Hybrid vision-force control of robot without calibrations

Zhao, Yu

2007

Zhao, Y. (2007). Hybrid vision-force control of robot without calibrations. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/39086>

<https://doi.org/10.32657/10356/39086>

9529593 129
3x
D/Blue

Hybrid Vision-Force Control of Robot without Calibrations

Zhao Yu

School of Electrical and Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirements for the degree of
Doctor of Philosophy

2007

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

.....

Date

.....

Zhao Yu

To my family.

Acknowledgement

I would like to express my sincere gratitude to my research supervisor, Dr. Cheah Chien Chern, for his guidance, inspiration, and support throughout my research. This work would not have been possible without his excellent guidance, suggestions and encouragement throughout all these years in Nanyang Technological University.

My gratitude to all my friends for their support, encouragement, valuable discussions and sharing of ideas during the progress of this research.

I would like to acknowledge the School of Electrical & Electronic Engineering, Nanyang Technological University, Singapore, for awarding me the Research Scholarship and providing me with excellent research facilities.

Last but not least, I would like to dedicate this thesis to my family who, from the day I came to the world, have provided tremendous support and constant encouragement throughout my educational career.

Summary

Robots have found wide applications in factories and most robot systems are designed and used to perform mainly repetitive operations in a structured environment. When the environment is not well structured, good performance can not be guaranteed. Vision-based control is a promising way to overcome the problems that many controllers currently face because it allows noncontact measurement of the environment. With rapid development in computing and sensor technologies, more sophisticated robots with vision can be designed. Considerable effort has been devoted to the development of vision-based controllers during past one decade. However, most research in vision-based control has focused on free motion control and the studies in vision and force control problems are still far from mature.

This thesis is intended for the development of hybrid vision-force controllers that do not require calibrations. Adaptive vision and force controllers are proposed for robots with uncertainties in the kinematics, manipulator dynamics, camera parameters and constraint surface. For control tasks defined in task space, traditional solutions in the literature resort to using exact kinematics or Jacobian matrix to develop the control schemes. However, when uncertainty exists in robot kinematics and environment, these control methods can not guarantee the system stability or may lead to position and force errors. This thesis is dedicated to the development of control schemes for robot that can deal with uncertainties in kinematics, dynamics, camera model and constraint surface. Toward this objective, a systematic study is

conducted in this thesis and adaptive control schemes are proposed for regulation and tracking control of robot manipulators.

First, the problem of vision-force setpoint control with uncertain kinematics, dynamics and constraint surface is studied. An approximate Jacobian controller based on regressors is firstly proposed to deal with the uncertainties. In many applications, not only the parameters of the constraint surface is unknown, but the structure is uncertain. An adaptive vision-force controller using neural networks is then proposed for robot manipulator with structure uncertainties. Compared with the previous controller, this controller do not require linear parametrization of the uncertain parameters and exact knowledge of the structure of constraint surface.

In some applications, the robots are required to track position and force trajectories. An adaptive Jacobian vision-force tracking controller is proposed to deal with uncertainties in kinematics, dynamics and constraint surface. A neural network Jacobian controller is also proposed to deal with the structure uncertainties in the Jacobian matrices and constraint surface. It is shown that uniformly ultimate boundedness can be guaranteed in the presence of the uncertainties.

Cooperative control of multi-fingered robot hands is another important application which has received increasing attention in recent years. However, today's robot hands are still far less flexible than human hands because robot hands can not adapt to uncertainties in contact points of fingers with the object. In this thesis, the vision based setpoint control problem of multi-fingered robot hands with uncertain kinematics, dynamics, object and camera parameters and contact points is studied. An adaptive setpoint control law using neural network is firstly proposed. Then an adaptive neural-network Jacobian control law is proposed which does not require knowledge of the structure of the Jacobian matrices. It is shown that the stability can be achieved in the presence of uncertainties.

Table of Contents

Acknowledgements i

Summary ii

List of Figures vii

1 Introduction 1

1.1 Literature Survey 2

1.2 Motivation of Research 9

1.3 Contributions of the Thesis 11

1.4 Organization of the Thesis 11

2 Robot Kinematics and Dynamics 13

2.1 Robot Kinematics 13

2.1.1 Direct Kinematics and Jacobian Matrix 13

2.1.1.1 Direct Kinematics 14

2.1.1.2 Jacobian Matrix 14

TABLE OF CONTENTS		v
2.1.2	Image Jacobian Matrix	15
2.2	Robot Dynamics	18
2.2.1	Dynamic Model of Robot Manipulator	19
3 Hybrid Vision-force Setpoint Control for Constrained Robots with Uncertainties		23
3.1	Introduction	23
3.2	Problem Formulation	24
3.3	Vision-force Control of Robots with Uncertainties	25
3.4	Hybrid Vision-force Control for Robot Manipulators using Neural Networks	37
3.4.1	Neural Network	37
3.4.2	Neural Network Control of Constrained Robots with Uncertainties	39
3.5	Simulation Results	46
3.5.1	Vision-force Control using Regressors	48
3.5.2	Vision-force Control using Neural Networks	51
3.6	Conclusion	58
4 Adaptive Jacobian Motion and Force Tracking Control for Constrained Robot with Uncertainties		61
4.1	Adaptive Vision-force Tracking Control	62
4.2	Adaptive Neural-network Vision and Force Tracking Control	74

TABLE OF CONTENTS	vi
4.3 Simulation Results	87
4.3.1 Adaptive Vision-force Tracking Control	87
4.3.2 Adaptive Neural-network Vision and Force Tracking Control .	91
4.4 Conclusion	92
5 Vision-force Cooperative Control of Multi-fingered Robot Hands Using Neural Networks	94
5.1 Dynamics Equations and Problem Formulation	95
5.2 Vision Based Neural Network Control of Multi-fingered Robot Hands	100
5.3 Vision Based Adaptive Jacobian Control of Multi-fingered Robot Hands	110
5.4 Simulation Results	117
5.4.1 Vision Based Neural-Network Control of Multi-fingered Robot Hands	119
5.4.2 Vision Based Adaptive Jacobian Control of Multi-fingered Robot Hands	120
5.5 Conclusion	128
6 Conclusion and Future Works	129
6.1 Conclusion	129
6.2 Future Research	131
Author's Publications	133
Bibliography	135

List of Figures

2.1	Camera model	16
3.1	Camera-manipulator system	24
3.2	Uncertain constraint surface	26
3.3	Rotation matrix	28
3.4	An RBF neural network	38
3.5	Example of constraint uncertainty	40
3.6	A two-link robot in contact with a constraint surface	47
3.7	Position error in X axis of Cartesian space controller	50
3.8	Position error in Y axis of Cartesian space controller	50
3.9	Force error of Cartesian space controller	50
3.10	Image error in X axis of PD controller	52
3.11	Image error in Y axis of PD controller	52
3.12	Force error of PD controller	52
3.13	Image error in X axis	53

<i>LIST OF FIGURES</i>	<i>viii</i>
3.14 Image error in Y axis	53
3.15 Force error	53
3.16 Image error in X axis with larger uncertainties	54
3.17 Image error in Y axis with larger uncertainties	54
3.18 Force error with larger uncertainties	54
3.19 A two-link robot in contact with a constraint surface	55
3.20 Image error in X axis using neural network controller	57
3.21 Image error in Y axis using neural network controller	57
3.22 Force error using neural network controller	57
3.23 Image error in X axis with larger uncertainties using neural network controller	59
3.24 Image error in Y axis with larger uncertainties using neural network controller	59
3.25 Force error with larger uncertainties using neural network controller .	59
4.1 Illustration of rotation matrix	68
4.2 Image tracking error	89
4.3 Force tracking error	89
4.4 Image tracking error with larger uncertainties	90
4.5 Force tracking error with larger uncertainties	90
4.6 Image tracking error in X	93
4.7 Image tracking error in Y	93

<i>LIST OF FIGURES</i>	ix
4.8 Force tracking error	93
5.1 Multi-fingered robot holding an object	97
5.2 Definitions of Coordinates	98
5.3 Rotation angles	103
5.4 Two three-link robots grasping an object	119
5.5 Position errors	121
5.6 Force errors	122
5.7 Position errors with larger uncertainties	123
5.8 Force errors with larger uncertainties	124
5.9 Position errors with adaptive Jacobians	126
5.10 Force errors with adaptive Jacobians	127

Chapter 1

Introduction

With the development of robot control theory and the demand of industrial automation, robots have been widely used in factories. The robot systems that are used in manufacturing applications perform mainly repetitive operations in a structured environment. When the environment is not well structured, good performance can not be guaranteed. Vision-based control is a promising way to overcome the problems that many controllers currently face. In particular, it would allow robots to work in an unstructured environment. With rapid development in computing and sensor technologies, more sophisticated robots with vision can be designed. Robot systems can now be used in applications where the environment is less structured than those inhabited by industrial robots, for example, outdoor environment. Most research on vision-based control of robot manipulators has been focused on free motion control and hence the applications are limited. When the end-effector is in contact with a constraint surface, it is necessary to control not only the motion but also the force.

This research is intended for the development of hybrid vision-force controllers that do not require calibrations. The proposed controllers will be able to cope with camera miscalibration and accomplish contact tasks effectively even when the kinematics, dynamics and environment are uncertain.

1.1 Literature Survey

Most of industrial robots are used only for positioning tasks such as spray-painting and spot welding. To expand the applications of robots, such as polishing and deburring, it is important to control not only position but also force of interaction between the robot and environment.

There are two major approaches for robot force control, namely impedance control and hybrid control. The impedance control approach was first proposed by Hogan [26]. In this method, the environment is generally modeled as a dynamic system with stiffness, damping, and inertia terms. This method uses a general environment model to represent many environments encountered by a manipulator end-effector. It can work well without accurate information on the shape and elasticity of the object or environment, but it may not be possible to achieve exact motion and force tracking control simultaneously. This approach is a generalization of stiffness control [56] and admittance control [67]. Its philosophy is that the manipulator control system should be designed not to track a motion or force trajectory, but rather to regulate the mechanical impedance of the manipulator. By proper choice of the impedance, dynamic interaction can be controlled to obtain proper force response.

The concept of hybrid position/force control was first proposed by Raibert and Craig [54]. In this method, the task related coordinated system was divided into two orthogonal subspace. Position was controlled in one subspace and force was controlled in the other. Usually, a hybrid position/force control system has two feedback loops for control of position and force separately. Corrections based on the position and force errors are applied by actuators to make the manipulator track the desired position and force trajectories.

According to different characteristics of the contact surfaces, two types of models

were proposed. The first is to consider the environment as rigid surfaces, such as constrained motion control, which was discussed in [40, 63]. In many cases the contact surface stiffness is so large that the surfaces must, in practical terms, be viewed as rigid. The second type models the environment as surfaces with finite stiffness [16, 55, 71]. Deformations of the surfaces along their normal directions are related to the normal contact forces applied by the robot end-effector by a finite stiffness matrix.

Yoshikawa et al. [78] proposed the dynamic hybrid control approach with consideration the robot dynamics and the end-effector constraint. A two-step method was proposed to design controllers for dynamic position/force hybrid control of manipulators. The first step is the linearization of the manipulator dynamics by nonlinear feedback in which the formulation of the constraint hypersurfaces plays an important role. The second step is the design of position and force servo-controllers for the linearized model. The merit of this servo-controller is that it can take account of both the command response and the robustness of the controllers to modeling errors and disturbances.

Mills and Goldenberg [41] analyzed the structure of constrained dynamic system and applied linear descriptor variable theory to position and force control of the constrained robots on rigid surface. The position and force control problem with uncertain dynamics is analyzed and a controller is proposed.

A main problem for most hybrid position/force controllers is that the exact knowledge of the environment is required. In [77], Yoshikawa and Sudou studied the problem of dynamic control with unknown constraint. An online estimation algorithm was developed, which estimates the local shape of the constraint surface by using measured data on the position and force of the end-effector. Guglielmo and Sadegh [23] proposed a hybrid learning force/position control scheme for moving tangentially along an unknown surface with a specified contact force. The learning

law was also applied to online trajectory generation for maintaining normal contact to the unknown surface. Wang et al. [64] designed a joint space controller that can achieve stability in the presence of the constraint uncertainties.

When the robot system is well calibrated, model based approaches [3, 40, 78] can achieve very good performance. However, exact models of the robot system are required in these approaches, which means the robot can not adapt to changes and uncertainties in the models and environment. For example, when the robot picks up different tools of unknown lengths, the kinematics and dynamics of the robot changes and are difficult to derive exactly.

To deal with uncertainty in dynamic parameters, several adaptive motion and force controllers were proposed. Arimoto et al. [3] proposed model based joint space adaptive controller based on passivity of robot dynamics and joint space orthogonalization. Using the fact that uncertain physical parameters can be linearly separated from the equation of motion of robots, dynamic regressors were used to design adaptive control laws. In [73], an adaptive motion and force tracking controller was proposed to deal with unknown dynamic parameters and unknown surface parameters such as stiffness and friction coefficient. It was also shown that stability can be achieved when the bounds of modeling errors are known. In [66], experiments results were presented and comparisons were made between several kinds of controllers to show that the performance of adaptive model-based controllers in [3, 5] is better than both non-model-based and non-adaptive controllers in a wide range of working conditions. Several adaptive control laws have also been proposed to deal with dynamics uncertainties [4, 8, 30, 51]. However, these controllers have assumed that the kinematics of the robot is exactly known.

Recently, Doulgeri and Arimoto [16] studied the position and force control problem for a robot finger with soft tip and uncertain kinematics. Kinematic parameter uncertainties considered in this result include the contact point position and the con-

straint surface orientation. It was shown that asymptotic stability can be achieved even with the uncertainties. Cheah et al. [12] proposed a hybrid position and force controller for constrained manipulators with uncertain kinematics and dynamics. Approximate Jacobian was used and update laws for the estimated parameters were given. Conditions for the bound of uncertainties and feedback gains were also presented to guarantee the stability. However, these results are limited to a class of kinematic uncertainty.

Cooperative control of multi-fingered robot hands is another important application of robots which requires controlling both the position and force. This research area has received increasing attention in recent years because of the requirement of dextrous manipulation and task flexibility. Many research efforts have been devoted to the study of control of multi-fingered robot hands [7,18,34,37]. In [76], Yoshikawa and Nagai analysed and defined the force components involved in manipulation of objects by multi-fingered robot hands. Arimoto et al. [6] derived a mathematical model of the dynamics of a set of dual fingers with soft and deformable tips, The control problem for dynamic stable grasping and dexterity in object manipulation was also addressed. Several control laws have also been proposed for dynamic control of multi-fingered robot hands [42,81]. Some visual based manipulation results can be found in [1,74,79]. However, these results have assumed that the kinematics and dynamics are exactly known. It is interesting to notice that human fingers do not need exact knowledge of the objects, kinematics and contact points of the fingers but still can manipulate things easily and skillfully. This is because humans can adapt to the uncertainty from previous experience. It is therefore important to develop multi-fingered robot controllers to cope with uncertainty in kinematics and dynamics. To deal with the uncertain dynamic parameters, Naniwa et al. [45] proposed a model-based adaptive controller for coordinated control of manipulators. In this result, joint space orthogonalization was used to design nominal reference signal and regressors were used to adapt dynamics modeling errors. A few adaptive

controllers are also proposed in [62,72]. However, these controllers still assume that the exact kinematics and Jacobian mapping from joint space to Cartesian space are known. In practice, due to the imperfect knowledge of the locations of the contact points, the geometry of the object, etc. the kinematics of the multi-fingered robot hands can not be determined exactly. For example, when the fingers are rolling on the object's surface, the contact points are often uncertain. When the robot finger tips are soft and deformable, the kinematics also becomes uncertain due to depression and area contact. To alleviate this problem, Cheah et al. [10] proposed a task space control law for setpoint control of multi-fingered robot hands with uncertain Jacobian matrices. In this result, the exact kinematics and Jacobian of the multi-fingered robot hand are not required. However, this result is limited to a class of kinematic uncertainty. In addition, it is assumed that the uncertain parameters can be linearly parameterized.

Most of today's robots operate in factories where the environment can be contrived to suit the robot. Robots are far less effective in applications where the work environment and object placement cannot be accurately controlled [29]. This limitation is largely due to the lack of sensory capability in robot systems. In order to extend the application of robots, it is necessary to make the robots operate in everyday world. Unlike the manufacturing application, one cannot make this environment to suit the robot, but make the robot adapt to the environment.

Vision is a useful sensor because it allows noncontact measurement of the environment. Visual servoing is the fusion of results from many elemental areas such as high-speed image processing, kinematics, dynamics, control theory and real-time computing. The first visual servoing systems were reported in the early 1980s, but the progress in visual control of robots has been very slow. In the last few years, the computing power has developed enormously so that it allows analysis of scenes at a sufficient rate to servo a robot manipulator.

Most research on vision control of robot manipulators has been focused on motion control [9, 14, 22, 24, 27, 38, 39, 65, 68]. A review of such approaches can be found in [29]. Therefore, the applications of current vision-based controllers are restricted to free motion, in which the robot end effector is not in contact with the environment. In order to expand the applications of vision-based controllers, a new challenge is to control the force in addition to the motion.

Nelson et al. [47] developed a control structure that integrated visual feedback with force feedback within the same manipulator feedback loop. The inner loop was a PID controller with gravity compensation. It was shown that vision can be used to simplify the design problem by allowing the effective use of low gain force control. In [46], the concept of force and vision resolvability using the same controller was studied. This concept was introduced as means of comparing the ability of the two sensing modes to provide useful information during robotic manipulation tasks. By monitoring the resolvability of the two sensing modes with respect to the task, the information provided by the disparate sensors can be seamlessly assimilated during task execution. Hosoda et al. [28] proposed a controller which has online estimators for the parameters of the camera-manipulator system and the unknown constraint surface. Then in [53], Pichler and Jagerstand showed how the constraint geometry can be learned and refined during contact manipulations, and how transformations between the two sensory (force and vision), motor and constraint frames can be learned online. In these approaches [27, 28, 46, 47, 53], the stability of close-loop system was also not proven with the consideration of the effects of robot dynamics.

Xiao et al. [69] discussed control problems of a robot manipulator in an uncalibrated workspace. A projection matrix was used to decouple control variables into two sub-spaces based on sensory information. One sub-space was for force control and the other was for constrained motion control. This decoupling allowed designing control schemes for regulation of force and for constrained motion separately. In [70], A new

control strategy was proposed based on multi sensor fusion using this decoupling method. In these papers [69, 70], the dynamics and kinematics of the robot was assumed to be exactly known.

When the kinematics of the robot system is uncertain, it is impossible to derive the desired joint angle from the desired manipulator path. When the control problem is formulated directly in task space or sensor space, the inverse kinematics problem is replaced by a Jacobian transpose in the control law. However, such schemes still require the exact knowledge of the Jacobian matrix of the mapping from joint space to task space or sensor space. Cheah et al. [12] showed that the asymptotic stability of the sensory feedback control law with uncertainties in dynamics and the entire Jacobian matrix from the joint coordinates to the sensor coordinates via Cartesian coordinates can be established. Sufficient conditions for the bound of the estimated Jacobian matrix and stability conditions for the feedback gains are presented to guarantee the stability and passivity of the robots. A gravity regressor and a force regressor with uncertain Jacobian matrix are also proposed for gravity and force compensation in the presence of uncertainties. This paper showed one promising way of developing new methods for hybrid force-vision control. This controller is based on visual feedback from cameras and the velocity measurements are obtained from differentiation of the image feature and hence are often contaminated by noise.

In [11], a hybrid position and force controller was proposed to deal with the problem of uncertain constraint surface. A projection matrix is designed to project the position error to the tangent plane of the constraint surface. In this result, the projection matrix is assumed to be full rank. However, the projection matrix is not of full rank due to its structure, therefore we can not conclude convergence of position error from convergence of projected position error.

Recently, Dean-Leon et al. [15] proposed an image-based position-force controller based on visual orthogonalization and passivity of the robots. Compensation of dy-

namic joint friction and viscous contact friction was considered. This controller can also cope with uncertain camera and kinematics parameters. However, constraint surface is again assumed to be exactly known.

1.2 Motivation of Research

When the constraint surface is exactly known, the desired position on the surface can be defined in Cartesian space and joint space. However, when the constraint surface is unknown, the desired position in Cartesian space is unknown and hence the desired joint position can not be computed using inverse kinematics. Using vision based control, the desired position or trajectory on the uncertain constraint surface can be defined in image space directly.

In the early works of hybrid vision/force control [28, 46, 47, 53], a vision and force feedback loop is added to the classical PID controller. In these controllers, the effects of the nonlinear dynamics and kinematics is not taken into consideration and the stability of the overall close-loop system has not been proven. Later in [69, 70], the computed torque method is proposed and the nonlinear dynamics is also taken into consideration. However, the dynamics and kinematics of the robot was assumed to be exactly known, which means the robot can not adapt to changes and uncertainties in the models and environment. For example, when the robot picks up different tools of unknown lengths, the kinematics and dynamics of the robot changes and are difficult to derive exactly. When the kinematics of the robot system is uncertain, it is impossible to derive the desired joint angle from the desired manipulator path. In practice, it is also difficult to model the constraint surface exactly. For example, the exact structure of the environment is usually unknown or the uncertain parameters of the environment can not be linearly parameterized. Hence robots are still far less effective in applications where the work environment and object placement cannot

be accurately controlled.

Cheah et al. [11, 12] proposed a task-space controller and showed the asymptotic stability of the vision and force feedback control law with uncertainty in dynamics and the Jacobian matrix. But one point noticeable is that this controller needs the image velocity from vision feedback, which is obtained from differentiation of the image feature and hence is often very noisy. It is also assumed that the exact model of the constraint surface in image space is known in this controller. But in actual implementations, the constraint surface is often formulated in Cartesian space because it is difficult to model the surface in image space. This result is also limited to a class of kinematic uncertainty and it is assumed that the uncertain parameters can be linearly parameterized.

In the vision and force controllers [11, 12, 28, 46, 47, 53, 69, 70], the exact structure of the environment is assumed to be known. In addition, no result has considered uncertain kinematics, dynamics, camera parameters and uncertainty of constraint surface together. It is interesting to notice that human beings do not need exact knowledge of the objects, lengths of arms but still can manipulate different tools and objects easily and skillfully on a constant surface which is not exactly known. This is because humans can adapt to the uncertainty from previous experience.

This thesis is devoted to the development of vision and force controllers for robot with uncertain kinematics, dynamics, camera model and constraint surface. Control problems of position and force regulation, trajectory tracking and multi-fingered robot manipulation are studied and adaptive controllers are proposed.

1.3 Contributions of the Thesis

This research is intended for the development of vision and force controllers for both manipulator and multi-fingered robot with uncertainties. The major contributions of this thesis are as follows:

1. Vision based position and force regulation results with uncertainties in robot dynamics, kinematics, camera model and constraint surface are proposed. An adaptive neural network controller is proposed to compensate the uncertainties when the uncertain parameters can not be linearly parameterized or the structure of constraint surface is unknown. Sufficient conditions are presented to guarantee the stability of the system.
2. Vision based position and force controllers for multi-fingered robots with uncertain dynamics, kinematics, object parameters and contact points are proposed. Adaptive neural-network Jacobian is introduced to deal with uncertain structure of Jacobian matrices and uncertain contact points.
3. A visually-servoed motion and force tracking controller with uncertain dynamics, kinematics and constraint surface is proposed. Adaptive Jacobian matrices are used to update the unknown parameters online and to ensure the convergence of motion and force tracking errors.
4. A novel rotation matrix is introduced instead of the use of projection matrix in [2]. Compared with the projection matrix, the rotation matrix can rotate arbitrary vectors, which makes it more flexible in designing force controllers.

1.4 Organization of the Thesis

The remainder of the thesis is organized as follows:

In *Chapter 2*, a brief introduction to the robot kinematics and dynamics is provided. The control task definition and kinematics properties of visual servoing are also briefly illustrated.

In *Chapter 3*, two controllers are proposed for the vision and force regulation problem of robot with uncertain kinematics, dynamics and constraint surface. First, a new adaptive control method using regressors is proposed for robot with uncertain kinematics, dynamics and constraint surface. Then an adaptive neural network setpoint controller is proposed to deal with uncertain structure of constraint surface. Simulation results are presented to show the effectiveness of the proposed control schemes.

Chapter 4, addresses the visually-servoed trajectory tracking problem of constrained robots with uncertain kinematics, dynamics, camera parameters and constraint surface. A new adaptive Jacobian vision and force trajectory tracking controller is proposed to solve the problem. Then a neural network Jacobian controller is proposed to deal with uncertain structure in the Jacobian matrix. Simulation results are provided to verify the proposed control schemes.

Chapter 5, addresses the hybrid vision-force cooperative control problem of multi-fingered robot hands with uncertainties in kinematics and dynamics. A neural network controller with fixed approximate Jacobian is first proposed. Then an adaptive Jacobian controller, which does not require exact structure of the Jacobian matrix and location of contact points is proposed. Simulation results are presented to illustrate the performance of the controllers.

Chapter 6 gives conclusion of the thesis and suggests the topics of future research.

Chapter 2

Robot Kinematics and Dynamics

In order to make robots complete some specific tasks, the robot is required to move according to the commands from the controller. Control of the end-effector requires an analysis of the structure of the robots to derive mathematical models of robot components. In this chapter, robot kinematics and dynamics are illustrated which constitute the basis of controller design in this thesis.

2.1 Robot Kinematics

Kinematics is the description of the manipulator motion with respect to a fixed reference coordinate frame by ignoring the forces and moments that cause motion of the manipulator.

2.1.1 Direct Kinematics and Jacobian Matrix

A robot manipulator consists of a series of rigid links connected by joints. The whole structure forms a kinematic chain. One end of the chain is constrained to a base and an end-effector or tool is connected to the other end. The mechanical structure

of a manipulator is characterized by a number of degrees of freedom (DOF) which uniquely determine its configuration. Each degree of freedom is typically associated with a joint articulation and constitutes a joint variable. The space in which the vector of joint variables denoted by

$$q = [q_1, \dots, q_n]^T \quad (2.1)$$

is often referred to as joint space, where n denotes the number of manipulator joints. The space in which the manipulator task is specified is referred to as task space which could be Cartesian space or camera image space depending on different control task definitions.

2.1.1.1 Direct Kinematics

The direct kinematics of a manipulator is the relationship between the position and orientation in the end-effector frame and the joint variables with respect to the base frame or other task space frames. The direct kinematics equation can be written in the form [2]

$$r = h(q), \quad (2.2)$$

where $r \in \mathbb{R}^m$ represents the task space vector of the end-effector and $h(\cdot) \in \mathbb{R}^n \rightarrow \mathbb{R}^m$ represents a generally nonlinear function which computes the task space variables from the joint space variables.

2.1.1.2 Jacobian Matrix

Differentiating the direct kinematics function (2.2) with respect to time, the relationship between joint velocities and the end-effector velocities is obtained as [43] [2]:

$$\dot{r} = J_m(q)\dot{q}, \quad (2.3)$$

where

$$J_m(q) = \frac{\partial h(q)}{\partial q} \quad (2.4)$$

is referred to as *analytical Jacobian matrix* or simply as *Jacobian matrix*. The Jacobian matrix is one of the most important tools for manipulator characterization. [75]

2.1.2 Image Jacobian Matrix

In this section, the kinematics in visual servoing control system is presented. Visual servoing control has attracted much research interest in recent years [29]. Visual servoing control is the fusion of image processing and feedback control. In visual servoing control, the control tasks are defined in image space directly.

Visual servoing systems usually use one of the two camera configurations [29]: The camera is fixed in workspace, or the camera is mounted on the end-effector. In both configurations, some image features are first defined on the end-effector or the object grasped by the robot, then the three-dimensional scene is project on the CCD plane in the camera and forms a two-dimensional (2D) image. The image is processed to extract image features. The mapping from feature space to image space requires a camera-lens model in order to present the projection of features onto the CCD image plane. The pinhole camera model [29] as shown in figure 2.1 is widely used and has proven adequate for most visual servoing tasks. Let $x \in \mathbb{R}^p$ denote a vector of image feature parameters and $\dot{x}$ the corresponding vector of image feature parameter rates of change. The relationship between Cartesian space and image space is represented by [29],

$$\dot{x} = J_I(r)\dot{r}, \quad (2.5)$$

where $J_I(r) \in \mathbb{R}^{p \times n_o}$ is the image Jacobian matrix. The image Jacobian was first introduced by Weiss et al. [65], who referred to it as the feature sensitivity matrix.

Figure 2.1: Camera model

It is also referred to as the interaction matrix [17] and the B matrix [49, 50]. Then the image velocity vector is related to the joint velocity vector as

$$\dot{x} = J_I(r)J_m(q)\dot{q} = J(q)\dot{q}, \quad (2.6)$$

where $J(q)$ is the manipulator Jacobian matrix mapping from joint space to Cartesian space.

A property of the kinematic equation described by equation (2.6) is stated as follows:

Property 2.1 The right hand side of equation (2.6) is linear in a set of kinematic parameters $\theta_k = (\theta_{k1}, \dots, \theta_{kq})^T$, such as link lengths, joint offsets, focus length. Hence, equation (2.6) can be expressed as,

$$x = J(q)\dot{q} = Y_k(q, \dot{q})\theta_k \quad (2.7)$$

where $Y_k(q, \dot{q}) \in \mathbb{R}^{n_q}$ is called the kinematic regressor matrix.

Next, an example of image Jacobian is provided [29]. For simplicity, we consider a fixed camera system and the camera coordinate frame is aligned with the task coordinated frame as shown in figure 2.1. Image Jacobian matrix for eye-in-hand configuration can be found in [31]. Let (x_u, x_v) be the image feature position on the image plane, s_u and s_v be the horizontal and vertical dimensions of the pixels on the CCD, f be the focal length, (r_{fx}, r_{fy}, r_{fz}) denotes the object feature position in camera coordinate and (r_x, r_y, r_z) denotes the object feature position in Cartesian space. It is assumed that $|r_{fz}| \gg |f|$, where f is the focal length, then the mapping from object feature velocity to image feature velocity on image plane can be obtained as [46]:

$$\begin{aligned}\dot{x}_u &= \frac{f\dot{r}_{fz}}{s_u r_{fz}} - x_u \frac{\dot{r}_{fz}}{r_{fz}} \\ \dot{x}_v &= \frac{f\dot{r}_{fy}}{s_v r_{fz}} - x_v \frac{\dot{r}_{fz}}{r_{fz}}.\end{aligned}\tag{2.8}$$

Next, task space velocities are transformed into the camera frame and then projected to the image plane to get the image Jacobian. Suppose the end-effector is moving with translation velocity $\dot{r}_p = (\dot{r}_x, \dot{r}_y, \dot{r}_z)^T$ and angular velocity $\dot{r}_o = (\omega_x, \omega_y, \omega_z)^T$ with respect to the task frame. One can write the derivatives of r_f in terms of the velocity in Cartesian space as [46]

$$\begin{aligned}\dot{r}_{fx} &= r_z \omega_y - r_y \omega_z + \dot{r}_x, \\ \dot{r}_{fy} &= r_x \omega_z - r_z \omega_x + \dot{r}_y, \\ \dot{r}_{fz} &= r_y \omega_x - r_x \omega_y + \dot{r}_z.\end{aligned}\tag{2.9}$$

From equations (2.8) and (2.9), the entire transformation from task space to image

space in matrix form can be written as [46]

$$\begin{bmatrix} \dot{x}_u \\ \dot{x}_v \end{bmatrix} = \begin{bmatrix} \frac{f}{s_u r_{fz}} & 0 & -\frac{x_u}{r_{fz}} & -\frac{r_{yu} x_u}{r_{fz}} & \left(\frac{f r_z}{s_u r_{fz}} + \frac{r_x x_u}{r_{fz}} \right) & -\frac{f r_y}{s_u r_{fz}} \\ 0 & \frac{f}{s_v r_{fz}} & -\frac{x_v}{r_{fz}} & -\left(\frac{f r_z}{s_v r_{fz}} + \frac{r_y x_v}{r_{fz}} \right) & \frac{r_x x_v}{r_{fz}} & \frac{f r_x}{s_v r_{fz}} \end{bmatrix} \begin{bmatrix} \dot{r}_x \\ \dot{r}_y \\ \dot{r}_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}, \quad (2.10)$$

which can be written as

$$\dot{x} = J_I(r) \dot{r}. \quad (2.11)$$

The above equation relates image-plane velocity of a point to the relative velocity of the point with respect to the camera. Equation (2.10) can be extended to a general case using k image features as [29]

$$\begin{bmatrix} \dot{x}_{u1} \\ \dot{x}_{v1} \\ \vdots \\ \dot{x}_{uk} \\ \dot{x}_{vk} \end{bmatrix} = \begin{bmatrix} \frac{f}{s_u r_{fz1}} & 0 & -\frac{x_{u1}}{r_{fz1}} & -\frac{r_{y1} x_{u1}}{r_{fz1}} & \left(\frac{f r_{z1}}{s_u r_{fz1}} + \frac{r_{x1} x_{u1}}{r_{fz1}} \right) & -\frac{f r_{y1}}{s_u r_{fz1}} \\ 0 & \frac{f}{s_v r_{fz1}} & -\frac{x_{v1}}{r_{fz1}} & -\left(\frac{f r_{z1}}{s_v r_{fz1}} + \frac{r_{y1} x_{v1}}{r_{fz1}} \right) & \frac{r_{x1} x_{v1}}{r_{fz1}} & \frac{f r_{x1}}{s_v r_{fz1}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{f}{s_u r_{fzk}} & 0 & -\frac{x_{uk}}{r_{fzk}} & -\frac{r_{yk} x_{uk}}{r_{fzk}} & \left(\frac{f r_{zk}}{s_u r_{fzk}} + \frac{r_{xk} x_{uk}}{r_{fzk}} \right) & -\frac{f r_{yk}}{s_u r_{fzk}} \\ 0 & \frac{f}{s_v r_{fzk}} & -\frac{x_{vk}}{r_{fzk}} & -\left(\frac{f r_{zk}}{s_v r_{fzk}} + \frac{r_{yk} x_{vk}}{r_{fzk}} \right) & \frac{r_{xk} x_{vk}}{r_{fzk}} & \frac{f r_{xk}}{s_v r_{fzk}} \end{bmatrix} \begin{bmatrix} \dot{r}_x \\ \dot{r}_y \\ \dot{r}_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}, \quad (2.12)$$

2.2 Robot Dynamics

Dynamic model of robot manipulators plays an important role in controller design, simulation of motion and analysis of system structures. The dynamic model describes the relationship between torques exerted on the joint actuators and motion

of the links. In this section, the dynamics of robot and some important properties are discussed.

2.2.1 Dynamic Model of Robot Manipulator

In this section, Lagrange formulation is used to derive the dynamic model. With Lagrange formulation, the equation of motion can be derived in a systematic way independently of the reference coordinate frame. First, the Lagrangian is defined as [2]:

$$L = K - P, \quad (2.13)$$

where K is the kinetic energy and P is the potential energy. Let $q \in \mathbb{R}^n$ be the joint variable vector and $\tau \in \mathbb{R}^n$ be a vector of input torques on each joint, Then the Lagrangian of the manipulator system can be written as

$$L(q, \dot{q}) = K(q, \dot{q}) - P(q) = \frac{1}{2} \dot{q}^T M(q) \dot{q} - P(q), \quad (2.14)$$

where $M(q) \in \mathbb{R}^{n \times n}$ is the manipulator inertial matrix.

The Lagrange's equation of motion for a rigid body system is given by [2]:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \tau - F. \quad (2.15)$$

where $F \in \mathbb{R}^n$ is the torque on the environment exerted by the robot expressed in the joint space. Substitute equation (2.14) into equation (2.15), the dynamic equation of a robot manipulator in joint space with n degrees of freedom is obtained as [2]:

$$M(q)\ddot{q} + \left(\frac{1}{2}\dot{M}(q) + S(q, \dot{q})\right)\dot{q} + g(q) = \tau - J_m^T(q)f, \quad (2.16)$$

where $S(q, \dot{q}) = \frac{1}{2}\dot{M}(q)\dot{q} - \frac{1}{2}\left\{\frac{\partial}{\partial \dot{q}}\dot{q}^T M(q)\dot{q}\right\}^T \in \mathbb{R}^n$, $\left(\frac{1}{2}\dot{M}(q) + S(q, \dot{q})\right)\dot{q}$ is the Corio-

lis/centripetal vector and $g(q) = \frac{\partial P(q)}{\partial q} \in \mathbb{R}^n$ is the gravity force vector, $f \in \mathbb{R}^m$ is the generalized force expressed in Cartesian space and $F = J_m^T(q)f$.

A constraint surface is defined in an algebraic term as,

$$\Psi(r) = 0, \quad (2.17)$$

where $\Psi(r) : \mathbb{R}^m \rightarrow \mathbb{R}^1$ is a given scalar function. Differentiating the above equation yields

$$\frac{\partial \Psi(r)}{\partial r} \dot{r} = 0. \quad (2.18)$$

If the constraint surface is rigid and frictionless and $\Psi(r)$ has a continuous gradient, the generalized force is then given by,

$$f = \frac{(\partial \Psi(r)/\partial r)^T}{\|\partial \Psi(r)/\partial r\|} \lambda = d(r)\lambda, \quad (2.19)$$

where $\lambda \in \mathbb{R}$ is defined as the magnitude of the contact force and $d(r) = \frac{(\partial \Psi(r)/\partial r)^T}{\|\partial \Psi(r)/\partial r\|} \in \mathbb{R}^m$ is a unit vector and denotes the direction of contact force.

Some important properties in equation (2.16) are summarized as follows [2]:

Property 2.2 The inertia matrix $M(q)$ is symmetric and positive definite. Furthermore, because each entry of $M(q)$ is constant or a trigonometric function of components of q , $M(q)$ is bounded above and below

$$\lambda_m I \leq M(q) \leq \lambda_M I \quad (2.20)$$

where λ_m and λ_M are positive scalars, which denote the lower and upper bound, respectively.

Property 2.3 Let $C(q, \dot{q}) = \frac{1}{2}\dot{M}(q) + S(q, \dot{q})$ and $C(q, \dot{q})$ satisfies

$$\begin{aligned} C(q, x)y &= C(q, y)x, \forall x, y \in \mathbb{R}^n \\ \|C(q, x)\| &\leq c_M \|x\|, \forall q, x \in \mathbb{R}^n \end{aligned} \quad (2.21)$$

where c_M is a positive scalar. The norm of a vector x is defined as

$$\|x\| = \sqrt{x^T x} \quad (2.22)$$

and the norm of a matrix A as

$$\|A\| = \sqrt{\lambda_{\max}(A^T A)} \quad (2.23)$$

where $\lambda_{\max}(\cdot)$ and $\lambda_{\min}(\cdot)$ denote the maximum and minimum eigenvalue respectively.

Property 2.4 The matrix $S(q, \dot{q})$ is skew-symmetric and satisfies

$$y^T S(q, \dot{q})y = 0, \forall q, \dot{q}, y \in \mathbb{R}^n \quad (2.24)$$

Property 2.5 The gravity force vector $g(q)$ can be completely characterized by a set of physical parameters $\phi = (\phi_1, \dots, \phi_p)^T$ as [2]:

$$g(q) = Z(q)\phi = [z_1\phi, \dots, z_n\phi]^T, \quad (2.25)$$

where $Z(q) \in \mathbb{R}^{n \times p}$ is the gravity regressor and $z_i(q) \in \mathbb{R}^{1 \times p}$ ($i = 1, \dots, n$) is the i^{th} row of $Z(q)$, ϕ is a $p \times 1$ parameter vector.

Property 2.6 The left hand side of equation (2.16) is linear in a set of physical parameters $\theta_d = (\theta_1, \dots, \theta_p)^T$ such as link masses, moments of inertia and damping

factors, hence can be expressed as

$$M(q)\ddot{q} + \left(\frac{1}{2}\dot{M}(q) + S(q, \dot{q})\right)\dot{q} + g(q) = Y_d(q, \dot{q}, \ddot{q})\theta_d, \quad (2.26)$$

where $Y_d(q, \dot{q}, \ddot{q})$ is called a dynamic regressor.

Chapter 3

Hybrid Vision-force Setpoint Control for Constrained Robots with Uncertainties

3.1 Introduction

As discussed in Chapter 1, most of the hybrid vision-force control methods assume that the kinematics, camera parameters and constraint surface are known. Recently, several controllers [12, 15, 16] using approximate Jacobian have been proposed to overcome the uncertainties in kinematics and dynamics. These position and force controllers do not need exact knowledge of kinematics and dynamics. However, uncertain constraint surface parameters or structure are not considered in these results.

This chapter is dedicated to the research of visually-servoed regulation problem of constrained robots with uncertain kinematics, dynamics, camera parameters and constraint surface. In section 3.2, an adaptive setpoint controller is proposed for

robot with uncertainties in kinematics, dynamics, camera parameters and constraint surface. In section 3.3, the controller is extended to a neural-network controller to deal with uncertainty in the structure of environment.

3.2 Problem Formulation

Consider a vision-force control system consisting of a robot manipulator, camera(s) fixed in the workspace and a force sensor, which is shown in figure 3.1. In this system, the end effector is in contact with a constraint surface. As discussed in Chapter

Figure 3.1: Camera-manipulator system

2.2.1, the dynamic equation of motion for the constrained robotic manipulator is given by [2] [60],

$$M(q)\ddot{q} + \left(\frac{1}{2}\dot{M}(q) + S(q, \dot{q})\right)\dot{q} + g(q) = \tau + J_m^T(q)f, \quad (3.1)$$

Equation (3.1) can also be represented as,

$$M(q)\ddot{q} + \left(\frac{1}{2}\dot{M}(q) + S(q, \dot{q})\right)\dot{q} + g(q) = \tau + D^T(q)\lambda, \quad (3.2)$$

where $D(q) = (\partial\Psi(r)/\partial r)^T / \|\partial\Psi(r)/\partial r\| J_m(q) : \mathbb{R}^n \rightarrow \mathbb{R}^{(1 \times m)}$ is a Jacobian of the constraint function. From equations (2.3) and (2.18), note that

$$D(q)\dot{q} = 0. \quad (3.3)$$

In a task space position and force control system, the desired position and force are both defined in Cartesian space. However, when the kinematics and constraint surface are both uncertain, the desired joint position can not be obtained using inverse kinematics, neither the desired position can be obtained in Cartesian space because the constraint surface is uncertain (see figure 3.2). Vision-force controller can overcome this problem because the desired position can be defined in image space. However, this makes the problem more complicated because the position and force are in two different coordinate frames, one is image space and the other is Cartesian space. The use of vision sensor also introduces additional uncertainty from Cartesian space to image space and hence it is not sure whether the system can still be stabilized.

3.3 Vision-force Control of Robots with Uncertainties

The kinematics of the robot is often uncertain because the exact parameters of a robot are hard to determine especially when the robot picks up different tools. In addition, it is often difficult to obtain exact models of the constraint surface. In

Figure 3.2: Uncertain constraint surface

the presence of uncertainties in kinematics and constraint surface, $D(q)$ and $\Psi(r)$ becomes uncertain and is estimated as

$$\hat{D}(q) = \frac{\partial \hat{\Psi}(\hat{r}) / \partial \hat{r}}{\|\partial \hat{\Psi}(\hat{r}) / \partial \hat{r}\|} \hat{J}_m(q) = \hat{d}^T(\hat{r}) \hat{J}_m(q) \quad (3.4)$$

where $\hat{J}_m(q) \in \mathbb{R}^{m \times n}$ is an estimation of $J_m(q)$, $\hat{\Psi}(\cdot) : \mathbb{R}^m \rightarrow \mathbb{R}$ is an estimation of $\Psi(\cdot)$ and $\hat{r} = \hat{h}(q)$ is an estimation of $r = h(q)$.

A fundamental benefit of image based control is to overcome the kinematics and constraint uncertainties. Using cameras, the position of the end effector and its desired position on the uncertain constraint surface can be obtained in image space. However, the use of cameras also introduce uncertainties in the camera parameters which should be taken into consideration. In the presence of uncertainty in the camera model, the image Jacobian matrix is uncertain and is denoted by $\hat{J}_I(\hat{r})$. In 3D visual servoing, the estimated depth information of the image Jacobian matrix can be obtained from the estimated end effector position $\hat{r} = \hat{h}(q)$ and the estimated distance between the robot and cameras.

Let us define a scalar potential function $S_j(\theta)$ and its derivative $s_j(\theta)$. The functions $S_j(\theta)$ and $s_j(\theta)$ have the following properties [2]:

- (1) $S_j(\theta) > 0$ for $\theta \neq 0$ and $S_j(0) = 0$.
- (2) $S_j(\theta)$ is twice continuously differentiable, and the derivative $s_j(\theta) = \frac{dS_j(\theta)}{d\theta}$ is strictly increasing in θ for $|\theta| < \gamma_j$ with some γ_j and saturated for $|\theta| \geq \gamma_j$, i.e. $s_j(\theta) = \pm s_j$ for $\theta \geq +\gamma_j$ and $\theta \leq -\gamma_j$ respectively where s_j is a positive constant.
- (3) There exist a constant $\bar{c}_j > 0$, such that,

$$S_j(\theta) \geq \bar{c}_j s_j^2(\theta), \quad (3.5)$$

for $\theta \neq 0$.

Some examples of saturation function are given in [2].

Next, an output vector y is defined as

$$y = \dot{q} + \alpha \hat{J}_x^+(q) s(\Delta x), \quad (3.6)$$

where $\hat{J}_x^+(\cdot)$ denotes the pseudo inverse of $\hat{J}_x(\cdot)$, $\hat{J}_x(q) = \hat{J}_I(r) \hat{J}_m(q)$ is an estimation of $J_x(q) = J_I(r) J_m(q)$. α is a positive scalar, $s(\Delta x) = (s_1(\Delta x_1), \dots, s_p(\Delta x_p))^T$, $\Delta x = x - x_d = (\Delta x_1, \dots, \Delta x_p)^T$ is a positional deviation from a desired position of the end-effector x_d in image space and R is a rotation matrix designed so that

$$y^T \hat{J}_m^T(q) R \hat{d}(\hat{r}) = 0. \quad (3.7)$$

In general, the position of the end effector can be partitioned as $r = (r_p^T, r_o^T)^T$, where r_p is the position vector and r_o is the orientation vector. Therefore, $\hat{J}_m(q)y$ can also be partitioned as $((\hat{J}_m(q)y)_p^T, (\hat{J}_m(q)y)_o^T)^T$ and $\hat{d}(\hat{r})$ can be written as $(\hat{d}_p^T(\hat{r}), \hat{d}_o^T(\hat{r}))^T$.

Hence the rotation matrix R can be introduced as,

$$R = \begin{bmatrix} R_p(n_p, \phi_p) & 0 \\ 0 & R_o(n_o, \phi_o) \end{bmatrix}, \quad (3.8)$$

where n_p is a unit vector normal to both the vectors $(\hat{J}_m(q)y)_p$ and $\hat{d}_p(\hat{r})$ (see figure 3.3). n_o is a unit vector normal to both the vectors $(\hat{J}_m(q)y)_o$ and $\hat{d}_o(\hat{r})$, ϕ_p is the angle between $\hat{d}_p(\hat{r})$ and $R_p\hat{d}_p(\hat{r})$, which can be determined from the angle ϕ between $\hat{d}_p(\hat{r})$ and $(\hat{J}_m(q)y)_p$ (see figure 3.3). ϕ_o is the angle between $\hat{d}_o(\hat{r})$ and $R_o\hat{d}_o(\hat{r})$, which can be determined from the angle between $\hat{d}_o(\hat{r})$ and $(\hat{J}_m(q)y)_o$. The rotation matrix R_p is to rotate the vector $\hat{d}_p(\hat{r})$ about the axis n_p , so that the vector $R_p\hat{d}_p(\hat{r})$ is perpendicular to the vector $(\hat{J}_m(q)y)_p$, as shown in figure 3.3. Here the symbol $\times$ means cross product. The rotation matrix R_o can be similarly designed. However, since the constraint surface is usually independent of r_o , the vector $\partial\hat{\Psi}(\hat{r})/\partial\hat{r}_o$ is equal to zero and hence the rotation matrix R_o can be set as an identity matrix.

Figure 3.3: Rotation matrix

From properties 2.4, the gravity term can be completely characterized by a set of

parameters $\theta = (\theta_1, \dots, \theta_i)^T$ [2] as

$$g(q) = Z(q)\theta, \quad (3.9)$$

where $Z(q) \in \mathbb{R}^{n \times i}$ is a gravity regressor matrix. Similarly, the contact force can be completely characterized by a set of parameters $\theta_f = (\theta_{f,1}, \dots, \theta_{f,j})^T$ [11] as

$$D^T(q)\lambda = Z_f(q, \lambda)\theta_f, \quad (3.10)$$

where $Z_f(q, \lambda) \in \mathbb{R}^{n \times j}$ is a force regressor matrix. In the presence of uncertainties, one has

$$\hat{D}^T(q)\lambda = Z_f(q, \lambda)\hat{\theta}_f, \quad (3.11)$$

where $\hat{\theta}_f$ is an estimated value of θ_f . Hence, from equations (3.10) and (3.11)

$$(\hat{D}^T(q) - D^T(q))\lambda = Z_f(q, \lambda)\bar{\theta}_f, \quad (3.12)$$

where $\bar{\theta}_f = \hat{\theta}_f - \theta_f$. The parameters θ of equation (3.9) and $\bar{\theta}_f$ in equation (3.12) are considered to be unknown.

The vision and force control law for the constrained robot is:

$$\begin{aligned} \tau = & -\hat{J}_x^T(q)K_p s(\Delta x) - \hat{J}_x^T(q)K_v \hat{J}_x(q)\dot{q} + Z_f(q, \lambda)\hat{\theta}_f + Z(q)\hat{\theta} - \hat{D}^T(q)\lambda \\ & + \hat{J}_m^T(q)R\hat{d}(\hat{r})(\beta(\lambda - \lambda_d) + \gamma \int_0^t (\lambda - \lambda_d)d\tau), \end{aligned} \quad (3.13)$$

where

$$\|J_x(q) - \hat{J}_x(q)\| \leq p, \quad (3.14)$$

and K_p and K_v are diagonal feedback gain matrices for the position error and joint velocity, γ is a positive constant. λ_d is a desired magnitude of the contact force, $\hat{\theta}$ is an estimator of θ , and $\hat{\theta}_f$ is an estimator of θ_f . The uncertain parameters in the

estimators $\hat{\theta}$ and $\hat{\theta}_f$ are updated, respectively, by the following parameters update laws:

$$\dot{\hat{\theta}} = -LZ^T(q)(\dot{q} + \alpha\hat{J}_x^+(q)s(\Delta x)), \quad (3.15)$$

$$\dot{\hat{\theta}}_f = -L_f Z_f^T(q, \lambda)(\dot{q} + \alpha\hat{J}_x^+(q)s(\Delta x)), \quad (3.16)$$

where $L \in \mathbb{R}^{i \times i}$, $L_f \in \mathbb{R}^{j \times j}$ are positive definite gain matrices, which affect update rate. The update laws are derived from the Lyapunov-like function which will be introduced later.

In the controller described by equations (3.13), (3.15) and (3.16), the task space vector x is measured by a vision system, q is provided by the encoder mounted on the joints and λ is obtained from a force sensor. The Jacobian matrices are assumed to be uncertain.

Substituting equation (3.13) into equation (3.2), the closed-loop equation is obtained as follows:

$$\begin{aligned} & M(q)\ddot{q} + \left(\frac{1}{2}\dot{M}(q) + S(q, \dot{q})\right)\dot{q} + \hat{J}_x^T(q)K_p s(\Delta x) \\ & + \hat{J}_x^T(q)K_v \hat{J}_x(q)\dot{q} + Z(q)\Delta\theta + Z_f(q, \lambda)\Delta\bar{\theta}_f \\ & = \hat{J}_m^T(q)R\hat{d}(\hat{r})(\beta(\lambda - \lambda_d) + \gamma \int_0^t (\lambda - \lambda_d)d\tau), \end{aligned} \quad (3.17)$$

where $\Delta\theta = \theta - \hat{\theta}$ and $\Delta\bar{\theta}_f = \bar{\theta}_f - \hat{\theta}_f$.

To carry out the stability analysis for the closed-loop system, taking the inner product of y in equation (3.6) with the closed-loop equation (3.17), using equations (3.7), (3.15), (3.16) and Property 2.2 and Property 2.4 of the robot dynamics yields

$$\frac{d}{dt}V = -W, \quad (3.18)$$

where

$$V = \frac{1}{2}\dot{q}^T M(q)\dot{q} + \alpha s^T(\Delta x)(\hat{J}_x^+(q))^T M(q)\dot{q} \\ + \sum_{j=1}^p (k_{pj} + \alpha k_{vj})S_j(\Delta x_j) + \frac{1}{2}\Delta\theta^T L^{-1}\Delta\theta + \frac{1}{2}\Delta\bar{\theta}_f^T L_f^{-1}\Delta\bar{\theta}_f, \quad (3.19)$$

$$W = \dot{q}^T \hat{J}_x^T(q)K_v \hat{J}_x(q)\dot{q} + \alpha s^T(\Delta x)K_p s(\Delta x) - s^T(\Delta x)(K_p + \alpha K_v)(J_x(q) - \hat{J}_x(q))\dot{q} \\ - \alpha(s^T(\Delta x)(\hat{J}_x^+(q))^T(\frac{1}{2}\dot{M}(q) - S(q, \dot{q}))\dot{q} + \alpha s^T(\Delta x)(\hat{J}_x^+(q))^T M(q)\dot{q} \\ + \dot{s}^T(\Delta x)(\hat{J}_x^+(q))^T M(q)\dot{q}), \quad (3.20)$$

and $\dot{x} = J_I(r)\dot{r} = J_I(r)J_m(q)\dot{q} = J_x(q)\dot{q}$. This will eventually lead to a Lyapunov-like function for the stability analysis of the robot control problem with uncertain kinematics, dynamics, camera model and constraint surface.

To show the positive definiteness of the Lyapunov function candidate V in equation (3.19), note that

$$\frac{1}{4}\dot{q}^T M(q)\dot{q} + \alpha s^T(\Delta x)(\hat{J}_x^+(q))^T M(q)\dot{q} + \sum_{j=1}^p (k_{pj} + \alpha k_{vj})S_j(\Delta x_j) \\ = \frac{1}{4}(\dot{q} + 2\alpha \hat{J}_x^+(q)s^T(\Delta x))M(q)(\dot{q} + 2\alpha \hat{J}_x^+(q)s(\Delta x)) \\ - \alpha^2 s^T(\Delta x)(\hat{J}_x^+(q))^T M(q)\hat{J}_x^+(q)s(\Delta x) + \sum_{j=1}^p (k_{pj} + \alpha k_{vj})S_j(\Delta x_j) \\ \geq \sum_{j=1}^p (k_{pj}\bar{c}_j + \alpha(k_{vj}\bar{c}_j - \alpha\lambda_m))\|s(\Delta x)\|^2, \quad (3.21)$$

where $\lambda_m = \lambda_{\max}[(\hat{J}_x^+(q))^T M(q)\hat{J}_x^+(q)]$ and $\lambda_{\max}[A]$ denote the maximum eigenvalue of the matrix A . Substituting the above inequality into equation (3.19), one has

$$V \geq \frac{1}{4}\dot{q}^T M(q)\dot{q} + \sum_{j=1}^p (k_{pj}\bar{c}_j + \alpha(k_{vj}\bar{c}_j - \alpha\lambda_m))\|s(\Delta x)\|^2 \\ + \frac{1}{2}\Delta\theta^T L^{-1}\Delta\theta + \frac{1}{2}\Delta\bar{\theta}_f^T L_f^{-1}\Delta\bar{\theta}_f, \quad (3.22)$$

Hence, V is positive definite in $s(\Delta x)$, $\dot{q}$, $\Delta\theta$ and $\Delta\bar{\theta}_f$ since k_{vj} and α can be chosen so that

$$k_{vj}\bar{c}_j - \alpha\lambda_m > 0. \quad (3.23)$$

Therefore, the function V represents a Lyapunov function candidate for the set-point control of the robot manipulator with uncertainties.

Next, it will be shown that the time derivative of the Lyapunov function is negative definite in $s(\Delta x)$ and $\dot{q}$. As seen from equation (3.18), this is equivalent to show that W is positive definite in $s(\Delta x)$ and $\dot{q}$. From the last term on the right-hand side of equation (3.20), since $s(\Delta x)$ is bounded, then there exist a constant c_0 so that [2]

$$\begin{aligned} & \alpha |s^T(\Delta x)(\hat{J}_x^+(q))^T(\frac{1}{2}\dot{M}(q) - S(q, \dot{q}))\dot{q} + s^T(\Delta x)(\hat{J}_x^+(q))^T M(q)\dot{q} \\ & + s^T(\Delta x)(\hat{J}_x^+(q))^T M(q)\dot{q}| \leq \alpha c_0 \|\dot{q}\|^2. \end{aligned} \quad (3.24)$$

Substituting the above inequality into equation (3.20) yields

$$\begin{aligned} W & \geq \dot{q}^T \hat{J}_x^T(q) K_v \hat{J}_x(q) \dot{q} + \alpha s^T(\Delta x) K_p s(\Delta x) \\ & - s^T(\Delta x) (K_p + \alpha K_v) (J_x(q) - \hat{J}_x(q)) \dot{q} - \alpha c_0 \|\dot{q}\|^2. \end{aligned} \quad (3.25)$$

Now, letting $\bar{\Delta} = J_x(q) - \hat{J}_x(q)$, equation (3.25) becomes

$$W \geq \dot{q}^T \hat{J}_x^T(q) K_v \hat{J}_x(q) \dot{q} - \alpha c_0 \|\dot{q}\|^2 + \alpha k_p s^T(\Delta x) K_p s(\Delta x) - s^T(\Delta x) (K_p + \alpha K_v) \bar{\Delta} \dot{q}. \quad (3.26)$$

The existence of the $\bar{\Delta}$ so that W is positive definite can be clearly seen from equation (3.26). In the following development, a sufficient condition will be derived

to guarantee the positive definiteness of W . Note that

$$W \geq (\lambda_{\min}[\hat{J}_x^T(q)K_v\hat{J}_x(q)] - \alpha c_0)\|\dot{q}\|^2 + \alpha k_{p\min}\|\Delta x\|^2 - p(k_{p\max} + \alpha k_{v\max})\|\dot{q}\|\|s(\Delta x)\|, \quad (3.27)$$

where $\lambda_{\min}[A]$ denotes the minimum eigenvalue of matrix A , $k_{p\min}$ is $\lambda_{\min}[K_p]$, $k_{p\max}$ and $k_{v\max}$ are the max eigenvalues of K_p and K_v . Next, note that

$$-\|s(\Delta x)\| \cdot \|\dot{q}\| \geq -\frac{1}{2}(\|s(\Delta x)\|^2 + \|\dot{q}\|^2), \quad (3.28)$$

Substituting inequality (3.28) into equation (3.27) gives,

$$W \geq (\lambda_1 - \alpha c_0 - \frac{1}{2}p(k_{p\max} + \alpha k_{v\max}))\|\dot{q}\|^2 + (\alpha k_{p\min} - \frac{1}{2}p(k_{p\max} + \alpha k_{v\max}))\|s(\Delta x)\|^2, \quad (3.29)$$

where $\lambda_1 = \lambda_{\min}[\hat{J}_x^T(q)K_v\hat{J}_x(q)]$.

Hence if

$$\begin{aligned} \lambda_1 - \alpha c_0 - \frac{1}{2}p(k_{p\max} + \alpha k_{v\max}) &> 0, \\ \alpha k_{p\min} - \frac{1}{2}p(k_{p\max} + \alpha k_{v\max}) &> 0, \end{aligned} \quad (3.30)$$

then $W \geq 0$.

Now the following Theorem can be stated:

Theorem 3.1 *The closed-loop system described by equations (3.17), (3.15) and (3.16), with the uncertain Jacobian matrix and constraint surface $\hat{J}_m(q)$, $\hat{J}_I(\hat{r})$ and $\hat{\Psi}(\hat{r})$ described by equation (3.14), rises to the convergence of $(\Delta x, \dot{q})$ to $(0, 0)$ as $t \rightarrow \infty$, if the feedback gains K_v and K_p are chosen to satisfy conditions (3.23) and (3.30). In addition, the contact force λ converges to λ_d as $t \rightarrow \infty$.*

Proof From equation (3.18), note that

$$\dot{V} = -W \leq 0, \quad (3.31)$$

where V is positive definite in $s(\Delta x)$, $\dot{q}$, $\Delta\theta$ and $\Delta\bar{\theta}_f$ and W is a positive semidefinite function. Integrating both sides of equation (3.18), one has

$$V - V(0) = - \int_0^t W(\tau) d\tau \leq 0. \quad (3.32)$$

Hence, V is bounded and one can conclude that $s(\Delta x)$, $\dot{q}$, $\Delta\theta$ and $\Delta\bar{\theta}_f$ are bounded vectors. From equations (3.29) and (3.32), one can conclude that $s(\Delta x)$, $\dot{q}$ are square integrable functions. Next, multiplying both sides of equation (3.17) by $DM^{-1}(q)$ yields,

$$\begin{aligned} & D(q)\ddot{q} + D(q)M^{-1}(q)\left\{\left(\frac{1}{2}\dot{M}(q) + S(q, \dot{q})\right)\dot{q} + \hat{J}_x^T(q)K_p s(\Delta x) \right. \\ & \quad \left. + \hat{J}_x^T(q)K_v \hat{J}_x(q)\dot{q} + Z(q)\Delta\theta + Z_f(q, \lambda)\Delta\bar{\theta}_f\right\} \\ & = D(q)M^{-1}(q)\hat{J}_m^T(q)R\hat{d}(\hat{r})(\beta(\lambda - \lambda_d) + \gamma \int_0^t (\lambda - \lambda_d) d\tau). \end{aligned} \quad (3.33)$$

From equation (3.12)

$$Z_f(q, \lambda)\Delta\bar{\theta}_f = Z_f(q, \lambda)\theta_f - Z_f(q, \lambda)\hat{\theta}_f = (D_1^T(q) - \hat{D}_1^T(q))\lambda \quad (3.34)$$

where $D_1^T(q) = \hat{D}^T(q) - D^T(q)$. Since $D(q)\ddot{q} = -\dot{D}(q)\dot{q}$, substituting the above

equation into equation (3.33) yields

$$\begin{aligned}
& -\dot{D}(q)\dot{q} + D(q)M^{-1}(q)\{(\frac{1}{2}\dot{M}(q) + S(q, \dot{q}))\dot{q} + \hat{J}_x^T(q)K_p s(\Delta x) \\
& \quad + \hat{J}_x^T(q)K_v \hat{J}_x(q)\dot{q} + Z(q)\Delta\theta + Z_f(q, \lambda_d)\Delta\bar{\theta}_f\} \\
& = D(q)M^{-1}(q)((\beta \hat{J}_m^T(q)R\hat{d}(\hat{r}) - (D_1^T(q) - \hat{D}_1^T(q)))(\lambda - \lambda_d) \\
& \quad + \gamma \hat{J}_m^T(q)R\hat{d}(\hat{r}) \int_0^t (\lambda - \lambda_d) d\tau). \tag{3.35}
\end{aligned}$$

where $Z_f(q, \lambda_d)\Delta\bar{\theta}_f = (D_1^T(q) - \hat{D}_1^T(q))\lambda_d$. Note that $D_1^T(q) - \hat{D}_1^T(q)$ is a bounded vector because $\Delta\bar{\theta}_f$ is bounded. From (3.35), the force error is subject to

$$\Delta(t) = (\lambda - \lambda_d) + \bar{\gamma}(t) \int_0^t (\lambda - \lambda_d) d\tau, \tag{3.36}$$

where

$$\begin{aligned}
\bar{\gamma}(t) &= \frac{\gamma}{\beta + \kappa(q)}, \\
\kappa(q) &= \frac{D(q)M^{-1}(q)(\hat{D}_1^T(q) - D_1^T(q))}{D(q)M^{-1}(q)\hat{J}_m^T(q)R\hat{d}(\hat{r})}, \\
\Delta(t) &= \frac{g(s(\Delta x), \dot{q}, \Delta\theta, \Delta\bar{\theta}_f)}{D(q)M^{-1}(q)\hat{J}_m^T(q)R\hat{d}(\hat{r})(\beta + \kappa(q))}, \\
g(s(\Delta x), \dot{q}, \Delta\theta, \Delta\bar{\theta}_f) &= -\dot{D}(q)\dot{q} + D(q)M^{-1}(q)\{(\frac{1}{2}\dot{M}(q) + S(q, \dot{q}))\dot{q} \\
& \quad + \hat{J}_x^T(q)K_p s(\Delta x) + \hat{J}_x^T(q)K_v \hat{J}_x(q)\dot{q} + Z(q)\Delta\theta + Z_f(q, \lambda_d)\Delta\bar{\theta}_f\}. \tag{3.37}
\end{aligned}$$

Here, γ and β can be chosen so that $\bar{\gamma}(t)$ is positive and therefore $\lambda - \lambda_d$ and its integration is also bounded. From equation (3.17), the boundedness of the force errors implies that $\ddot{q}$ is bounded because $s(\Delta x)$, $\dot{q}$, $\Delta\theta$ and $\Delta\bar{\theta}_f$ are bounded. Hence, $s(\Delta x)$, $\dot{q}$ are uniform continuous and it follows that [61]

$$\Delta x \rightarrow 0, \dot{q} \rightarrow 0 \text{ as } t \rightarrow \infty. \tag{3.38}$$

From equation (3.35), the maximum invariant set satisfies:

$$\begin{aligned}
 & D(q_d)M^{-1}(q_d)(Z(q_d)\Delta\theta_\infty + Z_f(q_d, \lambda_d)\Delta\bar{\theta}_{f,\infty}) \\
 &= D(q_d)M^{-1}(q_d)((\beta\hat{J}_m^T(q_d)R_\infty\hat{d}(\hat{r}_\infty) - (D_1^T(q_d) - \hat{D}_{1,\infty}^T(q_d)))(\lambda_\infty - \lambda_d) \\
 &\quad + \gamma\hat{J}_m^T(q_d)R_\infty\hat{d}(\hat{r}_\infty) \int_0^t (\lambda_\infty - \lambda_d)d\tau, \tag{3.39}
 \end{aligned}$$

where $\Delta\theta_\infty = \theta - \hat{\theta}_\infty$, $\Delta\bar{\theta}_{f,\infty} = \bar{\theta}_f - \hat{\bar{\theta}}_{f,\infty}$, $\hat{\theta}_\infty = \lim_{t \rightarrow \infty} \hat{\theta}(t)$, $\hat{\bar{\theta}}_{f,\infty} = \lim_{t \rightarrow \infty} \hat{\bar{\theta}}_f(t)$, $Z_f(q_d, \lambda_d)\Delta\bar{\theta}_{f,\infty} = (D_1^T(q_d) - \hat{D}_{1,\infty}^T(q_d))\lambda_d$, q_d is a desired joint configuration such that $h(q_d) = x_d$ and is defined for the purpose of analysis only. This implies that $\lambda_\infty = \lambda_d$ because

$$(\lambda_\infty - \lambda_d) + \bar{\gamma}_\infty \int_0^t (\lambda_\infty - \lambda_d)d\tau = \Delta_\infty, \tag{3.40}$$

where

$$\begin{aligned}
 \bar{\gamma}_\infty &= \frac{\gamma}{\beta + \kappa(q_d)}, \\
 \Delta_\infty &= \frac{D(q_d)M^{-1}(q_d)(Z(q_d)\Delta\theta_\infty + Z_f(q_d, \lambda_d)\Delta\bar{\theta}_{f,\infty})}{D(q_d)M^{-1}(q_d)\hat{J}_m^T(q_d)R_\infty\hat{d}(\hat{r}_\infty)(\beta + \kappa(q_d))}. \tag{3.41}
 \end{aligned}$$

Hence it is proved that $\Delta x \rightarrow 0$, $\dot{q} \rightarrow 0$ and $\lambda \rightarrow \lambda_d$ as $t \rightarrow \infty$. △△△

Remark 3.1 When $\Delta x \rightarrow 0$ and $\dot{q} \rightarrow 0$, the vector y reduces to zero but in this case, equation (3.7) is satisfied for any R . Hence, the updating of the rotation matrix is not required when $\Delta x \rightarrow 0$ and $\dot{q} \rightarrow 0$. That is, R approaches a constant matrix when Δx and $\dot{q}$ approach zero.

3.4 Hybrid Vision-force Control for Robot Manipulators using Neural Networks

In section 3.2, an adaptive controller is proposed for constrained robot with uncertain kinematics, dynamics and constraint surface. However, the structure of the constraint surface is assumed to be known. In many applications, the structure of the constraint surface may not be exactly known. In this section, an adaptive setpoint controller using neural networks will be proposed, which does not require exact knowledge of the structure of constraint surface.

3.4.1 Neural Network

Neural network has a well-known property that it can approximate arbitrary nonlinear functions and learn through examples. So it is a powerful tool for control. To deal with the nonlinearity and uncertainty in robot dynamics, some progress has been obtained on online adaptive neural network control [19, 20, 32, 35, 52, 57]. Some systematic approaches for structured dynamic modeling and adaptive control design for robots using neural network can also be found in [21]. Lewis [36] studied the problem of neural network control for robot manipulators and proposed online learning algorithms without preliminary offline training. Sanner and Slotine [58] proposed the use of dynamically structured networks using wavelets for the hidden layer. The nodes that are judged to be noncontributing to the approximation are deleted online. A proof of the stability and convergence properties is also provided. Kim and Lewis [33] developed a neural network output feedback controller for the motion control of robot manipulators without measuring joint velocities. In this controller, an observer was designed to estimate the joint velocities and it is shown that the closed-loop system is uniformly ultimately bounded. This controller does not need exact knowledge of robot dynamics.

3.4 Hybrid Vision-force Control for Robot Manipulators using Neural Networks 38

In order to approximate a function, an approximating function is chosen first, then the weights W are updated according to an algorithm based on the output errors [25]. For this purpose, different types of neural networks architecture can be used, such as multi-layer networks and Radial basis function (RBF) networks. In this thesis, the neural network is designed so that it can be linearly parameterized and update law can be used to update the weights of the neural network online. The RBF network is suitable for this case and is used in this thesis. The structure of the neural networks is shown in Fig. 3.4.

Figure 3.4: An RBF neural network

From Fig. 3.4, the function approximation using a RBF network is [21] [36]

$$f(x) = W\theta(x) + E, \quad (3.42)$$

where W is the matrix of neural network weights, E is called neural network functional approximation error, it generally decreases when the number of neurons increases. $\theta(x)$ is the activation function. There are many kinds of activation functions that can be chosen for RBF networks. It has shown that a linear superposition of

3.4 Hybrid Vision-force Control for Robot Manipulators using Neural Networks 39

Gaussian radial basis function results in an optimal mean square approximation to an unknown function which is infinitely differentiable and whose values are specified by a finite set of points in $\mathbb{R}$ [21]. Therefore, Gaussian RBF networks are used in this section. The Gaussian function is given as [25]

$$\theta(x) = \exp\left[-\frac{(x - \mu)^2}{\sigma^2}\right], \quad (3.43)$$

where μ is called center and σ is distance. The weight matrix is updated online and the update law will be derived from Lyapunov method.

3.4.2 Neural Network Control of Constrained Robots with Uncertainties

In most force control applications, the exact structure and parameters of the constraint function is often difficult to determine. When the constraint surface is uncertain, an estimation of the constraint Jacobian $D^T(q)$ is denoted as $\hat{D}^T(q)$. This results in an modelling error of $D^T(q) - \hat{D}^T(q)$ in the force control system.

Neural network is very useful in compensating the modelling error $D^T(q) - \hat{D}^T(q)$ especially when the structure of the actual environment is unknown or when the uncertain parameters of the environment are not linearly parameterizable. For example, if the structure of a curved surface (as shown in figure 3.5) is difficult to obtain or the uncertain parameter is nonlinear in $D(q)$, it can be estimated as a flat surface or any approximated surface and then neural network is used to compensate for the uncertainty. The desired point on the constraint surface is obtained by cameras.

In this section, the modelling error of the constraint Jacobian is approximated by a

Figure 3.5: Example of constraint uncertainty

neural network as

$$\hat{D}^T(q) - D^T(q) = W_f \theta_f(q) + E_f, \quad (3.44)$$

where

$$\theta_f(q) = \exp\left[\frac{-(q - \mu_f)^T(q - \mu_f)}{\sigma^2}\right]. \quad (3.45)$$

is a vector of activation functions, W_f is a constant matrix of network weights and E_f is approximation error.

Similarly, the gravity term in equation (3.1) can be approximated by a neural network [21] as

$$g(q) = W_g \theta_g(q) + E_g, \quad (3.46)$$

where W_g is the matrix of constant network weights, $\theta_g(q)$ is the vector of activation functions, E_g is approximation error.

The control input for the constrained robot is proposed as:

$$\begin{aligned} \tau = & -\hat{J}_x^T(q) K_p s(\Delta x) - \hat{J}_x^T(q) K_v \hat{J}_x(q) \dot{q} + \hat{W}_g \theta_g(q) + \hat{W}_f \theta_f(q) \lambda - \hat{D}^T(q) \lambda \\ & + \beta \hat{J}_m^T(q) R \hat{d}(\hat{r}) ((\lambda - \lambda_d) + \gamma \int_0^t (\lambda - \lambda_d) d\tau) - K_g \text{sgn}(y) - K_f \text{sgn}(y) \lambda. \end{aligned} \quad (3.47)$$

3.4 Hybrid Vision-force Control for Robot Manipulators using Neural Networks 41

where

$$\|J_x(q) - \hat{J}_x(q)\| \leq p, \quad (3.48)$$

K_p and K_v are diagonal feedback gains for the position error velocity, K_g and K_f are gain matrices for the switching function $\text{sgn}(y)$, γ is a positive constant. R is the rotation matrix defined as in equation (3.7). λ_d is a desired magnitude of the contact force, The neural network weights $\hat{W}_g$ and $\hat{W}_f$ are updated, respectively, by the following update laws:

$$\dot{\hat{W}}_{gk}^T = -L_g^{-1} \theta_g(q) y_k, \quad (3.49)$$

$$\dot{\hat{W}}_{fk}^T = -L_f^{-1} \theta_f(q) y_k \lambda, \quad (3.50)$$

where $\hat{W}_{gk}$ and $\hat{W}_{fk}$ are the k^{th} row vectors of $\hat{W}_g$ and $\hat{W}_f$. y_k is the k^{th} element of the vector y in equation (3.6) and $L_g, L_f \in \mathbb{R}^{i \times i}$ are positive definite matrices.

In the controller described by equations (3.47), (3.49) and (3.50), x is measured by vision system, q is provided by the encoder mounted on the joints and λ is obtained from a force sensor. The Jacobian matrices $J_m(q)$, $J_I(r)$ and constraint surface $\Psi(x)$ are assumed to be uncertain.

Substituting equation (3.47) into equation (3.2) and using equation (3.44) and (3.46), The closed-loop equation is obtained as follows:

$$\begin{aligned} & M(q)\ddot{q} + \left(\frac{1}{2}\dot{M}(q) + S(q, \dot{q})\right)\dot{q} + \hat{J}_x^T(q)K_p s(\Delta x) \\ & + \hat{J}_x^T(q)K_v \hat{J}_x(q)\dot{q} + \Delta W_f \theta_f(q)\lambda + \Delta W_g \theta_g(q) \\ & = \beta \hat{J}_m^T(q) R \hat{d}(\hat{r}) ((\lambda - \lambda_d) + \gamma \int_0^t (\lambda - \lambda_d) d\tau) - K_g \text{sgn}(y) - K_f \text{sgn}(y) \lambda, \end{aligned} \quad (3.51)$$

where $\Delta W_f = W_f - \hat{W}_f$ and $\Delta W_g = W_g - \hat{W}_g$.

Taking inner product of y in equation (3.6) with the closed-loop equation (3.51) using equations (3.7), (3.49), (3.50) and Property 2.2, Property 2.4 of the robot

3.4 Hybrid Vision-force Control for Robot Manipulators using Neural Networks 42

dynamics and equation (3.7), one has:

$$\frac{d}{dt}V = -W, \quad (3.52)$$

where

$$\begin{aligned} V = & \frac{1}{2}\dot{q}^T M(q)\dot{q} + \alpha s^T(\Delta x)(\hat{J}_x^+(q))^T M(q)\dot{q} + \sum_{j=1}^p (k_{pj} + \alpha k_{vj})S_j(\Delta x_j) \\ & + \frac{1}{2} \sum_{k=1}^n \Delta W_{fk} L_{fk} \Delta W_{fk}^T + \frac{1}{2} \sum_{k=1}^n \Delta W_{gk} L_{gk} \Delta W_{gk}^T, \end{aligned} \quad (3.53)$$

$$\begin{aligned} W = & \dot{q}^T \hat{J}_x^T(q) K_v \hat{J}_x(q) \dot{q} + \alpha s^T(\Delta x) K_p s(\Delta x) \\ & - s^T(\Delta x) (K_p + \alpha K_v) (J_x(q) - \hat{J}_x(q)) \dot{q} + y^T (K_g \text{sgn}(y) + E_g + K_f \text{sgn}(y) \lambda + E_f \lambda) \\ & - \alpha (s^T(\Delta x) (\hat{J}_x^+(q))^T (\frac{1}{2} \dot{M}(q) - S(q, \dot{q})) \dot{q} + \alpha \Delta x^T (\hat{J}_x^+(q))^T M(q) \dot{q} \\ & + \dot{x}^T (\hat{J}_x^+(q))^T M(q) \dot{q}), \end{aligned} \quad (3.54)$$

where $\Delta \dot{W}_f = -\dot{W}_f$ and $\Delta \dot{W}_g = -\dot{W}_g$.

Substituting the equation (3.21) into equation (3.53), one similarly show that

$$\begin{aligned} V \geq & \frac{1}{4}\dot{q}^T M(q)\dot{q} + \sum_{j=1}^p (k_{pj} \bar{c}_j + \alpha (k_{vj} \bar{c}_j - \alpha \lambda_m)) \|s(\Delta x)\|^2 \\ & + \frac{1}{2} \sum_{k=1}^n \Delta W_{fk} L_{fk} \Delta W_{fk}^T + \frac{1}{2} \sum_{k=1}^n \Delta W_{gk} L_{gk} \Delta W_{gk}^T, \end{aligned} \quad (3.55)$$

Hence, V is positive definite in $s(\Delta x)$, $\dot{q}$, ΔW_{fk} and ΔW_{gk} since k_v and α can be chosen so that

$$k_{vj} \bar{c}_j - \alpha \lambda_m > 0. \quad (3.56)$$

Therefore, the function V represents a Lyapunov function candidate for the neural-network setpoint control of the robot manipulator with uncertainties.

3.4 Hybrid Vision-force Control for Robot Manipulators using Neural Networks 43

Next, it will be shown that the time derivative of the Lyapunov function is negative definite in $s(\Delta x)$ and $\dot{q}$. As seen from equation (3.52), this is equivalent to showing that W is positive definite in $s(\Delta x)$ and $\dot{q}$. From the last term on the right-hand side of equation (3.54), since $s(\Delta x)$ is bounded, then there exists a constant c_0 so that [2]

$$\begin{aligned} & \alpha |s^T(\Delta x)(\hat{J}_x^+(q))^T(\frac{1}{2}\dot{M}(q) - S(q, \dot{q}))\dot{q} + \alpha s^T(\Delta x)(\hat{J}_x^+(q))^T M(q)\dot{q} \\ & + \dot{s}^T(\Delta x)(\hat{J}_x^+(q))^T M(q)\dot{q}| \leq \alpha c_0 \|\dot{q}\|^2. \end{aligned} \quad (3.57)$$

Substituting the above inequality into equation (3.20) yields

$$\begin{aligned} W \geq & \dot{q}^T \hat{J}_x^T(q) K_v \hat{J}_x(q) \dot{q} + \alpha s^T(\Delta x) K_p s(\Delta x) - s^T(\Delta x) (K_p + \alpha K_v) (J_x(q) - \hat{J}_x(q)) \dot{q} \\ & - \alpha c_0 \|\dot{q}\|^2 + y^T (K_g \text{sgn}(y) + E_g + K_f \text{sgn}(y) \lambda + E_f \lambda). \end{aligned} \quad (3.58)$$

Now, letting $\bar{\Delta} = J_x(q) - \hat{J}_x(q)$, one has

$$\begin{aligned} W \geq & \dot{q}^T \hat{J}_x^T(q) K_v \hat{J}_x(q) \dot{q} - \alpha c_0 \|\dot{q}\|^2 + \alpha s^T(\Delta x) K_p s(\Delta x) \\ & - s^T(\Delta x) (K_p + \alpha K_v) \bar{\Delta} \dot{q} + y^T (K_g \text{sgn}(y) + E_g + K_f \text{sgn}(y) \lambda + E_f \lambda). \end{aligned} \quad (3.59)$$

The existence of the $\bar{\Delta}$ so that W is positive definite can be clearly seen from equation (3.59). In the following development, a sufficient condition will be derived to guarantee the positive definiteness of W . From equation (3.59) one has

$$\begin{aligned} W \geq & (\lambda_{\min}[\hat{J}_x^T(q) K_v \hat{J}_x(q)] - \alpha c_0) \|\dot{q}\|^2 + \alpha k_{p\min} \|s(\Delta x)\|^2 \\ & - p(k_{p\max} + \alpha k_{v\max}) \|\dot{q}\| \|s(\Delta x)\| + \|y\| (k_g - \|E_g\|) + \|y\| \lambda (k_f - \|E_f\|), \end{aligned} \quad (3.60)$$

where $\lambda > 0$ since the end effector is in contact with the constraint surface. When

3.4 Hybrid Vision-force Control for Robot Manipulators using Neural Networks 44

$k_f > \|E_f\|$ and $k_g > \|E_g\|$, one has

$$W \geq (\lambda_1 - \alpha c_0 - \frac{1}{2}p(k_{pmax} + \alpha k_{vmax}))\|\dot{q}\|^2 + (\alpha k_{pmin} - \frac{1}{2}p(k_{pmax} + \alpha k_{vmax}))\|s(\Delta x)\|^2, \quad (3.61)$$

where $\lambda_1 = \lambda_{min}[\hat{J}_x^T(q)K_v\hat{J}_x(q)]$.

Hence if

$$\begin{aligned} \lambda_1 - \alpha c_0 - \frac{1}{2}p(k_{pmax} + \alpha k_{vmax}) &> 0, \\ \alpha k_{pmin} - \frac{1}{2}p(k_{pmax} + \alpha k_{vmax}) &> 0, \end{aligned} \quad (3.62)$$

then $W \geq 0$.

Now the following Theorem can be stated:

Theorem 3.2 *The closed-loop system described by equations (3.51), (3.49) and (3.50), with the uncertain Jacobian matrix and constraint surface $\hat{J}_x(q)$ and $\hat{\Psi}(x)$ described by equation (3.48), rises to the convergence of $(\Delta x, \dot{q})$ to $(0, 0)$ as $t \rightarrow \infty$, if the feedback gains K_v and K_p are chosen to satisfy conditions (3.56) and (3.62). In addition, the contact force λ converges to λ_d as $t \rightarrow \infty$.*

Proof Note that

$$\dot{V} = -W \leq 0, \quad (3.63)$$

where V is positive definite in $s(\Delta x), \dot{q}, \Delta W_{fk}$ and ΔW_{gk} , W is a positive semidefinite function. Integrating both sides of equation (3.52), one has

$$V - V(0) = - \int_0^t W(\tau) d\tau \leq 0. \quad (3.64)$$

Hence, V is bounded and one can conclude that $s(\Delta x), \dot{q}, \Delta W_{fk}$ and ΔW_{gk} are bounded vectors. From equations (3.61) and (3.64), one can conclude that $s(\Delta x), \dot{q}$ are square integrable functions. Next, multiplying both sides of equation (3.51) by

3.4 Hybrid Vision-force Control for Robot Manipulators using Neural Networks 45

$D(q)M^{-1}(q)$ and using $D(q)\ddot{q} = -\dot{D}(q)\dot{q}$, one has

$$\begin{aligned} & -\dot{D}(q)\dot{q} + D(q)M^{-1}(q)\left\{\left(\frac{1}{2}\dot{M}(q, \dot{q}) + S(q, \dot{q})\right)\dot{q} + \hat{J}_x^T(q)K_p s(\Delta x) + \hat{J}_x^T(q)K_v \hat{J}_x(q)\dot{q}\right. \\ & \quad \left. + \Delta W_f \theta_f(q)\lambda + \Delta W_g \theta_g(q) + E_g + E_f \lambda + K_g \text{sgn}(y) + K_f \text{sgn}(y)\lambda\right\} \\ & = D(q)M^{-1}(q)\hat{J}_m^T(q)R\hat{d}(\hat{r})(\beta(\lambda - \lambda_d) + \gamma \int_0^t (\lambda - \lambda_d)d\tau). \end{aligned} \quad (3.65)$$

The force error is subject to

$$\Delta(t) = (\lambda - \lambda_d) + \bar{\gamma}(t) \int_0^t (\lambda - \lambda_d)d\tau, \quad (3.66)$$

where

$$\begin{aligned} \bar{\gamma}(t) &= \frac{\gamma}{\beta + \kappa(q, \Delta W_f)}, \\ \kappa(q, \Delta W_f) &= \frac{-\hat{D}(q)M^{-1}(q)(\Delta W_f \theta_f(q) + E_f + K_f \text{sgn}(y))}{D(q)M^{-1}(q)\hat{J}_m^T(q)R\hat{d}(\hat{r})}, \\ \Delta(t) &= \frac{g(s(\Delta x), \dot{q}, \hat{W}_f, \hat{W}_g)}{\hat{D}(q)M^{-1}(q)\hat{D}^T(q)(\beta + \kappa(q))}, \\ g(s(\Delta x), \dot{q}, W_f, W_g) &= -\dot{D}(q)\dot{q} + D(q)M^{-1}(q)\left\{\left(\frac{1}{2}\dot{M}(q) + S(q, \dot{q})\right)\dot{q}\right. \\ & \quad \left. + \hat{J}_x^T(q)K_p s(\Delta x) + \hat{J}_x^T(q)K_v \hat{J}_x(q)\dot{q} + \Delta W_g \theta_g(q) + \Delta W_g \theta_g(q)\lambda_d\right. \\ & \quad \left. + E_g + E_f \lambda_d + K_g \text{sgn}(y) + K_f \text{sgn}(y)\lambda_d\right\}. \end{aligned} \quad (3.67)$$

Here, γ and β can be chosen so that $\bar{\gamma}(t)$ is positive and therefore $\lambda - \lambda_d$ and its integral are also bounded. From equation (3.51), The boundedness of the force errors implies that $\ddot{q}$ is bounded because $s(\Delta x), \dot{q}, \Delta W_{fk}$ and ΔW_{gk} are bounded. Hence, $s(\Delta x), \dot{q}$ are uniform continuous and it follows that [61]

$$\Delta x \rightarrow 0, \dot{q} \rightarrow 0 \text{ as } t \rightarrow \infty. \quad (3.68)$$

From equation (3.65), the maximum invariant set satisfies:

$$\begin{aligned} & \hat{D}(q_d)M^{-1}(q_d)(\Delta W_{f\infty}\theta_f(q_d)\lambda_\infty + \Delta W_{g\infty}\theta_g(q_d) + E_g + E_f\lambda_\infty) \\ &= D(q_d)M^{-1}(q_d)\hat{J}_m^T(q_d)R_\infty\hat{d}(\hat{r}_\infty)(\beta(\lambda_\infty - \lambda_d) + \gamma \int_0^t (\lambda_\infty - \lambda_d)d\tau). \end{aligned} \quad (3.69)$$

where q_d is a desired joint configuration such that $h(q_d) = x_d$ and is defined for the purpose of analysis only. This implies that $\lambda_\infty = \lambda_d$ because

$$(\lambda_\infty - \lambda_d) + \bar{\gamma}_\infty \int_0^t (\lambda_\infty - \lambda_d)d\tau = \Delta_\infty, \quad (3.70)$$

where

$$\begin{aligned} \bar{\gamma}_\infty &= \frac{\gamma}{\beta + \kappa(q_d, \Delta W_{f\infty})}, \\ \kappa(q_d, \Delta W_{f\infty}) &= \frac{-D(q_d)M^{-1}(q_d)(\Delta W_{f\infty}\theta_f(q_d) - E_f)}{D(q_d)M^{-1}(q_d)\hat{J}_m^T(q_d)R_\infty\hat{d}(\hat{r}_\infty)} \\ \Delta_\infty &= \frac{\hat{D}(q_d)M^{-1}(q_d)(\Delta W_{f\infty}\theta_f(q_d)\lambda_\infty + \Delta W_{g\infty}\theta_g(q_d) + E_g + E_f\lambda_\infty)}{D(q_d)M^{-1}(q_d)\hat{J}_m^T(q_d)R_\infty\hat{d}(\hat{r}_\infty)(\beta + \kappa(q_d, \Delta W_{f\infty}))}. \end{aligned} \quad (3.71)$$

Hence it is proved that $\Delta x \rightarrow 0$, $\dot{q} \rightarrow 0$ and $\lambda \rightarrow \lambda_d$ as $t \rightarrow \infty$.

Remark 3.2 For practical applications like robot control, chattering must be eliminated because it may excite the high frequency dynamics neglected in the modelling. This can be achieved by smooth implementation of the switching law. Such a smoothed switching law would have little alteration in the dynamic behaviour of the system. One of the methods to smooth switching law is the concept of boundary layer [61].

3.5 Simulation Results

In this section, simulation results are presented to illustrate the performance of the proposed controllers. A two-link manipulator whose end-effector is required to move

on a constraint surface is considered (see figure 3.6). A camera is placed at a fixed distance from the robot. The manipulator Jacobian matrix $J_m(q)$ mapping from joint space to Cartesian space for this manipulator is given by

$$J_m(q) = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{bmatrix}, \quad (3.72)$$

where $s_1 = \sin(q_1)$, $c_1 = \cos(q_1)$, $s_{12} = \sin(q_1 + q_2)$, $c_{12} = \cos(q_1 + q_2)$ and l_1, l_2 are the lengths of the first and second links, respectively.

The image Jacobian matrix $J_I(q)$ mapping from Cartesian space to image space is given by

$$J_I = \frac{f_1}{z - f_1} \begin{bmatrix} \beta_1 & 0 \\ 0 & \beta_2 \end{bmatrix} \begin{bmatrix} \cos\delta & \sin\delta \\ -\sin\delta & \cos\delta \end{bmatrix}, \quad (3.73)$$

where β_1, β_2 denote the scaling factors in pixels/m, δ is the angle of rotation of the vision coordinates relative to Cartesian coordinates. z is the perpendicular distance between the robot and the camera, f_1 is the focal length of the camera.

Figure 3.6: A two-link robot in contact with a constraint surface

3.5.1 Vision-force Control using Regressors

In this simulation, a flat constraint surface is considered, the constraint surface is described by

$$\Psi(r(q)) = x + \gamma_s y + c = l_1 c_1 + l_2 c_{12} + \gamma_s (l_1 s_1 + l_2 s_{12}) + c = 0, \quad (3.74)$$

where γ_s and c are constant. Then,

$$\frac{\partial \Psi(r)}{\partial r} = (1, \gamma_s) \quad (3.75)$$

and

$$D(q) = \frac{\partial \Psi(r)}{\partial r} J_m(q) = [-l_1 s_1 - l_2 s_{12} + \gamma_s l_1 c_1 + \gamma_s l_2 c_{12}, \\ -l_2 s_{12} + \gamma_s l_2 c_{12}]. \quad (3.76)$$

Therefore, $D^T(q)\lambda$ can be written as

$$D^T(q)\lambda = \lambda \begin{bmatrix} -s_1 & -s_{12} & c_1 & c_{12} \\ 0 & -s_{12} & 0 & c_{12} \end{bmatrix} \begin{bmatrix} \theta_{f,1} \\ \theta_{f,2} \\ \theta_{f,3} \\ \theta_{f,4} \end{bmatrix} = Z_f(q, \lambda) \theta_f, \quad (3.77)$$

where $\theta_{f,1} = l_1, \theta_{f,2} = l_2, \theta_{f,3} = \gamma_s l_1, \theta_{f,4} = \gamma_s l_2$. In the simulation, the exact masses of the two links are set to 17.4 and 4.8 kg, the exact lengths l_1 and l_2 of the links are set to 0.432 and 0.433 m, f_1 is chosen as 50mm, z is chosen as 0.55m and δ is chosen as $\pi/4$. $\beta_1 = \beta_2 = 10000\text{pixel/m}$ and γ_s is 0.5.

When the kinematics and the constraint are uncertain, one has

$$\hat{J}_m(q) = \begin{bmatrix} -\hat{l}_1 s_1 - \hat{l}_2 s_{12} & -\hat{l}_2 s_{12} \\ \hat{l}_1 c_1 + \hat{l}_2 c_{12} & \hat{l}_2 c_{12} \end{bmatrix}, \quad (3.78)$$

$$\hat{J}_I = \frac{\hat{f}_1}{\hat{z} - \hat{f}_1} \begin{bmatrix} \hat{\beta}_1 & 0 \\ 0 & \hat{\beta}_2 \end{bmatrix} \begin{bmatrix} \cos \hat{\delta} & \sin \hat{\delta} \\ -\sin \hat{\delta} & \cos \hat{\delta} \end{bmatrix}, \quad (3.79)$$

$$\hat{D}(q) = [-\hat{l}_1 s_1 - \hat{l}_2 s_{12} + \hat{\gamma}_s \hat{l}_1 c_1 + \hat{\gamma}_s \hat{l}_2 c_{12}, -\hat{l}_2 s_{12} + \hat{\gamma}_s \hat{l}_2 c_{12}].$$

where $\hat{l}_1, \hat{l}_2, \hat{f}_1, \hat{z}, \hat{\beta}_1, \hat{\beta}_2, \hat{\delta}$ and $\hat{\gamma}_s$ denote the estimated values of $l_1, l_2, f_1, z, \beta_1, \beta_2, \delta$ and γ_s , respectively.

To demonstrate the advantages of adaptive vision and force controller, some comparisons of the proposed controller are made with the adaptive PD controller in Cartesian space and the PD controller with visual feedback.

First, an adaptive PD controller in Cartesian space with uncertain constraint surface is considered. When the constraint surface is uncertain, the actual desired point on the actual constraint surface becomes uncertain. In this case, the robot is required to move from the initial point $[0.5, 0]$ m to the estimated desired point $[0.62, 0.2]$ m, note that the estimated desired point is not on the constraint surface because of the uncertainties, hence the end-effector cannot reach the desired point. In this simulation, the estimated parameters are set as $\hat{l}_1 = 0.4m, \hat{l}_2 = 0.5m$ and $\hat{\gamma}_s = 1/4$. The control gains are set as $\alpha = 1.5, L = 3.3I, \bar{L} = 0.2I, K_p = 350I, K_v = 220I, \beta = 0.1, \gamma = 10$. In figures 3.7, 3.8 and 3.9, the simulation results show that there are constant position errors in X and Y axis because of the distance between the desired point and constraint surface.

Next, a PD controller with visual feedback with uncertainties in kinematics and constraint surface is considered. In this simulation, the estimated parameters are set as $\hat{l}_1 = 0.4m, \hat{l}_2 = 0.5m, \hat{f}_1 = 40mm, \hat{z} = 0.5m, \hat{\beta}_1 = \hat{\beta}_2 = 8000, \hat{\delta} = \pi/6$

Figure 3.7: Position error in X axis of Cartesian space controller

Figure 3.8: Position error in Y axis of Cartesian space controller

Figure 3.9: Force error of Cartesian space controller

and $\hat{\gamma}_s = 1/4$. The control gains are set as $\alpha = 1.5 \times 10^{-6}$, $K_p = 0.00035I$, $K_v = 0.00022I$, $\beta = 0.1$, $\gamma = 10$. It can be seen from the simulation results in figures 3.10, 3.11 and 3.12 that the controller fails in moving the end-effector to the desired position with these uncertainties.

In the following development, the simulation results of the proposed adaptive vision and force controller will be presented. In this simulation, the robot was required to move from the initial point [354,146] pixels to the desired point [566,217] pixels, the offset is [0,500] pixels. The desired contact force is set as 10 N. In this simulation, the estimated parameters are set as $\hat{l}_1 = 0.4m$, $\hat{l}_2 = 0.5m$, $\hat{f}_1 = 40mm$, $\hat{z} = 0.5m$, $\hat{\beta}_1 = \hat{\beta}_2 = 8000$, $\hat{\delta} = \pi/6$ and $\hat{\gamma}_s = 1/4$. The control gains are set as $\alpha = 1.5 \times 10^{-6}$, $L = 3.3I$, $\bar{L} = 0.2I$, $K_p = 0.00035I$, $K_v = 0.00022I$, $\beta = 0.1$, $\gamma = 10$. The simulation results are shown in figures 3.13, 3.14 and 3.15. The results show the effectiveness of the proposed controller in dealing with uncertainties in the kinematics, dynamics, camera model and constraint surface and convergence of the image and force errors are guaranteed.

Next, the kinematics and camera parameters uncertainties are as $\hat{l}_1 = 0.25m$, $\hat{l}_2 = 0.4m$, $\hat{f}_1 = 30mm$, $\hat{z} = 0.4m$, $\hat{\beta}_1 = \hat{\beta}_2 = 5000$, $\hat{\delta} = 0$ and $\hat{\gamma}_s = 1/4$. Figures 3.16, 3.17 and 3.18 show the system response with the same control gains as used in the first simulation. As seen from the simulation results, although the transient performance is worse than the first simulation, the image and force errors converge to 0 even in the presence of larger uncertainties in the system.

3.5.2 Vision-force Control using Neural Networks

In the previous section, the simulation results of adaptive vision-force control using regressors are presented. However, the structure of the constraint surface is assumed to be known and the parameters can be linearly separated. In this simulation,

Figure 3.10: Image error in X axis of PD controller

Figure 3.11: Image error in Y axis of PD controller

Figure 3.12: Force error of PD controller

Figure 3.13: Image error in X axis

Figure 3.14: Image error in Y axis

Figure 3.15: Force error

Figure 3.16: Image error in X axis with larger uncertainties

Figure 3.17: Image error in Y axis with larger uncertainties

Figure 3.18: Force error with larger uncertainties

Figure 3.19: A two-link robot in contact with a constraint surface

unknown structure of constraint surface is considered. The constraint surface is described by

$$\Psi_1(x) = \sin(ax_1 + b) - x_2 = 0, \quad (3.80)$$

hence

$$\frac{\partial \Psi_1(x)}{\partial x} = (a \cos(ax_1 + b), -1), \quad (3.81)$$

and

$$\begin{aligned} D(q) &= \frac{\partial \Psi_1(x)}{\partial x} J(q) \\ &= [-a \cos(a(l_1 c_1 + l_2 c_{12}) + b)(l_1 s_1 + l_2 s_{12}) - l_1 c_1 - l_2 c_{12}, \\ &\quad -a \cos(a(l_1 c_1 + l_2 c_{12}) + b)l_2 s_{12} - l_2 c_{12}]. \end{aligned} \quad (3.82)$$

Note that in this constraint function, the parameters in $D(q)$ can not be linearly separated.

In this simulation, the unknown surface is estimated as a plane as follows:

$$\hat{\Psi}_1(x) = x + \hat{\gamma}_s y - c = 0. \quad (3.83)$$

hence

$$\frac{\partial \hat{\Psi}_1(x)}{\partial x} = (1, \hat{\gamma}_s). \quad (3.84)$$

The robot was required to move from the initial point [508, 273] pixels to the desired point [732, 384] pixels. The desired contact force was set as 15 N. In this simulation, the estimated parameters were set as $\hat{l}_1 = 0.4m, \hat{l}_2 = 0.5m, \alpha = 1.82 \times 10^{-6}, K_p = 0.0003I, K_v = 0.00017I, \beta = 0.01, \gamma = 14, \hat{\gamma}_s = -0.2294, \hat{f}_1 = 40mm, \hat{z} = 0.5m, \hat{\beta}_1 = \hat{\beta}_2 = 8000$ and $\hat{\delta} = \pi/6$, respectively.

In this simulation, Gaussian RBF neural networks with input q were used. The centers were chosen so that they were evenly distributed to span the input space of the network. The distance of neural networks was fixed at 1.2 and the number of neurons was set as 40. The gains for the networks were chosen as $L_f = 300I, L_g = 0.35I, k_g = 0.01, k_f = 0.0005$.

The simulation results are shown in figures 3.20, 3.21 and 3.22. The results show the effectiveness of the proposed neural network controller in dealing with uncertainties in the kinematics, dynamics and constraint surface and convergence of the position and force errors are guaranteed.

Next, the estimated parameters are set as $\hat{l}_1 = 0.25m, \hat{l}_2 = 0.4m, \hat{f}_1 = 30mm, \hat{z} = 0.4m, \hat{\beta}_1 = \hat{\beta}_2 = 5000, \hat{\delta} = 0$ and $\hat{\gamma}_s = 1/4$. Figures 3.23, 3.24 and 3.25 show the system response with the same control gains as used in the previous simulation. As seen from the simulation results, the image and force errors converge to 0 even in the presence of larger uncertainties in the system. Compared with the controller using regressors in the previous section, the neural network controller has better transient

Figure 3.20: Image error in X axis using neural network controller

Figure 3.21: Image error in Y axis using neural network controller

Figure 3.22: Force error using neural network controller

performance in dealing with larger uncertainties.

Remark 3.3 The simulation was implemented in Matlab on a Pentium 4 PC. If the algorithm is implemented on a robot, it should be transplanted to C/C++, which can significantly improve the computation time. In addition, the computation time of the controller will be much smaller because there are no numerical solutions in actual implementation. Another computational intensive process which demands further optimizations is image processing. Image data has a very high dimensionality. For instance, a typical VGA image at the resolution of 640×480 , its dimensionality is $640 \times 480 \times 3$. Any operation applied to this image must go through each pixel value, which makes image processing computationally intensive. One more common practical issue is the sensor noise. The results from camera and force sensor are often noisy; it must be correctly modeled in the control loop. The most popular error model is the Gaussian noise model, which is a second moment description for the uncertainty distribution of the measurement uncertainty. Nevertheless, more sophisticated models, such as Sum of Gaussian, are also available.

3.6 Conclusion

In this chapter, the stability problem of hybrid vision-force control system with uncertain kinematics, dynamics, camera model and constraint surface has been formulated and solved. An adaptive feedback control law has been proposed. A gravity regressor and a force regressor have also been proposed for gravitational and force compensation when the gravitational and forces are uncertain. An adaptive neural network controller has also been proposed to deal with uncertain structure of the constraint surface. New Lyapunov functions have been presented for the stability analysis of the control problem. Sufficient conditions for choosing the feedback gains to guarantee the stability have been presented. It has been shown that the stabil-

Figure 3.23: Image error in X axis with larger uncertainties using neural network controller

Figure 3.24: Image error in Y axis with larger uncertainties using neural network controller

Figure 3.25: Force error with larger uncertainties using neural network controller

ity can be achieved in presence of the above-mentioned uncertainties. Simulation results have been presented to illustrate the performance of the proposed control laws.

Chapter 4

Adaptive Jacobian Motion and Force Tracking Control for Constrained Robot with Uncertainties

In the previous chapter, adaptive controllers have been proposed for setpoint control of manipulators with uncertainties. However, in some applications, the robots are required to track position and force trajectories. Although many motion and force trajectory tracking controllers have been proposed in the literature, the trajectory tracking control problem of constrained robot with uncertain parameter and structure of constraint surface is not well studied. When the robots work in an outdoor environment, the parameters and structure of the environment are difficult to be modeled exactly.

This chapter is dedicated to the research of visually-servoed trajectory tracking problem of constrained robots with uncertain kinematics, dynamics, camera pa-

rameters and constraint surface. In section 4.2, An adaptive Jacobian controller is proposed for robots with uncertainties. The trajectory tracking control problem in the presence of uncertainties is formulated and solved based on a Lyapunov-like analysis. It is shown that the proposed controllers can follow desired position and force trajectories with the uncertainties. In section 4.3, a neural network Jacobian controller is proposed and boundedness of position and force tracking errors is proved. Simulation results are presented to illustrate the performance of the proposed controllers.

4.1 Adaptive Vision-force Tracking Control

Consider a vision-force control system consisting of a robot manipulator and camera(s) fixed in the work space. In this system, the end effector is in contact with a constraint surface. As discussed in Chapter 2, the equations of motion of the constrained robot can be expressed in joint coordinates as [2] [60],

$$M(q)\ddot{q} + \left(\frac{1}{2}\dot{M}(q) + S(q, \dot{q})\right)\dot{q} + g(q) = \tau + D^T(q)\lambda, \quad (4.1)$$

where $D(q) = \frac{(\partial\Psi(r)/\partial r)^T}{\|\partial\Psi(r)/\partial r\|} J_m(q)$ is a Jacobian of the constraint function such that

$$D(q)\dot{q} = 0. \quad (4.2)$$

In this section, an adaptive vision and force tracking controller is proposed. Using property 2.6 and equation (3.10), the dynamic model is linear in a set of physical parameters $\theta_d = (\theta_{d1}, \dots, \theta_{dp})^T$ and $\theta_f = (\theta_{f1}, \dots, \theta_{fi})^T$ as

$$M(q)\ddot{q} + \left(\frac{1}{2}\dot{M}(q) + S(q, \dot{q})\right)\dot{q} + g(q) = Y_d(q, \dot{q}, \ddot{q})\theta_d \quad (4.3)$$

$$D^T(q)\lambda = Y_f(q, f)\theta_f \quad (4.4)$$

where $Y_d(\cdot) \in \mathbb{R}^{n \times p}$ is the dynamic regressor matrix and $Y_f(q, \lambda) \in \mathbb{R}^{n \times i}$ is the force regressor matrix.

From property 2.1, the right hand side of equation (2.6) is linear in a set of kinematic parameters $\theta_{ki} = (\theta_{ki1}, \dots, \theta_{kii})^T$, such as link lengths, joint offsets and camera parameters [13]. Then equation (2.6) can be expressed as,

$$\dot{x} = J_I(r)J_m(q)\dot{q} = J(q)\dot{q} = Y_{ki}(q, \dot{q})\theta_{ki}. \quad (4.5)$$

where $Y_{ki}(q, \dot{q}) \in \mathbb{R}^{m \times q}$ is the kinematic regressor matrix.

When the kinematics is uncertain, the parameters of the Jacobian matrix are uncertain, and the parameters are estimated as:

$$\hat{\dot{x}} = \hat{J}(q, \hat{\theta}_{ki})\dot{q} = Y_{ki}(q, \dot{q})\hat{\theta}_{ki}. \quad (4.6)$$

where $\hat{J}(q, \hat{\theta}_{ki})$ is estimations of $J(q)$. The parameters $\hat{\theta}_{ki}$ will be updated by parameter update law to be defined later.

Let us define a vector $\dot{x}_r \in \mathbb{R}^m$ as,

$$\dot{x}_r = (\dot{x}_d - \alpha \Delta x) + \beta(\hat{J}_m(q, \hat{\theta}_f)\hat{J}^+(q, \hat{\theta}_{ki}))^{-1}Rd(r)\Delta F, \quad (4.7)$$

where α and β are positive constants, $x_d(t) \in \mathbb{R}^m$ is the desired image motion trajectory and $\dot{x}_d(t) \in \mathbb{R}^m$ is the desired speed trajectory, $\Delta x = x - x_d$ is image motion tracking error, $\hat{J}(q, \hat{\theta}_{ki})$ is an estimation of $J(q)$, $\hat{J}^+(q, \hat{\theta}_{ki}) = \hat{J}^T(q, \hat{\theta}_{ki})(\hat{J}(q, \hat{\theta}_{ki})\hat{J}^T(q, \hat{\theta}_{ki}))^{-1}$ is the pseudo inverse of the estimated Jacobian matrix and $\Delta F = \int_0^t (\lambda(\sigma) - \lambda_d(\sigma))d\sigma$, $\lambda_d(t)$ is the desired force trajectory and R is a rotation matrix which will be defined later. In the presence of kinematic and constraint uncertainties, $d(r)$ is uncertain. In this chapter, the normal direction $d(r)$ to the constraint surface is assumed be obtained from force measurement [2] [12] [44]. However, differentiation

of the possibly noisy contact force should be avoided in the controller. The desired trajectory on the uncertain constraint surface is defined in image space.

Differentiating equation (4.7) with respect to time yields

$$\begin{aligned}\ddot{x}_r = & (\ddot{x}_d - \alpha\Delta\dot{x}) + \beta(\hat{J}_m(q, \hat{\theta}_f)\hat{J}^+(q, \hat{\theta}_{ki}))^{-1}Rd(r)\Delta\lambda \\ & + \beta(\hat{J}_m(q, \hat{\theta}_f)\hat{J}^+(q, \hat{\theta}_{ki}))^{-1}Rd(r, \dot{r})\Delta F \\ & + \beta(\hat{J}_m(q, \hat{\theta}_f)\hat{J}^+(q, \hat{\theta}_{ki}))^{-1}\dot{R}d(r)\Delta F + \beta\dot{J}_I R d(r)\Delta F,\end{aligned}\quad (4.8)$$

where $\dot{J}_I = \frac{d}{dt}((\hat{J}_m(q, \hat{\theta}_f)\hat{J}^+(q, \hat{\theta}_{ki}))^{-1})$. To avoid the use of force derivative, one defines

$$\begin{aligned}\hat{\ddot{x}}_r = & (\ddot{x}_d - \alpha\Delta\dot{x}) + \beta(\hat{J}_m(q, \hat{\theta}_f)\hat{J}^+(q, \hat{\theta}_{ki}))^{-1}Rd(r)\Delta\lambda \\ & + \beta(\hat{J}_m(q, \hat{\theta}_f)\hat{J}^+(q, \hat{\theta}_{ki}))^{-1}\dot{R}d(r)\Delta F + \beta\dot{J}_I R d(r)\Delta F.\end{aligned}\quad (4.9)$$

without using $\dot{d}(r, \dot{r})$ or the derivative of the contact force.

In order to prove the stability of the vision-force tracking system, an adaptive sliding vector is defined using equation (4.7) as,

$$\hat{s}_x = \hat{\dot{x}} - \dot{x}_r = \hat{J}(q, \hat{\theta}_{ki})\dot{q} - \dot{x}_r, \quad (4.10)$$

Differentiating the above equation with respect to time, one has,

$$\dot{\hat{s}}_x = \hat{\ddot{x}} - \ddot{x}_r = \hat{J}(q, \hat{\theta}_{ki})\ddot{q} + \dot{\hat{J}}(q, \hat{\theta}_{ki})\dot{q} - \ddot{x}_r, \quad (4.11)$$

Next, let

$$\dot{q}_r = \hat{J}^+(q, \hat{\theta}_{ki})\dot{x}_r + (I_n - \hat{J}^+(q, \hat{\theta}_{ki})\hat{J}(q, \hat{\theta}_{ki}))\psi, \quad (4.12)$$

where $\psi \in \mathbb{R}^n$ is minus the gradient of the convex function to be optimized [48].

From equation (4.12), one has,

$$\begin{aligned}\ddot{q}_r = & \hat{J}^+(q, \hat{\theta}_{ki})\ddot{x}_r + \dot{\hat{J}}^+(q, \hat{\theta}_{ki})\dot{x}_r + (I_n - \hat{J}^+(q, \hat{\theta}_{ki})\hat{J}(q, \hat{\theta}_{ki}))\dot{\psi} \\ & - (\dot{\hat{J}}^+(q, \hat{\theta}_{ki})\hat{J}(q, \hat{\theta}_{ki}) + \hat{J}^+(q, \hat{\theta}_{ki})\dot{\hat{J}}(q, \hat{\theta}_{ki}))\psi, \end{aligned} \quad (4.13)$$

Next an estimation of $\ddot{q}_r$ is defined as follows

$$\begin{aligned}\hat{\ddot{q}}_r = & \hat{J}^+(q, \hat{\theta}_{ki})\hat{\ddot{x}}_r + \dot{\hat{J}}^+(q, \hat{\theta}_{ki})\dot{x}_r + (I_n - \hat{J}^+(q, \hat{\theta}_{ki})\hat{J}(q, \hat{\theta}_{ki}))\dot{\psi} \\ & - (\dot{\hat{J}}^+(q, \hat{\theta}_{ki})\hat{J}(q, \hat{\theta}_{ki}) + \hat{J}^+(q, \hat{\theta}_{ki})\dot{\hat{J}}(q, \hat{\theta}_{ki}))\psi, \end{aligned} \quad (4.14)$$

From equations (4.13) and (4.14), one has

$$\ddot{q}_r = \hat{\ddot{q}}_r + \beta \hat{J}^+(q, \hat{\theta}_{ki})(\hat{J}_m(q, \hat{\theta}_f)\hat{J}^+(q, \hat{\theta}_{ki}))^{-1} R \dot{d}(r, \dot{r}) \Delta F. \quad (4.15)$$

where $r = h(q)$ and $\dot{r} = J_m(q)\dot{q}$ as seen from equations (2.2) and (2.3), respectively. The unknown parameters in $\dot{d}(h(q), J_m(q)\dot{q})$ will be updated by a dynamic update law to be defined later.

Hence, an adaptive sliding vector is defined in joint space as,

$$s = \dot{q} - \dot{q}_r, \quad (4.16)$$

and

$$\dot{s} = \ddot{q} - \ddot{q}_r. \quad (4.17)$$

Multiplying both side of equation (4.16) by $\hat{J}(q, \hat{\theta}_{ki})$ and using equation (4.10), one has

$$\hat{J}(q, \hat{\theta}_{ki})s = \hat{J}(q, \hat{\theta}_{ki})\dot{q} - \dot{x}_r = \hat{s}_x, \quad (4.18)$$

Substituting equation (4.15) into equation (4.17) yields

$$\dot{s} = \ddot{q} - \hat{\ddot{q}}_r - \beta \hat{J}^+(q, \hat{\theta}_{ki})(\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^{-1} R \dot{d}(r, \dot{r}) \Delta F. \quad (4.19)$$

Substitute equations (4.16) and (4.19) into equation (4.1) to get,

$$\begin{aligned} M(q)\dot{s} + (\tfrac{1}{2}\dot{M}(q) + S(q, \dot{q}))s + M(q)\hat{\ddot{q}}_r + (\tfrac{1}{2}\dot{M}(q) + S(q, \dot{q}))\dot{q}_r + g(q) \\ + \beta M(q)\hat{J}^+(q, \hat{\theta}_{ki})(\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^{-1} R \dot{d}(r, \dot{r}) \Delta F \\ = \tau + D^T(q)\lambda, \end{aligned} \quad (4.20)$$

The last four term on the left hand side of equation (4.20) are linear in a set of dynamics parameters $\bar{\theta}_d$ and can be expressed as

$$\begin{aligned} M(q)\hat{\ddot{q}}_r + (\tfrac{1}{2}\dot{M}(q) + S(q, \dot{q}))\dot{q}_r + g(q) \\ + \beta M(q)\hat{J}^+(q, \hat{\theta}_{ki})(\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^{-1} R \dot{d}(r, \dot{r}) \Delta F \\ = \bar{Y}_d(q, \dot{q}, \dot{q}_r, \hat{\ddot{q}}_r, \hat{\theta}_f, \hat{\theta}_{ki})\bar{\theta}_d. \end{aligned} \quad (4.21)$$

Then the dynamics equation (4.20) can be expressed as:

$$M(q)\dot{s} + (\tfrac{1}{2}\dot{M}(q) + S(q, \dot{q}))s + \bar{Y}_d(q, \dot{q}, \dot{q}_r, \hat{\ddot{q}}_r, \hat{\theta}_f, \hat{\theta}_{ki})\bar{\theta}_d = \tau + D^T(q)\lambda, \quad (4.22)$$

The vision and force tracking controller is proposed as:

$$\begin{aligned} \tau = -\hat{J}^T(q, \hat{\theta}_{ki})K(\Delta\hat{x} + \alpha\Delta x) + \bar{Y}_d(q, \dot{q}, \dot{q}_r, \hat{\ddot{q}}_r, \hat{\theta}_f, \hat{\theta}_{ki})\hat{\bar{\theta}}_d \\ - Y_f(q, f)\hat{\theta}_f + \hat{J}_m^T(q, \hat{\theta}_f)Rd(r)(\Delta\lambda + \gamma\Delta F), \end{aligned} \quad (4.23)$$

where $\Delta\hat{x} = \hat{x} - \dot{x}_d$, $K \in \mathbb{R}^{m \times m}$ is a positive definite gain matrix, γ is a positive

constant. The estimated parameters $\hat{\theta}_d$, $\hat{\theta}_f$ and $\hat{\theta}_{ki}$ are updated by,

$$\dot{\hat{\theta}}_d = -L_d \bar{Y}_d^T(q, \dot{q}, \dot{q}_r, \hat{q}_r, \hat{\theta}_f, \hat{\theta}_{ki})s, \quad (4.24)$$

$$\dot{\hat{\theta}}_f = L_f Y_f^T(q, f)s, \quad (4.25)$$

$$\dot{\hat{\theta}}_{ki} = 2L_{ki} Y_{ki}^T(q, \dot{q})K(\Delta \dot{x} + \alpha \Delta x), \quad (4.26)$$

In the above controller, R is a rotation matrix designed [80] so that

$$s_{xN}^T R d(r) = 0, \quad (4.27)$$

where

$$\begin{aligned} s_{xN} &= \{\beta K(\Delta \dot{x} + \alpha \Delta x)^T (\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^{-1} \Delta F + s_m^T (\Delta \lambda + \gamma \Delta F)\}^T \\ s_m &= \hat{J}_m(q, \hat{\theta}_f) \{\dot{q} - \hat{J}^+(q, \hat{\theta}_{ki})(\dot{x}_d - \alpha \Delta x) - (I_n - \hat{J}^+(q, \hat{\theta}_{ki}) \hat{J}(q, \hat{\theta}_{ki}))\psi\}, \end{aligned} \quad (4.28)$$

In general, $\bar{s}_x$ can be partitioned as $\bar{s}_x = (\bar{s}_{xp}^T, \bar{s}_{xo}^T)^T$, where $\bar{s}_{xp}$ is the position vector and $\bar{s}_{xo}$ is the orientation vector. Therefore, $d(r)$ can also be partitioned as $(d_p^T(r), d_o^T(r))^T$. Hence the rotation matrix R can be introduced as,

$$R = \begin{bmatrix} R_p & 0 \\ 0 & R_o \end{bmatrix}, \quad (4.29)$$

where n_p is a unit vector normal to both the vectors $\bar{s}_{xp}$ and $d_p(r)$ (see figure 4.1). ϕ_p is the angle between $\bar{s}_{xp}^T R_p$ and $d_p(r)$, which can be determined from the angle ϕ between $\bar{s}_{xp}$ and $d_p(r)$ (see figure 4.1). The rotation matrix R_p is to rotate the vector $\bar{s}_{xp}$ about the axis n_p , so that the vector $\bar{s}_{xp}^T R_p$ is perpendicular to the vector $d_p(r)$, as shown in Figure 4.1. Here the symbol $\times$ means cross product. The rotation matrix R_o can be similarly designed. However, since the constraint surface is usually independent of $d_o(r)$, the rotation matrix R_o can be set as an identity matrix. Note

that when $\bar{s}_r$ reduces to zero, R can be set as any value because equation (4.27) is always satisfied when $\bar{s}_x = 0$.

Figure 4.1: Illustration of rotation matrix

Substituting equation (4.23) into equation (4.22), the closed-loop equation is obtained as

$$\begin{aligned} M(q)\dot{s} + (\frac{1}{2}\dot{M}(q) + S(q, \dot{q}))s + \hat{J}^T(q, \hat{\theta}_{ki})K(\Delta\hat{x} + \alpha\Delta x) + \bar{Y}_d(q, \dot{q}, \dot{q}_r, \hat{q}_r, \hat{\theta}_f, \hat{\theta}_{ki})\Delta\theta_d \\ = Y_f(q, f)\Delta\theta_f + \hat{J}_m^T(q, \hat{\theta}_f)Rd(r)(\Delta\lambda + \gamma\Delta F), \end{aligned} \quad (4.30)$$

where $\Delta\theta_d = \bar{\theta}_d - \hat{\theta}_d$ and $\Delta\theta_f = \theta_f - \hat{\theta}_f$.

To carry out the stability analysis, a Lyapunov-like function candidate V is defined as:

$$V = \frac{1}{2}s^T M(q)s + \alpha\Delta x^T K\Delta x + \frac{1}{2}\Delta\theta_d^T L_d^{-1}\Delta\theta_d + \frac{1}{2}\Delta\theta_f^T L_f^{-1}\Delta\theta_f + \frac{1}{2}\Delta\theta_{ki}^T L_{ki}^{-1}\Delta\theta_{ki} + \frac{1}{2}\beta\Delta F^2. \quad (4.31)$$

where $\Delta\theta_{ki} = \theta_{ki} - \hat{\theta}_{ki}$. Differentiating V with respect of time yields,

$$\begin{aligned}\dot{V} = & s^T M(q) \dot{s} + \frac{1}{2} s^T \dot{M}(q) s + 2\alpha \Delta x^T K \Delta \dot{x} \\ & - \Delta \theta_d^T L_d^{-1} \dot{\hat{\theta}}_d - \Delta \theta_{ki}^T L_{ki}^{-1} \dot{\hat{\theta}}_{ki} - \Delta \theta_f^T L_f^{-1} \dot{\hat{\theta}}_f + \beta \Delta F \Delta \lambda.\end{aligned}\quad (4.32)$$

Substituting equations (4.24), (4.25) and (4.30) into the above equation and using equation (4.18), one has,

$$\begin{aligned}\dot{V} = & -\hat{s}_x^T K(\Delta \hat{x} + \alpha \Delta x) + s^T \hat{J}_m^T(q, \hat{\theta}_f) R d(r) (\Delta \lambda + \gamma \Delta F) \\ & + 2\alpha \Delta x^T K \Delta \dot{x} - \Delta \theta_{ki}^T L_{ki}^{-1} \dot{\hat{\theta}}_{ki} + \beta \Delta F \Delta \lambda.\end{aligned}\quad (4.33)$$

From equations (4.16), (4.12), (4.7) and (4.28), note that

$$\begin{aligned}\hat{J}_m(q, \hat{\theta}_f) s = & \hat{J}_m(q, \hat{\theta}_f) \dot{q} - \hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}) (\dot{x}_d - \alpha \Delta x) \\ & - \hat{J}_m(q, \hat{\theta}_f) (I_n - \hat{J}^+(q, \hat{\theta}_{ki}) \hat{J}(q, \hat{\theta}_{ki})) \psi - \beta R d(r) \Delta F \\ = & s_m - \beta R d(r) \Delta F.\end{aligned}\quad (4.34)$$

Next, using equations (4.7) and (4.10), one has

$$\hat{s}_x = \Delta \hat{x} + \alpha \Delta x + \beta (\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^{-1} R d(r) \Delta F. \quad (4.35)$$

Substitute equations (4.35) and (4.34) into equation (4.33) and using equation (4.27) yields

$$\begin{aligned}\dot{V} = & -(\Delta \hat{x} + \alpha \Delta x)^T K (\Delta \hat{x} + \alpha \Delta x) + \beta (\Delta \hat{x} + \alpha \Delta x)^T K (\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^{-1} R d(r) \Delta F \\ & + s_m^T R d(r) (\Delta \lambda + \gamma \Delta F) - \beta \Delta F (\Delta \lambda + \gamma \Delta F) + 2\alpha \Delta x^T K \Delta \dot{x} - \Delta \theta_{ki}^T L_{ki}^{-1} \dot{\hat{\theta}}_{ki} + \beta \Delta F \Delta \lambda \\ = & -(\Delta \hat{x} + \alpha \Delta x)^T K (\Delta \hat{x} + \alpha \Delta x) + 2\alpha \Delta x^T K \Delta \dot{x} - \Delta \theta_{ki}^T L_{ki}^{-1} \dot{\hat{\theta}}_{ki} + \beta \gamma \Delta F^2,\end{aligned}\quad (4.36)$$

where $R^T R = I$ and $d^T(r)d(r) = 1$.

From equations (4.5) and (4.6), since $\hat{x} = \dot{x} - Y_{ki}(q, \dot{q})\Delta\theta_{ki}$, one has

$$\Delta\hat{x} = \Delta\dot{x} - Y_{ki}(q, \dot{q})\Delta\theta_{ki}. \quad (4.37)$$

Substituting equations (4.26) and (4.37) into equation (4.36), gives

$$\begin{aligned} \dot{V} &= -(\Delta\dot{x} + \alpha\Delta x)^T K (\Delta\dot{x} + \alpha\Delta x) + 2(\Delta\dot{x} + \alpha\Delta x)^T K Y_{ki}(q, \dot{q})\Delta\theta_{ki} \\ &\quad - 2\Delta\theta_{ki}^T Y_{ki}^T(q, \dot{q}) K (\Delta\dot{x} + \alpha\Delta x) + 2\alpha\Delta x^T K \Delta\dot{x} - \Delta\theta_{ki}^T Y_{ki}^T(q, \dot{q}) K Y_{ki}(q, \dot{q})\Delta\theta_{ki} - \beta\gamma\Delta F^2. \\ &= -\Delta\dot{x}^T K \Delta\dot{x} - \alpha^2\Delta x^T K \Delta x - \Delta\theta_{ki}^T Y_{ki}^T(q, \dot{q}) K Y_{ki}(q, \dot{q})\Delta\theta_{ki} - \beta\gamma\Delta F^2 \leq 0. \end{aligned} \quad (4.38)$$

We are now in a position to state the following Theorem:

Theorem 4.1 *The vision and force tracking control law (4.23) and the update laws (4.24), (4.26) and (4.25) for the robot system (4.1) result in the convergence of vision and force tracking errors. That is, $x(t) - x_d(t) \rightarrow 0$, $\dot{x}(t) - \dot{x}_d(t) \rightarrow 0$ and $\lambda(t) - \lambda_d(t) \rightarrow 0$ as $t \rightarrow \infty$.*

Proof: Since $M(q)$ is positive definite, V is positive definite in s , Δx , $\Delta\theta_d$, $\Delta\theta_{ki}$, $\Delta\theta_f$ and ΔF . Since $\dot{V} \leq 0$, V is bounded. Therefore s , Δx , $\Delta\theta_d$, $\Delta\theta_{ki}$, $\Delta\theta_f$ and ΔF are bounded vectors. This implies that $\hat{\theta}_d$, $\hat{\theta}_{ki}$, $\hat{\theta}_f$ and x are bounded, and $\hat{s}_x = \hat{J}(q, \hat{\theta}_{ki})s$ is also bounded.

Next $\dot{x}_r$, $\hat{x}$ are bounded as seen from equations (4.7) and (4.10). From equation (4.12) one can conclude that $\dot{q}_r$ is bounded when $\hat{J}(q, \hat{\theta}_{ki})$ is nonsingular. Therefore $\dot{q}$ is bounded since s is bounded. The boundedness of $\dot{q}$ means that $\dot{x}$, $\dot{r}$ are bounded. Hence $\Delta\dot{x}$ is bounded and $\ddot{x}_r - \beta(\hat{J}_m(q, \hat{\theta}_f)\hat{J}^+(q, \hat{\theta}_{ki}))^{-1}Rd(r)\Delta\lambda$ from equation (4.8) is also bounded if $\ddot{x}_d$ is bounded. $\dot{d}(r, \dot{r})$ is bounded because r , $\dot{r}$ are bounded.

As seen from equation (4.26), $\dot{\hat{\theta}}_{ki}$ is bounded because $\dot{q}$, $\Delta \dot{x}$ and Δx are all bounded.

Hence, $\dot{J}(q, \hat{\theta}_{ki})$ is bounded. Using equation (4.13), one has

$$\begin{aligned} & \ddot{q}_r - \beta \hat{J}^+(q, \hat{\theta}_{ki})(\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^{-1} R d(r) \Delta \lambda \\ &= \hat{J}^+(q, \hat{\theta}_{ki})(\ddot{x}_r - \beta(\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^{-1} R d(r) \Delta \lambda) + \dot{\hat{J}}^+(q, \hat{\theta}_{ki}) \dot{x}_r \\ &+ (I_n - \hat{J}^+(q, \hat{\theta}_{ki}) \hat{J}(q, \hat{\theta}_{ki})) \dot{\psi} - (\dot{\hat{J}}^+(q, \hat{\theta}_{ki}) \hat{J}(q, \hat{\theta}_{ki}) + \hat{J}^+(q, \hat{\theta}_{ki}) \dot{\hat{J}}(q, \hat{\theta}_{ki})) \psi \end{aligned} \quad (4.39)$$

Since all the terms in the right hand side of the above equation are bounded,

$\ddot{q}_r - \beta \hat{J}^+(q, \hat{\theta}_{ki})(\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^{-1} R d(r) \Delta \lambda$ is therefore bounded. Next, from equation (4.15), one has

$$\begin{aligned} & \hat{q}_r - \beta \hat{J}^+(q, \hat{\theta}_{ki})(\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^{-1} R d(r) \Delta \lambda \\ &= \ddot{q}_r - \beta \hat{J}^+(q, \hat{\theta}_{ki})(\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^{-1} R d(r) \Delta \lambda \\ &- \beta \hat{J}^+(q, \hat{\theta}_{ki})(\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^{-1} R \dot{d}(r, \dot{r}) \Delta F, \end{aligned} \quad (4.40)$$

and hence $\hat{q}_r - \beta \hat{J}^+(q, \hat{\theta}_{ki})(\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^{-1} R d(r) \Delta \lambda$ is also bounded. Next, let

$$\begin{aligned} & \bar{Y}_d(q, \dot{q}, \dot{q}_r, \hat{q}_r, \hat{\theta}_f, \hat{\theta}_{ki}) \Delta \theta_d \\ &= (M(q) - \hat{M}(q)) \hat{q}_r + (\frac{1}{2}(\dot{M}(q) - \hat{M}(q)) + (S(q, \dot{q}) - \hat{S}(q, \dot{q}))) \dot{q}_r + g(q) - \hat{g}(q) \\ &+ \beta M(q) \hat{J}^+(q, \hat{\theta}_{ki})(\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^{-1} R (\dot{d}(h(q), J_m(q) \dot{q}) - \hat{d}(\hat{h}(q), \hat{J}_m(q, \hat{\theta}_f) \dot{q})) \Delta F \\ &= (M(q) - \hat{M}(q)) \hat{q}_r + \bar{Z}_d(q, \dot{q}, \dot{q}_r, \hat{\theta}_f, \hat{\theta}_{ki}) \Delta \theta_d, \end{aligned} \quad (4.41)$$

where $\bar{Z}_d(q, \dot{q}, \dot{q}_r, \hat{\theta}_f, \hat{\theta}_{ki}) \Delta \theta_d = (\frac{1}{2}(\dot{M}(q) - \hat{M}(q)) + (S(q, \dot{q}) - \hat{S}(q, \dot{q}))) \dot{q}_r + g(q) - \hat{g}(q) + \beta M(q) \hat{J}^+(q, \hat{\theta}_{ki})(\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^{-1} R (\dot{d}(h(q), J_m(q) \dot{q}) - \hat{d}(\hat{h}(q), \hat{J}_m(q, \hat{\theta}_f) \dot{q})) \Delta F$.

Then from equations (4.30) and (4.41), one has

$$\begin{aligned}
 & M(q)\ddot{q} - M(q)\ddot{q}_r + (M(q) - \hat{M}(q))\ddot{\hat{q}}_r + (\frac{1}{2}\dot{M}(q) + S(q, \dot{q}))s \\
 & + \bar{Z}_d(q, \dot{q}, \dot{q}_r, \hat{\theta}_f, \hat{\theta}_{ki})\Delta\theta_d + \hat{J}^T(q, \hat{\theta}_{ki})K(\Delta\dot{x} + \alpha\Delta x) \\
 & = (D^T(q) - \hat{D}^T(q, \hat{\theta}_f))\lambda + \hat{J}_m^T(q, \hat{\theta}_f)Rd(r)(\Delta\lambda + \gamma\Delta F)
 \end{aligned} \quad (4.42)$$

Since $D(q)\ddot{q} = -\dot{D}(q)\dot{q}$, one has

$$\begin{aligned}
 & -\dot{D}(q)\dot{q} + D(q)M^{-1}(q)\{-M(q)(\ddot{q}_r - \beta\hat{J}^+(q, \hat{\theta}_{ki})(\hat{J}_m(q, \hat{\theta}_f)\hat{J}^+(q, \hat{\theta}_{ki}))^{-1}Rd(r)\Delta\lambda) \\
 & + (M(q) - \hat{M}(q))(\ddot{\hat{q}}_r - \beta\hat{J}^+(q, \hat{\theta}_{ki})(\hat{J}_m(q, \hat{\theta}_f)\hat{J}^+(q, \hat{\theta}_{ki}))^{-1}Rd(r)\Delta\lambda) + r(t)\} \\
 & = D(q)M^{-1}(q)\{D_1^T\Delta\lambda - \hat{J}_m^T(q, \hat{\theta}_f)Rd(r)\lambda_d \\
 & + \beta\hat{M}(q)\hat{J}^+(q, \hat{\theta}_{ki})(\hat{J}_m(q, \hat{\theta}_f)\hat{J}^+(q, \hat{\theta}_{ki}))^{-1}Rd(r)\Delta\lambda + D_1^T\lambda_d\}
 \end{aligned} \quad (4.43)$$

where $r(t) = (\frac{1}{2}\dot{M}(q) + S(q, \dot{q}))s + \bar{Z}_d(q, \dot{q}, \dot{q}_r, x, \hat{\theta}_f, \hat{\theta}_{ki})\Delta\theta_d + \hat{J}^T(q, \hat{\theta}_{ki})K(\Delta\dot{x} + \alpha\Delta x) - \gamma\hat{J}_m^T(q, \hat{\theta}_f)Rd(r)\Delta F$ and $D_1^T = (D^T(q) - \hat{D}^T(q, \hat{\theta}_f) + \hat{J}_m^T(q, \hat{\theta}_f)Rd(r))$.

The above equation can be written as:

$$\bar{r}(t) = k(t)\Delta\lambda \quad (4.44)$$

where

$$\begin{aligned}
 \bar{r}(t) = & -\dot{D}(q)\dot{q} + D(q)M^{-1}(q)\{-M(q)(\ddot{q}_r - \beta\hat{J}^+(q, \hat{\theta}_{ki})(\hat{J}_m(q, \hat{\theta}_f)\hat{J}^+(q, \hat{\theta}_{ki}))^{-1}Rd(r)\Delta\lambda) \\
 & + (M(q) - \hat{M}(q))(\ddot{\hat{q}}_r - \beta\hat{J}^+(q, \hat{\theta}_{ki})(\hat{J}_m(q, \hat{\theta}_f)\hat{J}^+(q, \hat{\theta}_{ki}))^{-1}Rd(r)\Delta\lambda) \\
 & + r(t) - (D^T(q) - \hat{D}^T(q, \hat{\theta}_f))\lambda_d\}
 \end{aligned} \quad (4.45)$$

and

$$k(t) = D(q)M^{-1}(q)\{D_1^T + \beta\hat{M}(q)\hat{J}^+(q, \hat{\theta}_{ki})(\hat{J}_m(q, \hat{\theta}_f)\hat{J}^+(q, \hat{\theta}_{ki}))^{-1}Rd(r)\} \quad (4.46)$$

are bounded scalars. Hence $\Delta\lambda$ is bounded and the boundedness of $\Delta\lambda$ implies that $\ddot{x}_r$, $\hat{q}_r$ and $\ddot{q}_r$ are bounded.

From the closed-loop equation (4.30), one can conclude that $\dot{s}$ is bounded. The boundedness of $\dot{s}$ imply the boundedness of $\ddot{q}$ as seen from equation (4.17). Since $\ddot{x} = J(q)\ddot{q} + \dot{J}(q)\dot{q}$, $\ddot{x}$ is bounded and hence $\Delta\ddot{x}$ is also bounded.

To apply Barbalat's lemma, let us check the uniform continuity of $\dot{V}$. Differentiating equation (4.38) with respect to time gives,

$$\begin{aligned}\ddot{V} = & -2\Delta\dot{x}^T K \Delta\ddot{x} - 2\alpha^2 \Delta x^T K \Delta\dot{x} + 2\Delta\theta_{ki}^T Y_{ki}^T(q, \dot{q}) K Y_{ki}(q, \dot{q}) \dot{\theta}_{ki} \\ & - 2\Delta\theta_{ki}^T Y_{ki}^T(q, \dot{q}) K \dot{Y}_{ki}(q, \dot{q}) \Delta\theta_{ki} - 2\beta\gamma \Delta F \Delta\lambda.\end{aligned}\quad (4.47)$$

This shows that $\ddot{V}$ is bounded since Δx , $\Delta\dot{x}$, $\Delta\ddot{x}$, ΔF , $\Delta\lambda$ are all bounded. Hence, $\dot{V}$ is uniformly continuous. Using Barbalat's lemma, one has $\Delta x \rightarrow 0$, $\Delta\dot{x} \rightarrow 0$, $Y_{ki}(q, \dot{q}) \Delta\theta_{ki} = \dot{x} - \hat{x} \rightarrow 0$, $\Delta F \rightarrow 0$ as $t \rightarrow \infty$.

Next, the convergence of the force tracking error will be shown. From equations (4.24), (4.25) and (4.26), note that $\dot{\theta}_d$, $\dot{\theta}_f$, $\dot{\theta}_{ki}$ are bounded. Let $z_x = \ddot{x}_r - \beta \hat{J}_I R d(r) \Delta\lambda$ and $\hat{J}_I = (\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^{-1}$. From equation (4.8) one has

$$z_x = \ddot{x}_d - \alpha \Delta\dot{x} + \beta \hat{J}_I R d(r, \dot{r}) \Delta F + \beta \hat{J}_I \dot{R} d(r) \Delta F + \beta \dot{\hat{J}}_I R d(r) \Delta F \quad (4.48)$$

and hence $\dot{z}_x$ is also bounded. Hence, the derivatives of $\ddot{q}_r - \beta \hat{J}^+(q, \hat{\theta}_{ki})(\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^{-1} R d(r) \Delta\lambda$ and $\hat{q}_r - \beta \hat{J}^+(q, \hat{\theta}_{ki})(\hat{J}_m(q, \hat{\theta}_f) \hat{J}^+(q, \hat{\theta}_{ki}))^{-1} R d(r) \Delta\lambda$ are also bounded as seen from equations (4.39) and (4.40). From equation (4.43), the derivative λ is therefore bounded. Finally, using Barbalat's lemma and the convergence of ΔF to 0, one has $\Delta\lambda \rightarrow 0$ as $t \rightarrow \infty$. $\triangle\triangle\triangle$

Remark 4.1 The estimation of $d(r)$ can not be used in the proposed controller because it is difficult to obtain update laws for $\hat{J}_m(q, \hat{\theta}_f)$ and $\hat{d}(\hat{r})$ separately. From equation (2.19), one has

$$d(r) = \frac{(\frac{\partial \Psi(r)}{\partial r})^T}{\|\frac{\partial \Psi(r)}{\partial r}\|} = \frac{f}{\|f\|}, \quad (4.49)$$

where $\|f\| = \lambda$. Hence the direction and magnitude of the contact force f can be obtained from measurement of force sensor without any knowledge of the constraint function [2] [12] [44]. However, the differentiation of $d(r)$ leads to the differentiation of f which is possibly noisy. To overcome this problem, the unknown parameters of $\dot{d}(r, \dot{r})$ can be put into the dynamic parameter vector θ_d and updated by the parameters update law [44].

Remark 4.2 In this thesis, it is assumed that the robot is operating in a finite task space such that the approximate Jacobian matrix is of full rank. Note that the inverse Jacobian is used only in the definition of control variable $\dot{q}_r$. Therefore, one should be able to control this by bounding the variable or using a singularity-robust inverse of the approximate Jacobian matrix [43].

4.2 Adaptive Neural-network Vision and Force Tracking Control

In some applications, the uncertain parameters can not be linearly separated and the structure of the Jacobian matrices and constraint surface are also unknown. Hence, the adaptive vision and force controller presented in section 4.1 can not be used to compensate the uncertainties. In this section an adaptive neural-network vision and force tracking controller is developed to deal with the above mentioned uncertainties.

The manipulator Jacobian $J_m(q)$ is approximated by neural networks as

$$J_m^T(q) = (W_{m1}\theta_m(q), \dots, W_{mm}\theta_m(q)) + E_m, \quad (4.50)$$

where $W_{mi}(i = 1 \dots m)$ are matrices of neural network weights, $\theta_m(q)$ is a vector of activation functions and E_m is a bounded and small approximation error.

The Jacobian matrix $J(q)$ in equation (2.6) can be similarly approximated by neural networks as

$$J(q) = (W_{x1}\theta_x(q), \dots, W_{xn}\theta_x(q)) + E_x, \quad (4.51)$$

where $W_{xi}(i = 1 \dots n)$ are matrices of neural network weights, $\theta_x(q)$ is a vector of activation functions and E_x is a matrix of approximation errors that is bounded and small.

When $J_m(q)$ and $J(q)$ are uncertain, they are estimated as

$$\begin{aligned} \hat{J}_m^T(q, \hat{W}_m) &= (\hat{W}_{m1}\theta_m(q), \dots, \hat{W}_{mm}\theta_m(q)), \\ \hat{J}(q, \hat{W}_x) &= (\hat{W}_{x1}\theta_x(q), \dots, \hat{W}_{xn}\theta_x(q)), \end{aligned} \quad (4.52)$$

where $\hat{J}_m(q, \hat{W}_m)$ and $\hat{J}(q, \hat{W}_x)$ are estimations of $J_m(q)$ and $J(q)$ respectively and the estimated neural network weights $\hat{W}_m$ and $\hat{W}_x$ will be updated by update laws to be defined later.

Next, a vector $\dot{x}_r \in \mathbb{R}^m$ is defined as,

$$\dot{x}_r = (\dot{x}_d - \alpha \Delta x) + \beta(\hat{J}_m(q, \hat{W}_m)\hat{J}^+(q, \hat{W}_x))^{-1}R\hat{d}(r)\Delta F, \quad (4.53)$$

where α and β are positive constants, $x_d(t) \in \mathbb{R}^m$ is the desired image motion trajectory and $\dot{x}_d(t) \in \mathbb{R}^m$ is the desired speed trajectory, $\Delta x = x - x_d$ is image motion tracking error, $\hat{J}(q, \hat{W}_x)$ is an estimation of $J(q)$ and $\Delta F = \int_0^t (\lambda(\sigma) -$

$\lambda_d(\sigma))d\sigma$, $\lambda_d(t)$ is the desired force trajectory, R is a rotation matrix which will be defined later, $\hat{d}(r)$ is a fixed estimation of $d(r)$.

Next, the estimation error $J_m(q)(d(r) - \hat{d}(r))$ is approximated by neural networks as

$$J_m(q)(d(r) - \hat{d}(r)) = W_f \theta_f(q) + E_f \quad (4.54)$$

where W_f is a matrix of neural network weights, $\theta_f(q)$ is a vector of activation functions and E_f is a vector of approximation errors that is bounded and small.

Differentiating equation (4.53) with respect to time, one gets

$$\begin{aligned} \ddot{x}_r = & (\ddot{x}_d - \alpha \Delta \dot{x}) + \beta(\hat{J}_m(q, \hat{W}_m) \hat{J}^+(q, \hat{W}_x))^{-1} R \dot{d}(r) \Delta \lambda \\ & + \beta(\hat{J}_m(q, \hat{W}_m) \hat{J}^+(q, \hat{W}_x))^{-1} R \dot{d}(r, \dot{r}) \Delta F \\ & + \beta(\hat{J}_m(q, \hat{W}_m) \hat{J}^+(q, \hat{W}_x))^{-1} \dot{R} \hat{d}(r) \Delta F + \beta \dot{\hat{J}}_I R \hat{d}(r) \Delta F, \end{aligned} \quad (4.55)$$

where $\dot{\hat{J}}_I = \frac{d}{dt}((\hat{J}_m(q, \hat{W}_m) \hat{J}^+(q, \hat{W}_x))^{-1})$.

In order to prove the stability of the vision-force tracking system, an adaptive sliding vector is defined using equation (4.53) as,

$$\hat{s}_x = \hat{\dot{x}} - \dot{x}_r = \hat{J}(q, \hat{W}_x) \dot{q} - \dot{x}_r, \quad (4.56)$$

Differentiating the above equation with respect to time, one has,

$$\dot{\hat{s}}_x = \hat{\ddot{x}} - \ddot{x}_r = \hat{J}(q, \hat{W}_x) \ddot{q} + \dot{\hat{J}}(q, \hat{W}_x) \dot{q} - \ddot{x}_r, \quad (4.57)$$

Next, let

$$\dot{q}_r = \hat{J}^+(q, \hat{W}_x) \dot{x}_r + (I_n - \hat{J}^+(q, \hat{W}_x) \hat{J}(q, \hat{W}_x)) \psi, \quad (4.58)$$

where $\psi \in \mathbb{R}^n$ is minus the gradient of the convex function to be optimized [48].

From equation (4.58), one has,

$$\begin{aligned}\ddot{q}_r = & \hat{J}^+(q, \hat{W}_x)\ddot{x}_r + \dot{\hat{J}}^+(q, \hat{W}_x)\dot{x}_r + (I_n - \hat{J}^+(q, \hat{W}_x)\hat{J}(q, \hat{W}_x))\dot{\psi} \\ & - (\dot{\hat{J}}^+(q, \hat{W}_x)\hat{J}(q, \hat{W}_x) + \hat{J}^+(q, \hat{W}_x)\dot{\hat{J}}(q, \hat{W}_x))\psi,\end{aligned}\quad (4.59)$$

Next, an adaptive sliding vector is defined in joint space as,

$$s = \dot{q} - \dot{q}_r, \quad (4.60)$$

and

$$\dot{s} = \ddot{q} - \ddot{q}_r. \quad (4.61)$$

Multiplying both side of equation (4.60) by $\hat{J}(q, \hat{W}_x)$ and using equation (4.56), one has

$$\hat{J}(q, \hat{W}_x)s = \hat{J}(q, \hat{W}_x)\dot{q} - \dot{x}_r = \hat{s}_x, \quad (4.62)$$

Substitute equations (4.60) and (4.61) into equation (4.1) to get,

$$\begin{aligned}M(q)\dot{s} + (\tfrac{1}{2}\dot{M}(q) + S(q, \dot{q}))s + M(q)\ddot{q}_r + (\tfrac{1}{2}\dot{M}(q) + S(q, \dot{q}))\dot{q}_r + g(q) \\ = \tau + D^T(q)\lambda,\end{aligned}\quad (4.63)$$

The last three terms on the left hand side of equation (4.63) can be expressed as

$$M(q)\ddot{q}_r + (\tfrac{1}{2}\dot{M}(q) + S(q, \dot{q}))\dot{q}_r + g(q) = W_d \theta_d(q, \dot{q}, \ddot{q}_r, \hat{W}_x, \hat{W}_m) + E_d. \quad (4.64)$$

where W_d is a matrix of neural network weights, $\theta_d(q, \dot{q}, \ddot{q}_r, \hat{W}_x, \hat{W}_m)$ is a vector of activation functions and E_d is a vector of approximation errors that is bounded and

small. Then the dynamics equation (4.63) can be expressed as:

$$M(q)\dot{s} + (\frac{1}{2}\dot{M}(q) + S(q, \dot{q}))s + W_d \theta_d(q, \dot{q}, \ddot{q}_r, \hat{W}_x, \hat{W}_m) + E_d = \tau + D^T(q)\lambda \quad (4.65)$$

The vision and force tracking controller is proposed as:

$$\begin{aligned} \tau = & -\hat{J}^T(q, \hat{W}_x)K_p\Delta x - K_v s + \hat{W}_d \theta_d(q, \dot{q}, \ddot{q}_r, \hat{W}_x, \hat{W}_m) - \hat{J}_m^T(q, \hat{W}_m)\hat{d}(r)\lambda \\ & - \hat{W}_f \theta_f(q)\lambda + \hat{J}_m^T(q, \hat{W}_m)R\hat{d}(r)(\kappa\Delta\lambda + \gamma\Delta F) - K_m\hat{d}(r)\lambda - K_f\lambda, \end{aligned} \quad (4.66)$$

where $\Delta\hat{x} = \hat{x} - \hat{x}_d$, κ and γ are positive constants, K_p and K_v are positive diagonal gain matrices, K_m is a matrix designed to compensate the estimation error of $\hat{J}_m^T(q, \hat{W}_m)$, $k_{mij} = -\bar{k}_{mij}\text{sgn}(s_i\hat{d}_j(r))$, k_{mij} is the element in the i^{th} row and j^{th} column of K_m , K_f is a vector to compensate the estimation error of $J_m(q)(d(r) - \hat{d}(r))$ and $K_{fi} = k_{fi}\text{sgn}(s_i)$. The estimated parameters $\hat{W}_d$, $\hat{W}_m$ and $\hat{W}_x$ are updated by,

$$\dot{\hat{W}}_{di}^T = \text{proj}(\Omega_{di}), \quad (4.67)$$

$$\dot{\hat{W}}_{mij}^T = \text{proj}(\Omega_{mij}), \quad (4.68)$$

$$\dot{\hat{W}}_{xij}^T = \text{proj}(\Omega_{xij}), \quad (4.69)$$

$$\dot{\hat{W}}_{fi}^T = \text{proj}(\Omega_{fi}), \quad (4.70)$$

where

$$\begin{aligned} \Omega_{di} &= -k_1\hat{W}_{di}^T - L_{di}s_i\theta_d(q, \dot{q}, \ddot{q}_r, \hat{W}_x, \hat{W}_m), \\ \Omega_{mij} &= -k_2\hat{W}_{mij}^T + L_{mij}s_i\hat{d}_j(r)\lambda\theta_m(q), \\ \Omega_{xij} &= -k_3\hat{W}_{xij}^T + L_{xij}\Delta x_i k_{pi}\dot{q}_j\theta_x(q) \\ \Omega_{fi} &= -k_4\hat{W}_{fi}^T + L_{fi}s_i\theta_f(q), \end{aligned} \quad (4.71)$$

where $\hat{W}_{di}$, $\hat{W}_{mij}$, $\hat{W}_{xij}$ and $\hat{W}_{fi}$ are the i^{th} row vectors of $\hat{W}_d$, $\hat{W}_m$, $\hat{W}_x$ and $\hat{W}_f$,

s_i , $\hat{d}_i(r)$, $\dot{q}_i$ are the i^{th} elements of s , $\hat{d}(r)$ and $\dot{q}$, Δx_j is the j^{th} element of Δx and k_{pj} is the j^{th} element of the diagonal matrix K_p , $L_{di} = l_{di}I$, $L_{mij} = l_{mij}I$, $L_{xij} = l_{xij}I$, $L_{fi} = l_{fi}I$ are positive gain matrices, k_1 , k_2 , k_3 and k_4 are positive constants and the function $proj(\Omega_d)$ is a projection algorithm defined as [59]

$$proj(\Omega_{di}) = \begin{cases} \Omega_{di} & \text{if } \hat{W}_{di} > \underline{W}_{di} \\ \Omega_{di} & \text{if } \hat{W}_{di} = \underline{W}_{di} \text{ and } \Omega_{di} \geq 0 \\ 0 & \text{if } \hat{W}_{di} = \underline{W}_{di} \text{ and } \Omega_{di} < 0 \\ 0 & \text{if } \hat{W}_{di} = \bar{W}_{di} \text{ and } \Omega_{di} > 0 \\ \Omega_{di} & \text{if } \hat{W}_{di} = \bar{W}_{di} \text{ and } \Omega_{di} \leq 0 \\ \Omega_{di} & \text{if } \hat{W}_{di} < \bar{W}_{di} \end{cases} \quad (4.72)$$

where $\underline{W}_{di}$ and $\bar{W}_{di}$ are the lower and upper bounds of W_{di} . The projection algorithms $proj(\Omega_m)$ and $proj(\Omega_d)$ can be similarly defined as above. The functions $proj(\cdot)$ are defined to ensure that $\hat{J}^T(q, \hat{W}_x)$ and $\hat{J}_m(q, \hat{W}_m)$ are bounded during adaption.

In the above controller, R is a rotation matrix designed [80] so that

$$s_{xN}^T R \hat{d}(r) = 0, \quad (4.73)$$

where

$$\begin{aligned} s_{xN} &= \{\beta \Delta x^T K_p (\hat{J}_m(q, \hat{W}_m) \hat{J}^+(q, \hat{W}_x))^{-1} \Delta F + s_m^T (\kappa \Delta \lambda + \gamma \Delta F)\}^T \\ s_m &= \hat{J}_m(q, \hat{W}_m) \{\dot{q} - \hat{J}^+(q, \hat{W}_x) (\dot{x}_d - \alpha \Delta x) - (I_n - \hat{J}^+(q, \hat{W}_x) \hat{J}(q, \hat{W}_x)) \psi\} \end{aligned} \quad (4.74)$$

Substituting equation (4.66) into equation (4.65), the closed-loop equation is ob-

tained as

$$\begin{aligned}
& M(q)\dot{s} + (\tfrac{1}{2}\dot{M}(q) + S(q, \dot{q}))s + \hat{J}^T(q, \hat{W}_x)K_p\Delta x + K_v s \\
& \quad + \Delta W_d \theta_d(q, \dot{q}, \ddot{q}_r, \hat{W}_x, \hat{W}_m) + E_d \\
& = \Delta W_m \theta_m(q) \hat{d}(r)\lambda + E_m \hat{d}(r)\lambda + \Delta W_f \theta_f(q)\lambda + E_f \lambda \\
& \quad + \hat{J}_m^T(q, \hat{W}_m) R \hat{d}(r)(\kappa \Delta \lambda + \gamma \Delta F) - K_m \hat{d}(r)\lambda - K_f \lambda,
\end{aligned} \tag{4.75}$$

where $\Delta W_d = W_d - \hat{W}_d$, $\Delta W_m = W_m - \hat{W}_m$ and $\Delta W_f = W_f - \hat{W}_f$.

To carry out the stability analysis, the Lyapunov-like function candidate V is defined as:

$$\begin{aligned}
V = & \tfrac{1}{2}s^T M(q)s + \tfrac{1}{2}\Delta x^T K_p \Delta x + \tfrac{1}{2}\sum_{i=1}^n \Delta W_{di} L_{di}^{-1} \Delta W_{di}^T + \tfrac{1}{2}\sum_{i=1}^n \sum_{j=1}^m \Delta W_{mij} L_{mij}^{-1} \Delta W_{mij}^T \\
& + \tfrac{1}{2}\sum_{i=1}^n \Delta W_{fi} L_{fi}^{-1} \Delta W_{fi}^T + \tfrac{1}{2}\sum_{i=1}^m \sum_{j=1}^n \Delta W_{xij} L_{xij}^{-1} \Delta W_{xij}^T + \tfrac{1}{2}\beta \kappa \Delta F^2.
\end{aligned} \tag{4.76}$$

where $\Delta W_{xij} = W_{xij} - \hat{W}_{xij}$. Differentiating V with respect of time yields,

$$\begin{aligned}
\dot{V} = & s^T M(q)\dot{s} + \tfrac{1}{2}s^T \dot{M}(q)s + \Delta x^T K_p \Delta \dot{x} - \sum_{i=1}^n \Delta W_{di}^T L_{di}^{-1} \dot{\hat{W}}_{di} \\
& - \sum_{i=1}^n \Delta W_{fi}^T L_{fi}^{-1} \dot{\hat{W}}_{fi} - \sum_{i=1}^n \sum_{j=1}^m \Delta W_{mij} L_{mij}^{-1} \dot{\hat{W}}_{mij}^T \\
& - \sum_{i=1}^m \sum_{j=1}^n \Delta W_{xij} L_{xij}^{-1} \dot{\hat{W}}_{xij}^T + \beta \kappa \Delta F \Delta \lambda.
\end{aligned} \tag{4.77}$$

Substituting equation (4.75) into the above equation and using equation (4.56), one

has,

$$\begin{aligned}
\dot{V} = & -\hat{s}_x^T K_p \Delta x - s^T K_v s + s^T \hat{J}_m^T(q, \hat{\theta}_f) R \hat{d}(r) (\kappa \Delta \lambda + \gamma \Delta F) \\
& -s^T \Delta W_d \theta_d(q, \dot{q}, \ddot{q}_r, \hat{W}_x, \hat{W}_m) + \Delta x^T K_p \Delta \dot{x} + \beta \kappa \Delta F \Delta \lambda \\
& -s^T E_d + s^T E_m \hat{d}(r) \lambda - s^T K_m \hat{d}(r) \lambda + s^T \Delta W_f \theta_f(q) \lambda + s^T E_f \lambda \\
& + s^T \Delta W_m \theta_m(q) \hat{d}(r) \lambda - \sum_{i=1}^m \sum_{j=1}^n \Delta W_{xij} L_{xij}^{-1} \dot{W}_{xij}^T - \sum_{i=1}^n \Delta W_{fi} L_{fi}^{-1} \dot{W}_{fi}^T \\
& - \sum_{i=1}^n \Delta W_{di} L_{di}^{-1} \dot{W}_{di}^T - \sum_{i=1}^n \sum_{j=1}^m \Delta W_{mij} L_{mij}^{-1} \dot{W}_{mij}^T. \tag{4.78}
\end{aligned}$$

From equations (4.60) and (4.53), note that

$$\begin{aligned}
& \hat{J}_m(q, \hat{W}_m) s = \\
& \hat{J}_m(q, \hat{W}_m) \dot{q} - \hat{J}_m(q, \hat{W}_m) \hat{J}^+(q, \hat{W}_x) (\dot{x}_d - \alpha \Delta x) \\
& - \hat{J}_m(q, \hat{W}_m) (I_n - \hat{J}^+(q, \hat{W}_x) \hat{J}(q, \hat{W}_x)) \psi - \beta R \hat{d}(r) \Delta F \\
& = s_m - \beta R \hat{d}(r) \Delta F. \tag{4.79}
\end{aligned}$$

Next, substitute equations (4.56), (4.53) and (4.79) into equation (4.78) and using equation (4.73), one has

$$\begin{aligned}
\dot{V} = & -\alpha \Delta x^T K_p \Delta x - s^T K_v s - \Delta x^T K_p \Delta \dot{x} + \Delta x^T K_p \Delta \dot{x} \\
& -\beta \gamma \Delta F^2 - s^T E_d - s^T (K_m - E_m) \hat{d}(r) \lambda - s^T (K_f - E_f) \lambda \\
& -s^T \Delta W_d \theta_d(q, \dot{q}, \ddot{q}_r, \hat{W}_x, \hat{W}_m) + s^T \Delta W_m \theta_m(q) \hat{d}(r) \lambda + s^T \Delta W_f \theta_f(q) \lambda \\
& - \sum_{i=1}^m \sum_{j=1}^n \Delta W_{xij} L_{xij}^{-1} \dot{W}_{xij}^T - \sum_{i=1}^n \Delta W_{di} L_{di}^{-1} \dot{W}_{di}^T \\
& - \sum_{i=1}^n \sum_{j=1}^m \Delta W_{mij} L_{mij}^{-1} \dot{W}_{mij}^T - \sum_{i=1}^n \Delta W_{fi} L_{fi}^{-1} \dot{W}_{fi}^T, \tag{4.80}
\end{aligned}$$

where $R^T R = I$ and $\hat{d}^T(r) \hat{d}(r) = 1$. From equations (4.51) and (4.52), since $\dot{x} =$

$\hat{x} + \Delta W_x \theta_x(q) \dot{q} + E_x \dot{q}$, one has

$$\Delta \dot{x} = \Delta \hat{x} + \Delta W_x \theta_x(q) \dot{q} + E_x \dot{q}. \quad (4.81)$$

Substituting equations (4.69) and (4.81) into equation (4.80), using (4.67)-(4.71), gives

$$\begin{aligned} \dot{V} = & -\alpha \Delta x^T K_p \Delta x - s^T K_v s + \Delta x^T K_p E_x \dot{q} - \beta \gamma \Delta F^2 \\ & - s^T E_d - s^T (K_m - E_m) \hat{d}(r) \lambda + k_3 \sum_{i=1}^m \sum_{j=1}^n \Delta W_{xij} L_{xij}^{-1} \hat{W}_{xij}^T \\ & + k_1 \sum_{i=1}^n \Delta W_{di} L_{di}^{-1} \hat{W}_{di}^T + k_2 \sum_{i=1}^n \sum_{j=1}^m \Delta W_{mij} L_{mij}^{-1} \hat{W}_{mij}^T + k_4 \sum_{i=1}^n \Delta W_{fi} L_{fi}^{-1} \hat{W}_{fi}^T \\ & + \sum_{i=1}^n \Delta W_{di} L_{di}^{-1} (\Omega_{di} - \dot{W}_{di}^T) + \sum_{i=1}^n \sum_{j=1}^m \Delta W_{mij} L_{mij}^{-1} (\Omega_{mij} - \dot{W}_{mij}^T) \\ & + \sum_{i=1}^m \sum_{j=1}^n \Delta W_{xij} L_{xij}^{-1} (\Omega_{xij} - \dot{W}_{xij}^T) + \sum_{i=1}^n \Delta W_{fi} L_{fi}^{-1} (\Omega_{fi} - \dot{W}_{fi}^T). \end{aligned} \quad (4.82)$$

Then using equation (4.72), one has

$$\begin{aligned} \dot{V} \leq & -\alpha \Delta x^T K_p \Delta x - s^T K_v s + \Delta x^T K_p E_x \dot{q} - \beta \gamma \Delta F^2 \\ & - s^T E_d - s^T (K_m - E_m) \hat{d}(r) \lambda \\ & + k_3 \sum_{i=1}^m \sum_{j=1}^n \Delta W_{xij} L_{xij}^{-1} \hat{W}_{xij}^T + k_1 \sum_{i=1}^n \Delta W_{di} L_{di}^{-1} \hat{W}_{di}^T \\ & + k_2 \sum_{i=1}^n \sum_{j=1}^m \Delta W_{mij} L_{mij}^{-1} \hat{W}_{mij}^T + k_4 \sum_{i=1}^n \Delta W_{fi} L_{fi}^{-1} \hat{W}_{fi}^T. \end{aligned} \quad (4.83)$$

From equations (4.59) and (4.60), one has

$$\begin{aligned} \dot{q} = & s + \dot{q}_r = s + \hat{J}^+(q, \hat{W}_x) \dot{x}_r + (I_n - \hat{J}^+(q, \hat{W}_x) \hat{J}(q, \hat{W}_x)) \psi \\ & = s + \hat{J}^+(q, \hat{W}_x) (\dot{x}_d - \alpha \Delta x) \\ & + \beta \hat{J}^+(q, \hat{W}_x) (\hat{J}_m(q, \hat{W}_m) \hat{J}^+(q, \hat{W}_x))^{-1} R \hat{d}(r) \Delta F \\ & + (I_n - \hat{J}^+(q, \hat{W}_x) \hat{J}(q, \hat{W}_x)) \psi. \end{aligned} \quad (4.84)$$

Substituting equation (4.84) into equation (4.83), one has

$$\begin{aligned}
\dot{V} \leq & -\alpha \Delta x^T K_p \Delta x - s^T K_v s - \beta \gamma \Delta F^2 - s^T E_d - s^T (K_m - E_m) \hat{d}(r) \lambda \\
& + k_3 \sum_{i=1}^m \sum_{j=1}^n \Delta W_{xij} L_{xij}^{-1} \hat{W}_{xij}^T + k_1 \sum_{i=1}^n \Delta W_{di} L_{di}^{-1} \hat{W}_{di}^T + k_2 \sum_{i=1}^n \sum_{j=1}^m \Delta W_{mij} L_{mij}^{-1} \hat{W}_{mij}^T \\
& + k_4 \sum_{i=1}^n \Delta W_{fi} L_{fi}^{-1} \hat{W}_{fi}^T + \Delta x^T K_p E_x s + \Delta x^T K_p E_x \hat{J}^+(q, \hat{W}_x) \dot{x}_d - \alpha \Delta x^T K_p E_x \hat{J}^+(q, \hat{W}_x) \Delta x \\
& + \beta \Delta x^T K_p E_x \hat{J}^+(q, \hat{W}_x) (\hat{J}_m(q, \hat{W}_m) \hat{J}^+(q, \hat{W}_x))^{-1} R \hat{d}(r) \Delta F \\
& + \Delta x^T K_p E_x \hat{J}^+(q, \hat{W}_x) (I_n - \hat{J}^+(q, \hat{W}_x) \hat{J}(q, \hat{W}_x)) \psi.
\end{aligned} \tag{4.85}$$

Next, note that

$$\begin{aligned}
s^T E_d & \leq \frac{1}{2} (\|s\|^2 + \|E_d\|^2) \\
\Delta x^T K_p E_x s & \leq \frac{b_{ex} b_p}{2} (\|\Delta x\|^2 + \|s\|^2) \\
\Delta x^T K_p E_x \hat{J}^+(q, \hat{W}_x) \dot{x}_d & \leq \frac{b_p b_{ex} b_1}{2} (\|\Delta x\|^2 + \|\dot{x}_d\|^2) \\
\alpha \Delta x^T K_p E_x \hat{J}^+(q, \hat{W}_x) \Delta x & \leq \alpha b_p b_{ex} b_1 \|\Delta x\|^2 \\
\beta \Delta x^T K_p E_x \hat{J}^+(q, \hat{W}_x) (\hat{J}_m(q, \hat{W}_m) \hat{J}^+(q, \hat{W}_x))^{-1} R \hat{d}(r) \Delta F & \leq \frac{\beta b_p b_{ex} b_1 b_2}{2} (\|\Delta x\|^2 + \Delta F^2) \\
\Delta x^T K_p E_x \hat{J}^+(q, \hat{W}_x) (I_n - \hat{J}^+(q, \hat{W}_x) \hat{J}(q, \hat{W}_x)) \psi & \leq \frac{b_p b_{ex} b_1 b_3}{2} (\|\Delta x\|^2 + \|\psi\|^2).
\end{aligned} \tag{4.86}$$

where $b_{ex}, b_p, b_1, b_2, b_3$ are upper bounds of $E_x, K_p, \hat{J}^+(q, \hat{W}_x), (\hat{J}_m(q, \hat{W}_m) \hat{J}^+(q, \hat{W}_x))^{-1}$ and $I_n - \hat{J}^+(q, \hat{W}_x) \hat{J}(q, \hat{W}_x)$. In addition,

$$\begin{aligned}
k_1 \Delta W_{di} L_{di}^{-1} \hat{W}_{di}^T & \leq \frac{k_1}{2l_{di}} (\|W_{di}\|^2 - \|\Delta W_{di}\|^2) \\
k_2 \Delta W_{mij} L_{mij}^{-1} \hat{W}_{mij}^T & \leq \frac{k_2}{2l_{mij}} (\|W_{fij}\|^2 - \|\Delta W_{fij}\|^2), \\
k_3 \Delta W_{xij} L_{xij}^{-1} \hat{W}_{xij}^T & \leq \frac{k_3}{2l_{xij}} (\|W_{xij}\|^2 - \|\Delta W_{xij}\|^2), \\
k_4 \Delta W_{fi} L_{fi}^{-1} \hat{W}_{fi}^T & \leq \frac{k_4}{2l_{fi}} (\|W_{fi}\|^2 - \|\Delta W_{fi}\|^2)
\end{aligned} \tag{4.87}$$

Using inequalities (4.86) and (4.87), then $\dot{V}$ becomes

$$\begin{aligned}\dot{V} \leq & -(\alpha k_{pmin} - \bar{b})\|\Delta x\|^2 - (k_{vmin} - \frac{b_{ex}b_p+1}{2})\|s\|^2 - s^T(K_m - E_m)\hat{d}(r)\lambda \\ & -\beta(\gamma - \frac{b_p b_{ex} b_1 b_2}{2})\Delta F^2 + \mu - \frac{k_1}{2l_{d1}}\sum_{i=1}^n \|\Delta W_{di}\|^2 \\ & - \frac{k_3}{2l_{xij}}\sum_{i=1}^m \sum_{j=1}^n \|\Delta W_{xij}\|^2 - \frac{k_2}{2l_{mij}}\sum_{i=1}^n \sum_{j=1}^m \|\Delta W_{mij}\|^2 - \frac{k_4}{2l_{fi}}\sum_{i=1}^n \|\Delta W_{fi}\|^2, \quad (4.88)\end{aligned}$$

where

$$\begin{aligned}\bar{b} = & \frac{b_{ex}b_p}{2} + \frac{b_p b_{ex} b_1}{2} + \alpha b_p b_{ex} b_1 + \frac{\beta b_p b_{ex} b_1 b_2}{2} + \frac{b_p b_{ex} b_1 b_3}{2} \\ \mu = & \frac{1}{2}\|E_d\|^2 + \frac{b_p b_{ex} b_2}{2}b_d^2 + \frac{b_p b_{ex} b_1 b_3}{2}b_\psi^2 + \frac{k_1}{2l_{d1}}\sum_{i=1}^n \|W_{di}\|^2 \\ & + \frac{k_3}{2l_{xij}}\sum_{i=1}^m \sum_{j=1}^n \|W_{xij}\|^2 + \frac{k_2}{2l_{mij}}\sum_{i=1}^n \sum_{j=1}^m \|W_{mij}\|^2 + \frac{k_4}{2l_{fi}}\sum_{i=1}^n \|W_{fi}\|^2, \quad (4.89)\end{aligned}$$

where b_d and b_ψ are upper bounds of $\dot{x}_d$ and ψ . When $|\bar{k}_{mij}|$ is set sufficiently large so that $|\bar{k}_{mij}| \geq |E_{mij}|$, one has

$$\begin{aligned}\dot{V} \leq & -(\alpha k_{pmin} - \bar{b})\|\Delta x\|^2 - (k_{vmin} - \frac{b_{ex}b_p+1}{2})\|s\|^2 - \beta(\gamma - \frac{b_p b_{ex} b_1 b_2}{2})\Delta F^2 + \mu \\ & - \frac{k_1}{2l_{d1}}\sum_{i=1}^n \|\Delta W_{di}\|^2 - \frac{k_3}{2l_{xij}}\sum_{i=1}^m \sum_{j=1}^n \|\Delta W_{xij}\|^2 - \frac{k_2}{2l_{mij}}\sum_{i=1}^n \sum_{j=1}^m \|\Delta W_{mij}\|^2 - \frac{k_4}{2l_{fi}}\sum_{i=1}^n \|\Delta W_{fi}\|^2.\end{aligned}\quad (4.90)$$

Let $k_{pmin}, k_{vmin}, \alpha, \beta, \gamma$ be chosen sufficiently large so that

$$\begin{aligned}k_{vmin} - \frac{b_{ex}b_p+1}{2} & > 0 \\ \alpha k_{pmin} - \bar{b} & > 0 \\ \gamma - \frac{b_p b_{ex} b_1 b_2}{2} & > 0.\end{aligned}\quad (4.91)$$

There exist positive constants $\bar{\gamma}_1, \bar{\gamma}_2, \bar{\gamma}_3$ such that

$$\begin{aligned} (k_{vmin} - \frac{b_{ex}b_p+1}{2})\|s\|^2 &\geq \frac{\bar{\gamma}_1}{2}s^T M(q)s \\ (\alpha k_{pmin} - \bar{b})\|\Delta x\|^2 &\geq \frac{\bar{\gamma}_2}{2}\Delta x^T K_p \Delta x \\ \beta(\gamma - \frac{b_p b_{ex} b_1 b_2}{2})\Delta F^2 &\geq \frac{\bar{\gamma}_3}{2}\beta \kappa \Delta F^2, \end{aligned} \quad (4.92)$$

Let $\bar{\gamma} = \min\{\bar{\gamma}_1, \bar{\gamma}_2, \bar{\gamma}_3, k_1, k_2, k_3, k_4\}$, one has

$$\dot{V} \leq -\bar{\gamma}V + \mu. \quad (4.93)$$

The above inequality implies

$$V \leq \frac{\mu}{\bar{\gamma}} + \{V(0) - \frac{\mu}{\bar{\gamma}}\}e^{-\bar{\gamma}t}. \quad (4.94)$$

Then it can be concluded that the system is uniformly ultimate bounded.

Theorem *The adaptive Jacobian control law (4.66) and the update laws (4.67), (4.69), (4.68) and (4.70) for the robot system (4.1) result in the uniformly ultimate boundedness of vision and force tracking errors when K_p, K_v, α and γ are chosen to satisfy condition (4.92). Moreover, the errors can be made arbitrarily small by adjusting the control gains.*

Proof: From inequality (4.94), it can be concluded that $s, \Delta x, \Delta W_d, \Delta W_m, \Delta W_x, \Delta W_f$ and ΔF are uniformly ultimate bounded. This implies that $\hat{W}_d, \hat{W}_m, \hat{W}_x, \hat{W}_f$ and x are bounded, and $\hat{s}_x = \hat{J}(q, \hat{W}_x)s$ is also bounded. Next $\dot{x}_r, \dot{\hat{x}}$ are bounded as seen from equations (4.53) and (4.56). From equation (4.58) one can conclude that $\dot{q}_r$ is bounded when $\hat{J}(q, \hat{W}_x)$ is nonsingular. Therefore $\dot{q}$ is bounded since s is bounded. The boundedness of $\dot{q}$ means that $\dot{x}, \dot{r}$ are bounded. Hence $\Delta \dot{x}$ is bounded and $\dot{d}(r, \dot{r})$ is also bounded because $r, \dot{r}$ are bounded.

Then from equations (4.75) and (4.64), one has

$$\begin{aligned}
 & M(q)\ddot{q} + \left(\frac{1}{2}\dot{M}(q) + S(q, \dot{q})\right)\dot{q}_r + g(q) + \hat{J}^T(q, \hat{W}_x)K_p\Delta x + K_v s \\
 & = \Delta W_m \theta_m(q) \hat{d}(r) \lambda + E_m \hat{d}(r) \lambda + \Delta W_f \theta_f(q) \lambda + E_f \lambda \\
 & + \hat{J}_m^T(q, \hat{W}_m) R \hat{d}(r) (\kappa \Delta \lambda + \gamma \Delta F) - K_m \hat{d}(r) \lambda - K_f \lambda.
 \end{aligned} \tag{4.95}$$

Since $D(q)\ddot{q} = -\dot{D}(q)\dot{q}$, one has

$$\begin{aligned}
 & -\dot{D}(q)\dot{q} + D(q)M^{-1}(q)r_1(t) \\
 & = D(q)M^{-1}(q)\{(\Delta W_m \theta_m(q) + E_m - K_m)\hat{d}(r)\Delta \lambda \\
 & + (\Delta W_f \theta_f(q) + E_f - K_f)\Delta \lambda + \kappa \hat{J}_m^T(q, \hat{W}_m) R \hat{d}(r)\Delta \lambda\}
 \end{aligned} \tag{4.96}$$

where $r_1(t) = (\frac{1}{2}\dot{M}(q) + S(q, \dot{q}))\dot{q}_r + g(q) + \hat{J}^T(q, \hat{W}_x)K_p\Delta x + K_v s - (\Delta W_m \theta_m(q) + E_m - K_m)\hat{d}(r)\lambda_d - (\Delta W_f \theta_f(q) + E_f - K_f)\lambda_d - \gamma \hat{J}_m^T(q, \hat{W}_m) R \hat{d}(r)\Delta F$.

The above equation can be written as:

$$\bar{r}_1(t) = k(t)\Delta \lambda \tag{4.97}$$

where

$$\bar{r}(t) = -\dot{D}(q)\dot{q} + D(q)M^{-1}(q)r_1(t) \tag{4.98}$$

and

$$\begin{aligned}
 k(t) = & D(q)M^{-1}(q)\{\Delta W_m \theta_m(q)\hat{d}(r) + E_m \hat{d}(r) + \Delta W_f \theta_f(q) + E_f \\
 & - K_m \hat{d}(r) - K_f + \kappa \hat{J}_m^T(q, \hat{W}_m) R \hat{d}(r)\}
 \end{aligned} \tag{4.99}$$

are bounded scalars. Hence the force tracking error $\Delta \lambda$ is also bounded.

4.3 Simulation Results

In this section, simulation results are presented to illustrate the performance of the proposed controller. Consider a two-link manipulator whose end-effector is required to move on a constraint surface as in section 3.5. A fixed camera is placed distance away from the robot.

4.3.1 Adaptive Vision-force Tracking Control

The Jacobian matrix $J(q)$ mapping from joint space to image space is given by

$$J(q) = J_I(r)J_m(q) = \frac{f_1}{z_1 - f_1} \begin{bmatrix} \beta_1 & 0 \\ 0 & \beta_2 \end{bmatrix} \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{bmatrix}, \quad (4.100)$$

where $s_1 = \sin(q_1)$, $c_1 = \cos(q_1)$, $s_{12} = \sin(q_1 + q_2)$, $c_{12} = \cos(q_1 + q_2)$ and l_1, l_2 are the lengths of the first and second link, respectively. β_1, β_2 denote the scaling factors in pixels/m, z_1 is the perpendicular distance between the robot and the camera, f_c is the focal length of the camera.

Hence $\dot{x}$ can be written as the product of a known regressor matrix $Y_{ki}(q, \dot{q})$ and an unknown vector θ_{ki} where

$$Y_{ki}(q, \dot{q}) = \begin{bmatrix} -s_1 \dot{q}_1 & -s_{12}(\dot{q}_1 + \dot{q}_2) & 0 & 0 \\ 0 & 0 & c_1 \dot{q}_1 & c_{12}(\dot{q}_1 + \dot{q}_2) \end{bmatrix}$$

$$\theta_{ki} = \left[\frac{f_c}{z_1 - f_c} \beta_1 l_1, \frac{f_c}{z_1 - f_c} \beta_1 l_2, \frac{f_c}{z_1 - f_c} \beta_2 l_1, \frac{f_c}{z_1 - f_c} \beta_2 l_2 \right]^T \quad (4.101)$$

The end-effector is required to move on a constraint surface described by

$$\Psi(r(q)) = x + \gamma_s y + c = l_1 c_1 + l_2 c_{12} + \gamma_s (l_1 s_1 + l_2 s_{12}) + c = 0, \quad (4.102)$$

where γ_s and c are constant. Let $f = (f_x, f_y)^T$, then $D^T(q)\lambda$ can be written as

$$\begin{aligned} D^T(q)\lambda &= J_m^T(q)f \\ &= \begin{bmatrix} -f_x s_1 + f_y c_1 & -f_x s_{12} + f_y c_{12} \\ 0 & -f_x s_{12} + f_y c_{12} \end{bmatrix} \begin{bmatrix} \theta_{f,1} \\ \theta_{f,2} \end{bmatrix} = Y_f(q, f)\theta_f, \end{aligned} \quad (4.103)$$

where $\theta_{f,1} = l_1, \theta_{f,2} = l_2$. In the simulation, the exact masses of the two links are set as 17.4 and 4.8 kg, the exact lengths l_1 and l_2 of the links are set to 0.43 and 0.43 m, f_c is set as 50mm, z_1 is set as 0.55m. $\beta_1 = \beta_2 = 10000$ and γ_s is 0.

The parameters of the function of the constraint surface are set as $c = -0.52, \gamma_s = 0$. The initial position of the end effector is set at (520, 199). The desired motion trajectory is set as $x_d(t) = 520$ pixels, $y_d(t) = 200 + 20 * t$ pixels. The desired contact force is set as $20 + 5\sin(2t)$ Newton. In this simulation, the initial estimated parameters are set as $\hat{l}_1(0) = 0.4m, \hat{l}_2(0) = 0.5m, \hat{f}_c(0) = 40mm, \hat{z}_1(0) = 0.5m, \hat{\beta}_1(0) = \hat{\beta}_2(0) = 8000$, respectively. The control gains are set as $\alpha = 0.45, L_{ki} = 0.002I, L_d = 0.002I, L_f = 0.002I, K = 300/10^6I, \beta = 0.01, \gamma = 15, \kappa = 1$.

The simulation results are shown in figures 4.2 and 4.3. The results show the effectiveness of the proposed controller in dealing with uncertainties in the kinematics, dynamics and camera model since convergence of the image and force errors are guaranteed.

Next, the initial estimated parameters are increased to $\hat{l}_1(0) = 0.6m, \hat{l}_2(0) = 0.65m, \hat{f}_1(0) = 30mm, \hat{z}(0) = 0.7m, \hat{\beta}_1(0) = 6000, \hat{\beta}_2(0) = 12000, \hat{\delta}(0) = \pi/4$ and $\hat{\gamma}_s(0) = 0.2$. The control gains are set as $\alpha = 0.55, L_{ki} = 0.002I, L_d = 0.002I, L_f = 0.002I, K = 380/10^6I, \beta = 0.02, \gamma = 30, \kappa = 1$. The simulation results in figures 4.4 and 4.5 show that the image and force tracking errors converge to zero even in the presence of larger uncertainties in the system.

Figure 4.2: Image tracking error

Figure 4.3: Force tracking error

Figure 4.4: Image tracking error with larger uncertainties

Figure 4.5: Force tracking error with larger uncertainties

4.3.2 Adaptive Neural-network Vision and Force Tracking Control

In the previous section, the simulation results of adaptive vision-force tracking controller using regressors are presented. However, the structure of the constraint surface and Jacobian matrix is assumed to be known and the parameters can be linearly separated. In this simulation, uncertain constraint surface and Jacobian matrix are considered. The constraint surface in Cartesian space is described by

$$\Psi_1(x) = \sin(ax_1 + b) - x_2 = 0, \quad (4.104)$$

Note that in this constraint function, the parameters in $\Psi_1(x)$ can not be linearly separated.

The initial value of the adaptive Jacobian matrices $J_m(q)$ and $J_x(q)$ are set as constant matrices:

$$J_{m0}(q) = \begin{bmatrix} -0.1 & 0.2 \\ 0.7 & 0.4 \end{bmatrix}, \quad (4.105)$$

and

$$J_{x0}(q) = \begin{bmatrix} -100 & 150 \\ 400 & 300 \end{bmatrix}. \quad (4.106)$$

An image path in image space is obtained from the camera, The initial position of the end effector on the path is set at (520, 199). $x_d(t)$ is set as $x_d(t) = 520 + 5t$ pixels and $y_d(t)$ is obtained from the image path. The desired contact force is set as $20 + 5\sin(2t)$ Newton. In this simulation, the control gains are set as $\alpha = 100, \beta = 0.005, \gamma = 25, \kappa = 0.6, K_p = 0.01I, K_v = 200I$.

In this simulation, Gaussian RBF neural networks with input q were used. The centers were chosen so that they were evenly distributed to span the input space of the network. The distance of neural networks was fixed at 1.8 and the number of

neurons was set as 40. The gains for the networks were chosen as $k_1 = k_2 = k_3 = k_4 = 0.001$, $L_d = 0.01$, $L_m = 0.002$, $L_x = 0.02$, $L_f = 0.01$, $\bar{k}_m = 0.001$, $k_f = 0.001$.

The simulation results are shown in figures 4.6, 4.7 and 4.8. The results show the effectiveness of the proposed controller in dealing with uncertain structure of constraint surface and Jacobian matrices.

4.4 Conclusion

In this chapter, the stability problem of visually-servoed motion and force tracking control system with uncertain kinematics, dynamics and constraint surface has been studied. An adaptive Jacobian controller using regressors has been proposed. A neural network Jacobian controller has also been proposed to deal with uncertain structure of the Jacobian matrices and constraint surface. Lyapunov-like functions have also been presented for the stability analysis of the control systems. It has been shown that the robot end-effector can track the desired motion and force trajectories in the presence of the above-mentioned uncertainties. Simulation results have been presented to illustrate the performance of the proposed control law.

Figure 4.6: Image tracking error in X

Figure 4.7: Image tracking error in Y

Figure 4.8: Force tracking error

Chapter 5

Vision-force Cooperative Control of Multi-fingered Robot Hands Using Neural Networks

Cooperative control of multi-fingered robot hands has received increasing attention in recent years. Much effort has been made to study the control problem of multi-fingered robot hands. However, today's robot hands are still far less flexible than human hands because robot hands can not adapt to uncertainties in contact points of fingers with the object.

In this chapter, the vision based setpoint control problem of multi-fingered robot hands with uncertain kinematics, dynamics, object and camera parameters and contact points is studied. It is assumed that the parameters of the object and contact points are unknown. In most cases, regressors [11] can not be used because the uncertain parameters of Jacobian matrices can not be linearly separated. A new vision-force cooperative controller for multi-fingered robot hands with uncertainties using neural networks is proposed. An adaptive neural network controller is also proposed to deal with uncertain structure of the Jacobian matrix. It is shown that

stability can be achieved even in presence of these uncertainties.

5.1 Dynamics Equations and Problem Formulation

In this section, kinematics and dynamics of multi-fingered robot hands are presented, Consider a set of k fingers holding an object as illustrated in Figures 5.1 and 5.2. Let Σ denotes the Cartesian coordinate, Σ_o is the object coordinate frame fixed at the mass center of the object and moving with the object, Σ_{ci} is the contact point frame located at the contact point of the i^{th} finger and Σ_{ei} is the finger coordinate frame located at the i^{th} finger as shown in figure i . The velocity vector v_o of the object in Σ_o is related to the velocity vector v_{ci} at the contact point of the i^{th} finger in Σ_{ci} as

$$v_{ci} = J_{oi}v_o, \quad (5.1)$$

where $J_{oi} \in \mathbb{R}^{n_o \times n_o}$ denotes a Jacobian matrix from Σ_o to Σ_{ci} . The velocity vector v_{ci} in Σ_{ci} is related to the velocity vector v_{ei} of the i^{th} finger in Σ_{ei} as

$$v_{ci} = J_{fi}v_{ei}, \quad (5.2)$$

where $J_{fi} \in \mathbb{R}^{n_o \times n_i}$ denotes a Jacobian matrix from Σ_{ei} to Σ_{ci} . Since V_{ei} in Σ_{ei} is expressed as

$$v_{ei} = J_{ei}(q_i)\dot{q}_i, \quad (5.3)$$

where $q_i \in \mathbb{R}^{n_i}$ ($n_i > n_o$) is the joint coordinates of the i^{th} finger and $J_{ei}(q_i) \in \mathbb{R}^{n_o \times n_i}$ is the Jacobian matrix of Σ_{ei} in q_i . From equations (5.1), (5.2) and (5.3), the velocity of the joint variables $q = [q_1^T, \dots, q_k^T]^T$ and the velocity of the object v_o are

constrained by the following equation:

$$J_o v_o = J_e(q) \dot{q}, \quad (5.4)$$

where $J_o = [J_{o1}^T, \dots, J_{ok}^T]^T$ and $J_e(q) = \text{diag}\{J_{f1}J_{e1}(q_1), \dots, J_{fk}J_{ek}(q_k)\}$. In addition, let $r \in \mathbb{R}^{n_o}$ denotes the position and orientation vector of Σ_o in Σ , then the velocity vector $\dot{r}$ is related to the velocity vector v_o as

$$v_o = J_v(r) \dot{r}, \quad (5.5)$$

where $J_v(r)$ is a non-singular Jacobian mapping from Σ to Σ_o . From equations (5.4) and (5.5), the kinematic constraint between $\dot{r}$ and $\dot{q}$ is

$$J(r) \dot{r} = J_e(q) \dot{q}, \quad (5.6)$$

where $J(r) = J_o J_v(r)$.

Cameras that are fixed in the workspace are used to observe a feature on the object. A feature coordinate frame Σ_f fixed on the feature of the object is defined and $r_f \in \mathbb{R}^{n_o}$ is the position and orientation vector relative to the origin of Σ_f in Σ . Then $\dot{r}$ has the following relation with $\dot{r}_f$

$$\dot{r} = J_{of} \dot{r}_f \quad (5.7)$$

where $J_{of} \in \mathbb{R}^{n_o \times n_o}$ denotes a Jacobian matrix from Σ_f to Σ_o . From equations (5.6) and (5.7), one can get

$$J_f(r) \dot{r}_f = J_e(q) \dot{q}, \quad (5.8)$$

where

$$J_f(r) = J(r) J_{of}, \quad (5.9)$$

Figure 5.1: Multi-fingered robot holding an object

Let $x \in \mathbb{R}^p$ denote a vector of image feature parameters and $\dot{x}$ the corresponding vector of image feature parameter rates of change. The relationship between Cartesian space and image space is represented by [29],

$$\dot{x} = J_I(r_f)\dot{r}_f, \quad (5.10)$$

where $J_I(r_f) \in \mathbb{R}^{p \times n_o}$ is the image Jacobian matrix.

The dynamics equation of the i^{th} finger is described in the joint coordinates q_i as

$$M_i(q_i)\ddot{q}_i + (B_i + \frac{1}{2}\dot{M}_i(q_i) + S_i(q_i, \dot{q}_i))\dot{q}_i + g_i(q_i) = \tau_i - J_{ei}^T(q_i)J_{fi}^T f_{ei}, \quad (5.11)$$

where $M_i(q_i) \in \mathbb{R}^{n_i \times n_i}$ is the inertia matrix which is symmetric and positive definite for all q_i , $B_i \in \mathbb{R}^{n_i \times n_i}$ denotes the viscous friction matrix, $g_i(q_i) \in \mathbb{R}^{n_i}$ is the

Figure 5.2: Definitions of Coordinates

gravitational force, $\tau_i \in \mathbb{R}^{n_i}$ is the control input, $f_{ei} \in \mathbb{R}^{n_i}$ is the force and moment exerted on the object by the i^{th} finger and $S_i(q_i, \dot{q}_i)$ is a skew-symmetric matrix:

$$S_i(q_i, \dot{q}_i)\dot{q}_i = \frac{1}{2}\dot{M}_i(q_i)\dot{q}_i - \left\{ \frac{\partial}{\partial q_i} \dot{q}_i^T M_i(q_i) \dot{q}_i \right\}^T. \quad (5.12)$$

Then, the dynamic equation of the k fingers can be described in q as

$$M(q)\ddot{q} + \left(B + \frac{1}{2}\dot{M}(q) + S(q, \dot{q})\right)\dot{q} + g(q) = \tau - J_e^T(q)f_e, \quad (5.13)$$

where $M(q) = \text{diag}\{M_1(q_1), \dots, M_k(q_k)\}$, $B = \text{diag}\{B_1, \dots, B_k\}$, $g(q) = [g_1^T(q), \dots, g_k^T(q)]^T$, $S(q) = \text{diag}\{S_1(q_1, \dot{q}_1), \dots, S_k(q_k, \dot{q}_k)\}$, $\tau = [\tau_1^T, \dots, \tau_k^T]^T$ and $f_e = [f_{e1}^T, \dots, f_{ek}^T]^T$. Note that $M(q)$ is also symmetric and positive definite for all q and $S(q, \dot{q})$ is skew-symmetric.

The equation of motion of the object can be written in Σ as

$$M_o(r)\ddot{r} + \left(\frac{1}{2}\dot{M}_o(r) + S_o(r, \dot{r})\right)\dot{r} + g_o(r) = F, \quad (5.14)$$

where F is represented by f_{ei} as

$$F = J^T(r)f_e, \quad (5.15)$$

$M_o(r)$ is a positive definite inertial matrix, $S_o(r, \dot{r})$ is a skew-symmetric matrix came from Coriolis and F is the total force exerted on the object by the fingers. Substituting equation (5.7) into equation (5.14), one get

$$M_o(r)\frac{d}{dt}(J_{of}\dot{r}_f) + \left(\frac{1}{2}\dot{M}_o(r) + S_o(r, \dot{r})\right)J_{of}\dot{r} + g_o(r) = F, \quad (5.16)$$

From equations (5.15) and (5.8), it follows that

$$f_e = (J^+(r))^T F + Zf_{int}, \quad (5.17)$$

$$\dot{r}_f = J_f^+ J_e \dot{q} = J_r(q)\dot{q}. \quad (5.18)$$

where Z is a matrix of orthonormals which are generated from a set of independent vectors of the null space of J^T , $f_{int} \in \mathbb{R}^{n_f}$ represents the internal force which does not affect motion of the object.

Substituting equations (5.17) and (5.18) into equation (5.13) and using equation (5.16), one can rewrite the dynamics of coordinated system as follows:

$$\begin{aligned} & M(q)\ddot{q} + J_e^T(q)(J^+(r))^T M_o(r)J_{of}\ddot{r}_f + \{B + \frac{1}{2}\dot{M}(q) + S(q, \dot{q})\}\dot{q} \\ & + J_e^T(q)(J^+(r))^T M_o(r)\dot{J}_{of}\dot{r}_f + J_e^T(q)(J^+(r))^T \left(\frac{1}{2}\dot{M}_o(r) + S_o(r, \dot{r})\right)J_{of}\dot{r}_f \\ & + g(q) + J_e^T(q)(J^+(r))^T g_o(r) \\ & = -J_m^T(q)f_{int} + \tau, \end{aligned} \quad (5.19)$$

where

$$J_m(q) = Z^T J_e. \quad (5.20)$$

From equations (5.20), (5.6) and (5.18), it can be also concluded that

$$J_m(q)\dot{q} = Z^T J_e \dot{q} = Z^T J(r)\dot{r} = 0, \quad (5.21)$$

which indicates the geometric constraint on the joint angle velocity vector.

5.2 Vision Based Neural Network Control of Multi-fingered Robot Hands

In this section, a neural network setpoint controller for multi-fingered robot hands with uncertain kinematics and dynamics is presented. In this approach, the exact knowledge of the kinematics and dynamics are assumed to be unknown because the fingers need to grasp different objects and the contact points are uncertain. When the robot hands are grasping various objects, the matrix $J(r)$ changes according to the geometry of the object. In addition, $J(r)$ is not constant when the finger tip is rolling on a curved surface of the object. Hence, the inverse matrix $J^+(r)$ is usually nonlinear in its parameters. The matrix Z in equation (5.17) is also nonlinear in its parameters since Z is a matrix of orthonormals which are generated from a set of independent vectors of the null space of $J^T(r)$. Neural networks are used to compensate the uncertainty and the neural network weights are updated online by an update law.

The gravity terms in equation (5.19) can be approximated by a neural network as

$$g(q) + J_e^T(q)(J^+(r))^T g_o(r) = W_g \theta_g(q) + E_g, \quad (5.22)$$

5.2 Vision Based Neural Network Control of Multi-fingered Robot Hands 101

where $\theta_g(q)$ is a vector of activation functions, W_g is a constant matrix of network weights and E_g is an approximation error of the gravity terms.

The uncertainty in compensating the force is also approximated by a neural network as

$$J_m^T(q) = (W_{f1}\theta_f(q), \dots, W_{fn_I}\theta_f(q)) + E_f, \quad (5.23)$$

where $\theta_f(q, f_{int})$ is a vector of activation functions, W_f is a constant matrix of network weights and E_f is an approximation error of the force term.

The neural network control input for multi-fingered hands is proposed as

$$\begin{aligned} \tau = & -\hat{J}_x^T(q)K_p s(\Delta x) - \hat{J}_x^T(q)K_v \hat{J}_x(q)\dot{q} + \hat{J}_m(q, \hat{W}_f)^T f_{int} \\ & + \hat{J}_m(q, \hat{W}_f)^T R K_m ((f_{int} - f_d) + \gamma \int_0^t (f_{int} - f_d) d\tau) \\ & + \hat{W}_g \theta_g(q) - K_g \text{sgn}(y) - K_f f_{int}, \end{aligned} \quad (5.24)$$

where $\hat{J}_x(q) = \hat{J}_I(\hat{r}_f)\hat{J}_r(q)$ is a fixed estimation of $J_x(q) = J_I(r_f)J_r(q)$, $\hat{J}_m(q, \hat{W}_f) = (\hat{W}_{f1}\theta_f(q), \dots, \hat{W}_{fn_I}\theta_f(q))$ is an estimation of $J_m(q)$, $s(\Delta x) = (s_1(\Delta x_1), \dots, s_p(\Delta x_p))^T$, $\Delta x = x - x_d = (\Delta x_1, \dots, \Delta x_p)^T$ is a positional deviation from a desired position of the image feature x_d in image space. K_p and K_v are diagonal feedback gain matrices for the position error and the velocity, respectively, γ is a positive constant and f_d is a desired internal force. K_m is a gain matrix and K_g is a gain matrix for the sign function $\text{sgn}(y)$ and $y = \dot{q} + \alpha \hat{J}_x^+(q)s(\Delta x)$. K_f is a gain matrix chosen as

$$K_{fij} = \bar{k}_{ij} \text{sgn}(y_i f_{intj}), \quad (5.25)$$

where $\bar{k}_{ij}$ is a positive constant, y_i and f_{intj} are the i^{th} and j^{th} elements of y and f_{int} . We assume that $\hat{J}_I$ and $\hat{J}_r(q)$ are chosen so that

$$\|J_x(q) - \hat{J}_x(q)\| \leq p, \quad (5.26)$$

The estimated weights $\hat{W}_g$ and $\hat{W}_f$ in equation (5.24) are updated, respectively, by the following adaptive laws:

$$\begin{aligned}\dot{\hat{W}}_{gj}^T &= -L_{gj}^{-1}\theta_g(q)y_j, \\ \dot{\hat{W}}_{fij}^T &= -L_{fij}^{-1}\theta_f(q)y_jf_{inti}.\end{aligned}\quad (5.27)$$

where $\hat{W}_{gj}$ is the j^{th} row vector of $\hat{W}_g$, $\hat{W}_{fij}$ is the j^{th} row vector of $\hat{W}_f$, y_j is the j^{th} element of the vector y , f_{inti} is the i^{th} element of the vector f_{int} , L_{fij} and L_{gj} are positive definite gain matrices.

A rotation matrix R is designed so that

$$y_r^T R \Delta F = 0. \quad (5.28)$$

where $R = \text{diag}\{R_1, \dots, R_{k-1}\}$ and $R_i \in \mathbb{R}^{n_o \times n_o}$. $y_r = \hat{J}_m(q, \hat{W}_f)y$ and $\Delta F = K_m((f_{int} - f_d) + \gamma \int_o^t (f_{int} - f_d)d\tau)$. Since there are k fingers in the system, y_r can be written as $y_r = (y_{r1}^T, \dots, y_{rk-1}^T)^T$ and ΔF as $\Delta F = (\Delta F_1^T, \dots, \Delta F_{k-1}^T)^T$. In general, the position of the object can be partitioned as $r = (r_p, r_o)^T$, where r_p is the position vector and r_o is the orientation vector. Similarly, y_{ri} can also be partitioned as (y_{rip}, y_{rio}) and ΔF_i can be written as $(\Delta F_{ip}, \Delta F_{io})^T$. Hence the rotation matrix R_i can be introduced as,

$$R_i = \begin{bmatrix} R_{ip}(n_{ip}, \phi_{ip}) & 0 \\ 0 & R_{io}(n_{io}, \phi_{io}) \end{bmatrix}, \quad (5.29)$$

where n_{ip} is a unit vector normal to both the vectors y_{rip} and ΔF_{ip} (see figure 5.3). n_{io} is a unit vector normal to both the vectors y_{rio} and ΔF_{io} , ϕ_{ip} is the angle between ΔF_{ip} and $R_{ip}\Delta F_{ip}$, which can be determined from the angle ϕ between ΔF_{ip} and y_{rip} (see figure 5.3). ϕ_{io} is the angle between ΔF_{io} and $R_{io}\Delta F_{io}$, which can be determined from the angle between ΔF_{io} and y_{rio} . The rotation matrix R_{ip} is to rotate the vector

ΔF_{ip} about the axis n_{ip} , so that the vector $R_{ip}\Delta F_{ip}$ is perpendicular to the vector y_{rip} , as shown in Figure 5.3. Here the symbol $\times$ means cross product. The rotation matrix R_{io} can be similarly designed. However, if the vector y_{ri} are independent of r_{io} , then the vector y_{rio} is equal to zero and the rotation matrix R_{io} can be set as an identity matrix.

Figure 5.3: Rotation angles

Substituting equations (5.24), (5.22) and (5.23) into equation (5.19), the following closed-loop equation can be obtained:

$$\begin{aligned}
 & M(q)\ddot{q} + J_e^T(q)(J^+(r))^T M_o(r) J_{of} \ddot{r}_f + \{B + \frac{1}{2}\dot{M}(q) + S(q, \dot{q})\}\dot{q} \\
 & + J_e^T(q)(J^+(r))^T M_o(r) \dot{J}_{of} \dot{r}_f + J_e^T(q)(J^+(r))^T (\frac{1}{2}\dot{M}_o(r) + S_o(r, \dot{r})) J_{of} \dot{r}_f \\
 & + \hat{J}_x^T(q) K_p s(\Delta x) + \hat{J}_x^T(q) K_v \hat{J}_x(q) \dot{q} + \Delta W_g \theta_g(q) + \Delta W_f \theta_f(q) f_{int} \\
 & = \hat{J}_m^T(q) R K_m ((f_{int} - f_d) + \gamma \int_0^t (f_{int} - f_d) d\tau) - E_f f_{int} - E_g - K_g \text{sgn}(y) - K_f f_{int}, \quad (5.30)
 \end{aligned}$$

where $\Delta W_f = W_f - \hat{W}_f$ and $\Delta W_g = W_g - \hat{W}_g$.

To carry out the stability analysis for the closed-loop system, a Lyapunov-like func-

tion candidate is defined as

$$\begin{aligned} V = & \frac{1}{2}\dot{q}^T M(q)\dot{q} + \frac{1}{2}\dot{r}_f^T J_{of}^T M_o(r) J_{of} \dot{r}_f + \alpha s^T(\Delta x)(\hat{J}_x^+(q))^T M(q)\dot{q} \\ & + \alpha s^T(\Delta x)(\hat{J}_x^+(q))^T J_e^T(q)(J^+(r))^T M_o(r) J_{of} \dot{r}_f \\ & + \frac{1}{2}\sum_{i=1}^{n_I} \sum_{j=1}^{k_{ni}} \Delta W_{fij} L_{fij} \Delta W_{fij}^T + \frac{1}{2}\sum_{j=1}^{k_{ni}} \Delta W_{gj} L_{gj} \Delta W_{gj}^T + \sum_{j=1}^p (k_{pj} + \alpha k_{vj}) S_j(\Delta x_j), \end{aligned} \quad (5.31)$$

To show the positive definiteness of the Lyapunov-like function candidate V in equation (5.31), note that

$$\begin{aligned} V = & \frac{1}{4}\dot{q}^T M(q)\dot{q} + \frac{1}{4}\dot{r}_f^T J_{of}^T M_o(r) J_{of} \dot{r}_f + \sum_{j=1}^p (k_{pj} + \alpha k_{vj}) S_j(\Delta x_j) \\ & + \frac{1}{4}(\dot{q} + 2\alpha \hat{J}_x^+(q)s(\Delta x))^T M(q)(\dot{q} + 2\alpha \hat{J}_x^+(q)s(\Delta x)) \\ & + \frac{1}{4}(J_{of} \dot{r}_f + 2\alpha J^+(r) J_e(q) \hat{J}_x^+(q)s(\Delta x))^T M_o(r)(J_{of} \dot{r}_f + 2\alpha J^+(r) J_e(q) \hat{J}_x^+(q)s(\Delta x)) \\ & - \alpha^2 s^T(\Delta x)((\hat{J}_x^+(q))^T M(q) \hat{J}_x^+(q) + (\hat{J}_x^+(q))^T J_e^T(q)(J^+(r))^T M_o(r) J^+(r) J_e(q) \hat{J}_x^+(q))s(\Delta x) \\ & + \frac{1}{2}\sum_{i=1}^{n_I} \sum_{j=1}^{k_{ni}} \Delta W_{fij} L_{fij} \Delta W_{fij}^T + \frac{1}{2}\sum_{j=1}^{k_{ni}} \Delta W_{gj} L_{gj} \Delta W_{gj}^T \\ \geq & \frac{1}{4}\dot{q}^T M(q)\dot{q} + \frac{1}{4}\dot{r}_f^T J_{of}^T M_o(r) J_{of} \dot{r}_f + \sum_{j=1}^p (k_{pj} \bar{c}_j + \alpha(k_{vj} \bar{c}_j - \alpha \lambda_m)) s_j^2(\Delta x_j) \\ & + \frac{1}{2}\sum_{i=1}^{n_I} \sum_{j=1}^{k_{ni}} \Delta W_{fij} L_{fij} \Delta W_{fij}^T + \frac{1}{2}\sum_{j=1}^{k_{ni}} \Delta W_{gj} L_{gj} \Delta W_{gj}^T, \end{aligned} \quad (5.32)$$

where $\lambda_m = \lambda_{\max}[\hat{J}_I(\hat{r}_f)(\hat{J}_r^+(q))^T M(q) \hat{J}_r^+(q) \hat{J}_I^T(\hat{r}_f) + \hat{J}_I(\hat{r}_f)(\hat{J}_r^+(q))^T J_e^T(q)(J^+(r))^T M_o(r) J^+(r) J_e(q) \hat{J}_r^+(q) \hat{J}_I^T(\hat{r}_f)]$ and $\lambda_{\max}[A]$ denote the maximum eigenvalues of the matrix A . Hence, V is positive definite in Δx , $\dot{q}$, ΔW_{gk} and ΔW_{fk} since α and k_v can be chosen so that

$$k_{vj} \bar{c}_j - \alpha \lambda_m > 0. \quad (5.33)$$

Then we will show this will lead to a Lyapunov-like function for the stability analysis

5.2 Vision Based Neural Network Control of Multi-fingered Robot Hands 105

of the multi-finger control problem. Differentiating V with respect to time, one has

$$\begin{aligned}
\frac{d}{dt}V = & \dot{q}^T M(q) \ddot{q} + \frac{1}{2} \dot{q}^T \dot{M}(q) \dot{q} + \dot{r}_f^T J_{of}^T M_o(r) J_{of} \ddot{r}_f + \dot{r}_f^T J_{of}^T M_o(r) \dot{J}_{of} \dot{r}_f \\
& + \frac{1}{2} \dot{r}_f^T J_{of}^T \dot{M}_o(r) J_{of} \dot{r}_f + \alpha s^T(\Delta x) (\hat{J}_x^+(q))^T M(q) \ddot{q} + \alpha s^T(\Delta x) (\hat{J}_x^+(q))^T \dot{M}(q) \dot{q} \\
& + \frac{d}{dt} \{ \alpha s^T(\Delta x) (\hat{J}_x^+(q))^T \} M(q) \dot{q} + \alpha s^T(\Delta x) (\hat{J}_x^+(q))^T J_e^T(q) (J^+(r))^T M_o(r) J_{of} \ddot{r}_f \\
& + \alpha s^T(\Delta x) (\hat{J}_x^+(q))^T J_e^T(q) (J^+(r))^T M_o(r) \dot{J}_{of} \dot{r}_f \\
& + \alpha s^T(\Delta x) (\hat{J}_x^+(q))^T J_e^T(q) (J^+(r))^T \dot{M}_o(r) J_{of} \dot{r}_f \\
& + \alpha \frac{d}{dt} \{ s^T(\Delta x) (\hat{J}_x^+(q))^T J_e^T(q) (J^+(r))^T \} M_o(r) J_{of} \dot{r}_f \\
& - \sum_{j=1}^{kni} \Delta W_{gj} L_{gj} \dot{W}_{gj}^T - \sum_{i=1}^{nI} \sum_{j=1}^{kni} \Delta W_{fij} L_{fij} \dot{W}_{fij}^T + s^T(\Delta x) (K_p + \alpha K_v) \dot{x}. \quad (5.34)
\end{aligned}$$

Substitute equation (5.30) into equation (5.34) and using equations (5.28) and (5.27) to get

$$\frac{d}{dt}V = -W, \quad (5.35)$$

where

$$\begin{aligned}
W = & \dot{q}^T (B + \hat{J}_x^T(q) K_v \hat{J}_x(q)) \dot{q} + \alpha s^T(\Delta x) K_p s(\Delta x) \\
& - s^T(\Delta x) (K_p + \alpha K_v) (J_x(q) - \hat{J}_x(q)) \dot{q} \\
& + y^T (E_f f_{int} + E_g + K_g \operatorname{sgn}(y) + K_f f_{int}) + h. \quad (5.36)
\end{aligned}$$

and

$$\begin{aligned}
h = & \alpha \{ s^T(\Delta x) (\hat{J}_x^+(q))^T (B - \frac{1}{2} \dot{M}(q) + S(q, \dot{q})) \dot{q} \\
& + s^T(\Delta x) (\hat{J}_x^+(q))^T J_e^T(q) (J^+(r))^T (-\frac{1}{2} \dot{M}_o(r) + S_o(r, \dot{r})) J_{of} \dot{r}_f \\
& - \dot{s}^T(\Delta x) (\hat{J}_x^+(q))^T M(q) \dot{q} - s^T(\Delta x) (\hat{J}_x^+(q))^T M(q) \dot{q} \\
& - s^T(\Delta x) (\hat{J}_x^+(q))^T J_e^T(q) (J^+(r))^T M_o(r) \dot{J}_{of} \dot{r}_f \\
& - \frac{d}{dt} [s^T(\Delta x) (\hat{J}_x^+(q))^T J_e^T(q) (J^+(r))^T] M_o(r) J_{of} \dot{r}_f \}. \quad (5.37)
\end{aligned}$$

Therefore, the function V represents a Lyapunov-like function candidate for the control of multi-fingered robot hands with uncertainties. Next, it will be shown that the time derivative of the Lyapunov-like function is negative definite in $s(\Delta x)$ and $\dot{q}$. As seen from equation(5.35), this is equivalent to show that W is positive definite in $s(\Delta x)$ and $\dot{q}$. From equation (5.37), there exist a constant c_0 so that

$$|h| \leq \alpha c_0 \|\dot{q}\|^2. \quad (5.38)$$

Substituting the above inequality into equation (5.36) yields

$$\begin{aligned} W \geq & \dot{q}(B + \hat{J}_x^T(q)K_v\hat{J}_x(q) - \alpha c_0 I)\dot{q} + \alpha s^T(\Delta x)K_p s(\Delta x) \\ & - s^T(\Delta x)(K_p + \alpha K_v)(J_x(q) - \hat{J}_x(q))\dot{q} + y^T(E_f f_{int} + E_g + K_g \text{sgn}(y) + K_f f_{int}). \end{aligned} \quad (5.39)$$

Next, a sufficient condition will be derived to guarantee the positiveness of W . From the above inequality, Let $k_{fij} > |E_{fij}|$ and $k_g > \|E_g\|$, one has

$$W \geq (\lambda_1 - \alpha c_0)\|\dot{q}\|^2 + \alpha k_{pmin}\|s(\Delta x)\|^2 - p(k_{pmax} + \alpha k_{vmax})\|\dot{q}\|\|s(\Delta x)\|, \quad (5.40)$$

where $\lambda_1 = \lambda_{min}[B + \hat{J}_x^T(q)K_v\hat{J}_x(q)]$, $k_{pmin} = \lambda_{min}[K_p]$, $k_{pmax} = \lambda_{max}[K_p]$ and $k_{vmax} = \lambda_{max}[K_v]$. Then one obtains

$$W \geq (\lambda_1 - \alpha c_0 - \frac{1}{2}p(k_{pmax} + \alpha k_{vmax}))\|\dot{q}\|^2 + (\alpha k_{pmin} - \frac{1}{2}p(k_{pmax} + \alpha k_{vmax}))\|s(\Delta x)\|^2. \quad (5.41)$$

Hence, if

$$\begin{aligned} \lambda_1 - \alpha c_0 - \frac{1}{2}p(k_{pmax} + \alpha k_{vmax}) &> 0 \\ \alpha k_{pmin} - \frac{1}{2}p(k_{pmax} + \alpha k_{vmax}) &> 0, \end{aligned} \quad (5.42)$$

or

$$\min\left\{\frac{2(\bar{\lambda}_1 - \frac{\alpha c_0}{k_{vmax}})}{\bar{a} + \alpha}, \frac{2\bar{a}\alpha\lambda_2}{\bar{a} + \alpha}\right\} > p, \quad (5.43)$$

where $\bar{a} = \frac{k_{pmax}}{k_{vmax}}$, $\bar{\lambda}_1 = \frac{\lambda_1}{k_{vmax}}$ and $\lambda_2 = \frac{k_{pmin}}{k_{pmax}}$ then $W \geq 0$.

We are now in a position to state the following Theorem:

Theorem 5.1 *The closed-loop system described by equation (5.30), with uncertain Jacobian matrices gives rise to the convergence of $(\Delta x, \dot{q}, f_{int})$ to $(0, 0, f_d)$ as $t \rightarrow 0$ if the feedback gains K_p and K_v are chosen to satisfy conditions (5.33), (5.43) and $\hat{J}_I(\hat{r}_f)$ and $\hat{J}_r(q)$ are chosen to satisfy condition (5.26).*

Proof:

When conditions (5.33) and (5.43) are satisfied and $\hat{J}_x(q)$ is chosen to satisfy condition (5.26), both V and W are positive definite, one has

$$\dot{V} = -W \leq 0, \quad (5.44)$$

where V is positive definite in $s(\Delta x), \dot{q}, \dot{r}_f, \Delta W_{gj}$ and ΔW_{fij} and W is a positive semidefinite function. Integrating both sides of equation (5.35), one has

$$V - V(0) = - \int_0^t W(\tau) d\tau \leq 0. \quad (5.45)$$

Hence, V is bounded and one can conclude that $s(\Delta x), \dot{q}, \dot{r}_f, \Delta W_{gj}$ and ΔW_{fij} are bounded vectors. One can conclude that $s(\Delta x), \dot{q}$ are square integrable functions.

Substituting equations (5.7), (5.8) and (5.9) into (5.30), one has

$$\begin{aligned} & (M(q) + \bar{M}_o)\ddot{q} + (B + \frac{1}{2}\dot{M}(q) + S(q, \dot{q}) + \frac{1}{2}\dot{\bar{M}}_o + \bar{S}_o)\dot{q} \\ & + \hat{J}_x^T(q)K_p s(\Delta x) + \hat{J}_x^T(q)K_v \hat{J}_x(q)\dot{q} \\ & + \Delta W_g \theta_g(q) + \Delta W_f \theta_f(q) f_{int} + E_f f_{int} + E_g + K_g \text{sgn}(y) + K_f f_{int} \\ & = \hat{J}_m^T(q, \hat{W}_f) R K_m ((f_{int} - f_d) + \gamma \int_0^t (f_{int} - f_d) d\tau), \end{aligned} \quad (5.46)$$

where $\bar{M}_o = J_e^T(q)(J^+(r))^T M_o(r) J^+(r) J_e(q)$ and $\bar{S}_o = J_e^T(q)(J^+(r))^T S_o(r, \dot{r}) J^+(r) J_e(q)$. Next, let $M(q) + \bar{M}_o = \bar{M}$, then multiply both sides of equation (5.46) by $J_m(q)\bar{M}^{-1}$ and using $J_m(q)\ddot{q} = -\dot{J}_m(q)\dot{q}$, one has

$$\begin{aligned} & -\dot{J}_m(q)\dot{q} + J_m(q)\bar{M}^{-1}\{(B + \frac{1}{2}\dot{M}(q) + S(q, \dot{q}) + \frac{1}{2}\dot{\bar{M}}_o \\ & + \bar{S}_o)\dot{q} + \hat{J}_x^T(q)K_p s(\Delta x) + \hat{J}_x^T(q)K_v \hat{J}_x(q)\dot{q} + \Delta W_g \theta_g(q) \\ & + \Delta W_f \theta_f(q)f_{int} + E_f f_{int} + E_g + K_g \text{sgn}(y) + K_f f_{int}\} \\ & = J_m(q)\bar{M}^{-1}\hat{J}_m^T(q, \hat{W}_f)RK_m((f_{int} - f_d) \\ & + \gamma \int_o^t (f_{int} - f_d)d\tau), \end{aligned} \quad (5.47)$$

The force error is subject to

$$\Delta(t) = (f_{int} - f_d) + \gamma(K_m - \bar{E})^{-1}K_m \int_o^t (f_{int} - f_d)d\tau. \quad (5.48)$$

where

$$\begin{aligned} \Delta(t) &= (K_m - \bar{E})^{-1}\kappa r(t), \\ \bar{E} &= J_m(q)\bar{M}^{-1}(\Delta W_f \theta_f(q) + E_f + K_f) \\ \kappa &= R^T(J_m(q)\bar{M}^{-1}\hat{J}_m^T(q, \hat{W}_f))^{-1} \\ r(t) &= -\dot{J}_m(q)\dot{q} + J_m(q)\bar{M}^{-1}\{(B + \frac{1}{2}\dot{M}(q) + S(q, \dot{q}) + \frac{1}{2}\dot{\bar{M}}_o + \bar{S}_o)\dot{q} \\ & + \hat{J}_x^T(q)K_p s(\Delta x) + \hat{J}_x^T(q)K_v \hat{J}_x(q)\dot{q} + \Delta W_g \theta_g(q) + \Delta W_f \theta_f(q)f_d \\ & + E_g + K_g \text{sgn}(y) + E_f f_d + K_f f_d\}. \end{aligned} \quad (5.49)$$

Since γ is a positive constant and all terms in left hand side of equation (5.48) are bounded, one can conclude that $f_{int} - f_d$ and its integration are bounded. The boundedness of the force errors implies that $\ddot{q}$ is bounded because $s(\Delta x)$, $\dot{q}$, $\dot{r}_f$, ΔW_{gj} and ΔW_{fij} are bounded. Hence, $s(\Delta x)$, $\dot{q}$ are uniform continuous and it follows

that [61]

$$\Delta x \rightarrow 0, \dot{q} \rightarrow 0 \text{ as } t \rightarrow \infty. \quad (5.50)$$

The maximum invariant satisfies:

$$\Delta_\infty = (f_\infty - f_d) + \gamma(K_m - \bar{E}_\infty)^{-1}K_m \int_0^t (f_\infty - f_d) d\tau. \quad (5.51)$$

where $\Delta_\infty = (K_m - \bar{E}_\infty)^{-1}\kappa_\infty J_m(q_d)\bar{M}^{-1}\{\Delta W_{f_\infty}\theta_f(q_d)f_d + \Delta W_{g_\infty}\theta_g(q_d) + E_g + E_f f_d\}$, $\kappa_\infty = R_\infty^T(J_m(q_d)\bar{M}^{-1}\hat{J}_m^T(q_d, \hat{W}_{f_\infty}))^{-1}$, $\bar{E}_\infty = J_m(q_d)\bar{M}^{-1}(\Delta W_{f_\infty}\theta_f(q_d) + E_f)$. q_d is a desired joint configuration for x_d and is defined for the purpose of analysis only. Therefore, K_m can be chosen so that the eigenvalues of $-(K_m - \bar{E}_\infty)^{-1}K_m - [(K_m - \bar{E}_\infty)^{-1}K_m]^T$ are in the left-half plane and hence $f_{int} \rightarrow f_d$ as $t \rightarrow \infty$.

Remark 5.1 When $\Delta x \rightarrow 0$ and $\dot{q} \rightarrow 0$, the vector y reduces to zero but in this case, equation (5.28) is satisfied for any R . Hence, the updating of the rotation matrix is not required when $\Delta x \rightarrow 0$ and $\dot{q} \rightarrow 0$. That is, R approaches a constant matrix when Δx and $\dot{q}$ approach zero.

Remark 5.2 In [10], it is assumed that the uncertain parameters can be linearly separated. However, in some applications of multi-fingered robot hands, the uncertain parameters of the Jacobian matrices cannot be linearly parameterized. For example, when the finger tip is rolling on a surface of an object during manipulation. In this paper, neural networks are proposed to compensate the approximation errors so that the above problem is solved.

Remark 5.3 It is assumed in [10] that the estimated Jacobian matrix satisfies

$$\hat{J}_m(q)\dot{q} = 0. \quad (5.52)$$

and hence only a limited class of kinematic uncertainty is considered. In this result, we introduce a new rotation matrix so that the above assumption (5.52) is removed. Hence the proposed controller can deal with general kinematic uncertainties.

5.3 Vision Based Adaptive Jacobian Control of Multi-fingered Robot Hands

When the finger tips are rolling on the surface of the objects or the finger tips are soft and deformable, the parameters and the structure of the Jacobian matrices are difficult to estimate exactly. To solve this problem, an adaptive Jacobian controller is proposed for multi-fingered robot hands in this section.

The Jacobian matrix $J_x(q)$ can be approximated by neural networks as

$$J_x(q) = (W_{x1}\theta_x(q), \dots, W_{xkni}\theta_x(q)) + E_x. \quad (5.53)$$

where $W_{xi}\theta_x(q)$ is the i^{th} column vectors of $J_x(q)$ and E_x is the estimation error.

When the parameters and structure of the Jacobian matrix are uncertain, the Jacobian matrix is estimated as:

$$\hat{J}_x(q, \hat{W}_x) = (\hat{W}_{x1}\theta_x(q), \dots, \hat{W}_{xkni}\theta_x(q)). \quad (5.54)$$

where $\hat{W}_{xi}\theta_x(q)$ is the i^{th} column vectors of $\hat{J}_x(q, \hat{W}_x)$.

The adaptive Jacobian controller is proposed as

$$\begin{aligned} \tau = & -\hat{J}_x^T(q, \hat{W}_x)K_p s(\Delta x) - \hat{J}_x^T(q, \hat{W}_x)K_v \hat{J}_x(q, \hat{W}_x)\dot{q} + \hat{J}_m(q, \hat{W}_f)^T f_{int} \\ & + \hat{J}_m(q, \hat{W}_f)^T R K_m ((f_{int} - f_d) + \gamma \int_0^t (f_{int} - f_d) d\tau) \\ & + \hat{W}_g \theta_g(q) - K_g \text{sgn}(y) - K_f f_{int}, \end{aligned} \quad (5.55)$$

5.3 Vision Based Adaptive Jacobian Control of Multi-fingered Robot Hands 111

where $s(\Delta x) = (s_1(\Delta x_1), \dots, s_p(\Delta x_p))^T$, $\Delta x = x - x_d = (\Delta x_1, \dots, \Delta x_p)^T$ is a positional deviation from a desired position of the image feature x_d in image space. K_p and K_v are diagonal feedback gain matrices for the position error and the velocity, respectively, γ is a positive constant and f_d is a desired internal force. K_m is a gain matrix and K_g is a gain matrix for the sign function $sgn(y)$ and $y = \dot{q} + \alpha \hat{J}_x^+(q)s(\Delta x)$. K_f is a gain matrix defined as

$$K_{fij} = \bar{k}_{ij} sgn(y_i f_{intj}), \quad (5.56)$$

where $\bar{k}_{ij}$ is a positive constant, y_i and f_{intj} are the i^{th} and j^{th} elements of y and f_{int} . The neural networks weights $\hat{W}_g$, $\hat{W}_f$ and $\hat{W}_x$ in equations (5.55) and (5.54) are updated, respectively, by the following adaptive laws:

$$\begin{aligned} \dot{\hat{W}}_{gj}^T &= proj(\Omega_{gj}), \\ \dot{\hat{W}}_{fij}^T &= proj(\Omega_{fij}), \\ \dot{\hat{W}}_{xij}^T &= proj(\Omega_{xij}), \end{aligned} \quad (5.57)$$

where

$$\begin{aligned} \Omega_{gj} &= -L_{gj}^{-1} \theta_g(q) y_j, \\ \Omega_{fij} &= -L_{fij}^{-1} \theta_f(q) y_j f_{inti}, \\ \Omega_{xij} &= L_{xij}^{-1} \theta_x(q) (k_{pi} + \alpha k_{vi}) s(\Delta x_i) \dot{q}_j, \end{aligned} \quad (5.58)$$

$\hat{W}_{xij}$ is the i^{th} row vector of $\hat{W}_{xj}$ and L_{xij} is a gain matrix for $\hat{W}_{xij}$, the function

$proj(\Omega_{gj})$ is a projection algorithm defined as [59]

$$proj(\Omega_{gj}) = \begin{cases} \Omega_{gj} & \text{if } \hat{W}_{gj} > \underline{W}_{gj} \\ \Omega_{gj} & \text{if } \hat{W}_{gj} = \underline{W}_{gj} \text{ and } \Omega_{gj} \geq 0 \\ 0 & \text{if } \hat{W}_{gj} = \underline{W}_{gj} \text{ and } \Omega_{gj} < 0 \\ 0 & \text{if } \hat{W}_{gj} = \bar{W}_{gj} \text{ and } \Omega_{gj} > 0 \\ \Omega_{gj} & \text{if } \hat{W}_{gj} = \bar{W}_{gj} \text{ and } \Omega_{gj} \leq 0 \\ \Omega_{gj} & \text{if } \hat{W}_{gj} < \bar{W}_{gj} \end{cases} \quad (5.59)$$

where $\underline{W}_{gj}$ and $\bar{W}_{gj}$ are the lower and upper bounds of W_{gj} . The projection algorithms $proj(\Omega_f)$ and $proj(\Omega_x)$ can be similarly defined as above. The functions $proj(\cdot)$ are defined to ensure that $\hat{J}_x^T(q, \hat{W}_x)$ and $\hat{J}_m(q, \hat{W}_f)$ are bounded during adaptation.

Substituting equations (5.55), (5.22) and (5.23) into equation (5.19), to get the following closed-loop equation:

$$\begin{aligned} & M(q)\ddot{q} + J_e^T(q)(J^+)^T M_o(r) J_{of} \ddot{r}_f + \{B + \frac{1}{2}\dot{M}(q) + S(q, \dot{q})\}\dot{q} \\ & + J_e^T(q)(J^+)^T M_o(r) \dot{J}_{of} \dot{r}_f + J_e^T(q)(J^+)^T (\frac{1}{2}\dot{M}_o(r) + S_o(r, \dot{r})) J_{of} \dot{r}_f \\ & + \hat{J}_x^T(q, \hat{W}_x) K_p s(\Delta x) + \hat{J}_x^T(q, \hat{W}_x) K_v \hat{J}_x(q, \hat{W}_x) \dot{q} + \Delta W_g \theta_g(q) + \Delta W_f \theta_f(q) f_{int} \\ & = \hat{J}_m^T(q, \hat{W}_f) R K_m ((f_{int} - f_d) + \gamma \int_0^t (f_{int} - f_d) d\tau) \\ & - E_g - E_f f_{int} - K_g \text{sgn}(y) - K_f f_{int}, \end{aligned} \quad (5.60)$$

To prove the stability of the adaptive Jacobian controller, a Lyapunov-like function is defined as

$$V_1 = V + \frac{1}{2} \sum_{j=1}^{kni} \sum_{i=1}^p \Delta W_{xij} L_{xij} \Delta W_{xij}^T. \quad (5.61)$$

where V is defined as in equation (5.31) and $\Delta W_{xij} = W_{xij} - \hat{W}_{xij}$.

5.3 Vision Based Adaptive Jacobian Control of Multi-fingered Robot Hands 113

Next, differentiate V_1 with respect to time, to get

$$\frac{d}{dt}V_1 = \dot{V} - \sum_{j=1}^{kni} \sum_{i=1}^p \Delta W_{xij} L_{xij} \dot{W}_{xij}^T. \quad (5.62)$$

Using equations (5.19) and (5.55), one has

$$\frac{d}{dt}V_1 = -W_1 \quad (5.63)$$

where

$$\begin{aligned} W_1 = & \dot{q}^T (B + \hat{J}_x^T(q, \hat{W}_x) K_v \hat{J}_x(q, \hat{W}_x)) \dot{q} + \alpha s^T(\Delta x) K_p s(\Delta x) \\ & - s^T(\Delta x) (K_p + \alpha K_v) (J_x(q) - \hat{J}_x(q, \hat{W}_x)) \dot{q} + y^T \Delta W_g \theta_g(q) + y^T \Delta W_f \theta_f(q) f_{int} \\ & + y^T (E_g + E_f f_{int} + K_g \text{sgn}(y) + K_f f_{int}) + h_1 \\ & + \sum_{j=1}^{kni} \sum_{i=1}^p \Delta W_{xij} L_{xij} \dot{W}_{xij}^T + \sum_{i=1}^{n_l} \sum_{j=1}^{kni} \Delta W_{fij} L_{fij} \dot{W}_{fij}^T + \sum_{j=1}^{kni} \Delta W_{gj} L_{gj} \dot{W}_{gj}^T. \end{aligned} \quad (5.64)$$

where

$$\begin{aligned} h_1 = & \alpha \{ s^T(\Delta x) (\hat{J}_x^+(q, \hat{W}_x))^T (B - \frac{1}{2} \dot{M}(q) + S(q, \dot{q})) \dot{q} \\ & + s^T(\Delta x) (\hat{J}_x^+(q, \hat{W}_x))^T J_e^T(q) (J^+(r))^T (-\frac{1}{2} \dot{M}_o(r) + S_o(r, \dot{r})) J_{of} \dot{r}_f \\ & - s^T(\Delta x) (\hat{J}_x^+(q, \hat{W}_x))^T M(q) \dot{q} - s^T(\Delta x) (\hat{J}_x^+(q, \hat{W}_x))^T M(q) \dot{q} \\ & - s^T(\Delta x) (\hat{J}_x^+(q, \hat{W}_x))^T J_e^T(q) (J^+(r))^T M_o(r) J_{of} \dot{r}_f \\ & - \frac{d}{dt} [s^T(\Delta x) (\hat{J}_x^+(q, \hat{W}_x))^T J_e^T(q) (J^+(r))^T] M_o(r) J_{of} \dot{r}_f \}. \end{aligned} \quad (5.65)$$

From equation (5.65), there exist a constant c_0 so that

$$|h_1| \leq \alpha c_0 \|\dot{q}\|^2. \quad (5.66)$$

Using equations (5.53), (5.54) and (5.58), one has

$$\begin{aligned}
 W_1 = & \dot{q}^T (B + \hat{J}_x^T(q, \hat{W}_x) K_v \hat{J}_x(q, \hat{W}_x)) \dot{q} + \alpha s^T(\Delta x) K_p s(\Delta x) \\
 & - s^T(\Delta x) (K_p + \alpha K_v) E_x \dot{q} + y^T (E_g + E_f f_{int} + K_g \operatorname{sgn}(y) + K_f f_{int}) + h_1 \\
 & + \sum_{j=1}^{k_{ni}} \sum_{i=1}^p \Delta W_{xij} L_{xij} (\dot{W}_{xij}^T - \Omega_{xij}) + \sum_{i=1}^{n_l} \sum_{j=1}^{k_{ni}} \Delta W_{fij} L_{fij} (\dot{W}_{fij}^T - \Omega_{fij}) \\
 & + \sum_{j=1}^{k_{ni}} \Delta W_{gj} L_{gj} (\dot{W}_{gj}^T - \Omega_{gj}). \quad (5.67)
 \end{aligned}$$

Then one can get

$$\begin{aligned}
 W_1 \geq & \dot{q}^T (B + \hat{J}_x^T(q, \hat{W}_x) K_v \hat{J}_x(q, \hat{W}_x)) \dot{q} + \alpha s^T(\Delta x) K_p s(\Delta x) \\
 & - s^T(\Delta x) (K_p + \alpha K_v) E_x \dot{q} + y^T (E_g + E_f f_{int} + K_g \operatorname{sgn}(y) + K_f f_{int}) + h_1. \quad (5.68)
 \end{aligned}$$

Since E_x is bounded, let b_{ex} be the upper bound of E_x , then

$$s^T(\Delta x) (K_p + \alpha K_v) E_x \dot{q} \leq \frac{1}{2} b_{ex} (k_{pmax} + \alpha k_{vmax}) (\|s(\Delta x)\|^2 + \|\dot{q}\|^2). \quad (5.69)$$

Substituting inequality (5.66) into equation (5.68) and using equation (5.69) yields

$$\begin{aligned}
 W_1 \geq & (\lambda_1 - \alpha c_0 - \frac{1}{2} b_{ex} (k_{pmax} + \alpha k_{vmax})) \|\dot{q}\|^2 \\
 & + (\alpha k_{pmin} - \frac{1}{2} b_{ex} (k_{pmax} + \alpha k_{vmax})) \|s(\Delta x)\|^2 \\
 & + y^T (E_g + E_f f_{int} + K_g \operatorname{sgn}(y) + K_f f_{int}), \quad (5.70)
 \end{aligned}$$

where $\lambda_1 = \lambda_{min}[B + \hat{J}_x^T(q, \hat{W}_x) K_v \hat{J}_x(q, \hat{W}_x)]$, $k_{pmin} = \lambda_{min}[K_p]$, $k_{pmax} = \lambda_{max}[K_p]$ and $k_{vmax} = \lambda_{max}[K_v]$. When K_g and $\bar{k}_{ij}$ are chosen sufficiently large so that $\|K_g\| > \|E_g\|$ and $\bar{k}_{ij} > |E_{fij}|$, one has

$$W_1 \geq (\lambda_1 - \alpha c_0 - \frac{1}{2} b_{ex} (k_{pmax} + \alpha k_{vmax})) \|\dot{q}\|^2 + (\alpha k_{pmin} - \frac{1}{2} b_{ex} (k_{pmax} + \alpha k_{vmax})) \|s(\Delta x)\|^2. \quad (5.71)$$

5.3 Vision Based Adaptive Jacobian Control of Multi-fingered Robot Hands 115

Hence, if

$$\begin{aligned} (\lambda_1 - \alpha c_0 - \frac{1}{2}b_{ex}(k_{pmax} + \alpha k_{vmax})) &> 0 \\ (\alpha k_{pmin} - \frac{1}{2}b_{ex}(k_{pmax} + \alpha k_{vmax})) &> 0, \end{aligned} \quad (5.72)$$

or

$$\min\left\{\frac{2(\bar{\lambda}_1 - \frac{\alpha c_0}{k_{vmax}})}{\bar{a} + \alpha}, \frac{2\bar{a}\alpha\lambda_2}{\bar{a} + \alpha}\right\} > b_{ex}, \quad (5.73)$$

where $\bar{a} = \frac{k_{pmax}}{k_{vmax}}$, $\bar{\lambda}_1 = \frac{\lambda_1}{k_{vmax}}$ and $\lambda_2 = \frac{k_{pmin}}{k_{pmax}}$ then $W \geq 0$.

Theorem 5.2 *The closed-loop system described by (5.60) with adaptive Jacobian matrices gives rise to the convergence of $(\Delta x, \dot{q}, f_{int})$ to $(0, 0, f_d)$ as $t \rightarrow \infty$ if the feedback gains K_p and K_v are chosen to satisfy conditions (5.33), (5.73).*

Proof: When condition (5.73) is satisfied, both V_1 and W_1 are positive definite, we have

$$\dot{V}_1 = -W_1 \leq 0, \quad (5.74)$$

where V_1 is positive definite in $s(\Delta x), \dot{q}, \dot{r}_f, \Delta W_{gj}, \Delta W_{fij}$ and ΔW_{xij} and W is a positive semidefinite function. Integrating both sides of equation (5.35), one has

$$V_1 - V_1(0) = - \int_0^t W_1(\tau) d\tau \leq 0. \quad (5.75)$$

Hence, V is bounded and one can conclude that $s(\Delta x), \dot{q}, \dot{r}_f, \Delta W_{gj}, \Delta W_{fij}$ and ΔW_{xij} are bounded vectors. One can conclude that $s(\Delta x), \dot{q}$ are square integrable functions.

5.3 Vision Based Adaptive Jacobian Control of Multi-fingered Robot Hands 116

Substituting equations (5.7), (5.8) and (5.9) into (5.60), one has

$$\begin{aligned}
 & (M(q) + \bar{M}_o)\ddot{q} + (B + \frac{1}{2}\dot{M}(q) + S(q, \dot{q}) + \frac{1}{2}\dot{\bar{M}}_o + \bar{S}_o)\dot{q} \\
 & + \hat{J}_x^T(q, \hat{W}_x)K_p s(\Delta x) + \hat{J}_x^T(q, \hat{W}_x)K_v \hat{J}_x(q, \hat{W}_x)\dot{q} \\
 & + \Delta W_g \theta_g(q) + \Delta W_f \theta_f(q) f_{int} + E_g + E_f f_{int} + K_g \text{sgn}(y) + K_f f_{int} \\
 & = \hat{J}_m^T(q, \hat{W}_f) R K_m ((f_{int} - f_d) + \gamma \int_o^t (f_{int} - f_d) d\tau). \quad (5.76)
 \end{aligned}$$

Multiply both sides of equation (5.76) by $J_m(q)\bar{M}^{-1}$ and using $J_m(q)\ddot{q} = -\dot{J}_m(q)\dot{q}$, one has

$$\begin{aligned}
 & -\dot{J}_m(q)\dot{q} + J_m(q)\bar{M}^{-1}\{(B + \frac{1}{2}\dot{M}(q) + S(q, \dot{q}) + \frac{1}{2}\dot{\bar{M}}_o + \bar{S}_o)\dot{q} \\
 & + \hat{J}_x^T(q, \hat{W}_x)K_p s(\Delta x) + \hat{J}_x^T(q, \hat{W}_x)K_v \hat{J}_x(q, \hat{W}_x)\dot{q} \\
 & + \Delta W_g \theta_g(q) + \Delta W_f \theta_f(q) f_{int} + E_g + E_f f_{int} + K_g \text{sgn}(y) + K_f f_{int}\} \\
 & = J_m(q)\bar{M}^{-1} \hat{J}_m^T(q, \hat{W}_f) R K_m ((f_{int} - f_d) + \gamma \int_o^t (f_{int} - f_d) d\tau), \quad (5.77)
 \end{aligned}$$

The force error is subject to

$$\Delta_\infty = (f_\infty - f_d) + \gamma(K_m - \bar{E}_\infty)^{-1} K_m \int_o^t (f_\infty - f_d) d\tau. \quad (5.78)$$

where

$$\begin{aligned}
 \Delta(t) &= (K_m - \bar{E})^{-1} \kappa r(t), \\
 \bar{E} &= J_m(q)\bar{M}^{-1}(\Delta W_f \theta_f(q) + E_f + K_f) \\
 \kappa &= R^T(J_m(q)\bar{M}^{-1} \hat{J}_m^T(q, \hat{W}_f))^{-1} \\
 r(t) &= -\dot{J}_m(q)\dot{q} + J_m(q)\bar{M}^{-1}\{(B + \frac{1}{2}\dot{M}(q) + S(q, \dot{q}) + \frac{1}{2}\dot{\bar{M}}_o + \bar{S}_o)\dot{q} \\
 & + \hat{J}_x^T(q, \hat{W}_x)K_p s(\Delta x) + \hat{J}_x^T(q, \hat{W}_x)K_v \hat{J}_x(q, \hat{W}_x)\dot{q} + \Delta W_g \theta_g(q) + \Delta W_f \theta_f(q) f_d \\
 & + E_g + K_g \text{sgn}(y) + E_f f_d + K_f f_d\}. \quad (5.79)
 \end{aligned}$$

Since γ is a positive constant and all terms in left hand side of equation (5.78) are bounded, one can conclude that $f_{int} - f_d$ and its integration are bounded. The boundedness of the force errors implies that $\ddot{q}$ is bounded because $s(\Delta x), \dot{q}, \dot{r}_f, \Delta W_{gj}$ and ΔW_{fij} are bounded. Hence, $s(\Delta x), \dot{q}$ are uniform continuous and it follows that [61]

$$\Delta x \rightarrow 0, \dot{q} \rightarrow 0 \text{ as } t \rightarrow \infty. \quad (5.80)$$

The maximum invariant satisfies:

$$\Delta_\infty = (f_\infty - f_d) + \gamma(K_m - \bar{E}_\infty)^{-1}K_m \int_0^t (f_\infty - f_d) d\tau. \quad (5.81)$$

where $\Delta_\infty = (K_m - \bar{E}_\infty)^{-1}\kappa_\infty J_m(q_d)\bar{M}^{-1}\{\Delta W_{f\infty}\theta_f(q_d)f_d + \Delta W_{g\infty}\theta_g(q_d) + E_g + E_f f_d\}$, $\kappa_\infty = R_\infty^T(J_m(q_d)\bar{M}^{-1}\hat{J}_m^T(q_d, \hat{W}_{f\infty}))^{-1}$, $\bar{E}_\infty = J_m(q_d)\bar{M}^{-1}(\Delta W_{f\infty}\theta_f(q_d) + E_f)$. q_d is a desired joint configuration for x_d and is defined for the purpose of analysis only. Therefore, K_m can be chosen so that the eigenvalues of $-(K_m - \bar{E}_\infty)^{-1}K_m - [(K_m - \bar{E}_\infty)^{-1}K_m]^T$ are in the left-half plane and hence $f_{int} \rightarrow f_d$ as $t \rightarrow \infty$.

5.4 Simulation Results

In this section, simulation results are presented to illustrate the performance of the proposed neural network controller for multi-fingered robots. Two planar fingers with three degrees of freedoms grasping a object as illustrated in figure 5.4 are considered. A fixed camera is placed distance away from the fingers. The position vector r of the object in Cartesian space is defined as

$$r = [r_x, r_y, \theta]^T, \quad (5.82)$$

where r_x and r_y are position variables on the x and y axis respectively and θ is the orientation angle of the object with respect to x axis, $\theta = q_1 + q_2 + q_3$. Similarly, the task-space vector is defined as

$$x = [x_1, x_2, \theta]^T, \quad (5.83)$$

where x_1 and x_2 are position variables in pixels on the two axis of image plane. The finger Jacobian matrix $J_e(q)$ mapping from joint space to Cartesian space for the finger is given by

$$J_e(q) = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} - l_3 s_{123} & -l_2 s_{12} - l_3 s_{123} & -l_3 s_{123} \\ l_1 c_1 + l_2 c_{12} + l_3 c_{123} & l_2 c_{12} + l_3 c_{123} & l_3 c_{123} \\ 1 & 1 & 1 \end{bmatrix}, \quad (5.84)$$

where $s_1 = \sin(q_1)$, $c_1 = \cos(q_1)$, $s_{12} = \sin(q_1 + q_2)$, $c_{12} = \cos(q_1 + q_2)$, $s_{123} = \sin(q_1 + q_2 + q_3)$, $c_{123} = \cos(q_1 + q_2 + q_3)$ and l_1, l_2, l_3 are the lengths of the first, second and third link, respectively. The image Jacobian matrix $J_I(r_f)$ mapping from Cartesian space to image space is given by

$$J_I = \begin{bmatrix} J_{Ip} & 0 \\ 0 & 1 \end{bmatrix} \quad (5.85)$$

where

$$J_{Ip} = \frac{f_1}{z - f_1} \begin{bmatrix} \beta_1 & 0 \\ 0 & \beta_2 \end{bmatrix} \begin{bmatrix} \cos\delta & \sin\delta \\ -\sin\delta & \cos\delta \end{bmatrix}, \quad (5.86)$$

where β_1, β_2 denote the scaling factors in pixels/m, δ is the angle of rotation of the vision coordinates relative to Cartesian coordinates. z is the perpendicular distance between the robot and the camera, f_1 is the focal length of the camera.

In the simulation, the exact masses of the three links are all set to 0.1 kg, the exact lengths l_1 , l_2 and l_3 of the links are all set to 0.05 m, f_1 is chosen as 50mm, z is

Figure 5.4: Two three-link robots grasping an object

chosen as $0.55m$ and δ chosen as 0 . $\beta_1 = \beta_2 = 10000$.

5.4.1 Vision Based Neural-Network Control of Multi-fingered Robot Hands

In this simulation, when the kinematics is uncertain, one has

$$\hat{J}_e(q) = \begin{bmatrix} -\hat{l}_1 s_1 - \hat{l}_2 s_{12} - \hat{l}_3 s_{123} & -\hat{l}_2 s_{12} - \hat{l}_3 s_{123} & -\hat{l}_3 s_{123} \\ \hat{l}_1 c_1 + \hat{l}_2 c_{12} + \hat{l}_3 c_{123} & \hat{l}_2 c_{12} + \hat{l}_3 c_{123} & \hat{l}_3 c_{123} \\ 1 & 1 & 1 \end{bmatrix}, \quad (5.87)$$

$$\hat{J}_{Ip}(\hat{r}_f) = \frac{\hat{f}_1}{\hat{z} - \hat{f}_1} \begin{bmatrix} \hat{\beta}_1 & 0 \\ 0 & \hat{\beta}_2 \end{bmatrix} \begin{bmatrix} \cos \hat{\delta} & \sin \hat{\delta} \\ -\sin \hat{\delta} & \cos \hat{\delta} \end{bmatrix}, \quad (5.88)$$

where $\hat{l}_1, \hat{l}_2, \hat{l}_3, \hat{f}_1, \hat{z}, \hat{\beta}_1, \hat{\beta}_2$ and $\hat{\delta}$ denote the estimated values of $l_1, l_2, l_3, f_1, z, \beta_1, \beta_2$ and δ , respectively.

The robot was required to move from the initial position $[100, 87]$ pixels to the desired

position $[110, 60]$ pixels, the initial and desired orientation angles are 0 and $\pi/10$. The desired force is set as $[9.511, 3.09]^T$ Newton. The estimated parameters are set as $\hat{l}_1 = 0.052m, \hat{l}_2 = 0.048m, \hat{l}_3 = 0.052m, \alpha = 4 \times 10^{-5}, K_v = 2.7 \times 10^{-4}I, K_m = I, \gamma = 0.003, \hat{f}_1 = 40mm, \hat{z} = 0.5m, \hat{\beta}_1 = \hat{\beta}_2 = 9000, \hat{\delta} = \pi/5, \bar{k}_{ij} = 0.02$, respectively. K_p is set as:

$$K_p = \begin{bmatrix} 2.9 \times 10^{-5} & 0 & 0 \\ 0 & 2.9 \times 10^{-5} & 0 \\ 0 & 0 & 520 \end{bmatrix}. \quad (5.89)$$

In this simulation, Gaussian RBF neural networks are used. The centers were chosen so that they were evenly distributed to span the input space of the network. The distance was fixed at $\pi/3$ and the number of neurons was set as 18. The gain for the networks was chosen as $L_f = 100I$. The simulation results are shown in figures 5.5 and 5.6. The results show the effectiveness of the proposed neural network controller in dealing with uncertainties in the kinematics and dynamics and convergence of the position and force errors are guaranteed.

Next, the uncertainties are increased by setting $\hat{l}_1 = 0.045m, \hat{l}_2 = 0.046m, \hat{l}_3 = 0.04m, \hat{f}_1 = 30mm, \hat{z} = 0.4m, \hat{\beta}_1 = 6000, \hat{\beta}_2 = 6000$ and $\hat{\delta} = 0$. Figures 5.7 and 5.8 shows the system response with the same control gains as used in the previous simulation. As shown by the simulation results, the image and force errors converge to 0 even in the presence of larger uncertainties in the system, however, the convergence is slower than previous simulation.

5.4.2 Vision Based Adaptive Jacobian Control of Multi-fingered Robot Hands

In the previous section, the simulation results of vision based neural network control of multi-fingered robot hands are presented. However, when the finger tips are

Figure 5.5: Position errors

Figure 5.6: Force errors

Figure 5.7: Position errors with larger uncertainties

Figure 5.8: Force errors with larger uncertainties

rolling on the surface of the objects or the finger tips are soft and deformable, the parameters and the structure of the Jacobian matrices are difficult to estimate exactly. In this simulation, unknown parameters of Jacobian matrix are considered.

In the simulation, the exact masses of the three links are all set to 0.1 kg, the exact lengths l_1 , l_2 and l_3 of the links are all set to 0.05 m, the focal length f_1 is chosen as 50mm, the perpendicular distance between the robot and the camera z , is chosen as 0.55m and the scaling factors in pixels/m are set as $\beta_1 = \beta_2 = 10000$.

The fingers were required to manipulate the object from the initial position [100, 87] pixels to the desired position [110, 60] pixels. The initial and desired orientation angles are set as 0 and $\pi/10$. The desired force is set as $[9.511, 3.09]^T$ Newton. The initial values of the adaptive Jacobian matrices are set as constant matrices. The control gains are set as $\alpha = 0.5$, $K_m = 0.3I$, $\gamma = 10$, $\bar{k}_{ij} = 0.001$, respectively. K_p and K_v are set as:

$$K_p = \begin{bmatrix} 5 \times 10^{-4} & 0 & 0 \\ 0 & 5 \times 10^{-4} & 0 \\ 0 & 0 & 5 \end{bmatrix}, K_v = \begin{bmatrix} 2 \times 10^{-4} & 0 & 0 \\ 0 & 2 \times 10^{-4} & 0 \\ 0 & 0 & 20 \end{bmatrix}. \quad (5.90)$$

In this simulation, Gaussian RBF neural networks are used. The centers were chosen so that they were evenly distributed to span the input space of the network. The distance was fixed at $\pi/15$ and the number of neurons was set as 18. The gain for the networks was chosen as $L_f = L_x = 100I$.

The simulation results are shown in figures 5.9 and 5.10. The results show the effectiveness of the proposed adaptive Jacobian controller in dealing with uncertainties in the kinematics and dynamics and convergence of the position and force errors are guaranteed.

Figure 5.9: Position errors with adaptive Jacobians

Figure 5.10: Force errors with adaptive Jacobians

5.5 Conclusion

In this chapter, the stability problem of vision based cooperative control of multi-fingered robot hands with uncertainty in kinematics and dynamics is studied. An adaptive setpoint control law using neural network has been proposed. Then an adaptive neural network Jacobian controller has been proposed to deal with uncertain structure in Jacobian matrices. New Lyapunov-like functions have also been presented for the stability analysis of the control problem. Sufficient conditions for choosing the feedback gains to guarantee the stability have been presented. It is shown that the stability can be achieved in presence of the above-mentioned uncertainties. Simulation results have been presented to illustrate the performance of the proposed control laws.

Chapter 6

Conclusion and Future Works

6.1 Conclusion

In this thesis, hybrid vision and force control problems of robots with uncertain kinematics, dynamics, camera parameters and constraint surface are studied. The control problems of multi-fingered robot with uncertain kinematics, dynamics, camera model and object parameters is also addressed. For control tasks defined in task space, traditional solutions in the literature resort to using exact kinematics or Jacobian matrix to develop the control schemes. However, when uncertainty exists in robot kinematics and environment, these control methods can not guarantee the system stability or may lead to position and force errors. This thesis is dedicated to the development of control schemes for robot that can deal with uncertainties in dynamics, kinematics, camera model, constraint surface or object parameters. Toward this objective, a systematic study is conducted in this thesis and the adaptive control schemes proposed for regulation and tracking control problems are presented. Some conclusions of this thesis are summarized as follows:

- The position and force regulation problem of robot manipulator with uncer-

tainties is formulated and solved. An approximate Jacobian controller based on regressors is first proposed. This controller does not need exact knowledge of dynamics, kinematics, camera model and constraint surface. An adaptive controller using neural networks is then proposed for robot manipulator with structured uncertainties. Compared with the previous controller, this controller do not require linear parametrization of the uncertain parameters and exact knowledge of the structure of constraint surface.

- The stability problem of visually-servoed motion and force tracking control system with uncertain kinematics, dynamics and constraint surface is studied. An adaptive Jacobian controller is firstly proposed and Lyapunov function is also presented for the stability analysis of the control systems. It is shown that the robot end-effector can track the desired motion and force trajectories in the presence of the uncertainties.
- The setpoint control problem of multi-fingered robot hands with uncertainty in kinematics and dynamics using visual feedback is studied. An adaptive setpoint control law using neural network is firstly proposed. Then an adaptive neural-network Jacobian control law is proposed which does not require knowledge of the structure of the Jacobian matrices. New Lyapunov-like functions are also presented for the stability analysis of the control problem. Sufficient conditions for choosing the feedback gains to guarantee the stability are presented. It is shown that the stability can be achieved in the presence of uncertainties.
- Simulation studies shows that the proposed controllers are effective for the position and force control of both manipulator and multi-fingered robot.

6.2 Future Research

In this section, some future research works in this thesis are presented.

- In Chapter 5, only the problem of setpoint controller of multi-fingered robot hands is considered and solved. Future research will be devoted to extending it to neural network tracking control of multi-fingered robot hands.
- The adaptive neural-network vision-force tracking controller proposed in this thesis can guarantee uniform ultimate boundedness of vision and force tracking errors. Future work will be devoted to extending the study to achieve convergence of the tracking errors.
- In this thesis, only simulation results have been presented to show the effectiveness of the proposed controllers. It is important to experimentally verify the effectiveness of the proposed controllers in future.
- There are some standard drawbacks for conventional rigid link robots, such as high power-consumption, actuators with high capacity, and low payload ratio. In order to overcome these problems, the links of the robot are made lighter. In some applications, such as space manipulators, the links are made of lightweight materials and a long reach is also required. Therefore, the structural flexibility can no longer be ignored. In these applications, the link flexibility of robot manipulator presents another source of uncertainty in both kinematics and dynamics. To consider the link flexibility together with other kinematic and dynamic uncertainties discussed in this thesis in the control design presents another interesting and challenging research topic.
- Mobile manipulator systems have been used for various outdoor applications. The base mobility greatly increases the robot workspace, and enables better

positioning of the manipulator for efficient task execution. However, the holonomic or nonholonomic kinematics constraints of the mobile base make the the control problem of mobile manipulator very challenging. In addition, the complex physical structure, the highly coupled dynamics between the mobile base and the mounted manipulator, and the mobility of the wheeled mobile base are some of the factors that greatly increase the difficulty of system design and control. Therefore, adaptive vision and force control of mobile manipulator systems is one of the interesting research topic.

- Rapid advances in sensing, computing and communication technologies have led to development of multi-robot systems. Multiple robots can improve efficiency in many tasks, such as moving a bulky item, area searching and space station assembling, which are difficult or impossible for a single robot. In many applications of multi robots systems, maintaining some spatial pattern is more efficient in area coverage and energy saving. In a tightly coupled task where multiple robots are used to handle a common object, the overall kinematics and dynamics of the multi-robot systems are uncertain when there are unexpected changes in task requirements, for example, by deploying new robots or removing faulty robots. The research in this thesis would be extended to cooperative control and formation control of multi-robot systems with unknown number of robots.

Author's Publications

1. Y. Zhao and C. C. Cheah, "Vision-based neural network control for constrained robots with constraint uncertainty," submitted for publication, 2007.
2. C. C. Cheah, Y. Zhao and J. J. E. Slotine, "Adaptive jacobian motion and force tracking control for constrained robots with uncertainties," submitted to *Automatica*, 2007.
3. Y. Zhao and C. C. Cheah, "Vision-force cooperative control of multi-fingered robot hands using neural networks," submitted to *IEEE Transactions on Neural Networks*, 2007.
4. Y. Zhao, C. C. Cheah and J. J. E. Slotine, "Adaptive Vision and Force Tracking Control of Constrained Robots with Structural Uncertainties," *2006 IEEE International Conference on Robotics and Automation*, 2007 Page(s): 2349 - 2354.
5. Y. Zhao and C. C. Cheah, "Vision-based control of multi-fingered robot hands using neural networks," *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006 Page(s):910 - 915.
6. Y. Zhao, C. C. Cheah and J. J. E. Slotine, "Adaptive vision and force tracking control for constrained robots," *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006 Page(s):1484 - 1489.
7. Y. Zhao and C. C. Cheah, "Hybrid vision-force control for robot manipulators using neural networks," *The Ninth International Conference on Control, Automation, Robotics and Vision (ICARCV 2006)*, 2006 Page(s):594 - 599.
8. C. C. Cheah, Y. Zhao and J. J. E. Slotine, "Adaptive jacobian motion and force tracking control for constrained robots with uncertainties," *IEEE International Conference on Robotics and Automation*, 2006 Page(s):2226 - 2231.

9. Y. Zhao and C. C. Cheah, "Position and force control of robot manipulators using neural networks," *IEEE Conference on Robotics, Automation and Mechatronics*, Volume 1, Dec. 2004 Page(s):300 - 305.
10. C. C. Cheah and Y. Zhao, "Inverse jacobian regulator for robot manipulator: theory and experiment," *43rd IEEE Conference on Decision and Control*, 2004, Volume 2, Dec. 2004 Page(s):1252 - 1257.
11. Y. Zhao and C. C. Cheah, "Hybrid vision-force control for robot with uncertainties," *IEEE International Conference on Robotics and Automation*, Volume 1, 2004 Page(s):261 - 266.

Bibliography

- [1] P. K. Allen, A. T. Miller, P. Y. Oh, and B. S. Leibowitz. Integration of vision and force sensors for grasping. In *Proc. of IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 349–356, 1996.
- [2] S. Arimoto. *Control Theory of Nonlinear Mechanical Systems- A Passivity-Based and Circuit-Theoretic Approach*. Clarendon Press, 1996.
- [3] S. Arimoto, H. Y. Liu, and T. Naniwa. Model-based adaptive hybrid control for geometrically constrained robots. In *Proc. of IEEE Conference on Robotics and Automation*, pages 618–623, 1993.
- [4] S. Arimoto, T. Naniwa, and Y. H. Liu. Model-based adaptive hybrid control for manipulators with geometric endpoint constraint. *Advanced Robotics*, 9(1):67–80, 1995.
- [5] S. Arimoto, T. Naniwa, and T. Tsubouchi. Principle of orthogonalization for hybrid control of robot manipulators. In *Proc. of IMACS'92*, pages 1293–1300, 1992.
- [6] S. Arimoto, P. T. A. Nguyen, H. Han, and Z. Doulgeri. Dynamics and control of a set of dual fingers with soft tips. *Robotica*, 18(1):71–80, 2000.
- [7] A. Bicchi. Hands for dexterous manipulation and robust grasping: a difficult road toward simplicity. *IEEE Transactions on Robotics and Automation*, 16(6):652–662, 2000.
- [8] R. Carelli and R. Kelly. An adaptive impedance/force controller for robot manipulators. *IEEE Transactions on Automatic Control*, 36(8):967–971, 1991.
- [9] A. Castano and S. Hutchinson. Visual compliance: task-directed visual servo control. *IEEE Transactions on Robotics and Automation*, 10(3):334–342, 1994.

- [10] C. C. Cheah, H. Y. Han, S. Kawamura, and S. Arimoto. Grasping and position control for multi-fingered robot hands with uncertain jacobian matrices. In *Proc. of IEEE International Conference on Robotics and Automation*, pages 2403–2408, 1998.
- [11] C. C. Cheah, S. Kawamura, and S. Arimoto. Hybrid position and force control for robotic manipulator with a class of constraint uncertainty. In *Proc. of USA/Japan Symposium on Flexible Automation*, pages 501–507, Otsu, Japan, 1998.
- [12] C. C. Cheah, S. Kawamura, and S. Arimoto. Stability of hybrid position and force control for robotic manipulator with kinematics and dynamics uncertainties. *Automatica*, 39:847–855, 2003.
- [13] C. C. Cheah, C. Liu, and J. J. E. Slotine. Adaptive tracking control for robots with unknown kinematic and dynamic properties. *International Journal of Robotic Research*, 25(3):283–296, 2006.
- [14] P. I. Corke and M. C. Good. Dynamic effects in visual closed-loop systems. *IEEE Transactions on Robotics and Automation*, 12(5):671–683, 1996.
- [15] E. C. Dean-Leon, L. G. Garcia-Valdovinos, V. Parra-Vega, and A. Espinosa-Romero. Visual servoing for constrained robots: a new complete theoretical framework and its experimental validation. In *Proc. of IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 1145–1452, 2005.
- [16] Z. Doulgeri and S. Arimoto. A position/force control for a robot finger with soft tip and uncertain kinematics. *Journal of Robotics Systems*, 19(3):115–131, 2002.
- [17] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3):313–326, 1992.
- [18] T. Fischer, D. Rapela, and H. Woern. Joint controller for the object-pose controlling on multifinger grippers. In *Proc. of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 422–427, 1999.
- [19] S. S. Ge and C. C. Hang. Structural network modeling and control of rigid body robots. *IEEE Transactions on Robotics and Automation*, 14(5):823–827, 1998.

- [20] 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. *International Journal of Systems Science*, 32(1):9–23, 2001.
- [21] S. S. Ge, T. H. Lee, and C. J. Harris. *Adaptive Neural Network Control of Robotic Manipulators*. World Scientific Publishing, 1998.
- [22] E. Grosso, G. Metta, A. Oddera, and G. Sandini. Robust visual servoing in 3-d reaching tasks. *IEEE Transactions on Robotics and Automation*, 12(5):732–742, 1996.
- [23] K. Guglielmo and N. Sadegh. Implementating a hybrid learning force control scheme. *IEEE Control Systems Magazine*, 14(1):72–79, 1994.
- [24] K. Hashimoto, T. Ebine, and H. Kimura. Visual servoing with hand-eye manipulator-optimal control approach. *IEEE Transactions on Robotics and Automation*, 12(5):766–774, 1996.
- [25] S. Haykin. *Neural Networks- A Comprehensive Foundation*. Prentice-Hall International, 1999.
- [26] N. Hogan. Impedance control: An approach to manipulation: Part I-theory; part II-implementation; part III-applications. *ASME Journal of Dynamic Systems, Measurement, and Control*, 107:1–24, 1985.
- [27] K. Hosoda and M. Asada. Versatile visual servoing without knowledge of true jacobian. In *Proc. of IEEE Conference on Intelligent Robots and Systems*, pages 186–193, 1994.
- [28] K. Hosoda, K. Igarashi, and M. Asada. Adaptive hybrid for visual and force servoing in an unknown environment. *IEEE Robotics and Automation Magazine*, December:39–43, 1998.
- [29] S. Hutchinson, G. D. Hager, and P. I. Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–670, 1996.
- [30] S. Jung and T. C. Hsia. Robust neural force control scheme under uncertainties in robot dynamics and unknown environment. *IEEE Transactions on Industrial Electronics*, 47(2):403–412, 2000.
- [31] R. Kelly, R. Carelli, O. Nasisi, B. Kuchen, and F. Reyes. Stable visual servoing of camera-in-hand robotic systems. *IEEE/ASME Transactions on Mechatronics*, 5(1):39–48, 2000.

- [32] R. Kelly, J. Favela, J. M. Ibarra, and D. Bassi. Asymptotically stable visual servoing of manipulators via neural networks. *Journal of Robotic Systems*, 17(12):659–669, 2000.
- [33] Y. H. Kim and F. L. Lewis. Neural network output feedback control of robot manipulator. *IEEE Transactions on Robotics and Automation*, 15(2):301–309, 1999.
- [34] K. Kleinmann and R. Wacker. On a selftuning decoupling controller for the joint control of a tendon driven multifingered robot gripper. In *Proc. of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*, pages 209–215, 1994.
- [35] C. Kwan, F. L. Lewis, and D. M. Dawson. Robust neural-network control of rigid-link electrically driven robots. *IEEE Transactions on Neural Networks*, 9(4):581–588, 1998.
- [36] F. L. Lewis. Neural network control of robot manipulators. *Intelligent Systems and Their Applications*, 11(3):64–75, 1996.
- [37] Z. Li, P. Hsu, and S. S. Sastry. Grasping and coordinated manipulation by a multifingered robot hand. *International Journal of Robotics Research*, 18(4):33–50, 1989.
- [38] E. Malis and F. Chaumette. Theoretical improvements in the stability analysis of a new class of model-free visual servoing methods. *IEEE Transactions on Robotics and Automation*, 18(2):176–186, 2002.
- [39] E. Malis, F. Chaumette, and S. Boudet. 2-1/2-d visual servoing. *IEEE Transactions on Robotics and Automation*, 15(2):238–250, 1999.
- [40] N. H. McClamroch and D. Wang. Feedback stabilization and tracking of constrained robots. *IEEE Transactions on Automatic Control*, 33(5):419–426, 1988.
- [41] J. K. Mills and A. A. Goldenberg. Force and position control of manipulators during constrained motion tasks. *IEEE Transactions on Robotics and Automation*, 5(1):30–46, 1989.
- [42] K. Nagai and T. Yoshikawa. Dynamic manipulation/grasping control of multifingered robot hands. In *Proc. of International Conference on Robotics and Automation*, pages 1027–1033, 1993.

- [43] Y. Nakamura. *Advanced Robotics-Redundancy and Optimization*. Addison-Wesley Publishing Company, 1991.
- [44] M. Namvar and F. Aghili. Adaptive force-motion control of coordinated robots interacting with geometrically unknown environments. *IEEE Transactions on Robotics*, 21(4):678–694, 2005.
- [45] T. Naniwa, S. Arimoto, and V. P. Vega. A model-based adaptive control scheme for coordinated control of multiple manipulators. In *Proc. of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*, pages 695–702, 1994.
- [46] B. J. Nelson and P. K. Khosla. Force and vision resolvability for assimilating disparate sensory feedback. *IEEE Transactions on Robotics and Automation*, 12(5):714–731, 1996.
- [47] B. J. Nelson, J. D. Morrow, and P. K. Khosla. Improved force control through visual servoing. In *Proc. of the American Control Conference*, pages 380–386, 1995.
- [48] G. Niemeyer and J. J. E. Slotine. Performance in adaptive manipulator control. *International Journal Robotics Research*, 10(2):149–161, 1991.
- [49] N. P. Papanikolopoulos and P. K. Khosla. Adaptive robot visual tracking: Theory and experiments. *IEEE Transactions on Automatic Control*, 38(3):429–445, 1993.
- [50] N. P. Papanikolopoulos, P. K. Khosla, and T. Kanade. Visual tracking of a moving target by a camera mounted on a robot: A combination of vision and control. *IEEE Transactions on Robotics and Automation*, 9(1):14–35, 1993.
- [51] V. Parra-Vega and S. Arimoto. A passivity-based adaptive sliding mode position-force control for robot manipulators. *International Journal of Adaptive Control and Signal Processing*, 10(4-5):365–377, 1996.
- [52] H. D. Patino, R. Carelli, and B. R. Kuchen. Neural networks for advanced control of robot manipulators. *IEEE Transactions on Neural Networks*, 13(2):343–354, 2002.
- [53] A. Pichler and M. Jagerstand. Uncalibrated hybrid force-vision manipulation. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1866–1871, 2000.

- [54] M. H. Raibert and J. J. Craig. Hybrid position/force control of manipulators. *ASME Journal of Dynamic Systems, Measurement, and Control*, 102:126–133, 1981.
- [55] E. Rossignol and J. L. Henrotte. Kinematical and dynamical control of robotic manipulators. In *Proc. of IEEE Conference on Systems, Man and Cybernetics*, pages 459–464, 1993.
- [56] J. K. Salisbury. Active stiffness control of a manipulator in cartesian coordinates. In *Proc. IEEE Conference on Decision and Control*, pages 95–100, 1980.
- [57] R. M. Sanner and J. E. Slotine. Gaussian networks for direct adaptive control. *IEEE Transactions on Neural Networks*, 3(6):837–863, 1992.
- [58] R. M. Sanner and J. E. Slotine. Structurally dynamic wavelet networks for adaptive control of robotic systems. *International Journal of Control*, 70(3):405–421, 1998.
- [59] S. Sastry and M. Bodson. *Adaptive Control: Stability, Convergence, and Robustness*. Prentice-Hall International, 1989.
- [60] B. Siciliano and L. Villani. *Robot Force Control*. Kluwer Academic Publishers, 1999.
- [61] J. J. E. Slotine and W. Li. *Applied Nonlinear Control*. Prentice-Hall International, 1991.
- [62] M. W. Walker, D. Kim, and J. Dionise. Adaptive coordinated motion control of two manipulator arms. In *Proc. of International Conference on Robotics and Automation*, pages 1084–1090, 1989.
- [63] D. Wang and N. H. McClamroch. Position/force control design for constrained mechanical systems: Lyapunov's direct method. *IEEE Journal on Robotics and Automation*, 9(3):308–313, 1993.
- [64] D. Wang, Y. C. Soh, Y. K. Ho, and P. C. Muller. Global stabilization for constrained robot motions with constraint uncertainties. *Robotica*, 16:171–179, 1998.
- [65] L. E. Weiss, A. C. Sanderson, and C. P. Neuman. Dynamic sensor-based control of robots with visual feedback. *IEEE Journal on Robotics and Automation*, 3(5):404–417, 1987.

- [66] L. L. Whitcomb, S. Arimoto, T. Naniwa, and F. Ozaki. Adaptive model-based hybrid control of geometrically constrained robot arms. *IEEE Transactions on Robotics and Automation*, 13(1):105–116, 1997.
- [67] D. E. Whitney. Force feedback control of manipulator fine motions. *ASME Journal of Dynamic Systems, Measurement, and Control*, 99(2):91–97, 1977.
- [68] W. J. Wilson, C. C. Williams Hulls, and G. S. Bell. Relative end-effector control using cartesian position based visual servoing. *IEEE Transactions on Robotics and Automation*, 12(5):684–696, 1996.
- [69] D. Xiao, B. K. Ghosh, N. Xi, and T. J. Tarn. Intelligent robotic manipulation with hybrid position/force control in an uncalibrated workspace. In *Proc. of IEEE International Conference on Robotics and Automation*, pages 1671–1676, Leuven, Belgium, 1998.
- [70] D. Xiao, B. K. Ghosh, N. Xi, and T. J. Tarn. Sensor-based hybrid position/force control of a robot manipulator in an uncalibrated environment. *IEEE Transactions on Control Systems Technology*, 8(4):635–645, 2000.
- [71] B. Yao. Unified approach to robust control of robot manipulator using variable structure model-reaching control strategy. Master's thesis, School of Electrical and Electronic Engineering, Nanyang Technological University, 1992.
- [72] B. Yao and M. Tomizuka. Adaptive coordinated control of multiple manipulators handling a constrained object. In *Proc. of IEEE International Conference on Robotics and Automation*, pages 624–629, 1993.
- [73] B. Yao and M. Tomizuka. Adaptive robust motion and force tracking control of robot manipulators in contact with compliant surfaces with unknown stiffness. *ASME J. of Dynamic System, Measurement and Control*, 120(2):232–240, 1998.
- [74] Y. Yokokohji, M. Sakamoto, and T. Yoshikawa. Vision-aided object manipulation by a multifingered hand with soft fingertips. In *Proc. of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 3201–3208, 1994.
- [75] T. Yoshikawa. *Foundations of Robotics: Analysis and Control*. MIT Press, 1990.
- [76] T. Yoshikawa and K. Nagai. Manipulating and grasping forces in manipulation by multifingered robot hands. *IEEE Transactions on Robotics and Automation*, 7(1):67–77, 1991.

- [77] T. Yoshikawa and A. Sudou. Dynamic hybrid position/force control of robot manipulators: online estimation of unknown constraint. *IEEE Transactions on Robotics and Automation*, 9(2):220–226, 1993.
- [78] T. Yoshikawa, T. Sugie, and M. Tanaka. Dynamic hybrid position/force control of robot manipulators: Controller design and experiment. *IEEE Journal on Robotics and Automation*, 4(6):699–705, 1988.
- [79] B. H. Yoshimi and P. K. Allen. Visual control of grasping and manipulation tasks. In *Proc. of International Conference on Robotics and Automation*, pages 575–582, 1999.
- [80] Y. Zhao and C. C. Cheah. Hybrid vision-force control for robot with uncertainties. In *Proc. of IEEE International Conference on Robotics and Automation*, pages 261–266, 2004.
- [81] X. Z. Zheng, R. Nakashima, and T. Yoshikawa. On dynamic control of finger sliding and object motion in manipulation with multifingered hands. *IEEE Transactions on Robotics and Automation*, 16(5):469–481, 2000.