DYNAMICS AND CONTROL OF LIGHT WEIGHT FLEXIBLE MANIPULATORS

A THESIS

**Submitted to the School of Graduate Studies and Research
and to the Ottawa Carleton Institute of Mechanical and Aerospace Engineering
in partial fulfillment of the requirements
for the Degree of
Doctor of Philosophy**

By

Amor Jnifene

University of Ottawa
Ottawa, Ontario, Canada

National Library of Canada

Acquisitions and Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada

Direction des acquisitions et des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

ISBN 0-612-15726-1

Canada

Abstract

Current robot manipulators are controlled using methods that do not take into account the links and joints flexibility. Consequently, tasks are usually carried in a slow fashion. The objective of this research is to improve the performance of light weight flexible manipulators by developing dynamic models and control strategies that include the effects of the links flexibility.

Toward achieving this objective, the dynamic model of a single link flexible manipulator has been developed. Two hermit cubic functions are used in conjunction with the deflection and the slope at the tip of the link in order to express the deflection at any point along the link. The model is used to simulate the response of the single link manipulator and its ability to follow various trajectories under collocated control. Even though collocated control such as the computed torque method leads to asymptotic stability in the case of rigid link manipulators, the end point of the single link flexible manipulator has shown sustained oscillations under this type of control. In order to improve the performance of the single link and reduce the tip deflection, dynamic stiffening has been added to the link by using the angular position of a point along the link as the controlled variable. This control method is referred to as the non collocated

control. Simulations of the robot motion under non collocated control has shown a significant reduction in the position error of the end point.

In order to increase the damping of the flexible link, a new approach referred to as the Computed Torque/Delayed Deflection (CTDD) has been used. The CTDD approach combines the non collocated control with a delayed deflection at the point being used for non collocation feedback. The fastest decay in the end point vibrations was obtained when the delay time is selected to be half the period of the lowest frequency in the system.

The CTDD has been extended to the case of a two link flexible manipulator. Simulations results for the tracking performance of the two link flexible manipulator have been carried out using two paths: A straight line path and a four points path. The advantages of using the CTDD approach are outlined by comparing its performance to those of a collocated control (computed torque method) and the non collocated control.

Acknowledgements

First, I would like to express my deepest gratitude to Professor Atef Fahim, my supervisor, for his continuous enthusiastic support and guidance over the course of this research. I also wish to thank the member of the advisory committee, Professor J. Z. Sasiadek and Professor D. Nesculescu for their valuables advises. Thanks also to Professor D. Gorman and Professor M. Munro who have taken the time to help me on many occasions.

I wish also to thank the administrative staff in the department of Mechanical Engineering at the University of Ottawa. In particular, I thank Mrs Solange Amyot, who was always ready to deal with my last minute administrative requests.

I am also indebted to my friends Mohsen, Sonia J., Sadok, Sonia R., Lymeim, Kilani, Brahim and Youcef and Fawzi for their continuous moral support.

Finally, I would like to thank my parents, Abdallah and Hafsyia, my brother Miloud and my sisters, Hana, Najet, Zohra, Dhaoua and Fawzia for their love and support through all these years. Special thanks goes to my wife Raja for her understanding and support for the past two years.

Table of Contents

List of Figures

List of Tables

Nomenclature

Roman Symbols

One Link Flexible Manipulator

A:	System matrix.
a:	Joint acceleration.
e:	Error between the desired and the actual joint positions.
EI:	Flexure rigidity of the link.
H_r:	Non-linear term associated with the rigid body motion
H_f:	2 by 1 matrix representing the non-linear terms associated with the flexible body motion.
I:	2 by 2 Identity matrix.
I_L:	Moment of inertia of the payload.
K_p :	Proportional controller gain.
K_d :	Derivative controller gain.
l:	Length of the link.
M:	3 by 3 Generalized Inertia Matrix (GIM).
M_r:	Inertia term associated with the rigid body motion.
M_f:	2 by 2 matrix representing the inertia associated with the flexible body

motion.

M_x: 2 by 2 matrix representing the inertia associated with the flexible body motion under non collocation control.

M_{rf}: 2 by 1 matrix representing the coupling between the rigid body motion and the flexible body motion.

M_L: GIM of the payload.

m: Linear mass density of the link.

m_L: Mass of the payload.

OXY: Inertial coordinate system.

oxy: Link coordinate system.

R_0^1: Transformation matrix from oxy to OXY.

q: Generalized coordinates vector.

q_d: Desired coordinates vector.

r: Absolute position vector of a point along the link.

r_e: Absolute position of a payload at the tip of the link.

r_{oxy}: Position vector of a point along the link with respect to oxy.

T: Kinetic energy.

T_L : Kinetic energy associated with the payload.

t: Time.

$u(x,t)$: Deflection of the link with respect to the link coordinate system.

Two Link Flexible Manipulator

E_{total}: Total position error of the end point of the two link flexible manipulator

E_x: Position error of the end point of the two link flexible manipulator along the x direction

E_y: Position error of the end point of the two link flexible manipulator along the y direction.

EI_1: Flexure rigidity of the first link.

EI_2: Flexure rigidity of the second link.

G_f : 2 by 1 matrix representing the stiffness terms.

$\boldsymbol{H_r}$: Non-linear term associated with the rigid body motion of the two link manipulator.

$\boldsymbol{H_f}$: Non-linear term associated with the flexible body motion.

I_a: Moment of inertia of a payload at the tip of the first link.

I_L: Moment of inertia of a payload at the tip of the second link.

$\hat{I}_1$: Highest inertia with respect to the axis of the joint of the first link.

$\hat{I}_2$: Highest inertia with respect to the axis of the joint of the second link.

J: Jacobian matrix.

$\boldsymbol{K_P}$: 2 by 2 diagonal matrix: Proportional gains.

$\boldsymbol{K_D}$: 2 by 2 diagonal matrix: Derivative gains.

K_{P_1}: Proportional gain for the joint of the first link.

K_{P_2}: Proportional gain for the joint of the second link.

K_{D_1}: Derivative gain for the joint of the first link.

K_{D_2}: Derivative gain for the joint of the second link.

L : Lagrangian.

l_i: Length of link i.

M : 6 by 6 matrix: GIM of the two link manipulator.

M_a: 6 by 6 matrix: GIM of a load at the tip of the first link.

M_f: 4 by 4 matrix representing the inertia terms associated with the flexible body motion alone.

M_i: 6 by 6 matrix: GIM of link i.

M_L: 6 by 6 matrix: GIM of the payload.

M_r: 2 by 2 matrix representing the inertia terms associated with the rigid body motion.

M_{rf}: 4 by 2 matrix representing the coupling between the rigid body motion and the flexible body motion.

m_1: Linear mass density of the first link.

m_2: Linear mass density of the second link.

m_L: Mass of the payload.

OXY: Inertial coordinate system.

$o_ix_iy_i$: Coordinate system of link i.

P_i: Initial position of the end point in the xy plane.

P_f: Final position of the end point in the xy plane.

q: Generalized coordinates vector.

$\boldsymbol{R}_0^1$:	Transformation matrix from $o_1x_1y_1$ to OXY.
$\boldsymbol{R}_1^2$:	Transformation matrix from $o_2x_2y_2$ to $o_1x_1y_1$.
$\boldsymbol{r}$:	Absolute position vector of a point along the link.
$\boldsymbol{r}_{e_1}$:	Absolute position of a payload at the tip of the fist link.
$\boldsymbol{r}_{e_2}$:	Absolute position of a payload at the tip of the second link.
$\boldsymbol{r}_{o_1/o_2}$:	Position vector of o_2 with respect to $o_1x_1y_1$.
$\boldsymbol{r}_{oxy}$:	Position vector of a point along the link with respect to oxy.
$\boldsymbol{T}$:	Total kinetic energy.
$\boldsymbol{T}_a$:	Kinetic energy associated with a payload at the tip of the first link.
$\boldsymbol{T}_L$:	Kinetic energy associated with a payload at the tip of the second link.
t:	Time.
$\boldsymbol{U}$:	Potential energy.
$\boldsymbol{U}_1$:	Potential energy of the fist link.
$\boldsymbol{U}_2$:	Potential energy of the second link.
$u_1(x,t)$:	Deflection of the first link with respect to $o_1x_1y_1$.
$u_2(x,t)$:	Deflection of the second link with respect to $o_2x_2y_2$.
x_e:	x position of the end point of the second link with respect to OXY.
y_e:	y position of the end point of the second link with respect to OXY.

Greek Symbols

One Link Flexible Manipulator

α_i:	Coefficient.
Γ_0:	Inertia term of the flexible link under collocated control.
Γ_x:	Inertia term of the flexible link under non collocated control.
δ_e:	Slope at the end point with respect to the rigid link.
ζ:	Damping ratio.
η:	2 by 1 vector representing the coordinates of the flexible sub-system
θ:	Joint angle
θ_d, $\dot{\theta}_d$, $\ddot{\theta}_d$:	Desired position, velocity and acceleration of the joint angle
θ_f:	Final joint position
θ_i:	Initial joint position
θ_x:	Angle of a vector describing the position of a point on the link
$\mathbf{K}$:	2 by 2 stiffness matrix.
Λ_0:	Non linear term in the flexible link under collocated control.
Λ_x:	Non linear term in the flexible link under non collocated control.
ν:	Delay time
ξ_e:	Deflection at the end point of the flexible link.
σ:	Standard deviation for the trajectory with with Gaussian velocity profile
τ:	Actuator's torque.

Φ: Vector respresenting the shape functions.

$\phi_1(x)$: Shape function associated with the deflection at the tip of the link.

$\phi_2(x)$: Shape function associated with the slope at the of link.

ω_n: Natural frequency

ω_e: Angular velocity at the tip of the flexible link.

Two Link Flexible Manipulator

$\boldsymbol{\Gamma}_0$: 2 by 2 matrix representing the inertia of the two link flexible manipulator under collocated control.

$\boldsymbol{\Gamma}_x$: 2 by 2 matrix representing the inertia of the two link flexible manipulator under non collocated control.

δ_{e_1}: Slope of the end point of the first link.

δ_{e_2}: Slope of the end point of the second link.

ζ_1 : Damping ratio for the joint motion of the first link.

ζ_2 : Damping ratio for the joint motion of the second link.

η : 4 by 1 vector representing the coordinates of the flexible sub-system.

η_{e_1}: 2 by 1 vector representing the deflection and the slope at the tip of the first link.

η_{e_2}: 2 by 1 vector representing the deflection and the slope at the tip of the second link.

$\Theta=[\theta_1 \quad \theta_2]^T$: Actual joint position vector.

$\Theta_d=[\theta_{1d} \quad \theta_{2d}]^T$: Desired joint position vector.

$\mathbf{K}$: 4 by 4 stiffness matrix.

K_1: 2 by 2 stiffness matrix of the first link.

K_2: 2 by 2 stiffness matrix of second link.

$\boldsymbol{\Lambda}_0$: 2 by 1 vector representing the non linear terms in the two link flexible manipulator under collocated control.

Λ_x: 2 by 1 vector representing the non linear terms in the two link flexible manipulator under non collocated control.

ξ_{e_1}: Deflection at the end point of the first link.

ξ_{e_2}: Deflection at the end point of the second link.

T: 2 by 1 vector of the actuators torques.

τ_1 : Torque of the actuator driving the first link.

τ_2 : Torque of the actuator driving the second link.

Φ: 4 by 1 vector respresenting the shape functions.

Φ_1: 2 by 1 vector respresenting the shape functions $\phi_{11}(x)$ and $\phi_{12}(x)$.

Φ_2: 2 by 1 vector respresenting the shape functions $\phi_{21}(x)$ and $\phi_{22}(x)$.

$\phi_{11}(x)$: Shape function associated with the deflection at the end of the first link.

$\phi_{12}(x)$: Shape function associated with the slope at the end of the first link.

$\phi_{21}(x)$: Shape function associated with the deflection at the end of the second link.

$\phi_{22}(x)$: Shape function associated with the deflection at the end of the first link.

ω_L : Angular velocity of a payload at the tip of the second link.

Chapter 1

General Introduction

1.1 Motivation

Mechanical manipulators, commonly referred to as robots, have been undergoing a very fast evolution from relatively simple systems performing simple tasks to very sophisticated industrial robots performing very demanding tasks both on earth and in space. The use of such mechanical devices is justified primarily by their ability to accurately repeat tedious tasks, and by the ease with which they can be reconfigured or reprogrammed. As the scope of applications of robots becomes wider, the performance requirements become more demanding. These requirements include high operating speeds and having manipulators with long and light weight members capable of handling large payloads. Current industrial robots have a very high ratio of arm weight to load carrying capacity. As an example, for an ASEA RB-6 robot this ratio is in the order of 40:1. This inherent design drawback is a result of the need for rigid links in order to avoid the inaccuracies in positioning caused by flexible links. The heavy weight of the rigid links makes such robots inefficient and not suitable for mobile or space applications. An alternative to the use of stiff links in the design of robot manipulator is to utilize robot controllers that can eliminate or counteract the vibrations that result when the robot is equipped with flexible links.

In the past few years, light weight flexible arms have been receiving the attention of an increasing number of researchers. This attention is directly related to the recent increase in space activities such as the deployment and retrieval of satellites, spacecraft servicing, and equipment transfer. A typical example of these manipulators is the Shuttle Remote Manipulator System (SRMS), also known as the CANADARM. This manipulator was built for the National Aeronautic and Space Administration (NASA) by Spar Aerospace Limited of Canada. To be able to accomplish its objectives in space, the SRMS is 15 meters long when fully extended and capable of handling payloads up to 30,000 kg. The SRMS is made out of light weight composite material and has a weight of about 450 kg. The lowest frequency of oscillation of the SRMS when the joints are locked varies between 0.04 and 0.4 Hz depending on the payload and on the configuration [51]. The control system employed to control the end point of the SRMS uses feedback information from the joint sensors in order to determine the torques of the actuators. The position, velocity and acceleration of the end point are not directly sensed, instead, they are inferred from the joint sensors. The speed of the end point varies between 0.06 m/s when the manipulator is fully loaded and 0.6 m/s with no load.

It is believed that light weight flexible manipulators similar to the SRMS will play a very important role in the construction of the Space Station Alpha. The design of the

Space Station Remote Manipulator System (SSRMS) has been envisioned and planned. The SSRMS will be approximately 17 meters in length with seven degrees of freedom, and like the SRMS, will be capable of handling large and heavy payloads.

The position of the end point in a robot manipulator is, in general, affected by the flexibility at the joints and by the deflections of the links. The contribution of each of the joint flexibility and the link deflection may vary considerably depending on the given robot design. The current heavy weight and rigid industrial robots with their bulky links exhibit a relatively large joint deflection that can seriously affect the system precision. On the other hand, the precision of light weight manipulators such as the SRMS, is severally affected by the deflections of the links.

Manipulators with lightweight flexible arms have many advantages as compared to their rigid counterparts. They require less material, smaller actuators, and lower energy consumption. They are also more maneuverable and more transportable, a feature very useful for space applications. Light weight manipulator are also less hazardous because of their low momentum, and their pliability when they come in contact with obstacles.

Historically, manipulator systems with light weight flexible arms did not find a favorable place in industry due to the fact that the response of the arm's end point is severally affected by the flexibility of the links. Even though the SRMS has been performing reliably for many years, operations are sometimes delayed while the operators wait for the end point oscillations to damp out. It was reported in the literature [1], that a recent study dealing with the initial phases of space station assembly suggested that out of 47 hours of robot operation time, 10 to 15 hours will be spend waiting for the oscillations to settle down.

1.2 Objective

In order to exploit the advantages of light weight manipulators and eliminate the end point oscillations, a sophisticated control system is needed. Also, to develop this control system, a suitable dynamic model including the flexible modes needs to be developed. Once an accurate model of the flexible manipulator is derived, the problem of controlling its end point has to be addressed. The main objective of this work is to contribute to the fundamental understanding of the dynamics and control of flexible manipulators. In the case of manipulators with rigid links, it is assumed that the tip position can be accurately deduced from the joint angle and the knowledge of the length of each link, and consequently, the controller uses the joint angle as the only feedback signal. This assumption is justified in view of the rigidity of the links. For the end point control of manipulators with flexible links, the joint angle cannot be used as the sole feedback signal. Furthermore, link oscillations are reduced either passively by introducing viscous dampers to increase the structural damping or actively by using external actuators and feedback control. Passive vibrations control has gained little support since it requires the addition of physical elements for energy dissipation. Active controllers are limited only by the capacities of the actuators and feedback elements. The position of the end point must be measured and used as another feedback signal. A control method widely advocated for the control of rigid link manipulators is that of computed torque. In this method the system equations are linearized and decoupled through a computed nonlinear feedback. When the same technique is used to control a flexible arm, the linearization property is lost owing to the kinematics and dynamics behavior of the flexible links [22].

1.3 Thesis Outline

The layout of this thesis is as follows. First, an extended review of the literature dealing with the dynamics and control of light weight flexible manipulators is presented in Chapter 2. Due to the large number of research papers dealing with this problem, this review has been divided into three parts: Dynamic modelling, Control, and Implementation issues. Chapter 3 deals with the dynamic modelling of a one link flexible manipulator having a payload at the tip. Some simulation results for the dynamic model are also presented in this chapter. In Chapter 4, the effect of the desired trajectory on the dynamic behavior of the tip of a one link flexible manipulator has been investigated using four types of trajectories. These trajectories have different degrees of smoothness. In this chapter, the effect of using the absolute angle of a point along the link for feedback has also been studied, and the non minimum phase behavior of the system when the absolute angle of the tip is used for feedback has been outlined. In Chapter 5, a new approach using time delay has been introduced as an alternative method for the control of the oscillations at the tip of a one link flexible manipulator. After a brief introduction to time delayed systems in Section 5.1, an overview of the posicast control and the input/output linearization using time delay is presented in Section 5.2. In Section 5.3, the Computed Torque/Delayed Deflection approach (CTDD) is introduced. Simulations results showing the effects of the delay time and the payload on the system's performance are presented in Section 5.3.2. Chapter 6 presents the derivations of the dynamic model of a manipulator having two flexible links. Simulation results for several initial conditions have been carried out and are presented in this Chapter. In Chapter 7, Collocated control is employed in an attempt to control the end point along a specified trajectory. The two collocated control schemes used are the independent joint PD and the computed torque method. Chapter 8 presents the non-collocated control of the end point of the two link flexible manipulator as well as the use of time delay in order to introduce active damping into the system. A comparative study of the collocated control (computed torque), the non-collocated control and the non collocated control with time delay has been carried out and simulation results are presented. Chapter 9 concludes this thesis with an outline of the results reached throughout this work and some recommendations for future research.

Chapter 2

Literature Review

2.1 Introduction

Reported studies on flexible manipulators has been primarily carried out with special emphasis either on the dynamic modelling or on the design of an appropriate control strategy to control the end effector. In this chapter, the literature review is presented in three sections. The first deals with the different techniques and formulations used to develop a dynamic model describing the motion of manipulators with flexible links. The second section deals with the problem of controlling the end effector and the different control techniques adopted in the literature to reduce the vibrations. In the third section, several reported experimental results are discussed.

2.2 Dynamic Modelling of Flexible Link Manipulators

Manipulators with flexible links are continuous dynamical systems represented by an infinite number of degrees of freedom and are governed by a set of nonlinear, coupled, partial differential equations. The exact solution of these equations is usually not feasible. Hence it is necessary to approximate these infinite dimensional systems by a finite number of degrees of freedom representing the deflection of the links. Two methods are commonly used: The assumed modes method and the finite elements method. In the assumed modes method, the deflection of the link is represented by a truncated series in terms of the mode shape functions and the time varying mode amplitudes. The mode shape functions can be obtained by solving the Euler-Bernoulli beam equation for flexural vibrations under a specified link boundary conditions. The clamped-free or pinned free boundary conditions are usually considered. There are several ways to choose the boundary conditions and the mode shapes of a flexible link. In [12], Book showed that clamped boundary conditions leads to a physically measurable joint variables and simpler coefficients of the joint torques. In [2], Asada et al. showed that using a pinned boundary condition reduces the complexity of the dynamic equations and it was found to be very effective in trajectory control. The coordinates system used in Asada's work is referred to as the Virtual Rigid Link System (VRLS). The main disadvantage of this type of coordinates system is that the control variables, the joint angles, are not expressed explicitly in the resulting dynamic equations. It was also reported by Rex J. Theodore and Ashitava Ghosal [73] that if the beam to hub inertia is very low (0.1 or less), the use of clamped boundary conditions gives better results when compared with the pinned boundary conditions. In order to investigate the case of flexible manipulators with revolute and prismatic joint, J. Yuh and T. Young [90], considered an axially moving flexible beam in rotation. the beam was treated as a clamped-free beam with variable length, and the assumed modes method was used to approximate the dynamics of the system. A similar case was considered by K. W. Buffinton [13] where the equations of motion of a two degrees of freedom elastic manipulator with prismatic joint have been derived. The manipulator consists of a uniform elastic beam capable of translating over a rigid support which is attached to a rotating base. It has been found that the use of only two modes is sufficient to describe the flexibility of the beam.

In the finite element method, each flexible link is considered as an assemblage of a finite number of elements. The displacement at any point along these elements is expressed in terms of a finite number of displacements at the nodal points (which are usually located at the two ends of the element) multiplied by the hermit cubics polynomial interpolation functions. The equation of motion of the entire flexible link is then obtained by deriving the equations of motion of the elements and then suitably assembling them. Compared to the assumed modes method, the finite element method proved to be more favourable as shown by Gaultier and Cleghorn [30]. In their work, Gaultier and Cleghorn have developed a finite element model of a spatially translating and rotating beam including the effects of axial forces on lateral deformations. The advantage of the finite element method over the assumed modes method is due in part to the fact that the eigenvalues of the system are not required for the calculation. In contrast, for the assumed modes method, the eigenvalues of the system have to be computed for each configuration of the manipulator. The finite element method can also accommodate links with irregular shapes by discritizing the link into several sections. Ben Jonker [34] presented a finite element based method for analyzing the dynamic behaviour of a flexible manipulator with three degrees of freedom. Both links and joints flexibilities were considered. Results including the effects of the manipulator's control system have been reported. Another finite element model has been reported by Karla and Sharan [35]. In this study, the dynamic equations of a two link flexible manipulator were derived and the effects of including the axial strains as one of the nodal coordinates in the finite element model were outlined. Cubic interpolation functions are used to model the axial and torsional deformations. Lateral deformations are modeled with quintic polynomial interpolation functions.

Similar to the case of rigid link manipulators, Newton-Euler and Lagrangian formulations are generally adopted to derive the dynamic equations of light weight flexible manipulators. When used in a recursive form, Newton-Euler is computationally advantageous especially when the system under consideration has a large number of degrees of freedom. The Lagrangian formulation is usually preferred when a closed form expressions of the dynamic terms are desired. W. J. Book [12], proposed a recursive method to develop a dynamic model for flexible manipulator arms. This method is based on the Lagrangian formulation and on a 4 by 4 homogeneous transformation matrices representing the rigid and flexible motion. The recursive nature of this method gives it a very high efficiency in terms of computation time, however the lack of a closed form representation of the system equations makes it not suitable for control purposes. A generalization of Newton-Euler formulation of the dynamic equations to flexible manipulators is proposed by Y. Hang and C.S. G. Lee [33]. In this formulation, a first order lumped mass/spring model is used. Each link is divided into segments and the parameters of segment i are lumped with the node connecting segment i to segment i-1. The deflection of segment i is modeled by a spring located at node i. This model is expanded to further deal with links exhibiting large deformation. A similar lumped mass model has been derived by V. Feliu et al. [29]. P. B. Usoro et al. [81] used a finite element Lagrangian approach which makes use of the Generalized Inertia Matrix (GIM) concept. Each link of the manipulator is treated as an assemblage of a finite number of elements. For each element, the kinetic and potential energies are derived and then suitably combined to derive the dynamic model of the over all system. An example of two link flexible manipulator with two elements per link is used in the analysis. The resulting dynamic model is extremely non linear and complex. M. Benati and Morro [7], used the Lagrangian approach and the assumed mode method to model a chain of flexible links. Only the first two modes of the clamped beam are considered. In dealing with the problem of the mass boundary condition, each link is assumed to have a mass at its end representing the masses of all the subsequent links. The two modes are expressed in terms of the deflection and the slope of the end point with the mass boundary condition. To illustrate this method, a two link chain is used with only the first link being flexible. The main advantage of the approach presented in this paper is that it leads to explicit dynamic equations in terms of well defined geometrical parameters. The deflection and the slope at the tip of a flexible link were used also by L. W Chang [17] to describe the motion of a flexible link with respect to the Equivalent Rigid Link System (ERLS) that was used to describe the joint motion. The idea of an Equivalent Rigid Link System was first introduced by L. W. Chang and J. F. Hamilton [18,19] in order to describe the kinematics and dynamics of manipulators with flexible links. Experimental validation of a dynamic model based on the Equivalent Rigid Link System has been reported by R. P. Petroka and L. W. Chang [55] and the results seem to be in good agreements with the theoretical ones. In [24], A. De Luca and B. Siciliano derived a closed form dynamic model of a planar two-link flexible arm where each link is modelled as an Euler-Bernouilli beam of uniform density with a payload at the tip of each link. The main problem encountered in this paper is the time varying boundary conditions of the inner link. In order to derive the shape functions describing the deflection of the inner link, it was assumed that the outer link is lumped at the tip of the inner link. Hamiltonian dynamics was also used by K. H. Low and M. Vidyasagar [45] and M. Benati and A Morro [8] in order to express the dynamic equations of manipulators containing both rigid and flexible links. The resulting equations were organized into a state space form for control purposes. Developing the equations of motion of a multi-link flexible manipulator by hand becomes almost impossible as the number of flexible links and the number of degrees of freedom increases. To solve this problem, and with increased computational power, several researchers such as W. J. Book [12] have developed methods that lead to computer generated equations of motion. These methods, in general do not give the equations of motion in a closed form which is usually used in designing a suitable controller. In order to obtain a computer generated equations of motion in a closed form, computer programs such as MAPLE V and MACSYMA are being used. These computer programs are capable of performing symbolic calculations.

2.3 Control of Flexible Manipulators

Most of today's robots use the same control strategies used to control rigid link manipulators. Such control strategies employ only sensors that are collocated with actuators and each joint is typically controlled independently. Thus, to control the end point, the desired position is converted through kinematics into the equivalent angle that each of the robot joints must assume. At each of these joints, a servo-loop made of an actuator along with a collocated sensor is used. These servo loops drive the joints simultaneously to the desired angles through a conventional PID control action. An attempt to use the collocated control method for the position control a chain of flexible robots have been made by E. Bayo [5], E. Bayo and B. Paden [6], A. De Luca and B. Siciliano [23]. Methods similar to the computed torque techniques have been used to control the end point position of an open chain flexible robots. A. Shchuka and A. A. Goldenberg [61], proposed a feedforward technique along with a PD feedback controller to control the tip of a single link flexible arm. This technique uses a pre-computed torque to achieve a dominant pole-zero cancellation. The inherent flexibility of light weight manipulator makes it impossible to achieve high precision when a collocated control scheme is used. In this respect, more efficient and sophisticated controllers are required to control the joint motion as well as to position the end effector. The end point control of flexible link manipulators is a non collocated control problem since the action is usually carried out at the joint while the controlled variable is the position of the end point. The non collocation of actuators and sensors may in fact lead to an unstable system as shown by Dong-Soo Kwon and W. J. Book [43]. In [72], V. A. Spector and H, Flashner investigated the sensitivity of structural models for non collocated control systems. A representative example of noncollocated control systems consisting of a pinned-free beam with end point mass and inertia has been used. The beam is equipped with several sensors which are distributed along its length. M. S. Kang and B. Yang [36] presented a discrete time non collocated control of flexible mechanical systems.

Recently, a new approach for non collocated control of flexible mechanical systems have been reported by B. Yang and C. D. Mote, Jr [85-88]. It has been shown that the destabilizing effect caused by the non collocation of actuators and sensors is eliminated through introduction of specific time delay in the control system. An axially moving flexible string has been used in [85] and in [88] to show that the time delay allows an exact prediction of the system response at one point by measuring the vibration at another point on the flexible string. A similar control scheme referred to as TDC (Time Delay Control) has been reported in the literature. The TDC consists of using delayed feedback signals in order to calculate the present control action. In [76], K. Youcef Toumi and O. Ito have used the TDC in the control of systems with unknown dynamics and unexpected disturbance. It was shown that by using past observation of the system's response and the control input, estimation of the function representing the effect of uncertainties can be achieved. It assumed that all the state variables and estimates of their delayed derivatives are accessible. Sufficient conditions for the stability of uncertain linear systems under time delay control have been presented by K. Youcef Toumi and J. Bobbett [74]. TDC has also been used to achieve Input/output linearization of a class of non-linear systems as shown by K. Youcef Toumi [78]. This linearization, unlike the control based on non-linear inversion introduced by S. N. Signh and A. A. Schy [68,69], requires a little apriori knowledge of the system dynamics. A decentralized version of the time delay controller was presented in [75] by K. Youcef Toumi and T. A. Fuhlbridge and in [31] by T. C. Hsia and L. S. Gao. In [31], the controller was implemented on a two degrees of freedom manipulator with rigid links and showed successful experimental results. It has been shown also that the simple structure of this type of controller allows a higher sampling rate to be used in order to meet the requirements of high speed operation. A combination of Model Reference Control and Time Delay for the control of non linear plants with unknown dynamics has been investigated by K. Youcef Toumi and O. Ito [77]. The controller uses the time delay to directly estimate the unknown dynamics and inserts the desired model dynamics into the plant.

Several researchers have investigated alternative methods to control the tip of light weight flexible manipulators. The use of perturbation techniques have shown to be effective. In these techniques, the dynamic system of the flexible manipulator is decomposed into two subsystems, a slow subsystem corresponding to the rigid body motion and a fast subsystem corresponding to the flexible motion. In this way, rigid link control techniques can be used to design a controller for the slow sub-system, and linear optimal control techniques can be used to design a controller for the fast sub-system. B. Siciliano and W. J. Book [62], and F. Khorrami and U. Ozguner [38], used a singular perturbation approach to control a one link flexible manipulator. Recently, B. Siciliano et al. [63] extended the use of this technique to control a two link flexible manipulator. S. N. Singh and A. A. Schy [67,68] used a non linear inversion technique along with a linear stabilizer to control a three degrees of freedom robot with the last link being flexible. The non linear decoupling control law is used for the control of the joint angles and the linear stabilizer is used to dampen the oscillations. This stabilizer is based on the LQ optimal control technique. F. Pfeiffer and B. Gebber [56] used a three stage controller to control a three degree of freedom robot with links and joints flexibility. The three stages are as follows: First an optimal reference trajectory for the tip of a rigid robot is generated. Second, the deviations from the path due to structural elasticity are compensated for by a correction in the feedforward path. The third step consists of dampening the residual vibrations by an additional feedback of the deflection at the tip. A non linear compensation technique has been used by I. A. Castelazo and H. Lee [16] in order to control the position of a single link flexible manipulator. This technique uses products of state variables in the feedback to obtain a lightly damped system in the beginning of a step response and a more heavily damped system when the link is close to the steady state position.

Variable structure system (VSS), has been also investigated as a mean for the control of robots with flexible links. P. J. Nathan and S. N. Singh [50], presented a design approach for controlling a two link flexible robot based on VSS theory and pole assignment techniques for stabilization. Simulation for tracking a third order filter generated trajectory was performed. The chattering effects associated with sliding mode control were considered by Y. P. Chen and K. S. Yeung [20] by introducing sliding layer functions.

Model Reference Adaptive Control (MRAC) techniques have been used by J. Z.

Sasiadek and R. Srinivazan [58] in order to control a single-link flexible manipulator. In [70], M. Skowronski used the MRAC to control a two arm flexible manipulators subjected to uncertainty in the payload and arms inertias.

The process of Input/Output feedback linearization has been used by A. De Luca and Siciliano [23], D. Wang and M. Vidiasagar [82,83], and P. Lucibello [46] and P. Lucibello and M. D. Di Benedetto [47]. It has been shown that because the number of outputs is larger than the number of inputs, part of the dynamics of the flexible manipulator becomes unobservable. These unobservable dynamics represent the motion of the flexible sub-system. To examine the stability of the unobservable dynamics, the systems outputs are forced to move along an assigned trajectory by choosing appropriate joint torques. The unobservable dynamics are also referred as the zero dynamics of the system. F. Khorami [37], showed that if the output of a single link flexible manipulator is chosen as the position of the joint angle, and no viscous damping is present in the modal, the zero dynamics are critically stable. However, the zero dynamics are unstable if the position of the tip is chosen as an output. The same results are reported by P. Lucibello [46] for the case of a two link planar robot where only the second link is considered to be flexible and only one parabolic mode is used to model the flexible link. Other approaches used to improve the response of flexible manipulators involves shaping the reference command input. P. H. Meckl and W. P. Seering [48] were able to reduce the vibration in a Cartesian robot by choosing shaped force inputs that minimize the energy of the system at the natural frequencies. N. C. Singer and W. P. Seering [64] have used a series of impulses in order to remove the undesirable frequencies. It has been shown that a sequence of three impulses is more robust than a sequence of two impulses. This method can be extended to systems with multiple modes of vibration. In [93], K. Zuo and David Wang have used the shape input technique in the closed loop control of a class of manipulators with a single flexible link. The controller design is based on the inverse dynamics of the rigid manipulator and the use of a closed loop shaped input filter to reduce the vibrations of the flexible link and to reject any external disturbances. Similar techniques were also used by A. H. Von Flotow and B. Schäfer [28] in order to design a wave absorbing controller for a flexible beam.

2.4 Implementation Issues

The assumptions made during the derivation of the dynamic model of a flexible manipulator render the construction of an experimental model very challenging. Robert H. Cannon and E. Schmitz [15] have constructed an experimental, 1 m long, single link flexible manipulator which can bend freely in the horizontal plane but not the vertical plane or in torsion. In order to limit the deflection to one plane, two thin aluminium beams were connected to a series of flexible bridges which act as pinned joints for horizontal bending. Oacley Celia Marie [52] used air cushion supports at the elbow and at the end point of a horizontal two link flexible manipulator in order to provide torsional stiffness for the link.

If sensors and actuators are not selected properly, they may significantly increase the complexity of manipulators with flexible links. the sensors are used to measure the joint angles and their rates as well as the manipulator end point location. Position encoders and tacho-generators are usually used to measure the joint variables. In order to measure the position and the velocity of the end point of a flexible arm, several techniques are being used. Robert H. Cannon and E. Schmitz [15] used a direct drive DC motor along with a film wire potentiometer and an analog tachometer for the feedback of the joint position and the joint velocity, respectively. The other end point of the arm is equipped with a small light bulb for optical sensing of the tip position. The tip sensor consists of a focusing lens located 1 m above the light bulb. A linear fast silicon photodetector is located behind the lens providing two analog voltages which represent the x and y position of the light spot created by the light bulb at the tip. In [52], Oacley Celia Marie used two brushless DC motors at each of the joints of a two link flexible manipulator. To measure the joint angles, rotary variable differential transformers (RVDT's) are being used. The angular rate signals were generated by passing the RVDT signals through analog bandpass filters. End point sensing has been achieved through a CCD attached to the ceiling directly above the two link manipulator and tracks an optical target located at the manipulator end point. Another end point position sensor has been proposed by F. Demeester and H. V. Brussel [25]. This sensor is called DIOMEDES (DIodes Optical System for MEasuring Structural DEflectionS). The sensor uses three semiconductor laser diodes and three Position Sensitive Detectors (PSD's). The laser diodes are mounted at one end of the link and the PSD's are mounted at the other end The PSD's are arranged such that they make an equilateral triangle in a plane parallel to the plane of the three laser diodes. To demonstrate the feasibility of the optical sensor, a flexible aluminum beam with a rectangular cross section is used in this experiment.

Chapter 3

Dynamics of a one link Flexible Manipulator

3.1. Introduction

In this chapter, the dynamic model of a one link flexible manipulator is derived. The method used in the derivation is based on the Lagrangian formulation and the Generalized Inertia Matrix concept [81]. Two cubic interpolation functions are used to model the deflection of the link [17,18,19]. In Section 3.2, the GIM (Generalized Inertia Matrix) of a single link and the GIM of a load attached at the tip of the link are derived. The total kinetic energy of the link and the load is derived from the overall inertia matrix of the system and presented in Section 3.3. The link is assumed to operate in a gravity-free environment, therefore, only the elastic potential energy is considered and its derivation is presented in Section 3.4. The dynamic equations of the flexible link are derived in Section 3.5. After obtaining the dynamic model, two simulations were performed. The first simulation is to check the response of the system when it is subjected to an initial deformation, and the second simulation is to investigate the effect of an input torque at the joint on the motion of the tip. The results of these two test are discussed in section 3.6 .

3.2. Generalized Inertia Matrix (GIM)

3.2.1 GIM of the flexible link

With reference to Figure 3.1, the transformation of ***oxy*** into ***OXY*** is given by:

$$R_0^1 = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \qquad 3.1$$

The position of a point P on the link is represented in the frame ***oxy*** by a vector r_{oxy} as:

$$r_{oxy} = [x \quad u(x,t)]^T \qquad 3.2$$

x is the position of P along an axis tangent to the link and $u(x,t)$ is the deflection of the link at P in the ***oxy*** plane. $u(x,t)$ can be expressed in terms of the tip deflection ξ_e, the tip slope δ_e, and the shape functions $\phi_1(x)$ and $\phi_2(x)$ as:

$$\begin{aligned} u(x,t) &= \phi_1(x)\,\xi_e + \phi_2(x)\delta_e \\ &= \Phi^T(x)\,\eta \end{aligned} \qquad 3.3$$

$\eta=[\ \xi_e \quad \delta_e\ ]^T$ is a vector representing the deflection and the slope at the tip of the link and $\Phi(x)=[\ \phi_1(x) \quad \phi_2(x)\ ]^T$ is a vector representing the shape functions used to model the deflection along the link. The functions $\phi_1(x)$ and $\phi_2(x)$ are the cubic interpolation functions given by [17,18,19]:

$$\phi_1(x) = 3\left(\frac{x}{l}\right)^2 - 2\left(\frac{x}{l}\right)^3 \qquad 3.4$$

$$\phi_2(x) = -l(\frac{x}{l})^2 + l(\frac{x}{l})^3 \qquad 3.5$$

The cubic interpolation functions are usually used in the finite elements method where the flexible link is considered as an assemblage of a finite number of small elements. By requiring the internal forces to be in balance and the displacements to be compatible at certain points known as *nodes*, the entire link is compelled to act as one entity [73]. It was shown in [73] that when n elements are used to model a flexible link with a clamped-free boundary conditions, the first n natural frequencies are within acceptable accuracy. Another set of shape functions commonly used in the modelling of flexible links is based on trigonometric functions. This set of functions is usually associated with the assumed modes method. Trigonometric functions are defined over the entire length of the link while the cubic interpolation functions are defined over each element. The use of the trigonometric functions leads to complicated integrations in order to obtain expressions for the mass and stiffness matrices. Furthermore, for the assumed modes methods, and when the exact boundary conditions are used, the frequency equation used to obtain the eigenvalues of the link becomes time dependent. As an example of this complication is the case of a flexible multi-link manipulator with revolute joints in which the mass boundary condition of the first link depends on the configuration of the other links.

In this study, the finite element method was adopted in order to derive a model for the flexible link manipulator. Due to the fact that only the first few modes are to be considered and to avoid further complication in the dynamic model, only one element is used to represent the flexible link.

Assuming that the element is homogeneous with a uniform cross section, the kinetic energy can be written as:

$$T_l = \frac{m}{2}\int_0^l \left[\frac{dr}{dt}\right]^T\left[\frac{dr}{dt}\right]dx \qquad 3.6$$

where r is the position of point P expressed in the fixed frame OXY and given by:

$$r = R_0^1 r_{oxy} \qquad 3.7$$

Since r is a function of the joint angle θ and the tip coordinates ξ_e and δ_e, its time derivative is obtained using the chain rule and is given by:

$$\frac{\partial r}{\partial t} = \begin{bmatrix} \frac{\partial r}{\partial \theta} & \frac{\partial r}{\partial \xi_e} & \frac{\partial r}{\partial \delta_e} \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{\xi}_e \\ \dot{\delta}_e \end{bmatrix} \qquad 3.8$$

Substituting Equation 3.8 into Equation 3.6, the kinetic energy can be written in matrix form as:

$$T_l = \frac{1}{2}\dot{q}^T M_l \, \dot{q} \qquad 3.9$$

Where $q = [\ \theta \ \ \xi_e \ \ \delta_e\ ]^T$ represents the generalized coordinates of the overall motion of the flexible link and M_l is the Generalized Inertia Matrix of the link and given by:

$$M_l = m\int_0^l \begin{bmatrix} \frac{\partial r^T}{\partial \theta} \\ \frac{\partial r^T}{\partial \xi_e} \\ \frac{\partial r^T}{\partial \delta_e} \end{bmatrix} \begin{bmatrix} \frac{\partial r}{\partial \theta} & \frac{\partial r}{\partial \xi_e} & \frac{\partial r}{\partial \delta_e} \end{bmatrix} dx \qquad 3.10$$

The elements of the 3x3, positive definite matrix M_l are computed using MAPLE [1]. Details of the derivation are presented in appendix B.

3.2.2 GIM of a payload at the tip of the link

The total kinetic energy of the one link flexible manipulator shown in Figure 3.2, is given by the kinetic energy associated with the link having a distributed mass m per unit length and the payload at the tip with a concentrated mass m_L and a moment of inertia I_L. The absolute position of the center of mass of the payload located at the tip of the link is given by:

$$r_e = R_1^0 [l \quad \xi_e]^T \qquad 3.11$$

The kinetic energy of the payload is:

$$T_L = \frac{1}{2} m_L \frac{\partial r_e^T}{\partial t} . \frac{\partial r_e}{\partial t} + \frac{1}{2} I_L \omega_e^2 \qquad 3.12$$

where $\omega_e = \dot{\theta} + \dot{\delta}_e$ is the total angular velocity of the tip. The vector r_e in Equation 3.11

[1] A system of algebraic computation

is a function of θ and ξ_e, therefore, its time derivative is given by:

$$\frac{\partial r_e}{\partial t}=\begin{bmatrix}\frac{\partial r_e}{\partial \theta} & \frac{\partial r_e}{\partial \xi_e}\end{bmatrix}\begin{bmatrix}\dot{\theta}\\ \dot{\xi}_e\end{bmatrix} \qquad 3.13$$

Substituting Equation 3.13 into 3.12 and rearranging lead to the following expression of the kinetic energy T_L.

$$T_L=\frac{1}{2}\dot{q}^T M_L \dot{q} \qquad 3.14$$

where M_L is the GIM associated with the payload and is given by:

$$M_L=m_L\begin{bmatrix}\frac{\partial r_e^T}{\partial \theta}\\ \frac{\partial r_e^T}{\partial \xi_e}\\ 0\end{bmatrix}\begin{bmatrix}\frac{\partial r_e}{\partial \theta} & \frac{\partial r_e}{\partial \xi_e} & 0\end{bmatrix}+I_L\begin{bmatrix}\frac{\partial \omega_e}{\partial \theta}\\ 0\\ \frac{\partial \omega_e}{\partial \delta_e}\end{bmatrix}\begin{bmatrix}\frac{\partial \omega_e}{\partial \theta} & 0 & \frac{\partial \omega_e}{\partial \delta_e}\end{bmatrix} \qquad 3.15$$

3.3 Total Kinetic Energy

The overall inertia matrix of the flexible link shown in Figure 3.2 is obtained by adding the mass matrix of the link without any payload, given by Equation 3.10, and the mass matrix of the load given by equation 3.15. The vector q and the overall inertia matrix can be partitioned into rigid and flexible sub-systems as follows:

$$q=[\theta \quad \eta^T]^T \qquad 3.16$$

$$M = \begin{bmatrix} [M_r] & [M_{rf}]^T \\ [M_{rf}] & [M_f] \end{bmatrix} \qquad 3.17$$

where θ represents the rigid link coordinate and η represent the link flexture coordinates. M_r is a scaler representing the inertia associated with the rigid body motion

$$M_r = m \int_0^l \left[\frac{\partial r}{\partial \theta}\right]^T \left[\frac{\partial r}{\partial \theta}\right] dx \qquad 3.18$$

M_{rf} is a vector representing the coupling between the rigid and the flexible motions

$$M_{rf} = m \int_0^l \left[\frac{\partial r}{\partial \theta}\right]^T \left[\frac{\partial r}{\partial \eta}\right] dx \qquad 3.19$$

and M_f is a 2x2 matrix representing the mass matrix of the flexible element.

$$M_f = m \int_0^l \left[\frac{\partial r}{\partial \eta}\right]^T \left[\frac{\partial r}{\partial \eta}\right] dx \qquad 3.20$$

The total kinetic energy of the flexible link is obtained by adding the kinetic energy of link without payload T_l and the kinetic energy of the payload T_L. Using the partitioned matrix M given by Equation 3.17, and the partioned vector q given by Equation 3.16, the total kinetic energy can be written as:

$$T = \frac{1}{2}[\dot\theta \quad \dot\eta^T] M \begin{bmatrix} \dot\theta \\ \dot\eta \end{bmatrix} \qquad 3.21$$
$$= \frac{1}{2}M_r\dot\theta^2 + \frac{1}{2}\dot\eta^T M_f \dot\eta + \dot\theta M_{rf}^T \dot\eta$$

In Equation 3.21, the first term represents the kinetic energy associated with the rigid system, the second term is the kinetic energy of the flexible system and the third term is the coupling between the rigid and the flexible sub-systems.

3.4 Potential Energy

In general, the potential energy is composed of two parts: gravitational potential energy and elastic potential energy. In this study, only the elastic part is considered since the link is confined to move and flex only in the horizontal plane. The elastic potential energy of a flexible link can be expressed in terms of the elastic modulus *E*, the cross sectional moment of inertia *I*, and the deflection *u(x,t)* as:

$$U = \frac{1}{2}\int_0^l EI\left[\frac{\partial^2 u}{\partial x^2}\right]^2 dx \qquad 3.22$$

The term $\left[\frac{\partial^2 u}{\partial x^2}\right]^2$ can be expressed in terms of $\Phi = [\phi_1(x) \quad \phi_2(x)]^T$ and η as follows:

$$\left[\frac{\partial^2 u}{\partial x^2}\right]^2 = \eta^T \left[\frac{d^2\Phi}{dx^2}\right]\left[\frac{d^2\Phi}{dx^2}\right]^T \eta \qquad 3.23$$

The potential energy can be written in a quadratic form as a function of the vector η as follows:

$$U=\frac{1}{2}\eta^T \mathrm{K}\eta \qquad 3.24$$

where K is the stiffness matrix of the link and is given by:

$$\mathrm{K}=EI\int_0^l \left[\frac{d^2\Phi}{dx^2}\right]\left[\frac{d^2\Phi}{dx^2}\right]^T dx \qquad 3.25$$

3.5 Equations of Motion

The equations of motion of the flexible link can be obtained using the Lagrangian formulation. The generalized force corresponding to the joint variable θ is the input torque τ. For the variables ξ_e and δ_e, the corresponding generalized forces are zero. The general form of Lagrange's equation of motion for the joint variable θ is given by:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial\dot{\theta}}\right)-\frac{\partial L}{\partial\theta}=\tau \qquad 3.26$$

and for the variables ξ_e and δ_e (represented by the vector η) is:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial\dot{\eta}}\right)-\frac{\partial L}{\partial\eta}=0 \qquad 3.27$$

where $L=T+U$ is the Lagrangian of the system, T is the total kinetic energy, and U is the total potential energy. Symbolic closed form equations can be obtained by substituting for the kinetic and potential energies in Equations 3.26 and 3.27 and differentiating. This gives the following system of equations:

$$\begin{bmatrix} [M_r] & [M_{rf}]^T \\ [M_{rf}] & [M_f] \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\eta} \end{bmatrix} + \begin{bmatrix} H_r \\ H_f \end{bmatrix} + \begin{bmatrix} G_r \\ G_f \end{bmatrix} = \begin{bmatrix} \tau \\ 0 \end{bmatrix} \qquad 3.28$$

all the terms in Equation 3.28 are derived using MAPLE and the detailed algorithms with the resulting equations are presented in appendix B. H_r and H_f represent the Coriolis and the centrifugal terms. G_f represents the stiffness term. $(.)_r$ and $(.)_f$ represent the rigid and flexible motion, respectively.

$$H_r = \dot{M}_r\dot{\theta} + \dot{M}_{rf}^T\dot{\eta} - \frac{1}{2}\frac{\partial M_r}{\partial \theta}\dot{\theta}^2 - \frac{\partial M_{rf}^T}{\partial \theta}\dot{\theta}\dot{\eta} - \frac{1}{2}\dot{\eta}^T\frac{\partial M_f}{\partial \theta}\dot{\eta} \qquad 3.29$$

$$G_r = \frac{1}{2}\eta^T\frac{\partial \mathrm{K}}{\partial \theta}\eta \qquad 3.30$$

$$H_f = \dot{M}_{rf}\dot{\theta} + \dot{M}_f\dot{\eta} - \frac{1}{2}\dot{\eta}^T\frac{\partial M_f}{\partial \eta}\dot{\eta} - \frac{1}{2}\frac{\partial M_r}{\partial \eta}\dot{\theta}^2 - \dot{\theta}\frac{\partial M_{rf}}{\partial \eta}\dot{\eta} \qquad 3.31$$

$$G_f = K\eta \tag{3.32}$$

For a one link flexible manipulator, the matrices M_f, K and M_{rf} are constant. M_r is time varying and depends only on the coordinates describing the flexible motion. Consequently, Equations 3.29, 3.30 and 3.31 reduce respectively to:

$$\begin{aligned} H_r &= \frac{\partial M_r}{\partial \eta}\dot{\eta}\dot{\theta} \\ &= \frac{\partial M_r}{\partial \xi_e}\dot{\xi}_e\dot{\theta} + \frac{\partial M_r}{\partial \delta_e}\dot{\delta}_e\dot{\theta} \end{aligned} \tag{3.33}$$

$$G_r = 0 \tag{3.34}$$

$$\begin{aligned} H_f &= -\frac{1}{2}\frac{\partial M_r}{\partial \eta}\dot{\theta}^2 \\ &= -\frac{\dot{\theta}^2}{2}\begin{bmatrix} \dfrac{\partial M_r}{\partial \xi_e} \\ \dfrac{\partial M_r}{\partial \delta_e} \end{bmatrix} \end{aligned} \tag{3.35}$$

3.6 Simulation Results

The physical parameters of the one link manipulators used in the analysis are shown in Table 3.1.

Mass (kg/m) m	*Length (m)* l	*Stiffness (N.m²)* EI
1.0	1.0	100.0

Table 3.1: Physical parameters of the one link flexible manipulator.

This link is equivalent to a steel bar with a modulus of elasticity $E=2.0\ 10^{11}$ N/m^2 and a circular cross section with a diameter d=10 mm. It is assumed that no structural damping is present.

In order to test the dynamic model of the flexible link, two cases were investigated. The first is when the joint is clamped and the tip is subjected to an initial deformation $\xi_e=0.01$ m as illustrated in Figure 3.3. The second case is when an input torque is being applied at the joint with zero initial deformation as illustrated in Figure 3.4. Figures 3.5 and 3.6 show the tip deflection and the tip slope respectively, when there is no load at the tip of the link. Due to the absence of any structural damping in the model, all the elastic energy stored in the system due to the initial tip deflection is transformed into kinetic energy. The presence of the two vibration modes is apparent in these two Figures. Figures 3.7 and 3.8 show the effect of adding a load of mass $m_L=1.0$ kg and inertia $I_L=0.1$ kg.m^2 at the tip of the flexible link. A significant reduction in the frequencies of the link is observed. The maximum tip deflection and maximum tip slope have also increased due to the added load.

The simulation results due to an applied torque and due to a zero initial deformation are shown in Figures 3.9 to 3.14. During this test, the motor at the joint was given a step input (from $\theta=0$ to $\theta=\pi/10$). The motion of the joint angle due to the step input is shown in Figure 3.9. The presence of the two modes representing the flexible motion as well as the one representing the rigid body motion can be clearly seen in this figure. The large overshoot in the rigid body motion is due the absence of joint damping. The deflection and the slope at the tip of the link during the rigid body motion are shown in Figures 3.10 and 3.11, respectively. It can be seen from these two figures that the maximum tip deflection and slope are attained when the rigid body motion is at the maximum acceleration. The effect of adding a load of mass m_L =1.0 kg and inertia I_L=0.1 kg.m^2 on the rigid body motion as well as on the motion of tip is shown in Figures 3.12 to 3.14. Figure 3.12 shows the motion of the joint angle. It can be seen from this figure that the frequencies of oscillation have been reduced due to the load at the tip and that the overall rigid body motion is slower than that in Figure 3.9. The tip deflection and slope, when a payload is attached at the tip of the link, are shown in Figures 3.13 and 3.14. The reduction in the oscillation frequencies due to the added payload is apparent in these two figures.

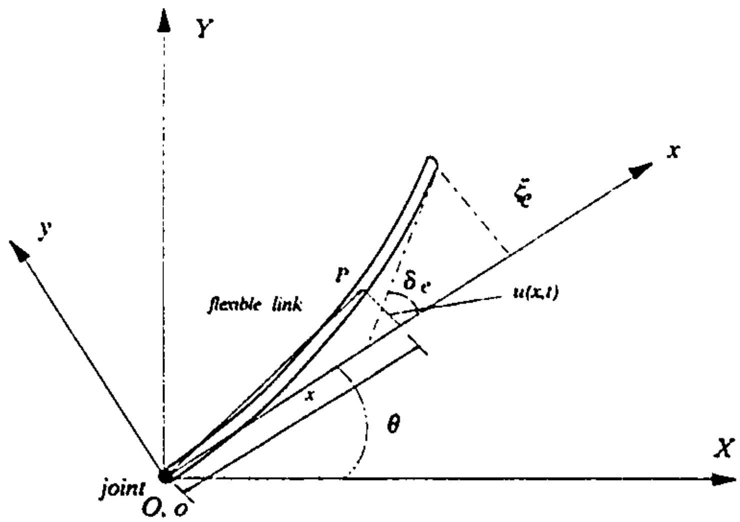


Figure 3.1: Coordinates System of a one flexible link with fixed base.

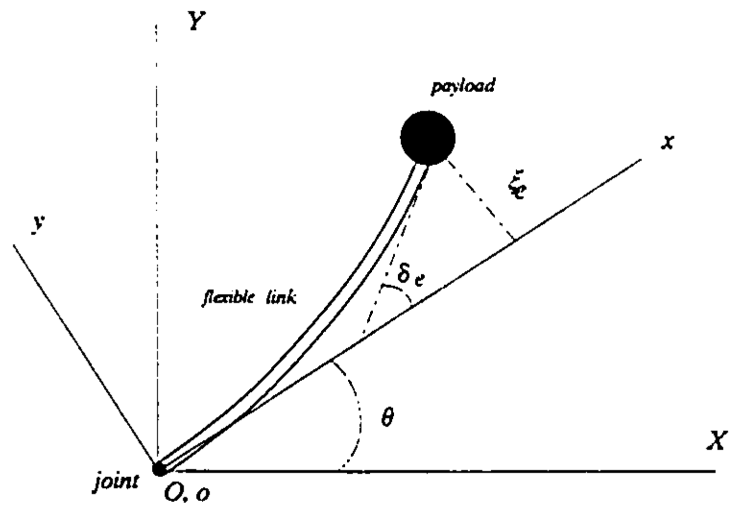


Figure 3.2: One Link Flexible Manipulator

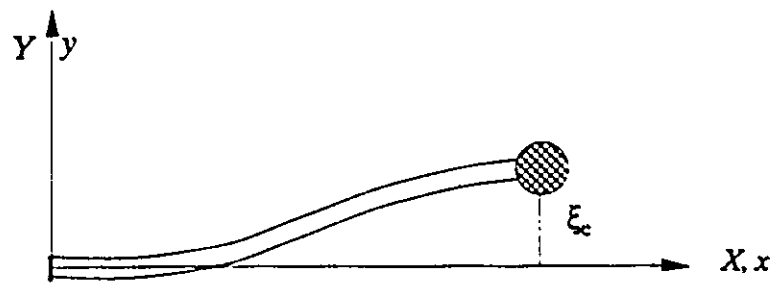


Figure 3.3: Configuration of the link with the initial deformation

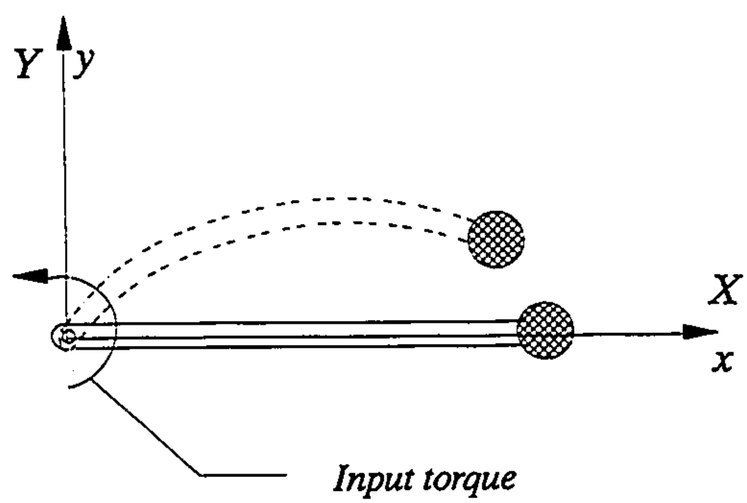


Figure 3.4: Configuration of the link subjected to a joint torque.

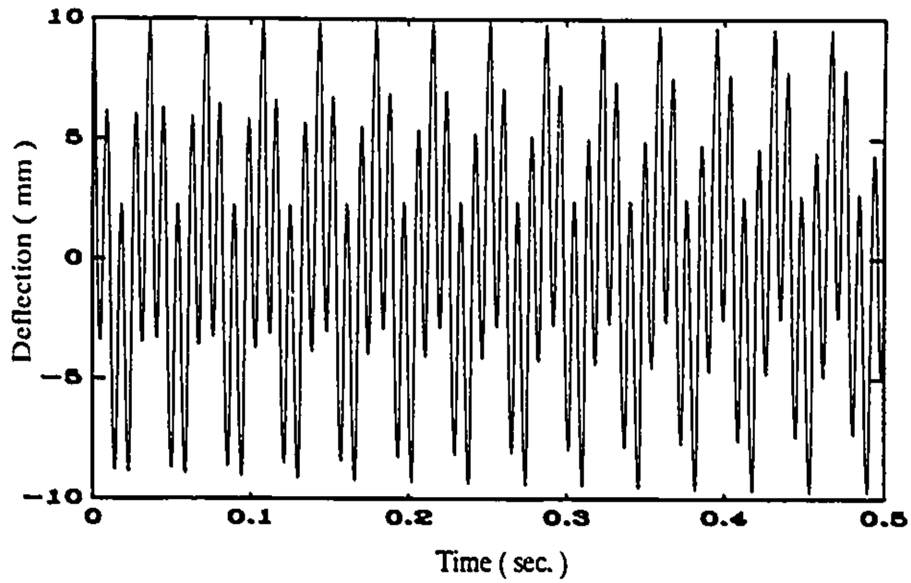


Figure 3.5: Tip deflection due to an initial deformation $\xi_e(0)$=0.01 m. m_L=0 and I_L=0.

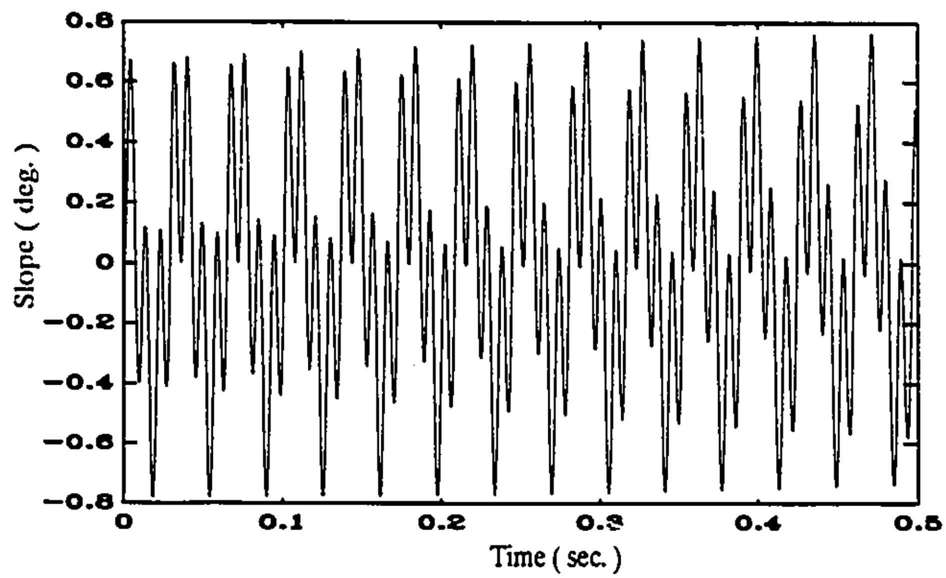


Figure 3.6: Tip slope due to an initial deformation $\xi_e(0)$=0.01 m. m_L=0 and I_L=0.

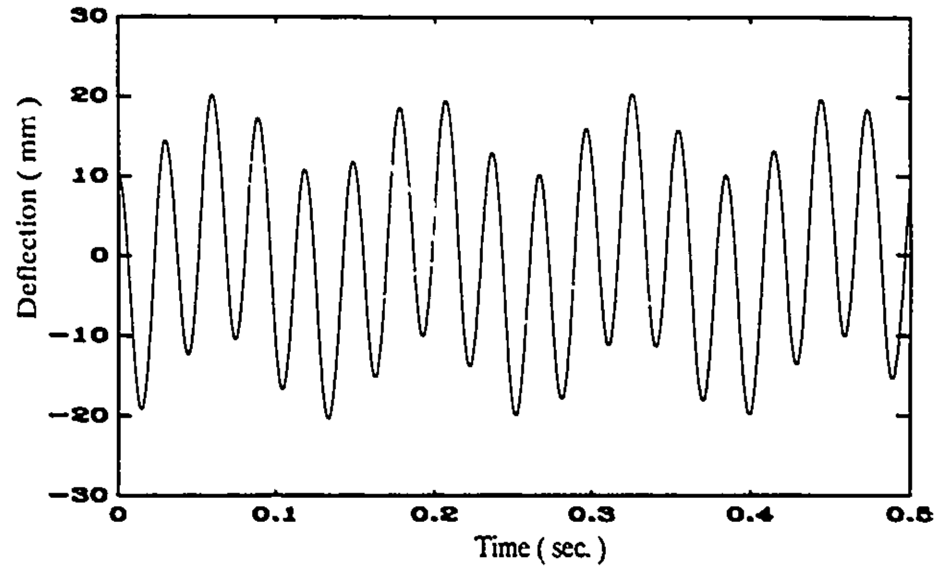


Figure 3.7: Tip deflection due to an initial deformation $\xi_c(0)$=0.01 m. m_L=1.0 kg and I_L=0.1 kg.m^2.

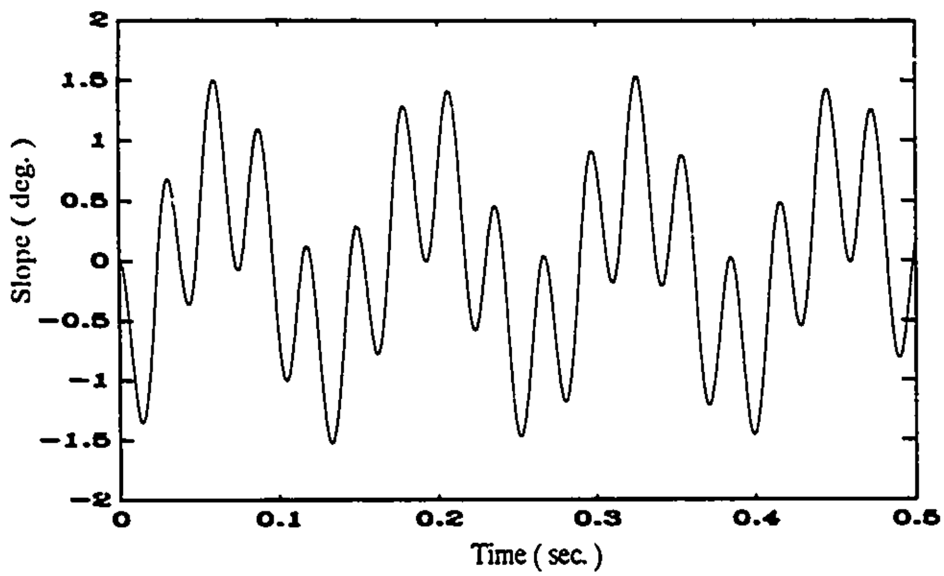


Figure 3.8: Tip slope due to an initial deformation $\xi_c(0)$=0.01 m. m_L=1.0 kg and I_L=0.1 kg.m^2.

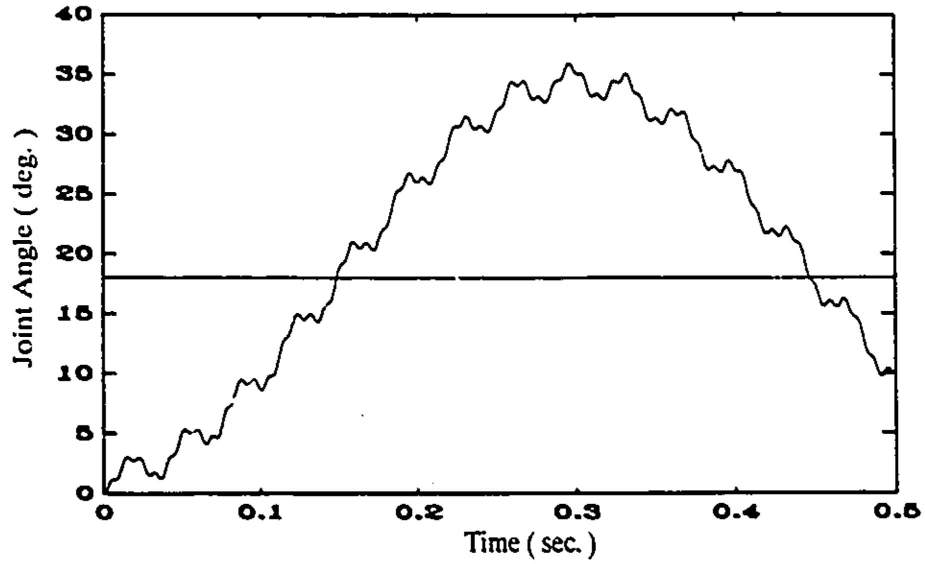


Figure 3.9: Motion of the joint angle due to an input torque and zero initial deformation. m_L=0.0 kg and I_L=0.0 kg.m^2.

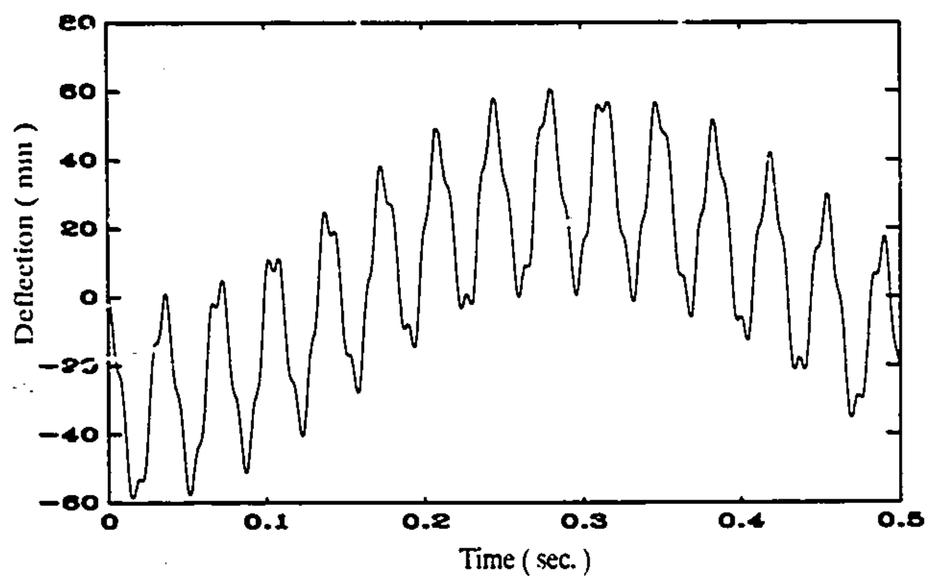


Figure 3.10: Tip deflection to an input torque and zero initial deformation. m_L=0 kg and I_L=0 kg.m^2.

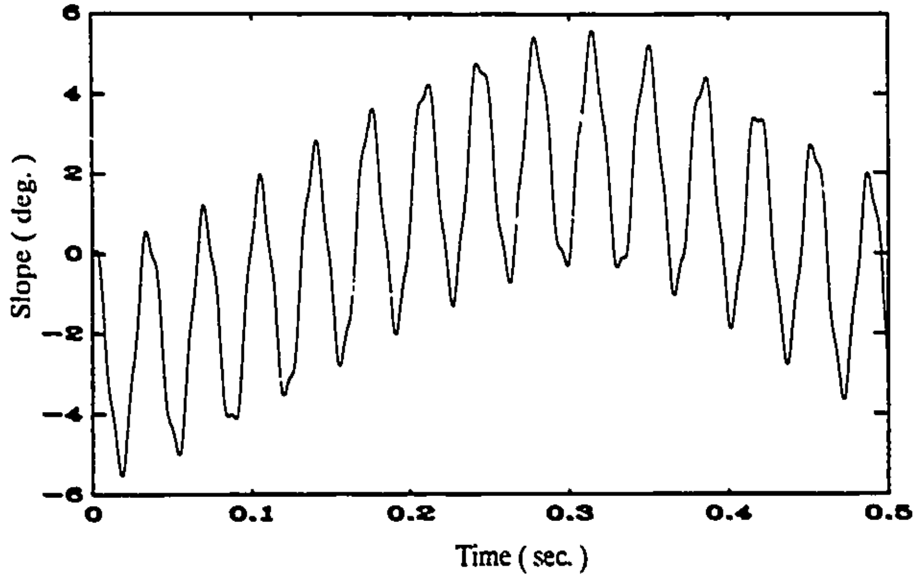


Figure 3.11: Tip slope due to an input torque zero initial deformation. m_L=0.0 kg and I_L=0.0 kg.m^2.

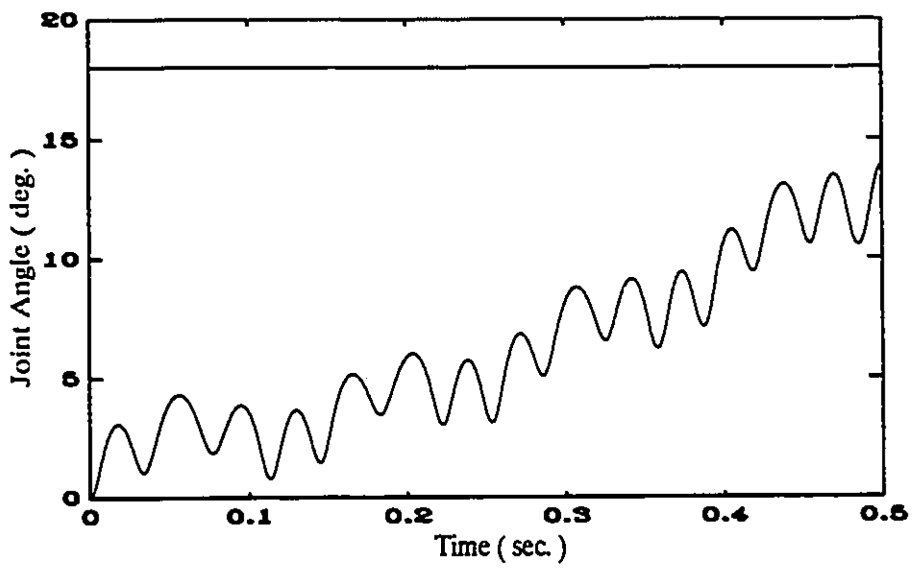


Figure 3.12: Motion of the joint angle due to an input torque and zero initial deformation. m_L=1.0 kg and I_L=0.1 kg.m^2.

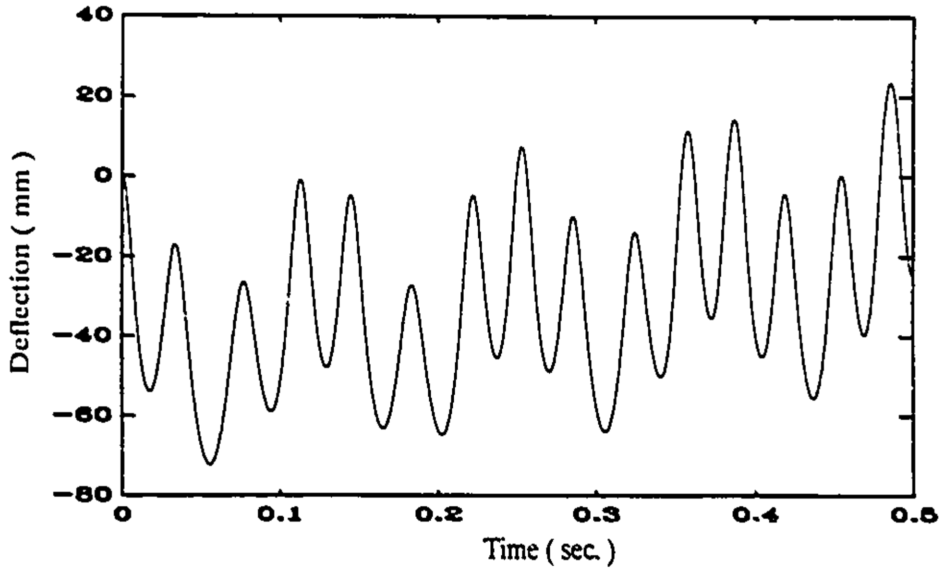


Figure 3.13: Tip deflection due to an input torque zero initial deformation. m_L=1.0 kg and I_L=0.1 kg.m^2.

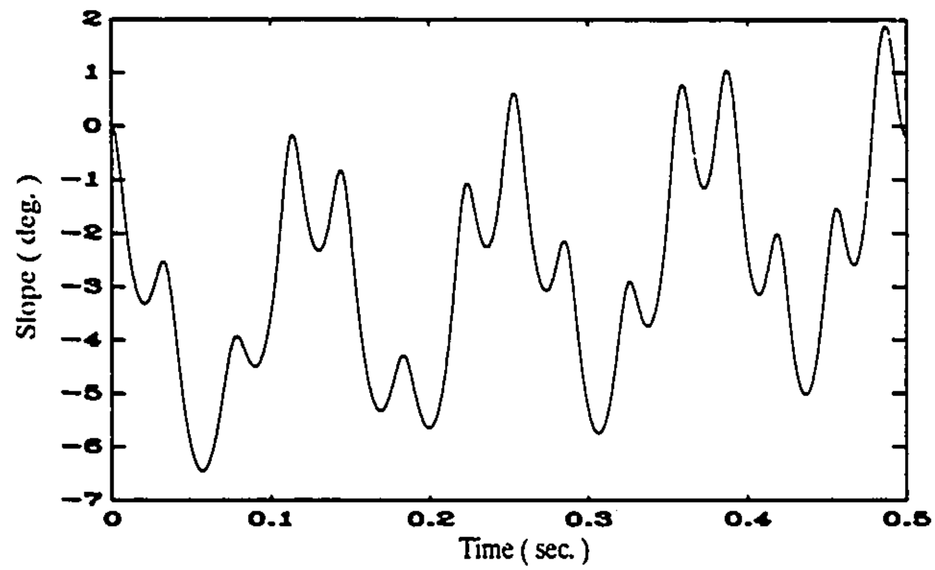


Figure 3.14: Tip slope due to an input torque and zero initial deformation. m_L=1.0 kg and I_L=0.1 kg.m^2.

Chapter 4

Control of a One link Flexible Manipulator

4.1 Introduction

In the case of the control of manipulators with rigid links, it is assumed that the tip position can be accurately deduced from the joint angle and the knowledge of the length of each link. This assumption is justified in view of the rigidity of the links, consequently, the controller uses the joint angle as the only feedback signal. In the case of the control of the end point position of a manipulator with flexible links, the joint angle cannot be used as the sole feedback signal. The position of the end point must be measured and used as another feedback signal. One way to control a rigid arm is to use the computed torque method, which decouples and linearizes the system equations through a non-linear feedback. When the same technique is used to control a flexible arm, the linearization property is lost [23]. The control objective when controlling a robot with flexible links can be defined as the regulation of the end point around a final position or as driving it along a feasible path. Another objective is the tracking of a pre-defined joint trajectories while limiting the arm deflections. In this chapter, two types of controls are presented. The first, presented in Section 4.2, is based on the control of the joint angle by considering the tip deflection as an external disturbance. In this approach, only the position and the velocity of the joint are used in the feedback loop. Because the actuator and the sensor are both located at the joint, this approach is referred to as the collocated control. The effect of the smoothness of the desired trajectory on the tip deflection is presented in Sections 4.2.1 to 4.2.3. The other approach to controlling a flexible link is based on using the absolute angle of a point along the link as the system's output. Since the actuator is located at the joint and the controlled variable is along the link, this approach is referred to as the non-collocated control. In Section 4.3, the response of the flexible link under the non-collocated control is presented.

4.2. Collocated Control

The open loop equations of motion given by Equation 3.28 can be written as:

$$M_r\ddot{\theta} + M_{rf}^T\ddot{\eta} + H_r = \tau \qquad 4.1$$

$$M_{rf}\ddot{\theta} + M_f\ddot{\eta} + H_f + K\eta = 0 \qquad 4.2$$

The first equation represents the motion of the rigid sub-system and the second represents the motion of the flexible sub-system. An equation governing the motion of the rigid sub-system, when the effects of the motion of the flexible sub-system are considered as a deterministic disturbance, can be obtained as follows:

Since the matrix M_f is non singular, the acceleration vector $\ddot{\eta}$ in Equation 4.2 can be written as:

$$\ddot{\eta} = -M_f^{-1}(M_{rf}\ddot{\theta} + H_f + K\eta) \qquad 4.3$$

Substituting Equation 4.3 into 4.1 and rearranging results in the single input single output system given by:

$$\Gamma_0\ddot{\theta} + \Lambda_0 = \tau \qquad 4.4$$

where

$$\Gamma_0 = M_r - M_{rf}^T M_f^{-1} M_{rf} \qquad 4.5$$

and

$$\Lambda_0 = H_r - M_{rf}^T M_f^{-1}(H_f + K\eta) \qquad 4.6$$

Equation 4.4 is equivalent to that of a rigid manipulator with Γ_0 as the inertia term and Λ_0 as the non linear term representing the coriolis, centrifugal and stiffness terms. The problem of controlling the system given by Equation 4.4 consists of finding a torque τ that can reproduce a given trajectory $\theta_d(t)$. A non-linear controller that linearizes the system in Equation 4.4 is given by:

$$\tau = \Gamma_0(\ddot{\theta}_d + K_p(\theta_d - \theta) + K_d(\dot{\theta}_d - \dot{\theta})) + \Lambda_0 \qquad 4.7$$

K_p and K_d are constant position and velocity gains for the joint variable, respectively. Substituting the control law given by Equation 4.7 into Equation 4.4 gives

$$\ddot{e} + K_d\dot{e} + K_p e = 0 \qquad 4.8$$

$e(t)=\theta_d(t)-\theta(t)$ is the error between the desired and the actual joint position.

Asymptotic stability of the system in equation 4.8 can be achieved by a proper selection of K_p and K_d. This implies that for a given initial condition, $\theta(t) \rightarrow \theta_d(t)$ as $t \rightarrow \infty$. For a rigid link manipulator, K_p and K_d are usually selected to produce a critically damped response of the joint angle.

The controller in Equation 4.7 depends not only on the control output, but also on the variables representing the deflection of the link and their first derivatives. In order to completely determine the required input torque, the variables representing the deflection can be inferred from sensors at the tip. An estimate of these variables can also be obtained by designing a non-linear observer [57]. A block diagram of the control system is shown in Figure 4.1. Even though the variables η and $\dot{\eta}$ are used to decouple the dynamic equations of the flexible manipulator, only the joint variable θ and its first time derivative $\dot{\theta}$ are used in the error dynamics given by Equation 4.8. The variables θ_d, $\dot{\theta}_d$ and $\ddot{\theta}_d$ represent the pre-defined desired trajectories which are used to compute the required actuators torques. In the case of manipulators with rigid links, the only restriction that should be taken into account when choosing a desired trajectory is related to the actuator limitations like torque saturation and bandwidth. In the case of robots with flexible links, a desired trajectory with high frequency content produces perturbations in the response of these robots, which lead to undesirable tip oscillations. In order to investigate the effects of high frequency content, four types of trajectories have been considered: Position step, Bang-Bang acceleration profile, Sinusoidal acceleration profile and a Gaussian velocity profile.

4.2.1 Joint Step Input

During the point to point control of robotic systems, the desired position is given in the form of a step command. The magnitude of the step is the difference between the initial position and the desired final position. For a step command, the joint actuators are commanded to move from the initial position to the desired position using the maximum power available. For such an input, the desired joint angle of a one link manipulator can be expressed as:

$$\theta_d(t) = \begin{cases} \theta_0 & t = 0 \\ \theta_f & t > 0 \end{cases} \qquad 4.9$$

θ_0 is the initial position and θ_f is the desired final position. The desired joint velocity and acceleration are not defined for t=0, which results in a discontinuous position function $\theta_d(t)$ at $t=0$. For $t>0$, $\dot{\theta}_d(t) = \ddot{\theta}_d(t) = 0$. The results of a step input in the joint angle given by $\theta_0=0$ and $\theta_f=\pi/2$ are shown in Figure 4.2. The position and velocity gains are selected such that the joint exhibits a critically damped response with a natural frequency ω_n=10 rad/sec. From Equation 4.8, the position gain K_p can be expressed in terms of the error dynamics as $K_p = \omega_n^2$ and the velocity gain $K_d = 2\zeta\omega_n$. For ω_n=10 rad/sec and ζ=1, the corresponding K_p and K_d are 100 and 20, respectively.

Even though the response of the joint angle is critically damped, the response of the absolute angle of the tip, which includes the response of the joint angle and the relative tip deflection is very oscillatory as shown in 4.2. This motion represents a critically stable response due to the fact that no structural damping was introduced in the system. The deflection of the tip with respect to the rigid link is shown in Figure 4.3. The oscillation amplitudes are relatively large because of the way the system was exited at the beginning of the motion.

4.2.2 Input Command with Bang-Bang Acceleration Profile

For a trajectory tracking command, the position, velocity and acceleration of the desired trajectory need to be specified. One type of trajectory is based on a bang-bang acceleration profile and can be generated by:

$$\ddot{\theta}_d(t) = \begin{cases} a & 0 < t \leq T_1 \\ -a & T_1 < t \leq T_2 \\ 0 & t > T_2 \end{cases} \tag{4.10}$$

$$\dot{\theta}_d(t) = \begin{cases} at & 0 < t \leq T_1 \\ -a(t-T_2) & T_1 < t \leq T_2 \\ 0 & t > T_2 \end{cases} \tag{4.11}$$

$$\theta_d(t) = \begin{cases} \frac{1}{2}at^2 & 0 < t \leq T_1 \\ -\frac{1}{2}a(t-T_2)^2 + aT_1^2 & T_1 < t \leq T_2 \\ \theta_f & t > T_2 \end{cases} \tag{4.12}$$

The desired trajectory is divided into three segments. The first segment is from $t=0$ to $t=T_1$ and has a constant angular acceleration $\ddot{\theta}_d(t)=a$. The second segment is from $t=T_1$ to $t=T_2$ and has a negative constant acceleration $\ddot{\theta}_d(t)=-a$. The third segment is when $t>T_2$ and describes the joint angle at its final position $\theta_d(t)=\theta_f$ and $\ddot{\theta}_d(t)=\dot{\theta}_d(t)=0$. The acceleration a is calculated from $a=\theta_f/T_1^2$. Figure 4.4 shows the desired position, the desired velocity and the desired acceleration for $T_1=0.5$ sec, $T_2=1.0$ sec and $\theta_f=\pi/2$. The bang-bang acceleration trajectory results in a continuous function $\theta_d(t)$, which represents the desired position angle, and has a continuous first derivative. Figure 4.5 shows the tip deflection of the one link flexible manipulator with respect to the rigid link. The figure shows that, even though the oscillation amplitudes are much smaller than those due to a step command, the flexible modes are being exited by the sudden change in the acceleration at $t=0$, $t=T_1$ and at $t=T_2$. For each of the three segments, the tip deflection is an oscillatory motion with constant amplitudes. The lowest amplitude was found during the first segment $(t<T_1)$ because the link started from rest with zero initial deflection. The highest amplitude was found during the second segment $(T_1<t<T_2)$ because of sudden change in the joint acceleration from $\ddot{\theta}_d(t)=a$ to $\ddot{\theta}_d(t)=-a$.

4.2.3 Input Command with a Sinusoidal Acceleration Profile

In order to eliminate the discontinuity in the acceleration, a sinusoidal acceleration profile is used instead of the bang-bang acceleration profile. In this type of trajectory, the desired joint angle $\theta_d(t)$ moves from an initial position $\theta_d(t)=0$ to a final position

$\theta_d(t)=\theta_f$. The duration of the motion T_1 and the acceleration is given by Equation 4.13. For $t>T_1$, the desired joint angle is locked at $\theta_d(t)=\theta_f$ with $\ddot{\theta}_d(t)=\dot{\theta}_d(t)=0$. The desired velocity and the desired position are given by Equations 4.14 and 4.15, respectively.

$$\ddot{\theta}_d(t)=\begin{cases} \dfrac{2\pi\theta_f}{T_1^2}\sin(\dfrac{2\pi t}{T_1}) & 0\le t\le T_1 \\ 0 & t>T_1 \end{cases} \qquad 4.13$$

$$\dot{\theta}_d(t)=\begin{cases} \dfrac{\theta_f}{T_1}(1-\cos(\dfrac{2\pi t}{T_1})) & 0\le t\le T_1 \\ 0 & t>T_1 \end{cases} \qquad 4.14$$

$$\theta_d(t)=\begin{cases} \theta_f(\dfrac{t}{T_1}-\dfrac{1}{2\pi}\sin(\dfrac{2\pi t}{T_1})) & 0\le t\le T_1 \\ \theta_f & t>T_1 \end{cases} \qquad 4.15$$

Figure 4.6 shows the position, velocity and acceleration generated using Equations 4.13 to 4.15 for $T_1=1.0$ sec and $\theta_f=\pi/2$. The only discontinuity in the trajectories is in the derivative of the acceleration at $t=T_1$ when the joint angle reaches its final destination. The tip deflection due to such trajectories is shown in Figure 4.7. As compared to the response due to the bang-bang acceleration trajectory, the oscillations of the tip are significantly reduced specially during the transient motion of the joint angle ($0<t<T_1$). The amplitudes of oscillations during the last portion of the trajectory (when $t=T_1$) are reduced from 7.5 mm for a bang-bang acceleration profile to a 3.1 mm for a sinusoidal

acceleration profile. This reduction is mainly due to the fact that for a bang-bang acceleration profile, the desired position angle $\theta_d(t)$ is a C^2 function[1], while it is a C^3 for a sinusoidal acceleration profile.

4.2.4 Input Command with on a Gaussian Velocity Profile

To overcome the problems caused by discontinuities in the command input, functions with continuous higher derivatives are needed to generate the desired trajectory. An example of such functions is the Gaussian distribution function. A standard form of the Gaussian function is, in general, given by [95]:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2} \qquad 4.16$$

where μ is the mean and σ is the standard deviation of the distribution. The curve of the function $f(x)$ is called the bell shaped curve and it is symmetric with respect to μ. By requiring that the desired velocity of the trajectory to have a bell shaped curve, the discontinuity in the derivative of the acceleration caused by a trajectory with a sinusoidal acceleration profile can be eliminated. A Gaussian velocity profile trajectory can be generated from:

$$\dot{\theta}_d(t) = \frac{\theta_f}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2\sigma^2}(t-\mu)^2} \qquad 4.17$$

The desired acceleration can be obtained from the time derivative of the desired velocity

---

[1] *a C^n function is a function with a continuous n^{th} derivative.*

in Equation 4.17 as:

$$\ddot{\theta}_d(t) = -\frac{\theta_f(t-\mu)}{\sigma^3\sqrt{2\pi}} e^{-\frac{1}{2\sigma^2}(t-\mu)^2} \qquad 4.18$$

It can be seen from Equation 4.18 that the desired acceleration reaches zero *as* $(t-\mu) \rightarrow \infty$, which results in a continuous jerk. The desired position can be obtained by integrating Equation 4.17. The resulting equation involve the error function *erf* and given by:

$$\theta_d(t) = \frac{\theta_f}{2}\left[erf(\frac{t-\mu}{\sigma\sqrt{2}}) - erf(-\frac{\mu}{\sigma\sqrt{2}}) \right] \qquad 4.19$$

Figure 4.8 shows the curves generated form Equations 4.17, 4.18 and 4.19 with σ=1/6 and μ=1.0. In order to get $\theta_d(t)$ to approach $\theta_f = \pi/2$ as $t \rightarrow T_f$ = *1 sec* for the purpose of comparison with the trajectories shown in Figure 4.5 and 4.7, the curves are shifted to the left by Δt=*0.4 sec*. The tip deflection of the one link flexible manipulator during the tracking of the Gaussian velocity profile trajectory is shown in Figure 4.9. The figure shows that the oscillation amplitudes are almost eliminated. In a similar study reported in [88], it was shown that using trajectory with continuous higher derivatives leads to a fast positioning of the tip of flexible link manipulators with a minimum excitation of high frequency modes.

In all the trajectories described earlier, the time required for the joint angle to reach its final destination in almost *1 sec*. As the joint angle moves, the flexible modes are being exited depending on the smoothness (continuity of higher derivatives) of the desired trajectory. When the joint angle θ reaches its final destination with $\dot{\theta} \rightarrow 0$ and

$\ddot{\theta} \rightarrow 0$, the system reduces to:

$$M_f \ddot{\eta} + K\eta + H_f = 0 \qquad 4.20$$

From Equation 3.35, it can be seen that the vector H_f is quadratic in the joint velocity, therefore, as $\dot{\theta} \rightarrow 0$, $H_f \rightarrow 0$. The resulting system representing the dynamics of the flexible link becomes:

$$M_f \ddot{\eta} + K\eta = 0 \qquad 4.21$$

Equation 4.21 represents an undamped and unforced second order system. When such a system is subjected to a non zero initial conditions, its output keeps oscillating around an equilibrium point with constant amplitudes. The amplitudes of these oscillations depend on the tip deflection and tip slope when the joint angle reaches its final destination. When the joint angle is at it its final destination, the actuator at the joint has to continually provide a reacting torque in order to compensate for the disturbance induced by the oscillations at the tip. Setting $\dot{\theta} = \ddot{\theta} = 0$ in Equation 4.1, results in an expression of this reacting torque which is given by:

$$\tau = -M_{rf}^T M_f^{-1} K\eta \qquad 4.22$$

Figure 4.10 shows the torques required to keep the joint angle along the desired trajectories with a bang-bang acceleration profile, a sinusoidal acceleration profile and a Gaussian velocity profile. With reference to the figure, it can be seen that the smoother the desired trajectory, the less torque is required to counteract the effects of the oscillations at the tip and to keep the joint angle along the trajectory.

In the analysis presented in this section, the tip deflection is not used in the feedback control, therefore, Equation 4.21 represents an open loop control system. In order to add robustness to the tip control of the flexible link, the tip deflection need to be included in the controller.

4.3 Non-Collocated Control

End point control of flexible links requires the feedback of the variables representing the motion of the tip. This type of control will be referred to as the non-collocated control because the actuator and the controlled variable are not at the same location. With reference to Figure 4.11, the absolute position angle of any point along the flexible link, under the assumption of small deflection, can be expressed as:

$$\theta_x = \theta + \frac{u(x,t)}{x} \tag{4.23}$$

Substituting for $u(x,t)$ from Equation 3.3, the joint angle θ can be expressed as:

$$\theta = \theta_x - \frac{\Phi^T(x)}{x}\eta \tag{4.24}$$

Substituting for θ in the governing Equations 4.1 and 4.2 gives:

$$M_r\ddot{\theta}_x + (M_{rf}^T - M_r\frac{\Phi^T(x)}{x})\ddot{\eta} + H_r = \tau \tag{4.25}$$

$$M_{rf}\ddot{\theta}_x + (M_f - M_{rf}\frac{\Phi^T(x)}{x})\ddot{\eta} + H_f + \mathrm{K}\eta = 0 \tag{4.26}$$

Using the same procedure followed in deriving Equation 4.4, a single input single output system involving $\ddot{\theta}_x$ and τ can be expressed as:

$$\Gamma_x \ddot{\theta}_x + \Lambda_x = \tau \tag{4.27}$$

where

$$\Gamma_x = M_r - (M_{rf}^T - M_r \frac{\Phi^T(x)}{x}) M_x^{-1} M_{rf} \tag{4.28}$$

and

$$\Lambda_x = H_r - (M_{rf}^T - M_r \frac{\Phi^T(x)}{x}) M_x^{-1} (H_f + \mathrm{K}\eta) \tag{4.29}$$

M_x is a 2 by 2 matrix representing the inertia of the system in Equation 4.2 and given by:

$$M_x = M_f - M_{rf} \frac{\Phi^T(x)}{x} \tag{4.30}$$

When the angle of point P in Figure 4.11 reaches its final position, i.e. $\ddot{\theta}_x = \dot{\theta}_x = 0$, Equation 4.26 is reduced to:

$$M_x \ddot{\eta} + \mathrm{K}\eta = -H_f \tag{4.31}$$

Considering only the homogeneous part of Equation 4.31, and introducing the state vector $z=[\eta^T \quad \dot{\eta}^T]^T$, Equation 4.31 can be expressed in state space form as:

$$\dot{z} = Az \tag{4.32}$$

where

$$A = \begin{bmatrix} 0 & I \\ -M_x^{-1}\mathrm{K} & 0 \end{bmatrix} \tag{4.33}$$

The eigenvalues of A are determined by the parameters of the link and by the position of point P with respect to the joint. The eigenvalues are computed for the one link manipulator with parameters given in Table 3.1, and when the location of P varies from 0 to l. These eigenvalues are shown in Table 4.1. It can be seen in this table that at $x/l=0$, the eigenvalues are all pure imaginary with $\lambda_{1,2}=\pm j35.3$ and $\lambda_{3,4}=\pm j348.0$. As P moves away from the joint and towards the tip, the eigenvalues increase along the imaginary axis in the complex plane. This indicates that as P moves away from the joint, the frequency of the system increases. When x/l is near 0.53, two of the eigenvalues become real with opposite sign. As x/l moves towards the tip, these eigenvalues move towards the origin of the complex plane. Figures 4.12 and 4.13 show the root locus of these eigenvalues as x/l varies from 0 to 1. From the results shown in table 4.1, it can be concluded that the position of the point for which the angle is to be used as an output should be located between 0 and $0.53l$. If this point is located such that $x/l \geq 0.53$, the resulting dynamic system will be unstable. Ideally, and for the purpose of controlling the absolute angle of tip, one would choose the location of P such that $x/l=1$.

x/l	0.00	0.10	0.30	0.4	0.51	0.53	0.89	0.90	1.00
$\lambda_{1,2}$	±j35.3	±j38.9	±j49.5	±j58.7	±j71.5	±15658	±124.9	±3203	± 389
$\lambda_{3,4}$	±j348	±j371	±j461	±j577	±j1351	±j74.8	±j1212	±121.4	± 99
			Stable region						

Table 4.1: Eigenvalues of the one link manipulator as x/l varies from 0 to 1.

By examining the results in table 4.1, it can be seen that two of the four eigenvalues lie in the unstable region of the complex plane. This non-minimum phase characteristic associated with the tip control of robots with flexible links was addressed in [47] using only one parabolic shape function to model a flexible beam. Similar results were also reported in [23].

Figure 4.14 shows the determinant of the matrix M_x as x/l varies between 0 and 1. With reference to this figure, it can be seen that the determinant is zero when $x/l=0.53$. Several simulations have been performed to investigate the response of the flexible link when different points along the link are used. Figure 4.15 shows the tip deflection when at $x/l=0$, $x/l=0.1$ and $x/l=0.3$ are used. The input command is a trajectory with a bang-bang acceleration profile. It can be seen from this figure that for $x/l=0$, and as the controlled angle reaches its final position, the tip keeps oscillating with a frequency ω=35.3 rad/sec and a constant amplitude of nearly 7.5 mm. When the absolute angle of a point located at $x/l=0.1$ is used as the system's output, the frequency of the tip oscillations increases to ω=38.9 rad/sec and the amplitude decreases to 2.4 mm. A further increase in the frequency of oscillation is obtained when $x/l=0.3$ is used which gives a frequency of ω=49.5 rad/sec and an amplitude of nearly 1.1 mm. Similar results are obtained when a trajectory with a sinusoidal acceleration profile is used as the desired input trajectory. These results are shown in Figure 4.17. From this figure, and for $t>1$ *sec*, the maximum deflection is reduced from 3.1 mm when $x/l=0$ to 1.9 mm when $x/l=0.1$. When $x/l=0.3$ is used, the maximum deflection is 1.0 mm. The results obtained from Figures 4.15 and 4.17 are tabulated in Table 4.2. From these results, it can be concluded that using the absolute angle of a point along the link as the system's output increases the stiffness of the flexible link provided that this point results in a stable system.

	Maximum tip deflection in mm for t >1 sec	
x/l	*Bang-Bang acceleration*	*Sinusoidal acceleration*
0	7.5	3.1
0.1	2.4	1.9
0.3	1.1	1.0

Table 4.2: Maximum tip deflection during the last second in Figures 4.15 and 4.17.

The required input torques for desired trajectories with a bang-bang acceleration profile and with a sinusoidal acceleration profile are shown in Figure 4.16 and 4.18, respectively. Since the amplitudes of the oscillations at the tip decreases with an increase in *x/l*, the disturbances caused by these oscillations on the joint angle decrease as well. This causes the input torque required to keep the output along the desired trajectory to decrease. Table 4.3 gives the maximum torque required to keep the controlled angle at its final position when $x/l=0$, $x/l=0.1$ and $x/l=0.3$ are used.

	Maximum input torque in N.m for t >1 sec	
x/l	*Bang-Bang acceleration*	*Sinusoidal acceleration*
0	2.67	1.09
0.1	0.94	0.60
0.3	0.46	0.34

Table 4.3: Maximum input torque during the last second in Figures 4.16 and 4.18.

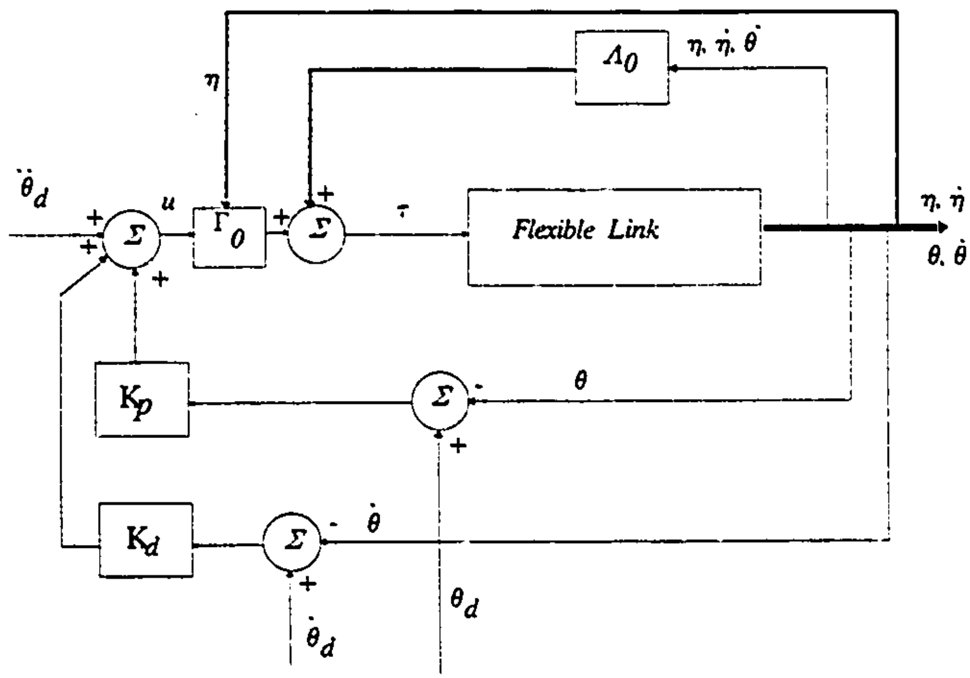


Figure 4.1: Block diagram of a joint based control

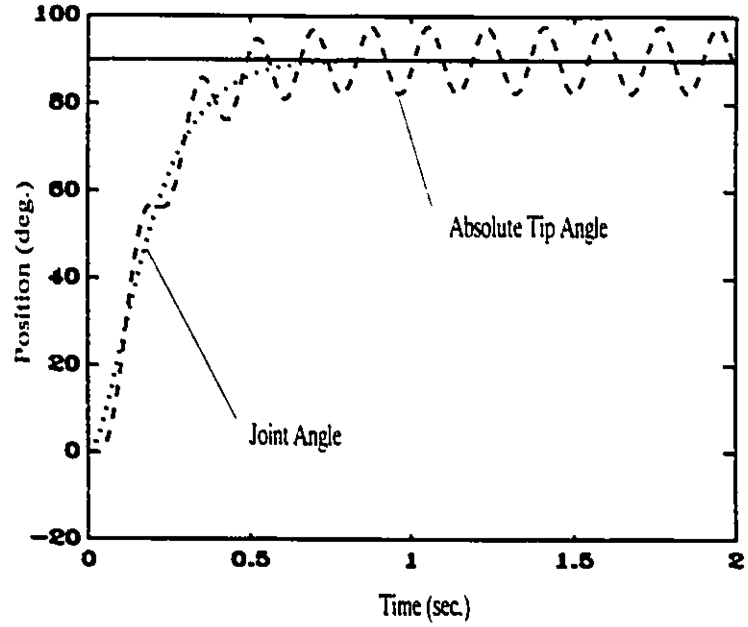


Figure 4.2: Joint Angle and Absolute Tip Angle due to a step input.

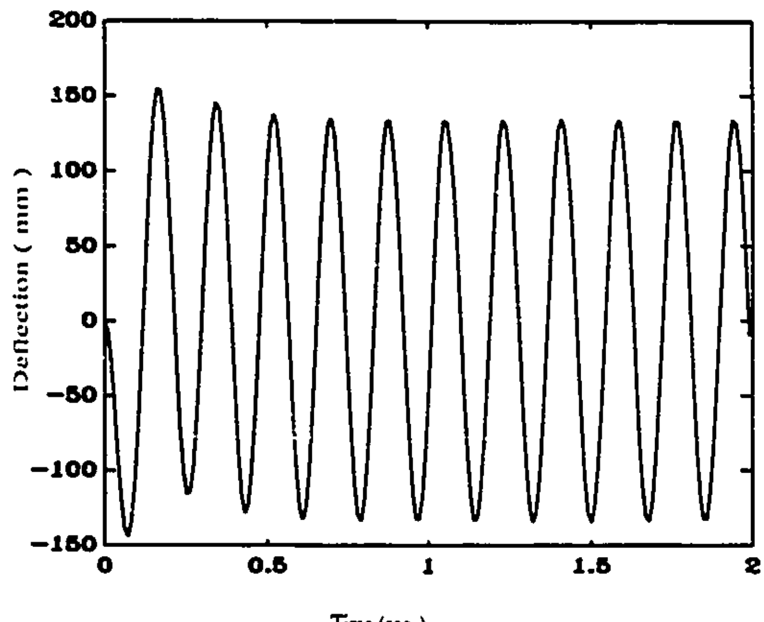


Figure 4.3: Tip deflection with respect to the rigid link due to a step input in the joint angle.

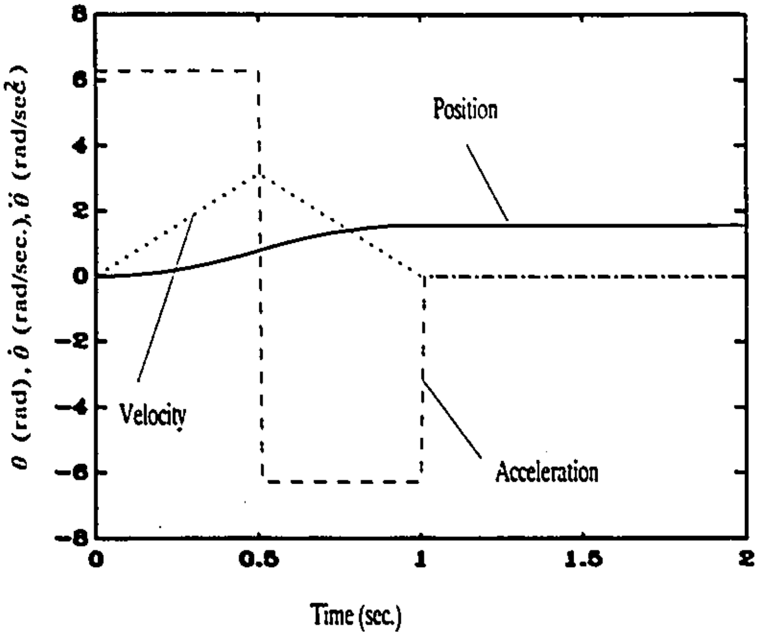


Figure 4.4: Desired trajectories with bang-bang acceleration profile.

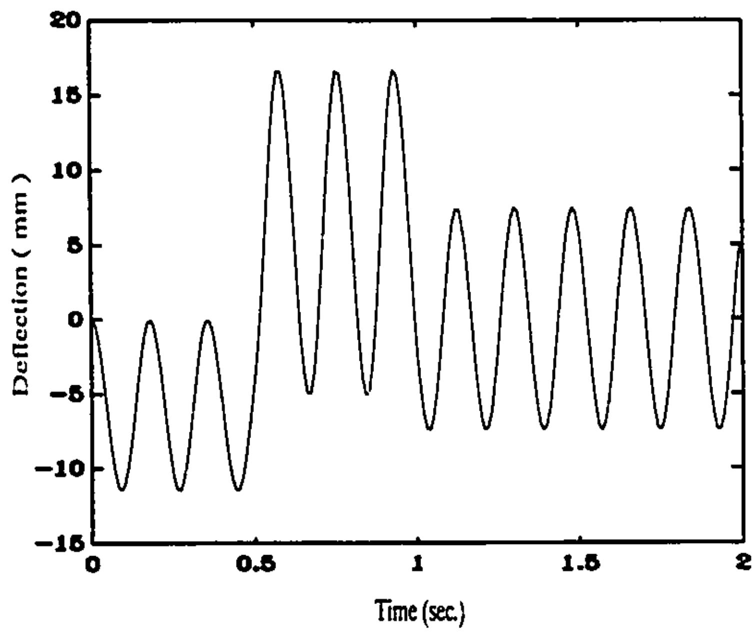


Figure 4.5: Tip deflection due to a bang-bang acceleration trajectory.

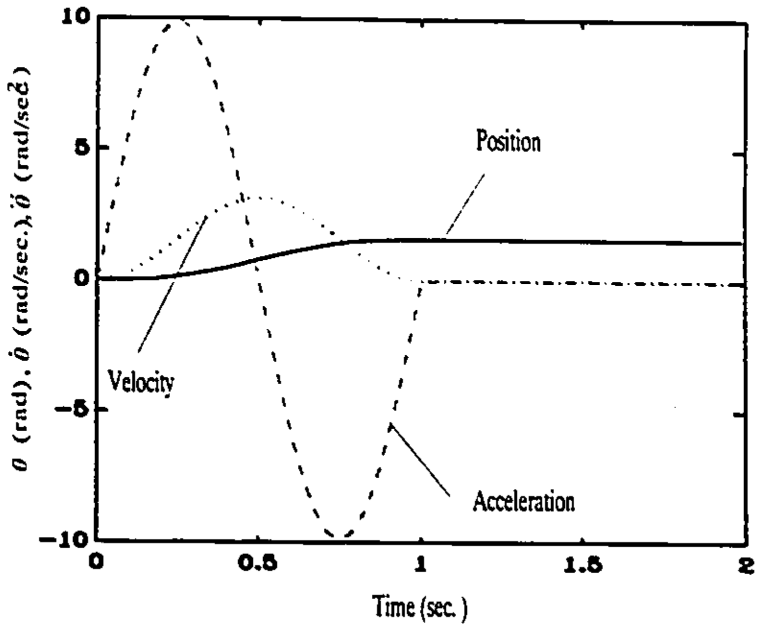


Figure 4.6: Desired trajectory with sinusoidal acceleration profile.

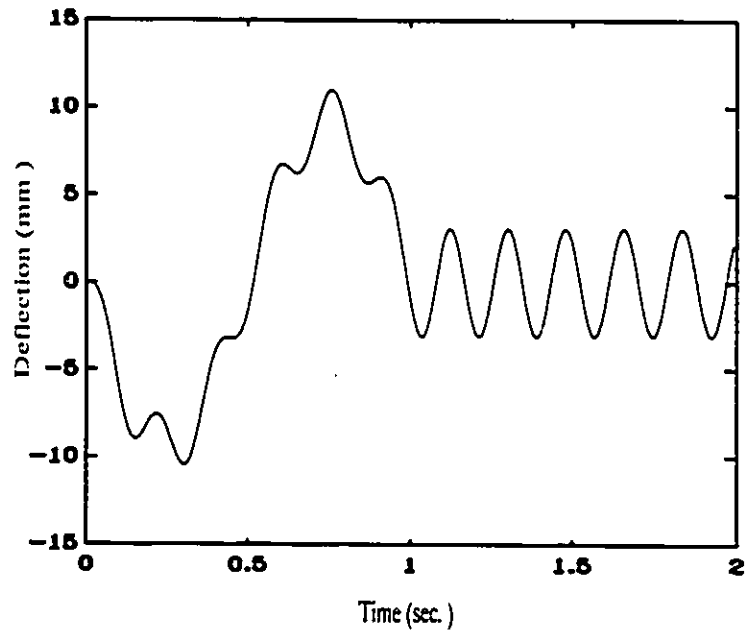


Figure 4.7: Tip deflection due to a trajectory with a sinusoidal acceleration profile.

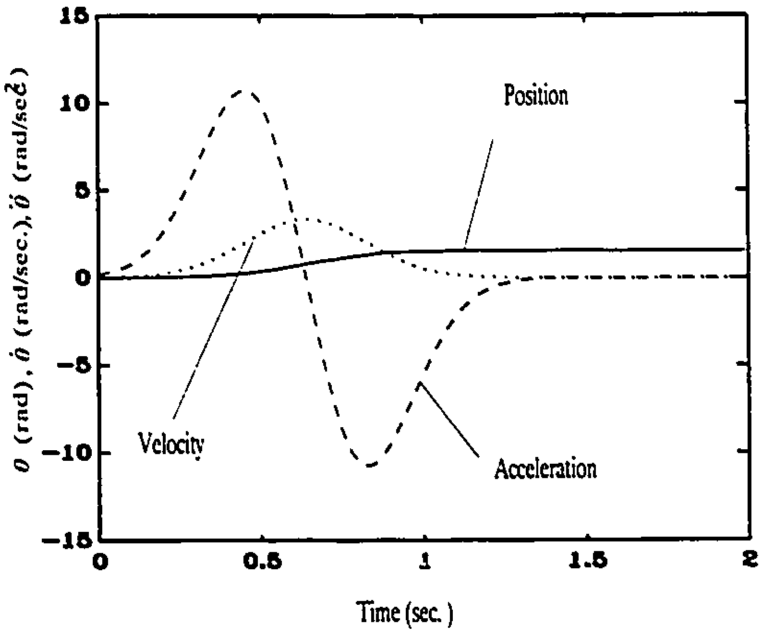


Figure 4.8: Desired trajectory with Gaussian velocity profile.

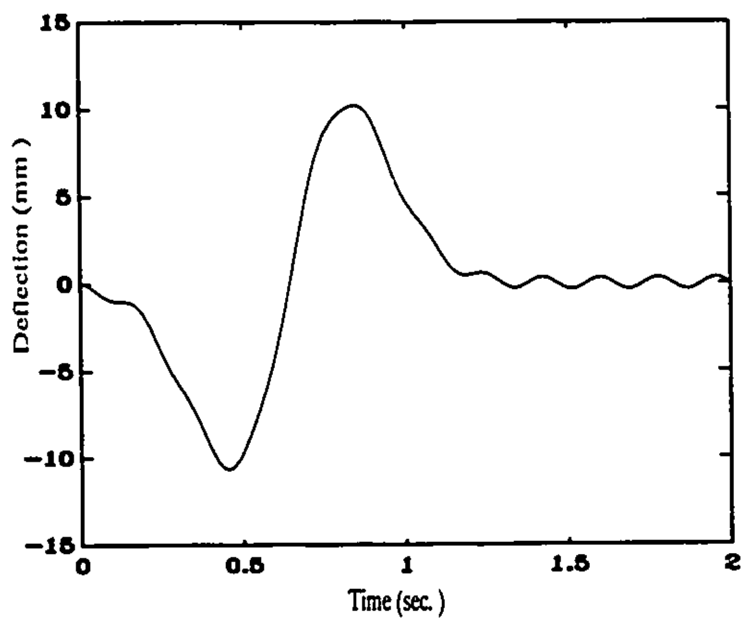


Figure 4.9: Tip deflection due to a trajectory with a Gaussian velocity profile.

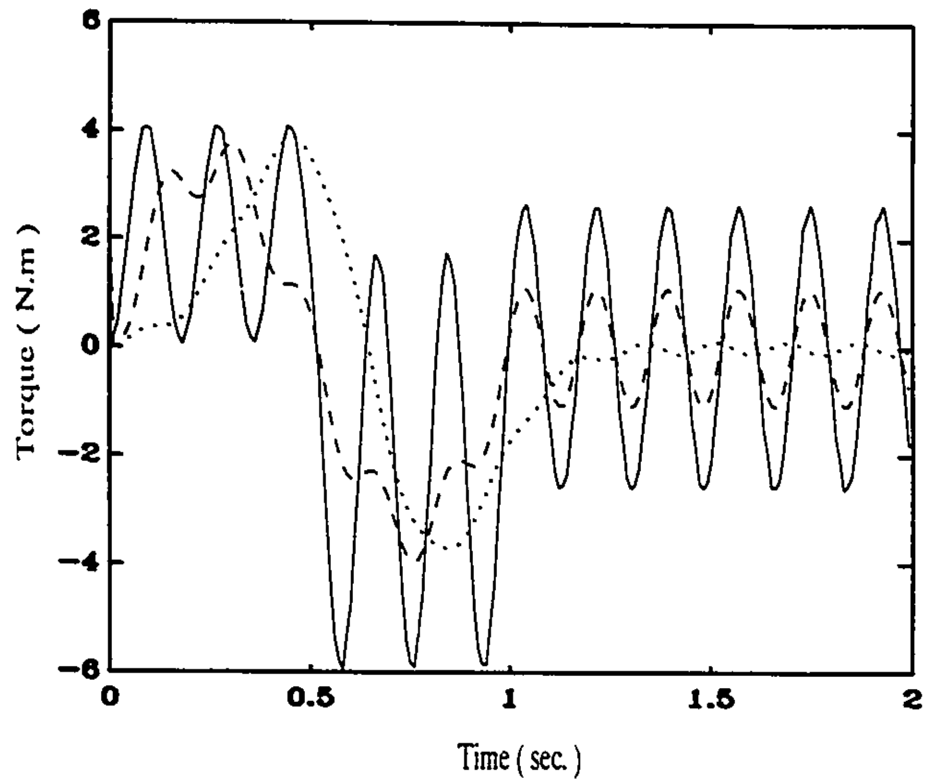


Figure 4.10: Input torque required to keep the joint angle along the desired trajectory.

———— Bang-Bang acceleration trajectory
-------- Sinusoidal acceleration trajectory
.... Gaussian velocity trajectory

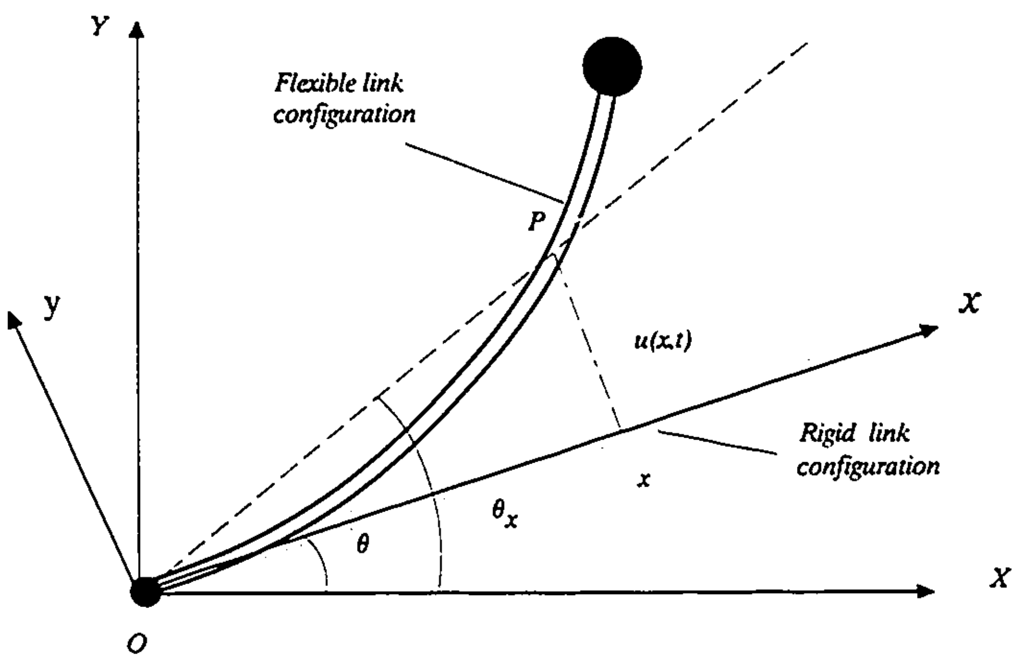


Figure 4.11: Rigid and Deformed Configurations of a one link flexible manipulator.

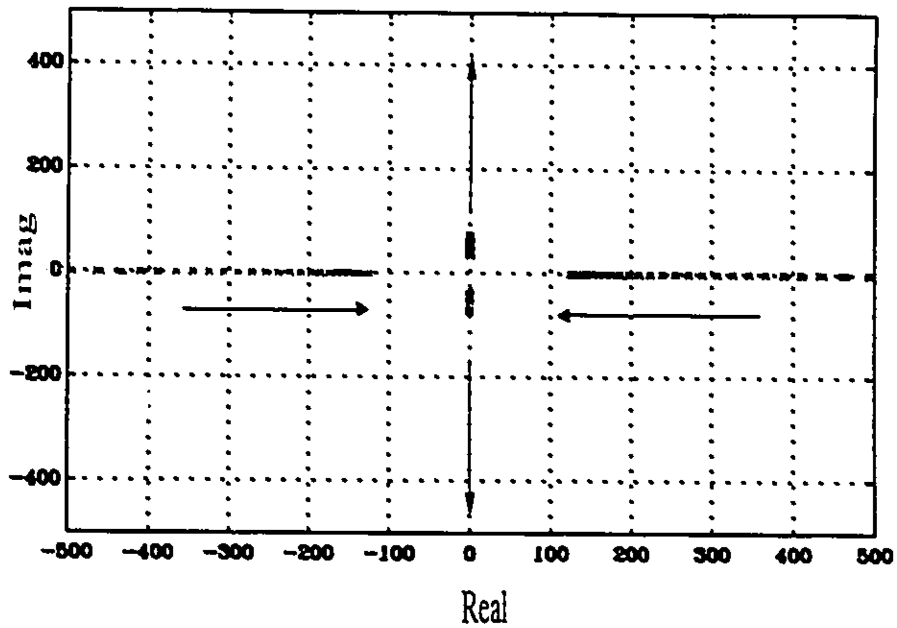


Figure 4.12: Locus of $\lambda_{1,2}$ in the complex plane.

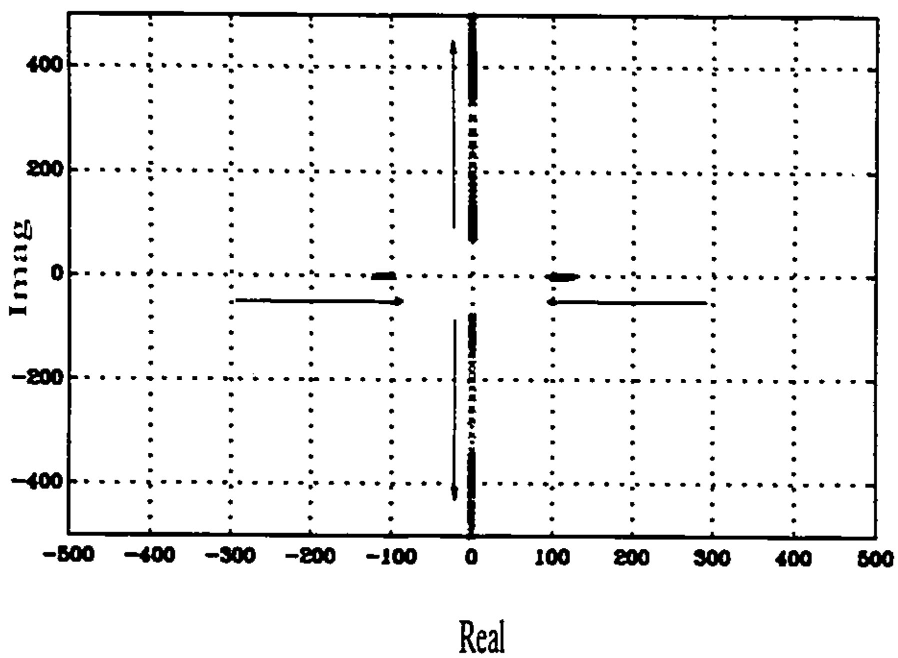


Figure 4.13: Locus of $\lambda_{3,4}$ in the complex plane.

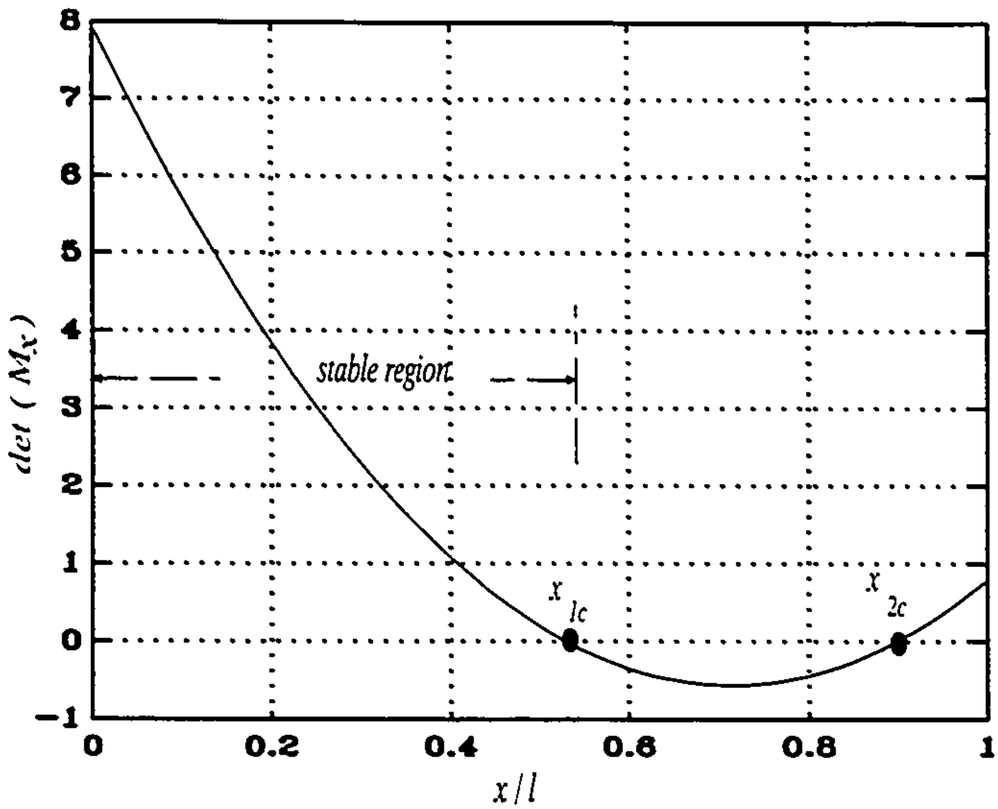


Figure 4.14: Determinant of the matrix M_x v.s x/l.

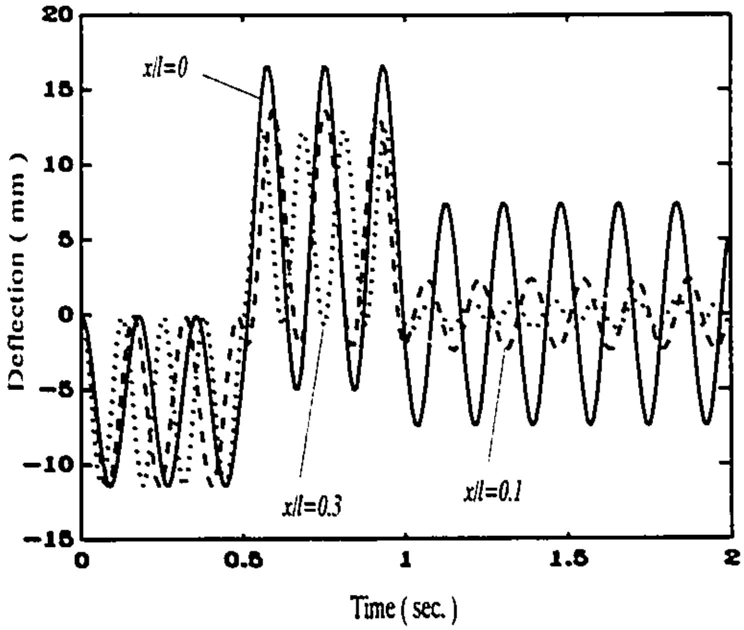


Figure 4.15: Tip deflection for *x/l=0*, *x/l=0.1* and *x/l=0.3*. Desired trajectory with a bang-bang acceleration profile.

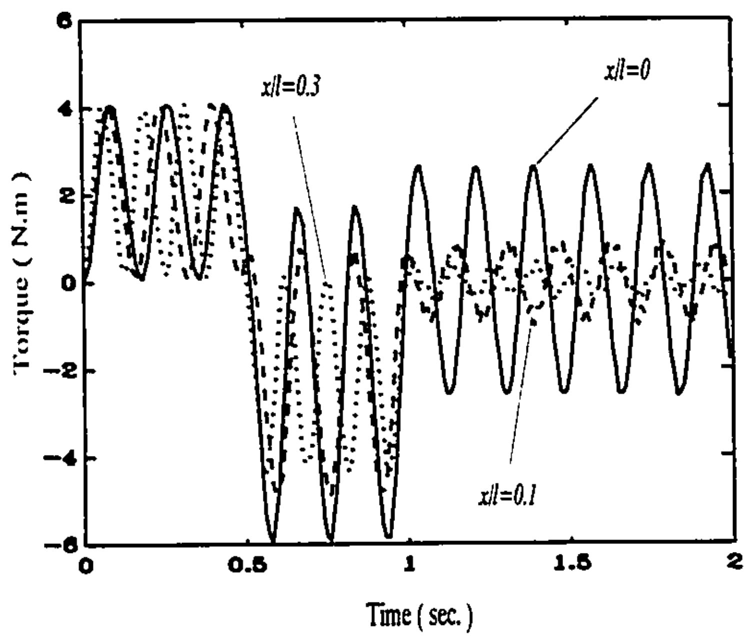


Figure 4.16: Input torque for *x/l=0*, *x/l=0.1* and *x/l=0.3*. Desired trajectory with a bang-bang acceleration profile.

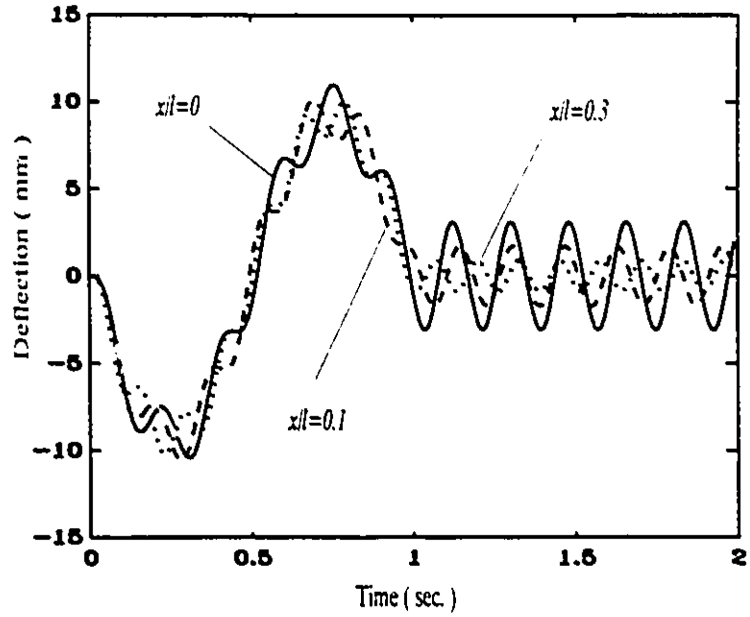


Figure 4.17: Tip deflection for $x/l=0$, $x/l=0.1$ and $x/l=0.3$. Desired trajectory with a Sinusoidal Acceleration profile.

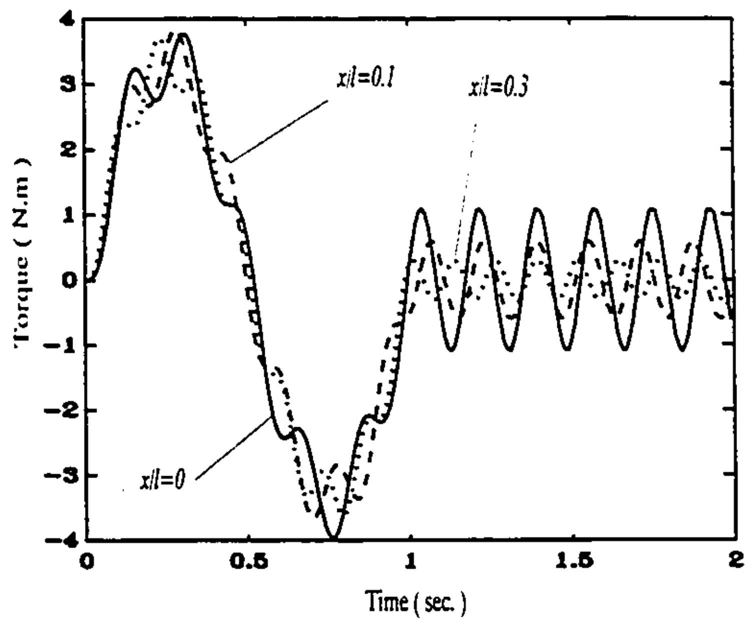


Figure 4.18: Input torque for $x/l=0$, $x/l=0.1$ and $x/l=0.3$. Desired trajectory with a Sinusoidal Acceleration profile.

Chapter 5

Tip Control Using Time Delay

5.1 Introduction

The models of flexible mechanical systems are generally described by partial differential equations which are approximated by a set of linear ordinary differential equations. These ordinary differential equations are usually characterized by very low damping. It was shown in Chapter 4 that when the joint angle of a one link flexible manipulator is used as the controlled variable, and in the absence of any structural damping, the end point keeps oscillating with respect to the hypothetical rigid link at constant amplitude. It was also shown that even though the amplitude is reduced when the angle of a point along the link is used as the controlled variable, the oscillations at the tip persist and never die down. In order to introduce active damping to the modes of the flexible structure, time delay has been introduced in the control algorithm. In this Chapter, an overview of Time Delay Control is presented in Section 5.2. In Section 5.3, a Computed Torque/Time Delay approach to the end point control of a one link flexible manipulator is introduced. Simulation results showing how a given delay time affects the oscillations at the tip when the angle of different points along the link is used as the controlled variable. The effects of different delay times on the tip response are also investigated when the angle of a given point along the link is used as a controlled variable.

5.2 Time Delay Control: An overview

5.2.1 Posicast Control

One of oldest time delay control approaches was introduced by O. J. Smith [94]. In this technique, the input command is divided into several segments such that the sum of all the transient terms is zero after the last input. In order to demonstrate this control strategy, a second order linear system is used. The transfer function of such a system is given by:

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{5.1}$$

The system is assumed to be underdamped; $0 \le \zeta < 1$ and the frequency at which the system oscillates is given by $\omega = \omega_n\sqrt{1-\zeta^2}$. It was shown in [91] that if a step input $r(t)=R$ is divided into two inputs $r_1(t)=R_1$ and $r_2(t)=R_2$, such that:

$$R_1 + R_2 = R \tag{5.2}$$

and

$$R_2 = R_1 e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \tag{5.3}$$

and if the input $r_1(t)$ is applied at $t=0$ and the input $r_2(t)$ is applied at $t=\pi/\omega$, the output $c(t)$ will reach the input without any oscillations. For a unit step input ($R=1$), R_1 and R_2 can be obtained from:

$$R_1 = \frac{1}{1 + e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}}} \tag{5.4}$$

$$R_2 = \frac{e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}}}{1 + e^{\frac{\zeta\pi}{\sqrt{1-\zeta^2}}}} \tag{5.5}$$

Figure 5.1 shows the response of the system given by Equation 5.1 when $\zeta=0.1$ and $\omega_n=30$ rad/sec. From Equations 5.4 and 5.5, the magnitudes of R_1 and R_2 are found to 0.578 and 0.402, respectively. The time by which the input R_2 should be delayed is $v=0.105$ sec. It can be seen from this figure that the output $c(t)=c_1(t)+c_2(t)$ reaches the input $r(t)=r_1(t)+r_2(t)$ with out any oscillation. This is in contrast to the well known response that would result from applying $R=1$ as the input to the system. Even though a perfect cancellation of the output oscillations can only be achieved with exact knowledge of the parameters of the system, a small mismatch in the delay time would also lead to considerable oscillation reduction. Figures 5.2 and 5.3 show the response of the second order system when an error of -5% and +5% was introduced in the delay time, respectively.

5.2.2 Input/Output Linearization Using Time Delay.

Consider a dynamic system described by:

$$M\ddot{q} + H = \tau \qquad 5.6$$

It was shown in Chapter 3 that if M and H can be accurately determined, a non linear controller that can reproduce any smooth trajectory described by $\ddot{q}_d$, $\dot{q}_d$ and q_d is given by:

$$\tau = M(\ddot{q}_d + K_d(\dot{q}_d - \dot{q}) + K_p(q_d - q)) + H \qquad 5.7$$

In most cases, accurate estimates of M and H are very difficult to obtain, especially when the system is non linear and time varying. In the case of manipulators with flexible links, estimating M and H is even more difficult because of the infinite number of degrees of freedom needed to fully describe the motion of the links. The matrix M in Equation 5.7 can be divided into two parts as:

$$M = \hat{M} + \Delta M \qquad 5.8$$

where $\hat{M}$ is a constant diagonal matrix representing the known inertia of the system and ΔM is the matrix representing the time varying and unknown inertia. In the case of a one link flexible manipulator, $\hat{M}$ can be taken as the inertia of the equivalent rigid link.

Substituting 5.8 into 5.6 and rearranging gives:

$$\hat{M}\ddot{q} + D = \tau \tag{5.9}$$

where $D = \Delta M\ddot{q} + H$ is the term representing the unknown dynamics, or terms that are difficult to estimate. τ can be computed based on Equation 5.7 as:

$$\tau = \hat{M}((\ddot{q}_d + K_d(\dot{q}_d - \dot{q}) + K_p(q_d - q)) + D \tag{5.10}$$

From Equation 5.9, the term D at a given time t can be written as:

$$D(t) = \tau(t) - \hat{M}\ddot{q}(t) \tag{5.11}$$

The input/output linearization using time delay consists of estimating the term $D(t)$ by its delayed value $D(t-\nu)$ for a very small delay time ν. From Equation 5.11, the delayed term $D(t-\nu)$ can be written in terms of the delayed input $\tau(t-\nu)$ and the delayed acceleration $\ddot{q}(t-\nu)$ as:

$$D(t-\nu) = \tau(t-\nu) - \hat{M}\ddot{q}(t-\nu) \tag{5.12}$$

Replacing $D(t)$ in Equation 5.9 by $D(t-\nu)$ gives:

$$\begin{aligned}\tau(t) &= \hat{M}(\ddot{q}_d(t) + K_d(\dot{q}_d - \dot{q})) + \tau(t-\nu) - \hat{M}\ddot{q}(t-\nu) \\ &= \hat{M}\upsilon + \tau(t-\nu) - \hat{M}\ddot{q}(t-\nu)\end{aligned} \tag{5.13}$$

where $\upsilon = \ddot{q}_d(t) + K_d(\dot{q}_d(t) - \dot{q}(t)) + K_p(q_d(t) - q(t))$

The control input in Equation 5.13 does not require the knowledge of the dynamics of the system. Only a rough estimate of the inertia matrix is required. The main requirement for this controller to give satisfactory results is based on the choice of the delay time ν. The accuracy of the estimated unknown dynamics improves as ν decreases. In practice, and when a digital controller is used, ν can be taken as the sampling period. Clearly, the stability of the controller depends on the choice of the constant diagonal matrix $\hat{\boldsymbol{M}}$. It was shown in [31] that a system such as the one given by Equation 5.6 is asymptotically stable if the matrix $\hat{\boldsymbol{M}}$ is selected such that all the eigenvalues of the matrix $\boldsymbol{I}-\boldsymbol{M}^{-1}(k)\hat{\boldsymbol{M}}$ are inside the unit circle for $k \geq 0$.

5.2.3 Proportional Plus Delayed Control (PDC)

In this method, the control input is calculated based on the present output plus the sum of the last n outputs. To illustrate this type of control, a pure inertia system with an inertia J is considered. The input torque τ of the motor driving the system is calculated by:

$$\tau(t)=K_p \sum_{i=0}^{n} \alpha_i (r(t-\nu_i) - c(t-\nu_i)) \qquad 5.14$$

where the ν_i's are the delay time with $\nu_0=0$. K_P is the proportionality gain and α_i is a coefficient associated with each delay time. The closed loop dynamic equation of the system with the inertia J can be written as:

$$J\ddot{c}(t) = K_p \sum_{i=0}^{n} \alpha_i (r(t-\nu_i) - c(t-\nu_i)) \qquad 5.15$$

The delayed output $c(t-\nu_i)$ can be expressed as a Taylors series:

$$c(t-\nu_i) = \sum_{k=0}^{\infty} \frac{(-\nu_i)^k}{k!} c^{(k)}(t) \qquad 5.16$$

where $c^{(k)}$ is the kth derivative of *c(t)* with respect to time. Substituting the series into Equation 5.15 and rearranging gives:

$$J\ddot{c}(t) + K_p \sum_{i=0}^{n} \sum_{k=0}^{\infty} \alpha_i \frac{(-\nu_i)^k}{k!} c^{(k)}(t) = K_p \sum_{i=0}^{n} \alpha_i r_d(t-\nu_i) \qquad 5.17$$

A first order approximation of the Taylor's series in Equation 5.16 can be obtained by ignoring all higher derivatives. The resulting Equation is given by:

$$c(t-\nu_i) \approx c(t) - \nu_i \dot{c}(t) \qquad 5.18$$

The equation resulting from substituting Equation 5.18 into 5.15 can be cast in the form of a second order system as:

$$\ddot{c} + 2\xi \omega_n \dot{c} + \omega_n^2 c = f \qquad 5.19$$

where

$$2\zeta \omega_n = -\frac{K_p}{J} \sum_{i=0}^{n} \alpha_i \nu_i \qquad 5.20$$

$$\omega_n^2 = \frac{K_p}{J}\sum_{i=0}^{n} \alpha_i \qquad 5.21$$

$$f = \frac{K_p}{J}\sum_{i=0}^{n} \alpha_i r(t-\nu_i) \qquad 5.22$$

From Equations 5.20 and 5.21, it can be deduced that the stability of the system necessitates that: (*i*) $\sum_{i=0}^{n} \alpha_i > 0$ and (*ii*) $\sum_{i=0}^{n} \alpha_i \nu_i < 0$. Furthermore, the α_i's must be decreasing in order to give more weight to the more recent outputs. A plausible choice of α_i that would satisfy conditions *(i)* and *(ii)* is $\alpha_i = (-1)^i |\alpha_i|$. This results in the alternating series described by Leibnitz theorem [92].

Theorem 1 (Leibnitz test for real series):

Let a_0, a_1, ..., a_n be real and

$$(1)\ a_0 \geq a_1 \geq a_2 \geq \ldots; \qquad (2)\ \lim_{n\to\infty} a_n = 0 \qquad 5.23$$

Then

$$0 \leq \sum_{i=0}^{n} (-1)^i a_i \leq a_0 \qquad 5.24$$

The delay times ν_i's are selected such that:

$$...\nu_n > \nu_{n-1} > ... > \nu_1 > \nu_0 = 0 \qquad 5.25$$

The expression given by Equation 5.25 and the *Leibnitz theorem* lead to the following proposition.

Proposition:

For the real set a_0 . a_1 ,...., a_n satisfying the Leibnitz conditions, and for the real set $\nu_0, \nu_1, \nu_2,...$ satisfying Equation 5.25:

if $\frac{a_{i-1}}{a_i} < \frac{\nu_i}{\nu_{i-1}}$; $\quad i=1,3,5,....$, then the series given by $S_n = \sum_{i=0}^{n} (-1)^i a_i \nu_i$ is always negative.

Finally, a theorem governing the stability of the system under the proportional plus the delayed control can be stated as follows:

Theorem 2: The system given by

$$J\ddot{c}(t) + K_p \sum_{i=0}^{n} \sum_{k=0}^{\infty} \alpha_i \frac{(-\nu_i)^k}{k!} c^{(k)}(t) = K_p \sum_{i=0}^{n} \alpha_i r_d(t-\nu_i)$$

is stable if:

$$\sum_{i=0}^{n} \alpha_i > 0 \text{ and } \frac{|\alpha_{i-1}|}{|\alpha_i|} < \frac{\nu_i}{\nu_{i-1}}; \qquad i=1,3,5,....$$

where $\alpha_i=(-1)^i\,|\alpha_i|$ and $\nu_n > \nu_{n-1} > \ldots > \nu_0 = 0$.

From Equations 5.20 and 5.21, the natural frequency and the damping ration of the inertia system under the PDC are given by:

$$\omega_n = \sqrt{\frac{K_p \sum \alpha_i}{J}} \qquad 5.27a$$

and

$$\zeta = \frac{\sum \alpha_i \nu_i}{2} \sqrt{\frac{K_p}{J \sum \alpha_i}} \qquad 5.27b$$

Its clear from Equation 5.27b that without any delay in the feedback, the damping ratio is $\zeta=0$. If at least one delay is used in the feedback, one can select the gain K_p, the coefficients α_i and the delay times ν_i's to target a desirable performance specification.

5.3 Computed Torque/Delayed Deflection Control (CTDD)

It has been shown in Section 4.3 that when the angle of a point along the link is used as the system's output, the flexible link appears to be stiffer resulting in a decrease in the amplitudes of the oscillations at the tip. It was also shown in the literature that using a delayed output in the feedback introduces active damping into the system. In this respect, and to make use of the added stiffness resulting from the non-collocated control, a Computed Torque/Delayed Deflection controller (CTDD) is introduced. In this type of controller, the actuator's torque τ is given by:

$$\tau = \Gamma_x(\ddot{\theta}_d + K_p(\theta_d - \theta_x) + K_d(\dot{\theta}_d - \dot{\theta}_x)) + \Lambda_x \tag{5.28}$$

where $\theta_x(t)$ is calculated based on the present joint angle and the delayed deflection $u(x,t-\nu)$ of a point along the link located at a distance x from the joint as:

$$\begin{aligned} \theta_x(t) &= \theta(t) + \frac{u(t-\nu_i)}{x} \\ &= \theta(t) + \frac{\Phi^T(x)}{x}\eta(t-\nu_i) \end{aligned} \tag{5.29}$$

A block diagram of a flexible link manipulator under the Computed Torque/Delayed Deflection control is shown in Figure 5.4. The main reason behind using only the delayed deflection and not the delayed angle $\theta_x(t-\nu_i)$ is to introduce active damping into the flexible modes without slowing down the joint angle.

5.3.1 Selection of Time Delay

To further investigate the effect of using delayed signals on the performance of the system and determine a relation between the delay time and the system's parameters, a simple time delay controller is being considered in this section. This controller is given in the Laplace domain by:

$$\frac{U(s)}{R(s)} = \alpha_0 + \sum_{i=1}^{n} \alpha_i e^{-s\nu_i} \tag{5.30}$$

$R(s)$ is the controller input and $U(s)$ is the controller output. ν_i is the delay time of the delayed signal, and α_i is its amplitude. The zeros of the controller are given by:

$$\alpha_0 + \sum_{i=1}^{n} \alpha_i e^{-s\nu_i} = 0 \qquad 5.31$$

Replacing the Laplace variable s in Equation 5.31 by $s = \sigma + j\omega$ gives:

$$\alpha_0 + \sum_{i=1}^{n} \alpha_i e^{-\sigma\nu_i} e^{-j\omega\nu_i} = 0 \qquad 5.32$$

Expressing the term $e^{-j\omega\nu_i}$ in Equation 5.32 by $\cos\omega\nu_i - j\sin\omega\nu_i$ and equating the real and imaginary parts to zero gives:

$$\alpha_0 + \sum_{i=1}^{n} \alpha_i e^{-\sigma\nu_i} \cos\omega\nu_i = 0 \qquad 5.33$$

$$\sum_{i=1}^{n} \alpha_i e^{-\sigma\nu_i} \sin\omega\nu_i = 0 \qquad 5.34$$

In particular, if only one time delay is used (n=1), the solution of Equation 5.34 reduces to:

$$\omega = (2k+1)\frac{\pi}{v_1} \qquad k=0,1,\ldots\ldots \qquad 5.35$$

If this controller is used to control a plant having complex poles at $s = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$, then the controller frequency has to match the frequency of plant. From Equation 5.35 and for *k=0*, a relation between the frequency of the plant and the time delay is given by:

$$v_1 = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}} \qquad 5.36$$

Substituting Equation 5.36 in Equation 5.33 and replacing σ by $-\zeta\omega_n$ result in a relation between α_0 and α_1 which is given by:

$$\frac{\alpha_0}{\alpha_i} = e^{\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \qquad 5.37$$

5.3.2 Simulations for a one link flexible manipulator using the CTDD controller

The one link flexible manipulator with the physical parameters shown in Table 3.1 is used in the simulation with a sampling time of 0.001 sec. The results shown in Figures 5.5 to 5.8 are obtained using a delay time *v=0.03 sec* to control a one link flexible manipulator along a trajectory having a sinusoidal acceleration profile. Since no structural damping was introduced in the system,(i.e. $\zeta=0$), the frequency that can be cancelled by the time delay controller can be obtained from Equation 5.36 as $\omega=104.67$ *rad/sec*. The deflection at several points along the link are used in the feedback with a delay time of *0.03 sec*. Figures 5.5 to 5.8 show the deflection at the tip and the joint torque for $x/l=0.1$; $x/l=0.3$ and $x/l=0.5$. With reference to Table 4.1, the frequencies corresponding to $x/l=0.1$; $x/l=0.3$ and $x/l=0.5$ are *38.9 rad/sec*; *49.5 rad/sec* and *70.4 rad/sec*, respectively. From Figure 5.6 it can be seen that using the delayed deflection of a point along the link adds more damping to the flexible link. Furthermore, the amount of damping increases as the frequency of the system approaches the frequency to be cancelled by the time delay. The input torque is also reduced as a direct result of the reduction in the oscillations at the tip as shown in Figures 5.7 a 5.8.

5.3.2.1 Effects of the delay time on the system performance

From the above discussion, It can be concluded that for a given x/l, there is a specific delay time to be used for maximum reduction in the oscillations at the tip. This delay time is given by Equation 5.36. In order to see the effects of a mismatch between the required delay time and the delay used by the controller, several delay times has been used to control the one link flexible manipulator when x/l is fixed. Figures 5.9 and 5.10 show the tip deflection and the input torque during the tracking of a desired trajectory with a bang-bang acceleration profile. x/l is set to *0.4*. This corresponds to $\omega=58.1$ *rad/sec* which in turns leads to a delay time of 0.054 sec. The delay times used in the simulation are 0.01 sec, 0.054 sec, and 0.1 sec. With reference to the figures, it can be seen that the best performance is obtained when a delay time of 0.054 sec is used. A desired trajectory with a sinusoidal acceleration profile is also used to investigate the response of the one link flexible manipulator when different delay times are used and the results are shown in Figures 5.11 and 5.12.

In practice, and when the dynamics of the system become more and more complicated, the frequency of the system becomes difficult to obtain. A good estimate of this frequency can be obtained by using a series of delays instead of a single one [48,64,93]. When a series of delays is used, the angle $\theta_x(t)$ in Equation 5.20 is calculated by:

$$\theta_x(t) = \theta(t) + \frac{\Phi^T(x)}{x} \sum_{i=1}^{n} \alpha_i \eta(t - \nu_i) \qquad 5.38$$

$\eta(t-\nu_i)$ is the deflection at the point located at x/l delayed by ν_i, and α_i is the corresponding coefficient satisfying $\sum_{i=1}^{n} \alpha_i = 1$. In the case of a series with three delays [64], the angle $\theta_x(t)$ is given by:

$$\theta_x(t) = \theta(t) + \frac{\Phi^T(x)}{x}(\alpha_1 \eta(t - \nu_1) + \alpha_2 \eta(t - \nu_2) + \alpha_3 \eta(t - \nu_3)) \qquad 5.39$$

where α_1=1/4; α_2=1/2 and α_3=1/4.

Figures 5.13 and 5.14 show the tip deflection and the input joint torque during the tracking of a desired trajectory with a bang-bang acceleration profile. These figures show a significant improvement in the response of the tip when a series of delays is used. The oscillations at the tip seem to decay much faster than when a single delay is used (refer to Figure 5.9). Using a series of delays seems to provide enough damping in the flexible link even without tip velocity feedback as shown in Figure 5.13. Figure 5.14 shows the joint reaction torque corresponding to oscillation shown in Figure 5.13. Similar results were obtained when a desired trajectory with a sinusoidal acceleration profile was used. The tip oscillation and the joint reaction torque are shown in Figures 5.15 and 5.16 respectively.

5.3.2.2 Effect of a payload at the tip on the system performance

The results shown so far are for a one link flexible manipulator with no payload at the tip. The presence of a payload at the tip of the link affects the frequency of the system which in turns affects the required delay time obtained from Equation 5.36. Figure 5.17 shows the relation between the payload m_L and the delay time ν for several x/l. It can be seen from this figure that for a given x/l, the required delay time increases with an increase in the payload. This increase in the required delay time is a direct result of reduction of the frequency of the system caused by the payload at the tip. Figures 5.18 and 5.19 show the tip deflections during the tracking of trajectories with a bang-bang acceleration profile and a sinusoidal acceleration profile, respectively. The results in these two Figures are obtained using an x/l=0.4 and a delay time of 0.054 sec. As shown earlier, the delay time of 0.054 sec is the required delay time when x/l=0.4 and m_L=0.

As the payload increases to 0.1 *kg*, the deflection at the tip increases and oscillations are observed during the last second of the simulation. A further increase in the tip deflection is observed when the payload is increased to 0.2 *kg*.

5.4 Comparison of the CTDD with other control methods

In this section the proposed CTDD approach is compared with a composite controller and a state feedback regulator presented by B. Siciliano and W. J. Book [62]. The composite controller consists of two part, one deals with the control of the slow subsystem which is represented by the equivalent rigid link, and the other deals with the stabilization the fast subsystem around the equilibrium trajectory set by the slow subsystem. The physical parameters of the one link flexible manipulator used in [62] are shown below:

Length of the link:	$l = 4\ ft$
Mass of the link:	$m = 0.005124\ lbf.s^2/ft^2$
Flexural Rigidity:	$EI = 28.6\ lbf.s^2/ft^4$
Mass of the payload:	$M_L = 0.0031\ lbf.s^2/ft$
Inertia of the payload:	$I_L = 0.00852\ s^2.ft$

The trajectory used in the simulation consists of moving the joint angle from an initial position $\theta_i=0$ to a final position $\theta_f=90$. The desired velosity is given by:

$$\dot{\theta} = 90[1 + \sin(360t/T - 90)]\ ^\circ/s \qquad 5.40$$

In the CTDD approach, the deflection of a point located at $x=0.6l$ from the joint is used in the feedback. A time delay of 0.27 sec was found to be sufficient to cancel the residual vibrations at the tip of the link. This delay represents half the period of the lowest frequency of the link.

The simulation results of the one link flexible manipulator are shown in Figures 5.20 to 5.23. It can be seen from these figures that the performance of the CTDD approach is much better than that of the state regulator. On the other hand, a very similar performance is obtained when the CTDD is compared with the composite controller. The dynamic stiffening effect introduced by non collocation causes some ripples in the deflection and the torque responses. These ripples are then eliminated through the time delay.

Even though the responses of the joint angle and the tip position have shown to be very similar for the composite controller and the for the CTDD, the later has a smaller settling time as shown in Figures 5.22 and 5.23. Another advantage of the CTDD over the composite controller is that the CTDD is much simpler and easier to implement. The only parameter that the CTDD needs is the delay time or a series of delays in order to increase robustness. The composite controller, however, requires a complete stability analysis based on singular perturbation theory.

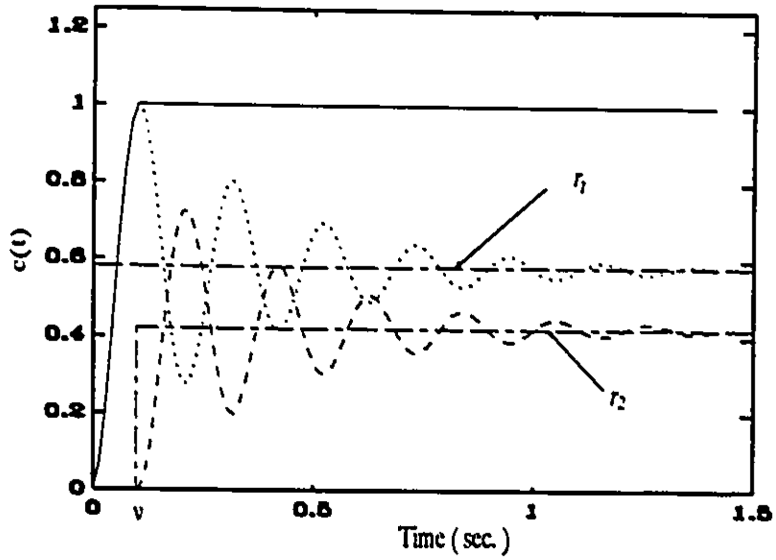


Figure 5.1: Posicast control of a linear second order system with a delay time of $v = \pi/\omega$.

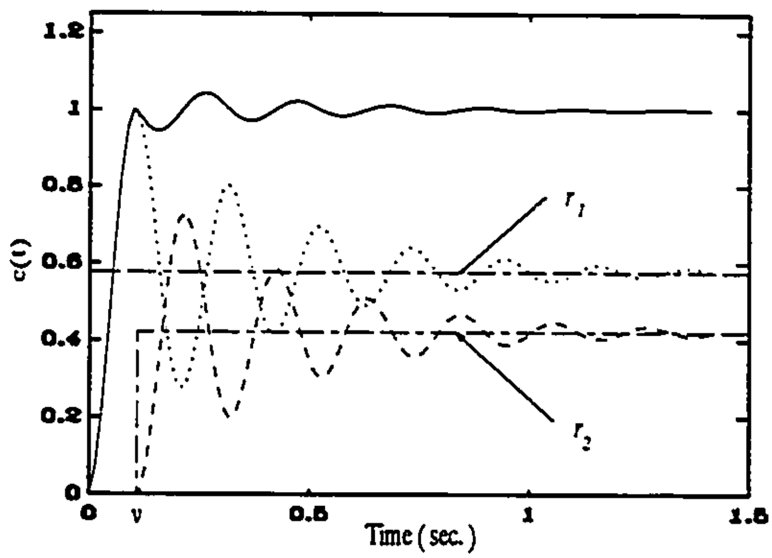


Figure 5.2: Posicast control of a linear second order system with a delay time mismatch. ($v = 0.95\pi/\omega$).

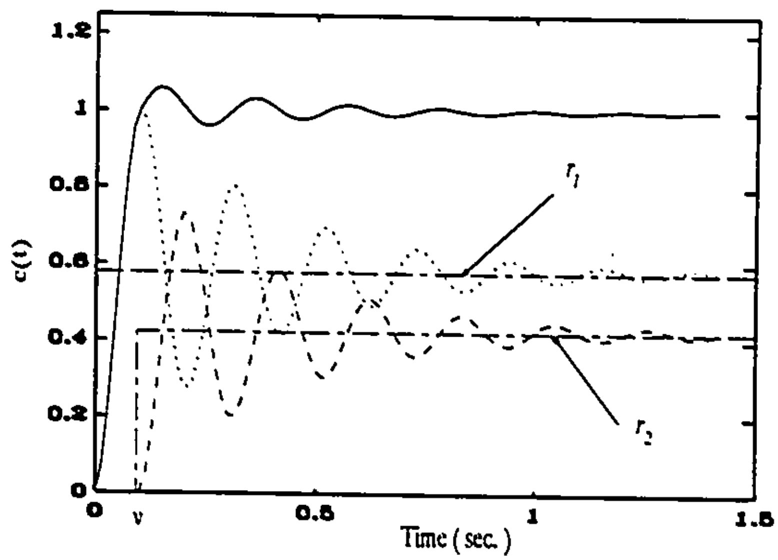


Figure 5.3: Posicast control of a linear second order system with a delay time mismatch ($\nu = 1.05\ \pi/\omega$).

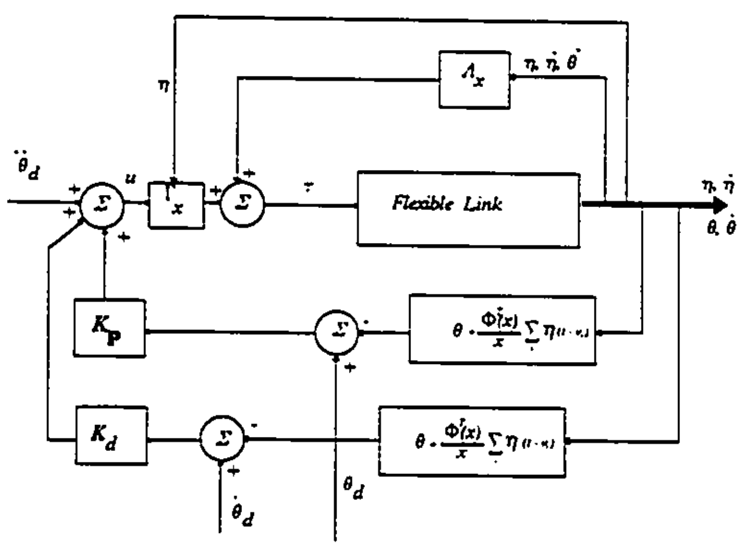


Figure 5.4: Block Diagram of the Computed Torque/Delayed Deflection Approach.

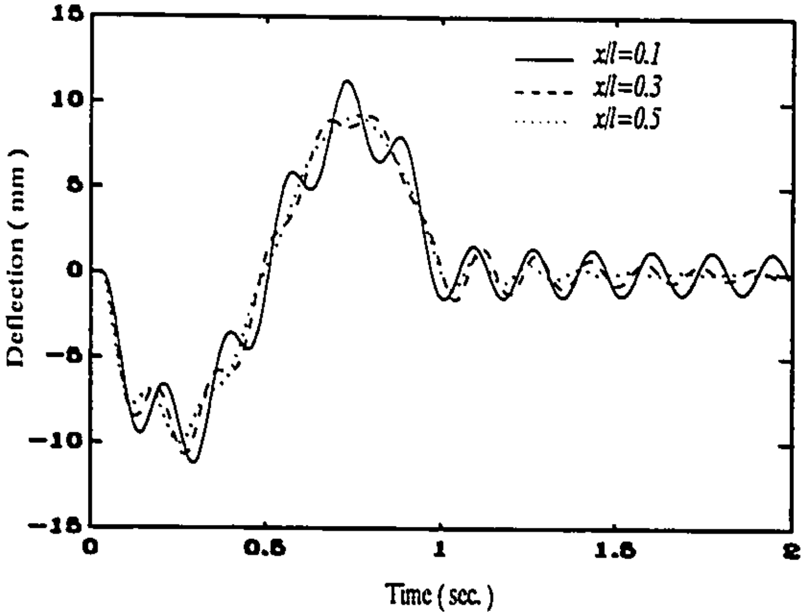


Figure 5.5: Tip deflection due to the tracking of a desired trajectory with a sinusoidal acceleration profile using the CTDD approach and a time delay v=0.03 sec.

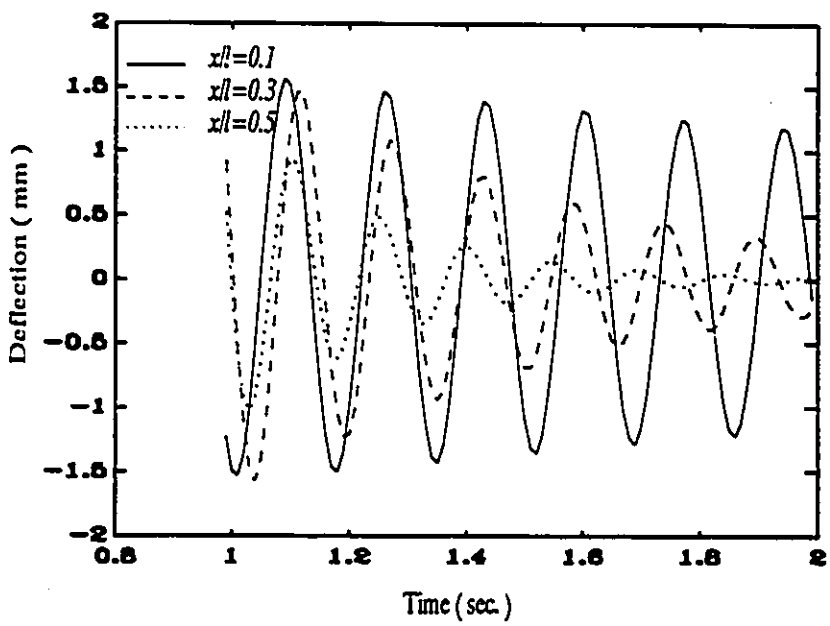


Figure 5.6: Zoom out of the last second in Figure 5.5.

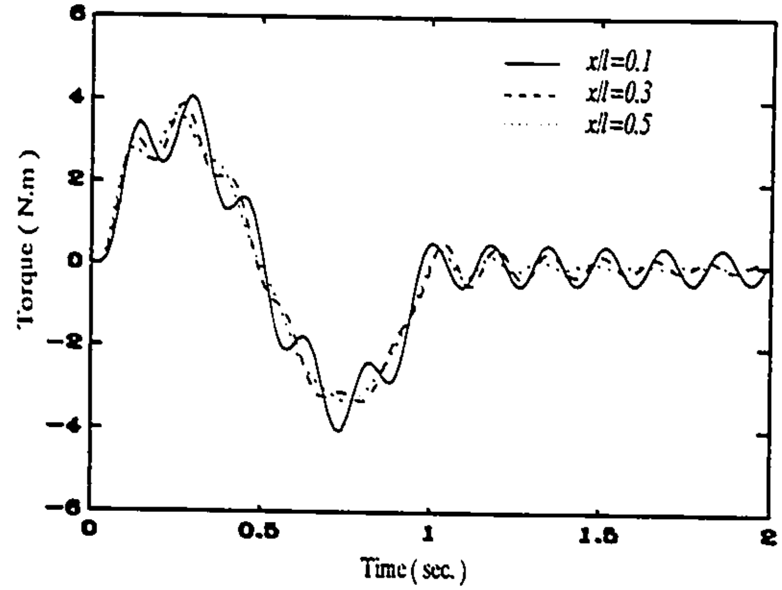


Figure 5.7: Input Torque for the tracking of a desired trajectory with a sinusoidal acceleration profile using the CTDD approach and a time delay ν=0.03 sec.

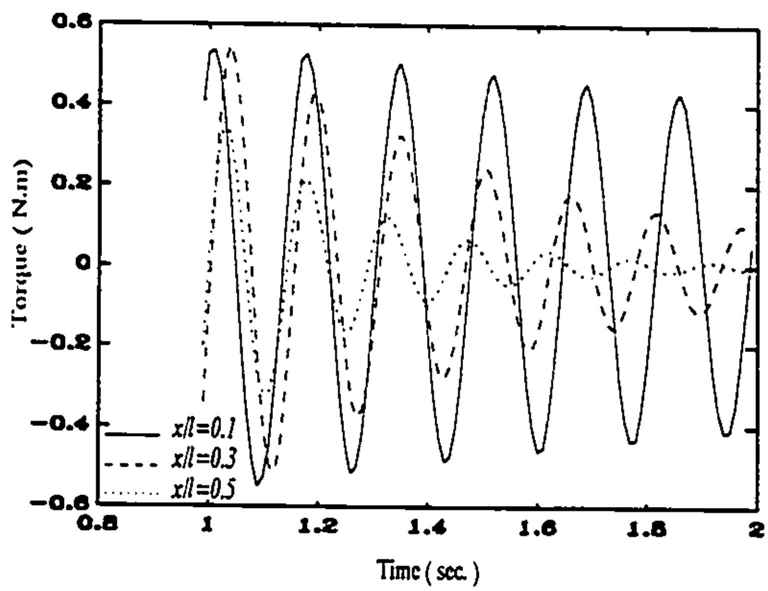


Figure 5.8: Zoom out of the last second in Figure 5.7.

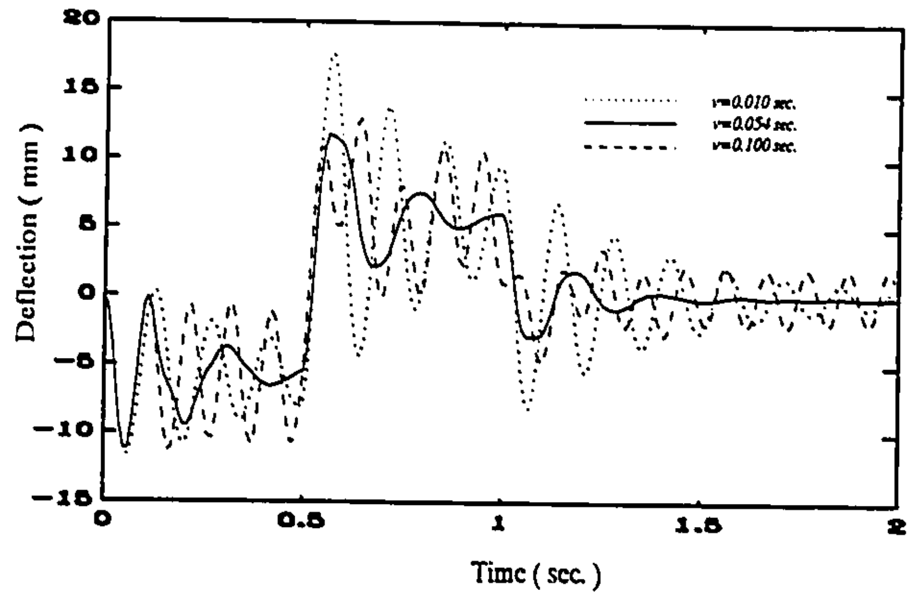


Figure 5.9: Tip deflection for the tracking of a desired trajectory with a bang-bang acceleration profile using the CTDD approach and $x/l = 0.4$.

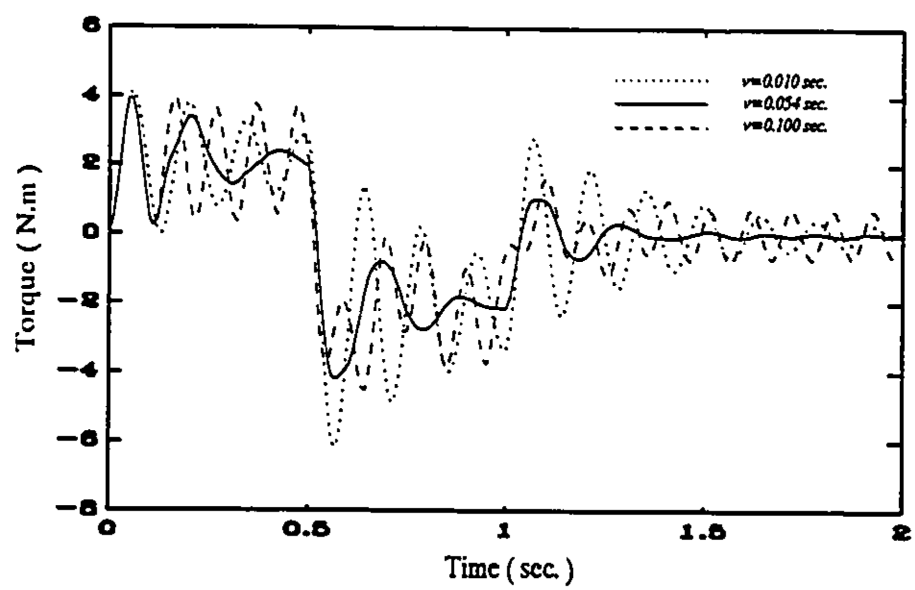


Figure 5.10: Input torque for the tracking of a desired trajectory with a bang-bang acceleration profile using the CTDD approach and $x/l = 0.4$.

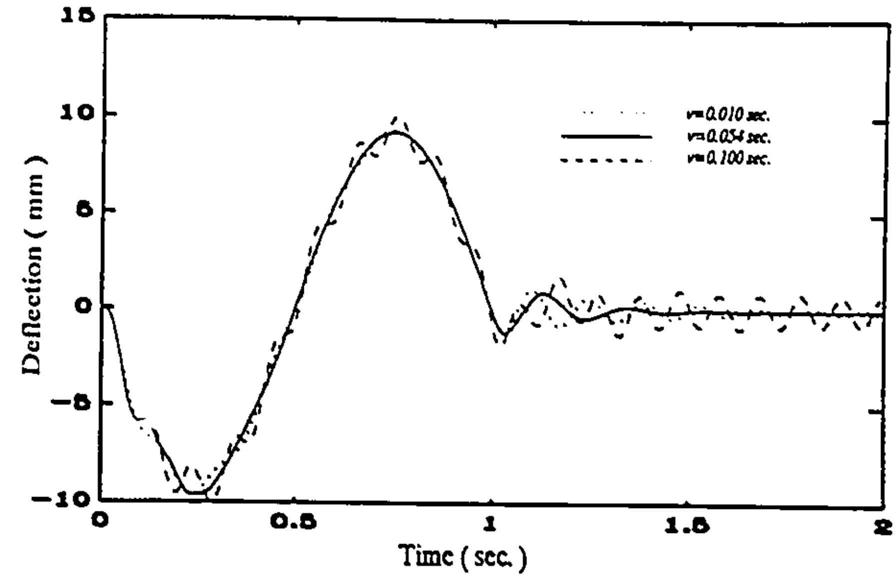


Figure 5.11: Tip deflection for the tracking of a desired trajectory with a sinusoidal acceleration profile using the CTDD approach and $x/l = 0.4$.

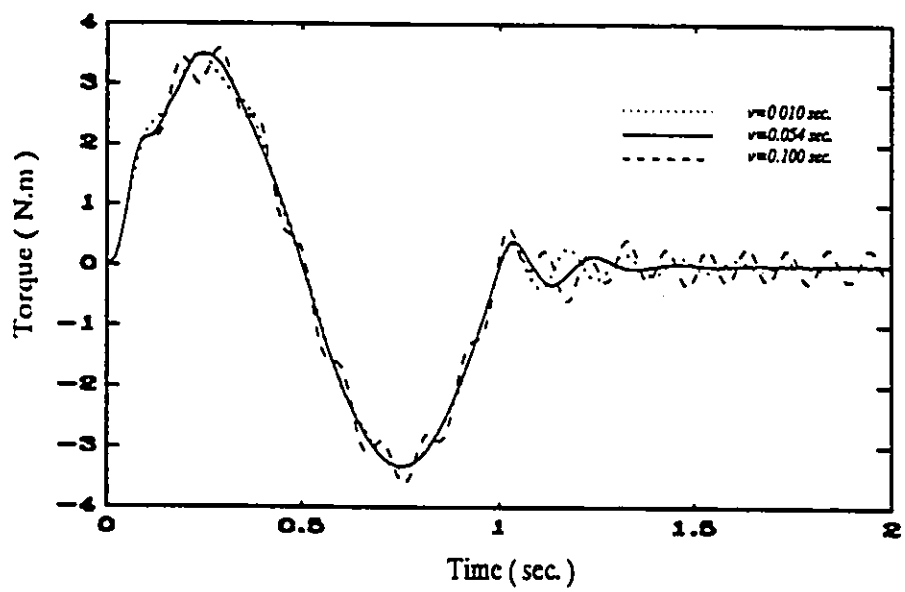


Figure 5.12: Input torque for the tracking of a desired trajectory with a sinusoidal acceleration profile using the CTDD approach and $x/l=0.4$.

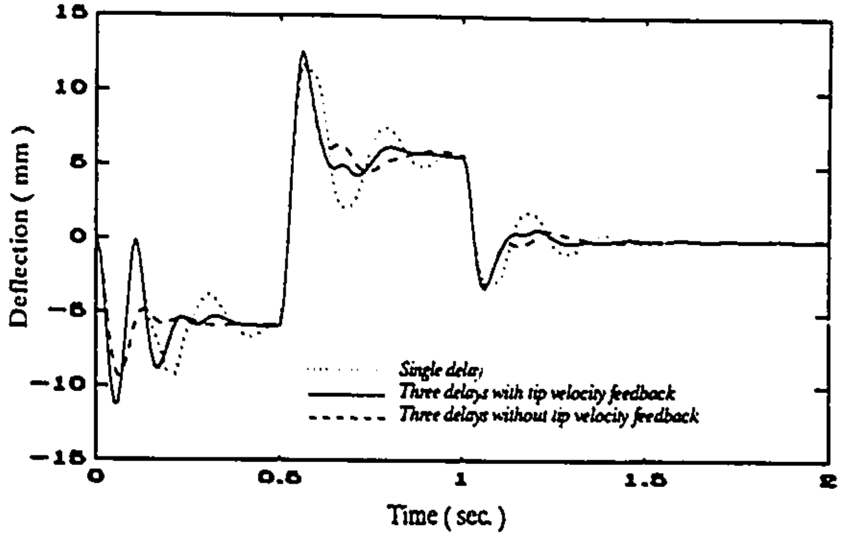


Figure 5.13: Tip deflection for the tracking of a desired trajectory with a Bang-Bang acceleration profile using the CTDD approach with single (v=0.054) and the sum of three delays (v_1=0.050 ,v_2=0.054 and v_3= 0.060). *x/l=0.4.*

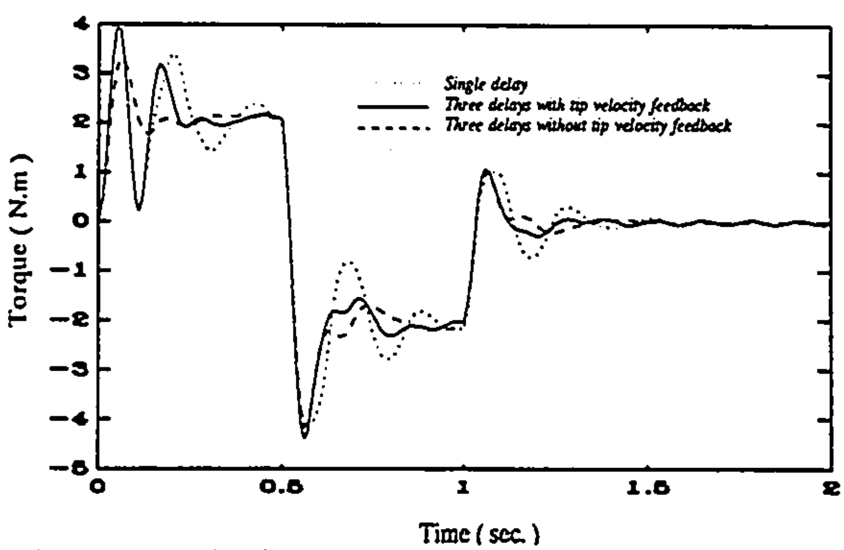


Figure 5.14: Input torque for the tracking of a desired trajectory with bang-bang acceleration using the CTDD approach with a single delay (v=0.054) and the sum of delays (v_1=0.050, v_2=0.054 and v_3=0.060). *x/l=0.4.*

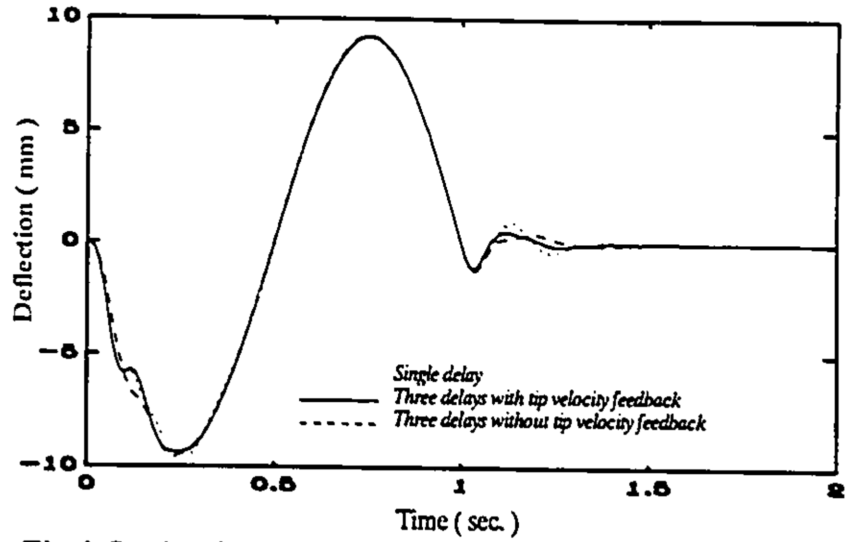


Figure 5.15: Tip deflection for the tracking of a desired trajectory with a sinusoidal acceleration profile using the CTDD approach with single delay $v=$ 0.054 and the sum of three delays $v_1=0.050$, $v_2=0.054$, $v_3=0.060$. *x/l=0.4.*

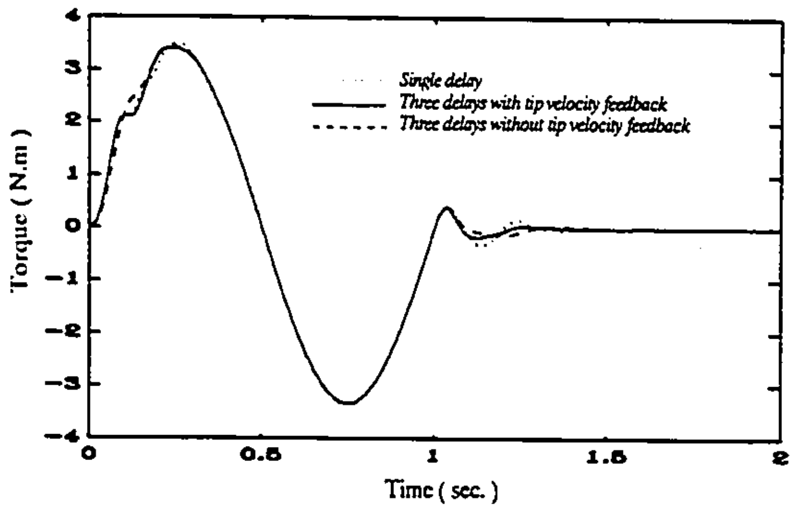


Figure 5.16: Input torque for the tracking of a desired trajectory with a sinusoidal acceleration profile using the CTDD approach with single delay $v=0.054$ and the sum of delays $v_1=0.050$, $v_2=0.054$, and $v_3=0.060$. *x/l=0.4.*

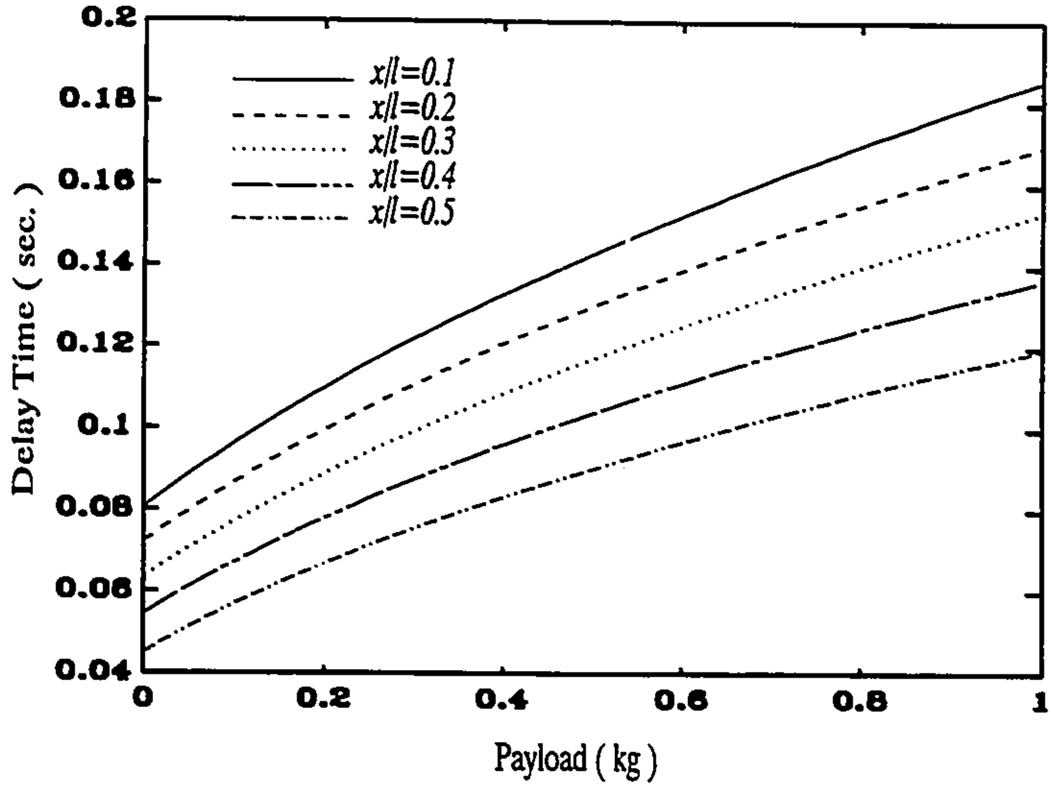


Figure 5.17: Variation of the required delay time with the payload.

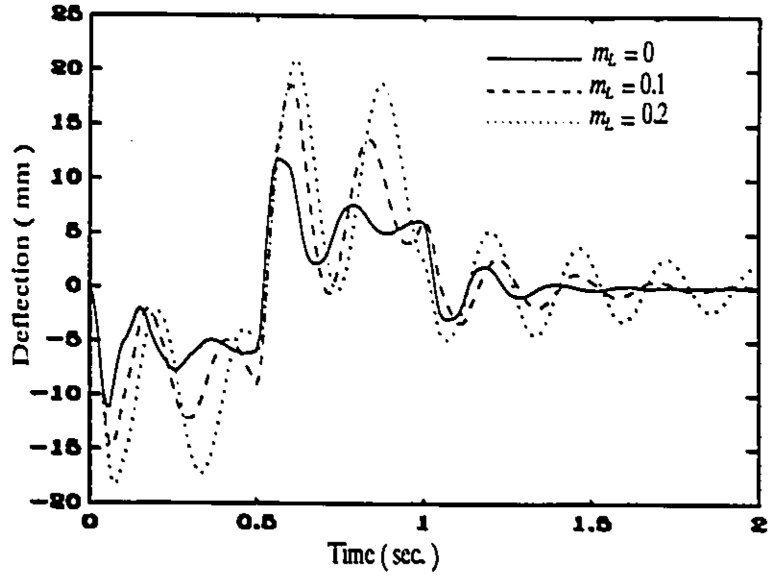


Figure 5.18: Tip deflection for x/l=0.4 and v=0.054 sec with m_L=0, 0.1 *kg* and 0.2 *kg*. Desired trajectory with bang-bang acceleration profile.

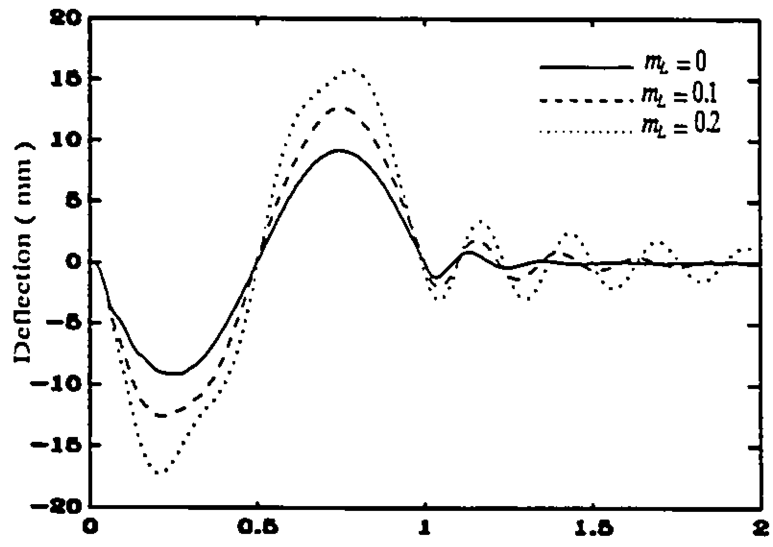


Figure 5.19: Tip deflection for x/l=0.4 and v=0.054 sec with m_L=0, 0.1 *kg* and 0.2 *kg*. Desired trajectory with sinusoidal acceleration profile.

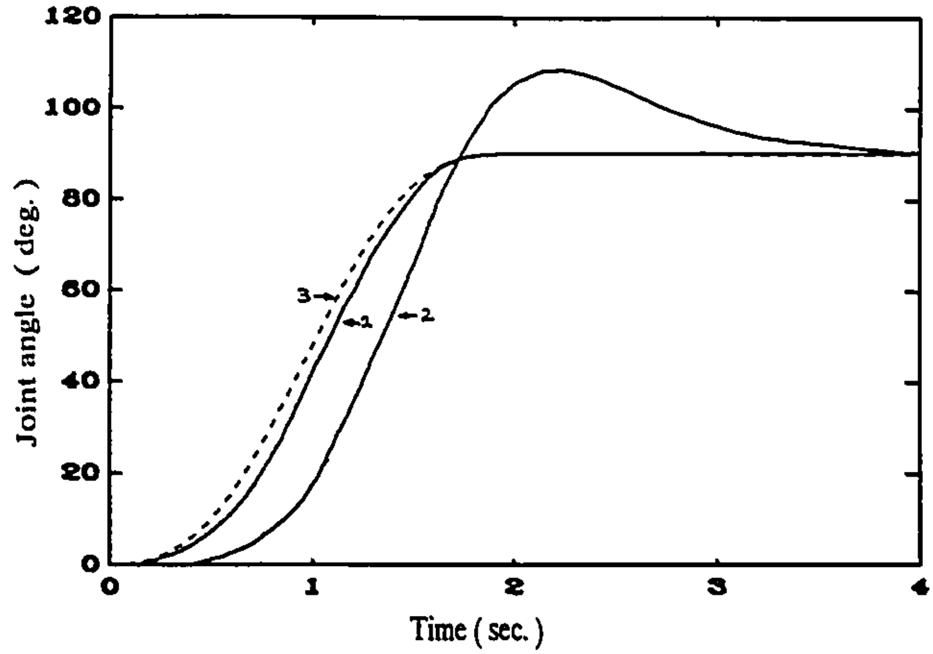


Figure 5.20: Joint angle: (1) Composite Control, (2) State Regulator, (3) CTDD.

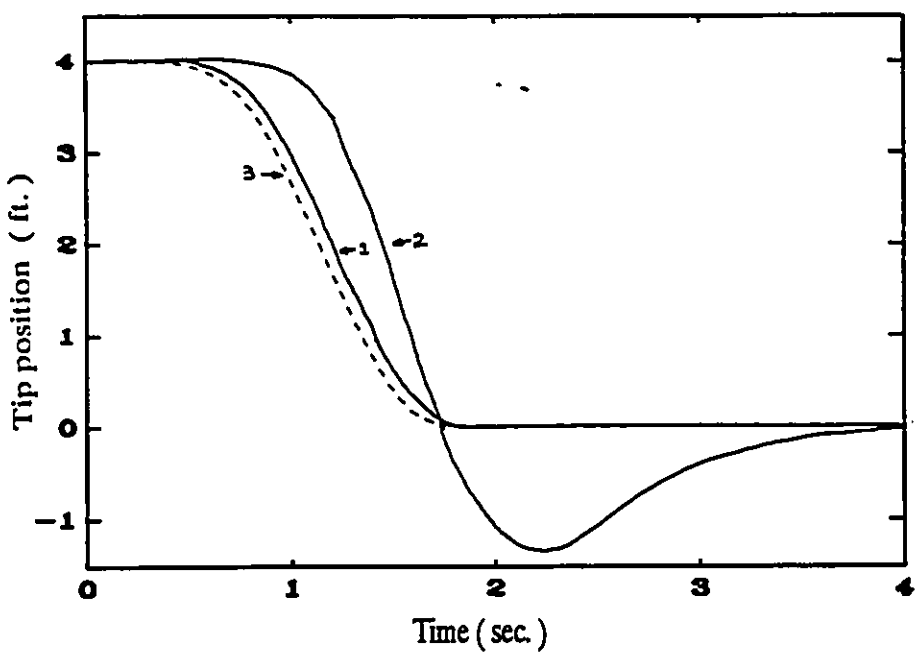


Figure 5.21: Tip position: (1) Composite Control, (2) State Regulator, (3) CTDD.

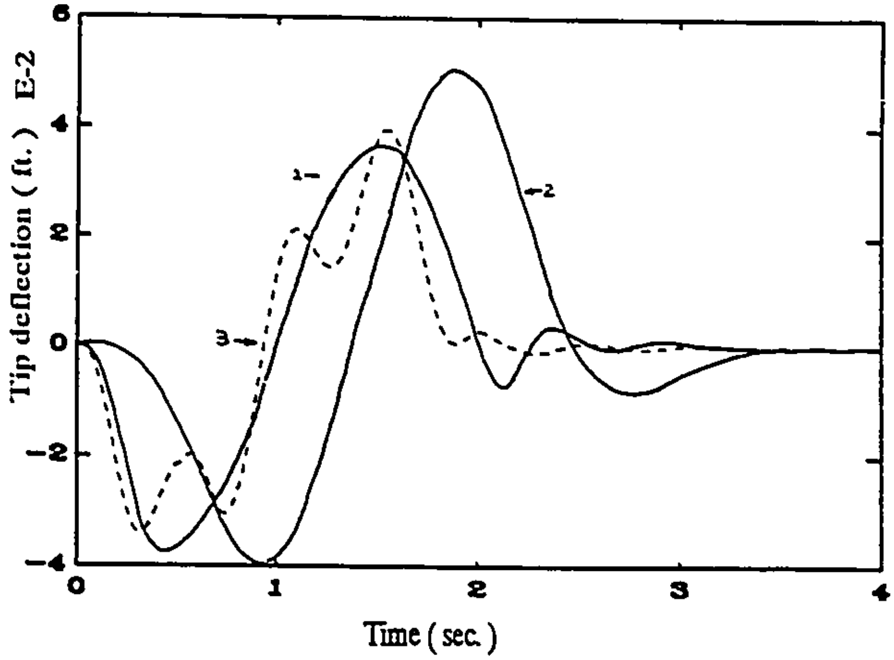


Figure 5.22: Tip deflection: (1) Composite Control, (2) State Regulator, (3)CTDD.

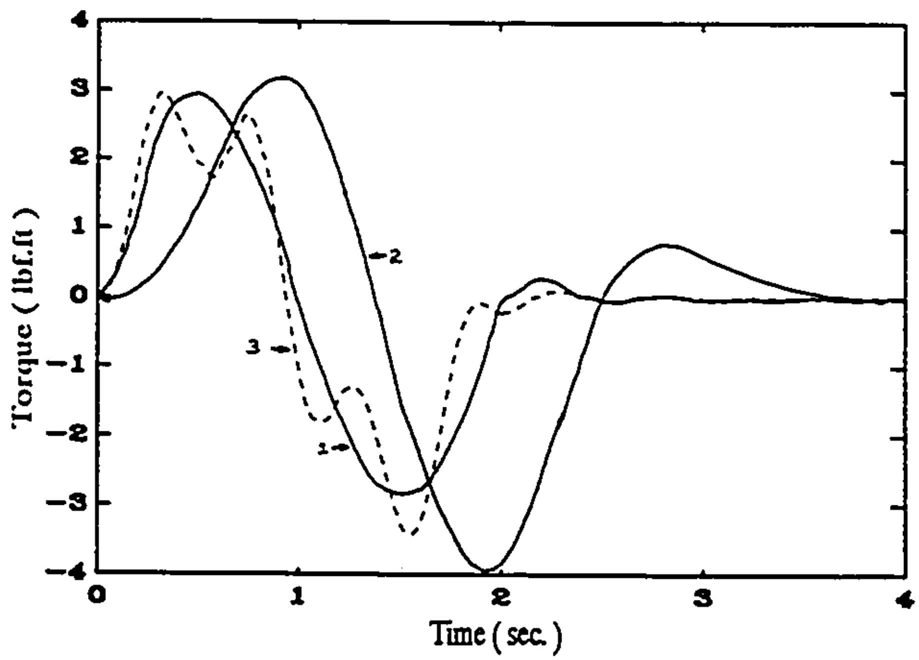


Figure 5.23: Control torque: (1) Composite Control, (2) State Regulator, (3) CTDD.

Chapter 6

Dynamics of a Two Link Flexible Manipulator

6.1 Introduction

In this chapter, the concept and formulations leading to an analytical model of a two-link flexible manipulator are discussed. The developed model is used to simulate the motion of the manipulator and to design controllers that are used to control the end point of the manipulator with minimum vibrations. These controllers will be presented in a subsequent chapter.

The two-link flexible manipulator can be viewed as a chain of two flexible links where the first link is attached to a fixed base and the second link is attached to a moving base. The moving base is represented by the end point of the first link. The GIM of a flexible link with fixed base is derived in Chapter 3. This chapter begins with the derivation of the GIM of a flexible link with moving base followed by a derivation of the total kinetic energy of the two-link flexible manipulator shown in Figure 6.2. Following the derivation of the kinetic energy, the potential energy of the manipulator is then derived. This potential energy is only due to the elastic property of the links. The manipulator is physically constrained to move in a plane normal to the gravitational field, hence the gravitational potential energy is zero. In order to investigate the effects of the flexibility of one link on the other, three cases were simulated: a) Rigid-Flexible case where the first link is rigid and the second link is flexible. b) Flexible-Rigid case where the first link is flexible and the second link is rigid. c) Flexible-Flexible case where both links are flexible.

6.2 GIM of a flexible link with a moving base

With reference to Figure 6.1, the position of o_2 with respect to the frame $o_1x_1y_1$ is given by:

$$r_{o_2/o_1} = \begin{bmatrix} l_1 & \xi_{e_1} \end{bmatrix}^T \qquad 6.1$$

The angle between $o_1x_1y_1$ and $o_2x_2y_2$ is $\theta_2+\delta_{e1}$.

The absolute position of a point on link 2 is given by:

$$r = R_0^1 \left[\begin{bmatrix} l_1 \\ \xi_{e_1} \end{bmatrix} + R_1^2 \begin{bmatrix} x_2 \\ u_2(x_2,t) \end{bmatrix} \right] \qquad 6.2$$

R_1^2 is the transformation matrix from $o_2x_2y_2$ to $o_1x_1y_1$ and given by:

$$R_1^2 = \begin{pmatrix} \cos(\theta_2+\delta_{e_1}) & -\sin(\theta_2+\delta_{e_1}) \\ \sin(\theta_2+\delta_{e_1}) & \cos(\theta_2+\delta_{e_1}) \end{pmatrix} \qquad 6.3$$

$u_2(x_2,t)$ is the deflection of a point along the link with respect to the link's coordinate frame. From Equation 3.3, $u_2(x_2,t)$ can be expressed in terms of the shape functions $\Phi(x)$ and the coordinates of the end point of the link as:

$$u_2(x_2,t) = \Phi^T(x_2)\eta_2 \qquad \eta_2 = \begin{bmatrix} \xi_{e_2} & \delta_{e_2} \end{bmatrix}^T \tag{6.4}$$

The kinetic energy of the link can be written as:

$$T = \frac{m}{2}\int_0^{l_2} \left[\frac{dr}{dt}\right]^T \left[\frac{dr}{dt}\right] dx \tag{6.5}$$

The position vector r is a function of the variables θ_1, θ_2, ξ_{e1}, ξ_{e2}, δ_{e1} and δ_{e2}, hence its time derivative can be expressed as:

$$\frac{\partial r}{\partial t} = \begin{bmatrix} \frac{\partial r}{\partial \theta_1} & \frac{\partial r}{\partial \theta_2} & \frac{\partial r}{\partial \xi_{e_1}} & \frac{\partial r}{\partial \delta_{e_1}} & \frac{\partial r}{\partial \xi_{e_2}} & \frac{\partial r}{\partial \delta_{e_2}} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\xi}_{e_1} \\ \dot{\delta}_{e_1} \\ \dot{\xi}_{e_2} \\ \dot{\delta}_{e_2} \end{bmatrix} \tag{6.5}$$

From Equations 6.4 and 6.5, the kinetic energy can be expressed in matrix form as:

$$T = \frac{1}{2}\dot{q}^T M \dot{q} \qquad 6.6$$

The mass matrix M is the GIM of the link and q=[θ_1 θ_2 ξ_{e1} δ_{e1} ξ_{e2} δ_{e2}]^T is the generalized coordinates vector describing the motion of a point on the link. An expression of the matrix M is given by:

$$M = m\int_0^{l_2} \begin{bmatrix} \frac{\partial r^T}{\partial \theta_1} \\ \frac{\partial r^T}{\partial \theta_2} \\ \frac{\partial r^T}{\partial \xi_{e_1}} \\ \frac{\partial r^T}{\partial \delta_{e_1}} \\ \frac{\partial r^T}{\partial \xi_{e_2}} \\ \frac{\partial r^T}{\partial \delta_{e_2}} \end{bmatrix} \begin{bmatrix} \frac{\partial r}{\partial \theta_1} & \frac{\partial r}{\partial \theta_2} & \frac{\partial r}{\partial \xi_{e_1}} & \frac{\partial r}{\partial \delta_{e_1}} & \frac{\partial r}{\partial \xi_{e_2}} & \frac{\partial r}{\partial \delta_{e_2}} \end{bmatrix} dx \qquad 6.7$$

The elements of the matrix M are derived in Appendix B.

6.3 Dynamic Model

With reference to Figure 6.2, both joints are assumed to be rotary and rigid. the motion of the arm is confined within the horizontal plane and so is the deformation of each link. The coordinates system OXY represents the inertial coordinates system. $o_i x_i y_i$

is the local coordinates system assigned to link i in its undeformed position. The mass and inertia of joint 2 including the mass and inertia of the actuator driving the second link are assumed to be lumped at the tip of the first link. A payload of mass m_L and inertia I_L is located at the tip of the second link.

6.3.1 Kinetic Energy

The total kinetic energy of the two link flexible manipulator shown in Figure 6.2 is obtained by adding the kinetic energies associated with each of the two links, as well as those associated with the mass and the inertia at the tip of each link.

$$T = T_1 + T_2 + T_a + T_L \qquad 6.8$$

T_1 and T_2 are the kinetic energies of first and second link respectively. T_a is the kinetic energy of the mass and inertia at the tip of the first link and T_L is the kinetic energy of the payload. The generalized coordinates describing the motion of the two link flexible manipulator are given by $q = [\theta_1 \;\; \theta_2 \;\; \xi_{e_1} \;\; \delta_{e_1} \;\; \xi_{e_2} \;\; \delta_{e_2}]^T$. T_1 and T_2 are expressed in terms of the GIMs of the first and second link M_1 and M_2, respectively as:

$$T_1 = \frac{1}{2}\dot{q}^T M_1 \dot{q} \qquad 6.9$$

$$T_2 = \frac{1}{2}\dot{q}^T M_2 \dot{q} \qquad 6.10$$

M_1 is the augmented matrix of the 3 by 3 matrix in Equation 3.10 and is given by:

$$M_1 = m_1 \int_0^{l_1} \begin{bmatrix} \frac{\partial r^T}{\partial \theta_1} \\ 0 \\ \frac{\partial r^T}{\partial \xi_{e_1}} \\ \frac{\partial r^T}{\partial \delta_{e_1}} \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} \frac{\partial r}{\partial \theta_1} & 0 & \frac{\partial r}{\partial \xi_{e_1}} & \frac{\partial r}{\partial \delta_{e_1}} & 0 & 0 \end{bmatrix} dx_1 \qquad 6.11$$

and M_2 can be obtained from Equation 6.7.

T_a is expressed in terms of the generalized coordinates vector q and the matrix M_a as:

$$T_a = \frac{1}{2}\dot{q}^T M_a \dot{q} \qquad 6.12$$

where M_a is a 6 by 6 matrix representing the GIM of the mass and inertia at the tip of the first link. M_a is obtained by augmenting the 3 by 3 matrix given by Equation 3.15 and is given by:

$$M_a = m_a \left[\frac{\partial r_{e_1}^T}{\partial q}\right] \left[\frac{\partial r_{e_1}}{\partial q}\right]^T + I_a \left[\frac{\partial \omega_{e_1}}{\partial \dot{q}}\right] \left[\frac{\partial \omega_{e_1}}{\partial \dot{q}}\right]^T \qquad 6.13$$

where

$$\left[\frac{\partial r_{e_1}}{\partial q}\right] = \begin{bmatrix} \frac{\partial r_{e_1}}{\partial \theta_1} & 0 & \frac{\partial r_{e_1}}{\partial \xi_{e_1}} & 0 & 0 & 0 \end{bmatrix}^T$$

ω_{e1} is the absolute angular velocity of the end point of the first link. m_a and I_a are the mass and moment of inertia of the actuator driving the second link. Similarly, T_L can be expressed in terms of $\boldsymbol{q}$ and a matrix $\boldsymbol{M}_L$ representing the GIM of the payload as:

$$T_L = \frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{M}_L \dot{\boldsymbol{q}} \tag{6.14}$$

Where the matrix $\boldsymbol{M}_L$ is given by:

$$\boldsymbol{M}_L = m_L \left[\frac{\partial \boldsymbol{r}_{e_2}^T}{\partial \boldsymbol{q}}\right]\left[\frac{\partial \boldsymbol{r}_{e_2}}{\partial \boldsymbol{q}}\right]^T + I_L \left[\frac{\partial \omega_L}{\partial \dot{\boldsymbol{q}}}\right]\left[\frac{\partial \omega_L}{\partial \dot{\boldsymbol{q}}}\right]^T \tag{6.15}$$

where

$$\left[\frac{\partial \boldsymbol{r}_{e_2}}{\partial \boldsymbol{q}}\right] = \left[\frac{\partial \boldsymbol{r}_{e_2}}{\partial \theta_1} \quad \frac{\partial \boldsymbol{r}_{e_2}}{\partial \theta_2} \quad \frac{\partial \boldsymbol{r}_{e_2}}{\partial \xi_{e_1}} \quad \frac{\partial \boldsymbol{r}_{e_2}}{\partial \delta_{e_1}} \quad \frac{\partial \boldsymbol{r}_{e_2}}{\partial \xi_{e_2}} \quad \frac{\partial \boldsymbol{r}_{e_2}}{\partial \delta_{e_2}}\right]^T$$

and

$$\left[\frac{\partial \omega_L}{\partial \dot{\boldsymbol{q}}}\right] = \left[\frac{\partial \omega_L}{\partial \dot{\theta}_1} \quad \frac{\partial \omega_L}{\partial \dot{\theta}_2} \quad \frac{\partial \omega_L}{\partial \dot{\xi}_{e_1}} \quad \frac{\partial \omega_L}{\partial \dot{\delta}_{e_1}} \quad \frac{\partial \omega_L}{\partial \dot{\xi}_{e_2}} \quad \frac{\partial \omega_L}{\partial \dot{\delta}_{e_2}}\right]^T$$

The vector $\boldsymbol{r}_{e_2}$ represents the position vector of the tip of the second link and is given by:

$$r_{e_2} = R_0^1\left[\begin{bmatrix} l_1 \\ \xi_{e_1} \end{bmatrix} + R_1^2\begin{bmatrix} l_2 \\ \xi_{e_2} \end{bmatrix}\right] \qquad 6.16$$

The second term in Equation 6.15 represents the inertia matrix associated with the rotational kinetic energy of the payload which absolute angular velocity is given by:

$$\omega_L = \dot{\theta}_1 + \dot{\theta}_2 + \dot{\delta}_{e_1} + \dot{\delta}_{e_2} \qquad 6.17$$

The generalized inertia matrix of the two link flexible manipulator can be obtained by assembling the matrices M_1, M_2, M_a and M_L as:

$$M = M_1 + M_2 + M_a + M_L \qquad 6.18$$

The matrix M can be partitioned into sub-matrices representing the rigid and flexible sub-systems as follows:

$$M = \begin{bmatrix} [M_r] & [M_{rf}]^T \\ [M_{rf}] & [M_f] \end{bmatrix} \qquad 6.19$$

M_r is a 2 by 2 symmetric matrix representing the mass of the rigid sub-system, M_f is a 4 by 4 symmetric matrix representing the mass of the flexible sub-system, and M_{rf} is a 4 by 2 matrix representing the coupling between the rigid and the flexible sub-systems. The vector q can be divided into two parts, a vector $\Theta=[\theta_1 \quad \theta_2]^T$ representing rigid motion coordinates, and a vector $\eta=[\,\xi_{e_1}\ \ \delta_{e_1}\ \ \xi_{e_2}\ \ \delta_{e_2}]^T$ representing the flexible motion coordinates. The kinetic energy can then be written as:

$$T=\frac{1}{2}[\Theta^T \quad \dot{\eta}^T]M\begin{bmatrix}\Theta\\ \dot{\eta}\end{bmatrix} \tag{6.20}$$

Substituting the partitioned matrix in the above equation and simplifying gives

$$T=\frac{1}{2}\Theta^T M_r\Theta+\frac{1}{2}\dot{\eta}^T M_f\dot{\eta}+\Theta^T M_{rf}^T\dot{\eta} \tag{6.21}$$

6.3.2 Potential Energy

In the absence of gravity, the total potential energy of the two link flexible manipulator is given by the sum of the elastic energies of each link.

$$U=U_1+U_2 \tag{6.22}$$

Where U_I is expressed in terms of the variables representing the elastic motion of the tip of the first link and its stiffness matrix as:

$$U_1=\frac{1}{2}\eta_{e_1}^T \mathrm{K}_1\eta_{e_1} \tag{6.23}$$

Where K_1 is the stiffness matrix of the first link and is given by:

$$K_1 = EI_1 \int_0^{l_1} \left[\frac{d^2\Phi_1}{dx_1^2}\right]\left[\frac{d^2\Phi_1}{dx_1^2}\right]^T dx_1 \qquad 6.24$$

and $\eta_{e_1}=[\xi_{e_1} \quad \delta_{e_1}]^T$. $\Phi_1=[\phi_{11} \quad \phi_{12}]^T$ is a vector representing the two shape functions that describe the deflection of the first link. Similarly, the elastic potential energy of the second link can be expressed in terms of the variables representing the motion of the tip of the second link and its stiffness as:

$$U_2 = \frac{1}{2}\eta_{e_2}^T K_2 \eta_{e_2} \qquad 6.25$$

Where K_2 is the stiffness matrix of the second link and given by:

$$K_2 = EI_2 \int_0^{l_2} \left[\frac{d^2\Phi_2}{dx_2^2}\right]\left[\frac{d^2\Phi_2}{dx_2^2}\right]^T dx_2 \qquad 6.26$$

and $\eta_{e_2}=[\xi_{e_2} \quad \delta_{e_2}]^T$. $\Phi_2=[\phi_{21} \quad \phi_{22}]^T$ is a vector representing the two shape functions that describe the deflection of the second link. The potential energy $\boldsymbol{U}$ can be expressed in terms of the generalized coordinates $\boldsymbol{q}$ as

$$\boldsymbol{U} = \frac{1}{2}\boldsymbol{q}^T K \boldsymbol{q} \qquad 6.27$$

with K being the generalized stiffness matrix

$$K = \begin{bmatrix} \mathbf{0}_{2x2} & \mathbf{0}_{2x2} & \mathbf{0}_{2x2} \\ \mathbf{0}_{2x2} & K_1 & \mathbf{0}_{2x2} \\ \mathbf{0}_{2x2} & \mathbf{0}_{2x2} & K_2 \end{bmatrix} \qquad 6.28$$

6.3.3 Equations of motion

Similar to the case of the one link manipulator, the equations of motion of the two-link flexible manipulator are derived using the Lagrangian formulation. The equation describing the motion of the rigid sub-system is given by:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\Theta}}\right) - \frac{\partial L}{\partial \Theta} = T \qquad 6.29$$

$T=[\tau_1 \quad \tau_2]^T$ is a vector representing the actuators torques at joint 1 and 2 and $L=T+U$ is the Lagrangian of the system. The equations describing the motion of the end points of link 1 and link 2, respectively are:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\eta}_{e_1}}\right) - \frac{\partial L}{\partial \eta_{e_1}} = 0 \qquad 6.30$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\eta}_{e_2}}\right) - \frac{\partial L}{\partial \eta_{e_2}} = 0 \qquad 6.31$$

Using the expressions of the kinetic and potential energies given by Equations 6.21 and 6.27, respectively, the equations representing the rigid sub-system and the flexible sub-system are:

$$M_r\ddot{\Theta} + M_{rf}^T\ddot{\eta} + H_r = T \qquad 6.32$$

$$M_{rf}\ddot{\Theta} + M_f\ddot{\eta} + H_f = 0 \qquad 6.33$$

H_r is a 2 by 1 vector representing the non linear terms associated with the rigid sub-system and H_f is a 4 by 1 vector representing the non linear terms associated with the flexible sub-system.

6.3.4 Simulation Results

Before investigating its response due to control inputs (joint torques), the two-link flexible manipulator has been subjected to some initial deformations in order to investigate its response due to non zero initial conditions and to understand the interactions between the two links. The physical characteristics of the two-link flexible manipulator used in the simulations are given in Table 6.1.

	Link 1	Link 2	Joint 2
mass	m_1=1.0 kg/m	m_2=1.0 kg/m	m_a=1.0 kg
Length	l_1=1.0 m	l_2=1.0 m	-
Stiffness	EI_1=200 N.m^2	EI_2=200 N.m^2	-
Mass moment of Inertia	-	-	0.1 kg.m^2

Table 6.1: Parameters of the two link flexible manipulator.

Three cases are going to be addressed. The first case is when the first link is assumed to be rigid while the second link is flexible and will be referred to as the Rigid-Flexible case. The second case is when the first link is assumed to be flexible while the second link is rigid and will be referred to as the Flexible-Rigid case. The third case is when both links are assumed to be flexible and will be referred to as the Flexible-Flexible case.

6.3.4.1 Rigid-Flexible Two-Link Manipulator

Figure 6.3 shows a Rigid-Flexible two-link manipulator. The flexible link of the manipulator is subjected to an initial deformation ξ_{e2}=0.01 m. Figure 6.4 shows the response of the flexible link when it is subjected to the initial deformation ξ_{e2}=0.01 m. When both joints are assumed to be clamped, the response of the tip is similar to that of the one link flexible manipulator described in Chapter 3 and shown in Figure 3.5. The effect of the flexibility of the second link on the motion of the first joint can be obtained by using pinned joints instead of clamped ones. The motion of the joint angles caused by the oscillations at the tip of the flexible link are shown in Figure 6.5. It can be seen from this figure that the flexibility of the second link has very small effect on the joint motion of the rigid first link. The induced motion at the joint of the first link is an oscillatory motion with a maximum amplitude of nearly 0.05 deg. The maximum amplitude of the motion of the second joint is 1.2 deg.

6.3.4.2 Flexible-Rigid Two-Link Manipulator

This case is illustrated in Figure 6.6. With reference to the figure, the first link, is subjected to an initial deformation ξ_{x1}=0.01 m. This case is rarely studied in the literature because of the changing boundary conditions at the end of the flexible link as θ_2 changes. A change in θ_2 results in a change in the natural frequencies of the flexible link. When both joints are clamped, the response of the flexible link is equivalent to that of a single link with a load at its tip. The deflection at the end of the flexible link when the arm is fully extended is shown in Figure 6.7. A significant change in the motion of the tip of the flexible link occurs when the second joint is clamped at θ_2=-90. Figure 6.8, shows the deflection at the tip of the first link. It can be seen from the figure that the oscillations of the flexible link are no longer symmetric. The asymmetry is caused by the offset of the center of mass of the second link from the axis of the first link.

6.3.4.3 Flexible-Flexible Two-Link Manipulator

This section treats the case when both links are flexible. Figure 6.9a shows a fully extended two-link flexible arm and Figure 6.9b shows a two-link flexible arm with θ_1=0 and θ_2=-90. In both cases, the links are subjected to initial deformations of $\xi_{x1}=\xi_{x2}$=0.01 m. Figures 6.10 and 6.11 show the deflections at the tip of the first link and the second link, respectively, when the arm is under the conditions shown in Figure 6.9a. The results shown in these two Figures are very similar to those shown in Figure 6.3 and 6.7. When the arm under the conditions shown in Figure 6.9b is released, the asymmetry of the deflection at the tip of the first link can be seen very clearly in Figure 6.12. Since the deflection is measured relative to the link itself, the asymmetry has little effect on the deflection of the second link as shown in Figure 6.13.

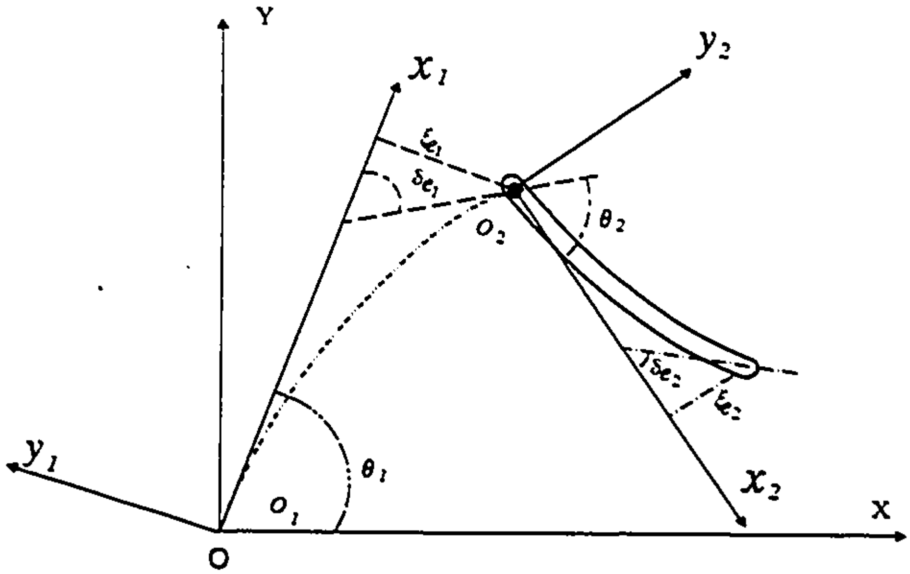


Figure 6.1: Coordinate system of a flexible link with a moving base

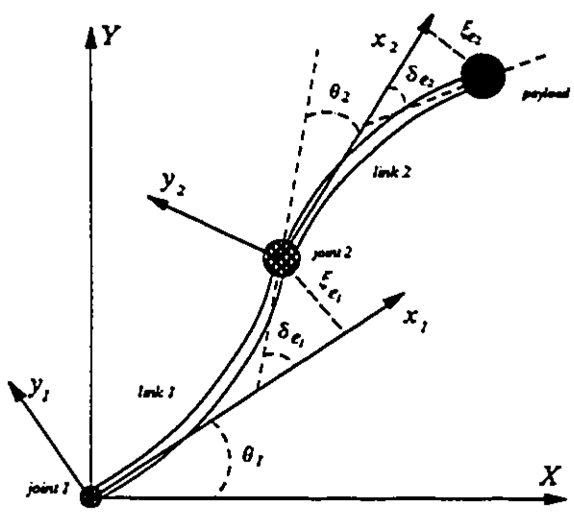


Figure 6.2: Two link flexible manipulator

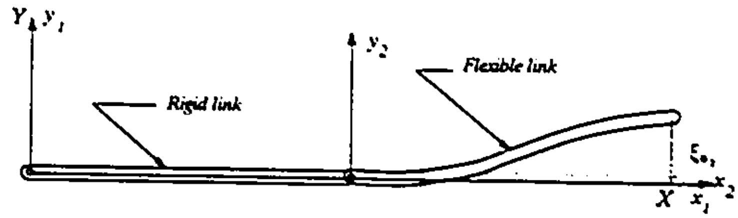


Figure 6.3: Rigid-Flexible Two-Link Manipulator.

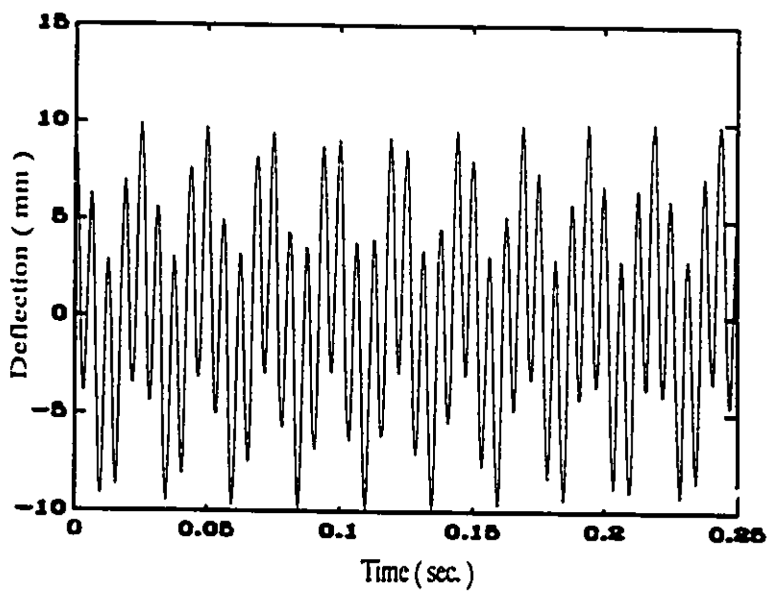


Figure 6.4: Deflection at the tip of the second link for the Rigid-Flexible case.

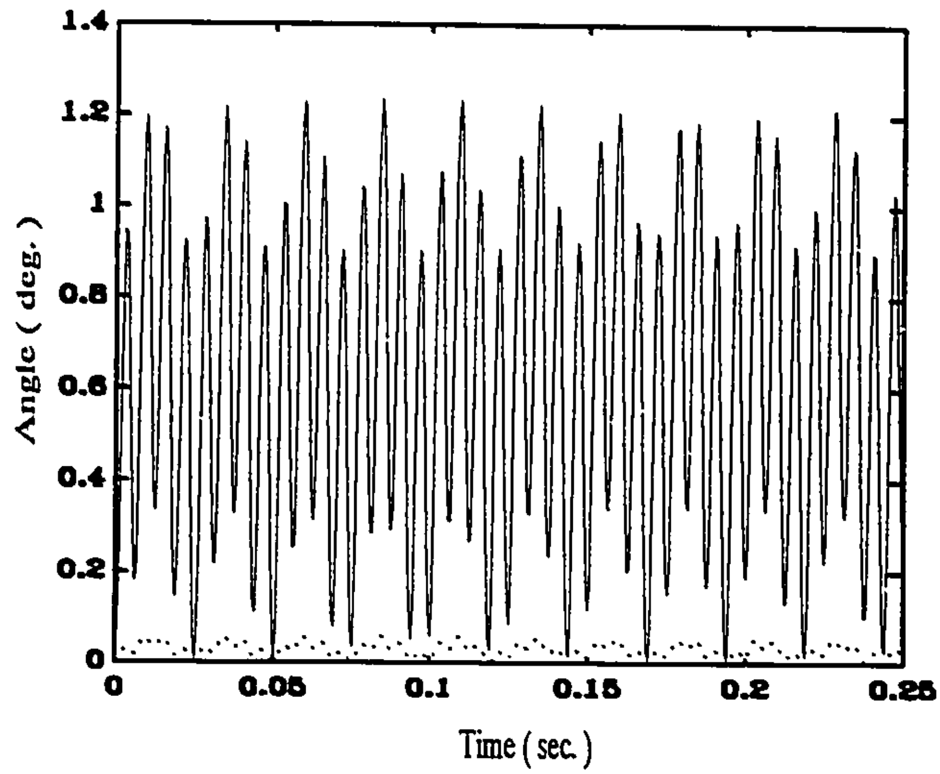
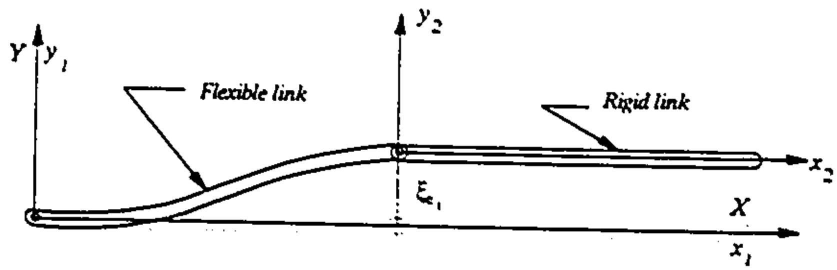
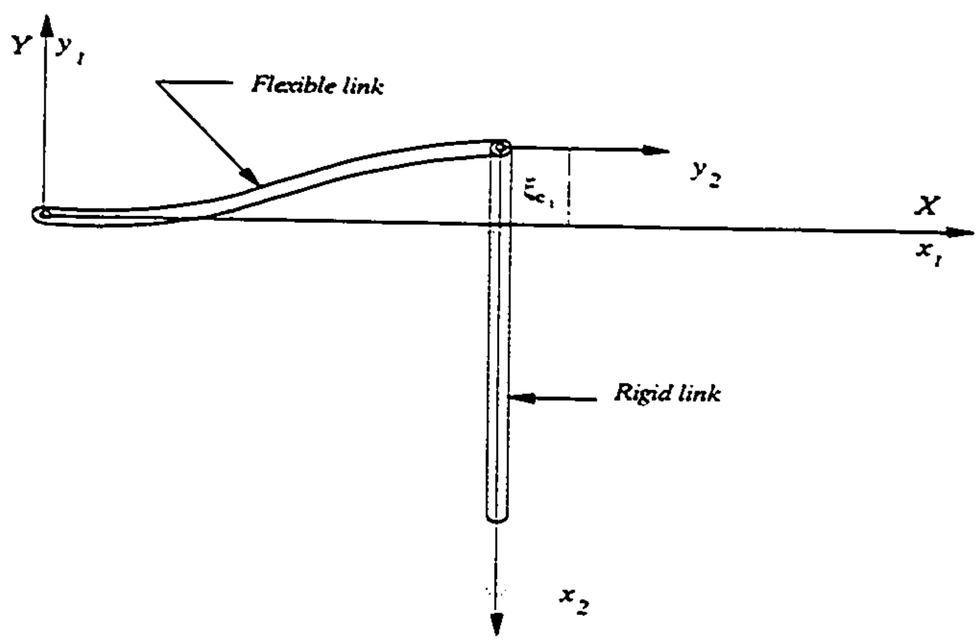


Figure 6.5: Motion of the joints induced by oscillations at the tip of the Rigid-Flexible Two-link Manipulator.
———— θ_2
············ θ_1.



a



b

Figure 6.6: Flexible-Rigid Two-Link Manipulator.
a: θ_2=0; b: θ_2=-90.

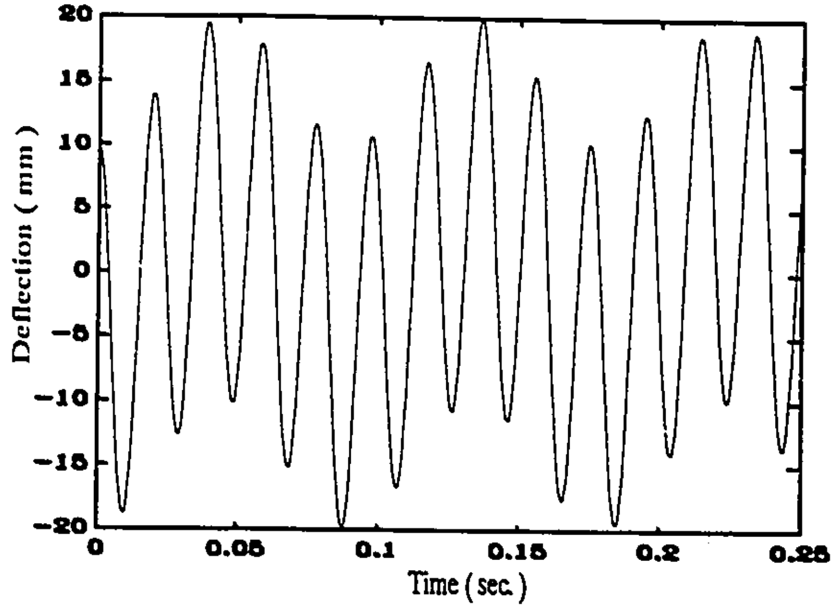


Figure 6.7: Deflection at the tip of the first link for the Flexible-Rigid case. θ_2=0.

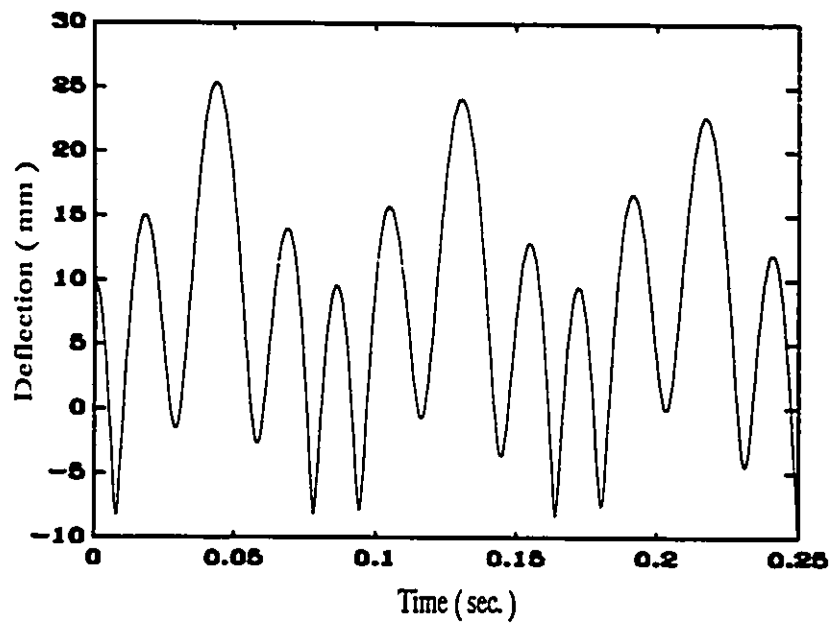
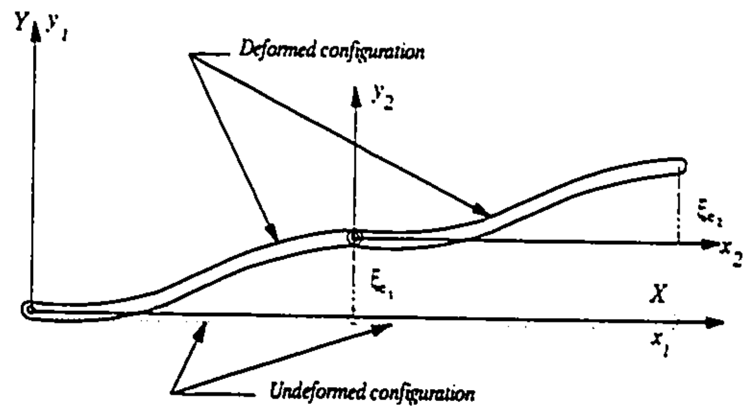
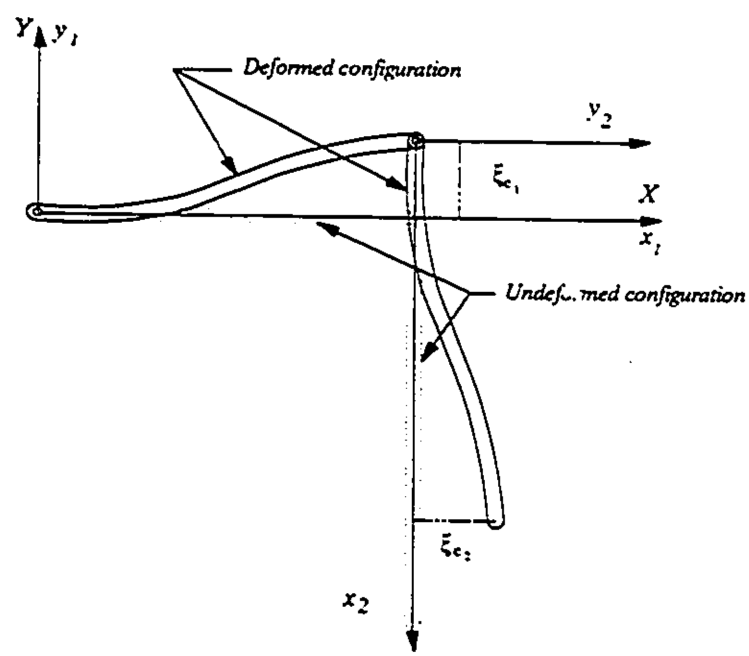


Figure 6.8: Deflection at the tip of the first link for the Flexible-Rigid case. θ_2=-90.



a



b

Figure 6.9: Flexible-Flexible Two-Link Manipulator.
a: θ_2=0; b: θ_2=-90.

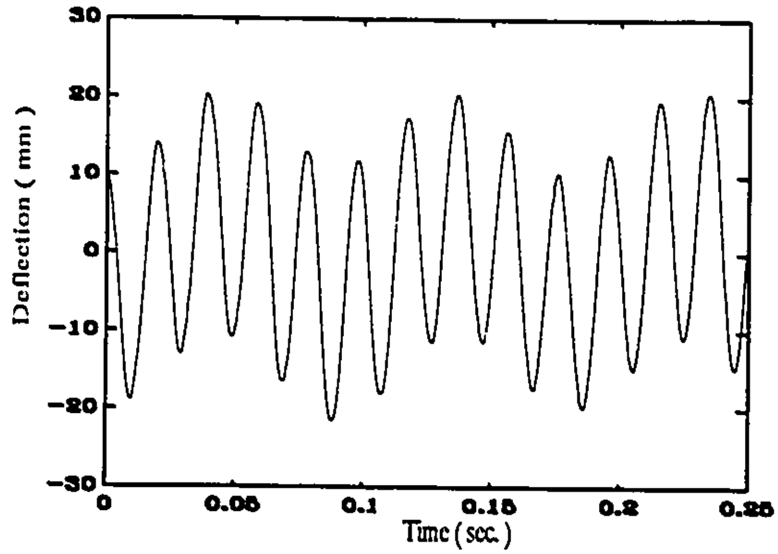


Figure 6.10: Deflection at the tip of the first link for the Two-Link Flexible Manipulator. θ_2=0.

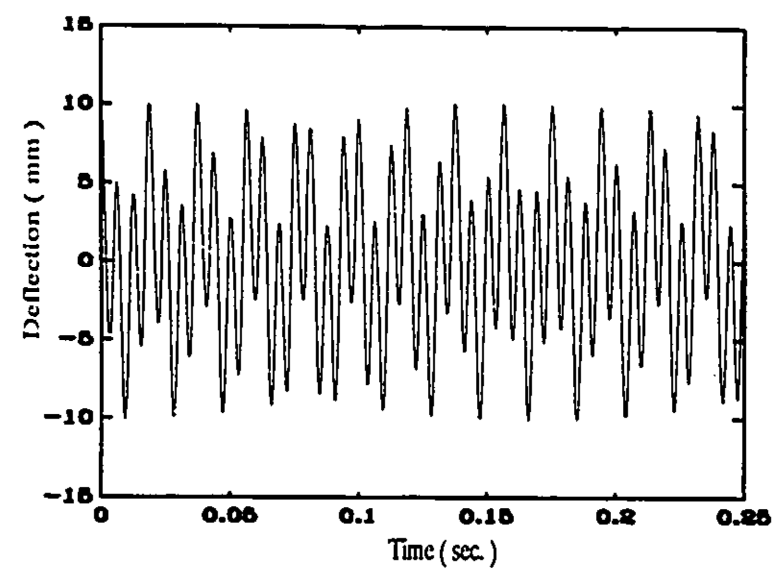


Figure 6.11: Deflection at the tip of the second link for the Two-Link Flexible Manipulator. θ_2=0.

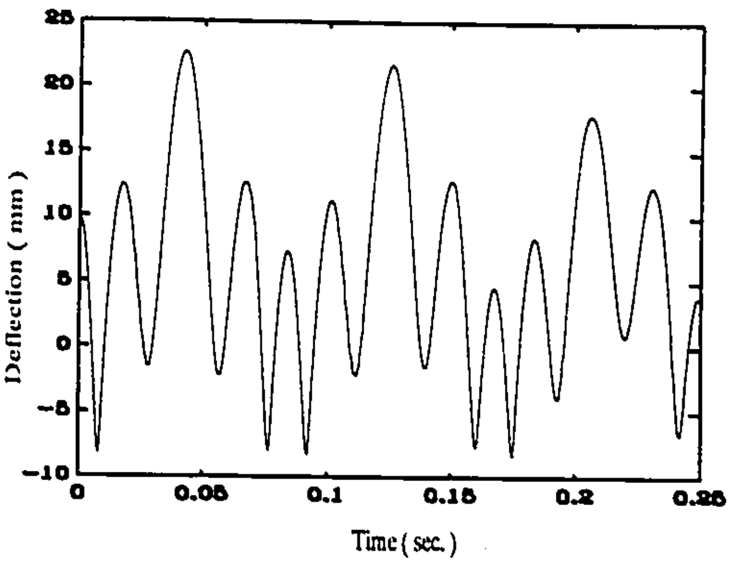


Figure 6.12: Deflection at the tip of the first link for the Two-Link Flexible Manipulator. θ_2=-90.

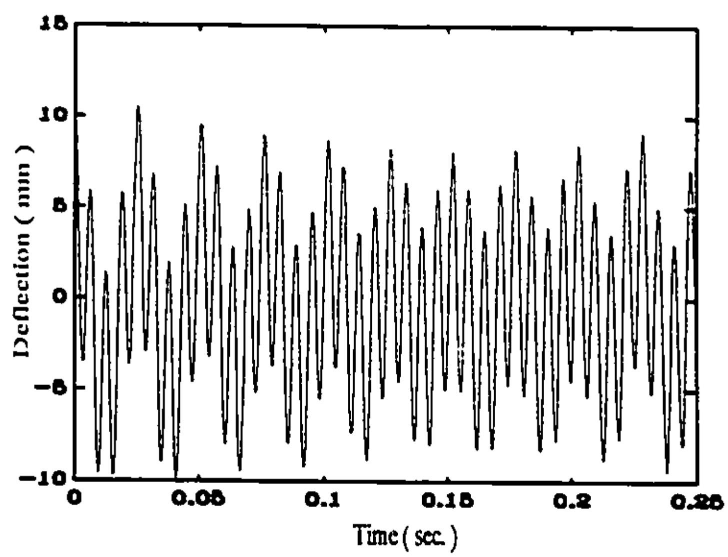


Figure 6.13: Deflection at the tip of the second link for the Two-Link Flexible Manipulator. Both joints are clamped and θ_2=-90.

Chapter 7

Collocated Control of a Two-Link Flexible Manipulator

7.1 Introduction

This chapter deals with the collocated control of a two-link flexible manipulator. In most robotic applications, the reference commands are given in Cartesian space. In the case of a planar two-link manipulator, the position of the end point of the second link is defined by the x and y coordinates. In order to obtain the joint angles that lead the end point of the two-link manipulator to the desired position, the inverse kinematics of the corresponding two-link rigid manipulator has to be carried out. The inverse kinematics of a two-link rigid manipulator has been extensively covered in the literature. In Sections 7.2 and 7.3, brief descriptions of the derivations leading to the equations used to perform the inverse kinematics and the forward kinematics of a two-link manipulator are included for completeness. A collocated control scheme that uses two independent joint PD controllers to control the two-link flexible manipulator is presented in Section 7.4,. Section 7.5 deals with the control of the end point of the two-link flexible manipulator along a pre-defined path using the computed torque method. Simulation results and discussions about the effectiveness of each of these controllers are also presented in this chapter.

7.2 Inverse Kinematics

The desired position of the end point of a two-link manipulator in Cartesian space is given by $r_e(t)=[x_e \quad y_e]^T$. The corresponding joint angles assuming an elbow-up configuration are given by:

$$\theta_1 = \tan^{-1}(\frac{x_e}{y_e}) - \cos^{-1}(\frac{l_1^2-l_2^2+d^2}{2l_1l_2}) \tag{7.1}$$

$$\theta_2 = \cos^{-1}(\frac{l_1^2+l_2^2-d^2}{2l_1l_2}) - \pi \tag{7.2}$$

where $d^2=x_e^2+y_e^2$.

The velocity of the end point in Cartesian space is given by:

$$\dot{r}_e = J\,\Theta \tag{7.3}$$

Where J is the Jacobean matrix of the manipulator and given by:

$$J=\begin{bmatrix} -l_1s_1-l_2s_{12} & -l_2s_{12} \\ l_1c_1+l_2c_{12} & l_2c_{12} \end{bmatrix} \tag{7.4}$$

The symbols s_1, c_1, s_{ij} and c_{ij} refer to $s_1=\sin\theta_1$, $c_1=\cos\theta_1$, $s_{12}=\sin(\theta_1+\theta_2)$ and $c_{12}=\cos(\theta_1+\theta_2)$.

Provided that J is non singular, the joint velocities can be obtained as:

$$\dot{\Theta} = J^{-1}\,\dot{r}_e \qquad 7.5$$

The end point acceleration is given by:

$$\ddot{r}_e = \frac{dJ}{dt}\,\dot{\Theta} + J\,\ddot{\Theta} \qquad 7.6$$

Given $\ddot{r}_e$, $\dot{\Theta}$ and J, the joint accelerations can be obtained from Equation 7.6 as:

$$\ddot{\Theta} = J^{-1}\left[\ddot{r}_e - \frac{dJ}{dt}\dot{\Theta}\right] \qquad 7.7$$

7.3 Forward Kinematics

The output of the two-link flexible manipulator is given in terms of the joint angles, the deflections and the slopes at the end of each link, as well as their corresponding first time derivatives. In order to get the absolute position of the end point in Cartesian space, the forward kinematics of the manipulator has to be used. Given θ_1, θ_2, ξ_{e1}, ξ_{e2}, δ_{e1} and δ_{e2}, as well as their corresponding first time derivatives, the end point position is given by:

$$r_e = R_0^1\left[\begin{bmatrix} l_1 \\ \xi_{e_1} \end{bmatrix} + R_1^2\begin{bmatrix} l_2 \\ \xi_{e_2} \end{bmatrix}\right] \qquad 7.8$$

and the end point velocity is given by differentiating Equation 7.8 with respect to time as:

$$\dot{r}_e = \dot{R}_0^1\left[\begin{bmatrix} l_1 \\ \xi_{e_1} \end{bmatrix} + R_1^2\begin{bmatrix} l_2 \\ \xi_{e_2} \end{bmatrix}\right] + R_0^1\left[\begin{bmatrix} 0 \\ \dot{\xi}_{e_1} \end{bmatrix} + \dot{R}_1^2\begin{bmatrix} l_2 \\ \dot{\alpha}_{e_2} \end{bmatrix} + R_1^2\begin{bmatrix} 0 \\ \dot{\xi}_{e_2} \end{bmatrix}\right] \qquad 7.9$$

7.4 Independent Joint PD Control

Independent joint PD (Proportional plus Derivative) Controllers have been used in the control of rigid manipulators. These controllers generate the commands to produce the actuators torques to be applied at each joint based only on signals from collocated joint sensors. The torque at each joint is given by:

$$\tau_i = K_{P_i} e_i + K_{D_i} \dot{e}_i \qquad i=1,2 \qquad 7.10$$

where $e_i=\theta_{di}-\theta_i$ is the error representing the deviation of the actual joint angle θ_i from the desired one given by θ_{di}. $\dot{e}_i$ is the time derivative of e_i. K_{Pi} and K_{Di} are the proportional and derivative gains, respectively. Figure 7.1 shows a block diagram of the independent joint PD controller for a two-link flexible manipulator. The reference commands describing the desired end point position are given in Cartesian space as x_d and y_d which represent the desired positions along the x and y coordinates, respectively. The corresponding joint angles are obtained from the inverse kinematics of the equivalent rigid manipulator. This controller is referred to as a collocated controller because only the joint angles and their first time derivatives measured at the joints are used in the feedback. The gains of each joint controller are usually selected to result in a critically damped response for for the configuration corresponding to the highest inertia with respect to the joint axis. For the two link flexible manipulator, it was shown that the inertia matrix depends not only on the joint angles but also on the variable describing the flexible sub-system. To determine the controller gains for the first joint, the two links are assumed to be rigid and fully extended with the second joint locked. This configuration represents the highest inertia with respect to the axis of the first joint. For the second joint, the highest inertia configuration results when the link is assumed to be rigid. With reference to Figure 6.2, the highest inertia with respect to the axis of the first joint is given by:

$$\hat{I}_1 = \left(\frac{m_2 l_2}{3} + m_L\right) l_2^2 + \left(\frac{m_1 l_1}{3} + m_2 l_2 + m_a + m_L\right) l_1^2 + (m_2 l_2 + 2m_L)\, l_1 l_2 + I_L + I_a \qquad 7.11$$

The highest inertia of the second link with respect to the axis of its joint is given by:

$$\hat{I}_2 = \left(\frac{m_2 l_2}{3} + m_L\right) l_2^2 + I_L \qquad 7.12$$

The error dynamics when independent PD controllers are used to control the two-links with inertias $\hat{I}_1$ and $\hat{I}_2$ is:

$$\begin{bmatrix} \hat{I}_1 & 0 \\ 0 & \hat{I}_2 \end{bmatrix} \begin{bmatrix} \ddot{e}_1 \\ \ddot{e}_2 \end{bmatrix} + \begin{bmatrix} K_{D_1} & 0 \\ 0 & K_{D_2} \end{bmatrix} \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \end{bmatrix} + \begin{bmatrix} K_{P_1} & 0 \\ 0 & K_{P_2} \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad 7.13$$

Which is equivalent to:

$$\begin{bmatrix} \ddot{e}_1 \\ \ddot{e}_2 \end{bmatrix} + \begin{bmatrix} 2\zeta_1\omega_{n_1} & 0 \\ 0 & 2\zeta_2\omega_{n_2} \end{bmatrix} \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \end{bmatrix} + \begin{bmatrix} \omega_{n_1}^2 & 0 \\ 0 & \omega_{n_1}^2 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad 7.14$$

where $2\zeta_i\omega_{n_1}^2 = K_{D_i}/\hat{I}_i$ and $\omega_{n_i} = (K_{P_i}/\hat{I}_i)^{1/2}$; $i=1,2$.

ζ_i is the desired damping ratio associated with the motion of joint i, and ω_{ni} is the control bandwidth. The independent joint PD controller can be viewed as having a torsional spring and a damper at each joint. Each of the spring-damper pairs guides the system to the desired configuration and removes energy from the system. Simulations of a two-link flexible manipulator under the independent joint PD controller were performed and the results are shown in Figures 7.2 to 7.6. The physical parameters of the two link flexible manipulator used in the simulations are shown in table 6.1. The controller gains are selected such that each joint is critically damped with ω_{ni} = 10 rad/sec. The initial position of the end point of the manipulator is represented in Cartesian space by the point $P_i(0.8,0.1)$ and the final position is represented by the point $P_f(0.1,1.6)$. The joint angles corresponding to the initial and final positions of the end point are obtained from the inverse kinematics and they are given by Θ_i=[73 -132] and Θ_f=[125 -75]. Figure 7.2 shows the error between the actual and desired joint angles. It should be noted that even though the controller gains were selected to give an critically damped joint response, both joints have overshot their target position. This overshoot is mainly due to the nonlinear terms representing the coriolis and centrifugal forces as well as the coupling between the rigid and flexible sub-systems. Figure 7.3 shows the deflection at the end of the first link and at the end of the second link. The deflection at the end of the first link is much higher than that at the end of the second link because the later acts as a load for the first link. For both links the largest deflection occurs at the early stage of the transient response and this is due to the high input command introduced by the large error between the initial and final joint positions of the manipulator. The input torques produced by the actuators are shown in Figure 7.4. The motion of the end point of the two link flexible manipulator in the Cartesian space in obtained through the forward kinematics and is shown in Figure 7.5. A large overshoot is observed in both x and y directions. This large overshoot in Cartesian space is due to the overshoot in the joint angles as well as the deflections at the end of each of the two links. An illustration of the motion of the two link flexible manipulator from P_i to P_f is shown in Figure 7.6.

7.5 Computed Torque Control

It was shown in the previous section that an independent joint control strategy is not suitable for controlling the manipulator along a pre-defined path. The computed torque control approach, which uses a nonlinear feedback in order to linearize and decouple the two-link manipulator, is used here as an alternative to the independent joint PD control scheme. A block diagram of the computed torque controller of the two-link flexible manipulator is shown in Figure 7.7. The reference trajectory to be followed by the manipulator's end point is specified in Cartesian space. Through the inverse kinematics of a two-link rigid manipulator, the corresponding angle, velocity and acceleration of each joint are obtained and they are given by $\Theta_d=[\theta_{1d} \quad \theta_{2d}]^T$; $\dot{\Theta}_d=[\dot{\theta}_{1d} \quad \dot{\theta}_{2d}]^T$ and $\ddot{\Theta}_d=[\ddot{\theta}_{1d} \quad \ddot{\theta}_{2d}]^T$, respectively. Since the feedback sensors and the actuators are collocated at the joints, an equation describing the motion of the rigid sub-system is needed. Similar to the procedure shown in Chapter 4, an equation similar to Equation 4.4 can be obtained. This equation is given by:

$$\boldsymbol{\Gamma}_0\ddot{\Theta} + \boldsymbol{\Lambda}_0 = \mathrm{T} \qquad 7.15$$

where $\boldsymbol{\Gamma}_0$ is a 2 by 2 positive definite matrix and Λ_0 is a 2 by 1 vector representing the non linear terms. From Figure 7.7, the vector representing the joint torques is given by:

$$\mathrm{T} = \boldsymbol{\Gamma}_0(\ddot{\Theta}_d + K_D(\dot{\Theta}_d - \dot{\Theta}) + K_P\ (\Theta_d - \Theta)) + \boldsymbol{\Lambda}_0 \qquad 7.16$$

K_P and K_D are two 2 by 2 diagonal matrices representing the proportional and derivative gains respectively. Substituting 7.16 into 7.15 gives a system of two linear and decoupled second order equations representing the dynamics of the error between the actual and the desired joint angles. This system of equations is given by :

$$\ddot{\Theta}_d - \ddot{\Theta} + K_D\ (\dot{\Theta}_d - \dot{\Theta}) + K_P\ (\Theta_d - \Theta) = 0 \qquad 7.17$$

The actual position of the end point in Cartesian space is obtained through the forward kinematics of the two-link flexible manipulator.

7.6 Straight Line Test Trajectory

In order to simulate the response of the two-link flexible manipulator under the computed torque control, a straight line trajectory of the end point is being used. Figure 7.8 shows the two-link flexible manipulator at several configurations along this trajectory. The equation of the straight line from $P_i\,(x_i, y_i)$ to $P_f\,(x_f, y_f)$ in the xy plane is given by:

$$y = \frac{y_i - y_f}{x_i - y_i}(x - x_i) + y_i \qquad 7.18$$

The coordinates x and y and their first and second time derivatives were selected to have continuous and smooth profiles in the time domain. The function *x(t)* and its time derivatives are given by:

$$x(t) = x_i + \frac{x_f - x_i}{T}t - \frac{x_f - x_i}{T\omega}\sin(\omega\ t) \qquad 7.19a$$

$$\dot{x}(t) = \frac{x_f - x_i}{T} - \frac{x_f - x_i}{T}\cos(\omega\ t) \qquad 7.19b$$

$$\ddot{x}(t) = \omega\frac{x_f - x_i}{T}\sin(\omega\ t) \qquad 7.19c$$

where $\omega=2\pi/T$ and T is the time required for the manipulator to move from P_i to P_f. Similarly, $y(t)$, $\dot{y}(t)$ and $\ddot{y}(t)$ are given by:

$$y(t) = y_i + \frac{y_f - y_i}{T}t - \frac{y_f - y_i}{T\omega}\sin(\omega\ t) \tag{7.20a}$$

$$\dot{y}(t) = \frac{y_f - y_i}{T} - \frac{y_f - y_i}{T}\cos(\omega\ t) \tag{7.20b}$$

$$\ddot{y}(t) = \omega\frac{y_f - y_i}{T}\sin(\omega\ t) \tag{7.20c}$$

Figure 7.9 shows the desired trajectories of the end point along the x and y directions and the corresponding trajectories in joint space. The manipulator starts form the point P_i $(0.8, 0.1)$ where it was at rest. After a duration of $T=2.0$ *sec*, the manipulator is expected to reach the point $P_f(0.1, 1.6)$ and stay at this location for one second so that the induced oscillations at the tip can be observed. The errors in the joint positions are shown in Figure 7.10. It should be noted that in the case of a rigid two-link manipulator under similar control strategy, these errors would be very small. The errors seen in this figure are mainly due to the flexibility of the links. The deflections at the end of each link during the tracking of the straight line trajectory are shown in Figure 7.11. The figure shows that the deflection at the end of the first link is much larger than that at the end of the second link and this is due to the different load condition for each link. Even though the errors in the joints positions are zero after a duration of 2 sec, the end point of each link keeps oscillating and the actuators at each joint continue to provide the necessary torques to maintain the joint angles at the desired position as shown in Figure 7.12. The path of the end point as it moves from P_i to P_f in the xy plane is shown in Figure 7.13. The deviations of the end point from the desired path cannot be clearly seen in this figure because they are very small compared to the overall path. These deviations can be seen very clearly in Figure 7.14, where the errors in the x and y positions of the end point are shown. Again, after a duration of 2 sec, the motion of the end point is that of an undamped oscillatory system. These oscillations represent the zero dynamics of the system.

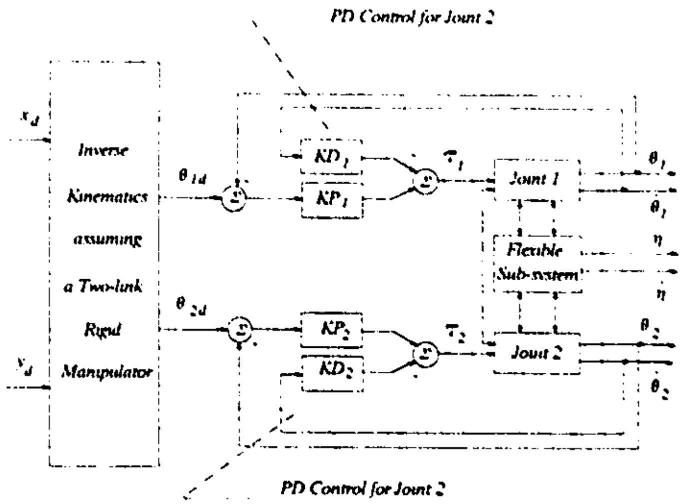


Figure 7.1: Block diagram of the independent joint PD controller.

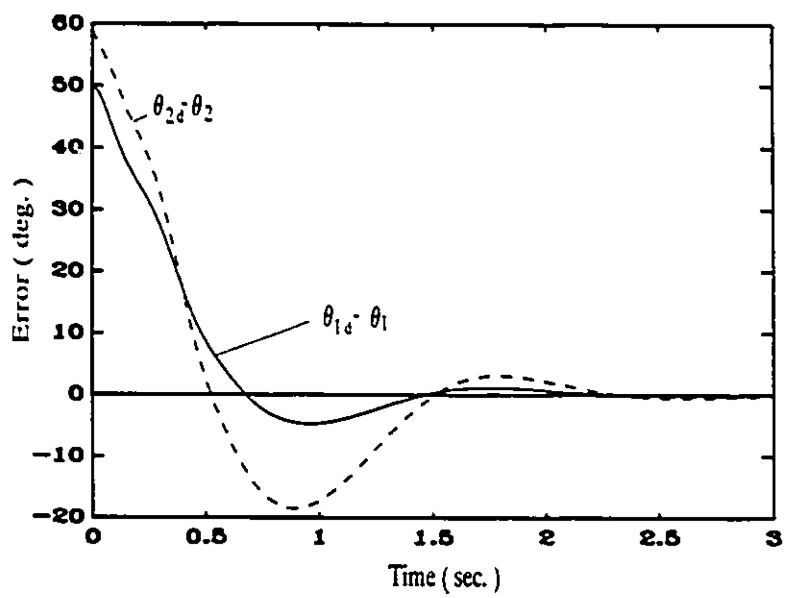


Figure 7.2: Error in the joint positions.

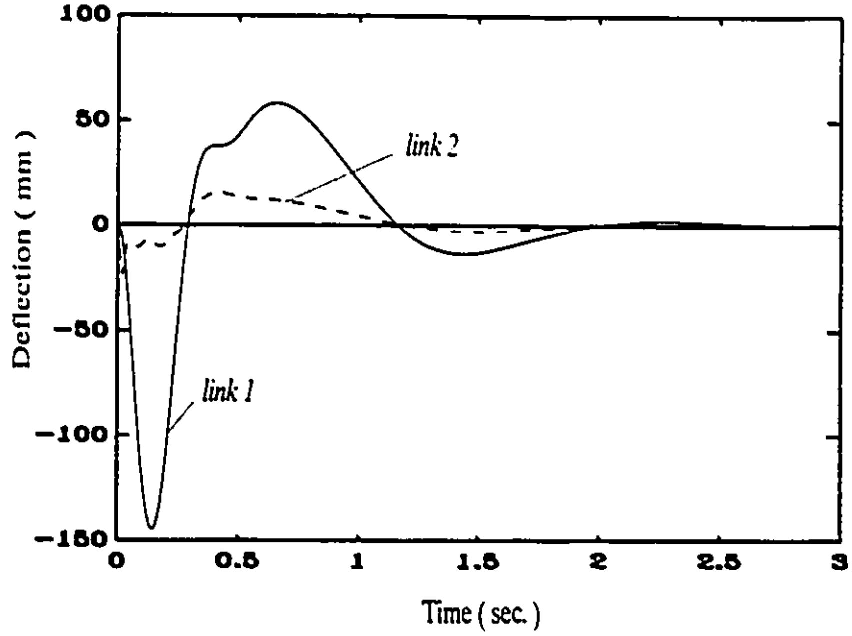


Figure 7.3: Deflections of the end points of link 1 and link 2.

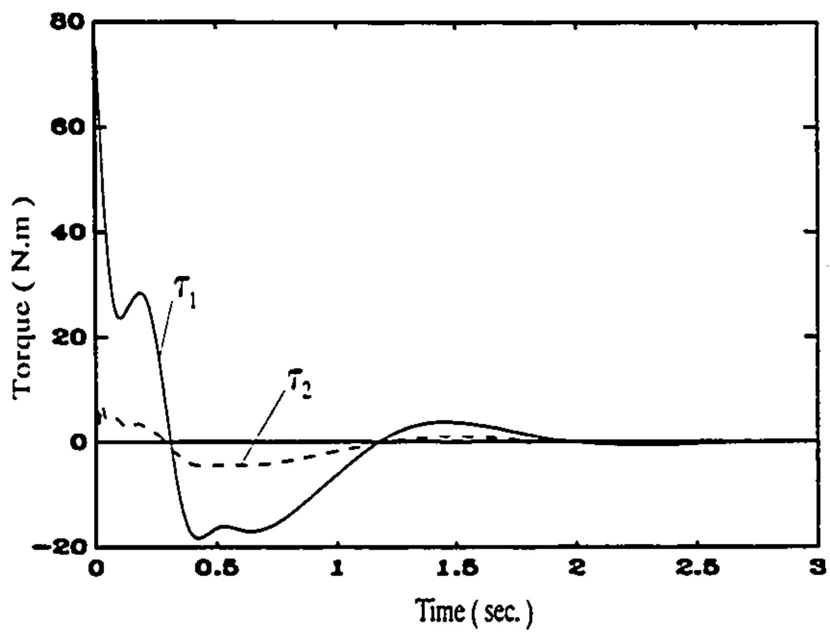


Figure 7.4: Actuators Torques at joint 1 and joint 2.

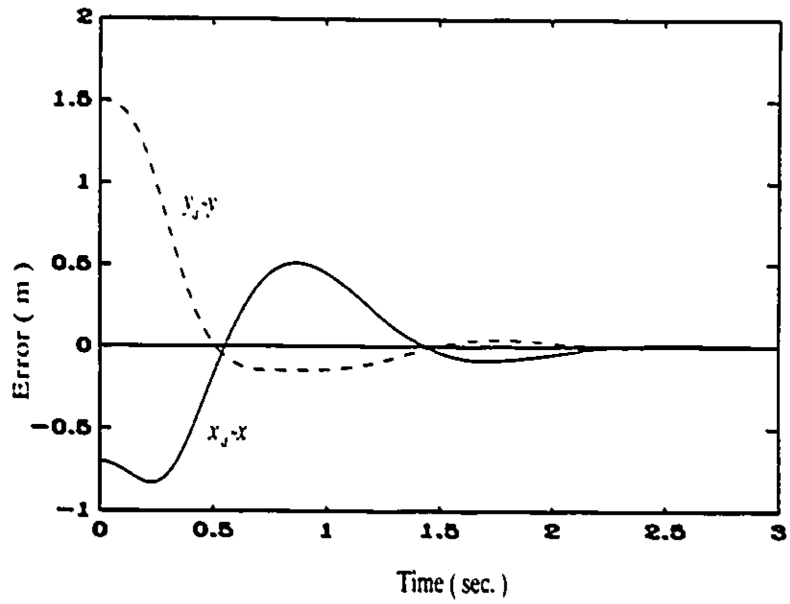


Figure 7.5: Errors in the x and y positions of the end point.

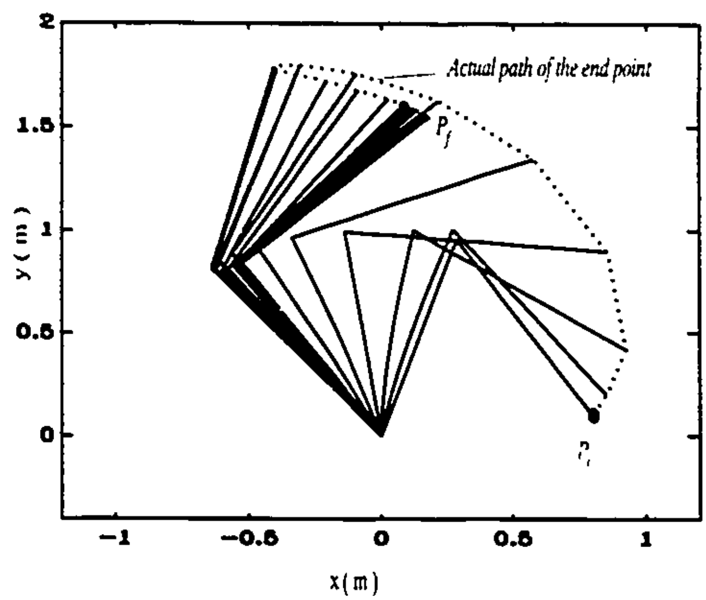


Figure 7.6: Path of the end point of the two link flexible manipulator in the x-y plane.

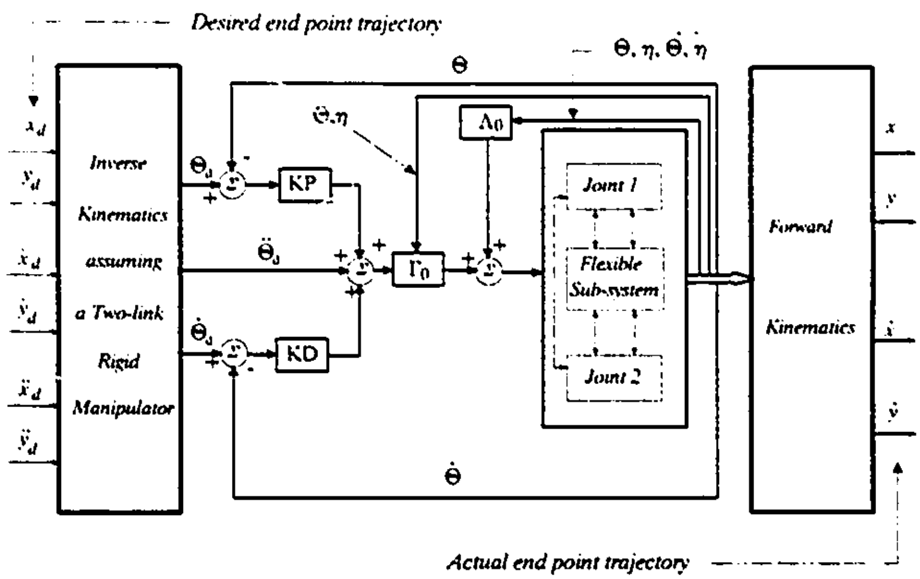


Figure 7.7: Block diagram of the Computed Torque for the Collocation Control of a Two-link Flexible Manipulator.

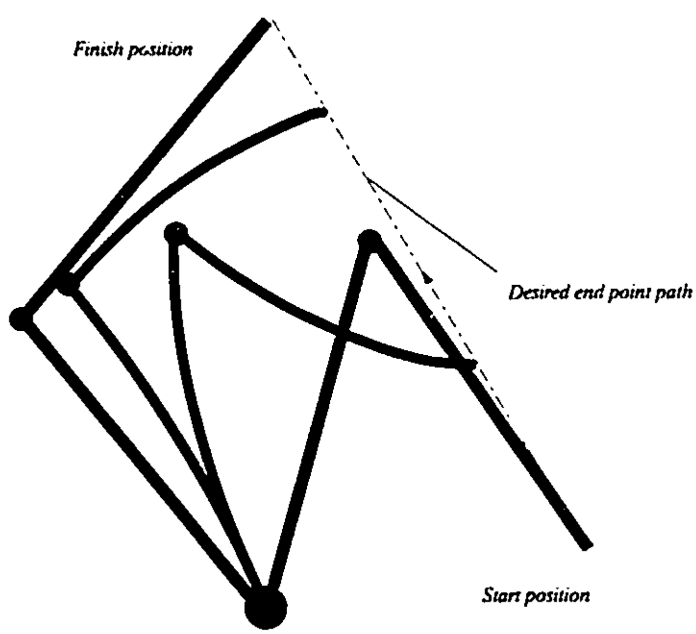


Figure 7.8: Desired straight line trajectory in Cartesian space.

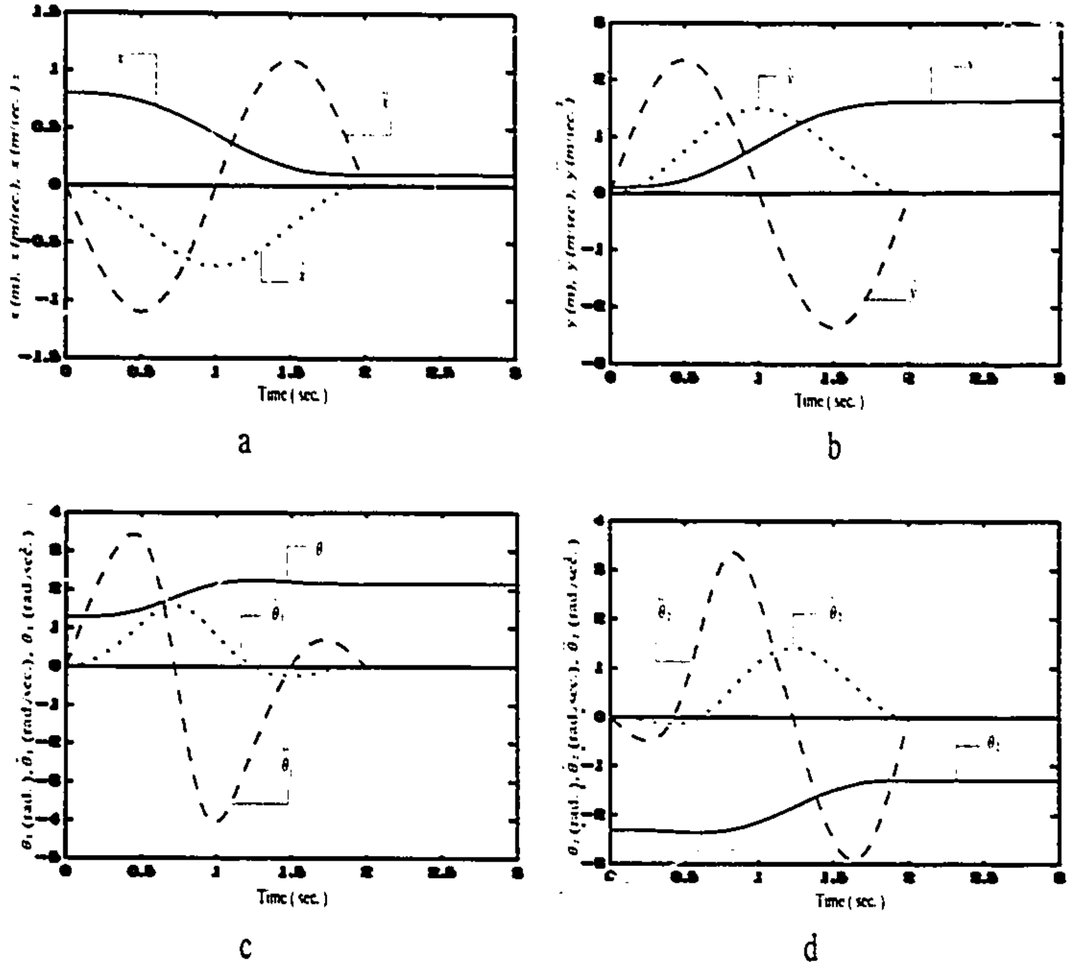


Figure 7.9: (a) and (b) represent the desired trajectory in the x and y plane.
(c) and (d) represent the desired trajectory in the joint coordinates.

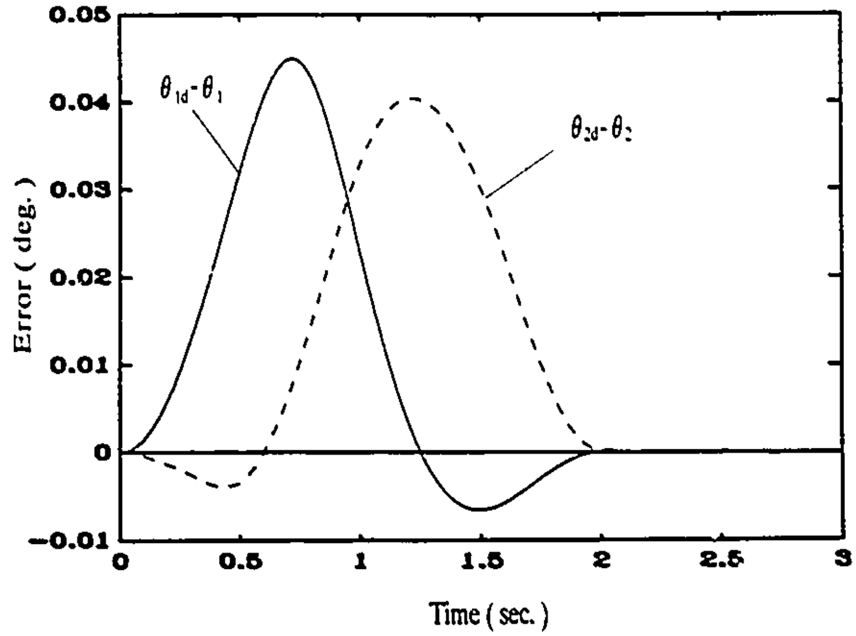


Figure 7.10: Error in joint angles using a computed torque control.

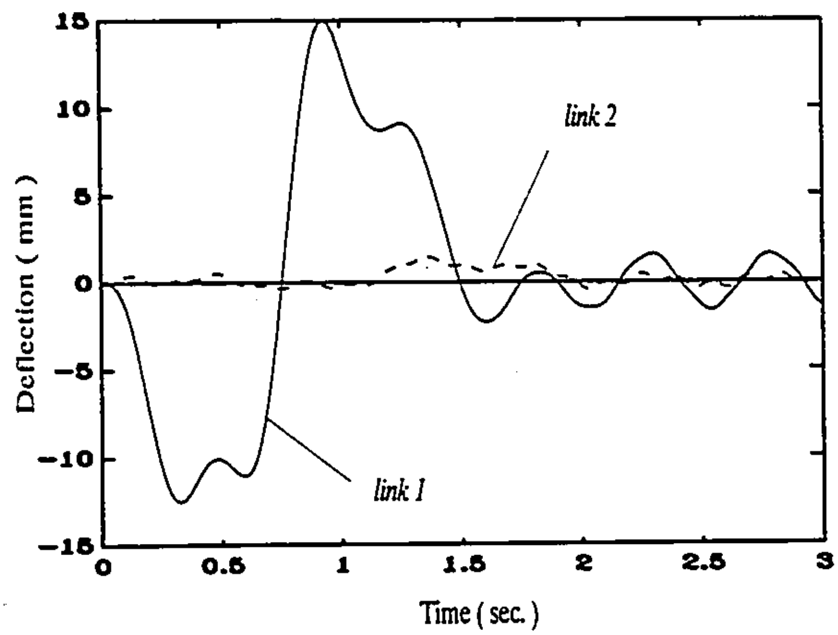


Figure 7.11: Deflections at the end of link 1 and link 2 using a computed torque control.

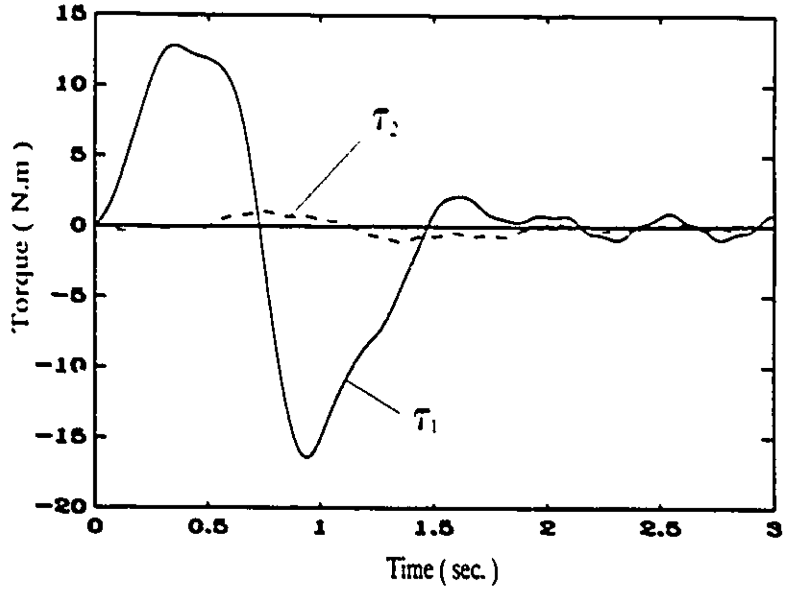


Figure 7.12: Input torques at the joints using a computed torque control.

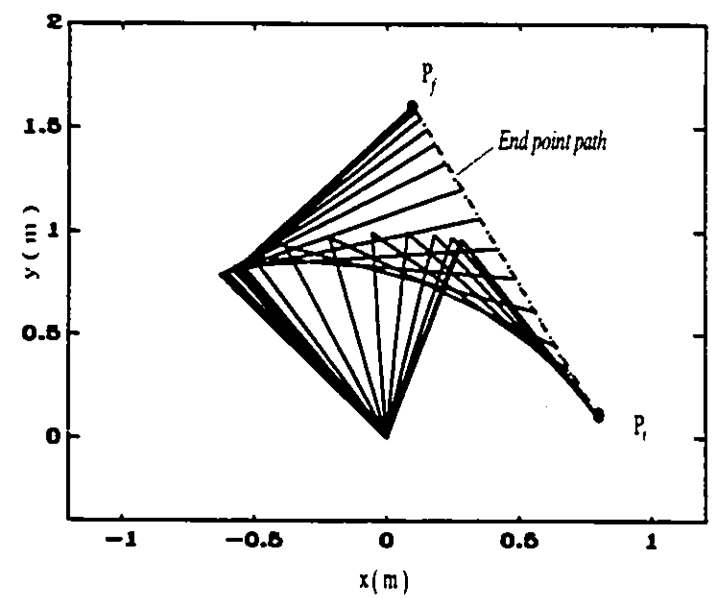


Figure 7.13: Trajectory of the end point of the two link flexible manipulator in the xy plane and using a computed torque control.

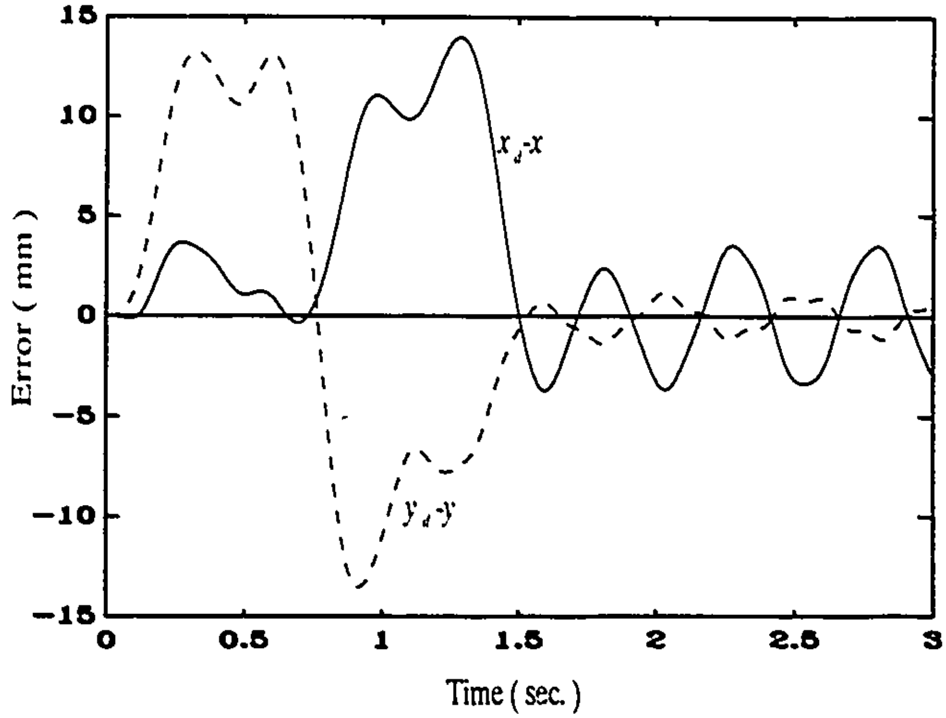


Figure 7.14: Errors in the x and y positions of the end point using a computed torque control.

Chapter 8

Non-Collocated/Time Delay Control

8.1 Introduction

It was shown in Chapter 5 that the use of the computed torque technique together with the delayed deflection of a point along the link introduces an active damping effect on the flexible modes of the one link flexible manipulator. This results in a significant reduction in end point vibrations. In this chapter, the same approach is extended to the two-link flexible manipulator. This approach requires feedback from one point along each link. These points are selected such that when the angles representing the positions of these points are used as the system output, the overall dynamic system is stable. Simulation of the response of the two-link flexible manipulator under non-collocated control is presented in Section 8.2. In Section 8.3, the Computed Torque/Delayed Deflection approach and the effects of the delay time on the end point position of the two-link flexible manipulator are investigated. A comparative study of the response of the two-link flexible manipulator under collocated control, non-collocated control, and non-collocated control with delayed deflection has been carried out in Section 8.4 and 8.5. In Section 8.4, a trajectory consisting of a straight line path has been used. A more complicated end point trajectory has also been used in Section 8.5 in order to test a wider area of the work plane.

8.2 Non-Collocated Control

The non-collocated control of the two-link flexible manipulator entails controlling the angle of a given point along each link. The necessary variables required to express the system dynamics in terms of the new output variables are shown in Figure 8.1. With reference to the figure, the angle of the position vector of a point on the first link located at a distance x_1 from the first joint is given by:

$$\theta_{x_1} = \theta_1 + \frac{u_1(x_1)}{x_1} \qquad 8.1$$

Similarly, the angle of a point located at a distance x_2 along the second link and measured from the location of the second joint is given by:

$$\theta_{x_2} = \theta_2 + \frac{u_2(x_2)}{x_2} \qquad 8.2$$

$u_1(x_1)$ and $u_2(x_2)$ are the deflections at x_1 and x_2 respectively. These deflections can be expressed in terms of the shape functions and the coordinates of the end point of each link as:

$$u_1(x_1) = \phi_{11}(x_1)\,\xi_{e_1} + \phi_{12}(x_1)\,\delta_{e_1} \qquad 8.3$$

and

$$u_2(x_2) = \phi_{21}(x_2)\,\xi_{e_2} + \phi_{22}(x_2)\,\delta_{e_2} \qquad 8.4$$

θ_{x1} and θ_{x2} can be expressed in vectors form as:

$$\Theta_x = \Theta + \Psi\eta \qquad 8.5$$

where

$$\Psi = \begin{bmatrix} \dfrac{\phi_{11}(x_1)}{x_1} & \dfrac{\phi_{12}(x_1)}{x_1} & 0 & 0 \\ 0 & 0 & \dfrac{\phi_{21}(x_2)}{x_2} & \dfrac{\phi_{22}(x_2)}{x_2} \end{bmatrix} \qquad 8.6$$

From Equation 8.5, the vector Θ can be written as:

$$\Theta = \Theta_x - \Psi\eta \qquad 8.7$$

Substituting Equation 8.7 into the dynamic Equations 6.32 and 6.33 gives:

$$M_r\,\Theta_x + (\,M_{rf}^T - M_r\,\Psi)\,\ddot{\eta} + H_r = \mathrm{T} \qquad 8.8$$

$$M_{rf}\,\Theta_x + (\,M_f - M_{rf}\,\Psi)\,\ddot{\eta} + H_f + \mathrm{K}\eta = 0 \qquad 8.9$$

Similar to the case of the one-link flexible manipulator that was discussed in Chapter 4, the equation representing the dynamic system when the output Θ_x reaches its final destination is given by:

$$(M_f - M_{rf}\ \Psi)\ \ddot{\eta} + H_f^{*} + K\eta = 0 \qquad 8.10$$

H_f^{*} is the term H_f with $\theta_{x_1} = \theta_{x_2} = 0$.

The stability of the system given by Equation 8.10 can be assured if the matrix $M_x = M_f - M_{rf}\Psi$ is positive definite. The main difficulty in dealing with the singularity of the matrix M_x results from its dependance not only on the non-collocation distances x_1 and x_2 but also on the variables θ_2, ξ_{e1}, ξ_{e2}, δ_{e1} and δ_{e2}. Sensitivity analysis of the matrix M_x to a change in θ_2, ξ_{e1}, ξ_{e2}, δ_{e1} or δ_{e2} shows it to be most sensitive to changes in θ_2. The determinant of M_x for different configurations and as θ_2 varies from $-\pi$ to 0 is shown in Figures 8.2, 8.3, 8.4 and 8.5. From these figures, one set of points $\{x_1, x_2\}$ for which the matrix M_x is non singular is $x_1=0.6l_1$ and $x_2=0.4l_2$. Figure 8.6 shows the error between the desired and the actual joint angle for the joints of the two-link flexible manipulator. The errors in the joint angles are relatively large as compared to those when a collocated controller is used (refer to Figure 7.7). These large errors are due to the fact that in the non collocated control, the joint angles θ_1 and θ_2 are not the controlled variables. Rather, the controlled variables are the angles of the position vectors of the points x_1 and x_2. The deflections at the end of the first and second link shown in Figure 8.7 are almost the same as those obtained under collocated control (refer to Figure 7.11). This is due to the fact that in both cases, the end point deflection is measured with respect to the tangent to the link at the location of the joint. The position error of the end point along the x and the y directions is shown in Figure 8.8. The advantage of using the non collocated control can be seen in Figures 8.9 and 8.10 where the position errors of the end point along the x and y directions are shown for both the collocated and the non collocated control. The maximum position error in the x direction has been reduced from *14 mm* under the collocated control to *5 mm* under the non collocated control. Similarly, the position error in the y direction has been reduced from *13 mm* to *6 mm*. Another measure of the trajectory tracking performance for each of the collocated and non collocated control is the total position error which gives the total deviation of the actual end point position from the desired one. Figure 8.11 is an illustration of this type of error. The total error is calculated from the errors along the x and y directions as:

$$e_{total} \quad \sqrt{(x_d-x)^2+(y_d-y)^2} \qquad 8.11$$

Figure 8.12 shows the total position error of the end point under both collocated and non collocated control. The figure clearly shows that throughout the motion, the non collocated control out performed the collocated one. The input torques at each joint under non collocated control are shown in Figure 8.13. These torques exhibit a high frequency switching in order to maintain the angles θ_{x1} and θ_{x2} along the desired trajectories. This high frequency switching is not present when a collocated control is used.

8.3 Time Delay Control

In this section, the computed torque/Time delay approach that was discussed in Chapter 5 is being used to reduce the vibrations at the end point of the two-link flexible manipulator. It was shown in the previous section that even though the position error of the end point is reduced through the use of non collocated control, residual vibrations are still present. In order to introduce active damping into the system, a delay was introduced in the deflections at points x_1 and x_2. In order to see the effect of the delay time on the tracking performance of the two-link flexible manipulator, simulations for a delay time of $\nu = 0.01$ *sec*, $\nu = 0.1$ *sec* and $\nu = 0.2$ *sec* have been carried out for the case where $x_1 = 0.6\ l_1$ and $x_2 = 0.4\ l_2$. Figures 8.14 and 8.15 show the deflections at the end of the first and the second links, respectively. These two figures show that by using a delay time of *0.01 sec*, the high frequency components resulting from the non-collocated control (v=0) have been virtually eliminated, thus implying that the time delay control acts as a low pass filter. Increasing the delay time to *0.1 sec* nearly eliminates all the vibrations that were present at the final stage of the trajectory. A further increase in delay time from *0.1 sec* to *0.2 sec* introduces another high frequency component in the signal. The effects of time delay control on the motion of the end point of the two-link flexible manipulator are shown in Figures 8.16 to 8.18. The figures show the error in the x and y directions and the total error between the desired and the actual position of the end point of the manipulator. The damping effect introduced by the CTDD approach can be seen very clearly in Figure 8.18 where the signal representing the total position error decays much faster in comparison to the non collocated control approach (v=0).

8.4 Comparative Study

In the following section, a comparative study has been carried out to asses the tracking performance of the two-link flexible manipulator under Collocated Control, Non-Collocated Control, and the CTDD approach. Two types of trajectories have been used in the study. The first trajectory is the straight line path shown in Figure 7.8. The second trajectory consists of a set of three straight line paths connecting four points in the xy plane as shown in Figure 8.23. The end point of the two link-flexible manipulator starts from an initial position $P_i\ (x_i\ .\ y_i\ )$ and finishes at the position given by $P_f\ (x_f\ ,y_f)$.

8.4.1 Straight Line Path

Figures 8.19 and 8.20 show the errors in the x and y directions respectively. It can be seen from these figures that the highest error is obtained under collocated control. Even though these errors have been reduced when non-collocated control was used, small end point oscillations are still appearing. By introducing active damping into the system through the use of the CTDD approach, these oscillations are successfully eliminated. The total error, as defined by Equation 8.11, is shown in Figure 8.21 and the trajectory of the end point in the xy plane is shown in Figure 8.22. It can be seen from zooming on the areas near the starting and finishing points that the collocated control gives the worst performance. The non-collocated and time delay controls showed some resemblance at the start of the trajectory but the advantage of the time delay control becomes very apparent near the finishing point. Table 8.1 is a summary of the results obtained from

Figures 8.19, 8.20 and 8.21. The table contains a qualitative summary of the performance of the three control strategies.

	Collocated	Non-Collocated Without delay	Non-Collocated With delay (v=0.1 sec)
$E_{x,max}$	13.96 mm (t=1.296 sec)	4.92 mm (t=0.396 sec)	4.93 mm (t=0.396 sec)
$E_{y,max}$	13.23 mm (t=0.33 sec)	5.91 mm (t=0.588 sec)	5.61 mm (t=0.588 sec)
$E_{total,max}$	16.91 mm (t=0.948 sec)	7.06 mm (t=0.396 sec)	7.09 mm (t=0.402 sec)

Table 8.1: Maximum position errors of the end point during the tracking of straight line path using the collocated, non collocated and time delay controls.

$E_{x,max}$, $E_{y,max}$ and $E_{total,max}$ are the maximum values of the error along x, the error along y and the total error of the position of the end point, respectively. The time t represents the time at which the maximum occurs.

8.4.2 Four Points Path

This type of trajectory is shown in Figure 8.23. During this trajectory, the end point of the manipulator starts from rest at $P_1(0.8,0.1)$ and finishes at $P_4(0.3,0.6)$ passing by $P_2(1.3,0.6)$ and $P_3(0.8,1.1)$. The trajectory is composed of three segments $P_1 \rightarrow P_2$, $P_2 \rightarrow P_3$ and $P_3 \rightarrow P_4$. Each of these is represented by a straight line path in the xy plane. The main reason behind using this type of trajectory is to allow a wider area of the work plane to be tested and to see the effect of non zero initial conditions on the performance of the controllers when the end point moves from P_2 to P_3 and from P_3 to P_4. The desired position, velocity and acceleration of the end point along the x and y directions are shown in Figure 8.24. With reference to this figure, the manipulator takes 1 sec to move from one point to another. At each of P_2 and P_3, there is a pause of 0.25 sec. When the end point reaches point P_4, the manipulator is expected to remain at this configuration for a duration of 1.5 sec to compare the settling time of each of the three controllers. The maximum velocity during each segment is $v_{max} = \sqrt{v_{x,max}^2 + v_{y,max}^2}$ =1.414 m/sec and the maximum acceleration is $a_{max} = \sqrt{a_{x,max}^2 + a_{y,max}^2}$=4.45 m/sec^2. The corresponding desired trajectory in the joint coordinates are shown in Figure 8.25. With reference to this figure, the maximum velocity and the maximum acceleration of each joint depend on the configuration of the manipulator. Table 8.2 shows the maximum velocity and maximum acceleration of each joint as the end point moves along each of the three segments of the four points path.

	$P_1 \rightarrow P_2$	$P_2 \rightarrow P_3$	$P_3 \rightarrow P_4$
$\lvert\theta_{1,max}\rvert$ (rad/sec^2)	2.09	3.86	3.74
$\lvert\theta_{1,max}\rvert$ (rad/sec)	0.35	1.15	1.09
$\lvert\theta_{2,max}\rvert$ (rad/sec^2)	5.36	2.15	5.46
$\lvert\theta_{2,max}\rvert$ (rad/sec)	1.54	0.38	1.62

Table 8.2: Maximum values of the desired velocity and desired acceleration of joint 1 and joint 2 along each segment of the desired path.

Figures 8.26, 8.27 and 8.28 show the simulation results for the position error of the end point along the x direction, the y direction and the total position error, respectively. In the three figures, the maximum position error occurs when the manipulator is moving from P_2 to P_3 and under the collocated control. This is because most of the motion for that segment of the trajectory is in the joint of the first link and since the second link acts as a load to the first link, the latter exhibits larger deflection when it is moving with high acceleration. A summary of the results shown in Figures 8.26 to 8.28 is presented in Table 8.3.

	Collocated Control	Non-Collocated Control	Time Delay Control (ν=0.1 sec)
$E_{x,max}$	29.18 mm (t=1.92 sec)	5.19 mm (t=0.18 sec)	5.19 mm (t=0.18 sec)
$E_{y,max}$	50.06 mm (t=1.59 sec)	17.88 mm (t=1.46 sec)	18.04 mm (t=1.47 sec)
$E_{total,max}$	54.86 mm (t=1.90 sec)	18.09 mm (t=1.46 sec)	18.26 mm (t=1.47 sec)

Table 8.3: Maximum position errors of the end point during the tracking of a four points path using the collocated, non collocated and time delay controls.

For the collocated control, the maximum total error was found to be 54.86 mm and occurs at a time t=1.90 sec. From Figure 8.25, it can be seen that at the time when the maximum total error occurs (t=1.90 sec), the acceleration of the joint of the first link has a peak value of -3.4 rad/sec^2. Furthermore, when the end point is moving between P_2 and P_3, the two link manipulator has the highest arm extension which causes larger deflections of the links. A significant reduction in the maximum total error was obtained when non collocated and time delay controls were employed. Even though the maximum total error is nearly the same for the non collocated control and for the time delay control, the latter shows a much improved performance by reducing the amount of oscillations at end of each segment of the path. Figures 8.29 and 8.30 shows the tracking performance of each of the collocated control, the non collocated control and the CTDD near each of the four points of the path.

As another measure of the tracking performance of the three controllers, the areas under the curves of the total errors have been calculated. This performance index is known as the IAE (Integral Absolute Error). Figures 8.31 and 8.32 show the IAE of the end point position for the straight line path and for the four points path respectively. It can be seen from these figures that the highest IAE is when the manipulator is under collocated control. The CTDD approach gives the lowest IAE. Table 8.4 shows the values of the IAE when t=3 seconds. As $t \rightarrow \infty$, the values of the IAE for the collocated and the non-collocated control become infinitely large because the end point keeps oscillating around the target position with constant amplitudes. In the case of the CTDD approach, the amplitude of the position error decreases due to the active structural damping introduced by the delayed deflection. The IAE becomes constant when the oscillations at the tip are completely eliminated.

	Collocated Control		Non-collocated Control		CTDD approach	
IAE	Straight line	Four points path	Straight line	Four points path	Straight line	Four points path
	3.2	6.3	1.3	2.1	1.2	.6

Table 8.4: IAE performance index for the collocated, non collocated and time delay control.

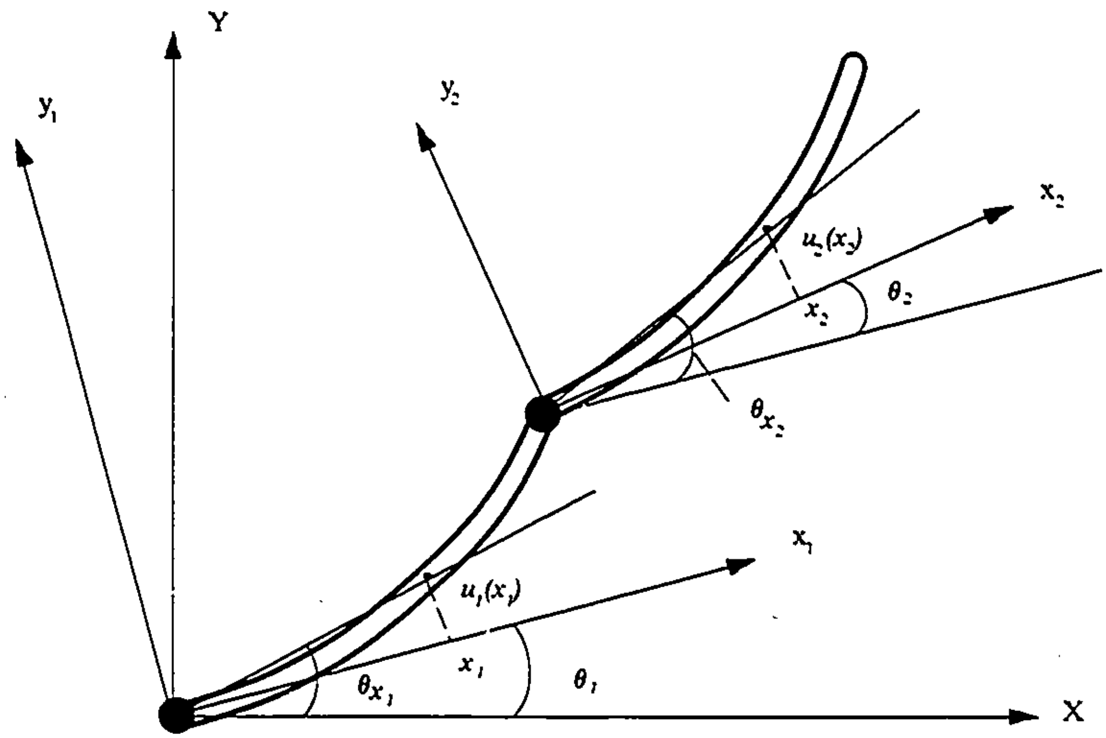


Figure 8.1: Coordinates Systems for the Non-Collocated Control of a Two-Link Flexible Manipulator.

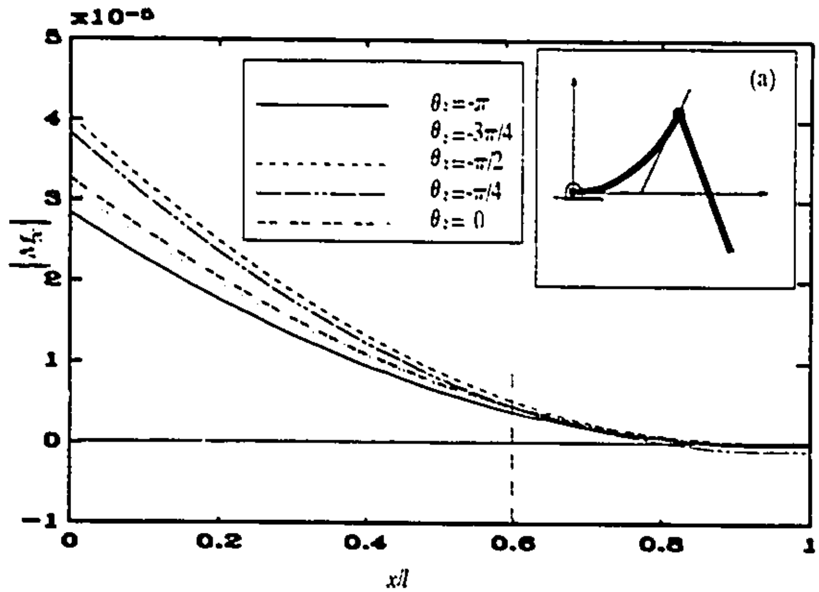


Figure 8.2: Determinant of the GIM of the two link flexible manipulator for ξ_{e1}=0.01l_f and δ_{e1} = 0.01.

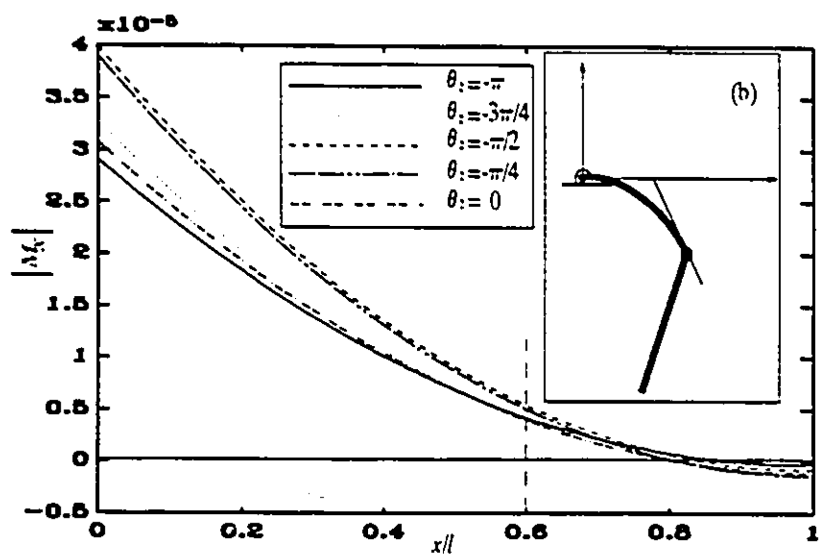


Figure 8.3: Determinant of the GIM of the two link flexible manipulator for ξ_{e1} = -0.01l_f and δ_{e1} = - 0.01. The second link is assumed to be rigid.

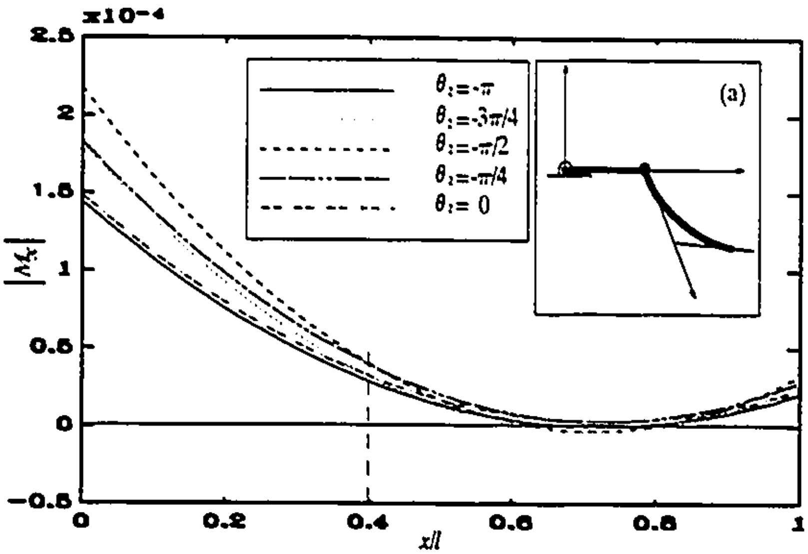


Figure 8.4: Determinant of the GIM of the two link flexible manipulator when $\xi_{e2} = 0.01\ l_2$ and $\delta_{e2} = 0.01$. The first link is assumed to be rigid.

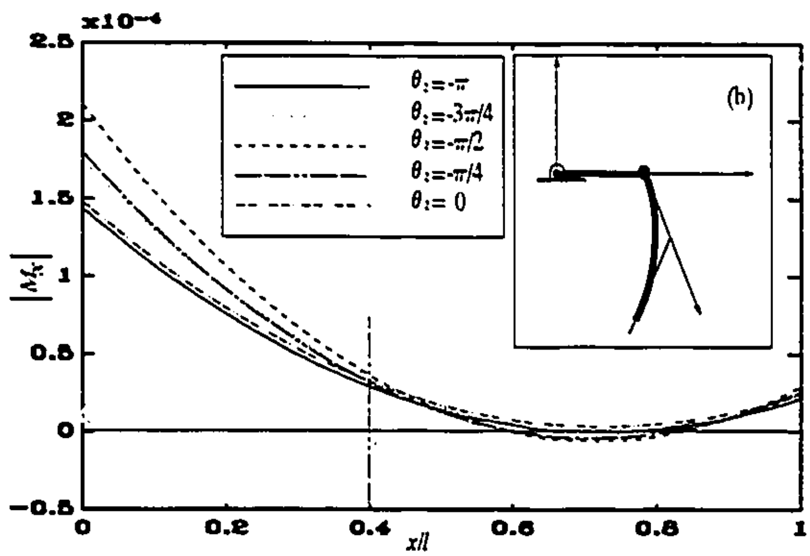


Figure 8.5: Determinant of the GIM of the two link flexible manipulator when $\xi_{e2} = -\ 0.01 l_2$ and $\delta_{e2} = -\ 0.01$. The first link is assumed to be rigid.

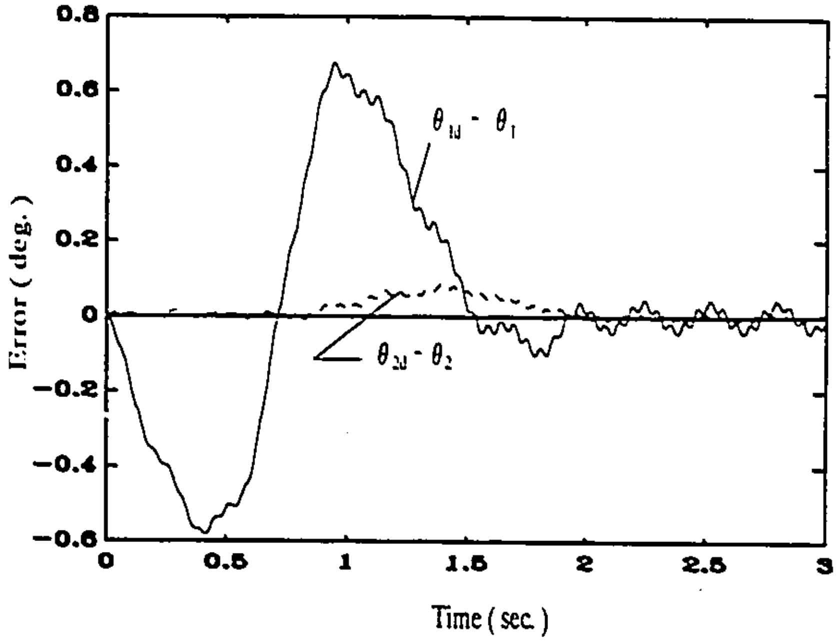


Figure 8.6: Error in the position of the joints using the Non-Collocated Control for the tracking of a straight line path. $x_1/l_1=0.6$ and $x_2/l_2=0.4$.

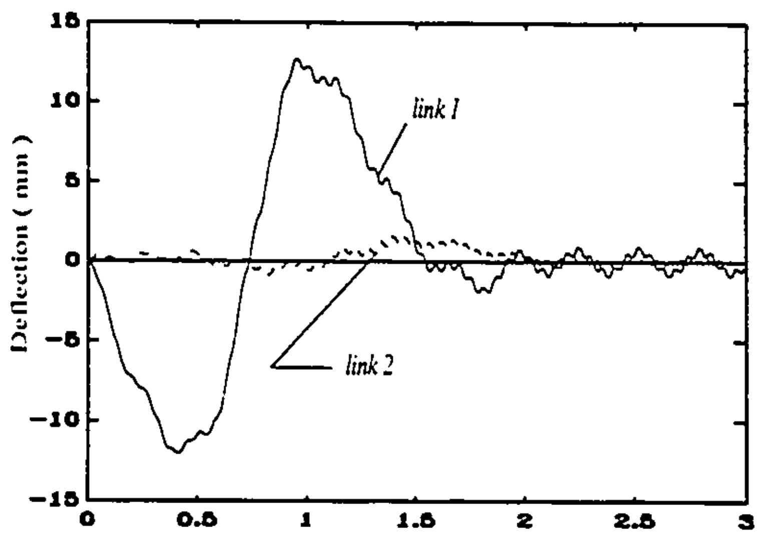


Figure 8.7: Deflections at the end of link 1 and link 2 using the Non-Collocated Control for the tracking of a straight line path. $x_1/l_1=0.6$ and $x_2/l_2=0.4$.

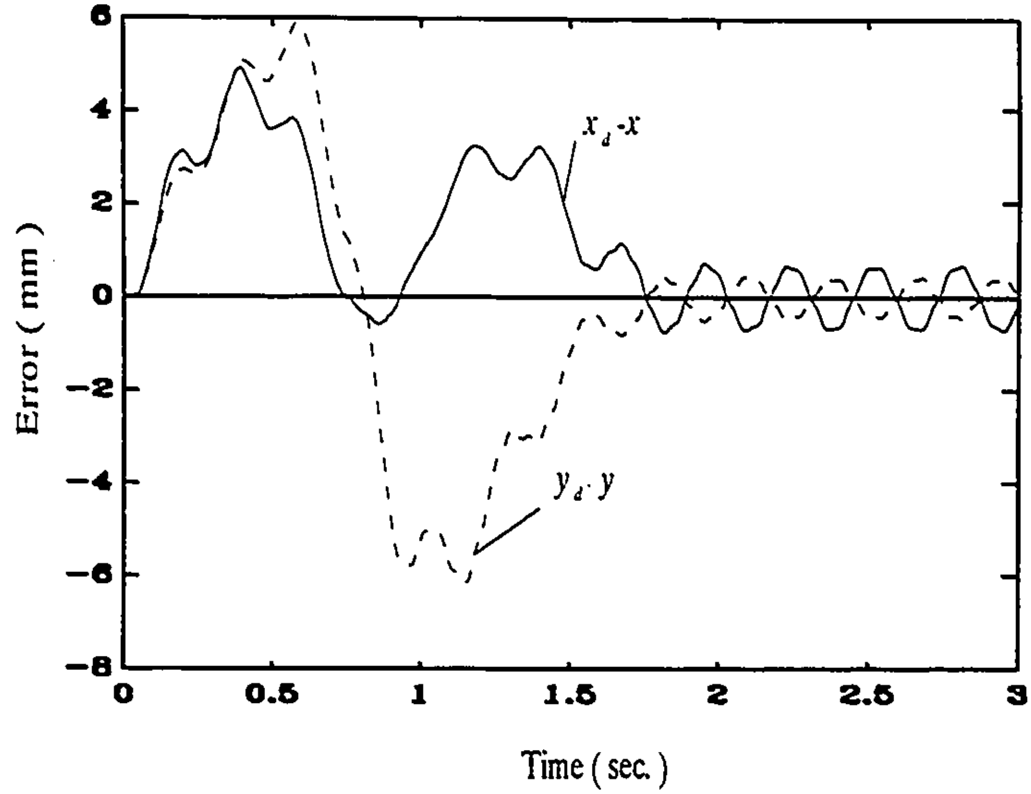


Figure 8.8: Errors in the end point position along the x and y directions using the Non-Collocated Control for the tracking of a straight line path. $x_1/l_1=0.6$ and $x_2/l_2=0.4$.

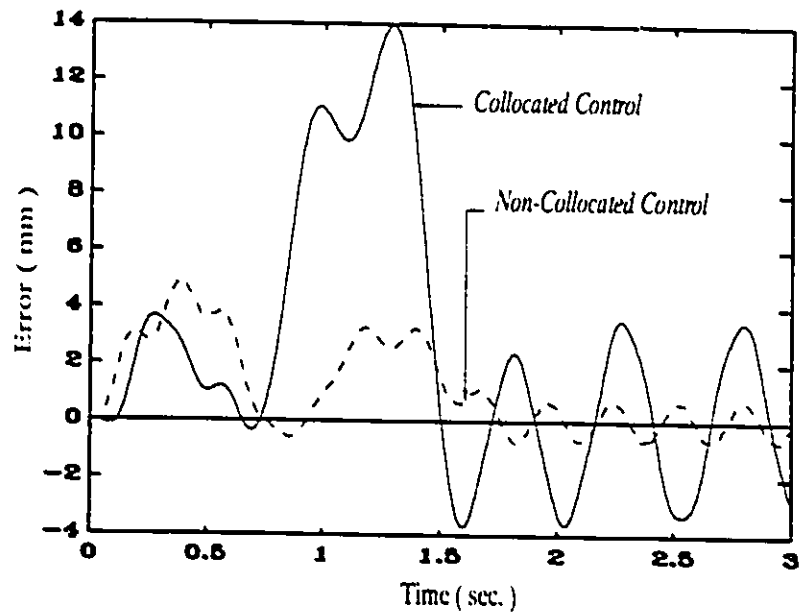


Figure 8.9: Error in the end point position along the x direction using the Conllocated Control and the Non-Collocated Control for the tracking of a straight line trajectory. $x_1/l_1=0.6$ and $x_2/l_2=0.4$.

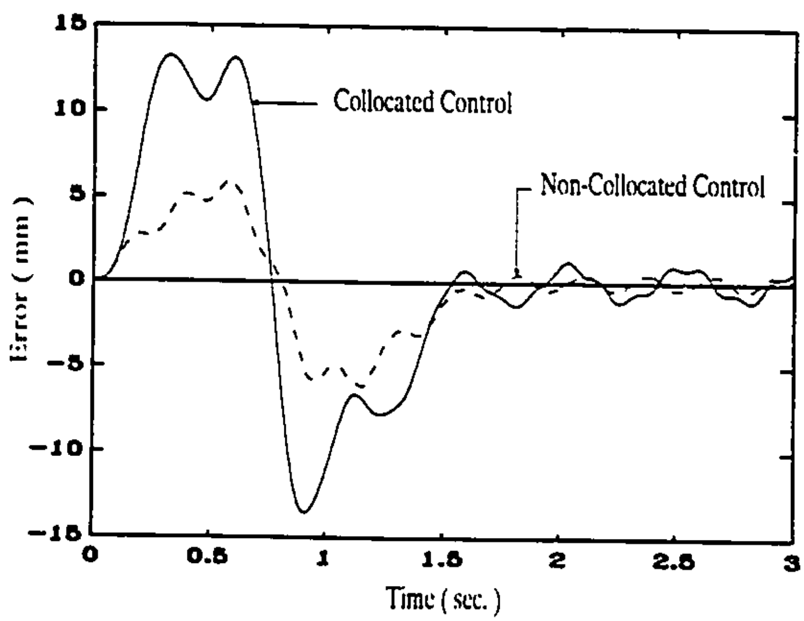


Figure 8.10: Error in the end point position along the y direction using the Collocated Control and the Non-Collocated Control for the tracking of a straight line trajectory. $x_1/l_1=0.6$ and $x_2/l_2=0.4$.

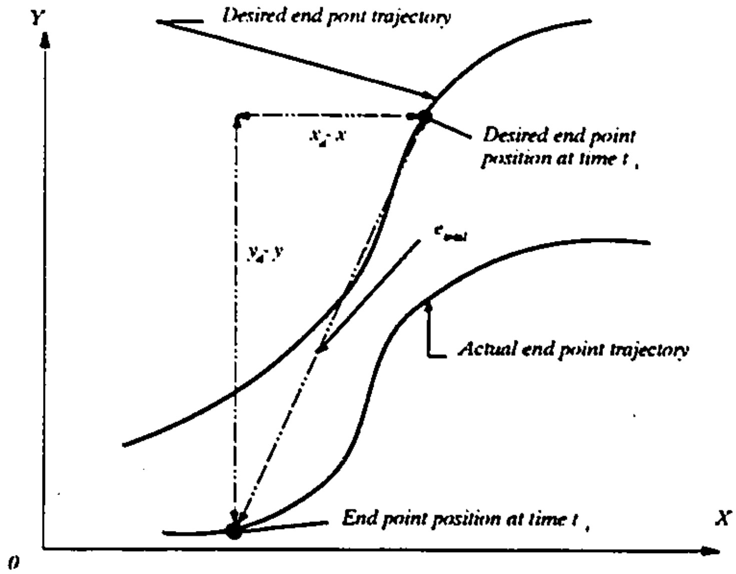


Figure 8.11: Total position error of the end point in the xy plane.

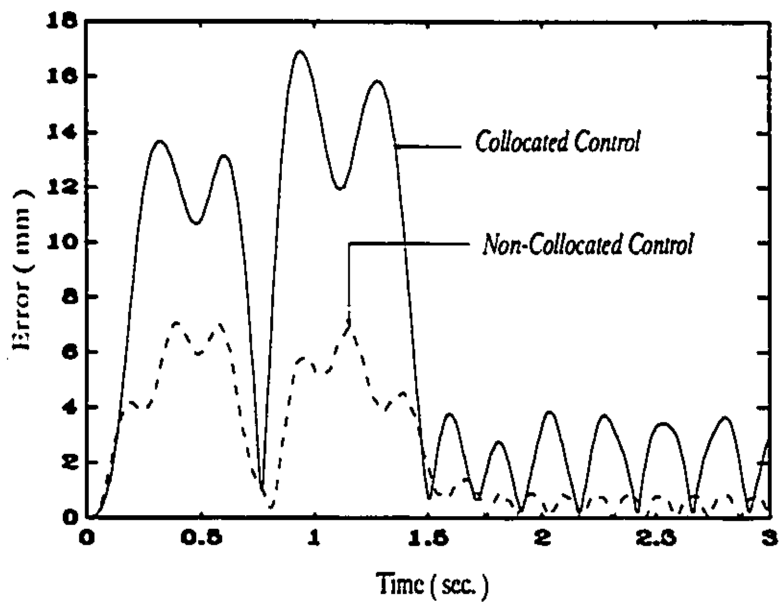


Figure 8.12: Total position error of the end point using the Collocated Control and the Non-Collocated Control for the tracking of a straight line path. $x_1/l_1=0.6$ and $x_2/l_2=0.4$.

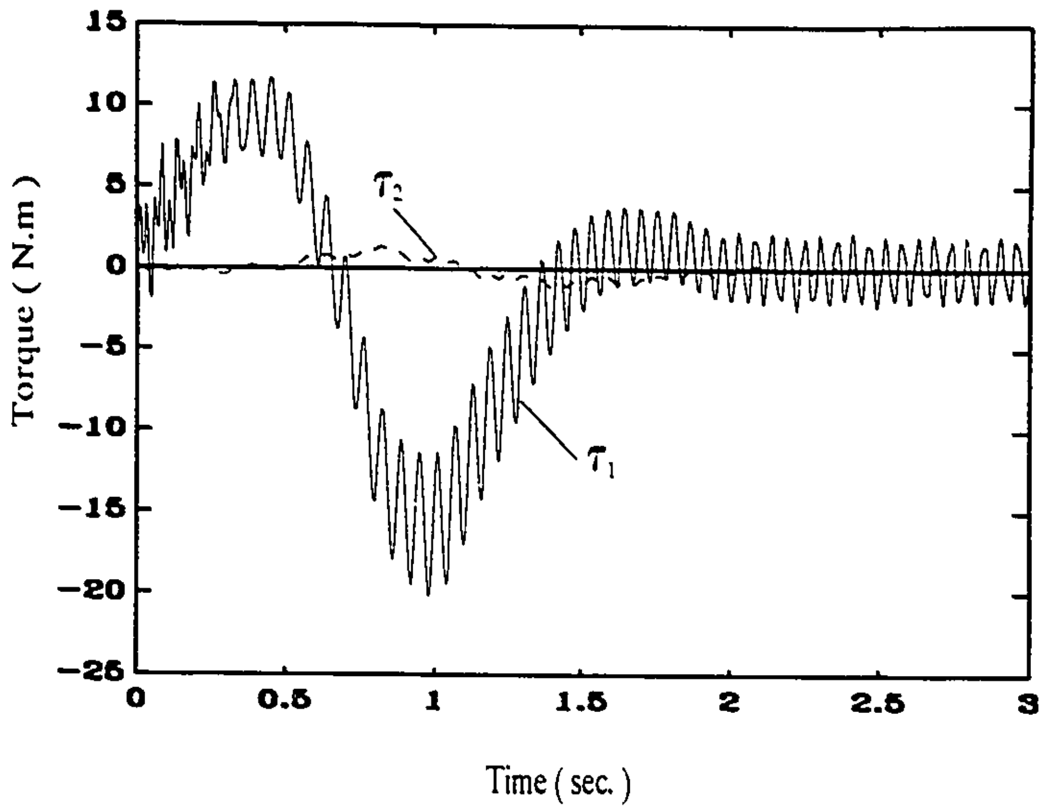


Figure 8.13: Input torques at the joints using Non-Collocated Control for the tracking of a straight line path. $x_1/l_1=0.6$ and $x_2/l_2=0.4$.

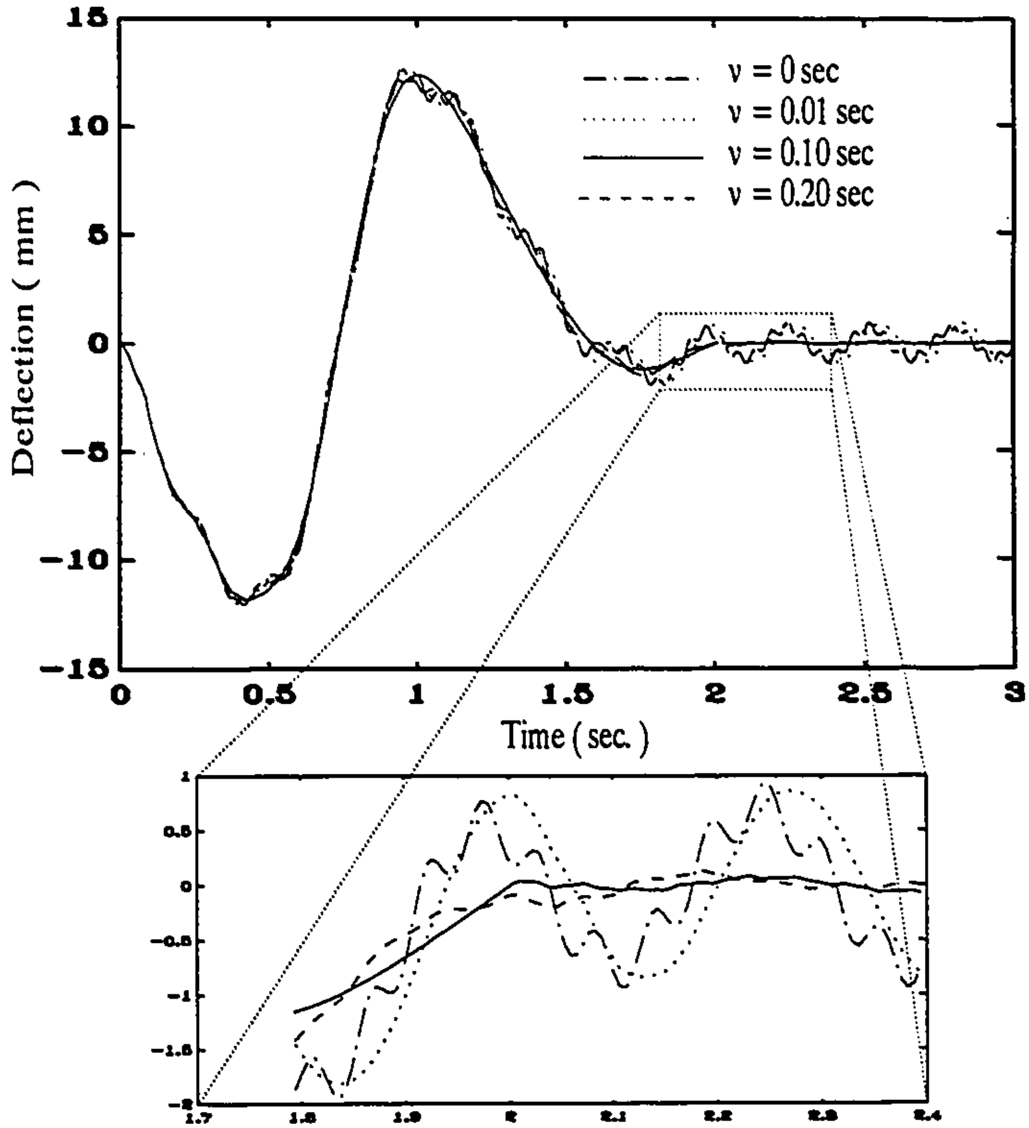


Figure 8.14: Deflection at the end of the first link with respect to its equivalent rigid link using the CTDD approach for the tracking of a straight line path. x_1/l_1=0.6 and x_2/l_2=0.4.

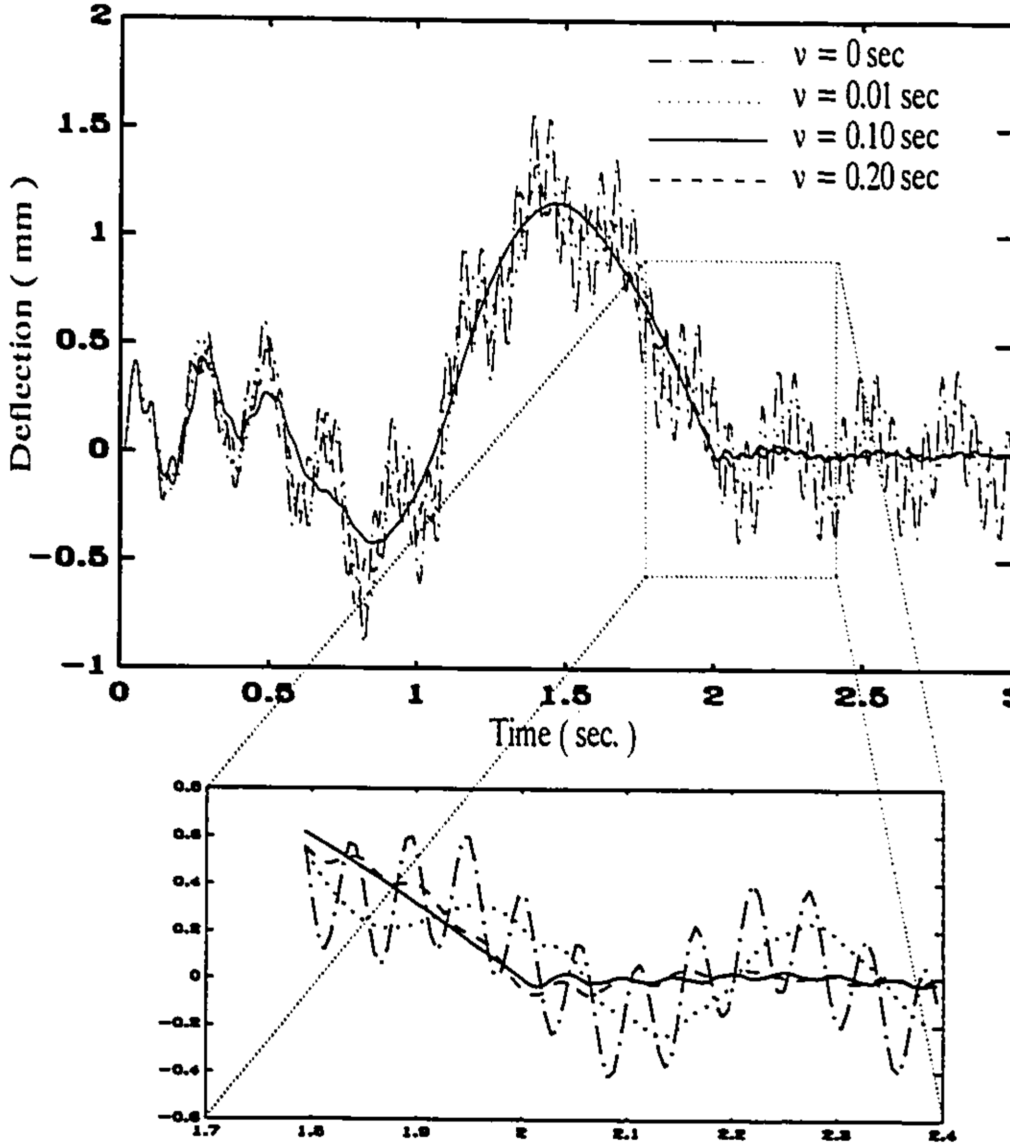


Figure 8.15: Deflection at the end of the second link with respect to its equivalent rigid link using the CTDD approach for the tracking of a straight line path. x_1/l_1=0.6 and x_2/l_2=0.4.

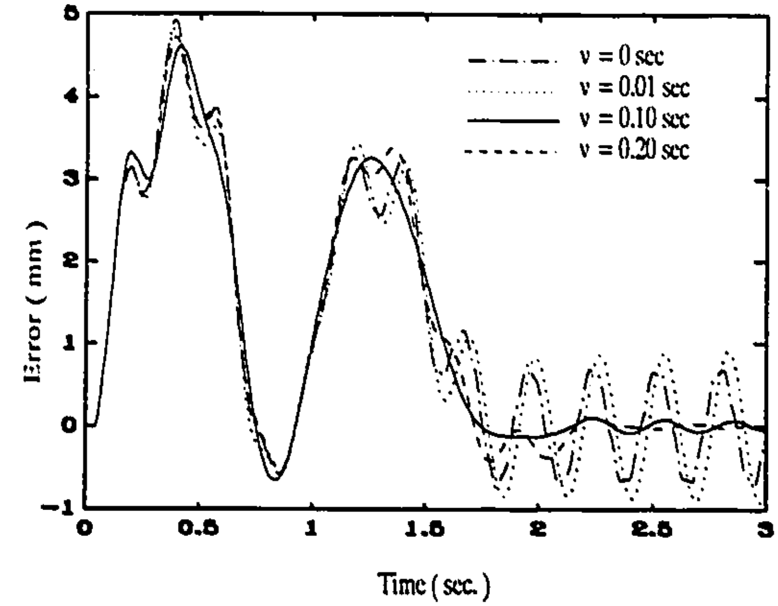


Figure 8.16: Error in the end point position along the x direction using the CTDD approach for the tracking of a straight line path. $x_1/l_1=0.6$ and $x_2/l_2=0.4$.

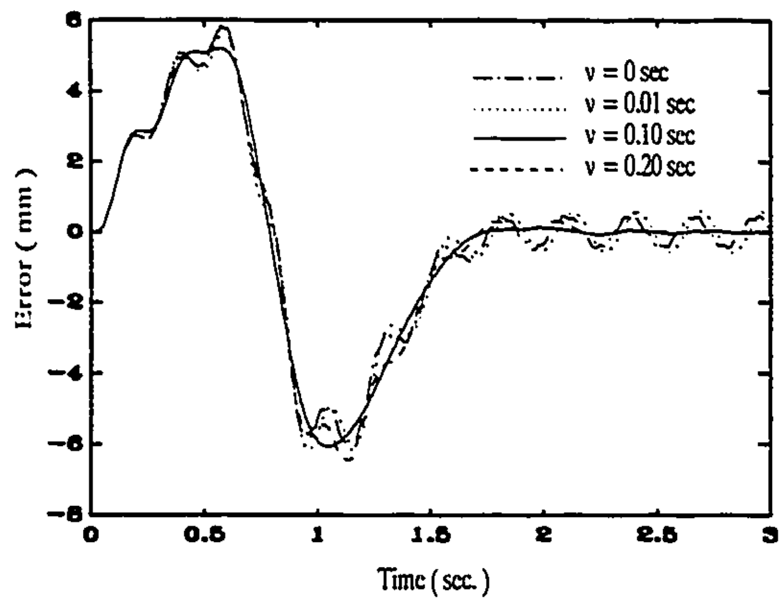


Figure 8.17: Error in the end point position along the y direction using the CTDD approach for the tracking of a straight line path. $x_1/l_1=0.6$ and $x_2/l_2=0.4$.

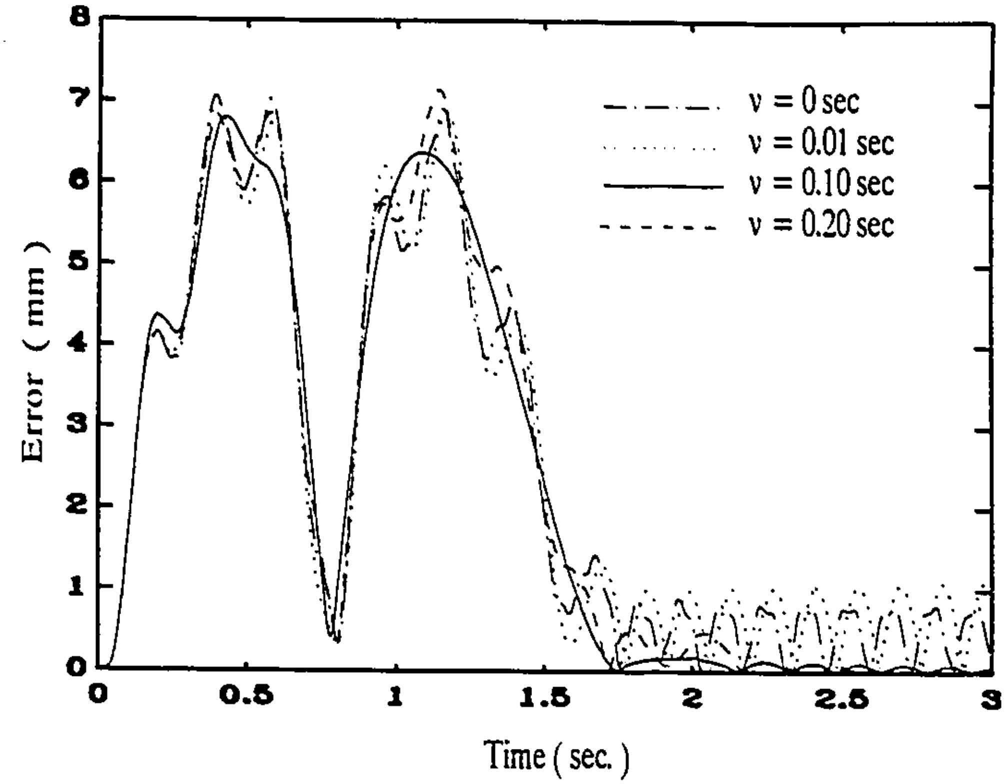


Figure 8.18: Total error in the end point position using the CTDD approach for the tracking of a straight line path. x_1/l_1=0.6 and x_2/l_2=0.4.

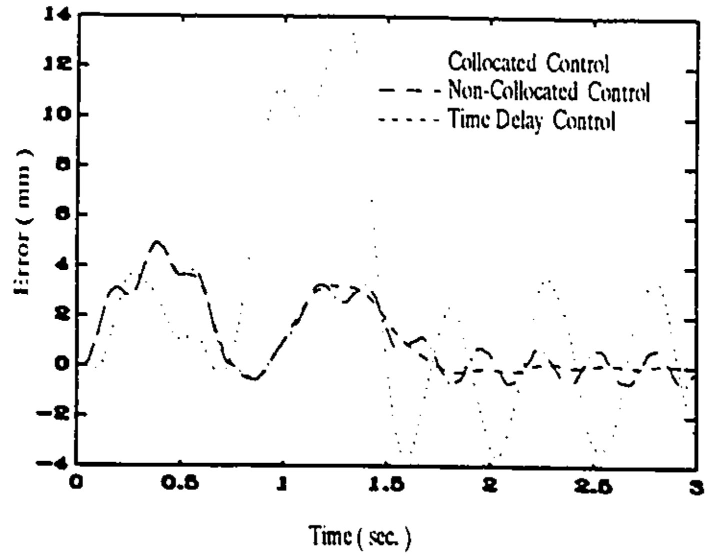


Figure 8.19: Error in the end point position along the x direction during the tracking of a straight line path using the Collocated Control, Non-Collocated Control and the CTDD with v=0.1 sec. $x_1/l_1=0.6$ and $x_2/l_2=0.4$.

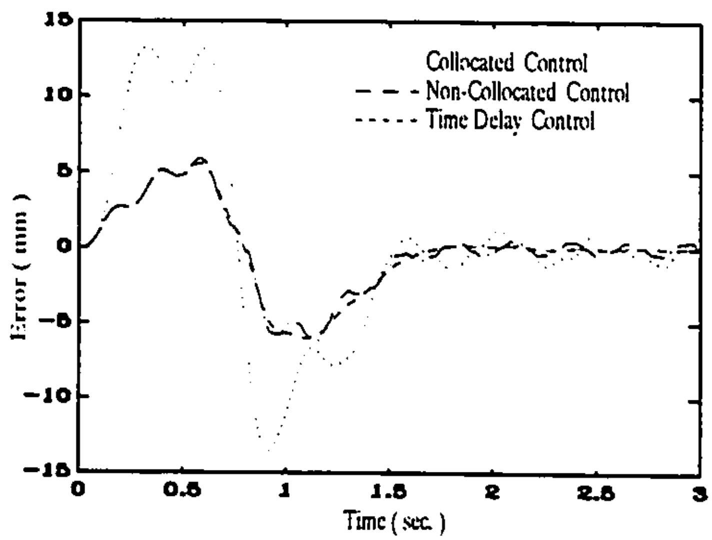


Figure 8.20: Error in the end point position along the y direction during the tracking of a straight line trajectory using the Collocated Control, Non-Collocated Control and the CTDD with v=0.1 sec. $x_1/l_1=0.6$ and $x_2/l_2=0.4$.

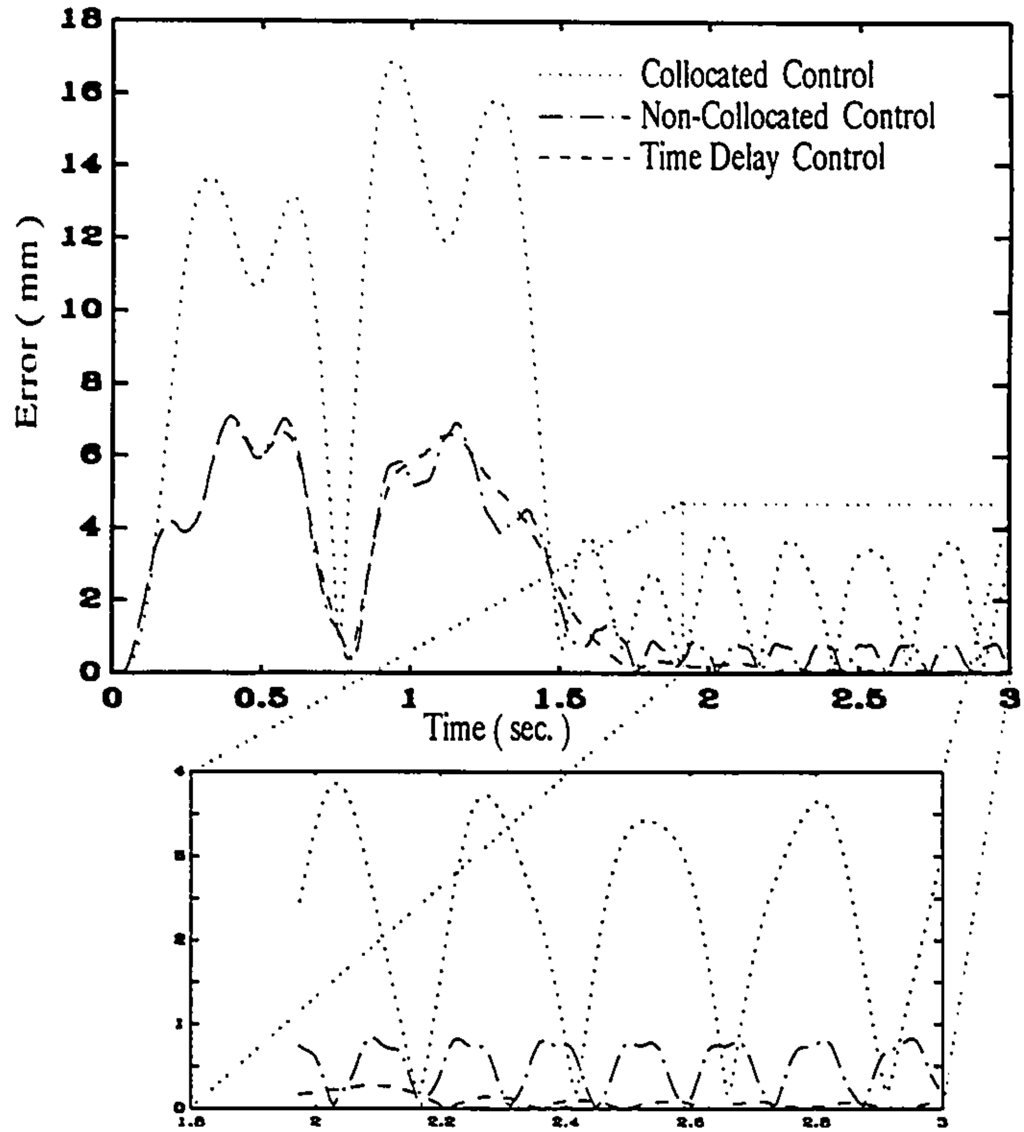


Figure 8.21: Total error in the end point position during the tracking of a straight line path using the Collocated Control, Non-Collocated Control and the CTDD with ν=0.1 sec. $x_1/l_1=0.6$ and $x_2/l_2=0.4$.

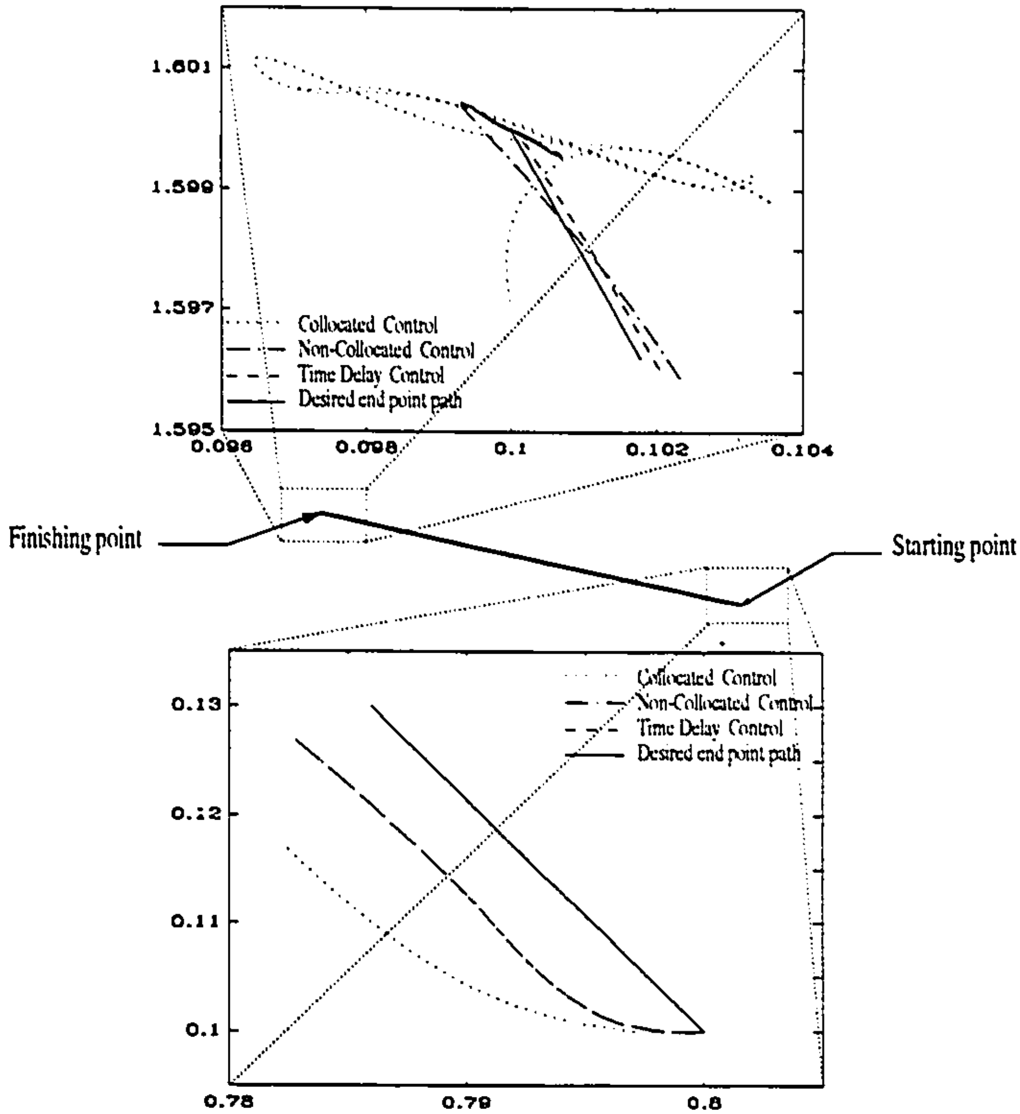


Figure 8.22: Position of the end point in the *xy* plane near the starting and the finishing points of the straight line path.

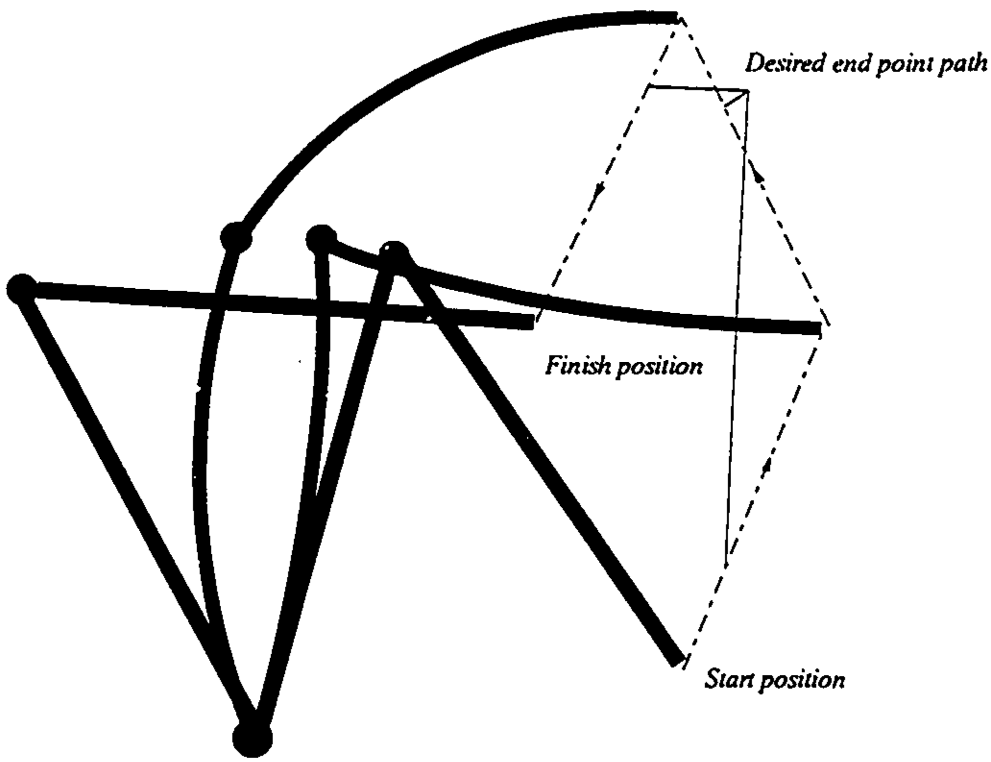


Figure 8.23: A two-link flexible manipulator perfoming a four points path.

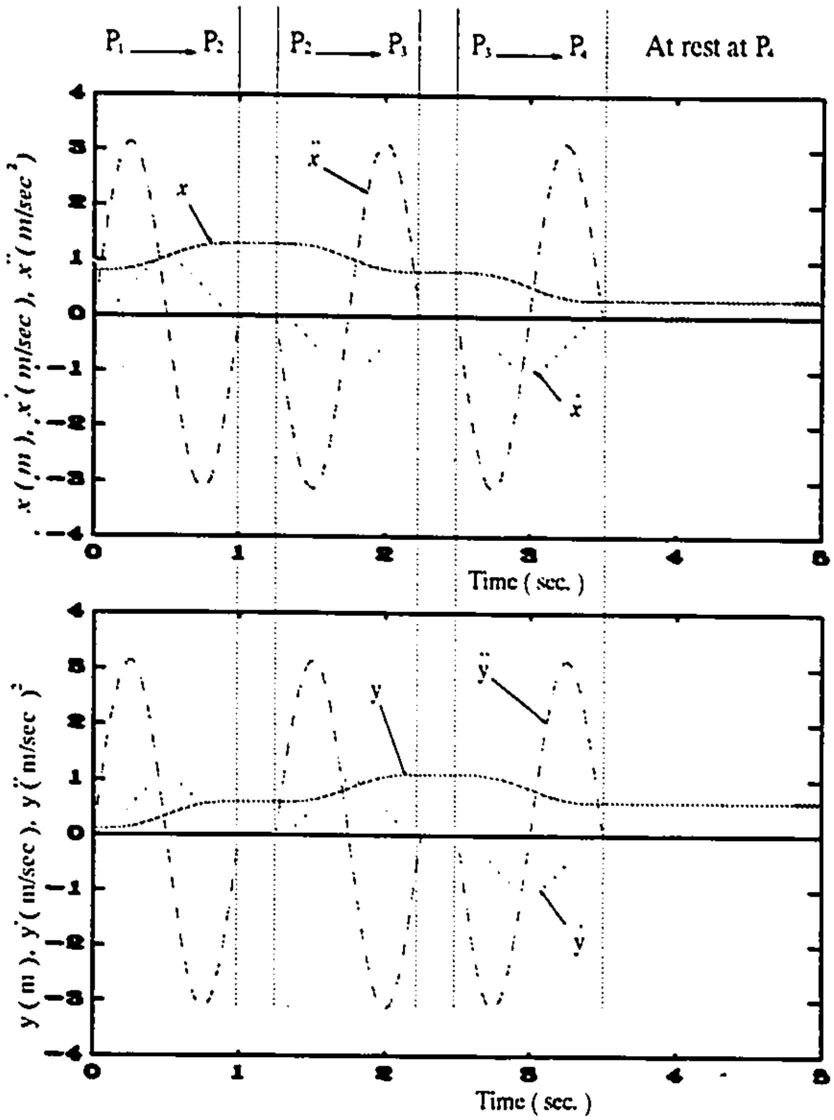


Figure 8.24: Position, velocity and acceleration representing the desired trajectory of the end point in the x and y coordinates.

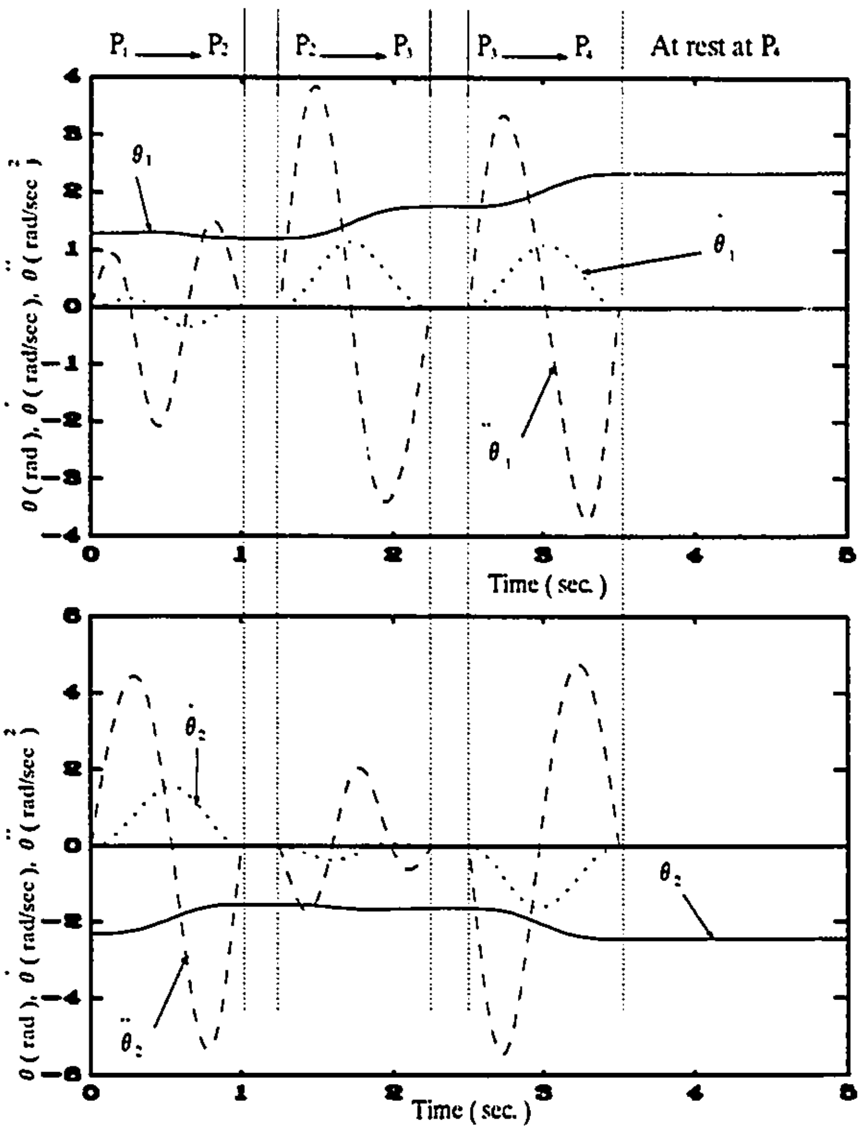


Figure 8.25: Position, velocity and acceleration representing the desired trajectory in the joint coordinates.

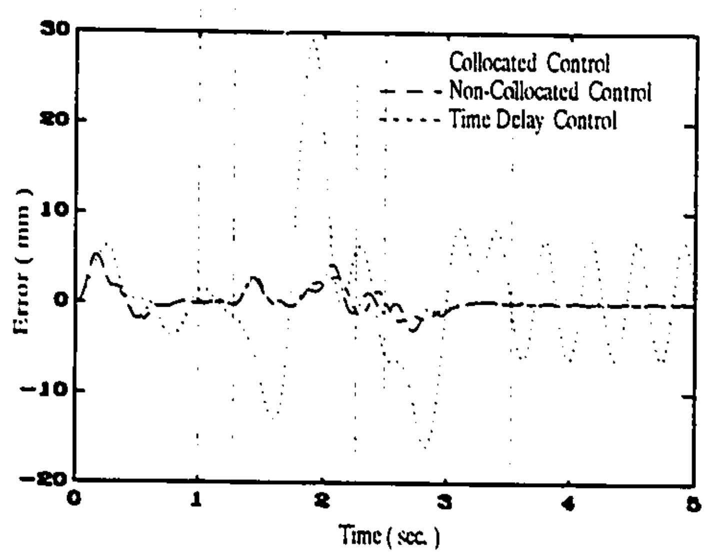


Figure 8.26: Error in the end point position along the x direction during the tracking of the four points path using the Collocated Control, Non-Collocated Control and the CTDD approach with v=0.1 sec. $x_1/l_1=0.6$ and $x_2/l_2=0.4$.

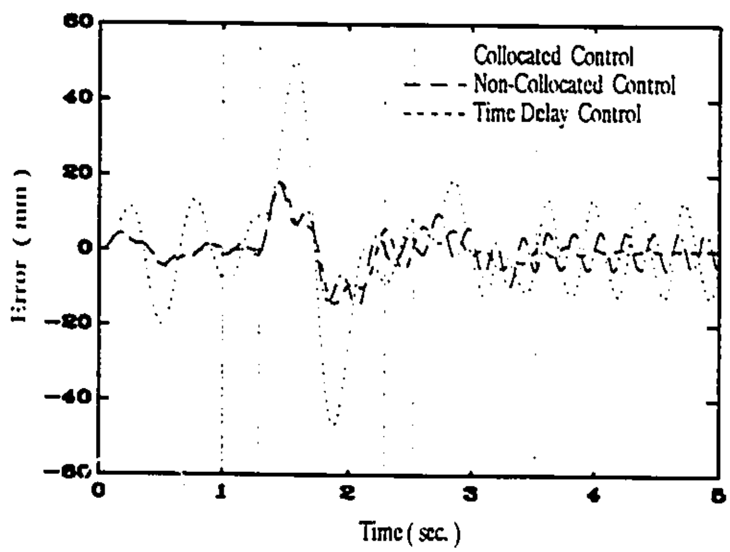


Figure 8.27: Error in the end point position along the y direction during the tracking of the four points path using the Collocated Control, Non-Collocated Control and the CTDD approach with v=0.1 sec. $x_1/l_1=0.6$ and $x_2/l_2=0.4$.

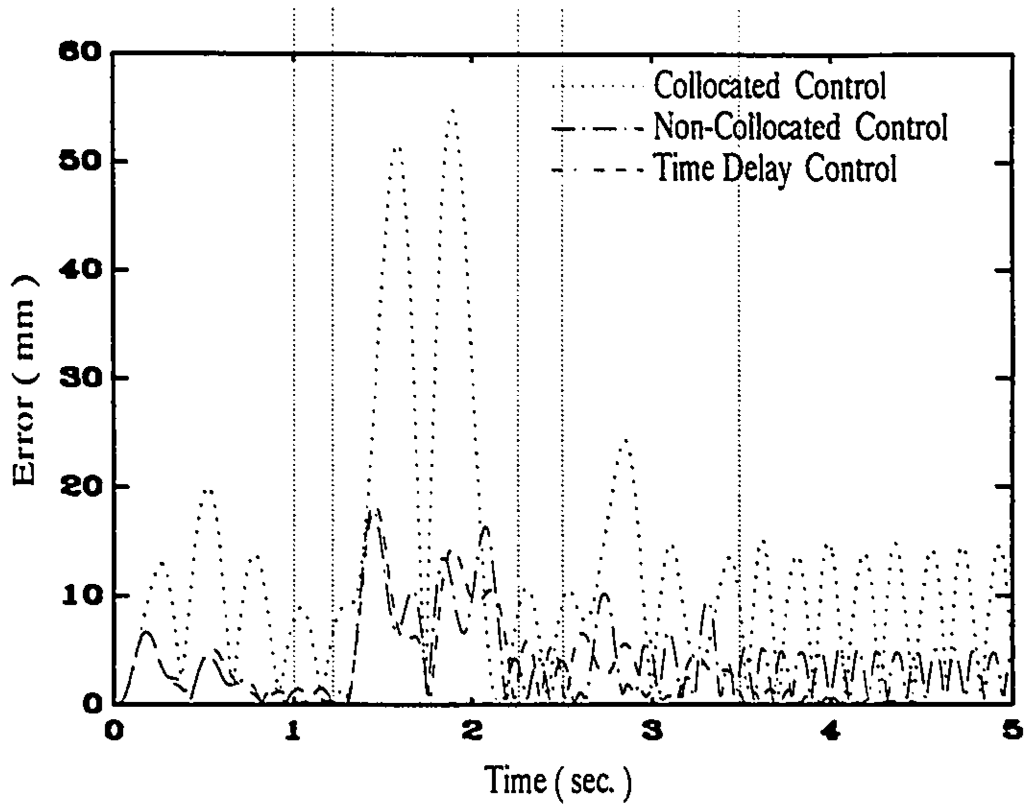


Figure 8.28: Total error in the end point position during the tracking of the four points path using the Collocated Control, Non-Collocated Control and the CTDD approach with v=0.1 sec. $x_1/l_1=0.6$ and $x_2/l_2=0.4$.

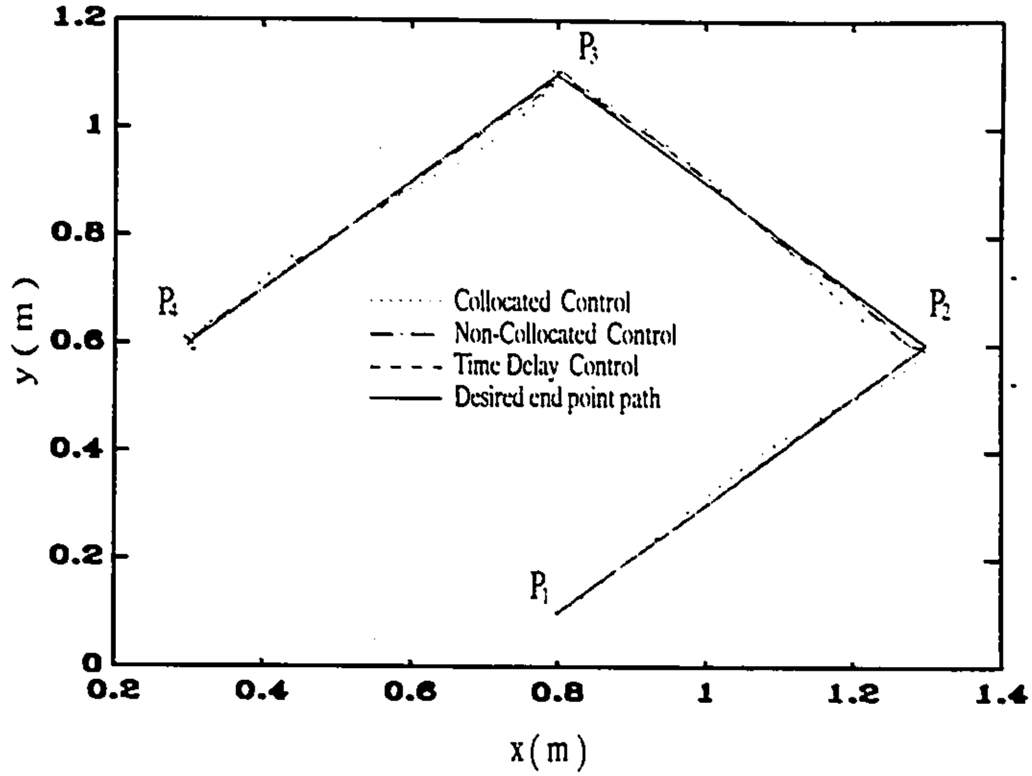


Figure 8.29: Trajectory of the end point in the xy plane during the tracking of the four points path using the Collocated Control, Non-Collocated Control and the CTDD approach with ν=0.1 sec. x_1/l_1=0.6 and x_2/l_2=0.4.

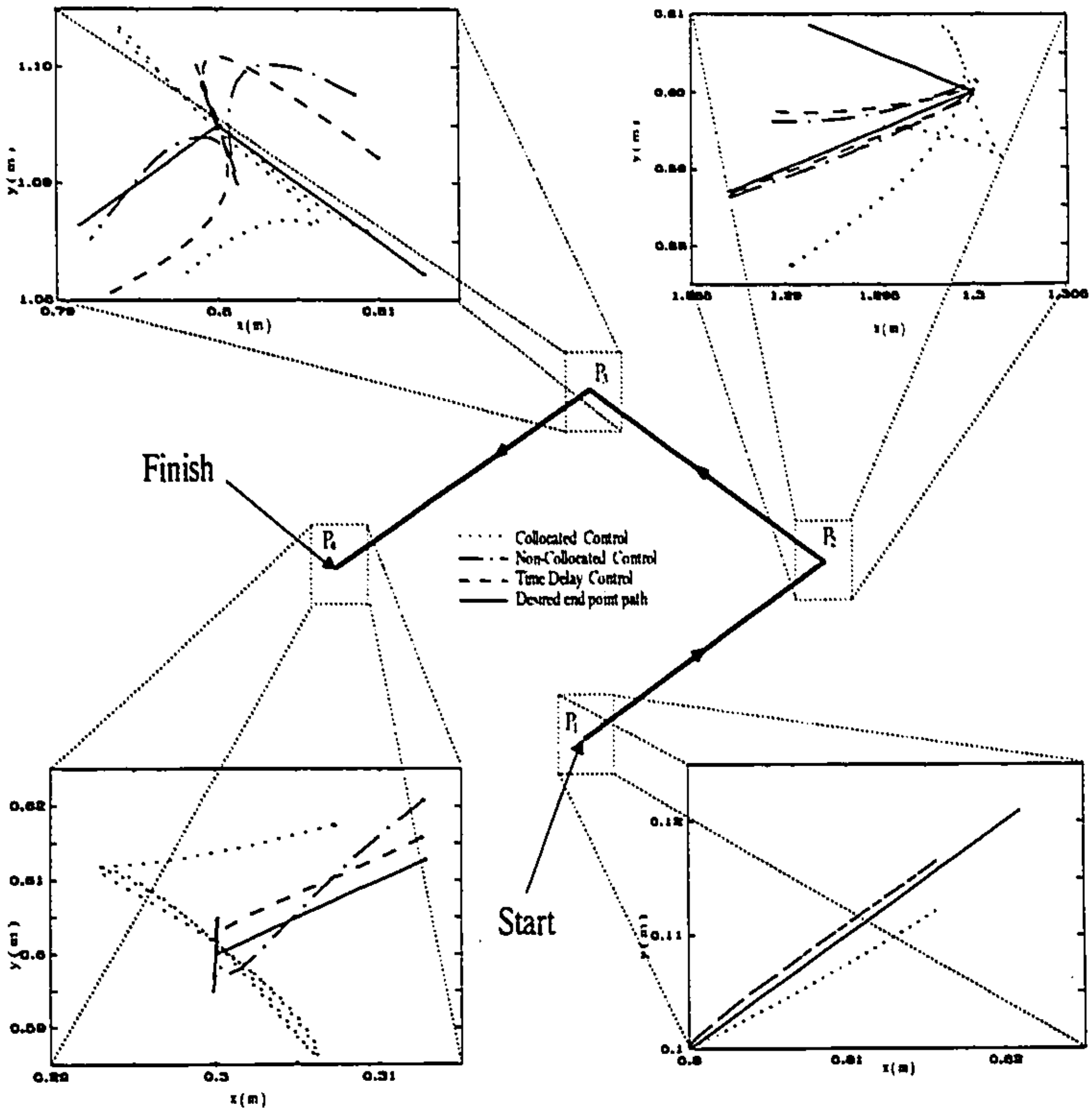


Figure 8.30: Trajectory of the end point in the xy plane near P_1, P_2, P_3, and P_4 during the tracking of the four points path using the Collocated Control, Non-Collocated Control and the CTDD approach with ν=0.1 sec. $x_1/l_1=0.6$ and $x_2/l_2=0.4$.

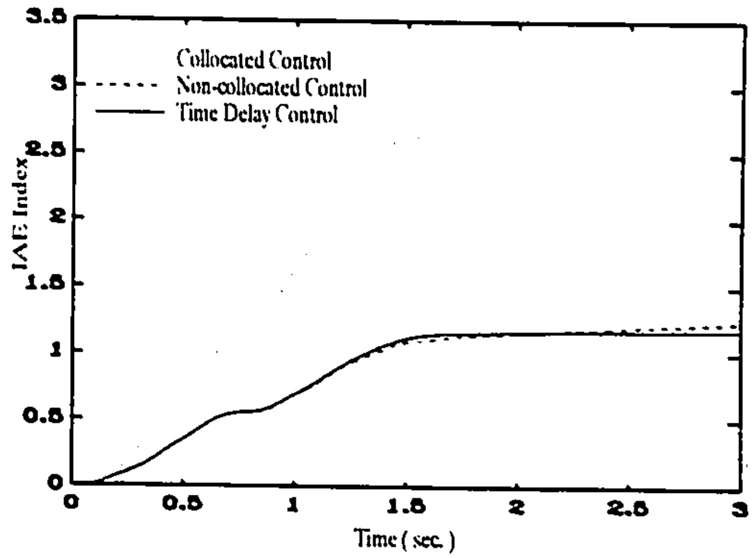


Figure 8.31: IAE performance index for the Collocated control, Non-collocated control and the CTDD approach during the tracking of the straight line path.

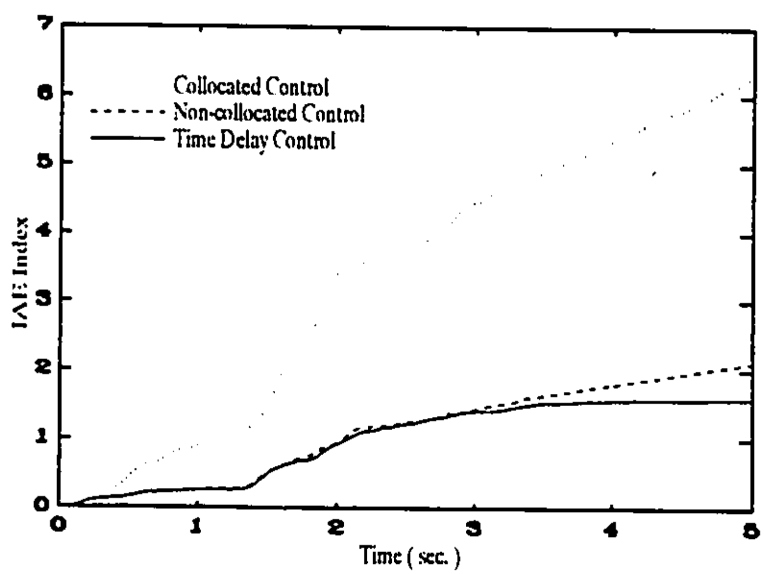


Figure 8.32: IAE performance index for the Collocated control, Non-collocated control and the CTDD approach during the tracking of the four points path.

Chapter 9

Conclusions and Recommendations

This chapter is divided into two sections. The first section is a summary of the research presented in the thesis and some of the conclusions that were drawn. The second section contains a discussion of future work to be pursued and some recommendations.

9.1 Summary and Observations

9.1.1 One link Flexible Manipulator

The first part of the thesis dealt with the control of a one link flexible manipulator. The dynamic model of the one link manipulator has been derived based on the Lagrangian formulation and the Generalized Inertia Matrix concept. Two methods are commonly used in the derivation of the dynamic model of robots with flexible links (a) the assumed modes method and (b) the finite element method. In the assumed modes method, the link flexibility is usually represented by a series expressed in terms of spatial shape functions with time varying amplitudes. The choice of the shape functions depends on the boundary conditions of the link. These functions are usually trigonometric in nature and they are defined over the entire link. In the finite element method, the cubic interpolation functions are usually used as the shape functions representing the flexibility of an element on the link. Furthermore, the functions resulting from the assumed modes method usually lead to a very complex dynamic model.

Analysis of the dynamic model has shown that when the actuator and the sensor are collocated at the joint, non-linear decoupling control such as the computed torque method does not lead to an asymptotic stability of the end point. In the absence of any structural damping, the end point keeps oscillating with constant amplitudes. These amplitudes depend on the smoothness of the command input. Four types of trajectories have been used to simulate the response of the flexible link. (a) Joint step input (b) Bang-Bang acceleration trajectory (c) Sinusoidal acceleration trajectory and (d) a trajectory with a Gaussian velocity profile. It has been found that the highest amplitude was obtained when a joint step input was used followed by the Bang-Bang acceleration trajectory. The lowest amplitude was found when the Gaussian velocity trajectory was used.

Choosing a desired trajectory with high degree of smoothness is not always possible. In practice, the system may be subjected to disturbances with very high frequency contents that can excite the higher frequencies of the flexible link. In order to increase the structural damping of the flexible link, non collocated control was introduced. It has been found that by using the angle representing the position of a point along the link as the controlled variable, the flexible link appears to be "stiffer". It was also found that when the angle representing the position of the end point is used as the controlled variable, the overall dynamic system becomes unstable.

Even though non collocated control increases the stiffness of the one link flexible manipulator, this control method fails to provide the required damping to eliminate the osciliations at the tip. In order to introduce active damping into the modes of the flexible link, a new approach referred to as the Computed Torque/Delayed Deflection approach (CTDD) has been presented. This approach was inspired by the posicast control that was introduced by O.J. Smith [94] in the late 1950's. It has been shown in the thesis that using a delayed output signal in the feedback increases the damping of the system. The CTDD approach takes advantage of the "dynamic stiffening" introduced by the non collocated control and the increased damping introduced by the delayed deflection. It was shown through simulations that the CTDD approach resulted in a very fast decay of the oscillations at the tip of the flexible link. It was also shown that the fastest decay of these oscillations can be achieved when a delay time of one half the period of the lowest frequency in the system is used.

A comparison between the CTDD approach and two other control methods has been carried out. these two methods are: A composite controller based on singular perturbation theory and a state feedback regulator. It was found that the CTDD performs much better than the state feedback regulator. Similar results were obtained when the CTDD was compared with the composite controller. The advantage of the CTDD over the composite controller is that the CTDD is much simpler and easier to implement.

9.1.2 Two Link Flexible Manipulator

In order to investigate the response of a multi-link flexible manipulator under the proposed control method, a planar two-link flexible manipulator has been used. The flexibility of each of the two links was expressed in terms of the two cubic interpolation functions used to model the one link flexible manipulator. Six variables were used to describe the motion of the two link manipulator. Two of these variables are the angles representing the joint motion of the equivalent rigid link. Unlike the one link manipulator, the reference commands for the two link manipulator are given in Cartesian space. In order to obtain the joint angles corresponding to the desired end point position in Cartesian space, the inverse kinematics of a rigid two link manipulator was carried out. Two test trajectories have been used to compare the tracking performances of the two link flexible manipulator. These two trajectories are (a) Straight line trajectory and (b) Four points trajectory. In the straight line trajectory, the end point is commanded to move between two points following a straight path. In the four points trajectory, the end point is commanded to move from an initial point to a final point passing by two intermediate points. The path between two consecutive points is represented by a straight line.

Among the collocated control methods used in the thesis is the independent joint PD controller which is similar to the controllers used on the Shuttle Remote Manipulator System. It is recommended that when this type of control is used to control flexible link manipulators, the joint angles have to be moved slowly to avoid large amplitude oscillations at the end point. These errors are due to the fact that no interaction between the joints is taken into account by the independent joint PD controller and no information from the end point sensor is used to compute the torques of the actuators. Unlike the independent joint PD control, the collocated control based on the computed torque method takes into account the interraction between the joint that is being controlled and the other joints. Even though the computed torque control has improved the performance of the two link flexible manipulator, this type of controller still has one major draw back which consists of the fact the end position - the specific quantity to be controlled - is not used in the feedback. The controlled quantities used in the collocated computed torque controller are the joint angles.

The non collocated control overcomes the limitations of the independent joint PD and the computed torque controllers by using the deflection of a point along the link in the feedback. The main difficulty in dealing with the non collocated control of the two link flexible manipulator is in the selection of the points at which the deflections are to be used in the feedback. Two points, one on each link, have been selected so that the overall dynamic system is stable. These points are $\frac{x_1}{l_1}=0.6$ and $\frac{x_2}{l_2}=0.4$. During the tracking of a straight line path in the xy plane, the total deviation of the end point from the desired path has been reduced from 16.91 mm for the computed torque control to 7.06 mm for the non collocated control. For the four points path, the total deviation of the end point has been reduced from 54.86 mm for the computed torque control to 18.09 mm for the non collocated control. This improvement in the tracking performance is a result of the dynamic stiffening introduced by the non collocated control. Even though non collocated control has improved the tracking performance of the two link flexible manipulator over that of the computed torque control, in the absence of structural damping, neither of these two control methods leads to an asymptotic stability of the end point. The CTDD approach was found to give satisfactory results in dampening the end point oscillations and in achieving asymptotic stability of the end point.

9.2 Recommendations and Future Work

The manipulator's configuration studied in this dissertation is one in which the motion occurs in the horizontal plane. To further improve our understanding of the behavior of manipulators with flexible links, other configurations need to be investigated. An example of these configurations is a multi-link flexible manipulator with a mobile base. Such a base may be a mobile space station or a space shuttle like the one carrying the CANADARM. Another case of special interest is that of multiple flexible link manipulators used to handle large objects. When these manipulators interact with each other, the overall dynamic system changes significantly.

Finally, to see the effectiveness of the proposed control method in reducing the position error of the end point of flexible link manipulators, experiments need to conducted in order to validate the assumptions made throughout this work. One of the main assumptions used in this work is that the deflection of each link occurs only in the horizontal plane. This can be achieved by choosing the cross section of each link to be relatively stiff in the vertical plane. To avoid torsional deformation, the manipulator can be supported by a horizontal table with air cushion in order to reduce the friction between the links and the surface of the table. Actuators and sensors play a major role in the implementation of control system. In general, the dynamics of these actuators and sensors need to be incorporated with the dynamics of the plant. In the case of a plant with complex dynamics, such as flexible link manipulators, sophisticated actuators and sensors are required. Among the actuators that can be used in the control of flexible link manipulators are the brushless direct drive motors. These motors do not require commutation and have very low friction. Because they are direct drives, these motors have no backlash or any other undesirable properties introduced by gear trains. Sensors in a flexible link manipulator can be located either on the links themselves or externally to monitor the position of the end point.

References

[1]: T. E. Alberts, H. Xia, Y. Chen, " Dynamic Analysis to Evaluate Viscoelastic Passive Damping Augmentation for the Space Shuttle Remote Manipulator System", Journal of Dynamic systems, Measurement, and Control, Vol. 114, pp. 468-475, September 1992.

[2]: H. Asada, Z-D MA, H. Tokumara, "Inverse Dymanics of Flexible Robot Arms: Modelling and Computation for Trajectory Control", Journal of Dynamic Systems, Measurement, and Control, Vol. 112, pp. 177-185, June 1990.

[3]: M. J. Balas, "Feedback Control of Flexible Systems", IEEE Trans. on Automatic Control, Vol. AC-23, No. 4, August 1978.

[4]: H. Baruh, S. S. K. Tadikonda," Issues in the Dynamics and Control of Flexible Robot Manipulators", AIAA Journal of Guidance, Vol. 12, No. 5, Sept.-Oct. 1989.

[5]: E. Bayo, "Computed Torque For The Position Control of Open Chain Flexible Robots", IEEE International Conference on Robotics and Automation, Phil. 1988, pp. 316-321.

[6]: E. Bayo, B. Paden, "On Trajectory Generation for Flexible Robots", Journal of Robotic Systems, 4 (2), 1987.

[7]: M. Benati, A Morro, " Dynamics of Chain of Flexible Links", Journal of Dynamic Systems, Measurement, and Control, Vol. 110, Dec. 1988, pp.410-415.

[8]: M. Benati, A Morro, " Formulation of Equations of Motion for a chain of Flexible links Using Hamilton's Principle", Journal of Dynamic Systems, Measurement, and Control, Vol. 116, March 1994, pp. 61-88.

[9]: W. Beres, Jerzy Z. Sasiadek, " Finite Element Dynamic Model of Multilink Flexible Manipulators", Appl. Math. and Comp. Sci., Vol. 5, No. 2, 1995, pp. 231-262.

[10]: S. K. Biswas, R. D. Klafter," Dynamic Modelling And Optimal Control of Flexible Robotic Manipulators", IEEE International Conference on Robotics and Automation, Phil. 1988, pp 15-20.

[11]: J. Bontsema, R. F. Curtain, J. M. Schumacher, " Robust Control of Flexible Structures: A Case Study", Automatica, Vol. 24, No. 2, pp. 177-186, 1988.

[12]: W. J. Book, " Recursive Lagrangian Dynamics of flexible Manipuiator Arms", The International Journal of Robotics Research, Vol. 3, No. 3, Fall 1984.

[13]: W. J. Book, O. Maizza-Neto, and D. E. Whitney, "Feedback Control of Two Beam, Two Joint Systems with Distributed Flexibility" Journal of Dynamic Systems, Measurement, and Control, December 1975, pp. 424-431.

[14]: K. W. Buffinton, " Dynamics of Elastic Manipulators With Prismatic Joints" Journal of Dynamic systems, Measurement, and Control, Vol. 114, pp. 41-49, March 1992.

[15]: Robert H. Cannon, Jr., Eric Schmitz, " Initial Experiments on The End-Point Control of a Flexible One-Link Robot", The International Journal of Robotics Research, Vol. 3, No. 3, Fall 1984.

[16]: A. Castelazo and H. Lee, "Nonlinear Compensation for Flexible Manipulators", Journal of Dynamic Systems, Measurement, and Control, Vol. 112, March 1990, pp. 62-68.

[17]: L. W. Chang, "Dynamics and Control of Vertical-plane Motion for an electro-hydraulically Actuated Single-Flexible-Link Arm", Journal of Dynamic systems, Measurement, and Control, Vol. 114, pp. 89-95, March 1992.

[18]: L. W. Chang and J. F. Hamilton, "The Kinematics of Robotic Manipulators With Flexible Links Using an Equivalent Rigid Link System (ERLS) Model", Journal of Dynamic Systems, Measurement, and Control, Vol. 113, March 1991, pp. 48-53.

[19]: L. W. Chang and J. F. Hamilton, "Dynamics of Robotic Manipulators With Flexible Links", Journal of Dynamic Systems, Measurement, and Control, Vol. 113, March 1991, pp. 54-53.

[20]: Y. P. Chen, K. S. Yeung, " Sliding Mode Control of Multi-link Flexible Manipulators" International Journal of Control, Vol. 54, No. 2, pp. 257-278, 1990.

[21]: S. Choura, S. Jayasuriya, and M. A. Medick,"On the Modeling, and Open-Loop

Control of a Rotating Thin Flexible Beam", Journal of Dynamic Systems, Measurement, and Control, Vol. 113, March 1991, pp. 26-33.

[22]: Dong H. Chyung,"Tracking Controller for Linear Systems With Time Delays in Both State and Control Variables", Journal of Dynamic Systems, Measurement, and Control, Vol. 115, March 1993, pp. 179-183.

[23]: A. De Luca, B. Siciliano," Trajectory Control of a nonlinear One-Link Flexible Arm", Int. J. Control, Vol. 50, No. 5,pp. 1699-1715, 1989.

[24]: A. De Luca, B. Siciliano," Explicit Dynamic Modeling of a Planar Two-link Flexible Manipulator", Proceedings of the 29th Conference on Decision and Control, Honolulu, Hawaii, December 1990, pp. 528-530.

[25]: F. Demeester and H. Van Brussel, "Real time optical Measurement of Robot Structure Deflections", Mechatronics: Mechanics, Electronics, Control, Int. Journal, Vol. 1, No.1, pp. 73-86, 1991, Pergman Press.

[26]: X. Ding, T. J. Tarn, A. K. Bejczy, "A Novel Approach to the Modelling and Control of Flexible Robot Arms", Proceedings of the 27th Conference on Decision and Control, Austin, Texas, December 1988, pp. 52-57.

[27]: G. R. Eisler, R. D. Robinett, D. J. Segalman, J. D. Feddema, "Approximate Optimal Trajectories For Flexible-Link Manipulator Slewing Using Recursive Quadratic Programming", Journal of Dynamic systems, Measurement, and Control, Vol. 115, pp. 405-410, September 1993.

[28]: A. H. von Flotow, B. Schäfer, "Wave-Absorbing Controllers for a Flexible Beam", AIAA Journal of Guidance, Vol. 9, No. 6, Nov.-Dec., 1986, pp. 673-680.

[29]: V. Feliu, K. S. Rattan, H. B. Brown, Jr, " Modeling and Control of Singl-Link Flexible Arms With Lumped Masses" Journal of Dynamic systems, Measurement, and Control, Vol. 114, pp. 59-69, March 1992.

[30]: Phillipe E. Gaultier, and William L. Cleghorn, " A Spatially Translating and Rotating Beam Finite Element For Modeling Flexible Manipulators", Journal of Mech. and Mach. Theory, Vol. 27, No. 4, pp. 415-433, 1992.

[31]: T. C. Hsia, L. S. Gao, " Robot Manipulator Control Using Decentralized Linear Time-Invariant Time Delayed Joint Controllers", Proceedings of the IEEE International Conference on Robotics and Automation, 1990, pp.2070-2075.

[32]: W. C. Hsu, A. A. Shabana, " Passive and Active Inertia Forces in Flexible Body Dynamics", Journal of Dynamic systems, Measurement, and Control, Vol. 114, pp.

571-579, December 1992.

[33]: Y. Huang, C. S. G. Lee, " Generalization of Newton-Euler Formulation of Dynamic Equations to Nonrigid Manipulators", Journal of Dynamic Systems, Measurement, and Control, Vol. 110, sept. 1988, pp.308-315.

[34]: Ben Jonker, " A Finite Element Dynamic Analysis of Flexible Manipulators", The International Journal of Robotics Research, Vol. 9, No. 4, August 1990.

[35]: Parveen Kalra and Annad M. Sharan, "Accurate Modelling of Flexible Manipulators Using Finite Element Analysis", Journal of Mechanisms and Machine Theory, Vol. 26, No. 3, 1991, pp.299-313.

[36]: M. S. Kang, B. Yang, " Discrete Time Noncollocated Control of Flexible Mechanical Systems Using Time Delay", Journal of Dynamic systems, Measurement, and Control, Vol. 116, pp. 216-222, June 1994.

[37]: F. Khorrami, "Dynamical Properties of Manipulators Exhibiting Flexibilities", Proceedings of the IEEE International Conference on Systems Engineering, Pittsburgh, PA, Aug. 1990, pp. 1-4.

[38]: F. Khorrami, Ümit Ozgüner," Perturbation Methods in Control of Flexible Link Manipulators", IEEE International Conference on Robotics and Automation, Phil. 1988, pp. 310- 315.

[39]: F. Khorrami, S. Zheng, " An Inner/Outer Loop Controller for Rigid-flexible Manipulators", Journal of Dynamic systems, Measurement, and Control, Vol. 114, pp. 580-587, December 1992.

[40]: F. Korolov, Y. H. Chen, " Controller Design Robust to Frequency Variation in a One-Link Flexible Robot Arm", Journal of Dynamic Systems, Measurement, and Control, Vol. 111, pp. 9-14, March 1989.

[41]: P. T. Kotnik, S. Yorkovich, Ümit Ozgüner," Acceleration Feedback Control For a Flexible Manipulator Arm", IEEE International Conference on Robotics and Automation, Phil. 1988, pp. 322-323.

[42]: H. Krishnan and M. Vidyasagar, "Control of a Single-Link Flexible beam using a Hankel-Norm based Reduced order Model", International Conference on Robotics and Automation, Phil. 1988, pp. 9-14.

[43]: Dong-Soo Kwon, W. J. Book, " A Time-Domain Inverse Dynamic Tracking Control of a Single-Link Flexible Manipulator", Journal of Dynamic systems, Measurement, and Control, Vol. 116, pp. 193-200, June 1994.

[44]: S. S. Lim, and N. U. Ahmed, "Modeling and Control of Flexible Space Stations", Int. Journal of Dynamics and Control, Vol. 2, pp. 5-33, 1992.

[45]: K. H. Low, M. Vidyasagar, " A Lagrangian Formulation of the Dynamic Model for Flexible Manipulator Systems", ASME Journal of Dynamic Systems, Measurement, and Control, Vol. 110, June 1988, pp.175-181.

[46]: P. Lucibello, "Non Linear Regulation With Internal Stability of a Two Link flexible Robot Arm", Proceedings of the 28th Conference on Decision and Control, Tempa, Florida, December 1989.

[47]: P. Lucibello, M. D. Di Benedetto, "Output Tracking for a Non Linear Flexible Arm", Journal of Dynamic systems, Measurement, and Control, Vol. 115, pp. 78-85, March 1993.

[48]: P. H. Meckel, W. P. Seering, "Experimental Evaluation of Shaped Inputs to Reduce Vibration for a Cartesian Robot", Journal of Dynamic systems, Measurement, and Control, Vol. 112, pp. 159-165, June 1990.

[49]: L. Meirovitch, " Elements of Vibration Analysis",McGraw-Hill Book Company, New York, 1986.

[50]: P. J. Nathan, S. N. Singh, "Sliding Mode Control and Elastic Mode Stabilization of a Robotic Arm With Flexible Links", Journal of Dynamic Systems, Mesurement, and Control, December 1991, Vol. 113, pp. 669-676.

[51]: S. Nicosia, P. Tomei and Tornambe, " Non-linear control and observation algorithms for a single-link flexible arm", Int. J. Control, Vol. 49, No. 3, pp. 827-840, 1989.

[52]: Oakley Celia Marie, "Experiments in modelling and End-point Control of Two-Link Flexible Manipulators", Ph.D thesis, Stanford University, Department of Mechanical Engineering, Stanford, CA 94305, April 1991.

[53]: B. Paden, D. Chen, R. Ledesma, E Bayo, "Exponentially Stable Tracking Control for Multijoint Flexible-Link Manipulators", Journal of Dynamic systems, Measurement, and Control, Vol. 115, pp. 53-59, March 1993.

[54]: T. R. Parks, H. A. Pak, " Effect of payload on the Dynamics of a Flexible Manipulator-Modeling for Control" Journal of Dynamic systems, Measurement, and Control, Vol. 113, pp. 409-418, September 1991.

[55]: R. P. Petroka and Liang-Wey Chang, "Experimental Validation of a Dynamic Model (Equivalent Rigid Link System) on a Single-Link Flexible Manipulator",

Journal of Dynamic Systems, Measurement, and Control, Vol. 111, December 1989, pp. 667-673.

[56]: F. Pfeiffer, B. Gebler," A Multistage Approach to the Dynamics and Control of Elastic Robots", IEEE International Conference on Robotics and Automation, Phil. 1988, pp 2-8

[57]: S. S. Rao, T. S. Pan, and V. B. Venkayya, " Modeling, Control, and design of structures: A survey", Journal of Appl. Mech. Rev., May 1990, Vol. 43, No. 5, pp 99-117.

[58]: Jerzy Z. Sasiadek, R. Srinivasan, " Dynamic Modelling and Adaptive Control of a Single-Link Flexible Manipulator", AIAA Guidance, Navigation, and Control Conference, Minneapolis, MN, Aug. 15-17, 1988.

[59]: Jerzy Z. Sasiadek, " Space Robotics and Manipulators - The Past and the Future", Control Engineering Practice, Vol. 2, No. 3, pp 491 - 497, 1994.

[60]: A. A. Shabana, "Dynamics of Flexible Bodies Using Gneralized Newton-Euler Equations", Journal of Dynamic Systems, Measurement, and Control, Vol. 112, September 1990, pp. 496-503.

[61]: A. Shchuka, A. A. Goldenberg, " Tip Control of a Single-Link Flexible Arm Using A Feedforward technique", Mech. Mach., Theory, Vol 24, No. 5, pp. 439-455, 1989

[62]: B. Siciliano, W. J. Book, " A Singular Perturbation Approach to Control of Light weight Flexible Manipulators", The international Journal of Robotics Research, Vol. 7, No. 4, August 1988.

[63]: B. Siciliano, J. V. R. Prasad, A. J. Calise, " Output Feedback Two-Time Scale Control of Multilink Flexible Arms" Journal of Dynamic systems, Measurement, and Control, Vol. 114, pp. 70-77, March 1992.

[64]: N. C. Singer, W. P. Seering, "Preshaping Command Inputs to Reduce System Vibration", Journal of Dynamic systems, Measurement, and Control, Vol. 112, pp. 76-82, March 1990.

[65]: N. C. Singer, "Residual Vibration Reduction in Computer Controlled Machines", PhD thesis, Department of Mechanical Engineering, MIT, Fall, 1988.

[66]: S. N. Singh, "Control and Stabilization of a Nonlinear Uncertain Elastic Robotic Arm", IEEE Trans. on Aerospace and Electronic Systems, Vol. 24, No. 2, March 1988.

[67]: S. N. Singh. A. A. Schy," Control of Elastic Robotic Systems by Nonlinear Inversion and Modal Damping". ASME Trans.. Journal of Dynamic Systems. Measurment. and Control. Vol. 108. pp. 180-189. Sept. 1986.

[68]: S. N. Singh. A. A. Schy. " Elastic Robot Control: Non-linear Inversion and Linear Stabilization". IEEE Trans. on Aerospace and Electronic Systems. Vol. AES-22. No. 4. July 1986.

[69]: T. Singh. R. R. Vadali." Robust Time Delay Control". Journal of Dynamic Systems. Measurment. and Control. Vol. 115. pp. 303-310. June 1993.

[70]: J. M. Skowronski." Algorithms for Adaptive Control of Two-Arm Flexible Manipulators Under Uncertainty".IEEE Trans. on Aerospace and Electronic Systems. Vol. 24. No. 5. Sept. 1988.

[71]: J. M. Skowronski. "Control of Nonlinear Mechanical Systems". Plenum Press. New York. 1991.

[72]: V. A. Spector and H. Flashner. "Sensitivity of Structural Models for Noncollocated Control Systems". Journal of Dynamic Systems. Measurement. And Control. Vol. 111. December 1989. pp. 646-655.

[73]: Rex J. Theodore. Ashitava Ghosal. "Comparison of the Assumed Modes and Finite Element Models for Flexible Multilink Manipulators". The International Journal of Robotics Research. Vol. 14. No. 2. April 1995. pp. 91-111.

[74]: K. Y. Toumi. J. Bobbett." Stability of Uncertain Linear Systems With Time Delay". Journal of Dynamic Systems. Measurement. and Control. Vol. 113. pp. 558-567. December 1991.

[75]: K. Y. Toumi. T. A. Fuhlbridge. " Application of a Decentralized Time Delay Controller to Robot Manipulators". Proceedings of the 1989 IEEE Conference on Robotics and Automation. pp.1786-1791.

[76]: K. Y. Toumi. O. Ito." A Time Delay Controller for Systems With Unknown Dynamics". Journal of Dynamic Systems. Measurement. and Control. Vol. 112. pp. 133-142. March 1990.

[77]: K. Y. Toumi. O. Ito. "Model Reference Control Using Time Delay For Non Linear Plants With Unknown Dynamics", Proceedings of International Federation of Automatic Control World Congress, Munich. Federal Republic of Germany, July 1987.

[78]: K. Y. Toumi, J. Reddy," Analysis of Linear Time Invariant Systems with Time

Delay", Journal of Dynamic Systems, Measurement, and Control, Vol. 114, pp. 544-555, December 1992.

[79]: K. Y. Toumi, S. T. Wu," Input/Output Linearization Using Time Delay Control", Journal of Dynamic Systems, Measurement, and Control, Vol. 114, pp. 10-19, March 1992.

[80]: A. P. Tzes and S. Yurkovich, "A Frequency Domain Identification Scheme for Flexible Structure Control", Journal of Dynamic Systems, Measurement, and Control, Vol.112, September 1990, pp. 427-434.

[81]: P.B. Usoro, R. Nadira, S.S. Mahil, " A Finite Element/Lagrange Approach to modelling lightweight Flexible Manipulators", ASME Journal of Dynamic Systems, Measurement, and Control, Vol. 108, sept. 1986, pp.198-205.

[82]: D. Wang, M. Vidyasagar, " Control of a Class of Manipulators With a Single Flexible Link. Part I: Feedback Linearization" Journal of Dynamic systems, Measurement, and Control, Vol. 113, pp. 655-661, December 1991.

[83]: D. Wang, M. Vidyasagar, " Control of a Class of Manipulators With a Single Flexible Link. Part II: Observer Controller Stabilization" Journal of Dynamic systems, Measurement, and Control, Vol. 113, pp. 662-668, December 1991.

[84]: Fei-Yue Wang, " On the Extremal Fundamental Frequencies of One-Link Flexible Manipulators", The International Journal of Robotics Research, Vol. 13, No. 2, April 1994, pp. 162-170.

[85]: B. Yang, C. D. Mote, Jr.," On Time Delay in Noncolocated Control of Flexible Mechanical Systems" Journal of Dynamic Systems, Measurement, and Control, Vol. 114, pp. 409-415, September 1992.

[86]: B. Yang, C. D. Mote, Jr.," Frequency-Domain Vibration Control of Distributed Gyroscopic Systems" Journal of Dynamic Systems, Measurement, and Control, Vol. 113, pp. 18-25, March 1991.

[87]: B. Yang, C. D. Mote, Jr.," Controllability and Observability of Distribured Gyroscopic Systems", Journal of Dynamic Systems, Measurement, and Control, Vol. 113, pp. 11-17, March 1991.

[88]: B. Yang, C. D. Mote, Jr.," Active Vibration Control of the Axially Moving String in thr S Domain", Journal of Applied Mechanics, Vol. 58, pp. 189-196, March 1991.

[89]: B. S. Yuan, W. J. Book, J. D. Huggins, "Dynamics of Flexible Manipulator Arms:

Alternative Derivation, Verification, and Characteristics for Control", Journal of Dynamic systems, Measurement, and Control, Vol. 115, pp. 394-404, September 1993.

[90]: J. Yuh and T. Young, "Dynamic Modeling of an Axially Moving Beam in Rotation: Simulation and Experiment", Journal of Dynamic Systems, Measurement, and Control, Vol. 113, March 1991, pp. 34-40.

[91]: M. M. Zavarei, Mohammad Jamshidi, "Time-Delay Systems Analysis, Optimization and Applications", North-Holland Systems and Control Series, Volume 9, 1987

[92]: Jack Zhijie Xia, Chia-Hsiang Menq, "R al Time Estimation of Elastic Deformation For End-Point Tracking Control of Felexible Ewo-Link Manipulators", Journal of Dynamic systems, Measurement, and Control, Vol. 115, pp. 385-393, September 1993.

[93]: K. Zuo, David Wang, "Closed Loop Shaped-Input Control of a Class of Manipulators with a Single Flexible Link", Proceedings of the 1992 IEEE International Conference on Robotics and Automation, Nice, France, May 1992, pp.782-787.

[94]: O. J. M. Smith, "Feedback Control Systems", McGraw-Hill, New York, 1958.

[95]: E. Kreyszig, "Advanced Engineering Mathematics, Fifth Edition" , John Wiley & Sons, New York, 1983.

Appendix A

In this appendix, the derivations of the dynamic models are presented. The model of a single link flexible manipulator is derived using the kinetic energy and the elastic potential energy of a flexible link with fixed base combined with the kinetic energy of a load attached at the tip of this link. The dynamic model of the two link flexible manipulator is derived using a flexible link with fixed base to represent the first link (lower arm) and a flexible link with a moving base to represent the second link (upper arm). A load at the end of the first link is used to represent the mass and the moment of inertia of the actuator driving the second link. At the tip of the second link, another mass is used to represent the payload carried by the two link flexible manipulator.

A.1 Generalized Inertia Matrix (GIM) of a link with fixed base

The absolute position vector of a point on the link shown in Figure 3.1 is given by:

$$\boldsymbol{r} = \begin{bmatrix} c_1 & -s_1 \\ s_1 & c_1 \end{bmatrix} \begin{bmatrix} x \\ \Phi^T(x)\eta \end{bmatrix} \tag{A.1}$$

$$\Phi^T(x)\eta = \phi_1(x)\xi_{e_1} + \phi_2(x)\delta_{e_1}$$

To derive the GIM given by Equation 3.10, the partial derivatives of the position vector $\boldsymbol{r}$

with respect to θ, ξ_e and δ_e are needed. These partial derivatives are given by:

$$\frac{\partial r}{\partial \theta} = \begin{bmatrix} -xs_1 - \Phi^T(x)\eta c_1 & xc_1 - \Phi^T(x)s_1 \end{bmatrix}^T \qquad \text{A.2}$$

$$\frac{\partial r}{\partial \xi_e} = \begin{bmatrix} -\phi_1(x)s_1 & \phi_1(x)c_1 \end{bmatrix}^T \qquad \text{A.3}$$

$$\frac{\partial r}{\partial \delta_e} = \begin{bmatrix} -\phi_2(x)s_1 & \phi_2(x)c_1 \end{bmatrix}^T \qquad \text{A.4}$$

The elements of the matrix $\boldsymbol{M}$ in Equation 3.10 are obtained by evaluating the following integrals :

$$(1,1) = m \int_0^l \frac{\partial r}{\partial \theta} \cdot \frac{\partial r}{\partial \theta} \, dx \qquad \text{A.5}$$

$$(1,2) = m \int_0^l \frac{\partial r}{\partial \theta} \cdot \frac{\partial r}{\partial \xi_e} \, dx \qquad \text{A.6}$$

$$(1,3) = m \int_0^l \frac{\partial r}{\partial \theta} \cdot \frac{\partial r}{\partial \delta_e} \, dx \qquad \text{A.7}$$

$$(2,2) = m\int_0^l \frac{\partial r}{\partial \xi_c}\cdot\frac{\partial r}{\partial \xi_c}\, dx \qquad \text{A.8}$$

$$(2,3) = m\int_0^l \frac{\partial r}{\partial \xi_c}\cdot\frac{\partial r}{\partial \delta_c}\, dx \qquad \text{A.9}$$

$$(3,3) = m\int_0^l \frac{\partial r}{\partial \delta_c}\cdot\frac{\partial r}{\partial \delta_c}\, dx \qquad \text{A.10}$$

The scalar products under the integral can expressed in terms of the shape functions $\phi_1(x)$ and $\phi_2(x)$, and the vector η representing the deflection and slope at the tip as:

$$\frac{\partial r}{\partial \theta}\cdot\frac{\partial r}{\partial \theta} = x^2 + \eta^T\Phi(x)\Phi^T(x)\eta \qquad \text{A.11}$$

$$\frac{\partial r}{\partial \theta}\cdot\frac{\partial r}{\partial \xi_c} = x\phi_1(x) \qquad \text{A.12}$$

$$\frac{\partial r}{\partial \theta}\cdot\frac{\partial r}{\partial \delta_c} = x\phi_2(x) \qquad \text{A.13}$$

$$\frac{\partial r}{\partial \xi_e} \cdot \frac{\partial r}{\partial \xi_e} = \phi_1^2(x) \qquad \text{A.14}$$

$$\frac{\partial r}{\partial \xi_e} \cdot \frac{\partial r}{\partial \delta_e} = \phi_1(x)\phi_2(x) \qquad \text{A.15}$$

$$\frac{\partial r}{\partial \delta_e} \cdot \frac{\partial r}{\partial \delta_e} = \phi_2^2(x) \qquad \text{A.16}$$

A.2 GIM of a load at the end of a link with fixed base:

The position vector of the end point of the link shown in Figure 3.1 is given by:

$$r_e = \begin{bmatrix} c_1 & -s_1 \\ s_1 & c_1 \end{bmatrix} \begin{bmatrix} l_1 \\ \xi_e \end{bmatrix} \qquad \text{A.17}$$

The elements of the matrix M_L in Equation 3.15, which represent the elements of the GIM associated with a load at the end of the link are given by:

$$(1,1) = m_L \frac{\partial r_e}{\partial \theta} \cdot \frac{\partial r_e}{\partial \theta} + I_L \qquad \text{A.18}$$

$$(1,2) = m_L \frac{\partial r_e}{\partial \theta} \cdot \frac{\partial r_e}{\partial \xi_e} \qquad \text{A.19}$$

$$(1,3) = m_L \frac{\partial r_e}{\partial \theta} \cdot \frac{\partial r_e}{\partial \delta_e} + I_L \qquad \text{A.20}$$

$$(2,2) = m_L \frac{\partial r_e}{\partial \xi_e} \cdot \frac{\partial r_e}{\partial \xi_e} \qquad \text{A.21}$$

$$(2,3) = m_L \frac{\partial r_e}{\partial \xi_e} \cdot \frac{\partial r_e}{\partial \delta_e} \qquad \text{A.22}$$

$$(3,3) = m_L \frac{\partial r_e}{\partial \delta_e} \cdot \frac{\partial r_e}{\partial \delta_e} + I_L \qquad \text{A.23}$$

The scalar products can be expressed as:

$$\frac{\partial r_e}{\partial \theta} \cdot \frac{\partial r_e}{\partial \theta} = l^2 + \xi_e^2 \qquad \text{A.24}$$

$$\frac{\partial r_e}{\partial \theta} \cdot \frac{\partial r_e}{\partial \xi_e} = l \qquad \text{A.25}$$

$$\frac{\partial r_e}{\partial \theta} \cdot \frac{\partial r_e}{\partial \delta_e} = 0 \qquad \text{A.26}$$

$$\frac{\partial r_e}{\partial \xi_e} \cdot \frac{\partial r_e}{\partial \xi_e} = 1 \qquad \text{A.27}$$

$$\frac{\partial r_e}{\partial \xi_e} \cdot \frac{\partial r_e}{\partial \delta_e} = 0 \qquad \text{A.28}$$

$$\frac{\partial r_e}{\partial \delta_e} \cdot \frac{\partial r_e}{\partial \delta_e} = 0 \qquad \text{A.29}$$

A.3 Stiffness Matrix of a link with fixed base

The stiffness matrix K is derived from the elastic potential energy U_e. The derivation is shown in Chapter 3 from Equation 3.22 to 3.25. The resulting expression of K is given by:

$$\mathrm{K} = EI \int_0^l \left[\frac{d^2\Phi}{dx^2}\right] \left[\frac{d^2\Phi}{dx^2}\right]^T dx \qquad \text{A.30}$$

Since $\Phi=[\phi_1(x) \quad \phi_2(x)]^T$, the term under the integral is a 2 by 2 matrix given by:

$$\left[\frac{d^2\Phi}{dx^2}\right]\left[\frac{d^2\Phi}{dx^2}\right]^T = \begin{bmatrix} \left(\frac{d^2\phi_1}{dx^2}\right)^2 & \left(\frac{d^2\phi_1}{dx^2}\right)\left(\frac{d^2\phi_2}{dx^2}\right) \\ \left(\frac{d^2\phi_1}{dx^2}\right)\left(\frac{d^2\phi_2}{dx^2}\right) & \left(\frac{d^2\phi_2}{dx^2}\right)^2 \end{bmatrix} \qquad \text{A.31}$$

When the hermit cubics given by Equations 3.4 and 3.5 are used to represent the shape functions $\phi_1(x)$ and $\phi_2(x)$, the elements of the symmetric matrix in Equation A.31 are:

$$(1,1) = \frac{12}{l^3} \qquad \text{A.32}$$

$$(1,2) = (2,1) = -\frac{6}{l^2} \qquad \text{A.33}$$

$$(2,2) = \frac{4}{l} \qquad \text{A.34}$$

A.4 GIM of a Link with moving base

A flexible link with a moving base is shown in Figure 6.1. Whith reference to this figure, the position vector of a point on the link with respect to the inertial frame OXY is given by:

$$r = \begin{bmatrix} c_1 & -s_1 \\ s_1 & c_1 \end{bmatrix} \left[\begin{bmatrix} l_1 \\ \xi_{e_1} \end{bmatrix} + \begin{bmatrix} c_{12} & -s_{12} \\ s_{12} & c_{12} \end{bmatrix} \begin{bmatrix} x \\ \Phi^T(x)\eta_{e_2} \end{bmatrix} \right] \tag{A.35}$$

Similar to the GIM of a link with fixed base, the elements of the GIM of a link with a moving base given by Equation 6.7 can be obtained from the evaluation of the following integrals:

$$(1,1) = m_2 \int_0^{l_2} \frac{\partial r}{\partial \theta_1} \cdot \frac{\partial r}{\partial \theta_1} \, dx \tag{A.36}$$

$$(1,2) = m_2 \int_0^{l_2} \frac{\partial r}{\partial \theta_1} \cdot \frac{\partial r}{\partial \theta_2} \, dx \tag{A.37}$$

$$(1,3) = m_2 \int_0^{l_2} \frac{\partial r}{\partial \theta_1} \cdot \frac{\partial r}{\partial \xi_{e_1}} \, dx \tag{A.38}$$

$$(1,4) = m_2 \int_0^{l_2} \frac{\partial r}{\partial \theta_1} \cdot \frac{\partial r}{\partial \delta_{e_1}} \, dx \tag{A.39}$$

$$(1,5) = m_2 \int_0^{l_2} \frac{\partial r}{\partial \theta_1} \cdot \frac{\partial r}{\partial \xi_{e_2}} \, dx \tag{A.40}$$

$$(1,6) = m_2 \int_0^{l_2} \frac{\partial r}{\partial \theta_1} \cdot \frac{\partial r}{\partial \delta_{e_2}} \, dx \qquad \text{A.41}$$

$$(2,2) = m_2 \int_0^{l_2} \frac{\partial r}{\partial \theta_2} \cdot \frac{\partial r}{\partial \theta_2} \, dx \qquad \text{A.42}$$

$$(2,3) = m_2 \int_0^{l_2} \frac{\partial r}{\partial \theta_2} \cdot \frac{\partial r}{\partial \xi_{e_1}} \, dx \qquad \text{A.43}$$

$$(2,4) = m_2 \int_0^{l_2} \frac{\partial r}{\partial \theta_2} \cdot \frac{\partial r}{\partial \delta_{e_1}} \, dx \qquad \text{A.44}$$

$$(2,5) = m_2 \int_0^{l_2} \frac{\partial r}{\partial \theta_2} \cdot \frac{\partial r}{\partial \xi_{e_1}} \, dx \qquad \text{A.45}$$

$$(2,6) = m_2 \int_0^{l_2} \frac{\partial r}{\partial \theta_2} \cdot \frac{\partial r}{\partial \delta_{e_2}} \, dx \qquad \text{A.46}$$

$$(3,3) = m_2 \int_0^{l_2} \frac{\partial r}{\partial \xi_{e_1}} \cdot \frac{\partial r}{\partial \xi_{e_1}} \, dx \qquad \text{A.47}$$

$$(3,4) \quad m_2 \int_0^{l_2} \frac{\partial r}{\partial \xi_{e_1}} \cdot \frac{\partial r}{\partial \delta_{e_1}} \, dx \qquad \text{A.48}$$

$$(3,5) = m_2 \int_0^{l_2} \frac{\partial r}{\partial \xi_{e_1}} \cdot \frac{\partial r}{\partial \xi_{e_2}} \, dx \qquad \text{A.49}$$

$$(3,6) \quad m_2 \int_0^{l_2} \frac{\partial r}{\partial \xi_{e_1}} \cdot \frac{\partial r}{\partial \delta_{e_2}} \, dx \qquad \text{A.50}$$

$$(4,4) = m_2 \int_0^{l_2} \frac{\partial r}{\partial \delta_{e_1}} \cdot \frac{\partial r}{\partial \delta_{e_1}} \, dx \qquad \text{A.51}$$

$$(4,5) = m_2 \int_0^{l_2} \frac{\partial r}{\partial \delta_{e_1}} \cdot \frac{\partial r}{\partial \xi_{e_2}} \, dx \qquad \text{A.52}$$

$$(4,6) = m_2 \int_0^{l_2} \frac{\partial r}{\partial \delta_{e_1}} \cdot \frac{\partial r}{\partial \delta_{e_2}} \, dx \qquad \text{A.53}$$

$$(5,5) = m_2 \int_0^{l_2} \frac{\partial r}{\partial \xi_{e_2}} \cdot \frac{\partial r}{\partial \xi_{e_2}} \, dx_2 \qquad \text{A.54}$$

$$(5,6) = m_2 \int_0^{l_2} \frac{\partial r}{\partial \xi_{e_2}} \cdot \frac{\partial r}{\partial \delta_{e_2}}\, dx \qquad \text{A.55}$$

$$(6,6) = m_2 \int_0^{l_2} \frac{\partial r}{\partial \delta_{e_2}} \cdot \frac{\partial r}{\partial \delta_{e_2}}\, dx \qquad \text{A.56}$$

The scalar products are expressed in terms of $\Phi(x_2)=[\phi_1(x_2) \quad \phi_2(x_2)]^T$, $\eta_{e2}=[\xi_{e2} \quad \delta_{e2}]^T$, l_1, ξ_{e1} and δ_{e2} as:

$$\frac{\partial r}{\partial \theta_1} \cdot \frac{\partial r}{\partial \theta_1} = l_1^2 + \xi_{e_1}^2 + x_2^2 + \eta_{e_2}^T \Phi(x_2)\Phi^T(x_2)\eta_{e_2} + 2(l_1 x_2 + \xi_{e_1}\Phi^T(x_2)\eta_{e_2})c_{12} - 2l_1\Phi^T(x_2)\eta_{e_2}s_{12} \qquad \text{A.57}$$

$$\frac{\partial r}{\partial \theta_1} \cdot \frac{\partial r}{\partial \theta_2} = x_2^2 + \eta_{e_2}^T \Phi(x_2)\Phi^T(x_2)\eta + (l_1 x_2 + \xi_{e_1}\Phi^T(x_2)\eta_{e_2})c_{12} + (\xi_{e_1} x_2 - l_1\Phi^T(x_2)\eta_{e_2})s_{12} \qquad \text{A.58}$$

$$\frac{\partial r}{\partial \theta_1} \cdot \frac{\partial r}{\partial \xi_{e_1}} = l_1 + x_2 c_{12} - \Phi^T(x_2)\eta_{e_2}s_{12} \qquad \text{A.59}$$

$$\frac{\partial r}{\partial \theta_1} \cdot \frac{\partial r}{\partial \delta_{e_1}} = \frac{\partial r}{\partial \theta_1} \cdot \frac{\partial r}{\partial \theta_2} \qquad \text{A.60}$$

$$\frac{\partial r}{\partial \theta_1} \cdot \frac{\partial r}{\partial \xi_{e_2}} = \phi_1(x_2)(l_1 c_{12} + \xi_{e_1} s_{12} + x_2) \qquad \text{A.61}$$

$$\frac{\partial r}{\partial \theta_1} \cdot \frac{\partial r}{\partial \delta_{e_2}} = \phi_2(x_2)(l_1 c_{12} + \xi_{e_1} s_{12} + x_2) \qquad \text{A.62}$$

$$\frac{\partial r}{\partial \theta_2} \cdot \frac{\partial r}{\partial \theta_2} = x_2^2 + \eta_{e_2}^T \Phi(x_2) \Phi^T(x_2) \eta_{e_2} \qquad \text{A.63}$$

$$\frac{\partial r}{\partial \theta_2} \cdot \frac{\partial r}{\partial \xi_{e_1}} = x_2 c_{12} - \Phi^T(x_2) \eta_{e_2} s_{12} \qquad \text{A.64}$$

$$\frac{\partial r}{\partial \theta_2} \cdot \frac{\partial r}{\partial \delta_{e_1}} = \frac{\partial r}{\partial \theta_2} \cdot \frac{\partial r}{\partial \theta_2} \qquad \text{A.65}$$

$$\frac{\partial r}{\partial \theta_2} \cdot \frac{\partial r}{\partial \xi_{e_2}} = x_2 \phi_1(x_2) \qquad \text{A.66}$$

$$\frac{\partial r}{\partial \theta_2} \cdot \frac{\partial r}{\partial \delta_{e_2}} = x_2 \phi_2(x_2) \qquad \text{A.67}$$

$$\frac{\partial r}{\partial \xi_{e_1}} \cdot \frac{\partial r}{\partial \xi_{e_1}} = 1 \qquad \text{A.68}$$

$$\frac{\partial r}{\partial \xi_{e_1}} \cdot \frac{\partial r}{\partial \delta_{e_1}} = x_2 c_{12} - \Phi^T(x_2)\eta_{e_2} s_{12} \qquad \text{A.69}$$

$$\frac{\partial r}{\partial \xi_{e_1}} \cdot \frac{\partial r}{\partial \xi_{e_2}} = \phi_1(x_2) c_{12} \qquad \text{A.70}$$

$$\frac{\partial r}{\partial \xi_{e_1}} \cdot \frac{\partial r}{\partial \delta_{e_2}} = \phi_2(x_2) c_{12} \qquad \text{A.71}$$

$$\frac{\partial r}{\partial \delta_{e_1}} \cdot \frac{\partial r}{\partial \delta_{e_1}} = x_2^2 + \eta_{e_2}^T \Phi(x_2) \Phi^T(x_2) \eta_{e_2} \qquad \text{A.72}$$

$$\frac{\partial r}{\partial \delta_{e_1}} \cdot \frac{\partial r}{\partial \xi_{e_2}} = x_2 \phi_1(x_2) \qquad \text{A.73}$$

$$\frac{\partial r}{\partial \delta_{e_1}} \cdot \frac{\partial r}{\partial \delta_{e_2}} = x_2 \phi_2(x_2) \qquad \text{A.74}$$

$$\frac{\partial r}{\partial \xi_{e_2}} \cdot \frac{\partial r}{\partial \xi_{e_2}} = \phi_1^2(x_2) \qquad \text{A.75}$$

$$\frac{\partial r}{\partial \xi_{e_2}} \cdot \frac{\partial r}{\partial \delta_{e_2}} = \phi_1(x_2)\phi_2(x_2) \qquad \text{A.76}$$

$$\frac{\partial r}{\partial \delta_{e_2}} \cdot \frac{\partial r}{\partial \delta_{e_2}} = \phi_2^2(x_2) \qquad \text{A.77}$$

A.5 GIM of a load at the end of a link with moving base

The position of the end point of the link in Figure 6.1 is given by:

$$r_{e_2} = \begin{bmatrix} c_1 & -s_1 \\ s_1 & c_1 \end{bmatrix} \left[\begin{bmatrix} l_1 \\ \xi_{e_1} \end{bmatrix} + \begin{bmatrix} c_{12} & -s_{12} \\ s_{12} & c_{12} \end{bmatrix} \begin{bmatrix} l_2 \\ \xi_{e_2} \end{bmatrix} \right] \qquad \text{A.78}$$

The matrix M_L representing the GIM of a load at the end of the flexible link is derived from the kinetic energy as shown in Equations 6.14 to 6.16. The elements of the matrix M_L in Equation 6.15 are given by:

$$(1,1) = m_L \frac{\partial r_{e_2}}{\partial \theta_1} \cdot \frac{\partial r_{e_2}}{\partial \theta_1} + I_L \qquad \text{A.79}$$

$$(1,2) = m_L \frac{\partial r_{e_2}}{\partial \theta_1} \cdot \frac{\partial r_{e_2}}{\partial \theta_2} + I_L \qquad \text{A.80}$$

$$(1,3) = m_L \frac{\partial r_{e_2}}{\partial \theta_1} \cdot \frac{\partial r_{e_2}}{\partial \xi_{e_1}} \qquad \text{A.81}$$

$$(1,4) = m_L \frac{\partial r_{e_2}}{\partial \theta_1} \cdot \frac{\partial r_{e_2}}{\partial \delta_{e_1}} + I_L \qquad \text{A.82}$$

$$(1,5) = m_L \frac{\partial r_{e_2}}{\partial \theta_1} \cdot \frac{\partial r_{e_2}}{\partial \xi_{e_2}} \qquad \text{A.83}$$

$$(1,6) = m_L \frac{\partial r_{e_2}}{\partial \theta_1} \cdot \frac{\partial r_{e_2}}{\partial \delta_{e_2}} + I_L \qquad \text{A.84}$$

$$(2,2) = m_L \frac{\partial r_{e_2}}{\partial \theta_2} \cdot \frac{\partial r_{e_2}}{\partial \theta_2} + I_L \qquad \text{A.85}$$

$$(2,3) = m_L \frac{\partial r_{e_2}}{\partial \theta_2} \cdot \frac{\partial r_{e_2}}{\partial \xi_{e_1}} \qquad \text{A.86}$$

$$(2,4) = m_L \frac{\partial r_{e_2}}{\partial \theta_2} \cdot \frac{\partial r_{e_2}}{\partial \delta_{e_1}} + I_L \qquad \text{A.87}$$

$$(2,5) = m_L \frac{\partial r_{e_2}}{\partial \theta_2} \cdot \frac{\partial r_{e_2}}{\partial \xi_{e_1}} \qquad \text{A.88}$$

$$(2,6) = m_L \frac{\partial r_{e_2}}{\partial \theta_2} \cdot \frac{\partial r_{e_2}}{\partial \delta_{e_2}} + I_L \qquad \text{A.89}$$

$$(3,3) = m_L \frac{\partial r_{e_2}}{\partial \xi_{e_1}} \cdot \frac{\partial r_{e_2}}{\partial \xi_{e_1}} \qquad \text{A.90}$$

$$(3,4) = m_L \frac{\partial r_{e_2}}{\partial \xi_{e_1}} \cdot \frac{\partial r_{e_2}}{\partial \delta_{e_1}} \qquad \text{A.91}$$

$$(3,5) = m_L \frac{\partial r_{e_2}}{\partial \xi_{e_1}} \cdot \frac{\partial r_{e_2}}{\partial \xi_{e_2}} \qquad \text{A.92}$$

$$(3,6) = m_L \frac{\partial r_{e_2}}{\partial \xi_{e_1}} \cdot \frac{\partial r_{e_2}}{\partial \delta_{e_2}} \qquad \text{A.93}$$

$$(4,4) = m_L \frac{\partial r_{e_2}}{\partial \delta_{e_1}} \cdot \frac{\partial r_{e_2}}{\partial \delta_{e_1}} + I_L \qquad \text{A.94}$$

$$(4,5) = m_L \frac{\partial r_{e_2}}{\partial \delta_{e_1}} \cdot \frac{\partial r_{e_2}}{\partial \xi_{e_2}} \qquad \text{A.95}$$

$$(4,6) = m_L \frac{\partial r_{e_2}}{\partial \delta_{e_1}} \cdot \frac{\partial r_{e_2}}{\partial \delta_{e_2}} + I_L \qquad \text{A.96}$$

$$(5,5) = m_L \frac{\partial r_{e_2}}{\partial \xi_{e_2}} \cdot \frac{\partial r_{e_2}}{\partial \xi_{e_2}} \qquad \text{A.97}$$

$$(5,6) = m_L \frac{\partial r_{e_2}}{\partial \xi_{e_2}} \cdot \frac{\partial r_{e_2}}{\partial \delta_{e_2}} \qquad \text{A.98}$$

$$(6,6) = m_L \frac{\partial r_{e_2}}{\partial \delta_{e_2}} \cdot \frac{\partial r_{e_2}}{\partial \delta_{e_2}} + I_L \qquad \text{A.99}$$

The results of the scalar products in Equations A.79 to A.99 can be expressed in terms of the variables ξ_{e1} and ξ_{e2}, and the lengths l_1 and l_2 by evaluating the shape functions at $x_2=l_2$. The resulting expressions are given by:

$$\frac{\partial r_{e_2}}{\partial \theta_1} \cdot \frac{\partial r_{e_2}}{\partial \theta_1} = l_1^2 + \xi_{e_1}^2 + l_2^2 + \xi_{e_2}^2 + 2(l_1 l_2 + \xi_{e_1}\xi_{e_2})c_{12} - 2l_1\xi_{e_2} \qquad \text{A.100}$$

$$\frac{\partial r_{e_2}}{\partial \theta_1} \cdot \frac{\partial r_{e_2}}{\partial \theta_2} = l_2^2 + \xi_{e_2}^2 + (l_1 l_2 + \xi_{e_1}\xi_{e_2})c_{12} + (\xi_{e_1} l_2 - l_1 \xi_{e_2})s_{12} \qquad \text{A.101}$$

$$\frac{\partial r_{e_2}}{\partial \theta_1} \cdot \frac{\partial r_{e_2}}{\partial \xi_{e_1}} = l_1 + l_2 c_{12} - \xi_{e_2} s_{12} \tag{A.102}$$

$$\frac{\partial r_{e_2}}{\partial \theta_1} \cdot \frac{\partial r_{e_2}}{\partial \delta_{e_1}} = \frac{\partial r_{e_2}}{\partial \theta_1} \cdot \frac{\partial r_{e_2}}{\partial \theta_2} \tag{A.103}$$

$$\frac{\partial r_{e_2}}{\partial \theta_1} \cdot \frac{\partial r_{e_2}}{\partial \xi_{e_2}} = l_1 c_{12} + \xi_{e_1} s_{12} + l_2 \tag{A.104}$$

$$\frac{\partial r_{e_2}}{\partial \theta_1} \cdot \frac{\partial r_{e_2}}{\partial \delta_{e_2}} = 0 \tag{A.105}$$

$$\frac{\partial r_{e_2}}{\partial \theta_2} \cdot \frac{\partial r_{e_2}}{\partial \theta_2} = l_2^2 + \xi_{e_2}^2 \tag{A.106}$$

$$\frac{\partial r_{e_2}}{\partial \theta_2} \cdot \frac{\partial r_{e_2}}{\partial \xi_{e_1}} = l_2 c_{12} - \xi_{e_2} s_{12} \tag{A.107}$$

$$\frac{\partial r_{e_2}}{\partial \theta_2} \cdot \frac{\partial r_{e_2}}{\partial \delta_{e_1}} = \frac{\partial r_{e_2}}{\partial \theta_2} \cdot \frac{\partial r_{e_2}}{\partial \theta_2} \tag{A.108}$$

$$\frac{\partial r_{e_2}}{\partial \theta_2} \cdot \frac{\partial r_{e_2}}{\partial \xi_{e_2}} = l_2 \tag{A.109}$$

$$\frac{\partial r_{e_2}}{\partial \theta_2} \cdot \frac{\partial r_{e_2}}{\partial \delta_{e_2}} = 0 \tag{A.110}$$

$$\frac{\partial r_{e_2}}{\partial \xi_{e_1}} \cdot \frac{\partial r_{e_2}}{\partial \xi_{e_1}} = 1 \tag{A.111}$$

$$\frac{\partial r_{e_2}}{\partial \xi_{e_1}} \cdot \frac{\partial r_{e_2}}{\partial \delta_{e_1}} = l_2 c_{12} - \xi_{e_2} s_{12} \tag{A.112}$$

$$\frac{\partial r_{e_2}}{\partial \xi_{e_1}} \cdot \frac{\partial r_{e_2}}{\partial \xi_{e_2}} = c_{12} \tag{A.113}$$

$$\frac{\partial r_{e_2}}{\partial \xi_{e_1}} \cdot \frac{\partial r_{e_2}}{\partial \delta_{e_2}} = 0 \tag{A.114}$$

$$\frac{\partial r_{e_2}}{\partial \delta_{e_1}} \cdot \frac{\partial r_{e_2}}{\partial \delta_{e_1}} = l_2^2 + \xi_{e_2}^2 \tag{A.115}$$

$$\frac{\partial r_{e_2}}{\partial \delta_{e_1}} \cdot \frac{\partial r_{e_2}}{\partial \xi_{e_2}} = l_2 \tag{A.116}$$

$$\frac{\partial r_{e_2}}{\partial \delta_{e_1}} \cdot \frac{\partial r_{e_2}}{\partial \delta_{e_2}} = 0 \tag{A.117}$$

$$\frac{\partial r_{e_2}}{\partial \xi_{e_2}} \cdot \frac{\partial r_{e_2}}{\partial \xi_{e_2}} = 1 \qquad \text{A.118}$$

$$\frac{\partial r_{e_2}}{\partial \xi_{e_2}} \cdot \frac{\partial r_{e_2}}{\partial \delta_{e_2}} = 0 \qquad \text{A.119}$$

$$\frac{\partial r_{e_2}}{\partial \delta_{e_2}} \cdot \frac{\partial r_{e_2}}{\partial \delta_{e_2}} = 0 \qquad \text{A.120}$$

A.6 Stiffness Matrix of a link with moving base

The stiffness matrix of a link with moving base is the same as that of a link with fixed base because the variables coordinates used to describe the flexibility of the link are measured with respect to the equivalent rigid link. The resulting expression for the stiffness matrix is given by:

$$K_2 = EI_2 \int_0^{l_2} \left[\frac{d^2\Phi_2}{dx^2}\right] \left[\frac{d^2\Phi_2}{dx^2}\right]^T dx \qquad \text{A.121}$$

Φ_2 is a vector representing the shape functions that describe the deflection of the link and given by $\Phi_2=[\phi_{21}(x) \quad \phi_{22}(x)]^T$, the term under the integral is a 2 by 2 matrix given by:

$$\left[\frac{d^2\Phi_2}{dx^2}\right]\left[\frac{d^2\Phi_2}{dx^2}\right]^T = \begin{bmatrix} \left(\frac{d^2\phi_{21}}{dx^2}\right)^2 & \left(\frac{d^2\phi_{21}}{dx^2}\right)\left(\frac{d^2\phi_{22}}{dx^2}\right) \\ \left(\frac{d^2\phi_{21}}{dx^2}\right)\left(\frac{d^2\phi_{22}}{dx^2}\right) & \left(\frac{d^2\phi_{22}}{dx^2}\right)^2 \end{bmatrix} \qquad \text{A.122}$$

When the hermit cubics are used to represent the shape functions $\phi_{21}(x)$ and $\phi_{22}(x)$, the elements of the symmetric matrix in Equation A.122 are:

$$(1,1) = \frac{12}{l_2^3} \qquad \text{A.123}$$

$$(1,2) = (2,1) = -\frac{6}{l_2^2} \qquad \text{A.124}$$

$$(2,2) = \frac{4}{l_2} \qquad \text{A.125}$$

Appendix B

This appendix consists of a listing of a series of Maple V programs. These programs were very useful in the derivation of the dynamic models because of their ability to perform symbolic calculations. Section B.1 shows a listing of the program used to derive the dynamic model of the one link flexible manipulator. An explicit form of the dynamic model of the one link flexible manipulator is shown in Section B2. A listing of the Maple V program used to derive the dynamic model of the two link flexible manipulator is shown in Section B.3 and an explicit form of this model is shown in Section B.4.

In order to reduce the complexity of the dynamic equations, the constant terms involving the integration of the shape functions over the length of the link are calculated first. These constants depend on the choice of the shape functions and they are given

$$a_{ij} = \int_0^l \phi_i(x)\phi_j(x)\,dx \qquad \text{B.1}$$

$$a_i = \int_0^l \phi_i(x)\,dx \qquad \text{B.2}$$

$$b_i = \int_0^l \phi_i(x)\, x\, dx \tag{B.3}$$

$$k_{ij} = \int_0^l \frac{\partial^2 \phi_i(x)}{\partial x^2} \frac{\partial^2 \phi_j(x)}{\partial x^2} dx \tag{B.4}$$

B.1 Maple Program to Generate the Dynamic Model of a One-Link Flexible Manipulator.

```
#----------------------------------------------------------
#
# Name:
#               Model1.m
#
# Description:
#
#       This program performs a symbolic derivation
#       of the dynamic model of a one link
#       flexible manipulator. It assumes that the #
#       elements of the 3 by 3 symmetric matrix
#       representing the GIM of the link are #
#       available in explicit forms. The programs #
#       performs the following:
#
#               1. Generates the GIM
#               2. Evaluates the elements of the
#                  stiffness matrix
#               3. Generates the GIM of a load at
#                  the end of link
#               4. Evaluates the nonlinear coriolus
#                  and centrifugal terms.
#
#       m: Mass/unit length.
#       l: Lengths of link.
#       t1: Joint angle.
#       t1d: Joint velocity.
#       u1: Tip deflection.
#       u1d: First time derivative of u1.
#       u2: Tip slope.
#       u2d: First time derivative of u2.
#----------------------------------------------------------
#
#       set the numerical values to 5 digits after the
#       decimal point.
#
Digits:=5;
#
#----------------------------------------------------------
#       ---------------- GIM of the link -----
# ----------------------
#  Element (1,1).

n11:=m*(l^3/3+u1^2*a11+2*u1*u2*a12+u2^2*a22);

#  Element (1,2).
     n12:=m*b1;

#  Element (1,3).
     n13:=m*b2;

#  Element (2,2).
     n22:=m*a11;

#  Element (2,3).
     n23:=m*a12;

#  Element (3,3).
     n33:=m*a22;
#----------------------------------------------------------
#----------------- Stiffness matrix. K ----------
#       EI: Stiffness of link
#----------------------------------------------------------
     k1:=0;
     k3:=EI*(k11*u1+k12*u2);
     k4:=EI*(k12*u1+k22*u2);
#----------------------------------------------------------
# Elements of the Inertia Matrix associated with a
# load at the
# end of the link.
#
#       mL: Mass of the load
#       IL: Moment of Inertia of the load
#----------------------------------------------------------
#  Element (1,1).
     J11:=mL*(l^2+u1^2)+IL;

#  Element (1,2).
     J12:=mL*l;

#  Element (1,3).
     J13:=IL;

#  Element (2,2).
     J22:=mL;

#  Element (2,3).
     J23:=0;

#  Element (3,3).
     J33:=IL;
#----------------------------------------------------------
#  Assembling of GIM of the link and GIM of the
#  load

m11:=n11+J11;
m12:=n12+J12;
m13:=n13+J13;
m22:=n22+J22;
m23:=n23+J23;
m33:=n33+J33;
#----------------------------------------------------------
```

```
#---- Calculate the coriolis and centrifugal terms -----
#

# Evaluate time derivatives of [m11 m12
#                               m12 m22]

temp11:=diff(m11,t1)*t1d+diff(m11,u1)*u1d
     +diff(m11,u2)*u2d;

temp12:=diff(m12,t1)*t1d+diff(m12,u1)*u1d
      +diff(m12,u2)*u2d;

temp13:=diff(m13,t1)*t1d+diff(m13,u1)*u1d
      +diff(m13,u2)*u2d;

temp22:=diff(m22,t1)*t1d+diff(m22,u1)*u1d
      +diff(m22,u2)*u2d;

temp23:=diff(m23,t1)*t1d+diff(m23,u1)*u1d
      +diff(m23,u2)*u2d;

temp33:=diff(m33,t1)*t1d+diff(m33,u1)*u1d
      +diff(m33,u2)*u2d;
#--------------------------------------------------
# Evaluate the total kinetic energy
#--------------------------------------------------

ke11:=m11*t1d+m12*u1d+m13*u2d;
ke12:=m12*t1d+m22*u1d+m23*u2d;
ke13:=m13*t1d+m23*u1d+m33*u2d;

KE:=(ke11*t1d+ke12*u1d+ke13*u2d)/2;

printlevel:=1;
#
#  Coriolus and centrifugal terms
#
h1:=simplify(temp11*t1d+temp12*u1d+temp13*u2d-diff(KE,t1));

h2:=simplify(temp12*t1d+temp22*u1d+temp23*u2d-diff(KE,u1));

h3:=simplify(temp13*t1d+temp23*u1d+temp33*u2d-diff(KE,u2));

save m11,m12,m13,m22,m23,m33, h1,h2,h3, k1,k2,k3, system1;
```

B.2. Dynamic Model of a One-Link Flexible Manipulator.

In this section, the dynamic model created by the program shown in Section B.1 is presented. The dynamic equations can be written in matrix form as:

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{12} & m_{22} & m_{23} \\ m_{13} & m_{23} & m_{33} \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\xi}_e \\ \ddot{\delta}_e \end{bmatrix} + \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} + \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ 0 \\ 0 \end{bmatrix}$$

The elements of the GIM, the coriolis and centrifugal terms, and the stiffness terms are shown below. (u1 represents ξ_e and u2 represents δ_e).

B.2.1 Elements of the GIM

```
m11 = m*(1/3*l^3+u1^2*a11+2*u1*u2*a12
+u2^2*a22)+mL*(l^2+u1^2)+IL;
m12 = m*b1+mL*l;
m13 = m*b2+IL;
m22 = m*a11+mL;
m23 = m*a12;
m33 = m*a22+IL;
```

B.2.2 Coriolis and Centrifugal Terms

```
h1 = (2*u1d*m*u1*a11+2*u1d*m*u2*a12
   +2*u1d*mL*u1+2*m*u2d*u1*a12
   +2*m*u2d*u2*a22)*t1d;
h2 = -(m*u1*a11+m*u2*a12+mL*u1)*t1d^2;
h3 = -m*(u1*a12+u2*a22)*t1d^2;
```

B.2.3 Stiffness Terms

```
k1 = 0;
k2 = EI*(k11*u1+k12*u2);
k3 = EI*(k12*u1+k22*u2);
```

B.3 Maple Program to Generate the Dynamic Model of a Two-Link Flexible Manipulator.

The constant terms involving the integration of the shape functions over the length of each link are calculated first as follows:

$$a_{ijk} = \int_0^{l_k} \phi_{ik}(x)\phi_{jk}(x)\,dx \qquad \text{B.5}$$

$$a_{ik} = \int_0^{l_k} \phi_{ik}(x)\,dx \qquad \text{B.6}$$

$$b_{ik} = \int_0^{l_k} \phi_{ik}(x)\,x\,dx \qquad \text{B.7}$$

$$k_{ijk} = \int_0^{l_k} \frac{\partial^2\phi_{ik}(x)}{\partial x^2}\frac{\partial^2\phi_{jk}(x)}{\partial x^2}\,dx \qquad \text{B.8}$$

The subscript k refers to link 1 or link 2.

```
#----------------------------------------------------
# Name:
#               Model2.m
#
# Description:
#
#       This program performs a symbolic derivation
#       of the dynamic model of a Two link
#       flexible manipulator. It assumes that the #
        elements of the 3 by 3 symmetric matrix
#       representing the GIM of the link are #
        available in explicit forms. The programs
#       performs the following:
#
#               1. Generates the GIM of link 1
#               2. Evaluates the elements of the #
                   stiffness matrix of link 1
#               3. Generates the GIM of a load at #
                   the end of link 1
#               4. Generates the GIM of link 2
#               5. Evaluates the elements of the #
                   stiffness matrix of link 2
#               6. Generates the GIM of a load at #
                   the end of link 1
#               7. Assembles the inertia matrices
#               8. Evaluates the  nonlinear coriolus
#                  and centrifugal terms.
#
#----------------------------------------------------
Digits:=5;
#----------------------------------------------------
#          GIM of link 1
#
#   m1: Mass/unit length  of link 1
#   l1: Length of link 1
#----------------------------------------------------
#   Element (1,1).
```

```
n11:=m1*(l1^3/3+u1^2*a111+2*u1*u2*a121+u2^2*a2
21);
#   Element (1,2).
      n12:=m1*b11;

#   Element (1,3).
      n13:=m1*b21;
#   Element (2,2).
      n22:=m1*a111;
#   Element (2,3).
      n23:=m1*a121;
#   Element (3,3).
      n33:=m1*a221;
#------------------------------------------------------
# Stiffness matrix.  K1
# EI1: Stiffness of link 1
#------------------------------------------------------

      k1:=0;
      k3:=EI1*(k111*u1+k121*u2);
      k4:=EI1*(k121*u1+k221*u2);

#------------------------------------------------------
# Elements of the Inertia Matrix associated with a
# load at the end of link 1 ( Mass and Moment of
# Inertia of the actuator  located at joint 2 )
# ma: Mass of the actuator
# Ia: Moment of Inertia of the actoator
#------------------------------------------------------
#   Element (1,1).
      J11:=ma*(l1^2+u1^2)+Ia;
#   Element (1,2).
      J12:=ma*l1;
#   Element (1,3).
      J13:=Ia;
#   Element (2,2).
      J22:=ma;
#   Element (2,3).
      J23:=0;
#   Element (3,3).
      J33:=Ia;
#------------------------------------------------------
#            GIM of link 2
#
#   m2: Mass/unit length  of link 2
#   l2: Length of link 2
#------------------------------------------------------

#   Element (1,1).

mm11:=m2*(l1^2*l2+u1^2*l2+l2^3/3+u3^2*a112
+2*u3*u4*a122+u4^2*a222 +l1*l2^2+2*u1*(a12*u3
+a22*u4)*cos(t2+u2)-2*l1*(a12*u3+a22*u4)*sin(t2+u
2));

#   Element (1,2).
mm12:=m2*(l2^3/3+u3^2*a112+2*u3*u4*a122+u4^2
*a222+(l1*l2^2/2+u1*(a12*u3+a22*u4))*cos(t2+u2)
+(u1*l2^2/2-l1*(a12*u3+a22*u4))*sin(t2+u2));
#   Element (1,3).
mm13:=m2*(l1*l2+l2^2/2*cos(t2+u2)-(a12*u3+a22*u
4)*sin(t2+u2));
#   Element (1,4).
      mm14:=mm12;
#   Element (1,5).
mm15:=m2*((l1*cos(t2+u2)+u1*sin(t2+u2))*a12
+b12);
#   Element (1,6).
mm16:=m2*((l1*cos(t2+u2)+u1*sin(t2+u2))*a12
+b22);
#   Element (2,2).
mm22:=m2*(l2^3/3+u3^2*a112+2*u3*u4*a122+u4^2
*a222);
#   Element (2,3).
mm23:=m2*(l2^2/2*cos(t2+u2)-(u3*a12+u4*a22)*sin(
t2+u2));
#   Element (2,4).
      mm24:=mm22;
#   Element (2,5).
      mm25:=m2*b12;
#   Element (2,6).
      mm26:=m2*b22;
#   Element (3,3).
      mm33:=m2*l2;
#   Element (3,4).
      mm34:=mm23;
#   Element (3,5).
      mm35:=m2*a12*cos(t2+u2);
#   Element (3,6).
      mm36:=m2*a22*cos(t2+u2);
#   Element (4,4).
      mm44:=mm22;
#   Element (4,5).
      mm45:=mm25;
#   Element (4,6).
      mm46:=mm26;
#   Element (5,5).
      mm55:=m2*a112;
#   Element (5,6).
      mm56:=m2*a122;
#   Element (6,6).
      mm66:=m2*a222;

#------------------------------------------------------
# Stiffness matrix.  K2
```

```
# EI2: Stiffness of link 2
#-----------------------------------------------------------
      k2:=0;
      k5:=EI2*(k112*u3+k122*u4);
      k6:=EI2*(k122*u3+k222*u4);
#-----------------------------------------------------------
# Elements of the Inertia Matrix associated with a
# load at the end of link 2
# mL: Mass of load
# IL: Moment of Inertia of the load with respect to
#       its princical axis
#-----------------------------------------------------------
#   Element (1,1).
I11:=mL*(l1^2+u1^2+l2^2+u3^2+2*(l1*l2+u1*u3)
    *cos(t2+u2)-2*l1*u3*sin(t2+u2))+IL;
#   Element (1,2).
    I12:=mL*(l2^2+u3^2+(l1*l2+u1*u3)*cos(t2+u2)
    +(u1*l2-l1*u2)*sin(t2+u2))+IL;
#   Element (1,3).
    I13:=mL*(l1+l2*cos(t2+u2)-u3*sin(t2+u2));
#   Element (1,4).
     I14:=I12;
#   Element (1,5).
     I15:=mL*(l1*cos(t2+u2)+u1*sin(t2+u2)+l2);
#   Element (1,6).
     I16:=IL;
#   Element (2,2).
     I22:=mL*(l2^2+u3^2)+IL;
#   Element (2,3).
     I23:=mL*(l2*cos(t2+u2)-u3*sin(t2+u2));
#   Element (2,4).
     I24:=I22;
#   Element (2,5).
     I25:=mL*l2;
#   Element (2,6).
     I26:=IL;
#   Element (3,3).
     I33:=mL;
#   Element (3,4).
     I34:=I23;
#   Element (3,5).
     I35:=mL*u3*cos(t2+u2);
#   Element (3,6).
     I36:=0;
#   Element (4,4).
     I44:=I22;
#   Element (4,5).
     I45:=I25;
#   Element (4,6).
     I46:=I26;
#   Element (5,5).
     I55:=mL;
#   Element (5,6).
     I56:=0;
#   Element (6,6).
     I66:=IL;
#-----------------------------------------------------------
#--- Assembling of GIM of link 1 and GIM of link 2

m11:=simplify(mm11+n11+I11+J11);
m12:=mm12+I12;
m13:=simplify(mm13+n12+I13+J12);
m14:=simplify(mm14+n13+I14+J13);
m15:=mm15+I15;
m16:=mm16+I16;
m22:=mm22+I22;
m23:=mm23+I23;
m24:=mm24+I24;
m25:=mm25+I25;
m26:=mm26+I26;
m33:=simplify(mm33+n22+I33+J22);
m34:=simplify(mm34+n23+I34+J23);
m35:=mm35+I35;
m36:=mm36+I36;
m44:=simplify(mm44+n33+I44+J33);
m45:=mm45+I45;
m46:=mm46+I46;
m55:=mm55+I55;
m56:=mm56+I56;
m66:=mm66+I66;
#-----------------------------------------------------------
#---- Calculate the coriolis and centrifugal terms
#
# Evaluate time derivatives of [m11 m12
#                               m12 m22]

temp11:=diff(m11,t1)*t1d+diff(m11,t2)*t2d+diff(m11,
u1)*u1d+diff(m11,u2)*u2d+diff(m11,u3)*u3d+diff(m1
1,u4)*u4d;
temp12:=diff(m12,t1)*t1d+diff(m12,t2)*t2d+diff(m12,
u1)*u1d+diff(m12,u2)*u2d+diff(m12,u3)*u3d+diff(m1
2,u4)*u4d;
temp22:=diff(m22,t1)*t1d+diff(m22,t2)*t2d+diff(m22,
u1)*u1d+diff(m22,u2)*u2d+diff(m22,u3)*u3d+diff(m2
2,u4)*u4d;

# Evaluate time derivatives of [m13 m14 m15 m16
#                               m23 m24 m25 m26]

temp13:=diff(m13,t1)*t1d+diff(m13,t2)*t2d+diff(m13,
u1)*u1d+diff(m13,u2)*u2d+diff(m13,u3)*u3d+diff(m1
3,u4)*u4d;
temp14:=diff(m14,t1)*t1d+diff(m14,t2)*t2d+diff(m14,
u1)*u1d+diff(m14,u2)*u2d+diff(m14,u3)*u3d+diff(m1
4,u4)*u4d;
temp15:=diff(m15,t1)*t1d+diff(m15,t2)*t2d+diff(m15,
```

```
u1)*u1d+diff(m15,u2)*u2d+diff(m15,u3)*u3d+diff(m1
5,u4)*u4d;
temp16:=diff(m16,t1)*t1d+diff(m16,t2)*t2d+diff(m16,
u1)*u1d+diff(m16,u2)*u2d+diff(m16,u3)*u3d+diff(m1
6,u4)*u4d;
temp23:=diff(m23,t1)*t1d+diff(m23,t2)*t2d+diff(m23,
u1)*u1d+diff(m23,u2)*u2d+diff(m23,u3)*u3d+diff(m2
3,u4)*u4d;
temp24:=diff(m24,t1)*t1d+diff(m24,t2)*t2d+diff(m24,
u1)*u1d+diff(m24,u2)*u2d+diff(m24,u3)*u3d+diff(m2
4,u4)*u4d;

temp25:=diff(m25,t1)*t1d+diff(m25,t2)*t2d+diff(m25,
u1)*u1d+diff(m25,u2)*u2d+diff(m25,u3)*u3d+diff(m2
5,u4)*u4d;
temp26:=diff(m26,t1)*t1d+diff(m26,t2)*t2d+diff(m26,
u1)*u1d+diff(m26,u2)*u2d+diff(m26,u3)*u3d+diff(m2
6,u4)*u4d;

#  Evaluate time derivatives of [m33 m34 m35 m36
#                                m34 m44 m45 m46
#                                m35 m45 m55 m56
#                                m36 m46 m56 m66]

temp33:=diff(m33,t1)*t1d+diff(m33,t2)*t2d+diff(m33,
u1)*u1d+diff(m33,u2)*u2d+diff(m33,u3)*u3d+diff(m3
3,u4)*u4d;
temp34:=diff(m34,t1)*t1d+diff(m34,t2)*t2d+diff(m34,
u1)*u1d+diff(m34,u2)*u2d+diff(m34,u3)*u3d+diff(m3
4,u4)*u4d;
temp35:=diff(m35,t1)*t1d+diff(m35,t2)*t2d+diff(m35,
u1)*u1d+diff(m35,u2)*u2d+diff(m35,u3)*u3d+diff(m3
5,u4)*u4d;
temp36:=diff(m36,t1)*t1d+diff(m36,t2)*t2d+diff(m36,
u1)*u1d+diff(m36,u2)*u2d+diff(m36,u3)*u3d+diff(m3
6,u4)*u4d;
temp44:=diff(m44,t1)*t1d+diff(m44,t2)*t2d+diff(m44,
u1)*u1d+diff(m44,u2)*u2d+diff(m44,u3)*u3d+diff(m4
4,u4)*u4d;
temp45:=diff(m45,t1)*t1d+diff(m45,t2)*t2d+diff(m45,
u1)*u1d+diff(m45,u2)*u2d+diff(m45,u3)*u3d+diff(m4
5,u4)*u4d;
temp46:=diff(m46,t1)*t1d+diff(m46,t2)*t2d+diff(m46,
u1)*u1d+diff(m46,u2)*u2d+diff(m46,u3)*u3d+diff(m4
6,u4)*u4d;
temp55:=diff(m55,t1)*t1d+diff(m55,t2)*t2d+diff(m55,
u1)*u1d+diff(m55,u2)*u2d+diff(m55,u3)*u3d+diff(m5
5,u4)*u4d;
temp56:=diff(m56,t1)*t1d+diff(m56,t2)*t2d+diff(m56,
u1)*u1d+diff(m56,u2)*u2d+diff(m56,u3)*u3d+diff(m5
6,u4)*u4d;
temp66:=diff(m66,t1)*t1d+diff(m66,t2)*t2d+diff(m66,
u1)*u1d+diff(m66,u2)*u2d+diff(m66,u3)*u3d+diff(m6
6,u4)*u4d;
#-------------------------------------------------------------
# Evaluate the total kinetic energy
#-------------------------------------------------------------
ke11:=m11*t1d+m12*t2d+m13*u1d+m14*u2d+m15*u
3d+m16*u4d;
ke12:=m12*t1d+m22*t2d+m23*u1d+m24*u2d+m25*u
3d+m26*u4d;
ke13:=m13*t1d+m23*t2d+m33*u1d+m34*u2d+m35*u
3d+m36*u4d;
ke14:=m14*t1d+m24*t2d+m34*u1d+m44*u2d+m45*u
3d+m46*u4d;
ke15:=m15*t1d+m25*t2d+m35*u1d+m45*u2d+m55*u
3d+m56*u4d;
ke16:=m16*t1d+m26*t2d+m36*u1d+m46*u2d+m56*u
3d+m66*u4d;

KE:=(ke11*t1d+ke12*t2d+ke13*u1d+ke14*u2d+ke15
*u3d+ke16*u4d)/2;

printlevel:=1;
#
#  Coriolus and centrifugal terms
#
h1:=simplify(temp11*t1d+temp12*t2d+temp13*u1d
+temp14*u2d+temp15*u3d+temp16*u4d-diff(KE,t1));
h2:=simplify(temp12*t1d+temp22*t2d+temp23*u1d
+temp24*u2d+temp25*u3d+temp26*u4d-diff(KE,t2));
h3:=simplify(temp13*t1d+temp23*t2d+temp33*u1d
+temp34*u2d+temp35*u3d+temp36*u4d-diff(KE,u1));
h4:=simplify(temp14*t1d+temp24*t2d+temp34*u1d
+temp44*u2d+temp45*u3d+temp46*u4d-diff(KE,u2));
h5:=simplify(temp15*t1d+temp25*t2d+temp35*u1d
+temp45*u2d+temp55*u3d+temp56*u4d-diff(KE,u3));
h6:=simplify(temp16*t1d+temp26*t2d+temp36*u1d
+temp46*u2d+temp56*u3d+temp66*u4d-diff(KE,u4));

save m11,m12,m13,m14,m15,m16,m22,m23,m24,m25,
m26,m33,m34,m35,m36,m44,m45,m46,m55,m56,m66,
 h1,h2,h3,h4,h5,h6, k1,k2,k3,k4,k5,k6,
system2;
```

B.4. Dynamic Model of a Two-Link Flexible Robot

In this section, the dynamic model of a two-link flexible manipulator is presented. The equations are created using the Maple program presented in Section B.3. The dynamic model can be written in matrix form as follows:

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} & m_{15} & m_{16} \\ m_{12} & m_{22} & m_{23} & m_{24} & m_{25} & m_{26} \\ m_{13} & m_{22} & m_{33} & m_{34} & m_{35} & m_{36} \\ m_{14} & m_{24} & m_{34} & m_{44} & m_{45} & m_{46} \\ m_{15} & m_{25} & m_{35} & m_{45} & m_{55} & m_{56} \\ m_{16} & m_{26} & m_{36} & m_{46} & m_{56} & m_{66} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\xi}_{e_1} \\ \ddot{\delta}_{e_1} \\ \ddot{\xi}_{e_2} \\ \ddot{\delta}_{e_2} \end{bmatrix} + \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \end{bmatrix} + \begin{bmatrix} k_1 \\ k_2 \\ k_3 \\ k_4 \\ k_5 \\ k_6 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The elements of the GIM, the coriolis and centrifugal terms, and the stiffness terms are shown below. (t2=θ_2, u1=ξ_{e1}, u2=δ_{e1}, u3=ξ_{e2}, u1=δ_{e2}).

B.4.1 Elements of the GIM

m11 =m2*u4^2*a222+m2*u3^2*a112+m1*u2^2*a221 +m2*l1*l2^2+m1*u1^2*a111+Ia+IL+m2*u1^2*l2+m2 *l1^2*l2+1/3*m2*l2^3+1/3*m1*l1^3+mL*l1^2 +mL*u1^2+mL*l2^2+mL*u3^2+ma*l1^2+ma*u1^2+2 *m2*u3*u4*a122+2*m1*u1*u2*a121+2*m2*u1*c22* a12*u3+2*m2*u1*c22*a22*u4-2*m2*l1*s22*a12*u3- 2*m2*l1*s22*a22*u4+2*mL*c22*l1*l2+2*mL*c22*u 1*u3-2*mL*l1*u3*s22;

m12 = m2*(1/3*l2^3+u3^2*a112+2*u3*u4*a122 +u4^2*a222+(1/2*l1*l2^2+u1*(a12*u3+a22*u4))*c22 +(1/2*u1*l2^2-l1*(a12*u3+a22*u4))*s22)+mL*(l2^2+ u3^2+(l1*l2+u1*u3)*c22+(u1*l2-l1*u2)*s22)+IL;

m13 = m2*l1*l2+1/2*m2*l2^2*c22-m2*s22*a12*u3 -m2*s22*a22*u4+m1*b11+mL*l1+mL*l2*c22-mL*u3 *s22+ma*l1;

m14 =1/3*m2*l2^3+m2*u3^2*a112+2*m2*u3*u4 *a122+m2*u4^2*a222+1/2*m2*c22*l1*l2^2+m2*u1* c22*a12*u3+m2*u1*c22*a22*u4+1/2*m2*s22*u1*l2^ 2-m2*l1*s22*a12*u3-m2*l1*s22*a22*u4+m1*b21 +mL*l2^2+mL*u3^2+mL*c22*l1*l2+mL*c22*u1*u3 +mL*s22*u1*l2-mL*s22*l1*u2+IL+Ia;

m15 = m2*((l1*c22+u1*s22)*a12+b12)+mL*(l1*c22 +u1*s22+l2);

m16 = m2*((l1*c22+u1*s22)*a12+b22)+IL;

m22 = m2*(1/3*l2^3+u3^2*a112+2*u3*u4*a122 +u4^2*a222)+mL*(l2^2+u3^2)+IL;

m23 = m2*(1/2*l2^2*c22-(a12*u3+a22*u4)*s22)+mL *(l2*c22-u3*s22);

m24 = m2*(1/3*l2^3+u3^2*a112+2*u3*u4*a122 +u4^2*a222)+mL*(l2^2+u3^2)+IL;

m25 = m2*b12+mL*l2;

m26 = m2*b22+IL;

m33 = m2*l2+m1*a111+mL+ma;

m34 =1/2*m2*l2^2*c22-m2*s22*a12*u3-m2*s22 *a22*u4+m1*a121+mL*l2*c22-mL*u3*s22;

```
m35 = m2*a12*c22+mL*u3*c22;

m36 = m2*a22*c22;

m44 = 1/3*m2*l2^3+m2*u3^2*a112+2*m2*u3*u4
*a122+m2*u4^2*a222+m1*a221+mL*l2^2+mL*u3^2
+IL+Ia;

m45 = m2*b12+mL*l2;

m46 = m2*b22+IL;

m55 = m2*a112+mL;

m56 = m2*a122;

m66 = m2*a222+IL;
```

B.4.2 Coriolis and Centrifugal Terms

```
h1=-2*t1d*u2d*mL*s22*l1*l2-2*t1d*u2d*m2*l1*c22
*a12*u3-2*t1d*u2d*mL*s22*u1*u3+2*t1d*u1d*m2
*c22*a12*u3+2*t1d*u1d*m2*c22*a22*u4+2*t1d*u3d
*mL*c22*u1-2*t1d*u3d*mL*l1*s22-2*t1d*u2d*mL*l
1*u3*c22-2*t1d*u3d*m2*l1*s22*a12-2*t1d*u2d*m2*
u1*s22*a12*u3-2*t1d*u2d*m2*l1*c22*a22*u4+2*t1d
*u1d*mL*u3*c22+2*t1d*u3d*m2*u3*a112+2*t1d*u2
d*m1*u1*a121+2*t1d*u2d*m1*u2*a221+2*t1d*u1d*
m1*u2*a121+2*t1d*u4d*m2*u4*a222+2*t1d*u4d*m2
*u3*a122+2*t1d*u1d*m1*u1*a111+2*t1d*u3d*m2*u
4*a122+2*t1d*u3d*mL*u3+2*t1d*u1d*m2*u1*l2+2
*t1d*u1d*mL*u1+2*t1d*u1d*ma*u1+2*t2d*u3d*mL*
u3+2*t2d*u4d*m2*u4*a222+2*t2d*u4d*m2*u3*a122
+2*t2d*u3d*m2*u4*a122+2*t2d*u3d*m2*u3*a112+2
*u2d*u4d*m2*u4*a222+2*u2d*u4d*m2*u3*a122+2*
u2d*u3d*m2*u4*a122+2*u2d*u3d*mL*u3+2*u2d*u3
d*m2*u3*a112-2*t1d*u2d*m2*u1*s22*a22*u4+2*t1d
*u4d*m2*u1*c22*a22+2*t1d*u3d*m2*u1*c22*a12-2*
t1d*t2d*m2*l1*c22*a22*u4-2*t1d*t2d*mL*s22*l1*l2
-t2d^2*mL*s22*l1*l2-2*t2d*u2d*mL*s22*u1*u3-t2d*
u4d*m2*l1*s22*a22-t2d*u2d*m2*s22*l1*l2^2-2*t2d*
u2d*mL*s22*l1*l2+t2d*u4d*m2*u1*c22*a22-t2d^2*
mL*s22*u1*u3-t2d*u2d*mL*l1*s22-t2d^2*m2*u1*s2
2*a12*u3+t2d^2*mL*c22*u1*l2-t2d^2*mL*c22*l1
*u2-2*t1d*t2d*mL*l1*u3*c22-2*t1d*t2d*m2*l1*c22*
a12*u3-2*t1d*t2d*m2*u1*s22*a12*u3-t2d^2*m2*l1*c
22*a22*u4-t2d^2*m2*u1*s22*a22*u4-2*t1d*t2d*mL*
s22*u1*u3-1/2*t2d^2*m2*s22*l1*l2^2+1/2*t2d^2*m2
*c22*u1*l2^2-t2d^2*m2*l1*c22*a12*u3+t2d*u2d*m2
*c22*u1*l2^2-2*t2d*u3d*m2*l1*s22*a12-2*t2d*u2d*
m2*l1*c22*a12*u3-2*t2d*u2d*m2*l1*c22*a22*u4-u1
d*m2*a22*s22*u4d+2*u2d*u3d*mL*c22*u1
-2*u2d*u3d*m2*l1*s22*a12-u2d^2*m2*l1*c22*a22*u
4-1/2*u2d^2*m2*s22*l1*l2^2+1/2*u2d^2*m2*c22*u1
*l2^2+u2d^2*mL*c22*u1*l2-u2d^2*mL*c22*l1*u2-2
*t1d*t2d*m2*u1*s22*a22*u4-2*t1d*u4d*m2*l1*s22*
a22-2*t2d*u2d*m2*u1*s22*a12*u3-2*t2d*u2d*m2
*u1*s22*a22*u4-u2d^2*mL*l1*s22+2*t2d*u2d*mL
*c22*u1*l2-2*t2d*u2d*mL*c22*l1*u2+2*t2d*u3d
*m2*u1*c22*a12+2*t2d*u3d*mL*c22*u1-u2d*u4d
*m2*l1*s22*a22+u2d*u4d*m2*u1*c22*a22+2*u2d*u
3d*m2*u1*c22*a12-u2d^2*m2*u1*s22*a12*u3-u2d^2
*m2*u1*s22*a22*u4-u2d^2*m2*l1*c22*a12*u3-u2d^
2*mL*s22*l1*l2-u4d*u2d*m2*l1*s22*a12-u4d*t2d
*m2*l1*s22*a12-u3d*t2d*mL*l1*s22-u3d*u2d*mL
*l1*s22+u4d*t2d*m2*u1*c22*a12-u2d^2*mL*s22*u1
*u3+u4d*m2*s22*a12*u1d+u4d*u2d*m2*u1*c22
*a12;

h2 = 2*t1d*u4d*m2*u4*a222+2*t1d*u4d*m2*u3
*a122+2*t1d*u3d*m2*u4*a122+2*t1d*u3d*m2*u3
*a112+2*t1d*u3d*mL*u3+2*t1d*u1d*m2*c22*a22
*u4-t1d*u4d*m2*l1*s22*a22+t1d*u4d*m2*u1*c22*a2
2+2*t2d*u3d*mL*u3+2*t2d*u4d*m2*u3*a122+2*t2d
*u4d*m2*u4*a222+2*t2d*u3d*m2*u3*a112+2*t2d
*u3d*m2*u4*a122+2*u2d*u4d*m2*u4*a222+2*u2d*
u4d*m2*u3*a122+2*u2d*u3d*mL*u3+2*u2d*u3d
*m2*u3*a112+2*u2d*u3d*m2*u4*a122+t1d*u3d*mL
*l1*s22+2*t1d*u1d*m2*c22*a12*u3+2*t1d*u1d*mL*
u3*c22-u1d*m2*a22*s22*u4d-t1d*u2d*mL*l1*s22
+2*t1d*u1d*mL*l2*s22+t1d*u1d*m2*l2^2*s22-u1d
*u3d*mL*s22+t1d^2*m2*u1*s22*a12*u3+t1d^2*m2*
u1*s22*a22*u4+t1d^2*m2*l1*c22*a12*u3+t1d^2*m2
*l1*c22*a22*u4+t1d^2*mL*s22*l1*l2+t1d^2*mL*s22
*u1*u3+t1d^2*mL*l1*u3*c22+t1d*m2*a12*u4d*l1
*s22-t1d*m2*a12*u4d*u1*c22+u1d*u3d*mL*u3*s22-
u1d*m2*a22*c22*u4d;

h3 =-t1d^2*mL*u1-t1d^2*ma*u1+mL*c22*u3d^2
-u2d^2*m2*c22*a22*u4-t1d*u2d*m2*l2^2*s22-2*t1d
*u2d*m2*c22*a12*u3-2*t1d*u2d*m2*c22*a22*u4-2*
t1d*u2d*mL*l2*s22-2*t1d*u2d*mL*u3*c22-2*t1d*u3
d*m2*s22*a12-2*t1d*u3d*mL*s22-t1d*m2*a22*s22*
u4d-2*t2d*u2d*mL*l2*s22-t1d^2*m2*c22*a12*u3-t1d
^2*m2*c22*a22*u4-t1d^2*mL*u3*c22-u3d*t2d*mL*
u3*s22-u3d*u2d*mL*u3*s22+u4d*m2*a22*c22*t2d+
u4d*m2*a22*c22*u2d-2*t2d*u2d*mL*u3*c22-2*t2d*
u3d*m2*s22*a12-t2d*u3d*mL*s22-t2d*m2*a22*s22*
u4d-1/2*u2d^2*m2*l2^2*s22-u2d^2*m2*c22*a12*u3-
t1d*t2d*m2*l2^2*s22-2*t1d*t2d*m2*c22*a12*u3-2
*t1d*t2d*m2*c22*a22*u4-2*t1d*t2d*mL*l2*s22
-2*t1d*t2d*mL*u3*c22-t1d^2*m1*u1*a111-t1d^2*m2
*u1*l2-t1d^2*m1*u2*a121-u2d^2*mL*l2*s22-u2d^2*
mL*u3*c22-2*u2d*u3d*m2*s22*a12-u2d*u3d*mL
*s22-u2d*m2*a22*s22*u4d-1/2*t2d^2*m2*l2^2*s22
-t2d^2*m2*c22*a12*u3-t2d^2*m2*c22*a22*u4-t2d^2
```

```
*mL*l2*s22-t2d^2*mL*u3*c22-t2d*u2d*m2*l2^2*s22
-2*t2d*u2d*m2*c22*a12*u3-2*t2d*u2d*m2*c22*a22
*u4-m2*s22*a12*t1d*u4d;

h4 = 2*t1d*u4d*m2*u4*a222+2*t1d*u4d*m2*u3
*a122+2*t1d*u3d*m2*u4*a122+2*t1d*u3d*m2*u3*a
112+2*t1d*u3d*mL*u3+2*t1d*u1d*m2*c22*a22*u4
-t1d*u4d*m2*l1*s22*a22+t1d*u4d*m2*u1*c22*a22-3
/2*t1d^2*m1*u2^2*a221-t1d^2*m1*u1*a121+t1d*t2d
*mL*l1*s22+2*t2d*u3d*mL*u3+2*t2d*u4d*m2*u3
*a122+2*t2d*u4d*m2*u4*a222+2*t2d*u3d*m2*u3
*a112+2*t2d*u3d*m2*u4*a122+2*u2d*u4d*m2*u4
*a222+2*u2d*u4d*m2*u3*a122+2*u2d*u3d*mL*u3
+2*u2d*u3d*m2*u3*a112+2*u2d*u3d*m2*u4*a122
+t1d*u3d*mL*l1*s22+2*t1d*u1d*m2*c22*a12*u3+2*
t1d*u1d*mL*u3*c22-u1d*m2*a22*s22*u4d+2*t1d*u1
d*mL*l2*s22+t1d*u1d*m2*l2^2*s22-u1d*u3d*mL*s2
2+t1d^2*m2*u1*s22*a12*u3+t1d^2*m2*u1*s22*a22*
u4+t1d^2*m2*l1*c22*a12*u3+t1d^2*m2*l1*c22*a22*
u4+t1d^2*mL*s22*l1*l2+t1d^2*mL*s22*u1*u3+t1d^
2*mL*l1*u3*c22+t1d*m2*a12*u4d*l1*s22-t1d*m2
*a12*u4d*u1*c22+u1d*u3d*mL*u3*s22-u1d*m2*a22
*c22*u4d;

h5 =-t1d^2*mL*u3-t1d*t2d*mL*l1*s22-t2d^2*mL
*u3-u2d^2*mL*u3+2*t1d*u1d*mL*s22+2*t1d*u1d
*m2*s22*a12+u1d*u2d*mL*s22+t2d*u1d*mL*s22-
t1d^2*m2*u1*c22*a12+t1d^2*m2*l1*s22*a12-t1d^2
*mL*c22*u1+t1d^2*mL*l1*s22-u1d*t2d*mL*u3*s22
-u1d*u2d*mL*u3*s22-u2d^2*m2*u3*a112-u2d^2*m2
*u4*a122-2*t1d*t2d*mL*u3-2*t1d*u2d*m2*u3*a112
-2*t1d*u2d*m2*u4*a122-2*t1d*u2d*mL*u3-t2d^2
*m2*u3*a112-t2d^2*m2*u4*a122-2*t2d*u2d*m2*u3
*a112-2*t2d*u2d*m2*u4*a122-2*t2d*u2d*mL*u3
-t1d^2*m2*u3*a112-t1d^2*m2*u4*a122-2*t1d*t2d
*m2*u3*a112-2*t1d*t2d*m2*u4*a122-t1d*u2d*mL*
l1*s22;

h6 = t1d*u1d*m2*s22*a12+t1d*u2d*m2*u1*c22*a12
+u1d*m2*a22*s22*u2d-t1d^2*m2*u1*c22*a22+t1d^2
*m2*l1*s22*a22-t1d*m2*t2d*u1*a22*c22+t1d*m2
*t2d*l1*a22*s22-t1d*u2d*m2*u1*c22*a22+t1d*u2d*
m2*l1*s22*a22+u1d*m2*a22*c22*t2d+u1d*m2*a22*
c22*u2d-m2*t2d^2*u3*a122-m2*t2d^2*u4*a222-2*t2
d*u2d*m2*u3*a122-2*t2d*u2d*m2*u4*a222-u2d^2*
m2*u3*a122-u2d^2*m2*u4*a222-t1d*t2d*m2*l1*s22
*a12+t1d*t2d*m2*u1*c22*a12-t1d*u2d*m2*l1*s22
*a12+t2d*m2*a22*s22*u1d-t1d^2*m2*u3*a122-2*t1d
*m2*t2d*u3*a122-2*t1d*m2*t2d*u4*a222-2*t1d*u2d
*m2*u3*a122-2*t1d*u2d*m2*u4*a222-t1d^2*m2*u4*
a222+t1d*m2*a22*s22*u1d;
```

4.3 Stiffness Terms

```
k1 = 0;
k2 = 0;

k3 = EI1*(k111*u1+k121*u2);
k4 = EI1*(k121*u1+k221*u2);
k5 = EI2*(k112*u3+k122*u4);
k6 = EI2*(k122*u3+k222*u4);
```

Appendix C

```
%-----------------------------------------------------------
% NAME: Model.m
%
% DESCRIPTIONS:
%     This is the main program that calls the
%     different functions neeeded to simulate the
%     motion of a the one Link Flexible Manipulator
%     with 3 DOFs.
%
% FUNCTIONS:
%     robot.m:    Robot and Load Characteristics
%     gains.m:    Controller Gains
%     xytraj1.m:  Desired trajectory in the XY plane
%     rk_45.m:    Numerical Integration
%     (RUNGE-KUTTA 4th order)
%
% LANGUAGE:
%     386-MATLAB                  Version 3.5j
%
% Written By: Amor Jnifene, University of Ottawa,
% June 1992.
%-----------------------------------------------------------

% global variables
global m1 l1 EI1 ML JL DAMP
global TAU x0 II
global fi1 fi2 dist DELAY thetax thetadx S
global KP KD
global theta thetad thetadd step THETAF T

% Get robot characteristics
robot;

omega=30; zita=1.;

% Get gains
[KP KD]=gains(omega,zita)

dist=0.4;
fi1 = 3*(dist/l1)^2-2*(dist/l1)^3;
fi2 = -dist^2/l1+dist^3/l1^2;
step=0.001;
T=1.0;
tfinal=2.0*T;
THETAF=pi/2;
DELAY=54;
S=0;

[t,theta,thetad,thetadd]=traj2(tfinal);
TAU=[0 0 0]';
t0=0;
x0=[ 0 0 0 0 0 0]';
[torque,t,x]=rk_45('system',t0,tfinal,step,x0);

%======== End of model.m===============


function xdot=system(t,x)
%-----------------------------------------------------------
%  NAME:
%     system.m
%
%  DESCRIPTION:
%    This program contains the function to be
%    integrated using the Range Kutta routine
%    (rk_45.m).
%
%  INPUTS:
%     t: time.
%     x: Vector containing the joint angle and its
%        first time derivative, the deflection and
%        slope at the end of the link and their
%        first time derivatives.
%
%  OUTPUTS:
%     System of first order differential equations.
%
%  GLOBAL VARIABLES:
%     m1: Mass of the link.
%     l1: Length of the first.
%     ML: Mass of the payload.
%     JL: Moment of Inertia of the payload..
%     KP: Proportional Gains.
```

```
%      KD: Derivative Gains.
%
%  LANGUAGE
%  386-MATLAB                    Version 3.5j
%
%  Written by: Amor Jnifene, University of Ottawa,
%   June 1992.
%--------------------------------------------------
t1=x(4);u1=x(5); u2=x(6);
t1d=x(1);u1d=x(2); u2d=x(3);
q=x([4:6]);qd=x([1:3]);
%--------------------------------------
a111 = 13/35*l1; a121 = -11/210*l1^2;
a221 = 1/105*l1^3;
b11 = 7/20*l1^2; b21 = -1/20*l1^3;
k111 = 12/l1^3; k121 = -6/l1^2; k221 = 4/l1;


n11 = m1*(1/3*l1^3+u1^2*a111+2*u1*u2*a121+u2^2
*a221);
n12 = m1*b11;
n13 = m1*b21;
n22 = m1*a111;
n23 = m1*a121;
n33 = m1*a221;

MLINK=[n11 n12 n13
      n12 n22 n23
      n13 n23 n33];

J11 = ML*(l1^2+u1^2)+JL;
J12 = ML*l1;
J13 = JL;
J22 = ML;
J23 = 0;
J33 = JL;

MLOAD=[J11 J12 J13
      J12 J22 J23
      J13 J23 J33];

h1 = 2*m1*(u1d*u1*a111+u1d*u2*a121...
+u2d*u1*a121+u2d*u2*a221)*t1d...
+2*u1d*ML*u1*t1d;
h2 = -(m1*(u1*a111+u2*a121)+ML*u1)*t1d^2;
h3 = -m1*(u1*a121+u2*a221)*t1d^2;

k1 = 0;
k2 = EI1*(k111*u1+k121*u2);
k3 = EI1*(k121*u1+k221*u2);
%--------------------------------------
A=MLINK+MLOAD;
%--------------------------------------
HVEC=[h1 h2 h3]';
DVEC=DAMP*qd;
%--------------------------------------
KVEC=[k1 k2 k3]';
%--------------------------------------
%Partition of the matrix A

MR=A(1,1);
MRF=[A(1,2) A(1,3)];
MF=[A(2,2) A(2,3)
   A(3,2) A(3,3)];
%--------------------------------------------------
IMR=1/MR;
IMF=inv(MF);
%--------------------------------------------------
%Partition of the vectors HVEC and KVEC
HR=HVEC(1);
GR=k1;
HF=[HVEC(2) HVEC(3)]';
GF=[k2 k3]';
%--------------------------------------------------
% System for the control of the angle of a point along
% the link
PHI=[fi1 fi2]/dist;
DEL=MF-MRF*PHI;
IDEL=inv(DEL);

GAMMARX=MR-(MRF-MR*PHI)*IDEL*MRF';
LAMDARX=HR+GR-(MRF-MR*PHI)*IDEL...
*(HF+GF);
%--------------------------------------------------
thetax(II)=(fi1*u1+fi2*u2)/dist;
thetadx(II)=(fi1*u1d+fi2*u2d)/dist;
%--------------------------------------------------
CON=1;
D=1+2*CON+CON^2;
A1=1/D;
A2=2*CON/D;
A3=CON^2/D;

if II<=DELAY,

TAUR=thetadd(II)+KP*(theta(II)-(q(1)+thetax(II)))...
   +KD*(thetad(II)-(qd(1)+thetadx(II)));
else
if S==1;
 TAUR=thetadd(II)+KP*(theta(II)-(q(1)...
 +A1*thetax(II)+A2*thetax(II-DELAY)...
 +A3*thetax(II-2*DELAY)))...
 +KD*(thetad(II)-(qd(1)+A1*thetadx(II)...
 +A2*thetadx(II-DELAY)...
 +A3*thetadx(II-2*DELAY)));
else
```

```
    TAUR=thetadd(II)...
    +KP*(theta(II)-(q(1)+thetax(II-DELAY)))...
    +KD*(thetad(II)-(qd(1)+thetadx(II-DELAY)));
  end;
  end;

TAU(1)=GAMMARX*TAUR+LAMDARX;
%--------------------------------------------------
B = -HVEC-KVEC-DVEC+TAU;
%--------------------------------------------------
qdd=A\B;
%--------------------------------------------------
xdot=[qdd',[qd]']';

%=== End of the function system.m =========


function robot
%--------------------------------------------------
%Link length
l1=1.0;

%Link mass
m1=1.0;

%Link stiffness
EI1=100.0;

% Load mass
ML=0.0;

%Load inertia
JL=0.0;

% Damping
DAMP=[0 0 0;0 0 0;0 0 0];

%====== End of function robot============


function [KP,KD]=gains(omega,zita)
%--------------------------------------------------
%  NAME:
%     gains.m
%
%  DESCRIPTION
%     Calculates the proportional and derivative
%     gains of the PD Controller. These gains are
%     calculated based on the highest inertia
%     configuration.
%
%  INPUTS
%     omega: Desired natural frequency.
%     zita: Desired damping ratio.
%
%  OUTPUTS
%     KP: Proportional gain.
%     KD: Derivative gain.
%
%  GLOBAL VARIABLES
%     m1: Mass of the link.
%     l1: Length of the first.
%     ML: Mass of the payload.
%     JL: Moment of inertia of the payload.
%
%  LANGUAGE
%     386-MATLAB                    Version 3.5j
%
%  Written by: Amor Jnifene, University of Ottawa,
%  June 1992.
%--------------------------------------------------
n11 = m1*1/3*l1^3;
J11 = ML*l1^2+JL;
INERTIA=n11+J11;
KP=omega^2*INERTIA;
KD=2*zita*omega*INERTIA;

%========End of gains.m ============


function [t,theta,thetad,thetadd]=traj1(tfinal)
%--------------------------------------------------
%  NAME:
%     traj1.m
%
%  DESCRIPTION
%     Generates a desired joint trajectory based on
%     step input.
%
%
%  INPUTS
%    THETAF: Desired final position
%    tfinal: Duration.
%    step: Time step.
%
%  OUTPUTS
%     theta:  Joint position.
%     thetad: Joint velocity.
%     thetadd: Joint acceleration.
%
%  LANGUAGE
%     386-MATLAB                    Version 3.5j
%
%--------------------------------------------------
for i=1:tfinal/step+1,
```

```
        theta(i)=THETAF;
        thetad(i)=0;
        thetadd(i)=0;
        t(i)=(i-1)*step;
end
%========= End of traj1.m =============
```

```
function [t,theta,thetad,thetadd]=traj2(tfinal)
%--------------------------------------------------------
% NAME:
%     traj2.m
%
% DESCRIPTION
%     Generates a desired trajectory with a
%     Bang-Bang acceleration profile.
%
% INPUTS
%    a: Acceleration.
%    THETAF: Desired final position
%    tfinal: Duration.
%    step: Time step.
%
% OUTPUTS
%     theta:  Joint position.
%     thetad: Joint velocity.
%     thetadd: Joint acceleration.
%
% LANGUAGE
%     386-MATLAB                    Version 3.5j
%--------------------------------------------------------
a=4*THETAF/T^2;
for i=1:tfinal/step+1,
tt=(i-1)*step;
if tt <= T,
  if tt<=T/2,
        theta(i)=a*tt^2/2;
        thetad(i)=a*tt;
        thetadd(i)=a;
  else
        theta(i)=-a*(T-tt)^2/2+a*T^2/4;
        thetad(i)=a*(T-tt);
        thetadd(i)=-a;
  end
else
  theta(i)=THETAF;
  thetad(i)=0;
  thetadd(i)=0;
end
t(i)=tt;
end

%========= End of traj2.m =============
```

```
function [t,theta,thetad,thetadd]=traj3(tfinal)
%--------------------------------------------------------
%  NAME:
%     traj3.m
%
%  DESCRIPTION
%     Generates a desired trajectory with a
%     Sinusoidal acceleration profile.
%
%  INPUTS
%
%    THETAF: Desired final position
%    tfinal: Duration.
%    step: Time step.
%
%  OUTPUTS
%     theta:  Joint position.
%     thetad: Joint velocity.
%     thetadd: Joint acceleration.
%
%  LANGUAGE
%     386-MATLAB                    Version 3.5j
%--------------------------------------------------------

for i=1:tfinal/step+1,
tt=(i-1)*step;
if tt <= T,

theta(i)=THETAF*tt/T-THETAF/(2*pi)*sin(2*pi*tt/T);
  thetad(i)=THETAF/T-THETAF/T*cos(2*pi*tt/T);
  thetadd(i)=(2*pi/T)*THETAF/T*sin(2*pi*tt/T);
else
  theta(i)=THETAF;
  thetad(i)=0;
  thetadd(i)=0;
end
t(i)=tt;
end

%========= End of traj3.m =============
```

```
function [t,theta,thetad,thetadd]=traj4(tfinal)
%--------------------------------------------------------
% NAME:
%     traj4.m
%
% DESCRIPTION
%     Generates a desired trajectory with a
%     Gaussian velocity profile.
%
% INPUTS
%
```

```
%     THETAF: Desired final position
%     tfinal: Duration.
%     step: Time step.
%
%  OUTPUTS
%      theta:  Joint position.
%      thetad: Joint velocity.
%      thetadd: Joint acceleration.
%
%  LANGUAGE
%      386-MATLAB                  Version 3.5j
%-----------------------------------------------------------

mu=1.0;
s=1/(9*pi)^.5,
a=1/(s*(2*pi)^.5)*THETAF,
b=1/(2*s^2),
for i=1:tfinal/step+1,
tt=(i-1)*step;

theta(i)=pi^.5*a/(2*b^.5)*(erf(b^.5*(tt-mu))-erf(-b^.5*
mu));
      thetad(i)=a*exp(-b*(tt-mu)^2);
      thetadd(i)=-2*a*b*(tt-mu)*exp(-b*(tt-mu)^2);
t(i)=tt;
end

%========= End of traj4.m ==============
```

```
%--------------------------------------------------------
% NAME: Model.m
%
% DESCRIPTIONS:
%      This is the main program that calls the
%      different functions needed to simulate the
%      motion of a Two-Link Flexible Manipulator
%      with 6 DOFs.
%
% FUNCTIONS:
%      robot.m:   Robot and Load Characteristics
%      gains.m:   Controller Gains
%      xytraj1.m: Desired trajectory in the XY plane
%      invkin.m:  Inverse Kinematics
%      forkin.m:  forward Kinematics
%      rk_45.m:   Numerical Integration
%      (RUNGE-KUTTA 4th order)
%
% LANGUAGE:
%      386-MATLAB                Version 3.5j
%
% Written By: Amor Jnifene, University of Ottawa,
% September 1994.
%--------------------------------------------------------
% --- Links characteristics ---------------------------

global l1 l2 m1 m2 EI1 EI2 DAMP

% --- Load characteristics ----------------------------
global ML JL Ma Ja

% --- Non-Collocation Parameters ----------------------
global DIST1 DIST2
global FI11 FI12 FI21 FI22

% --- Delay  Parameters -------------------------------
global IID1 IID2 UX1 UX2 UX1D UX2D

%---- Controller parameters --------------------------
global KP KD

%---- Global variables for the desired trajectory in
%      joint space -----------------------------------
global theta thetad thetadd T

%------------------------------------------------------
global step II TAU

%------------------------------------------------------
%   Get robot characteristics robot

%--- Shape Functions ---------------------------------
DIST1=0.8;
DIST2=0.4;

FI11=3*(DIST1/l1)^2-2*(DIST1/l1)^3;
FI12=-DIST1^2/l1+DIST1^3/l1^2;
FI21=3*(DIST2/l2)^2-2*(DIST2/l2)^3;
FI22=-DIST2^2/l2+DIST2^3/l2^2;

IID1=50;
IID2=50;
%------------------------------------------------------
%   Get controller gains

   omega1=30; zita1=1.0;
   omega2=30; zita2=1.0;
   [KP,KD]=gains(omega1,omega2,zita1,zita2);

step=0.001;

TRAJ=2;
if TRAJ==1,
   T=2.0;
   tfinal=1.5*T;
   PI=[0.8  0.1]; % Starting Point
   PF=[0.1  1.6]; % Finish Point
   [t,xye,xyed,xyedd]=xytraj1(PI,PF,tfinal);

 % Generate the desired trajectories in joint space
   [theta,thetad,thetadd]=invkin(t,xye,xyed,xyedd);
   plot(t,theta,t,thetad,t,thetadd);

end

if TRAJ==2,
   T=2.0;
   tfinal=2.5*T;
   POINTS=[0.8 0.1;1.3 0.6;0.8 1.1;0.3 0.6];
   [t,xye,xyed,xyedd]=xytraj2(POINTS,tfinal);

% Generate the desired trajectories in joint space
   [theta,thetad,thetadd]=invkin(t,xye,xyed,xyedd);
   plot(t,theta,t,thetad,t,thetadd);
end
%------------------------------------------------------
% Initial Conditions
TAU=[0 0 0 0 0 0]';
x0=[thetad(1,1) thetad(1,2) 0 0 0 0 theta(1,1)
theta(1,2) 0 0 0 0]';
t0=0.0;

%------------------------------------------------------
% Start the simulation of the robot motion
[t,x,torque]=RK_45('system',t0,tfinal,step,x0);
```

```
% Find the actual position of the end point in the xy
% coordinates
[XYEAC XYEACD]=forkin(t,x);

%======= End of model.m ===============


function [XYE2,XYE2D] = forkin(t,x)
%----------------------------------------------------
% NAME:
%    forkin.m: Forward Kinematics.
%
% DESCRIPTION:
%    Calculates the position and velocity of the end
%    point of a Two-Link Flexible Manipulator.
%
% INPUTS:
%
%    t: Time vector
%    x: Matrix containing:
%      - The joint angles and their corresponding
%        first time derivatives.
%      - Deflection and Slope at the end of the
%        first link and their corresponding first %
%        time derivatives.
%      - Deflection and slope at the end of the
%        second link their corresponding first time
%        derivatives.
%
% GLOBAL VARIABLES:
%    l1: Length of the first link
%    l2: Length of the second link
%
% OUTPUTS:
%    XYE2: X and Y cordinates of the end point
%    position of the second link
%    XYE2D: X and Y cordinates of the end point
%    velocity of the second link
%
% LANGUAGE:
%    386-MATLAB                 Version 3.5j.
%
% Written by:   Amor Jnifene, Ottawa University,
% June, 1994.
%
%----------------------------------------------------
N=max(size(t));

for i=1:N,

    T1=x(i,7); T2=x(i,8);
    u1=x(i,9); u2=x(i,10); u3=x(i,11); u4=x(i,12);
    T1D=x(i,1); T2D=x(i,2);
    u1d=x(i,3); u2d=x(i,4); u3d=x(i,5); u4d=x(i,6);

    C1=cos(T1);
    S1=sin(T1);
    C2=cos(T2+u2);
    S2=sin(T2+u2);

    R01=[ C1  -S1
          S1   C1 ];

    R12=[ C2 -S2
          S2  C2 ];

    R01D=[ -S1  -C1
            C1  -S1 ]*T1D;

    R12D=[ -S2  -C2
            C2  -S2 ]*(T2D+u2d);

    XYE2(i,1:2) = (R01*([l1 u1]'+R12*[l2 u3]'))';
    TEMP1 = [l1 u1]'+R12*[l2 u3]';
    TEMP2 = [0 u1d]'+R12D*[l2 u3]'...
             +R12*[0 u3d]';
    XYE2D(i,1:2)=(R01D*TEMP1+R01*TEMP2)';

end

%========== End of forkin.m ===========


function [KP,KD]=gains(omega1,omega2,zita1,zita2)
%----------------------------------------------------
% NAME:
%    gains.m
%
% DESCRIPTION
%    Calculates the proportional and derivative
%    gains of the PD Controller. These gains are
%    calculated based on the highest inertia
%    configuration.
%
% INPUTS
%    omega1: Desired bandwidth of the controller
%            at the first joint.
%    omega2: Desired bandwidth of the controller
%            at the second joint.
%    zita1: Desired damping ratio at the first joint.
%    zita2: Desired damping ratio at the second.
%
% OUTPUTS
%    KP: Diagonal matrix of the proportional gains.
%    KD: Diagonal matrix of the derivative gains.
%
```

```
% GLOBAL VARIABLES
%    m1: Mass of the first link.
%    m2: Mass of the second link.
%    l1: Length of the first.
%    l2: Length of the second.
%    Ma: Mass of the joint actuator at the end of
%        the first link.
%    Ja: Inertia of the joint actuator at the end of
%        the first link.
%    ML: Mass of the payload at the end of the
%        second link.
%    JL: Inertia of the payload at the end of the
%        second link.
%
% LANGUAGE
%    386-MATLAB                Version 3.5j
%
% Written by: Amor Jnifene, University of Ottawa,
% June, 1994.
%-----------------------------------------------------
% Highest inertia with respect to the axis of the first
% joint.
I1=(m2*l2/3+ML)*l2^2+(m1*l1/3+Ma+m2*l2...
+ML)*l1^2+(m2*l2+2*ML)*l1*l2+JL+Ja;

% Highest inertia with respect to the axis of the
% second joint.
I2=(m2*l2/3+ML)*l2^2+JL;

% Proportional Gains.
KP=[omega1^2*I1   0
    0   omega2^2*I2];

% Derivative Gains
KD=[2*omega1*zita1*I1   0
    0                   2*omega2*zita2*I2];

%============== End of gains.m ==============


function [theta,thetad,thetadd] = invkin(t,xye ,xyed,...
xyedd )
%-----------------------------------------------------
% NAME:
%    invkin.m: Inverse Kinematics.
%
% DESCRIPTION:
%    Calculates the positions, velocities and
%    accelerations of the joint angles assuming a
&    rigid twolink manipulator
%
% INPUTS:
%    t:       Time
%    xye:     X and Y positions of the end point
%    xyed:    X and Y velocities of the end point
%    xyedd:   X and Y Accelerations of the end
%             point
%
% GLOBAL VARIABLES:
%    l1: Length of the first link
%    l2: Length of the second link
%
% OUTPUTS:
%    theta:   Positions of joint 1 and joint 2
%    thetad:  Velocities of joint 1 and joint 2
%    thetadd: Accelerations of joint 1 and joint 2
%
% LANGUAGE:
%    386-MATLAB                Version 3.5j.
%
% Written by:   Amor Jnifene, Ottawa University,
% June, 1994.
%-----------------------------------------------------

for i=1:max(size(t)),

C2=(xye(i,1)^2+xye(i,2)^2-(l1^2+l2^2))/(2*l1*l2);

      if (C2>1.0) | (C2<-1.0),
        xye(i,:)
        error(' The above point is out of reach')
      end


S2=-(1-C2^2)^.5;
T2=atan2(S2,C2);
theta(i,2)=T2;
R1=C2*l2+l1;
R2=S2*l2;
T1=atan2(R1*xye(i,2)-R2*xye(i,1),R1*xye(i,1)...
+R2*xye(i,2));
theta(i,1)=T1;
C1=cos(T1);
S1=sin(T1);
T12=T1+T2;
C12=cos(T12);
S12=sin(T12);
JACOB=[-l1*S1-l2*S12  -l2*S12
        l1*C1+l2*C12   l2*C12];
thetad(i,1:2)=(inv(JACOB)*[xyed(i,1) xyed(i,2)]')';
T1D=thetad(i,1);
T2D=thetad(i,2);
T12D=T1D+T2D;
DJCOB=[-l1*C1*T1D-l2*C12*T12D  -l2*C12*T12D
       -l1*S1*T1D-l2*S12*T12D  -l2*S12*T12D];
thetadd(i,1:2)=(inv(JACOB)*([xyedd(i,1)
xyedd(i,2)]'-DJCOB*[T1D T2D]'))';
```

```
end

%======= End of invkin.m ===============

function [tout, yout,torque] = RK_45(FunFcn, t0,...
tfinal,h, y0)
%---------------------------------------------------
% RK45 Integrate a system of ordinary differential
% equations using 4th and 5th order Runge-Kutta
% formulas..
%
% [T,Y] = RK45('yprime', T0, Tfinal, Y0) integrates
%          the system of ordinary differential
%          equations described by the M-file
%          YPRIME.M over the interval T0 to Tfinal
%          and using initial conditions Y0.
%
% INPUT:
% F   - String containing name of user-supplied
%       problem description.
%       Call: yprime = fun(t,y) where F = 'fun'.
%        t       - Time (scalar).
%        y       - Solution column-vector.
%        yprime - Returned derivative column-vector;
%        yprime(i) = dy(i)/dt.
%        t0    - Initial value of t.
%        tfinal- Final value of t.
%        y0    - Initial value column-vector.
%
% OUTPUT:
% T  - Returned integration time points (row-vector).
% Y  - Returned solution, one solution column-vector
%       per tout-value.
%
% The result can be displayed by: plot(tout, yout).
% The Fehlberg coefficients:

alpha = [1/4  3/8  12/13  1  1/2]';
beta = [...
 [ 1      0      0      0      0      0] /4
 [ 3      9      0      0      0      0] /32
 [ 1932  -7200   7296    0      0      0] /2197
 [ 8341 -32832  29440  -845     0      0] /4104
 [-6080  41040 -28352  9295  -5643    0] /2052]';

gamma = [...
[902880  0  3953664  3855735  -1371249  277020] /
7618050
[ -2090  0   22528   21970   -15048   -27360]
/752400 ]';

pow = 1/5;
% Initialization
II=1;
t=t0;
torque=[TAU(1) TAU(2)];
y = y0(:);
f = y*zeros(1,6);
tout = t;
yout = y.';
% The main loop
while (t < tfinal)

% Compute the slopes
temp = feval(FunFcn,t,y);
f(:,1) = temp(:);
for j = 1:5
temp = feval(FunFcn, t+alpha(j)*h, y+h*f*beta(:,j));
f(:,j+1) = temp(:);
end
% Update the solution
 t = t + h;
 y = y + h*f*gamma(:,1);
 tout = [tout; t];
 yout = [yout; y.'];
 torque=[torque; [TAU(1) TAU(2)]];
II=II+1;
end;

%========== End of rk45.m ============

function robot
%---------------------------------------------------
%        This function contains the physical
%         parameters of the two link flexible
%         manipulator.
%---------------------------------------------------

% Length of the links  (m)
l1=1.0; l2=1.0;

% Masses of the links (kg/m)
m1=1.0; m2=1.0;

% Flexural regidities  (N-m^2)
EI1=200.0; EI2=200.0;

% Payload Inertia (kg.m^2)
JL=0.0;

%Payload mass (kg)
ML=0.0;

% Second Joint hub mass (kg)
Ma= 1.0;
```

```
% Second Joint hub inertia (kg.m^2)
Ja=0.1;

% Damping
DAMP=[0  0  0    0    0    0
      0  0  0    0    0    0
      0  0  0    0    0    0
      0  0  0    0    0    0
      0  0  0    0    0    0
      0  0  0    0    0    0]

%======== End of robot.m ==============


function [t,xye,xyed,xyedd]=xytraj1(PI,PF,tfinal)
%----------------------------------------------------------
%  NAME:
%     xytraj1.m
%
%  DESCRIPTION
%     Generates a desired trajectory for the end
%     point of the two link flexible manipulator.
%
%       - Staright line path.
%       - Sinusoidal acceleration profile.
%
%  INPUTS
%
%    PI: starting point in the XY plane.
%    PF: Finishing point in the XY plane.
%    tfinal: Duration.
%    step: Time step.
%
%  OUTPUTS
%     xye:  Desired position of the end point in the
%           XY plane position.
%    xyed: Desired velocity  of the end point in the
%           XY plane.
%    xyedd: Desired acceleration of the end point in
%           the XY plane thetadd.
%
%  LANGUAGE
%     386-MATLAB                   Version 3.5j
%----------------------------------------------------------
XI=PI(1);
YI=PI(2);
XF=PF(1);
YF=PF(2);

for i=1:tfinal/step+2,
tt=(i-1)*step;
if tt <= T,
  x=XI+(XF-XI)*tt/T-(XF-XI)/(2*pi)*sin(2*pi*tt/T);
  xd=(XF-XI)/T-(XF-XI)/T*cos(2*pi*tt/T);
  xdd=(2*pi/T)*(XF-XI)/T*sin(2*pi*tt/T);

  y=(YI-YF)/(XI-XF)*(x-XI)+YI;
  yd=(YI-YF)/(XI-XF)*xd;
  ydd=(YI-YF)/(XI-XF)*xdd;

else

  x=XF; xd=0; xdd=0;
  y=YF; yd=0; ydd=0;

end
t(i)=tt;
xye(i,1:2)=[x y];
xyed(i,1:2)=[xd yd];
xyedd(i,1:2)=[xdd ydd];
end

%======== End of xytraj1.m ============


function [t,xye,xyed,xyedd]=xytraj2(POINTS,tfinal)
%----------------------------------------------------------
%  NAME:
%     xytraj2.m
%
%  DESCRIPTION
%     Generates a desired trajectory for the end
%     point of the two link flexible manipulator.
%
%       - Four points  path.
%       - Sinusoidal acceleration profile.
%
%  INPUTS
%
%    POINTS: An array of four points in the XY %
%              plane.
%    tfinal: Duration.
%    step: Time step.
%
%  OUTPUTS
%     xye:  Desired position of the end point in the
%           XY plane position.
%    xyed: Desired velocity  of the end point in the
%           XY plane.
%    xyedd: Desired acceleration of the end point in
%           the XY plane thetadd.
%
%  LANGUAGE
%     386-MATLAB                   Version 3.5j
%----------------------------------------------------------
```

```
TCYCLE=tfinal/4;
T=0.8*TCYCLE;
XI=POINTS(1,1); YI=POINTS(1,2);
XF=POINTS(2,1); YF=POINTS(2,2);
for i=1:TCYCLE/step,
tt=(i-1)*step;
  if tt <= T,

x=XI+(XF-XI)*tt/T-(XF-XI)/(2*pi)*sin(2*pi*tt/T);
       xd=(XF-XI)/T-(XF-XI)/T*cos(2*pi*tt/T);
       xdd=(2*pi/T)*(XF-XI)/T*sin(2*pi*tt/T);
       y=(YI-YF)/(XI-XF)*(x-XI)+YI;
       yd=(YI-YF)/(XI-XF)*xd;
       ydd=(YI-YF)/(XI-XF)*xdd;
  else
       x=XF; xd=0; xdd=0;
       y=YF; yd=0; ydd=0;
  end

  t1(i)=tt;
  xy1(i,1:2)=[x y];
  xy1d(i,1:2)=[xd yd];
  xy1dd(i,1:2)=[xdd ydd];
end

XI=POINTS(2,1); YI=POINTS(2,2);
XF=POINTS(3,1); YF=POINTS(3,2);
for i=1:TCYCLE/step,
tt=(i-1)*step;
  if tt <= T,

x=XI+(XF-XI)*tt/T-(XF-XI)/(2*pi)*sin(2*pi*tt/T);
       xd=(XF-XI)/T-(XF-XI)/T*cos(2*pi*tt/T);
       xdd=(2*pi/T)*(XF-XI)/T*sin(2*pi*tt/T);
       y=(YI-YF)/(XI-XF)*(x-XI)+YI;
       yd=(YI-YF)/(XI-XF)*xd;
       ydd=(YI-YF)/(XI-XF)*xdd;
  else
       x=XF; xd=0; xdd=0;
       y=YF; yd=0; ydd=0;
  end

  t2(i)=tt+TCYCLE;
  xy2(i,1:2)=[x y];
  xy2d(i,1:2)=[xd yd];
  xy2dd(i,1:2)=[xdd ydd];
end

XI=POINTS(3,1); YI=POINTS(3,2);
XF=POINTS(4,1); YF=POINTS(4,2);

for i=1:2*TCYCLE/step+2,
tt=(i-1)*step;
  if tt <= T,

x=XI+(XF-XI)*tt/T-(XF-XI)/(2*pi)*sin(2*pi*tt/T);
       xd=(XF-XI)/T-(XF-XI)/T*cos(2*pi*tt/T);
       xdd=(2*pi/T)*(XF-XI)/T*sin(2*pi*tt/T);
       y=(YI-YF)/(XI-XF)*(x-XI)+YI;
       yd=(YI-YF)/(XI-XF)*xd;
       ydd=(YI-YF)/(XI-XF)*xdd;
  else
       x=XF; xd=0; xdd=0;
       y=YF; yd=0; ydd=0;
  end

  t3(i)=tt+2*TCYCLE;
  xy3(i,1:2)=[x y];
  xy3d(i,1:2)=[xd yd];
  xy3dd(i,1:2)=[xdd ydd];
end

t=[t1';t2';t3'];
xyc=[xy1;xy2;xy3];
xycd=[xy1d;xy2d;xy3d];
xycdd=[xy1dd;xy2dd;xy3dd];

%========= End of xytraj2.m =============

function xdot=system(t,x)
%----------------------------------------------------------
%  NAME:
%      system.m
%
%  DESCRIPTION:
%    This program contains the function to be
%    integration using the Range Kutta routine
%    (rk_45.m).
%
%  INPUTS:
%      t: time.
%      x: Vector containing the joint angles and their
%         first time derivatives, the deflection and
%         slope at the end of the first link and their
%         first time derivatives and the deflection and
%         slope at the end of the second link and their
%         first time derivatives.
%
%  OUTPUTS:
%      System of first order differential equations.
%
%  GLOBAL VARIABLES:
%      m1: Mass of the first link.
%      m2: Mass of the second link.
%      l1: Length of the first.
%      l2: Length of the second.
```

```
%       Ma: Mass of the joint actuator at the end of
%           the first link.
%       Ja: Inertia of the joint actuator at the end of
%           the first link.
%       ML: Mass of the payload at the end of the
%           second link.
%       JL: Inertia of the payload at the end of the
%           second link.
%       KP: Proportional Gains.
%       KD: Derivative Gains.
%
%  LANGUAGE
%  386-MATLAB                    Version 3.5j
%
%  Written by: Amor Jnifene, University of Ottawa,
%  December, 1993.
%-----------------------------------------------------

t1d=x(1);t2d=x(2);u1d=x(3);u2d=x(4);u3d=x(5);
u4d=x(6);
t1=x(7); t2=x(8); u1=x(9);
u2=x(10);u3=x(11);u4=x(12);
qd=[t1d t2d u1d u2d u3d u4d]';
%-----------------------------------------------------
s22=sin(t2+u2);c22=cos(t2+u2);
%-----------------------------------------------------
a11 = 1/2*l1; a21 = -1/12*l1^2;
a111 = 13/35*l1; a121 = -11/210*l1^2;
a221 = 1/105*l1^3;
b11 = 7/20*l1^2; b21 = -1/20*l1^3;
k111 = 12/l1^3; k121 = -6/l1^2; k221 = 4/l1;

a12=  1/2*l2;   a22=  -1/12*l2^2;
a112=  13/35*l2; a122=  -11/210*l2^2; a222=
1/105*l2^3;
b12=  7/20*l2^2; b22=  -1/20*l2^3;
k112=  12/l2^3;  k122=  -6/l2^2; k222= 4/l2;

%-----------------------------------------------------
n11 = m1*(1/3*l1^3+u1^2*a111+2*u1*u2*a121...
+u2^2*a221);
n12 = m1*b11;
n13 = m1*b21;
n22 = m1*a111;
n23 = m1*a121;
n33 = m1*a221;

MLINK1=[n11 0 n12 n13 0  0
         0  0 0    0  0  0
        n12 0 n22 n23 0  0
        n13 0 n23 n33 0  0
         0  0 0    0  0  0
         0  0 0    0  0  0];

J11 = Ma*(l1^2+u1^2)+Ja;
J12 = Ma*l1;
J13 = Ja;
J22 = Ma;
J23 = 0;
J33 = Ja;

MLOAD1=[J11 0 J12 J13  0  0
         0  0  0   0   0  0
        J12 0 J22 J23  0  0
        J13 0 J23 J33  0  0
         0  0  0   0   0  0
         0  0  0   0   0  0];

m11 = m2*(l1^2*l2+u1^2*l2+1/3*l2^3+u3^2*a112...
+2*u3*u4*a122+u4^2*a222+(l1*l2^2+2*u1...
*(a12*u3+a22*u4))*c22-2*l1*(a12*u3+a22*u4)*s22);

m12 = m2*(1/3*l2^3+u3^2*a112+2*u3*u4*a122...
+u4^2*a222+(1/2*l1*l2^2...
+u1*(a12*u3+a22*u4))*c22+(1/2*u1*l2^2-...
l1*(a12*u3+a22*u4))*s22);

m13 = m2*(l1*l2+1/2*l2^2*c22-(a12*u3+...
a22*u4)*s22);

m14 = m2*(1/3*l2^3+u3^2*a112+2*u3*u4*a122+...
u4^2*a222+(1/2*l1*l2^2...
+u1*(a12*u3+a22*u4))*c22+(1/2*u1*l2^2-...
l1*(a12*u3+a22*u4))*s22);

m15 = m2*((l1*c22+u1*s22)*a12+b12);

m16 = m2*((l1*c22+u1*s22)*a22+b22);

m22 = m2*(1/3*l2^3+u3^2*a112+2*u3*u4*a122+...
u4^2*a222);

m23 = m2*(1/2*l2^2*c22-(a12*u3+a22*u4)*s22);

m24 = m2*(1/3*l2^3+u3^2*a112+2*u3*u4*a122+...
u4^2*a222);

m25 = m2*b12;

m26 = m2*b22;

m33 = m2*l2;

m34 = m2*(1/2*l2^2*c22-(a12*u3+a22*u4)*s22);

m35 = m2*a12*c22;
```

```
m36 = m2*a22*c22;

m44 = m2*(1/3*l2^3+u3^2*a112+2*u3*u4*a122+...
u4^2*a222);

m45 = m2*b12;

m46 = m2*b22;

m55 = m2*a112;

m56 = m2*a122;

m66 = m2*a222;

MLINK2=[m11 m12 m13 m14 m15 m16
        m12 m22 m23 m24 m25 m26
        m13 m23 m33 m34 m35 m36
        m14 m24 m34 m44 m45 m46
        m15 m25 m35 m45 m55 m56
        m16 m26 m36 m46 m56 m66];

l11 = ML*(l1^2+u1^2+l2^2+u3^2+2*(l1*l2+...
u1*u3)*c22-2*l1*u3*sin(t2+u2))+JL;

l12 = ML*(l2^2+u3^2+(l1*l2+u1*u3)*c22+(u1*l2-...
l1*u3)*s22)+JL;
l13 = ML*(l1+l2*c22-u3*s22);
l14 = ML*(l2^2+u3^2+(l1*l2+u1*u3)*c22+...
(u1*l2-l1*u3)*s22)+JL;
l15 = ML*(l1*c22+u1*s22+l2);
l16 = JL;
l22 = ML*(l2^2+u3^2)+JL;
l23 = ML*(l2*c22-u3*s22);
l24 = ML*(l2^2+u3^2)+JL;
l25 = ML*l2;
l26 = JL;
l33 = ML;
l34 = ML*(l2*c22-u3*s22);
l35 = ML*u3*c22;
l36 = 0;
l44 = ML*(l2^2+u3^2)+JL;
l45 = ML*l2;
l46 = JL;
l55 = ML;
l56 = 0;
l66 = JL;

MLOAD2=[l11 l12 l13 l14 l15 l16
        l12 l22 l23 l24 l25 l26
        l13 l23 l33 l34 l35 l36
        l14 l24 l34 l44 l45 l46
        l15 l25 l35 l45 l55 l56
        l16 l26 l36 l46 l56 l66];

h11 = -2*t1d*t2d*ML*s22*u1*u3-2*t1d*t2d*ML...
*l1*u3*c22-2*t1d*u2d*m2*l1*c22*a22*u4-...
2*t1d*t2d*m2*u1*s22*a12*u3-2*t1d*t2d...
*ML*s22*l1*l2+2*t1d*u1d*m2*c22*a12*u3...
+2*t1d*u3d*ML*c22*u1-2*t1d*u2d*m2...
*u1*s22*a22*u4-2*t1d*u2d*m2*l1*c22*a12...
*u3-2*t1d*u2d*m2*u1*s22*a12*u3...
-2*t1d*u2d*ML*s22*l1*l2+2*t1d*u4d*m2...
*u1*c22*a22+2*t1d*u1d*ML*u1...
+2*t1d*u1d*Ma*u1+2*t1d*u1d*m2*u1*l2+...
2*t1d*u3d*m2*u3*a112...
+2*t1d*u3d*ML*u3+2*t1d*u1d*m1*u1*a111...
+2*t1d*u1d*m1*u2*a121...
+2*t1d*u4d*m2*u3*a122+2*t1d*u3d*m2*u4*a122...
+3*t1d*u2d*m1*u2^2*a221+2*t1d*u2d*m1...
*u1*a121+2*t1d*u4d*m2*u4*a222...
+2*t2d*u4d*m2*u4*a222+2*t2d*u4d*m2*u3*a122...
+2*t2d*u3d*m2*u4*a122+2*t2d*u3d*ML*u3...
+2*t2d*u3d*m2*u3*a112+2*u2d*u3d*ML*u3...
+2*u2d*u3d*m2*u4*a122+2*u2d*u3d*m2*u3*a112...
+2*u2d*u4d*m2*u4*a222+2*u2d*u4d*m2*u3*a122...
-2*t1d*u4d*m2*l1*s22*a22-2*t1d*u2d*ML*s22...
*u1*u3-t2d^2*ML*s22*l1*l2+2*t2d*u4d*m2...
*u1*c22*a22-2*t2d*u4d*m2*l1*s22*a22;

h12=2*t2d*u3d*m2*u1*c22*a12-2*t2d*u3d*m2...
*l1*s22*a12+2*t2d*u3d*ML*c22*u1-2*t2d*...
u2d*m2*l1...
*c22*a22*u4+2*u2d*u3d*ML*c22*u1-...
2*u2d*u3d*ML*l1*s22-2*u2d*u3d*m2*l1...
*s22*a12-u2d^2*m2*u1*s22*a22*u4...
-u2d^2*m2*l1*c22*a12*u3-u2d^2*ML*s22*u1*u3...
-u2d^2*m2*u1*s22*a12*u3+1/2*u2d^2*m2*c22...
*u1*l2^2-u2d^2*ML*l1*u3*c22-1/2*u2d^2*m2*s22...
*l1*l2^2+u2d^2*ML*c22*u1*l2-2*u2d*u4d...
*m2*l1*s22*a22+2*u2d*u4d*m2*u1*c22*a22...
+2*u2d*u3d*m2*u1*c22*a12...
-t2d^2*m2*l1*c22*a22*u4-t2d^2*ML*s22*u1*u3...
-t2d^2*ML*l1*u3*c22-1/2*t2d^2*m2*s22*l1*l2^2...
-2*t2d*u2d*ML*l1*u3*c22+1/2*t2d^2*m2*c22...
*u1*l2^2+t2d^2*ML*c22*u1*l2+t2d*u2d*m2*c22...
*u1*l2^2-t2d*u2d*m2*s22*l1*l2^2+2*t2d*u2d*ML...
*c22*u1*l2-u2d^2*m2*l1*c22*a22*u4-u2d^2*ML...
*s22*l1*l2-2*t2d*u2d*m2*u1*s22*a12*u3...
-2*t2d*u2d*m2*u1*s22*a22*u4-2*t2d*u3d...
*ML*l1*s22-2*t2d*u2d*m2*l1*c22*a12*u3-...
2*t2d*u2d...*ML*s22*u1*u3-t2d^2*m2*u1...
*s22*a22*u4-t2d^2*m2*l1*c22*a12*u3...
-t2d^2*m2*u1*s22*a12*u3-2*t2d*u2d*ML*s22*l1*l2;
h13=2*t1d*u3d*m2*u1*c22*a12-2*t1d*u2d*...
ML*l1*u3*c22-2*t1d*u3d*m2*l1*s22*a12-2*t1d*...
```

```
u3d*ML*l1*s22+2*t1d*u1d*m2*c22*a22*u4-...
2*t1d*t2d*m2*u1*s22*a22*u4-2*t1d*t2d*m2*...
l1*c22*a12*u3-2*t1d*t2d*m2*l1*c22*a22*u4...
+2*t1d*u1d*ML*u3*c22;
h1=h11+h12+h13;

h2 =2*t1d*u1d*m2*c22*a12*u3+2*t1d*...
u3d*m2*u3*a112+2*t1d*u3d*ML*u3...
+2*t1d*u4d*m2*u3*a122+2*t1d*u3d*m2*...
u4*a122+2*t1d*u4d*m2*u4*a222...
+2*t2d*u4d*m2*u4*a222+2*t2d*u4d*m2*...
u3*a122+2*t2d*u3d*m2*u4*a122...
+2*t2d*u3d*ML*u3+2*t2d*u3d*m2*u3*a112...
+2*u2d*u3d*ML*u3+2*u2d*u3d*m2*...
u4*a122+2*u2d*u3d*m2*u3*a112+2*u2d*...
u4d*m2*u4*a222+2*u2d*u4d*m2*u3*a122+2*t1d*...
u1d*m2*c22*a22*u4+2*t1d*u1d*ML*u3*c22...
+t1d*u1d*m2*l2^2*s22+2*t1d*u1d*ML*l2*...
s22-u1d*u3d*ML*s22+t1d^2*m2*u1*s22*a22*u4...
+t1d^2*m2*u1*s22*a12*u3+t1d^2*m2*l1...
*c22*a12*u3+t1d^2*m2*l1...*c22*a22*u4...
+t1d^2*ML*s22*l1*l2+t1d^2*ML*s22*u1*u3+...
t1d^2*ML*l1*u3*c22+u1d*u3d*ML*u3*s22;

h3 = -t1d*u2d*m2*l2^2*s22-2*t1d*t2d*m2*...
c22*a22*u4-2*t1d*u2d*m2*c22*a12*u3-...
2*t1d*u2d*m2*c22*a22*u4-1/2*t2d^2*m2...
*l2^2*s22-t1d^2*ML*u1-t1d^2*Ma*u1...
-t1d^2*m2*u1*l2-t1d^2*m1*u1*a111-...
t1d^2*m1*u2*a121-2*u2d*m2*s22*a22*...
u4d-1/2*u2d^2*m2*l2^2*s22-u2d^2*m2*c22...
*a12*u3-u2d^2*m2*c22*a22*u4...
-t2d^2*ML*l2*s22-u3d*t2d*ML*u3*s22-u3d...
*u2d*ML*u3*s22-2*u2d*u3d*m2*s22*a12-...
t2d^2*m2*c22*a12*u3-u2d*u3d*ML*s22-...
u2d^2*ML*l2*s22-2*t2d*u2d*m2*c22*a22...
*u4-2*t1d*t2d*m2*c22*a12*u3-t1d^2*m2*...
c22*a12*u3-t1d^2*m2*c22*a22*u4-t1d^2*...
ML*u3*c22-2*t1d*u3d*ML*s22...
-2*t1d*u3d*m2*s22*a12-2*t1d*t2d*ML...
*l2*s22-2*t1d*t2d*ML*u3*c22...
-2*t1d*u2d*ML*l2*s22-2*t1d*u2d*ML...
*u3*c22-t2d^2*m2*c22*a22*u4...
-t2d*u2d*m2*l2^2*s22-2*t2d*u3d*m2*s22*...
a12+ML*c22*u3d^2-2*t2d*u2d*ML...
*l2*s22-2*t2d*u2d*ML*u3*c22-t2d^2*ML*u3*c22...
-t1d*t2d*m2*l2^2*s22-u2d^2*ML*u3*c22...
-t2d*u3d*ML*s22-2*t2d*m2*s22*a22*u4d...
-2*t2d*u2d*m2*c22*a12*u3...
-2*m2*s22*a22*t1d*u4d;

h4 = 2*t1d*u1d*m2*c22*a12*u3...
-t1d^2*m1*u1*a121...
+2*t1d*u3d*m2*u3*a112+2*t1d*u3d*ML*u3...
-3/2*t1d^2*m1*u2^2*a221+2*t1d*u4d*m2*u3*a122...
+2*t1d*u3d*m2*u4*a122+2*t1d*u4d*m2*u4*a222...
+2*t2d*u4d*m2*u4*a222+2*t2d*u4d*m2*u3*a122...
+2*t2d*u3d*m2*u4*a122+2*t2d*u3d*ML*u3...
+2*t2d*u3d*m2*u3*a112+2*u2d*u3d*ML*u3...
+2*u2d*u3d*m2*u4*a122+2*u2d*u3d*m2*u3*a112...
+2*u2d*u4d*m2*u4*a222+2*u2d*u4d*m2*u3*a122...
+2*t1d*u1d*m2*c22*a22*u4+2*t1d*u1d*ML...
*u3*c22+t1d*u1d*m2*l2^2*s22...
+2*t1d*u1d*ML*l2*s22-u1d*u3d*ML*s22...
+t1d^2*m2*u1*s22*a22*u4...
+t1d^2*m2*u1*s22*a12*u3+t1d^2*m2*l1*c22...
*a12*u3+t1d^2*m2*l1*c22*a22*u4+t1d^2*ML*...
s22*l1*l2+t1d^2*ML*s22*u1*u3...
+t1d^2*ML*l1*u3*c22+u1d*u3d*ML*u3*s22;

h5 = -t1d^2*ML*u3+t2d*u1d*ML*s22...
+u1d*u2d*ML*s22-t1d^2*m2*u1*c22*a12...
+t1d^2*m2*l1*s22*a12-t1d^2*ML*c22*u1...
+t1d^2*ML*l1*s22-u1d*t2d*ML*u3*s22...
-u1d*u2d*ML*u3*s22+2*t1d*u1d*m2*s22*a12...
+2*t1d*u1d*ML*s22-2*t2d*u2d*m2*u3*a112...
-2*t2d*u2d*m2*u4*a122-2*t2d*u2d*ML*u3...
-u2d^2*m2*u3*a112-u2d^2*m2*u4*a122-...
t2d^2*m2*u3*a112-t2d^2*m2*u4*a122...
-t2d^2*ML*u3-u2d^2*ML*u3-t1d^2*m2*u3*a112...
-t1d^2*m2*u4*a122-2*t1d*t2d*m2*u3*a112...
-2*t1d*t2d*m2*u4*a122-2*t1d*t2d*ML*u3...
-2*t1d*u2d*m2*u3*a112-2*t1d*u2d*m2*u4*a122-...
2*t1d*u2d*ML*u3;

h6 = -2*t1d*t2d*m2*u3*a122-2*t1d*u2d*m2...
*u3*a122-2*t1d*u2d*m2*u4*a222...
-2*t1d*t2d*m2*u4*a222...
-t2d^2*m2*u3*a122-t2d^2*m2*u4*a222...
-2*t2d*u2d*m2*u3*a122...
-2*t2d*u2d*m2*u4*a222-u2d^2*m2*u3*a122-...
u2d^2*m2*u4*a222-t1d^2*m2*u4*a222...
-t1d^2*m2*u3*a122-m2*a22*t1d^2*u1*c22...
+m2*a22*t1d^2*l1*s22+2*t1d*m2*s22*a22*u1d;


k1 = 0;
k2 = 0;
k3 = EI1*(k111*u1+k121*u2);
k4 = EI1*(k121*u1+k221*u2);
k5 = EI2*(k112*u3+k122*u4);
k6 = EI2*(k122*u3+k222*u4);

KVEC=[k1 k2  k3 k4 k5 k6 ]';
%-------------------------------------------
% Partition of the mass matrix -------
```

```
A=MLINK1+MLOAD1+MLINK2+MLOAD2;

MR=[A(1,1) A(1,2)
   A(1,2) A(2,2)];

MRF=[A(1,3) A(1,4)   A(1,5) A(1,6)
     A(2,3) A(2,4)   A(2,5) A(2,6)];

MF=[A(3,3) A(3,4)  A(3,5)  A(3,6)
   A(3,4) A(4,4)  A(4,5)  A(4,6)
   A(3,5) A(4,5)  A(5,5)  A(5,6)
   A(3,6) A(4,6)  A(5,6)  A(6,6)];

%-------------------------------------------------
HVEC=[h1 h2 h3 h4 h5 h6]';
%-------------------------------------------------
%-Partition of HVEC----------------
HR=[HVEC(1) HVEC(2)]';
HF=[HVEC(3) HVEC(4) HVEC(5) HVEC(6)]';
%-------------------------------------------------
% Partition of KVEC
GR=[k1 k2]';
GF=[k3 k4 k5 k6]';
% System for the control of the angles of points along
%  the links

PHI=[ [FI11 FI12]/DIST1    0        0
         0        0        [FI21 FI22]/DIST2 ];

DEL=MF-MRF'*PHI;
IDEL=inv(DEL);

GAMMARX=MR-(MRF-MR*PHI)*IDEL*MRF';
LAMDARX=HR+GR-(MRF-MR*PHI)*IDEL...
*(HF+GF);
%-------------------------------------------------
UX1(II)= (FI11*u1+FI12*u2)/DIST1;
UX2(II)= (FI21*u3+FI22*u4)/DIST2;
UX1D(II)=(FI11*u1d+FI12*u2d)/DIST1;
UX2D(II)=(FI21*u3d+FI22*u4d)/DIST2;
%-------------------------------------------------
CON=1;
D=1+2*CON+CON^2;
A1=1/D;
A2=2*CON/D;
A3=CON^2/D;

if (II<=3*IID1),
      BETA1=t1+UX1(II);
      BETA1D=t1d+UX1D(II);
else

BETA1=t1+A1*UX1(II-IID1)+A2*UX1(II-2*IID1)+A
3*UX1(II-3*IID1);

BETA1D=t1d+A1*UX1D(II-IID1)+A2*UX1D(II-2*II
D1)+A3*UX1D(II-3*IID1);

end

if (II<=3*IID2),
      BETA2=t2+UX2(II);
      BETA2D=t2d+UX2D(II);
else

BETA2=t2+A1*UX2(II-IID2)+A2*UX2(II-2*IID2)...
+A3*UX2(II-3*IID2);
BETA2D=t2d+A1*UX2D(II-IID2)+A2...
*UX2D(II-2*IID2)+A3*UX2D(II-3*IID2);
end

TAUR=thetadd(II,:)'...
   +KP*(theta(II,:)-[BETA1 BETA2])'...
   +KD*(thetad(II,:)-[BETA1D BETA2D])';
%-------------------------------------------------
TAU(1:2)=GAMMARX*TAUR+LAMDARX;
%-------------------------------------------------
B=-KVEC-HVEC-DAMP*qd+TAU;
%-------------------------------------------------
qdd=A\B;
%-------------------------------------------------
xdot=[qdd',[qd]']';
```