

University of Windsor

## Scholarship at UWindor

---

Electronic Theses and Dissertations

Theses, Dissertations, and Major Papers

---

11-6-2015

### Model based control of a reconfigurable robot

Riyadh Al Saidi  
*University of Windsor*

Follow this and additional works at: <https://scholar.uwindsor.ca/etd>

---

#### Recommended Citation

Al Saidi, Riyadh, "Model based control of a reconfigurable robot" (2015). *Electronic Theses and Dissertations*. 5527.

<https://scholar.uwindsor.ca/etd/5527>

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email ([scholarship@uwindsor.ca](mailto:scholarship@uwindsor.ca)) or by telephone at 519-253-3000ext. 3208.

MODEL BASED CONTROL OF A  
RECONFIGURABLE ROBOT

by

Riyadh Al Saidi

A Dissertation

Submitted to the Faculty of Graduate Studies  
through Mechanical, Automotive & Materials Engineering  
in Partial Fulfillment of the Requirements for  
the Degree of Doctor of Philosophy at the  
University of Windsor

Windsor, Ontario, Canada

2015

© 2015 R. Al Saidi

MODEL BASED CONTROL OF A RECONFIGURABLE ROBOT

by

R. Al Saidi

APPROVED BY:

---

L. Rolland, External Examiner

Department of Mechanical Engineering, Memorial University

---

X. Chen, External Reader

Electrical & Computer Engineering

---

N. Zamani, Department Reader

Mechanical, Automotive & Materials Engineering

---

J. Johrendt, Department Reader

Mechanical, Automotive & Materials Engineering

---

B. Minaker, Supervisor

Mechanical, Automotive & Materials Engineering

July 8<sup>th</sup>, 2015

## **Author's Declaration of Originality**

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify, to the best of my knowledge, that my thesis does not infringe upon anyone's copyright or violate any proprietary rights and that any ideas, techniques, quotations, or any other materials from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted materials that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission of the copyright owner(s) to include such materials in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by committee and the Graduate Studies Office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

## Abstract

Robots with predefined kinematic structures are successfully applied to accomplish tasks between the robot and the environment. For more sophisticated and future applications, it is necessary to extend the capabilities of robots, and employ them in more complex applications, which generally require accurate and more changeable structural properties during the interaction with the environment. The central focus of this research was to propose a robot with new properties to address the reconfigurability problem, including its feasible solutions using model based control strategies. First, these reconfigurable robots have to combine as many properties of different open kinematic structures as possible and can be used for a variety of applications. The kinematic design parameters, i.e., their Denavit-Hartenberg (D-H) parameters, were modeled to be variable to satisfy any configuration required to meet a specific task. By varying the joint twist angle parameter (a configuration parameter), the presented model is reconfigurable to any desired open kinematic structure, such as Fanuc, ABB and SCARA robots. The joint angle and the offset distance of the D-H parameters are also modeled as variable parameters (a reconfigurable joint). The resulting reconfigurable robot hence encompasses different kinematic structures and has a reconfigurable joint to accommodate any required application in medical technology, space exploration and future manufacturing systems, for example. Second, a methodology was developed to automate model generation for  $n$ -DOF Global Kinematic Model ( $n$ -GKM). Then, advanced model based control strategies were employed to increase performance as compared to less structured approaches. An algorithm was developed to select a relevant kinematic structural robot configuration for any predefined geometric task. The main contribution of this research is that it combines a kinematic structural design with control design methods to optimize robot capability and performance. This combination has been established by developing an algorithm to select the optimal kinematic structure and the most applicable control approach to perform a predefined geometric task with high tracking performance.

## Acknowledgements

I would like to express my immense gratitude to Dr. Bruce Minaker for his supervision and encouragement. His enthusiasm and brilliance greatly motivated me throughout this pursuit. His support and compassion have helped me overcome the challenging obstacles along the way.

I would also like to thank the faculty and staff in the department of Mechanical Engineering for providing interesting courses and a supportive learning environment.

I am very grateful to the the members of my defense committee. In particular, I would like to thank Dr. X. Chen for agreeing to evaluate my work as the external reader, and Drs. N. Zamani and J. Johrendt. The committee members provided invaluable feedback on earlier drafts of this work. I would also like to thank Dr. Luc Rolland for agreeing to serve as external examiner.

Ms. Angela Haskell has been an incredible source of support and I am deeply indebted to her.

Finally, I would like to thank my ever supportive wife for standing by my side and sharing this journey with me.

## Contents

Author's Declaration of Originality	iii
Abstract	iv
Acknowledgments	v
List of Figures	x
List of Tables	xvii
Nomenclature	xviii
Chapter 1. Introduction and Preliminaries	1
1.1. Introduction to Robot Kinematics	1
1.2. Introduction to Reconfigurability Theory	2
1.3. Robot Control	5
1.4. Problem Statement	10
1.4.1. General Problem Statement	10
1.4.2. Research Approach	10
Chapter 2. Development of a Reconfigurable Robot Kinematics	12
2.1. Modeling of a Reconfigurable Joint	12
2.2. Modeling of Reconfigurable Open Kinematic Robots	15
2.2.1. Spherical Wrist	16
2.2.2. Assumption	17
2.3. Reconfigurable Jacobian Matrix	17
2.3.1. Decoupling of Singularities	19
2.4. Manipulability and Singularity	20
2.4.1. Manipulability	20
2.5. Jacobian Condition and Manipulability	24

2.5.1.	Joint-Space and Cartesian Trajectories	26
2.5.2.	Motion Through a Singularity	27
Chapter 3.	Reconfigurable Robot Dynamics	31
3.1.	Reconfigurable Global Dynamic Model	31
3.2.	Forward Computation for Reconfigurable Robot	32
3.3.	Backward Computation of Forces and Moments	34
3.4.	Development of the 3-GDM Model	38
3.4.1.	Model Validation	39
3.5.	Parameter Properties of a Reconfigurable Robot	45
3.5.1.	Gravity Load Term	46
3.5.2.	Inertia Matrix Term	47
3.5.3.	Coriolis Matrix Term	49
3.5.4.	Effect of Payload Term	51
Chapter 4.	Linear Control	52
4.1.	Optimal Robust Control	52
4.2.	Mixed Sensitivity $H_\infty$ Control	54
4.3.	$H_\infty$ Control Design	56
4.4.	$\mu$ -Optimal Control	61
4.5.	$H_\infty$ Gain Scheduled Control	63
4.5.1.	Analysis of LPV Polytopic Systems and Controllers	63
4.5.2.	Synthesis of LPV Polytopic Controllers	66
4.5.3.	Analysis of LPV Systems with LFT System Description	68
4.6.	Linear Control Applications (Simulation Results)	70
4.6.1.	Application of Mixed Sensitivity $H_\infty$ Control (Simulation and Results)	70
4.6.2.	Application of $H_\infty$ Control (Simulation and Results)	72
4.6.3.	Application of $\mu$ -Synthesis Control and DK Iterations (Simulation and Results)	80
4.6.4.	Comparison of $H_\infty$ and $\mu$ -Controllers	87
4.6.5.	Order Reduction of $\mu$ Control	90
4.6.6.	Application of LPV Control (Simulation and Results)	93



Chapter 5. Nonlinear Control	97
5.1. Position Control	98
5.1.1. Link Dynamics	98
5.1.2. PD-Gravity Control	99
5.1.3. A Reconfigurable Robot with PD-Gravity Control	103
5.1.4. Three-Axis Robot Structure with PD-Gravity Control	104
5.2. Trajectory Control	108
5.2.1. Feedback Linearization Control	108
5.2.2. Input-Output Linearization	109
5.2.3. Sliding Mode Control (SMC)	112
5.2.4. Sliding Mode Control Based on Estimated Model	114
5.2.5. Simulation of 3-DOF Reconfigurable Manipulator	117
5.2.6. Sliding Mode Control Based on Bounded Model	120
5.2.7. Sliding Mode Control Based on Computed Torque Method	123
5.2.8. Adaptive Control	126
5.2.9. Derivation of Adaptive Sliding Mode Control	127
5.2.10. Simulation Results of 3-DOF Reconfigurable Robot	129
Chapter 6. Kinematic and Control Selection Algorithm	132
6.1. Configuration Phase	132
6.1.1. Internal D–H Parameters Optimization	133
6.2. Control Phase	134
6.2.1. Dynamic Parameter Properties	136
6.2.2. The Reconfigurable Control Algorithm	136
6.3. Algorithm Implementation (Simulations and Results)	141
6.3.1. Trajectory of Two Circles in Joint Space Motion	141
6.3.2. Straight Line Trajectory in Joint Space	144
6.3.3. Position (Linear) and Trajectory (Nonlinear) Control Method Selection	148
6.4. Comparison between Linear and Nonlinear Control Approaches	156
6.5. Summary	157
Chapter 7. Conclusions and Recommendations	159

CONTENTS

ix

7.1. Conclusions	159
7.2. Recommendations	162
Appendix A: Bosch Scara Model	164
Appendix B: Bosch Scara Model	166
Appendix C: Kinematic Calculations	168
References	169
Vita Auctoris	175

## List of Figures

1.1	Definition of standard Denavit and Hartenberg (D–H) parameters. Source; Manseur, [63].....	2
1.2	Set of design variables of the 3-DOF configuration modular robots, this set is a commercial product of AMTEC GmbH Company. Source; I.M. Chen, [22].....	4
1.3	Mechanical set up of a modular and reconfigurable robot (left). The ICT power cube Mechatronical component (right). Source; Strasser, [84].....	5
1.4	Robotic system with motion control system, inner and outer loop controllers.....	6
1.5	Standard robust control problem. ....	9
2.1	Kinematic structures of the ABB and Stanford robots, D–H param- eters are from sources; Dawson, [57] and Spong, [80].....	15
2.2	The spherical wrist, joint axes 4, 5 and 6. ....	16
2.3	All possible configuration models of a reconfigurable hybrid joint.....	18
2.4	Workspace of RRR Configuration with four different twist angle val- ues $\pi/16$ , $\pi/8$ , $\pi/4$ , and $\pi/2$ . ....	21
2.5	Workspace of RRT Configuration with different link lengths 0.15, 0.3 and 0.45 m. ....	22
2.6	3D profile of the manipulability index measure of RRR configuration.	23
2.7	3D profile of the manipulability index measure of RRT configuration..	24
2.8	Cartesian velocity ellipsoid of RRR configuration with different twist angle values of $(\pi/16, \pi/6, \pi/2)$ degrees.....	26
2.9	Cartesian velocity ellipsoid of RRT configuration with different pris- matic lengths of 0.1, 0.2 and 0.3 m. ....	27

2.10	Joint angles during joint-space motion (left). Cartesian coordinates of the end-effector in $x$ , $y$ and $z$ directions (right).....	28
2.11	Cartesian position locus in the $xy$ -plane (left). Euler angles roll-pitch-yaw versus time (right). .....	28
2.12	Joint angles during Cartesian motion (left). Cartesian coordinates of the end-effector in $x$ , $y$ and $z$ directions (right).....	28
2.13	Cartesian position locus in the $xy$ -plane (left). Euler angles roll-pitch-yaw versus time (right). .....	29
2.14	Joint angles follow Cartesian path through a wrist singularity (left), joint angles follows joint-space path (right). .....	29
2.15	Manipulability measure of Cartesian and joint-space paths.....	30
3.1	RR planar kinematic structure.....	40
3.2	RT planar kinematic structure.....	41
3.3	TR planar kinematic structure.....	42
3.4	TT planar kinematic structure.....	42
3.5	Scara kinematic structure. ....	44
3.6	The workspace of a predefined kinematic structure of the PUMA 560.	46
3.7	The workspace of a reconfigurable manipulator.....	47
3.8	Gravity load variation with a reconfigurable manipulator pose (shoulder gravity load).....	48
3.9	Gravity load variation with a reconfigurable manipulator pose (Elbow gravity load).....	48
3.10	Variation of inertia matrix elements $M_{11}$ with configuration. ....	49
3.11	Variation of inertia matrix elements $M_{12}$ with configuration. ....	49
3.12	Variation of inertia matrix elements $M_{22}$ with configuration. ....	50
3.13	Joint 2 inertia as a function of joint 3 angle, $M_{22}(q_3)$ .....	50
4.1	Closed loop configuration with mixed sensitivity consideration.....	53
4.2	General $H_\infty$ control configuration. ....	55
4.3	S/SK mixed sensitivity optimization in regulation form (left), S/SK mixed sensitivity minimization in tracking form (right). ....	57
4.4	Uncertain system representation. ....	57

4.5	Standard $H_\infty/\mu$ control structure (left), A control structure for closed system analysis (right).	58
4.6	A control configuration with extended uncertainty structure.	61
4.7	$\mu$ -Synthesis Control Procedure.	62
4.8	The structure of the LPV system and control using Linear Fractional Transformation (LFT) description system.	68
4.9	Model of first two links and servo motors of the Bosch Scara robot.	71
4.10	Schematic top view of the Bosch Scara robot in null position.	71
4.11	Inverse of performance weight (dashed line) and the resulting sensitivity function (solid line) for two $H_\infty$ designs (1 and 2) for disturbance rejection.	73
4.12	Closed loop step responses for two alternative designs (1 and 2) for disturbance rejection problem.	73
4.13	Derived block diagram of the first link of the Bosch Scara robot.	74
4.14	LFT representation of the perturbed Bosch Scara model	75
4.15	Perturbed (set of family) open loop systems.	75
4.16	Singular value of the closed loop with $H_\infty$ controller.	76
4.17	$H_\infty$ control structure.	77
4.18	Sensitivity and performance weighting function with $H_\infty$ controller.	78
4.19	Robust stability analysis with $H_\infty$ control.	78
4.20	Nominal and robust performance with $H_\infty$ control.	79
4.21	Perturbed (set of family) of closed loop systems with different uncertainty values.	80
4.22	Transient response to reference input with $H_\infty$ control.	80
4.23	Transient response to disturbance input with $H_\infty$ control.	81
4.24	Sensitivity and weighting functions of Mu-Control.	83
4.25	Robust stability of Mu-control	83
4.26	Nominal and robust performance of Mu-Control.	83
4.27	Sensitivity functions of perturbed systems with Mu-Control.	84
4.28	Performance of perturbed systems with Mu-Control.	84
4.29	Frequency responses of perturbed closed loop systems with Mu-Control.	85
4.30	Transient response to step reference input with Mu-Control.	86

4.31	Transient response to disturbance input with Mu-Control. ....	86
4.32	Transient responses of perturbed closed loop with Mu-Control. ....	86
4.33	Frequency responses of $H_\infty$ and $\mu$ -controllers. ....	87
4.34	Frequency responses of closed loop systems. ....	88
4.35	Comparison of robust stability of $H_\infty$ and $\mu$ -controllers. ....	88
4.36	Comparison of nominal performance of $H_\infty$ and $\mu$ -controllers. ....	89
4.37	Comparison of robust performance of $H_\infty$ and $\mu$ -controllers. ....	89
4.38	Performance degradation of $H_\infty$ and $\mu$ -controllers. ....	90
4.39	Frequency responses of full and reduced order controllers. ....	92
4.40	Transient response for reduced order controller. ....	93
4.41	LPV control structure includes the LPV system $P(\theta(t))$ , performance weighting functions $W_p$ , robustness function $W_u$ , and LPV polytopic controllers $(K_1(\theta(t)), \dots, K_3(\theta(t)))$ and input filter $W_f$ . ....	94
4.42	Parameter trajectory. ....	96
4.43	Step response of the LPV closed loop system. ....	96
5.1	Performance of PD-gravity controller, RR configuration; tracking step position of joint 1 (left), motor torque (right). ....	104
5.2	Performance of PD-gravity controller, RR configuration; tracking step position of joint 2 (left), motor torque (right). ....	105
5.3	Performance of PD-gravity controller, RR configuration; tracking step position error joint 1 (left), tracking position error joint 2 (right). ....	105
5.4	Performance of PD-gravity controller, RT configuration; tracking position of prismatic joint 2 (left), motor torque (right). ....	106
5.5	RRT configuration (Scara); tracking circular path of joint 1 (left), motor torque of joint 1 (right). ....	107
5.6	RRT configuration (Scara); tracking circular path joint 2 (left), motor torque of joint 2 (right). ....	107
5.7	RRT configuration (Scara); tracking circular path error joint 1 (left), tracking circular path error joint 2 (right). ....	108
5.8	Motor torques of joints 1 and 2 using control law (211); the chattering is due to the sign function. ....	118

5.9	Motor torques of joints 1 and 2 when replacing the function $sgn(s)$ with $sat(s/\Phi)$ . .....	119
5.10	Tracking positions of joints 1 and 2 using the control law (209.) Reference position (solid), actual position (dotted).....	119
5.11	Tracking velocities of joints 1 and 2 using the control law (209.) Reference velocity (solid), actual position (dotted).....	120
5.12	The phase portrait of the trajectory errors of joints 1 and 2. ....	120
5.13	Motor torques of joints 1 and 2 using control law (217); the chattering is due to the sign function.....	121
5.14	Motor torques of joints 1 and 2 when replacing the function $sgn(s)$ with $sat(s/\Phi)$ . .....	122
5.15	Tracking positions of joints 1 and 2 using control law (209). Reference position (solid), actual positions (dotted). ....	122
5.16	Tracking velocities of joints 1 and 2 using control law (209). Reference velocities (solid), actual positions (dotted). ....	123
5.17	The phase portrait of the trajectory errors of joints 1 and 2. ....	123
5.18	Tracking positions of joints 1 and 2 using the control law (219). Reference position (solid), actual positions (dotted). ....	125
5.19	Motor torques of joints 1 and 2 using the control law (224), replacing the function $sgn(s)$ with $sat(s/\Phi)$ .....	125
5.20	Trajectory errors of joints 1 and 2. ....	126
5.21	Control structure diagram of the adaptive control.....	127
5.22	Tracking position of the joint 1 to a trigonometric function (above), reference position (solid) and actual position (dotted). The tracking error (central), and the control torque (below). ....	130
5.23	Tracking position of the joint 2 to a trigonometric function (above), reference position (solid) and actual positions (dotted). The tracking error (central), and the control torque (below). ....	130
5.24	The normalized values of the inertia parameters $\varphi_1$ (above) and $\varphi_2$ (below). ....	131
5.25	The normalized values of the inertia parameters $\varphi_3$ (above) and $\varphi_4$ (below). ....	131

6.1	Configuration phase. ....	133
6.2	Control phase. ....	135
6.3	A comprehensive algorithm for configuration and control of a reconfigurable robot. ....	140
6.4	A reconfigurable robot follows a trajectory motion (lower and upper circles) ....	142
6.5	The joint coordinates of a reconfigurable robot during the joint space path. ....	143
6.6	Cartesian position (x,y,z) of the end effector during the joint space trajectory. ....	143
6.7	The Euler angles (roll, pitch, yaw) of the end effector. ....	144
6.8	Manipulability of the reconfigurable robot following a joint space trajectory. ....	144
6.9	Front view of the workspace in the $xz$ -plane of a reconfigurable robot. ....	146
6.10	End effector motion from pose (0.4064, 0.5,0.3303) to (0.8032, 0.5,0.3303). ....	146
6.11	Joint coordinates motion versus time. ....	147
6.12	Cartesian position of the end effector versus time. ....	147
6.13	Cartesian position of the trajectory in the $xy$ -plane versus time. ....	148
6.14	Euler angles, roll-pitch-yaw of the end effector versus time. ....	148
6.15	Performance weighting function of the sensitivity function $S = \frac{1}{(1+GK)}$ ....	150
6.16	Step response of $H_\infty$ -control. ....	150
6.17	Disturbance response using $H_\infty$ -control. ....	151
6.18	Frequency response of a weighted sensitivity function $S = \frac{1}{(1+GK)}$ ....	152
6.19	Step response of 20 frozen parameters using gain-scheduled controller. ....	152
6.20	Gain-scheduled control amplitude. ....	153
6.21	Tracking error using gain-scheduled control. ....	153
6.22	Trajectory tracking for a trigonometric reference command using SMC method. ....	154
6.23	Tracking error for unknown parameters using SMC method. ....	155
6.24	Trajectory tracking for a trigonometric function using the adaptive control method. ....	155



6.25 Tracking error for time varying dynamic parameters using adaptive control method.....	156
--	-----

## List of Tables

2.1	D–H parameters of the $n$ -GKM model.....	12
2.2	D–H of a spherical wrist, source; Spong, [80]. .....	17
2.3	D–H parameters of the ABB manipulator robot, source; Spong, [80]. .	21
2.4	D–H parameters of the Stanford manipulator robot, source; Spong, [80].....	23
3.1	Reconfiguration parameters values.....	37
3.2	Reconfigurable D–H parameters of the 3-GKM model.....	39
3.3	Kinematic initial parameters of 2-DOF RR configuration .....	40
3.4	Dynamic initial parameters of 2-DOF RR configuration .....	40
3.5	Kinematic initial parameters of 2-DOF RT configuration.....	41
3.6	Dynamic initial parameters of 2-DOF RT configuration.....	41
3.7	Kinematic initial parameters of 2-DOF TR configuration .....	42
3.8	Dynamic initial parameters of 2-DOF TR configuration .....	42
3.9	Kinematic initial parameters of 2-DOF TT configuration .....	43
3.10	Dynamic initial parameters of 2-DOF TT configuration.....	43
3.11	Kinematic initial parameters of 3-DOF SCARA configuration.....	43
3.12	Dynamic initial parameters of 3-DOF SCARA configuration .....	44
3.13	D–H parameters of the PUMA 560 Manipulator, source; Fu, [41].....	46
6.1	Dynamic parameter properties of a reconfigurable robot.....	136
A.1	Nominal parameter values of the Bosch Scara robot arm .....	166
A.2	Nominal parameter values of the Bosch Scara robot arm .....	166
A.3	Different values for inertias and damping for different joint position of link 2 (left). Estimated parameter values for stiction and Coulomb friction (right). .....	167

## Nomenclature

$L_{mi}$	Inductance of motor coil of joint $i$	[ $H$ ]
$L_i$	Size of the Link $i$	[ $H$ ]
$R_{mi}$	Impedance of motor armature, included wires of joint $i$	[ $\Omega$ ]
$K_{mi}$	Servo motor constant of joint $i$	[ $N\ m/A$ ]
$J_{mi}$	Inertia of servo motor of joint $i$	[ $kgm^2$ ]
$J_{Li}$	Inertia of the Link $i$	[ $kgm^2$ ]
$J_i$	Total inertia of Link $i$	[ $kgm^2$ ]
$J_c$	Cross-coupling inertia between joints	[ $kgm^2$ ]
$F_{vi}$	Viscous friction of joint $i$	[ $N\ ms/rad$ ]
$F_{vL}$	Viscous friction in mechanical link 1	[ $N\ ms/rad$ ]
$F_{vm}$	Viscous friction in Servo motor of joint 1	[ $N\ ms/rad$ ]
$F_{ci}$	Coulomb friction of joint $i$	[ $N\ m$ ]
$F_{si}$	Static friction (Stiction) of joint $i$	[ $N\ m$ ]
$K_{Pi1}$	Proportional constant of tacho controller in Servo Amplifier $i$	[1]
$K_{Ii1}$	Integration constant of tacho controller in Servo Amplifier $i$	[ $s^{-1}$ ]
$K_{Pi2}$	Proportional constant of current controller in Servo Amplifier $i$	[ $V/A$ ]
$K_{Ii2}$	Integration constant of current controller in Servo Amplifier $i$	[ $V/As$ ]
$T_i$	Time constant of Tacho controller in Servo Amplifier $i$	[ $s^{-1}$ ]
$m_l$	Load mass	[ $kg$ ]
$K_S$	Spring constant	[ $N\ m/rad$ ]
$D_S$	Damper constant	[ $N\ ms/rad$ ]

$\theta_{mi}$	Position of motor axis of joint $i$	[rad]
$\theta_i$	Position of joint axis $i$	[rad]
$\dot{\theta}_{mi}$	Velocity of motor axis of joint $i$	[rad/s]
$\dot{\theta}_i$	Velocity of joint axis $i$	[rad/s]
$\ddot{\theta}_i$	Acceleration of joint axis $i$	[rad/s <sup>2</sup> ]
$C_1$	Cos( $\theta_1$ )	
$S_1$	Sin( $\theta_1$ )	
$C_{12}$	Cos( $\theta_1 + \theta_2$ )	
$S_{12}$	Sins( $\theta_1 + \theta_2$ )	
$N_1$	Gear ratio	[1]
$\tau_{LD,i}$	Disturbance torques on joint $i$	[N m]
$\delta_{ADi}$	Quantization error of sensor joint $i$	[rad]
$\delta_{DA}$	Resolution of DA-converter	[V]
$\delta_F$	Perturbation in friction	[N ms/rad]
$\delta_{1/J}$	Perturbation in inertia	[1/kgm <sup>2</sup> ]
$\delta_\theta$	Tracking error	[rad/s]
$p_1$	Interconnection output to inertia error	[N m]
$p_2$	Interconnection output to friction error	[rad/s]
$q_1$	Interconnection input to inertia error	[N/kgm]
$q_2$	Interconnection input to friction error	[N m]
$U_S$	Input voltage to Servo motor	[V]
$U_{SA}$	Input voltage to Tacho controller of Amplifier System	[V]
$I_{SA}$	Input current to Current controller of Amplifier System	[A]
$h$	Sample time	[s]
$W_\Delta$	Scaling function for perturbation matrix	

- $W_P$  Weighing function of tracking error (Performance Spec)
- $W_\theta$  Scaling function for velocity set point
- $W_\tau$  Scaling function for disturbance torque
- $W_{DA}$  Scaling function for Amplifier set point
- $W_F$  Scaling of Friction perturbation
- $W_J$  Scaling of inertia perturbation
- $K$  Controller
- $B$  Set of all normed perturbation matrices with  $\sigma_{max}(\Delta) \leq 1$
- $M$  Closed-loop interconnection matrix
- $M'$  Open-loop interconnection matrix
- $\Delta$  Noise-block, built of  $\Delta_1$  and  $\Delta_P$
- $\Delta_1$  Perturbation transfer matrix (structured)
- $\Delta_P$  Block structure appended to  $\Delta$  for robust performance calculations
- $d$  Input signals
- $e$  Tracking error
- $c$  Controller signal
- $w$  Input of perturbation matrix 1 into the interconnection matrix
- $z$  Output of the interconnection matrix to the perturbation matrix 1

## CHAPTER 1

### Introduction and Preliminaries

#### 1.1. Introduction to Robot Kinematics

A serial-link manipulator comprises a set of bodies called links connected in a chain by joints. Each joint has one degree of freedom, either translational (sliding or prismatic joint) or rotational (revolute joint). To describe the rotational and translational relationships between adjacent links, Denavit and Hartenberg proposed a matrix method of systematically establishing a coordinate frame to each link of an articulated chain. The Denavit-Hartenberg (D–H) representation [35] results a  $4 \times 4$  homogeneous transformation matrix representing each link's coordinate frame at the joint with respect to the previous link's coordinate frame. To analyze the motion of robot manipulator, coordinate frames are attached to each link starting from frame  $F_0$ , attached to the base of the manipulator link, all the way to the frame  $F_n$ , attached to the robot end-effector as shown in Figure 1.1. Every coordinate frame is determined and established on the basis of three rules:

- (1) The  $z_{i-1}$  axis lie along the axis of motion of the  $i$ th joint.
- (2) The  $x_i$  axis is normal to the  $z_{i-1}$  axis.
- (3) The  $y_i$  axis completes the right-handed coordinate system as required.

As the frames have been attached to the links, the following definitions of the link (D–H) parameters are valid:

- Joint angle  $\theta_i$  is the angle around  $z_{i-1}$  that the common perpendicular makes with vector  $x_{i-1}$ .
- Link offset  $d_i$  is the distance along axis  $z_{i-1}$  to the point where the common perpendicular to axis  $z_i$  is located.
- Link length  $a_i$  is the length of the common perpendicular to axes  $z_{i-1}$  and  $z_i$ .
- Link twist  $\alpha_i$  is the angle around  $x_i$  that vector  $z_i$  makes with vector  $z_{i-1}$ .

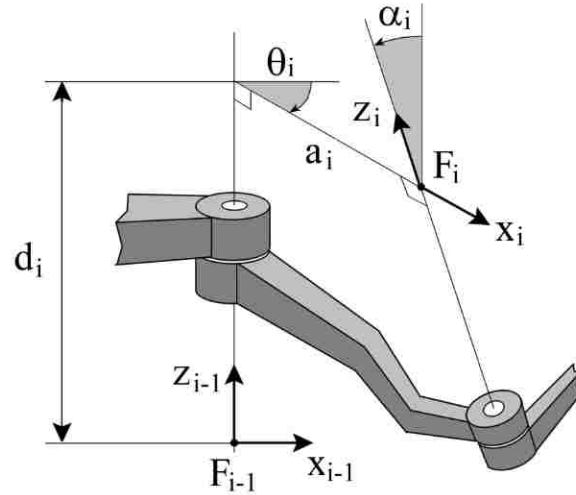


FIGURE 1.1. Definition of standard Denavit and Hartenberg (D-H) parameters. Source; Manseur, [63].

For a rotary joint,  $d_i$ ,  $a_i$ , and  $\alpha_i$  are the joint parameters and remain constant for a robot, while  $\theta_i$  is the joint variable that changes when link  $i$  rotates with respect to link  $i - 1$ . For a prismatic joint,  $\theta_i$ ,  $a_i$ , and  $\alpha_i$  are the joint parameters and remain constant for a robot, while  $d_i$  is the joint variable.

## 1.2. Introduction to Reconfigurability Theory

Robotics technology has been recently exploited in a variety of areas and various robots have been developed to accomplish sophisticated tasks in different fields and applications such as in space exploration, future manufacturing systems, medical technology, etc. In space, robots are expected to complete different tasks, such as capturing a target, constructing a large structure and autonomously maintaining in-orbit systems. In these missions, one fundamental task with the robot would be the tracking of changing paths, the grasping and the positioning of a target in Cartesian space. To satisfy such varying environments, a robot with changeable configuration (kinematic structure) is necessary to cope with these requirements and tasks. Another field of technology is the new manufacturing environment, which is characterized by frequent and unpredictable market changes. A manufacturing paradigm called Reconfigurable Manufacturing Systems (RMS) was introduced to address the new production challenges [52]. RMS is designed for rapid adjustments of production capacity and functionality in response to new

circumstances, by rearrangement or change of its components and machines. Such new systems provide exactly the capacity and functionality that is needed, when it is needed [39]. The rapid changes and adjustments of the RMS structure must happen in a relatively short time ranging between minutes and hours and not days or weeks. These systems' reconfigurability calls for their components, such as machines and robots to be rapidly and efficiently modifiable to varying demands [48]. Robot manipulators working in extreme or hazardous environments (biological, chemical, nuclear,..., etc.) often need to change their configuration and kinematic structures to meet the demands of specific tasks. It is desirable and cost effective to employ a single versatile robot capable of performing tasks such inspection, contact operations, assembly (insertion or removal parts), and carrying objects (pick and place). Robots with maximum manipulability are well conditioned for dexterous contact tasks [16, 61] and configurations that maximize the robot links and distance from the environment are suitable for payload handling [56]. The optimization of a robot workspace over its link lengths, as the design parameters, is reported in [55, 47], while optimization of kinematic parameters and criteria for fault tolerance are discussed in [50].

In the literature, modular robotic structures are presented as a solution to cope with reconfigurable structure of robots. A modular reconfigurable robot consists of a collection of individual link and joint components that can be arbitrarily assembled into a number of different geometries. Such a system can provide agility to the user to cope with a wide spectrum of tasks through proper selection and reconfiguration of a large inventory of functional components. Several prototyping systems have been demonstrated in various research institutions, [87], [42], [67] and [13]. An automated generation of D–H parameters methodology has developed for the modular manipulators [22]. The authors derived the kinematic and dynamic models of reconfigurable robots using D–H parameters for different sets of joints, links and gripper modules as shown in Figure 1.2. Furthermore, a library of modules is formed from which any module can be called with its associated kinematic and dynamic models. In [58], a modular and reconfigurable robot design is introduced with modular joints and links. The proposed design introduces zero



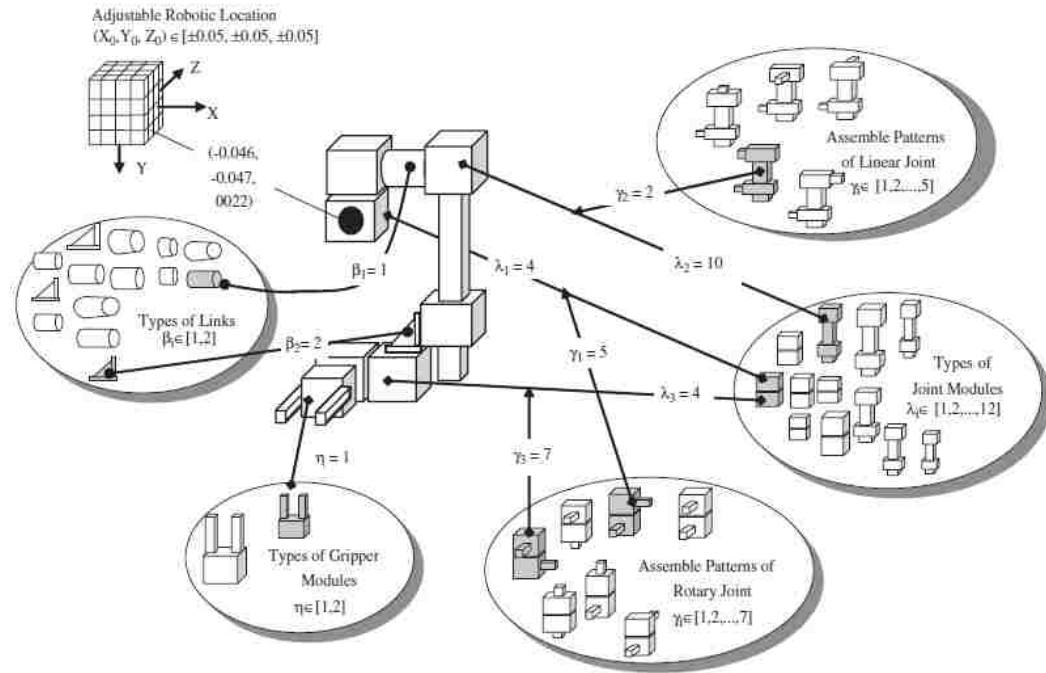


FIGURE 1.2. Set of design variables of the 3-DOF configuration modular robots, this set is a commercial product of AMTEC GmbH Company. Source; I.M. Chen, [22].

link offsets to increase the robot's dexterity and maximize its reachability. A modular and reconfigurable robot (MRR) with multiple working modes was designed [59]. In the proposed MRR design, each joint module can independently work in active modes with position or torque control, or passive modes with friction compensation. With the MRR, the joint module was designed as a hybrid joint in working modes and not in the sense of mechanical motion. A reconfigurable robot was proposed [5] and achieves the reconfigurability by utilizing passive and active joints. In [21], an automated approach was presented to build kinematic and dynamic models for assembled modular components of robots. The developed method is applicable to any robotic configuration with a serial, parallel or hybrid structure. Reconfigurable plug and play robot kinematic and dynamic modeling algorithms are developed [22]. These algorithms are the basis for the control and simulation of reconfigurable modular robots. The reconfigurable robot (RRS) was regarded as a modular system [20]. A task-based configuration optimization based on a generic algorithm was used to solve a predefined set of joint modules for specific kinematic configuration. A modular and reconfigurable robot for industrial

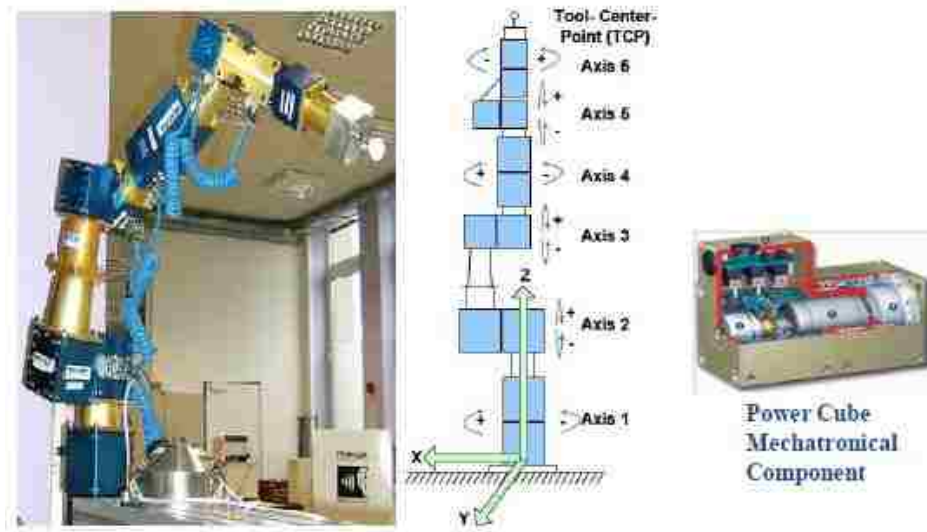


FIGURE 1.3. Mechanical set up of a modular and reconfigurable robot (left). The ICT power cube Mechatronical component (right). Source; Strasser, [84].

purposes has been introduced [84]. The PROFACTOR GmbH has presented a modular and reconfigurable robot with power cube (Mechatrical Components) modules depicted in Figure 1.3. These modules were designed to be identical and self-contained with actuation, memory, and mechanical, electrical and embedded programming. A reconfigurable robot has been introduced by [36] that unifies the kinematic structure of industrial robots. In that unification process, eight modules were reconfigured by changing configuration parameters. These parameters represent the trigonometric functions of the robot twist angles.

The main drawbacks of the modular robots proposed in the literature are the high initial investment necessary in modules that remain idle during many activities, and the significant lead time for replacement, attachment and detachment of the components prior to performing a specific task.

### 1.3. Robot Control

The use of advanced robot control laws may contribute significantly to improve the robot functions and properties. The improvement of the robot design itself can also contribute substantially to the desired increase in performance and capabilities. The combination of the controller with proper sensors can provide some

sense and awareness of the environment, and improve its accuracy and speed as well.

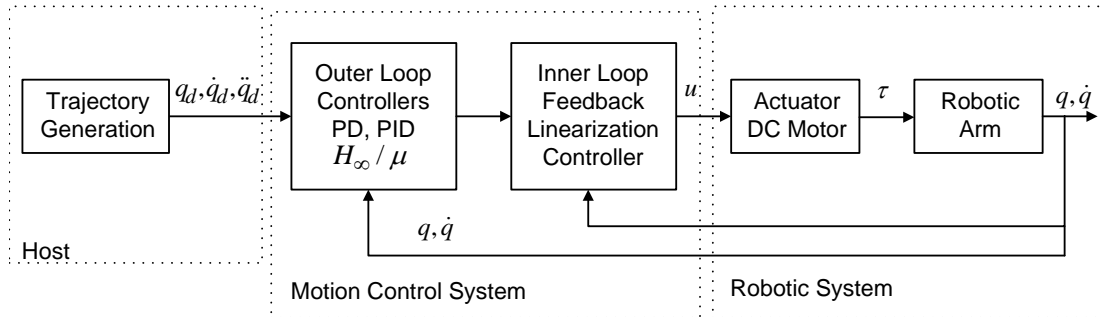


FIGURE 1.4. Robotic system with motion control system, inner and outer loop controllers.

The former and current problems in the robotics application fields has affected the research of robot control in a number of fundamental topics: modeling, position control, robust control and motion planning. This has motives research in robot modeling, simulation and control design. Therefore, position and trajectory control is an important research field in robotics control. The position/motion control problem has received a great deal of interest in robotics. Therefore, a survey that covers the important control strategies is given with examples and applications.

### PD and PID Control

The PD (Proportional-Derivative) and PID (Proportional-Derivative-Integral) controllers are the most applied in industry, which is also true for robotics. Some references propose a high gain PD controller to ensure global stabilization of the robot [68, 69, 71], which is unsuited for practical applications due to excitation of unavoidable higher dynamics and excessive noise amplification. PID controllers are more suitable to eliminate the steady state error of the final position response. These controllers introduce an integration action to the resulting closed loop improving the performance tracking requirements.

### Feedback Linearization Control

The application of feedback linearization theory to solve robotics control problems has led to the computed torque approach. Feedback linearization control

methods are inner-outer loop control methods: the inner loop must linearize the plant, whereas the outer loop must achieve the desired closed loop requirements. Figure 1.4 shows the control motion structure (inner and outer control loops) of a robot driven by a DC motor. The term Computed Torque Control (CTC) is the application of PD control at the outer loop to a linearized system by the feedback linearization control. In robotics, CTC is used to apply PD controllers at the outer loop independently (every joint controlled separately) [53]. There are two important features of the feedback linearization method that require attention: model error and the outer loop controller design. Feedback linearization is based on the exact model of the system. Therefore, the controller may be sensitive to modeling errors such as parameter errors and unmodeled dynamics. Parameter uncertainty is commonly addressed by either robust control methods or by the derivation of adaptive controllers [66, 77]. In particular, when a restricted amount of parameters must be estimated (in case of an unknown load), adaptive controllers can be a suitable approach. The feedback linearization control actively linearized the plant, such that the resulting system can be considered as a linear system. Therefore, it is possible to apply one of many linear control methods to close the loop and achieve the required performance. As a result, a large number of controllers for the outer loop control are proposed: the standard PD loop of CTC, linear optimal control [83, 78], sliding mode control, and  $H_\infty/\mu$  robust optimal controllers.

### Lyapunov Based Control

An important tool for control of rigid body systems is Lyapunov stability theory, which based on the strict dissipation of a suitable energy function [76]. Although this theory is not constructive to design a controller, a simple structure of the equations of motion with some relevant assumptions allow a derivation of stabilizing controllers. These assumptions may include bounded disturbances and bounded parameter variations. The passivity based control approach attempts to reshape the robot energy function, rather than imposing a completely different behavior as with the CTC approach [14, 19]. Experiments have shown and indicated the

passivity controllers are more robust than CTC. Another result of Lyapunov stability theory is the sliding mode control (SMC), which is considered to be a robust control approach.

### Robust Control

To ensure a suitable behavior of the closed loop robot, even in the presence of modeling errors and disturbances, it is desired to design controllers that are robust with respect to these errors and disturbances. Modeling errors are generally separated into parameter errors and unmodeled dynamics, which may have different affect on the closed loop system. The standard control framework, adopted in many textbooks on modern control [23, 60, 88], is shown in Figure 1.5. A controller  $K(s)$  is provided with measurement signals  $y$  and has to stabilize a plant  $P(s)$  with input signals  $u$  such that the cost variables  $z$  are minimal in some sense, despite the disturbance signals  $w$  and the parametric and dynamic uncertainties represented in  $\Delta(s)$ . The plant  $P(s)$  is often called the generalized plant or standard plant since it usually does not only consist of the plant to be controlled, but can also contain weightings, e.g., parametric and dynamic uncertainty weighting, input signal thresholds, and the robot dynamic model to be simulated. Also the other entities can be viewed in a generalized way, e.g., reference signals can be incorporated as disturbances  $w$  and additional feedback paths can be taken in case of a robust control problem to describe a set of systems, i.e., uncertainty. A survey can be found covering a number of robust robot position controller design methods: passivity control, sliding mode control and linear robust control by factorization approach in [82]. The most popular robust control design method in robotics literature is the sliding mode control (SMC), also known as Variable Structure Switching (VSS) control. Sliding mode control is commonly used to address parameter uncertainty and bounded disturbances [76, 86]. As mentioned, SMC is based on upon Lyapunov stability theory, and basically tries to determine the nominal feedback control law and a corrective control action that steers the controlled system to the desired behavior, defined as the ‘*sliding surface*’.

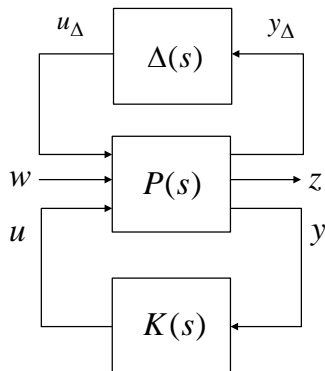


FIGURE 1.5. Standard robust control problem.

### Optimal Control

Apart from the application of linear outer loop control applied to the feedback linearized robot, there have been some attempts made to apply optimal control directly to the rigid body dynamics. The main problem with these approaches is the amount of assumptions and choices that have to be made to allow for a solution. For example in [30], an optimal quadratic control is considered with a special choice optimization criterion, which results in a nonlinear PID controller. Another example in [44], where an  $H_\infty$ -optimal control problem is considered, results in a nonlinear static state feedback PD controller. Following these methods to construct a nonlinear controller does not allow the versatility required for a controller design method needed to solve real-world problems.

The reason is that currently the nonlinear control theory cannot provide a general robust controller design methodology, due to high complexity of both the robot model and the involved design specifications. General cases addressed with optimal control infrequently allow a closed solution, and one has to resort to computationally intensive numerical methods. In the case of special properties of the uncertainty, e.g., signal roundedness, there do exist applicable controller design methods, e.g., sliding mode control. These control methods have restricted applicabilities as they cannot exploit structural knowledge of the uncertainty. Linear control theory does have that capability e.g.  $H_\infty/\mu$  controller design methods have limited means of specifying the desired properties of the closed loop system. Linear control theory also has its limits, but offers a larger variety of specifications.

## 1.4. Problem Statement

Current robot structures have physical limitations with respect to their configurations and capabilities. They are preconfigured to do specific tasks. For example: a robot structure with 5-DOF (3R-2T) would have three revolute (rotational motion) and two prismatic (translational motion) joints with fixed coordinate frames that cannot be automatically changed to any other configuration. The structure of most robots can be changed only by physically replacing their joints or links (modules). These limitations are reflected on the robot's path, workspace, inertia, torque, power concept,..., etc., making them unsuitable for future RMS.

### 1.4.1. General Problem Statement

The aforementioned leads to the following problem statement for this research: **Propose a robot with new properties to address the reconfigurability problem, including its feasible solutions using model based control strategies.**

### 1.4.2. Research Approach

Structural robot design and control methods are combined to solve the reconfigurability problem. A rotational/translational reconfigurable joint is investigated to add new properties necessary to extend the robot capabilities in performing more sophisticated tasks. The D-H parameters of a reconfigurable robot will be regarded as variable to describe all possible kinematic configurations. A Global Kinematic Model (GKM) is developed based on specific reconfigurable parameters to automate generation models for any robot configuration. Then, an automatic generation of dynamic equations using the Global Dynamic Model (GDM) is constructed to auto-generate the equations of motion of any specified configuration. The recursive Newton-Euler algorithm is employed to generate the dynamic elements: the inertia matrix, Coriolis torque matrix, centrifugal torque matrix, and the gravity torque vector. The parameters of a reconfigurable robot are often unknown, nonlinear or uncertain. Moreover, most of these parameters are time varying, position and orientation (pose) dependent. Consequently, the following control strategies were explored and analyzed thoroughly:

- Nonlinear PD-Gravity control.
- Optimal robust control such as  $H_\infty/\mu$  controllers.
- Gain Scheduling control.
- Sliding Mode Control (SMC).
- Adaptive control.

Based on the dynamic parameter types, a reconfigurable control algorithm is developed, which leads to optimize the control method selection for a specific kinematic structure.



## CHAPTER 2

### Development of a Reconfigurable Robot Kinematics

A development of the general  $n$ -DOF Global Kinematic Model ( $n$ -GKM) is necessary for supporting any open kinematic robotic arm, and possible redundant kinematic structures that are intended to support more than 6-DOF. The  $n$ -GKM model is generated by the D–H parameters, given in Table 2.1 and as proposed by Djuric, Al Saidi, and ElMaraghy [37]. All D–H parameters presented in the Table 2.1 are not fixed values; they are modeled as variables to satisfy the properties of all possible open kinematic structures of a robotic arm. The twist angle variable  $\alpha_i$  is limited to five different values,  $(0^0, \pm 90^0, \pm 180^0)$ , to maintain perpendicularity between joints' coordinate frames. Consequently, each joint has six different positive directions of rotations and/or translations.

TABLE 2.1. D–H parameters of the  $n$ -GKM model.

$i$	$d_i$	$\theta_i$	$a_i$	$\alpha_i$
1	$R_1 d_{DH1} + T_1 d_1$	$R_1 \theta_1 + T_1 \theta_{DH1}$	$a_1$	$0^0, \pm 180^0, \pm 90^0$
2	$R_2 d_{DH2} + T_2 d_2$	$R_2 \theta_2 + T_2 \theta_{DH2}$	$a_2$	$0^0, \pm 180^0, \pm 90^0$
...	...	...	...	...
3	$R_n d_{DHn} + T_n d_n$	$R_n \theta_n + T_n \theta_{DHn}$	$a_n$	$0^0, \pm 180^0, \pm 90^0$

The subscript  $DHn$  implies that the  $d_i$  or  $\theta_i$  parameter is constant.

#### 2.1. Modeling of a Reconfigurable Joint

The reconfigurable joint is a hybrid joint that can be configured to be a revolute or a prismatic type of motion, according to the required task. For the  $n$ -GKM model, a given joint's vector  $z_{i-1}$  can be placed in the positive or negative directions of the  $x$ ,  $y$ , and  $z$  axis in the Cartesian coordinate frame. This is expressed in Equations (1)-(2):

$$\text{Rotational Joints : } R_i = 1 \text{ and } T_i = 0 \quad (1)$$

$$\text{Translational Joints : } R_i = 0 \text{ and } T_i = 1 \quad (2)$$

The variables  $R_i$  and  $T_i$  are used to control the selection of joint type (rotational and/or translational). The orthogonality between the joint's coordinate frames is achieved by assigning appropriate values to the twist angles  $\alpha_i$ . Their trigonometric function are defined as the joint's reconfigurable parameters ( $K_{Si}$  &  $K_{Ci}$ ) and expressed in Equations (3)-(4):

$$K_{Si} = \sin(\alpha_i) \quad (3)$$

$$K_{Ci} = \cos(\alpha_i) \quad (4)$$

To construct a reconfigurable joint, all six different positive directions of rotations or translations must be included. The procedure will start from the first coordinate frame by defining the orientation of the vector,  $Z_0$ . Because there are six combinations of vector  $Z_0$ , the process starts from the first one, named  $Z_0^1$ . The selection of vector  $Z_0^1$ , can be combined with four more orientations of vectors  $X_0$  and  $Y_0$ . They are:  $X_0^{11}, Y_0^{11}, X_0^{12}, Y_0^{12}, X_0^{13}, Y_0^{13}, X_0^{14}, Y_0^{14}$ . The second combination of  $Z_0^1$  and its  $X_0$  and  $Y_0$  includes the new vector  $Z_0^2$  and the four combinations:  $X_0^{21}, Y_0^{21}, X_0^{22}, Y_0^{22}, X_0^{23}, Y_0^{23}, X_0^{24}, Y_0^{24}$ . Similarly, all other possible combinations of different  $Z_0^1$  and the  $X_0$  and  $Y_0$  vectors. This will produce a reconfigurable joint having 24 different possible coordinate frames. Thus, a reconfigurable joint model includes 6R and 6T different types of motion, which is the maximum number of motions that can be produced in 3D space. The following five definitions are developed for proper use of the model.

**DEFINITION 2.1.** The degree of the joints reconfigurability,  $R_J$  can be between 2 and 12. This parameter defines the level of the joints reconfigurable capabilities, Equation (5).

$$2 \leq R_j \leq 12 \quad (5)$$

DEFINITION 2.2. Similarly, the links reconfigurable parameter,  $R_L$ , which is simply the changeable link length, and can be any real number.

$$R_L \in \mathbb{R} \quad (6)$$

DEFINITION 2.3. For any joint to be reconfigurable, the following condition must be satisfied: the number of different motions should be a minimum of two, Equation (7).

$$\min(R_j) = 2 \quad (7)$$

From those three definitions a clear description of the reconfigurable robot is achieved.

DEFINITION 2.4. The robot is reconfigurable if and only if it has a minimum of one reconfigurable joint or link.

DEFINITION 2.5. The  $n$ -DOF Global Kinematic Model ( $n$ -GKM) is a reconfigurable model because it has all reconfigurable joints and links. Its joints satisfy the maximum number of reconfigurations ( $R_j = 12$ )

The  $n$ -GKM model starts from the base frame, which represents the coordinate frame  $(x_0, y_0, z_0)$  of the first joint, and has six possible frames for the second joint, presented with coordinate frame  $(x_1, y_1, z_1)$ . From the second joint coordinate frame  $(x_1, y_1, z_1)$ , there are again six different combinations for joint three's coordinate frame  $(x_2, y_2, z_2)$ , and so on, up to the flange frame  $(x_n, y_n, z_n)$ .

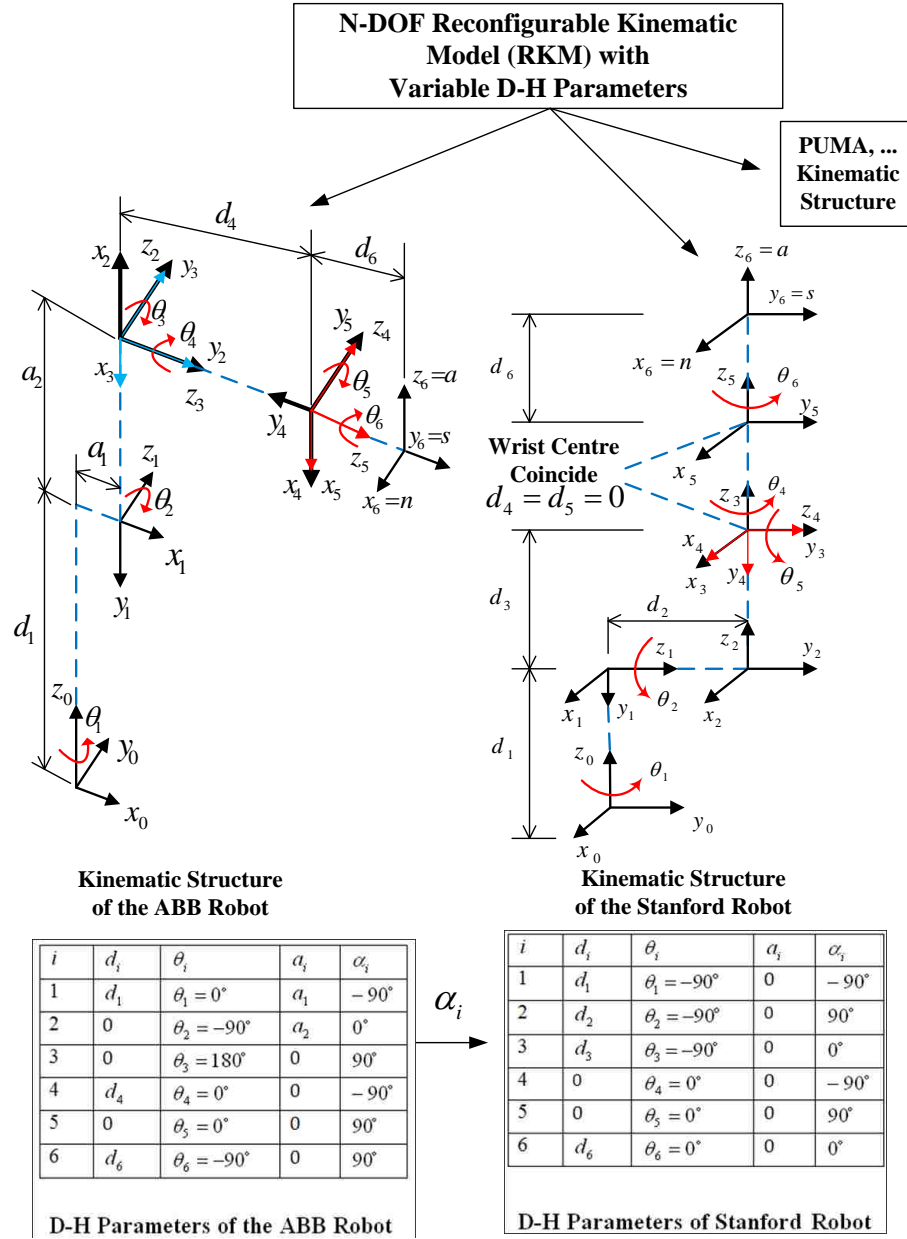


FIGURE 2.1. Kinematic structures of the ABB and Stanford robots, D-H parameters are from sources; Dawson, [57] and Spong, [80].

## 2.2. Modeling of Reconfigurable Open Kinematic Robots

The reconfigurability of a robotic arm is modeled based on the variable D-H parameters and especially, the variable twist angle between adjacent links. Defining the varying twist angle as the configuration parameter allows the model to achieve any kinematic structure by configuring the parameter accordingly. Figure 2.1 shows diverse industrial robots such as ABB and Stanford achieved as special

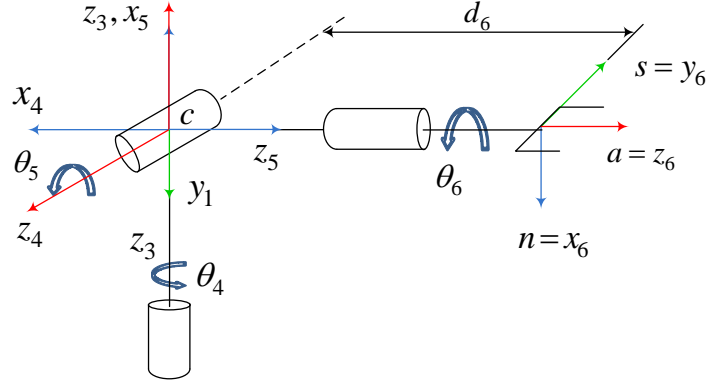


FIGURE 2.2. The spherical wrist, joint axes 4, 5 and 6.

cases by changes to the configuration parameter. The kinematics of the  $n$ -GKM model can be calculated using the multiplication of the all homogeneous matrices from the base to the flange frame. The homogeneous transformation matrix of the  $n$ -DOF Global Kinematic Model (GKM) is given by the following equation:

$${}^{i-1}A_i = \begin{bmatrix} \cos(\phi_i) & -K_{ci}\sin(\phi_i) & K_{si}\sin(\phi_i) & a_i\cos(\phi_i) \\ \sin(\phi_i) & K_{ci}\cos(\phi_i) & -K_{si}\cos(\phi_i) & a_i\sin(\phi_i) \\ 0 & K_{si} & K_{ci} & \phi_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

where  $\phi_i = R_i\theta_i + T_i\theta_{DHi}$ . Using this transformation matrix, models of different open kinematic structures can be automatically generated which characterizes the new reconfigurable robot.

### 2.2.1. Spherical Wrist

The spherical wrist, shown in Figure 2.2, is a three joint wrist mechanism for which the joints axes  $z_3$ ,  $z_4$  and  $z_5$  intersect at the center  $c$ . The D–H parameters of the mechanism are shown in Table 2.2. A spherical wrist satisfies Piper’s condition [45] when  $a_4 = 0$ ,  $a_5 = 0$  and  $d_5 = 0$ . The end effector coordinate frame is:  $n$  is the normal vector,  $s$  is the sliding vector and  $a$  is the approach vector.

TABLE 2.2. D–H of a spherical wrist, source; Spong, [80].

Link	$\theta_i$	$d_i$	$a_i$	$\alpha_i$
4	$\theta_4$	0	0	$-90^0$
5	$\theta_5$	0	0	$90^0$
6	$\theta_6$	$d_6$	0	$0^0$

### 2.2.2. Assumption

Assuming a spherical wrist is attached to the end effectors, the kinematic structures of the common industrial robots are determined by only the first three joints and links. This assumption also defines the external and internal workspace boundaries. A spherical wrist that satisfies Piper’s condition only serves to orient the end-effector within the workspace. A hybrid joint (revolute/prismatic) motion and its selection parameters are mathematically expressed in the following Equation:

$$q_i = R_i\theta_i + T_id_i \quad (9)$$

For a reconfigurable three links and joints (3-DOF), the resulting possible kinematic structure combinations are  $2^3 = 8$ : Articulated (RRR), Cylindrical (RTR), Spherical (RRT), SCARA (RRT), Cartesian (TTT), TRR, TTR, RTT and TRT. These kinematic structures are shown in Figure 2.3.

### 2.3. Reconfigurable Jacobian Matrix

The Jacobian matrix  $J \in R^{n \times m}$  is a linear transformation that maps an n-dimensional velocity vector  $\dot{q}_i$  into an m-dimensional velocity vector  $\dot{V}_i$ :

$$\dot{V}_i = \begin{bmatrix} v \\ w \end{bmatrix} = J(q)\dot{q}_i \quad (10)$$

where the vector  $[v^T, w^T]$  are the end effector velocities and  $\dot{q}_i$  is the joint velocities. For robot manipulators, the Jacobian is defined as the coefficient matrix of any set of equations that relate the velocity state of the tool coordinate described in the Cartesian space to the actuated joint rates of the joint velocity space. It is necessary that  $J(q)$  have six linearly independent columns for the end effector to

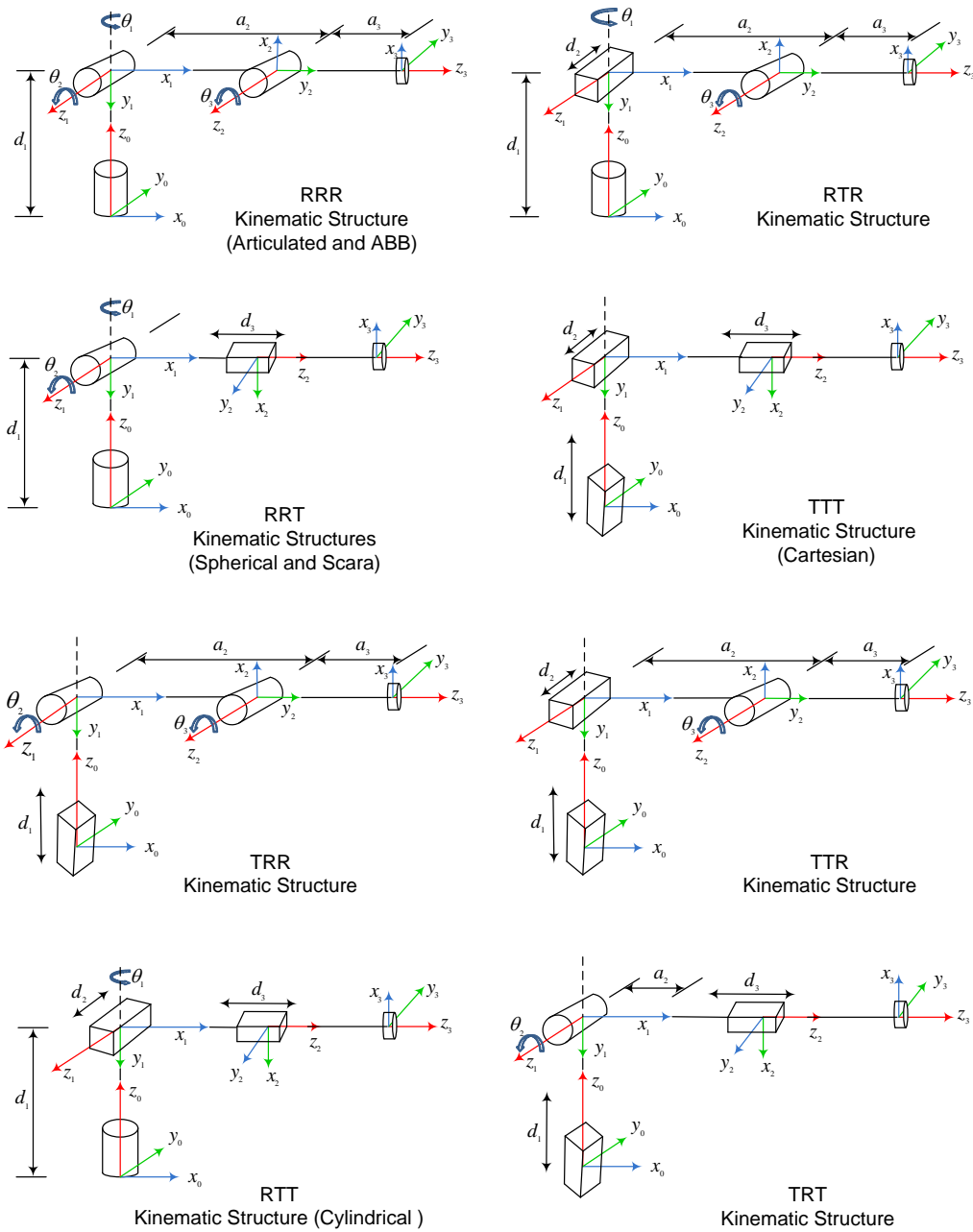


FIGURE 2.3. All possible configuration models of a reconfigurable hybrid joint.

be able to achieve any arbitrary velocity. Thus, when the rank  $J(q) = 6$ , the end effector can execute any arbitrary velocity. Actually, the rank of the manipulator Jacobian matrix will depend on the configuration  $q$ . Configurations for which

the rank  $J(q)$  is less than its maximum value are called *singular configuration*. Identifying manipulator singularities is important for several reasons:

- Singularities represent configurations from which certain directions of motion may not be achievable.
- Singularities correspond to points of maximum reach on the boundary of the manipulator workspace.
- At singularities, bounded end effector velocities may correspond to unbounded joint velocities.

### 2.3.1. Decoupling of Singularities

In general, it is difficult to solve the nonlinear equation  $\det J(q) = 0$ . Therefore, decoupling the singularities and division of singular configurations into arm and wrist singularities are considered [80]. The first step is to determine the singularities resulting from motion of the arm, and the second is to determine the wrist singularities resulting from motion of spherical wrist. For a manipulator of  $n = 6$  consisting of a 3-DOF arm and 3-DOF spherical wrist the Jacobian is a  $6 \times 6$  matrix and a configuration  $q$  is singular if and only if:

$$\det(J(q)) = 0 \quad (11)$$

where the Jacobian  $J(q)$  is partitioned into  $3 \times 3$  blocks as:

$$J(q) = [J_P \quad J_O] = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \quad (12)$$

Since the final three joints are always revolute and intersect at a common point  $c$ , Figure 2.2, then  $J_O$  becomes:

$$J_O = \begin{bmatrix} 0 & 0 & 0 \\ z_3 & z_4 & z_5 \end{bmatrix} \quad (13)$$

In this case the Jacobian matrix has the block triangle form:



$$J(q) = \begin{bmatrix} J_{11} & 0 \\ J_{21} & J_{22} \end{bmatrix} \quad (14)$$

with determinant:

$$\det J(q) = \det J_{11} \det J_{22} \quad (15)$$

As a result, the set of singular configurations of the manipulator is the union of the set of arm configurations satisfying  $\det J_{11} = 0$  and the set of wrist configurations satisfying  $\det J_{22} = 0$ .

## 2.4. Manipulability and Singularity

The workspace of a reconfigurable manipulator defines a variable volume depending on the variable D–H parameters of joint twist angle, link offset and link length. A variable workspace of a 3-DOF reconfigurable manipulator with an RRR configuration is shown in Figure 2.4. The variable workspace is calculated with twist angle change values of  $\pi/16$ ,  $\pi/8$ ,  $\pi/4$ , and  $\pi/2$ . The resulting workspace is a union set of spherical and elliptical volumes around the first manipulator joint. In a similar fashion, Figure 2.5 shows a variable workspace of a reconfigurable RRT configuration with different third link lengths of 0.15, 0.3 and 0.45 m. The workspace layers are spherical with increasing volume radially from the center of the first joint. To compute the results, the Matlab Robotic Toolbox was used [34].

### 2.4.1. Manipulability

A manipulability index was introduced by Yoshikawa [89] to measure the distance to singular configurations. The approach is based on evaluating the manipulability ellipsoid that is spanned by the singular values of a manipulator Jacobian. The manipulability index is given as:

$$\mu = \sqrt{\det(J(\theta)J^T(\theta))} = \sigma_1\sigma_2\dots\sigma_m \quad (16)$$

Manipulability can be used to determine the manipulator singularity and optimal configurations in which to perform certain tasks. In some cases, it is desirable to

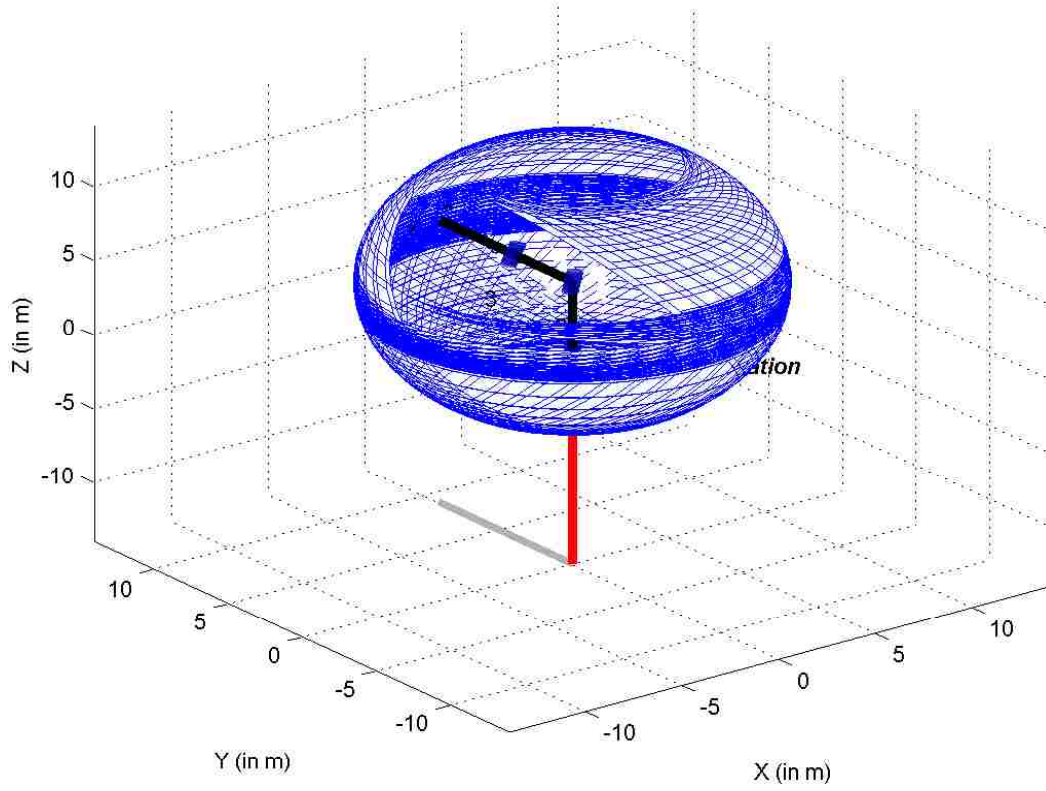


FIGURE 2.4. Workspace of RRR Configuration with four different twist angle values  $\pi/16$ ,  $\pi/8$ ,  $\pi/4$ , and  $\pi/2$ .

perform a task in a configuration for which the end effector has maximum manipulability. For the ABB manipulator robot with RRR kinematic structure and D–H parameters given in Table 2.3, the Jacobian matrix is calculated in Equation (17), where  $S_{23} = \sin(\theta_2 + \theta_3)$ . Then, the manipulability index is calculated and given in Equation (18).

TABLE 2.3. D–H parameters of the ABB manipulator robot, source; Spong, [80].

Link	$\theta_i$	$d_i$	$a_i$	$\alpha_i$
1	$\theta_1$	$d_1$	0	$\pi/2$
2	$\theta_2$	0	$a_2$	$0^0$
3	$\theta_3$	0	$a_3$	$0^0$

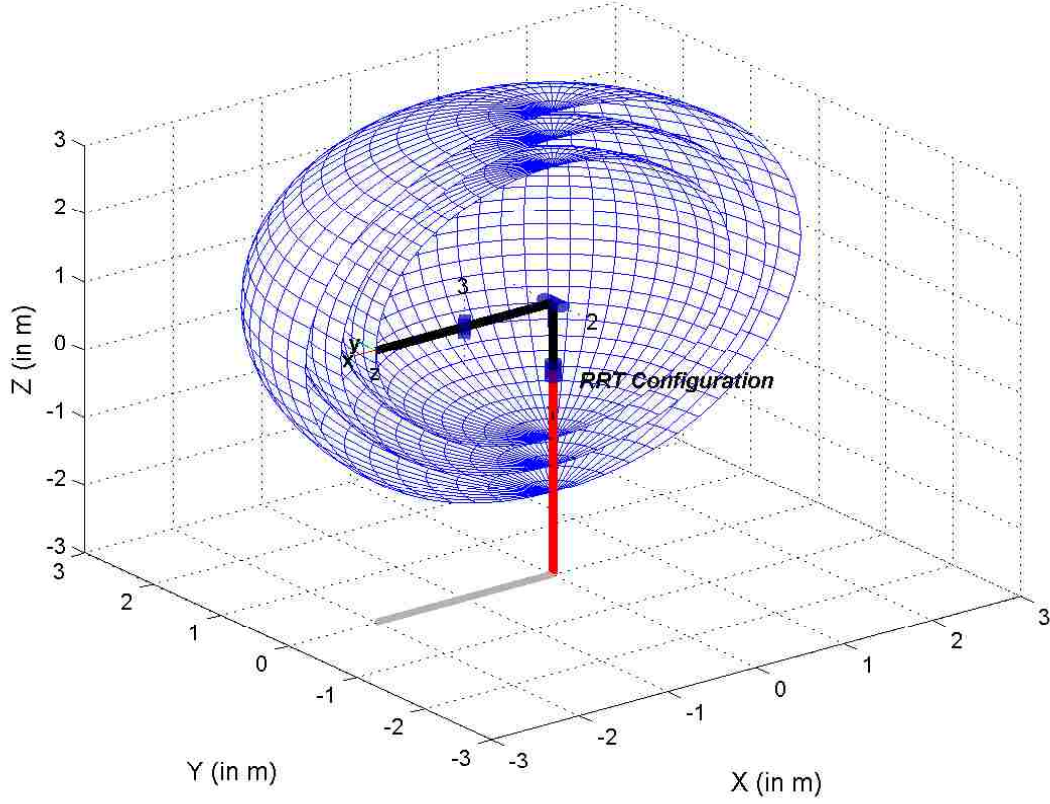


FIGURE 2.5. Workspace of RRT Configuration with different link lengths 0.15, 0.3 and 0.45 m.

$$J(\theta) = \begin{bmatrix} -a_2 S_1 C_2 - a_3 S_1 C_{23} & -a_2 S_2 C_1 - a_3 S_{23} C_1 & a_3 C_1 S_{23} \\ a_2 C_1 C_2 + a_3 C_1 C_{23} & -a_2 S_1 S_2 - a_3 S_1 S_{23} & a_3 S_1 S_{23} \\ 0 & a_2 C_2 + a_3 C_{23} & a_3 C_{23} \end{bmatrix} \quad (17)$$

$$\mu = |\lambda_1 \lambda_2 \dots \lambda_m| = |\det J| = a_2 a_3 |S_3| (a_2 |C_2| + a_3 |C_{23}|) \quad (18)$$

The resulting manipulability index, shown in Figure 2.6, is a function of  $\theta_2$  and  $\theta_3$  and the singularity configuration occurs when  $S_3 = 0$  and  $a_2 C_2 + a_3 C_{23} = 0$ . As a result, two types of singularities are present, at any  $\theta_1$ , for pair  $(\theta_2, \theta_3)$ :  $(\pm\pi/2, 0)$ ,  $(\pm\pi/2, \pm\pi)$  and for all  $\theta_3 = 0$  or  $\theta_3 = \pm\pi$ . The optimal value of the manipulability index  $\mu = 0.079$  was resulted with the associated  $(\theta_2, \theta_3) = (-159.37^\circ, -71.70^\circ)$  and also the singular configurations already found analytically. For the Stanford manipulator with RRT kinematic structure and D-H parameters given in Table 2.4, the Jacobian matrix is computed in Equation (19).

TABLE 2.4. D–H parameters of the Stanford manipulator robot, source; Spong, [80].

Link	$\theta_i$	$d_i$	$a_i$	$\alpha_i$
1	$\theta_1$	0	0	$-\pi/2$
2	$\theta_2$	$d_2$	0	$\pi/2$
3	0	$d_3$	0	$0^0$

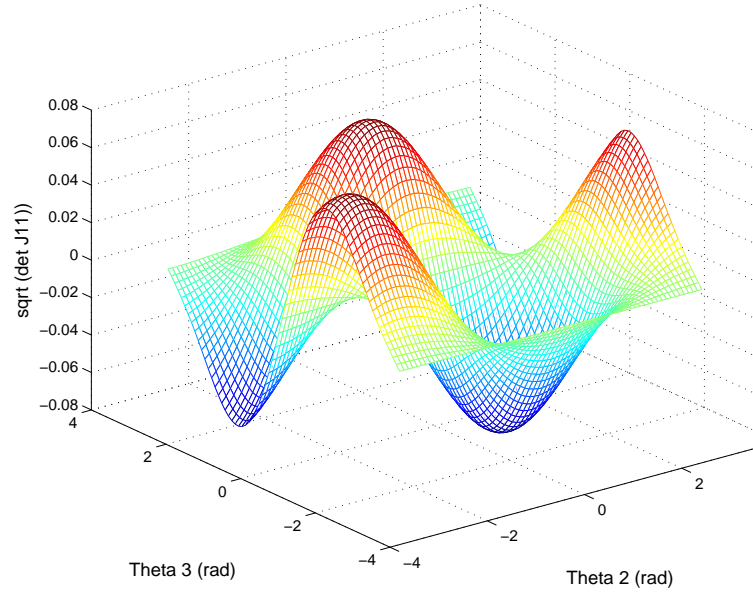


FIGURE 2.6. 3D profile of the manipulability index measure of RRR configuration.

$$J(\theta) = \begin{bmatrix} -d_2 C_1 - d_3 S_1 S_2 & d_3 C_1 C_2 & C_1 S_2 \\ -d_2 S_1 + d_3 C_1 S_2 & d_3 S_1 C_2 & S_1 S_2 \\ 0 & d_3 S_2 & C_3 \end{bmatrix} \quad (19)$$

The manipulability index is calculated as:

$$\mu = |\det J(\theta)| = -d_2 d_3 C_1 |S_1| - d_3^2 |S_2^3| - d_3^2 |S_2| C_2^2 + d_2 d_3 |S_1| C_1 \quad (20)$$

As a result, the singularity is present, at any  $\theta_1$ , for  $(\theta_2 = 0, \pm\pi)$ . Figure 2.7 shows the singularity and optimal manipulability index  $\mu = 0.999$  with configuration  $(\theta_2, \theta_3) = (-71.7^\circ, -92.33^\circ)$ . A reconfigurable robot manipulator spans the union of at least two configurations RRR and RRT and hence it has the ability to perform

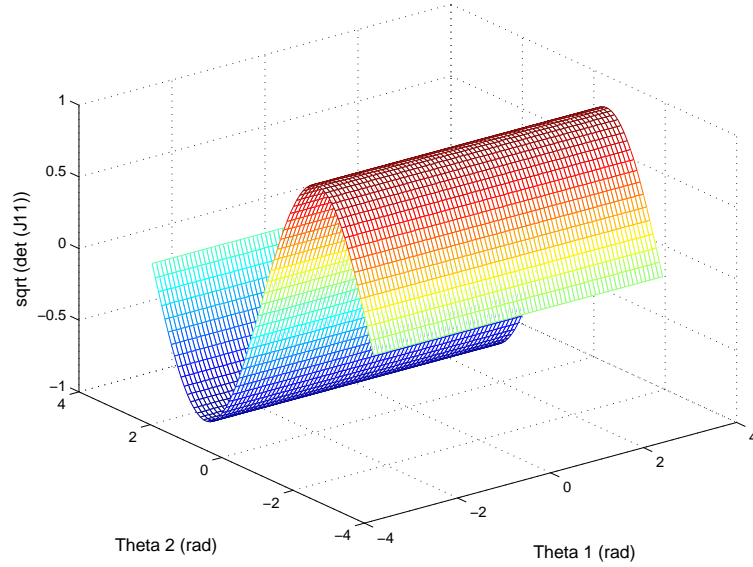


FIGURE 2.7. 3D profile of the manipulability index measure of RRT configuration.

a wider range of tasks within the workspace by avoiding the singular configurations.

## 2.5. Jacobian Condition and Manipulability

The Jacobian linear system  $v = J(q)\dot{q}$  maps the joint velocities to end-effector Cartesian velocity. The Jacobian is regarded as scaling the input  $q$  to yield the output. In a multidimensional case, the equivalent concept is to characterize the output in terms of an input that has unit norm as follows [80]:

$$\|\dot{q}\| = \dot{q}_1^2 + \dot{q}_2^2 + \dots + \dot{q}_n^2 \leq 1 \quad (21)$$

If the input (joint velocity) vector has unit norm, then the output (end-effector velocity) will be positioned within an ellipsoid and defined :

$$\begin{aligned} \|\dot{q}\|^2 &= \dot{q}^T \dot{q} \\ &= (J^+ v)^T J^+ v \\ &= v^T (J J^T)^{-1} v \end{aligned} \quad (22)$$

where  $J^+$  is the Jacobian pseudo inverse and the derivation of Equation (22) in Appendix (A). In particular, if the manipulator Jacobian is full rank, then

Equation (22) defines  $m$ -dimensional ellipsoid known as the *manipulability ellipsoid*. Replacing the Jacobian by its Singular Value Decomposition (SVD),  $J = U\Sigma V^T$

$$\begin{aligned} v^T(JJ^T)^{-1}v &= v^T [(U\Sigma V^T)(U\Sigma V^T)^T]^{-1}v \\ v^T(JJ^T)^{-1}v &= v^T [U\Sigma^2U^T]^{-1}v \\ v^T(JJ^T)^{-1}v &= (v^TU)\Sigma^{-2}(U^Tv) \end{aligned}$$

where  $U \in \mathbb{R}^{m \times m}$  and  $V \in \mathbb{R}^{n \times n}$  are orthogonal matrices and the singular value  $\Sigma \in \mathbb{R}^{m \times n}$  is given as follows:

$$\Sigma_m^{-2} = \begin{bmatrix} \sigma_1^{-2} & & & & \\ & \sigma_2^{-2} & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \sigma_m^{-2} \end{bmatrix}$$

Substitute:  $w = U^Tv$ , yields:

$$w^T \Sigma_m^{-2} w = \sum \frac{w_i^2}{\sigma_i^2} \leq 1 \quad (23)$$

Equation (23) represents a surface of 3-dimensional ellipsoid of the end-effector Cartesian velocity space. If this ellipsoid is close to spherical, its radii are of the same order value, the end-effector can achieve arbitrary Cartesian velocity. But when one or more radii are very small this indicates that the end-effector cannot achieve velocity in the directions corresponding to those small radii. Figure 2.8 shows the RRR manipulator (Elbow Configuration) with three different twist angle values ( $\pi/16, \pi/6, \pi/2$ ) of the second joint. End effector linear velocities with twist angle  $\pi/2$  were represented by an ellipsoid of almost equal radii in the  $y$  and  $z$  directions. This indicates that the end-effector can achieve higher Cartesian velocities in the  $y$  and  $z$ -directions than in the  $x$ -direction. While Cartesian velocities with twist angle ( $\pi/6$  and  $\pi/16$ ) were represented with ellipses indicating limited velocities in the  $y$  and  $z$  directions. This result shows that the optimal configuration to obtain maximum Cartesian velocities in the  $y$  and  $z$  directions is

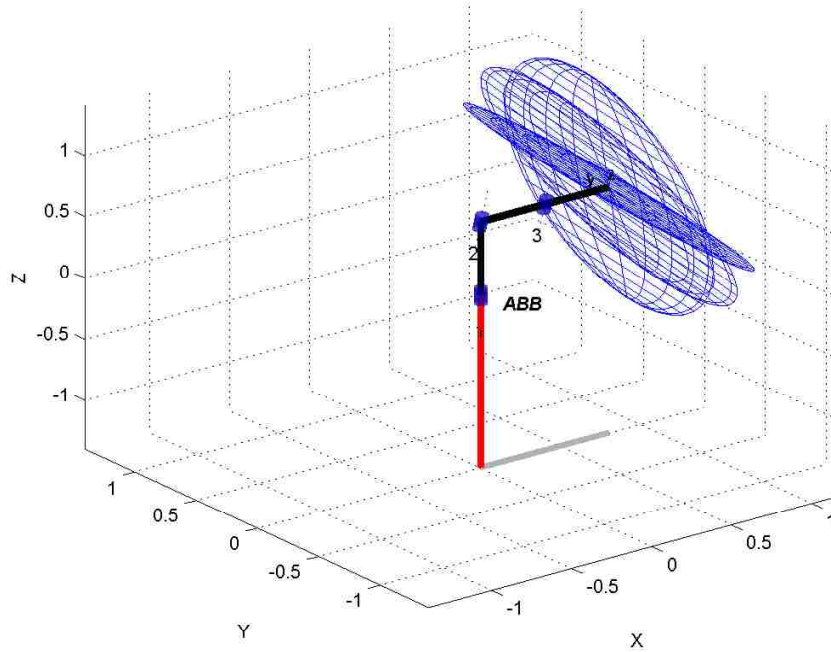


FIGURE 2.8. Cartesian velocity ellipsoid of RRR configuration with different twist angle values of  $(\pi/16, \pi/6, \pi/2)$  degrees.

when twist angle value of  $\pi/2$ . Figure 2.9 illustrates the Cartesian velocities of the end-effector of the RRT manipulator (Stanford Configuration). The velocity ellipsoid disks are calculated with second joint prismatic increment length values of 0.1, 0.2 and 0.3 m.

### 2.5.1. Joint-Space and Cartesian Trajectories

One of the most common requirements in robotics is to move the end-effector smoothly from pose A to pose B. Two approaches to generate trajectories are analyzed: straight lines in joint-space and straight lines in Cartesian space. In joint-space motion, it is considered that the motion of 6 axes (3-DOF links and 3-DOF spherical wrist) ABB robot is moved in straight line from initial y-axis (0.2 m) to final y-axis (-0.2 m) and the z-axis of the tool is rotated with  $(\pi/2)$  degrees. Thus, the end-effector motion lies in  $xy$ -plane with the approach vector oriented downwards. As shown in Figure 2.10 (left), the joint angles of the shoulder  $\theta_2$  and elbow  $\theta_3$  are constant values while the base joint angle changes its value with time to move the end-effector from pose A to B. The spherical wrist angles are changes to orient the end-effector approach vector downwards. The Cartesian motion of

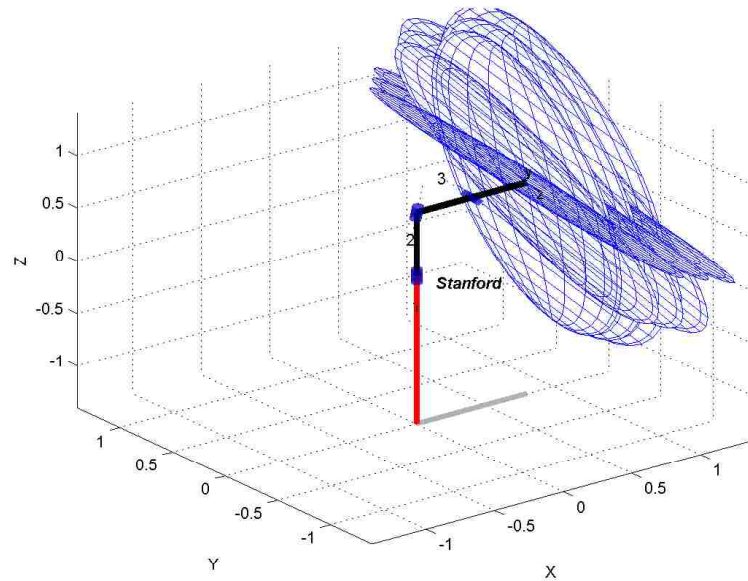


FIGURE 2.9. Cartesian velocity ellipsoid of RRT configuration with different prismatic lengths of 0.1, 0.2 and 0.3 m.

the end-motion in  $x$ ,  $y$  and  $z$  directions are shown in Figure 2.10 (right). The path of the end-effector in the  $xy$ -plane is shown in Figure 2.11 (left) and it is obviously not a straight line. The reason is that as the robot rotates around its base joint the end-effector will follow a circular path between the initial and final poses. The orientation of the end-effector, in roll-pitch-yaw angles form, is shown in Figure 2.11 (right), in which the roll angle varies from  $\pi$  to  $\pi/2$ .

On the other hand, a straight line in Cartesian space is needed in many applications, which is known as Cartesian motion. Following a straight line path, the ABB robot joint angles are shown in Figure 2.12 along with the path of the end-effector in Cartesian space and  $xy$ -plane. The first difference when comparing with the joint motion is that the end-effector in Cartesian motion follows a straight line in the  $xy$ -plane as shown in Figure 2.13 (left). The other difference is that the position and orientation of the end-effector varies linearly with the time as shown in the right of Figures 2.12 and 2.13.

### 2.5.2. Motion Through a Singularity

In the following, a Cartesian motion trajectory that moves through the ABB robot singularity is intentionally chosen. As shown in Figure 2.14 (left), the rate of change of the wrist joint angles  $\theta_4$  and  $\theta_6$  has become very high at time  $t \approx 0.7s$ .



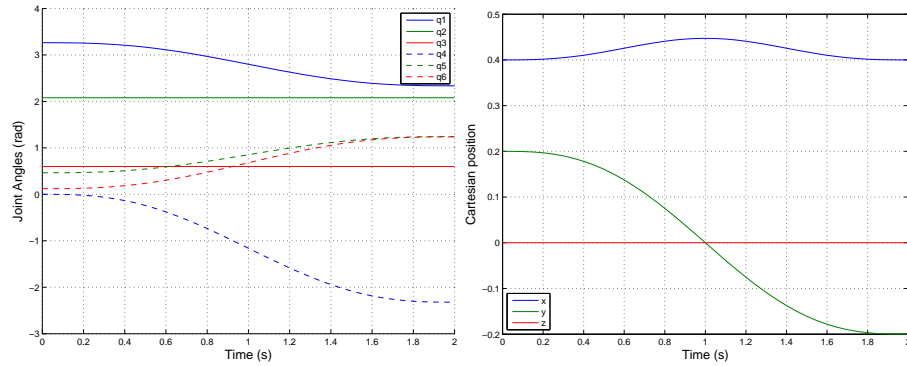


FIGURE 2.10. Joint angles during joint-space motion (left). Cartesian coordinates of the end-effector in  $x$ ,  $y$  and  $z$  directions (right).

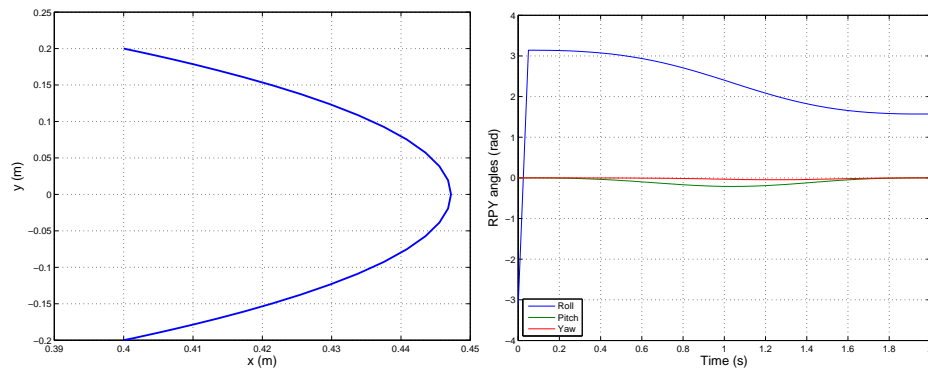


FIGURE 2.11. Cartesian position locus in the  $xy$ -plane (left). Euler angles roll-pitch-yaw versus time (right).

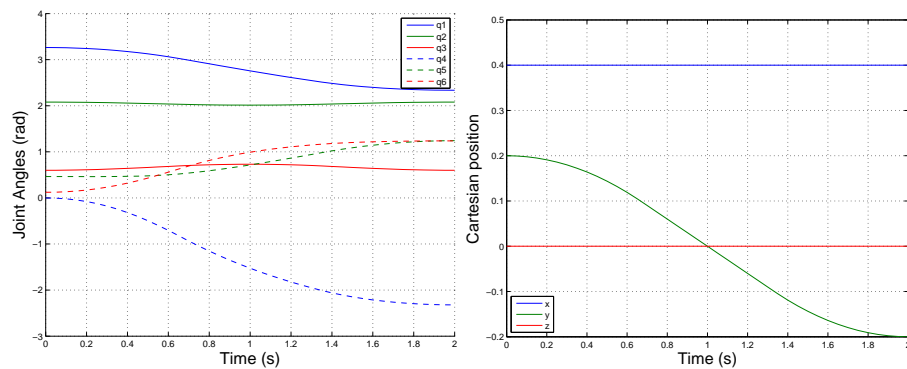


FIGURE 2.12. Joint angles during Cartesian motion (left). Cartesian coordinates of the end-effector in  $x$ ,  $y$  and  $z$  directions (right).

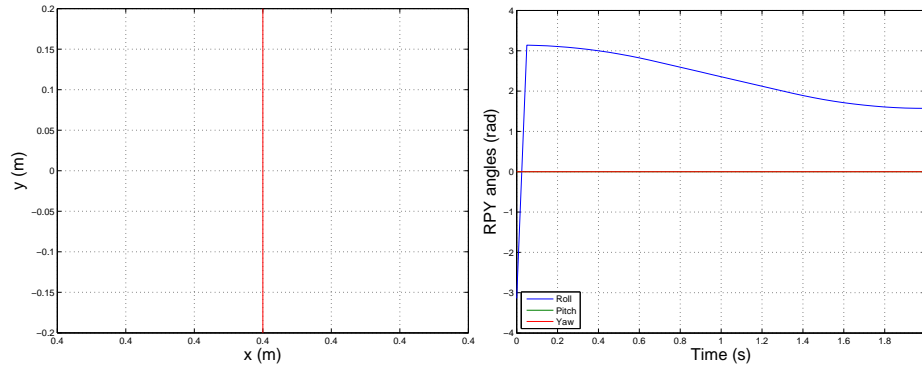


FIGURE 2.13. Cartesian position locus in the  $xy$ -plane (left). Euler angles roll-pitch-yaw versus time (right).

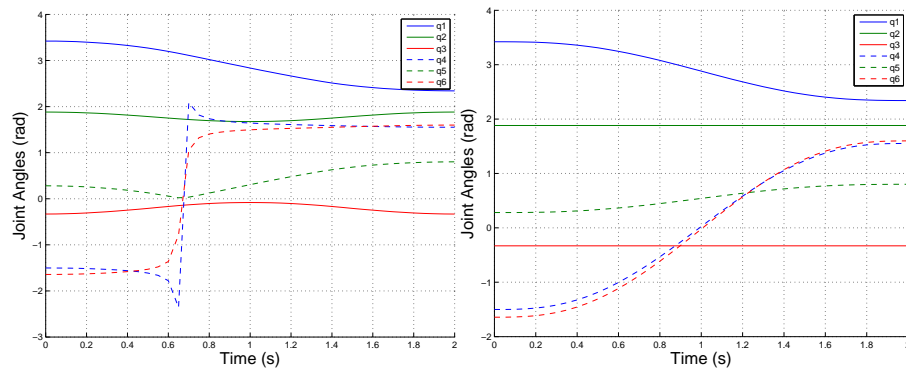


FIGURE 2.14. Joint angles follow Cartesian path through a wrist singularity (left), joint angles follows joint-space path (right).

The reason is that  $\theta_5$  has become almost zero which means the rotational axes of  $\theta_4$  and  $\theta_6$  are almost aligned, resulting in a singular robot position. The joint alignment means that the robot has lost one degree of freedom and is now 5-axis robot. The joint-space motion in Figure 2.14 (right) has not shown any unusual joint rate change since it does not require the solution of the inverse kinematics. The manipulability measure Equation (16) for this path is plotted in Figure 2.15 and shows that manipulability is almost zero around the time of rapid wrist joint motion.

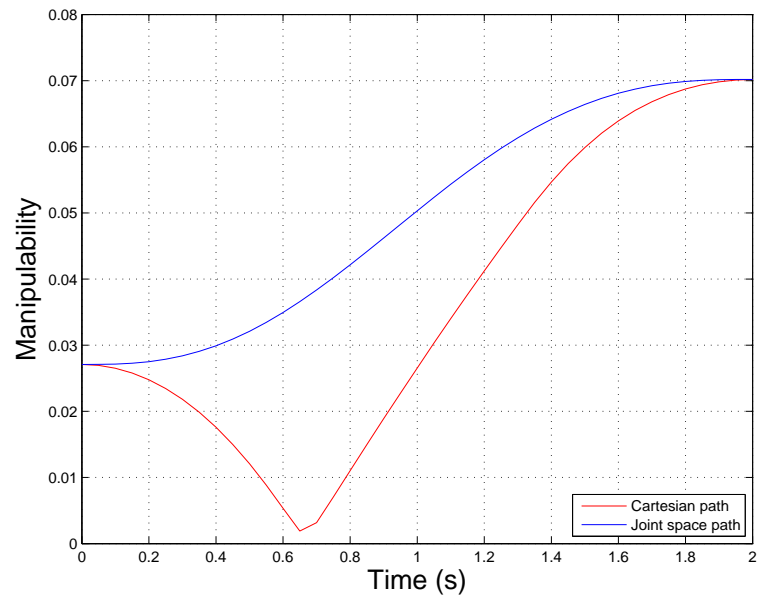


FIGURE 2.15. Manipulability measure of Cartesian and joint-space paths.

## CHAPTER 3

### Reconfigurable Robot Dynamics

In this chapter, the dynamics of a reconfigurable open kinematic robot is developed and analyzed following the method given by Djuric, Al Saidi and ElMaraghy [38]. The motor that actuates the  $j^{th}$  hybrid joint exerts a torque that causes the outward link,  $j$ , to accelerate and also exerts a reaction on the inwards link  $j - 1$ . Gravity acting on the outward links  $j$  to  $n$  exert a weight force, and rotating links also exert gyroscopic forces on each other. The resulting inertia from the motor exertion is a function of the configuration of the outward links.

#### 3.1. Reconfigurable Global Dynamic Model

The Global Dynamic Model ( $n$ -GDM) includes the link's masses,  $m_1, m_2, \dots, m_n$  and center of masses,  $P_{C1}, P_{C2}, \dots, P_{Cn}$ . The center of mass  $P_{C1}$  is between joint 1 and joint 2,  $P_{C2}$  is between joint 2 and joint 3, ..., etc., up to the last link. The coordinates of any center of mass  $P_{Ci}$  is defined relative to the joint  $i + 1$  frame:  $(x_{i+1}, y_{i+1}, z_{i+1})$ . For the  $n$ -GDM model, which includes  $n$  reconfigurable joints, the center of mass can be in 24 different places between any two successive joints. This means that for each selection of the  $z_{i+1}$  coordinate frames, there are four possible centers of mass:  $P_{Ci}^1, P_{Ci}^2, P_{Ci}^3, P_{Ci}^4$ . To find the center of mass of each link for the  $n$ -GDM model, all possible coordinate frames are included and for all joints. The analysis was done for the center of mass between joint 1 and joint 2. The same procedure can be applied to the other  $n - 1$  center of mass. A selection of the  $z_0^1$  axis can support four different  $x$ -axis:  $x_0^{11}, x_0^{12}, x_0^{13}, x_0^{14}$ . For joint 2, there are four different combinations of  $x_1$ :  $x_1^{11}, x_1^{12}, x_1^{13}, x_1^{14}$ . Each of the four  $x$ -combinations has four more combinations of the joint 2 coordinate frame. In total there are sixteen different combinations of the first coordinate frame. By observing the coordinates of each center of mass,  $P_{C1}^1, P_{C1}^2, P_{C1}^3$ , and  $P_{C1}^4$ , relative to

all sixteen possible combinations, the general unified solution has been developed and presented in Equation (24):

$$P_{Ci} = \begin{bmatrix} -\cos(\theta_{DH_i}) \frac{a_i}{2} \\ K_{Ci} \sin(\theta_{DH_i}) \frac{a_i}{2} - K_{Si} \frac{d_{DH_i}}{2} \\ -K_{Ci} \sin(\theta_{DH_i}) \frac{a_i}{2} - K_{Ci} \frac{d_{DH_i}}{2} \end{bmatrix}, \quad i = 1, 2, \dots, n \quad (24)$$

The moment of inertia about the center of mass for each link is related to its shape and dimension. The inertia tensor matrix of the  $n$ -GDM is shown in Equation (25):

$$I_i = \begin{bmatrix} I_{xi} & 0 & 0 \\ 0 & I_{yi} & 0 \\ 0 & 0 & I_{zi} \end{bmatrix}, \quad i = 1, 2, \dots, n \quad (25)$$

The  $n$ -GDM model is derived and presented in state space form using the aforementioned information. The equations of motion for an  $n$ -DOF manipulator are given as follows:

$$M(q)\ddot{q} + B(q)[\dot{q}\dot{q}] + C(\dot{q})[\dot{q}^2] + G(q) = \tau \quad (26)$$

Matrix  $M$  is the  $n \times n$  symmetric positive definite inertia matrix.  $B$  is the  $n \times \sum_{i=1}^{n-1} (n-i)$ , Coriolis torque matrix.  $C$  is the  $n \times n$  centrifugal torque matrix and  $G$  is the  $n \times 1$  gravity torque vector. The vectors  $\dot{q}, \ddot{q}$  are the vectors of joint velocity and acceleration, respectively. The velocity product vector  $[\dot{q}\dot{q}]$  is given as  $[\dot{q}\dot{q}] = [\dot{q}_1\dot{q}_2 \quad \dot{q}_1\dot{q}_3 \dots \dot{q}_1\dot{q}_n \dots \dot{q}_{n-1}\dot{q}_n]$  and  $[\dot{q}^2] = [\dot{q}_1^2 \quad \dot{q}_2^2 \dots \dot{q}_n^2]$  is the vector of squared velocity. The  $n$ -GDM model is calculated using the RNE (Recursive Newton-Euler) algorithm which is developed and can be applied for any reconfigurable open kinematic manipulator.

### 3.2. Forward Computation for Reconfigurable Robot

Using D-H parameters in Table 2.1, the general homogeneous transformation matrix is expressed in Equation (8). The upper  $3 \times 3$  sub-matrices of the each

homogeneous transformation matrix represent the rotational matrix for each joint, and they are presented as follows:

$${}^{i-1}R_i = \begin{bmatrix} \cos(R_i\theta_i + T_i\theta_{DH_i}) & -K_{ci} \sin(R_i\theta_i + T_i\theta_{DH_i}) & K_{si} \sin(R_i\theta_i + T_i\theta_{DH_i}) \\ \sin(R_i\theta_i + T_i\theta_{DH_i}) & K_{ci} \cos(R_i\theta_i + T_i\theta_{DH_i}) & -K_{si} \cos(R_i\theta_i + T_i\theta_{DH_i}) \\ 0 & K_{si} & K_{ci} \end{bmatrix},$$

$$i = 1, 2, \dots, n \quad (27)$$

The next step is to find the transpose of all rotational matrices  $({}^{i-1}R_i)^T$ , as shown in Equation (28):

$$({}^0R_1)^T = \begin{bmatrix} \cos(R_1\theta_1 + T_1\theta_{DH1}) & \sin(R_1\theta_1 + T_1\theta_{DH1}) & 0 \\ -K_{c1} \sin(R_1\theta_1 + T_1\theta_{DH1}) & K_{c1} \cos(R_1\theta_1 + T_1\theta_{DH1}) & K_{s1} \\ K_{s1} \sin(R_1\theta_1 + T_1\theta_{DH1}) & -K_{s1} \cos(R_1\theta_1 + T_1\theta_{DH1}) & K_{c1} \end{bmatrix},$$

$$i = 1, 2, \dots, n \quad (28)$$

The upper right  $3 \times 1$  sub-matrix for each homogeneous transformation matrix represent the position vectors for each joint and are stated in Equation (29):

$${}^{i-1}P_i = \begin{bmatrix} a_i \cos(R_i\theta_i + T_i\theta_{DH_i}) \\ a_i \sin(R_i\theta_i + T_i\theta_{DH_i}) \\ R_i d_{DH_i} + T_i d_i \end{bmatrix}, \quad i = 1, 2, \dots, n \quad (29)$$

The linear and angular velocity vectors and acceleration vectors for  $n$  joints are presented in Equation (30):

$${}^{i-1}\dot{P}_i = \begin{bmatrix} 0 \\ 0 \\ \dot{d}_i \end{bmatrix}, \quad {}^{i-1}\dot{\theta}_i = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_i \end{bmatrix}, \quad {}^{i-1}\ddot{P}_i = \begin{bmatrix} 0 \\ 0 \\ \ddot{d}_i \end{bmatrix}, \quad {}^{i-1}\ddot{\theta}_i = \begin{bmatrix} 0 \\ 0 \\ \ddot{\theta}_i \end{bmatrix}, \quad i = 1, 2, \dots, n \quad (30)$$

Using the rotation matrices and angular velocity vectors, the angular and linear velocities for rotational and translational joints can be calculated from Equations (31) and (32), respectively:

$${}^i({}^0w_i) = {}^iR_{i-1} \left[ {}^{i-1}({}^0w_{i-1}) + R_i({}^{i-1}\dot{\theta}_i) \right], \quad i = 1, 2, \dots, n \quad (31)$$

$$\begin{aligned} {}^i({}^0V_i) &= {}^iR_{i-1} {}^{i-1}({}^0V_{i-1}) + R_i [{}^i({}^0w_i) \times {}^iR_{i-1} {}^{i-1}P_i] + \\ &R_i \left\{ {}^iR_{i-1} \left[ {}^{i-1}\dot{P}_i + {}^{i-1}({}^0w_{i-1}) \times {}^{i-1}P_i \right] \right\}, \quad i = 1, 2, \dots, n \end{aligned} \quad (32)$$

The linear and angular acceleration for two joints are calculated from Equations (33) and (34) respectively:

$${}^i({}^0\alpha_i) = {}^iR_{i-1} \left\{ {}^{i-1}({}^0\alpha_{i-1}) + R_i \left[ {}^{i-1}({}^0w_{i-1}) \times {}^{i-1}\dot{\theta}_i + {}^{i-1}\ddot{\theta}_i \right] \right\}, \quad i = 1, 2, \dots, n \quad (33)$$

$$\begin{aligned} {}^i({}^0a_i) &= {}^iR_{i-1} {}^{i-1}({}^0a_{i-1}) + {}^{i-1}({}^0\alpha_{i-1}) \times {}^{i-1}P_i + {}^{i-1}({}^0w_{i-1}) \times ({}^{i-1}({}^0w_{i-1}) \times {}^{i-1}P_i) + \\ &R_i \left[ 2 \quad {}^{i-1}({}^0w_{i-1}) \times ({}^{i-1}\dot{\theta}_i \times {}^{i-1}P_i) + {}^{i-1}\ddot{\theta}_i \times {}^{i-1}P_i + {}^{i-1}w_i \times ({}^{i-1}\dot{\theta}_i \times {}^{i-1}P_i) \right] + \\ &T_i \left[ 2 \quad {}^{i-1}({}^0w_{i-1}) \times {}^{i-1}\dot{P}_i + {}^{i-1}\ddot{P}_i \right], \quad i = 1, 2, \dots, n \end{aligned} \quad (34)$$

The linear acceleration of the center of mass is calculated from Equation (35):

$${}^i({}^0a_{ci}) = {}^i({}^0a_i) + {}^i({}^0\alpha_i) \times {}^i({}^iP_{ci}) + {}^i({}^0w_i) \times [{}^i({}^0w_i) \times {}^i({}^iP_{ci})], \quad i = 1, 2, \dots, n \quad (35)$$

The calculation of the radial distances to center of mass  $P_{C_i}$  is already given in Equation (24).

### 3.3. Backward Computation of Forces and Moments

The general gravity vectors for the  $n$ -GKM model are expressed:

$${}^i g = {}^iR_0 {}^0g, \quad i = 1, 2, \dots, n \quad (36)$$

The gravity vector projection is expressed in  $x$ ,  $y$  and  $z$  coordinates:

$${}^0g = \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} \quad (37)$$

where  $g = 9.81 \text{ m/s}^2$ . The parameters  $g_x$ ,  $g_y$ , and  $g_z$  are used to define the gravity vector elements. These parameters depend on the selected kinematic configuration. Once the velocities and accelerations of the links are found, the joint forces and moments can be computed for one link at a time starting from the end-effector and ending at the base link. It is assumed that there is no load at the end-effector, therefore,  ${}^n({}^n f_{Tool}) = 0$ , and  ${}^n({}^n n_{Tool}) = 0$ .

$${}^i(f_i) = -m_i {}^i({}^0 a_{ci}), \quad i = 1, 2, \dots, n \quad (38)$$

$${}^i(n_i) = -{}^i I_i {}^i({}^0 \alpha_{ci}) - {}^i({}^0 w_{ci}) \times [{}^i I_i {}^i({}^0 w_{ci})], \quad i = 1, 2, \dots, n \quad (39)$$

The force and moment balance equations about the center of mass of  $i^{th}$  link in recursive form can be written as:

$${}^i(f_i) + {}^i({}^{i-1} f_i) - {}^i({}^i f_{i+1}) + m_i {}^i g = 0, \quad i = 1, 2, \dots, n \quad (40)$$

$$\begin{aligned} {}^i(n_i) &= {}^i({}^{i-1} n_i) - {}^i({}^i n_{i+1}) - [{}^i({}^{i-1} P_i) + {}^i({}^i P_{Ci})] \times {}^i({}^{i-1} f_i) + {}^i({}^i P_{Ci}) \times {}^i({}^i f_{i+1}) \\ &= 0, \quad i = 1, 2, \dots, n \end{aligned} \quad (41)$$

$$\begin{aligned} {}^i(n_i) &= {}^i({}^{i-1} n_i) - {}^i({}^i n_{i+1}) - [{}^i({}^{i-1} P_i) + {}^i({}^i P_{Ci})] \times {}^i({}^{i-1} f_i) + {}^i({}^i P_{Ci}) \times {}^i({}^i f_{i+1}) \\ &= 0, \quad i = 1, 2, \dots, n \end{aligned} \quad (42)$$

Once the reaction forces and moments are computed in the  $i^{th}$  link frame, they are converted into the  $(i-1)^{th}$  link frame by the following equations:

$${}^i({}^{i-1} f_i) = {}^{i-1} R_i {}^i({}^{i-1} f_i), \quad i = 1, 2, \dots, n \quad (43)$$

$${}^i({}^{i-1} n_i) = {}^{i-1} R_i {}^i({}^{i-1} n_i), \quad i = 1, 2, \dots, n \quad (44)$$



Torques  $\tau_i$  and forces  $f_i$  are obtained by projecting the moments and forces onto their corresponding joint axes respectively:

$$\tau_i = {}^i({}^{i-1}n_i)^T {}^{i-1}Z_{i-1}, \quad i = 1, 2, \dots, n \quad (45)$$

$$f_i = {}^i({}^{i-1}f_i)^T {}^{i-1}Z_{i-1}, \quad i = 1, 2, \dots, n \quad (46)$$

The RNE procedure produced the final  $n$  expressions of the actuators' torques and forces  $\tau_i$ ,  $i = 1, 2, \dots, n$ . Each of these  $n$  equations contains the sum of products of the matrices' elements  $M, B, C, G$ , and trigonometric terms. To be able to get a dynamic equation in the form of Equation (26), each matrix was generated i.e.  $M, B, C$ , and  $G$ , and their elements were calculated:

$$m_{11}, m_{12}, \dots, m_{1n}, \dots, m_{nn}, b_{112}, b_{113}, \dots, b_{1(n-1)n}, \dots, b_{n(n-1)n}, \\ c_{11}, c_{12}, \dots, c_{1n}, \dots, c_{nn}, g_1, g_2, \dots, g_n.$$

To avoid complications when factoring out each element in each expression, the Automatic Separation Method (ASM) is used, which produces an automatic generation of matrix elements [38]. Using the Newton-Euler recursive method for calculating the forces and/or torques of the links for any open kinematic chain will result  $n$ -equations. Each equation is a solution for force and/or torque of the link, which includes four elements: the first one is related to inertia force/torque vector, the second one is related to Coriolis force/torque vector, the third one is related to centrifugal force/torque vector and the fourth one to gravity force/torque vector. These results will be coupled with the dynamics of different motors to form a block diagram for control purposes. The elements of the matrices of the inertia matrix, Coriolis matrix, centrifugal matrix and the gravity vector were calculated to construct a complete block diagram of a robot. For these calculation, the ASM method was used by starting the calculation of angular acceleration elements related the inertia, Coriolis centrifugal and gravity elements, Equations (53)-(60). This procedure has three steps. The first step is to simplify and organize the angular and linear velocity equations. To overcome these problems it is necessary to

implement the basic trigonometric rules expressed in Equations (47)-(49):

$$\begin{aligned}
& \sin(R_i\theta_i + T_i\theta_{DH_i})\cos(R_{i+1}\theta_{i+1} + T_{i+1}\theta_{DH(i+1)}) + \\
& K_{C_i}\cos(R_i\theta_i + T_i\theta_{DH_i})\sin(R_{i+1}\theta_{i+1} + T_{i+1}\theta_{DH(i+1)}) \quad (47) \\
= & \sin(R_i\theta_i + T_i\theta_{DH_i} + K_{C_i}R_{i+1}\theta_{i+1} + T_{i+1}\theta_{DH(i+1)}), \quad i = 1, 2, \dots, n
\end{aligned}$$

$$\begin{aligned}
& \cos(R_i\theta_i + T_i\theta_{DH_i})\cos(R_{i+1}\theta_{i+1} + T_{i+1}\theta_{DH(i+1)}) + \\
& K_{C_i}\sin(R_i\theta_i + T_i\theta_{DH_i})\cos(R_{i+1}\theta_{i+1} + T_{i+1}\theta_{DH(i+1)}) \quad (48) \\
= & \cos(R_i\theta_i + T_i\theta_{DH_i} + K_{C_i}R_{i+1}\theta_{i+1} + T_{i+1}\theta_{DH(i+1)}), \quad i = 1, 2, \dots, n
\end{aligned}$$

$$\sin^2(R_i\theta_i + T_i\theta_{DH_i}) + \cos^2(R_i\theta_i + T_i\theta_{DH_i}) = 1, \quad i = 1, 2, \dots, n \quad (49)$$

The reconfigurable parameters of the model are given in Table 3.1. This leads to

TABLE 3.1. Reconfiguration parameters values

$K_{S_1} = \sin(\pm 90^0) = \pm 1$	$K_{C_1} = \cos(\pm 180^0, 0^0) = \pm 1$
$K_{S_2} = \sin(\pm 90^0) = \pm 1$	$K_{C_2} = \cos(\pm 180^0, 0^0) = \pm 1$
$K_{S_3} = \sin(\pm 90^0) = \pm 1$	$K_{C_3} = \cos(\pm 190^0, 0^0) = \pm 1$

two more simplifications shown in Equations (50)-(51)

$$K_{C_i}^2 = 1, \quad K_{S_i}^2 = 1, \quad i = 1, 2, \dots, n \quad (50)$$

$$K_{C_i}^3 = 1, \quad K_{S_i}^3 = 1, \quad i = 1, 2, \dots, n \quad (51)$$

The second step consists of ordering parameters of each element of the equation, which will help to continue the calculation without much complexity. The form is given in Equation (52):

$$\text{element} = K_{S_i}K_{C_i}a_i d_i \sin(R_i\theta_i + T_i\theta_{DH_i})\cos(R_i\theta_i + T_i\theta_{DH_i}), \quad i = 1, 2, \dots, n \quad (52)$$

The third step is separation of elements, which is important for further calculations. This method is first applied to the angular acceleration  ${}^i({}^0\alpha_i)_i$  and linear acceleration of the center of mass  ${}^i({}^0a_i)$ , such that these expressions are written as a sum of separated elements, as a function of the parameters of the vectors

$[\ddot{q}], [\dot{q}\dot{q}], [\dot{q}^2]$ :

$${}^i(0)\alpha_i)_{i(\alpha_i)} = \begin{bmatrix} {}^i(0)\alpha_i)_{i(\alpha_i)1} & {}^i(0)\alpha_i)_{i(\alpha_i)2} & {}^i(0)\alpha_i)_{i(\alpha_i)3} \end{bmatrix}, \quad i = 1, 2, \dots, n \quad (53)$$

$${}^i(0)\alpha_i)_{i(\omega_i^2)} = \begin{bmatrix} {}^i(0)\alpha_i)_{i(\omega_i^2)1} & {}^i(0)\alpha_i)_{i(\omega_i^2)2} & {}^i(0)\alpha_i)_{i(\omega_i^2)3} \end{bmatrix}, \quad i = 1, 2, \dots, n \quad (54)$$

$${}^i(0)\alpha_i)_{i(\omega_{ij})} = \begin{bmatrix} {}^i(0)\alpha_i)_{i(\omega_{ij})1} & {}^i(0)\alpha_i)_{i(\omega_{ij})2} & {}^i(0)\alpha_i)_{i(\omega_{ij})3} \end{bmatrix}, \quad \begin{matrix} i = 1, 2, \dots, n, \\ j = 1, 2, \dots, n \end{matrix} \quad (55)$$

$${}^i(0)\alpha_i)_{i(\ddot{d}_i)} = \begin{bmatrix} {}^i(0)\alpha_i)_{i(\ddot{d}_i)1} & {}^i(0)\alpha_i)_{i(\ddot{d}_i)2} & {}^i(0)\alpha_i)_{i(\ddot{d}_i)3} \end{bmatrix}, \quad i = 1, 2, \dots, n \quad (56)$$

$${}^i(0)a_{ci})_{i(\alpha_i)} = \begin{bmatrix} {}^i(0)a_{ci})_{i(\alpha_i)1} & {}^i(0)a_{ci})_{i(\alpha_i)2} & {}^i(0)a_{ci})_{i(\alpha_i)3} \end{bmatrix}, \quad i = 1, 2, \dots, n \quad (57)$$

$${}^i(0)a_{ci})_{i(\omega_i^2)} = \begin{bmatrix} {}^i(0)a_{ci})_{i(\omega_i^2)1} & {}^i(0)a_{ci})_{i(\omega_i^2)2} & {}^i(0)a_{ci})_{i(\omega_i^2)3} \end{bmatrix}, \quad i = 1, 2, \dots, n \quad (58)$$

$${}^i(0)a_{ci})_{i(\omega_{ij})} = \begin{bmatrix} {}^i(0)a_{ci})_{i(\omega_{ij})1} & {}^i(0)a_{ci})_{i(\omega_{ij})2} & {}^i(0)a_{ci})_{i(\omega_{ij})3} \end{bmatrix}, \quad i = 1, 2, \dots, n \quad (59)$$

$${}^i(0)a_{ci})_{i(\ddot{d}_i)} = \begin{bmatrix} {}^i(0)a_{ci})_{i(\ddot{d}_i)1} & {}^i(0)a_{ci})_{i(\ddot{d}_i)2} & {}^i(0)a_{ci})_{i(\ddot{d}_i)3} \end{bmatrix}, \quad \begin{matrix} i = 1, 2, \dots, n, \\ j = 1, 2, \dots, n \end{matrix} \quad (60)$$

The calculation of the inverse dynamics can be done such that all elements of the matrices from Equation (26) can be automatically generated. Using Equations (47)-(60), the automatic generation of all matrix elements can be produced, as shown below:

$$m_{ij} = {}^i(R_i^{i-1}n_{i(\alpha_i)} + T_i^{i-1}f_{i(\alpha_i)} + T_i^{i-1}f_{i(\ddot{d}_i)})^T {}^{i-1}Z_{i-1}, \quad \begin{matrix} i = 1, 2, \dots, n, \\ j = 1, 2, \dots, n \end{matrix} \quad (61)$$

$$b_{ij} = {}^i(R_i^{i-1}n_{i(\omega_{ij})} + T_i^{i-1}f_{i(\omega_{ij})})^T {}^{i-1}Z_{i-1}, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, n \quad (62)$$

$$c_{ij} = {}^i(R_i^{i-1}n_{i(\omega_i^2)} + T_i^{i-1}f_{i(\omega_i^2)})^T {}^{i-1}Z_{i-1}, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, n \quad (63)$$

$$g_i = {}^i(R_i^{i-1}n_{i(GR_i)} + T_i^{i-1}f_{i(GR_i)})^T {}^{i-1}Z_{i-1}, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, n \quad (64)$$

### 3.4. Development of the 3-GDM Model

In this section, a 3 DOF General Dynamic Model (3-GDM) model is developed to represent the first three reconfigurable joints of any open kinematic chain

TABLE 3.2. Reconfigurable D–H parameters of the 3-GKM model

$i$	$d_i$	$\theta_i$	$a_i$	$\alpha_i$
1	$R_1 d_{DH1} + T_1 d_1$	$R_1 \theta_1 + T_1 \theta_{DH1}$	$a_1$	$0^0, \pm 180^0, \pm 90^0$
2	$R_2 d_{DH2} + T_2 d_2$	$R_2 \theta_2 + T_2 \theta_{DH2}$	$a_2$	$0^0, \pm 180^0, \pm 90^0$
3	$R_n d_{DHn} + T_n d_n$	$R_n \theta_n + T_n \theta_{DHn}$	$a_n$	$0^0, \pm 180^0, \pm 90^0$

manipulator, assuming a spherical wrist is attached to the end effector. The D–H parameters of the 3-GKM kinematic model are given in Table 3.2.

$$\begin{aligned}
 & \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + \begin{bmatrix} b_{112} & b_{113} & b_{123} \\ b_{212} & b_{213} & b_{223} \\ b_{312} & b_{313} & b_{323} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \dot{q}_2 \\ \dot{q}_1 \dot{q}_3 \\ \dot{q}_2 \dot{q}_3 \end{bmatrix} + \\
 & \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} \dot{q}_1^2 \\ \dot{q}_2^2 \\ \dot{q}_3^2 \end{bmatrix} + \begin{bmatrix} G_1 \\ G_2 \\ G_3 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \quad (65)
 \end{aligned}$$

The matrices ( $M, B, C, G$ ) of the dynamic system Equation (26) are calculated for the 3-GKM model using the Equations (61)-(64) from which each element of the four matrices was automatically generated forming Equation (65).

### 3.4.1. Model Validation

The validation of the 3-GKM model was done by using 2-DOF and 3-DOF configurations. The selected configurations (RR, RT, TR, and TT planar and SCARA) are used to illustrate the model general capabilities.

#### RR Planar Manipulator

The RR planar kinematic structure is graphically shown in Figure 3.1. The kinematic and dynamic parameters of the planar kinematic structure are shown in tables 3.3 and 3.4, respectively. Using these parameters and the 3-GDM model, the RR configuration results the dynamic Equations (66).

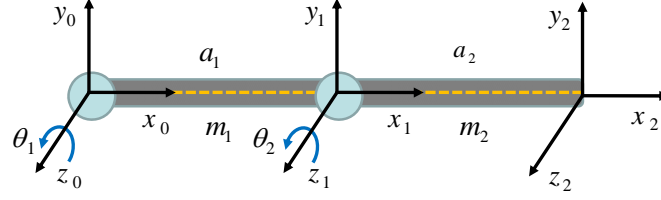


FIGURE 3.1. RR planar kinematic structure.

TABLE 3.3. Kinematic initial parameters of 2-DOF RR configuration

Configuration	$R_i$	$T_i$	$K_{Si}$	$K_{Ci}$	$d_{DH_i}$	$\theta_{DH_i}$
Joint 1	1	0	0	1	0	$0^0$
Joint 2	1	0	0	1	0	$0^0$

TABLE 3.4. Dynamic initial parameters of 2-DOF RR configuration

Configuration	$G_x$	$G_y$	$G_z$	$I_{x1}$	$I_{y1}$	$I_{z1}$	$I_{x2}$	$I_{y2}$	$I_{z2}$
RR Planar	0	-1	0	$I_{x1}$	$I_{y1}$	$I_{z1}$	$I_{x2}$	$I_{y2}$	$I_{z2}$

$$\begin{aligned}
& \begin{bmatrix} \frac{1}{4}(a_2^2 m_2 + 4a_2 m_2 \cos(\theta_2) a_1 + 4I_{z2} + m_1 a_1^2 + 4I_{z1}) & \frac{1}{4}(a_2^2 m_2 + 4a_2 m_2 \cos(\theta_2) a_1 + 4I_{z2}) \\ \frac{1}{4}(a_2^2 m_2 + 4a_2 m_2 \cos(\theta_2) a_1 + 4I_{z2}) & \frac{1}{4}(a_2^2 m_2 + 4a_2 m_2 \cos(\theta_2) a_1 + 4I_{z2}) \end{bmatrix} \\
& \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} \\
& \begin{bmatrix} -\frac{1}{2} a_1 \sin(\theta_2) m_2 w_1 w_2 a_2 \\ 0 \end{bmatrix} \dot{q}_1 \dot{q}_2 + \begin{bmatrix} 0 & -\frac{1}{2} a_1 \sin(\theta_2) m_2 w_2^2 a_2 \\ -\frac{1}{2} a_2 \sin(\theta_2) m_2 w_2^2 a_2 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1^2 \\ \dot{q}_2^2 \end{bmatrix} + \\
& \begin{bmatrix} \frac{1}{2} g (a_2 m_2 \cos(\theta_1 + \theta_2) + a_1 \sin(\theta_1) m_2 \sin(\theta_1 + \theta_2) + a_1 \cos(\theta_2) m_2 \cos(\theta_1 + \theta_2) + a_1 m_1 \cos(\theta_1)) \\ \frac{1}{2} a_2 m_2 \cos(\theta_1 + \theta_2) g \end{bmatrix} \\
& = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}
\end{aligned} \tag{66}$$

### RT Planar Manipulator

The RT planar kinematic structure is graphically presented in Figure 3.2. The

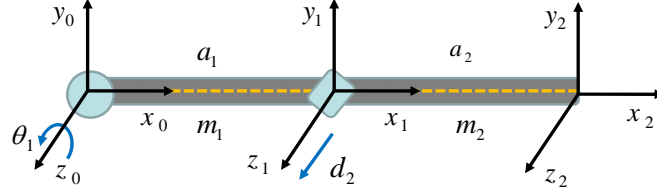


FIGURE 3.2. RT planar kinematic structure.

kinematic and dynamic parameters of the RT planar kinematic structure are shown in tables 3.5 and 3.6. Using these parameters and the 3-GDM model, the RT configuration results the dynamic equations (67).

TABLE 3.5. Kinematic initial parameters of 2-DOF RT configuration

Configuration	$R_i$	$T_i$	$K_{Si}$	$K_{Ci}$	$d_{DH_i}$	$\theta_{DH_i}$
Joint 1	1	0	0	1	0	$0^0$
Joint 2	0	1	0	1	0	$0^0$

TABLE 3.6. Dynamic initial parameters of 2-DOF RT configuration

Configuration	$G_x$	$G_y$	$G_z$	$I_{x1}$	$I_{y1}$	$I_{z1}$	$I_{x2}$	$I_{y2}$	$I_{z2}$
RT Planar	0	-1	0	$I_{x1}$	$I_{y1}$	$I_{z1}$	$I_{x2}$	$I_{y2}$	$I_{z2}$

$$\begin{aligned}
 & \begin{bmatrix} m_2 a_1 a_2 + \frac{1}{4} a_2^2 m_2 + I_{z2} \alpha_1 + m_2 a_1^2 + \frac{1}{4} a_1^2 m_2 + I_{z1} & 0 \\ 0 & m_2 \ddot{d}_2 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} \\
 & + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \dot{q}_1 \dot{q}_2 + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1^2 \\ \dot{q}_2^2 \end{bmatrix} + \begin{bmatrix} \frac{1}{4} \cos(\theta_1) g (a_2 m_2 + a_1 m_1) \\ 0 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (67)
 \end{aligned}$$

### TR Planar Manipulator

The TR planar kinematic structure is graphically presented in Figure 3.3. The kinematic and dynamic parameters of the TR planar kinematic structure are shown in tables 3.7 and 3.8. Using these parameters and the 3-GDM model, the TR configuration results the dynamic equations (68).

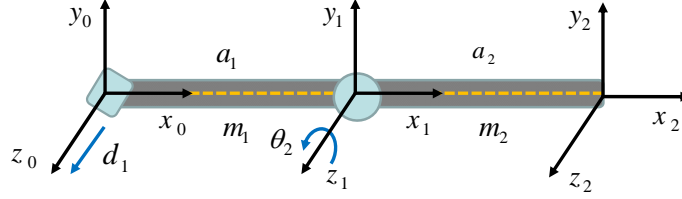


FIGURE 3.3. TR planar kinematic structure.

TABLE 3.7. Kinematic initial parameters of 2-DOF TR configuration

Configuration	$R_i$	$T_i$	$K_{Si}$	$K_{Ci}$	$d_{DH_i}$	$\theta_{DH_i}$
Joint 1	0	1	0	1	0	$0^0$
Joint 2	1	0	0	1	0	$0^0$

TABLE 3.8. Dynamic initial parameters of 2-DOF TR configuration

Configuration	$G_x$	$G_y$	$G_z$	$I_{x1}$	$I_{y1}$	$I_{z1}$	$I_{x2}$	$I_{y2}$	$I_{z2}$
TR Planar	0	-1	0	$I_{x1}$	$I_{y1}$	$I_{z1}$	$I_{x2}$	$I_{y2}$	$I_{z2}$

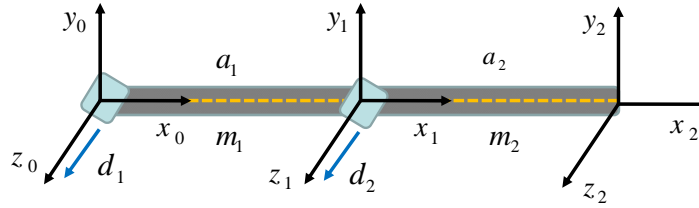


FIGURE 3.4. TT planar kinematic structure.

$$\begin{bmatrix} m_1 + m_2 & 0 \\ 0 & \frac{1}{4}a_2^2m_2 + I_{z2} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \dot{q}_1\dot{q}_2 + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1^2 \\ \dot{q}_2^2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{2}a_2m_2\cos(\theta_2)g \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (68)$$

### TT Planar Manipulator

The TT planar kinematic structure is graphically presented in Figure 3.4. The kinematic and dynamic parameters of the TT planar kinematic structure are shown in tables 3.9 and 3.10. Using these parameters and the 3-GDM model, the TT configuration results the following dynamic equations (69).

TABLE 3.9. Kinematic initial parameters of 2-DOF TT configuration

Configuration	$R_i$	$T_i$	$K_{S_i}$	$K_{C_i}$	$d_{DH_i}$	$\theta_{DH_i}$
Joint 1	0	1	0	1	0	$0^0$
Joint 2	0	1	0	1	0	$0^0$

TABLE 3.10. Dynamic initial parameters of 2-DOF TT configuration

Configuration	$G_x$	$G_y$	$G_z$	$I_{x1}$	$I_{y1}$	$I_{z1}$	$I_{x2}$	$I_{y2}$	$I_{z2}$
TT Planar	0	-1	0	$I_{x1}$	$I_{y1}$	$I_{z1}$	$I_{x2}$	$I_{y2}$	$I_{z2}$

TABLE 3.11. Kinematic initial parameters of 3-DOF SCARA configuration

Configuration	$R_i$	$T_i$	$K_{S_i}$	$K_{C_i}$	$d_{DH_i}$	$\theta_{DH_i}$
Joint 1	1	0	0	1	$d_{DH1}$	$0^0$
Joint 2	1	0	0	-1	0	$0^0$
Joint 3	0	1	0	1	$d_3$	$0^0$

$$\begin{bmatrix} m_1 + m_2 & m_2 \\ m_2 & m_2 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \dot{q}_1 \dot{q}_2 + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1^2 \\ \dot{q}_2^2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (69)$$

### 3-DOF Scara Kinematic Structure Manipulator

The validation of 3-GDM model is done using SCARA kinematic structure which is graphically shown in Figure 3.5. The kinematic and dynamic parameters of the SCARA kinematic structure are presented in tables 3.11 and 3.12. Using these parameters and the 3-GDM model, the Scara configuration results the dynamic equations (70). The dynamic model for the three-axis Scara robot in Equations (70) is only slightly more complex than the dynamic model for two-axis planar articulated robot. This is because a Scara robot is essentially a two-axis articulated robot used to establish the horizontal tool position, plus an orthogonal one-axis prismatic link used to establish vertical tool position.



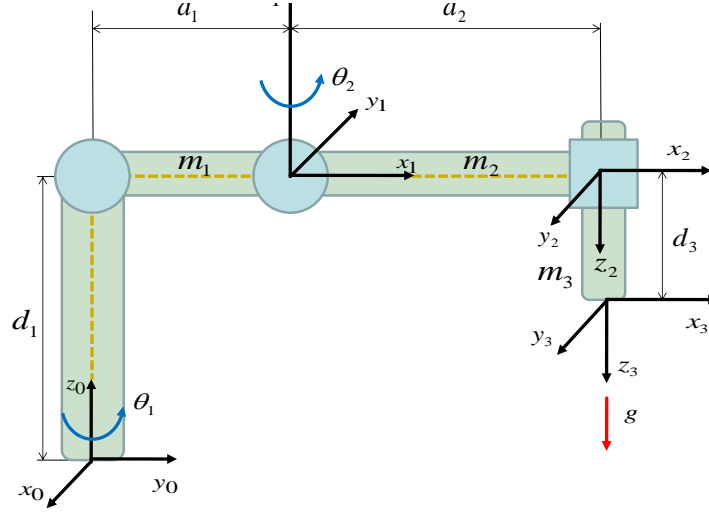


FIGURE 3.5. Scara kinematic structure.

TABLE 3.12. Dynamic initial parameters of 3-DOF SCARA configuration

Conf.	$G_x$	$G_y$	$G_z$	$I_{x1}$	$I_{y1}$	$I_{z1}$	$I_{x2}$	$I_{y2}$	$I_{z2}$	$I_{x3}$	$I_{y3}$	$I_{z3}$
Scara	0	0	-1	0	$\frac{1}{12}m_1a_1^2$	$\frac{1}{12}m_1a_1^2$	0	$\frac{1}{12}m_2a_2^2$	$\frac{1}{12}m_2a_2^2$	$\frac{1}{12}m_3d_3^2$	$\frac{1}{12}m_3d_3^2$	0

$$\begin{aligned}
 & \begin{bmatrix} \left(\frac{m_1}{3} + m_2 + m_3\right)a_1^2 + (m_2 + m_3)\cos(\theta_2)a_1a_2 + \left(\frac{m_2}{3} + m_3\right)a_2^2 \\ (m_3 + \frac{m_2}{3})a_2^2 + (m_3 + \frac{m_2}{2})\cos(\theta_2)a_1a_2 \\ 0 \end{bmatrix} \\
 & \begin{bmatrix} (m_3 + \frac{m_2}{3})a_2^2 + (m_3 + \frac{m_2}{2})\cos(\theta_2)a_1a_2 & 0 \\ (m_3 + \frac{m_2}{3})a_2^2 & 0 \\ 0 & m_3\ddot{d}_3 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + \begin{bmatrix} -2(m_3 + \frac{m_2}{2})\sin(\theta_2)a_1a_2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\
 & \begin{bmatrix} \dot{q}_1\dot{q}_2 \\ \dot{q}_1\dot{q}_3 \\ \dot{q}_2\dot{q}_3 \end{bmatrix} + \begin{bmatrix} 0 & -(m_3 + \frac{m_2}{2})\sin(\theta_2)a_1a_2 & 0 \\ (m_3 + \frac{m_2}{2})\sin(\theta_2)a_1a_2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1^2 \\ \dot{q}_2^2 \\ \dot{q}_3^2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -m_3g \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \quad (70)
 \end{aligned}$$

It is clear from (70) that the motion of the third joint is completely independent of the other two joints. The motion of the first two joints depends on the mass

of the third link, which acts as a load, but otherwise independent of the third joint.

### 3.5. Parameter Properties of a Reconfigurable Robot

The equations of motion (E.O.M) of various manipulators derived in the previous sections, can be described with a set of coupled differential equations in matrix form:

$$\tau = M(q)\ddot{q} + C(q, \dot{q}) + F\dot{q} + G(q) \quad (71)$$

where  $q$ ,  $\dot{q}$  and  $\ddot{q}$  are respectively the vector of generalized joint coordinates, velocities and accelerations.  $M$  is the joint-space inertia matrix,  $C$  is the Coriolis and centripetal coupling matrix,  $F$  is the friction force,  $G$  is the gravity loading, and  $\tau$  is the vector of generalized actuator torques associated with the generalized coordinates  $q$ . This equation describes the manipulator rigid-body dynamics and is known as the inverse dynamics, given the pose, velocity and acceleration it computes the required joint forces and torques. In the previous sections, An efficient reconfigurable recursive Newton-Euler algorithm was developed to compute the Equation (71) for any open kinematic chain manipulator. This algorithm starts at the base and working outwards adds the velocity and acceleration of each joint in order to determine the velocity and acceleration of each link. Then working from the tool back to the base, it computes the forces and moments acting on each link and thus the joint torques.

In this section, the effect of varying configurations on dynamic parameters such as the inertia and gravity load terms of a reconfigurable manipulator is analyzed and investigated thoroughly. The D–H parameters of a predefined kinematic structure (PUMA 560) are given in Table 3.13 by Corke [33]. Using these parameters, the workspace of the first three revolute joints and links is calculated and shown in Figure 3.6. The workspace of a reconfigurable robot manipulator with similar kinematic structure to the PUMA 560 is also calculated and shown in Figure 3.7. The resulting variable workspace shows a union of three layers that indicates the workspace variability property of any reconfigurable manipulator. The three workspace layers are calculated based on turning the third joint into a prismatic

TABLE 3.13. D–H parameters of the PUMA 560 Manipulator, source; Fu, [41].

Joint $j$	$\theta_j$	$d_j$	$a_j$	$\alpha_j$	Joint range
1	$\theta_1$	0.3302	0	$-\pi/2$	-160 to +160
2	$\theta_2$	0	0.2032	0	-225 to 45
3	$\theta_3$	-0.05	0	$\pi/2$	-45 to 225
4	$\theta_4$	0.2032	0	$-\pi/2$	-110 to 170
5	$\theta_5$	0	0	$\pi/2$	-100 to 100
6	$\theta_6$	0	0	0	-226 to 266

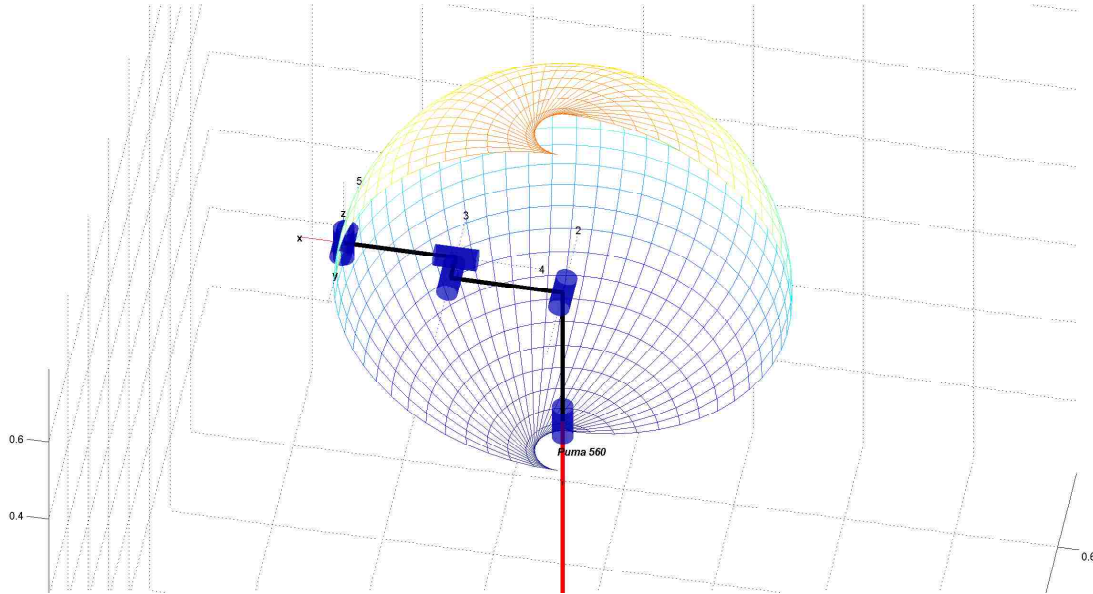


FIGURE 3.6. The workspace of a predefined kinematic structure of the PUMA 560.

one (translational motion). This variable workspace was generated when the third joint has translated with 0.1, 0.2, 0.3 m.

### 3.5.1. Gravity Load Term

Gravity load term in Equation (71) is generally a dominant term and present even when the manipulator is stationary or moving slowly. The torque on a joint due to gravity acting on the robot depends strongly on the robot's pose. The torque on the shoulder joint 2 is much greater when the robot is stretched out horizontally as shown in Figure 3.8. The gravity torque on the elbow is high when the pose changes due to joint reconfiguration from revolute to prismatic as the

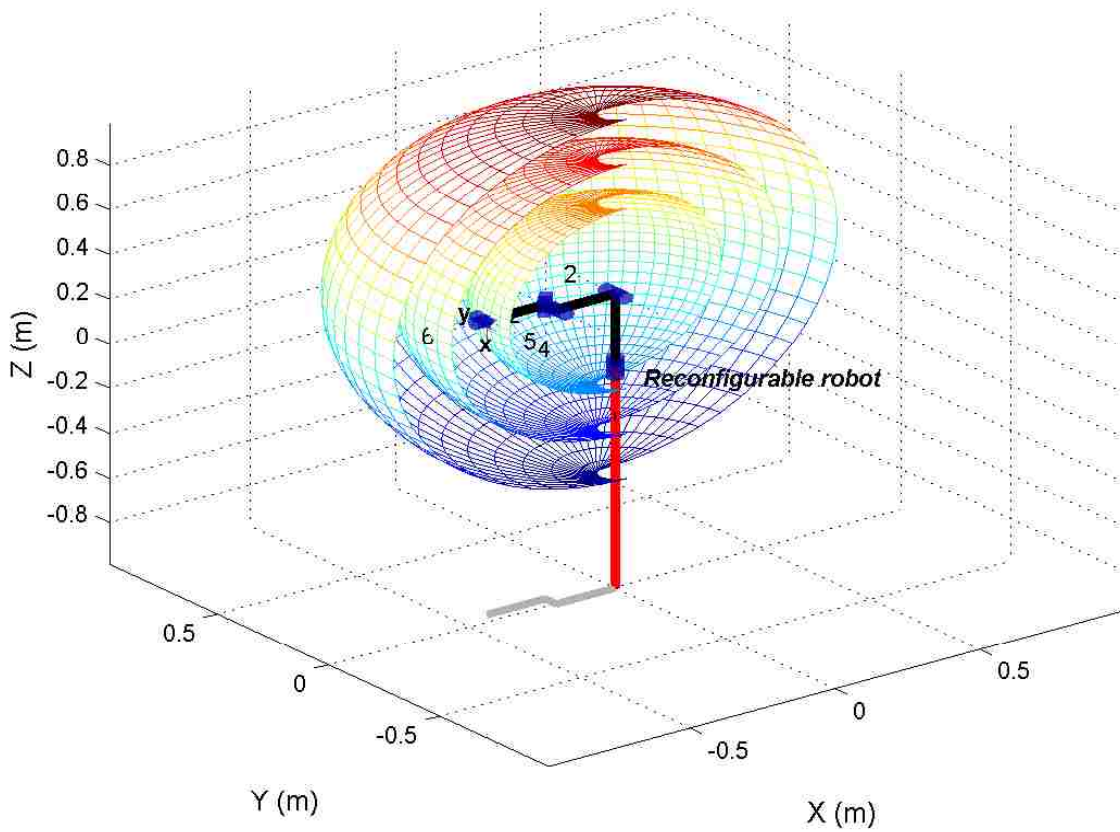


FIGURE 3.7. The workspace of a reconfigurable manipulator.

results are shown in Figure 3.9. From the analysis of the gravity load variations, the gravity torque on joint 2 varies between  $\pm 18Nm$  and for joint 3 varies between  $\pm 18Nm$ . This analysis is important to determine the required torque capacity for the motors.

### 3.5.2. Inertia Matrix Term

The inertia matrix is a positive definite symmetric matrix in which the matrix elements are functions of the manipulator pose. Locking the joint 4 into prismatic type, the inertia tensor matrix of the reconfigurable manipulator is calculated for nominal configuration  $q_n = [0, \pi/4, -\pi, 0, \pi/4, 0]$  in Appendix A.

The inertia matrix is symmetric and the diagonal elements  $M_{ij}$  describe the inertia exerting the joint  $j$  by torque  $Q_j = M_{ij}q_j$ . The first two diagonal elements, corresponding to the robot's waist and shoulder joints, are large since motion of

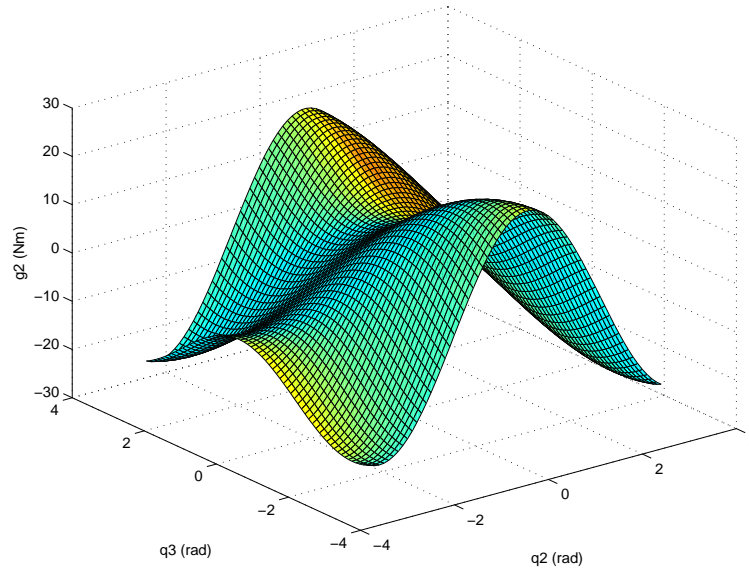


FIGURE 3.8. Gravity load variation with a reconfigurable manipulator pose (shoulder gravity load).

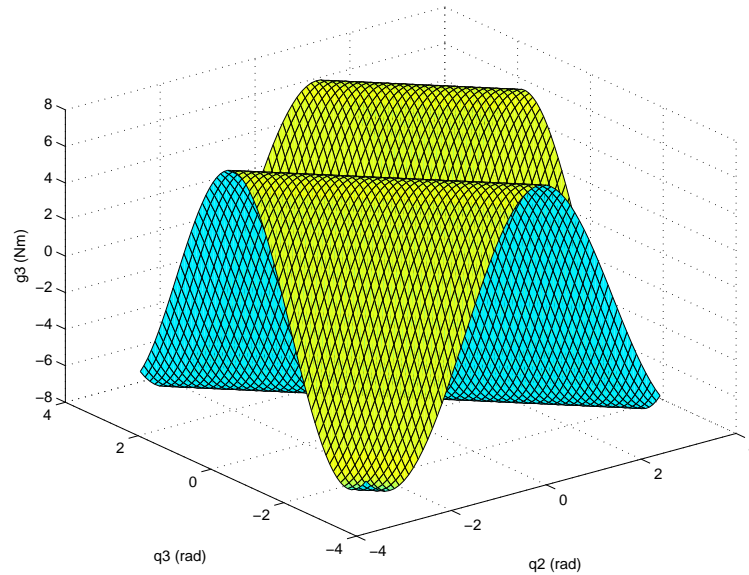


FIGURE 3.9. Gravity load variation with a reconfigurable manipulator pose (Elbow gravity load).

these joints involves rotation of the heavy upper- and lower-arm links. The off-diagonal terms  $M_{ij} = M_{ji}$ ,  $i \neq j$  represent coupling of acceleration from joint  $j$  to the torques and forces on joint  $j - 1$ . The variation of the inertia elements as a function of robot configurations are shown in Figures 3.10-3.11 and 3.12. The results indicate a significant variation in the value of  $M_{11}$  which changes by a factor of:  $\max(M_{11}(:)) / \min(M_{11}(:)) = 1.7683$ . The off-diagonal term  $M_{12}$  represents

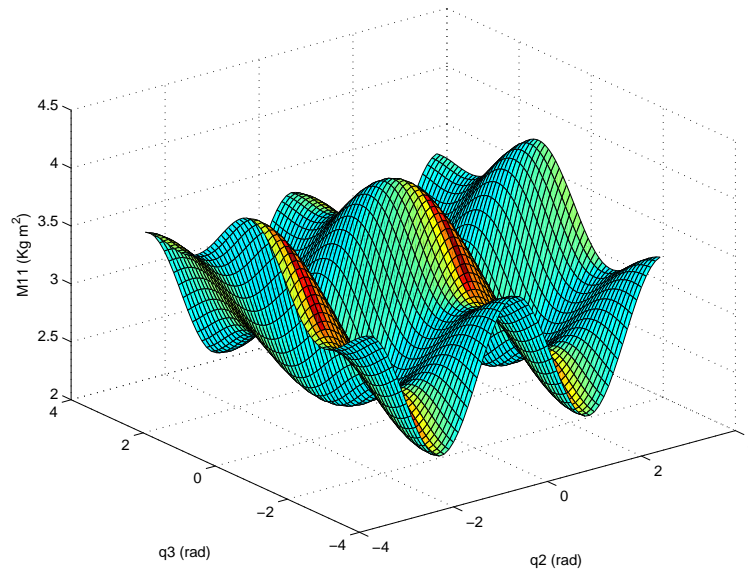


FIGURE 3.10. Variation of inertia matrix elements  $M_{11}$  with configuration.

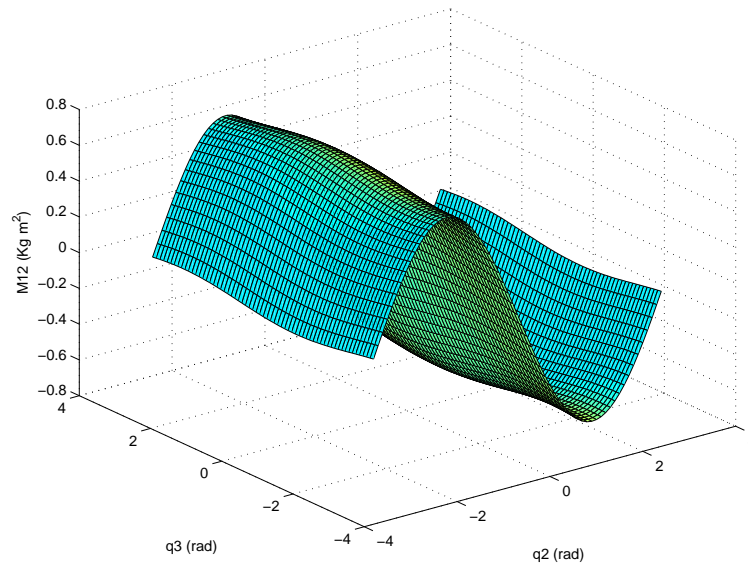


FIGURE 3.11. Variation of inertia matrix elements  $M_{12}$  with configuration.

coupling between the angular acceleration of joint 2 and the torque on joint 1. This means if the joint 2 accelerates then a torque will be exerted on joint 1 and vice versa.

### 3.5.3. Coriolis Matrix Term

The Coriolis matrix  $C(q)$  is a function of joint coordinates and joint velocity. The

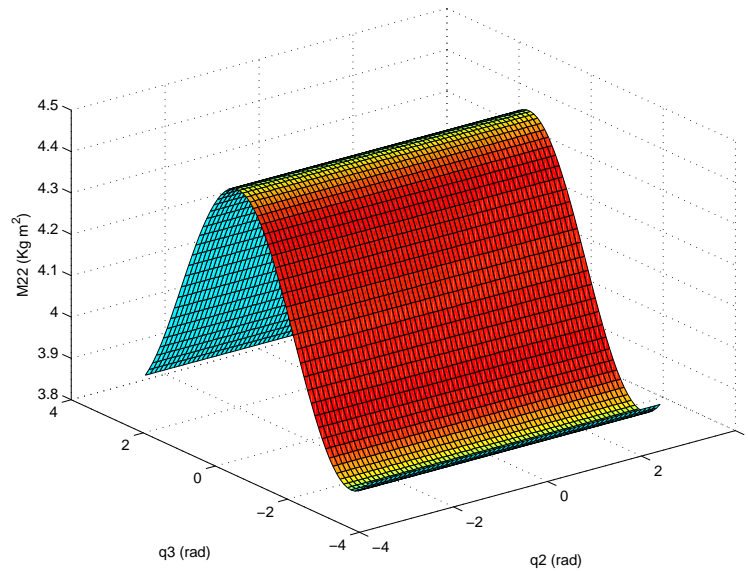


FIGURE 3.12. Variation of inertia matrix elements  $M_{22}$  with configuration.

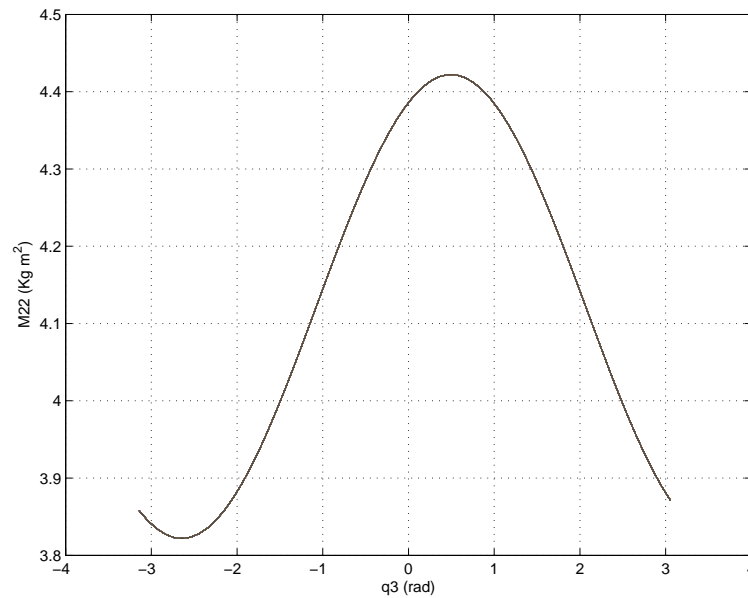


FIGURE 3.13. Joint 2 inertia as a function of joint 3 angle,  $M_{22}(q_3)$ .

centripetal torques are proportional to  $\dot{q}_i^2$ , while the Coriolis torques are proportional to  $\dot{q}_i \dot{q}_j$ . The Coriolis matrix is calculated for a nominal pose with all joints moving 0.5 rad/s:  $q_d = 0.5 * [1 \ 1 \ 0 \ 1 \ 1 \ 1]$ . The resulting numeric matrix has been shown in Appendix A. The off-diagonal terms  $C_{ij}$  represent coupling of joint  $j$  velocity to the generalized force acting on joint  $i$ .  $C_{1,2} = -0.9505$  is very significant and represents coupling from joint 2 velocity to torque on joint 1. Since

the elements of this matrix represent a coupling from velocity to joint force they have the same dimensions as viscous friction or damping, however the sign can be positive or negative. The joint torques in this example have been shown in Appendix A.

#### **3.5.4. Effect of Payload Term**

A specified maximum payload of any real robot shapes two dynamic effects. The first is that a mass at the end of the robot will increase the inertia influence by the joints which reduces acceleration and dynamic performance. The second is that mass generates a weight force which the joints needs to support. Consequently, the increased gravity torque component might exceed the rating of one or more motors. As an example, a 2.5 kg point mass is added to the reconfigurable manipulator end effector which is its rated maximum payload at an offset of 100 mm in the z-direction of the wrist frame. From the numerical results shown in Appendix (A) that the diagonal elements have increased significantly, for instance the elbow joint inertia has increased by 60% which reduces the maximum acceleration by nearly two thirds. Reduced acceleration impairs the robots ability to accurately follow a high speed path. The inertia of joint 6 is unaffected since this added mass lies on the axis of this joint's rotation. The off-diagonal terms have increased significantly, particularly in rows and column three, four (prismatic) and five. This indicates that motion of joints 4 and 5, the wrist joints, which are moving the offset mass give rise to large reaction forces that are covered by all the other robot joints.



## CHAPTER 4

### Linear Control

In this chapter, the synthesis and analysis of optimal linear controllers are viewed and derived for different robot configurations.

#### 4.1. Optimal Robust Control

In this section, a Scara robot model is considered for design and analysis of robust  $H_\infty/\mu$  controllers. The motivation of this consideration is that the model is essentially a two-axis articulated robot used to establish the horizontal tool position, plus an orthogonal one-axis prismatic link used to establish the vertical tool position. This robot has the minimum dynamic coupling, which reduces the complexity of the overall equations of motion. It is clear from Equation (70) that the motion of the third joint is completely independent of the other two joints. The motion of the first two joints depends on the mass of the third link, which acts as a load, but is otherwise independent of the third link. Since the first two joints only move the robot in the horizontal plane, the gravity load  $G$  is zero.

#### Nominal Systems

Linear dynamical systems are generally described in either state-space form, or as a transfer function. The state-space description of a general linear time invariant (LTI) system  $G$  is commonly represented by:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t), & x(t_0) &= x_0 \\ y(t) &= Cx(t) + Du(t)\end{aligned}\tag{72}$$

The system matrices,  $A, B, C, D$ , represent the linear nominal model of the system. The transfer function description can be related to the state-space form by

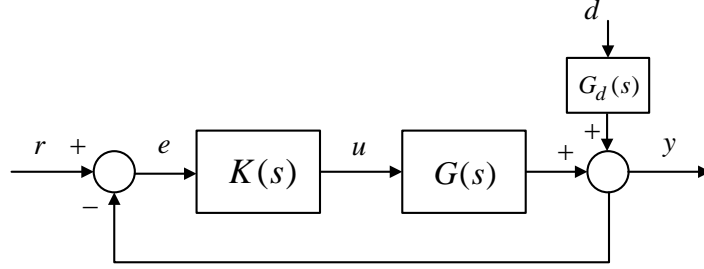


FIGURE 4.1. Closed loop configuration with mixed sensitivity consideration.

introducing the Laplace variable  $s \in \mathbb{C}$ :

$$\begin{aligned} sx(s) &= Ax(s) + Bu(s) \\ y(s) &= Cx(s) + Du(s) \end{aligned} \quad (73)$$

Eliminating  $x(s)$  from the equation results in:

$$y(s) = G(s)u(s), \quad G(s) = C(sI - A)^{-1}B + D \quad (74)$$

### Control Design Objectives

The standard feedback system is shown in Figure 4.1. The closed loop response is:

$$y = (I + GK)^{-1}GKr + (I + GK)^{-1}G_d d \quad (75)$$

where the following closed loop transfer functions are defined:

$L = GK$  loop transfer function

$S = (I + GK)^{-1} = (I + L)^{-1}$  sensitivity function

$T = (I + GK)^{-1}GK = (I + L)^{-1}L$  complementary sensitivity function where

$S + T = I$ . The control error is:

$$e = y - r = -Sr + SG_d d \quad (76)$$

The corresponding system input signal is;

$$u = K Sr - K S G_d d \quad (77)$$

The control design objectives in feedback control are:

- (1) Performance, good disturbance rejection: need large controller gains  $L$  large.
- (2) Performance, good command following:  $L$  large.
- (3) Stabilization of unstable system:  $L$  large.
- (4) Small magnitude of input signals:  $K$  small and  $L$  small.

The disturbance rejection and command tracking are obtained with  $S \approx 0$ , or equivalently,  $T \approx I$  in the low frequency range. The weighted sensitivity function  $S$  is an indicator of closed loop performance. The design performance specifications in terms of  $S$  include:

- (1) Minimum bandwidth frequency  $w_B$  (defined as the frequency where  $\|S(jw)\|$  crosses 0.707 ( $\approx -3\text{dB}$ )).
- (2) Maximum tracking error at selected frequencies.
- (3) System type, or maximum steady state tracking error.
- (4) Maximum peak magnitude of  $S$ ,  $\|S(jw)\| \leq M$ .

These specifications can be captured by an upper bound,  $\|1/w_P\|$ , on the magnitude  $S$ , where  $w_P$  is a weight selected by the designer. The performance requirements become:

$$\begin{aligned} |S(jw)| &< 1/|1/w_P(jw)|, \quad \forall w \\ \Leftrightarrow \|w_P S\| &< 1, \forall w \Leftrightarrow \|w_P S\|_\infty < 1 \end{aligned} \tag{78}$$

## 4.2. Mixed Sensitivity $H_\infty$ Control

A control system is robust if it remains stable and achieves certain performance criteria in the presence of parametric (Inertia and viscous friction) and dynamic (structural unmodeled high frequency) uncertainties.  $H_\infty$  control design uses a combination of cost functions to achieve good tracking and simultaneously limit the control signal energy. The mixed sensitivity  $S/KS$  control problem for a

reconfigurable robot is to find a stabilizing controller  $K$  that minimizes [75]:

$$\min_{K \text{ stabilizing}} \left\| \begin{array}{c} (I + GK)^{-1} \\ K(I + GK)^{-1} \end{array} \right\|_\infty \quad (79)$$

where the sensitivity function  $S = (I + GK)^{-1}$  is shaped along with the closed loop function transfer function  $KS$ . This cost function can be interpreted as the design objectives of nominal performance (without Uncertainties), good tracking or disturbance rejection, and robust stabilizing with additive uncertainties. The sensitivity function  $S$  is the transfer function between the disturbance  $d$  and the output  $y$  as shown in Figure 4.1. The  $KS$  is the transfer function between  $d$  and control signal  $u$ . The  $KS$  function is regarded as a mechanism for limiting the gain and bandwidth of the controller, and hence the control energy used. A scalar high pass (weighting) filter is designed with a crossover frequency approximately equal to that of the desired closed loop system. The disturbance  $d$  is usually a low frequency signal, and therefore it will be rejected if the maximum singular value of  $S$  is made small over the same low frequencies. A scalar low pass filter (weighting) function  $w_p$  is selected with a bandwidth equal to that of the disturbance frequency.

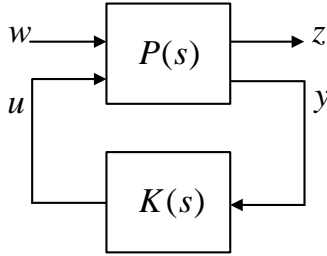


FIGURE 4.2. General  $H_\infty$  control configuration.

By combining these two objectives in one cost function Equation (79), the problem is to find a stabilizing control that minimizes this cost function while achieving the required performance. In order to implement a unified solution procedure, the above cost function is recast into a standard  $H_\infty$  configuration shown in Figure 4.2. The solution can be obtained by using the LFT (Linear Fractional Transform) technique, in which the signals are classified into sets of external inputs, outputs, input to the controller and output from the controller.

The external inputs (reference and disturbance) are denoted by  $w$ , the output signals to be minimized which includes both performance and robustness measures,  $y$  is the vector of measurements available to the controller  $K$  and  $u$  the vector of control signals.  $P(s)$  is called the generalized plant or interconnected system. The objective is to find a stabilizing controller  $K$  to minimize the output  $z$  over all  $w$  with energy less than or equal to 1. This is equivalent to minimizing the  $H_\infty$ -norm of the transfer function from  $w$  to  $z$ . The mixed sensitivity problem [75] shown in Figure 4.3 (left) is formulated to reject the disturbance when  $w = d$ . The output error signal is defined as  $z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$ , where  $z_1 = w_p y$  and  $z_2 = -w_u u$ . It is also calculated that  $z_1 = w_p S w$  and  $z_2 = w_u K S w$  and the elements of the generalized plant  $P$  are:

$$\begin{bmatrix} z_1 \\ z_2 \\ v \end{bmatrix} = \left[ \begin{array}{c|c} w_p & w_p G \\ \hline 0 & -w_u \\ \hline -I & -G \end{array} \right] \begin{bmatrix} w \\ u \end{bmatrix}$$

Then:

$$z = F_l(P, K)w = [P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21}]w$$

where the  $F_l(P, K)$  is the lower linear fractional transformation of  $P$  and  $K$ :

$$F_l(P, K) = \begin{bmatrix} w_p S \\ w_u K S \end{bmatrix} \quad (80)$$

Another formulation of the  $S/KS$  mixed sensitivity optimization is in the standard tracking control form as shown in Figure 4.3 (right). This is a tracking problem in which the input signal is the reference command  $r$  and the error signals are  $z_1 = -w_p e = w_p(r - y)$  and  $z_2 = w_u u$ . The results of the tracking problem is to minimize  $z_1 = w_p S r$  and  $z_2 = w_u K S r$ .

### 4.3. $H_\infty$ Control Design

This section is mainly concerned with the design of an  $H_\infty$  feedback control to stabilize the closed loop of a linear reconfigurable robot in the presence of uncertain

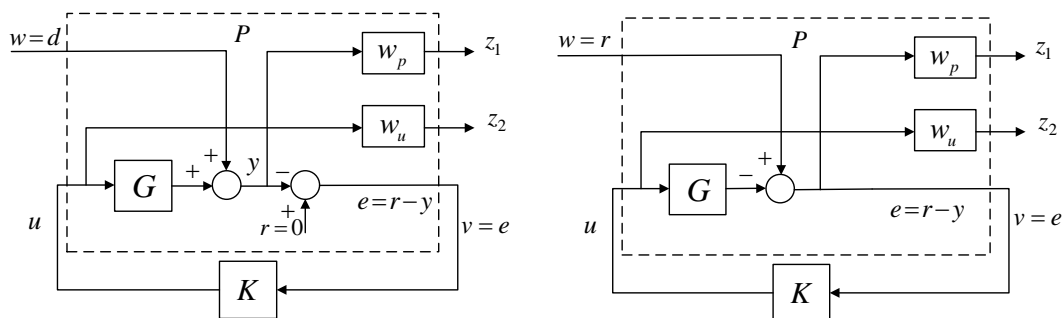


FIGURE 4.3. S/SK mixed sensitivity optimization in regulation form (left), S/SK mixed sensitivity minimization in tracking form (right).

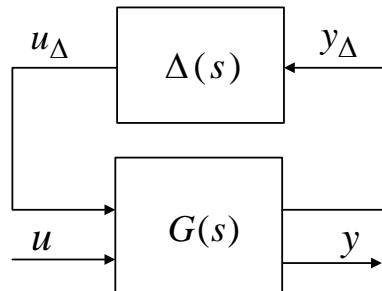


FIGURE 4.4. Uncertain system representation.

parameters. The robust stability of the resulting closed loop has been analyzed based on the structured singular value  $\mu$  approach [75]. The nominal and robust performance were also analyzed based on the minimizing of performance criteria functions such the sensitivity and limiting control functions.

### Uncertain Systems

For control design purposes, the possibly complex behavior of dynamical systems must be approximated by models of relatively low complexity. The difference between such models and the actual physical system is called the model uncertainty. Another cause of uncertainty is the imperfect knowledge of some components of the system, or the change of their behavior due to changes in operating conditions. Uncertainty also originates from physical parameters whose value is only approximately known or varies in time. There are two major classes of uncertainty:

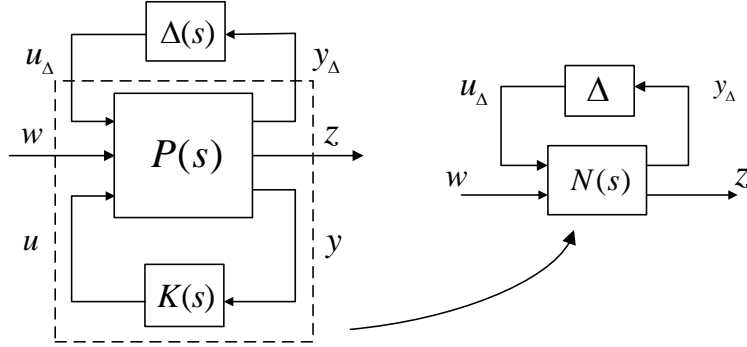


FIGURE 4.5. Standard  $H_\infty/\mu$  control structure (left), A control structure for closed system analysis (right).

- Dynamical uncertainty, which consists of dynamical components neglected in the linear model as well as of variations in the dynamical behavior during operation. For example, high-frequency flexible modes, nonlinearities for large inputs, slow time variations, ..., etc.
- Parameter uncertainty are usually expressed in terms of accuracy coming from calibration and its imperfect nature. Moreover, it originates from imperfect knowledge of the physical parameter values, or from variations of these parameters during operation. Examples of physical parameters include inertia, stiffness and damping coefficients in mechanical systems.

It is important to consider the errors between a model and the system it represents, which can be presented in figure 4.4. The system  $G(s)$  is the extended system with inputs and outputs that connect with the uncertainty  $\Delta$ , representing the model error.  $\Delta$  is assumed to be known and bounded, and may have a specific structure. The uncertainty structure is captured by defining the set  $\Delta$ , which is the set of block diagonal matrices with a specified fixed structure:

$$\Delta = \{ \text{diag}[\delta_1 I_{r_1}, \dots, \delta_s I_{r_s}, \Delta_1, \dots, \Delta_F] : \delta_i \in \mathbb{C}, \Delta_j \in \mathbb{C}^{m_j \times m_j}, \|\Delta\| < 1 \} \quad (81)$$

Hence any matrix in  $\Delta$  is block diagonal with two types of blocks: repeated scalar (complex or real) and full blocks.  $S$ ,  $F$  are the number of repeated scalar blocks and full blocks, respectively. The repeated scalar block dimension is  $r_i \times r_i$ , while the full block dimension is  $m_j \times m_j$ .

### System Interconnections

The standard control structure of the  $H_\infty/\mu$  controller design is shown in Figure 4.5 (left), representing the control design problem: the 'generalized plant  $P(s)$ ' contains the nominal system model as well as the system weighting functions representing the design specifications, disturbance spectra, uncertainty and input weightings, etc. The uncertainty  $\Delta(s)$  represents the uncertainty model which is assumed to be normalized. Finally, the controller  $K(s)$  has to be designed so that the influence of  $w$  on  $z$  is small. Here  $w$  includes external signals that excite the system, such as disturbance, noise and reference signals. The output  $z$  contains signals which are to be kept small, such as error and control signals. To illustrate the main idea of design and analysis of a perturbed system, it is useful to define the *lower* and *upper linear fractional transformation* (LFT) denoted by  $F_l$  and  $F_u$ , respectively. The closed loop transfer function matrix  $N$  between error signals and external inputs, shown in Figure 4.5 (right), is related to  $P$  and  $K$  by a lower LFT:

$$N = F_l(P, K) = P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21} \quad (82)$$

The closed loop  $N$  can be partitioned as follow:

$$N = F_l(P, K) = \begin{bmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{bmatrix} \quad (83)$$

Similarly, the uncertain closed loop transfer function from  $w$  to  $z$  is related to  $N$  and  $\Delta$  by an upper LFT:

$$F = F_u(N, \Delta) = N_{22} + N_{21}K(I - N_{11}K)^{-1}N_{12} \quad (84)$$

To analyze robust stability and performance of  $F$ , the perturbed close loop system is rearranged with structured uncertainties as shown in Figure 4.6 (right). Now, two important measures will be introduced that are used as performance measure for the  $H_\infty/\mu$  control configurations.



**$H_\infty$ -Norm**

A measure for the gain of a stable system  $N$  is the  $H_\infty$  norm, defined as:

$$\|N\|_\infty = \sup_{\|u\|_2 > 0} \frac{\|Nu\|_2}{\|u\|_2} \quad (85)$$

which can be viewed as the largest possible amplification that  $N$  will induce, given the worst-case input signal  $u(t)$ . It is shown by Qu [69], that:

$$\|N\|_\infty = \sup_{w \in R} \bar{\sigma}(N(jw)) \quad (86)$$

where  $\bar{\sigma}$  denotes the largest singular value. If  $N_{22}$  represents a nominal closed loop system, then  $\|N_{22}\|_\infty \leq 1$  indicates that the performance requirements have been met (Nominal Performance).

 **$\mu$ -Norm**

An important question is whether a nominal stable transfer function  $N$ , as defined in Equation (82) remains stable for all  $\Delta \in \mathbf{\Delta}$  and if not, what the smallest size is of a perturbation that can destabilize  $N$ . This is defined by the *structured singular value*  $\mu$ :

$$\mu_{\mathbf{\Delta}}(N_{11})^{-1} = \min_{\Delta} \{ \bar{\sigma}(\Delta) \mid \det(I - N_{11}\Delta) = 0 \text{ for structured } \Delta : \bar{\sigma}(\Delta) \leq 1 \} \quad (87)$$

An uncertain closed loop system has been shown in Figure 4.6 where the input  $w$  is to describe the disturbance and reference signals,  $z$  is the output to describe the error signal. Robust stability, nominal performance and robust performance are analyzed as follows:

- Robust stability (RS) is achieved if  $\mu_{\mathbf{\Delta}}(N_{11}(jw)) \leq 1, \forall w$ .
- Nominal performance (NP) is achieved if  $\|N_{22}(jw)\| \leq 1, \forall w$ .
- Robust performance (RP) is achieved if  $\mu_{\mathbf{\Delta}_e}(N(jw)) \leq 1, \forall w$

$$\text{where } \mathbf{\Delta}_e = \left\{ \left( \begin{array}{cc} \Delta & 0 \\ 0 & \Delta_p \end{array} \right) \mid \Delta \in \mathbf{\Delta}, \|\Delta_p\| < 1, \right\}.$$

where the performance complex uncertainty  $\Delta_p$  has the performance channel dimension of  $(\dim r + \dim d) \times (\dim e + \dim u)$ .

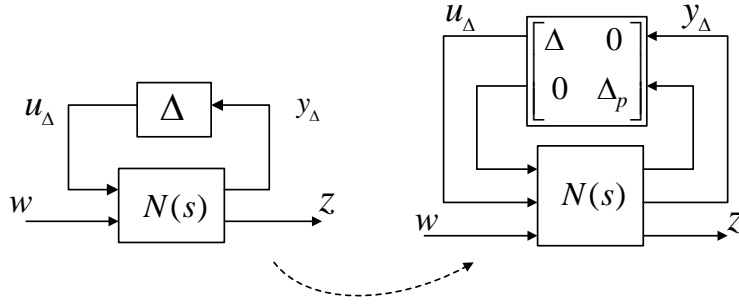


FIGURE 4.6. A control configuration with extended uncertainty structure.

### $H_\infty$ -Optimal Control

Based on the definition of the  $H_\infty$  norm it is straightforward to define the  $H_\infty$ -optimal control problem. Assume there is no uncertainty present, see Figure 4.2, such that the control problem can be viewed as:

$$K = \min_K \|F_l(P, K)\|_\infty \quad (88)$$

for stabilizing  $K$ . There are two problems associated with this definition. First, the controller  $K$  that minimizes the  $H_\infty$  norm is not unique. Secondly, it is not directly solvable. However, it is possible to construct a controller  $K$  that satisfies:

$$\|F_l(P, K)\|_\infty < \gamma \quad (89)$$

for a feasible  $\gamma$ , where  $\gamma$  is the lower bound on the  $H_\infty$  norm.

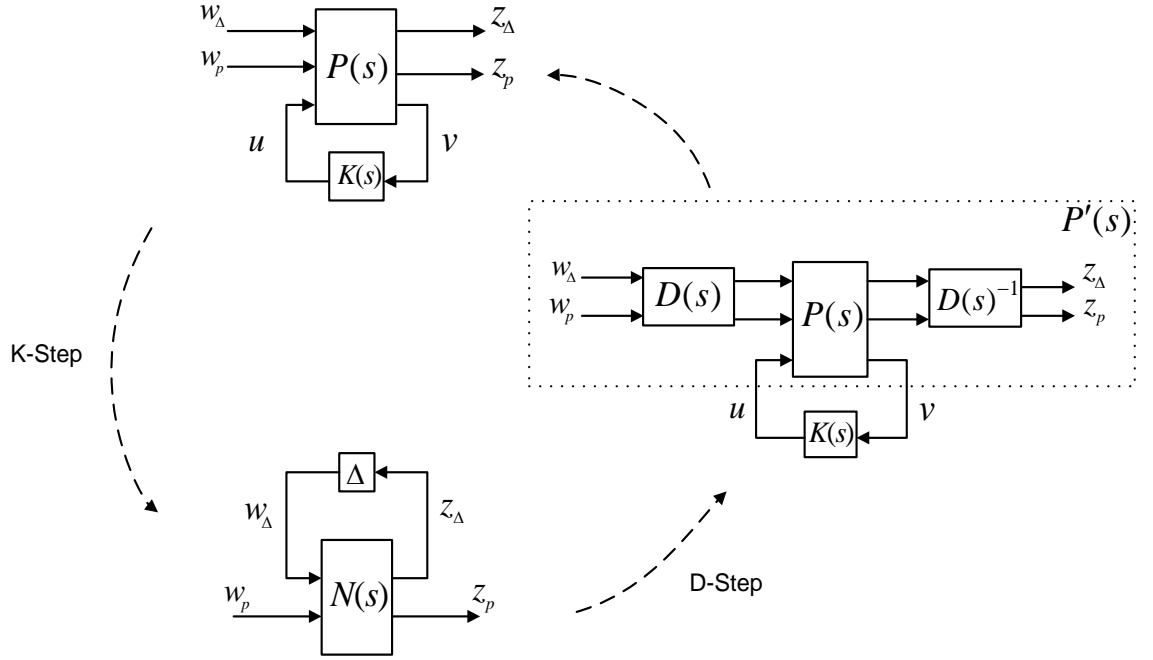
### 4.4. $\mu$ -Optimal Control

The next step is to include the uncertainty model into the control design problem and solve:

$$K = \min_K \|F_l(P, K)\|_\mu \quad (90)$$

The solution of this problem is iterative and referred to as *DK-iteration*. It starts with the  $\mu$  upper bound problem by minimizing the scaled singular value of the closed loop matrix  $F_l(P, K)$ :

$$\mu_\Delta(F_l(P, K)) \leq \inf_{D(w) \in \mathbb{D}} \bar{\sigma}(D(w)F_l(P, K)D(w))^{-1} \quad (91)$$

FIGURE 4.7.  $\mu$ -Synthesis Control Procedure.

where  $\mathbb{D}$  is the set of block diagonal matrices whose structure is compatible to that of  $\Delta$  described in Equation (81).  $D(w)$  is a frequency dependent scaling. Thus the  $\mu$  upper bound problem is to minimize:

$$\min_K \| D(w)F_l(P, K)D(w)^{-1} \|_\infty \quad (92)$$

over all controllers  $K$  that stabilizes  $P$  and over all functions  $D(w) \in \mathbb{D}$ . This problem is solved as follows:

- Solve an  $H_\infty$  optimization problem over all  $K$ :

$$\min_K \| D(w)F_l(P, K)D(w)^{-1} \|_\infty \quad (93)$$

over all stabilizing  $K$  and let the minimizing controller denoted by  $\hat{K}$ . Thus minimizing  $H_\infty$  norm of the scaled maximum singular value of the closed loop system matrix.

- Minimize  $\mu$  upper bound  $\bar{\sigma}[D(w)F_l(P, \hat{K})D(w)^{-1}]$  over  $D(w)$  point-wise across frequency. This minimization produces a new scaling function  $\hat{D}(w)$ .
- Replace  $D(w)$  with  $\hat{D}(w)$  and return to first step.

The DK-iteration procedure is displayed in Figure 4.7.

### 4.5. $H_\infty$ Gain Scheduled Control

There are two considerations that motivate the application of robust gain scheduled control, namely:

- In the previous chapter, an  $H_\infty$  control is designed in face of the involved parametric uncertainties  $\delta_J$  and  $\delta_V$ . The frequency and time domain responses have indicated that the desired performance specifications are not achieved with an acceptable control amplitude.
- The parameters of a reconfigurable robot dynamics such as the inertia  $M(q)$  and damping  $F(\dot{q})$  parameters are strongly dependent on robot configuration. These variable parameters completely define the operating point of the robot and are assumed to be measured in real time.

These two considerations motivate the design of a control system that is scheduled with the measured parameters  $M(q)$  and  $F(\dot{q})$  and such that it might provide higher performance for large variation in these parameters. Gain scheduling or linear parameter varying (LPV) techniques are used for controlling LPV systems. An LPV controller consists of designing a linear time invariant LTI controller that is adapting itself when the operating conditions change. In this control method, the system is assumed to depend affinely on a measured vector of time varying parameters. Assuming on-line measurements of these parameters, they can be fed to the controller to optimize the performance and robustness of the closed loop system.

#### 4.5.1. Analysis of LPV Polytopic Systems and Controllers

Linear parameter varying LPV systems are systems whose state-space matrices are fixed functions of a time varying parameter  $\theta(t)$ . Hence LPV systems are described

by state-space equations of the form:

$$\begin{aligned}\dot{x} &= A(\theta(t))x(t) + B(\theta(t))u \\ y &= C(\theta(t))x(t) + D(\theta(t))u\end{aligned}\tag{94}$$

where

$$\theta(t) = (\theta_1(t), \dots, \theta_n(t)), \quad a \leq \theta_i(t) \leq b\tag{95}$$

is a time varying vector  $\theta(t)$ . The state-space matrices of Equation (94) are considered to be affinely dependent on the time varying real parameters vector  $\theta(t)$  of Equation (95). LPV systems can be interpreted as follows:

- They can be viewed as linear time invariant LTI systems subject to time varying parameter uncertainty  $\theta(t)$ .
- They can be models of linear time varying systems or result from the linearization of a non-linear system along trajectories of the vector parameter  $\theta(t)$ .

The first class of LPV systems falls within the scope of robust controller synthesis. For the second class of LPV systems, the parameter  $\theta_i(t)$  is no longer uncertain and can be on-line measured during the system operation. Consequently, LPV controllers can exploit the available measurements of  $\theta_i(t)$  to improve the resulting performance. It is assumed that the designing controller has the same parameter dependence as the system:

$$\begin{aligned}\dot{x}_k &= A_k(\theta(t))x_k + B_k(\theta(t))y \\ u &= C_k(\theta(t))x_k + D_k(\theta(t))y\end{aligned}\tag{96}$$

where  $y$  denotes the vector of measurements and  $u$  the control input. By incorporating the parameter measurements, the controller Equation (96) adjusts to the variations in the system dynamics in order to maintain stability and high performance along all parameter trajectories  $\theta(t)$ . LPV control synthesis is related to the quadratic  $H_\infty$  performance, in which the controller seeks a single quadratic Lyapunov function to ensure  $H_\infty$  performance for all possible trajectories of the LPV system. Such an approach remains conservative in the face of slowly varying

parameters, since quadratic Lyapunov techniques allow for arbitrarily fast parameter variations. The underlying concepts (polytopic systems and quadratic  $H_\infty$  performance) of the LPV control synthesis are briefly introduced.

### Polytopic Systems

An LPV system is called polytopic when it can be represented by state-space matrices  $A(\theta(t))$ ,  $B(\theta(t))$ ,  $C(\theta(t))$  and  $D(\theta(t))$ , where the parameter vector  $\theta(t)$  ranges over a fixed polytope and the dependence of  $A(\cdot)$ ,  $B(\cdot)$ ,  $C(\cdot)$  and  $D(\cdot)$  on  $\theta$  is affine. The describing matrices of the LPV system Equation (94) are collected into a system matrix:

$$S(\theta(t)) = \begin{pmatrix} A(\theta(t)) & B(\theta(t)) \\ C(\theta(t)) & D(\theta(t)) \end{pmatrix} \quad (97)$$

and the parameter vector  $\theta(t)$  ranges within a box as given by Equation (95). LPV system Equation (97) is called a polytopic system matrix if it ranges within a convex hull of a finite number of vertex systems  $S_i$  with fixed dimension matrix:

$$S(\theta(t)) \in Co\{S_1, \dots, S_r\} = \left\{ \sum_{i=1}^r \alpha_i S_i, \quad \alpha_i \geq 0, \quad \sum_{i=1}^r \alpha_i = 1 \right\} \quad (98)$$

where  $S_1, \dots, S_r$  are the vertex (frozen) systems.

### Quadratic $H_\infty$ -Control Performance

The performance of an LPV system is assessed by using the energy gain ( $L_2$ -induced norm) as:

$$\|z\|_{L_2} < \gamma \|w\|_{L_2} \quad (99)$$

This  $H_\infty$  performance condition can be rewritten as:

$$\int_0^\infty \|z(t)\|^2 dt \leq \gamma^2 \int_0^\infty \|w(t)\|^2 dt \quad (100)$$

An LPV system has a quadratic  $H_\infty$  performance  $\gamma$  if and only if there exists a single positive definite matrix  $X > 0$  such that:

$$\begin{pmatrix} A(\theta(t))^T X + X A(\theta(t)) & X B(\theta(t)) & C(\theta(t))^T \\ B(\theta(t))^T X & -\gamma I & D(\theta(t))^T \\ C(\theta(t)) & D(\theta(t)) & -\gamma I \end{pmatrix} < 0 \quad (101)$$

for all admissible values of the parameter vector  $\theta(t)$ . Then the single quadratic Lyapunov function  $V(x) = x^T X x$  establishes global asymptotic stability, and the  $L_2$ -gain of the LPV system is bounded by  $\gamma$  along all possible parameter trajectories  $\theta(t)$ . The bounded real lemma Equation (101) imposes an infinite number of constraints that are difficult to solve. Using the convexity property of polytopic systems Equation (98), this condition can be reduced to a finite set of LMIs constraints given as:

$$\begin{pmatrix} A_i^T X + X A_i & X B_i & C_i^T \\ B_i^T X & -\gamma I & D_i^T \\ C_i & D_i & -\gamma I \end{pmatrix} < 0, \quad X > 0, \quad \text{for } i = 1, \dots, r \quad (102)$$

#### 4.5.2. Synthesis of LPV Polytopic Controllers

LPV control is applicable to time varying and non-linear systems whose linearized dynamics are approximated by an affine parameter dependent system in the form of:

$$\begin{aligned} \dot{x} &= A(\theta(t))x + B_1(\theta(t))w + B_2(\theta(t))u \\ z &= C_1(\theta(t))x + D_{11}(\theta(t))w + D_{12}(\theta(t))u \\ y &= C_2(\theta(t))x + D_{21}(\theta(t))w + D_{22}(\theta(t))u \end{aligned} \quad (103)$$

where the time varying parameter vector  $\theta(t)$  ranges within a known interval.  $w \rightarrow z$  is the channel to describe the performance specifications. The system

Equation (103) is further assumed to be polytopic:

$$\begin{pmatrix} A(\theta(t)) & B_1(\theta(t)) & B_2(\theta(t)) \\ C_1(\theta(t)) & D_{11}(\theta(t)) & D_{12}(\theta(t)) \\ C_2(\theta(t)) & D_{21}(\theta(t)) & D_{22}(\theta(t)) \end{pmatrix} \in Co \left\{ \begin{pmatrix} A_i & B_{1i} & B_{2i} \\ C_{1i} & D_{11i} & D_{12i} \\ C_{2i} & D_{21i} & D_{22i} \end{pmatrix}, i = 1, 2, \dots, r \right\} \quad (104)$$

where  $A_i, B_{1i}, \dots$ , denote the values of  $A(\theta(t)), B_1(\theta(t)), \dots$ , at the polytope vertices. The system matrix dimensions are given by:

$$A(\theta(t)) \in \mathbb{R}^{n \times n}, \quad D_{11}(\theta(t)) \in \mathbb{R}^{p_1 \times m_1}, \quad D_{22}(\theta(t)) \in \mathbb{R}^{p_2 \times m_2} \quad (105)$$

Then an LPV polytopic controller of the form Equation (96) can be employed to assure the quadratic  $H_\infty$  performance  $\gamma$  of the resulting closed loop system:

$$\begin{aligned} \dot{x}_{cl} &= A_{cl}(\theta(t))x + B_{cl}(\theta(t))w \\ z &= C_{cl}(\theta(t))x + D_{cl}(\theta(t))w \end{aligned} \quad (106)$$

where

$$\begin{aligned} A_{cl}(\theta(t)) &= \begin{bmatrix} A(\theta(t)) + B_2(\theta(t))D_k(\theta(t))C_2(\theta(t)) & B_2(\theta(t))C_k(\theta(t)) \\ B_k(\theta(t))C_2(\theta(t)) & A_k(\theta(t)) \end{bmatrix} \\ B_{cl}(\theta(t)) &= \begin{bmatrix} B_1(\theta(t)) + B_2(\theta(t))D_k(\theta(t))D_{21}(\theta(t)) \\ B_k(\theta(t))D_{21}(\theta(t)) \end{bmatrix} \\ C_{cl}(\theta(t)) &= [C_1(\theta(t)) + D_{12}(\theta(t))D_k(\theta(t))C_2(\theta(t)) \quad D_{12}(\theta(t))C_k(\theta(t))] \\ D_{cl}(\theta(t)) &= D_{11}(\theta(t)) + D_{12}(\theta(t))D_k(\theta(t))D_{21}(\theta(t)) \end{aligned} \quad (107)$$

Synthesis of an LPV control is to ensure the following:

- The resulting polytopic closed loop system Equation (106) is enforced to be stable over the entire parameter polytope and for arbitrary parameter variations.
- The  $L_2$ -induced norm of the performance channel  $w \rightarrow z$  is bounded by  $\gamma$  for all possible trajectories within the parameter vector  $\theta(t)$ .



### 4.5.3. Analysis of LPV Systems with LFT System Description

In this section, the analysis of LPV systems based on the linear fractional transformation LFT systems description will be introduced. The LPV system (103) can be represented by the upper LFT interconnection as:

$$\begin{bmatrix} z \\ y \end{bmatrix} = F_u(P(s), \Theta(t)) \begin{bmatrix} w \\ u \end{bmatrix} \quad (108)$$

where  $P(s)$  is an LTI system and  $\Theta(t)$  is a block diagonal time varying matrix with the block-diagonal structure:

$$\Theta(t) = \text{diag}(\theta_1(t)I_{r_1}, \dots, \theta_k(t)I_{r_k}) \quad (109)$$

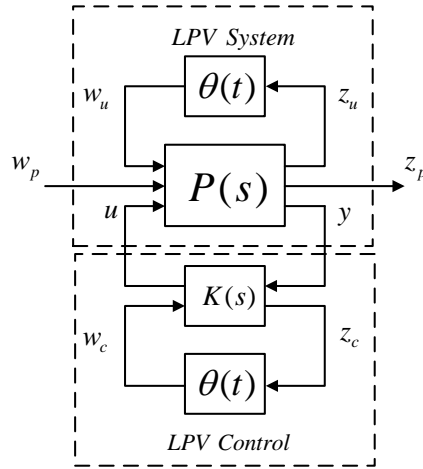


FIGURE 4.8. The structure of the LPV system and control using Linear Fractional Transformation (LFT) description system.

The LPV system structure with LFT interconnection is shown in Figure 4.8. In this control structure, the LTI system  $P(s)$  can be written in state-space form as:

$$\begin{bmatrix} \dot{x} \\ z_u \\ z_p \\ y \end{bmatrix} = \begin{bmatrix} A & B_u & B_p & B_1 \\ \hline C_u & D_{uu} & D_{up} & D_{u1} \\ C_p & D_{pu} & D_{pp} & D_{p1} \\ C_1 & D_{1u} & D_{1p} & D_{11} \end{bmatrix} \begin{bmatrix} x \\ w_u \\ w_p \\ u \end{bmatrix} \quad (110)$$

where  $A \in \mathbb{R}^{n \times n}$ ,  $D_{uu} \in \mathbb{R}^{r \times r}$ ,  $D_{pp} \in \mathbb{R}^{z_p \times w_p}$ ,  $D_{11} \in \mathbb{R}^{p_1 \times m_1}$ , with time varying parameters entering as

$$w_u = \Theta(t) z_u \quad (111)$$

The LTI system associated with the state-space form Equation (110) can be written as:

$$\begin{bmatrix} z_u \\ z_p \\ y \end{bmatrix} = \begin{bmatrix} P_{uu}(s) & P_{up}(s) & P_{u1}(s) \\ P_{pu}(s) & P_{pp}(s) & P_{p1}(s) \\ P_{1u}(s) & P_{1p}(s) & P_{11}(s) \end{bmatrix} \begin{bmatrix} w_u \\ w_p \\ u \end{bmatrix} \quad (112)$$

where  $w_u \rightarrow z_u$  is to capture the influence of the parameters, and  $w_p \rightarrow z_p$  is interpreted as the performance channel. Now, the LPV system Equation (108) can be written in terms of uncertain LTI system with  $\Theta(t)$  entering as:

$$\begin{aligned} \begin{bmatrix} z_p \\ y \end{bmatrix} &= F_u(P(s), \Theta(t)) \begin{bmatrix} w_p \\ u \end{bmatrix} \\ &= \begin{bmatrix} P_{pp} & P_{p1} \\ P_{1p} & P_{11} \end{bmatrix} + \begin{bmatrix} P_{pu} \\ P_{1u} \end{bmatrix} \Theta(t) (I - P_{uu})^{-1} \begin{bmatrix} P_{up} & P_{u1} \end{bmatrix} \end{aligned} \quad (113)$$

Consistently with (108), an LPV controller is selected as follows:

$$u = F_l(K(s), \Theta(t))y \quad (114)$$

where the LTI controller system:

$$K(s) = \begin{bmatrix} K_{11}(s) & K_{1c}(s) \\ K_{c1}(s) & K_{cc}(s) \end{bmatrix} \quad (115)$$

specifies the LFT dependence of the controller on the measured parameters  $\theta(t)$ . The resulting closed loop system from the exogenous input  $w_p$  to the controlled output  $z_p$  is given by:

$$z_p = F_l(F_u(P, \Theta(t)), F_l(K, \Theta(t)))w_p \quad (116)$$

## 4.6. Linear Control Applications (Simulation Results)

### 4.6.1. Application of Mixed Sensitivity $H_\infty$ Control (Simulation and Results)

The Bosch Scara robot with RRT kinematic structure is considered based on the experimental setup published in [73]. The first two joints and links are shown in Figure 4.9 and the third joint is considered to be mechanically decoupled from the motions of the other joints. The inertia and Coriolis matrices are given as follows:

$$M(\theta) = \begin{bmatrix} I_1 + 2I_2 \cos(\theta_2) & I_3 + I_2 \cos(\theta_2) \\ I_3 + I_2 \cos(\theta_2) & I_3 \end{bmatrix} \quad (117)$$

$$C(\theta, \dot{\theta}) = \begin{bmatrix} -2I_2 \sin(\theta_2) \dot{\theta}_1 \dot{\theta}_2 - I_2 \sin(\theta_2) \dot{\theta}_1^2 \\ I_2 \sin(\theta_2) \dot{\theta}_1^2 \end{bmatrix} \quad (118)$$

The complete system of the first two links is shown in Figure 4.9, including servo motors with gearboxes, and the dynamic cross-coupling torques (Coriolis effects). A spring-damper is introduced to model the torsion stiffness of the robot shaft between each DC motor and the link. The dynamic coupling appears in the joint systems as torques on the joint axes and is considered as an independent disturbance torque. The nominal values of the robot parameters are estimated at the null position of the first two joints shown in Figure 4.10 and given in Table A.1. The state equations of the first link is derived as follows:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{1}{J_{L1}} (K_s x_3 + D_s N_1 x_4 - F_{v1} x_2 + \tau_{DL}) \\ \dot{x}_3 &= N_1 x_4 - x_2 \\ \dot{x}_4 &= \frac{1}{J_{m1}} (K_{m1} x_5 - N_1 k_5 x_3 - F_{v1} x_4) \\ \dot{x}_5 &= \frac{1}{L_{m1}} (-R_{m1} x_5 - K_{m1} x_4 + x_6 + K_{p12} u - K_{p12} x_5) \\ \dot{x}_6 &= -k_{i12} x_5 + k_{i12} u \\ y &= x_1 \end{aligned} \quad (119)$$

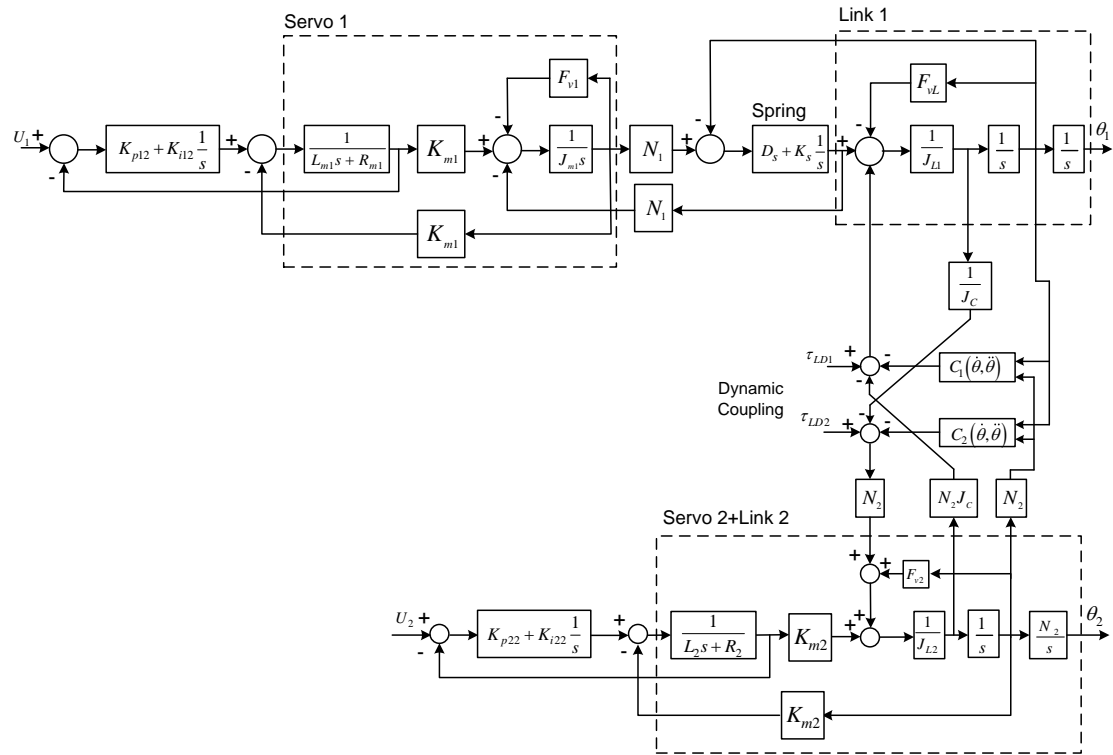


FIGURE 4.9. Model of first two links and servo motors of the Bosch Scara robot.

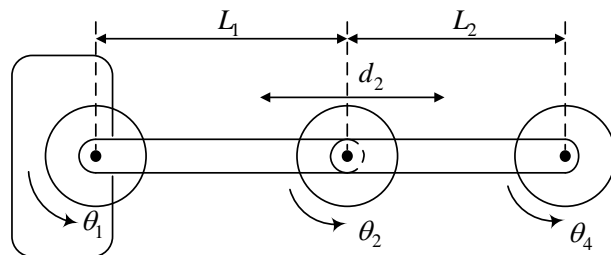


FIGURE 4.10. Schematic top view of the Bosch Scara robot in null position.

The state space and model description of the Bosch Scara robot are given in Appendix B, in which the input  $u$  is the motor torque and the output  $y$  is the joint angle of first link. Mixed sensitivity  $H_\infty$  controllers are designed and analyzed based on this model in Subsection 4.6.1.

For the nominal model derived in equation (119), the input weight is selected to be about 1 or less to bound the magnitude of the input signal, and therefore a simple weight  $w_u = 1$  is selected. The performance weight is selected in the form :

$$w_{p_1} = \frac{s/M + w_B}{s + w_B A}; \quad M = 1.5, \quad w_B = 10, \quad A = 10^{-4} \quad (120)$$

The value  $w_B = 10$  has been selected to achieve approximately the desired crossover frequency  $w_c = 10 \text{ rad/s}$ . The  $H_\infty$  problem is solved with the  $\mu$ -toolbox in Matlab. The simulation results are as follows: An optimal  $H_\infty$  norm of 0.9856, so the weighted sensitivity requirements are almost satisfied with  $\|S\|_\infty = 1.15$ ,  $\|T\|_\infty = 1.0$  and  $w_c = 9.96 \text{ rad/s}$ . This design shows the tracking is very good as shown by curve  $y_1$  in Figure 4.12 (Up), but from the curve  $y_1$  in Figure 4.12 (down), the disturbance response is very sluggish. In case of the disturbance rejection is the main concern, a performance weight can be selected to specify higher gains at low frequencies:

$$w_{p_2} = \frac{(s/M^{1/2} + w_B)^2}{(s + w_B A^{1/2})^2}; \quad M = 1.5, \quad w_B = 10, \quad A = 10^{-4} \quad (121)$$

The inverse of this weight is shown in Figure 4.11 and from the dashed line to cross 1 in magnitude at about the same frequency as weight  $w_{p_1}$ , but specifies tighter control at lower frequencies. With the weight  $w_{p_2}$ , the design with an  $H_\infty$ -optimal norm of 1.32, is yielding  $\|S\|_\infty = 1.35$ ,  $\|T\|_\infty = 1.3$  and  $w_c = 15.30 \text{ rad/s}$ . In conclusion, the first design is best for reference tracking whereas design 2 is best for disturbance rejection.

#### 4.6.2. Application of $H_\infty$ Control (Simulation and Results)

The parameter variation and structural dynamics are important properties of the reconfigurable dynamic system described by Equation (71). The inertia variations and the nonlinear behavior of the viscous friction can be modeled as parametric uncertainties bounded with upper and lower limit values. The nominal values of these parameters are measured and given in Tables (A.1 and A.2) for different joints links of the robot. The dynamic coupling affects the first link. The Coriolis torque, cross coupling inertia, and centrifugal torques are regarded as disturbance torque  $\tau_{LD1}$ . Figure 4.13 shows that the uncertain parts of the viscous friction (damping

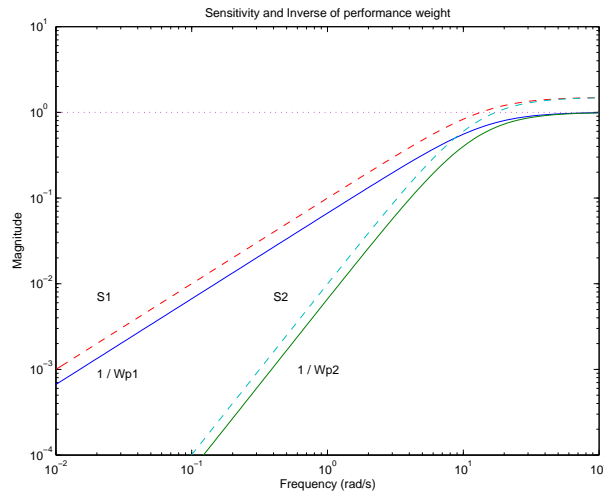


FIGURE 4.11. Inverse of performance weight (dashed line) and the resulting sensitivity function (solid line) for two  $H_\infty$  designs (1 and 2) for disturbance rejection.

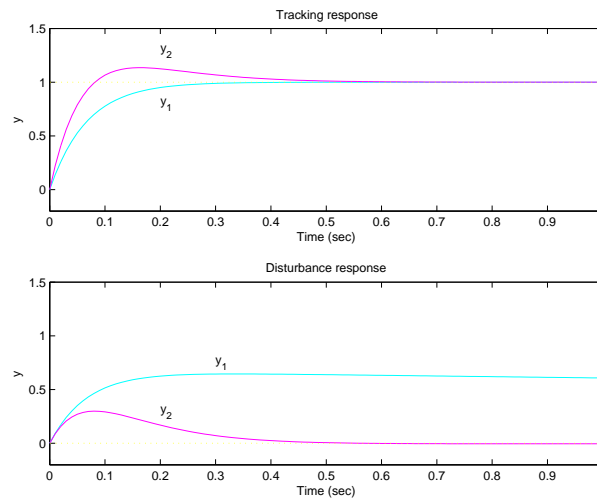


FIGURE 4.12. Closed loop step responses for two alternative designs (1 and 2) for disturbance rejection problem.

part) and link inertia are pulled out of the nominal system. The inertia variation of the first link is given in Table A.3 for different joint positions ( $\theta_2 = 0$ ,  $\theta_2 = \pi$ ) of link 2. These variations are modeled as multiplicative uncertainty as is the friction. The state equations of the perturbed first link system is derived as follows:

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= q_1 + \frac{1}{J_{L1}} (K_s x_3 + D_s N_1 x_4 - F_{v1} x_2 - q_2 + \tau_{DL1})\end{aligned}$$

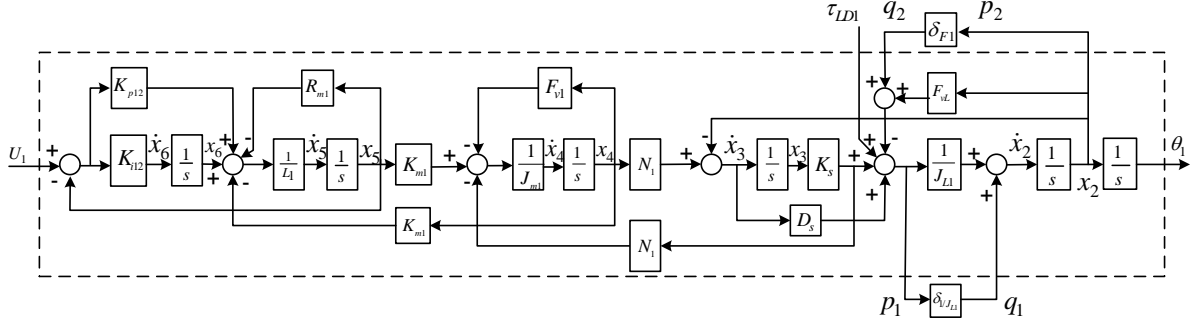


FIGURE 4.13. Derived block diagram of the first link of the Bosch Scara robot.

$$\begin{aligned}
 \dot{x}_3 &= N_1 x_4 - x_2 \\
 \dot{x}_4 &= \frac{1}{J_{m1}} (K_{m1} x_5 - N_1 k_5 x_3 - F_{v1} x_4) \\
 \dot{x}_5 &= \frac{1}{L_{m1}} (-R_{m1} x_5 - K_{m1} x_4 + x_6 + K_{p12} u - K_{p12} x_5) \\
 \dot{x}_6 &= -k_{i12} x_5 + k_{i12} u
 \end{aligned} \tag{122}$$

The outputs of the perturbed system are:

$$\begin{aligned}
 y &= x_1 \\
 p_1 &= q_1 + \frac{1}{J_{L1}} (K_s x_3 + D_s N_1 x_4 - F_{v1} x_2 - q_2 + \tau_{DL}) \\
 p_2 &= x_2
 \end{aligned} \tag{123}$$

Then, the state space representation is cast as follows:

$$\begin{bmatrix} \dot{x} \\ y_\Delta \\ y \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} x \\ u_\Delta \\ u \end{bmatrix} \tag{124}$$

The system  $G(s)$  includes all nominal parameter values of the model with diagonal uncertainty matrix  $\Delta = \text{diag}(\delta_J, \delta_V)$  as shown in Figure 4.14. The matrix  $\Delta$  is unknown and called the *uncertainty matrix* and has a fixed diagonal structure. The frequency response of the perturbed open loop system is computed for different values of the perturbed parameters for  $-1 \leq \delta_J, \delta_V \leq 1$  as shown in Figure 4.15.

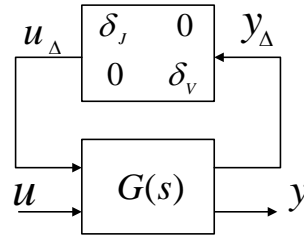


FIGURE 4.14. LFT representation of the perturbed Bosch Scara model

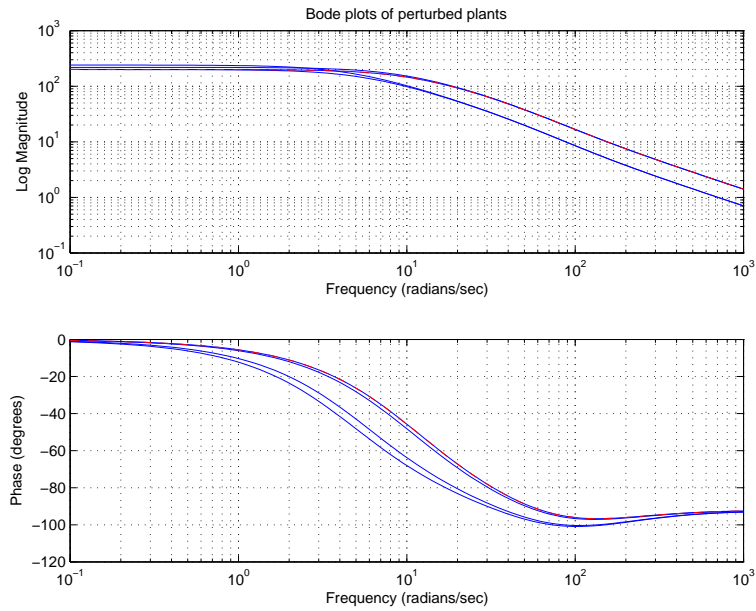


FIGURE 4.15. Perturbed (set of family) open loop systems.

### Design Requirements of Closed Loop System

The design objective is for a linear system  $G(s)$  to find a feedback control  $u(s) = K(s)y(s)$  which ensures the following properties of the closed loop system.

### Nominal Stability and Performance

The controller design should make the closed loop system internally stable and the required performance should be achieved for the nominal system  $G(s)$  by minimizing the following objective criterion:

$$\left\| \begin{array}{c} W_p(I + GK)^{-1} \\ W_u K(I + GK)^{-1} \end{array} \right\|_{\infty} < 1 \quad (125)$$

where  $S = (I + GK)^{-1}$  is the sensitivity function of the nominal system, and  $W_p$ ,  $W_u$  are weighting functions chosen to represent the frequency characteristics



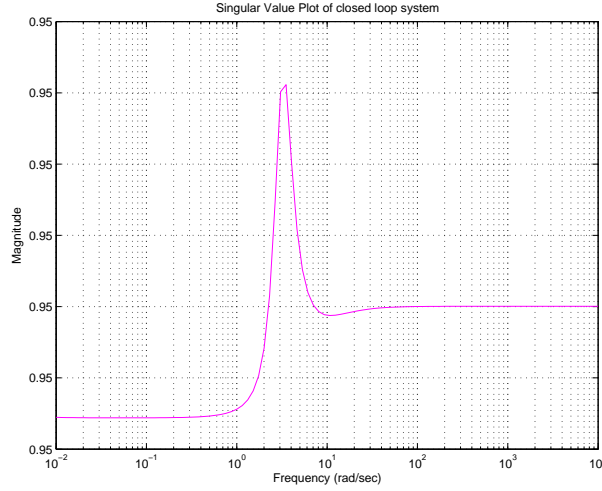


FIGURE 4.16. Singular value of the closed loop with  $H_\infty$  controller.

of the disturbance  $d$  and the input control level. The simulation results shown in Figure 4.16 indicate that the norm inequality has been satisfied and the closed loop system has reduced the effect of the disturbance  $\tau_{LD1}$  achieving the required performance.

### Selection of Weighting Functions

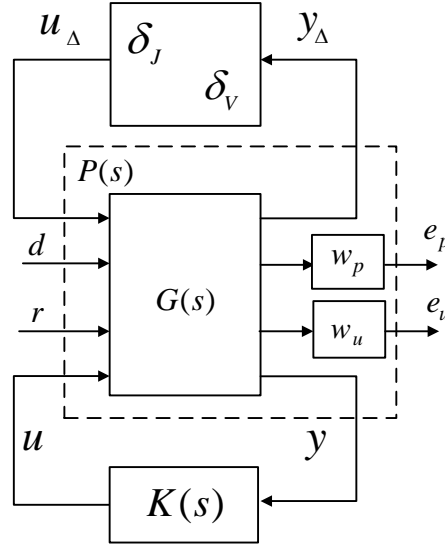
The weighting functions  $w_p$  and  $w_u$  are used to reflect the relative significance of the performance requirements over the frequency ranges. Finding appropriate weighting function is a crucial step in robust controller design and needs a few trials. The selected performance weighting function is a scalar function as:

$$W_p(s) = 0.95 \frac{s^2 + 1.8s + 10}{s^2 + 8s + 0.01} \quad (126)$$

which ensures for disturbance rejection and good transient response (settling time less than 10 and overshoot less than 20% for the nominal system). The control weighting function  $W_u$  is chosen as scalar  $W_u = 10^{-2}$ .

### $H_\infty$ -Controller Design

An  $H_\infty$  controller is designed using the system connection shown in Figure 4.17. The generalized plant  $P(s)$  includes the performance and uncertainty weightings. The inputs to the plant are  $u_\Delta, d, r, u$  and the outputs are  $y_\Delta, e_p, e_u, y$ . The controller minimizes the norm of  $F_l(P, K)$  over all stabilizing controllers, where the

FIGURE 4.17.  $H_\infty$  control structure.

transfer function matrix  $F_l(P, K)$  is:

$$\begin{bmatrix} d \\ r \end{bmatrix} = F_l(P, K) \begin{bmatrix} e_p \\ e_u \end{bmatrix} \quad (127)$$

To achieve the desired performance of disturbance rejection (or tracking reference), it is necessary to satisfy the inequality  $\|W_p(I + GK)^{-1}\|_\infty < 1$ . Since  $W_p$  is a scalar function, the singular values of the sensitivity function  $(I + GK)^{-1}$  over the frequency range must lie below that of  $\frac{1}{w_p}$ . This indicates  $\|W_p(I + GK)^{-1}\|_\infty < 1$  if and only if for all frequencies  $\sigma[(I + GK)^{-1}(jw)] < |1/W_p(jw)|$ . Figure 4.18 shows that the sensitivity function is below the performance weighting function for all frequencies.

### Analysis of Closed Loop System with $H_\infty$ -Controller

The robust stability has been analyzed based on the perturbed closed loop transfer function matrix  $F_l(P, K)$ . Since the uncertainty considered is structured, verification of the robust stability and robust performance needs the frequency response in terms of  $\mu$  values. To achieve robust stability it is necessary that the transfer function matrix  $(I - N_{11}K)^{-1}$  Equation (84) is not singular. This implies that the  $\mu(N_{11})$  must be less than one over the frequency range and that the closed

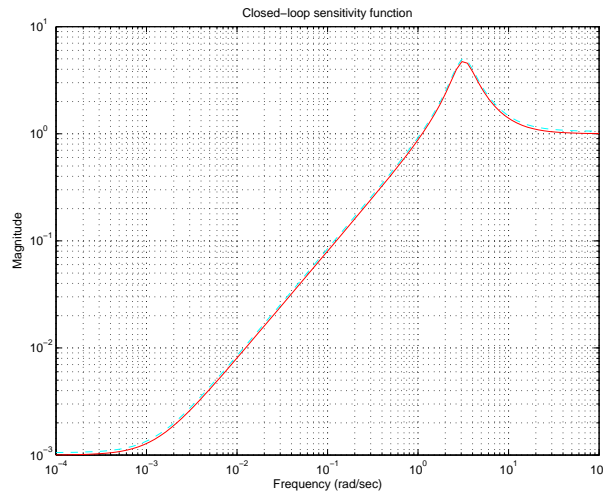


FIGURE 4.18. Sensitivity and performance weighting function with  $H_\infty$  controller.

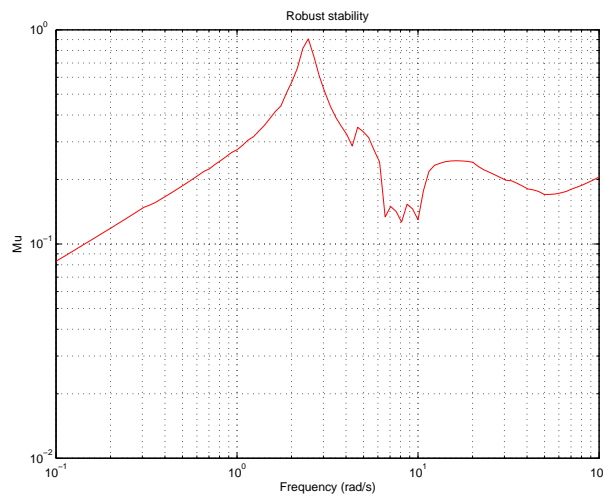
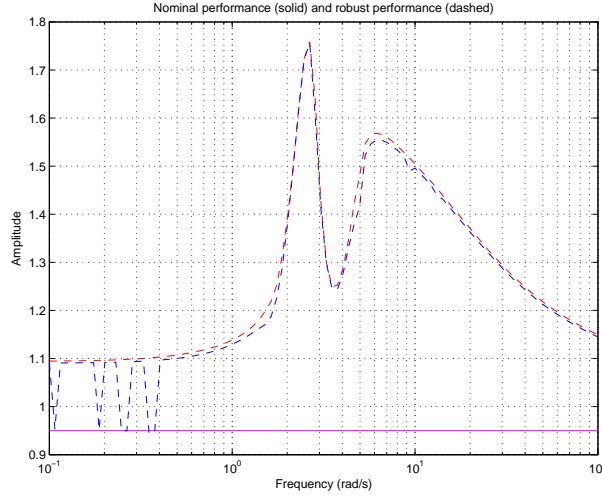


FIGURE 4.19. Robust stability analysis with  $H_\infty$  control.

loop system with  $H_\infty$  control achieves robust stability as shown in Figure 4.19. The maximum value of  $\mu$  is 0.90675 which implies that the structured perturbations with norm less than  $\frac{1}{0.90675}$  are allowable, i.e. the stability maintains for  $\|\Delta\|_\infty < \frac{1}{0.90675}$ . The nominal performance of the closed loop system is analyzed by means of the frequency response of the ( $N_{22}$ ) Equation (84). The nominal stability is achieved if and only if  $\mu(N_{22}) < 1$  for all frequency range. The robust performance of the closed loop system with  $H_\infty$  control is also tested by means of  $\mu$ -analysis. The block uncertainty structure includes  $2 \times 2$  diagonal parametric uncertainty block and  $1 \times 2$  performance block as follows:

FIGURE 4.20. Nominal and robust performance with  $H_\infty$  control.

$$\tilde{\Delta} = \left\{ \begin{bmatrix} \Delta & 0 \\ 0 & \Delta_p \end{bmatrix} : \Delta \in \mathbb{R}^{2 \times 2}, \Delta_p \in \mathbb{C}^{1 \times 2} \right\} \quad (128)$$

The robust performance of the closed loop system is achieved if and only if is less than one for each frequency. The frequency responses showing the nominal and robust performance are plotted in Figure 4.20. The frequency responses indicate that the system achieves the nominal performance with  $\mu(N_{22}) < 1$  but fails to satisfy the robust performance criterion. From the calculations, the nominal performance has maximum of 0.94998 while the  $\mu$  curve (blue dotted line is the  $\mu$  value and the red dotted line is the maximum singular value) for the robust performance has a maximum of 1.7584. With respect to the robust performance, the size of perturbation matrix  $\Delta$  must be limited to  $\|\Delta\|_\infty \leq \frac{1}{1.7584}$  to ensure the perturbed performance function satisfying:

$$\mu_{\tilde{\Delta}}(N) < 1, \forall(w, \tilde{\Delta})$$

The frequency responses of the perturbed closed loop systems are shown in Figure 4.21. The step and disturbance responses are shown in Figures 4.22 and 4.23, respectively. In both cases, the overshoot does not exceed 20% which demonstrates satisfactory performance in the presence of parametric perturbations.

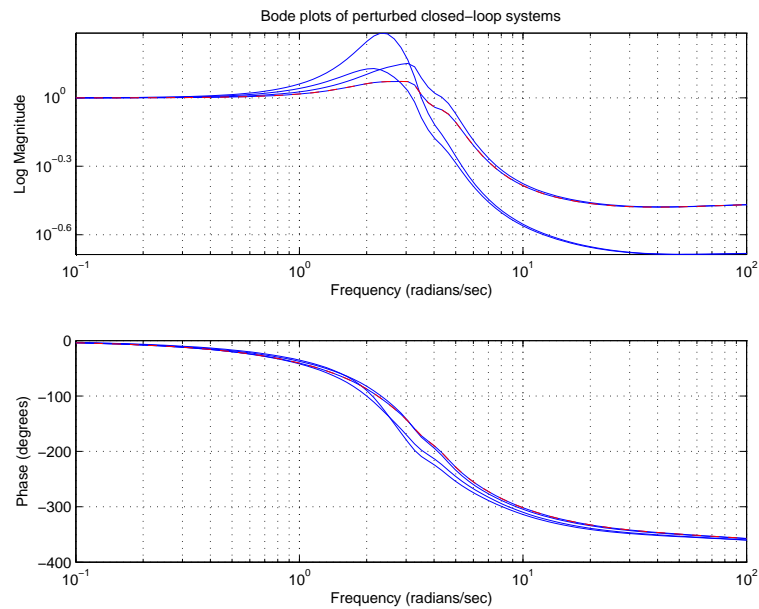


FIGURE 4.21. Perturbed (set of family) of closed loop systems with different uncertainty values.

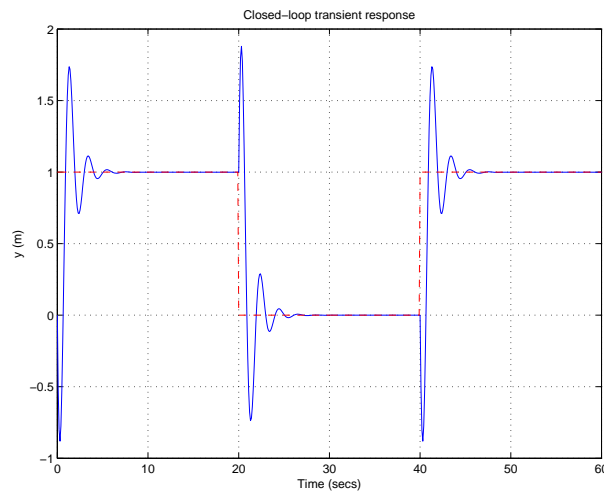


FIGURE 4.22. Transient response to reference input with  $H_\infty$  control.

#### 4.6.3. Application of $\mu$ -Synthesis Control and DK Iterations (Simulation and Results)

The uncertainty block  $\Delta$  given in Equation (128) is diagonal and corresponding to the inertia and viscous damping uncertainties of the robot link. The block  $\Delta_p$  is an uncertainty block that is introduced to represent the performance requirements in the control structure of the  $\mu$ -approach. The following optimization problem

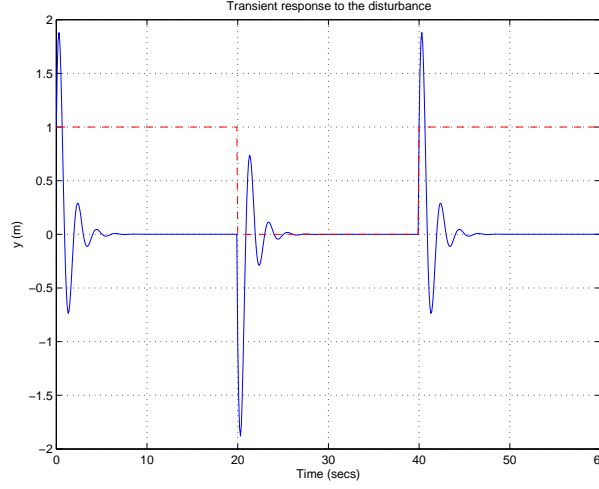


FIGURE 4.23. Transient response to disturbance input with  $H_\infty$  control.

is formed to minimize the upper bound of  $\mu$  values which in turn reduced the maximum value of  $\mu$ .

$$\min_K \min_{D_l(s), D_r(s)} \| D_l(s) F_l(P, K) D_r^{-1} \|_\infty \quad (129)$$

where

$$D_l(s) = \begin{bmatrix} d_1(s) & 0 & 0 \\ 0 & d_2(s) & 0 \\ 0 & 0 & d_3(s)I_2 \end{bmatrix} \quad (130)$$

and

$$D_r(s) = \begin{bmatrix} d_1(s) & 0 & 0 \\ 0 & d_2(s) & 0 \\ 0 & 0 & d_3(s)I_2 \end{bmatrix} \quad (131)$$

where  $d_1(s)$ ,  $d_2(s)$  and  $d_3(s)$  are scaling transfer functions. Then,  $\mu$ -synthesis is to find a minimum value of the cost function and construction of a stabilizing controller  $K$  such that for each frequency  $w \in [0, \infty]$  the structured singular value satisfy the condition:

$$\mu_{\Delta}[F_l(P, K)(jw)] < 1 \quad (132)$$

Satisfying the above condition ensures robust performance of the resulting closed loop.

### Robust Stability and Performance of $\mu$ -Control

$\mu$ -Control has been employed to achieve robust stability and performance against structured uncertainties. From the iteration summary, it is seen that the value of  $\gamma$  decreases to 0.97, which means that the robust performance has been achieved.

#### Iteration Summary

Iteration #	9	10	11	12	13
Controller Order	17	15	15	13	13
Total D-Scale Order	12	10	10	8	8
Gamma Achieved	1.020	1.013	1.016	1.013	1.01
Peak mu-Value	1.021	1.012	1.007	1.000	0.97
MU iteration number:	14				

The nonlinear properties of the inertia and viscous parameters are modeled as parametric uncertainties in addition to full block representing the performance channels. Figure 4.24 shows the sensitivity function of the closed loop system with the 13th-order controller. The sensitivity function is below the inverse of the performance weighting function, which implies that the nominal performance is achieved. The robust stability of the closed loop system is analyzed by the magnitude of the upper and lower bounds of  $\mu$  as shown in Figure 4.25. The robust stability of the closed loop system is achieved since the maximum value of  $\mu$  is equal to 0.49734. i.e. the system stability is preserved for  $\|\Delta\|_\infty < \frac{1}{0.49734}$ . The frequency responses of the nominal and robust performance criteria are obtained as shown in Figure 4.26.

The maximum value of  $\mu$  in the robust performance analysis is 0.97512. This means that the closed loop system with  $\mu$ -controller achieves robust performance since:

$$\left\| \begin{array}{c} W_p(I + GK)^{-1} \\ W_u K(I + GK)^{-1} \end{array} \right\|_\infty < 1 \quad (133)$$

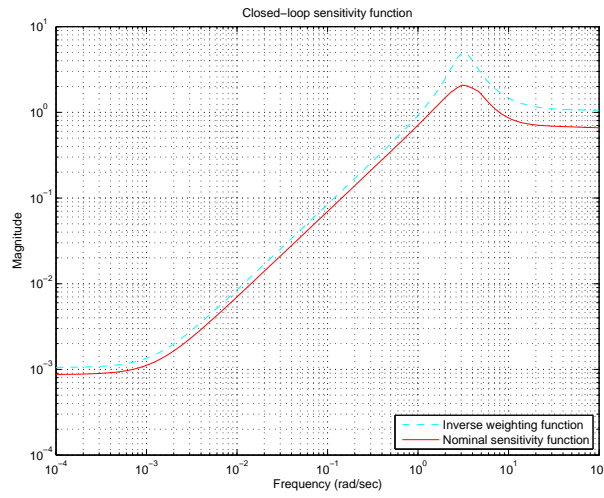


FIGURE 4.24. Sensitivity and weighting functions of Mu-Control.

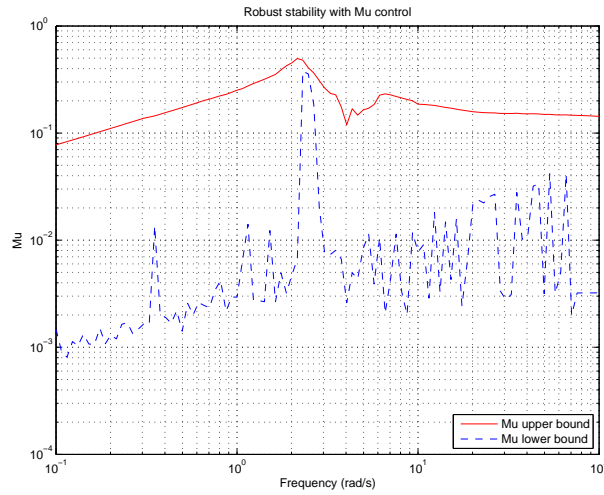


FIGURE 4.25. Robust stability of Mu-control

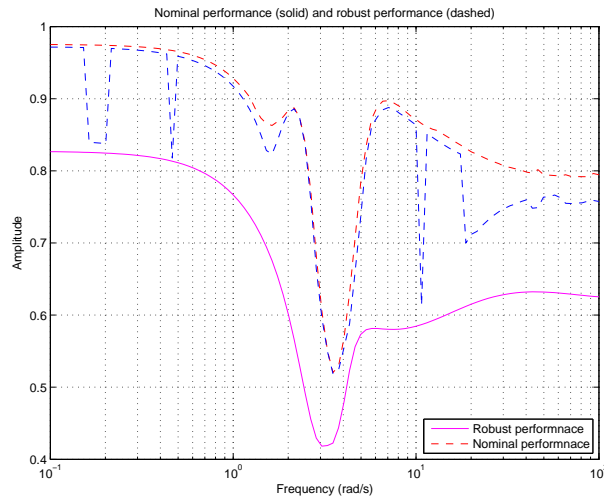


FIGURE 4.26. Nominal and robust performance of Mu-Control.



for every diagonal  $\Delta$  with  $\|\Delta\|_\infty < 1$ . The frequency response of the sensitivity functions of the perturbed closed loop systems Figure 4.27, shows the robust properties of system with the  $\mu$ -controller. These responses remain below the frequency response of the inverse of the performance weighting function. The magnitude response of the wighted mixed sensitivity function Equation (133) are shown in Figure 4.28.

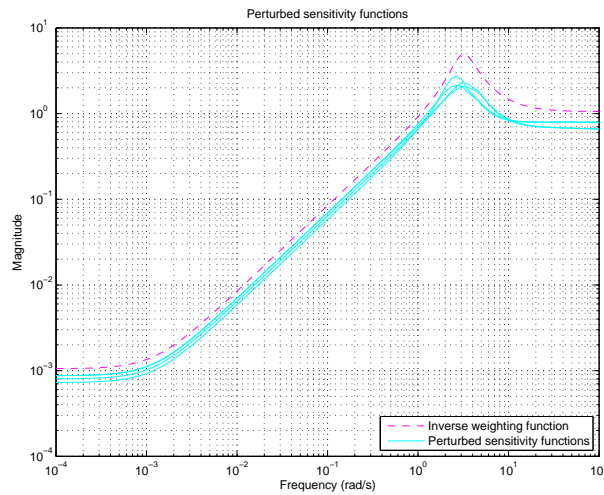


FIGURE 4.27. Sensitivity functions of perturbed systems with Mu-Control.

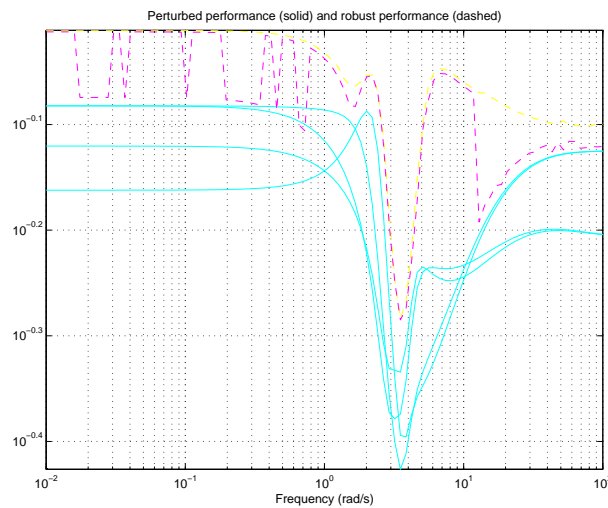


FIGURE 4.28. Performance of perturbed systems with Mu-Control.

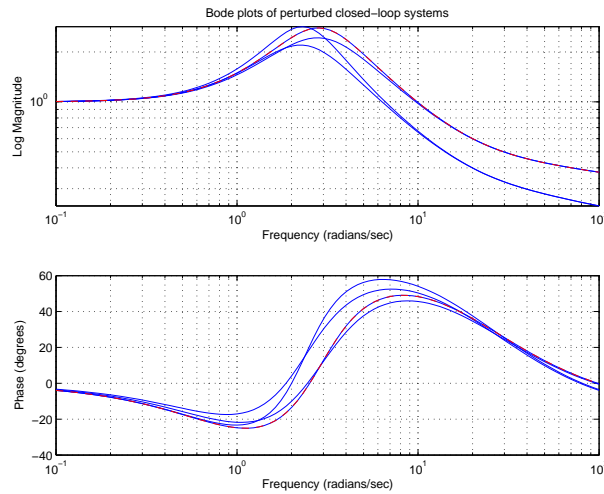


FIGURE 4.29. Frequency responses of perturbed closed loop systems with Mu-Control.

The robust performance has been satisfied for all perturbed systems, because the magnitudes over frequency range are below 1. The frequency responses of the perturbed closed loop systems are shown in Figure 4.29. We see from the figure that the closed loop perturbed systems maintain their magnitude over a wider frequency bandwidth. This would expect faster responses with the designed closed loop system. Figures 4.30 and 4.31 show the transient responses of the closed loop system to reference and disturbance inputs, respectively. Comparing with th responses Figures 4.22 and 4.23, we see that the  $\mu$ -controller ensures smaller overshoot (10 %) while maintaining the similar settling time. Figure 4.32 shows the transient responses (to input reference) of a family of perturbed closed loop systems with  $\mu$ -controller. In all cases the overshoot does not exceed 20 % which demonstrates satisfactory performance in the presence of parametric perturbations.

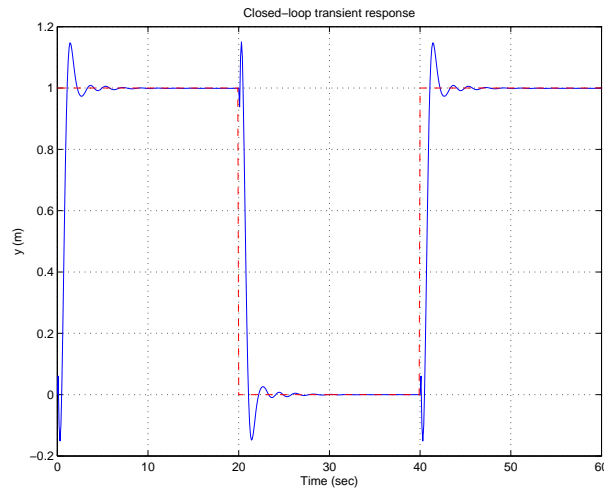


FIGURE 4.30. Transient response to step reference input with Mu-Control.

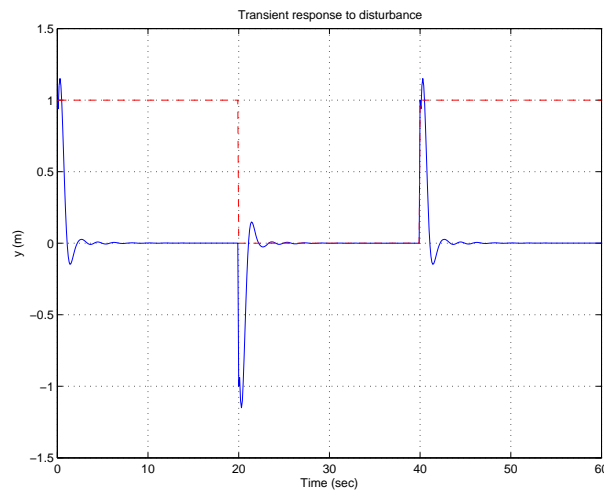


FIGURE 4.31. Transient response to disturbance input with Mu-Control.

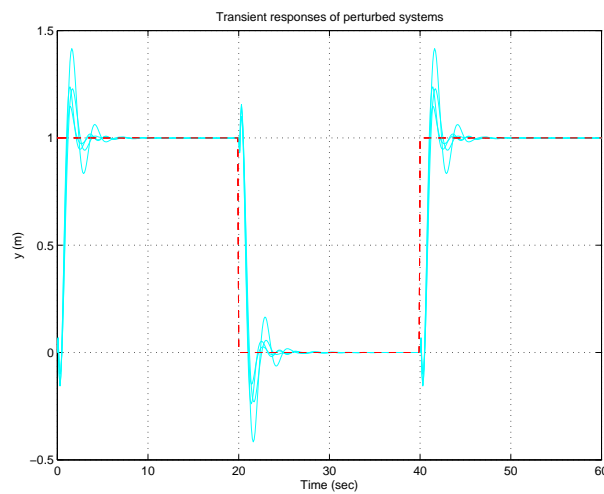


FIGURE 4.32. Transient responses of perturbed closed loop with Mu-Control.

#### 4.6.4. Comparison of $H_\infty$ and $\mu$ -Controllers

In this subsection, a comparison between  $H_\infty$  and  $\mu$  controller are given based on criteria such as robust stability, nominal and robust performance. The comparison of the designed systems with  $H_\infty$  and  $\mu$  controller begins with the frequency responses of these controllers.  $\mu$ -control is characterized by larger gains in the frequency range above 10 rad/s compared with the  $H_\infty$  control as shown in Figure 4.33. The phase response are close to each other up to 3 rad/s and after that frequency the  $\mu$  controller continues to introduce a larger phase delay. The closed loop system with the  $H_\infty$  and  $\mu$  controllers are characterized by larger bandwidth that leads to faster dynamics in response to reference inputs as shown in Figure 4.34.

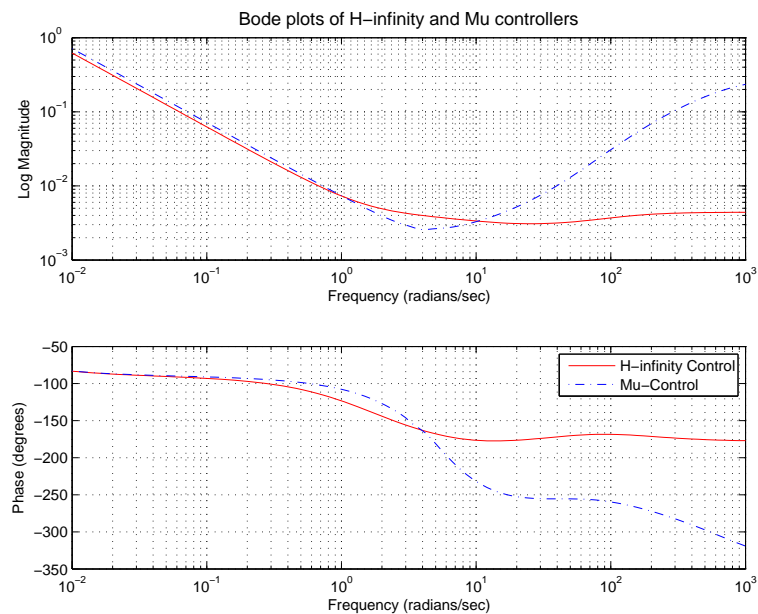


FIGURE 4.33. Frequency responses of  $H_\infty$  and  $\mu$ -controllers.

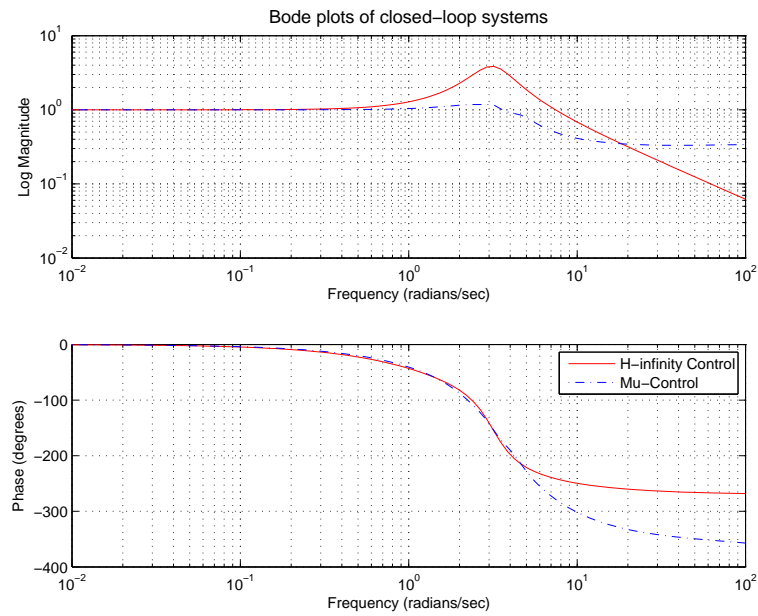
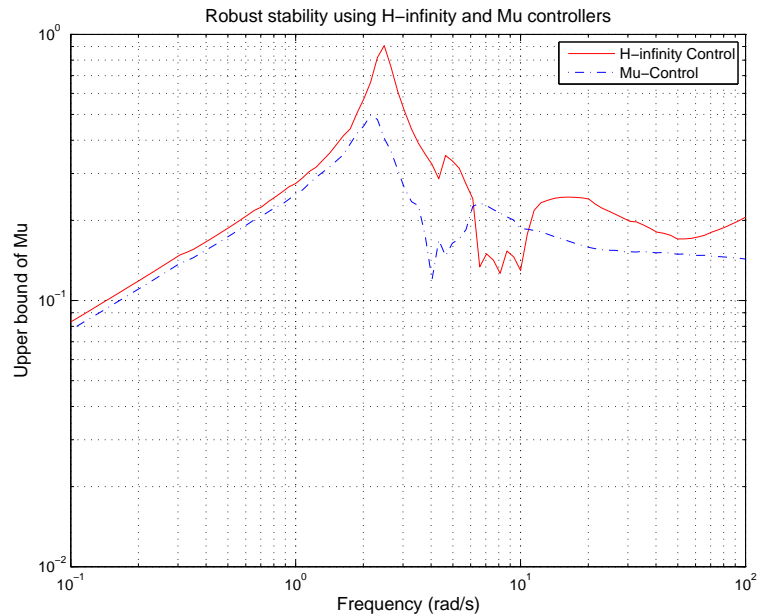


FIGURE 4.34. Frequency responses of closed loop systems.

FIGURE 4.35. Comparison of robust stability of  $H_\infty$  and  $\mu$ -controllers.

The comparison of robust stability is shown in Figure 4.35. The frequency response of  $\mu$  curve shows lower amplitude than the  $H_\infty$  control. Therefore, the system with the  $\mu$  controller allows larger norm of perturbations and maintains the robust stability. Figure 4.36 shows the nominal performance amplitude using  $\mu$  controller is less than the  $H_\infty$  curve, resulting in better performance. The  $\mu$  values over the

frequency for the two controller are plotted in Figure 4.37. The curves confirm that the system with  $H_\infty$  controller does not achieve the robust performance criteria. The  $\mu$  curve is less than one which indicates that the robust performance has been achieved against the specified structured uncertainties.

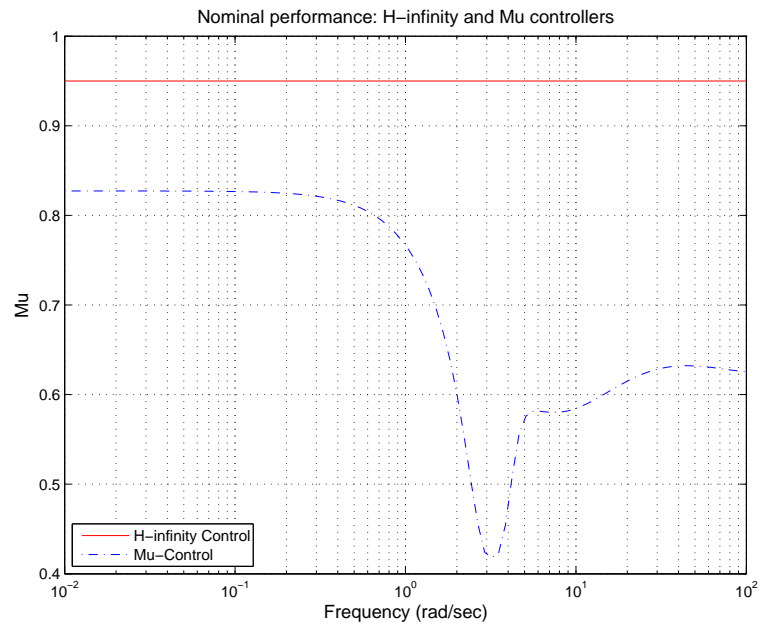


FIGURE 4.36. Comparison of nominal performance of  $H_\infty$  and  $\mu$ -controllers.

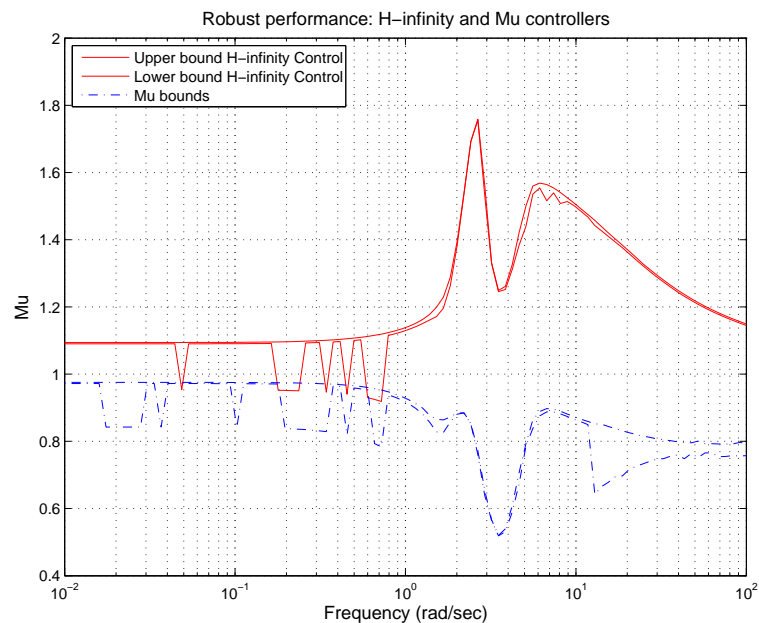


FIGURE 4.37. Comparison of robust performance of  $H_\infty$  and  $\mu$ -controllers.

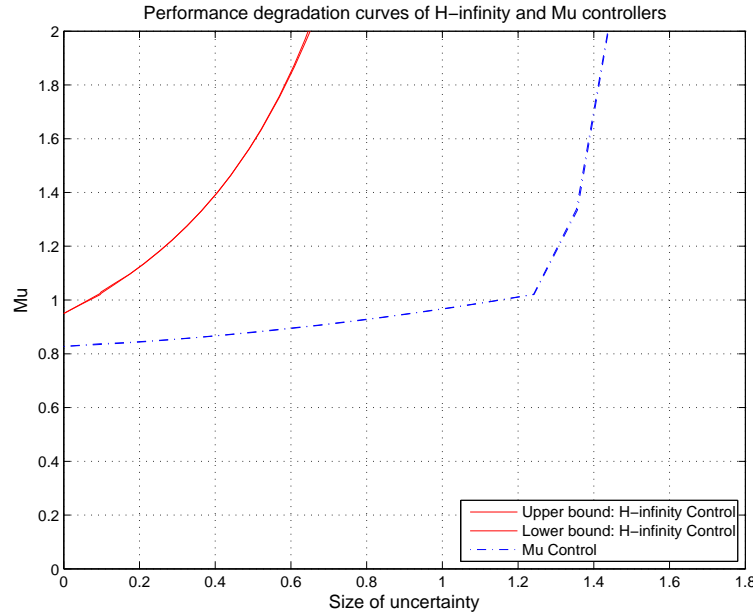


FIGURE 4.38. Performance degradation of  $H_\infty$  and  $\mu$ -controllers.

In summary, the two controllers ensure robust stability of the closed loop system with respect to the parametric perturbations that included in the  $2 \times 2$  diagonal uncertainty matrix. However, the closed loop system performance varies in a different manner than the action of these diagonal uncertainties. In the following, the worst-case perturbation is determined with respect to performance and to compare the two systems when the norm of perturbations increases. The results in Figure 4.38 show that the  $\mu$  ensure robust performance for large perturbations. The performance of the closed loop system deteriorate rapidly with the increasing of perturbation magnitude in case of  $H_\infty$  controller. From the simulations above, using  $\mu$ -controller in case of structured uncertainties will always produce more more satisfactory performance and less conservative controller.

#### 4.6.5. Order Reduction of $\mu$ Control

As given in the iteration summary, the order of the  $\mu$ -controller is 13, which makes it difficult to implement. The Hankel-norm approximation method is used and implemented to reduce the  $\mu$ -controller order. Applying the method, generates a balanced realization by removing the unobservable and uncontrollable modes, if the system is not minimal. The Hankel singular values of the system can be used in selecting the order of reduced controller.

SVD(H) =

2.869862534849350  
 0.286015061736286  
 0.173452442893747  
 0.119289861739874  
 0.006086495977984  
 0.000706061440223  
 0.000259732952475  
 0.000067884568817  
 0.000066859740534  
 0.000025222140549  
 0.000024129302080  
 0.000003363897582  
 0.000002670567341  
 0.000000242490855  
 0.000000022118556  
 0.000000003521916  
 0.000000002664316

some of these values are very small, which implies that the controller (13th-order) can be greatly reduced. A 4th-order reduced controller is tested and plotted with full order one in Figure 4.39. Actually, a 3rd-order controller has been tried as well, but a noticeable differences in th frequency responses in the range  $10^{-1} - 10^1$  rad/s. The system matrices of this fourth order controller are:

Ak =

1.0e+007 \*

-1.10885243042	-0.20413296615	-0.00661216668	0.00001181105
0	-0.01072683827	-0.00000616026	0.00000087279
0	0	-0.00008493253	0.00000009118
0	0	0	-0.00000000012

Bk =



```

1.0e+003 *
2.523918720864480
0.173155981648059
0.016190561288115
-0.000084889826165

```

Ck =

```

1.0e+003 *
2.51310707644 0.29103575422 -0.00129673347 -0.00008032023

```

Dk =

```

0.00567015468

```

The frequency responses coincide with each other for low frequencies which implies that the same closed loop performance for both controllers. The transient responses of the closed loop systems with full and reduced order are indistinguishably comparing the Figure 4.39 with Figure 4.30. Obviously, the implementation of the 4th order controller is much easier compared to the 13th order one.

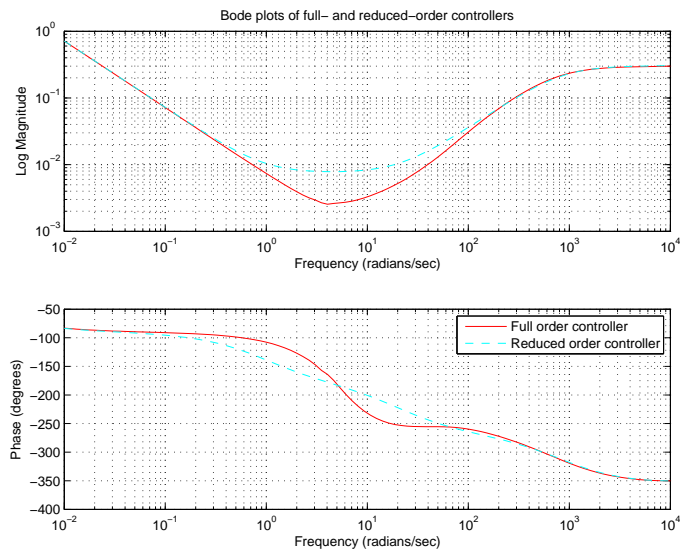


FIGURE 4.39. Frequency responses of full and reduced order controllers.

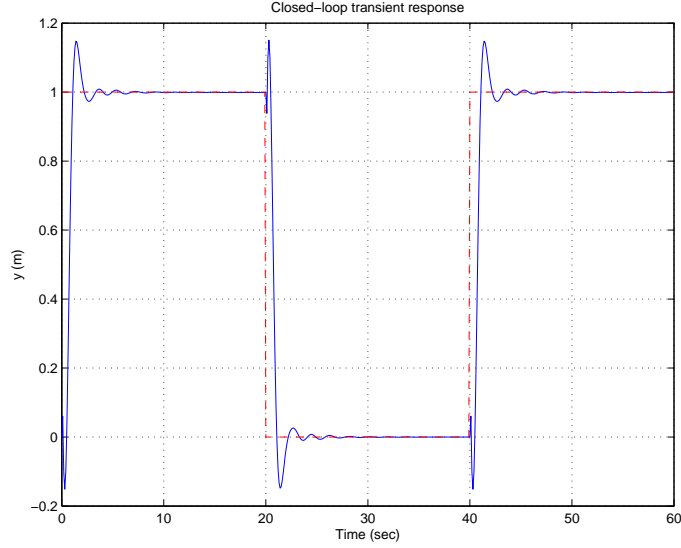


FIGURE 4.40. Transient response for reduced order controller.

#### 4.6.6. Application of LPV Control (Simulation and Results)

In this subsection, an LPV polytopic control is designed for the LPV linear system given in Equation (119) with two parameters configuration dependent  $M(q)$  and  $F(\dot{q})$ . These two coefficients, given in Table A.3, satisfy  $M(q) \in [M_{\min}, M_{\max}]$  and  $F(\dot{q}) \in [F_{\min}, F_{\max}]$ . Assuming that  $M(q)$  and  $F(\dot{q})$  are on-line measurable parameters, the controller is allowed to incorporate these measurements in the same fashion as the system. The resulting LPV controller exploits all the available measurements of  $M(q)$  and  $F(\dot{q})$  to provide a smooth and automatic gain scheduling. The LPV control design procedure is similar to the  $H_\infty$  loop shaping approach except that the closed loop functions are now parameter dependent. The LPV control structure displayed in Figure 4.41 consists of the LPV linear system  $P(\theta(t))$  and three LPV polytopic controllers. The reference velocity  $\dot{q}_d$  and position  $q_d$  are fed directly to the LPV feedforward controllers  $K_1(\theta(t))$  and  $K_2(\theta(t))$ , respectively, while the robot configuration position  $q$  is fed back to the LPV feedback controller  $K_3(\theta(t))$ . The LTI performance function  $W_p$  is chosen to weight the resulting error  $e$  between the reference position  $q_d$  and measured joint angle position  $q$ . The function  $K_3(\theta(t))S(\theta(t))$  is weighted using an LTI input weighting function  $W_u$  to ensure robustness against unmodeled dynamics. The LPV control objectives are as follows:

- (1) To get internal stability of the closed loop system.

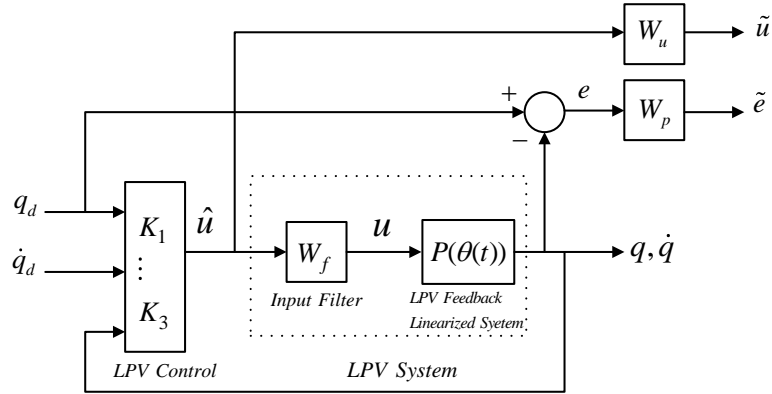


FIGURE 4.41. LPV control structure includes the LPV system  $P(\theta(t))$ , performance weighting functions  $W_p$ , robustness function  $W_u$ , and LPV polytopic controllers ( $K_1(\theta(t)), \dots, K_3(\theta(t))$ ) and input filter  $W_f$ .

- (2) To enforce the performance and robustness requirements by minimizing the  $L_2$ -gain of the closed loop performance channel.

These objectives should be satisfied for the time varying trajectories  $M(q)$  and  $F(\dot{q})$ . The design procedure is performed with describing the LPV system  $P(\theta(t))$  Equation (119) by two affine parameter-dependent models. Using the LPV loop shaping procedure, the resulting LPV polytopic system is placed within a polytope convex hull of four vertex systems  $Co\{P_i, i = 1, \dots, 4\}$ . The vertices  $P_i$  are the values of  $P(\theta(t))$  at the four vertices (the four corners  $P_1, \dots, P_4$ ) of the following parameter box:

$$\begin{aligned}
 p_1 &= \begin{bmatrix} M_{\max} \\ F_{\max} \end{bmatrix}, & p_2 &= \begin{bmatrix} M_{\min} \\ F_{\max} \end{bmatrix} \\
 p_3 &= \begin{bmatrix} M_{\max} \\ F_{\min} \end{bmatrix}, & p_4 &= \begin{bmatrix} M_{\min} \\ F_{\min} \end{bmatrix}
 \end{aligned} \tag{134}$$

The LPV synthesis problem illustrated in Figure 4.41 is solved using the Matlab/LMI control toolbox. To solve this problem, the input should be parameter independent [43]. This condition is satisfied by prefiltering the control input  $u$

with a high pass filter  $W_f$  as follows:

$$W_f = \frac{(s + 2\pi * 100 * 0.4)}{(1/2000 s + 2\pi * 100)}$$

The optimization problem is to find an LPV controller to minimize

$$\left\| \begin{array}{c} W_p S \\ W_u K S \end{array} \right\|_{L_2} \quad (135)$$

### Weight Selection

The performance weighting functions  $W_p$  and  $W_u$  are designed to enforce the performance and robustness specifications in Equation (135). An appropriate scaling of the system has been performed so that the input about or less than one in magnitude, and therefore select a simple input weight  $W_u = 1$ . The performance weight is chosen in a form as follows:

$$W_p = \frac{(s/M + w_c)}{s + w_c A} \quad (136)$$

The value  $w_c = 10$  has been selected to achieve approximately the desired crossover frequency  $w_c$  of  $10 \text{ rad/s}$ . The steady state error requirement is determined by the selection of the parameter value of  $A$ , which is chosen to be  $A = 10^{-4}$ .

### Analysis of LPV Control Design

The two parameter  $M(q)$  and  $F(\dot{q})$  are frozen to some values in the parameter box specified in Equation (134). The LPV close loop system is simulated for frozen parameters 10%, 30%, 60% and 90% of their nominal values. The step response of the gain scheduled system is simulated along the following spiral parameter trajectory shown in Figure 4.42:

$$\begin{aligned} M(q) &= 2.25 + 1.7e^{-4t} \cos(100t) \\ V(q) &= 50 + 49e^{-4t} \sin(100t) \end{aligned} \quad (137)$$

The step response shown in Figure 4.43 indicates the the performance requirements are satisfied in terms of the steady state and speed of response.

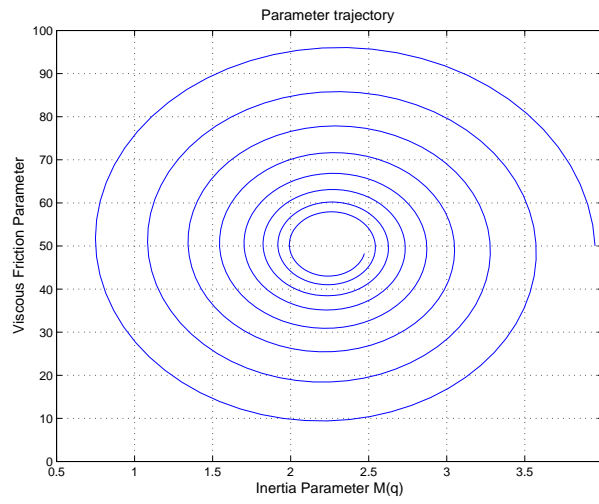


FIGURE 4.42. Parameter trajectory.

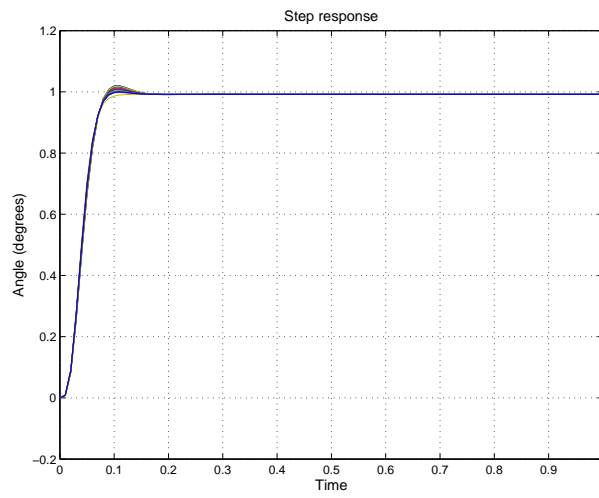


FIGURE 4.43. Step response of the LPV closed loop system.

## CHAPTER 5

### Nonlinear Control

In Chapter 5, the mechanical nonlinearities of the robot arm have been considered as torque disturbance within bounded limits between  $13.01 \leq \tau_{LD} \leq 53.32$ . Optimal robust controllers were created for a linearized robot, such that the controlled system can track the desired pose with a sufficiently high bandwidth. In fact, the nonlinearity inherent in the arm dynamics is so dominant and influences the arm performance enough that it promotes the development of control approaches that incorporate it into the control design process. The theory that will be treated here provides a tool for reducing the influence of the nonlinearities of the arm dynamics by means of feedback. Feedback linearization control is regarded as inner loop control; a robust nonlinear control has been designed to drive the error dynamics to a sliding surface, achieving high tracking performance of the robotic arm.

#### General Properties of Nonlinear Arm Dynamics

The general model of an  $n$ -joint robotic arm considered in this chapter is rewritten and explained as follows:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) + \tau_d = \tau \quad (138)$$

where  $q \in R^n$  is the joint coordinate vector,  $M(q) \in R^{n \times n}$  is the inertia matrix,  $C(q, \dot{q}) \in R^n$  describes the centrifugal and Coriolis forces,  $G(q) \in R^n$  is the gravity,  $F(\dot{q}) \in R^n$  is the frictional forces,  $\tau \in R^n$  is the control torque, and  $\tau_d \in R^n$  is the disturbance torque.

Consequently, the characteristics of the above dynamic model have the following properties:

- Highly nonlinear: each element of (138) contains nonlinear factors such as trigonometric functions, and other mathematical models such as saturation, square functions, ..., etc.
- High degree of coupling between the dynamics of adjacent joints and links.
- Model uncertainty and time-variance: the load will vary when the robot moves the objects or when the joint friction torque changes over time.

The mathematical formation of these characteristics are:

- The inertia matrix  $M(q)$  is a positive definite, symmetric and bounded matrix, i.e. there exists positive constants  $m_1$  and  $m_2$  such that  $m_1 I \leq M(q) \leq m_2 I$ .
- The centrifugal and Coriolis matrix  $C(q, \dot{q})$  is bounded, i.e., there exists a known  $c_b(q)$  such that  $|C(q, \dot{q})\dot{q}| \leq c_b(q)\|\dot{q}\|$ .
- The matrix  $\dot{M} - 2C$  is a skew-symmetric matrix, i.e.,  $x^T(\dot{M} - 2C)x = 0$ , where  $x$  is a vector.
- The known disturbance is satisfied with  $\|\tau_d\| \leq \tau_M$ , where  $\tau_M$  is a known positive constant.

## 5.1. Position Control

The position control problem is considered by designing a PD-Gravity control based on a Lyapunov function [74]. The resulting control compensates the gravity element and assures step tracking, and linear and curvature trajectories satisfying the required performance specifications. In Section 5.1.1, the dynamics of a servo DC motor is combined with the link dynamics to form a compact servo mechanism description. The derivation of the position control is given in Section 6.20 and the simulation results have been presented in Subsections 5.1.3 and 5.1.4.

**5.1.1. Link Dynamics.** Let  $q_i$  denote the generalized coordinate variable of the hybrid joint and  $\tau_i$  is the actuator torques vector of the  $n$ -axes of an open kinematic structure robot. The general equation of motion can be rewritten as:

$$\sum_{j=1}^n M_{ij}(q)\ddot{q}_j + \left( \sum_{k=1}^n \sum_{j \neq k} c_{kj}(q)\dot{q}_k\dot{q}_j + \sum_{k=1}^n c_{kk}(q)\dot{q}_k^2 \right) + g_i(q) + b_i(\dot{q}) = \tau_i \quad i = 1 \dots n \quad (139)$$

Equation (139) can be recast in a compact form as follows:

$$M(q)\ddot{q} + c(q, \dot{q})\dot{q} + g(q) + b(\dot{q}) = \tau \quad (140)$$

Here  $M(q)$  is the inertia tensor matrix, the matrix  $c(q, \dot{q})$  defines the generalized Coriolis and centrifugal forces. The matrices  $g(q)$  and  $b(\dot{q})$  are the gravity and frictional generalized forces, respectively. To facilitate control of the arm, it is useful to reformulate the nonlinear dynamic model as a first-order system of  $2n$  equations called *state equations*. The key to transforming to the state-space form is isolation of the acceleration vector  $\ddot{q}$  in Equation (140). This can be easily done because the manipulator inertia tensor  $M(q)$  is symmetric and positive-definite and therefore nonsingular. The state equations of a robotic manipulator are given in the following form:

$$\begin{aligned} \dot{q} &= v \\ \dot{v} &= M^{-1}(q) [\tau - c(q, v) - g(q) - b(v)] \end{aligned} \quad (141)$$

### 5.1.2. PD-Gravity Control

For a reconfigurable robotic arm system, an  $n$ -axis proportional-derivative (PD) control is developed such that the generalized joint position  $q(t)$  of the closed loop system closely tracks a constant reference input  $r(t) = a$  for  $t \geq 0$ . This is called the regulator problem or set-point control problem. The nonlinear controller PD with an explicit gravity compensation term is given as:

$$\begin{aligned} e &= r - q \\ \tau &= Ke + L\dot{e} + g(q) \end{aligned} \quad (142)$$

Here  $K$  and  $L$  are  $n \times n$  symmetric positive definite matrices called the position and velocity gain matrices, respectively. Notice that torque  $\tau$  consists of terms proportional to the error and the derivate of the error plus a nonlinear gravity term,  $g(q)$ . The controller Equation (142) is substituted in Equation (141), and this yields the following state closed loop equations of motion:

$$\dot{q} = v$$



$$\dot{v} = M^{-1}(q) [K(r - q) + L(\dot{r} - v) - c(q, v) - b(v)] \quad (143)$$

Assume  $r(t) = a$  for  $t \geq 0$ , for some set point  $a$ . To analyze the resulting equilibrium points of the closed loop system, the right hand side of the Equation (143) is set to zero and solved for  $q$  and  $v$ . This yields:

$$\begin{aligned} v &= 0 \\ M^{-1}(q) [K(a - q) - c(q, 0) - b(0)] &= 0 \end{aligned} \quad (144)$$

The Coriolis  $c(q, 0) = 0$  and  $b(0) = 0$ , as they are functions of velocity with a zero value at  $v = 0$ . Thus,  $K(a - q) = 0$  and the closed loop system has a single equilibrium point:

$$\hat{x}^T = [a^T, 0^T] \quad (145)$$

Then, the resulting equilibrium point at  $\hat{x}^T = [a^T, 0^T]$  of the closed loop system represents the desired steady state solution of the nonlinear system when the reference input is  $r(t)$ . To satisfy the required performance of the closed loop using the PD controller (142), it is sufficient to prove that the equilibrium point  $\hat{x}^T$  is asymptotically stable and its domain of attraction encompasses the entire state space [46].

**Theorem:** Let  $\hat{x}^T = [q^T, v^T]$  be the solution of the robotic arm described in (141), assuming that  $\tau$  is computed using the control law in (142). If  $r(t) = a$  for  $t \geq 0$ , then the equilibrium point  $\hat{x}^T = [a^T, 0^T]$  is asymptotically stable and the domain of attraction is  $\Omega = R^{2n}$ . This means, for each  $x(0) \in R^{2n}$ :

$$x(t) \rightarrow \hat{x} \quad \text{as} \quad t \rightarrow \infty$$

Proof: the equilibrium point is first moved to the origin by letting,  $z = q - a$ . Then, the closed loop equations in (143) can be written in terms of  $z$  and  $v$  as:

$$\begin{aligned} \dot{q} &= \dot{v} \\ \dot{v} &= -M^{-1}(z + a) [Kz + Lv + c(z + a, v) + b(v)] \end{aligned} \quad (146)$$

Since  $c(z + a, 0) = 0$  and  $b(0) = 0$ , it follows that  $(z, v) = (0, 0)$  is an equilibrium point of the transform system (146). The Lyapunov function candidate (energy

function) is:

$$V(z, v) = \frac{1}{2}z^T K z + \frac{1}{2}v^T M(z + a)v \quad (147)$$

Since the inertia tensor  $M(q)$  is a continuously differentiable positive definite matrix and the position gain matrix  $K$  is also a positive definite matrix, it follows that Equation (147) is a valid Lyapunov function, [51]. It is shown that the evaluation of the function  $\dot{V}(z(t), v(t))$  decreases along the solutions of the state Equation (146) as follows:

$$\begin{aligned} \dot{V}(z, v) &= \frac{d}{dt} \left( \frac{z^T K z + v^T M(z + a)v}{2} \right) \\ \dot{V}(z, v) &= z^T K \dot{z} + v^T M(z + a)\dot{v} + \frac{v^T \dot{M}(z + a)v}{2} \\ \dot{V}(z, v) &= z^T K v - v^T [Kz + Lv + c(z + a, v) + b(v)] + \frac{v^T \dot{M}(z + a)v}{2} \\ \dot{V}(z, v) &= (z^T K v)^T - v^T [Kz + Lv + c(z + a, v) + b(v)] + \frac{v^T \dot{M}(z + a)v}{2} \\ \dot{V}(z, v) &= v^T K^T z - v^T [Kz + Lv + c(z + a, v) + b(v)] + \frac{v^T \dot{M}(z + a)v}{2} \\ \dot{V}(z, v) &= v^T K z - v^T [Kz + Lv + c(z + a, v) + b(v)] + \frac{v^T \dot{M}(z + a)v}{2} \\ \dot{V}(z, v) &= \frac{v^T \dot{M}(z + a)v}{2} - v^T [Lv + c(z + a, v) + b(v)] \\ \dot{V}(z, v) &= v^T \left[ \frac{\dot{M}(z + a)v}{2} - c(z + a, v) \right] - v^T [Lv + b(v)] \quad (148) \end{aligned}$$

The Coriolis and centrifugal torque term  $c(z + a, v)$  can be expressed in terms of  $\dot{M}(q)$  and a skew-symmetric matrix  $N(q, v)$ , yielding:

$$c(z + a, v) = \frac{1}{2}[\dot{M}(z + a, v) - N(z + a, v)]v \quad (149)$$

Substituting Equation (149) in Equation (148) yields:

$$\dot{V}(z, v) = \frac{1}{2}v^T N(z + a, v)v - v^T [Lv + b(v)] \quad (150)$$

The first term does not contribute to  $\dot{V}(z + a, v)$  because it is a skew matrix:  $N^T(q, v) = -N(q, v)$ , which is given as follows:

$$v^T N(z + a, v)v = \frac{1}{2}v^T N(z + a, v)v + \frac{1}{2}v^T N(z + a, v)v$$

$$\begin{aligned}
v^T N(z+a, v)v &= \frac{1}{2}v^T N(z+a, v)v + \frac{1}{2}[v^T N(z+a, v)v]^T \\
v^T N(z+a, v)v &= \frac{1}{2}v^T N(z+a, v)v + \frac{1}{2}v^T N^T(z+a, v)v \\
v^T N(z+a, v)v &= \frac{1}{2}v^T N(z+a, v)v - \frac{1}{2}v^T N(z+a, v)v \\
v^T N(z+a, v)v &= 0
\end{aligned} \tag{151}$$

Combining equations (150) and (151), the expression for  $\dot{V}(z, v)$  along the solution of the closed loop state equation in (143) reduces into:

$$\dot{V}(z, v) = -v^T [Lv + b(v)] \tag{152}$$

$$\dot{V}(z, v) = -[v^T Lv + v^T b(v)] \tag{153}$$

Since  $L$  is positive definite and the friction coefficient is nonnegative, then  $\dot{V}(z, v) \leq 0$  along the solution of closed loop system (143). From Equation (153), the following yields:

$$\begin{aligned}
\dot{V}(z, v) \equiv 0 &\Rightarrow v(t) \equiv 0 \\
&\Rightarrow \dot{v}(t) \equiv 0 \\
&\Rightarrow M^{-1}(z(t) + a, 0)Kz(t) \equiv 0 \\
&\Rightarrow z(t) \equiv 0
\end{aligned} \tag{154}$$

Consequently, the equilibrium point  $(z, v) = (0, 0)$  is asymptotically stable. The domain of attraction is the set  $\Omega$  where:

$$\begin{aligned}
\Omega &= \{(z, v) : V(z, v) < \rho\} \\
\Omega &= \left\{ (z, v) : \frac{z^T Kz + v^T M(z+a)v}{2} < \rho \right\}
\end{aligned} \tag{155}$$

Here,  $V(z, v)$  is a Lyapunov function and the asymptotic stability conditions [51] are satisfied on  $\Omega_\rho$  for every  $\rho > 0$ . Since  $K$  and  $M(z+a, v)$  are positive definite matrices,  $V(z, v) \rightarrow \infty$  as  $\|z\| + \|v\| \rightarrow \infty$ . Thus, the domain of attraction is the entire state space  $\Omega = R^{2n}$ . This implies, the equilibrium point  $(q, v) = (a, 0)$  is asymptotically stable.

### 5.1.3. A Reconfigurable Robot with PD-Gravity Control

As an application of a PD-Gravity control, a two-axis reconfigurable robot is considered with RR and RT configurations. In the case of an RR configuration, i.e. both joints are revolute (rotational motion), the manipulator represents a two-axis planar articulated robot shown in Figure (3.1). The dynamic model of this planar robot was developed in Equation (66). Positive definite gain matrices are ensured by using Equation (142) to be diagonal matrices with positive diagonal gain elements:

$$K = \text{diag}\{k_1, k_2, \dots, k_n\} \quad (156)$$

$$L = \text{diag}\{l_1, l_2, \dots, l_n\} \quad (157)$$

The implementation of the PD-Gravity control in Equation (142) is given as follows:

$$e = r - q \quad (158)$$

$$\tau_1 = k_1 e_1 + l_1 \dot{e}_1 + g \left[ \left( \frac{m_1}{2} + m_2 \right) a_1 \cos(\theta_1) + \frac{1}{2} m_2 a_2 \cos(\theta_1 + \theta_2) \right] \quad (159)$$

$$\tau_2 = k_2 e_2 + l_2 \dot{e}_2 + \frac{1}{2} g m_2 a_2 \cos(\theta_1 + \theta_2) \quad (160)$$

In general,  $K$  and  $L$  can be any real symmetric matrices with positive values. By varying the values of the components of  $K$  and  $L$ , the position and speed can be controlled with which the two joints are driven to the desired set point. With some experimentation and control gain tuning, the gains  $k_1 = 80$  and  $l_1 = 50$  are found to work well to track a step function generated from second order differential shown in Figure 5.1 (step function is the blue line and step response is the green line). The motor torque needed to drive the first link is shown in Figure 5.1. The position tracking of the second link of the RR configuration is presented in Figure 5.2, with PD-Gravity gains  $k_2 = 310$  and  $l_2 = 30$ . The motor torque shows that higher torque in Figure 5.2 due to the static coupling with the first link. The tracking performance of the applied PD-Gravity is satisfied as the tracking errors of both revolute joints shown in Figures 5.3 are very low. For the two-axis reconfigurable robot, the RT configuration shown in Figure 3.6

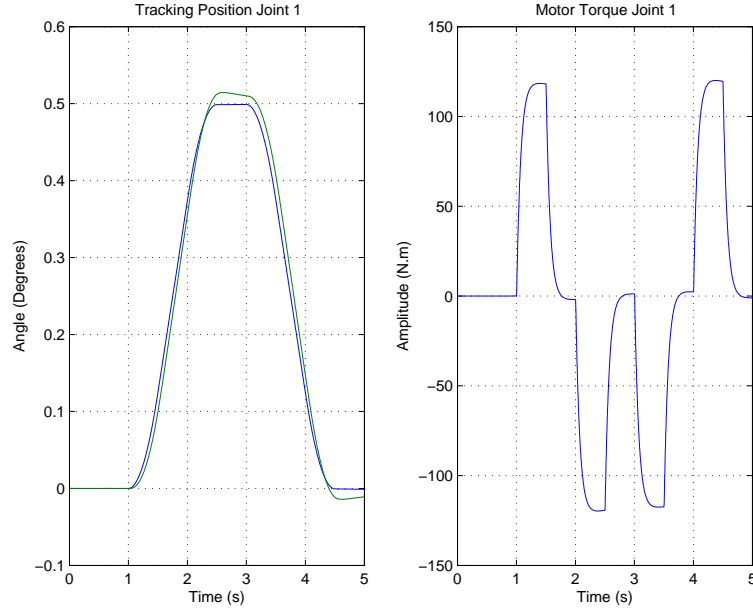


FIGURE 5.1. Performance of PD-gravity controller, RR configuration; tracking step position of joint 1 (left), motor torque (right).

is simulated using the same reference step function. The simulation result of the revolute link has similar tracking properties as the RR configuration. But with the RT configuration, the second joint turns to be prismatic and consequently the second link translates to follow the step function. The dynamic model was given in Equation (67) in which the gravity term of the second link is zero. The calculation of the PD-Gravity control is given as follows:

$$e = r - q \quad (161)$$

$$\tau_1 = k_1 e_1 + l_1 \dot{q}_1 + \left( \frac{a_1 m_1 + a_2 m_2}{4} \right) g \cos(\theta_1) \quad (162)$$

$$F_2 = k_2 e_2 + l_2 \dot{q}_2 \quad (163)$$

Figure 5.4 (left) shows that the reference position is well tracked by the second link using controller gains  $K_2 = 245$  and  $l_2 = 55$ . The control force needed to drive the prismatic link is shown in Figure 5.4 (right).

#### 5.1.4. Three-Axis Robot Structure with PD-Gravity Control

As a second example of a PD-Gravity controller, a three-axis Scara structure robot shown in Figure 3.5 is considered for control application. The selection of the Scara robot because of the robot kinematic structure RRT (two revolute and

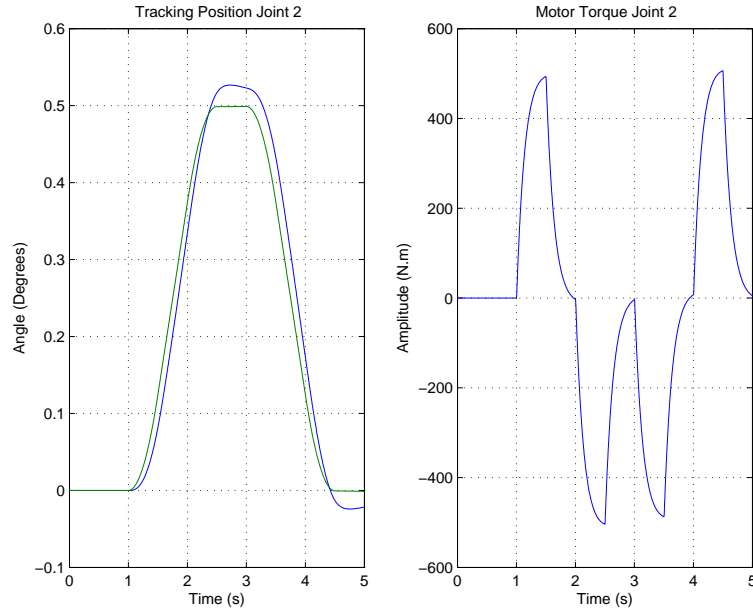


FIGURE 5.2. Performance of PD-gravity controller, RR configuration; tracking step position of joint 2 (left), motor torque (right).

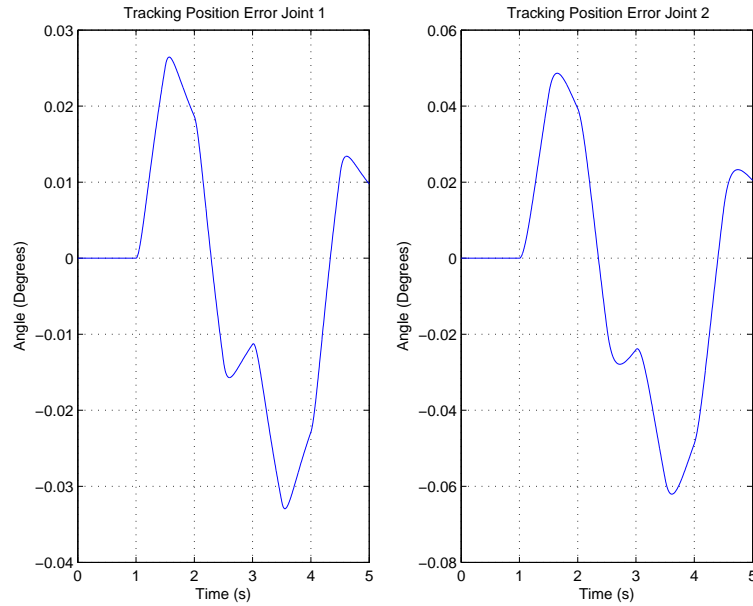


FIGURE 5.3. Performance of PD-gravity controller, RR configuration; tracking step position error joint 1 (left), tracking position error joint 2 (right).

one prismatic). The intent is to provide a thorough analysis in designing a PD-Gravity control to an industrial robot. The diagonal gains  $K$  and  $L$  are selected, as in Equation (156). Because of the kinematic structure and geometry of the Scara robot, there is no gravity load on the first two revolute joints. From the

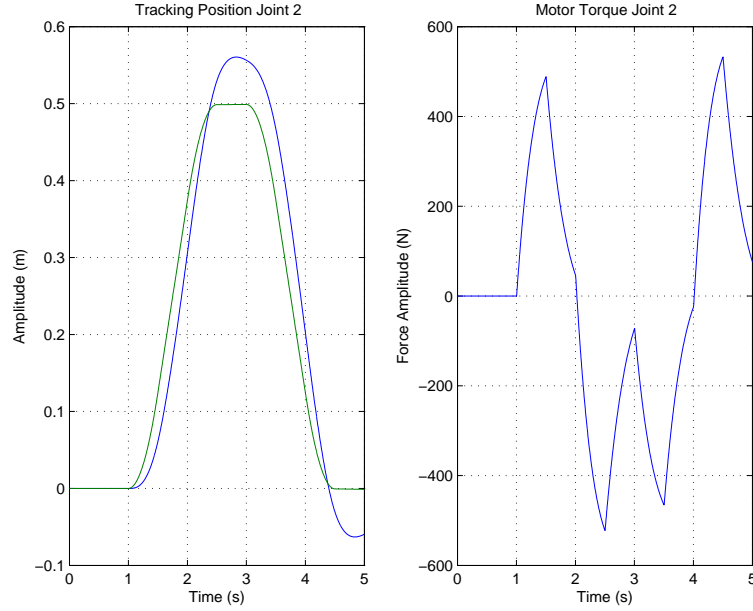


FIGURE 5.4. Performance of PD-gravity controller, RT configuration; tracking position of prismatic joint 2 (left), motor torque (right).

derived dynamic model in Equation (70), the gravity loading on the prismatic joint is  $G(q) = m_3g$ . Thus, the PD-Gravity controller given in Equation (142) is formulated as follows:

$$e = r - q \quad (164)$$

$$\tau_1 = k_1 e_1 + l_1 \dot{e}_1 \quad (165)$$

$$\tau_2 = k_2 e_2 + l_2 \dot{e}_2 \quad (166)$$

$$F_3 = k_3 e_3 + l_3 \dot{e}_3 + m_3 g \quad (167)$$

In the simulation, the joints of the Scara robot follow a numerical path described in degrees. The circular path is well tracked by the first joint of the robot as shown in Figure 5.5 (left). The motor torque needed to drive the first joint is plotted in Figure 5.5 (right). The tracking trajectory and the control torque of the second joint are shown in Figure 5.6. For the first two joints, the tracking error Figure 5.7 to the circular path shows very low values, which indicates suitable controller gain selection and well satisfied performance error requirements.

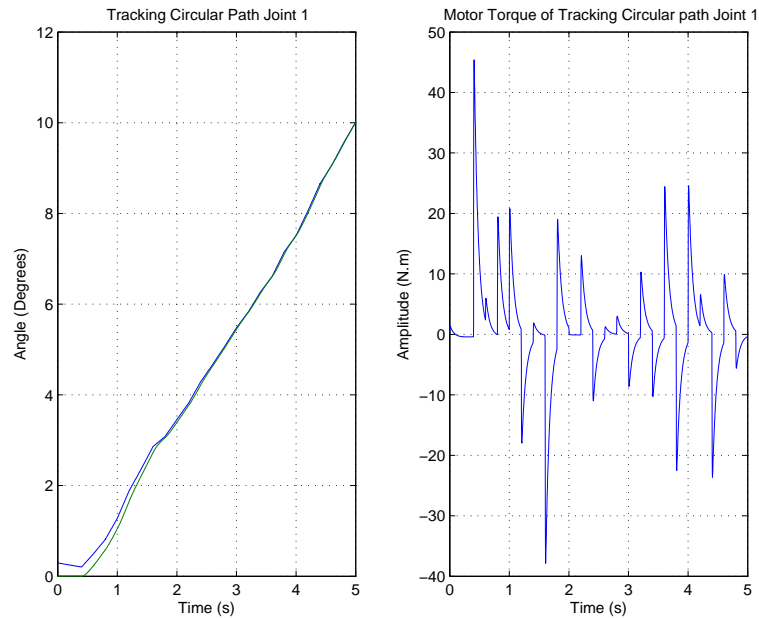


FIGURE 5.5. RRT configuration (Scara); tracking circular path of joint 1 (left), motor torque of joint 1 (right).

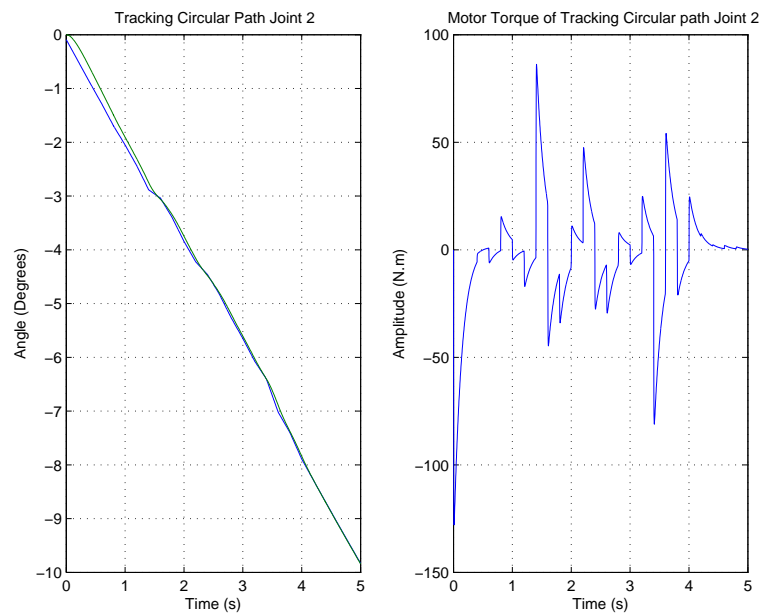


FIGURE 5.6. RRT configuration (Scara); tracking circular path joint 2 (left), motor torque of joint 2 (right).

## Conclusion

With PD-Gravity control, once the effects of the gravity have been eliminated, the nonlinear robotic arm can be controlled using a simple linear PD controller. The only constraints on the closed loop system to be stable is that the controller gains



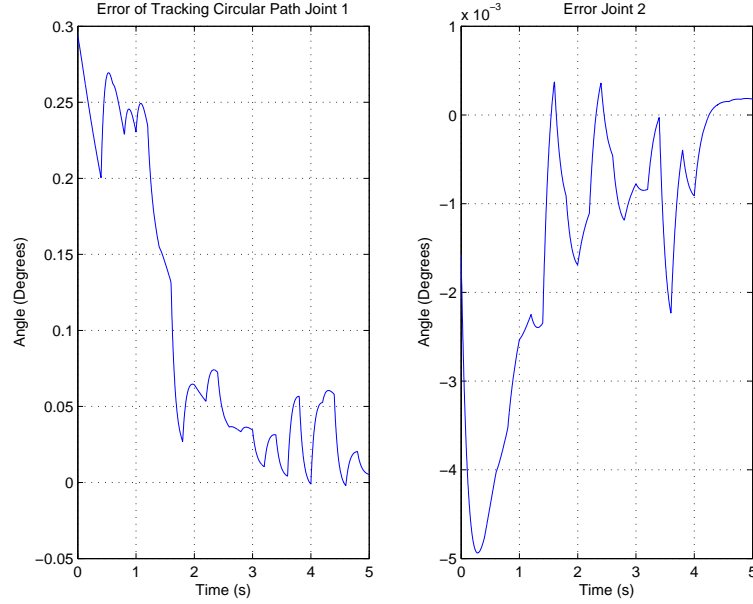


FIGURE 5.7. RRT configuration (Scara); tracking circular path error joint 1 (left), tracking circular path error joint 2 (right).

$K$  and  $L$  be symmetric positive definite matrices. The practical significance of the PD-gravity control Equation (142) lies that it requires no detailed knowledge of the manipulator inertia tensor  $M(q)$ , the Coriolis and centrifugal coupling vector  $c(q, v)$ , or the friction vector  $bv()$  but it does require knowledge of the gravity loading vector  $h(q)$ .

## 5.2. Trajectory Control

### 5.2.1. Feedback Linearization Control

One of the results of differential geometric nonlinear control theory [49, 62] concerns the equivalence of nonlinear systems with linear systems, in case of feedback. The (input-output) feedback linearization utilizes the feedback transforms (invertibility of the system) to render linear input-output dynamics.

**Mathematical Preliminaries** Some mathematical preliminaries will be introduced related to the differential geometry [4]: Suppose that  $h : R^n \rightarrow R^1$  is a smooth scalar function and  $f : R^n \rightarrow R^n, g : R^n \rightarrow R^n$  are vector fields, then:

$$\begin{aligned}\frac{\partial h}{\partial x} &= \left[ \frac{\partial h}{\partial x_1}, \dots, \frac{\partial h}{\partial x_n} \right] \\ L_f^0 h &= h \\ L_f^1 h &= \sum_{i=1}^n \frac{\partial h}{\partial x_i} f_i(x) = \frac{\partial h}{\partial x} f(x) \\ &\vdots \\ L_f^k h(x) &= L_f(L_f^{k-1} h(x)) = \frac{\partial L_f^{(k-1)} h}{\partial x} f(x) \quad \text{for } k = 0, 1, \dots, \\ L_g L_f^k h(x) &= L_g(L_f^k h(x)) = \frac{\partial L_f^k h}{\partial x} g(x)\end{aligned}$$

$\frac{\partial h}{\partial x}$  is called the *gradient* of  $h(x)$  and  $L_f^k h(x)$  is called the *Lie derivative* of  $L_f^{k-1} h(x)$  along the vector field ( $f$ ).

### 5.2.2. Input-Output Linearization

Consider an  $n$ -dimensional single-input single-output nonlinear system of the form:

$$\begin{bmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} f_1(x_1(t), \dots, x_n(t)) \\ \vdots \\ f_n(x_1(t), \dots, x_n(t)) \end{bmatrix} x(t) + \begin{bmatrix} g_1(x_1(t), \dots, x_n(t)) \\ \vdots \\ g_n(x_1(t), \dots, x_n(t)) \end{bmatrix} u(t) \quad (168)$$

where  $u(t)$  is the one-dimensional input,  $y(t)$  is the one-dimensional output,  $f_i : R^n \rightarrow R^1, g_i : R^n \rightarrow R^1, i = 1, \dots, n$ , and  $h : R^n \rightarrow R^1$  are smooth functions with  $f_1(0, \dots, 0) = \dots = f_n(0, \dots, 0) = h(0, \dots, 0) = 0$ . Equation (168) can be written into the following compact form:

$$\begin{aligned}\dot{x} &= f(x(t)) + g(x(t))u(t) \\ y &= h(x(t))\end{aligned} \quad (169)$$

where the  $f(x), g(x)$  and  $h(x)$  are  $n$ -dimensional vector-valued smooth functions defined on  $R^n$  called *vector fields* in  $R^n$ . The goal is now to find an integer  $\rho$  and a state feedback control law:

$$u = \alpha(x) + \beta(x)v \quad (170)$$

where  $\alpha(\cdot)$  and  $\beta(\cdot)$  are smooth functions defined in a neighborhood of some point (equilibrium point)  $x_0 \in R^n$  and  $\beta(x_0)$  such that the resulting system has the property that the  $\rho$ th-order derivative of the output is given by:

$$y^{(\rho)}(t) = v(t), \quad t \in \Gamma \quad (171)$$

where  $\Gamma$  is an open interval containing  $t = 0$ . The vector fields  $\alpha(x)$  and  $\beta(x)$  can be found by differentiating the output  $y$  until the input  $u$  appears explicitly. Differentiating  $y$  results in:

$$\begin{aligned} y^1(t) &= L_f h(x(t)) \\ &\dots \\ y^{(\rho-1)}(t) &= L_f^{(\rho-1)} h(x(t)) \\ y^{(\rho)}(t) &= L_f^\rho h(x(t)) + L_g L_f^{(\rho-1)} h(x(t)) u(t) \\ y^{(\rho)}(t) &= \alpha(x) + \beta(x) u(t) \end{aligned} \quad (172)$$

with

$$L_g L_f^{(\rho-1)} h(x_0) \neq 0$$

Then the control law (170) yields:

$$u = \frac{-L_f^\rho h(x(t)) + v(t)}{L_g L_f^{\rho-1} h(x(t))} \quad (173)$$

The new system (171) may not be fully observable, if  $\rho < n$ . Hence, it is important that the zero-dynamics of:

$$\begin{aligned} \dot{x} &= f(x(t)) + g(x(t))\alpha(x(t)) + g(x(t))\beta(x(t))v(t) \\ y &= h(x(t)) \end{aligned} \quad (174)$$

is stable. The zero dynamics is defined to be the part of the system dynamics that complies to  $y \equiv 0$ , which can be true for some nontrivial pairs of  $x(t), u(t)$ . If the zero dynamics are asymptotically stable, the system is called an *asymptotically*

*minimum phase system.*

### Asymptotic Tracking

The resulting system (174) has a special structure, where the input  $v$  directly governs the motion of the output. Therefore, the design of an inversion model controller for various objectives such as asymptotic tracking problem is addressed to design a control law such that the output  $y(t)$  asymptotically tracks a given trajectory  $r(t)$ . The desired tracking can be achieved by requiring that the error  $e(t) = y(t) - r(t)$  satisfies:

$$e^{(\rho)}(t) + \alpha_{(\rho-1)}e^{(\rho-1)} + \dots + \alpha_1e^{(1)}(t) + \alpha_0e(t) = 0 \quad (175)$$

where  $\alpha_0, \dots, \alpha_{\rho-1}$  are such that:

$$\lambda^\rho + \alpha_{(\rho-1)}\lambda^{(\rho-1)} + \dots + \alpha_1\lambda + \alpha_0$$

is a Hurwitz polynomial, then:

$$\begin{aligned} e(t) &= y(t) - r(t) \\ e^{(1)}(t) &= y^{(1)}(t) - r^{(1)}(t) \\ &\dots \\ e^\rho &= y^\rho - r^\rho(t) \\ e^\rho &= v(t) - r^\rho(t) \end{aligned} \quad (176)$$

solving for  $v(t)$  yields:

$$v(t) = r^{(\rho)}(t) - (\alpha_{(\rho-1)}e^{(\rho-1)} + \dots + \alpha_1e^{(1)}(t) - \alpha_0e(t)) \quad (177)$$

The composition of Equation (173) and (177) leads to form the following control law:

$$u(t) = \frac{-L_f^\rho h(x(t)) + r^{(\rho)}(t) - \sum_{k=0}^{\rho-1} \alpha_k e^{(k)}(t)}{L_g L_f^{\rho-1} h(x(t))} \quad (178)$$

which achieves asymptotic tracking for the system (169).

### 5.2.3. Sliding Mode Control (SMC)

One approach to robust control design is called sliding mode control (SMC) methodology, which is also a type of variable structure control system (VSCS) [76]. The most significant feature of SMC is the complete insensitivity to parametric uncertainty and external disturbances during the sliding mode. The VSCS uses a high speed switching control law to achieve two objectives. Firstly, it drives the nonlinear system's state trajectory along a specified and user chosen surface in the state space which is called the sliding or switching surface. This surface is named the switching surface because a control path has one gain if the state trajectory of the system is above the surface and a different gain if the trajectory drops below the surface. Secondly, it maintains the system's state trajectory on this surface for all subsequent times. During the process, the control system's structure varies from one to another and therefore it grants the name variable structure control. The feedback linearization control law (178) achieves asymptotic tracking by satisfying the error equation (175). Next, a *sliding surface* in  $R^n$  is defined as follows:

$$s(t) = e^{(\rho-1)}(t) + \alpha_{(\rho-1)}e^{(\rho-2)} + \cdots + \alpha_1 e(t) + \alpha_0 \int e(t)dt = 0 \quad (179)$$

where  $\alpha_{\rho-1}, \dots, \alpha_0$  are such that:

$$\lambda^\rho + \alpha_{(\rho-1)}\lambda^{(\rho-1)} + \cdots + \alpha_1\lambda + \alpha_0$$

is a Hurwitz polynomial. The derivative of the sliding surface  $s(t)$  is given by:

$$\dot{s}(t) = e^{(\rho)}(t) + \alpha_{(\rho-1)}e^{(\rho-1)}(t) + \cdots + \alpha_1 e^{(1)}(t) + \alpha_0 e(t) \quad (180)$$

To satisfy the asymptotic tracking objective, the requirement is:

$$\dot{s}(t) = 0 \quad (181)$$

Now, instead of satisfying the tracking error equation (175), a sliding condition is described as follows:

**Sliding Condition.** There exists a positive number  $\mu$  such that

$$\frac{1}{2} \frac{ds^2}{dt} \leq -\mu|s| \quad (182)$$

The sliding condition can be written in the following equivalent form:

$$\begin{aligned} \dot{s} &\leq -\mu, & s(t) &> 0 \\ \dot{s} &\geq \mu, & s(t) &< 0 \end{aligned} \quad (183)$$

Using the conditions of Equation (183), the surface becomes locally attractive in the sense that trajectories nearby the surface will be attracted to the surface defined by  $s(t) = 0$  in a finite time. Furthermore, since  $\dot{s}(t)$  is a stable differential equation in  $e$ , satisfaction of  $\dot{s}(t_0) = 0$  for some  $t_0$  implies:

$$\lim_{t \rightarrow \infty} e(t) = 0 \quad (184)$$

which leads to the asymptotic tracking. It is also possible to solve for  $u$ , given an exogenous input  $v$  such that:

$$\dot{s} = v \quad (185)$$

Equation (180) can be rewritten as:

$$\begin{aligned} \dot{s}(t) &= (y^{(\rho)}(t) - r^{(\rho)}(t)) + \alpha_{(\rho-1)}e^{(\rho-1)}(t) + \cdots + \alpha_{(1)}e^{(1)}(t) + \alpha_0e(t) \\ &= a(x) + b(x)u - r^{(\rho)}(t) + \alpha_{(\rho-1)}e^{(\rho-1)}(t) + \cdots + \alpha_{(1)}e^{(1)}(t) + \alpha_0e(t) \end{aligned} \quad (186)$$

Therefore, the control law  $u$  is calculated as:

$$u = \frac{v - a(x) + r^{(\rho)}(t) + \alpha_{(\rho-1)}e^{(\rho-1)}(t) + \cdots + \alpha_{(1)}e^{(1)}(t) + \alpha_0e(t)}{b(x)} \quad (187)$$

By letting:

$$v = -k \operatorname{sgn}(s) \quad (188)$$

where  $k > \mu$  and the sign function is:

$$\begin{aligned} \operatorname{sgn}(s) &= 1, & s &> 0 \\ \operatorname{sgn}(s) &= -1, & s &< 0 \end{aligned} \quad (189)$$

Substituting Equation (188) into Equation (187) gives the overall control law as follows:

$$u = \frac{-k \operatorname{sgn}(s) - a(x) + r^{(\rho)}(t) + \alpha_{(\rho-1)}e^{(\rho-1)}(t) + \cdots + \alpha_{(1)}e^{(1)}(t) + \alpha_0e(t)}{b(x)} \quad (190)$$

From Equation (190), the tracking control law (178) can be viewed as a special case of the sliding mode control by letting  $k = 0$  in Equation (190). The discontinuity of the sign function will cause *chattering* in the closed loop system. In practice, the sign function is often replaced by a saturation function  $\text{sat}(s/\epsilon)$ , where  $\text{sat}(\cdot)$  is defined as follows:

$$\begin{aligned} \text{sat}(x) &= x, & \text{if } |x| \leq 1 \\ \text{sat}(x) &= \text{sgn}(x), & \text{if } |x| > 1 \end{aligned} \quad (191)$$

Using this replacement will introduce tracking error. Trade-offs between the tracking error and control bandwidth can be made by suitably selecting the boundary layer.

#### 5.2.3.1. Linearity in Parameters

The dynamic parameters of a reconfigurable robot are not constant and function of the robot configuration. The robot dynamics (71) can be written in the form [80]:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = Y(q, \dot{q}, \ddot{q})\varphi \quad (192)$$

where  $Y(q, \dot{q}, \ddot{q})$  is an  $n \times r$  matrix of known time functions and  $\varphi$  is an  $r \times 1$  vector of unknown constant parameters. This property is formulated by Graig [46] in that it shows the separation of unknown parameters and known time functions. The reason that the robot dynamics can be separated in this form is that the robot dynamics are linear in the parameters expressed in the vector form  $\varphi$ . This separation of unknown parameters and known time functions will be used in the formulation of the adaptive update rule.

#### 5.2.4. Sliding Mode Control Based on Estimated Model

In this subsection, sliding mode controllers are derived based on estimated models and a 3-DOF reconfigurable robot is simulated to track a trigonometric reference signal. For the desired trajectory as  $q_d(t)$ , the tracking error is defined as follows:

$$e = q_d - q$$

Define the reference velocity [76]:

$$\dot{q}_r = \dot{q}_d + \Lambda(q_d - q) \quad (193)$$

Where  $\Lambda$  is a positive diagonal matrix. Define a parameter vector error to be  $\tilde{\varphi} = \hat{\varphi} - \varphi$ , where  $\hat{\varphi}$  is the estimated vector of  $\varphi$ . According to the parametric linear property, the robotic dynamic equation is now formulated as:

$$M(q)\ddot{q}_r + C(q, \dot{q})\dot{q}_r + G(q) = Y(q, \dot{q}, \ddot{q})\varphi \quad (194)$$

And the dynamic error equation is:

$$\tilde{M}(q)\ddot{q}_r + \tilde{C}(q, \dot{q})\dot{q}_r + \tilde{G}(q) = Y(q, \dot{q}, \ddot{q})\tilde{\varphi} \quad (195)$$

Where  $\tilde{M}(q) = M(q) - \hat{M}(q)$ ,  $\tilde{C}(q) = C(q) - \hat{C}(q)$  and  $\tilde{G}(q) = G(q) - \hat{G}(q)$ .

Define the sliding surface  $s$  as:

$$s = \dot{e} + \Lambda e \quad (196)$$

Select the Lyapunov function as:

$$V(t) = \frac{1}{2} s^T M(q) s \quad (197)$$

The Lyapunov function derivative is:

$$\dot{V}(t) = \frac{1}{2} \left[ s^T M(q) \dot{s} + s^T \dot{M}(q) s + \dot{s}^T M(q) s \right] \quad (198)$$

Using the symmetric property of matrices:

$$\dot{V}(t) = \frac{1}{2} \left[ 2s^T M(q) \dot{s} + s^T \dot{M}(q) s \right] \quad (199)$$

$$\dot{V}(t) = s^T M(q) \dot{s} + \frac{1}{2} s^T \dot{M}(q) s \quad (200)$$

Since  $s = \dot{q} - \dot{q}_r$  and  $\dot{s} = \ddot{q} - \ddot{q}_r$ , then:

$$\dot{V}(t) = s^T (M(q)\ddot{q} - M(q)\ddot{q}_r) + \frac{1}{2} s^T \dot{M}(q) s \quad (201)$$



From the equation of motion:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (202)$$

The following yields:

$$M(q)\ddot{q} = \tau - C(q, \dot{q})\dot{q} - G(q) \quad (203)$$

$$\dot{V}(t) = s^T [\tau - C(q, \dot{q})\dot{q} - G(q) - M(q)\ddot{q}_r] + \frac{1}{2}s^T \dot{M}(q)s \quad (204)$$

$$\dot{V}(t) = s^T [\tau - M(q)\ddot{q}_r - C(q, \dot{q})(s + \dot{q}_r) - G(q)] + \frac{1}{2}s^T \dot{M}(q)s \quad (205)$$

$$\dot{V}(t) = s^T [\tau - M(q)\ddot{q}_r - C(q, \dot{q})\dot{q}_r - G(q)] - s^T C(q, \dot{q})s + \frac{1}{2}s^T \dot{M}(q)s \quad (206)$$

Using the skew-symmetric property:

$$N(q, \dot{q}) = \frac{1}{2}s^T \left( \dot{M}(q) - 2C(q, \dot{q}) \right) s = 0 \quad (207)$$

The following yields:

$$\begin{aligned} \dot{V}(t) &= s^T [M(q)(\ddot{q}_r - \ddot{q}) + C(q, \dot{q})(\dot{q}_r - \dot{q})] \\ \dot{V}(t) &= s^T [M(q)\ddot{q}_r + C(q, \dot{q})\dot{q}_r + G(q) - \tau] \end{aligned} \quad (208)$$

The Lyapunov derivative is negative by designing the controller as:

$$\tau = \widehat{M}(q)\ddot{q}_r + \widehat{C}(q, \dot{q})\dot{q}_r + \widehat{G}(q)q + \tau_s \quad (209)$$

Where  $\tau_s$  is the robustness element to be designed to compensate for the parametric uncertainty. From equations (208) and (209), the following yields:

$$\begin{aligned} \dot{V}(t) &= s^T [M(q)\ddot{q}_r + C(q, \dot{q})\dot{q}_r + G(q) - \widehat{M}(q)\ddot{q}_r - \widehat{C}(q, \dot{q})\dot{q}_r - \widehat{G}(q)q - \tau_s] \\ \dot{V}(t) &= s^T [\tilde{M}(q)\ddot{q}_r + \tilde{C}(q, \dot{q})\dot{q}_r + \tilde{G}(q) - \tau_s] \\ \dot{V}(t) &= s^T [Y(q, \dot{q}, \ddot{q})\tilde{\varphi} - \tau_s] \end{aligned} \quad (210)$$

The robustness factor can be selected as:

$$\tau_s = k_i \operatorname{sgn}(s_i) + s_i = \begin{bmatrix} k_1 \operatorname{sgn}(s_1) + s_1 \\ \dots \\ k_i \operatorname{sgn}(s_i) + s_i \end{bmatrix} \quad (211)$$

or select a saturation function as:

$$\tau_{sat} = k_i \operatorname{sat}(s_i) + s_i = \begin{bmatrix} k_1 \operatorname{sat}(s_1) + s_1 \\ \dots \\ k_i \operatorname{sat}(s_i) + s_i \end{bmatrix} \quad (212)$$

### 5.2.5. Simulation of 3-DOF Reconfigurable Manipulator

A 3-DOF reconfigurable robot is considered with reconfigurable D–H parameters as given in tables 3.2 and 3.1. The parameter linearized matrix  $Y(q, \dot{q}, \ddot{q})$  and the unknown parameter vector  $\varphi$  are calculated and shown in Appendix D. The matrix and vector are reformulated as follows:

$$\tilde{\varphi} = \left[ \tilde{\varphi}_1 \quad \tilde{\varphi}_2 \quad \tilde{\varphi}_3 \quad \tilde{\varphi}_4 \quad \tilde{\varphi}_5 \right]^T, \quad |\tilde{\varphi}_i| \leq \bar{\varphi}_i, \quad i = 1, 2, 3, 4, 5$$

where  $\bar{\varphi}_i$  is the upper bound limit of the error parameter  $\tilde{\varphi}_i$ .

$$Y(q, \dot{q}, \ddot{q}) = [Y_{ij}], \quad |Y_{ij}| \leq \bar{Y}_{ij}, \quad i = 1, 2, \quad j = 1, 2, 3, 4, 5 \quad (213)$$

where  $Y_{ij}$  is a matrix element with dimension  $i \times j$ , then:

$$k_i = \sum_{j=1}^5 \bar{Y}_{ij} \bar{\varphi}_j, \quad i = 1, 2 \quad (214)$$

The parameters given in Equations (213) and (214) are substituted in Lyapunov derivative Equation (210) to proof that Lyapunov derivative is negative:

$$\begin{aligned} \dot{V}(t) &= \sum_{i=1}^2 \sum_{j=1}^5 s_i Y_{ij} \tilde{\varphi}_j - \sum_{i=1}^2 s_i k_i \operatorname{sgn}(s_i) - \sum_{i=1}^2 s_i^2 \\ &= \sum_{i=1}^2 \sum_{j=1}^5 s_i Y_{ij} \tilde{\varphi}_j - \sum_{i=1}^2 |s_i| \bar{Y}_{ij} \bar{\varphi}_j - \sum_{i=1}^2 s_i^2 \end{aligned}$$

$$\leq -\sum_{i=1}^2 s_i^2 \leq 0$$

This proves the negative definiteness of the Lyapunov function's derivative. The reconfigurable robot is simulated with desired trajectories  $q_{d1} = \sin(2\pi t)$ ,  $q_{d2} = \sin(2\pi t)$  and  $q_{d3} = \sin(2\pi t)$ . The tracking error parameter  $\Lambda$  of the sliding surface is selected  $\Lambda = \text{diag}(5)$ . Using the controller Equation (209), the bounded value of the error parameter between the estimated and constant parameter is chosen:  $\tilde{\varphi}_i = |\tilde{\varphi}_i| + 0.05$ . For both joints, the robust element values of the switching control (211) are chosen:  $k_i = 2$ . The sliding mode control with switching robust control is shown in Figure 5.8. To eliminate the chattering that the controller may cause, a saturation control with boundary layer thickness is  $\Phi_i = 0.05$  is implemented. Figure 5.9 shows that the sliding mode control that refined within the selected boundary layers. Figures 5.10 shows the tracking position to a trigonometric reference signal for both joints. It initially exhibits small tracking errors at the start of motion, but then converges to follow the trajectory with zero errors. The joint velocity response shown in Figure 5.11 has similar behavior as the position and for the both joints. The trajectory tracking errors shown in Figure 5.12 are due to the initial conditions and the nonlinear coupling of links and joints influencing the joint motion.

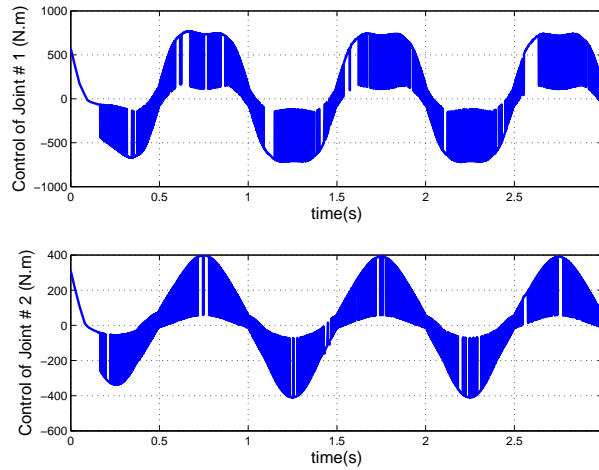


FIGURE 5.8. Motor torques of joints 1 and 2 using control law (211); the chattering is due to the sign function.

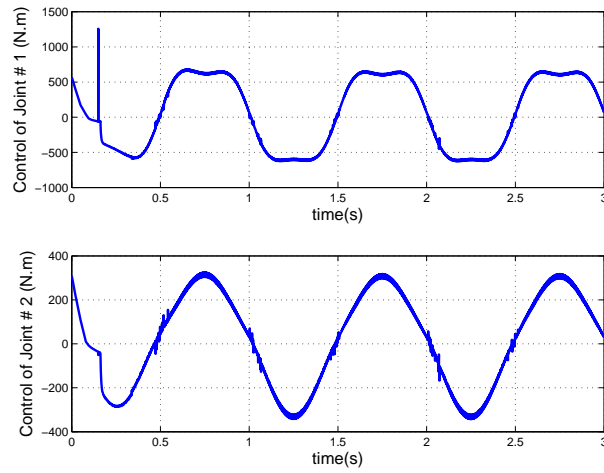


FIGURE 5.9. Motor torques of joints 1 and 2 when replacing the function  $sgn(s)$  with  $sat(s/\Phi)$ .

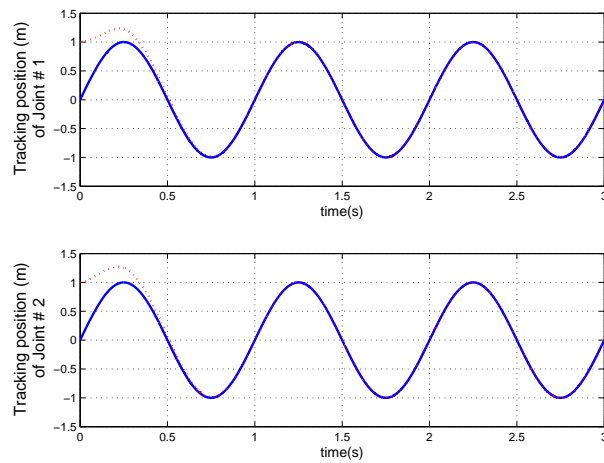


FIGURE 5.10. Tracking positions of joints 1 and 2 using the control law (209.) Reference position (solid), actual position (dotted).

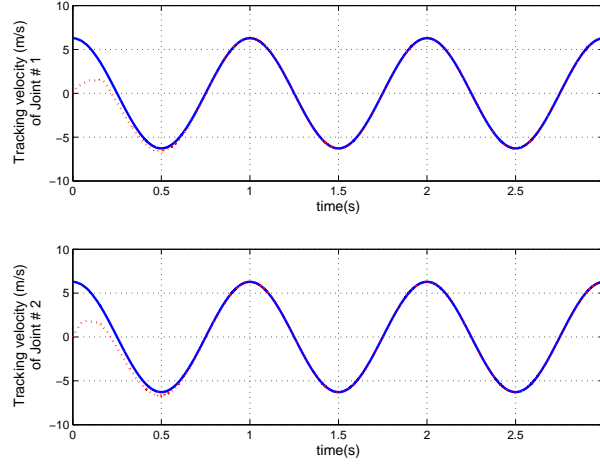


FIGURE 5.11. Tracking velocities of joints 1 and 2 using the control law (209.) Reference velocity (solid), actual position (dotted).

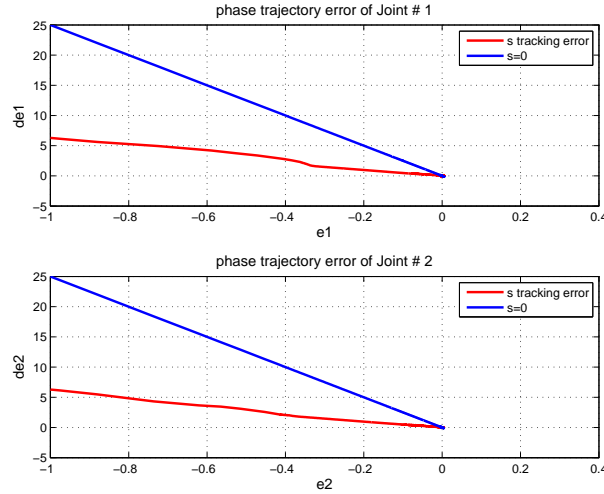


FIGURE 5.12. The phase portrait of the trajectory errors of joints 1 and 2.

### 5.2.6. Sliding Mode Control Based on Bounded Model

In this subsection, sliding mode control is derived based on bounded model parameter. The Lyapunov derivative function Equation (208) can be rewritten as:

$$\dot{V}(t) = -s^T[\tau - (M(q)\ddot{q}_r + C(q, \dot{q})\dot{q}_r + G(q))] \quad (215)$$

$$\dot{V}(t) = -s^T[\tau - Y(q, \dot{q}, \ddot{q})\varphi] \quad (216)$$

$$\tau = \bar{k}_i \operatorname{sat}(s_i) + s_i = \begin{bmatrix} \bar{k}_1 \operatorname{sat}(s_1) + s_1 \\ \dots \\ \bar{k}_i \operatorname{sat}(s_i) + s_i \end{bmatrix} \quad (217)$$

The 3-DOF reconfigurable robot is simulated with the following bounded  $\bar{k}_i$ :

$$\bar{k}_i = \sum_{j=1}^5 \bar{Y}_{ij} \bar{\varphi}_{ij}, \quad i = 1, 2 \quad (218)$$

Using the Equations (217) and (218), Lyapunov derivative function becomes:

$$\begin{aligned} \dot{V}(t) &= - \left[ \sum_{i=1}^2 s_i \bar{k}_i \operatorname{sgn}(s_i) + \sum_{i=1}^2 s_i^2 - \sum_{i=1}^2 \sum_{j=1}^5 s_i Y_{ij} \varphi_j \right] \\ &= - \left[ \sum_{i=1}^2 \sum_{j=1}^5 |s_i| \bar{Y}_{ij} \bar{\varphi}_j + \sum_{i=1}^2 s_i^2 - \sum_{i=1}^2 \sum_{j=1}^5 s_i Y_{ij} \varphi_j \right] \\ &\leq - \sum_{i=1}^2 s_i^2 \leq 0 \end{aligned}$$

This proves the negative definiteness of the Lyapunov function's derivative. The resulting simulations, which have similar behavior as the estimated control results.

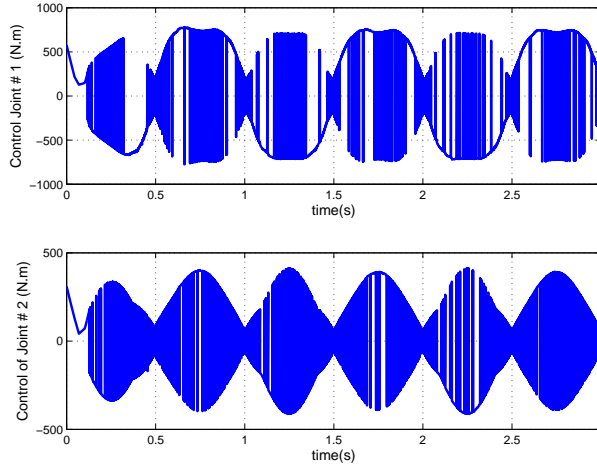


FIGURE 5.13. Motor torques of joints 1 and 2 using control law (217); the chattering is due to the sign function.

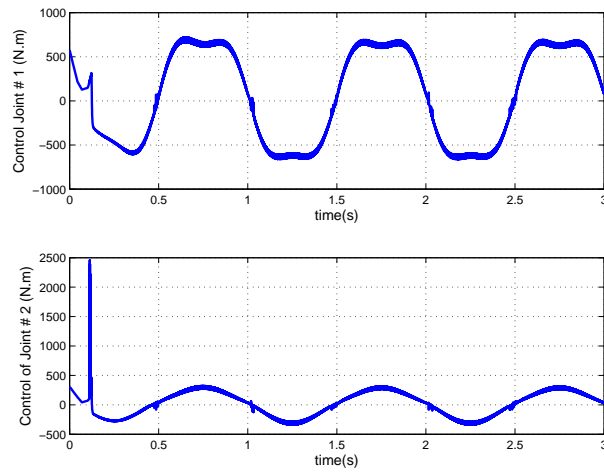


FIGURE 5.14. Motor torques of joints 1 and 2 when replacing the function  $sgn(s)$  with  $sat(s/\Phi)$ .

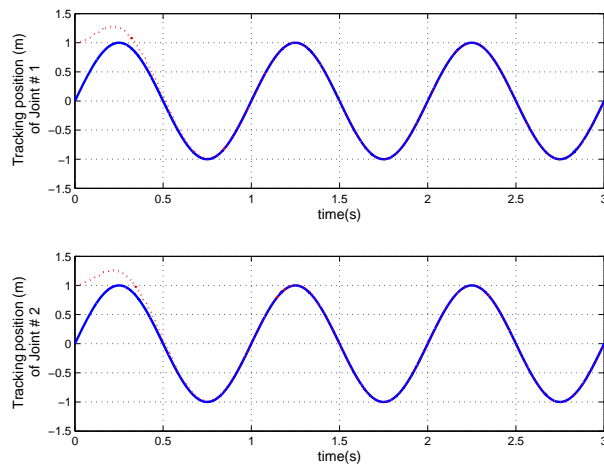


FIGURE 5.15. Tracking positions of joints 1 and 2 using control law (209). Reference position (solid), actual positions (dotted).

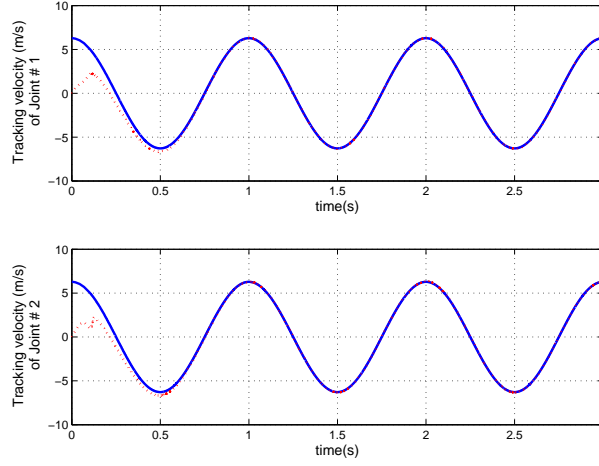


FIGURE 5.16. Tracking velocities of joints 1 and 2 using control law (209). Reference velocities (solid), actual positions (dotted).

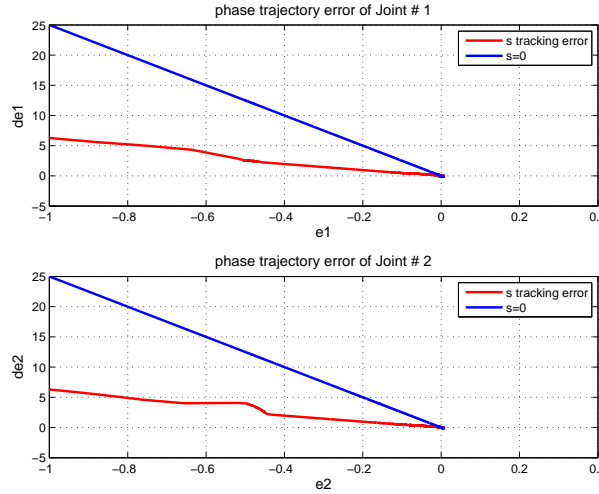


FIGURE 5.17. The phase portrait of the trajectory errors of joints 1 and 2.

### 5.2.7. Sliding Mode Control Based on Computed Torque Method

In this subsection, sliding mode controllers are derived based on the computed torque method. This control can be designed for the general equation of motion (71), as follows:

$$\widehat{M}(q)\nu + \widehat{C}(q, \dot{q})\dot{q} + \widehat{G}(q)q = \tau \quad (219)$$

where  $\nu$  is the auxiliary control input, and  $\widehat{M}(q)$ ,  $\widehat{C}(q, \dot{q})$ ,  $\widehat{G}(q)$  are the respective estimations of  $M(q)$ ,  $C(q, \dot{q})$  and  $G(q)$  by using inertial parameter  $\widehat{\varphi}$  of the manipulator, where  $\widetilde{\varphi} = \varphi - \widehat{\varphi}$ . Substituting Equation (219) into Equation (26)



results:

$$\begin{aligned}
M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q)q &= \widehat{M}(q)\nu + \widehat{C}(q, \dot{q})\dot{q} + \widehat{G}(q) \\
\widehat{M}(q)\ddot{q} &= \widehat{M}(q)\nu - [\widetilde{M}(q)\ddot{q} + \widetilde{C}(q, \dot{q})\dot{q} + \widetilde{G}(q)] \\
\widehat{M}(q)\ddot{q} &= \widehat{M}(q)\nu - Y(q, \dot{q}, \ddot{q})\tilde{\varphi}
\end{aligned} \tag{220}$$

If the vector parameter estimation  $\hat{\varphi}$  makes  $\widehat{M}(q)$  invertible, then the above equation can be rewritten as:

$$\ddot{q} = \nu - (\widehat{M}(q))^{-1}Y(q, \dot{q}, \ddot{q})\tilde{\varphi} = \nu - \tilde{r} \tag{221}$$

The sliding variable is selected as:

$$s = \dot{e} + \Lambda e \tag{222}$$

where  $e = q_d - q$ ,  $\dot{e} = \dot{q}_d - \dot{q}$ , for a set of sliding surfaces  $s = [s_1 s_2 \dots s_n]^T$  and  $\Lambda$  is a positive diagonal matrix. The derivation of the sliding surface results:

$$\dot{s} = \ddot{e} + \Lambda\dot{e} = (\ddot{q}_d - \ddot{q}) + \Lambda\dot{e} = \ddot{q}_d - \nu + \tilde{r} + \Lambda\dot{e} \tag{223}$$

Then, the control structure is proposed as follows:

$$\nu = \ddot{q}_d + \Lambda\dot{e} + r \tag{224}$$

where  $r = (\tilde{r} + \eta) \operatorname{sgn}(s)$ ,  $\|\tilde{r}\| \leq \tilde{r}$ ,  $\eta > 0$

This yields:

$$\dot{s} = \tilde{r} - r \tag{225}$$

The Lyapunov function is selected as:

$$V = \frac{1}{2}s^T s$$

The derivative of the Lyapunov function is given as follows:

$$\begin{aligned}
\dot{V} = s^T \dot{s} &= s^T(\tilde{r} - r) \\
&= s^T \tilde{r} - \tilde{r}^T s \operatorname{sgn}(s) - \eta s^T \operatorname{sgn}(s) \\
&\leq -\eta \|s\| \leq 0
\end{aligned}$$

The simulation results show, the more precise the estimation of  $\hat{\varphi}$  is, the smaller upper bounded limit of the parameter error and then smaller chattering behavior. The tracking position of the first two joints to trigonometric reference signals using the controllers (219) and (224) are shown in Figure 5.18. Using saturation function instead of sign function, the control of the first two joints are shown in Figure 5.19. The trajectory tracking error Figure 5.20 shows that the robot end effector perfectly tracks the desired trajectory and the origin equilibrium point is asymptotically stable.

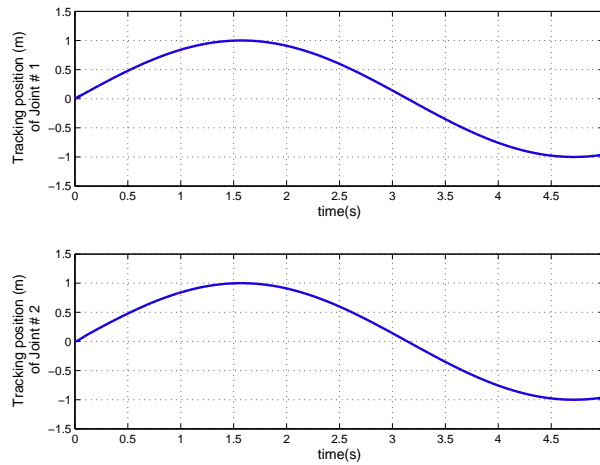


FIGURE 5.18. Tracking positions of joints 1 and 2 using the control law (219). Reference position (solid), actual positions (dotted).

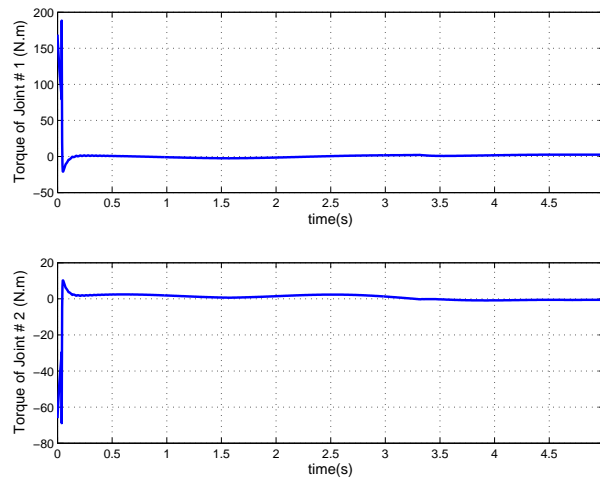


FIGURE 5.19. Motor torques of joints 1 and 2 using the control law (224), replacing the function  $sgn(s)$  with  $sat(s/\Phi)$ .

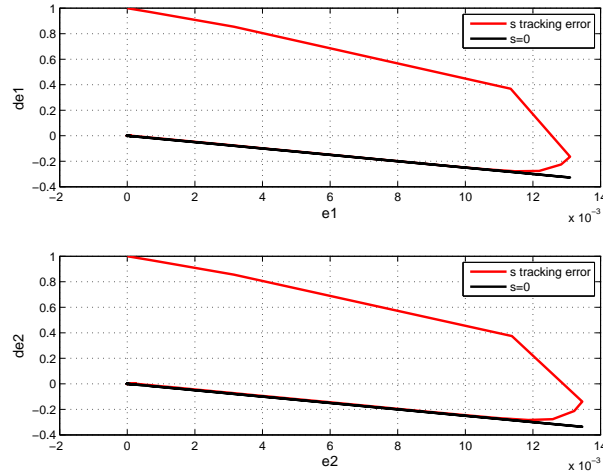


FIGURE 5.20. Trajectory errors of joints 1 and 2.

### 5.2.8. Adaptive Control.

Adaptive control is an approach to control systems, which have constant or slowly-varying uncertain parameters [76]. The basic theory in adaptive control is to estimate the uncertain system parameters on-line based on measured system signals, and exploit the estimated parameters in the control input computation. Thus, an adaptive control can be regarded as a control with on-line parameters estimations which maintain consistent performance of a system in the presence of uncertainty or unknown variation in system parameters. The resulting dynamic parameters of a reconfigurable manipulator are uncertain and time varying due to the configuration change and joint pose dependency. This leads to consider the adaptive control approach as a way of automatically adjusting the controller parameters in the face of changing robot dynamic parameters. An adaptive control system is depicted schematically in Figure 5.21. It is composed of three parts: a reconfigurable robot with unknown parameters, a feedback control law containing adjustable parameters, and an adaptation mechanism for updating the adjustable parameters. The operation of the adaptive controller is as follows: at each time instant, the estimator sends to the controller a set of estimated system parameters  $\hat{\varphi}$ , which was calculated based on the system input  $\tau$  and output  $q, \dot{q}$ . The controller finds its corresponding parameters and then computes a control input  $\tau$  based on the controller parameters and measured signals. The control input  $\tau$  causes a new

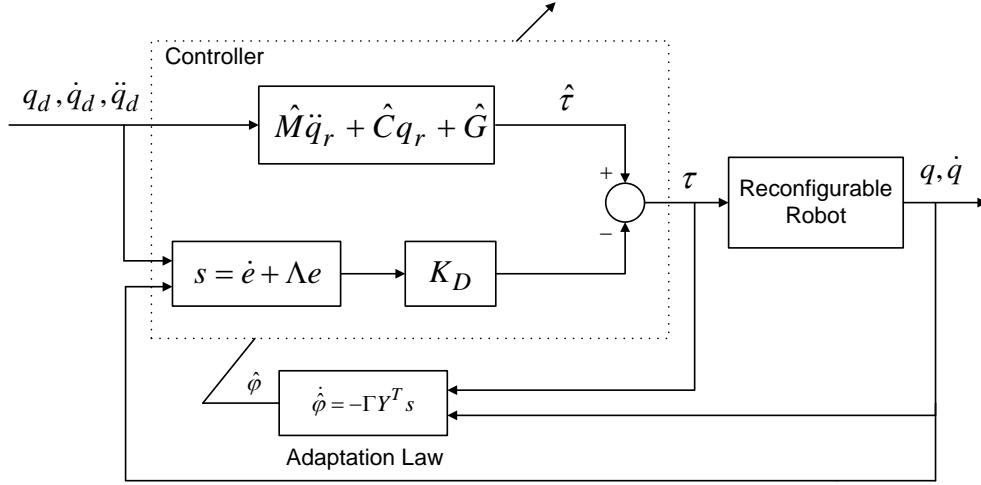


FIGURE 5.21. Control structure diagram of the adaptive control.

system output to be generated and whole cycle of parameter estimation and input updates is repeated.

### 5.2.9. Derivation of Adaptive Sliding Mode Control

An adaptive controller is derived and developed for a 3-DOF reconfigurable robot described in tables 3.2 and 3.1. The trajectory control problem is solved by developing a robust controller for the actuator joints, and an estimation law for the unknown parameters, such that the manipulator output  $q(t)$  closely tracks the desired trajectory  $q_d(t)$ . The tracking error is defined as in Equation (196). Define the reference velocity [76]:

$$\dot{q}_r = \dot{q}_d + \Lambda(q_d - q) \quad (226)$$

Where  $\Lambda$  is a positive diagonal matrix:

$$\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n), \quad \lambda_i > 0$$

The sliding variable is defined as:

$$s = \dot{e} + \Lambda e$$

The energy conservation of the system Equation (71) can be formalized by the Lyapunov function as follows:

$$V(t) = \frac{1}{2}s^T M s + \frac{1}{2}\tilde{\varphi}^T \Gamma \tilde{\varphi} \quad (227)$$

Where the estimation error is:  $\tilde{\varphi} = \hat{\varphi} - \varphi$ , and

$$\Gamma = \text{diag}(\gamma_1, \gamma_2, \dots, \gamma_n), \quad \gamma_i > 0$$

The Lyapunov function derivative is:

$$\dot{V}(t) = s^T M \dot{s} + \frac{1}{2}s^T \dot{M} s + \tilde{\varphi}^T \Gamma \dot{\tilde{\varphi}}$$

$$\dot{V}(t) = s^T (M \ddot{q} - M \ddot{q}_r) + \frac{1}{2}s^T \dot{M} s + \tilde{\varphi}^T \Gamma \dot{\tilde{\varphi}}$$

$$\dot{V}(t) = s^T (\tau - C \dot{q} - G - M \ddot{q}_r) + \frac{1}{2}s^T \dot{M} s + \tilde{\varphi}^T \Gamma \dot{\tilde{\varphi}}$$

$$\dot{V}(t) = s^T (\tau - C(s + \dot{q}_r) - G - M \ddot{q}_r) + \frac{1}{2}s^T \dot{M} s + \tilde{\varphi}^T \Gamma \dot{\tilde{\varphi}}$$

The proposed controller is selected to be:

$$\tau = \hat{\tau} - K_D s = \widehat{M}(q) \ddot{q}_r + \widehat{C}(q, \dot{q}) \dot{q}_r + \widehat{G} - K_D s \quad (228)$$

Where the dynamics of the first part were exactly known and  $K_D$  is a positive definite gain matrix:

$$K_D = \text{diag}(K_{d1}, \dots, K_{dn}), \quad K_{di} > 0$$

Using the proposed control Equation (228) yields:

$$\dot{V}(t) = s^T \left( \widehat{M}(q) \ddot{q}_r + \widehat{C}(q, \dot{q}) \dot{q}_r + \widehat{G} - K_D s - C(s + \dot{q}_r) - G - M \ddot{q}_r \right) + \frac{1}{2}s^T \dot{M} s + \tilde{\varphi}^T \Gamma \dot{\tilde{\varphi}}$$

$$\dot{V}(t) = s^T \left( \tilde{M}(q) \ddot{q}_r + \tilde{C}(q, \dot{q}) \dot{q}_r + \tilde{G} - K_D s - C s \right) + \frac{1}{2}s^T \dot{M} s + \tilde{\varphi}^T \Gamma \dot{\tilde{\varphi}}$$

The manipulator equation of motion are linear in the inertia parameters in the following sense. There exists an  $n \times l$  regression function,  $Y(q, \dot{q}, \ddot{q})$  and an  $l$

dimensional parameter vector  $\tilde{\varphi}$  such that the equation of motion (71) can be written as:

$$\tilde{M}(q)\ddot{q}_r + \tilde{C}(q, \dot{q})\dot{q}_r + \tilde{G} = Y(q, \dot{q}, \ddot{q})\tilde{\varphi} \quad (229)$$

Therefore,

$$\begin{aligned} \dot{V}(t) &= s^T (Y\tilde{\varphi} - K_D s - C s) + \frac{1}{2} s^T \dot{M} s + \tilde{\varphi}^T \Gamma \dot{\tilde{\varphi}} \\ \dot{V}(t) &= s^T (Y\tilde{\varphi} - K_D s - C s) + \frac{1}{2} s^T (\dot{M} - 2C) s + \tilde{\varphi}^T \Gamma \dot{\tilde{\varphi}} \\ \dot{V}(t) &= s^T (Y\tilde{\varphi} - K_D s) + \tilde{\varphi}^T \Gamma \dot{\tilde{\varphi}} \\ \dot{V}(t) &= \tilde{\varphi}^T Y^T s - s^T K_D s + \tilde{\varphi}^T \Gamma \dot{\tilde{\varphi}} \\ \dot{V}(t) &= \tilde{\varphi}^T (Y^T s + \Gamma \dot{\tilde{\varphi}}) - s^T K_D s \end{aligned}$$

Then, the parameter of the adaptive law is designed as follows:

$$\dot{\hat{\varphi}} = -\Gamma^{-1} Y^T s \quad (230)$$

Therefore,

$$\dot{V} = -\Gamma^{-1} K_D s \leq 0$$

This results that the tracking error goes to zero:  $\tilde{q} \rightarrow 0$  as  $t \rightarrow \infty$

### 5.2.10. Simulation Results of 3-DOF Reconfigurable Robot

The tracking position of the first joint to a trigonometric reference signal has been shown in Figure 5.22 (above), the trajectory tracking error (central) and the control torque required to drive the joint (below). The tracking position of the second joint shown in Figure 5.23 has similar features as the first joint. The convergence of trajectories tracking has been assured by the adaptive rule Equation (230). The estimation values of the inertia parameters  $(\varphi_1, \varphi_2, \varphi_3, \varphi_4)$  are normalized and showed in figures 5.24 and 5.25, respectively. The parameter estimation error remains bounded rather than going to zero. The reason for this type of bounded parameter error is that the error system given by Equation (229) is constantly being excited by the input dynamics on the right-side of Equation (229).

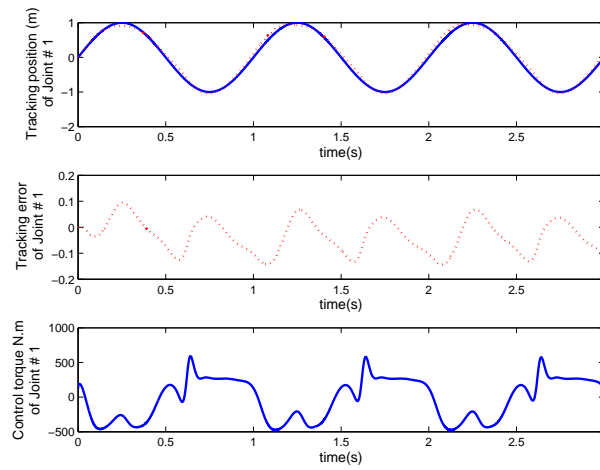


FIGURE 5.22. Tracking position of the joint 1 to a trigonometric function (above), reference position (solid) and actual position (dotted). The tracking error (central), and the control torque (below).

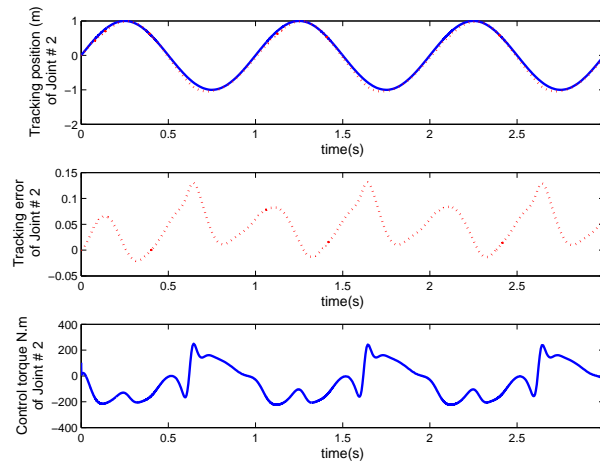


FIGURE 5.23. Tracking position of the joint 2 to a trigonometric function (above), reference position (solid) and actual positions (dotted). The tracking error (central), and the control torque (below).

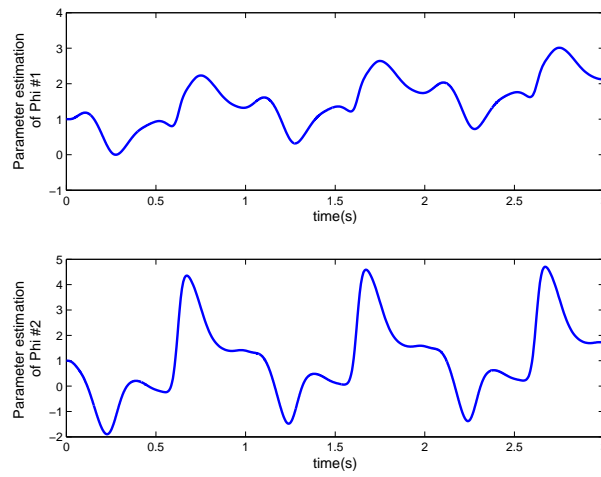


FIGURE 5.24. The normalized values of the inertia parameters  $\varphi_1$  (above) and  $\varphi_2$  (below).

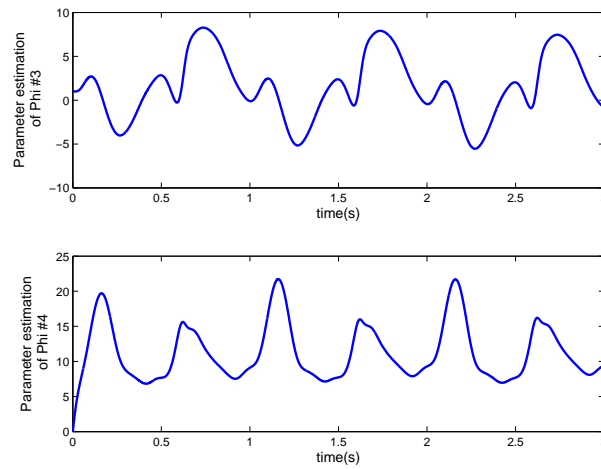


FIGURE 5.25. The normalized values of the inertia parameters  $\varphi_3$  (above) and  $\varphi_4$  (below).



## CHAPTER 6

### **Kinematic and Control Selection Algorithm**

To realize a reconfigurable robot, an algorithm has been developed that combines the selection of an applicable kinematic structural robot configuration with control design methods to optimize robot capability and performance. Assuming a spherical wrist attached to the end effector, the kinematic structures of common industrial robots are determined by the first three links and joints, which also define the external and internal workspace boundaries. A spherical wrist; which satisfies Piper's condition [45] when  $a_4 = 0, a_5 = 0$  and  $d_5 = 0$ ; only serves to orient the end effector within the workspace. The resulting reconfigurable robot includes eight kinematic structures: Cartesian TTT, Cylindrical RTT, TTR, TRT, Articulated RRR, Spherical/Scara RRT, TRR, and RTR.

In this chapter, an algorithm with configuration and control phases is developed to select an optimal kinematic configuration structure with the most applicable control approach to perform a specified trajectory with high tracking performance.

#### **6.1. Configuration Phase**

The configuration algorithm shown in Figure 6.1 starts by defining the geometric task in terms of its shape, dimensions, etc. To build the entire trajectory sequentially, the task is divided into segments according to trajectory shape (linear/curvature). To perform the first segment, the algorithm begins with the TTT kinematic structure. The kinematic model for this structure can be automatically generated using the generic kinematic algorithm developed in Chapter 2. The workspace of the kinematic structure is calculated from the Cartesian coordinates of the position vector partitioned from the homogeneous matrix transformation. Then, the workspace singularity of the configuration is evaluated by calculating the determinant of the Jacobian. The reachability property of the segment within the workspace is tested by calculating the inverse kinematics of the structure. In case

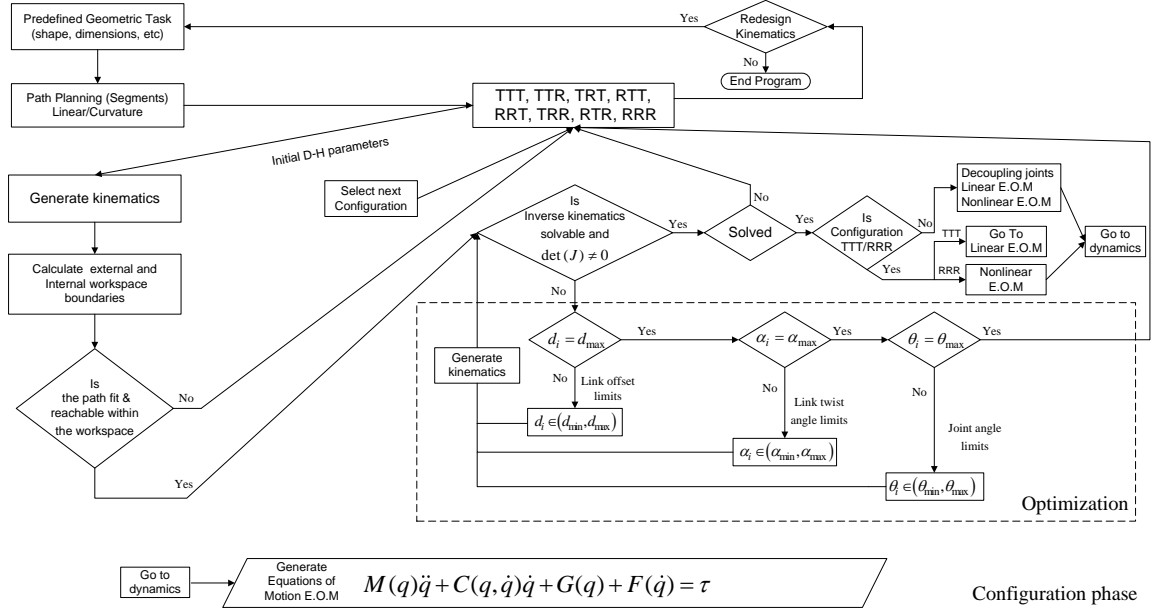


FIGURE 6.1. Configuration phase.

of a singular configuration, the algorithm will select the next kinematic structure and repeat the same tests. The structural kinematic selection continues until it arrives at a structure that satisfies the Jacobian and inverse kinematic conditions without singularity.

### 6.1.1. Internal D–H Parameters Optimization

For a selected configuration, the values of the three variable D–H parameters; link offset, link twist angle and the joint angle are used in internal loops to test the inverse kinematics and singularity conditions. The first internal loop starts with the minimum value of the link offset parameter. The kinematic model is generated according to the selected new link offset value. Then, the resulting kinematic model is tested for the inverse kinematics and singularity conditions. If the test fails, then the loop continues to choose the next offset value and repeat the same two condition tests. The internal loop continues till reaching the maximum link offset limit value. In case the two conditions are not satisfied, the algorithm will switch to the next loop, the link twist angle range values. The process of calculating the kinematic model and testing the two conditions will continue until reaching the maximum limit value of the link twist angle. If the two conditions are not satisfied,

the loop will switch to the joint angle internal loop, and the same process will be repeated as the previous loops till reaching the maximum joint angle value. If the inverse kinematics and Jacobian singularity are not satisfied using the three internal loops, then the algorithm will select the next kinematic structure and the entire process will be repeated again. The structural kinematic selection continues until it arrives at a structure that satisfies the inverse kinematic and Jacobian without singularity. A manipulability index (explained in details in Chapter 2) can be used here to determine the optimal configuration and its singularity. If the algorithm has called all the eight kinematic structures and the conditions are not satisfied, then the geometric task is out of the robot design and the kinematic specifications should be changed to adapt to the required geometry task. The algorithm calculate the dynamics whenever the two conditions are satisfied generating the required equations of motion automatically. If the configuration is TTT, then linear equations of motion would have to generate for control purposes. If the resulting configuration is RRR, then the generated equations of motion are nonlinear due to the dynamic coupling of revolute joints. In case the kinematic structure has revolute-translational joints, such as the Scara kinematic structure, the equations of motion can be decoupled into linear and nonlinear subsets.

## 6.2. Control Phase

Having generated the dynamic equations for a specific kinematic configuration, a reconfigurable control methodology shown in Figure 6.2 is developed to select a control approach depending on the parameters type. This control methodology is constructed to find an appropriate control approach that minimizes conservatism and maximizes obtained performance specifications. Selecting a control approach is not easy to perform as different methods are formulated in different time and frequency domains. It is difficult, if not impossible, to reformulate the reconfigurability problem in one domain without losing design insights and introducing conservatism.

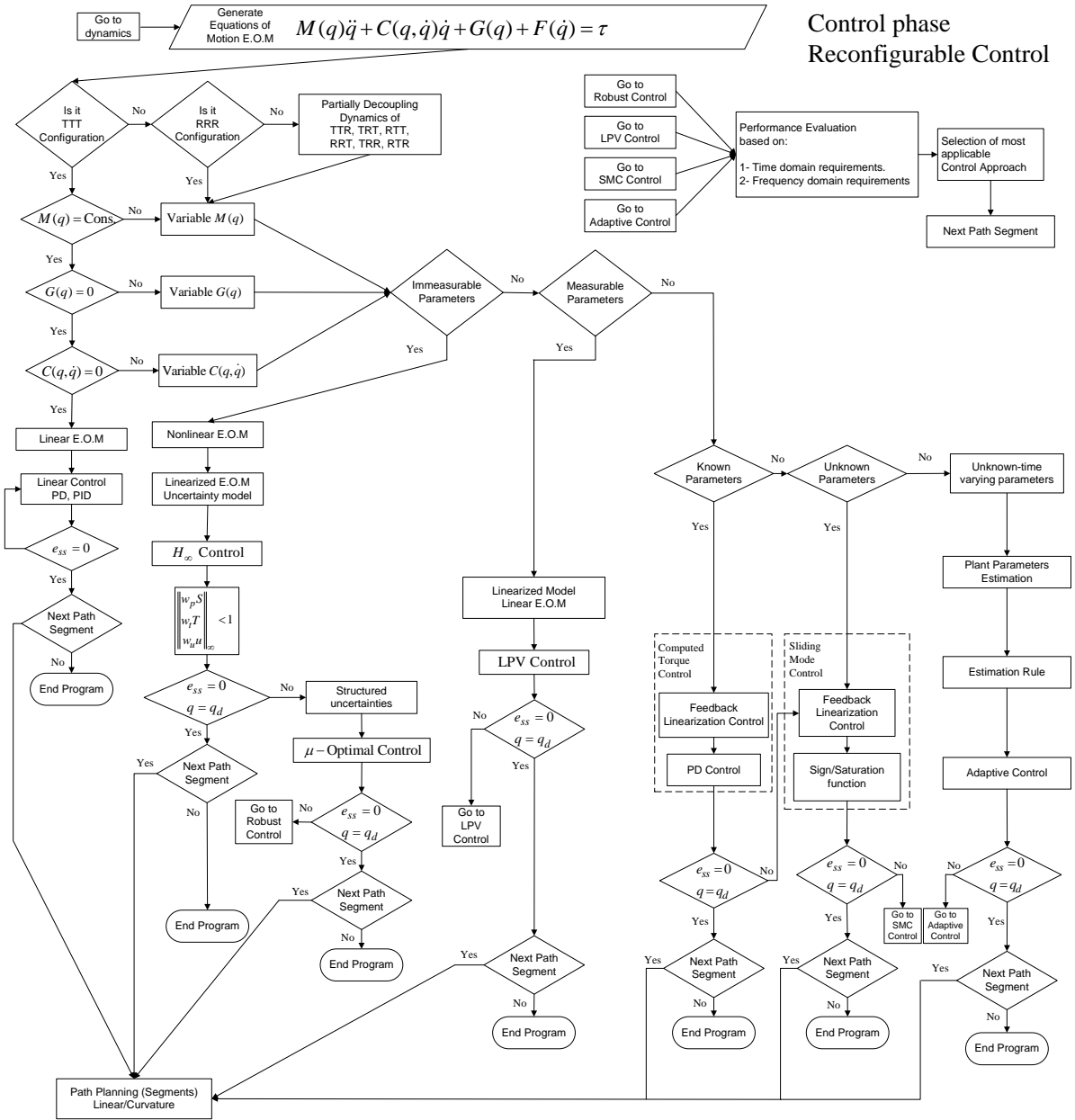


FIGURE 6.2. Control phase.

In this phase, the most applicable control approach is selected to achieve high tracking performance for the optimal kinematic structure. The general time and frequency performance requirements are given as follows:

**Time Domain Performance Requirements**

- Steady state error ( $e_{ss}$ ).
- Speed of response (Time constant).

- Rise time and settling time.
- Relative stability.

### Frequency Domain Performance Requirements

- At low frequency bandwidth: disturbance rejection and reference tracking of low frequency commands can be achieved by shaping the sensitivity function given in Equation (78).
- At high frequency bandwidth: trajectory tracking and noise attenuation can be achieved by shaping the complementary function  $T = I - S$ .

#### 6.2.1. Dynamic Parameter Properties

The dynamic equations of motion of a reconfigurable robot are fully characterized by the type of its parameters. In case of translational motion, these equations are linear with constant mass, gravity and Coriolis parameters. For rotational motion, the inertia, gravity and Coriolis parameters are variable and nonlinear due to inertial coupling, kinematic coupling, gravity load, friction term saturation, and transcendental functions. The properties of the dynamic parameters of a reconfigurable robot are presented in Table 6.1.

TABLE 6.1. Dynamic parameter properties of a reconfigurable robot

Parameter	Configuration dependent	Nonlinear	Bounded value	Structured uncertainty
Inertia	×	×	×	×
Coriolis Centrifugal	×	×		×
Gravity	×	×		×
Viscous friction			×	×

The inertial, Coriolis and gravity parameters are nonlinear and their values are configuration dependent as calculated and showed in Figures 3.8 and 3.10.

#### 6.2.2. The Reconfigurable Control Algorithm

In general, the reconfigurable control algorithm is classified into linear, nonlinear and robust control methods. This classification depends on the parameter properties of the equations of motion. The algorithm shown in Figure 6.2 starts when the

equations of motion are linear (TTT configuration), a linear control, such as PD or PID, is selected to track a reference trajectory achieving the required performance. For an Elbow manipulator (RRR configuration) and partially decoupled dynamics (TTR, TRT, RTT, RRT, TRR, RTR), the physical dynamic parameters are variable, uncertain and nonlinear. The types of these dynamic parameters can be specified as follows:

- Immeasurable parameters (nonlinearity feature).
- Measurable parameters (configuration dependent) parameters.
- Known parameters (exact parameter knowledge).
- Unknown parameters (constant mass change in case of pick and place).
- Unknown and time varying parameters (variable mass change).

When the uncertain dynamic parameters are not measurable, the algorithm selects the robust control approaches. With this approach, the nonlinear property of the dynamic parameters is modeled as parametric uncertainties for a linearized model. By applying a  $\mu$  controller, the resulting performance can be improved in case of structured parametric and dynamic uncertainties. With robust control methods such as  $H_\infty$  and  $\mu$  controllers, the immeasurable parameters are can be modeled as uncertainties into the following types:

- Dynamic uncertainty (structural modes).
- Parametric uncertainty (parameter variations).
- Time invariant (constant) uncertainty.
- Time varying uncertainty.

In case of parameter dependent configuration (on-line measurable parameters), a robust Linear Varying Parameter (LPV) controller is selected to obtained the specified performance requirements. In this approach, the measured and time varying parameters such the inertial and viscous friction are fed back to the controller to obtain high tracking performance specifications. If the dynamic parameters are exactly know, a feedback controller can be applied as an inner loop with linear PD or PID controllers in the outer loop control. Sliding mode control can be applied in case of unknown parameters which have the a robust component (sign/saturation function) in the outer loop control. In case the equations of motion are nonlinear, then a control approach is selected as follows:

- If the robot has variable and immeasurable parameters: the nonlinear equations of motion have to be linearized and the variations would be modeled as bounded uncertainties within known limits. An LFT (Linear Fractional Transformation) control problem is composed and an  $H_\infty$ -optimal control is designed to ensure the maximum norm of a stacked performance cost function is less than one. The resulting control design is conservative in allowing less sizable uncertainties to ensure Nyquist stability of the system. Therefore, for structured uncertainties,  $\mu$ -optimal controllers can be designed, which allows more sizable parameter variations and improves the robust performance.
- Variable and measurable parameters: if the parameters are measurable, then for a linearized system, a linear parameter varying (LPV) control can be designed, which results in high tracking performance in the presence of disturbances. Gain scheduling or linear parameter varying (LPV) techniques are used for controlling LPV systems. An LPV controller consists of designing a linear time invariant LTI controller that is adapting itself when the operating conditions change. In this control method, the system is assumed to depend affinely on a measured vector of time varying parameters. Assuming on-line measurements of these parameters, they can be fed to the controller to optimize the performance and robustness of the closed loop system.
- Known parameters of the nonlinear equations of motion: a feedback linearization can be designed to cancel the known parameters of the robot. This control inversion model step is regarded as an inner loop control to decouple the system dynamics and renders double integrators. Then, PD controllers can be designed in the outer loop control to drive the error system to zero exponentially.
- Unknown and uncertain parameters: a sliding mode control (SMC) (feedback linearization control plus robust component) can be designed to achieve stability and high tracking performance in the presence of parametric and unmodeled dynamic uncertainties. The most significant feature of the sliding mode control is the complete insensitivity to parametric

uncertainty and external disturbances during the sliding mode. The SMC uses a high speed switching control law to achieve two objectives. Firstly, it drives the nonlinear system's state trajectory along a specified and user chosen surface in the state space, which is called the sliding or switching surface. This surface is named the switching surface because a control path has one gain if the state trajectory of the system is above the surface and a different gain if the trajectory drops below the surface. Secondly, it maintains the system's state trajectory on this surface for all subsequent times.

- Constant or slowly time varying parameters: an adaptive control with adaptation rule can be designed to achieve stability and satisfy the tracking performance requirements. Adaptive control is an approach to control systems that have constant or slowly-varying uncertain parameters. The basic theory in adaptive control is to estimate the uncertain system parameters on-line based on measured system signals, and exploit the estimated parameters in the control input computation. Thus, an adaptive control can be regarded as a control with on-line parameters estimations which maintain consistence performance of a system in the presence of uncertainty or unknown variation in system parameters. This leads one to consider the adaptive control approach as a way of automatically adjusting the controller parameters in the face of changing robot dynamic parameters.

The algorithm shown in Figure 6.2 performs the selection (as per the criteria above) of the applicable kinematic configuration and control method for each segment sequentially, returning to the path planning node after completion of each segment to start the process for the next one. Thus, the reconfigurable control scheme shown solves the control design problem from a pragmatic perspective, rather than seeking an exact treatment of the performance problem.



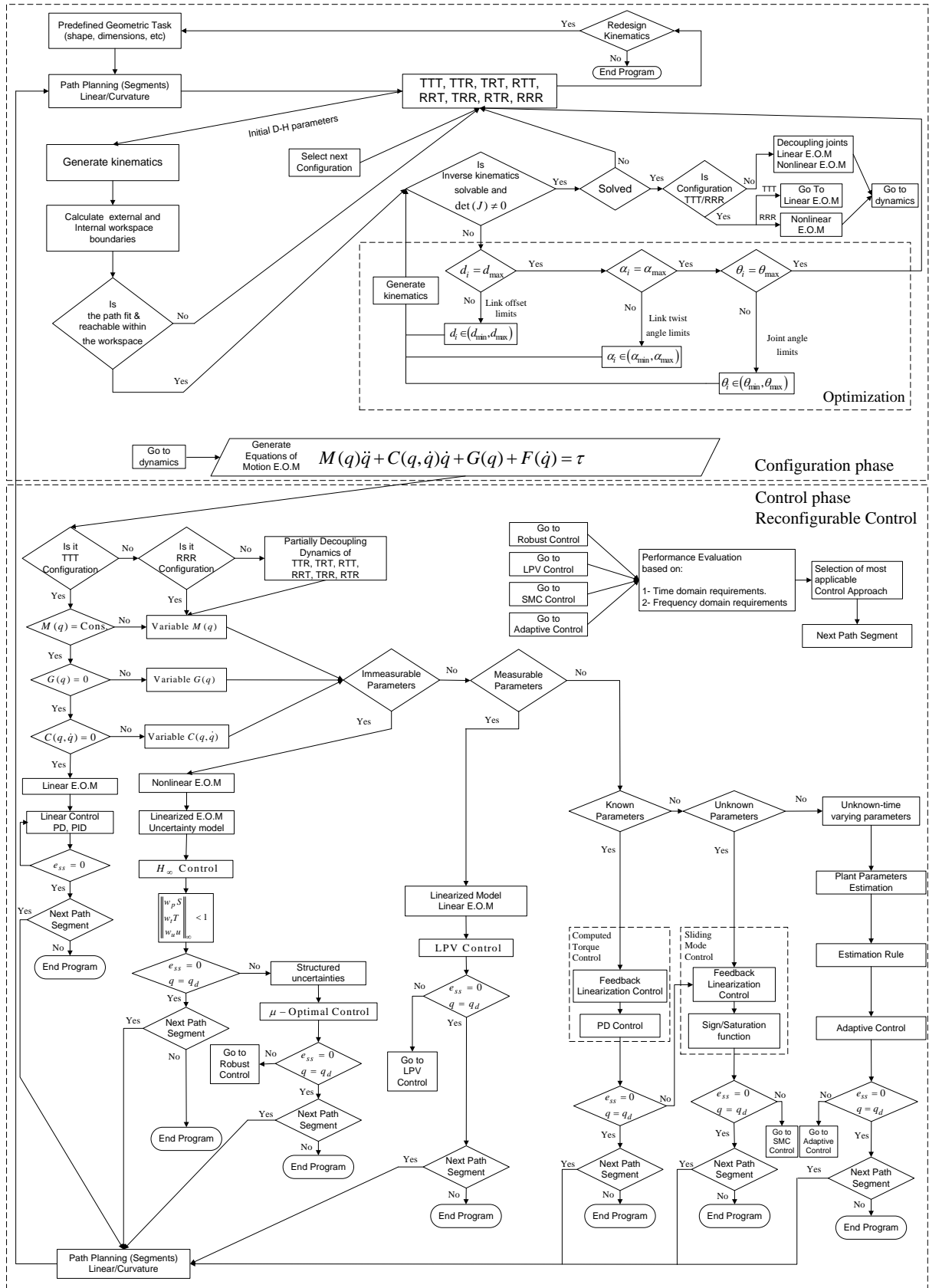


FIGURE 6.3. A comprehensive algorithm for configuration and control of a reconfigurable robot

### 6.3. Algorithm Implementation (Simulations and Results)

Based on the singularity conditions of the inverse kinematic and Jacobian of the given trajectory segments, two reconfigurable robots with certain kinematic structures are selected. For a specified segment, the singularity conditions are verified for each kinematic structure starting with the TTT, TTR, TRT, RTT, RRT, TRR, RTR, and RRR configurations. The kinematic structure with the smallest condition number of the Jacobian matrix is chosen. Two reconfigurable robots with specified kinematic structures are simulated and viewed to follow two different trajectories. Position and trajectory control methods are employed to achieve the specified performance specifications.

#### 6.3.1. Trajectory of Two Circles in Joint Space Motion

Two sheet cylinders are manufactured with radius 0.10 m and centered at (-0.25, 0.25, -0.5) and (-0.25, 0.25, 0) meters, as shown in Figure 6.4. The Jacobian singularity condition of the eight kinematic structures is determined for each of the following poses: The center point (-0.25, 0.25, -0.5) of the lower cylinder and for the other vertices (-0.35, 0.25, -0.5), (-0.25, 0.15, -0.5), (-0.15, 0.25, -0.5), and (-0.25, 0.35, -0.5). The RRR kinematic structure has the smallest value of the Jacobian condition number, which indicates that the cylinders are reachable within the RRR workspace and without singularity compared to the other configurations. The D-H parameters of the RRR reconfigurable manipulator are given in Table 3.13. The motion of the end effector starts at the lower circle executes a circular motion in the  $xy$  planes of radius 0.20 m around the manufactured cylinder. Then it follows a line in vertical motion to execute the upper circle. The joint coordinates of the lower circle are depicted in Figure 6.5 which show a change in the fourth joint angle from  $\pi$  to  $-\pi$ . This change occurs in a very short number of samples which indicates the flip of the end effector. The Cartesian motion of the end effector is shown in Figure 6.6 and it is clear that the motion was executed in the  $xy$ -plane. Euler angles of the end effector shown in Figure 6.7 indicate a change of the roll angle from  $-\pi/2$  to  $\pi/2$  and a high frequency oscillation of the yaw angle to maintain a circular motion in the  $xy$ -plane. The manipulability index is calculated for the circular motion and shown in Figure 6.8. The index value approaches zero

when the fourth joint angle of the end effector flips its direction with almost 180 degrees to execute the required circle path.

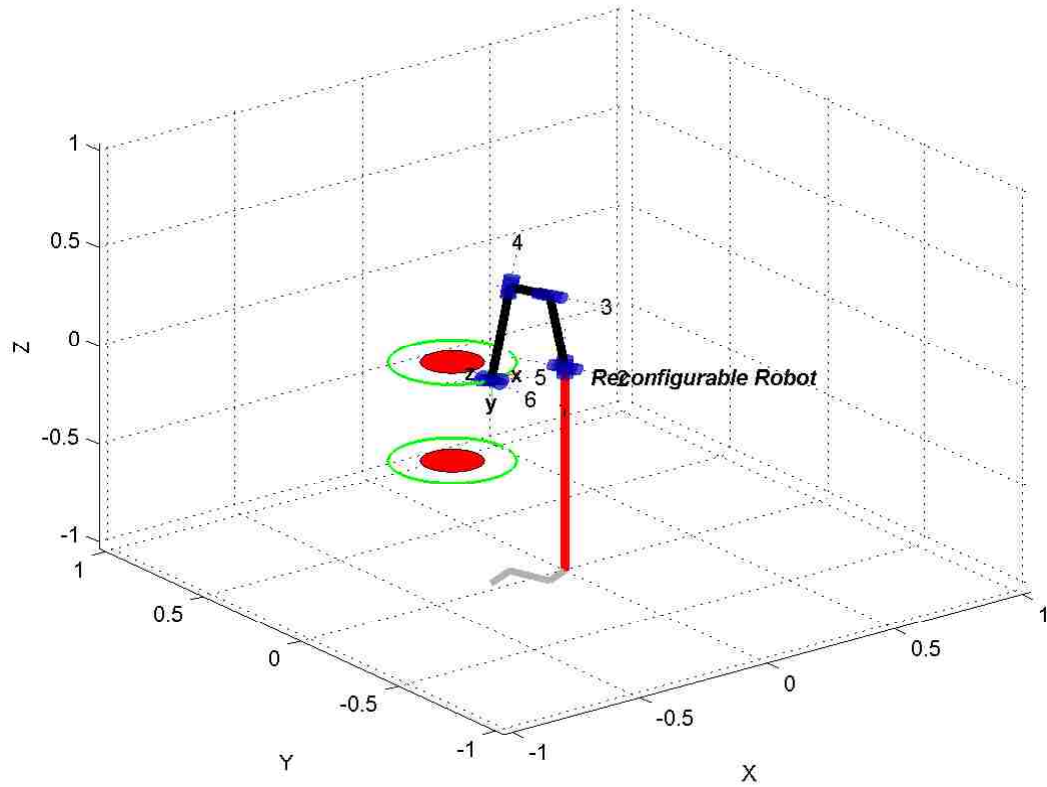


FIGURE 6.4. A reconfigurable robot follows a trajectory motion (lower and upper circles)

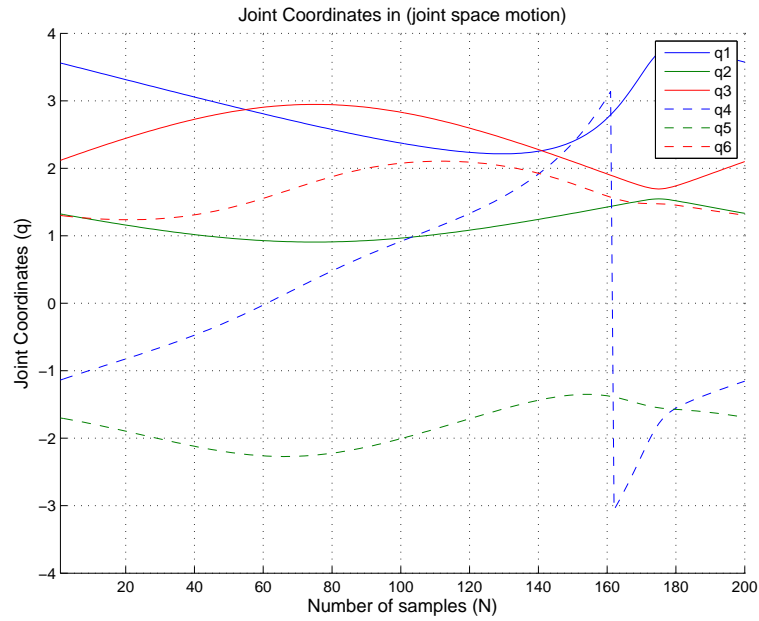


FIGURE 6.5. The joint coordinates of a reconfigurable robot during the joint space path.

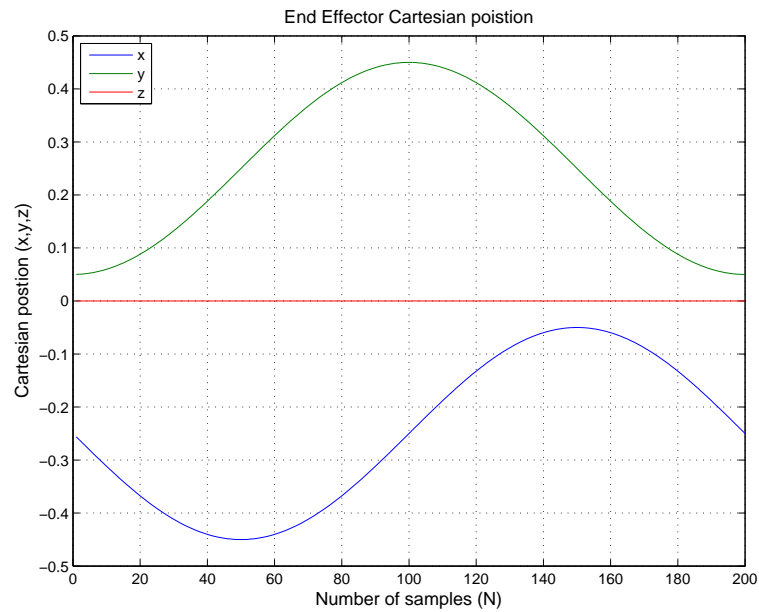


FIGURE 6.6. Cartesian position  $(x,y,z)$  of the end effector during the joint space trajectory.

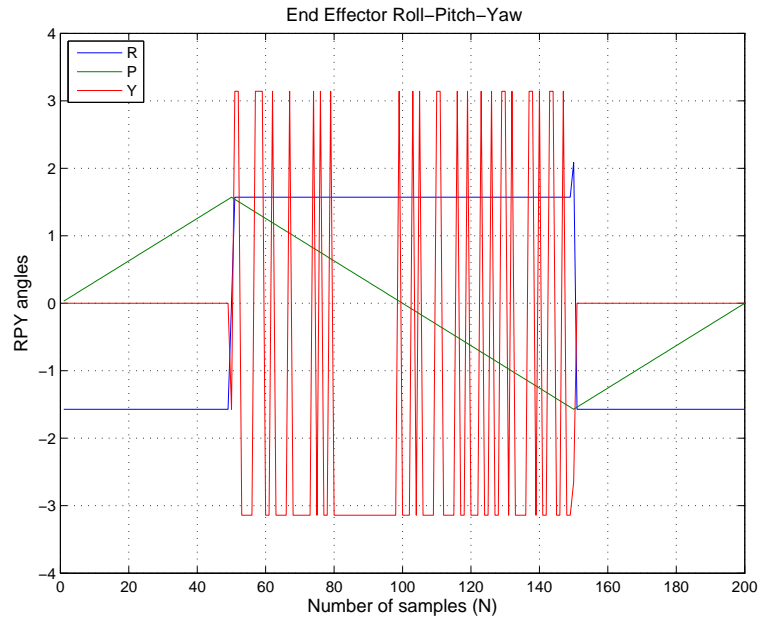


FIGURE 6.7. The Euler angles (roll, pitch, yaw) of the end effector.

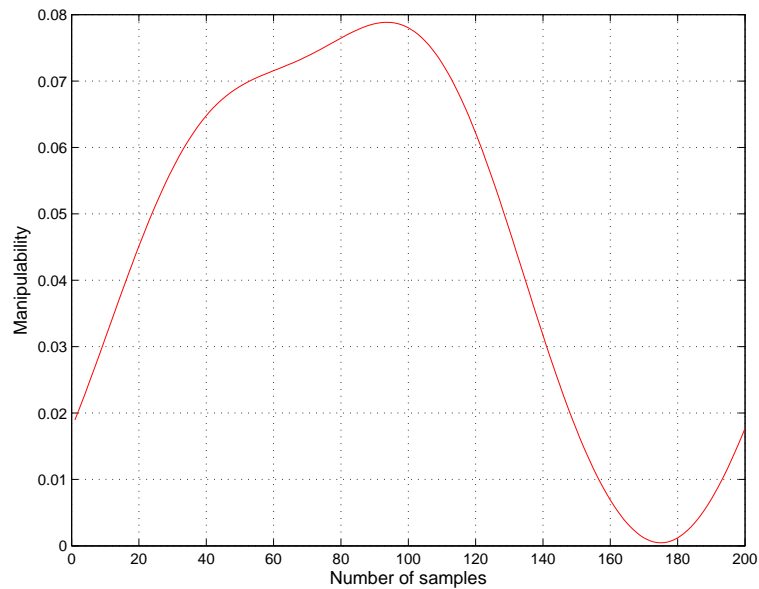


FIGURE 6.8. Manipulability of the reconfigurable robot following a joint space trajectory.

### 6.3.2. Straight Line Trajectory in Joint Space

Given the start  $(0.4064, 0.5, 0.3303)$  and final  $(0.8032, 0.5, 0.3302)$  poses of a straight line trajectory, the condition number of the Jacobian matrix is determined for each of the eight configurations. The RRT kinematic structure has the

smallest value of the Jacobian condition number compared to the other configurations. Therefore, a reconfigurable manipulator with RRT kinematic structure is selected and its workspace with three envelopes is generated and shown in Figure 6.9. The first envelope is generated when the joints of the robot are set up in a revolute structure. The second and third envelope are created when the reconfigurable joint moves translational to the next envelope. It is considered that the end effector is moving from pose  $(0.4064, 0.5, 0.3303)$  with end effector orientation of 180 degrees to the pose  $(0.8032, 0.5, 0.3302)$  with changing orientation to 90 degrees. This motion lies in the  $xy$ -plane with the end effector orientated downward. The required trajectory lies within the workspace envelope of the reconfigurable manipulator by calculating the Jacobian singularity at the start and end trajectories. Using the geometric inverse kinematics, the reachability condition of the final pose is validated and satisfied. The initial and final joint coordinates associated with the two poses are shown in Figure 6.10. The joint coordinates trajectories are moving smoothly in the joint space motion between the two poses as shown in Figure 6.11. The Cartesian motion of the end effector is shown in Figure 6.12 where the  $x$ -coordinate shows a smooth transition along  $x$ -coordinate from the location 0.4064 to 0.8032. The figure also shows a slight deviation in the  $y$ -axis due to the joint space motion of the manipulator. The Cartesian locus of the end effector in the  $xy$ -plane shown in Figure 6.13 indicates that the trajectory is not a straight line. This is expected since only the Cartesian coordinates of the end poses were specified. As the robot rotates about its waist joint during the motion, the end effector will follow a circular arc. The orientation of end effector in roll-pitch-yaw angles form are plotted in Figure 6.14 against time. The roll angle varies from  $\pi$  to  $\pi/2$  as specified in the initial and final poses.

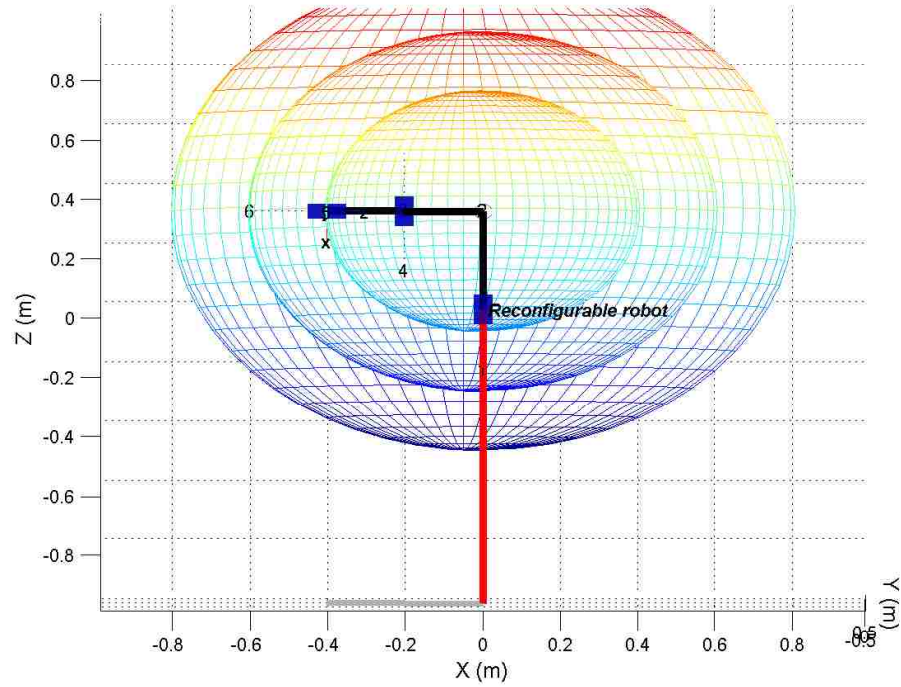


FIGURE 6.9. Front view of the workspace in the  $xz$ -plane of a reconfigurable robot.

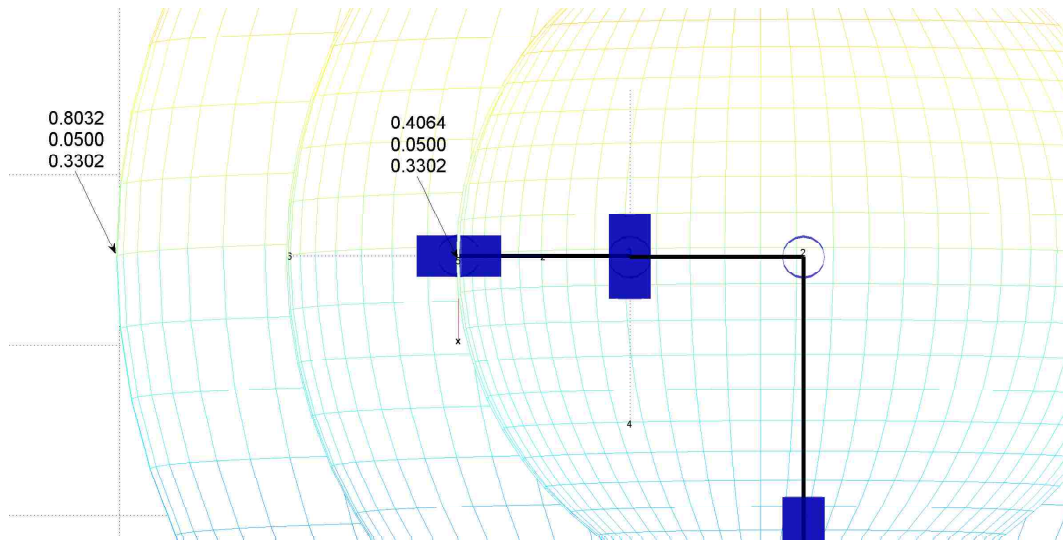


FIGURE 6.10. End effector motion from pose  $(0.4064, 0.5, 0.3303)$  to  $(0.8032, 0.5, 0.3303)$ .

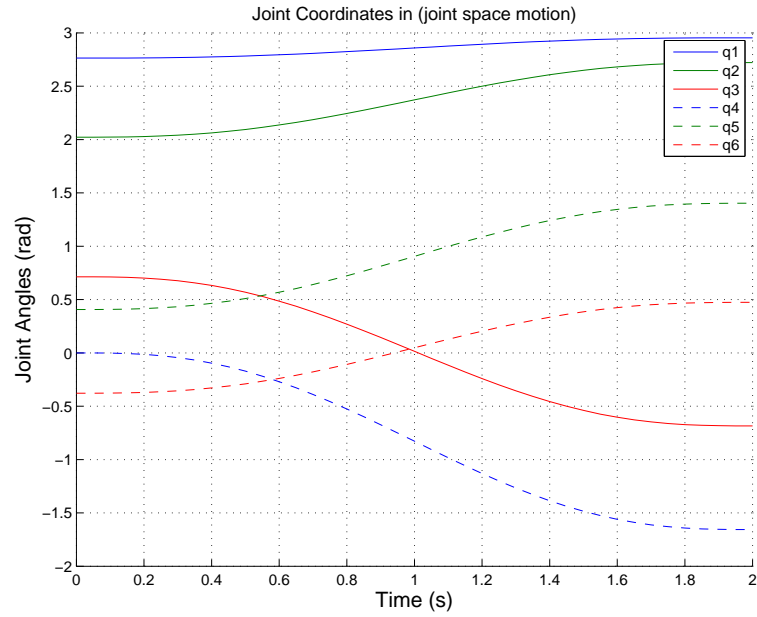


FIGURE 6.11. Joint coordinates motion versus time.

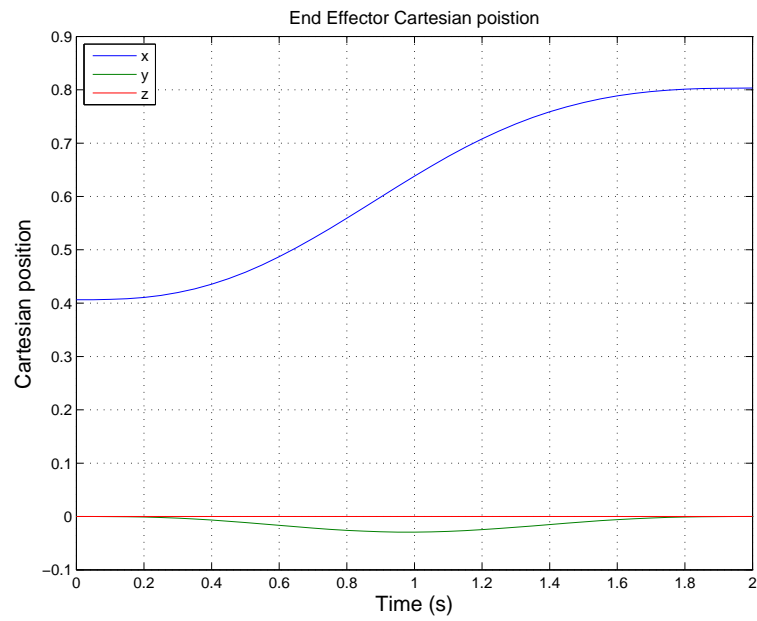


FIGURE 6.12. Cartesian position of the end effector versus time.



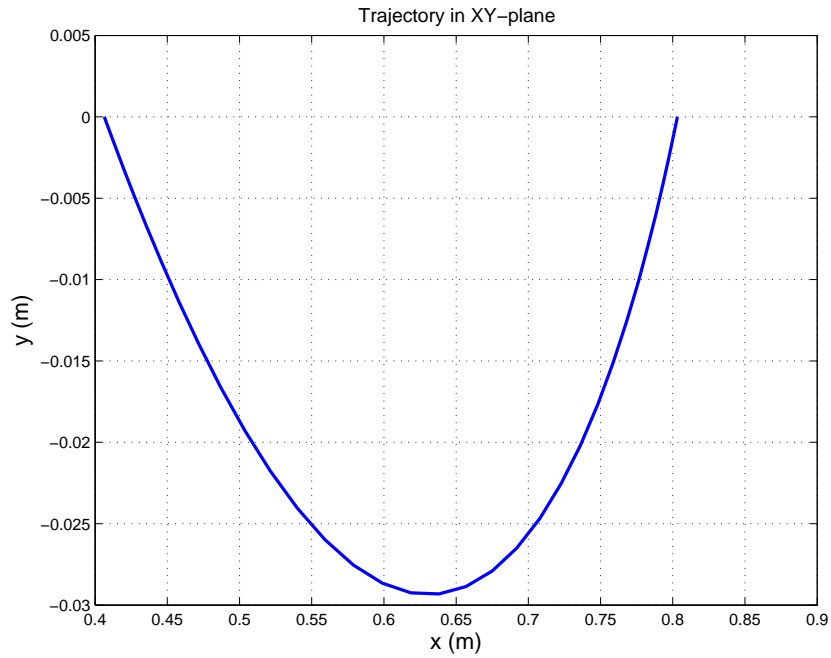


FIGURE 6.13. Cartesian position of the trajectory in the  $xy$ -plane versus time.

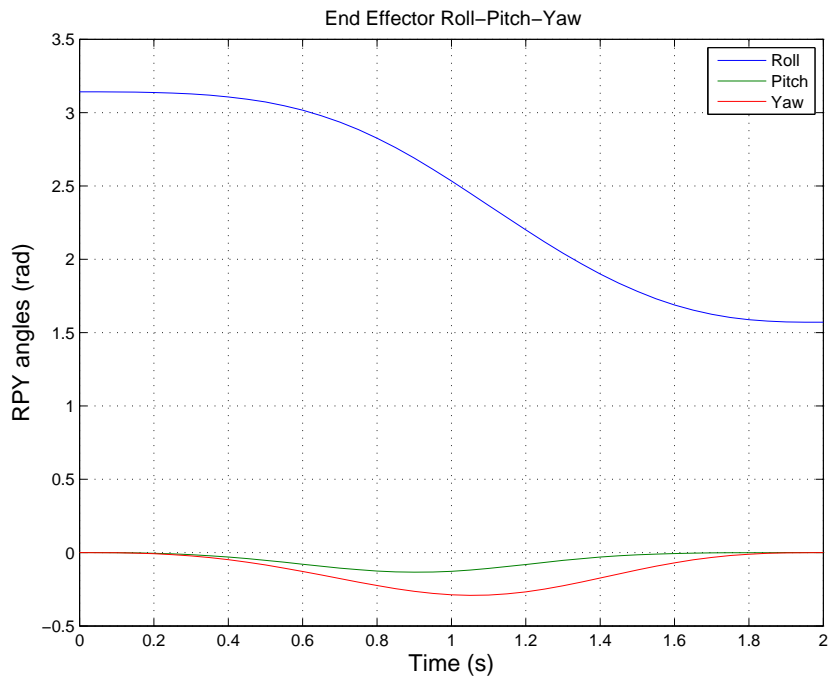


FIGURE 6.14. Euler angles, roll-pitch-yaw of the end effector versus time.

### 6.3.3. Position (Linear) and Trajectory (Nonlinear) Control Method Selection

Position and trajectory control methods are selected based on the dynamic parameter type as follows:

- measured or unmeasured parameters leads to position control methods.
- known or unknown parameters leads to trajectory control methods.
- constant or time varying parameters leads to adaptive control method.

Position control involves linear control methods such as robust  $H_\infty$ ,  $\mu$  and Linear Varying Parameter (LPV) controls. Robust control methods model the nonlinear property of the parameters as parametric uncertainties for a linearized plant. A manipulator with RRR configuration, initially at rest ( $q_1 = 0, q_2 = 0, q_3 = 0$ ), is commanded to a step signal. The inverse weighting function shown in Figure 6.15 is designed with large gain at low frequencies to reject disturbance and follow reference commands, which represents the required performance objectives. The nonlinear property of the inertia, Coriolis and viscous friction are modeled as parametric uncertainties. Using LFT control configuration, a linearized model with uncertainties is set up for robust control design. The resulting step and disturbance responses are shown in Figure 6.16 and 6.17, respectively. The step response shows an overshoot of 1.73 after 1.8 sec and a settling time of 6.85 sec. The time domain performance specifications can be improved by using  $\mu$  control for structured uncertainties.

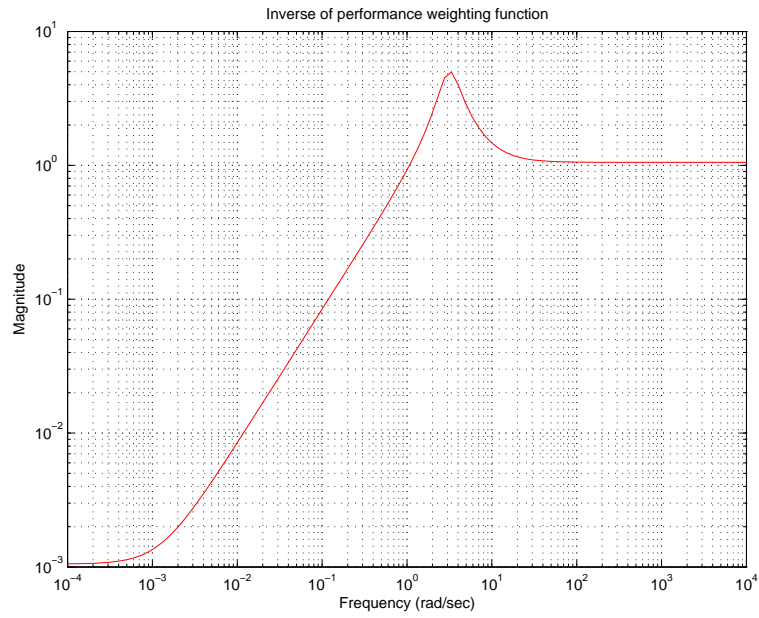


FIGURE 6.15. Performance weighting function of the sensitivity function  $S = \frac{1}{(1+GK)}$

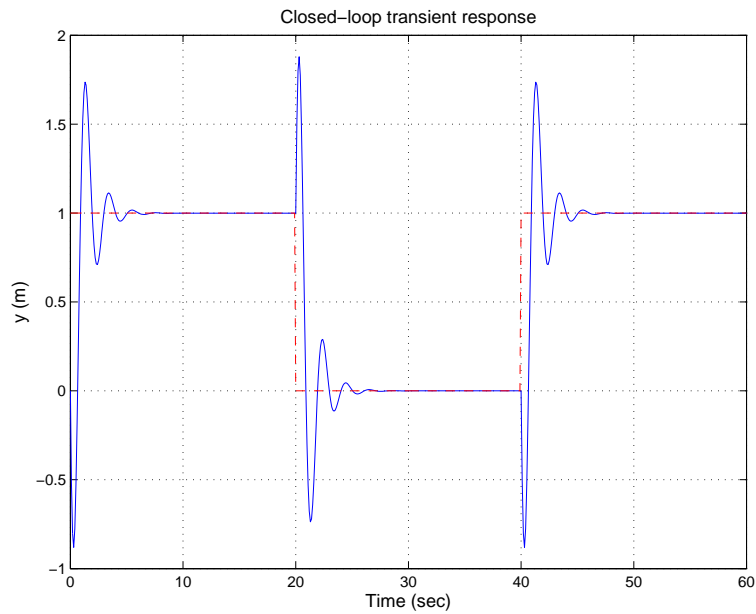


FIGURE 6.16. Step response of  $H_\infty$ -control.

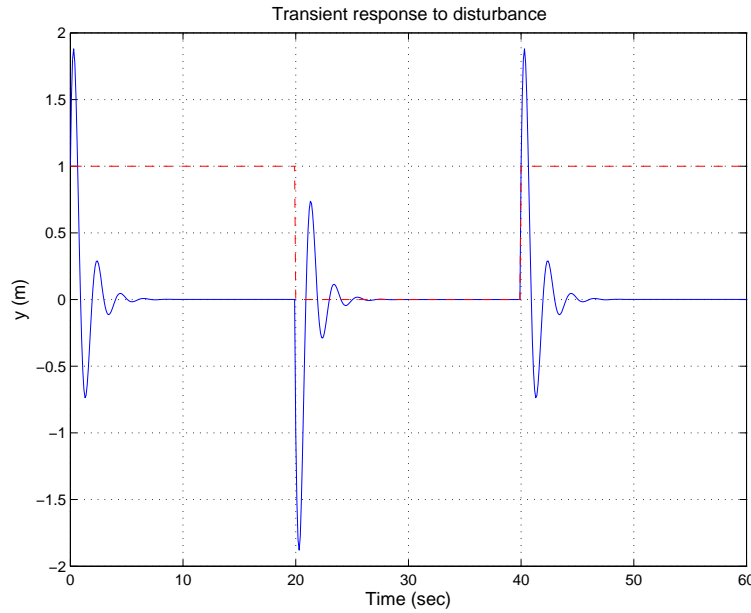


FIGURE 6.17. Disturbance response using  $H_{\infty}$ -control.

Selecting the LPV control method, the inertia and viscous friction are considered to be configuration dependent parameters and measured on-line. The performance weighting function shown in Figure 6.18 is designed to have high gain at low frequencies to satisfy the performance requirements in disturbance rejection and tracking reference command. The resulting step response for 20 frozen parameters shown in Figure 6.19 indicates the high speed of response, 1% overshoot and 1.56 sec settling time. The control amplitude (joint torque) shown in Figure 6.20 implies the torque needed is mostly high at the start time to move the first joint from its initial to final position. The steady state error keeps low value after for the simulated frozen parameters as shown in Figure 6.21.

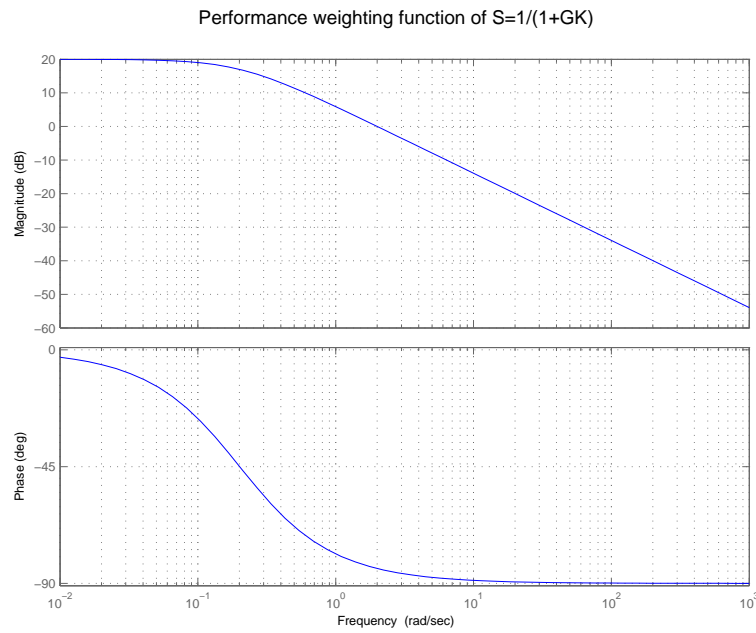


FIGURE 6.18. Frequency response of a weighted sensitivity function  $S = \frac{1}{(1+GK)}$

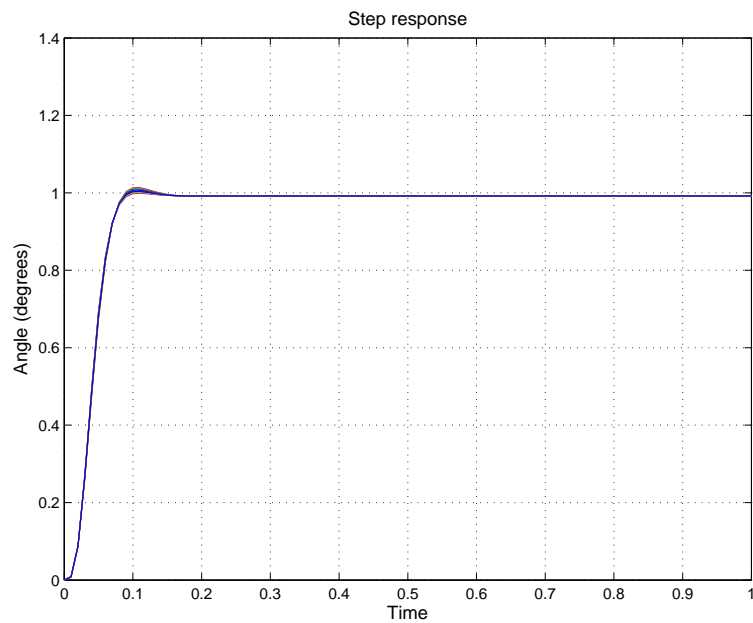


FIGURE 6.19. Step response of 20 frozen parameters using gain-scheduled controller.

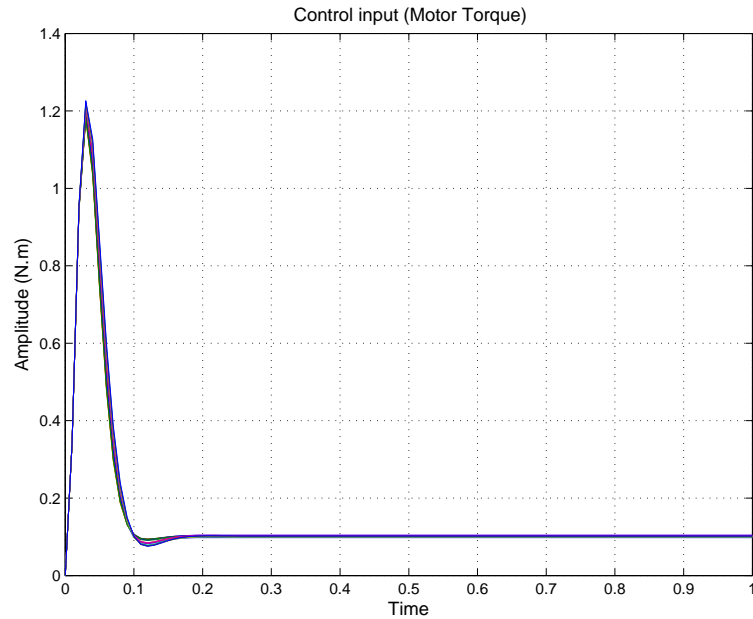


FIGURE 6.20. Gain-scheduled control amplitude.

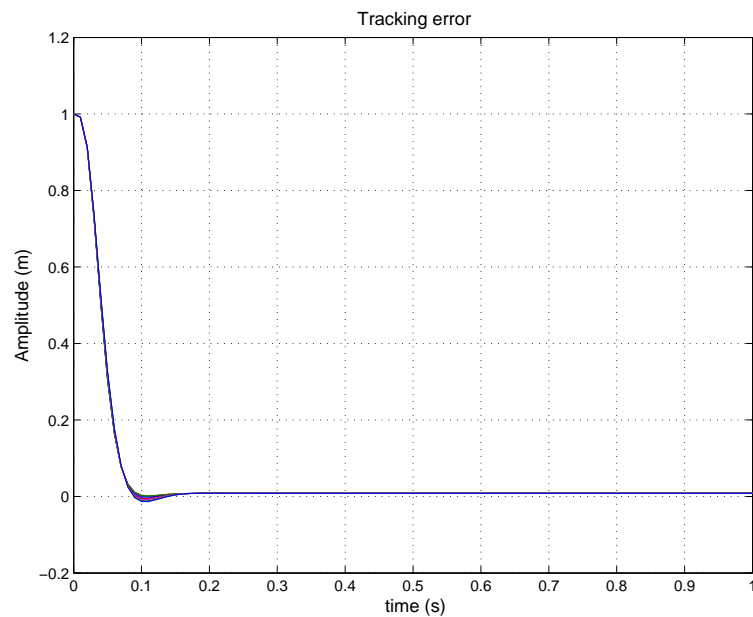


FIGURE 6.21. Tracking error using gain-scheduled control.

The algorithm leads to the SMC control method when the dynamic parameters are unknown. The feedback linearization control (inner loop control) cancels the nominal values of the parameters where the variation is compensated by a robust control component at the outer loop. The trajectory tracking of the first joint of the RRR configuration manipulator is shown in Figure 6.22. The tracking error

shown in Figure 6.23 points out that the tracking error goes to zero, achieving the performance requirements. In the case of time varying parameters (varying payload mass) the algorithm leads to employ the adaptive control method. The trajectory tracking of the first joint is shown in Figure 6.24. The tracking error shown in Figure 6.25 goes to the origin satisfying the performance requirements in disturbance rejection and following the reference command.

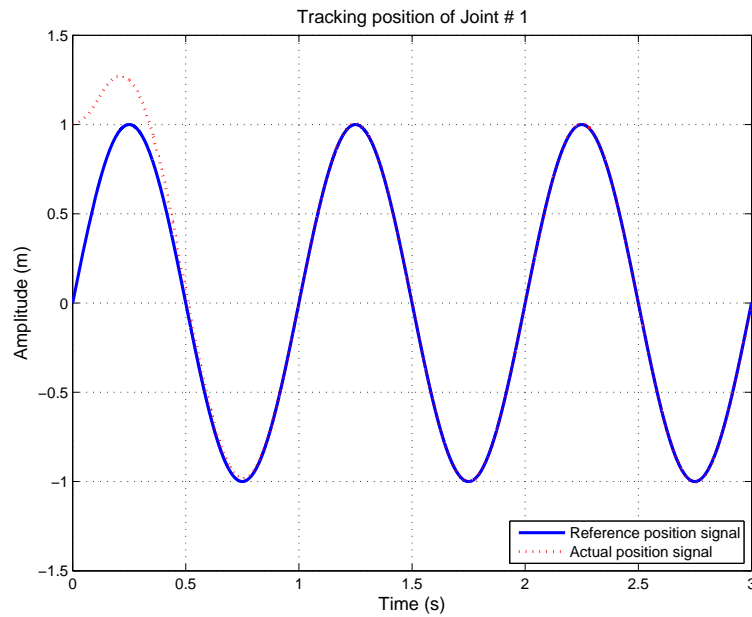


FIGURE 6.22. Trajectory tracking for a trigonometric reference command using SMC method.

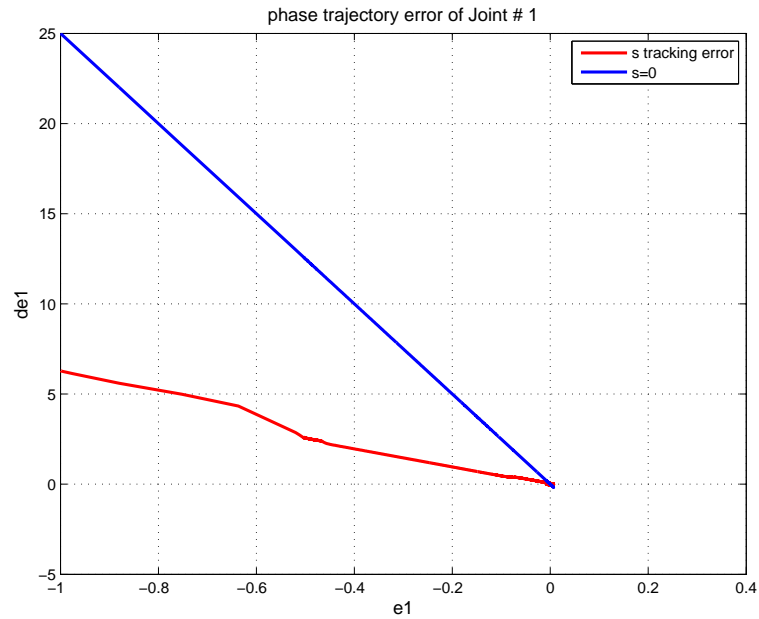


FIGURE 6.23. Tracking error for unknown parameters using SMC method.

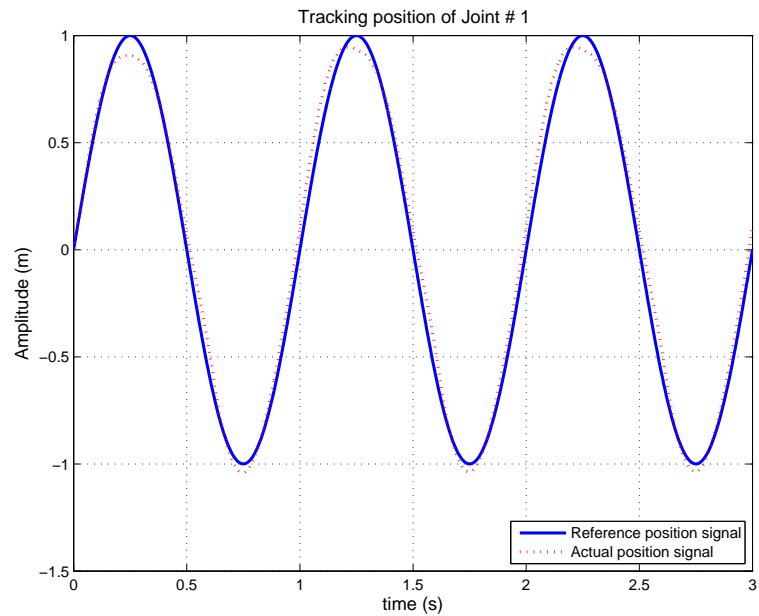


FIGURE 6.24. Trajectory tracking for a trigonometric function using the adaptive control method



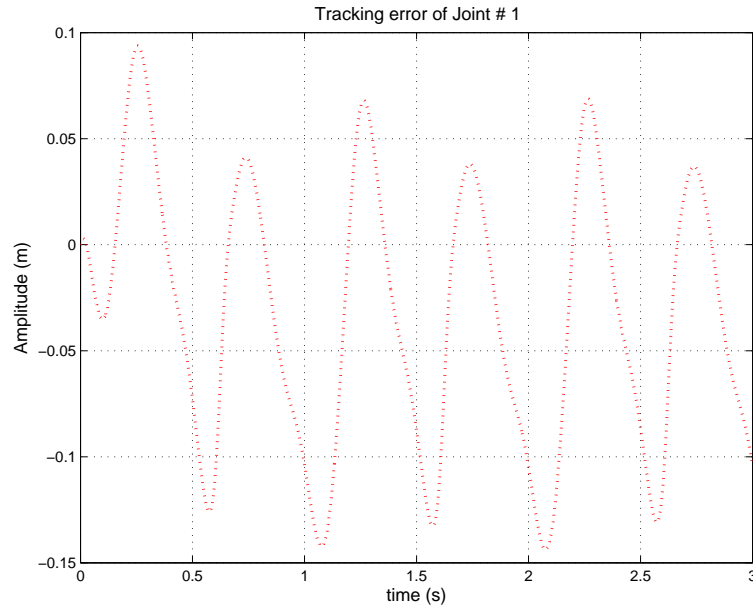


FIGURE 6.25. Tracking error for time varying dynamic parameters using adaptive control method

#### 6.4. Comparison between Linear and Nonlinear Control Approaches

Based on the dynamic parameter type, two main control approaches (linear robust and nonlinear controllers) have been employed to achieve the required tracking performance for the most applicable kinematic configuration. The properties of the two control methods are viewed and compared as follows:

- A robust control is capable of accounting for unmodeled dynamics, parameter variations, disturbance rejection, and command tracking. It is conservative against nonlinearities or large parameter variations. The adaptive control approach, such as parametric adaptive control, represents a design methodology which enhances system design response, especially against unpredicted variations.
- Gain scheduling controllers can be applied for systems with a high rate of parameter variation. The application of gain scheduling controllers in case of a system with highly varying dynamics is assumed to be beneficial in terms of the closed loop performance. The application of gain scheduling instead of robust control should be considered when the parameter is measurable.

- The major drawback of classical gain scheduling involves the lack of guaranteed global robustness, performance and stability. Only in the case of slowly varying parameters can these designs guarantee stability.
- Adaptive control is superior to robust control in dealing with uncertainties in constant or slowly-varying parameters. The basic reason lies in the learning behavior of adaptive control systems. An adaptive controller improves its performance as adaptation goes on, while a robust controller simply attempts to keep consistent performance. Another reason is that an adaptive controller requires little or no a priori information about the unknown parameters, while a robust controller usually requires a priori estimates of the parameter bounds.
- Conversely, robust control has some desirable features that adaptive control does not have, such as its ability to deal with disturbances, quickly varying parameters, and unmodeled dynamics.
- Adaptive control for nonlinear systems requires linear parametrization of the system dynamics. Full parametrization and thus adaptive control cannot be achieved in case of parametric uncertainty.

### 6.5. Summary

In this chapter, a methodology is developed to combine the design of a reconfigurable robot with control design methods as shown in Figure 6.3. The purpose of this combination is to extend the capability and properties of a reconfigurable robot to perform any required task. In comparison to predefined fixed kinematic structure robots, the D–H design parameters are modeled to be variable, which are the main constraints to improve any robot abilities. Using a reconfigurable robot will go beyond the control design constraints in performance improvement by changing the kinematic structure of the robot instantaneously as needed when it is needed. The methodology has been divided into the configuration phase and control phase.

In the configuration phase, a task defined by its shape and dimensions is partitioned into trajectory segments. To perform a segment, the algorithm selects a kinematic structure and test it for the reachability and Jacobian singularities.

In case of these two conditions are not satisfied, the algorithm run three internal loops. These loops can change the parameters; link offset, link twist angle and joint angle; within their range limits. Kinematic models were calculated for each parameter value and tested against the two mentioned singularity conditions. Then, equations of motion are generated automatically to represent the selected configuration.

In the control phase, a specific control design method is selected based on the parameter types.  $H_\infty/\mu$  controllers are selected for immeasurable parameters with linearized system. A Linear Parameter Varying (LPV) parameter is selected when the parameters are measured and can be fed back to the controller. A Computed Torque Control (CTC) is designed for known parameter that canceled by the control parameters. These parameter cancellation by the feedback linearization control represents the inner loop control. At the outer loop, PD or PID linear controllers can be designed to achieve high tracking performance. A Variable Structure Control (VSC) is designed for systems with unknown and uncertain parameters. The structure of this control includes a feedback linearization control and a robust element to drive the system state towards the sliding surface achieving the global stability. An adaptive control can be applied to systems with constant or slowly varying parameters to obtain high tracking performance.

## Conclusions and Recommendations

### 7.1. Conclusions

A reconfigurable robot has been investigated with features such as variable link twist angles, length links and (translational/rotational) joint types. The kinematic design parameters, i.e., the D–H parameters, were modeled to be variable and can generate any required kinematic structure to facilitate a specific application. The joint angle and the offset distance of the D–H parameters were also modeled as variable parameters (reconfigurable joint). The resulting reconfigurable robot hence encompasses different kinematic structures and has a reconfigurable joint to accommodate any required application in medical technology, space exploration and future manufacturing systems, among others. The reconfigurability problem was proven to be feasible and solved using model based control strategies.

First, a global kinematic model has been developed to automatically generate any kinematic configuration of three links and joints robot manipulator. In the next step, the recursive Euler-Newton algorithm was extended to automatically generate each element of the inertia matrix, Coriolis torque matrix, centrifugal torque matrix, and the gravity torque vector. Then, the following advanced model based control strategies were employed as follows:

- **PD-Gravity control.**

The practical significance of this control law lies in the fact that it requires no detailed knowledge of the manipulator parameters: inertia tensor matrix  $M(q)$ , the Coriolis and centrifugal coupling matrix  $C(q, \dot{q})$ , or the friction vector  $F(\dot{q})$ . It does require knowledge of the gravity loading vector  $G(q)$ , but this is relatively easy to determine in comparison to other elements of the dynamic equations of motion. Using Lyapunov's second method, a nonlinear control with gravity compensation was proved to satisfy the closed loop stability and performance requirements.

- **$H_\infty/\mu$  optimal controllers.**

With a reconfigurable robot, the dynamic parameters (the inertia, Coriolis, centrifugal and gravity) components were uncertain due to their dependency on the robot configuration. The parameter variations were modeled as bounded parametric uncertainties and an  $H_\infty$ -optimal control was designed to achieve the performance specifications in the presence of these uncertainties. The resulting closed loop system was analyzed using the singular structured value ( $\mu$ ) approach.

- **Sliding mode controller (SMC) controller.**

Sliding mode controllers were designed for a 3-DOF reconfigurable robot. The developed SMC controllers were based on estimated, bounded models and on the computed torque method. The resulting closed loop system was completely insensitive to the parametric uncertainties and external disturbances during sliding mode.

- **Adaptive control.**

The nonlinear dynamic equations of a 3-DOF reconfigurable robot were linearized using the regressor function property in which the dynamics are separated into a known matrix of time functions and an unknown constant parameter vector. This separation of the robot dynamics was used in the formulation of an adaptive update rule and also in the stability analysis of the tracking error system.

- **Gain scheduled or linear parameter varying (LPV) controller.**

In this control approach, the link inertia and joint viscous parameters were considered to be time varying and on-line measured parameters. Here the state-space matrices of the feedback linearized system are assumed to depend affinely on the two time varying parameters. These parameters are fed to the controller to optimize the performance and robustness requirements of the closed loop system. The resulting closed loop system was shown to perfectly fulfill the performance specifications along the parameter trajectories. The LPV control scheme employed here suffers from conservatism in the face of slowly varying parameters, since the employed

quadratic Lyapunov functions allow for arbitrarily fast parameter variations.

An algorithm was developed to select a relevant kinematic structural robot configuration for any predefined geometric task. Assuming a spherical wrist attached to the end effector, the kinematic structures of common industrial robots are determined by the first three links and joints, which also define the external and internal workspace boundaries. A spherical wrist, which satisfies Piper's condition only serves to orient the end effector within the workspace. The resulting reconfigurable robot includes eight kinematic structures: Cartesian TTT, Cylindrical RTT, TTR, TRT, Articulated RRR, Spherical/Scara RRT, TRR, and RTR. The algorithm starts by defining the path planning (linear/curvature) and testing the workspace singularity and manipulability index of the selected kinematic structure. In case of a singular configuration, the algorithm will select the next kinematic structure and test its workspace singularity. The structural kinematic selection continues until it arrives at a structure that satisfies the reachability condition without singularity. Then, equations of motion are automatically generated with elements that describe the involved dynamics of the selected structure. If the equations of motion are linear, a linear control, such as PD or PID, is selected to satisfy the performance requirements. In case the equations of motion are nonlinear, then a control approach is selected based on the parameter types as follows:

- Variable and immeasurable parameters: the nonlinear equations of motion have to be linearized and the variation would be modeled as bounded uncertainties within known limits. An LFT (Linear Fractional Transformation) control problem is composed and a  $H_\infty$ -optimal control is designed to satisfy the maximum norm of a stacked performance cost function. If the resulting control design is conservative and the uncertainties can be modeled as structured uncertainties, an  $\mu$ -optimal control can be designed, which allows more sizable parameter variations and improves the robust performance.
- Variable and measurable parameters: if the parameters are measurable, then for a linearized system, a linear parameter varying (LPV) control can

be designed, which results in a high tracking performance in the presence of disturbances.

- Known parameters of the nonlinear equations of motion: a feedback linearization with PD control can be designed to satisfy the performance requirements.
- Unknown and uncertain parameters: a sliding mode control (feedback linearization control plus robust component) can be designed to achieve stability and high tracking performance in the presence of parametric and unmodeled dynamic uncertainties.
- Slowly time varying parameters: an adaptive control with adaptation rule can be designed to achieve stability and satisfy the tracking performance requirements.

The predefined trajectory of the given task can be partitioned into segments according to their geometry (linear, circular, etc.). The algorithm performs the selection (as per the criteria above) of the applicable kinematic configuration and control method for each segment sequentially, returning to the path planning node after completion of each segment to start the process for the next one.

## 7.2. Recommendations

This research systematically went through a full design process for a reconfigurable robot by applying the most effective control approach for a kinematic structure. Yet, it still leaves many points of research open for more thorough investigation, some at the abstract level of modeling and simulation, and some at the lower level of detail in practice.

- Some applications require not only reaching a point in 3D space, but also with the desired orientation at that point. Therefore, wrist singularities should be investigated for a reconfigurable robot.
- Determination of the optimal reconfigurable configuration to perform a task using the maximum manipulability index.

- Integrating CAD drawing information in a manner that allows the top level control system to specify a geometric task and would turning a reconfigurable robot into a self-reconfigurable one.
- Develop and design a reconfigurable joint, an electro-mechanical design concept that can be perform dual rotational/linear motions.
- Dynamics, not taken into account by the model based controller and still occurring in practice should be considered in control design strategies. The most important part is the flexible deformations of the links and the friction (static/dynamic) components. Including these elements would lead to improve the tracking performance of the robot.



## Appendix A: Bosch Scara Model

This Appendix includes the simulation results calculated and explained in Chapter 3: inertia tensor matrix, Coriolis matrix, joint torques vector and payload matrix.

The inertia matrix:

$M =$

2.9074	-0.4458	0.0574	-0.1294	0.0000	0.0000
-0.4458	4.1243	0.8523	-1.2755	0.0446	-0.0000
0.0574	0.8523	1.8197	-2.0192	0.0805	-0.0000
-0.1294	-1.2755	-2.0192	3.8508	-0.1768	0.0000
0.0000	0.0446	0.0805	-0.1768	0.1963	0.0000
0.0000	-0.0000	-0.0000	0.0000	0.0000	0.1941

The Coriolis matrix:

$C =$

0.0000	-0.9505	-0.2611	0.4278	0.0062	-0.0000
0.3601	-0.0000	-0.0355	-0.1612	0.0351	0.0000
0.1688	0.0178	0.0000	-0.1612	0.0530	0.0000
-0.2786	0.0806	0.1612	0.0000	-0.0884	0.0000
-0.0000	-0.0351	-0.1061	0.1768	0.0000	-0.0000
0.0000	-0.0000	-0.0000	0.0000	0.0000	0.0000

The joint torques vector is:

Torque =

-0.2582

0.1170

0.0392

-0.1432

0.0708

-0.0000

The payload matrix:

Ratio =

1.0290	0.9295	9.0895	2.9280	-	805.6685	1.0000
0.9295	1.0689	1.9923	2.5822		268.7898	1.0000
9.0895	1.9923	1.6006	2.6996		138.8876	1.0000
2.9280	2.5822	2.6996	2.6727		86.8056	1.0000
-161.1389	268.7898	138.8876	86.8056		1.1454	1.0000
1.0000	1.0000	1.0000	1.0000		1.0000	1.0000

**Derivation of General Inverse (Pseudo Inverse) of the Jacobian Matrix:**

$$\begin{aligned}
 \|\dot{q}\|^2 &= \dot{q}^T \dot{q} \\
 &= (J^+ v)^T (J^+ v) \\
 &= [J^T (J J^T)^{-1} v]^T [J^T (J J^T)^{-1} v] \\
 &= v^T [(J J^T)^{-1}]^T J J^T (J J^T)^{-1} v \\
 &= v^T [(J J^T)^{-1}]^T v \\
 &= v^T [(J J^T)^T]^{-1} v \\
 &= v^T (J J^T)^{-1} v
 \end{aligned}$$

where  $A^+$  is an  $m \times n$  and computed as follows:

$$A^+ = \begin{cases} A^T (A A^T)^{-1} & m \leq n \\ A^{-1} & m = n \\ (A^T A)^{-1} A^T & m \geq n \end{cases}$$

## Appendix B: Bosch Scara Model

In this Appendix, the parameters of the Bosch Scara robot are given with the state space representation of the linearized model. Table A.1 includes the parameter values of the first two links and joints. The parameters of the first two DC motors are given in Table A.2.

TABLE A.1. Nominal parameter values of the Bosch Scara robot arm

Servo/ Link 1	Value	Units	Servo/ Link 2	Value	Units
$F_{v_1}$	0.001	Nms/rad	$F_{v_2}$	0.005	Nms/rad
$F_{v_L}$	90	Nms/rad	$K_{m_2}$	0.099	Nm/A
$K_{m_1}$	0.24	Nm/A	$R_{m_2}$	1.87	$\Omega$
$R_{m_1}$	1.45	$\Omega$	$J_{m_2}$	$4.2 \cdot 10^{-4}$	$\text{kgm}^2$
$J_{m_1}$	$1.075 \cdot 10^{-2}$	$\text{kgm}^2$	$L_{m_2}$	$\leq 1.0 \cdot 10^{-4}$	$H$
$L_{m_1}$	$\leq 1.5 \cdot 10^{-3}$	$H$	$N_2$	$101^{-1}$	[1]
$N_1$	$129^{-1}$	[1]	$L_2$	0.355	m
$L_1$	0.445	m			

TABLE A.2. Nominal parameter values of the Bosch Scara robot arm

Amplifier 1	Value	Units	Amplifier 1	Value	Units
$K_{P_{11}}$	27.9	[1]	$K_{P_{21}}$	28	[1]
$K_{I_{11}}$	646.5	$\text{sec}^{-1}$	$K_{I_{21}}$	413.3	$\text{sec}^{-1}$
$\tau_1$	$8 \cdot 10^{-5}$	$\text{sec}^{-1}$	$\tau_2$	$8 \cdot 10^{-5}$	$\text{sec}^{-1}$
$K_{P_{12}}$	1.42	V/A	$K_{P_{22}}$	2.67	V/A
$K_{I_{12}}$	227	V/As	$K_{I_{22}}$	280	V/As

TABLE A.3. Different values for inertias and damping for different joint position of link 2 (left). Estimated parameter values for stiction and Coulomb friction (right).

Parameter	$\theta_2 = 0$	$\theta_2 = \frac{1}{2}\pi$	Units	Torque	joint 1	joint 2
$M = J_{L_1}$	10.9	7.46	$\text{kgm}^2$	$F = F_s [N m]$	17.92	11
$J_c$	$8.2 \cdot 10^4$	$7.6 \cdot 10^4$	Nm/rad	$F_c [N m]$	14.1	9

$$\begin{aligned}
 \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -\frac{F_{VL}}{J_{L1}} & \frac{K_s}{J_{L1}} & -\frac{D_s N_1}{J_{L1}} & 0 & 0 \\ 0 & -1 & 0 & N_1 & 0 & 0 \\ 0 & 0 & -\frac{N_1 K_s}{J_{m1}} & -\frac{F_{V1}}{J_{m1}} & \frac{K_{m1}}{J_{m1}} & 0 \\ 0 & 0 & 0 & -\frac{K_{m1}}{L_{m1}} & \frac{-R_{m1} - K_{p12}}{L_{m1}} & 1 \\ 0 & 1 & 0 & 0 & -K_{i12} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ K_{i12} \end{bmatrix} u + \begin{bmatrix} 0 \\ \frac{1}{J_{L1}} \\ 0 \\ 0 \\ 0 \\ K_{i12} \end{bmatrix} \tau_{LD} \\
 y &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + [0]D \tag{231}
 \end{aligned}$$

The notations of state space parameters are given in the Nomenclature.

## Appendix C: Kinematic Calculations

The linear parameter matrix for a 3-DOF kinematic structure is calculated as follows:

$$Y(q, \dot{q}, \ddot{q}) = \begin{bmatrix} \ddot{q}_1 & \cos(q_2)(2\ddot{q}_1 + \ddot{q}_2) - \sin(q_2)(2\dot{q}_1\dot{q}_2 + \dot{q}_2^2) & \ddot{q}_2 & g \cos(q_1) & g \cos(q_1 + q_2) \\ 0 & \cos(q_2)\ddot{q}_1 + \sin(q_2)\dot{q}_1^2 & \ddot{q}_1 + \ddot{q}_2 & 0 & g \cos(q_1 + q_2) \end{bmatrix} \quad (232)$$

And the parameter vector is given as follows:

$$\varphi = \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \\ \varphi_4 \\ \varphi_5 \end{bmatrix} = \begin{bmatrix} m_1 a_{c1}^2 + m_2(a_1^2 + a_{c2}^2) + I_1 + I_2 \\ m_2 a_1 a_{c2} \\ m_2 a_{c2}^2 + I_2 \\ m_1 a_{c1} + m_2 a_1 \\ m_2 a_{c2} \end{bmatrix}$$

The linear parameter matrix for a 2-DOF kinematic structure is calculated as follows:

$$Y(q, \dot{q}, \ddot{q}) = \begin{bmatrix} \ddot{q}_1 & \cos(q_2)(\ddot{q}_1 - 2\ddot{q}_2) - \sin(q_2)(2\dot{q}_1\dot{q}_2 - \dot{q}_2^2) & \ddot{q}_1 + \ddot{q}_2 & 0 \\ 0 & -2\cos(q_2)(\ddot{q}_1 + 2\sin(q_2)\dot{q}_1^2) & \ddot{q}_1 + \ddot{q}_2 & 0 \\ 0 & 0 & 0 & \ddot{d}_3 \end{bmatrix} \quad (233)$$

And the parameter vector is given as follows:

$$\varphi = \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \\ \varphi_4 \end{bmatrix} = \begin{bmatrix} (\frac{m_2}{3} + m_2 + m_3)a_1^2 \\ (m_2 + 2m_3)a_1 a_2 \\ (\frac{m_2}{3} + m_3)a_2^2 \\ m_3 \end{bmatrix}$$

## References

- [1] C. Abdallah, D. Dawson, P. Dorato, and M. Jamshidi, *Survey of robust control for rigid robots*, IEEE Control Systems Magazine, **11** (1991), 24-29.
- [2] C. Abdallah, D. Dawson, P. Dorato, and M. Jamshidi, *Survey of robust control of robots*, Proceedings of American Control Conference, (1990), 718-721.
- [3] K. Abdel-Malik, H. J. Yeh, *Analytical Boundary of the Workspace for General 3-DOF Mechanism*, The International Journal of Robotics Research, **16** (1997), 198-213.
- [4] R. Abraham and J.E. Marsden, *Foundation of Mechanics (2nd edition)*, The Benjamin Cummings Publishing Company, 1978.
- [5] F. Aghili and K. Parsa, *A Reconfigurable Robot with Lockable Cylindrical Joints*, IEEE Transactions on Robotics **25** (2009), 785-797.
- [6] P. Apkarian, P. Gahinet, *A convex characterization of gain-scheduled  $H_\infty$  controllers*, IEEE Trans. Automat. Control. (1995), 853-864.
- [7] R. Al Saidi and B. Minaker, *Analysis of Generic kinematic for self-reconfigurable open kinematic machinery systems*, Proc. 24<sup>th</sup> Canadian Congress of Applied Mechanics, (2013).
- [8] R. Al Saidi and B. Minaker, *Analysis and development of self-reconfigurable open kinematic machinery systems*, Proc. IEEE International Conference on Automation Science and Engineering, (2013), 966-971.
- [9] R. Al Saidi and B. Minaker, *Development and control of self-reconfigurable open kinematic chain robot arms*, Proc. IEEE International Symposium on Robotic and Sensors Environments, (2013).
- [10] R. Al Saidi and B. Minaker, *An adaptive sliding mode control for trajectory tracking of a self-reconfigurable robotic system*, Robot Intelligence Technology and Applications 2: Results from the 2<sup>nd</sup> International Conference on Robotic Intelligence Technology and Applications, Ed. J-H Kim, E. T. Matson, H. Myung, P. Xu and F. Karray, Springer, (2013), 381-391.
- [11] R. Al Saidi and B. Minaker, *Kinematic properties and control for a reconfigurable robotic system*, Proc. IEEE International Conference on Systems, Man and Cybernetics, (2014), 2827-2832.
- [12] R. Al Saidi and B. Minaker, *Analysis and control of a reconfigurable robotic arm*, J. Autonomous Robots, (2014), Submitted.
- [13] R. O. Ambrose, *Interactive Robot Joint Design, Analysis and Prototyping*, IEEE International Conference on Robotics and Automation, (1995), 2119-2124.

- [14] R. J. Anderson and J.B. Moore, *Passive computed torque algorithms for robots*, Proceedings of the IEEE Conference on Decision and Control, (1989), 1638-1644.
- [15] J. Angeles, *Fundamentals of Robotic Mechanical Systems: Theory, Methods, and Algorithms*, 3rd ed. New York: Springer-Verlag, 2007.
- [16] J. Angeles, *A methodology for the optimum dimensioning of robotic manipulators*, Univ. Autonoma de San Luis Potos, San Luis Potos, Mexico, Memoria del 5o. Congreso Mexicano de Robotica, (2003), 190203.
- [17] S. Arimoto. *Control theory of non-linear mechanical systems: a passivity-based circuit theoretic approach*, Clarendon-Press, Oxford, 1996.
- [18] B. Benhabib, G. Zak, and M. G. Lipton, *A Generalized Kinematic Modeling Method for Modular Robots*, J. Robotics Systems **6** (1989), 545-571.
- [19] H. Berghuis, *Model based robot control: from theory to practice*, PhD thesis, University of Twente, The Netherlands, 1993.
- [20] Z. M. Bi, W. A. Gruver and S. Y. T. Lang, *Analysis and Synthesis of Reconfigurable Robotic Systems*, Robotics and Computer-Integrated Manufacturing **12** (2010), 553-562.
- [21] Z. M. Bi, W. A. Gruver, W.J. Zhang, A.Y.T. Lang, *Automated modeling of modular robotic configurations*, Robotics and Autonomous Systems **54** (2006), 1015-1025.
- [22] Z. M. Bi, W. J. Zhang, I-M. Chen and S. Y. T. Lang, *Automated Generation of the D-H Parameters for Configuration Design of Modular Manipulators*, Robotics and Computer-Integrated Manufacturing **23** (2007), 553-562.
- [23] S.P. Boyd and C.H. Barrat, *Linear controller design; limits of performance*, Prentice-Hall, 1991.
- [24] C. Canudas de Wit, B. Siciliano, and G. Bastin. *Theory of Robot Control*, Springer, 1996.
- [25] M. Ceccarelli, *A Formulation for the Workspace Boundary of General n-Revolute Manipulators*, Mechanism and Machine Theory **31** (1996), 637-646.
- [26] M. Ceccarelli and A., Vinciguerra, *On the Workspace of 4R manipulators*, The International Journal of Robotics Research **14** (1995) 152-160.
- [27] I-M. Chen, *Theory and Applications of Modular Reconfigurable Robotic Systems*, PhD thesis, California Institute of Technology, USA, 1994.
- [28] I-M. Chen and J. W. Brudick, *Rapid response manufacturing through a rapidly reconfigurable robotic work cell*, Robotics and Computer-Integrated Manufacturing **3** (2001), 199-213.
- [29] I-M. Chen and J. W. Burdick, *Enumerating the Non-Isomorphic Assembly Configurations of Modular Robotic Systems*, Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), (1993), 1985-1992.
- [30] B. S. Chen, T. S. Lee, and J. H. Feng, *A nonlinear  $H_\infty$  control design in robotic systems under parameter perturbation and external disturbance*, International Journal of Control **59** (1994), 439-461.

- [31] I-M. Chen and G. Yang, *Configuration Independent Kinematics for Modular Robots*, IEEE International Conference on Robotics and Automation, (1996) 1440-1445.
- [32] R. Cohen, M. G. Lipton, M. Q. Dai and B. Benhabib, *Conceptual Design of a Modular Robot*, ASME Journal of Mechanical Design **114**, (1992), 117-125.
- [33] P. Corke and B. Armstrong-Helouvy, *A search for consensus among model parameters reported for PUMA 560*, Proc. IEEE Int. Conf. Robotics and Automation, (1994), 1608-1613.
- [34] Corke. P, *Robotic toolbox*, (2002).
- [35] J. Denavit, R.S. Hartenberg, *A kinematic notation for lower-pair mechanisms based on matrices*, Journal of Applied Mechanics, (1955) 215-221.
- [36] A. Djuric, *Reconfigurable kinematics, dynamics and control process for industrial robots*, PhD dissertation, University of Windsor, (2007)
- [37] A. Djuric, R. Al Saidi and W. ElMaraghy, *Global Kinematic Model Generation of n-DOF Reconfigurable Machinery Structure*, Proc. IEEE International Conference on Automation Science and Engineering, (2010), 804-809.
- [38] A. Djuric, R. Al Saidi and W. ElMaraghy, *Dynamics solution of n-DOF global machinery model*, Robotics and Computer-Integrated Manufacturing **28**, (2012), 621-630.
- [39] H. A. ElMaraghy, *Changeable and Reconfigurable Manufacturing systems*, Springer Verlag., New York, 2009.
- [40] B. Friedland, *Control System Design*, MacGraw Hill, 1986.
- [41] K. S. FU, R. C. Gonzalez, C. S. G. Lee, *Robotics Control, Sensing, Vision, and Intelligence*, MacGraw Hill, 1987.
- [42] T. Fukuda and S. Nakagawa, *Dynamically Reconfigurable Robotic System*, IEEE International Conference on Robotics and Automation, (1988), 1581-1586.
- [43] P. Gahinet, P. Apkarian, *Self-scheduled  $H_\infty$  control of linear parameter varying systems*, Proc. Amer. Control. Conference. (1994), 856-860.
- [44] M. T. Grabbe and D. M. Dawson, *An application of optimal control theory to the trajectory of rigid robot manipulator*, Optimal Control Application & Methods **15** (1994), 237-249.
- [45] J. Graig, *Introduction to Robotics, Mechanics and Control*, Addison-Wesley, 1989.
- [46] J. Graig, *Adaptive Control of Mechanical Manipulators*, Englewood Cliffs, NJ: Prentice Hall, 1989.
- [47] K. C. Gupta and B. Roth, *Design consideration for manipulator workspace*, ASME J. Mech. Design, **104**, (1982), 704-712.
- [48] M. Hedelind and M. Jackson, *The need for Reconfigurable Robotic Systems*, Proc. Of the 2nd Int. Conf. on Changeable, Agile, Reconfigurable and Virtual Production, CARV (2007).
- [49] A. Isidori, *Nonlinear Control Systems: An Introduction, Lecture notes in Control and Information Science*, **72**, (1985) Springer Verlag.



- [50] Z. Jing and J. Yi, *Dimensional synthesis based on fault tolerant performance for redundant robots*, Proceedings 13th Int. Conf. Adv. Robot., (2007), 10231028.
- [51] H. K. Khalil, *Nonlinear systems, 3rd Edition*, Prentice Hall, N.J., 2002.
- [52] Koren Y., Heisel U., Jovane F., Moriwaki T., Pritschow G., Ulsoy G., van Brussel H., *Reconfigurable Manufacturing Systems*. Annals of the CIRP, **48/2**, (1999), 527540.
- [53] K. Kreutz. *On manipulator control by exact linearization*, IEEE Transactions on Automatic Control **34** (1989), 763767.
- [54] M. Krstic, I. Kanellakopoulos, and P. Kokotovic. *Nonlinear and Adaptive Control Design*, Wiley, 1995.
- [55] A. Kumar and K. J. Waldron, *The workspace of a mechanical manipulator*, ASME J. Mech. Design **103**, (1981), 665672.
- [56] S. J. Kwon, W. K. Chung, Y. Youm, and M. S. Kim, *Self-collision avoidance for n-link redundant manipulators*, Proceedings of the IEEE Systems, Man, Cybern., Decision Aiding Complex System., Charlottesville, VA, Oct. (1991), 937942.
- [57] F. Lewis, C. Abdallah, and D. Dawson. *Control of Robot Manipulators*, Macmillan, 1993.
- [58] Z. Li, W. Melek, and C. M. Clark, *Development and Characterization of a Modular and Reconfigurable Robot*, Proceedings of the International Conference on Changeable, Agile, Reconfigurable and Virtual Production, CARV (2007).
- [59] G. Liu, *Development of modular and reconfigurable with multiple working modes* Proceedings of IEEE Int. Conf. Robotics and Automation, (2008).
- [60] J. M. Maciejowski, *Multivariable Feedback Design*, Addison-Wesley, 1989.
- [61] J.-P. Merlet, *Parallel Robots*, 2nd ed. New York: Springer-Verlag, 2006.
- [62] H. Nijmeijer and A. J. van der Schaft, *Nonlinear Dynamical Control Systems*, (1990) Springer Verlag.
- [63] R. Manseur, *Robot Modeling and Kinematics*, De Vinci Engineering Press, 2006.
- [64] R. Manseur and K. L. Doty, *Structural Kinematics of 6-Revolute-Axis Robot Manipulators*. Mechanism and Machine Theory **31** (1996), 647-657.
- [65] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*, CRC, 1994.
- [66] R. Ortega and R. Spong, *Adaptive motion control of rigid robots*, Proceedings of the IEEE Conference on Decision and Control, (1988), 1575-1584.
- [67] C. J. J. Paredis, H. B. Brown, R. W. Casciola, J. E. Moody and P. K. Khosla, *A Rapidly Deployable Manipulator System*, International Workshop on Some Critical Issues in Robotics, (1995), 175-185.
- [68] Z. Qu, *Global stability of trajectory tracking of robot under PD control*, Proceedings of IEEE Conference of Control Applications, (1992), 8-12.

- [69] Z. Qu, *Global stabilization of nonlinear systems with a class of unmatched uncertainties*, Proceedings of IEEE Conference of Control Applications, (1992), 498-503.
- [70] Z. Qu and D. M. Dawson. *Robust Tracking Control of Robot Manipulators*. IEEE-Press, 1996.
- [71] Z. Qu and J. Dorsey, *Robust tracking control of robots by a linear feedback law*, IEEE Transactions on Automatic Control **36** (1992), 1081-1084.
- [72] M. C. Readman, *Flexible Joint Robots*, CRC, 1994.
- [73] Theodorus A, *Robot Trajectory Planning*, PhD thesis, Delft University of Technology, 1992.
- [74] R. J. Schilling, *Fundamentals of robotics*, Prentice Hall, 1990
- [75] S. Skogestad, I. Postlethwait, *Multivariable Feedback Control, Analysis and Control*, John Wiley & Sons, 1997.
- [76] J. E. Slotine and W. Li, *Applied Nonlinear Control*, Prentice Hall, N.J., 1991
- [77] J. J. E. Slotine and J. J. Craig, *Adaptive trajectory control of manipulators*, The Robotics Review **1** (1989), 369-377.
- [78] Y. D. Song, A. T. Alouani and J. N. Anderson, *Robust path tracking control of industrial robots: an  $H_\infty$  approach*, Proceedings of the IEEE Conference on Control Applications, (1992), 25-30.
- [79] M. W. Spong, *Modeling and control of elastic joint robots*, ASME Journal on Dyn., Sys., Measurement and Control **109** (1987), 310319.
- [80] M. W. Spong, S. Hutchinson, M. Vidyasagar, *Robot Modeling and Control*, John Wiley & Sons, 2006.
- [81] M. Spong and M. Vidyasagar. *Robot Dynamics and Control*, Wiley, 1989.
- [82] M. Spong and M. Vidyasagar, *Robust linear compensator design for nonlinear robotic control*, IEEE Transactions on Robotics and Automation, (1985), 954-959.
- [83] W. L. Stout and M. E. Swan, *Application of  $H_\infty$  theory to robot manipulator control*, Proceedings of the IEEE Conference on Control Applications, (1992) 148-153.
- [84] T. Strasser, M. N. Rooker and G. Ebenhofer, *Distributed Control Concept for a 6-DOF Reconfigurable Robot Arm*, Innovation production machines and systems: Fourth I\*Proms Virtual International Conference, (2008).
- [85] S. Tabandeh, C. Clark, and W. Melek, *Task-based Configuration Optimization of Modular and Reconfigurable Robots using a Multi-Solution Inverse Kinematics Solver*, International Conference on Changeable, Agile, Reconfigurable and Virtual Production. CARV (2007).
- [86] V. I. Utkin, *Sliding modes and their application in variable structure systems (English translation)*, MIR publications, 1978.
- [87] K. H. Wurst, *The Conception and Construction of a Modular Robot System*, 16th Int. Sym. Industrial Robotics (ISIR), (1986) 37-44.
- [88] K. Zhou, J. Doyle, and K. Glover, *Robust and Optimal Control*, Prentice-Hall, 1996.

- [89] T. Yoshikawa, *Manipulability of robotic mechanisms*, The Journal of Robotics Research, **4**, (1985), 39.

## **Vita Auctoris**

Riyadh Al Saidi was born in Al-Shamiya, Iraq in 1962. He completed a Bachelors of Engineering at the University of Mosul in 1984 and earned a Masters in Mechanical Engineering from the prestigious Delft University of Technology in the Netherlands in 1999. In addition to his academic contributions, Riyadh has been very active in industry and has practiced engineering design and applied work in a variety of settings.