Tactile Sensing and Control of a Planar Manipulator

by

Edward John Nicolson

B.S. (Princeton University) 1987
M.S. (University of California at Berkeley) 1990

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Engineering-Electrical Engineering and Computer Sciences

in the

GRADUATE DIVISION
of the
UNIVERSITY of CALIFORNIA at BERKELEY

Committee in charge:

Professor Ronald S. Fearing, Chair
Professor John F. Canny
Professor Masayoshi Tomizuka

1994

The dissertation of Edward John Nicolson is approved:

Chair Date

Date

Date

University of California at Berkeley

1994

Tactile Sensing and Control of a Planar Manipulator

by

Edward John Nicolson

Abstract

Tactile Sensing and Control of a Planar Manipulator

by

Edward John Nicolson

Doctor of Philosophy in Engineering - Electrical Engineering and Computer Sciences

University of California at Berkeley

Professor Ronald S. Fearing, Chair

This dissertation explores the shape sensing capabilities of cylindrical tactile sensing fingers. Starting with an elastostatic model for the deformation of rubber fingers, sensor spacing and depth requirements are determined to allow reconstruction of subsurface strain fields with insignificant aliasing. Given this bandlimited version of the strain field, theoretical limits are found to classification and scaling of the perceived indentation. These theoretical results lead to the design of a silicone rubber tactile sensor which is characterized and calibrated to the model. The reliability of curvature estimates from the sensor is then determined. Finally, use of the sensor during manipulation is demonstrated.

A spatial frequency domain model for the deformation of an elastic cylinder with a rigid core in plane strain is derived. Based on the transfer function from surface pressure to subsurface strain, several conclusions can be made about bandlimited tactile sensing. First, we show that shear strain measurements are not useful for shape estimation. Secondly we show that a ratio of core radius to outer radius greater than 0.85 is required for indenter classification given sensor noise of 1.7% peak strain. Thirdly we show that, for deep sensors, indenter wedge angle may be inferred from an indenter radius estimate.

These theoretical results are tested through experiments with a capacitive silicone rubber tactile sensor. The sensor has a noise level of 0.5% peak strain, linearity of 1% peak strain, and a sensitivity to nearby conductors of 3% peak strain. Identification of the map from surface pressure fourier coefficients to sensor output is accomplished with a residual error of 1.4% peak strain. Nine different indenter radii ranging from a radius of 0.5 mm to 12.7 mm are estimated with a standard deviation of 0.6 mm for 200 N/m loads over 40 degrees of the sensor. Contact location is estimated with an accuracy of 0.19^o

(0.043 mm).

Given the high accuracy of the position estimation, position feedback is integrated into a grasp controller to allow optimal regrasping and manipulation of disks and rectangles. Tactile curvature estimates are displayed to the operator at a 10Hz rate during the manipulation.

Contents

List of Figures

List of Tables

Glossary of symbols

Length in meters, angles in radians, force in Newtons.

$A_k(r)$ See Appendix A.

$B_k(r)$ See Appendix A.

$\mathbf{C}$ A $(2n_c+1)\times(n_c+1)$ real matrix that computes the truncated convolution of the input vector of fourier series with the coefficients for a Hann window.

$\mathbf{D}$ A $n_s \times n_t$ real matrix of percent changes in output whose jth column is the sensor array response to touch j.

$e_{rr}(r,\theta)$ Radial strain at the polar coordinates (r,θ).

$e^s_{rr_k}(r)$ kth fourier sine coefficient of radial strain at r due to p^s_k and q^c_k.

$e^c_{rr_k}(r)$ kth fourier cosine coefficient of radial strain at r due to p^c_k and q^s_k.

$\hat{e}^p_{rr_k}(r)$ kth fourier sine coefficient of radial strain at r due to $p^s_k = 1$. Also the kth fourier cosine coefficient due to $p^c_k = 1$.

$\hat{e}^q_{rr_k}(r)$ kth fourier sine coefficient of radial strain at r due to $q^c_k = 1$. Also the negative of the kth fourier cosine coefficient due to $q^s_k = 1$.

$e_{r\theta}(r,\theta)$ Shear strain at the polar coordinates (r,θ).

$e^s_{r\theta_k}(r)$ kth fourier sine coefficient of shear strain at r due to p^c_k and q^s_k.

$e^c_{r\theta_k}(r)$ kth fourier cosine coefficient of shear strain at r due to p^s_k and q^c_k.

E Young's modulus $= 4.11 \times 10^5$ for Dow Corning HSII silicone rubber.

E A $n_s \times (4n_c + 2)$ real matrix that maps from fourier series pressure coefficients to subsurface strain evaluated at each sensor.

G A $n_s \times n_s$ diagonal matrix of sensor gains used to map from percent change in output to normal strain.

L See Appendix A.

n_c Largest fourier series coefficient represented in the convolved fourier series coefficients. n_c is the frequency in cycles/revolution at which the fourier series is truncated.

n_p Number of points at which to evaluate the surface slope. $n_p >= n_c + 1$.

n_s Number of sensors.

n_t Number of touches during the calibration procedure.

$p(\theta)$ Radial surface traction at θ.

p_k^s kth fourier sine coefficient of the normal surface pressure.

p_k^c kth fourier cosine coefficient of the normal surface pressure.

$\mathbf{p}$ A $2n_c + 1$ vector of fourier series coefficients of the radial surface tractions.

P Total normal load in N/m.

$\mathbf{P}$ A $(2n_c+1)\times n_t$ real matrix of normal pressure coefficients where the jth column is the pressure distribution for touch j.

$q(\theta)$ Tangential surface traction at θ.

q_k^s kth fourier sine coefficient of the tangential surface pressure.

q_k^c kth fourier cosine coefficient of the tangential surface pressure.

$\mathbf{q}$ A $2n_c+1$ vector of fourier series coefficients of the tangential surface tractions.

Q Total tangential load in N/m.

r Radius at which to evaluate stress, strain, or displacement.

r_a Radius of hard core of the sensing finger.

r_b Outer radius of the sensing finger.

r_c Radius of indenting cylinder.

r_s Radius at which sensors are located on the sensing finger.

R Equivalent radius of a cylinder touching a planar sensor to a r_c radius cylinder touching a finger of radius r_b. $\frac{1}{R} = (\frac{1}{r_b} + \frac{1}{r_c})$ or $R = \frac{r_b r_c}{r_b + r_c}$.

$\mathbf{s}$ A n_p dimensional vector of slopes of the deformed material.

$\mathbf{S}$ The generic map from pressure fourier coefficients to indenter shape. It is $\mathbf{S}_p$, $\mathbf{S}_s$, or $\mathbf{S}_a$ depending on whether the indentation is frictionless, sliding, or fully adhesive.

$\mathbf{S}_a$ A $n_p \times (2n_c+1)$ real matrix that maps from fourier series coefficients of the normal pressure distribution to the slope of the deformed material in the contact region under the condition of zero tangential displacement.

$\mathbf{S}_p$ A $n_p \times (2n_c+1)$ real matrix that maps from fourier series coefficients of the normal pressure distribution to the slope of the deformed material in the contact region.

$\mathbf{S}_q$ A $n_p \times (2n_c + 1)$ real matrix that maps from fourier series coefficients of the tangential pressure distribution to the slope of the deformed material in the contact region.

$\mathbf{S}_s$ A $n_p \times (2n_c+1)$ real matrix that maps from fourier series coefficients of the normal pressure distribution to the slope of the deformed material in the contact region assuming that $\mathbf{q} = \mu_f \mathbf{p}$.

$\mathbf{T}$ The generic map from pressure fourier coefficients to tangential displacement in the contact region.

$\mathbf{T}_p$ A $n_p \times (2n_c+1)$ real valued matrix that maps from fourier series coefficients of the normal pressure distribution to the tangential displacement of the material in the contact region.

$\mathbf{T}_q$ — A $n_p \times (2n_c + 1)$ real valued matrix that maps from fourier series coefficients of the tangential pressure distribution to the tangential displacement of the material in the contact region.

$\mathbf{U}$ — The left singular vectors in the singular value decomposition $\mathbf{U}\Sigma\mathbf{V}^T$.

$\mathbf{V}$ — The right singular vectors in the singular value decomposition $\mathbf{U}\Sigma\mathbf{V}^T$.

$u_r(r, \theta)$ — Radial displacement at coordinates (r, θ).

$u^s_{r_k}(r)$ — kth fourier sine coefficient of radial displacement at r due to p^s_k and q^c_k.

$u^c_{r_k}(r)$ — kth fourier cosine coefficient of radial displacement at r due to p^c_k and q^s_k.

$\hat{u}^p_{r_k}(r)$ — kth fourier sine coefficient of radial displacement at r due to $p^s_k = 1$. Also the kth fourier cosine coefficient due to $p^c_k = 1$.

$\hat{u}^q_{r_k}(r)$ — kth fourier sine coefficient of radial displacement at r due to $q^c_k = 1$. Also the negative of the kth fourier cosine coefficient due to $q^s_k = 1$.

$u_\theta(r, \theta)$ — Tangential displacement at coordinates (r, θ).

α — $\alpha = \frac{r_a}{r_b}$,

δ_k — See Appendix A.

ϵ_i — Normal strain at sensor i, $e_{rr}(r_s, \theta_{s_i})$.

$\boldsymbol{\epsilon}$ — A n_s dimensional vector of strains at the sensor locations. $\boldsymbol{\epsilon}$ is also used for the $n_s \times n_t$ matrix of strains where the jth column is the sensor strain due to touch j.

$\hat{\boldsymbol{\epsilon}}$ — A n_s dimensional vector of model strains at the sensor locations. $\hat{\boldsymbol{\epsilon}}$ is also used for the $n_s \times n_t$ matrix of model strains where the jth column is the sensor strain due to touch j.

β — $\beta = \frac{r_a}{r}$.

ϕ — Angle between normal to finger and centerline of the indenter.

$\boldsymbol{\phi}(r, \theta)$ — Airy function in polar coordinates.

γ	$\gamma = \frac{r}{r_b}$.
μ	Shear modulus $= \frac{E}{2(1+\sigma)}$.
μ_f	Coefficient of kinetic friction. $1.0 < \mu_f < 2.0$ for dry Dow Corning HSII silicone rubber.
ω	Spatial frequency for the half plane model in $2\pi\frac{\text{cycles}}{\text{meter}}$
θ	Angle at which to evaluate stress, strain, or displacement.
θ_c	Center of contact.
θ_p	Half angle of indenting wedge. The wedge may also have a radius of curvature of r_c.
θ_w	Including angle of contact region, i.e. width of the contact in radians.
$\boldsymbol{\theta}_s$	n_s dimensional vector of sensor locations.
σ	Poisson's ratio $= 0.5$ for incompressible materials.
$\hat{\sigma}$	$\hat{\sigma} = 3 - 4\sigma$.
Σ	The diagonal matrix of singular values in the singular value decomposition $\mathbf{U}\Sigma\mathbf{V}^T$.
τ_{rr}	Radial stress.
$\tau_{r\theta}$	Shear stress.

Acknowledgements

The research and writing of this dissertation required the assistance and guidance of many people whom I would like to thank. This dissertation follows directly from the work of my advisor, Professor Ron Fearing, who has provided many hours of valuable feedback and discussion. I would like to thank my readers Professor Masayoshi Tomizuka and Professor John Canny for their suggestions on the final draft. I would also like to thank Professor D. Bogy for pointing out the solution to the linear elastic cylindrical plane strain problem, on which this dissertation is based. I also thank Professor R. Ellis for the suggestion to analyze the reliability of shape estimation and investigate indenter classification.

Kiwoon Kim spent many hours in the lab constructing the tactile sensors that were used in the experiments. The design used by Kiwoon was motivated by modifications Mike Lam had made to Fearing's original design. The electronics used were based on a circuit design of Matt Berkemeier. Based on his earlier experience with tactile sensing, Matt helped point the early directions of this study. I also am thankful for Neil Getz's answers to system theory questions, no matter how mundane they were. When it came to using the tactile sensors in grasping, Guenter Wolke, Brian Mirtich, and Carlos Ferrari provided valuable feedback on grasping techniques. I also thank Yoky Matsuoka for carrying out the friction measurement experiments discussed in Chapter 4.

The SIOMS device driver specification detailed in Appendix F was the result of many discussions. I would like to thank Richard Murray for his original NYMPH based controller, Professor John Canny for the inspiration to create a new solution, and Steve Burgett, David Busing, Isabelle Mazon, Max Mendel, Dave Moore, Eric Paulos, and Aaron Wallack for comments on the specification.

I would also like to thank my parents for there support, putting me through college, and providing me with the stamina to see this dissertation to its end. Finally I would like to thank my wife, Valerie, for being patient and supportive throughout this process.

This work was supported in part by California Microelectronics Fellowship, NSF Grants IRI-8810585, IRI-9114446 and IRI-9157051, ONR DURIP Grant N00014-89-J-1463, and NSF-PYI Grant MIP-9057466.

Chapter 1

Introduction

Tactile sensing, like vision, is an integral part of the human sensory system, yet, unlike vision, there does not exist one predominate transducer for the acquisition of tactile sensation. With the advent of the CCD camera, vision researchers can easily acquire and analyze images, whereas tactile researchers are faced with first choosing one of many transducer types for either static or dynamic sensing and then constructing and testing such a sensor without knowing if the particular transducer is the best for providing the information needed for the task. In addition, it is not clear whether a purely passive approach to tactile sensing is adequate, or, rather, if active sensing is required.

The choice of transducer, modality, and passivity will depend largely on the sensing application. One would like to find the combination of these sensing techniques that will provide the most reliable information for a large range of applications. Types of information of interest include: contact location, contact force, local shape, incipient slip, vibration, texture, thermal properties, and wetness.

The interest in tactile sensing in this dissertation follows from its desired target application: robust grasping of objects with unknown shape and frictional properties. In the presence of unknown coefficients of friction and slip, contact location can not be obtained through a manipulator model since estimates of the magnitude of slip will be unreliable. Without *a priori* information about the object shape, constraints on fingertip velocities of a manipulator during rolling motion will be unknown. Finally, to avoid or control slip, the normal and tangential contact forces must be known. Thus, for this task there are three important pieces of information that a tactile sensor should provide: contact location, local shape and force.

This dissertation focuses on static sensing of local contact shape from passive touch through the use of a capacitive array sensor embedded in an elastic medium. Through the use of elastostatic models, we can derive requirements for sensor depth and spacing. Additionally we can analyze the shape sensing problem and determine under what conditions reliable shape sensing is possible. Sensing and grasping experiments with sensors constructed according to the design specification support the modeling approach, but also indicate further challenges.

This chapter motivates the study of tactile sensing with results from neurophysiological research in human tactile sensing. Previous robotic tactile sensing research is summarized and open issues are presented along with the contributions of this dissertation.

1.1 Tactile sensing in humans

The human skin is responsive to both mechanical and thermal stimuli with varying spatial acuity depending on the location on the body. In addition, mechanical stimuli may be perceived through hairy or glabrous (non-hairy) skin. Glabrous skin is found on the palm of the hand and the sole of the foot. On the hand this skin is innervated with a high density of subcutaneous receptors of varying types. Since this dissertation is concerned with tactile sensing as it applies to grasping, we are most interested in the subcutaneous receptors of the glabrous skin of the hand. The capabilities of the human sensing system should help guide our design specifications for a robotic tactile sensing system.

First we review transducers for human touch that convert mechanical stimuli into neuronal pulse trains and then we review recent results on the limits of the human tactile perception processes that convert these pulse strains into a perceived stimulus.

1.1.1 Mechanoreception

At least four different types of mechanical transducers, or mechanoreceptors, can be found in the human skin: the Pacinian corpuscle, the Meissner corpuscle, the Merkel cell-neurite complexes, and the Ruffini endings (Sherrick and Cholewiak, 1986). Each of these responds maximally to different types of mechanical stimuli due to different transduction methods, depth, and location within the skin layers. Table 1.1 summarizes the receptor types with their dimensions, depths and innervation densities. The mechanoreceptors are connected, by one or more nerve fibers, to neuronal pathways that proceed, eventually, to

Mechano-receptor	Dimensions (mm)	Depth (mm)	Innervation density (mm^{-2})	Probable afferent class
Merkel cell-related endings	0.007×0.001	1 - 2	0.70	SAI
Ruffini endings	$(< 0.2) \times (0.5 - 3.0)$	1.5 - 2.5	0.09	SAII
Meissner corpuscles	0.03×0.08	0.5	1.40	FAI
Pacinian corpuscles	1×4	2 - 3	0.21	FAII

Table 1.1: *Mechanoreceptor types and their depths dimensions, and densities in the human fingertip. Dimensions and depths are from (Iggo and Andres, 1982; Sherrick and Cholewiak, 1986). Skin depths are approximate and are based on the location of the receptors relative to the skin layers. Innervation densities in units per square millimeter are from (Johansson and Vallbo, 1979) page 290. The SAII count disregards units clustered around the human finger nail which respond mainly to finger nail motion. The dimensions of the Merkel cell, not endings, are given. The Merkel cell is not the actual transducer but it is instrumental in the transduction process.*

the somatosensory cortex. Patterns of activity on these afferents are closely related to the four types of mechanoreceptors.

Patterns of activity on afferent fibers from the hand may be categorized according to their receptive field size and the temporal frequency of stimuli to which they respond. The four classes are known as SAI, SAII, FAI, and FAII. FAI and FAII are also known as RA and PC respectively. SAI and SAII are slowly adapting, meaning that they respond to a static stimulus. FAI and FAII respond to vibratory stimuli with 50Hz being the best frequency for FAI and 300Hz being the best for FAII. SAI and FAI have small receptive fields meaning that they respond to stimuli over an area of 3 to 4 mm. SAII and FAII have large receptive fields and respond to stimuli over areas of 10 mm or more. Within these receptive fields the SAI, FAI, and FAII afferents respond best to normal pressure while the SAII afferents respond to skin stretch.

It is generally accepted that Pacinian corpuscles feed the FAII afferents and Meissner corpuscles feed FAI afferents. For the SA afferents it is not clear if one or both of the Merkel endings and Ruffini endings feed the SAI and SAII afferents. This is difficult to determine since it requires simultaneous dissection, stimulation and neuronal recording.

The exact stimulus response characteristics for individual mechanoreceptors are also difficult to determine due to the spatial and temporal summation effects. That is, responses from one afferent are due to stimuli distributed both over space and time, thus complicating comparisons with robotic tactile sensors. Additionally, the response pattern typically depends on stimulus frequency, magnitude, and the time history of indentation.

It is, however, possible to model the SAI responses by linear elasticity. In (Phillips and Johnson, 1981) a linear elastic model for a monkey finger was used to predict responses of SAI fibers to indentation of the skin by gratings of various periods. It was found that the SAI responses could be predicted by the maximum compressive strain from a linear elastic plane stress model after sensor depth, gain, and threshold parameters had been fitted. The fitted depth parameters ranged from 0.8 mm to 1.2 mm, and thresholds ranged from 1 % to 10 % compressive strain over 9 different SAI afferents. What would be an interesting extension of this work, and which may be partially explained by the shear strain frequency response in Chapter 2, is the relationship between the large receptive field of the SAII afferents, the depth of the Ruffini endings, and the sensitivity of SAII afferents to skin stretch.

It has been shown that responses of SAI fibers vary noticeably with variation in indenter curvature (LaMotte and Srinivasan, 1987). Unfortunately it is still difficult to say what the neuronal coding is since the response pattern is distributed over many fibers and varies temporally. It does appear that the information is represented in some way on the pathways heading to the somatosensory cortex.

We have seen that the human tactile system utilizes both deep and shallow sensors with innervation density decreasing and receptive field increasing with depth. There are four different types of mechanoreceptor that provide information to the somatosensory cortex. The type of receptor that provides information for fine shape sensing has been a subject of debate. One view is that fine shape information is obtained from RA afferents during active touch while the alternative viewpoint is that the SA afferents, due to their greater density, provide fine shape information during static touch. In a recent review article, Johnson states:

> ... the spatial acuity of the FAI system may be as much as three times poorer than the SAI system, evidence suggesting that form perception is dominated by the SAI system[1].

This conclusion drives us to understand the limits of static shape sensing for similar robotic tactile sensors.

1.1.2 Perception

Based on the mechanoreceptor properties it is instructive to see what the human tactual perception capability is for comparison to the capabilities of robotic sensory systems. Again we limit our interest to passive touch, since that is the focus of this dissertation.

The classic test of static spatial acuity in humans is the *two-point threshold*. In this test, as first described by Weber in 1826, the distance threshold distance between discernible contact points is measured at different locations over the body. For the human fingertip this distance has been measured to be 2.5 mm (Weinstein, 1968). Another test often performed is *localization* in which the threshold at which one can sense that a contact is being made at a new location is measured. For the human fingertip this distance was measured to be 1.5 mm (Weinstein, 1968).

The apparently poor static spatial sensitivity give by these test contrasts with a high degree of precision in static discrimination of contacts of varying area. Vierck used a sequence of ends of cylinders of varying diameter to determine the area discrimination threshold on the human forearm (Vierck Jr and Jones, 1969). Surprisingly even with a two-point threshold of 30 mm on the arm, diameters varying as little as 4 mm could be distinguished. If this 7.5 times improvement in spatial acuity for area over the two-point threshold were to scale linearly with the two-point threshold than one would expect that humans to have the ability to discriminate diameters varying was little as 0.3 mm on the fingertip! Results showing that there is a direct relationship between innervation density and the two point threshold on the hand support this conjecture (Vallbo and Johansson, 1984).

Recently Goodwin conducted experiments to determine the ability of humans to discriminate spheres of varying radii pressed into human fingerpads at various loads after a period of training (Goodwin et al., 1991). It was found that, at a 75 % level, humans

[1] (Johnson and Hsiao, 1992), page 247.

could discriminate between a flat surface and a radius of 204 mm, and between a radius of 6.95 mm and 6.17 mm. Interestingly enough, this corresponds to a change in diameter of contact area of 0.6 mm based on contact area measurements made by Goodwin. In addition, subjects were asked to rate the perceived curvature of indenters on their own scale. For a fixed load subjects ascribed a monotonically increasing perceived curvature with actual curvature, however most subjects perceived an increase in load partly as a decrease in curvature. That is, a larger load made indenters feel flatter. The existence of this effect at a statistically relevant level indicates that more work needs to be done to determine what is actually being sensed: be it curvature, contact area, load, or a combination of these features.

As is pointed out by Goodwin, one conclusion to be made is that active touch is not required for fine perception of shape in humans. For comparison, humans are able to discriminate between a flat surface and a radius of 550 mm when they are allowed to move their fingers over the surface of the object (Gordon and Morison, 1982). Perhaps the most important difference between these two studies is training. For the static perceptual measurements feedback was given, whereas in the haptic case it was not. The reason being that static discrimination of curvature improved with training, but haptic discrimination did not require training. As far as this dissertation is concerned the former study is of more interest since it indicates that humans have the *capability* of discriminating fine surface features with the sensory system described in the previous section.

1.2 Robotic tactile sensing

Robotic tactile sensing is still in its infancy, with implementations of closed loop manipulators using tactile array feedback countable on one hand. We start first by describing different sensing technologies and then review what has been accomplished with them.

1.2.1 Sensing technology

Over the past 20 years many different tactile sensor arrays have been manufactured and reported in the literature. A recent summary can be found in (Nicholls and Lee, 1989; Nicholls, 1992). Of particular interest are implementations suitable for static array sensing, a list of which is given in Table 1.2. It is apparent by comparison of Tables 1.1 and 1.2 that

Transducer	Implementation	Density mm^{-2}	Size
Capacitive	(Boie, 1984)	0.18	8×8
	(Siegel et al., 1987)	0.27	8×8
	(Fearing, 1990)	0.07	8×8
Conductive plastic	(Raibert and Tanner, 1982)	1.00	6×3
Conductive silicone rubber	(Hillis, 1982)	2.56	16×16
	(Reynaerts and Brussel, 1993)	0.69	16×16
	(Shimojo et al., 1991)	1.00	64×64
Conductive rubber strain gauge	(Russell, 1987)	0.007	5×5
Magnetic dipoles in an elastic medium	(Hackwood et al., 1983)	0.25	7×7
Piezoresistive	(Speeter, 1988)	1.58	16×16
Optical waveguide	(Maekawa et al., 1992b)	0.08	10×10
Electrorheological	(Monkman, 1993)	0.25	
Polysilicon Piezoresistors	(Sugiyama et al., 1990)	4.00	32×32
Ultrasonic	(Hutchings et al., 1994)	0.31	16×16

Table 1.2: *Implementations of array sensors and their densities in units per square millimeter.*

sensors have been constructed with resolutions both above and below that of the human SA tactile system.

When choosing a particular technology, one must keep in mind the proposed application. Since we are interested in manipulation, a round finger is desirable. A simple argument is that you can roll round objects on both round and flat sensors, but you can not roll a flat object on a flat sensor. Of the designs listed, those of (Russell, 1987; Fearing, 1990; Maekawa et al., 1992b; Speeter, 1988) have been shown to be well suited for round finger geometries. Of those, (Speeter, 1988) achieved the highest spatial resolution, however the minimum sensed force was 30 grams, compared to 0.5 grams for (Fearing, 1990). As our intended manipulation platform uses grasping forces in the range from 0 to 4 Newtons, the rubber strain gauge transducer is not sensitive enough. The decreased spatial resolution of current cylindrical capacitive sensor designs can be can be compensated for with better sensor spatial response modeling. The optical waveguide design of (Maekawa et al., 1992b) has the same spatial resolution as (Fearing, 1990), however it can not provide pressure distribution information, only contact area and net force.

1.2.2 Analysis techniques

Though many sensors have been constructed, analysis of their performance in most cases has been limited to sensitivity of a single cell and their ability to produce “touch pictures”. Techniques from machine vision such as moment analysis (Montana, 1989; Speeter, 1990; Reynaerts and Brussel, 1993; Hillis, 1982; Maekawa et al., 1992b) and edge detection (Basu et al., 1988; Berger and Khosla, 1991; Sikka and Zhang, 1992) are often applied to these images.

With a high resolution tactile sensor and a Hertz contact model it is possible to estimate the radius of a cylinder using machine vision techniques (Reynaerts and Brussel, 1993). In this case the high spatial resolution is exploited to give an estimate of the area of contact. This information coupled with the total load allows for an estimate of the curvature when the sensor covering is sufficiently soft. If the sensor covering is not soft enough, then the change in contact area will be too small to sense. For the foam rubber used in the experiments loads greater than 20 N were required to estimate the diameter of a 41 mm cylinder with a standard error of 2.2 mm. Unfortunately most of the mean estimates were biased by as much as 3 mm from the actual value. In addition there does not appear to be a systematic trend to the bias.

The main problem with using machine vision techniques is that they are not well adapted to the low resolution typically afforded by tactile sensors. In vision the problem is information reduction, or, finding a feature in a high resolution array of pixels. Finding the area of a patch in an image is already difficult due to the need to choose a threshold value for the edge. Using the same technique for tactile data is much worse due to the reduced spatial resolution of the data. To use machine vision techniques it is first necessary to invert the spatial low pass filter caused by the rubber layer so as to reconstruct a higher resolution surface image (Berger and Khosla, 1991). This technique is ill-conditioned and leads to noise amplification. This technique also assumes the the sampled array is just a low pass filtered version of the normal surface *displacement* when in fact the sampled data is due to both normal and shear tractions on the surface. The influence of surface shear tractions on the subsurface normal displacements can be significant.

Another approach is to develop tactile sensor data analysis techniques based on the mechanics of the rubber material in which they are usually embedded. In (Fearing and Hollerbach, 1985; Fearing, 1990; Fearing and Binford, 1991) it is shown how the linear elastic half space model can be used to determine contact location, curvature, orientation, and force from subsurface normal strain data on a cylindrical finger. One of the key points of this work was that fine position and curvature information could be obtained from a very low sensor densities by using appropriate models. Open questions were the effect of the cylindrical geometry and the reliability of the shape estimates.

Recently these two questions have been addressed using finite element simulations in (Ricker and Ellis, 1992). Their conclusions, that shape interpretation is difficult due to the high degree of similarity of subsurface strain profiles, were confirmed and extended to the case of indenters of multiple classes in (Ellis and Qin, 1994). It was found that contact area could be estimated however indenter classification was not possible. Of particular interest is the conclusion that linear elastic models can not be used to predict subsurface strain profiles when hard inclusions such as copper strips exist in the material. One of the goals of this dissertation is to address the reliability and appropriateness of sensor interpretation algorithms based on linear elasticity.

A linear elastic model was used in (Canepa et al., 1992) to investigate the effectiveness of neural nets to predict contact shape. Singular value decomposition was used to analyze the linear map from indenter shape to subsurface stress for a fixed contact area and show that the rubber layer acts like a low-pass filter. It was shown that a neural net can be trained to identify the contact area and shape from simulated subsurface normal and shear stress in the presence of noise. The reliability of this technique and its ability to discriminate multiple indenter classes was not discussed.

Recently linear elastic models have been used to explain the spatial frequency characteristics of a rubber layer typically used for tactile sensors (Shimojo, 1994). Experimental results showed the dependency of the frequency response on the thickness of the rubber layer. As has been shown in theory, thicker rubber layers were shown to exhibit a lower spatial cut-off frequency.

1.2.3 Sensing and manipulation

It is only recently that tactile array sensors have been integrated into manipulation tasks. Tracking of the center of pressure during manipulation was demonstrated in (Fearing, 1989). Closed loop control of contact location with tactile feedback was demonstrated in (Montana, 1989). In (Speeter, 1990) a UTAH/MIT hand was equipped with tactile sensors, however the extreme complexity of the system coupled with noise, calibration, and a sensor with a high force threshold made feedback control difficult. Edge tracking using tactile feedback was shown in (Berger and Khosla, 1991). In (Sikka and Zhang, 1992) orientation of a line was determined using a binary vision technique but this was not integrated into a control algorithm. Slip was also detected using a static array, however the dynamic tactile sensing approach of (Howe and Cutkosky, 1989; Howe et al., 1990) is perhaps better suited for this task. A nice demonstration of planar manipulation with and without tactile feedback is given in (Maekawa et al., 1992a). Contact location information from the tactile sensor was shown to improve manipulation with rolling contacts.

Integration of tactile array sensors into grasping is still in its infancy, mainly due to the lack of availability of both robust, high performance, low noise tactile arrays and dextrous manipulators on which to use them. Unfortunately, until such devices become easier to obtain, it will not be clear exactly how tactile sensors should be used in manipulation. Using them like a camera to give static touch pictures does not seem practical since cameras are widely available. However they can be used to give valuable local information about the area of surface being touched, as opposed to the global shape of the object being manipulated. It is still an open question as to what local features are most important in grasping.

1.2.4 Sensing global shape

Sensing the global shape of an object from tactile sensor data does not fall into the focus of this dissertation, however it is useful to review techniques that are used to provide applications for our results. There are essentially two approaches to this problem, namely active or passive touch. In the passive touch approach an *a priori* set of contact locations is used and the global shape is determined from data from touches at these points. In (Berkemeier and Fearing, 1993) the axis of a surface of revolution is determined in this manner. In an active touch approach the sensor is scanned over the object using the tactile

1. Should subsurface strain or surface displacement sensors be used?
2. What tactile information is needed for grasping?
3. Can a linear homogeneous isotropic elastic model be used for typically inhomogeneous sensors?

Table 1.3: *Open questions for robotic tactile sensing.*

feedback to guide the search directions. Object classification using this approach has been demonstrated in (Allen, 1992; Stansfield, 1988). It is clear that both of these approaches would be well served with reliable local shape information.

1.3 Open questions

Based on the current state of research we see there being three main questions that need to be answered to determine the best approach to tactile sensing. First, and perhaps the most important, is whether surface displacement or subsurface strain measurements provide the most useful information. Secondly it is not clear what type, or resolution of shape information is needed for grasping. Thirdly, if we are going to use subsurface strain sensors, is a linear elastic model appropriate? The first of these I address now, since it has bearing on the whole direction of thesis.

As is well known, and will be shown in detail in Chapter 2, subsurface strain measurements can provide low-pass filtered estimates of the surface pressure distribution. It has been shown that subsurface strains can been estimated to 0.1 % strain (Fearing, 1990). Surface displacement sensors are capable of resolving 6.8 microns of normal displacement over a range of 800 microns (Hutchings et al., 1994). Figure 1.1 compares the sensed displacement and strain fields and the corresponding sample points for a capacitive strain and an ultrasonic displacement sensor. The strain sensors are located at a depth of 3.3 mm, deeper than the Ruffini endings of the human fingertip.

It is clear that while the displacement sensor is capable of resolving the difference in displacement fields for round and wedge indenters, its sampled signal will be highly aliased, meaning that interpolation is not possible. The subsurface strain field can be interpolated

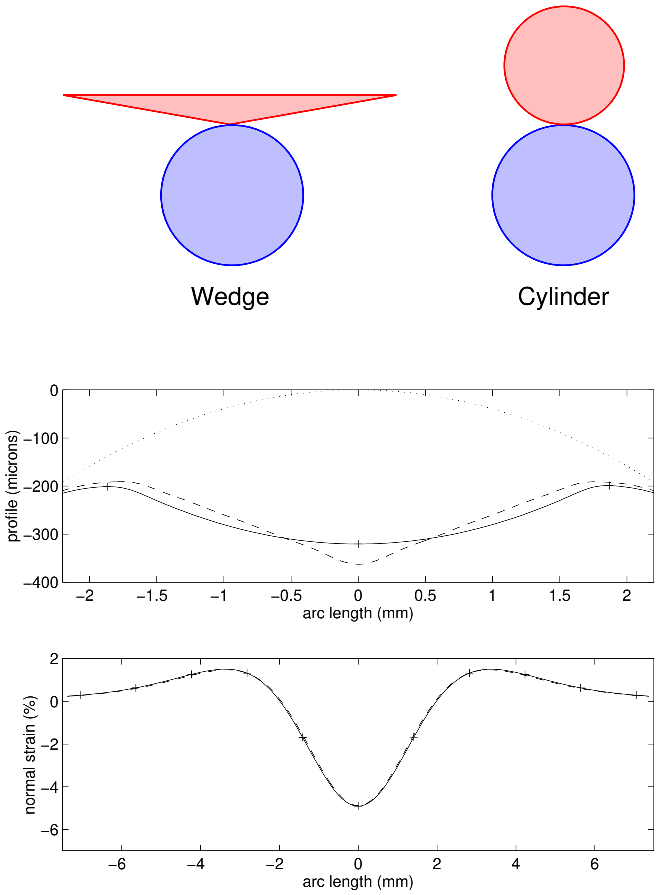


Figure 1.1: *Surface profile and subsurface strains for a rigid 83.3° wedge (dashed) and a rigid 10.6 mm radius cylinder (solid) indented with a load per unit length of 200 N/m on a 12.7 mm silicone rubber cylinder with a Young's modulus of* $E = 4.11 * 10^5 \, \frac{N}{m^2}$. *The upper displacement profile figure also includes the undeformed profile (dotted) for comparison. The crosses in the displacement field correspond to the sample locations for the ultrasonic displacement sensor of (Hutchings et al., 1994). The crosses in the strain field correspond to sample locations for the capacitive sensor of (Siegel et al., 1987).*

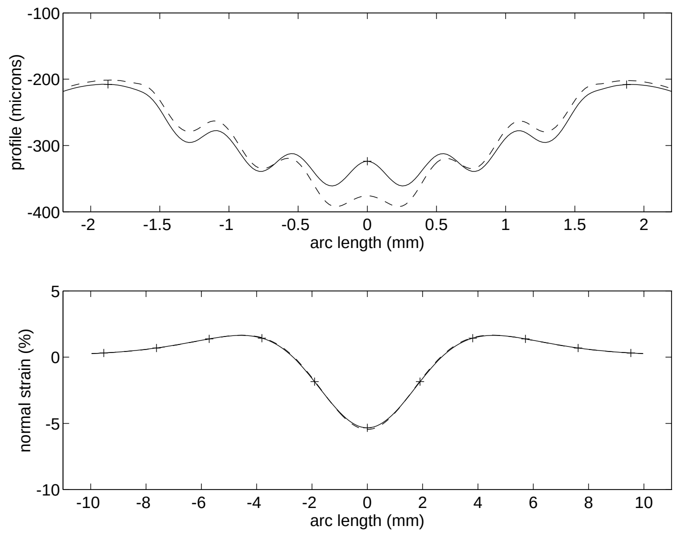


Figure 1.2: *The same indenters as the previous figure except that here 40 micron high bumps with 0.5 mm spacing have been added to the profiles. The subsurface strain can still be interpolated, however the surface displacements can not.*

and peak detected, thus leading to much higher accuracy in determining contact location. That is, the contact location estimate is not limited to the resolution of the sensor spacing. This is made even clearer in Figure 1.2 where bumps of 40 microns have been added to the indenter surfaces. Again the subsurface strain fields are smooth and indistinguishable and may be interpolated to find the center of pressure.

Figure 1.3 shows another problem with displacement sensors is that a sharp edge can be missed entirely, or be misinterpreted as a large radius indenter, when it appears directly between sensors. Subsurface strain sensors will always detect such an indentation. If the contact is moved by half a sensor spacing on the displacement sensor the displacement sensor output will change dramatically. The sensed displacement will appear to change from 20 microns to 200 microns. The sampled subsurface strain can be safely interpolated in both cases to provide an accurate estimate of contact location and load.

Finally, determining an area of contact with a displacement sensor requires picking a threshold that indicates the edge of contact. As with vision, such threshold techniques can be problematic when using a discrete array of sensors, especially at such a low resolution as tactile sensors.

Of course it easy to draw another conclusion from Figures 1.1, 1.2, and 1.3, namely

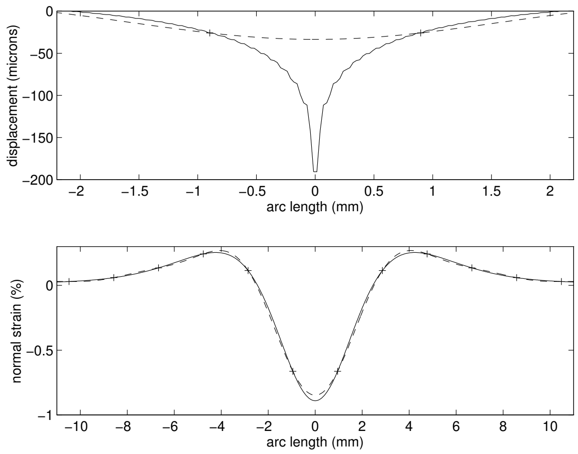


Figure 1.3: *A knife edge indented into an elastic sensor directly between displacement sensors. The crosses indicate sensor locations and the dashed lines are the result of sinc interpolation of the samples. The interpolated displacement profile of a knife edge could easily be mistaken for that of a rounded indenter. In addition the displacement at the displacement sensors is only 2 quantization units whereas the the peak subsurface strain is 10 times the quantization level. Again the rubber is incompressible and* $E = 4.11 * 10^5 \; \frac{N}{m^2}$.

1. For round fingers, what is the effect of cylindrical geometry on the subsurface strain?
2. What is the effect of shear strain and can it be used to enhance indenter shape estimates? How might it be related to the skin stretch sensitivity exhibited by SAII afferents?
3. What is the correct choice of sensor depth given sensor spacing? Can aliasing be tolerated to some degree as it was in (Fearing, 1990)?
4. Can we remove the proximity effect evidenced by other capacitive sensor designs without adversely affecting the sensor impulse response?
5. Can a real-time solution to the shape from strain inverse problem be found in the face of its nonlinear dependency on contact area?

Table 1.4: *Open issues for linear elastic models and capacitive sensors.*

that subsurface strain sensors are bad since they smooth out all the fine surface features. This is perhaps too extreme a conclusion since it neglects the fact that one can control the spatial frequency cut-off with subsurface strain sensors so that the maximal spatial frequency is known. This will be shown in Chapter 2. We will also see that that the transfer function from pressure to displacement is a poor low pass filter hence very high sampling densities are required to avoid aliasing. For subsurface strain sensors one can choose a sensor depth to match a given sensor density and a desired spatial frequency resolution. This allows reliable reconstruction of a bandlimited version of the surface pressure through interpolation. Given subsurface strain sensors the important question is then: what can be interpreted from this bandlimited version of the surface pressure? This question is addressed in detail in Chapter 2.

There are other, somewhat smaller, questions that remain given that we choose to use subsurface strain sensors and a linear elastic model. These are listed in Table 1.4. These questions will be addressed at points throughout this dissertation.

1.4 Approach

This chapter has reviewed human and robotic tactile sensing systems and shown that robotic sensors have been built with similar specifications to that of the human system. We have also seen that humans are capable of determining contact shape from static touch thus it should be possible to do the same with robotic tactile sensors. The previous section discussed open issues in the design of tactile sensors and interpretation methods that might allow robotic sensors to provide information similar to that of the human system. The open issues lead directly to a set of questions that this dissertation addresses.

Answering these questions requires developing a sensor model, analyzing that model to determine theoretical sensing limitations, building a sensor, comparing the response of that sensor to the predicted response, instrumenting multiple sensors in a manipulator, and using them during grasping tasks. With our attention focused on the eventual use of tactile sensors during grasping, we must be concerned with the reliability of methods of determining contact parameters. Unreliable estimates of contact parameters are of no use when closing the control loop. Reliability of contact parameter estimates can be considered in two parts: the first is random error in estimated error and the second is systematic error. While random errors can not be compensated for, systematic errors indicate that with a better model contact parameter estimates can be improved. With this in mind, the applicability of the linear elastic modeling approach must be analyzed. In addition, we must also analyze what contact parameters are the most useful during grasping, and whether that set is the most reliable.

To analyze the reliability of the strain to shape problem we need to understand two maps: first the map from indenter shape to surface pressure, and secondly the map from surface pressure to subsurface strain. Chapter 2 derives a linear elastostatic model for our sensor that allows both of these maps to be characterized. Requirements for sensor depth and spacing can be easily determined directly from this analysis.

The second task is to determine if a sensor can be constructed, instrumented, and calibrated so as to have small enough model error and noise to allow shape sensing. Chapter 3 describes the constructed sensor and its characterization. The results of Chapter 2 are used to implement a shape from strain algorithm and determine the reliability of its estimates.

Finally we should determine if such a sensor can provide extra information in a task as to justify its complexity. Chapter 4 describes a simple two-fingered hand and some manipulation tasks that were carried out with the sensors.

It will be shown that through careful calibration the modeling error can be made sufficiently small that linear elastic models are indeed practical. Also with the correct sensor density a bandlimited version of the surface pressure can be reliably constructed. The shape sensing capabilities of the sensor depend directly on the sensor density and hence sensor depth. It will be shown both theoretically and experimentally that deep sensors can provide contact area information that can be used to estimate indenter radius or wedge angle. Deep sensors can not provide fine shape information, other than a contact area estimate. Theoretically fine shape discrimination can be accomplished with shallower sensors, however a higher sensor density is required. Finally it will be shown how bandlimited shape information can be used to improve closed loop grasping during rolling motion by providing contact location and curvature information.

Chapter 2

Linear elastic tactile sensor model

Under the assumptions of material linearity, isotropy, and homogeneity the problem of determining indenter shape, location, and total load from subsurface strain measurements on a tactile sensor may be analyzed with linear elasticity. This chapter develops a linear elastic model specifically for a cylindrical sensor and analyzes the shape from strain problem using it. In particular we approach the problem through the analysis of two linear maps.

The shape from strain problem can be characterized by two linear maps. The first is the map from surface pressure to subsurface strain. The second is from indenter shape to surface pressure for a given contact area and contact location. Analysis of the first map allows us to formulate sensor spacing requirements and sensing capabilities for a given depth of sensor. In Chapter 3 techniques for identifying this map for a real sensor are discussed. The second map, from indenter shape to surface pressure, provides a convenient way to determine a pressure distribution for an arbitrary indenter shape.

Previous work on modeling cylindrical elastic bodies (Bentall and Johnson, 1967; Nowell and Hills, 1988; Poritsky, 1950) has focused on the surface loading and has not been concerned with the subsurface state of stress and strain. In these cases the elastic half-space model is appropriate. In our case we must know both the state for the material in the contact region at the surface, as well as beneath the surface, where sensors will be located. For subsurface strain modeling, a cylindrical model is more appropriate.

Solving the indenter shape from subsurface strain problem requires the solution of two classic inverse problems in linear elasticity. Typically ill-conditioned inverse problems are solved using a regularization technique (Maniatty, 1993; Constantinesc, 1993). In this chapter singular value decomposition (SVD) is used for both maps. Since the goal is to predict the subsurface strain of an actual sensor, we do not need to produce results to machine precision that other more complex regularization techniques might provide. What is needed is a clear way to interpret the characteristics of these two linear maps, which SVD can easily provide.

We proceed first by solving the mixed-mixed boundary value problem in polar coordinates that is defined by our tactile sensor geometry. This model can be used to find the state of the body given the normal and tangential load in the contact region. Section 2.2 investigates the subsurface strain impulse response that follows directly. In the next section we show how to determine the loading in the contact region given its size and the shape of the indenter in the region by solving the inverse problem from indenter shape to surface pressure. The following section investigates the properties of the solution using singular value decomposition. This leads to a formulation and analysis of the problem of "Band-limited shape interpretation" in Section 2.5. With these results and a noise model for the sensor, theoretical estimates for shape sensing capabilities are derived.

2.1 Stress and displacement fields given surface tractions

2.1.1 Problem statement

We assume that displacements are small and constant in time and that the rubber is homogeneous and isotropic. Under these assumptions we can use a linear elastostatic model. Additionally we make the assumption that the displacement field is independent of the axial direction of the cylinder, that is

$$u_r = u_r(r,\theta) \qquad u_\theta = u_\theta(r,\theta) \qquad u_z = 0 \tag{2.1}$$

where u_r denotes radial displacement, u_θ denotes displacement in the angular direction, and u_z denotes axial displacement. These equations constitute the plane strain assumption. By making the plane-strain assumption we require the indentation to be along the length of a long cylinder.

To solve the elasticity problem we first assume that the surface tractions in the radial direction, $p(\theta)$, and the tangential traction, $q(\theta)$, are known. In Section 2.4 this assumption is removed. We also assume that the rubber layer is rigidly bonded to the solid core in the center of the cylinder. We let r_a be the radius of the core and r_b be the outer

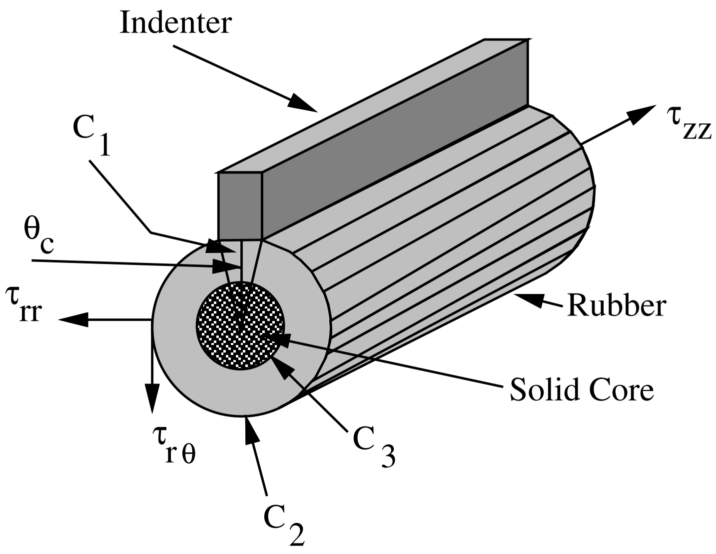


Figure 2.1: *Cylinder in plane strain.*

radius. The contact is centered at θ_c and has a total width, in radians, of θ_w. Referring to Figure 2.1, the boundary conditions may be stated as follows

$$\begin{aligned} C_1: \quad \tau_{rr} &= p(\theta), & \tau_{r\theta} &= q(\theta) \\ C_2: \quad \tau_{rr} &= p(\theta) = 0, & \tau_{r\theta} &= q(\theta) = 0 \\ C_3: \quad u_r &= 0, & u_\theta &= 0 \end{aligned} \tag{2.2}$$

where

$$q(\theta) = \begin{cases} \mu_f p(\theta) & \text{full sliding} \\ < \mu_f p(\theta) & \text{adhesion} \end{cases} \tag{2.3}$$

and μ_f is the coefficient of friction.

We have used C_1 to indicate the contact region. In this region non-zero tractions, both radial and tangential, are exerted by the indenter on the elastic surface. In C_2, the region outside the contact on the surface, there are zero tractions. On C_3 the elastic region is bonded to the rigid core. Note that the functions $p(\theta)$ and $q(\theta)$ are assumed to be zero outside the contact region. With these mixed-mixed boundary conditions the problem is well posed.

2.1.2 Fourier series decomposition

As we are using linear elasticity to solve this problem we may use the principle of superposition to sum independent partial solutions. The sum of these partial solutions gives the desired solution. Since all the partial solutions will be periodic in θ with period 2π, a fourier series decomposition is natural. Using a trigonometric fourier series we can write the surface tractions as follows

$$\begin{aligned}\tau_{rr}(r_b,\theta) = p(\theta) &= p_0 + \sum_{k=1}^{\infty} p_k^c \cos k\theta + \sum_{k=1}^{\infty} p_k^s \sin k\theta \\ \tau_{r\theta}(r_b,\theta) = q(\theta) &= q_0 + \sum_{k=1}^{\infty} q_k^c \cos k\theta + \sum_{k=1}^{\infty} q_k^s \sin k\theta. \end{aligned} \tag{2.4}$$

We desire a solution of the form

$$\begin{aligned}\tau_{rr}(r,\theta) &= \tau_{rr_0}(r) + \sum_{k=1}^{\infty} \tau_{rr_k}^c(r) \cos k\theta + \sum_{k=1}^{\infty} \tau_{rr_k}^s(r) \sin k\theta \\ \tau_{r\theta}(r,\theta) &= \tau_{r\theta_0}(r) + \sum_{k=1}^{\infty} \tau_{r\theta_k}^c(r) \cos k\theta + \sum_{k=1}^{\infty} \tau_{r\theta_k}^s(r) \sin k\theta \\ u_r(r,\theta) &= u_{r_0}(r) + \sum_{k=1}^{\infty} u_{r_k}^c(r) \cos k\theta + \sum_{k=1}^{\infty} u_{r_k}^s(r) \sin k\theta \\ u_\theta(r,\theta) &= u_{\theta_0}(r) + \sum_{k=1}^{\infty} u_{\theta_k}^c(r) \cos k\theta + \sum_{k=1}^{\infty} u_{\theta_k}^s(r) \sin k\theta \\ e_{rr}(r,\theta) &= e_{rr_0}(r) + \sum_{k=1}^{\infty} e_{rr_k}^c(r) \cos k\theta + \sum_{k=1}^{\infty} e_{rr_k}^s(r) \sin k\theta. \end{aligned} \tag{2.5}$$

In polar coordinates the shear strain can be computed directly from the shear stress with $e_{r\theta}(r,\theta) = \frac{1}{2\mu}\tau_{r\theta}(r,\theta)$ where μ is the shear modulus.

Figure 2.2 shows the first 6 components of the fourier series decomposition for radial displacements. This is the basis set for the complete solution. That is, since the solution set is complete, we can use linear combinations of these fourier components to generate the solution to an arbitrary surface pressure distribution up to a maximum frequency.

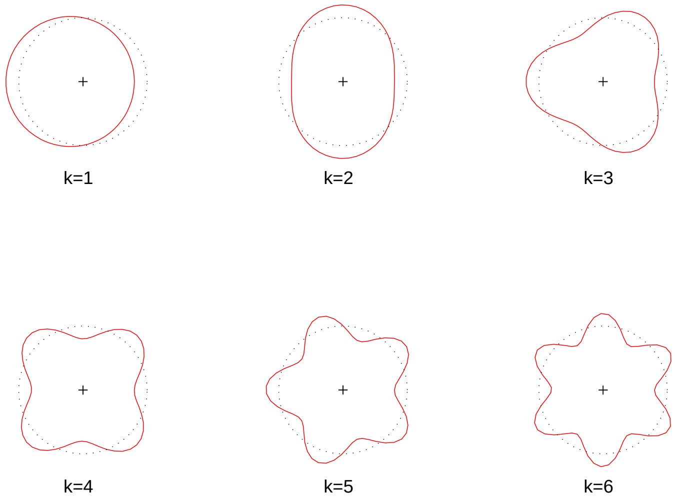


Figure 2.2: *Fourier series basis.*

2.1.3 Plane-strain solutions for cylindrical coordinates using Airy functions

We use an Airy function in polar coordinates to find the solution that satisfies the 3 basic equations of linear elastostatics: strain-displacement,

$$\begin{aligned} e_{rr} &= \frac{\partial u_r}{\partial r} \\ e_{\theta\theta} &= \frac{1}{r}\frac{\partial u_\theta}{\partial \theta} + \frac{u_r}{r} \\ e_{zz} &= \frac{\partial u_z}{\partial z} \\ e_{r\theta} &= \frac{1}{2}\left(\frac{1}{r}\frac{\partial u_r}{\partial \theta} + \frac{\partial u_\theta}{\partial r} - \frac{u_\theta}{r}\right) \end{aligned} \tag{2.6}$$

stress-strain,

$$\begin{aligned} e_{rr} &= \frac{1}{2\mu(1+\sigma)}\left(\tau_{rr} - \sigma\left(\tau_{\theta\theta} + \tau_{zz}\right)\right) \\ e_{\theta\theta} &= \frac{1}{2\mu(1+\sigma)}\left(\tau_{\theta\theta} - \sigma\left(\tau_{rr} + \tau_{zz}\right)\right) \\ e_{zz} &= \frac{1}{2\mu(1+\sigma)}\left(\tau_{zz} - \sigma\left(\tau_{\theta\theta} + \tau_{rr}\right)\right) \\ e_{r\theta} &= \frac{1}{2\mu}\tau_{r\theta} \end{aligned} \tag{2.7}$$

and the stress equations of equilibrium

$$\begin{aligned} \frac{\partial \tau_{rr}}{\partial r} + \frac{1}{r}\frac{\partial \tau_{r\theta}}{\partial \theta} + \frac{\tau_{rr} - \tau_{\theta\theta}}{r} &= 0 \\ \frac{\partial \tau_{r\theta}}{\partial r} + \frac{1}{r}\frac{\partial \tau_{\theta\theta}}{\partial \theta} + \frac{\tau_{r\theta}}{2r} &= 0. \end{aligned} \tag{2.8}$$

The derivation of the following may be found in (Sokolnikoff, 1941). (Timoshenko and Goodier, 1970) provides an overview of two dimensional problems in polar coordinates. Given a function $\phi(r, \theta)$ satisfying

$$\begin{aligned} \nabla^4 \phi &= 0 \\ \nabla^2 &= \frac{\partial^2}{\partial r^2} + \frac{1}{r}\frac{\partial}{\partial r} + \frac{1}{r^2}\frac{\partial^2}{\partial \theta^2} \end{aligned} \tag{2.9}$$

then ϕ generates the plane strain solution

$$\begin{aligned} \tau_{rr} &= \frac{1}{r}\frac{\partial \phi}{\partial r} + \frac{1}{r^2}\frac{\partial^2 \phi}{\partial \theta^2} \\ \tau_{r\theta} &= \frac{1}{r^2}\frac{\partial \phi}{\partial \theta} - \frac{1}{r}\frac{\partial^2 \phi}{\partial r \partial \theta} \\ L &= \nabla^2 \phi, \quad f(z) = L + iM, \quad z = re^{i\theta} \\ g(z) &= l + im = \int f(\eta) d\eta \\ 2\mu u_r &= -\frac{\partial \phi}{\partial r} + (1 - \sigma)(l \cos\theta + m \sin\theta) + \\ & a_1^o \cos\theta + a_2^o \sin\theta \\ 2\mu u_\theta &= -\frac{1}{r}\frac{\partial \phi}{\partial \theta} + (1 - \sigma)(-l \sin\theta + m \cos\theta) - \\ & a_1^o \sin\theta + a_2^o \cos\theta - w^o r. \end{aligned} \tag{2.10}$$

Here μ is the shear modulus and σ is Poisson's ratio. These parameters are related to the more well known Young's modulus, E, by $E = 2\mu(1 + \sigma)$. The function M is the conjugate harmonic to L. The details of the derivation of l, m, and M can be found in a standard text covering complex analysis. The following set of Airy functions, pointed out by (Bogy, 1991), will generate the required complete set of solutions.

$$
\begin{aligned}
\phi^0 &= a_0 \ln r + b_0 r^2 + c_0\theta \\
\phi^1 &= (\frac{a_1}{r} + b_1 r^3 + c_1 r \ln r)\left\{\begin{array}{c}\sin\theta \\ \cos\theta\end{array}\right\} + d_1 r\theta\left\{\begin{array}{c}\cos\theta \\ \sin\theta\end{array}\right\} \\
\phi^k &= (a_k r^k + b_k r^{-k} + c_k r^{2+k} + d_k r^{2-k})\left\{\begin{array}{c}\sin k\theta \\ \cos k\theta\end{array}\right\}
\end{aligned} \tag{2.11}
$$

Given this set we now must solve for the constants, a_k , b_k, c_k, d_k, a_1^o, a_2^o, and w^o given the boundary conditions. We do not give the explicit formulas for these constants, but instead give the equations for the fourier series coefficients which are based on these constants.

2.1.4 Solutions

By substituting in the boundary conditions, the constant coefficients in the previous equations are solved for each k. This gives surface displacement and subsurface strain from surface pressure. It is useful to define the following:

$$
\alpha = \frac{r_a}{r_b} \quad , \quad \beta = \frac{r_a}{r} \quad , \quad \gamma = \frac{r}{r_b} \quad , \quad \hat{\sigma} = 3 - 4\sigma \tag{2.12}
$$

Note that α, β, and γ are all less than 1.

k = 0

For $k = 0$ the following solution is obtained.

$$
\begin{bmatrix} \tau_{rr_0}(r) \\ \tau_{r\theta_0}(r) \\ u_{r_0}(r) \\ u_{\theta_0}(r) \\ e_{rr_0}(r) \end{bmatrix} = \begin{bmatrix} \frac{2+\beta^2(-1+\hat{\sigma})}{2+\alpha^2(-1+\hat{\sigma})} & 0 \\ 0 & \gamma^{-2} \\ \frac{\gamma r_b(-1+\hat{\sigma})(1-\beta^2)}{2\mu(2+\alpha^2(-1+\hat{\sigma}))} & 0 \\ 0 & \frac{r_b(1-\beta^2)}{2\alpha\beta\mu} \\ \frac{(1+\beta^2)(1-\hat{\sigma})}{2\mu(-2+\alpha^2(1-\hat{\sigma}))} & 0 \end{bmatrix} \begin{bmatrix} p_0 \\ q_0 \end{bmatrix} \tag{2.13}
$$

We should point out here that the shear stress and shear strain at the core, r_a , due to a constant shear load at the surface, r_b, increase as r_b^2. This is important to consider when designing rubber coated fingers. Also we note that $u_{\theta_0}(r)$ indicates how much rotation occurs about the axis of the cylinder due to a tangential load. Finally we note that for incompressible materials where $\sigma = \frac{1}{2}$ there is no radial displacement due to a constant load applied normally over the complete surface of the finger.

k = 1

For $k = 1$ the solution is more involved, but simplifies to the following form (A_1 and B_1 are given in Appendix A):

$$\begin{bmatrix} \tau^s_{rr_1}(r) \\ \tau^c_{r\theta_1}(r) \\ u^s_{r_1}(r) \\ u^c_{\theta_1}(r) \\ e^s_{rr_1}(r) \end{bmatrix} = B_1(r)q_1^c + A_1(r)\left(p_1^s + q_1^c\right)$$

$$\begin{bmatrix} \tau^c_{rr_1}(r) \\ -\tau^s_{r\theta_1}(r) \\ u^c_{r_1}(r) \\ -u^s_{\theta_1}(r) \\ e^c_{rr_1}(r) \end{bmatrix} = -B_1(r)q_1^s + A_1(r)\left(p_1^c - q_1^s\right) \tag{2.14}$$

k ≥ 2

For $k \geq 2$ the form of the solution is similar. Again A_k , B_k , δ_k, and L are given in Appendix A).

$$\begin{bmatrix} \tau^s_{rr_k}(r) \\ \tau^c_{r\theta_k}(r) \\ u^s_{r_k}(r) \\ u^c_{\theta_k}(r) \\ e^s_{rr_k}(r) \end{bmatrix} = \frac{1}{\delta_k} A_k(r) B_k L \begin{bmatrix} p_k^s \\ q_k^c \end{bmatrix}$$

$$\begin{bmatrix} \tau^c_{rr_k}(r) \\ -\tau^s_{r\theta_k}(r) \\ u^c_{r_k}(r) \\ -u^s_{\theta_k}(r) \\ e^c_{rr_k}(r) \end{bmatrix} = \frac{1}{\delta_k} A_k(r) B_k L \begin{bmatrix} p_k^c \\ -q_k^s \end{bmatrix} \tag{2.15}$$

A quick glance at the above solutions reveals that the sine terms for e_{rr}, the normal strain, and the cosine terms for $e_{r\theta}$, the shear strain, depend only on the sine terms for the normal surface pressures and the cosine terms for the tangential surface pressures. This is an important property which has implications for indenter shape sensing.

2.2 Impulse and frequency response

If the cylinder is indented with a sharp edge, then the loading will be distributed over a very small area. The fourier series coefficients for such an impulsive load with normal pressure magnitude $P\ \left(\frac{\mathrm{N}}{\mathrm{m}}\right)$ and tangential pressure $Q\ \left(\frac{\mathrm{N}}{\mathrm{m}}\right)$ at contact location θ_c are

$$\begin{array}{lllll} p_0 = \frac{P}{2\pi r_b} & , & p_k^c = \frac{P}{\pi r_b}\cos(k\theta_c) & , & p_k^s = \frac{P}{\pi r_b}\sin(k\theta_c) \\ q_0 = \frac{Q}{2\pi r_b} & , & q_k^c = \frac{Q}{\pi r_b}\cos(k\theta_c) & , & q_k^s = \frac{Q}{\pi r_b}\sin(k\theta_c). \end{array}$$

To allow comparison with the cylindrical model to the half plane used in (Fearing, 1990) we use the same parameters.

$$\begin{array}{rclrcl} \mu & = & 8.3 * 10^4 \frac{\mathrm{N}}{\mathrm{m}^2} & \sigma & = & 0.5 \\ r_a & = & 8.9\mathrm{mm} & r_b & = & 12.7\mathrm{mm} \end{array}$$

Additionally we assume that the sensors are located at radius $r_s = r_a + 0.5\mathrm{mm}$.

Figure 2.3 shows the displacement field due to a knife edge applied at 45 degrees to the surface of the cylinder at $\theta_c = 0$. The subsurface displacement field makes it immediately clear that a large degree of spatial frequency filtering is occurring. To get a better idea of the type of filtering, Figure 2.4 shows the shear and normal strain coefficients, e_{rr_k} and $e_{r\theta_k}$, as they vary with the index k. This plot is in effect the *spatial frequency* response due to the rubber layer. That is, since the frequency spectrum of the loading was flat, Figure 2.4 shows the relative effect of each loading frequency component on the corresponding frequency of the subsurface strain. We note that e_{rr_k} and $e_{r\theta_k}$ drop off to below 10 percent of their maximum value by the 20th coefficient. Due to this high degree of low pass filtering, the subsurface normal strain, as shown in Figure 2.5, does not change perceptibly with contacts that vary in pressure frequencies above the cut-off frequency.

For comparison of the filtering effects of rubber for strain and displacement sensors, Figure 2.6 shows the strain and displacement frequency responses in a conventional Bode plot. The displacement frequency response is the transfer function between surface pressure and surface displacement, much like the strain frequency response is the transfer function between surface pressure and subsurface strain. From conventional Bode plot analysis we know that a one pole filter has a decay of 20 dB/decade, which is very close to the decay of the displacement frequency response. We also know that one pole low pass filters are not very good anti-aliasing filters. On the other hand the subsurface strain responses show a very sharp decay of more than 100 dB/decade indicating that they are very good low

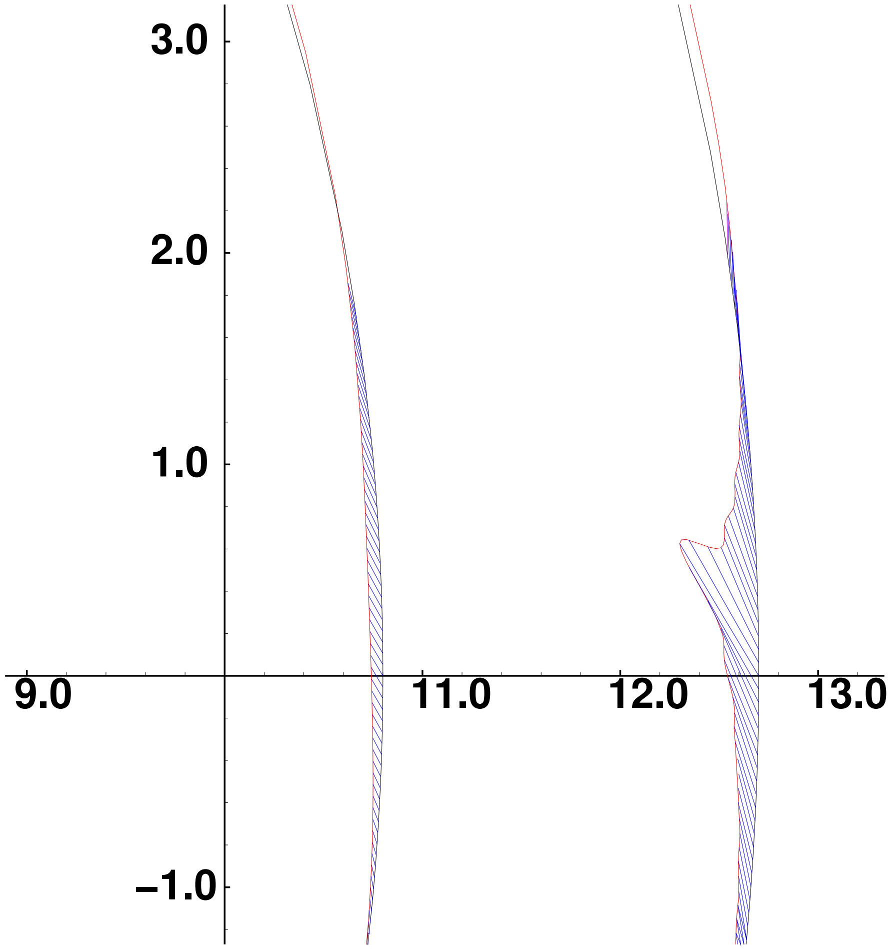


Figure 2.3: *Surface and subsurface displacements due to a concentrated load applied at 45 degrees to the surface. It is assumed that the friction coefficient, μ_f, is greater than 1.0. The dashed lines are drawn between the undeformed and deformed positions. The units are millimeters The center of the cylinder is located far off to the left hand side of the page.*

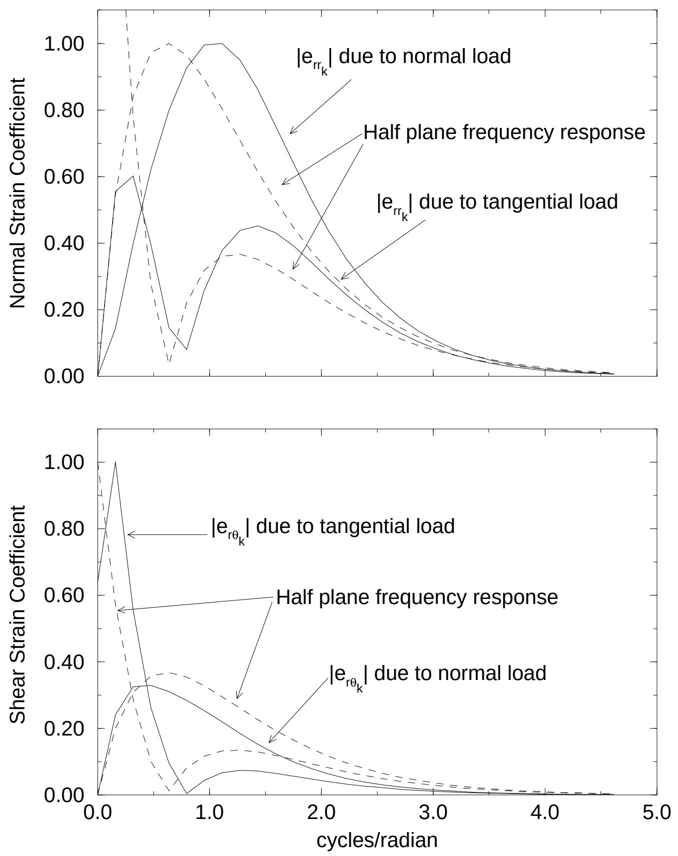


Figure 2.4: *Subsurface strain frequency responses for the cylindrical and half plane models.*

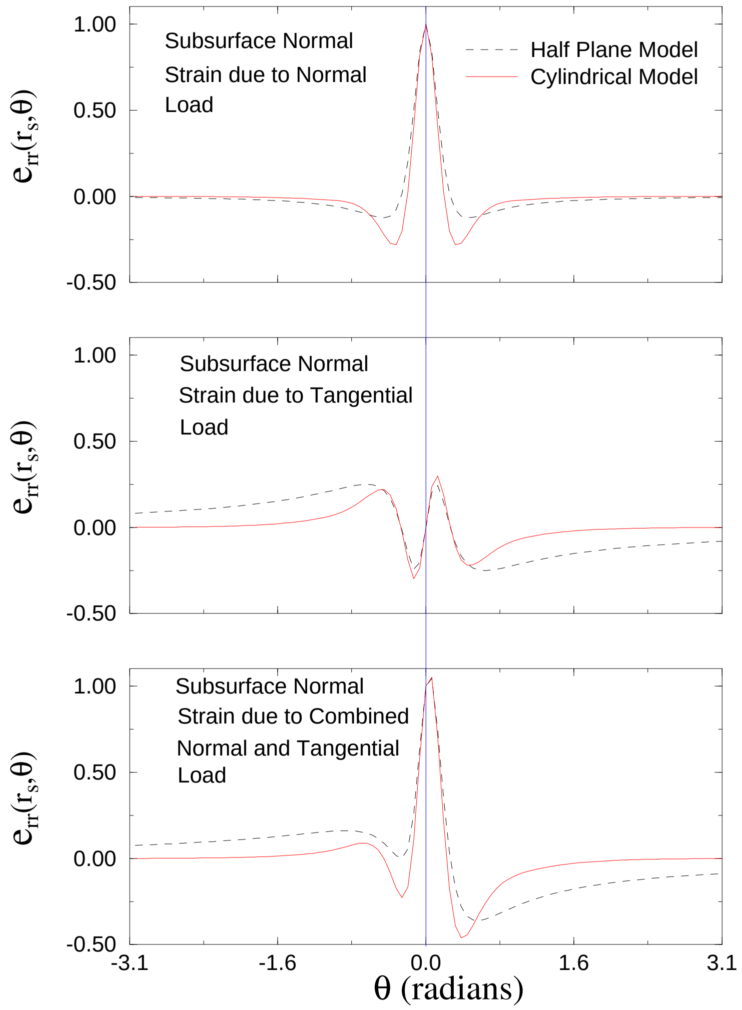


Figure 2.5: *Subsurface normal strain impulse response.*

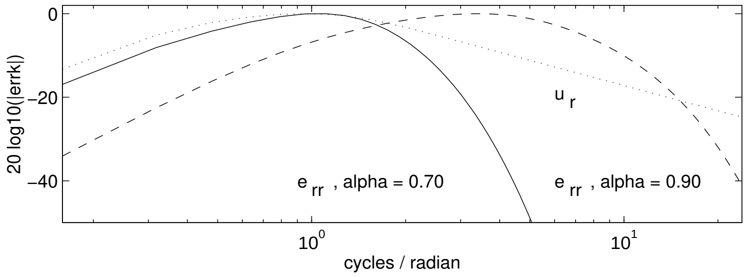


Figure 2.6: *Line load frequency response for normal surface displacement (dotted line) and subsurface strain at two different depths (solid and dashed lines). Notice that the displacement frequency response decays very close to that of a one pole low pass filter at 20 dB/decade while the strain decays much faster at more than 100 dB/decade.*

pass filters. Thus from a filter design point of view, one would much rather use subsurface strain measurements since the strain field has a well defined frequency content whereas the frequency content of the displacement field is not clearly bandlimited.

Figures 2.4 and 2.5 also show the comparable frequency and impulse response for the plane strain half plane model (Johnson, 1985; Fearing and Hollerbach, 1985) which is derived in Appendix B. By unwrapping the cylinder and treating arc length as distance we can see the effect of assuming the cylinder to be a half plane. When comparing the two models the spatial frequencies must be converted to the correct units since the cylindrical model expresses distance in radians around the circumference and the half plane model expresses distance in meters. Thus the fourier index k in the cylindrical model corresponds to the spatial frequency, $\omega = 2\pi\frac{\text{cycles}}{\text{meter}}$, of $\omega = 2\pi\frac{k}{2\pi r_b}$. At these frequencies the subsurface normal and shear strain frequency responses predicted by the half plane model for purely normal loads are given by

$$\begin{aligned}\hat{e}_{22_k} &= \frac{1}{2\mu}Pe^{-k(1-\gamma)}\left(-1+2\sigma-k(1-\gamma)\right)\\ \hat{e}_{12_k} &= j\frac{1}{2\mu}Pe^{-k(1-\gamma)}k(1-\gamma)\end{aligned}\tag{2.16}$$

and for a tangential load of Q N/m they are

$$\begin{aligned}\hat{e}_{22_k} &= j\frac{1}{2\mu}Qe^{-k(1-\gamma)}\left(-2\sigma+k(1-\gamma)\right)\\ \hat{e}_{12_k} &= -\frac{1}{2\mu}Qe^{-k(1-\gamma)}\left(1-k(1-\gamma)\right)\end{aligned}\tag{2.17}$$

where $j = \sqrt{-1}$ and $\gamma = \frac{r}{r_b}$ as before. This formulation looks different from that of (Fearing and Hollerbach, 1985) due to the substitution of the cylindrical geometry parameters into the half plane model. It is assumed that the depth, given by $r_b - r$ is less than r_b. That is, $r > 0$, which will always be the case when comparing the half-plane to the cylindrical model.

The half plane and cylindrical model predict different amplitudes for subsurface strain. The peak strain for an impulse at the same depth for the half plane model is 1.25 times that for the cylindrical model for our sensor parameters. This is due mainly to the rigid backing. Generally speaking, decreasing the core radius, r_a, will increase the strain predicted by the cylindrical model with a constant sensor radius. For this reason, the plots have been normalized to the peak strains.

In both figures the response due to normal and tangential loading has been separated to show the relative effect of normal and tangential loads on the normal and shear strain. The half space model shows a larger degree of filtering from normal loads to the normal strain given the same rubber thickness as is apparent from the smaller side lobes in the impulse response and the smaller cutoff frequency in the frequency response. In Figure 2.5 we see that the effect of tangential loading on normal strain is similar until the effect of the cylinder's curvature becomes apparent at $\theta = .4$ radians.

In Figure 2.4 we see the dominance at low frequencies of the tangential loading on the shear strain. Whereas the normal strain due to the normal and tangential loading are just a factor of 2 in comparative magnitude and have a similarly shaped frequency response, the shear strain is affected by a factor of 3 more by the tangential load than the normal load and the frequency response drops off much more sharply.

So far we have only discussed the filtering effects for one set of radii parameters. Figure 2.7 shows how the -20 dB cut-off frequencies and tangential load influence vary as $\alpha = \frac{r_a}{r_b}$ approaches 1.0. Fitting the cut-off frequency for normal strain due to normal load to the function $\frac{a}{(1-\alpha)+b}$ where a and b are fitted, we have

$$f_c(\alpha) = \frac{0.90}{(1-\alpha) - 0.0274} \tag{2.18}$$

where f_c is the cut-off frequency in cycles per radian. This function fits well for $\alpha < 0.9$. Using the Nyquist criteria, the required sampling density is $\frac{1}{2f_c(\alpha)}$ radians. If we multiply this quantity by r_b, then the spacing requirement, in meters, is

$$0.56(r_b - r_a) - 0.015r_b. \tag{2.19}$$

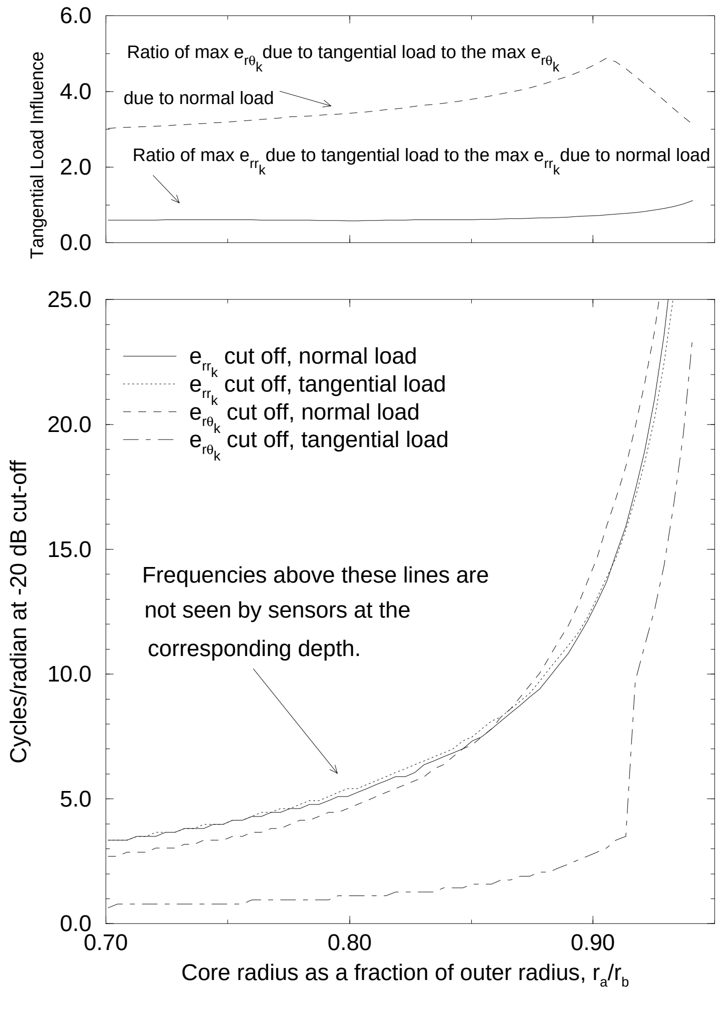


Figure 2.7: *Variation in -20dB filter cut-off with* $\alpha = \frac{r_a}{r_b}$. *Sensors are located at* $r_s = r_a + 0.5mm$.

Recognizing $(r_b - r_a)$ as the sensor depth, we see that a sensor spacing approximately 1/2 of the depth is required.

The shear strain cut-off due to tangential loading has a few interesting properties. First, as can be seen in Figure 2.4, the shear stress to shear strain frequency response is the combination of a low pass filter and a notch filter. This makes the definition of a cut-off frequency problematic since the frequency response may cross a given cut off threshold multiple times. For the results shown in Figure 2.7 the cut off frequency was defined to be the highest frequency at which the frequency response crossed the -20 dB line. This definition of the cut off frequency led to the discontinuity in the shear strain cut off frequency as α was increased.

Another interesting property of the shear strain cut off due to tangential loading is that the shear strain is filtered to only 1.6 cycles per radian when α is as large as 0.85. This indicates that shear strain sensing may be used as a good indicator of tangential load, but not of contact shape in the presence of tangential loading since shear strain measurements will be dominated by a response to the low frequencies of the shear loading. In reference to previous work, this indicates that the use of shear stress to predict contact area as used in (Canepa et al., 1992) would be difficult in the presence of tangential loading.

We may also use the shear strain cut off to explain the response of the SAII afferents of the human tactile sensory system. Recall that the SAII afferents responded best to static skin stretch. Skin stretch may also be thought of as shear loading. Additionally SAII afferents exhibited a large receptive field. A large receptive field may be interpreted as a spatial low pass filter. From the results of this section it makes sense then that a receptor that is sensitive to shear loading would exhibit a large receptive field, or , rather, would be sensitive to the low spatial frequencies of the shear loading.

2.3 The E map

Since the main interest in this model is its use in predicting the subsurface strain at a discrete number of points it is useful to define the matrix $\mathbf{E} \in \Re^{n_s \times (4n_c+2)}$ which maps from fourier series coefficients up to frequency n_c to n_s subsurface normal strains. $\mathbf{E}$ is shown pictorially in Figure 2.8. If we let

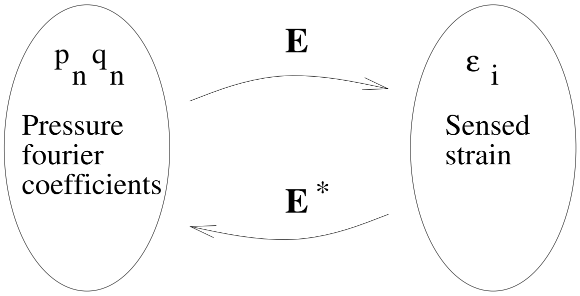


Figure 2.8: *The map* $\mathbf{E}$ *from surface pressure to subsurface strain.*

$$\mathbf{p} = \begin{bmatrix} p_0 \\ p_1^s \\ p_1^c \\ \vdots \\ p_{n_c}^s \\ p_{n_c}^c \end{bmatrix} , \quad \mathbf{q} = \begin{bmatrix} q_0 \\ q_1^s \\ q_1^c \\ \vdots \\ q_{n_c}^s \\ q_{n_c}^c \end{bmatrix} , \quad \boldsymbol{\epsilon} = \begin{bmatrix} e_{rr}(r_s, \theta_{s_0}) \\ \vdots \\ e_{rr}(r_s, \theta_{s_{(n_s-1)}}) \end{bmatrix} \tag{2.20}$$

where $\boldsymbol{\theta}_s$ is the vector of sensor locations, then the relationship between $\mathbf{p}, \mathbf{q}$, and $\boldsymbol{\epsilon}$ is

$$\boldsymbol{\epsilon} = \mathbf{E} \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \end{bmatrix} . \tag{2.21}$$

To construct $\mathbf{E}$ we first compute the fourier series coefficients for normal strain due to unit normal and tangential loads at each frequency as

$$\begin{aligned} \hat{e}^p_{rr_k} &= \frac{1}{\delta_k} A_{k_5}(r_b) B_k L \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ \hat{e}^q_{rr_k} &= \frac{1}{\delta_k} A_{k_5}(r_b) B_k L \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{aligned} \tag{2.22}$$

where A_{k_5} indicates the fifth row of A_k. Coefficients for $k = 0$ and $k = 1$ can be computed similarly. Now, for $j \neq 0$ and $j \neq 2n_c + 1$, $\mathbf{E}$ is given by

$$E_{ij} = \begin{cases} \hat{e}^p_{rr_k} \sin(k\theta_{s_i}) & j \text{ odd} \quad j < 2n_c + 1 \\ \hat{e}^p_{rr_k} \cos(k\theta_{s_i}) & j \text{ even} \quad j < 2n_c + 1 \\ -\hat{e}^q_{rr_k} \cos(k\theta_{s_i}) & j \text{ even} \quad j > 2n_c + 1 \\ \hat{e}^q_{rr_k} \sin(k\theta_{s_i}) & j \text{ odd} \quad j > 2n_c + 1 \end{cases} \tag{2.23}$$

where $k = \mathrm{int}((j+1)/2)$. The first and $2n_c + 1$ column of $\mathbf{E}$ are given by

$$E_{i0} = \hat{e}^p_{rr_0} \tag{2.24}$$

$$E_{i(2n_c+1)} = \hat{e}^q_{rr_0}. \tag{2.25}$$

We already have seen in Section 2.2 that this map may be characterized as a low-pass filter, or more precisely, a band-pass filter with a low center frequency. This section has shown how this map can be determined theoretically. Later, in Chapter 3 methods for identifying this map for an actual sensor will be explored. Discussion of the singular value decomposition and the conditioning of $\mathbf{E}$ will be reserved until that point.

2.4 Surface pressure from indenter shape, the S map

The model developed in the Section 2.1 assumed that the normal and tangential surface stresses were known. *Contact models* are used to determine the surface pressure from indenter parameters such shape and total load. A commonly used contact model is the *Hertz* contact. Hertz made the observation that when a rigid cylinder comes into frictionless contact with an elastic half-space the pressure distribution is parabolic. By assuming that the contact area is small the Hertz model can be used for the case of the contact of an elastic and a rigid cylinder. The advantage of using the Hertz model is that it has a closed form, thus it it easy to compute the fourier series coefficients of the pressure distribution directly from the indenter parameters. The disadvantages are that it does not take into account the rigid core of our sensor nor does it account for shear tractions in sliding and adhesive contacts. Appendix C gives the equations for the pressure distribution and contact area predicted by the Hertz model for the contact of a rigid cylinder of radius r_c and an elastic cylinder of radius r_b.

If we wish to analyze a contact with an arbitrary indenter shape with or without friction we must determine the combination of normal and tangential tractions that result in a deformed profile matching that of the indenter in the contact region. Pressure distribution for contact cylinders in which one cylinder has a rigid core have been addressed before (Nowell and Hills, 1988). In that case the primary interest was accurate reconstruction of the surface pressure. A fourier series solution like the one derived in Section 2.1 is not well suited for this problem since the fourier series must be truncated when solving the problem on a computer, thus causing the high frequencies to be lost. The fourier series is well suited for analyzing the subsurface strains since, as we have seen in Section 2.2, only the low frequencies are sensed. What we are interested in is how changes in the high frequencies for different contact shapes are seen in the low frequencies sensed by the strain sensors. Small errors in the surface pressure distribution can be tolerated when analyzing this problem, so the fourier series decomposition will continue to be used.

Since we want to determine $p(\theta)$ and $q(\theta)$ given an indenter shape, load, and location we need to invert the model derived in Section 2.1. Unfortunately these equations do not adapt easily to the shape and load formulation, instead we will state the equivalent constraints: indenter shape, location (θ_c), and the size of the area of contact (θ_w) . The slope of the deformed surface is given by

$$s(\theta) = \left.\frac{\partial u_r}{\partial \theta}\right|_{(r_b,\theta)} \tag{2.26}$$

With the correct loading, $s(\theta)$ will be the same as the slope of the indenter in the contact region.

To simplify discussions of both wedge and cylindrical shapes the following shape function, shown in Figure 2.9, will be used

$$\begin{aligned} s(\theta, r_c, \theta_p) &= \left.\frac{\partial u_r}{\partial \theta}\right|_{(r_b,\theta)} \\ &= \begin{cases} \frac{r_b(r_c+r_b)}{r_c}\theta & |\theta| < \frac{r_c}{r_b \tan\theta_p} \\ \frac{r_b}{\tan\theta_p}\mathrm{sgn}(\theta) + r_b\theta & |\theta| > \frac{r_c}{r_b \tan\theta_p} \end{cases} . \end{aligned} \tag{2.27}$$

This function gives the rate of change with θ of the radial displacement of the sensor such that the deformed finger profile matches that of the indenter in the contact region. This allows specification of wedge indenters with a half angle of θ_p and nonzero radius of curvature of r_c. By picking θ_p and r_c appropriately a continuum from wedge indenters to rounded indenters can be constructed. For round indenters, where θ_p is small, the composite radius, R,

$$\frac{1}{R} = \frac{1}{r_b} + \frac{1}{r_c} \tag{2.28}$$

will also be discussed in this dissertation. R can be interpreted as the equivalent radius of the indenter if the sensor were flat. It is necessary to discuss R and not r_c when the reliability of the indenter parameter estimates is addressed in Chapter 3. This is due to the fact that for flat indenters r_c is infinite hence the standard error of its estimate should also be infinite. The standard error of R can be discussed since it is equal to r_b for flat indenters.

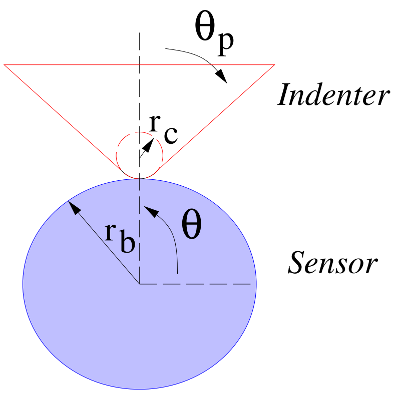


Figure 2.9: *Indenter shape as a function of θ_p and r_c. The indenter is shown applied to a cylindrical sensor of radius r_b. In the contact region the deformed profile of the sensor must match that of the rigid indenter.*

We wish to find the distribution of pressure in the contact region that results in a slope field given by the indenter parameters, θ_p and r_c. At this point we must resort to numerical techniques and truncate the fourier series. If we wish $n_{\max}$ periods of the highest frequency in the series in the contact region then we must compute the fourier series up to

$$n > \frac{2\pi n_{\max}}{\theta_w} \tag{2.29}$$

Since our main concern is with sensor data inversion, we can be satisfied with picking our $n = n_c = 300$ so that for a 2 mm contact we will have 7.6 periods of the highest frequency component in the contact region. For smaller contact areas $n_c = 600$ is used. This will provide an adequate approximation of the contact shape while easily including all the frequencies that will be sensed below the surface.

We analyze the maps from indenter shape to surface pressure for three different cases: frictionless indentation, indentation with sliding, and indentation with adhesion. In each case a slightly different version of the map $\mathbf{S}$ will be created and denoted $\mathbf{S}_p$, $\mathbf{S}_s$, and $\mathbf{S}_a$ respectively. They all map from surface pressure fourier coefficients to slope in the contact region. $\mathbf{S}$ is used when referring to the map in general and $\mathbf{S}_p$, $\mathbf{S}_s$, and $\mathbf{S}_a$ are used when discussing a particular case of tangential loading.

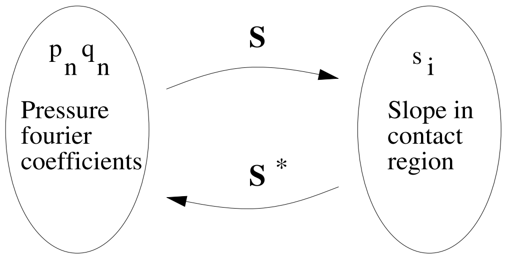


Figure 2.10: *The* **S** *map.*

2.4.1 Frictionless indentation

We start by computing the slope of the indenter at $n_p > (n_c + 1)$ points in the contact region as

$$\theta_i = \theta_c - \frac{2(i - \frac{n_p}{2})\theta_w}{n_p} \tag{2.30}$$

$$s_i = s(\theta_i) \tag{2.31}$$

$$0 \le i < n_p$$

and let $\mathbf{s}$ be the vector corresponding to s_i. We next compute a $n_p \times (2n_c + 1)$ real valued matrix, $\mathbf{S}_p$, which maps from the fourier series coefficients of the normal tractions (truncated to the frequency $n_c/(2\pi)$) to the slope, $\frac{\partial u_r}{\partial \theta}$, at each point in the contact region. Figure 2.10 illustrates the $\mathbf{S}$ map. To do this we first compute the fourier series coefficients for surface normal displacements due to unit normal loads at each frequency as

$$\hat{u}^p_{r_k} = \frac{1}{\delta_k} A_{k_3}(r_b) B_k L \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{2.32}$$

where A_{k_3} indicates the third row of A_k. Coefficients for $k = 0$ and $k = 1$ can be computed similarly. Now, for $j \neq 0$, $\mathbf{S}_p$ is given by

$$S_{p_{ij}} = \begin{cases} \hat{u}^p_{r_k} k \cos(k\theta_i) & j \text{ odd} \\ -\hat{u}^p_{r_k} k \sin(k\theta_i) & j \text{ even} \end{cases} \tag{2.33}$$

where $k = \text{int}((j+1)/2)$. The first column of $\mathbf{S}_p$ is given by

$$S_{p_{i0}} = \hat{u}^p_{r_0} \tag{2.34}$$

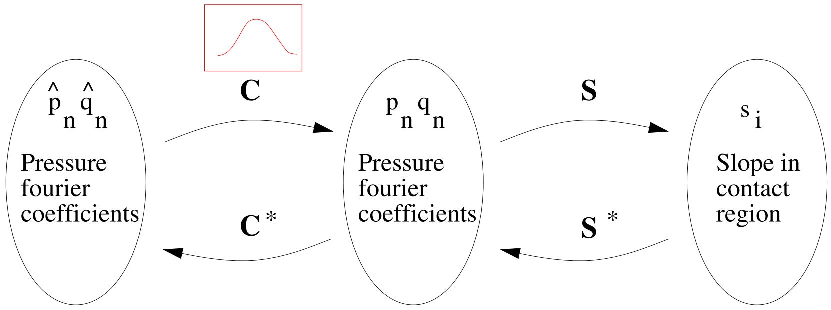


Figure 2.11: *The* $\mathbf{S}$ *and* $\mathbf{C}$ *maps.*

With $\mathbf{S}_p$ constructed as above, it may be applied to a vector of fourier series coefficients for normal tractions to give the slope in the contact region. With $\mathbf{p}$ as defined in Section 2.3

$$\mathbf{s} = \mathbf{S}_p\mathbf{p}. \tag{2.35}$$

If we were simply to invert $\mathbf{S}_p$ to determine the fourier coefficients given the slope in the contact region then we would have no guarantee that the pressures would be zero outside the contact region. To solve this problem we convolve the p_k with the coefficients for a Hanning window truncated to $n_c/2$ *before* multiplying them by $\mathbf{S}_p$. Call this convolution matrix $\mathbf{C}$. $\mathbf{C}$ has dimensions $(2n_c + 1) \times (n_c + 1)$. The maps $\mathbf{S}$ and $\mathbf{C}$ are illustrated in Figure 2.11. We have

$$\mathbf{p} = \mathbf{C}\tilde{\mathbf{p}} \tag{2.36}$$

$$\mathbf{s} = \mathbf{S}_p\mathbf{C}\tilde{\mathbf{p}} \tag{2.37}$$

where the ˜ symbol is used to note the fact that the coefficients $\tilde{\mathbf{p}}$ will have non-zero pressure outside the contact region, but $\mathbf{p}$ will have minimal energy outside the contact region due to the windowing.

A direct inversion of $\mathbf{S}_p\mathbf{C}$ would be numerically unstable, thus we have used singular value decomposition to compute its pseudo-inverse as:

$$\mathbf{S}_p\mathbf{C} = \mathbf{U}\Sigma\mathbf{V}^T \tag{2.38}$$

$$(\mathbf{S}_p\mathbf{C})^+ = \mathbf{V}\Sigma^{-1}\mathbf{U}^T \tag{2.39}$$

where Σ is diagonal and $\mathbf{U}$ and $\mathbf{V}$ correspond to pure rotations. This decomposition provides valuable insight into the shape-to-pressure map that will be discussed in Section 2.5.

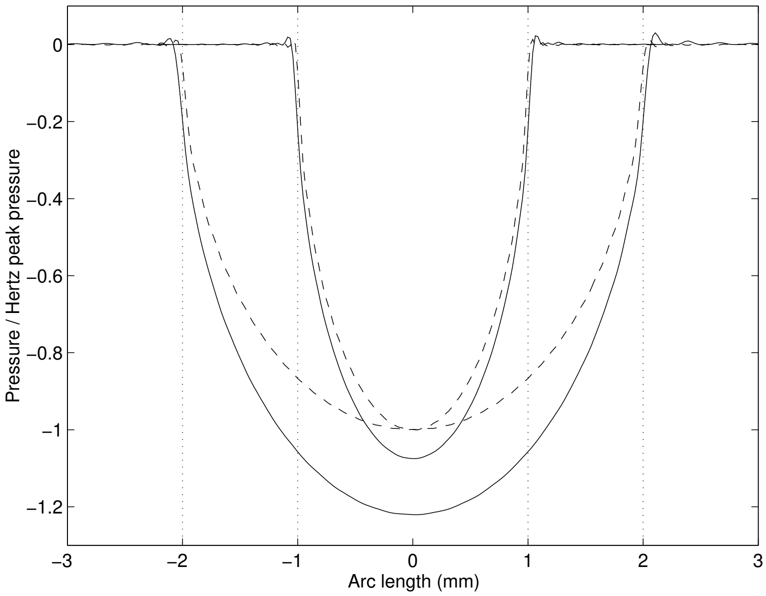


Figure 2.12: *Surface stress for flat frictionless indenters with 2 mm and 4 mm contact widths. The dotted vertical lines indicate the bounds of the contact region. Both pressure distributions are divided by the peak pressure predicted by a Hertz model. The Hertz pressure is plotted as a dashed line.*

We illustrate the use of the inverse with the following example. If we wish to model the frictionless contact of a flat surface with the cylinder with a contact length of 2 mm then we choose

$$\begin{aligned} \theta_w &= \frac{0.002}{r_b} \\ s_i &= r_b \theta_i \end{aligned}$$

and compute the surface tractions as

$$\mathbf{p} = \mathbf{C}\mathbf{V}\Sigma^{-1}\mathbf{U}^T\mathbf{s}. \tag{2.40}$$

The resulting surface stress is shown in Figure 2.12 with the normal pressure distribution that would be predicted by a Hertz contact. As was shown by Nowell and Hillis (Nowell and Hills, 1988), the rigid core of the cylindrical finger results in a larger load and peak pressure for a given radius indenter, r_c, and contact width, θ_w. Another way of interpreting this is that for the same total load, P, and r_c, the contact area for the finger with the rigid core is smaller than that for the Hertz model. Or, equivalently, for a contact on the finger with a rigid core to have the same P and θ_w as that of a Hertz contact, its r_c would have to be smaller.

This section has discussed how to derive $\mathbf{S}_p$ and have shown how it can be used to determine the contact pressure distribution. In Section 2.5 characteristics of the SVD of the map will be discussed.

2.4.2 Indentation with full sliding

The sliding contact case is not much more difficult to analyze than the frictionless case since the tangential pressure is just μ_f times the normal pressure

$$\mathbf{q} = \mu_f \mathbf{p}. \tag{2.41}$$

In this case the dimension of the $\mathbf{S}$ map remains the same, $n_p \times (2n_c + 1)$. If we define the fourier coefficients of normal displacement due to unit tangential load as

$$\hat{u}^q_{r_k} = \frac{1}{\delta_k} A_{k_3}(r_b) B_k L \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{2.42}$$

and the map $\mathbf{S}_q$ as we did $\mathbf{S}_p$

$$S_{q_{ij}} = \begin{cases} \hat{u}^q_{r_k} k \sin(k\theta_i) & j \text{ odd} \\ \hat{u}^q_{r_k} k \cos(k\theta_i) & j \text{ even} \end{cases} \tag{2.43}$$

then the slope in the contact region due to the normal and tangential tractions is

$$\mathbf{s} = \mathbf{S}_p \mathbf{p} + \mathbf{S}_q \mathbf{q} \tag{2.44}$$

$$= (\mathbf{S}_p + \mu_f \mathbf{S}_q)\mathbf{p}. \tag{2.45}$$

The slope map for sliding contacts, $\mathbf{S}_s$, may be defined as

$$\mathbf{S}_s = \mathbf{S}_p + \mu_f \mathbf{S}_q \tag{2.46}$$

so that $\mathbf{s} = \mathbf{S}_s \mathbf{p}$ and $\mathbf{p} = \mathbf{C}(\mathbf{S}_s \mathbf{C})^+ \mathbf{s}$.

2.4.3 Indentation with full adhesion

In the case of indentation with full adhesion we must determine the normal and tangential tractions that result in the surface slope matching that of the indenter and zero tangential displacements in the contact region. Using these boundary conditions we expect the somewhat unrealistic result that the ratio of tangential pressure to normal pressure at the edges of the contact will be infinite (Bentall and Johnson, 1967). To allow a more realistic solution would require the introduction of partial slip, or micro-slip, regions in the contact area. Recall, however, that we are mainly interested in the low frequency components of the pressure distribution. Small regions of micro-slip will be at a high spatial frequency so they should have a small effect on the lower frequencies of the pressure distribution.

To enforce the constraint of zero tangential displacements we must first construct the map from surface pressure to tangential displacement. Following the same procedure as in the previous sections, let the tangential displacements due to unit normal and tangential loads be given by

$$\begin{aligned} \hat{u}^p_{\theta_k} &= \frac{1}{\delta_k} A_{k_4}(r_b) B_k L \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ \hat{u}^q_{\theta_k} &= \frac{1}{\delta_k} A_{k_4}(r_b) B_k L \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \end{aligned} \tag{2.47}$$

Define $\mathbf{T}_p$ and $\mathbf{T}_q$ analogous to $\mathbf{S}_p$ and $\mathbf{S}_q$ for $j \neq 0$

$$\begin{aligned} T_{p_{ij}} &= \begin{cases} \hat{u}^p_{\theta_k} \cos(k\theta_i) & j \text{ odd} \\ -\hat{u}^p_{\theta_k} \sin(k\theta_i) & j \text{ even} \end{cases} \\ T_{q_{ij}} &= \begin{cases} \hat{u}^q_{\theta_k} \sin(k\theta_i) & j \text{ odd} \\ \hat{u}^q_{\theta_k} \cos(k\theta_i) & j \text{ even} \end{cases}. \end{aligned} \tag{2.48}$$

$\mathbf{T}_p$ and $\mathbf{T}_q$ then map from fourier series coefficients of the normal and tangential surface stress to tangential displacements in the contact region. The tangential displacement constraint may now be expressed as

$$\mathbf{0} = \mathbf{T}_p \mathbf{p} + \mathbf{T}_q \mathbf{q} \tag{2.49}$$

$$= \mathbf{T}_p \mathbf{C} \tilde{\mathbf{p}} + \mathbf{T}_q \mathbf{C} \tilde{\mathbf{q}}. \tag{2.50}$$

So with no tangential displacements, $\tilde{\mathbf{q}}$ may be found from $\tilde{\mathbf{p}}$

$$\tilde{\mathbf{q}} = -(\mathbf{T}_q \mathbf{C})^{+} \mathbf{T}_p \mathbf{C} \tilde{\mathbf{p}} \tag{2.51}$$

and the slope in the contact region, $\mathbf{s}$, is

$$\begin{aligned} \mathbf{s} &= \mathbf{S}_p\mathbf{p} + \mathbf{S}_q\mathbf{q} && (2.52) \\ &= \mathbf{S}_p\mathbf{C}\tilde{\mathbf{p}} - \mathbf{S}_q\mathbf{C}(\mathbf{T}_q\mathbf{C})^{+}\mathbf{T}_p\mathbf{C}\tilde{\mathbf{p}} && (2.53) \\ &= \left(\mathbf{S}_p - \mathbf{S}_q\mathbf{C}(\mathbf{T}_q\mathbf{C})^{+}\mathbf{T}_p\right)\mathbf{C}\tilde{\mathbf{p}}. && (2.54) \end{aligned}$$

The slope map for adhesive contacts, $\mathbf{S}_a$, may now be defined as

$$\mathbf{S}_a = \mathbf{S}_p - \mathbf{S}_q\mathbf{C}(\mathbf{T}_q\mathbf{C})^{+}\mathbf{T}_p \quad (2.55)$$

and the surface pressures can be found from

$$\begin{aligned} \mathbf{p} &= \mathbf{C}(\mathbf{S}_a\mathbf{C})^{+}\mathbf{s} && (2.56) \\ \mathbf{q} &= -\mathbf{C}(\mathbf{T}_q\mathbf{C})^{+}\mathbf{T}_p\mathbf{p}. && (2.57) \end{aligned}$$

Figure 2.13 shows the effect of full adhesion on the surface pressure and subsurface strain for flat indenters with a contact width of 4 mm. The resulting normal pressure is again parabolic and agrees with the results of (Bentall and Johnson, 1967; Nowell and Hills, 1988). The resulting shear tractions are not so well behaved due to the truncation of the fourier series. Results obtained using an integral equation approach, (Nowell and Hills, 1988), show a discontinuity in the shear tractions at the edges of the contact. Due to Gibbs phenomena, this leads to ringing in the truncated fourier series. Again we must qualify these results by recalling that we are interested in the low frequency portion of the pressure distribution and hence can tolerate this ringing effect. The advantage of using this technique over the more numerically stable techniques of Bentall and Nowell is that we automatically get the solution in the fourier domain, thus it is easy to use just the low frequency portion of the result.

The odd part of the shear strain that is present in the adhesive contact adds to the even component of the subsurface normal strain, as could be predicted by equation 2.15, causing a narrowing of the subsurface normal strain impulse response. This is the second of two effects that can cause narrowing of the impulse response that would be predicted by a Hertz pressure distribution. The first was due to a narrowing of the pressure distribution due to the rigid core, the second is the additional normal strain due to an adhesive contact.

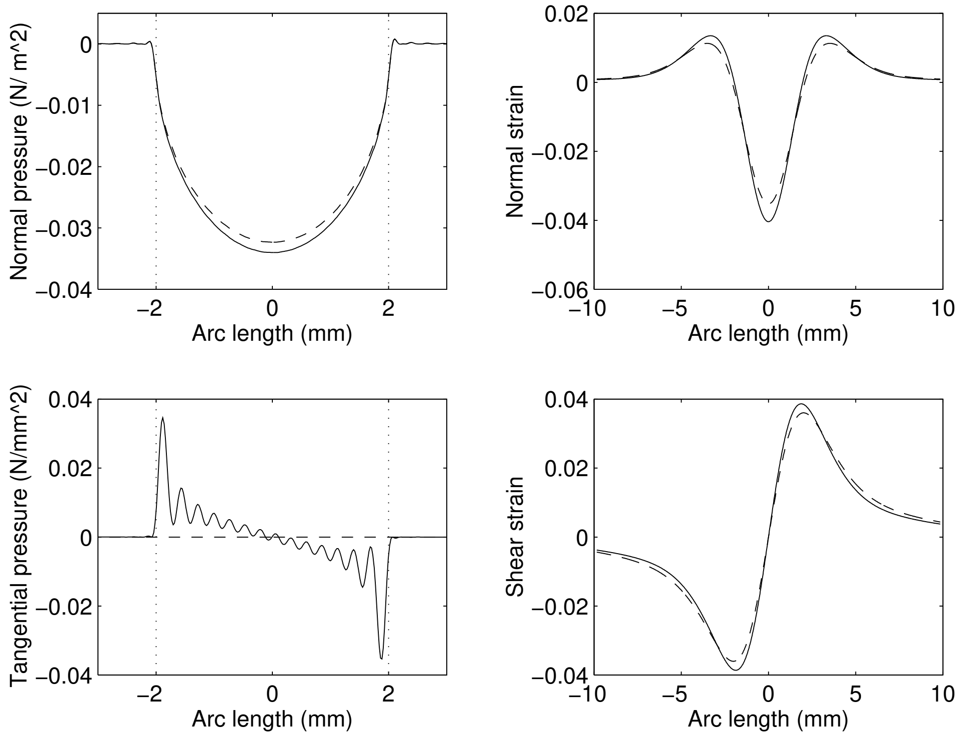


Figure 2.13: *Comparison of 4 mm flat contacts with and without adhesion. The solid line is for the adhesive contact, the dashed line is for the frictionless contact. The top left shows the normal surface pressure and the bottom left is the tangential surface pressure. The top right is the subsurface normal strain and the bottom right is the subsurface shear strain.*

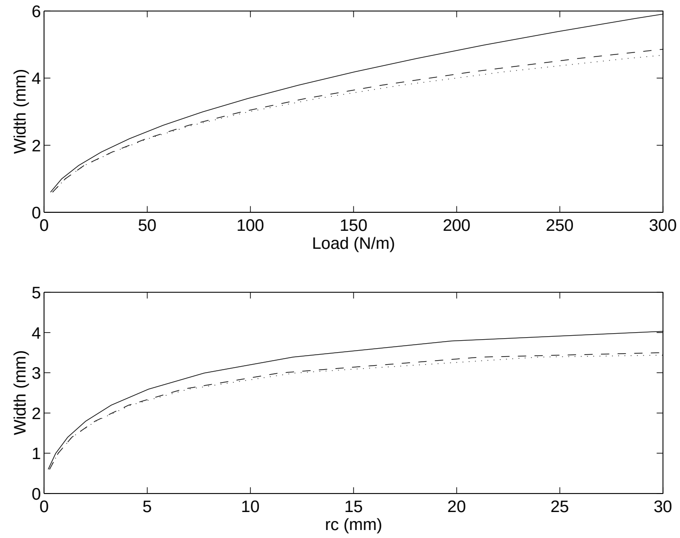


Figure 2.14: *Contact width for different contact models. The solid line is for the Hertz model, the dashed line is for the frictionless model, and the dotted line is for the adhesive model. The top plot shows how width varies with load for a flat contact. The bottom plot shows how width varies with r_c for a 200 N/m contact.*

2.4.4 Contact widths and contact models

We have discussed four different models that can be used to determine surface pressure: the Hertz contact, frictionless contact, contact with full sliding, and contact with full adhesion. For all these models it can be shown (see Section 2.6.1, equations 2.73 to 2.80) that determining the surface pressure for a rounded indenter is linear in the relative curvature if the contact area is held fixed. More precisely, if the pressure distribution due to a relative radius of curvature R is given by $\mathbf{p}$, then the relative radius of curvature for a contact with the same θ_w but with a load $\hat{\mathbf{p}}$ (that is $\mathbf{p}$ scaled by an arbitrary ζ) $\hat{\mathbf{p}} = \zeta\mathbf{p}$, is $\hat{R} = \frac{1}{\zeta}R$. Using this result we can compute the pressure distributions using each of the models for a flat contact and then scale the resulting pressure distributions to arrive at a desired P or r_c for a given contact area.

What is of particular interest is the predicted contact area for each of the models. We have already seen in Figure 2.12 that the the Hertz model predicts a larger contact area for the same load. Figure 2.14 shows how the contact area depends on load for a flat indenter and how it depends on r_c for a load of 200 N/m. With respect to the problem

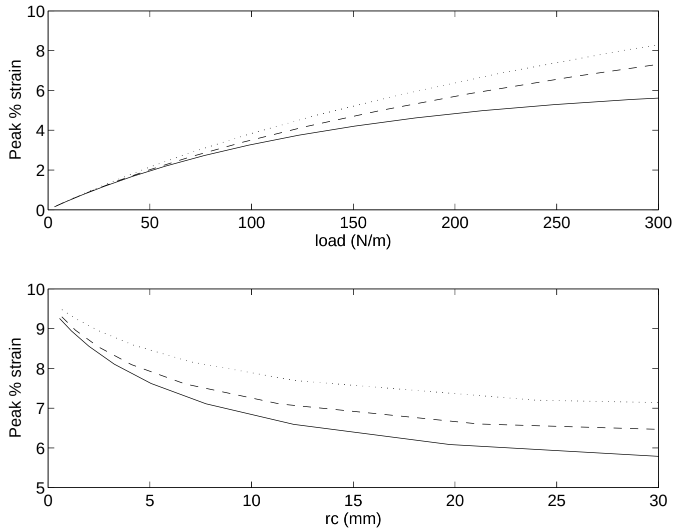


Figure 2.15: *Peak strain for different contact models. Solid line is Hertz model, dashed line is frictionless model, and dotted line is adhesive model. The top plot shows how peak strain varies with load for a flat contact. The bottom plot shows how peak strain varies with r_c for a 200 N/m contact.*

of predicting the subsurface strain in a tactile sensor, this indicates that there will be a narrowing of the impulse response over that predicted for the standard Hertz model.

The addition of adhesion to the contact model does not appreciably change the contact area, however it does significantly change the subsurface strain, as was seen in Figure 2.13. The even component of the normal strain is enhanced by the odd component of the tangential stress leading to a narrowing of the impulse response and a larger peak strain than would be predicted by the frictionless model. Figure 2.15 shows the variation in peak strain with r_c and P. The difference is significant for $r_c > 0.1 r_b$ and $P > 50$N/m.

Later, in Chapter 3, the significance of this result will be clearer when experimental sensor responses to edge and flat contacts are fitted to model parameters. We will see that for unbiased estimation of curvature the adhesive contact model must be used.

2.5 Bandlimited shape interpretation

The magnitudes of the singular values and the columns of the matrices $\mathbf{U}$ and $\mathbf{V}$ can be used to determine the ability of subsurface strain sensors to discriminate different

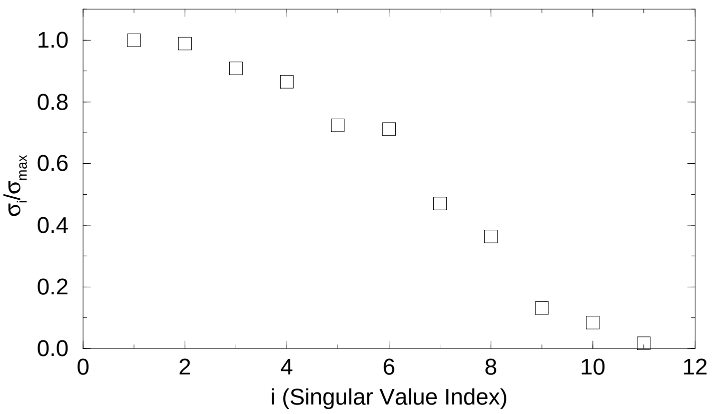


Figure 2.16: *Magnitudes of singular values of* $\mathbf{S}_p\mathbf{C}$*.*

shaped indenters on the surface. These vectors provide orthogonal basis vectors between indenter slope and fourier pressure. Since we know that only the low frequencies of the pressure distribution may be sensed, we are interested in seeing what kinds of indenters are distinguishable at a given cut-off frequency.

2.5.1 Analysis of the map $\mathbf{S}_p\mathbf{C}$

With r_a, r_s, r_b, and θ_w as before, the matrix $\mathbf{S}_p\mathbf{C}$ has 11 singular values with relative magnitude greater than 1/100 of the maximum singular value. Figure 2.16 shows the magnitudes. In fact there are only 8 singular values with magnitude greater than 1/10 of the maximum singular value. The singular values with smaller magnitude may be discarded since they will have little influence on the slope vector $\mathbf{s}$. That is, pressure distributions with coefficients orthogonal to the *columns* of $\mathbf{V}$ corresponding to the 8 most significant singular values will have a very small effect on the slope.

The small number of singular values is not surprising as it corresponds closely to the number of full periods of the highest frequency that we modeled that can occur in the contact region ($n_c = 300$, $\theta_w = 0.16$, number of periods = 7.6). Figure 2.17 shows how the number of singular values varies with contact width for $n_c = 300$.

The singular value decomposition gives us more than just a reduction in the problem size, it also helps us understand the shape-to-pressure map. The columns of $\mathbf{U}$ and $\mathbf{V}$ give matched pairs of pressure coefficients and shape in the contact region. Figure 2.18 plots the first four columns of the matrices giving the most significant directions in the

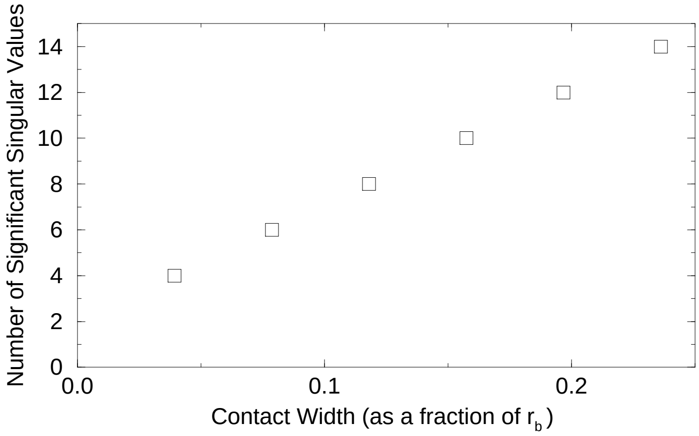


Figure 2.17: *Variation in the number of singular values of* $\mathbf{S}_p$ *with contact width.*

shape-to-pressure map. The following relationships are immediately clear

Odd Shape $\Leftrightarrow$ Even Slope $\Leftrightarrow$ Odd Pressure
Even Shape $\Leftrightarrow$ Odd Slope $\Leftrightarrow$ Even Pressure.

What is most important is that the spectrum of the odd shapes is near zero at the low frequencies. Given the sensor parameters of (Fearing, 1990) it would be very difficult to sense the odd part of the shape since, as Figure 2.4 shows, that information will be filtered out. In effect the low frequency component of the odd part of the shape can not be sensed without putting the sensors closer to the surface, with $\alpha > 0.85$.

In fact if one computes the singular values of $\mathbf{EC}(\mathbf{S}_p\mathbf{C})^+$ which maps from slope at a given location and contact width with no tangential tractions to the sampled subsurface normal strain there are only two significant directions, one for even shape, the other for an odd shape. The singular value for the odd shape is 0.11 of that for the even shape indicating that odd shapes are hard to sense.

2.5.2 Comparison of indenter classes

To further clarify the relevance of bandlimited interpretation techniques, the displacement, surface pressure, and subsurface strain fields for three different indenter classes were compared. Indenters were compared that had both the same load and contact area so as to generate as similar as possible subsurface strain profiles. A 10 mm cylinder, a

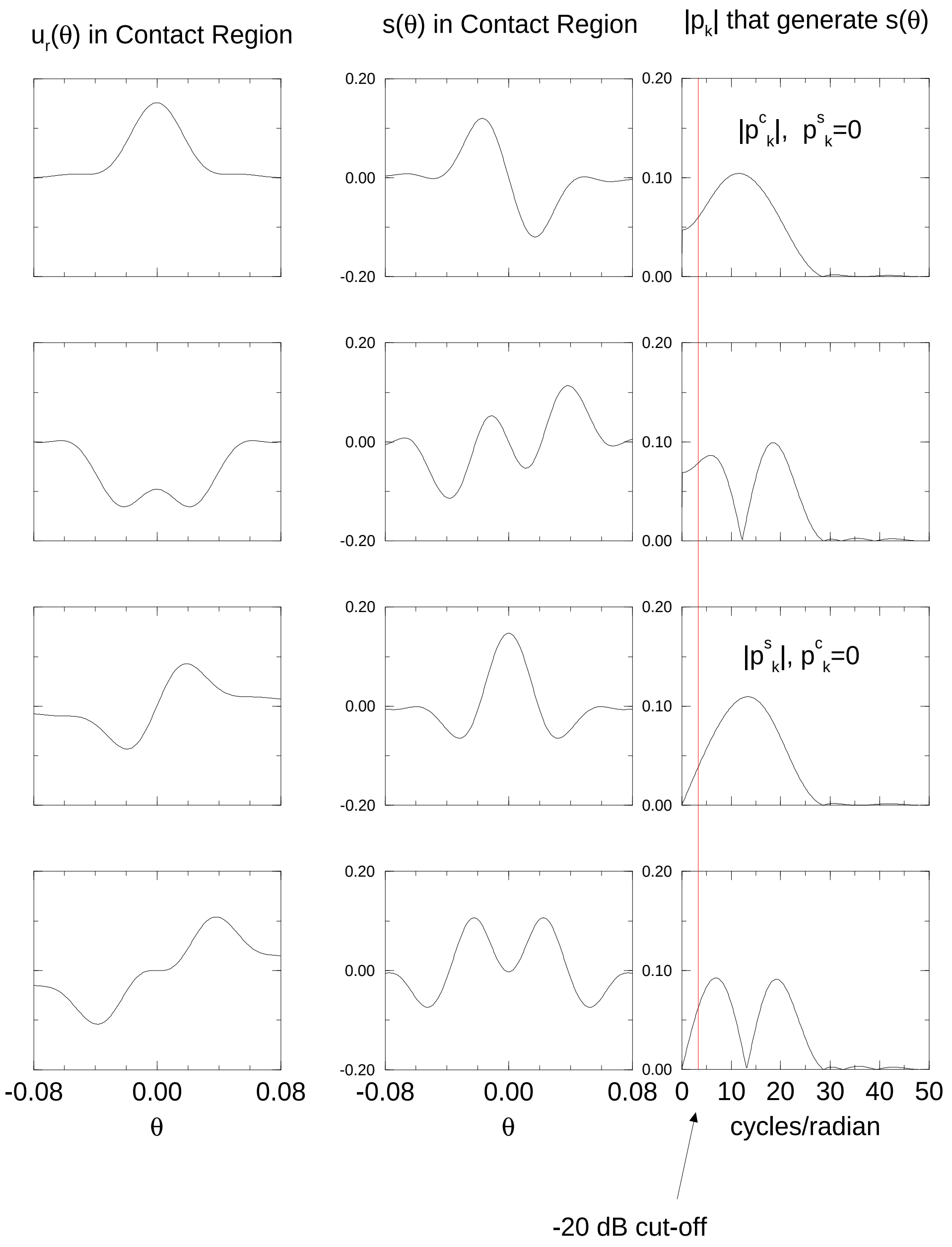


Figure 2.18: *Basis vectors for the map* $\mathbf{S}_p\mathbf{C}$*. The left singular vectors of* $\mathbf{S}_p\mathbf{C}$ *are shown in the center graph column as* $s(\theta)$*. The right singular vectors of* $\mathbf{S}_p\mathbf{C}$ *multiplied by* $\mathbf{C}$ *are shown in the rightmost graph column as* $|p_k|$*. The leftmost graph column shows the cumulative sum of the left singular vectors. This corresponds to the radial displacement,* u_r*, in the contact region.*

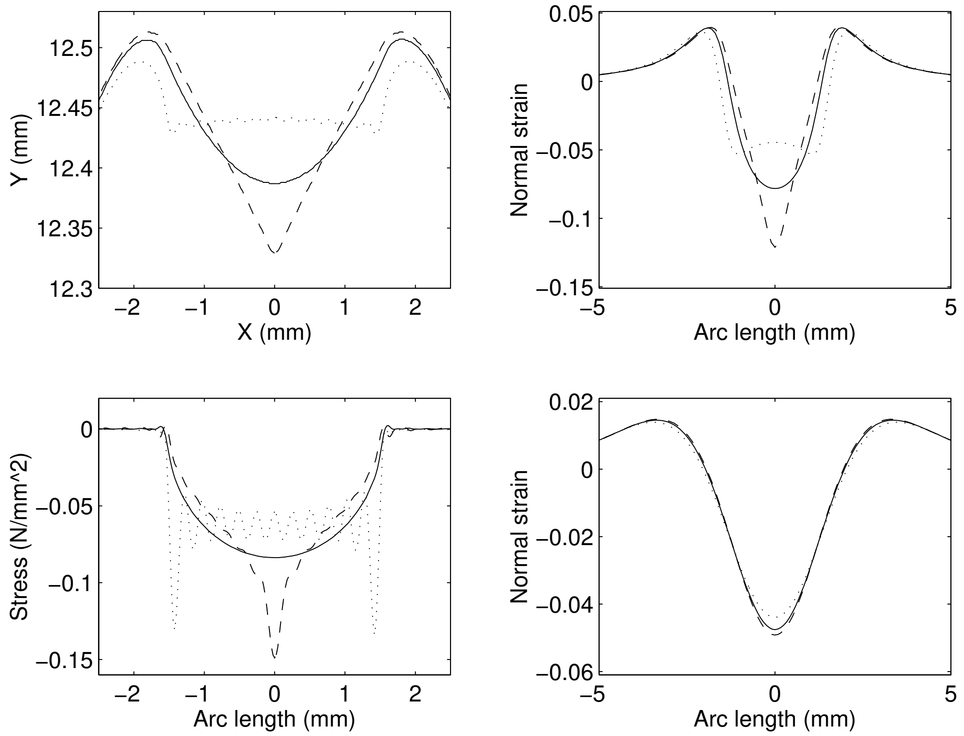


Figure 2.19: *Comparison of indenter classes for a round indenter (solid line), a wedge indenter (dashed line), and a punch indenter (dotted line) which all have the same contact area and total load. Upper left, displacement field; Lower left, surface pressure; Upper right, strain with $\alpha = 0.90, \gamma = 0.96$; Lower right, strain with $\alpha = 0.70, \gamma = 0.74$. In the upper left figure the displacement field is shown in the cartesian coordinates X and Y where the X axis is aligned with $\theta = 0$ and the Y axis is aligned with $\theta = \pi$.*

166 degree wedge, and a flat punch all generate a 3 mm contact area at 200 N/m. Figure 2.19 shows the displacement field, pressure distribution, and subsurface strain at two depths for the three classes. The displacement field is given in cartesian coordinates, not radial displacement and arc length, to show that the deformed profiles are indeed those of a punch, wedge, and round indenter. Figure 2.20 shows a Bode plot of the fourier coefficients for each of these different indenters. The pressure spectrums do not differ appreciably for spatial frequencies less than 2 cycles/radian. Recalling the Bode plots of the normal strain frequency responses in Figure 2.6, we can see that estimates of frequencies over 4.5 cycles/radian with a sensor with $\alpha = 0.70$ will be noisy since those spatial frequencies will be

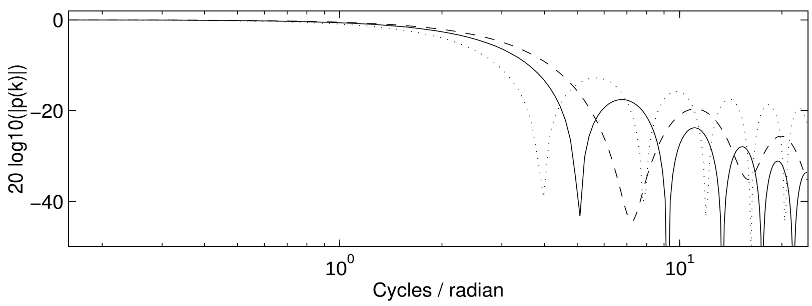


Figure 2.20: *Fourier coefficients for each indenter type: round (solid line), wedge (dashed line), and punch (dotted line).*

attenuated by at least 40 dB. However a sensor with $\alpha = 0.90$ has a -40 dB frequency of 12 cycles/radian, making discrimination possible with such a sensor.

The similarity of the pressure distributions at the low frequencies can be understood by thinking of the pressure distribution as a periodic shape multiplied by a window. In the frequency domain the window will appear as a sinc function. Since in the frequency domain all shapes will have a dc component convolved with the sinc function, the magnitude spectrums will all have a similar shape at the low frequencies. The width of the sinc function in the frequency domain is inversely proportional to the width of the contact, making the contact area the most discernible feature of the surface pressure after contact location and load. We will investigate this property in the next section where we hold the load and indenter class constant but vary the contact area.

2.5.3 Comparison of indenter curvatures

Given that the sensor we will be working with has $\alpha = 0.75$ and we have seen that indenter classification will be difficult in the presence of noise, we should investigate whether different scales of indenter of the same class are distinguishable. In this case we hold the load constant at 200 N/m, but vary the radius of the round indenter to see whether different radii indenters are distinguishable from low frequency information only. In this case we have picked 5 indenters with $r_c \in \{0.5, 1.6, 3.9, 12.3, 25.3\}$ mm. Figure 2.21 shows the surface displacement, surface stress, and subsurface strains at 2 different depths. Again the Bode plot of the magnitude of the pressure distribution is shown in Figure 2.22. In this case the magnitude spectrum of the pressure distribution does not differ appreciably for frequencies less than 1 cycle/radian. At 3 cycles/radian the frequency responses are clearly

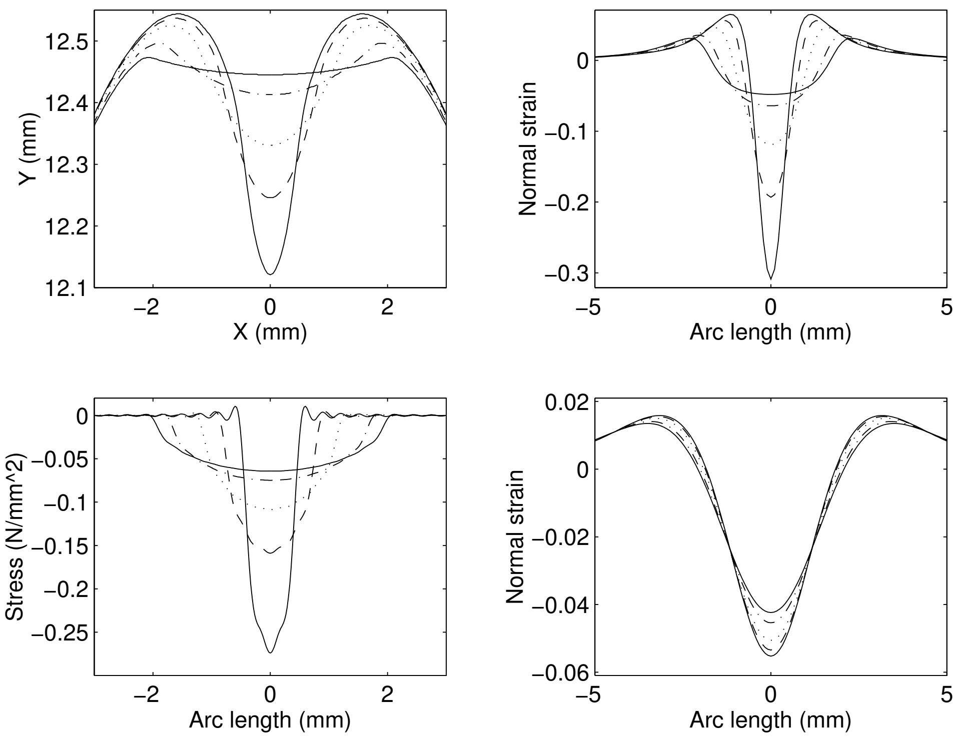


Figure 2.21: *Comparison of indentation for a fixed load (200 N/m) and varying $r_c \in 0.5, 1.6, 3.9, 12.3, 25.3$. Upper left, displacement field; Lower left, surface pressure; Upper right, strain with $\alpha = 0.70, \gamma = 0.74$; Lower right, strain with $\alpha = 0.90, \gamma = 0.96$. Again the displacement field is in cartesian coordinates.*

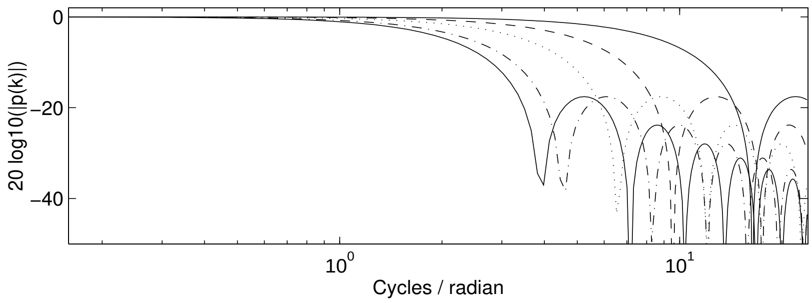


Figure 2.22: *Fourier coefficients for each indenter radius.*

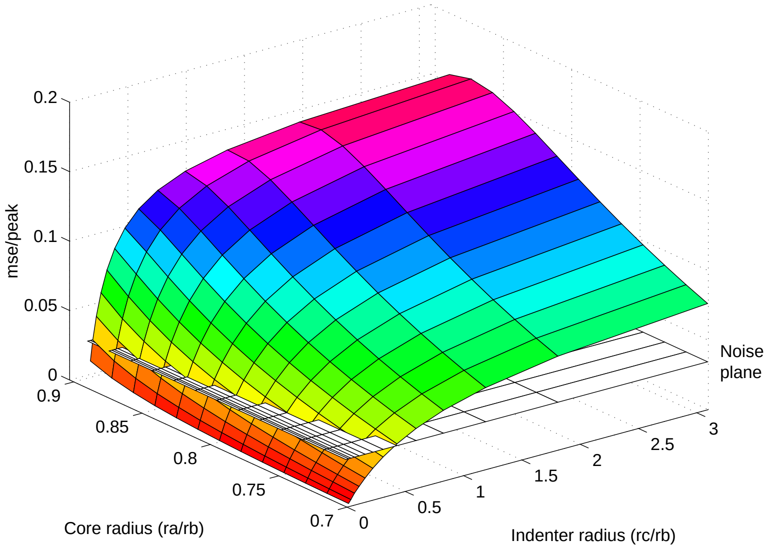


Figure 2.23: *Standard error as a fraction of peak strain between subsurface normal strains of knife edge and round indenters as a function of indenter radius and sensor depth. Load was held constant at 200 N/m. It is assumed that the strain profile was sampled at a high enough density to be interpolated. The noise plane is at 2 standard deviations of the noise of the sensor to be discussed in Chapter 3.*

different and should be distinguishable with $\alpha = 0.70$ and a noise level at -40 dB.

We can investigate the r_c at which a round indenter can be distinguished from knife edge indenter by plotting the mean square error between the strain profiles at a constant load. If we assume that the sensor noise is gaussian, then the round and edge indenters can be distinguished at a 95 % level when the square root of the mean square error exceeds 2 standard deviations of the noise. Figure 2.23 shows the error as a function of r_c/r_b and $\alpha = r_a/r_b$. A noise plane is also shown at 2 standard deviations of the typical noise for the sensor to be discussed in Chapter 3. The intersection of the noise plane with the error surface gives the boundary at which a round indenter my be distinguished from a knife edge. As would be expected, shallower sensors allow smaller radius indenters to be distinguished from a knife edge. For deep sensors with $\alpha = 0.70$, flat and edge indenters can be distinguished with noise levels as high as 3.7 % of the peak strain. If the noise is as low as 1 % of peak strain then deep sensors can distinguish an indenter with $r_c/r_b = 0.16$ from an edge indenter.

2.6 Equivalent wedge angle

Since we know that indentations with the same contact area are hard to distinguish for deep sensors, it is useful to determine when indenter classes generate the most similar subsurface strain profile. In this case we consider what the wedge angle is that results in the same contact area as a given radius cylindrical indenter at a given load. We expect to find that small contact areas and deep sensors make classification impossible, but as the contact area and core radius, r_a, increase the signal to noise ratio increases so that a wedge and the equivalent radius round indenter may be distinguished. We should note that all the results in this section assume frictionless indentation.

Using the closed form expressions for the pressure distributions of round and wedge indenters, equations (C.4) and (C.1) found in Appendix C, one can easily solve for the wedge angle that has the same contact area as a given radius indenter, r_c, at a given load, P, based on a half plane model.

$$\theta_p = \tan^{-1}\left(\sqrt{\frac{8\mu}{\pi P(1-\sigma)}\frac{r_b r_c}{r_b + r_c}}\right) \tag{2.58}$$

The closed form solution wedge pressure solution does not include the curvature of the finger nor does the Hertz model account for the rigid core so the equivalent wedge was also computed using pressure distributions generated by the SVD technique. We expect this technique to be unstable at small contact areas due to fourier leakage, but to be a better predictor of the equivalent wedge angle at large areas, as it includes the finger curvature and solid core.

For a fixed contact area we compute the equivalent wedge angle and radius at a given load $P = r_b \pi p_0$. Start with a nominal wedge angle $\hat{\theta}_p$, like $\pi/4$, and a nominal indenter radius $\hat{r}_c$, like 0.001 m. First we compute the wedge angle that arises from the given contact area and load.

Recognizing that $\hat{\mathbf{s}}_w$, the slope vector for the nominal wedge angle, may be decomposed into two parts, one due to the radius of the finger, r_b, and the other due to the wedge angle, $\hat{\theta}_p$,

$$\hat{s}_{w_i} \quad = \quad r_b\theta_i + r_b \mathrm{sgn}(\theta_i)\cot\hat{\theta}_p \tag{2.59}$$

we can vectorize the expression

$$\hat{\mathbf{s}}_w \quad = \quad \mathbf{s}_{w_{r_b}} + \hat{\mathbf{s}}_{w_{\theta_p}} \tag{2.60}$$

and then solve for the pressure distributions due to the separate components of the indenter shape

$$\hat{\mathbf{p}}_w = \mathbf{C}(\mathbf{S}_p\mathbf{C})^+(\mathbf{s}_{w_{r_b}} + \hat{\mathbf{s}}_{w_{\theta_p}}) \tag{2.61}$$
$$= \mathbf{C}(\mathbf{S}_p\mathbf{C})^+\mathbf{s}_{w_{r_b}} + \mathbf{C}(\mathbf{S}_p\mathbf{C})^+\hat{\mathbf{s}}_{w_{\theta_p}} \tag{2.62}$$
$$= \mathbf{p}_{w_{r_b}} + \hat{\mathbf{p}}_{w_{\theta_p}}. \tag{2.63}$$

Now define the scalar

$$\rho_w = \frac{p_o - \hat{p}_{w_{r_{b0}}}}{\hat{p}_{w_{\theta_{p0}}}} \tag{2.64}$$

and create a pressure distribution with the desired total load and determine the corresponding wedge angle. Let

$$\mathbf{p}_w = \mathbf{p}_{w_{r_b}} + \rho_w\hat{\mathbf{p}}_{w_{\theta_p}} \tag{2.65}$$

so that $\mathbf{p}_w$ has a total load of P. We know that the slope vector due to the resulting pressure distribution $\mathbf{p}_w$ is also composed of two parts

$$\mathbf{s}_w = \mathbf{S}_p\mathbf{p}_w \tag{2.66}$$
$$= \mathbf{S}_p(\mathbf{p}_{w_{r_b}} + \rho_w\hat{\mathbf{p}}_{w_{\theta_p}}) \tag{2.67}$$
$$= \mathbf{s}_{w_{r_b}} + \rho_w\hat{\mathbf{s}}_{w_{\theta_p}} \tag{2.68}$$
$$= \mathbf{s}_{w_{r_b}} + \mathbf{s}_{w_{\theta_p}} \tag{2.69}$$

so we may solve for the wedge angle

$$s_{w_{\theta_{p_i}}} = \rho_w r_b \mathrm{sgn}(\theta_i)\cot\hat{\theta}_p \tag{2.70}$$
$$= r_b \mathrm{sgn}(\theta_i)\cot\theta_p \tag{2.71}$$
$$\theta_p = \tan^{-1}(\frac{1}{\rho_w}\tan(\hat{\theta}_p)). \tag{2.72}$$

Figure 2.24 shows the equivalent wedge angle computed by using the above method and the closed form model of equation (2.58). Indenters with an $r_c < 0.5$ mm have the same contact area as that of wedge indenters with $\theta_p < 70^o$. With $r_c > 0.5$mm the equivalent wedge angle asymptotically approaches 90^o. It appears that there is a useful relation between radius and angle for wedge indenters with half angle greater than 70^o.

The closed form expression for the equivalent wedge angle does not yield the same angle as the method above since the closed form expression relies on pressure distributions computed for a half space. Neither closed form pressure distribution takes into account the

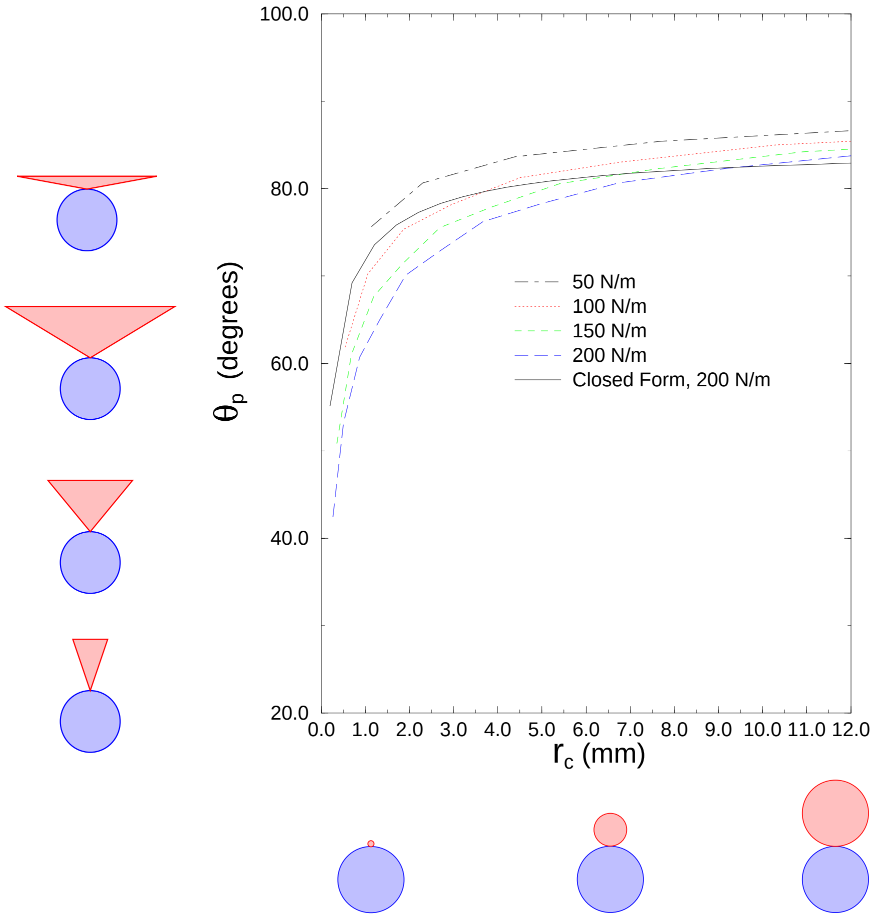


Figure 2.24: *Wedge angle that has the same contact area for a given radius indenter at a given load.*

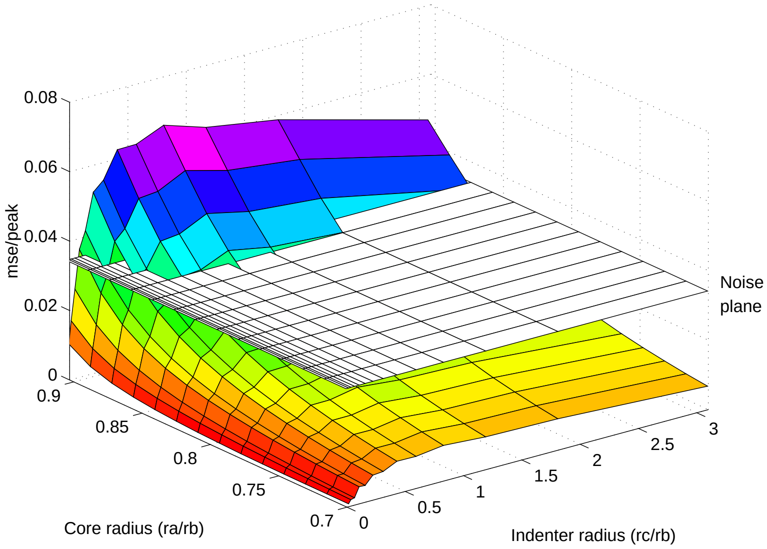


Figure 2.25: *Standard error as a fraction of peak strain between subsurface normal strains of equivalent wedge and round indenters as a function of indenter radius and sensor depth. Load was held constant at 200 N/m. It is assumed that the strain profile was sampled at a high enough density to be interpolated. The noise plane is shown at the level of 2 standard deviations of the noise for the sensor to be discussed in Chapter 3.*

rigid core and the wedge pressure distribution does not consider the curvature of the finger. One expects that for small r_c the closed form expression will predict a larger wedge angle than the SVD technique since the Hertz model will predict a larger contact area. A larger apparent contact area will result in a larger apparent wedge angle. At larger values of r_c the effect of the finger curvature will dominate and the contact area for a wedge indenter will be smaller than that predicted for the closed form model. This smaller contact area will be interpreted as a smaller wedge angle. This behavior is echoed in Figure 2.24.

To determine when such wedge indenters would be distinguishable from the equivalent radius indenter, the standard error between subsurface strain for equivalent area indenters was computed as a function of sensor depth and equivalent radius as is shown in Figure 2.25. It is interesting to see that no matter how thin the rubber layer is, when $\frac{r_c}{r_b} < 0.04$, or rather when the wedge angle is less than 70^o, noise levels would have to be extremely small to discriminate the two indenters. However, when the equivalent radius is the same as the finger radius the two indenter types reach a peak standard error for all sensor depths. This is the radius at which wedge and rounded indenters are maximally distinguishable. As the equivalent radius increases past the finger radius the standard error decreases slowly as both indenters start to look like flat indenters.

Again a noise plane is shown at the level of 2 standard deviations of the noise of the sensor to be discussed in Chapter 3. The intersection of the noise plane with the error surface is more interesting in this case than it was in Figure 2.23. In this case it is clear that by lowering the noise level through averaging or better electronics the range of sensor depths and indenter radii over which classification could be achieved will increase quickly due to the low gradient of the error surface. With the current noise level we must have $\alpha > 0.85$ for classification when the equivalent radius, r_c, is equal to r_b.

2.6.1 Quick pressure distributions for rounded indenters

For a fixed contact area one can compute the pressure distribution for the composite indenter radius, R, and the indenter radius, r_c, at an arbitrary load, p_0, from a given pair of load, $\hat{\mathbf{p}}_r$, and radius, $\hat{R}$ in a manner similar to that for the equivalent wedge.

$$\hat{s}_{r_i} = \frac{{r_b}^2}{\hat{R}}\theta_i = \frac{r_b(\hat{r}_c + r_b)}{\hat{r}_c}\,\theta_i \tag{2.73}$$

$$\hat{\mathbf{p}}_r = \mathbf{C}(\mathbf{S}_p\mathbf{C})^+\,\hat{\mathbf{s}}_r \tag{2.74}$$

$$\rho_r = \frac{p_0}{\hat{\mathbf{p}}_{r_0}} \tag{2.75}$$

$$\mathbf{p}_r = \rho_r\hat{\mathbf{p}}_r \tag{2.76}$$

$$\mathbf{s}_r = \rho_r\hat{\mathbf{s}}_r \tag{2.77}$$

$$s_{r_i} = \frac{{r_b}^2}{R}\theta_i = \rho_r\frac{{r_b}^2}{\hat{R}}\theta_i \tag{2.78}$$

$$R = \frac{1}{\rho_r}\hat{R} \tag{2.79}$$

$$r_c = \frac{r_b\hat{r}_c}{\rho_r r_b - (1-\rho_r)\hat{r}_c} \tag{2.80}$$

2.7 Summary

We have presented a linear elastic model for a cylindrical sensor and used it to analyze the impulse and frequency response and shape sensing capabilities of a cylindrical finger. In particular we have seen that determining indenter shape from subsurface strain amounts to solving a problem of bandlimited shape interpretation.

The frequency response by itself allows several important conclusions. First, from the Nyquist theorem, the sensor spacing requirements can be determined. It was thus assumed in subsequent analysis that sampled strain profiles may be interpolated to an arbitrary resolution since they are bandlimited. For good shape discrimination the sensors must be put close to the surface, at $\alpha > 0.85$, to detect the high frequency information. Secondly we found that shear strain sensing will not provide useful information for the shape sensing problem as it will be dominated at the low frequencies by the tangential loading. Since the human SAII afferents are sensitive to skin stretch, this may also indicate why the afferents have such a large receptive field. In other words, afferents sensitive to shear stress will respond to low spatial frequencies of the shear stress field. It was also shown that the spatial antialiasing filter for subsurface sensors is much better than that for surface displacement sensors due to its faster roll-off. With regards to the effect of the cylindrical geometry, the cylindrical geometry frequency response appears to be a shifted version of that of the half plane. This can be understood from the fact that the $k = 1$ solution of the fourier series corresponds to the DC term in the half plane model.

The $\mathbf{E}$ matrix was also introduced in this chapter as the map from the fourier coefficients of the surface pressure to the subsurface strain at the sensor locations. Thus with a known pressure distribution $\mathbf{p}$, the sampled strain values can be found as $\mathbf{Ep}$. Proper identification of this map for an actual sensor will be of great importance in the following chapter.

For an arbitrary indenter shape we have shown a way to compute an approximation of the pressure distribution up to a maximum spatial frequency. Comparison of pressure distributions computed in this manner agreed with those of (Nowell and Hills, 1988). In particular we have noted two effects, a rigid core and contact adhesion, that can lead to a narrower subsurface strain profile than would be predicted by a Hertz model. We have also seen that at low frequencies indenters of varying widths are more discriminable than indenters of varying class indicating that contact area is the easiest contact feature to determine from sampled strain after location and total load. We have also seen that indenter classification is possible in a range of indenter scales and sensor depths. This range increases as the noise level decreases. In general, classification is easier with a larger contact area.

These results will be used in the following chapter in which the design, construction, and performance of a tactile sensor are described. Of particular interest is the applicability of the linear elastic model in predicting the response of the sensor elements to different indenters.

Chapter 3

Sensor performance

In this chapter the design, construction, characterization and performance of a cylindrical tactile array sensor is discussed. Ideally, the linear elastic model developed in the previous chapter would predict subsurface strains due to a pressure distribution on the surface of the finger exactly, however in an actual sensor the material is no longer linear, isotropic, or homogeneous so some model error is expected. This chapter details the extent of model error due to inclusions in the material. First, however, the characteristics of individual sensing elements are determined to evaluate their sensitivity to nearby conductors and temperature as well as their linearity, hysteresis, and creep properties. Next techniques for calibrating the sensor and matching the element responses to predicted model responses are discussed. Finally, the reliability of indenter curvature estimation is explored.

It has already been shown that indenter curvature can be determined from static touch (Fearing and Binford, 1991), however recent results from finite-element modeling indicate that it is not possible to classify indenter types from subsurface strain measurements using linear elastic models (Ellis and Qin, 1994; Ricker and Ellis, 1992). Of course, with no model error or sensor noise, it should be possible to fit the strains predicted by an indenter type to simulated strain sensor data assuming that the nonlinear fitting program finds the optimum solution. If the error between the model predicted strains and the sensed strains can not be explained by sensor noise, then model error exists. If that model error is significant then linear elastic models are not appropriate for the rubber covered fingers with embedded sensors that are typically constructed. It was postulated by Ellis that the presence of rigid copper strips in an actual sensor would preclude the possibility of using linear elastic models which depend on isotropic and homogeneous materials. Although estimation of indenter curvature has already been demonstrated in experimental work using linear elastic models, Ellis notes that the reliability of the experimental estimates were not given.

This chapter addresses model error and its influence on estimation of indenter types, be it wedge, round, or a punch, and the indenter parameter, be it wedge angle, radius of curvature, or rigid body approach. We have already seen in Section 2.6 that indenter classification requires noise levels less than 0.05 % of peak strain with no model error for a finger with $\alpha = 0.70$. With the same noise level a round indenter with $r_c = 1.4$ mm and an edge indenter can be discriminated. Thus if the indenter class is known, fine shape discrimination can be accomplished since the contact area can be estimated. From contact area and load, indenter curvature can be determined. Again, these results depend on no model error. In this chapter it will be seen that while a linear elastic model may be calibrated to give reliable curvature estimates at one contact location, due to model error, curvature estimates at another location may be biased. The systematic nature of the bias suggests alternate calibration techniques and possible improvements in the sensor design and construction.

3.1 Sensing strain

As was discussed in Chapter 1, numerous tactile sensors have been designed and constructed. Unfortunately most of those sensors can not be used on a round sensor needed for grasping and manipulation. Commercially available sensors that can be used on round fingers do not provide the sensitivity required for shape discrimination. For this reason the design of (Fearing, 1990) was adopted with a few changes. Sensor spacing around the circumference was halved to reduce the aliasing that was predicted in Chapter 2. Dow Corning silicone rubber was used instead of isoprene rubber as silicone rubber demonstrated reduced hysterisis in preliminary experiments. A shield layer was added to reduce the proximity sensing effect that made the original design difficult to use during manipulation. Techniques for sensor fabrication were streamlined to allow quicker construction. Wire connections were improved so that cable flexing did not affect the measured capacitance. Finally, the electronics were modified to reduce noise and allow easy duplication through printed circuit board assembly.

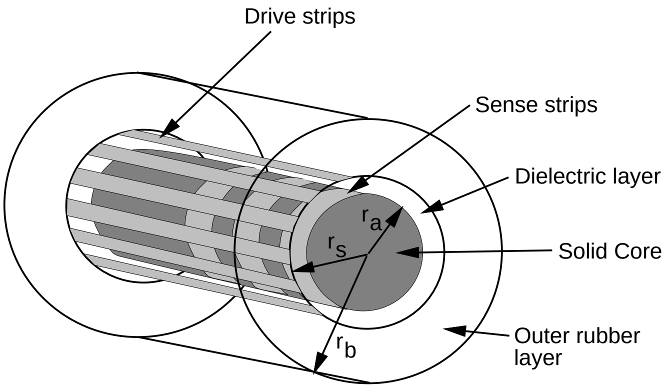


Figure 3.1: *Cylindrical tactile array sensor.*

3.1.1 Sensor construction

A cylindrical capacitive tactile sensor array was made from a delrin core, 0.050 mm thick copper foil, and Dow Corning HSII silicone rubber. Capacitor elements were formed by crossing strips of copper foil separated by a rubber dielectric. To create copper strips on the core a resist pattern was photomasked onto one side of the foil. The foil was then glued to the core and the unmasked copper removed by etching. A 0.2 mm dielectric layer was molded using the same rubber with 0.1 mm high, 0.6 mm diameter hemispheres spaced at 1.0 mm on a solid 0.1 mm thick layer. Copper foil masked with the outer trace pattern was glued to the dielectric and then etched to create reliably spaced upper strips. If a shield layer was used, a second set of copper strips were glued on top of the outer strips with thin nonconductive separation layer of nail polish. The dielectric and strips were then glued to the core together. The whole sensor was then placed in a cylindrical mold which was then filled with rubber. The copper strips were then cabled to an 18 pin connector after the rubber set. The steps are detailed in Appendix D. Figures 3.1 and 3.2 show the sensor.

Six different sensors were made, all with a delrin core of radius $r_a = 9.6$ mm and an outer radius of $r_b = 12.7$ mm, but differing dielectrics and sensor spacing as is shown in Table 3.1. To investigate possible nonlinearities caused by the bumpy dielectric, a solid dielectric sensor was constructed as well. Both shielded and unshielded sensors were made to compare proximity effects. According to the results of Section 2.2, the sensors should be spaced no less than 0.13 radians, or 1.6 mm, apart based on $\alpha = 0.75$. The sensor spacings that were chosen straddled this value to investigate the effect of sensor density. Figure 3.3

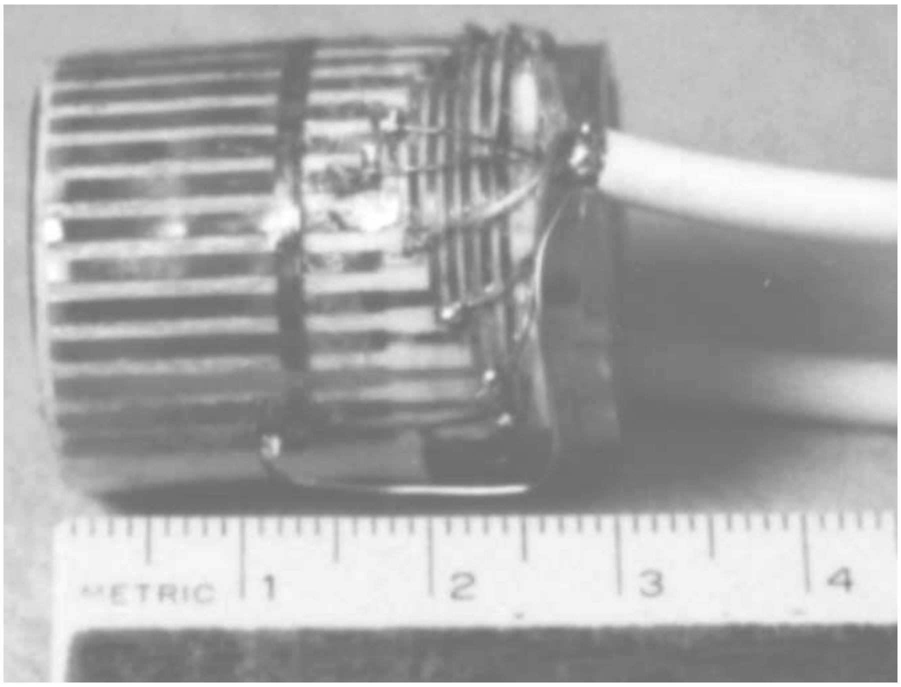


Figure 3.2: *A shielded tactile finger before the final molding.*

Finger number	Sensor spacing	Dielectric thickness	Dielectric type	Shield
0	1.52 mm	0.3 mm	Bumps	Yes
3	1.82 mm	0.5 mm	Bumps	No
4	1.82 mm	0.5 mm	Bumps	No
6	1.52 mm	0.2 mm	Solid	Yes

Table 3.1: *Finger construction parameters.*

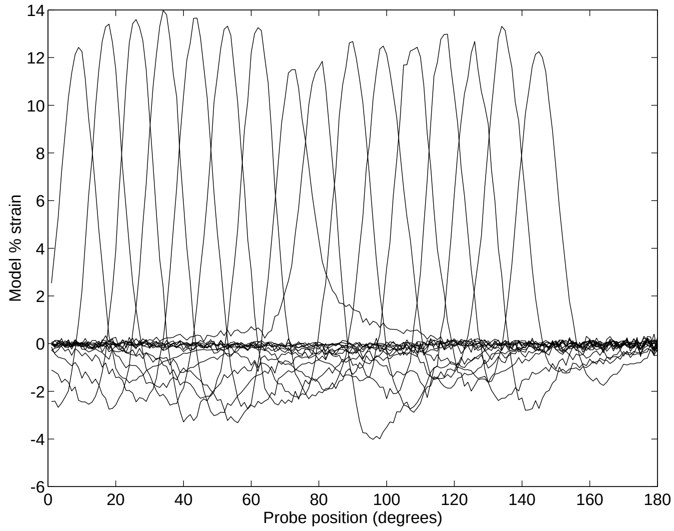


Figure 3.3: *Overlapping impulse responses for all the sensor elements of finger 0. The finger was touched at 1 degree intervals with a 200 N/m sharp edge. This plot shows the response of all the sensor elements around the circumference to a touch at each probe position . The response has been converted to % strain through gain and quadratic compensation. The tactile data was averaged over 20 samples.*

shows the response of the sensor elements to an edge contact with a load of 200 N/m. There is a great degree of overlap in the responses of the sensor elements as would be expected for unaliased sampling.

3.1.2 Electronics

Measurement of subsurface strain to 0.1 % with a capacitor with a plate area of 3.5 mm^2 and a separation of 0.4 mm required sensing changes of 0.1 % (0.3 femptofarads) in a capacitance of 325 femptofarads. Such a high sensitivity required a capacitance detection circuit utilizing a a coherently demodulated driver signal. The circuit used was a slight modification of that of (Berkemeier, 1990). Analog multiplexors were used to select a particular outer and inner strip at a time. The outer strips were driven by a 250 KHz sine wave and the resulting charge on the inner strips was amplified and coherently demodulated with a multiplier chip, as is shown in the block diagram of Figure 3.4. Detailed schematics for the circuit are given in Appendix E.

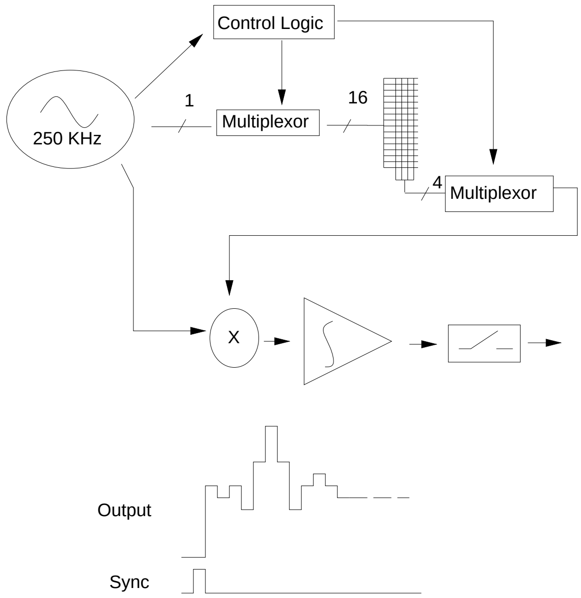


Figure 3.4: *Block diagram of electronics.*

Following (Fearing, 1990), we can relate the change in output voltage to normal strain as follows

$$\frac{v_s - v_o}{v_s} = \frac{u_r(r_a) - u_r(r_s)}{r_a - r_s} \simeq e_{rr}(r_s) \tag{3.1}$$

where v_o is the voltage output when nothing is touching the sensor and v_s is the voltage under load. Since the sensor output will not correspond directly to strain, the sensor array output will be given by $\mathbf{d}$. In the calibration process the relationship between e_{rr} and $\mathbf{d}$ is determined.

3.2 Calibration apparatus

To determine the mechanical and proximity sensing properties of the sensor an accurate positioning and force measuring device was required to deliver indenter touches at known locations and measure the applied contact forces. Figure 3.5 illustrates the robotic probing mechanism used. The prober has 3 degrees of freedom: θ_c, the contact location, r_p, the distance from the probe to the center of the finger, and ϕ, the angle of the indenter to the surface. The Sawyer motor two axis positioner is accurate to 0.030 mm and the rotational axis encoder has a resolution of 0.006 degree so the position resolution at the

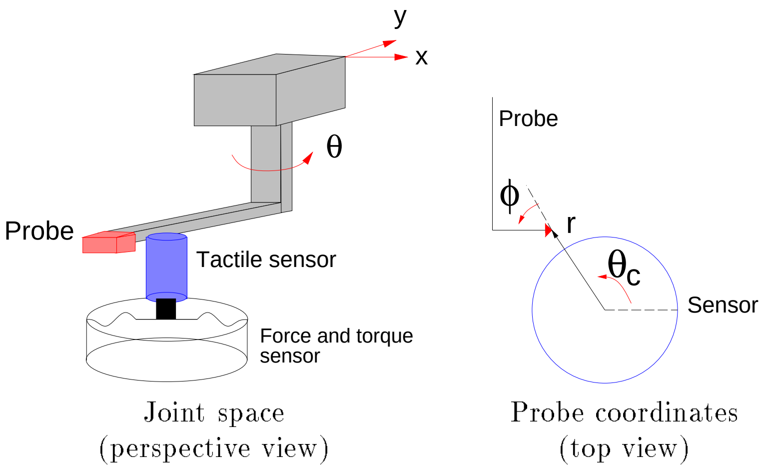


Figure 3.5: *Precision calibration prober.*

probe tip, which is 140 mm from the motor center, is 0.065 mm. The force sensor has a resolution of 0.05 N in the plane.

Experiments were conducted with a set of round probe tips made from machinable wax with r_c varying from 1.59 mm to 25.25 mm. In addition a 90^o corner and a flat indenter were used. All indenters had a length of 4 mm and were applied so that the 4 mm length was along the axis of the sensor. This length was chosen to correspond to the dimension of the sense strip, attached to the core, along the axis of the sensor. Indenters were applied directly above the sense strip at varying contact locations given by θ_c. Use of a longer indenter would have better suited the plane strain assumption, however such long indenters would be difficult to manipulate. It was decided that the averaging along the length of the sensor due to the rectangular sensing elements, 3.8 mm along the length and 0.9 mm around the circumference, would mitigate the effects of the finite indenter length. A few experiments were also carried out with 12 mm long indenters and no appreciable difference in the sensor response was found.

The output of the sensor circuit described in the previous section was converted to digital values by an analog to digital converter on a VME bus card. The tactile and force sensor data could then be read while different indenters were touched to the tactile sensor by the robot. Appendix F describes the software used to integrate the robot and sensors.

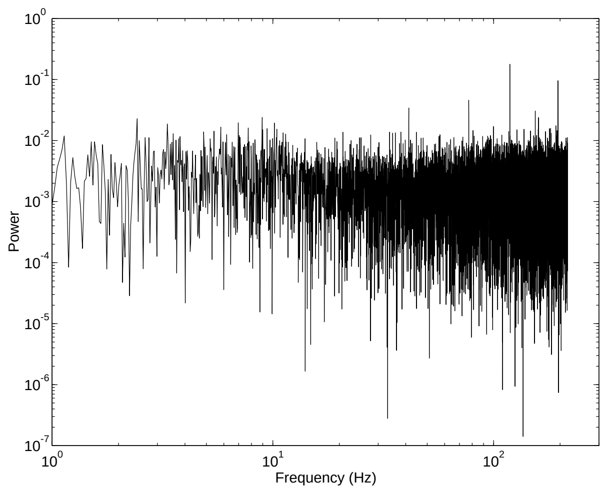


Figure 3.6: *Power spectral density of the sensor noise. One element of the tactile sensor array was sampled 16,384 times at 432 Hz. The maximum output was taken to be a 10% change.*

3.3 Characterization

The output of the sensor would ideally just depend on the normal strain of the material at which the sensor is located. Unfortunately there are many other effects that can contribute to the sensor value that need to be characterized. Besides random noise in the electronics, the output also depends on temperature, time history of indentation, and proximity to conductors. We will determine the magnitudes of each of these effects and discuss methods for compensation.

3.3.1 Electrical noise

One element of the tactile sensor was sampled at 432 Hz to determine the magnitude and spectrum of the noise. The quantization level of the analog to digital converter was 0.05% and the standard error was found to be the same, 0.05%. Since the noise due to quantization is $1/\sqrt{12}$ times the quantization level, the noise level is roughly 3 times that that would be predicted by pure quantization noise.

The power spectral density was also computed and is shown in Figure 3.6. The spectrum is flat except for one peak at 120 Hz. It is not clear what the cause of this peak is. Possibly it could come from the product of two 60 Hz noise signals in the multiplier.

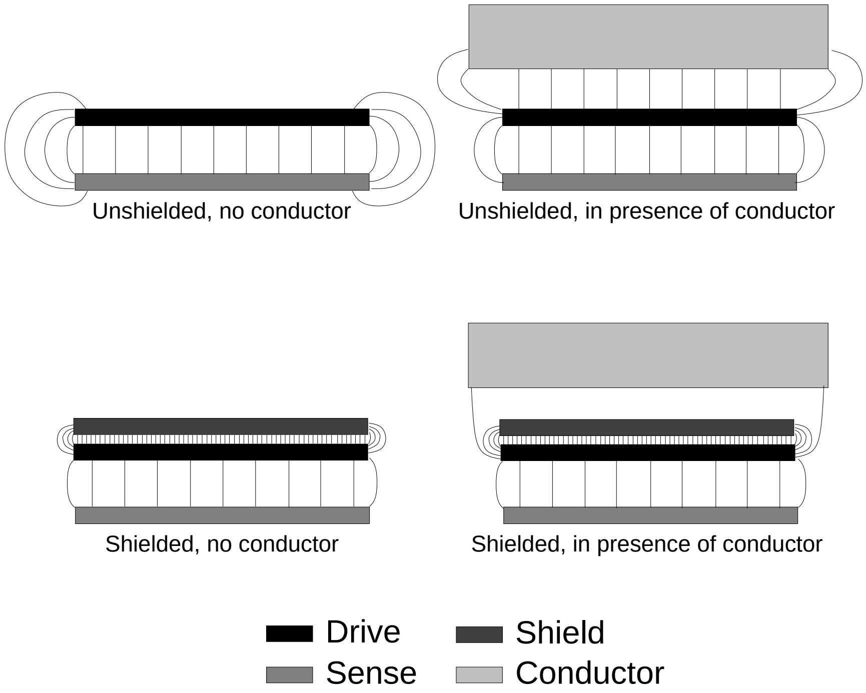


Figure 3.7: *Approximate field lines for shielded and unshielded sensors. When a conductor is brought near an unshielded sensor, the fringing field lines that used to terminate on the sense plate now terminate on the conductor decreasing the sensed capacitance. With the introduction of a shield layer there is little change in the fringing field lines from the drive to the sense plate so the capacitance changes only by a small amount.*

The noise increased markedly when the robot manipulator was brought near it. This is most likely due to electromagnetic radiation from the 20 KHz pulse width modulated motor driver beating with the 250 KHz tactile oscillator. The standard error of the noise increased to as much as 0.30% in the presence of the motors. By grounding the motor casing this effect was removed.

3.3.2 Proximity

For the capacitive sensors used here the plate separation is large compared to the plate area so the infinite plate assumption does not hold. In particular the fringing fields play an important role in the sensed capacitance. Figure 3.7 shows a sketch of the field lines for the shielded and unshielded sensors in the presence of a conductor. For an

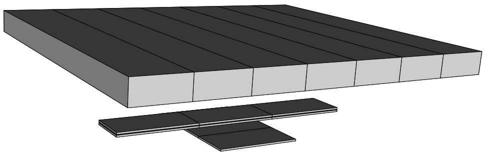


Figure 3.8: *Three dimensional model of the shielded variable capacitor in the presence of a large grounded conductor. The lines indicate the major discretization divisions for the finite element model.*

unshielded sensor the majority of the fringing field lines terminate on the sense plate. When a conductor is brought near the outer drive plate the fringing field lines now terminate on it, decreasing the accumulated charge on the sense plate. By adding a shield layer the majority of the fringing field lines terminate on the shield, reducing the effect of bringing a conductor near the sensor.

To numerically simulate the effect of a nearby conductor we used the FastCap program (Nabors and White, 1991). Two strips of copper, one 1.87 mm wide, the other 3.8 mm wide, were crossed and separated by a distance of 0.4 mm to simulate the sensor constructed. A 14 x 16 x 1 mm conductor was placed at distances from 1.0 mm to 15.0 mm from the top of the drive plate. For the simulated shielded sensor a second strip on top of the drive strip was added at a distance of 0.04 mm. Figure 3.8 shows the configuration.

The FastCap program computes a square capacitance matrix with dimensions equal to the number of conductors. By multiplying this matrix by a vector of conductor potentials the charge on the conductors can be determined. Figure 3.9 shows how the charge on the lower (sense) plate varies with the distance to the conductor for the shielded and unshielded design.

The simulations indicate that the shielded design is less sensitive to nearby conductors. To confirm this we constructed two sensors, one with a shield, the other without. The shielded sensor had a strip of copper the same width as the drive electrode directly above it. A manipulator was used to accurately position the conductor at 1 mm intervals from the surface of the sensor. The percent change in output of the sensor is shown in Figure 3.10. Both the simulated and experimental results indicate that the large plate separations for the capacitive array obligate shielding to avoid biased strain estimates when touching conductive objects.

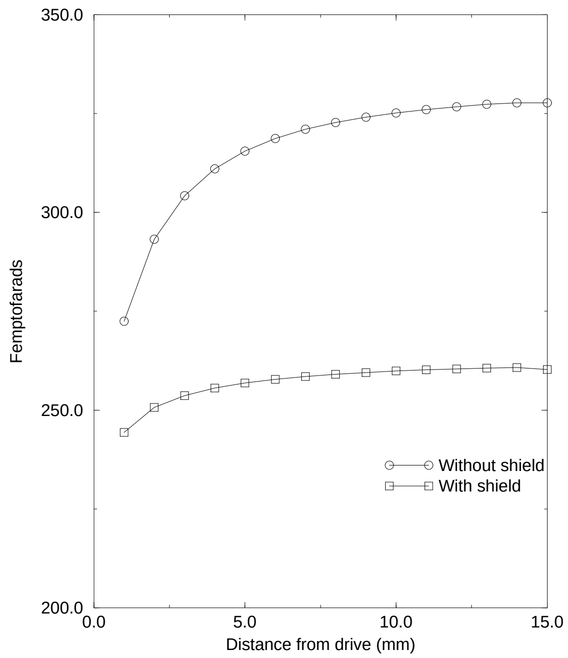


Figure 3.9: *Simulated capacitance in the presence of a conductor.*

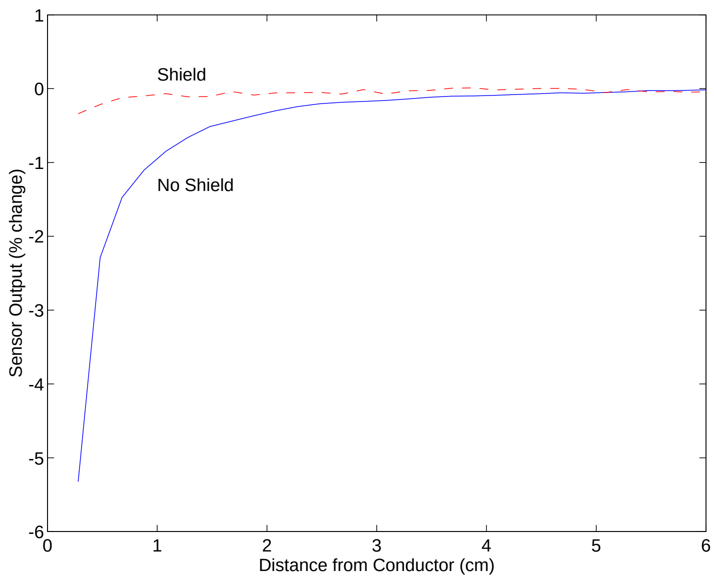


Figure 3.10: *Experimental sensitivity to a nearby conductor. The dashed line is for the shielded sensor. The tactile sensor data was averaged over 1000 samples at 100 Hz.*

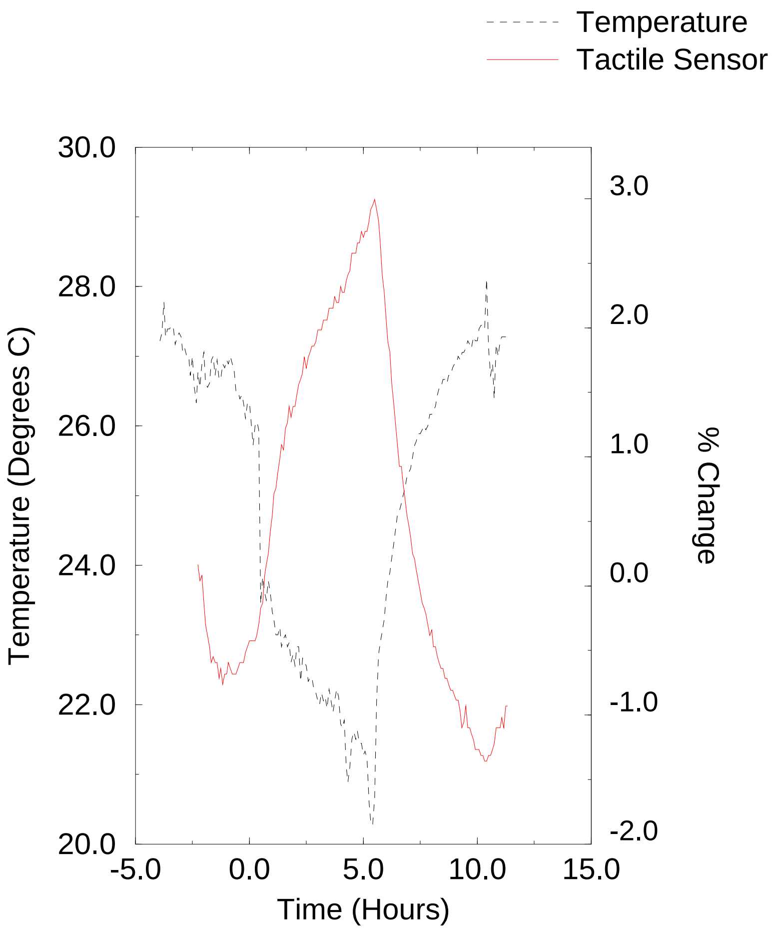


Figure 3.11: *Thermal sensitivity of the sensor. The tactile data was averaged over 100 samples.*

3.3.3 Thermal response

We had noticed that with earlier versions of the sensor and electronics the sensor offset required as much as 30 minutes to stabilize. We hypothesized that the circuit and sensor might be temperature sensitive. To test this hypothesis the sensor and room temperature were sampled at 5 minute intervals overnight. We knew that the lab temperature went through a heating and cooling cycle over this period, so we wished to correlate this slow temperature cycle with the sensor output. As can be see in Figure 3.11, there is a direct relationship between room temperature and sensor output. A linear regression between temperature and sensor output yields a slope of -0.81 with a standard error of 0.01. There was no quadratic term in the fit, although a hysterisis loop was evident.

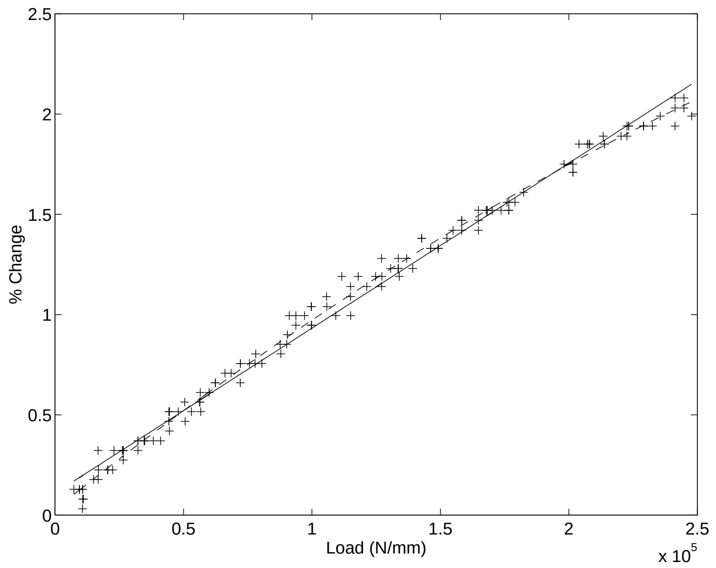


Figure 3.12: *Linear and quadratic fits to sensor data for finger 0, element 2. The crosses are data points, the solid line is a linear fit, and the dashed line is a quadratic fit. The tactile sensor data was not averaged.*

It was hypothesized that the sensitivity to temperature change was due to the thermal expansion of the rubber, however the linear coefficient of thermal expansion for silicone rubber is 0.03% per degree Centigrade. Our sensor had a temperature sensitivity coefficient of $0.81\frac{\%}{^{\circ}\mathrm{C}}$, a factor of 27 times greater. We must assume then that the sensitivity is due to the electronics. Small changes in both the integrator offset and the multiplier input adjustment due to drifts in the operational amplifiers and changes in capacitance could cause this magnitude of change.

3.3.4 Linearity

To test the sensor linearity a 12 mm edge indenter was attached to the calibration prober which was then touched to the tactile array directly above each sensor element. The contact force was increased linearly to 3 N (250 N/m) over 5 seconds while sampling the tactile array and force sensor at 33 Hz. The relation between sensor output and applied load was fitted to both a linear and quadratic function, as is shown in Figure 3.12.

In Table 3.2 the results of the linear and quadratic fits to the elements of three different fingers are given. Recall that finger 0 was a shielded, bumpy dielectric finger, finger

Finger	Gain (% / (N/mm))			Standard Error in linear fit			Standard Error in quadratic fit		
	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
0	21	27	41	0.09	0.13	0.19	0.05	0.07	0.10
4	32	68	124	0.09	0.18	0.39	0.07	0.13	0.22
6	26	42	63	0.06	0.18	0.39	0.05	0.09	0.14

Table 3.2: *Results of linear and quadratic fits. Errors are standard errors in units of % change in output. Minimum, average, and maximum values are over the 16 elements of the circumferential array.*

4 was an unshielded bumpy dielectric finger and finger 6 was a shielded solid dielectric finger. For each finger the minimum, average, and maximum of the sensor gain, linear fitting error, and quadratic fitting error are given.

From the gain data it is apparent that the shielded sensors are less sensitive. This is most likely due to the extra strip of copper used for the shield. It is interesting that the solid dielectric did not reduce the sensitivity, and in fact may have increased it. This contradicts the intuition that a solid dielectric would be less compressible. However, one must keep in mind that the solid dielectric was thinner than the bumpy one, increasing the capacitance. For the unshielded finger, the sensitivity was as high as 124 % change in output per N/mm. With a quantization and noise level of 0.05 %, it is then possible to sense a 1 mm long edge with a total load as small as 0.001 N. The gain values computed here are not in a useful set of units since they assume an edge contact. In Section 3.4.1 a model fitting procedure is used to determine gains that will convert % change in sensor output to % strain.

The fitting errors in Table 3.2 indicate that the data is better explained by a quadratic response than a linear response. With a quadratic fit the standard error is reduced to a level just above that of the noise, while a linear fit results in an error more than twice that of the noise. There was no appreciable difference between the solid and bumpy dielectric, indicating that both can be equally well explained by a quadratic fit. Since the unshielded sensor is more sensitive, one would expect its output to be more nonlinear over the same loading range. This is reflected in the somewhat larger error in the quadratic fit.

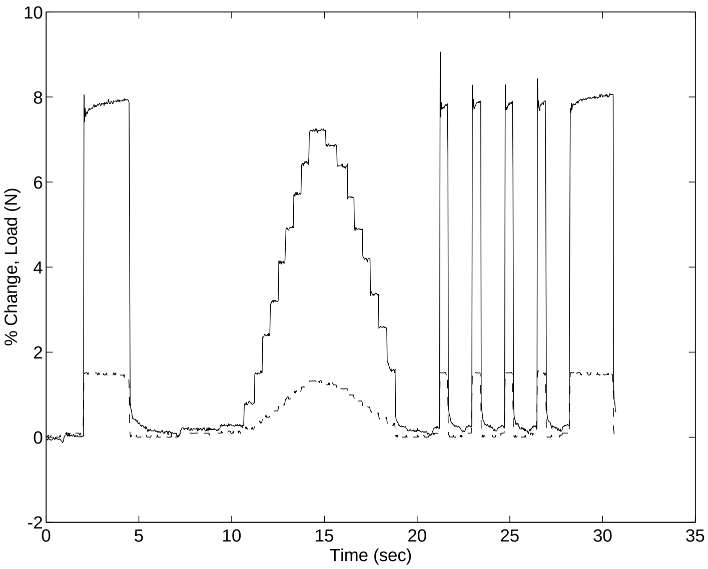


Figure 3.13: *Response of a single cell to cyclic loading. The solid line is tactile sensor and the dashed line is the force sensor.*

3.3.5 Hysterisis, creep, and relaxation

Perhaps more important than linearity is the time response of the sensor due to cyclic loading. To test this the sensor, again mounted on the force sensor, was touched with a 1.5 Newton load and held for 2.5 seconds. The load was then stepped up and down in 10 equal increments. Finally, 3 quick touches were applied. Figure 3.13 shows the output of the tactile and force sensor. As can be seen in the magnification of Figure 3.13 in Figure 3.14, there is a significant relaxation period for the rubber. While the 90% decay time is less than 0.1 seconds the 98% decay time is 1 second. Left long enough, the sensors returns to its original value, so the hysterisis is negligible. Creep, however, is another problem. As can be seen in the first touch of Figure 3.13, the sensor output increases as a decaying exponential with the same time characteristic as the relaxation. Compared to the dynamic response of isoprene rubber (Sladek and Fearing, 1990), the relaxation and creep times are approximately one half that of isoprene rubber.

3.3.6 Summary

Table 3.3 gives a summary of the sensor characteristics described in this section. It is apparent that the current sensor and electronics have a resolution of approximately 0.2 %

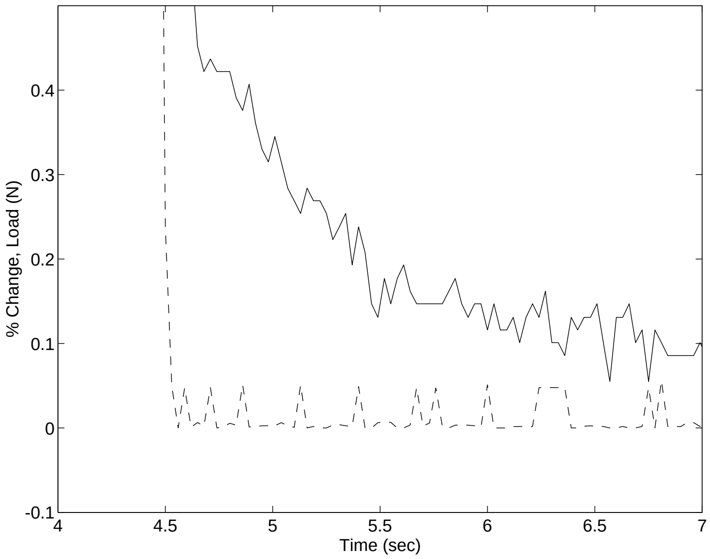


Figure 3.14: *Zoomed portion of the dynamic response. The solid line is tactile sensor and the dashed line is the force sensor.*

over all the static parameters, except temperature. Temperature changes and dynamic loadings with frequencies much over 10 Hertz pose difficult compensation problems. As this dissertation focuses on the static problem, temperature drift is the one parameter that will have to be compensated for in the experiments. For that reason, before every new contact, a new sensor output offset is recorded. Note that electrical and quantization noise is not so much a problem as all the other effects, especially proximity and temperature. Improving the noise characteristics of the electronics would help, but not as much as addressing the other problems.

The % changes in output that have been discussed do not correspond directly to model strain due to variances in element sensitivity. Typically model % strain is 1 to 5 times the sensor output, so strain errors will be 1 to 5 times the sensor noise. The next section discusses techniques for determining model strain from sensor output.

3.4 Fitting the E map

Having characterized the sensor, we can conclude that if we can fit a model to the sensor response which yields a residual error of less than 0.2 % of the sensor output then we can say that it explains the data. Residual errors much greater than this value will be

Characteristic	Measure	Finger	Value
Noise	std. error	All	0.05 %
Temperature response	Drift	All	0.81 % / ^oC
Proximity, unshielded	% Change at surface	4	-5.3 %
Proximity, shielded	% Change at surface	0	-0.3 %
Linear fit	std. error of residual	0	0.13 %
Quadratic fit	std. error of residual	0	0.07 %
Hysterisis	Permanent change	4	$<$ 0.1 %
Creep	Time to 98%	4	1 second
Relaxation	Time to 98%	4	1 second

Table 3.3: *Summary of sensor characteristics. % values are in units of sensor output, which is not the same as model strain. Depending on the sensor the model % strain can be from 2 to 5 times the sensor output.*

due to model error. It is the goal of this section to determine if a linear elastostatic, or even a general linear model, can explain the data to this precision.

Previous calibration methods have used the impulse response of the sensor exclusively. In this technique a sharp point indenter is touched to the sensor at carefully measured displacements from the sensor location. The response of the sensor to each touch is recorded and model parameters are fitted to the impulse response obtained. Since deformations are assumed to be linear, the impulse response completely characterizes the sensor response.

There are two ways of looking at the impulse response. One is the response of the whole array to a sharp edge, as is shown in Figure 3.15, the other is the response of one sensor element to touches all the way around the sensor, as is shown in Figure 3.16. The first of these is the *array* impulse response, the second is the *element* impulse response. The array impulse response corresponds to the information available during manipulation. Since we desire to determine the contact location at a resolution much higher than the sensor spacing, it must be interpolated. Both the raw, uncalibrated data and the calibrated data are shown interpolated in Figure 3.16. Correct interpolation requires that the sensor element output be adjusted so that all elements are in the same units, in this case % strain. By fitting model parameters to each element impulse response and then using this model to convert each sensor output to % strain the array response to a single touch can be predicted.

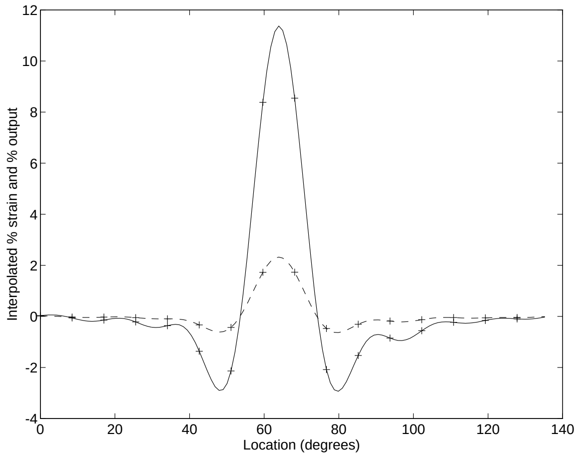


Figure 3.15: *Sensor array response to a sharp edge. The dashed line is the interpolated raw sensor data, the solid line is the interpolated data after gain and nonlinearity compensation. The array responses have been convolved with a sinc function to generate the interpolated response.*

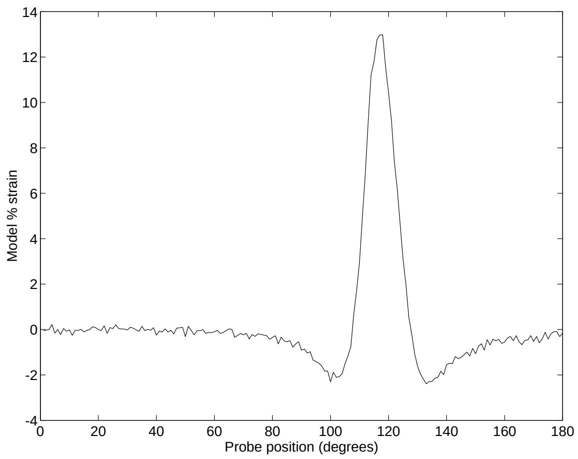


Figure 3.16: *Sensor element response to a sharp edge at 1^o intervals after nonlinearity and gain compensation.*

This amounts to computing the **E** map as described in Section 2.3.

An alternative to fitting the model parameters is to determine **E** directly by a linear least squares fit of the map. Using this technique one only assumes that the map from surface pressure to subsurface strain obeys linearity and superposition. It need not obey a linear elastostatic model which assumes a homogeneous and isotropic medium. This will allow compensation for some of the peculiarities in the impulse responses due to the copper strips and other non ideal characteristics of the sensor. The disadvantage to this technique is that it has less noise reduction.

Once the **E** map has been determined, sensor responses due to a pressure distribution, **p** and **q**, can be predicted. The pressure distribution can be computed based on a Hertz contact model or from the general method presented in Section 2.4. In the following sections the ability of the two calibration techniques to predict sensor element responses to sharp and flat loads will be discussed.

Before going into the details of the fitting procedures, we should first discuss the units to be used for standard error calculations. We would like to compare different fitting techniques and contact models so the fitting errors must be in the same units. If the standard error were expressed in terms of % strain then it would not be possible to compare strains for fits using different values of r_s and σ. Instead, all fitting error summary statistics are given as a percentage of the peak strain for an edge contact. Since the same load is used for all touches during the calibration procedure this error measure will allow comparisons between fits that yield different depths and poisson's ratios.

3.4.1 Model fitting

Using the calibration apparatus described in Section 3.2, 180 touches at 200 N/m were made with a 90^o wedge at 1^o intervals. The applied load and and sensor values were sampled and averaged over 20 readings. The contact force and location were used with a Hertz model for a $r_c = 0.1$ mm indenter to generate a pressure distribution for each contact. If we let the total number of touches be n_t and the highest fourier coefficient be n_c, then the fourier coefficients for each contact may be assigned to the columns of a matrix $\mathbf{P} \in \Re^{(2n_c+1)\times n_t}$. **P** is then the matrix of fourier coefficients for the set of calibration touches. Similarly the averaged sensor data for each touch, *after* the nonlinearity compensation discussed in Section 3.3.4, can be assigned to the columns of the matrix $\mathbf{D} \in \Re^{n_s \times n_t}$ where

n_s is the number of sensors. The nonlinearly compensated response of the ith sensor element to the jth contact is then D_{ij}.

After nonlinearity compensation, we expect the relation between output and strain, $\boldsymbol{\epsilon}$, to be given by a simple diagonal gain matrix, $\mathbf{G} \in \Re^{n_s \times n_s}$

$$\boldsymbol{\epsilon} = \mathbf{GD}.$$

Note that $\boldsymbol{\epsilon}$ is now a matrix. If the modeled strains are given by $\hat{\boldsymbol{\epsilon}}$,

$$\hat{\boldsymbol{\epsilon}} = \mathbf{EP},$$

then we wish to minimize, for each sensor element i,

$$\sum_j (\epsilon_{ij} - \hat{\epsilon}_{ij})^2 = \sum_j \left(G_{ii} D_{ij} - \sum_k (E_{ik} P_{kj}) \right)^2 .$$

In the case of the model-fitted map, $\mathbf{E}$ is a nonlinear function of the sensor location, θ_{s_j}, radius, r_s, and poisson's ratio, σ. For each sensor element, the nonlinear simplex method (Press et al., 1992) was used to minimize the squared strain error over these parameters. The simplex method was chosen due to its simplicity and robustness to initial parameter estimates. In each iteration of the optimization a gain value was chosen based on a linear least squares fit of the experimental strain values and those predicted by the nonlinear parameters. This additional gain parameter is needed since we expect that due to sensor manufacture there will be disparities in the amount of glue or the condition of the dielectric under each element leading to variations among elements from the strain predicted by the model. Table 3.4 gives the fitted parameters for finger 0 including the standard error and a gain value for each element. To allow comparison with the sensor characterization, which was expressed in units of % output, the gain value is given as the % of peak strain per % output. For example, from Table 3.3 we know that the electrical noise is 0.05 % output, so for the first element of finger 0 the noise level is 1 % of peak strain for a 200 N/m edge contact. Figure 3.17 shows an example of one set of fitted data.

Table 3.5 gives summary statistics of the parameter fits over 3 fingers. The noise level is given for unaveraged data in units of % peak strain, which will vary depending on the gain of each element. With these fits to noisy data we wish to determine if the model error is significant. In comparing the noise level and the fitting error we must include the noise from both the tactile sensor and the force sensor. The force sensor noise is significant,

θ_c	% Peak / % Output	r_s	σ	Std error % Peak
143.0	20.5	9.77	0.47	1.77
132.9	19.2	10.31	0.41	1.28
123.8	57.4	9.77	0.48	3.95
115.3	41.4	10.43	0.45	2.26
105.9	35.6	10.36	0.43	2.23
97.4	42.3	9.91	0.46	2.05
87.7	44.0	10.22	0.47	1.62

Table 3.4: *Fitted parameters for the first seven elements of finger 0. The error is in units of % peak strain.* r_s *is the sensor radius and* σ *is Poisson's ratio.*

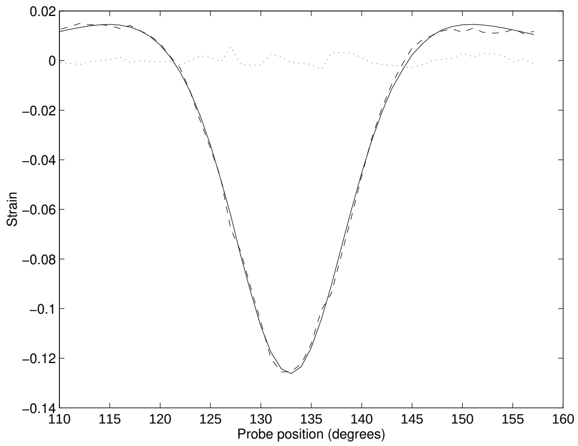


Figure 3.17: *Model based calibration results with an edge contact for second element of finger 0. Solid line is model fit, dashed is experimental data, dotted is error*

Finger	σ			r_s (mm)			Standard Error % Peak			Noise level % Peak		
	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
0	0.41	0.46	0.48	9.8	10.3	10.7	1.28	2.16	3.95	0.95	1.72	2.87
4	0.36	0.43	0.48	9.8	10.1	10.7	1.04	2.46	4.29	0.42	0.97	1.50
6	0.38	0.46	0.50	10.7	10.9	11.1	1.91	3.62	5.99	0.88	2.38	4.39

Table 3.5: *Results of model fits for 3 fingers. Minimum, average, and maximum values are over the 16 elements of the circumferential array.*

3.6 N/m for a 4 mm contact. Averaged 20 times and converted to equivalent % output using Table 3.2 the force sensor has a noise level of 0.024 % output, assuming the noise is independent and identically distributed. Using Table 3.4 this corresponds approximately to 0.5 % peak strain. The averaged tactile sensor data, which is independent of the force data, has a noise standard deviation of 0.4 % peak strain so the noise levels are roughly equivalent. Adding the standard deviations reveals a predicted error of 0.9 % peak strain. Comparing this with the errors in Table 3.5, we see that in the best case this limit is nearly achieved while on average the error is twice the noise level. This indicates that the model can explain the data to a precision limited by the noise for some of the sensor elements, however, on average, model error does exist and it is on the same order of magnitude as the noise.

The fitted Poisson's ratio is not the ideal 0.5 for incompressible rubber, but typically a little less. This has important consequences for the frequency response since a smaller Poisson's ratio implies that the material is behaving more like plane stress and not plane strain. This in turn implies that the frequency response is no longer 0 at 0 spatial frequency. Figure 3.18 shows the frequency response for different Poisson's ratios. We will see later that with a non-zero dc component, $\mathbf{E}$ improves in conditioning and thus is more stably inverted. It is interesting to see that the unshielded sensors fit to smaller σ values than the shielded sensors. These smaller values are similar to that found by (Fearing, 1990).

Based on the successful calibration for the edge contact, calibration was also performed for both flat and edge contacts. That is, a flat contact was touched to the sensor at 1^o intervals just like the edge contact. The $\mathbf{P}$ matrix then included both contact sets using a Hertz model for the flat contact. Figure 3.19 shows the fitted strain profiles. In

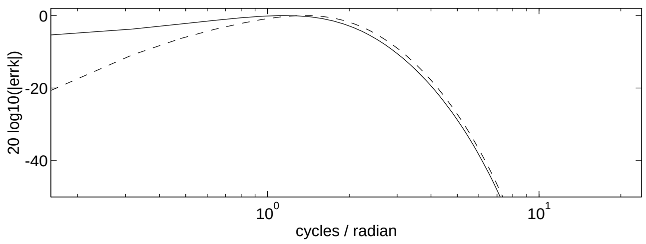


Figure 3.18: *Bode plot of subsurface normal strain frequency response for varying σ. Solid line is for $\sigma = 0.44$, dashed is for $\sigma = 0.50$*

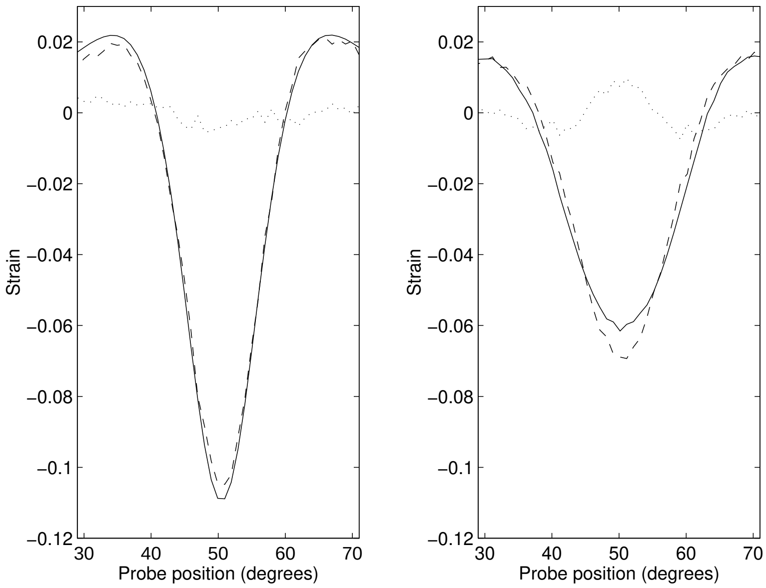


Figure 3.19: *Model based calibration results for simultaneous parameters fits to an edge and flat contact for the eleventh element of finger 0. Solid line is model fit, dashed is experimental, dotted is error*

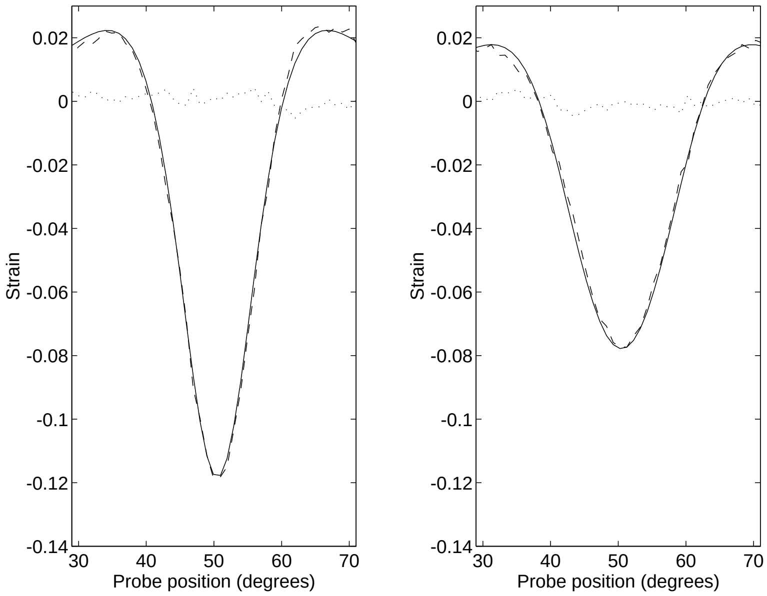


Figure 3.20: *Model based calibration results for simultaneous parameter fits to an edge and flat contact using a frictionless contact model for the eleventh element of finger 0. Solid line is model fit, dashed is experimental, dotted is error*

accordance with the discussion in Chapter 2 that predicted narrower pressure distributions in the presence of adhesion and rigid backing, it was found that the Hertz model predicted a wider strain response than was seen in the data. Since the parameters were fit to both contacts this resulted in the model predicting a higher peak strain for the edge contact and a smaller peak strain for the flat contact. This model error is significant and suggests that a Hertz model can not be used to predict pressure distributions and subsurface strain on a rubber sensor.

Alternatively, in Figure 3.20, we see the result of a calibration for edge and flat contacts using the frictionless contact model. The model is superior in explaining the data as the mean squared error, 2.25 % peak strain, is the same as for the edge only calibration. Table 3.6 gives summary statistics for fits to edge and flat contacts using the three different contact models. It is not clear whether the frictionless or adhesive contact model is a better predictor of the sensor response since the one which generates the smaller error is different for the two fingers. Since it is likely that the contact is best explained by a partially adhesive model this is not surprising.

Finger	Hertz			Frictionless			Full Adhesion		
	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
0	3.44	4.24	5.73	1.75	2.25	3.60	2.17	2.77	4.08
4	3.33	4.02	5.40	1.83	3.39	5.38	1.67	3.19	6.89

Table 3.6: *Standard errors as a percent of the peak strain for a 200 N/m edge contact for model fits to flat and edge contacts using different contact models.*

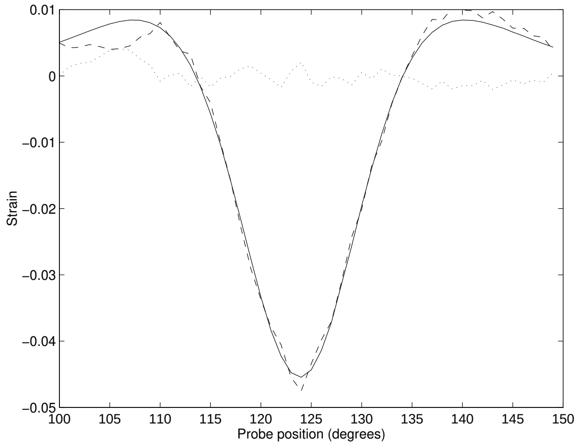


Figure 3.21: *Example of a poor model calibration for third element of finger 0. The solid line is the model fitted response, the dashed is the experimental data, and the dotted line is the difference.*

3.4.2 Direct identification

Not all of the sensor elements fit as well as the one shown in the previous section. Figure 3.21 shows a sensor element which has significant fitting error. Obviously, if the sensor were constructed perfectly all the elements should have the same response, however with a manual construction method this is not always possible. Typically the model errors are most significant in the tails of the impulse response. For this reason the standard errors in the previous section were computed over a 50^o range centered about the sensor. In this section we will see that direct identification of $\mathbf{E}$, or so-called empirical calibration, does not suffer from this problem. Both the tails and the center response can be predicted with minimal error.

To allow comparison with the model fitted results, an appropriate set of sensor

Finger	σ	r_s (mm)	Standard error % Peak			Noise level % Peak		
			Min	Avg	Max	Min	Avg	Max
0	0.45	10.3	1.86	3.18	5.07	0.95	1.73	2.80
4	0.43	10.1	3.17	5.61	8.13	0.39	0.96	1.50
6	0.45	10.9	2.91	4.38	6.96	0.86	2.36	4.28

Table 3.7: *Results of uniform gain fits for 3 fingers. r_s and σ were held constant so that strains at all elements would be equivalent for the same load.*

gains were needed. Fitting separate radius and and Poisson's ratios to each element provided a better fit, however it also meant that the % strain values were not the same for the same load centered above each sensor. For this reason a so-called "uniform" calibration was used performed using a constant σ and r_s and the resulting gains were used to scale the sensor data before the identification of $\mathbf{E}$. Summary statistics from the uniform calibration are given in Table 3.7.

We can accommodate for inconsistencies in construction by approaching the calibration as the problem of fitting a generic linear map, without the constraints imposed by linear elasticity. Contact models are still used to generate the pressure distribution matrix, $\mathbf{P}$, but $\mathbf{E}$ is treated as an unknown matrix that must be determined by solving for it in the equation

$$\mathbf{EP} = \mathbf{GD}.$$

Notice that the gain matrix from the model based calibration is retained so that the units on the right hand side are the same. Taking the transpose of this equation we have

$$\mathbf{P}^T\mathbf{E}^T = \mathbf{D}^T\mathbf{G}^T.$$

Using standard techniques from linear least squares or total least squares (Van Huffel and Vandewalle, 1991) the map $\mathbf{E}$ can be determined if $\mathbf{P}$ is full rank. For $\mathbf{P}$ to be full rank the fourier coefficients of the pressure distributions for the set of touches used, the columns of $\mathbf{P}$, must span the $2n_c + 1$ dimensional space of pressure fourier coefficients. One way to do this would be to apply sinusoidal pressure distributions, two at each spatial frequency. Since this is difficult, if $2n_c + 1$ impulses are applied at equally spaced intervals all the way

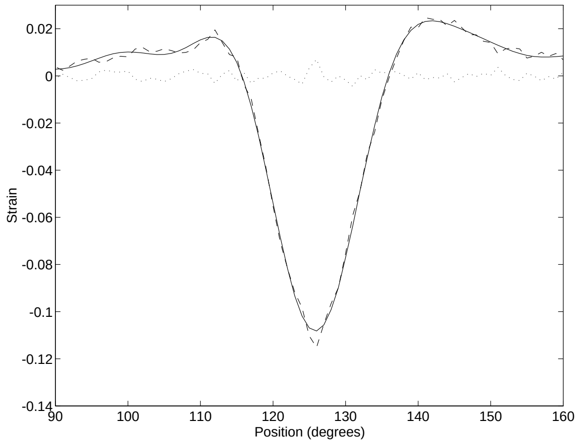


Figure 3.22: *Empirical calibration results for the third element of finger 0. Solid line is empirical fit, dashed is experimental, dotted is error*

Finger	Standard Error % Peak strain		
	Min	Avg	Max
0	0.93	1.41	1.76
4	0.45	0.92	1.93
6	1.07	2.30	2.84

Table 3.8: *Errors for empirical fits to edge contacts. These standard deviations are computed over touches over* 180^o*, not just a* 50^o *window around the center of the sensor.*

around the finger **P** will also have full rank. In our case only half the finger has sensors and touches were made around only half the finger resulting in **P** having half the rank it should. Hence **P** was regularized using SVD before the following computations were carried out.

E is determined directly from

$$\mathbf{E}^T = (\mathbf{P}\mathbf{P}^T)^{-1}\mathbf{P}\mathbf{D}^T\mathbf{G}^T.$$

The empirically derived **E** is much better at predicting the tails of the sensor as can be seen by comparing Figures 3.22 and 3.17. The errors, now computed over touches over the full 180^o are less than the errors for the model based fit. From Table 3.8 we see that the average for finger 4 is the 0.9 % error predicted by our noise model. The larger errors for fingers 0 and 6 are expected due to their reduced sensitivity and hence greater noise.

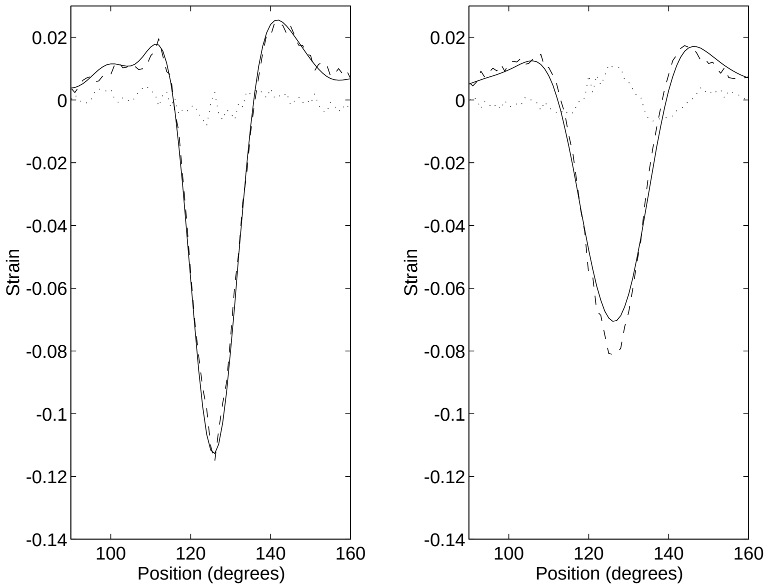


Figure 3.23: *Empirical calibration results for simultaneous parameters fits to an edge and flat contact for the eleventh element of finger 0 using a Hertz contact model. The solid line is the fitted data, the dashed is experimental, and the dotted is the error.*

It was also found that the empirical fit was no better at predicting the strain for flat contacts using a Hertz model, as similar results to those of Figure 3.19 were obtained and are shown in Figure 3.23. As we know that pressure distributions vary nonlinearly with indenter shape and load it is not surprising that direct identification of the linear map **E** does not fit the flat contact any better than the model fitted **E**. If we again use the frictionless and adhesive contact models the fitting error is reduced to the same level as for the fit to the edge contact. Table 3.9 gives summary statistics for identifications of **E** using different contact models. In this case the adhesive contact model is a better fit for both

Finger	Hertz			Frictionless			Full Adhesion		
	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
0	1.73	2.10	2.42	1.36	1.59	2.09	1.04	1.42	2.02
4	1.47	2.28	3.11	0.87	1.71	2.36	0.49	1.27	2.14

Table 3.9: *Standard errors as a percent of the peak strain for empirical* **E** *fits to flat and edge contacts using different contact models.*

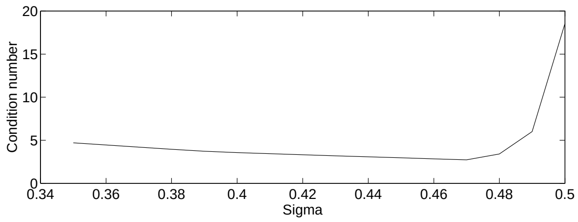


Figure 3.24: *Conditioning of* **E** *as a function of Poisson's ratio,* σ.

fingers indicating that it may be the best model to use.

The disadvantage of the identification technique is that there are no fitted parameters which can be related to the physical parameters of the sensor. It is also not possible to predict the sensor response to shear loading without also applying a set of contacts with a shear load. Our current calibration apparatus does not allow independent application of normal and shear loading, making this difficult. The model based calibration can predict response to shear loading by using the derived parameters. However, the superior ability of the empirical technique to predict sensor responses clearly justifies it use.

In fact it is best to use both techniques together. Parameter estimates and gains from a model based calibration can be used to give sensor values in useful set of units, % strain, that allow comparisons between the techniques and other theoretical results. In the next section we will explore the SVD of the identified **E** and show how it can be related to the linear elastic model developed in Chapter 2.

3.4.3 Conditioning of the E map

One way of analyzing the linear least squares **E**, or empirical **E**, is to compare its SVD to that of the model fitted **E**. Since they are both linear maps between the same spaces they should have similar characteristics. Both maps are well-conditioned. The model fitted **E** has a condition number of 2.7 and the empirical **E** has a condition number of 3.3 Actually the conditioning of **E** strongly depends on σ as can be seen in Figure 3.24 where the condition number is plotted as a function of σ. With $\sigma < 0.49$ inversion of the maps is possible and relative noise amplification of the different components of the singular vectors will be no more than 5.

To get a better idea of what the better conditioned directions of the **E** map are we can look at the left and right singular vectors, or the columns of **U** and **V** respectively.

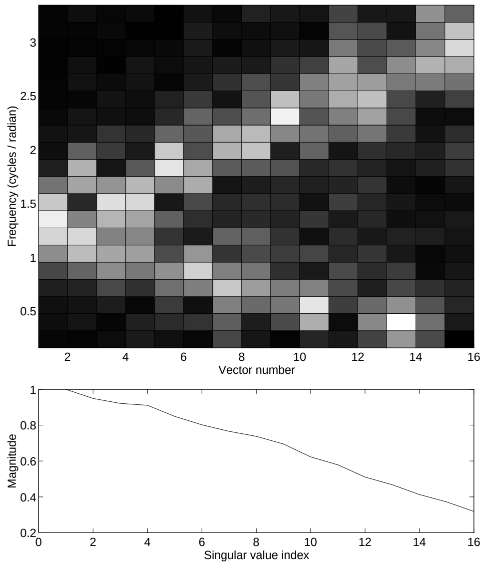


Figure 3.25: *Grayscale plot of the right singular vectors of the* **E** *map for finger 0 with their corresponding magnitudes. The right singular vectors are fourier coefficients of the pressure distribution. The best conditioned frequencies are at 1.5 cycles/radian while the worst are at the low and high frequencies.*

The columns of **U** form a basis in sensor response space, the columns of **V** are in pressure fourier space. As would be expected by the frequency response in Section 2.2, Figure 3.25 shows that the well conditioned directions in **V** space have most of their energy in the center frequency of the frequency response. The more poorly conditioned directions have their energy concentrated in the very low and high frequencies. We already knew that the high frequencies would be hard to estimate, but it is interesting to see that the total load information contained in the very lowest frequencies is also difficult to obtain. Both the SVD of the empirical **E** and the subsurface strain impulse response of Chapter 2 indicate that estimation of the total load directly from the first fourier coefficient would be noisy. A better approach would be to fit fourier coefficients from a pressure distribution model to the estimated fourier coefficients.

3.4.4 Summary

This section has discussed the ability of linear elastic models and general linear models to predict subsurface strains in a tactile sensor. There are two linear elastic models of concern here, one that maps surface pressure to subsurface strain, the **E** map, which is linear, the other is the map from indenter shape to surface pressure, **S**, which is linear only for a fixed contact area. We have seen that **E** can predict strains to a precision commensurate with signal noise for a well constructed sensor, however if there are nonhomogeneous artifacts in the sensor response due to construction, the linear least squares approach will produce a better result. On the other hand, the model based technique allows the prediction of responses to shear loading, allows the prediction of stress and strain at arbitrary points in the sensor, requires fewer contacts for calibration and provides improved noise rejection.

It is clear that the shape-to-pressure map, **S**, can not be characterized by an unmodified Hertz contact. Due to friction and the rigid backing of the sensor the strain response is narrower and sharper than would otherwise be the case. There are significant computational drawbacks to using the frictionless or adhesion models, however, as computation of the pressure distribution requires the inversion of a large matrix. To reduce computation, one could use the observation that the normal pressure distribution for the frictionless model, which is very close to that of the adhesive model, is parabolic, hence a simple scaling relationship between it and the Hertz model could be developed. This scaling would squeeze and stretch the Hertz pressure as a function of load and indenter shape.

It has also been shown that the **E** map can be inverted to obtain the pressure distribution directly from strain, however, doing this will amplify the noise at the high and low frequencies. Inversion of **E** would allow the identification of contact parameters directly from a match between pressure fourier coefficients. In this next section we will discuss techniques for contact parameter estimation.

3.5 Shape from strain

From the characterization and model fitting results we have seen that it is possible to predict the sensor element values from a linear elastic pressure distribution. The next question is how to invert this model so that the indenter parameters, location, load, and curvature, can be determined from a single touch. In the previous section it was shown that although **E** is invertible, using its inverse will increase the noise. Instead a nonlinear fitting procedure is used that only requires the forward model. Since we know from the equivalent wedge formulation of Section 2.6 that wedge and round indenters will be indistinguishable for our sensor radius, only round indenters are used.

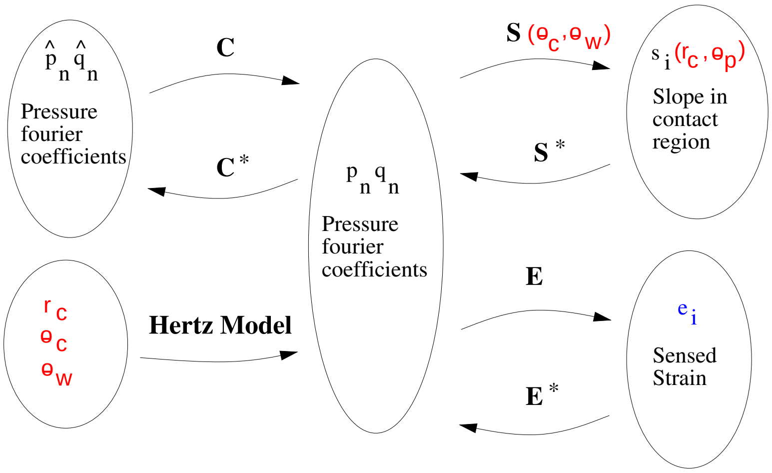


Figure 3.26: *The* **S**, **C**, *and* **E** *maps. The forward, or well-conditioned, direction for each map is from left to right.*

In Figure 3.26 the relationship between the various maps and contact models is shown with the forward directions going from left to right. It is clear that if we use a Hertz model there is an easy forward path from contact parameters to sensor values. A nonlinear fit to the contact parameters requires repetitively solving this forward path, computing the sum of the squared error between the model and sensor strain values, and updating the contact parameters to reduce the error.

To investigate the reliability of curvature estimates, first simulations using modeled strains with added gaussian noise will be fitted using various loads, curvatures, and contact locations, Those results will be compared with fits to experimental data. If the noise model is correct then the simulations and experiments should yield the same variances of parameter estimates. If the experimental results are worse, then the sensor model is not correct and there is model error.

In the discussion of the parameter fits to experimental data we will investigate four different phenomena that affect the indenter radius estimation errors: noise, the **E** matrix, the contact model, and the use of force sensor data.

3.5.1 Theoretical estimation errors

To generate theoretical estimation errors for the nonlinear fitting procedure Monte Carlo techniques were used. Gaussian white noise with a magnitude of 0.1 % strain was

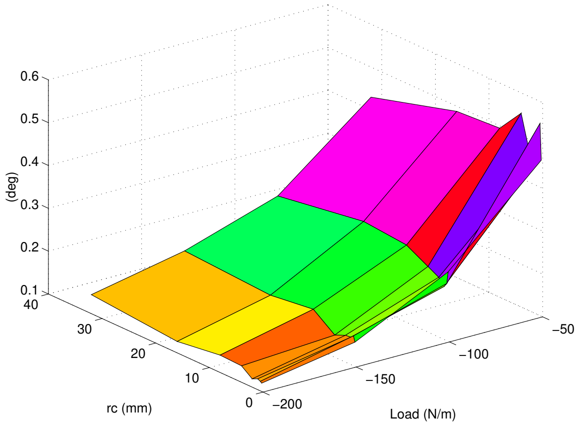


Figure 3.27: *Estimated standard deviation in the estimate of* θ_c.

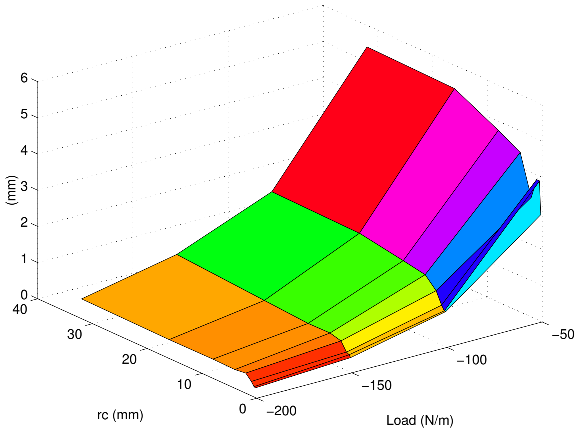


Figure 3.28: *Estimated standard deviation in the estimate of* R.

added to a data set of 100 pressure distributions at 4 different loads, and 9 different radii. The nonlinear fitting procedure was applied to this data set and the standard errors of the estimated parameters were computed. Figures 3.29, 3.27, and 3.28 show the estimated standard deviations in the fitted parameters at this noise level.

It is apparent that the estimates of contact location and $R = \frac{r_c r_b}{r_c + r_b}$, the composite radius, improve with load but are independent of curvature. Load estimates errors are smaller for small radius indenters, but in general are the same over both load and radius. The improved estimate of location and curvature with load makes sense since increasing the load increases the signal to noise ratio. According to this noise model, estimates of the

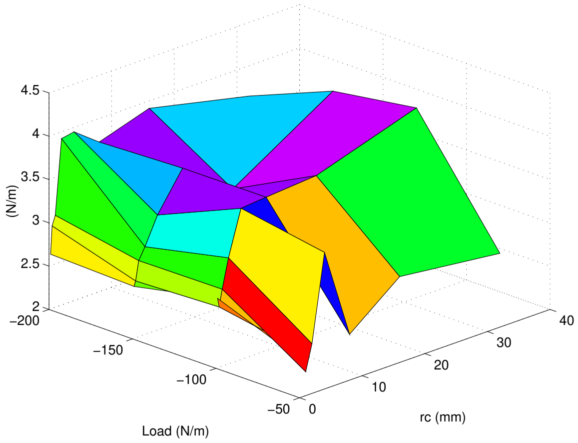


Figure 3.29: *Estimated standard deviation in the estimate of load.*

curvature can be very fine, with standard deviations of R in the tenths of a millimeter for a 200 N/m load. This is in accordance with Fearing's results (Fearing and Binford, 1991) using a linear elastic model, however it is much better than Ellis's results (Ellis and Qin, 1994) using a finite element model that included the copper strips of the sensing elements. The question is, what is the reliability of parameter estimates from experimental data?

3.5.2 Experimental curvature estimates

Using the same apparatus and techniques as in Section 3.4, indenters of width $r_c \in \{1.59, 3.90, 6.01, 7.55, 9.30, 12.30, 18.71, 25.25\}$mm were used in addition to the edge and flat contacts. Indenters were applied at 1^o degree intervals at a load of 200 N/m. Both the tactile and force data were sampled 20 times at each point. Again the sensor response and applied load were averaged over 20 readings at 33 Hz. We will first review the fitting method and its relation to the calibration techniques, then we will investigate bias in the estimate of r_c due to the choice of contact model. Next we will look at sources of variance in the parameter estimates and finally we will investigate the effect of adding total load information to the fitting procedure. For the sake of brevity all the results presented in this section were obtained with finger 0, the shielded sensor.

Method

Figure 3.30 illustrates the relation between the calibration and fitting procedures. In the following sections we will be exploring the effect of both different calibration techniques and contact models on the parameter estimation. Even when we use the empirical **E** we do not abandon the use of linear elasticity theory since the contact models are based on elasticity theory. Regardless of how **E** is obtained, we are still able to use linear elastic contact models to generate pressure distributions based on estimated model parameters.

Parameter fits were again accomplished with the nonlinear Simplex algorithm (Press et al., 1992) due to its simplicity. In particular one does not need to compute the gradient of the error term. Typical execution time was 0.1 second on a Sparc 20. The execution time was reduced to 0.04 second when an total load information from a force sensor was used. The 10 Hz rate is near that required for real time applications. With a more efficient optimization technique this time could undoubtedly be reduced.

Hertz contact models result in a biased estimate

As might have been predicted by the fitting results in Section 3.4.1, it was found that use of a Hertz contact model to predict subsurface strains resulted in a biased estimate of the indenter radius. Table 3.10 gives the mean estimate of r_c over 40 degrees for 9 different indenters. Concentrating for now on the results for the Hertz model, it was found that r_c was consistently underestimated when using both the empirical and model based calibration.

This is unfortunate since the other two contact models that have been discussed, the frictionless and adhesive model, both require the inversion of a large matrix to determine the pressure distribution for a given θ_w and r_c. Using the matrix inversion technique to determine the pressure distribution during a nonlinear fitting procedure would be much too slow. Instead it was decided to approximate the frictionless model by squeezing and stretching a Hertz model.

The Hertz model is based on the assumption of an elliptic pressure distribution, a frictionless contact, and an elastic halfspace whereas the "frictionless" model we have been using includes the effect of the rigid core. The pressure distributions generated by the frictionless model for constant radius indenters are approximately parabolic but yield a different contact area and peak pressure than that predicted by the Hertz model. Using

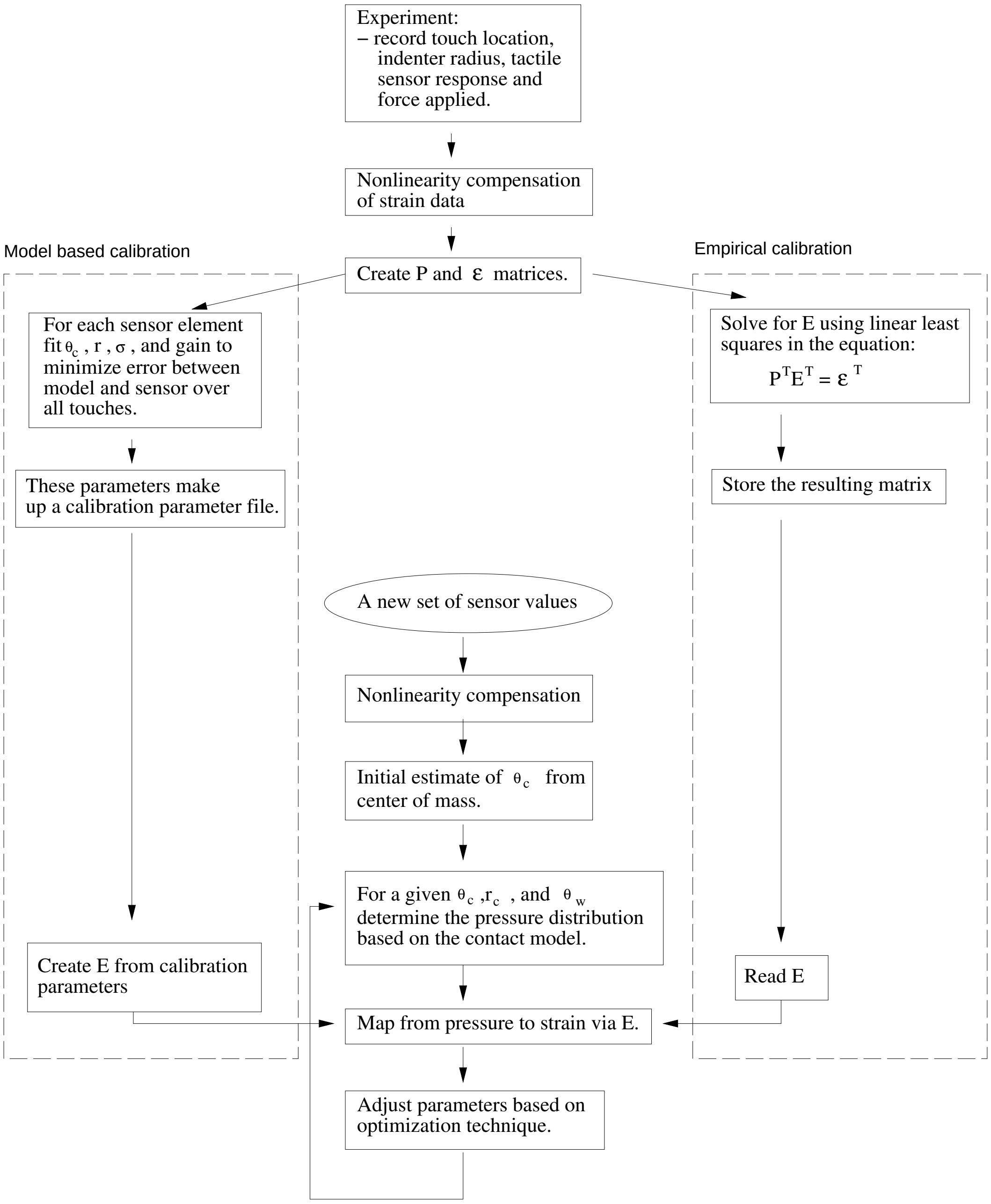


Figure 3.30: *Outline of the calibration and fitting procedures.*

Actual r_c	Hertz contact model		Approximate frictionless contact model	
	Model calibration	Empirical calibration	Model calibration	Empirical calibration
0.50	0.41	0.52	0.95	0.99
1.59	1.34	1.77	2.39	2.53
3.90	2.09	2.46	3.37	3.33
6.01	3.71	4.28	5.84	6.01
7.55	5.22	5.58	8.17	7.92
9.30	5.85	6.23	9.41	9.03
12.30	7.60	7.81	12.94	11.89
18.71	8.58	8.64	15.51	13.77
25.25	9.90	9.91	19.36	16.50

Table 3.10: *Mean r_c estimates using the Hertz and the approximate frictionless model.*

this observation one can approximate the frictionless pressure distribution by using a Hertz model with a different width and peak pressure. The first step is to determine a relationship between P, r_c, and θ_w for the frictionless model. Recalling the results of Section 2.4.4 where a smooth relationship between contact width, θ_w, and indenter radius, r_c, was shown, it is clear that for a fixed set of finger radius parameters, material constants, and P one may interpolate the relationship between θ_w and composite radius, R, so as to obtain an approximate θ_w for a given R. Alternatively for a fixed R one may interpolate to obtain θ_w from P. Thus if one starts with a set of data points of R, P, θ_w, one may determine an approximate contact width. Recalling the relation that if $\hat{P} = \zeta P$ then $\hat{R} = \frac{1}{\zeta}R$, then it is clear that the interpolation data set can be constant over R or P. In the following results 17 data points with a fixed R were scaled and interpolated to obtain θ_w. Given θ_w, one can solve for the Hertz pressure with that contact area and then scale the pressure distribution so it has the correct total load. Following this procedure one can arrive at an approximate pressure distribution for a frictionless contact quickly.

Returning to the results of Table 3.10, we see that this approximate frictionless model is much better at giving an unbiased estimate of r_c for indenters of radius 13 mm and less. Above that radius the estimates increase monotonically, however r_c is underestimated. This is likely due to frictional effects that would be accounted for in an adhesive contact

		θ_c (degrees)	θ_c (mm)	P (N / m)	R (mm)
Model based	Total	0.24	0.054	13.1	1.26
	Location	0.20	0.045	9.1	1.00
	Noise	0.14	0.031	9.6	0.79
Empirical based	Total	0.24	0.053	11.7	0.96
	Location	0.19	0.043	6.4	0.60
	Noise	0.14	0.032	9.8	0.77

Table 3.11: *Standard deviation of errors in parameters fitted for 40 touches at 1 degree intervals of 6 different indenter radii ranging from edge to flat contact.* Total: *standard deviation of parameters fitted to raw sensor data with 20 samples at each location.* Location: *standard deviation of parameters fitted to 20 times averaged data at each location.* Noise: *standard deviation of parameters fitted to raw data after subtraction of the mean estimate at each location.*

model. Unfortunately it is more difficult to create a quick approximate solution for the adhesive contact case. All the results in the following sections were obtained using the approximate frictionless contact model.

Sources of variance

If the only source of variance in the data were random, uncorrelated, and stationary noise and time response were not a consideration then estimation errors could be reduced by averaging the data over an adequate number of samples. Unfortunately we will see that there is another factor about equal in magnitude to sensor noise which is a location dependence of the parameter estimates. That is, the parameter estimate depends, in a consistent manner, on where the probe touches the sensor.

This effect became apparent when comparing summary statistics for averaged and raw data. Table 3.11 gives the standard deviations of the errors in the fitted contact parameters. The rows labeled "Total" are the standard deviations computed over 20 samples each of 40 touches using 6 different indenter radii. The rows labeled "Location" are the standard deviations of the errors in parameters fitted to 20 times averaged data. These rows indicate the variance due to location. The rows labeled "Noise" are the standard deviations of the errors in parameters fitted to raw data after subtraction of the estimate to the averaged data at each location. These rows indicate the variance due to noise. One would expect that after averaging the standard deviations in the parameter estimate would reduce as $\frac{1}{\sqrt{n}}$ where n is the length of the average, however, after 20 times averaging the standard deviation is only reduced by approximately 30 %. It is interesting to see that if we remove the location effect, the standard deviation of errors in parameters fitted to the raw data is reduced by about the same amount as averaging the data.

The good news is that contact location estimation error is very small. The total standard deviation in the contact location error for raw data is $\frac{1}{25}$th of the sensor spacing. If we include only the sensor noise than the standard deviation is halved. This value is the same as was predicted by theory. As might be expected from our frequency domain and SVD analysis of $\mathbf{E}$, total load estimates are noisy, with a standard deviation of 5 % of the applied load. The standard deviation in the load error is about 3 times that predicted by the theoretical estimation errors indicating some degree of model error. Figure 3.31 shows a plot of the location and load estimates for the averaged and raw data as a function of location. Fits using the empirical calibration are given since, for load and location estimation, both calibration methods yielded similar results. From this plot one can see that at a given location there is often a bias in the error that is greater than one standard deviation of the noise.

Table 3.11 gives the standard deviation of the radius estimate in terms of R. This is necessary when discussing flat contacts since one would expect the standard deviation of r_c in that case to be infinite. For small radius contacts the standard deviation in R and r_c will be approximately the same. For $r_c = r_b$ the standard deviation in r_c will be twice that for R. The standard deviation in R due to noise is 0.75 mm, meaning that for small r_c indenters, radii differences of 1.5 mm should be distinguishable at a 95 % level at a fixed location. Figure 3.32 shows estimates for r_c for averaged and raw data. Again we see that the standard deviation of the estimate at a particular location is often less than the bias due to location. It is also apparent that the standard deviation due to noise also depends on location. In other words, in addition to the systematic bias some locations provide more consistent estimates than other locations.

It is interesting to see that the location dependent bias in the estimate of r_c is consistent across indenters. Figure 3.33 shows r_c estimates for 5 different indenters as a function of location. For both calibrations the estimate of r_c is monotonic at each

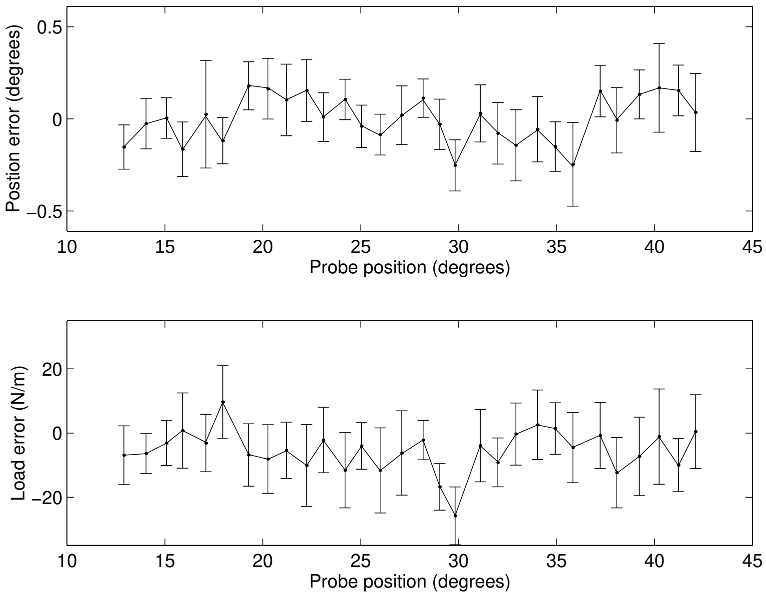


Figure 3.31: *Errors in estimates of θ_c (top) and P (bottom) for a 12.3 mm radius indenter using an empirical* $\mathbf{E}$. *The solid line is the parameter error for fits to 20 times averaged data and the error bars indicate the mean plus and minus one standard deviation for fits to the raw data. The applied load was 200 N/m.*

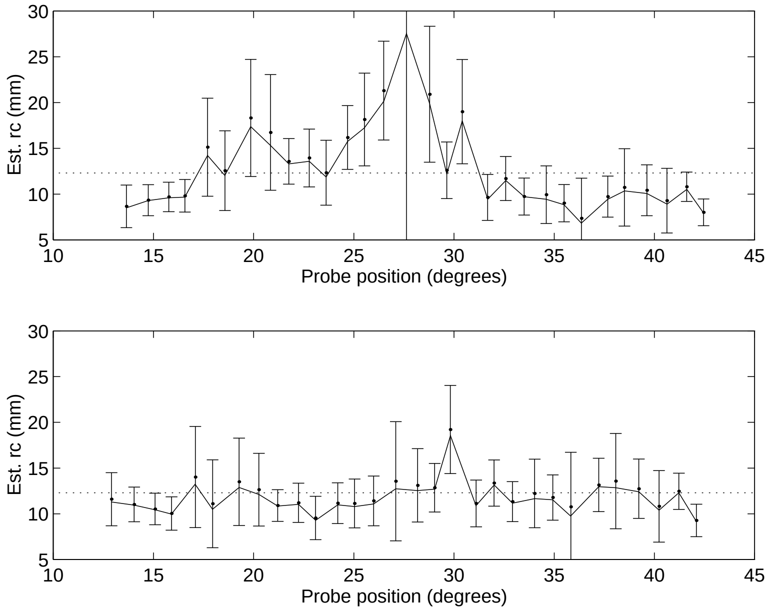


Figure 3.32: *Estimates of r_c for a 12.3 mm radius indenter. The top plot uses a model based* $\mathbf{E}$ *while the bottom plot uses an empirical* $\mathbf{E}$*. In both cases the dotted line is the actual r_c, the solid line is the fitted r_c to 20 times averaged data, and the error bars indicate the mean plus and minus one standard deviation for fits to the raw data.*

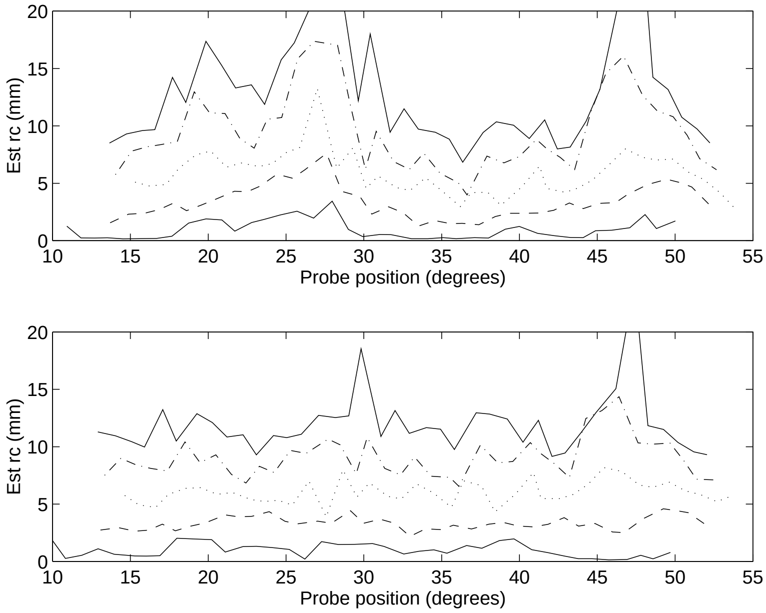


Figure 3.33: *Estimate of r_c using model-based (top) and empirical (bottom) calibration on 20 times averaged data. From lower to upper trace* $r_c = \{0.5, 3.90, 6.01, 9.30, 12.30\}$mm.

location, although the bias might be considerable. This indicates that the information about curvature is contained in the data, however the model used to estimate curvature is not quite correct.

The location dependence in the bias of the estimate of R is reduced by using empirical calibration over model based calibration. This is shown both in Table 3.11 and Figures 3.32 and 3.33. This is to be expected since the empirically derived $\mathbf{E}$ can compensate for some of the anomalies in sensor construction.

Adding load information

It has been suggested that by using an additional source of load information the estimate of r_c might be improved. To test this hypothesis fits to the averaged data were augmented with the total load, thus constraining θ_w to lie on a line in θ_w and R space. Table 3.12 summarizes results of fits with and without load information. The tabulated results are inconclusive since in some cases the fit with load is better and in other cases the fit without load is better. The fits without load appear to be better at producing an unbiased estimate of curvature over the middle range of indenter radii. With load information the

Actual r_c	Mean		Standard deviation	
	No load	With load	No load	With load
0.50	0.99	0.36	0.60	0.14
1.59	2.53	2.29	0.77	1.04
3.90	3.33	6.17	0.58	1.52
6.01	6.01	6.69	0.98	1.49
7.55	7.92	6.30	1.32	1.18
9.30	9.03	9.09	1.73	1.64
12.30	11.89	10.43	2.64	1.78
18.71	13.77	15.95	2.23	3.45
25.25	16.50	19.24	3.60	5.26

Table 3.12: *Fitting errors for r_c with and without load information.*

extreme indenter radii were better predicted.

Since summary statistics can be misleading, a comparison of 4 different fitting methods for $r_c = 3.9$ mm is shown in Figure 3.34. Fits with load information using either calibration method result in a periodicity in the r_c estimate at the same spatial frequency as the sensor spacing. This is most likely due to inhomogeneities brought on by the inclusion of copper in the rubber medium.

This result, that adding load information does not improve the parameter estimate is not intuitive so additional simulations were carried out to see if the cause could be determined. It was found that the variance in the parameter estimate was larger when the additional load information was noisy, however when the load information was not noisy the variance was slightly less than when no load information was given. It is apparent then that the load sensor we are using is too noisy and that the tactile sensor data is more reliable without it. By using load information the parameters θ_w and R are constrained to a line in θ_w and R space. A noisy P value will cause the parameters to be constrained to the wrong line, however the parameter search space is reduced. If a better load sensor were used than the variance in the parameter estimate as well as the time to fit the parameters could be reduced.

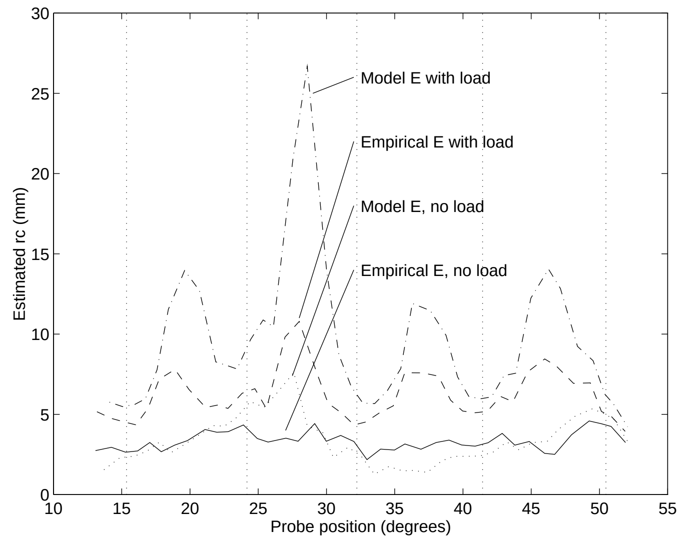


Figure 3.34: *Estimates of r_c for a 3.9 mm radius indenter for empirical (solid, dashed) and model based (dash dot, dotted) fits with load information (dashed, dash dot) and without load information (solid, dotted). The vertical dashed lines indicate sensor locations.*

3.6 Summary

This chapter has described the construction, characterization, and shape sensing capabilities of a tactile sensor. The characterization results indicate that temperature, creep, relaxation, and proximity effects are bigger problems than the random electrical noise generated by the electronics. Compensation for some of these effects was described and allowed a repeatability of 0.2 % output at a constant temperature.

In the calibration section it was shown how the model parameters, r_s, σ, θ_c, and a gain value could be fitted to each sensor element. Some of the sensors allowed a fitting error as small as 1 % of the peak strain for a 200 N/m edge contact. This value is near the standard deviation predicted by the noise. While some elements fit well, others did not. Fitting errors were largely seen in the tails of the impulse response. For this reason a so-called "empirical" calibration was also undertaken. This calibration method identified the **E** map through a linear least squares fit to the data. The resulting **E** compared qualitatively to the **E** predicted by theory in its resemblance to a spatial band pass filter. Fitting errors using this calibration technique were consistently smaller than the fits for the model based calibration. In particular the tails of the impulse response were better predicted. One may conclude that one can construct a sensor that will result in an impulse response like that predicted by a linear elastic model. Put another way, we see *no* problem with the use of linear elastic models in theoretical estimates of tactile sensor capabilities. However, when it comes to using an actual sensor it is apparent that an empirical technique can provide a more consistent prediction of sensor responses.

We have also seen that use of the standard Hertz contact model will lead to biased estimates of indenter radius This was seen both in the calibration section and the parameter identification section. It was also shown that a modified Hertz model could be used to approximate the rigid core frictionless contact model to produce unbiased estimates.

The parameter fitting results showed that contact location can be estimated to a precision of $\frac{1}{25}$ th of the sensor spacing. Such precision in the estimate of contact location would not be possible with a surface displacement sensor with the same sensor density due to the high degree of aliasing. We also saw that the standard deviation of the total load was 10 N/m and that the load estimate suffered from a location dependence making it hard to achieve the 3 N/m predicted by theory. The spatial frequency response for the sensor indicates that the lowest frequencies are hard to estimate, so this result is not surprising.

Estimates of R and r_c were seen to be the hardest to obtain and the most sensitive to the calibration technique. Results using the model based calibration indicate that curvature estimation at a fixed location could be reliable and monotonic, however each location showed a bias. We saw that this bias often appeared to be periodic at the same spatial frequency as that of the sensors indicating that the inclusion of the rigid copper in the elastic material led to model error. This is not surprising. The empirical calibration is able to compensate for this effect to a large degree and allows considerably less position dependent bias in the curvature estimates. Using 20 times averaging and the empirical calibration R could be estimated with a standard deviation of 0.64 mm. The addition of load information to the fitting process did not improve the estimate, however it did reduce the execution time due to the reduction in the parameter search space.

Overall this chapter has shown that estimation of curvature can be accomplished with a subsurface strain sensing tactile sensor. There is room for improvement in the estimation of the $\mathbf{E}$ map. Linear least squares is perhaps not the best way to achieve this, but investigation of other more probabilistic approaches are beyond the scope of this dissertation. It would be interesting to see if a general nonlinear map approximator, such as a neural net, could compensate for the inhomogenous, anisotropic, and nonlinear qualities of an actual sensor. Alternatively there is a lot of room for improvement in the construction methods which might allow more uniform and linear sensor responses. We have shown that is is possible to construct a sensor element that is well predicted by a linear elastic model, however it is difficult to accomplish over the entire array.

Chapter 4

Grasping experiments with tactile sensors

We have seen how a tactile sensor may be constructed and modeled so as to provide contact shape and location information. The next question is, can this information be used to improve manipulative capabilities of dextrous hands. This chapter explores this issue through the use of a two fingered hand equipped with tactile sensors. Previous work using tactile sensors in a dextrous hand has focused on improved translation and rotation of the grasped object (Maekawa et al., 1992a). In the following experiments the redundant degrees of freedom in a 6 degree of freedom planar manipulator are used with the contact location and curvature estimates provided by the tactile sensor to improve grasp quality through in-hand manipulation.

We proceed by describing the kinematics of the manipulator and its six degrees of freedom in the plane. Next the frictional properties of the rubber fingers are determined through experiments. Using grasp stability analysis and tactile feedback, optimal grasps of a disk and a rectangle are obtained through regrasping as well as manipulation.

4.1 Hand control

A two fingered hand was constructed from two modules of the RobotWorld (Scheinman, 1987) system. Each module has three degrees of freedom in the plane, two prismatic and one revolute, in addition to a vertical prismatic joint. By attaching a link as shown in Figure 4.1 to the revolute joint of each module a two fingered hand is constructed. We

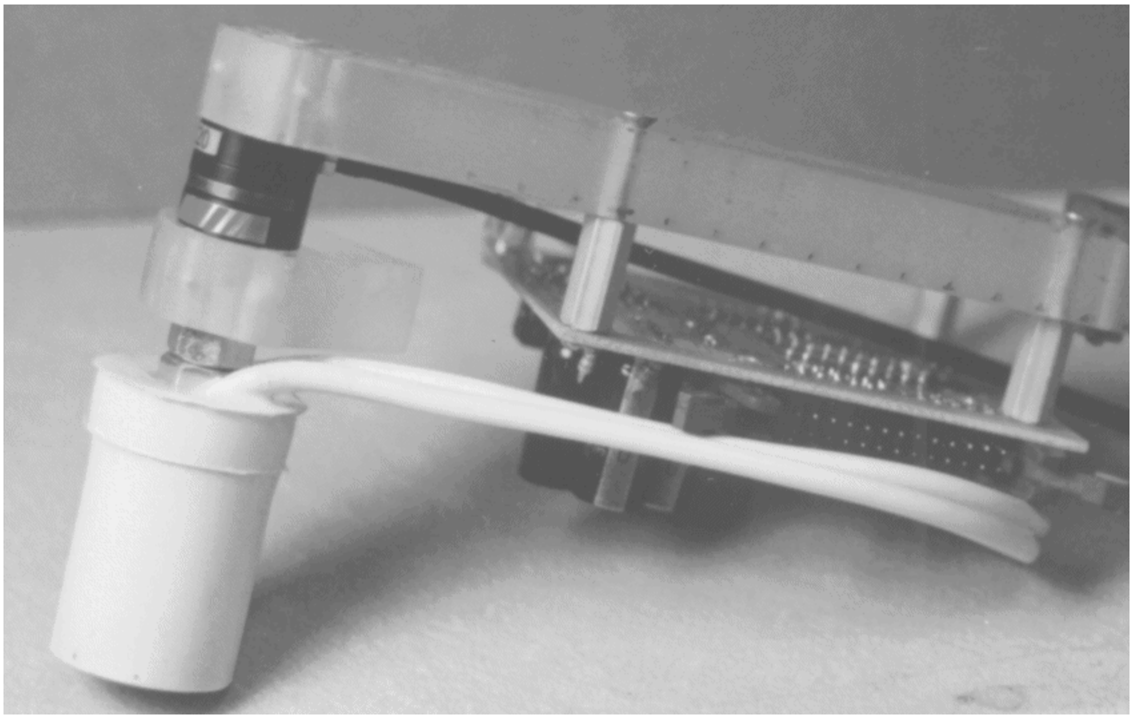


Figure 4.1: *One finger link with a tactile sensor.*

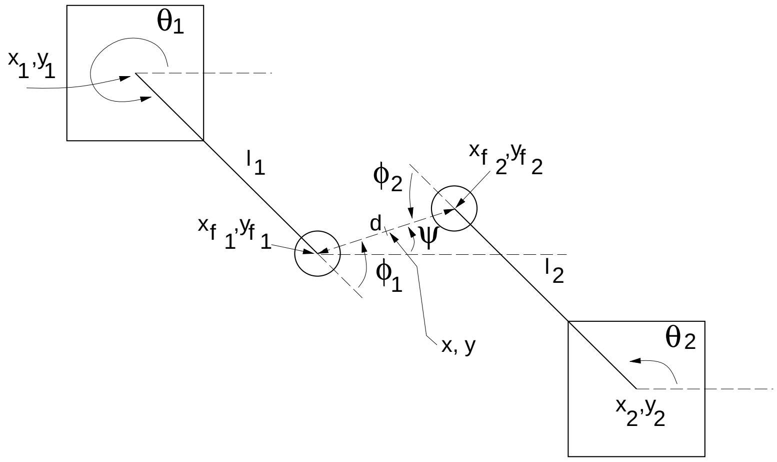


Figure 4.2: *Two fingered hand using two RobotWorld modules.*

first explore a convenient set of configuration variables and controllers for free space motion then we go on to explore kinematics of motion in contact.

4.1.1 Free space kinematics

Figure 4.2 shows the configuration of the two modules as they are used for the hand. When controlled together the two modules allow six degrees of freedom in the plane. Since an object in the plane has only three degrees of freedom and one degree of freedom is required to apply grasp force, the extra two degrees of freedom can be used to manipulate the object held between the fingers. This extra freedom allows so-called "in-hand" manipulation.

The configuration of the degrees of freedom of each module in the plane can be given by x_i, y_i, and θ_i where $i \in 1, 2$. It is more convenient to discuss motion of the center point between the fingers, x, y, the distance between the centers of the fingers, d, and the orientation of the line between the finger centers, ψ. In addition the angle of the intersection between the line between the finger centers and the surface of each finger is given by ϕ_i. The map from the module configuration, or joint space, to the configuration space just discussed is given by the *forward* kinematics

$$\begin{aligned}
x_{f_i} &= x_i + l_i \cos\theta_i \\
y_{f_i} &= y_i + l_i \sin\theta_i \\
d^2 &= (x_{f_1} - x_{f_2})^2 + (y_{f_1} - y_{f_2})^2
\end{aligned}$$

$$
\begin{aligned}
\psi &= \operatorname{atan2}(y_{f_2} - y_{f_1}, x_{f_2} - x_{f_1}) \\
x &= \frac{1}{2}(x_{f_1} + x_{f_2}) \\
y &= \frac{1}{2}(y_{f_1} + y_{f_2}) \\
\phi_1 &= \psi - \theta_1 \\
\phi_2 &= \psi - \theta_2 + \pi
\end{aligned}
$$

and the map from the configuration space to the joint space is given by the *inverse* kinematics

$$
\begin{aligned}
x_{f_1} &= x - \frac{d}{2}\cos\psi \\
y_{f_1} &= y - \frac{d}{2}\sin\psi \\
x_{f_2} &= x + \frac{d}{2}\cos\psi \\
y_{f_2} &= y + \frac{d}{2}\sin\psi \\
\theta_1 &= \psi - \phi_1 \\
\theta_2 &= \psi - \phi_2 + \pi \\
x_i &= x_{f_i} - l_i\cos\theta_i \\
y_i &= y_{f_i} - l_i\sin\theta_i.
\end{aligned}
$$

In addition we can map from desired configuration space velocities to joint velocities using

$$
\begin{aligned}
\dot{x}_{f_1} &= \dot{x} - \frac{1}{2}\dot{d}\cos\psi + \frac{1}{2}d(\sin\psi)\dot{\psi} \\
\dot{y}_{f_1} &= \dot{y} - \frac{1}{2}\dot{d}\sin\psi - \frac{1}{2}d(\cos\psi)\dot{\psi} \\
\dot{x}_{f_2} &= \dot{x} + \frac{1}{2}\dot{d}\cos\psi - \frac{1}{2}d(\sin\psi)\dot{\psi} \\
\dot{y}_{f_2} &= \dot{y} + \frac{1}{2}\dot{d}\sin\psi + \frac{1}{2}d(\cos\psi)\dot{\psi} \\
\dot{\theta}_i &= \dot{\psi} - \dot{\phi}_i \\
\dot{x}_i &= \dot{x}_{f_i} + l_i(\sin\theta_i)\dot{\theta}_i \\
\dot{y}_i &= \dot{y}_{f_i} - l_i(\cos\theta_i)\dot{\theta}_i.
\end{aligned}
$$

A simple controller can be used to regulate the hand to a desired set of configuration space variables. Since our focus is on quasistatic grasping, the current controller does not take the dynamics of the manipulator into consideration. Instead a simple diagonal

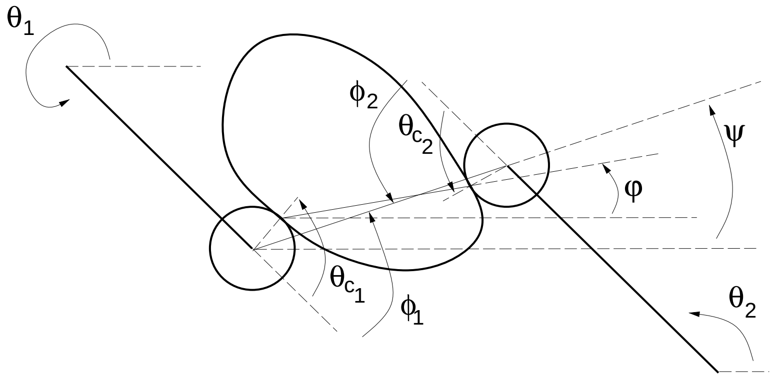


Figure 4.3: *Fingers in contact with an object.*

proportional control is used in configuration space to drive the error between desired and actual position to zero. Using the joint velocity map a velocity trajectory in configuration space can also be followed.

4.1.2 Kinematics in contact

When the fingers come in contact with an object, contact location variables, θ_{c_i}, and contact forces $\mathbf{f}_i$ are introduced. Since we are dealing with quasistatic grasping the forces and torques on the object must sum to zero. Therefore $\mathbf{f}_1 = -\mathbf{f}_2$ and the force direction must lie along the line between the contacts. If φ is the orientation of the line between the contact points, then the force vectors $\mathbf{f}_i$, which are collinear with this line, also have direction φ and $-\varphi$. Figure 4.3 shows the configuration of the hand in contact with an object.

From this Figure 4.3 we can also see the relation between the free space configuration variables ϕ_i and ψ and the contact configuration variables θ_{c_i} and φ . If the coefficient of friction between the object and the fingers, μ_f, were equal to zero then the only possible stable grasp points on a smooth object would be antipodal. That is, they would have equal and opposite normals. For contact with a frictionless smooth object the surface normals of the fingers and the object must be equal and opposite at the contact locations. Since the fingers are round, the direction of the surface normal is the same as the vector from the finger center to the contact location. Hence, for a given joint configuration the only contact

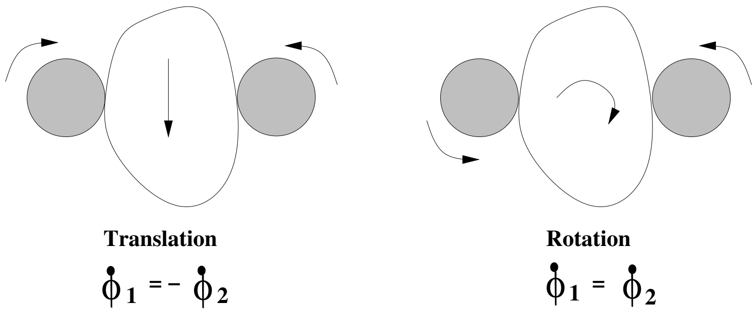


Figure 4.4: *Interior motions of the two fingered hand. Pure translation or rotation only occur if the object curvature at both contact points is the same.*

points that would allow an antipodal grasp would be given by ϕ_i. In other words, the free space configuration variables give stable frictionless grasp configurations of the hand. Yet another way of viewing this is that if we were given a hand configuration and went to grasp a frictionless smooth object by varying only the configuration space variable d, the object would either slip out of the fingers or it would be grasped with contact locations ϕ_i.

We use this fact to justify the use of the free space configuration variables for forming a grasp of an object. Before picking up an object, the hand will be used in the free space coordinates to position the fingers around the object. The object will then be grasped using a damper control, to be discussed shortly, on the configuration variable d. Due to μ_f not being zero the actual contact locations will be different from ϕ_i. One goal of a grasp stabilization routine is to drive the contact locations to ϕ_i so as to be maximally distant from the edges of the friction cone.

Assuming the object can be held stably, it can be translated and rotated *without* changing the contact locations through control of x, y, and ψ. Manipulation of the object within the grasp can be achieved with the remaining two degrees of freedom, ϕ_i. Figure 4.4 shows how the relative velocities of ϕ_i lead to rotation and translation within the grasp. Later in this chapter we will show how these internal motions can be used to improve grasp quality without letting go of the object.

Axis	Resolution	Stiffness
x, y	0.030 mm	17 N/mm
θ	0.006 degree	4.8 N mm/degree
z	0.005 mm	6.9 N/mm

Table 4.1: *Mechanical properties of one manipulator module.*

4.1.3 Internal force control

The previous section assumed the object could be held stably during manipulation. In our quasistatic experiments a simple control on the distance between the finger centers, d, will suffice to maintain stability. We can exploit the low stiffness in the θ_i joints, shown in Table 4.1, to control d to generate an internal force when ϕ_i and θ_{c_i} are near $\pi/2$. The stiffness in the θ_i axes corresponds to a translation stiffness of 0.014 N/mm at the center of the finger.

When in free space $\dot{d}$ will be controlled according to a stiffness controller

$$\dot{d} = K_d \left(d_d - d_a - \frac{f_d - f_a}{K_s} \right)$$

and when in contact with the object the damper law

$$\dot{d} = -K_v(f_d - f_a)$$

is used. Here d_d is the desired d, d_a is the actual d, f_d is the desired internal force, and f_a is the actual internal force. K_s is the desired virtual spring constant in the stiffness mode and K_v is the inverse damper constant in $\frac{\mathrm{m}}{\mathrm{sN}}$. K_d is the proportional gain in $\frac{1}{\mathrm{s}}$. The controller update rate of 20 Hz limits the range over which these parameters may be varied and still maintain controller stability.

4.2 Friction experiments

One of the most important parameters, and often the most variable, in grasp stability analysis is the coefficient of friction between the finger and the object, μ_f. In addition to the types of the contacting materials, the condition of the surfaces in terms of roughness and contamination and the normal loading can lead to large variations in μ_f. For dry contacts it is generally accepted that the coefficient of friction is proportional to the true area of contact (Bowden and Tabor, 1950). For hard materials the true area of contact varies linearly with normal load, however, for elastic contacts, Hertzian analysis reveals that the real contact area is proportional to the 2/3 power of the normal load (Schallamach, 1952). It has also been shown that compliant materials with a large μ_f seemingly suitable for grasping can have their friction coefficient reduced by a factor of 10 with the introduction of water contamination (Cutkosky et al., 1987). It is then not clear whether a large μ_f or a consistent μ_f is more important for grasping. It can be argued that a consistent μ_f, obtained perhaps even by boundary lubrication to reduce slip-stick effects (Dupont and Dunlap, 1993), is more amenable to grasping tasks since sliding can be more reliably anticipated and, if necessary, controlled. Alternatively one could actively estimate the onset of sliding, and hence the friction coefficient, by use of slip sensors (Tremblay and Cutkosky, 1993).

For this dissertation we will consider manipulation of dry wax and aluminum objects with rubber fingers made from Dow Corning HSII silicone rubber. Of particular interest to our grasp analysis is the effect of local object curvature on μ_f. Specifically, for the same normal load and friction interface conditions, does an edge contact require a larger tangential force to slide than a round or flat contact? Intuitively one might think that an edge would be less likely to slip than would a flat contact due to its superior "gripping" capabilities. Comparisons of friction coefficients for round and chisel indenters support this intuition (Schallamach, 1969). To investigate this question the apparatus shown in Figure 4.5 was constructed to measure the friction coefficients for different radii indenters. A flat sample of 3 mm thick rubber was dragged at constant velocity over an indenter attached to a force and torque sensor. Weights were placed above the rubber slab to apply the normal force. The friction coefficient during the sliding motion was computed from the normal and tangential force information.

There was no significant variation in μ_f over loadings from 50 grams to 500 grams, however there was a significant relation between the radius of the indenter and μ_f as can be seen in Table 4.2. Since the rubber sample was flat, the indenter radius is equal to R and not r_c. In Table 4.2 the equivalent r_c for contact between a finger of radius $r_b = 12.7$ mm and the indenter is also given. For a conforming contact, $R = \infty$, μ_f reaches a maximum, however it is often not well defined due to the large amount of sticking that occurs during sliding. Often this leads to an oscillatory sliding. For $r_c = \infty$, a flat contact on a 12.7 mm

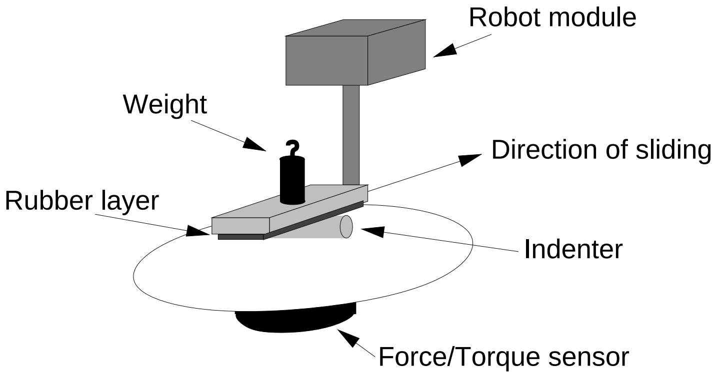


Figure 4.5: *Friction measurement apparatus.*

R (mm)	r_c (mm)	μ_f
∞	-12.7	3.4
12.7	∞	2.0
2.5	3.1	2.0
0.3	0.3	1.4

Table 4.2: *Dependence of friction coefficient, μ_f, on indenter radius, r_c. The coefficients were measured for a 54 mm long contact between flat Dow Corning HSII silicone rubber and a rounded piece of aluminum. They did not vary significantly over loadings from 50 g to 500 g.*

radius cylinder, μ_f is still quite large, however for an edge contact, $R = r_c = 0.3$ mm, μ_f is reduced to 1.4. These results contradict the intuition that an edge contact would have a larger μ_f and instead indicate that contacts which generate a larger contact area result in a larger μ_f. This is consistent with the notion that μ_f is proportional to contact area. This is not consistent with (Schallamach, 1969), however that study used much larger loads in the range of 1 kg to 10 kg. For larger loads one can expect non-linear deformations that will result in a "piling-up" of the rubber material. For the purposes of the grasp stability analysis in this chapter, we can assume that $\mu_f >= 1.4$ hence we can allow angles between surface normals and contact normals as large as 55 degrees and assume no sliding will occur.

4.3 Finding a good grasp

Given the kinematics and frictional properties of the hand it is now possible to investigate the quality of a grasp of an object. Since the main investigation of this chapter is the integration of tactile sensing into grasping, the focus will be on how to use tactile feedback to improve a grasp. The simplest approach is to close the fingers on an object located between the fingers, analyze the tactile data, and then regrasp the object at a better set of contact locations. This assumes, of course, that a method for determining grasp quality is available.

Many grasp quality measures have been defined in the literature. They may be divided into force-closure approaches and local contact approaches. In the force-closure approach the grasp quality may be defined as the ratio of the magnitude of the applied object wrench to the magnitude of the finger forces (Li and Sastry, 1988; Murray et al., 1994; Mirtich and Canny, 1994; Ferrari and Canny, 1992). A grasp is better if it can resist the same object wrench using smaller finger forces. An alternative approach is to define the quality as the minimum of the dot products between the surface normals at the contacts and the contact forces (Jameson and Leifer, 1987; Kao, 1990). A grasp is better in this case if the finger closest to sliding has a smaller angle between the contact force and surface normal. The first technique can be thought of as a global grasp measure, usually used with an object model, the second as a local measure, usually used without an object model. Both of these measures are complicated by the need to determine the magnitude of vectors which include both forces and moments. Since forces and moments have different units, some sort of scaling is required. The choice of the scaling factor can lead to different "optimal" grasps. Through the use of friction limit surfaces (Goyal et al., 1991) proper scaling between forces and moments for soft finger contacts can be accomplished, removing the ambiguity of "optimal" grasps.

In this chapter we will be focusing on regrasping of convex planar objects. With the convexity restriction we can use results from both global and local grasp quality analysis to develop a regrasping strategy. Since we are restricted to planar grasping with a two fingered hand the task of determining a regrasping strategy is greatly simplified. First it has been shown that every planar convex object has at least two possible contact pairs (Jameson, 1985) for a frictionless two-fingered grasp. The grasp pairs correspond to locations of the double normals on the object. Of these contact pairs there are only two different directions of the contact line between the grasp points. For example a rectangle can be grasped anywhere along opposite points on its sides, however there are only two different lines that are parallel to lines between the contact pairs. Furthermore, it can be shown that for grasps with frictional contacts any other grasp, or so-called "non-parallel" grasp, can not be optimal in the force-closure sense (Mirtich and Canny, 1994). This leaves us with two possible "optimal" grasps, one at the maximum diameter of the object, the other at the minimum diameter of the object. Mirtich argues that the maximal diameter is optimal since it will be able to resist larger moments than the minimal diameter grasp for the same magnitude of contact forces. From a practical standpoint, however, the minimal diameter is better since it will be stable to small perturbations in contact location.

This last point is important and bears directly on whether Mirtich's optimal grasp can be obtained by an iterative regrasping strategy. Consider the problem of iteratively regrasping an ellipse until the maximal diameter is found. At each step in the iteration the contact locations, and hence surface normals, are sensed. Assume for now that there is a way of using this information to determine a new set of contact points that will result in a larger diameter grasp. If the contacts are frictionless then we know that when the fingers are closed on the object it will slide until it either is no longer grasped or an antipodal set of contacts is found. As is shown in Figure 4.6, the first situation occurs when the contact points lie on the same half of the ellipse, the second when the contacts are on opposite halves. For a frictionless grasp, then, we can only grasp at the minimal diameter, or not at all, unless we are extremely lucky and place the fingers exactly at the maximal diameter. We can determine this case using curvature estimates from the tactile sensors, since, as Jameson points out, the radius of curvature at the contact locations for the maximal diameter *must*

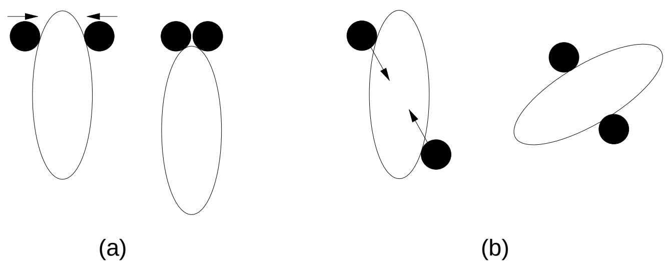


Figure 4.6: *Two possible initial grasps of an ellipse and the resulting stable states for frictionless contacts.*

be less than the distance between the contact points ((Jameson, 1985), page110). Since some object geometries do not admit iterative regrasping techniques to achieve the maximal diameter grasp, we will instead search for the closest parallel grasp from the initial grasp.

4.3.1 Friction angle

For two fingered quasistatic grasping with a planar two-fingered hand it is not necessary to use force sensing to determine the angle between the surface normal and the applied force at each contact. We only need to compute the line between the contact points from the tactile sensor data to determine the force direction since for quasistatics the force must be along the line between the contact points. Since the fingers are round, the finger surface normals $\hat{\mathbf{n}}_i$ are found directly from the contact locations.

$$\hat{\mathbf{n}}_i = \begin{bmatrix} \cos(\theta_i + \theta_{c_i}) \\ \sin(\theta_i + \theta_{c_i}) \end{bmatrix} \tag{4.1}$$

Recall that the finger centers were given by (x_{f_i}, y_{f_i}) so the locations of the contacts, (x_{c_i}, y_{c_i}), in inertial coordinates are

$$\begin{bmatrix} x_{c_i} \\ y_{c_i} \end{bmatrix} = \begin{bmatrix} x_{f_i} \\ y_{f_i} \end{bmatrix} + r_b \hat{\mathbf{n}}_i \tag{4.2}$$

Defining the line to contact i from contact j by $\mathbf{c}_{ij}$ and the unit direction of this line by $\hat{\mathbf{c}}_{ij}$ then the angle between the normal and force and each contact is

$$\xi_i = \sin^{-1}(\hat{\mathbf{n}}_i \times \hat{\mathbf{c}}_{ij}) \tag{4.3}$$

For no sliding to occur we must have $|\xi_i| < \tan^{-1} \mu_f$. Optimal grasping based on local measures of contact stability would minimize the largest of the ξ_i.

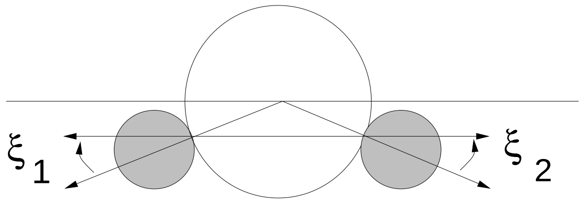


Figure 4.7: *Friction angles when grasping a disk. Rotation of the fingers about the center of the disk by the angle between the surface normal and the chord between the contacts results in a grasp along the diameter.*

4.3.2 Grasping a disk

For this section we constrain discussion to grasping of the simplest planar convex object, a disk, and formulate a two step procedure for obtaining the optimal grasp using tactile feedback through regrasping. In addition we demonstrate manipulation of the disk to obtain the optimal grasp without letting go of it.

An optimal grasp of a disk is simply one that is along a diameter. We assume that, due to friction, the disk can first be grasped along a chord. From the tactile feedback we can determine the angle between the surface normal and the chord at each contact point in addition to the center of the disk. By rotating each finger around the disk by an angle equal to that between the chord and normal the optimal grasp is obtained. Figure 4.7 shows the friction angles when grasping a disk.

Creating a regrasping algorithm amounts to determining a new set of configuration variables, $\tilde{x}$, $\tilde{y}$, $\tilde{\psi}$, $\tilde{d}$, and $\tilde{\phi}_i$ from the current configuration and tactile feedback. The easiest way to do this is to set $\tilde{\phi}_i = \frac{\pi}{2}$, and determine r_c, the radius of the disk, from position information or local curvature information, so that the new center of grasp is

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} = \begin{bmatrix} x_{c_1} \\ y_{c_1} \end{bmatrix} + r_c \hat{\mathbf{n}}_1 \tag{4.4}$$

and the contact positions are

$$\begin{bmatrix} \tilde{x}_{c_i} \\ \tilde{y}_{c_i} \end{bmatrix} = \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - r_c \begin{bmatrix} \cos(\theta_i + \theta_{c_i} + \xi_i) \\ \sin(\theta_i + \theta_{c_i} + \xi_i) \end{bmatrix} \tag{4.5}$$

The last two configuration variables are then

$$\tilde{\psi} = \arctan(\tilde{y}_{c_2} - \tilde{y}_{c_1}, \tilde{x}_{c_2} - \tilde{x}_{c_1}) \tag{4.6}$$

$$\tilde{d} = 2r_c \tag{4.7}$$

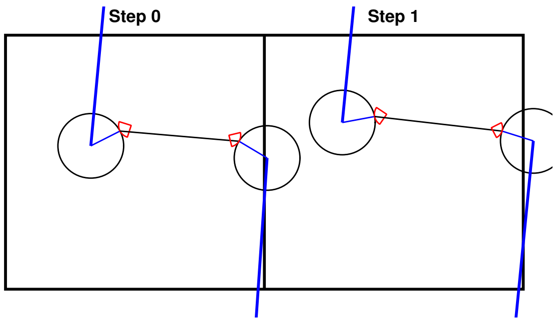


Figure 4.8: *Regrasping of a disk using tactile position and curvature feedback. The left image shows the tactile position and curvature feedback from the initial grasp. The right figure shows the grasp configuration after regrasping. Local r_c estimates were used to determine the disk radius.*

Figure 4.8 shows regrasping of a disk with the manipulator discussed in the previous sections. In this case r_c was determined from the curvature feedback from the tactile sensor. As can be see in the figure the r_c estimates obtained in this manner are biased below the actual value so the optimal grasp is not obtained in one step. For comparison, regrasping using r_c estimates based on position feedback is shown in Figure 4.9. In this case r_c is computed from

$$r_{c_i} = \frac{1}{2} \left\| \begin{bmatrix} x_{c_2} - x_{c_1} \\ y_{c_2} - y_{c_1} \end{bmatrix} \right\|_2 \cos(\xi_i). \tag{4.8}$$

This leads to a much better estimate of r_c so the optimal grasp is achieved in one regrasp step.

Using one of the two interior motions discussed in Section 4.1.2 the disk can also be translated with respect to the fingers to achieve an optimal grasp without letting go of it. The simplest way to do this is to set $\mathrm{sgn}(\dot{\phi}_\mathrm{i}) = -\mathrm{sgn}(\dot{\xi}_\mathrm{i})$ with $|\dot{\phi}_i| =$ constant until $\xi_i = 0$. This manipulation is shown in Figure 4.10. Both the friction angles and the distance between the contact points indicate that a parallel grasp is reached at the end of the manipulation. This technique of improving grasps could be used continuously during a task to maintain an optimal grasp in the presence of disturbance forces.

4.3.3 Grasping a rectangle

The second simplest object to grasp in the plane is a rectangle. In this case we again analyze the angle between the surface normal at the contacts and the line between

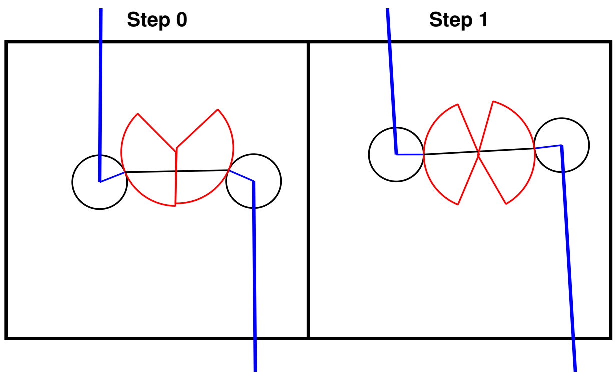


Figure 4.9: *Regrasping of a disk using tactile position feedback. The left image shows the tactile position and estimated radius from the initial grasp. The right figure shows the grasp configuration after regrasping. r_c estimates were based on the assumption that the grasped object was a disk.*

the contacts, but this time the fingers rotate, or more precisely, translate, in an angular direction opposite to that for the disk. Figure 4.11 illustrates the friction angles when grasping a rectangle.

Following the same steps as with the disk we can compute the new configuration variables for regrasping from the current configuration and the tactile feedback using

$$\begin{aligned}
\begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} &= \frac{1}{2}\begin{bmatrix} x_{c_2} + x_{c_1} \\ y_{c_2} + y_{c_1} \end{bmatrix} \\
\tilde{\psi} &= \theta_1 + \theta_{c_1} = -(\theta_2 + \theta_{c_2}) \\
\tilde{d} &= d\cos(\xi_1) = d\cos(\xi_2) \\
\tilde{\phi}_i &= \frac{\pi}{2}
\end{aligned}$$

Again we demonstrate the regrasping procedure in Figure 4.12.

We can also manipulate the rectangle within the grasp to obtain the optimal grasp by using the interior rotation motion discussed in Section 4.1.2. In this case we set $\mathrm{sgn}(\dot{\phi}_\mathrm{i}) = \mathrm{sgn}(\dot{\xi}_\mathrm{i})$ with $|\dot{\phi}_i| =$ constant until $\xi_i = 0$. Figure 4.13 illustrates the trajectory of the rectangle during manipulation to obtain the parallel grasp. In this case the distance between the contact points and the friction angle are shown as the the grasp through the parallel grasp. As would be expected the distance between the contact points reaches a minimum when the friction angles go to zero.

The regrasping and manipulation techniques discussed for the rectangle and disk

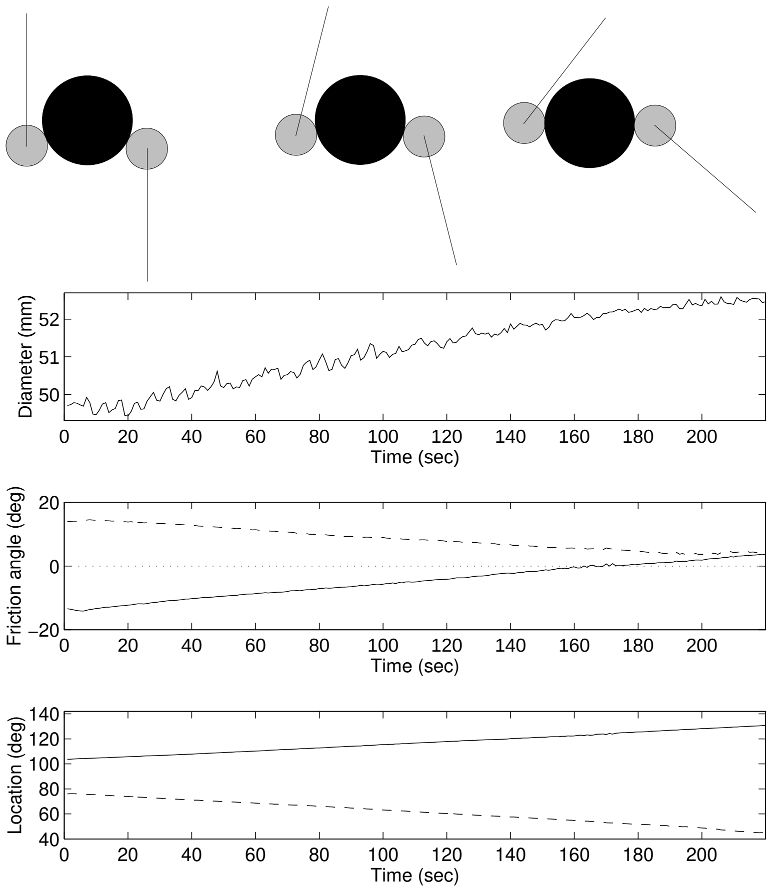


Figure 4.10: *Trajectory of the disk during translation to the optimal grasp.* $\dot{\phi}_1 = -\dot{\phi}_2 = 0.2$ deg/sec. *The top plot shows the distance between the contact points, the middle plot shows the friction angles,* ξ_1 *and* ξ_2*, and the bottom plot shows the contact locations from tactile feedback,* θ_{c_1} *and* θ_{c_2}*.*

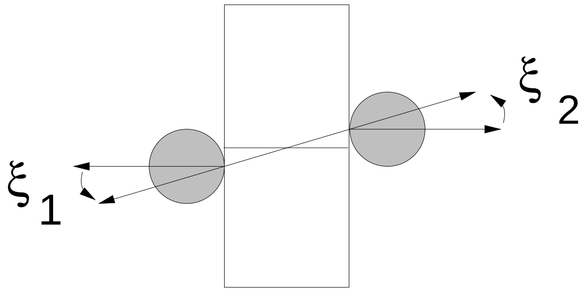


Figure 4.11: *Friction angles when grasping a rectangle.*

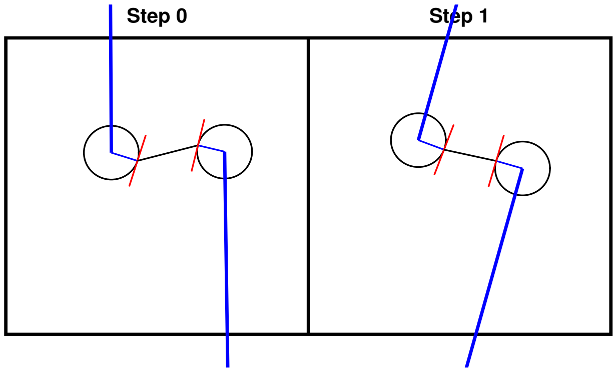


Figure 4.12: *Regrasping of a rectangle using tactile feedback. The left image shows the tactile position feedback from the initial grasp. The right figure shows the grasp configuration after regrasping. The surface tangent lines are derived directly from the contact locations.*

can easily be integrated into a single grasping algorithm if the class of objects to be grasped is restricted only to rectangles and disks. If the initial grasp of the object yields $\mathrm{sgn}(\xi_1) = \mathrm{sgn}(\xi_2)$ then the object is rectangular, otherwise it is a disk. If the initial grasp yields $\xi_i = 0$ then a simple rolling of one contact point will be required for classification. Once the object has been classified then the appropriate regrasping or manipulation strategy can be used to achieve the optimal grasp.

4.3.4 Rolling around vertices

So far we have discussed the grasping of circles and rectangles, but what about general polygons? Addressing a general regrasping and manipulation strategy for polygons is beyond the scope of this dissertation, however we can investigate what to expect from tactile feedback when a finger rolls from an edge of the polygon to a vertex. We would expect that the contact location should move while rotating along an edge, however it should stay constant when rolling around a vertex. In addition we expect local curvature estimates to indicate a large r_c along the edge and a small r_c when rotating about the vertex. Figure 4.14 illustrates this phenomena when rotating about the vertex of a rectangle. As expected the contact location remains constant around the vertex and has the same angular velocity as the link when rolling on the edge. It is curious that the contact velocity smoothly goes to zero, where one might expect it to abruptly go to zero when the vertex is reached. This

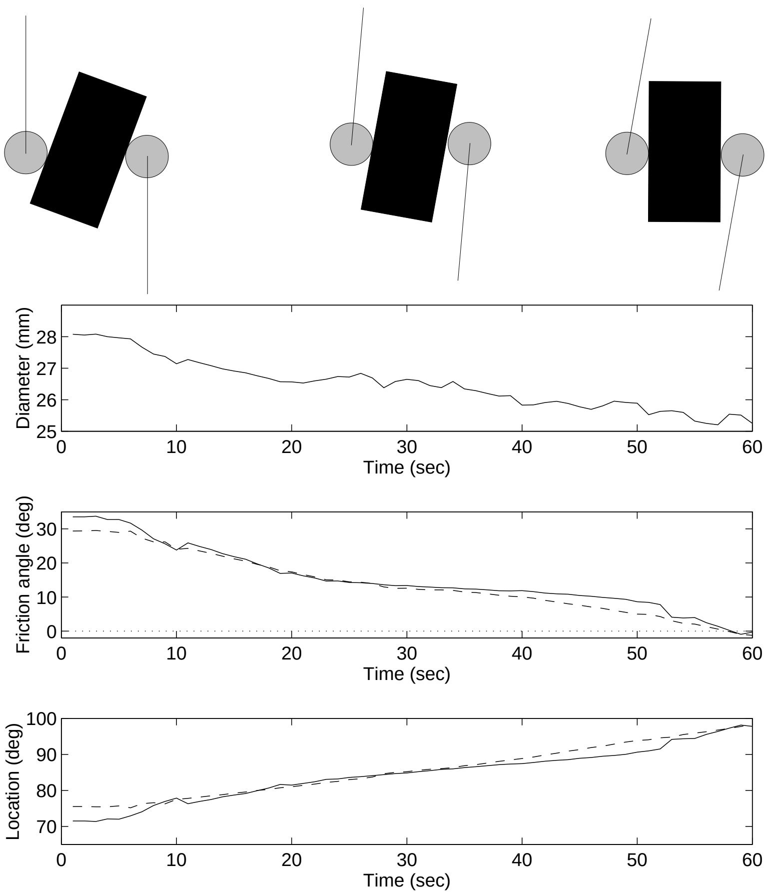


Figure 4.13: *Rotation of a rectangle within the fingers to achieve a parallel grasp.* $\dot{\phi}_1 = \dot{\phi}_2 = 0.2$ deg/sec. *Again the distance between the contact points is shown along with the friction angles, ξ_i, and the contact locations θ_{c_i}.*

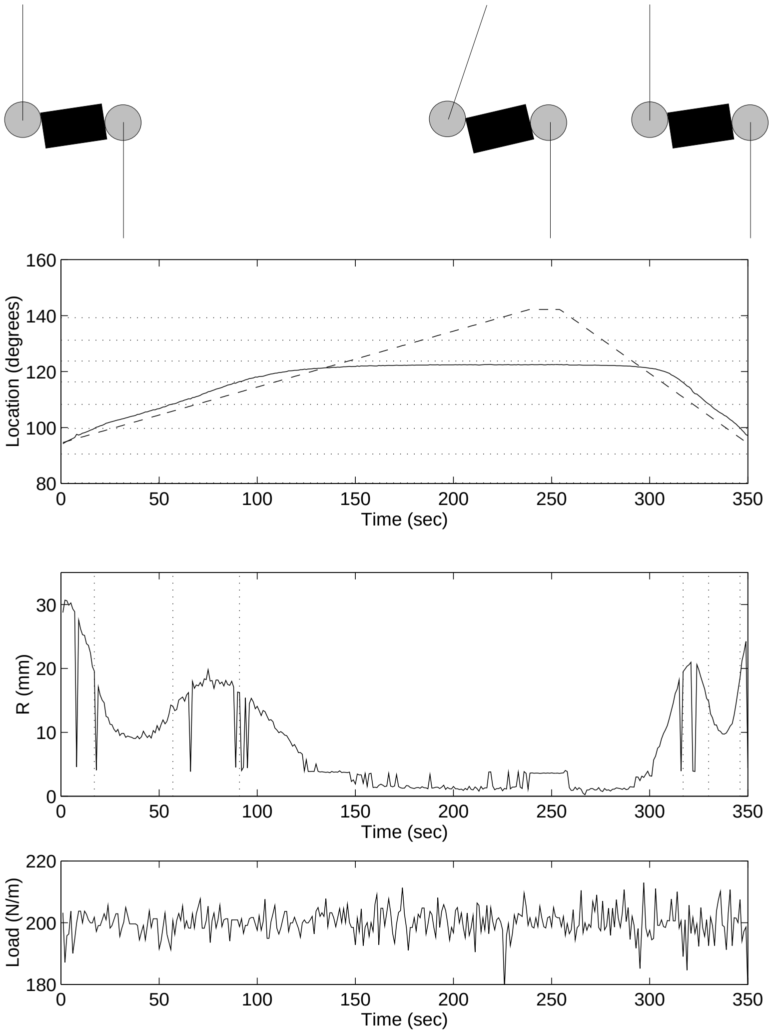


Figure 4.14: *Rolling a finger from a face to a vertex and back.* $\dot{\phi}_1 = 0.2$ deg/sec *until the vertex is reached. The configuration is held for 10 seconds and then the finger rotates back to the face with* $\dot{\phi}_1 = -0.5$ deg/sec. *The top plot shows the contact location,* θ_{c_1} *(solid), and the frictionless contact location,* ϕ_1 *(dashed), while the middle plot shows the estimated* R. *The bottom plot shows the grasp force. The dotted lines show the sensor locations. While rolling along the edge,* ϕ_1 *and* θ_{c_1} *remain the same, however, around the vertex, when* R *is small, the contact location remains constant.*

can most likely be attributed to the fact that the fingers are compliant. In addition we see that the radius of curvature estimates also undergo a smooth transition. As we have argued that curvature estimates mainly come from contact area this makes sense. In the transition from an edge to a vertex the contact area will smoothly change resulting in estimated radii in between that of a flat contact and a vertex.

Aside from the obvious drop-out in the R estimates, due most likely to the nonlinear optimization routine getting stuck in local minima, it is curious to again see that the variation in the R estimate has the same periodicity of that of the sensors, 10^o. This once again points out that the inclusion of copper strips in the material may require a nonlinear estimate of the $\mathbf{E}$ map in order to improve estimates of R. On the positive side the variance in R due to noise is small.

From this investigation of rolling around vertices it is clear that valuable information can be used from the tactile sensors during grasping of polygons to determine when a face is being touched and when a vertex is being touched. Curvature estimates from an initial static grasp can be used as an initial estimate of the contact feature while contact velocity can be used during manipulation as confirmation.

4.4 Summary

We have demonstrated the use of tactile sensing for regrasping and manipulation to improve grasp quality. This is by no means a thorough investigation of the topic, but some initial conclusions about the use of tactile sensing in grasping can be made. First it is clear that contact location information is invaluable in the monitoring of grasp stability. For a two fingered hand this information is enough to determine the finger closest to sliding and the appropriate way to adjust the grasp. A complete strategy has been demonstrated for disks and rectangles that requires no a priori information about the object.

Extension of this approach to general planar objects would require more sophisticated classification techniques to acquire a globally optimum grasp. However, for the class of convex objects, it would be possible to use local methods to roll around an object to find a parallel grasp. For example, from an initial grasp of an ellipse it could be classified initially as being like a rectangle, with $\mathrm{sgn}(\xi_1) = \mathrm{sgn}(\xi_2)$, or a disk. The first case amounts to the fingers being at opposite ends of the ellipse, and in the latter the fingers are at the same case. In either case the rolling strategy for the disk or rectangle will lead to a grasp at the minimal diameter.

Chapter 5

Conclusions

This dissertation has investigated the sensing capabilities of linear elastic cylindrical fingers through both a linear elastic model and sensing experiments. In addition grasping and manipulation of convex planar objects was demonstrated. We have seen that subsurface strain sensors can reliably provide the high resolution contact location information that is necessary for obtaining and maintaining stable grasps. Reliable estimation of curvature was shown to be more difficult, especially during grasping. At this point we do have a better understanding of the capabilities of such sensors and how they should be used.

Study of the human tactile system revealed that sensors were at a 20 % higher density and half the depth as the ones constructed in this dissertation. Psychophysiological experiments demonstrated that static discrimination of fine surface features is possible. The curvature estimation experiments in Chapter 3 of this dissertation showed that indenter radii could be estimated to 0.6 mm. Goodwin's studies, however, did not indicate if it was indenter curvature or contact area that was being perceived. Certainly the results of Chapter 2 indicate that it is likely that curvature is being inferred from contact area. In fact in some of Goodwin's results there was a bias towards perceiving a smaller curvature (a flatter contact) at higher loads for the same radius indenter. This would indicate that subjects perhaps cuing on the area of contact. It is also well known that contact area plays an important roll in the coefficient of friction, so it is interesting to see its importance in tactile sensing. Both robotic and human grasping research should perhaps focus on whether contact area is a more relevant and reliable variable than curvature. In particular, do humans modulate the coefficient of friction with a grasped object by changing the contact area, using tactile feedback to control the contact area?

We saw in Chapter 2 the relation between the spatial frequency cut-off and the depth of the rubber layer. This well-known phenomena gives a clear indication of what sensor densities should be to remove aliasing effects from the sampled subsurface strain. When the strain field is sampled at the appropriate density, Chapter 3 showed that contact location could be estimated to a resolution of $\frac{1}{25}$th of that of the sensor spacing. As we saw in Chapter 4, contact location is the most important information that a tactile sensor can provide Two fingered grasping of disks and rectangles can be accomplished solely with this information. With an aliased sensor, such as the displacement sensor discussed in Chapter 1, such accurate estimates of contact location are not possible, making grasp stability analysis difficult.

The relevance of shear sensing was discussed in Chapter 2 and it was shown that shear sensing would be dominated at the low frequencies by the tangential loading. Based on that analysis, shear sensing will not be helpful in the general problem of determining indenter shape from subsurface strain. It is more likely that shear information could be used to better estimate tangential loading on fingers and hence allow a better estimate of grasp stability. Due to the depth of shear sensors in the human hand, it is likely that this is how they are used by humans.

In Chapter 2 we also investigated the depth at which indenter classification is theoretically possible given the current sensor noise. It was shown that for a ratio of core radius to outer radius of less than 0.85 classification is not possible, however the scale of an indenter within a class can be determined. For shallow sensors, with this ratio greater than 0.85 classification should be possible. Chapter 3 investigated shape sensing capabilities within the class of round indenters using a cylindrical tactile array with radius ratio of 0.75. With current construction techniques, we were not able to make a tactile array with the required sensor density for a radius ratio greater than 0.85.

Chapter 3 also revealed the importance of calibration techniques on the reliability of curvature estimates from tactile sensors. Although it is clear that some sensor elements fit the linear elastic model well, other sensor elements do not. This implies that linear elastic models can be used to discuss the capabilities of tactile sensors, but not to predict responses from an actual sensor. To predict responses it is better to fit a general linear map between the fourier coefficients of pressure distributions and the sampled strain values. This approach was shown to almost halve the variance of the estimated curvatures. This fitted map has similar characteristics to the linear elastic model as was shown by its singular value decomposition. Both the model and the empirical derived map indicate that the map from surface pressure to sampled strain is one of a spatial band-pass filter with a very low center frequency. So far only linear techniques have been used to estimate this map. Newer nonlinear techniques might produce better results.

At this point we can say we have adequately addressed the questions brought up in Chapter 1. We have shown that subsurface sensing is better than surface sensing for determining contact location and contact area. We have shown the importance of contact location feedback in grasping. Curvature estimates are useful when an object model is not available. Calibration techniques showed that a general linear model is better for an actual sensor, while a linear elastic model is adequate for theoretical analysis. We have seen that a cylindrical model which takes into account the rigid core of the sensor is a better predictor of sensor responses than a half-plane model. Analysis of the capacitive sensor indicated that shielding is necessary. Finally we saw that by using an external source of contact load information the nonlinear estimation of curvature could be accomplished in real-time (40 ms).

Bibliography

Allen, P. K. (1987). *Robotic Object Recognition Using Vision and Touch.* Kluwer Academic Publishers, Norwell, MA.

Allen, P. K. (1992). Object recognition using active tactile sensing. In Nicholls, H. R., editor, *Advanced Tactile Sensing for Robotics*, chapter 10, pages 221–248. World Scientific, New Jersey.

Bailey, D. M. and Sayles, R. S. (1991). Effect of roughness and sliding friction on contact stresses. *Journal of Tribology*, 113:729–738.

Basu, P. R., Russell, R. A., and Trott, G. (1988). Extracting 3-dimensional surface features from tactile sensor data. In *International Symposium and Exposition on Robots*, pages 908–920, Sydney, Australia.

Bentall, R. H. and Johnson, K. L. (1967). Slip in the rolling contact of two dissimilar elastic rollers. *International Journal Mechanical Sciences*, 9:389–404.

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

Berkemeier, M. D. (1990). Determining the axis of a surface of revolution using tactile sensing. Master's thesis, U.C. Berkeley.

Berkemeier, M. D. and Fearing, R. S. (1993). Determining the axis of a surface of revolution using tactile sensing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1079–1087.

Bogy, D. B. (1991). Lecture notes from course on linear elasticity. Mechanical Engineering Department, U.C. Berkeley.

Boie, R. A. (1984). Capacitance impedance readout tactile image sensor. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 370–378, Atlanta, Georgia.

Booch, G. (1991). *Object oriented design with applications.* The Benjamin/Cummings Publishing Company, Inc,, Redwood City, CA.

Bowden, F. P. and Tabor, D. (1950). *The friction and lubrication of solids.* Clarendon Press, Oxford.

Canepa, G., Morabito, M., De Rossi, D., Caiti, A., and Parisini, T. (1992). Shape from touch by a neural net. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 2075–2080, Nice, France.

Chen, J. B., Fearing, R. S., Armstrong, B. S., and Burdick, J. W. (1986). Nymph: A multiprocessor for manipulation applications. In *Proceedings, IEEE International Conference on Robotics and Automation*, San Francisco, CA.

Constantinesc, A. (1993). A numerical investigation of the elastic moduli in an inhomogeneous body. In Zabaras, N., Woodbury, K. A., and Raynaud, M., editors, *Inverse problems in engineering : theory and practice*, pages 77–84, Palm Coast, Florida. American Society of Mechanical Engineers.

Cutkosky, M. R., Jourdain, J. M., and Wright, P. K. (1987). Skin materials for robotic fingers. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 1649–1654, Washington, DC.

Cutkosky, M. R. and Wright, P. K. (1986). Friction, stability and the design of robotic fingers. *The International Journal of Robotics Research*, 5(4):20–37.

Dinc, O. S., Ettles, C. M., Calabrese, S. J., and Scarton, H. A. (1991). Some parameters affecting tactile friction. *Transactions of the ASME*, 113:512–517.

Dupont, P. E. and Dunlap, E. P. (1993). Friction modeling and control in boundary lubrication. In *American Control Conference*, pages 1910–1914, San Francisco, California.

Egan, J. I. and Teixeira, T. J. (1992). *Writing a UNIX Device Driver.* John Wiley and Sons, New York, 2nd edition.

Ellis, R. E. and Qin, M. (1994). Singular-value and finite-element analysis of tactile shape recognition. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 2529–2535, San Diego, CA.

Fearing, R. S. (1989). Some experiments with tactile sensing during grasping. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 1637–1643, Scottsdale, Arizona.

Fearing, R. S. (1990). Tactile sensing mechanisms. *The International Journal of Robotics Research*, 9(3):3–23.

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

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

Ferrari, C. and Canny, J. (1992). Planning optimal grasps. In *Proceedings, IEEE International Conference on Robotics and Automation*, volume 3, pages 2290–2295, Nice, France.

Fraser, C. J., Milne, J. S., and Logan, G. M. (1993). An educational perspective on applied mechatronics. *Mechatronics*, 3(1):49–57.

Goodwin, A. W., John, K. T., and Marceglia, A. H. (1991). Tactile discrimination of curvature by humans using only cutaneous information from the fingerpads. *Experimental Brain Research*, 86(3):663–672.

Gordon, I. E. and Morison, V. (1982). The haptic perception of curvature. *Perception and Psychophysics*, 31(5):446–450.

Goyal, S., Ruina, A., and Papadopoulos, J. (1991). Planar sliding with dry friction. 1. Limit surface and moment function. *Wear*, 143(2):307–30.

Gradshtein, I. S. and Ryzhik, I. M. (1980). *Table of integrals, series, and products*. Academic Press, New York.

Hackwood, S., Beni, G., Hornak, L., Wolfe, R., and Nelson, T. J. (1983). A torque-sensitive tactile array for robotics. *The International Journal of Robotics Research*, 2(2):46–50.

Hillis, W. D. (1982). A high-resolution imaging touch sensor. *The International Journal of Robotics Research*, 1(2):33–44.

Howe, R. D. and Cutkosky, M. R. (1989). Sensing skin acceleration for slip and texture perception. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 145–150, Scottsdale, Arizona.

Howe, R. D., Popp, N., Akella, P., Kao, I., and Cutkosky, M. R. (1990). Grasping, manipulation, and control with tactile sensing. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 1258–1263, Cincinatti, OH.

Hutchings, B. L., Grahn, A. R., and Petersen, R. J. (1994). Multiple-layer cross-field ultrasonic tactile sensor. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 2522–2528, San Diego, CA.

Iggo, A. and Andres, K. H. (1982). Morphology of cutaneous receptors. *Annual Review of Neuroscience*, 5:1–31.

Jameson, J. W. (1985). *Analytic Techniques for Automated Grasp.* PhD thesis, Stanford University.

Jameson, J. W. and Leifer, L. J. (1987). Automatic grasping: An optimization approach. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-17(5):806–813.

Johansson, R. S. and Vallbo, A. B. (1979). Tactile sensibility in the human hand: relative and absolute densities of four types of mechanoreceptive units in glabrous skin. *Journal of Physiology*, 286:283–300.

Johnson, K. L. (1985). *Contact Mechanics.* Cambridge University Press, Cambridge, England.

Johnson, K. O. and Hsiao, S. S. (1992). Neural mechanisms of tactual form and texture perception. *Annual Review of Neuroscience*, 15:227–50.

Kao, I. (1990). *Quasistatic Manipulation with Compliance and Friction.* PhD thesis, Stanford University.

LaMotte, R. H. and Srinivasan, M. A. (1987). Tactile discrimination of shape: Responses of slowly adapting mechanoreceptive afferents to a step stroked across the monkey fingerpad. *The Journal of Neurosciences*, 7(6):1655–1671.

Li, Z. and Sastry, S. S. (1988). Task-oriented optimal grasping by multifingered robot hands. *IEEE Journal of Robotics and Automation*, 4(1):32–44.

LynxOS. *LynxOS User's Manual.* Lynx Real-Time Systems, Inc., Los Gatos, CA.

Maekawa, H., Komoriya, K., and Tanie, K. (1992a). 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, Raleigh, North Carolina.

Maekawa, H., Tanie, K., Komoriya, K., Kaneko, M., Horiguchi, C., and Sugawara, T. (1992b). Development of a finger-shaped tactile sensor and its evaluation by active touch. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 1327–1334, Nice, France.

Maniatty, A. M. (1993). A numerical solution with error estimating for inverse elasticity problems. In Zabaras, N., Woodbury, K. A., and Raynaud, M., editors, *Inverse problems in engineering : theory and practice*, pages 179–185, Palm Coast, Florida. American Society of Mechanical Engineers.

Mirtich, B. and Canny, J. (1994). Easily computable optimum grasps in 2-d and 3-d. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 739–747, San Diego, CA.

Miu, D. K. (1993). *Mechatronics: Electromechanics and Contromechanics.* Springer-Verlag, New York.

Monkman, G. J. (1993). 3-D tactile image display. *Sensor Review*, 13(2):27–31.

Montana, D. J. (1989). The kinematics of contact with compliance. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 770–775, Scottsdale, Arizona.

Murray, R. M., Li, Z., and Sastry, S. S. (1994). *A Mathematical Introduction to Robotic Manipulation.* CRC Press, Boca Raton.

Murray, R. M. and Sastry, S. S. (1989). Control experiments in planar manipulation and grasping. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 624–629, Scottsdale, Arizona.

Nabors, K. and White, J. (1991). Fastcap: A multipole accelerated 3-d capacitance extraction program. *IEEE Transactions on Computer-Aided Design*, 10(11):1447–1459.

Nicholls, H. R., editor (1992). *Advanced Tactile Sensing for Robotics*. World Scientific, New Jersey.

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

Nicolson, E. J. (1994). Standardizing I/O for mechatronic systems (SIOMS) using real-time unix device drivers. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 3489–3494, San Diego, CA.

Nicolson, E. J. and Fearing, R. S. (1993). Compliant control of threaded fastener insertion. In *Proceedings, IEEE International Conference on Robotics and Automation*, volume 1, pages 484–490, Atlanta, GA.

Nowell, D. and Hills, D. A. (1988). Tractive rolling of tyred cylinders. *International Journal of Mechanical Sciences*, 30(12):945–957.

Oberhettinger, F. (1990). *Fourier transforms of distributions and their inverses; a collection of tables*. Springer-Verlag, New York.

Paulos, E. and Canny, J. (1994). Accurate insertion strategies using simple optical sensors. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 1656–1662.

Phillips, J. R. and Johnson, K. O. (1981). Tactile spatial resolution. III. A continuum mechanics model of skin predicting mechanoreceptor responses to bars, edges, and gratings. *Journal of Neurophysiology*, 46(6):1204–1225.

Poritsky, H. (1950). Stresses and deflections of cylindrical bodies in contact with application to contact of gears and locomotive wheels. *Journal of Applied Mechanics*, 17(2):191–201.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992). *Numerical Recipes in C : The Art of Scientific Computing.* Cambridge University Press, New York, 2nd edition.

Raibert, M. H. and Tanner, J. E. (1982). Design and implementation of a vlsi tactile sensing computer. *The International Journal of Robotics Research*, 1(3):3–18.

Reynaerts, D. and Brussel, H. V. (1993). Tactile sensing data interpretation for object manipulation. In *Sensors and Acutators A*, pages 268–273.

Ricker, S. L. and Ellis, R. E. (1992). Some implications of solid mechanics for tactile feedback in teleoperation. In *SPIE Proceedings 1833*, pages 208–219, Boston, MA.

Russell, R. A. (1987). Compliant-skin tactile sensor. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 1645–1648, Raleigh, NC.

Schallamach, A. (1952). The load dependence of rubber friction. *The Proceedings of the Physical Society, Section B*, 65(9):657–661.

Schallamach, A. (1969). Friction and frictional rise of wedge sliders on rubber. *Wear*, 13:13–25.

Scheinman, V. (1987). RobotWorld: A multiple robot vision guided assembly system. In *Proceedings of the 4th International Symposium on Robotics Research.*

Sherrick, C. E. and Cholewiak, R. W. (1986). Cutaneous sensitivity. In Boff, K. R., Kaufman, L., and Thomas, J. P., editors, *Handbook of perception and human performance*, chapter 12, pages 12–1 – 12–58. Wiley.

Shimojo, M. (1994). Spatial filtering chracteristic of elastic cover for tactile sensor. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 287–292, San Diego, CA.

Shimojo, M., Ishikawa, M., and Kanaya, K. (1991). A flexible high resolution tactile imager with video signal output. In *Proceedings, IEEE International Conference on Robotics and Automation*, volume 1, pages 384–389, Sacramento, CA.

Siegel, D. M., Drucker, S. M., and Garabieta, I. (1987). Performance analysis of a tactile sensor. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 1493–1499, Raleigh, NC.

Sikka, P. and Zhang, H. (1992). Tactile-assisted robotic object manipulation. In *Fourth International Symposium on Robotics and Manufacturing*, pages 83–88, Santa Fe, New Mexico. ASME press.

Sladek, E. M. and Fearing, R. S. (1990). The dynamic response of a tactile sensor. In *Proceedings, IEEE International Conference on Robotics and Automation*, volume 2, pages 962–967, Cincinnati, OH.

Sokolnikoff, I. S. (1941). *Mathematical Theory of Elasticity*. Robert E. Krieger Publishing Company, Malabar, Florida. Notes from Summer session for advanced instruction and research in mechanics.

Speeter, T. H. (1988). Flexible, piezoresistive touch sensing array. In *SPIE - The International Society for Optical Engineering*, volume 1005, pages 31–43.

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

Stansfield, S. A. (1988). A robotic perceptual system utilizing passive vision and active touch. *The International Journal of Robotics Research*, 7(6):138–161.

Stewart, D. B., Schmitz, D. E., and Khosla, P. K. (1992). The Chimera II real-time operating system for advanced sensor-based control applications. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(6):1282–1295.

Sugiyama, S., Kawahata, K., Yoneda, M., and Igarashi, I. (1990). Tactile image detection using a 1k-element silicon pressure sensor array. In *Sensors and Acutators A*, volume 22, pages 397–400.

Timoshenko, S. P. and Goodier, J. N. (1970). *Theory of Elasticity*. McGraw-Hill, New York, 3d edition.

Tremblay, M. R. and Cutkosky, M. R. (1993). Estimating friction using incipient slip sensing during a manipulation task. In *Proceedings, IEEE International Conference on Robotics and Automation*, volume 1, pages 429–434, Atlanta, GA.

Vallbo, A. B. and Johansson, R. S. (1984). Properties of cutaneous mechanoreceptors in the human hand related to touch sensation. *Human Neurobiology*, 3(1):3–14.

Van Huffel, S. and Vandewalle, J. (1991). *The Total Least Squares Problem : Computational Aspects and Analysis*. Society for Industrial and Applied Mathematics, Philadelphia.

Vierck Jr, C. J. and Jones, M. B. (1969). Size discrimination on the skin. *Science*, 163(866):488–489.

Wallack, A., Canny, J., and Manocha, D. (1993). Object localization using crossbeam sensing. In *Proceedings, IEEE International Conference on Robotics and Automation*, volume 1, pages 692–699, Atlanta, GA.

Weber, E. H. (1978). *The sense of touch*. Academic Press, New York. (*De Tactu*, H. E. Ross, trans. and *Der Tastsinn*, D. J. Murray, trans. Originally published in 1826).

Weinstein, S. (1968). Intensive and extensive aspects of tactile sensitivity as a function of body part, sex, and laterality. In Kenshalo, D. R., editor, *The skin senses*. Thomas, Springfield, Illinois.

Williams, T. (1990). Bios standard would link real-time kernels to new hardware. *Computer Design*, 29(9):24+.

Williams, T. (1991). Real-time unix develops multiprocessing muscle. *Computer Design*, 30(5):26+.

Appendix A

Solutions to the linear elastic model

$$
B_1(r) = \begin{bmatrix} -\frac{\left(\gamma+\alpha\beta^3\hat{\sigma}\right)}{1+\alpha^4\hat{\sigma}} \\ \frac{\left(\gamma+\alpha\beta^3\hat{\sigma}\right)}{1+\alpha^4\hat{\sigma}} \\ \frac{r_b\left(\gamma^2(2-\hat{\sigma})-\alpha^2\left(2-\beta^2\hat{\sigma}\right)\right)}{4\mu(1+\alpha^4\hat{\sigma})} \\ \frac{r_b\left(\gamma^2(2+\hat{\sigma})-\alpha^2\left(2+\beta^2\hat{\sigma}\right)\right)}{4\mu(1+\alpha^4\hat{\sigma})} \\ \frac{\gamma\left(2-\hat{\sigma}-\beta^4\hat{\sigma}\right)}{2\mu(1+\alpha^4\hat{\sigma})} \end{bmatrix}
$$

$$
A_1(r) = \begin{bmatrix} \frac{1}{2\gamma(1+\hat{\sigma})}\left(3+\hat{\sigma}+\frac{2\left(-\beta^2+\alpha^2\gamma^2\right)+(-1+\hat{\sigma})\left(\gamma^2+\alpha^2\beta^2\hat{\sigma}\right)}{1+\alpha^4\hat{\sigma}}\right) \\ \frac{-1}{2\gamma(1+\hat{\sigma})}\left(1-\hat{\sigma}+\frac{2\left(-\beta^2+\alpha^2\gamma^2\right)+(-1+\hat{\sigma})\left(\gamma^2+\alpha^2\beta^2\hat{\sigma}\right)}{1+\alpha^4\hat{\sigma}}\right) \\ \frac{r_b}{2\mu(1+\hat{\sigma})}\left(-\frac{1}{2}-\hat{\sigma}\log(\beta)+\frac{2\alpha^4+\beta^2+\alpha^2\gamma^2(-2+\hat{\sigma})+(-1+\hat{\sigma})\left(\alpha^2-\gamma^2+\frac{\hat{\sigma}}{2}\left(-\alpha^2\beta^2+\gamma^2\right)\right)}{2(1+\alpha^4\hat{\sigma})}\right) \\ \frac{r_b}{2\mu(1+\hat{\sigma})}\left(\frac{1}{2}-\hat{\sigma}\log(\beta)+\frac{(-1+\hat{\sigma})\left(\alpha^2-\gamma^2-\frac{\hat{\sigma}}{2}\left(-\alpha^2\beta^2+\gamma^2\right)\right)-\left(-2\alpha^4+\beta^2+\alpha^2\gamma^2(2+\hat{\sigma})\right)}{2(1+\alpha^4\hat{\sigma})}\right) \\ \frac{\gamma}{2\mu(1+\hat{\sigma})}\left(\frac{\hat{\sigma}}{\gamma^2}+\frac{-\frac{\beta^2}{\gamma^2}+\alpha^2(-2+\hat{\sigma})+(-1+\hat{\sigma})\left(-1+\frac{\hat{\sigma}}{2}\left(1+\beta^4\right)\right)}{1+\alpha^4\hat{\sigma}}\right) \end{bmatrix}
$$

$$
\delta_k = \left(1-\alpha^2\right)^2\left(1-k^2\right)-\left(\alpha^{-2k}+\alpha^2\hat{\sigma}\right)\left(\alpha^{2k}+\alpha^2\hat{\sigma}\right)
$$

$$
B_k = \begin{bmatrix} 1-\alpha^2 & -\frac{1+\alpha^{2-2k}\hat{\sigma}}{1+k} \\ -\frac{1+\alpha^{2+2k}\hat{\sigma}}{1-k} & 1-\alpha^2 \end{bmatrix}
$$

$$L = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$A_k(r) = \begin{bmatrix} \gamma^{k}(1+k)\left(2-k+\beta^{2+2k}\hat{\sigma}-\beta^{2}(1-k)\right) & \gamma^{-k}(1-k)\left(2+k+\beta^{2-2k}\hat{\sigma}-\beta^{2}(1+k)\right) \\ \gamma^{k}(1+k)\left(-k-\beta^{2+2k}\hat{\sigma}+\beta^{2}(-1+k)\right) & \gamma^{-k}(-1+k)\left(k-\beta^{2-2k}\hat{\sigma}-\beta^{2}(1+k)\right) \\ \frac{1}{2\mu}\gamma^{k}r\left(\hat{\sigma}\left(1-\beta^{2+2k}\right)+\left(-1+\beta^{2}\right)(1+k)\right) & \frac{1}{2\mu}\gamma^{-k}r\left(\hat{\sigma}\left(1-\beta^{2-2k}\right)+\left(-1+\beta^{2}\right)(1-k)\right) \\ \frac{1}{2\mu}\gamma^{k}r\left(\hat{\sigma}\left(-1+\beta^{2+2k}\right)+\left(-1+\beta^{2}\right)(1+k)\right) & \frac{1}{2\mu}\gamma^{-k}r\left(\hat{\sigma}\left(1-\beta^{2-2k}\right)+\left(1-\beta^{2}\right)(1-k)\right) \\ \frac{1}{2\mu}\gamma^{k}(1+k)\left(-1-k+\hat{\sigma}+\beta^{2+2k}\hat{\sigma}+\beta^{2}(-1+k)\right) & \frac{1}{2\mu}\gamma^{-k}(1-k)\left(-1+k+\hat{\sigma}+\beta^{2-2k}\hat{\sigma}-\beta^{2}(1+k)\right) \end{bmatrix}$$

Appendix B

Half plane model

Without a model for a cylindrical geometry sensor, half-plane models have been used as an approximation in previous work, (Fearing and Hollerbach, 1985; Fearing, 1990). This appendix reviews the half-plane equations and derives the frequency response. The coordinate system used in shown in Figure B.1.

Johnson (Johnson, 1985) derives the following stress and strain equations for the half plane linear elastic plane strain model. P is the normal load and Q is the tangential load in units N/m for a line contact.

$$
\begin{aligned}
\tau_{11} &= \frac{-2}{\pi} \frac{P x_1^2 x_2 + Q x_1^3}{\left(x_1^2 + x_2^2\right)^2} \\
\tau_{22} &= \frac{-2}{\pi} \frac{P x_2^3 + Q x_1 x_2^2}{\left(x_1^2 + x_2^2\right)^2} \\
\tau_{12} &= \frac{-2}{\pi} \frac{P x_1 x_2^2 + Q x_1^2 x_2}{\left(x_1^2 + x_2^2\right)^2} \qquad \text{(B.1)}
\end{aligned}
$$

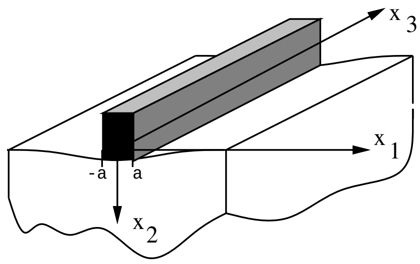


Figure B.1: *Half-plane indentation.*

$$
\begin{aligned}
e_{22} &= \frac{1}{E}\Big((1-\sigma^2)\tau_{22} - \sigma(1+\sigma)\tau_{11}\Big) \\
e_{12} &= \frac{1+\sigma}{E}\tau_{12}
\end{aligned} \tag{B.2}
$$

Defining the fourier transform as

$$
\begin{aligned}
F(\omega) &= \int_{-\infty}^{\infty} f(x)e^{-j\omega x}dx \\
f(x) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)e^{j\omega x}d\omega
\end{aligned}
$$

and using following fourier transform pair (Gradshtein and Ryzhik, 1980)

$$
\frac{2a}{a^2+x^2} \quad \Leftrightarrow \quad 2\pi e^{-a|\omega|}
$$

the convolution and differentiation properties can be used to obtain

$$
\begin{aligned}
\frac{2}{\pi}\frac{a^3}{(a^2+x^2)^2} &\quad \Leftrightarrow \quad e^{-|a\omega|}(a\omega+1) \\
\frac{2}{\pi}\frac{a^2x}{(a^2+x^2)^2} &\quad \Leftrightarrow \quad -ja\omega e^{-|a\omega|} \\
\frac{2}{\pi}\frac{ax^2}{(a^2+x^2)^2} &\quad \Leftrightarrow \quad e^{-|a\omega|}(1-a\omega) \\
\frac{2}{\pi}\frac{x^3}{(a^2+x^2)^2} &\quad \Leftrightarrow \quad je^{-|a\omega|}(a\omega-2)
\end{aligned}
$$

By using these transform pairs the half-plane normal and shear strain frequency responses may be found

$$
\begin{aligned}
\hat{\tau}_{11}(\omega) &= -e^{-x_2\omega}\left(p(1-x_2\omega) + jq(x_2\omega-2)\right) \\
\hat{\tau}_{22}(\omega) &= -e^{-x_2\omega}\left(p(1+x_2\omega) - jqx_2\omega\right) \\
\hat{\tau}_{12}(\omega) &= -e^{-x_2\omega}\left(-pjx_2\omega + q(1-x_2\omega)\right).
\end{aligned}
$$

Using equations B.2 the subsurface normal and shear strain impulse response functions can be found.

Care must be taken when comparing the fourier series coefficients of the cylindrical model to the fourier transform of the half plane model. A fourier series coefficient k is equivalent to the half plane frequency $\omega = 2\pi\frac{k}{2\pi r_b} = \frac{k}{r_b}$. In addition, to compare magnitudes in the frequency domain, $\hat{e}_{22}(0) = 2\pi r_b p_0$ and $\|\hat{e}_{22}(\frac{k}{r_b})\| = \pi r_b \|p_k^s + p_k^c\|$.

Appendix C

Hertz and other closed form contact models

Though the fourier basis is well suited to analyzing the pressure to strain problem, it is difficult to get high numerical accuracy for pressure profiles with a large high frequency component due to truncation and leakage. This appendix summarizes closed-form expressions for a set of different contact types as derived in (Johnson, 1985).

C.1 The Hertz contact

Under the assumption of frictionless contact between a cylinder and an elastic half space the Hertz model predicts an elliptical pressure distribution. (See (Johnson, 1985) page 129 and following.) For the contact of a rigid cylinder and and elastic cylinder, as shown in Figure C.1, the equivalent radius, R, may be used. It should be noted that the Hertz assumption is no longer valid for cylinders with a rigid core (Nowell and Hills, 1988), however when the contact area is small and the elastic layer is thick the approximation is sufficient.

The elliptic pressure distribution is given by

$$p(\theta) = \frac{\mu}{1-\sigma}\left(1+\frac{r_b}{r_c}\right)\left(\left(\frac{\theta_w}{2}\right)^2 - \theta^2\right)^{\frac{1}{2}} \tag{C.1}$$

where the arc of contact, in radians, θ_w, is given by

$$\theta_w = \frac{2}{r_b}\sqrt{\frac{2P(1-\sigma)}{\pi\mu}\left(\frac{1}{r_b}+\frac{1}{r_c}\right)} \tag{C.2}$$

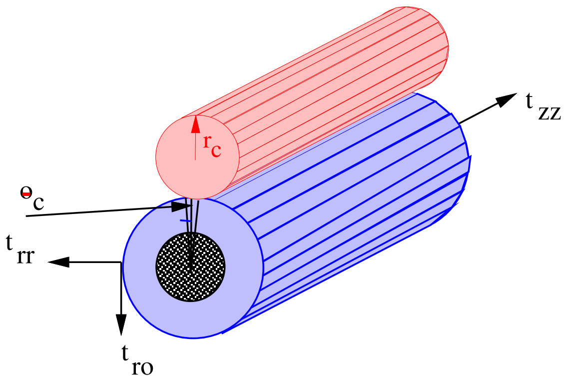


Figure C.1: *Hertzian contact of cylinders in plane strain.*

and P is the load in N/m.

We can find the fourier series coefficients for the Hertz contact, defined as

$$\begin{aligned} p_0 &= \frac{1}{2\pi}\int_{-\pi}^{\pi} p(\theta)d\theta \\ p_k^c &= \frac{1}{\pi}\int_{-\pi}^{\pi} p(\theta)\cos k\theta d\theta \\ p_k^s &= \frac{1}{\pi}\int_{-\pi}^{\pi} p(\theta)\sin k\theta d\theta, \end{aligned}$$

at θ_c from a standard table (Oberhettinger, 1990)

$$\begin{aligned} p_0 &= \frac{\mu}{4(1-\sigma)}\left(1+\frac{r_b}{r_c}\right)\left(\frac{\theta_w}{2}\right)^2 \\ p_k^s &= \frac{\mu}{2(1-\sigma)}\left(1+\frac{r_b}{r_c}\right)\left(\frac{\theta_w}{k}\right)J_1\left(\frac{\theta_w}{2}k\right)\sin(k\theta_c) \\ p_k^c &= \frac{\mu}{2(1-\sigma)}\left(1+\frac{r_b}{r_c}\right)\left(\frac{\theta_w}{k}\right)J_1\left(\frac{\theta_w}{2}k\right)\cos(k\theta_c). \end{aligned} \tag{C.3}$$

where J_1 is the Bessel function of the first kind.

C.2 Wedge contact

Johnson (Johnson, 1985), page 112, also gives the pressure distribution for a wedge contact on an elastic half-plane as

$$p(\theta) = \frac{2\mu}{\pi(1-\sigma)}\cot\theta_p \cosh^{-1}\left(\frac{\theta_w}{2\theta}\right) \tag{C.4}$$

where the contact area may be found from

$$\theta_w = P\frac{1-\sigma}{\mu r_b \cot\theta_p}. \tag{C.5}$$

Appendix D

Finger construction steps

In this procedure, "Upper mask" refers to a piece of copper which has been photomasked with the resist pattern for the upper electrodes and "lower mask" refers to the piece masked for the lower electrodes. Figure D.1 shows the sensor core and sense strips with the groove (wire channel) for the 8 conductor wire.

1. Prepare core
   (a) Choose radii r_a and r_b and sampling density.
   (b) Cut stock to length (32 mm final, 35 mm rough), lathe to desired radius, face ends (speed above 200, 3.0 inches/min).
   (c) Center drill, Drill # 7 hole, tap for 1/4-20 thread.
   (d) Cut 1/4-20 threaded rod stock.
2. Prepare mask

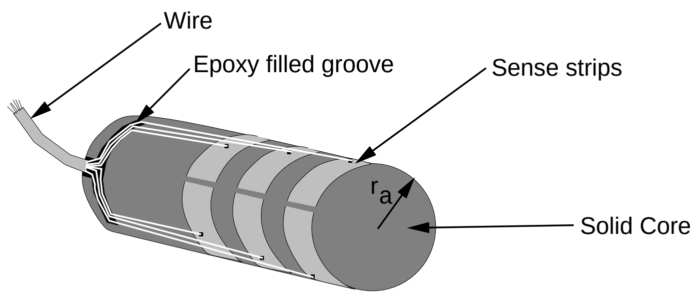


Figure D.1: *Finger core with wire channel.*

(a) Paint nail polish onto back of mask.

(b) Cut out bottom portion, leave about 0.25" copper at the end, cut sides close. Length should be $< 2\pi r_a$.

3. Attach lower mask

(a) Glue mask to core using Devcon 5 minute epoxy. Be sure not to get any epoxy on front copper since it will act as a resist. This is most important in the sensing area, but not so important at the ends. Sensing electrodes should be at the end opposite to the tapped end.

(b) Etch in Ferric Chloride. Add 25% hot (25 ml) (near boiling) water by volume to etching fluid to speed process. Swirl for about 20 minutes.

(c) Wash off in water. Let dry.

(d) Drill alignment hole. 1.6mm diameter $\simeq$ #52 approximately half way between center and outside. Err to the outside to leave enough room for the nut. The location is to the left side looking from the top with the front up. Drill Air escape holes , #29 drill, all the way through.

(e) Use the mill to cut channels for the wires. Turn on the monitor, turn on the mill power, turn on the air. Set up the vice with the pin in the third hole from the left. Insert the tool. Mount the sensor back facing up. Use manual control to find the approximate zero at 9in/min. Reduce the speed to 2in/min and take small z steps to find the correct height. Set the zero position from the Initialize menu. Load c:\ ed \ filename. Run. For new tool find new z origin.

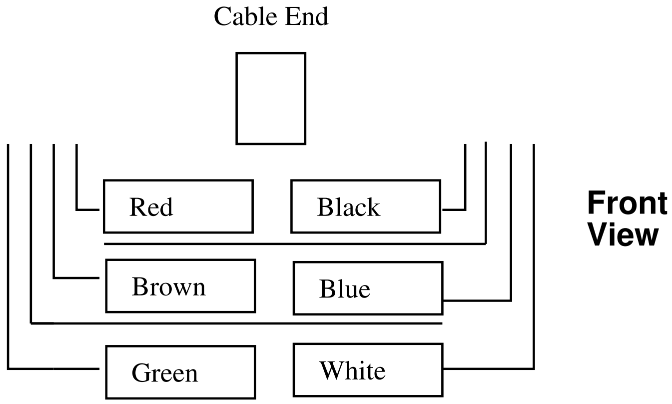
When doing the front, mill a small cut in the center, front, and bottom end to use for alignment with center of the upper mask. Use the 0.025" tool. Be very careful since the tool breaks easily. To make the notch in the bottom end, jog .05" in x, .01" deeper in z each time at a speed of 1"/min.

Mill Programs, mount core with $x + ive$ = cylinder axis to top				
$r_a * 2$	Side	Program	Tool	Origin
22 mm	Back	BC1	5/32" flat end	center back, cable end
		BC2	1/16" flat end	center back, cable end
	Front	FC1	5/32" flat end	center front, cable end + 5/64"
19.5 mm	Back	FC1	5/32" flat end	center back, cable end
		BC2	1/16" flat end	center back, cable end
	Front	FC1	5/32" flat end	center front, cable end + 5/64"

(f) Repair strips by gluing with "superglue" any loose traces. Cut off any copper that was not properly etched.

(g) Scratch off resist for wire connections at ends of traces. Tin these patches with a small amount of solder.

(h) Use 8 conductor, 30 gauge, shielded, Cooner wire (part number AS148). Tin 8 wires. Glue the wire casing into the core wire channel and solder to traces using the color code



(i) Test connections for shorts between traces and open circuits.

(j) Fill wire channel with 5 minute epoxy to glue the cable in place and immobilize it.

4. Attach upper mask

(a) Create dielectric.

i. Use the machinable wax mold with 0.025" ball end holes at 0.040" spacing to a depth of 0.004".

ii. Mix HSII rubber using a ratio of 10:1 by weight of base to catalyst. Mix by mushing more than stirring to avoid introducing too much air into the mixture.

iii. Smear enough on the mold to cover the holes, but not much extra.

iv. Vacuum for 3-4 minutes until large bubbles start to form. Do not over vacuum.

v. Press on a clear plexiglass cover and push hard around outside edges of the mold to ensure a thin layer.

vi. Leave weights on top of the plexiglass for about 12 hours.

(b) Glue on the upper mask using Silicone glue. Glue the mask to the smooth (not bumpy) side of the dielectric. Use a healthy amount of glue. Trim to 1/4 inch on the ends, but as close as possible on the top and bottom ends. Let this dry.

(c) Etch as in 3b. Be gentle so that the copper does not separate from the dielectric.

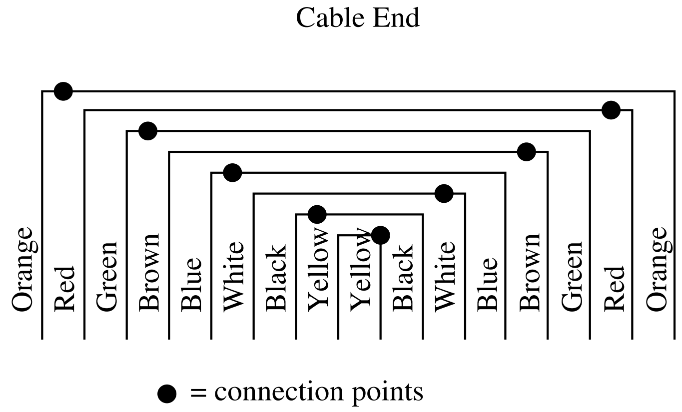
(d) Wash in clear water and let it dry.

5. Assemble upper mask on lower.

(a) Glue upper mask to the core using Silicone glue. Use plenty around edges on both surfaces. In the sensing area put only a thin film on the core. (Be careful not to get glue on the copper areas between the traces.) Align center of upper mask to lower mask using the alignment notch. Let the glue dry.

(b) Scrape off the resist on alternating upper traces near the corners. Looking at the front of the sensor with the cable end at the top, start with left side removing the resist and tinning carefully.

(c) Tin another 8 conductor cable and super glue the cable cover to the wire channel. Use the following color code.



(d) Create the shield by using a modified upper mask. Paint nail polish on the back and over connecting wires on the front including 1mm over the tops of the drive lines. This shorts all the electrodes together. Etch this as before. At this point the shield is very fragile as the only thing holding it together is the nail polish. Glue the shield over the drive lines with a small separation given by the silicone glue. CUT the nail polish between strips when the glue is dry.

(e) Make a copper circle for the end and glue it with epoxy or superglue. Connect it to the shield of upper (drive) cable. This will be grounded.

(f) Fill the wire channel AND the space between connecting wires up to sensing area with 5 minute epoxy. These connecting wires must be immobilized. Let the glue harden.

(g) Solder the cable to an 18 pin header with the following color code.

Pin	Sense Color	Pin	Driver Color
1	White	18	Yellow
2	Green	17	Black
3	Blue	16	White
4	Brown	15	Blue
5	Black	14	Brown
6	Red	13	Green
7		12	Red
8		11	Orange
9	Shield	10	Shield

(h) Test wiring. Check for shorts between all combinations of connector pins. Check that it works when plugged into the circuit. Check that all connections are good.

(i) Screw in a 1/4-20 shaft.

(j) Smear the finger mold with vaseline and warm it with a heat gun. Blow off the excess vaseline with compressed air. Place the finger in the mold. Tighten the end against the bottom of mold for axis alignment. Check that there is equal space on the sides and the front and back.

(k) Close up the mold. Tape down the reservoir ring with electrical tape. Seal the top with tape so that air must be replaced by rubber in the vacuum. Insert alignment pin!

(l) Mix 20g HSII base with 1/10 catalyst by weight. Smash, don't stir.

(m) Pour into the mold. Vacuum 2-3 minutes until large bubbles START to form. Remove. Fill reservoir again. Vacuum in cycles of 2-3 min on, 1 min off for 15 minutes. Don't let it "boil".

(n) Let it sit 1 day.

(o) Remove pin. Remove from mold, insert pin.

Appendix E

Schematics of electronics

On the following pages are the final schematics for the circuit used with the tactile sensors.

TIMING2T.SCH1 - Thu Oct 28 14:53:15 1993

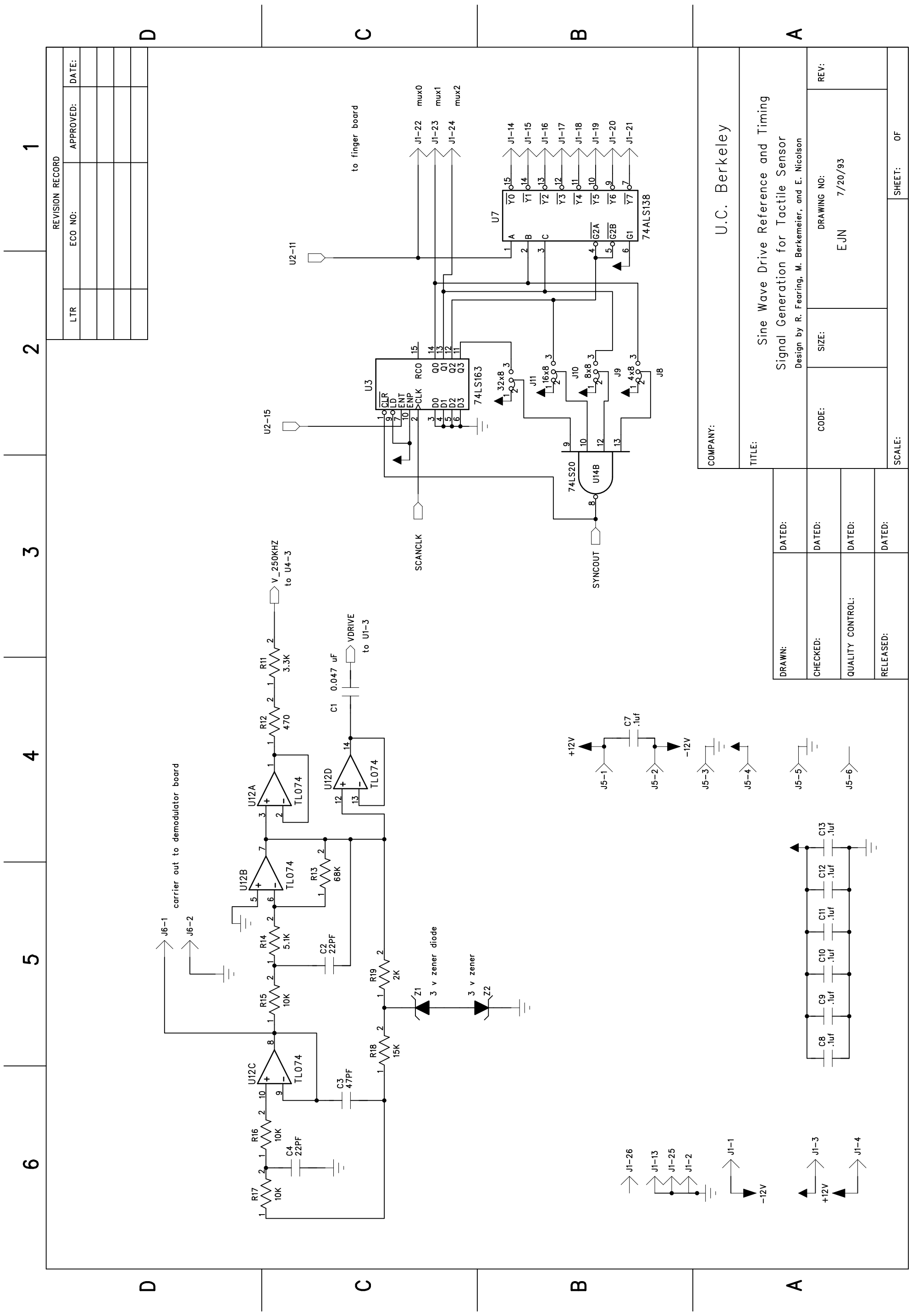


Figure E.1: *250 KHz sine wave and square wave oscillator.*

SCANCLK1.SCH1 - Thu Oct 28 14:31:05 1993

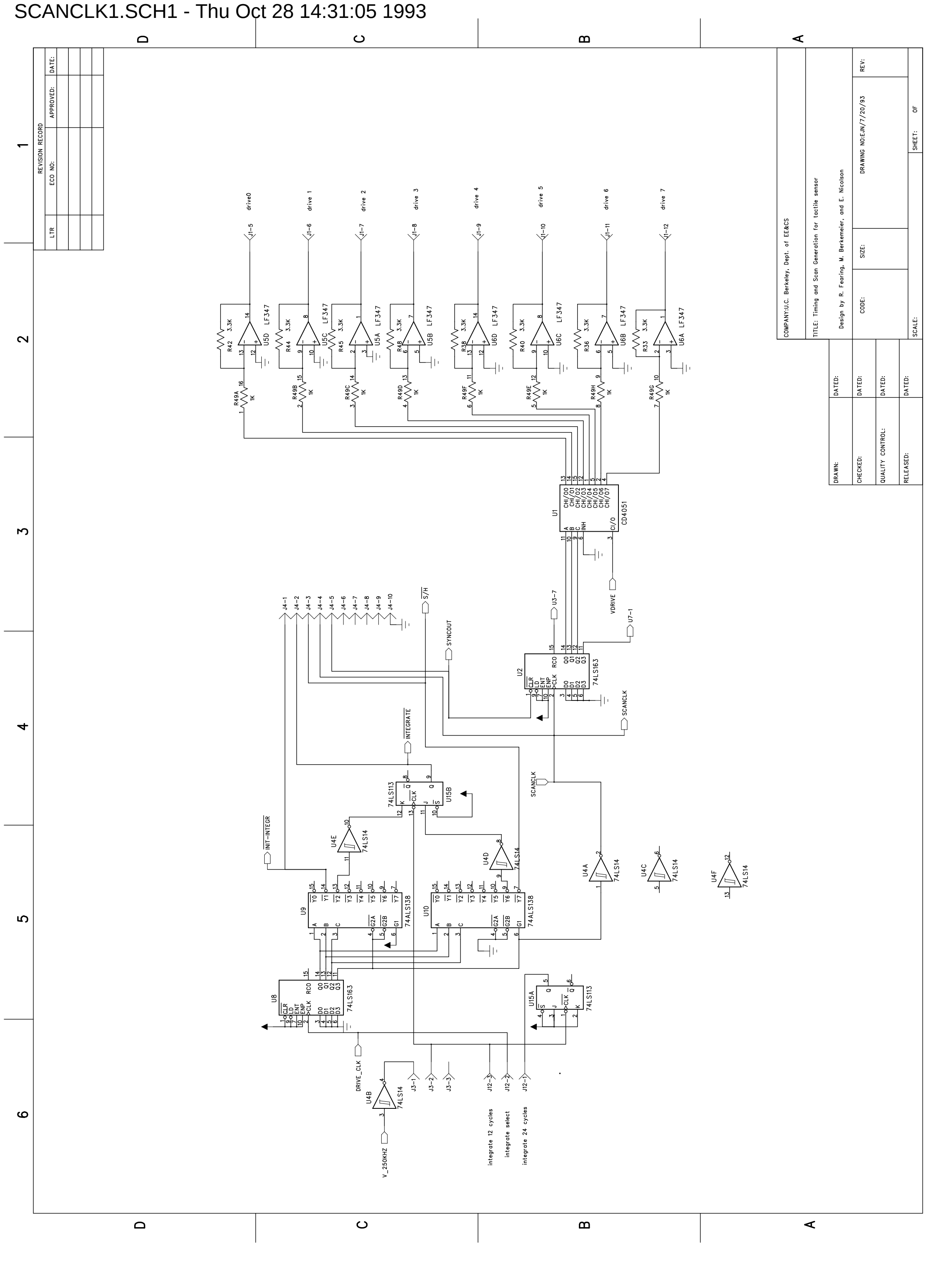


Figure E.2: *Timing signal generation.*

TACTCHRG.SCH1 - Thu Oct 28 13:46:53 1993

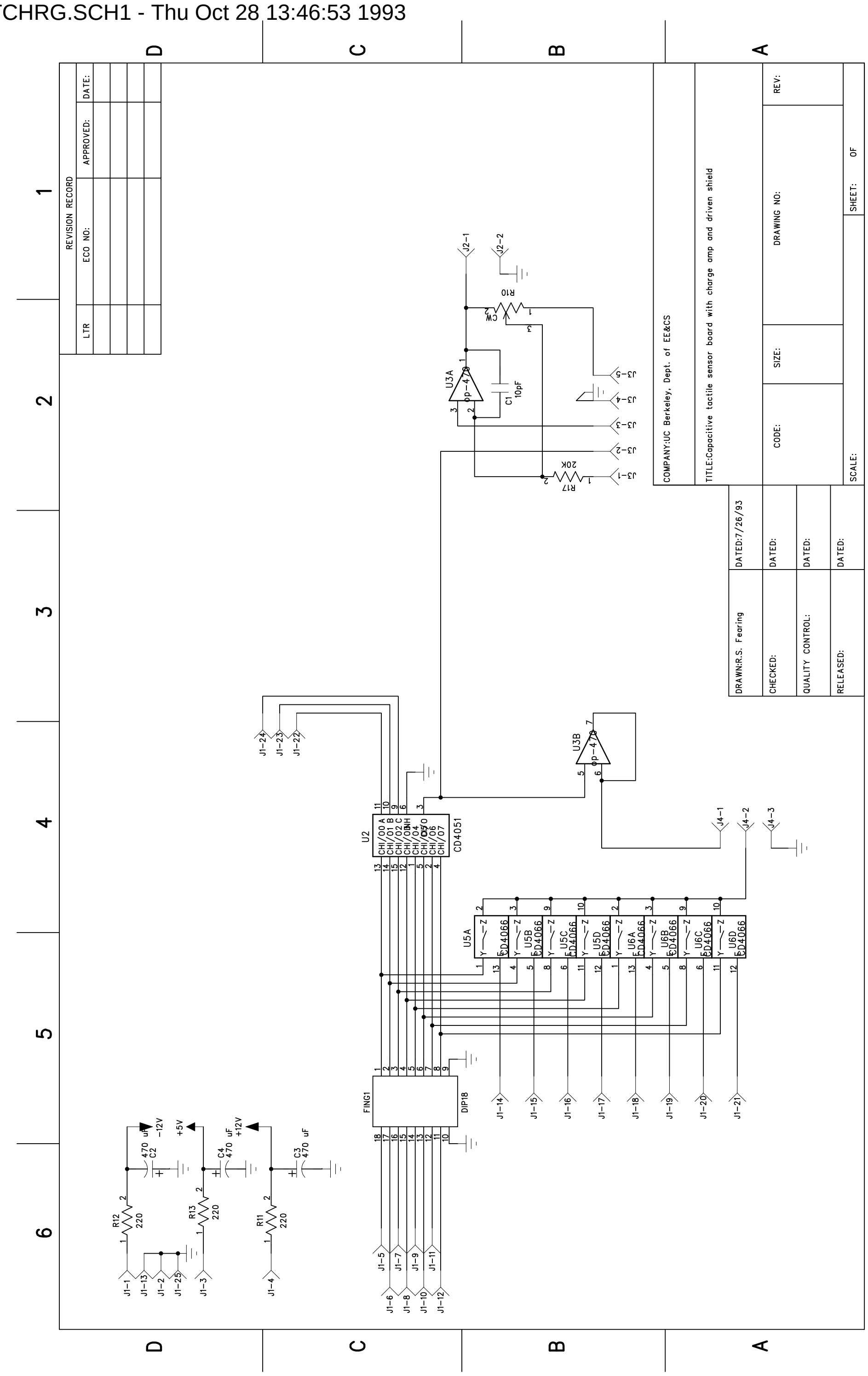


Figure E.3: *Charge amplifier for finger interface board.*

DEMODRF.SCH1 - Thu Oct 28 13:46:05 1993

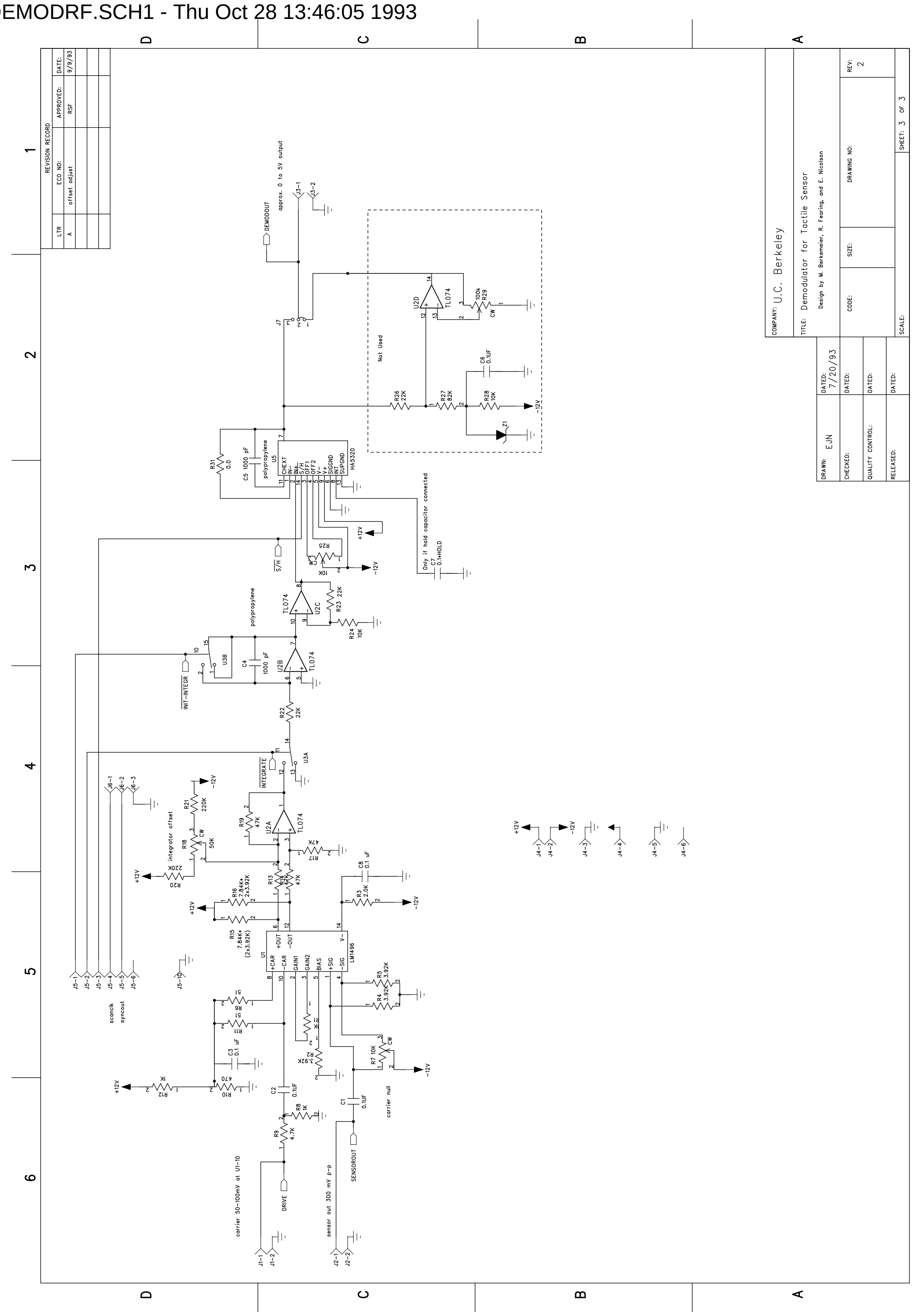


Figure E.4: *Demodulator, integrator, and sampler.*

Appendix F

SIOMS real time device driver specification

This appendix describes a solution to controlling and communicating with an automated workcell based on a customized commercially available real time operating system [1]. Two workcells, one for assembly and the other for machining, have been successfully controlled and integrated using much of the same software despite different applications, actuators, sensors, and computer hardware. Portability to different computer hardware arises from support by the operating system vendor while application portability arises from the development of this appendix : standardized input and output with mechatronic devices (SIOMS). The real time UNIX device drivers used to create the standard encapsulate complexities of I/O and present users with a simple interface which allows concurrent access to the devices from multiple processes. Recent advances in personal computer technology make this a cost effective solution.

F.1 Introduction

Suppose you are faced with the task of taking a collection of mechatronic components such as stepper motors, dc motors, binary and analog sensors and actuators and combining them in such a way as to automate an assembly or production sequence. The motion control algorithms are simple: acceleration, velocity, position, and force control on a per axis basis. It is important that the axes move synchronously. The usual systems

[1] This appendix originally appeared in (Nicolson, 1994) in a slightly different form

integration solution to this problem is to connect the several different electronic interfaces of the various controller boxes to a computer. Each interface will be different and will require knowledge of special hardware interface commands. Due to the low bandwidth of some interfaces, such as serial, exact motion synchronization is difficult. Finally, you wish to perform analysis of the system efficiency, but it is not possible to access to the state of the actuators and sensors from a separate monitoring process.

It has been pointed out by many developers of real-time systems that system development time would be greatly reduced with encapsulation and standardization of input and output specific code (Stewart et al., 1992; Williams, 1990). This appendix proposes a possible standard interface through device drivers. The challenge is to create an interface standard that does not require an inordinate amount of overhead. The device drivers discussed in this appendix allow efficient use of the processor with minimal overhead for interfacing to the devices at rates less than 100Hz.

The SIOMS driver solution was originally developed to address problems of maintenance, portability, concurrent access to sensors and effectors, control mode switches, and encapsulated input and output that existed in our previous workcell controller for a Sawyer motor small parts assembly system (Scheinman, 1987). The previous controller was based on a modification of NYMPH (Chen et al., 1986), a 68020 VME Bus based multiprocessor system. Experience with that system indicated that for many of the tasks in manufacturing, high performance and dynamic control is not as important as ease of integration of sensors and actuators to accomplish a task. Also we discovered that applications were not portable since they relied on specific hardware for I/O. We therefore decided to focus on providing the most portable solution at reasonable cost.

This appendix proceeds first by discussing the operating system requirements of mechatronic systems and the applicability of a real-time UNIX. We follow with a description of the SIOMS driver protocol and their performance. We conclude by discussing applications and possible extensions.

F.1.1 Mechatronics

Mechatronics is a field which originated in Japan as the study of the combination of mechanical and electronic systems (Fraser et al., 1993; Miu, 1993). A mechatronic device is a transducer or actuator that has an electronic interface. Examples of mechatronic devices

Mechatronic Device	Type
LED Beam Sensor	Binary
Force/Torque Sensor	Analog
Stepper Motor	Axis
DC Motor	Axis

Table F.1: *Mechatronic device types.*

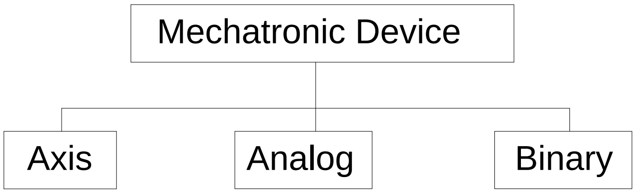


Figure F.1: *Device class hierarchy.*

include stepper motors, DC motors, light beam sensors, force and torque sensors, tactile sensors, tactile stimulators, and pneumatic grippers. These simple components are used in larger devices such as mills, lathes, conveyor belts, and robotic manipulators. We can classify the basic types of mechatronic devices as axis, binary, or analog. Examples are shown in Table F.1.

The classification of the devices reflects the information the mechatronic device provides or controls not the type of communication required, be it serial, parallel, or shared memory. Following the object oriented paradigm (OOP) (Booch, 1991), we seek an I/O solution that will encapsulate and hide complexities of communication and give devices a state and a set of methods to access them.

Conducting a robotic experiment or constructing a manufacturing workcell typically involves the computerized integration of several mechatronic devices. The integration requires the programming of time dependent tasks such as control loops and sensor event responses. The computational complexity of the tasks is low, however they must either occur at periodic intervals or respond with a minimum latency. This requires task scheduling on a multiprocessing operating system to be preemptive and prioritized. So as not to restrict communication with a device to one process at a time, the device I/O interface must allow

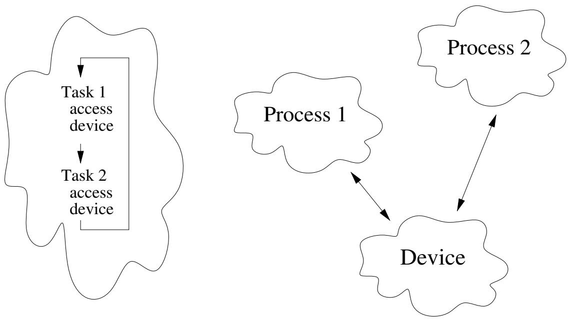


Figure F.2: *Advantages of a multithreaded architecture. A single threaded architecture requires all tasks to be coded together. With multithreading and device drivers tasks may be added incrementally.*

concurrent access by multiple processes.

F.1.2 Real time UNIX

Standard UNIX operating systems already provide the important features for portable open systems software development such as a graphical interface via X windows, networking via TCP/IP and NFS, and multiple processes and users. With the recent IEEE-1003.4 POSIX real time extensions a standard full-featured development environment for real-time applications is available (Williams, 1991).

Besides offering a standard way of scheduling and communicating with real time tasks, a POSIX 1003.4 compliant UNIX allows access to hardware via device drivers: pieces of code that are dynamically linked into the kernel of the operating system. The device driver mechanism is an ideal way to standardize I/O with mechatronic devices for the following reasons: 1) the I/O is functionally standardized through the use of UNIX system calls, 2) the operating system kernel provides robust features that automatically close open devices and return the I/O channel to an initial state when a user program that was accessing them exits or crashes unexpectedly, 3) I/O contention brought on by concurrent I/O requests by multiple processes is resolved with semaphores in the driver code, 4) more recent real-time UNIX operating systems allow system call interruption inside the kernel so that driver code becomes just an extension of the user code, without task-switching penalties, and 5) the drivers encapsulate and hide the special hardware knowledge required to communicate with mechatronic devices.

It is the kernel and the standardized I/O through device drivers that provide much of the power of UNIX. Many of the tools that make up the executable utility programs of UNIX rely on device drivers for terminal, disk, and network I/O. By adding mechatronic device drivers and additional tools we simply extend the operating system's capabilities.

Once created for a particular piece of hardware a device driver may be reused in other applications, or used simultaneously by multiple applications. These applications programs, if general enough, may be taken to another system with drivers following the same interface allowing two forms of portability, one in the creation of input and output code, the other in the creation of programs that use the drivers.

F.2 Using the SIOMS drivers

For the above reasons a set of device driver interfaces specifically designed for mechatronic devices was created and termed "SIOMS drivers". Access to the devices is achieved with the 6 UNIX system calls: `open()`, `close()`, `read()`, `write()`, `ioctl()`, and `select()`. `open()` and `close()` perform the functions of initiating and finishing access to a device. `read()` and `write()` allow input and output respectively, `ioctl()` is used usually to modify the I/O and `select()` is usually used to block, or halt, the calling process until an event.

F.2.1 The root class

We begin by discussing features common to all types of SIOMS drivers. In OOP terminology, this describes the root class. Next we go into the details of reading, writing, and special `ioctl()` options for each type.

open()

To get access to a device a process must first open a device node file. Such a node may be thought of as an instance of a device of a particular type. In the UNIX operating system a device node is a special type of file that usually resides in the `/dev` directory of the file system. Suppose there is a motor driver, then the first motor node, or "instance" of a axis devices, might typically have the file name `"motor0"` and the C language code

```
motor0fd = open("/dev/motor0",O_RDONLY,0);
```

would assign the integer motor0fd to be the file descriptor to be used in all subsequent access to the motor. When the `open()` call is executed the UNIX kernel determines which driver the device node file corresponds to and executes the `open()` call registered for that particular driver. It is possible to have multiple device node files corresponding to the same physical device.

Besides associating a file descriptor with a set of driver functions, the open call also allows the driver to resolve device contention. The second argument of the open call is the mode which can either be `O_RDONLY`, `O_WRONLY`, or `O_RDWR` corresponding to read only, write only, and read and write access. To ensure that two processes do not try to control the same output, SIOMS output devices (device nodes that may be written to) may only be opened by one process at a time *regardless* of the mode. If a duplicate request occurs the system error code `errno` is set to `EBUSY` and `open()` returns -1.

close()

The `close(fd)` call, where `fd` is the file descriptor to be closed, is used when the process is finished accessing the device. For the SIOMS axis driver this call will free up a particular axis to be opened by another process. As a safety feature the SIOMS axis driver ensures that all axes associated with an axis device node being closed are halted. This feature combined with the automatic closing of devices on process termination allows safe recovery from user program crashes.

Device type identification

Although a user will most likely already know the type of node they are opening, it is useful for utility programs to be able to get this information automatically. For this reason a special `ioctl()` call is supported by all devices to return the device type. The call:

```
struct io_struct_info {
  int struct_type;
  int read_only;
  int number_structs;
  int read_type;
  int write_type;
  int read_struct_size;
  int write_struct_size;
};
```

Device Node File	Type	Number	Description
tactile0	analog	48	tactile array sensor
gripper0	binary	1	pneumatic gripper
mod0.readonly	axis	4	read only node for a 4 axis manipulator

Table F.2: *Examples of device node files.*

```
struct io_struct_info sinfo;

.
.

ioctl(fd, GET_IO_STRUCT_INFO, &sinfo);
```

would fill in the fields of the `sinfo` structure for the opened device. The type of device, be it binary, axis, or analog, can be determined from the `read_type` and `write_type` fields. We also note that all device types support nodes which correspond to arrays of sensors, effectors, or axes. The `number_structs` field identifies the dimension of the array.

Node naming conventions

Though the name of a device node can be arbitrary, it is useful to use a systematic naming convention. Besides the usual convention of naming nodes using a root name and numbers, the extension `.readonly` is appended to the name of companion nodes to output nodes. For example, if `axis0` is an output node then it may be opened only once, regardless of the open mode. The node `axis0.readonly` may be opened multiple times and, when read, will provide the identical information as when reading from `axis0`.

Due to the device type identification support described above it is not necessary to follow this naming convention since a node can be identified as an input/output or input only from the `read_only` field of the `struct io_struct_info` structure. Table F.2 gives examples of device node files currently in use.

Reading and writing

The `read()` and `write()` calls are used in the following way:

```
read(fd, struct_size, struct_ptr);
write(fd, struct_size, struct_ptr);
```

where `struct_ptr` is a pointer to a structure used for access and `struct_size` is its size in bytes. The structures used for reading from and writing to the three different SIOMS driver types are listed in Table F.3. The calls return the number of bytes read or written and -1 if there was an error. The most common error is an invalid structure field entry.

F.2.2 The Binary class

The simplest type of driver is for binary sensors and effectors. A write to a binary device node file will set an output bit to the value specified by data in the `binary_write` structure. Similarly a read from a binary device node file will cause the state of an input bit to be written to the data field of the `binary_read` structure.

Binary nodes may be for a single bit or a whole byte. If an output bit node is opened then the corresponding output byte node is marked in use. Similarly if an output byte node is opened, then all corresponding output bit nodes are marked in use.

It is often necessary to have a thread block on a binary node until its value changes. Support for such binary events is provided with the `ioctl()` call:

```
struct event_descriptor {
  int struct_type;
  long mask;
  long initial_event;
  long final_event;
} evt;

.
.
.

retval = ioctl(fd, WAIT_UNTIL_EVENT, &evt);
```

`mask` is used for multiple bit nodes to indicate which bits to watch. `initial_event` indicates the state when waiting began, `final_event` indicates the final state. The `ioctl()` will return when one or more of the bits has changed or when an error occurs.

Possible errors are: `EFAULT` for a bad pointer, `EINTR` if the thread received a software interrupt. Currently the wait can only be terminated by an event or by sending

	Reading	Writing

Binary	`struct binary_read{`
	`   int struct_type;`
	`   int data;`
	`};`
		`struct binary_write {`
		`   int struct_type;`
		`   int data;`
		`};`

Analog	`struct analog_read{`
	`   int struct_type;`
	`   double data;`
	`};`
		`struct analog_write {`
		`   int struct_type;`
		`   double data;`
		`};`

Axis	`struct axis_read{`
	`   int struct_type;`
	`   int axis;`
	`   int in_motion;`
	`   double position;`
	`   double velocity;`
	`   double k,kz,kp;`
	`   double kb;`
	`   double force;`
	`};`
		`struct axis_write {`
		`   int struct_type;`
		`   int axis;`
		`   int control_mode;`
		`   int signal_marker_number;`
		`   int wait_marker_number;`
		`   double position;`
		`   double velocity;`
		`   double acceleration;`
		`   double jerk;`
		`   double max_velocity;`
		`   double double max_acceleration;`
		`   double time;`
		`   double k, kz, kp, kb;`
		`   double force_offset;`
		`   double force;`
		`   double force_velocity;`
		`   double max_force;`
		`};`

Table F.3: *Structures used to read from and write to the drivers.*

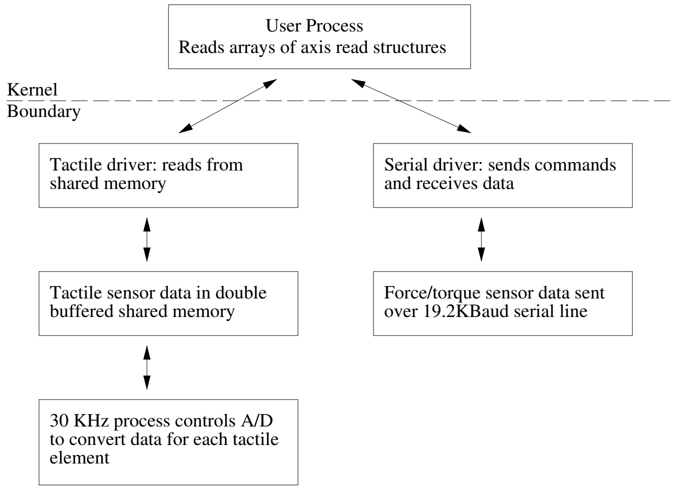


Figure F.3: *Hiding handshaking details inside analog mechatronic device drivers.*

the thread a signal. Perhaps this would be better implemented as a `select` call to allow the specification of a time out.

F.2.3 The Analog class

Use of an analog device node file is functionally identical to a binary device node file. Output values are put in the data field of an `analog_write` structure and input values are read into `analog_read` structures. For the data to be correctly interpreted the system of units used in the driver must be known.

Though just as simple an interface as the binary device type, typically an analog device driver is more complicated internally. Consider for example two analog mechatronic devices, a tactile array sensor and a force/torque sensor. As shown in Figure F.3, the tactile sensor data is collected by a 68020 processor and stored in shared memory on the VME bus whereas the force/torque data is retrieved over a serial line. The type of data, double precision numbers, is identical, yet the methods of retrieval is different. Fortunately the user is completely unaware of these complexities since they must only deal with the `analog_read` structures.

Control Mode Flags	Description
JERK, ACCELERATION, VELOCITY, POSITION, FORCE, or FORCE_VELOCITY	Command the specified quantity for the axis.
SPECIFIED_TIME	The time field contains the duration for the command in seconds. If time <0 then the command is executed until a new command is received.
WAIT_FOR_EXTERNAL_SYNCH	The command will be stored in the buffer but execution will be delayed until a synchronization signal is sent via an ioctl() call
SPECIFIED_GAINS	Set the control gains as specified the fields.
SPECIFIED_FORCE_OFFSET	Set the force offset in the axis controller.
SPECIFIED_MAX_VEL_ACC	Set the current maximum acceleration and velocity.
SPECIFIED_MAX_FORCE	Set the maximum output force.
SPECIFIED_WAIT_MARKER	Wait for wait_marker_number before executing the next command.
SPECIFIED_SIGNAL_MARKER	Signal signal_marker_number when this command is done.

Table F.4: *Control mode options. Depending on the axis type, combinations of these options may be specified at the same time.*

F.2.4 The Axis class

The most complex of the drivers is that for an axis. We would like to communicate with DC servo, stepper and other motor controllers in the same way. Providing this rich set of features necessitates the complexity of the `axis_read` and `axis_write` structures shown in Table F.3.

When a `read()` is executed on an axis device node file, the fields of the `axis_read` structure are filled in with the current position, velocity, controller gains, and force applied by the actuator. The `in_motion` flag indicates if the axis is currently moving in position or force mode or if it is being commanded in any other mode.

The fields of the `axis_write` structure are self explanatory except for `control_mode`.

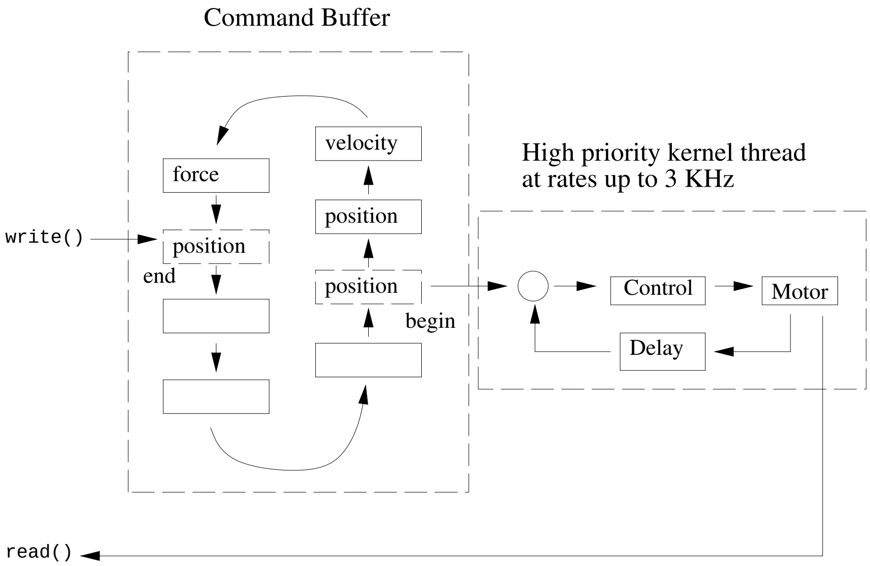


Figure F.4: *Axis driver architecture.*

The `control_mode` is set to the logical or of the flags shown in Table F.4. Of course not all combinations of the flags are supported and the support will vary depending on the type of actuator.

Internally the axis driver is implemented as a high priority kernel thread and a command buffer. Writes to an axis device node file result in commands being appended to the buffer. If an array of structures is written to an axis device the commands will not be executed until the whole array is read into the buffer. In this manner complex trajectories may be specified. As the command blocks expire the controller moves onto the next command and handles any special requirements for switching between control modes. Figure F.4 illustrates command buffering.

Except in force mode, DC motors compute a desired position and velocity based on the current command. The motor torque is given by a PID law with velocity damping and torque offset:

$$\hat{\tau}[n] \quad = \quad k(e[n] + k_z e[n-1]) - k_p \tau[n-1] + k_b v[n] + \tau_o$$

ioctl command	Description
`GET_IO_STRUCT_INFO`	As described in section F.2.1.
`CALIBRATE_DEVICE`	Set position to the `double` given by `*arg`.
`ZERO_DEVICE`	Set the current position to 0.
`WAIT_UNTIL_IDLE`	Return when motion is finished.
`SEND_EXTERNAL_SYNCH`	Commence pending motions commands
`EMERGENCY_STOP`	Stop motion. In force mode, set output to 0.
`SIGNAL_MARKER`	Set marker number given by `*arg`.
`WAIT_MARKER`	Wait for marker number given by `*arg`.
`RESERVE_MARKER`	On return `*arg` will contain a free marker number.
`FREE_MARKER`	Free up the marker number given by `*arg`.

Table F.5: *Axis ioctl options.*

$$\tau[n] = \begin{cases} \tau_m & \hat{\tau}[n] > \tau_m \\ \hat{\tau}[n] & -\tau_m < \hat{\tau} < \tau_m \\ -\tau_m & \hat{\tau}[n] < -\tau_m \end{cases}$$

where $e[n]$ is the error between the actual and desired position, $v[n]$ is filtered actual velocity, τ_o is the torque offset, τ_m is the maximum torque, and $\tau[n]$ is the output torque. In force mode the output is simply the desired torque plus the velocity damping and offset torque. For stepper motors a desired position and velocity is computed and the output position slewed using a PID law on the error. The accelerations and velocities are clipped according to the motor limits. In addition to reading from and writing to an axis there are additional commands available through `ioctl()` for calibration, synchronization, and emergency stopping, as shown in Table F.5.

F.3 Driver performance

The important performance specifications to keep in mind when considering the SIOMS driver solution are operating system overhead and the number of axes that can be controlled simultaneously on one processor. These specifications will depend on the processor and bus used. To evaluate the system call overhead a simple program computed the average time to write a byte to the device /dev/null. Task switching overhead was

Processor / Bus	Task switching	Write to device
68030 / VME	130 us	86 us
68040 / VME	45 us	20 us
486DX266 / ISA	18 us	12 us

Table F.6: *Operating system overhead.*

	68040 25MHz VME	486DX2 66MHz, ISA
OS scheduling	50us	20us
Typical variable latency	50us	20us
Additional peak latency	50us	80us
Bus I/O, 12 axes	70us	26us
Control, 12 axes	125us	90us

Table F.7: *Breakdown of time from entry into the interrupt service routine until completion of the control update.*

computed by creating two processes and requesting them to yield alternately. The results for the 3 different processors used to date is shown in Table F.6. For access to the drivers at rates less than 100Hz the 68040 and 486 provide insignificant overhead in task switching and system calls.

To quantify the capability of the operating system to control multiple axes 4 events were timed in the control loop: 1) entry in to the interrupt service routine, 2) entry into the high priority controller thread, 3) completion of bus I/O, and 4) completion of control computation. The time from (1) to (2) is often termed the scheduling latency. Since scheduling latency depends on the total system load, it was monitored under two conditions, first while only the driver was running and second while a `tar xvf - / | rsh` process was running. Under the full system loading of the second case the scheduling latency increased by up to 100 microseconds. The results are shown in Table F.7. Based on these times one can estimate the maximum number of axes that can be controlled by a single processor at a given update rate, as is shown in Table F.8.

Update Rate	68040 25MHz VME	486DX2 66MHz, ISA
5 KHz	1 axis	4 axes
3 KHz	10 axes	19 axes
2 KHz	20 axes	35 axes
1 KHz	52 axes	90 axes

Table F.8: *Maximum number of axes of control as a function of update rate.*

F.4 Utility programs

The power of standardized drivers is not fully realized without a general set of tools to manipulate them. A set of utility programs have been written to provide a command line interface to the drivers. The `devin` program reads from a list of devices at a specified rate and prints the current state in ASCII format to the standard output. The `devout` program reads ASCII numbers from standard input and writes them to a list of devices. The `sdev` program gives a command line interface to the `ioctl()` options. In addition, synchronization points can be set in trajectories using the `smarkers` utility.

F.4.1 devin

```
usage: devin [-l#h#n#pvfmNVrt#.#L#] dev1 [-l#h#n#pvfmNV] [dev2]

reads from dev1, dev2 and output to standard out. In its simplest
form devin is used as:

devin mod0

will produce the output:

0       0       0.0000e+00      0.0000e+00      2.56e+02
1       0       0.0000e+00      0.0000e+00      2.56e+02
2       0       1.9920e+00      0.0000e+00      0.00e+00
3       0       -1.1096e-04     0.0000e+00      0.00e+00

which gives the axis number, the is_moving flag, the position,
velocity, and current motor force.  Multiple devices may be
read as in:

devin mod0 gripper0 atod0

produces the output:

0       0       0.0000e+00      0.0000e+00      2.56e+02
1       0       0.0000e+00      0.0000e+00      2.56e+02
2       0       1.9915e+00      0.0000e+00      0.00e+00
3       0       -1.1096e-04     0.0000e+00      0.00e+00
0       0
0       -3.59e+00

devin first tries to open the device file in the directory
~rws/dev.  If that fails it looks in the current directory
or the full specified path.  (i.e. devin ~rws/dev/mod0 also
works,) If you want to rename the devices, you can make
soft links from your current directory to the dev directory
as in "ln ~rws/dev/mod0 m0" and then do "devin m0".

As you can see devin is capable of determining the type of
device node and formats the output accordingly.  There
are numerous options to devin to allow different
output formats.  The flags are described below.

-V      verbose output, overrides the p, v, f, and m flags.

-p      print position in floating point form for axis devices.

-v      print velocity in floating point form for axes devices,

-f      print motor force for axis devices in floating point form

-m      print is moving flag for axis devices in integer format

-N      for each axis, or analog or binary input, for a given device
        precede the data with the structure number and terminate
        with a newline.
```

```
        The default output is equivalent to -pvfmN.  If any of
        p,v,f,m, or N are specified then only the specified items are
        printed.  For analog or binary nodes the p flag is used
        to print the data.

        Example:

        talisker nicolson (8) devin -p atod0
        -3.60e+00       talisker nicolson (9)

        Notice that without -N no newlines are printed.

To control which axes, or inputs, are printed for multi-device nodes:

-l#     Start printing from device #.
-h#     Stop printing at device #.
-n#     Print only device #

        Examples:

        talisker nicolson (10) devin -n0pN mod0
        0       0.0000e+00
        talisker nicolson (11) devin -l1h3v mod0
        0.0000e+00      0.0000e+00      0.0000e+00      talisker nicolson (12)

To cause timed output:  all device data is printed, ignoring the -N flag
        set for each device, as a stream of numbers.  If -N is anywhere
        on the command line, newlines and sample number are printed.

-r      Repeat printing at 1 second intervals.
-t#.##  Repeat printing at #.## second intervals.
-L#     Print # samples, default 100.  -1 causes infinite repeat
-N      Anywhere in the command line causes the sample time to be
        printed at the beginning of the line and newlines to be printed.


Authors: E. Nicolson, M. Singh

FILES:
   ~rws/src/devutil.c
   ~rws/bin/devin
   ~rws/doc/devin.doc
```

F.4.2 devout

```
Utility to write  a stream of floating point numbers in ASCII format
to a list of SIOMS devices.


devout [-l#h#n#rpvfajtsm] device ... [#.##  #.## ] < file

where file consists only of floating point ASCII numbers.
Obviously this can also be used as a pipe as in the following
command line control loop.

Example: damper control of module 0 x axis based on lord1 x axis
with a 50 ms sample time, runs 500 loops = 25 seconds

devin -t0.05 -n0 -L500 lord1 | devout -rv -L-1 -W mod0.xaxis


Flags affecting all listed devices:

   r    repeat
   L#   repeat # times (default 100), -1 means until the end of file

   w     wait for axis devices at end of each iteration of scanning
         the input and writing to the devices.

   W     don't wait for axis devices at the end of standard input

Flags set for each device:

   l#   start writing at # structure of device
   h#   stop writing at # structure of device
   n#   write only # structure

   For axis or binary devices the number of floating point numbers
   expected is equal to the number of structures specified.
   For axis devices the following flags are used to indicate
   what the numbers in the input stream correspond to.

   The default of the following possible control modes is p, for position
   p    position
   v    velocity
   a    acceleration
   j    jerk
   f    force
   F    force velocity
   o    force_offset

   t    times are specified, default -1 (until next command is received)

   s    marker number to signal at the completion of the command
   m    marker number to wait on before executing the command

   Regardless of the order of the flags on the command line, the numbers
   are always read in the following order if the corresponding option
   is specified:
```

```
position, velocity, acceleration, jerk, force, force_velocity,
force_offset, time, wait_marker_number, signal_marker_number
```

Numbers are read from the command line followed by standard input until enough numbers are read for the devices specified. After all the numbers are read, the appropriate data is written to the devices. If -r is specified then this is repeated for the number of times specified by the -L option.

FILES

~rws/include/device_io.h
~rws/src/devutil/devout.c
~rws/bin/devout
~rws/doc/devout.doc

Authors

Ed Nicolson

F.4.3 sdev

```
SDEV

  utility program to set ioctl options for a SIOMS device

    usage: sdev [-zsewcg] [##.## .. #.## ] device [-zsewcg] device < file

   -z   zero device
   -s   send external synch
   -e   emergency stop
   -w   wait until idle
   -c   calibrate: set axis position to position given by ##.##'s
   -g   set tactile gain matrix to #.##'s

   the command line is scanned for options, args starting with '-', device
   names, args starting with an alpha, and #'s, args starting with a number.
   Numbers are stored into a buffer.

   Options requiring floating point numbers use the number buffer,
   if not enough numbers are on the command line, sdev attempts to read #s from
   stdin.

FILES
   ~rws/include/device_io.h
   ~rws/src/devutil/sdev.c
   ~rws/bin/sdev
   ~rws/doc/sdev.doc

Authors
   Ed Nicolson
```

F.4.4 smarkers

```
SMARKERS

 smarkers is  utility program to reserve, free, signal, and wait for markers.

 usage: smarkers [-fsw] marker [marker] [marker]
        or smarkers -r  [nummarkers]

   -r   reserve marker(s)
   -f   free marker(s)
   -s   signal marker(s)
   -w   wait for marker(s)

  Markers are used to synchronize tasks with trajectories.  The smarkers
  utility provides a command line interface to the ioctl() marker calls
  to reserve, free, signal, and wait for markers.

  When markers are initialized any waits on that marker using smarkers -w marker
  will block until the marker is signaled using smarkers -s marker.
  Subsequent calls to wait on the signaled marker will return immediately.
  Markers are intended only to be signaled once as subsequent signals
  will have no effect.

  Typically a marker is used to indicate that an event has  occurred.
  They can be used by multiple threads and processes.  They can also
  be signaled by an axis trajectory to indicate that point in the
  trajectory has been reached.  A trajectory can also pause until
  a marker is signaled.  For more details on this see "devout".

  smarkers uses the following four ioctl calls on the file descriptor fd
  returned from opening ~rws/dev/markers as read only.

        #include <device_io.h>

        int marker;
        int fd;

        fd = open("/home/lynx1/tools/rws/dev/markers", O_RDONLY,0);

        retval = ioctl(fd, RESERVE_MARKER, &marker);
        retval = ioctl(fd, FREE_MARKER, &marker);
        retval = ioctl(fd, WAIT_MARKER, &marker);
        retval = ioctl(fd, SIGNAL_MARKER, &marker);


Return Values:

  0: If successful
 -1: Error:
    System error number, errno
      EINVAL:   invalid marker number
      EBUSY:    no markers available

PROBLEMS
  Currently there is no ownership of markers so any process can free
```

```
  a marker if it knows the marker number.  In fact a process could
  just free all the markers and wreak havoc on a carefully planned
  motion.  It might be good to add an encryption key to restrict access
  to freeing and signaling the markers.


FILES:
   ~rws/include/device_io.h
   ~rws/src/devutil/smarkers.c
   ~rws/bin/smarkers
   ~rws/doc/smarkers.doc

Authors:
   Ed Nicolson, Aaron Wallack
```

F.5 Applications

SIOMS compatible drivers have been written for LynxOS (LynxOS,) on a Motorola 68040 VME based system and an Intel 486 ISA bus system. The first set of drivers control a small parts assembly workcell, the second set controls a small machine tool and part handling system.

F.5.1 Small parts assembly

The small parts assembly system consists of multiple 2 axis Sawyer (linear stepper) motors which move in the XY plane of a 0.8m x 1.2m platen. DC motors are attached to move in the z and θ_z directions. A total of 16 axes of motion are controlled at 1.8KHz. A variety of sensors and effectors, some of which are shown in Figure F.5, are used for different robotic experiments. The following applications have all been programmed using different combinations of SIOMS drivers.

- Compliant control of screw threading: A force torque sensor and a module were used in a 25Hz servo loop to correct for positioning errors in screw insertion (Nicolson and Fearing, 1993).
- Localization and recognition of objects using cross beam sensing: Manipulator position was correlated with the breaking of 3 LED beams with 500us maximum latency (Wallack et al., 1993).
- Derailleur assembly: SPARA, a LISP based world model controlled the assembly of a bicycle derailleur through TCP/IP. item Peg-in hole: Using one 4 axis module to locate holes and another to handle pegs, multiple peg insertions were efficiently carried out (Paulos and Canny, 1994).
- Tele-taction experiments: A tactile sensor and a tactile display attached to different modules gave the sense of remote touch.
- Planar hand: A planar two fingered hand was controlled by grouping two 4 axis manipulators in synchronized motion.
- Pallet handling: Beam sensors on the conveyor belt alerted the modules to pick up arriving pallets.

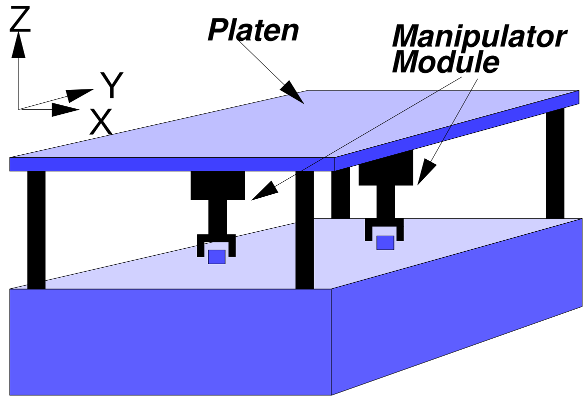


Figure F.5: *Small parts assembly workcell.*

F.5.2 Machining system

The second SIOMS system was created to replace serial line controllers for a machining workcell. A 486 PC controls the 24 axes of 3 mills, 3 lathes, a conveyor and a 6 axis arm. In addition limit switches and other binary sensors and effectors are used to monitor and control operations.

F.6 Interpreters and programming interfaces

Given the device drivers as specified three different approaches have been used for programming of robot motions. The first, and simplest, is the piping together of compiled C programs, such as "devin" and "devout", in shell scripts. The TCL/TK scripting and X interface toolkit can be used to start direct input and output amongst shell processes. The simplest application involves tying the output of a TK slider bar to the input of a "devout" process so that the motion of the robot axis is controlled by the slider bar. More complicated scripts, such as for calibration of tactile sensors, have also been written in TCL/TK.

The second programming approach uses a Lisp front-end with a world model database to coordinate pick and place operations. Lisp programs communicate through the network using TCP/IP to a multi-threaded server running on the real-time UNIX. The server interprets commands from the Lisp process and reads and writes to the SIOMS drivers as necessary. Additional threads are started when required for tasks such as sensor polling and sensor guided insertion. The Lisp environment combined with the drivers provided a powerful high level interface for robot programming.

The third approach, which is perhaps the most visual, uses a customized version of the NeXT Interface Builder program and the same multi-threaded server as was used for the Lisp front-end. A custom palette of robot "proxy" objects is linked into the Interface Builder program so that a application may be programmed by dragging, dropping, and connecting objects together. The use of Interface Builder for programming can be demonstrated in a few simple steps.

In Figure F.6 the initial work area is shown. The top left window contains the object palette, the bottom left window shows the instantiated objects that will not be visible during the running of the program, and the top right window shows the instantiated objects that will be visible during execution, such as buttons and sliders.

The programmer starts by dragging, and hence instantiating, a connection object, symbolized by a robot arm superimposed on a globe. This is shown in Figure F.7. The user then instantiates a robot module object as is shown in Figure F.8. Attributes of both of these objects, such as the machine to connect to and the robot number to use can be specified in the inspector panel. The inspector panel can also be used to connect objects together. In this case we wish to indicate which connection object the robot module object should use. In Figure F.9 a connection has been dragged from the connection "outlet", or field if you like, of the robot object to the connection object. In C programming terms this is like assigning the connection field of the robot object to point to the previously instantiated connection object. Now the robot module object can use the resources of the connection object to send commands to the real-time server.

Next we would like to create a button that will cause the robot to stop. In Figure F.10 a button has been dragged from the standard buttons and widgets palette to the user interface window in the top right. Again using the connection inspector we can specify the target of the new button to be the emergency stop method of the robot module, as is shown in Figure F.11. At this point we have created a simple interface that allows the user to send emergency stop commands to the robot. A more complex interface can be built up by continuing this process of instantiating and linking objects. Figure F.12 shows a program for a pick and place operation. The top left button connects and disconnects the interface from the robot server. The top left procedure button calls the array of buttons in the lower left. The method displayed on each button is then called. The

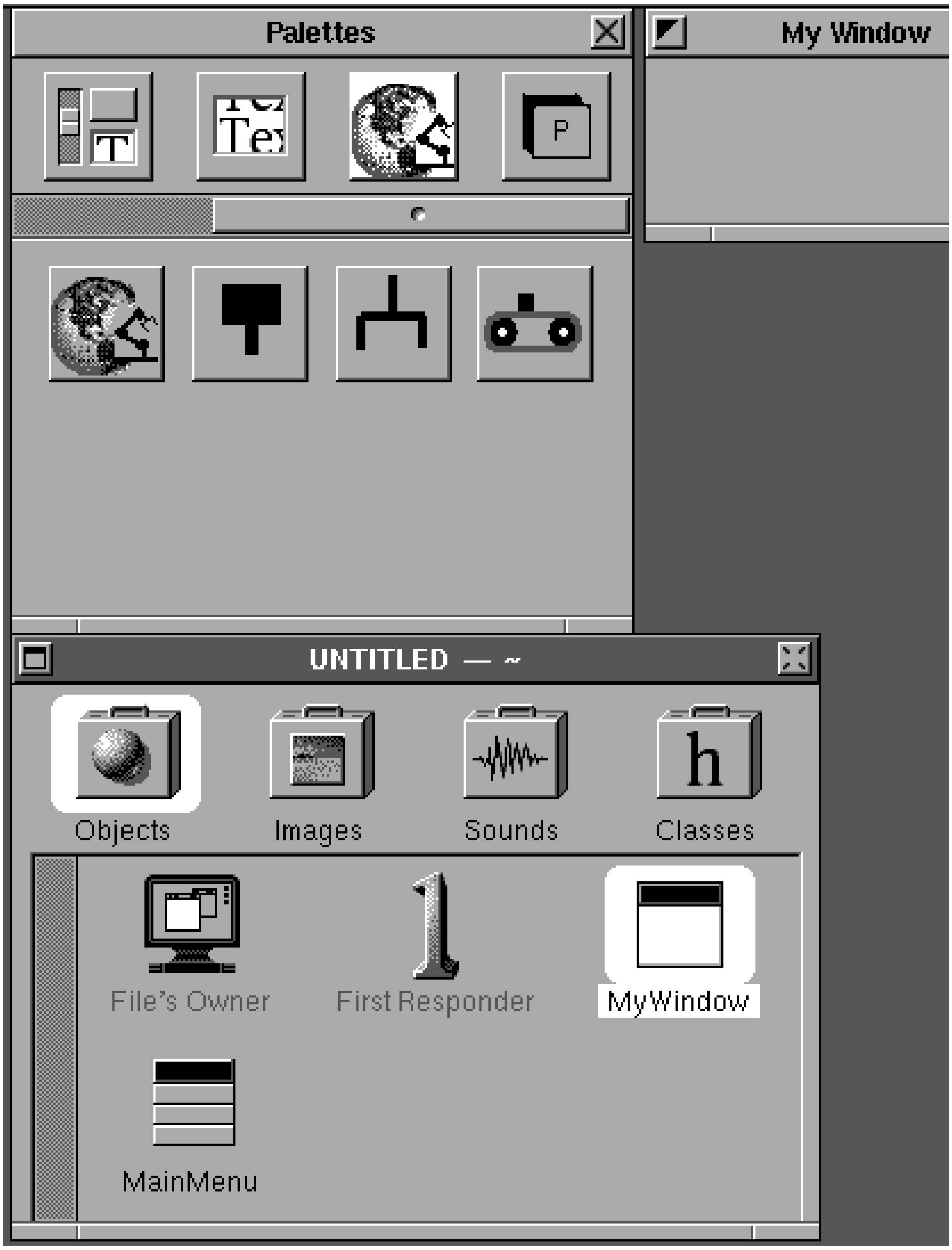


Figure F.6: *Initial work area for Interface Builder programming.*

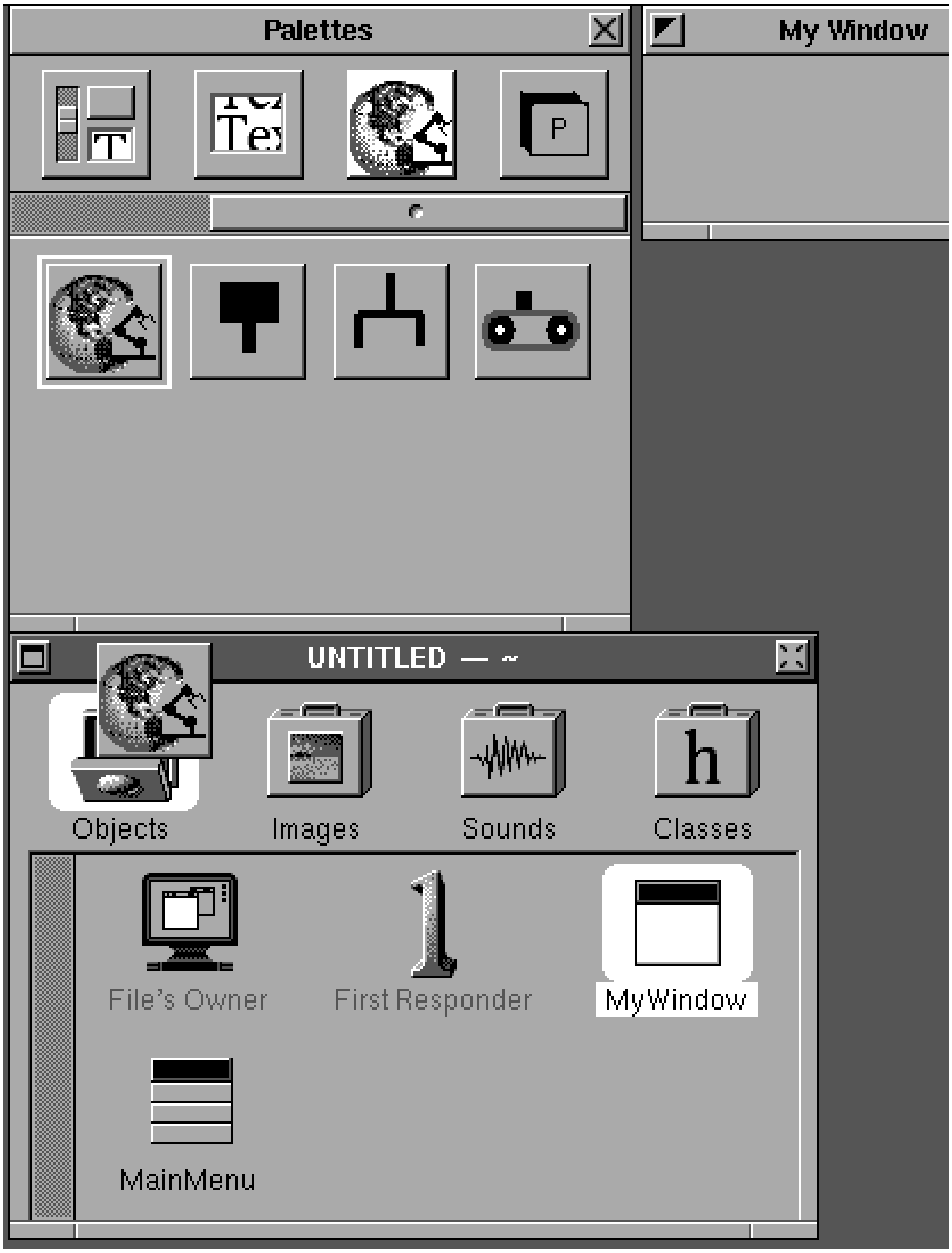


Figure F.7: *Instantiating a connection object.*

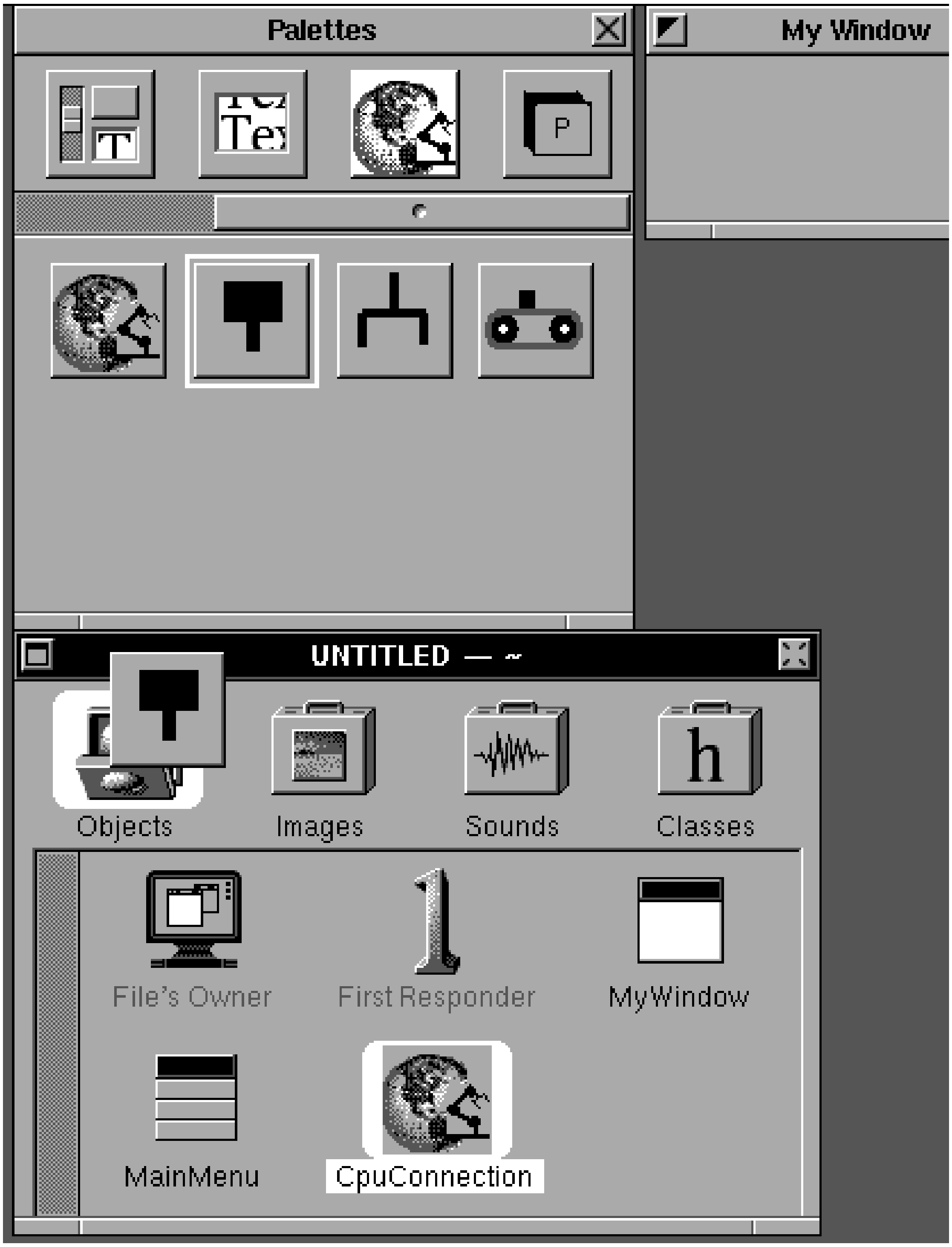


Figure F.8: *Instantiating a robot object.*

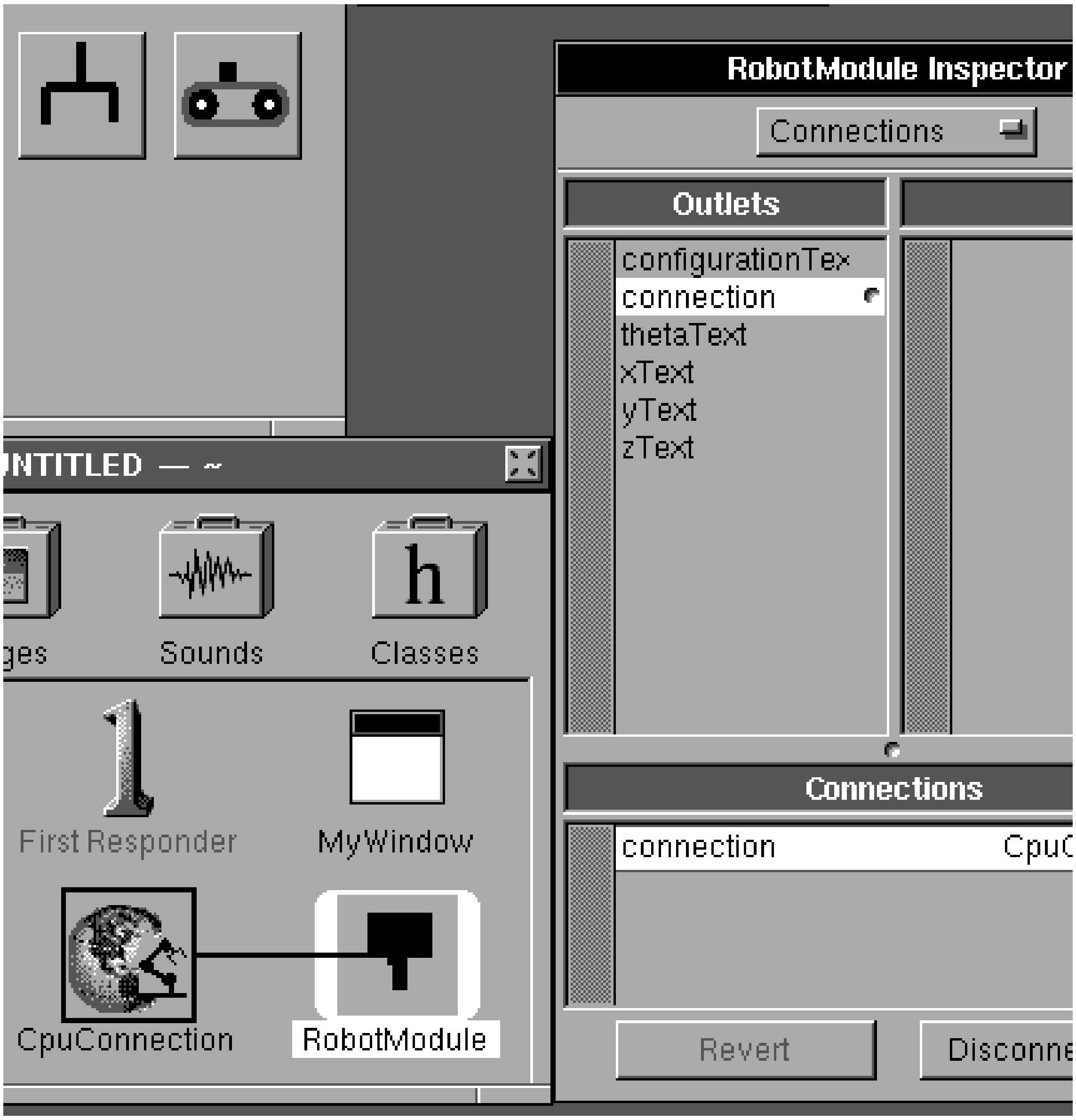


Figure F.9: *Linking the connection object to the robot object.*

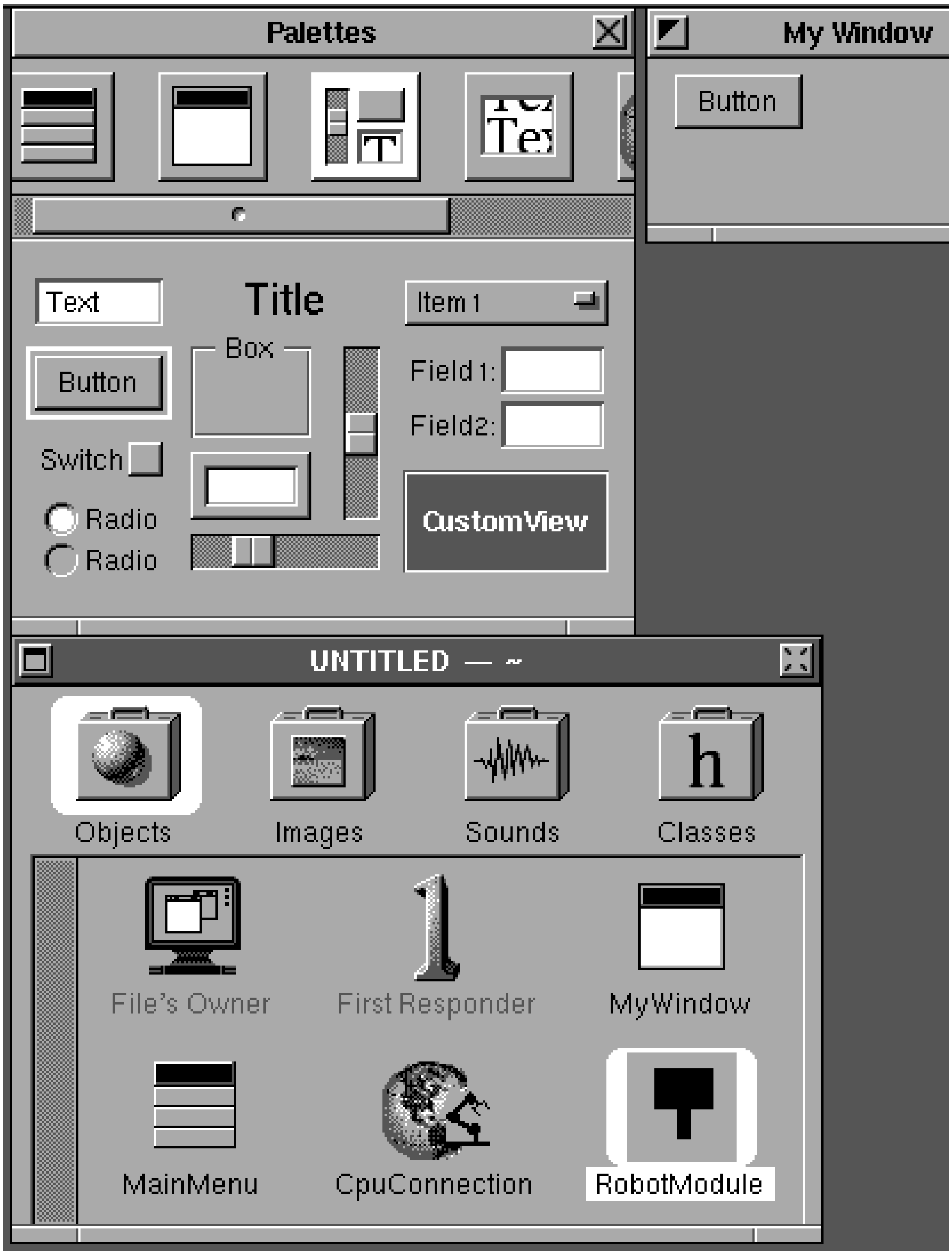


Figure F.10: *Instantiating a user interface button.*

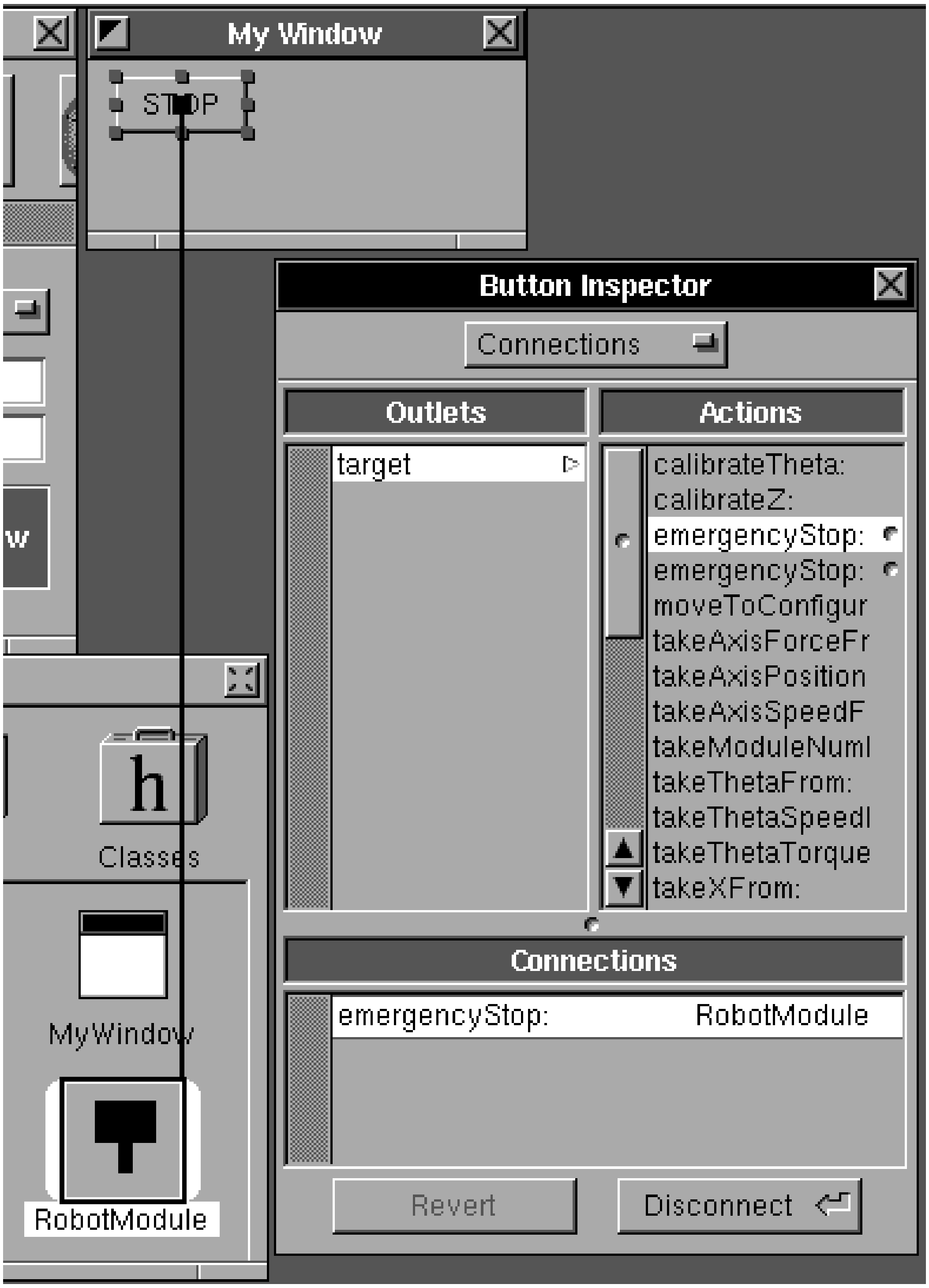


Figure F.11: *Connecting the button to the stop method for the robot.*

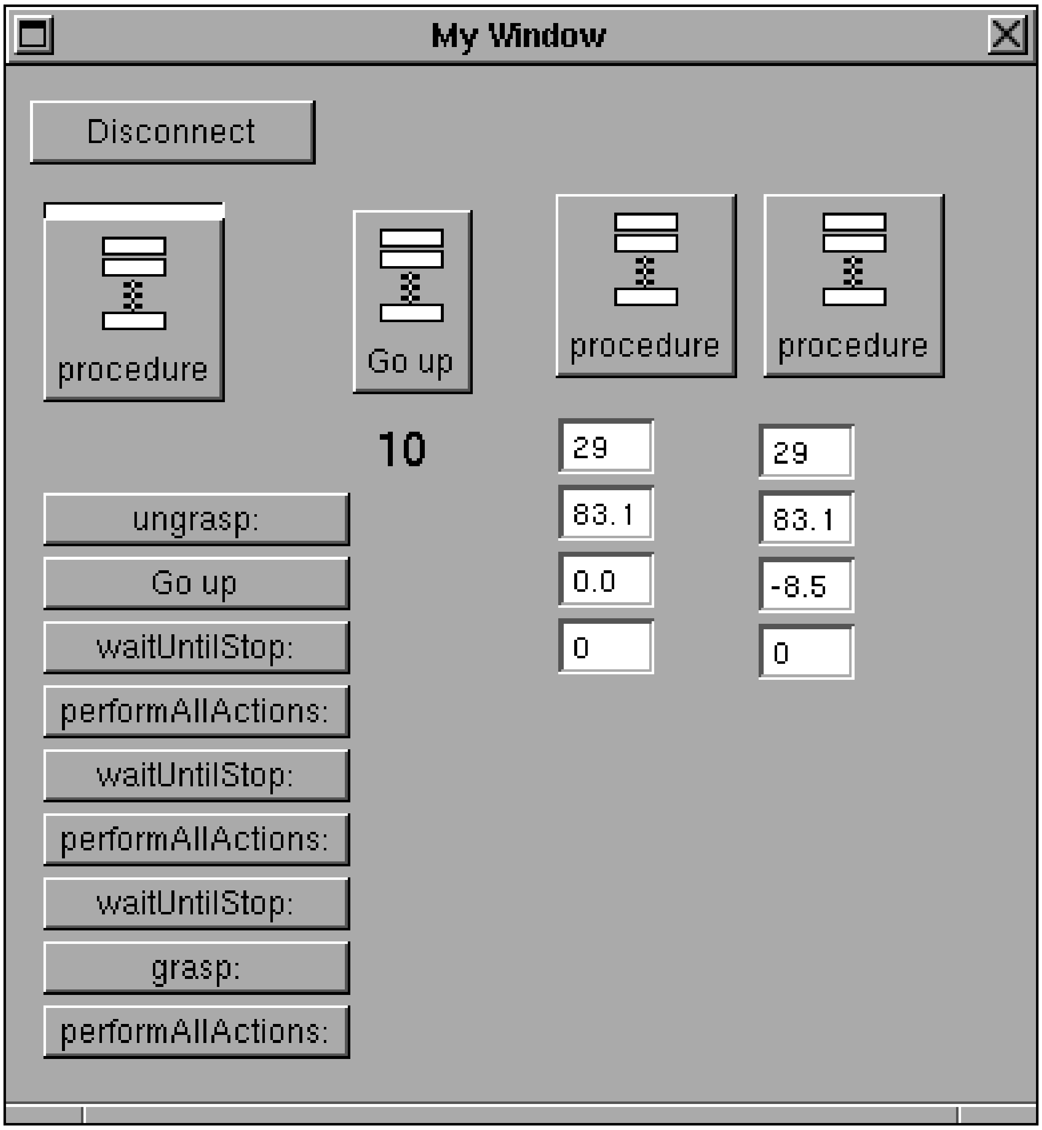


Figure F.12: *A final program made up of robot object methods.*

"performAllActions" methods call other procedure buttons causing other arrays of methods to be executed.

The NeXT Interface Builder provides a visual programming environment that was originally designed for building interfaces. By adding custom palettes we have created a visual robot programming environment. For users accustomed to the object oriented paradigm this should allow quick programming of robot motions.

F.7 Summary

The SIOMS driver specification has proved to be a useful and time saving development for programming of real time tasks in robotics and manufacturing. Its most significant contribution is standardization and encapsulation of the interface to mechatronic devices on an open, commercially available platform. It is clear that a driver specification like SIOMS could be added to the list of standardized I/O tools available on UNIX. Due to its axis based control it is not suitable for mechanisms with significantly coupled dynamics (Murray and Sastry, 1989), however for simpler systems this is not a problem. The advantages are that upgrades to newer and faster processors are simplified. The solution is portable. It is quick to learn. Finally, since the system is open, drivers and applications may be freely written and shared.