [\cos(q_2)\dot{q}_1\dot{q}_2 + \sin(q_2)\ddot{q}_1] \quad (\text{D.23})$$

$$\dot{C}_{12} = \dot{C}_{11} - \dot{C}_{21} \quad (\text{D.24})$$

$$\dot{C}_{22} = 0 \quad (\text{D.25})$$

$$\ddot{C}_{11} = -15.2634 [3 \cos(q_2)\ddot{q}_2\dot{q}_2 - \sin(q_2)\dot{q}_2^3 + \sin(q_2)\ddot{\ddot{q}}_2] \quad (\text{D.26})$$

$$\begin{aligned} \ddot{C}_{21} = 15.2634 [\sin(q_2)\ddot{\ddot{q}}_1 + \cos(q_2)\ddot{\ddot{q}}_1 + \cos(q_2)\dot{q}_1\ddot{q}_2 + \cos(q_2)\ddot{q}_1\dot{q}_2 \\ - \sin(q_2)\dot{q}_1\dot{q}_2^2] \end{aligned} \quad (\text{D.27})$$

$$\ddot{C}_{12} = \ddot{C}_{11} - \ddot{C}_{21} \quad (\text{D.28})$$

$$\ddot{C}_{22} = 0 \quad (\text{D.29})$$

To implement this controller, no acceleration or jerk measurements are required. Indeed, the signals \ddot{q} and $\ddot{\ddot{q}}$ can be obtained analytically by inverting the dynamics, that is

$$\ddot{q} = M^{-1}(q) [k(q_m - q) - C(q, \dot{q})\dot{q}] \quad (\text{D.30})$$

Time-differentiating (D.30) yields

$$\ddot{\ddot{q}} = M^{-1}(q) \left\{ k(\dot{q}_m - \dot{q}) - \dot{C}(q, \dot{q}, \ddot{q})\dot{q} - \left[\dot{M}(q, \dot{q}) + C(q, \dot{q}) \right] \ddot{q} \right\} \quad (\text{D.31})$$

D.2 Stability Analysis

Theorem B.1: The flexible-joint space manipulator system defined by (4.3) and (4.4) and the backstepping controller given by (D.1) yield a globally asymptotically stable closed-loop system in the sense of Lyapunov.

Proof: The essential idea of the backstepping technique is to start from the knowledge of a smooth feedback such that the system is globally asymptotically stable. Then, an input error equation is generated and an integrator is added. For the augmented system, a new stabilizing feedback law is explicitly designed and shown to stabilize a new Lyapunov function, and so on. By following this design procedure, the resulting backstepping controller ensures global asymptotic stability of the closed-loop system in the sense of Lyapunov. The design procedure of the nonlinear backstepping controller is taken from Brogliato *et al.* [28] and adapted herein for space-based robots.

To apply the backstepping methodology, the dynamic system has to be expressed as a cascade connection of integrators and link dynamics. This can be achieved with the following feedback control law:

$$\tau = J_m v - k(q - q_m) \quad (\text{D.32})$$

Substituting (D.32) into the flexible-joint dynamics (4.3) and (4.4) gives

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} = k(q_m - q) \quad (\text{D.33})$$

$$\ddot{q}_m = v \quad (\text{D.34})$$

Equations (D.33) and (D.34) represents the manipulator system expressed in a cascade form, with q_m as the input of (D.33). Considering a simple feedback law of the form $q_m = q_{m_d}$, where q_{m_d} is defined in (D.3), the closed-loop link dynamics is obtained by replacing q_m with q_{m_d} in the link dynamics equation, a follows

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} = k(q_{m_d} - q) \quad (\text{D.35})$$

Substituting q_{m_d} with (D.3) in the previous expression yields, after some manipulations

$$M(q)\dot{s} + C(q, \dot{q})s + K_d s = 0 \quad (\text{D.36})$$

Let consider the following Lyapunov function [208]

$$V = \frac{1}{2}s^T M(q)s + \tilde{q}^T \Lambda^T K_d \tilde{q} \quad (\text{D.37})$$

Time-differentiating (D.37) along (D.36) gives

$$\dot{V} = \frac{1}{2}s^T \dot{M}(q)s + s^T [-C(q, \dot{q})s - K_d s] + 2\tilde{q}^T \Lambda^T K_d \dot{\tilde{q}} \quad (\text{D.38})$$

Recalling the skew-symmetric property, the previous equation can be simplified to

$$\dot{V} = -s^T K_d s + 2\tilde{q}^T \Lambda^T K_d \dot{\tilde{q}} \quad (\text{D.39})$$

Replacing s in (D.39) with $\dot{\tilde{q}} + \Lambda \tilde{q}$ yields

$$\dot{V} = -(\dot{\tilde{q}} + \Lambda \tilde{q})^T K_d (\dot{\tilde{q}} + \Lambda \tilde{q}) + 2\tilde{q}^T \Lambda^T K_d \dot{\tilde{q}} \quad (\text{D.40})$$

Expanding the term on the left hand-side gives

$$\dot{V} = -\dot{\tilde{q}}^T K_d \dot{\tilde{q}} - \dot{\tilde{q}}^T K_d \Lambda \tilde{q} - \tilde{q}^T K_d \Lambda \dot{\tilde{q}} - \tilde{q}^T \Lambda^T K_d \Lambda \tilde{q} + 2\tilde{q}^T \Lambda^T K_d \dot{\tilde{q}} \quad (\text{D.41})$$

Cancelling the second and the third terms with the fifth term gives

$$\dot{V} = -\dot{\tilde{q}}^T K_d \dot{\tilde{q}} - \tilde{q}^T \Lambda^T K_d \Lambda \tilde{q} \leq 0 \quad (\text{D.42})$$

Hence, from (D.42), it is clear that the system (D.36) is globally asymptotically stable in the sense of Lyapunov.

Next, an input error equation \tilde{q}_m is generated between q_m and q_{m_d} and an integrator is added at the input. Assuming \dot{q}_m is the new input of the link equation, the augmented system is defined by

$$M(q)\dot{s} + C(q, \dot{q})s + K_d s = k\tilde{q}_m \quad (\text{D.43})$$

$$\dot{\tilde{q}}_m = \dot{q}_m - \dot{q}_{m_d} \quad (\text{D.44})$$

For the augmented system (D.43) and (D.44), let consider the following Lyapunov

function candidate

$$V_2 = V + \frac{1}{2} \tilde{q}_m^T k \tilde{q}_m \quad (\text{D.45})$$

Along the augmented system (D.43) and (D.44), the time-derivative of V_2 is given by

$$\dot{V}_2 = \dot{V} + \tilde{q}_m^T k \dot{\tilde{q}}_m \quad (\text{D.46})$$

$$= \dot{V} + \tilde{q}_m^T k (\dot{q}_m - \dot{q}_{m_d}) \quad (\text{D.47})$$

where \dot{V} is given by (D.42) plus the extra term appearing on the right hand side of (D.43) times s^T . Thus, \dot{V}_2 is given

$$\dot{V}_2 = -\tilde{q}^T K_d \dot{\tilde{q}} - \tilde{q}^T \Lambda^T K_d \Lambda \tilde{q} + s^T k \tilde{q}_m + \tilde{q}_m^T k (\dot{q}_m - \dot{q}_{m_d}) \quad (\text{D.48})$$

By selecting the feedback law $\dot{q}_m = \dot{q}_{m_d} - \tilde{q}_m - s$, \dot{V}_2 becomes

$$\dot{V}_2 = -\tilde{q}^T K_d \dot{\tilde{q}} - \tilde{q}^T \Lambda^T K_d \Lambda \tilde{q} + s^T k \tilde{q}_m - \tilde{q}_m^T k \tilde{q}_m - \tilde{q}_m^T k s \quad (\text{D.49})$$

hence cancelling the term $s^T k \tilde{q}_m$ and ensuring global asymptotic stability of the augmented system.

As before, an input error equation e_m is defined, this time between \dot{q}_m and $\dot{q}_{m_d} - \tilde{q}_m - s$ and an integrator is added at the input. The augmented system then becomes

$$M(q)\dot{s} + C(q, \dot{q})s + K_d s = k \tilde{q}_m \quad (\text{D.50})$$

$$\dot{\tilde{q}}_m \equiv \dot{q}_m - \dot{q}_{m_d} = e_m - \tilde{q}_m - s \quad (\text{D.51})$$

$$\dot{e}_m = \ddot{q}_m - \ddot{q}_{m_d} + \dot{\tilde{q}}_m + \dot{s} \quad (\text{D.52})$$

Recalling (D.34), (D.52) can be rewritten as

$$\dot{e}_m = v - \ddot{q}_{m_d} + \dot{\tilde{q}}_m + \dot{s} \quad (\text{D.53})$$

For the overall system equations (D.50), (D.51) and (D.53), let consider the following

candidate Lyapunov function

$$V = V_2 + \frac{1}{2}e_m^T k e_m \quad (\text{D.54})$$

and the time-derivative of (D.54) along (D.50), (D.51) and (D.53) is given by

$$\dot{V} = \dot{V}_2 + e_m^T k \dot{e}_m \quad (\text{D.55})$$

where \dot{V}_2 is given by (D.48) and \dot{e}_m is given by (D.53)

$$\begin{aligned} \dot{V} = & -\tilde{q}^T K_d \dot{\tilde{q}} - \tilde{q}^T \Lambda^T K_d \Lambda \tilde{q} + s^T k \tilde{q}_m \\ & + \tilde{q}_m^T k (\dot{q}_m - \dot{q}_{m_d}) + e_m^T k (v - \ddot{q}_{m_d} + \dot{\tilde{q}}_m + \dot{s}) \end{aligned} \quad (\text{D.56})$$

Selecting the feedback law v as

$$v = -e_m - \tilde{q}_m + \ddot{q}_{m_d} - \dot{\tilde{q}}_m - \dot{s} \quad (\text{D.57})$$

and inserting back in (D.56) yields

$$\begin{aligned} \dot{V} = & -\tilde{q}^T K_d \dot{\tilde{q}} - \tilde{q}^T \Lambda^T K_d \Lambda \tilde{q} + s^T k \tilde{q}_m + \tilde{q}_m^T k \dot{q}_m \\ & - \tilde{q}_m^T k \dot{q}_{m_d} - e_m^T k e_m - e_m^T k \tilde{q}_m \end{aligned} \quad (\text{D.58})$$

By substituting e_m in the last term on the right hand-side of (D.58) by $\dot{q}_m - \dot{q}_{m_d} + \tilde{q}_m + s$ (see (D.51)), (D.58) can be rewritten as

$$\begin{aligned} \dot{V} = & -\tilde{q}^T K_d \dot{\tilde{q}} - \tilde{q}^T \Lambda^T K_d \Lambda \tilde{q} + s^T k \tilde{q}_m + \tilde{q}_m^T k \dot{q}_m \\ & - \tilde{q}_m^T k \dot{q}_{m_d} - e_m^T k e_m - \dot{q}_m^T k \tilde{q}_m + \dot{q}_{m_d}^T k \tilde{q}_m \\ & - \tilde{q}_m^T k \tilde{q}_m - s^T k \tilde{q}_m \end{aligned} \quad (\text{D.59})$$

Finally, (D.59) can be simplified to

$$\dot{V} = -\tilde{q}^T K_d \dot{\tilde{q}} - \tilde{q}^T \Lambda^T K_d \Lambda \tilde{q} - e_m^T k e_m - \tilde{q}_m^T k \tilde{q}_m \leq 0 \quad (\text{D.60})$$

which establishes the global asymptotic stability of the closed-loop system with the nonlinear backstepping control law obtained by inserting (D.57) in (D.32).

□

Appendix E

Extended Kalman Filter Theory

This appendix provides a brief review of the EKF theory. A more detailed derivation can be found in the work of Lewis [209]. Let a nonlinear system be described by the following dynamics equation

$$x_{k+1} = f(x_k, u_k, w_k) \quad (\text{E.1})$$

and whose observations are described by the nonlinear measurement equation

$$z_k = h(x_k, v_k) \quad (\text{E.2})$$

where $x_k \in \mathbb{R}^n$ is the state vector, $u_k \in \mathbb{R}^p$ is the control input vector, $w_k \in \mathbb{R}^n$ is the process noise vector, and $z_k, v_k \in \mathbb{R}^m$ are the measurement vector and the measurement noise, respectively, defined at a discrete time t_k . It is assumed that w_k and v_k are uncorrelated, zero-mean Gaussian noises with covariance

$$E[w_k w_k^T] = Q_k \in \mathbb{R}^{n \times n} \quad (\text{E.3})$$

$$E[v_k v_k^T] = R_k \in \mathbb{R}^{m \times m} \quad (\text{E.4})$$

where $E[\cdot]$ denotes the expectation. The process noise covariance matrix denoted by Q_k represents the uncertainties in the dynamics model. In other words, Q_k determines the level of confidence the EKF has about the dynamics model. Similarly, the measurement noise covariance matrix denoted by R_k represents the uncertainties in the measurements. The initial mean and covariance of the state vector, denoted by $\hat{x}_0 \in \mathbb{R}^n$ and $P_0 \in \mathbb{R}^{n \times n}$ are respectively given by

$$\hat{x}_0 = E[x_0] \quad (\text{E.5})$$

$$P_0 = E \left[(x_0 - \hat{x}_0) (x_0 - \hat{x}_0)^T \right] \quad (\text{E.6})$$

where x_0 is uncorrelated with w_k and v_k , and where $\hat{\cdot}$ denotes the estimate of a vector.

Given the initial mean and state covariance matrix, and before taking into account any measurements, the estimate of the state vector at time t_k denoted by $\hat{x}_k \in \mathbb{R}^n$ is given by

$$\hat{x}_k = E[x_k] \quad (\text{E.7})$$

This predicted estimate satisfies the following differential equation

$$\dot{\hat{x}}_{k+1} = E[f(x_k, u_k)] = f(\hat{x}_k, u_k) \quad (\text{E.8})$$

which may be integrated to give $\hat{x}_{k+1}^- \in \mathbb{R}^n$, the *a priori*, or the propagated, state vector. In this work, a fifth-order Runge-Kutta integration algorithm has been used for this purpose as this method is known to be more accurate than the fourth order Runge-Kutta algorithm with a slight trade-off in computation speed. Although the propagation of the state vector is done with the exact nonlinear model of the dynamics, the propagation of the state error covariance matrix $P_k \in \mathbb{R}^{n \times n}$ defined by

$$P_k = E \left[(x_k - \hat{x}_k) (x_k - \hat{x}_k)^T \right] \quad (\text{E.9})$$

is done with the discrete-time linearized model of the plant, as follows

$$P_{k+1}^- = F_k P_k F_k^T + Q_k \quad (\text{E.10})$$

where $P_{k+1}^- \in \mathbb{R}^{n \times n}$ is the *a priori*, or propagated state error covariance matrix and where $F_k \in \mathbb{R}^{n \times n}$ is the discrete state transition matrix given by

$$F_k \equiv \Phi_k = \left. \frac{\partial}{\partial x_k} f(x_k, t_k) \right|_{x_{k+1}^-} \quad (\text{E.11})$$

Following a measurement, the propagated state vector is updated to take into account the measurements, as follows

$$x_{x+1}^+ = x_{x+1}^- + K_{k+1} (z_k - \hat{z}_k) \quad (\text{E.12})$$

where $x_{x+1}^+ \in \mathbb{R}^n$ is the *a posteriori*, or the estimated state vector, and where the predicted measurement, denoted by \hat{z}_k , is given by

$$\hat{z}_k = h(x_{x+1}^-) \quad (\text{E.13})$$

In (E.12), $K_{k+1} \in \mathbb{R}^{n \times m}$ denotes the Kalman gain matrix, which is calculated as follows

$$K_{k+1} = P_{k+1}^- H_k^T (H_k P_{k+1}^- H_k^T + R_k)^{-1} \quad (\text{E.14})$$

where $H_k \in \mathbb{R}^{m \times n}$ denotes the linearized measurement model given by

$$H_k = \left. \frac{\partial}{\partial x_k} h(x_k) \right|_{\hat{x}_{k+1}^-} \quad (\text{E.15})$$

Finally, the state error covariance matrix is updated as follows

$$P_{k+1}^+ = P_{k+1}^- - K_{k+1} (H_k P_{k+1}^- H_k^T + R_k) K_{k+1}^T \quad (\text{E.16})$$

In (E.16), the term $(z_k - \hat{z}_k)$ in (E.12) is called the residuals, or innovations. It reflects the degree to which the model fits the data. The complete EKF implementation details are illustrated in Fig. E.1.

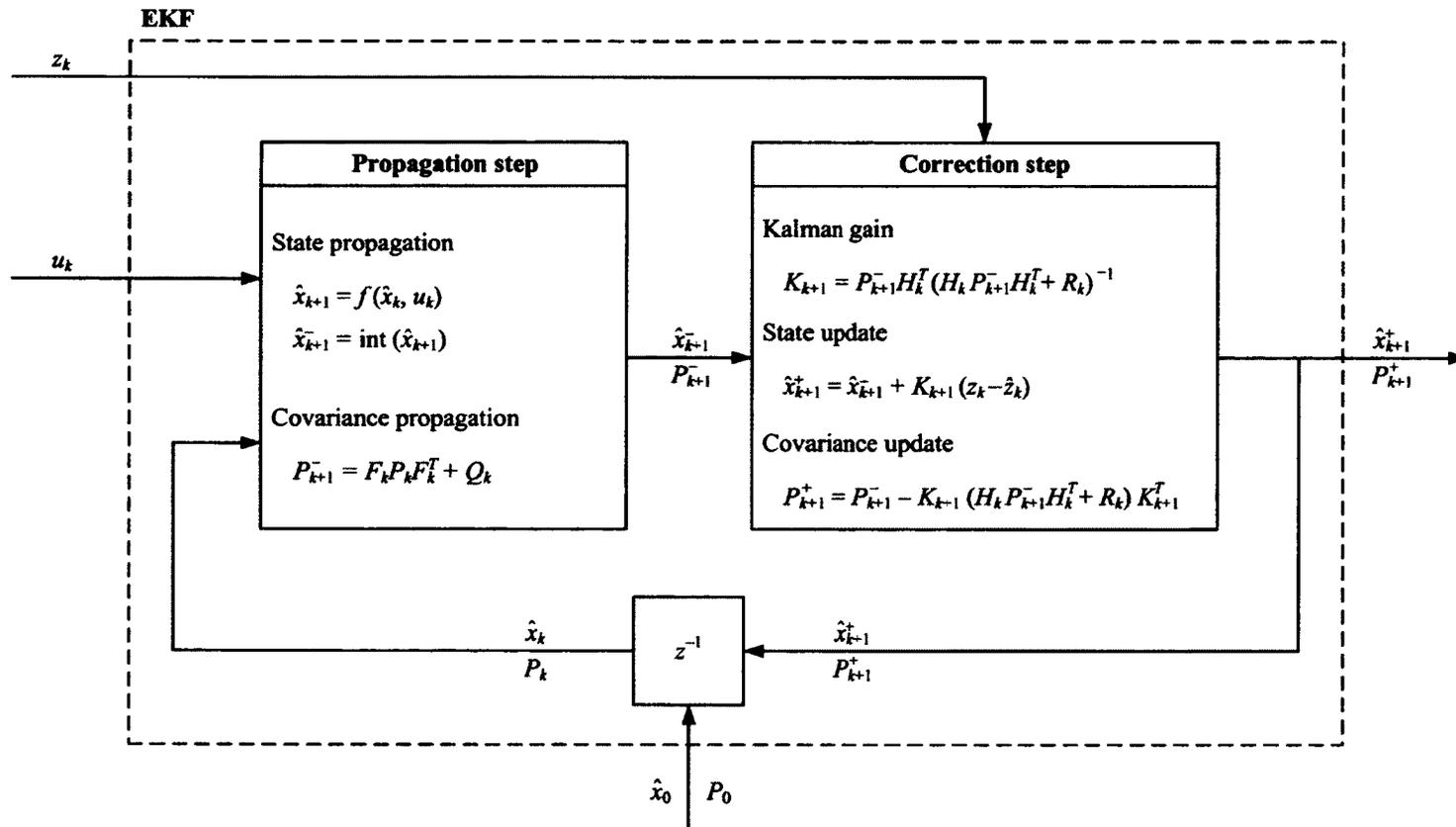


Figure E.1: Implementation details of EKF estimator.

Appendix F

Extended Kalman Filter MATRIXx Code

This appendix includes a sample programming code for calculating the dynamical state estimates using the nonlinear joint stiffness-based EKF.

F.1 State Estimation Function

```
1  #=====
   # State estimation using a nonlinear joint stiffness-based
   # extended Kalman filter (EKF)
   # Written by: Steve Ulrich
   # Copyright(c) 2010 by Steve Ulrich
   #=====

function [y] = NAV_EKF_NL(t,u)

10 #-----
   # Input vector assignation
   #-----

   q_pre = u(1:2);
   q_dot_pre = u(3:4);
   qm_pre = u(5:6);
   qm_dot_pre = u(7:8);
   P_pre = [u(9:16)';u(17:24)';u(25:32)';u(33:40)';u(41:48)';
            u(49:56)';u(57:64)';u(65:72)'];
20 torque_cmd = u(73:74);
   qm_mes = u(75:76);
   qm_dot_mes = u(77:78);
   q_ddot_pre = u(79:80);
   qm_ddot_pre = u(81:82);

   #-----
   # Parameter initialization
   #-----
30   T = NAV_EKFCUB.T;
   Q = NAV_EKFCUB.Q;
   R = NAV_EKFCUB.R;

   # Robot parameters

   m1 = DYN_FLEXCUB.m1;
   m2 = DYN_FLEXCUB.m2;
   l1 = DYN_FLEXCUB.l1;
40   l2 = DYN_FLEXCUB.l2;
   lc1 = DYN_FLEXCUB.lc1;
   lc2 = DYN_FLEXCUB.lc2;
```

```

I1 = DYN_FLEXCUB.I1;
I2 = DYN_FLEXCUB.I2;
ks1 = DYN_FLEXCUB.ks1;
ks2 = DYN_FLEXCUB.ks2;
kw = DYN_FLEXCUB.kw;
aw = DYN_FLEXCUB.aw;
50 a1 = DYN_FLEXCUB.a1;
a2 = DYN_FLEXCUB.a2;
a3 = DYN_FLEXCUB.a3;
a4 = DYN_FLEXCUB.a4;
a5 = DYN_FLEXCUB.a5;
a6 = DYN_FLEXCUB.a6;
Jmw = DYN_FLEXCUB.Jmw;
Jmw_inv = inv(Jmw);
S = DYN_FLEXCUB.S;

60 #-----
# Propagation phase
#-----

# State propagation with nonlinear model using a KMI
# algorithm (Kutta-Merson - a Fifth-Order Runge-Kutta)
state_pre = [q_pre;q_dot_pre;qm_pre;qm_dot_pre];
ddot_pre = [q_ddot_pre;qm_ddot_pre];

70 K0 = NAV_EKF_NL_DOT([state_pre;torque_cmd;ddot_pre]);
K1 = K0*T;
K2 = NAV_EKF_NL_DOT([state_pre+(K1)/3;torque_cmd;
                    ddot_pre])*T;
K3 = NAV_EKF_NL_DOT([state_pre+(K1+K2)/6;torque_cmd;
                    ddot_pre])*T;
K4 = NAV_EKF_NL_DOT([state_pre+(K1+3*K3)/8;torque_cmd;
                    ddot_pre])*T;
K5 = NAV_EKF_NL_DOT([state_pre+(K1+4*K4-3*K3)/2;torque_cmd;
                    ddot_pre])*T;

80 state_minus = state_pre+(K1+4*K4+K5)/6;
q_minus = state_minus(1:2);
q_dot_minus = state_minus(3:4);
qm_minus = state_minus(5:6);
qm_dot_minus = state_minus(7:8);
q_ddot_minus_pre = K0(3:4);
qm_ddot_minus_pre = K0(7:8);

90 # Covariance matrix propagation using the linearized model
# Matrices definition
M11 = m1*lc1^2+m2*(l1^2+lc2^2+2*l1*lc2*cos(q_minus(2)))+I1+I2;
M12 = m2*(lc2^2+l1*lc2*cos(q_minus(2)))+I2;
M21 = m2*(lc2^2+l1*lc2*cos(q_minus(2)))+I2;
M22 = m2*lc2^2+I2;
100 M = [M11, M12; M21, M22];

```

```

M_inv = inv(M);
h = -m2*l1*lc2*sin(q_minus(2));
C11 = h*q_dot_minus(2);
C12 = h*(q_dot_minus(1)+q_dot_minus(2));
C21 = -h*q_dot_minus(1);
C22 = 0;
C = [C11, C12; C21, C22];
110 # Link and motor accelerations
k_minus11 = ks1(1,1)*(qm_minus(1)-q_minus(1))^2 + ks2(1,1)
        -kw*exp(-aw*(qm_minus(1)-q_minus(1))^2);
k_minus22 = ks1(2,2)*(qm_minus(2)-q_minus(2))^2 + ks2(2,2)
        -kw*exp(-aw*(qm_minus(2)-q_minus(2))^2);
k_minus = [k_minus11, 0; 0, k_minus22];
120 fs1 = tanh(a2*q_dot_minus(1))-tanh(a3*q_dot_minus(1));
fs2 = tanh(a2*q_dot_minus(2))-tanh(a3*q_dot_minus(2));
fv1 = a6*q_dot_minus(1);
fv2 = a6*q_dot_minus(2);
fc1 = a4*tanh(a5*q_dot_minus(1));
fc2 = a4*tanh(a5*q_dot_minus(2));
f1 = a1*fs1+fc1+fv1;
f2 = a1*fs2+fc2+fv2;
f_minus = [f1;f2];
130 q_ddot_minus = M_inv * (k_minus*(qm_minus-q_minus) - f_minus
        -C*q_dot_minus - S*qm_ddot_minus_pre);
qm_ddot_minus = Jmw_inv * (torque_cmd
        -k_minus*(qm_minus-q_minus)
        -S'*q_ddot_minus_pre);
# Partial derivatives
dM_dq1_qddot = [0,0;0,0]*q_ddot_minus;
dM_dq2_qddot = h*[2, 1; 1, 0]*q_ddot_minus;
140 dM_dq_qddot = [dM_dq1_qddot, dM_dq2_qddot];
dC_dq1_qdot = [0,0;0,0]*q_dot_minus;
dC_dq2_qdot = -m2*l1*lc2*cos(q_minus(2))*[q_dot_minus(2),
        (q_dot_minus(1)+q_dot_minus(2));
        -q_dot_minus(1), 0]*q_dot_minus;
dC_dq_qdot = [dC_dq1_qdot, dC_dq2_qdot];
150 dC_dq1dot_qdot = h*[0,1;-1,0]*q_dot_minus;
dC_dq2dot_qdot = h*[1,1;0,0]*q_dot_minus;
dC_dqdot_qdot = [dC_dq1dot_qdot, dC_dq2dot_qdot];
dK_dq1_delta = [-2*ks1(1,1)*(qm_minus(1)-q_minus(1))
        -2*kw*aw*(qm_minus(1)-q_minus(1))
        *exp(-aw*(qm_minus(1)-q_minus(1))^2),0;0,0]
160 *(qm_minus-q_minus);

```

```

dK_dq2_delta = [0,0;0,-2*ks1(2,2)*(qm_minus(2)-q_minus(2))
                -2*kw*aw*(qm_minus(2)-q_minus(2))
                *exp(-aw*(qm_minus(2)-q_minus(2))^2)]
                *(qm_minus-q_minus);

dK_dq_delta = [dK_dq1_delta, dK_dq2_delta];

170 dK_dqm1_delta = [2*ks1(1,1)*(qm_minus(1)-q_minus(1))
                  +2*kw*aw*(qm_minus(1)-q_minus(1))
                  *exp(-aw*(qm_minus(1)-q_minus(1))^2),0;0,0]
                  *(qm_minus-q_minus);

dK_dqm2_delta = [0,0;0,2*ks1(2,2)*(qm_minus(2)-q_minus(2))
                  +2*kw*aw*(qm_minus(2)-q_minus(2))
                  *exp(-aw*(qm_minus(2)-q_minus(2))^2)]
                  *(qm_minus-q_minus);

180 dK_dqm_delta = [dK_dqm1_delta, dK_dqm2_delta];

df_dq1dot = [a1*(a2*sech(a2*q_dot_minus(1))^2
              -a3*sech(a3*q_dot_minus(1))^2
              +a4*a5*sech(a5*q_dot_minus(1))^2 + a6; 0];

df_dq2dot   = [0; a1*(a2*sech(a2*q_dot_minus(2))^2
              -a3*sech(a3*q_dot_minus(2))^2
              +a4*a5*sech(a5*q_dot_minus(2))^2 + a6];

df_dqdot = [df_dq1dot, df_dq2dot];

190 # State matrix

F11 = zeros(2,2);
F12 = eye(2,2);
F13 = zeros(2,2);
F14 = zeros(2,2);

F21 = -M_inv*(dM_dq_qddot+dC_dq_qdot-dK_dq_delta+k_minus);
F22 = -M_inv*(dC_dqdot_qdot+C+df_dqdot);
200 F23 = M_inv*(k_minus + dK_dqm_delta);
F24 = zeros(2,2);

F31 = zeros(2,2);
F32 = zeros(2,2);
F33 = zeros(2,2);
F34 = eye(2,2);

F41 = -Jmw_inv*(dK_dq_delta - k_minus);
F42 = zeros(2,2);
210 F43 = -Jmw_inv*(dK_dqm_delta + k_minus);
F44 = zeros(2,2);

F = [F11, F12, F13, F14;
     F21, F22, F23, F24;
     F31, F32, F33, F34;
     F41, F42, F43, F44];

phi = (F*T)*(((F*T)/2)*(((F*T)/3)
             *((F*T)/4 + eye(8,8))+eye(8,8))+eye(8,8);

220 P_minus = phi*P_pre*phi'+ Q;

```

```

#-----
# Correction phase
#-----

# Linearized measurement model
230   H11 = zeros(2,2);
      H12 = zeros(2,2);
      H13 = eye(2,2);
      H14 = zeros(2,2);

      H21 = zeros(2,2);
      H22 = zeros(2,2);
      H23 = zeros(2,2);
      H24 = eye(2,2);
240   H = [H11, H12, H13, H14; H21, H22, H23, H24];

# Kalman gain
      K = (P_minus*H')*inv(H*P_minus*H' + R);

# Previous states
      states_minus = [q_minus;q_dot_minus;qm_minus;qm_dot_minus];
250 # Measured outputs
      obs_mes = [qm_mes;qm_dot_mes];

# Estimated outputs
      obs_est = H*states_minus;

# State correction
260   states_plus = states_minus + K*(obs_mes-obs_est);

      q_plus = states_plus(1:2,:);
      q_dot_plus = states_plus(3:4,:);
      qm_plus = states_plus(5:6,:);
      qm_dot_plus = states_plus(7:8,:);

# Covariance matrix correction
270   P_plus = (eye(8,8) - K*H) * P_minus;

#-----
# Output vector preparation
#-----

      P_plus = [P_plus(1,:),P_plus(2,:),P_plus(3,:),P_plus(4,:),
                P_plus(5,:),P_plus(6,:),P_plus(7,:),P_plus(8,:)];
280   q_ddot_plus = q_ddot_minus;
      qm_ddot_plus = qm_ddot_minus;

      y = [q_plus',q_dot_plus',qm_plus',qm_dot_plus',
           P_plus,obs_est(1),obs_est(2),obs_est(3),obs_est(4),
           q_ddot_plus',qm_ddot_plus'];

endfunction

```

F.2 State Derivative Calculation Function

```

1  #=====
   # State derivative calculation
   # Written by: Steve Ulrich
   # Copyright(c) 2010 by Steve Ulrich
   #=====

function [minus] = NAV_EKF_NL_DOT(pre)

10 #-----
   # Input vector assignation
   #-----

   q_pre = pre(1:2);
   q_dot_pre = pre(3:4);
   qm_pre = pre(5:6);
   qm_dot_pre = pre(7:8);
   torque_cmd = pre(9:10);
   q_ddot_pre = pre(11:12);
20  qm_ddot_pre = pre(13:14);

   #-----
   # Parameter initialization
   #-----

   # Robot parameters

   m1 = DYN_FLEXCUB.m1;
   m2 = DYN_FLEXCUB.m2;
30  l1 = DYN_FLEXCUB.l1;
   l2 = DYN_FLEXCUB.l2;
   lc1 = DYN_FLEXCUB.lc1;
   lc2 = DYN_FLEXCUB.lc2;
   I1 = DYN_FLEXCUB.I1;
   I2 = DYN_FLEXCUB.I2;
   ks1 = DYN_FLEXCUB.ks1;
   ks2 = DYN_FLEXCUB.ks2;
   kw = DYN_FLEXCUB.kw;
40  aw = DYN_FLEXCUB.aw;
   a1 = DYN_FLEXCUB.a1;
   a2 = DYN_FLEXCUB.a2;
   a3 = DYN_FLEXCUB.a3;
   a4 = DYN_FLEXCUB.a4;
   a5 = DYN_FLEXCUB.a5;
   a6 = DYN_FLEXCUB.a6;
   Jmw = DYN_FLEXCUB.Jmw;
   Jmw_inv = inv(Jmw);
   S = DYN_FLEXCUB.S;

50 #-----
   # Dynamics equations
   #-----

```

```

# Matrices definition
M11 = m1*lc1^2+m2*(l1^2+lc2^2+2*l1*lc2*cos(q_pre(2)))+I1+I2;
M12 = m2*(lc2^2+l1*lc2*cos(q_pre(2)))+I2;
M21 = m2*(lc2^2+l1*lc2*cos(q_pre(2)))+I2;
60 M22 = m2*lc2^2+I2;
M = [M11, M12; M21, M22];
M_inv = inv(M);

h = -m2*l1*lc2*sin(q_pre(2));

C11 = h*q_dot_pre(2);
C12 = h*(q_dot_pre(1)+q_dot_pre(2));
C21 = -h*q_dot_pre(1);
C22 = 0;
70 C = [C11, C12; C21, C22];

# Nonlinear stiffness and friction
k_pre11 = ks1(1,1)*(qm_pre(1)-q_pre(1))^2 + ks2(1,1)
          -kw*exp(-aw*(qm_pre(1)-q_pre(1))^2);
k_pre22 = ks1(2,2)*(qm_pre(2)-q_pre(2))^2 + ks2(2,2)
          - kw*exp(-aw*(qm_pre(2)-q_pre(2))^2);
k_pre = [k_pre11, 0; 0, k_pre22];
80 fs1 = tanh(a2*q_dot_pre(1))-tanh(a3*q_dot_pre(1));
fs2 = tanh(a2*q_dot_pre(2))-tanh(a3*q_dot_pre(2));
fv1 = a6*q_dot_pre(1);
fv2 = a6*q_dot_pre(2);
fc1 = a4*tanh(a5*q_dot_pre(1));
fc2 = a4*tanh(a5*q_dot_pre(2));
f1 = a1*fs1+fc1+fv1;
f2 = a1*fs2+fc2+fv2;
f_pre = [f1;f2];

90 # Derivatives of state variables
q_dot = q_dot_pre;
q_ddot = M_inv * (k_pre*(qm_pre-q_pre)
                - f_pre - C*q_dot_pre - S*qm_ddot_pre);

qm_dot = qm_dot_pre;
qm_ddot = Jmw_inv * (torque_cmd
                   - k_pre*(qm_pre-q_pre) - S'*qm_ddot_pre);
100 minus = [q_dot;q_ddot;qm_dot;qm_ddot];
endfunction

```

Bibliography

- [1] Salaberger, C., "Canadian Space Robotic Activities," *Acta Astronautica*, Vol. 41, No. 4, 1998, pp. 239–246.
- [2] Bekey, G., Ambrose, R., Kumar, V., Lavery, D., Sanderson, A., Wilcox, B., Yuh, J., and Zheng, Y., *Robotics: State of the Art and Future Challenges*, Imperial College Press, 2008.
- [3] Peuter, W. D., "Development Program on Space Automation and Robotics," *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Toulouse, France, Sep 1992.
- [4] Acquatella, P., "Development of Automation and Robotics in Space Exploration," *4th Annual International Planetary Probe Workshop*, Pasadena, CA, JUN 2006.
- [5] King, D., "On-orbit Servicing Business; an Industry Vision," *1st Bilateral DLR-CSA Workshop on On-Orbit Servicing of Space Infrastructure Elements via Automation and Robotics Technologies*, Cologne, Germany, NOV 2002.
- [6] Currie, N. J. and Peacock, B., "International Space Station Robotic Systems Operations - A Human Factors Perspective," Tech. rep., Lyndon B. Johnson Space Center, Habitability and Environmental Factors Office, Houston, Texas, 2005.
- [7] Reintsema, D., Landzettel, K., and Hirzinger, G., "DLRs Advanced Telerobotic Concepts and Experiments for On-Orbit Servicing," *Advances in Telerobotics*, edited by M. Ferre, M. Buss, R. Aracil, C. Melchiorri, and C. Balaguer, Vol. 31 of *Springer Tracts on Advanced Robotics*, Springer-Verlag Berlin Heidelberg, 2007, pp. 323–347.
- [8] Sasiadek, J. Z. and Srinivasan, R., "Dynamic Modeling and Adaptive Control of a Single-Link Flexible Manipulator," *Journal of Guidance, Dynamics, and Control*, Vol. 12, No. 6, 1989, pp. 834–844.
- [9] Hecht, N. K. and Junkins, J. L., "Near-Minimum-Time Control of a Flexible Manipulator," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 2, 1992, pp. 477–481.
- [10] de Luca, A. and Siciliano, B., "Inversion-Based Nonlinear Control of Robot Arms with Flexible Links," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 6, 1993, pp. 1169–1176.

- [11] Carusone, J., Buchan, K. S., and D'Eleuterio, G. M. T., "Experiments in End-Effector Tracking Control for Structurally Flexible Space Manipulators," *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 5, 1993, pp. 553–560.
- [12] Banerjee, A. K. and Singhose, W. E., "Command Shaping in Tracking Control of a Two-Link Flexible Robot," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 6, 1998, pp. 1012–1015.
- [13] Damaren, C. J., "Modal Properties and Control System Design for Two-Link Flexible Manipulators," *International Journal of Robotics Research*, Vol. 17, No. 6, 1998, pp. 667–678.
- [14] Romano, M., Agrawal, B. N., and Bernelli-Zazerra, F., "Experiments on Command Shaping Control of a Manipulator with Flexible Links," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 2, 2002, pp. 232–239.
- [15] Green, A. and Sasiadek, J. Z., "Adaptive Control of a Flexible Robot Using Fuzzy Logic," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 1, 2005, pp. 36–42.
- [16] Sweet, L. M. and Good, M. C., "Re-definition of the Robot Motion Control Problem: Effects of Plant Dynamics, Drive System Constraints, and User Requirements," *23rd IEEE Conference on Decision and Control*, Las Vegas, NV, JUN 1984.
- [17] van Woerkoma, P. T. L. M. and Misrab, A. K., "Robotic Manipulators in Space: A Dynamics and Control Perspective," *Acta Astronautica*, Vol. 38, No. 4-8, 1996, pp. 411–421.
- [18] Spong, M. W., Hutchinson, S., and Vidyasagar, M., *Robot Modeling and Control*, John Wiley and Sons, 2006.
- [19] Kahraman, A. and Vijayakar, S., "Effect of Internal Gear Flexibility on the Quasi-Static Behavior of a Planetary Gear Set," *Journal of Mechanical Design*, Vol. 123, No. 3, 2001, pp. 408–415.
- [20] Chen, T., Jin, J., and Wu, X., "Analysis of Dynamical Behavior of a Planetary Gear Train," *Intelligent Robotics and Applications*, edited by C. X. et al., Series on Lecture Notes in Computer Science, Springer Berlin, 2008, pp. 46–53.
- [21] Sasiadek, J. Z., "Space Robotics and Manipulators - The Past and the Future," *Control Engineering Practice*, Vol. 2, No. 3, 1994, pp. 491–497.
- [22] Spong, M. W., Khorasani, K., and Kokotovic, P. V., "An Integral Manifold Approach to the Feedback Control of Flexible Joint Robots," *IEEE Journal of Robotics and Automation*, Vol. 3, No. 4, 1987, pp. 291–300.

- [23] Cetinkunt, S. and Book, W. J., "Flexibility Effects on the Control System Performance of Large Scale Robotic Manipulators," *The Journal of the Astronautical Sciences*, Vol. 38, No. 4, 1990, pp. 531–556.
- [24] Ghorbel, F., Hung, J. Y., and Spong, M. W., "Adaptive Control of Flexible Joint Manipulators," *IEEE Control Systems Magazine*, Vol. 9, No. 7, 1989, pp. 9–13.
- [25] Sasiadek, J. Z., "Space Robotics and Manipulators: Lessons Learned from the Past and Future Missions and Systems," *12th International IFAC Symposium on Automatic Control in Aerospace*, Ottobrunn, Germany, SEP 1992.
- [26] Albu-Schaffer, A., Ott, C., and Hirzinger, G., "A Unified Passivity-based Control Framework for Position, Torque and Impedance Control of Flexible Joint Robots," *International Journal of Robotics Research*, Vol. 26, No. 1, 2007, pp. 23–39.
- [27] Bridges, M. M., Dawson, D. M., and Abdallah, C. T., "Control of Rigid-Link, Flexible-Joint Robots: A Survey on Backstepping Approaches," *Journal of Robotics Systems*, Vol. 12, No. 3, 1995, pp. 199–216.
- [28] Brogliato, B., Ortega, R., and Lozano, R., "Global Tracking Controllers for Flexible Joint Manipulators: A Comparative Study," *Automatica*, Vol. 31, No. 7, 1995, pp. 941–956.
- [29] Troch, I. and Kopacek, P., "Control Concepts and Algorithms for Flexible Robots: An Expository Survey," *Symposium on Robot Control*, Karlsruhe, Germany, OCT 1988.
- [30] Spong, M. W., *New Trends and Applications of Distributed Parameter Control Systems*, chap. The Control of Flexible Joint Robots: A Survey, Marcel Dekker, New York, 1990.
- [31] Ozgoli, S. and Taghirad, H. D., "A Survey on the Control of Flexible Joint Robots," *Asian Journal of Control*, Vol. 8, No. 4, 2006, pp. 332–344.
- [32] Tomei, P., Nicosia, S., and Ficola, A., "An Approach to the Adaptive Control of Elastic at Joints Robots," *1986 IEEE International Conference on Robotics and Automation*, San Fransisco, CA, APR 1986.
- [33] Kuntze, H. B., Jacubasch, A., Richalet, J., and Arber, C., "On the Predictive Functional Control of an Elastic Industrial Robot," *IEEE Conference on Decision and Control*, Athens, Greece, DEC 1986.
- [34] Marino, R. and Spong, M. W., "Nonlinear Control Techniques for Flexible Joint Manipulators: A single link case study," *IEEE International Conference on Robotics and Automation*, San Fransisco, CA, APR 1986.

- [35] Tomei, P., "A Simple PD Controller for Robots with Elastic Joints," *IEEE Transactions on Automatic Control*, Vol. 36, No. 10, 1991, pp. 1208–1213.
- [36] Tomei, P., "Adaptive PD Controller for Robot Manipulators," *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 4, 1991, pp. 565–570.
- [37] Lozano, R., Valera, A., Albertos, P., Arimoto, S., and Nakayama, T., "PD Control of Robot Manipulators with Joint Flexibility, Actuators Dynamics and Friction," *Automatica*, Vol. 35, No. 10, 1999, pp. 1697–1700.
- [38] de Luca, A., Siciliano, B., and Zollo, L., "PD Control with On-Line Gravity Compensation for Robots with Elastic Joints: Theory and Experiments," *Automatica*, Vol. 41, No. 10, 2005, pp. 1809–1819.
- [39] Chow, J. H. and Kokotovic, P. V., "Two Time Scale Feedback Design of a Class of Nonlinear Systems," *IEEE Transactions on Automatic Control*, Vol. 23, No. 3, 1978, pp. 438–443.
- [40] Khorasani, K. and Spong, M. W., "Invariant Manifolds and their Application to Robot Manipulators with Flexible Joints," *IEEE International Conference on Robotics and Automation*, St-Louis, Missouri, MAR 1985.
- [41] Khorasani, K. and Kokotovic, P. V., "Feedback Linearization of a Flexible Manipulator Near its Rigid Body Manifold," *Systems and Control Letters*, Vol. 6, No. 3, 1985, pp. 187–192.
- [42] Spong, M. W., Khorasani, K., and Kokotovic, P. V., "A Slow Manifold Approach to Feedback Control of Nonlinear Flexible Systems," *American Control Conference*, Boston, MA, JUN 1985.
- [43] Ghorbel, F. and Spong, M. W., "Adaptive Integral Manifold Control of Flexible Joint Robot Manipulators," *IEEE International Conference on Robotics and Automation*, Nice, France, MAY 1992.
- [44] Cesareo, G. and Marino, R., "On the Controllability Properties of Elastic Robots," *International Conference on Analytical and Optimization Systems*, Nice, France, JUN 1984.
- [45] Spong, M. W., "Modeling and Control of Elastic Joint Robots," *Journal of Dynamic Systems, Measurement and Control*, Vol. 109, No. 4, 1987, pp. 310–319.
- [46] Nicosia, S. and Tomei, P., "On the Feedback Linearization of Robots with Elastic Joints," *27th IEEE Conference on Decision and Control*, Austin, Texas, DEC 1988.

- [47] de Luca, A., "Feedforward/Feedback Laws for the Control of Flexible Robots," *IEEE International Conference on Robotics and Automation*, San Francisco, CA, APR 2000.
- [48] Lin, L.-C. and Yuan, K., "Control of Flexible Joint Robots via External Linearization Approach," *Journal of Robotic Systems*, Vol. 7, No. 1, 1990, pp. 1–22.
- [49] de Luca, A., "Dynamic Control of Robots with Joint Elasticity," *IEEE International Conference on Robotics and Automation*, Philadelphia, PA, APR 1988.
- [50] de Luca, A. and Lucibello, P., "A General Algorithm for Dynamic Feedback Linearization of Robots with Elastic Joints," *IEEE International Conference on Robotics and Automation*, Leuven, Belgium, JUN 1998.
- [51] Ge, S. S., Sun, Z., and Lee, T. H., "Nonregular Feedback Linearization for a Class of Second-Order Nonlinear Systems," *Automatica*, Vol. 31, No. 11, 2001, pp. 1819–1824.
- [52] Luca, A. D., "Decoupling and Feedback Linearization of Robots with Mixed Elastic/Rigid Joints," *International Journal of Robust and Nonlinear Control*, Vol. 8, No. 5, 1998, pp. 965–977.
- [53] de Luca, A., Farina, R., and Lucibello, P., "On the Control of Robots with Visco-Elastic Joints," *IEEE International Conference on Robotics and Automation*, Barcelona, Spain, APR 2005.
- [54] Palli, G., Melchiorri, C., and de Luca, A., "On the Feedback Linearization of Robots with Variable Joint Stiffness," *IEEE International Conference on Robotics and Automation*, Pasadena, CA, MAY 2008.
- [55] Lahdhiri, T. and ElMaraghy, H. A., "Optimal Position Controller of a Two-Link Flexible Joints Robot Manipulator," *American Control Conference*, Philadelphia, PA, JUN 1998.
- [56] Consolini, L., Gereli, O., Bianco, C. G. L., and Piazzi, A., "Minimum-Time Control of Flexible Joints with Input and Output Constraints," *IEEE International Conference on Robotics and Automation*, Rome, Italy, APR 2007.
- [57] Merabet, A. and Gu, J., "Robust Nonlinear Predictive Control with Modeling Uncertainties and Unknown Disturbance for Single-Link Flexible Joint Robot," *7th World Congress on Intelligent Control and Automation*, Chongqing, China, JUN 2008.
- [58] Mrad, F. and Ahmad, S., "Adaptive Control of Flexible Joint Robots with Stability in the Sense of Lyapounov," *29th IEEE Conference on Decision and Control*, Honolulu, HI, DEC 1990.

- [59] Lozano, R. and Brogliato, B., "Adaptive control of robot manipulators with flexible joints," *IEEE Transactions on Automatic Control*, Vol. 37, No. 2, 1992, pp. 174–181.
- [60] Yuan, J. and Stepanenko, Y., "Adaptive Control of Flexible Joint Robots with Arbitrary Stiffness," *Control Engineering Practice*, Vol. 2, No. 5, 1994, pp. 903–906.
- [61] Nicosia, S. and Tomei, P., "Design of Global Tracking Controllers for Flexible-Joint Robots," *Journal of Robotic Systems*, Vol. 10, No. 6, 1993, pp. 835–846.
- [62] Kim, M. S. and Lee, J. S., "Adaptive Tracking Control of Flexible-Joint Manipulators without Overparametrization," *Journal of Robotic Systems*, Vol. 21, No. 7, 2004, pp. 369–379.
- [63] Yim, W., "Adaptive Control of a Flexible Joint Manipulator," *IEEE International Conference on Robotics and Automation*, Seoul, Korea, MAY 2001.
- [64] Fu, S. H.-S. and Cheng, C.-C., "Robust Direct Adaptive Control of Nonlinear Uncertain Systems with Unknown Disturbances," *2005 American Control Conference*, Portland, Oregon, JUN 2005.
- [65] Liu, C., Cheah, C. C., and Slotine, J.-J. E., "Adaptive Task-space Regulation of Rigid-Link Flexible-Joint Robots with Uncertain Kinematics," *Automatica*, Vol. 44, No. 7, 2008, pp. 1806–1814.
- [66] Tomei, P., "Tracking Control of Flexible Joint Robots with Uncertain Parameters and Disturbances," *IEEE Transactions on Automatic Control*, Vol. 39, No. 5, 1994, pp. 1067–1072.
- [67] Dawson, D. M., Qu, Z., Bridges, M., and Carroll, J., "Robust Tracking of Rigid-link Flexible-Joint Electrically-Driven Robots," *30th IEEE Conference on Decision and Control*, Brighton, England, DEC 1991.
- [68] Jain, S. and Khorrami, F., "Robust Adaptive Control of Flexible Joint Manipulators," *Automatica*, Vol. 34, No. 5, 1998, pp. 609–615.
- [69] Elmaraghy, H. A., Lahdhiri, T., and Ciuca, F., "Robust Linear Control of Flexible Joint Robot Systems," *Journal of Intelligent and Robotic Systems*, Vol. 34, No. 4, 2002, pp. 335–356.
- [70] Taghirad, H. D. and Rahimi, H., "Composite QFT Controller Design For Flexible Joint Robots," *2005 IEEE Conference on Control Applications*, Toronto, Canada, AUG 2005.
- [71] Taghirad, H. D. and Shaterian, M., "Nonlinear H-infinity Controller Design for Flexible Joint Robots," *IEEE Conference on Control Applications*, Toronto, Canada, AUG 2005.

- [72] Moghaddam, M. M., *Robust H Infinity-Based Control of Flexible Joint Robots With Harmonic Drive Transmission*, Ph.D. thesis, Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Ontario, 1997.
- [73] Yim, J.-G., Yeon, J. S., Lee, J., Park, J. H., Lee, S.-H., and Hur, J.-S., "Robust Control of Flexible Robot Manipulators," *SICE-ICASE International Joint Conference*, Bexco, Busan, Korea, OCT 2006.
- [74] Yim, J.-G., Yeon, J. S., Park, J. H., Lee, S.-H., and Hur, J.-S., "Robust Control using Recursive Design Method for Flexible Joint Robot Manipulator," *2007 IEEE International Conference on Robotics and Automation*, Rome, Italy, APR 2007.
- [75] Lee, J. Y., Yeon, J. S., and Park, J. H., "Robust Nonlinear Control for Flexible Joint Robot Manipulators," *SICE Annual Conference 2007*, Kagawa, Japan, SEP 2007.
- [76] Yeon, J. S. and Park, J. H., "Practical Robust Control for Flexible Joint Robot Manipulators," *2008 IEEE International Conference on Robotics and Automation*, Pasadena, CA, MAY 2008.
- [77] de Luca, A., Isidori, A., and Nicolo, F., "Control of Robot Arm with Elastic Joints via Nonlinear Dynamic Feedback," *24th IEEE Conference on Decision and Control*, Fort Lauderdale, FL, JUN 1985.
- [78] Khorasani, K., "Nonlinear Feedback Control of Flexible Joint Manipulators: A Single Link Case Study," *IEEE Transactions on Automatic Control*, Vol. 35, No. 10, 1990, pp. 1145–1149.
- [79] Oh, J. H. and Lee, J. S., "Control of Flexible Joint Robot System by Back Stepping Design Approach," *Intelligent Automation and Software Computing*, Vol. 5, No. 4, 1999, pp. 267–278.
- [80] Macnab, C. J. B., D'Eleuterio, G. M. T., and Meng, M., "CMAC Adaptive Control of Flexible-Joint Robots Using Backstepping with Tuning Functions," *2004 IEEE International Conference on Robotics and Automation*, New Orleans, Louisiana, APR 2004.
- [81] Lee, J. Y., Yeon, J. S., Park, J. H., and Lee, S., "Robust Back-Stepping Control for Flexible-Joint Robot Manipulators," *2007 IEEE International Conference on Intelligent Robots and Systems*, San Diego, CA, NOV 2007.
- [82] Abouelsoud, A. A., "Robust Regulator for Flexible-Joint Robots Using Integrator Backstepping," *Journal of Intelligent and Robotic Systems*, Vol. 22, No. 1, 1998, pp. 23–38.

- [83] Nicosia, S. and Tomei, P., "A New Approach to Control Elastic Joint Robots with Application to Adaptive Control," *30th IEEE Conference on Decision and Control*, Brighton, England, DEC 1991.
- [84] Brogliato, B., Rey, D., Pastore, A., and Barnier, J., "Experimental Comparison of Nonlinear Controllers for Flexible Joint Manipulators," *The International Journal of Robotics Research*, Vol. 17, No. 3, 1998, pp. 260–281.
- [85] Brogliato, B. and Rey, D., "Further Experimental Results on Nonlinear Control of Flexible Joint Manipulators," *1998 American Control Conference*, Philadelphia, PA, JUN 1998.
- [86] Wang, D., "A Simple Iterative Learning Controller for Manipulators with Flexible Joints," *Automatica*, Vol. 31, No. 9, 1995, pp. 1341–1344.
- [87] Fu, J. and Sinha, N. K., "Iterative Learning Control of Flexible-Joint Robots Using Neural Networks," *Control Engineering Practice*, Vol. 1, No. 5, 1993, pp. 882–883.
- [88] Albu-Schaffer, A. and Hirzinger, G., "A Globally Stable State Feedback Controller for Flexible Joint Robots," *Journal of Advanced Robotics, Special Issue: Selected papers from IROS 2000*, Vol. 15, No. 8, 2001, pp. 799–814.
- [89] Craig, J. J., *Introduction to Robotics: Mechanics and Control*, Prentice Hall, 3rd ed., 2005.
- [90] Passino, K. M. and Yurkovich, S., *Fuzzy Control*, Addison-Wesley, Menlo Park, CA, 1998.
- [91] Kaufman, H., Barkana, I., and Sobel, K., *Direct Adaptive Control Algorithms: Theory and Applications*, Communications and Control Engineering, Springer, 2nd ed., 1997.
- [92] Hidaka, T., Ishida, T., Zhang, Y., Sasahara, M., and Tanioka, Y., "Vibration of a Strain-Wave Gearing in an Industrial Robot," *ASME International Power Transmission and Gearing Conference*, 1990, pp. 789–794.
- [93] Tuttle, T. D. and Seering, W. P., "Modeling a Harmonic Drive Gear Transmission," *IEEE International Conference on Robotics and Automation*, Atlanta, Georgia, MAY 1993.
- [94] Tuttle, T. D. and Seering, W. P., "A Nonlinear Model of a Harmonic Drive Gear Transmission," *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 3, 1996, pp. 368–374.

- [95] Kircanski, N. M. and Goldenberg, A. A., "An Experimental Study of Nonlinear Stiffness, Hysteresis, and Friction Effects in Robot Joints with Harmonic Drives and Torque Sensors," *The International Journal of Robotics Research*, Vol. 16, No. 2, 1997, pp. 214–239.
- [96] Taghirad, H. D. and Belanger, P. R., "Modeling and Parameter Identification of Harmonic Drive Systems," *Journal of Dynamic Systems, Measurement and Control*, Vol. 120, No. 4, 1998, pp. 439–444.
- [97] Li, Z., Melek, W. W., and Clark, C., "Decentralized Robust Control of Robot Manipulators with Harmonic Drive Transmission and Application to Modular and Reconfigurable Serial Arms," *Robotica*, Vol. 27, 2009, pp. 291–302.
- [98] Alexander, H. L. and Cannon, R. H., "An Extended Operational-Space Control Algorithm for Satellite Manipulators," *The Journal of the Astronautical Sciences*, Vol. 38, No. 4, 1990, pp. 473–486.
- [99] Green, A. and Sasiadek, J. Z., "Heuristic Design of a Fuzzy Controller for a Flexible Robot," *IEEE Transactions on Control Systems Technology*, Vol. 14, No. 2, 2006, pp. 293–300.
- [100] de Silva, C. W., *Intelligent Control: Fuzzy Logic Applications*, CRC Press, 1995.
- [101] Yi, S. Y. and Chung, M. J., "A Robust Fuzzy Logic Controller for Robot Manipulators with Uncertainties," *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, Vol. 27, No. 4, 1997, pp. 706–713.
- [102] Green, A. and Sasiadek, J. Z., "Robot Manipulator Control for Rigid and Assumed Mode Flexible Dynamics Models," *AIAA Guidance, Navigation and Control Conference and Exhibit*, Austin, TX, AUG 2003.
- [103] Santibanez, V., Kelly, R., and Llama, M. A., "Global Asymptotic Stability of a Tracking Sectorial Fuzzy Controller for Robot Manipulators," *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, Vol. 34, No. 1, 2004, pp. 710–718.
- [104] Llama, M. A., Kelly, R., and Santibanez, V., "Stable Computed-Torque Control of Robot Manipulators via Fuzzy Self-Tuning," *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, Vol. 30, No. 1, 2000, pp. 143–150.
- [105] Ham, C., Qu, Z., and Johnson, R., "Robust Fuzzy Control for Robot Manipulators," *IEE Proceedings Control Theory and Applications*, Vol. 147, No. 2, 2009, pp. 212–216.
- [106] Islam, S. and Liu, P. X., "Robust Adaptive Fuzzy Output Feedback Control System for Robot Manipulators," *IEEE/ASME Transactions on Mechatronics*, Vol. 16, No. 2, 2011, pp. 288–296.

- [107] Brady, M., Hollerbach, J. M., Johnson, T. L., and Loranzo-Perez, T., *Robot Motion: Planning and Control*, MIT Press, 1984.
- [108] Asada, H. and Slotine, J. J. E., *Robot Analysis and Control*, John Wiley and Sons, 1986.
- [109] Sciavicco, L. and Siciliano, B., *Modelling and Control of Robot Manipulators*, Advanced Textbooks in Control and Signal Processing, Springer, 2nd ed., 2001.
- [110] Ceccarelli, M., *Fundamentals of Mechanics of Robotic Manipulation*, Intelligent Systems, Control and Automation: Science and Engineering, Springer, 2004.
- [111] Green, A. and Sasiadek, J. Z., "Dynamics and Trajectory Tracking Control of a Two-Link Robot Manipulator," *Journal of Vibration and Control*, Vol. 10, No. 10, 2004, pp. 1415–1440.
- [112] Passino, K. M. and Yurkovich, S., *Fuzzy Control*, AddisonWesley, 1990.
- [113] Green, A., *Intelligent Tracking Control of Fixed-Base and Free Flying Flexible Space Robots*, Ph.D. thesis, Department of Mechanical and Aerospace Engineering, Carleton University, Ottawa, Canada, 2007.
- [114] Palm, R., Driankov, D., and Hellendoorn, H., *Model-Based Fuzzy Control*, Springer-Verlag, 1997.
- [115] Misir, D., Malki, H. A., and Chen, G., "Graphical Stability Analysis for a Fuzzy PID Controlled Robot Arm Model," *IEEE International Conference on Fuzzy Systems*, Anchorage, AK, MAY 1998.
- [116] MATRIXx, "SystemBuild FuzzyLogic Block," April 2007, National Instruments Corporation.
- [117] Barkana, I. and Kaufman, H., "Global Stability and Performance of an Adaptive Control Algorithm," *International Journal of Control*, Vol. 42, No. 6, 1985, pp. 1491–1505.
- [118] Fradkov, A. L., "Synthesis of an Adaptive System for Linear Plant Stabilization," *Automation and Remote Control*, Vol. 35, No. 12, 1974, pp. 1960–1966.
- [119] Fradkov, A. L., "Quadratic Lyapunov Function in the Adaptive Stabilization Problem of a Linear Dynamic Plant," *Siberian Mathematics Journal*, Vol. 2, 1976, pp. 341–341.
- [120] Barkana, I., "Positive Realness in Multivariable Stationary Linear Systems," *Journal of The Franklin Institute*, Vol. 328, No. 4, 1991, pp. 403–417.

- [121] Barkana, I., "Comments on: Design of Strictly Positive Real Systems Using Constant Output Feedback," *IEEE Transactions on Automatic Control*, Vol. 49, No. 11, 2004, pp. 2091–2093.
- [122] Weiss, H., Wang, Q., and Speyer, J. L., "System Characterization of Positive Real Conditions," *IEEE Transactions on Automatic Control*, Vol. 39, No. 3, 1994, pp. 540–544.
- [123] Huang, C.-H., Ioannou, P. A., Maroulas, J., and Safonov, M. G., "Design of Strictly Positive Real Systems Using Constant Output Feedback," *IEEE Transactions on Automatic Control*, Vol. 44, No. 3, 1999, pp. 569–573.
- [124] Astrom, K. J. and Wittenmark, B., *Adaptive Control*, Addison-Wesley, 1995.
- [125] Landau, I. D., *Adaptive Control: the Model Reference Approach*, Marcel Dekker, 1979.
- [126] Dubowsky, S. and DesForges, D. T., "The Application of Model-Referenced Adaptive Control to Robotic Manipulators," *Journal of Dynamic Systems, Measurement and Control*, Vol. 101, No. 3, 1979, pp. 193–200.
- [127] Takegaki, M. and Arimoto, S., "An Adaptive Trajectory Control of Manipulators," *International Journal of Control*, Vol. 34, No. 3, 1981, pp. 219–230.
- [128] Craig, J. J., Hsu, P., and Sastry, S. S., "Adaptive Control of Mechanical Manipulators," *IEEE International Conference on Robotics and Automation*, San Francisco, CA, APR 1986.
- [129] Nicosia, S. and Tomei, P., "Model Reference Adaptive Control Algorithms for Industrial Robots," *Automatica*, Vol. 20, No. 5, 1984, pp. 635–644.
- [130] Horowitz, R. and Tomizuka, M., "An Adaptive Control Scheme for Mechanical Manipulators-Compensation of Nonlinearity and Decoupling Control," *Journal of Dynamic Systems, Measurement and Control*, Vol. 108, No. 2, 1986, pp. 127–135.
- [131] Slotine, J. J. E. and Li, W., "On the Adaptive Control of Robot Manipulators," *International Journal of Robotics Research*, Vol. 6, No. 3, 1987, pp. 49–59.
- [132] Sobel, K., Kaufman, H., and Mabius, L., "Implicit Adaptive Control for a Class of MIMO Systems," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 18, No. 5, 1982, pp. 576–589.
- [133] Barkana, I., Kaufman, H., and Balas, M., "Model Reference Adaptive Control of Large Structural Systems," *Journal of Guidance, Control, and Dynamics*, Vol. 6, No. 2, 1983, pp. 112–118.

- [134] Barkana, I. and Kaufman, H., "Some Applications of Direct Adaptive Control to Large Structural Systems," *Journal of Guidance, Control, and Dynamics*, Vol. 7, No. 6, 1984, pp. 717–724.
- [135] Broussard, J. R. and O'Brien, M. J., "Feedforward Control to Track the Output of a Forced Model," *IEEE Transactions on Automatic Control*, Vol. 25, No. 4, 1980, pp. 851–853.
- [136] Ioannou, P. A. and Kokotovic, P. V., "Singular Perturbation and Robust Redesign of Adaptive Control," *21st IEEE Conference on Decision and Control*, Orlando, FL, DEC 1982.
- [137] Ioannou, P. A. and Kokotovic, P. V., *Adaptive Systems with Reduced Models*, Springer-Verlag, 1983.
- [138] Ulrich, S. and de Lafontaine, J., "Autonomous Atmospheric Entry on Mars: Performance Improvement Using a Novel Adaptive Control Algorithm," *The Journal of the Astronautical Sciences*, Vol. 55, No. 4, 2007, pp. 431–449.
- [139] Barkana, I., "Output Feedback Stabilizability and Passivity in Nonstationary and Nonlinear Systems," *International Journal of Adaptive Control and Signal Processing*, Vol. 24, No. 7, 2010, pp. 568–591.
- [140] Goulet, J. F., de Silva, C. W., Modi, V. J., and Misra, A. K., "Hierarchical Control of a Space-Based Deployable Manipulator Using Fuzzy Logic," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 2, 2001, pp. 395–405.
- [141] Ahmad, M. A., Ismail, R. M. T. R., Ramli, M. S., Zawawi, M. A., Hambali, N., and Ghani, N. M. A., "Vibration Control of Flexible Joint Manipulator using Input Shaping with PD-type Fuzzy Logic Control," *IEEE International Symposium on Industrial Electronics*, Seoul, Korea, JUL 2009, pp. 1184–1189.
- [142] Park, C.-W. and Cho, Y.-W., "Adaptive Tracking Control of Flexible Joint Manipulator Based on Fuzzy Model Reference Approach," *IEE Proceedings on Control Theory and Applications*, Vol. 150, No. 2, 2003, pp. 198–204.
- [143] Weiming, T., Guanrong, C., and Rongde, L., "A Modified Fuzzy PI Controller for a Flexible-Joint Robot Arm with Uncertainties," *Fuzzy Sets and Systems*, Vol. 118, No. 1, 2001, pp. 109–119.
- [144] Khalil, H. K., *Nonlinear Systems*, PrenticeHall, 3rd ed., 2002.
- [145] Spong, M. W., "Adaptive Control of Flexible Joint Manipulators," *Systems and Control Letters*, Vol. 13, 1989, pp. 15–21.
- [146] Spong, M. W., "Adaptive Control of Flexible Joint Manipulators: Comments on Two Papers," *Automatica*, Vol. 31, No. 4, 1995, pp. 585–590.

- [147] Chang, Y. Z. and Daniel, R. W., "On the Adaptive Control of Flexible Joint Robots," *Automatica*, Vol. 28, No. 5, 1992, pp. 969–974.
- [148] Ott, C., Albu-Schaffer, A., and Hirzinger, G., "Comparison of Adaptive and Nonadaptive Tracking Control Laws for a Flexible Joint Manipulator," *IEEE International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, OCT 2002.
- [149] Huang, L., Ge, S. S., and Lee, T. H., "Position/Force Control of Uncertain Constrained Flexible Joint Robots," *Mechatronics*, Vol. 16, No. 2, 2006, pp. 111–120.
- [150] Subudhi, B. and Morris, A. S., "Singular Perturbation Approach to Trajectory Tracking of Flexible Robot with Joint Elasticity," *International Journal of Systems Science*, Vol. 34, No. 3, 2003, pp. 167–179.
- [151] Cao, Y. and de Silva, C. W., "Dynamic Modeling and Neural-Network Adaptive Control of a Deployable Manipulator System," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 1, 2006, pp. 192–194.
- [152] Kokotovic, P. V., Khalil, A., and O'Reilly, J., *Singular Perturbation Methods in Control: Analysis and Design*, Classics in Applied Mathematics Series, SIAM, 1999.
- [153] Wit, C. D., Noel, P., Aubin, A., and Brogliato, B., "Adaptive Friction Compensation in Robot Manipulators: Low Velocities," *International Journal of Robotics Research*, Vol. 10, No. 3, 1991, pp. 189–199.
- [154] Makkar, C., Dixon, W. E., Sawyer, W. G., and Hu, G., "A New Continuously Differentiable Friction Model for Control Systems Design," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Lausanne, Switzerland, JUL 2005, pp. 600–605.
- [155] Patre, P., MacKunis, W., Makkar, C., and Dixon, W. E., "Asymptotic Tracking for Systems With Structured and Unstructured Uncertainties," *IEEE Transactions on Control Systems Technology*, Vol. 16, No. 2, 2008, pp. 373–379.
- [156] Lightcap, C. A., *Measurement and Control Issues in a Novel Dynamic Radiographic Imaging System*, Ph.D. thesis, University of Florida, 2008.
- [157] Lightcap, C. A. and Banks, S. A., "An Extended Kalman Filter for Real-Time Estimation and Control of a Rigid-Link Flexible-Joint Manipulator," *IEEE Transactions on Control Systems Technology*, Vol. 18, No. 1, 2010, pp. 91–103.
- [158] Ott, C., *Cartesian Impedance Control of Redundant and Flexible-Joint Robots*, Springer Tracts in Advanced Robotics, Springer, 2008.

- [159] Readman, M. C., *Flexible Joint Robots*, Mechatronics, CRC Press, 1994.
- [160] Slotine, J. J. E. and Li, W., "Adaptive Manipulator Control: A Case Study," *IEEE Transactions on Automatic Control*, Vol. 33, No. 11, 1988, pp. 995–1003.
- [161] Sporer, N., "DLR Light Weight Robot III Data Sheet," Tech. rep., German Aerospace Center Institute of Robotics and Mechatronics, 2012.
- [162] Oh, J. H. and Lee, J. S., "Backstepping Control Design of Flexible Joint Manipulator Using Only Position Measurements," *IEEE Conference on Decision and Control*, Tempa, FL, Dec 1998.
- [163] Leon-Morales, J. D., Alvarez-Leal, J. G., Castro-Linares, R., and Alvarez-Gallegos, J. A., "Control of a Flexible Joint Robot Manipulator Via a Non-linear Control-Observer Scheme," *International Journal of Control*, Vol. 74, No. 3, 2001, pp. 290–302.
- [164] Lim, S. Y., Dawson, D. M., Hu, J., and de Queiroz, M. S., "An Adaptive Link Position Tracking Controller for Rigid-Link Flexible-Joint Robots Without Velocity Measurements," *IEEE Transaction on Systems, Man and Cybernetics*, Vol. 27, No. 3, 1997, pp. 412–427.
- [165] Loria, A., "Global Tracking Control of One Degree of Freedom EulerLagrange Systems Without Velocity Measurements," *European Journal of Control*, Vol. 2, No. 2, 1996, pp. 144–151.
- [166] Dixon, W. E., Zergeroglu, E., de Queiroz, M. S., and Dawson, D. M., "Global Output Feedback Tracking Control for Rigid-Link Flexible-Joint Robots," *IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May 1998.
- [167] Nicosia, S., Tomei, P., and Tornambe, A., "A Nonlinear Observer for Elastic Robots," *IEEE Transactions on Robotics and Automation*, Vol. 4, No. 1, 1988, pp. 45–52.
- [168] Tomei, P., "An Observer for Flexible Joint Robots," *IEEE Transactions on Robotics and Automation*, Vol. 35, No. 6, 1990, pp. 739–743.
- [169] Chatlatanagulchai, W., Nho, H. C., and Meckl, P., "Robust Observer Backstepping Neural Network Control of Flexible-Joint Manipulators," *American Control Conference*, Boston, MA, JUN 2004, pp. 1154–1159.
- [170] Abdollahi, F., Talebi, H. A., and Patel, R. V., "A Stable Neural Networkbased Observer with Application to Flexible-Joint Manipulators," *IEEE Transactions on Neural Network*, Vol. 17, No. 1, 2006, pp. 118–129.

- [171] Nicosia, S. and Tomei, P., "A Tracking Controller for Flexible-Joint Robots Using Only Link Position Feedback," *IEEE Transactions on Automatic Control*, Vol. 40, No. 5, 1995, pp. 885–890.
- [172] Dixon, W. E., Zergeroglu, E., Dawson, D. M., and Hannan, M. W., "Global Adaptive Partial State Feedback Tracking Control of Rigid-Link Flexible-Joint Robots," *Robotica*, Vol. 18, No. 3, 2000, pp. 325–336.
- [173] Kim, M. S. and Lee, J. S., "Adaptive Tracking Control of Flexible-Joint Manipulators without Overparameterization," *Journal of Robotic Systems*, Vol. 12, No. 7, 2004, pp. 369–379.
- [174] Melhem, K. and Wang, W., "Global Output Tracking Control of Flexible Joint Robots via Factorization of the Manipulator Mass Matrix," *IEEE Transactions on Robotics*, Vol. 25, No. 2, 2009, pp. 428–437.
- [175] Gourdeau, R. and Schwartz, H. M., "Adaptive Control of Robotic Manipulators: Experimental Results," *IEEE Conference on Robotics and Automation*, Sacramento, CA, APR 1991, pp. 8–15.
- [176] Lertpiriyasuwat, V. and Berg, M. C., "Adaptive real-time estimation of end-effector position and orientation using precise measurements of end-effector position," *IEEE/ASME Transactions on Mechatronics*, Vol. 11, No. 3, 2006, pp. 304–319.
- [177] Timcenko, A. and Kircanski, N., "Control of Robots with Elastic Joints: Deterministic Observer and Kalman Filter Approach," *IEEE Conference on Robotics and Automation*, Nice, France, MAY 1992, pp. 722–727.
- [178] Hollars, M. G. and Jr., R. H. C., "Experimental Implementation of a Nonlinear Estimator in the Control of Flexible Joint Manipulators," *IFAC Aerospace Conference*, Tsukuba, Japan, JUL 1990.
- [179] Kobayashi, T. and Litt, D. L. S. J. S., "Applications of a Constant Gain Extended Kalman Filter for In-Flight Estimation of Aircraft Engine Performance Parameters," Tech. Rep. TM-2005-213865, NASA, 2005.
- [180] Isidori, A., *Nonlinear Control Systems*, Springer-Verlag, 3rd ed., 1995.
- [181] Ulrich, S. and Sasiadek, J. Z., "Direct Fuzzy Adaptive Control of a Manipulator with Elastic Joints," *accepted for publication in the Journal of Guidance, Control, and Dynamics*.
- [182] Ulrich, S., Sasiadek, J. Z., and Barkana, I., "Modeling and Direct Adaptive Control of a Flexible-Joint Manipulator," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 1, 2012, pp. 25–39.

- [183] Ulrich, S. and Sasiadek, J. Z., "Trajectory Tracking Control of Flexible-Joint Space Manipulators," *Canadian Aeronautics and Space Journal*, Vol. 58, No. 1, 2012, pp. 47–59.
- [184] Ulrich, S. and Sasiadek, J. Z., "Real-Time Estimation and Adaptive Control of Flexible-Joint Space Manipulators," *Robot Motion and Control*, edited by K. Kozłowski, Vol. 442 of *Series on Lecture Notes in Control and Information Sciences*, Springer-Verlag, 2012, pp. 215–224.
- [185] Ulrich, S., Sasiadek, J. Z., and Barkana, I., "On a New Class of Direct Adaptive Output Feedback Controllers for Nonlinear Square Systems," *51st IEEE Conference on Decision and Control*, Maui, HI, DEC 2012.
- [186] Ulrich, S. and Sasiadek, J. Z., "Composite Simple Adaptive Control for Flexible-Joint Space Manipulators with Uncertainties and Modeling Errors," *ASME Dynamic Systems and Control Conference*, Fort Lauderdale, FL, OCT 2012.
- [187] Ulrich, S. and Sasiadek, J. Z., "Direct Fuzzy Adaptive Control of a Manipulator with Elastic Joints," *AIAA Guidance, Navigation, and Control Conference*, Minneapolis, MN, AUG 2012.
- [188] Ulrich, S., Sasiadek, J. Z., and Barkana, I., "Decentralized Simple Adaptive Control for Nonsquare Euler-Lagrange Systems," *American Control Conference*, Montreal, Canada, JUN 2012.
- [189] Ulrich, S. and Sasiadek, J. Z., "Methods of Trajectory Tracking for Flexible Joint Space Manipulators," *18th IFAC World Congress*, Milan, Italy, SEP 2011, pp. 10307–10312.
- [190] Ulrich, S. and Sasiadek, J. Z., "Direct Adaptive Fuzzy Control for a Two-Link Space Robot," *16th IEEE/IFAC International Conference on Methods and Models in Automation and Robotics*, Miedzyzdroje, Poland, AUG 2011, pp. 290–295.
- [191] Ulrich, S. and Sasiadek, J. Z., "Control Strategies for Flexible Joint Manipulators," *AIAA Guidance, Navigation, and Control Conference*, Portland, OR, AUG 2011.
- [192] Ulrich, S. and Sasiadek, J. Z., "Extended Kalman Filtering for Flexible Joint Space Robot Control," *American Control Conference*, San Francisco, CA, JUL 2011, pp. 1021–1026.
- [193] Ulrich, S. and Sasiadek, J. Z., "Real-Time Estimation and Adaptive Control of Flexible Joint Space Manipulators," *8th International Workshop on Robot Motion and Control*, Bukowy Dworek, Poland, JUN 2011.

- [194] Ulrich, S. and Sasiadek, J. Z., "Direct Model Reference Adaptive Control of a Flexible Joint Robot," *AIAA Guidance, Navigation and Control Conference*, Toronto, Canada, AUG 2010.
- [195] Ulrich, S. and Sasiadek, J. Z., "Modified Simple Adaptive Control for a Two-Link Space Robot," *American Control Conference*, Baltimore, MD, JUN 2010, pp. 3654–3659.
- [196] Ulrich, S. and Sasiadek, J. Z., "Control of Rigid and Flexible Joints Space Manipulators," *15th CASI Conference on Astronautics*, Toronto, Canada, MAY 2010.
- [197] Ulrich, S. and Sasiadek, J. Z., "Autonomous Control for Flexible Joint Space Robotic Manipulators," *60th International Astronautical Congress*, Daejeon, Korea, OCT 2009.
- [198] Fradkov, A. L., "Passification of Non-Square Linear Systems and Feedback Yakubovich-Kalman-Popov Lemma," *European Journal of Control*, Vol. 9, No. 6, 2003, pp. 573–582.
- [199] Mendel, J. M., *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*, Prentice-Hall, 2001.
- [200] Mendel, J. M. and John, R. I. B., "Type-2 Fuzzy Sets Made Simple," *IEEE Transactions of Fuzzy Systems*, Vol. 10, No. 2, 2002, pp. 117–127.
- [201] Hagrais, H. A., "A Hierarchical Type-2 Fuzzy Logic Control Architecture for Autonomous Mobile Robots," *IEEE Transactions of Fuzzy Systems*, Vol. 12, No. 4, 2003, pp. 524–539.
- [202] Chaoui, H. and Gueaieb, W., "Type-2 Fuzzy Logic Control of a Flexible-Joint Manipulator," *Journal of Intelligent and Robotic Systems*, Vol. 51, No. 2, 2008, pp. 159–186.
- [203] Biglarbegian, M., Melek, W. W., and Mendel, J. M., "Design of Novel Interval Type-2 Fuzzy Controllers for Modular and Reconfigurable Robots: Theory and Experiments," *IEEE Transactions on Industrial Electronics*, Vol. 58, No. 4, 2011, pp. 1371–1384.
- [204] Zhu, W.-H. and de Schutter, J., "Adaptive Control of Electrically Driven Space Robots Based on Virtual Decomposition," *Journal of Guidance, Control, and Dynamics*, Vol. 22, No. 2, 1999, pp. 329–339.
- [205] Côté, J. and de Lafontaine, J., "Magnetic-Only Orbit and Attitude Estimation Using the Square-Root Unscented Kalman Filter: Application to the PROBA-2 Spacecraft," *AIAA Guidance, Navigation and Control Conference and Exhibit*, Honolulu, HI, AUG 2008.

- [206] Julier, S. J. and Uhlmann, J. K., "A New Extension of the Kalman Filter to Nonlinear Systems," *11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Multi Sensor Fusion, Tracking and Resource Management II*, Orlando, FL, APR 1997.
- [207] Kokotovic, P. V. and Sussmann, H., "A positive real condition for global stabilization of nonlinear systems," *Systems and Control Letters*, Vol. 13, 1989, pp. 125-133.
- [208] Spong, M. W., Ortega, R., and Kelly, R., "Comments on: Adaptive Manipulator Control: a Case Study," *IEEE Transactions on Automatic Control*, Vol. 35, No. 6, 1990, pp. 761-762.
- [209] Lewis, F. L., *Optimal Estimation with an Introduction to Stochastic Control Theory*, Wiley, 1986.

Biographical Sketch

STEVE ULRICH received his B.Eng. and M.A.Sc. degrees in electrical engineering from Université de Sherbrooke (Sherbrooke, Canada), in 2004 and 2006, respectively. From 2006 to 2008, he was Research Engineer in spacecraft guidance, navigation and control (GN&C) at NGC Aerospace Ltd. (Sherbrooke, Canada) where he namely served as key technical leader in the development of an innovative nonlinear attitude disturbance observer for Proba-2; a spacecraft being part of ESAs In-orbit Technology Demonstration program, which was successfully launched in 2009.

He is the recipient of several awards, including the *J.Y and E.W. Wong Research Award in Mechanical/Aerospace Engineering* from Carleton University, the *Alexander Graham Bell Canada Graduate Scholarship* from the Natural Sciences and Engineering Research Council of Canada (NSERC), and the *Leonardo da Vinci Medal*; the highest distinction awarded by the Faculty of Engineering at Université de Sherbrooke. In 2007, he was awarded a fellowship to attend the 9-week summer session program of the International Space University (ISU), which was held at the Beijing University of Aeronautics and Astronautics, in China.

His research interests are in the area of theory and practice of advanced GN&C systems for aerospace applications. Non-controls technical interests include nonlinear filtering; real-time state estimation; and nonlinear disturbance observation. Applications include: control and vibration suppression for flexible space manipulators; spacecraft attitude GN&C; aircraft control.

Steve Ulrich is a registered Professional Engineer in the province of Ontario and registered Engineer in the province of Quebec, a member of the AIAA, the IEEE, and the ASME.