

INTELLIGENT MODEL PREDICTIVE CONTROL OF FLEXIBLE LINK
ROBOTIC MANIPULATORS

by

TAO FAN

B. Eng., Xian Jiaotong University, 1991

M.A.Sc., The University of British Columbia, 2003

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE STUDIES

(Mechanical Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

May 2007

© Tao Fan, 2007

Abstract

This thesis develops and evaluates an intelligent model predictive control (IMPC) strategy for motion control of a flexible link robotic manipulator through analysis, computer simulation, and physical experimentation.

The developed IMPC is based on a two-level hierarchical control architecture. This control structure is used to combine the advantages of the conventional model predictive control (MPC) with those of knowledge-based soft control techniques. The upper level of the structure is a fuzzy-rule based intelligent decision-making system. The lower level consists of two modules: a real-time system identification module (which adjusts the model parameters and accommodates payload changes of the manipulator), and a model predictive control (MPC) module (which develops control inputs based on the linear model generated by the system identification module). The upper-level intelligent fuzzy rule-based tuner interacts with the lower level modules. Based on the desired system performance, the state feedback signals, and the knowledge base, the upper-level fuzzy tuner automatically adjusts the tuning parameters of the MPC controller. It is also able to adjust the model structure of the system-identification module, if necessary, to accommodate large model errors, and will increase the robustness of the controller.

An explicit, complete, and accurate nonlinear dynamic model of the system is developed using the assumed mode method. More realistic boundary conditions, which represent the balance of moments and shear forces separately, at the ends of each link, are used for the dynamic model development of the system. A computationally efficient multi-stage MPC algorithm with guaranteed stability is developed as well. This algorithm is used by the MPC module to enable real-time implementation of the overall scheme. A fuzzy knowledge base for tuning the MPC controller is developed based on analysis, computer simulations and experimental testing of the prototype flexible-link manipulator system (FLMS). A fuzzy tuner is designed based on this fuzzy knowledge base. The performance of the developed IMPC scheme is evaluated using computer simulations and experiments of the prototype FLMS. The results show that IMPC can more effectively control the motion of a flexible link robot manipulator when compared with conventional MPC.

Table of Contents

Abstract	ii
Table of Contents	iii
List of Tables	vi
List of Figures	viii
Nomenclature	xiii
Acknowledgments	xviii
Chapter 1 Introduction	1
1.1 Goals of the Research	1
1.2 Motivation	3
1.3 Related Work	7
1.3.1 Model Predictive Control	7
1.3.2 Controller Tuning	9
1.3.3 Control of Flexible Link Robotic Manipulators	10
1.4 Organization of the Thesis	14
Chapter 2 Manipulator Model	16
2.1 Kinematics	17
2.2 Flexible Link Model	21
2.2.1 Flexible Beam Equation	21
2.2.2 Boundary Conditions	22
2.2.3 Solution of the Flexible Beam Equation	24
2.2.4 The Assumed Mode Method	28
2.3 Derivation of Equations of Motion	37
2.3.1 Closed-form n -link Model	37
2.3.2 Single-link Model	41
2.3.3 Two-link Model	46
2.4 Linearization of the Equations of Motion	51

2.5	Non-minimum Phase Characteristic	54
2.6	Summary	62
Chapter 3	Intelligent Model Predictive Control	64
3.1	The Overall Structure of IMPC	64
3.2	MPC Controller Design	65
3.2.1	Basic Principles of MPC	66
3.2.2	State-space Model MPC Formulation	69
3.2.3	Prediction	71
3.2.4	MPC Computation and Constraint Handling	77
3.2.5	Stability Analysis	83
3.3	System Identification for Unknown Payload	88
3.4	Fuzzy Rule-Based Auto-Tuning	90
3.4.1	Effects of the Tuning Parameters	91
3.4.2	Fuzzy Inference System	96
3.4.3	Overall Scheme of the Fuzzy Tuner	104
3.5	Summary	108
Chapter 4	Control of a Single-link Flexible Manipulator	109
4.1	Dynamic Model	109
4.1.1	Flexible Modes	109
4.1.2	Nonlinearity	114
4.1.3	Effects of Payload	118
4.2	MPC Controller	121
4.3	Payload Identification	127
4.4	Fuzzy Tuner	131
4.5	Computer Simulations of IMPC	133
4.6	Summary	139
Chapter 5	Control of a Two-link Flexible Manipulator	141
5.1	Internal Model	141
5.2	IMPC Controller Design	142
5.3	Computer Simulations of IMPC	144
5.4	Summary	154

Chapter 6 Experiments with Flexible-link Manipulator	155
6.1 The Experimental Test Bed	155
6.1.1 Actuation Subsystem	158
6.1.2 Measurement Subsystem	160
6.1.3 Data Acquisition Subsystem	163
6.1.4 Software Design	167
6.1.5 Operations of the Overall System	175
6.2 Experimental Results	175
6.3 Summary	180
Chapter 7 Conclusion	181
7.1 Primary Contributions	181
7.2 Recommendations for Future Research	182
Bibliography	184
Appendix A Dynamic Models	192
A.1 Single-link Flexible Manipulator Model	192
A.1.1 Nonlinear Model	192
A.1.2 Continuous-time State-space Model	193
A.1.3 Discrete-time State-space Model	194
A.2 Two-link Flexible Manipulator Model	195
Appendix B Assembling of the Prototype FLMS	203
B.1 Mechanical Assembling of the FLMS.	203
B.2 Electrical Wiring of the FLMS	205
Appendix C Installation Guide for Motion Control Interface Card Hardware	206
C.1 Jumper and Connector Locations	206
C.2 Connections	207

List of Tables

2.1	Parameters of the links	32
2.2	First four modes of link 1.	33
2.3	First four modes of link 2.	33
3.1	Adjustable controller parameters	96
3.2	General linguistic rules for tuning an MPC controller	101
4.1	MPC controller parameters	123
4.2	Effects of the MPC tuning parameters	132
4.3	Rule base of the FIS	132
4.4	Predetermined parameters of MPC controller for the single-link manipulator	134
4.5	Tuning parameters of MPC controller for the single-link manipulator	134
5.1	Predetermined parameters for the MPC controller of the two-link manipulator	147
5.2	Tuning parameters for the MPC controller of the two-link manipulator	148
6.1	FLMS system components	157
6.2	Specifications of the harmonic-drive motors	159
6.3	Specifications of the PWM amplifiers	160
6.4	Specifications of the DC power supply.	160
6.5	Specifications of the load cell.	161
6.6	Specifications of the force sensor signal conditioner.	162
6.7	Parameters of the ultrasonic sensor.	163
6.8	Specifications of the motion control interface card	166
6.9	Description of the input parameters of the MPC controller	171

6.10	MPC controller parameters	177
------	-------------------------------------	-----

List of Figures

1.1	Prototype flexible link manipulator system	3
2.1	Schematic diagrams of flexible link manipulators: (a) Two-link manipulator; (b) General manipulator	17
2.2	Flexible link i and the free-body diagram for an elemental segment	21
2.3	Free body diagram of the link end element.	24
2.4	Mode shape of link 1 with zero and maximum mechanical offsets	34
2.5	Mode shapes of link 1 with different payloads (zero, nominal, maximum) compared to a free-end link	35
2.6	Mode shapes of link 2 with different payloads	36
2.7	Single-link flexible robotic manipulator	41
2.8	External dynamics and internal dynamics of a nonlinear system	55
2.9	Pole-zero map of a single-link flexible robotic manipulator (aluminum rod)	61
2.10	Pole-zero map of a single-link flexible robotic manipulator (aluminum thin beam)	61
2.11	Pole-zero map of a two-link flexible robotic manipulator	62
3.1	Structure of the intelligent model predictive control (IMPC) system	65
3.2	Illustration of the strategy of SISO MPC at k th sampling instant	68
3.3	Basic structure of MPC	69
3.4	Unconstrained MPC controller structure	81
3.5	The basic idea of multi-stage MPC	88
3.6	The fuzzy inference system.	97
3.7	Membership functions for the input variables	100
3.8	Membership functions for the output variables	100
3.9	Fuzzy inference process	102

3.10	Overall scheme of the fuzzy tuner	107
4.1	Case 1: Joint angle responses of 1-mode model and 2-mode models with an initial tip deflection of 51.2mm	111
4.2	Case 1: Link tip deflection responses of 1-mode model and 2-mode models with an initial tip deflection of 51.2mm	112
4.3	Case 2: Joint angle responses of 1-mode model and 2-mode models under a bang-bang control input ($\theta_1(0) = 0, w_{tip}(0) = 0$)	112
4.4	Case 2: Link tip deflection responses of 1-mode model and 2-mode models under a bang-bang control input ($\theta_1(0) = 0, w_{tip}(0) = 0$)	113
4.5	Case 2: Bang-bang control input	113
4.6	Case 1: Joint angle responses of the linear and nonlinear models with an initial tip deflection of 51.2mm	116
4.7	Case 1: Link tip deflection responses of the linear and nonlinear models with an initial tip deflection of 51.2mm	116
4.8	Case 2: Joint angle responses of the linear and nonlinear models under a bang-bang control input ($\theta_1(0) = 0, w_{tip}(0) = 0$)	117
4.9	Case 2: Link tip deflection responses of the linear and nonlinear models under a bang-bang control input ($\theta_1(0) = 0, w_{tip}(0) = 0$)	117
4.10	Case 1: Joint angle responses of the linear model for different payloads with an initial tip deflection of 51.2mm	119
4.11	Case 1: Link tip deflection responses of the linear model for different payloads with an initial tip deflection of 51.2mm	119
4.12	Case 2: Joint angle responses of the linear model for different payloads under a bang-bang control input ($\theta_1(0) = 0, w_{tip}(0) = 0$)	120
4.13	Case 2: Link tip deflection responses of the linear model for different payloads under a bang-bang control input ($\theta_1(0) = 0, w_{tip}(0) = 0$)	120
4.14	Simulation block diagram	123
4.15	Case 1: Joint angle responses for the constrained and unconstrained MPC	124
4.16	Case 1: Link tip deflection responses for the constrained and unconstrained MPC	124

4.17	Case 1: Control inputs for the constrained and unconstrained MPC	125
4.18	Case 2: Control inputs for the constrained and unconstrained MPC with the same parameters	125
4.19	Case 2: Control inputs for the constrained and unconstrained MPC with different parameters	126
4.20	Case 2: Joint angle responses for the constrained and unconstrained MPC with different parameters	126
4.21	Case 2: Link tip deflection responses for the constrained and unconstrained MPC with different parameters	127
4.22	Case 1: Input and output responses of the single-link flexible manipulator (nominal payload)	129
4.23	Case 1: Measured output and simulated model outputs (nominal payload)	130
4.24	Case 2: Input and output responses of the single-link flexible manipulator (double the nominal payload)	130
4.25	Case 2: Measured outputs and simulated model outputs (double the nominal payload)	131
4.26	FIS input-output diagram	131
4.27	Case 1: Joint angle responses with IMPC	135
4.28	Case 1: Link tip deflection responses with IMPC	135
4.29	Case 1: Control input with IMPC	136
4.30	Case 2: Joint angle responses with IMPC	136
4.31	Case 2: Link tip deflection responses with IMPC	137
4.32	Case 2: Control input with IMPC.	137
4.33	Case 3: Joint angle responses with IMPC	138
4.34	Case 3: Link tip deflection responses with IMPC	138
4.35	Case 3: Control input with IMPC.	139
5.1	Simulation block diagram of the two-link flexible manipulator system	144
5.2	Joint angle response with three different internal models	146

5.3	Link tip deflection responses with three different internal models	146
5.4	Case 1: Joint angle responses of link 1 with IMPC	148
5.5	Case 1: Joint angle responses of link 2 with IMPC	149
5.6	Case 1: Link tip deflection responses of link 1 with IMPC	149
5.7	Case 1: Link tip deflection responses of link 2 with IMPC	150
5.8	Case 1: Control input of link 1 with IMPC	150
5.9	Case 1: Control input of link 2 with IMPC	151
5.10	Case 2: Joint angle responses for link 1 with IMPC	151
5.11	Case 2: Link tip deflection responses for link 1 and link 2 with IMPC	152
5.12	Case 2: Control inputs of link 1 and link 2 with IMPC	152
5.13	Case 3: Joint angle responses for link 1 and link 2 with IMPC	153
5.14	Case 3: Link tip deflection responses for link 1 and link 2 with IMPC	153
5.15	Case 3: Control inputs of link 1 and link 2 with IMPC	154
6.1	The prototype flexible link manipulator system (FLMS)	156
6.2	Schematic diagram of the electrical system of the manipulator	158
6.3	Pull down menu of the FLMS software package	167
6.4	Analog-to-digital conversion dialog box	168
6.5	Digital-to-analog conversion dialog box	169
6.6	Encoder count dialog box	169
6.7	Manual control dialog box	170
6.8	MPC motion control dialog box	171
6.9	Save dialog box	171
6.10	Program structure of the FLMS software package	174
6.11	Flow diagram of the FLMS dialog box	174
6.12	Case 1: Joint angle responses (experiment and simulation)	177

6.13	Case 1: Tip position responses (experiment and simulation)	178
6.14	Case 1: Control input (experiment and simulation)	178
6.15	Case 2: Joint angle responses (experiment and simulation)	179
6.16	Case 2: Tip position responses (experiment and simulation)	179
6.17	Case 2: Control input (experiment and simulation)	180
B.1	Overall CAD model of the FLMS prototype	203
B.2	Assembly drawing of joint module 1	204
B.3	Assembly drawing of joint module 2	204
B.4	Electrical wiring diagram of the prototype manipulator system	205

Nomenclature

Notations

\mathfrak{R}	set of real numbers
\mathfrak{R}^n	vector space of n -tuples over \mathfrak{R}
$\ \mathbf{x}\ $	Euclidean norm of vector $\mathbf{x} \in \mathfrak{R}^n$ (equivalent to $\ \mathbf{x}\ _2^2$ or $\sqrt{x_1^2 + \dots + x_n^2}$)
A, B, C, D	state-space representation of a system
α_i	angle of rotation of link i due to bending; $\alpha_i = \left. \frac{\partial w_i(x_i, t)}{\partial x_i} \right _{x_i=l_i} = w_i'(l_i, t)$
b	equivalent viscous damping coefficient
B	diagonal damping matrix
$c_{\alpha_{i-1}, i}$	$\cos(\alpha_{i-1}(t) + \theta_i(t))$
E_i	Young's modulus of link i
$E_i I_i$	flexural rigidity of link i
F^0	fixed world coordinate frame
F^i	rigid-body moving frame of link i , fixed at the joint of link i
\hat{F}^i	flexible body moving frame of link i , fixed at the end of link i
$G(s)$	transfer function
$\mathbf{H}_{i,j}$	homogeneous transformation matrix that transforms the coordinates of a point from frame F^j to frame F^i
\mathbf{h}	vector of Coriolis and centrifugal forces
i_{ENC}	encoder counts
i_{ADC}	ADC counts
H_p	prediction horizon

H_{p_old}	prediction horizon before tuning
H_{p_new}	prediction horizon after tuning
H_{p_ini}	initial prediction horizon
H_{p_end}	final prediction horizon generated by fuzzy tuner
H_{p_sen}	sensitivity parameter of prediction horizon
H_u	control horizon
$I_i(x_i)$	moment of inertia of the link i cross section about the Y_i axis
I_{nijk}	n th constant integral parameter of link i (j and k are mode shape function indices)
J_{Li}	equivalent moment of inertia at the end of link i
J_{0i}	mass moment of inertia of link i about joint i axis
J_p	payload inertia
K	stiffness matrix
K_{MPC}	MPC controller gain matrix
$k + i k$	value predicted for time $k + i$ based on the information available up to time k
l_i	length of link i
L	Lagrangian $L = T - V$
J	cost function
m_{Li}	equivalent mass at the end of link i
m_i	mass of link i
m_{hi}	mass of hub i
m_p	payload mass
$M_i(x_i, t)$	bending moment at x_i of link i
$(MO)_i$	contributions of the masses of payload and link $i + 1, \dots, n$. due to the mechanical offsets
M	positive-definite symmetric mass matrix

p_{cv}	count to voltage constant of ADC
p_{vlb}	voltage to lb constant of force sensor
p_{vcm}	centimeter to voltage constant of ultrasonic sensor
\mathbf{q}	generalized coordinates; $\mathbf{q} = (\theta_1, \dots, \theta_n, \delta_{11}, \dots, \delta_{1n_{m,1}}, \dots, \delta_{n1}, \dots, \delta_{nn_{m,n}})$
\mathbf{Q}	tracking error weighting matrix (output weighting matrix)
Q_{w_old}	link deflection weight before the tuning
Q_{w_new}	link deflection weight after the tuning
Q_{w_ini}	initial link deflection weight
Q_{w_sen}	sensitivity parameter of link deflection weight
Q_{yw}	overall weight for joint angle and link deflection
$\mathbf{r}_1^1(t)$	point \mathbf{r}_1 on link 1 in frame F^1
$\mathbf{r}_1^0(t)$	absolute position of \mathbf{r}_1 in frame F^0
R	total energy dissipation function
\mathbf{R}	control penalty weighting matrix (input rate weighting matrix)
R_{u_old}	input rate weight before the tuning
R_{u_new}	input rate weight after the tuning
R_{u_ini}	initial input rate weight
R_{u_end}	final tuned initial input rate weight
R_{u_sen}	sensitivity parameter of input rate weight
r	reference trajectory
r_θ	joint angle setpoint
r_w	link deflection constraint (maximum allowed link deflection)
r_u	control input constraint (maximum allowed control input)
$s_{\alpha_{i-1},i}$	$\sin(\alpha_{i-1}(t) + \theta_i(t))$
T	kinetic energy
T_s	sampling period

$\mathbf{v}_i^0(t)$	absolute velocity of \mathbf{r}_i in frame F^0
$V_i(x_i, t)$	shear force at x_i of link i
V	potential energy
U_i	elastic potential energy of link i
\mathbf{u}	vector of inputs
u_{\max}	maximum control input
$w_i(x_i, t)$	transverse deflection of link i at abscissa x_i ; $w_i(x_i, t) = \sum_{j=1}^{\infty} \phi_{ij}(x_i) \delta_{ij}(t)$
w_{\max}	absolute maximum link deflection
w_{tip}	link tip deflection
\hat{y}	predicted controlled output
y_{tip}	link tip position
$\phi_{ij}(x_i)$	mode shape function (eigenfunction) of link i
θ_i	joint angle of link i
$\dot{\theta}_i^0$	absolute angular velocity of frame F^i
ρ_i	linear mass density (mass per unit length) of link i
$\Delta \hat{\mathbf{u}}$	future control moves
$\delta_{ij}(t)$	mode coordinates
ω_i	the natural frequency of vibration of link i

Abbreviations

ADC	Analog-to-Digital Converter
CARIMA	Controlled Auto-Regressive and Integrated Moving Average
DAC	Digital-to-Analog Converter
DMC	Dynamic Matrix Control
FIS	Fuzzy Inference System

FLMS	Flexible Link Manipulator System
GPC	Generalized Predictive Control
GUI	Graphical User Interface
IMPC	Intelligent Model Predictive Control
LMIs	Linear Matrix Inequalities
LP	Linear Programming
LQG	Linear Quadratic Gaussian
LQR	Linear Quadratic Regulator
LS	Least Squares
LTI	Linear Time-Invariant System
MIMO	Multi-Input Multi-Output
MPC	Model Predictive Control
PDE	Partial Differential Equation
PEM	Prediction-error method
PID	Proportional-Integral-Derivative
PWM	Pulse-Width Modulation
QP	Quadratic Programming
SISO	Single-Input Single-Output
SMC	Sliding Mode Control

Acknowledgments

I wish to thank my research supervisor, Prof. Clarence W. de Silva, for his continuous advice and guidance of my research for his willingness to discuss my progress. I am thankful for his kind encouragement to keep me motivated, outstanding supervision, comprehensive knowledge, and effort throughout the entire process of this research. I greatly appreciate his constructive comments and criticism, and the valuable discussions that speeded up the progress of this project.

I would also like to thank my previous research supervisor, the late Prof. Dale Cherchas, for his advice and contributions to this research project.

I wish to express my appreciation to the external examiner for his careful reading of the thesis and insightful comments. I would also like to express my gratitude to the supervisor committee, Prof. Mohamed S. Gadala, Prof. Farrokh Sassani, and Dr. Ricky Min-Fan Lee for their constructive comments and suggestions. This thesis would not have attained its final state without the positive feedback and invaluable suggestions I received from other faculty members as well. Their kind support is gratefully acknowledged.

Financial support for the research presented in this thesis has been provided through various sources. These include the scholarships from the University of British Columbia in the form of Graduate Entrance Scholarship, Ph.D. Tuition Fee Awards; research assistantships and equipment funding from the Natural Sciences and Engineering Research Council (NSERC) of Canada which are held by Prof. Clarence W. de Silva.

I take this opportunity to thank my good friends and colleague with whom I shared my pleasant moments during the study at UBC; especially, Dr. Poi Loon Tang, Mr. Ying Wang, and Mr. Jason J. Zhang, for many fruitful discussions. I truly appreciate my parents and my family members for their unwavering encouragement and unconditional support.

Chapter 1

Introduction

1.1 Goals of the Research

High-speed and high-precision motion control of robot manipulators with flexible links is naturally more complex and difficult than the motion control of those with conventional rigid links. The inherent non-minimum phase behavior, dynamic model complexity, lack of complete sensing, multi-modal vibration and incapability of precise positioning due to system flexibility are among the primary reasons for the difficulty of controlling a flexible-link manipulator in achieving high accuracy and robustness simultaneously. The stability of the robotic system is particularly relevant in this context. Consequently, practical application of flexible-link manipulators, which need efficient control algorithms, is rather limited. The motion control problem of a flexible robot manipulator may be solved by a combination of the following approaches (Book, 1993):

- Design of material and shapes such that high stiffness/mass ratio is achieved.
- Passive damping treatment of flexible elements in order to reduce vibrations.
- Improved dynamic modeling that allows reliable design and control.
- Active feedback control of flexible vibrations.
- Trajectory planning algorithms to constrain the excitement of flexible modes.
- Use of sensors and actuators distributed in the structure.
- Sensors for directly detecting end-of-arm position and states.

Research to examine the advantages and limitations of these complementary approaches in accommodating link flexibility is crucial in making flexible manipulators desirable for practical manipulation tasks. The intelligent model predictive control (IMPC) which is developed in the present thesis represents a focused effort toward this goal. It involves the design of an active feedback control strategy that is effective in the associated motion control

problem.

The main objectives of the research presented in this thesis are the following:

1. Develop an explicit, realistic, and accurate dynamic model for a typical flexible link manipulator. The model must be capable of capturing the process dynamics so as to predict the future outputs with sufficient precision, while being simple enough to implement and analyze for controller design. It must incorporate realistic boundary conditions.
2. Design an effective model predictive controller for high-speed real-time implementations with guaranteed nominal stability.
3. Extend the scheme of model predictive control (MPC) to intelligent model predictive control (IMPC) and investigate relevant analytical and practical issues of the enhanced control scheme.
4. Apply IMPC for the motion control of structurally flexible robotic manipulators and investigate its performance through analysis, computer simulation, and experimentation.

The developed controller is expected to possess the advantages of conventional MPC and also those of intelligent control. In particular, the controller will be efficient, intelligent, adaptive, and robust. It should be useful in the real-time control of high-speed motions of a robot with flexible links. Also, it should be able to automatically adjust the controller parameters and model structures to guarantee a sufficiently small end-point tracking error and closed-loop system stability for different payloads and disturbances. The developed control approach will incorporate the experience and other knowledge of human experts in tuning the parameters of a conventional MPC controller, in order to achieve high system performance.

Both computer simulation and experimentation will be used to evaluate the performance of the developed control methodology. In particular, an IMPC controller will be designed to control the end-effector motions of a planar revolute robot manipulator with flexible link. As a test bed, a prototype flexible link manipulator system (FLMS) is designed and constructed. It will be used to verify and refine the modeling and control strategies that are developed in the present work. Both simulation and experimental studies will be carried out using the flexible link robotic system, in order to validate the control methodologies developed in the present work. A view of the prototype manipulator, which has been designed and developed

in our laboratory is shown in Figure 1.1.

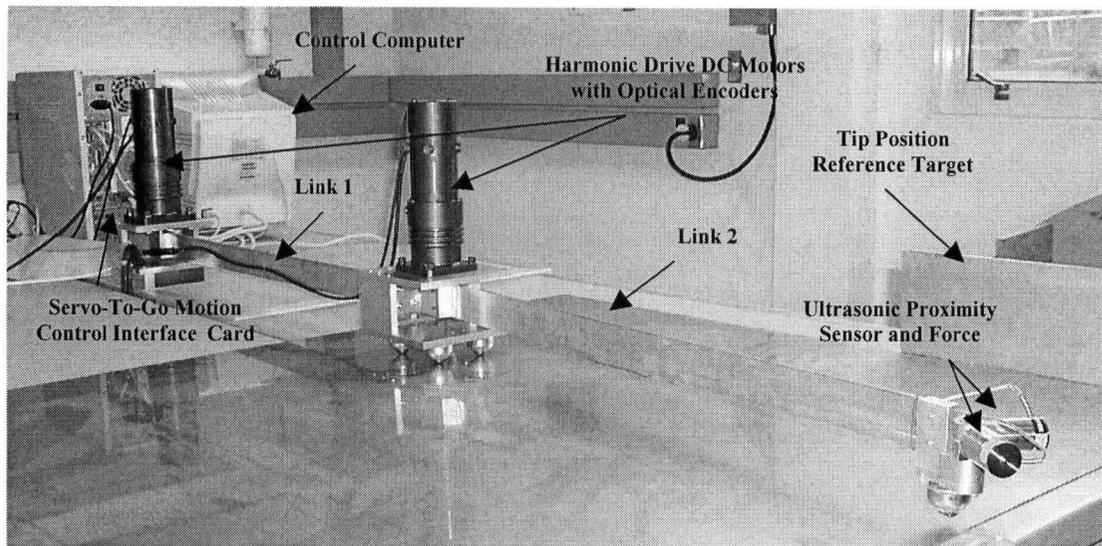


Figure 1.1 Prototype flexible link manipulator system.

1.2 Motivation

In this section, first some basic concepts and definitions that are relevant to the control of flexible manipulators, which is the application domain of intelligent model predictive control (IMPC) as developed in the present research, are reviewed. The advantages of flexible manipulators, and challenges and difficulties of the related motion control problem are addressed next, which provides the rationale for the investigation into IMPC.

Robot manipulators have a wide range of applications, from industrial automation and medical operations to exploring hazardous environments such as space, underground, underwater and nuclear plants. In all these applications, completion of a generic task requires accurate control the movements of the end-effector of one or more manipulators. In general, the control of a robotic manipulator consists of motion control and contact force control. For unconstrained (free) motion, where there is no physical interaction between the end-effector and the environment, only motion control is required. For constrained motion, where contact forces will arise between the end-effector and the environment, both motion and contact forces may have to be controlled.

The motion control problem of a robot manipulator can be generally divided into two categories: regulation (or stabilization) and tracking (or servoing). In the regulation problem, one is concerned with devising a control algorithm such that the system states are driven to a desired final equilibrium point and stabilized around that point. In the tracking problem, one is faced with designing a controller such that the system output tracks a reference spatial path with respect to time (i.e., a trajectory). The regulation problem can be regarded as the special case of tracking problem when the desired trajectory is a constant spatial configuration with respect to time. Trajectory of the robot end-effector or joints, as generated by path planning, is realized by the appropriate reference inputs to the motion control system of the manipulator. The problem of manipulator control generally involves the determination of the time behavior of the forces and torques to be delivered by the joint actuators so as to ensure the execution of the reference trajectories. The controller has to drive the outputs (responses) close to the desired trajectories while maintaining stability of the overall robot system.

Structural flexibility refers to the deflection of a structure under applied or inertial (acceleration-related) forces/torques. Here, the structural flexibility may be addressed from the control point of view. A robotic manipulator becomes flexible if the flexural effects are significant such that they cannot be neglected during the controller design stage in order for the system to meet the performance specifications. Flexibilities in a robotic manipulator may result from joint flexibilities (Spong, 1987) and link flexibilities. Joint flexibility arises primarily because of the elastic behavior of the joint transmission elements such as gears, harmonic drives, and shafts of the actuators. Link flexibility is a consequence of light-weight constructional features of large-dimension manipulators, which result in low link stiffness. They are designed to operate at high speeds with low inertia or to handle heavy payloads.

Research on the dynamic modeling and control of flexible manipulators has received increased attention during the past 30 years due to the advantages of flexible manipulators over rigid ones (Book, 1984). Two types of robotic systems motivate the application domain of the present research: light-weight high-speed industrial manipulators, and large dimension manipulators capable of handling heavy payloads such as space exploration manipulators. Advantages of flexible manipulators over their rigid counterparts include: higher payload to robot weight ratio, low energy consumption, use of smaller actuators, greater maneuverability, increased transportability, lower mounting strength requirements, reduced overall cost, and

safer operation due to reduced inertia. Achieving accurate, high-speed manipulation with a light-weight structure is clearly a desirable objective and a significant challenge.

The motion control of the flexible manipulator is more difficult than the rigid manipulator. For a rigid manipulator, the end-point trajectory is completely defined by the trajectory of the joints. Effective control of the joints is equivalent to good control of the end-point motion. The situation is not as straightforward for a flexible manipulator, and difficulties arise when one tries to track a specified end-effector position trajectory by applying the torque at the joint. When a distributed parameter system is forced at one point in its spatial domain and its response is measured at another point, the system is said to be non-collocated. In this case, the control becomes rather difficult due to the non-collocated nature of the sensor and actuator positions, which can result in unstable zero dynamics (Wang and Vidyasagar, 1991). In other words, the system can be non-minimum phase. Then, the system has an unstable inverse. Many theoretical results of control are complicated by or even totally voided by a system of non-minimum phase. It can be shown that the non-minimum phase property of a flexible manipulator makes exact asymptotic tracking of a desired tip trajectory impossible by means of a causal controller (Slotine and Li, 1991). Thus, in practice, one may be satisfied with small (rather than zero) tracking errors. Furthermore, use of a reduced-order model for the controller design may also lead to control and observation spillover. Control spillover is the excitation of the residual modes of the system by the control action. Observation spillover is the contamination of sensor readings by the residual modes. When control and observation spillovers are present, the closed-loop system may become unstable. Further complications arise because of the highly nonlinear nature of the system and the difficulties involved in accurately modeling various friction and backlash terms. Moreover, a change in the arm configuration and in the payload also leads to a change in the system dynamics. This change has a degrading effect on the performance of the controllers.

Due to the complexity of the dynamics of structurally flexible robotic manipulators, designers of conventional robotic manipulators usually avoid confronting the flexibility issues in an explicit manner. The motion control algorithms of these systems are based on the rigid model of the system, the flexibility of the system being considered negligible. There are two primary solutions in current usage to avoid the flexibility problems of the system. The first is to make the manipulator rigid by increasing the stiffness of the system. This will reduce the

system vibration, and good positional accuracy can be achieved. High stiffness usually is achieved by using heavy and bulky structural components. Most of the conventional industrial robotic manipulators are designed based on this concept. This will reduce the efficiency and operating speed/bandwidth, increased the cost, and limit the performance of the manipulator. In fact, most existing industrial robot manipulators are limited to a load-carrying capacity of about 5-10 percent of their own weight. Heavy weight of the system means large inertia of the system. In order to drive the system at high speed, large actuators will be needed. This will increase the overall cost and energy consumption. The second solution is to reduce the speed of the manipulator. By moving slowly enough, it might be possible to ensure that the flexible modes of the system are not excited, and then the system can be controlled as a rigid system. This conservative strategy is used, for example to control space-based robotic manipulators. Space applications require low-mass designs of the robotic manipulator to achieve escape velocity, and in order to accomplish a mission with better fuel economy. The workspace of a space robotic manipulator is usually large. Weightlessness of space means that mobility provided by wheels and legs on ground-based systems is ineffective for space-based system. Thus a long robotic manipulator is used to move equipment and material into and out of the cargo bay. The primary control technique for space manipulator that is employed at present is to move the joints slowly and wait for the tip of the arm to settle to equilibrium. This can lead to slow performance and extended the task completion time past acceptable limits. In order to overcome such performance limitations of existing robotic system, we have to explicitly address the flexibility issue of the robotic manipulator systems. With advances in hardware and software, the implementations of high performance advanced motion control methods have become possible even for complex plants.

There are a variety of advanced control techniques, which enjoy various degrees of success in practical applications. Among them is the class of techniques known as model predictive control (MPC). These techniques are model-based approaches where the future response of the plant is predicted, and the controller seeks to make the predicted response approach a desired response. Such an approach is desirable when a satisfactory model of the plant is available and desired response is rather complex. Many industrial applications meet these characteristics. Since an accurate plant model may not be present in many practical applications and since there are parameters of MPC that can be tuned online to achieve

improved performance, particular for complex plants and under model uncertainties, it is useful to investigate procedures for “tuning” an MPC scheme. Control expertise, past experience, and insight into the MPC scheme and the specific plant may be used in developing a knowledge-based “intelligent” scheme to tune the parameters of MPC. This is the main motivation for the proposed investigation in developing an IMPC scheme.

In summary, the non-minimum phase characteristic, coupling effects, nonlinearities, parameter variations, and unmodelled dynamics all contribute to make the problem of controlling the end-effector of a flexible link manipulators much more difficult than that of a rigid-link manipulator. Control strategies that ignore these uncertainties and nonlinearities generally fail to provide satisfactory closed-loop performance. It is expected that the proposed technique of intelligent model predictive control will provide a suitable solution for this class of applications.

1.3 Related Work

Three main research aspects have to be investigated in order to achieve the objectives of the research. They are: model predictive control (MPC), controller tuning, and control of flexible link robotic manipulators. In the following sections, we will review relevant existing work and approaches in this three research areas.

1.3.1 Model Predictive Control

Model predictive control (MPC) is a model-based control technique. The term MPC generally denotes a collection of controllers, which determine the control effort by minimizing a cost function (usually quadratic) in a receding horizon manner using an explicit model while satisfying some imposed constraints. Physical limitations of the system; e.g., valve saturations, may be represented by an input constraints. The state constraints are imposed for states or outputs that may not have set-points, but are required to remain within certain limits during the intended system operation. The existing MPC schemes share the following essential features: an explicit internal model, the receding horizon concept, and the computation of the control signal by optimizing predicted plant behavior.

MPC strategies were developed independently by industrial groups in France and the

United States in the late 1970's. The pioneers of MPC are mostly industrial practitioners who implemented MPC several years before the first publication appeared. Richalet, *et al.* (1978), of the French company Adersa, proposed predictive control based on impulse-response model under the name model predictive heuristic control (MPHC), later known as model algorithmic control (MAC). This control method is based on intuitive concepts and offers ease of tuning. It can be applied to problems that are too difficult to be handled by conventional proportional-integral-derivative (PID) control. Constraint handling and optimality are not the principal objectives. Culter and Ramaker (1985), of the Shell Oil Co., Texas, proposed dynamic matrix control (DMC). This control algorithm is an input-output approach based on step/impulse response models. It emphasizes optimal plant operation under constraints, and computes the control signal by repeatedly solving a linear programming (LP) problem.

Clarke, *et al.* (1987) proposed one of the most popular predictive control algorithms, generalized predictive control (GPC), with an analysis of its properties. This receding-horizon method depends on predicting the output of the plant over several steps, based on assumptions on future control actions. In the absence of constraints, it can provide an analytical solution. It can deal with unstable and non-minimum phase plants, and incorporates the concept of control horizon as well as the consideration of weighting of control increments in the cost function. The roles of the output and control horizons were explored for processes with non-minimum phase, unstable and variable dead-time models. The general set of choice available for GPC leads to a greater variety of control objectives compared to other approaches, some of which can be considered subsets or limiting cases of GPC. Furthermore, the offsets are eliminated by the consequence of assuming a controlled auto-regressive and integrated moving average (CARIMA) plant model. The stability of the GPC scheme is guaranteed using end-point weighting of the cost function, which gives the required monotonicity to the Difference Riccati Equation associated with the cost function by Demircioglu and Clarke (1993). Rawlings and Muske (1993) provided a stability proof of constrained receding horizon control based on a finite number of optimization parameters but an infinite prediction horizon.

Kothare, *et al.* (1996) presented a technique for the synthesis of a robust model predictive control law, using linear matrix inequalities (LMIs). The technique allows incorporation of a large class of plant uncertainty descriptions, and is shown to be robustly stabilizing. This

work exploits the powerful methods, which have recently become available for solving LMIs and the connections which exist between LMIs and control theory. Although the formulation no longer leads to a Quadratic Programming (QP) problem, it does lead to an LMIs optimization problem, which is a convex problem. The algorithm available for solving LMIs optimization is efficient, making it a plausible candidate for online use.

1.3.2 Controller Tuning

One of the challenges for MPC, however, is how to tune the parameters of the controller. So far, only a few guidelines related to the tuning of the parameters of MPC controllers have been provided in the literature. Clarke and Mohtadi (1989) have developed some general guidelines on how to tune a GPC. They have shown that variable dead-time and unstable/inverse-unstable plants can be stably controlled by correct choices of the horizon. They recommended that the first costing horizon should at least exceed the plant dead-time. The prediction horizon should be set comparable to the rise-time of the open loop system. The control horizon determines the degrees-of-freedom in future control increments, and it can be set the number of unstable/underdamped poles of the open loop system. Lee and Yu (1994) presented practical on/off-line tuning rules for state-space MPC controllers. Guidelines for tuning the predictive controller for robust performance are provided based on a fixed tuning of the cost function weights and a manipulation of the observer covariance matrices. Based on the frequency-domain analysis of the closed-loop behavior of MPC controllers, the effect of various tuning parameters on the performance and robustness of the closed-loop system is characterized, and quantitative guidelines on how these parameters are best determined are established. It is shown that the choice and the settings of the tuning parameters play a critical role in the overall robustness of the resulting closed-loop system and also the ease of design and tuning. Liu and Wang (2000) proposed an auto-tuning procedure for a predictive controller. Their auto-tuning procedure is based on a recursive multi-objective optimization algorithm.

Now we will review some existing approaches related to knowledge-based tuning of controllers. De Silva (1991) investigated a knowledge-based tuner consisting of a two-level control structure. The lower level is occupied by a conventional servo loop. The upper level consists of a knowledge-based tuner for the servo parameters. In his work, knowledge-based

tuner was expressed in the conventional fuzzy-logic formulation of a linguistic rule base to which the compositional rule of inference is applicable; concepts of rule dissociation, fuzzy resolution, and resolution relationships were defined; and the stability of the overall control system and the computational requirements for the tuning system were analyzed. The developed strategy was used to tune a PID controller. Wu and De Silva (1993) investigated the effect of fuzzy resolutions on the processing speed, storage requirement, and response accuracy of a hierarchical fuzzy tuning system. The paper suggests, in practice, the fuzzy resolution has to be chosen to satisfy the requirements of the particular situation, by weighing various factors.

1.3.3 Control of Flexible Link Robotic Manipulators

For the modeling and control of flexible link robotic manipulator, most of researchers using planar single-link manipulator flexible model or planar two-link flexible manipulator model. Many of the advanced control algorithms have been applied to the flexible link manipulator. A flexible link manipulator is a nonlinear and distributed parameter system. The model of this system is an infinite-dimensional model. A finite-dimensional nonlinear model can be obtained by the assumed mode techniques or finite element method. Three different approaches have been used in the control design. In the first approach, the control law is designed based on finite-dimensional model. In the second approach, the control law is designed based on distributed parameter model in spite of the model complexities. In the third approach, the controller is designed using soft computing control techniques. Next we will review some major contributions in these three different approaches.

The first approach is based on discretized dynamic models. In an order to reduce the complexity, many of the work on flexible manipulator control perform local linearization and truncate the number of flexible modes.

Cannon and Schmitz (1984) designed a Linear Quadratic Gaussian (LQG) controller for a single-link flexible manipulator. Estimation of the system's states is based on dynamic calculation using signals from the tip-position sensor and the hub-rate sensor. They showed that the speed of response to commands is ultimately limited by the inherent wave-propagation delay for the beam. Morris and Madani (1998) developed a quadratic optimal control algorithm for a two link flexible manipulator. Static deflection of flexible links under

gravity was considered. Availability of all the states of the system was assumed.

Singer and Seering (1990) proposed an input shaping control technique. This method essentially involves the convolution of a sequence of impulses with the reference input to suppress the vibration of flexible modes. They showed that the controller robustness with respect to uncertain frequencies or damping can be increased by using a larger number of impulses. They also showed that multiple modes of vibration could be handled. Magee and Book (1992) extend the input shaping control method to deal with variable frequency systems. They proposed a command shaping algorithm that can filter out the frequencies around the flexible modes. Lee and Park (2002) presented a modified input shaping method, which can be applicable to linear time-varying system.

In a practical sense, linear feedback control with observer can be undesirably sensitive to small changes in parameters or payload. Many adaptive and robust control algorithms based on bounded uncertainty estimates have been presented for flexible link manipulator with large payload changes and parameter uncertainties. Banavar and Dominic (1995) presented an LQG/ H_{∞} controller for a single-link flexible manipulator. While the flexible modes are damped out in the inner-loop by the LQG controller, the outer-loop ensures stability of the system in the presence of uncertainties. Feliu, *et al.* (1997) has studied the control of single-link flexible manipulator in the presence of joint friction and payload changes. They developed a robust control method based on two nested loops: an inner feedback loop to control the motor position which is robust to joint friction, and an outer loop to control the tip position which is robust to payload changes. Bai, *et al.* (1998) presented an adaptive augment state feedback controller for a two-link flexible manipulator. The controller design is based on the Linear Quadratic Regulator (LQR) technique in conjunction with an adaptive compensator, where all states are adaptively estimated by employing the strong tracking filter. This will increase the robustness of the controller against payload variations. Ryu, *et al.* (2000) proposed a robust LQR based on the descriptor form for the control of a single-link flexible manipulator, which has a large uncertain payload variation and parameter uncertainties. The controller design problem is formulated as a convex programming problem and is solved using LMIs. The controller designed with this method shows the improved robust performance and the reduced conservatism.

Although linear controllers may yield desirable closed-loop performance, their region of

operation is limited due to the nonlinearities present in the original system. To have a wider region of operation, one has to take into account the nonlinearities.

Yim designed a modified predictive controller based on the nonlinear predictive control theory. A vector function s has been chosen as a linear combination of tracking error, its higher order derivatives, and the integral of the tracking error for robustness. A control law has been derived by minimizing a quadratic function of the predicted value of s and control torque. Wilson, *et al.* (2002) proposed an augmented sliding mode control (SMC) scheme, which ensures a good performance in the rigid body motion and provides sufficient damping in the measured flexible body modes. SMC takes advantages of control law switching to move a system from an initial state to a prescribed surface in the state space. Once on that surface, a second control law is used to keep the state from leaving the surface while moving toward the desired final state. It is robust to model-parameter uncertainty and disturbances if bounds are known a priori. Due to the use of Lyapunov stability theory during the controller design, SMC is appealing for nonlinear system control. The novel aspect of the proposed augmented SMC is that the control law neglects the flexible body generalized accelerations, but the system remains stable even when flexible body generalized accelerations, unmodeled dynamics, disturbances and model uncertainties are present.

The most common approach to compensate the nonlinear dynamics is the so-called inverse dynamics or computed torque strategy. De Luca and Siciliano (1989) designed a nonlinear control law using the input-output inversion algorithm. They have shown that the closed-loop dynamics is always stable when joint angle is the output; the zero dynamics of the system may become unstable when the output is the angular position of a point along the link. Thus, in general, end-point trajectories cannot be tracked exactly without going unstable. Wang and Vidyasagar (1991) have shown that the nonlinear flexible link system is not in general input-state linearizable. However, the system is locally input-output linearizable but the associated zero dynamics may become unstable when the tip position is considered as an output of the system. In order to overcome the non-minimum phase characteristic of the system, two most common used methods are multiple time scale composite control based on the singular perturbation (Siciliano and Book, 1988) and output redefinition approach (Wang and Vidyasagar, 1989, Madhavan and Singh, 1991). Singular perturbation control strategy is based on a two-time scale model of the flexible manipulator. This allows the definition of a

slow subsystem corresponding to the rigid body motion, and a fast subsystem describing the flexible motion. A composite control strategy is then applied. A linear stabilizer is used to stabilize the fast dynamics (flexible modes) and a nonlinear controller is used to make the slow dynamics (joint variables) track the desired trajectories. In order to improve the performance of a singular perturbation approach, several researchers used the integral manifold approach introduced by Spong, *et al.*, (1987) to control the flexible link manipulator (Siciliano, *et al.*, 1986, Moallem, *et al.*, 1997). The integral manifold approach facilitates the inclusion the effects of higher frequency flexible modes into the corrected models. The output redefinition approach is also used to overcome the non-minimum phase characteristics of the system. The basic idea is to redefine the system outputs such that the zero dynamics of the system is stable. Based on the new outputs, the input-output (inverse dynamics) approach can be used to control the flexible link manipulator.

In the second approach, the control law is designed based on the distributed parameter model. This will avoid the undesired model truncation. The main advantage of this approach is that it is free from control spillover. Luo (1993) proposed an asymptotically stable strain feedback control for regulating control of the link vibration. This strategy directly introduces a damping term into the system model, and precise dynamic model is not needed for real-time control. Ge *et al.* (1998) designed a strain feedback regulating control, where the payload mass was considered. The controller is independent of system parameters and thus robust to system parameters uncertainties. The closed-loop stability of the system is proven based on the partial differential equations (PDE) of the system. Lee (2004) developed a moment-feedback trajectory-tracking control scheme based on a distributed parameter model. The stability of the closed-loop system is proven based on the Lyapunov stability theorem.

Intelligent control methods have been applied to flexible link manipulators. Kubica and Wang (1993) developed a two stages direct fuzzy logic controller. One stage controls the rigid motion while the second stage monitors the elastic deformations and modifies the output of the first stage to reduce the induced vibrations. This approach was taken to reduce the number of rules needed in the associative fuzzy knowledge base. Garcia-Bentiez, *et al.* (1993) proposed a two level hierarchical control strategy. A fuzzy logic supervisory level is used for the selection of a lower-level conventional controller. Moudgal, *et al.* (1994) designed a two-level hierarchical rule-based controller. This scheme employs an upper-level “expert

controller” that captures the knowledge about how to supervise the application of low-level fuzzy controllers during movements in the robot workspace. Gutierrez and Lewis (1998) developed a neural network tracking controller. The controller is composed of an outer PD tracking loop, a singular perturbation inner loop for stabilization of the fast dynamics, and a neural network inner loop used to feedback linearize the slow dynamics. The neural network controller requires no off-line learning phase. Song and Koivo (1999) have designed a nonlinear predictive controller. A neural network has been constructed to represent the input-output relation of a dynamic model. Based on the neural network plant model, the control inputs are calculated by minimizing a projected cost function that penalizes the future tracking errors. Nho and Meckl (2003) proposed a control architecture that incorporates a neural network, a fuzzy logic and a PD controller. The neural network controller is trained offline to capture the inverse dynamics of the plant, and the fuzzy logic estimator, trained offline to optimize the membership functions, is placed in the feedback loop to estimate the unknown payload. The estimated payload is used as an additional input to the neural network controller to update the inverse dynamics after payload changes. Wai and Lee (2004) developed an intelligent optimal control system, in which the proposed controller was made of an optimal controller which minimize a quadratic performance index, a fuzzy neural network (FNN) controller which learn a nonlinear function to implement the optimal controller, and a robust controller which compensate the approximation error of the FNN controller.

1.4 Organization of the Thesis

As an overview, the remainder of the thesis presents the theoretical and experimental research on the modeling and control of flexible link robotic manipulators in sufficient detail. The thesis is divided into seven chapters and three appendices. Included in **Chapter 1** are the goals and motivation for this research, related work, contributions and organization of the thesis.

Chapter 2 presents the development of kinematic and dynamic models of flexible link robotic manipulator. Boundary conditions and mode shape selections for flexible link model are discussed. Equations of motion for a planar n -link flexible manipulator are presented. Detailed nonlinear mathematical models for a single-link and two-link flexible manipulator

are derived. Linearization of the equations of motion technique is developed. Non-minimum phase characteristic of the system are analyzed.

Chapter 3 presents the IMPC algorithm, and its underlying strategy, and addresses issues related to the application of IMPC in the motion control of flexible link robot manipulators.

Chapter 4 presents IMPC controller design for a single-link flexible manipulator. The performance of the controller is evaluated using computer simulations.

Chapter 5 presents IMPC controller design for a two-link flexible manipulator and computer simulation results.

Chapter 6 presents the experimental flexible link manipulator system (FLMS). Experimental results are presented and discussed.

Chapter 7 concludes this thesis with a summary of the major results and recommendations for future research in the area.

Chapter 2

Manipulator Model

Advanced model-based control of a complex system generally involves three main steps: model development and analysis, parameter estimation/measurement, and controller design and implementation. This chapter represents the first step of the process of controller design. Model-based predictive control uses a plant model to predict the future response of the system and on this basis decides the control inputs to the system. It follows that the development of a suitable dynamic model is a crucial part in the process of controller design. The dynamic model should capture important features of the plant that are relevant to designing the controller. Yet, it should not be so complex as to mask the essence of the problem. With this backdrop, in the present work, a dynamic model of a flexible robotic manipulator is used for the design, simulation and control of the manipulator.

In this chapter kinematic and dynamic (kinetic) modeling of flexible-link robotic manipulator is investigated in both general and special-case perspectives. In Section 2.1, kinematic formulation of a general, planar n -link flexible manipulator with revolute joints is presented. In Section 2.2, a flexible link model of the manipulator is developed based on the Bernoulli-Euler beam theory. Realistic boundary conditions for the manipulator under some common practical conditions are given and discussed. Mode shape selection for analysis and computer simulation of the manipulator are discussed. In Section 2.3, the overall dynamic model of the manipulator is derived using Lagrange's equations. Closed-form equations of motion for a planar n -link flexible manipulator are presented. Detailed mathematical models for a single-link and two-link flexible manipulator are derived. In Section 2.4, linearization of the equations of motion is presented. In Section 2.5, the non-minimum phase characteristic of the flexible manipulator system is analyzed.

2.1 Kinematics

Consider a planar n -link flexible manipulator with revolute joints. The links are subjected to bending deformations only in the plane of motion, and torsional effects are neglected. Figure 2.1 (a) shows a two-link example, and Figure 2.1 (b) shows a serial chain of n -link flexible manipulator. The geometry of the system is described in terms of the coordinates shown in Figure 2.1. Frame F^0 is the fixed, world coordinate frame with the joint of link 1 located at its origin. Frame F^i is the rigid-body moving frame of link i , and is fixed at the joint of link i . Frame \hat{F}^i is the flexible-body moving frame of link i , and is fixed at the end of link i . The rigid motion of link i is described by the joint angle θ_i , and the transversal deflection of link i at abscissa x_i is denoted by $w_i(x_i, t)$.

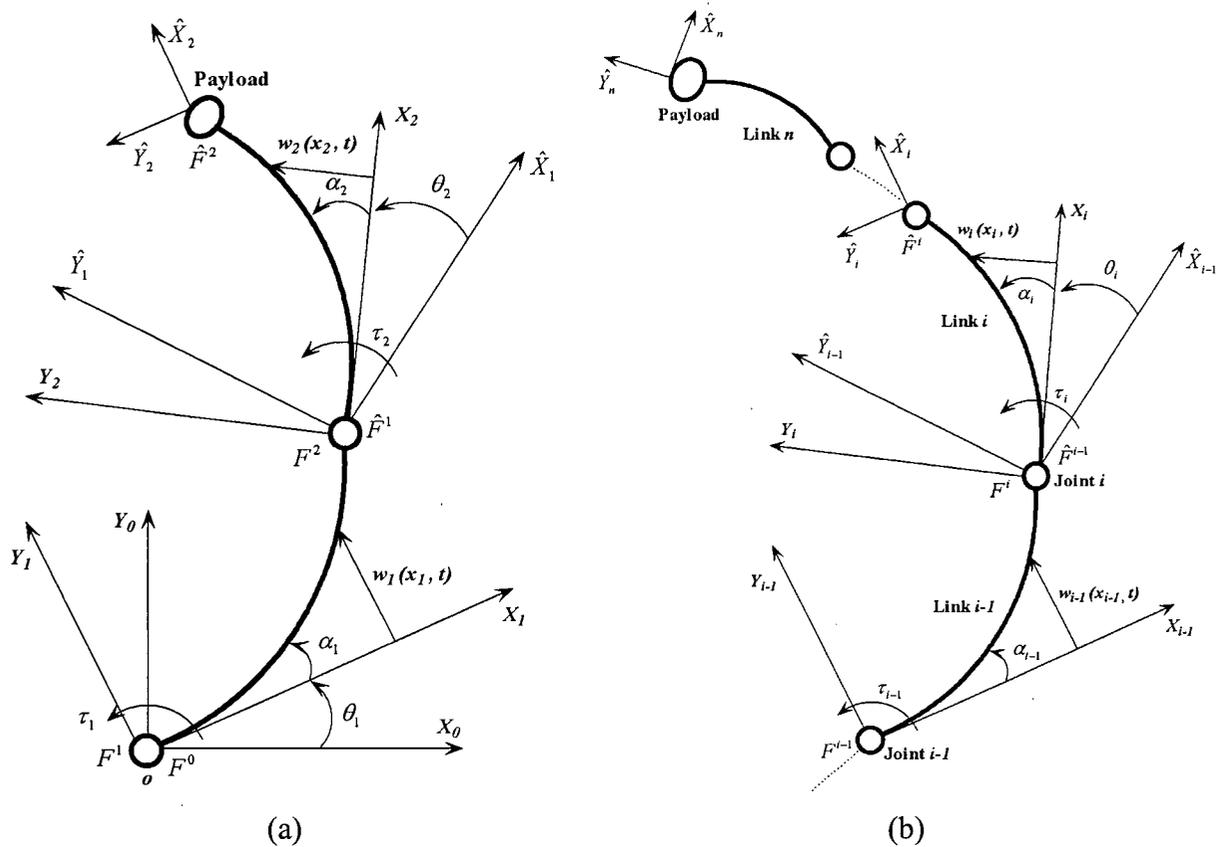


Figure 2.1 Schematic diagrams of flexible link manipulators: (a) Two-link manipulator; (b) General manipulator.

The geometric relations among the coordinate frames can be described using homogeneous transformation matrices (Spong and Vidyasagar, 1989). The homogeneous transformation matrix that transforms the coordinates of a point from frame F^j to frame F^i is denoted by $\mathbf{H}_{i,j}$. Specifically, the homogeneous transformation between frames F^1 and F^0 is

$$\mathbf{H}_{0,1}(t) = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

where $s_1 = \sin \theta_1(t)$ and $c_1 = \cos \theta_1(t)$. The rigid-body components of rotation and translation between frames F^i and F^{i-1} are given by the homogeneous transformation between frame F^i and frame \hat{F}^{i-1} :

$$\mathbf{H}_{i-1,i}(t) = \begin{bmatrix} c_i & -s_i & 0 & 0 \\ s_i & c_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

where $s_i = \sin \theta_i(t)$ and $c_i = \cos \theta_i(t)$. Additional rotation and translation due to the structural deformation of link $i-1$ must be considered. The deflections of the flexible links are assumed to be small compared to their lengths, so the high-order terms of deformations can be neglected. The changes of link lengths due to deformation are also neglected. The flexible body components of rotation and translation between frames F^i and F^{i-1} is the homogeneous transformation between frame \hat{F}^{i-1} and frame F^{i-1} :

$$\mathbf{H}_{i-1,\hat{i}-1}(t) = \begin{bmatrix} c_{\alpha_{i-1}} & -s_{\alpha_{i-1}} & 0 & l_{i-1} \\ s_{\alpha_{i-1}} & c_{\alpha_{i-1}} & 0 & w_{i-1}(l_{i-1}, t) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

where l_{i-1} is the length of link $i-1$, $s_{\alpha_{i-1}} = \sin \alpha_{i-1}(t)$, $c_{\alpha_{i-1}} = \cos \alpha_{i-1}(t)$, and α_{i-1} is the angle of rotation of link $i-1$ due to bending. Note that

$$\alpha_i = \left. \frac{\partial w_i(x_i, t)}{\partial x_i} \right|_{x_i=l_i} \quad (2.4)$$

The homogeneous transformation between frames F^i and frame F^{i-1} is

$$\mathbf{H}_{i-1,i}(t) = \mathbf{H}_{i-1,i-1}(t)\mathbf{H}_{i-1,i}(t) = \begin{bmatrix} c_{\alpha_{i-1,i}} & -s_{\alpha_{i-1,i}} & 0 & l_{i-1} \\ s_{\alpha_{i-1,i}} & c_{\alpha_{i-1,i}} & 0 & w_{i-1}(l_{i-1}, t) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

where $s_{\alpha_{i-1,i}} = \sin(\alpha_{i-1}(t) + \theta_i(t))$ and $c_{\alpha_{i-1,i}} = \cos(\alpha_{i-1}(t) + \theta_i(t))$.

In general, the homogeneous transformation of frame F^i with respect to the world coordinate frame F^0 can be characterized by the following composition of consecutive transformations:

$$\mathbf{H}_{0,i}(t) = \mathbf{H}_{0,1}(t)\mathbf{H}_{1,2}(t)\cdots\mathbf{H}_{i-2,i-1}(t)\mathbf{H}_{i-1,i}(t) \quad (2.6)$$

The absolute position and linear velocity at any point on link i can be calculated based on these transformations. For a point \mathbf{r}_i on link i in frame F^i one has

$$\mathbf{r}_i^i(t) = [x_i \quad w_i(x_i, t) \quad 0 \quad 1]^T \quad (2.7)$$

The absolute position of \mathbf{r}_i in frame F^0 is

$$\mathbf{r}_i^0(t) = \mathbf{H}_{0,i}(t)\mathbf{r}_i^i(t) \quad (2.8)$$

The absolute velocity of this point in frame F^0 is given by

$$\mathbf{v}_i^0(t) = \frac{d\mathbf{r}_i^0(t)}{dt} \quad (2.9)$$

The absolute angular velocity of frame F^i is

$$\dot{\theta}_i^0 = \sum_{j=1}^i (\dot{\theta}_j + \dot{\alpha}_{j-1}) \quad (2.10)$$

The position vector $\mathbf{r}_i^0(t)$, the velocity vector $\mathbf{v}_i^0(t)$, and the absolute angular velocity $\dot{\theta}_i^0$

are used in Section 2.3 to form the kinetic energy, potential energy and dissipation function expressions for the flexible link manipulators.

The absolute position and linear velocity for any point on link 1 and link 2 can be calculated based on the transformations given above. For a point \mathbf{r}_1 on link 1 in frame F^1 one has

$$\mathbf{r}_1^1(t) = [x_1 \quad w_1(x_1, t) \quad 0 \quad 1]^T \quad (2.11)$$

The absolute position of \mathbf{r}_1 in frame F^0 is

$$\mathbf{r}_1^0(t) = \mathbf{H}_{0,1}(t)\mathbf{r}_1^1(t) = [c_1x_1 - s_1w_1(x_1, t), \quad s_1x_1 + c_1w_1(x_1, t), \quad 0 \quad 1]^T \quad (2.12)$$

The absolute velocity of this point in frame F^0 is

$$\mathbf{v}_1^0(t) = \frac{d\mathbf{r}_1^0(t)}{dt} = \begin{bmatrix} -[x_1s_1 + c_1w_1(x_1, t)]\dot{\theta}_1 - \dot{w}_1(x_1, t)s_1 \\ [x_1c_1 - s_1w_1(x_1, t)]\dot{\theta}_1 + \dot{w}_1(x_1, t)c_1 \\ 0 \\ 0 \end{bmatrix} \quad (2.13)$$

For a point \mathbf{r}_2 on link 2 in frame F^2 one has

$$\mathbf{r}_2^2(t) = [x_2 \quad w_2(x_2, t) \quad 0 \quad 1]^T \quad (2.14)$$

The absolute position of \mathbf{r}_2 in frame F^0 is

$$\begin{aligned} \mathbf{r}_2^0(t) &= \mathbf{H}_{0,2}\mathbf{r}_2^2(t) = \mathbf{H}_{0,1}(t)\mathbf{H}_{1,2}(t)\mathbf{r}_2^2(t) \\ &= \begin{bmatrix} l_1c_1 - w_1(l_1, t)s_1 + x_2c_{1,\alpha_1,2} - w_2(x_2, t)s_{1,\alpha_1,2} \\ l_1s_1 + w_1(l_1, t)c_1 + x_2s_{1,\alpha_1,2} + w_2(x_2, t)c_{1,\alpha_1,2} \\ 0 \\ 1 \end{bmatrix} \end{aligned} \quad (2.15)$$

The absolute velocity of this point in frame F^0 is

$$\begin{aligned}
\mathbf{v}_2^0(t) &= \frac{d\mathbf{r}_2^0(t)}{dt} \\
&= \begin{bmatrix} -(l_1 s_1 + w_1(l_1, t) c_1) \dot{\theta}_1 - \dot{w}_1(l_1, t) s_1 - x_2 s_{1, \alpha_1, 2} (\dot{\theta}_1 + \dot{\alpha}_1 + \dot{\theta}_2) \\ -w_2(x_2, t) c_{1, \alpha_1, 2} (\dot{\theta}_1 + \dot{\alpha}_1 + \dot{\theta}_2) - \dot{w}_2(x_2, t) s_{1, \alpha_1, 2} \\ (l_1 c_1 - w_1(l_1, t) s_1) \dot{\theta}_1 + \dot{w}_1(l_1, t) c_1 + x_2 c_{1, \alpha_1, 2} (\dot{\theta}_1 + \dot{\alpha}_1 + \dot{\theta}_2) \\ -w_2(x_2, t) s_{1, \alpha_1, 2} (\dot{\theta}_1 + \dot{\alpha}_1 + \dot{\theta}_2) + \dot{w}_2(x_2, t) c_{1, \alpha_1, 2} \\ 0 \\ 0 \end{bmatrix} \quad (2.16)
\end{aligned}$$

2.2 Flexible Link Model

2.2.1 Flexible Beam Equation

Consider the link i of the n -link flexible robotic manipulator shown in Figure 2.1(b), which undergoes transverse vibrations. We use the Bernoulli-Euler beam model to represent this flexible link, for which rotary inertia and shear deformation effects are assumed small (De Silva, 2006). The free-body diagram for a segment of this link with width dx_i at position x_i is shown on the right-hand side of Figure 2.2. The displacement of the segment from the X_i axis is $w_i(x_i, t)$. Also $V_i(x_i, t)$ is the shear force and $M_i(x_i, t)$ is the bending moment.

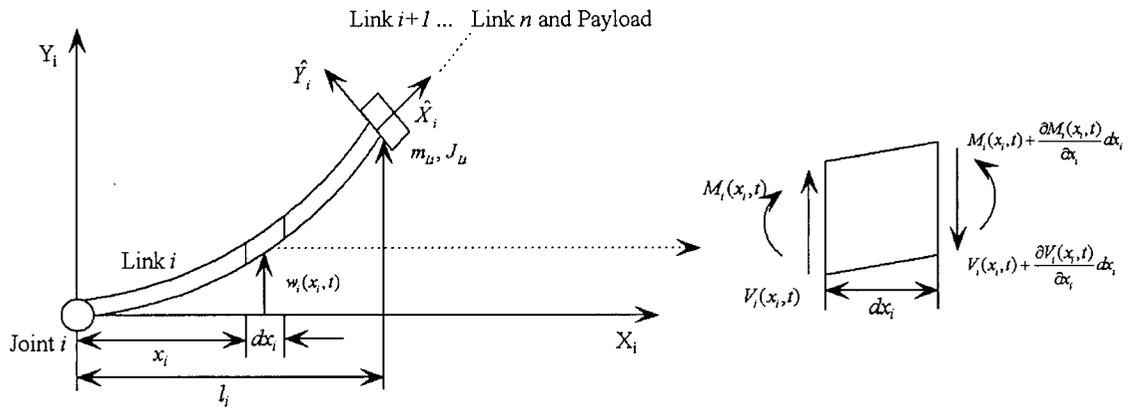


Figure 2.2 Flexible link i and the free-body diagram for an elemental segment.

The force equation of motion (Newton's 2nd law) in the Y_i direction is

$$V_i(x_i, t) - \left(V_i(x_i, t) + \frac{\partial V_i(x_i, t)}{\partial x_i} dx_i \right) = \rho_i(x_i) dx_i \frac{\partial^2 w_i(x_i, t)}{\partial t^2} \quad (2.17)$$

where $\rho_i(x_i)$ is the linear mass density (mass per unit length) of the beam. The moment equilibrium condition (since rotary inertia is neglected) is

$$\left(M_i(x_i, t) + \frac{\partial M_i(x_i, t)}{\partial x_i} dx_i \right) - M_i(x_i, t) - \left(V_i(x_i, t) + \frac{\partial V_i(x_i, t)}{\partial x_i} dx_i \right) dx_i = 0 \quad (2.18)$$

Ignore the second order term with dx_i in equation (2.18), we have

$$V_i(x_i, t) = \frac{\partial M_i(x_i, t)}{\partial x_i} \quad (2.19)$$

Since deformation due to shear is neglected, the relationship between bending moment and deflection can be expressed as (De Silva, 2006)

$$M_i(x_i, t) = E_i I_i(x_i) \frac{\partial^2 w_i(x_i, t)}{\partial x_i^2} \quad (2.20)$$

where $E_i(x_i)$ is Young's modulus and $I_i(x_i)$ is the 2nd moment of area of the link cross section about the Z_i axis. By substituting equation (2.19) and equation (2.20) into equation (2.17), we obtain the Bernoulli-Euler beam equation

$$\frac{\partial^2}{\partial x_i^2} \left[E_i I_i(x_i) \frac{\partial^2 w_i(x_i, t)}{\partial x_i^2} \right] + \rho_i(x_i) \frac{\partial^2 w_i(x_i, t)}{\partial t^2} = 0 \quad (2.21)$$

By assuming $\rho_i(x_i)$ and the flexural rigidity $E_i I_i(x_i)$ to be constant along the length of the link, we obtain the Bernoulli-Euler equation for a uniform beam

$$E_i I_i \frac{\partial^4 w_i(x_i, t)}{\partial x_i^4} + \rho_i \frac{\partial^2 w_i(x_i, t)}{\partial t^2} = 0 \quad (2.22)$$

2.2.2 Boundary Conditions

In order to solve the Bernoulli-Euler beam equation for a flexible-link robotic manipulator, realistic and proper boundary conditions have to be imposed at the joint and end of each link.

The commonly used boundary conditions are given below.

1. Free end:

$$\begin{aligned} \text{Bending moment: } E_i I_i \frac{\partial^2 w_i(x_i, t)}{\partial x^2} &= 0 \\ \text{Shear force: } E_i I_i \frac{\partial^3 w_i(x_i, t)}{\partial x^3} &= 0 \end{aligned} \quad (2.23)$$

2. Simply supported (pinned) end:

$$\begin{aligned} \text{Deflection: } w_i(x_i, t) &= 0 \\ \text{Bending moment: } E_i I_i \frac{\partial^2 w_i(x_i, t)}{\partial x^2} &= 0 \end{aligned} \quad (2.24)$$

3. Fixed (clamped) end:

$$\begin{aligned} \text{Deflection: } w_i(x_i, t) &= 0 \\ \text{Slope: } \frac{\partial w_i(x_i, t)}{\partial x} &= 0 \end{aligned} \quad (2.25)$$

For a flexible-link robotic manipulator, the link inertia is small compared to the hub inertia, so the constrained boundary conditions can be used. It is usually assumed that boundary conditions for each link are clamped-free; i.e., the flexible link is fixed (clamped) at the $x_i = 0$ end and is free of dynamic constraints at the $x_i = l_i$ end. This assumption is made due to the difficulty of accounting for the time-varying or unknown masses and moments of inertia. In the present work we use realistic and more accurate boundary conditions representing the balance of bending moment and shear force at the $x_i = l_i$ end of the flexible link i (clamped-mass). From the free body diagram of the link end element (see Figure 2.3), we have the following boundary conditions for link i :

$$\begin{aligned} \text{At } x_i = 0, \\ w_i(0, t) = 0, \frac{\partial w_i(0, t)}{\partial x_i} &= 0, \\ i = 1, \dots, n. \end{aligned} \quad (2.26)$$

At $x_i = l_i$,

Bending moment:

$$E_i I_i \left. \frac{\partial^2 w_i(x_i, t)}{\partial x_i^2} \right|_{x_i=l_i} = -J_{Li} \frac{d^2}{dt^2} \left(\left. \frac{\partial w_i(x_i, t)}{\partial x_i} \right|_{x_i=l_i} \right) - (MO)_i \left. \frac{\partial^2 w_i(x_i, t)}{\partial t_i^2} \right|_{x_i=l_i} \quad (2.27)$$

Shear force:

$$E_i I_i \left. \frac{\partial^3 w_i(x_i, t)}{\partial x_i^3} \right|_{x_i=l_i} = m_{Li} \frac{d^2}{dt^2} \left(w_i(x_i, t) \Big|_{x_i=l_i} \right) + (MO)_i \frac{\partial^2}{\partial t_i^2} \left(\left. \frac{\partial w_i(x_i, t)}{\partial x_i} \right|_{x_i=l_i} \right)$$

$i = 1, \dots, n$.

where J_{Li} and m_{Li} are the equivalent moment of inertia and mass at the end of link i , respectively, and $(MO)_i$ represents the contributions of the masses of payload and link $i + 1, \dots, n$. due to the mechanical offsets.

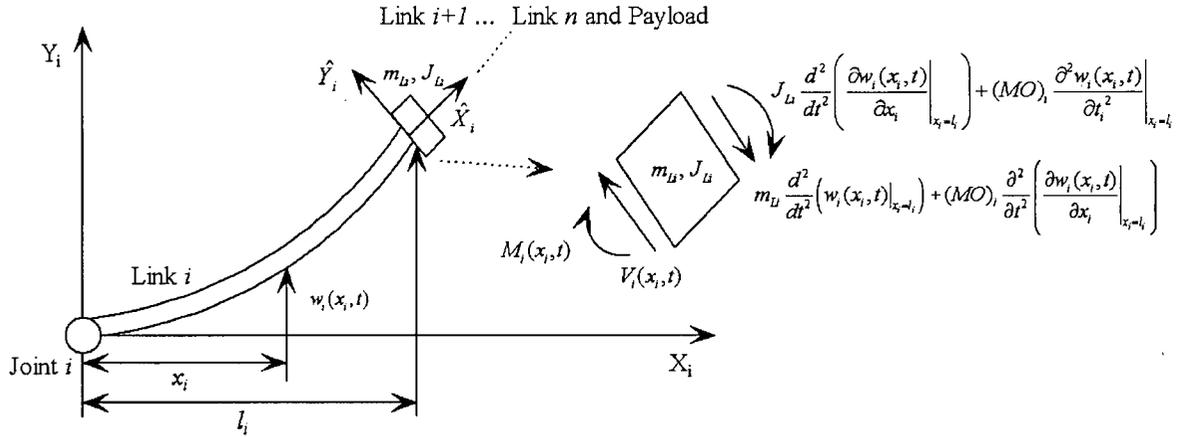


Figure 2.3 Free body diagram of the link end element.

2.2.3 Solution of the Flexible Beam Equation

The solution of the Bernoulli-Euler equation of link i can be found using the method of separation of variables. Specifically, for modal motion, the solution can be separated into a time function and a space function as

$$w_i(x_i, t) = \phi_i(x_i) \delta_i(t) \quad (2.28)$$

Then the beam equation (2.22) can be rewritten as

$$\frac{E_i I_i}{\rho_i \phi_i(x_i)} \frac{d^4 \phi_i(x_i)}{dx_i^4} = \frac{-1}{\delta_i(t)} \frac{d^2 \delta_i(t)}{dt^2} = \omega_i^2 \quad (2.29)$$

where ω_i is a constant since it is simultaneously equal to a completely spatial function and a completely temporal function. The solution of the temporal part of equation (2.29) is

$$\delta_i(t) = A_i \cos(\omega_i t) + B_i \sin(\omega_i t) \quad (2.30)$$

where A_i and B_i are constants which are determined by the initial conditions of motion. The solution of the spatial part of equation (2.29) is

$$\phi_i(x_i) = C_{1,i} (\cosh(\beta_i x_i) + \frac{C_{2,i}}{C_{1,i}} \sinh(\beta_i x_i) + \frac{C_{3,i}}{C_{1,i}} \cos(\beta_i x_i) + \frac{C_{4,i}}{C_{1,i}} \sin(\beta_i x_i)) \quad (2.31)$$

where $C_{1,i}$, $C_{2,i}$, $C_{3,i}$ and $C_{4,i}$ are constants, and $\beta_i^4 = \frac{\rho_i \omega_i^2}{E_i I_i}$. Also, as clear from equation

(2.30), ω_i is the natural frequency of vibration. The function $\phi_i(x_i)$ is known as the mode shape function (or eigenfunction) of the beam and describes the shape of each natural mode of motion of the beam. Application of the clamped-mass boundary conditions (equation (2.26) and equation (2.27)) allows the determination of the constant coefficients in equation (2.31).

The clamped conditions at the joint base of link i yield

$$C_{1,i} = -C_{3,i}, \quad C_{2,i} = -C_{4,i} \quad (2.32)$$

Substituting equation (2.28) into the boundary conditions of the link i : equations (2.26) and (2.27), and using equation (2.29), we obtain

$$\begin{aligned} \phi_i(0) &= 0 \\ \phi_i'(0) &= 0 \\ \phi_i''(l_i) &= \frac{\beta_i^4}{\rho_i} (J_{L_i} \phi_i'(l_i) + (MO)_i \phi_i(l_i)) \\ \phi_i'''(l_i) &= \frac{-\beta_i^4}{\rho_i} (m_{L_i} \phi_i(l_i) + (MO)_i \phi_i'(l_i)) \end{aligned} \quad (2.33)$$

where a prime indicates differentiation with respect to x_i . Substituting equations (2.31) and

(2.32) into equation (2.33) yields

$$\frac{C_{2,i}}{C_{1,i}} = \frac{\cosh(\beta_i l_i) + \cos(\beta_i l_i) - \frac{J_{Li} \beta_i^3}{\rho_i} (\sinh(\beta_i l_i) + \sin(\beta_i l_i)) - \frac{(MO)_i \beta_i^2}{\rho_i} (\cosh(\beta_i l_i) - \cos(\beta_i l_i))}{\sinh(\beta_i l_i) + \sin(\beta_i l_i) - \frac{J_{Li} \beta_i^3}{\rho_i} (\cosh(\beta_i l_i) - \cos(\beta_i l_i)) - \frac{(MO)_i \beta_i^2}{\rho_i} (\sinh(\beta_i l_i) - \sin(\beta_i l_i))} \quad (2.34)$$

$$\frac{C_{2,i}}{C_{1,i}} = \frac{\sinh(\beta_i l_i) - \sin(\beta_i l_i) + \frac{m_{Li} \beta_i}{\rho_i} (\cosh(\beta_i l_i) - \cos(\beta_i l_i)) + \frac{(MO)_i \beta_i^2}{\rho_i} (\sinh(\beta_i l_i) + \sin(\beta_i l_i))}{\cosh(\beta_i l_i) + \cos(\beta_i l_i) + \frac{m_{Li} \beta_i}{\rho_i} (\sinh(\beta_i l_i) - \sin(\beta_i l_i)) + \frac{(MO)_i \beta_i^2}{\rho_i} (\cosh(\beta_i l_i) - \cos(\beta_i l_i))} \quad (2.35)$$

Substituting equation (2.34) into equation (2.35), we obtain the frequency equation

$$\begin{aligned} & 1 + \cos(\beta_i l_i) \cosh(\beta_i l_i) \\ & - \frac{m_{Li} \beta_i}{\rho_i} (\sin(\beta_i l_i) \cosh(\beta_i l_i) - \cos(\beta_i l_i) \sinh(\beta_i l_i)) \\ & - \frac{2}{\rho_i} (MO)_i \beta_i^2 \sin(\beta_i l_i) \sinh(\beta_i l_i) \\ & - \frac{J_{Li} \beta_i^3}{\rho_i} (\sin(\beta_i l_i) \cosh(\beta_i l_i) + \cos(\beta_i l_i) \sinh(\beta_i l_i)) \\ & + \frac{(m_{Li} J_{Li} - (MO)_i^2) \beta_i^4}{\rho^2} (1 - \cos(\beta_i l_i) \cosh(\beta_i l_i)) = 0 \end{aligned} \quad (2.36)$$

This frequency equation has an infinite number of solutions, $\{\beta_{ij}, j = 1, \dots, \infty\}$. Each solution β_{ij} is related to one natural frequency ω_{ij} . The natural frequencies are computed as

$$\omega_{ij} = \beta_{ij}^2 \sqrt{\frac{E_i I_i}{\rho_i}} \quad (2.37)$$

Substituting equation (2.32) into equation (2.31) and using equation (2.34) yield the corresponding mode shape function for the j th mode for link i :

$$\phi_{ij}(x_i) = C_{1,ij} \left[\cosh(\beta_{ij}x_i) - \cos(\beta_{ij}x_i) + \frac{C_{2,ij}}{C_{1,ij}} (\sinh(\beta_{ij}x_i) - \sin(\beta_{ij}x_i)) \right] \quad (2.38)$$

$$\frac{C_{2,ij}}{C_{1,ij}} = \frac{\cosh(\beta_{ij}l_i) + \cos(\beta_{ij}l_i) - \frac{J_{Li}\beta_{ij}^3}{\rho_i} (\sinh(\beta_{ij}l_i) + \sin(\beta_{ij}l_i)) - \frac{(MO)_i\beta_{ij}^2}{\rho_i} (\cosh(\beta_{ij}l_i) - \cos(\beta_{ij}l_i))}{\sinh(\beta_{ij}l_i) + \sin(\beta_{ij}l_i) - \frac{J_{Li}\beta_{ij}^3}{\rho_i} (\cosh(\beta_{ij}l_i) - \cos(\beta_{ij}l_i)) - \frac{(MO)_i\beta_{ij}^2}{\rho_i} (\sinh(\beta_{ij}l_i) - \sin(\beta_{ij}l_i))} \quad (2.39)$$

Let $\phi_{ij}(x_i)$ and $\phi_{ik}(x_i)$ be the mode shape functions corresponding to the natural frequencies ω_{ij} and ω_{ik} ($j \neq k$). From equation (2.29), we have

$$\frac{E_i I_i}{\rho_i} \phi_{ij}''''(x_i) = \omega_{ij}^2 \phi_{ij}(x_i) \quad (2.40)$$

$$\frac{E_i I_i}{\rho_i} \phi_{ik}''''(x_i) = \omega_{ik}^2 \phi_{ik}(x_i) \quad (2.41)$$

From integration by parts, one has

$$\begin{aligned} & \int_0^{l_i} \phi_{ij}(x_i) \phi_{ik}''''(x_i) dx_i \\ &= \left[\phi_{ij}(x_i) \phi_{ik}'''(x_i) - \phi_{ij}'(x_i) \phi_{ik}''(x_i) \right]_0^{l_i} + \int_0^{l_i} \phi_{ij}''(x_i) \phi_{ik}''(x_i) dx_i \end{aligned} \quad (2.42)$$

Multiplying equation (2.40) by $\phi_{ik}(x_i)$; equation (2.41) by $\phi_{ij}(x_i)$, subtracting the resulting equations one from the other, and integrating from 0 to l_i gives

$$\begin{aligned} & \int_0^{l_i} \phi_{ij}(x_i) \phi_{ik}(x_i) dx_i \\ &= \frac{E_i I_i}{\rho_i (\omega_{ij}^2 - \omega_{ik}^2)} \int_0^{l_i} (\phi_{ik}(x_i) \phi_{ij}''''(x_i) - \phi_{ij}(x_i) \phi_{ik}''''(x_i)) dx_i \\ &= \frac{1}{(\beta_{ij}^4 - \beta_{ik}^4)} \int_0^{l_i} (\phi_{ik}(x_i) \phi_{ij}''''(x_i) - \phi_{ij}(x_i) \phi_{ik}''''(x_i)) dx_i \end{aligned} \quad (2.43)$$

The right-hand side of equation (2.43) can be evaluated using equation (2.42) as

$$\int_0^{l_i} \phi_{ij}(x_i) \phi_{ik}(x_i) dx_i = \frac{E_i I_i}{\rho_i (\omega_{ij}^2 - \omega_{ik}^2)} \left(\phi_{ik}(x_i) \phi_{ij}'''(x_i) - \phi_{ij}(x_i) \phi_{ik}'''(x_i) + \phi_{ij}'(x_i) \phi_{ik}''(x_i) - \phi_{ik}'(x_i) \phi_{ij}''(x_i) \right) \Big|_0^{l_i} \quad (2.44)$$

Substituting equation (2.33) and (2.37) into equation (2.44), we obtain the orthogonality relations for the clamped-mass beam mode shape functions:

$$\int_0^{l_i} \phi_{ij}(x_i) \phi_{ik}(x_i) \rho_i dx_i + m_{Li} \phi_{ij}(l_i) \phi_{ik}(l_i) + (MO)_i (\phi_{ij}(l_i) \phi_{ik}'(l_i) + \phi_{ik}(l_i) \phi_{ij}'(l_i)) + J_{Li} \phi_{ij}'(l_i) \phi_{ik}'(l_i) = 0 \quad (j \neq k) \quad (2.45)$$

The constant $C_{1,ij}$ normalizes the mode shape functions so that

$$\int_0^{l_i} \phi_{ij}^2(x_i) dx_i = l_i \quad (2.46)$$

In this manner, one obtains an infinite set of mode shape function $\phi_{ij}(x_i)$ for link i and the corresponding natural frequencies ω_{ij} . Hence, according to the solution in equation (2.28), there is a corresponding infinite set of generalized coordinates, $\delta_{ij}(t)$, $j = 1, \dots, \infty$, which satisfy equation (2.29). It follows that the overall response of the link i is

$$w_i(x_i, t) = \sum_{j=1}^{\infty} \phi_{ij}(x_i) \delta_{ij}(t) \quad (2.47)$$

2.2.4 The Assumed Mode Method

From Section 2.2.3 we notice that the flexible links of a robotic manipulator are distributed parameter systems whose equations of motion may be described by a set of Bernoulli-Euler partial differential equations (PDE). Equation (2.47) states that the exact solution to the Bernoulli-Euler PDE requires an infinite number of modes. For typical applications, however, the higher order modes may be truncated since they represent frequencies that are well outside the operating bandwidth (De Silva, 2006). Furthermore, this modal truncation is required in order to meet the computation time constraints. A finite-dimensional model with link flexibility can be obtained by the assumed mode approach. The deflection of each link can be approximated by using a finite number of modes, and is expressed by a weighted summation

of the functions $\varphi_{ij}(x_i)$; thus

$$w_i(x_i, t) = \sum_{j=1}^{n_{m,i}} \varphi_{ij}(x_i) \delta_{ij}(t) \quad (2.48)$$

where $\varphi_{ij}(x_i)$ are any functions that satisfy the boundary conditions. These are termed admissible functions. If the functions $\varphi_{ij}(x_i)$ are selected to be a set of polynomials in x_i , the resulting frequencies of vibration are only approximations to the actual natural frequencies of the system. If the functions $\varphi_{ij}(x_i)$ are chosen as the mode shape functions $\phi_{ij}(x_i)$, the derived frequencies of vibration are equal the natural frequencies of the system. Therefore, the deflection of link i of the flexible-link robotic manipulator can be approximated by

$$w_i(x_i, t) = \sum_{j=1}^{n_{m,i}} \phi_{ij}(x_i) \delta_{ij}(t) \quad (2.49)$$

For the flexible link robotic manipulator, the number of modes $n_{m,i}$ and the admissible functions $\varphi_{ij}(x_i)$ determine the accuracy of the approximation. If we increase the number of modes $n_{m,i}$ the accuracy of the finite-dimensional model will be improved. But this will also increase the complexity of the analysis and computation. The high-order model will make controller design more difficult. In practice, as noted before, the contribution of the higher order modes to $w_i(x_i, t)$ is minimal, and $n_{m,i}$ is typically assumed to be 1 or 2. This corresponds to 1 or 2 significant modes. For the same number of modes, the accuracy of the assumed mode approximation is determined by the admissible functions $\varphi_{ij}(x_i)$. The goal is to select the admissible functions as close as possible to the exact mode shape functions $\phi_{ij}(x_i)$ without increasing the complexity of the system model. From equations (2.36) through (2.39), we can see that the mode shape functions $\phi_{ij}(x_i)$ are influenced by the physical parameters of the system, the boundary conditions, the payload, and the arm configuration. Next we will investigate the influence of these factors on the prototype flexible link manipulator shown in Figure 1.1. The schematic diagram of this planar two-link flexible manipulator is shown in Figure 2.1(a). The links of this experimental manipulator can be interchanged. Two types of manipulator links are used in this research. Type 1 link is a 1 m

long, 0.003 m thick, and 0.051 m tall aluminum thin beam. Type 2 link is an aluminum rod that is 2.0 cm in diameter and 1.0 m long. We will use the following configuration: the first (inboard) link is type 2 aluminum rod and the second (outboard) link is type 1 aluminum thin beam. Their physical parameters are given in Table 2.1.

For link 1, the equivalent moment of inertia J_{L1} , mass m_{L1} at the end of the link 1, and the contributions of the masses of payload and link 2 due to the mechanical offset $(MO)_1$ are

$$\begin{aligned} M_{L1} &= m_2 + m_{h2} + m_p \\ J_{L1} &= J_{o2} + J_{h2} + J_p + m_p l_2^2 \\ (MO)_1 &= (m_2 d_2 + m_p l_2) \cos(\theta_2) \end{aligned} \quad (2.50)$$

For link 2, the equivalent moment of inertia J_{L2} , mass m_{L2} at the end of the link 2, and the contributions of the masses of payload due to the mechanical offset $(MO)_2$ are

$$\begin{aligned} M_{L2} &= m_p \\ J_{L2} &= J_p \\ (MO)_2 &= 0 \end{aligned} \quad (2.51)$$

where m_2 is the mass of link 2, m_{h2} is the mass of hub 2, m_p is the payload mass, J_{o2} is mass moment of inertia of link 2 about joint 2 axis, J_{h2} is the inertia of hub 2, J_p is the payload inertia, l_2 is link length, d_2 is the center of mass, and θ_2 is the joint angle of link 2.

Table 2.2 and Table 2.3 present the natural frequencies and the link end deflections of the first four modes of link 1 and link 2 under different boundary conditions, payloads and arm configurations.

The exact mode shape functions for link 1 is time-varying and depend on the mechanical offset $(MO)_1$ given in equation (2.50). The maximum value of $(MO)_1$ is $(MO)_{1\max} = m_2 d_2 + m_p l_2$, which occurs when $\theta_2 = \pm\pi$. The minimum value of $(MO)_1$ is $(MO)_{1\min} = 0$, which occurs when $\theta_2 = \pm\frac{\pi}{2}$. Figure 2.4 shows the first four mode shapes of link 1 with zero and maximum mechanical offsets and nominal payload. We can see that changes of mechanical offset do not significantly affect the mode shapes of link 1. Since the mechanical offset of link 2 is zero, we will ignore the influence of mechanical offset from

now on.

Figure 2.5 shows the first four mode shapes of link 1 with clamped-free boundary conditions and clamped-mass boundary conditions with different payloads (zero, nominal, maximum payload) and zero mechanical offset. We can see that clamped-free mode shapes can only be used to approximate the first mode of link 1. For higher modes the differences between clamped-free and clamped-mass mode shapes are too excessive. Furthermore, the clamped-mass mode shapes of link 1 do not significantly change with the payload.

Figure 2.6 shows the first four mode shapes of link 2 with clamped-free boundary conditions and clamped-mass boundary conditions with different payloads (nominal, maximum payload) and zero mechanical offset. We can see that like in the case in link 1, only the first clamped-free mode shape is close to the clamped-mass mode shape for link 2. The effect of payload on mode shapes of link 2 is greater than that of link 1.

In summary, we can make following conclusions for the mode shapes of the two-link flexible robotic manipulator:

- The mode shapes and natural frequencies are dependent on the physical parameters of the link, the payload, and the arm configuration. Because of link flexibility, the resonant frequency is dramatically reduced. This means that the robot has to move relatively slowly to reduce vibration under normal operation. Active vibration controller must be designed and implemented to achieve better performance.
- In general, the mode shapes for link 1 are time-varying. They change with the position of link 2. But the influence of the mechanical offset is small and can be ignored for both links. Accordingly the constant mode shape functions may be used in the assumed mode method.
- Only the first one or two modes of motion are significant for both links. This is verified using the frequencies given in Table 2.2 and Table 2.3. The high frequency terms in the system dynamics can be dropped in order to reduce the model order. These terms may be treated as model uncertainties in the controller design.
- The larger the payload the smaller the natural frequencies.
- For the first mode, the difference between the clamped-free and clamped-mass mode shapes is small. For higher modes, the difference cannot be ignored. In order

to achieve the same model accuracy using the assumed mode method, the number of modes has to be increased if we use the clamped-free mode shapes.

- The effect of payload on the mode shapes of link 2 is greater than those of link 1. The effect of payload on mode shapes is not significant and can be ignored.
- The deflections of a flexible link robotic manipulator may be approximated by the assumed mode method. The constant mode shapes with zero mechanical offset and nominal payload, will be used in the assumed mode method.

Table 2.1 Parameters of the links.

	Link 1: rod	Link 2: thin beam
Uniform density: ρ (kg/m)	0.8451	0.4035
Link length: l (m)	1	1
Center of mass: d (m)	0.5	0.5
Link mass: m (kg)	0.8451	0.4035
Nominal payload mass: m_p (kg)	0.38	0.38
Maximum payload mass: $m_{p,max}$ (kg)	1	1
Link 1 hub mass: m_{h1} (kg)	6.4	6.4
Link 2 hub mass: m_{h2} (kg)	4.2	4.2
Link moment of inertia: J_o (kg.m ²)	0.2817	0.1345
Hub 1 moment of inertia: J_{h1} (kg.m ²)	4.5	4.5
Hub 2 moment of inertia: J_{h2} (kg.m ²)	1.9	1.9
Nominal payload inertia: J_p (kg.m ²)	0.001	0.001
Maximum payload inertia: $J_{p,max}$ (kg.m ²)	0.0026	0.0026
Link flexural rigidity: EI (N.m ²)	574.024	8.2238

Table 2.2 First four modes of link 1.

Link 1 (rod)		Mode (<i>j</i>)	1	2	3	4
	Offset (kg.m)	Payload (kg)				
β_{1j}	Clamped-free		1.8751	4.6941	7.8548	10.9955
	0	0	0.7065	1.3602	4.7697	7.8769
	0	Nominal	0.6837	1.3221	4.7664	7.8750
	Max	Nominal	0.6585	1.3803	4.7709	7.8770
	0	Max	0.6529	1.2719	4.7622	7.8725
f_{1j} (Hz)	Clamped-free		14.5916	91.4438	256.0453	501.7467
	0	0	2.0719	7.6786	94.4118	257.4884
	0	Nominal	1.9397	7.2546	94.2847	257.3687
	Max	Nominal	1.7996	7.9078	94.4622	257.4955
	0	Max	1.7691	6.7139	94.1190	257.2098
$\phi_{1j}(l_1)$ (m)	Clamped-free		2.0000	-2.0000	2.0000	- 2.0000
	0	0	2.1493	1.3660	-0.0736	0.0458
	0	Nominal	2.1540	1.3898	-0.0682	0.0423
	Max	Nominal	2.1516	1.3769	-0.0737	0.0449
	0	Max	2.1596	1.4163	-0.0609	0.0378

Table 2.3 First four modes of link 2.

Link 2 (thin beam)	Mode (<i>j</i>)	1	2	3	4
	Payload (kg)				
β_{2j}	0	1.8751	4.6941	7.8548	10.9955
	Nominal	1.2616	3.9334	6.5820	8.8400
	Max	1.0240	3.7406	5.9522	8.2571
f_{2j} (Hz)	0	2.5276	15.8401	44.3526	86.9134
	Nominal	1.1442	11.1224	31.1435	56.1767
	Max	0.7537	10.0587	25.4686	49.0122
$\phi_{2j}(l_2)$ (m)	0	2.0000	- 2.0000	2.0000	- 2.0000
	Nominal	2.0486	-0.3124	0.0868	0.1267
	Max	2.0554	-0.1131	-0.0459	0.0888

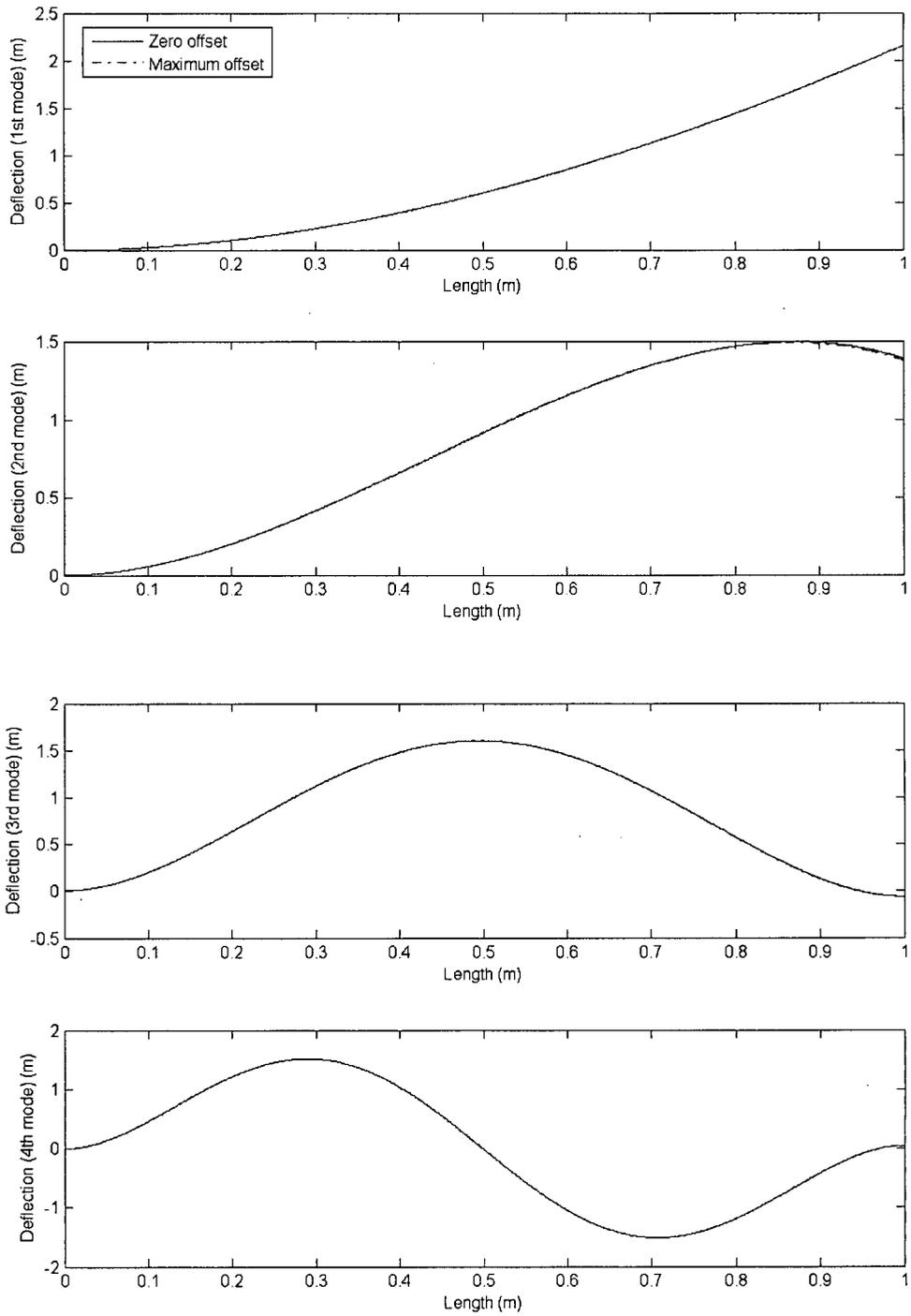


Figure 2.4 Mode shapes of link 1 with zero and maximum mechanical offsets.

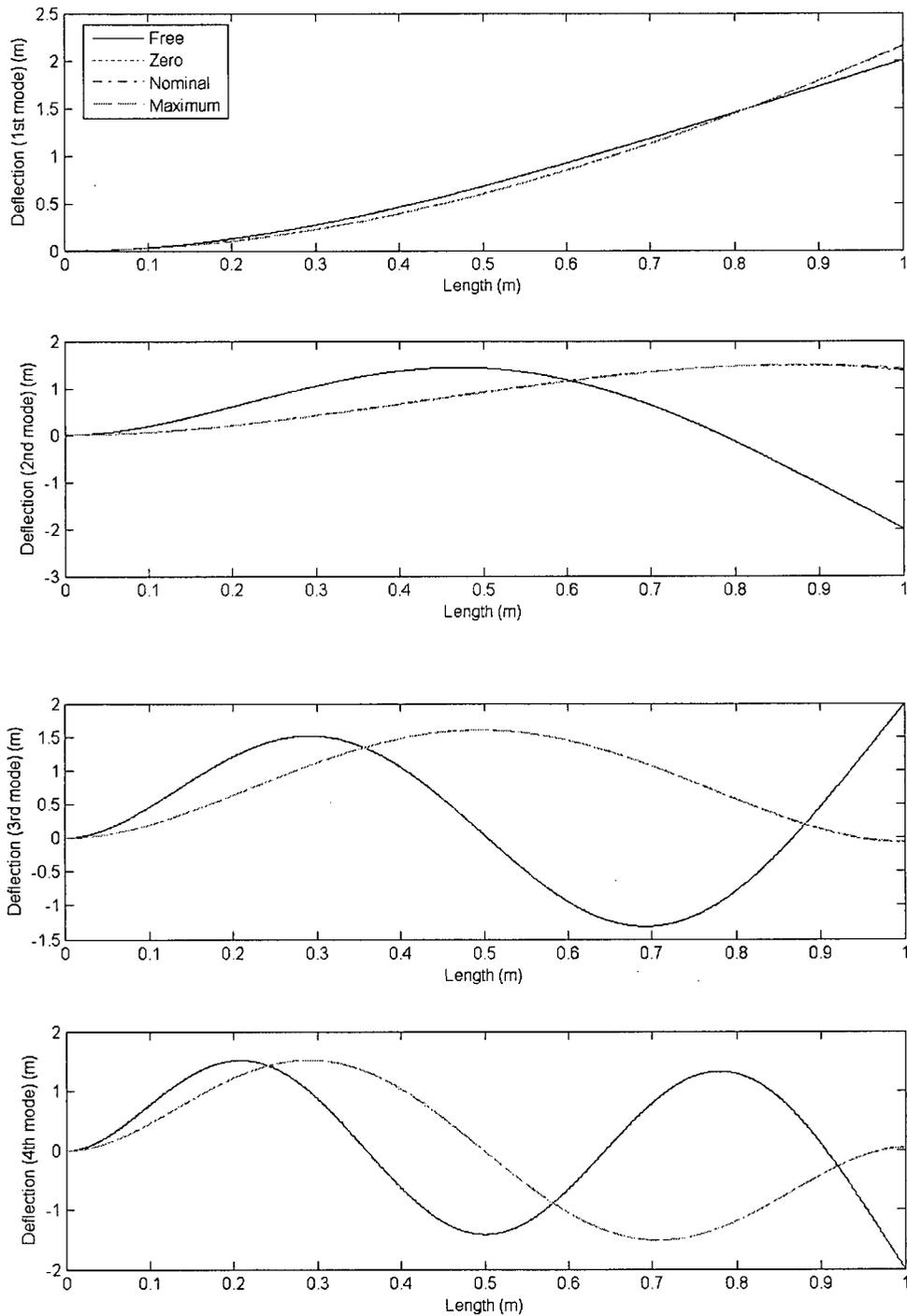


Figure 2.5 Mode shapes of link 1 with different payloads (zero, nominal, maximum) compared to a free-end link.

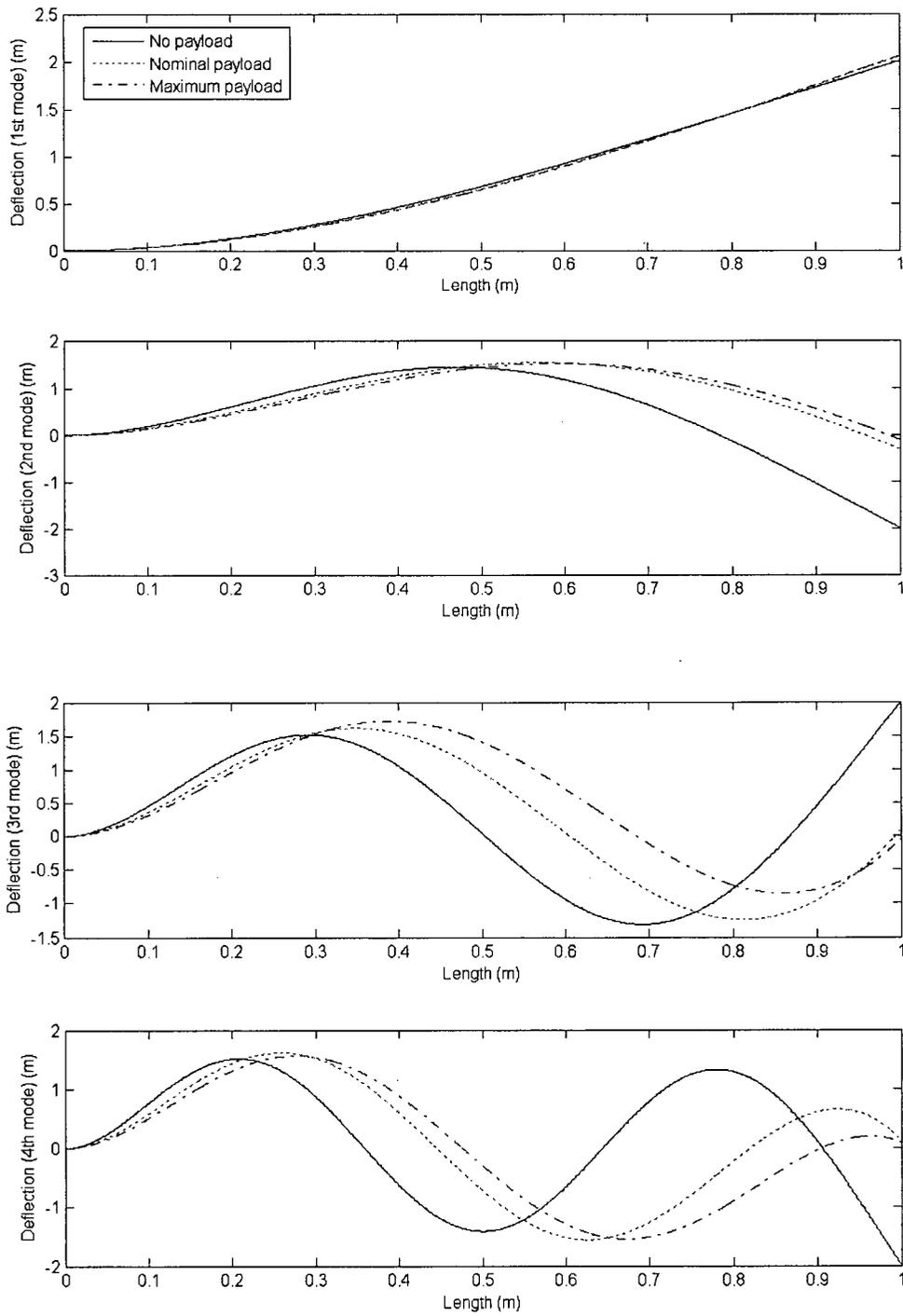


Figure 2.6 Mode shapes of link 2 with different payloads.

2.3 Derivation of Equations of Motion

2.3.1 Closed-form n -link Model

The dynamic equations of a planar n -link flexible robotic manipulator shown in Figure 2.1(b) can be derived using the well known energy method of Lagrangian dynamics. First we need to calculate the kinetic energy T and potential energy V of the system and then form the Lagrangian $L = T - V$. On the basis of the discretization introduced in Section 2.2.4, the Lagrangian L becomes a function of a set of N generalized coordinates $\{q_i(t)\}$, and the dynamic model is obtained by satisfying the Lagrange's equations

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} + \frac{\partial R}{\partial \dot{q}_i} = f_i, \quad i = 1, \dots, N. \quad (2.52)$$

where $\{f_i(t)\}$ are the generalized forces and R is the dissipation function.

For a point \mathbf{r}_i on link i in frame F^i one has

$$\mathbf{r}_i^i(t) = [x_i \quad w_i(x_i, t) \quad 0 \quad 1]^T \quad (2.53)$$

The link deflection can be approximated by the assumed mode method as given in equation (2.48). Based on the analysis in Section 2.24, we will use the constant clamped-mass mode shape functions with zero mechanical offset and nominal payload as admissible functions. Then

$$w_i(x_i, t) = \sum_{j=1}^{n_{m,i}} \phi_{ij}(x_i) \delta_{ij}(t) \quad (2.54)$$

Set $(MO)_i = 0$ in equation (2.38) and (2.39). We obtain the clamped-mass mode shape function for the j th mode for link i as

$$\phi_{ij}(x_i) = C_{1,ij} \left[\cosh(\beta_{ij}x_i) - \cos(\beta_{ij}x_i) + \frac{C_{2,ij}}{C_{1,ij}} (\sinh(\beta_{ij}x_i) - \sin(\beta_{ij}x_i)) \right]$$

$$\frac{C_{2,ij}}{C_{1,ij}} = - \frac{\cosh(\beta_{ij}l_i) + \cos(\beta_{ij}l_i) - \frac{J_{Li}\beta_{ij}^3}{\rho_i} (\sinh(\beta_{ij}l_i) + \sin(\beta_{ij}l_i))}{\sinh(\beta_{ij}l_i) + \sin(\beta_{ij}l_i) - \frac{J_{Li}\beta_{ij}^3}{\rho_i} (\cosh(\beta_{ij}l_i) - \cos(\beta_{ij}l_i))} \quad (2.55)$$

From equation (2.36), we have the corresponding frequency equation

$$1 + \cos(\beta_i l_i) \cosh(\beta_i l_i) - \frac{m_{Li}\beta_i}{\rho_i} (\sin(\beta_i l_i) \cosh(\beta_i l_i) - \cos(\beta_i l_i) \sinh(\beta_i l_i)) - \frac{J_{Li}\beta_i^3}{\rho_i} (\sin(\beta_i l_i) \cosh(\beta_i l_i) + \cos(\beta_i l_i) \sinh(\beta_i l_i)) + \frac{m_{Li}J_{Li}\beta_i^4}{\rho^2} (1 - \cos(\beta_i l_i) \cosh(\beta_i l_i)) = 0 \quad (2.56)$$

The position vector $\mathbf{r}_i^0(t)$, the velocity vector $\mathbf{v}_i^0(t)$, and the absolute angular velocity $\dot{\theta}_i^0$ (equations (2.8) through (2.10)) in Section 2.1 can be used to form the kinetic energy, potential energy, and dissipation function expressions for the flexible link manipulator.

The kinetic energy of the hub i is

$$T_{hi} = \frac{1}{2} m_{hi} (\mathbf{v}_i^0)^T \mathbf{v}_i^0 \Big|_{x_i=0} + \frac{1}{2} J_{hi} (\dot{\theta}_i^0)^2 \quad (2.57)$$

where m_{hi} is the hub mass and J_{hi} is the hub moment of inertia.

The kinetic energy of link i is

$$T_{li} = \frac{1}{2} \int_0^{l_i} \rho_i(x_i) (\mathbf{v}_i^0)^T \mathbf{v}_i^0 dx_i \quad (2.58)$$

The kinetic energy of a payload with mass m_p and moment of inertia J_p located at the end of link n is

$$T_p = \frac{1}{2} m_p (\mathbf{v}_n^0)^T \mathbf{v}_n^0 \Big|_{x_n=l_n} + \frac{1}{2} J_p (\dot{\theta}_n^0 + \dot{\alpha}_n)^2 \quad (2.59)$$

The total kinetic energy of the system is

$$T = \sum_{i=1}^n T_{hi} + \sum_{i=1}^n T_{li} + T_p \quad (2.60)$$

The total potential energy of the system is given by

$$V = \sum_{i=1}^n U_i = \sum_{i=1}^n \frac{1}{2} \int_0^{l_i} (EI)_i(x_i) \left[\frac{\partial^2 w_i(x_i, t)}{\partial x_i^2} \right]^2 dx_i \quad (2.61)$$

where U_i is the elastic potential energy of link i .

Assuming that the energy dissipation in the system can be modeled using equivalent viscous damping, the total dissipation function of the system is

$$R = \frac{1}{2} \sum_{i=1}^n b_{\theta_i} \dot{\theta}_i^2 + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^{n_{m,i}} b_{ij} \dot{\delta}_{ij}^2 \quad (2.62)$$

where the subscripted b terms are equivalent viscous damping coefficients. The first term represents the energy dissipated due to friction in the link joints and the second term is the energy dissipated due to structural damping.

Substituting equations (2.60) through (2.62) into equation (2.52), the closed-form equations of motion of a planar n -link flexible manipulator are obtained as

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{K}\mathbf{q} + \mathbf{B}\dot{\mathbf{q}} = \mathbf{u} \quad (2.63)$$

where $\mathbf{q} = (\theta_1, \dots, \theta_n, \delta_{11}, \dots, \delta_{1n_{m,1}}, \dots, \delta_{n1}, \dots, \delta_{nn_{m,n}})$ are the generalized coordinates, θ_i are the joint variables of link i , δ_{ij} are the deflection variables, \mathbf{M} is the positive-definite symmetric mass matrix, \mathbf{h} is the vector of Coriolis and centrifugal forces, \mathbf{K} is the stiffness matrix, \mathbf{B} is the diagonal damping matrix, and \mathbf{u} is the $n \times 1$ vector of input torques.

Since all the links are modeled by uniform Bernoulli-Euler beams having constant clamped-mass mode shapes with zero mechanical offsets and nominal payload, the following integrals are constant parameters in the equations of motion and can be computed offline:

$$m_i = \int_0^{l_i} \rho_i dx_i = \rho_i l_i \quad (2.64)$$

$$J_{oi} = \int_0^{l_i} \rho_i x_i^2 dx_i \quad (2.65)$$

$$I_{1ij} = \int_0^{l_i} \rho_i \phi_{ij}(x_i) x_i dx_i \quad (2.66)$$

$$I_{2ijk} = \int_0^{l_i} \rho_i \phi_{ij}(x_i) \phi_{ik}(x_i) dx_i \quad (2.67)$$

$$I_{3ijk} = \int_0^{l_i} (EI)_i \phi_{ij}''(x_i) \phi_{ik}''(x_i) dx_i \quad (2.68)$$

$$I_{4ij} = \int_0^{l_i} \rho_i \phi_{ij}(x_i) dx_i \quad (2.69)$$

Recall the normalization of the clamped-mass mode shape functions in equation (2.46). We have

$$I_{2ijj} = \int_0^{l_i} \rho_i \phi_{ij}^2(x_i) dx_i = \rho_i l_i = m_i \quad (2.70)$$

The orthogonality relations for the clamped-mass beam mode shape functions in equation (2.45) with $MO_i = 0$ become

$$\int_0^{l_i} \phi_{ij}(x_i) \phi_{ik}(x_i) \rho_i dx_i + m_{Li} \phi_{ij}(l_i) \phi_{ik}(l_i) + J_{Li} \phi_{ij}'(l_i) \phi_{ik}'(l_i) = 0 \quad (j \neq k) \quad (2.71)$$

From equation (2.68) and (2.42) we have

$$\begin{aligned} I_{3ijk} &= \int_0^{l_i} (EI)_i \phi_{ij}''(x_i) \phi_{ik}''(x_i) dx_i = \\ & (EI)_i \left\{ \int_0^{l_i} \phi_{ij}(x_i) \phi_{ik}''''(x_i) dx_i - \left[\phi_{ij}(x_i) \phi_{ik}''''(x_i) - \phi_{ij}'(x_i) \phi_{ik}'''(x_i) \right] \Big|_0^{l_i} \right\} \end{aligned} \quad (2.72)$$

Substituting equation (2.41) into equation (2.72) and using the boundary conditions in equation (2.33), we obtain

$$\begin{aligned} I_{3ijk} &= \int_0^{l_i} (EI)_i \phi_{ij}''(x_i) \phi_{ik}''(x_i) dx_i = \\ & (EI)_i \left\{ \int_0^{l_i} \phi_{ij}(x_i) \frac{\rho_i}{(EI)_i} \omega_{ik}^2 \phi_{ik}(x_i) dx_i - \left[\phi_{ij}(x_i) \phi_{ik}''''(x_i) - \phi_{ij}'(x_i) \phi_{ik}'''(x_i) \right] \Big|_0^{l_i} \right\} \\ &= \omega_{ik}^2 \int_0^{l_i} \rho_i \phi_{ij}(x_i) \phi_{ik}(x_i) dx_i - (EI)_i \frac{\beta_{ik}^4}{\rho_i} \left[-m_{Li} \phi_{ij}(l_i) \phi_{ik}(l_i) - J_{Li} \phi_{ij}'(l_i) \phi_{ik}'(l_i) \right] \\ &= \omega_{ik}^2 \left[\int_0^{l_i} \rho_i \phi_{ij}(x_i) \phi_{ik}(x_i) dx_i + m_{Li} \phi_{ij}(l_i) \phi_{ik}(l_i) + J_{Li} \phi_{ij}'(l_i) \phi_{ik}'(l_i) \right] \end{aligned} \quad (2.73)$$

Substituting equation (2.70) and (2.71) into above equation we obtain

$$\begin{aligned}
 I_{3ijk} &= \int_0^{l_i} (EI)_i \phi_{ij}''(x_i) \phi_{ik}''(x_i) dx_i \\
 &= \begin{cases} 0 & (j \neq k) \\ \omega_{ij}^2 \left[m_i + m_{Li} \phi_{ij}(l_i)^2 + J_{Li} \phi_{ij}'(l_i)^2 \right] & (j = k) \end{cases} \quad (2.74)
 \end{aligned}$$

The orthogonality relations and normalization of the mode shape functions result in convenient simplifications in the diagonal blocks of the mass matrix \mathbf{M} relative to the deflections of each link, due to the particular values attained by I_{2ijk} in equation (2.67). From equation (2.74) it can be shown that the stiffness matrix becomes diagonal and take the values $\omega_{ij}^2 \left[m + m_{Li} \phi_{ij}(x_i)^2 + J_{Li} \phi_{ij}'(l_i)^2 \right]$.

2.3.2 Single-link Model

In this section, we develop an explicit finite-dimensional dynamic model for a planar single-link flexible robotic manipulator shown in Figure 2.7.

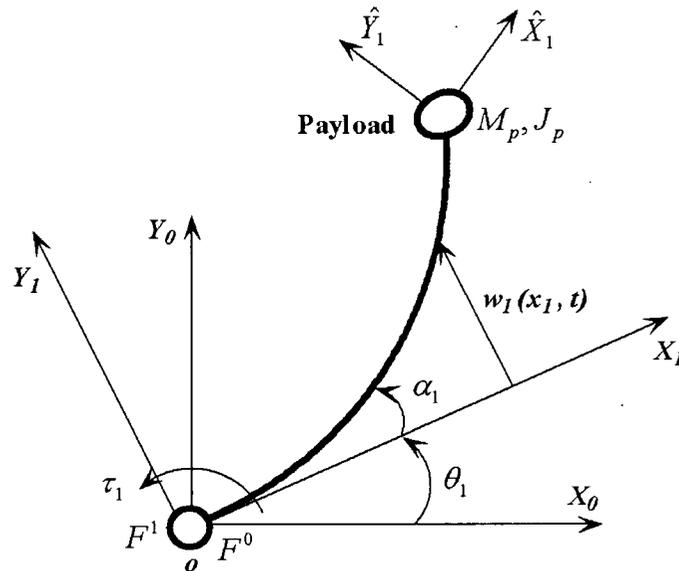


Figure 2.7 Single-link flexible robotic manipulator.

The link deflection can be obtained from equation (2.54) by setting $i = 1$; thus

$$w_1(x_1, t) = \sum_{j=1}^{n_{m,1}} \phi_{1j}(x_1) \delta_{1j}(t) \quad (2.75)$$

where $\phi_{1j}(x_1)$ are the constant clamped-mass mode shape functions with zero mechanical offset and nominal payload.

The kinetic energy of the hub is

$$T_{h1} = \frac{1}{2} J_{h1} \dot{\theta}_1^2 \quad (2.76)$$

The kinetic energy of link is

$$T_{l1} = \frac{1}{2} \int_0^{l_1} \rho_1 (\mathbf{v}_1^0)^T \mathbf{v}_1^0 dx_1 \quad (2.77)$$

Substituting equation (2.13) into equation (2.77), we have

$$\begin{aligned} T_{l1} &= \frac{1}{2} \int_0^{l_1} \rho_1 (\mathbf{v}_1^0)^T \mathbf{v}_1^0 dx_1 \\ &= \frac{1}{2} \int_0^{l_1} \rho_1 \left\{ [x_1 s_1 + c_1 w_1(x_1, t)] \dot{\theta}_1 + \dot{w}_1(x_1, t) s_1 \right\}^2 + \left\{ [x_1 c_1 - s_1 w_1(x_1, t)] \dot{\theta}_1 + \dot{w}_1(x_1, t) c_1 \right\}^2 dx_1 \quad (2.78) \\ &= \frac{1}{2} \int_0^{l_1} \rho_1 \left\{ [x_1^2 + w_1^2(x_1, t)] \dot{\theta}_1^2 + 2 \dot{\theta}_1 \dot{w}_1(x_1, t) x_1 + \dot{w}_1^2(x_1, t) \right\} dx_1 \end{aligned}$$

Substituting equation (2.75) into equation (2.78), we obtain

$$\begin{aligned} T_{l1} &= \frac{1}{2} \int_0^{l_1} \rho_1 \left\{ [x_1^2 + w_1^2(x_1, t)] \dot{\theta}_1^2 + 2 \dot{\theta}_1 \dot{w}_1(x_1, t) x_1 + \dot{w}_1^2(x_1, t) \right\} dx_1 \\ &= \frac{1}{2} \int_0^{l_1} \rho_1 \left\{ \left[x_1^2 + \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}(x_1) \delta_{1j}(t) \right)^2 \right] \dot{\theta}_1^2 + 2 \dot{\theta}_1 \sum_{j=1}^{n_{m,1}} \phi_{1j}(x_1) \dot{\delta}_{1j}(t) x_1 \right. \\ &\quad \left. + \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}(x_1) \dot{\delta}_{1j}(t) \right)^2 \right\} dx_1 \quad (2.79) \end{aligned}$$

The kinetic energy of payload is obtained by setting $i = 1$ in equation (2.59); thus

$$T_p = \frac{1}{2} m_p (\mathbf{v}_1^0)^T \mathbf{v}_1^0 \Big|_{x_1=l_1} + \frac{1}{2} J_p (\dot{\theta}_1^0 + \dot{\alpha}_1)^2 \quad (2.80)$$

Substituting equations (2.4), (2.10), and (2.13) into equation (2.80), we obtain

$$\begin{aligned}
T_p &= \frac{1}{2} m_p (\mathbf{v}_1^0)^T \mathbf{v}_1^0 \Big|_{x_1=l_1} + \frac{1}{2} J_p (\dot{\theta}_1^0 + \dot{\alpha}_1)^2 \\
&= \frac{1}{2} m_p \left\{ \left[l_1^2 + \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \delta_{1j}(t) \right)^2 \right] \dot{\theta}_1^2 + 2\dot{\theta}_1 \sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \dot{\delta}_{1j}(t) l_1 + \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \dot{\delta}_{1j}(t) \right)^2 \right\} \\
&\quad + \frac{1}{2} J_p \left(\dot{\theta}_1 + \sum_{j=1}^{n_{m,1}} \phi_{1j}'(l_1) \dot{\delta}_{1j}(t) \right)^2
\end{aligned} \tag{2.81}$$

The total kinetic energy of the system is

$$\begin{aligned}
T &= T_{h1} + T_{l1} + T_p \\
&= \frac{1}{2} J_{h1} \dot{\theta}_1^2 + \frac{1}{2} \int_0^{l_1} \rho_1 \left\{ \left[x_1^2 + \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}(x_1) \delta_{1j}(t) \right)^2 \right] \dot{\theta}_1^2 + 2\dot{\theta}_1 \sum_{j=1}^{n_{m,1}} \phi_{1j}(x_1) \dot{\delta}_{1j}(t) x_1 \right. \\
&\quad \left. + \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}(x_1) \dot{\delta}_{1j}(t) \right)^2 \right\} dx_1 \\
&\quad + \frac{1}{2} m_p \left\{ \left[l_1^2 + \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \delta_{1j}(t) \right)^2 \right] \dot{\theta}_1^2 + 2\dot{\theta}_1 \sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \dot{\delta}_{1j}(t) l_1 \right. \\
&\quad \left. + \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \dot{\delta}_{1j}(t) \right)^2 \right\} \\
&\quad + \frac{1}{2} J_p \left(\dot{\theta}_1 + \sum_{j=1}^{n_{m,1}} \phi_{1j}'(l_1) \dot{\delta}_{1j}(t) \right)^2
\end{aligned} \tag{2.82}$$

The total potential energy of the system is given by setting $i = 1$ in equation (2.61); thus

$$V = \frac{1}{2} \int_0^{l_1} (EI)_1 \left[\frac{\partial^2 w_1(x_1, t)}{\partial x_1^2} \right]^2 dx_1 \tag{2.83}$$

Substituting equation (2.75) into equation (2.83), we obtain

$$V = \frac{1}{2} \int_0^{l_1} (EI)_1 \left[\sum_{j=1}^{n_{m,1}} \phi_{1j}''(x_1) \delta_{1j}(t) \right]^2 dx_1 \tag{2.84}$$

Use equations (2.82) and (2.84) to form the Lagrangian of the system:

$$L = T - V \tag{2.85}$$

From equation (2.62), the total energy dissipation function of the system is obtained as

$$R = \frac{1}{2} b_{\theta_1} \dot{\theta}_1^2 + \frac{1}{2} \sum_{j=1}^{n_{m,1}} b_{1j} \dot{\delta}_{1j}^2 \quad (2.86)$$

The dynamic model of the single-link flexible robotic manipulator is obtained by substituting equations (2.85) and (2.86) into the Lagrange's equations

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} + \frac{\partial R}{\partial \dot{q}_i} = f_i, \quad i = 1, \dots, 1 + n_{m,1}. \quad (2.87)$$

The generalized coordinates of the system are $\mathbf{q} = (q_1, \dots, q_{1+n_{m,1}}) = (\theta_1, \delta_{11}, \dots, \delta_{1n_{m,1}})$. One obtains

$$\begin{aligned} \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_1} &= J_{h1} \ddot{\theta}_1 + \left[\int_0^l \rho_1 x_1^2 dx_1 + \int_0^l \rho_1 \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}(x_1) \delta_{1j}(t) \right)^2 dx_1 \right] \ddot{\theta}_1 \\ &+ 2\rho_1 \dot{\theta}_1 \left[\int_0^l \sum_{j=1}^{n_{m,1}} \phi_{1j}(x_1) \delta_{1j}(t) \sum_{j=1}^{n_{m,1}} \phi_{1j}(x_1) \dot{\delta}_{1j}(t) dx_1 \right] \\ &+ \int_0^l \rho_1 \sum_{j=1}^{n_{m,1}} \phi_{1j}(x_1) x_1 dx_1 \ddot{\delta}_{1j}(t) + m_p \left[l_1^2 + \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \delta_{1j}(t) \right)^2 \right] \ddot{\theta}_1 \\ &+ 2m_p \dot{\theta}_1 \left[\sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \delta_{1j}(t) \sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \dot{\delta}_{1j}(t) \right] + m_p \sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \ddot{\delta}_{1j}(t) l_1 \\ &+ J_p \ddot{\theta}_1 + \sum_{j=1}^{n_{m,1}} J_p \phi_{1j}'(l_1) \ddot{\delta}_{1j}(t) \end{aligned} \quad (2.88)$$

$$\begin{aligned} \frac{\partial L}{\partial \theta_1} &= 0 \\ \frac{\partial R}{\partial \dot{\theta}_1} &= b_{\theta_1} \dot{\theta}_1 \end{aligned} \quad (2.89)$$

$$\begin{aligned} \frac{d}{dt} \frac{\partial L}{\partial \dot{\delta}_{1j}} &= \int_0^l \rho_1 \phi_{1j}(x_1) x_1 dx_1 \ddot{\theta}_1 + \int_0^l \rho_1 \phi_{1j}(x_1) \sum_{k=1}^{n_{m,1}} \phi_{1k}(x_1) \ddot{\delta}_{1k}(t) dx_1 \\ &+ \left(m_p l_1 \phi_{1j}(l_1) + J_p \phi_{1j}'(l_1) \right) \ddot{\theta}_1 \\ &+ m_p \phi_{1j}(l_1) \sum_{k=1}^{n_{m,1}} \phi_{1k}(l_1) \ddot{\delta}_{1k}(t) + J_p \phi_{1j}'(l_1) \sum_{k=1}^{n_{m,1}} \phi_{1k}'(l_1) \ddot{\delta}_{1k}(t) \end{aligned} \quad (2.90)$$

$$\begin{aligned}
-\frac{\partial L}{\partial \delta_{1j}} &= -\int_0^{l_1} \rho_1 \phi_{1j}(x_1) \sum_{k=1}^{n_{m,1}} \phi_{1k}(x_1) \delta_{1k}(t) dx_1 \dot{\theta}_1^2 \\
&\quad - m_p \phi_{1j}(l_1) \sum_{k=1}^{n_{m,1}} \phi_{1k}(l_1) \delta_{1k}(t) dx_1 \dot{\theta}_1^2 \\
&\quad + (EI)_1 \int_0^{l_1} \phi_{1j}''(x_1) \sum_{k=1}^{n_{m,1}} \phi_{1k}''(x_1) \delta_{1k}(t) dx_1 \\
\frac{\partial R}{\partial \dot{\delta}_{1j}} &= b_{1j} \dot{\delta}_{1j}
\end{aligned} \tag{2.91}$$

For the θ_1 generalized coordinate the generalized force is the torque applied to the joint τ_1 . The rigid body equation for the link is obtained by substituting equations (2.88) and (2.89) into (2.87) and using the constant integrals defined in equation (2.64)-(2.70); thus

$$\begin{aligned}
&\ddot{\theta}_1 (J_{h1} + J_{o1} + J_p + m_p l_1^2) \\
&+ \ddot{\theta}_1 \left[\int_0^{l_1} \rho_1 \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}(x_1) \delta_{1j}(t) \right)^2 dx_1 + m_p \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \delta_{1j}(t) \right)^2 \right] \\
&+ \sum_{j=1}^{n_{m,1}} \left(I_{11j} + m_p l_1 \phi_{1j}(l_1) + J_p \phi_{1j}'(l_1) \right) \ddot{\delta}_{1j}(t) \\
&+ 2\dot{\theta}_1 \left[\int_0^{l_1} \rho_1 \sum_{j=1}^{n_{m,1}} \phi_{1j}(x_1) \delta_{1j}(t) \sum_{j=1}^{n_{m,1}} \phi_{1j}(x_1) \dot{\delta}_{1j}(t) dx_1 \right. \\
&\quad \left. + m_p \sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \delta_{1j}(t) \sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \dot{\delta}_{1j}(t) \right] \\
&+ b_{\theta_1} \dot{\theta}_1 = \tau_1
\end{aligned} \tag{2.92}$$

The time variation of the j th mode shape function of the link is given by the generalized coordinate δ_{1j} for which the corresponding generalized force is zero. The deflection equation for the j th mode is obtained by substituting equations (2.90) and (2.91) into (2.87) and using the constant integrals defined in equation (2.64)-(2.70); thus

$$\begin{aligned}
& \left(I_{11j} + m_p l_1 \phi_{1j}(l_1) + J_p \phi_{1j}'(l_1) \right) \ddot{\theta}_1 \\
& + \sum_{k=1}^{n_{m,1}} \left[\int_0^{l_1} \rho_1 \phi_{1j}(x_1) \phi_{1k}(x_1) dx_1 + m_p \phi_{1j}(l_1) \phi_{1k}(l_1) + J_p \phi_{1j}'(l_1) \phi_{1k}'(l_1) \right] \ddot{\delta}_{1k}(t) \\
& - \sum_{k=1}^{n_{m,1}} \left[\int_0^{l_1} \rho_1 \phi_{1j}(x_1) \phi_{1k}(x_1) dx_1 + m_p \phi_{1j}(l_1) \phi_{1k}(l_1) \right] \delta_{1k}(t) \dot{\theta}_1^2 \\
& + \omega_{1j}^2 \left[m_1 + m_p \phi_{1j}(l_1)^2 + J_p \phi_{1j}'(l_1)^2 \right] \delta_{1j}(t) + b_{1j} \dot{\delta}_{1j} = 0
\end{aligned} \tag{2.93}$$

2.3.3 Two-link Model

In this section we develop an explicit finite-dimensional dynamic model for the planar two-link flexible robotic manipulator shown in Figure 2.1(a).

The link deflection for the first link is

$$w_1(x_1, t) = \sum_{j=1}^{n_{m,1}} \phi_{1j}(x_1) \delta_{1j}(t) \tag{2.94}$$

The link deflection for the second link is

$$w_2(x_2, t) = \sum_{j=1}^{n_{m,2}} \phi_{2j}(x_2) \delta_{2j}(t) \tag{2.95}$$

where $\phi_{1j}(x_1)$ and $\phi_{2j}(x_2)$ are the constant clamped-mass mode shape functions with zero mechanical offset and nominal payload.

The kinetic energy of hub 1 is

$$T_{h1} = \frac{1}{2} J_{h1} \dot{\theta}_1^2 \tag{2.96}$$

The kinetic energy of hub 2 is

$$\begin{aligned}
T_{h2} &= \frac{1}{2} m_{h2} (\mathbf{v}_2^0)^T \mathbf{v}_2^0 \Big|_{x_2=0} + \frac{1}{2} J_{h2} (\dot{\theta}_2^0)^2 \\
&= \frac{1}{2} m_{h2} (\mathbf{v}_1^0)^T \mathbf{v}_1^0 \Big|_{x_1=l_1} + \frac{1}{2} J_{h2} (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\alpha}_1)^2 \\
&= \frac{1}{2} m_{h2} \left\{ \left[l_1^2 + \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \delta_{1j}(t) \right)^2 \right] \dot{\theta}_1^2 + 2\dot{\theta}_1 \sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \dot{\delta}_{1j}(t) l_1 + \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \dot{\delta}_{1j}(t) \right)^2 \right\} \quad (2.97) \\
&\quad + \frac{1}{2} J_{h2} \left(\dot{\theta}_1 + \dot{\theta}_2 + \sum_{j=1}^{n_{m,1}} \phi_{1j}'(l_1) \dot{\delta}_{1j}(t) \right)^2
\end{aligned}$$

The kinetic energy of link 1 is

$$\begin{aligned}
T_{l1} &= \frac{1}{2} \int_0^{l_1} \rho_1(x_1) (\mathbf{v}_1^0)^T \mathbf{v}_1^0 dx_1 \\
&= \frac{1}{2} \int_0^{l_1} \rho_1 \{ [x_1^2 + w_1^2(x_1, t)] \dot{\theta}_1^2 + 2\dot{\theta}_1 \dot{w}_1(x_1, t) x_1 + \dot{w}_1^2(x_1, t) \} dx_1 \quad (2.98) \\
&= \frac{1}{2} \int_0^{l_1} \rho_1 \left\{ \left[x_1^2 + \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}(x_1) \delta_{1j}(t) \right)^2 \right] \dot{\theta}_1^2 + 2\dot{\theta}_1 \sum_{j=1}^{n_{m,1}} \phi_{1j}(x_1) \dot{\delta}_{1j}(t) x_1 \right. \\
&\quad \left. + \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}(x_1) \dot{\delta}_{1j}(t) \right)^2 \right\} dx_1
\end{aligned}$$

The kinetic energy of link 2 is

$$\begin{aligned}
T_{l2} &= \frac{1}{2} \int_0^{l_2} \rho_2 (\mathbf{v}_2^0)^T \mathbf{v}_2^0 dx_2 \\
&= \frac{1}{2} \int_0^{l_2} \rho_2 \left(l_1^2 + w_1^2(l_1, t) \right) \dot{\theta}_1^2 + \left(x_2^2 + w_2^2(x_2, t) \right) (\dot{\theta}_1 + \dot{\alpha}_1 + \dot{\theta}_2)^2 + \dot{w}_1^2(l_1, t) + \dot{w}_2^2(x_2, t) dx_2 \\
&\quad + \frac{1}{2} \int_0^{l_2} \rho_2 (\dot{\theta}_1 + \dot{\alpha}_1 + \dot{\theta}_2) \left\{ \begin{aligned} &2x_2 \left[l_1 c_{\alpha_1,2} \dot{\theta}_1 + w_1(l_1, t) s_{\alpha_1,2} \dot{\theta}_1 + \dot{w}_1(l_1, t) c_{\alpha_1,2} \right] \\ &+ 2x_2 \dot{w}_2(x_2, t) - 2w_2(x_2, t) \left[l_1 s_{\alpha_1,2} \dot{\theta}_1 - w_1(l_1, t) c_{\alpha_1,2} \dot{\theta}_1 \right] \\ &+ \dot{w}_1(l_1, t) s_{\alpha_1,2} \end{aligned} \right\} dx_2 \\
&\quad + \frac{1}{2} \int_0^{l_2} \rho_2 2l_1 \dot{\theta}_1 \dot{w}_1(l_1, t) + 2\dot{w}_2(x_2, t) \left[l_1 c_{\alpha_1,2} \dot{\theta}_1 + w_1(l_1, t) s_{\alpha_1,2} \dot{\theta}_1 + \dot{w}_1(l_1, t) c_{\alpha_1,2} \right] dx_2
\end{aligned}$$

$$\begin{aligned}
&= \frac{1}{2} m_2 \left(l_1^2 + \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \delta_{1j}(t) \right)^2 \right) \dot{\theta}_1^2 \\
&+ \frac{1}{2} \left[J_{02} + \int_0^{l_2} \rho_2 \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(x_2) \delta_{2j}(t) \right)^2 dx_2 \right] (\dot{\theta}_1 + \sum_{j=1}^{n_{m,1}} \phi_{1j}'(l_1) \dot{\delta}_{1j}(t) + \dot{\theta}_2)^2 \\
&+ \frac{1}{2} m_2 \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \dot{\delta}_{1j}(t) \right)^2 + \frac{1}{2} \int_0^{l_2} \rho_2 \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(x_2) \dot{\delta}_{2j}(t) \right)^2 dx_2 \\
&+ (\dot{\theta}_1 + \sum_{j=1}^{n_{m,1}} \phi_{1j}'(l_1) \dot{\delta}_{1j}(t) + \dot{\theta}_2) \\
&\left\{ \begin{aligned} &\left[\begin{aligned} &l_1 \cos \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}'(l_1) \delta_{1j}(t) + \theta_2 \right) \dot{\theta}_1 \\ &+ \sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \delta_{1j}(t) \sin \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}'(l_1) \delta_{1j}(t) + \theta_2 \right) \dot{\theta}_1 + \sum_{j=1}^{n_{m,2}} I_{12j} \dot{\delta}_{2j}(t) \\ &+ \sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \dot{\delta}_{1j}(t) \cos \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}'(l_1) \delta_{1j}(t) + \theta_2 \right) \end{aligned} \right] \\ &- \sum_{j=1}^{n_{m,2}} I_{42j} \delta_{2j}(t) \left[\begin{aligned} &l_1 \sin \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}'(l_1) \delta_{1j}(t) + \theta_2 \right) \dot{\theta}_1 \\ &- \sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \delta_{1j}(t) \cos \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}'(l_1) \delta_{1j}(t) + \theta_2 \right) \dot{\theta}_1 \\ &+ \sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \dot{\delta}_{1j}(t) \sin \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}'(l_1) \delta_{1j}(t) + \theta_2 \right) \end{aligned} \right] \end{aligned} \right\} \\
&+ m_2 l_1 \dot{\theta}_1 \sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \dot{\delta}_{1j}(t) \\
&\left[\begin{aligned} &l_1 \cos \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}'(l_1) \delta_{1j}(t) + \theta_2 \right) \dot{\theta}_1 \\ &+ \sum_{j=1}^{n_{m,2}} I_{42j} \dot{\delta}_{2j}(t) + \sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \delta_{1j}(t) \sin \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}'(l_1) \delta_{1j}(t) + \theta_2 \right) \dot{\theta}_1 \\ &+ \sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \dot{\delta}_{1j}(t) \cos \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}'(l_1) \delta_{1j}(t) + \theta_2 \right) \end{aligned} \right] \tag{2.99}
\end{aligned}$$

The kinetic energy of the payload is

$$\begin{aligned}
T_p &= \frac{1}{2} m_p (\mathbf{v}_2^0)^T \mathbf{v}_2^0 \Big|_{x_2=l_2} + \frac{1}{2} J_p (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\alpha}_1 + \dot{\alpha}_2)^2 \\
&= \frac{1}{2} m_p \left\{ [l_1^2 + w_1^2(l_1, t)] \dot{\theta}_1^2 + [l_2^2 + w_2^2(l_2, t)] (\dot{\theta}_1 + \dot{\alpha}_1 + \dot{\theta}_2)^2 + \dot{w}_1^2(l_1, t) + \dot{w}_2^2(l_2, t) \right\} \\
&\quad + \frac{1}{2} m_p (\dot{\theta}_1 + \dot{\alpha}_1 + \dot{\theta}_2) \left\{ \begin{aligned} &2l_2 [l_1 c_{\alpha_1, 2} \dot{\theta}_1 + w_1(l_1, t) s_{\alpha_1, 2} \dot{\theta}_1 + \dot{w}_1(l_1, t) c_{\alpha_1, 2}] \\ &+ 2l_2 \dot{w}_2(l_2, t) - 2w_2(l_2, t) [l_1 s_{\alpha_1, 2} \dot{\theta}_1 - w_1(l_1, t) c_{\alpha_1, 2} \dot{\theta}_1 + \dot{w}_1(l_1, t) s_{\alpha_1, 2}] \end{aligned} \right\} \\
&\quad + m_p l_1 \dot{\theta}_1 \dot{w}_1(l_1, t) + m_p \dot{w}_2(l_2, t) [l_1 c_{\alpha_1, 2} \dot{\theta}_1 + w_1(l_1, t) s_{\alpha_1, 2} \dot{\theta}_1 + \dot{w}_1(l_1, t) c_{\alpha_1, 2}] \\
&\quad + \frac{1}{2} J_p (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\alpha}_1 + \dot{\alpha}_2)^2 \\
&= \frac{1}{2} m_p \left\{ \begin{aligned} &\left[l_1^2 + \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \delta_{1j}(t) \right)^2 \right] \dot{\theta}_1^2 \\ &+ \left[l_2^2 + \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(l_2) \delta_{2j}(t) \right)^2 \right] (\dot{\theta}_1 + \sum_{j=1}^{n_{m,1}} \phi_{1j}'(l_1) \dot{\delta}_{1j}(t) + \dot{\theta}_2)^2 \\ &+ \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \dot{\delta}_{1j}(t) \right)^2 + \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(l_2) \dot{\delta}_{2j}(t) \right)^2 \end{aligned} \right\} \\
&\quad + \frac{1}{2} m_p (\dot{\theta}_1 + \sum_{j=1}^{n_{m,1}} \phi_{1j}'(l_1) \dot{\delta}_{1j}(t) + \dot{\theta}_2) \\
&\quad \left\{ \begin{aligned} &\left[\begin{aligned} &l_1 \cos \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}'(l_1) \delta_{1j}(t) + \theta_2 \right) \dot{\theta}_1 \\ &+ \sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \delta_{1j}(t) \sin \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}'(l_1) \delta_{1j}(t) + \theta_2 \right) \dot{\theta}_1 + 2l_2 \sum_{j=1}^{n_{m,2}} \phi_{2j}(l_2) \dot{\delta}_{2j}(t) \\ &+ \sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \dot{\delta}_{1j}(t) \cos \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}'(l_1) \delta_{1j}(t) + \theta_2 \right) \end{aligned} \right] \\ &- 2 \sum_{j=1}^{n_{m,2}} \phi_{2j}(l_2) \delta_{2j}(t) \left[\begin{aligned} &l_1 \sin \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}'(l_1) \delta_{1j}(t) + \theta_2 \right) \dot{\theta}_1 \\ &- \sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \delta_{1j}(t) \cos \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}'(l_1) \delta_{1j}(t) + \theta_2 \right) \dot{\theta}_1 \\ &+ \sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \dot{\delta}_{1j}(t) \sin \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}'(l_1) \delta_{1j}(t) + \theta_2 \right) \end{aligned} \right] \end{aligned} \right\}
\end{aligned}$$

$$\begin{aligned}
& +m_p l_1 \dot{\theta}_1 \sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \dot{\delta}_{1j}(t) \\
& +m_p \sum_{j=1}^{n_{m,2}} \phi_{2j}(l_2) \dot{\delta}_{2j}(t) \left[\begin{aligned} & l_1 \cos \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}'(l_1) \delta_{1j}(t) + \theta_2 \right) \dot{\theta}_1 \\ & + \sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \delta_{1j}(t) \sin \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}'(l_1) \delta_{1j}(t) + \theta_2 \right) \dot{\theta}_1 \\ & + \sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \dot{\delta}_{1j}(t) \cos \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}'(l_1) \delta_{1j}(t) + \theta_2 \right) \end{aligned} \right] \quad (2.100) \\
& + \frac{1}{2} J_p (\dot{\theta}_1 + \dot{\theta}_2 + \sum_{j=1}^{n_{m,1}} \phi_{1j}'(l_1) \delta_{1j}(t) + \sum_{j=1}^{n_{m,2}} \phi_{2j}'(l_2) \delta_{2j}(t))^2
\end{aligned}$$

The total kinetic energy of the system is

$$T = T_{h1} + T_{h2} + T_{l1} + T_{l2} + T_p \quad (2.101)$$

The total potential energy of the system is given by

$$\begin{aligned}
V &= \frac{1}{2} \int_0^{l_1} (EI)_1 \left[\frac{\partial^2 w_1(x_1, t)}{\partial x_1^2} \right]^2 dx_1 + \frac{1}{2} \int_0^{l_2} (EI)_2 \left[\frac{\partial^2 w_2(x_2, t)}{\partial x_2^2} \right]^2 dx_2 \\
&= \frac{1}{2} \int_0^{l_1} (EI)_1 \left[\sum_{j=1}^{n_{m,1}} \phi_{1j}''(x_1) \delta_{1j}(t) \right]^2 dx_1 + \frac{1}{2} \int_0^{l_2} (EI)_2 \left[\sum_{j=1}^{n_{m,2}} \phi_{2j}''(x_2) \delta_{2j}(t) \right]^2 dx_2 \quad (2.102)
\end{aligned}$$

Use equations (2.101) and (2.102) to form the Lagrangian of the system:

$$L = T - V \quad (2.103)$$

The total dissipation function of the system is

$$R = \frac{1}{2} \sum_{i=1}^2 b_{\theta_i} \dot{\theta}_i^2 + \frac{1}{2} \sum_{i=1}^2 \sum_{j=1}^{n_{m,i}} b_{ij} \dot{\delta}_{ij}^2 \quad (2.104)$$

The generalized coordinates of the system are

$$\mathbf{q} = (q_1, \dots, q_{2+n_{m,1}+n_{m,2}}) = (\theta_1, \theta_2, \delta_{11}, \dots, \delta_{1n_{m,1}}, \delta_{21}, \dots, \delta_{2n_{m,2}}).$$

The dynamic model of the two-link flexible robotic manipulator is obtained by substituting equations (2.103) and (2.104) into the Lagrange's equations

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} + \frac{\partial R}{\partial \dot{q}_i} = f_i, \quad i = 1, \dots, 2 + n_{m,1} + n_{m,2}. \quad (2.105)$$

The above steps are coded to obtain the dynamic equations of the system using the symbolic manipulation software MAPLE (Maplesoft, 2005). This method protects against errors which are common in manual algebraic manipulation; for example, transcription errors, sign errors and misapplication of the chain rule of differentiation. The derived dynamic equations are given in Appendix A. The MATLAB (Mathworks, 2005) code and the C code of this model are obtained by using the Code Generation package of MAPLE. This nonlinear model is used to simulate the two-link flexible manipulator. The linear model derived from this nonlinear model is used in the design of the MPC controller as presented in Chapter 5.

2.4 Linearization of the Equations of Motion

From Section 2.31 the closed-form equations of motion of a planar n -link flexible manipulator are obtained in the form

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{K}\mathbf{q} + \mathbf{B}\dot{\mathbf{q}} = \mathbf{u} \quad (2.106)$$

These equations are nonlinear. If a linear model of the nonlinear system can be obtained that is sufficiently accurate over a suitable range of operating conditions, then techniques of linear control analysis and design may be applied to the linear model. In using such a linear model it must be ascertained that it remains valid over a suitable range of operation of the plant. For a wide operating range more than one linear model is needed to cover different regions of the operating range. Here the proper linear model should be used is selected depending on the operating conditions.

In this section, local linearization technique is used to derive a linear model for a nonlinear planar n -link flexible manipulator. The resulting linear model is used for controller design and analysis, as presented in the subsequent chapters of the thesis.

The operating point of a dynamic system defines its overall state under a specified operating condition. A linearized model is an approximation that is valid in a small region around an operating point of the system. Near the operating point the approximation will be good, while far away it will become increasingly poor. When generating a linearized model from a nonlinear model, the choice of operating point is important as it will determine the

accuracy of the approximation. The operating range of motion of a planar n -link flexible manipulator can be large (e.g., large changes of joint angles). Consequently, in linearization of the equations of motion, often many operating points have to be considered. For this reason, in this section we present an approach for deriving closed-form linearized equations about any operating point. Consider the nonlinear equations of motion given by equation (2.106). For an operating point $(\mathbf{q}_0, \dot{\mathbf{q}}_0, \ddot{\mathbf{q}}_0, \mathbf{u}_0)$, a linear time-invariant approximation to this system is valid in a small region around this operating point. To derive the linearized model, it helps to first define a new set of variables centered about the operating point of the states, inputs, and outputs, as

$$\begin{aligned}\Delta \mathbf{q} &= \mathbf{q} - \mathbf{q}_0; & \Delta \dot{\mathbf{q}} &= \dot{\mathbf{q}} - \dot{\mathbf{q}}_0 \\ \Delta \ddot{\mathbf{q}} &= \ddot{\mathbf{q}} - \ddot{\mathbf{q}}_0; & \Delta \mathbf{u} &= \mathbf{u} - \mathbf{u}_0\end{aligned}\quad (2.107)$$

where the subscript “0” denotes the operating configuration and “ Δ ” indicates the variation from the operating point. By definition, the operating point is a solution of equation (2.106), as given by

$$\mathbf{M}(\mathbf{q}_0)\ddot{\mathbf{q}}_0 + \mathbf{h}(\mathbf{q}_0, \dot{\mathbf{q}}_0) + \mathbf{K}\mathbf{q}_0 + \mathbf{B}\dot{\mathbf{q}}_0 = \mathbf{u}_0 \quad (2.108)$$

The motion near the operating point can be described by

$$\begin{aligned}\mathbf{M}(\mathbf{q}_0 + \Delta \mathbf{q})(\ddot{\mathbf{q}}_0 + \Delta \ddot{\mathbf{q}}) + \mathbf{h}(\mathbf{q}_0 + \Delta \mathbf{q}, \dot{\mathbf{q}}_0 + \Delta \dot{\mathbf{q}}) \\ + \mathbf{K}(\mathbf{q}_0 + \Delta \mathbf{q}) + \mathbf{B}(\dot{\mathbf{q}}_0 + \Delta \dot{\mathbf{q}}) = \mathbf{u}_0 + \Delta \mathbf{u}\end{aligned}\quad (2.109)$$

Perform a Taylor series expansion on equation (2.109). Assume the deviations from the operating point are small so that the Taylor series can be truncated after the first term, resulting in the following equation:

$$\begin{aligned}\left[\mathbf{M}(\mathbf{q}_0) + \left. \frac{\partial \mathbf{M}}{\partial \mathbf{q}} \right|_{\mathbf{q}_0} \Delta \mathbf{q} \right] (\ddot{\mathbf{q}}_0 + \Delta \ddot{\mathbf{q}}) + \left[\mathbf{h}(\mathbf{q}_0, \dot{\mathbf{q}}_0) + \left. \frac{\partial \mathbf{h}}{\partial \mathbf{q}} \right|_{\mathbf{q}_0, \dot{\mathbf{q}}_0} \Delta \mathbf{q} + \left. \frac{\partial \mathbf{h}}{\partial \dot{\mathbf{q}}} \right|_{\mathbf{q}_0, \dot{\mathbf{q}}_0} \Delta \dot{\mathbf{q}} \right] \\ + \mathbf{K}(\mathbf{q}_0 + \Delta \mathbf{q}) + \mathbf{B}(\dot{\mathbf{q}}_0 + \Delta \dot{\mathbf{q}}) = \mathbf{u}_0 + \Delta \mathbf{u}\end{aligned}\quad (2.110)$$

Substituting equation (2.108) into equation (2.110) and neglecting the high-order terms, we obtain the linearized equations of motion of the system:

$$\mathbf{M}(\mathbf{q}_0)\Delta\ddot{\mathbf{q}} + \left(\left. \frac{\partial \mathbf{h}}{\partial \dot{\mathbf{q}}} \right|_{\mathbf{q}_0, \dot{\mathbf{q}}_0} + \mathbf{B} \right) \Delta\dot{\mathbf{q}} + \left[\left. \frac{\partial \mathbf{M}}{\partial \mathbf{q}} \right|_{\mathbf{q}_0} \ddot{\mathbf{q}}_0 + \left. \frac{\partial \mathbf{h}}{\partial \mathbf{q}} \right|_{\mathbf{q}_0, \dot{\mathbf{q}}_0} + \mathbf{K} \right] \Delta\mathbf{q} = \Delta\mathbf{u} \quad (2.111)$$

In this thesis, the controller design is based on a linearized model of the plant. Specifically, the nonlinear model is linearized about an operating configuration where the each joint angle has a specific value θ_{i0} while all the other generalized coordinates, speeds and inputs are equal to zero. This is the equilibrium operating point of the system when each link of the planar flexible robotic manipulator rests at a specific joint angle with no input and no deflection. The equilibrium operating point remains steady and constant with time, with all states in the model at equilibrium. It is also known as a steady-state or trimmed operating point. A model that is linearized about a stable equilibrium operating point is likely to remain within the region around the equilibrium value. Such a model will give a good approximation to the nonlinear model. A model linearized about an unstable equilibrium operating point is not likely to remain within the region around the equilibrium value. Such a model will become a poor approximation to the nonlinear model once it deviates considerably from this equilibrium point. So for a robotic manipulator with a large range of movement, a piecewise linear model (linearized at a sequence of equilibrium operating points) should be used to reduce the model error. This is a special case of the general linearized model presented in equation (2.111). For an n -link flexible manipulator, \mathbf{h} in equation (2.111) is the vector of centripetal and Coriolis terms which involve products of the generalized velocities. So, each term of the derivative $\left. \frac{\partial \mathbf{h}}{\partial \dot{\mathbf{q}}} \right|_{\mathbf{q}_0, \dot{\mathbf{q}}_0}$ consists of the generalized velocities. In this case then

$$\begin{aligned} \left. \frac{\partial \mathbf{h}}{\partial \dot{\mathbf{q}}} \right|_{\mathbf{q}_0, \dot{\mathbf{q}}_0} &= \left. \frac{\partial \mathbf{h}}{\partial \dot{\mathbf{q}}} \right|_{\mathbf{q}_0, 0} = 0 \\ \left. \frac{\partial \mathbf{M}}{\partial \mathbf{q}} \right|_{\mathbf{q}_0} \ddot{\mathbf{q}}_0 &= 0 \end{aligned} \quad (2.112)$$

By substituting equation (2.112) into equation (2.111), the linearized equations of motion of the system are obtained as

$$\mathbf{M}(\mathbf{q}_0)\Delta\ddot{\mathbf{q}} + \mathbf{B}\Delta\dot{\mathbf{q}} + \mathbf{K}\Delta\mathbf{q} = \Delta\mathbf{u} \quad (2.113)$$

where $\mathbf{q}_0 = (\theta_1, \dots, \theta_n, 0, \dots, 0)$, and \mathbf{M} , \mathbf{B} , and \mathbf{K} are constant matrices. Also

$$\mathbf{q} = (\theta_1, \dots, \theta_n, \delta_{11}, \dots, \delta_{1n_{m,1}}, \dots, \delta_{n1}, \dots, \delta_{nn_{m,n}})$$

We assume that the system outputs are the joint angles and the link deflections. The linear state-space equations of the system are

$$\begin{aligned}\dot{\mathbf{X}} &= \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{U} \\ \mathbf{Y} &= \mathbf{C}\mathbf{X}\end{aligned}\tag{2.114}$$

where

$$\begin{aligned}\mathbf{X}^T &= [\Delta\mathbf{q}, \Delta\dot{\mathbf{q}}] \\ &= \begin{bmatrix} \Delta\theta_1, \dots, \Delta\theta_n, \Delta\delta_{11}, \dots, \Delta\delta_{1n_{m,1}}, \dots, \Delta\delta_{n1}, \dots, \Delta\delta_{nn_{m,n}}, \\ \Delta\dot{\theta}_1, \dots, \Delta\dot{\theta}_n, \Delta\dot{\delta}_{11}, \dots, \Delta\dot{\delta}_{1n_{m,1}}, \dots, \Delta\dot{\delta}_{n1}, \dots, \Delta\dot{\delta}_{nn_{m,n}} \end{bmatrix} \\ \mathbf{A} &= \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{B} \end{bmatrix} \\ \mathbf{B} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}^{-1} \end{bmatrix} \\ \mathbf{C} &= \begin{bmatrix} \mathbf{I}_{n \times n} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Phi_{1 \times n_{m,1}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \Phi_{1 \times n_{m,n}} & \mathbf{0} \end{bmatrix} \\ \mathbf{U} &= \begin{bmatrix} \mathbf{0} \\ \Delta\mathbf{u} \end{bmatrix} \\ \Phi_{1 \times n_{m,i}} &= [\phi_{i1}(l_i), \dots, \phi_{in_{m,i}}(l_i)]\end{aligned}\tag{2.115}$$

2.5 Non-minimum Phase Characteristic

In this section, we investigate the non-minimum phase characteristic of a flexible link robotic manipulator. For a linear continuous-time system, if all the zeros are in the left half of the complex plane, such a system is called a minimum-phase system. A minimum-phase system with a given magnitude curve in the Bode plot will produce the smallest net change in the associated phase angle. If there are system zeros in the closed right half of the complex plane, such a system is called non-minimum phase system. In the linear discrete-time case, the

system is non-minimum phase if there are zeros that lie outside the unit circle.

The non-minimum phase concept can be extended to a nonlinear system. By using the inverse dynamics (input-output linearization) approach, the dynamics of a nonlinear system are decomposed into an external part and an internal part, as shown in Figure 2.8. The zero-dynamics of the system are defined as the internal dynamics of the system when the control input maintains the system output at zero. A nonlinear system is minimum-phase if its zero-dynamics are asymptotically stable; otherwise, the system is non-minimum phase.

Instability of the internal dynamics implies instability of the overall closed-loop system. The inverse dynamics approach cannot be applied directly to non-minimum phase systems because they cannot be inverted (inverted dynamics is unstable). This is a generalization of the linear system result that the inverse of the transfer function of a non-minimum phase linear system is unstable. Therefore, for such systems, control laws that achieve perfect (asymptotically convergent) tracking error should not be pursued in general. Instead, one should find controllers that lead to acceptably small tracking errors for the desired trajectories.

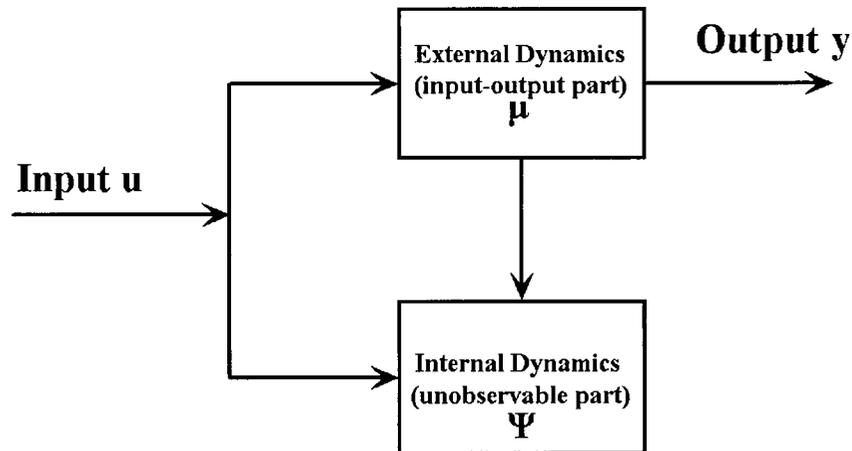


Figure 2.8 External dynamics and internal dynamics of a nonlinear system.

Next we investigate non-minimum phase characteristics of a linear system using the root locus method. The root locus gives the closed-loop pole trajectories as a function of the feedback gain K , assuming negative feedback (or some other variable parameter).

Consider a single-input single-output (SISO) system with open-loop transfer function

$$G(s) = \frac{N(s)}{D(s)} \quad (2.116)$$

The zeros of the system are the roots of

$$N(s) = 0 \quad (2.117)$$

The closed-loop transfer function is

$$G(s) = \frac{KN(s)}{D(s) + KN(s)} \quad (2.118)$$

The closed-loop poles are the roots of

$$D(s) + KN(s) = 0 \quad (2.119)$$

Divided both sides of equation (2.119) by K . We have

$$\frac{D(s)}{K} + N(s) = 0 \quad (2.120)$$

From equation (2.120) and equation (2.117), we notice that as the feedback gain increases, the closed-loop poles of the system are attracted towards the open-loop zeros of the system. If the open-loop system is non-minimum phase (i.e., has RHS zeros), then the closed-loop system can become unstable under static output feedback.

For multi-input multi-output (MIMO) linear systems, the transfer function $G(s)$ becomes a transfer-function matrix $\mathbf{G}(s)$. The transmission zeros (Maciejowski, 1989) of a controllable and observable m input and r output n th-order linear state-space system

$$\begin{aligned} \dot{\mathbf{X}} &= \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{u} \\ \mathbf{Y} &= \mathbf{C}\mathbf{X} + \mathbf{D}\mathbf{u} \end{aligned} \quad (2.121)$$

are defined as the values of s for which the normal rank of the system matrix drops to

$$\text{rank} \begin{bmatrix} \mathbf{A} - s\mathbf{I}_n & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} < n + \min(r, m) \quad (2.122)$$

It can be shown that the open-loop transmission zeros of the system are the eigenvalues of

$$\mathbf{A} - \mathbf{KBC} \quad \text{as } K \rightarrow \infty \quad (2.123)$$

The closed-loop characteristic roots when negative feedback of the form $K\mathbf{I}$ is applied to the system are the eigenvalues of the closed-loop 'A' matrix

$$\mathbf{A}_c = \mathbf{A} - K\mathbf{B}\mathbf{C} \quad (2.124)$$

From equations (2.123) and (2.124) we observe that as the static feedback gain K is increased, the closed-loop poles of the system are attracted towards the open-loop transmission zeros of the system. If the open-loop system has non-minimum phase transmission zeros (RHS zeros), then the closed-loop system can become unstable under static output feedback.

Controller design for a non-minimum phase system is more difficult than that for a minimum phase system owing to the limited gain margin. The improper controller design can make the instability of the zero-dynamics to have an effect on the closed-loop system stability. The non-minimum phase characteristics of the system limit the loop bandwidth, and the achievable performance of the feedback system is reduced. For example, when using inverse dynamics algorithm, the right hand side (RHS) zeros will become unstable poles in the inverse system. The controller now has unstable poles that can cause the overall system to become unstable. One noticeable characteristic of a non-minimum phase system is the nature of its step response. For a non-minimum phase system, the step response initially starts to move in the direction opposite to final steady-state value.

To illustrate the non-minimum phase characteristic of a flexible link robotic manipulator system, consider the single-link flexible manipulator shown in Figure 2.7. The nonlinear dynamic model of the system is given by equations (2.92) and (2.93). Using equation (2.113), the linearized model of the system with a single flexible mode is obtained as

$$\begin{aligned} M_{11}\ddot{\theta}_1 + M_{12}\ddot{\delta}_{11}(t) + b_{\theta}\dot{\theta}_1 &= \tau_1 \\ M_{21}\ddot{\theta}_1 + M_{22}\ddot{\delta}_{11}(t) + k_1\delta_{11}(t) + b_{11}\dot{\delta}_{11} &= 0 \end{aligned} \quad (2.125)$$

where

$$\begin{aligned}
M_{11} &= (J_{h1} + J_{o1} + J_p + m_p l_1^2) \\
M_{12} = M_{21} &= I_{111} + m_p l_1 \phi_{11}(l_1) + J_p \phi_{11}'(l_1) \\
M_{22} &= m_1 + m_p \phi_{11}(l_1)^2 + J_p \phi_{11}'(l_1)^2 \\
k_1 &= \omega_{11}^2 \left[m_1 + m_p \phi_{11}(l_1)^2 + J_p \phi_{11}'(l_1)^2 \right]
\end{aligned} \tag{2.126}$$

Assume that the overall system input is the joint torque τ_1 and the outputs are the joint angle θ_1 and the link deflection $w_1(l_1, t) = \phi_{11}(l_1) \delta_{11}(t)$. The transfer function from input torque τ_1 to joint angle θ_1 is

$$\begin{aligned}
G_{\tau\theta}(s) &= \frac{\theta_1(s)}{\tau_1(s)} \\
&= \frac{M_{22}s^2 + b_{11}s + k_1}{s \left[(M_{11}M_{22} - M_{12}^2)s^3 + (M_{11}b_{11} + M_{22}b_{\theta 1})s^2 + (M_{11}k_1 + b_{\theta 1}b_{11})s + b_{\theta 1}k_1 \right]}
\end{aligned} \tag{2.127}$$

The transfer function from input torque τ_1 to link deflection $w_1(l_1, t) = \phi_{11}(l_1) \delta_{11}(t)$ is

$$\begin{aligned}
G_{\tau w}(s) &= \frac{w_1(s)}{\tau_1(s)} \\
&= \frac{-M_{12}\phi_{11}(l_1)s}{(M_{11}M_{22} - M_{12}^2)s^3 + (M_{11}b_{11} + M_{22}b_{\theta 1})s^2 + (M_{11}k_1 + b_{\theta 1}b_{11})s + b_{\theta 1}k_1}
\end{aligned} \tag{2.128}$$

In this case, from equations (2.127) and (2.128) we observe that there are no RHS transmission zeros. The overall system is a minimum-phase system. But there is a zero at the origin of the s -plane (i.e., at $s = 0$) in equation (2.128). If we assume that the overall system output is the link deflection w_1 , there is a zero in the closed right half of the complex plane, so this system is a non-minimum phase system.

If we assume that the overall system output is the tip position

$$\theta_{1tip} = \theta_1 + \frac{w_1(l_1, t)}{l_1} = \theta_1 + \frac{\phi_{11}(l_1) \delta_{11}(t)}{l_1} \tag{2.129}$$

the overall transfer function of the system becomes

$$G_{\tau\theta_{tip}}(s) = \frac{\theta_{tip}(s)}{\tau_1(s)} = \frac{\theta_1(s)}{\tau_1(s)} + \frac{w_1(l_1, t)}{l_1\tau_1(s)}$$

$$= \frac{\left[M_{22} - \frac{M_{12}\phi_{11}(l_1)}{l_1} \right] s^2 + b_{11}s + k_1}{s \left[(M_{11}M_{22} - M_{12}^2)s^3 + (M_{11}b_{11} + M_{22}b_{\theta 1})s^2 + (M_{11}k_1 + b_{\theta 1}b_{11})s + b_{\theta 1}k_1 \right]} \quad (2.130)$$

The zero dynamics of this system depends on the sign of the term $C_{z1} = \left[M_{22} - \frac{M_{12}\phi_{11}(l_1)}{l_1} \right]$ in equation (2.130). If this term is negative then the zero dynamics of the system is unstable, and the system is non-minimum phase.

For a single-link flexible robotic manipulator without a payload, substituting equation (2.126) into C_{z1} , we have

$$C_{z1} = M_{22} - \frac{M_{12}\phi_{11}(l_1)}{l_1}$$

$$= m_1 - \frac{I_{111}\phi_{11}(l_1)}{l_1} \quad (2.131)$$

From equation (2.56), the frequency equation without payload is

$$1 + \cos(\beta_1 l_1) \cosh(\beta_1 l_1) = 0 \quad (2.132)$$

The clamped-free mode-shape function is obtained from equations (2.38) and (2.46) as

$$\phi_{11}(x_1) = \cosh(\beta_1 x_1) - \cos(\beta_1 x_1) - \frac{\cosh(\beta_1 l_1) + \cos(\beta_1 l_1)}{\sinh(\beta_1 l_1) + \sin(\beta_1 l_1)} [\sinh(\beta_1 x_1) - \sin(\beta_1 x_1)] \quad (2.133)$$

By using equations (2.132) and (2.133), the closed form of the integral parameter in equation (2.131) is found as

$$I_{111} = \int_0^{l_1} \rho_1 \phi_{11}(x_1) x_1 dx_1 = \frac{2m_1 l_1}{(\beta_1 l_1)^2} \quad (2.134)$$

By substituting equation (2.134) into equation (2.131), we have

$$C_{z1} = m_1 - \frac{2m_1\phi_{11}(l_1)}{(\beta_1 l_1)^2} = m_1 \left[1 - \frac{2\phi_{11}(l_1)}{(\beta_1 l_1)^2} \right] \quad (2.135)$$

where $\beta_1 l_1$ is obtained by solving frequency equation (2.132), and it has the constant value

$$\beta_1 l_1 = 1.8751 \quad (2.136)$$

From equation (2.133), we see that the first clamped-free mode shape $\phi_{11}(x_1)$ is always positive and is an increasing function. Substituting equation (2.136) into (2.133), we have

$$\phi_{11}(l_1) = 2 \quad (2.137)$$

Substituting equation (2.136) and (2.137) into equation (2.135), we have

$$C_{z1} = m_1 \left[1 - \frac{6}{(1.8751)^2} \right] < 0 \quad (2.138)$$

If the system output is the tip position, the single-link flexible robotic manipulator without a payload is always a non-minimum phase system. The non-minimum phase condition in this case is a result of the non-collocated sensor and actuator positions.

For a system with payload, the non-minimum phase characteristic will also depend on the payload. If we assume the single link to be an aluminum rod or an aluminum thin beam with the parameters given in Table 2.1, we can calculate the system zeros using equation (2.130). Figure 2.9 and Figure 2.10 show the pole-zero maps of these systems. We can see that both systems have RHS zeros, so the systems are non-minimum phase.

For the two-link flexible manipulator system with the parameters given in Table 2.1, we can find the system zeros and poles by using the nonlinear model developed in Section 2.3.3 and linearizing it about an operating point. We assume that the system outputs are the tip positions of link 1 and link 2. Figure 2.11 shows the pole-zero map of the system which is linearized at the operating point where all the states are equal to zero. We notice that the system has a RHS zero; so the system is non-minimum phase.

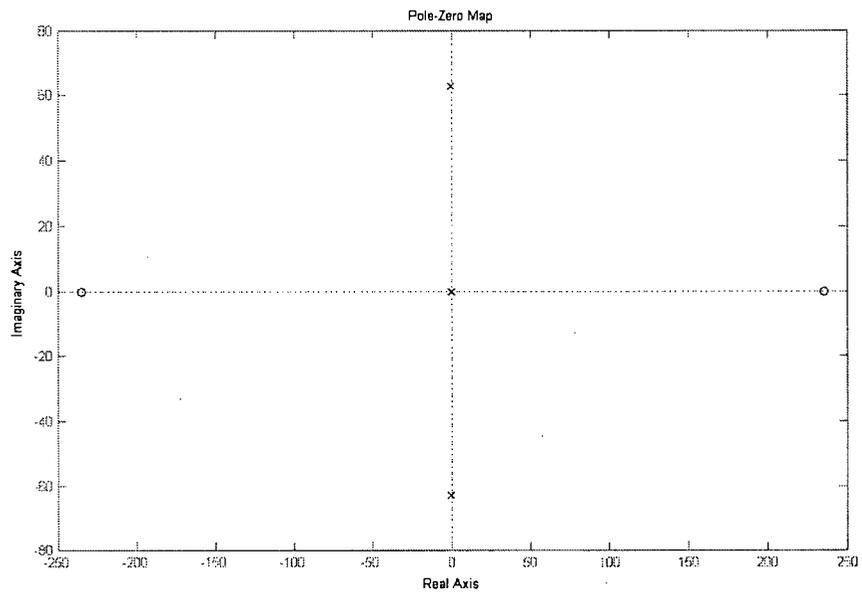


Figure 2.9 Pole-zero map of a single-link flexible robotic manipulator (aluminum rod).

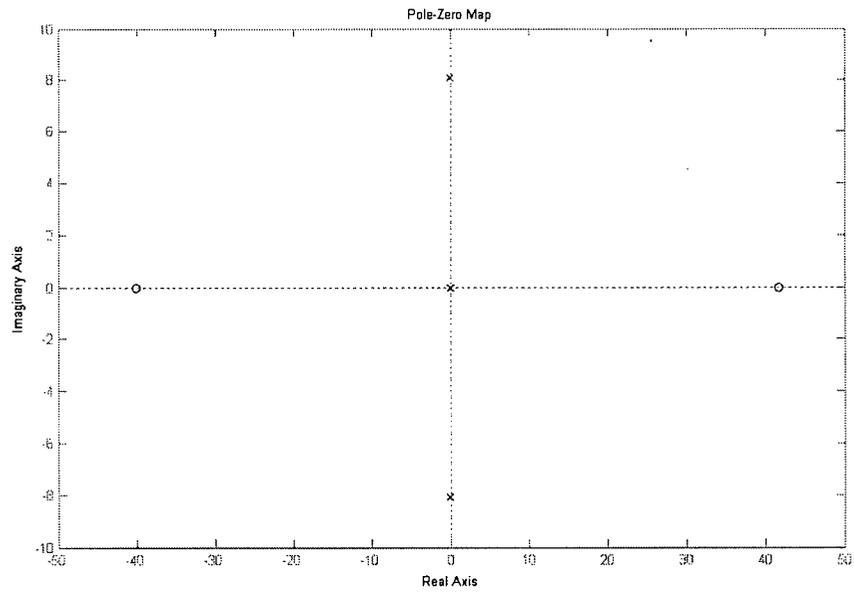


Figure 2.10 Pole-zero map of a single-link flexible robotic manipulator (aluminum thin beam).

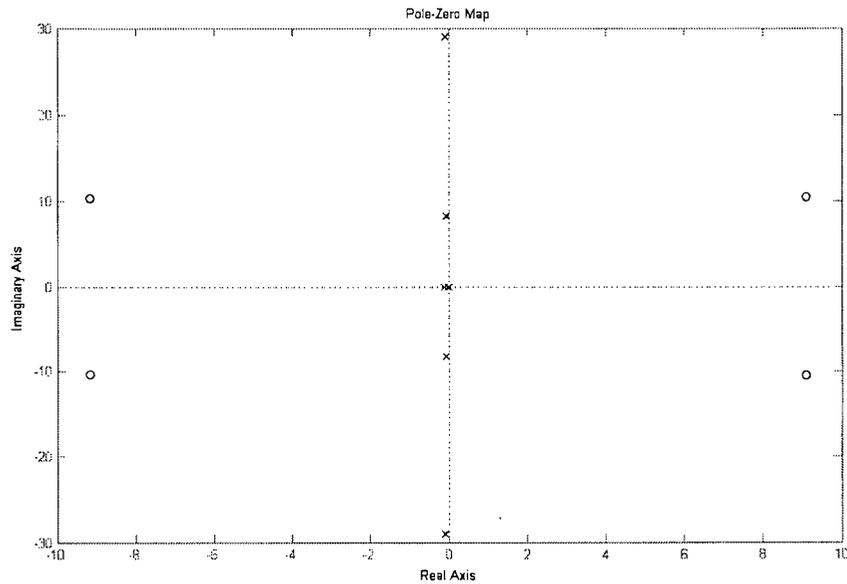


Figure 2.11 Pole-zero map of a two-link flexible robotic manipulator.

2.6 Summary

This chapter presented the derivation of the models for the kinematics and dynamics (kinetics) of a planar flexible link robotic manipulator. A flexible link model was developed based on the Bernoulli-Euler beam theory. Boundary conditions and mode shape selection were investigated. More realistic and accurate boundary conditions were obtained through the balance of moments and shear forces at the end of each link in the dynamic model development of the system. A dynamic model for the manipulator was derived using Lagrange's equations. Closed-form equations of motion for a planar n -link flexible manipulator were established. Detailed nonlinear models for a single-link and two-link flexible manipulator were derived. Local linearization techniques were used to derive a linear state-space model of the system at an equilibrium operating point. Non-minimum phase characteristic of the flexible-link system was analyzed. It was shown that if the overall system output was the tip position, the system would be non-minimum. The non-minimum phase characteristic of a flexible link manipulator system was shown to be a result of the non-collocated sensor and actuator positions. For such a system, perfect or asymptotic convergent

tracking would not be achieved, and controllers should be designed that would lead to acceptably small tracking errors for the trajectories of interest.

Chapter 3

Intelligent Model Predictive Control

In this chapter an intelligent model predictive control (IMPC) approach is developed for the motion control of a flexible-link robotic manipulator. First, in Section 3.1, overall structure of the IMPC algorithm and its underlying strategy are given. Main components of the IMPC system are developed in the subsequent sections. In Section 3.2, a multi-stage, unconstrained, model predictive control (MPC) scheme with guaranteed nominal stability is developed. Application of the IMPC algorithm for a robotic system with an unknown payload is presented in Section 3.3. Fuzzy rule-based intelligent auto-tuning of the MPC controller, which constitutes the IMPC, is addressed in Section 3.4.

3.1 Overall Structure of IMPC

Figure 3.1 shows a schematic diagram of the intelligent model predictive control (IMPC) system. The IMPC is based on a two-level hierarchical control architecture. This control structure is used to combine the advantages of conventional MPC control and knowledge-based soft control, resulting in the IMPC for control of the plant, which is a flexible-link robotic manipulator in the present application.

The top level of the two-level architecture is a fuzzy rule-based intelligent decision-making system. The bottom level consists of two modules: System identification module, which accommodates the model parameter variations in the system, and MPC module, which generates the control inputs based on the linear model generated by the system identification module. The upper-level intelligent fuzzy rule-based tuner interacts with the bottom level modules. The upper-level fuzzy tuner automatically adjusts the tuning parameters of the MPC controller based on the output feedback. Proper tuning parameters are selected by the fuzzy tuner in order to achieve the desired closed-loop stability, and performance while satisfying the input and output constraints. The fuzzy tuner is also able to adjust the model structure of

the system identification module, if necessary, for large model errors, and as a result improves the robustness of the controller. Design and development of each component of the IMPC for motion control of a flexible-link robotic manipulator system are presented in the following sections.

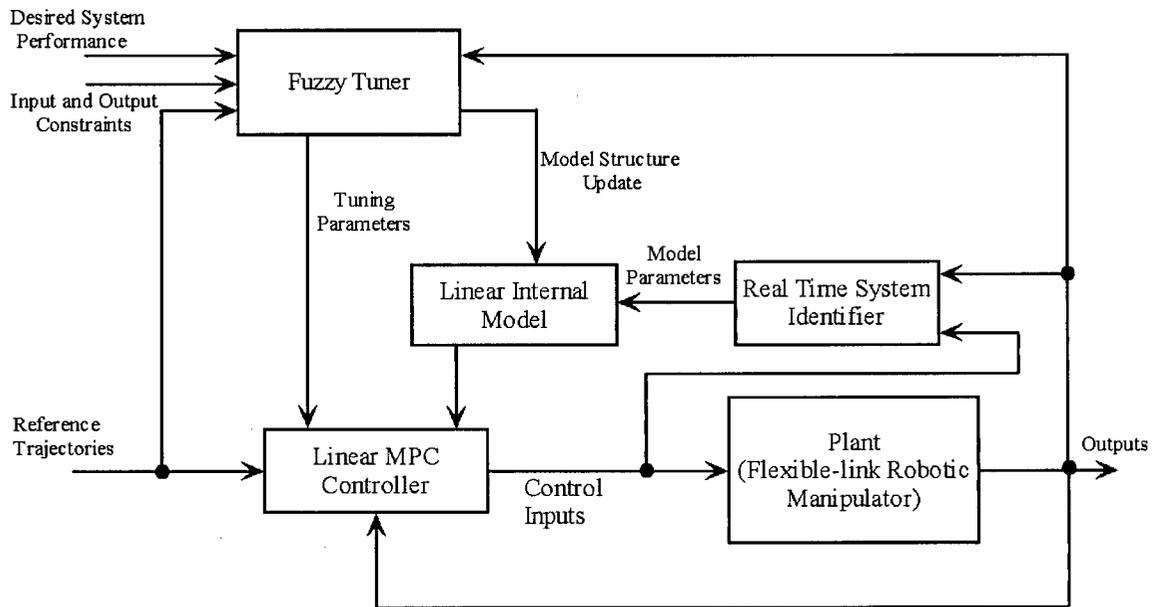


Figure 3.1 Structure of the intelligent model predictive control (IMPC) system.

3.2 MPC Controller Design

A computationally efficient multi-stage MPC algorithm with guaranteed nominal stability is developed in this section. It will be used as the lower-level direct controller of the flexible-link robotic manipulator system. Among the reasons for choosing MPC for low-level direct control of the flexible manipulator are the following:

- MPC deals with multivariable systems in a systematic way. The design of an MPC is based on a plant model and a cost function (i.e., performance index). This results in more systematic design and analysis than with traditional proportional-integral-derivative (PID) control. Also this is an effective way to deal with dynamic interaction (dynamic coupling) in multivariable systems and gives good overall closed-loop performance.

- It can take account of actuator limitations, and can systematically handle input and output constraints. This represents a major difference between MPC and other popular multivariable control techniques (e.g., Linear-Quadratic-Gaussian or LQG method). MPC controllers explicitly consider constraints and allocate the available plant resources intelligently as the system evolves over time. It makes the plant operate closer to the constraints, and hence provides better performance, which can lead to more profitable industrial operation.
- The design objectives can be specified in a flexible manner by changing the cost function of MPC.
- It can effectively control systems with time delays, nonminimum-phase characteristics, and instabilities. As shown in Chapter 2, the flexible-link robotic manipulator system has nonminimum-phase characteristics. MPC can deal with these characteristics through proper setting up of the cost function of the controller. A nonminimum-phase system has a shorter term response in one direction and a longer term response in the opposite direction. The MPC optimization should focus primarily on the longer-term behavior in order to avoid system movements in the wrong direction.
- It takes full advantage of the power available in modern control computer hardware.

3.2.1 Basic Principles of MPC

Model predictive control (MPC) is a model based control technique. It generates the control input signals to the plant by combining a prediction and a receding-horizon control strategy. It uses an explicit internal model to generate predictions of the future plant behavior. Usually, an approximate, linear discrete-time plant model is used to predict the plant behavior over a future prediction horizon. A cost function (i.e., performance index) which represents the desired system performance over a future horizon has to be defined. The control strategy determines the future control inputs to the plant so as to minimize the cost function while taking into account the plant constraints at each time instance. Such constraints can include the physical limits of the actuators, boundaries of safe operation, and allowable lower limits for product quality. In the receding horizon control framework, only the first computed

control action is implemented. At the next sampling time, the optimization is solved again with new measurements from the plant. The length of the prediction horizon remains the same as before, but slides forward by one sampling interval at each step. The on-line optimization can be typically reduced to either linear programming or quadratic programming. The purpose of taking new measurements at each time step is to compensate for unmeasured disturbances and model inaccuracies, both of which cause the system output to be different from the one predicted by the model. This also introduces a feedback element into the controller. If the exact measurements of the states are not available, a state observer should be designed to estimate the states at each sampling time.

Figure 3.2 shows the basic strategy of predictive control for a single-input, single-output (SISO) plant. We assume a discrete-time setting. Integer k represents the current time instant. The measured latest plant output $y(k)$ and the previous plant outputs $y(k-1)$, $y(k-2)$, ..., are known, and are shown by the filled circles in the upper segment of Figure 3.2. The previous moves of the controller are shown by the filled circles in the lower segment of Figure 3.2. The control input will only change at the time instants $k, k+1, \dots, k+H_u-1$, and will remain constant after that. At the current sampling instant k , we need to calculate the next control move $u(k)$. The prediction horizon H_p is the number of control intervals over which the outputs are to be optimized. The control horizon H_u sets the number of control intervals over which the manipulated variables are to be optimized.

MPC may be considered an open-loop control design procedure where at each sampling time k , plant measurements are obtained, and an internal model of the plant is used to predict the future outputs of the system over the prediction horizon. (Note: The feedback comes implicitly by way of plant measurements used here.) Using these predictions, the future control actions $\Delta u(k+i|k)$, $i = 0, 1, \dots, H_u - 1$, are computed by minimizing a cost function. The cost function usually is a quadratic function that is indicative of the desired performance over the considered horizon. Once the future optimal moves are chosen, only the first control move $u(k)$ is sent to the plant. The plant operates with this constant input until the next sampling instant. Then the entire cycle of output measurement, prediction, and optimal input trajectory determination is repeated at each future sampling period. Since the horizons remain to be of the same length as before, but slide along by one sampling interval at each step, this

method of controlling a plant is often called a receding horizon strategy. Reformulation at each sampling instant is essential for achieving good closed-loop system performance. The predictions made during the optimization stage are not perfect due to the model errors and unmeasured plant disturbances. Periodic measurement feedback allows the controller to correct for this error and for unexpected disturbances.

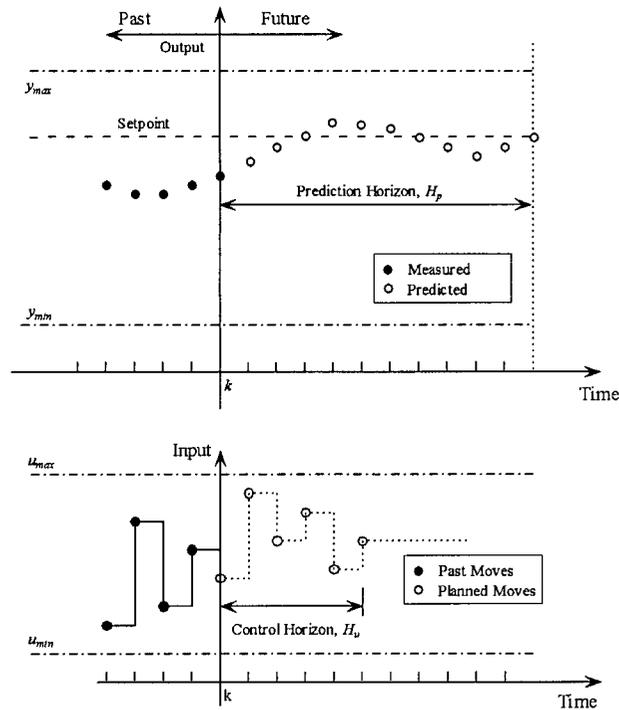


Figure 3.2 Illustration of the strategy of SISO MPC at k th sampling instant.

The basic structure shown in Figure 3.3 is used to implement the MPC. The inputs to the MPC controller are the references (target values for the outputs) and the measured plant outputs. The MPC controller consists of an internal model and an optimizer. The purpose of the internal model is to predict the future behavior of the plant. It can be linear or nonlinear. We will use a discrete-time, linear, state-space internal model in the present work. The reason for using a linear model is that the optimization problem of linear MPC is always convex, and this will guarantee termination of the optimization problem. The optimizer of the MPC is an optimization algorithm that minimizes a user defined cost function. The main purpose of the regulator mode of MPC is to hold the plant outputs at the reference values by adjusting the manipulated variables while satisfying all the constraints. The optimizer will guarantee that at

each time instance optimal manipulated variables are obtained to control the plant.

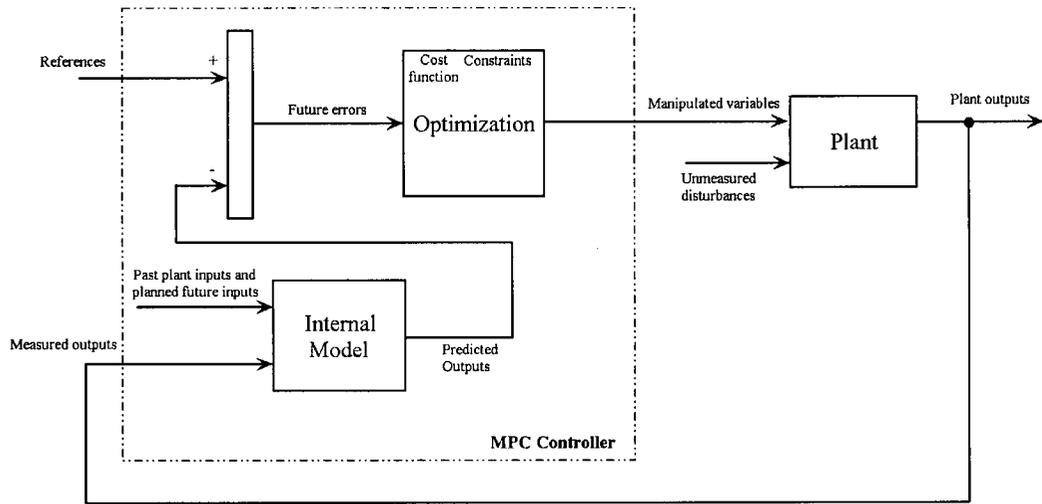


Figure 3.3 Basic structure of MPC.

3.2.2 State-space Model MPC Formulation

For motion control of a flexible-link robotic manipulator, we assume the internal model of the MPC to be a linearized, discrete-time, state-space model of the form

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) \end{aligned} \quad (3.1)$$

In Section 2.4 of Chapter 2, we have developed a linear, time-invariant (LTI) continuous-time state-space model of a planar n -link flexible manipulator in the following form:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) \end{aligned} \quad (3.2)$$

A discrete-time model can be obtained by converting the continuous-time model using zero-order hold on the inputs at the sampling instants, and a sample time period of T_s seconds. Accordingly, the control inputs are piecewise constant over the sampling period T_s ; thus

$$\mathbf{u}(kT_s) = \mathbf{u}[(k+1)T_s] \quad (3.3)$$

The solution of equation (3.2) is

$$\mathbf{x}(t) = e^{At} \mathbf{x}(0) + \int_0^t e^{A(t-\tau)} \mathbf{B} \mathbf{u}(\tau) d\tau \quad (3.4)$$

At $t = kT_s$, the solution is

$$\mathbf{x}(kT_s) = e^{AkT_s} \mathbf{x}(0) + \int_0^{kT_s} e^{A(kT_s-\tau)} \mathbf{B} \mathbf{u}(\tau) d\tau \quad (3.5)$$

At $t = (k+1)T_s$, the solution is

$$\mathbf{x}[(k+1)T_s] = e^{A(kT_s+T_s)} \mathbf{x}(0) + \int_0^{(kT_s+T_s)} e^{A(kT_s+T_s-\tau)} \mathbf{B} \mathbf{u}(\tau) d\tau \quad (3.6)$$

By using equations (3.4) and (3.5), equation (3.6) can be expressed as

$$\begin{aligned} \mathbf{x}[(k+1)T_s] &= e^{A(kT_s+T_s)} \mathbf{x}(0) + \int_0^{(kT_s+T_s)} e^{A(kT_s+T_s-\tau)} \mathbf{B} \mathbf{u}(\tau) d\tau \\ &= e^{AT_s} e^{AkT_s} \mathbf{x}(0) + e^{AT_s} \int_0^{(kT_s+T_s)} e^{A(kT_s-\tau)} \mathbf{B} \mathbf{u}(\tau) d\tau \\ &= e^{AT_s} \left[e^{AkT_s} \mathbf{x}(0) + \int_0^{kT_s} e^{A(kT_s-\tau)} \mathbf{B} \mathbf{u}(\tau) d\tau \right] + e^{AT_s} \int_{kT_s}^{(kT_s+T_s)} e^{A(kT_s-\tau)} \mathbf{B} \mathbf{u}(\tau) d\tau \\ &= e^{AT_s} \mathbf{x}(kT_s) + \int_{kT_s}^{(kT_s+T_s)} e^{A(kT_s+T_s-\tau)} d\tau \mathbf{B} \mathbf{u}(kT_s) \\ &= e^{AT_s} \mathbf{x}(kT_s) + \int_0^{T_s} e^{A\tau} d\tau \mathbf{B} \mathbf{u}(kT_s) \end{aligned} \quad (3.7)$$

Let $kT_s \rightarrow k$. We have

$$\begin{aligned} \mathbf{x}(k+1) &= e^{AT_s} \mathbf{x}(k) + \int_0^{T_s} e^{A\tau} d\tau \mathbf{B} \mathbf{u}(k) \\ &= \mathbf{A} \mathbf{x}(k) + \mathbf{B} \mathbf{u}(k) \end{aligned} \quad (3.8)$$

where

$$\begin{aligned} \mathbf{A} &= e^{AT_s} \\ \mathbf{B} &= \int_0^{T_s} e^{A\tau} d\tau \end{aligned} \quad (3.9)$$

Using equation (3.8), we can convert the continuous-time model to a discrete-time one. Note that matrix \mathbf{C} remains the same in the discrete-time model.

We define the cost function of the MPC controller to be

$$J = \sum_{i=1}^{H_p} \|\hat{\mathbf{y}}(k+i|k) - \mathbf{r}(k+i)\|_{\mathbf{Q}(i)}^2 + \sum_{i=1}^{H_u} \|\Delta \hat{\mathbf{u}}(k+i-1|k)\|_{\mathbf{R}(i)}^2 \quad (3.10)$$

where H_p is the prediction horizon, H_u is the control horizon, $\mathbf{Q}(i)$ is the tracking error weight matrix (or output weight matrix), and $\mathbf{R}(i)$ is the control penalty weight matrix (or input rate weight matrix). We assume both $\mathbf{Q}(i)$ and $\mathbf{R}(i)$ to be positive semi-definite diagonal matrices. Also “ $k+i|k$ ” denotes the value predicted for time $k+i$ based on the information available up to time k . The notation $\hat{\mathbf{y}}(k+i|k)$ indicates the predicted outputs, and $\Delta\hat{\mathbf{u}}(k+i-1|k)$ denotes the future control moves. The cost function J penalizes deviations of the predicted controlled output $\hat{\mathbf{y}}(k+i|k)$ from a reference trajectory $r(k+i|k)$. The following input and output constraints are used:

$$\begin{aligned} \mathbf{u}_{j\min} \leq \mathbf{u}_j(k+i|k) \leq \mathbf{u}_{j\max} \\ \mathbf{y}_{j\min} \leq \mathbf{y}_j(k+i|k) \leq \mathbf{y}_{j\max} \quad i = 0, \dots, H_p \end{aligned} \quad (3.11)$$

where the subscript “ $()_j$ ” denotes the j -th component of a vector, and \mathbf{u}_{\max} and \mathbf{u}_{\min} are the vectors that contain the maximum and minimum control inputs for each joint of the flexible-link robotic manipulator. In order to reduce the link vibrations we assume the deflections for each link to be constrained, and $\mathbf{y}_{j\max}$ and $\mathbf{y}_{j\min}$ denote the vectors that include the maximum and minimum link deflections. The MPC control action at time k is obtained by solving the optimization problem

$$\min_{\Delta\hat{\mathbf{u}}(k|k), \dots, \Delta\hat{\mathbf{u}}(k+H_u-1|k)} J \quad (3.12)$$

with the constraints given in equation (3.11). At time k the control inputs to the plant (robotic manipulator) are

$$\mathbf{u}(k) = \mathbf{u}(k-1) + \Delta\mathbf{u}(k) = \mathbf{u}(k-1) + \Delta\hat{\mathbf{u}}(k|k)^* \quad (3.13)$$

where $\Delta\hat{\mathbf{u}}(k|k)^*$ is the first element of the optimal control input sequence. The remaining samples $\Delta\hat{\mathbf{u}}(k+1|k)^*$ are discarded, and a new optimization problem based on the newly measured outputs $\mathbf{y}(k+1)$ is solved at the next sampling step $k+1$.

3.2.3 Prediction

In order to solve the predictive control problem, we have to compute the predicted values of

the controlled variables $\hat{\mathbf{y}}(k+i|k)$ in equation (3.10), from our best estimate of the current plant state $\hat{\mathbf{x}}(k|k)$, the latest inputs $\mathbf{u}(k-1)$, and the assumed future input changes $\Delta\hat{\mathbf{u}}(k+i-1|k)$. The discrete-time model, given by the state-space model in equation (3.1), is used as the internal model of the MPC for the plant output prediction. The details are given next. Here we assume that the full state vector $\mathbf{x}(k)$ is available. If the full state vector is not available, a state observer has to be designed to estimate the state vector from the measured output $\mathbf{y}(k)$. Then, $\mathbf{x}(k)$ has to be replaced by $\hat{\mathbf{x}}(k|k)$, in the following prediction equations. At current time k , the control inputs $\mathbf{u}(k)$ are not known. We have to use $\hat{\mathbf{u}}(k|k)$ rather than $\mathbf{u}(k)$ in the prediction. The internal model of the plant gives the one-step-ahead prediction:

$$\begin{aligned}\hat{\mathbf{x}}(k+1|k) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\hat{\mathbf{u}}(k|k) \\ \hat{\mathbf{y}}(k+1|k) &= \mathbf{C}\hat{\mathbf{x}}(k+1|k) = \mathbf{C}\mathbf{A}\mathbf{x}(k) + \mathbf{C}\mathbf{B}\hat{\mathbf{u}}(k|k)\end{aligned}\quad (3.14)$$

We can find the predicted plant outputs in the prediction horizon by iterating this internal model; thus

$$\begin{aligned}\hat{\mathbf{x}}(k+2|k) &= \mathbf{A}\hat{\mathbf{x}}(k+1|k) + \mathbf{B}\hat{\mathbf{u}}(k+1|k) \\ &= \mathbf{A}^2\mathbf{x}(k) + \mathbf{A}\mathbf{B}\hat{\mathbf{u}}(k|k) + \mathbf{B}\hat{\mathbf{u}}(k+1|k) \\ \hat{\mathbf{y}}(k+2|k) &= \mathbf{C}\mathbf{A}^2\mathbf{x}(k) + \mathbf{C}\mathbf{A}\mathbf{B}\hat{\mathbf{u}}(k|k) + \mathbf{C}\mathbf{B}\hat{\mathbf{u}}(k+1|k) \\ &\vdots \\ \hat{\mathbf{x}}(k+i|k) &= \mathbf{A}^i\mathbf{x}(k) + \mathbf{A}^{i-1}\mathbf{B}\hat{\mathbf{u}}(k|k) + \cdots + \mathbf{B}\hat{\mathbf{u}}(k+i-1|k) \\ \hat{\mathbf{y}}(k+i|k) &= \mathbf{C}\mathbf{A}^i\mathbf{x}(k) + \mathbf{C}\mathbf{A}^{i-1}\mathbf{B}\hat{\mathbf{u}}(k|k) + \cdots + \mathbf{C}\mathbf{B}\hat{\mathbf{u}}(k+i-1|k) \\ &\vdots \\ \hat{\mathbf{x}}(k+H_p|k) &= \mathbf{A}^{H_p}\mathbf{x}(k) + \mathbf{A}^{H_p-1}\mathbf{B}\hat{\mathbf{u}}(k|k) + \cdots + \mathbf{B}\hat{\mathbf{u}}(k+H_p-1|k) \\ \hat{\mathbf{y}}(k+H_p|k) &= \mathbf{C}\mathbf{A}^{H_p}\mathbf{x}(k) + \mathbf{C}\mathbf{A}^{H_p-1}\mathbf{B}\hat{\mathbf{u}}(k|k) + \cdots + \mathbf{C}\mathbf{B}\hat{\mathbf{u}}(k+H_p-1|k)\end{aligned}\quad (3.15)$$

We assume that the inputs will change only at times $k, k+1, \dots, k+H_u-1$, and will remain constant after that. Specifically we have

$$\hat{\mathbf{u}}(k+i|k) = \hat{\mathbf{u}}(k+H_u-1|k) \quad \text{for } H_u \leq i \leq H_p-1 \quad (3.16)$$

At time k , the latest control inputs $\mathbf{u}(k-1)$ are known. We have

$$\begin{aligned}
\hat{\mathbf{u}}(k|k) &=_{\Delta} \hat{\mathbf{u}}(k|k) + \mathbf{u}(k-1) \\
\hat{\mathbf{u}}(k+1|k) &=_{\Delta} \hat{\mathbf{u}}(k+1|k) + \hat{\mathbf{u}}(k|k) \\
&=_{\Delta} \hat{\mathbf{u}}(k+1|k) +_{\Delta} \hat{\mathbf{u}}(k|k) + \mathbf{u}(k-1) \\
&\vdots \\
\hat{\mathbf{u}}(k+i|k) &=_{\Delta} \hat{\mathbf{u}}(k+i|k) +_{\Delta} \hat{\mathbf{u}}(k+i-1|k) + \cdots +_{\Delta} \hat{\mathbf{u}}(k|k) + \mathbf{u}(k-1) \\
&\vdots \\
\hat{\mathbf{u}}(k+H_p-1|k) &=_{\Delta} \hat{\mathbf{u}}(k+H_p-1|k) +_{\Delta} \hat{\mathbf{u}}(k+H_p-2|k) + \cdots +_{\Delta} \hat{\mathbf{u}}(k|k) + \mathbf{u}(k-1)
\end{aligned} \tag{3.17}$$

The prediction in terms of $_{\Delta} \mathbf{u}(k+i|k)$ can be obtained by substituting equation (3.17) into (3.14) and (3.15); thus

$$\begin{aligned}
\hat{\mathbf{x}}(k+1|k) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}_\Delta \hat{\mathbf{u}}(k|k) + \mathbf{B}\mathbf{u}(k-1) \\
\hat{\mathbf{y}}(k+1|k) &= \mathbf{C}\mathbf{A}\mathbf{x}(k) + \mathbf{C}\mathbf{B}_\Delta \hat{\mathbf{u}}(k|k) + \mathbf{C}\mathbf{B}\mathbf{u}(k-1) \\
\hat{\mathbf{x}}(k+2|k) &= \mathbf{A}^2\mathbf{x}(k) + \mathbf{A}\mathbf{B}_\Delta \hat{\mathbf{u}}(k|k) + \mathbf{A}\mathbf{B}\mathbf{u}(k-1) \\
&\quad + \mathbf{B}_\Delta \hat{\mathbf{u}}(k+1|k) + \mathbf{B}_\Delta \hat{\mathbf{u}}(k|k) + \mathbf{B}\mathbf{u}(k-1) \\
&= \mathbf{A}^2\mathbf{x}(k) + (\mathbf{B} + \mathbf{A}\mathbf{B})_\Delta \hat{\mathbf{u}}(k|k) + \mathbf{B}_\Delta \hat{\mathbf{u}}(k+1|k) \\
&\quad + (\mathbf{B} + \mathbf{A}\mathbf{B})\mathbf{u}(k-1) \\
\hat{\mathbf{y}}(k+2|k) &= \mathbf{C}\mathbf{A}^2\mathbf{x}(k) + (\mathbf{C}\mathbf{B} + \mathbf{C}\mathbf{A}\mathbf{B})_\Delta \hat{\mathbf{u}}(k|k) + \mathbf{C}\mathbf{B}_\Delta \hat{\mathbf{u}}(k+1|k) \\
&\quad + (\mathbf{C}\mathbf{B} + \mathbf{C}\mathbf{A}\mathbf{B})\mathbf{u}(k-1) \\
&\quad \vdots \\
\hat{\mathbf{x}}(k+H_u|k) &= \mathbf{A}^{H_u}\mathbf{x}(k) + (\mathbf{B} + \mathbf{A}\mathbf{B} + \dots + \mathbf{A}^{H_u-1}\mathbf{B})_\Delta \hat{\mathbf{u}}(k|k) \\
&\quad + \dots + \mathbf{B}_\Delta \hat{\mathbf{u}}(k+H_u-1|k) + (\mathbf{B} + \mathbf{A}\mathbf{B} + \dots + \mathbf{A}^{H_u-1}\mathbf{B})\mathbf{u}(k-1) \\
\hat{\mathbf{y}}(k+H_u|k) &= \mathbf{C}\mathbf{A}^{H_u}\mathbf{x}(k) + (\mathbf{C}\mathbf{B} + \mathbf{C}\mathbf{A}\mathbf{B} + \dots + \mathbf{C}\mathbf{A}^{H_u-1}\mathbf{B})_\Delta \hat{\mathbf{u}}(k|k) \\
&\quad + \dots + \mathbf{C}\mathbf{B}_\Delta \hat{\mathbf{u}}(k+H_u-1|k) + (\mathbf{C}\mathbf{B} + \mathbf{C}\mathbf{A}\mathbf{B} + \dots + \mathbf{C}\mathbf{A}^{H_u-1}\mathbf{B})\mathbf{u}(k-1) \\
\hat{\mathbf{x}}(k+H_u+1|k) &= \mathbf{A}^{H_u+1}\mathbf{x}(k) + (\mathbf{B} + \mathbf{A}\mathbf{B} + \dots + \mathbf{A}^{H_u}\mathbf{B})_\Delta \hat{\mathbf{u}}(k|k) \\
&\quad + \dots + (\mathbf{B} + \mathbf{A}\mathbf{B})_\Delta \hat{\mathbf{u}}(k+H_u-1|k) \\
&\quad + (\mathbf{B} + \mathbf{A}\mathbf{B} + \dots + \mathbf{A}^{H_u}\mathbf{B})\mathbf{u}(k-1) \\
\hat{\mathbf{y}}(k+H_u+1|k) &= \mathbf{C}\mathbf{A}^{H_u+1}\mathbf{x}(k) + (\mathbf{C}\mathbf{B} + \mathbf{C}\mathbf{A}\mathbf{B} + \dots + \mathbf{C}\mathbf{A}^{H_u}\mathbf{B})_\Delta \hat{\mathbf{u}}(k|k) \\
&\quad + \dots + (\mathbf{C}\mathbf{B} + \mathbf{C}\mathbf{A}\mathbf{B})_\Delta \hat{\mathbf{u}}(k+H_u-1|k) \\
&\quad + (\mathbf{C}\mathbf{B} + \mathbf{C}\mathbf{A}\mathbf{B} + \dots + \mathbf{C}\mathbf{A}^{H_u}\mathbf{B})\mathbf{u}(k-1) \\
&\quad \vdots \\
\hat{\mathbf{x}}(k+H_p|k) &= \mathbf{A}^{H_p}\mathbf{x}(k) + (\mathbf{B} + \mathbf{A}\mathbf{B} + \dots + \mathbf{A}^{H_p-1}\mathbf{B})_\Delta \hat{\mathbf{u}}(k|k) \\
&\quad + \dots + (\mathbf{B} + \mathbf{A}\mathbf{B} + \dots + \mathbf{A}^{H_p-H_u}\mathbf{B})_\Delta \hat{\mathbf{u}}(k+H_u-1|k) \\
&\quad + (\mathbf{B} + \mathbf{A}\mathbf{B} + \dots + \mathbf{A}^{H_p-1}\mathbf{B})\mathbf{u}(k-1) \\
\hat{\mathbf{y}}(k+H_p|k) &= \mathbf{C}\mathbf{A}^{H_p}\mathbf{x}(k) + (\mathbf{C}\mathbf{B} + \mathbf{C}\mathbf{A}\mathbf{B} + \dots + \mathbf{C}\mathbf{A}^{H_p-1}\mathbf{B})_\Delta \hat{\mathbf{u}}(k|k) \\
&\quad + \dots + (\mathbf{C}\mathbf{B} + \mathbf{C}\mathbf{A}\mathbf{B} + \dots + \mathbf{C}\mathbf{A}^{H_p-H_u}\mathbf{B})_\Delta \hat{\mathbf{u}}(k+H_u-1|k) \\
&\quad + (\mathbf{C}\mathbf{B} + \mathbf{C}\mathbf{A}\mathbf{B} + \dots + \mathbf{C}\mathbf{A}^{H_p-1}\mathbf{B})\mathbf{u}(k-1)
\end{aligned} \tag{3.18}$$

For a plant with n_x states, n_y measured outputs and n_u inputs, the vector of output predictions up to a horizon H_p can be written in the matrix-vector form

$$\hat{\mathbf{Y}}(k) = \mathbf{S}_x \mathbf{x}(k) + \mathbf{S}_u \mathbf{u}(k-1) + \mathbf{S}_{\Delta u} \Delta \hat{\mathbf{U}}(k) \quad (3.19)$$

where

$$\hat{\mathbf{Y}}(k) = \begin{bmatrix} \hat{y}(k+1|k) \\ \hat{y}(k+2|k) \\ \vdots \\ \hat{y}(k+H_u|k) \\ \hat{y}(k+H_u+1|k) \\ \vdots \\ \hat{y}(k+H_p|k) \end{bmatrix} \in \mathfrak{R}^{n_y H_p \times 1}, \quad \Delta \hat{\mathbf{U}}(k) = \begin{bmatrix} \Delta \hat{u}(k|k) \\ \Delta \hat{u}(k+1|k) \\ \vdots \\ \Delta \hat{u}(k+H_u-1|k) \end{bmatrix} \in \mathfrak{R}^{n_u H_u \times 1} \quad (3.20)$$

$$\mathbf{S}_x = \begin{bmatrix} \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{H_u} \\ \mathbf{CA}^{H_u+1} \\ \vdots \\ \mathbf{CA}^{H_p} \end{bmatrix} \in \mathfrak{R}^{n_y H_p \times n_x}, \quad \mathbf{S}_u = \begin{bmatrix} \mathbf{CB} \\ \mathbf{CB} + \mathbf{CAB} \\ \vdots \\ \sum_{i=0}^{H_u-1} \mathbf{CA}^i \mathbf{B} \\ \sum_{i=0}^{H_u} \mathbf{CA}^i \mathbf{B} \\ \vdots \\ \sum_{i=0}^{H_p-1} \mathbf{CA}^i \mathbf{B} \end{bmatrix} \in \mathfrak{R}^{n_y H_p \times n_u} \quad (3.21)$$

$$\mathbf{S}_{\Delta u} = \begin{bmatrix} \mathbf{CB} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{CB} + \mathbf{CAB} & \mathbf{CB} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ \sum_{i=0}^{H_u-1} \mathbf{CA}^i \mathbf{B} & \sum_{i=0}^{H_u-2} \mathbf{CA}^i \mathbf{B} & \dots & \mathbf{CB} \\ \sum_{i=0}^{H_u} \mathbf{CA}^i \mathbf{B} & \sum_{i=0}^{H_u-1} \mathbf{CA}^i \mathbf{B} & \dots & \mathbf{CB} + \mathbf{CAB} \\ \vdots & \vdots & \vdots & \vdots \\ \sum_{i=0}^{H_p-1} \mathbf{CA}^i \mathbf{B} & \sum_{i=0}^{H_p-2} \mathbf{CA}^i \mathbf{B} & \dots & \sum_{i=0}^{H_p-H_u} \mathbf{CA}^i \mathbf{B} \end{bmatrix} \in \mathfrak{R}^{n_y H_p \times n_u H_u} \quad (3.22)$$

Equation (3.19) is the prediction equation. It can be used to predict the future plant outputs.

The matrices S_x, S_u and $S_{\Delta u}$ are the constant matrices that depend on the internal model (matrices A, B, C) and the MPC horizons (H_p, H_u) only. In order to compute the future prediction at time k , we need the current state vector $\mathbf{x}(k)$ and the past input vector $\mathbf{u}(k-1)$ of the plant. For the flexible link robotic manipulator system, the plant states are the joint angles $(\theta_1, \dots, \theta_n)$, the link deflection variables $(\delta_{11}, \dots, \delta_{1n_{m,1}}, \dots, \delta_{n1}, \dots, \delta_{nn_{m,n}})$ and their derivatives. We assume the measured system outputs to be the joint angles and link deflections. The derivative terms of the plant states are not measured directly. This means the full state vector $\mathbf{x}(k)$ is not available, and an observer has to be designed to estimate the state vector. Note that if a state observer is used, still the form of the output prediction equation (3.19) remains the same, but $\mathbf{x}(k)$ has to be replaced by $\hat{\mathbf{x}}(k|k)$, which is the estimation of state vector at time k .

The state observer is a copy of the plant (equation (3.1)) with feedback from the measured plant outputs, through the gain matrix L , to correct the state estimation. The equations of the observer are given below.

Predicted output computation:

$$\hat{\mathbf{y}}(k|k-1) = \mathbf{C}\hat{\mathbf{x}}(k|k-1) \quad (3.23)$$

Measurement update:

$$\begin{aligned} \hat{\mathbf{x}}(k|k) &= \hat{\mathbf{x}}(k|k-1) + \mathbf{L}'[\mathbf{y}(k) - \hat{\mathbf{y}}(k|k-1)] \\ &= \hat{\mathbf{x}}(k|k-1) + \mathbf{L}'[\mathbf{y}(k) - \mathbf{C}\hat{\mathbf{x}}(k|k-1)] \end{aligned} \quad (3.24)$$

Time update:

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{A}\hat{\mathbf{x}}(k|k) + \mathbf{B}\mathbf{u}(k) \quad (3.25)$$

The notation used in these equations is

- $\hat{\mathbf{x}}(k|k-1)$ is the estimate of $\mathbf{x}(k)$ given past measurements up to $\mathbf{y}(k-1)$.
- $\hat{\mathbf{x}}(k|k)$ is the update estimate of $\mathbf{x}(k)$ based on the current measurements $\mathbf{y}(k)$.
- \mathbf{L}' is the innovation gain.

Given the past state estimate $\hat{\mathbf{x}}(k-1|k-1)$, the time update predicts the state value at the next sample k (one-step-ahead predictor). The measurement update then adjusts this

prediction based on the new measurement $\mathbf{y}(k)$. The correction term is a function of the *innovation*, which is the discrepancy between the measured and predicted values of $\mathbf{y}(k)$:

$$\mathbf{y}(k) - \hat{\mathbf{y}}(k|k-1) = \mathbf{y}(k) - \mathbf{C}\hat{\mathbf{x}}(k|k-1) \quad (3.26)$$

By combining equations (3.23), (3.24) and (3.25), the overall state observer is obtained as

$$\begin{aligned} \hat{\mathbf{x}}(k+1|k) &= \mathbf{A}(\mathbf{I} - \mathbf{L}'\mathbf{C})\hat{\mathbf{x}}(k|k-1) + \mathbf{B}\mathbf{u}(k) + \mathbf{A}\mathbf{L}'\mathbf{y}(k) \\ &= (\mathbf{A} - \mathbf{L}\mathbf{C})\hat{\mathbf{x}}(k|k-1) + \mathbf{B}\mathbf{u}(k) + \mathbf{L}\mathbf{y}(k) \end{aligned} \quad (3.27)$$

where $\mathbf{L} = \mathbf{A}\mathbf{L}'$. This is a stable system if the eigenvalues of $\mathbf{A} - \mathbf{L}\mathbf{C}$ are within the unit circle. Furthermore, if we define the state estimation error vector as $\mathbf{e}(k) = \mathbf{x}(k) - \hat{\mathbf{x}}(k|k-1)$, then using equations (3.1) and (3.27) we have

$$\mathbf{e}(k+1) = (\mathbf{A} - \mathbf{L}\mathbf{C})\mathbf{e}(k) \quad (3.28)$$

which shows that the state estimation error converges to zero if the observer is stable, at a rate determined by the eigenvalues of $\mathbf{A} - \mathbf{L}\mathbf{C}$. If the pair (\mathbf{A}, \mathbf{C}) is observable, then given an arbitrary set of locations in the complex plane, a gain matrix \mathbf{L} exists which places the observer eigenvalues at these locations. The gain matrix \mathbf{L} can be chosen to minimize the steady-state covariance of the estimation error with a known noise covariance matrix. This observer is then known as a *Kalman filter*. We can design the steady-state *Kalman filter* described above with the function *kalman* in the Control System Toolbox of Matlab (Mathworks, 2002).

3.2.4 MPC Computation and Constraint Handling

In this section we will solve the optimization problem associated with model predictive control (MPC). The cost function which we must minimize is given in equation (3.10). We can rewrite this cost function in the matrix form by using the notations defined in equations (3.20) through (3.22), as

$$J = \left\| \hat{\mathbf{Y}}(k) - \mathbf{T}(k) \right\|_{\mathbf{Q}}^2 + \left\| \Delta \hat{\mathbf{U}}(k) \right\|_{\mathbf{R}}^2 \quad (3.29)$$

where

$$\hat{\mathbf{Y}}(k) = \begin{bmatrix} \hat{\mathbf{y}}(k+1|k) \\ \vdots \\ \hat{\mathbf{y}}(k+H_p|k) \end{bmatrix}, \quad \mathbf{T}(k) = \begin{bmatrix} \mathbf{r}(k+1) \\ \vdots \\ \mathbf{r}(k+H_p) \end{bmatrix}, \quad \Delta\hat{\mathbf{U}}(k) = \begin{bmatrix} \Delta\hat{\mathbf{u}}(k|k) \\ \vdots \\ \Delta\hat{\mathbf{u}}(k+H_u-1|k) \end{bmatrix} \quad (3.30)$$

and the weighting matrices are given by

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}(1) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}(2) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{Q}(H_p) \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} \mathbf{R}(0) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{R}(1) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{R}(H_u-1) \end{bmatrix} \quad (3.31)$$

From equation (3.19), the predicted system outputs using a state observer are

$$\hat{\mathbf{Y}}(k) = \mathbf{S}_x \hat{\mathbf{x}}(k|k) + \mathbf{S}_u \mathbf{u}(k-1) + \mathbf{S}_{\Delta u} \Delta\hat{\mathbf{U}}(k) \quad (3.32)$$

Define

$$\begin{aligned} \Delta\mathbf{T}(k) &= \mathbf{T}(k) - \hat{\mathbf{Y}}(k) + \mathbf{S}_{\Delta u} \Delta\hat{\mathbf{U}}(k) \\ &= \mathbf{T}(k) - \mathbf{S}_x \hat{\mathbf{x}}(k|k) - \mathbf{S}_u \mathbf{u}(k-1) \end{aligned} \quad (3.33)$$

We will call $\Delta\mathbf{T}(k)$ a 'tracking error.' This is the difference between the future plant reference response and the 'free response' of the system, which is the response that would occur over the prediction horizon if no input changes were made ($\Delta\mathbf{U}(k) = \mathbf{0}$).

We assume the weighting matrices are diagonal, also $\mathbf{Q} \geq 0$ and $\mathbf{R} \geq 0$. Their 'square-roots' matrices are

$$\mathbf{Q} = \mathbf{S}_Q^T \mathbf{S}_Q, \quad \mathbf{R} = \mathbf{S}_R^T \mathbf{S}_R \quad (3.34)$$

By using equations (3.33) and (3.34), the cost function of the system as given by equation (3.29) becomes

$$\begin{aligned} J &= \left\| \hat{\mathbf{Y}}(k) - \mathbf{T}(k) \right\|_{\mathbf{Q}}^2 + \left\| \Delta\hat{\mathbf{U}}(k) \right\|_{\mathbf{R}}^2 \\ &= \left[\hat{\mathbf{Y}}(k) - \mathbf{T}(k) \right]^T \mathbf{S}_Q^T \mathbf{S}_Q \left[\hat{\mathbf{Y}}(k) - \mathbf{T}(k) \right] + \Delta\hat{\mathbf{U}}(k)^T \mathbf{S}_R^T \mathbf{S}_R \Delta\hat{\mathbf{U}}(k) \\ &= \left\| \begin{bmatrix} \mathbf{S}_Q \left[\hat{\mathbf{Y}}(k) - \mathbf{T}(k) \right] \\ \mathbf{S}_R \Delta\hat{\mathbf{U}}(k) \end{bmatrix} \right\|^2 = \left\| \begin{bmatrix} \mathbf{S}_Q \left[\mathbf{S}_{\Delta u} \Delta\hat{\mathbf{U}}(k) - \Delta\mathbf{T}(k) \right] \\ \mathbf{S}_R \Delta\hat{\mathbf{U}}(k) \end{bmatrix} \right\|^2 \end{aligned} \quad (3.35)$$

Equation (3.35) indicates that the cost function J is equal to the squared ‘length’ of vector

$$\begin{bmatrix} \mathbf{S}_Q [\mathbf{S}_{\Delta u} \Delta \hat{\mathbf{U}}(k) - \Delta \mathbf{T}(k)] \\ \mathbf{S}_R \Delta \hat{\mathbf{U}}(k) \end{bmatrix} \quad (3.36)$$

For the unconstrained MPC, there is no constraint on the inputs and outputs of the system. The optimal value of $\Delta \hat{\mathbf{U}}(k)$ is $\Delta \hat{\mathbf{U}}^*(k)$ which minimize the cost function J given by equation (3.35). So $\Delta \hat{\mathbf{U}}^*(k)$ is the ‘least-squares’ solution of the equation

$$\begin{bmatrix} \mathbf{S}_Q [\mathbf{S}_{\Delta u} \Delta \hat{\mathbf{U}}(k) - \Delta \mathbf{T}(k)] \\ \mathbf{S}_R \Delta \hat{\mathbf{U}}(k) \end{bmatrix} = 0 \quad (3.37)$$

or

$$\begin{bmatrix} \mathbf{S}_Q \mathbf{S}_{\Delta u} \\ \mathbf{S}_R \end{bmatrix} \Delta \hat{\mathbf{U}}(k) = \begin{bmatrix} \mathbf{S}_Q \Delta \mathbf{T}(k) \\ \mathbf{0} \end{bmatrix} \quad (3.38)$$

The ‘least-squares’ solution of a set of algebraic equations of the form $\mathbf{Ax} = \mathbf{b}$ is obtained by minimizing the least-squared performance function:

$$J = \min_{\mathbf{x}} (\mathbf{Ax} - \mathbf{b})^T (\mathbf{Ax} - \mathbf{b}) \quad (3.39)$$

By differentiating J with respect to \mathbf{x} and equating to zero will yield \mathbf{x}^* , which is the ‘least-squares’ solution of the equations $\mathbf{Ax} = \mathbf{b}$, as

$$\mathbf{x}^* = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} \quad (3.40)$$

The numerical solution of $\mathbf{Ax} = \mathbf{b}$ can be obtained in a least-squares sense using the ‘QR’ algorithm. In Matlab[®] this solution is obtained as $\mathbf{x}^* = \mathbf{A} \setminus \mathbf{b}$.

By using equations (3.40), (3.34) and (3.33), we have the optimal set of future input moves as

$$\begin{aligned}
\Delta \hat{\mathbf{U}}^*(k) &= \left(\begin{bmatrix} \mathbf{S}_Q \mathbf{S}_{\Delta u} \\ \mathbf{S}_R \end{bmatrix}^T \begin{bmatrix} \mathbf{S}_Q \mathbf{S}_{\Delta u} \\ \mathbf{S}_R \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{S}_Q \mathbf{S}_{\Delta u} \\ \mathbf{S}_R \end{bmatrix}^T \begin{bmatrix} \mathbf{S}_Q \Delta \mathbf{T}(k) \\ \mathbf{0} \end{bmatrix} \\
&= \left\{ \begin{bmatrix} \mathbf{S}_{\Delta u}^T \mathbf{S}_Q & \mathbf{S}_R^T \end{bmatrix} \begin{bmatrix} \mathbf{S}_Q \mathbf{S}_{\Delta u} \\ \mathbf{S}_R \end{bmatrix} \right\}^{-1} \begin{bmatrix} \mathbf{S}_{\Delta u}^T \mathbf{S}_Q & \mathbf{S}_R^T \end{bmatrix} \begin{bmatrix} \mathbf{S}_Q \Delta \mathbf{T}(k) \\ \mathbf{0} \end{bmatrix} \\
&= \left(\mathbf{S}_{\Delta u}^T \mathbf{S}_Q \mathbf{S}_Q \mathbf{S}_{\Delta u} + \mathbf{S}_R^T \mathbf{S}_R \right)^{-1} \mathbf{S}_{\Delta u}^T \mathbf{S}_Q \mathbf{S}_Q \Delta \mathbf{T}(k) \\
&= \left(\mathbf{S}_{\Delta u}^T \mathbf{Q} \mathbf{S}_{\Delta u} + \mathbf{R} \right)^{-1} \mathbf{S}_{\Delta u}^T \mathbf{Q} \Delta \mathbf{T}(k) \\
&= \left(\mathbf{S}_{\Delta u}^T \mathbf{Q} \mathbf{S}_{\Delta u} + \mathbf{R} \right)^{-1} \mathbf{S}_{\Delta u}^T \mathbf{Q} [\mathbf{T}(k) - \mathbf{S}_x \hat{\mathbf{x}}(k|k) - \mathbf{S}_u \mathbf{u}(k-1)]
\end{aligned} \tag{3.41}$$

From this result we can see that for the unconstrained predictive control problem, an analytical solution can be obtained by using equation (3.41). The numerical solution of the unconstrained predict control problem can be obtained by using the Matlab notation:

$$\Delta \hat{\mathbf{U}}^*(k) = \begin{bmatrix} \mathbf{S}_Q \mathbf{S}_{\Delta u} \\ \mathbf{S}_R \end{bmatrix} \setminus \begin{bmatrix} \mathbf{S}_Q \Delta \mathbf{T}(k) \\ \mathbf{0} \end{bmatrix} \tag{3.42}$$

According to the receding horizon strategy of MPC, only the first step of the computed control move in equation (3.41) is applied to the plant at time k . If the number of plant inputs is n_u , we just use the first n_u rows of the vector $\Delta \hat{\mathbf{U}}^*(k)$, that is

$$\begin{aligned}
\Delta \mathbf{u}(k) &= \Delta \hat{\mathbf{u}}(k|k)^* \\
&= \begin{bmatrix} I_{n_u \times n_u}, \mathbf{0}_{n_u \times n_u(H_u-1)} \end{bmatrix} \Delta \hat{\mathbf{U}}^*(k) \\
&= \begin{bmatrix} I_{n_u \times n_u}, \mathbf{0}_{n_u \times n_u(H_u-1)} \end{bmatrix} \left(\mathbf{S}_{\Delta u}^T \mathbf{Q} \mathbf{S}_{\Delta u} + \mathbf{R} \right)^{-1} \mathbf{S}_{\Delta u}^T \mathbf{Q} [\mathbf{T}(k) - \mathbf{S}_x \hat{\mathbf{x}}(k|k) - \mathbf{S}_u \mathbf{u}(k-1)] \\
&= \mathbf{K}_{MPC} [\mathbf{T}(k) - \mathbf{S}_x \hat{\mathbf{x}}(k|k) - \mathbf{S}_u \mathbf{u}(k-1)] \\
&= \mathbf{K}_{MPC} \Delta \mathbf{T}(k)
\end{aligned} \tag{3.43}$$

where $I_{n_u \times n_u}$ is the $n_u \times n_u$ identity matrix, and $\mathbf{0}_{n_u \times n_u(H_u-1)}$ is the $n_u \times n_u(H_u-1)$ zero (null) matrix.

The MPC controller gain matrix \mathbf{K}_{MPC} is defined as

$$\mathbf{K}_{MPC} = \begin{bmatrix} I_{n_u \times n_u}, \mathbf{0}_{n_u \times n_u(H_u-1)} \end{bmatrix} \left(\mathbf{S}_{\Delta u}^T \mathbf{Q} \mathbf{S}_{\Delta u} + \mathbf{R} \right)^{-1} \mathbf{S}_{\Delta u}^T \mathbf{Q} \tag{3.44}$$

Here \mathbf{K}_{MPC} is a constant matrix which depends only on the internal model (matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$),

the weighting matrices \mathbf{Q} and \mathbf{R} , and the controller horizons (H_p, H_u) .

The unconstrained MPC control input vector at time k is

$$\begin{aligned} \mathbf{u}(k) &= \mathbf{u}(k-1) + \Delta\mathbf{u}(k) \\ &= \mathbf{u}(k-1) + \mathbf{K}_{MPC} [\mathbf{T}(k) - \mathbf{S}_x \hat{\mathbf{x}}(k|k) - \mathbf{S}_u \mathbf{u}(k-1)] \end{aligned} \quad (3.45)$$

The numerical value of \mathbf{K}_{MPC} can be computed using the *Matlab* notation ‘\’ and the operator ‘:’ to pick out the first n_u rows of the solution:

$$\begin{aligned} \mathbf{K}_{full} &= \begin{bmatrix} \mathbf{S}_Q \mathbf{S}_{\Delta u} \\ \mathbf{S}_R \end{bmatrix} \setminus \begin{bmatrix} \mathbf{S}_Q \\ \mathbf{0} \end{bmatrix} \\ \mathbf{K}_{MPC} &= \mathbf{K}_{full}(1:n_u,:) \end{aligned} \quad (3.46)$$

The structure of MPC controller with no constraints is shown in Figure 3.4. From this figure we notice that the unconstrained MPC controller is a linear time-invariant system (LTI).

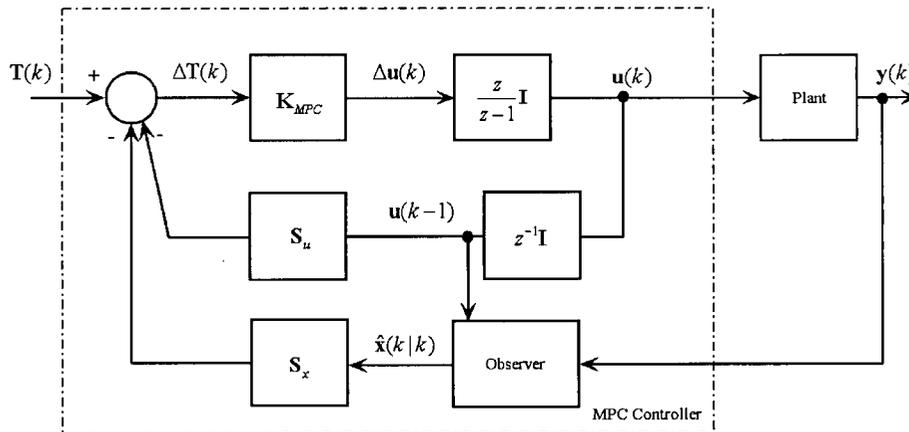


Figure 3.4 Unconstrained MPC controller structure.

Now we will deal with the case where the constraints are present. Recall that for a flexible-link robotic manipulator the constraints are given by equation (3.11). In this case the simple ‘linear least-squares’ solution of unconstrained MPC has to be replaced by a ‘constrained least-squares’ solution. The closed-form solution for the constrained MPC cannot be found, and some form of an iterative optimization algorithm must be employed to find the

numerical solution. It can be shown that if the constraints are in the form of linear inequalities, then the constrained MPC problem becomes a quadratic programming (QP) optimization problem (Maciejowski, 2002). If the constraints are inactive, the solution of the predictive controller is the same as that in the unconstrained case. But if the constraints become active, then the MPC controller becomes nonlinear. The QP problem has to be solved at each time step. For fast dynamic systems with very low sampling rates (e.g., a flexible-link robotic manipulator system), the standard QP algorithm may not be efficient enough to find an optimal solution at each time step. Next we will develop a new computationally efficient ‘anti-windup’ MPC to deal with the constraints of the flexible-link robotic manipulator system.

The new ‘anti-windup’ MPC controller is based on the unconstrained MPC controller shown in Figure 3.4, with an additional constraint-handling strategy. The input moves $\Delta \hat{\mathbf{u}}(k|k)^*$ are computed without taking any constraints into account. This is a computationally efficient algorithm because we have the closed-form solution of the controller as given in equation (3.43).

The input constraints for the controller are the minimum and maximum joint torques for each link:

$$\begin{aligned} \mathbf{u}_{\min} &\leq \mathbf{u}(k) \leq \mathbf{u}_{\max} \\ \mathbf{u}_{\min} &< 0, \mathbf{u}_{\max} > 0 \end{aligned} \quad (3.47)$$

The output constraints are the minimum and maximum link deflections for each link:

$$\mathbf{y}_{\min} \leq \mathbf{y}(k) \leq \mathbf{y}_{\max} \quad (3.48)$$

The constraints of the system are handled by using fuzzy rule-based auto-tuning of the unconstrained MPC controller which is developed in Section 3.4. From the structure of the unconstrained MPC controller, we can see that the control input vector $\mathbf{u}(k)$ and the plant output vector $\mathbf{y}(k)$ depend on the gain matrix \mathbf{K}_{MPC} . The gain matrix \mathbf{K}_{MPC} can be tuned by adjusting the weighting matrices \mathbf{Q} and \mathbf{R} , and the controller horizons (H_p, H_u) as given in equation (3.44). It follows that the basic idea of the constraint-handling strategy of the ‘anti-windup’ MPC controller is to use a fuzzy rule-based auto-tuner to adjust the tuning parameters of the unconstrained MPC controller in order to make sure that the input and

output constraints are satisfied. We will use the MPC parameters that satisfy $\mathbf{u}_{\min} < \mathbf{u}(k) < \mathbf{u}_{\max}$ and $\mathbf{y}_{\min} < \mathbf{y}(k) < \mathbf{y}_{\max}$. The constraints are not active, so the unconstrained MPC is equal to the constrained MPC with the constraints within the actual system constraints ($\mathbf{u}_{\min} \leq \mathbf{u}(k) \leq \mathbf{u}_{\max}$, $\mathbf{y}_{\min} \leq \mathbf{y}(k) \leq \mathbf{y}_{\max}$). This means that the ‘anti-windup’ MPC controller will not generate the true optimal solution in general but we will provide a sub-optimal solution. We can make this sub-optimal solution as close as possible to the true optimal solution by adjusting the tuning parameters of the controller. Details of how to tune these parameters are discussed in Section 3.4. The MPC is designed using the tuned parameters. This will guarantee that the system constraints are satisfied.

The move that is actually applied to the plant is

$$\Delta \mathbf{u}(k) = \begin{cases} \min(\Delta \hat{\mathbf{u}}(k|k)^*, \mathbf{u}_{\max} - \mathbf{u}(k-1)) & \text{if } \Delta \hat{\mathbf{u}}(k|k)^* > 0 \\ \max(\Delta \hat{\mathbf{u}}(k|k)^*, \mathbf{u}_{\min} - \mathbf{u}(k-1)) & \text{if } \Delta \hat{\mathbf{u}}(k|k)^* < 0 \end{cases} \quad (3.49)$$

Equation (3.49) will ensure that the input constraints in equation (3.47) are always satisfied. The same move must be applied to the internal model of MPC. This not only helps to keep the predictions accurate, but also avoids problems analogous to ‘integrator wind-up.’ The fact that the MPC controller is aware of the input constraints, in particular of the actuator saturation constraints, and never generates input signals that attempt to violate them, removes the problem of ‘integrator wind-up.’ This problem occurs when conventional controllers are used, if long-duration set-point errors cause integrator outputs to exceed to the saturation limits, and can result in large overshoots and even instability.

3.2.5 Stability Analysis

In this section we will investigate the stability of the MPC controller which has been developed for the flexible-link robotic manipulator system. If the MPC controller is designed off line, the nominal stability of the closed-loop system can be obtained by tuning the parameters in the problem formulation. Closed-loop stability is much more of an issue if the MPC controller is used in such a way that requires on-line redesign. Predictive control, using the receding horizon idea, is a feedback control strategy. There is therefore a risk that the resulting closed-loop system might be unstable. Even though the performance of the system is

optimized over the prediction horizon, and even though the optimization is repeated at each time step, each optimization does not consider the system response beyond the prediction horizon. This could place the plant in such a state that eventually it will be impossible to stabilize the system due to the constraints on the control input signals. In order to design a MPC with guaranteed nominal stability, we need to introduce the Lyapunov theorem and related definitions.

Equilibrium: The system equation $\mathbf{x}(k+1) = f(\mathbf{x}(k), \mathbf{u}(k))$ has an equilibrium at the state vector \mathbf{x}_0 and the input vector \mathbf{u}_0 if $\mathbf{x}_0 = f(\mathbf{x}_0, \mathbf{u}_0)$. We can always assume that an equilibrium is at $(\mathbf{0}, \mathbf{0})$ by changing of the coordinates according to $\mathbf{z}(k) = \mathbf{x}(k) - \mathbf{x}_0$ and $\mathbf{v}(k) = \mathbf{u}(k) - \mathbf{u}_0$.

Stability: For nonlinear systems one has to consider the stability of a particular equilibrium point. An equilibrium point $(\mathbf{0}, \mathbf{0})$ is stable (in the sense of Lyapunov) if for a small perturbation of the states or inputs, given any $\varepsilon > 0$, there is a $\delta > 0$ such that if $\|[\mathbf{x}(0)^T, \mathbf{u}(0)^T]\| < \varepsilon$ then $\|[\mathbf{x}(k)^T, \mathbf{u}(k)^T]\| < \delta$ for all $k > 0$. It is asymptotically stable if $\|[\mathbf{x}(k)^T, \mathbf{u}(k)^T]\| \rightarrow 0$ as $k \rightarrow \infty$.

Lyapunov's Theorem: If there is a function $V(\mathbf{x}, \mathbf{u})$ which is positive-definite, namely it satisfies $V(\mathbf{x}, \mathbf{u}) \geq 0$ and $V(\mathbf{x}, \mathbf{u}) = 0$ only if $(\mathbf{x}, \mathbf{u}) = (\mathbf{0}, \mathbf{0})$, and satisfies the 'decrecent' property

$$\|[\mathbf{x}_1^T, \mathbf{u}_1^T]\| > \|[\mathbf{x}_2^T, \mathbf{u}_2^T]\| \Rightarrow V(\mathbf{x}_1, \mathbf{u}_1) > V(\mathbf{x}_2, \mathbf{u}_2) \quad (3.50)$$

and if, along any trajectory of the system $\mathbf{x}(k+1) = f(\mathbf{x}(k), \mathbf{u}(k))$ in some neighborhood of $(\mathbf{0}, \mathbf{0})$, the property

$$V(\mathbf{x}(k+1), \mathbf{u}(k+1)) \leq V(\mathbf{x}(k), \mathbf{u}(k)) \quad (3.51)$$

holds, then $(\mathbf{0}, \mathbf{0})$ is a stable equilibrium point. In addition, if $V(\mathbf{x}(k), \mathbf{u}(k)) \rightarrow 0$ as $k \rightarrow \infty$

then it is asymptotically stable. Such a function $V(\mathbf{x}, \mathbf{u})$ is called a Lyapunov function.

For the purpose of proving stability it is enough to consider the case when the states are to be driven to the origin, from some arbitrary initial conditions. We also assume a ‘regulator’ formulation of MPC; namely, we want to drive the system outputs to the origin.

Suppose that a predictive controller is obtained for the plant

$$\mathbf{x}(k+1) = f(\mathbf{x}(k), \mathbf{u}(k)) \quad (3.52)$$

by minimizing the cost function

$$J(k) = \sum_{i=1}^{H_p} \|\hat{\mathbf{y}}(k+i|k)\|_{\mathbf{Q}(i)}^2 + \sum_{i=1}^{H_u} \|\Delta \hat{\mathbf{u}}(k+i-1|k)\|_{\mathbf{R}(i)}^2 \quad (3.53)$$

We assume that $\mathbf{Q}(i) \geq 0$, and $\mathbf{R}(i) \geq 0$, so the cost function is $J(k)$ positive-definite, and satisfies the ‘decreasing’ property.

Now assume, as usual in proofs of nominal stability, that the plant model is perfect so that the predicted and real output trajectories coincide: $\mathbf{y}(k+i) = \hat{\mathbf{y}}(k+i|k)$ if $\mathbf{u}(k+i) = \hat{\mathbf{u}}(k+i|k)$. Let $J^*(k)$ be the optimal value of the cost function at time k , and $\Delta \mathbf{u}^*(k)$ and $\mathbf{y}^*(k)$ be the corresponding optimal control moves and optimal outputs. We have

$$J^*(k) = \sum_{i=1}^{H_p} \|\mathbf{y}^*(k+i)\|_{\mathbf{Q}(i)}^2 + \sum_{i=1}^{H_u} \|\Delta \mathbf{u}^*(k+i-1)\|_{\mathbf{R}(i)}^2 \quad (3.54)$$

From the definition of the control horizon H_u of MPC, we have

$$\Delta \mathbf{u}^*(k+i) = 0 \quad \forall i \geq H_u \quad (3.55)$$

The value of the cost function J evaluated at time $k+1$, while maintaining the same control inputs $\mathbf{u}^*(k)$ as computed at time k , would be

$$\begin{aligned}
J'(k+1) &= \sum_{i=1}^{H_p} \left\| \mathbf{y}^*(k+1+i) \right\|_{\mathbf{Q}(i)}^2 + \sum_{i=1}^{H_u} \left\| \Delta \mathbf{u}^*(k+1+i-1) \right\|_{\mathbf{R}(i)}^2 \\
&= \sum_{i=1}^{H_p} \left\| \mathbf{y}^*(k+i) \right\|_{\mathbf{Q}(i)}^2 + \sum_{i=1}^{H_u} \left\| \Delta \mathbf{u}^*(k+i-1) \right\|_{\mathbf{R}(i)}^2 - \left\| \mathbf{y}^*(k+1) \right\|_{\mathbf{Q}(1)}^2 - \left\| \Delta \mathbf{u}^*(k) \right\|_{\mathbf{R}(1)}^2 \\
&\quad + \left\| \mathbf{y}^*(k+H_p+1) \right\|_{\mathbf{Q}(H_p)}^2 + \left\| \Delta \mathbf{u}^*(k+H_u) \right\|_{\mathbf{R}(H_u)}^2 \\
&= J^*(k) - \left\| \mathbf{y}^*(k+1) \right\|_{\mathbf{Q}(1)}^2 - \left\| \Delta \mathbf{u}^*(k) \right\|_{\mathbf{R}(1)}^2 + \left\| \mathbf{y}^*(k+H_p+1) \right\|_{\mathbf{Q}(H_p)}^2
\end{aligned} \tag{3.56}$$

At time $k+1$ the new optimization problem, with initial condition $\mathbf{y}^*(k+1)$ is solved. The new optimal cost function is

$$\begin{aligned}
J^*(k+1) &\leq J'(k+1) \\
&= J^*(k) - \left\| \mathbf{y}^*(k+1) \right\|_{\mathbf{Q}(1)}^2 - \left\| \Delta \mathbf{u}^*(k) \right\|_{\mathbf{R}(1)}^2 \\
&\quad + \left\| \mathbf{y}^*(k+H_p+1) \right\|_{\mathbf{Q}(H_p)}^2 + \left\| \Delta \mathbf{u}^*(k+H_u) \right\|_{\mathbf{R}(H_u)}^2
\end{aligned} \tag{3.57}$$

Suppose that we add ‘terminal constraints’ to the MPC formulation, which force the plant outputs to reach the origin at the end of the prediction horizon. The terminal constraints are

$$\hat{\mathbf{y}}(k+H_p | k) = \mathbf{0} \tag{3.58}$$

We have assumed that the plant model is perfect and the equilibrium point is at the origin ($(\mathbf{x}, \mathbf{u}) = (\mathbf{0}, \mathbf{0})$, $\mathbf{0} = f(\mathbf{0}, \mathbf{0})$). If the optimization problem is feasible, so the terminal constraints force the plant outputs to reach the origin at time $k+H_p$, we have

$$\mathbf{y}^*(k+i | k) = \mathbf{0} \quad \forall i \geq H_p \tag{3.59}$$

Using equation (3.59), equation (3.57) becomes

$$\begin{aligned}
J^*(k+1) &\leq J'(k+1) \\
&= J^*(k) - \left\| \mathbf{y}^*(k+1) \right\|_{\mathbf{Q}(1)}^2 - \left\| \Delta \mathbf{u}^*(k) \right\|_{\mathbf{R}(1)}^2 + \left\| \mathbf{y}^*(k+H_p+1) \right\|_{\mathbf{Q}(H_p)}^2 \\
&= J^*(k) - \left\| \mathbf{y}^*(k+1) \right\|_{\mathbf{Q}(1)}^2 - \left\| \Delta \mathbf{u}^*(k) \right\|_{\mathbf{R}(1)}^2 \\
&< J^*(k)
\end{aligned} \tag{3.60}$$

So $J^*(k)$ is a Lyapunov function for the closed-loop system, and we conclude by Lyapunov’s

stability theorem that the closed-loop system is stable. With the assumption that the optimization problem has a solution at each step, we have shown that stability of MPC can be achieved by imposing terminal constraints. The problem with this approach is that the general constrained optimization problems can be difficult to solve, and just adding terminal constraints may not be feasible. From equation (3.60) we see that the condition for $J^*(k)$ to be a Lyapunov function is

$$\begin{aligned}
 & -\|\mathbf{y}^*(k+1)\|_{Q(1)}^2 - \|\Delta\mathbf{u}^*(k)\|_{R(1)}^2 + \|\mathbf{y}^*(k+H_p+1)\|_{Q(H_p)}^2 < 0 \\
 \text{or} \\
 & \|\mathbf{y}^*(k+1)\|_{Q(1)}^2 + \|\Delta\mathbf{u}^*(k)\|_{R(1)}^2 > \|\mathbf{y}^*(k+H_p+1)\|_{Q(H_p)}^2
 \end{aligned} \tag{3.61}$$

Equation (3.61) shows that imposing terminal constraints to obtain stability is unnecessarily restrictive. We can relax the terminal constraints $\hat{\mathbf{y}}(k+H_p | k) = \mathbf{0}$ to a terminal constraint set

$$\hat{\mathbf{y}}(k+H_p | k) < \lambda(k)\mathbf{y}_0 \tag{3.62}$$

which contains the origin rather than a single point, where \mathbf{y}_0 is a constant system output vector. The size of the terminal constraint set can be adjusted by a variable $\lambda(k)$. Based on the idea of terminal constraint set, a multi-stage MPC with guaranteed stability may be developed as follows:

- At each time step k , define a terminal constraint set $\hat{\mathbf{y}}(k+H_p | k) < \lambda(k)\mathbf{y}_0$.
- First stage: at time $k=0$, choose an initial terminal constraint set such that MPC can drive the plant into this set with $\lambda(0) = 1$.
- Gradually decrease $\lambda(k)$ at each time step, until MPC achieves guaranteed system stability for initial conditions within the set with $\lambda(k) \ll 1$, assuring that the system outputs approach the origin.

Figure 3.5 illustrates the basic idea behind multi-stage MPC for a single-input-single-output (SISO) system. At time step k , the terminal constraint $\hat{\mathbf{y}}(k+H_p | k)$ is within the terminal constraint set $\lambda(k)\mathbf{y}_0$ (the broken-line circle at time $k+H_p$). The size of the terminal constraint set is gradually decreased by adjusting the parameter λ at each time step so that

$\lambda \rightarrow 0$. When $\lambda = 0$, the terminal constraint set $\lambda \mathbf{y}_0$ becomes the terminal constraints $\hat{\mathbf{y}}(k + H_p | k) = \mathbf{0}$. The cost function $J^*(k)$ becomes a Lyapunov function for the closed-loop system, and by Lyapunov's stability theorem the closed-loop system is stable.

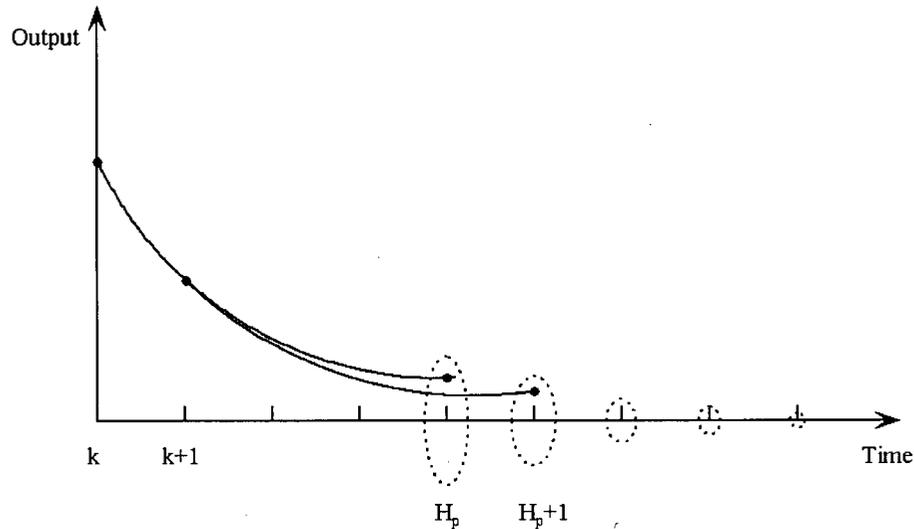


Figure 3.5 The basic idea of multi-stage MPC.

3.3 System Identification for Unknown Payload

In this section we will design the system identification module of the overall MPC scheme for the flexible-link robotic manipulator, as shown in Figure 3.1. A linear dynamic model of the flexible link system has been developed in Section 2.4. The corresponding linear state-space model of the system is

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} \end{aligned} \quad (3.63)$$

Equation (3.63) gives the model structure of the system. The parameters of the system matrices \mathbf{A} , \mathbf{B} and \mathbf{C} are dependent on the physical parameters of the flexible-link robotic manipulator, the mode shape functions, and the payload.

The physical parameters of the flexible-link robotic manipulator, for example the mass, length, and moment of inertia of each link, can be determined off line. From Chapter 2, we know that the effect of the payload on mode shapes is not significant for both one-link and the

two-link flexible robotic manipulators. Accordingly we will use the mode shapes corresponding to the nominal payload in the linear state-space model.

The payload of the system is usually not known and we need to identify it on line in order to increase the efficiency of system operation. Due to the nonlinearity of the system, the linear model of the system is valid only around the operating point. We will design the system identification procedure as follows. For system identification, move only the link that directly handles the payload and lock the other links at their initial positions for a multi-link system. Design an MPC controller that moves the link with payload at low speed from the initial position with a small angle ($<10^\circ$) and then move it back to the initial position. The identification module then estimates the unknown parameters of the matrices \mathbf{A} , \mathbf{B} and \mathbf{C} from the input and output data of the system. The payload mass and inertia can be calculated from the term of matrix \mathbf{A} or \mathbf{B} . The system model (3.63) is updated using the newly estimated payload value.

The identification module uses prediction-error methods (PEM) to find the parameters of the system model. Given a set of measurements from the system:

$$\mathbf{Z}^N = [\mathbf{y}(1), \mathbf{u}(1), \mathbf{y}(2), \mathbf{u}(2), \dots, \mathbf{y}(N), \mathbf{u}(N)] \quad (3.64)$$

the prediction-error vector is:

$$\boldsymbol{\varepsilon}(t, \mathbf{P}) = \mathbf{y}(t) - \hat{\mathbf{y}}(t | t-1, \mathbf{P}) \quad (3.65)$$

where \mathbf{P} is the parameter vector. In our case, it is the vector of unknown payload mass and inertia $\mathbf{P} = [m_p \quad J_p]^T$. Also $\hat{\mathbf{y}}(t | t-1, \mathbf{P})$ are the predicted system outputs based on the past data set \mathbf{Z}^{t-1} . When the data set \mathbf{Z}^N is known, the prediction-errors can be computed for $t=1, 2, \dots, N$. The estimated parameter vector $\hat{\mathbf{P}}$ is then defined by minimizing a quadratic prediction error criterion $V_N(\mathbf{P})$:

$$V_N(\mathbf{P}) = \sum_{t=1}^N \|\boldsymbol{\varepsilon}(t, \mathbf{P})\|^2 \quad (3.66)$$

$$\hat{\mathbf{P}}_N = \arg \min V_N(\mathbf{P}, \mathbf{Z}^N)$$

where 'arg min' denotes the minimizing argument; i.e., the value of \mathbf{P} that minimizes $V_N(\mathbf{P})$.

At time N , when the data set \mathbf{Z}^N is known, the function $V_N(\mathbf{P})$ is just a finite-dimensional real parameter vector \mathbf{P} . But in general, the function $V_N(\mathbf{P})$ cannot be minimized by analytical methods. The numerical solution then has to be found by an iterative search algorithm.

Computation of the estimated parameters can be done using the Matlab System Identification Toolbox (Mathworks, 2003). The function 'idss' is used to construct the state-space model structures with various parameterizations. To define the model structure, the so called *structure matrices* are used. These are 'shadow matrices' to \mathbf{A} , \mathbf{B} and \mathbf{C} , and have the same size and coincide with them at all known matrix elements. The structure matrices are denoted by \mathbf{As} , \mathbf{Bs} and \mathbf{Cs} , and have the entry *NaN* at those elements that correspond to unknown parameters to be estimated. The unknown parameters are determined by using the function 'pem'. It uses an iterative search algorithm to minimize a prediction error criterion.

3.4 Fuzzy Rule-Based Auto-Tuning

In this section we will design the fuzzy rule-based auto-tuning module of the overall IMPC scheme for a flexible-link robotic manipulator, as shown in Figure 3.1. This knowledge-based fuzzy tuner is developed for auto-tuning of the MPC and for adjusting the model structure of the real-time system identifier. In this context we will rely on the step response of the flexible-link robotic manipulator. The setpoint is a constant value for each link. The desired closed-loop response is to move the manipulator from the initial position to the final position as quickly as possible, without generating undesirable link vibrations at the end of the movement while satisfying the input and output constraints. The ideal goal for the fuzzy tuner is to adjust the tuning parameters of MPC controller to achieve an overdamped system that is as close as possible to the critical-damped system, while not violating the constraints of the system. This will make the closed-loop response of the system as fast as possible without causing an overshoot. Following steps are taken in order to design the fuzzy tuner:

- Identify the adjustable parameters in predictive control. Examine the effects of these parameters on the controller performance.
- Design the membership functions for the input and output variables of the fuzzy inference system (FIS).

- Develop a set of fuzzy rules based on the knowledge that has been gained by studying the problem. This is the knowledge base used by the fuzzy tuner. This knowledge base can be updated and changed based on the experience gained from experiments and simulations of the system.
- Develop a fuzzy inference system (FIS) that can easily map the input space to the output space. This FIS will be used for tuning the MPC controller.

The first step develops a knowledge base for tuning the MPC controller. In the second step fuzzy logic is used as a tool to transform this knowledge base into an intelligent decision making system that can automatically tune the parameters of the MPC controller in order to achieve the desired closed-loop system performance.

3.4.1 Effects of the Tuning Parameters

To date only a few guidelines are known related to tuning the parameters of an MPC. In fact, conventionally, the tuning parameters are determined using the experience gained from simulations and tests of the system. Possible tuning parameters for the present control scheme are: the number of modes which the internal model needs to accurately predict the system response, control horizon H_u , prediction horizon H_p , tracking error weighting matrix $\mathbf{Q}(i)$, control penalty weighting matrix $\mathbf{R}(i)$, and the parameters of the disturbance model and the observer dynamics. In the present work, we examine the effects of these parameters on the controller performance, and obtain insight into appropriate parameters for on-line tuning. Theoretical analysis, experiments, and simulations are used to investigate this problem. One goal here is to develop a systematic method for adjusting the parameters. The definitions of control horizon, prediction horizon, and weighting parameters are intuitive. Consequently, fuzzy logic is suitable in the design of the knowledge base of controller tuning. Recall that the cost function of the MPC controller for a flexible-link robotic manipulator is:

$$J = \sum_{i=1}^{H_p} \left\| \hat{\mathbf{y}}(k+i|k) - \mathbf{r}(k+i) \right\|_{\mathbf{Q}(i)}^2 + \sum_{i=1}^{H_u} \left\| \Delta \hat{\mathbf{u}}(k+i-1|k) \right\|_{\mathbf{R}(i)}^2 \quad (3.67)$$

This is the cost function that has to be minimized at each time step by selecting a suitable control move $\Delta \hat{\mathbf{u}}(k+i-1|k)$. The control horizon H_u is the number of future time steps after

which the control inputs will be held constant; that is, all future input moves are set to zero. Increasing H_u will result in a more active controller, and will generally increase the magnitude and rate of change of the control input signals. To a certain extent this increased activity will improve the controller performance, but too large a value for H_u may result in excessive ringing of the control signals and significant high frequency components in the step response transients. This horizon also has a significant effect on the computational load of the algorithm. It has been recommended that H_u should be greater than or equal to the number of unstable or poorly damped zeros in the open loop system (Clark, 1987). In our case, we will use a relatively small control horizon; e.g., $5 \rightarrow 15$. We will pick a suitable H_u such that the controller performance is insensitive to small adjustments in this horizon based on simulations and experiments of the system. The H_u will be fixed to this value, and will not be tuned by the fuzzy tuner. This will reduce the number of tunable parameters for the fuzzy tuner, simplify the tuning process, and increase the speed of the fuzzy decision making system.

The prediction horizon H_p is related to the stability and the speed of response of the system. It specifies the number of future time steps for which the predicted system outputs will be calculated and included in the control law. From the cost function in equation (3.67), we can see that a cost function with larger H_p will include greater number of future prediction points. The resulting MPC controller is long-sighted. Given a sufficiently long prediction horizon, the controller can 'see' a potential constraint and can take corrective action immediately to avoid it, or at least minimize its adverse effects. A flexible-link manipulator is a non-minimum phase plant. It has a short-term response in one direction, but a longer-term response in the opposite direction. In this case, the prediction horizon has to be sufficiently long, so the optimization should focus primarily on the longer-term behavior. Otherwise, the controller would move in the wrong direction. Also, H_p should be sufficiently large to include the negatively moving part of a non-minimum phase system. Too short a prediction horizon can cause the controller to work with insufficient information and the control is likely to be poor or even unstable. In general, increasing H_p can reduce the overshoot and increase stability for open-loop stable system. On the other hand, increasing H_p will reduce the speed response of the system, and will also increase the computational

load of the algorithm. It is recommended that H_p be set comparable with the rise time of the open loop system (Clark, 1987). The fuzzy tuner will adjust the H_p based on the closed-loop response of the system. If the closed-loop response has a large overshoot, the fuzzy tuner will increase the H_p to reduce that overshoot. If the closed-loop speed of the response is low, the fuzzy tuner will decrease the H_p to increase the speed of response.

The primary control objective is to force the plants to their setpoints. The controller predicts how much each output will deviate from its setpoints within the prediction horizon. It multiplies each derivation by the output weight. The cost function that is minimized by the controller includes the weighted sum of squared derivations as shown in equation (3.67). The output weighting matrix $\mathbf{Q}(i)$ at time step i is a diagonal matrix. We will use a constant output weighting matrix at each time step, that is

$$\mathbf{Q} = \mathbf{Q}(i) \quad \forall i = 1, \dots, H_p$$

$$\mathbf{Q} = \begin{bmatrix} Q_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & Q_2 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & Q_{n_y} \end{bmatrix} \quad (3.68)$$

where n_y is the number of outputs, and Q_1, \dots, Q_{n_y} are the constant output weights. The weights must not be negative. Different outputs can have different output weights. If $Q_a \gg Q_b$ the controller does its best to track the setpoint r_a , sacrificing r_b tracking if necessary. On the other hand if $Q_a = 0$, the controller completely ignores the derivation output y_a from its setpoint r_a . In general, increasing Q_a can reduce the steady state error and the overshoot or undershoot of the output y_a . But increasing Q_a will also increase the control input u_a , and increase the derivations of the other outputs from their setpoints. For the flexible-link robotic manipulator system, the outputs are the joint angle and the deflection of each link. Choosing the output weights is a critical step. We need to tune the controller, varying the weights to achieve the desired behavior. The fuzzy tuner has to adjust the weights of these outputs in order to reduce the overshoots or undershoots of the joint angle and keep the link deflections within acceptable ranges, and the input torques of each joint within the corresponding input

constraints.

If the controller only focuses on setpoint tracking, it might choose to make larger manipulated-variable adjustments. These could lead to violation of the input constraints and make it impossible to achieve perfect setpoint tracking. These may also accelerate equipment wear or lead to control system instability. Thus one simultaneous objective of MPC is to minimize the weighted sum of controller adjustments as given in equation (3.67). The control penalty weight matrix $\mathbf{R}(i)$ at time step i is a diagonal matrix. We will use a constant control penalty weight matrix at each time step; that is

$$\mathbf{R} = \mathbf{R}(i) \quad \forall i = 1, \dots, H_u$$

$$\mathbf{R} = \begin{bmatrix} R_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & R_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & R_{n_u} \end{bmatrix} \quad (3.69)$$

where n_u is the number of inputs, and R_1, \dots, R_{n_u} are the constant input rate weights. The weights must not take negative values. Increasing the input rate weights has the effect of reducing the control activity. So they are also called move suppression factors for this reason. In general, increasing these weights will force the controller to make smaller and more cautious adjustments. This can bring the control inputs within their constraints. The controller will be less sensitive to prediction errors (i.e., more robust). Increasing these weights indefinitely will reduce the control activity to zero, which will ‘turn off’ the feedback action. Thus with a stable plant, one can expect to obtain a stable closed loop system by sufficiently increasing the input rate weights. But the setpoint tracking of the controller will degrade and the speed of response will decrease, since only small control actions will result. With an unstable plant one can expect an unstable feedback loop, if the weights are increased excessively. For a flexible-link robotic manipulator system, increasing the input rate weights for each link will reduce the input torques. This will reduce the link deflections, but will increase the overshoot of the joint angles. The fuzzy tuner will adjust these weights based on the closed-loop response of the system in order to bring the control inputs and the link deflections within their constraints and reduce the system overshoots.

The performance of the MPC controller is also dependent on the accuracy of the output

prediction $\hat{y}(k+i|k)$ as given in equation (3.67). If the prediction is good, minimizing the cost function will reduce the setpoint tracking error and force the plant to its setpoints. But if the prediction outputs are far from the real plant outputs, the controller will take wrong control moves. This will degrade the system performance, and the closed-loop may become unstable.

In the current application, output predictions of the flexible-link robotic manipulator depend on the internal model and the observer dynamics of the MPC controller. The accuracy of the internal model depends on the model structure and model parameters.

One of the parameters that relates to the internal model structure is the number of flexible modes which must be included in the model. Increasing the number of flexible modes can improve the prediction accuracy, but this will also increase the complexity of the model, make the controller design more difficult, and increase the computational load of the control algorithm. From Table 2.2 we can see that the frequencies for the third mode or higher are much greater than the frequencies of the first two modes for both links. Thus initially we will use only one mode for the internal model. If satisfactory performance cannot be achieved by a single mode model structure, we will include the second mode as well in the internal model. In order to prevent the excitation of the higher modes, we need to modify the input and output constraints. This will reduce the maximum absolute control inputs and the maximum absolute link deflections.

Since the flexible-link manipulator is a nonlinear system, the linear internal model is only valid around its operating point. If there are large setpoint changes, we need to update the internal model. A re-linearized plant model at the new operating point can improve the output predictions of the controller. The fixed interval of re-linearization will be decided by the designer and will not be tuned by the fuzzy tuner in the current application.

At the beginning of each sampling instant the controller estimates the current plant state. Accurate knowledge of the state improves the prediction accuracy, which in turn improves the controller performance. It follows that the observer dynamics represent another adjustable factor of the MPC controller. Choosing deadbeat dynamics gives a fast response to disturbances. But this fast-dynamics observer will also react relatively strongly to high-frequency vibrations in the measured output signals. If there is excessive measurement noise in these signals, it may be better to have slower observer dynamics, in order to produce some

low-pass filtering of the noise. The optimal trade-off for state estimation can be made by using Kalman filter theory to design the observer as we have shown in Section 3.2.3.

In summary, there are many adjustable parameters and factors in an MPC controller which can be tuned to improve the closed-loop system performance. In order to reduce the complexity of the fuzzy tuner design, only a few parameters with large adjustable ranges will be chosen and tuned by the fuzzy tuner. The other parameters will be chosen by the designer, and their values will be fixed for online application of the controller. Table 3.1 gives the adjustable MPC parameters. Those parameters that will be tuned by the fuzzy tuner are also identified in the table.

Table 3.1 Adjustable controller parameters.

Controller parameters	Control horizon H_u	Prediction horizon H_p	Output weight Q	Input rate weight R	Internal model	Observer dynamics
Tuned by fuzzy tuner	No	Yes	Yes	Yes	No	No

3.4.2 Fuzzy Inference System

In this section we will design a fuzzy inference system (FIS) to tune the MPC controller of a flexible link robotic manipulator system. Fuzzy logic is a convenient way to map an input space to an output space. The fuzzy inference system consists of a knowledge base in the form of a set of rules and an inference mechanism. The tuning actions are generated by applying the existing system performance data to the knowledge base and by using the inference mechanism. The knowledge base and the inference mechanism can handle imprecise and incomplete information, and the knowledge itself will improve and evolve through 'learning' and past experience.

Input and Output Variables

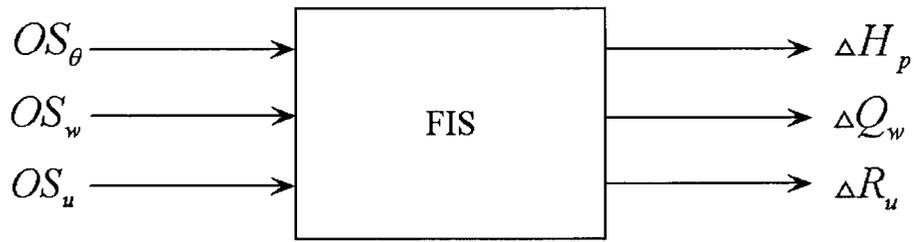


Figure 3.6 The fuzzy inference system.

Figure 3.6 shows a schematic diagram of the fuzzy inference system for the flexible link robotic manipulator system. The input variables are: OS_θ , OS_w , and OS_u . The output variables are: ΔH_p , ΔQ_w , ΔR_u .

OS_θ is defined as:

$$OS_\theta = |\theta_{\max}| - |r_\theta| \quad (3.70)$$

where θ_{\max} is the absolute maximum joint angle of the closed-loop system response, and r_θ is the absolute desired joint angle. We assume that the initial joint angle for each link is zero. The desired OS_θ value is $OS_\theta < 0$ and $|OS_\theta| \approx 0$. This means we want the step response of the system to be an overdamped system that is as close as possible to the critical damped system.

OS_w is defined as:

$$OS_w = |w_{\max}| - r_w \quad (3.71)$$

where w_{\max} is the absolute maximum link deflection of the closed-loop system response of the link and r_w is the link deflection constraint (maximum allowed link deflection). The desired OS_w value is $OS_w \leq 0$. This will make sure that the link deflection stays within its bound.

OS_u is defined as:

$$OS_u = |u_{\max}| - r_u \quad (3.72)$$

where u_{\max} is the absolute maximum control input of the closed-loop system response of the link and r_u is the control input constraint (maximum allowed control input). The desired OS_u value is $OS_u \leq 0$. This will guarantee that the control input stays within its bound.

ΔH_p is defined as:

$$\Delta H_p = (H_{p_new} - H_{p_old}) H_{p_sen} \quad (3.73)$$

where H_{p_old} is the prediction horizon before tuning, H_{p_new} is the prediction horizon after tuning, and H_{p_sen} is a sensitivity parameter. H_{p_sen} is introduced for adjusting the sensitivity of tuning.

ΔQ_w is defined as:

$$\Delta Q_w = (Q_{w_new} - Q_{w_old}) Q_{w_sen} \quad (3.74)$$

where Q_{w_old} is the link deflection weight before the tuning, Q_{w_new} is the link deflection weight after the tuning, and Q_{w_sen} is the sensitivity parameter. We normalize the joint angle weight as $Q_y = 1$. The overall output weights are

$$Q_{yw} [1 \quad Q_w] \quad (3.75)$$

where Q_{yw} is a predetermined constant value.

ΔR_u is defined as:

$$\Delta R_u = (R_{u_new} - R_{u_old}) R_{u_sen} \quad (3.76)$$

where R_{u_old} is the input rate weight before tuning, R_{u_new} is the input rate weight after the tuning, and R_{u_sen} is the sensitivity parameter.

Fuzzy Membership Functions

Next we will design the fuzzy membership functions for the input and output variables. Fuzzy logic starts with the concept of a fuzzy set. A fuzzy set is a set without a precise and clearly defined boundary, and it can describe non-precise or qualitative concepts. It may contain

elements with only a partial degree of membership. A membership function is a curve that defines how each point in the input space is mapped to a membership value (or degree or grade of membership) between 0 and 1. The membership function associated with a given fuzzy set maps an input value to its appropriate membership value. The main condition a membership function must satisfy is that its values must lie between 0 and 1. The function itself can be an arbitrary curve. We will use two types of membership functions: 1. functions formed by straight line segments, 2. Gaussian functions. The first type of function has the advantage of simplicity. The Gaussian membership function obeys the Gaussian distribution curve whose main advantage is its smoothness. Figure 3.7 and Figure 3.8 show the membership functions for the input and output variables. These membership functions depend on common knowledge, experience, judgment, and so on. The numerical values are used to define the memberships only in a relative sense. An appropriate physical meaning should be attached to each value. These numerical values will depend on the hardware of the system. They must be scaled and converted to achieve physical and dimensional compatibility for the particular practical application. The ranges of the membership function are normalized as $[0 \ 10]$. The notation used in the figures showing membership functions are given below:

VS = Very small value

S = Small value

M = Moderate value

L = Large value

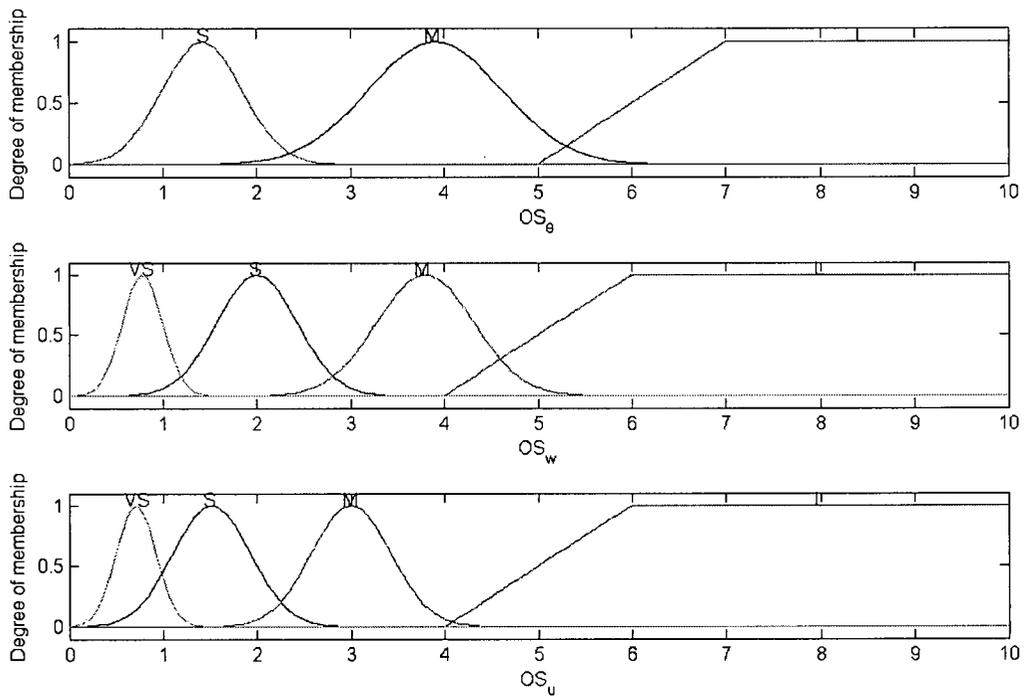


Figure 3.7 Membership functions for the input variables.

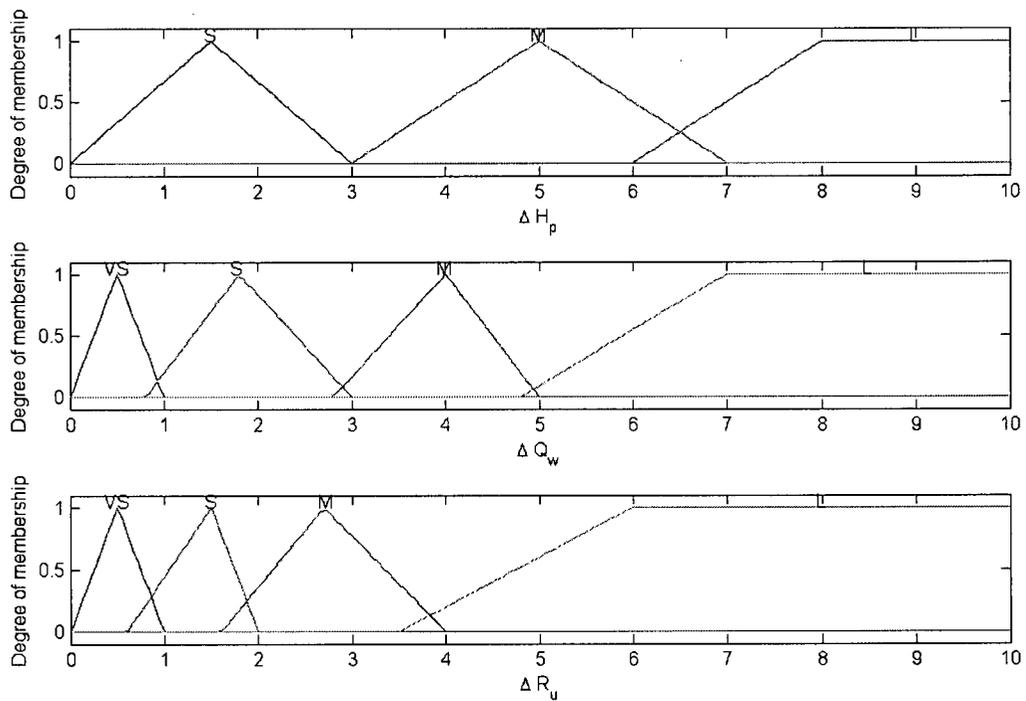


Figure 3.8 Membership functions for the output variables.

Fuzzy Knowledge Base

The fuzzy rules for tuning the controller are developed next. The required expert knowledge for tuning an MPC has been established in Section 3.4.1. Direct knowledge on tuning the controller can also be obtained through computer simulations and experiments of the system. In Chapter 4, we will investigate through simulations the effects of the tuning parameters on the system performance. A basic reasoning for adjusting the MPC control parameters can be expressed linguistically as given in Table 3.2.

Table 3.2 General linguistic rules for tuning an MPC controller.

<i>If the joint angle response has no overshoot and the link deflection and control input are within their constraints, then do not change the control parameters.</i>
<i>When the control input is within constraint, if the joint angle response has a large overshoot and the link deflection constraint violation level is not small, or if the joint angle overshoot is moderate, and the link deflection constraint violation level is large, then increase the prediction horizon by a large value.</i>
<i>When the control input is within constraint, if the joint angle overshoot is moderate, and the link deflection constraint violation level is moderate, then increase the prediction horizon by a moderate value.</i>
<i>When the control input is within constraint, if the joint angle overshoot and the link deflection constraint violation value are not zero, and at least one of them is small, then slightly increase the prediction horizon.</i>
<i>When the joint angle response has no overshoot and the link deflection is within constraint, if control input constraint violation level is large, then slightly increase the prediction horizon and moderately increase the input rate weight, or if control input constraint violation level is not large, then slightly increase the prediction horizon and the input rate weight.</i>
<i>When the joint angle response has an overshoot and the link deflection is within constraint, and control input constraint violation level is very small, then slightly increase the prediction horizon.</i>
<i>When the joint angle response has no overshoot, if the link deflection constraint violation level is moderate or large and the control input constraint violation level is large, then moderately increase the link deflection weight.</i>
<i>When the joint angle response has no overshoot, if the link deflection constraint violation level is not very small and the control input constraint violation level is large, then slightly increase the link deflection weight.</i>

When the joint angle response has no overshoot, if the link deflection constraint violation level is not large and the control input constraint violation level is not large, then increase the link deflection weight by a very small value.

These linguistic rules reflect the actions of a human expert in tuning an MPC controller by observing the response of the closed-loop system. More rules can be added, and also these rules can be modified to improve the tuning accuracy.

Fuzzy Inference Mechanism

Fuzzy inference is the process of mapping the input space to the output space using fuzzy logic. The mapping provides a basis from which decisions can be made.

Figure 3.9 shows the block diagram of the fuzzy inference process which will be used to tune the MPC controller. There are five parts in the fuzzy inference process: fuzzification of the input variables, application of the fuzzy operator (AND or OR) in the antecedent of the fuzzy rules, implication from the antecedent to the consequent, aggregation of the consequents across the rules, and defuzzification. Next we will discuss each step in more detail.

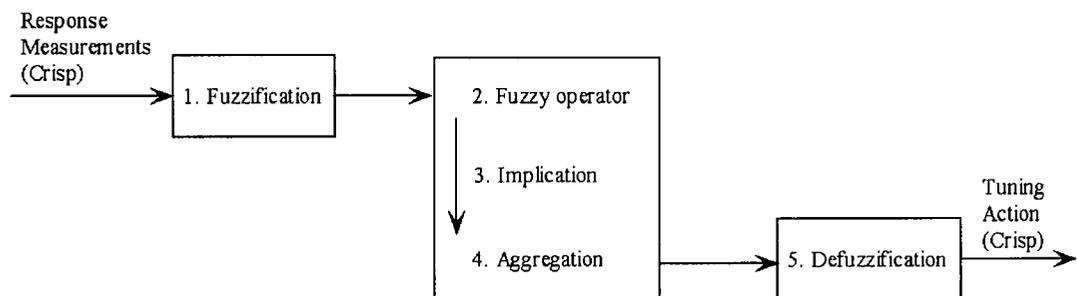


Figure 3.9 Fuzzy inference process.

Step 1. Fuzzification

The inputs to the FIS are the measured response of the closed-loop system which is controlled by MPC with the initial set of tuning parameters. These are crisp numerical values limited to the universe of discourse of the input variables (in the current application within the interval between 0 and 10). The first step of the fuzzy inference process is to generate membership values for a fuzzy variable using membership functions.

Step 2. Apply Fuzzy Operator

The fuzzy rule base is a set of if-then rules. A rule in a fuzzy knowledge base is generally a relation of the form:

$$\text{IF } x_i \text{ is } A_i \text{ AND (OR) } y_i \text{ is } B_i \text{ THEN } C_i \quad (3.77)$$

where A_i , B_i , and C_i are fuzzy sets. The 'if-part' of a fuzzy rule is called the antecedent, and the 'then-part' is called the consequent. After the inputs are fuzzified and their membership values are generated in step one, the next step is to present these membership values to the fuzzy operator. If there are multiple parts to the antecedent, we need to apply fuzzy logic operators and resolve the antecedent to a single number between 0 and 1. This is the degree of support for the rule. The fuzzy operators are defined as follow:

$$\begin{aligned} A_i \text{ AND } B_i &== \min(A_i, B_i) \\ A_i \text{ OR } B_i &== \max(A_i, B_i) \end{aligned} \quad (3.78)$$

where 'AND' corresponds to a 'minimum' operation, and 'OR' corresponds to a 'maximum' operation.

Step 3. Apply Implication Method

Interpretation of an if-then fuzzy rule involves: evaluating the antecedent (which requires fuzzification of the input and application of the necessary fuzzy operators), and applying that result to the consequent (known as implication). A consequent is a fuzzy set represented by a membership function (C_i in (3.77)). The input to the implication process is a single number given by the antecedent, and the output is a fuzzy set. The implication process modifies the fuzzy set C_i to the degree specified by the antecedent. The most common method for modifying the output fuzzy set is truncation using the 'minimum operation.' Implication is implemented for each rule in the fuzzy rule base.

Step 4. Aggregate All Outputs

Since the decisions are based on the testing of all the rules in the fuzzy rule base, the rules must be combined in some way in order to make a decision. Aggregation is the process by

which the fuzzy sets that represent the outputs of each rule are combined into a single fuzzy set. The ‘maximum operation’ will be used to aggregate the output fuzzy sets for each rule into a single output fuzzy set.

Step 5. Defuzzification

In the last step, the resulting fuzzy set from the aggregation process is defuzzified. The output is a crisp single number. The centroid method is used for the defuzzification process. The centroid method returns the center of area under the membership function curve.

3.4.3 Overall Scheme of the Fuzzy Tuner

We may consider the fuzzy tuner as a nonlinear function that maps the input variables to the output variables. The inputs to the fuzzy tuner are the measured closed-loop system response, and the outputs are the tuning action for adjusting the MPC controller parameters.

In order to simplify the tuning process, we set the MPC parameters to some relatively small values. In the fuzzy tuner these parameters are gradually increased based on the measured system responses until the desired system performance is achieved. In this thesis, we mainly focus on tuning an MPC controller with respect to the step response of the system. The goal is to adjust the MPC controller parameters such that the closed-loop response of the system reaches the following characteristics: the joint angle step response for each link is underdamped and is as close as possible to the critical damped system, and the link deflection and control input for each link are within their design limits. In this way, we will obtain a closed-loop system that has a fast response with no overshoot and all the constraints satisfied.

Overall scheme of the fuzzy tuner for a flexible-link robotic manipulator is shown in Figure 3.10. First, we input the setpoints for each link, initial values of the controller parameters (H_p , H_u , \mathbf{Q} , \mathbf{R}), and the input and output constraints of the system. The setpoints are the desired final joint angles of the manipulator, and we assume that the initial joint angle of each link is zero. The prediction horizon H_p is a tuning parameter. We will start with a relatively small H_p . The closed-loop system will produce an underdamped step response with large joint angle overshoots. The fuzzy tuner will increase H_p until the system has no joint angle overshoot. We will use a relatively small control horizon; e.g., $5 \rightarrow 15$. Note that \mathbf{Q} is

the vector that includes the output weights for all the links. For each link the output weight is $Q_{yw} [1 \ Q_w]$ and the joint angle weight is normalized to 1. Here Q_{yw} is a predetermined constant value, and is the overall weight for joint angle and link deflection. Furthermore Q_w is a tuning parameter. A large value of Q_w will reduce the link deflection, but the joint angle tracking error will be increased. The parameter Q_w will start with a relatively small initial value and will be increased by the fuzzy tuner. The vector that includes the input rate weights for all the links is \mathbf{R} . For each link, the input weight R_u is a tuning parameter, which starts with a relatively small initial value and is increased by the fuzzy tuner.

After the initialization step, the MPC controller with the initial parameter values will control the flexible-link robotic manipulator system. The measured system outputs are sent to a signal preprocessor. This signal preprocessor generates the values of OS_θ , OS_w , and OS_u for each link, based on the system outputs. The ranges of OS_θ , OS_w , and OS_u are normalized in order to match the ranges of their membership functions. We set the range for each variable from 0 to 10. The preprocessor first calculates the values of the variables by using the following equations:

$$\begin{aligned} OS_\theta &= |\theta_{\max}| - |r_\theta| \\ OS_w &= |w_{\max}| - r_w \\ OS_u &= |u_{\max}| - r_u \end{aligned} \quad (3.79)$$

The preprocessor then normalizes these values in the range [0 10] using the following relationships:

$$\begin{aligned} \text{If } OS_\theta > 10 \text{ then set } OS_\theta &= 10 \\ \text{If } OS_\theta < \varepsilon_\theta \text{ then set } OS_\theta &= 0 \\ \text{If } OS_w > 10 \text{ then set } OS_w &= 10 \\ \text{If } OS_w < 0 \text{ then set } OS_w &= 0 \\ \text{If } OS_u > 10 \text{ then set } OS_u &= 10 \\ \text{If } OS_u < 0 \text{ then set } OS_u &= 0 \end{aligned} \quad (3.80)$$

where ε_θ is the absolute tolerance of joint angle, $OS_w < 0$ means the link deflection is within its limit, and $OS_u < 0$ means the control input is within its limit.

If the following conditions are satisfied, then the tuning process will stop, and we will have a closed-loop system with fast response, no joint angle shoots, and all the constraints satisfied:

$$OS_{\theta} = 0 \quad OS_w = 0 \quad OS_u = 0 \quad (3.81)$$

If the above termination conditions are not satisfied, OS_{θ} , OS_w , and OS_u will be sent to the fuzzy inference system. The fuzzy inference system will decide the tuning actions based on its knowledge base, and the tuning parameters will be updated using the following equations:

$$\begin{aligned} H_{p_new} &= H_{p_old} + \max(\Delta H_{p_1}, \dots, \Delta H_{p_n}) / H_{p_sen} \\ Q_w_{new} &= Q_w_{old} + \Delta Q_w / Q_w_{sen} \\ R_u_{new} &= R_u_{old} + \Delta R_u / R_u_{sen} \end{aligned} \quad (3.82)$$

where the subscript 'new' denotes the updated value, and 'old' denotes the previous value. The sensitivity parameters H_{p_sen} , Q_w_{sen} , and R_u_{sen} are introduced for adjusting the sensitivity of tuning. They can also be used to scale the tuning actions to proper numerical values. For a multi-link robotic manipulator, the tuning process is carried out on each link separately at the same time, and the maximum value of the overall tuning action for the prediction horizon ($\max(\Delta H_{p_1}, \dots, \Delta H_{p_n})$) will be used for updating the prediction horizon.

In the next step, the MPC controller will use the updated parameters to test the system again. The entire tuning process will be repeated until the desired closed-loop performance is achieved.

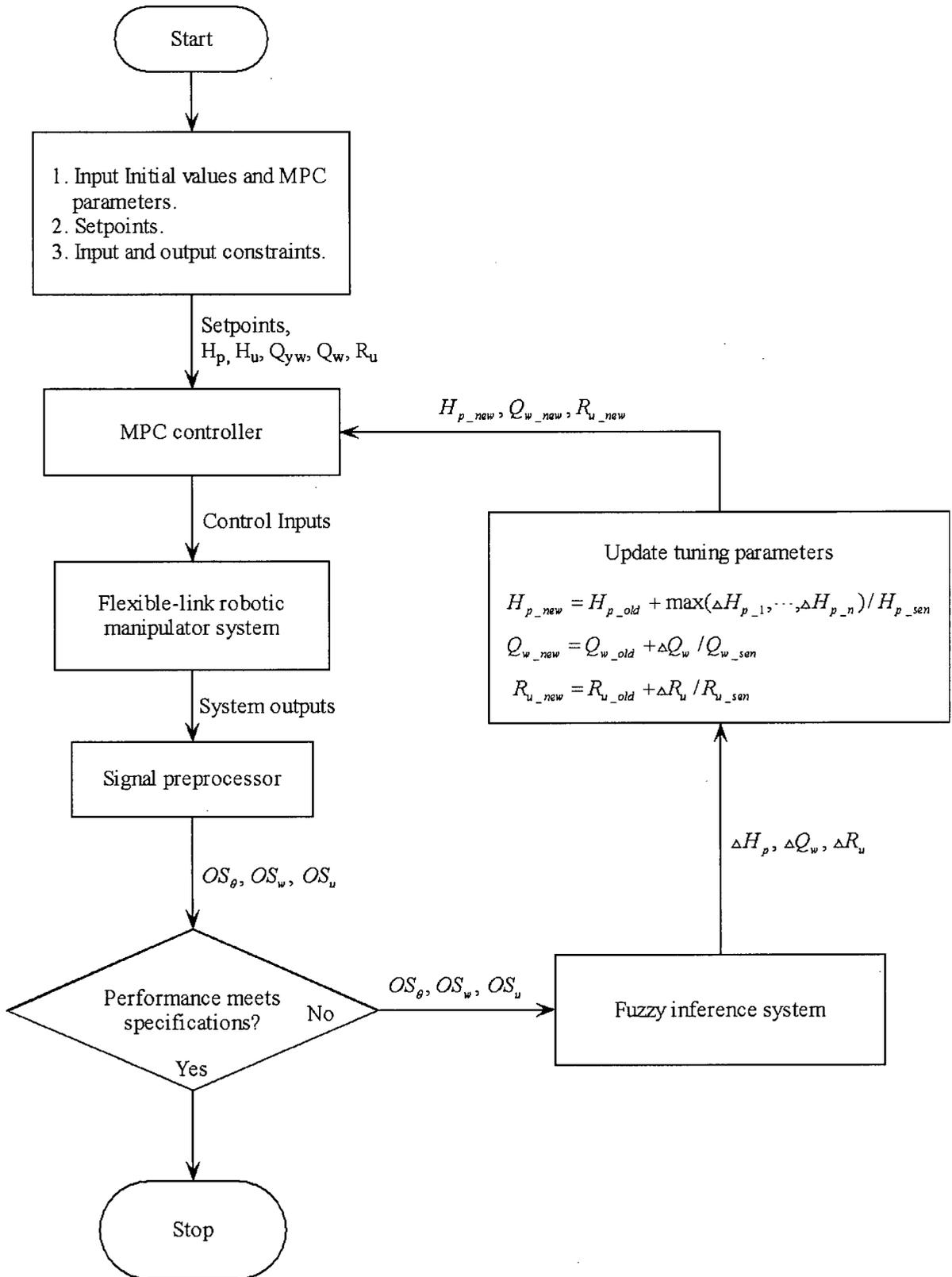


Figure 3.10 Overall scheme of the fuzzy tuner.

3.5 Summary

In this chapter a knowledge-based intelligent model based predictive control strategy was developed for the motion control of a flexible-link robot manipulator. The new control scheme, called intelligent model predictive control (IMPC), combined the advantages of conventional model predictive control (MPC) and knowledge-based soft control.

Overall structure of the IMPC algorithm and its underlying strategy were presented. The following main components of the IMPC system were designed: a computationally efficient MPC module with guaranteed nominal stability, a system identification module for handling systems with unknown payload, and a fuzzy rule-based intelligent auto-tuning module for adjusting the MPC controller parameters.

Chapter 4

Control of a Single-link Flexible Manipulator

In this chapter, the intelligent model predictive control (IMPC) scheme developed in Chapter 3 is applied for motion control of a single-link flexible robotic manipulator. The link of the manipulator is a 1 m long aluminum thin beam and its physical parameters are given in Table 2.1. This link is identical to the link 2 (outboard link) of the two-link flexible manipulator described in Chapter 2. A dynamic model for the system is developed in Section 4.1. Model predictive control (MPC) approach for the system is designed in Section 4.2. A system identification module was designed to identify the unknown system payload in Section 4.3. The knowledge for tuning the MPC controller is established in Section 4.4, and using the resulting fuzzy rule base a fuzzy tuner is designed. In Section 4.5 the performance of the IMPC scheme for the single-link system is evaluated using computer simulations.

4.1 Dynamic Model

We have developed the dynamic equations for a single-link flexible robotic manipulator in Section 2.32 using the assumed mode method. The dynamic equations are given in equations (2.92) and (2.93). These equations are nonlinear and include a finite number of flexible modes. In this section we develop a model for this system that is suitable for controller design and simulation.

4.1.1 Flexible Modes

Using computer simulations we investigate the number of flexible modes that are needed in order to adequately represent the elastic behavior of the single-link flexible robotic manipulator. From Table 2.3 we notice that the natural frequencies of the first four modes of the system with a nominal payload are:

$$f_1 = 1.1442 \text{ Hz}, f_2 = 11.1224 \text{ Hz}, f_3 = 31.1435 \text{ Hz}, \text{ and } f_4 = 56.1767 \text{ Hz}$$

The natural frequency of the third mode is much higher than that of the first two modes. In view of this, we now investigate the behavior of a model with just the first mode in comparison to a model with the first two modes. In the following simulations, we assume the system carries a nominal payload corresponding to $m_p = 0.38$ kg and $J_p = 0.001$ kg.m².

Case 1:

In this case there is no control input to the system, and the system starts with the following initial position:

$$\theta_1(0) = 0, w_{ip}(0) = 51.2 \text{ mm} \quad (4.1)$$

The initial joint angle is zero and the initial tip deflection is 51.2 mm. The simulation results are shown in Figure 4.1 and Figure 4.2. From the simulation results we see that there is very little difference between the responses of the single-mode model and the two-mode model, in this case.

Case 2:

In this case the system starts with the following initial position:

$$\theta_1(0) = 0, w_{ip}(0) = 0 \quad (4.2)$$

Both the initial joint angle and the initial tip deflection are equal to zero in this case. A symmetric bang-bang input torque of magnitude of 6 N.m is given to the system as shown in Figure 4.5. The system responses under this input torque are shown in Figure 4.3 and Figure 4.4. In this case as well, the system responses of the single-mode model and the two-mode model do not deviate significantly.

In summary, for the single-link flexible manipulator, the simulation results suggest that the fundamental mode alone is able to capture the flexible character of the system response quite accurately. Accordingly, we will use the model with just the first flexible mode for the design of MPC controller. The nonlinear model of the plant is represented using the first two flexible modes, however. Furthermore, in order to prevent the higher modes being excited by the MPC controller, we will impose constraints for the system input and output, in the MPC controller design, as follows:

$$\begin{aligned} |u_1| &\leq 6 \text{ N.m} \\ |w_{tip}| &\leq 25 \text{ mm} \end{aligned} \tag{4.3}$$

where u is the joint input torque and w_{tip} is the deflection of the link tip.

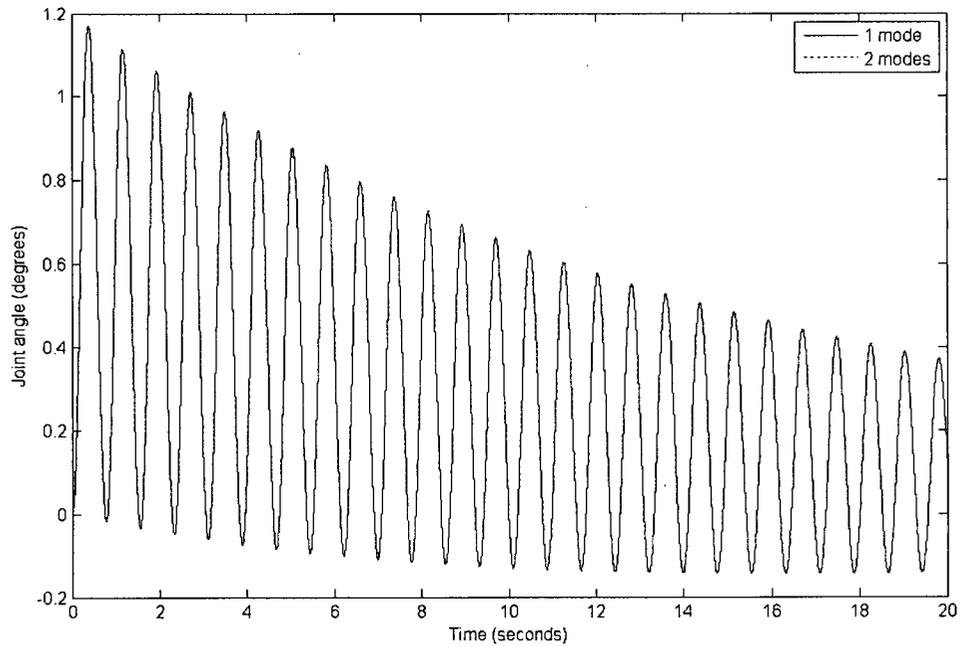


Figure 4.1 Case 1: Joint angle responses of 1-mode model and 2-mode models with an initial tip deflection of 51.2 mm.

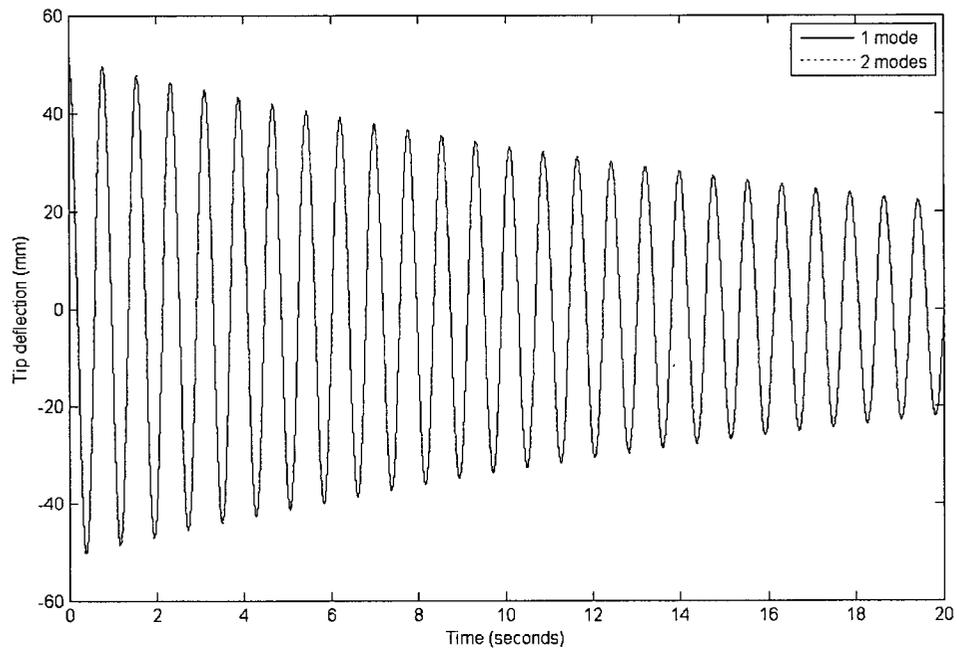


Figure 4.2 Case 1: Link tip deflection responses of 1-mode model and 2-mode models with an initial tip deflection of 51.2 mm.

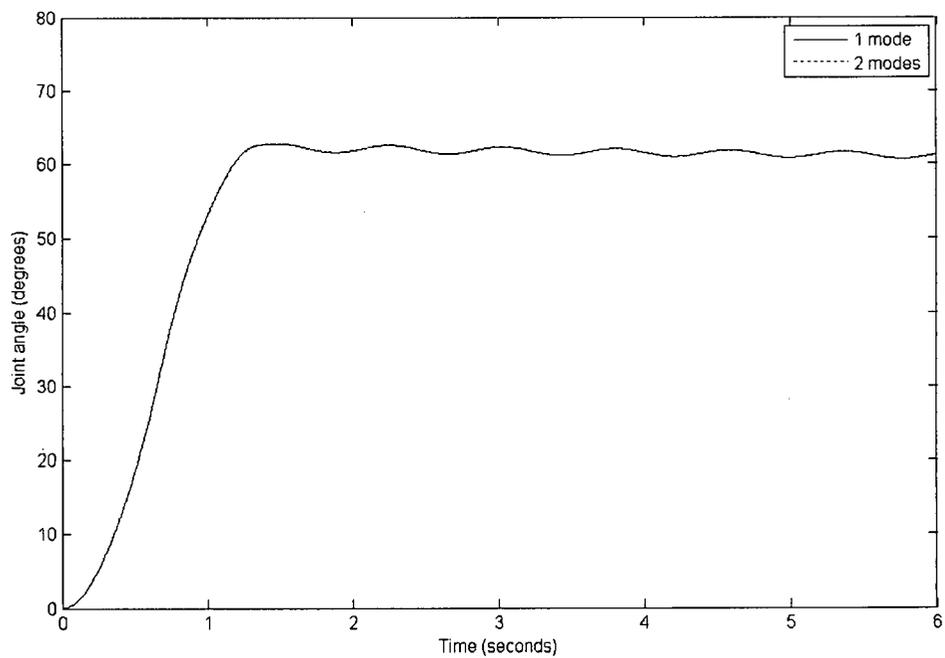


Figure 4.3 Case 2: Joint angle responses of 1-mode model and 2-mode models under a bang-bang control input ($\theta_1(0) = 0$, $w_{tip}(0) = 0$).

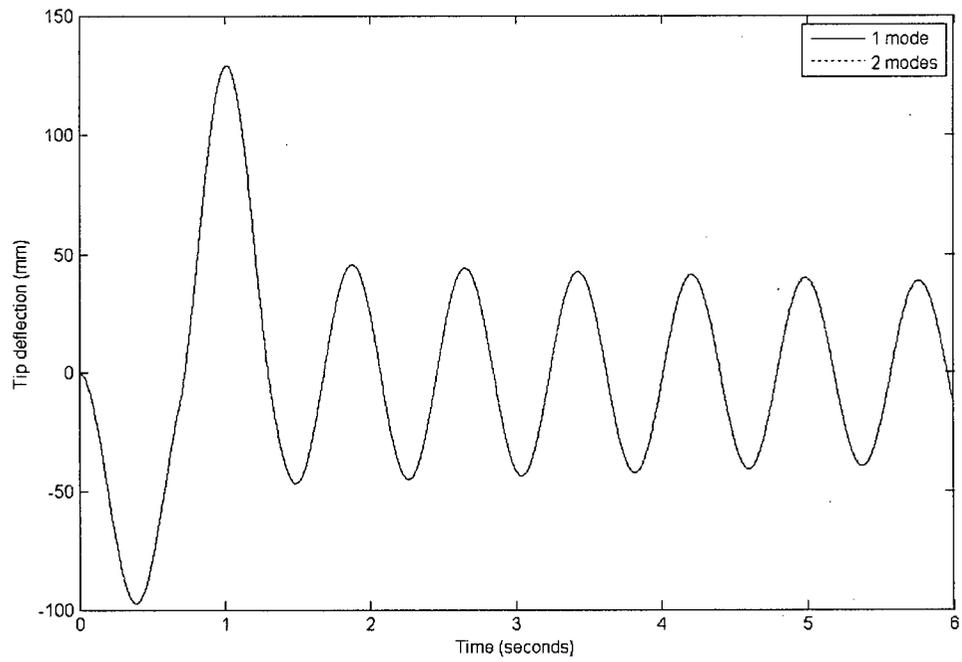


Figure 4.4 Case 2: Link tip deflection responses of 1-mode model and 2-mode models under a bang-bang control input ($\theta_1(0) = 0, w_{tip}(0) = 0$).

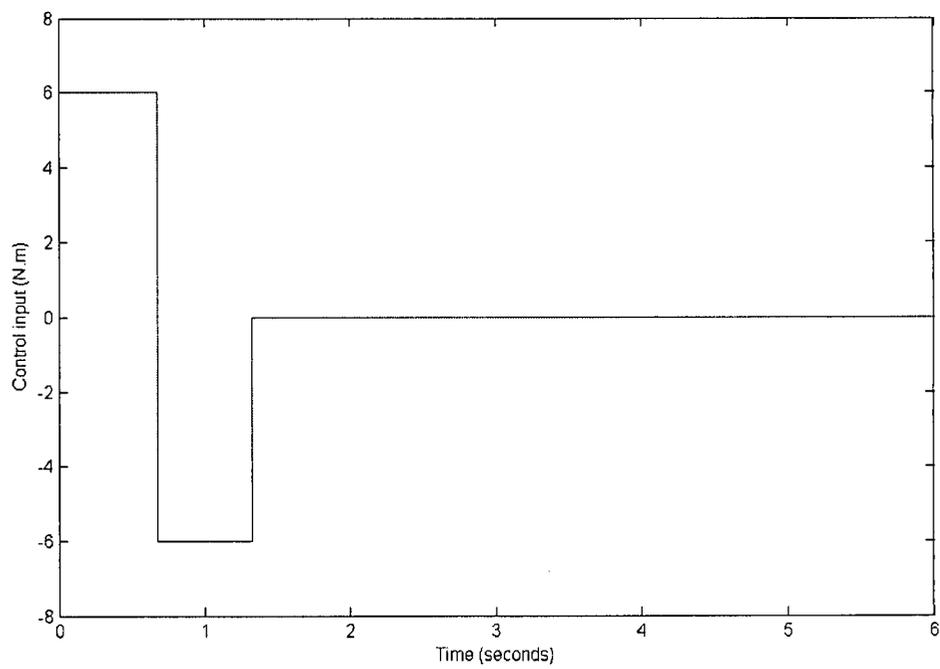


Figure 4.5 Case 2: Bang-bang control input.

4.1.2 Nonlinearity

From equations (2.92) and (2.93) we obtain the dynamic model of the single-link flexible robotic manipulator using the first flexible mode as

$$\begin{aligned} & \left\{ \left[J_{h1} + J_{o1} + J_p + m_p l_1^2 \right] + \left[m_1 + m_p \phi_{11}(l_1)^2 \right] \delta_{11}^2(t) \right\} \ddot{\theta}_1 \\ & + \left(I_{111} + m_p l_1 \phi_{11}(l_1) + J_p \phi_{11}'(l_1) \right) \ddot{\delta}_{11}(t) \\ & + \left[2m_1 + 2m_p \phi_{11}(l_1)^2 \right] \delta_{11}(t) \dot{\delta}_{11}(t) \dot{\theta}_1 + b_{\theta_1} \dot{\theta}_1 = \tau_1 \end{aligned} \quad (4.4)$$

$$\begin{aligned} & \left[I_{111} + m_p l_1 \phi_{11}(l_1) + J_p \phi_{11}'(l_1) \right] \ddot{\theta}_1 + \left[m_1 + m_p \phi_{11}(l_1)^2 + J_p \phi_{11}'(l_1)^2 \right] \ddot{\delta}_{11}(t) \\ & - \left[m_1 + m_p \phi_{11}(l_1)^2 \right] \delta_{11}(t) \dot{\theta}_1^2 + I_{3111} \delta_{11} + b_{\delta_{11}} \dot{\delta}_{11} = 0 \end{aligned} \quad (4.5)$$

where I_{111} and I_{3111} are constant parameters which depend on the mode shape function of the first flexible mode, and are defined as

$$\begin{aligned} I_{111} &= \int_0^l \rho_1 \phi_{11}(x_1) x_1 dx_1 \\ I_{3111} &= \int_0^l (EI)_1 \phi_{11}''(x_1)^2 dx_1 \end{aligned} \quad (4.6)$$

From these equations we notice that the dynamic model of the system is nonlinear. The nonlinearity of the system is due to the following terms in the dynamic equations: $\delta_{11}^2(t)$, $\delta_{11}(t) \dot{\delta}_{11}(t) \dot{\theta}_1$, and $\delta_{11}(t) \dot{\theta}_1^2$. In the development of the nonlinear dynamic equations of the system in Chapter 2, we have assumed that the link deflection is small. This implies that the higher order terms involving products of deformations can be neglected. It follows that if the joint angle velocity is not very large, the nonlinearity of the system is not significant and can be neglected.

A linear model for the single-link flexible robotic manipulator can be obtained using the techniques presented in Section 2.4. Assuming that the equilibrium operating point of the system is at the initial position of the link ($\theta_1 = 0$, $\delta_{11} = 0$), the linear model of the system is given by

$$\begin{aligned}
& \left[J_{h1} + J_{o1} + J_p + m_p l_1^2 \right] \ddot{\theta}_1 \\
& + \left(I_{111} + m_p l_1 \phi_{11}(l_1) + J_p \phi_{11}'(l_1) \right) \ddot{\delta}_{11}(t) + b_{\theta_1} \dot{\theta}_1 = \tau_1 \\
& \left[I_{111} + m_p l_1 \phi_{11}(l_1) + J_p \phi_{11}'(l_1) \right] \ddot{\theta}_1 \\
& + \left[m_1 + m_p \phi_{11}(l_1)^2 + J_p \phi_{11}'(l_1)^2 \right] \ddot{\delta}_{11}(t) + I_{3111} \delta_{11} + b_{11} \dot{\delta}_{11} = 0
\end{aligned} \tag{4.7}$$

In Chapter 2 we have found that the effect of payload on the mode shapes is not significant. Accordingly, the system model will use a constant mode shape for the first flexible mode. A linear state-space model for the system is obtained from equations (4.7), (2.114) and (2.115) as

$$\begin{aligned}
\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}u \\
\mathbf{y} &= \mathbf{C}\mathbf{x}
\end{aligned} \tag{4.8}$$

where $\mathbf{x} = [\theta_1, \delta_{11}, \dot{\theta}_1, \dot{\delta}_{11}]^T$ is the state vector and $\mathbf{y} = [\theta_1, w_{tip}]^T$ is the output vector. The system input u is the joint torque. The outputs are the joint angle θ_1 and the deflection w_{tip} at the tip of the link. The numerical values of the system matrices \mathbf{A} , \mathbf{B} , and \mathbf{C} are given in Appendix A.1.2.

Next, computer simulations are used to compare the linear and nonlinear system models. Two cases are considered. The control input and initial states of the first case are the same as those of case 1 in Section 4.1.1. There is no control input to the system. The initial joint angle is zero and the initial tip deflection is 51.2 mm. The simulation results are shown in Figure 4.6 and Figure 4.7. The control input and initial states for the second case are the same as those of case 2 in Section 4.1.1. Both the initial joint angle and the initial tip deflection are equal to zero. The input to the system is a symmetric bang-bang torque with the magnitude of 6 N.m, as shown in Figure 4.5. The simulation results are shown in Figure 4.8 and Figure 4.9. From the simulation results we observe that the system responses for the linear model and the nonlinear model are very close in both cases.

In summary, the nonlinearity of the single-flexible link manipulator considered here is not significant and can be neglected when the link deflection is small and when the velocity terms are not very large. Accordingly, the linear model in equation (4.8) will be used in the

MPC controller design.

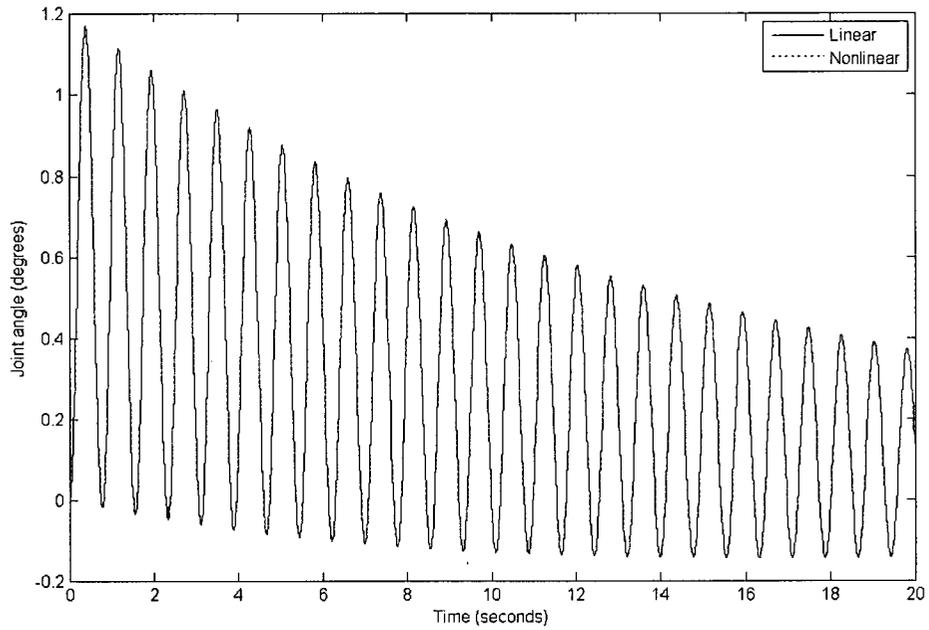


Figure 4.6 Case 1: Joint angle responses of the linear and nonlinear models with an initial tip deflection of 51.2 mm.

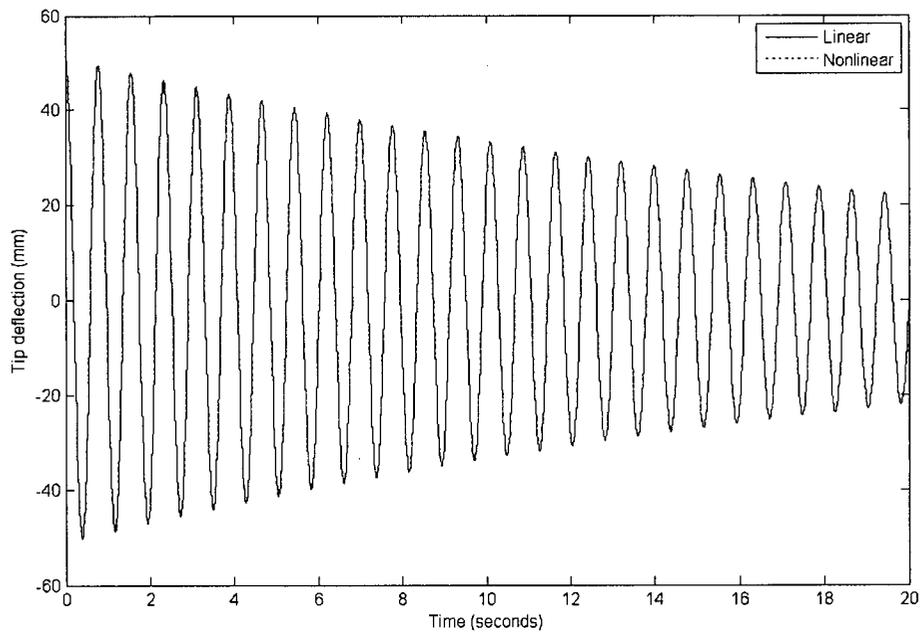


Figure 4.7 Case 1: Link tip deflection responses of the linear and nonlinear models with an initial tip deflection of 51.2 mm.

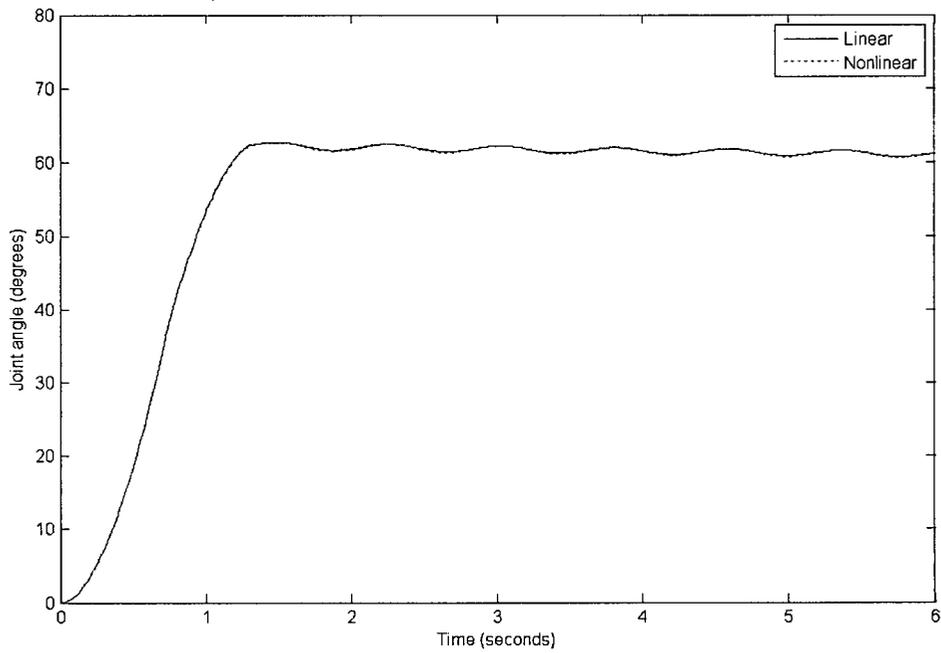


Figure 4.8 Case 2: Joint angle responses of the linear and nonlinear models under a bang-bang control input ($\theta_1(0) = 0, w_{tip}(0) = 0$).

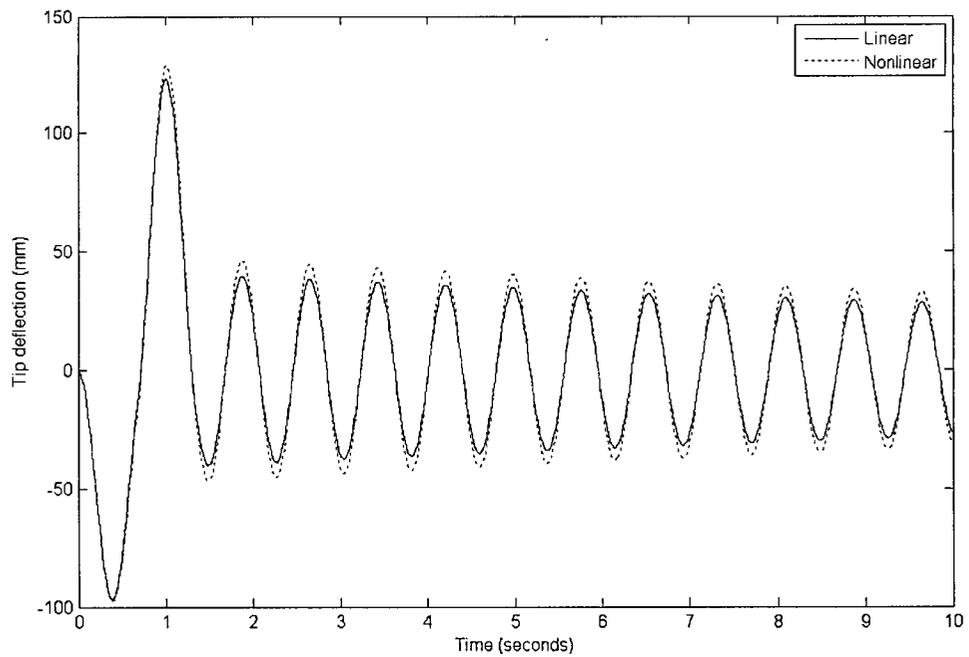


Figure 4.9 Case 2: Link tip deflection responses of the linear and nonlinear models under a bang-bang control input ($\theta_1(0) = 0, w_{tip}(0) = 0$).

4.1.3 Effects of Payload

The linear model of the single-link flexible robotic manipulator given by equation (4.8) is dependent on the system payload (m_p, J_p). Specifically, the elements of the system matrices **A** and **B** are functions of m_p and J_p , as shown in Appendix A.1.2.

Now the effects of the system payload on the linear dynamics of the robotic system are investigated through computer simulations. Two cases are considered. The control input and the initial states of the first case are identical to those of case 1 in Section 4.1.1. In this case there is no control input to the system. Also, the initial joint angle is zero and the initial tip deflection is 51.2 mm. The simulation results are shown in Figure 4.10 and Figure 4.11. The control input and the initial states of the second case are identical to those of case 2 in Section 4.1.1. In this case, both the initial joint angle and the initial tip deflection are equal to zero. The input to the system is a symmetric bang-bang torque with the magnitude 6 N.m, as shown in Figure 4.5. The simulation results are shown in Figure 4.12 and Figure 4.13. From the simulation results it is observed that the payload has a significant effect on the system dynamics. Under the same input and initial-state conditions, increasing the system payload will increase the link deflection. Increasing the system payload reduces the natural frequencies of the system as given in Table 2.3. This corresponds to a system that is less stiff, resulting in increased deflection (flexibility).

In summary, the effect of the payload on system dynamics is significant and cannot be neglected. If the system payload is not known and if direct measurement is not available, a system identification module has to be designed and incorporated in order to obtain the correct linear model of the system. A system identification module for the single-link flexible manipulator will be designed in Section 4.3.

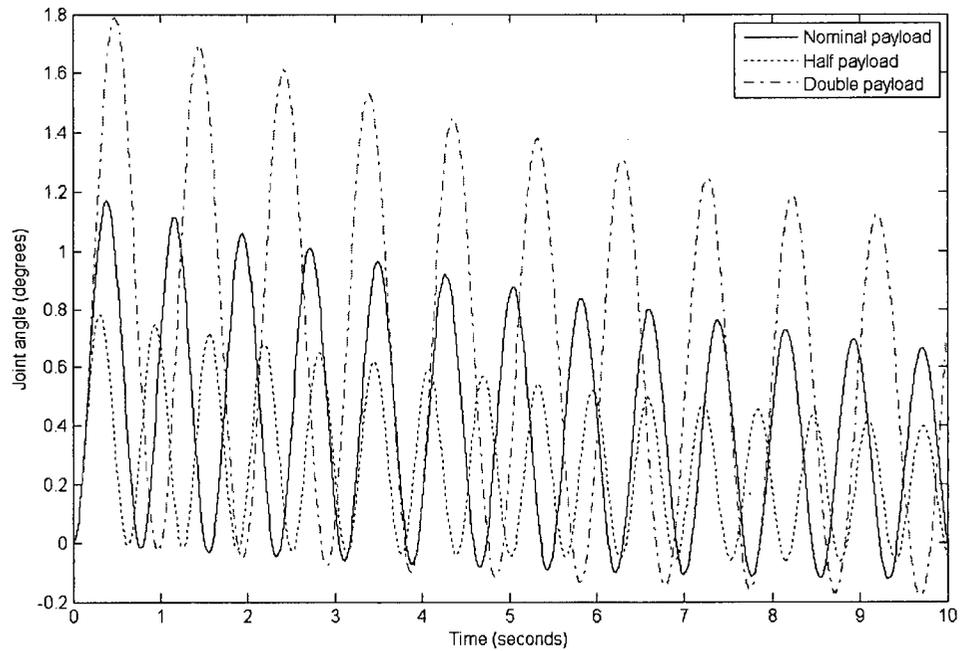


Figure 4.10 Case 1: Joint angle responses of the linear model for different payloads with an initial tip deflection of 51.2 mm.

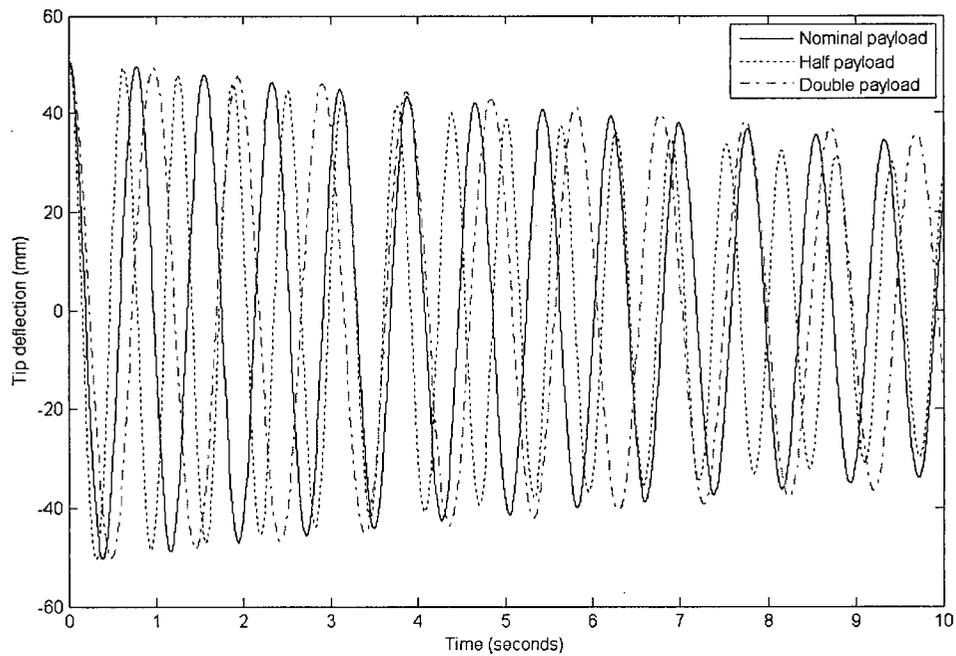


Figure 4.11 Case 1: Link tip deflection responses of the linear model for different payloads with an initial tip deflection of 51.2 mm.

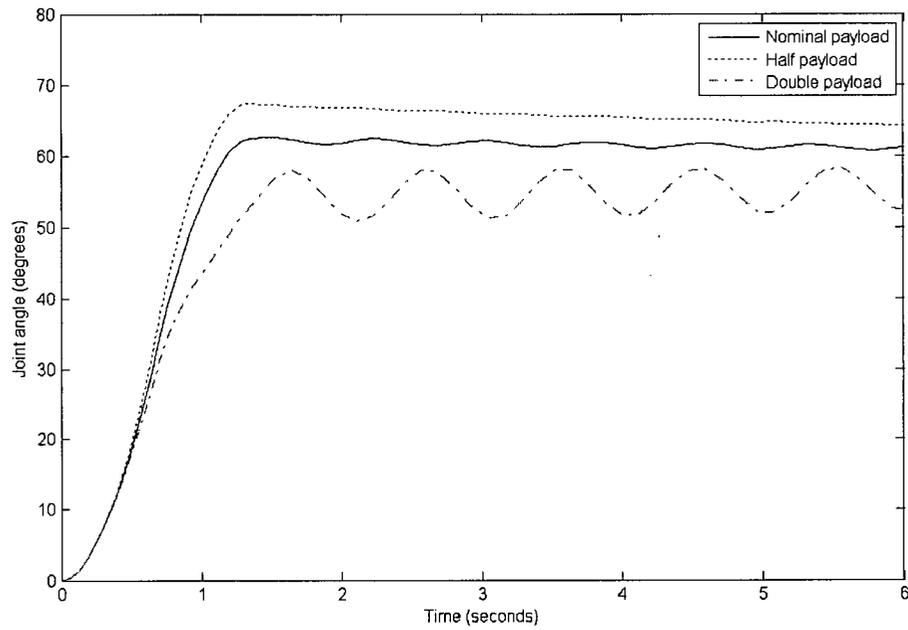


Figure 4.12 Case 2: Joint angle responses of the linear model for different payloads under a bang-bang control input ($\theta_1(0) = 0, w_{tip}(0) = 0$).

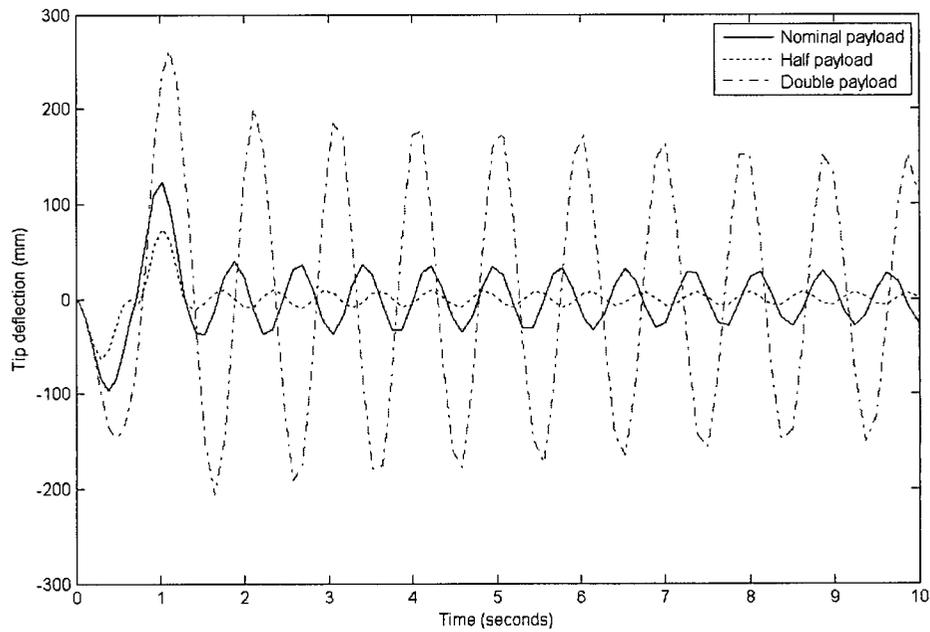


Figure 4.13 Case 2: Link tip deflection responses of the linear model for different payloads under a bang-bang control input ($\theta_1(0) = 0, w_{tip}(0) = 0$).

4.2 MPC Controller

In this section, we will develop a model predictive control (MPC) scheme for the single-link flexible manipulator using the techniques presented in Chapter 3. The MPC is a model-based control algorithm. The controller performance is directly related to the accuracy of the internal model of the controller. In Section 4.1, we have shown that the linear continuous-time state-space model with just the first flexible mode is adequate to accurately capture the flexible dynamics of the robotic manipulator.

First we develop a discrete state-space internal model for the MPC controller. In Chapter 2 we established that the effect of payload changes on the mode shapes was not significant. Accordingly, the internal model will use a constant first flexible mode. With a data sampling period of $T_s = 10$ ms and a nominal payload, by using equations (3.8) and (3.9) we obtain the numerical discrete-time state-space internal model of the MPC controller as

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k)\end{aligned}\quad (4.9)$$

where $\mathbf{x}(k) = [\theta_1(k), \delta_{11}(k), \dot{\theta}_1(k), \dot{\delta}_{11}(k)]^T$ is the state vector and $\mathbf{y}(k) = [\theta_1(k), w_{tip}(k)]^T$ is the output vector. The system input $u(k)$ is the joint torque. The outputs are the joint angle $\theta_1(k)$ and the deflection $w_{tip}(k)$ at the tip of the link. The numerical values of the system matrices **A**, **B**, and **C** are given in Appendix A.1.3.

The cost function of the MPC controller is

$$J = \sum_{i=1}^{H_p} Q_{yw} \left\{ [\hat{y}_1(k+i|k) - r_\theta]^2 + Q_w [\hat{y}_2(k+i|k)]^2 \right\} + \sum_{i=1}^{H_u} R_u \Delta \hat{u}(k+i-1|k)^2 \quad (4.10)$$

where r_θ is the setpoint for joint angle. The setpoint for link deflection is zero.

We take the input and output constraints to be

$$\begin{aligned}-6 \text{ N.m} &\leq u(k+i|k) \leq 6 \text{ N.m} \\ -25 \text{ mm} &\leq y_2(k+i|k) \leq 25 \text{ mm}\end{aligned}\quad (4.11)$$

A computationally efficient ‘anti-windup’ MPC controller is designed using the strategy developed in Chapter 3. This controller is an unconstrained MPC controller. The fuzzy tuner

handles the constraints. The control input at time k of the unconstrained MPC is calculated using the analytical equation (3.45). The fuzzy tuner adjusts the MPC parameters such that the constraints are satisfied.

Next we will compare the performance of this computationally efficient ‘anti-windup’ MPC controller with the constrained MPC controller. The constrained MPC controller solves a QP optimization problem at each time step. The constrained MPC controller is designed using the Model Predictive control toolbox in MATLAB (Mathworks, 2005).

We use Simulink[®] to perform the simulation. The Simulink block diagram is shown in Figure 4.14. Both constrained and unconstrained MPC controllers can be tested in the MPC control block to generate the control input signal. The nonlinear dynamic model of the single-link flexible robotic manipulator with the first two flexible modes is used as the plant. The saturation block is used to limit the control input signal to the upper and lower saturation values.

Two cases are tested. The MPC controller parameters are as given in Table 4.1. The joint angle setpoint is $\frac{\pi}{3}$ for both cases.

In case 1, the MPC controllers have the same parameters. Figure 4.15 through Figure 4.17 show the simulation results. In this case the constraints are inactive. Both the control input and the link deflection are within their ranges $|u| \leq 6 \text{ N.m}$, $|y_2| \leq 25 \text{ mm}$. The simulation results verify that the constrained and unconstrained MPC solutions are identical when the constraints are inactive.

In case 2, the input constraint is active, as shown in Figure 4.18. The constrained MPC considers the constraints in the solution, so the control input for constrained MPC is within its range $|u| \leq 6 \text{ N.m}$ and reaches its maximum value in the first 3 time steps. If the unconstrained MPC uses the same set of control parameters as the constrained MPC, the calculated control input will exceed its maximum range as shown in Figure 4.18. In order to satisfy the input constraint, the unconstrained MPC has to increase the value of the input rate weight R_u from 3 to 5. This will be done by the fuzzy tuner. Figure 4.18 through Figure 4.21 show the simulations for the constrained and unconstrained MPC controllers with different input rate weight values. The simulation results show that the responses of the unconstrained

MPC are very close to those of the constrained MPC when the constraints are active.

In summary, the developed computationally efficient ‘anti-windup’ MPC controller has the same optimal solution as the constrained MPC when the constraints are inactive; and has the sub-optimal solution when the constraints are active. The system response performance of the unconstrained MPC is comparable to that of the constrained MPC response.

Table 4.1 MPC controller parameters.

		r_θ	H_p	H_u	Q_{yw}	Q_w	R_u
Case 1	Constrained MPC	$\frac{\pi}{3}$	160	9	20	30	8
	Unconstrained MPC	$\frac{\pi}{3}$	160	9	20	30	8
Case 2	Constrained MPC	$\frac{\pi}{3}$	160	9	20	30	3
	Unconstrained MPC	$\frac{\pi}{3}$	160	9	20	30	5

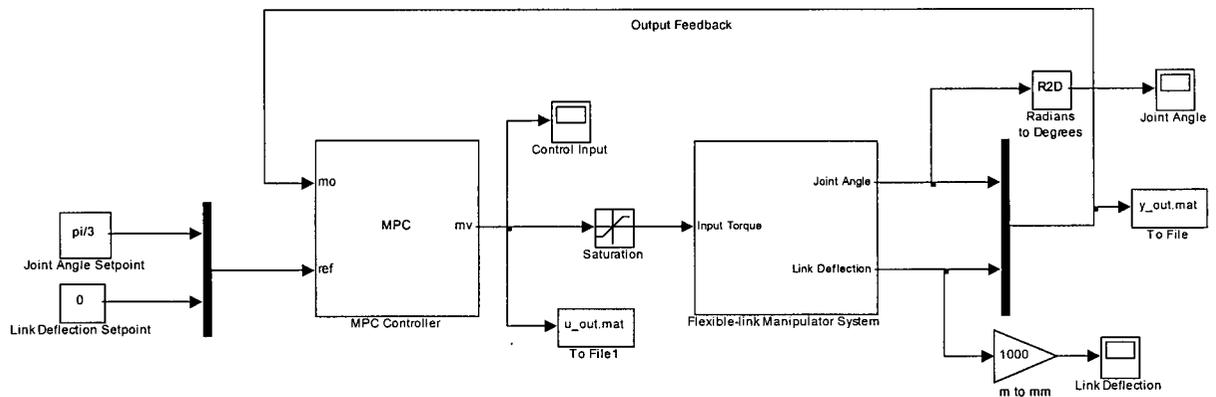


Figure 4.14 Simulation block diagram.

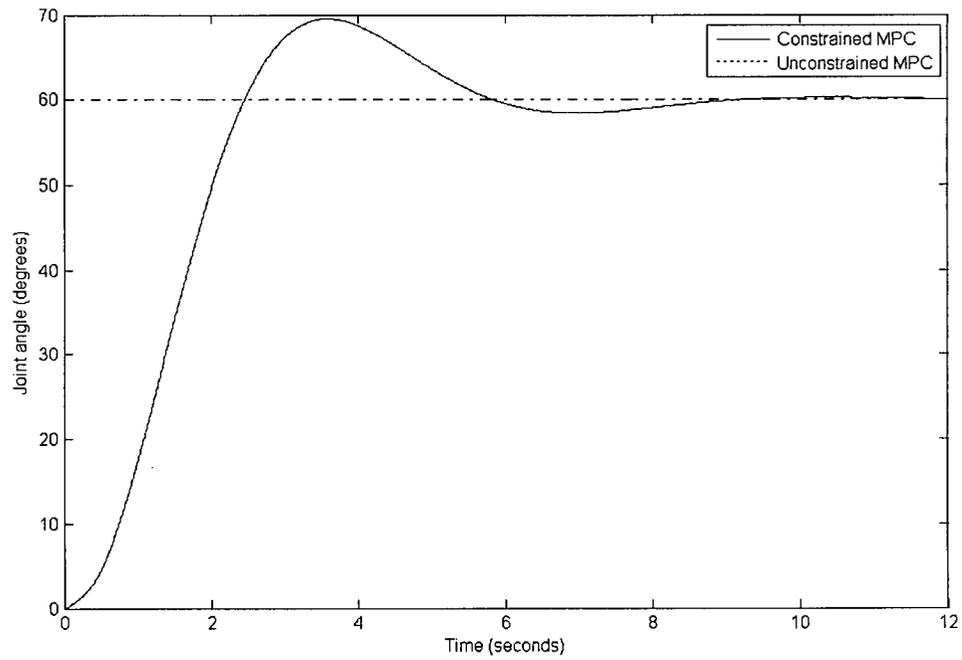


Figure 4.15 Case 1: Joint angle responses for the constrained and unconstrained MPC.

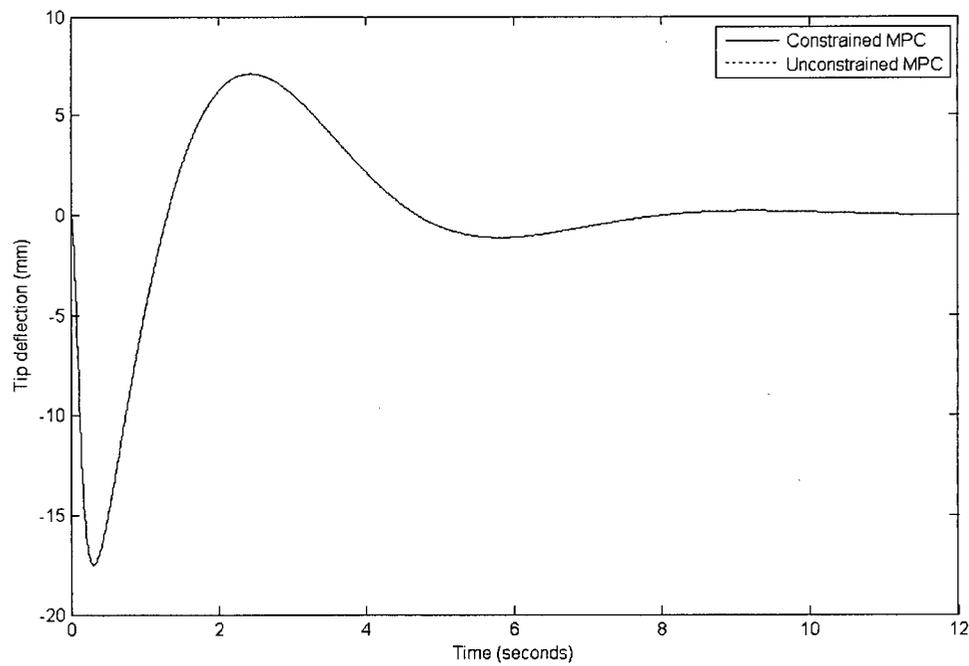


Figure 4.16 Case 1: Link tip deflection responses for the constrained and unconstrained MPC.

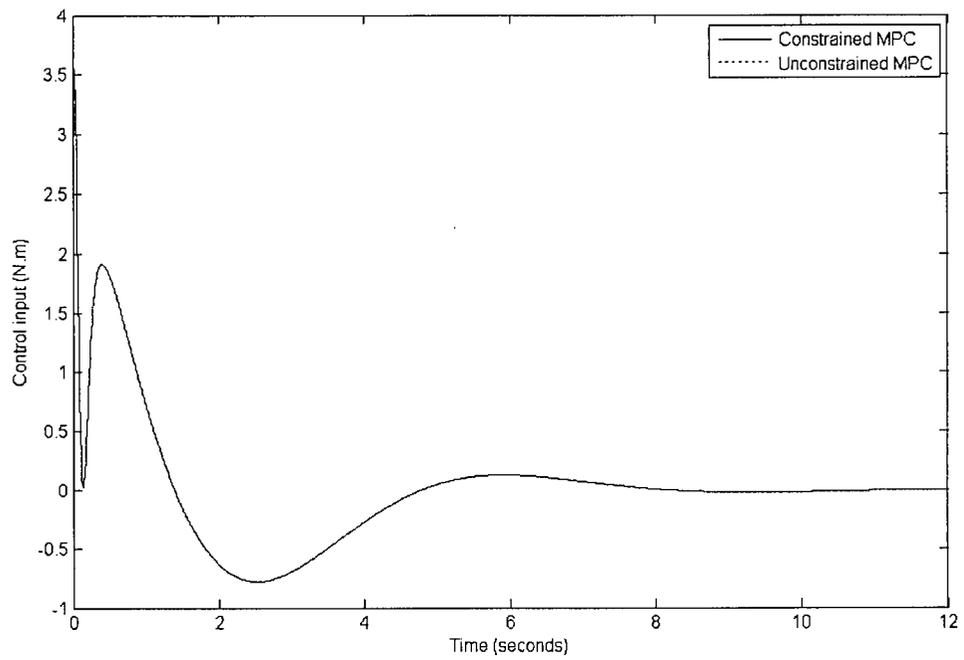


Figure 4.17 Case 1: Control inputs for the constrained and unconstrained MPC.

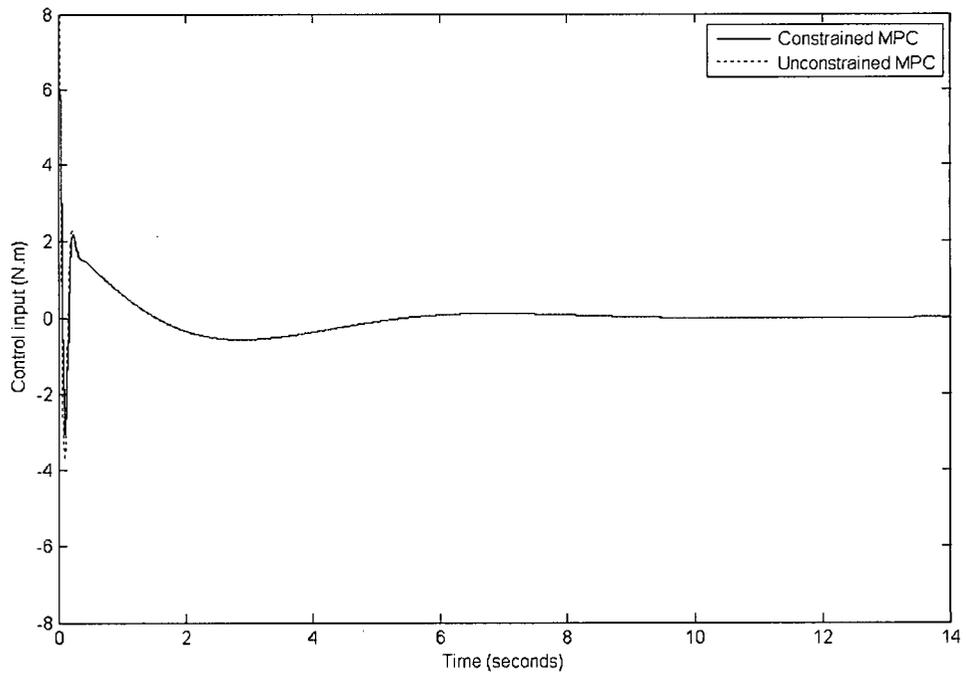


Figure 4.18 Case 2: Control inputs for the constrained and unconstrained MPC with the same parameters.

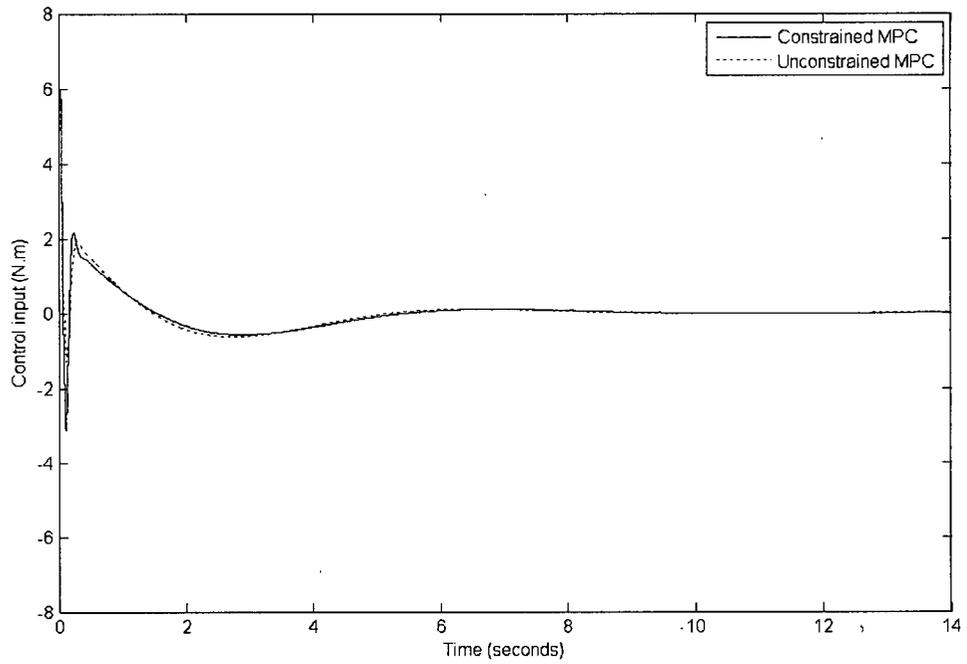


Figure 4.19 Case 2: Control inputs for the constrained and unconstrained MPC with different parameters.

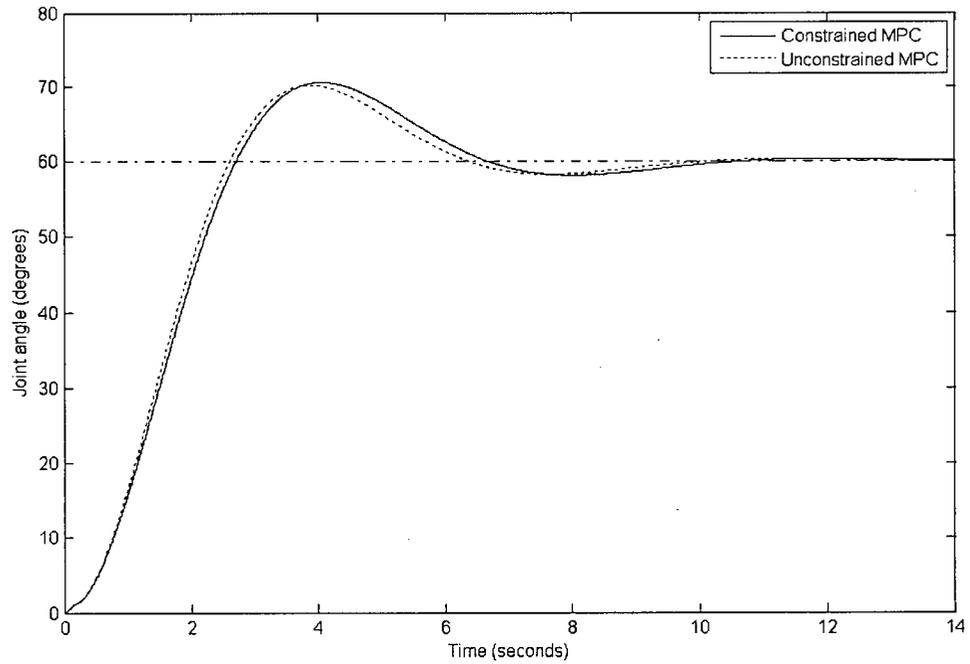


Figure 4.20 Case 2: Joint angle responses for the constrained and unconstrained MPC with different parameters.

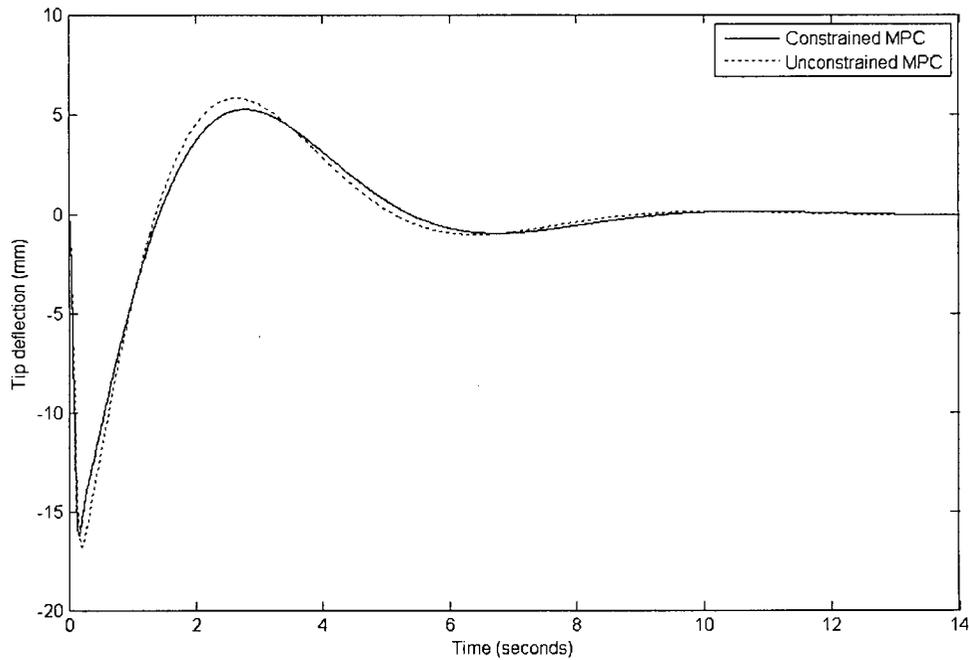


Figure 4.21 Case 2: Link tip deflection responses for the constrained and unconstrained MPC with different parameters.

4.3 Payload Identification

In Section 4.1.3 we have shown the effect of the payload on system dynamics is significant and cannot be neglected. If the system is not known, and if direct measurement is not available, we need to develop a system identification module to find the unknown payload. We will use the techniques developed in Section 3.3 to design the system identification module for the single-flexible link robotic manipulator. The continuous-time state-space model of the system is given in Appendix A.1.2. The system matrices \mathbf{A} and \mathbf{B} are dependent on the payload. We assume that the nominal model is the one with a nominal payload. The structure matrices as denoted by \mathbf{A}_s and \mathbf{B}_s have the entry NaN at those elements that correspond to the unknown parameters which have to be estimated. The structure matrices are defined by

$$\mathbf{As} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & NaN & NaN & NaN \\ 0 & NaN & NaN & NaN \end{bmatrix}, \quad \mathbf{Bs} = \begin{bmatrix} 0 \\ 0 \\ NaN \\ NaN \end{bmatrix} \quad (4.12)$$

The unknown parameters of the structure matrices are determined by using the system identification module and the input and output data of the system. The system payload m_p , J_p can be calculated by solving the following equations using the elements of the identified system matrix \mathbf{B} :

$$\frac{0.404 + 4.197m_p + 9.167J_p}{0.769 + 17.669J_p + 8.0m_p + 0.959J_p m_p} = \mathbf{B}(3) \quad (4.13)$$

$$\frac{-2.049m_p - 0.229 - 3.028J_p}{0.769 + 17.669J_p + 8.0m_p + 0.959J_p m_p} = \mathbf{B}(4)$$

Next computer simulations are carried out to test the accuracy of the system identification module. An MPC controller moves the link with a payload at low speed from the initial position to 5° and then moves it back to the initial position. The nonlinear model of the single-link flexible robotic manipulator with the first two flexible modes is used as the plant model. The Simulink block diagram in Figure 4.14 is used to generate the input and output data for the system. The simulation results for two different system payloads are presented below.

Case 1:

In this case, the payload is nominal, and we assume that there are no disturbances and measurement noise in the system. First the Simulink model of the system is tested. The input and output responses of the system are shown in Figure 4.22. The input and output data are placed in the Matlab workspace. We split the data record into two parts. The first part is used to identify the system model and the second part is used to verify the accuracy of the identified model. Figure 4.23 shows that a perfect fit is achieved by the model. In this case the numerical values of the system matrix \mathbf{B} are

$$\mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 0.524358 \\ -0.263886 \end{bmatrix} \quad (4.14)$$

By solving equation (4.13) the unknown system payload is obtained as: $m_p = 0.38\text{kg}$,
 $J_p = 9.794391e-004\text{kg.m}^2$.

Case 2:

In this case the payload is equal to double the nominal payload, and we assume that there are no disturbances and measurement noise in the system. The input and output responses of the system are shown in Figure 4.24. Figure 4.25 shows that a perfect model fit is achieved. By solving equation (4.13) the unknown system payload is obtained as $m_p = 0.765593\text{kg}$,
 $J_p = 0.001766\text{kg.m}^2$.

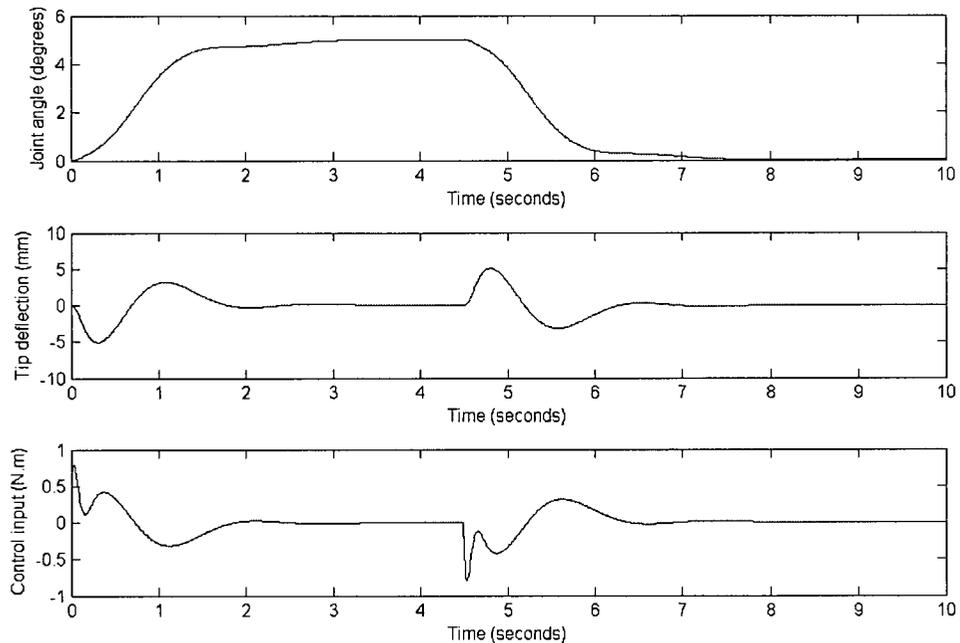


Figure 4.22 Case 1: Input and output responses of the single-link flexible manipulator (nominal payload).

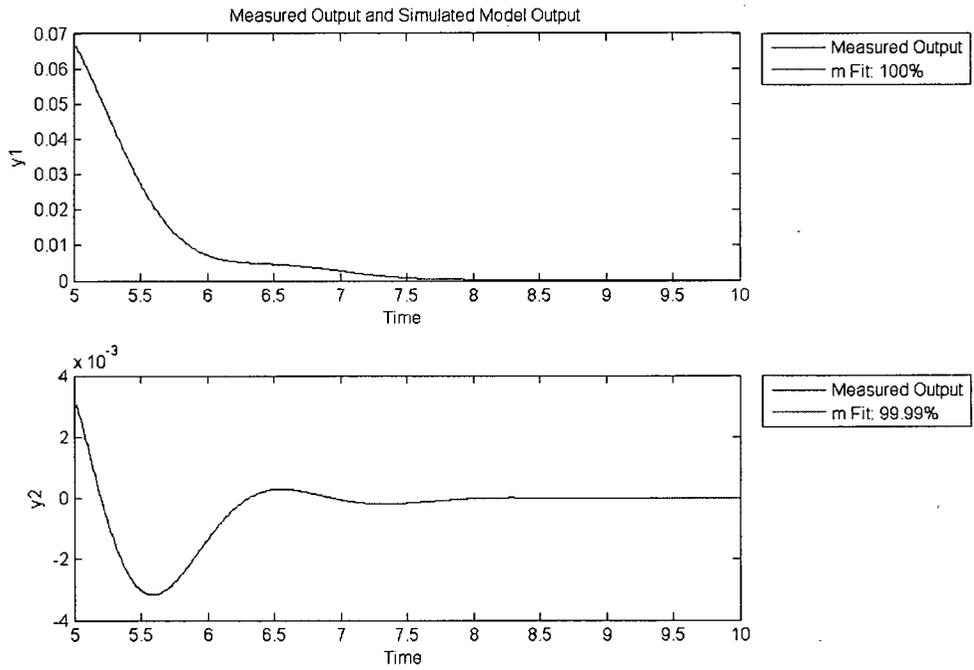


Figure 4.23 Case 1: Measured outputs and simulated model outputs (nominal payload).

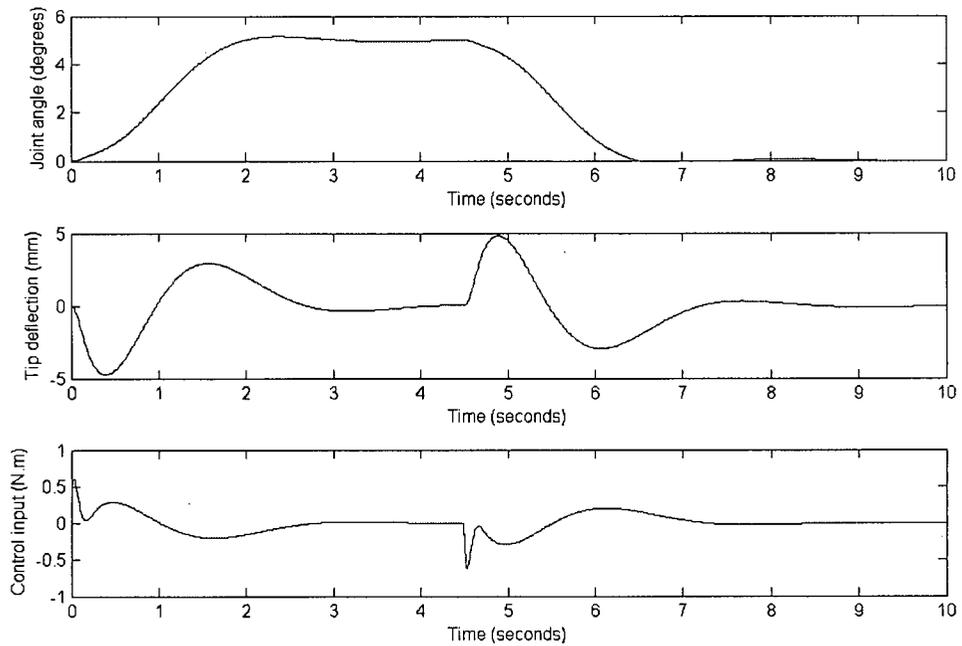


Figure 4.24 Case 2: Input and output responses of the single-link flexible manipulator (double the nominal payload).

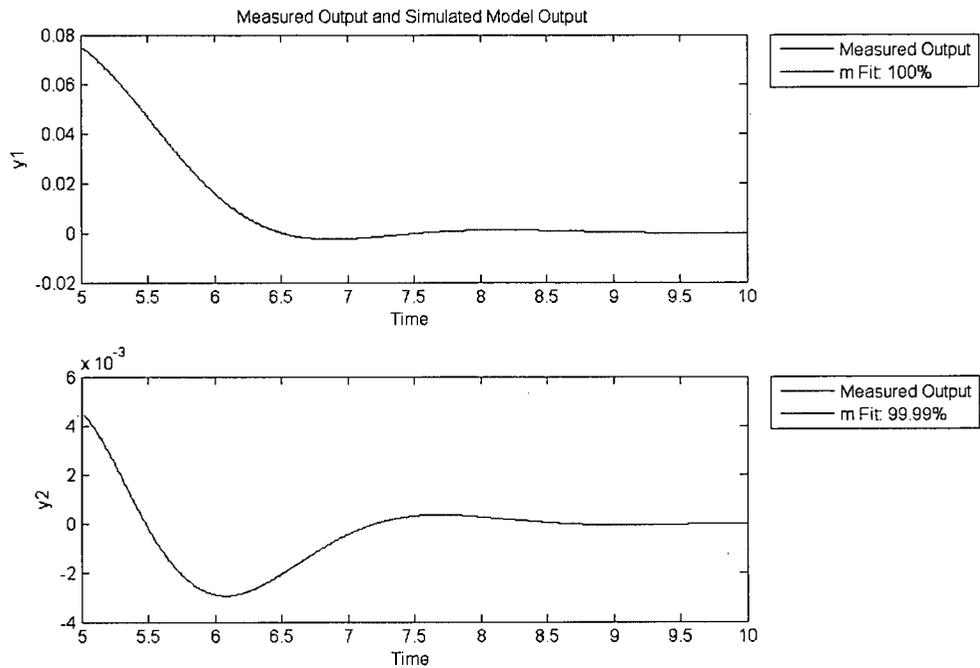


Figure 4.25 Case 2: Measured outputs and simulated model outputs (double the nominal payload).

4.4 Fuzzy Tuner

A fuzzy tuner for the single-link flexible robotic manipulator is designed using the procedures presented in Chapter 3. Matlab fuzzy logic control toolbox is used to build the fuzzy inference system (FIS). Figure 4.26 shows the FIS input-output diagram of the system.

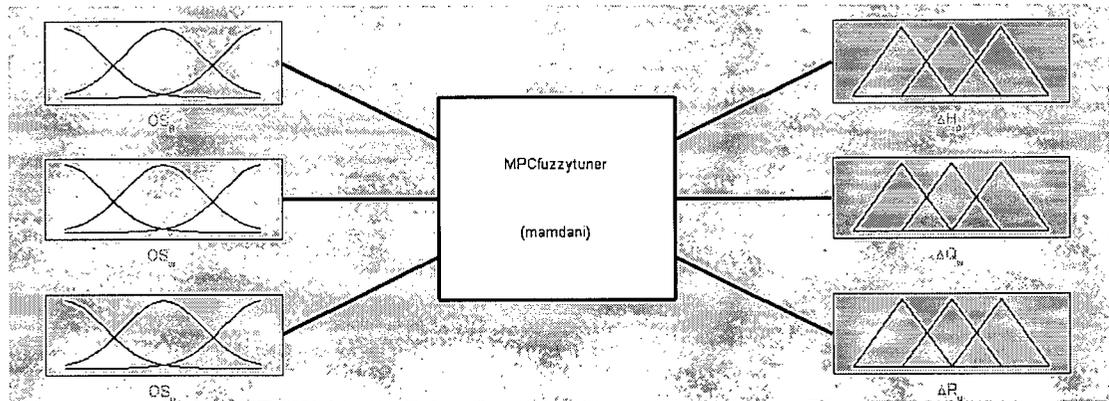


Figure 4.26 FIS input-output diagram.

The membership functions for the input and output variables are defined in Section 3.4.2, and are shown in Figure 3.7 and Figure 3.8. The link deflection constraint (maximum allowed link deflection) is $r_w = 25$ mm and the control input constraint (maximum control input allowed) is $r_u = 6$ N.m.

Based on the analysis in Section 3.4.1 and through extensive computer simulations, the primary functions and the side effects of the tuning parameters H_p , Q_w , and R_u are determined. These are summarized in Table 4.2.

Table 4.2 Effects of the MPC tuning parameters.

Controller Parameters	Functions	Undesired Side Effects
H_p	Reduces joint angle overshoot; increases stability.	Can slow down the system response. Increases the computational load.
Q_w	Reduces link deflection and control input.	Increases joint angle tracking error and joint angle overshoot.
R_u	Reduces control input and link deflection.	Can slow down the system response. Increases joint angle overshoot.

The rule base of the FIS is developed using Table 4.2, the general linguistic MPC tuning rules in Table 3.2, and the knowledge gained through computer simulations. The fuzzy rule base has 32 rules as given in Table 4.3.

Table 4.3 Rule base of the FIS.

	IF			THEN				IF			THEN		
	OS_θ	OS_w	OS_u	ΔH_p	ΔQ_w	ΔR_u		OS_θ	OS_w	OS_u	ΔH_p	ΔQ_w	ΔR_u
1	L	L		L			17	M	VS		VS		
2	L	M		L			18	S	L		S		
3	L	S		S			19	S	M		VS		
4	M	L		L			20	S	S		VS		
5	M	M		M			21	S	VS		VS		
6	M	S		S			22	VS	L		VS		

	IF			THEN				IF			THEN		
7	S	L		S			23		VS	M		VS	
8	S	M		S			24		VS	S		VS	
9	S	S		S			25		VS	VS		VS	
10		L	L		M		26	L	VS			S	
11		L	M		S		27	M	VS			S	
12		L	S		VS		28	S	VS			S	
13		L	VS		VS		29			L	S		M
14		M	L		M		30			M	S		S
15		M	M		S		31			S	S		S
16		M	S		VS		32			VS	S		VS

4.5 Computer Simulations of IMPC

The developed intelligent model predictive control (IMPC) algorithm is now simulated for motion control of the single-link flexible robotic manipulator. The system payload is taken to be nominal. The nonlinear dynamic model of the single-link flexible robotic manipulator with first two flexible modes is used as the plant. The internal model of MPC is the linear model at the zero initial position with the first flexible mode. Three different cases of motion configurations are tested. Table 4.4 shows the predetermined MPC parameters for both cases. Here T_s is the time step; r_w is the link deflection constraint; r_u is the control input constraint; H_{p_sen} , Q_{w_sen} , and R_{u_sen} are the sensitivity parameters of the tuning actions; H_u is the control horizon; and Q_{yw} is the overall weighting for the joint angle and link deflection. The joint angle setpoints and the tuning parameters for the three cases are given in Table 4.5. Here r_θ is the joint angle setpoint; H_{p_ini} is the initial prediction horizon; H_{p_end} is the final prediction horizon generated by the fuzzy tuner; Q_{w_ini} is the initial value for the link deflection weight; Q_{w_end} is its tuned value; R_{u_ini} is the initial input rate weight; and R_{u_end} is its tuned value.

Figure 4.27 through Figure 4.29 show the system responses of Case 1. The initial, intermediate, and final system responses are shown in these figures. It is observed that the desired system performance cannot be achieved by using the initial set of MPC parameters. The joint angle response has a large overshoot, and the link deflection and the control input

exceed their limits. The IMPC tunes the controller using the fuzzy knowledge until the desired system performance is achieved. The results show that the system responses using the tuned parameters from IMPC meet the design specifications. There is no joint angle overshoot, and the link deflection and the control input are within their limits.

The system responses for Case 2 and Case 3 using the IMPC control algorithm are shown in Figure 4.30 through Figure 4.35.

In Chapter 2 we have shown that if the system output is taken as the link-end deflection, then the system is a non-minimum phase one. For a non-minimum phase system, the step response initially starts to move in the direction opposite to the final steady-state value. This behavior is verified by the deflection responses for the two cases shown in Figure 4.28, Figure 4.31 and Figure 4.34. In these figures we observe that initially the tip response takes up negative values.

The simulation results verify that the IMPC control technique is quite effective in controlling the tip position of a single-link flexible robotic manipulator.

Table 4.4 Predetermined parameters of the MPC controller for the single-link manipulator.

T_s (ms)	r_w (mm)	r_u (N.m)	H_{p_sen}	Q_{w_sen}	R_{u_sen}	H_u	Q_{yw}
10	25	6	4.8	4.6	9.6	9	20

Table 4.5 Tuning parameters of the MPC controller for the single-link manipulator.

	r_θ	H_{p_ini}	H_{p_end}	Q_{w_ini}	Q_{w_end}	R_{u_ini}	R_{u_end}
Case 1	$\frac{\pi}{6}$	100	530	1	7.8	1	3.0
Case 2	$\frac{\pi}{3}$	100	584	1	14.4	1	6.8
Case 3	$\frac{\pi}{2}$	100	728	1	19.8	1	9.8

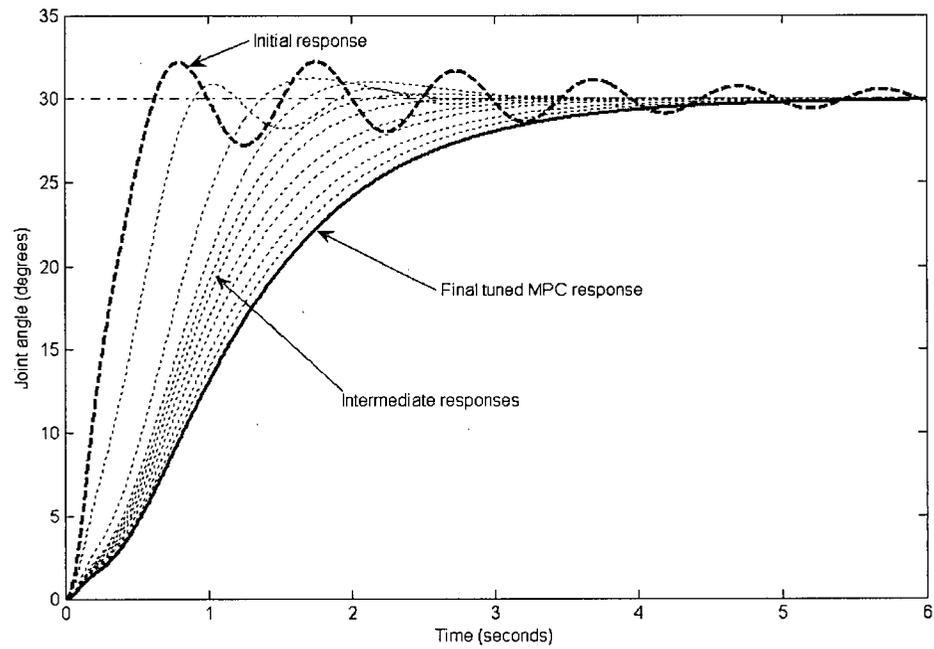


Figure 4.27 Case 1: Joint angle responses with IMPC.

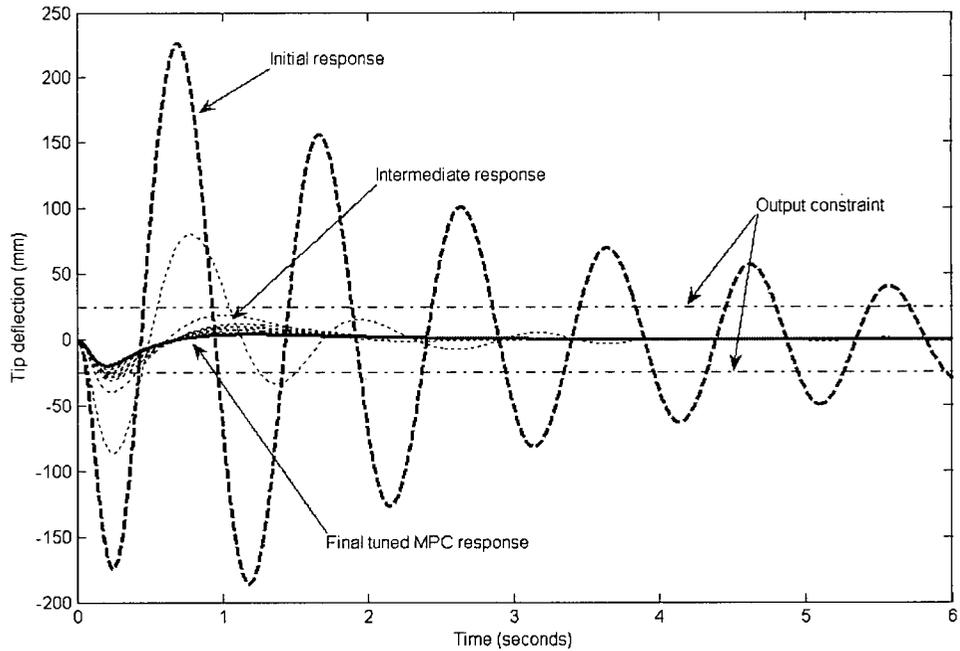


Figure 4.28 Case 1: Link tip deflection responses with IMPC.

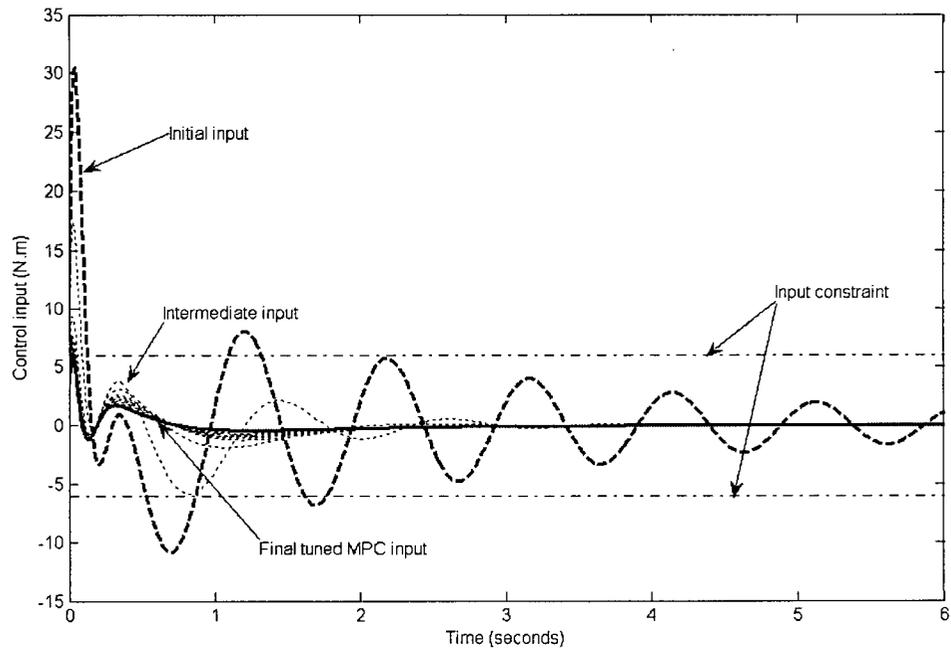


Figure 4.29 Case 1: Control input with IMPC.

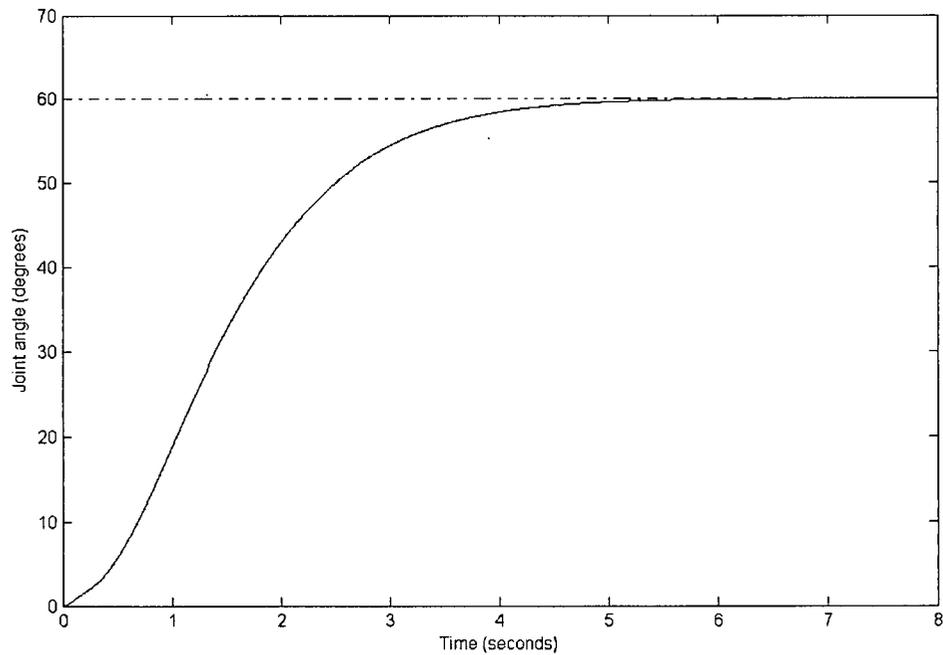


Figure 4.30 Case 2: Joint angle responses with IMPC.

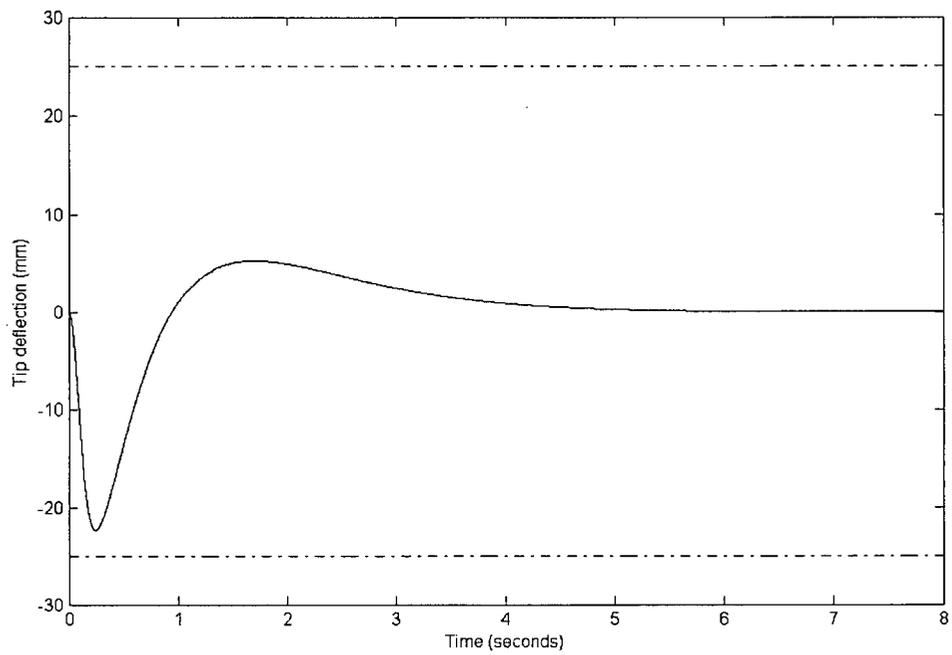


Figure 4.31 Case 2: Link tip deflection responses with IMPC.

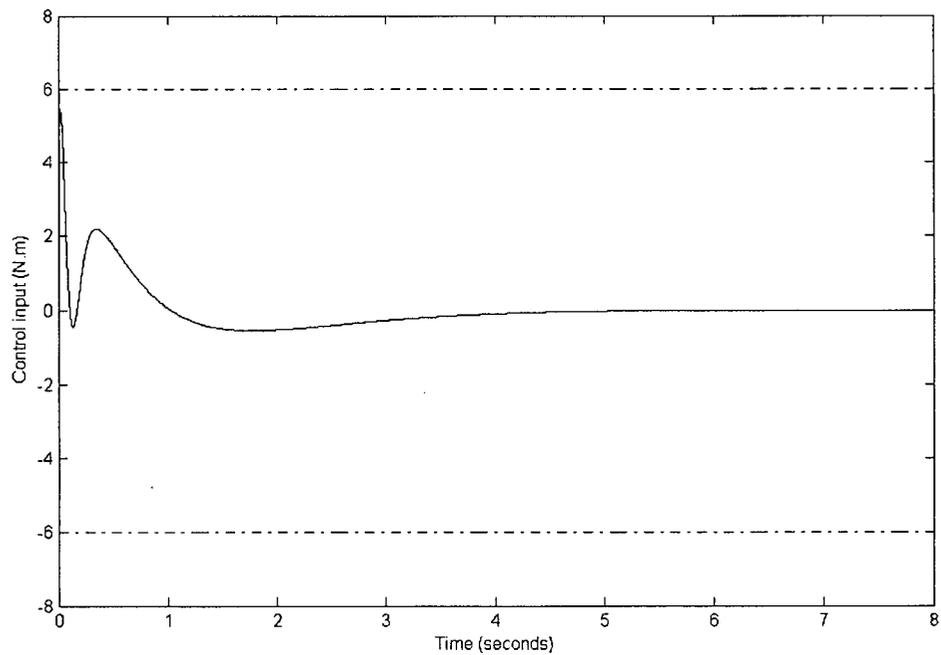


Figure 4.32 Case 2: Control input with IMPC.

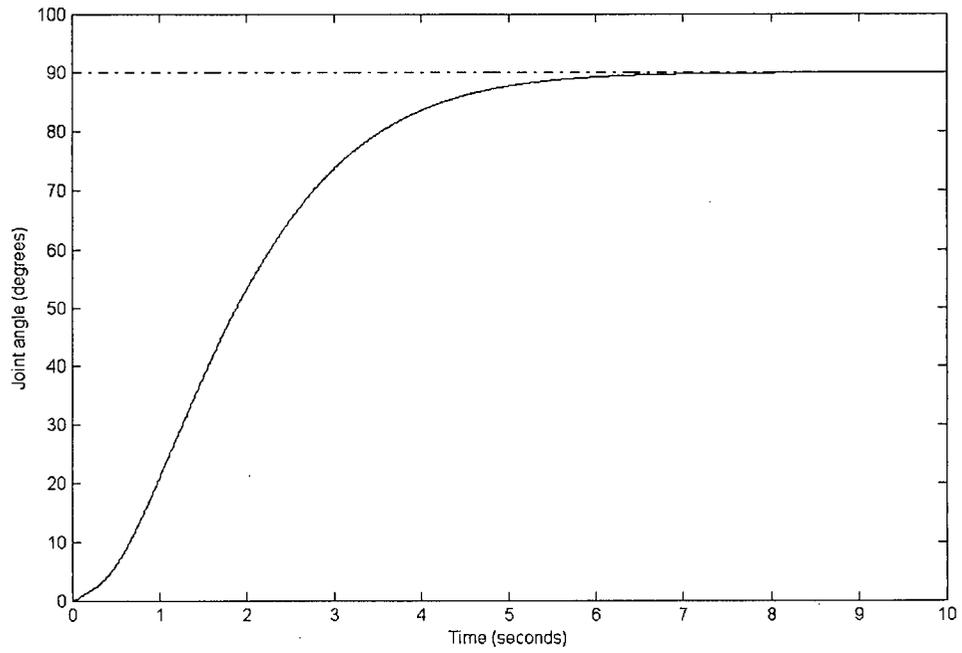


Figure 4.33 Case 3: Joint angle responses with IMPC.

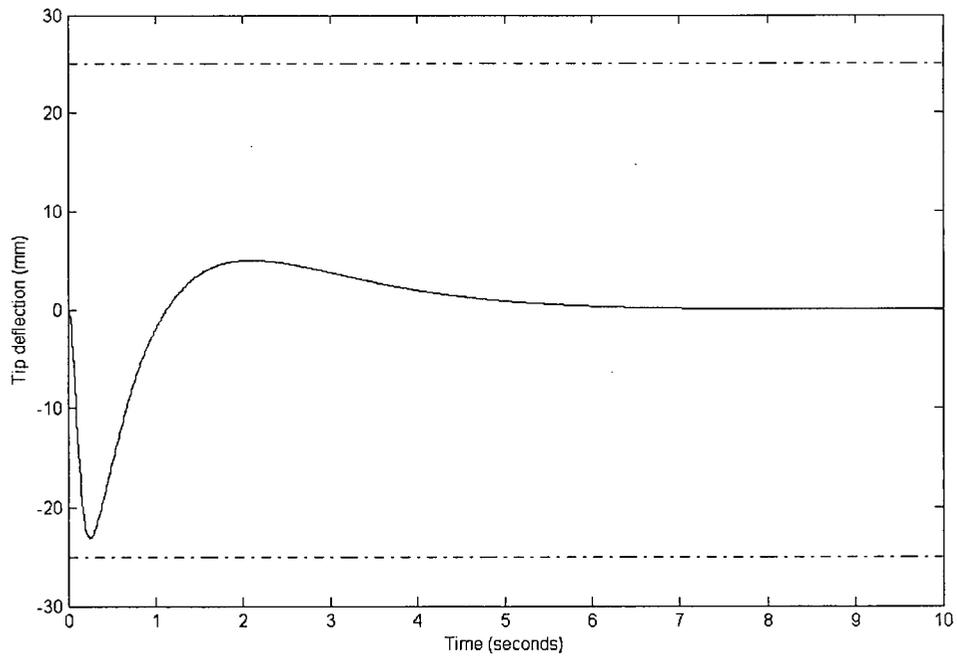


Figure 4.34 Case 3: Link tip deflection responses with IMPC.

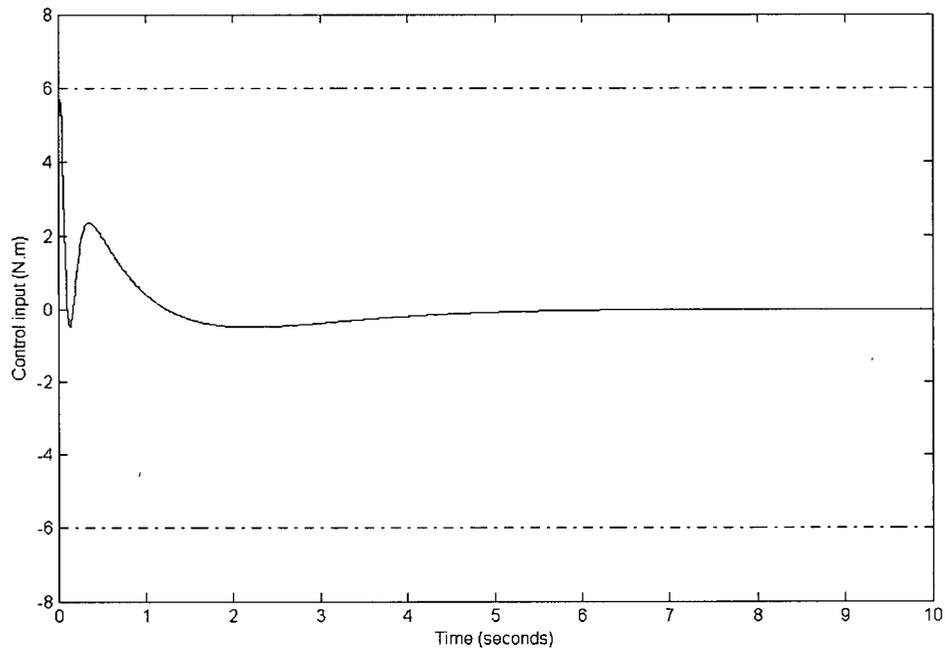


Figure 4.35 Case 3: Control input with IMPC.

4.6 Summary

In this Chapter we designed an intelligent model predictive control (IMPC) scheme for the motion control of a single-link flexible robotic manipulator. A system model that is suitable for the controller design and simulation was developed. The simulation results indicated that the fundamental mode alone was able to quite accurately capture the flexible dynamics of the system response. The nonlinearity of the single-flexible link manipulator was found to be insignificant and could be neglected when link deflection and the velocity terms were not very large. The linear model with the first flexible mode model was used in the MPC controller design.

The effect of the payload on system dynamics was found to be significant and could not be neglected. A system identification module was designed to identify the unknown system payload. The fuzzy rule base and the fuzzy tuner were developed to tune the parameters of the MPC controller.

The performance of the IMPC scheme was evaluated through computer simulations. The

simulation results showed that the IMPC control technique was quite effective in auto-tuning of the MPC controller parameters for achieving the desired system performance and for manage the input and output constraints in the motion control of a single-link flexible robotic manipulator.

Chapter 5

Control of a Two-link Flexible Manipulator

In this chapter the intelligent model predictive control (IMPC) algorithm developed in Chapter 3 is applied for the motion control of a two-link flexible robotic manipulator. The first link of the manipulator is a 1 m long aluminum rod, and the second link a 1 m long thin aluminum beam. Its physical parameters are as given in Table 2.1. The internal dynamic model design of the model predictive controller (MPC) is presented in Section 5.1. The IMPC scheme for the system is presented in Section 5.2. The performance of the IMPC scheme for the two-link system is evaluated using computer simulations in Section 5.3.

5.1 Internal Model

In Section 2.33 we have developed the dynamic equations for a two-link flexible robotic manipulator using the assumed mode method. The dynamic equations are given in Appendix A.2. These equations are highly nonlinear and incorporate a finite number of flexible modes. In the present section we develop a linear internal model for the model predictive controller (MPC).

First we investigate how many flexible modes are needed to adequately represent the flexible dynamics of the two-link robotic manipulator. From Table 2.2 and Table 2.3, the natural frequencies of the first four modes of link 1 and link 2 with nominal payload, are given by

$$\text{Link 1: } f_{11} = 1.9397 \text{ Hz, } f_{12} = 7.2546 \text{ Hz, } f_{13} = 94.2847 \text{ Hz, } f_{14} = 257.3687 \text{ Hz}$$

$$\text{Link 2: } f_{21} = 1.1442 \text{ Hz, } f_{22} = 11.1224 \text{ Hz, } f_{23} = 31.1435 \text{ Hz, } f_{24} = 56.1767 \text{ Hz}$$

For both links, the natural frequency of the third mode is much larger than those of the first two modes. Consequently, it is reasonable to use only the first two flexible modes in the internal model. The natural frequency of the second flexible mode for link 1 is low ($f_{12} = 7.2546 \text{ Hz}$), and it is likely to be excited during the robot motion. We will verify this in

Section 5.3 using computer simulations. Accordingly, the internal model of MPC will include the first two flexible modes.

Next, we investigate the nonlinearity of the system model. From the nonlinear dynamic equations of the system as given in Appendix A.2, it is seen that the nonlinearity of the system comes from the following terms in the dynamic equations: Coriolis and centrifugal terms, higher order terms of link deflection, and 'sin' and 'cos' terms of θ_2 . In Chapter 2 we have assumed that the link deflection is small, in the development of the nonlinear dynamic equations of the system. This implies that the higher order terms involving products of deformations can be neglected. Accordingly, if the velocity terms are not very large, the equilibrium point of the system may be taken as $\mathbf{x}_0 = [0 \ \theta_2 \ \mathbf{0}]^T$. In this case the nonlinearity of the system is mainly due to the movement of link 2. The nonlinearity of the system is significant and cannot be neglected when the movement of link 2 is large. We will verify this in Section 5.3 using computer simulations. In order to deal with the nonlinearity of the system, we will re-linearize the system at time intervals of one second.

In summary, the MPC internal model is a piecewise linear model (linearized every second) incorporating the first two flexible modes. The piecewise linear model can be obtained using the techniques presented in Section 2.4.

5.2 IMPC Controller Design

The process of IMPC controller design for the two-link flexible robotic manipulator is similar to that for the single-link case as presented in Chapter 4, and its details are not repeated here. In this section we will outline the major aspects of the controller design process for the two-link problem.

The physical parameters of the outboard link of the two-link flexible robotic manipulator are the same as those for the single-flexible manipulator presented in Chapter 4. The system identification module for the two-link system is also the same as that for the one-link case presented there. In the system identification stage, we lock joint 1 and only move link 2 to identify the unknown payload.

The discrete-time state-space internal model of the MPC controller is given by

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k)\end{aligned}\quad (5.1)$$

where

$$\mathbf{x}(k) = [\theta_1(k), \theta_2(k), \delta_{11}(k), \delta_{12}(k), \delta_{21}(k), \delta_{22}(k), \dot{\theta}_1(k), \dot{\theta}_2(k), \dot{\delta}_{11}(k), \dot{\delta}_{12}(k), \dot{\delta}_{21}(k), \dot{\delta}_{22}(k)]^T$$

is the state vector, $\mathbf{u}(k) = [u_1(k), u_2(k)]^T$ is the input vector, and

$\mathbf{y}(k) = [\theta_1(k), \theta_2(k), w_{ip_{-1}}(k), w_{ip_{-2}}(k)]^T$ is the output vector. The system inputs $u_1(k)$ and

$u_2(k)$ are the joint torques. The outputs are the joint angles $\theta_1(k)$, $\theta_2(k)$ and the link deflections $w_{ip_{-1}}(k)$, $w_{ip_{-2}}(k)$ at the ends of the links.

The cost function of the MPC controller is

$$\begin{aligned}J &= \sum_{i=1}^{H_p} Q_{yw_{-1}} \left\{ [\hat{y}_1(k+i|k) - r_{\theta_1}]^2 + Q_{w_{-1}} [\hat{y}_3(k+i|k)]^2 \right\} \\ &+ \sum_{i=1}^{H_p} Q_{yw_{-2}} \left\{ [\hat{y}_2(k+i|k) - r_{\theta_2}]^2 + Q_{w_{-2}} [\hat{y}_4(k+i|k)]^2 \right\} \\ &+ \sum_{i=1}^{H_u} [R_{u_{-1}} \Delta \hat{u}_1(k+i-1|k)^2 + R_{u_{-2}} \Delta \hat{u}_2(k+i-1|k)^2]\end{aligned}\quad (5.2)$$

where r_{θ_1} and r_{θ_2} are the setpoints for the joint angles. The setpoints for the link deflections are zero. The input and output constraints are taken as

$$\begin{aligned}-6 \text{ N.m} &\leq u_1(k+i|k) \leq 6 \text{ N.m} \\ -6 \text{ N.m} &\leq u_2(k+i|k) \leq 6 \text{ N.m} \\ -5 \text{ mm} &\leq y_3(k+i|k) \leq 5 \text{ mm} \\ -20 \text{ mm} &\leq y_4(k+i|k) \leq 20 \text{ mm}\end{aligned}\quad (5.3)$$

Note that the maximum allowed tip deflection of link 1 is smaller than that of link 2. The reason for this is that the link vibration in link 1 has a significant effect on link 2, and the third mode nature frequency of link 2 $f_{13} = 31.1435 \text{ Hz}$ may be excited if the link 1 vibration is excessive. Limiting the maximum allowable tip deflections can prevent the excitation of the higher system modes. A computationally efficient ‘anti-windup’ MPC controller is designed for the two-link flexible manipulator using the approach presented in Chapter 3.

The fuzzy inference system for the two-link manipulator uses the same membership

functions and the rule base as for the single-link manipulator presented in Chapter 4. Each link can be considered as a single flexible-link system, and its MPC parameters can be adjusted separately. The tuning actions for each link can be scaled to the proper level by adjusting the sensitivity parameters $H_{p_sen_1}$, $H_{p_sen_2}$, $Q_{w_sen_1}$, $Q_{w_sen_2}$, $R_{u_sen_1}$, and $R_{u_sen_2}$. The larger values of the prediction horizon tuning actions of the two links are used in the controller tuning; specifically

$$\Delta H_p = \max(\Delta H_{p_1}, \Delta H_{p_2}) \quad (5.4)$$

5.3 Computer Simulation of IMPC

The developed intelligent model predictive control (IMPC) scheme is now simulated for motion control of the two-link flexible robotic manipulator. The system payload is taken to be nominal. Figure 5.1 shows the Simulink simulation block diagram of the system. The plant is represented by a nonlinear dynamic model of the two-link flexible robotic manipulator incorporating the first two flexible modes. Initial set of MPC controller parameters is used to generate the system responses. The fuzzy tuner then adjusts the controller parameters until the desired system performance is achieved.

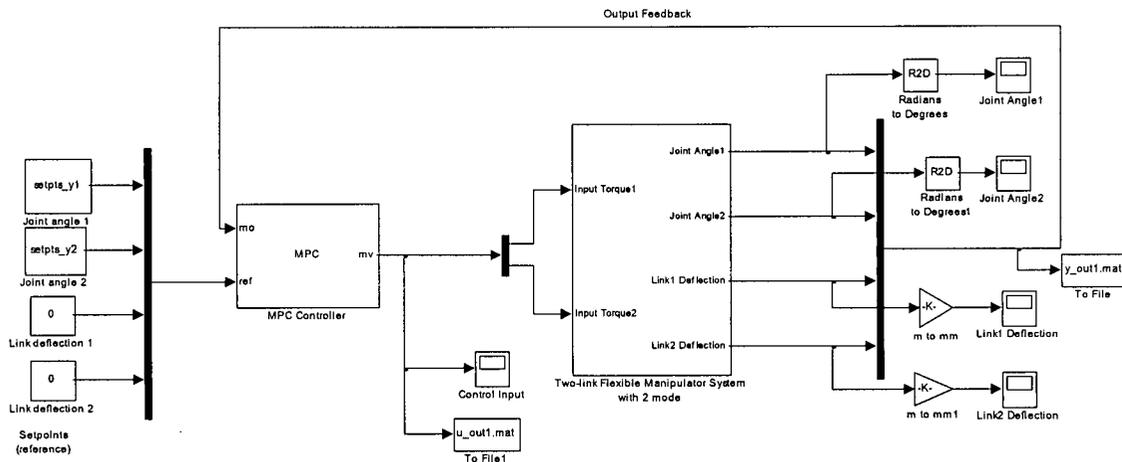


Figure 5.1 Simulation block diagram of the two-link flexible manipulator system.

First the internal model selection analysis is done as presented in Section 5.1. The Simulink simulation block diagram in Figure 5.1 is used to perform the simulations. Three different internal model MPC controllers are tested and their closed-loop system responses are compared. The first MPC controller uses a linear model linearized at the initial link positions, incorporating only the first flexible mode. The second MPC controller uses a linear model linearized at the initial link positions, and incorporating the first two flexible modes. The third MPC controller uses a piecewise linear model linearized at every second and incorporating the first two flexible modes. The joint angle setpoint is $\frac{\pi}{3}$ for both links. Figure 5.2 and Figure 5.3 show the system responses for these three internal model MPC controllers. From Figure 5.3 it is seen that if MPC uses a first-mode internal model, both links exhibit constant vibrations. Including the second mode in the internal model will reduce the link vibrations. Due to the nonlinearity of the system, however, the link deflections for the two-flexible-mode model remain large, and the joint 2 angle response is rather slow. Linearizing the system at every second improves the system performance. In view of this, the piecewise linear internal model (linearized every second) incorporating the first two flexible modes is used in the MPC controller.

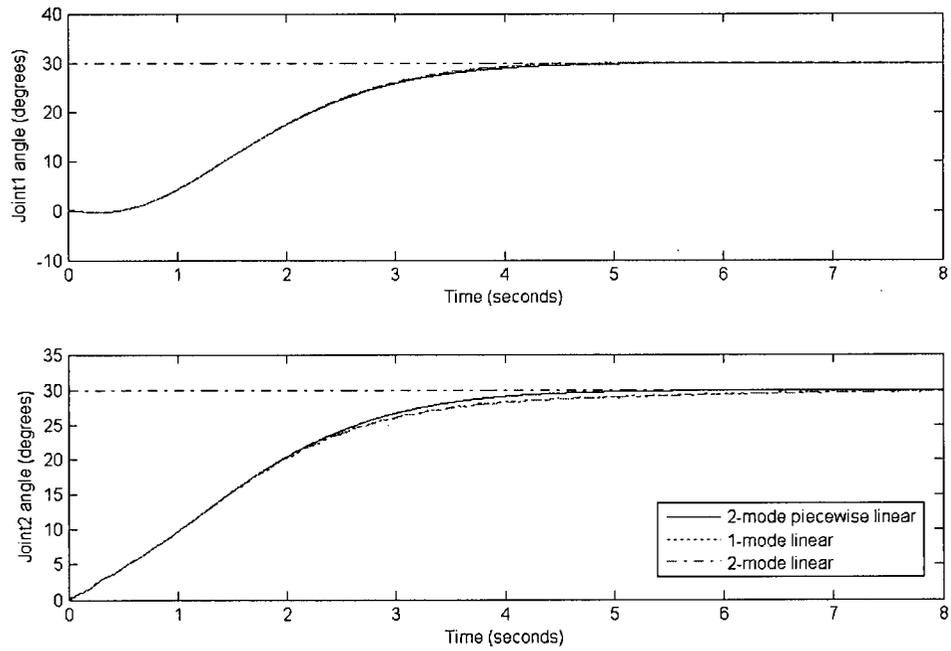


Figure 5.2 Joint angle response with three different internal models.

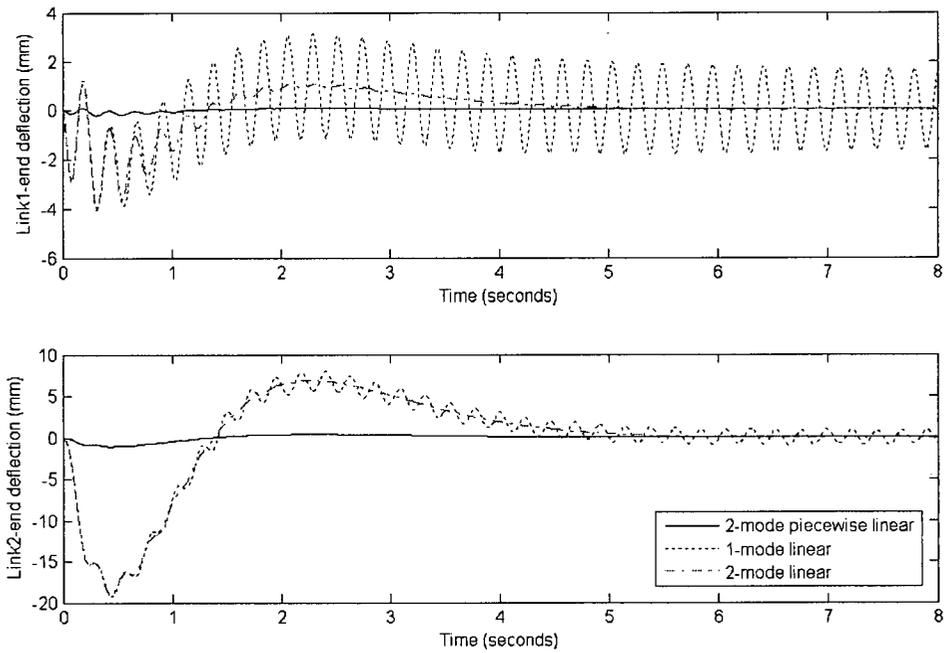


Figure 5.3 Link tip deflection responses with three different internal models.

Next, three different cases of motion configurations are tested in order to evaluate the performance of the IMPC algorithm. Table 5.1 gives the predetermined MPC controller parameters for both cases. The joint angle setpoints and the MPC tuning parameters for the three cases are given in Table 5.2.

Figure 5.4 through Figure 5.9 shows the system responses for Case 1. Here the initial, intermediate, and final system responses are shown. It is noted that the desired system performance cannot be achieved by using the initial set of MPC parameters. For both links, the joint angle response exhibits a large overshoot, and the link deflection and the control input exceed their limits. The IMPC tunes the controller using the fuzzy knowledge base until the desired system performance is achieved. The results show that the system responses using the parameters tuned by IMPC meet the design specifications. There are no joint angle overshoots, and the link deflections and the control inputs are within their limits.

The system responses for Case 2 and Case 3 with the IMPC scheme are shown in Figure 5.10 through Figure 5.15. The simulation results demonstrate that the IMPC control technique is quite effective in controlling the tip position of the two-link flexible robotic manipulator.

Table 5.1 Predetermined parameters for the MPC controller of the two-link manipulator.

Link 1	r_{w_1} (mm)	r_{u_1} (N.m)	$H_{p_sen_1}$	$Q_{w_sen_1}$	$R_{u_sen_1}$	H_u	Q_{yw_1}
	5	6	7.8	2	2.5	5	40
Link 2	r_{w_2} (mm)	r_{u_2} (N.m)	$H_{p_sen_2}$	$Q_{w_sen_2}$	$R_{u_sen_2}$		Q_{yw_2}
	20	6	7.8	3.2	1.05		40

Table 5.2 Tuning parameters for the MPC controller of the two-link manipulator.

Link 1	r_{θ_1}	r_{θ_2}	H_{p_ini}	H_{p_end}	$Q_{w_ini_1}$	$Q_{w_end_1}$	$R_{u_ini_1}$	$R_{u_end_1}$
Case 1	$\frac{\pi}{6}$	$\frac{\pi}{6}$	100	594	1	11.6	1	8.1
Case 2	$\frac{\pi}{6}$	$\frac{\pi}{3}$	100	845	5	25.7	5	22.4
Case 3	$\frac{\pi}{3}$	$\frac{\pi}{2}$	200	1100	10	34.1	10	36.1
Link 2					$Q_{w_ini_2}$	$Q_{w_end_2}$	$R_{u_ini_2}$	$R_{u_end_2}$
Case 1					1	14.6	1	8.0
Case 2					5	19.4	5	43.0
Case 3					10	37.4	10	53.1

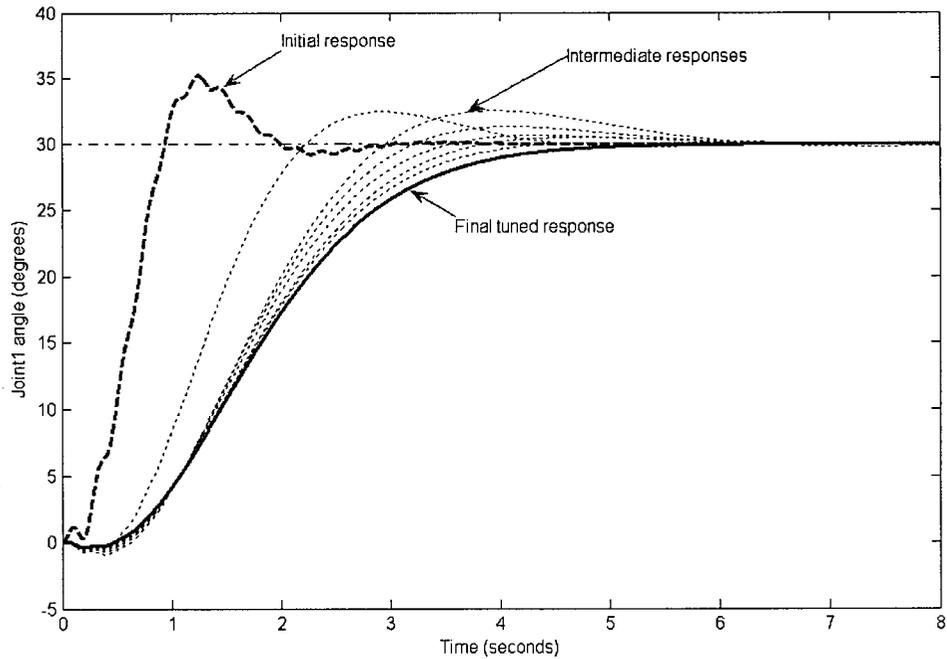


Figure 5.4 Case 1: Joint angle responses of link 1 with IMPC.

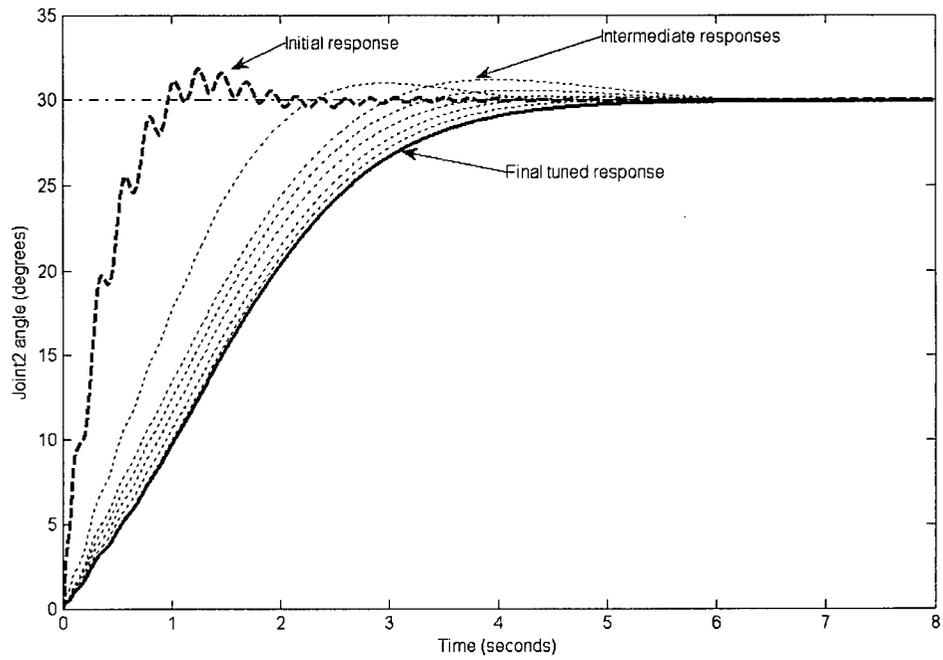


Figure 5.5 Case 1: Joint angle responses of link 2 with IMPC.

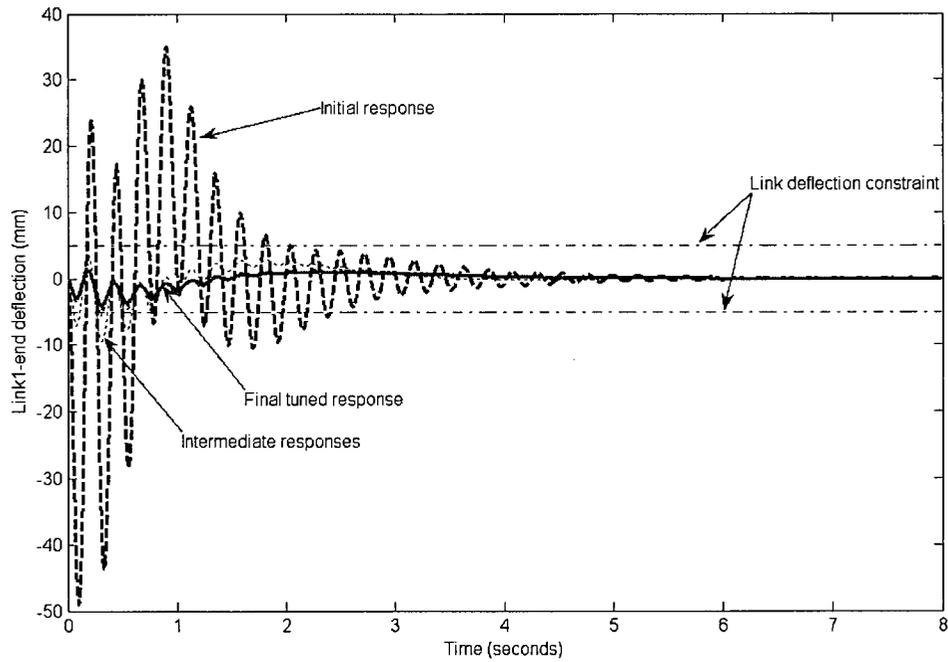


Figure 5.6 Case 1: Link tip deflection responses of link 1 with IMPC.

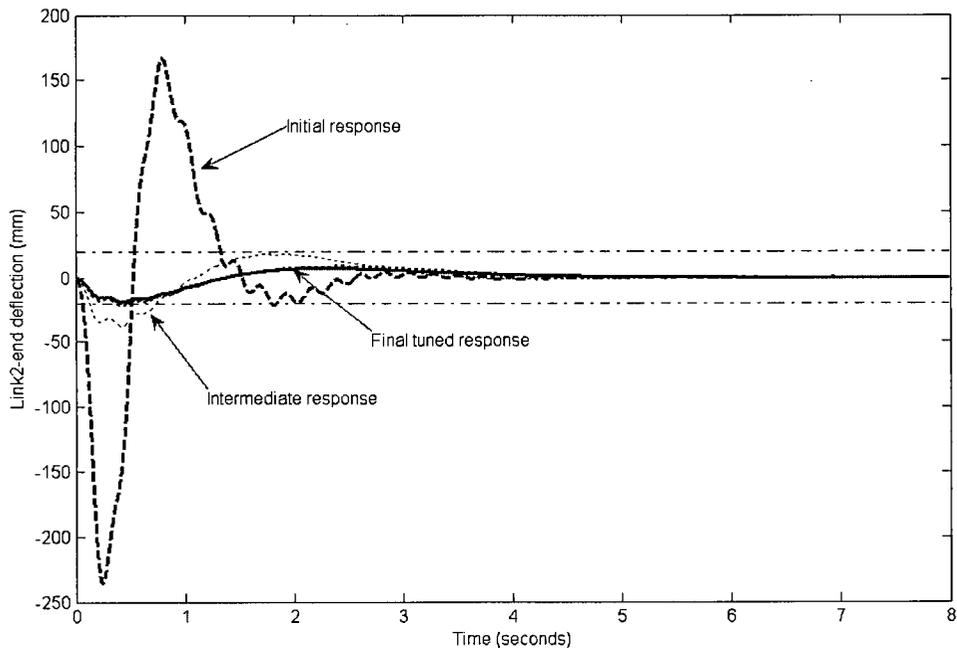


Figure 5.7 Case 1: Link tip deflection responses of link 2 with IMPC.

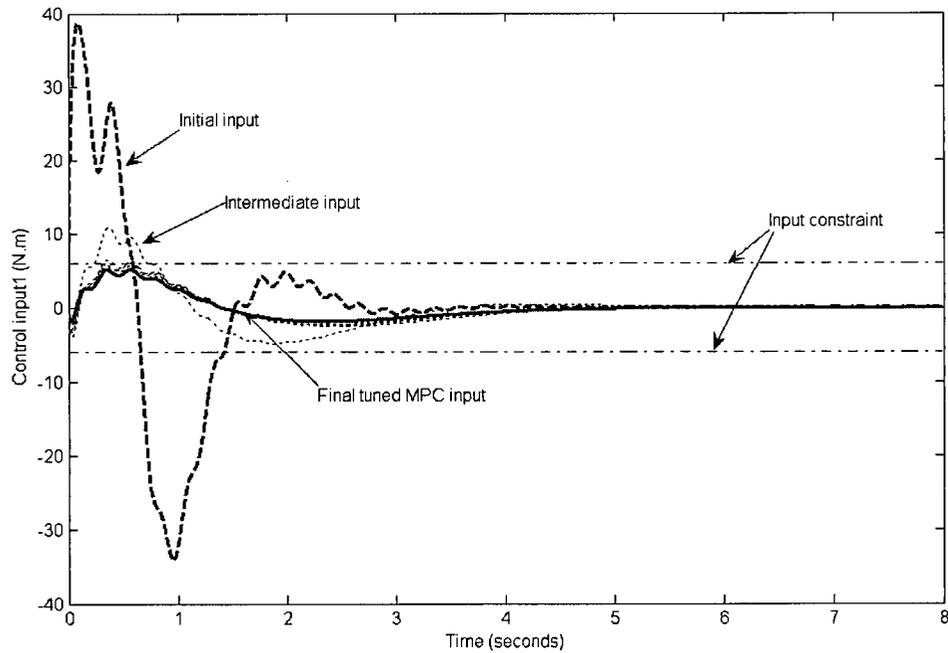


Figure 5.8 Case 1: Control input for link 1 with IMPC.

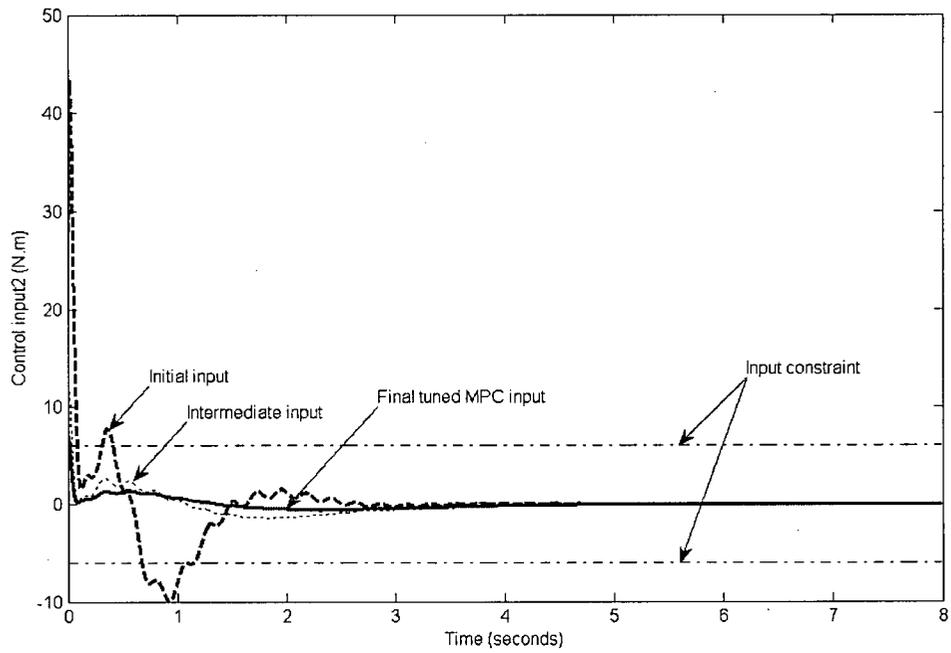


Figure 5.9 Case 1: Control input for link 2 with IMPC.

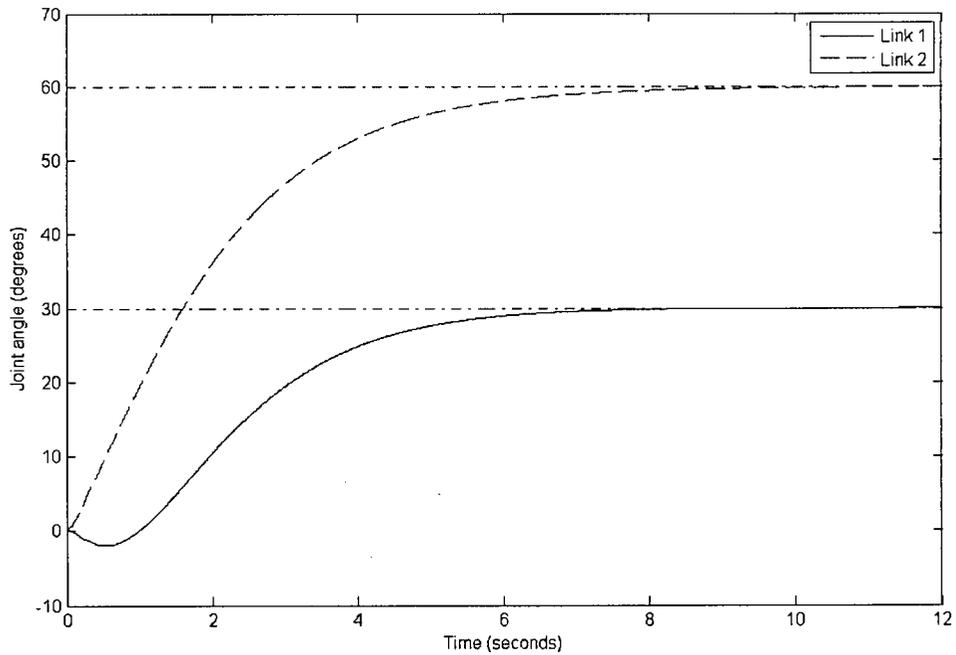


Figure 5.10 Case 2: Joint angle responses of link 1 and link 2 with IMPC.

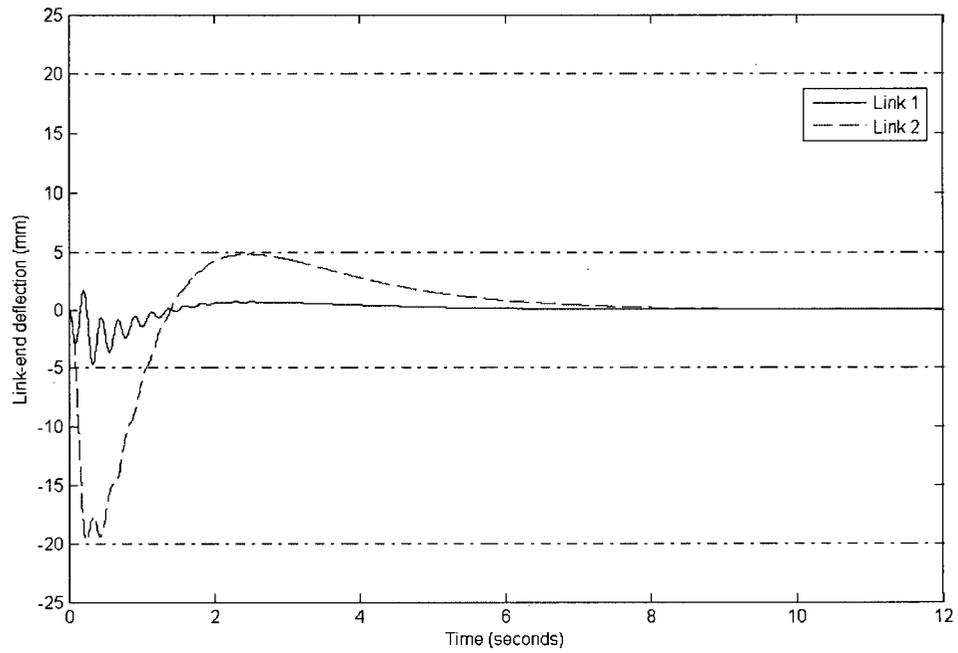


Figure 5.11 Case 2: Link tip deflection responses of link 1 and link 2 with IMPC.

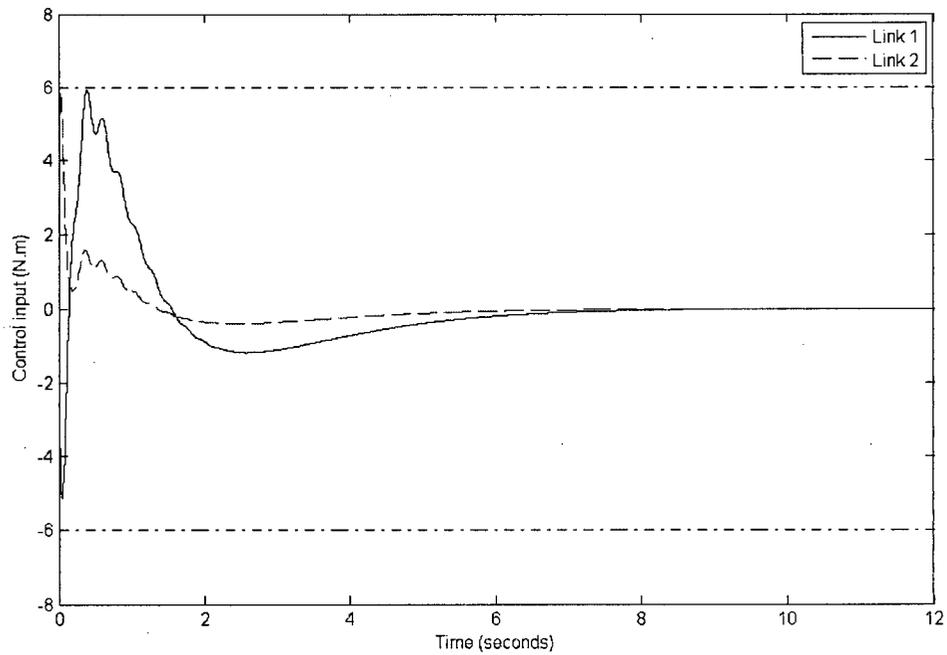


Figure 5.12 Case 2: Control inputs for link 1 and link 2 with IMPC.

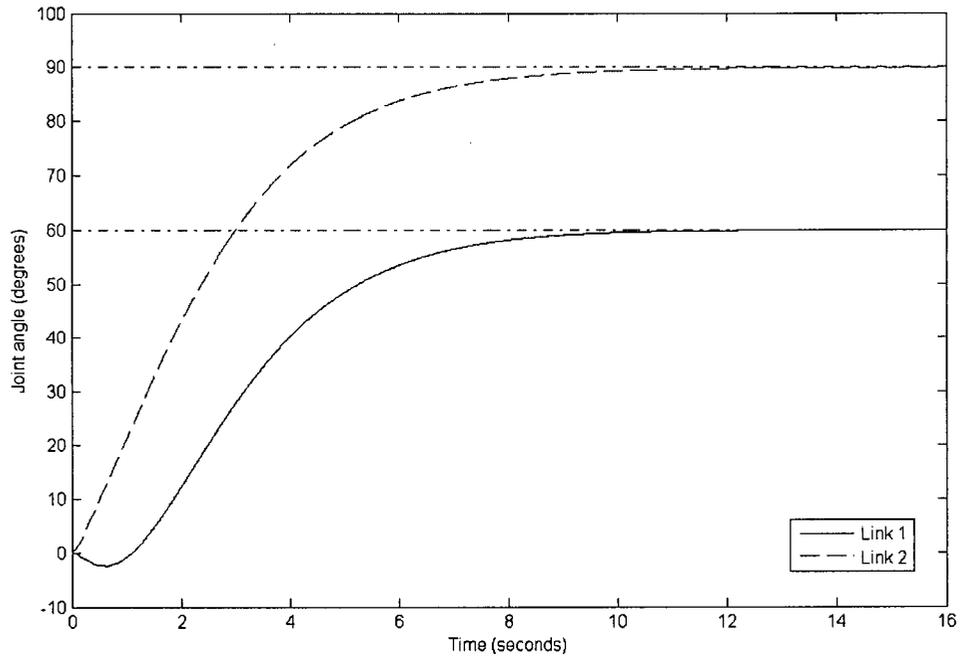


Figure 5.13 Case 3: Joint angle responses of link 1 and link 2 with IMPC.

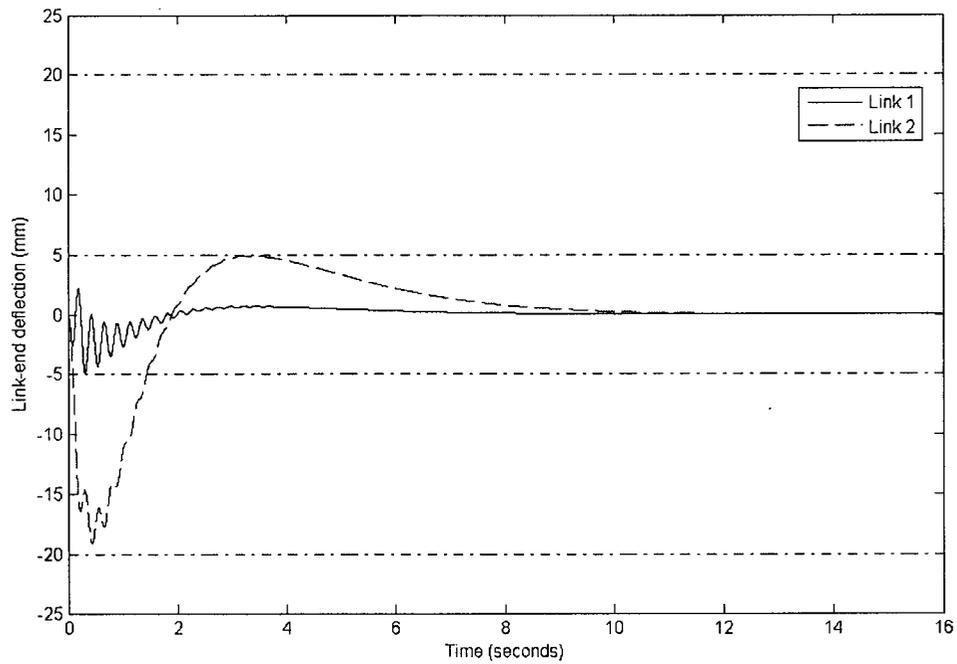


Figure 5.14 Case 3: Link tip deflection responses of link 1 and link 2 with IMPC.

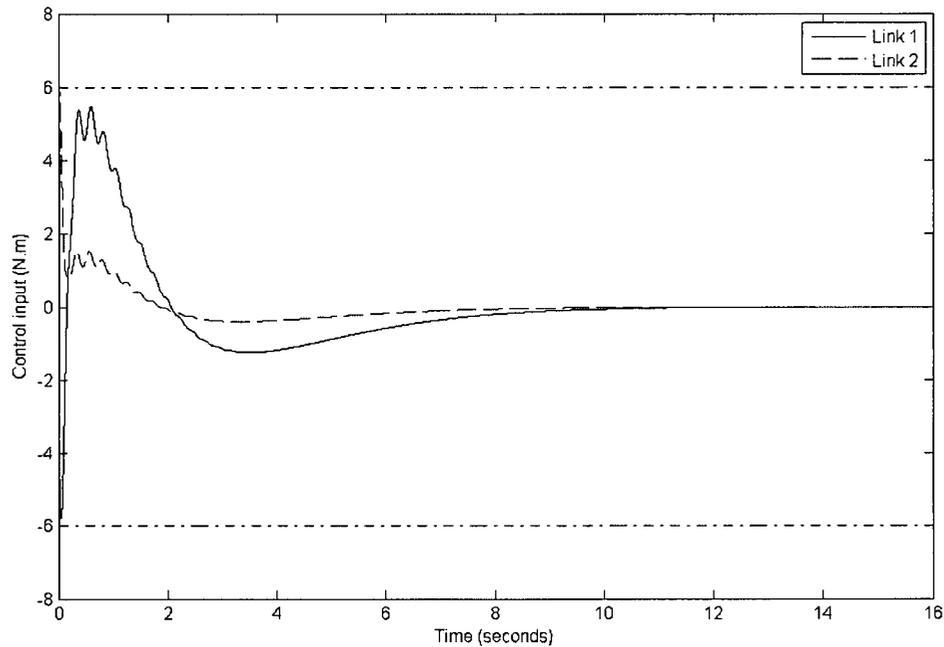


Figure 5.15 Case 3: Control inputs for link 1 and link 2 with IMPC.

5.4 Summary

In this Chapter we developed an intelligent model predictive control (IMPC) scheme for motion control of a two-link flexible robotic manipulator. The internal model for model predictive control (MPC) was developed. The simulation results showed that the first flexible mode alone was unable to accurately capture the flexible dynamics of the system. Also, the nonlinearity of the system was found to be significant and could not be neglected. Consequently, a piecewise linear internal model (linearized every second) and incorporating the first two flexible modes was used in the MPC. A fuzzy rule base and a fuzzy tuner were developed to tune the MPC parameters. The performance of the IMPC scheme was evaluated through computer simulations. The simulation results showed that the IMPC control technique was quite effective in auto-tuning the MPC parameters, in achieving the desired system performance and in managing the input and output constraints in motion control of the two-link flexible robotic manipulator.

Chapter 6

Experiments with Flexible-link Manipulator

In this chapter we focus on physical experimentation with a flexible-link manipulator. A prototype flexible-link robot manipulator system has been designed and developed in our laboratory at the University of British Columbia. It is suitable for validating and investigating the performance of various motion/force control strategies and algorithms which have been developed in the present research. In Section 6.1, the experimental test bed is described. The system components of the experimental apparatus, and the software design and operation of the overall system are presented there. In Section 6.2, implementation of the developed intelligent model predictive control (IMPC) scheme on the test-bed is described. Experiments using the prototype robot are outlined and experimental results are presented. The performance of the IMPC scheme is evaluated using the experimental results.

6.1 The Experimental Test-Bed

In order to investigate the dynamics and control of flexible manipulators, the prototype flexible link manipulator system (FLMS) shown in Figure 6.1 has been designed and developed in our laboratory. A realistic model of this prototype robotic system has been used in the simulation studies carried out in the present work. The prototype FLMS forms the test-bed for experimental studies of the IMPC scheme developed in the present work.

The prototype manipulator is composed of two revolute joints having a vertical axis of rotation so that they can move in a plane that is perpendicular to the field of gravity. A relatively long link is attached to each of these revolute joints. Two types of links are available, one being more flexible than the other. The detailed assembly drawings of joint 1 and joint 2 are found in Appendix B.1. Four rolling ball transfers are attached to the bottom of the second joint. The rolling support is important as it can reduce the axial loading of the motor shaft and provide a mobile base so that link 2 is free to move with respect to joint 1.

Without the rolling support, the lengthy link and the heavy weight of the motor will introduce a permanent bending stress on the motor shaft of joint 1, which may eventually damage the motor.

The links of the experimental manipulator can be interchanged. This allows one to investigate the dynamic characteristics of the robot and test the performance of the developed motion/force control algorithms, for different physical parameters. As mentioned in Chapter 2, two types of manipulator links are used in this research. Type 1 link is a 1 m long, 0.003 m thick, and 0.051 m high aluminum thin beam. Type 2 link is an aluminum rod 2.0 cm in diameter and 1.0 m in length. In addition to the flexible links, all other machined parts such as motor mounts, joint supports and shaft couplings are made of Aluminum 6061 due to its light weight (density $\rho_{Al} = 2710 \text{ kg/m}^3$), high strength (yield strength = 255 MPa) and ease of machining. The numerical values for the physical parameters of the manipulator are given in Table 2.1.

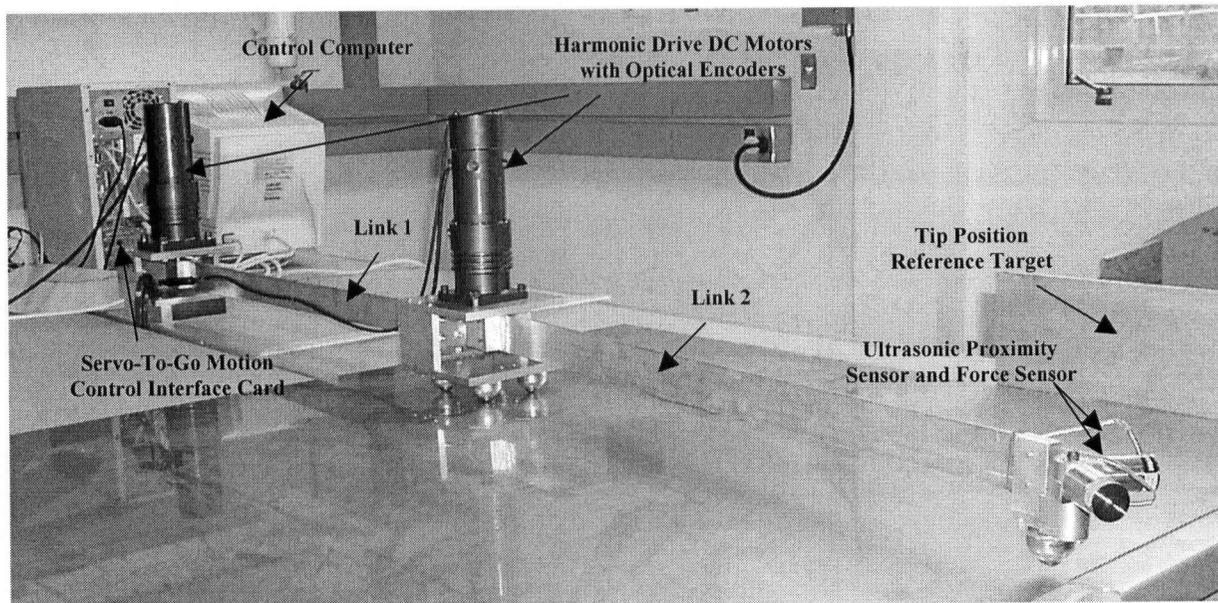


Figure 6.1 The prototype flexible link manipulator system (FLMS).

A PC-based control system is used as it allows different control strategies to be implemented easily using software, and modifications to the control algorithms can be made conveniently in the investigation. The electrical system of the FLMS manipulator may be

classified into three main subsystems: the actuation subsystem, the measurement subsystem and the data acquisition subsystem. The major components of each of these subsystems are summarized in Table 6.1. The schematic diagram of the overall electrical system is shown in Figure 6.2. The detailed electrical wiring diagram of the system can be found in Appendix B.2. We describe these subsystems in the subsequent sections.

Table 6.1 FLMS system components.

System components	Manufacturer	Model No.
Actuation subsystem:		
<ul style="list-style-type: none"> • 2 Harmonic-drive motors with optical encoders 	HD Systems	RHS-20-3012-E050DO RHS-32-3018-E050DO
<ul style="list-style-type: none"> • 2 PWM amplifiers 	Advanced Motion Control	12A8 25A8
<ul style="list-style-type: none"> • 1 DC Power supply 	Advanced Motion Control	PS16L72
Measurement subsystem:		
<ul style="list-style-type: none"> • 1 Bridge type loadcell 	Futek	L1605 (25lb)
<ul style="list-style-type: none"> • 1 Signal conditioner 	Transducer Techniques	TMO-1
<ul style="list-style-type: none"> • 1 Power adapter 	Transducer Techniques	APD-12VDC
<ul style="list-style-type: none"> • 2 Optical encoders 	HD Systems	E050DO
<ul style="list-style-type: none"> • 1 Analogue ultrasonic proximity sensor 	Honeywell	PK104015-11
Data acquisition subsystem:		
<ul style="list-style-type: none"> • 1 Motion control interface card (ISA bus) 	Servo-To-Go	STGII-8
<ul style="list-style-type: none"> • 2 50-pin screw terminal blocks 	National Instruments	CB-50LP
<ul style="list-style-type: none"> • 1 Pentium III personal computer 	LCF Advanced Technology	
<ul style="list-style-type: none"> • Operating system 	Microsoft	Windows NT 4.0 Workstation
<ul style="list-style-type: none"> • C++ software 	Microsoft	Visual C++ 6.0

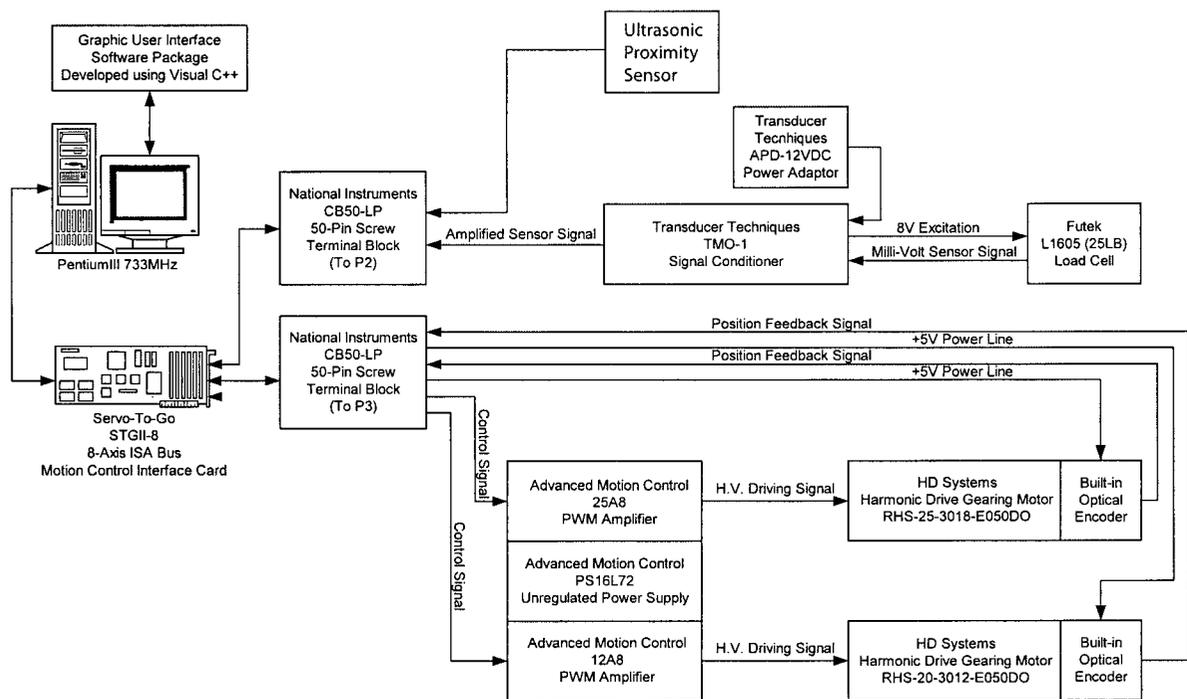


Figure 6.2 Schematic diagram of the electrical system of the manipulator.

6.1.1 Actuation Subsystem

The actuation subsystem includes two DC servomotors, two matched pulse-width modulation (PWM) amplifiers and a DC power supply.

A DC servomotor is located at each joint of the manipulator. These motors are used to control the motions and contact forces of the manipulator. DC motors are used as the actuators in this experimental setup due to their high precision, easy control, low power, and high-speed capabilities. Two motors with harmonic drive gearing, RHS-20-3012-E050DO and RHS-32-3018-E050DO, from HD system Inc., are chosen for our application. The harmonic drive gearing provides zero backlash (de Silva, 1989 and 2007), high positional accuracy and stiffness, and low torque/mass ratio. This makes them particularly suitable for applications that call for precise motion control. The motor unit comes in a compact package consisting of a servomotor, a harmonic drive gear-head, and an optical encoder for motion sensing (de Silva, 1989 and 2007). Their specifications are summarized in Table 6.2.

A power amplifier is required to translate the low energy control signals from the

controller into high-energy signals (e.g., motor voltage or current) in order to drive the DC motor. Two PWM amplifiers, 12A8 and 25A8, from Advanced Motion Control are selected to drive the HD harmonic-drive motors. The specifications of the PWM amplifiers are summarized in Table 6.3.

A DC power supply is required to provide the energy for the amplifiers. It can be an unregulated one because the PWM amplifiers can compensate for the power supply output variations and the AC ripples. A single unregulated DC power supply, PS16L72, from Advanced Motion Control is selected to complement the PWM amplifiers. The specifications of the power supply are summarized in Table 6.4.

Table 6.2 Specifications of the harmonic-drive motors.

Model Number	RHS-20-3012-E050DO	RHS-25-3018-E050DO
Manufacturer	HD Systems	HD Systems
Rated voltage	75 V	75 V
Rated current	2.7 A	3.9 A
Peak current	5.0 A	8.3 A
Rated output torque	30 N.m	60 N.m
Rated output speed	3000 rpm	3000 rpm
Max. continuous stall torque	43 N.m	72 N.m
Peak torque	84 N.m	160 N.m
Torque constant	21.0 N.m/A	22.9 N.m/A
Actuator accuracy	1 arc-minute	1 arc-minute
Gear ratio	100:1	100:1
Encoder resolution	500 ppr	500 ppr
Weight	4.2 kg	6.4 kg

Table 6.3 Specifications of the PWM amplifiers.

Model Number	12A8	25A8
Manufacturer	Advanced Motion Control	Advanced Motion Control
DC Supply Voltage	20-80 V	20-80 V
Peak Current (2 sec. max.)	±12 A	±25 A
Max. Continuous Current	±6 A	±12.5 A
Switching Frequency	33 kHz	33 kHz
Bandwidth	2.5 kHz	2.5 kHz

Table 6.4 Specifications of the DC power supply.

Model Number	PS16L72
Manufacturer	Advanced Motion Control
Input Voltage	120VAC
Output Voltage	72VDC
Nominal Output Current	10A

6.1.2 Measurement Subsystem

The measurement subsystem consists of feedback elements including a force sensor, a signal conditioner and its associated power adapter, two optical encoders, and an analog ultrasonic proximity sensor. The force sensor is for detecting the contact force at the end-effector. The signal conditioner is for impedance matching, signal amplification and noise filtering. There is an optical encoder at each joint, which gives the position measurements from the manipulator joints for feedback into the robot controller. An ultrasonic proximity sensor is used for measuring the link tip position.

A stain gage load cell (force sensor) is located at the end of the second link to measure the contact force. A stain gage load cell is a transducer which senses the load/force acting on

using strain gage elements, and converts the measurement into an analog electrical signal. This conversion is achieved by the physical deformation of the strain gages that are wired in a wheatstone bridge configuration. In our application, a compression type force sensor is required in order to measure the contact force at the tip of the end-effector. The load cell L1605 from Futek is selected for our application because it is thread mounted, which simplifies the design of the end-effector. Its specifications are summarized in Table 6.5.

A force sensor usually has high output impedance and produces a low output signal in the millivolt range making further processing difficult. In view of this, a signal conditioner is connected between the force sensor and the data acquisition system. It has a low output impedance and a high input impedance (de Silva, 1989 and 2007). As a result it consumes little power from the force sensor and avoids electrical loading errors. This is referred to as impedance matching. It also consists of a high gain amplifier which can step up the signal level to a useful range. The signal conditioner TMO-1 from Transducer Techniques is selected due to its low cost and dedicated conditioning features. The signal conditioner plus its associated power adaptor APD-12VDC are required for the force sensor. The specifications of this unit are given in Table 6.6.

Table 6.5 Specifications of the load cell.

Model Number	L1605
Manufacturer	Futek
Force range (Compression)	25 lb
Excitation	10VDC, 18VDC max
Bridge resistance	350 Ω
Rated output (RO)	2 mV/V
Non-linearity	$\pm 0.2\%$ to 0.5% RO
Weight	0.16 kg

Table 6.6 Specifications of the force sensor signal conditioner.

Model Number	TMO-1
Manufacturer	Transducer Techniques
Gain range	75 to 1000
Input sensitivity	1 mV/V min. for 8 VDC output
Output voltage	0 to ± 8 VDC
Accuracy	$\pm 0.05\%$ of FS
Frequency response	DC to 220 Hz
Excitation voltage	8 VDC ± 0.25 V
Power required	12 VDC
Sensor resistance	120 Ω min., 1000 Ω max.

The optical encoder E050DO from HD Systems is used to measure the actual angular position of the motor shaft and hence the rotation angle of the joint. The encoder comes assembled with the harmonic-drive motor; hence the mounting problems are avoided.

The optical encoder is composed of light emitting diodes (LED), a code wheel and detectors (de Silva, 1989 and 2007). As the code wheel rotates, the light beam is interrupted and two pulsating signals are generated. The angular position of the shaft is measured by counting the number of pulse while the direction of rotation is indicated by the relative phase angle between the two output signals. Usually, there is one additional output channel Z that gives an index pulse corresponding to one full shaft revolution.

As the optical encoder is a digital transducer, analog-to-digital (A/D) conversion is not required and the quantization error does not exist. The chosen encoder E050DO has a resolution of 500 PPR (pulse/rev.), with a gear reduction ratio of 100:1. The smallest joint angle that it can measure is

$$\theta_{\min} = \frac{360^\circ}{\gamma \times 500 \times 100} = \frac{360^\circ}{4 \times 500 \times 100} = (1.8 \times 10^{-3})^\circ \quad (6.1)$$

where $\gamma = 4$ is the encoder multiplier.

An analog ultrasonic proximity sensor from Honeywell is used for measuring the link tip position. It is necessary in evaluating the performance of the developed motion control algorithms, which is performed by measuring the link-end vibration when it reaches the final desired position. Table 6.7 lists the relevant parameters of the ultrasonic device.

Table 6.7 Parameters of the ultrasonic sensor.

Model Number	PK104015-11
Manufacturer	Honeywell
Type	Analog ultrasonic proximity sensor
Range	10 cm to 75 cm
Input power	10-30 VDC, 50 mA
Output	0-5 VDC
Response speed	50 ms
Bandwidth	20 Hz
Repeat accuracy	1% of full scale
Resolution	1 mm

6.1.3 Data Acquisition Subsystem

The data acquisition subsystem consists of a motion control interface card, two 50-pin screw terminal blocks, and one personal computer. A motion control interface card is needed to manage the data transfer between different hardware components and the computer. Motion control interface STGII-8 from Servo-To-Go is selected and used in the system. Its specifications are summarized in Table 6.8. The hardware installation guide of this card can be found in Appendix C. This card is a PC-based motion control card. When compared to a DSP based card, it offers more flexibility to the user, to experiment with different control algorithms. It is a general purpose, motion control input/output board which can control up to

eight motors simultaneously from an ISA-bus based computer such as IBM compatible PC. This board is simply and efficiently accessed by the set registers located in the I/O space of the PC.

The card consists of four 50-pin connectors and one 2-pin connector for connection to the external hardware devices. The connector designations are as follows:

- P1: 24 bits of opto-22 compatible digital I/O.
- P2: 8 channels of analog input, 8 bits of user I/O, and 8 motor direction bits.
- P3: encoder input and analog output for 1-4 axes.
- P4: encoder input and analog output for 5-8 axes.
- P5: battery backup input.

In our application, only two (P2 and P3) out of the five connectors are used to acquire the encoder inputs, the force sensor input, the ultrasonic sensor input and the control signal outputs. Two 50-pin CB-50LP screw terminal blocks from National Instruments are used to make all the connections to the encoders, force sensor, ultrasonic sensor and the PWM amplifiers.

Because optical encoder is a digital transducer, analog to digital (A/D) conversion is not required here. Two optical encoders are connected to the encoder input port of the motion control card, and the joint position feedback signals are transmitted to the computer. With an encoder multiplier of 4, a gear reduction ratio of 100:1, and an encoder resolution of 500 PPR, the joint angle θ can be obtained as

$$\theta = i_{ENC} \frac{360^\circ}{4 \times 100 \times 500} \quad (6.2)$$

where i_{ENC} is the number of encoder counts.

The analog inputs can be configured as ± 5 V span to match the outputs of the selected signal conditioner of the force sensor and the ultrasonic sensor. With 13-bit resolution, the count to voltage constant p_{cv} is given by

$$p_{cv} = \frac{5}{4096} \quad (6.3)$$

The measurement signal of the force sensor is sent to the signal conditioner and the

output signal from signal conditioner is supplied to the motion control card. The motion control card then uses an analog to digital converter (ADC) to convert the analog signal into a digital signal. The resulting digital signal is then supplied to the computer. The force can be calculated according to

$$\begin{aligned} F &= i_{ADC} p_{cv} p_{vlb} \\ &= i_{ADC} \frac{5}{4096} \frac{25}{8} \end{aligned} \quad (6.4)$$

where i_{ADC} is the number of ADC counts and p_{vlb} is the voltage to lb constant.

The analog measurement signal from the ultrasonic sensor is supplied to the motion control card. The motion control card then uses ADC to convert the analog signal into a digital signal. The resulting digital signal is then supplied to the computer. The link-end position is calculated according to

$$y_{tip} = i_{ADC} p_{cv} p_{vcm} + y_c - y_r \quad (6.5)$$

where $p_{vcm} = 7.965 \text{ cm/V}$ is the centimeter to voltage constant, $y_c = 22.61 \text{ cm}$ is a constant value obtained from calibration of the sensor, and $y_r = 30 \text{ cm}$ is the distance between the final link tip position and the sensor reference surface.

The analog output voltage range of the card is $\pm 10 \text{ V}$ and the range of the 13bit digital to analog converter (DAC) is -4095 to 4096 . The control inputs calculated from the control algorithms are the joint torques (u_1, u_2) for each joint. Following equations are used to convert the joint torque inputs into digital control signals:

$$\begin{aligned} i_{DAC_1} &= 409.6 V_{in_1} + 0.5 = 409.6 \frac{u_1}{60} + 0.5 \\ i_{DAC_2} &= 409.6 V_{in_2} + 0.5 = 409.6 \frac{u_2}{30} + 0.5 \end{aligned} \quad (6.6)$$

where i_{DAC_1} and i_{DAC_2} are the DAC counts for joint 1 and joint 2, respectively. V_{in_1} and V_{in_2} are the analog output voltages. Their values can be calculated from the torque inputs as

$$V_{in_1} = \frac{u_1}{6}, \quad V_{in_2} = \frac{u_2}{3} \quad (6.7)$$

The digital control signals (i_{DAC_1}, i_{DAC_2}) from the computer are converted into analog signals (V_{in_1}, V_{in_2}) through the DAC of the motion control card. The resulting analog signals are then sent into the PWM Amplifier to generate the control torques (u_1, u_2) which drive the DC motors.

A personal computer is needed for processing the input signals and generating the control signals. A Pentium III 733 MHz PC with Windows NT 4.0 Workstation operating system is chosen from LCF. Microsoft Visual C++ 6.0 is installed for the control software design.

Table 6.8 Specifications of the motion control interface card.

Model Number	STGII-8
Manufacturer	Servo To Go
Encoder input	Up to 8 channels of encoder input A, B, I input 24bit counters Single-ended or differential (RS422 compatible) input signals
Analog input	Up to 8 channels of analog input 13 bit resolution Configurable as ± 10 V or ± 5 V spans
Analog output	Up to 8 channels of analog output $+10$ V to -10 V span 13 bit resolution
Digital input and output	32 bits, configurable in various input and output combinations Opto-22 compatible
Interval timers	Capable of interrupting the PC Timer interval is programmable to 10 minutes in 25 μ s increments
Battery backup input	Used to maintain encoder counting capability in the event of power failure

6.1.4 Software Design

A graphical user interface (GUI) software package is developed for the prototype flexible link manipulator system (FLMS) using Visual C++ in order to illustrate the satisfactory operation of the experimental setup and to test different motion/force control algorithms developed in the present research. The pull down menu of the FLMS is shown in Figure 6.3.

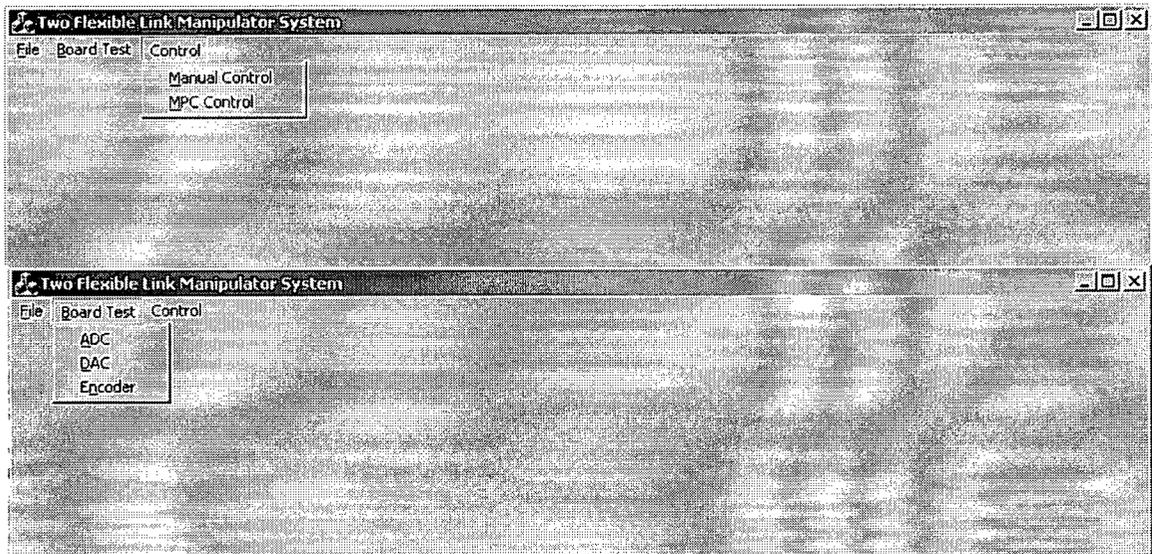


Figure 6.3 Pull down menu of the FLMS software package.

The 'Board Test' option is developed to verify the accuracy of the data acquisition of the motion control interface card. It consists of 'ADC,' 'DAC,' and 'Encoder' dialog boxes. By choosing 'ADC' from the pull down menu, an analog-to-digital dialog box as shown in Figure 6.4 appears. The dialog box displays the analog input voltage of all the 8 A/D conversion channels simultaneously. By selecting the radio button in the graphic display group box, the graph in the middle of the dialog box can keep tracks of the voltage in one of the A/D channels. The sampling rate is 100 Hz. The accuracy of the A/D conversion can be checked by connecting a variable DC power supply to each channel and then comparing the reading displayed on screen with the terminal voltage measured from the digital voltmeter.

The second board testing function available in the software package is the digital-to-analog conversion. By choosing 'DAC' from the pull down menu, the digital-to-analog dialog box as shown in Figure 6.5 pops up. The output voltages of all the 8 D/A channels can

be set on the screen by first picking the corresponding check boxes and then pressing the 'Send to Axis' button. The accuracy of the conversion can be verified by measuring the voltages at each D/A channel using a digital voltmeter.

The third board testing function is provided for testing the correctness of the encoder input. The corresponding dialog box is shown in Figure 6.6. To test the accuracy of the encoder readings, the up-down counters of the motion control interface card are first reset to zero by pressing the 'ZeroEncoderAll' button. Then the encoder is connected to one of the encoder channels via the screw terminal block. A low voltage reference signal (e.g. 0.5 VDC) is sent to the actuation subsystem so as to instruct the motor to rotate by a certain angle θ in degrees. The rotated angle θ should be equal to that calculated using equation (6.2) based on the encoder counts.

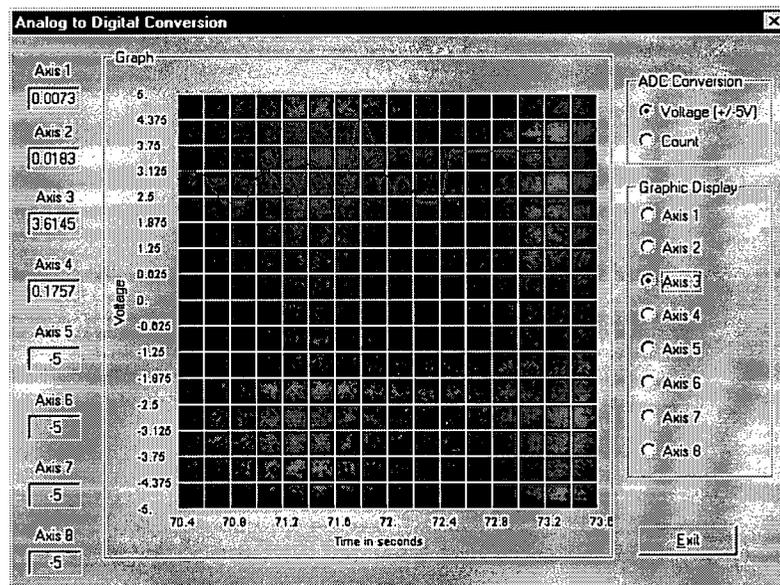


Figure 6.4 Analog-to-digital conversion dialog box.

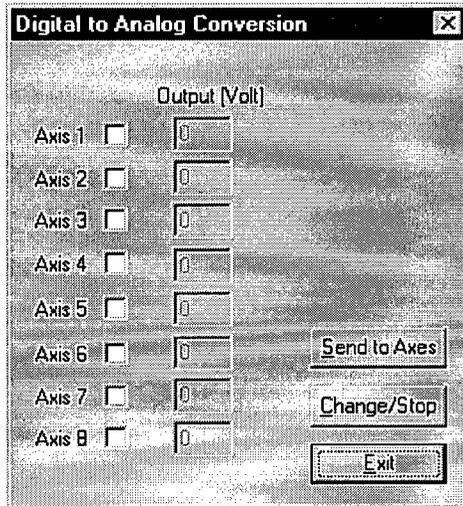


Figure 6.5 Digital-to-analog conversion dialog box.

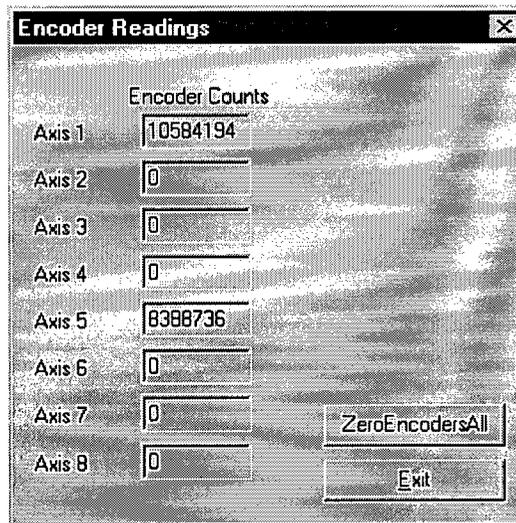


Figure 6.6 Encoder count dialog box.

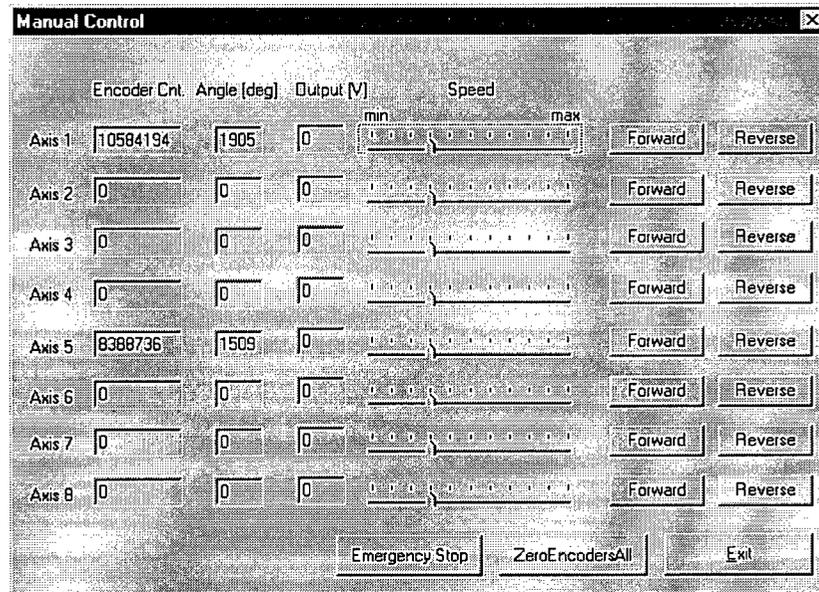


Figure 6.7 Manual control dialog box.

The 'Control' option on the pull down menu of the FLMS software package contains the 'Manual Control' and 'MPC Control' operations. The manual control dialog box shown in Figure 6.7 is used to test the motion operation of the manipulator. It consists of control buttons for all 8 axes in which both speed and direction of rotation can be controlled manually. The motion can be stopped by pressing the same button again so that it returns to its 'release' state or by pressing the 'Emergency Stop' button. The dialog box also displays the encoder count, its equivalent angle, and the output voltage for all 8 axes sampled at 100 Hz so that their accuracy can be verified.

The MPC motion control dialog box is developed to investigate the performance of the IMPC control strategy. The experimental results are presented in Section 6.2. The MPC motion control dialog box can be found from the pull down menu and is shown in Figure 6.8. The dialog box provides real-time plots of the system response and the control action. All input parameters are shown on the screen. Their definitions are given in Table 6.9. The MPC controller parameters can be determined by using the fuzzy tuner which has been developed in Chapter 4 based on the dynamic model of the system.

In addition, the output data can be stored in a file in text format by specifying the file name in the save dialog box as shown in Figure 6.9. The output data text file can be imported into Matlab for further data analysis and to plot the system responses.

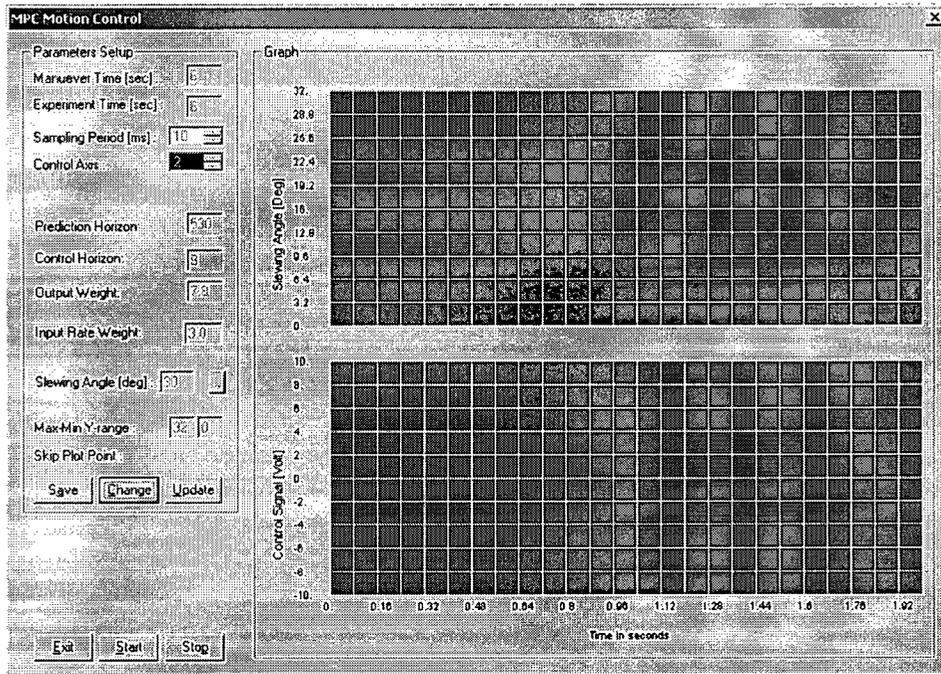


Figure 6.8 MPC motion control dialog box.

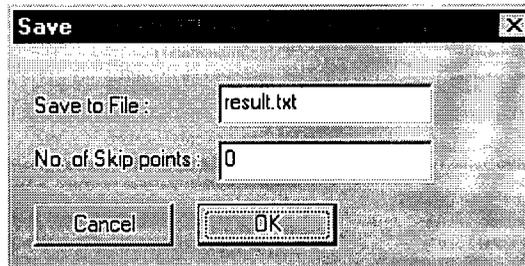


Figure 6.9 Save dialog box.

Table 6.9 Description of the input parameters of the MPC controller.

Input Parameters	Description
Maneuver Time (s)	The desired time taken for the manipulator to move to the specified slewing angle.
Experiment Time (s)	The total duration of closed loop control.
Sampling Period (ms)	The time period at which sensor signals are sampled and control action is generated.

Input Parameters	Description
Control Axis	Axes 1 through 8 can be selected for control.
Prediction Horizon	Prediction horizon H_p of the MPC controller.
Control Horizon	Control horizon H_u of the MPC controller.
Output Weight	The link deflection weight Q_w of the MPC controller.
Input Rate Weight	Input rate weight R_u of the MPC controller.
Slewing Angle (deg)	The desired angle of rotation.
Max-Min Y Range	Maximum and minimum plot range of the system response in the y -axis of the graphic display.
Skip Plot Point, n	The number of sampling points skipped for plotting; i.e., the data are plotted at a rate of $(n+1)T_s$ ms.
Save Button	To activate the Save Dialog Box at which the name of the output data file and the number of skipped data points can be specified.

The FLMS software package is composed of several dialog boxes that operate within the main application program. Each dialog box belongs to an individual class with its implementation statements contained in the 'xxx.cpp' file and its declaration statements contained in the 'xxx.h' header file. Figure 6.10 shows the structure of the overall software program.

To ensure that the data acquisition functions provided by the motion control card can be used within the main application, all the header files and the C++ files in the 'stgconap.zip' should be added to the project workspace as has been done. The 'stgconap.zip' folder can be downloaded from the Servo To Go Website (<http://www.servotogo.com>). Moreover, the 'stgconap.cpp' file, which is a demo program running under the DOS prompt, should be modified and incorporated into our Visual C++ main application in order to have input and output displays in the Windows environments. All the global variables in the files should be added to the 'GUI.cpp' file and all the existing files should be included in the 'GUI.h' header file. The purpose of this modification is not only to have a user-friendly interface but also to

increase the operating speed of the control loop. It takes time for the data to pass from Windows to DOS for display; so, the original demo program can only have a maximum sampling rate of approximately 200 Hz. A polling function is implemented using the high-resolution performance counter provided by Visual C++. It keeps on polling for the next sampling interval before allowing the control loop to execute again. Since the counter has a very high resolution, the control cycle rate can be controlled accurately. By using this polling function, and by putting the control algorithm and the plotting function in a single control loop, fast real-time control is possible up to a sampling frequency of 1000 Hz.

The 'GUIDlg.cpp' file consists of the 'CGUIDlg' Class, which is derived from the built-in Microsoft Foundation Class 'Cdialog'. It is responsible for displaying the FLMS Main Dialog Box with pull down menu, to monitor if there is a menu item being selected and generate the corresponding 'COMMAND' event. Each 'COMMAND' event triggers the corresponding subroutine to display the functional dialog box by calling the 'DoModal()' member function of the 'Cdialog' Class. The flow diagram of the FLMS Dialog Box is shown in Figure 6.11. The menu item is linked to five dialog boxes including 'ADC,' 'DAC,' 'Encoder,' 'Manual Control,' and 'MPC Motion Control' with a unique ID. These dialog boxes are described in a separate C++ file in which the data flow of the control algorithms and the operating mechanisms of the onscreen controls are defined. With this software structure, the program is easy to maintain and can be extended to include additional software modules for testing different control algorithms.

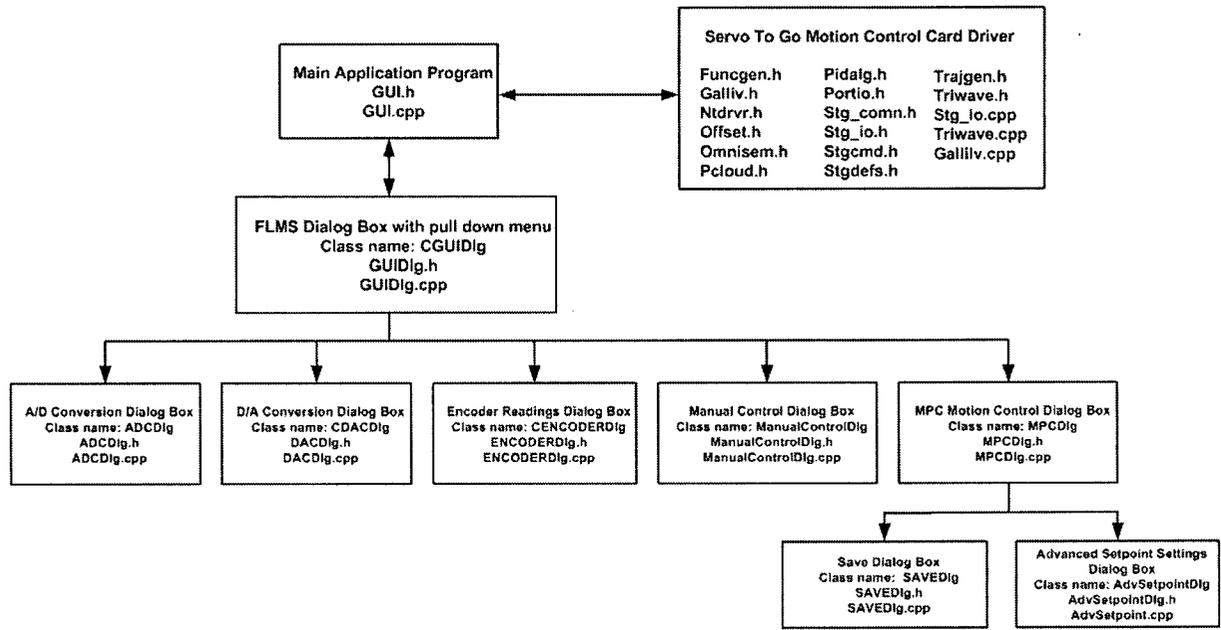


Figure 6.10 Program structure of the FLMS software package.

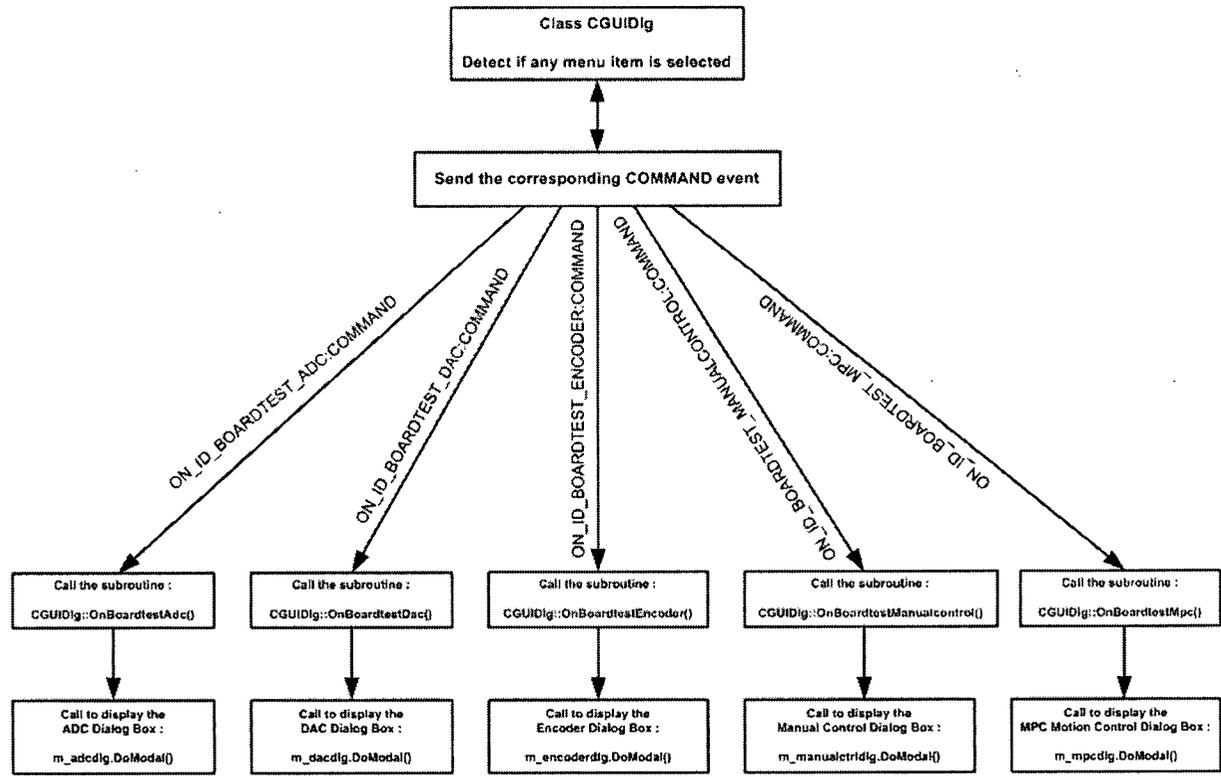


Figure 6.11 Flow diagram of the FLMS dialog box.

6.1.5 Operations of the Overall System

How the overall system operates is described now. First the user inputs the desired position through the GUI of the computer. Based on the control algorithm installed in the computer, the digital control signals are generated and sent to the motion control interface card. The motion control interface card then converts the digital control signals into analog signals. The resulting analog control signals for the joints are sent to the corresponding PWM amplifiers. The PWM amplifiers generate the necessary voltage signals to drive the two harmonic-drive motors. As a result, the manipulator moves toward the target location. Built-in optical encoders measure the angular displacements of the links. These digital position signals are fed back to the control computer through the motion control interface card. The force sensor measures the contact force at the end-effector. The signal conditioner processes the sensor signal. The amplified analog sensor signals are compatible with the motion control interface card. The ultrasonic sensor measures the end-effector position and sends the corresponding analog signal to the motion control interface card. The motion control interface card converts these analog signals into digital signals and sends them to the computer. The computer processes the position and force signals based on the control algorithm and generates the control signals for the robot actuators. The update control signals then sent to the actuator system to adjust the motion of the manipulator until the desired values are reached.

6.2 Experimental Results

Now we present the experimental results from the developed IMPC algorithm. The prototype FLMS has two flexible links; however, only the outboard link is used for the experiments. Joint module 2 is fixed by placing iron blocks on its sides. This system setup is identical to the single-link flexible manipulator shown in Chapter 4. The physical parameters of the manipulator are shown in Table 2.1.

The input to the system is the motor voltage. The system outputs are the joint angle and the link tip position. The joint angle is measured using the encoder. The link tip position is measured using the ultrasonic sensor. Only the measured joint angle is fed back into the MPC controller. The link tip position measurement is not used as a feedback for the controller, and is used only to evaluate the controller performance.

We assume that the link tip position at the desired final joint angle is zero. The reference target surface of the ultrasonic sensor is placed parallel to the final position of the link, at a distance of $y_r = 30$ cm. The link tip position can be obtained by using equation (6.5). The maximum inclined angle of the ultrasonic sensor is about 7° . Consequently we are able to use the ultrasonic sensor measurement data for link tip positions up to $y_{tip_measured} < 12$ cm.

Two different cases of motion configuration are tested on the FLMS. The desired joint angle for Case 1 is 30° and for Case 2 it is 60° . First, the MPC controller parameters are determined based on the dynamic model of the system using the fuzzy tuner, as we have done in Chapter 4. The MPC controller parameters are given in Table 6.10. Next, these controller parameters are imported into the FLMS MPC controller to control the motion of the system. The experimental and computer simulation results are shown in Figure 6.12 through Figure 6.17. From the experimental results we observe that good system responses are achieved in the motion control of the FLMS using IMPC control algorithm. The joint angle response has no overshoot, the control input is within its constraints (input voltage $|v(t)| \leq 2$ V or joint torque $|\tau(t)| \leq 6$ N.m), and the end-effector reaches its final position with no vibration and no steady-state error. Note that in the physical experiments, the measurement of the tip position response of Case 1 shown in Figure 6.13 starts from 2 seconds, and that of Case 2 shown in Figure 6.16 starts from 3 seconds. These delays in measurement are due to the limitation of the ultrasonic sensor. Specifically, in view of the inclined angle, we are able to accurately measure the link tip position only when the tip is close to the final position. Consequently, the non-minimum phase ‘dip’ in the initial tip response is not shown in these figures.

The experimental results are somewhat different from the simulation results. In particular, the joint angle and tip position responses of the prototype system are more sluggish than the corresponding simulated responses for both cases. The discrepancy of simulation and experimental results comes mainly due to the model errors and unmeasured disturbances. The model errors are caused primarily by inaccurate viscous and structural damping coefficients of the joint and the resulting inaccurate friction forces in the nonlinear model. In the computer model, we assume that the viscous damping coefficients are $b_{\theta_i} = 0.2$ and the structural damping coefficients are $b_{1_j} = 0.1$. Model error is unavoidable because these are not the true

values in the experimental system. Also, the nonlinear model used in computer simulations does not include the friction force of the rolling ball transfer at the end of the link. Due to these factors, the nonlinear model used in computer simulations is not an exact representation of the real manipulator. Even with the model error, it is clear from the experimental results that good system performance can be achieved by using the IMPC controller. In this sense the experimental results exhibit robustness of the controller with respect to model error. We can further improve the system performance by fine tuning the MPC controller parameters on line.

Table 6.10 MPC controller parameters.

	r_θ	H_p	H_u	Q_{yw}	Q_w	R_u
Case 1	$\frac{\pi}{3}$	530	9	20	7.8	3.0
Case 2	$\frac{\pi}{6}$	584	9	20	14.4	6.8

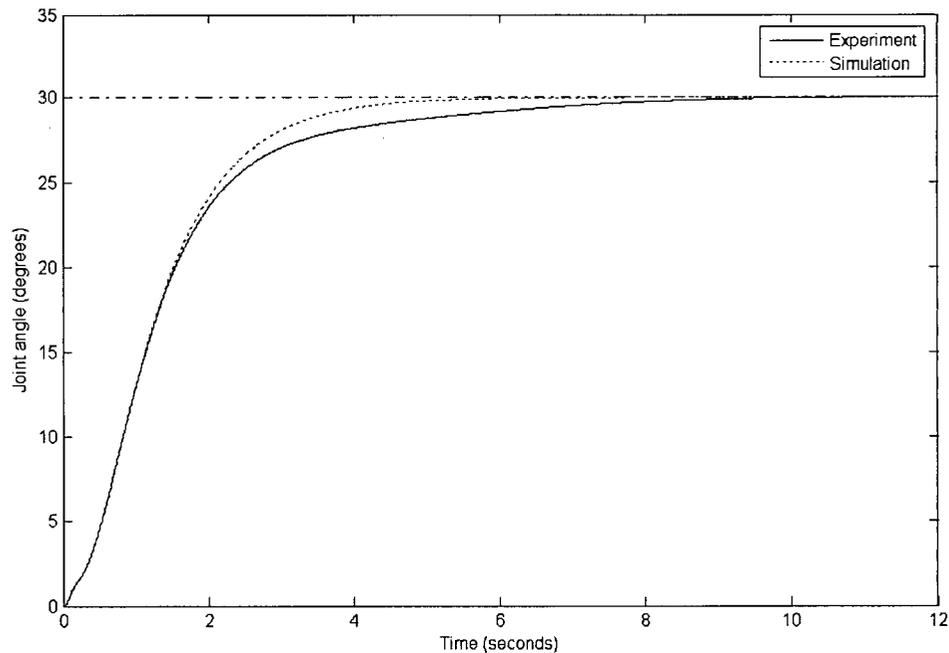


Figure 6.12 Case 1: Joint angle responses (experiment and simulation).

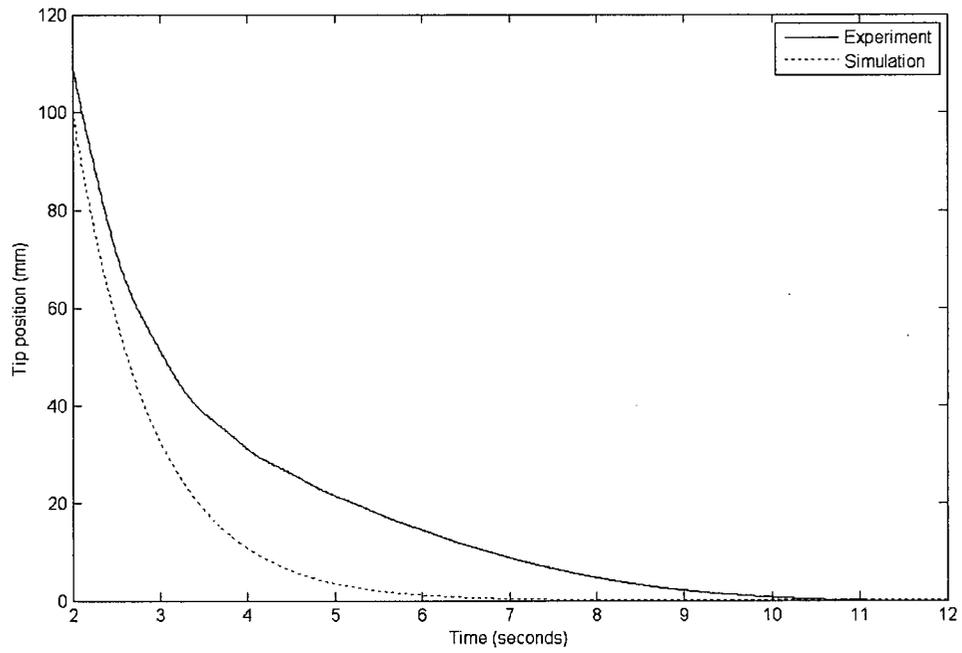


Figure 6.13 Case 1: Tip position responses (experiment and simulation).

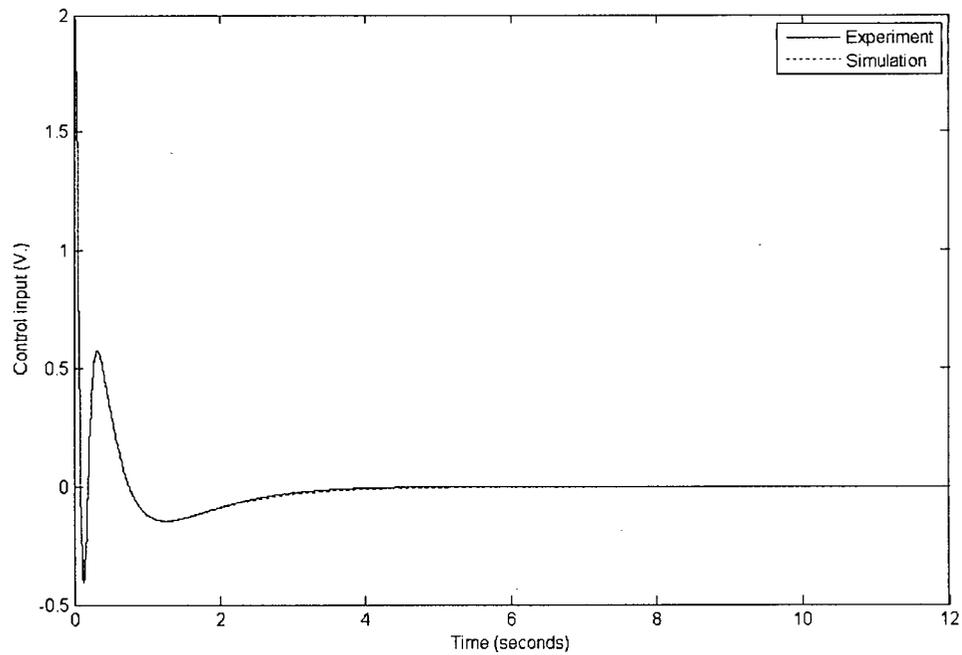


Figure 6.14 Case 1: Control input (experiment and simulation).

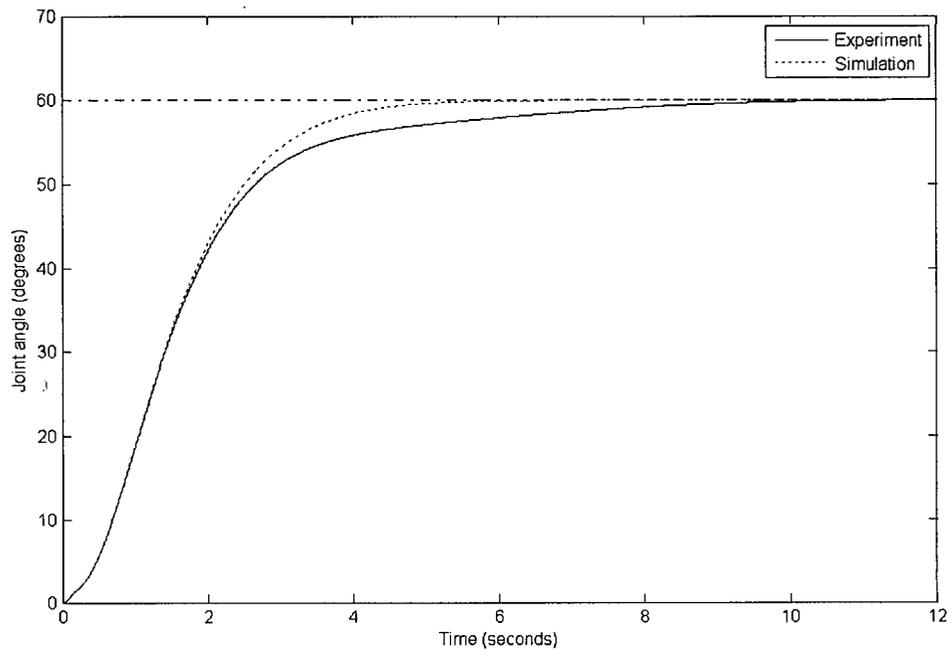


Figure 6.15 Case 2: Joint angle responses (experiment and simulation).

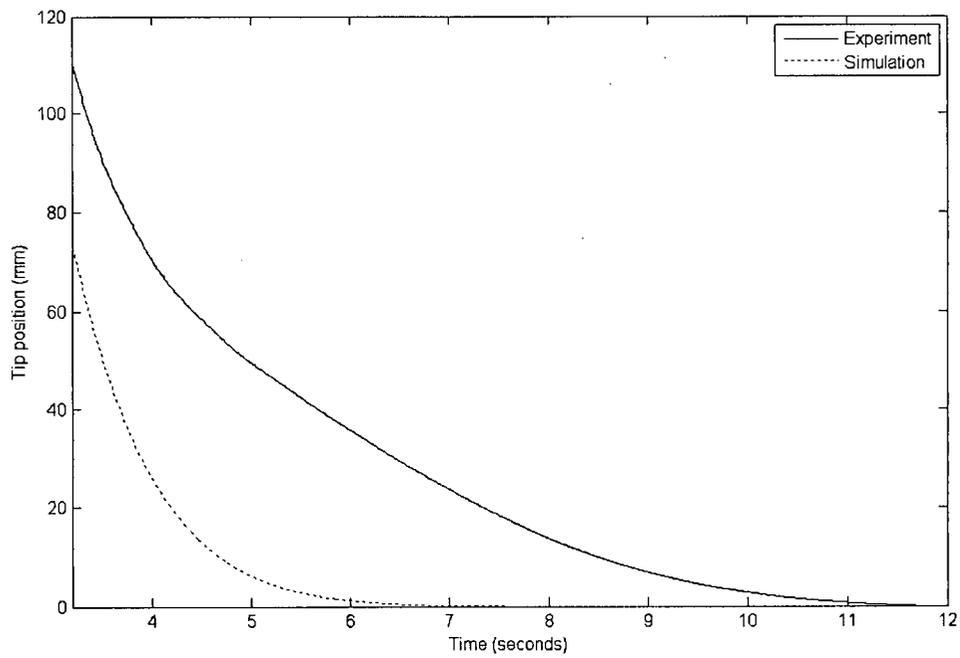


Figure 6.16 Case 2: Tip position responses (experiment and simulation).

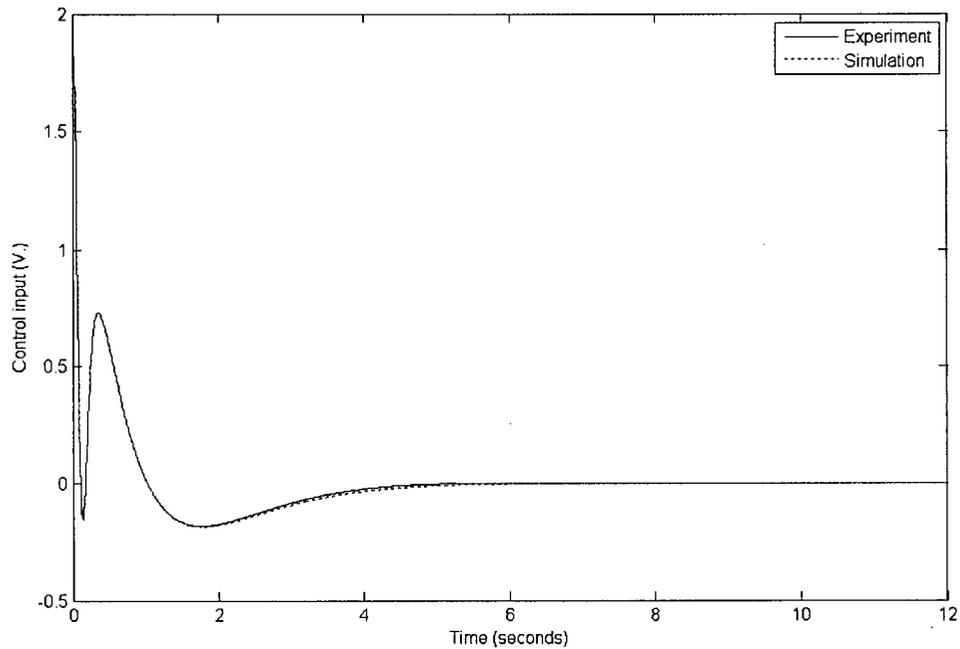


Figure 6.17 Case 2: Control input (experiment and simulation).

6.3 Summary

In this chapter we presented the development of the prototype flexible-link robot manipulator system (FLMS) in our laboratory, to investigate the performance of various motion/force control strategies and algorithms. The implementation of the developed IMPC control algorithm on the robot test-bed was described. Experimental results from the robot were presented under the developed IMPC scheme. The experimental results demonstrated that the IMPC control technique developed in the present research was quite effective in controlling the tip position of a prototype flexible-link manipulator.

Chapter 7

Conclusion

7.1 Primary Contributions

The main contributions of this research are summarized below:

- Kinematic and dynamic models for a class of flexible link robotic manipulators were developed. An accurate nonlinear dynamic model of a flexible-link manipulator system (FLMS) was developed using the assumed mode method. More realistic boundary conditions that represent the balance of moments and shear forces separately were incorporated in the analysis and simulation of the system. Effects of the payload on the mode shapes and the dynamics of the system were systematically analyzed. The non-minimum phase characteristics of the system were studied. Based on the model analysis and computer simulations, a linear model suitable for controller design was established. The developed model is rather simple to implement and analyze, yet capable of capturing the process dynamics accurately so as to properly predict the future outputs for use in the control scheme.
- An intelligent model predictive control (IMPC) strategy for motion control of the flexible link robot manipulator was developed based on a two-level hierarchical control architecture. This control structure was used to combine the advantages of conventional model predictive control (MPC) and knowledge-based soft control incorporating fuzzy logic.
- A computational efficient multi-stage MPC algorithm with guaranteed stability was developed. This MPC algorithm is used by the MPC module of the control structure for real time implementation of the overall adaptive scheme. In the developed scheme, the input constraints are not managed through optimization

but using a local anti-windup scheme and an intelligent fuzzy tuner. This reduces the optimization of a quadratic programming (QP) problem to a simple least square (LS) problem.

- Real time system identification module was designed to estimate an unknown payload carried by the robotic manipulator.
- A knowledge-based fuzzy tuner for auto-tuning the MPC controller was developed based on analysis, simulation, and experimentation of the prototype FLMS.
- Implementation of the IMPC scheme in a prototype robotic system was investigated. A flexible link manipulator system (FLMS) was designed and constructed in our laboratory. This system was used to evaluate and refine the developed modeling and control strategies. The performance of the developed IMPC scheme was evaluated using computer simulations and experimentation with the prototype FLMS. The results show that the IMPC developed in the present work is able to effectively control the motion of a flexible link robot manipulator.

7.2 Recommendations for Future Research

The present subject of research has not been exhausted in the work presented in this thesis. There is still work to be done on the study of flexible-link robotic manipulators. Some aspects of possible future research are summarized below:

- Develop a path planning technique for the motion control of the manipulator. This may result in reduced link vibrations and improved performance of the system.
- Develop an efficient nonlinear MPC controller. The controller may use the nonlinear model of the system as developed in the present work, for prediction and optimization. The nonlinear MPC controller can be used for highly nonlinear flexible-link manipulator systems.
- Refine the fuzzy knowledge of the fuzzy inference system based on physical experimentation using the prototype robotic system.
- Integrate intelligent learning techniques into the control architecture of the robotic

system. The system will be able to improve its knowledge base using offline training and online learning.

- Extend the developed IMPC technique to include contact force control of flexible-link manipulator systems.

Bibliography

- [1] Astrom, K.J. and B. Wittenmark, *Computer-Controlled Systems: Theory and Design*, 2nd Edition, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1990.
- [2] Astrom, K.J. and B. Wittenmark, *Adaptive Control*, 2nd Edition, Addison-Wesley Publishing, Inc., Reading, MA, 1995.
- [3] Bai, M., D.H. Zhou, and H. Schwarz, "Adaptive Augmented State Feedback Control for an Experimental Planar Two-Link Flexible Manipulator," *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 6, pp. 940-950, 1998.
- [4] Banavar, R.N. and P. Dominic, "An LQG/ H_∞ Controller for a flexible Manipulator," *IEEE Transactions on Control Systems Technology*, Vol. 3, pp. 409-416, 1995.
- [5] Bellezza, F., L. Lanari, and G. Ulivi, "Exact modeling of the slewing flexible link," *IEEE International Conference on Robotics and Automation*, Cincinnati, OH, pp. 734-739, 1990.
- [6] Benosman, M., F. Boyer, G. Le Vey, and D. Primault, "Flexible Links Manipulators: from Modelling to Control," *Journal of Intelligent and Robotic Systems*, Vol. 34, pp. 381-414, 2002.
- [7] Book, W.J., "Recursive Lagrangian Dynamics of Flexible Manipulator Arms," *International Journal of Robotics Research*, Vol. 3, No. 3, pp. 87-101, 1984.
- [8] Book, W.J., "Control Motion in an Elastic World," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 115, pp. 252-261, 1993.
- [9] Book, W.J., "Structural Flexibility of Motion Systems in the Space Environment," *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 5, pp. 524-530, 1993.
- [10] Cannon, R.H. and Jr. E. Schmitz, "Initial Experiments on the End-Point Control of a Flexible One-Link Robot," *International Journal of Robotics Research*, Vol. 3, No. 3, pp. 62-75, 1984.

- [11] Caswara, F.M. and H. Unbehauen, "A Neurofuzzy Approach to the Control of a Flexible-Link Manipulator," *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 6, pp. 932- 944, 2002.
- [12] Cetinkunt, S. and W. Yu, "Closed-Loop Behavior of a Feedback-Controlled Flexible Arm: A Comparative Study," *International Journal of Robotics Research*, Vol. 10, No. 3, pp. 263-275, 1991.
- [13] Choi, B.O. and K. Krishnamurthy, "Unconstrained and Constrained Motion Control of a Planar Two-Link Structurally Flexible Robotic Manipulator," *Journal of Robotic Systems*, Vol. 11, No. 6, pp. 557-571, 1994.
- [14] Clarke, D.W., C. Mohtadi, and P.S. Tuffs, "Generalized Predictive Control - Part I and II," *Automatica*, Vol. 23, No. 2, pp. 137-160, 1987.
- [15] Clarke, D.W. and C. Mohtadi, "Properties of Generalized Predictive Control," *Automatica*, Vol. 25, No. 6, pp. 859-875, 1989.
- [16] Culter, C.R. and B.L. Ramaker, "Dynamic Matrix Control - a Computer Control Algorithm," *Proc. Automatic Control Conference*, Paper WP5-B, San Francisco, CA, 1980.
- [17] De Luca, A. and B. Siciliano, "Trajectory Control of a Non-Linear One-Link Flexible Arm," *International Journal of Control*, Vol. 50, No. 5, pp. 1699-1715, 1989.
- [18] De Luca, A. and B. Siciliano, "Closed-Form Dynamic Model of Planar Multilink Lightweight Robots," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 21, pp. 826-839, 1991.
- [19] De Luca, A. and B. Siciliano, "Regulation of Flexible Arms Under Gravity," *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 4, pp. 463-467, 1993.
- [20] De Silva, C.W., *Control Sensors and Actuators*, Prentice Hall, Englewood Cliifs, NJ, 1989.
- [21] De Silva, C.W., "An Analytical Framework for Knowledge-Based Tuning of Servo Controllers," *Engineering Applications of Artificial Intelligence*, Vol. 4, No. 3, pp. 177-189, 1991.
- [22] De Silva, C.W., *Intelligent Control—Fuzzy Logic Applications*, CRC Press, Boca Raton, FL, 1995.
- [23] De Silva, C.W., *Mechatronics—An Integrated Approach*, Taylor & Francis/ CRC Press, Boca Raton, FL, 2005.

- [24] De Silva, C.W., *Vibration—Fundamentals and Practice*, 2nd Edition, Taylor & Francis/ CRC Press, Boca Raton, FL, 2006.
- [25] De Silva, C.W., *Sensors and Actuators—Control System Instrumentation*, Taylor & Francis/ CRC Press, Boca Raton, FL, 2007.
- [26] Demircioglu, H. and D.W. Clarke, “Generalized Predictive Control with End-Point State Weighting,” *IEE Proceedings-d Control Theory and Applications*, Vol. 140, No. 4, pp 275-282, 1993.
- [27] Fan, T. and C.W. de Silva, “Intelligent Model Predictive Control of a Flexible-Link Robotic Manipulator,” *Proc. ASME International Mechanical Engineering Congress and Exposition*, Orlando, FL, Paper IMECE2005-81899, 2005.
- [28] Fan, T. and C.W. de Silva, “Model Predictive Control of a Flexible-Link Robotic Manipulator,” *Proc. International Symposium on Collaborative Research in Applied Science*, Vancouver, B.C., Canada, pp. 166-173, 2005.
- [29] Feliu, V., J.A. Somolinos, C. Cerrada, and J.A. Cerrada, “A New Control Scheme of Single-link Flexible Manipulators Robust to Payload Changes,” *Journal of Intelligent and Robotic Systems*, Vol. 20, pp. 349-373, 1997.
- [30] Formalsky, A.M. and E.K. Lavrovsky, “Stabilization of Flexible One-Link Arm Position: Stability Domains in the Space of Feedback Gains,” *International Journal of Robotics Research*, Vol. 15, No. 5, pp. 492-504, 1996.
- [31] Garcia-Bentiez, E., S. Yurkovich, and K.M. Passino, “Rule-Based Supervisory Control of a Two-Link Flexible Manipulator,” *Journal of Intelligent and Robotics Systems*, Vol. 7, pp. 195-213, 1993.
- [32] Ge, S.S., T.H. Lee, and G. Zhu, “Improving Regulation of a Single-Link Flexible Manipulator with Strain Feedback,” *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 1, pp. 179-185, 1998.
- [33] Geniele, H., R.V. Patel, and K. Khorasani, “End-Point Control of a Flexible-Link Manipulator: Theory and Experiments,” *IEEE Transactions On Control Systems Technology*, Vol. 5, No. 6, 1997.
- [34] Green, A. and J.Z. Sasiadek, “Dynamics and Trajectory Tracking Control of a Two-Link Robot Manipulator,” *Journal of Vibration and Control*, Vol. 10, pp. 1415-1440, 2004.
- [35] Gutierrez, L.B. and F.L. Lewis, “Implementation of a Neural Network Tracking Controller for a Single Flexible Link: Comparison with PD and PID Controllers,” *IEEE Transactions on Industrial Electronics*, Vol. 45, No. 2, pp. 307-318, 1998.

- [36] Karray, F. and C.W. de Silva, *Soft Computing and Intelligent Systems Design*, Addison-Wesley Publishing, Inc., New York, NY, 2004.
- [37] Khalil, H.K., *Nonlinear Systems*, 3rd Edition, Prentice-Hall, Inc., Upper Saddle River, NJ, 2002.
- [38] Konno, A., D. Liu, and M. Uchiyama, "A Singularly Perturbed Method for Pole Assignment Control of a Flexible Manipulator," *Robotica*, Vol. 20, pp. 637-651, 2002.
- [39] Kouvaritakis, B. and M. Cannon, *Nonlinear Predictive Control: Theory and Practice*, The Institution of Electrical Engineers, London, UK, 2001.
- [40] Kothare, M.V., V. Balakrishnan, and M. Morari, "Robust Constrained Model Predictive Control Using Linear Matrix Inequalities," *Automatica*, Vol. 32, No. 10, pp. 1361-1379, 1996.
- [41] Kubica, E. and D. Wang, "A Fuzzy Control Strategy for a Flexible Single Link Robot," *Proc. IEEE International Conference on Robotics and Automation*, pp. 236-241, 1993.
- [42] Lee, H.H., "A New Trajectory Control of a Flexible-Link Robot Based on a Distributed-Parameter Dynamic Model," *International Journal of Control*, Vol. 77, No. 6, pp. 546-553, 2004.
- [43] Lee, J.H. and Z.H. Yu, "Tuning of Model Predictive Controllers for Robust Performance," *Computers & Chemical Engineering*, Vol. 18, No. 1, pp. 15-37, 1994.
- [44] Lee, K.S. and Y. Park, "Residual Vibration Reduction for a Flexible Structure Using a Modified Input Shaping Technique," *Robotica*, Vol. 20, pp. 553-561, 2002.
- [45] Lee, S.H. and C.W. Lee, "Hybrid Control Scheme for Robust Tracking of Two-Link Flexible Manipulator," *Journal of Intelligent and Robotic Systems*, Vol. 34, pp. 431-452, 2002.
- [46] Li, Y.F. and X.B. Chen, "End-Point Sensing and State Observation of a Flexible-Link Robot," *IEEE/ASME Transactions on Mechatronics*, Vol. 6, No. 3, pp. 351-356, 2001.
- [47] Liu, W. and G. Wang, "Auto-Tuning Procedure for Model-Based Predictive Controller," *Proc. IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 5, pp. 3421 - 3426, 2000.
- [48] Ljung, L., *System Identification - Theory for the User*, 2nd Edition, Prentice-Hall, Inc., Upper Saddle River, NJ, 1999.
- [49] Luo, Z. H., "Direct Strain Feedback Control of Flexible Robot Arms: New Theoretical and Experimental Results," *IEEE Transactions on Automatic Control*, Vol. 38, No. 11, pp. 1610-1622, 1993.

- [50] Luo, Z. H. and B. Guo, "Further Theoretical Results on Direct Strain Feedback Control of Flexible Robot Arms," *IEEE Transactions on Automatic Control*, Vol. 40, No. 4, pp. 747-751, 1995.
- [51] Maciejowski, J.M., *Multivariable Feedback Design*, Addison-Wesley Publishing, Inc., Reading, MA, 1989.
- [52] Maciejowski, J.M., *Predictive Control: With Constraints*, Pearson Education Limited, Harlow, England, 2002.
- [53] Madhavan, S.K. and S.N. Singh, "Inverse Trajectory Control and Zero Dynamics Sensitivity of an Elastic Manipulator," *International Journal of Robotics and Automation*, Vol. 6, No. 4, pp. 179-192, 1991.
- [54] Magee, D.P. and W.J. Book, "Eliminating Multiple Modes of Vibration in a Flexible Manipulator," *Proceedings of the IEEE Conference on Robotics and Automation*, pp. 474-479, 1993.
- [55] Maplesoft, *Maple 10 User Manual*, Waterloo Maple, Inc., Waterloo, ON, Canada, 2005.
- [56] Mathworks, *Control System Toolbox: For Use with Matlab - User's Guide*, Version 5, The Mathworks, Inc., Natick, MA, 2002.
- [57] Mathworks, *Fuzzy Logic Toolbox: For Use with Matlab - User's Guide*, Version 2, The Mathworks, Inc., Natick, MA, 2002.
- [58] Mathworks, *Model Predictive Control Toolbox: For Use with Matlab - User's Guide*, Version 2, The Mathworks, Inc., Natick, MA, 2005.
- [59] Mathworks, *System Identification Toolbox: For Use with Matlab - User's Guide*, Version 6, The Mathworks, Inc., Natick, MA, 2003.
- [60] Moallem, M., K. Khorasani and R.V. Patel, "An Integral Manifold Approach for Tip Position Tracking of Flexible Multi-Link Manipulators," *IEEE Transactions on Robotics and Automation*, Vol. 13, No. 6, pp. 823-837, 1997.
- [61] Moallem, M., R.V. Patel, and K. Khorasani, *Flexible-Link Robot Manipulators: Control Techniques and Structural Design - (Lecture Notes in Control and Information Sciences; 257)*, Springer-Verlag, London, UK, 2000.
- [62] Morris, A.S. and A. Madani, "Quadratic Optimal Control of a Two-Flexible-Link Robot Manipulator," *Robotica*, Vol. 16, pp. 97-108, 1998.

- [63] Moudgal, V.G., K.M. Passino, and S. Yurkovich, "Rule-Based Control for a Flexible-Link Robot," *IEEE Transactions on Control Systems Technology*, Vol. 2, No. 4, pp. 392-405, 1994.
- [64] Nemir, D.C., A.J. Koivo, and R.L. Kashyap, "Pseudolinks and the Self-Tuning Control of a Nonrigid Link Mechanism," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 18, No. 1, pp. 40-48, 1988.
- [65] Nho, H.C. and P. Meckl, "Intelligent Feedforward Control and Payload Estimation for a Two-Link Robotic Manipulator," *IEEE/ASME Transactions On Mechatronics*, Vol.8, No. 2, pp. 277-283, 2003.
- [66] Prett, D., G. Garcia, and C. Cutler, "The Dynamic Matrix Control Method," *US Patent 4349869*, 1982.
- [67] Pistikopoulos, E.N., V. Dua, N.A. Bozinis, A. Bemporad, and M. Morari, "On-Line Optimization Via Off-Line Parametric Optimization Tools," *Computers and Chemical Engineering*, Vol. 24, pp. 183-188, 2000.
- [68] Rao, S.S., *Mechanical Vibrations*, 3rd Edition, Addison-Wesley Publishing, Inc., Reading, MA, 1995.
- [69] Rawlings, J.B. and K.R. Muske, "The Stability of Constrained Receding Horizon Control," *IEEE Transactions on Automatic Control*, Vol. 38, No. 10, pp. 1512-1516, 1993.
- [70] Rossiter, J.A., *Model-Based Predictive Control: A Practical Approach*, CRC Press, Boca Raton, FL, 2000.
- [71] Rossi, M., D. Wang, and K. Zuo, "Issues in the Design of Passive Controllers for Flexible-Link Robots," *International Journal of Robotics Research*, Vol. 16, No. 4, pp. 577-588, 1997.
- [72] Ryu, J.H., D.S. Kwon, and Y. Park, "A Robust Controller Design Method for a Flexible Manipulator with a Large Time Varying Payload and Parameter Uncertainties," *Journal of Intelligent and Robotic Systems*, Vol. 27, pp. 345-361, 2000.
- [73] Shaheed, M.H. and M.O. Tokhi, "Dynamic Modelling of a Single-Link Flexible Manipulator: Parametric and Non-Parametric Approaches," *Robotica*, Vol. 20, pp. 93-109, 2002.
- [74] Siciliano, B., W.J. Book and G. De Maria, "An Integral Manifold Approach to Control of a One-Link Flexible Arm," *Proc. 25th IEEE Conference On Decision and Control*, pp. 1131-1134, 1986.

- [75] Siciliano, B. and W.J. Book, "A Singular Perturbation Approach to Control of Lightweight Flexible Manipulators," *International Journal of Robotics Research*, Vol. 7, pp. 79-90, 1988.
- [76] Siciliano, B., J.V.R. Prasad, and A.J. Calise, "Output Feedback Two-Time Scale Control of Multi-Link Flexible Arms," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 114, pp. 70-77, 1992.
- [77] Singer, N.C. and W.P. Seering, "Preshaping Command Inputs to Reduce System Vibration," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 112, pp. 76-82, 1990.
- [78] Slotine, J.-J.E. and W. P. Li, *Applied Nonlinear Control*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1991.
- [79] Song, B. J. and A.J. Koivo, "Nonlinear Predictive Control with Application to Manipulator with Flexible Forearm," *IEEE Transactions on Industrial Electronics*, Vol. 46, No. 5, pp. 923-932, 1999.
- [80] Spong, M.W., "Modeling and Control of Elastic Joint Robots," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 109, No. 3, pp. 310-319, 1987.
- [81] Spong, M.W., K. Khorasani and P.V. Kokotovic, "An Integral Manifold Approach to the Feedback Control of Flexible Joint Robots," *IEEE Transactions on Robotics and Automation*, Vol. 3, No. 4, pp. 291-299, 1987.
- [82] Spong, M.W. and M. Vidyasagar, *Robot Dynamics and Control*, John Willey & Sons, Inc., New York, NY, 1989.
- [83] Subudhi, B. and A.S. Morris, "Dynamic Modelling, Simulation and Control of a Manipulator with Flexible Links and Joints," *Robotics and Autonomous Systems*, Vol. 41, No. 4, pp. 257-270, 2002.
- [84] Subudhi, B. and A.S. Morris, "Singular Perturbation Approach to Trajectory Tracking of Flexible Robot with Joint Elasticity," *International Journal of Systems Science*, Vol. 34, No. 3, pp. 167-179, 2003.
- [85] Talebi, H.A., R.V. Patel, and K. Khorasani, *Control of Flexible-Link Manipulators Using Neural Networks - (Lecture Notes in Control and Information Sciences; 261)*, Springer-Verlag, London, UK, 2001.
- [86] Tokhi, M.O. and A.K.M. Azad, "Design and Development of an Experimental Flexible Manipulator System," *Robotica*, Vol. 15, pp. 283-292, 1997.
- [87] Tokhi, M.O., Z. Mohamed, and M.H. Shaheed, "Dynamic Characterisation of a Flexible Manipulator System," *Robotica*, Vol. 19, pp. 571-580, 2001.

- [88] Tondel, P., T.A. Johansena, and A. Bemporad, "Evaluation of Piecewise Affine Control Via Binary Search Tree," *Automatica*, Vol. 39, pp. 945-950, 2003.
- [89] Wai, R.J. and M.C. Lee, "Intelligent Optimal Control of Single-Link Flexible Robot Arm," *IEEE Transactions on Industrial Electronics*, Vol. 51, No. 1, pp. 201-220, 2004.
- [90] Wang, F.Y. and Y. Gao, *Advanced Studies of Flexible Robotic Manipulator - Modeling, Design, Control and Applications*, World Scientific Publishing, River Edge, NJ, 2003.
- [91] Wang, D. and M. Vidyasagar, "Transfer Functions for a Single Flexible Link," *Proc. IEEE International Conference on Robotics and Automation*, pp. 1042-1047, 1989.
- [92] Wang, D. and M. Vidyasagar, "Control of a Class of Manipulators with a Single Flexible Link - Part I: Feedback Linearization," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 113, No. 4, pp. 655-661, 1991.
- [93] Wang, D. and M. Vidyasagar, "Control of a Class of Manipulators with a Single Flexible Link - Part II: Observer-Controller Stabilization," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 113, No. 4, pp. 662-668, 1991.
- [94] Wilson, D.G., R.D. Robinett, G.G. Parker and G.P. Starr, "Augmented Sliding Mode Control for Flexible Link Manipulators" *Journal of Intelligent and Robotic Systems*, Vol. 34, pp. 415-430, 2002.
- [95] Wu, Q.M. and C.W. de Silva, "The Effect of Fuzzy Resolution in Servo Tuning," *Proc. Second IEEE Conference on Control Applications*, Vol. 1, pp. 77-82, Vancouver, B.C., 1993.
- [96] Yim, W., "Modified Nonlinear Predictive Control of Elastic Manipulators," *Proc. IEEE International Conference on Robotics and Automation*, Minneapolis, MN, pp. 2097-2102, 1996.
- [97] Yurkovich, S. and A.P. Tzes, "Experiments in Identification and Control of Flexible-Link Manipulators," *IEEE Control Systems Magazine*, Vol. 10, No. 2, pp. 41 - 46, 1990.
- [98] Zhu, G., S.S. Ge, and T.H. Lee, "Simulation Studies of Tip Tracking Control of a Single-Link Flexible Robot Based on a Lumped Model," *Robotica*, Vol. 17, pp. 71-78, 1999.

Appendix A

Dynamic Models

This appendix presents the dynamic models for the flexible-link robotic manipulators which have been studied in the course of the research presented in this thesis.

A.1 Single-link Flexible Manipulator Model

A.1.1 Nonlinear Model

The rigid body equation for the link:

$$\begin{aligned}
 & \left(J_{h1} + J_{o1} + J_p + m_p l_1^2 \right) \ddot{\theta}_1 + \left[\int_0^{l_1} \rho_1 \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}(x_1) \delta_{1j}(t) \right)^2 dx_1 + m_p \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \delta_{1j}(t) \right)^2 \right] \ddot{\theta}_1 \\
 & + \sum_{j=1}^{n_{m,1}} \left(I_{11j} + m_p l_1 \phi_{1j}(l_1) + J_p \phi_{1j}'(l_1) \right) \ddot{\delta}_{1j}(t) \\
 & + 2\dot{\theta}_1 \left[\int_0^{l_1} \rho_1 \sum_{j=1}^{n_{m,1}} \phi_{1j}(x_1) \delta_{1j}(t) \sum_{j=1}^{n_{m,1}} \phi_{1j}(x_1) \dot{\delta}_{1j}(t) dx_1 + m_p \sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \delta_{1j}(t) \sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \dot{\delta}_{1j}(t) \right] + b_\theta \dot{\theta}_1 = \tau_1
 \end{aligned} \tag{A.1}$$

The deflection equation for the j th mode of the link:

$$\begin{aligned}
& \left(I_{11j} + m_p l_1 \phi_{1j}(l_1) + J_p \phi_{1j}'(l_1) \right) \ddot{\theta}_1 \\
& + \sum_{k=1}^{n_{m-1}} \left[\int_0^{l_1} \rho_1 \phi_{1j}(x_1) \phi_{1k}(x_1) dx + m_p \phi_{1j}(l_1) \phi_{1k}(l_1) + J_p \phi_{1j}'(l_1) \phi_{1k}'(l_1) \right] \ddot{\delta}_{1k}(t) \\
& - \sum_{k=1}^{n_{m-1}} \left[\int_0^{l_1} \rho_1 \phi_{1j}(x_1) \phi_{1k}(x_1) dx + m_p \phi_{1j}(l_1) \phi_{1k}(l_1) \right] \delta_{1k}(t) \dot{\theta}_1^2 \\
& + \omega_{1j}^2 \left[m_1 + m_p \phi_{1j}(l_1)^2 + J_p \phi_{1j}'(l_1)^2 \right] \delta_{1j}(t) + b_{1j} \dot{\delta}_{1j} = 0
\end{aligned} \tag{A.2}$$

A.1.2 Continuous-time State-space Model

The continuous-time state-space model of the system with first flexible modes is:

$$\begin{aligned}
\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}u \\
\mathbf{y} &= \mathbf{C}\mathbf{x}
\end{aligned} \tag{A.3}$$

The numerical values of the elements of the system matrix \mathbf{A} are:

$$\begin{aligned}
A(1,1) &= 0, & A(1,2) &= 0, & A(1,3) &= 1, & A(1,4) &= 0 \\
A(2,1) &= 0, & A(2,2) &= 0, & A(2,3) &= 0, & A(2,4) &= 1 \\
A(3,1) &= 0, & A(3,2) &= \frac{212.32m_p + 23.708 + 313.804J_p}{0.769 + 17.669J_p + 8.0m_p + 0.959J_p m_p} \\
A(3,3) &= \frac{-0.081 - 0.839m_p - 1.833J_p}{0.769 + 17.669J_p + 8.0m_p + 0.959J_p m_p} \\
A(3,4) &= \frac{0.205m_p + 0.023 + 0.303J_p}{0.769 + 17.669J_p + 8.0m_p + 0.959J_p m_p} \\
A(4,1) &= 0, & A(4,2) &= \frac{-210.861 - 103.643J_p - 103.643m_p}{0.769 + 17.669J_p + 8.0m_p + 0.959J_p m_p} \\
A(4,3) &= \frac{0.41m_p + 0.046 + 0.606J_p}{0.769 + 17.669J_p + 8.0m_p + 0.959J_p m_p} \\
A(4,4) &= \frac{-0.203 - 0.1J_p - 0.1m_p}{0.769 + 17.669J_p + 8.0m_p + 0.959J_p m_p}
\end{aligned}$$

The numerical values of the system matrices \mathbf{B} and \mathbf{C} are:

$$\mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ \frac{0.404 + 4.197m_p + 9.167J_p}{0.769 + 17.669J_p + 8.0m_p + 0.959J_p m_p} \\ \frac{-2.049m_p - 0.229 - 3.028J_p}{0.769 + 17.669J_p + 8.0m_p + 0.959J_p m_p} \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2.049 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

For the nominal payload ($m_p = 0.38\text{kg}$, $J_p = 0.001\text{kg.m}^2$), the numerical values of the system matrices \mathbf{A} , \mathbf{B} and \mathbf{C} are:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 27.350 & -0.105 & 0.026 \\ 0 & -65.394 & 0.0528 & -0.063 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 0.524 \\ -0.264 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2.049 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

A.1.3 Discrete-time State-space Model

The discrete-time state-space model of the system with just the first flexible mode and the nominal payload is:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) \end{aligned} \quad (\text{A.4})$$

The numerical values of the system matrices \mathbf{A} , \mathbf{B} and \mathbf{C} are:

$$\mathbf{A} = \begin{bmatrix} 1 & 0.001366 & 0.009995 & 5.873\text{e-}006 \\ 0 & 0.9967 & 2.636\text{e-}006 & 0.009986 \\ 0 & 0.273 & 0.999 & 0.001629 \\ 0 & -0.653 & 0.0005268 & 0.9961 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 2.62\text{e-}005 \\ -1.318\text{e-}005 \\ 0.005239 \\ -0.002634 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2.049 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

A.2 Two-link Flexible Manipulator Model

Rigid body equation for link 1:

$$\left. \begin{aligned}
 & J_{h1} + J_{h2} + J_{o1} + J_{o2} + J_p \\
 & + m_p l_2^2 + (m_2 + m_{h2} + m_p) [l_1^2 + w_1^2(l_1, t)] \\
 & + (m_2 + 2m_p) l_2 [w_1(l_1, t) \sin(\alpha_1 + \theta_2) + l_1 \cos(\alpha_1 + \theta_2)] \\
 & - 2 [l_1 \sin(\alpha_1 + \theta_2) - w_1(l_1, t) \cos(\alpha_1 + \theta_2)] \sum_{j=1}^{n_{m,2}} [I_{42j} + m_p \phi_{2j}(l_2)] \delta_{2j}(t) \\
 & + \int_0^{l_1} \rho_1 \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}(x_1) \delta_{1j}(t) \right)^2 dx_1 + \int_0^{l_2} \rho_2 \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(x_2) \delta_{2j}(t) \right)^2 dx_2 \\
 & + m_p \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(l_2) \delta_{2j}(t) \right)^2
 \end{aligned} \right\} \ddot{\theta}_1$$

$$+ \left. \begin{aligned}
 & J_{h2} + J_{o2} + J_p + m_p l_2^2 + \left(\frac{1}{2} m_2 + m_p \right) l_2 [w_1(l_1, t) \sin(\alpha_1 + \theta_2) + l_1 \cos(\alpha_1 + \theta_2)] \\
 & - [l_1 \sin(\alpha_1 + \theta_2) - w_1(l_1, t) \cos(\alpha_1 + \theta_2)] \sum_{j=1}^{n_{m,2}} [I_{42j} + m_p \phi_{2j}(l_2)] \delta_{2j}(t) \\
 & + \int_0^{l_2} \rho_2 \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(x_2) \delta_{2j}(t) \right)^2 dx_2 + m_p \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(l_2) \delta_{2j}(t) \right)^2
 \end{aligned} \right\} \ddot{\theta}_2$$

$$+ \sum_{j=1}^{n_{m,1}} I_{11j} \ddot{\delta}_{1j}(t) + \left. \begin{aligned}
 & l_1 (m_2 + m_{h2} + m_p) + \left(\frac{1}{2} m_2 + m_p \right) l_2 \cos(\alpha_1 + \theta_2) \\
 & - \sin(\alpha_1 + \theta_2) \sum_{j=1}^{n_{m,2}} [I_{42j} + m_p \phi_{2j}(l_2)] \delta_{2j}(t)
 \end{aligned} \right\} \sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \ddot{\delta}_{1j}(t)$$

$$+ \left. \begin{aligned}
 & J_{h2} + J_{o2} + J_p + m_p l_2^2 \\
 & + \left(\frac{1}{2} m_2 + m_p \right) l_2 [w_1(l_1, t) \sin(\alpha_1 + \theta_2) + l_1 \cos(\alpha_1 + \theta_2)] \\
 & - \begin{bmatrix} l_1 \sin(\alpha_1 + \theta_2) \\ -w_1(l_1, t) \cos(\alpha_1 + \theta_2) \end{bmatrix} \sum_{j=1}^{n_{m,2}} [I_{42j} + m_p \phi_{2j}(l_2)] \delta_{2j}(t) \\
 & + \int_0^{l_2} \rho_2 \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(x_2) \delta_{2j}(t) \right)^2 dx_2 + m_p \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(l_2) \delta_{2j}(t) \right)^2
 \end{aligned} \right\} \sum_{j=1}^{n_{m,1}} \phi_{1j}'(l_1) \ddot{\delta}_{1j}(t)$$

$$\begin{aligned}
& + \sum_{j=1}^{n_{m,2}} \left[I_{12j} + J_p \phi_{2j}'(l_2) + m_p l_2 \phi_{2j}(l_2) \right] \ddot{\delta}_{2j}(t) \\
& + \left[w_1(l_1, t) \sin(\alpha_1 + \theta_2) + l_1 \cos(\alpha_1 + \theta_2) \right] \sum_{j=1}^{n_{m,2}} \left[I_{42j} + m_p \phi_{2j}(l_2) \right] \ddot{\delta}_{2j}(t) \\
& - \left\{ \begin{aligned}
& \left(m_2 + 2m_p \right) l_2 \dot{w}_1'(l_1, t) \left[l_1 \sin(\alpha_1 + \theta_2) - w_1(l_1, t) \cos(\alpha_1 + \theta_2) \right] \\
& - \left(m_2 + 2m_p \right) \dot{w}_1(l_1, t) \left[l_2 \sin(\alpha_1 + \theta_2) + 2w_1(l_1, t) \right] + 2(m_p - m_{n2}) w_1(l_1, t) \dot{w}_1(l_1, t) \\
& + 2 \left[\dot{w}_1'(l_1, t) \left[w_1(l_1, t) \sin(\alpha_1 + \theta_2) + l_1 \cos(\alpha_1 + \theta_2) \right] \right. \\
& \quad \left. - \dot{w}_1(l_1, t) \cos(\alpha_1 + \theta_2) \right] \sum_{j=1}^{n_{m,2}} \left[I_{42j} + m_p \phi_{2j}(l_2) \right] \delta_{2j}(t) \\
& + 2 \left[l_1 \sin(\alpha_1 + \theta_2) - w_1(l_1, t) \cos(\alpha_1 + \theta_2) \right] \sum_{j=1}^{n_{m,2}} \left[I_{42j} + m_p \phi_{2j}(l_2) \right] \dot{\delta}_{2j}(t) \\
& - 2\rho_1 \int_0^{l_1} \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}(x_1) \delta_{1j}(t) \right) \left(\sum_{j=1}^{n_{m,1}} \phi_{1j}(x_1) \dot{\delta}_{1j}(t) \right) dx_1 \\
& - 2\rho_2 \int_0^{l_2} \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(x_2) \delta_{2j}(t) \right) \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(x_2) \dot{\delta}_{2j}(t) \right) dx_2 \\
& - 2m_p \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(l_2) \delta_{2j}(t) \right) \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(l_2) \dot{\delta}_{2j}(t) \right)
\end{aligned} \right\} \dot{\theta}_1 \\
& - \left\{ \begin{aligned}
& \left(m_2 + 2m_p \right) l_2 \dot{w}_1'(l_1, t) \left[l_1 \sin(\alpha_1 + \theta_2) - w_1(l_1, t) \cos(\alpha_1 + \theta_2) \right] \\
& + 2 \left[l_1 \sin(\alpha_1 + \theta_2) - w_1(l_1, t) \cos(\alpha_1 + \theta_2) \right] \sum_{j=1}^{n_{m,2}} \left[I_{42j} + m_p \phi_{2j}(l_2) \right] \dot{\delta}_{2j}(t) \\
& + 2 \dot{w}_1'(l_1, t) \left[w_1(l_1, t) \sin(\alpha_1 + \theta_2) + l_1 \cos(\alpha_1 + \theta_2) \right] \sum_{j=1}^{n_{m,2}} \left[I_{42j} + m_p \phi_{2j}(l_2) \right] \delta_{2j}(t) \\
& - 2\rho_2 \int_0^{l_2} \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(x_2) \delta_{2j}(t) \right) \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(x_2) \dot{\delta}_{2j}(t) \right) dx_2 \\
& - 2m_p \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(l_2) \delta_{2j}(t) \right) \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(l_2) \dot{\delta}_{2j}(t) \right)
\end{aligned} \right\} \dot{\theta}_2 \\
& - \left\{ \begin{aligned}
& \left(m_2 + 2m_p \right) l_2 \left[l_1 \sin(\alpha_1 + \theta_2) - w_1(l_1, t) \cos(\alpha_1 + \theta_2) \right] \\
& + 2 \left[w_1(l_1, t) \sin(\alpha_1 + \theta_2) + l_1 \cos(\alpha_1 + \theta_2) \right] \sum_{j=1}^{n_{m,2}} \left[I_{42j} + m_p \phi_{2j}(l_2) \right] \delta_{2j}(t)
\end{aligned} \right\} \dot{\theta}_1 \dot{\theta}_2
\end{aligned}$$

$$\begin{aligned}
& - \left\{ \begin{aligned} & \left(\frac{1}{2} m_2 + m_p \right) l_2 \left[l_1 \sin(\alpha_1 + \theta_2) - w_1(l_1, t) \cos(\alpha_1 + \theta_2) \right] \\ & + \left[w_1(l_1, t) \sin(\alpha_1 + \theta_2) + l_1 \cos(\alpha_1 + \theta_2) \right] \sum_{j=1}^{n_{m,2}} \left[I_{42j} + m_p \phi_{2j}(l_2) \right] \delta_{2j}(t) \end{aligned} \right\} \dot{\theta}_2^2 \\
& - \dot{w}_1'(l_1, t) \left\{ \begin{aligned} & 2 \left[l_1 \sin(\alpha_1 + \theta_2) - w_1(l_1, t) \cos(\alpha_1 + \theta_2) \right] \sum_{j=1}^{n_{m,2}} \left[I_{42j} + m_p \phi_{2j}(l_2) \right] \dot{\delta}_{2j}(t) \\ & - 2 \rho_2 \int_0^{l_2} \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(x_2) \delta_{2j}(t) \right) \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(x_2) \dot{\delta}_{2j}(t) \right) dx_2 \\ & - 2 m_p \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(x_2) \delta_{2j}(t) \right) \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(x_2) \dot{\delta}_{2j}(t) \right) \end{aligned} \right\} \\
& - \left[\dot{w}_1'(l_1, t) \right]^2 \left\{ \begin{aligned} & \left(\frac{1}{2} m_2 + m_p \right) l_2 \begin{bmatrix} l_1 \sin(\alpha_1 + \theta_2) \\ -w_1(l_1, t) \cos(\alpha_1 + \theta_2) \end{bmatrix} \\ & + \begin{bmatrix} w_1(l_1, t) \sin(\alpha_1 + \theta_2) \\ +l_1 \cos(\alpha_1 + \theta_2) \end{bmatrix} \sum_{j=1}^{n_{m,2}} \begin{bmatrix} I_{42j} \\ +m_p \phi_{2j}(l_2) \end{bmatrix} \delta_{2j}(t) \end{aligned} \right\} + b_{\theta_1} \dot{\theta}_1 = \tau_1 \quad (\text{A.5})
\end{aligned}$$

Rigid body equation for link 2:

$$\begin{aligned}
& \left\{ \begin{aligned} & J_{h2} + J_{o2} + J_p + m_p l_2^2 + \left(\frac{1}{2} m_2 + m_p \right) l_2 \left[w_1(l_1, t) \sin(\alpha_1 + \theta_2) + l_1 \cos(\alpha_1 + \theta_2) \right] \\ & - \left[l_1 \sin(\alpha_1 + \theta_2) - w_1(l_1, t) \cos(\alpha_1 + \theta_2) \right] \sum_{j=1}^{n_{m,2}} \left[I_{42j} + m_p \phi_{2j}(l_2) \right] \delta_{2j}(t) \\ & + \int_0^{l_2} \rho_2 \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(x_2) \delta_{2j}(t) \right)^2 dx_2 + m_p \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(l_2) \delta_{2j}(t) \right)^2 \end{aligned} \right\} \ddot{\theta}_1 \\
& + \left[\begin{aligned} & J_{h2} + J_{o2} + J_p + m_p l_2^2 \\ & + \int_0^{l_2} \rho_2 \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(x_2) \delta_{2j}(t) \right)^2 dx_2 + m_p \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(l_2) \delta_{2j}(t) \right)^2 \end{aligned} \right] \ddot{\theta}_2
\end{aligned}$$

$$\begin{aligned}
& + \left[\begin{aligned} & \left(\frac{1}{2} m_2 + m_p \right) l_2 \cos(\alpha_1 + \theta_2) \\ & - \sin(\alpha_1 + \theta_2) \sum_{j=1}^{n_{m,2}} (I_{42j} + m_p \phi_{2j}(l_2)) \delta_{2j}(t) \end{aligned} \right] \sum_{j=1}^{n_{m,1}} \phi_{1j}(l_1) \ddot{\delta}_{1j}(t) \\
& + \left[\begin{aligned} & J_{h2} + J_{o2} + J_p + m_p l_2^2 \\ & + \int_0^{l_2} \rho_2 \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(x_2) \delta_{2j}(t) \right)^2 dx_2 + m_p \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(l_2) \delta_{2j}(t) \right)^2 \end{aligned} \right] \sum_{j=1}^{n_{m,1}} \phi_{1j}'(l_1) \ddot{\delta}_{1j}(t) \\
& + \sum_{j=1}^{n_{m,2}} \left[I_{12j} + J_p \phi_{2j}'(l_2) + m_p l_2 \phi_{2j}(l_2) \right] \ddot{\delta}_{2j}(t) \\
& + \left. \left\{ \begin{aligned} & (m_2 + 2m_p) l_2 \dot{w}_1(l_1, t) \sin(\alpha_1 + \theta_2) \\ & + 2\dot{w}_1(l_1, t) \cos(\alpha_1 + \theta_2) \sum_{j=1}^{n_{m,2}} [I_{42j} + m_p \phi_{2j}(l_2)] \delta_{2j}(t) \\ & + 2\rho_2 \int_0^{l_2} \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(x_2) \delta_{2j}(t) \right) \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(x_2) \dot{\delta}_{2j}(t) \right) dx_2 \\ & + 2m_p \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(l_2) \delta_{2j}(t) \right) \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(l_2) \dot{\delta}_{2j}(t) \right) \end{aligned} \right\} \dot{\theta}_1 \\
& + \left. \left\{ \begin{aligned} & \left(\frac{1}{2} m_2 + m_p \right) l_2 [l_1 \sin(\alpha_1 + \theta_2) - w_1(l_1, t) \cos(\alpha_1 + \theta_2)] \\ & + [w_1(l_1, t) \sin(\alpha_1 + \theta_2) + l_1 \cos(\alpha_1 + \theta_2)] \sum_{j=1}^{n_{m,2}} [I_{42j} + m_p \phi_{2j}(l_2)] \delta_{2j}(t) \end{aligned} \right\} \dot{\theta}_1^2 \\
& + \left. \left\{ \begin{aligned} & 2\rho_2 \int_0^{l_2} \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(x_2) \delta_{2j}(t) \right) \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(x_2) \dot{\delta}_{2j}(t) \right) dx_2 \\ & + 2m_p \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(l_2) \delta_{2j}(t) \right) \left(\sum_{j=1}^{n_{m,2}} \phi_{2j}(l_2) \dot{\delta}_{2j}(t) \right) \end{aligned} \right\} \left[\dot{w}_1'(l_1, t) + \dot{\theta}_2 \right] + b_{\theta_2} \dot{\theta}_2 = \tau_2
\end{aligned} \tag{A.6}$$

Deflection equation for j th mode of link 1:

$$\begin{aligned}
 & I_{11j} \ddot{\theta}_1 + \left\{ \begin{aligned} & l_1 (m_2 + m_{h2} + m_p) + \left(\frac{1}{2} m_2 + m_p \right) l_2 \cos(\alpha_1 + \theta_2) \\ & - \sin(\alpha_1 + \theta_2) \sum_{k=1}^{n_{m,2}} [I_{42k} + m_p \phi_{2k}(l_2)] \delta_{2k}(t) \end{aligned} \right\} \phi_{1j}(l_1) \ddot{\theta}_1 \\
 & + \left\{ \begin{aligned} & J_{h2} + J_{o2} + J_p + m_p l_2^2 \\ & + \left(\frac{1}{2} m_2 + m_p \right) l_2 [w_1(l_1, t) \sin(\alpha_1 + \theta_2) + l_1 \cos(\alpha_1 + \theta_2)] \\ & - \left[\begin{aligned} & l_1 \sin(\alpha_1 + \theta_2) \\ & - w_1(l_1, t) \cos(\alpha_1 + \theta_2) \end{aligned} \right] \sum_{k=1}^{n_{m,2}} [I_{42k} + m_p \phi_{2k}(l_2)] \delta_{2k}(t) \\ & + \int_0^{l_2} \rho_2 \left(\sum_{k=1}^{n_{m,2}} \phi_{2k}(x_2) \delta_{2k}(t) \right)^2 dx_2 + m_p \left(\sum_{k=1}^{n_{m,2}} \phi_{2k}(l_2) \delta_{2k}(t) \right)^2 \end{aligned} \right\} \phi_{1j}'(l_1) \ddot{\theta}_1 \\
 & + \left[\begin{aligned} & \left(\frac{1}{2} m_2 + m_p \right) l_2 \cos(\alpha_1 + \theta_2) \\ & - \sin(\alpha_1 + \theta_2) \sum_{k=1}^{n_{m,2}} [I_{42k} + m_p \phi_{2k}(l_2)] \delta_{2k}(t) \end{aligned} \right] \phi_{1j}(l_1) \ddot{\theta}_2 \\
 & + \left[\begin{aligned} & J_{o2} + J_{h2} + J_p + m_p l_2^2 \\ & + \int_0^{l_2} \rho_2 \left(\sum_{k=1}^{n_{m,2}} \phi_{2k}(x_2) \delta_{2k}(t) \right)^2 dx_2 + m_p \left(\sum_{k=1}^{n_{m,2}} \phi_{2k}(l_2) \delta_{2k}(t) \right)^2 \end{aligned} \right] \phi_{1j}'(l_1) \ddot{\theta}_2 \\
 & + \int_0^{l_1} \rho_1 \phi_{1j}(x_1) \sum_{k=1}^{n_{m,1}} \phi_{1k}(x_1) \ddot{\delta}_{1k}(t) dx_1 \\
 & + \left\{ \begin{aligned} & (m_2 + m_{h2} + m_p) \phi_{1j}(l_1) \\ & + \phi_{1j}'(l_1) \left[\begin{aligned} & \left(\frac{1}{2} m_2 + m_p \right) l_2 \cos(\alpha_1 + \theta_2) \\ & - \sin(\alpha_1 + \theta_2) \sum_{k=1}^{n_{m,2}} [I_{42k} + m_p \phi_{2k}(l_2)] \delta_{2k}(t) \end{aligned} \right] \end{aligned} \right\} \sum_{k=1}^{n_{m,1}} \phi_{1k}(l_1) \ddot{\delta}_{1k}(t) \\
 & + \left[\begin{aligned} & \left(\frac{1}{2} m_2 + m_p \right) l_2 \cos(\alpha_1 + \theta_2) \\ & - \sin(\alpha_1 + \theta_2) \sum_{k=1}^{n_{m,2}} [I_{42k} + m_p \phi_{2k}(l_2)] \delta_{2k}(t) \end{aligned} \right] \phi_{1j}(l_1) \sum_{k=1}^{n_{m,1}} \phi_{1k}'(l_1) \ddot{\delta}_{1k}(t)
 \end{aligned}$$

$$\begin{aligned}
& + \left[\begin{aligned} & J_{o2} + J_{h2} + J_p + m_p l_2^2 \\ & + \int_0^{l_2} \rho_2 \left(\sum_{k=1}^{n_{m,2}} \phi_{2k}(x_2) \delta_{2k}(t) \right)^2 dx_2 \\ & + m_p \left(\sum_{k=1}^{n_{m,2}} \phi_{2k}(l_2) \delta_{2k}(t) \right)^2 \end{aligned} \right] \phi_{1j}'(l_1) \sum_{k=1}^{n_{m,1}} \phi_{1k}'(l_1) \ddot{\delta}_{1k}(t) \\
& + \cos(\alpha_1 + \theta_2) \phi_{1j}(l_1) \sum_{k=1}^{n_{m,2}} [I_{42k} + m_p \phi_{2k}(l_2)] \ddot{\delta}_{2k}(t) \\
& + \phi_{1j}'(l_1) \sum_{k=1}^{n_{m,2}} [I_{12k} + J_p \phi_{2k}'(l_2) + m_p l_2 \phi_{2k}(l_2)] \ddot{\delta}_{2k}(t) \\
& + (EI)_1 \int_0^{l_1} \phi_{1j}''(x_1) \sum_{k=1}^{n_{m,1}} \phi_{1k}''(x_1) \delta_{1k}(t) dx_1 \\
& - \left\{ \begin{aligned} & (m_2 + 2m_p) l_2 \sin(\alpha_1 + \theta_2) \dot{w}_1'(l_1, t) \\ & + 2 \sin(\alpha_1 + \theta_2) \sum_{k=1}^{n_{m,2}} [I_{42k} + m_p \phi_{2k}(l_2)] \dot{\delta}_{2k}(t) \\ & + 2 \cos(\alpha_1 + \theta_2) \dot{w}_1'(l_1, t) \sum_{k=1}^{n_{m,2}} [I_{42k} + m_p \phi_{2k}(l_2)] \delta_{2k}(t) \end{aligned} \right\} \phi_{1j}(l_1) \dot{\theta}_1 \\
& + \left\{ \begin{aligned} & (m_2 + 2m_p) l_2 \sin(\alpha_1 + \theta_2) \dot{w}_1(l_1, t) \\ & + 2 \cos(\alpha_1 + \theta_2) \dot{w}_1(l_1, t) \sum_{k=1}^{n_{m,2}} [I_{42k} + m_p \phi_{2k}(l_2)] \delta_{2k}(t) \\ & + 2 \int_0^{l_2} \rho_2 \left(\sum_{k=1}^{n_{m,2}} \phi_{2k}(x_2) \delta_{2k}(t) \right) \left(\sum_{k=1}^{n_{m,2}} \phi_{2k}(x_2) \dot{\delta}_{2k}(t) \right) dx_2 \\ & + 2m_p \left(\sum_{k=1}^{n_{m,2}} \phi_{2k}(l_2) \delta_{2k}(t) \right) \left(\sum_{k=1}^{n_{m,2}} \phi_{2k}(l_2) \dot{\delta}_{2k}(t) \right) \end{aligned} \right\} \phi_{1j}'(l_1) \dot{\theta}_1
\end{aligned}$$

$$\begin{aligned}
& - \left\{ \begin{aligned} & \left[\begin{aligned} & (m_2 + 2m_p)l_2 \sin(\alpha_1 + \theta_2) \dot{w}_1'(l_1, t) \\ & + 2 \sin(\alpha_1 + \theta_2) \sum_{k=1}^{n_{m,2}} [I_{42k} + m_p \phi_{2k}(l_2)] \dot{\delta}_{2k}(t) \\ & + 2 \cos(\alpha_1 + \theta_2) \dot{w}_1'(l_1, t) \sum_{k=1}^{n_{m,2}} [I_{42k} + m_p \phi_{2k}(l_2)] \delta_{2k}(t) \end{aligned} \right] \phi_{1j}(l_1) \\ & \left[\begin{aligned} & 2 \int_0^{l_2} \rho_2 \left(\sum_{k=1}^{n_{m,2}} \phi_{2k}(x_2) \delta_{2k}(t) \right) \left(\sum_{k=1}^{n_{m,2}} \phi_{2k}(x_2) \dot{\delta}_{2k}(t) \right) dx_2 \\ & + 2m_p \left(\sum_{k=1}^{n_{m,2}} \phi_{2k}(l_2) \delta_{2k}(t) \right) \left(\sum_{k=1}^{n_{m,2}} \phi_{2k}(l_2) \dot{\delta}_{2k}(t) \right) \end{aligned} \right] \phi_{1j}'(l_1) \end{aligned} \right\} \dot{\theta}_2 \\
& - \left\{ \begin{aligned} & (m_2 + 2m_p)l_2 \sin(\alpha_1 + \theta_2) \\ & + 2 \cos(\alpha_1 + \theta_2) \sum_{k=1}^{n_{m,2}} [I_{42k} + m_p \phi_{2k}(l_2)] \delta_{2k}(t) \end{aligned} \right\} \phi_{1j}(l_1) \dot{\theta}_1 \dot{\theta}_2 \\
& - \left\{ \begin{aligned} & \int_0^{l_1} \rho_1 \phi_{1j}(x_1) \sum_{k=1}^{n_{m,1}} \phi_{1k}(x_1) \delta_{1k}(t) dx_1 \\ & + \left[\begin{aligned} & (m_2 + m_{h2} + m_p) w_1(l_1, t) + \left(\frac{1}{2} m_2 + m_p \right) l_2 \sin(\alpha_1 + \theta_2) \\ & + \cos(\alpha_1 + \theta_2) \sum_{k=1}^{n_{m,2}} [I_{42k} + m_p \phi_{2k}(l_2)] \delta_{2k}(t) \end{aligned} \right] \phi_{1j}(l_1) \\ & + \left[\begin{aligned} & \left(\frac{1}{2} m_2 + m_p \right) l_2 [l_1 \sin(\alpha_1 + \theta_2) - w_1(l_1, t) \cos(\alpha_1 + \theta_2)] \\ & + [w_1(l_1, t) \sin(\alpha_1 + \theta_2) + l_1 \cos(\alpha_1 + \theta_2)] \sum_{k=1}^{n_{m,2}} [I_{42k} + m_p \phi_{2k}(l_2)] \delta_{2k}(t) \end{aligned} \right] \phi_{1j}'(l_1) \end{aligned} \right\} \dot{\theta}_1^2 \\
& - \left\{ \begin{aligned} & \left(\frac{1}{2} m_2 + m_p \right) l_2 \sin(\alpha_1 + \theta_2) \\ & + \cos(\alpha_1 + \theta_2) \sum_{k=1}^{n_{m,2}} [I_{42k} + m_p \phi_{2k}(l_2)] \delta_{2k}(t) \end{aligned} \right\} \phi_{1j}(l_1) \dot{\theta}_2^2 \\
& - \left\{ \begin{aligned} & \left(\frac{1}{2} m_2 + m_p \right) l_2 \sin(\alpha_1 + \theta_2) \\ & + \cos(\alpha_1 + \theta_2) \sum_{k=1}^{n_{m,2}} [I_{42k} + m_p \phi_{2k}(l_2)] \delta_{2k}(t) \end{aligned} \right\} \phi_{1j}(l_1) [\dot{w}_1'(l_1, t)]^2
\end{aligned}$$

$$- \left\{ \begin{array}{l} 2 \sin(\alpha_1 + \theta_2) \phi_{1j}(l_1) \sum_{k=1}^{n_{m,2}} [I_{42k} + m_p \phi_{2k}(l_2)] \dot{\delta}_{2k}(t) \\ \left[2 \int_0^{l_2} \rho_2 \left(\sum_{k=1}^{n_{m,2}} \phi_{2k}(x_2) \delta_{2k}(t) \right) \left(\sum_{k=1}^{n_{m,2}} \phi_{2k}(x_2) \dot{\delta}_{2k}(t) \right) dx_2 \right. \\ \left. + 2m_p \left(\sum_{k=1}^{n_{m,2}} \phi_{2k}(l_2) \delta_{2k}(t) \right) \left(\sum_{k=1}^{n_{m,2}} \phi_{2k}(l_2) \dot{\delta}_{2k}(t) \right) \right] \phi_{1j}'(l_1) \end{array} \right\} \dot{w}_1'(l_1, t) + b_{1j} \dot{\delta}_{1j} = 0 \quad (A.7)$$

Deflection equation for j th mode of link 2:

$$\begin{aligned} & \left\{ I_{12j} + J_p \phi_{2j}'(l_2) + m_p l_2 \phi_{2j}(l_2) + \left[\begin{array}{l} w_1(l_1, t) \sin(\alpha_1 + \theta_2) \\ + l_1 \cos(\alpha_1 + \theta_2) \end{array} \right] [I_{42j} + m_p \phi_{2j}(l_2)] \right\} \ddot{\theta}_1 \\ & + [I_{12j} + J_p \phi_{2j}'(l_2) + m_p l_2 \phi_{2j}(l_2)] \ddot{\theta}_2 + \cos(\alpha_1 + \theta_2) [I_{42j} + m_p \phi_{2j}(l_2)] \sum_{k=1}^{n_{m,1}} \phi_{1k}(l_1) \ddot{\delta}_{1k}(t) \\ & + [I_{12j} + J_p \phi_{2j}'(l_2) + m_p l_2 \phi_{2j}(l_2)] \sum_{k=1}^{n_{m,1}} \phi_{1k}'(l_1) \ddot{\delta}_{1k}(t) + \int_0^{l_2} \rho_2 \phi_{2j}(x_2) \sum_{k=1}^{n_{m,2}} \phi_{2k}(x_2) \ddot{\delta}_{2k}(t) dx_2 \\ & + m_p \phi_{2j}(l_2) \sum_{k=1}^{n_{m,2}} \phi_{2k}(l_2) \ddot{\delta}_{2k}(t) + J_p \phi_{2j}'(l_2) \sum_{k=1}^{n_{m,2}} \phi_{2k}'(l_2) \ddot{\delta}_{2k}(t) \\ & + (EI)_2 \int_0^{l_2} \phi_{2j}''(x_2) \sum_{k=1}^{n_{m,2}} \phi_{2k}''(x_2) \delta_{2k}(t) dx_2 \\ & - 2 \left\{ \begin{array}{l} \theta_2 \left[\int_0^{l_2} \rho_2 \phi_{2j}(x_2) \sum_{k=1}^{n_{m,2}} \phi_{2k}(x_2) \delta_{2k}(t) dx_2 + m_p \phi_{2j}(l_2) \sum_{k=1}^{n_{m,2}} \phi_{2k}(l_2) \delta_{2k}(t) \right] \\ - [I_{42j} + m_p \phi_{2j}(l_2)] \dot{w}_1(l_1, t) \sin(\alpha_1 + \theta_2) \\ + \dot{w}_1'(l_1, t) \left[\int_0^{l_2} \rho_2 \phi_{2j}(x_2) \sum_{k=1}^{n_{m,2}} \phi_{2k}(x_2) \delta_{2k}(t) dx_2 + m_p \phi_{2j}(l_2) \sum_{k=1}^{n_{m,2}} \phi_{2k}(l_2) \delta_{2k}(t) \right] \end{array} \right\} \dot{\theta}_1 \\ & + \left\{ \begin{array}{l} - \int_0^{l_2} \rho_2 \phi_{2j}(x_2) \sum_{k=1}^{n_{m,2}} \phi_{2k}(x_2) \delta_{2k}(t) dx_2 - m_p \phi_{2j}(l_2) \sum_{k=1}^{n_{m,2}} \phi_{2k}(l_2) \delta_{2k}(t) \\ + [I_{42j} + m_p \phi_{2j}(l_2)] [l_1 \sin(\alpha_1 + \theta_2) - w_1(l_1, t) \cos(\alpha_1 + \theta_2)] \end{array} \right\} \dot{\theta}_1^2 \\ & - [\dot{w}_1'(l_1, t) + \dot{\theta}_2]^2 \left[\begin{array}{l} \int_0^{l_2} \rho_2 \phi_{2j}(x_2) \sum_{k=1}^{n_{m,2}} \phi_{2k}(x_2) \delta_{2k}(t) dx_2 \\ + m_p \phi_{2j}(l_2) \sum_{k=1}^{n_{m,2}} \phi_{2k}(l_2) \delta_{2k}(t) \end{array} \right] + b_{2j} \dot{\delta}_{2j} = 0 \quad (A.8) \end{aligned}$$

Appendix B

Assembling of the Prototype FLMS

B.1 Mechanical Assembling of the FLMS

A sketch of the final CAD model of the prototype flexible link manipulator system (FLMS) developed in our laboratory is shown in Figure B.1. The assembly drawings of joint 1 and joint 2 of FLMS are shown in Figure B.2 and Figure B.3, respectively.

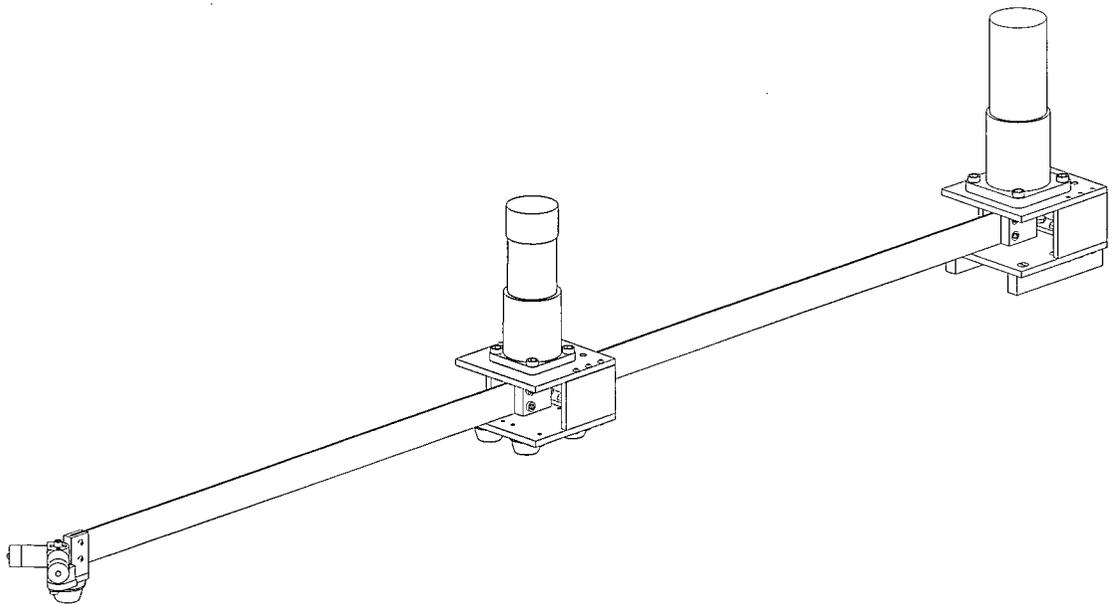


Figure B.1 Overall CAD model of the FLMS prototype.

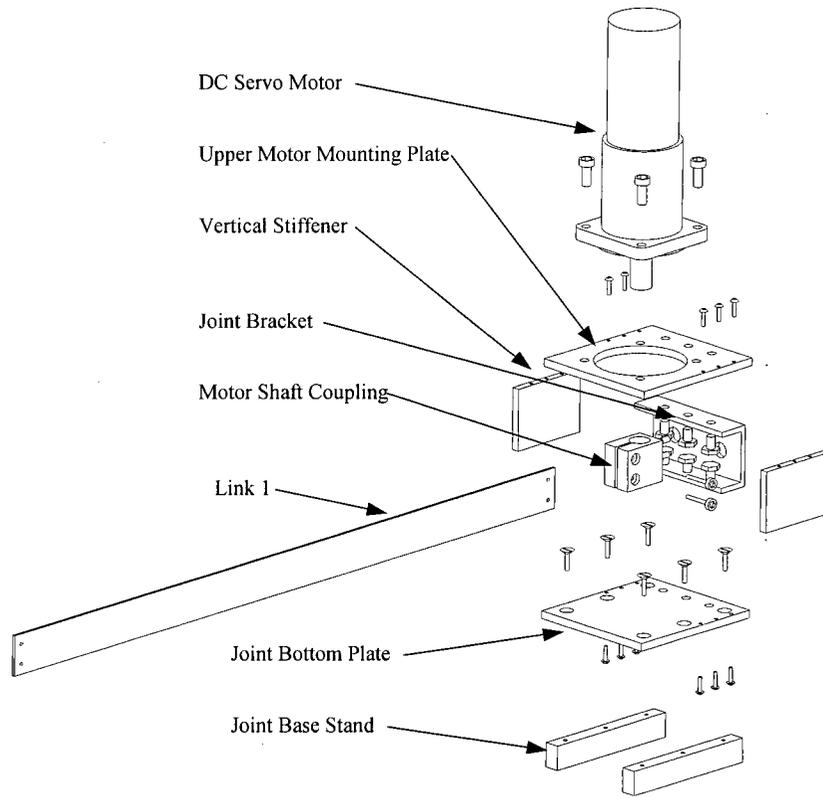


Figure B.2 Assembly drawing of joint module 1.

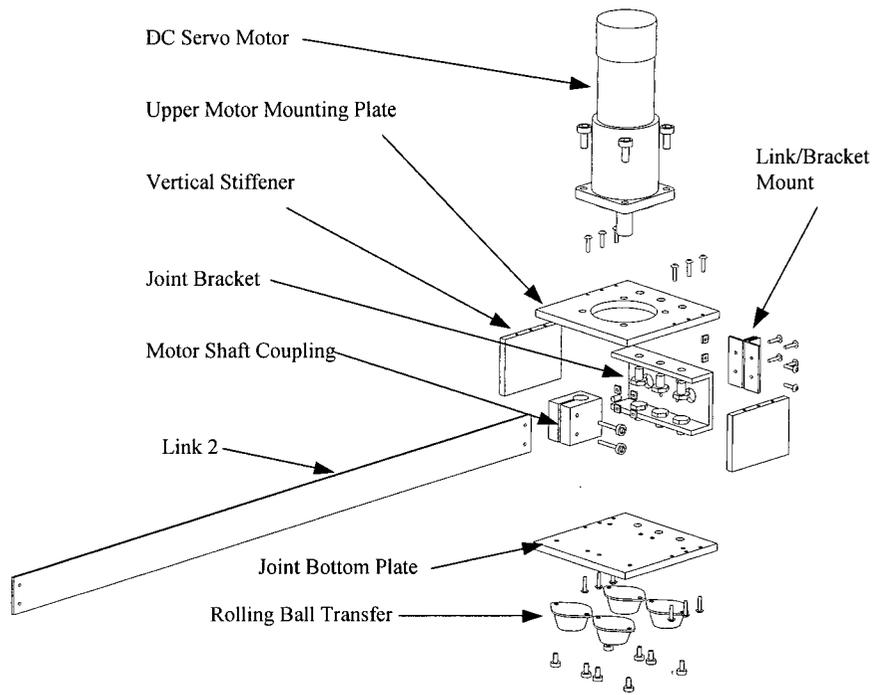


Figure B.3 Assembly drawing of joint module 2.

B.2 Electrical Wiring of the FLMS

The power cables and signal wires of the FLMS are connected according to the diagram given in Figure B.4.

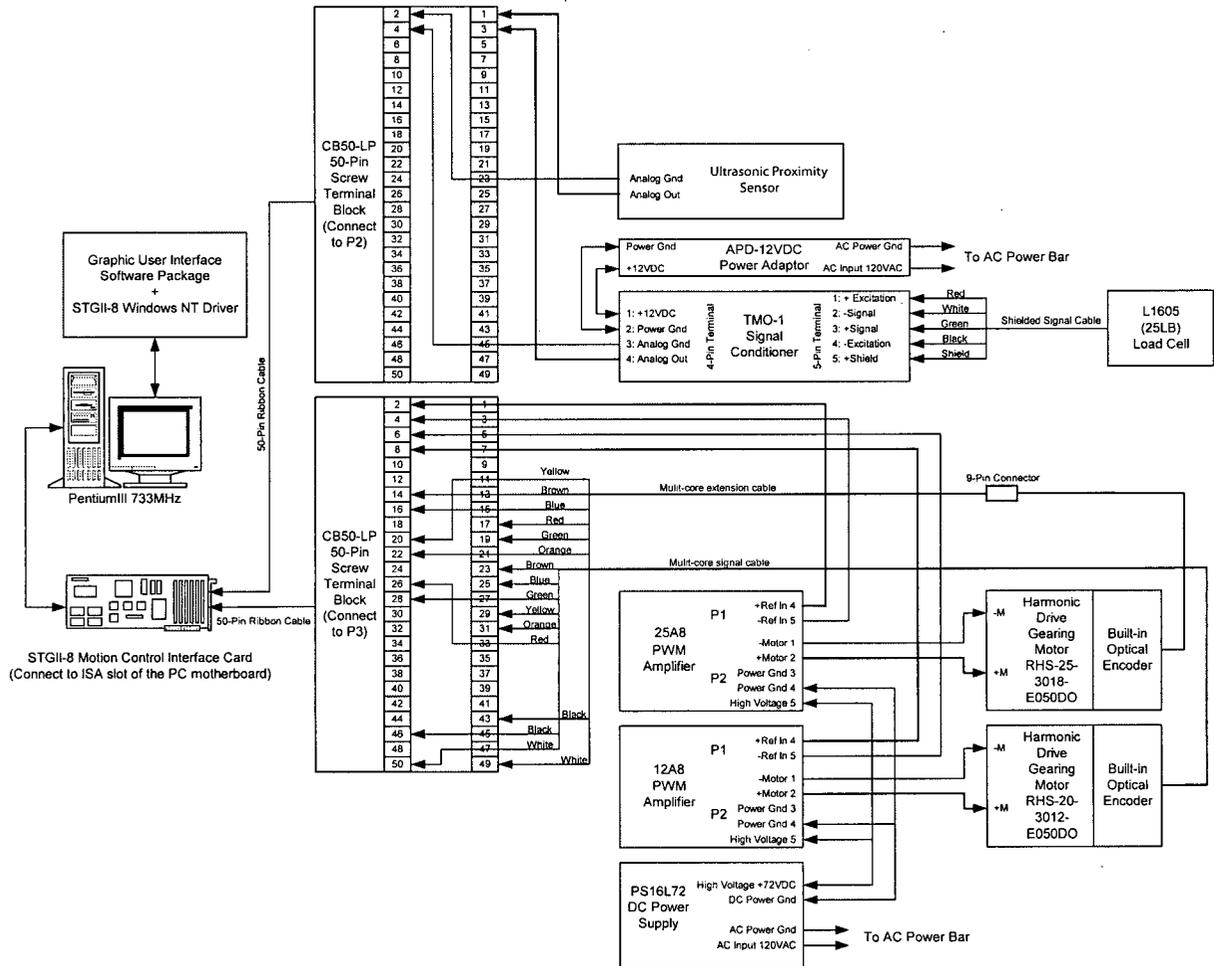
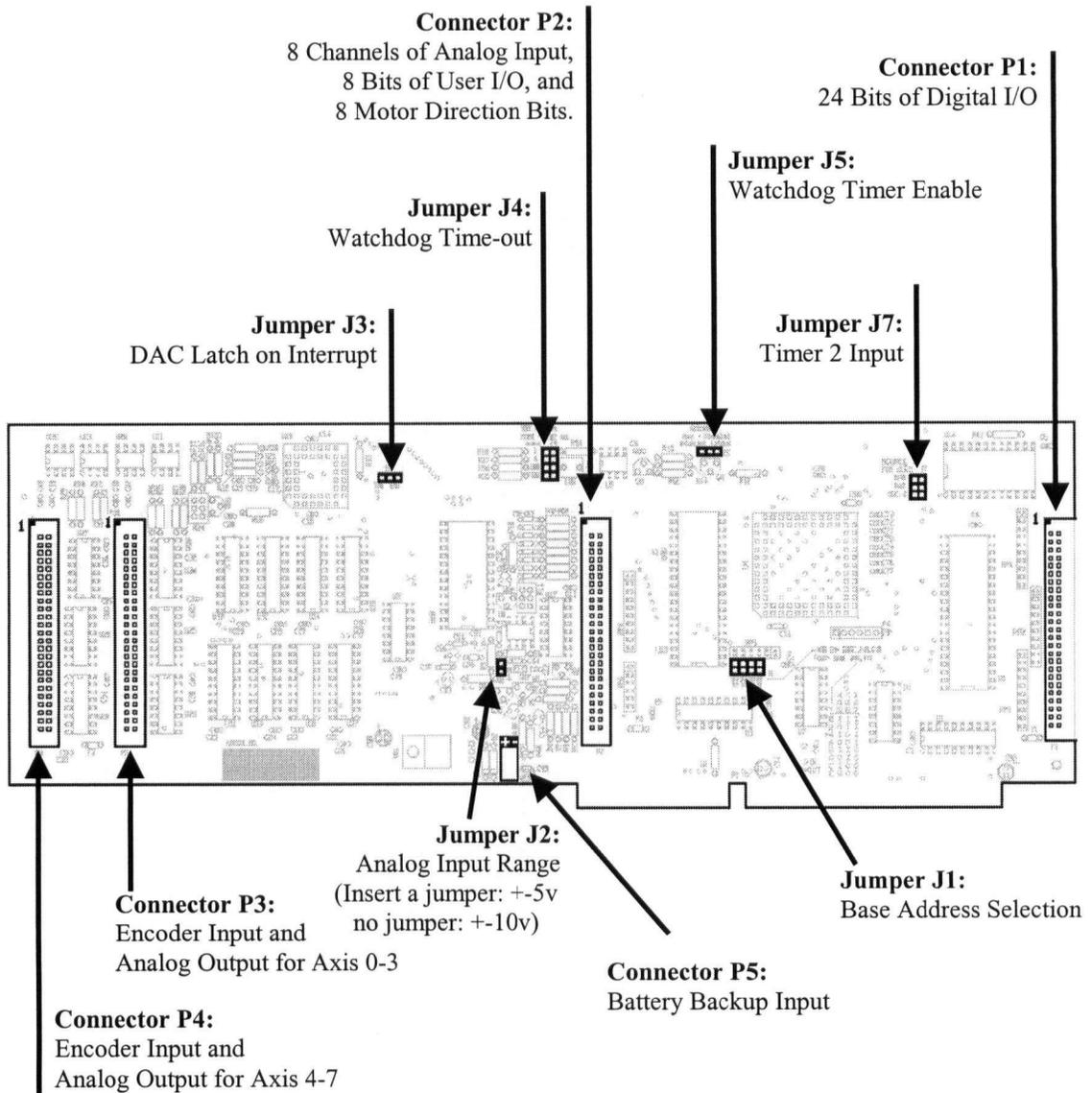


Figure B.4 Electrical wiring diagram of the prototype manipulator system.

Appendix C

Installation Guide for Motion Control Interface Card Hardware

C.1 Jumper and Connector Locations



C.2 Connections

8 Channel Encoder Input and Analog Output Connectors

Connector P3, Motion I/O Axis 0-3			
Pin	Name	Pin	Name
1	Analog Gnd	2	DAC 0
3	Analog Gnd	4	Analog Gnd
5	DAC 2	6	Analog Gnd
7	Analog Gnd	8	DAC 1
9	Analog Gnd	10	Analog Gnd
11	DAC 3	12	Analog Gnd
13	Gnd	14	A 0 +
15	A 0 -	16	Gnd
17	B 0 +	18	B 0 -
19	Gnd	20	I 0 +
21	I 0 -	22	Gnd
23	A 1 +	24	A 1 -
25	Gnd	26	B 1 +
27	B 1 -	28	Gnd
29	I 1 +	30	I 1 -
31	Gnd	32	A 2 +
33	A 2 -	34	Gnd
35	B 2 +	36	B 2 -
37	Gnd	38	I 2 +
39	I 2 -	40	Gnd
41	A 3 +	42	A 3 -
43	Gnd	44	B 3 +
45	B 3 -	46	Gnd
47	I 3 +	48	I 3 -
49	+5	50	+5

Connector P4, Motion I/O Axis 4-7			
Pin	Name	Pin	Name
1	Analog Gnd	2	DAC 4
3	Analog Gnd	4	Analog Gnd
5	DAC 6	6	Analog Gnd
7	Analog Gnd	8	DAC 5
9	Analog Gnd	10	Analog Gnd
11	DAC 7	12	Analog Gnd
13	Gnd	14	A 4 +
15	A 4 -	16	Gnd
17	B 4 +	18	B 4 -
19	Gnd	20	I 4 +
21	I 4 -	22	Gnd
23	A 5 +	24	A 5 -
25	Gnd	26	B 5 +
27	B 5 -	28	Gnd
29	I 5 +	30	I 5 -
31	Gnd	32	A 6 +
33	A 6 -	34	Gnd
35	B 6 +	36	B 6 -
37	Gnd	38	I 6 +
39	I 6 -	40	Gnd
41	A 7 +	42	A 7 -
43	Gnd	44	B 7 +
45	B 7 -	46	Gnd
47	I 7 +	48	I 7 -
49	+5	50	+5

32 Bit Digital I/O, 8 Channel Analog Input and Sign Bit Output Connectors

Connector P1, Digital I/O			
Pin	Name	Pin	Name
1	Opto-23, C7	2	Gnd
3	Opto-22, C6	4	Gnd
5	Opto-21, C5	6	Gnd
7	Opto-20, C4	8	Gnd
9	Opto-19, C3	10	Gnd
11	Opto-18, C2	12	Gnd
13	Opto-17, C1	14	Gnd
15	Opto-16, C0	16	Gnd
17	Opto-15, B7	18	Gnd
19	Opto-14, B6	20	Gnd
21	Opto-13, B5	22	Gnd
23	Opto-12, B4	24	Gnd
25	Opto-11, B3	26	Gnd
27	Opto-10, B2	28	Gnd
29	Opto-9, B1	30	Gnd
31	Opto-8, B0	32	Gnd
33	Opto-7, A7	34	Gnd
35	Opto-6, A6	36	Gnd
37	Opto-5, A5	38	Gnd
39	Opto-4, A4	40	Gnd
41	Opto-3, A3	42	Gnd
43	Opto-2, A2	44	Gnd
45	Opto-1, A1	46	Gnd
47	Opto-0, A0	48	Gnd
49	+5V	50	Gnd

Connector P2, Analog & Digital I/O			
Pin	Name	Pin	Name
1	ADC Chan 0	2	Analog Gnd
3	ADC Chan 1	4	Analog Gnd
5	ADC Chan 2	6	Analog Gnd
7	ADC Chan 3	8	Analog Gnd
9	ADC Chan 4	10	Analog Gnd
11	ADC Chan 5	12	Analog Gnd
13	ADC Chan 6	14	Analog Gnd
15	ADC Chan 7	16	Analog Gnd
17	Opto-15, D7	18	Gnd
19	Opto-14, D6	20	Gnd
21	Opto-13, D5	22	Gnd
23	Opto-12, D4	24	Gnd
25	Opto-11, D3	26	Gnd
27	Opto-10, D2	28	Gnd
29	Opto-9, D1	30	Gnd
31	Opto-8, D0	32	Gnd
33	IN2	34	Gnd
35	IN1	36	Gnd
37	IN0	38	Gnd
39	EXLATCH	40	Gnd
41	T2GATE	42	Gnd
43	TA2	44	Gnd
45	/WATCHDOG	46	Gnd
47	NC	48	Gnd
49	+5V	50	Gnd

Connector P1 can be directly connected to a 24 channel I/O card, such as made by OPTO-22, and other companies.