Founded 1905

CONTROL OF CONSTRAINED ROBOT SYSTEMS

BY

HUANG LOULIN
(BEng, MEng)

A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE
2004

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Associate Professor S. S. Ge, for his guidance and support throughout my research. He not only set the right direction for my research, but also took care of each step I took.

I would also like to thank my co-supervisor Professor T. H. Lee for his kind help, encouragement and suggestions throughout my research.

Thanks Miss F. Hong, Dr. C. J. Zhou, Dr. C. Wang and Dr. Z. P. Wang and all of my friends at National University of Singapore and Singapore Polytechnic for their assistances.

Last but not least, my deepest gratitude goes to my dearest wife Chen Xin, my parents and parents-in-law for their love, understanding and sacrifice. Their continuous and unconditional support is an indispensable source of my strength and confidence to face up any challenge.

Contents

Summary

This thesis focuses on some issues of control of constrained robots. The control objectives are to make the position of the robot and the constraint force achieve their desired values in various situations which were not studied sufficiently in the past. These situations include that the constraint is in motion, that the dynamics of the constraint is unknown as well as that of the robot, and that the robot's joints are flexible while the joint stiffness is unknown. The issue of position/force tracking of constrained robot with impedance control is also addressed. The controller design for keeping the contact between the end effector of the robot and the constraint is also studied.

In the study of constrained robot control, the motion of the constraint object is usually neglected. However, in many industrial applications, such as assembling or machining mechanical parts, the constraint (mechanical part) is required to move with respect to not only the world coordinates but also the end effectors of the robotic arms. In this thesis, the dynamic model of constrained robot system when the constraint is in motion is set up. A model-based adaptive controller and a model-free neural network controller are developed. Both controllers guarantee the asymptotic tracking of the position of the constraint object to its desired trajectory and the boundedness of constraint force tracking error. Asymptotic convergence of the constraint force to its desired value can also be achieved under certain conditions.

Impedance control is aimed to make the dynamic impedance between the robot and the environment follow a desired one. In this thesis, adaptive, robust or neural network based control approaches are used to provide the traditional impedance control scheme with position/force tracking capabilities. The varying desired impedance is adaptively tuned with the robot position tracking errors. The controllers guarantee the convergence of position tracking errors and the boundedness of force tracking errors. The convergence of force error to zero can also be achieved under some conditions.

The thesis also addresses the explicit force control of a constrained robot considering the dynamics of the constraint. The constraint is modeled as a chain of multiple mass-spring-damper (CMMSD) units which describes the constraint's dynamic behaviors during contact and noncontact motions. Considering the difficulties in obtaining the dynamic model and the internal states of the constraint, a model reference adaptive controller (MRAC) and an adaptive backstepping controller are designed to control the constraint force. The proposed controllers are independent of system parameters and guarantee the asymptotic convergence of the force to its desired value and the boundedness of all the closed-loop signals.

Though maintaining the contact between the robot end effector and the constraint is essential to many controllers developed for constrained robots, how to achieve it is not addressed explicitly in the literature. In this thesis, the unidirectionality of the contact force for maintaining the contact is explicitly included in modeling and control of a constrained robot system. A fuzzy tuning mechanism is developed to adjust the impedance between the robot and the constraint according to the contact situations. A unidirectional force controller is developed based on a set of fuzzy rules and the nonlinear feedback technique.

The thesis also addresses the issue of adaptive position/force control of uncertain constrained flexible joint robots. The controller is designed without the assumption of sufficient large joint stiffness used in many singular perturbation based controllers. The controller design relies on the feedback of joint state variables, and avoids noisy joint torque feedback. The traditional singular perturbation approach for free flexible joint robots is also extended to control constrained flexible joint robot with sufficiently large joint stiffness. By properly defining the fast and

the slow variables with the robot position and the constraint force tracking errors, a boundary layer system and a quasi-steady-state system are established and are made exponentially stable with the controller developed. Both controllers achieve the robot position tracking and the boundedness of constraint force tracking errors.

List of Figures

Chapter 1

Introduction

This thesis focuses on some issues of control of constrained robots. The control objectives are to make the position of the robot and the constraint force achieve their desired values. Various controllers are developed considering the following situations which were not sufficiently covered in the past:

1. the constraint is in motion;
2. the constraint dynamics is taken into account in the controller design;
3. both the dynamic model of the robot and that of the constraint are unknown;
4. the joints of the constrained robot are flexible and the joint stiffness is unknown.

The issue of position/force tracking of constrained robot with impedance control is addressed. The controller design for keeping the contact between the end effector of the robot and the constraint is also studied.

In this chapter, the background and the previous work of constrained robot control are examined. In the later part of the chapter, the motivation and the organization of the thesis are presented.

1.1 Background and Previous Work

The control of constrained robotic manipulators has been studied extensively in the last two decades. There are mainly three different control approaches, namely *hybrid position/force control* [1][2][3], *impedance control* [4][5][6] and *constrained robot control* [7][8][9]. These different control approaches are also combined in some applications [10]. In *hybrid position/force control* scheme, the robot's workspace is divided into two subspaces orthogonal to each other, among which, one is for position control and the other is for force control. With a so-called "selection matrix", the control action is switched between these two subspaces. The selection matrix requires accurate modeling of the robot, the environment and the contact between the robot and the environment. The robustness of the controller is compromised by discontinuity resulted from switching of the control actions. In *constrained robot control* scheme, the constraint is assumed to be ideally rigid and the end effector of the robot is kept on the constraint surface. Through a nonlinear transformation, the dynamics of the constrained robot system is described by a set of differential and algebraic equations. The differential equations describe an unconstrained robot motion along the constraint manifold, and the algebraic equation describes the relationship between the constraint force and the system dynamics. Both the force and the position are explicitly controlled with nonlinear feedback control scheme. In *impedance control* scheme, the interaction between the robot and the environment is modeled as a general impedance. Instead of accurate tracking of robot position or constraint force, the objective of the controller is to achieve a desired generalized dynamic impedance between the robot and the constraint. Most impedance control schemes are based on model-based nonlinear feedback control which requires exact dynamic models of the robot and the constraint. Most controllers developed are for the robots with serial links. Recently these controllers are also extended to the parallel robots where closed kinematic chains exist [11].

Nonlinear feedback control, or *computed torque control* is the foundation of most control approaches for constrained robots. It contains a feed forward loop for compensating the nonlinear robot dynamics, and a servo compensator to make the controlled variables (position of the robot, constraint force or impedance) converge to their desired values. Traditional nonlinear feedback control needs accurate modeling of the robot and the constraint environment. To deal with system uncertainties, *adaptive control* [12][13][14][15][16][17][18], *robust control* [19][20][21][22][23], *neural network control* [24][26] and their combinations are used in the controller design. The property that the dynamics of the robot is linear with respect to a set of robot parameters, or in another word, the robot dynamics can be expressed in a linear-in-parameters (LIP) form, is essential for designing the parameter adaptation laws in the adaptive control scheme. *Robust control* approach, mostly sliding mode control, is designed for compensating the dynamic modeling errors and external noises. The switching surface is a function of the tracking errors of the controlled variables (position, force or impedance). By making the closed loop system evolve along the switching surface, the tracking of the controlled variables to their desired values is also achieved. *Neural network control* is a model free control approach in which the dynamic model of the robot is approximated by a multi-layer neural network. The weights of the neural network are tuned with the tracking errors of the controlled variables.

Most controllers for the constrained robotic manipulators are designed with one or more than one of the followings assumptions:

1. the constraint surface is rigid;
2. the constraint is stationary and the dynamic models of the constraint and the contact are ignored;
3. the end-effector of the robot is always on the surface of the constraint surface;
4. the links/joints of the robot are rigid, or their stiffness are known.

These assumptions are restrictive in some applications where the constraint surface may be flexible, the constraint is in motion or the contact between the robot and the constraint is not always maintained. The joints or the links of the robot can be flexible and their stiffness can take any values.

In the past years, some control approaches have been developed with less restrictive assumptions. One area attracting much attention is the controller design when the constraint is not necessarily rigid. In this case, the constraint's dynamic behavior under the contact should be taken into consideration in the controller design. In [27], the constraint surface is modeled as a first order damper and spring system. With the assumption that the stiffness and damping ratio of the constraint are sufficiently large, a singular perturbation approach is applied to regulate the displacement of the constraint surface and the constraint force. In [28], the constraint is modeled as a general spring of an unknown stiffness. The constraint force is accommodated by adjusting the desired constraint displacement. In [29], the constraint is modeled as a second order mass-spring-damper system. The tracking of the constraint force is achieved by scaling down the desired displacement adaptively. A common feature of these approaches is that the constraint force is indirectly controlled through the modification of the displacement of the constraint surface.

A more comprehensive dynamic model of a constraint is proposed in [30]. In this model, the motion of the constraint is divided into three stages: *constrained motion* (rigid contacts), *compliant motion* (compliant contact) and *collision* (transition between the constrained motion and the free motion of the robot). A singular perturbation approach is used to analyze and simulate the force response with different constraint parameters. With the same constraint model, the theory of generalized dynamic system (GDS) is applied to develop discontinuous force/position controllers [8][32][33]. Different control actions are activated in different constraint motion stages determined by the internal states and the parameters of the constraint which though are difficult to measure in practical applications. A more complicated case where multiple rigid bodies make contacts each other is discussed in [34].

Another key assumption of the controller design for constrained robots is that the constraint is stationary with respect to the world coordinate. In some applications, the motion of the constraint with respect to the world coordinate and its relative motion with respect to the end effector of the manipulator are both required. A typical example is that one robotic arm performs assembling or machining task on a work piece held tightly by another robotic arm. In some machining processes such as deburring, grinding and polishing, the motion of the part with respect to the robotic manipulators is needed to expand the operational space of the robot and increase the efficiency of the work [38]. For this reason, it is important to investigate the control of constrained robot when the constraint is in motion.

In many researches in constrained robot control, it has been assumed that the end effector of the robotic manipulator is kept on the constraint surface all the time. How to keep this contact has been neglected by most researchers so far. As pointed out in [39], the force control schemes developed with the assumption that the end effector of the robotic manipulator always keeps contact with the environment are not effective when the contact is lost. Some researchers have tried to model the transition from non-contact into contact and vice-versa [32][40] and some . Most models established for analyzing the behaviors of the contact ate too complicated to be used in the dynamic control synthesis. Model-free approaches such as fuzzy control or neural network control [60][61] should be effective alternatives to solve this problem.

Regarding the requirement of keeping the contact between the end effector of the robotic manipulator and the constraint, impedance control is an exception as it takes care of both unconstrained and constrained motion of the robot. Under impedance control, the robot position tracking can only be achieved during its motion in free space. The position and force are indirectly controlled during the robot's constrained motion. This feature makes it very appealing in applications where the stable impedance relation between the constraint and the robot is important. Most impedance controllers are designed with the model-based computed torque method which requires exact dynamic models of the robot and the constraint. To handle uncertainties, adaptive control, robust control or neural network control approaches are introduced into the impedance control scheme. In [15], the concept of *target-impedance reference trajectories* (TIRT) is proposed. A TIRT is solved from the desired impedance model under the desired constraint force. The dynamic parameters of the system are updated adaptively with the error between the actual robot trajectory and the TIRT. The desired impedance is achieved indirectly by making this error asymptotically stable. In [21], a sliding model control approach is developed. The switching function is defined with the so called *impedance error* — an error between the actual impedance and the desired impedance. The traditional sliding model control approach is used to make the impedance error asymptotically stable. In [22], the results in [21] is extended to a more general second order impedance model. The constraint force tracking errors are also considered in the definition of the switching function. In [24], a neural network based adaptive impedance control approach is proposed. The weights of the neural network are tuned with the error between the robot trajectory and the TIRT. In [25], an impedance control scheme with a programmable impedance is developed for an one-degree-of-freedom elastic joint robot. The uncertainties of the constraint is not considered in the above adaptive/robust impedance control schemes.

There are a few impedance control approaches dealing with the robot's position tracking and force tracking with impedance control [5][39][41]. In [5], direct control of position or force is achieved with a PI adaptive control law in which the robot's desired trajectory varies with the force tracking errors and environment parameter estimation errors. In [39], a so-called *parallel control* scheme is proposed. In this control scheme, the impedance control action is projected along two directions, one along the normal and the other along the tangent at the contact point on the constraint surface. The control actions along these two directions are force control and position control respectively. This control scheme relies on the accurate modeling of the controlled system and the assumption of zero stiffness along the tangent of the constraint surface. In [41], a model reference adaptive control law is proposed in which the position (force) tracking is achieved by updating the desired impedance with position (force) tracking errors. These adaptive control schemes require the exact dynamic modeling of the robot and the constraint.

Flexibilities of the joints of constrained robot pose another challenge for the controller design. The control of flexible joint robotic manipulators has been studied extensively in the last decades, though mostly in the area of free flexible joint robots. With singular perturbation (SP) analysis, controllers developed for rigid robotic manipulators can be extended for the robots of *weak* joint flexibilities (the joint stiffness is sufficiently large) [64][81][82][83][84][96]. Feedback linearization method is also used in the controller design, but it requires exact dynamic modeling of the robot and the measurements of its joint accelerations and jerks [86][87]. To deal with uncertainties of robotic systems, many adaptive control schemes [88][89][90] are developed from the pioneering work on adaptive control of rigid robotic manipulators in [13]. The controller design requires joint acceleration feedback, filtering of system dynamics and the calculation of the inverse inertia matrix of the robot. The controller is complex in structure and is computationally intensive. Treating a flexible joint robotic manipulator as a cascading system, joint torque feedback and backstepping approaches are also applied in the controller design [91][92]. The measurement of joint torques and their noisy derivatives are required in the design of the above controllers.

Compared with those for free flexible joint robots, much fewer research results are reported on controlling constrained flexible joint robotic manipulators and most of which are on the robot systems with *known* parameters and *weak* joint flexibility [93][94]. In [95], a Cartesian-space robot model is used to develop the position control and the force control along certain curvilinear directions as proposed in [16]. The joint torque and its up to 2nd order derivatives are needed in the controller design. In [96], a Cartesian impedance control of flexible joint robots is developed based on joint torque feedback. With computed torque control, the joint dynamics and the link dynamics are decoupled and the desired impedance is achieved. The controller requires an exact knowledge of the system dynamics and the noisy joint torque feedback.

1.2 Motivations and Contributions of the Thesis

As discussed above, many idealistic assumptions are made for modeling and controlling constrained robotic manipulators. The typical assumptions include that the constraint is stationary, the constraint and the robot joints are rigid and the end effector of the manipulator is always kept on the constraint surface. The robot position or constraint force tracking with impedance control and how to make the constraint force unidirectional during the constrained motion are some issues which are worth further investigation.

The first issue studied in the thesis is the constrained robot control when the constraint is in motion. The system dynamic model is firstly established by assuming that the constraint is held and manipulated by one robotic manipulator and the end effector of the another robotic manipulator moves on the constraint surface. The properties of the dynamic model are then explored. Due to the complex system configuration and its uncertainties, both model-based adaptive controller and model free neural network controller are developed.

Another focus of the thesis is the position/force control of a constrained robot with impedance control. Though impedance control can handle constrained and unconstrained motions of the robot, how to achieve robot position or force tracking during the robot's constrained motion is still challenging problem. In this thesis, various control approaches such as robust, adaptive or neural network control are used to solve this problem.

As a departure from many controllers developed, the dynamic model of the constraint under the contact is treated as equally as that of the robot dynamics for the explicit force control of constrained robots. We model the contact between the end effector of the robot and the constraint as a chain of multiple mass - spring-damper units (CMMSD) which is more general than many other models proposed in the past. Applying the adaptive output feedback force controllers for a general CMMSD system – one based on model reference adaptive control (MRAC) and another based on backstepping control, the explicit force tracking is achieved without the knowledge of the dynamic models of the robot and the constraint and the internal states of the constraint.

A fuzzy control approach is applied to make the constraint force unidirectional in a constrained robot system. Though there are many models developed for the contacts between the rigid bodies [34][97][98], they are too complicated to be used for design a controller to achieve the unidirectionality of the constraint force. The controller design can be made simpler by observing how a person keeps his finger on an object with a force. What he does is to press his finger roughly along the normal vector “penetrating” the constraint surface at the contact point and to adjust the gesture of his hand to make the force felt at a reasonable level. From this observation and analyzing the relation between the constraint force and the parameters of the impedance between the robot and the constraint, some rules are derived and used in the development of a unidirectional force controller.

The adaptive position/force control for an uncertain constrained robot with flexible joints is very general as both the joint stiffness and the motor inertia are assumed to be unknown in addition to the robot inertia parameters. It mainly relies on the feedbacks of *joint state variables* (joint positions and velocities) and avoids noisy joint torque feedback. The singular perturbation approach in controlling free flexible joint robots is also extended to the positing/force control of constrained flexible joint robots. In this case, both the force and position signals are used to define slow and fast variables of the controlled system.

In summary, the following are the main contributions of the thesis:

1. Modeling and control of the robotic manipulator constrained by a moving object; model-based adaptive and model-free neural network control approached are developed respectively;

2. Development of robust, adaptive and neural network impedance control considering the uncertainties of the system; the controller achieves the robot’s position tracking and the boundedness of constraint force tracking error;

3. Development of an explicit force controller for constrained robotic manipulators by taking the dynamics of the constraint and the contact into consideration; the contact between the end effector and the constraint is modeled as a chained multiple mass-spring-damper system (CMMSD) and adaptive output feedback control methods are applied;

4. Development of a fuzzy controller to make the constraint force unidirectional

essential for keeping the contact between the robot's end effector and the constraint;

5. Development of adaptive position/force controllers for an uncertain constrained robot with flexible joints; the traditional singular perturbation approach is also extended to control the constrained flexible joint robots.

1.3 Outlines of the Thesis

The thesis contains seven chapters. The introduction of the thesis is given in Chapter 1. Chapter 2 covers the modeling and control of a robotic manipulator constrained by a moving object. Chapter 3 focuses on the position/force control of a constrained robot with robust adaptive and neural network based impedance control. Chapter 4 is on the explicit force control of a constrained robot by taking the dynamic model of the constraint into consideration. Chapter 5 is on the fuzzy unidirectional force control for a constrained robot. Chapter 6 is dedicated to the robust adaptive or singular perturbation based position/force control of constrained flexible joint robot. The conclusion and the future research are given in Chapter 7.

Chapter 2

Control of a Robot Constrained by a Moving Object

In this chapter, we investigate position and force control for a robotic manipulator constrained by an object which is held and manipulated by another robotic manipulator. It is required that the object follows a planned motion trajectory in the work space and the end-effector of the constrained robotic manipulator follows a planned trajectory on the object with a desired force.

The chapter is organized as follows. In Section 2.1, the kinematics and dynamic models of the robotic system are presented. In Section 2.2, a model-based adaptive controller is presented first, then it is extended to a model-free neural network based adaptive controller in Section 2.3. Both controllers are designed to control the positions of the constraint object and the robots' end-effectors, and the constraint forces asymptotically. In Section 2.4, simulation studies are used to show the effectiveness of the controllers. The conclusion is given in Section 2.5.

2.1 Kinematics and Force Model

The system under study is schematically shown in Figure 2.1. The object is held tightly and is moved as required in space by the end effector of manipulator 2. The

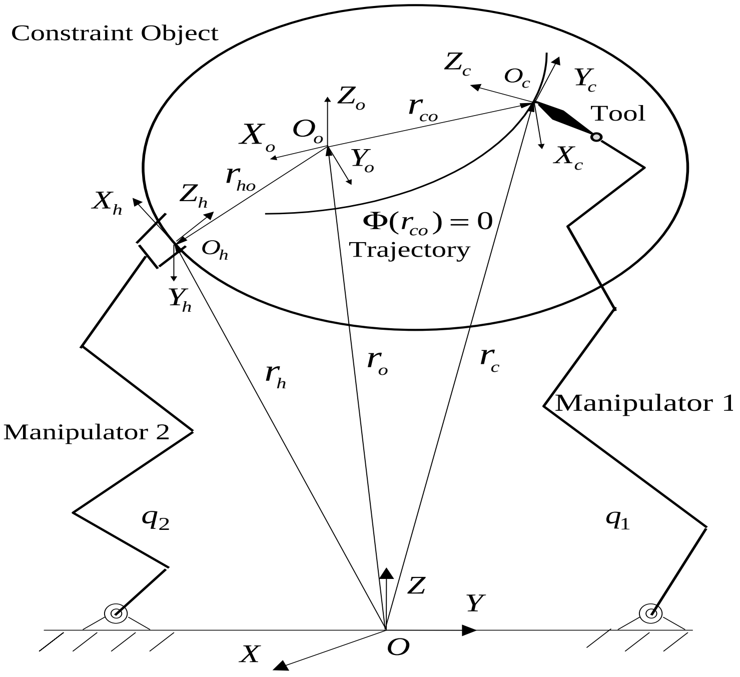

Figure 2.1: The Robot Constrained by a Moving Object

end effector of manipulator 1 follows a trajectory on the surface of the object, and exerts a certain force on it at the same time.

The following notations are used to describe the system in Figure 1:

O_c : the contact point between the end effector of manipulator 1 and the object;

O_o : the mass center of the object;

O_h : the point where the end effector of manipulator 2 holds the object;

$OXYZ$: the world coordinates;

$O_cX_cY_cZ_c$: the frame fixed with the tool of manipulator 1 with its origin at the contact point O_c;

$O_oX_oY_oZ_o$: the frame fixed with the object with its origin at the mass center O_o;

$O_hX_hY_hZ_h$: the frame fixed with the end-effector or hand of manipulator 2 with its origin at point O_h;

x_c : the position vector of O_c, the origin of frame $O_cX_cY_cZ_c$;

θ_c : the orientation vector of frame $O_cX_cY_cZ_c$

x_o : the position vector of O_o, the origin of frame $O_oX_oY_oZ_o$;

θ_o : the orientation vector of frame $O_oX_oY_oZ_o$;

x_h : the position vector of O_h, the origin of frame $O_hX_hY_hZ_h$;

θ_h : the orientation vector of frame $O_hX_hY_hZ_h$;

x_{ho} : the position vector of O_h, the origin of frame $O_hX_hY_hZ_h$ expressed in $O_oX_oY_oZ_o$;

θ_{ho} : the orientation vector of frame $O_hX_hY_hZ_h$ expressed in $O_oX_oY_oZ_o$;

x_{co} : the position vector of O_c, the origin of frame $O_cX_cY_hZ_c$ expressed in $O_oX_oY_oZ_o$;

θ_{co} : the orientation vector of frame $O_cX_cY_cZ_c$ expressed in $O_oX_oY_oZ_o$;

$r_c = [x_c^T\ \theta_c^T]^T$: the vector describing the posture of frame $O_cX_cY_cZ_c$;

$r_o = [x_o^T\ \theta_o^T]^T$: the vector describing the posture of frame $O_oX_oY_oZ_o$;

$r_h = [x_h^T\ \theta_h^T]^T$: the vector describing the posture of frame $O_hX_hY_hZ_h$;

$r_{ho} = [x_{ho}^T\ \theta_{ho}^T]^T \in R^6$: the vector describing the posture of frame $O_hX_hY_hZ_h$ expressed in $O_oX_oY_oZ_o$;

$r_{co} = [x_{co}^T\ \theta_{co}^T]^T \in R^6$: the vector describing the posture of frame $O_cX_cY_cZ_c$ expressed in $O_oX_oY_oZ_o$;

$q_1 \in R^{n_1}$: the joint variables of manipulator 1;

$q_2 \in R^{n_2}$: the joint variables of manipulator 2; and

$\Phi(r_{co}) = 0$: the trajectory expressed in the object frame $O_oX_oY_oZ_o$

The closed kinematic relationships of the system are given by the following equations

$$\begin{aligned} x_c &= x_o + R_o(\theta_o)x_{co} & (2.1)\\ x_h &= x_o + R_o(\theta_o)x_{ho} & (2.2)\\ R_c &= R_o(\theta_o)R_{co}(\theta_{co}) & (2.3)\\ R_h &= R_o(\theta_o) & (2.4) \end{aligned}$$

where $R_o(\theta_o) \in R^{3\times3}$ and $R_{co}(\theta_{co}) \in R^{3\times3}$ are the rotation matrices of θ_o and θ_{co}

respectively; $R_c \in R^{3\times 3}$ and $R_h \in R^{3\times 3}$ given above are the rotation matrices of frames $O_cX_cY_cZ_c$ and $O_hX_hY_hZ_h$ with respect to the world coordinate respectively.

Differentiating the above equations with respect to time t and considering the fact that the object is held by manipulator 2 tightly (accordingly, $\dot{x}_{ho} = 0$ and $\omega_{ho} = 0$), we have

$$\begin{aligned} \dot{x}_c &= \dot{x}_o + R_o(\theta_o)\dot{x}_{co} - S(R_o(\theta_o)x_{co})\omega_o & (2.5) \\ \dot{x}_h &= \dot{x}_o - S(R_o(\theta_o)x_{ho})\omega_o & (2.6) \\ \omega_c &= \omega_o + R_o(\theta_o)\omega_{co} & (2.7) \\ \omega_h &= \omega_o & (2.8) \end{aligned}$$

with

$$S(u) := \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix}$$

for a given vector $u = [u_1\ u_2\ u_3]^T$.

Define $v_c = [\dot{x}_c^T \quad \omega_c^T]^T$, $v_h = [\dot{x}_h^T \quad \omega_h^T]^T$, $v_o = [\dot{x}_o^T \quad \omega_o^T]^T$, $v_{co} = [\dot{x}_{co}^T \quad \omega_{co}^T]^T$ and $v_{ho} = [\dot{x}_{ho}^T \quad \omega_{ho}^T]^T$. From equations (2.5) to (2.8), the following velocity relations are established

$$\begin{aligned} v_c &= Av_o + R_A v_{co} & (2.9) \\ v_h &= Bv_o & (2.10) \end{aligned}$$

where

$$\begin{aligned} R_A &= \begin{bmatrix} R_o(\theta_o) & 0 \\ 0 & R_o(\theta_o) \end{bmatrix} \\ A &= \begin{bmatrix} I^{3\times 3} & -S(R_o(\theta_o)x_{co}) \\ 0 & I^{3\times 3} \end{bmatrix} \\ B &= \begin{bmatrix} I^{3\times 3} & -S(R_o(\theta_o)x_{ho}) \\ 0 & I^{3\times 3} \end{bmatrix} \end{aligned}$$

and $I^{3\times 3}$ is an identity matrix of dimension 3. In this thesis, $I^{n\times n}$ will be used to represent an identify matrix of dimension $n \times n$.

Assume that the end-effector of manipulator 1 follows the trajectory $\Phi(r_{co}) = 0$ in the object coordinates. The contact force f_c and the resulting force f_o are given by

$$
\begin{aligned}
f_c &= n_c\lambda && (2.11)\\
f_o &= -A^T f_c = -A^T n_c\lambda && (2.12)\\
n_c &= R_A(\partial\Phi/\partial r_{co})^T / \|(\partial\Phi/\partial r_{co})^T\| && (2.13)
\end{aligned}
$$

where λ is a Lagrange multiplier related to the magnitude of the force.

2.2 Dynamic Modeling

To obtain the dynamic model of manipulator 2, the constraint object is treated as a part of the end-effector. The dynamic models of manipulators 1 and 2 are described by the following equations

$$
\begin{aligned}
M_1(q_1)\ddot{q}_1 + C_1(q_1,\ \dot{q}_1)\dot{q}_1 + G_1(q_1) &= \tau_1 + J_1^T(q_1)f_c = \tau_1 + J_1^T(q_1)n_c\lambda && (2.14)\\
M_2(q_2)\ddot{q}_2 + C_2(q_2,\ \dot{q}_2)\dot{q}_2 + G_2(q_2) &= \tau_2 + J_2^T(q_2)f_o = \tau_2 - J_2^T(q_2)A^T n_c\lambda && (2.15)
\end{aligned}
$$

where $M_i(q_i)$ is the inertia matrix, $C_i(q_i, \dot{q}_i)$ is the coriolis and centrifugal force matrix, $G_i(q_i)$ is the gravitational force, τ_i are the joint torques and $J_i(q_i)$ is the Jacobian matrix ($i = 1, 2$).

Combining equations (2.14) and (2.15) gives the following dynamic equation

$$M(q)\ddot{q} + C(q,\ \dot{q})\dot{q} + G(q) = \tau + J^T(q)n_c\lambda \qquad (2.16)$$

where

$$
\begin{aligned}
M(q) &= \begin{bmatrix} M_1(q_1) & 0 \\ 0 & M_2(q_2) \end{bmatrix},\ C(q,\ \dot{q}) = \begin{bmatrix} C_1(q_1\ \dot{q}_1) & 0 \\ 0 & C_2(q_2\ \dot{q}_2) \end{bmatrix}\\
G(q) &= \begin{bmatrix} G_1(q_1) \\ G_2(q_2) \end{bmatrix},\ q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix},\ \tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix},\ J(q) = [J_1(q_1) \quad -AJ_2(q_2)]
\end{aligned}
$$

Assume a set of *independent* n coordinates $q^1 = [q_1^1 \ldots q_n^1]^T$ are chosen from the joint variables q, such that q is the function of q^1, i.e.,

$$q = q(q^1) \qquad (2.17)$$

Differentiating equation (2.17) with respect to time t, we have

$$\begin{aligned} \dot{q} &= L(q^1)\dot{q}^1 && (2.18) \\ \ddot{q} &= L(q^1)\ddot{q}^1 + \dot{L}(q^1)\dot{q}^1 && (2.19) \end{aligned}$$

where $L(q^1) = \partial q/\partial q^1$. It is obvious that $L(q^1)$ is of full column rank.

Substituting equations (2.18) and (2.19) into equation (2.16), we obtain the following reduced order dynamic model of the system

$$M^1(q^1)\ddot{q}^1 + C^1(q^1,\ \dot{q}^1)\dot{q}^1 + G^1(q^1) = \tau + J^{1T}(q^1)n_c\lambda \qquad (2.20)$$

where

$$\begin{aligned} M^1(q^1) &= M(q^1)L(q^1) \\ C^1(q^1,\ \dot{q}^1) &= M(q^1)\dot{L}(q^1) + C(q^1,\ \dot{q}^1)L(q^1) \\ G^1(q^1) &= G(q(q^1)) \\ J^1(q^1) &= J(q(q^1)) \end{aligned}$$

To facilitate controller design, the structural properties of dynamic model (2.20) are listed as follows.

Property 2.1 The terms $L(q^1)$, $J^1(q^1)$ and n_c satisfy the relationship:

$$L^T(q^1)J^{1T}(q^1)n_c = 0$$

Property 2.2 The term $M_L(q^1) \triangleq L^T(q^1)M^1(q^1)$ is symmetric positive definite ($s.p.d$), and bounded upper and below.

Property 2.3 Define $C_L(q^1,\ \dot{q}^1) = L^T(q^1)C^1(q^1,\ \dot{q}^1)$, then $N_L = \dot{M}_L(q^1) - 2C_L(q^1,\ \dot{q}^1)$ is skew-symmetric if $C_i(q_i,\ \dot{q}_i)(i = 1, 2)$ is in the Christoffel form, i.e., $x^T N_L x = 0$, $\forall x \in R^n$.

Property 2.4 The dynamics described by equation (2.20) is linear in parameters, i.e.,

$$M^1(q^1)\ddot{\chi} + C^1(q^1,\ \dot{q}^1)\dot{\chi} + G^1(q^1) = \Psi P \qquad (2.21)$$

where $P \in R^l$ are the parameters of interest, $\Psi = \Psi(q^1, \dot{q}^1, \dot{\chi}, \ddot{\chi}) \in R^{n\times l}$ is the regressor matrix, and $\dot{\chi}$, $\ddot{\chi} \in R^n$.

Properties 2.2, **2.3** and **2.4** can be easily derived from the properties of the dynamic model of a single robot ([13]). The proof of **Property 2.1** can be found in **Appendix A**.

2.3 Controller Design

In this section, the model-based adaptive controller is developed for the case when the system parameters are unknown, followed by the model-free neural network based adaptive controller in which there is no need for the derivation of the known regressor $\Psi(*)$.

Let $r_{od}(t)$ be the desired trajectory of the object, $r_{cod}(t)$ be the desired trajectory on the object and $\lambda_d(t)$ be the desired constraint force. The first control objective is to drive the manipulators such that $r_o(t)$ and $r_{co}(t)$ track their desired trajectories $r_{od}(t)$ and $r_{cod}(t)$ respectively, accordingly it is only necessary to make $q^1(t)$ track the desired trajectory $q^1_d(t)$ since $q^1(t)$ completely determines $r_o(t)$ and $r_{co}(t)$. The second objective is to make $\lambda(t)$ to track its desired trajectory $\lambda_d(t)$.

In practice, the parameters of the system are usually unknown. Let $\hat{P}$ be the estimates of parameters P, and $\tilde{P} = P - \hat{P}$. Define the following variables for the ease of discussion

$$e^1 = q^1_d - q^1 \tag{2.22}$$

$$e_\lambda = \lambda_d - \lambda \tag{2.23}$$

$$r^1 = \dot{e}^1 + K_e e^1 \tag{2.24}$$

$$\dot{q}^1_r = \dot{q}^1_d + K_e e^1 \tag{2.25}$$

where constant $K_e \in R^{n\times n}$ is positive definite. It is obvious that

$$r^1 = \dot{q}^1_r - \dot{q}^1 \tag{2.26}$$

2.3.1 Model-based Adaptive Control

For dynamic system (2.20), consider the following controller

$$\tau = \Psi_r \hat{P} - J^{1T}(q^1) n_c (\lambda_d + k_\lambda \int_0^t e_\lambda(\tau) d\tau) + L^{-T}(q^1) K r^1 \tag{2.27}$$

where constants $k_\lambda \in R$ and $K \in R^{n \times n}$ are all positive definite, and

$$\Psi_r = \Psi(q^1, \dot{q}^1, \dot{q}_r^1, \ddot{q}_r^1)$$

Applying control law (2.27) to dynamic system (2.20), the closed-loop dynamics are obtained

$$M^1(q^1)\ddot{q}^1 + C^1(q^1,\ \dot{q}^1)\dot{q}^1 + G^1(q^1) = \Psi_r \hat{P} - J^{1T}(q^1) n_c (e_\lambda + k_\lambda \int_0^t e_\lambda(\tau) d\tau) + L^{-T}(q^1) K r^1 \tag{2.28}$$

From **Property 2.4**, we know that

$$M^1(q^1)\ddot{q}^1 + C^1(q^1,\ \dot{q}^1)\dot{q}^1 + G^1(q^1) = \Psi_0 P \tag{2.29}$$

where

$$\Psi_0 = \Psi(q^1, \dot{q}^1, \dot{q}^1, \ddot{q}^1) \tag{2.30}$$

Combining equations (2.28) and (2.29) leads to

$$J^{1T}(q^1) n_c (e_\lambda + k_\lambda \int_0^t e_\lambda(\tau) d\tau) = L^{-T}(q^1) K r^1 - \Psi_r \tilde{P} + (\Psi_r - \Psi_0) P \tag{2.31}$$

Pre-multiplying both sides of equation (2.28) by $L^T(q^1)$ and using **Property 2.1**, we have

$$M_L(q^1)\ddot{q}^1 + C_L(q^1, \dot{q}^1)\dot{q}^1 + G_L(q^1) = L^T(q^1)\Psi_r \hat{P} + K r^1 \tag{2.32}$$

Pre-multiplying equation (2.21) by $L^T(q^1)$ and noting the change of variables, we have

$$M_L(q^1)\ddot{q}_r^1 + C_L(q^1, \dot{q}^1)\dot{q}_r^1 + G_L(q^1) = L^T(q^1)\Psi_r P \tag{2.33}$$

By subtracting equation (2.32) from equation (2.33) and using equation (2.26), it yields

$$M_L(q^1)\dot{r}^1 + C_L(q^1,\ \dot{q}^1) r^1 + K r^1 = L^T(q^1)\Psi_r \tilde{P} \tag{2.34}$$

Note that equation (2.34) describes the dynamic behavior of the tracking errors r^1, whereas equation (2.31) describes the behavior of the force tracking error e_λ. It is obvious that r^1 is mainly affected by the parameter estimation errors $\tilde{P}$; while the force error e_λ is affected by both $\tilde{P}$ and the term $\Psi_r - \Psi_0$ resulted from the tracking errors e^1. For the convergence of the tracking errors e^1 and e_λ, we have the following theorem.

Theorem 2.3.1 *For the closed-loop dynamic system (2.32), if the parameters are updated by*

$$\dot{\hat{P}} = \Gamma \Psi_r^T L(q^1) r^1 \tag{2.35}$$

where Γ *is a constant positive definite matrix, then* $e^1 \to 0$ *and* e_λ *is bounded as* $t \to \infty$*, and all the closed loop signals are bounded.*

Proof:

Choose the following Lyapunov function candidate

$$V = \frac{1}{2} r^{1T} M_L(q^1) r^1 + \frac{1}{2} \tilde{P}^T \Gamma^{-1} \tilde{P} \tag{2.36}$$

Differentiating equation (2.36) with respect to time t gives rise to

$$\dot{V} = r^{1T} M_L(q^1) \dot{r}^1 + \frac{1}{2} r^{1T} \dot{M}_L(q^1) r^1 + \tilde{P}^T \Gamma^{-1} \dot{\tilde{P}} \tag{2.37}$$

From **Property 2.3**, we have

$$\dot{V} = r^{1T} (M_L(q^1) \dot{r}^1 + C_L(q^1,\ \dot{q}^1) r^1) + \tilde{P}^T \Gamma^{-1} \dot{\tilde{P}} \tag{2.38}$$

From equation (2.34), we obtain

$$\dot{V} = \tilde{P}^T \Gamma^{-1} (\Gamma \Psi_r^T L(q^1) r^1 - \dot{\hat{P}}) - r^{1T} K r^1 \tag{2.39}$$

where the fact that $\dot{\tilde{P}} = -\dot{\hat{P}}$ has been used.

Substituting the adaptation law (2.35) into the above equation leads to

$$\dot{V} = -r^{1T} K r^1 \tag{2.40}$$

As $K > 0$, $\dot{V} \leq 0$, thus $r^1 \in L_2^n$. From the definition of r^1 in equation (2.24), $e^1 \to 0$, $q^1(t) \to q_d^1(t)$ as $t \to \infty$, and $\dot{e}^1 \in L_2^n$. From the closed kinematics (2.17), we can conclude that $q \to q_d$ when $t \to \infty$. Obviously the same conclusion cannot be made for $\tilde{P}$, but it is bounded in the sense of Lyapunov stability.

Because $e^1 \to 0$, $\dot{e}^1 \in L_2^n$ and $\tilde{P}$ is bounded, it can be concluded that $\dot{r}^1 \in L_\infty^n$ from equation (2.34). It has been proven that $r^1 \in L_2^n$, thus $r^1 \to 0$ as $t \to \infty$. From the definition of r^1 in equation (2.24), we have $\dot{e}^1 \to 0$, $e^1 \to 0$ as $t \to \infty$.

Because $r^1 \to 0$, $e^1 \to 0$, $\dot{e}^1 \to 0$ and $\tilde{P}$ is bounded when $t \to \infty$, from the definitions of $\dot{q}_r^1$, r, Ψ_r and Ψ_0, we can conclude that the right hand side of equation (2.31) is bounded, thus e_λ is bounded and its size can be adjusted by choosing a proper gain matrix k_λ. The integral of the force error is for reducing its static error.

Q.E.D.

Controller (2.27) and adaptation law (2.35) guarantee $e^1 \to 0$, but they can only make the force error e_λ bounded. Before proceeding on a way to make e_λ converge to zero, the following definitions and lemmas in [47] can be used and are reproduced below for the completeness of the presentation.

Definition 1 [47] *Almost Everywhere Uniform Continuity (a.e.u.c)*: A function $f(t) : R^+ \to R^n$ is said to be uniformly continuous almost everywhere *iff* for any given t_0 and any given ε there exist $\delta(\varepsilon)$ such that

$$\|f(t) - f(t_0)\| \leq \varepsilon \text{ for all } t \in [t_0,\ t_0 + \delta] \textit{ or } t \in [t_0 - \delta,\ t_0] \tag{2.41}$$

Definition 2 [47]*Persistent Excitation*: A matrix function $W(t) : R^+ \to R^{m \times n}(m \leq n)$ is said to be persistently exciting (P.E.) *iff* there exist a $\delta > 0$ and an $\alpha > 0$ such that for all $t \in R^+$ we have

$$\int_t^{t+\delta} W^T(\tau) W(\tau) d\tau \geq \alpha I \tag{2.42}$$

Lemma 2.3.1 [47] *Let $f(t) : R^+ \to R^n$ be a uniformly continuous almost everywhere (u.c.a.e) function. Then for any $p_0 > 0$,*

$$\lim_{t \to \infty} f(t) = 0 \textit{ iff } \lim_{t \to \infty} \int_t^{t+p} f(\tau) d\tau = 0 \textit{ for all } 0 < p < p_0 \tag{2.43}$$

Now we are ready to present the following theorem about the convergence of e_λ.

Theorem 2.3.2 *For the closed loop system consisting of dynamic model (2.20), control law (2.27) and adaptation law (2.35), if*

1. $\ddot{q}_d^1$ *is uniformly continuous almost everywhere (u.c.a.e.), and*
2. $\Psi_{Ld} = L^T(q_d^1)\Psi(q_d^1, \dot{q}_d^1, \dot{q}_d^1, \ddot{q}_d^1)$ *is persistently exciting,*

then $e_\lambda \rightarrow 0$ *as* $t \rightarrow \infty$.

Proof:

For clarity, define the following terms

$$\begin{aligned}
\Psi_{Lr} &= L^T(q^1)\Psi(q^1, \dot{q}^1, \dot{q}_r^1, \ddot{q}_r^1) \\
f(t) &= \Psi_{Lr}\tilde{P}(t) \\
f_1(t) &= M_L(q^1)\dot{r}^1(t) \\
f_2(t) &= C_L(q^1,\ \dot{q}^1)r^1(t) + Kr^1(t)
\end{aligned}$$

Taking the same approach as in [47], equation (2.34)is rewritten as

$$f_1(t) + f_2(t) = f(t) \tag{2.44}$$

Integrating both sides of equation (2.44) in the interval $[t, t+p]$ $(0 \leq p \leq p_0)$, it follows that

$$\int_t^{t+p} f_1(\tau)d\tau + \int_t^{t+p} f_2(\tau)d\tau = \int_t^{t+p} f(\tau)d\tau \tag{2.45}$$

and

$$\int_t^{t+p} f_1(\tau)d\tau = \int_t^{t+p} M_L(q^1)(\tau)\dot{r}^1(\tau)d\tau \tag{2.46}$$

$$\int_t^{t+p} f_2(\tau)d\tau = \int_t^{t+p} [C_L(q^1,\ \dot{q}^1)r^1(\tau) + Kr^1(\tau)]d\tau \tag{2.47}$$

By expanding the integral $\int_t^{t+p} f_1(\tau)d\tau$, we have

$$\int_t^{t+p} f_1(\tau)d\tau = M_L(q^1)(t+p)r^1(t+p) - M_L(q^1)(t)r^1(t) - \int_t^{t+p} \dot{M}_L(q^1)(\tau)r^1(\tau)d\tau \tag{2.48}$$

which leads to

$$\begin{aligned} \| \int_t^{t+p} f_1(\tau) d\tau \| &\leq \sup_{q^1} \|M_L(q^1)\| (\|r^1(t+p)\| + \|r^1(t)\|) \\ &+ p_0 (\sup_{t \leq \tau \leq t+p}) \|\dot{M}_L(q^1)(\tau)\| \|r^1(\tau)\|) \end{aligned} \tag{2.49}$$

Note that $\dot{M}_L(q^1)(t)$ can be written as

$$\dot{M}_L(q^1)(\tau) = \sum_{i=1}^{n} \frac{\partial M_L(q_i^1)}{\partial q_i^1}(\tau) \dot{q}_i^1 \tag{2.50}$$

As proved in **Theorem 2.3.1**, $r^1 \to 0$, $e^1 \to 0$ and $\dot{e}^1 \to 0$ when $t \to \infty$, thus $\dot{M}_L(q^1)(\tau)$ is bounded. In addition, $\sup_{q^1} \|M_L(q^1)\|$ is bounded from **Property 2.1**. Therefore

$$\lim_{t \to \infty} \int_t^{t+p} f_1(\tau) d\tau = 0 \tag{2.51}$$

From **Lemma 2.3.1**, we have

$$\lim_{t \to \infty} f_1(t) = 0 \tag{2.52}$$

From the fact that $r^1 \to 0$ when $t \to \infty$, it is obvious that

$$\lim_{t \to \infty} f_2(t) = 0 \tag{2.53}$$

From equation (2.52) and (2.53), we have

$$\lim_{t \to \infty} f(t) = \lim_{t \to \infty} (f_1(t) + f_2(t)) = \lim_{t \to \infty} \Psi_{Lr} \tilde{P}(t) = 0 \tag{2.54}$$

Consider the following inequality

$$\|\Psi_{Ld} \tilde{P}\| \leq \|\Psi_{Ld} - \Psi_{Lr}\| \|\tilde{P}\| + \|\Psi_{Lr} \tilde{P}\| \tag{2.55}$$

For $r^1 \to 0$ when $t \to \infty$, and $\dot{q}_r^1 = \dot{q}_d^1 + K_e e^1$, $\ddot{q}_r^1 = \ddot{q}_d^1 + K_e \dot{e}^1$, we have

$$\lim_{t \to \infty} \|\Psi_{Lr} - \Psi_{Ld}\| = 0 \tag{2.56}$$

From equations (2.54) — (2.56), we conclude that

$$\Psi_{Ld} \tilde{P} \to 0 \text{ when } t \to \infty$$

Let $Q(t,t+\delta) = \int_t^{t+\delta} \Psi_{Ld}^T(\tau)\Psi_{Ld}(\tau)d\tau$. Since Ψ_{Ld} is persistent exciting, then for some $\delta > 0$ and all t, we have

$$Q(t,t+\delta) \geq \alpha I > 0 \tag{2.57}$$

From adaptation law (2.35) and the integration by parts, we obtain,

$$\begin{aligned} \tilde{P}^T(t)Q(t,t+\delta)\tilde{P}(t) &= -2\int_t^{t+\delta} \tilde{P}^T(\tau)Q(\tau,\tau+\delta)\Gamma\Psi_{Lr}r^1(\tau)d\tau \\ &+ \int_t^{t+\delta} \tilde{P}^T(\tau)\Psi_{Ld}^T(\tau)\Psi_{Ld}(\tau)\tilde{P}(\tau)d\tau \end{aligned}$$

From equation (2.57) and the fact that $r^1 \to 0$ when $t \to \infty$ proven in **Theorem 2.3.1**, we can see that the right-hand side of the above equation converges to zero as $t \to \infty$. Since $Q(t,t+\delta) \geq \alpha I > 0$, then it can be concluded that $\tilde{P} \to 0$ as $t \to \infty$.

It has been proven that $r^1 \to 0$, $e^1 \to 0$ and $\dot{e}^1 \to 0$ as $t \to \infty$ in **Theorem 2.3.1**. With $\tilde{P} \to 0$, we can conclude that $\Psi_r - \Psi_0 \to 0$ as $t \to \infty$. Thus, from equation (2.31), we have

$$J^{1T}(q^1)n_c(e_\lambda + k_\lambda \int_0^t e_\lambda(\tau)d\tau) \to 0 \tag{2.58}$$

As $J^{1T}(q^1)$ is of full column rank, we conclude that

$$e_\lambda + k_\lambda \int_0^t e_\lambda(\tau)d\tau \to 0 \tag{2.59}$$

which leads to $e_\lambda(\tau) \to 0$ as $t \to \infty$ for $k_\lambda > 0$.

Q.E.D.

Remark 2.3.1 *The condition for the convergence of force is more stringent than those for the convergence of position. It requires that the trajectory q_d^1 be planned such that $\ddot{q}_d^1$ and $L^T(q_d^1)\Psi(q_d^1, \dot{q}_d^1, \dot{q}_d^1, \ddot{q}_d^1)$ meet the conditions listed in* **Theorem 2.3.2**.

Remark 2.3.2 *The above model-based adaptive controller relies on accurate dynamic modeling of the system. The calculation of regressor matrix Ψ is very time consuming. To eliminate the need for dynamic modeling, a model-free adaptive neural network controller is presented in the next section.*

2.3.2 Neural Network Based Controller

It is well known that the Gaussian radial-basis function (RBF) neural network can be used to approximate any smooth function [66]. For a given smooth function $F(x) : R^n \rightarrow R^m$, there exist optimal parameters $w_{ji} \in R$ such that

$$\begin{aligned} \hat{F}_j(x) &= \sum_{i=1}^{l} w_{ji} a_i(x) = w_j^T a(x) \ \ (j = 1, 2, \dots m) && (2.60) \\ \hat{F}(x) &= [\hat{F}_1(x)\ \hat{F}_2(x) ... \hat{F}_m(x)]^T && (2.61) \\ F(x) &= \hat{F}(x) + \epsilon(x) && (2.62) \end{aligned}$$

where $\epsilon(x)$ is the minimum approximation error and $a_i(x)$ $(i = 1, 2, \dots l)$ are the Gaussian functions defined as

$$a_i(x) = exp(\frac{-(x - \mu_i)^T (x - \mu_i)}{\sigma^2}) \tag{2.63}$$

with $\mu_i \in R^n$ being the centers of the functions, and $\sigma^2 \in R$ being the variance.

Equation (2.60) can be expressed in a matrix form as follows

$$\hat{F}(x) = W^T a(x) \tag{2.64}$$

where $W = [w_1 \ \ w_2 \ \dots w_m]^T$.

The above RBF neural network is schematically shown in Figure 2.2. It has an input layer, a hidden layer and an output layer. In the hidden layer, each node contains a Gaussian function $a_i(x)$. Note that only the connections between the hidden layer and the output layer are weighted by w_{ji}.

Consider the reduced dynamic model (2.20) and let $m_{kj}^1(q^1)$ and $c_{kj}^1(q^1,\ \dot{q}^1)$ denote the kjth element of matrices $M^1(q^1)$ and $C^1(q^1,\ \dot{q}^1)$, respectively, and $g_k^1(q^1)$ be the kth element of $G^1(q^1)$. Let $M^1(q^1)$, $C^1(q^1,\ \dot{q}^1)$ and $G^1(q^1)$ be approximated by the following RBF neural networks [24]:

$$\begin{aligned} m_{kj}^1(q^1) &= \theta_{kj}^T \xi_{kj}(q^1) + \epsilon_{mkj}(q^1) && (2.65) \\ c_{kj}^1(q^1,\ \dot{q}^1) &= \alpha_{kj}^T \zeta_{kj}(z) + \epsilon_{ckj}(z) && (2.66) \\ g_k^1(q^1) &= \beta_k^T \eta_k + \epsilon_{gk}(q^1) && (2.67) \end{aligned}$$

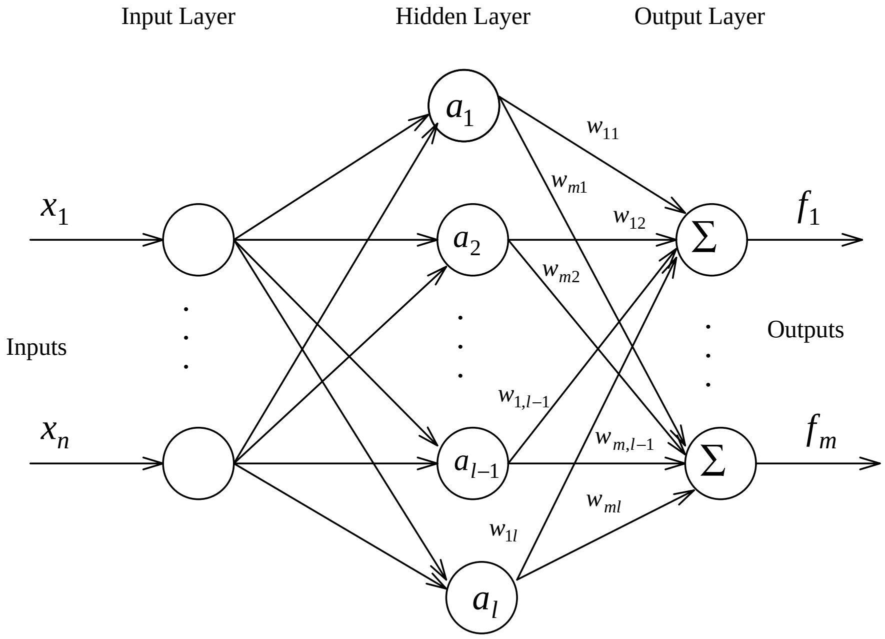

Figure 2.2: RBF neural network

where $z = [(q^1)^T \ (\dot{q}^1)^T]^T \in R^{2n}$, $\xi_{kj}(q^1) \in R^{l_{Mkj}}$, $\zeta_{kj}(z) \in R^{l_{Ckj}}$ and $\eta_k(q^1) \in R^{l_{Gk}}$ are the vectors of Gaussian functions defined in equation (2.63); $\theta_{kj} \in R^{l_{Mkj}}$, $\alpha_{kj} \in R^{l_{Ckj}}$ and $\beta_k \in R^{l_{Gk}}$ are the vectors of optimal weights of the neural network which make the modeling errors $\epsilon_{mkj}(q^1)$, $\epsilon_{ckj}(z)$ and $\epsilon_{gk}(q^1)$ be minimum. To simplify the above algebraic expressions of neural networks, we adopt the notation of GL matrix [24] in the following discussion. A GL matrix is normally expressed in a form $\{*\}$ to differentiate it from a normal matrix $[*]$. The unique characteristics of the GL matrices are that the transposes and the product of the matrices are done "locally". For example, given two GL matrices $\{\Theta\}$, $\{\Xi(q^1)\}$ and a normal square matrix Γ_k such that

$$\{\Theta\} \quad = \left\{ \begin{array}{cccc} \theta_{11} & \theta_{12} & \cdots & \theta_{1n} \\ \theta_{21} & \theta_{22} & \cdots & \theta_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{n1} & \theta_{n2} & \cdots & \theta_{nn} \end{array} \right\} = \left\{ \begin{array}{c} \{\theta_1\} \\ \{\theta_2\} \\ \vdots \\ \{\theta_n\} \end{array} \right\}$$

$$
\begin{aligned}
\{\Xi(q^1)\} &= \left\{ \begin{array}{cccc} \xi_{11}(q^1) & \xi_{12}(q^1) & \cdots & \xi_{1n}(q^1) \\ \xi_{21}(q^1) & \xi_{22}(q^1) & \cdots & \xi_{2n}(q^1) \\ \vdots & \vdots & \ddots & \vdots \\ \xi_{n1}(q^1) & \xi_{n2}(q^1) & \cdots & \xi_{nn}(q^1) \end{array} \right\} = \left\{ \begin{array}{c} \{\xi_1\} \\ \{\xi_2\} \\ \vdots \\ \{\xi_n\} \end{array} \right\} \\
\Gamma_k &= \Gamma_k^T = [\gamma_{k1}\ \gamma_{k2}\ \cdots \gamma_{kn}]
\end{aligned}
$$

we have

$$
\begin{aligned}
\{\Theta\}^T &= \left\{ \begin{array}{cccc} \theta_{11}^T & \theta_{12}^T & \cdots & \theta_{1n}^T \\ \theta_{21}^T & \theta_{22}^T & \cdots & \theta_{2n}^T \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{n1}^T & \theta_{n2}^T & \cdots & \theta_{nn}^T \end{array} \right\} \\
\{\Theta\}^T \bullet \{\Xi(q^1)\} &= \left[\begin{array}{cccc} \theta_{11}^T\xi_{11}(q^1) & \theta_{12}^T\xi_{12}(q^1) & \cdots & \theta_{1n}^T\xi_{1n}(q^1) \\ \theta_{21}^T\xi_{21}(q^1) & \theta_{22}^T\xi_{22}(q^1) & \cdots & \theta_{2n}^T\xi_{2n}(q^1) \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{n1}^T\xi_{n1}(q^1) & \theta_{n2}^T\xi_{n2}(q^1) & \cdots & \theta_{nn}^T\xi_{nn}(q^1) \end{array} \right] \\
\Gamma_k \bullet \{\xi_k\} &= \{\Gamma_k\} \bullet \{\xi_k\} := [\gamma_{k1}\xi_{k1}\ \gamma_{k2}\xi_{k2} \cdots \gamma_{kn}\xi_{kn}]
\end{aligned}
$$

where $\{*\} \bullet \{*\}$ represents the multiplication of two GL matrices and $[*] \bullet \{*\}$ represents the multiplication of a square matrix with a GL matrix of compatible dimension. Note that their products are all normal matrices.

Let α_{kj} and ζ_{kj} be the kjth elements of GL matrices $\{A\}$ and $\{Z(z)\}$ respectively; β_k and η_k be the kth elements of the GL matrices $\{B\}$ and $\{H(q^1)\}$ respectively. By using these GL matrices defined, equations (2.65) – (2.67) can be rewritten as follows

$$
\begin{aligned}
M^1(q^1) &= [\{\Theta\}^T \bullet \{\Xi(q^1)\}] + E_M(q^1) && (2.68) \\
C^1(q^1,\ \dot{q}^1) &= [\{A\}^T \bullet \{Z(z)\}] + E_C(z) && (2.69) \\
G^1(q^1) &= [\{B\}^T \bullet \{H(q^1)\}] + E_G(q^1) && (2.70)
\end{aligned}
$$

where $E_M(q^1)$, $E_C(z)$ and $E_G(q^1)$ are matrices with $\epsilon_{mkj}(q^1)$, $\epsilon_{ckj}(z)$ and $\epsilon_{gk}(q^1)$ being their elements respectively.

Let the estimates of Θ, A and B be $\hat{\Theta}$, $\hat{A}$ and $\hat{B}$ respectively. The neural network estimates of $M^1(q^1)$, $C^1(q^1,\ \dot{q}^1)$ and $G^1(q^1)$ are expressed as follows

$$\begin{aligned}
\hat{M}^1{}_{nn}(q^1) &= [\{\hat{\Theta}\}^T \bullet \{\Xi(q^1)\}] && (2.71)\\
\hat{C}^1{}_{nn}(q^1,\ \dot{q}^1) &= [\{\hat{A}\}^T \bullet \{Z(z)\}] && (2.72)\\
\hat{G}^1{}_{nn}(q^1) &= [\{\hat{B}\}^T \bullet \{H(q^1)\}] && (2.73)
\end{aligned}$$

Consider the following neural network based controller

$$\begin{aligned}
\tau &= \hat{M}^1{}_{nn}(q^1)\ddot{q}^1_r + \hat{C}^1{}_{nn}(q^1,\ \dot{q}^1)\dot{q}^1_r + \hat{G}^1{}_{nn}(q^1) - J^{1T}(q^1)n_c(\lambda_d + k_\lambda \int_0^t e_\lambda dt)\\
&+ L^{-T}(q^1)(Kr^1 + K_s sgn(r^1)) \qquad (2.74)
\end{aligned}$$

where the control parameters k_λ, K and K_s are all positive definite.

Applying the control law (2.74) to the dynamic system (2.20), we have

$$\begin{aligned}
M^1(q^1)\ddot{q}^1 + C^1(q^1,\ \dot{q}^1)\dot{q}^1 + G^1(q^1) &= \hat{M}^1{}_{nn}(q^1)\ddot{q}^1_r + \hat{C}^1{}_{nn}(q^1,\ \dot{q}^1)\dot{q}^1_r + \hat{G}^1{}_{nn}(q^1)\\
&-J^{1T}(q^1)n_c(e_\lambda + k_\lambda \int_0^t e_\lambda dt) + L^{-T}(q^1)(Kr^1 + K_s sgn(r^1)) \qquad (2.75)
\end{aligned}$$

Multiplying both sides of equation (2.75) by $L^T(q^1)$ and making use of **Property 2.1**, we have

$$\begin{aligned}
&L^T(q^1)(M^1(q^1)\ddot{q}^1 + C^1(q^1,\ \dot{q}^1)\dot{q}^1 + G^1(q^1))\\
&= L^T(q^1)(\hat{M}^1{}_{nn}(q^1)\ddot{q}^1_r + \hat{C}^1{}_{nn}(q^1,\ \dot{q}^1)\dot{q}^1_r + \hat{G}^1(q^1)) + L^T(q^1)(Kr^1 + K_s sgn(r^1)) (2.76)
\end{aligned}$$

Substituting equations (2.68) —(2.73) into equation (2.76), it follows that

$$\begin{aligned}
&L^T(q^1)(M^1(q^1)\dot{r}^1 + C^1(q^1,\ \dot{q}^1)r^1) + Kr^1 + K_s sgn(r^1)\\
&= L^T([\{\tilde{\Theta}\}^T \bullet \{\Xi(q^1)\}]\ddot{q}^1_r + [\{\tilde{A}\}^T \bullet \{Z(z)\}]\dot{q}^1_r + [\{\tilde{B}\} \bullet \{H(q^1)\}]) + L^T(q^1)E(2.77)
\end{aligned}$$

where

$$E = L^T(q^1)(E_M(q^1)\ddot{q}^1_r + E_C(q^1,\ \dot{q}^1)\dot{q}^1_r + E_G(q^1)) \qquad (2.78)$$

and $(\tilde{*}) = (*) - (\hat{*})$.

Equation (2.77) describes the dynamic behavior of the tracking errors r^1 under the proposed controller. The right hand side of the equation is a function of neural network estimations. The dynamic behavior of the force variable λ is described in equation (2.75), which is directly affected by the error r^1.

For the convergence of r^1, e^1 and the boundedness of e_λ, we have the following theorem

Theorem 2.3.3 *For the closed-loop dynamic system (2.77) with $K_s > \|E\|$, if the terms $M^1(q^1)$, $C^1(q^1,\ \dot{q}^1)$ and $G(q^1)$ are approximated by the neural networks (2.71), (2.72) and (2.73) respectively with the weight matrices being updated as*

$$\begin{aligned}\dot{\hat{\theta}}_k &= \Gamma_k \bullet \{\xi_k(q^1)\}\ddot{q}_r^1(L(q^1)r^1)_k && (2.79)\\ \dot{\hat{\alpha}}_k &= Q_k \bullet \{\zeta_k(z)\}\dot{q}_r^1(L(q^1)r^1)_k && (2.80)\\ \dot{\hat{\beta}}_k &= U_k\eta_k(q^1)(L(q^1)r^1)_k && (2.81)\end{aligned}$$

where $\Gamma_k = \Gamma_k^T > 0$, $Q_k = Q_k^T > 0$ and $U_k = U_k^T > 0$, and $\hat{\theta}_k$, $\hat{\alpha}_k$, $\hat{\beta}_k$, $\xi_k(q^1)$, $\zeta_k(z)$, and $\eta_k(q^1)$ represents the kth column vector of the corresponding matrices $\{\hat{\Theta}\}$, $\{\hat{A}\}$, $\{\hat{B}\}$, $\{\Xi(q^1)\}$, $\{Z(z)\}$ and $\{H(q^1)\}$ respectively, then $\hat{\theta}_k$, $\hat{\alpha}_k$, $\hat{\beta}_k \in L_\infty$, $e^1 \in L_2^n \cap L_\infty^n$, e^1, $\dot{e}^1 \to 0$ and e_λ is bounded when $t \to \infty$.

Proof:

Choose a Lyapunov function

$$V = \frac{1}{2}r^{1T}M_L(q^1)r^1 + \frac{1}{2}\sum_{k=1}^{n}(\tilde{\theta}_k^T\Gamma_k^{-1}\tilde{\theta}_k + \tilde{\alpha}_k^TQ_k^{-1}\tilde{\alpha}_k + \tilde{\beta}_k^TU_k^{-1}\tilde{\beta}_k) \qquad (2.82)$$

where $\tilde{\theta}_k = \theta_k - \hat{\theta}_k$, $\tilde{\alpha}_k = \alpha_k - \hat{\alpha}_k$ and $\tilde{\beta}_k = \beta_k - \hat{\beta}_k$.

Differentiating equation (2.82), and noting that $\dot{\tilde{\theta}}_k = -\dot{\hat{\theta}}_k$, $\dot{\tilde{\alpha}}_k = -\dot{\hat{\alpha}}_k$ and $\dot{\tilde{\beta}}_k = -\dot{\hat{\beta}}_k$, we have

$$\dot{V} = r^{1T}M_L(q^1)\dot{r}^1 + \frac{1}{2}r^{1T}\dot{M}_L(q^1)r^1 - \sum_{k=1}^{n}(\tilde{\theta}_k^T\Gamma_k^{-1}\dot{\hat{\theta}}_k - \tilde{\alpha}_k^TQ_k^{-1}\dot{\hat{\alpha}}_k + \tilde{\beta}_k^TU_k^{-1}\dot{\hat{\beta}}_k) \quad (2.83)$$

From equations (2.79), (2.80) and (2.81) and **Property 2.3**, it follows that

$$\begin{aligned}\dot{V} \;&=\; r^{1T}M_L(q^1)\dot{r}^1 + r^{1T}C_L(q^1\ \dot{q}^1)r^1 - \sum_{k=1}^{n}\{\tilde{\theta}_k\}^T \bullet \{\xi_k(q^1)\}\ddot{q}_r^1(L(q^1)r^1)_k\\ &\quad - \sum_{k=1}^{n}\{\tilde{\alpha}_k\}^T \bullet \{\zeta_k(z)\}\dot{q}_r^1(L(q^1)r^1)_k - \sum_{k=1}^{n}\tilde{\beta}_k^T\eta_k(q^1))(L(q^1)r^1)_k \qquad (2.84)\end{aligned}$$

From the definition of the product of GL matrices, we have

$$\begin{aligned}\sum_{k=1}^{n}\{\tilde{\theta}_k\}^T \bullet \{\xi_k(q^1)\}\ddot{q}_r^1(L(q^1)r^1)_k &= r^{1T}L^T(q^1)[\{\tilde{\Theta}\}^T \bullet \{\Xi(q^1)\}]\ddot{q}_r^1 \quad (2.85)\\ \sum_{k=1}^{n}\{\tilde{\alpha}_k\}^T \bullet \{\zeta_k(z)\}\dot{q}_r^1(L(q^1)r^1)_k &= r^{1T}L^T(q^1)[\{\tilde{A}\}^T \bullet \{Z(z)\}]\dot{q}_r^1 \quad (2.86)\\ \sum_{k=1}^{n}\tilde{\beta}_k^T\eta_k(q^1)(L(q^1)r^1)_k &= r^{1T}L^T(q^1)[\{\tilde{B}\} \bullet \{H(q^1)\}] \quad (2.87)\end{aligned}$$

Substituting equations (2.85) – (2.87) into equation (2.84), we have

$$\begin{aligned}\dot{V} &= r^{1T}L^T(q^1)(M^1(q^1)\dot{r}^1 + C^1(q^1\ \dot{q}^1)r^1)\\ &\quad -r^{1T}L^T(q^1)([\{\tilde{\Theta}\}^T \bullet \{\Xi(q^1)\}]\ddot{q}_r^1 + [\{\tilde{A}\}^T \bullet \{Z(z)\}]\dot{q}_r^1 + [\{\tilde{B}\} \bullet \{H(q^1)\}]) \quad (2.88)\end{aligned}$$

From equation (2.77), we have

$$\dot{V} = -r^{1T}Kr^1 - (r^{1T}K_s sgn(r^1) - r^{1T}E) \quad (2.89)$$

As $K > 0$ and $K_s > \|E\|$, thus

$$\dot{V} \leq -r^{1T}Kr^1 \leq 0 \quad (2.90)$$

As $V > 0$ and $\dot{V} \leq 0$, $V \in L_\infty$. From the definition of V, it follows that $r^1 \in L_2^n$ and θ_k, α_k , $\beta_k \in L_\infty$. From the definition of r^1 in equation (2.24), $e^1 \to 0$, $q^1(t) \to q_d^1(t)$ as $t \to \infty$, and $\dot{e}^1 \in L_2^n$. From the closed kinematics (2.17), we can conclude that $q \to q_d$ when $t \to \infty$.

Because $e^1 \to 0$, $\dot{e}^1 \in L_2^n$ and θ_k, α_k , $\beta_k \in L_\infty$ as proved above, $E \in L_\infty$ from its definition (2.78). From equation (2.77), $\dot{r}^1 \in L_\infty^n$. It has been proven that $r^1 \in L_2^n$, thus $r^1 \to 0$ as $t \to \infty$. From the definition of r^1 in equation (2.24), we have $\dot{e}^1 \to 0$, $e^1 \to 0$ as $t \to \infty$. Based on the above conclusions, it is obvious that e_λ is bounded from equation (2.75).

Q.E.D

Remark 2.3.3 *The above neural network controller is called model free for it does not need the regressor matrix Ψ which relies on the accurate modeling of the robot dynamics.*

Remark 2.3.4 *The controller can only guarantee the boundedness of the force error e_λ. More stringent conditions are required to make it converge to zero including that $\ddot{q}_d^1$ must be uniformly continuous almost everywhere as discussed in* **Theorem 2.3.2.**

Remark 2.3.5 *The weights of the neural networks are updated on-line by the position tracking errors. The time-consuming off-line training of neural networks are thus not required.*

Remark 2.3.6 *The chattering caused by the sign function $sgn(r^1)$ is inevitable. Many effective methods are available to diminish the chattering, one of which is to introduce a boundary layer into the controller as suggested in [13][24].*

Remark 2.3.7 *Both the model-based controller (2.27) and the neural network controller (2.74) neglect the dynamics of the actuators of the robot and use the joint torques as the inputs. For better control performance at high operational speed, the actuator dynamics have to be taken into consideration [19][65]*

2.4 Simulation

The system used for simulation is schematically shown in Figure 2.3. The rectangular object is held rigidly by manipulator 2 which has only one degree of freedom and moves in the horizontal plane. The end effector of manipulator 1 of two degrees of freedom is to track a specified trajectory on the object.

The world coordinate is denoted by XOY, the object coordinate $X_oO_oY_o$ is at the object mass center O_o, the length, the mass and the moment of inertia of each link of manipulator 1 are denoted by d_i, m_i and I_i $(i = 1, 2)$ respectively. Let l_i $(i = 1, 2)$ be the distance of the mass center of each link from the respective joint. The mass of manipulator 2 together with the object is M_2. The joint variables for the two manipulators are $q_1 = [\theta_1 \ \theta_2]^T$ and $q_2 = x$ respectively. Note that x is actually a linear displacement of the object in the horizontal plane. The gravitational acceleration is denoted by $g = 9.8m/s^2$. For simplicity, let $c_i = \cos(\theta_i)$, $s_i = \sin(\theta_i)$, $c_{ij} = \cos(\theta_i + \theta_j)$, and $s_{ij} = \sin(\theta_i + \theta_j)$. From Figure 2.3, the following position vectors are derived

$$\begin{aligned} r_c &= [d_1c_1 + d_2c_{12} + a \quad d_1s_1 + d_2s_{12}]^T && (2.91)\\ r_{co} &= [d_1c_1 + d_2c_{12} - q_2 + a \quad d_1s_1 + d_2s_{12} - b]^T && (2.92)\\ r_o &= [q_2 \;\; b]^T && (2.93) \end{aligned}$$

where r_c and r_o are described with respect to the world coordinates, while r_{co} is described with respect to the object frame.

The trajectory on the object is assumed to be a straight line with reference to the object frame

$$\Phi(r_{co}) = x_{co} - y_{co} = 0 \qquad (2.94)$$

The inverse kinematic equations of manipulator 1 are given by

$$\begin{aligned} \theta_1 &= \arctan s_1/c_1 && (2.95)\\ \theta_2 &= \arctan s_2/c_2 && (2.96) \end{aligned}$$

where

$$\begin{aligned} c_2 &= ((x_c - a)^2 + y_c^2 - d_1^2 - d_2^2)/2d_1d_2 && (2.97)\\ s_2 &= \pm\sqrt{1 - c_2^2} && (2.98)\\ s_1 &= ((d_1 + d_2c_2)y_c - d_2s_2(x_c - a))/((x_c - a)^2 + y_c^2) && (2.99)\\ c_1 &= ((d_1 + d_2c_2)y_c + d_2s_2(x_c - a))/((x_c - a)^2 + y_c^2) && (2.100) \end{aligned}$$

Choose $q^1 = [\theta_1 \; \theta_2]^T$. It is obvious that

$$\begin{aligned} q_1(q^1) &= q^1 && (2.101)\\ q_2(q^1) &= d_1(c_1 - s_1) + d_2(c_{12} - s_{12}) + a + b && (2.102) \end{aligned}$$

From the above equations, the following quantities are derived

$$A = R_o = I^{2\times 2},\; n_{co} = n_c = \begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix},\; J_2(q_2) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\begin{aligned}
J_1(q_1) &= \begin{bmatrix} -d_1s_1 - d_2s_{12} & -d_2s_{12} \\ d_1c_1 + d_2c_{12} & d_2c_{12} \end{bmatrix} \\
L(q^1) &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -d_1(s_1+c_1) - d_2(s_{12}+c_{12}) & -d_2(s_{12}+c_{12}) \end{bmatrix} \\
\dot{L}(q^1) &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ d_1(s_1-c_1)\dot{\theta}_1 + d_2(s_{12}-c_{12})\dot{\theta}_{12} & d_2(s_{12}-c_{12})\dot{\theta}_{12} \end{bmatrix}
\end{aligned}$$

It can be verified that $L^T(q^1)J^{1T}(q^1)n_c = 0$ as stated in **Property 2.1**.

The dynamic model for manipulator 1 (two-link arm) is given by

$$M_1(q_1)\ddot{q_1} + C_1(q_1, \dot{q}_1)(q_1,\ \dot{q}_1)\dot{q}_1 + G_1(q_1)(q_1) = \tau_1 + J_1^T(q_1)f_c = \tau_1 + J_1^T(q_1)n_c\lambda \tag{2.103}$$

where

$$\begin{aligned}
M_1(q_1) &= \begin{bmatrix} I_1 + m_1l_1^2 + I_2 + m_2(d_1^2 + l_2^2 + 2d_1l_2c_2) & I_2 + m_2(l_2^2 + d_1l_2c_2) \\ I_2 + m_2(l_2^2 + d_1l_2c_2) & I_2 + m_2l_2^2 \end{bmatrix} \\
C_1(q_1, \dot{q}_1) &= \begin{bmatrix} -m_2d_1l_2s_2\dot{\theta}_2 & -m_2d_1l_2s_2(\dot{\theta}_1 + \dot{\theta}_2) \\ m_2d_1l_2s_2\dot{\theta}_1 & 0 \end{bmatrix} \\
G_1(q^1) &= [(m_1l_1 + m_2d_1)gc_1 + m_2l_2gc_{12} \quad m_2l_2gc_{12}]^T
\end{aligned}$$

The dynamic model for manipulator 2 is as follows

$$M_2\ddot{q}_2 = \tau_2 - J_2^T(q_2)A^Tn_c\lambda \tag{2.104}$$

The reduced dynamic model is described by

$$M^1(q^1)\ddot{q}^1 + C^1(q^1,\ \dot{q}^1)\dot{q}^1 + G^1(q^1) = \tau + J^{1T}(q^1)n_c\lambda \tag{2.105}$$

with all the terms as defined in equation (2.20).

Define $q_r^1 = [q_{r1}^1\ q_{r2}^1]^T$. The regressor matrix $\Psi(q^1, \dot{q}^1, \dot{q}_r^1\ \ddot{q}_r^1)$ and the parameter vector P in equation (2.21) are as follows

$$\Psi(q^1, \dot{q}^1, \dot{q}_r^1, \ddot{q}_r^1) = \begin{bmatrix} \ddot{\theta}_1 & \Psi_{12} & \ddot{q}_{r2}^1 & c_1 & c_{12} & 0 & 0 \\ 0 & \Psi_{22} & \Psi_{23} & 0 & c_{12} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \Psi_{36} & \Psi_{37} \end{bmatrix}$$

$$P = [p_1 \quad m_2 d_1 l_2 \quad I_2 + m_2 l_2^2 \quad (m_1 l_1 + m_2 d_1) g \quad m_2 l_2 g \quad M_2 d_1 \quad M_2 d_2]^T$$

where

$$\begin{aligned} \Psi_{12} &= 2c_2\ddot{q}_{r1}^1 + c_2\ddot{q}_{r2}^1 - s_2(\dot{\theta}_1 + \dot{\theta}_2)\dot{q}_{r2}^1 - s_2\dot{\theta}_2\dot{q}_{r1}^1 \\ \Psi_{22} &= c_2\ddot{q}_{r1}^1 + s_2\dot{\theta}_1\dot{q}_{r1}^1 \\ \Psi_{23} &= \ddot{q}_{r1}^1 + \ddot{q}_{r2}^1 \\ \Psi_{36} &= (s_1 - c_1)\dot{\theta}_1\dot{q}_{r1}^1 - (s_1 + c_1)\ddot{q}_{r1}^1 \\ \Psi_{37} &= (s_{12} - c_{12})(\dot{\theta}_1 + \dot{\theta}_2)(\dot{q}_{r1}^1 + \dot{q}_{r2}^1) - (s_{12} + c_{12})(\ddot{q}_{r1}^1 + \ddot{q}_{r2}^1) \\ p_1 &= I_1 + m_1 l_1^2 + I_2 + m_2(d_1^2 + l_2^2) \end{aligned}$$

Assume that the geometric parameters are $d_1 = d_2 = 0.3m$, $l_1 = l_2 = 0.15m$, $a = 0.2$ and $b = 0.5m$. The true values of the mass and inertia parameters are assumed to be $m_1 = m_2 = 0.1kg$, $M_2 = 0.2kg$, $I_1 = I_2 = 0.3kgm^2$, which are unknown for controller design. The true parameters are $P = [0.6 \;\; 0.04 \;\; 0.3 \;\; 0.44 \;\; 0.15 \;\; 0.06 \;\; 0.06]^T$, while the initial estimates of parameters are $\hat{P}(0) = [0.2 \;\; 0.01 \;\; 0.4 \;\; 0.4 \;\; 0.1 \;\; 0.2 \;\; 0.2]^T$.

The trajectory for the robot end effector to follow on the object is given by

$$x_{co} = y_{co} = -\frac{1}{12}\cos(t + 2) \tag{2.106}$$

The trajectory for the object to follow is given by

$$x_o = \frac{1}{10}(1 - \sin(4t + 12)) \tag{2.107}$$

The desired force is $\lambda_d = 2N$. From equations (2.97) and (2.98), we can obtain the desired q_d^1, $\dot{q}_d^1$ and $\ddot{q}_d^1$ which are required by the controller.

Note that the above two-link robot model is frequently used in the robotics literature. Its parameters (length of the link: $0.3m$ ($2l_1$), mass of the link: $0.1kg$ and

$0.2kg$ and the moment of inertia of the link: $0.3kgm^2$) are also within the ranges of the popular choices [16][17][18].

The constraint surface specified by equation (2.94) is similar to those used in the robotics literature where the constraint surfaces are defined in a form [16][17][18]:

$$ax + by + c = 0$$

The trajectories in equations (2.106) and (2.107) are the sine functions of time which are normally used to describe the trajectories in the robot systems [16][17][18]. The parameters in the equations are chosen by considering the workspace of the robot and the limits on its velocity and acceleration.

Simulation of the Adaptive Control Scheme

The gain matrices are chosen as $K_e = diag[20] \in R^{2\times2}$, $K = diag[15] \in R^{2\times2}$ and $k_\lambda = 15$. The adaptation gain matrix Γ in adaptation law (2.35) is chosen as $\Gamma = diag[15] \in R^{7\times7}$. The position tracking performances of the object and the force tracking performances are plotted in Figures 2.4 and 2.5 respectively. The control torques for the manipulators are given in Figure 2.6. From the figures, it can be seen that the position and force tracking errors approach to zero. The torque of the robotic arms are also in the reasonable range. We can conclude that the proposed adaptive controller effectively control the position and the force though the true parameters are unknown.

Simulation of the Neural Network Control Scheme

Based on the planned trajectories, the range of the angular displacement q^1 is $[-1.2, \;\; 1.7]$ rad and the range of the angular velocity $\dot{q}^1$ is $[-1.0, \;\; 1.0]$ rad/sec. The 2-dimensional input space for $\hat{M}_{nn}(q^1)$ and $\hat{G}_{nn}(q^1)$ is spanned by q^1 and the 4-dimensional input space for $\hat{C}_{nn}(q^1, \;\; \dot{q}^1)$ is spanned by $[q^1 \;\; \dot{q}^1]^T$. The centers of the RBF functions in the neural network are the crossing point of the grids evenly distributed in the input spaces of $\hat{M}_{nn}(q^1)$, $\hat{G}_{nn}(q^1)$ and $\hat{C}_{nn}(q^1, \;\; \dot{q}^1)$ respectively

[63]. In the simulation, a 120-node neural network with $\delta^2 = 40$ is used to estimate each element of $M^1(q^1)$, $C^1(q^1, \dot{q}^1)$ and $G^1(q^1)$ respectively. The controller gain matrices are chosen as $K_e = diag[20] \in R^{2\times 2}$, $K = diag[15] \in R^{2\times 2}$ and $k_\lambda = 15$. The boundary layer is chosen as $\|\Delta\| = 0.01$. The updating of the weights of the neural works are activated with $\Gamma_{kij} = 0.1$, $Q_{kij} = 0.2$ and $U_{kij} = 5.0$, $i = 120, k = 3, j = 2$. The position tracking performances of the object and the force tracking performances are plotted in Figures 2.7 and 2.8 respectively. The control torques for the manipulators are given in Figure 2.9. The neural network approximation performances are also shown from Figure 2.10 to Figure 2.12. From the simulation results, it can be seen that under the proposed adaptive neural network controller, the positions and the forces converge to their desired values and the torques are in the reasonable ranges. While $\hat{G}^1_{nn}$ almost converge to its true value G, $\hat{M}^1_{nn}$ and $\hat{C}^1_{nn}$ do not converge to M and C respectively. The approximation errors are affected by the persistent excitation condition. The overall performance of the controller is satisfactory with the model of the system unknown.

Comparing the performance of the neural network based adaptive controller with that of the model-based adaptive controller, there is not much difference in the accuracy and the speed of position tracking. While the force error is bounded without being convergent to zero in both controllers, the magnitudes of the fluctuations of the force signals are bigger under neural network based adaptive control than those under the model based adaptive controller, especially at the initial stage of control. The neural network based controller involves more matrix manipulations than model-based adaptive controller and its computing efficiency is relatively lower.

2.5 Conclusion

In most control schemes for constrained robots, the constraint is assumed to be motionless and its dynamics is neglected. In this chapter, a more general approach is taken for dynamic modeling and control of a constrained robotic manipulator where the constraint is treated to be in motion. In addition to the motion of

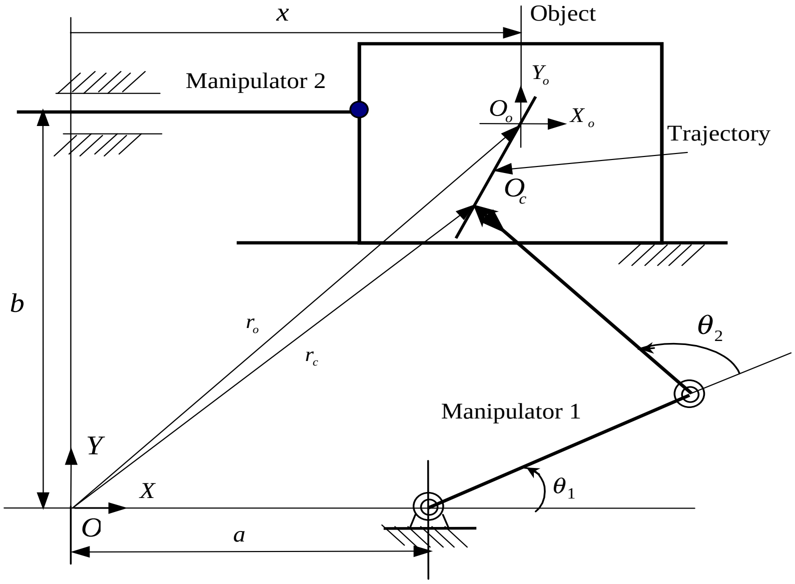

Figure 2.3: Simulation example

the constraint with respect to the world coordinate, its relative motion with respect to the manipulators is also taken into consideration The dynamic model of such a system is established and its properties are discussed. Both model-based and neural network based adaptive controllers are developed which guarantee the asymptotic convergence of positions, and boundedness of the constraint force. The condition for the convergence of the constraint force is also discussed. Among the two controllers developed, the neural network based adaptive controller is model free and is more suitable for the applications where the dynamic modeling is difficult. Simulation results are presented to verify the effectiveness of the proposed controllers.

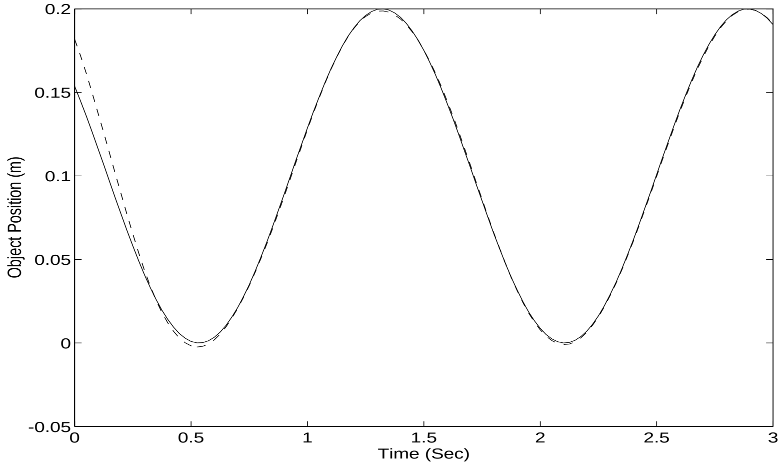

Figure 2.4: Position tracking under adaptive control (Solid: $r_d(t)$; Dashed: $r(t)$)

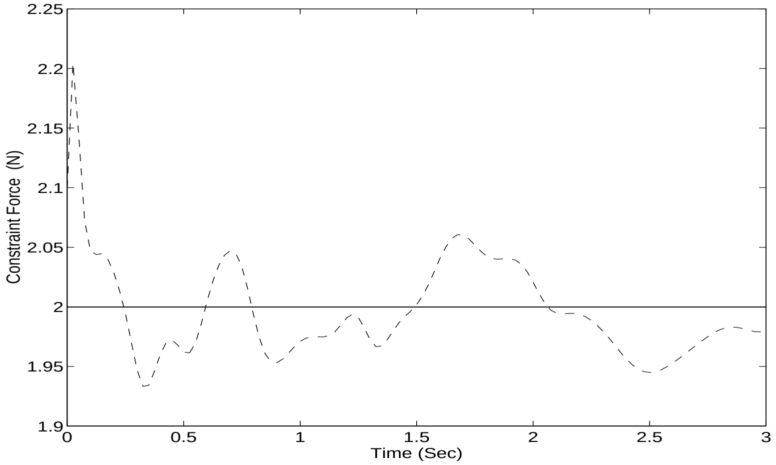

Figure 2.5: Constraint force tracking under adaptive control (Solid: $\lambda_d(t)$; Dashed: $\lambda(t)$)

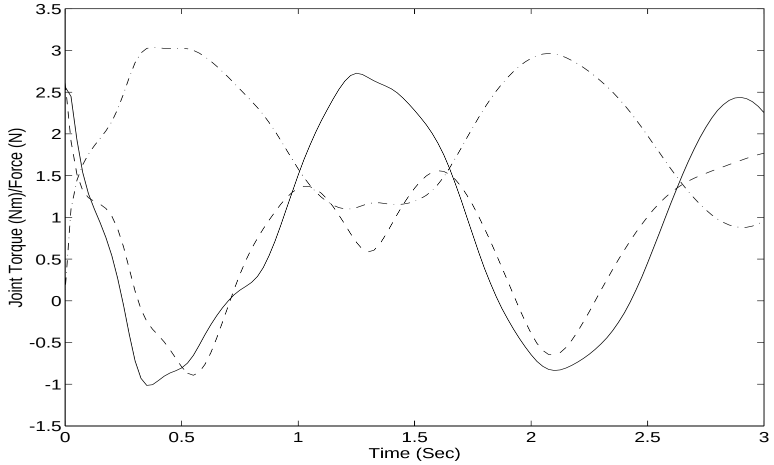

Figure 2.6: Torques/forces of the manipulators under adaptive control (Solid and Dashed: τ_1; Dash dotted: τ_2)

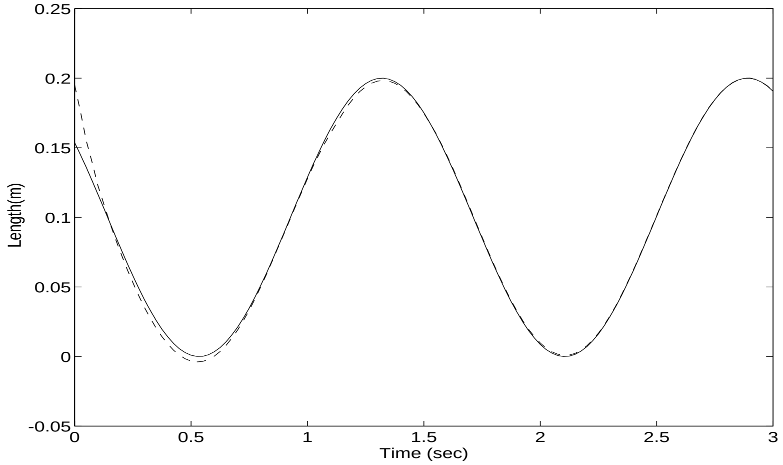

Figure 2.7: Object position tracking under neural network control (Solid: $r_d(t)$; Dashed: $r(t)$)

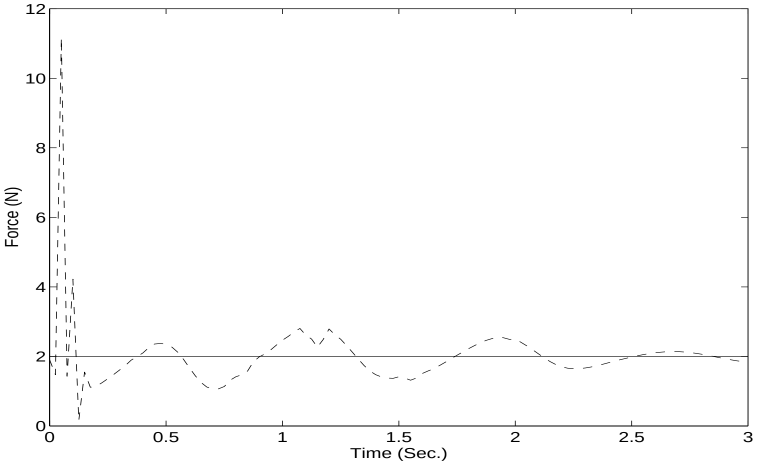

Figure 2.8: Constraint force tracking under neural network control (Solid: $\lambda_d(t)$; Dashed: $\lambda(t)$)

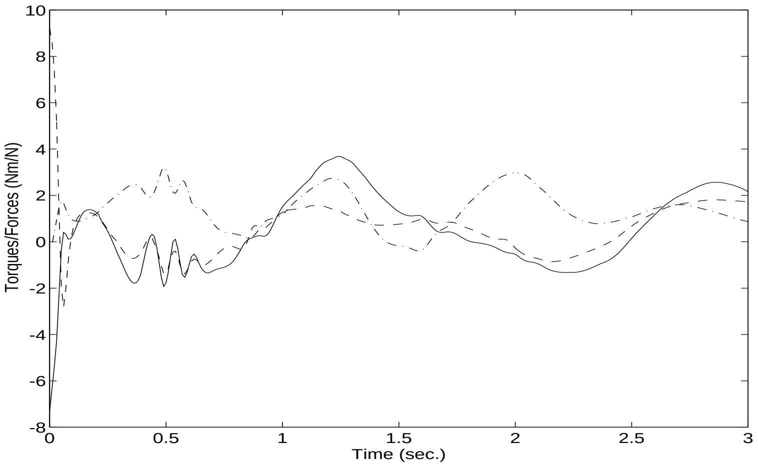

Figure 2.9: Torques/forces of the manipulators under neural network control (Solid and Dashed: τ_1; Dash dotted: τ_2)

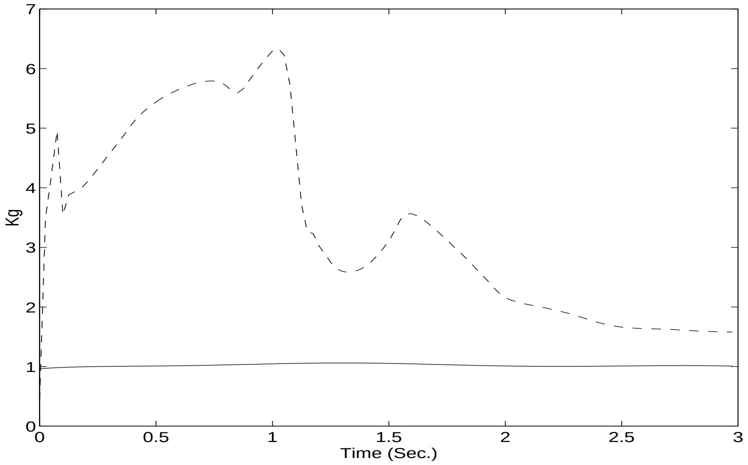

Figure 2.10: The approximation of M^1 (Solid: $\|M^1\|$; Dashed: $\|\hat{M}^1\|$)

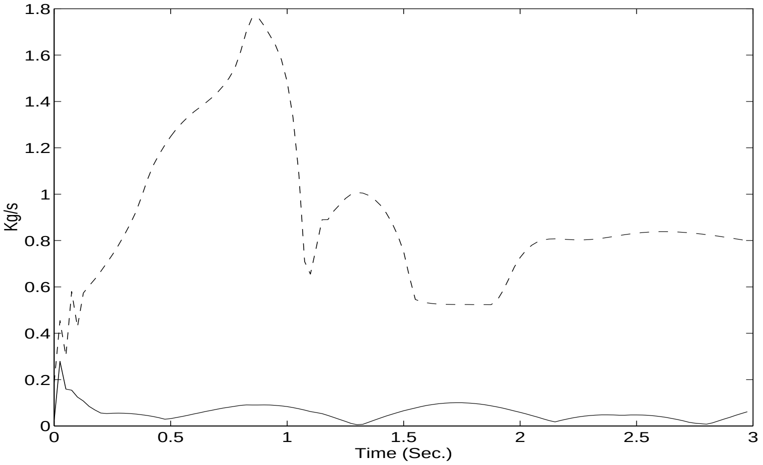

Figure 2.11: The approximation of C^1 (Solid: $\|C^1\|$; Dashed: $\|\hat{C}^1\|$)

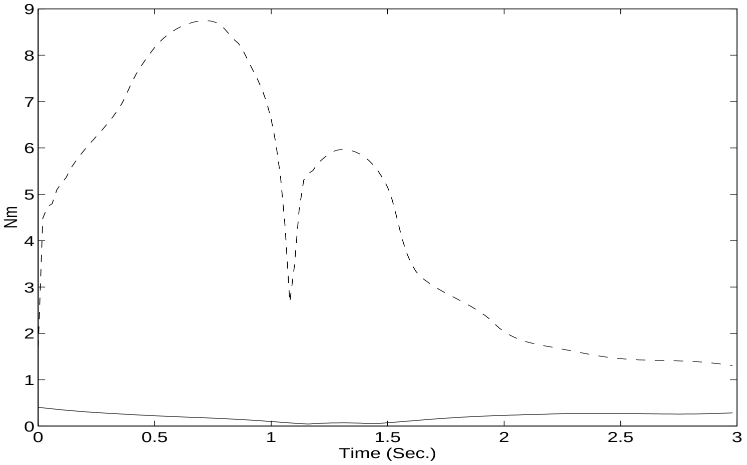

Figure 2.12: The approximation of G^1 (Solid: $\|G^1\|$; Dashed: $\|\hat{G}^1\|$)

Chapter 3

Robust Adaptive and NN Based Impedance Control

Impedance control is one of the important control approaches for constrained robots. It covers both constrained and unconstrained motion of the robot and demonstrates good robustness to uncertainties and disturbances [4]. These advantages make it very useful to many practical applications such as grinding and deburring mechanical parts. One of the challenges in applying impedance control is that it cannot guarantee the tracking of the robot's position and the constraint force to their desired values as required by some applications.

In most impedance control schemes, the desired impedance is normally selected without rigorous justification. In fact, the desired impedance for a given constraint environment is difficult to be quantified, let alone for an uncertain constraint environment [25]. As the desired impedance describes a property of interactions between the manipulators and the environments (inertia, damping and stiffness), it should also reflect the uncertainty of the system. For example, the desired impedance for grinding a soft workpiece should be different from that for grinding a harder workpiece by the same robotic arm. The desired impedance may even be different for different areas on the same workpiece.

In this chapter, adaptive and robust adaptive impedance control schemes with position tracking capabilities are developed. Being different from most traditional impedance control schemes, parameters of the desired impedance are varying and are tuned adaptively. The proposed controller guarantees the asymptotic convergence of the position tracking errors and the boundedness of the constraint force tracking error.

The chapter is organized as follows. In Section 3.1, the dynamic model and impedance model of a constrained robot is given. In Section 3.2, an adaptive impedance control scheme is presented which guarantees the asymptotical stability of the robot position and the boundedness of the constraint force error. In Section 3.3, the adaptive impedance control scheme in Section 3.2 is robustified to counter the dynamic modeling errors and the external disturbances. In Section 3.4, a model free neural network based adaptive impedance control approach is presented. In Section 3.5, simulation studies are done to show the effectiveness of the proposed controllers. The conclusion of the work in this chapter is given in Section 3.6.

3.1 Dynamic and Impedance Models

The dynamic model of a constrained manipulator in joint space is described by

$$M(q)\ddot{q} + C(q,\ \dot{q})\dot{q} + G(q) = \tau + J^T(q)f \tag{3.1}$$

where $q \in R^n$ are the joint displacements, $\dot{q} \in R^n$ are the joint velocities, $M(q) \in R^{n\times n}$ is the inertia matrix, $C(q, \dot{q}) \in R^{n\times n}$ is the coriolis and centrifugal force matrix, $G(q) \in R^n$ is the gravitational force, $\tau \in R^n$ are the joint torques, $J(q) \in R^{m\times n}$ is the Jacobian matrix and $f \in R^m$ is the constraint force, n is the degree of freedom of the robot and m is the dimension of the work space.

The dynamics model in task space is described by

$$M_r(q)\ddot{r} + C_r(q,\ \dot{q})\dot{r} + G_r(q) = J^{-T}(q)\tau + f \tag{3.2}$$

where

$$M_r(q) = J^{-T}(q)M(q)J^{-1}(q)$$

$$
\begin{aligned}
C_r(q,\ \dot{q}) &= J^{-T}(q)(M(q)\dot{J}^{-1}(q)+C(q,\dot{q})J^{-1}(q)) \\
G_r(q) &= J^{-T}(q)G(q)
\end{aligned}
$$

It is easy to verify that the dynamic model (3.2) has the following properties.

Property 3.1 [15] The inertial matrix M_r is symmetric positive definite matrix, provided that J is of full rank.

Property 3.2 [15] The matrix $\dot{M}_r - 2C_r$ is skew-symmetric given that the matrix $\dot{M} - 2C$ is skew symmetric.

In traditional impedance control, the desired impedance is normally modeled as [28]

$$f_d - f = M_m(\ddot{r}_d - \ddot{r}) + D_m(\dot{r}_d - \dot{r}) + K_m(r_d - r) \tag{3.3}$$

where $f_d \in R^m$ is the desired constraint force, $f \in R^m$ is the actual constraint force, $M_m \in R^{m\times m}$, $D_m \in R^{m\times m}$ and $K_m \in R^{m\times m}$ are the constant inertia matrix, damping matrix and the stiffness matrix respectively.

The environment is modeled as a general spring with the following displacement-force relation

$$f = K_e(r_e - r) \tag{3.4}$$

where $K_e \in R^{m\times m}$ is the stiffness matrix of the environment, $r_e \in R^m$ is the rest location where the contact force is null, and $r \in R^m$ is the positional vector of the contact point made by the end effector of the robot.

From equations (3.3) and (3.4), we have

$$\ddot{e} + A\dot{e} + Be = c \tag{3.5}$$

where

$$
\begin{aligned}
e &= r_d - r \\
A &= M_m^{-1}D_m \\
B &= M_m^{-1}(K_m + K_e) \\
c &= M_m^{-1}(f_d + K_e(r_d - r_e))
\end{aligned}
$$

Usually, it is assumed that $M_m \in R^{m \times m}$, $D_m \in R^{m \times m}$ and $K_m \in R^{m \times m}$, K_e and r_e are known constants. When the environment is uncertain in terms of its shape, material and position etc., it would be more appropriate to assume that these values are unknown. Accordingly, A, B and c in equation (3.5) are unknown. Considering these uncertainties, an adaptive impedance controller is developed in the next section.

3.2 Adaptive Impedance Control

The control objective is to make r convergent to their desired trajectories r_d and $f_d - f$ bounded. It is assumed that f can be measured on-line.

Consider the following controller

$$\tau = J^T(M_r u_0 + C_r \dot{r} + G_r - f) \tag{3.6}$$

where

$$u_0 = \ddot{r}_d + \hat{A}(\dot{r}_d - \dot{r}) + \hat{B}(r_d - r) - \hat{c}$$

with $\hat{A}$, $\hat{B}$ and $\hat{c}$ being the estimates of uncertain parameters A, B and c respectively.

Substituting τ into the dynamic model (3.2), we have

$$\ddot{e} + \hat{A}\dot{e} + \hat{B}e = \hat{c} \tag{3.7}$$

We will use the adaptive control approach in [5] for the controller development. Suppose that the reference model of the position tracking error e is specified by

$$\ddot{e}_m + A_m \dot{e}_m + B_m e_m = 0 \tag{3.8}$$

where e_m and $\dot{e}_m$ are the state variables and A_m and B_m are positive definite diagonal matrices. Obviously $e_m \to 0$ and $\dot{e}_m \to 0$ when $t \to \infty$.

Subtracting equation (3.8) from equation (3.7), we have

$$\ddot{\xi} + A_m \dot{\xi} + B_m \xi = (A_m - \hat{A})\dot{e} + (B_m - \hat{B})e + \hat{c} \tag{3.9}$$

where $\xi = e - e_m$.

Defining $x = [\xi^T \ \dot{\xi}^T]^T$, equation (3.9) is re-written in a state space form

$$\dot{x} = \begin{bmatrix} 0^{n\times n} & I^{n\times n} \\ -B_m & -A_m \end{bmatrix} x + \begin{bmatrix} 0^{n\times n} & 0^{n\times n} \\ B_m - \hat{B} & A_m - \hat{A} \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ \hat{c} \end{bmatrix} \tag{3.10}$$

For the convergence of error e to zero, we have the following theorem.

Theorem 3.2.1 *For the closed-loop dynamic system (3.10), $e \to 0$ and $\dot{e} \to 0$ when $t \to \infty$ if the parameters are updated by*

$$\hat{c} = \hat{c}(0) - Q_c \int_0^t \varpi(\tau) d\tau - Q_c^* \varpi \tag{3.11}$$

$$\hat{A} = \hat{A}(0) + Q_a \int_0^t \varpi(\tau) \dot{e}^T(\tau) d\tau + Q_a^* \varpi \dot{e}^T \tag{3.12}$$

$$\hat{B} = \hat{B}(0) + Q_b \int_0^t \varpi(\tau) e^T(\tau) d\tau + Q_b^* \varpi e^T \tag{3.13}$$

where Q_c, Q_c^*, Q_a, Q_a^*, Q_b, Q_b^* are all positive definite matrices, ϖ is a vector defined by

$$\varpi = P_1^T \xi + P_2^T \dot{\xi} \tag{3.14}$$

P_1 and P_2 are the sub-matrices from a symmetric positive matrix

$$P = \begin{bmatrix} P_0 & P_1 \\ P_1 & P_2 \end{bmatrix} \tag{3.15}$$

which satisfies the following Lyapunov equation

$$P \begin{bmatrix} 0^{n\times n} & I^{n\times n} \\ -B_m & -A_m \end{bmatrix} + \begin{bmatrix} 0^{n\times n} & -B_m \\ I^{n\times n} & -A_m \end{bmatrix} P = -Q \tag{3.16}$$

with Q being a positive definite matrix.

Proof:

Let $\hat{a}_i$, $\hat{b}_i$, a_{mi} and b_{mi} ($i = 1, \ 2, \ldots, \ n$) be the column vectors of matrices $\hat{A}$, $\hat{B}$, A_m and B_m respectively.

Choose the following Lyapunov function candidate

$$\begin{aligned} V &= x^T P x + (\hat{c} - c^*)^T Q_0 (\hat{c} - c^*) + \sum_{i=1}^{n} (\hat{a}_i - a_{mi} - a_i^*)^T Q_{1i} (\hat{a}_i - a_{mi} - a_i^*) \\ &+ \sum_{i=1}^{n} (\hat{b}_i - b_{mi} - b_i^*)^T Q_{2i} (\hat{b}_i - b_{mi} - b_i^*) \end{aligned} \tag{3.17}$$

where Q_0, Q_{1i} and Q_{2i} $(i = 1,\ 2,\ \ldots\ n)$ are positive definite matrices, and c^*, a_i^* and b_i^* are the vectors to be decided later.

Differentiating V with respect to time t and considering equation (3.16), we have

$$\begin{aligned} \dot{V} &= -x^T Q x + 2\varpi^T [(B_m - \hat{B})e + (A_m - \hat{A})\dot{e}] + 2\sum_{i=1}^{n} (\hat{a}_i - a_{mi} - a_i^*)^T Q_{1i} (\dot{\hat{a}}_i - \dot{a}_i^*) \\ &+ 2\sum_{i=1}^{n} (\hat{b}_i - b_{mi} - b_i^*)^T Q_{2i} (\dot{\hat{b}}_i - \dot{b}_i^*) + 2(\hat{c} - c^*)^T Q_0 (\dot{\hat{c}} - \dot{c}^*) + 2\varpi^T \hat{c} \end{aligned} \tag{3.18}$$

where the fact that $\dot{a}_{mi} = \dot{b}_{mi} = 0$ has been used.

Letting

$$\dot{\hat{c}} - \dot{c}^* = -Q_0^{-1} \varpi \tag{3.19}$$

$$\dot{\hat{a}}_i - \dot{a}_i^* = Q_{1i}^{-1} \varpi \dot{e}_i \tag{3.20}$$

$$\dot{\hat{b}}_i - \dot{b}_i^* = Q_{2i}^{-1} \varpi e_i \tag{3.21}$$

and substituting them into equation (3.18), we obtain

$$\dot{V} = -x^T Q x + 2c^{*T} \varpi - 2\sum_{i=1}^{n} a_i^{*T} \varpi \dot{e}_i - 2\sum_{i=1}^{n} b_i^{*T} \varpi e_i \tag{3.22}$$

Letting

$$c^* = -Q_c^* \varpi \tag{3.23}$$

$$a_i^* = Q_{ai}^* \varpi \dot{e}_i \tag{3.24}$$

$$b_i^* = Q_{bi}^* \varpi e_i \tag{3.25}$$

where Q_c^*, Q_{ai}^* and Q_{bi}^* are all positive definite, and substituting them in equation (3.22), we have

$$\begin{aligned} \dot{V} &= -x^T Q x - 2\varpi^T Q_c^* \varpi - 2\sum_{i=1}^{n} (\varpi \dot{e}_i)^T Q_{ai}^* \varpi \dot{e}_i - 2\sum_{i=1}^{n} (\varpi e_i)^T Q_{bi}^* \varpi e_i \\ &\leq -x^T Q x < 0 \end{aligned} \tag{3.26}$$

From equation (3.26), it can be concluded that the system (3.9) is stable and $x \to 0$ $(e \to e_m \; and \; \dot{e} \to \dot{e}_m)$ when $t \to \infty$. Note that $e_m \to 0$ and $\dot{e}_m \to 0$ when $t \to \infty$, and accordingly we have $e \to 0$ and $\dot{e} \to 0$ when $t \to \infty$.

For simplicity in expression without losing generality, let $Q_{1i} = Q_1$, $Q_{2i} = Q_2$, $Q^*_{ai} = Q_a$ and $Q_{bi} = Q_b$ for $i = 1,\ 2, \ldots n$.

From equations (3.19) to (3.25), we have the following adaptation laws

$$\hat{c}{=}\hat{c}(0){-}Q_c\int_0^t \varpi(\tau)d\tau{-}Q^*_c\varpi \tag{3.27}$$

$$\hat{a}_i{=}\hat{a}_i(0){+}Q_a\int_0^t \varpi(\tau)\dot{e}_i(\tau)d\tau{+}Q^*_a\varpi\dot{e}_i \tag{3.28}$$

$$\hat{b}_i{=}\hat{b}_i(0){+}Q_b\int_0^t \varpi(\tau)e_i(\tau)d\tau{+}Q^*_b\varpi e_i \tag{3.29}$$

where $Q_c = Q_0^{-1}$, $Q_a = Q_1^{-1}$ and $Q_b = Q_2^{-1}$.

Re-writing equations (3.28) to (3.29) in matrix forms, we have

$$\hat{A} = \hat{A}(0) + Q_a\int_0^t \varpi(\tau)\dot{e}^T(\tau)d\tau + Q^*_a\varpi\dot{e}^T \tag{3.30}$$

$$\hat{B} = \hat{B}(0) + Q_b\int_0^t \varpi(\tau)e^T(\tau)d\tau + Q^*_b\varpi e^T \tag{3.31}$$

Q. E. D

Following the **Theorem 3.2.1**, the complete adaptive impedance controller is then given as

$$\tau{=}\ J^T[M_r(\ddot{r}_d + \hat{A}\dot{e} + \hat{B}e - \hat{c}) + C_r\dot{r} + G_r - f] \tag{3.32}$$

From equations (3.4) and (3.5), we have

$$f_d - f = M_m c - K_e e \tag{3.33}$$

As c is determined by f_d, r_e, r_d and M_m, it is bounded. The boundedness of c and $e \to 0$ leads to the boundedness of $f_d - f$.

If K_e and r_e are known exactly and f_d is planned as

$$f_d = K_e(r_e - r_d) \tag{3.34}$$

we have $c = 0$ and $(f_d - f) \to 0$ from equation (3.33).

Remark 3.2.1 *An explicit force error loop is not included in the controller, but the measurement of f is required. The controller can only make the force error to be bounded if the exact model of the constraint is unknown.*

Remark 3.2.2 *The controller presented above can achieve position tracking and the boundedness of the force tracking errors within the framework of impedance control approach. Thus it keeps the advantages of impedance control such as the abilities to accommodate both unconstrained and constrained motion and the good robustness to disturbances.*

3.3 Robust Adaptive Impedance Control

Adaptive impedance controller in Section 3.2 is designed without consideration of the dynamic modeling errors and the external disturbances in the robot system. To compensate these errors and disturbances, a robust adaptive impedance controller is designed in this section.

Considering modeling errors and disturbances, the dynamics of the constrained robot in task space (3.2) is re-written as

$$M_r(q)\ddot{r} + C_r(q,\ \dot{q})\dot{r} + G_r(q) = J^{-T}(q)\tau + f + \tilde{f} \tag{3.35}$$

where $\tilde{f} \in R^m$ is external disturbances. The definitions of other terms are the same as those in equation (3.2), but their exact values are not known. Assume that their estimates are $\hat{M}_r(q)$, $\hat{C}_r(q,\ \dot{q})$ and $\hat{G}(r)$ respectively. The modeling errors and external disturbances are assumed to be bounded such that

$$\begin{aligned} \|\Delta M_r(q)\| &= \|M_r(q) - \hat{M}_r(q)\| \leq \delta M && (3.36)\\ \|\Delta C_r(q,\ \dot{q})\| &= \|C_r(q,\ \dot{q}) - \hat{C}_r(q,\ \dot{q})\| \leq \delta C && (3.37)\\ \|\Delta G_r(q)\| &= \|G_r(q) - \hat{G}(r)\| \leq \delta G && (3.38)\\ \|\tilde{f}\| &\leq \delta f && (3.39) \end{aligned}$$

Consider the following dynamic compensator [22]

$$\dot{z} = -Dz + K_v\dot{e} + K_p e - K_c\hat{c} \tag{3.40}$$

where $e = r_d - r$, $\dot{e} = \dot{r}_d - \dot{r}$ and $D \in R^{m\times m}$ is any negative definite matrix. The matrices $K_p \in R^{m\times m}$, $K_v \in R^{m\times m}$ and $K_c \in R^{m\times m}$ are to be determined later.

The output z of compensator (3.40) and the tracking errors e and $\dot{e}$ are used to form the following switching function

$$s(e, \dot{e}, z) = \dot{e} + K_1 e + K_2 z \tag{3.41}$$

where $K_1 \in R^{m\times m}$ and $K_2 \in R^{m\times m}$ are constant positive definite matrices.

From equations (3.40) and (3.41), we have

$$\ddot{e} + (K_1 + K_2 K_v - K_2 D K_2^{-1})\dot{e} + K_2(K_p - D K_2^{-1} K_1)e = K_2(K_c \hat{c} - D K_2^{-1} s) + \dot{s} \tag{3.42}$$

If the design parameters in equation (3.40) are chosen such that

$$\begin{aligned} K_v &= K_2^{-1}(\hat{A} + K_2 D K_2^{-1} - K_1) && (3.43)\\ K_p &= K_2^{-1}\hat{B} + D K_2^{-1} K_1 && (3.44)\\ K_c &= K_2^{-1} && (3.45) \end{aligned}$$

equation (3.42) becomes

$$\ddot{e} + \hat{A}\dot{e} + \hat{B}e = \hat{c} + \dot{s} - K_2 D K_2^{-1} s \tag{3.46}$$

Obviously if the state of the controlled system is kept in the sliding surface $s = 0$ asymptotically , the desired impedance (3.7) is achieved.

The position tracking and the boundedness of the force errors is another objective of the controller. We still use the following reference model 3.8 which is reproduced below for the completeness of the presentation.

$$\ddot{e}_m + A_m \dot{e}_m + B_m e_m = 0 \tag{3.47}$$

The robust adaptive impedance control law for achieving the above objectives is presented in the following theorem.

Theorem 3.3.1 *For the constrained robot system (3.35), the desired impedance (3.7), position tracking and the boundedness of the force tracking errors can be achieved through the following controller*

$$\tau = J^T[\hat{M}_r \ddot{r}_{eq} + K_s s + d\, sgn(s) + \hat{C}_r \dot{r}_{eq} + \hat{G}_r - f] \tag{3.48}$$

with an adaptively tuned impedance

$$\begin{aligned}\hat{c} &= \hat{c}(0) - Q_c\int_0^t \varpi(\tau)d\tau - Q_c^*\varpi \\ \hat{A} &= \hat{A}(0) + Q_a\int_0^t \varpi(\tau)\dot{e}^T(\tau)d\tau + Q_a^*\varpi\dot{e}^T \\ \hat{B} &= \hat{B}(0) + Q_b\int_0^t \varpi(\tau)e^T(\tau)d\tau + Q_b^*\varpi e^T\end{aligned}$$

where

$$\dot{r}_{eq} = \dot{r}_d + K_1 e + K_2 z \tag{3.49}$$

$$\ddot{r}_{eq} = \ddot{r}_d + K_1\dot{e} + K_2\dot{z} \tag{3.50}$$

$$d \geq \delta M\|\ddot{r}_{eq}\| + \delta C\|\dot{r}_{eq}\| + \delta G + \delta f \tag{3.51}$$

$K_s \in R^{m\times m}$ is a constant positive definite matrix, $sgn(s)$ is a sign function which is -1 if $s < 0$ and 1 if $s \geq 0$, Q_c, Q_c^*, Q_a, Q_a^*, Q_b and Q_b^* are all positive definite matrices and ϖ is a vector defined by

$$\varpi = P_1^T\xi + P_2^T\dot{\xi} \tag{3.52}$$

with P_1 and P_2 being the sub-matrices from a symmetric positive matrix

$$P = \begin{bmatrix} P_0 & P_1 \\ P_1 & P_2 \end{bmatrix} \tag{3.53}$$

which satisfies the following Lyapunov equation

$$P\begin{bmatrix} 0^{n\times n} & I^{n\times n} \\ -B_m & -A_m \end{bmatrix} + \begin{bmatrix} 0^{n\times n} & -B_m \\ I^{n\times n} & -A_m \end{bmatrix}P = -Q \tag{3.54}$$

where Q is a positive definite matrix.

Proof:

To prove the reaching of sliding mode defined in equation (3.41), choose a Lyapunov function which is a positive definite function of s such that

$$V_1(s) = \frac{1}{2}s^T M_r s \tag{3.55}$$

Note that M_r is positive definite as stated in **Property 3.1**.

Differentiating V_1 with respect to time t and considering **Property 3.2**, we have

$$\dot{V}_1 = s^T M_r \dot{s} + s^T C_r s \tag{3.56}$$

Considering $s = \dot{r}_{eq} - \dot{r}$ and $\dot{s} = \ddot{r}_{eq} - \ddot{r}$, equation (3.56) is re-written as

$$\dot{V}_1 = s^T (M_r \ddot{r}_{eq} + C_r \dot{r}_{eq} - M_r \ddot{r} - C_r \dot{r}) \tag{3.57}$$

Substituting control τ in equation (3.48) into system dynamics (3.2), it follows that

$$M_r \ddot{r} + C_r \dot{r} + G_r = \hat{M}_r \ddot{r}_{eq} + K_s s + d\, sgn(s) + \hat{C}_r \dot{r}_{eq} + \hat{G}_r + \tilde{f} \tag{3.58}$$

From equations (3.57) and (3.58), we have

$$\dot{V}_1 = s^T (\delta M_r \ddot{r}_{eq} + \delta C_r \dot{r}_{eq} + \delta G_r - \tilde{f} - K_s s - d\, sgn(s)) \tag{3.59}$$

From equations (3.37) –(3.39) and (3.51), we have

$$\begin{aligned} \dot{V}_1 &\leq s^T (\delta M \|\ddot{r}_{eq}\| + \delta C \|\dot{r}_{eq}\| + \delta G + \delta f - \tilde{f} - K_s s - d\, sgn(s)) \\ &\leq -s^T K_s s \leq 0 \end{aligned} \tag{3.60}$$

From the definition of V_1 (3.55) and equation (3.60), it can be concluded that when $t \rightarrow \infty$, $V_1 \rightarrow 0$ and $s \rightarrow 0$. After sliding surface $s = 0$ is reached asymptotically, $\dot{s} = 0$. From equation (3.46), the desired impedance is achieved.

Following the same procedure as that in **Theorem 3.2.1**, we can also prove that $e \rightarrow 0$ and $\dot{e} \rightarrow 0$ when $t \rightarrow \infty$ and $f_d - f$ is bounded when the following adaptive laws are applied

$$\hat{c} = \hat{c}(0) - Q_c \int_0^t \varpi(\tau) d\tau - Q_c^* \varpi \tag{3.61}$$

$$\hat{a}_i = \hat{a}_i(0) + Q_a \int_0^t \varpi(\tau) \dot{e}_i(\tau) d\tau + Q_a^* \varpi \dot{e}_i \tag{3.62}$$

$$\hat{b}_i = \hat{b}_i(0) + Q_b \int_0^t \varpi(\tau) e_i(\tau) d\tau + Q_b^* \varpi e_i \tag{3.63}$$

where $Q_c = Q_0^{-1}$, $Q_a = Q_1^{-1}$ and $Q_b = Q_2^{-1}$,or in matrix forms

$$\hat{A} = \hat{A}(0) + Q_a \int_0^t \varpi(\tau) \dot{e}^T(\tau) d\tau + Q_a^* \varpi \dot{e}^T \tag{3.64}$$

$$\hat{B} = \hat{B}(0) + Q_b \int_0^t \varpi(\tau) e^T(\tau) d\tau + Q_b^* \varpi e^T \tag{3.65}$$

Q. E. D

Remark 3.3.1 *It is well known that the sign function in the controller cause chattering which can be eliminated with the boundary layer approach [58]. In this approach, the sign function $sgn(s)$ is replaced by s/Δ when $s < \Delta$. Δ is defined as a boundary layer.*

Remark 3.3.2 *The dynamic modeling errors in the robot system can also be compensated by other methods such as neural network control described in Chapter 2. In this approach, M_r, C_r and G_r are approximated by adaptive tuned neural networks and this is to be covered in the next section.*

3.4 Robust NN Adaptive Impedance Control

In practice, the exact model of the robot dynamics is not known. To approximate it, Gaussian radial-basis function (RBF) neural networks can be utilized as they can approximate any smooth functions [66]. The effectiveness of this approximation approach has been shown in the control of a robot constrained by a moving constraint in Chapter 2.

Denote the elements of M_r, C_r and G_r as m_{kj}, c_{kj} and g_k ($k = 1,\ 2\ldots m$, $j = 1,\ 2\ldots m$) respectively. Their neural network approximations are represented by

$$m_{kj}(q) = \theta_{Mkj}^T \xi_{Mkj}(q) + \epsilon_{Mkj}(q) \tag{3.66}$$

$$c_{kj}(q,\ \dot{q}) = \theta_{Ckj}^T \xi_{Ckj}(z_1) + \epsilon_{Ckj}(z_1) \tag{3.67}$$

$$g_k(q) = \theta_{Gk}^T \xi_{Gk}(q) + \epsilon_{Gk}(q) \tag{3.68}$$

where $z_1 = (q,\ \dot{q})^T$, $\epsilon_{Mkj}(q)$, $\epsilon_{Ckj}(z_1)$ and $\epsilon_{Gk}(q)$ are the approximation errors, θ_{Mkj}, θ_{Ckj} and θ_{Gkj} are the column vector of the neural network weights, $\xi_{Mkj}(q)$, $\xi_{Ckj}(z_1)$, and $\xi_{Gk}(q)$ are the vectors of the RBF Gaussian functions (activation functions) of the form defined as

$$a(x) = exp(\frac{-(x-\mu_i)^T(x-\mu_i)}{\sigma^2}) \tag{3.69}$$

with μ_i being the centers of the functions, and $\sigma^2 \in R$ being the variance.

By using the notations of GL matrix [24] as done in Chapter 2, equations (3.66) and equations (3.68) can be written in a compact form such that

$$\begin{aligned} M_r(q) &= [\{\Theta_M\}^T \bullet \{\Xi_M(q)\}] + E_M(q) && (3.70)\\ C_r(q,\ \dot{q}) &= [\{\Theta_C\}^T \bullet \{\Xi_C(z_1)\}] + E_C(z_1) && (3.71)\\ G_r(q) &= [\{\Theta_G\}^T \bullet \{\Xi_G(q)\}] + E_G(q) && (3.72) \end{aligned}$$

where $\{\Theta_M\}$, $\{\Theta_C\}$ and $\{\Theta_G\}$ are GL matrices formed by θ_{Mkj}, θ_{Ckj} and θ_{Gk} respectively, $\{\Xi_M(q)\}$, $\{\Xi_C(z_1)\}$ and $\{\Xi_G(q)\}$ are GL matrices formed by ξ_{Mkj}, ξ_{Ckj} and ξ_{Gk} respectively. $E_M(q)$, $E_C(z_1)$ and $E_G(q)$ are the matrices of neural network approximation errors with $\epsilon_{Mkj}(q)$, $\epsilon_{Ckj}(z_1)$ and $\epsilon_{Gk}(q)$ being their elements.

Let the estimates of $M_r(q)$, $C_r(q,\ \dot{q})$ and $G_r(q)$ be $\hat{M}_r(q)$, $\hat{C}_r(q,\ \dot{q})$ and $\hat{G}_r(q)$ respectively. They are written as

$$\begin{aligned} \hat{M}_r(q) &= [\{\hat{\Theta}_M\}^T \bullet \{\Xi_M(q)\}] && (3.73)\\ \hat{C}_r(q,\ \dot{q}) &= [\{\hat{\Theta}_C\}^T \bullet \{\Xi_C(z_1)\}] && (3.74)\\ \hat{G}_r(q) &= [\{\hat{\Theta}_G\}^T \bullet \{\Xi_G(q)\}] && (3.75) \end{aligned}$$

which will be used for the development of the neural network based impedance control.

Consider the following controller

$$u = \hat{M}_r \ddot{r}_{eq} + K_s s + d\, sgn(s) + \hat{C}_r \dot{r}_{eq} + \hat{G}_r - f \qquad (3.76)$$

where s, $sgn(s)$, $\dot{r}_{eq}$, $\ddot{r}_{eq}$ are defined in **Theorem 3.3.1**, K_s is a positive definite matrix and d is a scalar constant to be determined later.

Substituting equation (3.76) into system dynamic equation (3.2) and considering equations (3.70) –(3.75), we have the following error dynamics

$$\begin{aligned} & M_r \dot{s} + C_r s - (E_M \ddot{r}_{eq} + E_C \dot{r}_{eq} + E_G) + K_s s + d\, sgn(s) \\ & = [\{\tilde{\theta}_M \bullet \{\Xi_M\}] \ddot{r}_{eq} + [\{\tilde{\theta}_C \bullet \{\Xi_C\}] \dot{r}_{eq} + [\{\tilde{\theta}_G \bullet \{\Xi_G\}] \qquad (3.77) \end{aligned}$$

where $\tilde{\theta}_M = \theta_M - \hat{\theta}_M$, $\tilde{\theta}_C = \theta_C - \hat{\theta}_C$ and $\tilde{\theta}_G = \theta_G - \hat{\theta}_G$.

From the error dynamics in equation (3.77), we have the following theorem about the performance of the closed-loop system under the control law (3.76).

Theorem 3.4.1 *Under the neural network control law (3.76), the desired impedance (3.7), the position tracking and the boundedness of the force errors can be achieved if*

$$d \geq \|E_M(q)\ddot{r}_{eq} + E_C(z_1)\dot{r}_{eq} + E_G(q)\| \tag{3.78}$$

and the weights of the neural network and the parameters of the desired impedance are tuned adaptively by

$$\dot{\hat{\theta}}_{Mkj} = \Gamma_{Mkj}\xi_{Mkj}\ddot{r}_{eqj}s_k \tag{3.79}$$

$$\dot{\hat{\theta}}_{Ckj} = \Gamma_{Ckj}\xi_{Ckj}\dot{r}_{eqj}s_k \tag{3.80}$$

$$\dot{\hat{\theta}}_{Gk} = \Gamma_{Gk}\xi_{Gk}s_k \tag{3.81}$$

$$\hat{c} = \hat{c}(0) - Q_c\int_0^t \varpi(\tau)d\tau - Q_c^*\varpi \tag{3.82}$$

$$\hat{A} = \hat{A}(0) + Q_a\int_0^t \varpi(\tau)\dot{e}^T(\tau)d\tau + Q_a^*\varpi\dot{e}^T \tag{3.83}$$

$$\hat{B} = \hat{B}(0) + Q_b\int_0^t \varpi(\tau)e^T(\tau)d\tau + Q_b^*\varpi e^T \tag{3.84}$$

where Γ_{Mkj}, Γ_{Ckj} Γ_{Gk}, Q_c, Q_c^*, Q_a, Q_a^*, Q_b and Q_b^* are constant symmetric positive definite matrices, s_k is the kth element of s and r_{eqj} is the jth element of r_{eq}, and ϖ is a vector defined by

$$\varpi = P_1^T\xi + P_2^T\dot{\xi} \tag{3.85}$$

with P_1 and P_2 being the sub-matrices from a symmetric positive matrix

$$P = \begin{bmatrix} P_0 & P_1 \\ P_1 & P_2 \end{bmatrix} \tag{3.86}$$

which satisfies the following Lyapunov equation

$$P\begin{bmatrix} 0^{n\times n} & I^{n\times n} \\ -B_m & -A_m \end{bmatrix} + \begin{bmatrix} 0^{n\times n} & -B_m \\ I^{n\times n} & -A_m \end{bmatrix}P = -Q < 0 \tag{3.87}$$

Proof:

Choose the following Lyapunov function candidate

$$V_1 = \frac{1}{2}s^TM_rs + \frac{1}{2}\sum_{k=1}^{m}\sum_{j=1}^{m}(\tilde{\theta}_{Mkj}\Gamma_{Mkj}^{-1}\tilde{\theta}_{Mkj} + \tilde{\theta}_{Ckj}\Gamma_{Ckj}^{-1}\tilde{\theta}_{Ckj}) + \frac{1}{2}\sum_{k=1}^{m}\tilde{\theta}_{Gk}\Gamma_{Gk}^{-1}\tilde{\theta}_{Gk} \tag{3.88}$$

where M_r is positive definite by **Property 3.1**, $\tilde{\theta}_{Mkj} = \theta_{Mkj} - \hat{\theta}_{Mkj}$, $\tilde{\theta}_{Ckj} = \theta_{Ckj} - \hat{\theta}_{Ckj}$ and $\tilde{\theta}_{Gk} = \theta_{Gk} - \hat{\theta}_{Gk}$.

Differentiating V_1 with respect to time t and considering **Property 3.2**, we have

$$\dot{V}_1 = s^T M_r \dot{s} + s^T C_r s + \sum_{k=1}^{m}\sum_{j=1}^{m}(\tilde{\theta}_{Mkj}\Gamma^{-1}_{Mkj}\dot{\tilde{\theta}}_{Mkj} + \tilde{\theta}_{Ckj}\Gamma^{-1}_{Ckj}\dot{\tilde{\theta}}_{Ckj}) + \sum_{k=1}^{m}\tilde{\theta}_{Gk}\Gamma^{-1}_{Gk}\dot{\tilde{\theta}}_{Gk} \tag{3.89}$$

From equations (3.77), (3.89), (3.79) – (3.81) and noting that

$$\begin{aligned} s^T\hat{M}_r\ddot{r}_{eq} &= \sum_{k=1}^{m}\sum_{j=1}^{m}\tilde{\theta}^T_{Mkj}\xi_{Mkj}\ddot{r}_{eqj}s_k \\ s^T\hat{C}_r\dot{r}_{eq} &= \sum_{k=1}^{m}\sum_{j=1}^{m}\tilde{\theta}^T_{Ckj}\xi_{Ckj}\dot{r}_{eqj}s_k \\ s^T\hat{G}_r &= \sum_{k=1}^{m}\tilde{\theta}^T_{Gk}\xi_{Gk}s_k \end{aligned}$$

we have

$$\dot{V}_1 = -s^T K_s s - s^T d\, sgn(s) + s^T(E_M\ddot{r}_{eq} + E_C\dot{r}_{eq} + E_G) \tag{3.90}$$

The fact that $\dot{\tilde{\theta}}_{Mkj} = -\dot{\hat{\theta}}_{Mkj}$, $\dot{\tilde{\theta}}_{Ckj} = -\dot{\hat{\theta}}_{Ckj}$ and $\dot{\tilde{\theta}}_{Gk} = -\dot{\hat{\theta}}_{Gk}$ is used in the above derivations.

The selection of K_s and d make

$$\dot{V}_1 \leq -s^T K_s s \leq 0 \tag{3.91}$$

From the definition of V_1 (3.88), it can be concluded that when $t \to \infty$, $V_1 \to 0$ and $s \to 0$. After sliding surface $s = 0$ is reached asymptotically, $\dot{s} = 0$. From equation (3.46), the desired impedance is achieved. We can also conclude that $\hat{\theta}_{Mkj}$, $\hat{\theta}_{Ckj}$ and $\hat{\theta}_{Gk}$ are bounded.

As proved in **Theorem 3.3.1**, the robot position tracking and the boundedness of the force tracking errors are achieved after the desired impedance is reached. **Q. E. D**

Remark 3.4.1 *The centers of the RBF functions in the neural network can be selected in such a way that they are evenly distributed in the input space of q and $\dot{q}$.*

3.5 Simulation

The simulation example is schematically shown in Figure 3.1. In the example, the end effector of the manipulator moves along a part of the constraint surface and exerts a force on it at the same time.

The length, inertia and the mass of each link of the manipulator are $l_i = 0.3m$, $I_i = 0.3kgm^2$ and $m_i = 0.1kg$ respectively $(i = 1, 2)$. The mass center of each link is assumed to be in the middle of the link. These parameters are the same as those used in the simulation example in Chapter 2.

In Figure 3.1, oxy is the world coordinates. The constraint surface is described by

$$\Phi(r_e) = x_e - y_e + 0.25 = 0 \tag{3.92}$$

and the planned trajectory of the end effector is

$$\Phi(r_d) = x_d - y_d + 0.35 = 0 \tag{3.93}$$

Their trajectories in the time domain are represented by

$$\begin{aligned} x_d(t) &= -\frac{1}{10}\cos(2t) \\ y_d(t) &= 0.35 - \frac{1}{10}\cos(2t) \\ x_e(t) &= 0.05 - \frac{1}{10}\cos(2t) \\ y_e(t) &= 0.3 - \frac{1}{10}\cos(2t) \end{aligned}$$

Assume that the rest position of the constraint surface is the same as the constraint surface (3.92). The planned force is set as $f_d = [f_{xd}\ f_{yd}]^T = [5\ \ -5]^T$. For the simulation, the actual values of K_e is set as $150I^{2\times 2}$ and r_e is still described by equation (3.92).

3.5.1 Simulation for Adaptive Impedance Control

In this case, the kinematic and the inertia parameters of the robot are known. The control parameters are chosen as follows: $A_m = 20I^{2\times 2}$, $B_m = 400I^{2\times 2}$, $P_1 =$

$1.25I^{2\times 2}$, $P_2 = 6.56I^{2\times 2}$, $\hat{A}(0) = 45I^{2\times 2}$, $\hat{B}(0) = 30I^{2\times 2}$, $\hat{c}(0) = [1\ 1]^T$, $Q_a = Q_b = Q_c = Q_{a*} = 5I^{2\times 2}$ and $Q_{b*} = Q_{c*} = 1.5I^{2\times 2}$.

The robot position tracking and the constraint force tracking performances are plotted in Figures 3.2 and 3.3 respectively. The control torques for the manipulators are given in Figure 3.4. It can be observed that under the proposed controller and the adaptation law, the positions converge to their desired values and the force errors are bounded. The control torques are in the reasonable ranges.

3.5.2 Simulation for Robust Adaptive Impedance control

In this case, the inertial parameters of the robot are unknown. Assume that the estimated inertia and the mass of each link of the manipulator are $\hat{I}_i = 0.2kgm^2$ and $\hat{m}_i = 0.05kg$ respectively ($i = 1, 2$). The modeling uncertainty bounds for the dynamic terms are $\delta M = 0.1$, $\delta C = 0.1$, $\delta G = 0.05$ respectively and the bound of external disturbance is $\delta f = 0.1$. The control parameters are chosen as follows: $D = -2I^{2\times 2}$,$K_1 = K_2 = 2I^{2\times 2}$ and $K_s = 4I^{2\times 2}$. The boundary layer is chosen as $\Delta = 0.05$. Other parameters: A_m, $B_m =$, P_1, P_2, $\hat{A}(0) =$, $\hat{B}(0)$, $\hat{c}(0)$, Q_a, Q_b, Q_c, Q_{a*}, Q_{b*} and Q_{c*} are the same as in Section 3.5.1.

The position tracking performances of the robot and the force tracking performances are plotted in Figures 3.5 and 3.6 respectively. The control torques for the manipulators are given in Figure 3.7. It can be seen that under the proposed controller and the adaptation law, the positions converge to their desired values and the force errors are bounded. The control torques are in the reasonable ranges. It is noted that there are some chattering in torque and force signals, but they are smoothed out in most of the time through boundary layer approach.

3.5.3 Simulation for Robust NN Adaptive Impedance control

The control parameters are chosen as follows: $D = -5I^{2\times 2}$, $K_1 = K_2 = 5I^{2\times 2}$, $K_s = 36I^{2\times 2}$. The boundary layer is chosen as $\Delta = 0.05$. $\Gamma_{Mkj} = diag[8.0]$, $\Gamma_{Ckj} = diag[6.0]$ and $\Gamma_{Gk} = diag[10.0]$. The centers of RBF functions span evenly in the input space of q and $\dot{q}$ and their variances are set to be 50. $A_m = 20I^{2\times2}$, $B_m = 40I^{2\times2}$, $P_1 = 0.625I^{2\times2}$, $P_2 = 6.5I^{2\times2}$, $\hat{A}(0) = I^{2\times2}$, $\hat{B}(0) = 110I^{2\times2}$, $\hat{c}(0) = [-4\ 4]^T$, $Q_a = Q_b = Q_c = Q_a^* = 4I^{2\times2}$ and $Q_b^* = Q_c^* = 8I^{2\times2}$.

The position of the robot and the constraint force are plotted in Figures 3.8 and 3.9 respectively. To show the impedance tracking performance, the response of the switching function (s) is plotted in Figure 3.10. The control torques for the manipulators are given in Figure 3.11. The neural network approximations to M_r, C_r and G_r in terms of their norms (largest singular values) are plotted from Figure 3.12 to Figure 3.14. It can be seen that under the proposed controller, the position of the robot converged to its desired trajectory and tracking errors of the constraint force are bounded. The responses of the switching function and its derivatives are stabilized around zero. The control torques are in the reasonable ranges and parameter estimation are also bounded. Note that there are some chatterings in the force response and the torque at the beginning of the simulation, but they are almost smoothed out quickly due to the introduction of the boundary layer.

3.6 Conclusion

In this chapter, an adaptive, a robust adaptive and a neural network based robust adaptive control schemes are developed for position/force tracking of a constrained robot within the framework of impedance control. Compared with other impedance controllers, the uniqueness of the controllers developed is that the desired impedance is treated as time varying and is adapted with the robot position tracking errors. The uncertainties of both the impedance and the robot dynamics are considered in the controllers design, and this makes the controllers developed more general than other control schemes. Under the proposed controllers, the position of the robot converges to its desired trajectory and the constraint force error is bounded. Extensive simulations are done to verify the effectiveness of the control schemes.

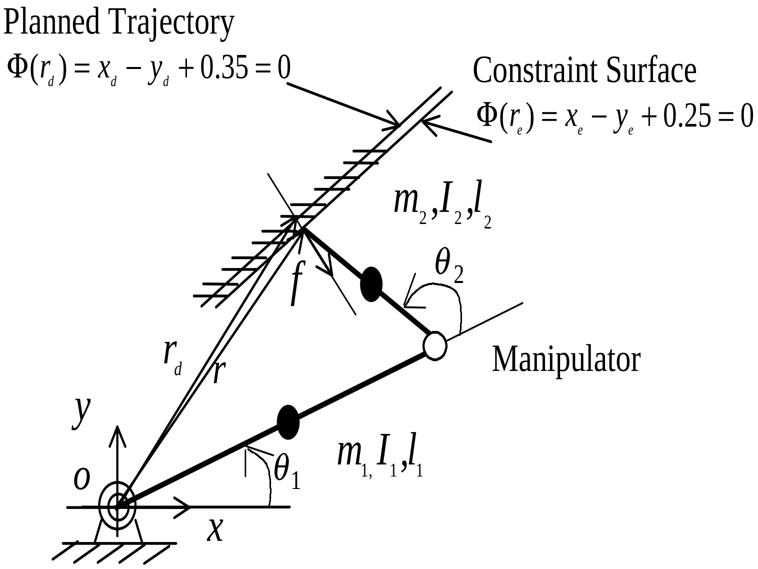

Figure 3.1: Simulation Example

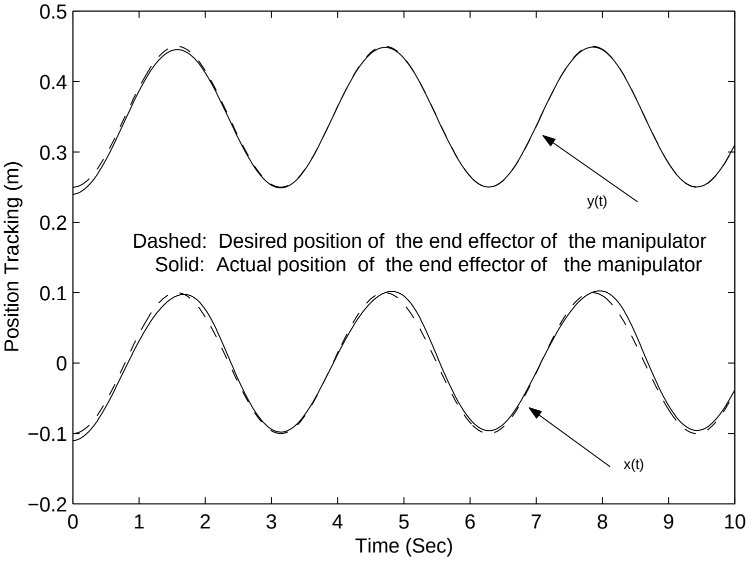

Figure 3.2: Position Tracking under Adaptive Impedance Control

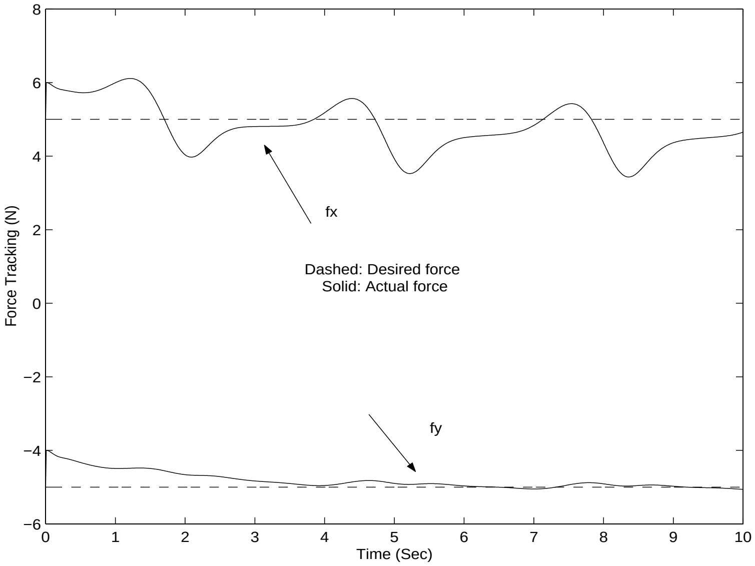

Figure 3.3: Force Tracking under Adaptive Impedance Control

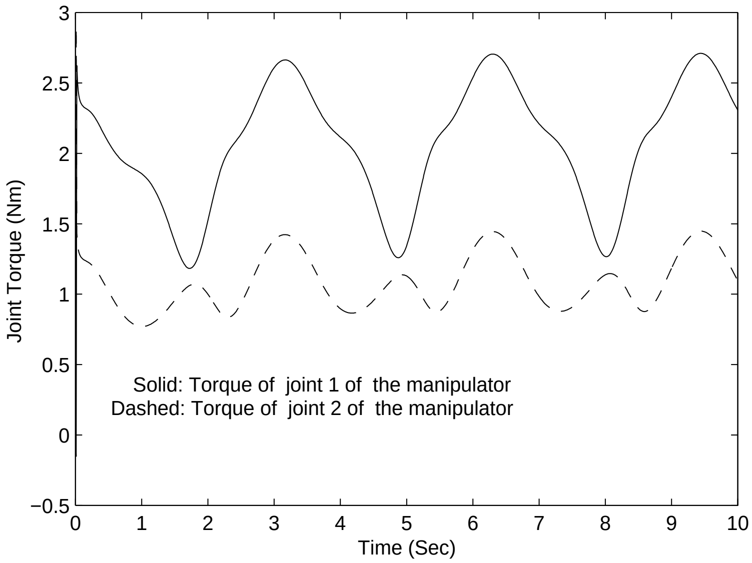

Figure 3.4: Joint Torques of the Robotic Arm under Adaptive Impedance Control

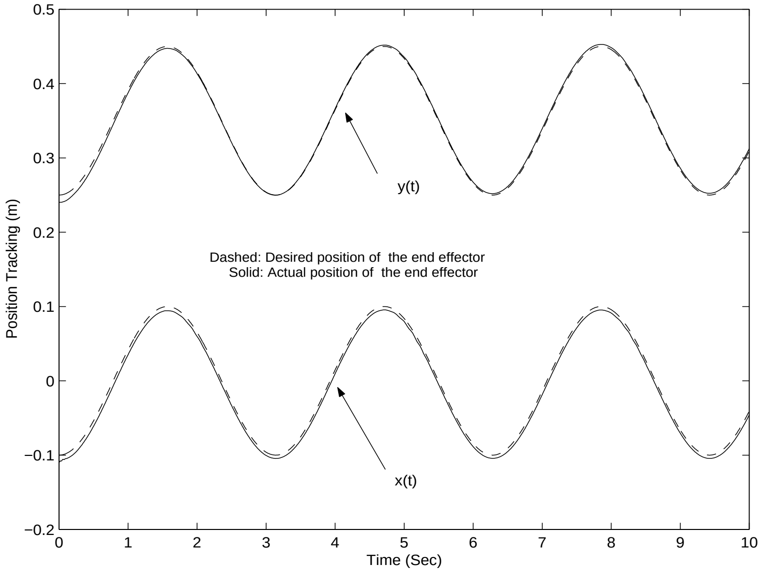

Figure 3.5: Position Tracking under Robust Adaptive Impedance Control

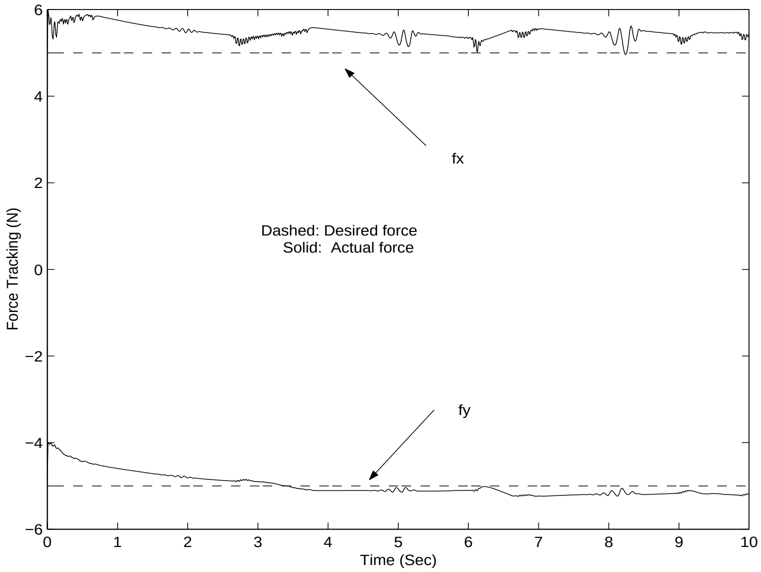

Figure 3.6: Force Tracking under Robust Adaptive Impedance Control

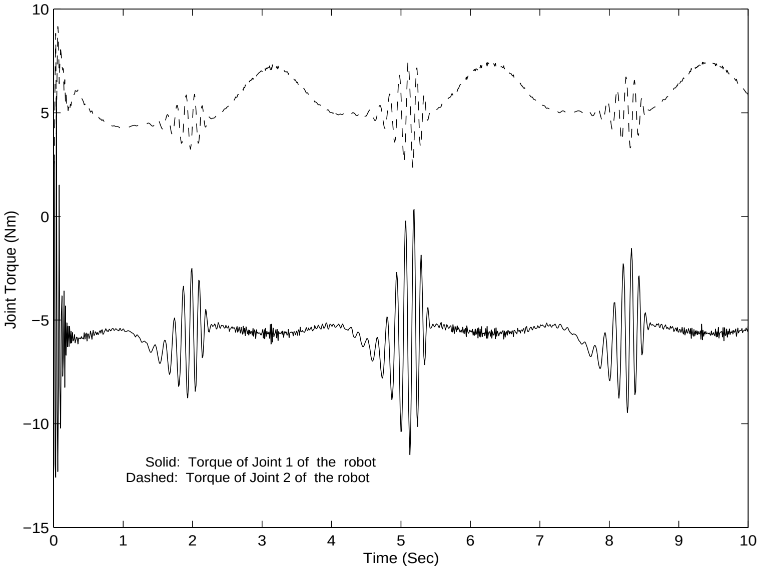

Figure 3.7: Joint Torques of the Robotic Manipulator under Robust Adaptive Impedance Control

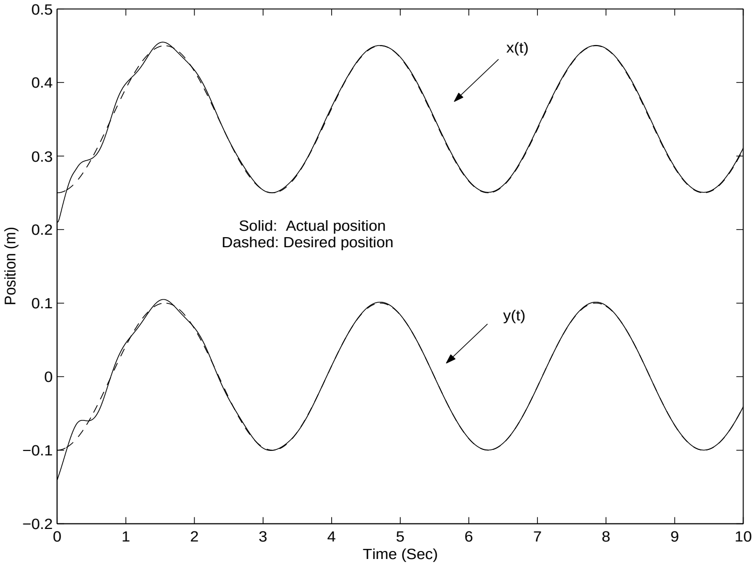

Figure 3.8: Position Tracking under Neural Network based Controller

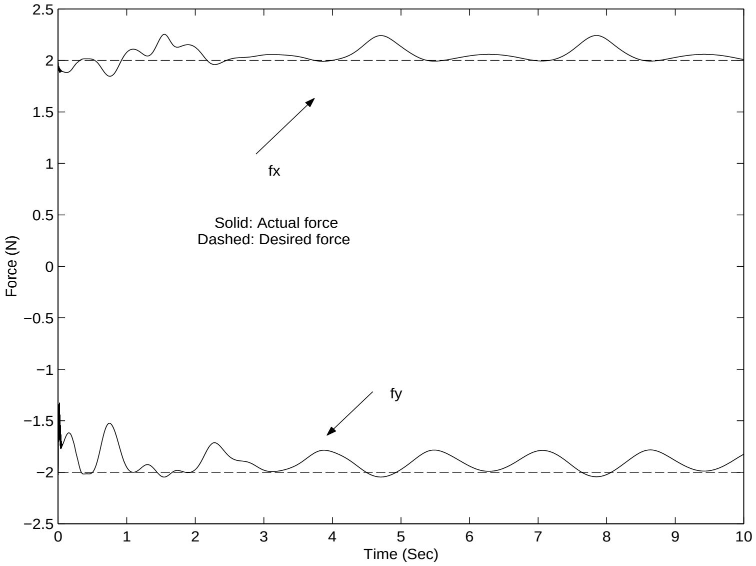

Figure 3.9: Force Tracking under Neural Network based Controller

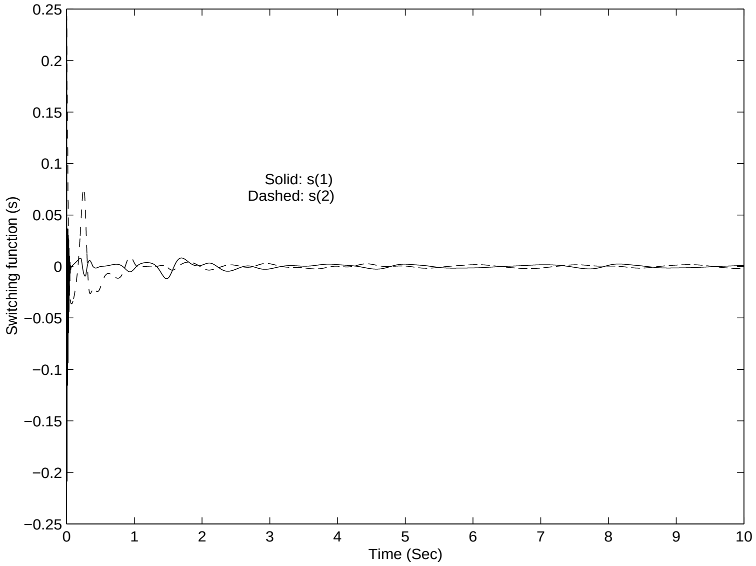

Figure 3.10: Response of the switching functions (s) under Neural Network based Controller

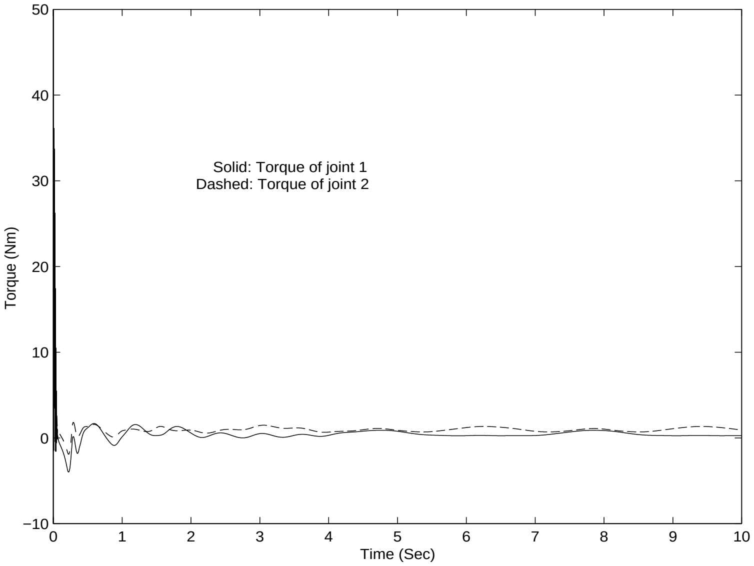

Figure 3.11: Joint Torques of the Robotic Manipulator under Neural Network based Controller

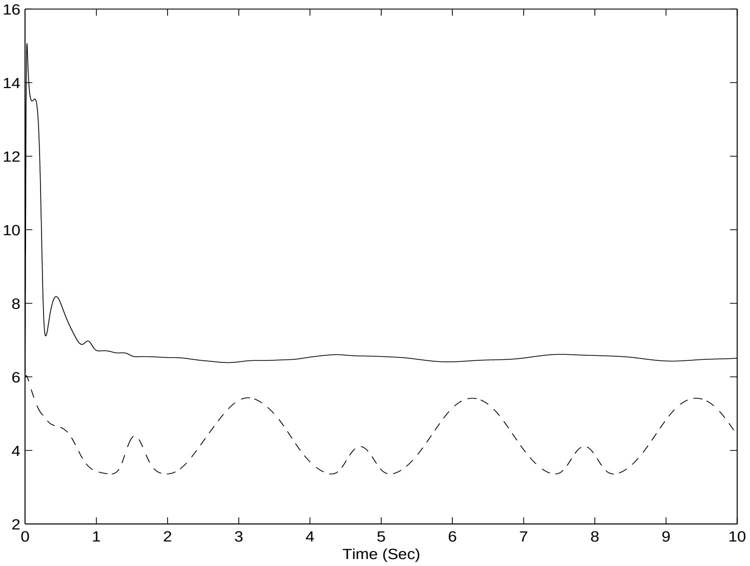

Figure 3.12: Comparison of $\|M_r\|$ (dashed)and $\|\hat{M}_r\|$(solid) under Neural Network based Controller

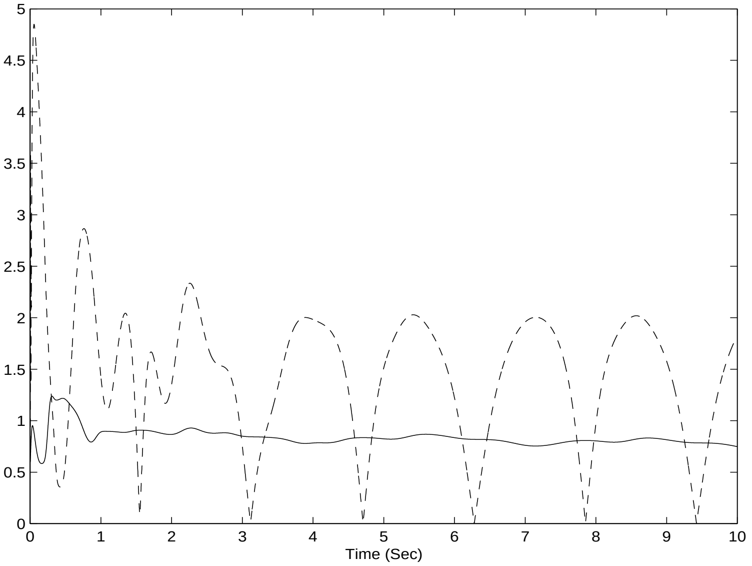

Figure 3.13: Comparison of $\|C_r\|$ (dashed) and $\|\hat{C}_r\|$(solid) under Neural Network based Controller

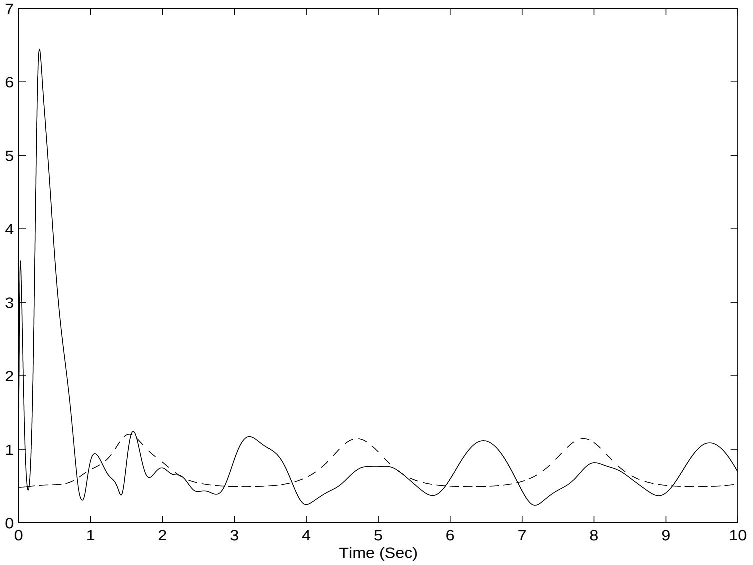

Figure 3.14: Comparison of $\|G_r\|$ (dashed) and $\|\hat{G}_r\|$ (solid) under Neural Network based Controller

Chapter 4

Explicit Force Control of a Dynamically Constrained Robot

In most controllers for the constrained robots, constraint force is controlled indirectly while the robot position is made to converge to its desired trajectory. In those controllers, the constraint is assumed to be rigid or it is modeled as a simple spring.

In this chapter, the direct or explicit force control is addressed for the applications where the accurate constraint force is required. A more general dynamic model of the constraint is used in the controller design. Considering that the internal states of the constraint are not measurable, the adaptive output feedback control approaches are adopted in the controller design.

The rest of the chapter is organized as follows. In Section 4.1, the dynamic model of the constrained robot system is presented and some of its properties are discussed. In Section 4.2, the developments of the force controllers with MRAC and backstepping approaches are presented respectively. In Section 4.3, the simulation study is done to verify and compare the effectiveness of the controllers. The conclusion is given in Section 4.4.

4.1 Dynamic Model

The dynamic model used for the controller design is taken from [30][32][33] and is schematically shown in Figure 4.1. This model is selected because it covers various phenomena occuring during the constrained motion of the robot – rigid contact, compliant motion and collision [30]. It is more comprehensive than the mass-spring or spring models used in many constrained robot control approaches in describing the motion of the contact. It is assumed that the robot itself can be controlled properly with its own position controller, and thus only the dynamics of the robot's end effector is considered together with that of the constraint.

For simplicity and without losing generality, it is assumed that the contact between the robotic manipulator and the constraint is a point contact and the constraint force acts along the normal of the constraint at the contact point. In Figure 4.1, the displacement of the constraint δ and the force f are measured along the normal vector of the constraint. The constraint is divided into two parts: the outer layer with large stiffness and the compliant layer with small stiffness. The constants m_c, k_c and b_c are the inertia, stiffness and damping ratio respectively of the mass-spring-damping units in the compliant layer. The constants k_s, k_o and b_s are the stiffness and damping ratio in the outer layer. The variables x_1 and x_2 are the displacements of a spring (k_o) and a mass unit (m_c) respectively. All the displacement are measured from the equilibrium points of the corresponding units.

The displacement δ is related to the joint position q of the robot through the following forward kinematics of the robot:

$$\delta = \phi(q) \tag{4.1}$$

and for the undeflected position of the constraint, $\phi(q) = 0$.

Based on the classical Newton mechanics, the dynamic model of the constraint is written as [30]

$$\begin{aligned} \dot{x} &= A_c x + B_c \delta \\ f &= k_s(\delta - x_2) + k_0(\delta - x_1) \end{aligned} \tag{4.2}$$

$$= \quad Ex + D\delta \tag{4.3}$$

where

$$\begin{aligned}
x &= [x_1\; x_2\; x_3]^T, \quad x_3 = \dot{x}_2 \\
A_c &= \begin{bmatrix} -b_s^{-1}k_o & 0 & 1 \\ 0 & 0 & 1 \\ -m_c^{-1}k_o & -m_c^{-1}(k_s+k_c) & -m_c^{-1}b_c \end{bmatrix}, \quad B_c = \begin{bmatrix} b_s^{-1}k_o \\ 0 \\ m_c^{-1}(k_o+k_s) \end{bmatrix} \\
E &= [-k_o\; -k_s\; 0], \quad D = k_o + k_s
\end{aligned}$$

In the above state space representation of the environment, the constraint surface displacement δ is taken as the input, and the force f is taken as the output. When the contact loses ($f = 0$), the states x and δ continue to evolve in time. The internal states of the environment x and the environment parameters are normally difficult to obtain.

Remark 4.1.1 *The dynamic constraint model represented by equations (4.2) and (4.3) contains the constraint's internal states x which are normally not measurable. The parameter matrices A_c, B_c, E and D contain the parameters of the constraints which are also unknown. As such, the adaptive output feedback approach should be used for the force controller development.*

From equations (4.2) and (4.3), the transfer function from input δ to output f is derived as

$$H(s) = \frac{F(s)}{\Delta(s)} = \frac{d_3 s^3 + d_2 s^2 + d_1 s + d_0}{s^3 + l_2 s^2 + l_1 s + l_0} \tag{4.4}$$

where $d_3 = k_o + k_s$, $d_2 = b_s^{-1}k_o k_s + m_c^{-1}b_c(k_o+k_s)$, $d_1 = m_c^{-1}b_s^{-1}b_c k_o k_s + m_c^{-1}k_c(k_o + k_s)$, $d_0 = m_c^{-1}k_c k_o k_s$, $l_2 = b_s^{-1}k_o + m_c^{-1}b_c$, $l_1 = m_c^{-1}b_s^{-1}b_c k_0 + m_c^{-1}(k_s+k_o) + m_c^{-1}k_c$ and $l_0 = m_c^{-1}b_s^{-1}(k_s+k_c)k_o$.

Equation (4.4) represents a 3rd order dynamic system which is more comprehensive than simple mass-spring or spring models used in many constrained robot control approaches. To obtain δ in equation (4.4), a command displacement δ_c to the robot position control system and its relation with δ should be established. As shown in [28] and [31], δ is related to δ_c through the following transfer function

$$\frac{\Delta(s)}{\Delta_c(s)} = \frac{k}{ms^2 + cs + k} \tag{4.5}$$

where m, k and c are the end effector mass, spring and damping parameters. Obviously the dynamics of the end effector plays an important role in shaping the relation between δ and δ_c.

From equations (4.4) and (4.5), we have

$$\frac{F(s)}{\Delta_c(s)} = \frac{b_3 s^3 + b_2 s^2 + b_1 s + b_0}{s^5 + a_4 s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0} \tag{4.6}$$

where $b_i = kd_i/m$ $(i = 0, \ \ldots 3)$, $a_4 = (ml_2 + c)/m$, $a_3 = (ml_1 + cl_2 + k)/m$, $a_2 = (ml_0 + cl_1 + kl_2)/m$, $a_1 = (cl_0 + kl_1)/m$ and $a_0 = kl_0/m$

For the dynamic model in equation (4.6), we have the following lemma.

Lemma 4.1.1 *The dynamic model in equation (4.6) is of minimum phase, or* $b_3 s^3 + b_2 s^2 + b_1 s + b_0$ *is Hurwitz.*

Lemma 4.1.1 is important for the controller development. Its proof can be found in **Appendix B**.

Remark 4.1.2 *Equation (4.6) describes the relation between the constraint force and the command displacement considering the dynamics of both the constraint and the robot's end effector. The coefficients of the equation are the functions of the system parameters.*

Remark 4.1.3 *The dynamic model (4.6) is of minimum phase and thus its is suitable for applying the backstepping method in the controller design.*

Following the same step as in [48], the state space equations in observer canonical form of the system model is

$$\dot{y}_1 = y_2 - a_4 y_1 \tag{4.7}$$

$$\dot{y}_i = y_{i+1} - a_{5-i} y_1 + b_{5-i}\delta_c \ \ (i = 2, \ 3, \ 4) \tag{4.8}$$

$$\dot{y}_5 = b_0 \delta_c - a_0 y_1 \tag{4.9}$$

or equivalently

$$\dot{y} = Ay - ay_1 + [0 \ b^T]^T \delta_c \tag{4.10}$$

$$y_1 = f = c_1^T y \tag{4.11}$$

where

$$\begin{aligned}
A &= \begin{bmatrix} 0^{4\times 1} & I^{4\times 4} \\ 0 & 0 \end{bmatrix} \\
a &= [a_4\ a_3\ a_2\ a_1\ a_0]^T \\
b &= [b_3\ b_2\ b_1\ b_0]^T
\end{aligned}$$

$I^{4\times 4}$ is an identity matrix of dimension 4, $0^{4\times 1}$ is a zero matrix with dimension 4×1, c_1 is a unit vector with the first element being 1 and the rest of elements are 0. Generally, c_j is a unit vector with the jth element being 1 and the rest of elements are 0.

Equation (4.10) is rearranged as follows

$$\dot{y} = Ay + B^T(y_1, \delta_c)\theta \tag{4.12}$$

where

$$\begin{aligned}
B^T(y_1, \delta_c) &= \begin{bmatrix} \begin{bmatrix} 0^{1\times 4} \\ I^{4\times 4} \end{bmatrix} \delta_c & -y_1 I^{5\times 5} \end{bmatrix} \in R^{5\times 9} \\
\theta &= [b^T \ \ a^T]^T \in R^9
\end{aligned}$$

The system dynamics described in observer canonical form (4.10) is needed for the design of the adaptive output feedback control.

4.2 Controller Design

In this section, an adaptive output feedback controller is developed to control the constraint force modeled by equation (4.10). The control objective is to control the constraint force f to reach its desired value f_d through the control input δ_c – the command displacement of the constraint.

The dynamic relation between the force and the command displacement (4.6) is that of a typical linear minimum-phase system. For such a system, many methods can be found to develop an adaptive controller when the system parameters are unknown. In this chapter, the controller is designed based on adaptive backstepping [48] and MRAC approaches [49] which require only the input and output measurements.

4.2.1 Adaptive Output Feedback Force Controller with Backstepping

In the following, the backstepping control approach in [48] is to be used for the controller design. The estimation of the state y is first made through the following observer [48]

$$\hat{y} = \xi + \Omega^T \theta \tag{4.13}$$

where

$$\begin{aligned} \dot{\xi} &= A_0 \xi + \lambda y_1 && (4.14) \\ \dot{\Omega}^T &= A_0 \Omega^T + B(y_1, \delta_c)^T && (4.15) \\ A_0 &= A - \lambda c_1^T && (4.16) \end{aligned}$$

and $\lambda \in R^5$ is selected to make A_0 Hurwitze, or there exists $P \in R^{5\times5}$ and $G \in R^{5\times5}$ such that

$$PA_0 + A_0^T P = -G < 0, \quad P = P^T > 0 \tag{4.17}$$

Define the column vectors of Ω^T as

$$\Omega^T = [v_3 \; v_2 \; v_1 \; v_0 \; \eta_1 \; \eta_2 \; \ldots \eta_5] \tag{4.18}$$

where $v_i \in R^5$ $(i = 1, \; \ldots 4)$ and $\eta_j \in R^5$ $(j = 1, \; 2 \; \ldots 5)$.

From equations (4.15) and (4.18) , we have

$$\begin{aligned} \dot{\eta}_i &= A_0 \eta_i - c_i y_1 && (4.19) \\ \dot{v}_i &= A_0 v_i + c_{5-i} \delta_c && (4.20) \end{aligned}$$

From equation (4.16), we have $A_0^5 c_5 = -\lambda$, $\eta_j = A_0 \eta_{j+1}$ $(j = 1, \; 2, \; 3, \; 4)$ and

$$\begin{aligned} \xi &= A_0^5 \eta_5 && (4.21) \\ \eta_j &= A_0^{5-j} \eta_5 && (4.22) \end{aligned}$$

It can be proven that if θ is known, we have

$$\begin{aligned} y &= \hat{y} + \epsilon = \xi + \Omega^T\theta + \epsilon && (4.23) \\ \dot{\epsilon} &= A_0\epsilon && (4.24) \end{aligned}$$

and $\epsilon \to 0$ when $t \to \infty$.

From equations (4.12) and (4.23), we have

$$\begin{aligned} \dot{y}_1 &= c_2^T A_0^5 \eta_5 + w^T\theta + c_2^T\epsilon = c_2^T A_0^5 \eta_5 + b_3 v_{3,2} + \overline{w}^T\theta + c_2^T\epsilon && (4.25) \\ \dot{v}_{3,2} &= v_{3,3} - \lambda_2 v_{3,1} + \delta_c && (4.26) \end{aligned}$$

where

$$\begin{aligned} w &= [v_{3,2} \;\; v_{2,2} \;\; v_{1,2} \;\; v_{0,2} \;\; (\eta_5^T A_\eta^T - y_1 c_1^T)]^T && (4.27) \\ \overline{w} &= [0 \;\; v_{2,2} \;\; v_{1,2} \;\; v_{0,2} \;\; (\eta_5^T A_\eta^T - y_1 c_1^T)]^T && (4.28) \\ A_\eta^T &= [(A_0^4)^T c_2 \;\; (A_0^3)^T c_2 \;\; (A_0^2)^T c_2 \;\; c_2] && (4.29) \end{aligned}$$

Equations (4.25) and (4.26) are in parameter strict feedback forms which are suitable for applying the backstepping approach in the controller design. In each step (i) of backstepping, a stabilizing function α_i and a tuning function τ_i are generated and the control input δ_c appears in the last step.

Step 1. Define

$$z_1 = y_1 - y_{1d} \qquad (4.30)$$

From equation (4.25), we have

$$\begin{aligned} \dot{z}_1 &= c_2^T A_0^5 \eta_5 + \overline{w}^T(\eta_5,\; y_1)\theta + c_2^T\epsilon + b_3 v_{3,2} \\ &= b_3\alpha_1 + c_2^T A_0^5 \eta_5 + \overline{w}^T\theta + b_3 z_2 + \epsilon_2 && (4.31) \end{aligned}$$

where $z_2 = v_{3,2} - \alpha_1$ and $\epsilon_2 = c_2^T\epsilon$. α_1 is called stabilizing function in this step.

To find the stabilizing function α_1, it is first rewritten as

$$\alpha_1 = \hat{d}\overline{\alpha}_1 \qquad (4.32)$$

where $\hat{d}$ is the estimate of $1/b_3$.

Substitute equation (4.32) into equation (4.31), we have

$$\dot{z}_1 = \overline{\alpha}_1 + c_2^T A_0^5 \eta_5 + \overline{w}^T \theta - b_3 \tilde{d} \overline{\alpha}_1 + b_3 z_2 + \epsilon_2 \tag{4.33}$$

If $\overline{\alpha}_1$ is chosen such that

$$\overline{\alpha}_1 = -h_1 z_1 - c_2^T A_0^5 \eta_5 - \overline{w}^T \hat{\theta} \tag{4.34}$$

equation (4.33) becomes

$$\dot{z}_1 = -h_1 z_1 + \overline{w}^T \tilde{\theta} - b_3 \tilde{d} \overline{\alpha}_1 + b_3 z_2 + \epsilon_2 \tag{4.35}$$

where $h_1 > 0$ is a control parameter and $\hat{\theta}$ is the estimate of parameters θ.

From equations (4.27), (4.28) and (4.32), we have

$$\overline{w}^T \tilde{\theta} + b_3 z_2 = (w - \hat{d} \overline{\alpha}_1 c_1)^T \tilde{\theta} + c_1^T \hat{\theta} z_2 \tag{4.36}$$

Substituting equation (4.36) into equation (4.35), we have

$$\dot{z}_1 \;\; = \;\; -h_1 z_1 + (w - \hat{d} \overline{\alpha}_1 c_1)^T \tilde{\theta} - b_3 \tilde{d} \overline{\alpha}_1 + c_1^T \hat{\theta} z_2 + \epsilon_2 \tag{4.37}$$

Consider the Lyapunov function

$$V_1 = \frac{1}{2} z_1^2 + \frac{1}{2} \tilde{\theta}^T \Gamma^{-1} \tilde{\theta} + \frac{b_3}{2\gamma} \tilde{d}^2 + \frac{1}{2} \epsilon^T P \epsilon \tag{4.38}$$

where $\Gamma > 0$, $\gamma > 0$ are the gain matrix and the gain respectively, and P is a positive definite symmetric matrix defined in equation (4.17). Note that $b_3 > 0$ by definition.

Differentiating V_1 with respect to time t and considering equation (4.37), we have

$$\begin{aligned} \dot{V}_1 \;\; = \;\; & -h_1 z_1^2 + z_1 \epsilon_2 + c_1^T \hat{\theta} z_1 z_2 - \epsilon^T G \epsilon - \frac{1}{\gamma} b_3 \tilde{d} (\dot{\hat{d}} + \gamma \overline{\alpha}_1 z_1) \\ & + \tilde{\theta}^T \Gamma^{-1} [\Gamma (w - \hat{d} \overline{\alpha}_1 c_1) z_1 - \dot{\hat{\theta}}] \end{aligned} \tag{4.39}$$

If

$$\dot{\hat{d}} \;\; = \;\; -\gamma \overline{\alpha}_1 z_1 \tag{4.40}$$

$$\dot{\hat{\theta}} \;\; = \;\; \Gamma \tau_1, \;\; \tau_1 = (w - \hat{d} \overline{\alpha}_1 c_1) z_1 \tag{4.41}$$

and $z_2 = 0$ ($v_{3,2} = \alpha_1$), we have

$$\dot{V}_1 = -(h_1 - 1)z_1^2 - (z_1 - \frac{\epsilon_2}{2})^2 + \frac{\epsilon_2^2}{4} - \epsilon^T G \epsilon \tag{4.42}$$

Letting $h_1 > 1$ and G be in a diagonal form $G = diag(g_i)$ with $g_i > 0$ and $g_2 > \frac{1}{4}$ ($i = 1, 2, \ldots 2n$) and from equation (4.42), we have

$$\dot{V}_1 = -(h_1 - 1)z_1^2 - (z_1 - \frac{\epsilon_2}{2})^2 - (g_2 - \frac{1}{4})\epsilon_2^2 - \Sigma_{j=1,\, j\neq 2}^{5} g_j \epsilon_j^2 \leq 0 \tag{4.43}$$

As $v_2 \neq \alpha_1$ ($z_2 \neq 0$) and $\dot{\hat{\theta}} \neq \Gamma\tau_1$, it follows that

$$\dot{V}_1 \leq -(h_1 - 1)z_1^2 + c_1^T \hat{\theta} z_1 z_2 + \tilde{\theta}^T(\tau_1 - \Gamma^{-1}\dot{\hat{\theta}}) \tag{4.44}$$

Step 2. Differentiating z_2 with respect to time t, we have

$$\begin{aligned} \dot{z}_2 &= v_{3,3} - \lambda_2 v_{3,1} + \delta_c - \hat{d}\dot{\overline{\alpha}}_1 - \overline{\alpha}_1 \dot{\hat{d}} && (4.45) \\ &= v_{3,3} + \delta_c - \lambda_2 v_{3,1} - \hat{d}\dot{\overline{\alpha}}_1 - \overline{\alpha}_1 \dot{\hat{d}} && (4.46) \end{aligned}$$

From equation (4.34), we have

$$\dot{\alpha}_1 = \frac{\partial \overline{\alpha}_1}{\partial \eta_5}\dot{\eta}_5 + \frac{\partial \overline{\alpha}_1}{\partial y_1}\dot{y}_1 + \frac{\partial \overline{\alpha}_1}{\partial \hat{\theta}}\dot{\hat{\theta}} \tag{4.47}$$

Substituting equation (4.47) into equation (4.45), we have

$$\dot{z}_2 = \delta_c + v_{3,3} - \gamma_2(w^T\tilde{\theta} + \epsilon_2) - \hat{d}\frac{\partial \overline{\alpha}_1}{\partial \hat{\theta}}\dot{\hat{\theta}} - \beta_2 \tag{4.48}$$

where

$$\gamma_2 = \hat{d}\frac{\partial \overline{\alpha}_1}{\partial y_1} \tag{4.49}$$

$$\beta_2 = \lambda_2 v_{3,1} + \hat{d}\frac{\partial \overline{\alpha}_1}{\partial \eta_5}(A_0\eta_5 - c_5 y_1) - \gamma\overline{\alpha}_1^2 z_1 + \gamma_2(c_2^T A_0^5 \eta_5 + w^T\hat{\theta}) \tag{4.50}$$

with all the signals being measurable.

Consider the following Lyapunov function

$$V_2 = V_1 + \frac{1}{2}z_2^2 + \frac{1}{2}\epsilon^T P \epsilon \tag{4.51}$$

Differentiating V_2 with respect to time t and considering equations (4.44 and (4.48), we have

$$\begin{aligned}
\dot{V}_2 &\leq -(h_1-1)z_1^2 + c_1^T\hat{\theta}z_1z_2 + \tilde{\theta}^T(\tau_1 - \Gamma^{-1}\dot{\hat{\theta}}) + z_2\dot{z}_2 - \epsilon^T G\epsilon && (4.52)\\
&= (1-h_1)z_1^2 + \tilde{\theta}^T(\tau_1 - \gamma_2 w z_2 - \Gamma^{-1}\dot{\hat{\theta}}) && (4.53)\\
&+ z_2(\delta_c + v_{3,3} + c_1^T\hat{\theta}z_1 - \beta_2 - \hat{d}\frac{\partial\overline{\alpha}_1}{\partial\hat{\theta}}\dot{\hat{\theta}}) - \gamma_2 z_2\epsilon_2 - \epsilon^T G\epsilon && (4.54)
\end{aligned}$$

By selecting

$$\begin{aligned}
\dot{\hat{\theta}} &= \Gamma\tau_2, \quad \tau_2 = \tau_1 - \gamma_2 w z_2 && (4.55)\\
\delta_c &= -h_2z_2 - c_1^T\hat{\theta}z_1 + \beta_2 + \frac{\partial\alpha_1}{\partial\hat{\theta}}\Gamma\tau_2 - v_{3,3}, \quad h_2 > 1 && (4.56)
\end{aligned}$$

and from equation (4.54), we have

$$\begin{aligned}
\dot{V}_2 \leq\; & -(h_1-1)z_1^2 - (h_2-1)z_2^2 - (z_2 + \frac{\gamma_2\epsilon_2}{2})^2\\
& -(g_2 - \frac{\gamma_2^2}{4})\epsilon_2^2 + \Sigma_{j=1,\, j\neq 2}^{5} g_j\epsilon_j^2 < 0 && (4.57)
\end{aligned}$$

where g_2 is selected such that

$$g_2 > \frac{\gamma_2^2}{4} \qquad (4.58)$$

Note that the equation (4.56) defines the control input δ_c.

Substituting equations (4.55) and (4.56) in equation (4.48), we have

$$\dot{z}_2 = -h_2z_2 - c_1^T\hat{\theta}z_1 - \gamma_2(w^T\tilde{\theta} + \epsilon_2) \qquad (4.59)$$

Combining equations (4.37) and (4.59), we have the following error system

$$\dot{z} = W_z z + W_\theta\tilde{\theta} - b_3\overline{\alpha}_1 c_1\tilde{d} + W_\epsilon\epsilon_2 \qquad (4.60)$$

where

$$\begin{aligned}
W_z &= \begin{bmatrix} -h_1 & c_1^T\hat{\theta} \\ -c_1^T\hat{\theta} & -h_2 \end{bmatrix}\\
W_\epsilon &= [1 \quad -\gamma_2]^T\\
W_\theta &= W_\epsilon w^T - \hat{d}\overline{\alpha}c_1c_1^T
\end{aligned}$$

Note that

$$W_z + W_z^T = -2\ diag[h_i], \quad i = 1,\ 2\ \ldots 2n \tag{4.61}$$

The stability of the above error systems is established as $\dot{V}_2 \leq 0$ under the control law δ_c. It can be concluded that z, $\tilde{\theta}$, $\tilde{d}$ and ϵ are all bounded. As the desired force y_{1d} is bounded, the output y_1 is also bounded. From equation (4.20) and the fact that the system dynamics (4.6) is of minimum phase, $v_i (i = 1,\ 2)$ are bounded. Thus, all of the signals in the closed-loop system are bounded, $z(t) \to 0$ (i.e., $y_1 \to y_{1d}$) as $t \to \infty$.

The results of the above discussion are summarized in the following theorem.

Theorem 4.2.1 *For the constrained robot system modeled by equations (4.6) and (4.5), the constraint force f approaches to its desired value f_d when $t \to \infty$ if the command displacement of the constraint is given by equation (4.56) and the uncertain parameters $\hat{d}$ and $\hat{\theta}$ are tuned by adaptation laws (4.40) and (4.55) respectively. The closed loop signals are also bounded.*

Remark 4.2.1 *The adaptive output feedback controller with backstepping is based on the 3rd dynamic model (4.4) of the constraint. It is a special case of general chain multiple mass spring damper (CMMSD) systems with any degrees of freedom introduced in* **Appendix C**. *Though the controller in* **Appendix C** *is for position control, it can be readily used for force control by replacing the position variable with the force variable.*

In addition to the adaptive output feedback controller with backstepping discussed above, we will show that the same objective can also be achieved through model reference adaptive control approach in the next section.

4.2.2 MRAC Based Adaptive Output Feedback Force Controller

This controller is designed with the MRAC approach in [49]. The desired behavior of the constraint force is specified by the following reference model

$$f_m = \frac{B_m}{A_m} f_d \tag{4.62}$$

where $B_m = \omega^2$, $A_m = p^2 + 2\zeta\omega p + \omega^2$, $p = d/dt$ is the differential operator, f_d is the command input (desired constraint force) and $f_m(t)$ is the desired output of the reference model. From the linear control theory, $f_m(t) \rightarrow f_d$ when ω and ζ are selected properly.

The dynamic relation between the force and the command displacement (4.6) is re-written as

$$f = \frac{b_3 B_f}{A_f} \delta_c \tag{4.63}$$

where $B_f = p^3 + b_2 b_3^{-1} p^2 + b_1 b_3^{-1} p + b_0 b_3^{-1}$ and $A_f = p^5 + a_4 p^4 + a_3 p^3 + a_2 p^2 + a_1 p + a_0$

Our task now is to find a control input δ_c such that the closed-loop system follows the reference model (4.62). According to the poles placement procedure in [49], the exact model following is achieved if δ_c satisfies

$$P_r \delta_c = P_t f_d - P_s f \tag{4.64}$$

where $P_s = s_4 p^4 + s_3 p^3 + s_2 p^2 + s_1 p + s_0$, $P_t = t_4 p^4 + t_3 p^3 + t_2 p^2 + t_1 p + t_0$ and $P_r = P_{r1} B_f = p^4 + r_3 p^3 + r_2 p^2 + r_1 p + r_0$ and their coefficients s_i, t_i and r_i $(i = 0,\ 1, \dots 4)$ can be solved from the following equations

$$\begin{aligned} A_f P_{r1} \quad + \quad b_3 P_s &= P_o A_m & (4.65) \\ P_t \quad &= \quad P_o B_m / b_3 & (4.66) \end{aligned}$$

given an pre-defined *observer polynomial* $P_o = p^4 + o_3 s^3 + o_2 s^2 + o_1 s + o_0$.

Equation (4.65) is called *Diophantine* equation. Observer polynomial P_o is selected such that it is stable with a faster dynamic characteristic than that of A_m. From equations (4.64) and (4.65), the coefficients of P_r, P_s and P_t are derived such that $r_0 = r b_0$, $r_1 = r b_1 b_3^{-1} + b_0,$, $r_2 = r b_2 b_3^{-1}$, $r_3 = r + b_2 b_3^{-1}$, $t_i = (\omega^2/b_3) o_i$,

$s_i = (\omega^2 o_i + 2\zeta\omega o_{i-1} + o_{i-2} - a_{i-1} - a_i r)/b_3$ $(i = 0, \ldots 3)$, $t_4 = \omega^2/b_3$, $s_4 = (\omega^2 + 2\zeta\omega o_3 + o_2 - a_3 - a_4 r)/b_3$ and $r = o_3 - a_4 + 2\zeta\omega$.

The control parameters r_i, s_i and t_i $(i = 0, \ldots 4)$ are the functions of the parameters of the system reference model and the observer polynomial. The parameter adaptation laws are needed for their estimates as the system parameters a and b are unknown.

From equations (4.62), (4.63) and (4.64), the error between the output of the controlled loop and the reference model is obtained

$$e = f - f_m = \frac{b_3}{P_o A_m}(P_r \delta_c + P_s f - P_t f_d) \tag{4.67}$$

To express e in a linear-in-parameter (LIP) form, re-arrange equation (4.67) such that

$$e = b_3(\frac{1}{P_1}\delta_c + \frac{P_r - P_2}{P_f}\delta_c + \frac{P_s}{P_f}f - \frac{P_t}{P_f}f_d) \tag{4.68}$$

where $P_f = P_1 P_2$, $P_1 = A_m$ and $P_2 = P_o$. Obviously, $P_r - P_2$ is a polynomial of p with coefficients being $r'_3 = r_3 - o_3$, $r'_2 = r_2 - o_2$, $r'_1 = r_1 - o_1$ and $r'_0 = r_0 - o_0$.

Define a vector of the coefficients of the polynomials $P_r - P_2$, P_s and P_t such that

$$\theta_f = [r'_3 \; \ldots r'_0 \; s_4 \; \ldots s_0 \; t_4 \; \ldots t_0]^T \tag{4.69}$$

and another vector consisting of filtered input , output and the command inputs such that

$$\varphi = [\frac{p^3}{P_f}\delta_c \ldots \frac{1}{P_f}\delta_c \; \frac{p^4}{P_f}f \ldots \frac{1}{P_f}f \; \frac{-p^4}{P_f}f_d \ldots \frac{-1}{P_f}f_d]^T \tag{4.70}$$

With φ and θ_f defined above, the error e is expressed in a more compact form

$$e = b_3(\frac{1}{P_1}\delta_c + \varphi^T \theta_f) \tag{4.71}$$

Let $\hat{\theta}_f$ be the estimate of θ_f and define the output feedback control law as

$$\delta_c = -\hat{\theta}_f^T (P_1 \varphi) \tag{4.72}$$

Substituting the control law (4.72) in the equation (4.71), we have

$$e = \varepsilon + b_3 \eta \tag{4.73}$$

where $\varepsilon = b_3\varphi^T\tilde{\theta}_f$, $\eta = -\delta_c/P_1 - \varphi^T\hat{\theta}_f$ and $\tilde{\theta}_f = \theta_f - \hat{\theta}_f$.

It is obvious that ε (called *augmented error*) is linear in $\tilde{\theta}_f$ and thus $\hat{\theta}$ can be tuned adaptively through *gradient approach* [49] such that

$$\dot{\hat{\theta}}_f = \gamma_f\varphi\varepsilon \tag{4.74}$$

where $\gamma_f > 0$ is the adaptation gain. Note that the unknown parameter b_3 is absorbed in γ_f .

If b_3 is unknown, the augmented error ε in equation (4.74) can be replaced by *prediction error*

$$\varepsilon_p = e - \hat{b}_3(\varphi^T\hat{\theta}_f + \frac{1}{P_1}\delta_c) \tag{4.75}$$

where

$$\dot{\hat{b}}_3 = \gamma_f(\varphi^T\hat{\theta}_f + \frac{1}{P_1}\delta_c)\varepsilon_p \tag{4.76}$$

and $\hat{\theta}$ is now estimated through

$$\dot{\hat{\theta}}_f = \gamma_f\varphi\varepsilon_p \tag{4.77}$$

Note that the adaptation law (4.74) is a special case of adaptation law (4.77) when b_3 is known.

Following the same procedure to prove the stability of general MRAC controllers [49][50], it can be showed that under the controller (4.72) and adaptation laws (4.76) and (4.77), $f \to f_m$ asymptotically. As $f_m \to f_d$, thus we can conclude $f \to f_d$ asymptotically. It can also be showed that the close loop signals are all bounded.

The above results can be summarized in the following theorem.

Theorem 4.2.2 *For the constrained robot system where the relation between the displacement of the constraint surface and the command displacement is shaped by equation (4.63) and with an assumption that its position can be well controlled by the robot's position controller, the constraint force f approaches to its desired value f_d when $t \to \infty$ if the command displacement of the constraint is given by equation (4.72) and the parameters $\hat{b}_3$ and $\hat{\theta}_f$ are tuned by adaptation laws (4.76) and (4.77). The closed loop signals are also bounded.*

Note that the controller (4.72) and adaptation laws (4.76) and (4.77) require more control parameters (the coefficients of the reference model (4.62), observer polynomials P_o and adaptation gains) than those required backstepping approach. Several filters are also needed to filter the outputs, command inputs and the control inputs.

4.3 Simulation

For simulation, the true parameters of the end effector are selected as $k = 10$, $m = 20$ and $c = 0.5$. The constraint parameters are selected as $k_c = 10\ N/m$, $b_c = b_s = 1.0\ Ns/m$, $m_c = 50\ kg$. The desired constraint force is set to be $f_d = 1N$ and the initial values of the force is set to be 1.5 N.

The rest of the parameters varies based on different stiffness of the constraint surface such that

Case 1: $k_0 = k_s = 20$. The stiffness of the out surface of the constraint is closer to that of the compliant structure.

Case 2: $k_0 = k_s = 50$. The stiffness of the outer surface of the constraint is higher than that of the compliant structure;

Case 3: $k_0 = k_s = 80$. The constrain surface is much stiffer than compliant structure;

Case 4: $k_0 = k_s = 20$ (same as those of Case 1) and the adaptive tuning of the parameters in backstepping approach is stopped.

Case 5: $k_0 = k_s = 20$ (same as those of Case 1) and the MRAC adaptive output feedback controller is used.

Note that the values of the parameters k, m and c are chosen so that they make the second order system (4.5) stable. The parameters of the constraint k_c, b_c, b_s and m_c vary with the material and structure of the constraint [32]. Their values in simulation are set for a constraint surface softer than that in [32] where k_c and b_c can be up to 2000 N/m and 90 Ns/m respectively.

From Case 1 to Case 3, the control parameters are chosen such that $h_1 = h_2 = 2$, $\gamma = 0.01$ and $\Gamma = diag(45,\ 60,\ 60,\ 60,\ 45,\ 45,\ 45,\ 45,\ 10)$. To make A_0 positive definite, we choose $\lambda = [12\ 5\ 15\ 2\ 4]^T$. The nominal parameter vectors $\theta = [20\ 200.3\ 8\ 40\ 20.04\ 21.52\ 27.01\ 0.9\ 8]^T$, $d = 0.05$ and the initial value of $\hat{\theta}$ and $\hat{d}$ are set as 0.65θ and $0.65d$.

Case 4 is used to study the results for the controller with backstepping when the adaptation is activated or stopped by setting $\gamma = 0$ and $\Gamma = diag[0] \in R^{9\times 9}$ respectively. The other control parameters are kept the same as those of Case 1 to Case 3.

Case 5 is designed to study the performance of the MRAC adaptive output feedback controller and to compare it with that of backstepping approach. The parameters of the reference model are set as $\zeta = 0.7$ and $\omega = 0.5$. The observer polynomial P_o is specified by $P_o = p^4 + 19p^3 + 65p^2 + 77p + 30$. The true values of the parameter θ_f are $\theta_f = [-9.33\ -68.1\ -37.14\ -43.8\ 3.2\ 5.38\ 5.43\ 1.63\ 0.5\ 0.012\ 0.24\ 0.81\ 0.96\ 0.38]^T$. The initial values of $\hat{\theta}_f$ and $\hat{b}_3$ are set to be 0.65θ and $0.65b_3$ respectively. The parameter adaptation gain is set as $\gamma_f = 0.8$.

After simulations, displacements and force responses for Cases 1, 2 and 3 are plotted in Figures 4.2 and 4.3 respectively. The parameter adaptations in Case 1 are plotted from Figure 4.4 to 4.6. The comparison of the performances of adaptive control (Case 1) to non-adaptive control (Case 4) is also made and the simulation results are plotted in Figures 4.7 and 4.8 respectively.

Studying the simulation results for Cases 1, 2 and 3 in Figures 4.2 and 4.3, it can be seen that the constraint force is convergent in each case though the stiffness of the constraint surface varies. The command displacement shows a surge at the beginning, but becomes smooth within a small range quickly. Some fast and big chattering of the force and command displacement appear in Cases 2 and 3 where the stiffness of the constraint is big. The results of parameter estimations for Case 1 (Figures 4.4 to 4.6) show that the parameters are convergent but are different from their true values. This fact doesn't affect the force tracking of the controller. To make the parameter estimation's error zero, more stringent conditions such as persistent excitations are needed [47]. As for Case 4 where the adaptation process stops, the force response cannot be converged to the desired value while that of adaptive control achieves a good force tracking as shown in Figure 4.8.

The simulation results for Case 5 are plotted from Figures 4.9 to 4.11. It can be seen that the output force f converged to its desired values $f_d = 1$ and the estimated parameters are bounded (note only four parameters in the same scale are shown to save the space). Compared with the results in Case 1, the force takes a longer time to settle with a larger over shoot, but the frequency of its fluctuation is much less.

4.4 Conclusion

In this chapter, two adaptive output feedback controllers have been presented to achieve the explicit force control of robots with dynamic constraints whose parameters and internal states are unknown. While most explicit force control schemes rely on a simple model of the contact (general spring), a very general and comprehensive dynamic model of the contact is used in this chapter which covers various behaviors of the constraint such as rigid contact, compliant motion and collision with the robot. Another advantage of the proposed controllers is that they require the measurement of the output (force) only and does not need the knowledge of system parameters and internal states of the constraint. The asymptotical stability of the force error is guaranteed by the controllers. The simulations are used to verify and compare the effectiveness of the control approaches.

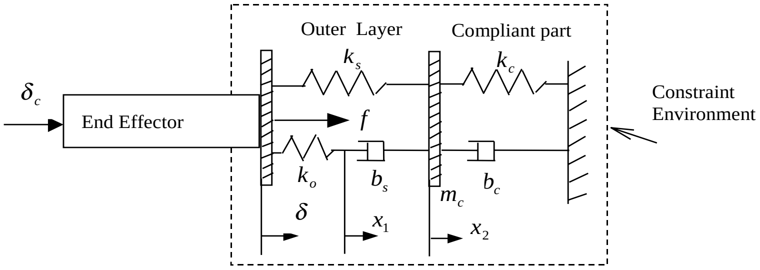

Figure 4.1: Schematic of the constraint and the end effector

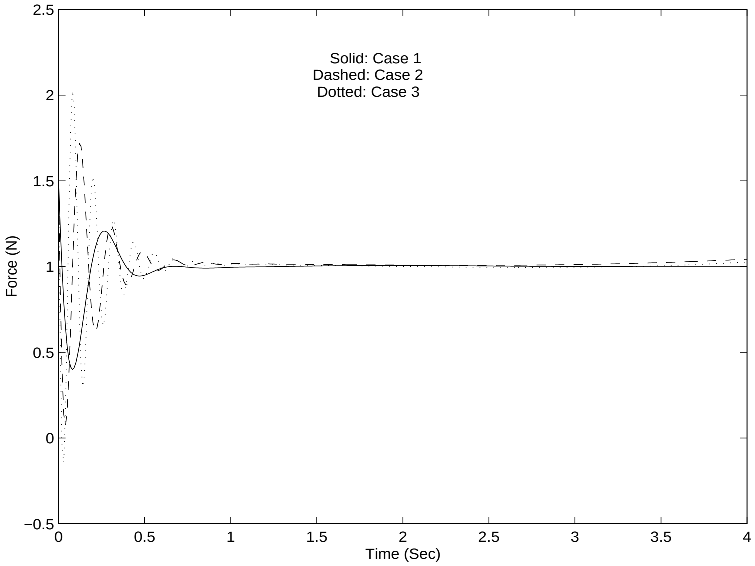

Figure 4.2: Force response (Backstepping Approach) – f

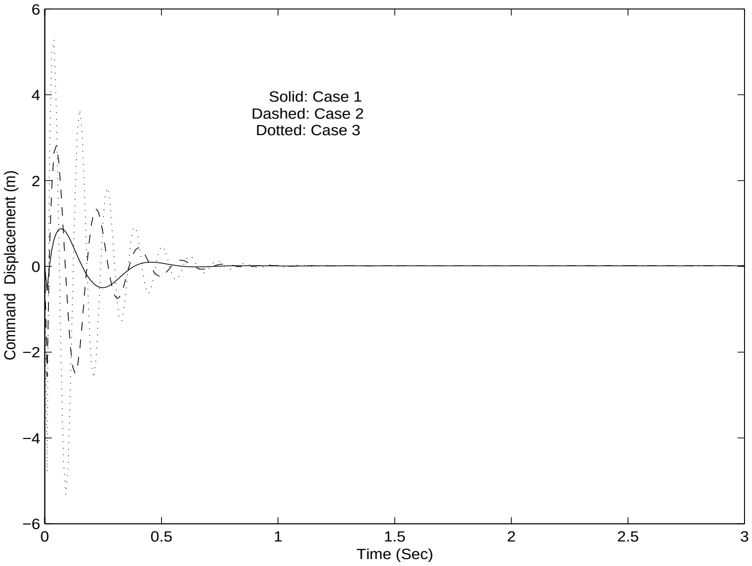

Figure 4.3: Command displacements (Backstepping Approach) – δ_c

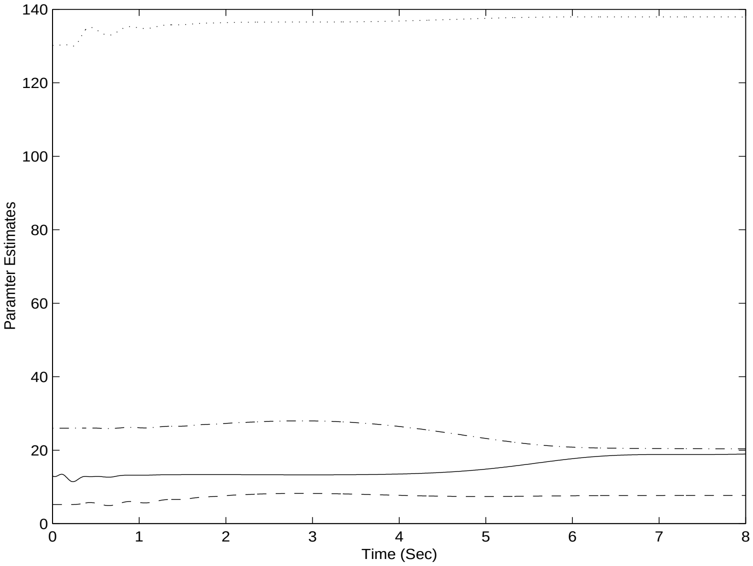

Figure 4.4: Parameters Estimation (Backstepping Approach) –Solid: $\hat{\theta}_1$,dotted: $\hat{\theta}_2$, dashed: $\hat{\theta}_3$, dashdot: $\hat{\theta}_4$

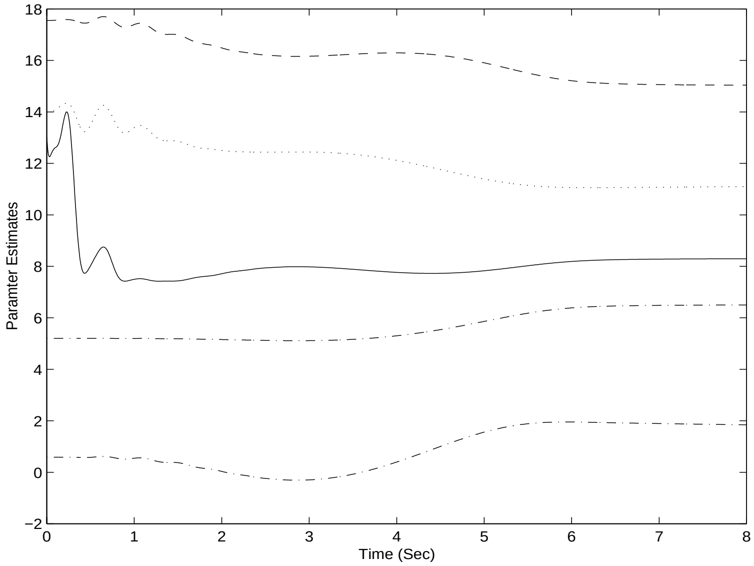

Figure 4.5: Parameters Estimation (Backstepping Approach)– Solid: $\hat{\theta}_5$, dotted: $\hat{\theta}_6$, dashed: $\hat{\theta}_7$, dashdot: $\hat{\theta}_8$, thick dashdot: $\hat{\theta}_9$

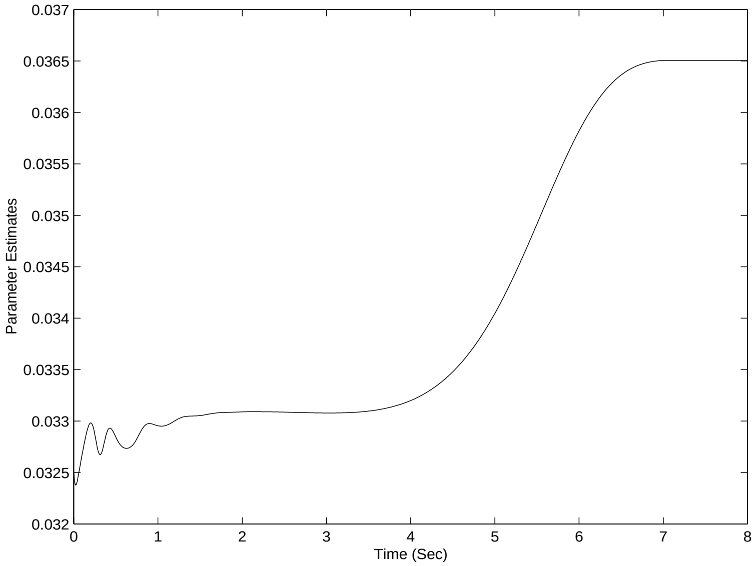

Figure 4.6: Parameter Estimation (Backstepping Approach): $\hat{d}$

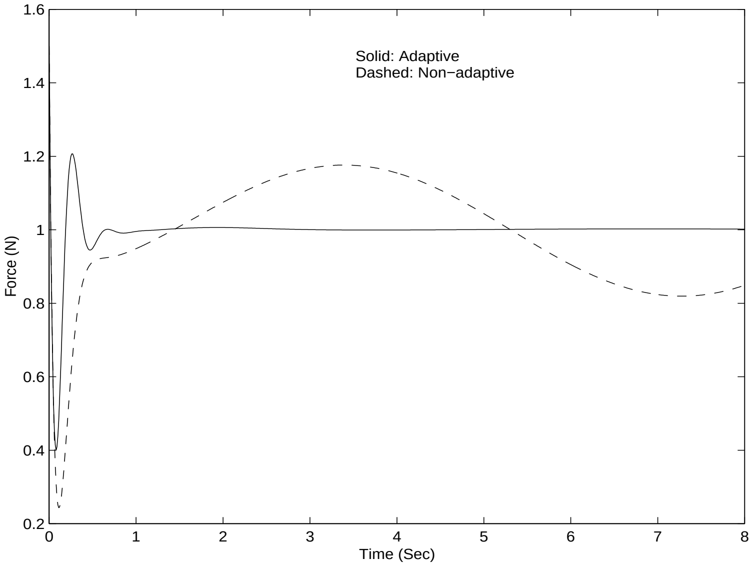

Figure 4.7: Force response (Backstepping Approach with/without parameters adaptation) – f

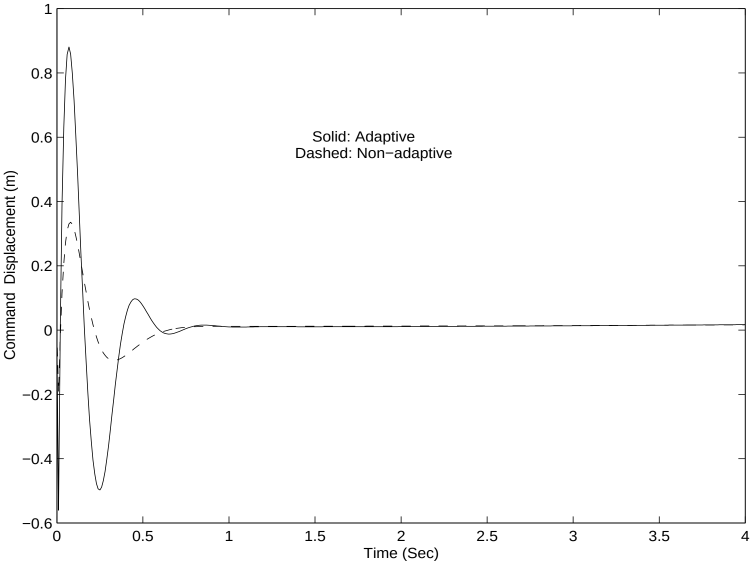

Figure 4.8: Command displacements (Backstepping Approach without/without parameters adaptation) – δ_c

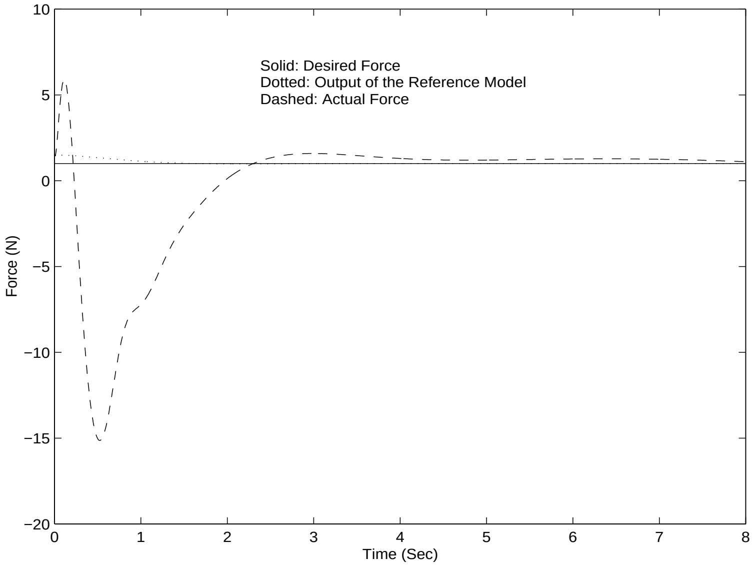

Figure 4.9: Force response (MRAC Approach) – f

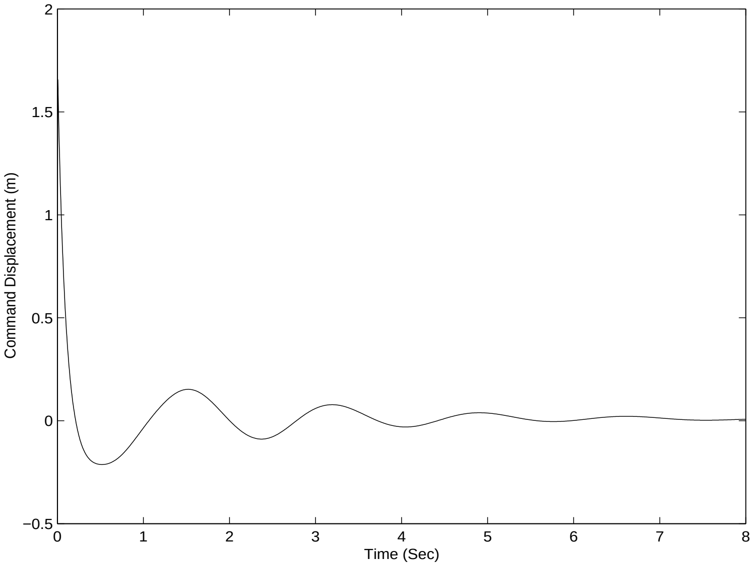

Figure 4.10: Command displacement (MRAC Approach) – f

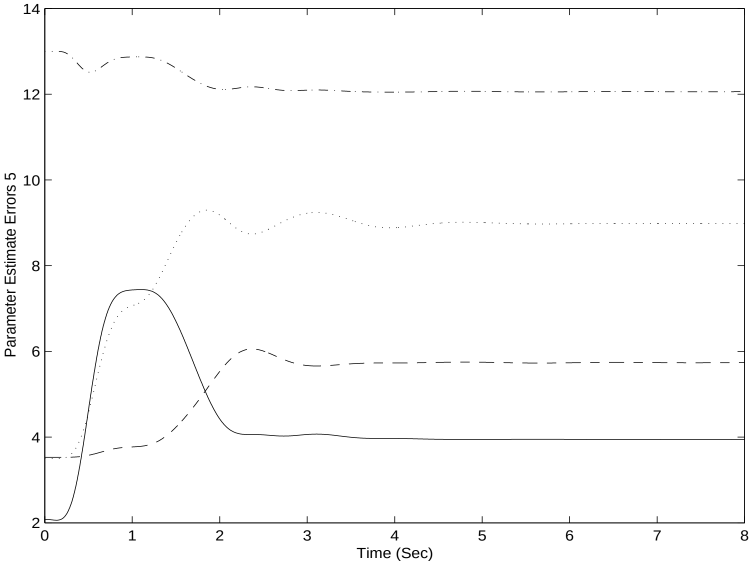

Figure 4.11: Some Parameter Estimates (MRAC Approach) – Solid: $\hat{\theta}_f(5)$, dotted: $\hat{\theta}_f(6)$, dashed: $\hat{\theta}_f(7)$, dashdot: $\hat{b}_3$

Chapter 5

Fuzzy Unidirectional Force Control of Constrained Robots

This chapter studies how to keep the contact between the end effector of the robot and the constraint surface during the robot's constrained motion, a key assumption for many constrained robot controllers.

Force control schemes developed on the assumption that the robot's end effector always keeps a contact with the constraint cannot handle the impact caused by the lose of contact. It is difficult to model the robot's state during the transition from non-contact into contact and vice-versa. The models established with some assumptions are also too complicated for the controller design [32][40]. Noting the relation between the unidirectionality of the constraint force and the maintenance of the contact of the robot's end effector on the constraint surface, a fuzzy unidirectional force control is developed based on the general impedance between the robotic arm and the constraint. The simulation is carried out to verify the effectiveness of the approach.

The chapter is organized as follows. In Section 5.1, the dynamic model of the constrained robot is given. In Section 5.2, the impedance model of the robot and environment is described and the fuzzy unidirectional force control scheme is developed. In Section 5.3, simulation studies are carried out to show the effectiveness of the proposed controller. In Section 5.4, the conclusion about the work in this chapter is described.

5.1 Dynamic Model

As presented in Chapter 3, the dynamic model of a constrained robotic manipulator in joint space is described by

$$M(q)\ddot{q} + C(q,\ \dot{q})\dot{q} + G(q) = \tau + J^T(q)f \tag{5.1}$$

where q are the joint displacements, $\dot{q}$ are the joint velocities, $M(q)$ is the inertia matrix, $C(q, \dot{q})$ is the coriolis and centrifugal force matrix, $G(q)$ is the gravitational force, τ are the joint torques, $J(q)$ is the Jacobian matrix and f is the contact force.

Through the following kinematic relations between the Cartesian position r and velocity $\dot{r}$ of the end effector and joint position q and velocity $\dot{q}$

$$\begin{aligned} r &= \phi(q) && (5.2)\\ \dot{r} &= J(q)\dot{q} && (5.3) \end{aligned}$$

the dynamic model (5.1) is expressed in workspace as

$$M_r(q)\ddot{r} + C_r(q,\ \dot{q})\dot{r} + G_r(q) = J^{-T}(q)\tau + f \tag{5.4}$$

where

$$\begin{aligned} M_r(q) &= J^{-T}(q)M(q)J^{-1}(q)\\ C_r(q,\ \dot{q}) &= J^{-T}(q)(M(q)\dot{J}^{-1}(q) + C(q,\dot{q})J^{-1}(q))\\ G_r(q) &= J^{-T}(q)G(q) \end{aligned}$$

For clarity, the arguments of the terms will be dropped if there is no ambiguity in the context of the discussion in the following.

5.2 Controller Design

Our discussion begins with commonly-used impedance control scheme in which the closed-loop dynamics is specified by a general impedance described by equation (3.3) in Chapter 3. It is reproduced below for completeness in discussion.

$$f_d - f = M_m(\ddot{r}_d - \ddot{r}) + D_m(\dot{r}_d - \dot{r}) + K_m(r_d - r) \tag{5.5}$$

where $M_m \in R^{m\times m}$, $D_m \in R^{m\times m}$ and $K_m \in R^{m\times m}$ are the constant inertia matrix, damping matrix and the stiffness matrix respectively, $f_d \in R^m$ is the desired constraint force and $f \in R^m$ is the actual constraint force

If M_m, D_m and K_m are taken as diagonal matrices: $M_m = m_m I^{l\times l}$, $D_m = d_m I^{l\times l}$, $K_m = k_m I^{l\times l}$ with m_m, d_m and k_m being constant scalars and l being the dimension of work space, the resulted impedance is then re-written as

$$e_f = m_m\ddot{e}_r + d_m\dot{e}_r + k_m e_r \tag{5.6}$$

where $e_f = f_d - f$, $\ddot{e}_r = \ddot{r}_d - \ddot{r}$, $\dot{e}_r = \dot{r}_d - \dot{r}$, $e_r = r_d - r$.

Considering the following controller

$$\tau = J^T[M_r(\ddot{r}_d + m_m^{-1}(d_m\dot{e}_r + k_m e_r + f - f_d)) + C_r\dot{r} + G_r - f] \tag{5.7}$$

and substituting it into the dynamic model (5.4) and considering the properties (**Property 3.1** and **Property 3.2** in Chapter 3) of the model, it is easy to verify that the desired impedance (5.6) is achieved.

Obviously, the resulted impedance doesn't guarantee that f is unidirectional, as the force in a normal mass-spring-damper system can act in pushing or pulling direction. In practice f should be acted along the normal pointing out of the constraint surface at the contact point, or mathematically

$$f = f_m n \tag{5.8}$$

where n is the normal vector at he contact point on the constraint surface and $f_m = \|f\|$ is the magnitude of the constraint force.

Considering equation (5.8), the impedance model (3.3) is re-written as

$$f_{dm}n - f_m n = m_m\ddot{e}_r + d_m\dot{e}_r + k_m e_r \tag{5.9}$$

$$0 < f_m < f_{max} \tag{5.10}$$

where $f_{dm} = \|f_d\|$ is the desired magnitude of the contact force and f_{max} is an additional constant representing the maximum contact force allowed.

Projecting equation (5.9) along n by multiplying its both sides with n^T and rearranging equation (5.10), we have

$$e_{fm} = m_m n^T\ddot{e}_r + d_m n^T\dot{e}_r + k_m n^T e_r \tag{5.11}$$

$$e_{fmin} < e_{fm} < e_{fmax} \tag{5.12}$$

where $e_{fm} = f_{dm} - f_m$, $e_{fmin} = f_{dm} - f_{max}$ and $e_{fmax} = f_d$

Note that the equation (5.11) describes the behaviors of contact force along the normal n and its relations with state tracking errors of the system. Equation (5.12) specifies the condition to keep the contact between the end effector of the robot and constraint surface.

By treating d_m and k_m as the weights determining the contributions of $\ddot{e}_r$, $\dot{e}_r$ and e_r to the overall force difference respectively, they can be adjusted for the realization of unidirectional force control. The trends of the changes in acceleration/velocity/position errors and force errors can be used to determine how these adjustments should be made. By observing equation (5.11), the following fuzzy rules are thus derived to adjust d_m and k_m.

Fuzzy Rules Set 1:

- IF $n^T\dot{e}_r$ is *positive* and $n^T e_r$ is *positive* and $e_{fm} - e_{fmax}$ is *positive* THEN d_m is *small* and k_m is *small*,
- IF $n^T\dot{e}_r$ is *positive* and $n^T e_r$ is *positive* and $e_{fm} - e_{fmin}$ is *negative* THEN d_m is *big* and k_m is *big* ,
- IF $n^T\dot{e}_r$ is *positive* and $n^T e_r$ is *negative* and $e_{fm} - e_{fmax}$ is *positive* THEN d_m is *small* and k_m is *big* ,

- IF $n^T \dot{e}_r$ is *positive* and $n^T e_r$ is *negative* and $e_{fm} - e_{fmin}$ is *negative* THEN d_m is *big* and k_m is *small*,
- IF $n^T \dot{e}_r$ is *negative* and $n^T e_r$ is *positive* and $e_{fm} - e_{fmax}$ is *positive* THEN d_m is *big* and k_m is *small*,
- IF $n^T \dot{e}_r$ is *negative* and $n^T e_r$ is *positive* and $e_{fm} - e_{fmin}$ is *negative* THEN d_m is *small* and k_m is *big*,
- IF $n^T \dot{e}_r$ is *negative* and $n^T e_r$ is *negative* and $e_{fm} - e_{fmax}$ is *positive* THEN d_m is *big* and k_m is *big*,
- IF $n^T \dot{e}_r$ is *negative* and $n^T e_r$ is *negative* and $e_{fm} - e_{fmin}$ is *negative* THEN d_m is *small* and k_m is *small*,
- IF $e_{fm} - e_{fmin}$ is *positive* and $e_{fm} - e_{fmax}$ is *negative* THEN d_m is *medium* and k_m is *medium.*

where *positive*, *negative*, *big*,*small* and *constant* are linguistic terms Following the same methods in [61], the membership functions of their corresponding fuzzy sets are selected as

$$\mu_{positive}(x) = \frac{1}{1+e^{-k_p x}} \tag{5.13}$$

$$\mu_{negative}(x) = \frac{1}{1+e^{k_n x}} \tag{5.14}$$

$$\mu_{big}(y) = e^{-k_b(y-d_b)^2} \tag{5.15}$$

$$\mu_{small}(y) = e^{-k_s(y-d_s)^2} \tag{5.16}$$

$$\mu_{medium}(y) = e^{-k_{med}(y-d_{med})^2} \tag{5.17}$$

where x takes as $\dot{e}_r$, e_r or e_f, y takes d_m or k_m, k_p, k_n, k_b, k_s, k_{med}, d_b, d_s and d_{med} are the positive constants determining the shapes of the membership functions.

Note that m_m is not tuned due to the difficulty to obtain the acceleration feedback in practice. Even though, the above fuzzy rules are still valid. m_m can be set to a small value to reduce the contribution to the overall control effort due to the acceleration errors.

How to choose parameters k_p, k_n, k_s, k_{med}, k_b, d_s, d_{med} and d_b relies on the knowledge about the constrained robot system and the experience of controlling robotic manipulators. For example, the membership functions may introduce switching behaviors into the controlled system and cause chattering if k_p and k_n are too big. On the other hand, if they are too small the fuzzy adaptation might become less responsive to the change of the states of the system.

Considering the fact that the effects of $n^T e_r$ and $n^T \dot{e}_r$ on the contact force are different, a new variable s is defined as the weighted combination of $n^T \dot{e}_r$ and $n^T e_r$

$$s = n^T \dot{e}_r + \lambda n^T e_r \tag{5.18}$$

where $\lambda > 0$ is a constant. To assign a higher weights to $n^T e_r$, we should choose $\lambda > 1$.

With s being defined and letting $k_m = \lambda d_m$, the impedance model (5.11) is modified as

$$e_{fm} = m_m n^T \ddot{e}_r + d_m s \tag{5.19}$$

With the assumption that the effect of acceleration to the force error is small after assigning m_m to a small value , we can now produce another set of fuzzy rules to tune d_m as follows.

Fuzzy Rules Set 2:

- IF s is *positive* and $e_{fm} - e_{fmax}$ is *positive* THEN d_m is *small* ,
- IF s is *positive* and $e_{fm} - e_{fmin}$ is *negative* THEN d_m is *big,*
- IF s is *negative* and $e_{fm} - e_{fmax}$ is *positive* THEN d_m is *big,*
- IF s is *negative* and $e_{fm} - e_{fmin}$ is *negative* THEN d_m is *small,*
- IF $e_{fm} - e_{fmin}$ is *positive* and $e_{fm} - e_{fmax}$ is *negative* THEN d_m is *medium.*

where the linguistic variable and their membership function are the same as those in **Fuzzy Rules Set 1**.

Fuzzy Rules Set 2 has much less rules than that of **Fuzzy Rules Set 1**. By using the singleton fuzzifier, product inference engine and center average defuzzifier, the crisp d_m is derived such that

$$d_m = w_1 d_s + w_2 d_b + w_3 d_{med} \tag{5.20}$$

where

$$\begin{aligned}
w_1 &= w^{-1}[\mu_{positive}(s)\mu_{positive}(e_{fm} - e_{fmax}) + \mu_{negative}(s)\mu_{negative}(e_{fm} - e_{fmin})] \\
w_2 &= w^{-1}[\mu_{positive}(s)\mu_{negative}(e_{fm} - e_{fmin}) + \mu_{negative}(s)\mu_{positive}(e_{fm} - e_{fmax})] \\
w_3 &= w^{-1}[\mu_{positive}(e_{fm} - e_{fmin})\mu_{negative}(e_{fm} - e_{fmax})] \\
w &= [\mu_{positive}(s) + \mu_{negative}(s)][\mu_{positive}(e_{fm} - e_{fmax}) + \mu_{negative}(e_{fm} - e_{fmin})] \\
&\quad + \mu_{positive}(e_{fm} - e_{fmin})\mu_{negative}(e_{fm} - e_{fmax})
\end{aligned}$$

The controller is thus formed by combining equations (5.7) and (5.20) and is schematically sketched in Fig. 5.1.

Remark 5.2.1 *The stability of e_r and e_{fm} is also guaranteed by the controller. This is due to the fact that the right hand side of equation (5.6) remains Hurwitz for d_m and k_m obtained through the fuzzy laws.*

Fuzzy tuning of impedance model needs the knowledge of normal vector n which can be estimated from information of the measured force [54]

5.3 Simulation

The system used for simulation is schematically shown in Fig. 5.2. The end effector of the two-link manipulator moves along a circular constraint surface described by

$$(x_d - 0.8)^2 + (y + 0.4)^2 = 0.09$$

in the world coordinates XOY.

The possibility of the robot end effector's losing contact with a circular surface is much higher than on a straight surface, thus the above circular constraint is very suitable to verify the effectiveness of the unidirectional force controller to keep the end effector of the robot on the constraint.

The length, inertia and the mass of each link of the manipulator is $l_i = 0.6m$, $I_i = 0.3kgm^2$ and $m_i = 0.1kg$ respectively ($i = 1, 2$). The mass center of each link is assumed to be in the middle of the link. The joint displacements of the robot is $q = [\theta_1 \ \theta_2]^T$ and the actual position of the end effector is $r = [x \ y]^T$. The parameters of the two-link robot used are the same as those of Chapter 2 except for the link length which is made longer ($0.6m$) to cover the whole circular constraint surface.

Considering the fact that the loss of the contact is normally caused by the external disturbances, a disturbance $\tilde{f}$ with magnitude $\|\tilde{f}\| = 2$ is added to the system at $t = 2$ second for the verification of the effectiveness of the proposed approach. The disturbance last 0.06 seconds and during this period, the system dynamics becomes

$$M_r(q)\ddot{r} + C_r(q, \ \dot{q})\dot{r} + G_r(q) = J^{-T}(q)\tau + f + \tilde{f}$$

The planned trajectory of the end effector of the manipulator is specified as

$$x_d(t) \quad = \quad 0.8 - 0.3 \sin t \tag{5.21}$$

$$y_d(t) \quad = \quad -0.4 + 0.3 \cos t \tag{5.22}$$

and the desired force along the normal of the constraint surface is set to be $f_{dm} = 10N$. The maximum contact force is limited to $f_{max} = 12N$ and the minimum contact force is set to $f_{min} = 1N$.

The traditional impedance control without fuzzy adaptation is simulated first where the desired impedance parameters are fixed as $m_m = 1.2$, $d_m = 12$ and $k_m = 60$. The responses of position and force under the controller are plotted in Fig. 5.3 and Fig. 5.4 respectively. The control torques for the manipulators are given in Fig. 5.5.

For the fuzzy impedance controller, the control parameters are chosen as $m_m = 1.2$, $d_s = 5$, $d_{med} = 15$, $d_b = 25$, $\lambda = 5$ and $k_p = k_n = 1$. The responses of position and force under the controller are plotted in Fig. 5.6 and Fig. 5.7 respectively. The control torques for the manipulators and the impedance parameter are shown in Fig. 5.8 and Fig. 5.9 respectively.

From the simulation results, it can be seen that under the proposed controller, the performances of position response and force response are better than those of traditional impedance controller. The loss of contact of the end effector to the constraint surface may happen for the traditional impedance controller (negative force along the normal), whereas the contact is maintained under the proposed fuzzy impedance controller as the force is always positive along the normal. The control torques are also in the reasonable ranges.

5.4 Conclusion

The unidirectionality of the constraint force is required by the unilateral contact between the robot and the constraint, and is the assumption used in most control schemes for constrained robots. In this chapter, how to achieve the unidirectionality of the constraint force within a position/force control scheme is explicitly addressed. A fuzzy unidirectional force control scheme is presented. The controller aims at keeping the constraint force unidirectional necessary for maintaining the contact between the robot end effector and the constraint surface. A fuzzy tuning mechanism is developed to tune the control parameters. Theoretical analysis and simulation results are provided to show the effectiveness of the proposed controller.

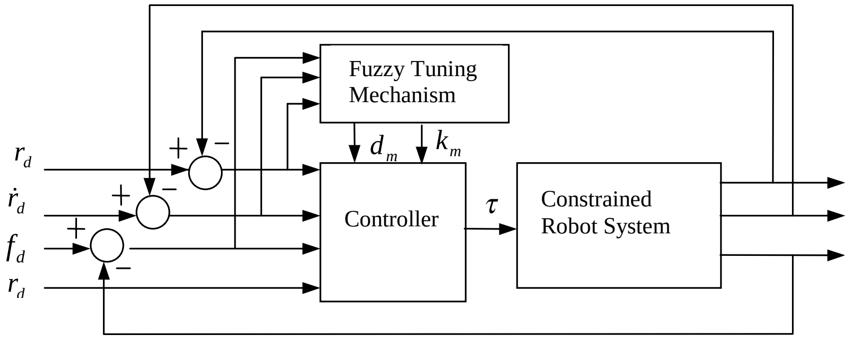

Figure 5.1: Fuzzy unidirectional force controller

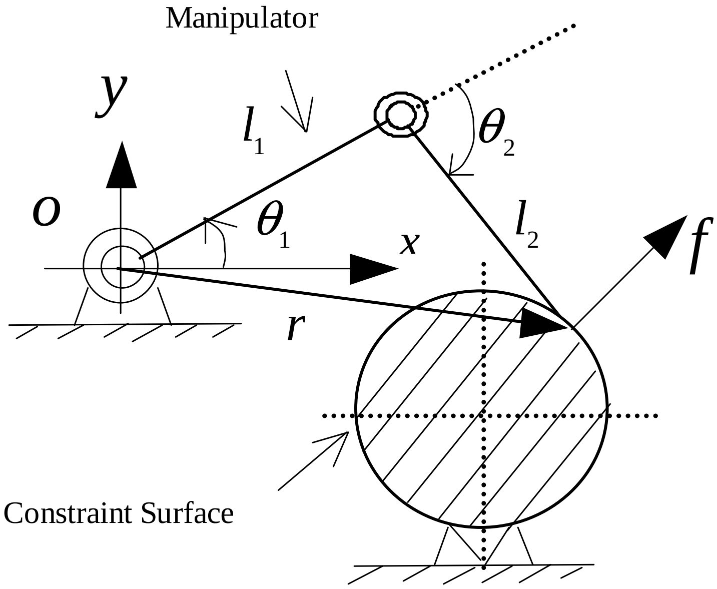

Figure 5.2: Simulation example

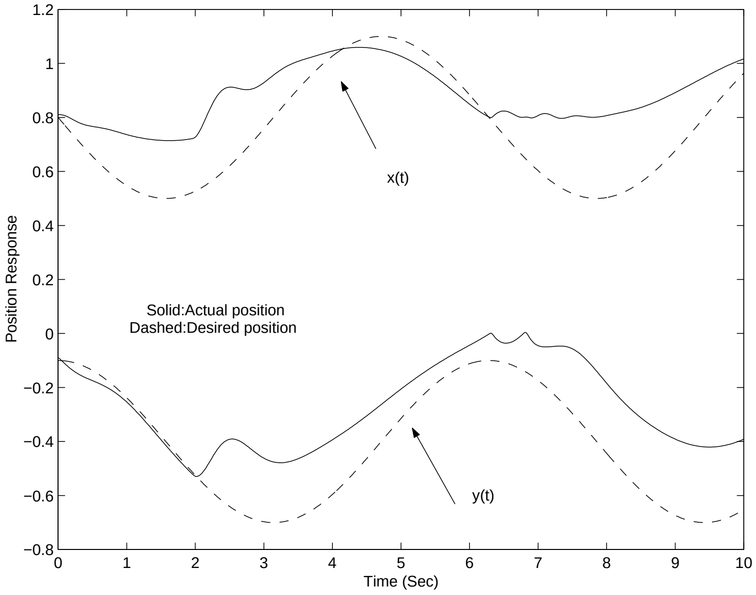

Figure 5.3: Position response without fuzzy adaption

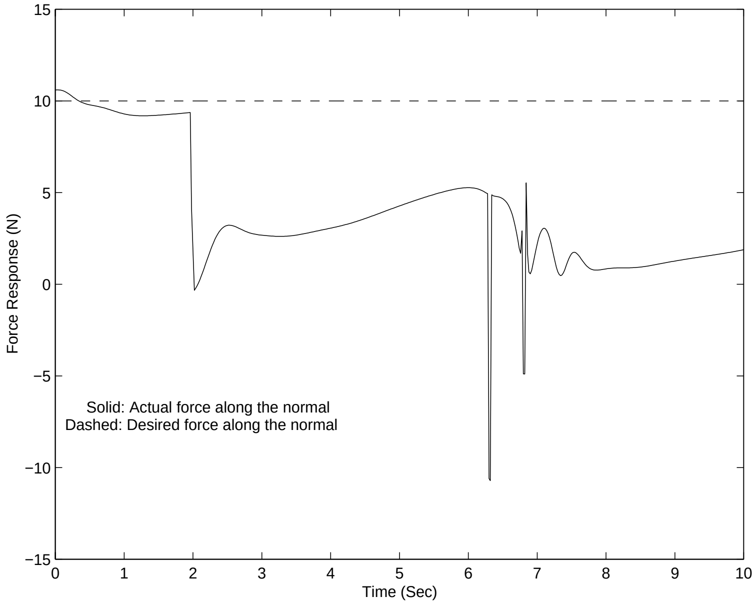

Figure 5.4: Force response without fuzzy adaptation

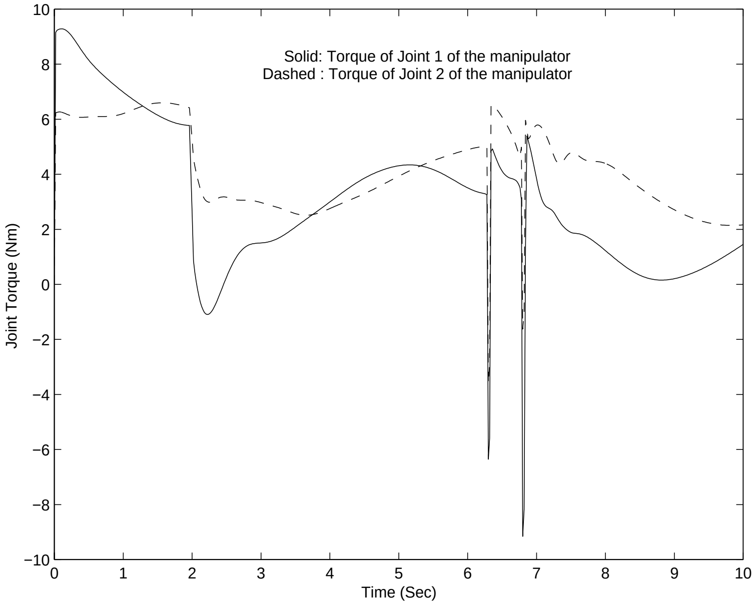

Figure 5.5: Torques of the Manipulator without fuzzy adaptation

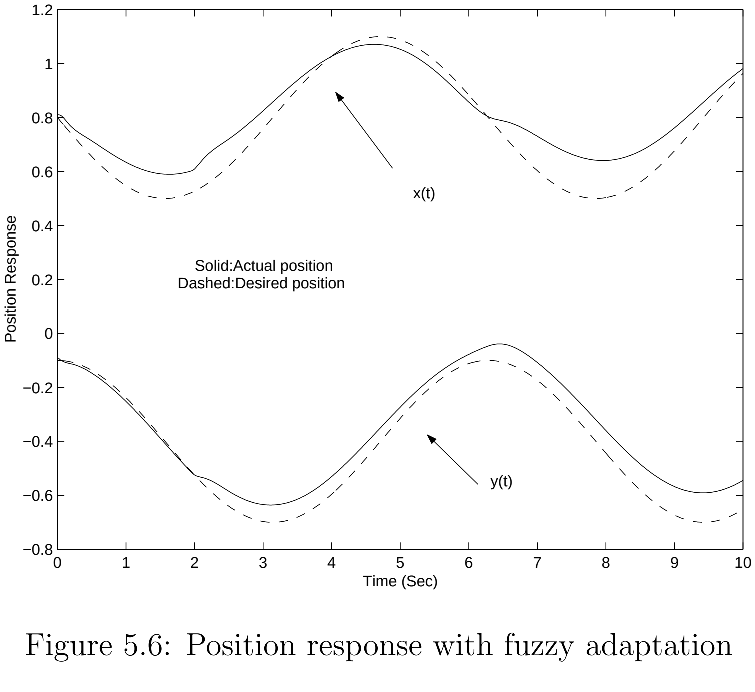

Figure 5.6: Position response with fuzzy adaptation

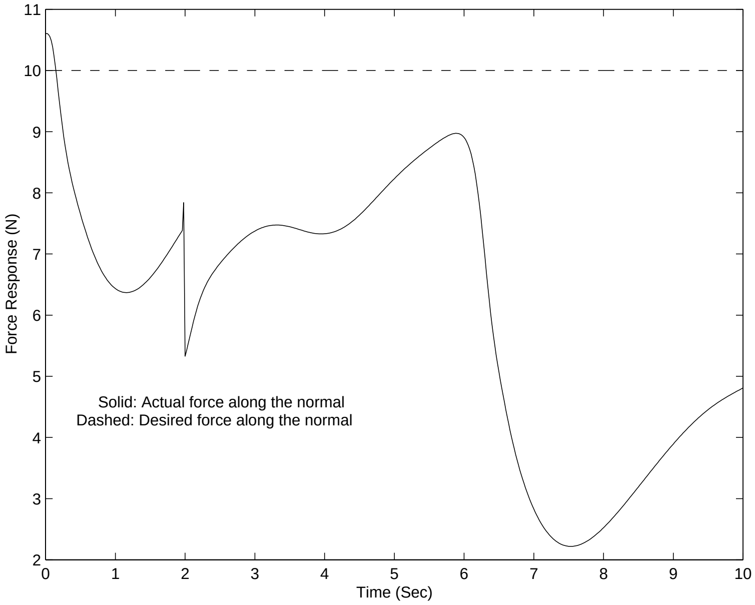

Figure 5.7: Force response with fuzzy adaptation

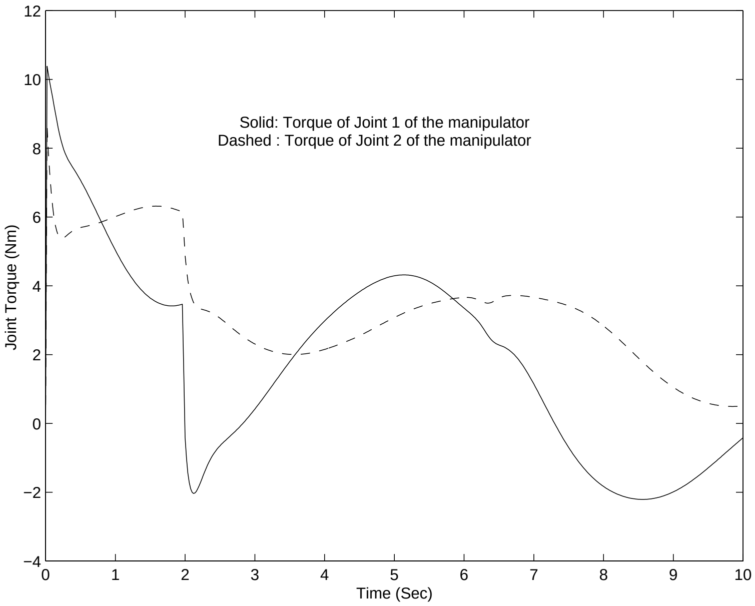

Figure 5.8: Torques of the manipulator with fuzzy adaptation

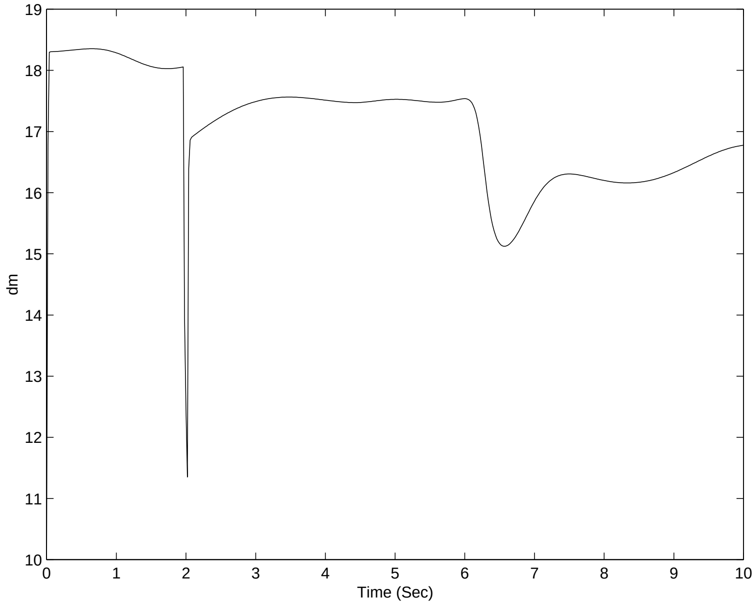

Figure 5.9: Impedance parameter *dm*

Chapter 6

Position/Force Control of Constrained Flexible Joint Robots

In this chapter, the position and force control of a constrained flexible joint robot is tackled. One of the controllers is designed with robust adaptive control approach in two steps. In the first step, a desired joint position is found to make a Lyapunov function of link position tracking error and the constraint force tracking error non-increasing. In the second step, the joint torque is derived to make the joint position to track its trajectory obtained in the first step. The controller does not have any limit on the joint flexibility. In addition, the joint stiffness and the motor inertia are assumed to be unknown as well as the robot inertia parameters. It mainly relies on the feedbacks of *joint state variables* (joint positions and velocities) and thus avoids noisy joint torque feedback. As the joint torque feedback is not used in the controller, the joint stiffness becomes an uncertain parameter scaling the control input. This new challenge to the controller design is solved with the method presented in [99].

Another controller is designed with singular perturbation approach. The fast variables and the slow variables are defined by combing the constraint force and robot's position signals. The controller is developed by combining a motor feedback controller with an exponentially stable controller for a quasi-steady-state system. It relies on the joint state feedback and achieves the robot's position tracking and the boundedness of the constraint force errors.

The chapter is organized as follows. In Section 6.1, the system dynamics and problem formulation is described. In Section 6.2, the design of a robust adaptive controller and its stability analysis are provided. Section 6.3 presents the controller designed with singular perturbation approach. In Section 6.4, simulations are done to verify the effectiveness of the controllers. The conclusion is given in Section 6.5.

6.1 Dynamical Model and Properties

Consider the dynamic model of a constrained flexible joint robot [93],

$$\begin{aligned} M(q_l)\ddot{q}_l + C(q_l, \dot{q}_l)\dot{q}_l + G(q_l) &= K_s\theta + f && (6.1)\\ J_m\ddot{q}_m + K_s\theta &= \tau_m && (6.2)\\ \theta &\triangleq q_m - q_l && (6.3) \end{aligned}$$

where $q_l \in R^n$ and $q_m \in R^n$ are the positions of the robot links and the motor shafts respectively, $M(q_l) \in R^{n\times n}$ is the inertia matrix of rigid links, $C(q_l,\ \dot{q}_l)$ is the Coriolis and centrifugal force matrix, $G(q_l)$ is the gravitational force, $J_m = diag[j_{mi}] \in R^{n\times n}$ is the positive definite diagonal matrix of the moments of inertia of the motors, $K_s = diag[k_{si}] \in R^{n\times n}$ is the positive definite diagonal matrix of the joint stiffness, $f \in R^n$ are the joint torques contributed by the constraint force and $\tau_m \in R^n$ are the input torques of the motors. j_{mi} and k_{si} $(i = 1,\ \ 2\ldots n)$ are the inertia and the stiffness of the ith joint. n is the degree of freedom of the robotic manipulator.

Assuming that the holonomic and frictionless constraint surface is described by

$$\Phi(q_l) = 0 \in R^m \qquad (6.4)$$

where $\Phi(q_l)$ is twice continuously differentiable [7].

Constraint force in the joint space, f, can then be expressed by

$$\begin{aligned} f &= J^T(q_l)\lambda && (6.5)\\ J(q_l) &\triangleq \frac{\partial\Phi(q_l)}{\partial q_l} \in R^{m\times n} \end{aligned}$$

where $\lambda \in R^m$ is a generalized Lagrange multiplier representing the magnitude of the constraint force [7]. m is the dimension of the constraint surface and it is assumed that $m < n$.

Due to the constraint, m degrees of freedom of the robot are lost. Partitioning the link position vector q_l to $q_l^1 \in R^{n-m}$ and $q_l^2 \in R^m$, we have

$$q_l = [q_l^{1T} \;\; q_l^{2T}]^T \tag{6.6}$$

and accordingly, the Jacobian $J(q_l)$ is decomposed as

$$\begin{aligned} J(q_l) &= [J_1(q_l) \;\; J_2(q_l)] \\ J_1(q_l) &\triangleq \frac{\partial \Phi(q_l)}{\partial q_l^1} \in R^{m \times n-m} \\ J_2(q_l) &\triangleq \frac{\partial \Phi(q_l)}{\partial q_l^2} \in R^{m \times m} \end{aligned} \tag{6.7}$$

As stated in [100], it is possible to have a partition such that $J_2^{-1}(q_l)$ exists and

$$\dot{q}_l = L(q_l)\dot{q}_l^1, \qquad L(q) = \begin{bmatrix} I^{(n-m)\times(n-m)} \\ -J_2^{-1}(q_l)J_1(q_l) \end{bmatrix} \tag{6.8}$$

With the partition of the link position vector in equation (6.6), the position of the robot can be uniquely determined by q_l^1. The original dynamical model in equations (6.1) and (6.2) is transformed to

$$M^1(q_l)\ddot{q}_l^1 + C^1(q_l, \dot{q}_l)\dot{q}_l^1 + G^1(q_l) = K_s\theta + J^T(q_l)\lambda \tag{6.9}$$

$$J_m\ddot{q}_m + K_s\theta = \tau_m \tag{6.10}$$

where

$$\begin{aligned} M^1(q_l) &= M(q_l)L(q_l) \in R^{n \times m} \\ C^1(q_l, \;\; \dot{q}_l) &= M(q_l)\dot{L}(q_l) + C(q_l, \;\; \dot{q}_l)L(q_l) \in R^{n \times m} \\ G^1(q_l) &= G(q_l) \in R^n \end{aligned}$$

Define $M_l(q_l) = L^T(q_l)M^1(q_l) \in R^{m \times m}$, $C_l(q_l, \;\; \dot{q}_l) = L^T(q_l)C^1(q_l, \;\; \dot{q}_l) \in R^{m \times m}$ and $G_l(q_l) = L^T(q_l)G^1(q_l) \in R^m$. It can be proved that the dynamic models (6.1) and (6.9) have the following properties.

Property 6.1 $L^T(q_l)J^T(q_l) = 0$.

Property 6.2 $M(q_l)$, $C(q_l, \dot{q}_l)$, $G(q_l)$, $M^1(q_l)$, $C^1(q_l, \dot{q}_l)$, $G^1(q_l)$, $M_l(q_l)$, $C_l(q_l, \dot{q}_l)$, $G_l(q_l)$, $L(q_l)$ $\dot{L}(q_l)$, and $J(q_l)$ are uniformly bounded and continuous if q_l and $\dot{q}_l$ are uniformly bounded and continuous; $M(q_l)$ and $M_l(q_l)$ are symmetric positive definite ($s.p.d$).

Property 6.3 $\dot{M}(q_l) - 2C(q_l,\ \dot{q}_l)$ and $\dot{M}_L(q_l) - 2C_L(q_l,\ \dot{q}_l)$ are skew-symmetric if $C(q_l,\ \dot{q}_l)$ is in the Christoffel form, i.e., $x_1^T(\dot{M}(q_l) - 2C(q_l,\ \dot{q}_l))x_1 = 0$, $x_2^T(\dot{M}_L(q_l) - 2C_L(q_l,\ \dot{q}_l))x_2 = 0$, $\forall x_1 \in R^n$ and $x_2 \in R^{n-m}$.

Property 6.4 The robot link dynamics described by equation (6.1) is linear in the robot link parameters, i.e. given an arbitrary vector $\chi \in R^n$

$$M(q_l)\dot{\chi} + C(q_l,\ \dot{q}_l)\chi + G(q_l) = \Psi(\dot{\chi}, \chi, \dot{q}_l, q_l)p \tag{6.11}$$

where $p \in R^l$ is a vector of the lumped parameters of interest, $\Psi(\dot{\chi}, \chi, \dot{q}_l, q_l) \in R^{n\times l}$ is a regressor matrix.

If the estimates of $M(q_l)$, $C(q_l,\ \dot{q}_l)$ and $G(q_l)$ are denoted by $\hat{M}(q_l)$, $\hat{C}(q_l,\ \dot{q}_l)$ and $\hat{G}(q_l)$, then

$$\hat{M}(q_l)\dot{\chi} + \hat{C}(q_l,\ \dot{q}_l)\chi + \hat{G}(q_l) = \Psi(\dot{\chi}, \chi, \dot{q}_l, q_l)\hat{p} \tag{6.12}$$

where $\hat{p}$ is the estimate of p.

Property 6.5 [99] If the regressor matrix $\Psi(\dot{\chi}, \chi, \dot{q}_l, q_l)$ and the vector p in equation (6.11) are given in the following forms

$$\begin{aligned}\Psi(\dot{\chi}, \chi, \dot{q}_l, q_l) &= \begin{bmatrix} \psi_1^T(\dot{\chi}, \chi, \dot{q}_l, q_l) & 0 & ... & 0 \\ 0 & \psi_2^T(\dot{\chi}, \chi, \dot{q}_l, q_l) & ... & 0 \\ ... & ... & ... & .... \\ 0 & 0 & ... & \psi_n^T(\dot{\chi}, \chi, \dot{q}_l, q_l) \end{bmatrix} \in R^{n\times l} \\ p &= [p_1^T\ \ p_2^T\ \ ...p_n^T]^T \in R^l\end{aligned}$$

then

$$K_s\Psi(\dot{\chi}, \chi, \dot{q}_l, q_l)p = \Psi(\dot{\chi}, \chi, \dot{q}_l, q_l)p_s$$

where

$$\begin{aligned} p_s &\triangleq [k_{s1}p_1^T \;\; k_{s2}p_2^T \;\; \ldots k_{sn}p_n^T]^T = \Lambda p \in R^l \\ \Lambda &\triangleq \begin{bmatrix} diag[k_{s1}] & 0 & ... & 0 \\ 0 & diag[k_{s2}] & ... & 0 \\ ... & ... & ... & ... \\ 0 & 0 & ... & diag[k_{sn}] \end{bmatrix} \in R^{l\times l} \end{aligned}$$

and $\psi_i(\dot{\chi}, \chi, \dot{q}_l, q_l) \in R^{n_i}$, $p_i \in R^{n_i}$, $diag[k_{si}] \in R^{n_i \times n_i}$, $\sum_{i=1}^{n} n_i = l$, $i = 1, \;\; 2, \ldots n$.

Assume that K_s and J_m are unknown and their estimates are denoted as $\hat{K}_s$, and $\hat{J}_m$ respectively. Their estimate errors are defined as $\tilde{K}_s \triangleq K_s - \hat{K}_s$ and $\tilde{J}_m \triangleq J_m - \hat{J}_m$ respectively. For the controller design, the following assumptions are made for these terms:

Assumption 6.1 K_s and J_m are unknown and bounded.

Assumption 6.2 $\tilde{K}_s$ and $\tilde{J}_m$ are bounded such that $\|\tilde{K}_s\| \leq \delta_K$ and $\|\tilde{J}_m\| \leq \delta_J$, where δ_K and δ_J are known positive constants.

Assumption 6.3 $q_l(t)$, $q_m(t)$, $\dot{q}_l(t)$, $\dot{q}_m(t)$, $\ddot{q}_m$ and $\lambda(t)$ are all measurable.

Assumption 6.4 The desired link position $(q_{ld}(t))$ and the constraint force $(\lambda_d(t))$ and their derivatives are bounded and continuously differentiable.

6.2 Robust and Adaptive Control Design

Let $q_{ld}(t)$ be the desired trajectory of the link position and $\lambda_d(t)$ be the desired magnitude of the constraint force. The control objective is to find a driving torque τ_m under which $q_l(t)$ tracks $q_{ld}(t)$ and the error between $\lambda(t)$ and $\lambda_d(t)$ is bounded. It is only necessary to make $q_l^1(t)$ track its desired trajectory $q_{ld}^1(t)$ since $q_l^1(t)$ completely determines $q_l(t)$.

From dynamic model represented by equations (6.9) and (6.10), the controller is designed in two steps. In the first step, a desired value of θ: θ_d, is determined to make $q_l^1(t)$ track $q_{ld}^1(t)$ and $\lambda_d(t) - \lambda(t)$ is bounded. Then, the control input τ_m is obtained to make θ to track θ_d in the second step.

Define the following variables related to the tracking errors of the link position and the constraint force:

$$e^1 \triangleq q_{ld}^1 - q_l^1, \qquad e_\lambda \triangleq \lambda_d - \lambda \tag{6.13}$$

$$r^1 \triangleq \dot{e}^1 + K_e e^1, \qquad \dot{q}_r^1 \triangleq \dot{q}_{ld}^1 + K_e e^1 \tag{6.14}$$

where $K_e \in R^{(n-m)\times(n-m)}$ is a positive definite constant matrix.

With variables r^1 and q_r^1, the following variables are also defined:

$$\sigma \triangleq L(q_l)r^1 + \mu \tag{6.15}$$

$$\nu \triangleq L(q_l)\dot{q}_r^1 + \mu \tag{6.16}$$

where μ is a variable introduced to compensate the force error e_λ and it will be determined later.

From equations (6.15) and (6.16), it is obvious that

$$r^1 = \dot{q}_r^1 - \dot{q}_l^1 \tag{6.17}$$

$$\sigma = \nu - L(q_l)\dot{q}_l^1 \tag{6.18}$$

$$\dot{\sigma} = \dot{\nu} - L(q_l)\ddot{q}_l^1 - \dot{L}(q_l)\dot{q}_l^1 \tag{6.19}$$

As a preparation for designing the system parameters updating law, the following vectors are also constructed with the elements of diagonal matrices K_s, $\hat{K}_s$ and $\hat{K}_s^{-1}$.

$$k_s^{-1} \triangleq [k_{s1}^{-1} \ \ k_{s2}^{-1} \ldots k_{sn}^{-1}]^T \tag{6.20}$$

$$\hat{k}_s^{-1} \triangleq [\hat{k}_{s1}^{-1} \ \ \hat{k}_{s2}^{-1} \ldots \hat{k}_{sn}^{-1}]^T \tag{6.21}$$

$$\tilde{k}_s^{-1} = k_s^{-1} - \hat{k}_s^{-1} \tag{6.22}$$

$$\varphi_1(K_s, \hat{K}_s^{-1}) \triangleq K_s\hat{k}_s^{-1} = [k_{s1}\tilde{k}_{s1}^{-1} \ \ k_{s2}\tilde{k}_{s2}^{-1} \ \ \ldots k_{sn}\tilde{k}_{sn}^{-1}]^T \tag{6.23}$$

where k_{si} is the stiffness of ith joint and $\hat{k}_{si}^{-1}$ is the estimate of k_{si}^{-1}.

Note that $\hat{K}_s^{-1}$ and $\hat{k}_s^{-1}$ here only serve as the estimates of K_s^{-1} and k_s^{-1} respectively. They are NOT the inverses of $\hat{K}_s$ and $\hat{k}_s$ respectively.

Step 1. Determination of θ_d: the desired value of θ

In this step, θ_d, the desired value of θ, is obtained to make a Lyapunov function of e^1, e_λ and the system parameter estimation errors nonincreasing.

Consider the following Lyapunov function

$$V_1 = \frac{1}{2}\sigma^T M(q_l)\sigma + \frac{1}{2}(p-\hat{p}_s)^T\Gamma_1(p-\hat{p}_s) + \frac{1}{2}\mu^T\mu + \frac{1}{2}\varphi_1^T(K_s,\hat{K}_s^{-1})\Gamma_2\varphi_1(K_s,\hat{K}_s^{-1}) \tag{6.24}$$

where $\hat{p}_s$ is the estimate of p_s defined in **Property 6.5**, $\Gamma_1 \in R^{l\times l}$ and $\Gamma_2 \in R^{n\times n}$ are positive definite diagonal matrices.

Differentiating V_1 with respect to time t and considering **Property 6.3** and **Property 6.5**, we have

$$\begin{aligned}\dot{V}_1 &= \sigma^T M(q_l)\dot{\sigma} + \frac{1}{2}\sigma^T\dot{M}(q_l)\sigma - (p-\hat{p}_s)^T\Gamma_1\dot{\hat{p}}_s + \mu^T\dot{\mu} + \varphi_1^T(K_s,\hat{K}_s^{-1})\Gamma_2\dot{\varphi}_1(K_s,\hat{K}_s^{-1})\\ &= \sigma^T(M(q_l)\dot{\sigma} + C(q_l,\dot{q}_l)\sigma) - (p-\hat{p}_s)^T\Gamma_1\dot{\hat{p}}_s\\ &+\mu^T\dot{\mu} + \varphi_1^T(K_s,\hat{K}_s^{-1})\Gamma_2\dot{\varphi}_1(K_s,\hat{K}_s^{-1})\end{aligned} \tag{6.25}$$

Substituting σ in equation (6.18) and $\dot{\sigma}$ in equation (6.19) into equation (6.25), we have

$$\begin{aligned}\dot{V}_1 &= \sigma^T(M(q_l)\dot{\nu} + C(q_l,\dot{q}_l)\nu + G(q_l) - M^1(q_l)\ddot{q}_l^1 - C^1(q_l,\dot{q}_l)\dot{q}_l^1 - G^1(q_l))\\ &\quad -(p-\hat{p}_s)^T\Gamma_1\dot{\hat{p}}_s + \mu^T\dot{\mu} + \varphi_1^T(K_s,\hat{K}_s^{-1})\Gamma_2\dot{\varphi}_1(K_s,\hat{K}_s^{-1})\end{aligned} \tag{6.26}$$

Considering equation (6.9) and **Property 6.4**, equation (6.26) becomes

$$\begin{aligned}\dot{V}_1 &= \sigma^T(\Psi(\dot{\nu},\nu,\dot{q}_l,q_l)p - K_s\theta - J^T(q_l)\lambda)) - (p-\hat{p}_s)^T\Gamma_1\dot{\hat{p}}_s\\ &+ \mu^T\dot{\mu} + \varphi_1^T(K_s,\hat{K}_s^{-1})\Gamma_2\dot{\varphi}_1(K_s,\hat{K}_s^{-1})\end{aligned} \tag{6.27}$$

Our aim is to find θ_d to make $\dot{V}_1$ non-positive. Define the difference between θ and θ_d as

$$z \triangleq \theta_d - \theta$$

Equation (6.27) is re-written as

$$\begin{aligned}\dot{V}_1 &= \sigma^T(\Psi(\dot{\nu},\nu,\dot{q}_l,q_l)p - K_s\theta_d - J^T(q_l)\lambda)) - (p-\hat{p}_s)^T\Gamma_1\dot{\hat{p}} + \mu^T\dot{\mu} \\ &\quad +\varphi_1^T(K_s,\hat{K}_s^{-1})\Gamma_2\dot{\varphi}_1(K_s,\hat{K}_s^{-1}) + \sigma^T K_s z \qquad (6.28)\end{aligned}$$

Letting

$$\theta_d = K_\sigma\sigma - \hat{K}_s^{-1}J^T(q_l)\lambda_d + \Psi(\dot{\nu},\nu,\dot{q}_l,q_l)\hat{p} \qquad (6.29)$$

where $K_\sigma \in R^{n\times n}$ is positive definite and $\hat{p}$ is the estimate of p, and substituting it into equation (6.28), we have

$$\begin{aligned}\dot{V}_1 &= \sigma^T(\Psi(\dot{\nu},\nu,\dot{q}_l,q_l)p - K_s\Psi(\dot{\nu},\nu,\dot{q}_l,q_l)\hat{p}) - \sigma^T K_s K_\sigma\sigma \\ &+ \sigma^T K_s\hat{K}_s^{-1}J^T(q_l)\lambda_d - \sigma^T J^T(q_l)\lambda - (p-\hat{p}_s)^T\Gamma_1\dot{\hat{p}}_s + \mu^T\dot{\mu} \\ &+ \varphi_1^T(K_s,\hat{K}_s^{-1})\Gamma_2\dot{\varphi}_1(K_s,\hat{K}_s^{-1}) + \sigma^T K_s z \qquad (6.30)\end{aligned}$$

From the definition of σ in equation (6.15) and **Property 6.1**, it is obvious that

$$\begin{aligned}\sigma^T J^T(q_l)\lambda &= \mu^T J^T(q_l)\lambda \\ \sigma^T K_s\hat{K}_s^{-1}J^T(q_l)\lambda_d &= \mu^T J^T(q_l)\lambda_d - \sigma^T K_s\tilde{K}_s^{-1}J^T(q_l)\lambda_d\end{aligned}$$

Substituting the above terms into equation (6.30) and considering **Property 6.5**, we have

$$\begin{aligned}&\dot{V}_1 = (p-\hat{p}_s)^T(\Psi^T(\dot{\nu},\nu,\dot{q}_l,q_l)\sigma - \Gamma_1\dot{\hat{p}}_s) - \sigma^T K_s K_\sigma\sigma + \mu^T J^T(q_l)e_\lambda \\ &-\sigma^T K_s\tilde{K}_s^{-1}J^T(q_l)\lambda_d + \mu^T\dot{\mu} + \varphi_1^T(K_s,\hat{K}_s^{-1})\Gamma_2\dot{\varphi}_1(K_s,\hat{K}_s^{-1}) + \sigma^T K_s z(6.31)\end{aligned}$$

Letting μ evolve according to

$$\dot{\mu} + K_\mu\mu = -J^T(q_l)e_\lambda \qquad (6.32)$$

where $K_u \in R^{n\times n}$ is a positive definite constant matrix, and substituting it into equation (6.31), we have

$$\begin{aligned}\dot{V}_1 &= (p-\hat{p}_s)^T(\Psi^T(\dot{\nu},\nu,\dot{q}_l,q_l)\sigma - \Gamma_1\dot{\hat{p}}_s) - \sigma^T K_s\tilde{K}_s^{-1}J^T(q_l)\lambda_d \\ &\quad -\sigma^T K_s K_\sigma\sigma - \mu^T K_\mu\mu + \varphi_1^T(K_s,\hat{K}_s^{-1})\Gamma_2\dot{\varphi}_1(K_s,\hat{K}_s^{-1}) + \sigma^T K_s z \ (6.33)\end{aligned}$$

As both K_s and $\tilde{K}_s^{-1}$ are diagonal matrices, it is easy to verify that

$$\sigma^T K_s\tilde{K}_s^{-1}J^T(q_l)\lambda_d = \varphi_1^T(K_s,\hat{K}_s^{-1})\varphi_2(\sigma,q_l,\lambda_d) \qquad (6.34)$$

where

$$\varphi_2(\sigma, q_l, \lambda_d) \triangleq [(J^T(q_l)\lambda_d)_1\sigma_1 \quad (J^T(q_l)\lambda_d)_2\sigma_2 \ldots (J^T(q_l)\lambda_d)_n\sigma_n]^T \tag{6.35}$$

and $(J^T(q_l)\lambda_d)_i$ and σ_i are the ith elements of $J^T(q_l)\lambda_d$ and σ respectively.

Substituting equation (6.34) into equation (6.33), we have

$$\begin{aligned} \dot{V}_1 \;=\;& (p - \hat{p}_s)^T(\Psi^T(\dot{\nu}, \nu, \dot{q}_l, q_l)\sigma - \Gamma_1\dot{\hat{p}}_s) - \varphi_1^T(K_s, \hat{K}_s^{-1})(\varphi_2(\sigma, q_l, \lambda_d) \\ & -\Gamma_2\dot{\varphi}_1(K_s, \hat{K}_s^{-1})) + \sigma^T K_s z - \mu^T K_\mu \mu - \sigma^T K_s K_\sigma \sigma \end{aligned} \tag{6.36}$$

Letting

$$\dot{\hat{p}}_s = \Gamma_1^{-1}\Psi^T(\dot{\nu}, \nu, \dot{q}_l, q_l)\sigma \tag{6.37}$$

$$\dot{\varphi}_1(K_s, \hat{K}_s^{-1}) = \Gamma_2^{-1}\varphi_2(\sigma, q_l, \lambda_d) \tag{6.38}$$

and substituting them into equation (6.36), it follows that

$$\dot{V}_1 = \sigma^T K_s z - \mu^T K_\mu \mu - \sigma^T K_s K_\sigma \sigma \tag{6.39}$$

As K_μ and $K_s K_\sigma$ are all positive definite, it follows that

$$\dot{V}_1 \leq \sigma^T K_s z \leq 0 \tag{6.40}$$

when $z = 0$, that is, $\theta = \theta_d$.

From equation (6.37) and the definition of p_s in **Property 6.5**, we have

$$\dot{\hat{p}} = \Gamma_p \Psi^T(\dot{\nu}, \nu, \dot{q}_l, q_l)\sigma \tag{6.41}$$

where $\Gamma_p \triangleq \Lambda^{-1}\Gamma_1^{-1}$.

From equation (6.38) and the definition of $\varphi_1(K_s, \hat{K}_s^{-1})$ in equation (6.23), we have

$$\dot{\hat{k}}_s^{-1} = -\Gamma_k \varphi_2(\sigma, q_l, \lambda_d) \tag{6.42}$$

where $\Gamma_k \triangleq K_s^{-1}\Gamma_2^{-1}$.

Once the elements of $\hat{k}_s^{-1}$ are determined through the adaptation law in equation (6.42), they are then used to form the diagonal matrix $\hat{K}_s^{-1}$ for the calculation of θ_d in equation (6.29).

Remark 6.2.1 *The uncertain terms Λ and K_s are "absorbed" in the parameter adaptation gains Γ_p and Γ_k respectively. Γ_p and Γ_k are always positive definite due to the fact that Λ, K_s , Γ_1 and Γ_2 are positive definite.*

Step 2. Determination of Input Torque: τ_m

In this step, the control input τ_m will be derived to make a Lyapunov function of z, e^1 and e_λ non-increasing.

Defining a Lyapunov function

$$\begin{aligned} V_2 &= V_1 + \frac{1}{2}z^T K_s z = \frac{1}{2}\sigma^T M(q_l)\sigma + \frac{1}{2}(p-\hat{p}_s)^T\Gamma_1(p-\hat{p}_s) \\ &\quad +\frac{1}{2}\mu^T\mu + \varphi_1^T(K_s,\hat{K}_s^{-1})\Gamma_2\varphi_1(K_s,\hat{K}_s^{-1}) + \frac{1}{2}z^T K_s z \end{aligned} \tag{6.43}$$

and differentiating it with respect to time t, we have

$$\dot{V}_2 = \dot{V}_1 + \dot{z}^T K_s z \tag{6.44}$$

With θ_d given in equation (6.29) and $\hat{p}$ and $\hat{k}_s^{-1}$ adaptively tuned in equations (6.41) and (6.42) respectively, it has been shown that

$$\dot{V}_1 = \sigma^T K_s z - \mu^T K_\mu \mu - \sigma^T K_s K_\sigma \sigma \tag{6.45}$$

and, as a result,

$$\begin{aligned} \dot{V}_2 &= (\sigma+\dot{z})^T K_s z - \mu^T K_\mu\mu - \sigma^T K_s K_\sigma\sigma \\ &= (\sigma+\dot{z})^T(K_s\theta_d - K_s\theta) - \mu^T K_\mu\mu - \sigma^T K_s K_\sigma\sigma \end{aligned} \tag{6.46}$$

Solving $K_s\theta$ in equations (6.2) and substituting it into equation (6.46), we have

$$\dot{V}_2 = (\sigma+\dot{z})^T(K_s\theta_d - \tau_m + J_m\ddot{q}_m) - \mu^T K_\mu\mu - \sigma^T K_s K_\sigma\sigma \tag{6.47}$$

Letting

$$\tau_m = \hat{K}_s\theta_d + \hat{J}_m\ddot{q}_m + k_\tau sgn(\sigma+\dot{z}) \tag{6.48}$$

$$k_\tau \geq \delta_K\|\theta_d\| + \delta_J\|\ddot{q}_m\| \tag{6.49}$$

where $\hat{K}_s$ and $\hat{J}_m$ are the estimates of K_s and J_m respectively, and δ_K and δ_J are the bounds of $\tilde{K}_s$ and $\tilde{J}_m$ respectively. $sgn(\sigma + \dot{z})$ is a sign function applying on $\sigma + \dot{z}$ element wise such that 1 is returned when an element of $\sigma + \dot{z}$ is non-negative or -1 otherwise.

Substituting τ_m into equation (6.47), it turns out that

$$\dot{V}_2 = (\sigma + \dot{z})^T (\tilde{K}_s \theta_d + \tilde{J}_m \ddot{q}_m - k_\tau sgn(\sigma + \dot{z})) - \mu^T K_\mu \mu - \sigma^T K_s K_\sigma \sigma \tag{6.50}$$

Noting the definition of k_τ, we have

$$(\sigma + \dot{z})^T (\tilde{K}_s \theta_d + \tilde{J}_m \ddot{q}_m - k_\tau sgn(\sigma + \dot{z})) \leq 0 \tag{6.51}$$

and

$$\dot{V}_2 \leq -\mu^T K_\mu \mu - \sigma^T K_s K_\sigma \sigma \leq 0 \tag{6.52}$$

As $\dot{V}_2 \leq 0$, V_2 is non-increasing. The uniform boundedness of σ, $p - \hat{p}_s$, μ, $\varphi_1(K_s, \hat{K}_s^{-1})$ and z are guaranteed. From the definition of σ in equation (6.15) and **Property 6.2**, r^1 is also uniformly bounded. From the definition of r^1, the uniform boundedness of r^1 guarantees the uniform boundedness of e^1 and $\dot{e}^1$.

It is thus concluded that $e^1 \rightarrow 0$ or $q_l^1 \rightarrow q_{ld}^1$. As q_l is uniquely determined by q_l^1 , it can be concluded that $q_l \rightarrow q_{ld}$.

To analyze the force tracking, $\Psi(\dot{\nu}, \nu, \dot{q}_l, q_l)\hat{p}$ is expanded from **Property 6.4**,

$$\Psi(\dot{\nu}, \nu, \dot{q}_l, q_l)\hat{p} = \hat{M}(q_l)\dot{\nu} + \hat{C}(q_l, \dot{q}_l)\nu + \hat{G}(q_l) \tag{6.53}$$

Substituting ν in equation (6.16) into equation (6.53) and considering equation (6.32), we have

$$\begin{aligned} &\Psi(\dot{\nu}, \nu, \dot{q}_l, q_l)\hat{p} \\ &= \hat{M}^1(q_l)\ddot{q}_r^1 + \hat{C}^1(q_l, \dot{q}_l)\dot{q}_r^1 + \hat{G}^1(q_l) - (\hat{M}(q_l)K_\mu - \hat{C}(q_l, \dot{q}_l))\mu - \hat{M}(q_l)J^T(q_l)e_\lambda \end{aligned} \tag{6.54}$$

and θ_d in equation (6.29) is re-written as

$$\begin{aligned} \theta_d &= K_\sigma \sigma - \hat{K}_s^{-1} J^T(q_l)\lambda_d + \hat{M}^1(q_l)\ddot{q}_r^1 + \hat{C}^1(q_l, \dot{q}_l)\dot{q}_r^1 + \hat{G}^1(q_l) \\ &- (\hat{M}(q_l)K_\mu - \hat{C}(q_l, \dot{q}_l))\mu - \hat{M}(q_l)J^T(q_l)e_\lambda \end{aligned} \tag{6.55}$$

Noting that $\theta = \theta_d - z$ and substituting it into equation (6.10), the following closed-loop system dynamics is obtained

$$(K_s\hat{M}(q_l) + I^{(n-m)\times(n-m)})e_\lambda = M^1(q_l)\dot{r}^1 + C^1(q_l,\dot{q}_l)r^1 + K_s\tilde{K}_s^{-1}J^T(q_l)\lambda_d - K_s z - \beta \tag{6.56}$$

where

$$\beta \triangleq (M(q_l) - K_s\hat{M}^1(q_l))\ddot{q}_r^1 + (C(q_l,\dot{q}_l) - K_s\hat{C}^1(q_l,\dot{q}_l))\dot{q}_r^1 + G(q_l) - K_s\hat{G}^1(q_l)$$

and $\tilde{K}_s^{-1} = K_s^{-1} - \hat{K}_s^{-1}$.

Multiplying both sides of equation (6.56) by $J(q_l)M^{-1}(q_l)$ and noting that $J(q_l)M(q_l)M^1(q_l) = J(q_l)L(q_l) = 0$, we have

$$\begin{aligned} & J(q_l)M^{-1}(q_l)(K_s\hat{M}(q_l) + I^{(n-m)\times(n-m)})e_\lambda \\ &= J(q_l)M^{-1}(q_l)(C^1(q_l,\dot{q}_l)r^1 + K_s\tilde{K}_s^{-1}J^T(q_l)\lambda_d - K_s z - \beta) \end{aligned} \tag{6.57}$$

We can conclude that β is bounded as $\ddot{q}_r^1 = \dot{q}_{ld}^1 + K_e e^1$ and the other terms in equation (6.2) are bounded.

With β and other terms in equation (6.57) being bounded and the term $J(q_l)M^{-1}(q_l)(K_s\hat{M}(q_l)+I^{(n-m)\times(n-m)})$ being non-singular, it is concluded that e_λ is bounded from equation (6.56).

Summarizing what discussed in **Step 1** and **Step 2**, we have the following theorem.

Theorem 6.2.1 *For a constrained flexible joint robotic manipulator modeled by equations (6.1) and (6.2), the robot's position q_l converges to its desired value q_{ld} and the force tracking error $\lambda_d - \lambda$ is uniformly bounded if*

$$\tau_m = \hat{K}_s\theta_d + \hat{J}_m\ddot{q}_m + k_\tau sgn(\sigma + \dot{z}) \tag{6.58}$$

$$k_\tau \geq \delta_K\|\theta_d\| + \delta_J\|\ddot{q}_m\| \tag{6.59}$$

where

$$\begin{aligned} z &= \theta_d - \theta \\ \theta_d &= K_\sigma\sigma - \hat{K}_s^{-1}J^T(q_l)\lambda_d + \Psi(\dot{\nu},\nu,\dot{q}_l,q_l)\hat{p} \end{aligned}$$

$$\begin{aligned}
\theta &= q_m - q_l \\
\dot{\hat{p}} &= \Gamma_p \Psi^T(\dot{\nu}, \nu, \dot{q}_l, q_l)\sigma \\
\dot{\hat{k}}_s^{-1} &= -\Gamma_k \varphi_2(\sigma, q_l, \lambda_d) \\
\hat{k}_s^{-1} &= [k_{s1}^{-1}\ k_{s2}^{-1}\ \ldots k_{sn}^{-1}]^T \\
\hat{K}_s^{-1} &= diag[k_{si}^{-1}],\ i = 1,\ 2, \ldots n \\
\sigma &= L(q_l)r^1 + \mu \\
\nu &= L(q_l)\dot{q}_r^1 + \mu \\
\dot{\mu} &+ K_\mu \mu = -J^T(q_l)e_\lambda \\
r^1 &= \dot{e}^1 + K_e e^1 \\
e^1 &= q_{ld}^1 - q_l^1 \\
e_\lambda &= \lambda_d - \lambda \\
\dot{q}_r^1 &= \dot{q}_{ld}^1 + K_e e^1
\end{aligned}$$

$K_\sigma > 0$, $\Gamma_p > 0$ and $\Gamma_k > 0$ are control parameters, $\hat{K}_s$ and $\hat{J}_m$ are the estimates of K_s and J_m respectively. Matrices $\Psi(\dot{\nu}, \nu, \dot{q}_l, q_l)$, δ_K and δ_J are defined in **Property 6.4** *and* **Assumption 6.2** *respectively, and $\varphi_2(\sigma, q_l, \lambda_d)$ is defined in equation (6.35). K_σ, K_e, K_μ, Γ_p and Γ_k are all positive definite.*

Remark 6.2.2 *To avoid the calculation of $\dot{z}$ in equation (6.58), a new variable s_z is defined such that*

$$s_z = \int_0^t \sigma dt + z \tag{6.60}$$

Obviously

$$sgn(\sigma + \dot{z}) = sgn(\dot{s}_z) \tag{6.61}$$

By comparing the value of s_z in consecutive sampling times, the sign of $\dot{s_z}$ or $\sigma + \dot{z}$ can be obtained. If s_z is non-decreasing, the sign should be positive, or negative otherwise.

Remark 6.2.3 *It is well known that the sign function in the controller causes chattering which can be eliminated with the boundary layer approach [58]. In this approach, the sign function $sgn(s)$ is replaced by s/Δ when $\|s\| < \Delta$. $\Delta > 0$ is defined as a boundary layer.*

Remark 6.2.4 *When the joint stiffness reaches a sufficiently large value, the control system becomes sensitive to the variations of the signals in the control loops. This will cause large fluctuations even divergency of the control variables. The singular perturbation based controller in [64] for free flexible joint robot can be extended to solve this problem and it will be presented in the next section.*

6.2.1 Controller Design – Singular Perturbation Approach

In this section, a singular perturbation based position/force controller for constrained robot is developed. It is an extension of the work on singular perturbation based controller for free flexible joint robot in [64].

For the controller design, the desired motor position: q_{md}, velocity $\dot{q}_{md}$ and acceleration $\ddot{q}_{md}$ are needed. From equation (6.1) and **Property 6.4**, they can be calculated by

$$q_{md} = K_s^{-1}(\Psi(\ddot{q}_{ld}, \dot{q}_{ld}, \dot{q}_{ld}, q_{ld})p - f_d) + q_{ld} \tag{6.62}$$

$$\dot{q}_{md} = K_s^{-1}(\dot{\Psi}(\ddot{q}_{ld}, \dot{q}_{ld}, \dot{q}_{ld}, q_{ld})p - \dot{f}_d) + \dot{q}_{ld} \tag{6.63}$$

$$\ddot{q}_{md} = K_s^{-1}(\ddot{\Psi}(\ddot{q}_{ld}, \dot{q}_{ld}, \dot{q}_{ld}, q_{ld})p - \ddot{f}_d) + \ddot{q}_{ld} \tag{6.64}$$

where $f_d = J^T(q_{ld})\lambda_d$. From **Assumption 6.4**, q_{md}, $\dot{q}_{md}$ and $\ddot{q}_{md}$ are bounded and continuous.

As p is unknown, it is impossible to obtain the exact values of q_{md}, $\dot{q}_{md}$ and $\ddot{q}_{md}$. Their estimates $\hat{q}_{md}$, $\dot{\hat{q}}_{md}$ and $\ddot{\hat{q}}_{md}$ are obtained through

$$\hat{q}_{md} = K_s^{-1}(\Psi(\ddot{q}_{ld}, \dot{q}_{ld}, \dot{q}_{ld}, q_{ld})\hat{p} - f_d) + q_{ld} \tag{6.65}$$

$$\dot{\hat{q}}_{md} = K_s^{-1}(\dot{\Psi}(\ddot{q}_{ld}, \dot{q}_{ld}, \dot{q}_{ld}, q_{ld})\hat{p} - \dot{f}_d) + \dot{q}_{ld} \tag{6.66}$$

$$\ddot{\hat{q}}_{md} = K_s^{-1}(\ddot{\Psi}(\ddot{q}_{ld}, \dot{q}_{ld}, \dot{q}_{ld}, q_{ld})\hat{p} - \ddot{f}_d) + \ddot{q}_{ld} \tag{6.67}$$

Note that $\hat{q}_{md}$, $\dot{\hat{q}}_{md}$ and $\ddot{\hat{q}}_{md}$ are independent each other, that is, there are no differentiation or integration relations among them. Their estimation errors are denoted by

$$\tilde{q}_{md} = q_{md} - \hat{q}_{md} \tag{6.68}$$

$$\dot{\tilde{q}}_{md} = \dot{q}_{md} - \dot{\hat{q}}_{md} \tag{6.69}$$

$$\ddot{\tilde{q}}_{md} = \ddot{q}_{md} - \ddot{\hat{q}}_{md} \tag{6.70}$$

The motor position and velocity tracking errors and their estimates are defined as

$$e_m = q_{md} - q_m \tag{6.71}$$

$$\dot{e}_m = \dot{q}_{md} - \dot{q}_m \tag{6.72}$$

$$\hat{e}_m = \hat{q}_{md} - q_m \tag{6.73}$$

$$\dot{\hat{e}}_m = \dot{\hat{q}}_{md} - \dot{q}_m \tag{6.74}$$

Obviously $\hat{e}_m = e_m - \tilde{q}_{md}$.

For the controller design, the following variables related to the tracking errors of the link position and the constraint force are also defined.

$$e \triangleq q_{ld} - q_l, \qquad e_\lambda \triangleq \lambda_d - \lambda \tag{6.75}$$

$$r \triangleq \dot{e} + K_e e, \qquad \dot{q}_r \triangleq \dot{q}_{ld} + K_e e \tag{6.76}$$

where $K_e \in R^{n\times n}$ is a positive definite constant matrix. Note that

$$r = \dot{q}_r - \dot{q}_l \tag{6.77}$$

6.2.2 Quasi-steady-state and Boundary-layer Models

Consider the following control law

$$\tau_m = \tau_s + J_m(\ddot{\hat{q}}_{md} + K_v\dot{\hat{e}}_m + K_p\hat{e}_m) - J^T(q_l)e_\lambda \tag{6.78}$$

where τ_s is a slow-time-scale control to be decided later, K_v and K_p are positive definite constant matrices.

Considering equations (6.71) and (6.73), the control law in equation (6.78) can be rewritten as

$$\tau_m = \tau_s + J_m(\ddot{q}_{md} + K_v\dot{e}_m + K_p e_m) + \beta\tilde{p} - J^T(q_l)e_\lambda \tag{6.79}$$

where

$$\beta = -J_m K_s^{-1}(\ddot{\Psi}(\ddot{q}_{ld}, \dot{q}_{ld}, \dot{q}_{ld}, q_{ld}) + K_v \dot{\Psi}(\ddot{q}_{ld}, \dot{q}_{ld}, \dot{q}_{ld}, q_{ld}) + K_p \Psi(\ddot{q}_{ld}, \dot{q}_{ld}, \dot{q}_{ld}, q_{ld}))$$

Define new variables z_0 and z_1 as follows

$$z_0 = K_s(q_{md} - q_l) \tag{6.80}$$

$$z_1 = K_s e_m + J^T(q_l)e_\lambda \tag{6.81}$$

Substituting τ_m (6.79) into equation (6.2) and noting that

$$K_s\theta = K_s(q_m - q_l) = z_0 - z_1 - J^T(q_l)e_\lambda$$

we have

$$\ddot{z}_1 + K_v\dot{z}_1 + (K_p + K_s J_m^{-1})z_1 = K_s J_m^{-1}(z_0 - \tau_s - \beta\tilde{p}) \tag{6.82}$$

With sufficiently large stiffness K_s and the control gains K_v and K_p, a very small parameter $\epsilon \to 0$ can be defined such that

$$K_p + K_s J_m^{-1} = \frac{K_1}{\epsilon^2} \tag{6.83}$$

$$K_v = \frac{K_2}{\epsilon} \tag{6.84}$$

where K_1 and K_2 are control parameters with limited magnitudes.

With the above definitions, equation (6.82) is written as

$$\epsilon^2\ddot{z}_1 + \epsilon K_2\dot{z}_1 + K_1 z_1 = (K_1 - \epsilon^2 K_p)(z_0 - \tau_s - \beta\tilde{p}) \tag{6.85}$$

Noting that

$$K_s\theta + f = K_s\theta + J^T(q_l)\lambda = z_0 - z_1$$

the controlled system can then be described by

$$M(q_l)\ddot{q}_l + C(q_l, \dot{q}_l)\dot{q}_l + G(q_l) = z_0 - z_1 \tag{6.86}$$

$$\epsilon^2\ddot{z}_1 + \epsilon K_2\dot{z}_1 + K_1 z_1 = (K_1 - \epsilon^2 K_p)(z_0 - \tau_s - \beta\tilde{p}) \tag{6.87}$$

Equations (6.86) and (6.87) are in a standard singular perturbation form

$$\dot{x} = f_1(t, x, y, \epsilon),\ x \in R^{2(n-m)+m} \tag{6.88}$$

$$\epsilon\dot{y} = f_2(t, x, y, \epsilon, \tau_s),\ y \in R^{2n} \tag{6.89}$$

where $x = [q_l^1 \; \dot{q}_l^1]$, $y = [z_1 \; \dot{z}_1]$ and f_1 and f_2 are well-defined functions.

By setting $\epsilon = 0$ in equation (6.87), we get

$$\overline{z}_1 = \overline{z}_0 - \tau_s - \beta\tilde{p} \tag{6.90}$$

where $\overline{z}_1$ and $\overline{z}_0$ are the values of z_1 and z_0 respectively when $\epsilon = 0$.

Replacing z_1 and z_0 by $\overline{z}_1$ and $\overline{z}_0$ respectively in equation (6.86), we have the following *quasi-steady-state model*

$$M(q_l)\ddot{q}_l + C(q_l, \dot{q}_l)\dot{q}_l + G(q_l) = \overline{z}_0 - \overline{z}_1 = \tau_s + \beta\tilde{p} \tag{6.91}$$

Defining $\eta = z_1 - \overline{z}_1$ and assuming that $\overline{z}_1$ is constant in a fast timescale $\tau = \frac{t}{\epsilon}$, we have the following *boundary layer system* model from equation (6.87)

$$\frac{d^2\eta}{d\tau^2} + K_2\frac{d\eta}{d\tau} + K_1(\eta + \overline{z}_1) = K_1(\overline{z}_0 - \tau_s - \beta\tilde{p}) \tag{6.92}$$

Substituting equation (6.90) into equation (6.92), we have

$$\frac{d^2\eta}{d\tau^2} + K_2\frac{d\eta}{d\tau} + K_1\eta = 0 \tag{6.93}$$

Obviously the boundary layer system in equation (6.93) is exponentially uniformly stable given $K_1 > 0$ and $K_2 > 0$. According to *Tychonov's Theorem* [57], if the quasi-steady-state system (6.91) has a unique solution $\overline{q}_l^1(t)$ for $t \in [0, t_1]$, then there is ϵ^* such that for all $\epsilon < \epsilon^*$

$$\begin{aligned} z_1(t) &= \overline{z}_1(t) + \eta(t/\epsilon) + O(\epsilon) && (6.94) \\ q_l(t) &= \overline{q}_l^1 + O(\epsilon) && (6.95) \end{aligned}$$

which holds uniformly for $t \in [0, t_1]$.

To make equations (6.94) and (6.95) valid for an infinite time interval, the quasi-steady-state system (6.91) is required to be exponentially stable [57]. This is the objective of the slow-timescale control τ_s to be developed in the next section.

6.2.3 Slow-timescale Exponentially Stable Adaptive Controller

The slow-timescale exponentially stable adaptive controller to be developed in this section is the extension of the results of [64] by considering the force tracking.

Define a stable and proper filter $w(t)$ which has the following Laplace transform

$$w(s) = \frac{\alpha}{s+\alpha} \tag{6.96}$$

where α is a positive constant.

Convoluting both sides of the link dynamic model (6.1) with $w(t)$, we have

$$\int_0^t w(t-s)(K_s\theta + J^T(q_l)\lambda)ds = \int_0^t w(t-s)(M(q_l)\ddot{q}_l + C(q_l,\dot{q}_l)\dot{q}_l + G(q_l))ds \tag{6.97}$$

It can be shown that

$$\int_0^t w(t-s)(M(q_l)\ddot{q}_l + C(q_l,\dot{q}_l)\dot{q}_l + G(q_l))ds = W(q_l,\dot{q}_l)p \tag{6.98}$$

where

$$W(q_l,\dot{q}_l) = \int_0^t w(t-s)\Psi(\ddot{q}_l,\dot{q}_l,\dot{q}_l,q_l)ds \tag{6.99}$$

Denoting the left side of equation (6.97) as $z(t)$ and considering equation (6.98), equation (6.97) can be written in a compact form

$$z(t) = W(q_l,\dot{q}_l)p \tag{6.100}$$

and the estimates of p and $z(t)$ are linked by

$$\hat{z}(t) = W(q_l,\dot{q}_l)\hat{p} \tag{6.101}$$

where $\hat{p}$ and $\hat{z}(t)$ are the estimates of p and $z(t)$ respectively.

For the exponentially stable adaptive controller for the quasi-steady-state dynamic system (6.91), we have the following theorem

Theorem 6.2.2 *The quasi-steady-state dynamic system (6.91) is exponentially stable given the following slow-time scale control and the parameter adaptation* *law:*

$$\begin{aligned}
\tau_s &= \Psi(\ddot{q}_r, \dot{q}_r, \dot{q}_l, q_l)\hat{p} + k_c \hat{M}(q) r && (6.102)\\
\dot{\hat{p}} &= \Gamma^{-1}(\Psi_s^T r + \gamma_1 W^T(q_l, \dot{q}_l)\tilde{z}) && (6.103)\\
\dot{\Gamma} &= -\gamma(t)\Gamma + 2\gamma_2 W^T(q_l, \dot{q}_l) W(q_l, \dot{q}_l) && (6.104)\\
\Psi_s &= \Psi(\ddot{q}_r + k_c r, \dot{q}_r, \dot{q}_l, q_l) - \beta && (6.105)\\
\gamma(t) &= \gamma_0(1 - \|\Gamma^{-1}\|/k_0) && (6.106)
\end{aligned}$$

where $\tilde{z} = z - \hat{z}$, $k_c > 0$ and $\gamma_1 > \gamma_2 > 0$ are control parameters. $\Gamma > 0$ is the parameter adaptation gain matrix and $\gamma(t)$ is the forgetting factor. The constants $\gamma_0 > 0$ and $k_0 > 0$ are introduced to limit the magnitude of the parameter adaptation gain and forgetting factor respectively.

Proof:

Consider a Lyapunov function candidate

$$V = \frac{1}{2}(r^T M(q_l) r + \tilde{p}^T \Gamma \tilde{p}) \tag{6.107}$$

Differentiating V with respect to time t, we have

$$\dot{V} = r^T M(q_l)\dot{r} + \frac{1}{2}\dot{M}(q_l) r - \tilde{p}^T \Gamma \dot{\hat{p}} + \frac{1}{2}\tilde{p}^T \dot{\Gamma} \tilde{p} \tag{6.108}$$

Note the fact that $\dot{\tilde{p}} = -\dot{\hat{p}}$ is used in the above derivation.

From **Property 6.3**, equation (6.108) is rewritten as

$$\dot{V} = r^T M(q_l)\dot{r} + C(q_l, \dot{q}_l) r - \tilde{p}^T \Gamma \dot{\hat{p}} + \frac{1}{2}\tilde{p}^T \dot{\Gamma} \tilde{p} \tag{6.109}$$

Substituting $r = \dot{q}_r - \dot{q}_l$ into the above equation, we have

$$\begin{aligned}
\dot{V} &= r^T(M(q_l)\ddot{q}_r + C(q_l, \dot{q}_l)\dot{q}_r - M(q_l)\ddot{q}_l - C(q_l, \dot{q}_l)\dot{q}_l) - \tilde{p}^T \Gamma \dot{\hat{p}} + \frac{1}{2}\tilde{p}^T \dot{\Gamma} \tilde{p}\\
&= r^T(\Psi(\ddot{q}_r, \dot{q}_r, \dot{q}_l, q_l)p - M(q_l)\ddot{q}_l - C(q_l, \dot{q}_l)\dot{q}_l - G(q_l)) - \tilde{p}^T \Gamma \dot{\hat{p}} + \frac{1}{2}\tilde{p}^T \dot{\Gamma} \tilde{p} \quad (6.110)
\end{aligned}$$

Note that **Property 6.4** is used in the above derivation.

Substituting equation (6.91) into equation (6.110), it follows that

$$\dot{V} = r^T(\Psi(\ddot{q}_r, \dot{q}_r, \dot{q}_l, q_l)p - \tau_s - \beta\tilde{p}) - \tilde{p}^T \Gamma \dot{\hat{p}} + \frac{1}{2}\tilde{p}^T \dot{\Gamma} \tilde{p} \tag{6.111}$$

Substituting τ_s in equation (6.102) into equation (6.111), we have

$$\begin{aligned}\dot{V} &= r^T(\Psi(\ddot{q}_r, \dot{q}_r, \dot{q}_l, q_l)\tilde{p} - k_c\hat{M}(q_l)r - \beta\tilde{p}) - \tilde{p}^T\Gamma\dot{\hat{p}} + \frac{1}{2}\tilde{p}^T\dot{\Gamma}\tilde{p} \\ &= r^T((\Psi(\ddot{q}_r, \dot{q}_r, \dot{q}_l, q_l)\tilde{p} + \tilde{M}(q_l)k_c r) - k_c M(q_l)r - \beta\tilde{p}) - \tilde{p}^T\Gamma\dot{\hat{p}} + \frac{1}{2}\tilde{p}^T\dot{\Gamma}\tilde{p} \quad (6.112)\end{aligned}$$

Considering the definition of Ψ_s in equation (6.105), equation (6.112) is rewritten as

$$\dot{V} = -k_c r^T M(q_l)r + r^T\Psi_s\tilde{p} - \tilde{p}^T\Gamma\dot{\hat{p}} + \frac{1}{2}\tilde{p}^T\dot{\Gamma}\tilde{p} \qquad (6.113)$$

Substituting $\dot{\hat{p}}$ in equation (6.103) into equation (6.113) and noting that $\tilde{z} = W(q_l, \dot{q}_l)\tilde{p}$, we have

$$\dot{V} = -k_c r^T M(q_l)r + (\gamma_2 - \gamma_1)\tilde{p}^T W^T(q_l, \dot{q}_l)W(q_l, \dot{q}_l) - \frac{1}{2}\gamma(t)\tilde{p}^T\Gamma\tilde{p} \qquad (6.114)$$

As $\gamma_1 > \gamma_2$, it follows that

$$\dot{V} \leq -k_c r^T M(q_l)r - \frac{1}{2}\gamma(t)\tilde{p}^T\Gamma\tilde{p} \leq -k_m(\frac{1}{2}r^T M(q_l)r + \frac{1}{2}\tilde{p}^T\Gamma\tilde{p}) = -k_m V \quad (6.115)$$

where $k_m = min(2k_c, \eta(t)) > 0$.

As $\dot{V} \leq -k_m V$, $V \leq V(0)e^{-k_m t}$. It means that $r \rightarrow 0$ and $\tilde{p} \rightarrow 0$ exponentially. From the definition of r, it can be concluded that $e \rightarrow 0$ and $\dot{e} \rightarrow 0$ exponentially.

Q. E. D

The parameter adaptation gain matrix Γ^{-1} is time varying and its behavior is specified by equations (6.104) and (6.106) which were proposed in [101]. For its boundedness, we have the following lemma.

Lemma 6.2.1 *From equations (6.104) and (6.106),* $\gamma(t) \geq 0$ *and* $\Gamma^{-1} \leq k_0 I$, $\forall t > 0$, *where* I *is an identity matrix with a compatible dimension. If* $W(q_l, \dot{q}_l)$ *is persistently exciting, then* Γ^{-1} *is uniformly upper and lower bounded.*

The proof of **Lemma 6.2.1** can be found in **Appendix D**.

Remark 6.2.5 *Given the quasi-steady-state system (6.91) being exponentially stable as proven above, it is thus concluded that equations (6.94) and (6.95) are valid for infinite time interval.*

Remark 6.2.6 *The behavior of link position q_l is clearly described by equation (6.95), whereas that of the constraint force λ is included in z_1 defined in equation (6.81). To analyze the performance of the constraint force tracking, multiplying both sides of equation (6.81) by $L^T(q_l)$ and considering* **Property 6.1**, *we have*

$$L^T(q_l)z_1 = L^T(q_l)K_s e_m$$

From equation (6.2.6) and the fact that z_1 is bounded and $L(q_l)$ is normally nonsingular, e_m is bounded. Now that z_1 and e_m are all bounded and $J(q_l)$ is nonsingular in equation (6.94), it can be concluded that e_λ is also bounded.

Remark 6.2.7 *Link flexibility is not considered in the design of the above controllers. The impact energy is reduced with the flexible links when the robot collides with the constraint, but the position accuracy is reduced and the oscillation of the robotic arm may be manifested [102]. It is well known that control of a flexible link robot is tough if without some idealistic assumptions [103]. Though there are some research results for the low degrees freedom flexible link robots based on lumped masses method, the control of constrained flexible link robotic manipulators is still an open problem partially due to the requirement of infinite-dimensional analysis in the dynamic modeling [104]. The problem will become much tougher or intractable if the flexibilities of the joints and the links of the robot are addressed at the same time.*

6.3 Simulation

The simulation example is the same as that schematically shown in Figure 3.1 in Chapter 3 except that the robot joints are flexible. For the completeness in discussion, it is shown again in Figure 6.1. In this example, the end effector of a flexible joint manipulator moves along a part of the constraint surface and exerts a force on it at the same time. The length, inertia and the mass of each link of the manipulator are $l_i = 0.3m$, $I_i = 0.3kgm^2$ and $m_i = 0.1kg$ respectively ($i = 1, 2$). The half of the link length is $d_i = \frac{l_i}{2} = 0.15m$ ($i = 1, 2$). The mass center of each link is assumed to be in the middle of the link.

In Figure 6.1, the world coordinates is denoted by *oxy*. The constraint surface is described by

$$\Phi(r_d) = x_d - y_d + 0.25 = 0 \tag{6.116}$$

It is planned that the end effector of the robot moves along the following trajectory on the constraint surface

$$\begin{aligned} x_d(t) &= -\frac{1}{10}\cos(2t) \\ y_d(t) &= 0.35 - \frac{1}{10}\cos(2t) \end{aligned}$$

while the magnitude of the constraint force is kept at $\lambda_d = 2N$.

Let the link position $q_l = [\theta_1 \;\; \theta_2]^T$ and its partitions are $q_l^1 = \theta_1$ and $q_l^2 = \theta_2$ respectively. The desired link positions $q_{ld} = [\theta_{1d} \; \theta_{2d}]^T$ are obtained from $x_d(t)$ and $y_d(t)$ such that

$$\begin{aligned} \cos(\theta_{2d}(t)) &= \frac{x_d^2(t) + y_d^2(t) - d_1^2 - d_2^2}{2d_1 d_2} \\ \sin(\theta_{2d}(t)) &= -\sqrt{1 - \cos^2(\theta_{2d}(t))} \\ \cos(\theta_{1d}(t)) &= \frac{(d_1 + d_2\cos(\theta_{2d}(t))x_d(t) + d_2\sin(\theta_{2d}(t)y_d(t)}{x_d^2(t) + y_d^2(t)} \\ \sin(\theta_{1d}(t)) &= \frac{(d_1 + d_2\cos(\theta_{2d}(t)))y_d(t) - d_2\sin(\theta_{2d}(t)x_d(t)}{x_d^2(t) + y_d^2(t)} \end{aligned}$$

The Jacobians $J(q_l)$ and $L(q_l^1)$ are derived as follows

$$J(q_l) = \begin{bmatrix} -d_1\sin(\theta_1) - d_2\sin(\theta_1+\theta_2) & -d_2\sin(\theta_1+\theta_2) \\ d_1\cos(\theta_1) + d_2\cos(\theta_1+\theta_2) & d_2\cos(\theta_1+\theta_2) \end{bmatrix} \tag{6.117}$$

$$L(q_l^1) = [1 \quad -1 - \frac{d_1(\sin(\theta_1) + \cos(\theta_1))}{d_2\sin(\theta_1+\theta_2)\cos(\theta_1+\theta_2)}] \tag{6.118}$$

The desired link velocity $\dot{q}_{ld}$ is obtained through $\dot{q}_{ld} = J(q_{ld})[\dot{x}_d \; \dot{y}_d]^T$.

Choosing the system parameter vector p as

$$p = [p_1 \;\; p_2 \;\; p_3 \;\; p_4 \;\; p_5] \tag{6.119}$$

where

$$p_1 = I_1 + m_1 l_1^2 + I_2 + m_2(d_1^2 + l_2^2)$$

$$\begin{aligned}
p_2 &= m_2 d_1 l_2 \\
p_3 &= I_2 + m_2 l_2^2 \\
p_4 &= (m_1 l_1 + m_2 d_1) g \\
p_5 &= m_2 l_2 g
\end{aligned}$$

and $g = 9.8 m/s^2$ is the gravitational acceleration, the regressor matrix $\Psi(\dot{x}, x, \dot{q}_l, q_l)$ for a given vector $x = [x_1 \quad x_2]^T$ is derived such that

$$\Psi(\dot{x}, x, \dot{q}_l, q_l) = \begin{bmatrix} \Psi_{11} & \Psi_{12} & \Psi_{13} & \Psi_{14} & \Psi_{15} \\ \Psi_{21} & \Psi_{22} & \Psi_{23} & \Psi_{24} & \Psi_{25} \end{bmatrix} \qquad (6.120)$$

where

$$\begin{aligned}
\Psi_{11} &= \dot{x}_1 \\
\Psi_{12} &= 2\cos(\theta_2)\dot{x}_1 + \cos(\theta_2)\dot{x}_2 - \sin(\theta_2)(\dot{\theta}_1 + \dot{\theta}_2)x_2 - \sin(\theta_2)\dot{\theta}_2 x_1 \\
\Psi_{13} &= \dot{x}_2 \\
\Psi_{14} &= \cos(\theta_1) \\
\Psi_{15} &= \cos(\theta_1 + \theta_2) \\
\Psi_{21} &= 0 \\
\Psi_{22} &= \cos(\theta_2)\dot{x}_1 + \sin(\theta_2)\dot{\theta}_1 x_1 \\
\Psi_{23} &= \dot{x}_2 + \dot{x}_1 \\
\Psi_{24} &= 0 \\
\Psi_{25} &= \cos(\theta_1 + \theta_2)
\end{aligned}$$

The parameter vector p and the regressor matrix Ψ used for the robust adaptive controller in equation (6.58)(**Theorem th:theoremFle**) are expanded to one with a higher dimension such that

$$p = [p_1 \quad p_2 \quad p_3 \quad p_4 \quad p_5 \quad p_1 \quad p_2 \quad p_3 \quad p_4 \quad p_5]$$

and

$$\Psi(\dot{x}, x, \dot{q}_l, q_l) = \begin{bmatrix} \Psi_{11} & \Psi_{12} & \Psi_{13} & \Psi_{14} & \Psi_{15} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \Psi_{21} & \Psi_{22} & \Psi_{23} & \Psi_{24} & \Psi_{25} \end{bmatrix}$$

where p_i and Ψ_{ij} are the same as those defined in equations (6.119) and (6.120) respectively.

With the given physical parameters, it can be calculated that $p_1 = 0.6135$, $p_2 = 0.0045$, $p_3 = 0.30225$, $p_4 = 0.441$ and $p_5 = 0.147$. For the simulation purpose, the initial values of their estimates are set as $\hat{p}_1(0) = 0.092$, $\hat{p}_2(0) = 0.0007$, $\hat{p}_3(0) = 0.045$, $\hat{p}_4(0) = 0.066$ and $\hat{p}_5(0) = 0.022$. Assume that the robotic manipulator is initially at rest with $q_l(0) = [2.85 \quad -1.77]^T(rad)$, $\dot{q}_l(0) = [0 \quad 0]^T(rad/sec)$ and $\lambda(0) = 2.0N$.

6.3.1 Simulation for Robust Adaptive Controller

Case 1: *Robot with Strong Joint Flexibility*

In this case, the joint stiffness are set as $k_{s1} = k_{s2} = 10.0\ Nm$. The moments of inertia of the motors are set as $j_{m1} = j_{m2} = 0.5\ Nms^2$. The estimates of J_m and K_s are set to be $\hat{K}_s = diag[8.0] \in R^{2\times2}$ and $\hat{J}_m = diag[0.4] \in R^{2\times2}$ respectively. The up bounds of the norms of $\tilde{K}_s$ and $\tilde{J}_m$ are selected as $\delta_K = 4$ and $\delta_J = 2$ respectively. The control and the parameter adaptation gains are chosen as $K_\sigma = diag[1.0] \in R^{2\times2}$, $K_e = diag[3.5] \in R^{2\times2}$, $K_\mu = diag[1.5] \in R^{2\times2}$, $\Gamma_p = \Gamma_K = diag[0.08] \in R^{2\times2}$. The width of the boundary layer is chosen to be $\Delta = 0.002$.

The position and force tracking performances of the robot are plotted in Figures 6.2 and 6.3 respectively. The control torques are given in Figure 6.4. The performance of parameter adaptations are plotted in Figure 6.5 and Figure 6.6. It can be seen that under the proposed controller, the link positions of the robot converge to their desired values and the force tracking error is bounded. The control torques demonstrate a big fluctuation at the beginning of the simulation. It is gradually reduced to a reasonable range after some time. The parameter estimations are also stabilized and bounded. Due to the introduction of boundary layer, the torque and force signals are quite smooth.

Case 2: *Robot with Weak Joint Flexibility*

In this case, the joint stiffness are set to be bigger values: $k_{s1} = k_{s2} = 50.0\ Nm$. The other parameters of the controller remain the same as in the case when the joint flexibility is strong. The simulation results are plotted from Figures 6.7 to 6.11. Though the position tracking is almost as good as that in the case when the joint flexibility is strong, the responses of the constraint force and the control inputs demonstrate bigger fluctuations. The magnitudes of the joint torques also increase. Once K_s becomes very big(e.g $K_s = diag[100.0] \in R^{2\times 2}$, it can be observed that the responses of the control system become divergent.

6.3.2 Simulation for Singular Perturbation Based Controller

In this case, the joint stiffness and the moments of inertia of the motors are known. Their values are selected as $k_{s1} = k_{s2} = 300.0\ Nm$ and $j_{m1} = j_{m2} = 1.0\ Nms^2$. Note that the joint stiffness is very big. The control gains are chosen as $K_p = diag[80.0]$, $K_v = diag[80.0]$ and $k_c = 1.50$. For the parameter adaptations, we choose $\gamma_0 = 1.0$, $\gamma_1 = 7.0$, $\gamma_2 = 2.0$ and $\Gamma(0) = diag[60.0]$. The parameter α in the filter $w(s)$ is chosen as $\alpha = 5.0$.

The position and force tracking performances of the robot are plotted in Figures 6.12 and 6.13 respectively. The control torques are given in Figure 6.14. The parameter adaptations are plotted in Figure 6.15. It can be seen that under the proposed controller, the link positions of the robot converge to their desired values and the force tracking error is bounded. The control torque for each joint is in a reasonable range. The parameter estimation are also stabilized and bounded after some time.

The simulation is also done when the adaptive slow-time-scale controlτ_s is not included in the motor control input τ_m. The responses of the link position, constraint force and the control torques are plotted from Figures 6.16 to 6.18. It can be seen that the link position cannot converge to its desired trajectory and the constraint force shows bigger tracking errors.

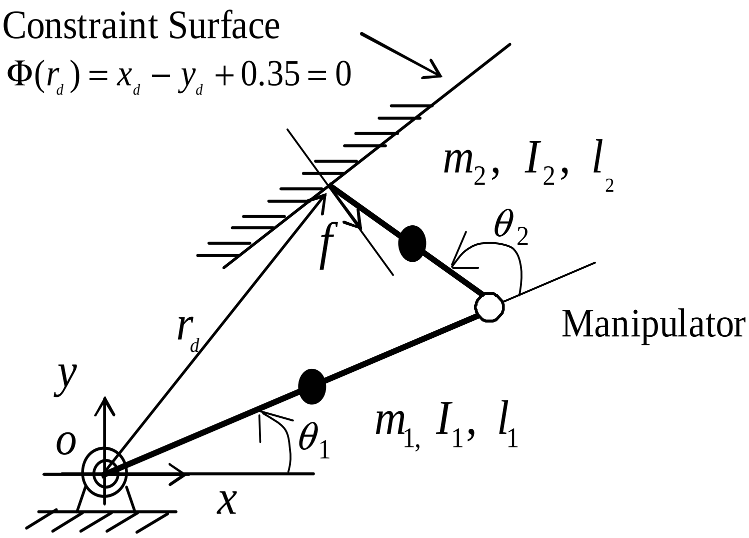

Figure 6.1: Simulation Example

6.4 Conclusion

A robust adaptive controller and a singular perturbation based controller are developed to achieve position/force controller for an uncertain constrained flexible joint robots (FJRs). The robust adaptive controller is developed for the constrained robot with arbitrary unknown joint stiffness. This makes it more general than many commonly used controllers which require that the stiffness of the robot joint is sufficiently large. A singular perturbation based controller was the extension of the one for free robots. By properly defining the fast and the slow variables with the position and force tracking errors, a boundary layer system and the quasi-steady-state system were established which are exponentially stable. Both controllers mainly relied on the robot's position, velocity and constraint force feedback. The position tracking and the boundedness of force errors were achieved. The simulation study was conducted to verify the effectiveness of the proposed approaches.

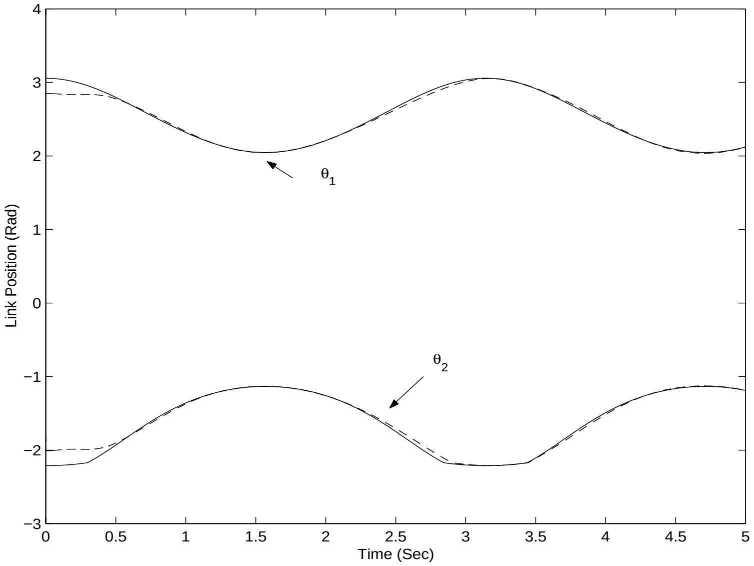

Figure 6.2: Position Tracking when $K_s = diag[10.0]$ (Solid: Desired position, Dashed: Actual Position)

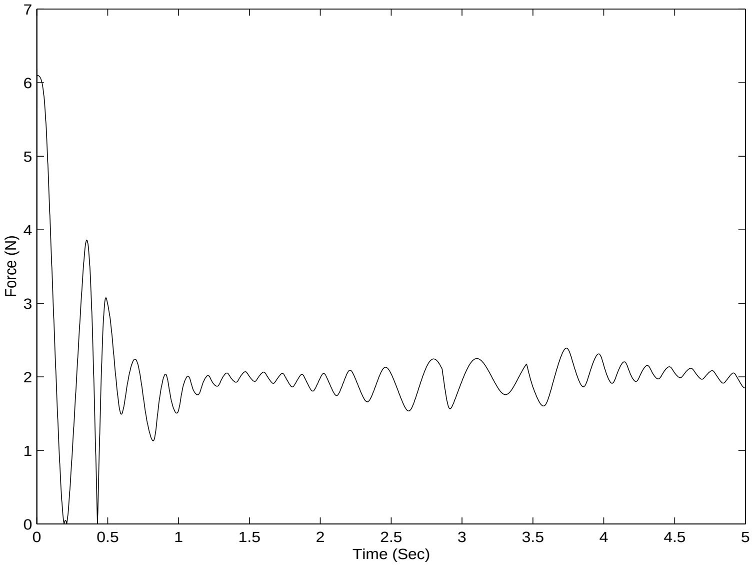

Figure 6.3: Force Tracking when $K_s = diag[10.0]$ (Solid: Desired force, Dashed: Actual force)

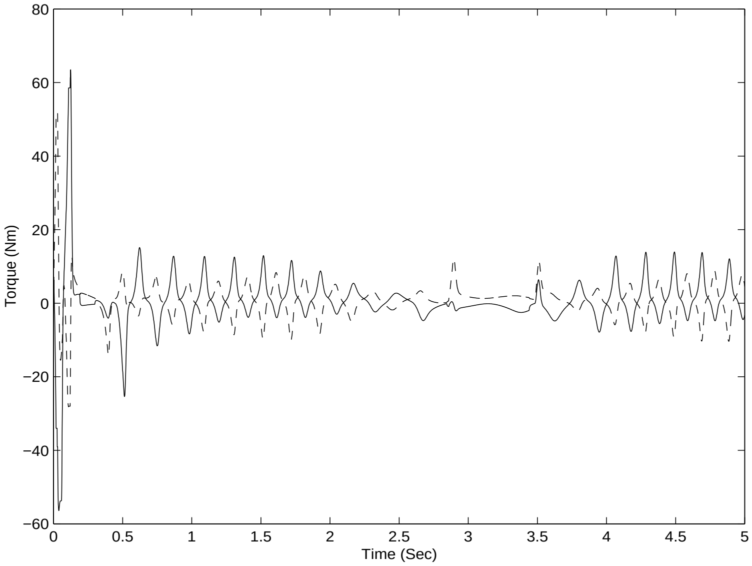

Figure 6.4: Joint Torques when $K_s = diag[10.0]$ (Solid: Joint 1 torque, Dashed: Joint 2 torque)

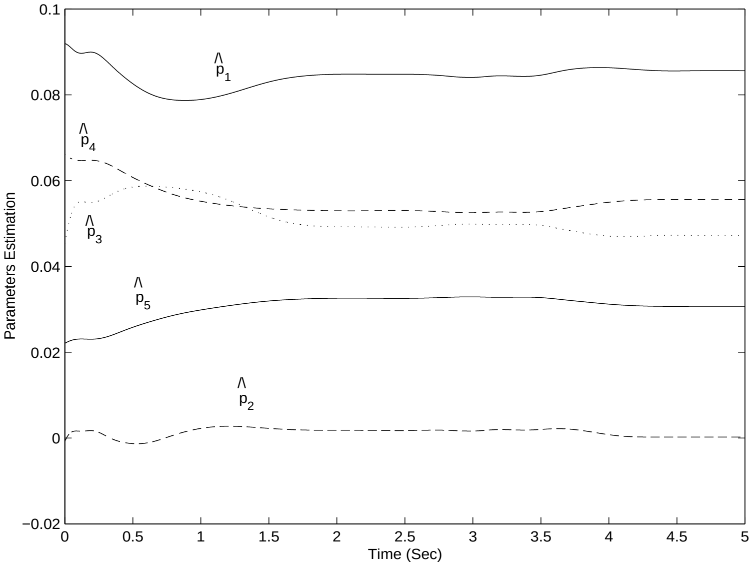

Figure 6.5: Parameter Estimations ($\hat{p}_1$, $\hat{p}_2$, $\hat{p}_3$, $\hat{p}_4$ and $\hat{p}_5$) when $K_s = diag[10.0]$

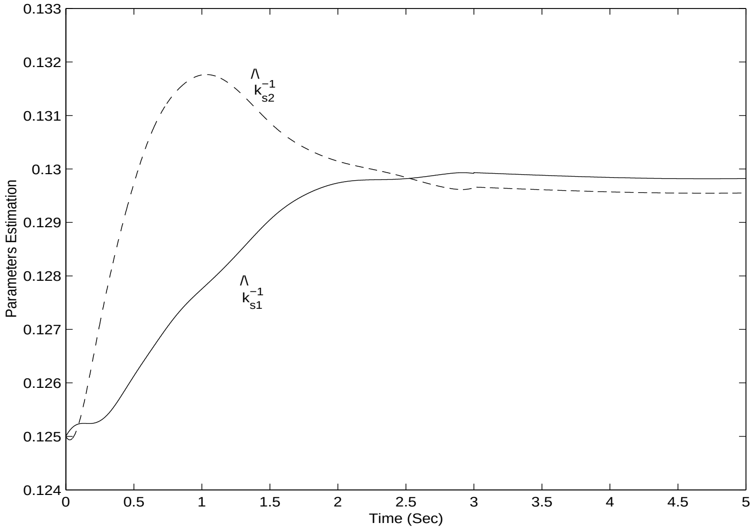

Figure 6.6: Parameter Estimations ($\hat{k}_{s1}^{-1}$ and $\hat{k}_{s2}^{-1}$) when $K_s = diag[10.0]$

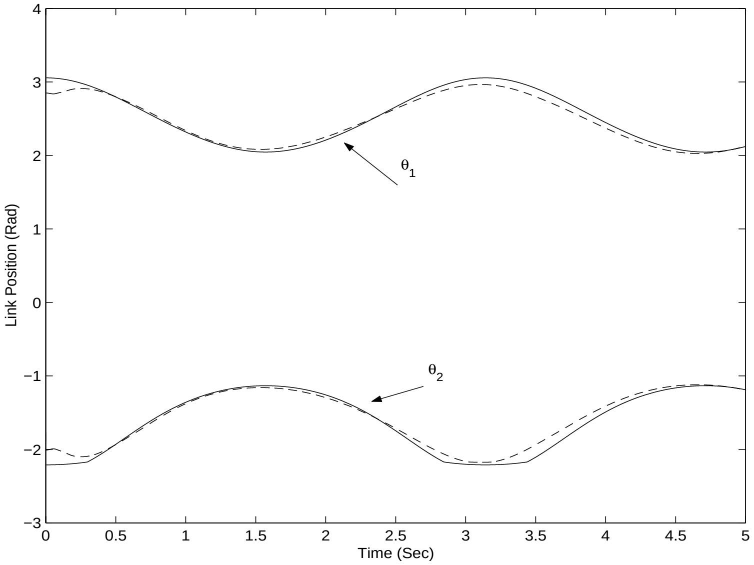

Figure 6.7: Position Tracking when $K_s = diag[50.0]$ (Solid: Desired position, Dashed: Actual Position)

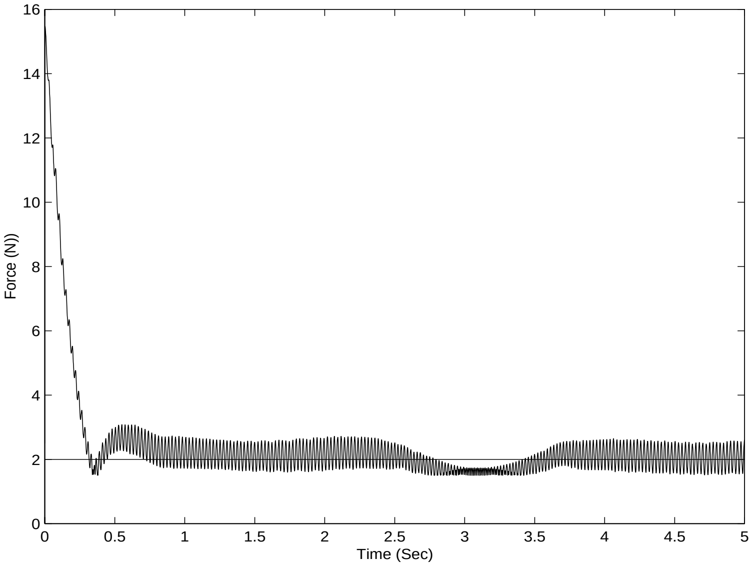

Figure 6.8: Force Tracking when $K_s = diag[50.0]$ (Solid: Desired force, Dashed: Actual force)

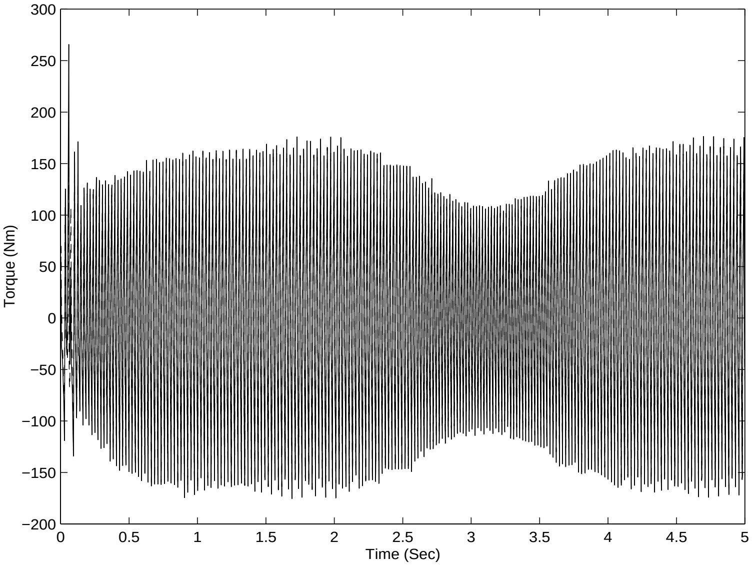

Figure 6.9: Joint Torques when $K_s = diag[50.0]$ (Solid: Joint 1 torque, Dashed: Joint 2 torque)

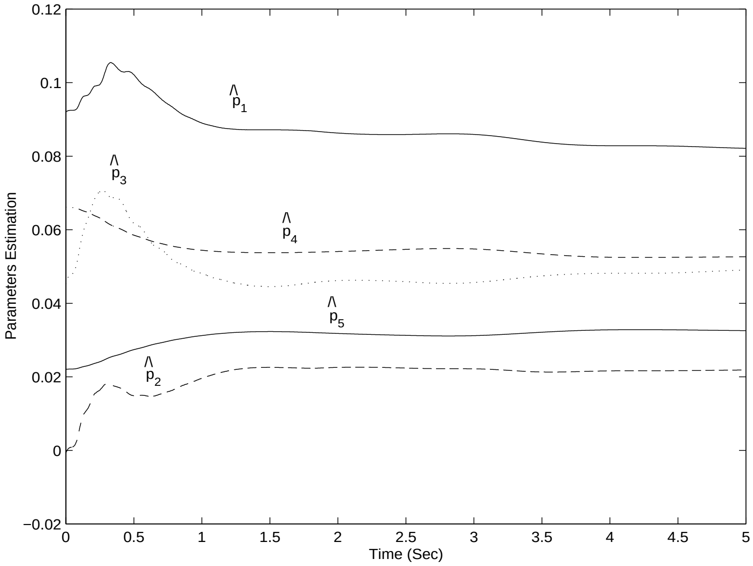

Figure 6.10: Parameter Estimations ($\hat{p}_1$, $\hat{p}_2$, $\hat{p}_3$, $\hat{p}_4$ and $\hat{p}_5$)when $K_s = diag[50.0]$

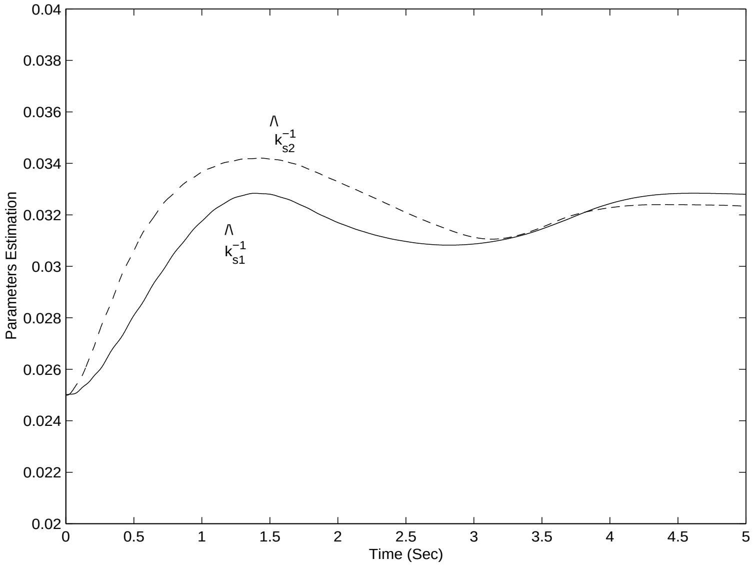

Figure 6.11: Parameter Estimations ($\hat{k}_{s1}^{-1}$ and $\hat{k}_{s2}^{-1}$) when $K_s = diag[50.0]$

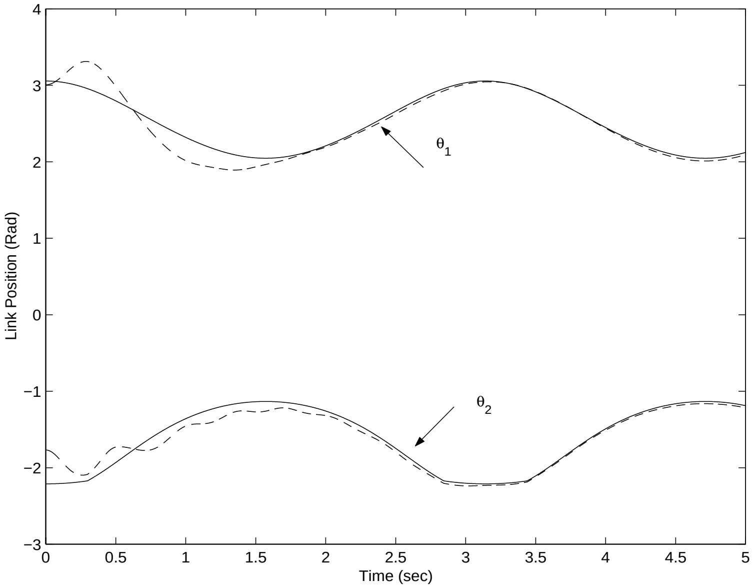

Figure 6.12: Position tracking (Solid: desired position, Dashed: actual position)

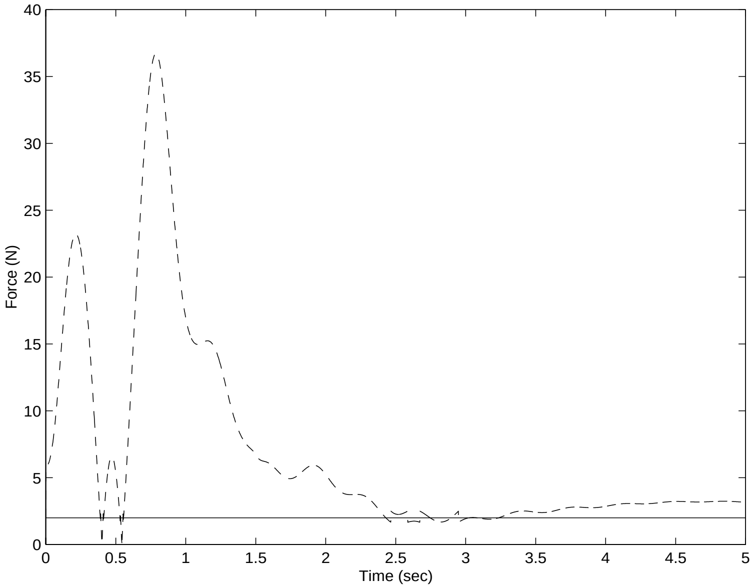

Figure 6.13: Force tracking (Solid:λ_d, Dashed: λ)

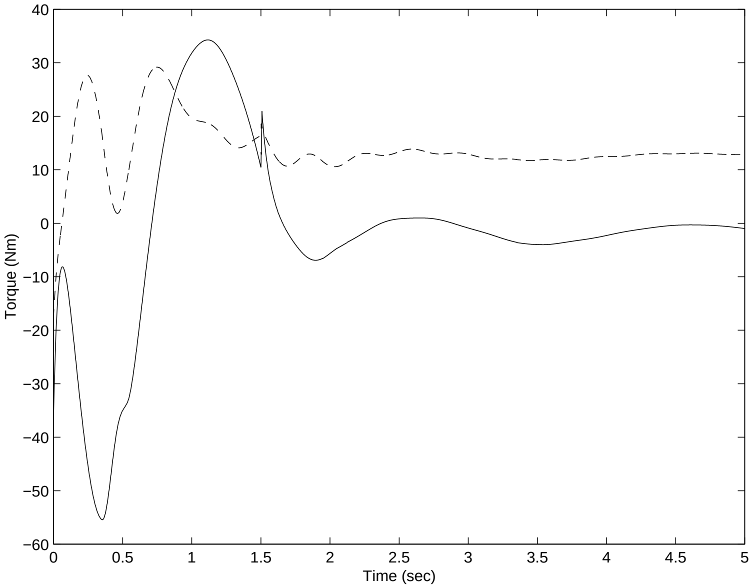

Figure 6.14: Joint torques (Solid: τ_{m1}, Dashed: τ_{m2})

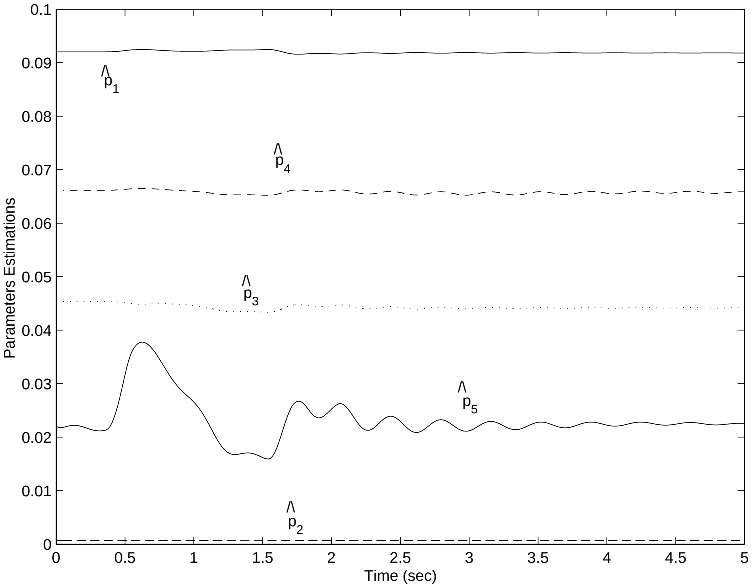

Figure 6.15: Parameter estimates

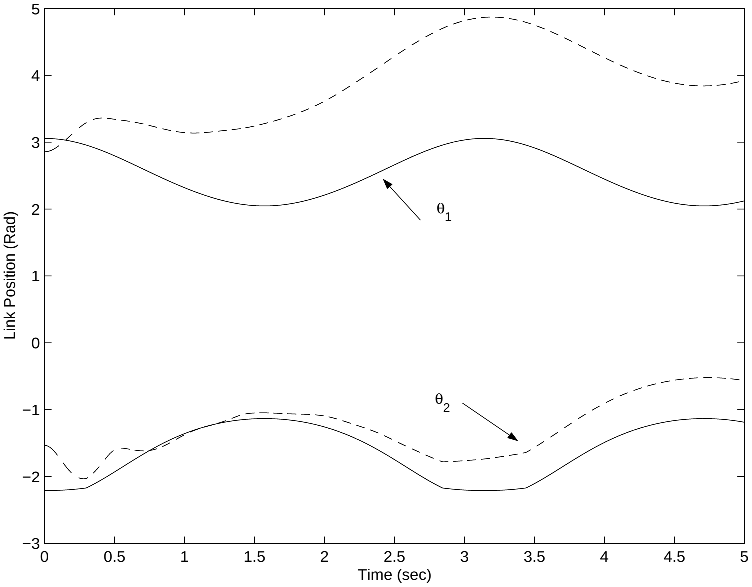

Figure 6.16: Position tracking with only motor feedback control (Solid: desired position, Dashed: actual position)

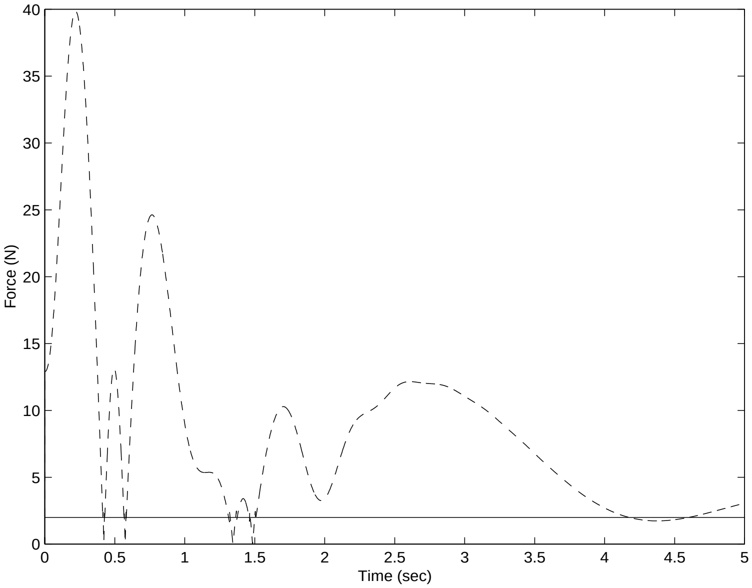

Figure 6.17: Force tracking with only motor feedback control (Solid:λ_d, Dashed: λ

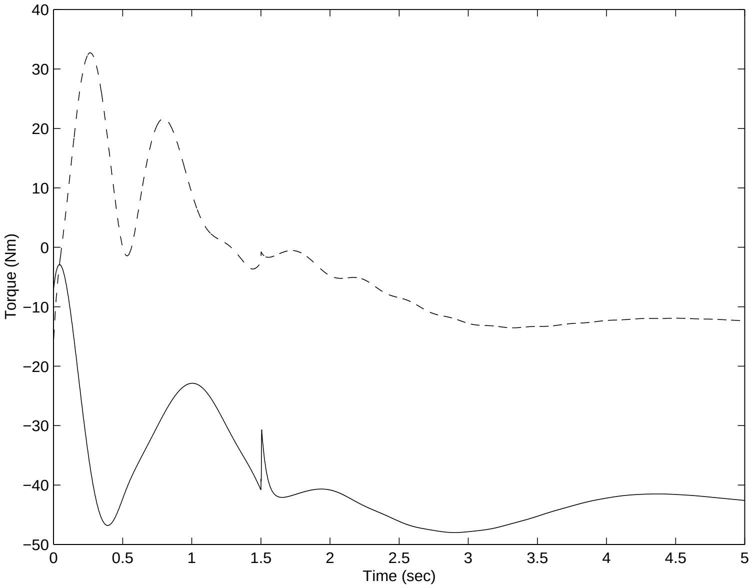

Figure 6.18: Joint torques with only motor feedback control (Solid: τ_{m1}, Dashed: τ_{m2}

Chapter 7

Conclusions and Future Research

7.1 Conclusions

Position/force control of constrained robotic manipulators are addressed in the thesis. The controllers are designed considering various issues which were overlooked or were not fully addressed in the past. These include that the constraint is in motion, that the dynamics of the constraint affects the constrained robot system and that the robot joints are flexible with unknown stiffness, to name a few. The unidirectional force control and position/force tracking within impedance control framework are also studied. Various control approaches such as nonlinear feedback, adaptive control, robust control, fuzzy and neural network control are used for the controller design. The effectiveness of the control approaches are verified by the simulation results.

In Chapter 2, the system dynamic model of a robotic manipulator constrained by a moving object is established and its properties are studied. A model based adaptive controller and a model free neural network controller for position/force control of the robot are developed.

In Chapter 3, robust adaptive and neural network based impedance control schemes are developed for robot's position/force tracking as well as regulation of the general impedance between the robot and the constraint. The parameters of the desired impedance are tuned together with other parameters of the robot system. The robot position tracking and the boundedness of the force tracking are achieved with the controllers developed.

Chapter 4 is dedicated to the explicit force control of a constrained robot considering the dynamics of the contact between the robot's end effector and the constraint. The dynamic behavior of the contact is modeled as that of a general chained multiple mass-spring-damper (CMMSD) system. A model reference and a backstepping adaptive output feedback controllers are developed respectively.

Chapter 5 focuses on the unidirectional force control for keeping the contact between the end effector of a robot and the constraint. Several fuzzy rules are extracted from human being's experience in making a contact on an object with fingers and the robot's impedance model. The controller is developed by combining the fuzzy control and the impedance control.

Chapter 6 is on the position/force control for a constrained flexible joint robot. The controller developed is more general than other existing control approaches as the stiffness of the joint is treated unknown without restriction on its magnitude. The controller is developed using Lyapunov cascade design approach first, and then the singular perturbation approach for free flexible joint robot is extended to control a constrained flexible joint robot with large joint stiffness.

7.2 Future Research

There are still some investigations that can be done to extend the work in the thesis. For example,

1. *The dynamic modeling of the collision between the end effector of the robot and the constraint should be investigated for position/force control for a constrained robot.*

   Most controllers developed in the thesis did not take the collision between the robot's end effector and the constraint into consideration. The unidirectional force controller developed in Chapter 5 is aimed to avoid such a collision to happen. Though there are many models available for analyzing collision behaviors in the area of multi-body mechanics, they are too complicated to be suitable for controller synthesis. Further investigations can be made on using or modifying those available collision models for the purpose of position/force control design.

2. *Integration of unidirectional force control with constrained robot control approaches.*

   The fuzzy unidirectional force control proposed in Chapter 5 is aimed at keeping the contact between the end effector of the robot and the constraint. It is developed within the framework of impedance control. Further studies should be done in integrating it with constrained motion controllers where the maintenance of the contact of the robot's end effector on the constraint is important.

3. *Unified position/force control for constrained robots with arbitrary joint stiffness.*

   Singular perturbation based position/force controllers are only applicable for the robots with sufficiently large stiffness. The controller developed in Chapter 6 does away with the assumption of large joint stiffness, but the responses of position and constraint force are still sensitive to the change of joint stiffness. Further researches should be done to unify various control approaches so that the consistent performance of the controlled system can be kept regardless of joint stiffness. The effects of the force signals on the stability of the overall controlled system should be investigated further.

4. *Position/force control for constrained robots with flexible links.*

   The robots with flexible links is normally lighter than those built with rigid heavy links. They can also accommodate the impact with the constraint more easily than rigid link robots. There are many open problems about the control of a constrained flexile link robot including the dynamic modeling, system stability, position and force tracking accuracy etc. Considering the range and depth of the issues concerned, it should be more appropriate to address them in a separate thesis.

5. *Implementation of the proposed controllers.*

   An obstacle for the implementation of the proposed controllers are intensive computations involved in the control design. To cope with this problem, the control algorithms should be made more computationally efficient together with the design of the hardware of the control system.

Appendix A

Proof of Property 2.1

Solving for v_{co} from equation (2.9) yields

$$v_{co} = R_A^{-1}(v_c - Av_o) = R_A^T(v_c - Av_o) \tag{A.1}$$

From the kinematics of the robots, we have

$$v_c = J_1(q_1)\dot{q}_1 \tag{A.2}$$

$$v_o = J_2(q_2)\dot{q}_2 \tag{A.3}$$

Substituting equations (A.2) – (A.3) into equation (A.1) and noting equation (2.18), we have

$$v_{co} = R_A^T J^1(q^1)L(q^1)\dot{q}^1 \tag{A.4}$$

As v_{co} and n_{co} are orthogonal to each other, we have

$$n_{co}^T v_{co} = 0 \tag{A.5}$$

From equations (A.1)—(A.4), we obtain

$$n_c^T J^1(q^1)L(q^1)\dot{q}^1 = 0 \tag{A.6}$$

As $\dot{q}^1$ are independent variables, the following equation holds

$$L^T(q^1)J^{1T}(q^1)n_c = 0 \tag{A.7}$$

Appendix B

Proof of Lemma 4.1.1

For applying Routh's stability criterion, the coefficients of the polynomial: $b_3s^3 + b_2s^2 + b_1s + b_0$ are arranged in the following pattern:

$$\begin{array}{ccc} s^3 & b_3 & b_1 \\ s^2 & b_2 & b_0 \\ s^1 & m_1 & \\ s^0 & n_1 & \end{array} \tag{B.1}$$

where

$$m_1 = \frac{b_2b_1 - b_3b_0}{b_2} \tag{B.2}$$

$$n_1 = b_0 \tag{B.3}$$

By definition, $b_3 > 0$, $b_2 > 0$ and $n_1 = b_0 > 0$. To exam the sign of m_1 by substituting b_2, b_1, b_3 and b_0 in the equation (B.2), we have

$$m_1 = \frac{k^2k_s}{m^2m_c}((k_0k_sk_c + k_ck_0^2)(\frac{1}{b_s} - 1) + R_1(a,b)) \tag{B.4}$$

$$= \frac{k^2k_s}{m^2m_c}(k_ck_0^2(1 - \frac{1}{b_s}) + k_0k_s(k_0\frac{b_c}{b_s^2} - k_c) + R_2(a,b)) \tag{B.5}$$

where $R_i(a,b) > 0$ $(i = 1, 2)$ is a positive polynomial depending on the way the terms are grouped.

From equation (B.4), $m_1 > 0$ if $b_s < 1$. If $b_s > 1$, m_1 is still positive from equation (B.5). From Ruth's stability criterion, $b_3s^3 + b_2s^2 + b_1s + b_0$ is Hurwitz and thus the model (4.6) is of minimum phase.

Appendix C

CMMSD Systems – Modeling and Control

C.1 Dynamic Modeling and Problem Formulation

A chained multiple mass spring damper (CMMSD) system with n mass units is schematically shown in Figure C.1, where m_i is the mass, b_i is the viscous coefficient and q_i is the displacement measured from the equilibrium position along the X axis of the ith unit $(i = 1, 2 \dots n)$. There are $n-1$ springs connecting all the mass units with k_i being the linear spring constant$(i = 1, 2 \dots n-1)$. q_1 is the only measurable output of the system and u is the input force. According to Newton's second law, the system dynamic model is derived such that

$$\begin{aligned} m_1\ddot{q}_1 &= -b_1\dot{q}_1 + k_1(q_2 - q_1) = -b_1\dot{q}_1 - k_1q_1 + k_1q_2 \\ m_2\ddot{q}_2 &= -b_2\dot{q}_2 - (k_1 + k_2)q_2 + k_1q_1 + k_2q_3 \\ m_i\ddot{q}_i &= -b_i\dot{q}_i - (k_{i-1} + k_i)q_i + k_{i-1}q_{i-1} + k_iq_{i+1} \\ &\qquad (i = 3, 4 \dots n-1) \\ m_n\ddot{q}_n &= -b_n\dot{q}_n - k_{n-1}q_n + k_{n-1}q_{n-1} + u \end{aligned} \tag{C.1}$$

Define $x_1 = q_1$, $x_2 = \dot{q}_1$, $x_3 = q_2$, $x_4 = \dot{q}_2$, …, $x_{2i-1} = q_i$, $x_{2i} = \dot{q}_i$, …, $x_{2n-1} = q_n$, $x_{2n} = \dot{q}_n$, $x = [x_1 \ x_2 \ \dots x_{2n}]^T \in R^{2n}$ and c_j is the jth column vector of identity matrix $I^{2n\times 2n}$.

The original dynamic system (C.1) is transformed to the following state space model

$$\begin{aligned} \dot{x} &= A_x x + b_x u & \text{(C.2)} \\ x_1 &= c_1^T x & \text{(C.3)} \end{aligned}$$

where

$$\begin{aligned} b_x &= m_n^{-1} c_{2n} \\ A_x &= [A_1^T \;\; A_2^T \;\; A_{2i-1}^T \;\; A_{2i}^T \ldots A_{2n-1}^T \; A_{2n}^T]^T \\ A_1 &= c_2^T \\ A_2 &= -m_1^{-1}(k_1 c_1^T + b_1 c_2^T - k_1 c_3^T) \\ A_{2i-1} &= c_{2i}^T \\ A_{2i} &= m_i^{-1}(k_{i-1} c_{2i-3}^T - (k_{i-1} + k_i) c_{2i-1}^T - b_i c_{2i}^T) + m_i^{-1} k_i c_{2i+1}^T \;\; (i = 2,\; 3 \ldots n-1) \\ A_{2n-1} &= c_{2n}^T \\ A_{2n} &= m_n^{-1}(k_{n-1} c_{2n-3}^T - k_{n-1} c_{2n-1}^T - b_n c_{2n}^T) \end{aligned}$$

Through Laplace transformation on state space model (C.2)–(C.3), we have

$$X_1(s) = H_{2n}(s) U(s) \tag{C.4}$$

where $X_1(s)$ and $U(s)$ are the Laplace transformation of x_1 and u respectively, H_{2n} is the transfer function defined as

$$H_{2n}(s) = c_1^T (sI - Ax)^{-1} b_x = \frac{d_{2n}}{s^{2n} + \Sigma_{j=0}^{2n-1} a_{2n,j} s^j} \tag{C.5}$$

where

$$\begin{aligned} d_{2i} &= m_i^{-1} d_{2i-2} \\ a_{2i,j} &= a_{2i-2,j-2} + m_i^{-1} b_i (a_{2i-2,j-1} + \sigma(j - 2i + 1)) + m_i^{-1} k_{i-1} (a_{2i-2,j} + \sigma(j - 2i + 2)) \\ &\quad - d_{2i-4}^{-1} d_{2i-2} m_i^{-1} k_{i-1} (a_{2i-4,j} + \sigma(j - 2i + 4)) \\ d_{2l} &= m_l^{-1} k_l d_{2l-2} \\ a_{2l,j} &= a_{2l-2,j-2} + m_l^{-1} b_l (a_{2l-2,j-1} + \sigma(j - 2l + 1)) + m_l^{-1} (k_{l-1} + k_l)(a_{2l-2,j} + \sigma(j - 2l + 2)) \\ &\quad - d_{2l-4}^{-1} d_{2l-2} m_l^{-1} k_{l-1} (a_{2l-4,j} + \sigma(j - 2l + 4)) \end{aligned}$$

$$(l = 2,\ 3,\ \dots i-1,\ \ j = 0,\ 1\,2, \dots l-1)$$
$$d_2 = \frac{k_1}{m_1}, a_{2,0} = \frac{k_1}{m_1}, \quad a_{2,1} = \frac{b_1}{m_1}$$

Expressing the state space equations in observer canonical form for system (C.4), we have

$$\begin{aligned} \dot{y} &= Ay + B(y_1, u)^T \theta && \text{(C.6)} \\ y_1 &= x_1 = c_1^T y && \text{(C.7)} \end{aligned}$$

where

$$\begin{aligned} \dot{y}_1 &= y_2 - a_{2n,2n-1} y_1 && \text{(C.8)} \\ \vdots & \quad \vdots \\ \dot{y}_i &= y_{i+1} - a_{2n,2n-i} y_1 && \text{(C.9)} \\ \vdots & \quad \vdots \\ \dot{y}_{2n} &= d_{2n} u && \text{(C.10)} \\ A &= \begin{bmatrix} 0 & & \\ \vdots & I^{2n-1} & \\ 0 & \dots & 0 \end{bmatrix} \\ B^T &= [c_{2n} u \quad - I^{2n \times 2n} y_1] \\ \theta &= [d_{2n} \ \ a_{2n,2n-1} \ \ a_{2n,2n-2} \ \ \dots a_{2n,1} \ \ a_{2n,0}]^T \end{aligned}$$

Note that the structure of the dynamic model (C.6) is the same as that presented in [48], thus the methods of controller design in [48] can be applied for the CMMSD system.

Assuming that only the state of the first unit (y_1 and $\dot{y}_1$) of the CMMSD system is measurable, the control input u should be derived to regulate the output y_1 to zero. This is a typical adaptive output feedback control problem and the backstepping control procedure in [48] can be used to tackle it.

C.2 Adaptive Output Feedback Control

As a preparation for the controller design, the following filters are designed,

$$\dot{\xi} = A_0\xi + \lambda y_1 \tag{C.11}$$

$$\dot{\Omega}^T = A_0\Omega^T + B(y_1, u)^T \tag{C.12}$$

where $\xi \in R^{2n}$ and $\Omega^T \in R^{2n\times(2n+1)}$ are the outputs of the filters and $\lambda = [\lambda_1\ \lambda_2 \ldots \lambda_{2n}]^T \in R^{2n}$ are parameters chosen to make

$$A_0 = A - \lambda c_1^T \in R^{2n\times 2n} \tag{C.13}$$

$$PA_0 + A_0^T P = -I^{2n\times 2n} < 0 \tag{C.14}$$

given $P \in R^{2n\times 2n}$ and $S \in R^{2n\times 2n}$ are symmetric positive definite.

To reduce the order of the filter's, Ω^T is decomposed such that

$$\Omega^T = [v\ \ \Omega_2] \tag{C.15}$$

where $v = [v_1\ v_2 \ldots v_{2n}]^T \in R^{2n}$, $\Omega_2 = [\eta_1\ \eta_2\ \ldots \eta_{2n}] \in R^{2n\times 2n}$, and $\eta_j \in R^{2n}$ $(j = 1,\ 2\ \ldots 2n)$.

With v and Ω_2 defined, we have

$$\dot{v} = A_0 v + c_{2n} u \tag{C.16}$$

$$\dot{\Omega}_2 = A_0\Omega_2 - I^{2n} y_1 \tag{C.17}$$

Due to the special structure of A_0 and from equations (C.11) and (C.17), we have

$$\dot{\eta}_{2n} = A_0\eta_{2n} - c_{2n} y_1 \tag{C.18}$$

$$\eta_j = A_0^{2n-j}\eta_{2n} \tag{C.19}$$

$$\xi = A_0^{2n}\eta_{2n} \tag{C.20}$$

Let the unknown state y be estimated by

$$\hat{y} = \xi + \Omega^T\theta \tag{C.21}$$

Accordingly, the state estimation error $\epsilon = [\epsilon_1 \ \epsilon_2 \ \dots \epsilon_n]^T = y - \hat{y}$ follows

$$\dot{\epsilon} = A_0 \epsilon \tag{C.22}$$

Based on equations (C.14) and (C.22), the derivative of $V_\epsilon = \epsilon^T P \epsilon$ with respect to time t is given by

$$\dot{V}_\epsilon = -\|\epsilon\|^2 \tag{C.23}$$

From equations (C.20),(C.21), (C.6) and (C.12), we have

$$\begin{aligned}
\dot{y}_1 &= c_2^T A_0^{2n} \eta_{2n} + w^T \theta + \epsilon_2 = c_2^T A_0^{2n} \eta_{2n} + d_{2n} v_2 + \overline{w}^T \theta + \epsilon_2 && \text{(C.24)}\\
\dot{v}_2 &= v_3 - \lambda_2 v_1 && \text{(C.25)}\\
\dot{v}_i &= v_{i+1} - \lambda_i v_1 \ (i = 3, 4 \dots 2n-1) && \text{(C.26)}\\
\dot{v}_{2n} &= -\lambda_{2n} v_1 + u && \text{(C.27)}
\end{aligned}$$

where

$$\begin{aligned}
w &= [v_2 \ \ \eta_{2n}^T A_\eta^T - y_1 c_1^T]^T && \text{(C.28)}\\
\overline{w} &= [0 \ \ \eta_{2n}^T A_\eta^T - y_1 c_1^T]^T && \text{(C.29)}\\
A_\eta &= [(A_0^{2n-1})^T c_2 \dots A_0^T c_2 \ \ c_2]^T && \text{(C.30)}
\end{aligned}$$

Equations (C.24) to (C.27) represent a transformed dynamic system with the measurable v and y being its states. For controller design with backstepping method, the following variables are also needed,

$$\begin{aligned}
z_1 &= y_1 && \text{(C.31)}\\
z_i &= v_i - \alpha_{i-1} \ \ i \geq 2 && \text{(C.32)}\\
z &= [z_1 \ z_2 \dots z_{2n}]^T && \text{(C.33)}
\end{aligned}$$

where α_i is the so called *stabilization function* to be determined.

The backstepping design involves $2n$ steps. In each step, a *stabilizing function*: α_i, and a *tuning function*: τ_i, are generated. The control input u is derived in the last step $2n$.

Step 1. From equations (C.24), (C.31) and (C.32), we have

$$\dot{z}_1 = d_{2n}\alpha_1 + c_2^T A_0^{2n}\eta_{2n} + \overline{w}^T\theta + d_{2n}z_2 + \epsilon_2 \tag{C.34}$$

Letting

$$\alpha_1 = \hat{d}\overline{\alpha}_1 \tag{C.35}$$

and substituting it into equation (C.34), we have

$$\dot{z}_1 = \overline{\alpha}_1 + c_2^T A_0^{2n}\eta_{2n} + \overline{w}^T\theta - d_{2n}(\tilde{d}\overline{\alpha}_1 - z_2) + \epsilon_2 \tag{C.36}$$

where $\hat{d}$ is the estimate of $1/d_{2n}$.

Consider a Lyapunov function candidate

$$V_1 = \frac{1}{2}z_1^2 + \frac{1}{2}\tilde{\theta}^T\Gamma^{-1}\tilde{\theta} + \frac{d_{2n}}{2\gamma}\tilde{d}^2 + V_\epsilon \tag{C.37}$$

where $\Gamma > 0$, $\gamma > 0$ are the gain matrix and gain respectively, and V_ϵ is defined in equation (C.23). Note that $d_{2n} > 0$ by definition.

The derivative of V_1 with respect to time t along the solution of (C.36) is rendered as

$$\dot{V}_1 \le -\zeta_1 z_1^2 + c_1^T\hat{\theta}z_1z_2 + \tilde{\theta}^T(\tau_1 - \Gamma^{-1}\dot{\hat{\theta}}) \tag{C.38}$$

by choosing

$$\overline{\alpha}_1 = -(\zeta_1 + \frac{1}{2})z_1 - c_2^T A_0^{2n}\eta_{2n} - \overline{w}^T\hat{\theta} \tag{C.39}$$

$$\dot{\hat{d}} = -\gamma\overline{\alpha}_1 z_1 \tag{C.40}$$

$$\tau_1 = w - \hat{d}\overline{\alpha}_1 z_1 \tag{C.41}$$

where $\zeta_1 > 0$ is a control parameter and $\hat{\theta}$ is the estimate of parameters θ.

Step 2. From equations (C.25), (C.32) and (C.35), we have

$$\dot{z}_2 = \alpha_2 + z_3 - \gamma_2(w^T\tilde{\theta} + \epsilon_2) - \hat{d}\frac{\partial\overline{\alpha}_1}{\partial\hat{\theta}}\dot{\hat{\theta}} - \beta_2 \tag{C.42}$$

$$\gamma_2 = \hat{d}\frac{\partial\overline{\alpha}_1}{\partial y_1} \tag{C.43}$$

$$\beta_2 = \lambda_2 v_1 + \hat{d}\frac{\partial\overline{\alpha}_1}{\partial\eta_{2n}}(A_0\eta_{2n} - c_{2n}y_1) - \gamma\overline{\alpha}_1^2 z_1 + \gamma_2(c_2^T A_0^{2n}\eta_{2n} + w^T\hat{\theta}) \tag{C.44}$$

Consider the following Lyapunov function candidate

$$V_2 = V_1 + \frac{1}{2}z_2^2 + V_\epsilon \tag{C.45}$$

Differentiating V_2 with respect to time t along the solutions of (C.36) and (C.42), we have

$$\dot{V}_2 \leq -\zeta_1 z_1^2 + z_2 z_3 + \tilde{\theta}^T(\tau_2 - \Gamma^{-1}\dot{\hat{\theta}}) + z_2(\alpha_2 + c_1^T\hat{\theta}z_1 - \beta_2 - \hat{d}\frac{\partial\overline{\alpha}_1}{\partial\hat{\theta}}\dot{\hat{\theta}}) - \gamma_2 z_2 \epsilon_2 - \|\epsilon\|^2 \tag{C.46}$$

where $\tau_2 = \tau_1 - \gamma_2 w z_2$.

If we select

$$\alpha_2 = -(\zeta_2 + \frac{\gamma_2^2}{4})z_2 - c_1^T\hat{\theta}z_1 + \beta_2 + \hat{d}\frac{\partial\overline{\alpha}_1}{\partial\hat{\theta}}\Gamma\tau_2 \tag{C.47}$$

where $\zeta_2 > 0$, it follow that

$$\dot{V}_2 \leq -\zeta_1 z_1^2 - \zeta_2 z_2^2 + z_2 z_3 + \tilde{\theta}^T(\tau_2 - \Gamma^{-1}\dot{\hat{\theta}}) + z_2\hat{d}\frac{\partial\overline{\alpha}_1}{\partial\hat{\theta}}(\Gamma\tau_2 - \dot{\hat{\theta}}) \tag{C.48}$$

Step i ($3 \leq i \leq 2n-1$) Assuming that stabilizing functions $\alpha_1,\ \alpha_2,\ \ldots\alpha_{i-1}$ and tuning functions $\tau_1,\ \tau_2, \ldots \tau_{i-1}$ are derived in previous steps. Choose a Lyapunov function candidate such that

$$V_i = V_{i-1} + \frac{1}{2}z_i^2 + V_\epsilon \tag{C.49}$$

Following the same procedure as in previous steps, the derivate of V_i with respective to time t is rendered as

$$\dot{V}_i \ \leq\ -\Sigma_{j=1}^i \zeta_j z_j^2 + z_i z_{i+1} + \tilde{\theta}^T(\tau_i - \Gamma^{-1}\dot{\hat{\theta}}) + \Sigma_{j=2}^i z_j \frac{\partial\alpha_{j-1}}{\partial\hat{\theta}}(\Gamma\tau_i - \dot{\hat{\theta}})$$

by selecting

$$\alpha_i = -(\zeta_i + \frac{\gamma_i^2}{4})z_i - z_{i-1} + \beta_i + \frac{\partial\alpha_{i-1}}{\partial\hat{\theta}}\Gamma\tau_i - \Sigma_{j=2}^{i-1} z_j \frac{\partial\alpha_{j-1}}{\partial\hat{\theta}}\Gamma\gamma_i w$$

where $\zeta_i > 0$ and

$$\begin{aligned}
\gamma_i &= \frac{\partial\alpha_{i-1}}{\partial y_1} \\
\tau_i &= \tau_{i-1} - \gamma_i w z_i \\
\beta_i &= \Sigma_{j=1}^{i-1}\frac{\partial\alpha_{i-1}}{\partial v_j}(v_{j+1} - \lambda_j v_1) + \gamma_i(w^T\hat{\theta} + c_2^T A_0^{2n}\eta_{2n}) \\
&\quad + \lambda_i v_1 + \frac{\partial\alpha_{i-1}}{\partial\eta_{2n}}(A_0\eta_{2n} - c_{2n}y_1) - \frac{\partial\alpha_{i-1}}{\partial\hat{d}}\gamma\overline{\alpha}_1 z_1
\end{aligned}$$

Step 2n. At step 2n, the control input is determined in the same way as that used to determine α_i in the previous steps such that

$$u = \alpha_{2n} = -\zeta_{2n} z_{2n} - z_{2n-1} + \beta_{2n} + \frac{\partial \alpha_{2n-1}}{\partial \hat{\theta}} \Gamma \tau_{2n} - \Sigma_{j=2}^{2n-1} z_j \frac{\partial \alpha_{j-1}}{\partial \hat{\theta}} \Gamma \gamma_{2n} w,\ \zeta_{2n} > 0 \quad (C.50)$$

With control input u in equation (C.50), parameter updating laws for $\hat{d}$ and $\hat{\theta}$ in equations (C.40) and (C.56) respectively, and α_i and $\dot{\hat{\theta}} - \Gamma\tau_i$ in each step, the close loop system with state vector $[z_1\ z_2\ \ldots z_{2n}]^T$ is derived such that

$$\dot{z}_1 = -(\zeta_1 + \frac{1}{4}) z_1 + c_1^T \hat{\theta} z_2 + \epsilon_2 + (w - \hat{d}\overline{\alpha}_1 c_1)^T \tilde{\theta} - d_{2n} \overline{\alpha}_1(\eta_{2n},\ y_1,\ \hat{\theta}) \tilde{d} \quad (C.51)$$

$$\dot{z}_2 = -c_1^T \hat{\theta} z_1 - (\zeta_2 + \frac{\gamma_2^2}{4}) z_2 + z_3 + \hat{d} \frac{\partial \overline{\alpha}_1}{\partial \hat{\theta}} \Sigma_{j=3}^{2n} \Gamma \gamma_j w z_j - \gamma_2 (w^T \tilde{\theta} + \epsilon_2) \quad (C.52)$$

$$\begin{aligned}\dot{z}_i = & -\Sigma_{j=2}^{i-2} \frac{\partial \alpha_{j-1}}{\partial \hat{\theta}} \Gamma \gamma_i w z_j - (1 + \frac{\partial \alpha_{i-2}}{\partial \hat{\theta}} \Gamma \gamma_i w) z_{i-1} - (\zeta_i + \frac{\gamma_i^2}{4}) z_i + z_{i+1} \\ & + \frac{\partial \alpha_{i-1}}{\partial \hat{\theta}} \Sigma_{j=i+1}^{2n} \Gamma \gamma_j w z_j - \gamma_i (w^T \tilde{\theta} + \epsilon_2) \quad 3 \le i \le 2n-1 \end{aligned} \quad (C.53)$$

$$\begin{aligned}\dot{z}_{2n} = & -\Sigma_{j=2}^{2n-2} \frac{\partial \alpha_{j-1}}{\partial \hat{\theta}} \Gamma \gamma_{2n} w z_j - (1 + \frac{\partial \alpha_{2n-2}}{\partial \hat{\theta}} \Gamma \gamma_{2n} w) z_{2n-1} \\ & -(\zeta_{2n} + \frac{\gamma_{2n}^2}{4}) z_{2n} - \gamma_{2n} (w^T \tilde{\theta} + \epsilon_2) \end{aligned} \quad (C.54)$$

Choosing a Lyapunov function candidate

$$V_{2n} = V_{2n-1} + \frac{1}{2} z_{2n}^2 + V_\epsilon = \frac{1}{2} z^T z + \frac{1}{2} \tilde{\theta}^T \Gamma^{-1} \tilde{\theta} + \frac{d_{2n}}{2\gamma} \tilde{d}^2 + 2n V_\epsilon \quad (C.55)$$

and differentiating it with respective to time t along the solutions of (C.51) to (C.54), we have

$$\dot{V}_{2n} \le -\Sigma_{j=1}^{2n} \zeta_j z_j^2 + \tilde{\theta}^T (\tau_{2n} - \Gamma^{-1} \dot{\hat{\theta}}) + \Sigma_{j=2}^{2n} z_j \frac{\partial \alpha_{j-1}}{\partial \hat{\theta}} (\Gamma \tau_{2n} - \dot{\hat{\theta}})$$

where $\tau_{2n} = \tau_1 - \Sigma_{j=2}^{2n} r_j w z_j$ and $\gamma_1 = -1$ is a new constant introduced to keep the consistency in expression.

Letting

$$\dot{\hat{\theta}} = \Gamma \tau_{2n} \quad (C.56)$$

It follows that

$$\dot{V}_{2n} \le -\Sigma_{j=1}^{2n} \zeta_j z_j^2 \le 0$$

From equation (C.2), V_{2n} is non-increasing, $\tilde{\theta}$, $\tilde{d}$ and ϵ are all bounded. Based on LaSalle-Yoshizawa theorem [48], $z \rightarrow 0$ when $t \rightarrow \infty$. Obviously $y_1 \rightarrow 0$ when $t \rightarrow \infty$. **Q.E.D**

The above results can be summarized in the following theorem.

Theorem C.2.1 *For the chained multiple mass-spring-damper system (C.6) and the re-constructed dynamic model represented by equations (C.24) to (C.27), the regulation of the position y_1 is achieved ($y_1 \rightarrow 0$ when $t \rightarrow \infty$) under the control law (C.50) and the parameter adaptation laws (C.40) and (C.56).*

Remark C.2.1 *The above design procedure is mostly the same as that in [48], though there are some differences in selection of control parameters.*

Remark C.2.2 *The CMMSD system considered is assumed to be free of external disturbances. To keep the robustness of the controlled system under the external disturbances, various robustification approaches can be used, such as dead-zone modification or δ-modification [78][79], though the resulting controllers tend to be more complicated. As pointed out in [78] and [80], the adaptive controller developed with backstepping methods shows much higher degree of robustness than that of conventional adaptive controller even in the absence of robustification mechanisms.*

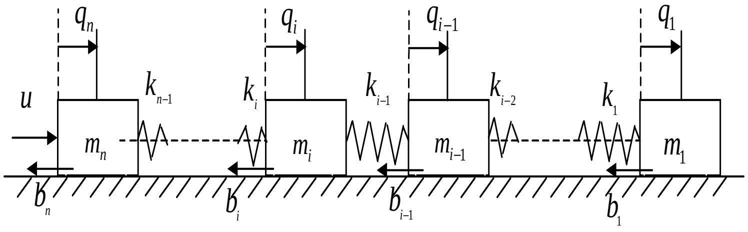

Figure C.1: General Chained Multiple Mass Spring System

Appendix D

Proof of Lemma 6.2.1

The proof of the boundedness of Γ^{-1} follows that in [64] and is produced below for the completeness of the presentation.

Substituting the forgetting factor in equation (6.106) into equation (6.104), we have

$$\dot{\Gamma} = -\gamma_0\Gamma + \frac{\gamma_0}{k_0}\|\Gamma^{-1}\|\Gamma + 2\gamma_2 W^T W \tag{D.1}$$

Solving $\Gamma(t)$ from equation (D.1), it leads to

$$\Gamma(t) = \Gamma(0)e^{-\gamma_0 t} + \int_0^t e^{-\gamma_0(t-s)}(\frac{\gamma_0}{k_0}\|\Gamma^{-1}\|\Gamma + 2\gamma_2 W^T W)ds \tag{D.2}$$

Noting that $\|\Gamma^{-1}\|\Gamma > I$ where I is an identity matrix with the same dimension of that of Γ, we have

$$\int_0^t e^{-\gamma_0(t-s)}\frac{\gamma_0}{k_0}\|\Gamma^{-1}\|\Gamma ds \leq k_0^{-1}I\int_0^t e^{-\gamma_0(t-s)}\gamma_0 ds \leq k_0^{-1}I(1-e^{-\gamma_0 t}) \tag{D.3}$$

From equations (D.2) and (D.3), we have

$$\Gamma(t) \geq (\Gamma(0) - k_0^{-1}I)e^{-\gamma_0 t} + k_0^{-1}I + 2\int_0^t e^{-\gamma_0(t-s)}\gamma_2 W^T W)ds \tag{D.4}$$

As $\Gamma^{-1}(0) \leq k_0 I$, thus $\Gamma(t) > 0$ and

$$\Gamma(t) \geq k_0^{-1}I, \quad for\ \gamma_2 > 0 \tag{D.5}$$

which is equivalent to $\Gamma^{-1}(t) \leq k_0 I$ or $\Gamma^{-1}(t)$ is bounded. As $\Gamma^{-1}(t) \leq k_0 I$, it follows that $\gamma(t) \geq 0$ from equation (6.106).

If $W(q_l, \dot{q}_l)$ is persistently exciting, that is, for a positive constants T and α_1

$$\int_t^{t+T} W^T(q_l, \dot{q}_l) W(q_l, \dot{q}_l) ds \geq \alpha_1 I, \quad \forall t \geq 0 \tag{D.6}$$

it can be proved that Γ^{-1} is uniformly lower bounded.

From equations (D.4) and (D.6), it follows that given $t \geq \delta$

$$\begin{aligned} \Gamma(t) &\geq (k_0^{-1} + 2\gamma_2 e^{-\gamma_0 \delta \alpha_1} I & \text{(D.7)} \\ \Gamma^{-1}(t) &\leq k_0(1 + 2k_0 \alpha_1 \gamma_2 e^{-\gamma_0 \delta}) I & \text{(D.8)} \end{aligned}$$

From the definition of $\gamma(t)$ in equation (6.106), we have

$$\gamma(t) \geq (1 + 2k_0 \alpha_1 \gamma_2 e^{-\gamma_0 \delta})^{-1} (2\gamma_0 k_0 \alpha_1 \gamma_2 e^{-\gamma_0 \delta}) \tag{D.9}$$

and thus $\gamma(t)$ is lower bounded.

From equation (D.1), $\Gamma(t)$ can be written as

$$\Gamma(t) = \Gamma(0) exp(-\int_0^t \gamma(s) ds) + 2\gamma_2 \int_0^t exp(-\int_s^t \gamma(v) dv) W^T(s) W(s) ds \tag{D.10}$$

From equations (D.9) and (D.10), we have

$$\Gamma(t) \leq \Gamma(0) + 2\gamma_2 \in_0^t e^{-\gamma_1 (t-s)} W^T(s) W(s) ds \tag{D.11}$$

As the second term of the right-hand side of equation (D.11) is the output M of the stable filter

$$\dot{M} + \gamma_1 M = W^T W \tag{D.12}$$

and W is bounded, M is bounded. From equations (D.8) and (D.12), $\Gamma^{-1}(t)$ is upper and lower bounded uniformly.

Bibliography

[1] M. H. Raibert and J. J. Craig, "Hybrid position/force control of manipulators," *ASME Journal of Dynamic System, Measurement and Control*, Vol 102, No.2, pp.126–133, 1981.

[2] T. Yoshikawa, "Dynamic hybrid position/force control of robot manipulators - description of hand constraints and calculation of joint driving force," *IEEE Journal of Robotics and Auto*, RA-3(5), pp. 386-392, 1987.

[3] O. Khatib, "A unified approach for motion and force control of robot manipulators: the operational space formulation," *IEEE J. of Robotics and Auto.*, RA-3, pp. 43–53, 1987.

[4] N. Hogan, "Impedance control, an approach to manipulation: Part I–III," *Journal of Dynamic Systems*, Measurement, and Control, Vol 107, pp. 1–24, 1985.

[5] H. Seraji and R. Colbaugh, "Force tracking in impedance control," *The Int. J of Robotics Research*, Vol 16, No 1, pp. 97–117, 1997.

[6] R. Anderson and M. W. Spong, "Hybrid impedance control of robotic manipulators," *Proc. of the 1987 IEEE Conf. on Robotics and Auto.*, pp. 1073-1080, 1987.

[7] N. H. McClamroch and D. W. Wang, "Feedback stabilization and tracking of constrained robots," *IEEE Trans. on Auto . Contr.*, Vol.33, No.5, pp.419–426, 1988.

[8] J. K. Mills and A. A. Goldenberg, “Force and Position Control of Manipulators During Constrained Motion Tasks,” *IEEE Trans. on Robotics and Auto*, Vol 5, No 1, pp.30–46, 1989.

[9] R. K. Kankaanranta and H. N. Koivo, “Dynamics and Simulation of Compliant Motion of a Manipulator,” *IEEE J. of Robotics and Auto*, Vol 4, No 2, pp. 163–173, 1988.

[10] D. Botturi, A. Castellani, D. Moschini and P. Fiorini, “Peformance evaluation of task control in teleoperation ”, *Proc. of 2004 IEEE Conf. on Robotics and Auto*, New Orleans, pp. 3690-3695, 2004.

[11] J. Kovecses, J.-C. Piedboeuf and C. Lange, “Dynamics modeling and simulation of constrained robotic systems”, *IEEE/ASME Trans. on Mechatronics*, Vol. 8, No. 2, pp. 165–177, 2003.

[12] J. J. E. Slotine and W. Li, “Composite adaptive control of robot manipulators,” *Automatica*, Vol 25, No 4, pp. 509-519, 1989.

[13] J. J. E. Slotine and W. Li, “On the adaptive control of robot manipulators,” *Int. J. of Robotics Research*, Vol 6, No 3, pp 147-157, 1987.

[14] J. Yuan, “Composite adaptive control of constrained robots,” *IEEE Trans. on Robotics and Auto.*, Vol 12, No 4, pp.640-645, 1996.

[15] W.-S. Lu and Q.-H. Meng, “Impedance control with adaptation for robotic manipulators,” *IEEE Trans. on Robotics and Auto.*, Vol 7, No 3, pp. 408–415, 1991.

[16] B. Yao and M. Tomizuka, “Adaptive control of robot manipulators in constrained motion - controller design,” *ASME J. Dynamic Sys. Measurement and Contr.*, Vol 117, pp. 320-328, 1995.

[17] B. Yao and M. Tomizuka, ”Adaptive robust motion and force tracking control of robot manipulators in contact with compliant surfaces with unknown stiffness”, *ASME Trans. On Dynamic Sys., Measurement and Control*, Vol 120, pp. 232-240, 1998.

[18] C.C. Cheah, S. Kawamurab and S. Arimotob, “Stability of hybrid position and force control for robotic manipulator with kinematics and dynamics uncertainties”, *Automatica*, 39(2003), pp.847–855, 2003.

[19] C.-Y. Su, and Y. Stepanenko, “Hybrid adaptive/robust motion control of rigid-link electrically-driven robot manipulators,” *IEEE Trans. on Robotics and Automation*, No. 3, Vol. 11, pp. 426–432, 1995.

[20] B. Yao, S. P. Chan and D. W. Wang, “Unified formulation of variable structure control schemes for robot manipulators,” *IEEE Trans. on Auto. Contr.*, Vol 39, No.2, pp.371–376, 1994.

[21] Z. Lu, S. Kawamura and A. A. Goldenberg, “An approach to sliding-based impedance control,” *IEEE Trans. on Robotics and Auto.*, Vol 11, No 5, pp. 754–759, 1995.

[22] S. P. Chan and H. C. Liaw, “Robust generalized impedance control of robotics for compliant manipulation,” *Int. J. of Robotics and Automation*, Vol 12, pp. 146–155, 1997.

[23] B. Yao, S. P. Chan and D. Wang, ”VSC motion and force control of robot manipulators in the presence of environment constraint uncertainties”, *J. of Robotic Systems*, 11(6), pp. 503-515, 1994.

[24] S. S. Ge, C. C. Hang, L. C. Woon and X. Q. Chen, “Impedance control of robot manipulators using adaptive neural networks,” *International Journal of Intelligent Control and Systems*, Vol.2, No.3, pp.433-452, 1998.

[25] R. Ozawa and H. Kobayashi, “A new impedance control concept for elastic joint robots,”, *Proc. of 2003 IEEE Int. Conf. on Robotics and Auto.*, Taiwan, pp.3126-3131, 2003.

[26] S. S. Ge, L. Huang and T. H. Lee, “Model-based and neural-network-based adaptive control of two robotic arms manipulating an object with relative motion,” *Int. J. of Systems Sci.*, Vol 32, No 1, pp. 9–23, 2001.

[27] N. H. McClamroch, “A singular perturbation approach to modeling and control of manipulators constrained by a stiff environment,” *Proc. 28th Conf. on Decision and Control*, pp. 2407–2411, 1989.

[28] H. Seraji, “Nonlinear and adaptive control of force and compliance in manipulators,” *Int. J. of Robotics Research*, Vol 17, No 5, pp 467–484, 1998.

[29] L. Villani, C. Canudas de Wit and B. Brogliato, “An exponentially stable adaptive control for force and position tracking of robot manipulators,” *IEEE Trans. on Auto. Control*, Vol 44, No 4, pp 798–802, 1999.

[30] J. K. Mills and C. V. Nguyen, “Robotic manipulator collisions: modeling and simulation,” *ASME J. of Dynamic Systems, Measurement, and Control*, Vol 114, pp. 650–659, 1992.

[31] J. De Schutter, D. Torfd, H. Bruyninckx and S. Dutre, “Invariant hybrid force/position control of a velocity controlled robot with compliant end effector using modal decoupling,” *Int. J, of Robotics Research*, Vol 16, No 3, pp. 340–356, 1997.

[32] J. K. Mills and D. M. Lokhorst, “Control of robotic manipulators during general task execution: a discontinuous control approach,” *The Int. J. of Robotics Research*, Vol 12, No 2, pp. 146–163, 1993.

[33] J. K. Mills and D. M. Lokhorst, “Stability and Control of Robotic Manipulators During Contact/Noncontact Task Transition,” *IEEE Trans. on Robotics and Automation*, Vol 9, No 3, pp. 335–345, 1993.

[34] R. Featherstone,“Modeling and control of contact between constrained rigid bodies,” *IEEE Trans. on Robotics and Auto.*, Vol. 20, No. 1, pp. 82–92. 2004

[35] S. Arimoto, F. Miyazaki, and S. Kawamura, “Cooperative motion control of multi-robot arms or fingers,” *Proc. IEEE Int. Conf. on Robotics and Auto.*, pp. 1407–1412, 1987.

[36] P. Hsu, “Coordinated control of multiple manipulator systems,” *IEEE Trans. on Robotics and Auto.*, Vol. 9, No. 4, pp. 400–410, 1993.

[37] M. A. Unseren, "A rigid body model and decoupled control architecture for two manipulators holding a complex object," *Robotics and Autonomous Systems*, No.10, pp. 115–131, 1992.

[38] S. S. Ge, X. Q. Chen, S. Xie and D. L. Gu, "Motion and force control of a Cartesian arm and a rotary table," *Proc. of the IEEE Singapore International Symposium on Control Theory and Applications (SISCTA'97)*, Singapore, pp.286-289. 1997.

[39] C. Canudas de Wit, B. Siciliano and G. Bastin(Eds), *Theory of Robot Control*, Springer, 1996.

[40] B. Brogliato and P. Orhant, "On the transition phase in robotics: Impact models, dynamics and Control," *Proc. 1994 Int. Conf. on Robotics and Automations*, San Diego, CA, pp. 346–351, 1994.

[41] H.-P. Huang and S.-S. Chen, "Compliant motion control of robots by using variable impedance," *Int. J. Adv. Manuf. Technol.*, Vol 7, pp. 322–332, 1992.

[42] K. Ogata, *Modern control engineering*, Prentice-Hall,1995.

[43] G. Zhu, S. S. Ge and T. H. Lee, "Simulation studies of tip tracking control of a single-link flexible robot based on a lumped model," *Robotica*, Vol 17, pp.71-78, 1999.

[44] Y. F. Zheng and H. Hemami, "Mathematical modeling of a robot collision with its environment," *J. of Robotic Systems*, Vol 2, No 3, pp 289–307, 1985.

[45] Y. Wang and R. P. Paul, "Modeling impact dynamics for robotic operations," *Proc. 1988 IEEE Int. Conf on Robotics and Automation*, pp 678–683, 1988.

[46] S. C. Wu, S. M. Yang and E. J. Haug, "Dynamics of mechanical systems with coulomb friction, stiction, impact and constraint addition–deletion II (III)," *Mechanism and Machine Theory*, Vol 21, No 5, pp 407–425, 1986.

[47] N. Sadegh and R. Horowitz, "Stability and robustness analysis of a class of adaptive controller for robotic manipulators," *Int. J. of Robotics Research*, Vol 9, No 3, pp 74-92, 1990.

[48] M. Krstic, I. Kanellakopoulos and P. Kokotovic, *Nonlinear and adaptive control design*, New York, John Wiley, 1995.

[49] K. J. Astrom and B. Wittenmark, *Adaptive Control*, Addison-Wesley, 1989.

[50] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems*, Prentice Hall, 1989.

[51] N. Hogan, "On the stability of manipulators performing contact task," *IEEE J. of Robot. Automat.*, 4(6), pp. 677–686, 1988.

[52] J. S. Bay, *Constrained Motion of a 3-D Manipulator Over Unknown Constraint: The Robotic Groping Problem*, Ph.D dissertation, The Ohio State University, USA, 1988.

[53] J. S. Bay and H. Hemami, "Localization of Hybrid Controllers for Manipulation on Unknown Constraints," *J. of Intelligent and Robotic Systems*, 7:301–320, 1993.

[54] T. Yoshikawa and A. Sudou, "Dynamic hybrid position/force control of robot manipulators: on-line estimation of unknown constraint," *Proceedings of the 1990 Int. Conf. on Robotics and Auto*, pp. 1231–1236, 1990.

[55] D. Wang, Y. C. Soh, Y. K. Ho and P. Muller, "Global stabilization for constrained robot motions with constraint uncertainties," *Robotica*, Vol 16, pp. 171–179, 1998.

[56] P. Kazanzides, N. Scott Bradley and W. A Wolovich, "Dual-drive Force/Velocity Control: Implementation and Experiment Results," *Proceedings of the 1989 IEEE Int Conf on Robotics and Auto.*, pp. 92–97, 1989.

[57] H. K. Khalil, *Nonlinear Systems*, Prentice Hall, NJ, 1996.

[58] J. J. E. Slotine, "Sliding controller design for nonlinear systems," *Int. J. Contr.*, Vol 40, No 2, pp. 421–434, 1984.

[59] Feng-Yi Hsu and Li-Chen Fu, “Intelligent Robot Deburring Using Adaptive Fuzzy Hybrid Position/Force Control,” *IEEE Trans. on Robotics and Automation*, Vol 16, No 4, pp. 325–335, 2000.

[60] K. Kiguchi and T. Fukuda, “Position/Force Control of Robot Manipulators for Geometrically Unknown Objects Using Fuzzy Neural Networks,” *IEEE Trans. on Industrial Electronics*, Vol 47, No 3, pp. 641–649, 2000.

[61] M. Margaliot and G. Langholz, “Fuzzy Lyapunov-based approach to the design of fuzzy controllers,” *Fuzzy Sets and Systems*, Vol. 106, pp. 49–59, 1999.

[62] C. Desoer and M. Vidyasagar, *Feedback systems: input-output properties*, New York: Academic, 1975.

[63] S. S. Ge, T. H. Lee and C. J. Harris, *Adaptive Neural Network Control of Robotic Manipulators*, World Scientific, Singapore, 1998.

[64] S. S. Ge, “Adaptive controller design for flexible joint manipulators,” *Automatica*, Vol.32, No.2, pp.273-278, 1996.

[65] S. S. Ge, “Non-linear adaptive control of robots including motor dynamics, Part I: J. of Systems and Control Engineering”, *Proc. Inst. Mech Engrs*, Vol 209, pp. 89-99. 1994.

[66] S. Haykin, *Neural Networks – A Comprehensive Foundation*, Macmillan College, USA, 1994.

[67] R. Murray, Z. Li and S. Sastry, *A Mathematical Introduction to robotic manipulation*, CRC Press, USA. 1993.

[68] A. K. Ramadorai, T. J. Tarn and A. K. Bejczy “Task definition, decoupling and redundancy resolution by nonlinear feedback in multi-robot object handling,” *Robotic Lab. Report*, SSM–RL–92-16, Washington University, 1992.

[69] L. Sciavicco and B. Siciliano, *Modeling and control of robot manipulators*, McGraw-Hill, USA. 1996.

[70] Y. F. Zheng and H. Hemami, "Mathematical modeling of a robot collision with its environment," *J. of Robotic Systems*, Vol 2, No 3, pp 289–307, 1985.

[71] S. Hara and F. Matsuno, "System theoretic approach to dynamics of hyper-redundant mechanical systems," *Proceedings of TITech COE/Super Mechano-Systems Workshop'99*, Tokyo, pp.73-82, 1999.

[72] G. Zhu, S. S. Ge and T. H. Lee, "Simulation studies of tip tracking control of a single-link flexible robot based on a lumped model," *Robotica*, Vol 17, pp.71-78, 1999.

[73] R. Volpe and P. Khosla, "A theoretical and experimental investigation of impact control for manipulators," *The Int. J. of Robotics Research*, Vol 12, No 4, pp.351-365, 1993.

[74] L. B. Gutierrez, F. L. Lewis and J. Andy Lowe, "Implementation of neural network tracking controller for a single flexible link: comparison with PD and PID controller," *IEEE Trans. on Industrial Electronics*, Vol 45, No 2, pp.307–318, 1998.

[75] S. Eppinger and W. Seering, "Understanding bandwidth limitations on robot force control," *Proceedings of IEEE Conf. on Robotics and Auto.*, Raleigh, pp.904-909, 1987.

[76] S. Q. Zhu, S. Commuri and F. L. Lewis, "A singular perturbation approach to stabilization of the internal dynamics of multilink flexible robots," *Proceedings of the American Control Conf.*, Ballimors, pp. 1386–1390, 1994.

[77] C. Rohrs, L. Valavani, M. Athans and G. Stein, "Robustness of continuous-time adaptive control algorithms in the presence of unmodeled dynamics," *IEEE Trans. on Auto. Control*, Vol 30, pp.881-889, 1985.

[78] F. Ikhouane and M. Krstic, "Robustness of the tuning functions adaptive backstepping design for linear systems," *IEEE Trans. on Auto. Contr.*, Vol 43, No 3, pp. 431-437, 1998.

[79] Z. Ding, “Analysis and design of robust adaptive control for nonlinear output feedback systems under disturbances with unknown bounds,” *IEE Proc. Control Theory*, Vol 147, No 6, 2000.

[80] Z. H. Li, C. Wen and C. B. Soh, “Robustness of Kristic’s new adaptive control scheme,” *Proc. IFAC Sym. Nonlinear Contr. Syst. Design*, Tahoe City, CA, 1995.

[81] M. W. Spong, “Modeling and control of elastic joint robots,” *ASME J. Dyn. Syst., Meas. Contr.*, Vol 109, pp. 310-319, 1987.

[82] M. W. Spong, K. Khorasani and P. V. Kokotovic, “An integral manifold approach to the feedback control of flexible joint robots,” *IEEE J. Robot Automat.*, Vol RA-3, pp. 291-300, 1987.

[83] K. Khorasani, “Adaptive control of flexible joint robots,” *Proc. IEEE Int Conf Robot. Automat.*, pp 2127-2134, 1991.

[84] S. S. Ge and I. Postlethwaute, “Adaptive neural network controller design for flexible joint robots using singular perturbation technique,” *Trans Int MC*, Vol 17, No 3, pp. 120-131, 1995.

[85] C. Ott, A. Albu-schaffer and G. Hirzinger, “Comparison of adaptive and non-adaptive control laws for a flexible joint manipulator,” *Proc. 2002 IEEE/RSJ Int Conf. on Intelligent Robots and Systems*, pp 2018 -2024, 2002.

[86] A. De Luca, A. Isidori and F. Nicolo, “Control of robot arm with elastic joints via nonlinear dynamic feedback,” *Proc. 24th IEEE Conf. on Decision and Control*, Ft. Landerdale, pp. 1671-1679, 1985.

[87] K. Khorasani, “Nonlinear feedback control of flexible joint manipulators: a single link case study,” *IEEE Trans. on Auto. Control*, Vol 35, pp. 1145-1149, 1990.

[88] J. Yuan and Y. Stepenanko, “Composite adaptive control of flexible joint robots,” *Automatica*, Vol 29, No 3, pp. 609 - 619, 1993.

[89] R. Lozano and B. Brogliato, “Adaptive control of robot manipulators with flexible joints,” *IEEE Trans. Automat. Contr.*, Vol AC-37, pp. 174 -181, 1992.

[90] B. Brogliato and R. Lozano, “Correction to ’ Adaptive control of robot manipulators with flexible joints,” *IEEE Trans. on Auto. Contr.*, Vol 41, No 6, 1996.

[91] L. Tian and A. A. Goldenberg, “Robust adaptive control of flexible joint robots with joint torque feedback,” *Proc. 1995 IEEE Int. Conf. on Robotics and Auto*, pp. 1229-1234, 1995.

[92] J. H. Oh and J. S. Lee, “Control of flexible joint robot system by backstepping design approach,” *Intelligent Auto. and Soft Computing*, Vol 5, No 4, pp. 267 - 278, 1999.

[93] M. W. Spong, “On the force control problem for flexible joint manipulators,” *IEEE Trans. on Auto. Control*, Vol 34, No 1, pp. 107-111, 1989.

[94] J. K. Mills, “Stability and control of elastic-joint robotic manipulators during constrained-motion tasks,” *IEEE Trans. on Robotics and Auto.*, Vol 8, No 1, 1992.

[95] L. Tian and A. A. Goldenberg, “A unified approach to motion and force control of flexible joint robots,” *Proc. 1996 IEEE Int. Conf. on Robotics and Auto*, pp. 1115-1120, 1996.

[96] C. Ott, A. Albu-schaffer, A. Kugi and G. Hirzinger, “Decoupling based Cartesian impedance control of flexible joint robots,” *Proc 2003 IEEE Int Conf. on Robotics and Auto*, 2003.

[97] I. Kao and F. Yang, “Stiffness and contact mechanics for soft fingers in grasping and manipulation”, *IEEE Trans. on Robotics and Auto.*, Vol. 20, No. 1, pp. 132–135, 2004.

[98] S. A. Mascaro and H. H. Asada, “ Measurement of finger posture and three-axis fingertip touch force using fingernail sensors”, *IEEE Trans. on Robotics and Auto.*, Vol. 20, No. 1, pp. 26–35, 2004.

[99] S. S. Ge, "Adaptive control of robots having both dynamical parameter uncertainties and unknown input scalings," *Mechatronics*, Vol 6, No 5, pp. 557 - 569, 1996.

[100] J. Yuan, "Adaptive control of a constrained robot - ensuring zero tracking and zero force errors," *IEEE Trans. on Auto. Contr.*, Vol 42, No 12, pp. 1709-1714, 1997.

[101] J. J. E. Slotine and W. Li, "Applied nonlinear control," Prentice-Hall, Englewood Cliffs, NJ, 1991.

[102] A. Garcia and V. Feliu, "Force control of a single-link flexible robot based on a collision detection mechanism", *IEE Proc.-Control Theory Appl.*, Vol 147, No 6, pp.588–595, 2000.

[103] V. Feliu, J. A. Somolinos and A. Garcia, "Inverse dynamics based control system for three-degree-of-freedom flexible arm", *IEEE Trans. on Robotics and Auto.*, Vol. 19, No. 6, pp.1007–1014. 2003

[104] F. M. C. Ching and D. Wang, "Exact solution and infinite-dimensional stability analysis of a single flexible link in collision", *IEEE Trans. on Robotics and Auto.*, Vol. 19, No. 6, pp. 1015–1020, 2003.

Author's Papers

[1] S. S. Ge, L. Huang and T. H. Lee, "Model-based and neural-network-based adaptive control of two robotic arms manipulating an object with relative motion," *Int. J. of Systems Sci.*, Vol 32, No 1, pp. 9-23, 2001.

[2] L. Huang, S. S. Ge and T. H. Lee, "Fuzzy unidirectional force control of a constrained robot," *Fuzzy Sets and Systems*, Vol 134, pp.135 -146, 2003.

[3] L. Huang, S. S Ge and T. H. Lee, "Neural network based adaptive impedance control of constrained robots," *International Journal of Robotics and Automation*, Vol 19, Issue 3, pp.117-124, 2004.

[4] L. Huang, S. S. Ge and T. H . Lee, "Position control of chained multiple mass-spring-damper systems–adaptive output feedback control approaches," *International Journal of Control, Automation and Systems*, Vol 2, No 2, pp.1-12, 2004.

[5] L. Huang, S. S. Ge and T. H. Lee, "An adaptive impedance control scheme for constrained robots," accepted by *International Journal of Computers, Systems and Signals*, 2004.

[6] S. S. Ge, L. Huang and T. H Lee, "Explicit Force Control of Manipulators Constrained by Uncertain Dynamic Constraints – Adaptive Output Feedback Approaches," conditionally accepted by *International Journal on Mechatronics*, 2004 (under revision).

[**7**] S. S. Ge, L. Huang and T. H. Lee, “Adaptive Control of Coordinated Operation of Two Robotic Manipulators,” *Proceedings of the International Federation of Automatic Control (IFAC) 14TH Word Congress*, Beijing, pp.153 – 158, 1999.

[**8**] L. Huang, S. S. Ge and T. H. Lee, ”Neural network based adaptive impedance control of constrained Robots,” *Proceedings of the 2002 IEEE International Symposium on Intelligent Control*, Vancouver, pp.615-621, 2002,

[**9**] L. Huang, S. S. Ge and T. H. Lee, “Unified control for generalized chained multiple mass-spring-damper systems,” *Proceedings of the 6th Int. Conf. on Auto., Robotics and Computer Vision* , Singapore, 2000.

[**10**] L. Huang, S. S. Ge and T. H. Lee, “Robust adaptive nonlinear feedback control of two robotic manipulator handling an object,” *Proceedings of the 3rd Int. Conf on Industrial Automation*, Montreal, Canada, pp.11.5–11.8, 1999.

[**11**] L. Huang, S. S. Ge and T. H. Lee, “Adaptive impedance control of constrained robots,” *Asia Control Conference’2000*, Shanghai, 2000.

[**12**] L. Huang , S. S. Ge and T. H. Lee, “Adaptive output feedback control for general chained multiple mass-spring-damper systems,” *Proceedings of the 4th Asian Control Conf. on Robotics and its Applications*, Singapore, 2001.

[**13**] L. Huang, S. S. Ge and T. H. Lee, “Position/force control of an uncertain constrained flexible joint robots ,” submitted to *International Journal on Mechatronics*, 2004.