1

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS
STANDARD REFERENCE MATERIAL 1010a
(ANSI and ISO TEST CHART No. 2)

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Adaptive Control of Robotic Manipulators

by

RICHARD GOURDEAU, B.Sc.A., M.Sc.

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfilment of
the requirements for the degree of
Doctor of Philosophy

Department of Mechanical and Aerospace Engineering

Carleton University

Ottawa, Ontario

October 30, 1990

© Copyright

1990, Richard Gourdeau

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

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

ISBN 0-315-68904-8

Canada

The undersigned hereby recommend to
the Faculty of Graduate Studies and Research
acceptance of the thesis,

Adaptive Control of Robotic Manipulators

submitted by

RICHARD GOURDEAU, B.Sc.A., M.Sc.

in partial fulfilment of the requirements
for the degree of Doctor of Philosophy

Chairman, Department of
Mechanical and Aerospace Engineering

Thesis Supervisor

External Examiner

Carleton University

April, 22, 1991

Abstract

This work presents a new adaptive motion control scheme for robotic manipulators. This is an adaptive computed torque method (CTM) that requires only position measurements. These measurements and the input torques are used in an extended Kalman filter (EKF) to estimate the inertial parameters of the full non-linear robot model as well as the joint positions and velocities. These estimates are used by the CTM to generate the input torques. This combination of the EKF and the CTM is shown to result in a stable adaptive control scheme. The theory behind Kalman filtering provides clear guide-lines on the selection of the design parameters for the controller when noise is present. Simulations illustrate the performance of this scheme and demonstrate its noise rejection properties. Experimental results with a two-degree-of-freedom direct-drive robot show the efficiency and the applicability of the scheme. Also, a review of the existing adaptive controller schemes is made and these are compared to the new scheme using simulations in both deterministic and stochastic contexts. Optimal path planning for point-to-point motion is also treated.

Acknowledgements

I would like to thank my supervisor, professor Howard M. Schwartz, for the good and long discussions we had, his guidance and the enthusiasm he showed toward this project.

The financial support received to complete this degree, from the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Fonds pour la Formation des Chercheurs et l'Aide à la Recherche du Québec (FCAR), is greatly appreciated.

I would like to thank Gabriel Warshaw for the help he gave me while he introduced me to the operation of the experimental robot. Finally, I want to thank the staff from the Mechanical & Aerospace Engineering Department and the System & Computer Engineering Department.

Table of Contents

Acceptance Sheet	ii
Abstract	iii
Acknowledgements	iv
Table of Contents	v
List of Tables	x
List of Figures	xi
List of Symbols	xviii
1 Introduction	1
1.1 Rationale for the Research	1
1.1.1 SISO Decoupled Systems	2
1.1.2 MIMO in the Deviation from a Nominal Path	2
1.1.3 Robot Nonlinear Dynamics Cancellations	3
1.2 Outline of the Thesis	4
1.3 The Original Contribution	5
2 Fundamentals	6
2.1 Robot Dynamic Model	6

2.1.1	Continuous-Time Model	6
2.1.2	Perturbation Model	7
2.1.3	Model with Unknown Parameters	8
2.2	Parameter Estimation Problem	10
2.2.1	Problem Statement	11
2.2.2	Identifiability of the Parameters	11
2.2.3	Choice of the Parameters to be Estimated	12
2.2.4	Possible Estimators	12
2.3	Kalman Filters	14
2.3.1	Continuous-Time Filters	15
2.3.2	Discrete-Time Filters	19
3	Adaptive Control of Robots: Review	24
3.1	Introduction	24
3.2	Schemes Based on Non-Linear Robot Models	25
3.2.1	Measurable τ , q , $\dot{q}$ and $\ddot{q}$	25
3.2.2	Measurable τ , q and $\dot{q}$	31
3.3	Robustness to Disturbances	36
3.4	Implementation	37
3.5	Simulations Examples	39
3.6	Conclusion	44
4	The New Adaptive Controller	59
4.1	Introduction	59
4.2	Adaptive Control Algorithm	60
4.3	Adding Robustness to the Scheme	72
4.4	Implementation	77
4.4.1	Selection of the Filter Initial Conditions	77
4.4.2	Selection of the Covariance Matrices	78

4.4.3	Discrete-Time Controller Implementation	79
4.5	Example	82
4.5.1	Two DOF Robot Example	82
4.5.2	Robustness Improvement Example	84
4.6	Conclusion	87
5	Experimental Results	97
5.1	Experimental Robot	97
5.1.1	Robot Dynamics	97
5.1.2	Hardware	99
5.1.3	Computational Aspect	100
5.1.4	Tachometer Calibration	100
5.2	One DOF Results	102
5.2.1	Desired Path	103
5.2.2	Adaptive Controller Parameters	104
5.2.3	Results	104
5.3	Two DOF Results	105
5.3.1	Desired Path	105
5.3.2	Adaptive Controller Parameters	106
5.3.3	Results	106
5.3.4	Computational Remarks	107
5.4	Conclusion	108
6	Optimal Path Planning	120
6.1	Introduction	120
6.2	Minimum-Time Cost Function	122
6.2.1	Solution to the Time-Optimal Problem	124
6.3	Weighted Time-Energy Cost Function	125
6.3.1	Solution to the TPBVP	127

6.4	Energy Cost Function	128
6.4.1	Solution to the TPBVP	129
6.5	Examples	130
6.6	Computational Remarks	133
6.7	Conclusion	134
7	Conclusions	143
A	Lagrangian Dynamics	145
A.1	Motion Equations	145
A.2	Partial Derivatives of $D(q)$, $h(q, \dot{q})$ and $g(q)$	148
B	Recursive Least Squares	150
B.1	Continuous-Time Recursive Least Squares	150
B.1.1	Unnormalized	150
B.1.2	With Forgetting Factor	151
B.2	Discrete-Time Recursive Least Squares	152
C	Observability and Controllability	153
C.1	Continuous Time Systems	153
C.1.1	Observability	153
C.1.2	Controllability	155
C.2	Discrete Time Systems	156
C.2.1	Observability	156
C.2.2	Controllability	157
D	Stability Theory	158
E	Two DOF Robot Dynamics	163
F	Experimental Robot Dynamics	166

List of Tables

3.1	Adaptive Control Algorithms	26
5.1	Adaptive Controller Computing Times	109
5.2	Non-Adaptive Controller Computing Times	109
6.1	Results for the Automelec ACR	135
6.2	Results for the IBM 7535 B 04	139

List of Figures

3.1	2 DOF Robot Schematic	45
3.2	Desired Path q_{1d} = solid line, $\dot{q}_{1d}/5$ = dashed line, $\ddot{q}_{1d}/10$ = dotted line	46
3.3	Tracking Error: Computed Torque: $q_1 - q_{1c}$ = solid line, $q_2 - q_{2d}$ = dashed line, Sliding Controller: $q_1 - q_{1d}$ = dotted line, $q_2 - q_{2d}$ = dash-dot line	46
3.4	Craig: Continuous-Time Controller: Parameter Errors: $\hat{p}_1 - p_1$ = solid line, $\hat{p}_2 - p_2$ = dashed line, $\hat{p}_3 - p_3$ = dotted line, $\hat{p}_4 - p_4$ = dash-dot line	47
3.5	Craig: Continuous-Time Controller: Tracking Errors: $q - q_{1d}$ = solid line, $q_2 - q_{2d}$ = dashed line	47
3.6	Indirect: Continuous-Time Controller: Parameter Errors: $\hat{p}_1 - p_1$ = solid line, $\hat{p}_2 - p_2$ = dashed line, $\hat{p}_3 - p_3$ = dotted line, $\hat{p}_4 - p_4$ = dash-dot line	48
3.7	Indirect: Continuous-Time Controller: Tracking Errors: $q_1 - q_{1d}$ = solid line, $q_2 - q_{2d}$ = dashed line	48
3.8	Slotine: Continuous-Time Controller: Parameter Errors: $\hat{p}_1 - p_1$ = solid line, $\hat{p}_2 - p_2$ = dashed line, $\hat{p}_3 - p_3$ = dotted line, $\hat{p}_4 - p_4$ = dash-dot line	49
3.9	Slotine: Continuous-Time Controller: Tracking Errors: $q_1 - q_{1d}$ = solid line, $q_2 - q_{2d}$ = dashed line	49

3.10 Indirect with Filter: Continuous-Time Controller: Parameter Errors:	
$\hat{p}_1 - p_1 =$ solid line, $\hat{p}_2 - p_2 =$ dashed line, $\hat{p}_3 - p_3 =$ dotted line, $\hat{p}_4 - p_4 =$ dash-dot line	50
3.11 Indirect with Filter: Continuous-Time Controller: Tracking Errors:	
$q_1 - q_{1d} =$ solid line, $q_2 - q_{2d} =$ dashed line	50
3.12 Craig: Discrete-Time Controller: Parameter Errors: $\hat{p}_1 - p_1 =$ solid line, $\hat{p}_2 - p_2 =$ dashed line, $\hat{p}_3 - p_3 =$ dotted line, $\hat{p}_4 - p_4 =$ dash-dot line	51
3.13 Craig: Discrete-Time Controller: Tracking Errors: $q_1 - q_{1d} =$ solid line, $q_2 - q_{2d} =$ dashed line	51
3.14 Indirect: Discrete-Time Controller: Parameter Errors: $\hat{p}_1 - p_1 =$ solid line, $\hat{p}_2 - p_2 =$ dashed line, $\hat{p}_3 - p_3 =$ dotted line, $\hat{p}_4 - p_4 =$ dash-dot line	52
3.15 Indirect: Discrete-Time Controller: Tracking Errors: $q_1 - q_{1d} =$ solid line, $q_2 - q_{2d} =$ dashed line	52
3.16 Slotine: Discrete-Time Controller: Parameter Errors: $\hat{p}_1 - p_1 =$ solid line, $\hat{p}_2 - p_2 =$ dashed line, $\hat{p}_3 - p_3 =$ dotted line, $\hat{p}_4 - p_4 =$ dash-dot line	53
3.17 Slotine: Discrete-Time Controller: Tracking Errors: $q_1 - q_{1d} =$ solid line, $q_2 - q_{2d} =$ dashed line	53
3.18 Indirect with Filter: Discrete-Time Controller: Parameter Errors: $\hat{p}_1 - p_1 =$ solid line, $\hat{p}_2 - p_2 =$ dashed line, $\hat{p}_3 - p_3 =$ dotted line, $\hat{p}_4 - p_4 =$ dash-dot line	54
3.19 Indirect with Filter: Discrete-Time Controller: Tracking Errors: $q_1 - q_{1d} =$ solid line, $q_2 - q_{2d} =$ dashed line	54
3.20 Craig with Noise: Discrete-Time Controller: Parameter Errors: $\hat{p}_1 - p_1 =$ solid line, $\hat{p}_2 - p_2 =$ dashed line, $\hat{p}_3 - p_3 =$ dotted line, $\hat{p}_4 - p_4 =$ dash-dot line	55

3.21	Craig with Noise: Discrete-Time Controller: Tracking Errors: $q_1 - q_{1d}$ = solid line, $q_2 - q_{2d}$ = dashed line	55
3.22	Indirect with Noise: Discrete-Time Controller: Parameter Errors: $\hat{p}_1 - p_1$ = solid line, $\hat{p}_2 - p_2$ = dashed line, $\hat{p}_3 - p_3$ = dotted line, $\hat{p}_4 - p_4$ = dash-dot line	56
3.23	Indirect with Noise: Discrete-Time Controller: Tracking Errors: $q_1 - q_{1d}$ = solid line, $q_2 - q_{2d}$ = dashed line	56
3.24	Slotine with Noise: Discrete-Time Controller: Parameter Errors: $\hat{p}_1 - p_1$ = solid line, $\hat{p}_2 - p_2$ = dashed line, $\hat{p}_3 - p_3$ = dotted line, $\hat{p}_4 - p_4$ = dash-dot line	57
3.25	Slotine with Noise: Discrete-Time Controller: Tracking Errors: $q_1 - q_{1d}$ = solid line, $q_2 - q_{2d}$ = dashed line	57
3.26	Indirect with Filter and Noise: Discrete-Time Controller: Parameter Errors: $\hat{p}_1 - p_1$ = solid line, $\hat{p}_2 - p_2$ = dashed line, $\hat{p}_3 - p_3$ = dotted line, $\hat{p}_4 - p_4$ = dash-dot line	58
3.27	Indirect with Filter and Noise: Discrete-Time Controller: Tracking Errors: $q_1 - q_{1d}$ = solid line, $q_2 - q_{2d}$ = dashed line	58
4.1	Adaptive controller structure	88
4.2	The $\sigma(\hat{\theta})$ Function	89
4.3	Unmodified and Modified parameter models simulation: modified ($Q_2 = 1, \alpha = 10, \theta_{min} = -0.4, \theta_{max} = 0.4$) = solid line, unmodified = dashed and dotted lines	89
4.4	Discrete-Time Controller without Noise: Parameter Errors: $\hat{p}_1 - p_1$ = solid line, $\hat{p}_2 - p_2$ = dashed line, $\hat{p}_3 - p_3$ = dotted line, $\hat{p}_4 - p_4$ = dash-dot line	90
4.5	Discrete-Time Controller without Noise: Tracking Errors: $q_1 - q_{1d}$ = solid line, $q_2 - q_{2d}$ = dashed line	90

4.6	Discrete-Time Controller without Noise: Filtering Errors: $\hat{q}_1 - q_1 =$ solid line, $\hat{q}_2 - q_2 =$ dashed line	91
4.7	Discrete-Time Controller without Noise: Filtering Errors: $\hat{\dot{q}}_1 - \dot{q}_1 =$ solid line, $\hat{\dot{q}}_2 - \dot{q}_2 =$ dashed line	91
4.8	Discrete-Time Controller with Noise: Parameter Errors: $\hat{p}_1 - p_1 =$ solid line, $\hat{p}_2 - p_2 =$ dashed line, $\hat{p}_3 - p_3 =$ dotted line, $\hat{p}_4 - p_4 =$ dash-dot line	92
4.9	Discrete-Time Controller with Noise: Tracking Errors: $q_1 - q_{1d} =$ solid line, $q_2 - q_{2d} =$ dashed line	92
4.10	Discrete-Time Controller with Noise: Filtering Errors: $\hat{q}_1 - q_1 =$ solid line, $\hat{q}_2 - q_2 =$ dashed line	93
4.11	Discrete-Time Controller with Noise: Filtering Errors: $\hat{\dot{q}}_1 - \dot{q}_1 =$ solid line, $\hat{\dot{q}}_2 - \dot{q}_2 =$ dashed line	93
4.12	Discrete-Time Controller with Noise: Joint 2 Velocities: real = solid line, filtered = dashed line, direct differentiation = dotted line	94
4.13	Discrete-Time Controller Lyapunov Function: without noise = solid line, with noise = dashed line	94
4.14	Parameter: Modified Filter: $\alpha = 5 =$ solid line, $\alpha = 1 =$ dashed line, $\alpha = 0.5 =$ dotted line $\alpha = 0.1 =$ dash-dot line, $\alpha = 0 =$ solid line . .	95
4.15	Parameter Variance: Modified Filter: $\alpha = 5 =$ solid line, $\alpha = 1 =$ dashed line, $\alpha = 0.5 =$ dotted line $\alpha = 0.1 =$ dash-dot line, $\alpha = 0 =$ solid line	95
4.16	Tracking Error: Modified Filter: $\alpha = 5 =$ solid line, $\alpha = 1 =$ dashed line, $\alpha = 0.5 =$ dotted line $\alpha = 0.1 =$ dash-dot line, $\alpha = 0 =$ solid line	96
5.1	Direct-Drive Robot Schematic	110
5.2	Desired Path	110
5.3	Measured and Desired Trajectories: Motor 1: q_m first case = solid line, q_m second case = dashed line, $q_d =$ dotted line	111

5.4	Tracking Errors: Motor 1: $q_m - q_d$ first case = solid line, $q_m - q_d$ second case = dashed line	111
5.5	Measured and Filtered Velocities: Motor 1: First Case: $\hat{q}$ = solid line, $\dot{q}_m$ = dashed line	112
5.6	Parameter Estimates: Motor 1: $\hat{p}_1$ first case = solid line, $\hat{p}_2$ first case = dashed line, $\hat{p}_1$ second case = dotted line, $\hat{p}_2$ second case = dash-dot line	112
5.7	Tracking Errors: Motor 2: $q_m - q_d$ first case = solid line, $q_m - q_d$ second case = dashed line	113
5.8	Parameter Estimates: Motor 2: $\hat{p}_1$ first case = solid line, $\hat{p}_2$ first case = dashed line, $\hat{p}_1$ second case = dotted line, $\hat{p}_2$ second case = dash-dot line	113
5.9	Control Inputs: Motor 1: u first case = solid line, u second case = dashed line	114
5.10	Measured and Desired Trajectories: q_{1m} = solid line, q_{1d} = dashed line, q_{2m} = dotted line, q_{2d} = dash-dot line	114
5.11	Tracking Errors: $q_{1m} - q_{1d}$ = solid line, $q_{2m} - q_{2d}$ = dashed line . . .	115
5.12	Measured and Filtered Positions: q_{1m} = solid line, $\hat{q}_1$ = dashed line, q_{2m} = dotted line, $\hat{q}_{2d}$ = dash-dot line	115
5.13	Filtered velocities: $\dot{q}_{1d}$, $\dot{q}_{2d}$, = solid line, $\hat{\dot{q}}_1$ = dashed line, $\hat{\dot{q}}_2$ = dash-dot line	116
5.14	Estimated Parameters	116
5.15	Tracking Errors: $q_{1m} - q_{1d}$ = solid line, $q_{2m} - q_{2d}$ = dashed line . . .	117
5.16	Estimated Parameters	117
5.17	Non-Adaptive controller: Tracking Errors: $q_{1m} - q_{1d}$ = solid line, $q_{2m} - q_{2d}$ = dashed line	118
5.18	Errors RMS: 60Hz RMS($q_d - q_m$) = +, 60Hz RMS($q_d - \hat{q}$) = ×, 30Hz RMS($q_d - q_m$) = o, 30Hz RMS($q_d - \hat{q}$) = *	118

5.19	Filtered and Measured Velocities: $\dot{q}_1$ = solid line, $\dot{q}_{1m}$ = dashed line .	119
6.1	Robot Automelec ACR schematic	135
6.2	ACR robot: Time-optimal trajectory, joint 1: position = solid line, velocity = dashed line, normalized input = dash-dot line	136
6.3	ACR robot: Time-optimal trajectory, joint 2: position = solid line, velocity = dashed line, normalized input = dash-dot line	136
6.4	ACR robot: Optimal trajectory $\mu = 0.002$, joint 1: position = solid line, velocity = dashed line, normalized input = dash-dot line	137
6.5	ACR robot: Optimal trajectory $\mu = 0.002$, joint 2: position = solid line, velocity = dashed line, normalized input = dash-dot line	137
6.6	ACR robot: Optimal trajectory $\mu = 0.005$, joint 1: position = solid line, velocity = dashed line, normalized input = dash-dot line	138
6.7	ACR robot: Optimal trajectory $\mu = 0.005$, joint 2: position = solid line, velocity = dashed line, normalized input = dash-dot line	138
6.8	Robot IBM 7535 B 04 schematic	139
6.9	IBM robot: Time-optimal trajectory, joint 1: position = solid line, velocity = dashed line, normalized input = dash-dot line	140
6.10	IBM robot: Time-optimal trajectory, joint 2: position = solid line, velocity = dashed line, normalized input = dash-dot line	140
6.11	IBM robot: Optimal trajectory $\mu = 0.001$, joint 1: position = solid line, velocity = dashed line, normalized input = dash-dot line	141
6.12	IBM robot: Optimal trajectory $\mu = 0.001$, joint 2: position = solid line, velocity = dashed line, normalized input = dash-dot line	141
6.13	IBM robot: Optimal trajectory $\mu = 0.0025$, joint 1: position = solid line, velocity = dashed line, normalized input = dash-dot line	142
6.14	IBM robot: Optimal trajectory $\mu = 0.0025$, joint 2: position = solid line, velocity = dashed line, normalized input = dash-dot line	142

7.1 Combined adaptive and optimal control	144
---	-----

List of Symbols

Symbols

q	manipulator joint position vector
$\dot{q}$	manipulator joint velocity vector
$\ddot{q}$	manipulator joint acceleration vector
θ	system parameter vector
$D(q)$	manipulator inertia matrix
$g(q)$	manipulator gravitational force vector
$h(q)$	manipulator Coriolis and centrifugal force vector
$\nu(q)$	manipulator frictional force vector
τ	forces (or moments) applied at the manipulator joints
$\frac{\partial D}{\partial q} \xi$	$\left[\frac{\partial D}{\partial q_1} \xi \quad \dots \quad \frac{\partial D}{\partial q_n} \xi \right]$ with $n = \dim(q)$
$\frac{\partial D}{\partial \theta} \xi$	$\left[\frac{\partial D}{\partial \theta_1} \xi \quad \dots \quad \frac{\partial D}{\partial \theta_m} \xi \right]$ with $m = \dim(\theta)$
$q_d(t)$	desired joint position trajectory
e	tracking error vector

K_v	controller velocity feedback gain (positive definite matrix)
K_p	controller position feedback gain (positive definite matrix)
ϕ	parameter error vector
n	manipulator number of degree-of-freedom
t	time
$Y(q, \dot{q}, \ddot{q})$	manipulator regression matrix
$x(t)$	state space vector
$z(t)$	observation vector
$P(t), P(k)$	covariance matrices

Operations and Properties

$()^T$	transpose of a matrix (or vector)
$()^{-1}$	inverse of a matrix
$\dot{()}$	time derivative of a matrix (or vector)
$\hat{()}$	estimated value of a matrix (or vector)
$\tilde{()}$	error in the estimated value of a matrix (or vector)
$()^*$	value of a matrix (or vector) evaluated along the optimal path
$A > 0$	positive definite matrix
$A \geq 0$	positive semi-definite matrix

$\|x\|$ norm of a vector

$$\|x\|^2 = x^T x$$

$\|A\|$ norm of a matrix

$$\|A\| = \sup_{\|x\|=1} \|Ax\|$$

$\mathcal{E}\{\cdot\}$ expected value of a matrix (or vector)

L_p For any fixed $p \in [1, \infty)$, f is said to belong to L_p if, and only if, f is locally integrable and

$$\|f\|_p \equiv \left(\int_0^\infty \|f\|^p dt \right)^{1/p} < \infty$$

and when $p = \infty$, $f \in L_\infty$ if, and only if,

$$\|f\|_\infty \equiv \sup_{t \geq 0} \|f\| < \infty$$

$o(\|x\|)$ denotes a function of x , $g(x)$, such that

$$\frac{\|g(x)\|}{\|x\|} \rightarrow 0 \quad \text{as} \quad \|x\| \rightarrow 0$$

$O(\|x\|)$ denotes a function of x , $h(x)$, such that

$$\frac{\|h(x)\|}{\|x\|} \leq c \quad \text{as} \quad \|x\| \rightarrow 0$$

Chapter 1

Introduction

1.1 Rationale for the Research

The rationale for using adaptive controllers for robotic manipulators comes from the fact that, even after careful modeling, precise evaluation of the inertial parameters of the robot is very difficult and that every time a load is grasped the dynamic model is no longer an accurate representation of the robot behaviour. Also, as we are trying to operate these robots at higher speeds, the necessity for an accurate control model becomes imperative. Furthermore, the decreasing cost of computing power allows for the realization of complex nonlinear controllers that would have been impractical a few years ago. It is with this in mind that most adaptive motion control algorithms for robots were developed.

Adaptive control laws have been used to develop algorithms for robot motion control. Some of these control laws use linear or linear perturbation models for the manipulator and implement adaptive control techniques for linear systems in order to track the time variations of these approximate linear (or linearized) models (see Hsia [36] for a review of these methods). Their stability is based on the assumption that the linear system parameters are varying slowly which is not true in high speed motions, hence these methods might not be able to capture the fast changes in the manipulator's non-linear dynamics.

1.1.1 SISO Decoupled Systems

The results based on this approach (see [24, 52, 58, 79, 89] for examples) consider the robot as a decoupled system of second order time-varying single-input-single-output (SISO) systems in which the coupling are considered as disturbances. Then, a model reference adaptive controller (MRAC) or a self tuning regulator (STR) based on this model is used. For low speed motion these methods give reasonable results and are computationally simple, but the simplification of the model cannot cope with the nonlinear nature of robotic manipulator in high speed motion (the dynamics are changing at the same speed as the states and this violates the hypothesis on which these schemes are based). In these methods, the nonlinear model of the robot is not computed which is the advantage claimed by the authors. We believe that with the availability of fast low cost computers and efficient schemes to compute the robot dynamics these methods are outperformed by the methods based on nonlinear dynamics cancellation.

1.1.2 MIMO in the Deviation from a Nominal Path

Other results consider the robot model for the deviations from a nominal path. This model is a time-variant multi-input-multi-output (MIMO) system which is estimated in the adaptive controller (see [54, 55, 85] for examples). This results in a controller with a fixed feedforward term and adaptive feedback gains. These methods will only achieve bounded deviations from the nominal path since the feedforward term is not adjusted and the feedback term uses the deviations in a PD type controller. In these methods the feedforward term is computed off-line so as to avoid computing the dynamics in real time, but as mentioned earlier this is not a problem with modern computers. Furthermore, these methods are not proved to be superior to possible non-adaptive feedback laws.

1.1.3 Robot Nonlinear Dynamics Cancellations

Nonlinear dynamic cancellation can be applied to robotic manipulators to obtain error dynamics that are linear. However perfect path tracking will be achieved only if the robot inertial parameters and friction coefficients are perfectly known. The fact that the dynamic model of the robots can be put in a form which is linear in the parameters [1, 7, 27, 49, 50] has permitted the design of adaptive algorithms for these nonlinear controllers [18, 20, 19, 37, 40, 48, 56, 57, 61, 62, 63, 66, 68, 70, 69, 80, 81, 86]. Some of these methods will be the investigated in this work.

The fact that the manipulator dynamics are linear in the inertial parameters was pointed out by Atkeson [8] and Khosla [50]. In fact, Khosla [50] suggested an indirect adaptive controller structure that used this linearity and the availability of joint position, velocity and acceleration measurements. Craig [18, 20, 19] subsequently proposed a direct adaptive controller based on the computed torque method (CTM) also using joint position, velocity and acceleration measurements. The previous two results are globally stable, however the use of acceleration in the parameter update law leads to implementation problems, particularly if acceleration measurements are unavailable or very noisy. Middleton [63] introduced filtering of the regression matrix and modified the CTM to obtained a stable adaptive controller that uses only joint position and velocity measurements. Middleton [61, 62] also introduced a hybrid controller for discrete-time implementation using position, velocity and acceleration measurements. Slotine [80, 81] presented results similar to Craig's but in which only joint position and velocity measurements are used, however, strictly speaking Slotine's results are based on a quasi-Lyapunov function (this function is not strictly positive definite) and the comments found in Spong et al [82] resolved this problem. More recently, Johansson [40] proposed a more general direct method using position and velocity measurement from which Slotine's results [80, 81] can be view as a special case. Other similar results are available and one can refer to Ortega [66] for a review of adaptive schemes based on non-linear models.

The fact that velocity and acceleration measurements are corrupted by noise can lead to some problems in the application of these methods (see Schwartz [75]). It has been shown that disturbances can lead to instability due to parameter drift (see Reed and Ioannou [68]). Gaussian velocity (and acceleration when needed) measurement noise can also result in instability due to parameter drift since squared measured quantities appear in the adaptation laws and the regressors become stochastic. Robust adaptive controllers and composite adaptive controllers have been proposed to overcome these problems. However, these robust methods result in larger trajectory tracking errors which can nullify the usefulness of the adaptive control algorithms.

The fact that position measurements are very accurate was the incentive to try to develop an adaptive controller that uses only position measurements. In this work, an extended Kalman filter (EKF) is designed to provide parameter estimates for the full non-linear robot model. Furthermore, the EKF would also provide position and velocity estimates. The EKF estimates are used in the CTM to generate the input torques. This combination is shown to be stable. The theory behind Kalman filtering provides clear guide-lines on the selection of the design parameters of the controller. This is an advantage with respect to Craig's and Slotine's methods in which the adaptation gains have to be found by trial and error. Simulation and experimental results illustrate the performance of this scheme and demonstrate its good noise rejection properties and its robustness to random disturbances.

1.2 Outline of the Thesis

Chapter 2 presents fundamental results needed for the development of the control algorithms. Section 2.1 presents the robot non-linear and linear perturbation models. In section 2.2, a brief outline of the robot parameter estimation problem is presented. In section 2.3, results for Kalman filters are stated. In chapter 3, a review of non-linear adaptive controllers for motion control is made and simulation

examples are provided. The new adaptive controller is described in chapter 4 as well as its implementation and an example with simulations for a two degree-of-freedom planar robot. Chapter 5 presents experimental results for the new controller with a two-degree-of-freedom direct-drive robot. Finally, in chapter 6, we present results in the area of optimal path planning. Also, numerous appendices provide basic results for the reader not familiar with some topics.

1.3 The Original Contribution

This work presents new results in adaptive control for robotic manipulators. These are presented in chapter 4 in which Theorem 4.1 is main result. Simulations (chapter 4) as well as experimental results (chapter 5) confirm the validity of the approach. Also, the weighted time-energy cost function introduced in chapter 6, presents a new application of optimal control theory to robot path planning. All these results, were or will be presented in various conferences and journal articles [29, 31, 30, 32].

Chapter 2

Fundamentals

In this chapter, we present fundamental results that will be used in the remaining chapters.

2.1 Robot Dynamic Model

In this section, the dynamic model for the robotic manipulator is developed. Different formulations are used depending on the purpose for which the model is used. These models are based on the assumption that the links are rigid bodies and the joints are driven by direct drive d-c motors.

2.1.1 Continuous-Time Model

The manipulator's model can be formulated using Lagrangian dynamics (see appendix A) as

$$\tau = D(q)\ddot{q} + h(q, \dot{q}) + g(q) + \nu(\dot{q}) \quad (2.1)$$

where q , $\dot{q}$ and $\ddot{q}$ are respectively the manipulator joint position, velocity and acceleration vectors. $D(q)$ is the inertia matrix, $h(q, \dot{q})$ the Coriolis and centrifugal force vector, $g(q)$ the gravitational force vector, $\nu(\dot{q})$ the vector function of frictional

forces, and τ is the torque vector applied by the motors to the robot joints. Also , the manipulator acceleration is given by

$$\ddot{q} = D^{-1}(q)\{\tau - \nu(\dot{q}) - h(q, \dot{q}) - g(q)\} \quad (2.2)$$

This equation can be written in state space form by defining a state vector such as

$$x = \begin{bmatrix} q \\ \dot{q} \end{bmatrix} \quad (2.3)$$

and the input $u(t) = \tau$ so that

$$\dot{x} = f(x, u) \quad (2.4)$$

where

$$f(x, u) = \begin{bmatrix} \dot{q} \\ D^{-1}(q)\{u(t) - \nu(\dot{q}) - h(q, \dot{q}) - g(q)\} \end{bmatrix} \quad (2.5)$$

2.1.2 Perturbation Model

To apply either optimal control theory or extended Kalman filters to the robotic manipulator, perturbation models will be needed. The perturbation models are based on first order Taylor series expansion of the system model with respect to a nominal point or trajectory (x_0, u_0) . Let us define

$$\dot{x}_0 = f(x_0, u_0) \quad (2.6)$$

$$x = x_0 + \delta x \quad (2.7)$$

$$u = u_0 + \delta u \quad (2.8)$$

$$\dot{x}_0 + \delta \dot{x} = f(x_0 + \delta x, u_0 + \delta u) \quad (2.9)$$

$$\dot{x}_0 + \delta \dot{x} \approx f(x_0, u_0) + \frac{\partial f(x_0, u_0)}{\partial x} \delta x + \frac{\partial f(x_0, u_0)}{\partial u} \delta u \quad (2.10)$$

from equations (2.6) and (2.10), we have

$$\delta \dot{x} \approx \frac{\partial f(x_0, u_0)}{\partial x} \delta x + \frac{\partial f(x_0, u_0)}{\partial u} \delta u \quad (2.11)$$

with

$$\frac{\partial f}{\partial x} = \begin{bmatrix} 0 & I \\ F_{21}(t) & F_{22}(t) \end{bmatrix} \quad (2.12)$$

$$\frac{\partial f}{\partial u} = \begin{bmatrix} 0 \\ D^{-1}(q) \end{bmatrix} \quad (2.13)$$

$$F_{21}(t) = -D^{-1} \left\{ \frac{\partial D}{\partial q} \xi + \frac{\partial h}{\partial q} + \frac{\partial g}{\partial q} \right\} \quad (2.14)$$

$$F_{22}(t) = -D^{-1} \left\{ \frac{\partial \nu}{\partial \dot{q}} + \frac{\partial h}{\partial \dot{q}} \right\} \quad (2.15)$$

where

$$\xi(q, \dot{q}, u) = D^{-1}(q) \{ \tau - \nu(\dot{q}) - h(q, \dot{q}) - g(q) \}$$

The different partial derivatives in the last two equations ($\frac{\partial D}{\partial q}$, $\frac{\partial h}{\partial q}$, $\frac{\partial h}{\partial \dot{q}}$, and $\frac{\partial g}{\partial q}$) are described in appendix A. These equations can be used to compute the optimal path as shown in chapter 6 or to design a feedback controller around a nominal path, once a path is defined, equation (2.11) can be written as a linear time-variant system:

$$\delta \dot{x} = F(t) \delta x + G(t) \delta u \quad (2.16)$$

2.1.3 Model with Unknown Parameters

In equation (2.2), one assumes that all the parameters are known, but in practice this might not be true. One can form a parameter vector θ which includes the unknown parameters (e.g. mass of the load, viscous friction coefficient, inertia and mass of the links). Accordingly, equation (2.2) can now be written so it is dependent on θ :

$$\ddot{q} = D^{-1}(q, \theta) \{ u(t) - \nu(\dot{q}, \theta) - h(q, \dot{q}, \theta) - g(q, \theta) \} \quad (2.17)$$

The parameter vector θ is treated as constant vector so $\dot{\theta} = 0$. Let us define a new state vector:

$$x = \begin{bmatrix} q \\ \dot{q} \\ \theta \end{bmatrix} \quad (2.18)$$

so that

$$\dot{x} = f(x, u) \quad (2.19)$$

where

$$f(x, u) = \begin{bmatrix} \dot{q} \\ D^{-1}(q, \theta)\{u(t) - \nu(\dot{q}, \theta) - h(q, \dot{q}, \theta) - g(q, \theta)\} \\ 0 \end{bmatrix} \quad (2.20)$$

These equations represent a deterministic system, so if we want to take into account the possibility of input disturbances and parameter variations, u should be replaced by $u + w_1$ and $\dot{\theta} = 0$ by $\dot{\theta} = w_2$ (the possible parameters variations are modeled as random walk processes as suggested in [59, 72]) where w_1 and w_2 are random variables. This results in

$$\dot{x} = f(x, u) + G(x)w \quad (2.21)$$

$$G(x) = \begin{bmatrix} 0 & 0 \\ D^{-1}(q, \theta) & 0 \\ 0 & I \end{bmatrix} \quad (2.22)$$

$$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \quad (2.23)$$

As in section 2.1.2, one can define a perturbation model such as

$$\delta \dot{x} \approx \frac{\partial f(x_0, u_0)}{\partial x} \delta x + G(x_0)w \quad (2.24)$$

with

$$\frac{\partial f}{\partial x} = \begin{bmatrix} 0 & I & 0 \\ F_{21}(t) & F_{22}(t) & F_{23}(t) \\ 0 & 0 & 0 \end{bmatrix} \quad (2.25)$$

$$F_{21}(t) = -D^{-1} \left\{ \frac{\partial D}{\partial q} \xi + \frac{\partial h}{\partial q} + \frac{\partial g}{\partial q} \right\} \quad (2.26)$$

$$F_{22}(t) = -D^{-1} \left\{ \frac{\partial \nu}{\partial \dot{q}} + \frac{\partial h}{\partial \dot{q}} \right\} \quad (2.27)$$

$$F_{23}(t) = -D^{-1} \left\{ \frac{\partial D}{\partial \theta} \xi + \frac{\partial \nu}{\partial \theta} + \frac{\partial h}{\partial \theta} + \frac{\partial g}{\partial \theta} \right\} \quad (2.28)$$

$$\xi(q, \dot{q}, \theta, u) = D^{-1}(q, \theta) \{ u(t) - \nu(\dot{q}, \theta) - h(q, \dot{q}, \theta) - g(q, \theta) \}$$

where all the partial derivatives are evaluated along the nominal trajectory. The different partial derivatives in equation (2.28) ($\frac{\partial D}{\partial \theta}$, $\frac{\partial \nu}{\partial \theta}$, $\frac{\partial h}{\partial \theta}$, and $\frac{\partial g}{\partial \theta}$) are described in appendix A. These equations can be written as a linear time-variant system:

$$\delta \dot{x} = F(t) \delta x + G(t) w \quad (2.29)$$

Equation (2.29) describes a linearization of the nonlinear manipulator dynamics about the nominal or desired trajectory. Therefore, one can use the model dynamics in equation (2.29) to derive the extended Kalman filter.

2.2 Parameter Estimation Problem

In this section, we briefly outline the problem of the determination of the inertial parameters of robotic manipulators. The problem is linear in the parameters and dependent on the available measurements—the input torques, the joint positions, velocities and accelerations. Different estimators for the parameters are presented. Solutions to this problem will be treated from the adaptive control point of view in chapters 3 and 4.

2.2.1 Problem Statement

Using Lagrangian dynamics, the robot model can be written as (see section 2.1):

$$\tau = D(q)\ddot{q} + h(q, \dot{q}) + g(q) + \nu(\dot{q}) \quad (2.30)$$

Since this equation is linear in the parameters (see Atkeson [8] and Khosla [50]), we have

$$\tau = Y(q, \dot{q}, \ddot{q})\theta \quad (2.31)$$

where

$$Y(q, \dot{q}, \ddot{q}) = \left\{ \frac{\partial D}{\partial \theta} \ddot{q} + \frac{\partial \nu}{\partial \theta} + \frac{\partial h}{\partial \theta} + \frac{\partial g}{\partial \theta} \right\} \quad (2.32)$$

Since friction forces are linear in the friction coefficients, equation (2.31) was written to include these forces with the friction coefficients as part of θ . If τ , $\ddot{q}$, $\dot{q}$ and q are measurable¹, τ and $Y(q, \dot{q}, \ddot{q})$ are known quantities while θ is the only unknown term in equation (2.31). These results are well known and can be found in [1, 7, 27, 49, 50].

2.2.2 Identifiability of the Parameters

Each link is characterized by 10 inertial parameters (see appendix A), thus a n degree-of-freedom robot has $10n$ parameters. But from the lagrangian formulation, only the working forces and the generalized coordinates appear in equation (2.31). This results in the loss of identifiability of some parameters as shown in [1, 7, 27, 49, 50]. Therefore, there are three categories of parameters:

- fully identifiable parameters,
- parameters identifiable in linear combinations,

¹In this work, the word *measurable* is used in the sense of *experimentally measurable* and not in the sense of measure theory.

- completely unidentifiable parameters.

Thus, we can define a reduced set of parameters which will completely describe the robot. This new parameter set θ will have $\dim(\theta) \leq 10n$.

2.2.3 Choice of the Parameters to be Estimated

In the off-line estimation problem, there is no problem in identifying all the parameters. However, when the estimation is done on-line, the parameters can be split in two classes:

- known parameters (load-free manipulator estimated parameters),
- unknown parameters (load characteristics).

The constant parameters of the load free manipulator can be estimated from an experiment using off-line estimation procedures. Therefore, in subsequent tasks, only the parameters that are known to vary should be estimated. This is done to reduce the number of parameters to be estimated, hence the computational burden of the on-line estimation algorithms is also reduced. Equation (2.31) can then be written as

$$\tau = Y_1(q, \dot{q}, \ddot{q})\theta_k + Y_2(q, \dot{q}, \ddot{q})\theta_u \quad (2.33)$$

where θ_k and θ_u are respectively the known (or previously estimated) and unknown parameter vectors. In the sequel, all the estimators and adaptive controllers will be developed as if all parameters were unknown but it is a very simple task to modify these methods so as to only identify the unknown parameters.

2.2.4 Possible Estimators

Different estimators can be used to identify θ . The choice of the estimator is dependent on the available measurements. Three cases will be treated.

Measurable τ , q , $\dot{q}$ and $\ddot{q}$

If τ , q , $\dot{q}$ and $\ddot{q}$ are measurable, then τ and $Y(q, \dot{q}, \ddot{q})$ are known quantities and equation (2.31) is in a linear regression form with τ as the measured quantity, $Y(q, \dot{q}, \ddot{q})$ as the regressor and θ as the unknown. One can use recursive least squares (RLS) to estimate θ either continuously or discretely (see appendix B for a description of the RLS). Convergence of the estimates depends on the properties of $Y(q, \dot{q}, \ddot{q})$. Therefore, in an experiment, the path should be sufficiently *rich* to guarantee that some positive and finite constants σ , γ_1 and γ_2 exist such that for all t_1

$$\gamma_1 I \leq \int_{t_1}^{t_1+\sigma} Y^T(q, \dot{q}, \ddot{q}) Y(q, \dot{q}, \ddot{q}) dt \leq \gamma_2 I \quad (2.34)$$

This is the persistent excitation criterion (see appendix B for details).

Measurable τ , q , and $\dot{q}$

The assumption that $\ddot{q}$ is available is in most cases not valid. Robot manipulators are usually equipped such that only q and $\dot{q}$ are measurable. Thus, $\ddot{q}$ has to be obtained by direct differentiation and/or filtering. In most cases, the resulting $\ddot{q}$ can be very noisy. To avoid this problem it has been suggested [37, 63], that one filters both sides of equation (2.31) with the operator

$$F = \frac{a}{\mathcal{D} + a} \quad (2.35)$$

where a is a positive constant and $\mathcal{D}$ is the differential operator. Since θ is assumed to be constant, this results in

$$\tau_f = Y_f(q, \dot{q})\theta \quad (2.36)$$

where

$$\tau_f = F\tau \quad (2.37)$$

$$Y_f(q, \dot{q}) = F Y(q, \dot{q}, \ddot{q}) \quad (2.38)$$

The filtering of $Y(q, \dot{q}, \ddot{q})$ has the effect of eliminating $\ddot{q}$ from $Y(q, \dot{q}, \ddot{q})$ (see chapter 3, section 3.2.2).

Equation (2.36) is still in a linear regression form, thus RLS can be applied to the filtered τ_f and $Y_f(q, \dot{q})$ to estimate θ . The richness condition is now

$$\gamma_1 I \leq \int_{t_1}^{t_1+\sigma} Y_f^T(q, \dot{q}) Y_f(q, \dot{q}) dt \leq \gamma_2 I \quad (2.39)$$

Measurable τ and q

The velocity $\dot{q}$ is usually obtained from tachometer measurements. These have a high noise content unless high quality (and very expensive) tachometers are used. Also, some manipulators are simply not equipped with velocity sensors. Trying to avoid direct differentiation of the resolver signals (q), one should use an estimator that requires only τ and q . This can be done using an extended Kalman filter (EFK) (see section 2.3) based on equations (2.21)–(2.28). This will be treated in detail in chapter 4.

2.3 Kalman Filters

In this section, we briefly state known formulations and results for Kalman filters and extended Kalman filters. Theorem 2.1 is particularly important because it will be referred to in chapter 4 to prove the stability of the adaptive control algorithm.

2.3.1 Continuous-Time Filters

Continuous-Time Kalman Filter

The following results is taken from [47]. Let a system state $x(t)$ be generated by the following model²:

$$\dot{x}(t) = F(t)x(t) + G(t)w(t) \quad (2.40)$$

For which we can observe

$$z(t) = H(t)x(t) + v(t) \quad (2.41)$$

where $w(t)$ and $v(t)$ are gaussian independent random processes with the following statistics:

$$\begin{aligned} \mathcal{E}\{w(t)\} &= 0 \\ \text{cov}\{w(t), w(\tau)\} &= Q(t)\delta_D(t - \tau) \\ \mathcal{E}\{v(t)\} &= 0 \\ \text{cov}\{v(t), v(\tau)\} &= R(t)\delta_D(t - \tau) \\ \text{cov}\{w(t), v(\tau)\} &= \text{cov}\{w(t), x(t_o)\} = 0 \\ \text{cov}\{v(t), x(\tau)\} &= 0 \end{aligned} \quad (2.42)$$

The optimal estimate of $x(t)$ using $z(\tau)$ for $t_o \leq \tau \leq t$ is $\hat{x}(t)$ and is given by the filter:

$$\frac{d\hat{x}(t)}{dt} = F(t)\hat{x}(t) + P(t)H^T(t)R^{-1}(t)\{z(t) - H(t)\hat{x}(t)\} \quad (2.43)$$

where $P(t)$ is given by the error variance algorithm:

$$\frac{dP}{dt} = F(t)P(t) + P(t)F^T(t) + G(t)Q(t)G^T(t) - P(t)H^T(t)R^{-1}(t)H(t)P(t) \quad (2.44)$$

²This representation of the system and observation models is not rigorous since a stochastic differential equation should be used to model such systems [39]. However, we shall use this representation since the reader not familiar with stochastic processes theory will probably be more at ease with this notation.

for which the initial conditions are:

$$\begin{aligned}\hat{x}(t_o) &= \mathcal{E}\{x(t_o)\} \\ P(t_o) &= \text{var}\{\tilde{x}(t_o)\}\end{aligned}\tag{2.45}$$

Theorem 2.1 *Assume that the model of the system is*

1. *uniformly completely R observable³*
2. *uniformly completely Q controllable*
3. $\alpha_1 \leq \|Q(t)\| \leq \alpha_2$
4. $\alpha_3 \leq \|R(t)\| \leq \alpha_4$
5. $\|F(t)\| \leq \alpha_5, \|G(t)\| \leq \alpha_6, \|H(t)\| \leq \alpha_7,$

Then the following is true

1. *The unforced optimum filter*

$$\frac{d\tilde{x}}{dt} = [F(t) - P(t)H^T(t)R^{-1}(t)H(t)]\tilde{x}$$

is uniformly asymptotically stable

2. *Every solutions $\Pi(t, P_1(t_o), t_o)$ and $\Pi(t, P_2(t_o), t_o)$ of the Riccati equation (variance equation(2.44)) satisfies for $t > t_o$*

$$\|\Pi(t, P_1(t_o), t_o) - \Pi(t, P_2(t_o), t_o)\| \leq Ae^{-\beta(t-t_o)}$$

with $A > 0, \beta > 0$ depending only on $P_1(t_o), P_2(t_o)$.

³The reader not familiar with the notions of observability and controllability should refer to appendix C for information on these topics.

The reader is referred to either [12, 14, 15, 43, 45, 47] for the proof. This proof is based on

$$V(\tilde{x}) = \tilde{x}^T P^{-1}(t) \tilde{x}$$

as a Lyapunov function. Uniform complete controllability establishes a lower bound on $P(t)$, hence an upper bound on $P^{-1}(t)$. Uniform complete observability establishes an upper bound on $P(t)$, hence a lower bound on $P^{-1}(t)$. Thus, the Lyapunov function is valid. Its derivative is given by

$$\dot{V}(\tilde{x}) = -\tilde{x}^T [H^T R^{-1} H + P^{-1} G Q G^T P^{-1}] \tilde{x} \leq -\beta \|\tilde{x}\|^2$$

For the stochastic case with perfect knowledge of the prior statistics, $P(t)$ is the covariance of the estimates, hence the Lyapunov function $\tilde{x}^T P^{-1}(t) \tilde{x}$ is a random variable with a chi-squared distribution of d degrees of freedom ($\chi^2(d)$, $d = \dim \tilde{x}$).

Under less restrictive conditions, the Kalman filter can be shown to be asymptotically stable (see [2]).

Continuous-Time Extended Kalman Filter

Since the applications made in this work deal with observations which are constant linear combinations of the states, the observation matrix $H(t)$ is not a function of time, thus $H(t)$ has been replaced by H . The filtering technique described in the previous section can be extended to consider the nonlinear system model⁴:

$$\dot{x}(t) = f[x(t), t] + G[x(t), t]w(t) \quad (2.46)$$

⁴Once more, we are not using the stochastic differential equation formulation. Furthermore, to be rigorous, the equivalent stochastic differential equation should be solved using Ito stochastic integral. However, standard integration rules can be used since the white noise processes are approximations for correlated noise with very small correlation time. This is consistent with the modeling of the physical system we are dealing with. For more details about stochastic differential equations, Ito calculus and modeling, one can refer to [39].

and the observations:

$$z(t) = Hx(t) + v(t) \quad (2.47)$$

where $w(t)$ and $v(t)$ are gaussian independent random processes with the following statistics:

$$\begin{aligned} \mathcal{E}\{w(t)\} &= 0 \\ \text{cov}\{w(t), w(\tau)\} &= Q(t)\delta_D(t - \tau) \\ \mathcal{E}\{v(t)\} &= 0 \\ \text{cov}\{v(t), v(\tau)\} &= R(t)\delta_D(t - \tau) \\ \text{cov}\{w(t), v(\tau)\} &= \text{cov}\{w(t), x(t_0)\} = 0 \\ \text{cov}\{v(t), x(\tau)\} &= 0 \end{aligned} \quad (2.48)$$

If the value of $f[x(t), t]$ can be approximated by

$$f[x(t), t] \approx f[\hat{x}(t), t] + \frac{\partial f}{\partial \hat{x}}[x(t) - \hat{x}(t)]$$

where terms of order higher than one in the series expansion are neglected and $\hat{x}(t)$ is an estimate of $x(t)$ ⁵, the following filter can be used:

$$\frac{d\hat{x}(t)}{dt} = f[\hat{x}(t), t] + P(t)H^T R^{-1}(t)\{z(t) - H\hat{x}(t)\} \quad (2.49)$$

where $P(t)$ is given by the approximate error variance algorithm:

$$\frac{dP}{dt} = \frac{\partial f}{\partial \hat{x}} P(t) + P(t) \frac{\partial f^T}{\partial \hat{x}} + G(t)Q(t)G^T(t) - P(t)H^T R^{-1}(t)HP(t) \quad (2.50)$$

⁵In general, the EKF does not results in $\hat{x}(t)$ being the conditional mean of $x(t)$. However, Picard [67] showed under which conditions this estimate tends to the conditional mean as the noise tends to 0. As stated by Picard: "Under these conditions, the extended Kalman filter provides a good suboptimal filter". In our case the observation are linear, hence, for the EKF to work properly, the noise should be small, $f[x(t), t]$ and $G[x(t), t]$ should be continuous with bounded and Lipschitz derivatives and the initial conditions should be in an initial layer as stated in Theorem 3.1.1 of [67].

where the initial conditions are:

$$\begin{aligned}\hat{x}(t_o) &= \mathcal{E}\{x(t_o)\} \\ P(t_o) &= \text{var}\{\tilde{x}(t_o)\}\end{aligned}\tag{2.51}$$

The reader is referred to [11, 39, 67, 71, 72] for a complete discussion relative to the extended Kalman filter.

2.3.2 Discrete-Time Filters

Discrete-Time Kalman Filter

The following results are taken from [44]. Let a system state $x(k)$ be generated by the following model:

$$x(k+1) = \Phi(k+1, k)x(k) + \Gamma(k)w(k)\tag{2.52}$$

For which we can observe

$$z(k) = H(k)x(k) + v(k)\tag{2.53}$$

where $w(k)$ and $v(k)$ are gaussian independent random sequences with the following statistics:

$$\begin{aligned}\mathcal{E}\{w(k)\} &= 0 \\ \text{cov}\{w(k), w(j)\} &= Q(k)\delta_D(k-j) \\ \mathcal{E}\{v(k)\} &= 0 \\ \text{cov}\{v(k), v(j)\} &= R(k)\delta_D(k-j) \\ \text{cov}\{w(k), v(j)\} &= \text{cov}\{w(k), x(k_o)\} = 0 \\ \text{cov}\{v(k), x(j)\} &= 0\end{aligned}\tag{2.54}$$

The optimal estimate of $x(k)$ using $z(j)$ for $k_o \leq j \leq k$ is $\hat{x}(k)$ and is given by the filter:

$$\hat{x}(k+1) = \hat{x}(k+1|k) + K(k+1)\{z(k+1) - H(k)\hat{x}(k+1|k)\}\tag{2.55}$$

Here, $\hat{x}(k+1|k)$ is the estimate of $x(k+1)$ using measurements up to $z(k)$ and it comes from the one step predictor:

$$\hat{x}(k+1|k) = \Phi(k+1, k)\hat{x}(k) \quad (2.56)$$

The filter gain algorithm is:

$$K(k+1) = V(k+1)H^T(k)R^{-1}(k+1) \quad (2.57)$$

where $V(k+1)$ is computed using the prior error variance algorithm:

$$V(k+1|k) = \Phi(k+1, k)V(k)\Phi^T(k+1, k) + \Gamma(k)Q(k)\Gamma^T(k) \quad (2.58)$$

and the error variance algorithm:

$$\begin{aligned} V(k+1) &= V(k+1|k) \\ &\quad - V(k+1|k)H^T(k)[H(k)V(k+1|k)H^T(k) + R(k+1)]^{-1}H(k)V(k+1|k) \end{aligned} \quad (2.59)$$

for which the initial conditions are:

$$\begin{aligned} \hat{x}(k_o) &= \mathcal{E}\{x(k_o)\} \\ V(k_o) &= \text{var}\{\tilde{x}(k_o)\} \end{aligned} \quad (2.60)$$

Theorem 2.2 Assume that the model of the system is

1. uniformly completely R observable
2. uniformly completely Q controllable
3. $\alpha_1 \leq \|Q(k)\| \leq \alpha_2$
4. $\alpha_3 \leq \|R(k)\| \leq \alpha_4$
5. $\|\Phi(k+1, k)\| \leq \alpha_5, \|\Gamma(k)\| \leq \alpha_6, \|H(k)\| \leq \alpha_7,$

Then the following is true

1. *The unforced optimum filter*

$$\tilde{x}(k) = [I - H(k)K(k)]\Phi(k+1, k)\tilde{x}(k)$$

is uniformly asymptotically stable

2. *Every solutions $\Pi(k, V_1(k_o), k_o)$ and $\Pi(k, V_2(k_o), k_o)$ of the Riccati equation (variance equations (2.58) and (2.59)) satisfies for $k > k_o$*

$$\|\Pi(k, V_1(k_o), k_o) - \Pi(k, V_2(k_o), k_o)\| \leq Ae^{-\beta(k-k_o)}$$

with $A > 0$, $\beta > 0$ depending only on $V_1(k_o), V_2(k_o)$.

The reader is referred to either [13, 23, 39] for the proof.

Under less restrictive conditions, the Kalman filter can be shown to be asymptotically stable (see [2]).

Discrete-Time Extended Kalman Filter

Since the applications made in this work deal with observations which are constant linear combinations of the states, the observation matrix $H(k)$ is not a function of time, thus $H(k)$ has been replaced by H . The filtering technique described in the previous section can be extended to consider the nonlinear system model:

$$x(k+1) = \phi[x(k), k] + \Gamma[x(k), k]w(k) \quad (2.61)$$

and the observations:

$$z(k) = Hx(k) + v(k) \quad (2.62)$$

where $w(k)$ and $v(k)$ are gaussian independent random sequences with the following statistics:

$$\mathcal{E}\{w(k)\} = 0$$

$$\begin{aligned}
\text{cov}\{w(k), w(j)\} &= Q(k)\delta_D(k-j) \\
\mathcal{E}\{v(k)\} &= 0 \\
\text{cov}\{v(k), v(j)\} &= R(k)\delta_D(k-j) \\
\text{cov}\{w(k), v(j)\} = \text{cov}\{w(k), x(k_0)\} &= 0 \\
\text{cov}\{v(k), x(j)\} &= 0
\end{aligned} \tag{2.63}$$

If the value of $\phi[x(k), k]$ can be approximated by

$$\phi[x(k), k] \approx \phi[\hat{x}(k), k] + \frac{\partial \phi}{\partial \hat{x}}[x(k) - \hat{x}(k)]$$

where terms of order higher than one in the series expansion are neglected, the following filter can be used:

$$\hat{x}(k+1) = \hat{x}(k+1|k) + K(k+1)\{z(k+1) - H\hat{x}(k+1|k)\} \tag{2.64}$$

Here, $\hat{x}(k+1|k)$ is the estimate of $x(k+1)$ using measurements up to $z(k)$ and it comes from the one step predictor:

$$\hat{x}(k+1|k) = \phi[\hat{x}(k), k] \tag{2.65}$$

The filter gain algorithm is:

$$K(k+1) = V(k+1)H^T R^{-1}(k+1) \tag{2.66}$$

where $V(k+1)$ is computed using the prior error variance algorithm:

$$V(k+1|k) = \frac{\partial \phi(k)}{\partial \hat{x}} V(k) \frac{\partial \phi^T(k)}{\partial \hat{x}} + \Gamma(k)Q(k) \quad (k) \tag{2.67}$$

and the error variance algorithm

$$\begin{aligned}
V(k+1) &= V(k+1|k) \\
&\quad - V(k+1|k)H^T [HV(k+1|k)H^T + R(k+1)]^{-1} HV(k+1|k)
\end{aligned} \tag{2.68}$$

for which with initial conditions are:

$$\begin{aligned}\hat{x}(k_o) &= \mathcal{E}\{x(k_o)\} \\ V(k_o) &= \text{var}\{\tilde{x}(k_o)\}\end{aligned}\tag{2.69}$$

The reader is referred to [11, 39, 71, 72] for a complete discussion relative to the extended Kalman filter.

Chapter 3

Adaptive Control of Robots: Review

3.1 Introduction

The purpose of this chapter is to investigate the sensitivity with respect to random disturbances of different adaptive controllers for robotic manipulators. The methods used in this review are developed in a deterministic context, are based on nonlinear dynamic cancellation and are justified using stability theory.

The theory behind each methods as well as simulations of a two-degree-of-freedom robot will be presented so as to compare the efficiency and robustness of the respective schemes. Different situations will be treated:

- position, velocity and acceleration are measurable,
- position and velocity are measurable,
- noise free measurements are available,
- noise free input torques are produced,
- measurements are corrupted with noise.

3.2 Schemes Based on Non-Linear Robot Models

This section reviews four adaptive control methods which correspond to the different cases:

- Measurements available:
 - $\tau, q, \dot{q}, \ddot{q}$
 - $\tau, q, \dot{q}$
- Adaptive controller type:
 - Direct: the controller parameters are updated directly using the tracking error. The adaptation law is completely dependent of the control law.
 - Indirect: the system parameters are estimated using the prediction error and the controller design problem is solved. The adaptation law is independent of the control law.

Table 3.1 present the methods that will be used. The stability of these systems is shown using either Lyapunov's method or Bounded-Input-Bounded-Output (BIBO) stability¹.

3.2.1 Measurable $\tau, q, \dot{q}$ and $\ddot{q}$

The case when accelerations are available is treated using direct and indirect approaches.

¹Appendix D introduces stability concepts as well as Lyapunov's method (Theorem D.1) and BIBO stability (Theorem D.2).

Table 3.1: Adaptive Control Algorithms

Controller	Type	Measurements	Controller	Adaptation	Stability
Craig [18, 19]	Direct	$\tau, q, \dot{q}, \ddot{q}$	Computed Torque	Gradient	Lyap.
Middleton [63]	Indirect	$\tau, q, \dot{q}, \ddot{q}$	Computed Torque	Least Squares	BIBO
Slotine [80, 81]	Direct	$\tau, q, \dot{q}$	Sliding Controller	Gradient	Lyap.
Middleton [63]	Indirect	$\tau, q, \dot{q}$	Computed Torque	Least Squares	BIBO

Direct Adaptive Controller

The following result is from Craig [18, 19]. Let us consider the following robot dynamic model:

$$\tau = D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) \quad (3.1)$$

where τ is the input torque vector, q is the joint position vector, $D(q)$ the positive definite symmetric mass matrix, $C(q, \dot{q})\dot{q}$ the Coriolis and centrifugal force vector and $g(q)$ the gravitational force vector.

As shown in [1, 7, 27, 49, 50], this equation is linear in the inertial parameters and can be written as

$$\tau = Y(q, \dot{q}, \ddot{q})\theta \quad (3.2)$$

where θ is the vector of possibly unknown parameters and $Y(q, \dot{q}, \ddot{q})$ is a known quantity.

Let us define a desired trajectory q_d with continuous and bounded first and second

time derivatives, and the tracking error:

$$e = q_d - q \quad (3.3)$$

and assume that an estimate $\hat{\theta}$ of θ is available. Under these assumptions, the following control law (this is the well known computed torque method):

$$\tau = \hat{D}(q)[\ddot{q}_d + K_v \dot{e} + K_p e] + \hat{C}(q, \dot{q})\dot{q} + \hat{g}(q) \quad (3.4)$$

where the hats denote the use of the parameter estimates to compute τ , results in the following error equation:

$$\ddot{e} + K_v \dot{e} + K_p e = \hat{D}^{-1}(q)Y(q, \dot{q}, \ddot{q})\phi \quad (3.5)$$

with

$$\phi = \theta - \hat{\theta} \quad (3.6)$$

For $\phi = 0$, one sees that error dynamics are fixed. However, ϕ is more likely to be nonzero, thus the need to try to estimate θ . Let us define the state vector:

$$x = \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \quad (3.7)$$

which is composed of the tracking error and its derivative. The error dynamics can be written as

$$\dot{x} = Ax + B\hat{D}^{-1}(q)Y(q, \dot{q}, \ddot{q})\phi \quad (3.8)$$

with

$$A = \begin{bmatrix} 0 & I \\ -K_p & -K_v \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ I \end{bmatrix}$$

If both position and velocity vectors are available, one can define a filtered error:

$$z = Cx \quad (3.9)$$

with

$$C = \begin{bmatrix} \Psi & I \end{bmatrix}$$

such that $C(sI - A)^{-1}B$ is a strictly positive real (SPR) transfer function [64]. If K_p , K_v and Ψ are diagonal matrices, the SPR condition means that

$$0 < K_{pi}$$

$$0 < \Psi_i$$

$$\Psi_i < K_{vi}$$

for all the i elements on the diagonal. This choice is made because if $C(sI - A)^{-1}B$ is SPR, then from the Kalman-Yakubovich Lemma (KYL) [64] there are positive definite matrices P and Q such that

$$(A^T P + PA) = -Q \quad (3.10)$$

$$PB = C^T \quad (3.11)$$

Let us now select a Lyapunov function candidate:

$$V(x, \phi) = x^T P x + \phi^T \Gamma^{-1} \phi \quad (3.12)$$

where P and Γ are positive definite matrices. Taking the time derivative of this function results in

$$\dot{V}(x, \phi) = x^T (A^T P + PA) x + 2\phi^T Y^T(q, \dot{q}, \ddot{q}) \hat{D}^{-1}(q) B^T P x + 2\phi^T \Gamma^{-1} \dot{\phi}$$

but from the KYL we have $A^T P + PA = -Q$, yielding

$$\dot{V}(x, \phi) = -x^T Q x + 2\phi^T (Y^T(q, \dot{q}, \ddot{q}) \hat{D}^{-1}(q) B^T P x + \Gamma^{-1} \dot{\phi}) \quad (3.13)$$

We can obtain

$$\dot{V}(x, \phi) = -x^T Q x \leq 0 \quad (3.14)$$

by selecting

$$\dot{\phi} = -\Gamma Y^T(q, \dot{q}, \ddot{q}) \hat{D}^{-1}(q) B^T P x \quad (3.15)$$

Again from the KYL, we have

$$\begin{aligned} \dot{\phi} &= -\Gamma Y^T(q, \dot{q}, \ddot{q}) \hat{D}^{-1}(q) C x \\ \dot{\phi} &= -\Gamma Y^T(q, \dot{q}, \ddot{q}) \hat{D}^{-1}(q) z \end{aligned}$$

and since θ is assumed constant, $\dot{\phi} = -\dot{\theta}$, yielding

$$\dot{\theta} = \Gamma Y^T(q, \dot{q}, \ddot{q}) \hat{D}^{-1}(q) z \quad (3.16)$$

From the fact that $\dot{V} \leq 0$, we can conclude that the system is globally stable. To establish uniform asymptotic stability for this system, we have to consider the error dynamics written as

$$\begin{bmatrix} \dot{x} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} A & B U^T \\ -\Gamma U C & 0 \end{bmatrix} \begin{bmatrix} x \\ \phi \end{bmatrix} \quad (3.17)$$

with

$$U = \left(\hat{D}^{-1}(q) Y(q, \dot{q}, \ddot{q}) \right)^T \quad (3.18)$$

for which the condition

$$\alpha' I \leq \int_t^{t+\sigma} U U^T d\tau \leq \beta' I \quad (3.19)$$

for some positive α' , β' , σ and all t , imply uniform asymptotic stability (this is the persistency excitation criterion, see [64]). Craig has shown that

$$\alpha I \leq \int_t^{t+\sigma} Y^T Y d\tau$$

is an equivalent condition as long as $\hat{D}(q)$ remains positive definite. Craig has also proposed a parameter resetting rule to maintain the positive definiteness of $\hat{D}(q)$ if a priori bounds are known for the parameters [18, 19].

We should mention that any choice of Γ which is positive definite will result in a stable system, however from the practical point of view, Γ should be chosen by trial and error to compromise between alertness of the adaptive scheme and its sensitivity to noise. There is no theoretical guide-line for the choice of Γ .

Indirect Adaptive Controller

The following result is a simplification of the method described in Middleton [63] when the joint accelerations are available. Once again, we have the manipulator dynamics:

$$\tau = D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) \quad (3.20)$$

which can be written in a linear regression form with respect to the parameter θ (see [1, 7, 27, 49, 50]):

$$\tau = Y(q, \dot{q}, \ddot{q})\theta \quad (3.21)$$

If one defines the error caused by the choice of the estimate $\hat{\theta}$ by

$$e = Y(q, \dot{q}, \ddot{q})\hat{\theta} - \tau \quad (3.22)$$

$$e = Y(q, \dot{q}, \ddot{q})(\hat{\theta} - \theta) \quad (3.23)$$

and the tracking error by

$$\tilde{q} = q - q_d \quad (3.24)$$

the use of the computed torque method based on the estimated parameters:

$$\tau = \hat{D}(q)[\ddot{q}_d - K_v\dot{\tilde{q}} - K_p\tilde{q}] + \hat{C}(q, \dot{q})\dot{q} + \hat{g}(q) \quad (3.25)$$

will give the following error equation:

$$\ddot{\tilde{q}} + K_v \dot{\tilde{q}} + K_p \tilde{q} = \hat{D}^{-1}(q)e \quad (3.26)$$

Using continuous-time unnormalized standard recursive least squares to estimate θ results in e which is in L_2 and $\hat{\theta}$, $\hat{D}(q)$ which are bounded (see appendix B.1 for this result), thus provided that $\hat{D}^{-1}(q)$ exist, $\tilde{q}$ is in L_2 and $\tilde{q}(t) \rightarrow 0$ as $t \rightarrow \infty$ since $(Is^2 + K_v s + K_p)^{-1}$ is a strictly proper exponentially stable transfer function (see Theorem D.2). In this case, when

$$\alpha I \leq \int_t^{t+\sigma} Y^T Y d\tau \leq \beta I \quad (3.27)$$

for some positive α , β , σ and all t , then the parameter error converges to zero uniformly (see appendix B.1).

3.2.2 Measurable τ , q and $\dot{q}$

In this section, direct and indirect methods that do not require acceleration measurements are presented.

Direct Adaptive Controller

The following result is taken from Slotine [80, 81]. Let us once more consider the robot model:

$$\tau = D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) \quad (3.28)$$

for which the choice of $C(q, \dot{q})$ makes $(\dot{D}(q) - 2C(q, \dot{q}))$ skew symmetric (see [66, 80, 81]). Let us define the tracking error by

$$e = q - q_d \quad (3.29)$$

the reference trajectory q_r and s by

$$\dot{q}_r = \dot{q}_d - \Lambda e \quad (3.30)$$

$$\ddot{q}_r = \ddot{q}_d - \Lambda \dot{e} \quad (3.31)$$

$$s = \dot{e} + \Lambda e \quad (3.32)$$

where Λ is a positive definite matrix. Note that the equation $s = \dot{e} + \Lambda e = 0$ defines a stable sliding surface in the tracking error. Now, if parameters estimates $\hat{\theta}$ are used with the following control law:

$$\tau = \hat{D}(q)\ddot{q}_r + \hat{C}(q, \dot{q})\dot{q}_r + \hat{g}(q) - K_d s \quad (3.33)$$

it results in the following

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \hat{D}(q)\ddot{q}_r + \hat{C}(q, \dot{q})\dot{q}_r + \hat{g}(q) - K_d s \quad (3.34)$$

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = D(q)\ddot{q}_r + C(q, \dot{q})\dot{q}_r + g(q) - K_d s \quad (3.35)$$

$$+ \tilde{D}(q)\ddot{q}_r + \tilde{C}(q, \dot{q})\dot{q}_r + \tilde{g}(q) \quad (3.36)$$

If we define the parameter error by

$$\phi = \hat{\theta} - \theta \quad (3.37)$$

then

$$Y(q, \dot{q}, \dot{q}_r, \ddot{q}_r)\phi = \tilde{D}(q)\ddot{q}_r + \tilde{C}(q, \dot{q})\dot{q}_r + \tilde{g}(q) \quad (3.38)$$

and using the definitions of q_r and s results in

$$D(q)(\ddot{q} - \ddot{q}_r) = -C(q, \dot{q})(\dot{q} - \dot{q}_r) - K_d s + Y(q, \dot{q}, \dot{q}_r, \ddot{q}_r)\phi \quad (3.39)$$

$$D(q)\dot{s} = -C(q, \dot{q})s - K_d s + Y(q, \dot{q}, \dot{q}_r, \ddot{q}_r)\phi \quad (3.40)$$

Let us choose

$$V(s, \phi) = \frac{1}{2}s^T D(q)s + \frac{1}{2}\phi^T \Gamma \phi \quad (3.41)$$

as a Lyapunov function candidate² where Γ is a positive definite matrix. The time derivative of this function is

$$\dot{V}(s, \phi) = \frac{1}{2}s^T \dot{D}(q)s + s^T D(q)\dot{s} + \phi^T \Gamma \dot{\phi} \quad (3.42)$$

Substitution of $D(q)\dot{s}$ yields

$$\begin{aligned} \dot{V}(s, \phi) &= \frac{1}{2}s^T \dot{D}(q)s + s^T (-C(q, \dot{q})s - K_d s + Y(q, \dot{q}, \ddot{q}_r, \ddot{q}_r)\phi) + \phi^T \Gamma \dot{\phi} \\ \dot{V}(s, \phi) &= \frac{1}{2}s^T (\dot{D}(q) - 2C(q, \dot{q}))s + s^T (-K_d s + Y(q, \dot{q}, \ddot{q}_r, \ddot{q}_r)\phi) + \phi^T \Gamma \dot{\phi} \end{aligned}$$

Since $(\dot{D}(q) - 2C(q, \dot{q}))$ is skew symmetric, we get

$$\dot{V}(s, \phi) = -s^T K_d s \leq 0 \quad (3.43)$$

using the update law

$$\dot{\phi} = -\Gamma^{-1} Y^T(q, \dot{q}, \ddot{q}_r, \ddot{q}_r)s \quad (3.44)$$

$$\dot{\theta} = -\Gamma^{-1} Y^T(q, \dot{q}, \ddot{q}_r, \ddot{q}_r)s \quad (3.45)$$

As with Craig results, $\dot{V} \leq 0$ imply the stability of the system. If the system is persistently excited, one should expect uniform asymptotic stability, but no theoretical results are available to prove this assumption. Also, as with Craig method, Γ^{-1} has to be chosen by trial and error.

Indirect Adaptive Controller

The following result is from Middleton [63]. Once again, we have the manipulator dynamics:

$$\tau = D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) \quad (3.46)$$

²Strictly speaking, $V(s, \phi)$ is not a Lyapunov function since it is not positive definite, however Spong et al [82] showed that this system is stable in the sense of Lyapunov and that a Lyapunov function exists for the proposed adaptive controller.

which can be written in a linear regression form with respect to the parameter θ (see [1, 7, 27, 49, 50]):

$$\tau = Y(q, \dot{q}, \ddot{q})\theta \quad (3.47)$$

This equation is dependent on $\ddot{q}$, but we can filter this equation using

$$F = \frac{a}{\mathcal{D} + a} \quad (3.48)$$

where a is a positive constant and $\mathcal{D}$ is the differential operator. Since θ is assumed to be constant, this results in

$$\tau_f = Y_f(q, \dot{q})\theta \quad (3.49)$$

$$e_f = Y_f(q, \dot{q})\hat{\theta} - \tau_f \quad (3.50)$$

where

$$\tau_f = F\tau \quad (3.51)$$

$$Y_f(q, \dot{q}) = FY(q, \dot{q}, \ddot{q}) \quad (3.52)$$

Since $Y(q, \dot{q}, \ddot{q})$ has elements of the form:

$$Y_{ij}(q, \dot{q}, \ddot{q}) = f^T(q)\ddot{q} + g(q, \dot{q}) \quad (3.53)$$

the filtering of $Y(q, \dot{q}, \ddot{q})$ has the effect of eliminating $\ddot{q}$ from $Y(q, \dot{q}, \ddot{q})$ as shown below. Let

$$y = u(t)\dot{v}(t) + w(t) \quad (3.54)$$

and

$$y_f = Fy \quad (3.55)$$

since

$$F = (a - aF)\frac{1}{\mathcal{D}} \quad (3.56)$$

and

$$\frac{1}{\mathcal{D}}y = u(t)v(t) - \frac{1}{\mathcal{D}}(\dot{u}(t)v(t) - w(t)) \quad (3.57)$$

y_f is given by

$$y_f = au(t)v(t) + F(w(t) - au(t)v(t) - \dot{u}(t)v(t)) \quad (3.58)$$

where the derivative of v has been eliminated. Now, let us use the computed torque method based on the parameter estimates:

$$\tau = \hat{D}(q)[\ddot{q}_d - K_v\dot{\tilde{q}} - K_p\tilde{q}] + \hat{C}(q, \dot{q})\dot{q} + \hat{g}(q) + \tau_c \quad (3.59)$$

where τ_c is a correction torque to be determined later. This results in the following error equation:

$$\ddot{\tilde{q}} + K_v\dot{\tilde{q}} + K_p\tilde{q} = \hat{D}^{-1}(q)(Y(q, \dot{q}, \tilde{q})\hat{\theta} - \tau + \tau_c) \quad (3.60)$$

But only the properties of e_f can be determined since only $Y_f(q, \dot{q})$, $\hat{\theta}$ and τ_f are available. Now, using continuous-time unnormalized standard recursive least squares to estimate θ results in e_f which is in L_2 and $\hat{\theta}$, $\hat{D}(q)$ which are bounded (see appendix B.1 for this result). One would want to relate $Y(q, \dot{q}, \tilde{q})\hat{\theta} - \tau$ to $Y_f(q, \dot{q})\hat{\theta} - \tau_f$. Middleton [63] solved this problem by requiring the error equation to be given by

$$\ddot{\tilde{q}} + K_v\dot{\tilde{q}} + K_p\tilde{q} = F^{-1}(\hat{D}^{-1}(q)e_f) \quad (3.61)$$

thus provided that $\hat{D}^{-1}(q)$ exist, $\tilde{q}$ is in L_2 and $\tilde{q}(t) \rightarrow 0$ as $t \rightarrow \infty$ since $F^{-1}(s)(Is^2 + K_v s + K_p)^{-1}$ is a strictly proper exponentially stable transfer function³ (see Theorem D.2). It now remains to evaluate the correction torque τ_c such that the previous assumption is true.

First, let us state the following property of the filter F (see [64]):

$$F^{-1}u(t)F = u(t) + \frac{1}{a}\dot{u}(t)F \quad (3.62)$$

³Here, $F(s)$ is the transfer function representation of the operator F

then

$$F^{-1}(\hat{D}^{-1}(q)e_f) = F^{-1}(\hat{D}^{-1}(q)FF^{-1}e_f) \quad (3.63)$$

$$= \hat{D}^{-1}(q)F^{-1}e_f + \frac{1}{a} \frac{d\hat{D}^{-1}(q)}{dt} e_f \quad (3.64)$$

$$= \hat{D}^{-1}(q)F^{-1}e_f - \frac{1}{a} D^{-1}(q)\dot{\hat{D}}(q)D^{-1}(q)e_f \quad (3.65)$$

$$F^{-1}e_f = F^{-1}(Y_f(q, \dot{q})\hat{\theta} - \tau_f) \quad (3.66)$$

$$= Y(q, \dot{q}, \ddot{q})\hat{\theta} - \tau + \frac{1}{a} Y_f(q, \dot{q})\dot{\hat{\theta}} \quad (3.67)$$

hence

$$F^{-1}(\hat{D}^{-1}(q)e_f) = \hat{D}^{-1}(q)[Y(q, \dot{q}, \ddot{q})\hat{\theta} - \tau + \frac{1}{a}(Y_f(q, \dot{q})\dot{\hat{\theta}} - \dot{\hat{D}}(q)D^{-1}(q)e_f)] \quad (3.68)$$

By comparing this result to the first error equation obtained, one sees that to assure that $\tilde{q}$ is in L_2 , τ_c is given by

$$\tau_c = \frac{1}{a}[Y_f(q, \dot{q})\dot{\hat{\theta}} - \dot{\hat{D}}(q)D^{-1}(q)e_f] \quad (3.69)$$

Therefore, the control law is given by

$$\tau = \hat{D}(q)[\ddot{q}_d - K_v\dot{\tilde{q}} - K_p\tilde{q}] + \hat{C}(q, \dot{q})\dot{q} + \hat{g}(q) + \frac{1}{a}[Y_f\dot{\hat{\theta}} - \dot{\hat{D}}(q)\hat{D}^{-1}(q)e_f] \quad (3.70)$$

In this case, when

$$\alpha I \leq \int_t^{t+\sigma} Y_f^T Y_f d\tau \leq \beta I \quad (3.71)$$

for some positive α , β , σ and all t , then the parameter error converges to zero uniformly (see appendix B.1).

3.3 Robustness to Disturbances

The system robustness to disturbances is very important. If the system is persistently excited, the uniform asymptotic stability guarantees bounded output under

bounded disturbances. Since the case when the system is not persistently excited leads to asymptotic stability (not uniform), bounded input disturbances can produce unbounded errors. This was clearly demonstrated in [68] and where robustness improvements to the adaptive schemes are suggested. We shall not discuss these modifications but investigate the unmodified schemes' behaviour in the presence of random disturbances.

3.4 Implementation

The complexity of these algorithms requires that the implementation should be done using a digital computer (controller). However, the discretization of the controllers might lead to instability since the stability of these system was proven under the assumption that a continuous-time controller was used.

This raises the question of how should the controller be discretized for the application. This is not a simple problem and falls into the area of hybrid adaptive controllers. For example, one should refer to the work of Middleton [61, 62] to have an example of the complexity of this issue.

However, one can use a discretization scheme realizable in real-time and sampled fast enough such that the system remains stable. Also, the continuous-time least squares estimator should be replaced by the discrete least squares as mentioned in appendix B.2. One more point is worth mentioning; the indirect method with filtering requires $\dot{\hat{\theta}}$ which is not available if the discrete least squares are used and one should rely on finite differences approximation to compute the correction torque. Furthermore, the filter F has to be discretized. All these modifications violate the assumptions under which the stability was proven, but high enough sampling frequency should minimize the effect these modifications.

Let us summarize how the different controllers were implemented (here Δ is the sampling interval):

- Craig:

- The torques are computed at discrete points in time and are held constant between samples
- The adaptation law is:

$$\hat{\theta}(t + \Delta) = \hat{\theta}(t) + \Delta \dot{\hat{\theta}}(t)$$

- Middleton with $\tilde{q}$:

- The torques are computed at discrete points in time and are held constant between samples
- The continuous-time least squares are replaced by the discrete-time least squares.

- Slotine:

- The torques are computed at discrete points in time and are held constant between samples
- The adaptation law is:

$$\hat{\theta}(t + \Delta) = \hat{\theta}(t) + \Delta \dot{\hat{\theta}}(t)$$

- Middleton with filter:

- The torques are computed at discrete points in time and are held constant between samples
- The continuous-time least squares are replaced by the discrete-time least squares.
- The finite differences approximation is used for

$$\dot{\hat{\theta}} = \frac{\hat{\theta}(t + \Delta) - \hat{\theta}(t)}{\Delta}$$

- The filter is discretized as

$$y_f(t + \Delta) = e^{-a\Delta} y_f(t) + (1 - e^{-a\Delta}) y(t)$$

3.5 Simulations Examples

In this section, we present simulation results for the four adaptive controllers. Three situations are treated:

- Case A: Simulation of the continuous-time system with a continuous-time controller in noiseless conditions. In this situation, an adaptive step-size Runge-Kutta integration algorithm was use in order to establish the theoretical performance of the controller.
- Case B: Simulation of the continuous-time system with an adaptive step-size Runge-Kutta with a discrete-time controller (200hz) in noiseless conditions. In this situation, the discretization effects will be seen.
- Case C: Simulation of the continuous-time system with an adaptive step-size Runge-Kutta with a discrete-time controller (200hz) with random noise in the measurements.

In the simulations, the dynamic model of the planar 2 DOF serial robot shown in figure 3.1 will be used. Defining the parameters:

$$\begin{aligned} p_1 &= m_1(c_1^2 + o_1^2) + I_1 + m_2 l_1^2 \\ p_2 &= m_2(c_2^2 + o_2^2) + I_2 \\ p_3 &= m_2 l_1 c_2 \\ p_4 &= m_2 l_1 o_2 \end{aligned}$$

and applying Lagrange's equation to this system yields (see Appendix E):

$$\tau_1 = (p_1 + p_2 + 2p_3 \cos q_2 - 2p_4 \sin q_2) \ddot{q}_1 + (p_2 + p_3 \cos q_2 - p_4 \sin q_2) \ddot{q}_2$$

$$\begin{aligned} & -(p_3 \sin q_2 + p_4 \cos q_2) \dot{q}_2 (2\dot{q}_1 + \dot{q}_2) \\ \tau_2 = & (p_2 + p_3 \cos q_2 - p_4 \sin q_2) \ddot{q}_1 + p_2 \ddot{q}_2 + (p_3 \sin q_2 + p_4 \cos q_2) \dot{q}_1^2 \end{aligned}$$

In matrix form:

$$\begin{aligned} D(q) &= \begin{bmatrix} p_1 + p_2 + 2p_3 \cos q_2 - 2p_4 \sin q_2 & p_2 + p_3 \cos q_2 - p_4 \sin q_2 \\ p_2 + p_3 \cos q_2 - p_4 \sin q_2 & p_2 \end{bmatrix} \\ C(q, \dot{q}) \dot{q} &= \begin{bmatrix} -(p_3 \sin q_2 + p_4 \cos q_2) \dot{q}_2 (2\dot{q}_1 + \dot{q}_2) \\ (p_3 \sin q_2 + p_4 \cos q_2) \dot{q}_1^2 \end{bmatrix} \\ g(q) &= 0 \end{aligned}$$

with

$$C(q, \dot{q}) = \begin{bmatrix} -(p_3 \sin q_2 + p_4 \cos q_2) \dot{q}_2 & -(p_3 \sin q_2 + p_4 \cos q_2) (\dot{q}_1 + \dot{q}_2) \\ (p_3 \sin q_2 + p_4 \cos q_2) \dot{q}_1 & 0 \end{bmatrix}$$

It is straight forward to put these equations in the form of $\tau = Y(q, \dot{q}, \ddot{q})\theta$ with the parameter vector:

$$\theta = \begin{bmatrix} p_1 = m_1(c_1^2 + o_1^2) + I_1 + m_2 l_1^2 \\ p_2 = m_2(c_2^2 + o_2^2) + I_2 \\ p_3 = m_2 l_1 c_2 \\ p_4 = m_2 l_1 o_2 \end{bmatrix}$$

and the regression matrix:

$$Y^T(q, \dot{q}, \ddot{q}) = \begin{bmatrix} \ddot{q}_1 & 0 \\ \ddot{q}_1 + \ddot{q}_2 & \ddot{q}_1 + \ddot{q}_2 \\ \cos(q_2)(2\ddot{q}_1 + \ddot{q}_2) - \sin(q_2)\dot{q}_2(2\dot{q}_1 + \dot{q}_2) & \cos(q_2)\ddot{q}_1 + \sin(q_2)\dot{q}_1^2 \\ -\sin(q_2)(2\ddot{q}_1 + \ddot{q}_2) - \cos(q_2)\dot{q}_2(2\dot{q}_1 + \dot{q}_2) & -\sin(q_2)\ddot{q}_1 + \cos(q_2)\dot{q}_1^2 \end{bmatrix}$$

The filtered regression matrix is:

$$Y_f^T(q, \dot{q}) = a \begin{bmatrix} \dot{q}_1 & 0 \\ \dot{q}_1 + \dot{q}_2 & \dot{q}_1 + \dot{q}_2 \\ \cos(q_2)(2\dot{q}_1 + \dot{q}_2) & \cos(q_2)\dot{q}_1 \\ -\sin(q_2)(2\dot{q}_1 + \dot{q}_2) & -\sin(q_2)\dot{q}_1 \end{bmatrix}$$

$$+ F \begin{bmatrix} -a\dot{q}_1 & 0 \\ -a(\dot{q}_1 + \dot{q}_2) & -a(\dot{q}_1 + \dot{q}_2) \\ -a \cos(q_2)(2\dot{q}_1 + \dot{q}_2) & \sin(q_2)(\dot{q}_1^2 + \dot{q}_1\dot{q}_2) - a \cos(q_2)\dot{q}_1 \\ a \sin(q_2)(2\dot{q}_1 + \dot{q}_2) & \cos(q_2)(\dot{q}_1^2 + \dot{q}_1\dot{q}_2) + a \sin(q_2)\dot{q}_1 \end{bmatrix}$$

The robot is assumed to be composed of two rectangular links of length $0.3m$ and mass $2kg$, hence

$$\theta = \begin{bmatrix} p_1 = 0.24 \text{ kgm}^2 \\ p_2 = 0.06 \text{ kgm}^2 \\ p_3 = 0.09 \text{ kgm}^2 \\ p_4 = 0 \text{ kgm}^2 \end{bmatrix}$$

The desired path is given by

$$q_1 = \arccos(0.45 - 0.45 \sin(\omega t + \pi/2))$$

$$q_2 = -2q_1$$

with $\omega = 5$ (period of $T_p = 1.255$ sec) and is shown in figure 3.2. This correspond to the robot and the path described in [75].

All simulations will be started with the parameter estimates:

$$\hat{\theta} = \begin{bmatrix} \hat{p}_1 = 0.3 \text{ kgm}^2 \\ \hat{p}_2 = 0.2 \text{ kgm}^2 \\ \hat{p}_3 = 0.1 \text{ kgm}^2 \\ \hat{p}_4 = 0 \text{ kgm}^2 \end{bmatrix}$$

For the computed torque method, we want the system error dynamics to have a damping coefficient of $\zeta = 1/\sqrt{2}$ and a natural frequency of $\omega_n = 10$ yielding

$$K_v = 14.1I$$

$$K_p = 100I$$

This correspond to settling time of $T_s = 0.4$ sec (5% criterion).

For sliding controller we have chosen

$$\Lambda = 10I$$

$$K_d = 1I$$

which give a similar time response to the computed torque method (see [80, 81]).

Figure 3.3 show the respective errors for the controllers.

Now, let us present the simulation results for the three cases

- Case A:

- Craig:

$$\Psi = 4I$$

$$\Gamma = 0.001I$$

- Indirect with $\tilde{q}$:

$$P(0) = 0.1I$$

$$\alpha_1 = 0$$

$$\alpha_2 = 1$$

- Slotine:

$$\Gamma^{-1} = 0.02I$$

- Indirect with filter:

$$P(0) = 0.1I$$

$$\alpha_1 = 0$$

$$\alpha_2 = 1$$

$$a = 10$$

The results (see fig. 3.4–3.11) show the asymptotic behaviour of the errors. For the indirect methods, the continuous-time forgetting factors were zero but non-zero α_1 would have increased the speed of convergence the parameters. Craig and Slotine methods are gradient schemes, hence these have slower convergence than the indirect methods which are based on least squares estimators. All the tracking errors have reasonable values.

- Case B:

- Craig:

$$\Psi = 4I$$

$$\Gamma = 0.001I$$

- Indirect with $\tilde{q}$:

$$P(0) = 0.001I$$

$$\lambda = 0.999$$

- Slotine:

$$\Gamma^{-1} = 0.02I$$

- Indirect with filter:

$$P(0) = 0.001I$$

$$\lambda = 0.999$$

$$a = 10$$

The discretization process (see fig. 3.12–3.19) produce almost identical results for the direct methods. For the indirect methods some forgetting was used ($\lambda = 0.999$) which correspond to a time constant of 5 seconds. The parameter

errors goes quickly to zero. With $\lambda = 1$, the indirect method gives results very similar to the indirect method Case A. One notices that the tracking errors cannot go to zero even for zero parameter errors since the control laws are implemented in discrete form.

- Case C: Like Case B but with the measurement noise (gaussian)

$$\sigma_{pos} = 0.00045 \text{ rad, (12 bits quantization error)}$$

$$\sigma_{vel} = 0.1 \text{ rad/sec}$$

$$\sigma_{acc} = 0.5 \text{ rad/sec}^2$$

In this situation the controllers are clearly less efficient (see fig. 3.20–3.27). Craig and Slotine methods produce bounded parameter and tracking errors but simulations with higher level of noise caused the direct methods to diverge. The indirect method with $\tilde{q}$ seems more robust and the parameters converged to biased values. This is caused by the fact that regressors are no longer deterministic but stochastic. The indirect method with filtering seems less robust and showed an interesting behaviour: the parameters are drifting slowly and at some point the errors get very high (see fig. 3.27) and results in some resetting action. However the tracking errors are much higher than the other methods. Also, the correction torque which is based on $\dot{\hat{\theta}}$ will contain some amount of noise.

In all methods, the measurements of position and velocity are directly fed back by the controller, this results in noisy inputs that might excite the unmodeled dynamics of the robot and this might be problematic.

3.6 Conclusion

In this chapter, we have compared different adaptive controllers. All these can be discretized successfully to be implemented using a digital computer, however the

presence of random noise in the measurements leads to a deterioration of the performance of these controllers. Also, no guide-lines are available on how to select the adaptation gains in order to maintain alertness to parameter changes and at the same time not be overly sensitive to noise. For the indirect methods, the forgetting factor can vary with time such that alertness to parameter variations can be easily changed. Some modifications suggested by [68] can guarantee the robustness of the scheme to disturbances but again the controller has to be tuned by trial and error. In Chapter 4, we intend to address the optimal filtering solution to this problem.

Figure 3.1: 2 DOF Robot Schematic

Figure 3.2: Desired Path q_{1d} = solid line, $\dot{q}_{1d}/5$ = dashed line, $\ddot{q}_{1d}/10$ = dotted line

Figure 3.3: Tracking Error: Computed Torque: $q_1 - q_{1c}$ = solid line, $q_2 - q_{2c}$ = dashed line, Sliding Controller: $q_1 - q_{1d}$ = dotted line, $q_2 - q_{2d}$ = dash-dot line

Figure 3.4: Craig: Continuous-Time Controller: Parameter Errors: $\hat{p}_1 - p_1$ = solid line, $\hat{p}_2 - p_2$ = dashed line, $\hat{p}_3 - p_3$ = dotted line, $\hat{p}_4 - p_4$ = dash-dot line

Figure 3.5: Craig: Continuous-Time Controller: Tracking Errors: $q - q_{1d}$ = solid line, $q_2 - q_{2d}$ = dashed line

Figure 3.6: Indirect: Continuous-Time Controller: Parameter Errors: $\hat{p}_1 - p_1$ = solid line, $\hat{p}_2 - p_2$ = dashed line, $\hat{p}_3 - p_3$ = dotted line, $\hat{p}_4 - p_4$ = dash-dot line

Figure 3.7: Indirect: Continuous-Time Controller: Tracking Errors: $q_1 - q_{1d}$ = solid line, $q_2 - q_{2d}$ = dashed line

Figure 3.8: Slotine: Continuous-Time Controller: Parameter Errors: $\hat{p}_1 - p_1$ = solid line, $\hat{p}_2 - p_2$ = dashed line, $\hat{p}_3 - p_3$ = dotted line, $\hat{p}_4 - p_4$ = dash-dot line

Figure 3.9: Slotine: Continuous-Time Controller: Tracking Errors: $q_1 - q_{1d}$ = solid line, $q_2 - q_{2d}$ = dashed line

Figure 3.10: Indirect with Filter: Continuous-Time Controller: Parameter Errors:
 $\hat{p}_1 - p_1$ = solid line, $\hat{p}_2 - p_2$ = dashed line, $\hat{p}_3 - p_3$ = dotted line, $\hat{p}_4 - p_4$ = dash-dot line

Figure 3.11: Indirect with Filter: Continuous-Time Controller: Tracking Errors:
 $q_1 - q_{1d}$ = solid line, $q_2 - q_{2d}$ = dashed line

Figure 3.12: Craig: Discrete-Time Controller: Parameter Errors: $\hat{p}_1 - p_1$ = solid line, $\hat{p}_2 - p_2$ = dashed line, $\hat{p}_3 - p_3$ = dotted line, $\hat{p}_4 - p_4$ = dash-dot line

Figure 3.13: Craig: Discrete-Time Controller: Tracking Errors: $q_1 - q_{1d}$ = solid line, $q_2 - q_{2d}$ = dashed line

Figure 3.14: Indirect: Discrete-Time Controller: Parameter Errors: $\hat{p}_1 - p_1$ = solid line, $\hat{p}_2 - p_2$ = dashed line, $\hat{p}_3 - p_3$ = dotted line, $\hat{p}_4 - p_4$ = dash-dot line

Figure 3.15: Indirect: Discrete-Time Controller: Tracking Errors: $q_1 - q_{1d}$ = solid line, $q_2 - q_{2d}$ = dashed line

Figure 3.16: Slotine: Discrete-Time Controller: Parameter Errors: $\hat{p}_1 - p_1$ = solid line, $\hat{p}_2 - p_2$ = dashed line, $\hat{p}_3 - p_3$ = dotted line, $\hat{p}_4 - p_4$ = dash-dot line

Figure 3.17: Slotine: Discrete-Time Controller: Tracking Errors: $q_1 - q_{1d}$ = solid line, $q_2 - q_{2d}$ = dashed line

Figure 3.18: Indirect with Filter: Discrete-Time Controller: Parameter Errors: $\hat{p}_1 - p_1$ = solid line, $\hat{p}_2 - p_2$ = dashed line, $\hat{p}_3 - p_3$ = dotted line, $\hat{p}_4 - p_4$ = dash-dot line

Figure 3.19: Indirect with Filter: Discrete-Time Controller: Tracking Errors: $q_1 - q_{1d}$ = solid line, $q_2 - q_{2d}$ = dashed line

Figure 3.20: Craig with Noise: Discrete-Time Controller: Parameter Errors: $\hat{p}_1 - p_1$ = solid line, $\hat{p}_2 - p_2$ = dashed line, $\hat{p}_3 - p_3$ = dotted line, $\hat{p}_4 - p_4$ = dash-dot line

Figure 3.21: Craig with Noise: Discrete-Time Controller: Tracking Errors: $q_1 - q_{1d}$ = solid line, $q_2 - q_{2d}$ = dashed line

Figure 3.22: Indirect with Noise: Discrete-Time Controller: Parameter Errors: $\hat{p}_1 - p_1$ = solid line, $\hat{p}_2 - p_2$ = dashed line, $\hat{p}_3 - p_3$ = dotted line, $\hat{p}_4 - p_4$ = dash-dot line

Figure 3.23: Indirect with Noise: Discrete-Time Controller: Tracking Errors: $q_1 - q_{1d}$ = solid line, $q_2 - q_{2d}$ = dashed line

Figure 3.24: Slotine with Noise: Discrete-Time Controller: Parameter Errors: $\hat{p}_1 - p_1$ = solid line, $\hat{p}_2 - p_2$ = dashed line, $\hat{p}_3 - p_3$ = dotted line, $\hat{p}_4 - p_4$ = dash-dot line

Figure 3.25: Slotine with Noise: Discrete-Time Controller: Tracking Errors: $q_1 - q_{1d}$ = solid line, $q_2 - q_{2d}$ = dashed line

Figure 3.26: Indirect with Filter and Noise: Discrete-Time Controller: Parameter Errors: $\hat{p}_1 - p_1$ = solid line, $\hat{p}_2 - p_2$ = dashed line, $\hat{p}_3 - p_3$ = dotted line, $\hat{p}_4 - p_4$ = dash-dot line

Figure 3.27: Indirect with Filter and Noise: Discrete-Time Controller: Tracking Errors: $q_1 - q_{1d}$ = solid line, $q_2 - q_{2d}$ = dashed line

Chapter 4

The New Adaptive Controller

4.1 Introduction

In chapter 3, it has been shown that the motion adaptive controllers reviewed were sensitive to measurement noise and that biased parameters will most likely be produced. Also, measurements were directly used in the feedback of these control laws, such that unmodeled dynamics might be excited by these *noisy* inputs.

One solution to this problem is to introduce filtering of the measurements so as to obtain smooth signals, but these filters, when introduced in the feedback loop, can destabilize the controller since they introduce phase shift in the filtered signals. However, we must point out that in practice, the measured positions are very accurate and are often used to obtain velocities and accelerations by direct differentiation. Nevertheless, these velocities and accelerations have to be filtered since the noise content of the position measurements is greatly enhanced by direct differentiation.

Thus, we are seeking a way of using only position measurements so as to obtain a controller in which direct differentiation is never used. In order to achieve this goal, we will use the computed torque method combined with an extended Kalman filter (EKF) producing estimates of positions, velocities and parameters.

The use of EKF in adaptive control goes back to the paper of Kopp and Orford [53] who presented results for a second order system for which the controller uses

the EKF outputs to adjust the control law and to generate the errors used in the feedback. These results showed the noise rejection properties of the proposed scheme, however, the stability of the overall system was not established. As one can see from the literature in adaptive control (see for example [3, 5, 64]), this approach was certainly not favored since the stability analysis of such systems is often very hard (sometimes impossible) to achieve and that results are limited to the special cases treated. Hence, most results present an *ad hoc* application of the technique where experimental and/or simulation results are provided to confirm the usefulness of the approach.

In this chapter, we present the main contribution of this thesis. The new adaptive controller is presented along with a proof of its stability in section 4.2. Modification for improved robustness are presented in section 4.3. Section 4.4 describes the suggested discrete-time implementation and examples are found in section 4.5.

4.2 Adaptive Control Algorithm

The new scheme is an indirect adaptive controller. The control law is based on the computed torque method using position, velocity and parameter estimates. Therefore the control torque vector is given by

$$\tau = \hat{D}(\hat{q})[\ddot{q}_d + K_v(\dot{q}_d - \dot{\hat{q}}) + K_p(q_d - \hat{q})] + \hat{h}(\hat{q}, \dot{\hat{q}}) + \hat{g}(\hat{q}) + \hat{v}(\dot{\hat{q}}) \quad (4.1)$$

where the hats denote the use of the estimated values of q , $\dot{q}$ and θ . The estimates are provided by the application of the continuous-time extended Kalman filter described by equations (2.49)–(2.51) to the robot model based on equations (2.21)–(2.28) with

$$H = \begin{bmatrix} I & 0 & 0 \end{bmatrix}$$

which correspond to position measurements only. The controller structure is shown in figure 4.1 where CTM denotes the computed torque method and EKF the extended Kalman filter. The EKF has two purposes: to provide estimates of the parameters,

and to reconstruct the velocity from position measurements. This results in a locally stable controller. We shall demonstrate that the unperturbed system ($w(t) = v(t) = 0$) is locally stable. Thus, we are dealing with deterministic case where $R(t)$ and $Q(t)$ are design parameters that are chosen such that the nonlinear observer built using the EKF structure is locally stable. However, the stochastic interpretation of the Kalman filter will allow us to select $R(t)$ and $Q(t)$ (the covariance matrices) from practical considerations (see section 4.4).

Theorem 4.1 *The unperturbed system ($w(t) = v(t) = 0$) described by equations (2.21)–(2.28) with the control law (4.1) and the filtering described by equations (2.49)–(2.51) is locally stable (the tracking error $e \in L_2 \cap L_\infty$), provided that*

1. K_v and K_p are bounded and positive definite matrices,
2. q_d , $\dot{q}_d$ and $\ddot{q}_d$ are bounded and continuous,
3. $\hat{D}(\hat{q})$ is positive definite,
4. $\alpha_1 I \leq R(t) \leq \alpha_2 I$,
5. $\alpha_3 I \leq Q(t) \leq \alpha_4 I$,
6. $H = \begin{bmatrix} I & 0 & 0 \end{bmatrix}$,
7. the following is true:

$$\alpha_5 I \leq \int_t^{t+\sigma} F_{23}^T(\tau) F_{23}(\tau) d\tau \leq \alpha_6 I$$

where $F_{23}(\tau)$ is evaluated along $\hat{x}$. with

$$F_{23}(\tau) = \left[-D^{-1} \left\{ \frac{\partial D}{\partial \theta} \xi + \frac{\partial \nu}{\partial \theta} + \frac{\partial h}{\partial \theta} + \frac{\partial g}{\partial \theta} \right\} \right]_{\text{along } \hat{x}}$$

for some positive constants $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \sigma$ and all $t > t_0$.

Proof outline:

We first show that if $\tilde{q}$, $\dot{\tilde{q}}$ and $\tilde{\theta}$ are in $L_2 \cap L_\infty$ and $\hat{D}(\hat{q})$ is positive definite then $e \in L_2 \cap L_\infty$. The extended Kalman filter is shown to be uniformly asymptotically stable, hence that $\tilde{q}$, $\dot{\tilde{q}}$ and $\tilde{\theta}$ are in $L_2 \cap L_\infty$ by the use of Theorem 2.1. Therefore one must show that the linearized system is uniformly completely observable and uniformly completely controllable. Thus, the filter would be uniformly asymptotically stable in the large if the linearization of the system were exact since the linearized system is uniformly completely observable and controllable. Then, it is shown that the filter is locally uniformly asymptotically stable when the series expansion truncation error is taken into account.

Proof:

1. We now show that if $\tilde{q}$, $\dot{\tilde{q}}$ and $\tilde{\theta}$ are in $L_2 \cap L_\infty$ and $\hat{D}(\hat{q})$ is positive definite then $e \in L_2 \cap L_\infty$.

The system is given by

$$\tau = D(q)\ddot{q} + h(q, \dot{q}) + g(q) + \nu(\dot{q}) = Y(q, \dot{q}, \ddot{q})\theta$$

and the controller by

$$\tau = \hat{D}(\hat{q})[\ddot{q}_d + K_v(\dot{q}_d - \dot{\hat{q}}) + K_p(q_d - \hat{q})] + \hat{h}(\hat{q}, \dot{\hat{q}}) + \hat{g}(\hat{q}) + \hat{\nu}(\dot{\hat{q}})$$

$$\tau = Y(\hat{q}, \dot{\hat{q}}, \ddot{\hat{q}})\hat{\theta} + \hat{D}(\hat{q})[\ddot{q}_d - \ddot{\hat{q}} + K_v(\dot{q}_d - \dot{\hat{q}}) + K_p(q_d - \hat{q})]$$

if we define the errors by: $\hat{q} = q + \tilde{q}$, $\dot{\hat{q}} = \dot{q} + \dot{\tilde{q}}$, $\hat{\theta} = \theta + \tilde{\theta}$ and $e = q - q_d$, then

$$\hat{D}(\hat{q})[\ddot{e} + K_v(\dot{e} + \dot{\tilde{q}}) + K_p(e + \tilde{q})] = Y(q, \dot{q}, \ddot{q})\hat{\theta} - Y(\hat{q}, \dot{\hat{q}}, \ddot{\hat{q}})\hat{\theta}$$

which gives us

$$\ddot{e} + K_v\dot{e} + K_p e = d(t)$$

with

$$d(t) = -K_v\dot{\tilde{q}} - K_p\tilde{q} + \hat{D}^{-1}(\hat{q}) \{ Y(q, \dot{q}, \ddot{q})\tilde{\theta} + [Y(\hat{q}, \dot{\hat{q}}, \ddot{\hat{q}}) - Y(q, \dot{q}, \ddot{q})]\hat{\theta} \}$$

Now, if $\tilde{q}$, $\tilde{\dot{q}}$ and $\tilde{\theta}$ are in $L_2 \cap L_\infty$ and $\hat{D}(\hat{q})$ is positive definite, we have $K_v \tilde{\dot{q}} \in L_2 \cap L_\infty$, $K_p \tilde{q} \in L_2 \cap L_\infty$, $Y(q, \dot{q}, \tilde{q}) \in L_\infty$, $Y(q, \dot{q}, \tilde{q})\tilde{\theta} \in L_2 \cap L_\infty$ and $Y(q, \dot{q}, \tilde{q})$ is continuous in q and $\dot{q}$, has continuous and bounded partial derivatives with respect to q and $\dot{q}$, and satisfies a local Lipschitz condition:

$$\|Y(\hat{q}, \hat{\dot{q}}, \tilde{q}) - Y(q, \dot{q}, \tilde{q})\| \leq k \left\| \begin{bmatrix} \hat{q} - q \\ \hat{\dot{q}} - \dot{q} \end{bmatrix} \right\|$$

for some $0 < k < \infty$. Hence, $\tilde{q}$, $\tilde{\dot{q}}$ and $\tilde{\theta} \in L_2 \cap L_\infty$ imply that $Y(\hat{q}, \hat{\dot{q}}, \tilde{q}) - Y(q, \dot{q}, \tilde{q}) \in L_2 \cap L_\infty$, $\hat{\theta} \in L_\infty$ and $[Y(\hat{q}, \hat{\dot{q}}, \tilde{q}) - Y(q, \dot{q}, \tilde{q})]\hat{\theta} \in L_2 \cap L_\infty$. Therefore, $d(t) \in L_2 \cap L_\infty$ which imply that $e \in L_2 \cap L_\infty$ and $e(t) \rightarrow 0$ as $t \rightarrow \infty$ since $(Is^2 + K_v s + K_p)^{-1}$ is a strictly proper exponentially stable transfer function (see Theorem D.2).

2. One must now show that the extended Kalman filter is locally uniformly asymptotically stable, hence that $\tilde{q}$, $\tilde{\dot{q}}$ and $\tilde{\theta}$ are in $L_2 \cap L_\infty$. To do this, we invoke Theorem 2.1 and we therefore have to show that the system is observable and controllable. Item (a) shows the condition under which the system is observable, item (b) shows that the system is controllable. Item (c) states the results of Theorem 2.1 and item (d) uses Theorem 4.4 to show local stability even if the linearization is not exact.

- (a) The linearized system is uniformly completely observable if conditions 3,4,6,7 of the Theorem are satisfied. To show observability, only the homogeneous part of the linearized system have to be considered. Let us consider the system and the measurements:

$$\dot{x}(t) = F(t)x(t)$$

$$y(t) = H(t)x(t)$$

with the transition matrix:

$$\frac{d\Phi(t, t_o)}{dt} = F(t)\Phi(t, t_o)$$

$$\Phi(t_o, t_o) = I$$

and

$$F(t) = \frac{\partial f(\hat{x})}{\partial \hat{x}} = \begin{bmatrix} 0 & I & 0 \\ F_{21}(t) & F_{22}(t) & F_{23}(t) \\ 0 & 0 & 0 \end{bmatrix}$$

$$H(t) = \begin{bmatrix} I & 0 & 0 \end{bmatrix}$$

where I , $F_{21}(t)$ and $F_{22}(t)$ are $m \times m$ matrices, $F_{23}(t)$ is $m \times p$ matrix, and $H(t)$ is a $m \times (2m + p)$ matrix. Also, the matrices $F_{21}(t)$, $F_{22}(t)$ and $F_{23}(t)$ are bounded and locally integrable since $f(\hat{x})$ has continuous and bounded partial derivatives when $\hat{D}(\hat{q})$ is positive definite. The vector $x(t)$ can be divided in three subvectors such as:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

where x_1 and x_2 are m component vectors and x_3 is p component vector. Since $\dot{x}_3 = 0$, x_3 is constant. Thus, one can define a reduced state vector x_r :

$$x_r = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

The dynamic system can be written as

$$\dot{x}_r = \begin{bmatrix} 0 & I \\ F_{21}(t) & F_{22}(t) \end{bmatrix} x_r + \begin{bmatrix} 0 \\ F_{23} \end{bmatrix} x_3$$

and the measurements as

$$y = H_r x_r$$

$$H = \begin{bmatrix} \underbrace{I \ 0}_{H_r} & 0 \end{bmatrix} = \begin{bmatrix} H_r & 0 \end{bmatrix}$$

Let us define

$$F_r(t) = \begin{bmatrix} 0 & I \\ F_{21}(t) & F_{22}(t) \end{bmatrix}$$

Using Theorem 4.2, it is simple to show that the system $(F_r(t), H_r(t))$ is totally observable.

Theorem 4.2 *The system $(F(t), H(t))$ is totally observable if and only if for every t the matrix:*

$$\Gamma = \begin{bmatrix} \Gamma_1 & \Gamma_2 & \cdots & \Gamma_n \end{bmatrix}$$

is of rank n ($x(t)$ is a vector of n components) where:

$$\Gamma_1 = H^T(t) \quad \text{and} \quad \Gamma_k = \frac{d\Gamma_{k-1}}{dt} + F^T(t)\Gamma_{k-1}$$

Proof (see [15]).

Hence the following integral is positive definite for all $t_f > t_0$.

$$M_r(t_0, t_f) = \int_{t_0}^{t_f} \Phi_r^T(t, t_0) H_r^T(t) H_r(t) \Phi_r(t, t_0) dt$$

One can write $\Phi_r(t, t_0)$ in partitioned form as:

$$\Phi_r(t, t_0) = \begin{bmatrix} \Phi_{r11}(t, t_0) & \Phi_{r12}(t, t_0) \\ \Phi_{r21}(t, t_0) & \Phi_{r22}(t, t_0) \end{bmatrix} \quad (4.2)$$

this results in

$$M_r(t_0, t_f) = \int_{t_0}^{t_f} \begin{bmatrix} \Phi_{r11}^T(t, t_0) \Phi_{r11}(t, t_0) & \Phi_{r11}^T(t, t_0) \Phi_{r12}(t, t_0) \\ \Phi_{r12}^T(t, t_0) \Phi_{r11}(t, t_0) & \Phi_{r12}^T(t, t_0) \Phi_{r12}(t, t_0) \end{bmatrix} dt$$

From the Sylvester criterion for positive definiteness, and since $M_r(t_0, t_f)$ is positive definite for all $t_f > t_0$:

$$\int_{t_0}^{t_f} \Phi_{r12}^T(t, t_0) \Phi_{r12}(t, t_0) dt > 0 \quad (4.3)$$

Since x_3 is constant, we can write:

$$x_r(t) = \Phi_r(t, t_0)x_r(t_0) + \left[\int_{t_0}^t \Phi_r(t, \tau) \begin{bmatrix} 0 \\ F_{23}(\tau) \end{bmatrix} d\tau \right] x_3$$

where $\Phi_r(t, t_0)$ is the transition matrix of the reduced system which is totally observable. Since $\Phi_r(t, \tau) = \Phi_r(t, t_0)\Phi_r^{-1}(\tau, t_0)$, we have:

$$x_r(t) = \Phi_r(t, t_0)x_r(t_0) + \Phi_r(t, t_0) \left[\int_{t_0}^t \Phi_r^{-1}(\tau, t_0) \begin{bmatrix} 0 \\ F_{23}(\tau) \end{bmatrix} d\tau \right] x_3$$

We can write the previous equation in terms of the full state vector $x(t)$:

$$x(t) = \underbrace{\begin{bmatrix} \Phi_r(t, t_0) & \Phi_r(t, t_0) \int_{t_0}^t \Phi_r^{-1}(\tau, t_0) \begin{bmatrix} 0 \\ F_{23}(\tau) \end{bmatrix} d\tau \\ 0 & I \end{bmatrix}}_{\Phi(t, t_0)} x(t_0)$$

We can now write:

$$H(t)\Phi(t, t_0) = H_r\Phi_r(t, t_0) \left[I \int_{t_0}^t \Phi_r^{-1}(\tau, t_0) \begin{bmatrix} 0 \\ F_{23}(\tau) \end{bmatrix} d\tau \right]$$

Let us define

$$\begin{aligned} \mathcal{P}(t) &= H_r\Phi_r(t, t_0) \\ \Psi(\tau) &= \Phi_r^{-1}(\tau, t_0) \begin{bmatrix} 0 \\ F_{23}(\tau) \end{bmatrix} \\ \mathcal{Q}(t) &= \int_{t_0}^t \Psi(\tau) d\tau \end{aligned}$$

thus

$$M(t_0, t_f) = \int_{t_0}^{t_f} \begin{bmatrix} I & \mathcal{Q}(t) \end{bmatrix}^T \mathcal{P}^T(t) \mathcal{P}(t) \begin{bmatrix} I & \mathcal{Q}(t) \end{bmatrix} dt$$

and $M(t_0, t_f)$ is positive definite if $x^T M(t_0, t_f) x > 0$ for all unit vectors x . Let

$$x = \begin{bmatrix} x_r \\ x_3 \end{bmatrix}$$

this yields

$$\int_{t_0}^{t_f} \|\mathcal{P}(t)(x_r + Q(t)x_3)\|^2 dt > 0$$

First, we have to show under which conditions there is no unit vector x such that $\mathcal{P}(t)(x_r + Q(t)x_3) = 0$ is true for all $t_0 \leq t \leq t_f$ (this result is similar to the proof of Theorem 2.3 in [3]). Let us define

$$S(t_0, t_f) = \int_{t_0}^{t_f} \|\mathcal{P}(t)(x_r + Q(t)x_3)\|^2 dt$$

By definition for all $t_0 < t < t_f$, we have

$$\begin{aligned} & \frac{1}{2} \left| \|\mathcal{P}(t)(x_r + Q(t)x_3)\|^2 - x_r^T \mathcal{P}^T(t) \mathcal{P}(t) x_r \right| \\ &= \left| \int_{t_0}^t x_3^T (\dot{\mathcal{P}}(\tau) Q(\tau) + \mathcal{P}(\tau) \dot{Q}(\tau))^T \mathcal{P}(t)(x_r + Q(\tau)x_3) d\tau \right| \quad (4.4) \end{aligned}$$

Let us define

$$\Omega(t) = \dot{\mathcal{P}}(t) Q(t) + \mathcal{P}(t) \dot{Q}(t)$$

One can easily show that

$$\Omega(t) = \int_{t_0}^t \Phi_{r22}(t, \tau) F_{23}(\tau) d\tau$$

where the transition matrix $\Phi_r(t, t_0)$ has been partitioned as in equation (4.2). The Schwarz's inequality applied to the right hand side of the equation (4.4) results in

$$\frac{1}{2} \left| \|\mathcal{P}(t)(x_r + Q(t)x_3)\|^2 - x_r^T \mathcal{P}^T(t) \mathcal{P}(t) x_r \right|$$

$$\begin{aligned}
&\leq \left[\int_{t_0}^t \|\Omega(\tau)x_3\|^2 d\tau \right]^{\frac{1}{2}} \left[\int_{t_0}^t \|\mathcal{P}(t)(x_r + Q(\tau)x_3)\|^2 d\tau \right]^{\frac{1}{2}} \\
&\leq (t - t_0)^{\frac{1}{2}} \sup_{\tau \in [t_0, t]} \|\Omega(\tau)\| S^{\frac{1}{2}}(t_0, t) \\
&\leq (t_f - t_0)^{\frac{1}{2}} \sup_{\tau \in [t_0, t_f]} \|\Omega(\tau)\| S^{\frac{1}{2}}(t_0, t_f)
\end{aligned}$$

Since $b - a \leq |a - b|$, integrating for t from t_0 to t_f yields:

$$\int_{t_0}^{t_f} x_r^T \mathcal{P}^T(t) \mathcal{P}(t) x_r dt \leq 2(t_f - t_0)^{\frac{3}{2}} \sup_{\tau \in [t_0, t_f]} \|\Omega(\tau)\| S^{\frac{1}{2}}(t_0, t_f) + S(t_0, t_f)$$

Applying the triangle inequality to $S(t_0, t_f)$ gives us

$$\int_{t_0}^{t_f} \|\mathcal{P}(t)Q(t)x_3\|^2 dt \leq \int_{t_0}^{t_f} x_r^T \mathcal{P}^T(t) \mathcal{P}(t) x_r dt + S(t_0, t_f)$$

Combining the last two equations yields

$$\begin{aligned}
\int_{t_0}^{t_f} \|\mathcal{P}(t)Q(t)x_3\|^2 dt &\leq 2(t_f - t_0)^{\frac{3}{2}} \sup_{\tau \in [t_0, t_f]} \|\Omega(\tau)\| S^{\frac{1}{2}}(t_0, t_f) + 2S(t_0, t_f) \\
&\leq 2\{kS^{\frac{1}{2}}(t_0, t_f) + S(t_0, t_f)\}
\end{aligned}$$

where k is a non-negative constant. If the matrix

$$\int_{t_0}^{t_f} Q^T(t) \mathcal{P}^T(t) \mathcal{P}(t) Q(t) dt$$

is positive definite, the last equation implies that $S(t_0, t_f) > 0$ for a non-null x_3 and if $x_3 = 0$ we have

$$S(t_0, t_f) = x_r^T \left(\int_{t_0}^{t_f} \mathcal{P}^T(t) \mathcal{P}(t) dt \right) x_r = x_r^T M_r(t_0, t_f) x_r > 0$$

by definition (the reduced system is totally observable). Hence, we can conclude that $\int_{t_0}^{t_f} Q^T(t) \mathcal{P}^T(t) \mathcal{P}(t) Q(t) dt$ being positive definite is a necessary and sufficient condition for the observability of the system in the interval $[t_0, t_f]$. From the definition of $\mathcal{P}(t)$ and $Q(t)$, one has

$$\begin{aligned}
&\int_{t_0}^{t_f} Q^T(t) \mathcal{P}^T(t) \mathcal{P}(t) Q(t) dt \\
&= \int_{t_0}^{t_f} \left(\int_{t_0}^t \Phi_{r12}(t, \tau) F_{23}(\tau) d\tau \right)^T \left(\int_{t_0}^t \Phi_{r12}(t, \tau) F_{23}(\tau) d\tau \right) dt \quad (4.5)
\end{aligned}$$

where the transition matrix $\Phi_r(t, t_0)$ has been partitioned as in equation (4.2). If there is no unit vector x such that

$$\left\| \int_{t_0}^t \Phi_{r_{12}}(t, \tau) F_{23}(\tau) x d\tau \right\| = 0 \quad (4.6)$$

for all $t_0 \leq t \leq t_f$, then the integral in equation (4.5) is positive definite. If $\dot{F}_{23}(\tau)$ is bounded, then the conditions for equation (4.6) can be replaced by (see [3]):

$$\int_{t_0}^{t_f} \|\Phi_{r_{12}}(t, \tau) F_{23}(\tau) x\| d\tau > 0 \quad (4.7)$$

The substitution of the control law in the filter results in

$$\begin{bmatrix} \dot{\hat{q}} \\ \dot{\hat{q}} \\ \dot{\hat{\theta}} \end{bmatrix} = \begin{bmatrix} \hat{q} \\ \bar{q}_d + K_v(\dot{q}_d - \dot{\hat{q}}) + K_p(q_d - \hat{q}) \\ 0 \end{bmatrix} + K(t)[q - \hat{q}]$$

hence the filter is trying to track the desired path while it is perturbed by $K(t)[q - \hat{q}]$ where $K(t)$ is the Kalman gain matrix. Since $K(t)$ and $[q - \hat{q}]$ are bounded and the robot model has continuous partial derivatives when $\hat{D}(\hat{q})$ is positive definite, the last equation guarantees that the matrices $F_{21}(t)$, $F_{22}(t)$ and $F_{23}(t)$ are bounded and locally integrable and have bounded derivatives if conditions 1,2,3 are satisfied. Since $\dot{F}_{23}(\tau)$ is bounded and equation (4.3) imply that

$$\int_{t_0}^t \Phi_{r_{12}}^T(t, \tau) \Phi_{r_{12}}(t, \tau) d\tau$$

spans the whole space for an infinitely small $t - t_0 > 0$, we can conclude that:

$$\int_{t_0}^{t_f} F_{23}^T(\tau) F_{23}(\tau) d\tau > 0$$

is a sufficient and necessary condition for the observability of the system in the interval $[t_0, t_f]$.

- (b) The linearized system is uniformly completely controllable if conditions 3 and 5 are satisfied. It is immediate to show that the linearized system is totally controllable by the use of Theorem 4.2 since

$$G(t) = \begin{bmatrix} 0 & 0 \\ D^{-1}(q, \theta) & 0 \\ 0 & I \end{bmatrix}_{\text{along } \hat{x}}$$

Theorem 4.3 *The system $(F(t), G(t))$ is totally controllable if and only if for every t the matrix:*

$$\Gamma = \begin{bmatrix} \Gamma_1 & \Gamma_2 & \cdots & \Gamma_n \end{bmatrix}$$

is of rank n ($x(t)$ is a vector of n components) where:

$$\Gamma_1 = G(t) \quad \text{and} \quad \Gamma_k = \frac{d\Gamma_{k-1}}{dt} - F(t)\Gamma_{k-1}$$

Proof (see [15]).

- (c) The extended Kalman filter would be uniformly asymptotically stable in the large if the linearization of the system were exact. In more details, we have the real system given by

$$\begin{aligned} \dot{\hat{x}} = \dot{\hat{x}} + \dot{\tilde{x}} &= f(\hat{x} + \tilde{x}) = f(\hat{x}) + \frac{\partial f(\hat{x})}{\partial \hat{x}} \tilde{x} \\ &+ \frac{1}{2} \tilde{x}^T \left[\frac{\partial}{\partial \hat{x}} \left(\frac{\partial f(\hat{x})}{\partial \hat{x}} \right) \right] \tilde{x} + \text{higher-order terms} \end{aligned}$$

The unperturbed extended Kalman filter and the filtering error are given by

$$\begin{aligned} \dot{\hat{x}} &= f(\hat{x}) + P(t)H^T R^{-1}(t)H \tilde{x} \\ \dot{\tilde{x}} &= \left(\frac{\partial f(\hat{x})}{\partial \hat{x}} - P(t)H^T R^{-1}(t)H \right) \tilde{x} \\ &+ \frac{1}{2} \tilde{x}^T \left[\frac{\partial}{\partial \hat{x}} \left(\frac{\partial f(\hat{x})}{\partial \hat{x}} \right) \right] \tilde{x} + \text{higher-order terms} \end{aligned}$$

hence

$$\dot{\tilde{x}} = A(t)\tilde{x} + g(\tilde{x}, t)$$

where

$$\begin{aligned} A(t) &= \left(\frac{\partial f(\hat{x})}{\partial \hat{x}} - P(t)H^T R^{-1}(t)H \right) \\ g(\tilde{x}, t) &= \frac{1}{2}\tilde{x}^T \left[\frac{\partial}{\partial \hat{x}} \left(\frac{\partial f(\hat{x})}{\partial \hat{x}} \right) \right] \tilde{x} + \text{higher-order terms} \end{aligned}$$

Since the pair $(\frac{\partial f(\hat{x})}{\partial \hat{x}}, H)$ is uniformly complete observable (step 2a) and the pair $(\frac{\partial f(\hat{x})}{\partial \hat{x}}, G(\hat{x}))$ is uniformly completely controllable (step 2b), from Theorem 2.1 and Theorem 4.1 statement, $A(t)$ is a uniformly asymptotically stable matrix.

- (d) The extended Kalman filter system is locally uniformly asymptotically stable when the series expansion truncation error is taken into account. The filtering error is given by

$$\dot{\tilde{x}} = A(t)\tilde{x} + g(\tilde{x}, t)$$

Since $f(\hat{x})$ is analytic when $\hat{D}(\hat{q})$ is positive definite, $g(\tilde{x}, t) = o(\|\tilde{x}\|)$. Also, $A(t)$ is a uniformly asymptotically stable matrix, thus, by Theorem 4.4, the extended Kalman filter is locally uniformly asymptotically stable. Hence, $\tilde{x} \in L_2 \cap L_\infty$, therefore $\tilde{q}, \dot{\tilde{q}}, \tilde{\theta} \in L_2 \cap L_\infty$.

Theorem 4.4 (Perron) *Let*

$$\dot{x} = A(t)x + g(x, t)$$

where $A(t)$ is a real matrix such that the system $\dot{y} = A(t)y$ is uniformly asymptotically stable and $\|g(x, t)\| = o(\|x\|)$ (uniformly in t) then $x = 0$ is (locally) uniformly asymptotically stable.

Proof (see [33, 46]).

The proof is now completed.

One notices that the observability criterion (condition 7 of Theorem 4.1)

$$\alpha_5 I \leq \int_t^{t+\sigma} F_{23}^T(\tau) F_{23}(\tau) d\tau \leq \alpha_6 I$$

with

$$F_{23}(\tau) = \left[-D^{-1} \left\{ \frac{\partial D}{\partial \theta} \xi + \frac{\partial \nu}{\partial \theta} + \frac{\partial h}{\partial \theta} + \frac{\partial g}{\partial \theta} \right\} \right]_{\text{along } \hat{x}}$$

is the persistency excitation criterion of Craig's method except that it is evaluated along $\hat{q}$ and $\dot{\hat{q}}$ instead of q and $\dot{q}$. Since the local stability imply that the filter output error $q - \hat{q}$ goes to zero, this observability criterion can be evaluated along q_d in a reduced form like it is shown in Craig's results.

These results show the coherence between the two approaches. Hence, as with Craig's results, if the path is persistently exciting, the filter is uniformly stable (hence the controller is stable) provided that $\hat{D}(\hat{q})$ is positive definite.

4.3 Adding Robustness to the Scheme

The system robustness to disturbances depends on the observability of the linearized system, hence indirectly on the desired path trajectory. As in Craig's result, if this trajectory is not persistently exciting, uniform asymptotic stability is lost. This is not a unique feature of this adaptive controller as it is shown that under these conditions other adaptive controllers [19, 81] can produce unbounded errors to bounded disturbances as shown in [68] where robustifying modifications are suggested (bounded output under bounded input disturbances is guaranteed only under uniform asymptotic stability).

Two modifications will be introduced in order to improve the robustness and stability of the system. These modifications use a priori bounds on the unknown parameters:

$$\theta_{\min} < \theta < \theta_{\max}$$

When θ satisfies these bounds, $\hat{D}(q)$ is guaranteed to be positive definite. This not very restrictive since from physical considerations these bounds can be easily selected for most parameters (e.g. mass and inertia are positive and their maximum possible values can easily guessed).

A first modification will be introduced in order to assure stability of the system. This modification deals with the problem that if the linearized system is not observable, only parameters can become unobservable since q and $\dot{q}$ are totally observable under all circumstances. Hence, only the portion of the covariance matrix corresponding to the parameter P_θ can grow without bounds. One must remember that $P(t)$ is a positive definite matrix. Let us partition $P(t)$ as

$$P(t) = \begin{bmatrix} P_q & P_{q\dot{q}} & P_{q\theta} \\ P_{\dot{q}q} & P_{\dot{q}} & P_{\dot{q}\theta} \\ P_{\theta q} & P_{\theta\dot{q}} & P_\theta \end{bmatrix}$$

Since the stability of the filter is based on the Lyapunov function candidate:

$$V = \tilde{x}^T P^{-1}(t) \tilde{x}$$

and only the portion of the covariance matrix corresponding to the parameter P_θ can grow without bounds, $P^{-1}(t)$ will become positive semi-definite if $\|P_\theta\| \rightarrow \infty$, thus our Lyapunov function candidate will not be valid. Therefore, we should prevent P_θ from increasing without bounds. This is similar to the problem which is discussed by Li and Slotine in [56, 57]. If in addition, we have the a priori information that:

$$\theta_{\min} < \theta < \theta_{\max}$$

hence it seems reasonable to try to assure that the diagonal of P_θ be upper bounded by

$$P_{\theta ii} \leq (\theta_{i\max} - \theta_{i\min})^2$$

since P_θ represent the covariance of $\hat{\theta} - \theta$ if $R(t)$ and $Q(t)$ are chosen adequately (see next section). From the covariance equation, it is simple to show that

$$Q_{2ii} = \begin{cases} 0 & \text{if } P_{\theta ii} \geq (\theta_{i\max} - \theta_{i\min})^2 \\ \beta_i > 0 & \text{if } P_{\theta ii} < (\theta_{i\max} - \theta_{i\min})^2 \end{cases}$$

will assure that P_θ is bounded. Hence, the Lyapunov function candidate will remain valid.

The second modification can be considered as a σ -modification that use the same a priori bounds on the unknown parameters. The state equation for the modified model are

$$\dot{x} = f(x, u) + \begin{bmatrix} 0 \\ 0 \\ \sigma(\theta) \end{bmatrix}$$

where $\sigma(\theta)$ is a vector function given by

$$\sigma_i(\theta) = \begin{cases} -\alpha(\theta_i - \theta_{i\min}) & \text{if } \theta_i < \theta_{i\min} \\ 0 & \text{if } \theta_{i\min} < \theta_i < \theta_{i\max} \\ -\alpha(\theta_i - \theta_{i\max}) & \text{if } \theta_i > \theta_{i\max} \end{cases}$$

where $\theta_{i\min}$ and $\theta_{i\max}$ are the known bounds on the i th parameters and $\alpha > 0$ (see fig. 4.2). Let us consider the equation:

$$\dot{\hat{\theta}} = \sigma(\hat{\theta})$$

This last equation is uniformly stable in the large (but not asymptotically). This is shown by using the Lyapunov function:

$$V = (\hat{\theta} - \theta)^T (\hat{\theta} - \theta)$$

its time derivative is

$$\begin{aligned} \dot{V} &= (\hat{\theta} - \theta)^T \dot{\hat{\theta}} \\ &= \sum \dot{V}_i \end{aligned}$$

where

$$\dot{V}_i = \begin{cases} -\alpha \overbrace{(\hat{\theta}_i - \theta_i)^T}^{(-)} \overbrace{(\hat{\theta}_i - \theta_{i\min})}^{(-)} & \text{if } \hat{\theta}_i < \theta_{i\min} \\ 0 & \text{if } \theta_{i\min} < \hat{\theta}_i < \theta_{i\max} \\ -\alpha \overbrace{(\hat{\theta}_i - \theta_i)^T}^{(+)} \overbrace{(\hat{\theta}_i - \theta_{i\max})}^{(+)} & \text{if } \hat{\theta}_i > \theta_{i\max} \end{cases}$$

so that

$$\begin{aligned} \dot{V} &\leq 0 \\ \dot{V} &\rightarrow -\infty \text{ as } \|\hat{\theta} - \theta\| \rightarrow \infty \end{aligned}$$

Therefore, the problem that the parameter vector θ can become unobservable will be solved by making the possible unobservable vector $\hat{\theta} - \theta$ uniformly stable. This results in a linearized system with totally observable $\tilde{q}$ and $\hat{q}$ and possible unobservable but stable $\tilde{\theta}$. This modification when it is included in the filter preserves its stability (results from Fitzgerald [25] showed that a Kalman filter will remain stable if its unobservable states are stable) by using

$$\begin{aligned} f'(x, u) &= f(x, u) + \begin{bmatrix} 0 \\ 0 \\ \sigma(\theta) \end{bmatrix} \\ \frac{\partial f'}{\partial x} &= \frac{\partial f}{\partial x} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{\partial \sigma(\theta)}{\partial \theta} \end{bmatrix} \end{aligned}$$

where $\frac{\partial \sigma(\theta)}{\partial \theta}$ is a negative semi-definite diagonal matrix with:

$$\frac{\partial \sigma_i(\theta)}{\partial \theta_i} = \begin{cases} -\alpha & \text{if } \theta_i < \theta_{i\min} \\ 0 & \text{if } \theta_{i\min} < \theta_i < \theta_{i\max} \\ -\alpha & \text{if } \theta_i > \theta_{i\max} \end{cases}$$

3.10 Indirect with Filter: Continuous-Time Controller: Parameter Errors:	
$\hat{p}_1 - p_1 =$ solid line, $\hat{p}_2 - p_2 =$ dashed line, $\hat{p}_3 - p_3 =$ dotted line, $\hat{p}_4 - p_4 =$ dash-dot line	50
3.11 Indirect with Filter: Continuous-Time Controller: Tracking Errors:	
$q_1 - q_{1d} =$ solid line, $q_2 - q_{2d} =$ dashed line	50
3.12 Craig: Discrete-Time Controller: Parameter Errors: $\hat{p}_1 - p_1 =$ solid line, $\hat{p}_2 - p_2 =$ dashed line, $\hat{p}_3 - p_3 =$ dotted line, $\hat{p}_4 - p_4 =$ dash-dot line	51
3.13 Craig: Discrete-Time Controller: Tracking Errors: $q_1 - q_{1d} =$ solid line, $q_2 - q_{2d} =$ dashed line	51
3.14 Indirect: Discrete-Time Controller: Parameter Errors: $\hat{p}_1 - p_1 =$ solid line, $\hat{p}_2 - p_2 =$ dashed line, $\hat{p}_3 - p_3 =$ dotted line, $\hat{p}_4 - p_4 =$ dash-dot line	52
3.15 Indirect: Discrete-Time Controller: Tracking Errors: $q_1 - q_{1d} =$ solid line, $q_2 - q_{2d} =$ dashed line	52
3.16 Slotine: Discrete-Time Controller: Parameter Errors: $\hat{p}_1 - p_1 =$ solid line, $\hat{p}_2 - p_2 =$ dashed line, $\hat{p}_3 - p_3 =$ dotted line, $\hat{p}_4 - p_4 =$ dash-dot line	53
3.17 Slotine: Discrete-Time Controller: Tracking Errors: $q_1 - q_{1d} =$ solid line, $q_2 - q_{2d} =$ dashed line	53
3.18 Indirect with Filter: Discrete-Time Controller: Parameter Errors: $\hat{p}_1 - p_1 =$ solid line, $\hat{p}_2 - p_2 =$ dashed line, $\hat{p}_3 - p_3 =$ dotted line, $\hat{p}_4 - p_4 =$ dash-dot line	54
3.19 Indirect with Filter: Discrete-Time Controller: Tracking Errors: $q_1 - q_{1d} =$ solid line, $q_2 - q_{2d} =$ dashed line	54
3.20 Craig with Noise: Discrete-Time Controller: Parameter Errors: $\hat{p}_1 - p_1 =$ solid line, $\hat{p}_2 - p_2 =$ dashed line, $\hat{p}_3 - p_3 =$ dotted line, $\hat{p}_4 - p_4 =$ dash-dot line	55

2

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS
STANDARD REFERENCE MATERIAL 1010a
(ANSI and ISO TEST CHART No. 2)

Hence, this modification will come into effect when at least one parameter is outside the a priori bounds, thus no parameter error can increase without bounds since they are stabilized as soon as they cross the bounds and the corresponding entry in the covariance matrix is guaranteed to remain bounded as long as the parameter is outside the bound. The higher α is chosen, the lower is the depth of the penetration of $\hat{\theta}$ across the boundaries defined by

$$\theta_{\min} < \theta < \theta_{\max}$$

This modification uses the same a priori information as the resetting rule of Craig's method and is similar to the σ -modification used in [68]. Before the modification, the parameters are modeled as

$$\dot{\tilde{\theta}} = w_2(t)$$

hence as a random walk process. If the parameters becomes unobservable, the variance of $\tilde{\theta}$ will increase without bounds. By itself, this process is an infinitely growing variance process ($P_{\tilde{\theta}} = (t - t_0)Q_2$). This is analogous to the unbounded increase of the covariance matrix when RLS are used with a forgetting factor when the system is not persistently excited (see Li and Slotine [56, 57]). After this modification, the parameter model is

$$\dot{\tilde{\theta}} = \sigma(\tilde{\theta}) + w_2(t)$$

which has a finite variance for all t . A Monte Carlo simulation of both models is shown in figure 4.3 and clearly demonstrate this point. Together, the two modifications suggested guarantee that both $\|\tilde{\theta}\|$ and $\|P_{\tilde{\theta}}\|$ are bounded by above at all time even if the parameters are unobservable for a long period of time (bounded-input-bounded-output (BIBO) stable).

4.4 Implementation

The implementation of this adaptive controller requires the choice of some design parameters. However, these choices can be made using guide-lines which are well justified theoretically. The fact that Kalman filtering is developed for stochastic processes will provide us with these guide-lines. Since this scheme will be implemented using a digital computer, a discrete-time implementation is suggested.

4.4.1 Selection of the Filter Initial Conditions

From filtering theory, the initial filtered state estimates are the expected values of these states at the beginning of control. Hence, for the robotic manipulator starting at rest and at a known position (up to the accuracy of the resolver), the initial filtered states are:

$$\hat{x}_0 = \begin{bmatrix} z(t_0) \\ 0 \\ \theta_0 \end{bmatrix}$$

where $z(t_0)$ is the first measurement of position and θ_0 the best estimate available of the unknown parameters.

In the same way, the initial covariance matrix is given by

$$P_0 = \begin{bmatrix} P_q & 0 & 0 \\ 0 & P_{\dot{q}} & 0 \\ 0 & 0 & P_\theta \end{bmatrix}$$

where P_q is a diagonal matrix of the estimated variances of the respective measurements (these values are lower bounded by the accuracy of the analog-to-digital A/D converters used to measure position), $P_{\dot{q}}$ is a diagonal matrix of the variances representing the confidence that we have that the robot is indeed starting to operate from rest, and finally P_θ is the estimated values of the covariance of the possible errors in

the estimated parameters. The higher P_θ is, the less we are sure of the accuracy of our initial estimates.

4.4.2 Selection of the Covariance Matrices

The matrices $R(t)$ and $Q(t)$ are other parameters that must be defined to implement the extended Kalman filter. These matrices should be representative of the noise content of the measurements and control inputs.

- $R(t)$ is simply a diagonal matrix for the measurement variances. This matrix is the same as P_q mentioned above.
- $Q(t)$ is composed of two diagonal matrices $Q_1(t)$ and $Q_2(t)$.

$$Q(t) = \begin{bmatrix} Q_1(t) & 0 \\ 0 & Q_2(t) \end{bmatrix}$$

- $Q_1(t)$ reflects the magnitude of the disturbances caused by unmodeled dynamics, hence the confidence in our model. It also includes the effect of the truncation error in the series expansion, hence $Q_1(t) > 0$.
- $Q_2(t)$ represents the speed at which the parameter vector is estimated to vary in a random walk fashion (this is the Bayesian approach to time varying parameters, see Ljung and Soderstrom [59]). One can use a low value of $Q_2(t)$ when the system is almost perfectly tuned and increase it momentarily when an object is grasped. This allows for a very versatile and simple way of providing time-varying alertness to parameter changes.

From these comments, one realizes that $Q(t)$ and $R(t)$ can be picked in accordance to the physical set-up of the system, hence producing a high performance (sub-optimal) filter if the measurement noise and torque disturbances are white and gaussian.

In the preceding analysis, $R(t)$ can be precisely evaluated, but $Q(t)$ is the component of the filter for which the information is less accurate. However, the Kalman filter provides us with an estimate of what the residual variance should be. (this is based on the innovation approach to Kalman filtering [42]). The residual can be computed, hence their variance can be estimated. If the filter is well tuned, the residual's variance based on the filter covariance matrix should agree with the estimated covariance from measurements. With this in mind, researchers [38, 60] presented methods to adaptively estimate the covariance matrices so as to make the residual's statistics and the filter prediction agree with each other. Obviously, these methods can be used with the discrete-time implementation described below.

4.4.3 Discrete-Time Controller Implementation

The previous results assume that the controller is implemented in continuous-time form. However, in experimental applications the controller will be implemented in discrete form where the CTM and EKF will operate at the same sampling frequency. The sampling interval T should be picked such that a perfectly tuned controller would have the desired performance (hence T is much smaller than the manipulator's smallest *time constant*). In order to discretize this adaptive controller, the model will be discretized and the discrete formulation of the extended Kalman filter will be applied. We have a direct and theoretically justified way of implementing the controller in discrete-time form.

In order to discretize this adaptive controller, two choices are available:

- Direct discretization of the continuous-time filter. This discretization should produce a causal controller (e.g. Euler integration rule). However, the covariance equation is more likely to be numerically unstable for high values of the initial covariance matrix. We do not recommend this approach.
- Discretization of the model and application of the discrete formulation of the

extended Kalman filter to this discrete-time model. This is the chosen and preferable approach. Hence, we have a direct and theoretically justified way of implementing the controller in discrete-time form.

The continuous-time system is given by its unperturbed and perturbation models:

$$\begin{aligned}\dot{x} &= f(x, u) \\ \delta\dot{x} &= F(t)\delta x + G(t)\delta u\end{aligned}$$

Let T be the sampling interval, $u(\tau)$ the control signal and $\delta u(\tau)$ the input disturbances. For the discrete time controller, $u(\tau)$ is constant and $\delta u(\tau)$ is assumed constant for $t < \tau < t + T$ (with $t = kT$ and k an integer). Hence,

$$x(t + T) = x(t) + \int_t^{t+T} f(x(\tau), u(t))d\tau$$

Using the notation $x(t = kT) = x(k)$, we have

$$x(k + 1) = \phi[x(k), k] = x(k) + \int_{kT}^{(k+1)T} f(x(\tau), u(kT))d\tau$$

The discrete-time perturbation model is given by (this follows directly from linear systems theory [65])

$$\delta x(t + T) = \Phi(t + T, t)\delta x(t) + \Gamma(t)\delta u(t)$$

where

$$\dot{\Phi}(t + \Delta, t) = F(t + \Delta)\Phi(t + \Delta, t) \text{ with } 0 \leq \Delta \leq T \text{ and } \Phi(t, t) = I$$

and

$$\Gamma(t) = \Phi(t + T, t) \int_t^{t+T} \Phi(\tau, t)G(\tau)d\tau$$

using the k index notation

$$\delta x(k + 1) = \Phi(k + 1, k)\delta x(k) + \Gamma(k)\delta u(k)$$

The system model is now in a suitable form to apply the discrete-time extended Kalman filter (see section 2.3.2). However, for high sampling rates, the previous discretized model equations are too complex to be implemented in real time. We shall introduce some simplifications in order to diminish the computational burden of this model.

Since T is very small compared to the dynamics of the manipulator and $u(t)$ is constant between samples, one can introduce the following approximations:

- the manipulator joints are under constant acceleration between samples,
- the matrices $F(t)$ and $G(t)$ remain constant between samples.

This results in the approximations:

$$\phi(x(k), k) \approx \begin{bmatrix} q(k) + T\dot{q}(k) + \frac{T^2}{2}\ddot{q}(k) \\ \dot{q}(k) + T\ddot{q}(k) \\ \theta(k) \end{bmatrix}$$

$$\Phi(k+1, k) \approx e^{F(kT)T} = \sum_{i=0}^{\infty} \frac{F^i(kT)T^i}{i!}$$

$$\Gamma(k) \approx \left(\int_{kT}^{(k+1)T} e^{F(kT)(\tau-kT)} d\tau \right) G(kT) = \left(\sum_{i=0}^{\infty} \frac{F^i(kT)T^i}{(i+1)!} \right) G(kT)T$$

With these new models, the CTM method can be used at each sampling time in combination with the discrete EKF formulation. For fast sampling rates, the matrix exponential can be well approximated by only a few terms in the series.

We now describe how we can take advantage of the particular structure of $F(kT)$ so as to increase the efficiency of the discrete-time scheme. First, one notices that $F(kT)$ and $G(kT)$ are in a form equivalent to a system with bias parameters (for m parameters the last m rows of $F(kT)$ are zeros). This can be taken into account when computing $\Phi(k+1, k)$ and $\Gamma(k)$, hence reducing significantly the time needed to obtain the discrete-time model. The bias partitioned $U-D$ factorization formulation of the discrete filter, which is computationally efficient and numerically stable, will

be used for this model (see [9, 83]). If measurements of velocity are available, all the results presented above apply without modifications with the observation matrix replaced by

$$H = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \end{bmatrix}$$

4.5 Example

In this section, we present simulation examples. The first example considers a two DOF robot and the second case presents how in a simple one DOF example the robustness modifications improve the behaviour of the adaptive scheme.

4.5.1 Two DOF Robot Example

In the simulations, the dynamic model of the planar 2 DOF serial robot shown in figure 3.1 will be used. Defining the parameters:

$$\begin{aligned} p_1 &= m_1(c_1^2 + o_1^2) + I_1 + m_2 l_1^2 \\ p_2 &= m_2(c_2^2 + o_2^2) + I_2 \\ p_3 &= m_2 l_1 c_2 \\ p_4 &= m_2 l_1 o_2 \end{aligned}$$

and applying Lagrange's equation to this system yields:

$$\begin{aligned} \tau_1 &= (p_1 + p_2 + 2p_3 \cos q_2 - 2p_4 \sin q_2) \ddot{q}_1 + (p_2 + p_3 \cos q_2 - p_4 \sin q_2) \ddot{q}_2 \\ &\quad - (p_3 \sin q_2 + p_4 \cos q_2) \dot{q}_2 (2\dot{q}_1 + \dot{q}_2) \\ \tau_2 &= (p_2 + p_3 \cos q_2 - p_4 \sin q_2) \ddot{q}_1 + p_2 \ddot{q}_2 + (p_3 \sin q_2 + p_4 \cos q_2) \dot{q}_1^2 \end{aligned}$$

In matrix form:

$$D(q) = \begin{bmatrix} p_1 + p_2 + 2p_3 \cos q_2 - 2p_4 \sin q_2 & p_2 + p_3 \cos q_2 - p_4 \sin q_2 \\ p_2 + p_3 \cos q_2 - p_4 \sin q_2 & p_2 \end{bmatrix}$$

$$\begin{aligned}
h(q, \dot{q}) &= \begin{bmatrix} -(p_3 \sin q_2 + p_4 \cos q_2) \dot{q}_2 (2\dot{q}_1 + \dot{q}_2) \\ (p_3 \sin q_2 + p_4 \cos q_2) \dot{q}_1^2 \end{bmatrix} \\
g(q) &= \nu(\dot{q}) = 0
\end{aligned}$$

The robot is assumed to be composed of two rectangular links of length $0.3m$ and mass $2kg$, hence

$$\theta = \begin{bmatrix} 0.24 & 0.06 & 0.09 & 0 \end{bmatrix} \text{ kgm}^2$$

All simulations will be started with the parameter estimates:

$$\hat{\theta} = \begin{bmatrix} 0.3 & 0.2 & 0.1 & 0 \end{bmatrix} \text{ kgm}^2$$

The desired path is given by

$$q_1 = \arccos(0.45 - 0.45 \sin(\omega t + \pi/2)) = -q_2/2$$

with $\omega = 5$ (period of $T_p = 1.255$ sec). This correspond to the robot and the path described in [75]. For the computed torque method, we want the system error dynamics to have a damping coefficient of $\zeta = 1/\sqrt{2}$ and a natural frequency of $\omega_n = 10$ yielding $K_v = 14.1I$ and $K_p = 100I$. This correspond to settling time of $T_s = 0.4$ sec (5% criterion).

The simulations present results for a 200hz discrete-time implementation without and with measurement noise. In both cases, the discrete covariance matrices were set as $R(k) = 0.0001I$, $Q_1(k) = 0.01I$, $Q_2(k) = 0.0025I$, $P(0) = 0.0001I$. The first simulation presents results for the controller without noise. Figures 4.4– 4.7 respectively show the estimated parameter errors, the tracking and filtering errors. The convergence is fast and small tracking errors are obtained after one cycle of the desired path.

The next simulation presents results for a 200hz discrete-time controller with measurement noise of 0.01 rad (0.6 deg) standard deviation (gaussian). This is more than the quantization noise of 8 bits resolvers. The results are presented in figures

4.8–4.11. The adaptive controller shows good noise rejection properties. After one cycle, the filtering errors are significantly lower than the measurement noise and tracking errors are small. In this first cycle, the tracking error is a combination of the effect of parameter errors and filtering errors on positions and velocities which are used in the feedback control law. In all cases, filtered velocities are very smooth and their errors get very small after one cycle. One should note that with this level of measurement noise, velocities obtained by direct differentiation are very noisy (see fig.4.12). Also in the presence of measurement noise, this scheme performed better than the methods reviewed in chapter 3 (faster convergence, unbiased estimates and lower tracking errors).

Finally, figure 4.13 presents the Lyapunov function for both cases. For the case without noise, the Lyapunov function is decreasing most of the time. However, it cannot decrease indefinitely because of the approximations used in the discrete-time predictor. Furthermore, the non-linearities in the model is the cause of increasing values of the Lyapunov function in the initial period (≈ 1 sec). In the case with noise, the Lyapunov function is decreasing down to a certain “mean” level around which it is fluctuating.

4.5.2 Robustness Improvement Example

The next example illustrates the effectiveness of the robustness modification. In order to make this example clear, a simple but illustrative system will be treated. Let us consider a one DOF robot which is composed of a rotating inertia J without friction driven by a torque τ . This yields

$$\ddot{q} = \tau/J$$

Let us define $u = \tau$ and the state vector:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} q \\ \dot{q} \\ J \end{bmatrix}$$

such that

$$\dot{x} = \begin{bmatrix} x_2 \\ u/x_3 \\ 0 \end{bmatrix}$$

Let us assume we know that

$$J_{\min} < J < J_{\max}$$

Hence, we can define

$$\dot{x} = \begin{bmatrix} x_2 \\ u/x_3 \\ \sigma(x_3) \end{bmatrix} = f(x, u)$$

$$\sigma(x_3) = \begin{cases} -\alpha(x_3 - J_{\min}) & \text{if } x_3 < J_{\min} \\ 0 & \text{if } J_{\min} < x_3 < J_{\max} \\ -\alpha(x_3 - J_{\max}) & \text{if } x_3 > J_{\max} \end{cases}$$

Let us suppose the input torque is corrupted by noise such that $\tau = u + w_1$ and that we assume that J might be time varying, this results in

$$\begin{aligned} \dot{x} &= \begin{bmatrix} x_2 \\ u/x_3 \\ \sigma(x_3) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1/x_3 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \\ &= f(x, u) + G(x)w \end{aligned}$$

We can apply the EKF to this model with

$$\frac{\partial f}{\partial x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -u/x_3^2 \\ 0 & 0 & \frac{\partial \sigma}{\partial x_3} \end{bmatrix}$$

$$\frac{\partial \sigma}{\partial x_3} = \begin{cases} 0 & \text{if } J_{\min} < x_3 < J_{\max} \\ -\alpha & \text{otherwise} \end{cases}$$

with the measurements

$$z = q + v$$

and the computed torque

$$u = x_3(\ddot{q}_d + K_v(\dot{q}_d - \dot{\hat{q}}) + K_p(q_d - \hat{q}))$$

Let us simulate this system using the following conditions:

- the controller and filter are implemented at 200Hz;
- $J = 2$, $J_{\min} = 1$, $J_{\max} = 3$;
- $K_v = 14.1$, $K_p = 100$, $q_d = 0$;
- $q(0) = 2$ rad, $\dot{q}(0) = 0$, $\dot{J}(0) = 1.5$;
- $R(k) = 0.0001$, $Q(k) = 0.01I$, $P(0) = I$;
- $w_1 = \bar{w}_1 + \tilde{w}_1$ with $\bar{w}_1 = 0.5$, $\mathcal{E}\{\tilde{w}_1\} = 0$ and $\text{cov}\{\tilde{w}_1(k), \tilde{w}_1(j)\} = Q(k)\delta_D(k-j)$

Hence, the constant torque bias $\bar{w}_1$ is unaccounted for in the filter. Also, the desired path is not persistently exciting and the system would be unobservable when the tracking error is zero. Simulation results for $\alpha = 5, 1, 0.5, 0.1, 0$ are presented in figures 4.14–4.16. From these results, one can conclude that the unmodified scheme ($\alpha = 0$) is sensitive to the torque bias and this causes the parameter estimate to drift. Also, the parameters error will increase without bounds as $t \rightarrow \infty$. However, the modified controller ($\alpha > 0$) stop this parameter drift as $\hat{x}_3$ crosses the upper bound (at $t \approx 6$ sec) and the depth of penetration gets smaller as α is increased. Furthermore, the parameter's variance stops increasing (actually decreases to a steady

state value which is influenced by the choice of α) when the parameter estimate goes outside the bounds as opposed to the monotonic increase with the unmodified scheme.

One should note that in the initial phase of control, the system is rotating until the errors get to a very small value. Thus in the initial 2 seconds, the system is estimating the value of J . However, there is a residual error caused by the unmodeled torque bias (see fig. 4.16). This torque bias causes a systematic error in the state predictions of the filter and this effect cumulates in time to make the parameter estimate drift. However, this problem is avoided by the modified scheme which guarantees bounded parameter variance and error.

4.6 Conclusion

A new adaptive computed torque controller was presented and shown to be stable under conditions that are closely related to the persistency of excitation of the desired path. The parameter estimation is done using an extended Kalman filter (EKF) based on input torques and position measurements. However, unobservability of the parameter in the linearized system can lead to instability since P_θ might not remain bounded and the parameters are more likely to drift away from their real values. Also, $\hat{D}(q)$ is not guaranteed to remain positive definite. Two, modifications using a priori bounds on θ were introduced such that if $\hat{\theta}$ satisfies these bounds $\hat{D}(q)$ will remain positive definite. Hence, these modifications make the controller bounded-input-bounded-output (BIBO) stable even if the system is not persistently excited.

A discrete-time implementation was suggested and simulations illustrated the performance of the method in both deterministic and stochastic context. Simulations also show that the scheme can be implemented even on a robot with inexpensive 8 bits resolvers and no velocity sensors, hence allowing less expensive robot designs. This example also showed that the scheme is less sensitive to measurement noise

than the methods reviewed in chapter 3. A simple example using the robustness modifications showed that parameter drift is indeed eliminated and that the depth of penetration of the bounds is simply controlled by the value of α in the modified parameter model.

However, the computations required for the model, its partial derivatives, the matrix exponential, the computed torque method and the discrete-time filter make this adaptive scheme very computationally intensive. Detailed computational requirements for the filtering part of the controller depends on the filter mechanization and results from Bierman and Thornton [9, 83] provides all the information needed to determine these requirements. The computation of the robot model and partial derivatives are dependent of the algorithms used and on the type of robot. However, the next chapter presents experimental implementation for a 2 DOF robot in which the real time controller requirements are analysed.

Figure 4.1: Adaptive controller structure

Figure 4.2: The $\sigma(\hat{\theta})$ Function

Figure 4.3: Unmodified and Modified parameter models simulation: modified ($Q_2 = 1$, $\alpha = 10$, $\theta_{\min} = -0.4$, $\theta_{\max} = 0.4$) = solid line, unmodified = dashed and dotted lines

Figure 4.4: Discrete-Time Controller without Noise: Parameter Errors: $\hat{p}_1 - p_1 =$ solid line, $\hat{p}_2 - p_2 =$ dashed line, $\hat{p}_3 - p_3 =$ dotted line, $\hat{p}_4 - p_4 =$ dash-dot line

Figure 4.5: Discrete-Time Controller without Noise: Tracking Errors: $q_1 - q_{1d} =$ solid line, $q_2 - q_{2d} =$ dashed line

Figure 4.6: Discrete-Time Controller without Noise: Filtering Errors: $\hat{q}_1 - q_1 =$ solid line, $\hat{q}_2 - q_2 =$ dashed line

Figure 4.7: Discrete-Time Controller without Noise: Filtering Errors: $\dot{\hat{q}}_1 - \dot{q}_1 =$ solid line, $\dot{\hat{q}}_2 - \dot{q}_2 =$ dashed line

Figure 4.8: Discrete-Time Controller with Noise: Parameter Errors: $\hat{p}_1 - p_1$ = solid line, $\hat{p}_2 - p_2$ = dashed line, $\hat{p}_3 - p_3$ = dotted line, $\hat{p}_4 - p_4$ = dash-dot line

Figure 4.9: Discrete-Time Controller with Noise: Tracking Errors: $q_1 - q_{1d}$ = solid line, $q_2 - q_{2d}$ = dashed line

Figure 4.10: Discrete-Time Controller with Noise: Filtering Errors: $\hat{q}_1 - q_1$ = solid line, $\hat{q}_2 - q_2$ = dashed line

Figure 4.11: Discrete-Time Controller with Noise: Filtering Errors: $\hat{\dot{q}}_1 - \dot{q}_1$ = solid line, $\hat{\dot{q}}_2 - \dot{q}_2$ = dashed line

Figure 4.12: Discrete-Time Controller with Noise: Joint 2 Velocities: real = solid line, filtered = dashed line, direct differentiation = dotted line

Figure 4.13: Discrete-Time Controller Lyapunov Function: without noise = solid line, with noise = dashed line

Figure 4.14: Parameter: Modified Filter: $\alpha = 5 =$ solid line, $\alpha = 1 =$ dashed line, $\alpha = 0.5 =$ dotted line $\alpha = 0.1 =$ dash-dot line, $\alpha = 0 =$ solid line

Figure 4.15: Parameter Variance: Modified Filter: $\alpha = 5 =$ solid line, $\alpha = 1 =$ dashed line, $\alpha = 0.5 =$ dotted line $\alpha = 0.1 =$ dash-dot line, $\alpha = 0 =$ solid line

Figure 4.16: Tracking Error: Modified Filter: $\alpha = 5$ = solid line, $\alpha = 1$ = dashed line, $\alpha = 0.5$ = dotted line $\alpha = 0.1$ = dash-dot line, $\alpha = 0$ = solid line

Chapter 5

Experimental Results

In this chapter, we present experimental results for a two-degree-of-freedom direct-drive robot and we provide a description of the experimental apparatus.

5.1 Experimental Robot

5.1.1 Robot Dynamics

Figure 5.1 presents a schematic of the experimental robot (this is the same configuration as the M.I.T Direct-Drive robot Model IV, see [4]). It is a closed kinematic chain composed of four links (numbered from 0 to 3) operating in the horizontal plane. Here, I_0 and I_1 are links 0 and 1 inertias with respect to the motors axis, I_2 and I_3 are the inertias of links 2 and 3 with respect to their centre of mass, m_2 and m_3 are the masses of links 2 and 3. Defining the parameters:

$$p_1 = I_0 + I_2 + m_2 l_4^2 + m_3 l_2^2$$

$$p_2 = I_1 + I_3 + m_2 l_1^2 + m_3 l_3^2$$

$$p_3 = m_2 l_1 l_4 + m_3 l_2 l_3$$

results in the following kinetic energy for the manipulator (see appendix F):

$$T = \frac{1}{2}p_1\dot{q}_1^2 + \frac{1}{2}p_2\dot{q}_2^2 + p_3\dot{q}_1\dot{q}_2 \cos(q_1 - q_2) \quad (5.1)$$

Applying Lagrange's equation to equation (5.1) yields (see appendix F)

$$\tau_1 = p_1\ddot{q}_1 + p_3 \cos(q_1 - q_2)\ddot{q}_2 + p_3 \sin(q_1 - q_2)\dot{q}_2^2$$

$$\tau_2 = p_2\ddot{q}_2 + p_3 \cos(q_1 - q_2)\ddot{q}_1 - p_3 \sin(q_1 - q_2)\dot{q}_1^2$$

If we want to take into account the viscous friction, in matrix form, we have

$$D(q) = \begin{bmatrix} p_1 & p_3 \cos(q_1 - q_2) \\ p_3 \cos(q_1 - q_2) & p_2 \end{bmatrix}$$

$$h(q, \dot{q}) = p_3 \sin(q_1 - q_2) \begin{bmatrix} \dot{q}_2^2 \\ -\dot{q}_1^2 \end{bmatrix}$$

$$\nu(\dot{q}) = \begin{bmatrix} p_4 \dot{q}_1 \\ p_5 \dot{q}_2 \end{bmatrix}$$

$$g(q) = 0$$

where p_4 and p_5 are the viscous force coefficients.

The input torque are produced by brushless dc motors which are directly mounted on the joints. Here we assume a very simple model for the output torque of the motors (first order model):

$$\tau_i = b_i u_i - c_i \dot{q}_i$$

where the motor inertias have been assimilated to the links 0 and 1, b_i , c_i and u_i are respectively the motor torque constant, the viscous force coefficient and the servo-amplifier input. Hence the model can now be written as

$$Bu = D(q)\ddot{q} + h(q, \dot{q}) + \nu(\dot{q})$$

$$B = \begin{bmatrix} b_1 & 0 \\ 0 & b_2 \end{bmatrix}$$

We can eliminate b_2 by multiplying both sides by $1/b_2$. This results in the same dynamics with new—but still constant—parameters (all the parameters are scaled down by b_2):

$$\ddot{q} = \begin{bmatrix} p_1 & p_3 \cos(q_1 - q_2) \\ p_3 \cos(q_1 - q_2) & p_2 \end{bmatrix}^{-1} \times \left\{ \begin{bmatrix} b_1 & 0 \\ 0 & 1 \end{bmatrix} u - \begin{bmatrix} p_4 & 0 \\ 0 & p_5 \end{bmatrix} \dot{q} - p_3 \sin(q_1 - q_2) \begin{bmatrix} \dot{q}_2^2 \\ -\dot{q}_1^2 \end{bmatrix} \right\} \quad (5.2)$$

where b_1 is now the ratio of the motor torque constants. This ratio will be identified by restricting the angle between links 2 and 3, and by identifying the dynamics of the resulting one-degree-of-freedom robot using either motor 1 or motor 2. Hence, by comparing the experimental results for each motor, it will be straight forward to identify this ratio.

5.1.2 Hardware

The robot is controlled using a 286 based PC compatible running at 10MHz with math coprocessor. The computer is equipped with a data acquisition card (DAC) for analog and digital input/output. The position measurements are obtained from resolvers mounted directly on the motor shafts and these resolver analog outputs are converted by the 12 bits A/D of the servo-amplifiers which are driving the brushless motors (the servo-amplifier use these resolver measurements to do the commutation logic necessary for the operation of the brushless motors). These 12 bits position measurements are read with the digitals ports of the DAC. Also, tachometers are directly mounted on the motor shafts. The tachometers outputs (volts) are converted using the 12 bits A/D of the DAC. The servo-amplifier inputs ($\pm 10V$) are supplied by the 12 bits D/A from the DAC. Timing of the I/O operations is assured by the DAC clock which has a base frequency of 20kHz.

5.1.3 Computational Aspect

In order to implement the adaptive controller in real-time, unnecessary computations should be avoided. Hence, the robot model, its discretization, and the discrete EKF were derived using symbolic programs written in the C language. This was done to generate analytical expressions for the model and the filter so as to obtain efficient code for the real time implementation. In the discretization, the matrix exponential is approximated using one term ($e^{At} \approx I + At$). Also, symmetry of the covariance matrix is implicit in the formulation of the filter code generated by the symbolic program.

The real-time control programs were written in Pascal and the main control loop looked like this:

```

repeat
    t := t+dt;
    compute_desired_state;
    get_measurements;
    filter_states;
    compute_control_input; {compute and send control inputs}
    predict_next_states;
    kalman_gains; {compute covariance matrix and Kalman gains}
until (t >= finaltime);

```

5.1.4 Tachometer Calibration

In the experimental apparatus, both position and velocity in the joints can be measured. Position measurements (angle θ) are obtained using a resolver. Angular velocity ($\dot{\theta}$) are derived from tachometer measurements in volts (u) which are proportional to the angular speed.

$$\dot{\theta} \simeq au$$

The problem of finding the constant of proportionality a is made harder when the robot is assembled because it is no longer possible to make the motor turn in the same direction for a long time to obtain a from a simple integration:

$$a = \frac{\theta(t_f) - \theta(t_0)}{\int_{t_0}^{t_f} u dt} \quad (5.3)$$

When $\theta(t_f) - \theta(t_0)$ and $\int_{t_0}^{t_f} u dt$ are small, equation (5.3) gets very inaccurate. To avoid this problem, we will use an extended Kalman filter to estimate the tachometer gain a . Let us define:

$$x_1 = \theta$$

$$x_2 = a$$

then the system equations are:

$$\dot{x}_1 = x_2 u$$

$$\dot{x}_2 = 0$$

But the tachometer measurements are corrupted by noise, thus:

$$\dot{x}_1 = x_2(u + w)$$

$$\dot{x}_2 = 0$$

where w represent the measurement noise in the tachometer. The position measurements z are also available:

$$z = x_1 + v$$

where v is the noise on position.

Let us reformulate these equations in discrete-time form (T is the time between samples):

$$x_1(k+1) = x_1(k) + T x_2(k) [u(k) + w(k)]$$

$$x_2(k+1) = x_2(k)$$

$$z(k) = x_1(k) + v(k)$$

or

$$\begin{aligned}x(k+1) &= \phi(k+1, k) + \Gamma(x(k))w(k) \\ z(k) &= Hx(k) + v(k)\end{aligned}$$

where

$$\begin{aligned}\phi(k+1, k) &= \begin{bmatrix} x_1(k) + Tx_2(k)u(k) \\ x_2(k) \end{bmatrix} \\ \frac{\partial \phi(k+1, k)}{\partial x(k)} &= \begin{bmatrix} 1 & Tu(k) \\ 0 & 1 \end{bmatrix} \\ \Gamma(x(k)) &= \begin{bmatrix} Tx_2(k) \\ 0 \end{bmatrix} \\ H &= \begin{bmatrix} 1 & 0 \end{bmatrix}\end{aligned}$$

The noiseless system ($w(k) = v(k) = 0$) is linear and is observable at time k (one can reconstruct $x(k)$ from $z(k)$ and $z(k+1)$) if (see appendix C):

$$\sum_{i=k+1}^{k+2} \frac{\partial \phi^T(i, k)}{\partial x} H^T H \frac{\partial \phi(i, k)}{\partial x} > 0 \quad (5.4)$$

where

$$\frac{\partial \phi(i, k)}{\partial x} = \prod_{j=k}^i \frac{\partial \phi(j+1, j)}{\partial x(j)}$$

Equation (5.4) is satisfied for $u(k) \neq 0$. As long as the motor is moving, the system is observable, therefore the discrete extended Kalman filter described in section 2.3 was applied to calibrate the tachometers. This resulted in estimated tachometer gains for motor 1 and 2 of 328 and 244 *deg/sV* respectively.

5.2 One DOF Results

The first experimental results were obtained when the angle between links 2 and 3 was restricted to 90 degrees ($q_{1d} = q_{2d} + 90$). Equation 5.2 is simplified by this

constraint and this results in a robot with one DOF with the following dynamics:

$$\ddot{q} = (p_1 + p_2)^{-1}[b_1 u_1 + u_2 - (p_4 + p_5)\dot{q}]$$

If only motor 1 is used we have

$$\ddot{q} = [u_1 - p_{21}\dot{q}]/p_{11}$$

and if motor 2 is used

$$\ddot{q} = [u_2 - p_{22}\dot{q}]/p_{12}$$

where we have define

$$p_{11} = (p_1 + p_2)/b_1$$

$$p_{21} = (p_4 + p_5)/b_1$$

$$p_{12} = (p_1 + p_2)$$

$$p_{22} = (p_4 + p_5)$$

Thus, p_{11} and p_{12} can be identified in different experiments and we can compute an estimate of b_1 using

$$b_1 \approx \hat{p}_{12}/\hat{p}_{11}$$

5.2.1 Desired Path

The desired trajectory was given by a fifth order polynomial with pauses at the beginning and the end of the displacements (see fig.5.2). Hence, q_d is given by

$$q_d = \begin{cases} q_0 & \text{if } t_0 < t < t_1 \\ q_0 + (q_f - q_0)q_n(\frac{t-t_1}{t_2-t_1}) & \text{if } t_1 < t < t_2 \\ q_f & \text{if } t_2 < t < t_3 \end{cases}$$

$$q_n(t) = 6t^5 - 15t^4 + 10t^3$$

such that $\dot{q}_d(t_1) = \dot{q}_d(t_2) = \ddot{q}_d(t_1) = \ddot{q}_d(t_2) = 0$. This type of trajectory was used to go back and forth with $q_f - q_0 = 50\text{deg}$ with $t_1 - t_0 = 0.1\text{sec}$, $t_2 - t_1 = 1.3\text{sec}$ and $t_3 - t_2 = 0.1\text{sec}$.

5.2.2 Adaptive Controller Parameters

The controller was implemented at the sampling frequency (f_s) of 100Hz. The feedback gains were set at $K_p = \omega_n^2 I$ and $K_v = 2\zeta\omega_n I$ with $\zeta = 1/\sqrt{2}$ and $\omega_n = 10$. Hence, $f_n = \omega_n/2\pi$ was roughly 30 times smaller than the Nyquist frequency ($f_s/2$).

Resolvers' technical specifications from the manufacturer stated an accuracy of $\pm 1/6$ deg (3×10^{-3} rad) which is more than the discretization error of the 12 bits A/D ($\approx 1/24$ deg). Hence, the measurement noise covariance matrix was chosen as $R(k) = 9 \times 10^{-6}$ rad².

Since we used a very simple model for the motors and that we suspect there is stiction in the joints, we selected the input noise of 1% of the full range ± 10 V, yielding $Q_1(k) = 0.04 I V^2$. The value of $Q_2(k)$ was chosen as 10^{-2} for very fast tracking.

5.2.3 Results

The adaptive controller was implemented with the parameter vector

$$\theta = \begin{bmatrix} p_{1i} & p_{2i} \end{bmatrix}^T$$

This resulted in a filter state vector of dimension 4. The controller was first run with motor 1 and

$$\begin{aligned} \hat{\theta}(0) &= \begin{bmatrix} 0.1 & 0.1 \end{bmatrix}^T \\ P(0) &= 0.1 I \end{aligned}$$

Another experiment was done but with

$$\hat{\theta}(0) = \begin{bmatrix} 0.1 & 0.75 \end{bmatrix}^T$$

which are better initial estimates. The results are presented in figures 5.3-5.6. The estimated velocities are very smooth (compared to the velocities that were measured

but not used) and the tracking errors get to very low values after the initial tuning period (1 sec). In the second case, the initial controller performance is better. In both cases, the parameter estimates converged to the same values.

These experiments were repeated with motor 2 and figures 5.7–5.8 present the results. From both sets of results, it is clear that the filtered velocities are better suited for use in the feedback than the measured velocities. This is obvious from the smoothness of the control inputs (see fig. 5.9).

From these results, one notices that the measured positions contain some outliers which are the results of faulty bits in the digital representation of the position. Even if the cause of these outliers was not found, the performance of the adaptive controller does not deteriorate.

5.3 Two DOF Results

From the results of the previous section we were able to estimate the value of $\hat{b}_1 \approx 1.077$. This make sense since the motors are supposed to be identical. The adaptive controller was implemented for the two DOF robot using this value of $\hat{b}_1$.

5.3.1 Desired Path

The desired trajectories were given by fifth order polynomials with pauses at the beginning and the end of the displacements (see fig.5.2). Hence, q_d is given by

$$q_d = \begin{cases} q_0 & \text{if } t_0 < t < t_1 \\ q_0 + (q_f - q_0)q_n\left(\frac{t-t_1}{t_2-t_1}\right) & \text{if } t_1 < t < t_2 \\ q_f & \text{if } t_2 < t < t_3 \end{cases}$$

$$q_n(t) = 6t^5 - 15t^4 + 10t^3$$

such that $\dot{q}_d(t_1) = \dot{q}_d(t_2) = \ddot{q}_d(t_1) = \ddot{q}_d(t_2) = 0$. This type of trajectory was used to go back and forth between $q_0 = \begin{bmatrix} 155 & 70 \end{bmatrix}^T$ deg and $q_f = \begin{bmatrix} 205 & 120 \end{bmatrix}^T$ deg with $t_1 - t_0 = 0.1\text{sec}$, $t_2 - t_1 = 1.8\text{sec}$ and $t_3 - t_2 = 0.1\text{sec}$.

5.3.2 Adaptive Controller Parameters

Since the controller was implemented with a computer with very modest power, the sampling frequency (f_s) was 30Hz. The feedback gains were set at $K_p = \omega_n^2 I$ and $K_v = 2\zeta\omega_n I$ with $\zeta = 1/\sqrt{2}$ and $\omega_n = 5$. Hence, $f_n = \omega_n/2\pi$ was roughly 20 times smaller than the Nyquist frequency ($f_s/2$).

Resolvers' technical specifications from the manufacturer stated an accuracy of $\pm 1/6$ deg (3×10^{-3} rad) which is more than the discretization error of the 12 bits A/D ($\approx 1/24$ deg). Hence, the measurement noise covariance matrix was chosen as $R(k) = 9 \times 10^{-6}$ rad².

Since we used a very simple model for the motors and that we suspect there is stiction in the joints, we selected the input noise of 1% of the full range ± 10 V, yielding $Q_1(k) = 0.04 I V^2$. The value of $Q_2(k)$ was chosen as 9×10^{-4} for slower adaptation than the example in section 5.2.

5.3.3 Results

The adaptive controller was implemented with the parameter vector

$$\theta = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 & p_5 \end{bmatrix}^T$$

This resulted in a filter state vector of dimension 9 (2 positions, 2 velocities, 5 parameters). The controller was first run with

$$\begin{aligned} \hat{\theta}(0) &= \begin{bmatrix} 0.2 & 0.2 & 0.05 & 0.1 & 0.1 \end{bmatrix}^T \\ P(0) &= 0.1 I \end{aligned}$$

The results are presented in figures 5.10–5.14. The estimated velocities are very smooth and the tracking errors get very small after the initial tuning period (8 sec) even if the parameters were still changing slowly. Another experiment was done but with

$$\hat{\theta}(0) = \begin{bmatrix} 0.46 & 0.3 & 0.03 & 0.1 & 0.02 \end{bmatrix}^T$$

which are better initial estimates. The controller performance is better and the parameter estimates converged (see fig.5.15–5.16).

Using these estimated values of the parameters, we implemented a non-adaptive controller with a reduced order EKF (a filter with 4 states, positions and velocities). The reduced order filter allowed us to use a 60Hz sampling frequency and gains with $\omega_n = 9$. This resulted in higher tracking accuracy as shown in figure 5.17. In fact, figure 5.18 shows the RMS tracking errors for different ω_n . From this, it is clear that a faster computer would make the adaptive scheme more efficient (higher sampling rate) and that increasing ω_n does not imply a decrease of the tracking errors since this is a discrete-time controller.

Even if velocity measurements were not used, they were available. Figure 5.19 shows the filtered and measured velocities. Clearly, the filtered velocities are more suitable for use in the feedback law since they are smooth, and therefore, less likely to excite unmodeled dynamics. As in the one DOF cases, the position measurements contained some outliers, but the scheme is robust enough to handle these.

5.3.4 Computational Remarks

The programs used to control the two DOF robot were analysed using a profiler program to verify which parts of the control scheme were the most time-consuming. Tables 5.1–5.2 show the respective times for the routines of the program's main control loop. One can conclude that the extended Kalman filter computations are using most of the CPU time between samples (adaptive controller, 9 state filter). The EKF for the non-adaptive controller (4 state filter) is three times faster hence allowing us to raise the sampling frequency from 30 to 60Hz. This illustrates the fact that only parameters that are known to change should be included in this (or any) adaptive controller so as to obtain faster sampling frequency.

5.4 Conclusion

The new adaptive computed torque controller was implemented with an experimental two-degree-of-freedom direct-drive robot. The parameter estimation is done using an extended Kalman filter (EKF) based on input torques and position measurements. At the same time, the computed torque method uses the filter's estimates of position, velocity and parameters to generate the inputs.

The controller was implemented at a 30Hz sampling frequency with very modest computing power and produced accurate trajectory tracking. The non-adaptive version of this controller was used and this resulted in better tracking. This was mainly caused by the higher sampling frequency (60Hz). Hence, the latest technology in PC computers (386 or 486 based computers) would allow the adaptive controller to be used at frequencies around 100Hz.

This chapter also showed that the scheme was not very sensitive to unmodeled dynamics since we used a very simple model for the dc brushless motors (first order for the velocities). This scheme also simplifies the design of the robot since velocity sensors are no longer needed.

However, this scheme is very computationally intensive and careful coding for the real-time implementation is a crucial step if one wants the application to be successful, but this problem is related to the optimization of real-time computer code and not to adaptive control theory.

Table 5.1: Adaptive Controller Computing Times

Routine	Time per call (msec)	% of T (33.3 msec)
kalman_gains	21.6	64.9
predict_next_states	3.51	10.5
compute_control_input	2.25	6.8
compute_desired_state	2.13	6.4
filter_states	1.25	3.7
get_measurements	0.68	2.1

Table 5.2: Non-Adaptive Controller Computing Times

Routine	Time per call (msec)	% of T (16.7 msec)
kalman_gains	6.88	41.3
predict_next_states	2.41	14.4
compute_control_input	2.11	12.7
compute_desired_state	2.10	12.6
filter_states	0.68	4.1
get_measurements	0.61	3.6

Figure 5.1: Direct-Drive Robot Schematic

Figure 5.2: Desired Path

Figure 5.3: Measured and Desired Trajectories: Motor 1: q_m first case = solid line, q_m second case = dashed line, q_d = dotted line

Figure 5.4: Tracking Errors: Motor 1: $q_m - q_d$ first case = solid line, $q_m - q_d$ second case = dashed line

Figure 5.5: Measured and Filtered Velocities: Motor 1: First Case: $\hat{q}$ = solid line, $\dot{q}_m$ = dashed line

Figure 5.6: Parameter Estimates: Motor 1: $\hat{p}_1$ first case = solid line, $\hat{p}_2$ first case = dashed line, $\hat{p}_1$ second case = dotted line, $\hat{p}_2$ second case = dash-dot line

Figure 5.7: Tracking Errors: Motor 2: $q_m - q_d$ first case = solid line, $q_m - q_d$ second case = dashed line

Figure 5.8: Parameter Estimates: Motor 2: $\hat{p}_1$ first case = solid line, $\hat{p}_2$ first case = dashed line, $\hat{p}_1$ second case = dotted line, $\hat{p}_2$ second case = dash-dot line

Figure 5.9: Control Inputs: Motor 1: u first case = solid line, u second case = dashed line

Figure 5.10: Measured and Desired Trajectories: q_{1m} = solid line, q_{1d} = dashed line, q_{2m} = dotted line, q_{2d} = dash-dot line

Figure 5.11: Tracking Errors: $q_{1m} - q_{1d}$ = solid line, $q_{2m} - q_{2d}$ = dashed line

Figure 5.12: Measured and Filtered Positions: q_{1m} = solid line, $\hat{q}_1$ = dashed line, q_{2m} = dotted line, $\hat{q}_{2d}$ = dash-dot line

Figure 5.13: Filtered velocities: $\hat{q}_{1d}, \hat{q}_{2d}$, = solid line, $\hat{q}_1$ = dashed line, $\hat{q}_2$ = dash-dot line

Figure 5.14: Estimated Parameters

Figure 5.15: Tracking Errors: $q_{1m} - q_{1d}$ = solid line, $q_{2m} - q_{2d}$ = dashed line

Figure 5.16: Estimated Parameters

Figure 5.17: Non-Adaptive controller: Tracking Errors: $q_{1m} - q_{1d}$ = solid line, $q_{2m} - q_{2d}$ = dashed line

Figure 5.18: Errors RMS: 60Hz $\text{RMS}(q_d - q_m)$ = +, 60Hz $\text{RMS}(q_d - \hat{q})$ = x, 30Hz $\text{RMS}(q_d - q_m)$ = o, 30Hz $\text{RMS}(q_d - \hat{q})$ = *

Figure 5.19: Filtered and Measured Velocities: $\dot{q}_1$ = solid line, $\dot{q}_{1m}$ = dashed line

Chapter 6

Optimal Path Planning

In this chapter, we deal with the path planning problem for point-to-point motion using optimal control techniques. Different cost functions will be considered and a non-linear shooting algorithm will be used to solve the minimization problem. These results are not valid for general nonlinear systems. Hence, in this chapter, only robotic manipulators will be considered. Simulations for two-degree of freedom arms are presented to demonstrate the method.

6.1 Introduction

Optimal control techniques have been used to determine manipulator trajectories which minimize a given cost function. The minimum-time problem along a specified path was investigated in [10, 77, 76, 78]. In these cases, the path constraints reduced the $2n$ dimensions problem (position and velocity of the n joints) to a two dimensions problem (position and velocity along the path), thus reducing the computational burden required to find the optimal solution. It is our opinion that those results should be used only when the path constraints have to be strictly enforced to avoid obstacles in the workspace. In most cases, imposing a path constraint when only point-to-point motion is required will result in a significant loss of performance. In [41, 28, 74, 87, 17], the problem of time-optimal path planning for point-to-point

motion was dealt with using different methods. Kahn and Roth [41] were among the first to deal with the time-optimal problem and designed a closed-loop *bang-bang* controller using linearization of the dynamics with respect to the final position. Geering et al [28] solved the two point boundary value problem resulting from the application of Pontryagin's minimum principle, thus they obtained an open-loop solution. Sahar and Hollerbach [74] used dynamic programming to solve the time optimal point-to-point motion problem yielding interesting results, but for robots of three degrees of freedom and more the *curse of dimensionality* inherent to dynamic programming methods arise. Weinreb and Bryson [87] used a modified gradient algorithm to deal with the control bounds and found time-optimal solution for a two degree of freedom robot.

One notices that most of the work dealing with the planning of high speed motion for robotic manipulators involves time-optimal control. The solution to the minimum-time problem has a *bang-bang* control structure. A *bang-bang* controller may excite high frequency unmodeled dynamics, thus cause the manipulator to chatter. Operating a manipulator under such conditions will have deleterious effects on the electrical and mechanical components of the manipulator and will result in higher operating and maintenance cost. In this chapter, we propose a modified cost function which weights both time and energy required for the manipulator to go from one specified point to another. A weighted minimum time-fuel criterion was introduced in Kim and Shin [51], however the resulting control inputs switch rapidly from positive to negative saturation. One would like to design a control method which would result in high speed motion of the manipulator while avoiding abrupt changes in the control inputs. This chapter develops optimal control algorithms which are based on three cost functions:

- minimum-time;

$$J = \int_0^{t_f} dt$$

- weighted time-energy;

$$J = \int_0^{t'} \left(1 + \frac{1}{2} \mu u^T(t) R u(t) \right) dt$$

- energy with fixed final time.

$$J = \int_0^{t'} \frac{1}{2} u^T(t) R u(t) dt$$

These results are not valid for general nonlinear systems as shown in Hermes and LaSalle [34]. However, Chen and Desrochers [16] have shown that for the case of robotic manipulators the optimal solution does exist. Hence, in what follows, only robotic manipulators will be considered. The minimization problems are solved using a parameter optimization procedure or a modified non-linear shooting algorithm. Simulations of two-degree of freedom serial link manipulators are used to evaluate the performance of the controller. The resulting optimal trajectories and control inputs are compared to the time optimal case.

6.2 Minimum-Time Cost Function

As describe in section 2.1, one can formulate the dynamics of the robotic manipulator in state space form such as:

$$\dot{x}(t) = f(x(t), u(t)) = a(x(t)) + B(x(t))u(t) \quad (6.1)$$

where $x(t)$ is a vector of length $2n$ composed of the positions and velocities of each joint (n is the number of degrees of freedom of the manipulator), and the input vector is constrained by;

$$|u_i(t)| \leq 1 \quad \text{for } i = 1 \text{ to } n \quad (6.2)$$

If the inputs saturation values differ from 1, one can normalize the inputs so that equation (6.2) has to be satisfied.

In this section, the following cost function will be considered:

$$J = \int_0^{t_f} dt \quad (6.3)$$

As stated above, one is interested in the point-to-point motion, thus the manipulator should start from $x(0) = x_0$ and then finish its path at $x(t_f) = x_f$, where t_f is free and the control is chosen to satisfy equation (6.2) and minimize (6.3). Using Pontryagin's minimum principle [73, 6], the Hamiltonian is formed:

$$H(x(t), u(t), p(t)) = 1 + p^T(t)f(x(t), u(t)) \quad (6.4)$$

The optimal trajectory is given by:

$$\dot{x}^* = \frac{\partial H}{\partial p} = f(x^*(t), u^*(t)) \quad (6.5)$$

$$\dot{p}^* = -\frac{\partial H}{\partial x} = -\frac{\partial f^T(x^*(t), u^*(t))}{\partial x} p^*(t) \quad (6.6)$$

Where $u^*(t)$ is chosen to provide:

$$H(x^*(t), u^*(t), p^*(t)) \leq H(x^*(t), u(t), p^*(t))$$

for all $|u(t)| \leq 1$

which yields:

$$u^*(t) = -SGN\{B^T(x^*(t))p^*(t)\} \quad (6.7)$$

$$\text{where } SGN\{v\} = \begin{cases} -1 & \text{if } v < 0 \\ 1 & \text{if } v > 0 \end{cases}$$

with boundary conditions:

$$x(0) = x_0, x(t_f) = x_f \text{ and } t_f \text{ free} \quad (6.8)$$

Also, one gets an additional condition:

$$H^*(t) = H^*(t_f) = 0 \quad (6.9)$$

In the equation (6.7) for $u^*(t)$, one notices that, if the problem is not singular, the inputs are sequences of positive and negative saturations (a *bang-bang* controller).

6.2.1 Solution to the Time-Optimal Problem

Equations (6.5), (6.6), (6.7), (6.8) and (6.9) form a non-linear two point boundary value problem (TPBVP) with unknown final time which is very hard to solve. Thus, to avoid to solve this TPBVP, the method described in [88] will be used. In this method, one specifies the sequences of the sign of the different inputs and then tries to minimize the final miss by adjusting the switching times and the final time. let us give the details for this technique. First, one defines a cost function for the final miss:

$$J = \frac{1}{2}(x(t_f) - x_f)^T(x(t_f) - x_f) \quad (6.10)$$

which results from the applications of the inputs $u(t)$ for $0 \leq t \leq t_f$. One can define the N switching times t_i at which the S_i component of $u(t)$ changes sign with $0 \leq t_i \leq t_f$ and $1 \leq S_i \leq n$. For notation convenience, let us define $t_{N+1} = t_f$. If the gradient:

$$\frac{\partial J}{\partial t} = \left[\frac{\partial J}{\partial t_1}, \frac{\partial J}{\partial t_2}, \dots, \frac{\partial J}{\partial t_{N+1}} \right]^T \quad (6.11)$$

can be computed, then a conjugate gradient algorithm can be used to minimize J . The following procedure from [88] shows how $\frac{\partial J}{\partial t_i}$ can be obtained.

For the system

$$\dot{x}(t) = f(x(t), u(t)) = a(x(t)) + B(x(t))u(t) \quad (6.12)$$

let us define

$$\eta_i(t) = \frac{\partial x(t)}{\partial t_i} \quad (6.13)$$

which results in:

$$\frac{\partial J}{\partial t_i} = (x(t_{N+1}) - x_f)^T \eta_i(t_{N+1}) \quad (6.14)$$

The vectors η_i are the influence vectors which are obtained from:

$$\eta_i(t) = 0 \text{ for } t < t_i \quad (6.15)$$

$$\eta_i(t_i) = B(x)(u(t_i^-) - u(t_i^+)) \quad (6.16)$$

$$\dot{\eta}_i(t) = \frac{\partial f(x, u)}{\partial x} \eta_i(t) \text{ for } t > t_i \quad (6.17)$$

$$\eta_{N+1}(t_{N+1}) = f(x(t_{N+1}), u(t_{N+1})) \quad (6.18)$$

where $u(t_i^-)$ are the inputs values just before t_i and $u(t_i^+)$ just after t_i .

6.3 Weighted Time-Energy Cost Function

The dynamics of the robot can be given in state space form:

$$\dot{x}(t) = f(x(t), u(t)) = a(x(t)) + B(x(t))u(t) \quad (6.19)$$

where the input vector is constrained by;

$$|u_i(t)| \leq 1 \text{ for } i = 1 \text{ to } n \quad (6.20)$$

Most of the work found in the literature was dealing with the minimum time problem, or the minimization of the following criterion:

$$J = \int_0^{t_f} dt$$

This minimization results in a controller with a *bang-bang* structure which is undesirable. Trying to avoid this problem, one wants to minimize a criterion which resembles the time-optimal criterion, but has an *energy* term added to it:

$$J = \int_0^{t_f} \left(1 + \frac{1}{2} \mu u^T(t) R u(t) \right) dt \quad (6.21)$$

Here, μ is the relative weight of the *energy* with respect to the time and R is a positive definite symmetric matrix. This could be chosen to either represent the respective economical cost of the two terms, or provide fast response and smooth transition (an

allowable jerk) between input saturations. Here is an example, although simplistic, which illustrate how μ can be chosen. One can define $\mu = \beta/\alpha$; where α is the ratio of the fixed cost related to the manipulator and time it can be used, and β is twice the cost of the energy. So the respective units could be $\alpha = [\$/\text{hr}]$, $\beta = [\$/\text{Kwhr}]$, $u^T R u = [\text{kW}]$, and $\mu = [1/\text{kW}]$.

As stated above, one is interested in the point-to-point motion, thus the manipulator should start from $x(0) = x_0$ and then finish its path at $x(t_f) = x_f$, where t_f is free and the control is chosen to satisfy equation (6.20) and minimize (6.21). Using Pontryagin's minimum principle [73, 6], the Hamiltonian is formed:

$$H(x(t), u(t), p(t)) = 1 + \frac{1}{2} \mu u^T(t) R u(t) + p^T(t) f(x(t), u(t)) \quad (6.22)$$

The optimal trajectory is given by:

$$\dot{x}^* = \frac{\partial H}{\partial p} = f(x^*(t), u^*(t)) \quad (6.23)$$

$$\dot{p}^* = -\frac{\partial H}{\partial x} = -\frac{\partial f^T(x^*(t), u^*(t))}{\partial x} p^*(t) \quad (6.24)$$

Where $u^*(t)$ is chosen to provide:

$$H(x^*(t), u^*(t), p^*(t)) \leq H(x^*(t), u(t), p^*(t))$$

for all $|u(t)| \leq 1$

which yields:

$$u^*(t) = -SAT \left\{ \frac{1}{\mu} R^{-1} B^T(x^*(t)) p^*(t) \right\} \quad (6.25)$$

$$\text{where } SAT\{v\} = \begin{cases} -1 & \text{if } v < -1 \\ v & \text{if } |v| \leq 1 \\ 1 & \text{if } v > 1 \end{cases}$$

with boundary conditions:

$$x(0) = x_0, x(t_f) = x_f \text{ and } t_f \text{ free} \quad (6.26)$$

Also, one gets an additional condition:

$$H^*(t) = H^*(t_f) = 0 \quad (6.27)$$

Equations (6.23), (6.24), (6.25), (6.26) and (6.27) form a non-linear two point boundary value problem (TPBVP) with unknown final time. An advantage of this criterion is that the problem of singular path can not arise with this criterion because the Hamiltonian can not be independent of $u(t)$ at any moment and $SAT\{0\}$ is well defined at 0. For μ going to zero the saturation function ($SAT\{v/\mu\}$) is equivalent to the sign function ($SGN\{v\}$) of the time optimal problems as shown in [17]. In the above reference, the cost function (6.21) is also introduced but as a means to find the time optimal control not as a goal in itself.

6.3.1 Solution to the TPBVP

To solve this problem, a non-linear shooting algorithm has been used with t_f and $p(0)$ as adjustable parameters, and $x(t_f) = x_f$ and $H(t_f) = 0$ as targets. The procedure goes as follow [11]:

1. Guess t_f and p_0 .
2. Integrate (6.23) and (6.24) until $t = t_f$, with $p(0) = p_0$, and $u(t)$ given by (6.25). Memorize $\delta x = x(t_f) - x_f$ and $H(t_f)$.
3. Put $t'_f = t_f + \delta t_f$. Integrate (6.23) and (6.24) until $t = t'_f$, with $p(0) = p_0$, and $u(t)$ given by (6.25). Memorize $\delta x_i = x(t'_f) - x(t_f)$ and $\delta H_i = H(t'_f) - H(t_f)$.
4. For $i = 1$ to n
 - (a) Put $p(0) = p_0$, and $p_i(0) = p_{0i} + \delta p_i$.
 - (b) Integrate (6.23) and (6.24) until $t = t_f$, with $p(0)$ given by 4a, and $u(t)$ given by (6.25).

(c) Memorize $\delta x_i = x_i(t_f) - x(t_f)$ and $\delta H_i = H_i(t_f) - H(t_f)$.

5. Form the influence matrix ($n + 1$ by $n + 1$):

$$A = \begin{bmatrix} \frac{\delta x_1}{\delta p_1} & \frac{\delta x_2}{\delta p_2} & \dots & \frac{\delta x_n}{\delta p_n} & \frac{\delta x_1}{\delta t_f} \\ \frac{\delta H_1}{\delta p_1} & \frac{\delta H_2}{\delta p_2} & \dots & \frac{\delta H_n}{\delta p_n} & \frac{\delta H_1}{\delta t_f} \end{bmatrix}$$

6. Solve the linear system:

$$\begin{bmatrix} \delta x \\ H \end{bmatrix} = A \begin{bmatrix} dp \\ dt_f \end{bmatrix}$$

7. Update $p_0 = p_0 + \epsilon dp$ and $t_f = t_f + \epsilon dt_f$, where $0 < \epsilon \leq 1$. ϵ is chosen so that $\|x(t_f) - x_f\| + \|H(t_f)\|$ is minimized. If $\|x(t_f) - x_f\| + \|H(t_f)\|$ is small enough stop, otherwise go to 2.

6.4 Energy Cost Function

The dynamics of the robot can be given in state space form:

$$\dot{x}(t) = f(x(t), u(t)) = a(x(t)) + B(x(t))u(t) \quad (6.28)$$

where the input vector is constrained by;

$$|u_i(t)| \leq 1 \quad \text{for } i = 1 \text{ to } n \quad (6.29)$$

For cases when the final time is specified, one can use the following cost function:

$$J = \int_0^{t_f} \frac{1}{2} u^T(t) R u(t) dt \quad (6.30)$$

where R is a positive definite symmetric matrix.

As stated above, one is interested in the point-to-point motion, thus the manipulator should start from $x(0) = x_o$ and then finish its path at $x(t_f) = x_f$, where t_f is

fixed and the control is chosen to satisfy equation (6.29) and minimize (6.30). Using Pontryagin's minimum principle [73, 6], the Hamiltonian is formed:

$$H(x(t), u(t), p(t)) = \frac{1}{2}u^T(t)Ru(t) + p^T(t)f(x(t), u(t)) \quad (6.31)$$

The optimal trajectory is given by:

$$\dot{x}^* = \frac{\partial H}{\partial p} = f(x^*(t), u^*(t)) \quad (6.32)$$

$$\dot{p}^* = -\frac{\partial H}{\partial x} = -\frac{\partial f^T(x^*(t), u^*(t))}{\partial x}p^*(t) \quad (6.33)$$

Where $u^*(t)$ is chosen to provide:

$$H(x^*(t), u^*(t), p^*(t)) \leq H(x^*(t), u(t), p^*(t))$$

$$\text{for all } |u(t)| \leq 1$$

which yields:

$$u^*(t) = -SAT \{R^{-1}B^T(x^*(t))p^*(t)\} \quad (6.34)$$

with boundary conditions:

$$x(0) = x_0 \text{ and } x(t_f) = x_f \quad (6.35)$$

Equations (6.32), (6.33), (6.34) and (6.35) form a non-linear two point boundary value problem (TPBVP).

6.4.1 Solution to the TPBVP

To solve this problem, a non-linear shooting algorithm has been used with $p(0)$ as adjustable parameters, and $x(t_f) = x_f$ as target. The procedure goes as follow [11]:

1. Guess p_0 .
2. Integrate (6.32) and (6.33) until $t = t_f$, with $p(0) = p_0$, and $u(t)$ given by (6.34). Memorize $\delta x = x(t_f) - x_f$.

3. For $i = 1$ to n

(a) Put $p(0) = p_0$, and $p_i(0) = p_{0i} + \delta p_i$.

(b) Integrate (6.32) and (6.33) until $t = t_f$, with $p(0)$ given by 3a, and $u(t)$ given by (6.34).

(c) Memorize $\delta x_i = x_i(t_f) - x(t_f)$.

4. Form the influence matrix (n by n):

$$A = \begin{bmatrix} \frac{\delta x_1}{\delta p_1} & \frac{\delta x_2}{\delta p_2} & \dots & \frac{\delta x_n}{\delta p_n} \end{bmatrix}$$

5. Solve the linear system:

$$\delta x = A \delta p$$

6. Update $p_0 = p_0 + \epsilon \delta p$, where $0 < \epsilon \leq 1$. ϵ is chosen so that $\|x(t_f) - x_f\|$ is minimized. If $\|x(t_f) - x_f\|$ is small enough stop, otherwise go to 2.

6.5 Examples

For the first example, the planar section of the cylindrical robot Automelec ACR (fig.6.1) described in [28] is used. Its dynamics are given by:

$$\begin{aligned} \dot{x}_1 &= x_2(t) \\ \dot{x}_2 &= \frac{u_1(t) + m_a x_1(t) x_4^2(t) + m_n \{r_o + x_1(t)\} x_4^2(t)}{m_a + m_n} \\ \dot{x}_3 &= x_4(t) \\ \dot{x}_4 &= \frac{u_2(t) - m_2 x_1(t) x_2(t) x_4(t) - 2m_n \{r_o + x_1(t)\} x_2(t) x_4(t)}{\theta_{tot}(x_1(t))} \end{aligned}$$

where

$$\begin{aligned} \theta_{tot}(x_1(t)) &= \theta_t + \theta_o + m_a x_1^2(t) + m_n \{r_o + x_1(t)\}^2 \\ |u_1(t)| &\leq F_{max} \quad \text{and} \quad |u_2(t)| \leq M_{max} \end{aligned}$$

where $x_1(t)$ is the prismatic joint position (radius in [m]), $x_3(t)$ is the position rotating joint (angle in [rad]), $u_1(t)$ is the input force in the prismatic joint, and $u_2(t)$ is the input torque in the rotating joint. Also θ_t is the mass moment of inertia of the robot with the arm and load, θ_o is the mass moment of inertia of the arm with respect to its center of gravity, m_a is the mass of the arm, m_n is the mass of the load, and r_o is the distance between the load and the center of gravity. The following numerical values were used:

$$\begin{aligned} m_a &= 3.7kg, & m_n &= 4.6kg, & r_o &= 0.37m, & \theta_t &= 0.28m^2kg, \\ \theta_o &= 0.09m^2kg, & F_{max} &= 15N, & M_{max} &= 5Nm. \end{aligned}$$

For this robot, we want the path to start at $x_o = [0.15, 0, \pi/2, 0]$, and finish at $x_f = [0.15, 0, 0, 0]$. Optimal solutions were generated for $\mu = 0.002, 0.005$ with $R = I$ (the identity matrix) and are compared with the time optimal solution (fig.6.2–6.7).

One sees that the final times are almost the same but the control efforts are quite different (see table 6.1). The trade-off between the augmentation of the final time and the control effort is very favourable to the use of the weighted time-energy cost function. Also the time-optimal solutions have *corners* in their velocity profiles (thus infinite jerk). The trajectory with $\mu = 0.01$ and 0.005 do not have those *corners*, the jerk stays finite everywhere. One notices that even if the initial and final radii are equal, the radius diminishes in the first part of the path and comes back to its initial value so that the effective inertia of the rotating joint is smaller and the response is faster and requires less energy. The results with $\mu = 0$ are very good agreement with those obtained in [28]. Imposing an a priori path leads to a decrease in performance of the robotic manipulator (in this example the final time is 1.46 sec, 25% higher, if the radius is constraint to be constant for the minimum-time trajectory).

The second example also comes from [28], which is the planar section of the IBM 7535 B 04 (fig.6.8), composed of two revolute joints. The dynamics are:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \{\theta_7[u_1 - u_2 + \theta_6(x_2 + x_4)^2 \sin(x_3)]\} \end{aligned}$$

$$\begin{aligned}
& -\theta_6[u_2 - \theta_6 x_2^2 \sin(x_3)] \cos(x_3) \} / \theta_{tot} \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= \{ [\theta_5 + \theta_6 \cos(x_3)] [u_2 - \theta_6 x_2^2 \sin(x_3)] \\
& - [\theta_7 + \theta_6 \cos(x_3)] [u_1 - u_2 + \theta_6 (x_2 + x_4)^2 \sin(x_3)] \} / \theta_{tot}
\end{aligned}$$

where

$$\begin{aligned}
\theta_{tot}(x_3(t)) &= \theta_7 \theta_5 - \theta_6^2 \cos(x_3) \\
|u_1(t)| &\leq M_{1max} \quad \text{and} \quad |u_2'(t)| \leq M_{2max}
\end{aligned}$$

where $x_1(t)$ and $x_3(t)$ are respectively the first and second revolute joint position (angle in [rad]), $u_1(t)$ and $u_2(t)$ the first and second joint input torques. Also θ_1 is the mass moment of inertia of link 1 with respect to the first axis, θ_2 is the mass moment of inertia of link 2 with respect to the second axis, θ_3 is the mass moment of inertia of the hand and load with respect to the hand, m_2 is mass of the second link, m_3 is mass of the hand and load, l_1 and l_2 are the respective length of the links, and r_2 is the distance between the center of gravity of the second link and its driving axis. The following numerical values were used:

$$\begin{aligned}
\theta_1 &= 1.6m^2kg, & \theta_2 &= 0.43m^2kg, & \theta_3 &= 0.01m^2kg, \\
l_1 &= 0.4m, & l_2 &= 0.25m, & r_2 &= 0.125m, \\
m_2 &= 15.0kg, & m_3 &= 6.0m, & \theta_4 &= \theta_2 + l_2^2 m_3, \\
\theta_5 &= \theta_1 + l_1^2(m_2 + m_3), & \theta_6 &= l_1(r_2 m_2 + l_2 m_3), & \theta_7 &= \theta_3 + \theta_4
\end{aligned}$$

The path should start at $x_o = [0.78, 0, 0, 0]$, and end at $x_f = [0, 0, 0, 0]$. This correspond to the case 1 in [17], and the A_o class in [28], which correspond to a control sequence of $\{-, +\}$ for u_1 and $\{+, -, +\}$ for u_2 . Optimal solutions were generated for $\mu = 0.001, 0.0025$ with $R = I$ (the identity matrix) and are compared with the time optimal solution (fig.6.9–6.14).

All the previous comments about control effort, high jerk and *corners* in the velocity profiles also apply here (see table 6.2). The trajectories show there is a sort

of whip effect at the end of the path which is mainly caused by the high velocities of the second link. The results with $\mu = 0$ are consistent with those in [28, 17]. The optimal solutions for this manipulator required more computational effort than the one of the previous example because of the higher non-linearity of this robot dynamics.

We have not presented results for the energy cost function since once the final time is fixed we obtain the same results as the one for which the value of μ in the weighted time-energy cost function would have produced the same final time (e.g. with the IBM 7535 B 04 robot, fixing $t_f = 1.17$ gives exactly the same results as with $\mu = 0.0025$).

6.6 Computational Remarks

The non-linear shooting algorithms are very sensitive to the choice of δp (and δt_f for the weighted time-energy cost function), because those are used to do a numerical differentiation to estimate the matrix A . A Runge-Kutta scheme with adaptive step size is used to assure a desired precision, otherwise the matrix A can lose all significance and often becomes ill-conditioned. Also, one can not try to find the solution for very small μ without proceeding from higher one to smaller one in steps that assure convergence of the algorithm. The non-linear shooting algorithm was first implemented in single precision and worked well with the Automelec ACR robot. However, we had to convert the program to double precision to find solutions for the IBM 7535 B 04 robot. Even in double precision, the algorithm was not robust, and should be improved before trying it with three degree of freedom robots.

6.7 Conclusion

The weighted time-energy criterion generated a path with slightly longer travel time ($\approx 5\text{--}10\%$) than the time-optimal trajectory, but with almost half the control effort. As stated earlier singularity problems do not arise. Another advantage of this criterion is that if some joints of the robot are completely decoupled the time optimal solution will more than likely be singular in some points; with the energy term added, this is avoided. Avoiding very large change in the control effort, by adding an energy term to the time-optimal cost function, will augment the life expectancy of the mechanical parts and the actuators of the robot. In the examples, the matrix R was chosen as the identity matrix, using different values for R should lead to other interesting results. A more complete study with robots of more than 2 DOF should provide interesting results, but one would like to improve the efficiency of the computational algorithm used to solved the TPBVP which has a tendency to diverge for highly non-linear systems. However, when one has computed an optimal path, other paths (neighboring paths) are easier to obtained.

Figure 6.2: ACR robot: Time-optimal trajectory, joint 1: position = solid line, velocity = dashed line, normalized input = dash-dot line

Figure 6.3: ACR robot: Time-optimal trajectory, joint 2: position = solid line, velocity = dashed line, normalized input = dash-dot line

Figure 6.4: ACR robot: Optimal trajectory $\mu = 0.002$, joint 1: position = solid line, velocity = dashed line, normalized input = dash-dot line

Figure 6.5: ACR robot: Optimal trajectory $\mu = 0.002$, joint 2: position = solid line, velocity = dashed line, normalized input = dash-dot line

Figure 6.6: ACR robot: Optimal trajectory $\mu = 0.005$, joint 1: position = solid line, velocity = dashed line, normalized input = dash-dot line

Figure 6.7: ACR robot: Optimal trajectory $\mu = 0.005$, joint 2: position = solid line, velocity = dashed line, normalized input = dash-dot line

Table 6.2: Results for the IBM 7535 B 04

μ	t_f (sec)	$\int_0^{t_f} (u^T R u) dt$
0	0.99	703.8
0.001	1.05	423.3
0.0025	1.17	270.7

Figure 6.8: Robot IBM 7535 B 04 schematic

Figure 6.9: IBM robot: Time-optimal trajectory, joint 1: position = solid line, velocity = dashed line, normalized input = dash-dot line

Figure 6.10: IBM robot: Time-optimal trajectory, joint 2: position = solid line, velocity = dashed line, normalized input = dash-dot line

Figure 6.11: IBM robot: Optimal trajectory $\mu = 0.001$, joint 1: position = solid line, velocity = dashed line, normalized input = dash-dot line

Figure 6.12: IBM robot: Optimal trajectory $\mu = 0.001$, joint 2: position = solid line, velocity = dashed line, normalized input = dash-dot line

Figure 6.13: IBM robot: Optimal trajectory $\mu = 0.0025$, joint 1: position = solid line, velocity = dashed line, normalized input = dash-dot line

Figure 6.14: IBM robot: Optimal trajectory $\mu = 0.0025$, joint 2: position = solid line, velocity = dashed line, normalized input = dash-dot line

Chapter 7

Conclusions

In this work, we first reviewed and compared different adaptive controllers. Although these can be discretized successfully, it was shown that they are sensitive to measurement noise. For the direct methods, convergence of the parameter estimates is slow since these are gradient schemes as opposed to the fast convergence of the indirect methods which are least-squares estimators. Also, there is no guide-lines to choose the adaptation gains used in the direct methods.

Then, a new adaptive controller was developed and conditions under which the scheme is stable were established. Two modifications were introduced so as to increase the robustness of the scheme and to preserve the positive definiteness of $\hat{D}(q)$. This resulted in a bounded-input-bounded-output stable controller even when the system is not persistently excited. A discrete-time implementation was suggested and simulations illustrated the performance and the noise rejection properties of the new controller. This controller was implemented successfully to control an experimental two-degree-of-freedom robot. However, this implementation showed that the main disadvantage of this scheme is its computational complexity. From the experimental results, one can also conclude that the scheme is not very sensitive to unmodeled dynamics (very simple models were used for the motors).

Finally, we presented results in the area of optimal path planning. They showed that the weighted time-energy cost function generated paths with slightly longer

travel time with smoother inputs and less control effort than the time-optimal paths.

Since the computational burden of the proposed controller is its main disadvantage, would it be possible to separate the parameter estimation and the filtering used to generate velocity (and perhaps acceleration) estimates so as to obtain a stable controller requiring less computational effort? Also, optimal path planning and adaptive control were treated independently, but what would be the results if these two were combined (see fig. 7.1)? Are optimal paths persistently exciting? These are certainly interesting topics for future work.

Figure 7.1: Combined adaptive and optimal control

Appendix A

Lagrangian Dynamics

A.1 Motion Equations

In this section we state well known results [26, 35, 84] used to describe the motion equations of robotic manipulators made of an open kinematic chain of rigid bodies. Applying Lagrange's equation to the robotic manipulator for which the position is described using 4x4 matrices (Denavit-Hartenberg representation [21]) results in the following equation for the dynamics of the robotic manipulator:

$$\tau = D(q)\ddot{q} + h(q, \dot{q}) + g(q)$$

where q , $\dot{q}$ and $\ddot{q}$ are respectively the manipulator joint position, velocity and acceleration vectors; τ is torque vector applied to the joints. Also:

- $D(q)$ is the symmetric inertia or mass matrix whose elements are:

$$D_{ik} = \sum_{j=\max(i,k)}^n \text{tr}(U_{jk}J_jU_{ji}^T)$$

- $h(q, \dot{q})$ is the Coriolis and centrifugal force vector given by:

$$h(q, \dot{q}) = \begin{bmatrix} \dot{q}^T H_1(q) \dot{q} \\ \dot{q}^T H_2(q) \dot{q} \\ \dots \\ \dot{q}^T H_n(q) \dot{q} \end{bmatrix}$$

where the $H_i(q)$ are symmetric matrices whose elements are:

$$H_{ikm} = \sum_{j=\max(i,k,m)}^n \text{tr}(U_{jkm} J_j U_{ji}^T)$$

$h(q, \dot{q})$ can be written as:

$$h(q, \dot{q}) = C(q, \dot{q}) \dot{q}$$

$$C(q, \dot{q}) = \begin{bmatrix} \dot{q}^T H_1(q) \\ \dot{q}^T H_2(q) \\ \dots \\ \dot{q}^T H_n(q) \end{bmatrix}$$

- and $g(q)$ is the gravitational force vector:

$$g_i = - \sum_{j=i}^n m_j g U_{ji} \bar{r}_j$$

- $\bar{r}_j$ is the position of link j center of mass given with respect to link j coordinate frame:

$$\bar{r}_j = \begin{bmatrix} \bar{x}_j & \bar{y}_j & \bar{z}_j & 1 \end{bmatrix}^T$$

- g is gravity row vector expressed in the based coordinate frame (link 0):

$$g = \begin{bmatrix} g_x & g_y & g_z & 0 \end{bmatrix}$$

- J_i are the pseudo inertia matrices in which all the inertias are expressed in the link i coordinate frame:

$$J_i = \begin{bmatrix} \frac{-I_{ixx} + I_{iyy} + I_{izz}}{2} & I_{ixy} & I_{ixz} & m_i \bar{x}_i \\ I_{ixy} & \frac{I_{ixx} - I_{iyy} + I_{izz}}{2} & I_{iyz} & m_i \bar{y}_i \\ I_{ixz} & I_{iyz} & \frac{I_{ixx} + I_{iyy} - I_{izz}}{2} & m_i \bar{z}_i \\ m_i \bar{x}_i & m_i \bar{y}_i & m_i \bar{z}_i & m_i \end{bmatrix}$$

- U_{ij} and U_{ijk} are partial derivatives of the transformation matrices W_0^i such as:

$$\begin{aligned}
 U_{ij} &= \frac{\partial W_0^i}{\partial q_j} = \begin{cases} W_0^{j-1} Q_j W_{j-1}^i & \text{if } j \leq i \\ 0 & \text{if } j > i \end{cases} \\
 U_{ijk} &= \frac{\partial^2 W_0^i}{\partial q_j \partial q_k} = \begin{cases} W_0^{j-1} Q_j W_{j-1}^{k-1} Q_k W_{k-1}^i & \text{if } j \leq k \leq i \\ W_0^{k-1} Q_k W_{k-1}^{j-1} Q_j W_{j-1}^i & \text{if } k \leq j \leq i \\ 0 & \text{if } j > i \text{ or } k > i \end{cases} \\
 U_{ijk} &= U_{ikj}
 \end{aligned}$$

where

$$\begin{aligned}
 Q_i &= \begin{bmatrix} 0 & s_i - 1 & 0 & 0 \\ 1 - s_i & 0 & 0 & 0 \\ 0 & 0 & 0 & s_i \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
 s_i &= \begin{cases} 0 & \text{if } q_i = \theta_i \text{ revolute joint} \\ 1 & \text{if } q_i = d_i \text{ prismatic joint} \end{cases}
 \end{aligned}$$

- W_0^i is the position (4x4 matrix) of the coordinate frame attached to link i given with respect to the based coordinate frame and W_{i-1}^i is the transformation matrix from link $i-1$ to link i using the Denavit-Hartenberg representation:

$$\begin{aligned}
 W_{i-1}^i &= \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 W_{m-1}^n &= W_{m-1}^m W_m^{m+1} \dots W_{n-1}^n
 \end{aligned}$$

A.2 Partial Derivatives of $D(q)$, $h(q, \dot{q})$ and $g(q)$

From the results of section A.1 it is straight forward to derive the partial derivatives:

$\frac{\partial D}{\partial q}$, $\frac{\partial h}{\partial q}$, $\frac{\partial h}{\partial \dot{q}}$, and $\frac{\partial g}{\partial q}$.

- $\frac{\partial D}{\partial q}$ is a three dimensional array whose components are:

$$\begin{aligned}\frac{\partial D_{ik}}{\partial q_m} &= \sum_{j=\max(i,k,m)}^n \text{tr}(U_{jkm} J_j \ddot{U}_{ji}^T + U_{jkm} J_j U_{jim}^T) \\ \frac{\partial D_{ik}}{\partial q_m} &= \frac{\partial D_{ki}}{\partial q_m} = H_{ikm} + H_{kim}\end{aligned}$$

- $\frac{\partial h}{\partial \dot{q}}$ is a matrix given by:

$$\begin{aligned}\frac{\partial h(q, \dot{q})}{\partial \dot{q}} &= 2 \begin{bmatrix} \dot{q}^T H_1(q) \\ \dot{q}^T H_2(q) \\ \dots \\ \dot{q}^T H_n(q) \end{bmatrix} \\ \frac{\partial h(q, \dot{q})}{\partial \dot{q}} &= 2C(q, \dot{q})\end{aligned}$$

- $\frac{\partial h}{\partial q}$ is a matrix given by:

$$\frac{\partial h(q, \dot{q})}{\partial q} = \begin{bmatrix} \dot{q}^T \frac{\partial H_1(q)}{\partial q} \dot{q} \\ \dot{q}^T \frac{\partial H_2(q)}{\partial q} \dot{q} \\ \dots \\ \dot{q}^T \frac{\partial H_n(q)}{\partial q} \dot{q} \end{bmatrix}$$

where $\frac{\partial H_i(q)}{\partial q}$ are three dimensional arrays such as:

$$\frac{\partial H_{ikm}}{\partial q_l} = \sum_{j=\max(i,k,m,l)}^n \text{tr}(U_{jklm} J_j U_{ji}^T + U_{jkm} J_j U_{jl}^T)$$

- $\frac{\partial g}{\partial q}$ is a matrix:

$$\frac{\partial g_i}{\partial q_m} = - \sum_{j=\max(i,m)}^n m_j g U_{jim} \bar{r}_j$$

Lagrangian formulation leads to a matrice $D(q)$ and vectors $h(q, \dot{q})$ and $g(q)$ which are linear in the matrices J_i . The matrices J_i are linear in the inertial parameters of the links. This results in:

$$\begin{aligned}\sum_{i=1}^n \frac{\partial D}{\partial \theta_i} \theta_i &= D(q, \theta) \\ \frac{\partial h}{\partial \theta} \theta &= h(q, \dot{q}, \theta) \\ \frac{\partial g}{\partial \theta} \theta &= g(q, \theta)\end{aligned}$$

where

$$\begin{aligned}\frac{\partial D_{ik}}{\partial \theta_p} &= \sum_{j=\max(i,k)}^n \text{tr}(U_{jk} \frac{\partial J_j}{\partial \theta_p} U_{ji}^T) \\ \frac{\partial H_{ikm}}{\partial \theta_p} &= \sum_{j=\max(i,k,m)}^n \text{tr}(U_{jk} \frac{\partial J_j}{\partial \theta_p} U_{ji}^T) \\ \frac{\partial g_i}{\partial \theta_p} &= - \sum_{j=i}^n g U_{ji} \frac{\partial m_j \bar{r}_j}{\partial \theta_p}\end{aligned}$$

Appendix B

Recursive Least Squares

B.1 Continuous-Time Recursive Least Squares

The results from this section follow [5, 56, 57].

B.1.1 Unnormalized

If a system model satisfies the equation:

$$y(t) = \varphi^T(t)\theta$$

where θ is unknown but constant, and $y(t)$, $\varphi(t)$ are bounded (in L_∞), then the following identification scheme with $\alpha > 0$ produces an $e \in L_2 \cap L_\infty$ and $\theta - \hat{\theta} \in L_\infty$.

$$\begin{aligned}\frac{d\hat{\theta}}{dt} &= \alpha P(t)\varphi(t)e(t) \\ e(t) &= y(t) - \varphi^T(t)\hat{\theta}(t) \\ \frac{dP(t)}{dt} &= -\alpha P(t)\varphi(t)\varphi^T(t)P(t)\end{aligned}$$

Proof: Choose the Lyapunov function:

$$V(t) = (\theta - \hat{\theta}(t))^T P^{-1}(t)(\theta - \hat{\theta}(t))$$

then

$$\begin{aligned}
 \dot{V}(t) &= (\theta - \hat{\theta}(t))^T \frac{dP^{-1}(t)}{dt} (\theta - \hat{\theta}(t)) - 2(\theta - \hat{\theta}(t))^T P^{-1}(t) \dot{\hat{\theta}} \\
 &= -(\theta - \hat{\theta}(t))^T P^{-1}(t) \dot{P}(t) P^{-1}(t) (\theta - \hat{\theta}(t)) \\
 &\quad - 2(\theta - \hat{\theta}(t))^T P^{-1}(t) (\alpha P(t) \varphi(t) e(t)) \\
 &= \alpha (\theta - \hat{\theta}(t))^T \varphi(t) \varphi^T(t) (\theta - \hat{\theta}(t)) - 2\alpha (\theta - \hat{\theta}(t))^T \varphi(t) e(t) \\
 &= -\alpha e^T e \leq 0
 \end{aligned}$$

Hence, $e \in L_2$, $\theta - \hat{\theta} \in L_\infty$ and if $y(t), \varphi(t) \in L_\infty$, then $e \in L_\infty$

B.1.2 With Forgetting Factor

For the system described by

$$y(t) = \varphi^T(t) \theta$$

if $\varphi(t)$ is persistently exciting, thus there exist some positive and finite constants σ , γ_1 and γ_2 such that

$$\gamma_1 I \leq \int_t^{t+\sigma} \varphi(\tau) \varphi^T(\tau) d\tau \leq \gamma_2 I$$

for all t , then the following RLS with forgetting factor with $\alpha_1 \geq 0$, $\alpha_2 > 0$ will produce $e \in L_2 \cap L_\infty$, $\theta - \hat{\theta} \in L_2 \cap L_\infty$ and $\|P(t)\| \in L_\infty$

$$\begin{aligned}
 \frac{d\hat{\theta}}{dt} &= \alpha_2 P(t) \varphi(t) e(t) \\
 e(t) &= y(t) - \varphi^T(t) \hat{\theta}(t) \\
 \frac{dP(t)}{dt} &= \alpha_1 P(t) - \alpha_2 P(t) \varphi(t) \varphi^T(t) P(t)
 \end{aligned}$$

B.2 Discrete-Time Recursive Least Squares

If the measurements are only available at discrete points in time, one should use the following estimator which is the well known RLS with forgetting factor $0 < \lambda \leq 1$ [5, 59].

$$y(t) = \varphi^T(t)\theta$$

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)(y(t) - \varphi^T(t)\hat{\theta}(t-1))$$

$$K(t) = P(t)\varphi(t)$$

$$P(t) = \frac{P(t-1) - P(t-1)\varphi(t)(\lambda I + \varphi^T(t)P(t-1)\varphi(t))^{-1}\varphi^T(t)P(t-1)}{\lambda}$$

Asymptotic properties of this scheme are similar to the continuous-time RLS and can be found in [5, 59]. Numerical integration of the results of appendix B.1 leads to an unstable scheme for large $P(t)$ and should not be used.

Appendix C

Observability and Controllability

In this appendix the concepts of observability and controllability of linear systems are defined. These are presented in the form of definitions and theorems that can be found in [15, 73] along with their proofs.

C.1 Continuous Time Systems

Let us consider the linear time varying continuous time system described by:

$$\dot{x}(t) = F(t)x(t) + G(t)u(t) \quad (\text{C.1})$$

with the transition matrix:

$$\frac{d\Phi(t, t_o)}{dt} = F(t)\Phi(t, t_o) \quad (\text{C.2})$$

$$\Phi(t_o, t_o) = I \quad (\text{C.3})$$

and the measurements:

$$y(t) = H(t)x(t) \quad (\text{C.4})$$

C.1.1 Observability

Definition C.1 *The system $(F(t), H(t))$ is completely observable if and only if every present state $x(t_f)$ can be determined exactly when $F(t)$, $H(t)$ and $y(t)$ for*

$t_o \leq t \leq t_f$ are known for some $t_o(t_f) < t_f$.

Theorem C.1 *The system $(F(t), H(t))$ is completely observable if and only if for every t_f there exist a $t_o(t_f) < t_f$ such that*

$$M(t_o, t_f) = \int_{t_o}^{t_f} \Phi^T(t, t_o) H^T(t) H(t) \Phi(t, t_o) dt$$

is positive definite.

Definition C.2 *If Theorem C.1 is true for every t_o and every $t_f > t_o$, the system is totally observable.*

Theorem C.2 *The system $(F(t), H(t))$ is totally observable if and only if for every t the matrix:*

$$\Gamma = \begin{bmatrix} \Gamma_1 & \Gamma_2 & \cdots & \Gamma_n \end{bmatrix}$$

is of rank n ($x(t)$ is a vector of n components) where:

$$\begin{aligned} \Gamma_1 &= H^T(t) \\ \Gamma_k &= \frac{d\Gamma_{k-1}}{dt} + F^T(t)\Gamma_{k-1} \end{aligned}$$

Definition C.3 *The system $(F(t), H(t))$ is uniformly completely R observable if and only if there exist γ, δ, τ so that for every t_1 , $W_R(t_1 - \tau, t_1)$ is positive definite and*

$$0 < \gamma I \leq W_R(t_1 - \tau, t_1) \leq \delta I$$

with

$$W_R(t_o, t_f) = \int_{t_o}^{t_f} \Phi^T(t, t_o) H^T(t) R^{-1}(t) H(t) \Phi(t, t_o) dt$$

where $R(t)$ is a positive definite matrix.

C.1.2 Controllability

Definition C.4 *If the system $(F(t), G(t))$ with $x(t_0) = c$ admits a solution such that $x(t_f) = 0$ for some finite time $t_f > t_0$ and for some measurable $u(t)$ for each $c \in R^n$ then $(F(t), G(t))$ is called completely controllable.*

Theorem C.3 *The system $(F(t), G(t))$ is completely controllable if and only if for every t_0 there exist a $t_f > t_0$ such that*

$$C(t_0, t_f) = \int_{t_0}^{t_f} \Phi(t_f, t) G(t) G^T(t) \Phi^T(t_f, t) dt$$

is positive definite.

Definition C.5 *If Theorem C.3 is true for every t_0 and every $t_f > t_0$, the system is totally controllable.*

Theorem C.4 *The system $(F(t), G(t))$ is totally controllable if and only if for every t the matrix:*

$$\Gamma = \begin{bmatrix} \Gamma_1 & \Gamma_2 & \cdots & \Gamma_n \end{bmatrix}$$

is of rank n ($x(t)$ is a vector of n components) where:

$$\begin{aligned} \Gamma_1 &= G(t) \\ \Gamma_k &= \frac{d\Gamma_{k-1}}{dt} - F(t)\Gamma_{k-1} \end{aligned}$$

Definition C.6 *The system $(F(t), G(t))$ is uniformly completely Q controllable if and only if there exist σ, α, β so that for every t_0 , $C_Q(t_0, t_0 + \sigma)$ is positive definite and*

$$0 < \alpha I \leq C_Q(t_0, t_0 + \sigma) \leq \beta I$$

with

$$C_Q(t_0, t_f) = \int_{t_0}^{t_f} \Phi(t_f, t) G(t) Q(t) G^T(t) \Phi^T(t_f, t) dt$$

where $Q(t)$ is a positive definite matrix.

C.2 Discrete Time Systems

Let us consider the linear time varying discrete time system described by:

$$x(k+1) = \Phi(k+1, k)x(k) + \Gamma(k)u(k) \quad (\text{C.5})$$

with the transition matrix:

$$\Phi(k_0 + k, k_0) = \Phi(k_0 + k, k_0 + k - 1) \cdots \Phi(k_0 + 1, k_0) \quad (\text{C.6})$$

and the measurements:

$$y(k) = H(k)x(k) \quad (\text{C.7})$$

C.2.1 Observability

Definition C.7 The system $(\Phi(k+1, k), H(k))$ is completely observable if and only if every present state $x(k_f)$ can be determined exactly when $\Phi(k+1, k)$, $H(k)$ and $y(k)$ for $k_o \leq k \leq k_f$ are known for some $k_o(k_f) < k_f$.

Theorem C.5 The system $(\Phi(k+1, k), H(k))$ is completely observable if and only if for every k_f there exist a $k_o(k_f) < k_f$ such that

$$M(k_o, k_f) = \sum_{k=k_o}^{k_f} \Phi^T(k, k_o) H^T(k) H(k) \Phi(k, k_o)$$

is positive definite.

Definition C.8 The system $(\Phi(k+1, k), H(k))$ is uniformly completely R observable if and only if there exist γ , δ , N so that for every k_1 , $W_R(k_1 - N, k_1)$ is positive definite and

$$0 < \gamma I \leq W_R(k_1 - N, k_1) \leq \delta I$$

with

$$W_R(k_o, k_f) = \sum_{k=k_o}^{k_f} \Phi^T(k, k_o) H^T(k) R^{-1}(k) H(k) \Phi(k, k_o)$$

where $R(k)$ is a positive definite matrix.

C.2.2 Controllability

Definition C.9 *If the system $(\Phi(k+1, k), \Gamma(k))$ with $x(k_o) = c$ admits a solution such that $x(k_f) = 0$ for some finite time $k_f > k_o$ and for some measurable $u(k)$ for each $c \in R^n$ then $(\Phi(k+1, k), \Gamma(k))$ is called completely controllable.*

Theorem C.6 *The system $(\Phi(k+1, k), \Gamma(k))$ is completely controllable if and only if for every k_o there exist a $k_f > k_o$ such that*

$$C(k_o, k_f) = \sum_{k=k_o}^{k_f} \Phi(k_f, k) \Gamma(k) \Gamma^T(k) \Phi^T(k_f, k)$$

is positive definite.

Definition C.10 *The system $(\Phi(k+1, k), \Gamma(k))$ is uniformly completely Q controllable if and only if there exist N, α, β so that for every k_o , $C_Q(k_o, k_o + N)$ is positive definite and*

$$0 < \alpha I \leq C_Q(k_o, k_o + N) \leq \beta I$$

with

$$C_Q(k_o, k_f) = \sum_{k=k_o}^{k_f} \Phi(k_f, k) \Gamma(k) Q(k) \Gamma^T(k) \Phi^T(k_f, k)$$

where $Q(k)$ is a positive definite matrix.

Appendix D

Stability Theory

In this appendix, stability definitions and theorems are stated without proofs and are taken directly from [22, 46].

Let us consider the free dynamical system governed by the differential equation:

$$\dot{x} = f(x, t)$$

We shall denote the solution (or motion) of this system as

$$\phi(t, x_0, t_0)$$

where $x = x_0$ at t_0 and $t \geq 0$.

Definition D.1 *A state x_e of a free dynamical system is an equilibrium state if*

$$f(x_e, t) = 0 \text{ for all } t$$

or, equivalently,

$$\phi(t, x_e, 0) = x_e \text{ for all } t$$

Definition D.2 *An equilibrium state x_e of a free dynamical system is stable if for every real number $\epsilon > 0$ there exist a real number $\delta(\epsilon, t_0) > 0$ such that $\|x_0 - x_e\| \leq \delta$ implies*

$$\|x(t, x_0, t_0) - x_e\| \leq \epsilon \text{ for all } t \geq t_0$$

Definition D.3 *An equilibrium state x_e of a free dynamical system is asymptotically stable if*

1. *it is stable and*
2. *every motion starting sufficiently near x_e converges to x_e as $t \rightarrow \infty$. In order words, there is some real constant $r(t_0) > 0$ and to every real number $\mu > 0$ there correspond a real number $T(\mu, x_0, t_0)$ such that $\|x_0 - x_e\| \leq r(t_0)$ implies*

$$\|x(t, x_0, t_0) - x_e\| \leq \mu \text{ for all } t \geq t_0 + T$$

Definition D.4 *An equilibrium state x_e of a free dynamical system is equiasymptotically stable if*

1. *it is stable and*
2. *every motion starting sufficiently near x_e converges to x_e as $t \rightarrow \infty$ uniformly in x_0 . In order words, T in definition D.3.2 is of the form $T(\mu, r(t_0), t_0)$*

Definition D.5 *An equilibrium state x_e of a free dynamical system is asymptotically stable in the large [equiasymptotically stable in the large] if*

1. *it is stable and*
2. *every motion converges to x_e [converges to x_e uniformly in x_0 for $\|x_0\| \leq r$ where r is fixed but arbitrarily large] as $t \rightarrow \infty$.*

Definition D.6 *Uniform stability is stability such that δ does not depend on t_0 .*

Definition D.7 *Uniform asymptotic stability is equiasymptotic stability such that δ, r, T do not depend on t_0 .*

Definition D.8 *An equilibrium state x_e of a free dynamical system is uniformly asymptotically stable in the large if*

1. *it is uniformly stable,*
2. *it is uniformly bounded, i.e., given any $r > 0$ there is some $B(r)$ such that $\|x_0 - x_e\| \leq r$ implies $\|\phi(t, x_0, t_0) - x_e\| \leq B$ for all $t \geq t_0$,*
3. *every motion converges to x_e as $t \rightarrow \infty$ uniformly in t_0 and $\|x_0\| \leq r$ where r is fixed but arbitrarily large; i.e., given any $r > 0$ and $\mu > 0$ there is some $T(\mu, r)$ such that $\|x_0 - x_e\| \leq r$ implies $\|\phi(t, x_0, t_0) - x_e\| \leq \mu$ for all $t \geq t_0 + T$.*

Definition D.9 Uniform stability in the large is *uniform stability* such that

$$\lim_{\epsilon \rightarrow \infty} \delta(\epsilon) = \infty$$

Theorem D.1 (Lyapunov) Consider the continuous-time, free dynamic system

$$\dot{x} = f(x, t)$$

where $f(0, t) = 0$ for all t .

Suppose there exist a scalar function $V(x, t)$ with continuous first partial derivatives with respect to x and t such that $V(0, t) = 0$ and

1. $V(x, t)$ is positive definite; i.e., there is a continuous, non-decreasing scalar function α such that $\alpha(0) = 0$ and, for all t and all $x \neq 0$

$$0 < \alpha(\|x\|) \leq V(x, t);$$

2. There exists a continuous function γ such that $\gamma(0) = 0$ and the derivative $\dot{V}$ of V along the motion starting at t, x satisfies, for all t and all $x \neq 0$,

$$\dot{V}(x, t) = \frac{\partial V}{\partial t} + (\nabla V)^T f(x, t) \leq \gamma(\|x\|) < 0;$$

3. *There exists a continuous, nondecreasing scalar function β such that $\beta(0) = 0$ and, for all t ,*

$$V(x, t) \leq \beta(\|x\|);$$

4. *$\alpha(\|x\|) \rightarrow \infty$ with $\|x\| \rightarrow \infty$*

then the equilibrium state $x_e = 0$ is uniformly asymptotically stable in the large; $V(x, t)$ is called a Lyapunov function of the system.

Corollary D.1.1 *The following conditions are sufficient for the various weaker types of stability:*

1. *Uniform asymptotic stability: (1-3).*
2. *Equiasymptotic stability in the large: (1-2), (4).*
3. *Equiasymptotic stability: (1-2).*
4. *Uniform stability: (1), (3) and (2₁): $\dot{V}(x, t) \leq 0$ for all x, t .*
5. *Stability: (1-2₁)*

Proof (see [46]).

Theorem D.2 *Let the closed-loop transfer function $H(s)$ be exponentially stable and strictly proper; then*

1. *for any minimal representation of $H(s)$ (i.e., any minimal state representation of the form $\dot{x} = Ax + Bu$, $y = Cx$), the equilibrium point $x = 0$ is globally, exponentially, uniformly (in t) stable;*
2. *if $u \in L_1$, then $y \in L_1 \cap L_\infty$, $\dot{y} \in L_1$, y is absolutely continuous and $y(t) \rightarrow 0$ as $t \rightarrow \infty$;*

3. if $u \in L_2$, then $y \in L_2 \cap L_\infty$, $\dot{y} \in L_2$, y is continuous and $y(t) \rightarrow 0$ as $t \rightarrow \infty$;
4. if $u \in L_\infty$, then $y \in L_\infty$, $\dot{y} \in L_\infty$, y is uniformly continuous;
5. if $u \in L_\infty$ and if, as $t \rightarrow \infty$, $u(t) \rightarrow u_\infty$, a constant vector, then, in addition $y(t) \rightarrow H(0)u_\infty$ as $t \rightarrow \infty$ and the convergence is exponential;
6. if $u \in L_p$ and $1 < p < \infty$, then $y \in L_p$ and $\dot{y} \in L_p$.

Proof (see [22]).

Appendix E

Two DOF Robot Dynamics

In this appendix, the dynamic model of a planar 2 DOF serial robot is developed using Lagrange's equation. A schematic of this robot is shown in figure 3.1. We will use Lagrange's equation:

$$\tau = \frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}} \right) - \frac{\partial T}{\partial q} \quad (\text{E.1})$$

where T is the total kinetic energy (since the robot is planar the lagrangian is equal to the kinetic energy), q is the motors angular position vector (generalized coordinates) and τ is the input torques vector (generalized forces).

The kinetic energies for the respective links are

$$T_1 = \frac{1}{2}(m_1 v_1^2 + I_1 \dot{q}_1^2)$$

$$T_2 = \frac{1}{2}(m_2 v_2^2 + I_2 \dot{q}_2^2)$$

where I_1 and I_2 are the inertias of links 1 and 2 with respect to their center of mass, and v_1 and v_2 are these center of mass velocities. Link 1 and 2 position are given by

$$x_1 = c_1 \cos q_1 - o_1 \sin q_1$$

$$y_1 = c_1 \sin q_1 + o_1 \cos q_1$$

$$x_2 = l_1 \cos q_1 + c_2 \cos(q_1 + q_2) - o_2 \sin(q_1 + q_2)$$

$$y_2 = l_1 \sin q_1 + c_2 \sin(q_1 + q_2) + o_2 \cos(q_1 + q_2)$$

and their velocities are

$$\begin{aligned}
 \dot{x}_1 &= -(c_1 \sin q_1 + o_1 \cos q_1) \dot{q}_1 \\
 \dot{y}_1 &= (c_1 \cos q_1 - o_1 \sin q_1) \dot{q}_1 \\
 v_1^2 &= (c_1^2 + o_1^2) \dot{q}_1^2 \\
 \dot{x}_2 &= -l_1 \sin(q_1) \dot{q}_1 - [c_2 \sin(q_1 + q_2) + o_2 \cos(q_1 + q_2)](\dot{q}_1 + \dot{q}_2) \\
 \dot{y}_2 &= l_1 \cos q_1 \dot{q}_1 + [c_2 \cos(q_1 + q_2) - o_2 \sin(q_1 + q_2)](\dot{q}_1 + \dot{q}_2) \\
 v_2^2 &= l_1^2 \dot{q}_1^2 + (c_2^2 + o_2^2)(\dot{q}_1 + \dot{q}_2)^2 + 2l_1(c_2 \cos q_2 - o_2 \sin q_2) \dot{q}_1(\dot{q}_1 + \dot{q}_2)
 \end{aligned}$$

The respective kinetic energies of both links are

$$\begin{aligned}
 T_1 &= \frac{1}{2} [m_1(c_1^2 + o_1^2) + I_1] \dot{q}_1^2 \\
 T_2 &= \frac{1}{2} m_2 l_1^2 \dot{q}_1^2 + \frac{1}{2} [m_2(c_2^2 + o_2^2) + I_2] (\dot{q}_1 + \dot{q}_2)^2 \\
 &\quad + m_2 l_1 c_2 \cos(q_2) \dot{q}_1 (\dot{q}_1 + \dot{q}_2) - m_2 l_1 o_2 \sin(q_2) \dot{q}_1 (\dot{q}_1 + \dot{q}_2)
 \end{aligned}$$

Defining the parameters:

$$\begin{aligned}
 p_1 &= m_1(c_1^2 + o_1^2) + I_1 + m_2 l_1^2 \\
 p_2 &= m_2(c_2^2 + o_2^2) + I_2 \\
 p_3 &= m_2 l_1 c_2 \\
 p_4 &= m_2 l_1 o_2
 \end{aligned}$$

results in the following total kinetic energy:

$$\begin{aligned}
 T &= T_1 + T_2 \\
 &= \frac{1}{2} p_1 \dot{q}_1^2 + \frac{1}{2} p_2 (\dot{q}_1 + \dot{q}_2)^2 + p_3 \cos(q_2) \dot{q}_1 (\dot{q}_1 + \dot{q}_2) - p_4 \sin(q_2) \dot{q}_1 (\dot{q}_1 + \dot{q}_2) \quad (E.2)
 \end{aligned}$$

Applying Lagrange's equation (E.1) to equation (E.2) yields

$$\begin{aligned}
 \tau_1 &= (p_1 + p_2 + 2p_3 \cos q_2 - 2p_4 \sin q_2) \ddot{q}_1 + (p_2 + p_3 \cos q_2 - p_4 \sin q_2) \ddot{q}_2 \\
 &\quad - (p_3 \sin q_2 + p_4 \cos q_2) \dot{q}_2 (2\dot{q}_1 + \dot{q}_2) \\
 \tau_2 &= (p_2 + p_3 \cos q_2 - p_4 \sin q_2) \ddot{q}_1 + p_2 \ddot{q}_2 + (p_3 \sin q_2 + p_4 \cos q_2) \dot{q}_1^2
 \end{aligned}$$

In matrix form:

$$\begin{aligned} D(q) &= \begin{bmatrix} p_1 + p_2 + 2p_3 \cos q_2 - 2p_4 \sin q_2 & p_2 + p_3 \cos q_2 - p_4 \sin q_2 \\ p_2 + p_3 \cos q_2 - p_4 \sin q_2 & p_2 \end{bmatrix} \\ h(q, \dot{q}) &= \begin{bmatrix} -(p_3 \sin q_2 + p_4 \cos q_2) \dot{q}_2 (2\dot{q}_1 + \dot{q}_2) \\ (p_3 \sin q_2 + p_4 \cos q_2) \dot{q}_1^2 \end{bmatrix} \\ g(q) &= 0 \end{aligned}$$

Also, we know that

$$\begin{aligned} h(q, \dot{q}) &= C(q, \dot{q}) \dot{q} \\ C(q, \dot{q}) &= \begin{bmatrix} -(p_3 \sin q_2 + p_4 \cos q_2) \dot{q}_2 & -(p_3 \sin q_2 + p_4 \cos q_2) (\dot{q}_1 + \dot{q}_2) \\ (p_3 \sin q_2 + p_4 \cos q_2) \dot{q}_1 & 0 \end{bmatrix} \end{aligned}$$

If there is gravity acting in the negative y axis direction $g(q)$ is replaced by

$$g(q) = \begin{bmatrix} p_5 \cos q_1 - p_6 \sin q_1 + p_7 \cos(q_1 + q_2) - p_8 \sin(q_1 + q_2) \\ p_7 \cos(q_1 + q_2) - p_8 \sin(q_1 + q_2) \end{bmatrix}$$

where the new parameters are

$$p_5 = (m_1 c_1 + m_2 l_1)g$$

$$p_6 = m_1 o_1 g$$

$$p_7 = m_2 c_2 g$$

$$p_8 = m_2 o_2 g$$

Appendix F

Experimental Robot Dynamics

In this appendix, the dynamic model of the experimental direct drive robot is developed using Lagrange's equation. A schematic of the experimental robot is shown in figure 5.1. This is a planar robot with a closed kinematic chain. Results of appendix A cannot be applied directly to closed kinematic chain, thus we will use Lagrange's equation:

$$\tau = \frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}} \right) - \frac{\partial T}{\partial q} \quad (\text{F.1})$$

where T is the total kinetic energy (since the robot is planar the lagrangian is equal to the kinetic energy), q is the motors angular position vector (generalized coordinates) and τ is the input torques vector (generalized forces).

The kinetic energies for the respective links are

$$\begin{aligned} T_0 &= \frac{1}{2} I_0 \dot{q}_1^2 \\ T_1 &= \frac{1}{2} I_1 \dot{q}_2^2 \\ T_2 &= \frac{1}{2} (m_2 v_2^2 + I_2 \dot{q}_1^2) \\ T_3 &= \frac{1}{2} (m_3 v_3^2 + I_3 \dot{q}_2^2) \end{aligned}$$

where I_0 and I_1 are links 0 and 1 inertias with respect to the motors axis, I_2 and I_3 are the inertias of links 2 and 3 with respect to their center of mass, and v_2 and v_3

are these center of mass velocities. For link 2, we have

$$\begin{aligned}
 x_2 &= l_1 \cos q_2 + l_4 \cos q_1 \\
 y_2 &= l_1 \sin q_2 + l_4 \sin q_1 \\
 \dot{x}_2 &= -l_1 \sin q_2 \dot{q}_2 - l_4 \sin q_1 \dot{q}_1 \\
 \dot{y}_2 &= l_1 \cos q_2 \dot{q}_2 + l_4 \cos q_1 \dot{q}_1 \\
 v_2^2 &= \dot{x}_2^2 + \dot{y}_2^2 \\
 &= l_1^2 \dot{q}_2^2 + l_4^2 \dot{q}_1^2 + 2l_1 l_4 \dot{q}_1 \dot{q}_2 \cos(q_1 - q_2)
 \end{aligned}$$

For the link 3, we have

$$v_3^2 = l_2^2 \dot{q}_1^2 + l_3^2 \dot{q}_2^2 + 2l_2 l_3 \dot{q}_1 \dot{q}_2 \cos(q_1 - q_2)$$

The total kinetic energy is

$$\begin{aligned}
 T &= T_0 + T_1 + T_2 + T_3 \\
 &= \frac{1}{2}(I_0 + I_2 + m_2 l_4^2 + m_3 l_2^2) \dot{q}_1^2 \\
 &\quad + \frac{1}{2}(I_1 + I_3 + m_2 l_1^2 + m_3 l_3^2) \dot{q}_2^2 \\
 &\quad + (m_2 l_1 l_4 + m_3 l_2 l_3) \dot{q}_1 \dot{q}_2 \cos(q_1 - q_2)
 \end{aligned}$$

Defining the parameters:

$$\begin{aligned}
 p_1 &= I_0 + I_2 + m_2 l_4^2 + m_3 l_2^2 \\
 p_2 &= I_1 + I_3 + m_2 l_1^2 + m_3 l_3^2 \\
 p_3 &= m_2 l_1 l_4 + m_3 l_2 l_3
 \end{aligned}$$

results in the followings:

$$T = \frac{1}{2} p_1 \dot{q}_1^2 + \frac{1}{2} p_2 \dot{q}_2^2 + p_3 \dot{q}_1 \dot{q}_2 \cos(q_1 - q_2) \quad (\text{F.2})$$

Applying Lagrange's equation (F.1) to equation (F.2) yields

$$\begin{aligned}
 \tau_1 &= p_1 \ddot{q}_1 + p_3 \cos(q_1 - q_2) \ddot{q}_2 + p_3 \sin(q_1 - q_2) \dot{q}_2^2 \\
 \tau_2 &= p_2 \ddot{q}_2 + p_3 \cos(q_1 - q_2) \ddot{q}_1 - p_3 \sin(q_1 - q_2) \dot{q}_1^2
 \end{aligned}$$

In matrix form

$$\begin{aligned}
 D(q) &= \begin{bmatrix} p_1 & p_3 \cos(q_1 - q_2) \\ p_3 \cos(q_1 - q_2) & p_2 \end{bmatrix} \\
 h(q, \dot{q}) &= p_3 \sin(q_1 - q_2) \begin{bmatrix} \dot{q}_2^2 \\ -\dot{q}_1^2 \end{bmatrix} \\
 g(q) &= 0
 \end{aligned}$$

References

- [1] C. H. An, C. G. Atkeson, and J. M. Hollerbach, *Model-Based Control of a Robot Manipulator*. Cambridge, Massachusetts: MIT Press, 1988.
- [2] B. D. O. Anderson, "Stability properties of Kalman-Bucy filters," *Journal of the Franklin Institute*, vol. 291, pp. 137–144, Feb. 1971.
- [3] B. D. O. Anderson, R. R. Bitmead, C. R. Johnson, Jr., P. V. Kokotovic, R. L. Kosut, I. M. Y. Mareels, L. Praly, and B. D. Riedle, *Stability of Adaptive Systems: Passivity and Averaging Analysis*. Cambridge, Massachusetts: MIT Press, 1986.
- [4] H. Asada and K. Youcef-Toumi, *Direct-Drive Robots*. Cambridge, Massachusetts: MIT Press, 1987.
- [5] K. J. Astrom and B. Wittenmark, *Adaptive Control*. Reading, Massachusetts: Addison-Wesley, 1989.
- [6] M. Athans and P. L. Falb, *Optimal Control: An Introduction to the Theory and Its Applications*. New York: McGraw-Hill, 1966.
- [7] C. G. Atkeson, C. H. An, and J. M. Hollerbach, "Estimation of inertial parameters of manipulator loads and links," *The International Journal of Robotics Research*, vol. 5, pp. 101–119, Fall 1986.

- [8] C. G. Atkeson, C. H. An, and J. M. Hollerbach, "Estimation of inertial parameters of manipulator loads and links," in *Proceedings of the Third International Symposium on Robotic Research*, 1985.
- [9] G. J. Bierman, "Measurement updating using the $U - D$ factorization," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 337-346, 1975.
- [10] J. E. Bobrow, S. Dubowsky, and J. S. Gibson, "Time-optimal control of robotic manipulators along specified paths," *The International Journal of Robotics Research*, vol. 4, pp. 3-17, Fall 1985.
- [11] A. E. Bryson, Jr. and Y. Ho, *Applied Optimal Control*. New-York: Hemisphere Publishing Corp., 1975.
- [12] R. S. Bucy, "Global theory of the Riccati equation," *J. Computer and System Sciences*, vol. 1, no. 4, pp. 349-361, 1967.
- [13] R. S. Bucy, "A priori bounds for the Riccati equation," in *Sixth Berkeley Symposium on Mathematical Statistics and Probability*, pp. 645-656, 1972.
- [14] R. S. Bucy, "The Riccati equation and its bounds," *J. Computer and System Sciences*, vol. 6, no. 4, pp. 343-353, 1972.
- [15] R. S. Bucy and P. D. Joseph, *Filtering for Stochastic Processes with Applications to Guidance*. New York: Chelsea, second ed., 1987.
- [16] Y. Chen and A. A. Desrochers, "A proof of the structure of the time-optimal control law of robotic manipulators using a hamiltonian formulation," *IEEE Jour. of Robotics and Automation*, vol. RA-6, pp. 388-393, June 1990.
- [17] Y. Chen and A. A. Desrochers, "Time-optimal control of two-degree of freedom robots arms," in *Proceedings of the IEEE Conference on Robotics and Automation*, pp. 1210-1215, Apr. 1988.

- [18] J. J. Craig, *Adaptive Control of Mechanical Manipulators*. Reading, Mass.: Addison-Wesley, 1988.
- [19] J. J. Craig, P. Hsu, and S. S. Sastry, "Adaptive control of mechanical manipulators," *The International Journal of Robotics Research*, vol. 6, pp. 16-28, Summer 1987.
- [20] J. J. Craig, P. Hsu, and S. S. Sastry, "Adaptive control of mechanical manipulators," in *Proceedings of the IEEE Conference on Robotics and Automation*, pp. 190-195, 1986.
- [21] J. Denavit and R. S. Hartenberg, "A kinematic notation for lower pair mechanisms based on matrices," *ASME Jour. of Applied Mechanics*, vol. , pp. 215-221, June 1955.
- [22] C. Desoer and M. Vidyasagar, *Feedback Systems: Input-Output Properties*. New York: Academic Press, 1975.
- [23] J. J. Deyst and C. F. Price, "Conditions for asymptotic stability of the discrete minimum-variance linear estimator," *IEEE Trans. on Automatic Control*, vol. AC-13, pp. 702-705, Dec. 1968.
- [24] S. Dubowsky and T. D. Desforges, "The application of model-referenced adaptive control to robotic manipulators," *ASME Jour. of Dynamic Systems, Measurement, and Control*, vol. 101, pp. 193-200, Sep. 1979.
- [25] R. J. Fitzgerald, "Divergence of the Kalman filter," *IEEE Trans. on Automatic Control*, vol. AC-16, pp. 736-747, Dec. 1971.
- [26] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*. New York: McGraw-Hill, 1987.

- [27] M. Gautier and W. Khalil, "Direct calculation of minimum set of inertial parameters of serial robots," *IEEE Jour. of Robotics and Automation*, vol. RA-6, pp. 368-373, June 1990.
- [28] H. P. Geering, L. Guzzella, S. A. R. Hepner, and C. H. Onder, "Time-optimal motions of robots in assembly tasks," *IEEE Trans. on Automatic Control*, vol. AC-31, pp. 512-517, June 1986.
- [29] R. Gourdeau and H. M. Schwartz, "Adaptive control of robotic manipulators: experimental results," in *Proceedings of the IEEE Conference on Robotics and Automation*, 1991.
- [30] R. Gourdeau and H. M. Schwartz, "Adaptive control of robotic manipulators using an extended Kalman filter," in *Proceedings of the ASME Winter Annual Meeting*, 1990.
- [31] R. Gourdeau and H. M. Schwartz, "Adaptive control of robotic manipulators using an extended Kalman filter," submitted to the *ASME Jour. of Dynamic Systems, Measurement, and Control*.
- [32] R. Gourdeau and H. M. Schwartz, "Optimal control of a robot manipulator using a weighted time-energy cost function," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 1628-1631, 1989.
- [33] W. Hahn, *Stability of Motion*. New York: Springer-Verlag, 1967.
- [34] H. Hermes and J. P. LaSalle, *Functional Analysis and Time-Optimal Control*. New York: Academic, 1970.
- [35] J. M. Hollerbach, "A recursive lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity," *IEEE Trans. on Systems, Man and Cybernetics*, vol. SMC-10, pp. 730-736, Nov. 1980.

of/de

3

- [36] T. C. Hsia, "Adaptive control of robot manipulators — a review," in *Proceedings of the IEEE Conference on Robotics and Automation*, pp. 183–189, 1986.
- [37] P. Hsu, M. Bodson, S. Sastry, and B. Paden, "Adaptive identification and control for manipulators without using joint accelerations," in *Proceedings of the IEEE Conference on Robotics and Automation*, pp. 1210–1215, 1987.
- [38] Jazwinski, "Adaptive filtering," *Automatica*, vol. 5, pp. 475–485, July 1969.
- [39] Jazwinski, *Stochastic Processes and Filtering Theory*. New York: Academic Press, 1970.
- [40] R. Johansson, "Adaptive control of manipulator motion," *IEEE Jour. of Robotics and Automation*, vol. RA-6, pp. 483–490, Aug. 1990.
- [41] M. E. Kahn and B. Roth, "The near-minimum-time control of open-loop articulated kinematic chains," *ASME Jour. of Dynamic Systems, Measurement, and Control*, vol. 93, pp. 164–172, Sep. 1971.
- [42] T. Kailath, "An innovations approach to least-squares estimation part I: linear filtering in additive white noise," *IEEE Trans. on Automatic Control*, vol. AC-13, pp. 646–655, Dec. 1968.
- [43] R. E. Kalman, "Contributions to the theory of optimal control," *Boletin Sociedad Matematica Mexicana*, vol. 5, pp. 102–119, 1960.
- [44] R. E. Kalman, "A new approach to linear filtering and prediction problems," *ASME Jour. of Basic Engineering*, vol. 82, pp. 35–45, March 1960.
- [45] R. E. Kalman, "New methods in Weiner filtering," in *Proceedings of First Symposium on Engineering Applications of Random Function Theory and Probability*, (J. L. Bogdanoff and F. Fozin, eds.), (New York), pp. 270–388, Wiley, 1963.

- [46] R. E. Kalman and J. E. Bertram, "Control system analysis and design via the "second method" of Lyapunov: I continuous-time systems," *ASME Jour. of Basic Engineering*, vol. 82, pp. 371-393, June 1960.
- [47] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *ASME Jour. of Basic Engineering*, vol. 83, pp. 95-108, March 1961.
- [48] R. Kelly and R. Carelli, "Unified approach to adaptive control of robotic manipulators," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 1598-1603, 1988.
- [49] W. Khalil, M. Gauthier, and J. F. Kleinfinger, "Automatic generation of identification models of robots," *International Journal of Robotics and Automation*, vol. 1, no. 1, pp. 2-6, 1986.
- [50] P. K. Khosla and T. Kanada, "Parameter identification of robot dynamics," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 1754-1760, 1985.
- [51] B. K. Kim and K. G. Shin, "Suboptimal control of industrial manipulators with a weighted minimum time-fuel criterion," *IEEE Trans. on Automatic Control*, vol. AC-30, pp. 1-10, Jan. 1985.
- [52] A. J. Koivo and T. H. Guo, "Adaptive linear control for robotic manipulators," *IEEE Trans. on Automatic Control*, vol. AC-28, pp. 162-171, Feb. 1983.
- [53] R. E. Kopp and R. J. Orford, "Linear regression applied to system identification for adaptive control systems," *AIAA Journal*, vol. 1, pp. 2300-2306, Oct. 1963.
- [54] C. S. G. Lee and M. J. Chung, "An adaptive control strategy for mechanical manipulators," *IEEE Trans. on Automatic Control*, vol. AC-29, pp. 837-840, Sep. 1984.

- [55] C. S. G. Lee and B. H. Lee, "Resolved motion adaptive control for mechanical manipulators," *ASME Jour. of Dynamic Systems, Measurement, and Control*, vol. 106, pp. 134–142, June 1984.
- [56] W. Li and J. E. Slotine, "Indirect adaptive robot control," in *Proceedings of the IEEE Conference on Robotics and Automation*, pp. 704–709, 1988.
- [57] W. Li and J. E. Slotine, "Parameter estimation strategies for robotic manipulators," in *Proceedings of the ASME Winter Annual Meeting*, 1987.
- [58] K. Y. Lim and M. Eslami, "Adaptive controller designs for robot manipulator systems using Lyapunov direct method," *IEEE Trans. on Automatic Control*, vol. AC-30, pp. 1229–1233, Dec. 1985.
- [59] L. Ljung and T. Soderstrom, *Theory and Practice of Recursive Identification*. Cambridge, Massachusetts: MIT Press, 1983.
- [60] R. K. Mehra, "On the identification of variances and adaptive Kalman filtering," *IEEE Trans. on Automatic Control*, vol. AC-15, pp. 175–184, Apr. 1970.
- [61] R. H. Middleton, "Adaptive control for robot manipulators using discrete time identification," *IEEE Trans. on Automatic Control*, vol. AC-35, pp. 633–637, May 1990.
- [62] R. H. Middleton, "Hybrid adaptive control for robot manipulators," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 1592–1597, 1988.
- [63] R. H. Middleton and G. C. Goodwin, "Adaptive computed torque control for rigid link manipulators," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 68–73, 1986.
- [64] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems*. Englewood Cliffs, New Jersey: Prentice Hall, 1989.

- [65] K. Ogata, *Discrete-Time Control Systems*. Englewood Cliffs, New Jersey: Prentice Hall, 1987.
- [66] R. Ortega and M. W. Spong, "Adaptive motion control of rigid robots: a tutorial," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 1575–1584, 1988.
- [67] J. Picard, "Efficiency of the extended Kalman filter for non linear systems with small noise," Tech. Rep. 1068, INRIA Sophia Antipolis, Aug. 1989.
- [68] J. S. Reed and P. A. Ioannou, "Instability analysis and robust adaptive control of robotic manipulators," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 1607–1612, 1988.
- [69] N. Sadegh and R. Horowitz, "An exponentially stable adaptive control law for robot manipulators," *IEEE Jour. of Robotics and Automation*, vol. RA-6, pp. 491–496, Aug. 1990.
- [70] N. Sadegh and R. Horowitz, "Stability analysis of an adaptive controller for robotic manipulators," in *Proceedings of the IEEE Conference on Robotics and Automation*, pp. 1223–1229, 1987.
- [71] A. P. Sage and J. L. Melsa, *Estimation Theory with Applications to Communication and Control*. New York: McGraw-Hill, 1971.
- [72] A. P. Sage and J. L. Melsa, *System Identification*. New York: Academic Press, 1971.
- [73] A. P. Sage and C. C. White, III, *Optimum Systems Control*. Englewood Cliffs, New Jersey: Prentice-Hall, 2nd ed., 1977.
- [74] G. Sahar and J. M. Hollerbach, "Planning of minimum-time trajectories for robot arms," *The International Journal of Robotics Research*, vol. 5, pp. 90–100, Fall 1986.

- [75] H. M. Schwartz, G. Warshaw, and T. Janabi, "Issues in robot adaptive control," in *Proceedings of the American Control Conference*, 1990.
- [76] K. G. Shin and N. D. McKay, "A dynamic programming approach to trajectory planning of robotic manipulators," *IEEE Trans. on Automatic Control*, vol. AC-31, pp. 491-500, June 1986.
- [77] K. G. Shin and N. D. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Trans. on Automatic Control*, vol. AC-30, pp. 531-541, June 1985.
- [78] S. Singh and M. C. Leu, "Optimal trajectory generation for robotic manipulators using dynamic programming," *ASME Jour. of Dynamic Systems, Measurement, and Control*, vol. 109, pp. 88-96, June 1987.
- [79] S. N. Singh, "Adaptive model following control of nonlinear robotic systems," *IEEE Trans. on Automatic Control*, vol. AC-30, pp. 1099-1100, Nov. 1985.
- [80] J. E. Slotine and W. Li, "Adaptive manipulator control: a case study," *IEEE Trans. on Automatic Control*, vol. AC-33, pp. 995-1003, Nov. 1988.
- [81] J. E. Slotine and W. Li, "On the adaptive control of robot manipulators," *The International Journal of Robotics Research*, vol. 6, pp. 49-59, Fall 1987.
- [82] M. W. Spong, R. Ortega, and R. Kelly, "Comments on "Adaptive manipulator control: a case study"," *IEEE Trans. on Automatic Control*, vol. AC-35, pp. 761-762, June 1990.
- [83] C. L. Thornton and G. J. Bierman, "Gram-Schmidt algorithms for covariance propagation," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 489-498, 1975.
- [84] J. J. Uicker, "Dynamic force analysis of spatial linkages," *ASME Jour. of Applied Mechanics*, vol. 34, pp. 418-424, June 1967.

- [85] M. Vukobratovic and D. Stokic, "Towards nonadaptive and adaptive control of manipulation robots," *IEEE Trans. on Automatic Control*, vol. AC-29, pp. 841-844, Sep. 1984.
- [86] M. W. Walker, "Adaptive control of manipulators containing closed kinematic loops," *IEEE Jour. of Robotics and Automation*, vol. RA-6, pp. 10-19, Feb. 1990.
- [87] A. Weinreb and A. E. Bryson, "Optimal control of systems with hard control bounds," *IEEE Trans. on Automatic Control*, vol. AC-30, pp. 1135-1138, Nov. 1985.
- [88] J. Wen and A. A. Desrochers, "An algorithm for obtaining bang-bang control laws," *ASME Jour. of Dynamic Systems, Measurement, and Control*, vol. 109, pp. 171-175, June 1987.
- [89] Q. Zhang, A. J. Koivo, and T. Guo, "Industrial manipulator control using Kalman filter and adaptive controller," *International Journal of Robotics and Automation*, vol. 2, no. 1, pp. 32-39, 1987.

END

06-03-92

FIN