INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

University of Alberta

TACTILE SENSING AND DIRECT TOUCH-DRIVEN ROBOT CONTROL

by

Ning Chen 

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**.

Department of Electrical Engineering

Edmonton, Alberta

Spring 1997

University of Alberta

Library Release Form

Name of Author: Ning Chen

Title of Thesis: Tactile Sensing and Direct Touch-Driven Robot Control

Degree: Doctor of Philosophy

Year this Degree Granted: 1997

. . . . . .

Ning Chen

302, 3083 - E. Kent Avenue
Vancouver, B.C.
Canada V5S 4R2

Date: Oct. 12th, 1996

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Tactile Sensing and Direct Touch-Driven Robot Control** submitted by Ning Chen in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**.

Dr. H. Zhang, Supervisor, Computing Science

Dr. R. E. Rink, Supervisor, Electrical Engineering

Dr. R. D. Howe, External Examiner, Harvard University

Dr. Z. J. Koles, Biomedical Engineering

Dr. Q. M. Meng, Electrical Engineering

Dr. R. W. Toogood, Mechanical Engineering

Dr. J. F. Vaneldik, Electrical Engineering

Dr. P. J. Wojcik, Alberta Research Council

Date: Oct. 15, 1996

To my parents.

Abstract

Robotic tactile sensing has received much attention in the past decade, however the study on tactile sensing is still in its infancy in comparison with studies on many other sensing modalities such as computer vision and force/torque sensing. This thesis is devoted to the fundamental research on tactile data processing and the effective use of tactile information in low-level robot control. In the first part of the thesis, analytical inverse models for point and line contact are derived. A kinematics-based local object shape sensing algorithm is also proposed. The framework for direct touch-driven robot control is presented in the second part of the thesis, under which two basic control schemes are abstracted and verified with extensive experiments.

Acknowledgements

I would like to express my thankful appreciation to my supervisors Dr. Hong Zhang and Dr. Ray Rink. Without their direction, this thesis could not have been completed. Special thanks are also due to Steve Sutphen for providing technical support through out the thesis work. I would like to thank Dr. Koles, Dr. Meng, Dr. Toogood, Dr. Vaneldik and Dr. Wojcik for their comments and suggestions during the thesis review.

Contents

List of Figures

List of Tables

List of Symbols

P	*Normal contact force*
Q	*Shear contact force*
σ	*Normal stress component*
τ	*Shear stress component*
M	*Moment of a tactile image*
S	*Surface matrix*
n	*Surface normal*
T	*Homogeneous transformation* (4×4)
R	*Rotation transformation* (3×3)
F	*Tactile feature vector*
J_θ	***Robot Jacobian matrix***
J_t	***Tactile feature Jacobian matrix***

Chapter 1

Introduction

Tactile sensing has long been recognized as an important sensation for both humans and intelligent robots; however, research on tactile sensing is still in its infancy in comparison with many other sensing modalities in robotics such as force/torque sensing and computer vision. To develop intelligent robots with tactile sensation, the first step is to design tactile sensors capable of providing functionalities similar to human touch. This has represented the majority of the work within robot tactile sensing [29]. To date various tactile sensors employing different physical principles have been designed, mostly in research laboratories. Several commercial tactile sensors are also already available.

With a tactile sensor in hand, the problem to understand and further to effectively use tactile sensing in object recognition and robot control is still not trivial, just as the invention of the video camera did not mean that we knew how to make use of visual images. In fact it is only in recent years that image processing and vision-based robot control have been fully studied and successfully applied to industrial applications. Robotic tactile sensing and the use of tactile information in performing control tasks, on the other hand, have not been studied until recent years. Many theoretical and practical issues are still open in this emerging yet challenging area.

This thesis is devoted to an investigation of the fundamental problems of tactile data processing and the application of tactile sensing to low-level robot control.

Specifically, the objectives were to model tactile sensors in order to understand the tactile sensing processes and to incorporate tactile feedback into a robot servo controller in order to accomplish more complex manipulation and exploration tasks with the assistance of tactile sensing. The problems studied in this thesis are in some sense parallel to visual image processing and visual servoing in vision-based control [22][56]. However, one must keep in mind that, although tactile sensing bears a lot of similarities to computer vision, they are fundamentally different sensing modalities. Incorporation of tactile sensing into robotics applications (either in recognition or manipulation) raises many theoretical and practical questions that are unique to tactile sensing.

In this introduction we will first briefly review human tactile sensation and tactile perception behaviors, because human sensory systems are in many ways the direct inspiration of the development of robotic sensing systems. This brief review does not attempt to provide a complete picture for the human tactile sensing system and human touch-related behaviors. Only important concepts and insights directly stimulating and guiding the studies on robot tactile sensing and touch-based robot control in this thesis are selected and presented. After reviewing human tactile sensation and tactile perception, definitions of robotic tactile sensing and touch-based robot control are discussed. Finally, an overview of the thesis is given and major contributions are outlined.

1.1 A brief review of human tactile sensation and tactile perception behaviors

The human tactile sensing system is a highly developed system that is capable of dealing with dynamic and uncertain environments. Studies in physiology indicate that the human tactile sensing system consists of functionally different tactile sensors. These sensors are usually distributed at different locations and depth under the skin. For example, in the smooth skin of a human hand, there are four types

of specialized mechanoreceptors (or tactile sensors), Merkel discs, Meissner corpuscles, Pacinian corpuscles, and Ruffini organs [31]. Merkel discs respond to static and slowly changing impressions; Meissner corpuscles respond only to changing stimuli; Pacinian corpuscles are sensitive to vibrations; Ruffini organs are sensitive to static and slowly changing skin stretch. While Merkel discs and Meissner corpuscles have smaller receptive fields, normally 3-5 mm in diameter, Pacinian corpuscles and Ruffini organs have larger receptive fields of more than 10 mm in diameter.

Psychophysical studies revealed that human touch sensation actually consists of two different senses; cutaneous sensing, which responds to constant stimuli through skin, and kinesthetic sensing, which monitors the position and motion of limbs and force applied by muscles [36] [48]. The simultaneous use of cutaneous and kinesthetic sensing to gain information about the environment is referred to as haptic perception in psychophysics. It has been demonstrated that haptic sensing is fast and effective in recognizing common 3-D objects [34].

Another interesting observation on human tactile sensing is that tactile sensing usually deals with small areas of contact; therefore motion of the body is usually necessary in tactile perception of larger objects [54]. There are also studies indicating that human tactile sensing is more sensitive to motion signals.

1.2 Robot tactile sensing and touch-based robot control

There have been various definitions for robot tactile sensing in the past. Gindy defined tactile sensing as "the continuous monitoring of force in an array"[24]. Harmon referred to tactile sensing as "the graded sensing of contact force in an array of points" [26]. Nicholls and Lee defined a tactile sensor to be "a device which measures parameters of contact interaction between the device and some physical stimuli" [42]. Howe and Cutkosky considered tactile sensing as "the use of robot-mounted sensors to derive information from contact between the robot and its environment" [29]. It seems

that, while earlier researchers tended to consider robot tactile sensor as a distributed force sensing device, more recent studies attempted to extend the functionality of tactile sensing to more general contact sensing.

In this thesis, we adopt the definition by Howe and Cutkosky and focus on tactile sensing that measures *distributed* contact quantities over a small area. We emphasize that tactile sensing is to *derive* information from raw tactile stimuli, and a tactile sensing device is a device capable of capturing signals of tactile stimuli, such as force, texture, object geometry, heat and vibration, etc; the study of tactile sensor design is different from that of tactile sensing although they were often mixed before.

In the past, simple tactile data processing algorithms were usually provided for special sensors by the designer of the sensors. Only a very limited number of reports have been published on the basic research of tactile data processing. Until today, the development of tactile data processing algorithms still seems to depend on either intuition or direct transplant from its counterpart, visual image processing. On the other hand, although research in the field of tactile sensing has scarcely come out of its infancy, we believe it has become necessary and also possible to study tactile sensing in a more general manner. We also believe that it is necessary and possible to further apply the available tactile sensing technologies to higher level control tasks, although most of the existing tactile sensing technologies are still far from producing robust and accurate tactile sensors from a practical application point of view.

In this thesis, we investigate tactile sensing in a systematic manner, and we study tactile sensing at the image processing level (also referred to as tactile image processing or tactile data processing in the thesis). We also explore the direct touch-driven control problem, or the so called "tactile-servo" problem, which is to give a robot the ability to adjust its position and orientation according to feedback from tactile sensing. Direct touch-driven control requires the integration of tactile sensing with low level robot control, which can be considered as providing a robot with the "reflexive

response" [1] to tactile stimuli. Direct touch-driven control aims at establishing a basic relationship between tactile sensing and robot motion. Such a basic relationship would allow the development of higher-level physical and functional organizations. It is a basic functionality that can be used by an intelligent robot to execute more sophisticated actions toward autonomous perception of its environment. Direct touch-driven control is an important area of fundamental research for not only general robot manipulation but also for areas such as teleoperations and rehabilitation engineering, where sensitive and accurate artificial limbs are desired.

1.3 An overview of the thesis and its contributions

The contents of the thesis can be divided into two parts. In the first part, we tackle the problem of tactile sensing, or tactile data processing. In the second part we explore the integration of tactile sensing with robot servo control. These two steps are closely related. Modelling is the basis for touch-based control. Touch-driven control, on the other hand, is necessary for tactile sensing, as will be discussed later in the thesis.

From the physiology point of view, there are many physical quantities associated with tactile sensing. However, not all of them are of great interest to robotic applications. For fine robot tasks, such as precise fingertip manipulation where tactile sensing is most needed, contact force, location and local object shape are among the most useful information to be obtained from an inverse model [17], a mathematical model with which to interpret tactile stimuli from the raw tactile sensor readings. Inverse modelling of tactile sensing constitutes the topics of the first part of the thesis. Recovering contact location and forces is discussed in chapter 2 and 3, for the cases of point and line contact, respectively; local contact shape sensing is studied in chapter 4. In chapter 5 and 6 we propose two different schemes to perform direct touch-driven robot control, or tactile-servo. Extensive experimental results are provided to support

[1] *Reflexive response* is a low-level spinal-coordination response, rather than a higher-level *cortical response*.

the proposed control schemes. Finally the summary of the thesis is given in chapter 7.

The major contributions of the thesis are:

1. Previous studies on inverse tactile modelling were mostly not model-based, with the notable exception of a study by Fearing [19], where analytical inverse modelling was studied for a cylindrical tactile sensor. In this thesis, analytical inverse models for line and point contact are designed using moment analysis. These models have proven to be simple and effective for real-time robot control applications.

2. An algorithm to recover local object shape information from tactile sensing and robot kinematics is presented. Instead of attempting to resolve position, force, and geometric (i.e., shape) information from a single solid mechanics model, we propose to solve position (and force) information first from a simplified solid mechanics model, and recover geometric information using the contact information obtained in the first stage together with the contact and robot kinematic constraints. This kinematics-based approach is novel. The surface matrix concept introduce in deriving the algorithm can be used as a convenient tool to analyze other robot contact kinematics problems.

3. A framework for direct touch-driven robot control is presented. It is an abstraction and extension of earlier concepts [49] [51]. Two basic control schemes, using an inverse tactile model and using a tactile Jacobian, are proposed and verified with extensive experiments. The basic touch-driven control schemes provide a foundation for the execution of more complex motion procedures such as grasping and precise manipulation.

Chapter 2

Tactile sensing of 3-D frictional point contact: contact force and location sensing

Contact location and force are among the most useful parameters for grasping and dextrous manipulation. This chapter presents the theoretical results of extracting these parameters from a tactile sensor for the case of point contact. Using the model of a 3-D frictional point force acting on an elastic half-space, moment analysis is employed to interpret a tactile image. Analytical relationship between the first three moments of a tactile image and contact location and force (both the magnitude and direction) is established, for the cases of single-layer and two-layer one-dimensional tactile sensors and a single layer multi-dimensional tactile sensor.

2.1 Introduction

Touch provides rich information about the contact between a robot and its environment. Tactile information is important in many situations such as grasping and dextrous manipulation where the tactile sensor is often the only sensor modality available to monitor the process of a task involving contact.

There are many kinds of tactile sensors designed for robotic applications, among which a tactile array sensor that senses distributed contact stress or strain is most commonly used. The array sensing approach has the advantage of efficiently determining local contact details, such as contact shape, in comparison with the concentrated-sensing approaches.

Typically a tactile array sensor is constructed with an array of normal stress or strain transducers, and is covered by a layer of rubber-like soft "skin", which protects the transducers, improves grasp stability, and facilitates contact force control [15]. Unfortunately, the compliant layer also filters the contact force and thus complicates the task of interpreting the pressure values measured by the transducers. In other words, because of the "blurring" effect, the measurement by a particular tactile element (tactel) is determined not only by the forces acting on the point directly above it, but also by the forces at other points on the entire compliant layer. Because of such a characteristic, mathematical modelling of tactile sensing has been, to some extent, one of the well known difficulties preventing tactile sensing from being applied in robotics.

Various attempts have been made to understand the interaction between tactile readings and the unknown objects. Such a process involves two distinct steps, forward modeling and inverse modeling. In the first step, given contact force and location, a *forward model* is established to calculate the strain or stress distribution at a certain depth of the covering elastic layer where the tactile array sensor is embedded. The forward model provides guidance for sensor design and has been solved by analytical

approaches based on various assumptions [20][53][32], and by numerical methods such as the finite element method [45][50].

In the second step of the analysis of tactile sensing, an *inverse model* needs to be established to relate the strain/stress distribution to the contact variables causing the distribution. A simple and closed-form solution to the tactile sensor modelling problem is highly desirable, not only because such a solution facilitates easy implementation of inversion algorithms in real-time robot control applications, but also because it will shed light on and provide insight into the basic principles of tactile sensing. The objective of our modelling study is to provide efficient *inverse* models which can be further applied to higher level real-time robot control or recognition applications.

The contact type to be studied in this chapter, point contact, is one of the most important contact types. Many high-level studies rely on such a contact model, for example, in grasping. A grasp is usually described as the combination of individual contacts, which in turn are modelled as the combination of a few primitive contacts, hard or soft point contacts with or without friction [47]. Also, point contact is a fundamental step toward the study of tactile sensing with more complicated geometries.

The rest of this chapter is organized as follows. In section 2.2, previous research is reviewed. In section 2.3, the point contact problem is defined and notations introduced. Moments of various orders of a tactile image are derived for different sensing structures. The basic equations that compute contact variables from the moments are presented. An iterative algorithm is presented in section 2.4 in order to improve the modelling accuracy. Finally, experimental results and the conclusions are given in 2.5 and 2.6, respectively.

2.2 Previous research

The approach taken in this chapter is to use basic elasticity theory to model the tactile sensing processes[14]. Solid mechanics was first used by Fearing and Hollerbach [20]

to model tactile sensing systems. In this study, using solid mechanics results for the 2-D case of an infinite long line load acting on an elastic half-space, equations relating the location, angle and magnitude of a 2-D line force to the subsurface strain measurements were derived. No analytical inverse algorithms, however, were given in this study. In a later study, the inverse modelling problem was further investigated for a cylindrical tactile sensor using "inverse linear filtering"[18], but the main focus of this study was on sensor design and the efficiency of the algorithm was a concern. In a 3-D strain/stress field study, moments were used to characterize the tactile image[53]. Given the force and location of contact, the first three moments of stress and strain at the subsurface were numerically calculated in a forward manner. Through the numerical results of a sphere indenting a planar surface, a linear relationship was observed between the translation of center of the tactile image and the ratio of the normal to shear forces. However, neither analytical investigation nor experimentation was further pursued. In addition to these studies, there were also reports on using numerical methods to solve the inverse modelling problem, such as that by De Rossi *et al* where contact sensing of a second order surface was discussed [16]. Neural networks were also used to resolve tactile parameters from tactile data, such as in work by Pati *et al* [44]. None of the these approaches, however, are efficient enough for most real-time applications.

In this chapter, we establish the analytical relationships among the first three moments of a tactile image, contact location, and three components of the contact force, for the case of a 3-D frictional point contact on a planar tactile sensor. Closed-form inverse models resolving contact parameters from moments (features) of a tactile image are provided. It is shown that the first three moments contain sufficient information to compute contact location and force, and simple algorithms exist to retrieve these parameters from a tactile image.

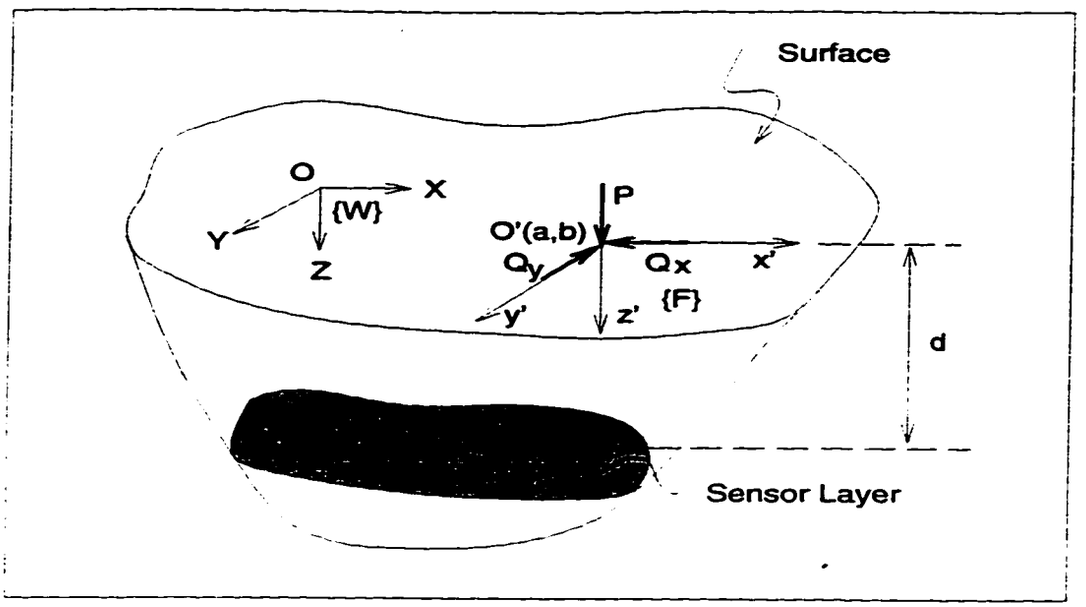


Figure 2.1: *3-D frictional point contact model*

2.3 Problem definition and the proposed solution

2.3.1 Notations and problem definition

The frictional point contact model

Consider an arbitrary shaped object contacting a planar tactile sensor. A tactile array sensor is mounted in the subsurface, at depth d. According to St. Venant's Principle in the theory of elasticity [55], one may approximate the contact as that of a frictional point load acting on an planar elastic "skin," under the condition that the contact area is relatively small. We further approximate the skin layer as an elastic half-space, as is done in many contact mechanics problems.

Given the stress measurements from the sensor, five parameters are to be determined from tactile sensing, two that define the location of the point of contact $O'(a, b, z = 0)$ in the world reference frame $\{W\}$, and three others, namely P, Q_x and Q_y, which correspond to the three components of the force vector acting at $O'(a, b, 0)$ along the three axes of force frame $\{F\}$. Note that $\{W\}$ and $\{F\}$ share the same orientation.

	Zeroth	First		Second		
Notation	M_0	M_x	M_y	M_{xx}	M_{yy}	M_{xy}
p	0	1	0	2	0	1
q	0	0	1	0	2	1

Table 2.1: *The definition of moments*

Moments of tactile image

In this study moments are used as "features" to characterize tactile data (tactile image) which are discrete spatial sample of the stress distribution over the effective sensor area. Features, or moments of the tactile image, are then used as the input to the inverse models to calculate the contact parameters.

Generally moments with respect to the axes of a Cartesian coordinate frame can be defined as

$$M_{x^p y^q} = \int \int x^p y^q f(x,y) dx dy \tag{2.1}$$

where $f(x,y)$ is a two dimensional Cartesian function; p and q are integers. $(p+q)$ is defined as the order of the moment. Table 2.1 summarizes the definition of the first three moments used in this chapter.

2.3.2 Interpreting tactile image with moments: basic algorithms

In this section we derive the analytical relationship among moments of a tactile image and the contact parameters causing the image. Contact location, normal and shear contact forces are then resolved using moments of the tactile image. Two types of tactile sensors are discussed, a *one-dimensional* tactile array sensor which measures only a single stress component (usually the normal stress component perpendicular to the sensor surface), and a *multi-dimensional* tactile sensor which measures more than one stress component.

0th Moment	$M_0 = -P$
1st Moments	$M_x = -(dQ_x + aP)$ $M_y = -(dQ_y + bP)$
2nd Moments	$M_{xx} = -[2adQ_x + P(d^2 + a^2)]$ $M_{yy} = -[2bdQ_y + P(d^2 + b^2)]$
Center of Tactile Image	$X_c = d\frac{Q_x}{P} + a$ $Y_c = d\frac{Q_y}{P} + b$

Table 2.2: *Summary of moments of σ_{zz} in the world frame $\{W\}$*

Tactile Sensor measuring a single stress component

A tactile sensor which measures only the normal stress perpendicular to the sensor surface, namely a 1-D tactile array sensor, is the most commonly used tactile sensor. Many commercial tactile sensors use this sensing structure.

From linear elasticity, the normal stress component perpendicular to the sensor surface can be expressed in the force frame {F} as [32]:

$$\sigma_{zz}(x', y') = -\frac{3}{2\pi} d^2 \frac{Pd + Q_x x' + Q_y y'}{(x'^2 + y'^2 + d^2)^{\frac{5}{2}}}, \tag{2.2}$$

where d is the depth at which the tactile array sensor is embedded. P, Q_x and Q_y are normal and tangential force components along z, x, and y axis, respectively. Note that both the normal and tangential forces contribute to the normal stress at subsurface, and the normal stress $\sigma_{zz}(x', y')$ is independent of the Possion's ratio.

In order to retrieve the contact parameters from moments, it is necessary to calculate moments of tactile image with respect to the world frame $\{W\}$. This can be done by first establishing the analytical relationship between moments and contact parameters in the force frame $\{F\}$, and then transforming them to those in the world frame $\{W\}$. Table 2.2 summerizes the result of the moment derivation according to (2.1) and (2.2). For simplicity, all moments are integrated over an infinite area of the "half-space." Modelling error due to the actual finite sensor size is introduced by such a simplification. Further discussion on this effect is given in section 2.4 and 2.5.

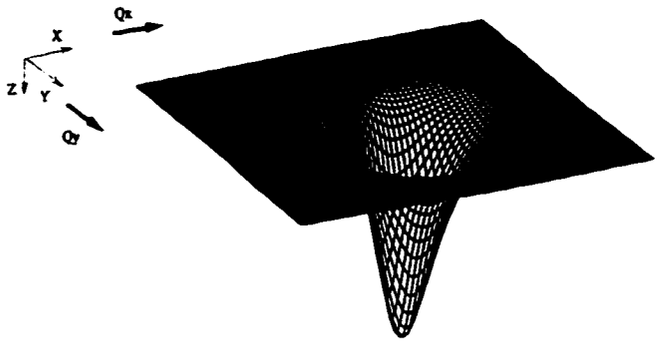


Figure 2.2: *The direction of the shear forces*

1-D tactile sensor with a single layer of transducers

Using the analytical results tabulated in Table 2.2, all zeroth, first and second moments are required to determine the five contact parameters if only a single 1-D tactile array is used. Solving the five moment equations simultaneously, a unique inverse solution can be obtained as follows if the direction of shear force is known in advance.

$$P = -M_0$$

$$Q_x = sign(Q_x)\frac{\Delta_x^{\frac{1}{2}}}{d} \quad Q_y = sign(Q_y)\frac{\Delta_y^{\frac{1}{2}}}{d} \tag{2.3}$$

$$a = \frac{M_x + sign(Q_x)\Delta_x^{\frac{1}{2}}}{M_0} \quad b = \frac{M_y + sign(Q_y)\Delta_y^{\frac{1}{2}}}{M_0}$$

where $sign(Q_x)$ and $sign(Q_y)$ are the signs of Q_x and Q_y in force frame $\{F\}$, and

$$\Delta_x = (dM_0)^2 + (M_x)^2 - M_0 M_{xx} \geq 0.$$

$$\Delta_y = (dM_0)^2 + (M_y)^2 - M_0 M_{yy} \geq 0.$$

One drawback of this model is that in practice Δ_x and Δ_y may not always be positive because of the sensor noise, causing the algorithm to fail in some situations.

Also, notice that in computing the shear forces and the position of contact point, $sign(Q_x)$ and $sign(Q_y)$ must be obtained first. One way is to examine the relative position of the positive and negative stress peak. Due to the nature of the stress function given in (2.2), when at least one of the shear forces exist, there are always two peaks of stress distribution at the subsurface; one negative, reflecting compression, and one positive, reflecting tension caused by the shear force(s). If the approximate positions of the two peaks are detectable by the tactile sensor, then the shear forces simply point from the positive peak to the negative peak. As an example, refer to Figure 2.2. In this case, the vector from the "peak" to the "valley" has positive components in both directions. Therefore both $sign(Q_x)$ and $sign(Q_y)$ are positive. However, in practice many sensors can not measure positive stress and also very often the positive stress is too small to detect, and this makes the algorithm not practical in these situations.

1-D tactile sensor with two layers of transducers

From physiology, there are different types of human tactile sensors distributed at different depths under the human skin and different human tactile sensors function differently [31]. However it is not clear whether humans use these tactile sensors simultaneously to resolve contact identities. Inspired by stereovision in computer vision, Speeter first suggested the use of tactile arrays at different depths to find the direction of the shear forces [52]. The sandwich style sensing structure was referred to as *stereotaction* in analogy to stereovision in computer vision. Unfortunately neither theoretical nor experimental study was pursued in [52].

Suppose two tactile arrays are mounted within the elastic medium at depths d_1 and $d_2(d_1 < d_2)$ respectively. Using the results in Table 2.2, a unique inverse solution can be found as:

$$P = -M_0$$

0th Moment	$SM_0 = -Q_x$
1st Moments	$SM_x = -(dP + aQ_x)$
	$SM_y = -bQ_x$

Table 2.3: *Summary of the moments for* τ_{xz}

$$Q_x = \frac{(M_x)_1 - (M_x)_2}{d_2 - d_1} \quad Q_y = \frac{(M_y)_1 - (M_y)_2}{d_2 - d_1} \tag{2.4}$$

$$a = \frac{(M_x)_1 d_2 - (M_x)_2 d_1}{M_0(d_2 - d_1)} \quad b = \frac{(M_y)_1 d_2 - (M_y)_2 d_1}{M_0(d_2 - d_1)}$$

where $(M_i)_1$ and $(M_i)_2$, $(i = 0, x, y)$ stand for moments of tactile image at $d1$ and $d2$, respectively.

Obviously, using the double-layered sensing structure, the first two moments are enough to determine all unknowns in force and location. Structure of more than two layers is not necessary for point contact.

Multi-dimensional tactile array sensor

Instead of using multiple tactile array sensors to provide "stereo" tactile data, efforts have also been made to build multi-dimensional tactile sensors where more than one stress/strain component can be sensed [5] [35]. Ricker and Ellis have shown by simulation that shear strain/stress is a key component in tactile recognition [45].

Let us assume the shear stress τ_{xz} to be available on the sensor surface. From [32], the analytical expression for the shear stress is,

$$\tau_{xz}(x', y', d) = -\frac{3}{2\pi} x d \frac{Pd + Q_x x' + Q_y y'}{(x'^2 + y'^2 + d^2)^{\frac{5}{2}}} \tag{2.5}$$

Similar to the process of normal stress, we first derive the corresponding moment equations. These results are summarized in Table 2.3, where SM_i $(i = 0, x, y)$ are the moments of the shear stress distribution.

Using the first two moments from *both* normal and shear stress measurements an inverse model can be obtained as the follows for the multi-dimensional sensing

structure.

$$P = -M_0$$
$$a = \frac{M_y - dSM_0}{M_0} \quad b = \frac{SM_x}{M_0} \tag{2.6}$$
$$Q_x = -SM_0 \quad Q_y = -\frac{M_x - bM_0}{d}$$

Similar to the 1-D two-layer structure, this model requires only the zeroth and first moments of a tactile image.

2.4 Improving the modelling accuracy

The moment equations derived in Table 2.2 and Table 2.3 are based on the integration of the stress function over the entire surface of the half-space. When the contact point is close to the center of the sensor, it is reasonable to speculate that the most significant stress portion is "received" by a tactile sensor of reasonable size, because from equation (2.2) and (2.5) stress is highly concentrated near the point of contact. Nonetheless, care needs to be taken when the point of contact moves towards the border of the tactile sensor where large modelling errors may result.

To improve modelling accuracy, we introduce "moment compensation factors" [14], which are defined as the ratios between the actual moments $M_i(n)$ calculated within a finite-size square mask centered at the contact point and the moments from an infinite large sensor area, i.e,

$$K_i(n) = \frac{M_i(n)}{\lim_{x\to\infty} M_i(n)} \tag{2.7}$$

where $i = 0, x, y, xx, yy$, and n is the half length of the moment calculation mask.

$K_i(n)$ can be derived analytically if all contact parameters are given. However, from the stress function (2.1) it is easy to observe that, when the moment calculation mask is symmetric with respect to the moment calculation axes and centered at the contact point, K_i is a function only of the sensor depth and of the size of the moment calculation mask. Therefore, instead of using the analytical expression for K_i, the

compensation factors can be pre-calculated numerically as a function of the size of the mask by using typical force values. The results are then saved in a look up table to be retrieved at run-time. This is very efficient for real-time computing.

Using the moment compensation factors, the following iterative algorithm can be used to improve the modelling accuracy:

1. Use the basic algorithm shown in equation (2.3), (2.4) or (2.6) to calculate $a(0)$, $b(0)$, $P_x(0)$, $Q_x(0)$ and $Q_y(0)$ as initial values.

2. Recalculate moments using an $l \times l$ mask centered at point $(a(j-1), b(j-1))$ where $l = min(L_a - |a|, L_b - |b|)$, L_a and L_b are the half length of the sensor, and j is the iteration index.

3. Divide the moments obtained in Step 2 by the corresponding compensation factors $K_i(l)$, and substitute them into the basic algorithms to calculate the new parameters $a(j)$, $b(j)$, $P_x(j)$, $Q_x(j)$ and $Q_y(j)$.

4. Repeat step two and three until the position of the moment calculation mask does not change, i.e, $|a(j) - a(j-1)| < tol_a$ and $|b(j) - b(j-1)| < tol_b$, where tol_a and tol_b are the stop criteria.

The convergence of the iterative algorithm is not proved in this chapter; however, by simulation we have examined the input space of $\{1 \leq P \leq 15$ Newton, $0 \leq Q_x \leq 0.5P$, $0 \leq Q_y \leq 0.5P$, $0 \leq a \leq (L_a - 3)$, $0 \leq b(L_b - 3)\}$. Fortunately, the algorithm converges for all three models and the typical number of iterations is less than 10.

2.5 Discussion

2.5.1 The relationship between contact forces and the shift of the center of tactile image

In [52][53], it was observed empirically that the direction of the translation in tactile image center is highly correlated with the direction of the shear force. Results in

Table 2.2 are readily available to investigate the relationship between contact forces and the amount of translation in the center of the tactile image.

Firstly, for the case of two tactile array sensors embedded at depths d_1 and d_2, the x-coordinates of the image center at the two depths are given by

$$X_{c1} = d_1 \frac{Q_x}{P} + a \qquad X_{c2} = d_2 \frac{Q_x}{P} + a \tag{2.8}$$

Subtract the two equations. The shift in image center between two layers is therefore:

$$\Delta X_c = X_{c2} - X_{c1} = (d_2 - d_1)\frac{Q_x}{P} \tag{2.9}$$

Equation (2.9) supports the empirical observation made in [52] where it was found that, for the case of stereotaction, the shift in image center (referred to as the "error") is proportional to the shear force when a constant normal force is applied.

Secondly, for each individual tactile sensor, if the ratio between the normal and shear forces is increased by a factor of k, the x-coordinate of the image center will shift by

$$\Delta X_c = d(k-1)\frac{Q_x}{P}. \tag{2.10}$$

Equation (2.10) explains the linear correlation between the translation of the image center and the normal to shear force ratio, found empirically in [53]. It also provides a means to measure this ratio, which is important in ensuring grasp stability.

2.5.2 Effect of finite sensor size on modelling accuracy

Simulations were conducted in order to investigate the effect of the finite sensor size on the modelling accuracy. The tactile sensor chosen for the numerical studies is a 26x26 planar tactile array sensor, 26x26 mm^2 in size (roughly 1 inch by 1 inch), mounted at $d_1 = 2mm$ and $d_2 = 4mm$ under the skin for the two-layer structure. Stress distribution is generated according to equation (2.2) and (2.5) for the following cases.

$$\text{Case 1: } P = 1(N), Q_x = 0.3P, Q_y = 0.1P;$$

Case 2: $P = 11(N), Q_x = 0.3P, Q_y = 0.1P$;

Case 3: $P = 11(N), Q_x = 0.3P, Q_y = 0.5P$.

In each case, forces are applied at 12×12 locations over the sensor surface. Table 2.4 summarizes the results of the simulation, when the criteria for calculating the "sensible area", i.e., the sensor area within which the contact parameters can be recovered, is taken as ± 1 tactel for contact location estimation and $\pm$ 1 Newton for force estimation. Model 1 to 3 in Table 2.4 correspond to the models for a 1-D single-layer sensor, 1-D two-layer sensor and multi-dimensional sensor, respectively. Notice that case 3 is a very unfavorable contact condition where both large normal and large shear forces are applied.

The simulation indicates that the models work well when contact happens in a smaller area inside the sensor area. When contact happens very close to the tactile sensor boundary, large estimation error occurs. In such a situation, the contact parameters are impossible to retrieve accurately even with the iterative improving algorithm. The size of the "sensible area" depends on the acting forces.

Simulation results in Table 2.4 also indicate that the models for the 1-D two-layer sensor and the multiple dimensional sensor have roughly the same "sensible area". The single layer 1-D sensor model has the smallest "sensible area" and its accuracy is the worst compared with the other models, a phenomenon which we believe is due to the fact that this model makes use of the second moments, which are less accurate than the zeroth and first moments because, by definition, the stress is weighted to the square of the distance for the second order moments.

2.5.3 Experiments

We were not able to perform experiments on the previous models with shear force applied. The reasons were that a multi-dimensional tactile array sensor was not available to us; the 1-D tactile array sensor we had was built on two plastic layers (see Figure 2.3 for a picture of the tactile sensor), which are very heterogeneous

	Model 1	Model 2	Model 3
Case 1	14x14 mm^2	20x20 mm^2	20x20 mm^2
Case 2	14x14 mm^2	18x18 mm^2	20x20 mm^2
Case 3	12x12 mm^2	16x16 mm^2	16x16 mm^2

Table 2.4: *The size of the "sensible area". Model 1 to 3 correspond to the models for 1-D single layer sensor, 1-D two layer sensor and multiple dimensional sensor, respectively.*

when mounted in the middle of two rubber layers; these plastic layers also do not "transfer" well the tension caused by the shear force. However, with the available equipment, we were able to test the 1-D one-layer model when no shear force is applied. A simple experimental apparatus, picture shown in Figure 2.4, was built for the experiment, where a spring loaded pin can move down toward the sensor to apply varying normal forces. A force sensor was placed under the tactile array sensor to provide the reference force reading. The ruber used to cover the tactile sensor is 2 mm in thinkness. The white rectangular area in Figure 2.4 is the area under which the tactile sensor is mounted.

Table 2.5 summarizes some of the experimental data. Figure 2.5 shows the percent error of the inverse model. The experimental results indicate that the error in contact parameter estimation is less than 15 %.

Although sensor design is beyond the scope of the thesis, through our experiments, especially the unsuccessful experience with shear contact force, we feel that, in order to make tactile sensor reading more reliable, meaningful and easy to process, an integrate-designed tactile sensor is essential. Sensing elements (ideally elastic) should be fabricated directly into the soft skin rather than mounted to rubber layers afterwards like in our experimental system. Heterogeneous properties of the sensing system should be eliminated if possible. The entire sensing system should be flexible to enable easy attachment to end-effectors of different shapes.

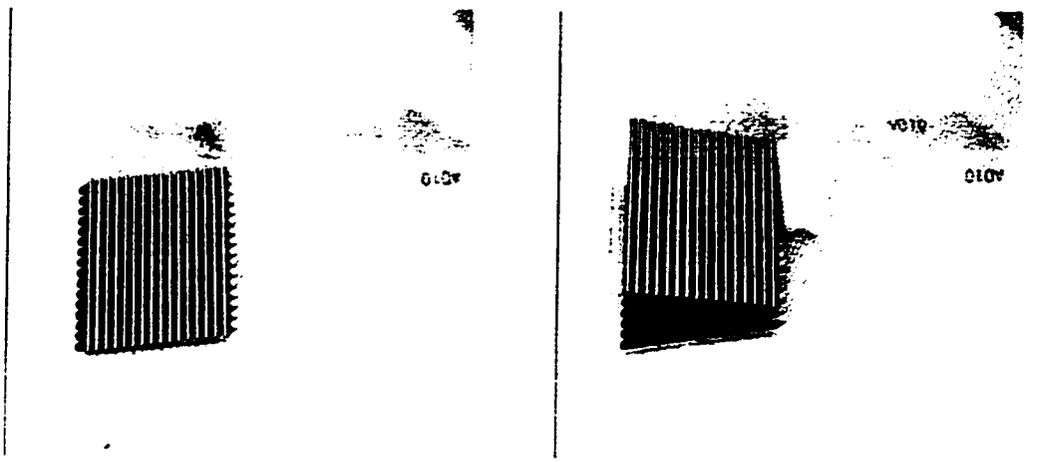


Figure 2.3: *The tactile sensor used for experiments*

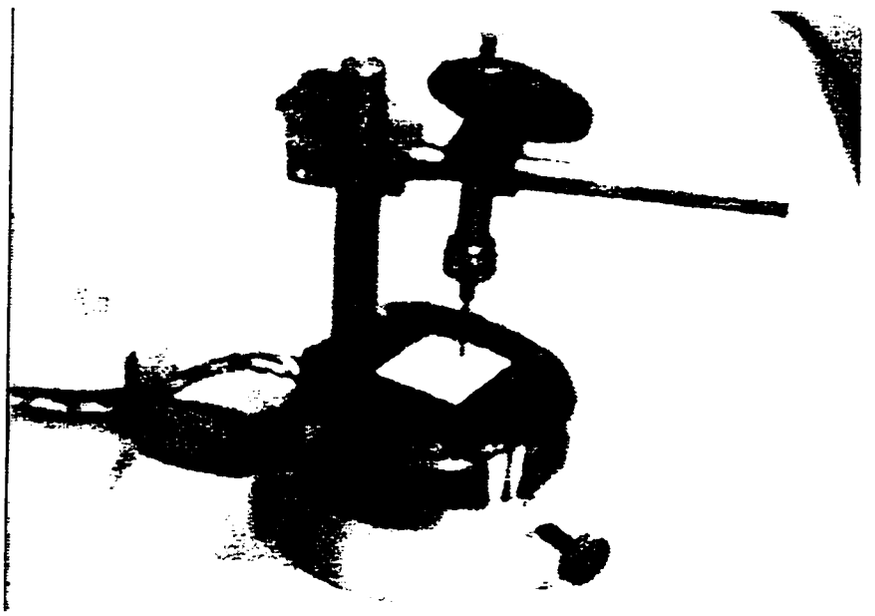


Figure 2.4: *The experiment apparatus*

	Calculated Values			Real Values		
	a (tactel)	b (tactel)	P (Newton)	a (tactel)	b (tactel)	P (Newton)
Case 1	-0.075	-0.042	-0.643	0.00	0.00	-0.640
Case 2	0.005	0.070	-0.849	0.00	0.00	-0.979
Case 3	1.258	1.070	-1.050	1.25	1.25	-1.080
Case 4	2.168	2.221	-0.998	2.50	2.50	-1.014
Case 5	3.485	3.794	-1.009	3.75	3.75	-1.164

Table 2.5: *The experimental results*

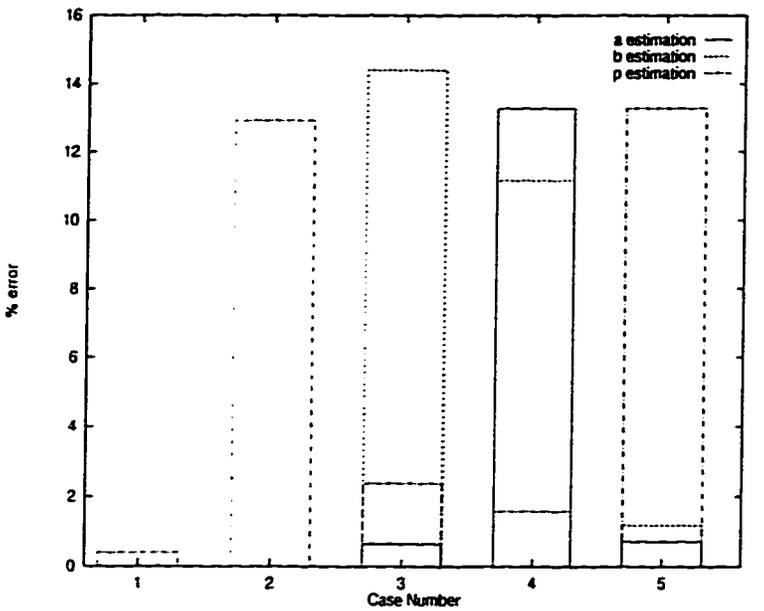


Figure 2.5: *The percent error of the parameter estimations.*

2.6 Conclusions

Analytical inverse models calculating contact location and contact force from the first three moments of a tactile image are presented in this chapter. Several different sensing structures are discussed.

It is shown in this chapter that the first three moments of a tactile image contain sufficient information to determine force and location of contact. For a single layer one dimensional tactile sensing system, moments up to second order are required in order to uniquely determine all three force components and the contact location, while for a two-layer one dimensional tactile sensor and a multi-dimensional tactile

sensor with both normal and shear stress measurement, the zeroth and first moments are sufficient to recover all the contact parameters.

A 1-D single layer sensing structure is simple to implement but the corresponding algorithm requires more computation and is in general less accurate, according to our numerical simulation. Furthermore, in order to provide an unambiguous solution, the single layer algorithm requires the direction of the shear forces to be determined in advance by associated methods such as finding the relative position of positive and negative stress peaks.

A two-layer one dimensional system and a multi-dimensional tactile sensor require less computation and are less sensitive to the error in moments for tactile image processing. These sensing structures have the advantage of being able to find both the magnitude and the direction of shear forces. The disadvantage, however, is that they are more difficult to implement.

Chapter 3

Tactile sensing of line contact: contact force and location sensing

A real-time tactile image processing algorithm for edge contact is presented in this chapter. Based on basic elasticity results, closed-form solutions for calculating contact force and local contact geometries, the location and orientation of the line of contact, from the first three moments of a tactile image are derived. Computational complexity of the proposed algorithm and those of the previously published results are compared and passive tactile sensing experiments are performed. It is shown that the proposed algorithm has the advantage of preserving force information and is more consistent and computationally efficient in comparison with the previous results.

3.1 Introduction

Line or edge information is important for many vision and touch systems. The extracted lines or edges are usually used to build object models and, in some cases. to segment images for further processing. From the manipulation point of view, detecting and tracking object edges is one of the most important and frequently used "task primitives", which, when applied in a proper sequence, can accomplish a complex robot exploration or manipulation task [39].

Tactile data processing of line/edge contact has received considerable attention in the past, mainly due to the importance of edge information in both recognition and control. In this chapter we study the problem of extracting contact force, contact location and orientation of a line/edge contact[11]. The modelling method to be used is the same as that used in the previous chapter, i.e., to use elasticity theory to model the contact. A simple solid mechanics model is introduced in this chapter. It enables us to obtain a simple yet practical edge detecting algorithm that can be applied directly to real-time tactile-based robot control and active tactile sensing applications.

The rest of this chapter is organized as follows. In section 3.2, previous research is reviewed. Section 3.3 presents the formal statement of the problem and the proposed solution. In section 3.4, the computational complexity of the proposed algorithm and those of several previous approaches are compared. Experimental results are presented in section 3.5. Finally, the conclusions are drawn in section 3.6.

3.2 Previous research

3.2.1 Vision-like algorithms

Early research on edge/line extraction from tactile data has been heavily influenced by visual image processing techniques. Muthukrishnan *et al.* (1987) applied some of the well-known edge detection techniques in computer vision to tactile image process-

ing [41]. In this study, a 3×3 median filter is first used to remove noise while retaining edges. For objects smaller than the effective area of the sensor, "local operation" and "sequential method" are used to find edge contours. For larger objects, a straight-line extraction algorithm is adopted. Since the result of this method is a visual image, it does not provide quantitative information about local contact. Besides, this method is computationally expensive [3].

Berger and Khosla (1991) have applied simple visual image processing techniques to edge extraction from tactile image and successfully conducted real-time edge tracking experiments with tactile feedback [2]. Their edge extraction algorithm consists of an edge detector and a line parameter calculator. The edge detector is essentially a thresholder which removes the "crosstalk" or blurring effect caused by the elastic covering of the tactile sensor. For each force sensor in the tactile sensor array, or the so called "*tactel*", the force values of its four-connected neighbors are first checked. If any of these neighbors has a force reading larger than a preset threshold value, the current tactel reading is assumed to be caused by crosstalk and its force value is set to zero. A simple edge finding rule is then applied to locate the edge in the image without crosstalk. In order to extract quantitative information about the location and orientation of the edge, an adaptive Hough transform [3] is applied. In [2] normal force control using tactile sensor is also implemented; however, the force estimation is based on the thresholded image without the "blurring" or "cross-talk" among tactels. The relationship between the force so estimated and the real contact force is not discussed.

The accuracy of Berger and Khosla's method largely depends on the setting of the threshold value which is used to remove crosstalk among tactels and is determined by simple empirical observations. A consistent threshold value, however, is difficult to choose, due to the fact that the threshold is affected by not only the sensor noise, but also the crosstalk characteristics that change with the magnitude and direction of the applied force, indentor shape, and material properties of the sensor covering.

Obviously, an inaccurate threshold may mistakenly retain or remove tactels.

3.2.2 Solid mechanics based algorithms

Instead of treating a tactile image the same as a visual image, a different approach to tactile data processing is to investigate the mechanism of the stress or strain distribution on the tactile sensor according to solid mechanics [20] [33] [53].

Using basic results from linear elasticity, Fearing and Hollerbach (1985) have studied the problem of an infinitely long edge contacting a planar tactile sensor [20]. In this study, each tactel of the sensor array is treated as an individual force sensor to provide one constraint on the contact parameters. For an edge contact, therefore, the three unknown contact parameters in contact location, orientation and contact forces could in principle be solved from strain equations at three different tactels. However, an analytical solution was not been given due to the complexity of the strain equations for each individual tactel.

The advantage of using stress/strain analysis for tactile data processing is that it is based on the physical principles of the tactile sensing process. Force information, which is critical for manipulation, is retained unlike the vision-like approaches. Unfortunately, it is extremely difficult to solve the general problem of finding contact parameters, such as local contact forces, location, and object geometry, from tactile readings involving arbitrary-shaped tactile sensor and object because of the complexity in the relevant contact mechanics (see appendix B).

In this chapter, we address the issue of extracting contact force, contact location and orientation for the case of a long edge contact, using a *single* tactile image obtained from a planar tactile array sensor. We tackle the problem by a simplified mechanics model with an infinite long normal line force acting on an elastic half-space. Tangential forces are ignored for simplicity. Instead of considering each tactel as an individual force sensor, like that in [20], features of the entire image or its first three moments are used to retrieve the contact parameters. It is shown that the first three

moments of tactile image provide sufficient information for detecting edge contact. The corresponding algorithm is more efficient than the previous algorithms, and is appropriate for real-time applications.

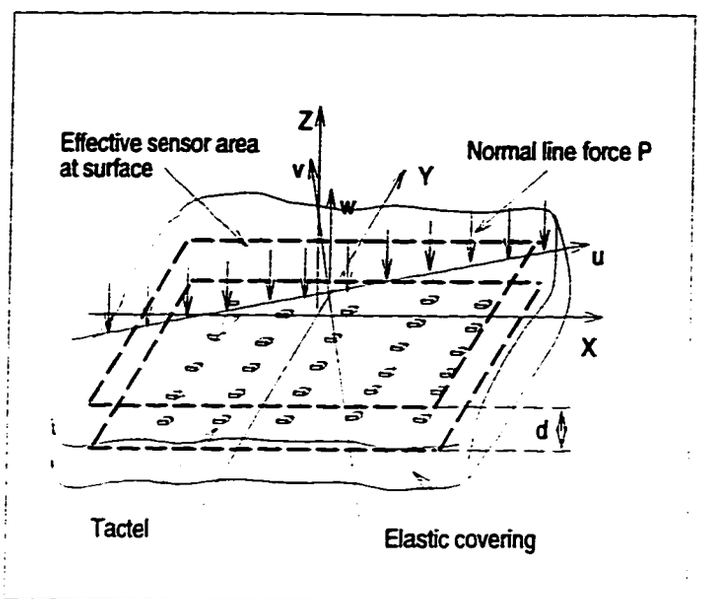
3.3 Problem definition and the proposed solution

3.3.1 Problem definition

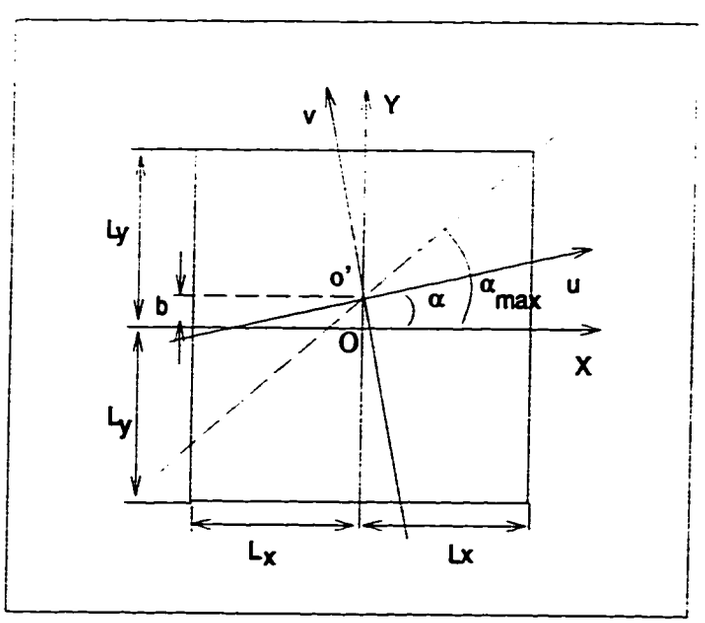
Without loss of generality, we attach a world reference frame $\{W : XYZ\}$ to the geometric center of the sensor surface, where X and Y axes lie parallel to the edges of the tactile sensor, as illustrated in Figure 1(a). A force frame $\{F : uvw\}$ is also defined, where w and u axes are along the direction of normal force and the line of contact, respectively. The origin of the force frame is chosen as the intersection of the Y axis and the v axis, as shown in Figure 3.1(a). Our task is to extract contact forces, and the location and orientation of the line of contact from a tactile image, which is taken as the normal stress distribution at the subsurface in this discussion.

To simplify the problem, we assume the edge is infinitely long and intersects the left and right edges of the tactile sensor, as shown in Figure 3.1(b); i.e., we require $\alpha < atan\left(\frac{L_x - b}{L_y}\right)$. Since in most tactile perception and manipulation tasks, the robot arm or dextrous hand is guided to an approximate contact position by other sensors such as a visual sensor before tactile information is used, this condition is not difficult to satisfy in practice. Under this assumption, the x coordinate of the center of the contact line within the sensor surface is always zero. Therefore, mathematically we have reduced the number of unknown contact parameters to three, namely, the normal force P, relative contact location b, and contact orientation α, as shown in Figure 3.1(b).

Notice that the case of a straight line contact is not general; however, one can obtain a polygonal approximation of a long edge by sequential probing and integrating the straight-line segments so extracted. We shall use the moment definitions and



(a)



(b)

Figure 3.1: *The problem definition of tactile sensing of edge contact*

notations introduced in the previous chapter in the following discussion. Also we assume $L_x = L_y$ for simplicity.

3.3.2 Basic elasticity for tactile image processing of edge contact

The tactile sensing system with a soft compliant layer is usually modeled as an elastic half-space. From linear elasticity (see, for example, [32] and [55]), the normal stress distribution at the subsurface is given in the force frame {F} by the following equation:

$$\sigma(u,v) = -\frac{2Pd^3}{\pi(v^2+d^2)^2}, \tag{3.1}$$

where σ is the normal stress along w direction and d is the depth at which the tactile sensor is embedded. P has a dimension of force per unit length.

In principle, stress remains constant along any line parallel to u axis of the force frame within the half-space. The stress function corresponding to normal force P is independent of the material property of the elastic covering.

3.3.3 Derivation of moment equations of a tactile image

In order to retrieve contact parameters from moments, it is necessary to calculate the moments of the tactile image with respect to the world frame $\{W\}$. This is done by first establishing the analytical relationship among the moments and the contact parameters in the force frame $\{F\}$, results shown in Table 3.1. This relationship is then transformed to the corresponding moments in the world frame $\{W\}$ by rotating about axis w and then translating along axis v. Equation(3.2) through (3.7) summarize the final results of the moment derivation.

$$M_0 = -\frac{2pL}{\cos(\alpha)} \tag{3.2}$$

$$M_x = bM_0 \tag{3.3}$$

$$M_y = 0 \tag{3.4}$$

Zeroth moment	M	$-\frac{2pL}{\cos(\alpha)}$
First moment	M_u	0
	M_v	0
Second moment	M_{uu}	$-\frac{2d^2pL}{\cos(\alpha)}$
	M_{vv}	$-\frac{2pL^3}{3\cos(\alpha)^3}$
	M_{uv}	0

Table 3.1: *Summary of moments expressed in the force frame*

$$M_{xx} = -\frac{d^2pL}{\cos(\alpha)} - \frac{pL^3}{3\cos(\alpha)^3} + \left(-\frac{d^2pL}{\cos(\alpha)} + \frac{pL^3}{3\cos(\alpha)^3}\right) \cos(2\alpha) + b^2M_0 + 2bM_x \tag{3.5}$$

$$M_{yy} = -\frac{d^2pL}{\cos(\alpha)} - \frac{pL^3}{3\cos(\alpha)^3} - \left(-\frac{d^2pL}{\cos(\alpha)} + \frac{pL^3}{3\cos(\alpha)^3}\right) \cos(2\alpha) \tag{3.6}$$

$$M_{xy} = \left(\frac{d^2pL}{\cos(\alpha)} - \frac{pL^3}{3\cos(\alpha)^3}\right) \sin(2\alpha) + bM_y \tag{3.7}$$

For simplicity, all moments are integrated over the infinite area of the elastic half-space. Modeling error arises because the actual sensor size is finite. However, according to (3.1), the stress concentrates in a small area close to the contact line and decays at the fourth power of the distance from the line of contact. Therefore the modeling error is expected to be small.

3.3.4 Solution: contact parameters from moments of tactile image

To solve for the three unknown contact parameters we could in principle choose any three independent moment equations from (3.2) to (3.7). However, the solution from only three moments will introduce a $cotan(\alpha)$ term, which is noisy when angle α is small. To overcome the problem, we use all of the four independent moment equations to obtain a robust solution shown below.

$$b = \frac{M_x}{M_0} \tag{3.8}$$

$$\alpha = \frac{1}{2} atan\left(\frac{-M_{xy}}{M_{xx} - M_{yy} - 2bM_x - b^2 M_0}\right) \tag{3.9}$$

$$P = -\frac{Mcos(\alpha)}{2L} \tag{3.10}$$

3.4 Comparison of computational complexities

The computational complexity of the edge-detection algorithm is of concern because edge detection and edge tracking are often performed in real-time. Usually a passive tactile sensing can provide only extremely local information and motion is needed to effectively learn about the environment. One advantage of the algorithm presented in this chapter is that it is a simple low-level tactile information processing algorithm that can be integrated into real-time robot controller [13]. In [3], the computational complexity up to their edge detector is discussed and compared with Muthukrishnan's algorithm [41]. Obviously a more complete discussion should also include the computation of extracting quantitative contact parameters which are then used to provide quantitative information and to drive the robot.

For the purpose of comparison, in this chapter we distinguish between the *preprocessing* of the original image and the *parameter extraction* subsequent to the preprocessing. Preprocessing operates on the raw tactile image and produces as the output an image of the same dimension as the raw image. Parameter extraction, on the other hand, determines the numerical parameters that define the location and orientation of the edge.

3.4.1 Complexities of preprocessing

For preprocessing, the average complexities of Muthukrishnan's and Berger and Khosla's algorithms are given in [3] as:

$$Cp_M = 2K(M \times N)n^2 log(n) + 2n^2(M \times N) + 2(M \times N) \tag{3.11}$$

$$Cp_B = 6(M \times N) \tag{3.12}$$

where Cp_M and Cp_B are the number of numerical operations needed for Muthukrishnan's and Berger and Khosla's algorithms, respectively, where $M \times N$ is the dimension of the original tactile image, n is the dimension of the window used for convolution and median filtering, and K is the number of iterations of the median filter in Muthukrishnan's algorithm.

In comparison, for the method proposed in this study, we use a constant threshold in preprocessing to remove noise, as described later in section 3.5. For such a preprocessing, the number of operations is $M \times N$.

3.4.2 Complexities of parameter extraction

In [2], the numerical parameters of the edge function are calculated by the modified adaptive Hough transform (MAHT) [4], which is based on Illingworth's adaptive Hough transform (AHT) [30], except more heuristic rules are applied. As a very conservative estimation of the complexity of MAHT, we use the formula given for AHT to calculate the minimum number of operations required for MAHT.

Suppose we wish to resolve each line parameter to an accuracy of 1/R of its full range, the calculation required for standard Hough transform (HT) is $R \times M \times N$ [25]. The ratio between the computational complexities of the standard HT and AHT is given in [30] as:

$$R_c = O\left(\frac{Rlog(\gamma)}{2(\beta+1)log(R)}\right) \tag{3.13}$$

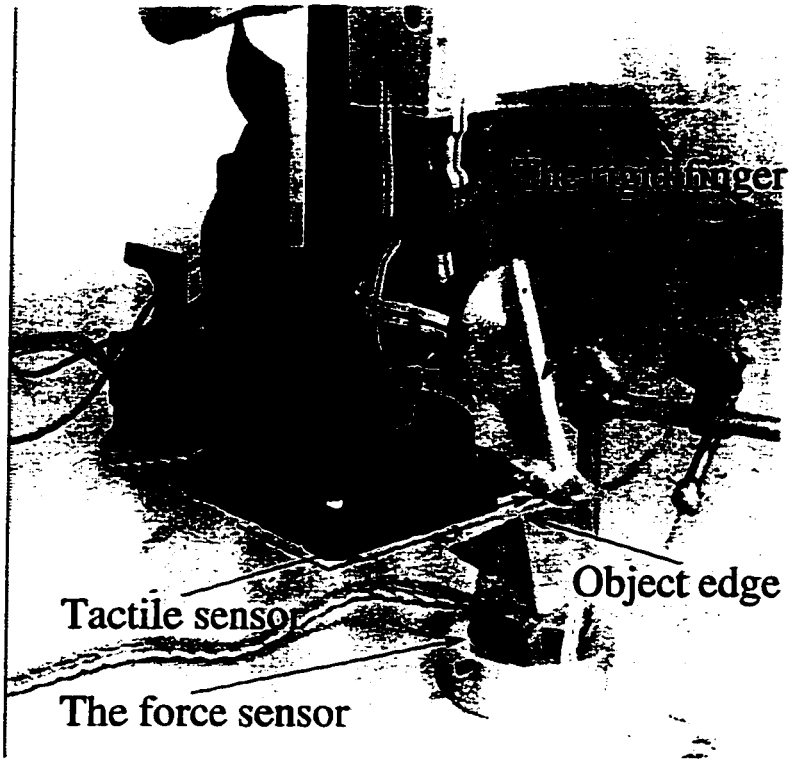


Figure 3.2: *The experiment apparatus. Tactile sensor is mounted at the robot end-effector. Force is applied to the object edge through the robot motion. A 6-D force/torque sensor is place under the object to provide reference force reading.*

where $1/\gamma$ is the amount by which the parameter range is reduced at each iteration, and β is the size of the accumulator for AHT. Usually $\beta = 3 \times 3$ and $\gamma = 3/3 = 3$. From Equation(3.13), the minimum number of calculations for MAHT can be estimated as

$$C > O\left(\frac{2(\beta+1)MNlog(R)}{log(\gamma)}\right). \tag{3.14}$$

For the proposed algorithm in this chapter, the major computation involved is the calculation of the four moments, which has a complexity of the order $4 \times M \times N$.

As a numerical example, Table 3.2 shows the computation in different stages of processing as well as the total computation required for the three different methods discussed above. We have assumed a tactile image of 16×16 tactels and an calculation accuracy of 0.5 mm for the algorithm by Berger and Khosla (1989). In this example,

	Preprocessing	Parameter Calculation	total
Muthukrishnan *et al* (best case)	9129K + 3200	N/A	$> 9132K$
Berger and Khosla (average case)	$> 6 \times 16 \times 16$	> 7297	8833
Algorithm in this chapter	16×16	$4 \times 16 \times 16$	1280

Table 3.2: *A numerical example of the computational complexities*

our algorithm is about seven times more efficient than the algorithm by Berger and Khosla (1989)[3].

3.5 Experiments

The algorithm described in the previous section was tested with a piezoresistive tactile array sensor, mounted on a planar end-effector of a PUMA 260 industrial robot shown in Figure 3.2. The tactile sensor used for this experiment has an tactile array of 16×16 distributed over an area of $25.4mm \times 25.4mm$. A three-dimensional force/torque sensor, mounted between the end-effector and the object (edge), is employed to give the reference force values.

The tactile image obtained from our sensor is very noisy. Figure 3.3 shows the original tactile image of an edge contact, where the brightness represents the intensity of stress value at each tactel. To remove noise, we first apply thresholding with a threshold of roughly four times of the maximum sensor reading when no force is applied. Figure 3.4 shows the resulting image after this simple preprocessing.

In order to compare our algorithm with the algorithm by Berger and Khosla (1991), we applied the algorithm described in [2] to find the appropriate threshold values in order to remove the "crosstalk" among tactels. Figure 3.5 is the resulting image after applying the edge detector[2]. Note that it is a binary image and has completely lost the force information. The Hough transform is then applied to this image to find the coefficients of the edge.

Table 3.3 shows the experimental results using the algorithm described in this chapter and the algorithm by Berger and Khosla [2], for the cases of three passive

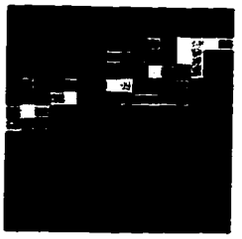


Figure 3.3: *An example of the original tactile image of an edge contact*

Figure 3.4: *An image after the preprocessing (thresholding)*

tactile sensing experiments. The force readings from the tactile sensor are compared with the reference readings from the force sensor.

The experimental results show that the proposed algorithm has similar or better accuracy in contact position and orientation estimation, compared with the previous approaches, while the former also calculates the contact force fairly accurately within an applied force range. When the contact force is too small, the signal to noise ratio is low, or when the contact force is too large and some of the tactels become saturated, large error in force estimation may occur.

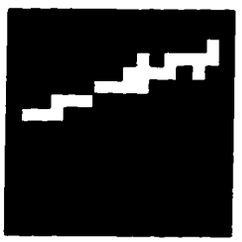


Figure 3.5: *An image after the edge detector in Berger and Khosla (1991)*

	Real Values			With Our Algorithm			Using Berger & Khosla (1991)		
	b tactel	α degree	P N/M	b tactel	α degree	P Newton	b tactel	α degree	P Newton
Case 1	3.8	12.0°	1.29	3.62	12.8°	1.27	3.9	13.6°	N/A
Case 2	2.5	18.6°	0.96	2.48	19.8°	0.97	2.56	23.0°	N/A
Case 3	2.2	21.5°	0.43	2.31	21.8°	0.57	2.26	22.0°	N/A
Average error	-	-	-	0.1	0.8°	0.06	0.07	2.0°	N/A

Table 3.3: *The experimental results*

3.6 Conclusions

Analytical expressions relating the first three moments of a tactile image to the contact location, orientation and contact force of a line/edge contact are provided in this chapter. The corresponding model has not only the advantage of preserving force information, but also a computational simple form in comparison with the previous approaches, and can be easily implemented in real-time. This chapter has presented only the results for straight line contact. A polygonal approximation of the contour of a larger object edge, however, can be obtained by integrating the straight line segments extracted with this algorithm.

The limitation of the proposed algorithm comes from the simplifications that ignore the tangential forces and require the contact line to be longer than the sensor dimension. Also the algorithm may fail to identify the edge during edge-tracking of a curved edge with small radius of curvature, or when tracking two edges intersecting at a small angle, where a very sharp turn has to be made.

Chapter 4

Local object shape from tactile sensing

Shape sensing is one of the most important aspects of tactile sensing. A discussion of tactile sensing is incomplete without considering shape sensing. A major difficulty in the study of robotic contact problems, such as the shape recovery problem, is that they usually involve complex geometry and contact kinematics. In the first part of this chapter, we introduce a convenient matrix expression for local 3-D surface geometry. Such an expression can be easily applied to manipulate object geometry, contact constraints and robot kinematics equations using homogeneous transformation. In the second part, we apply the theoretical results introduced in the first part to derive an active tactile sensing algorithm.

4.1 Introduction

Contact force, location and object shape are the most important information to obtained from tactile sensing. We have discussed tactile sensing of contact location and force in the previous chapters. In this chapter, we focus on recovering local object shape from the tactile measurements.

Generally the local object surface about a given point can be approximated by a second order surface patch, which in differential geometry is characterized by the location of the point, the tangential plane and the principal curvatures at that point. When contact position is available from, for example, the algorithms given in the previous chapters, the shape sensing problem is reduced to a curvature finding problem in nature.

There are two major approaches toward local surface or curvature recovery from tactile sensing; using solid mechanics and using kinematics and differential geometry. In the first approach, the spatial "tactile pattern", i.e. the distributed strain or stress measurements from tactile array sensors, is related directly to the contact force, location and the parameters of the geometry of the contact objects (see, e.g., [9] [19] [44]). This approach is similar to that taken in the previous chapters in finding contact force and location. The difficulty with this method in finding shape, however, is that the general shape recovery problem is more complicated. In fact, an analytical model for the contact problems with arbitrary object geometries is in most cases impossible to obtain [1][61].

An alternative and more indirect approach is to use kinematics and differential geometry [7] [8] [12] [40]. This method does not consider local phenomena such as contact stress or strain distribution. Instead, it assumes point or line contact between bodies and ignores the object deformation due to contact. We will adopt such an approach in this chapter to develop a shape sensing algorithm. We will follow the

[1] see Appendix B for an introduction to the general contact problem.

previous authors in assuming rigid body contact between bodies, and assume that the contact position is directly or indirectly available from contact sensing. In other words, in taking the kinematics and differential geometry approach, we consider object shape sensing as a higher level perception over position sensing. The position information must be derived *before* shape sensing commerces. The contact position information can be obtained from tactile sensing using algorithms given in the previous chapters or from other algorithms based on different sensing mechanism, such as force/torque sensing.

The rest of the chapter is organized as follows. In section 4.2, previous research is reviewed. In section 4.3, the surface matrix is defined and basic contact constraint equations are established. An active tactile sensing algorithm is presented in section 4.4, which combines robot direct kinematics and contact sensing to obtain an analytical expression of the local object surface. The results of numerical simulations are presented in section 4.5. Conclusions are drawn in section 4.6.

4.2 Previous research

Most robot manipulation and exploration tasks involve direct contact between a robot and the object it interacts with. In these situations direct contact constrains the motion of both bodies (the robot and the object/environment) which creates a more complicated problem than that of free body kinematics.

The problem of direct contact between two rigid bodies has been studied as early as the 1940's in the areas of gear manufacturing and design. The relative motion investigated in those studies, however, is usually of special types that are of interest to gear and cam design and analysis only. In the area of robotics, there is essentially no restriction on the relative motion between a robot and its environment, and the uncertainties during robot motion have to be considered. Many contact problems and their applications remain interesting and challenging issues to robotics researchers.

To date the most influential studies on robot contact kinematics are perhaps those

by Cai and Roth [7] [8] and Montana[40]. In Cai and Roth [7], basic contact constraints on point contact between planar curves were introduced. This study brought many valuable results from the previous gear studies into robotics. In Cai and Roth [8] the results were extended to point contact of 3-D rigid bodies. Independently, Montana [40] studied the same problem but described his result in a totally different manner. In comparison, Cai and Roth gave a general solution to the problem of point contact of rigid bodies, using higher order contact constraints, although the higher order constraints are often noisy and therefore of restricted utility in practice. Montana, on the other hand, considered only the zeroth and first order contact constraints and presented his result under a totally different but somewhat inconvenient kinematics framework for some applications such as active tactile sensing.

Studies by Cai and Roth and Montana deal with general contact kinematics. When applied to practical problems, such as the active tactile sensing problem, they are often not convenient. For instance, the way the coordinate frame is established is not convenient for robot control once the tactile perception is done, and the object surface function is expressed in a single frame and is therefore usually in a complex structure and of higher order. Furthermore, when these results are applied to the shape sensing (or curvature finding) problems, they often rely on an assumption that a certain type of differential motion is applied between contacting bodies to simplify the problem. For example, as an application given in [40], a set of small rolling-probing motions was suggested at every single sampling point, in order to find the local curvature at that point. This is inefficient and difficult to implement in practice, because, during recognition or manipulation, large relative motion between robot and the object is often necessary. Also, due to uncertainties, a special type of contact, rolling, for example, is difficult to maintain.

In this study, we introduce a matrix expression of the local object surface. Such an expression is based on a second order approximation of a small surface patch, rather than a general and global function of the entire object surface. Instead of attempting

to express the entire object surface as an overall surface function with respect to a single frame, the active sensing scheme presented in this chapter expresses the unknown object in a set of local contact frames and the surface parameters (the surface matrix) associated with each frame. We demonstrate that taking the second order approximation of a surface patch and using the surface matrix as a mathematical tool facilitates the convenient treatment of 3-D surface geometry and robot kinematics using homogeneous transformations; therefore the problem is greatly simplified. The assumption of a special type of the relative motion between the robot and object (rolling, sliding, or rolling and sliding) is not necessary in our algorithm. Although the accuracy of such an algorithm still depends on the spatial sampling rate over the object surface, it does not depend on the assumption of infinitesimal relative motion implied by previous studies. Therefore the restriction on the speed of robot motion over the object surface is greatly reduced.

4.3 Contact kinematics

4.3.1 The definition of surface matrix

At a contact point on a 3-D surface, we establish a Cartesian coordinate system such that its Z axis coincides with the inward pointing normal at this point. The X and Y axes are orthogonal to each other and lie in the tangential plane. We name such a coordinate frame a *contact frame.*

Consider a general 3-D surface which is expressed in the contact frame as

$$f(x, y, z) = 0. \tag{4.1}$$

The general surface function can be approximated locally in the contact frame as the following second order polynomial equation (notice that this is different from a "small area" in the differential sense).

$$c_{11}x^2 + 2c_{12}xy + 2c_{13}xz + 2c_{14}x +$$

$$c_{22}y^2 + 2c_{23}yz + 2c_{24}y+$$

$$c_{33}z^2 + 2c_{34}z + c_{44} = 0 \tag{4.2}$$

Equation (4.2) can be rewritten into a quadratic matrix equation

$$[x, y, z, 1] \begin{pmatrix} c_{11}, c_{12}, c_{13}, c_{14} \\ c_{12}, c_{22}, c_{23}, c_{24} \\ c_{13}, c_{23}, c_{33}, c_{34} \\ c_{14}, c_{24}, c_{34}, c_{44} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = 0, \tag{4.3}$$

or

$$X^T S X = 0, \tag{4.4}$$

where we define a *surface matrix* S of a 3-D surface as

$$S = \begin{pmatrix} c_{11}, c_{12}, c_{13}, c_{14} \\ c_{21}, c_{22}, c_{23}, c_{24} \\ c_{31}, c_{32}, c_{33}, c_{34} \\ c_{41}, c_{42}, c_{43}, c_{44} \end{pmatrix}, \tag{4.5}$$

and

$$X = [x, y, z, 1]^T, \tag{4.6}$$

is the generalized robot position vector [38].

Some examples of the surface matrices are

1. A plane in 3-D:

$$S_p = \begin{pmatrix} 0, 0, 0, a \\ 0, 0, 0, b \\ 0, 0, 0, c \\ 0, 0, 0, d \end{pmatrix},$$

2. A sphere of radius r:

$$S_s = \begin{pmatrix} 1, 0, & 0, & 0 \\ 0, 1, & 0, & 0 \\ 0, 0, & 1, & -r \\ 0, 0, & -r, & 0 \end{pmatrix},$$

3. A cylinder of radius r whose axis is aligned with y:

$$S_c = \begin{pmatrix} 1, 0, & 0, & 0 \\ 0, 0, & 0, & 0 \\ 0, 0, & 1, & -r \\ 0, 0, & -r, & 0 \end{pmatrix}.$$

Note that all the surface matrices are described in their contact frame by our definition of the surface matrix. We will refer to a second order surface with a surface matrix S as *surface* S in the rest of the discussion.

4.3.2 Basic operations on surface matrices

In this section we show some of the useful operations required for later discussions. We will omit some of the obvious steps in the proofs of the operations.

1. Addition/subtraction:

   The surface S resulting from the addition or subtraction of two surfaces S_1 and S_2 is

   $$S = S_1 \pm S_2. \tag{4.7}$$

   This property indicates that a complex surface can be expressed as the sum of some of the simple primitive surfaces, such as those given as examples in section 4.3.1.

2. Frame transformation:

   The surface S resulting from the coordinate transformation from the original surface S_0 can be expressed as

   $$S = T^T S_0 T, \tag{4.8}$$

   where T is the homogeneous transformation from the original frame to the new frame.

   Proof:

   Suppose before transformation the surface function is

   $$X_0^T S_0 X_0 = 0. \tag{4.9}$$

Due to the frame transformation, the generalized position vectors in the two different frames are related by

$$X_0 = TX. \tag{4.10}$$

Substituting (4.10) into the generalized function in the original frame and rearranging, we have

$$X^T(T^T S_0 T)X = 0. \tag{4.11}$$

It is easy to show that $S = T^T S_0 T$ is a transformed surface matrix which preserves all the properties and operations of a surface matrix.

3. Surface normal:

The surface normal $\vec{n}$ at position $\vec{x}$ on surface S is given by

$$\vec{n} = 2LS\vec{x}, \tag{4.12}$$

or in generalized vectors,

$$\overline{n} = 2\overline{L}SX + l, \tag{4.13}$$

where

$$L = \begin{pmatrix} 1,0,0,0 \\ 0,1,0,0 \\ 0,0,1,0 \end{pmatrix}, \tag{4.14}$$

$$\overline{L} = \begin{pmatrix} 1,0,0,0 \\ 0,1,0,0 \\ 0,0,1,0 \\ 0,0,0,0 \end{pmatrix}, \tag{4.15}$$

and

$$l = [0,0,0,1]^T.$$

Proof:

Suppose f(x, y, z) is the polynomial equivalent of surface S. Therefore we have

$$\vec{n} = \nabla f,$$

where ∇ is the gradient operator.

Carrying out the differentiation, we have

$$\vec{n} = \begin{pmatrix} 2(c_{11}x + c_{12}y + c_{13}z + c_{14}) \\ 2(c_{21}x + c_{22}y + c_{23}z + c_{24}) \\ 2(c_{31}x + c_{32}y + c_{33}z + c_{34}) \end{pmatrix} = 2LSx.$$

or

$$\overline{n} = 2\overline{L}SX + l$$

4.3.3 Contact kinematic constraints

Let us label robot end-effector (in many cases a robot finger) as f and the object as o. At each instant we distinguish between the contact point P_f on the robot and the contact point P_o on the object. In addition to a fixed global frame $\{W\}$, we introduce two contact frames $\{C_f\}$ and $\{C_o\}$ which at each instant have contact point P_f and P_o as their origins, respectively, and their Z axes share the inward pointing normal of the robot and the object, respectively. The X and Y axes of P_f and P_o lie in the tangential plane and are orthogonal to each other.

The basic contact constraint we start with is the well known fact that at the point of contact both bodies share the same position in space and have colinear surface normals, i.e.,

$$^{w}P_f = ^{w}P_o, \tag{4.16}$$

and

$$^{w}n_f = -^{w}n_o, \tag{4.17}$$

where the superscripts on the left side denote the frame that the quantity is with respect to. Such a rule of notation is also used throughout the rest of the chapter.

The above basic equations hold for any motion as long as the contact is not broken. To introduce kinematics, let us consider the change in contact within a time interval Δt. Suppose the contact point moves from P_{f_1} to P_{f_2} on the robot end-effector side and from P_{o_1} to P_{o_2} on the object side within the same time interval. Since P_{f_1} and

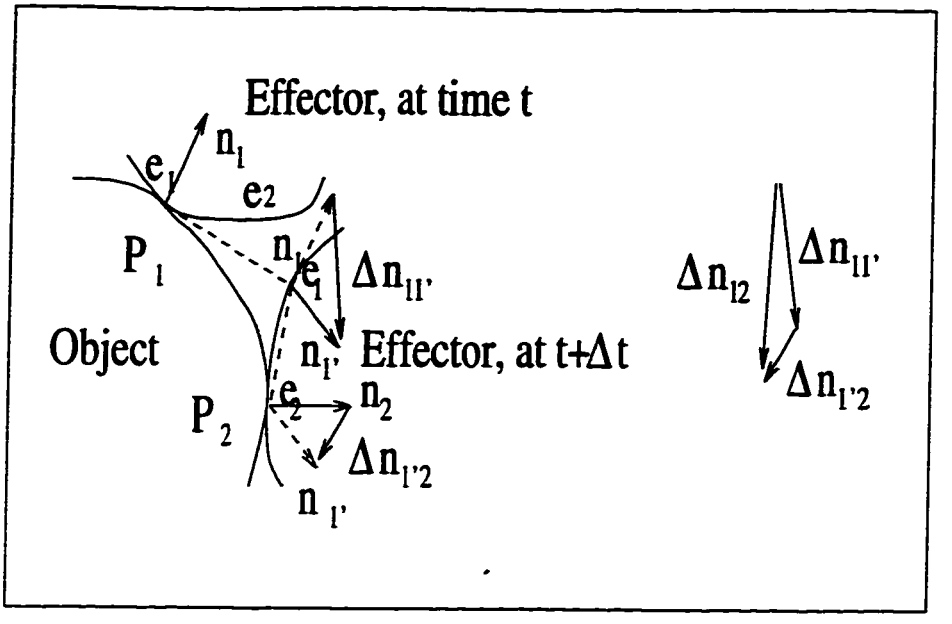


Figure 4.1: *The change in surface normal. The change in normal from t to $t + \Delta t$ is the superposition of $\Delta n_{11'}$ and $\Delta n_{1'2}$, where $\Delta n_{11'} = n_1 \times \omega \Delta t$ is the change in normal due to rotation, and $\Delta n_{1'2} = 2LSx\Delta t$ is that due to the translation from point e_1 to e_2 on the end-effector surface.*

P_{o_1} and P_{f_2} and P_{o_2} are contact point pairs before and after Δt, they have colinear surface normals before and after. Therefore,

$$\Delta n_f = -\Delta n_o. \tag{4.18}$$

Because translation does not affect the change in surface normal (see Figure 4.1 for illustration), the change in normal vector is composed of the change due to rotation, $(w \times n\Delta t)$, and that due to the contact point sliding on the surface, which from (4.12) is $2LSv\Delta t$, where v is the ***sliding velocity*** at the contact. Therefore (4.18) can be rewritten as

$$w_f \times n_f + 2LS_f v_f = -^{cf}R_{c_o}(w_o \times n_o + 2LSv_o), \tag{4.19}$$

where v_f and v_o are sliding velocities on the end-effector and the object side, respectively. w_f and w_0 are angular velocities of the robot end-effector and the object, respectively. n_f and n_0 are normals at contact point of the robot end-effector and

the object, respectively. S_f and S_o are the surface matrices of the robot end-effector and the object, respectively. ${}^{C_f}R_{C_o}$ is the rotation from the contact frame $\{C_f\}$ to contact frame $\{C_o\}$.

The sliding velocity v_f on the robot/dextrous hand is generally available from tactile sensing by differentiating the contact location measurement. However, the sliding velocity along the object surface is difficult to obtain in most cases because there are no sensors on the object. To eliminate the sliding velocity v_o from (4.19), we imagine a pair of points Q_f and Q_o, which do not belong to either contacting body and move along the robot and the object surface respectively, such that they are always at the instantaneous contact point. It is a well known result that the absolute velocities of Q_f and Q_o are equal (see, e.g., [6]), i.e.,

$$ {}^{w}V_{Q_f} = {}^{w}V_{Q_o}. \tag{4.20} $$

If we denote the relative velocity between two arbitrary points P_1 and P_2 as V_{p_1/p_2}, (4.20) can be written as

$$ V_{p_f} + V_{Q/P_f} = V_{p_o} + V_{Q/P_o}. \tag{4.21} $$

From (4.21) the sliding velocity at the object can be obtained as

$$ v_o = v_{Q/P_o} = V_{P_f/P_o} + v_f. \tag{4.22} $$

4.4 An active tactile sensing strategy for shape recovery

Touch provides only extremely local measurements. Gathering of more global information requires a robot to maintain rolling and/or sliding contact with the object while collecting and inferring information about the unknown object. This type of combined robotic manipulation and tactile sensing, or so called *active tactile sensing* [27], is essential to tactile perception. In this chapter, we assume that a robot system capable of performing active sensing is already available. We will focus on the

shape sensing algorithm only and leave the discussion on the design of the control system to the following chapters.

The shape sensing algorithm proposed in this section is to describe a general 3-D surface in terms of small surface patches defined at their contact frames (or at the spatial sampling points). We first locate the contact frame using robot direct kinematics and tactile sensing, and then solve surface matrices using contact information available from sensor readings.

Finding the contact frame

We choose the world frame $\{W\}$ as the robot base frame. The object frame is named as $\{O\}$. We also introduce a sensor frame $\{S\}$ which is attached at a convenient position on the robot effector (or finger tip).

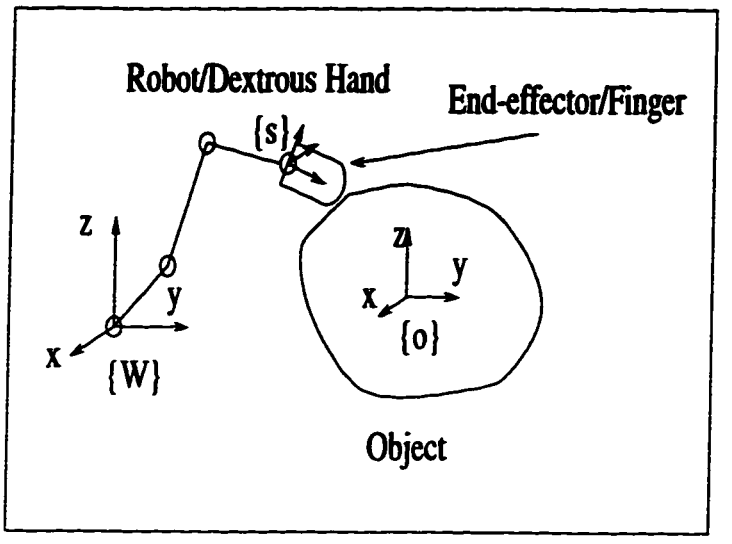
In active tactile sensing the motion of the robot/finger is usually available from direct kinematics using the robot/finger joint angle measurements. In other words, the position ${}^{w}r_s(t)$ and the orientation ${}^{w}R_s(t)$ of the sensor frame is known and time dependent due to the robot motion. Tactile sensing provides the relative position of the contact with respect to the sensor frame, i.e., the position ${}^{s}r_f(t)$ and velocity ${}^{s}v_f$ of the contact point on the robot/finger side are measurable from a tactile sensor. The surface normal at the contact point can be measured by finding the direction of the normal contact force from tactile sensing, or, in the case of active tactile sensing where the finger shape is known, the surface normal at the contact point can simply be calculated from the end-effector/finger shape given the contact position, by using (4.13). Once the surface normal is defined, the orientation of the contact frame ${}^{s}R_c$ is also defined.

From the transformation graph shown in Figure 4.2(b), we have

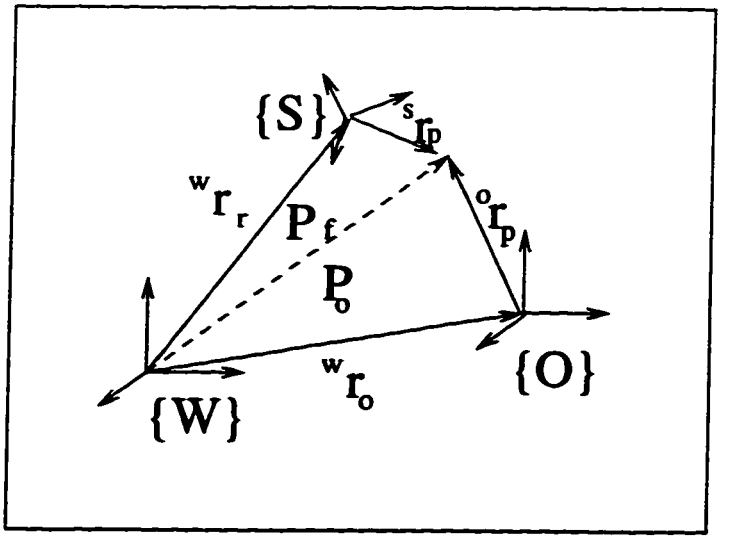
$$P_f = {}^{w}r_s + {}^{w}R_s . {}^{s}r_P, \tag{4.23}$$

and

$$P_o = {}^{w}r_o + {}^{w}R_o . {}^{o}r_P. \tag{4.24}$$



(a)



(b)

Figure 4.2: *(a) Active shape sensing using tactile feedback (b) The transformation graph*

Equating (4.23) and (4.24) (the common position constraint) the position of the contact frame with respect to the object frame can be obtained as

$$^{o}r_{P_o} = {}^{w}R_o^T({}^{w}r_s - {}^{w}r_o + {}^{w}R_S.{}^{S}r_{P_f}). \tag{4.25}$$

where ${}^{W}R_o$ is the orientation of the object frame $\{O\}$.

Similarly, using the common normal constraint (4.18), the orientation of the contact frame can be found as

$$^{o}R_{P_o} = {}^{W}R_o^T.{}^{W}R_S.{}^{s}R_C.{}^{C_f}R_{C_o}. \tag{4.26}$$

Finding the local surface parameters in each contact frame

There are ten independent elements within a surface matrix S. To solve for all of the unknowns in a surface matrix with respect to a contact frame, we use N measurements at a time; for example, we may use the current and the past N-1 tactile sensor readings.

Rewriting (4.19) for all N measurements and noting that we express all quantities with respect to the contact frames yields,

$$\begin{aligned} &{}^{C_f(t-i)}\{W(t-i) + 2\overline{L}S_f(t-i)v(t-i) + l\} = \\ &-{}^{C_f(t-i)}\{{}^{C_f}T_{C_o}[2\overline{L}S(t-i)(V(t-i) + v(t-i)) + l]\} \\ &i = 1, 2, ..., N, \end{aligned} \tag{4.27}$$

where t is the time index, W is defined as

$$\begin{aligned} W(t-i) &= \omega(t-i) \times n(t-i) \\ &= \begin{pmatrix} -\omega_y(t-i) \\ \omega_x(t-i) \\ 0 \\ 0 \end{pmatrix}, \end{aligned}$$

$\omega(t-i)$ is the *relative* angular velocity expressed in $C_f(t-i)$, relative velocity $V = [V_x, V_y, V_z, 0]$ and the sliding velocity $v = [v_x, v_y, v_z, 1]$.

Transforming all object surface matrices to the current contact frame $C_f(t)$ (refer to operation (4.8)), we have

$$S(t-i) = {}^{C_f(t)}T^{T}_{C_o(t-i)} S(t) {}^{C_f(t)}T_{C_o(t-i)} \tag{4.28}$$

Substituting the above equations into (4.27) and dropping the time index in $S(t)$ yields

$$x_i = -Sy_i, \quad i = 1 \ to \ N. \tag{4.29}$$

where

$$x_i = \frac{1}{2}\{{}^{C_f(t)}T_{C_o(t-i)} \cdot {}^{C_f(t-i)}(W(t-i) + 2\overline{L}S_f(t-i)v(t-i)) + l\}_{4\times N} \tag{4.30}$$

and

$$y_i = \{{}^{C_f(t)}T_{C_o(t-i)}[{}^{C_f(t-i)}(V(t-i) + v(t-i)) + l]\}_{4\times N}. \tag{4.31}$$

Note that we now have all of the N matrix equations expressed in terms of the surface matrix S in the current contact frame. We need, however, to drop the equations corresponding to the fourth element of x_i in (4.29), because the surface normal operation given in (4.13) provides only three non-trivial rows.

Assemble (4.29) into the following form suitable for least squares solution:

$$Ac = b, \tag{4.32}$$

where

$$c = [c_{11}, c_{12}, c_{13}, c_{14}, c_{22}, c_{23}, c_{24}, c_{33}, c_{34}, c_{44}]^{T}_{10\times 1}, \tag{4.33}$$

$$A = [A_1, A_2, ..., A_N]^{T}_{3N\times 10}, \tag{4.34}$$

$$A_i = \begin{pmatrix} y_{i1}, y_{i2}, y_{i3}, y_{i4}, 0, 0, 0, 0, 0, 0 \\ 0, y_{i1}, 0, 0, y_{i2}, y_{i3}, y_{i4}, 0, 0, 0 \\ 0, 0, y_{i1}, 0, 0, y_{i2}, 0, y_{i3}, y_{i4}, 0 \end{pmatrix}_{(i=1 \ to \ N)}, \tag{4.35}$$

$$b = [b_1, b_2, ..., b_N]^{T}_{3N\times 1}, \tag{4.36}$$

y_{ij} ($i = 1$ to N, $j = 1$ to 4) are elements of y_i given in (4.31), and b_i ($i = 1$ to N) is assembled from the first three rows of x_i which is given in (4.30). Note that trivial equations have been eliminated after the matrix assembly.

The definition of the contact frame also introduces a boundary condition $\{z = 0,$ at $x = y = 0\}$. To apply such a boundary condition, an extra row $[0, 0, ..., 0, 1]$ needs to be added to the A matrix, and an extra element [0] to the b vector. This brings the final dimension of A to $((3N + 1) \times 10)$ and that of b to $((3N+1) \times 1)$ in (4.32). A minimum of three measurements ($N \geq 3$) is required to solve the local surface S which has a total of 10 independent variables.

Finally, the least squares solution for (4.32) can be obtained as (see, for example, [10])

$$c = (A^T A)^{-1} A^T x. \tag{4.37}$$

4.5 Simulations

4.5.1 Conditions of simulation

To verify the algorithm developed in the previous section, we consider the following numerical example, where a robot manipulator equipped with a spherical tactile sensor tracks the surface of a cylindrical object. The object is unknown to the robot.

We assume ideal robot control, so that a fixed point on the tactile sensor slides on the surface of the cylindrical object. For simplicity we choose the contact frame such that its $y-$ axis coincides with the $y-$axis of the global frame $\{W\}$, as shown in Figure 4.3. We consider, at the the contact point, a constant relative velocity V_y along the cylinder axis and a constant relative velocity V_x along the cylinder tangential plane. The sign of velocity V_y, i.e., the direction the robot travels along the $y-$ axis changes periodically (this is to simulate a blind searching), also illustrated in Figure 4.3. The kinematics and geometry parameters used in the simulations are listed in Table 4.1. Sampling rate for the simulation is chosen as 1 Hertz. Note that in this example we have chosen the contact frame to initially align with the cylinder frame, for the sake of simplicity. In real situations, when the object is totally unknown to the robot, a random orientation of the contact frame on the object can be chosen, and this will

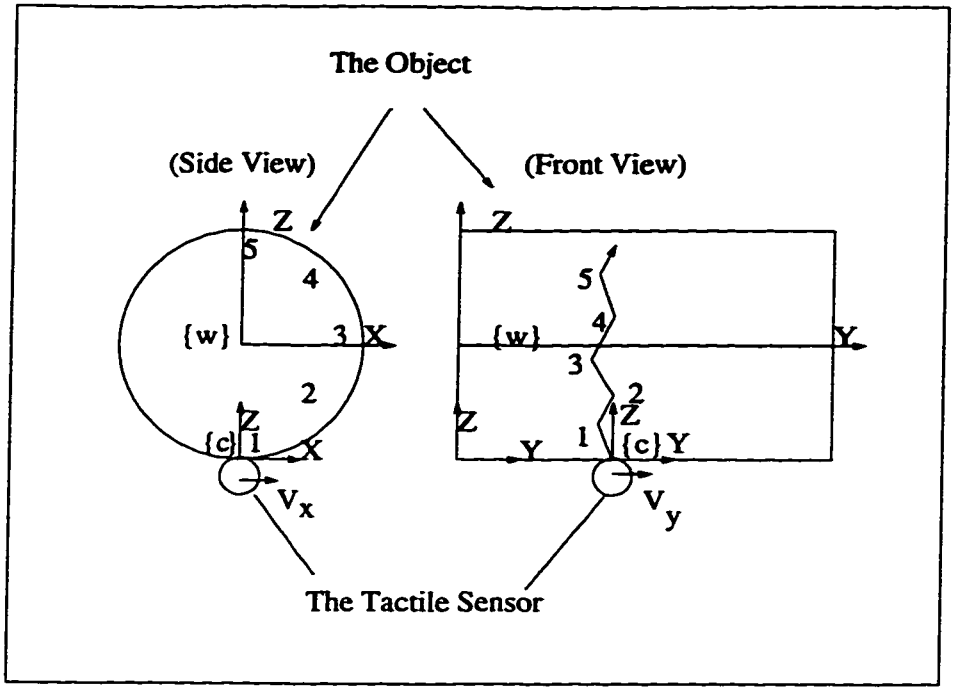


Figure 4.3: *The trajectory of the surface probing simulation*

Geometry		Kinematic parameters	
$R_{cylinder}$	R_{finger}	$V_x(mm/sec)$	$V_y(mm/sec)$
$100mm$	$10mm$	0.01 to 50	$\pm$(1 to 50)

Table 4.1: *The parameters used from the simulation.*

only affect the final analytical expression of the surface.

Because a fixed point on the tactile sensor is maintained, the pose in the contact frame is static. Therefore, we have $\omega(t) = 0$, and $v(t) = 0$ in (4.30) and (4.31). Also, the relative velocity between the tactile sensor and the object can be expressed in the contact frame as

$$^{C(t)}V(t) = \begin{bmatrix} V_x \\ V_y \\ 0 \end{bmatrix} = \text{constant.} \tag{4.38}$$

Because the geometry of the tactile sensor is known to be spherical, the transformations relating the past and the current contact frames can therefore be obtained

as

$$^{C(t)}T_{C(t-1)} = Trans(y, sign(t)V_y dt)Trans(x, -V_x dt)Rot(y, V_x/R_{cyl} dt) \qquad (4.39)$$

$$^{C(t)}T_{C(t-2)} = ^{C(t)}T_{C(t-1)} \cdot ^{C(t-1)}T_{C(t-2)}$$

$$= ^{C(t)}T_{C(t-1)} . Trans(y, sign(t)V_y dt)Trans(x, -V_x dt)Rot(y, V_x/R_{cyl} dt) \qquad (4.40)$$

...

$$^{C(t)}T_{C(t-n)} = ^{C(t)}T_{C(t-n+1)} \cdot ^{C(t-n+1)}T_{C(t-n)}$$

$$= ^{C(t)}T_{C(t-n+1)} . Trans(y, sign(t)V_y dt)Trans(x, -V_x dt)Rot(y, V_x/R_{cyl} dt) \qquad (4.41)$$

where

$$sign(t) = \begin{cases} +1, & if\ t = 2n\Delta t \\ -1, & if\ t = (2n+1)\Delta t \end{cases} \qquad (n = 0, 1, 2, ...) \qquad (4.42)$$

A simulator was programmed such that it would generate the trajectory of the contact and the frame transformations automatically based on the above discussion. Five data points are used at a time. These data, together with the given motion parameters and the geometry of the tactile sensor, are then used to test the shape sensing algorithms.

4.5.2 Simulation results

Figure 4.4 illustrates an estimated object surface patch (20 mm in x-direction and 4 mm in y-direction), when using different tracking velocity V_x. In Figure 4.5, the estimation error of the simulation is plotted at a cross-section of the cylinder. It is observed that for the same relative velocity V_x along the cylinder tangential plane, the estimation error increases at positions further away from the contact point. Also

when relative velocity V_x increases, the estimation error increases. These match our intuition that, with a larger relative velocity V_x, we sacrifice spatial sampling density along the cylinder tangent and therefore lose accuracy.

In Figure 4.6, the variation in estimation error is plotted at a cross-section of the cylinder, when random noise is applied to the relative velocity V_x. The random noise applied in this and the later simulations is generated through a uniform random number generator. The range of the random noise is taken as $\pm$ 15% of the nominal value in each case. In this figure, the middle solid dot is the actual cylinder position. The height of the upper and lower bars stand for the maximum and minimum error, respectively.

Figure 4.7 and 4.8 illustrate the surface estimation and estimation error when different V_y values are used. Figure 4.9 illustrates the estimation error when noise is introduced to the relative velocity V_y measurement. Not surprisingly, the change in V_y and the random error in V_y are not quite relevant to the sensing of a cylindrical surface.

Finally, the sensitivity of the algorithm to the noise in sliding velocity v is investigated. Figure 4.10 illustrates the estimation in a cross-section of the cylinder when random noise is introduced. It shows that the algorithm is relatively more sensitive to the sliding velocity measurement, and therefore an accurate tactile sensing of the contact position (velocity) is important to the successful object shape sensing.

The simulation indicates that, within a 20×4 mm^2 surface patch, the error in the shape estimation is less than $\pm 0.02mm$ without noise being introduced, $\pm 0.3mm$ when random noise in relative velocity is introduced, and $\pm 0.4mm$ when noise in sliding velocity is introduced. In other words, the maximum error is within 0.4 % of the radius of curvature (100 mm for the simulation), over an distance of 1/5 of the radius of curvature along the surface.

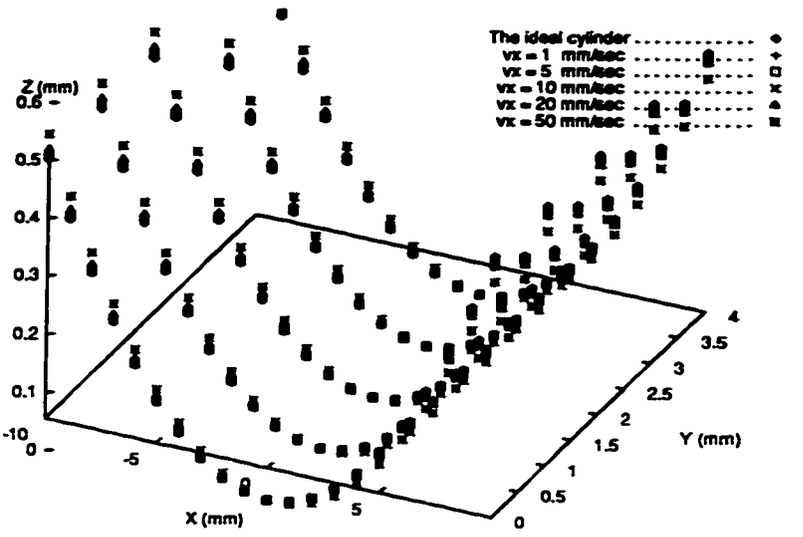


Figure 4.4: *A 3-D plot of the estimated surface patch ($20 \times 4mm^2$) when using different probing velocity V_x.*

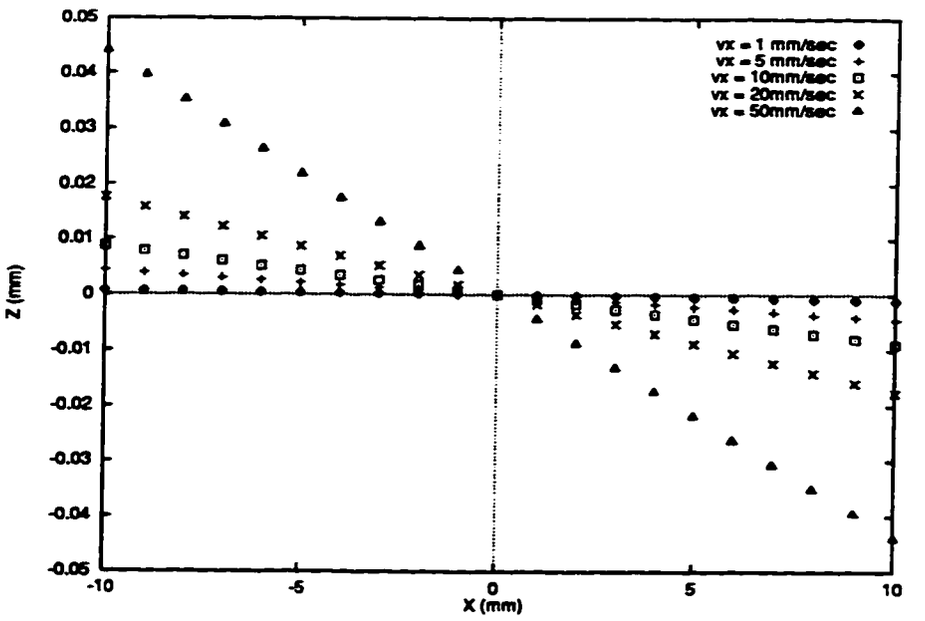


Figure 4.5: *The estimation error when using different probing velocity V_x.*

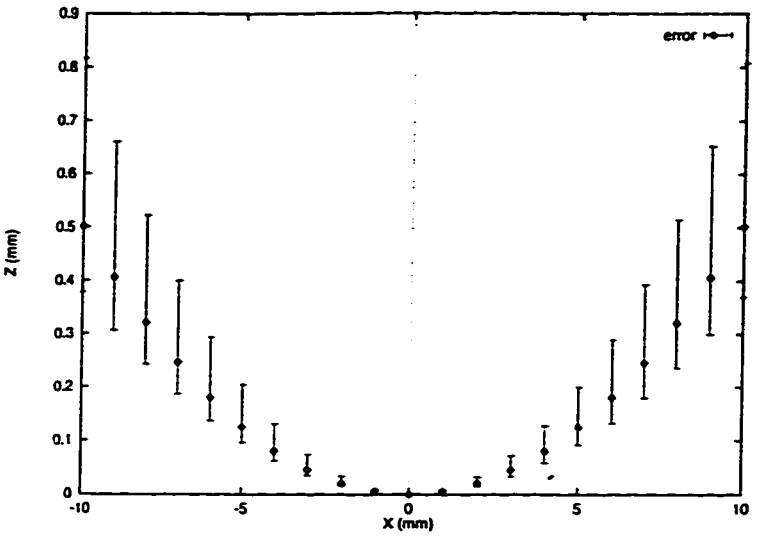


Figure 4.6: *The error in shape estimation when introducing random error in probing velocity V_x.*

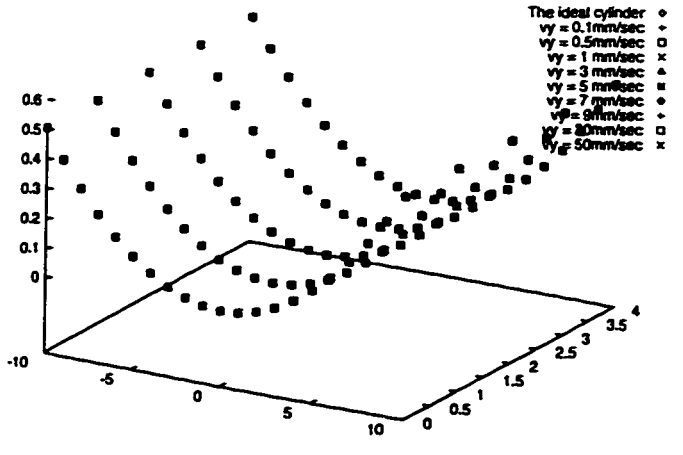


Figure 4.7: *A 3-D plot of the estimated surface patch ($20 \times 4mm^2$) when using different probing velocity V_y.*

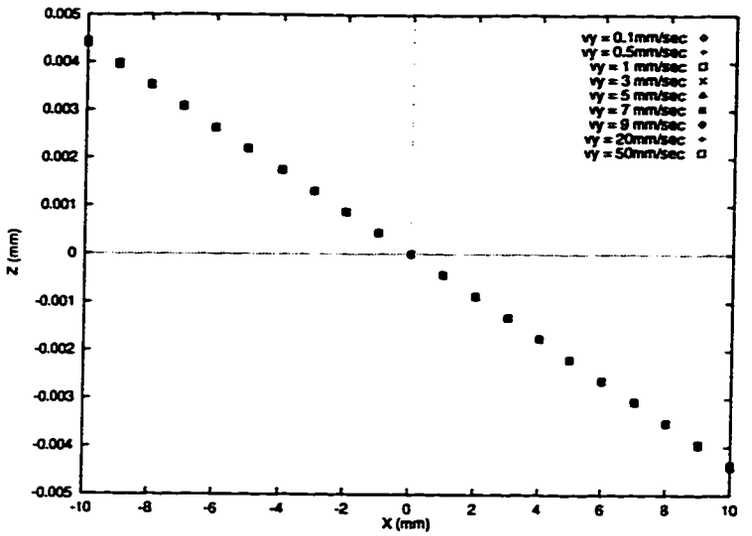


Figure 4.8: *The error in shape estimation when using different probing velocity V_y.*

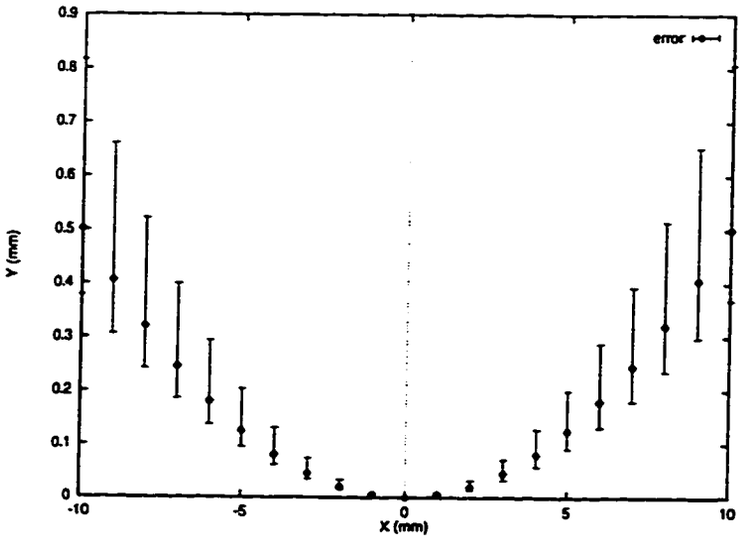


Figure 4.9: *The error in shape estimation when introducing random error in probing velocity V_y.*

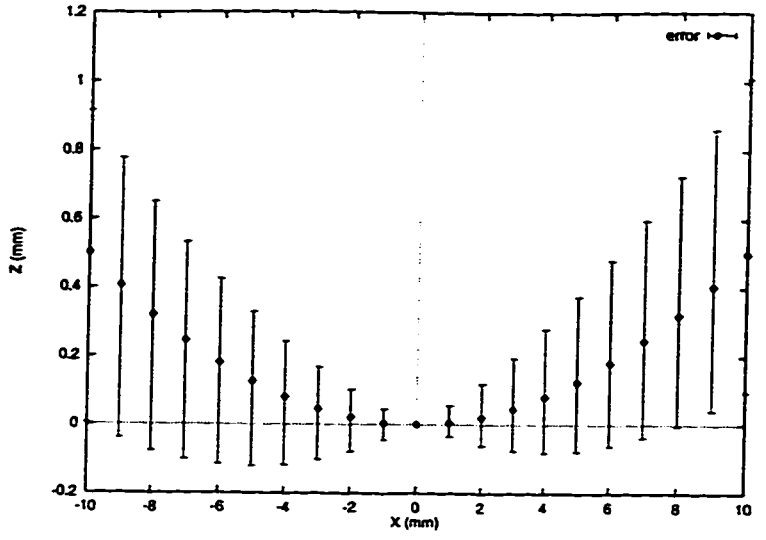


Figure 4.10: *The error in shape estimation when introducing random error in sliding velocity v.*

4.6 Conclusions

A theoretical framework to describe contact kinematics using a surface matrix is introduced in this chapter. The surface matrix is a mathematical tool that is convenient to manipulate 3-D surface geometry, contact constraints and robot kinematics equations.

An active tactile sensing algorithm to obtain an analytical expression of the local object shape is also developed. Such a surface recovery algorithm relies on robot direct kinematics and tactile sensing. Simulation results show that the maximum surface error is within 0.4 % of the radius of curvature over an area of $20 \times 4mm^2$.

Chapter 5

Direct touch-driven robot control: Using an inverse model

A method to perform direct touch-driven robot control is presented in this chapter. Using the tactile servo scheme, a robot manipulator is driven only by real-time tactile feedback from the array tactile sensors mounted on the robot end-effector. An edge tracking example is presented in detail to illustrate the implementation of the proposed method. Experimental results indicate that the control scheme presented in this chapter is more consistent and efficient than those in previous studies.

5.1 Introduction

Humans are able to perform numerous actions using touch without much conscious effort. A close look at some of the most common human actions, such as grasping, reveals that they are essentially the result of combined operations of limited numbers of primitive operations, or the so called "task primitives" [39]. A task primitive can be considered as a basic functional unit in task execution. It usually consists of the sensory perception and the associated reaction. For task primitives involving tactile sensing, we need a common mechanism to adjust the motion of a robot end-effector or fingers of a robot hand, automatically and adaptively, according to tactile readings. Such a mechanism, named *direct touch-driven control*, or "*tactile servo*"[11] [49], is the essence of robotic tactile exploration and manipulation.

Direct touch-driven control gives a robot the ability to adjust its position and orientation according to the feedback from tactile sensing, and provides a robot with an "instinct response" to tactile stimuli. Once such a basic mechanism is abstracted and established, functional primitives such as edge-tracking can be implemented, and higher level tasks can be performed as a sequential execution of the task primitives. The objective of this and the following chapter is to establish the framework to build such a basic mechanism of tactile perception.

The rest of the chapter is organized as follows. In section 5.2, previous research on touch-driven manipulation and active edge tracking is reviewed. In section 5.3 we introduce the principles of the tactile servo using an inverse tactile model. Section 5.4 provides the details of applying the tactile servo to active edge tracking. Sections 5.5 and 5.6 present the experimental results and the conclusions, respectively.

5.2 Previous research

The use of tactile information in controlling manipulation has been explored to a limited extent in recent years. In [28] a simple task to place a rectangular parallelepiped

object on a planar surface was studied. The use of tactile sensor was limited to finding the angle between the object and the gripper. In [37], an optical finger-tip tactile sensor capable of detecting the location of point contact was employed to control a two-fingered hand manipulating an unknown object. As a result of taking into account the change in contact location due to object rolling, stable grasping was achieved. In both [28] and [37] the tactile sensor was used only as a sensor of the contact geometry, but not the contact force.

In [49], the concept of tactile servo was introduced and was used to monitor both the contact position and contact force with a planar tactile array sensor, in achieving the desired trajectory of a rolling-pin. This study was novel because large relative motion between the object and the robot finger was actively controlled. An attempt was also made to establish the quantitative relationship between the state of the task and the tactile sensor output, in the form of the so called "sensitivity matrix". The generation of the sensitivity matrix in this study, however, was based on intuition and empirical observations.

In this chapter we propose a control scheme which integrates tactile feedback into low-level robot controller to drive a robot. The proposed tactile servo scheme, demonstrated by the edge tracking example[13], is theoretically more rigorous, and can be generalized to control other contact tasks. In our experiments, the direction of motion as well as the position and orientation of the robot finger are actively controlled by real-time tactile feedback. The simplicity of the analytical model to the edge tracking problem makes it possible to achieve a high sampling rate of 200 Hz, in contrast to the 4.2 Hz reported in [2].

5.3 Tactile servo using an inverse tactile model

The concept of *tactile servo* is analogous to that of visual servo in that the state of the task is characterized and controlled through a chosen feature set F, which represents and is derived from the original tactile image. In general, a task is controllable using

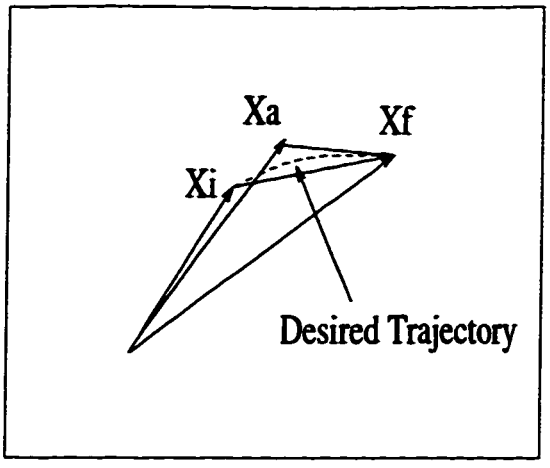


Figure 5.1: *The resolved motion rate*

a particular tactile feature set only if the controlled robot states can be sufficiently characterized by a tactile image and uniquely defined by the chosen tactile feature set.

An *inverse model* of a tactile sensing system calculates contact parameters, such as contact force, location and local object shape, from tactile data measurements (or, indirectly, from the derived features). An inverse model is opposite to a *forward model* where given contact parameters the stress/strain distribution is predicted. Inverse modelling is in general more difficult than forward modelling [29] [51]; however, an inverse model facilitates easy implementation of robot control applications.

Suppose a tactile inverse model of the following form is available.

$$X_a = f_s^{-1}(F_a) \tag{5.1}$$

where $f_s^{-1}()$ is the inverse tactile model, X_a is the actual Cartesian position of the robot end-effector/dextrous hand, and F_a is the measured feature vector of the tactile image.

The robot joint motion can be calculated using the robot inverse kinematics or using robot Jacobian, as illustrated below. Subsequently, the robot joint torque servo can be obtained from the position servo [57].

1. Using robot inverse kinematics:

   Robot inverse kinematics gives the direct mapping from the Cartesian motion of a robot end-effector to the robot joint motions. When combined with tactile sensing, the following procedure can be used to generate the robot joint angles based on the resolved motion [58].

$$\begin{aligned} \theta(t+1) &= f_\theta^{-1}(X(t) + \Delta X) \\ dX &= \frac{X_f - X_a(t)}{T_{seg} - t} dt \qquad (t < T_{seg}) \\ X_a(t) &= f_s^{-1}(F_a(t)) \end{aligned} \tag{5.2}$$

   where $f_\theta^{-1}(.)$ is the robot inverse kinematics, X_a is the actual feedback state (the end-effector transformation) obtained from the tactile sensor, t is the time index, and T_{seg} is the time segment within which the robot end-effector moves from the initial position X_i (X_i=$X_a(0)$) to the desired final position X_f (see Figure 5.1).

2. Using robot Jacobian:

   An alternative way to obtain the robot position servo is to solve the robot joint angle in velocity space using the robot Jacobian matrix. This method, when combined with tactile sensing, is expressed as follows.

$$\begin{aligned} \theta(t+1) &= \theta(t) + d\theta \\ d\theta &= J_\theta^{-1} dX \\ dX &= \frac{X_f - X_a(t)}{T_{seg} - t} dt \qquad (t < T_{seg}) \\ X_a(t) &= f_s^{-1}(F_a(t)) \end{aligned} \tag{5.3}$$

   where J_θ^{-1} is the robot/finger Jacobian matrix, and the other notations are the same as those introduced before.

5.4 An implementation: edge tracking through tactile servo

As an implementation of this method, active edge tracking using tactile servo is studied. The reason for choosing edge-tracking is that it is an important task primitive for tactile perception. In both vision and tactile sensing, edges are extremely informative cues in object recognition and exploration tasks. Many vision and tactile object recognition algorithms have been developed assuming that edge information is available. Although it is possible in many cases to locate edges using vision only, there are many other situations where vision is not convenient or applicable. Tactile sensing, on the other hand, is often a good alternative.

The control objective of an edge tracking task can be described as (refer to Figure 5.2):

- Move the finger at speed V_{set} along y axis in the tactile sensor frame;
- Maintain constant contact force P_{set} along z axis;
- Move along x axis to keep the line of contact at the center of the sensor frame;
- Rotate around z axis (adjust angle α) to maintain zero contact angle between the edge and the y axis;
- Rotate around x axis (adjust angle θ) to keep the sensor surface "flat" to a 3-D curved edge.

Notice that the above task is defined in the sensor frame, reflecting that a tactile control task has to be local.

Mathematically, the above control objectives can be expressed as:

$$\mathbf{C_d} = \begin{bmatrix} P_d \\ x_d \\ y_d \\ \theta d \\ \alpha_d \end{bmatrix} = \begin{bmatrix} P_{set} \\ 0 \\ V_{set}\Delta t \\ 0 \\ 0 \end{bmatrix} \tag{5.4}$$

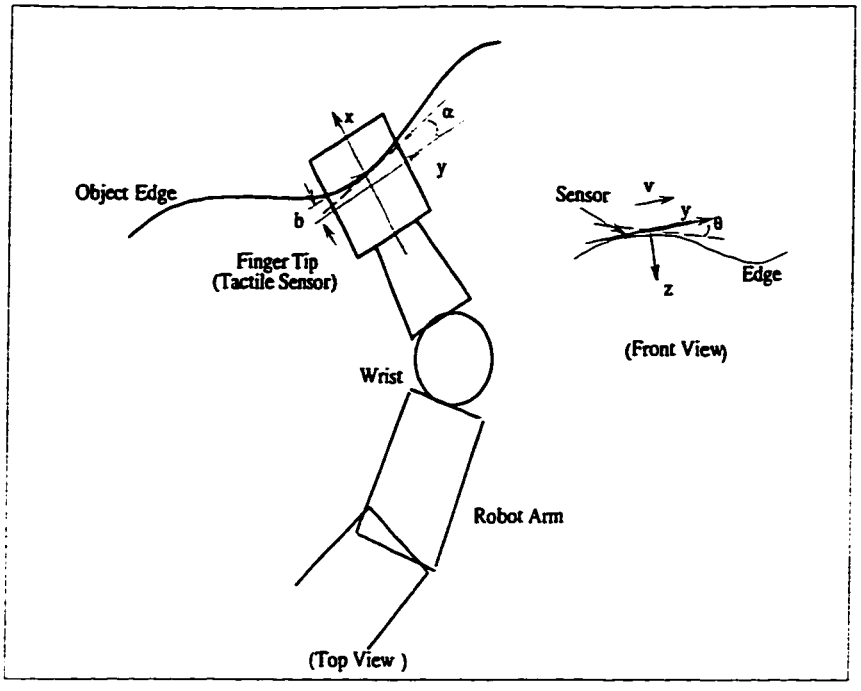


Figure 5.2: *One finger edge tracking*

where C_d represents the desired contact condition, and Δt is the control period.

The inverse model expressed in (5.1) requires position information, and we need to transform the force measurements to the relevant position quantities. This can be achieved by using the spring constants of the robot system [46]. also notice that the control of contact force in edge-tracking is implicit, and it is assumed that there is sufficient compliance provided by the elastic layer covering the tactile sensor, such that the position control is adequate.

Suppose that the overall robot and sensor spring constant along Z direction is K_z. Equation (5.4) and (5.1) can be written as:

$$\mathbf{X_d} = \begin{bmatrix} x_d \\ y_d \\ z_d \\ \delta_{x_d} \\ \delta_{z_d} \end{bmatrix} = \begin{bmatrix} 0 \\ V_{set}\Delta t \\ P_{set}/K_z \\ 0 \\ 0 \end{bmatrix} \tag{5.5}$$

$$\mathbf{X_a} = [x_a, V_{set}\Delta t, z_a, \delta_{x_a}, \delta_{z_a}]^T \tag{5.6}$$

where, from the inverse model given in (3.8) to (3.10),

$$x_a = \frac{M_x}{M_0}$$

$$z_a = -\frac{M_0 cos(\alpha)}{2LK_z}$$

$$\delta_{z_a} = \frac{1}{2} atan\left(\frac{-M_{xy}}{M_{xx} - M_{yy} - 3b^2 M_0}\right)$$

To track a curved edge in 3-D, control of the rotation around the x axis in the sensor frame is sometimes necessary. This degree of freedom (DOF) can be modelled by the first moment with respect to x axis, provided that the other three DOF's in (5.6) are under control, i.e. the fourth controlled DOF can be modelled as,

$$\delta_{x_a} = M_x$$

Note that the tracking speed along Y axis is not controlled during edge tracking (we set $y_d = y_a = V_{set}\Delta t$ in (5.6)), because the translation along y is not detectable by a tactile sensor (not a "tactile controllable" DOF), unless the frictional force can be effectively sensed.

In our edge following experiments, we use the inverse model given in (5.6) together with the robot Jacobian to calculate joint motions, as outlined in method 2 in Section 5.2. The control diagram for the edge tracking experiment is shown in Figure 5.3.

5.5 Experiments

The control scheme outlined in the previous section was implemented on our experimental system consisting of a PUMA 260 robot arm, a rigid finger, and a piezoresistive tactile array sensor mounted directly to the finger tip. Figure 5.4 shows a picture of the experimental apparatus.

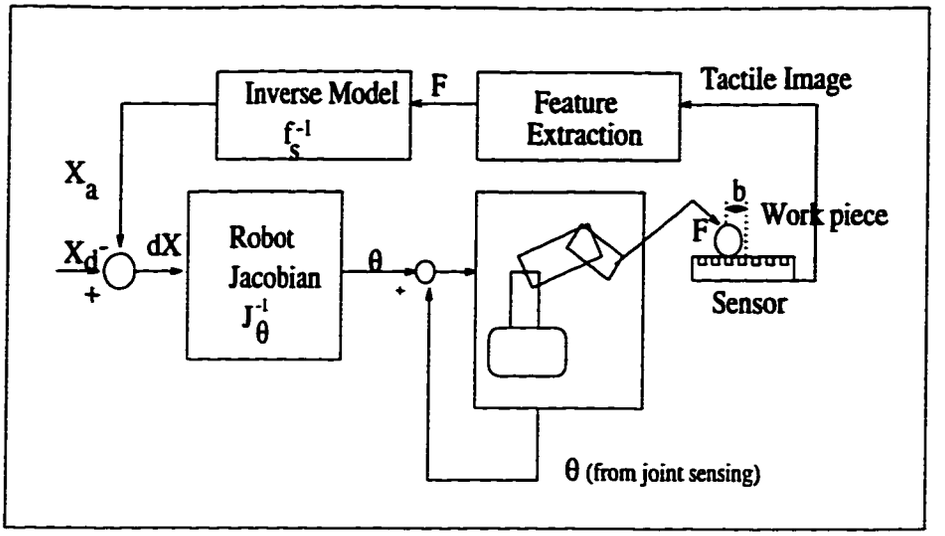


Figure 5.3: *The diagram of the controller.*

Figure 5.4: *The edge tracking experiment*

The tactile sensor used for the experiments is a planar tactile array sensor with 16x16 stress sensing elements over an area of one square inch. The sensory data processing and the real-time tactile servo were performed in a Sun 3/VME based multi-processor robot control system developed at the Robotics and Vision Research Laboratory at the University of Alberta. The detailed description of the experiment setup is given in Appendix A. For our edge tracking experiments, a control update rate of 400Hz was achieved, while the tactile data was collected at a sampling rate of 200 Hz. These were significant improvements compared with those of the previous study [2].

In our active edge following experiment, the robot finger is first guided to an approximate position close to the object. The finger then moves forward until the edge of the object is touched. Edge tracking starts as soon as the edge is within the effective sensor area.

In Figure 5.5, the experimental results are shown for planar straight line tracking. The curve represents the trajectory of the center of the tactile sensor surface. The tracking speed for this experiment is 30mm/sec. As shown in the figure, the robot finger started at an approximate position where the center of the tactile sensor is away from the tracked edge. The large position error is quickly corrected, and in the steady tracking state the error is less than 1mm according to our measurement.

Figure 5.6 is the experimental result for tracking two straight line segments. The tracking speed is 26mm/sec for this experiment and the tracking error is less than 2mm. Similarly, the initial trajectory reflects the initial finger correction movement. Figure 5.7 shows the result of tracking a curved line. The tracking speed is 23 mm/sec for this experiment.

In Figure 5.8, the result for tracking a straight line is plotted in 3-D. The tracking speed is set to 30 mm/sec for this experiment. The vertical movement at the beginning stage indicates the movement of the finger toward the edge before contact happened. Small over-correction exists at the beginning of tracking but the steady state tracking

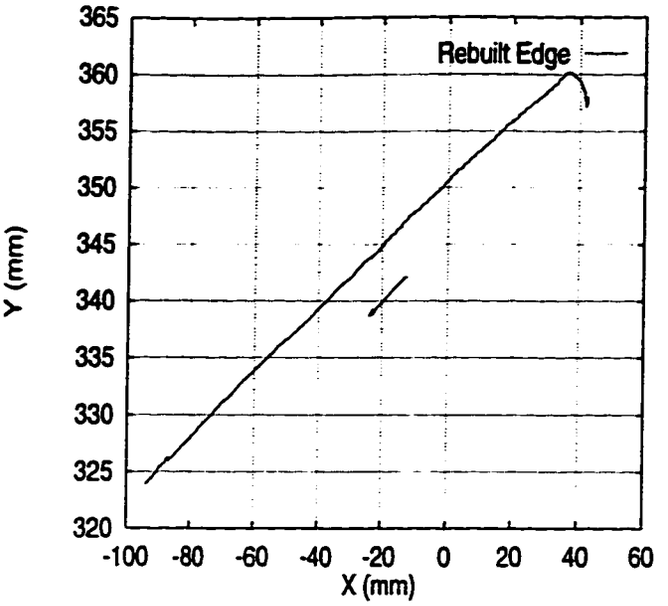


Figure 5.5: *Planar straight line tracking. The solid line is the trajectory of the center point of the tactile sensor in the world frame. The arrow indicates the direction of the tracking. The tracking speed is 30mm/sec for this example.*

is satisfactory.

Finally in Figure 5.9, an object recognition example is demonstrated for the robotic haptic sensing of a tape case. Figure 5.9(a) is a picture of the real object and Figure 5.9(b) is the rebuilt object model (outer edge only). The rebuilt object edge is obtained by combining robot direct kinematics [59] and the collected tactile features and robot joint angles during active sensing.

5.6 Conclusions

A method to perform direct touch-driven robot control using an inverse tactile system model is proposed in this chapter. The tactile servo scheme presented in this chapter combines real-time tactile sensing and low-level robot control, and is proven to be more effective, reliable and consistent than previous methods when applied to active edge-tracking. The experiments indicate that this method can be effectively used to control the tracking of straight, curved, 2-D planar and general 3-D object edges with satisfactory speed and accuracy. An object contour can be rebuilt from the collected

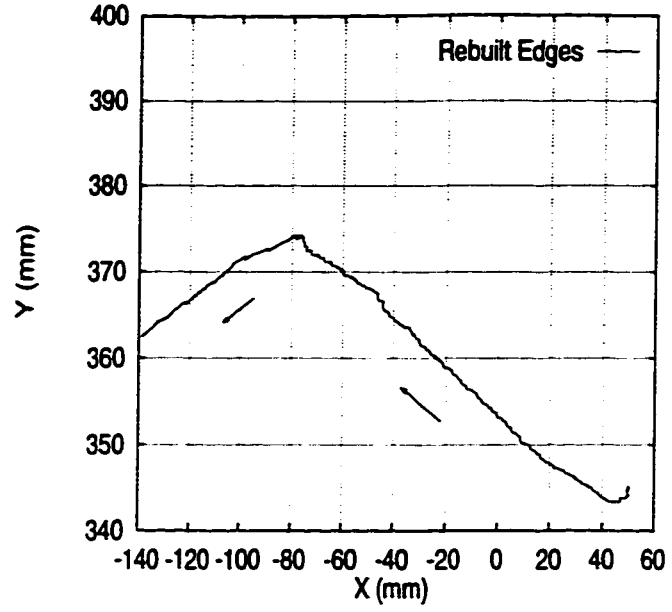


Figure 5.6: *Tracking of straight line segments. The solid line is the trajectory of the center point of the tactile sensor in the world frame. The arrow indicates the direction of the tracking. The tracking speed is 26mm/sec for this example.*

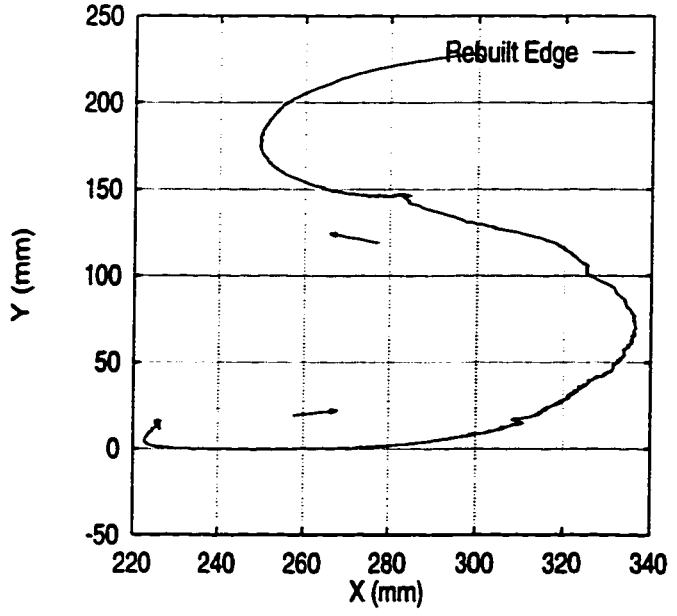


Figure 5.7: *Planar curved edge tracking. The solid line is the trajectory of the center point of the tactile sensor in the world frame. The arrow indicates the direction of the tracking. The tracking speed is 23mm/sec for this example.*

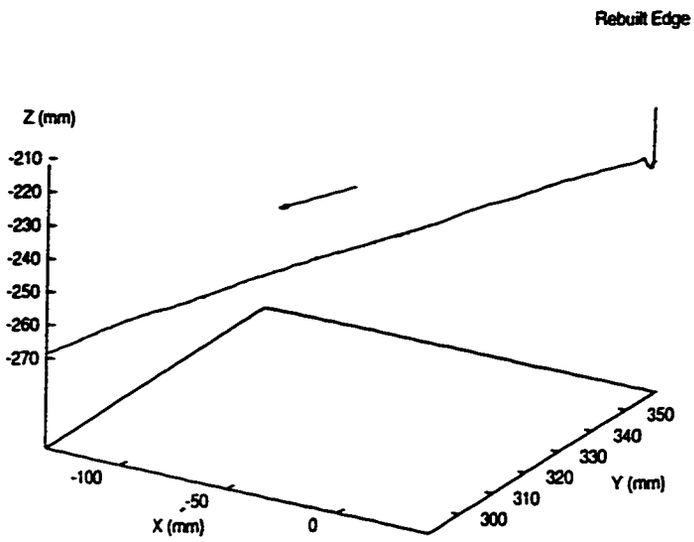
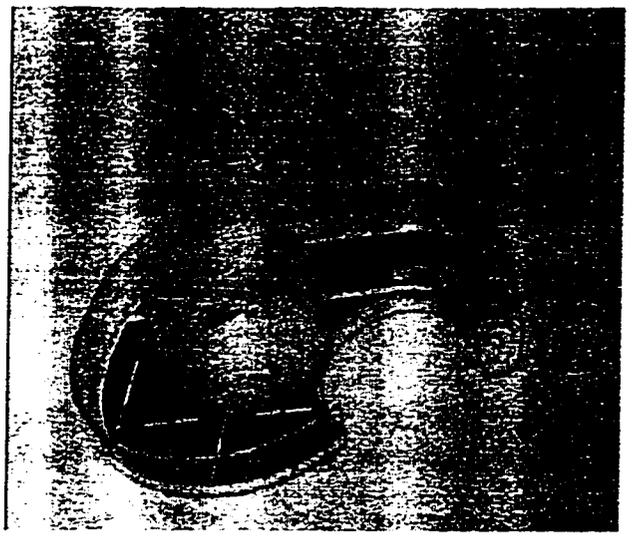


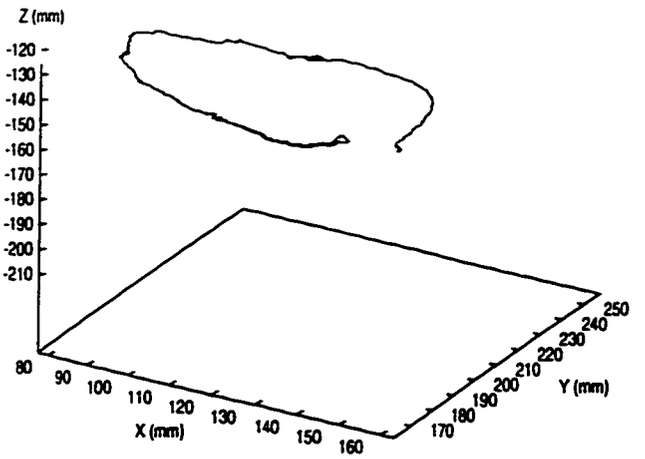
Figure 5.8: *3-D straight line tracking. The solid line is the trajectory of the center point of the sensor in the world frame. The arrow indicates the direction of the tracking. The tracking speed is 30mm/sec.*

edge information as well as the robot direct kinematics during active sensing.

This chapter has only demonstrated the implementation of the inverse modelling based tactile servo scheme with the edge tracking example. The same approach can be applied to more general control tasks where tactile or contact states are to be controlled. Although the experiments in this chapter are based on a single robot arm with a single tactile sensor, the proposed control method can be extended to control multi-fingered hand with multiple tactile sensors, where each finger can be treated as an independent (or coupled) robot arm.



(a)



(b)

Figure 5.9: *Robotic haptic sensing of an unknown object. (a) The real object. (b) The rebuilt object model (the outer edge only)*

Chapter 6

Direct touch-driven robot control: Using a tactile feature Jacobian

A scheme to perform touch-driven robot control using a tactile Jacobian is developed in this chapter. This tactile servo scheme relies on a forward model of the tactile sensing system and has several advantages over the scheme introduced in the previous chapter. Basic steps to apply the control scheme are described in detail in this chapter.

6.1 Introduction

In the previous chapter, we introduced a tactile servo scheme based upon implicit inverse modelling of the tactile sensing processes. The advantage of the inverse model based control approach is that it is intuitive and very often efficient. This is important for task primitives such as edge tracking, which is applied frequently in the execution of many complicated tasks. However, an inverse model of tactile sensing is generally difficult to obtain except for simple contact geometries [14] [29]. This limits the application of the method introduced in the previous chapter.

In this chapter we propose a different tactile-servo scheme that makes use of a forward tactile model only. Comparing with the method in the previous chapter, the main advantage of this method is that it requires only a forward model of the tactile sensing system and, since theoretically a numerical forward model such as a finite element (FE) model is available in general [60][61], such a control scheme is general. Another unique feature of this control scheme is that, with this method, the planning of robot/dextrous hand motion can be performed in the tactile feature space directly, which is desirable when applying an inverse model and going back to the task space is a noise sensitive process.

The rest of the chapter is organized as follows. In section 6.2, the basic tactile servo scheme is introduced. A discussion on constructing a tactile Jacobian matrix numerically is given in 6.3. Detailed procedures to implement the proposed tactile servo scheme are introduced in section 6.4. Finally in section 6.5 and 6.6, experimental results are presented and conclusions are drawn.

6.2 A tactile-servo scheme using a forward tactile model and tactile Jacobian

In the following discussion, we shall use the same notations which have been introduced in the previous chapter.

6.2.1 The basic control scheme

The key concept of this method is that of the "tactile Jacobian", which we define as:

$$J_t = \frac{dF}{dX} = J_t(X), \tag{6.1}$$

where X is the robot position vector; F is the "feature" of the tactile image, and it has a general form of

$$F = g(I), \tag{6.2}$$

where I is the tactile image, and $g()$ is the feature extraction function.

A tactile Jacobian relates the differential change in feature space to that in the task space, i.e., we can write

$$dX = J_t^{-1} dF, \tag{6.3}$$

where dF is the error between the desired feature F_d and the feedback feature F_a. Combining (6.3) into the velocity-based servo control using robot Jacobian J_θ, the basic steps of our tactile servo scheme can be presented in the following procedure:

For ($t = 0$ to T_{max} by Δt)

$$dF \leftarrow F_d - F_a$$

$$J_t \leftarrow J_t(X(t))$$

$$J_\theta \leftarrow J_\theta(\theta(t))$$

$$\theta(t+1) \leftarrow \theta(t) + J_\theta^{-1} J_t^{-1} dF$$

update

$$X(t+1) \leftarrow X(t) + J_t^{-1} dF$$

end of for.

6.2.2 Further comments on the control scheme

1. The definition of the tactile Jacobian J_t given in (6.1) is different from its counter-part, the "sensitivity matrix", in vision-based control. The sensitivity matrix (or the visual Jacobian) is defined in [56] as

$$J_s = \frac{df_t^{-1}}{dX}, \tag{6.4}$$

i.e., it is the differentiation of the *inverse model* f_t^{-1}. To obtain the sensitivity matrix, it therefore requires that the inverse model f_t^{-1} first be obtained.

By differentiating the inverse model, the inversion of the sensitivity matrix itself is not necessary, and the state updating, $X(t+1) = X(t) + J_t^{-1}dF$, can be avoided. This is suitable for vision-based control, because in computer vision the camera model is relatively simple. It involves only simple algebra to obtain an analytical expression for the position of the object, i.e., the inverse model. However, for tactile sensing, the process model is non-linear and an inverse model of the tactile sensing system is difficult to obtain. In other words, a direct transplant of the vision-based method is inappropriate to tactile-based control.

2. Only limited classes of contact problems have closed-form solutions[29][61], and even for these problems, analytical inverse models are difficult to obtain because the analytical solutions for the forward models are usually in complex non-linear expressions of the contact parameters. On the other hand, numerical methods have achieved much success in contact problems[60][61]. With the use of numerical methods, such as finite element (FE) methods, no restriction is, in principle, necessary on the geometry, material properties, and deformation pattern of the contact bodies in modelling the contact problems. Therefore, from the modelling point of view, a numerical forward model of a tactile sensing

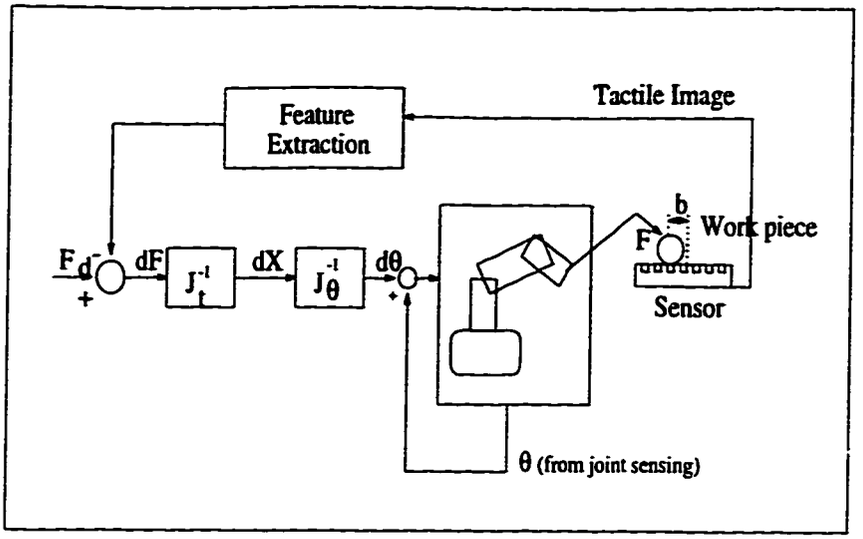


Figure 6.1: *Tactile-servo using an analytical tactile Jacobian*

system is in general available, and such a control scheme that relies on a forward model is general.

3. In this scheme, a forward model can be used to generate the desired "feature trajectory" from the desired control states in the task space. This can be done on-line or off-line. Another possibility is to obtain the desired feature trajectory through "teaching by showing", i.e., the task is run and the history of the desired tactile feature is recorded synchronously.

   Once the desired feature trajectory is obtained, the tactile servo can be performed in the *feature space* directly. Similar to computer vision, due to the fact that to estimate the robot states using an inverse model and go back to robot task space is sometimes a time-consuming and noise sensitive process [22], it is usually more effective to perform control in the feature space directly.

4. Since the tactile Jacobian is a function of robot states, we need to update the robot state vector using the actual feature measurements in the last step in the loop. Figure 6.1 illustrates the block diagram of such a control scheme when an analytical tactile Jacobian is available. When an analytical tactile Jacobian

is not available, a numerical forward model can be used to calculate the tactile Jacobian.

6.3 Constructing a tactile Jacobian numerically

A major difficulty in modelling tactile sensing is that it involves contact between objects. Extra kinematic constraints are introduced for contact problems in comparison with the free body problems. Although the general contact problem has been well defined (refer to Appendix B), it has been proven that only very limited and well-formulated contact problems can be solved analytically. A more general and practical approach is to study contact problems numerically.

Because the tactile Jacobian is a means to relate the differential change in chosen features to the differential change in contact, the idea to construct tactile Jacobian numerically is therefore to use tactile sensing to obtain the current state of contact, super-impose the differential changes in all variables of the current contact state, and calculate, using a forward numerical model, the changes in tactile features accordingly. In other words, we can use a numerical model to "observe" or simulate the possible changes in features of the tactile image by virtually applying differential changes in all contact state variables through a numerical forward model, and then record the resulting changes in features with respect to each differential change in contact variables.

6.3.1 Constructing a tactile Jacobian through an FE model

The finite element (FE) method is a powerful tool to model complex physical systems governed by higher order differential or integral equations, such as those for contact problems. The general principles of FE methods have been discussed in many references such as [1]. Procedures to apply FE method to the general contact problems have also been summarized in [60]. Here we will omit some of the details on the general FE procedures, and introduce, from an implementation point of view, the

basic steps to use a FE model to generate the tactile Jacobian in a numerical form.

The ideal case for tactile-servo is to generate the tactile Jacobian for the feedback contact state (that is obtained through tactile sensing) in real-time. Such a process may be broken down into the following steps:

•Step one: Pre-processing

Most commercial FE packages provide convenient geometry modelling tools, usually in graphical form. A mesh generation algorithm is then applied to the model to discretize both bodies in contact and provide input data files (node coordinates, boundary conditions, loading conditions, etc) to the FE solution program. While this approach is convenient for off-line simulation, it is not suitable for real-time contact problems where contact location and load change dynamically. However, such a static model provides a good reference for dynamic modelling. During preprocessing, we first apply such a model based on a reference contact condition to produce the reference input data files.

• Step two: Dynamic loading

In a real-time tactile servo, the sensor position and the contact force are available from the tactile sensor (or together with other sensors). Given this information, the actual loading condition can be computed from the reference data files produced by proper frame transformations and adjustment of the loading factors in step one. Note that the reference data files created in step one need to be read only once and no extra I/O is necessary at run time.

• Step three: Finding the numerical FE solution

In this step we will apply differential changes to all controlled DOF's, and use the FE solution routine to solve the sensing system and produce the numerical stress/strain solution dynamically.

• Step four: Calculating the tactile Jacobian

Once the stress or strain distribution with respect to each differential change is calculated, the corresponding changes in all features can be evaluated and collected

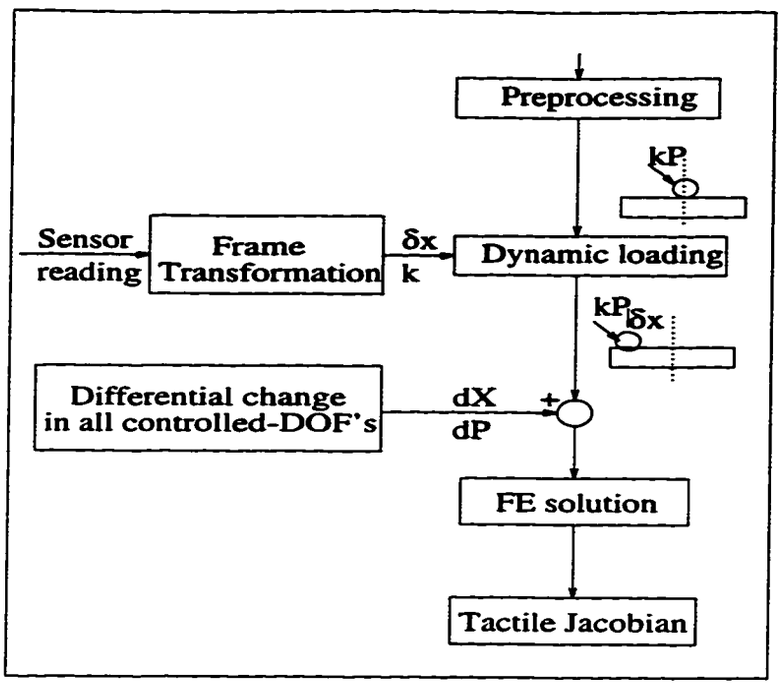


Figure 6.2: *Steps to construct a numerical tactile Jacobian*

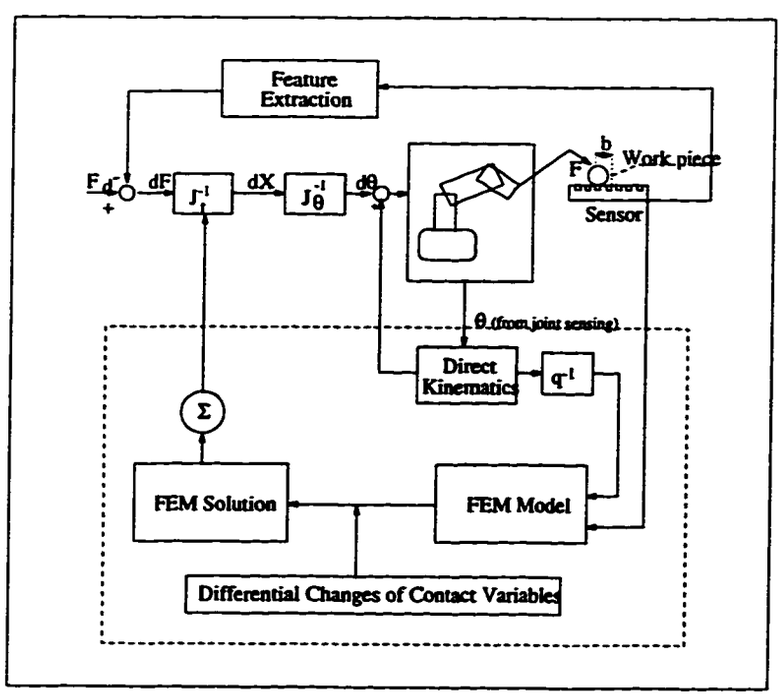


Figure 6.3: *A Tactile-servo scheme using a numerical tactile Jacobian*

into the tactile Jacobian. Figure 6.2 illustrates the basic steps discussed before. Figure 6.3 illustrates the block diagram of the control scheme when using such a procedure to obtain a numerical tactile Jacobian on-line.

The main focus of the previous procedure is to reduce the amount of on-line calculation involved. We have experimented such a procedure for a very simple case of a point contact on a spherical tactile sensor. For this simple example, however, the core FE program takes seconds or longer to converge to a solution. In other words, with the computing power available today, it is impractical to construct a tactile Jacobian in real-time through a FE model.

Fortunately, given a specific robot arm or a dextrous hand, together with the object it manipulates, the possible force and torque ranges and the contact positions at the robot end-effector or different parts of a mechanical hand are often predictable or known in advance. This enables us to discretize the task space, construct a tactile Jacobian for each contact state in advance and store the tactile Jacobian matrices for each contact state in a look-up table. During task execution, the proper tactile Jacobian can be retrieved from the look-up table according to the actual feedback features derived from the tactile sensing. Obviously, while the off-line "tactile planning" may suffer from inaccurate tactile Jacobian "interpolation" during task execution, it has the advantage of high efficiency for real-time control.

6.4 Using the control scheme: a surface tracking example

To further elucidate some of the concepts described in the previous section, let us consider a surface tracking task. Through this example we would like to illustrate the basic steps involved in applying the direct touch-driven control scheme.

Suppose a robot equipped with a spherical end-effector (the probe) is to track the surface of an unknown object in 3-D. A single tactile array sensor is mounted on the end-effector. The control objective is to maintain a fixed contact location on

the tactile sensor, and move the sensor at a constant tracking speed $[v_x, v_y, 0]^T$ in the tangential plane of the object surface. The object is unknown to the robot but the geometry of the tactile sensor attached to the robot is known.

6.4.1 Basic steps

We may first decompose the control into two steps. In step one, we always exert joint commands corresponding to the desired tracking speed $V_d = [v_x, v_y, 0]^T$ in the tangential plane of the object. The corresponding joint motion is therefore

$$d\theta_d = J_\theta^{-1} dX_d = J_\theta^{-1} V_d dt \tag{6.5}$$

where J_θ is the robot Jacobian.

In the second step, we design a tactile servo loop that is responsible for generating the "correction" motion to compensate the error in tracking speed and contact force. In other words, such a tactile servo scheme will act as a "regulator" rather than a "controller".

Because in most cases, the spherical sensor contacts an object in a small area, we can approximate the contact as point contact between the spherical probe and the object. From the model given in chapter 2, we know that, when ignoring the frictional force, the zeroth and first moments are sufficient to recover contact force and position information. We therefore choose the first two moments (M_0, M_x, M_y) as the tactile features for this task.

Before actually tracking the surface of an unknown object, a "tactile planning" is necessary. We can numerically or analytically calculate the desired features from the desired contact state using a forward model. We can also "teach" the robot by "showing" it the desired contact position and force, and then record the time history of the desired tactile features corresponding to the desired contact states. The desired contact states, or the desired tactile feature states, are to be repeated through the tracking process. For example, for the surface tracking example, the desired tactile

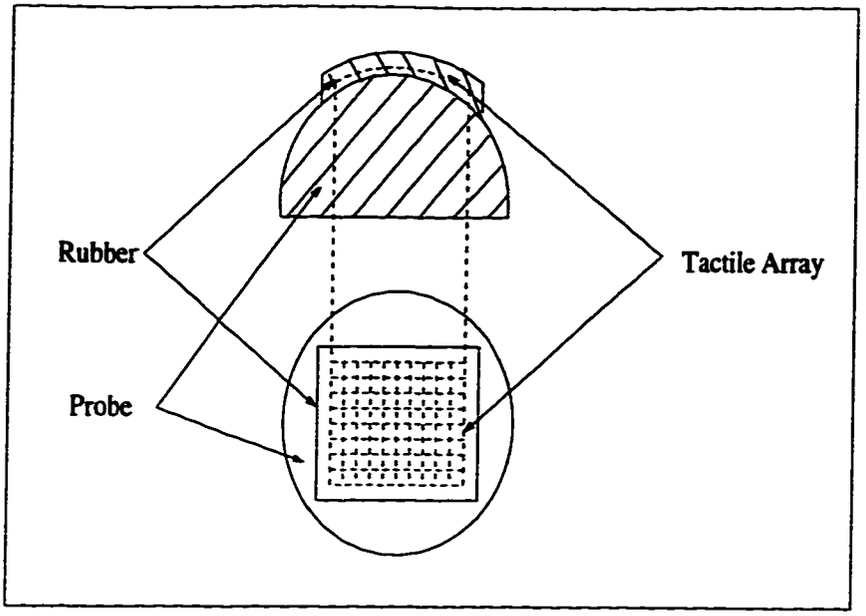


Figure 6.4: *The end-effector and the tactile array sensor.*

feature state can be chosen as a constant state given in the following feature vector:

$$M_d = [M_0, M_x, M_y]^T = [k, 0, 0]^T, \tag{6.6}$$

which corresponds to a constant normal contact force and a desired contact position at the origin of the sensor frame.

6.4.2 Obtaining tactile Jacobian matrices numerically through an FE model

To model the tactile sensing system mounted on top of the spherical end-effector (refer to Figure 6.5 of the real sensor system), a 3-D FE model, shown in Figure 6.6, is created. A total of 2500 3-D elements were used for this model. The bottom surface of the rubber pad is assumed to be glued to the rigid end-effector surface, and such a boundary condition is illustrated in Figure 6.7, where all DOF's of the bottom surface nodes are constrained.

The loading conditions of the FE model depend on the discretization of the task space. For this example, we divide the sensor surface into small areas and consider 6×6 possible contact locations evenly spaced over the sensor surface, as shown in

Figure 6.5: *The actual end-effector and the tactile sensor system*

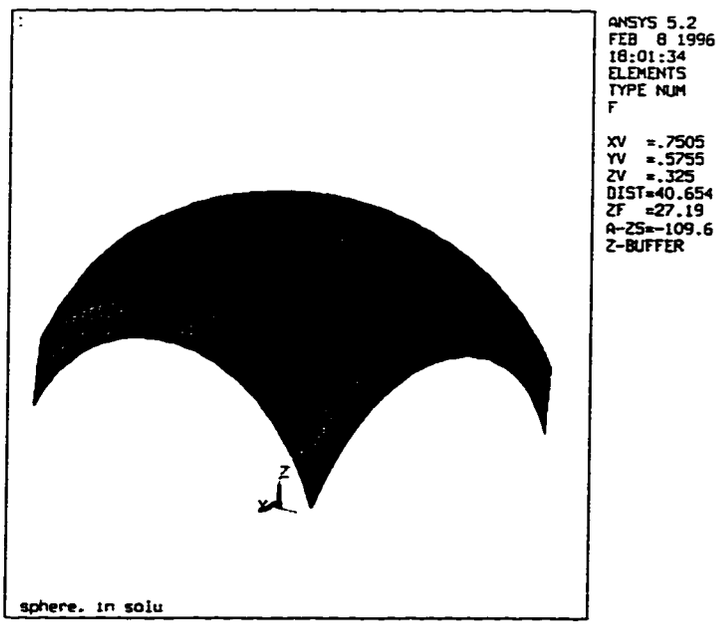


Figure 6.6: *The FE model. Only the rubber layer is modelled. The tactile array is embedded in the middle layer of the rubber skin.*

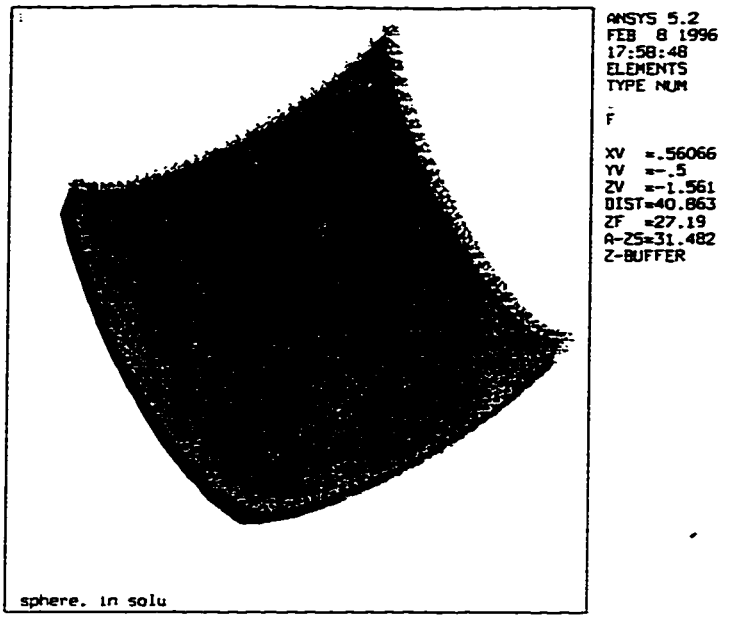


Figure 6.7: *The boundary condition in the FE model. All DOF's are constrained at the bottom layer nodes.*

$(x, y, p+dp)$	$(x, y, p-dp)$
$(x+dx, y, p)$	$(x-dx, y, p)$
$(x, y+dy, p)$	$(x, y-dy, p)$

Table 6.1: *The addition loading cases at position (x, y) with a normal force p.*

Figure 6.8.

At each contact state, defined by contact location (x, y) and contact force p, we examine the changes in features corresponding to the differential changes in contact location, $\pm dx, \pm dy$, and also the change in contact force, $\pm dp$. Table 6.1 shows the combinations of the additional loading conditions applied at each base contact state (x, y, p) when obtaining the tactile Jacobian numerically.

Friction is also modelled in the FE model. Coulomb friction is considered when the differential sliding dx and dy is applied to the base contact position. The direction of the friction is always opposite to the direction of the differential sliding dx and dy. Since the Coulomb friction is calculated from the normal contact force, it is not

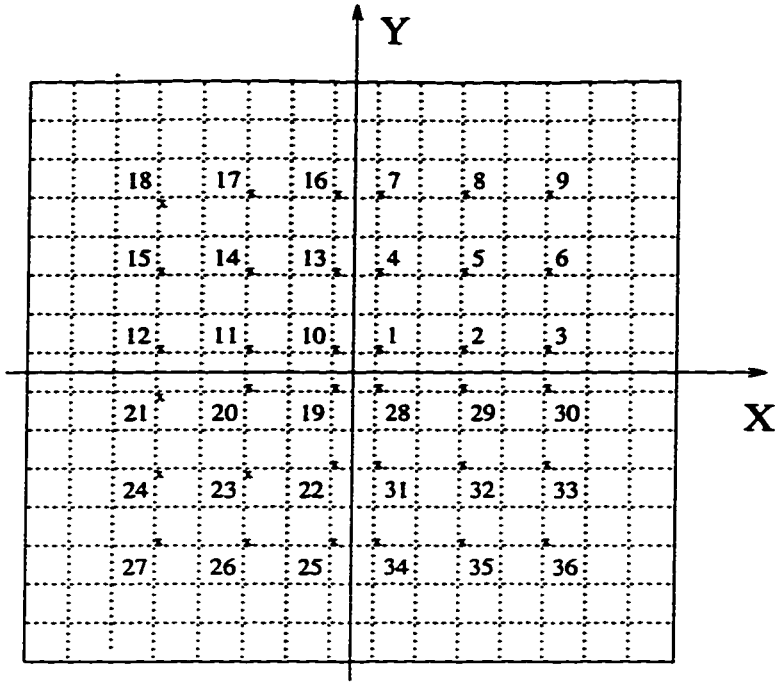


Figure 6.8: *The loading positions (top view of the sensor area on the spherical surface).*

considered as an independent contact parameter.

Due to the symmetric nature of the spherical end-effector, only a quarter of the sensor area needs to be examined through the FE model. This gives us a total of $9 \times 7 = 63$ loading cases (1 base loading case plus 6 loading cases given in Table 6.1 at each location), in order to obtain the tactile Jacobians at 36 locations over the sensor surface.

The tactile Jacobian at each base contact state (x, y, p) can be calculated in the following steps:

1. At each contact state $C = (x, y, p)$, six differential changes $dC_i = [dx_i, dy_i, dp_i]^T$ are applied, respectively.

2. The change in feature state $dF_i = [dM_{0i}, dM_{xi}, dM_{y_i}]^T$ is then calculated from the FE solution by subtracting the image features from those of the base contact state (x, y, p).

3. According to the definition of the tactile Jacobian, we have for each loading

case

$$dF_i = J_t dX_i \tag{6.7}$$

where J_t has a dimension of 3×3, and

$$dX_i = dC_i/K = [dx, dy, dp/K_z]^T.$$

K_z is the stiffness factor of the sensor in the z direction.

Substitute dC_i and dF_i for each differential change into (6.7), we have six (3×3) matrix equations containing the tactile Jacobian J_t at the base contact state C, or 12 linear equations containing 9 elements of J_t. The tactile Jacobian J_t for this contact state can therefore be solved from these equations.

The detailed procedures to obtain a tactile Jacobian using the least squares method, and also the resulting tactile Jacobians for the surface tracking example are given in Appendix C. Note that because J_t^{-1} is used to calculate the final robot joint command, we solve for J_t^{-1} directly instead of for J_t. The inversion sign in J_t^{-1} is therefore symbolic and the numerical inversion of a tactile Jacobian matrix is not necessary.

6.5 Experiments

Surface tracking experiments were performed to verify the tactile servo scheme. In the experiments, the inverse of the tactile Jacobian matrices obtained in Appendix C is stored in a lookup table and retrieved at run-time. During table look-up, the tactile Jacobian J_t^{-1} for the closest "point" in the feature space (defined by (M_0, M_x, M_y), in this example) is selected and used to calculate the robot joint motion.

To improve the efficiency of the table-lookup, hashing technique [23] was used. The Jacobian matrices are indexed by

$$r^2 = M_0^2 + M_x^2 + M_y^2,$$

Figure 6.9: *Surface tracking experiment. The object is unknown to the robot.*

i.e., by the square of the distance from each feature point (m_0, m_x, m_y) to the origin of the feature space. A hash table is created from this index and every "empty slot" in the hash table is pre-sorted to point to the previous and next non-empty "slot". During table-lookup, the index r^2 of the feedback tactile image is first evaluated and used to hash to the corresponding memory location. A search towards both lower and higher memory locations (or the inner and outer space separated by a spherical surface of radius r in the feature space) is then performed. The search stops when the closest feature state in the existing look-up table is found, or a maximum number of search steps has been reached. In the latter situation, the last tactile Jacobian is used. Note that the index r^2 is only used for hashing, not for the selection of the Jacobian matrix. The tactile Jacobian stored in the lookup table is selected through the difference, or distance in the feature space, between the feedback feature vector and each of the feature vectors in the lookup table.

As an example, suppose Jacobians for the following three feature states are to be stored in a look-up table.

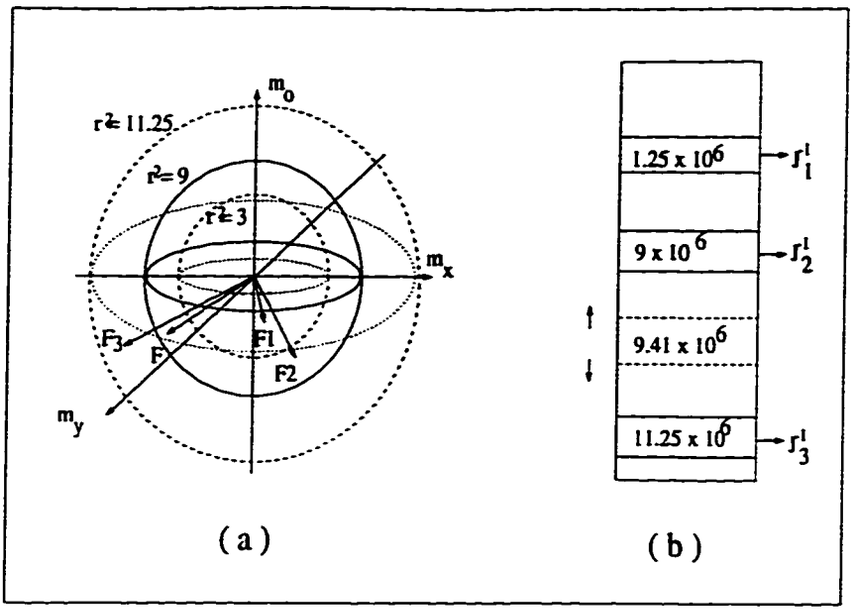


Figure 6.10: *Example of the hash table and the search in the feature space.*

$$F_1 = (M_0, M_x, M_y)^T = (-1000, 1000, 1000)^T,$$
$$F_2 = (M_0, M_x, M_y)^T = (-1000, 2000, 2000)^T,$$
$$F_3 = (M_0, M_x, M_y)^T = (-1000, -2500, 2000)^T.$$

The indices for these three feature states are $r_1^2 = 3 \times 10^6$, $r_2^2 = 9 \times 10^6$ and $r_3^2 = 11.25 \times 10^6$, respectively. The three feature states corresponds to the points F_1, F_2 and F_3 in the feature space, as shown in Figure 6.10(a). After indexing and sorting, the tactile Jacobian is stored in the memory locations shown in Figure 6.10(b).

If during task execution, a feature state $(-1000, -2100, 2000)^T$ is obtained from the tactile feedback, then the index for the current contact state is obtained as

$$r^2 = (-1000)^2 + (-2100)^2 + 2000^2 = 9.41 \times 10^6.$$

A search is therefore initiated at a spherical surface at radius $r = \sqrt{9.41} \times 10^3$ in the feature space. The search will be performed towards both the inner and outer feature space separated by the spherical surface with a radius $\sqrt{9.41} \times 10^3$. The inner and outer space in the feature space corresponds to lower and higher memory in the hash-table.

Suppose we set the search stop criterion as $\epsilon = 0.5 \times 10^6$. Then, for this case, J_1 will be rejected because the squared distance between the feedback feature point F and the feature point F_1 in the feature space is larger then the stop criterion, i.e.,

$$| \vec{F} - \vec{F_2} |^2 = [-1000-(-1000)]^2 + [-2100-2000)]^2 + [2000-2000]^2 = 16.81 \times 10^6 > \epsilon;$$

J_2 will be selected because the squared distance between F and F_2 is smaller than the given selection criterion, i.e, $| \vec{F} - \vec{F_3} |^2 = 0.16 \times 10^6 < \epsilon$.

The experimental apparatus, for which a picture shown in Figure 6.9, is the same as that used in the previous chapter, except that, internally, the pre-sorting of the hash-table, feature extraction, feature comparison, table look-up, and calculation of dX in (6.3) are now implemented on Synergey, a 68040-based single board computer responsible for tactile data processing (refer to appendix A for the architecture of the experimental system).

Two groups of objects are selected for the experiments: cylindrical objects and spherical objects. Two representative objects of different geometry are chosen for each group. Cylindrical and spherical objects are selected because they are easy to measure and compare with the experimental results. In the experiments all objects are unknown to the robot.

Figure 6.11 and Figure 6.12 illustrate the trajectories of the contact point when cylindrical objects of different radii were probed. The trajectories are obtained by combining robot direct kinematics [59] and the relative positions of the contact trajectory measured through tactile sensing.

The first object in the experiment is a small object (a cola can), roughly 65 mm in diameter. The second is a larger object (part of an electrical motor), roughly 120 mm in diameter. A least squares fitting of the trajectory indicates that the tracking error is within 3mm in the surface normal direction for both experiments. It is also interesting to note that tracking smaller objects is more difficult than larger ones. Usually tracking of smaller object requires faster adjustments in the orientation of

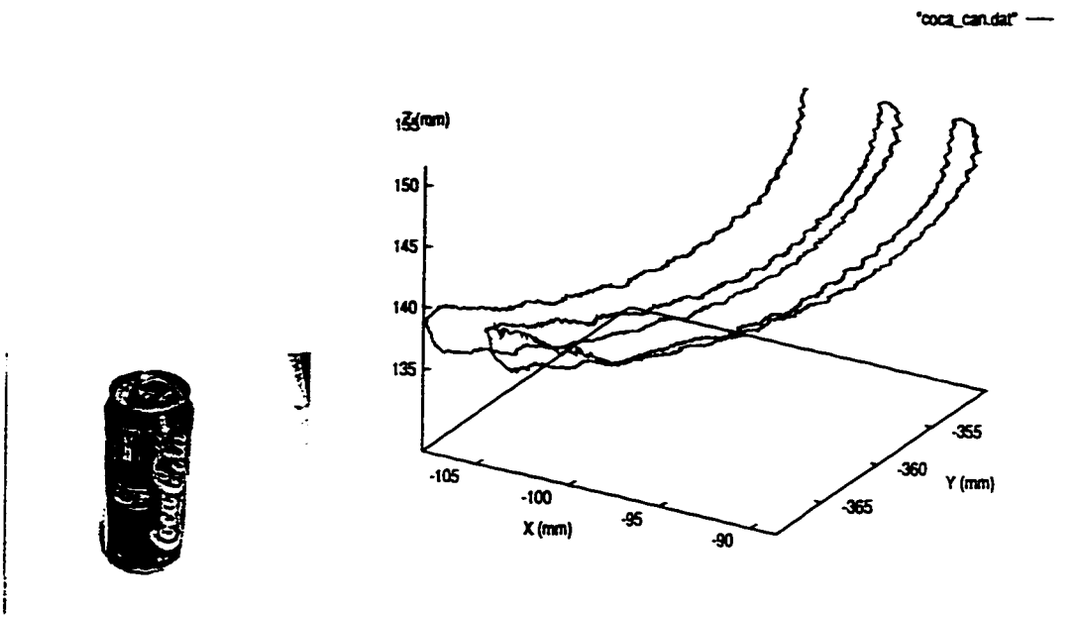


Figure 6.11: *Trajectory of the contact point when probing a small can*

the spherical probe.

Figure 6.13 and Figure 6.14 are the recorded trajectories of the contact point when tracking spherical objects of different radii. The first object is a smaller light bulb, about 70 mm in diameter. The second object is a larger light bulb, roughly 126 mm in diameter. Both object are fragile and require "gentle" touch during surface probing. The tracking error is within 3mm in the surface normal direction for these experiments.

6.6 Conclusions

A tactile servo scheme using a forward tactile model and tactile Jacobian is presented in this chapter. The control scheme avoids the inverse modelling problem and can be designed directly from a numerical forward model, such as a finite element model. Because the robot joint commands are calculated from the difference of the desired and the feedback tactile feature, not from the robot contact state (contact position and force) in the task space, the desired feature "trajectory" can be planned directly

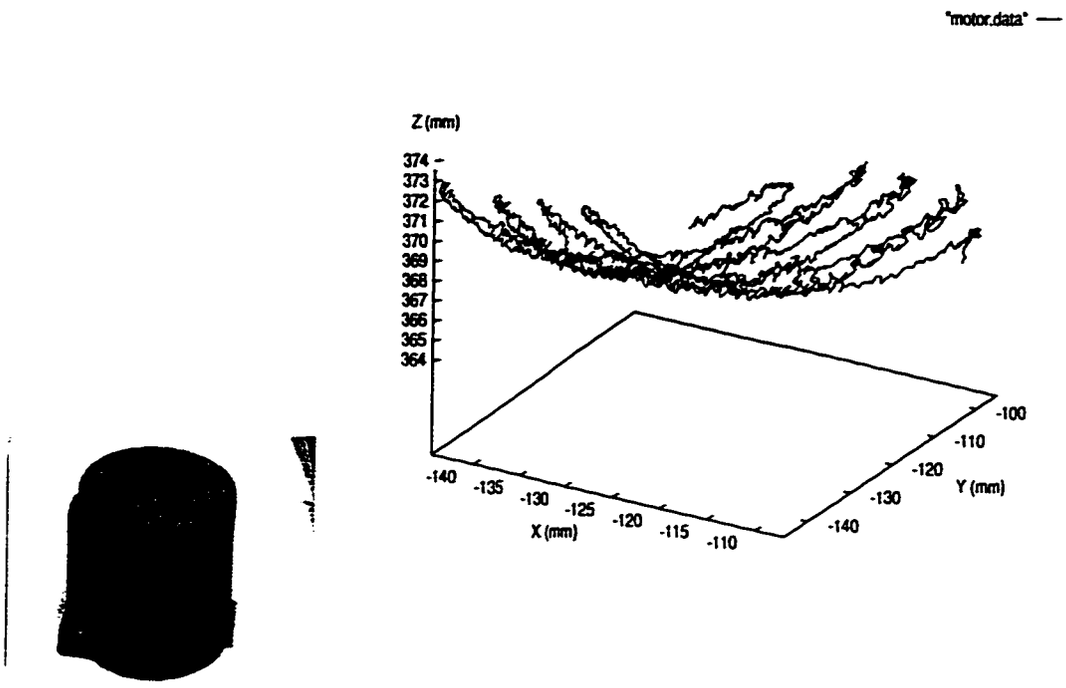


Figure 6.12: *Trajectory of contact when probing a cylinder-like object*

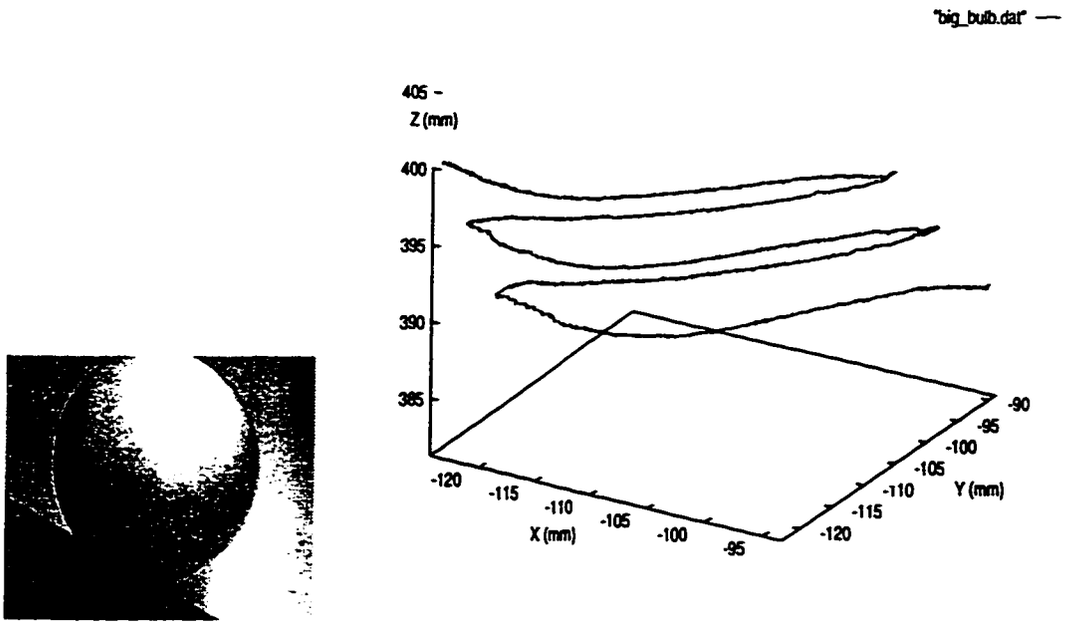


Figure 6.13: *Trajectory of contact when probing a bigger spherical light bulb.*

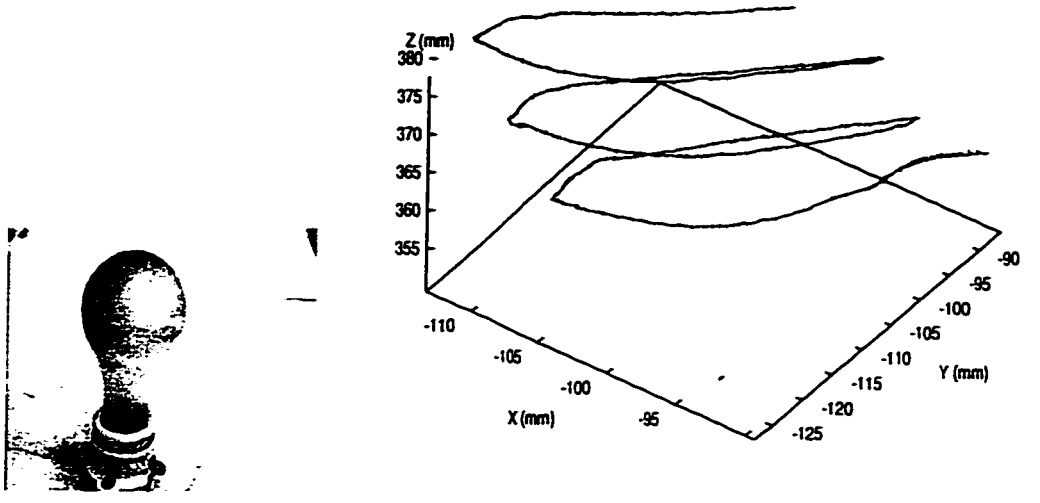


Figure 6.14: *Trajectory of contact when probing a smaller light bulb.*

in the tactile feature space. This is desirable in the situations where obtaining tactile feature and then going back to the task space is a noisy and computationally expensive process.

The proposed tactile servo scheme was demonstrated and tested through a surface tracking example, which is another important task primitive in object exploration and manipulation besides edge tracking. Real-time experiments were performed using this control scheme. Our experience proves that this approach is easy to implement and also reliable.

Chapter 7

Summary

Although research in tactile sensing and its applications have scarcely come out of their infancy, it has become necessary and possible to study tactile sensing in a more systematic manner.

In this thesis, the tactile data processing problem was first investigated. We have studied the inverse modelling using simplified point and line contact models based on elasticity. The information obtained from these simple models can be further used to derive local object shape information under a kinematic framework.

In addition to modelling of tactile sensing, another focus of the thesis is to provide general methodologies to integrate robot tactile sensing with low-level robot control, i.e, the direct touch-driven control or tactile servo. Direct touch-driven control is an important step toward more complicated robot control applications. The direct touch-driven control schemes we presented effectively integrate tactile sensing with low level robot control. It provides a basic relationship between tactile sensing and robot motion. Such a basic relationship would allow the establishment of an effective physical and functional organization and it is a fundamental step towards building intelligent robots that are capable of autonomous perception of their environment. In this thesis, we have provided two basic tactile servo schemes, demonstrated the basic steps to apply these control schemes, and supported these schemes with extensive experiments.

This thesis contributes to the research in robotics and other areas relevant to artificial tactile sensing in the following aspects:

1. Previous studies on the "inverse tactile model" were mostly either numerical or based on non-model methods such as neural networks. In this thesis, we have designed analytical inverse models for line and point contact. These models have proven to be suitable for real-time robot control applications.

2. We have provided an algorithm to recover local object shape information from tactile sensing and robot kinematics. This kinematics-based approach is novel. The surface matrix we introduced in deriving the algorithm facilitates the convenient treatment of 3-D surface geometry, contact constraints and robot kinematics using homogeneous transformations, and can be used to analyze other robot contact kinematics problems.

3. The framework on direct touch-driven robot control has been presented, which is an abstraction and extension of the early concepts from [13], [49] and [51]. Two basic control schemes, using an inverse tactile model and using a tactile Jacobian, are presented and verified with extensive experiments. These basic direct touch-driven schemes provide a foundation toward performing more complex motion procedures such as grasping and manipulation.

This thesis has studied tactile sensing and provided basic schemes for touch-driven robot control. A natural extension of this thesis is to apply modern control theories to the tactile servo problem. Modern control theories have enjoyed much success in vision-based control (see, e.g., [21] [56] [43]). We believe there is a great potential for modern control theories to improve the performance of the current pure kinematics-based tactile servo schemes.

Another issue arising in tactile servo is the choice of tactile features. The choice of a set of tactile features that uniquely characterize the contact states is the key to the

success of touch-based control. However, such a feature set is sometimes difficult to choose in practice. It is also possible that the proper choice of features may depend on the states of the contact, and this further complicates the problem. It is necessary, in the next stage of tactile-servo study, to further investigate the mapping from the robot task space (or the state of the contact) to the tactile feature space, and to provide general guidelines for optimal feature selection.

Bibliography

[1] K. J. Bathe. *Finite element procedures in engineering analysis.* Prentice-Hall, Englewood Cliffs, 1982.

[2] A. D. Berger and P. K. Khosla. Using tactile data for real-time feedback. *International Journal of Robotics Research*, 10(2):88–102, 1991.

[3] A. D. Berger and P. K. Khoslar. Edge dectection for tactile sensing. In *Proc. NASA conf. on space telerobotics*, pages 163–172, 1989.

[4] A.D. Berger and P. K. Khosla. The modified adaptive hough transform (maht). *Journal of Robotic Systems*, pages 277–290, 1990.

[5] B Borovac, M Nikolic, and L Nagy. A new type of force sensor for contact tasks. *94'IEEE International Conf. on Robotics and Automation*, 2:1791–1796, 1994.

[6] O. Bottema and B. Roth. *Theoretical Kinematics.* Elsvier, Amsterdam, 1979.

[7] C. Cai and B. Roth. On the planar motion of rigid bodies with point contact. *Mechanism and Machine Theory*, 21(6):453–466, 1966.

[8] C. Cai and B. Roth. On the spacial motion of a rigid body with point contact. In *87'IEEE International Conf. on Robotics and Automation*, pages 686–695, 1987.

[9] A. Cameron, R. Daniel, and H Durrant-Whyte. Touch and motion. In *88'IEEE International Conf. on Robotics and Automation*, pages 1062–1067, 1988.

[10] S.L. Campbell and C.D Meyer. *Generalized inverses of linear transformations.* London:Pitman, 1979.

[11] Ning Chen, Ray E. Rink, and Hong Zhang. Efficient edge detection from tactile data. *95'IEEE/RJS International Conf. on Intelligent Robots and Systems*, 1:386–391, 1995.

[12] Ning Chen, Ray E. Rink, and Hong Zhang. Local surface shape from tactile sensing. *96'IEEE International Conf. on Robotics and Automation*, 1996.

[13] Ning Chen, Hong Zhang, and Ray E. Rink. Edge tracking using tactile servo. *95'IEEE/RJS International Conf. on Intelligent Robots and Systems*, 2:84–99, 1995.

[14] Ning Chen, Hong Zhang, and Ray. E. Rink. Tactile sensing of point contact. *95'IEEE International Conf. on Systems, Man, and Cybernetics*, pages 574–579, 1995.

[15] M. R. Cutkosky and P. K. White. Friction, stability and the design of robotics fingers. *International Journal of Robotics Research*, 5(4):20–37, 1986.

[16] D. De Rossi, A Caiti, R Bianchi, and G Canepa. Fine-form tactile discrimination through inversion of data from a skin-like sensor, 1991.

[17] R. S. Fearing. Simplified grasping and manipulation with dextrous robot hands. In *Proc. American Control Conference*, pages 32–38, 1984.

[18] R. S. Fearing. Tactile sensing mechanism. *International Journal of Robotics Research*, 9(3):3–23, 1990.

[19] R. S. Fearing. Using a cylindrical tactile sensor for determining curvature. *IEEE Transactions on Robotics and Automation*, 7(6):806–817, 1991.

[20] R. S. Fearing and J. M. Hollerbach. Basic solid mechanics for tactile sensing. *International Journal of Robotics Research*, 4(3):40–54, 1985.

[21] J.T. Feddema and C.S.G. Lee. Adaptive image feature prediction and control for visual tracking with a hand-eye coordinated camera. *IEEE Trans. Syst., Man, Cybern.*, 20(5):1172–1183, 1990.

[22] J.T. Feddema, C.S.G. Lee, and O. R. Mitchell. Weighted selection of image features for resolved rate visual feedback control. *IEEE Trans. Robot Automation*, 7(1):31–47, 1991.

[23] M.J. Folk and B. Zoellick. *File Structures.* Addison-Wesley, 2nd edition, 1992.

[24] S. S. Gindy. A new concept for an intelligent touch sensor. *Proc. of Robotic Intelligence and Productivity Conf.*, pages 67–73, 1983.

[25] R.C. Gonzalez and R. E. Woods. *Digital image processing.* Addison-Wesley, 1992.

[26] L. D. Harmon. Tactile sensing for robotics. In M. Brady, L. A. Gerhardt, and H. F. Davidson, editors, *Robotics and artifical intelligence.* Springer-Verlag. 1984.

[27] W.D Hillis. Active touch sensing. Master's thesis, MIT, Dept of Electrical Engineering and Computer Science, 1981.

[28] N. Houshangi and A. J. Koivo. Tactile sensor control for robotic manipulations. In *IEEE 1989 International Conference on Systems, Man, and Cybernetics*, pages 1258–1259, 1989.

[29] R. D. Howe and M. R. Cutkosky. Tactile sensing for robotic manipulation and recognition. In O. Khatib and J. J. Craig, editors, *The Robotics Review 2.* MIT Press, 1992.

[30] J. Illingworth and J. Kittler. The adaptive hough transform. *IEEE PAMI*, 9(5):690–698, 1987.

[31] R.S. Johansson and A.B. Vallbo. Tactile sensory coding in the glabrous skin of the human hand. *Trends in NeuroSciences*, 6(1):27–32, 1983.

[32] K. L. Johnson. *Contact Mechanics*. Cambridge University Press, 1985.

[33] G.I Kinoshita. Classification of grasped object's shape by an artificial hand with multi-element tactile sensors. In *Proc. of IFAC Symp. Info. Control. Prob. in Manufact. Tech.*,, pages 111 - 118, 1977.

[34] R. L. Klatzky, S.J. Lederman, and V.A Metzger. Identifying objects by touch: an expert system. *Perception and Psychophysics*, 7(6):337–341, 1970.

[35] M Kobayashi, T Shinokura, and S Sagisawa. Three-component detection type tactile sensor. *Journal of Robotics and Mechatronics*, 2(1):44–47, 1989.

[36] J.M. Loomis and S.J Lederman. Tactile perception. In K. Boff, L. Kaufman, and J. Thomas, editors, *Handbook of human perception and performance*. Wiley, New York, 1986.

[37] H. Maekawa, K. Komoriya, and K. Tanie. Manipulation of an unknown object by multifingered hands with rolling contact using tactile feedback. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1877–1882, 1992.

[38] P.J McKerrow. *Introduction to Robotics*. Addison-Wesley, 1991.

[39] P. Michelman and P. Allen. Forming complex dextrous manipulations from task primitives. In *94'IEEE International Conf. on Robotics and Automation*, pages 3383–3388, 1994.

[40] D. J Montana. The kinematics of contact and grasp. *International Journal of Robotics Research*, 7(3):17–30, 1988.

[41] C. Muthukrishnan, D. Smith, D. Myers, J. Rebman, and A. Koivo. Edge detection in tactile images. In *87'IEEE International Conf. on Robotics and Automation*, 1987.

[42] H. R. Nicholls and M. H. Lee. A survey of robot tactile sensing technology. *International Journal of Robotics Research*, 8(1):3–30, 1989.

[43] N. P. Papanikolopoulos and P. K. Khosla. Adaptive robotc visual tracking: Theory ad experiments. *IEEE Transactions on Automatic Control*, 38(3):429–444, 1993.

[44] Y.C Pati, D. Friedman, P.S Krishnaprasad, C.T. Yao, and M.C Peckerar. Neural networks for tactile perception. In *88'IEEE International Conf. on Robotics and Automation*, pages 134–139, 1988.

[45] S. L. Ricker and R. E. Ellis. 2-d finite element models of tactile sensors. In *Proceedings of 1993 IEEE International Conference on Robotics and Automation, Atlanta, Georgia*, pages 941–947, Atlanta, Georgia, 1993.

[46] J. K. Salisbury, Jr. Active stiffness control of a manipulator in cartesian coordinates. In *19th IEEE Decision and Control Conference*, 1980.

[47] J. K. Salisbury, Jr. and B. Roth. Kinematic and force analysis of articulated mechanical hands. *Journal of mechanisms, transmissions, and automation in design*, 105(82-DET-13):35–41, 1983.

[48] W. Schiff and E. Foulke. *Tactual Perception: A Sourcebook.* Cambridge University Press, England, 1982.

[49] P. Sikka, H. Zhang, and S. Sutphen. Tactile servo: Control of touch-driven robot motion. *Proceedings of the 3rd International Symposium on Experimental Robotics, Kyoto, Japan*, 1993.

[50] P. Sikka, H. Zhang, and R. W. Toogood. On modeling tactile sensors. *Technical Report TR-08-93, Department of Computing Science, University of Alberta*, 1993.

[51] Pavan Sikka. *The role of touch sensing in robot manipulation.* PhD thesis, Uinversity of Alberta. Dept of Computing Science, 1994.

[52] T. H. Speeter. A tactile sensing system for robotic manipulation. *International Journal of Robotics Research*, 9(6):25–36, 1990.

[53] T. H. Speeter. Three-dimensional finite element analysis of elastic continua for tactile sensing. *International Journal of Robotics Research*, 11(1):1–19, 1992.

[54] M.M. Taylor, S.J. Lederman, and R.H. Gibson. Tactual perception of texture. In E.C Carterette and M.P. Friedman, editors, *Handbook of perception*, volume 3. New York: Academic, 1973.

[55] S. P. Timoshenko and J. N Goodier. *Theory of Elasticity.* McGraw-Hill, 3rd edition, 1979.

[56] L. E. Weiss, A. C. Sanderson, and C. P. Neuman. Dynamic sensor-based control of robot with visual feedback. *IEEE Journal of Robotics and Automation*, 5:404–417, 1987.

[57] C-H Wu and R.P. Paul. Manipulator comliance based on joint torque control. In *19th IEEE Conference on Decision and Control*, pages 88–94, 1986.

[58] H. Zhang and B. Zhang. Improved trajectory generation schemes based on resolved motion rate control. *International Journal of Robotics and Automation*, 10(2):70–77, 1995.

[59] Hong Zhang. *Design and Implementation of A Robot Force And Motion Server.* PhD thesis, Purdue University, School of Electrical Engineering, 1986.

[60] Z. Zhong. *Finite element procedures for contact-impact prolems.* Oxford, 1993.

[61] Z. Zhong and J. Mackerle. Static contact problems - a review. *Engineering Computations*, 9:3–37, 1992.

Appendix A

The real-time robot controller

The diagram of the real-time robot controller for the tactile servo experiments is shown in Fig.A.1. This is a multiprocessor parallel system constisting of three parallel 68000 family boards, D/A converters and A/D converters. A Sun-3 VME bus hosts all three boards. The Sun-3 is also used for displaying and interacting with the user.

The first board, named Heurikon, has a 68020 microprocessor with a 68881 math co-processor and 1 MB memory. The second board, named Synergy, has a 68040 microprocessor with 2 MB memory. The third board, named IOM1, has a 68000 microprocessor, 512 KB memory and a customer-built analog input interface to the tactile and force sensors.

The real-time sensing and control is distributed to the three parallel microprocessors. IOM1 is responsible for collecting and preprocessing data from all sensors (1 force torque sensor, and 2 tactile sensors). Synergy is responsible for tactile data processing. Heurikon is responsible to calculate the robot joint commands from the sensory information. The calculated joint commands are applied to the robot joint motors through the D/A converters. All three boards communicate through the VME bus.

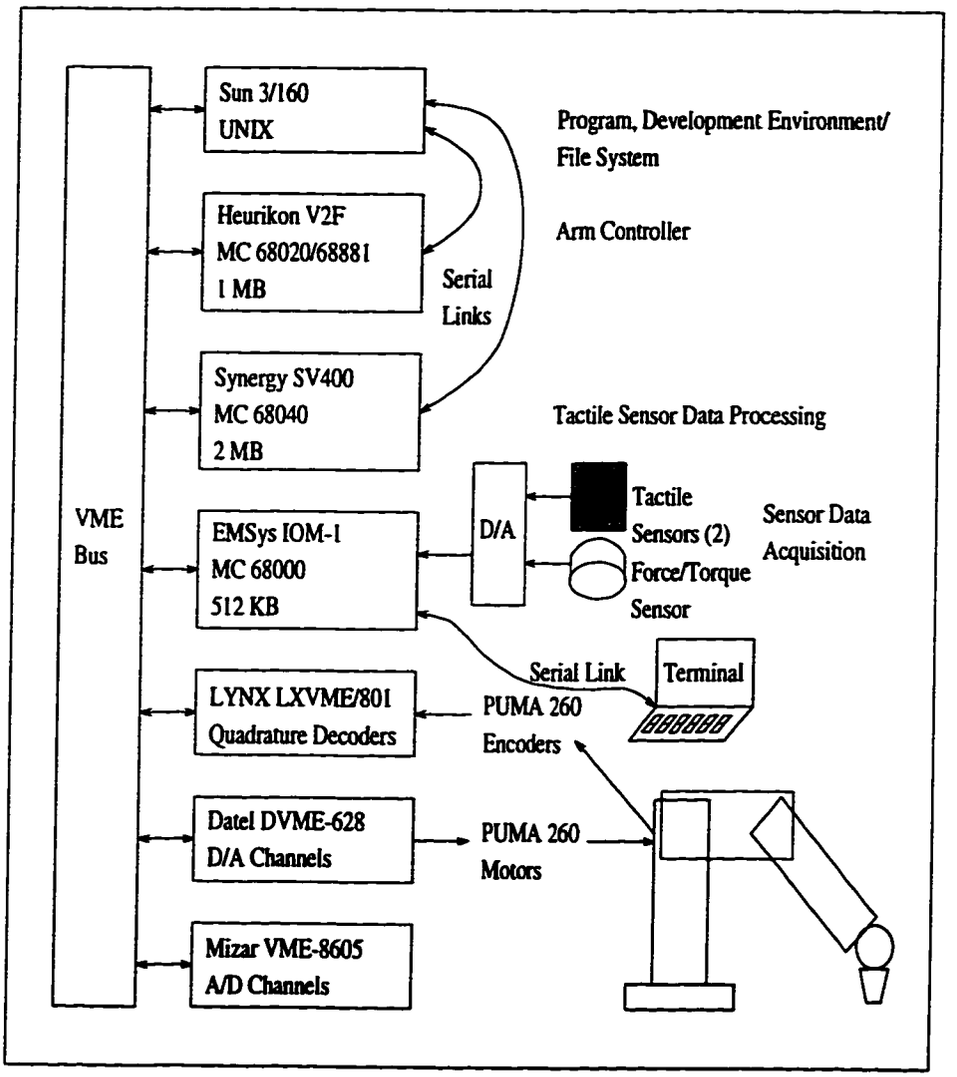


Figure A.1: *The Experiment setup for tactile servo*

Appendix B

An introduction to the general contact problem

Studies on contact problems date back hundreds of years. Newton's third law and Coulomb friction law are the most important early results. Rigid body contact problems such as that depicted in Fig. B.1 can be formulated through these basic laws, and global phenomena such as the total contact force can be solved.

While Newton's and Coluomb's law can be used to analyze the global behavior of contact, they are sometimes not sufficient in many other situations where we are also interested in local behavior of contact such as how force is distributed on the contact surface. Studies of local phenomena, such as stress distribution on the contact surface, began in the 1880s. Among these studies, the most successful and, until today, still widely-used are the studies on the Hertz contact problems. Hertz first simplified the contact bodies to elastic half-space with small deformation and small contact areas. He also assumed that the contact area is frictionless. Following Hertz's original work, many other researchers have studied contact problems between elastic bodies of different shapes and under different circumstances [32]. A common limitation in these studies, however, is that "*the geometry and deformation of a contact body is assumed in such a way that available mathematical and mechanical tools can be used to obtain a closed-form solution of the problem... The approaches used in this stage are obviously very restrictive and can only be applied to very special problems*"[60].

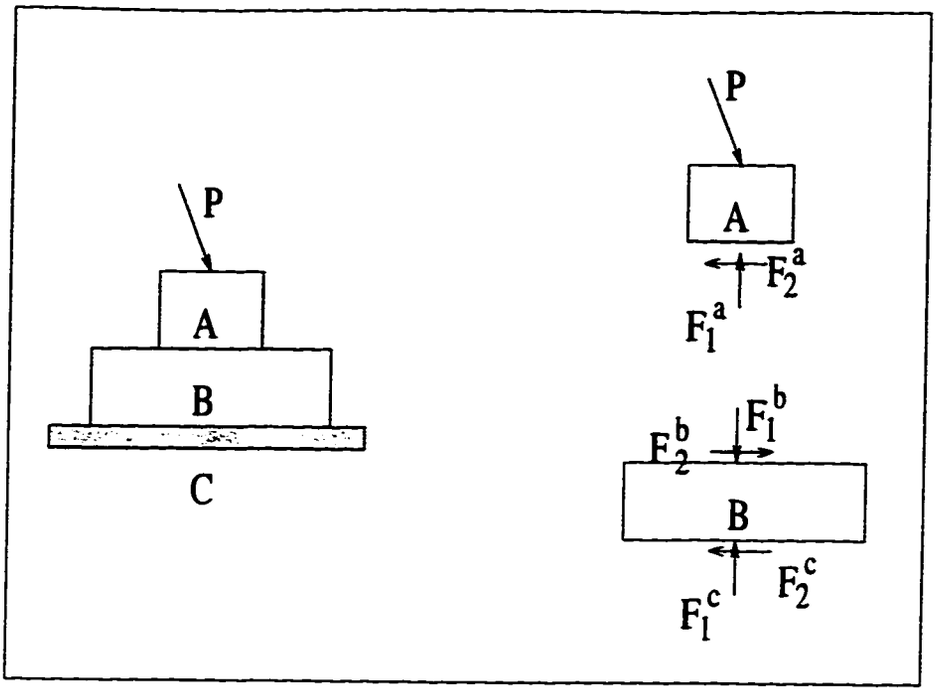


Figure B.1: *A contact system where object A and B are elastic, object B is rigid, and P is the external force applied. The system can be decomposed into free bodies and solved by Newton's and Coulomb's law.*

The third stage in the study of contact problems is the numerical study stage. Among the different numerical methods, the finite element (FE) method has been most widely accepted in studying contact problems. In the FE method, contact bodies are approximated by connected small elements, and the contact bodies can have complicated geometries and material properties. When formulating the mathematical equations for FE solution, no simplifications, such as those taken in the Hertz contact problems, needed to be applied. Only the most fundamental laws, such as the equation of motion, constitutive equations, and the basic contact constraints, are considered. The general contact problem has been introduced in [60] using the following notations:

q - boundary traction vector.

Ω - space taken by the interior particles of a body.

Γ - space taken by a boundary surface.

b - body force vector.

u - displacement vector with component u_1, u_2,u_2.

σ - Cauchy stress tensor.

ρ - mass density of material.

a - acceleration vector.

s- second Piola-Kirchhoff stress tensor.

ϵ - Green-Lagrange strain tensor.

v - velocity vector with components v_1, v_2 and v_3.

N - boundary unit vector.

Also, the left superscripts in the following equations are the time indexes. Summation convention is used for the following discussion.

The general contact problem can be mathematically stated as follows:

Given ${}^tQ$ on ${}^0\Gamma$ and ${}^tb$ on ${}^0\Omega$, $t \in [0,T]$, find $u(x,t)$ for all $t \in [0,T]$ such that all the following conditions are satisfied:

1. The equation of motion:

$$\partial^t\sigma_{ij}/\partial^t x_j +^t b_i =^t \rho^t a_i \qquad on^t\Omega \tag{B.1}$$

2. The constitutive equations (the generalized Hooke's law):

$$^ts_{ij} = c_{ijkl}.^t\epsilon_{kl} \qquad on\ ^t\Omega. \tag{B.2}$$

where ${}^ts_{ij}$ are material parameters.

3. The initial conditions:

$$u(X,0) = u$$

$$v(X,0) = v, \qquad both\ on\ {}^0\Omega. \tag{B.3}$$

4. The boundary conditions:

$$u(X,t) =^t u \qquad on\ {}^0\Gamma_D \tag{B.4}$$

$$ {}^t\sigma_{ij}.{}^tN_{1j} = {}^t q_i \qquad on\ {}^t\Gamma_F \tag{B.5} $$

where Γ_D and ${}^t\Gamma_F$ are the parts of the boundary where the displacements are prescribed and the forces are prescribed, respectively. In general, there exists

$$ \Gamma = \Gamma_D + \Gamma_F + \Gamma_C, $$

where Γ_C is the part of the boundary where contact may occur.

5. The contact constraints:

(1) The normal traction ${}^tq_1$ must be compressive.

$$ {}^tq_1 \leq 0, \qquad on {}^t\Gamma_C^1 \cup {}^t\Gamma_C^2. \tag{B.6} $$

(2) The contact boundary should not penetrate into the ground.

$$ p({}^tx, t) = 0, \qquad {}^tx \in {}^t\Gamma_c^1. \tag{B.7} $$

where $p({}^tx, t)$ is the penetration function [60].

(3). Coluomb's law holds on the contact boundary.

$$ {}^tq_t \leq v\ {}^tq_1, \tag{B.8} $$

$$ {}^tv_t = 0 \qquad if\ {}^tq_t < v{}^tq_1, \tag{B.9} $$

$$ {}^tv_t = -\lambda {}^tq_t^2 \qquad if\ {}^tq_t = v{}^tq_1, \tag{B.10} $$

$$ {}^tv_t = 0 \qquad if {}^tq_t < v{}^tq_1, \tag{B.11} $$

where v is the friction coefficient.

(4) No normal stress on the free surfaces.

$$ {}^t\sigma_{ij}^t N_{1j} = 0 \qquad on\ {}^t\Gamma_0 - {}^t\Gamma_c^1 \cup {}^t\Gamma_C^2 \tag{B.12} $$

Equation (B.1) to (B.12) are the basic equations governing the contact problems. A closed-form solution to these equations is beyond the mathematics available to date; however, these equations can be further formulated into variational equations which are convenient for FE solution. The details on the formulation of the variational equations can be found in [60].

Appendix C

A least square solution to tactile Jacobian matrices

Rewrite equation (6.7) to the following form suitable for least square solution.

$$dX_i = A_i J_t^{-1}, \tag{C.1}$$

where

$$J_t^{-1} = [J_{11}, J_{12}, J_{13}, J_{21}, J_{22}, J_{23}, J_{31}, J_{32}, J_{33}], \tag{C.2}$$

$$dX_i = [dx, dy, dp/k_z]^T, \tag{C.3}$$

$$dF_i = [dm_{0i}, dm_{xi}, dm_{y_i}]^T \tag{C.4}$$

$$A_i = \begin{pmatrix} dm_{xi}, dm_{yi}, dm_{0i}, 0, 0, 0, 0, 0, 0 \\ 0, 0, 0, dm_{xi}, dm_{yi}, dm_{0i}, 0, 0, 0 \\ 0, 0, 0, 0, 0, 0, dm_{xi}, dm_{yi}, dm_{0i} \end{pmatrix}. \tag{C.5}$$

J_{ij}, (i = 1..3, j = 1..3), in (C.2) are the elements in J_t^{-1}. k_z in (C.3) is the robot stiffness factor in the z direction in the sensor frame.

Assemble equation (C.1) for all 6 differential motions into the following single matrix equation:

$$dX = AJ_t^{-1}, \tag{C.6}$$

where

$$dF = [dF_1, dF_2, ..., dF_6]^T \tag{C.7}$$

$$A = [A_1, A_2, ..., A_6]^T, \tag{C.8}$$

$$dX = [dX_1, dX_2, ..., dX_6]^T \tag{C.9}$$

The least square solution for J_t^{-1} can be obtained as,

$$J_t^{-1} = (A^T A)^{-1} A^T dX. \tag{C.10}$$

The following is the resulting least square solutions for J_t^{-1} at all chosen contact states, depicted in Fig. 6.8. The base contact force chosen to generate the tactile Jacobians is $800 Newton$, and a differential change of $\pm 200 Newton$ is applied. The differential change in contact location is taken as ± 1 *tactel*.

At point 1, 10, 19, 28:

$$J_t^{-1} = \begin{bmatrix} 0.11 \times 10^{-2} & 0.15 \times 10^{-5} & 0.15 \times 10^{-5} \\ 0.91 \times 10^{-3} & -0.50 \times 10^{-7} & -0.11 \times 10^{-2} \\ 0.91 \times 10^{-3} & -0.11 \times 10^{-2} & -0.50 \times 10^{-7} \end{bmatrix}.$$

At point 2, 11, 20, 29:

$$J_t^{-1} = \begin{bmatrix} 0.11 \times 10^{-2} & 0.74 \times 10^{-5} & 0.15 \times 10^{-5} \\ 0.91 \times 10^{-3} & -0.48 \times 10^{-6} & -0.11 \times 10^{-2} \\ 0.45 \times 10^{-2} & -0.11 \times 10^{-2} & -0.18 \times 10^{-8} \end{bmatrix}.$$

At point 3, 12, 21, 30:

$$J_t^{-1} = \begin{bmatrix} 0.78 \times 10^{-4} & 0.79 \times 10^{-6} & 0.10 \times 10^{-6} \\ 0.92 \times 10^{-2} & -0.15 \times 10^{-5} & -0.11 \times 10^{-2} \\ 0.94 \times 10^{-2} & -0.11 \times 10^{-2} & 0.16 \times 10^{-5} \end{bmatrix}.$$

At point 4, 13, 22, 31:

$$J_t^{-1} = \begin{bmatrix} 0.11 \times 10^{-2} & 0.15 \times 10^{-5} & 0.74 \times 10^{-5} \\ 0.46 \times 10^{-2} & -0.18 \times 10^{-6} & -0.11 \times 10^{-2} \\ 0.91 \times 10^{-3} & -0.11 \times 10^{-2} & -0.48 \times 10^{-6} \end{bmatrix}.$$

At point 5, 14, 23, 32:

$$J_t^{-1} = \begin{bmatrix} 0.11 \times 10^{-2} & 0.72 \times 10^{-5} & 0.72 \times 10^{-5} \\ 0.46 \times 10^{-2} & -0.19 \times 10^{-5} & -0.11 \times 10^{-2} \\ 0.46 \times 10^{-2} & -0.11 \times 10^{-2} & -0.19 \times 10^{-5} \end{bmatrix}.$$

At point 6, 15, 24, 33:

$$J_t^{-1} = \begin{bmatrix} 0.77 \times 10^{-4} & 0.78 \times 10^{-6} & 0.50 \times 10^{-6} \\ 0.46 \times 10^{-2} & -0.63 \times 10^{-3} & -0.11 \times 10^{-2} \\ 0.94 \times 10^{-2} & -0.11 \times 10^{-2} & 0.67 \times 10^{-5} \end{bmatrix}.$$

At point 7, 16, 25, 34:

$$J_t^{-1} = \begin{bmatrix} 0.78 \times 10^{-4} & 0.10 \times 10^{-6} & 0.79 \times 10^{-6} \\ 0.94 \times 10^{-2} & 0.16 \times 10^{-5} & -0.11 \times 10^{-2} \\ 0.92 \times 10^{-3} & -0.11 \times 10^{-2} & -0.15 \times 10^{-5} \end{bmatrix}.$$

At point 8, 17, 26, 35:

$$J_t^{-1} = \begin{bmatrix} 0.77 \times 10^{-4} & 0.51 \times 10^{-6} & 0.78 \times 10^{-6} \\ 0.94 \times 10^{-2} & 0.66 \times 10^{-5} & -0.11 \times 10^{-2} \\ 0.46 \times 10^{-2} & -0.11 \times 10^{-2} & -0.63 \times 10^{-5} \end{bmatrix}.$$

At point 9, 18, 27, 36:

$$J_t^{-1} = \begin{bmatrix} 0.39 \times 10^{-4} & 0.38 \times 10^{-6} & 0.38 \times 10^{-6} \\ 0.88 \times 10^{-2} & -0.10 \times 10^{-5} & -0.11 \times 10^{-2} \\ 0.88 \times 10^{-2} & -0.11 \times 10^{-2} & -0.10 \times 10^{-5} \end{bmatrix}.$$