INFORMATION TO USERS

The most advanced technology has been used to photograph and reproduce this manuscript from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

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

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

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

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

U·M·I

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

Order Number 9035303

Signal processing for sensor arrays

Soykan, Orhan, Ph.D.

Case Western Reserve University, 1990

U·M·I

**300 N. Zeeb Rd.
Ann Arbor, MI 48106**

—

SIGNAL PROCESSING FOR SENSOR ARRAYS

by

ORHAN SOYKAN

Submitted in partial fulfillment of the requirements
for the Degree of Doctor of Philosophy

Dissertation Advisors: Michael R. Neuman, M.D., Ph.D.
Howard J. Chizeck, Sc.D.

Department of Electrical Engineering and Applied Physics
CASE WESTERN RESERVE UNIVERSITY

May 20, 1990

CASE WESTERN RESERVE UNIVERSITY
GRADUATE STUDIES

We hereby approve the thesis of

Orhan SOYKAN
candidate for the Ph.D.
degree.*

Signed:

Michael R. Neum
(Chairman)
Francis L. Merat
Herb J. Phillips
Patrick E. Cagg
Kenneth A. Romano

Date April 10, 1990

*We also certify that written approval has
been obtained for any proprietary material
contained therein.

SIGNAL PROCESSING FOR SENSOR ARRAYS

Abstract

by

ORHAN SOYKAN

Five methods have been developed to process output signals from sensor arrays to improve sensor characteristics. A multielement capacitive force sensor array with nonlinear characteristics and inter-element cross-talk was used to demonstrate and compare the performance of these methods: (1) Variable Threshold, which calculated a threshold value from a histogram of the data to determine whether the output of each element was due to an input or cross-talk; (2) Algebraic Cross-Talk Reduction, which reduced the cross-talk between array elements by solving for the input matrix from the input-output relation of the sensor; (3) 2x2 Edge and Shape Detection, which was adapted from image processing techniques and enhanced the resolution of the sensor output; (4) Deconvolution using FFT; and (5) Artificial Neural Network, which was trained to learn the inverse relation of the sensor transfer function. The ability of these methods to reduce the cross-talk seen

between sensor elements and more closely estimate the sensor input was studied.

Preprocessing of the sensor output, which consisted of calibration of each element using non-linear curve fitting, was used to compensate for the non-linear, non-uniform sensitivity characteristics of the sensor array elements. Postprocessing compensated for the dependency on the number of excited elements.

A comparative study of eight force sensor arrays using a 200 element test set showed that the simpler signal processing methods, Variable Threshold and 2x2 Edge and Shape Detection, performed well only for a limited number of input conditions especially when forces were applied through small objects. In general, neither reduced the error in determining total force by more than 30 percent. The Artificial Neural Network method, the most complicated and computationally intensive method during initial training but simple to implement thereafter, produced superior results for the majority of the input patterns applied to the sensor array, reducing the overall error by 6 to 83 percent.

All five signal processing techniques reduced the overall errors of the sensor outputs, some more consistently than others. However, when the overall error magnitude was less than 10 percent, only the Artificial Neural Network reduced the error consistently, making it the method of choice.

The results of this study showed that sensors having outputs with high error values can be converted into sensors with much lower output error using appropriate signal processing methods. Incorporation of such signal processing methods into sensor arrays allows the use of sensors with less than ideal characteristics.

ACKNOWLEDGMENTS

I would like to first thank my research advisor Dr. Michael R. Neuman for his scientific and financial support for my work. I learned a lot from him and I enjoyed working with him. I must also thank the rest of my committee members, Francis L. Merat, Howard J. Chizeck, Patrick E. Crago and Kenneth A. Loparo for their advice during this research. I also want everyone in the Electronics Design Center to know that I did like working with them.

Second, I want to thank my family for supporting me patiently during my graduate studies abroad both financially and morally.

Perhaps the person who worked second hardest (me first), Lisa P. Vervena must get all the credit she deserves (which may be not as much as $\frac{2}{3}$, as MRN suggested). I don't even want to think where my dissertation would be without her help.

Last, but not least, I would like to thank "The National Institute of Neurological and Communicative Disorders and Stroke" for supporting this work under contract number NO-1-NS-6-2302.

TABLE OF CONTENTS

Abstract	ii
Acknowledgments	v
Table of Contents	vi
List of Figures	x
List of Tables	xv
Chapter 1: Introduction	1
1.1 Statement of the Problem	1
1.2 Advantages and Disadvantages of Sensor Arrays	2
1.3 Organization	4
Chapter 2: Background and Previous Work	5
2.1 Literature Review	5
2.2 An Example Sensory System	9
Chapter 3: Sensor System and Preprocessing	12
3.1 Measurement System	12
3.2 Circuitry	15
3.3 Sensor Characterization	16
3.4 Reduction of Non-Uniformity	21

Chapter 4:	Variable Threshold Method	26
4.1	Description of the Variable Threshold Method	26
4.2	Training of the Variable Threshold Method	31
4.3	Implementation of the Variable Threshold Method	32
Chapter 5:	Algebraic Cross Talk Reduction Methods .	35
5.1	Algebraic Cross Talk Reduction Methods .	35
5.2	An Example for the Determination of an Algebraic Input-Output Relation	35
5.3	A Simplified Solution	41
5.4	A Full Solution for Algebraic Cross Talk Reduction	43
Chapter 6:	Edge and Shape Detection Methods	50
6.1	Edge and Gradient Detection	50
6.2	3x3 Gradient Detection	51
6.3	3x3 Shape Detection Masks	57
6.4	2x2 Detectors	62
6.5	Optimization of Mask Coefficients	67
6.6	Usage of 2x2 Masks	79

Chapter 7: Deconvolution Method	81
7.1 Training of the Deconvolution Method . .	81
7.2 Execution of the Deconvolution Method . .	85
 Chapter 8: Signal Processing with Artificial	
Neural Networks	88
8.1 Structure of Artificial Neural Networks .	88
8.2 Associated Memories	91
8.3 Feed Forward Artificial Neural Networks .	93
8.4 Implementing the Feed Forward Signal	
Processor	104
 Chapter 9: Experiments	107
 Chapter 10: Results	116
10.1 Relative Error of Total Force	
Determination	116
10.2 Preliminary Results Obtained with	
Sensor A6	125
10.3 Post Processing	128
10.4 Improvement Factors for Relative Error	
Values	144
10.5 Error Dependence on Object Area	160
10.6 Error Dependence on the Applied Pressure	161

10.7	Error Dependence on Object Alignment . .	166
10.8	Error Dependence on Object Shape	169
10.9	Range of Error Values	170
10.10	Error in the Area Estimations	174
10.11	Summary	175
Chapter 11: Conclusions		184
References		191

APPENDICES

Appendix A:	Circuit Diagrams	197
Appendix B:	Generalized Delta Rule (The Perceptron Algorithm)	201
Appendix C:	Test Element Set	208

LIST OF FIGURES

2.1	Mechanical Structure of the Example Sensor Used in These Studies	11
4.1	Histogram of Sensor Output for Washer Shaped Input	28
4.2	Raw Data Obtained by Applying a Washer Shaped Object to the Multi Element Force Sensor	29
4.3	Output of Variable Threshold Method for the Washer Example	34
5.1	Cross Talk Seen When 50 g Weight is Applied to Element [5,4] of a 64-Element Capacitive Force Sensor	36
5.2	Graphical Illustration of Equation 5.1	38
5.3	Graphical Illustration of $C F = V$	39
5.4	Graphical Illustration of $F R = V$	40
5.5	Output of Algebraic Cross Talk Reduction Method with Column Cross Talk Reduction Alone for the Washer Example	44
5.6	Output of Algebraic Cross Talk Reduction Method with Full Cross Talk Reduction for the Washer Example	49

6.1	Directional Definitions	52
6.2	3x3 Gradient Masks (Edge Detection Masks) . .	54
6.3	An Example Edge	55
6.4	3x3 Shape Detection Masks	58
6.5	Results of Applying 3x3 Masks to Washer Example	61
6.6	A 2x2 Detector Placed on the Common Corner of Elements [4,3], [4,4], [5,3] and [5,4] . .	65
6.7	2x2 Detectors	66
6.8	Numbering Sequence Used in 2x2 Data Sets and Masks	69
6.9	2x2 Masks to be Optimized	71
6.10	Test Data Containing Fifteen Sets of 2x2 Patterns	72
6.11	Optimized 2x2 Masks	77
6.12	Results of Applying 2x2 Masks to the Washer Example	80
7.1	Results of Applying Deconvolution Method to the Washer Example	87
8.1	Transfer Function of a Neuron	89
8.2	A Two Layer Neural Network	95
8.3	Output of Neural Network Signal Processor for the Washer Input	106

9.1	Results of the Repeatability Study Obtained with Sensor A6 by Application of Ten Objects Ten Times to the Sensor Surface	113
10.1	An Example for Negative Cross-Talk	120
10.2	Mechanical Structure of the Sensor Array . . .	124
10.3	Boxplot of Relative Error Values Presented in Table 10.1, for Sensor A6, Pressure = 50 Grams/Cell	129
10.4	Bar Graphs of Relative Error Values Presented in Table 10.1, for Sensor A6, Pressure = 50 Grams/Cell	130
10.5	Scatterplot of Relative Error Values for VTM, ALG and FFT, Presented in Table 10.1 for Sensor A6, Pressure = 50 Grams/Cell	132
10.6	Boxplot of Relative Error Values Presented in Table 10.4, for Sensor A6, Pressure = 50 Grams/Cell	137
10.7	Bar Graph of Relative Error Values Presented in Table 10.4, for Sensor A6, Pressure = 50 Grams/Cell	138
10.8	Bar Graph of Overall Relative Error Values for All Sensors, All Pressures, shown in Table 10.6	141

10.9	Bar Graph of Overall Relative Error Values for All Sensors, All Pressures, shown in Table 10.7	143
10.10	Boxplot of Improvement Factors Presented in Table 10.8, for Sensor A6, Pressure = 50 Grams/Cell	147
10.11	Boxplot of Improvement Factors Presented in Table 10.9, for Sensor A6, Pressure = 50 Grams/Cell, Post Processed	149
10.12	Boxplot of Relative Errors Presented in Table 10.10, for All Sensors, All Pressures	152
10.13	Boxplot of Relative Errors Presented in Table 10.11, for All Sensors, All Pressures Combined and Post Processed	154
10.14	Boxplot of Improvement Factors Presented in Table 10.12, for All Sensors, All Pressures Combined	156
10.15	Boxplot of Improvement Factors Presented in Table 10.13, for All Sensors, All Pressures Combined and Post Processed	158
10.16	Comparative Boxplots for Improvement Factors Presented in Tables 10.12 and 10.13, for All Sensors, All Pressures Combined, Before and After Post Processing	159

10.17	Bar Graph of Overall Improvement Factors Presented in Table 10.14, for All Sensors, All Pressures	163
10.18	Bar Graph of Improvement Factors Presented in Table 10.15, for All Sensors, All Pressures, Post Processed	165
10.19	Object Number 14 Applied to Sensor A6 at Five Locations, Five Pressures	176
10.20	Raw and Processed Area Data Obtained by Applying Object #14 (Area of 12 Cells) To Sensor A6, at Five Locations with Five Pressures	177
10.21	Boxplot of Relative Area Errors Presented in Table 10.20, for All Sensors and All Pressures	179
10.22	Bargraph of Relative Area Errors Presented in Table 10.20, for All Sensors and All Pressures	180
10.23	Boxplot of Improvement Factors Presented in Table 10.21, for Area Estimations Using All Sensors, All Pressures	182
10.24	Bar Diagram of Improvement Factors Presented in Table 10.21, for Area Estimations Using All Sensors and All Pressures	183

LIST OF TABLES

6.1	The Responses of Different Masks to the Test Patterns	78
10.1	Relative Error Values for Sensor A6, Pressure = 50 grams/cell	118
10.2	Cross Talk and Relative Error Values for Sensors Used in the Comparative Study of Signal Processing Techniques	119
10.3	Cross Talk Values for Sensor A6 at Different Force Levels	123
10.4	Relative Error Values for Sensor A6, Pressure = 50 grams/cell, Post Processed . . .	135
10.5	Effect of Post Processing on the Overall Total Force Estimation for Sensor A6, Pressure = 50 Grams/Cell	136
10.6	Overall Relative Error Values for All Sensors, All Pressures	140
10.7	Overall Relative Error Values for All Sensors, All Pressures, Post Processed . .	142
10.8	Improvement Factors for Sensor A6, Pressure = 50 grams/cell	146
10.9	Improvement Factors for Sensor A6, Pressure = 50 grams/cell, Post Processed . . .	148

10.10	Relative Errors for All Sensors, All Pressure Levels	151
10.11	Relative Errors for All Sensors, All Pressure Levels, Post Processed	153
10.12	Improvement Factors for All Sensors, All Pressure Levels Combined	155
10.13	Improvement Factors for All Sensors, All Pressures Combined and Post Processed	157
10.14	Overall Improvement Factors for All Sensors, All Pressures	162
10.15	Overall Improvement Factors for All Sensors, All Pressures, Post Processed	164
10.16	Relative Error Values for All Sensors, All Pressures	167
10.17	Relative Error Values With and Without Alignment for All Sensors, All Pressures, Post Processed	168
10.18	Overall Relative Error Values for All Sensors, All Pressures, Post Processed and Presented as Median and Range	172
10.19	Overall Improvement Factors for All Sensors, All Pressures, Post Processed and Presented as Median and Range	173
10.20	Relative Error Values for Area Estimations from All Sensors, All Pressures	178

10.21	Improvement Factors for All Sensors, All	
	Pressures for Area Estimations	181

Chapter 1. INTRODUCTION

1.1 Statement of the Problem

This dissertation considers the problem of extracting measurement information about a quantity from an array of less than optimal sensors. New technologies are being utilized to produce novel types of sensors and more extensive sensing and instrumentation systems. As these systems become more complex, and as compromises with regard to manufacturing cost and sensor quality need to be made, it becomes evident that new techniques in signal processing must be developed to optimize the performance of the signals from these sensors. Although it may be possible to manufacture sensors with close to ideal characteristics, this is likely to be quite costly. The question then becomes, given a sensor easy and inexpensive enough to mass manufacture which does not have very good characteristics, is it possible to use signal processing to compensate for these non-ideal characteristics and make the signal coming from the sensor look as if it is coming from a sensor with close to ideal characteristics?

This question is addressed by first developing and adapting five methods designed to process two dimensional signals coming from a sensor array. Secondly, the

performance of these signal processing systems is comparatively analyzed for an example force sensor array. The unique approach of this dissertation is the introduction of signal processing methods for use with sensor arrays in an effort to improve sensor performance. In this way, easy to manufacture sensor arrays with less than ideal characteristics can be made to appear to have better performance.

1.2 Advantages and Disadvantages of Sensor Arrays

Single sensors can be used to measure a quantity in the vicinity of a given point in space. If the distribution of the measured quantity across an area instead of at a single point is desired, then sensor arrays must be used. There are two ways to form sensor arrays: (1) one can place N rows of M individually connected sensors to form a $N \times M$ sensor array, or (2) one can lay strips of conductive material at right angles to each other to form the rows and columns of the sensor array with the individual sensors formed at the intersections of the rows and columns. Both of these methods have advantages and disadvantages. The former, discrete design will have more leads and interconnections, and is therefore less reliable. The increased number of wires can be a disadvantage if space is limited and some signal conditioning electronics are

needed to be placed on the sensor. However, it has no electrical cross coupling between the elements which is an advantage. Also, one can obtain and process the data from the sensors in parallel. On the other hand, the latter design will have fewer lead wires but may suffer from the cross coupling between the elements, and only one element can be addressed at a time. Even though this cross coupling (referred to as *cross-talk*), can be reduced in some cases by proper scanning and feed-back circuits, accurate measurements from the sensor array can only be obtained after processing the signal from the sensor.

All sensor arrays also suffer from the common drawbacks of the individual sensors that make up the array, such as non-linearity and non-uniformity. *Non-linearity* is the phenomenon where a sensor (whether single or in an array) does not have a linear input-output relationship, and therefore, the superposition principle does not hold. That is, if the input to the sensor is doubled in magnitude, the output does not double in magnitude.

If the elements of a sensor array have characteristics different from each other, this is referred to as *non-uniformity*. Furthermore, if the sensor

characteristics are distributed non-uniformly across the sensor surface, then the signal from each sensor element must be processed separately.

1.3 Organization

This dissertation considers five different signal processing methods developed to process the signals coming from a sensor array. These five methods were either specifically developed or adapted from other disciplines for sensor array signal processing. Chapter 2 will describe the example sensor array upon which this signal processing study was concentrated. Chapter 3 demonstrates methods to compensate for non-linearity and non-uniformity of sensor arrays. Chapters 4 through 8 describe five signal processing methods to reduce cross-talk. Chapter 9 will describe the experiments performed to evaluate the performance of all five methods and the results will be presented in Chapter 10. The final chapter presents conclusions drawn from this research.

Chapter 2. BACKGROUND AND PREVIOUS WORK

This chapter reviews the relevant literature published to date. Next, a description of the example sensor array on which the various signal processing studies were tested is presented. The problems associated with this sensor array are then considered.

2.1 Literature Review

In the past, there have been many attempts to build tactile sensor arrays for various applications such as robotics and biomedical use. All of these sensor arrays provided signals related to the sensor input, but most of them required their signals to be processed before the final information could be obtained.

S. A. Stansfield (1986) used a sensor array composed of photo-sensors attached to a flexible substrate such that application of a normal force decreased the amount of light received by the photo-sensors. The sensor array was attached to a robot arm and was intended for use in exploring the contours of surfaces, the hardnesses of which was indicated by the grey values from the sensor array. This work used the ratio of the standard deviation to the mean (coefficient of variation) of array entries to determine the texture of the region of contact. Since the elements of the sensor array had

non-linear sensitivity characteristics and cross-talk among them, a single element force sensor placed under the photo-sensor array was used to measure the total force. This total force measurement was required for the computations of compliance and elasticity where the sensor array was used to determine the force distribution.

P. Dario and G. Butazzo (1987), developed a cylindrical sensor array consisting of piezoelectric elements, again to be used for exploratory procedures. This sensor's elements suffered from extensive drift between successive measurements. Therefore, signal processing was performed to reduce the effects of drift by a form of double-sampling of the sensor output.

The Utah-MIT hand [C. Jacobsen, et al., 1984; D. Siegel, 1986] used a capacitive force sensor array. Researchers noted and reported cross-talk between the sensor elements as well as non-linear and non-uniform sensitivity distributions. They corrected their data to reduce the effects of the non-uniformity, but ignored the effects of the non-linearity and cross-talk. Further, they claimed that it is advantageous to have cross-talk between the elements since it provides additional input which compensates for the loss of data due to the

insensitive gap between elements. They did not provide any data about sensor performance.

R.S. Fearing and J.M. Hollerbach (1985) described the mechanical properties of the materials used for the construction of tactile sensors. Their results were used by R.S. Fearing, et al. (1986) in the construction of a sensor to estimate the force magnitude and angle of contact by simply measuring the strain in different locations in the sensor material. This sensor was then used in the Stanford-JPL hand [R.S. Fearing, 1987] to measure the forces on the fingertips.

G. Kinoshita, et al. (1975) built their sensor array by molding semi-elastic materials between piezoelectric sensors. The goal of most of their signal processing was recognition of the input pattern of the sensor array. Later, G. Kinoshita (1983) used strain-gauges placed under a flexible substrate to build a sensor array. This study used a simple estimated relation between the input forces and measured strains. Deconvolution was used to determine the input values.

M.R. Driels (1986) used tactile sensor arrays composed of conductive elastomers to determine the position and orientation of an object placed on the

sensor array by calculating the inertia of the image. The signal was processed assuming that the sensor was ideal and concentrated on the calculation of the inertia.

R. Yang and S. Krishnaprasad (1987) used strain gauges to measure strains in the compliant material comprising a sensor to determine the sensor input. They used the Discrete Fourier Transform to invert the output data to obtain the input to the sensor array. The results were accurate only for some specific input geometries. Y.C. Pati, et al. (1987), carried out a similar study using artificial neural networks. Their results were obtained via computer simulation and showed that stress profiles can be deconvolved from stress measurements.

In general, the research reported above was case specific, meaning that the signal processing was not generally applicable to other sensors since either the imperfections in the sensor characteristics were ignored, or were corrected only for that particular sensor. In most of the literature cited above, signal processing neither corrected the grey level information of each element nor restored the overall sensor input precisely. Therefore, the need exists for generalized signal processing methods for sensor arrays. This dissertation

will investigate some general methods for signal processing for sensor arrays. The following section will describe the example sensor used for evaluation and comparison of the signal processing methods developed.

2.2 An Example Sensory System

A 64-element capacitive force sensor array has been designed for use in the feedback control of a functional electrical stimulation system to provide hand grasping function for C5/C6 spinal cord injury patients. Microelectronic technology is used to fabricate the sensor, and these production techniques can introduce some problems in the performance of the finished device. Each of the 64 capacitive elements of the sensor has non-linear and non-uniform sensitivity characteristics, and there is a significant amount of cross-talk between the elements. These limitations are inherent to the sensor design.

The 64-element capacitive force sensor array is formed by two sets of eight parallel conducting strips on flexible polyimide substrates, positioned such that the strips on one substrate are at right angles to the strips on the opposite substrate. A patterned silicone elastomer dielectric film is printed on both the top and bottom substrate. The sensor is then formed by

contacting the silicone surfaces of each substrate together and allowing the elastomer to cure completely. This compliant dielectric permits a change in the separation between the top and bottom electrodes when a normal force is applied, resulting in a change in the capacitance. A diagram of the sensor can be seen in Figure 2.1. The substrates extend approximately 2.5 cm past either end of the sensitive region of the sensor to allow easy attachment on a finger and provide active electrical shields to the lead conductors. A hybrid microelectronic circuit placed adjacent to the sensor scans the sensor array to sequentially charge each capacitive element (which is formed at the intersections of the strips of electrodes) then sample the voltage on each charged element. A similar design is described in detail by A. Berec, et al. (1987). Variations in the materials and the techniques used for the sensor manufacturing caused considerable differences in the characteristics of the sensor arrays from different batches. For example, sensors from some of the batches over-estimated the total force applied to the sensor array while a sensor from another batch under-estimated the same force. These differences will be discussed as they are encountered in the dissertation.

Figure 2.1 Mechanical structure of the example sensor used in these studies.

Chapter 3. SENSOR SYSTEM AND PREPROCESSING

This chapter describes the underlying principles of the measurement system developed for the capacitive force sensor array. The design of the circuits used is then presented. Finally, solutions to some of the problems of the sensor array such as non-linearity and non-uniformity are described.

3.1 Measurement System

Individual elements of the sensor array described in Chapter 2 were charged sequentially. During the charging of one element, potentials of the remaining cells were forced to specific levels in order to reduce electrical cross-talk. When the sampling (and the charging) were done at a very high speed, there was not enough time to discharge the capacitive elements between successive charging pulses. This caused a charge build-up on the elements which was a function not only of the previous voltage values of each element, but also of the previous values of the other elements due to the circuit providing the driven shield. If a fixed duration charging pulse were sent to the RC circuit and the capacitor voltage was sampled to determine the capacitance value, this sampled voltage would be a function of both the capacitance and the initial charge. In order to measure the capacitance

independent of the initial charge, a double sample method was developed.

The output voltage of an RC circuit with an initial charge on the capacitor is given as follows:

$$V_o(t) = (V_f - V_i) (1 - e^{-t/RC}) + V_i \quad (3.1)$$

where V_o is the output voltage (capacitor voltage)
 V_f is the magnitude of the charging pulse,
 V_i is the initial charge on the capacitor

Rewriting equation 3.1,

$$V_o(t) = V_f - (V_f - V_i) e^{-t/RC} \quad (3.2)$$

or

$$V_o(t) - V_f = - (V_f - V_i) e^{-t/RC} \quad (3.3)$$

Solving for the exponential term,

$$e^{-t/RC} = \frac{V_o(t) - V_f}{V_i - V_f} \quad (3.4)$$

or

$$- \frac{t}{R C} = \ln \left(\frac{V_o - V_f}{V_1 - V_f} \right) \quad (3.5)$$

If the output voltage at $t = t_1$ is V_1 and V_2 at $t = t_2$, then,

$$- \frac{t_1}{R C} = \ln \left(\frac{V_1 - V_f}{V_1 - V_f} \right) \quad (3.6)$$

$$+ \frac{t_2}{R C} = \ln \left(\frac{V_1 - V_f}{V_2 - V_f} \right) \quad (3.7)$$

Defining $\Delta t = t_2 - t_1$, and adding equations 3.6 and 3.7,

$$\frac{\Delta t}{R C} = \ln \left(\frac{V_1 - V_f}{V_2 - V_f} \right) \quad (3.8)$$

Solving for the capacitance value,

$$C = \frac{\Delta t}{R \ln \left(\frac{V_1 - V_f}{V_2 - V_f} \right)} \quad (3.9)$$

If Δt and R are fixed, only the following term determines the value of the capacitance from its charging curve independent of the initial charge:

$$\left(\frac{V_1 - V_f}{V_2 - V_f} \right)$$

The remainder of the non-linear operations involved in capacitance calculations using equation 3.9, such as natural logarithm and reciprocal computations, are ignored because the overall relation between applied force and sensor output is determined by curve fitting, which will be explained in detail in Section 3.3.

3.2 Circuitry

A complete schematic of the circuit used to scan the sensor array is presented in Appendix A. An analog divider as well as the addressing and signal conditioning circuitry provides the multiplexed signal from the sensor array to a micro-controller. The micro-controller, which also governs the timing of the entire system, converts the analog signal to a digital signal and stores it in memory. Signal processing can be done either in the micro-controller or in a host computer which is connected to the micro-controller via a standard RS-232 port.

3.3 Sensor Characterization

To determine whether a sensor has a linear input-output relationship, a number of inputs can be applied and the outputs recorded. A simple plot of the input versus output will show if the sensor is linear. If the sensor has a linear input-output relationship, then the calibration coefficient for each element consists of a gain constant and an offset value which can be determined by linear regression. Conversely, if the sensor has a non-linear input-output relationship, then a non-linear model of this relationship must be determined for calibration purposes. This model can be obtained either by a theoretical analysis of the sensor mechanics or it can be determined empirically. Whichever method is used for the determination of the model, calibration coefficients can be obtained by curve fitting. Although there are many ways to curve fit mathematically, one of the most straightforward techniques is to linearize the terms in the model, then carry out a linear regression. The following example should help clarify the explanation of this process.

Assume a non-linear model is needed in the following form:

$$v(m) = a \frac{1}{c + m} + b \quad (3.10)$$

where v is the measured output of the sensor,
 m is the input to the sensor,
 a and b are constant coefficients, and
 c is the non-linear parameter.

This relationship can be linearized by defining a new function, $z(c,m)$ as follows:

$$z(c,m) = \frac{1}{c + m} \quad (3.11)$$

Equation 3.10 can now be rewritten as

$$v(m) = a z(c,m) + b \quad (3.12)$$

The goal of the curve fitting algorithm is to eliminate the error which is the difference between the model output and the sensor output. Given a set of v_i and m_i , containing n data points, the objective is to choose parameters a , b and c to minimize the sum of squared errors. The sum of squared errors function is as follows:

$$g(a,b,c) = \sum_{i=1}^n \{ v_i - [a z(c, m_i) + b] \}^2 \quad (3.13)$$

which has the solution, in terms of $z_i = z(c, m_i)$:

$$a = \frac{\sum_{i=1}^n z_i v_i - \sum_{i=1}^n z_i \sum_{i=1}^n v_i}{\sum_{i=1}^n z_i^2 - \sum_{i=1}^n z_i \sum_{i=1}^n v_i} \quad (3.14)$$

$$b = \frac{-\sum_{i=1}^n z_i \sum_{i=1}^n z_i v_i + \sum_{i=1}^n z_i \sum_{i=1}^n v_i}{\sum_{i=1}^n z_i^2 - \sum_{i=1}^n z_i \sum_{i=1}^n v_i} \quad (3.15)$$

Therefore, for a given set of v_i and m_i values, optimum values of a and b can be uniquely determined for a chosen c via equations 3.14 and 3.15 which yield the linear regression coefficients [Devore, 1987]. Hence g is a non-linear function of c . The optimization problem can now be restated as finding a constant, c , which will minimize the sum of the squared errors, $g(c)$:

$$g(c) = \sum_{i=1}^n [v_i - v(m_i)]^2 \quad (3.16)$$

A simple non-linear optimization technique (e.g. golden section search [Luenberger, 1984]) can be used to determine the optimum value of the variable c .

The empirical non-linear relation given by equation 3.11 was chosen for a variety of reasons. A previous study [Berec, 1989] showed that the sensor array used for this study can be modeled using this reciprocal type relation. Even though exponential and logarithmic relations (e.g., equations 3.17 and 3.18)

$$z(c,m) = e^{c \cdot m} \quad (3.17)$$

$$z(c,m) = \ln(c+m) \quad (3.18)$$

also fit well to the experimental data, they did not give g values as low as did the reciprocal model. In addition, the exponential and logarithmic models were not suitable for the real time floating point arithmetic implementations since they required calculations of Taylor Series expansions. Therefore, the reciprocal

model was used for the force calibration curves generated for the sensor arrays described in Chapter 2.

If there are more than one non-linear functions (i.e., more than one z) or more than one non-linear variable (i.e., more than one c), the problem could be treated similarly. First, the linear and non-linear variables are separated. If the relation representing the sensor model is written as the sum of the products of some scalar numbers and some non-linear functions, then the scalar numbers will be the linear coefficients and the constants in the non-linear functions will be the non-linear coefficients. The numerical values of the linear coefficients can then be determined via linear regression and the non-linear coefficients by some non-linear optimization technique. For the case described in equation 3.10, there is only one non-linear coefficient and the golden section search algorithm is sufficient. If there are more than one non-linear coefficients to be determined, then it is necessary to use a multi-dimensional non-linear optimization technique such as steepest descent, Fletcher-Reeves or Davidon-Fletcher-Powell [Luenberger, 1984]. Among these three, Davidon-Fletcher-Powell has been shown to be the most effective [Soykan, 1986].

3.4 Reduction of Non-Uniformity

Another problem associated with sensor arrays is the non-uniform distribution of the sensor characteristics across the sensor surface. Sensor elements show variations not only in their sensitivity profile, but also in their cross-talk characteristics. As with non-linear sensitivity distributions, non-linear cross-talk characteristic distributions are usually due to the non-uniform mechanical structure of the sensor. Regardless of the cause, non-uniformity makes development of cross-talk reduction algorithms very difficult. A simple way to reduce the undesired effects of the non-uniformity is explained in this section.

The first step in the reduction of non-uniformity is the determination of the two dimensional impulse response, or the *point spread function*. Since each element of the sensor behaves differently, one needs to determine this function independently for each element of the sensor array. This can be done by simply applying a known input to one of the elements and observing the outputs of all the elements in the sensor array. When these outputs are converted into the estimated values using the calibration curves described in the previous section, the excited cell should reflect the actual

input while all other cells should reflect the cross-talk values. These numbers should be normalized by dividing all the outputs by the maximum output, where the maximum output will come from the cell actually excited. This final matrix contains the point spread function for the element under consideration. A step by step formulation of this procedure for reduction of the effects of non-uniformity is as follows:

Assume the sensor array is $N \times N$, then,

1. Apply a known input, $F[m,n]$ to cell $[m,n]$,
2. Measure the outputs from all the cells,
 $v[i,j]$ where $i=1,\dots,N$ and $j=1,\dots,N$
3. Convert these output measurements into estimates of the input quantities using the previously obtained calibration curves,
 $f[i,j]$ where $i=1,\dots,N$ and $j=1,\dots,N$
4. Scale these estimates by the maximum, $f[m,n]$,

$$h_{mn}[i,j] \leftarrow f[i,j]/f[m,n],$$

where $i=1,\dots,N$ and $j=1,\dots,N$
5. Repeat the previous four steps for all m and n values.

At the end of this procedure, one obtains a number of h matrices, equal to N^2 , each of which is $N \times N$ in size. These h matrices are sufficient to reduce the non-uniformity if only one cell is excited. However, during normal usage of the sensor, more than one cell will be excited. Therefore, the h matrices calculated will not be very useful. Even when only one cell is excited, it is not known which cell receives the input, so the choice of h is not clear. To eliminate this problem, it is possible to combine all of the h matrices to form a $N^2 \times N^2$ matrix, H , which will represent the overall cross-talk matrix. If $F[i,j]$ represents the actual excitation of the cell $[i,j]$ and $f[i,j]$ represents the estimation resulting from the predetermined calibration, then the formulation of the H matrix is as follows:

$$\begin{bmatrix} f[1,1] \\ f[1,2] \\ \vdots \\ f[1,N] \\ \hline f[2,1] \\ f[2,2] \\ \vdots \\ f[2,N] \\ \hline \vdots \\ \hline f[N,1] \\ f[N,2] \\ \vdots \\ f[N,N] \end{bmatrix} = \begin{bmatrix} h_{11}[1,1] & h_{12}[1,1] & \dots & h_{NN}[1,1] \\ h_{11}[1,2] & h_{12}[1,2] & \dots & h_{NN}[1,2] \\ \hline h_{11}[1,N] & h_{12}[1,N] & \dots & h_{NN}[1,N] \\ \hline h_{11}[2,1] & h_{12}[2,1] & \dots & h_{NN}[2,1] \\ h_{11}[2,2] & h_{12}[2,2] & \dots & h_{NN}[2,2] \\ \hline h_{11}[2,N] & h_{12}[2,N] & \dots & h_{NN}[2,N] \\ \hline \hline \hline h_{11}[N,1] & h_{12}[N,1] & \dots & h_{NN}[N,1] \\ h_{11}[N,2] & h_{12}[N,2] & \dots & h_{NN}[N,2] \\ \hline h_{11}[N,N] & h_{12}[N,N] & \dots & h_{NN}[N,N] \end{bmatrix} \begin{bmatrix} F[1,1] \\ F[1,2] \\ \vdots \\ F[1,N] \\ \hline F[2,1] \\ F[2,2] \\ \vdots \\ F[2,N] \\ \hline \vdots \\ \hline F[N,1] \\ F[N,2] \\ \vdots \\ F[N,N] \end{bmatrix}$$

(3.19)

or

$$\mathbf{f}_{N^2 \times 1} = \mathbf{H}_{N^2 \times N^2}^{-1} \mathbf{F}_{N^2 \times 1} \quad (3.20)$$

Equation 3.20 describes the relation between the actual inputs to the sensor, F , and the outputs of the

sensor after estimations, f , based on the predetermined calibration curves. At any given time, the output of the sensor is known, as are the estimated inputs. In order to find the actual inputs of the sensor array, F , one needs to premultiply f with H^{-1} as shown below:

$$F_{N^2 \times 1} = H_{N^2 \times N^2}^{-1} f_{N^2 \times 1} \quad (3.21)$$

If the sensor behaves linearly and obeys the superposition principle after calibration, then equation 3.20 will determine the inputs of the sensor array from its outputs. That is, in the case where only one cell is excited, if the input to that cell is doubled, all outputs should double. Also, the sensor response to a multi-cell excitation should be equal to the sum of the individual responses. If the superposition principle is not applicable even after calibration, equation 3.21 will not be sufficient for the determination of the sensor input from its outputs, and further processing of the signal is needed. The following five chapters describe various signal processing methods for cases where superposition does not hold.

Chapter 4. VARIABLE THRESHOLD METHOD

4.1 Description of the Variable Threshold Method

One of the fundamental requirements of array signal processing is to discriminate between the cells actually measuring the input quantities and those measuring artifacts such as cross-talk and noise. The magnitude of the noise seen at the output of a sensor is generally not a function of the input, but the outputs of the cells reflecting cross-talk will vary as the input pattern and input magnitude vary. For example, if the magnitude of the input is increased without changing its shape, the responses from the elements reflecting cross-talk will increase as well as the outputs of the elements which are directly excited. The elements which measure only noise will still have the same output as before. Therefore, for a given sensor array, cells can be classified into three categories: (1) elements responding to the actual excitation which will have varying output magnitude as the input changes, (2) elements showing a response as a result of cross-talk alone, which again have varying output magnitude, and (3) elements simply measuring noise whose outputs should have upper limits much lower than the outputs of the cells in the other two categories.

One of the most straightforward ways to distinguish the excited elements from the others is to choose a threshold level and label all elements with output levels higher than this threshold as the excited elements since the elements reflecting cross-talk or noise should have much lower output levels than the excited cells. In order for this method to work properly, a different threshold level must be chosen each time the sensor output array is sampled. One way to do this is to statistically analyze the output distribution of the sensor elements at each sample time. This is done by plotting a *histogram* of the data where the histogram is a plot of output level versus the number of cells showing that level of output. An example of such a histogram is presented in Figure 4.1 for the sensor output presented in Figure 4.2. In this case, a washer shaped object was pressed against the sensor. As can be seen from Figure 4.2, the washer shape is hardly recognizable justifying the need for a signal processing system. An input pattern with fairly uniform excitation (over the cells which it excites) will cause a cluster of cells on the right most side of the histogram. There will be another cluster(s) formed by the cells reflecting cross-talk and cells measuring noise to the left hand side of this hill shaped cluster. The threshold required to separate the

Figure 4.1 Histogram of sensor output for washer shaped input.

Figure 4.2 Raw data obtained by applying a washer shaped object to the multi-element force sensor.

excited cells from the others should be placed in the deepest point of the valley surrounded by the last and penultimate hills of the histogram. If there is only one hill or the histogram is completely flat, then all the elements of the sensor are assumed to be excited. If this assumption is not correct, the method fails. Since the histogram is generated by quantizing the output levels, the actual value of the threshold must be set somewhere in the quantization interval chosen as the threshold interval. A logical selection for the actual threshold would be the center of the threshold interval. In order to have maximum resolution for a given number of quantization intervals (n), one can first determine the maximum output from the sensor array then choose the quantization level as one n -th of this maximum output. Therefore, for any given set of outputs, all the intervals in the histogram would be utilized due to the automatic scaling of the quantization interval. For example, if the maximum output is 400 grams for a given force sensor array, and it is desired to have eight quantization intervals, a quantization level of 50 grams will be chosen.

4.2 Training of the Variable Threshold Method

In order for the variable threshold method to work properly, the thresholds must be chosen as accurately as possible. Since the thresholds are determined by the shape of the histogram, it is also important to have the proper number of quantization intervals. If there are very few quantization levels in the histogram, then the cluster formed due to directly excited cells would be indistinguishable from the clusters due to cross-talk or noise, resulting in a very low threshold or none at all. On the other hand, if there are too many quantization levels, there is always a chance that outputs of the excited cells would form multiple peaks on the right side of the histogram (multiple peaks could be due to either non-uniform excitation at the input or non-uniform gain characteristics of the sensor array itself). Nevertheless, placing the threshold between the last two hills would cause some of the directly excited cells to be grouped with the elements reflecting cross-talk. Therefore, it is necessary to determine the optimum number of quantization levels and the number of hills to be left on the right hand side of the selected threshold value in the histogram such that the method would yield a close estimate of the sensor array input. An exhaustive

search¹, which constitutes the training of this method, can be used to determine these numbers using a known set of input and output values from the sensor array under study.

4.3 Implementation of the Variable Threshold Method

After the training phase for the variable threshold method is completed, the optimum number of quantization levels (n_q) and the number of hills on the right hand side of the threshold (n_h) are known. During actual run time, these two numbers should not be changed. When an input pattern is applied to the sensor array, the output of each cell is read and stored. The maximum of these stored output values is determined and the quantization level is chosen as one n_q -th of this maximum allowing n_q quantization intervals in the histogram. After the histogram is formed, the threshold is placed in the deepest point of the valley between hills n_h and n_h+1 . If there are not enough hills in the histogram (i.e. existing number of hills in the histogram is less than the optimum number of hills determined by the training

¹ In an exhaustive search, every possibility is explored. In this case, values of n_q and n are varied and the pair which minimizes the errors is chosen.

algorithm, then the threshold should be placed between the first and second hills from the left of the histogram since this would be the closest approximation to a typical case, separating noise from the rest of the elements. Again, the actual threshold value selection within the quantization interval should be consistent with the selection scheme used during training, e.g., choosing the actual threshold level at the center of the threshold interval. Once the threshold value is selected, all the cells in the stored output array with value larger than the threshold are identified as excited cells.

The results of applying the variable threshold method to the washer example seen in Figure 4.2 are presented in Figure 4.3. In this example, the threshold value was chosen as shown in Figure 4.1, i.e., 38.3 percent. As can also be seen from Figure 4.1, 54 elements of the 64 element sensor array generated outputs lower than the threshold value, so they were pruned away leaving a clear image of the washer shaped object which was applied to the sensor array.

Figure 4.3 Output of variable threshold method for the washer example.

Chapter 5. ALGEBRAIC CROSS TALK REDUCTION METHODS

5.1 Algebraic Cross Talk Reduction Methods

In certain cases, it is possible to obtain an algebraic input-output relation for a sensor array and use the inverse of this relation to recover (or at least estimate) the sensor inputs from its outputs. Even though there is no cook-book method to determine an algebraic relation between a sensor input and output, such a relation can be obtained just by looking at the two dimensional analog of the impulse response (the *point spread function*) of the sensor. Since each input-output relation would have a different inverse, this chapter will demonstrate the method with an example.

5.2 An Example for the Determination of an Algebraic Input-Output Relation

An algebraic relation may be obtained between the input and output of the sensor by observing its point spread function. An example of the point spread function for the sensor described in Chapter 2 can be seen in Figure 5.1. In this case, a 50 gram mass was applied only to element [5,4] of the 64 element force sensor. As can be seen from the figure, Element [5,4] showed the maximum response at the sensor output of all the

Figure 5.1 Cross-talk seen when 50 g weight is applied to element [5,4] of a 64 element capacitive force sensor

elements. However, some elements other than element [5,4] showed some response due to the effects of cross talk. A careful observation will show that most of the elements reflecting cross talk are either in the same row or in the same column as element [5,4]. It is apparent from the point spread function that the amount of cross talk measured decreases as the distance from the excited element increases. Also, the row cross talk appears to be much less than the column cross talk. If the sensor's response is assumed to be linear, which will require that superposition applies, then one can write an algebraic input-output relation for this case as follows:

$$C F + F R = V \quad (5.1)$$

where F is the 8x8 force matrix (sensor input)

C is the 8x8 column cross talk matrix

R is the 8x8 row cross talk matrix

V is the 8x8 voltage matrix (sensor output).

Equation 5.1 not only assumes that the sensor acts as a linear system, but also that all rows and all columns have identical cross talk characteristics. That is, if a force f applied to element $[i,j]$ causes a voltage difference V at element $[i,j+k]$, then the

application of the same force f to element $[m,j]$ will cause the voltage of element $[m,j+k]$ to change by V volts as well. Assumptions such as these are sensor dependent, and were valid for the sensor used with the washer example in Figure 4.2. Figure 5.2, which is a graphical illustration of equation 5.1, should help to understand the effects of the column and row cross talk matrices on sensor output. In this example, the force matrix has only one non-zero entry, $[2,3]$. The elements of the C and R matrices which generate the column and row cross talk are labelled with a lower case letter and the cross talk they generate is labelled with the upper case of the same letter in the V matrix.

$$\begin{array}{c} C \end{array} \quad \begin{array}{c} F \end{array} + \begin{array}{c} F \end{array} \quad \begin{array}{c} R \end{array} = \begin{array}{c} V$$

	a			
	b			
	d			
	e			
	h			

x	y	z	t	w

		A		
X	Y		T	W
		D		
		E		
		H		

Figure 5.2 Graphical illustration of equation 5.1.

In order to solve equation 5.1, it is necessary to determine the column and row cross talk matrices. The C and R matrices can be determined from the point spread

function using the superposition principle. In order to determine the C matrix, assume $R = 0$, and vice versa.

In order to determine the C matrix, consider the C F product,

Figure 5.3 Graphical illustration of $C F = V$.

then,

$$C_{k,i} = \frac{V_{k,j}}{F_{i,j}} \quad k = 1, \dots, 8 \quad (5.2)$$

In order to determine the R matrix, consider the F R product:

Figure 5.4 Graphical illustration of $F R = V$.

then

$$R_{j,k} = \frac{V_{i,k}}{F_{i,j}} \quad k = 1, \dots, 8 \quad (5.3)$$

Using equations 5.2 and 5.3, one can determine the C and R matrices simply by applying one known force to all elements, one by one, along the main diagonal of the sensor, then measuring the outputs of the cells which are in the same row or in the same column as the excited diagonal element. In the case of an 8x8 sensor, this would require a total of 8 measurements. If the gain of each element in the sensor array is assumed to be unity, then all diagonal elements in the C and R matrices should be equal to unity also since the diagonal elements are due to the actual excitation, not cross talk. As a matter of fact, if there were no cross talk, then both C and R matrices would be equal to the identity matrix. In

reality, when an input is applied to element $[n,m]$, both the m -th diagonal element of the C matrix and the n -th diagonal element of the R matrix would contribute to the value of the $V_{n,m}$, doubling its value. This can be corrected by simply dividing the diagonal entries by 2 in both the C and R matrices. Therefore, the final formulation of the elements of the C and R matrices is as follows:

$$C_{k,i} = \begin{cases} \frac{V_{k,j}}{F_{i,j}} & \text{if } k \neq i \\ \frac{V_{k,j}}{2 F_{i,j}} & \text{if } k = i \end{cases} \quad k = 1, \dots, 8 \quad (5.4)$$

$$R_{j,k} = \begin{cases} \frac{V_{i,k}}{F_{i,j}} & \text{if } j \neq k \\ \frac{V_{i,k}}{2 F_{i,j}} & \text{if } j = k \end{cases} \quad k = 1, \dots, 8 \quad (5.5)$$

5.3 A Simplified Solution

The C and R matrices can be calculated off line using the techniques described in the previous section. Next, knowing the C and R matrices, the F matrix must be found for a given V matrix by solving equation 5.1 for F ,

which is not trivial. Although a full solution to this problem exists (and will be presented in the following section), a simplified solution can also be utilized with some sacrifice in the accuracy.

As indicated in Section 5.2, the row cross talk is significantly less than the column cross talk. When an input with a complicated geometry is presented to the sensor, this input will excite many elements and the cross talk at the output of the sensor will be mostly due to the column cross talk. An example of this was seen in Figure 4.2, where a washer shaped object was placed on the force sensor. Therefore it would be reasonable (and quite helpful) to ignore the row cross talk ($R=0$). Then equation 5.1 reduces into the following form:

$$C F = V \quad (5.6)$$

In this case, the C matrix is determined using equation 5.2 since the R matrix no longer applies and scaling of the diagonal entries is unnecessary. Then, the solution of equation 5.1 will be in the form:

$$F = C^{-1} V \quad (5.7)$$

where the C^{-1} matrix can be calculated off line and stored in order to save computation time during the run. Figure 5.5 shows the resulting F matrix in a three dimensional plot for the washer example of Figure 4.2. As can be seen from the figure, most of the cross talk is removed except that which is primarily due to row cross talk. The following section will describe how to reduce this residual cross talk further.

5.4 A Full Solution for Algebraic Cross Talk Reduction

In order to reduce the residual cross talk, which is mainly due to the row cross talk between elements, a full solution for equation 5.1 is needed. This section will describe how this solution can be obtained by diagonalization of the C and R matrices [Bartels, 1972; Golub, 1979].

Figure 5.5 Output of algebraic cross talk reduction method with column cross talk reduction alone for the washer example.

If we define:

N = size of the sensor array $[N,N]$

$\lambda_{C_1}, \lambda_{C_2}, \dots, \lambda_{C_N}$ = eigenvalues of the C matrix

$\lambda_{R_1}, \lambda_{R_2}, \dots, \lambda_{R_N}$ = eigenvalues of the R matrix

K = eigenvectors of the C matrix

L = eigenvectors of the R matrix

then

$$P = K^{-1} C K = \text{diag} (\lambda_{C_1}, \lambda_{C_2}, \dots, \lambda_{C_N}) \quad (5.8)$$

$$Q = L^{-1} R L = \text{diag} (\lambda_{R_1}, \lambda_{R_2}, \dots, \lambda_{R_N}) \quad (5.9)$$

Now, premultiplying equation 5.1 by K^{-1} and postmultiplying by L , the following can be obtained:

$$K^{-1} C F L + K^{-1} F R L = K^{-1} V L \quad (5.10)$$

Since $K K^{-1} = L L^{-1} = I$, where I is the identity matrix,

$$K^{-1} C K K^{-1} F L + K^{-1} F L L^{-1} R L = K^{-1} V L \quad (5.11)$$

Then, regrouping terms in the above equation,

$$(K^{-1} C K) (K^{-1} F L) + (K^{-1} F L) (L^{-1} R L) = K^{-1} V L \quad (5.12)$$

Next, we define

$$X = K^{-1} V L \quad (5.13)$$

$$Y = K^{-1} F L \quad (5.14)$$

Substituting equations 5.8, 5.9, 5.13 and 5.14 into 5.12, the following equation is obtained:

$$P Y + Y Q = X \quad (5.15)$$

Once the C and R matrices are determined, their eigenvalues and eigenvectors will also be known. Therefore, matrices K , K^{-1} , L , L^{-1} , P , and Q will be known and constant. Furthermore, the P and Q matrices are diagonal and their diagonal entries are the eigenvalues of the C and R matrices by definition. At any given time, the sensor output matrix V will be known via direct measurements, as will the X matrix. However, since the input F to the sensor array is unknown, Y will

be unknown also. Entries of the Y matrix can be determined using equation 5.15 as follows:

$$Y_{i,j} = \frac{X_{i,j}}{\lambda_{C_i} + \lambda_{R_j}} \quad i,j = 1, \dots, N \quad (5.16)$$

Based upon the above equations, a general algorithm to obtain F is as follows:

Determine C, R, K, K⁻¹, L, L⁻¹, P and Q matrices offline as described above. Except C and R, all other matrices must be stored to be used during run time. The steps for each cycle of the run time are

1. Read the sensor output matrix V
2. Calculate $X \leftarrow K^{-1} V L$
3. Solve Y using equation 5.16
4. Calculate $F \leftarrow K Y L^{-1}$

The eigenvalues and eigenvectors needed for this algorithm can be found using well established numerical analysis techniques. The power method for eigenvalue computation [Gerald, 1980] and inverse iteration method for eigenvector computation [Press, 1986] yielded good results for this case. Although both of these methods

are iterative in nature (therefore time consuming), they are needed only during the training session and do not increase the online computation time.

Application of the above described method to the washer example is shown in Figure 5.6. As seen from this figure, the full implementation of the algebraic cross talk reduction method reduced both row and column cross talk, making the washer shape even more easily recognizable.

Figure 5.6 Output of algebraic cross talk reduction method with full cross talk reduction for the washer example.

Chapter 6. EDGE AND SHAPE DETECTION METHODS

The location of objects in a given image can be determined using edge and shape detectors. Since edge and shape detectors have principally been developed for use with images containing a large number of pixels, they must be modified to fit the needs of small scale sensor array signal processing.

6.1 Edge and Gradient Detection

Edge and gradient detection is a well established image processing technique used extensively to extract object location information from visual images obtained via ordinary television cameras [Frei, 1977]. This method makes use of the fact that an object can be located in space if the positions of its edges are known. Once the edges are found, they form a wire-frame model of the object. The main problem is how to determine the locations of the edges, or how to detect edges of the object within a given image which can be either tactile or visual.

Edges are defined as sets of points where the sensor array output is discontinuous according to some measure. In the case of a tactile sensor, pressure created by the

object in contact with the sensor creates a change in sensor array output. If the two dimensional gradient of the sensor output is calculated, any given edge of the object within the image will be perpendicular to the gradient direction at that point. Once the edges are found from the gradient directions (where the magnitude of the gradient is required to exceed a given threshold to avoid false detection due to noise), they should form a closed area describing the area subtended by the object in contact.

6.2 3x3 Gradient Detection

Once an image is obtained by a sensor array (visual, tactile etc), it typically is in a discrete form which requires an approximation of the gradient, rather than the actual continuous gradient, to be calculated. This is typically done in image processing by using edge detection masks which estimate the gradient by a first order approximation [Pratt, 1978; Robinson, 1977]. Some relevant definitions and gradient masks are described below.

First we define some relative directions borrowed from navigational traditions:

Figure 6.1 Directional definitions.

In order to estimate the gradient at any image point, a gradient mask is centered on the element corresponding to that image point. The sensor array outputs are multiplied by the corresponding entries in the mask and the results are summed. (This process will be demonstrated in an example case on the following pages.) The sum, called the "voting number", is the response of the cell to the particular gradient mask. After all the masks have been sequentially applied to a particular cell, the mask which gives the maximum

absolute response determines the gradient magnitude and direction for that cell. It is necessary to use the absolute value of the responses since the gradient in the South-South direction is simply the negative of the gradient in the North-North direction. This is why only four masks are defined in Figure 6.2 although Figure 6.1 defines 8 gradient directions. The gradients in the West-West, South-West, South-South and South-East directions can be obtained from the negatives of the East-East, North-East, North-North and North-West gradients respectively. The following example details the operation of the gradient masks:

Assume the cell under investigation is at array position $[i,j]$, and its neighbors have the values shown in Figure 6.3.

↓ J

	0.25	0.50	0.25
I →	0.00	0.00	0.00
	-0.25	-0.50	-0.25

north-north
gradient detector

↓ J

	-0.25	0.00	0.25
I →	-0.50	0.00	0.50
	-0.25	0.00	0.25

east-east
gradient detector

	0.50	0.25	0.00
	0.25	0.00	-0.25
	0.00	-0.25	-0.50

north-west
gradient detector

	0.00	0.25	0.50
	-0.25	0.00	0.25
	-0.50	-0.25	0.00

north-east
gradient detector

Figure 6.2 3x3 gradient masks (edge detection masks).

j
↓

	8	8	8	8	8	8	
	8	8	8	8	8	8	
i →	4	4	4	4	4	4	
	0	0	0	0	0	0	
	0	0	0	0	0	0	

Figure 6.3 An example edge.

In Figure 6.3, it can be observed that row i coincides with the edge of an object which resides on the rows above row i . Application of the North-North gradient detection mask to element $[i, j]$ yields the following result:

8	8	8
4	4	4
0	0	0

x

0.25	0.50	0.25
0.00	0.00	0.00
-0.25	-0.50	-0.25

=

$$\begin{aligned}
 &8 \times 0.25 + 8 \times 0.50 + 8 \times 0.25 + 4 \times 0.00 + \\
 &4 \times 0.00 + 4 \times 0.00 - 0 \times 0.25 - \\
 &0 \times 0.50 - 0 \times 0.25 = 8.0
 \end{aligned}$$

where 8.0 is the voting number of the North-North gradient detector for element $[i,j]$. If the voting numbers for all the gradient directions shown in Figure 6.2 were calculated, it would be seen that the North-North detector gives the highest response among all the detectors. Therefore, at element $[i,j]$, the gradient is in the North-North direction with magnitude equal to 8.0. The edge, therefore, will be a horizontal edge (edges are perpendicular to the direction of the gradient).

If the sensor array used has a large number of cells (e.g. 512x512) and the object imaged covers a large number of cells in the sensor array (e.g. 50 or more), then the above described method can be used to detect the candidates for edge points. Once these candidate edges are found, they can be connected together to define the

shape of the object in the image. But, if the sensor array has a small number of cells (e.g. 8x8 for the tactile sensor array described in Chapter 2) and the objects imaged are small (e.g., 2 or 3 cells in extent), detecting the edges in a given image can be quite difficult since the resolution would be very low. This problem can be solved using two different approaches: introducing additional masks which detect shapes as well as edges, or by increasing the resolution of the image via image enhancement methods (described in detail in Section 6.4).

6.3 3x3 Shape Detection Masks

For an edge to be detected in an image by the edge detection method described in Section 6.2, the object must be at least two cells in extent. If the object dimension is less than two cells, its edges will not be detected by a 3x3 gradient detector. An object less than two cells across can be either a line or a single point which can be detected using the following 3x3 masks:

Figure 6.4 3x3 shape detection masks.

The elements in the edge and shape detection masks of Figures 6.2 and 6.4 are chosen such that, for a given mask, the sum of all nine elements is zero. Therefore,

if a mask is centered on an element where that element and all of its neighbors are excited by an object, all the masks will give zero output since there is close to a uniform excitation across all the elements.

A careful observation of the masks described in Figures 6.2 and 6.4 would also show that, for any mask, the sum of the positive entries is equal to unity. Also, the positive numbers in each mask are located in the form of the pattern it will detect. Therefore, a mask will give its maximum output when all of the sensor elements corresponding to its positive elements are excited but none of its negative elements. This also guarantees that for a given input pattern, only the mask exactly corresponding to that pattern will generate the maximum output since the input pattern will excite some negative entries in all the other masks and lower their outputs. It should be kept in mind that for edge detection, negative outputs are acceptable since they represent a gradient in the opposite direction (i.e. the edge is in the same direction as before but the object is on the opposite side of the edge). For the shape detectors, negative outputs mean cavities instead of objects (e.g. a negative output for a point detector indicates a hole in the object).

Using the 3x3 edge and shape detection masks described above, each cell of the sensor array can be classified as being a member of one of the following groups:

1. The edge of an object,
2. Part of a thin object which is long in one dimension but narrower in the other dimension (e.g. a line),
3. Part of a thin cavity (negative of group 2),
4. An element of a uniform excitation surface (either totally enclosed by or outside of an object),
5. A point object, or
6. A point cavity.

Figure 6.5 shows the results of applying the 3x3 edge and shape detection masks to the washer example described in Chapter 4. As can be seen from this figure, the washer shape is clearly identified by the masks.

The solid areas represent lines passing through a cell, arrows show the gradient direction.

Figure 6.5 Results of applying 3x3 masks to washer example.

6.4 2x2 Detectors

This section will describe 2x2 detectors which allow finer discrimination of small objects. 3x3 masks consider neighboring cells in every direction during classification of each cell. If the mask size is reduced to 2x2, however, corners of the cells rather than the cells themselves should be considered for identification in order to preserve the symmetry. In other words, to determine the relationship of a given point to its neighbors, the same amount of information must be considered on both sides of that point, in every direction, e.g., up-down, right-left, etc. 3x3 masks utilize a central element and one cell to either side for each of the four directions. Therefore, a symmetric distance is traveled from the center of the mask to each side. When the mask size is reduced to 2x2, one cell cannot be the center of the mask. Hence, the intersection of four cells is chosen to be the center of the mask, such that an equivalent distance is traveled in each direction from the center of each mask.

Each cell is divided into four quadrants when the 2x2 masks are placed on the common corners of four cells (see Figure 6.6). 2x2 masks identify one quadrant in each of these four cells, and each cell contains four

identifiable quadrants. Therefore, using 2×2 masks, it is possible to obtain $2N \times 2N$ of data from an $N \times N$ sensor, effectively doubling the resolution of the sensor.

There are only two possible outcomes from a quadrant of a cell identified by 2×2 masks; it is either a part of an object or not. Since each 2×2 mask covers four quadrants, $2^4 = 16$ is the total number of different 2×2 masks. 15 masks which can be used as 2×2 detection masks are listed in Figure 6.7. Again these masks were constructed by the symmetry principle. The mask in which all entries are zero is not included in Figure 6.7 since its outcome is always zero.

In Figures 6.2 and 6.4, a total of nine 3×3 masks were defined. For any given cell, all nine masks should be tested, requiring a total of $9 \times 9 = 81$ multiplications. For 2×2 masks, Figure 6.7 defines 15 masks, requiring a total of $4 \times 15 = 60$ multiplications for each element to be identified (since each 2×2 mask identifies four adjacent quadrants, it can be assumed that one 2×2 mask identifies one complete cell). Therefore, the use of 2×2 masks reduces computation by approximately 25 percent and effectively doubles the resolution of the image obtained from the sensor.

Nevertheless, performance of the 2x2 masks is too poor to be useful as long as the masks remain as described in Figure 6.7. The performance of the 2x2 masks can be improved by changing the mask coefficients.

Figure 6.6 A 2x2 detector placed on the common corner of elements [4,3], [4,4], [5,3] and [5,4].

Corner Detectors	<table><tr><td>1.00</td><td>-0.33</td></tr><tr><td>-0.33</td><td>-0.33</td></tr></table>	1.00	-0.33	-0.33	-0.33	<table><tr><td>-0.33</td><td>1.00</td></tr><tr><td>-0.33</td><td>-0.33</td></tr></table>	-0.33	1.00	-0.33	-0.33
	1.00	-0.33								
	-0.33	-0.33								
	-0.33	1.00								
-0.33	-0.33									
<table><tr><td>-0.33</td><td>-0.33</td></tr><tr><td>-0.33</td><td>1.00</td></tr></table>	-0.33	-0.33	-0.33	1.00	<table><tr><td>-0.33</td><td>-0.33</td></tr><tr><td>1.00</td><td>-0.33</td></tr></table>	-0.33	-0.33	1.00	-0.33	
-0.33	-0.33									
-0.33	1.00									
-0.33	-0.33									
1.00	-0.33									
Side Detectors	<table><tr><td>0.50</td><td>0.50</td></tr><tr><td>-0.50</td><td>-0.50</td></tr></table>	0.50	0.50	-0.50	-0.50	<table><tr><td>-0.50</td><td>0.50</td></tr><tr><td>-0.50</td><td>0.50</td></tr></table>	-0.50	0.50	-0.50	0.50
	0.50	0.50								
	-0.50	-0.50								
	-0.50	0.50								
-0.50	0.50									
<table><tr><td>-0.50</td><td>-0.50</td></tr><tr><td>0.50</td><td>0.50</td></tr></table>	-0.50	-0.50	0.50	0.50	<table><tr><td>0.50</td><td>-0.50</td></tr><tr><td>0.50</td><td>-0.50</td></tr></table>	0.50	-0.50	0.50	-0.50	
-0.50	-0.50									
0.50	0.50									
0.50	-0.50									
0.50	-0.50									
Diagonal Side Detectors	<table><tr><td>0.33</td><td>0.33</td></tr><tr><td>-1.00</td><td>0.33</td></tr></table>	0.33	0.33	-1.00	0.33	<table><tr><td>-1.00</td><td>0.33</td></tr><tr><td>0.33</td><td>0.33</td></tr></table>	-1.00	0.33	0.33	0.33
	0.33	0.33								
	-1.00	0.33								
	-1.00	0.33								
0.33	0.33									
<table><tr><td>0.33</td><td>-1.00</td></tr><tr><td>0.33</td><td>0.33</td></tr></table>	0.33	-1.00	0.33	0.33	<table><tr><td>0.33</td><td>0.33</td></tr><tr><td>0.33</td><td>-1.00</td></tr></table>	0.33	0.33	0.33	-1.00	
0.33	-1.00									
0.33	0.33									
0.33	0.33									
0.33	-1.00									
Diagonal Line Detectors	<table><tr><td>0.50</td><td>-0.50</td></tr><tr><td>-0.50</td><td>0.50</td></tr></table>	0.50	-0.50	-0.50	0.50	<table><tr><td>-0.50</td><td>0.50</td></tr><tr><td>0.50</td><td>-0.50</td></tr></table>	-0.50	0.50	0.50	-0.50
	0.50	-0.50								
-0.50	0.50									
-0.50	0.50									
0.50	-0.50									
Inside Detector	<table><tr><td>0.25</td><td>0.25</td></tr><tr><td>0.25</td><td>0.25</td></tr></table>	0.25	0.25	0.25	0.25					
	0.25	0.25								
0.25	0.25									

Figure 6.7 2x2 detectors.

6.5 Optimization of Mask Coefficients

For both 3x3 and 2x2 masks, the mask coefficients were chosen by intuition, even though some rules of thumb were used (such as all the positive values in a mask must add up to unity, 3x3 edge detectors approximate the Laplacian etc.). These rules made the masks sensitive to the patterns they should recognize. To improve the performance of the masks, one alternative is to choose their coefficients such that each mask not only will be very sensitive to the pattern that it should recognize but also will be very insensitive to all other patterns. This will guarantee that there will be large safety margins to decrease the chance of incorrect identifications. A similar method, known as Gram-Schmidt procedure, is described in the literature [Wozencraft et al.; Luenberger, 1984]. This section will describe how coefficient optimization is performed for 2x2 masks. A similar approach can be taken for 3x3 masks.

First a set of patterns should be chosen to be identified by the optimized masks. Since there are 15 (2^4-1) combinations for a 2x2 mask, there are $N = 15$ patterns that can be identified, and thus 15 recognition masks to be defined. However, some of these masks are simply rotations of each other; therefore, it is

sufficient to optimize one mask in each category and rotate these unique masks to obtain the others. Since there are only five unique patterns in Figure 6.7, the number of masks N_M is chosen to be 5. Since the goal of this optimization is to maximize the mask response to one particular pattern and minimize it for all the rest of the patterns, an objective function was defined for each mask as follows :

$$\max \omega_j = X_j M_j^T - \sum_{i \neq j}^N (X_i M_j^T)^2 \quad j = 1 \dots N_M \quad (6.1)$$

where N_M is the number of unique masks, N is the number of possible input patterns, M represents a mask, and X represents an input pattern to be classified. In particular, M_j is the mask which should only recognize input X_j . Therefore, it is desired that the objective function ω should show the $X_j M_j$ product as positive and all the others as negative terms where the goal is to maximize ω for each mask. A generic mask and input are as follows:

m_1	m_2
m_3	m_4

A generic mask

x_1	x_2
x_3	x_4

A generic input pattern

Figure 6.8 Numbering sequence used in 2x2 data sets and masks.

Five generic masks constructed for performance optimization are shown in Figure 6.9. Each one of these five masks is a representative of one of the detector types listed in Figure 6.7. If the sensor has similar cross-talk characteristics in both directions (horizontal and vertical), then some of the elements in each mask can be defined to be identical. For example, mask number 1 in Figure 6.9 detects only the input pattern where only the portion of the element corresponding to the top left corner of the mask is excited. In this case, the quadrant labelled a_1 gets the maximum output from the sensor, where quadrants b_1 get equally low responses since they are equally far from the source of excitation. Quadrant c_1 gets a lower response than both a_1 and b_1 . In general, all the masks have fewer than four coefficients to be determined; mask number 1 has only three. This decreases the number of variables to be

determined in each mask and increases the speed of computations.

A pattern set containing the 15 possible binary input patterns for 2x2 masks is used as the test data. In the set shown in Figure 6.10, the first five elements are the unique patterns to be detected by the masks. The rest of the patterns are rotations of the first five. Therefore, each mask should show a high response to only one of the patterns in the test set and should show low responses to all the others.

a_1	b_1
b_1	c_1

Mask number 1

Corner detector

a_2	a_2
b_2	b_2

Mask number 2

Side detector

a_3	b_3
c_3	a_3

Mask number 3

Diagonal side detector

a_4	b_4
b_4	a_4

Mask number 4

Diagonal line detector

a_5	a_5
a_5	a_5

Mask number 5

Inside detector

Figure 6.9 2x2 masks to be optimized.

Figure 6.10 Test data containing fifteen sets of 2x2 patterns.

In order to optimize the performance of the edge and shape detection masks, equation 6.1 is differentiated with respect to each mask element. Since the number of variables is less than four for each mask, fewer than four equations for each mask will give the optimum values for the mask elements. For example, for mask number 1, only three equations will be needed :

$$\frac{\partial \omega_1}{\partial a_1} = 0 \quad (6.2)$$

$$\frac{\partial \omega_1}{\partial b_1} = 0 \quad (6.3)$$

$$\frac{\partial \omega_1}{\partial c_1} = 0 \quad (6.4)$$

When a mask is applied to the corners of four cells of the data coming from the sensor array, each element of the mask is multiplied with the corresponding element in the incoming data set and the results of these multiplications are added in order to find the mask output. In the case of mask number 1, two elements of the 2x2 data set will be multiplied with the same element

of the mask, namely b_1 . The same result could be obtained by first summing the values of the two entries in the 2×2 data set corresponding to the positions of b_1 , multiplying that sum by b_1 and then adding the other two terms $a_1 x_1$ and $c_1 x_4$. Defining :

m_{jp} : mask j , variable p

x_{ik} : 2×2 test data i , element k , and

θ_{ip} : set containing the locations of element p in mask i ,

equation 6.1 can be rewritten as follows:

$$\begin{aligned} \max \omega_j = & \sum_{p=1}^{N_j} \left(\sum_{k \in \theta_{jp}} x_{jk} \right) m_{jp} \\ & - \sum_{i \neq j}^N \left[\sum_{p=1}^{N_j} \left(\sum_{k \in \theta_{jp}} x_{ik} \right) m_{jp} \right]^2 \end{aligned} \quad (6.5)$$

where,

N : number of 2×2 data in test data (15 in this case)

N_j : number of variables in mask j

Defining y_{ip} for each mask j , input i as follows :

$$y_{ip} = \sum_{k \in \theta_{jp}} x_{ik} \quad (6.6)$$

equation 6.5 can be rewritten as follows :

$$\max \omega_j = \sum_{p=1}^{N_j} y_{jp} m_{jp} - \sum_{l \neq j}^N \left[\sum_{p=1}^{N_j} y_{lp} m_{jp} \right]^2 \quad (6.7)$$

Differentiating ω_j with respect to m_{jq} , the following is obtained:

$$\frac{\partial \omega_j}{\partial m_q} = y_{jq} - 2 \sum_{l \neq j}^N \left[y_{lq} \left(\sum_{p=1}^{N_j} y_{lp} m_{jp} \right) \right] \quad (6.8)$$

$$\text{If } \frac{\partial \omega_j}{\partial m_q} = 0$$

then

$$Y_{jq} = 2 \sum_{i \neq j}^N \sum_{p=1}^{N_j} \left(Y_{iq} Y_{ip} m_{jp} \right) \quad (6.9)$$

or in matrix form

$$\begin{bmatrix} 2 \sum_{i \neq j}^N Y_{i1} Y_{i1} & \dots & 2 \sum_{i \neq j}^N Y_{i1} Y_{iN_j} \\ 2 \sum_{i \neq j}^N Y_{i2} Y_{i1} & \dots & 2 \sum_{i \neq j}^N Y_{i2} Y_{iN_j} \\ \vdots & \ddots & \vdots \\ 2 \sum_{i \neq j}^N Y_{iN_j} Y_{i1} & \dots & 2 \sum_{i \neq j}^N Y_{iN_j} Y_{iN_j} \end{bmatrix} \begin{bmatrix} m_{j1} \\ m_{j2} \\ \vdots \\ m_{jN_j} \end{bmatrix} = \begin{bmatrix} Y_{j1} \\ Y_{j2} \\ \vdots \\ Y_{jN_j} \end{bmatrix} \quad (6.10)$$

Solving equation 6.10 for m_{jp} for each mask, one can determine the optimum values of the mask entries. Optimized values of the 2x2 masks are listed in Figure 6.11.

0.2500	-0.0625
-0.0625	-0.0625

Mask number 1

Corner detector

0.2143	0.2143
-0.1429	-0.1429

Mask number 2

Side detector

0.1429	0.1429
-0.2143	0.1429

Mask number 3

Diagonal side detector

0.2143	-0.1429
-0.1429	0.2143

Mask number 4

Diagonal line detector

0.0625	0.0625
0.0625	0.0625

Mask number 5

Inside detector

Figure 6.11 Optimized 2x2 masks.

To evaluate the performance of this optimization method, 15 test patterns defined in Figure 6.10 were applied to the optimized masks and results are presented in Table 6.1. Each row of the table corresponds to one of the input patterns. The maximum output in each column occurred in the row where the column number is equal to the row number meaning that each mask showed the maximum output to the appropriate pattern. All masks showed low responses to test patterns 6 through 15, as expected.

Test Pattern	Mask 1 Response	Mask 2 Response	Mask 3 Response	Mask 4 Response	Mask 5 Response
1	0.2500	0.2143	0.1429	0.2143	0.0625
2	0.1875	0.4286	0.2857	0.0714	0.1250
3	0.1250	0.2857	0.4286	0.2857	0.1875
4	0.1875	0.0714	0.2857	0.4286	0.1250
5	0.0625	0.1429	0.2143	0.1429	0.2500
6	-0.0625	0.2143	0.1429	-0.1429	0.0625
7	-0.0625	-0.1429	0.1429	0.2143	0.0625
8	-0.0625	-0.1429	-0.2143	-0.1429	0.0625
9	-0.1250	0.0714	0.2857	0.0714	0.1250
10	-0.1250	-0.2857	-0.0714	0.0714	0.1250
11	0.1875	0.0714	-0.0714	0.0714	0.1250
12	-0.1875	-0.0714	0.0714	-0.0714	0.1875
13	0.1250	-0.0714	0.0714	0.2857	0.1875
14	0.1250	0.2857	0.0714	-0.0714	0.1875
15	-0.1250	0.0714	-0.0714	-0.2857	0.1250

Table 6.1 The responses of different masks to the test patterns.

6.6 Usage of 2x2 Masks

At run time, a full set of data is acquired from the sensor array. All of the masks are applied to each group of four cells with a common corner, and the mask giving the maximum output identifies the corner. When all the corners are identified, it is necessary to distinguish between the elements which are actually excited and those reflecting cross-talk effects. This can be done by thresholding. Since the threshold value depends on the total intensity of the imaged object, a variable threshold is needed for each set of data. This variable threshold is chosen by the method described in Section 4.1. In order to obtain the total output of the sensor array (total force for a force sensor array), outputs of the cells in which at least one of the quadrants is identified as actually excited by the variable threshold method must be summed. The results of applying 2x2 masks to the washer example described in Chapter 4 can be seen in Figure 6.12 where the method not only recovered the shape of the imaged object, but also enhanced the resolution of the image by providing four times more pixels.

Figure 6.12 Results of applying 2x2 masks to washer example.

Chapter 7. DECONVOLUTION METHOD

The deconvolution method reduces cross-talk and thereby determines the sensor input by deconvolving the sensor output from the sensor transfer function. This method assumes that the sensor is a two-dimensional linear system, such that the superposition theorem applies. This assumption is valid if either the sensor is actually linear, or the sensor output is linear after the reduction of non-uniformity via the methods described in Chapter 3.

7.1 Training of the Deconvolution Method

If the sensor array is assumed to be a linear system, then the relation between the input and the output of the sensor can be written as follows:

$$y[i,j] = t[i,j] * x[i,j] \quad (7.1)$$

where $x[i,j]$ is the input to the sensor array,

$t[i,j]$ is the sensor impulse response (or the point spread function),

$y[i,j]$ is the output of the sensor array (with or without the reduction of non-uniformity), and

* denotes the convolution operation in two dimensions.

Or, in the frequency domain, this equation can be written as:

$$Y[u,v] = T[u,v] X[u,v] \quad (7.2)$$

At any given time, all $y[i,j]$ entries are known and $Y[u,v]$ can be calculated from these values using well known Fourier Transform techniques (e.g., Fast Fourier Transform). If the values of $T[u,v]$ are known, both $x[i,j]$ and $X[u,v]$ can be obtained from equation 7.2, as rewritten:

$$X[u,v] = \frac{Y[u,v]}{T[u,v]} \quad (7.3)$$

Similarly, $T[u,v]$ can be obtained using equation 7.4:

$$T[u,v] = \frac{Y[u,v]}{X[u,v]} \quad (7.4)$$

Equation 7.4 shows that the transfer function of the sensor array, $T[u,v]$, is the ratio of the Fourier

Transform of the sensor output to the Fourier Transform of the sensor input. Theoretically, application of any one input pattern and measurement of the resultant sensor output should be sufficient to determine the sensor transfer function. Unfortunately, calculation of the sensor transfer function using equation 7.4 is not practical due to two reasons. First, it requires the exact input to all of the sensor elements to be known during the measurements, which may not be possible. Usually only the total input to the sensor array is known, not the individual entries. Second, when the magnitude of $X[u,v]$ is small, $T[u,v]$ tends to increase without bound. Even though this can be controlled by using techniques such as windowing or optimal filtering [Press, 1986], any noise in the measurement would yield erroneous values for the terms in the corresponding frequencies of the transfer function. Therefore, a more reliable way of determining the transfer function of the sensor array is needed.

During the determination of the transfer function of a sensor array, in general, only the total input is known. Since this provides only one known variable, and the complete solution of equation 7.4 requires N^2 variables, some *a priori* knowledge about the sensor

transfer function is needed. For the sensor array described in Chapter 2, a simple transfer function is assumed to be in the following form:

$$t(x,y) = \begin{cases} 1 & \text{if } x=y=0 \\ \frac{a}{x^2 + y^2} & \text{otherwise} \end{cases} \quad (7.5)$$

where the only unknown is the coefficient "a". Therefore, at any given time, the value of this coefficient can be determined from the only known quantity, the total input to the sensor array.

To determine the value of "a," equation 7.2 is rewritten after substituting both u and v with zero,

$$Y[0,0] = T[0,0] X[0,0] \quad (7.6)$$

All the terms in the above equation are the DC values of the functions. Therefore,

$$Y[0,0] = \sum_{i=1}^N \sum_{j=1}^N y[i,j] \quad (7.7)$$

$$X[0,0] = \sum_{i=1}^N \sum_{j=1}^N x[i,j] \quad (7.8)$$

$$T[0,0] = \sum_{i=-N}^N \sum_{j=-N}^N \frac{a}{i+j} + 1 \quad (7.9)$$

(i and j not equal to zero simultaneously)

Finally, combining equations 7.4 and 7.9, the value of coefficient "a" is given as follows:

$$a = \frac{1}{\left(\sum_{i=-N}^N \sum_{j=-N}^N \frac{1}{i+j} \right)} \left(\frac{\sum_{i=1}^N \sum_{j=1}^N y[i,j]}{\sum_{i=1}^N \sum_{j=1}^N x[i,j]} - 1 \right) \quad (7.10)$$

7.2 Execution of the Deconvolution Method

The transfer function of an NxN sensor array can be calculated by first calculating the value of coefficient "a" using equation 7.10 and then the transfer function itself using equation 7.5. This transfer function can then be stored and used during the online implementation of the deconvolution method, which is given as follows:

1. Measure the output of the sensor array,
 $Y[i,j], \quad i,j = 1, \dots, N$
2. Convert $y[i,j]$ into frequency domain values,
 $Y[u,v],$
3. Estimate $X[u,v]$ using equation 7.3, then
4. Convert $X[u,v]$ into estimates of the actual input quantities, $x[i,j],$ via Inverse Fourier Transform.

The results of applying this method to the washer example described in Chapter 4 can be seen in Figure 7.1. As can be seen, use of the deconvolution method reduces the cross-talk and makes the washer shape more recognizable.

Figure 7.1 Results of applying deconvolution method to the washer example.

Chapter 8. SIGNAL PROCESSING WITH ARTIFICIAL NEURAL NETWORKS

Artificial neural networks are a relatively new approach in the pattern recognition field. They "learn" the relation between sets of inputs and outputs and are able to extrapolate from the results of their training. Artificial neural networks can be realized either by using analog devices such as operational amplifiers (op-amps) and resistor matrices or by computer simulation. In either case, their arrangement is similar to the biological neural networks found in human and animal nervous systems. This chapter first describes the structure and operation of neural networks, and then explains how they can be used as signal processors.

8.1 Structure of Artificial Neural Networks

Artificial neural networks consist of neurons and the connections between them. A *neuron* is a unit which generates a single output that is based upon multiple inputs. In general, the inputs are summed and the output is a function of this sum. Different implementations use different functions to determine the output from the input, and some examples are presented in Figure 8.1.

Figure 8.1 Transfer function of a neuron.

Inputs to a neuron are provided by connections from the outputs of other neurons or from external inputs. A *connection* is a unidirectional link which multiplies a neuron output or an external input by a gain constant (which differs both in magnitude and sign for each connection) and presents the result as an input to another neuron. A connection with a positive gain imitates an excitatory synapse and a connection with a negative gain imitates an inhibitory synapse. The output of one neuron can be supplied to many neurons via connections.

The neurons and multiple connections between them form a neural network. A neural network has at least one (usually more than one) input and at least one output. In an analog circuit implementation of a neural network, op-amps with proper gain (linear or non-linear) and dual output represent neurons and resistor matrices provide the desired connections between neurons. In a digital computer simulation, both neurons and connections are simulated in the computer, requiring no external hardware.

8.2 Associated Memories

The simplest form of an artificial neural network is an associated memory. An associated memory is a neural network in which any neuron can be connected to any other neuron in the network. All neurons have binary outputs which are selected by simple thresholding of the input, as shown in Figure 8.1.a. The output of the associated memory consists of the output of every neuron, therefore the network also has only binary outputs. Gain constants of the connections between neurons are chosen according to the statistical cross-correlation of the desired outputs of the neurons at either end of each connection. A positive correlation between two neurons (i.e., when two neurons have synchronized outputs) indicates an excitatory synapse; a negative correlation indicates an inhibitory synapse.

Connection values for associated memories are determined offline using the set of patterns which will be remembered by the network. Once the connection values are determined statistically, the training patterns are memorized by the network's connections, and this data is no longer necessary for normal operation of the network. If any one of the training set patterns is distorted and presented to the network, the network begins to change

the output states of its neurons and eventually converges to one of the members of the training set. If only a few of the inputs are distorted, the connections from neurons with undistorted inputs will correct the states of the distorted neurons and the entire network will stabilize and converge to the pattern which is the undistorted version of the input. This resembles normal recollection by the human mind [Hopfield, 1987]. However, if many of the inputs are distorted, then the network may converge to an incorrect pattern. Also, if a novel pattern is presented, the network will converge to one of the patterns in its training set, which is a major drawback of the use of associated memories.

As mentioned in Section 8.1, neural networks can be either implemented by analog circuitry or simulated in a digital computer. For both implementations, the training phase for an associated memory must be performed on a computer to determine the connection values. These connection values will constitute the resistor matrix in the analog hardware implementation. Convergence time of the circuit will be much shorter than that of the computer simulation since the convergence occurs immediately after the input values are loaded. In a

digital computer simulation, however, convergence occurs only after an iteration paradigm.

It is possible to use an associated memory as a signal processor for a sensor array. This can be done by directly connecting each element of the array to a neuron, and determining the connection values using a set of training data which consists of known inputs to the sensor array and their resulting outputs. Then, when one of the input patterns is presented to the sensor during actual usage, the associated memory will compensate for distortions in the sensor output (e.g., noise, cross-talk, etc.) and converge to the most similar pattern in the training set. Therefore, if there is only a limited number of input patterns used with the sensor, then associated memories provide efficient signal processing. Their training phase consists only of calculation of correlation coefficients, and run time is a simple iterative process which is nothing more than additions and multiplications.

8.3 Feed Forward Artificial Neural Networks

Frequently, the number and form of sensor input patterns are not known *a priori*, and it is desired to have a range of outputs from a sensor array rather than

zeros and ones alone. Therefore, a more sophisticated neural network is often necessary.

Feed forward neural networks are a modified version of associated memories, intended to eliminate the disadvantages of the latter. They are formed of a passive layer of inputs and two or three layers of neurons known as active layers, with no connections between neurons in the same layer. Furthermore, there is no feedback from the layers above [Rumelhart 1986]. In other words, every neuron is unidirectionally connected only to neurons on the layer above. Inputs form a passive layer below the layers of neurons. A schematic of this type of network can be seen in Figure 8.2.

Similar to associated memories, feed forward networks also store their knowledge of input-output relationships in their connections. Associated memories remember only a certain number of patterns and attempt to recall one of them when an input pattern is presented. Feed forward networks, however, do not retain any of the training set elements, but instead learn an input output relationship which is encoded in their connections. If the input output relation is learned correctly, then they not only can deal with any pattern presented previously,

Figure 8.2 A two layer neural network.

but also novel patterns never seen before. If the networks do not yield the proper output to a novel input, then they must be retrained with a larger set of training patterns.

The physical connections of a feed forward network are similar to the connections of an associated memory. Basically, they convey the output of one neuron to another, multiplied by some gain constant. However, the neurons of a feed forward network are quite different from the neurons of an associated memory. Neurons in a feed forward network can provide outputs in the range of zero to one. They have a non-linear sigmoidal transfer function as seen in Figure 8.1.b.

The definition for the sigmoidal function is as follows:

$$\text{sigmoidal}(x) = \frac{1}{1 + \exp(-x)} \quad (8.1)$$

where x is the sum of all the inputs to a neuron plus a set threshold value. If ω_{ij} represents the connection from neuron i to neuron j , θ_i represents the threshold for neuron i , and o_i represents the output of neuron i , then:

$$o_i = \frac{1}{1 + \exp \left[- \left(\sum_j \omega_{ij} o_j + \theta_i \right) \right]} \quad (8.2)$$

To describe the training phase of a feed forward network of two active layers and one passive input layer, such as that shown in Figure 8.2, we define the following:

- i : index for the inputs in the lowest layer,
- j : index for the hidden layer neurons,
- k : index for the output layer neurons,
- p : index for training set patterns,
- t_{pk} : target output for pattern p , output k ,
- and
- o_{pk} : actual network output for pattern p ,
- output k ,

then, for each pattern, the system error is:

$$E_p = \frac{1}{2} \sum_k (t_{pk} - o_{pk})^2 \quad (8.3)$$

and the average system error is given by:

$$E = \frac{1}{2P} \sum_p \sum_k (t_{pk} - o_{pk})^2 \quad (8.4)$$

When feed forward networks learn input output relationships, the difference between target output levels and the network output is minimal, as is the average system error. This fact can be used to determine the optimal values of the connections, ω_{ij} , ω_{jk} , and the threshold values θ_j , θ_k iteratively during training. This method is referred to as the *Generalized Delta Rule* and is summarized in Appendix B. This method tries to minimize the total system error by first varying the values of ω and θ for the output layer and repeating the same procedure for the lower layers by back propagation of the error from the layers above [Rumelhart, et al., 1986]. Details of the method of back propagation of error can be found in Appendix B.

Like many other iterative computational methods, convergence time of the Generalized Delta Rule is a function of the initial estimates. Assume a three layer feed forward network with N neurons in the input and hidden layers and a single neuron at the output layer. There will be $2N^2+N$ connections and $2N+1$ thresholds which need to be determined. In the case of an 8×8 sensor array, N will be 64, meaning that there are 8385 unknowns to be determined iteratively. Starting from initial values which are not reasonably close to the optimal

values would be quite costly in terms of computational time. Furthermore, if the initial values are not close to the optimal values, then the network generally converges to one of the local minima instead of the global minimum. Therefore, as a first step, a method to determine reasonable training initial values must be developed.

Of primary concern during the selection of the initial values is the selection of values which will not saturate the network. Since the neurons do not provide outputs outside the range 0-1, the input and output to the network should also be limited to the range between 0 and 1. An output equal to zero can only be obtained from a neuron if its input approaches negative infinity. Similarly, the input to a neuron should approach positive infinity for its output to equal unity. To initialize the network and drive it to a reasonable working range, the output of the neurons can be kept between 0.25 and 0.75. In this case, using equation 8.2, the maximum and minimum inputs to a neuron are found to be $\ln 3$ and $-\ln 3$, respectively.

Studies have shown that training of a neural network begun with identical connection values will never

converge [Pao, 1988]. Therefore, at least some of the connection values between each neuron layer should be varied. One way to assign values is to group the connections according to the indices of the neurons at each end. If a connection, for example ω_A , lies between two neurons with the same index numbers, then it will transmit the excitation for the input element with the same indices. Otherwise, the connection will transmit signals due to the effect of the cross-talk between the elements. If the indices of the neurons represent neighboring elements, then the cross-talk would be significant, and the gain of the connection (ω_B) will be large. If the corresponding elements are not adjacent, then the cross-talk would be weaker, and the connection (ω_C) would be smaller. Connections in each group (ω_A , ω_B , ω_C) were assigned the same initial value.

For any neuron, the input is the sum of the signals carried by the connections from the previous layers and the threshold input, which is fixed. For an $N \times N$ sensor array, for any neuron, there can be only one ω_A , eight ω_B and $(N^2 - 9) \omega_C$ connections providing input. Defining ω_A , ω_B , and ω_C as above, the initial values for the connections between the layers can be calculated as explained below.

Assume that the maximum input to any cell in the sensor array will be $x_{\max}$. If all of the cells in the sensor array were excited, then the total input of neuron $[i,j]$ in the input layer would be:

$$x_{\max} \omega_A + 8 x_{\max} \omega_B + (N^2-9) x_{\max} \omega_C + \theta = + \ln 3 \quad (8.5)$$

If only the sensor element which has the same indices as neuron $[i,j]$ was excited, then the total input of that neuron would be:

$$x_{\max} \omega_A + 8 (0) \omega_B + (N^2-9) (0) \omega_C + \theta = + \ln 3 \quad (8.6)$$

If only a sensor element which is far from the neuron considered in the input layer was excited, then:

$$(0) \omega_A + 8 (0) \omega_B + (N^2-10) (0) \omega_C + \omega_C x_{\max} + \theta = - \ln 3 \quad (8.7)$$

If none of the sensor elements was excited, then

$$(0) \omega_A + 8 (0) \omega_B + (N^2-9) (0) \omega_C + \theta = - \ln 3 \quad (8.8)$$

Equations 8.4 through 8.8 can be written in matrix form as shown below:

$$\begin{bmatrix} x_{\max} & 8 x_{\max} & (N^2-9) x_{\max} & 1 \\ x_{\max} & 0 & 0 & 1 \\ 0 & 0 & x_{\max} & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \omega_A \\ \omega_B \\ \omega_C \\ \theta \end{bmatrix} = \begin{bmatrix} \ln 3 \\ \ln 3 \\ -\ln 3 \\ -\ln 3 \end{bmatrix} \quad (8.9)$$

A similar analysis can be carried out for the connections between the input and hidden layer neurons. Connections between the network input and the input layer neurons were fed by the sensor array output which varied between zero and $x_{\max}$. The connections between the input layer and the hidden layer were fed by the input layer neurons which can provide outputs between 0.25 and 0.75. Substituting 0.25 for zero and 0.75 for $x_{\max}$, the following set of equations can be obtained for the connections between the input and hidden layer neurons:

$$\begin{bmatrix} 0.75 & 8(0.75) & (N^2-9)(0.75) & 1 \\ 0.75 & 8(0.25) & (N^2-9)(0.25) & 1 \\ 0.25 & 8(0.25) & (N^2-10)(0.25) + 0.75 & 1 \\ 0.25 & 8(0.25) & (N^2-9)(0.25) & 1 \end{bmatrix} \begin{bmatrix} \omega_A \\ \omega_B \\ \omega_C \\ \theta \end{bmatrix} = \begin{bmatrix} \ln 3 \\ \ln 3 \\ -\ln 3 \\ -\ln 3 \end{bmatrix}$$

(8.10)

For the connections between the hidden and output layers, initial value calculations will be slightly different since there is only one output neuron, and N^2 connections which are all of the same type. In this case, there is a total of two unknowns, ω and θ , so two equations would be sufficient. Therefore, instead of considering four different cases as with the input and hidden layers, only two cases were considered. In the first case, all of the cells in the sensor array were excited, and the input to the output neuron was:

$$N^2 (0.75) \omega + \theta = + \ln 3 \quad (8.11)$$

In the second case, none of the elements in the sensor array were excited, resulting in an input of:

$$N^2 (0.25) \omega + \theta = - \ln 3 \quad (8.12)$$

Rewriting equations 8.11 and 8.12 in matrix form,

$$\begin{bmatrix} N^2 (0.75) & 1 \\ N^2 (0.25) & 1 \end{bmatrix} \begin{bmatrix} \omega \\ \theta \end{bmatrix} = \begin{bmatrix} \ln 3 \\ -\ln 3 \end{bmatrix} \quad (8.13)$$

Initial values for the neural network training can be obtained by solving equations 8.9, 8.10 and 8.13 for θ and ω values. In order to avoid stalling the convergence of the network during training due to identical connection or threshold values, each actual initial value should be chosen as the calculated value plus a small random number. Once these values are determined, the neural network can be trained to learn the relationship between the total sensor input and the outputs of the sensor array elements, by providing a set of example cases which consist of known inputs and outputs of the sensor array.

8.4 Implementing the Feed Forward Signal Processor

Despite the fact that training of feed forward neural networks is quite difficult and computationally intensive, these networks give quick, straightforward

results at run time. With computer simulations, only routine calculations with no iterations are needed. Since all the neurons run in parallel, analog implementations of these structures would be even faster.

A feed forward neural network was trained for the same sensor array that was used for the washer example in Chapter 4. After the training, the same output pattern seen in Figure 4.2 was presented to the trained neural network signal processor. Figure 8.3 shows the output of this signal processor. As can be seen, the washer shape became apparent after processing by the trained neural network.

Figure 8.3 Output of neural network signal processor for the washer input.

Chapter 9. EXPERIMENTS

Chapters 4 through 8 described various signal processing techniques and demonstrated their performance on specific examples. In general, a signal processing method is expected to work under all conditions, not just under a specific set of circumstances. In order to determine whether a method works for different input configurations or which method works better in any given case, a set of experiments that represents a range of conditions likely to be encountered has been performed. This chapter describes the approach used to investigate the signal processing methods described in Chapters 2 through 8.

The entire sensor array was initially calibrated to compensate for the non-linear and non-uniform sensitivity characteristics of the sensor elements. Each element was excited at five different force levels by applying five different weights. Since the sensors' sensitivity decreases as the applied force increases, 0, 50, 100, 200 and 400 gram weights were chosen so that successive force levels were doubled. Software written to collect the data scanned the sensor array nine times after a weight

was applied and used a median filter¹ to eliminate any outlying points. A hyperbolic curve was fit to the collected data to obtain a calibration curve for each element (Section 3.3). These curves were used to estimate the input of each element from its output.

In order to compare the five different signal processing methods, input patterns were generated by forcing differently shaped objects against the surface of the force sensors. These test objects were intended to imitate real life conditions where a human finger might come into contact with other objects. Ideally, all possible inputs should be considered; however, any experimental attempt must be limited to a small, finite set in that domain. Once a certain number of objects are machined, however, many input patterns can be generated by simple rotations of these objects.

Twenty-two objects, which varied both in size and shape, were fabricated. It is not likely that more than half of the cells in the sensor would be excited at the

¹ A Median Filter with length 9 sorts the list of data in ascending order and then chooses the 5th entry in the sorted list.

same time due to the curvature of the human finger. Therefore, objects larger than 32 cells in size are rarely necessary for testing. The 22 object test set is illustrated in Appendix C. As can be seen, the set contains objects with different areas and dimensions as well as different shapes such as separate parallel lines, connected parallel lines and objects with cavities. By changing the orientation and position of the objects with respect to the sensor, 200 different test cases were created.

Another variable in the application of an input to the sensor array is the magnitude of the total force. The force applied to a test object passes entirely to the sensor, but, since there is no guarantee that the test object will distribute the force uniformly, there is no way of knowing how much force is seen by any given cell in the sensor array. However, as a result of an attempt to ensure a uniform force distribution, objects were constructed such that the force could be applied through their centers of mass. For example, an object shaped like a square frame with a cavity in the middle was forced at its center point by placing a glass microscope slide over it. The objects were coated with a 0.5 mm thick silicone elastomer sheet to prevent damage to

unpackaged sensors. The resulting non-uniform force distribution is actually closer to the real life situation, since during normal operation of the sensor (worn on a finger), the force distribution will most likely not be uniform.

When an object is applied to the sensor array at a high force level, some of the cells may experience three or four times the average force exerted over the cells if the force distribution is very uneven. Under these circumstances, the pressure level should be kept as low as possible in order not to damage the sensor. Since the sensor is more sensitive at the low end of the input force range, operating at a low pressure level also reduces the interpolation and quantization errors due to the finite word length of the analog-to-digital converter used for data collection. On the other hand, when a large object is applied with a low force level, most of the cells will see less than the applied pressure², and the signal to noise ratio will increase causing the signal to be lost in the noise. Therefore, during the experiments, a constant pressure was maintained by

² Pressure was defined as the force per cell in the sensor array.

applying a total force directly proportional to the area of the object. To examine the effects of the total force, two pressure levels were used during the data collection: 25 and 50 grams/cell. A total of eight sensors were tested with these two pressure levels using the 200 element test set, for a total of 16 experiments.

During the actual data collection process, each object was placed on the sensor using a computer display of both the desired and the apparent position of the object on the sensor, to prevent malpositioning. Once the object was placed in the predetermined location and orientation, data were collected by scanning the sensor array nine times as in the calibration procedure described earlier in this chapter. After scanning, the output of each element was median filtered to reduce the measurement noise. Data were saved in raw form to be later converted into mass estimations via interpolation from the calibration curves. These mass estimations were used to evaluate the various signal processing methods.

Two test sensors were selected from each of four different fabrication batches. Even though all sensors had similar structures, they differed in some way, such as gap width between electrodes, type of silicone

elastomer used between capacitive electrodes, etc. As a result of these manufacturing differences, the sensors' responses to similar inputs were different. For example, some sensors consistently underestimated the total force while others overestimated the total applied force. These variations in sensor characteristics allowed a more general comparison of the performance of the signal processing methods developed.

To determine the repeatability of the data from the application of the test objects to one of the force sensor arrays, ten test cases randomly chosen from the 200 element test set were applied ten times to one of the sensor arrays. The normalized sum of the outputs of all 64 elements of the sensor array is displayed in Figure 9.1. This repeatability study showed that the data collected was reproducible within 12 percent³.

As was described in Chapters 4 through 8, each of the signal processing methods require training data. During the training, some sensor dependent parameters had

³ Repeatability was calculated as the mean value of the interquartile ranges displayed in Figure 9.1 (i.e., the mean value of the length of the ten boxes).

Figure 9.1 Results of the repeatability study obtained with sensor A6 by application of ten objects ten times to the sensor surface. Outputs of the sensor elements were added and scaled for each object such that the median of each trial would be equal to 100. (Object numbers shown on the abscissa are listed in Appendix C.)

to be determined for each signal processing method. Training for the variable threshold method consisted of the determination of the optimum number of bins in the histograms used. For the algebraic cross talk reduction method, eigenvalues and eigenvectors of the cross-talk matrices were needed before the method could be implemented. For the 2x2 edge and shape detection method, mask coefficients had to be determined during the training. In the case of deconvolution via FFT, one must first determine the transfer function of the sensor array. Finally, for the artificial neural networks, connection values between neurons had to be found. All these unknowns were determined before the signal processing methods could be used to estimate the actual input of the sensor array from its outputs.

Each of the twenty-two differently shaped objects was used at a different frequency in the test set. For example, an object which was 8x8 in size could be applied to the sensor only in one orientation (exactly aligned on the sensor). On the other hand, an object which was much smaller could be applied to the sensor in many locations and orientations. Therefore, the frequency with which an object was applied to the sensor varied from one to sixteen.

One hundred fifty of the 200 data points in the test data sets were arbitrarily chosen as the training data. The training data set contained at least one case for each of the shapes. The necessary parameters for all methods were determined using this training data set. After the training phase, raw data were processed by each method using the parameters determined during the training phase, the results of which were estimations of the sensor input. These estimations were stored in the computer for later comparison.

Chapter 10. RESULTS

This chapter presents the results obtained from the analysis of the comparative study described in Chapter 9. The performance of each method was assessed using sixteen data sets which were processed by each of the signal processing methods.

10.1 Relative Error of Total Force Determination

To evaluate the performance of various signal processors, a relative error term was defined as follows:

$$\epsilon = \frac{F_E - F_A}{F_E} \times 100 \% \quad (10.1)$$

where

ϵ : relative error

F_E : total estimated force

F_A : actual applied force

Positive relative errors indicate that the sensor array overestimated the total applied force, where negative relative errors indicate underestimation.

Since all objects were not used an equal number of times in the test data set, a fair comparison can be obtained only if the results are combined for each object. This reduces the size of the data set from 200 to 22 elements. Table 10.1 shows the relative errors for the raw data and data processed by each of the signal processing methods for each object using sensor A6, with pressure equal to 50 grams/cell. The entries in the table are the median of the relative error followed by the interquartile range¹ of the relative error values for each shape. Median and interquartile range were used instead of the mean and standard deviation since the former are independent of the statistical distribution of the error values. The last row of Table 10.1 contains the overall error values which are the median and interquartile range for all the data listed in each column. As can be seen from this table, the overall error for the raw data is negative, meaning that sensor A6 in general underestimated the total applied force.

Approximately half of the force sensor arrays underestimated the total force. In those cases,

1 Interquartile range is defined as the difference between the 25th and 75th percentiles in a given distribution.

Shape Number (Area)	RELATIVE ERRORS (%) (Median Inter Quartile Range)						
	RAW	VTM	ALG	2x2	FFT	ANN (2,400)	ANN (50,000)
1- 1	29.0 116	15.0 68	26.0 468	-38.0 50	-5.0 128	-100 0.0	-14.0 94
2- 4	-9.8 16	-23.3 11	-6.5 93	-28.8 10	12.5 22	-1.3 99	0.5 1.8
3- 9	-12.2 4.6	-22.9 6.7	-4.4 47	-24.4 5.2	59.8 16	7.3 9.7	0.2 1.9
4-16	-18.1 7.1	-29.0 11	-5.3 38	-24.1 6.3	77.0 20	-3.3 5.1	0.0 0.1
5-25	-23.1 7.2	-30.7 12	-8.3 41	-28.3 9.9	84.2 14	-2.0 9.2	0.0 5.5
6-36	-15.0 1.3	-23.7 8.6	3.9 1.9	-23.2 18	118.6 5.8	-0.2 1.1	0.0 0.0
7-49	-15.5 2.1	-34.2 15	7.1 17	-25.6 5.5	119.2 6.1	0.2 0.8	0.0 0.9
8-64	-12.6 0.0	-17.9 0.0	10.7 0.0	-15.3 0.0	126.6 0.0	-0.3 0.0	0.0 0.0
9- 4	-16.0 19	-24.0 26	-13.5 68	-46.5 12	-4.5 28	-8.5 36	1.5 2.0
10- 8	-27.3 12	-34.5 5.8	-20.3 151	-30.3 9.3	1.5 7.3	-0.3 30	0.0 0.3
11-16	-29.6 7.8	-32.8 13	-17.4 13	-34.9 4.9	44.5 17	-0.6 2.4	0.0 0.1
12-32	-21.1 16	-26.5 7.8	-2.3 71	-23.4 15	89.6 48	0.3 2.7	0.0 1.8
13-48	-19.2 4.7	-33.5 30	1.7 24	-20.0 4.5	105.7 12	-2.8 5.1	-4.5 8.9
14-12	-15.3 8.2	-28.0 5.7	1.5 35	-24.6 15	95.3 24	10.4 10	0.0 0.2
15-32	-12.6 4.9	-24.8 11	8.3 19	-19.6 5.9	123.3 16	-0.1 1.0	0.0 0.0
16- 8	-14.5 16	-37.8 30	-1.5 104	-46.8 20	73.0 46	-2.0 10	0.0 0.1
17-24	-13.9 25	-36.1 29	2.6 107	-25.3 27	110.9 74	1.1 2.8	0.0 0.0
18- 9	-33.8 12	-44.0 11	-28.0 46	-51.9 34	13.6 27	-5.0 7.6	-0.2 0.7
19-12	-49.1 23	-62.1 23	-46.7 72	-65.7 20	-25.6 53	-4.0 25	-0.2 1.8
20-12	-24.3 12	-38.1 15	-14.5 57	-32.2 23	57.5 44	3.6 11	0.7 1.7
21- 8	-26.3 13	-38.8 19	-20.0 22	-53.8 20	16.8 19	9.5 27	0.3 1.5
22-24	-33.3 11	-50.7 19	-17.7 42	-57.4 27	57.9 36	-0.8 4.0	0.0 1.0
Overall	-17.1 12	-31.7 14	-4.8 100	-28.5 22	66.4 92	-0.5 3.1	0.0 0.0

VTM: Variable Threshold Method
 ALG: Algebraic Cross-Talk Reduction
 2x2: 2x2 Edge and Shape Detection
 FFT: Deconvolution Method Using FFT
 ANN: Artificial Neural Network

* Numbers in paranthesis are the number of iterations performed during traing of ANN.

Table 10.1 Relative error values for sensor A6,
 pressure = 50 grams/cell.

cross-talk between sensor elements remained a problem, but was negative at times and positive at others (see Table 10.2 and Figure 10.1).

Sensor Number	Cross-Talk (%)				Overall Relative Error	
	Min	Mean	Max	Abs	Pressure= 25 g/cell	Pressure= 50 g/cell
05	-2.32	1.28	6.38	7.01	25:29	-3:16
10	-2.46	0.95	6.23	6.81	-3:19	-18:11
21	-15.50	-3.06	6.71	18.32	7:27	-4:15
22	-6.66	1.08	11.39	14.55	26:36	17:19
36	-1.79	1.53	8.63	8.78	-1:15	8:24
37	-2.04	1.12	7.08	7.51	-6:12	-3:20
A5	-4.79	-0.32	5.88	7.91	-7:12	-4:12
A6	-5.62	-2.35	2.48	6.72	-30:26	-17:12

Table 10.2 Cross talk and relative error values for sensors used in the comparative study of signal processing techniques. Cross talk values were calculated using equations 10.2 a,b,c and d.

The cross-talk values in Table 10.2 were calculated from the calibration data of the sensor arrays. During calibration, each element in each sensor array was excited individually and the mean cross talk was

Figure 10.1 An example for negative cross-talk.

calculated as the ratio of the mean output of the eight neighboring cells to the output of the excited cell.

$$\text{mean cross-talk} = \frac{\text{mean value of eight adjacent neighbors}}{\text{voltage output of the excited element}} \quad (10.2a)$$

Minimum and maximum cross-talk values can be calculated similarly, as well as the absolute value.

$$\text{minimum cross-talk} = \frac{\text{minimum value of eight neighbors}}{\text{voltage output of the excited element}} \quad (10.2b)$$

$$\text{maximum cross-talk} = \frac{\text{maximum value of eight neighbors}}{\text{voltage output of the excited element}} \quad (10.2c)$$

$$\text{absolute value} = \frac{\text{mean absolute value of eight neighbors}}{\text{voltage output of the excited element}} \quad (10.2d)$$

As can be seen from Table 10.2, negative cross-talk was as significant as positive cross-talk. As a result of this, three of the eight sensors had negative mean cross-talk characteristics. During the excitation of any sensor element, if the force probe was not aligned

precisely with the element, then at least one of the neighboring elements experienced some of the force from the probe, which would appear as positive cross-talk. Therefore, almost all the sensor arrays showed some amount of negative cross-talk as well as positive cross-talk as shown in Table 10.2. When this negative cross-talk was significant, the sensor arrays tended to underestimate the total force. The cross-talk values shown in Table 10.2 were obtained using a point source. When larger objects were applied to the sensor, underestimation was even more pronounced.

Application of pressure to any of the sensor elements would decrease the distance between the sensor substrates and appear as a positive force. In order for negative cross-talk to exist, the forces measured from elements adjacent to the excited element must be negative, meaning that the distance between substrates of the sensor array is increased. This phenomenon may possibly be explained using beam theory. The two substrates of the sensor array are separated by thin strips of compliant dielectric. When a force is applied to one element, the portion of the top substrate under that element approaches the bottom substrate (see Figure 10.2). The portions of the top and bottom substrates

adjacent to the excited element separate from each other, generating negative cross-talk. However, when the applied pressure is sufficient to compress the dielectric supports, the substrates of the neighboring elements approach each other as well, generating positive cross-talk. According to this explanation, the sensor underestimates the input under localized low pressures. As the applied pressure is increased, eventually the sensor will overestimate.

Table 10.3 shows the minimum, mean, maximum and absolute cross talk values calculated for sensor A6 using different force levels. As the applied force increased, the minimum (negative) cross talk decreased. These data support the hypothesis in the paragraph above.

Single Cell Force (grams)	CROSS TALK (%)			
	Min	Mean	Max	Abs
50	- 10.92	- 5.74	0.38	11.00
100	- 5.74	- 2.69	1.63	6.15
200	- 3.55	- 1.22	2.71	4.35
400	- 2.25	0.25	5.18	5.39

Table 10.3 Cross talk values for sensor A6 at different force levels.

Figure 10.2 Mechanical structure of the sensor array. A local pressure is applied and neighboring elements show negative cross-talk.

10.2 Preliminary Results Obtained with Sensor A6

The final four sensors listed in Table 10.2 yielded data that supported the above hypothesis, i.e., their underestimation decreased as the pressure increased. However, the dielectric used in the first four sensors in Table 10.2 was slightly different from that used in the last four and the first four did not have the top driven shield (explained in Chapter 2) to prevent electrical interference which makes the sensors sensitive to the existence of objects in their vicinity (proximity effect). This change in sensor design caused these first four sensors to react differently than the last four sensors, i.e., relative error increased as the pressure increased. Nevertheless, the use of sensor arrays which either under- or over-estimated the total force according to the given circumstances created an opportunity to evaluate the signal processing methods for both cases, broadening the applicability of the results.

For the case presented in Table 10.1, each signal processing method reduced the relative error for at least one of the test shapes. However, the algebraic cross-talk reduction method and the artificial neural network were the only two methods which reduced the absolute value of the overall error (bottom row of Table 10.1).

Both of these methods reduced the absolute value of the relative error for each of the shapes applied (each row of Table 10.1 corresponds to a different test object) except when the artificial neural network was applied to the sensor output from shape number 1. Since this object has the area of one cell, its contact with the sensor array excited only very few cells, therefore, very few neurons in the network were excited also. This small amount of excitation provided so little input that the network produced zero output, yielding -100 percent relative error. The data shown in Table 10.1 was obtained after a 2400 iteration training phase of the neural network. Extensive training of the network (almost 50,000 iterations) substantially reduced this problem (see the last column of Table 10.1). Therefore, the algebraic cross talk reduction method and the artificial neural network signal processor (after sufficient training) both reduced the relative error values consistently for sensor A6, at a pressure of 50 grams/cell, using 22 objects at 200 different orientations. However, the reduction in the relative error was much greater using the artificial neural network signal processor.

It is not surprising that the algebraic cross-talk and artificial neural network signal processing methods performed as expected (i.e., reduced the relative error) in the above example. These two methods are the only ones that do not rely on any assumptions regarding the behavior of the sensor, such as overestimation of the total force. The variable threshold method and 2x2 edge and shape detection method both eliminate the outputs of some elements to reduce the effect of the cross-talk. When sensors underestimated the total applied force, however, elimination of some of the element outputs further reduced the total force estimation, thereby increasing the absolute value of the error. For sensors which overestimated the total force, method worked as expected (quantitative results will be presented later in this chapter).

Unsatisfactory results were obtained using the deconvolution via FFT method. This was again due to the negative cross-talk. If the cross-talk is negative, then the sensor transfer function will have a zero-crossing on both sides of the origin. This requires the existence of a high frequency component in the frequency spectrum at a frequency equal to or higher than the maximum frequency allowed by the Nyquist Criterion. This causes aliasing

and therefore a mis-representation of the sensor transfer function. As a result, the deconvolution yields an incorrect estimate of the input signal. When the cross-talk is positive, zero crossing never occurs, therefore the results would be quite satisfactory (results obtained using sensors with positive cross-talk characteristics will be presented later in this chapter).

Figure 10.3 shows boxplots of the relative error distribution for the raw and processed data. As can be seen from this figure, the artificial neural network signal processor is the only signal processor which not only reduced the magnitude of the relative error by moving its median value close to zero, but also compressed the distribution of the relative error values, making the performance of this method most consistent.

10.3 Post Processing

The same information presented in Figure 10.3 is presented in a different format in Figure 10.4. Errors were sorted by the area of the test object. As can be seen from Figure 10.4, the relative error values for some of the signal processing (e.g. deconvolution and 2x2 edge and shape detection) methods exhibited a dependence upon the area of the object in contact with the sensor array.

Figure 10.3 Boxplot of relative error values presented in Table 10.1, for sensor A6, pressure = 50 grams/cell.

Figure 10.4 Bar graphs of relative error values presented in Table 10.1, for sensor A6, pressure = 50 grams/cell.

Figure 10.5 presents the relative error as a function of object area using the data of Table 10.1. Close observation of Figure 10.5 again shows a relationship between the relative error and the area of the object applied to the sensor surface. Therefore, it might be possible to compensate for some of this error by determining the relation between the relative error and the object size. Such a relation can be expressed by fitting a curve to the data. One empirically determined curve may have a linear portion and a hyperbolic region. This relationship is given by:

$$\epsilon = a A + b + \frac{c}{d + e A} \quad (10.3)$$

where a , b , c , d and e are constants, A is the area of contact and ϵ is the relative error. This equation was chosen to encompass both the linear and non-linear behaviour of the relative error and object area relationship.

Coefficients a , b , c , d and e can be determined using the nonlinear regression methods described in Chapter 3 and the 150 element training data set. Area and relative error terms were both obtained from the outputs of the signal processors to generate the

Figure 10.5 Scatterplot of relative error values for VTM, ALG and FFT, presented in Table 10.1 for sensor A6, pressure = 50 grams/cell. Continuous curve is the result of curve-fitting for post processing.

regression data, except in the case of the artificial neural network signal processor. For the variable threshold, algebraic cross-talk and deconvolution methods, the area was calculated as the number of cells with positive outputs after the signal processing. For the 2x2 edge and shape detection method, the double resolution area information generated by the method was used for the regression. Since the artificial neural network signal processor generated a single output, which was an estimate of the total sensor input, the area information did not exist in the signal processor output. Therefore, in the regression calculations, the area information for the artificial neural network signal processor was obtained from the area data generated by the 2x2 edge and shape detectors since they provided an accurate measure of the area. At run time, each signal processing method was applied to the raw data as described in previous chapters. Once the processed data were obtained, the object area could be determined from these data. Knowing the estimated object area as well as the relation between the object area and relative error, one can determine the relative error from equation 10.3. Then, rewriting equation 10.1:

$$\varepsilon = \frac{F_E - F_A}{F_A} \quad (10.1)$$

or,

$$F_A = \frac{F_E}{1 + \varepsilon} \quad (10.4)$$

where F_E (total estimated force) can be obtained from the signal processing method, and ε (relative error) can be obtained from equation 10.3. Equation 10.4 then gives a new estimate of F_A (actual applied force). Table 10.4 shows the same data as presented in Table 10.1 after this post processing. As can be seen from the final row of Table 10.4, all the methods reduced the overall relative error after post processing with respect to the error in the raw data. However, for all methods other than the artificial neural network, interquartile ranges increased. This meant that the spread in the error values increased as well. Table 10.5 compares the relative error values before and after post processing using the bottom rows of Tables 10.1 and 10.4.

Shape Number (Area)	Number of Trials	RELATIVE ERRORS (%) (Median : Inter Quartile Range)					
		RAW	VTM	ALG	2x2	FFT	ANN* (2,450)
1- 1	6	29.0:116	78.0:104	-21.0:52	6.0:84	-15.0:108	-100:0.0
2- 4	16	-9.8:16	17.3:16	17.3:30	12.5:16	-27.5:15	-2.8:98
3- 9	15	-12.2:4.6	16.4:10	9.1:16	18.4:8.1	-26.2:7.6	6.7:9.4
4-16	15	-18.1:7.1	3.5:16	-4.6:12	0.1:8.5	-21.3:9.1	-3.1:5.1
5-25	8	-23.1:7.2	-1.0:18	-12.0:12	-4.6:13	-18.0:6.4	-1.8:9.2
6-36	5	-15.0:1.3	9.6:12	4.6:0.4	12.9:26	10.4:2.9	-0.1:1.1
7-49	7	-15.5:2.1	-6.7:21	4.8:5.1	7.2:7.9	0.2:2.8	0.3:0.8
8-64	1	-12.6:0.0	0.4:0.0	-8.1:0.0	-18.6:0.0	-2.3:0.0	-0.8:0.0
9- 4	7	-16.0:19	16.0:39	31.0:26	-10.5:20	-33.5:19	-7.0:36
10- 8	9	-27.3:12	-0.5:9.0	-12.0:65	7.3:14	-27.0:5.5	-0.3:30
11-16	9	-29.6:7.8	-1.4:20	-17.3:3.9	-3.3:7.4	-24.3:8.9	-0.5:2.4
12-32	7	-21.1:16	3.8:11	-1.3:22	13.3:23	-4.0:24	0.5:2.7
13-48	2	-19.2:4.7	-18.6:37	-11.0:5.9	-21.9:4.3	-11.1:5.5	-3.3:5.1
14-12	16	-15.3:8.2	4.0:8.1	-12.2:9.1	0.7:20	-15.4:10	10.6:10
15-32	7	-12.6:4.9	6.1:15	2.6:5.6	8.0:8.1	-0.4:7.1	0.0:1.0
16- 8	7	-14.5:16	-3.5:47	11.0:32	-9.5:35	14.0:31	-0.8:10
17-24	7	-13.9:25	-2.4:44	2.8:32	12.4:41	19.3:42	1.1:2.8
18- 9	16	-33.8:12	-19.8:16	-31.3:14	-33.4:48	-40.7:14	-4.8:7.6
19-12	8	-49.1:23	-42.1:35	-50.4:24	-46.3:31	-55.9:30	-4.7:25
20-12	10	-24.3:12	-7.5:23	-22.6:17	-0.1:34	-16.9:22	3.8:11
21- 8	13	-26.3:13	-6.3:29	-14.0:7.8	-27.3:31	-15.0:13	8.5:26
22-24	9	-33.3:11	-27.5:27	-30.4:12	-36.9:40	-16.2:19	-0.7:4.0
Overall Error		-17.1:12	-0.8:13	-9.6:22	0.0:27	-15.8:24	-0.6:3.6

* Number in parenthesis is the number of training iterations for ANN.

VTM: Variable Threshold Method
 ALG: Algebraic Cross Talk Method
 2x2: 2x2 Edge and Shape Detection Method
 FFT: Deconvolution Method Using FFT
 ANN: Artificial Neural Network Method

Table 10.4 Relative error values for sensor A6,
 pressure = 50 grams/cell,
 post processed.

OVERALL RELATIVE ERRORS BEFORE AND AFTER POST PROCESSING (%)		
Signal Processor	Before Post Processing	After Post Processing
RAW	- 17.1 : 12	- 17.1 : 12
VTM	- 31.7 : 14	- 0.8 : 13
ALG	- 4.8 : 100	- 9.6 : 22
2x2	- 28.5 : 22	0.0 : 27
FFT	66.4 : 92	- 15.8 : 24
ANN	- 0.5 : 3.1	- 0.6 : 3.6

Table 10.5 Effect of post processing on the overall total force estimation for sensor A6, pressure 50 gram/cell.

Figure 10.6 presents the same information presented in Figure 10.6, after post processing. Figure 10.6 shows that after post processing, all the methods brought the median value of the relative error close to zero, meaning that the magnitude of the median error was reduced. However, compression of the error distribution, which is an indication of the robustness of the method, was only significant for the artificial neural network signal processor. The artificial neural network signal processor reduced the interquartile range of the overall error from 12 percent to 3.6 percent. Figure 10.7 shows

Figure 10.6 Boxplot of relative error values presented in Table 10.4, for sensor A6, pressure = 50 grams/cell.

Figure 10.7 Bar graph of relative error values presented in Table 10.4, for sensor A6, pressure = 50 grams/cell.

the data from Table 10.4 in bar graph form. As can be seen from the bar graph, relative errors were reduced after signal processing for most test objects, using the majority of the signal processors. The results of some signal processors were more consistent than others.

Tables similar to 10.1 and 10.4 were generated for each sensor and each pressure used in this study. Tables 10.6 and 10.7 show the overall relative error values, summarizing the bottom lines of the error tables for each sensor at both pressure levels. As can be seen from Table 10.6, the artificial neural network signal processor always reduced the median relative error to a very small percentage where the others reduced the errors only occasionally. However, after post processing, all of the signal processing methods reduced the median error with the exception of a few cases (Table 10.7). The data from these tables are shown in Figures 10.7 and 10.8, respectively.

Sensor	Pressure	RELATIVE ERRORS (%) (Median : Inter Quartile Range)					
		Raw	VTM	ALG	2x2	FFT	ANN
05	25	25:29	-34:38	43:55	0:47	24:26	- 2:12
05	50	- 3:16	-16:18	9:50	-21:26	17:21	- 1: 6
10	25	- 3:19	-25:16	- 6:21	-34:19	29:22	- 1:22
10	50	-18:11	-39:13	-22:11	-39:18	15:16	- 1: 4
21	25	7:27	-30:23	18:854	-24:37	-78:191	- 0: 5
21	50	- 4:15	-32:12	9:841	-20:25	45:48	- 1: 2
22	25	26:36	-24:49	87:78	-23:44	23:38	- 1: 9
22	50	17:19	-18:13	59:70	- 9:32	15:20	0: 2
36	25	- 1:15	-48:38	1:52	-31:31	4:20	0:18
36	50	8:24	-24:19	6:46	-19:36	8:23	- 0: 5
37	25	- 6:12	-29:21	15:76	-30:28	10:17	0:17
37	50	- 3:20	-35:21	24:90	-21:22	9:22	- 0: 4
A5	25	- 7:12	-26:10	25:21	-37:21	37:22	- 2:25
A5	50	- 4:12	-29:22	25:20	-29:24	21:23	0: 4
A6	25	-30:26	-67:13	-23:90	-42:22	63:79	- 3:10
A6	50	-17:12	-32:14	- 5:100	-29:22	66:92	- 1: 3
Overall		-3:12	-31:11	12:27	-26:12	19:23	- 1: 1

VTM Variable Threshold Method
 ALG Algebraic Cross Talk Method
 2x2 2x2 Edge and Shape Detection Method
 FFT Deconvolution Method Using FFT
 ANN Artificial Neural Network Method

Table 10.6 Overall relative error values for all sensors, all pressures.

Figure 10.8 Bar graph of overall relative error values for all sensors, all pressures, shown in Table 10.6.

Sensor Number	Pressure	POST-PROCESSED RELATIVE ERRORS (%) (Median : Inter Quartile Range)					
		Raw	VTM	ALG	2x2	FFT	ANN
05	25	25:29	- 2:30	- 4:35	7:34	7:21	- 6:12
05	50	- 3:16	2:16	- 4:44	- 1:13	1:17	- 0: 4
10	25	- 3:19	3:19	6:22	- 1:22	4:25	- 1:10
10	50	-18:11	1:17	2:10	- 2:23	1:13	- 0: 4
21	25	7:27	- 1:33	- 1:22	6:26	- 1:30	1: 4
21	50	- 4:15	- 4:11	- 9:17	- 2:17	- 0: 8	- 1: 2
22	25	26:36	3:44	0:37	6:49	11:41	- 1:12
22	50	17:19	- 1:16	- 7:43	4:27	- 0:14	- 0: 2
36	25	- 1:15	6:39	2:19	- 6:31	- 0:17	0:27
36	50	8:24	1:24	- 1:16	- 4:27	1:14	- 0: 5
37	25	- 6:12	- 1:19	7:35	1:28	1:13	1:15
37	50	- 3:20	- 6:29	1:28	1:15	4:12	0: 5
A5	25	- 7:12	0:14	3:16	7:21	5:14	- 2:25
A5	50	- 4:12	3:24	- 1:15	- 2:23	- 1: 9	0: 3
A6	25	-30:26	6:52	- 7:30	6:34	-17:35	- 3:13
A6	50	-17:12	- 1:13	-10:22	0:27	-16:24	- 1: 4
Overall		-3:14	1: 4	- 1: 7	0: 8	1: 4	0: 1

VTM: Variable Threshold Method
 ALG: Algebraic Cross Talk Method
 2x2: 2x2 Edge and Shape Detection Method
 FFT: Deconvolution Method Using FFT
 ANN: Artificial Neural Network Method

Table 10.7 Overall relative error values for all sensors, all pressures, post processed.

Figure 10.9 Bar graph of overall relative error values for all sensors, all pressures, shown in Table 10.7.

10.4 Improvement Factors for Relative Error Values

The comparisons among the various signal processing methods carried out above reflected the results from only one sensor at one pressure level at a time. In order to derive more general conclusions, the performance of each signal processor should be evaluated across all sensors and all objects. Since the relative error in the raw data varied both in magnitude and sign from sensor to sensor, it was difficult to compare the performance of signal processing methods from one sensor to another using the relative error values. Different signal processing methods could be compared, however, if their improvement of the relative error was considered. To achieve this end, an "improvement factor" was defined as follows:

$$\beta = \frac{|\epsilon_{\text{unprocessed}}| - |\epsilon_{\text{processed}}|}{|\epsilon_{\text{unprocessed}}|} \times 100 \% \quad (10.5)$$

where β is the improvement factor and ϵ is the relative error as defined in equation 10.1.

The higher the improvement factor, the better the performance of the signal processor. An improvement

factor of zero means that the signal processor did not improve the total force estimation. On the other hand, 100 percent improvement means that the signal processor completely eliminated the force estimation error. However, if the improvement factor is less than zero, this indicates an increase in the relative error caused by the signal processor which is an undesired result.

Tables 10.8 and 10.9 contain representative improvement factors for the data presented above for sensor A6, at a pressure of 50 grams/cell, before and after post processing. The penultimate column of Table 10.8 contains the improvement factors where the artificial neural network was trained up to 2400 iterations. The column on the extreme right of Table 10.8 shows the improvement factors with 50,000 step training of the network (as in Table 10.1). However, as can be seen from Table 10.8, the artificial neural network signal processor provided an overall improvement of more than 99 percent meaning that the network reduced the error to less than one percent of its original value. A graphic representation of the improvement values is presented in Figure 10.10 for the signal processed data before post processing. Figure 10.11 contains the improvement factor data after post processing. As can be

Shape Number (Area)	IMPROVEMENT FACTORS (%)							
	(Median : Inter Quartile Range)							
	VTM	ALG	2x2	FFT	ANN [*] (2,488)	ANN [*] (50,000)		
1- 1	40.2 : 75	2.3 : 3165	-50.0 : 197	-100.0 : 121	-127.7 : 308	59.5 : 213		
2- 4	-81.1 : 246	-1.8 : 2813	-103.2 : 383	-25.8 : 448	-251.8 : 656	89.7 : 108		
3- 9	-82.1 : 107	54.1 : 1100	-106.3 : 68	-452.1 : 321	36.0 : 83	97.6 : 20		
4-16	-55.7 : 35	65.6 : 533	-29.3 : 32	-297.4 : 243	81.5 : 11	99.4 : 1.1		
5-25	-24.2 : 34	62.2 : 436	-16.4 : 17	-285.4 : 166	87.8 : 32	100 : 22		
6-36	-65.2 : 51	72.3 : 160	-37.9 : 161	-674.4 : 95	95.2 : 9	100 : 0.0		
7-49	-90.5 : 46	56.2 : 323	-78.3 : 62	-670.8 : 133	97.0 : 3	100 : 5.3		
8-64	-42.4 : 0	14.9 : 0.0	-21.1 : 0	-905.5 : 0.0	97.8 : 0	100 : 0.0		
9- 4	-41.4 : 55	8.9 : 2368	-212.5 : 338	17.9 : 61	-14.3 : 80	78.6 : 54		
10- 8	-13.9 : 35	29.6 : 567	-17.3 : 32	79.2 : 47	29.4 : 52	99.4 : 24		
11-16	-17.1 : 19	37.6 : 173	-18.1 : 12	-50.2 : 88	89.6 : 14	100 : 0.3		
12-32	-11.9 : 61	49.5 : 1206	-14.9 : 12	-325.2 : 573	97.0 : 7	100 : 5.2		
13-48	-67.4 : 116	77.4 : 452	-4.2 : 2	-463.5 : 203	86.8 : 23	79.3 : 41		
14-12	101.2 : 151	57.1 : 1391	-100.0 : 70	-511.9 : 658	-1.1 : 85	98.9 : 9.0		
15-32	-87.0 : 107	32.1 : 820	-56.2 : 34	-906.6 : 393	96.3 : 4	100 : 0.0		
16- 8	-75.2 : 126	37.5 : 1751	-106.9 : 177	-413.8 : 593	60.3 : 113	100 : 36		
17-24	-50.4 : 59	29.1 : 2699	-29.9 : 21	-697.0 : 1326	88.6 : 17	100 : 0.0		
18- 9	-32.0 : 24	17.2 : 294	-53.4 : 62	59.2 : 76	79.6 : 27	99.4 : 7.6		
19-12	-21.5 : 14	4.9 : 211	-30.7 : 51	37.2 : 32	85.0 : 34	99.7 : 3.8		
20-12	-38.1 : 26	37.9 : 389	-38.7 : 33	-149.4 : 221	77.8 : 20	96.9 : 9.9		
21- 8	-43.0 : 51	15.9 : 227	-60.2 : 126	22.6 : 51	55.2 : 51	96.6 : 55		
22-24	-37.2 : 45	47.8 : 248	-53.5 : 67	-73.7 : 145	96.8 : 32	100 : 69		
Overall	-42.7 : 51	37.5 : 1293	-44.4 : 57	-291.4 : 486	83.2 : 59	99.6 : 3.1		

* Numbers in parenthesis are the training cycle iterations for ANN.

VTM: Variable Threshold Method
 ALG: Algebraic Cross Talk Method
 2x2: 2x2 Edge and Shape Detection Method
 FFT: Deconvolution Method Using FFT
 ANN: Artificial Neural Network Method

* Numbers in parentheses are the number of iterations performed during training of ANN.

Table 10.8 Improvement factors for sensor A6,
 pressure = 50 grams/cell.

Figure 10.10 Boxplot of improvement factors presented in Table 10.8, for sensor A6, pressure = 50 grams/cell.

Shape Number (Area)	Number of Trials	IMPROVEMENT FACTORS (%) (Median : Inter Quartile Range)				
		VTM	ALG	2x2	FFT	ANN [*] (2,480)
1- 1	6	-5.2 : 164	56.5 : 78	1.1 : 125	-89.7 : 166	-127.7 : 308
2- 4	16	-74.0 : 367	-76.9 : 385	-83.5 : 246	-127.9 : 226	-253.0 : 644
3- 9	15	-11.1 : 67	16.7 : 171	-27.3 : 136	-136.4 : 104	42.0 : 81
4-16	15	56.1 : 28	74.6 : 17	71.5 : 25	-12.3 : 14	82.0 : 10
5-25	8	54.0 : 48	45.9 : 36	73.3 : 31	21.1 : 18	88.8 : 31
6-36	5	37.9 : 32	67.2 : 3.4	-3.6 : 93	32.1 : 26	94.1 : 8.7
7-49	7	17.9 : 28	68.7 : 28	49.9 : 47	89.5 : 13	97.4 : 3.3
8-64	1	96.5 : 0.0	35.5 : 0.0	-47.4 : 0.0	81.6 : 0.0	93.8 : 0.0
9- 4	7	-35.7 : 316	-22.4 : 422	-46.2 : 110	-72.4 : 184	-10.7 : 104
10- 8	9	75.2 : 43	11.3 : 80	45.7 : 88	3.5 : 20	29.4 : 52
11-16	9	69.2 : 39	48.6 : 16	74.2 : 18	18.1 : 12	89.2 : 15
12-32	7	66.3 : 10	34.3 : 42	37.1 : 131	45.2 : 21	97.7 : 7.8
13-48	2	13.7 : 171	43.6 : 17	-14.2 : 5.7	42.8 : 14	84.5 : 23
14-12	16	53.2 : 56	13.2 : 30	38.3 : 85	-4.9 : 16	-2.9 : 85
15-32	7	12.8 : 52	78.6 : 40	36.3 : 77	85.3 : 31	96.9 : 5.7
16- 8	7	0.0 : 153	-10.3 : 221	33.3 : 224	-3.7 : 229	62.5 : 132
17-24	7	4.3 : 171	40.2 : 136	-67.7 : 141	-38.3 : 325	88.6 : 17
18- 9	16	47.3 : 29	-1.5 : 17	0.3 : 110	-29.5 : 12	79.6 : 26
19-12	8	14.3 : 37	6.7 : 19	11.9 : 21	-8.8 : 16	85.3 : 33
20-12	10	47.4 : 50	23.6 : 63	63.6 : 92	25.6 : 31	77.0 : 20
21- 8	13	49.5 : 50	40.2 : 24	23.2 : 94	40.9 : 26	53.1 : 49
22-24	9	14.0 : 62	16.7 : 15	-5.8 : 85	51.5 : 32	97.1 : 32
Overall Impr.		27.9 : 50	34.9 : 37	17.5 : 60	10.8 : 72	83.2 : 52

* Number in parenthesis is the number of training cycle iterations for ANN.

VTM: Variable Threshold Method
 ALG: Algebraic Cross Talk Method
 2x2: 2x2 Edge and Shape Detection Method
 FFT: Deconvolution Method Using FFT
 ANN: Artificial Neural Network Method

Table 10.9 Improvement factors for sensor A6,
 pressure = 50 grams/cell,
 post processed.

Figure 10.11 Boxplot of improvement factors presented in Table 10.9, for sensor A6, pressure = 50 grams/cell, post processed.

seen from this figure, the improvement factor was positive for the algebraic cross-talk reduction and artificial neural network signal processors even without the post processing. After the post processing, the other methods also showed higher improvement factors.

Relative error and improvement factor tables similar to Tables 10.1, 10.4, 10.8 and 10.9 were constructed for all sixteen of the data sets (eight sensors, two pressure levels), and the relative error and overall improvement factor analysis results are shown in Tables 10.10 through 10.13 and Figures 10.12 through 10.15. The entries in these tables are the median and interquartile range of the entries in the relative error and improvement factor tables for all sixteen data sets, before and after post processing. Figure 10.16 compares the overall improvement factors before and after the signal processing. The artificial neural network signal processor provided superior results prior to and after post processing. In addition, the post processing significantly improved the performance of all the signal processing methods.

Shape Number (Area)	Number of Trials	RELATIVE ERRORS (%) (Median Inter Quartile Range)					
		RAW	VTM	ALG	2x2	FFT	ANN [*] (2,400)
1- 1	6	111.0 198	39.5 118	173.5 260	27.0 111	142.0 268	-100 0.0
2- 4	16	15.8 46	-18.4 26	43.6 88	-20.3 22	17.6 42	-33.0 61
3- 9	15	1.4 19	-31.8 21	29.8 42	-27.6 18	16.6 39	9.7 7.0
4-16	15	-7.9 29	-30.4 11	10.8 41	-27.4 22	14.1 14	-0.1 5.9
5-25	8	-3.3 23	-29.9 14	13.8 43	-19.5 21	18.9 16	0.1 3.6
6-36	5	-1.9 31	-23.4 21	11.9 42	-14.3 23	22.3 13	-0.7 4.6
7-49	7	1.1 17	-25.2 26	17.5 31	-12.8 24	21.5 28	0.0 1.1
8-64	1	7.5 24	-15.8 34	15.2 38	-41.0 42	26.1 22	-0.5 0.5
9- 4	7	12.8 57	-23.0 32	33.3 72	-31.3 28	25.5 54	-38.3 65
10- 8	9	-4.5 27	-31.8 14	7.9 63	-28.5 24	8.1 26	-2.8 22
11-16	9	-6.4 20	-27.3 20	1.1 22	-25.9 26	15.2 20	1.3 5.1
12-32	7	-5.6 20	-23.4 12	11.0 43	-19.2 37	18.7 43	-0.0 0.6
13-48	2	3.0 20	-33.9 22	4.5 20	-24.6 58	25.8 31	-0.2 6.5
14-12	16	-4.3 20	-29.3 10	13.5 27	-26.3 13	15.3 19	2.4 6.2
15-32	7	-4.5 17	-27.1 15	10.8 38	-17.1 18	15.8 30	-0.3 0.8
16- 8	7	-13.8 35	-36.6 24	17.3 60	-47.1 20	2.9 19	-11.9 20
17-24	7	-5.0 19	-35.7 23	5.8 19	-28.0 24	11.5 38	-0.6 2.8
18- 9	16	-5.7 25	-39.0 14	8.4 32	-45.4 16	24.3 38	2.8 24
19-12	8	-11.7 16	-48.6 14	-2.3 30	-51.3 18	10.6 38	0.2 7.4
20-12	10	-3.2 16	-35.3 16	12.8 24	-30.6 13	16.0 34	3.3 6.5
21- 8	13	-9.8 20	-39.4 19	4.1 58	-44.9 22	11.3 25	2.1 25
22-24	9	-7.6 18	-45.4 18	3.5 28	-44.3 19	16.2 39	-0.4 1.3
Overall Error		-3.1 14	-29.4 10	12.1 27	-26.2 12	18.8 23	-0.5 0.8

* Number in parenthesis is the number of training cycle iterations for ANN.

VTM: Variable Threshold Method
 ALG: Algebraic Cross Talk Method
 2x2: 2x2 Edge and Shape Detection Method
 FFT: Deconvolution Method Using FFT
 ANN: Artificial Neural Network Method

Table 10.10 Relative errors for all sensors,
all pressure levels.

Figure 10.12 Boxplot of relative errors presented in Table 10.10, for all sensors, all pressures.

Shape Number (Area)	Number of Trials	RELATIVE ERRORS (%) (Median Inter Quartile Range)					
		RAW	VTM	ALG	2x2	FFT	ANN [*] (2,486)
1- 1	6	131.0 198	105.5 197	153.0 297	37.5 120	5.0 76	-100 0.0
2- 4	16	15.8 46	27.4 27	30.5 34	18.8 10	18.5 27	-26.9 49
3- 9	15	1.4 19	9.1 23	7.9 19	13.4 19	4.9 26	7.1 11
4-16	15	-7.9 29	-2.9 17	-4.9 19	-1.0 6.2	-3.5 15	-1.1 6.1
5-25	8	-3.3 23	-1.5 27	2.8 20	10.7 12	1.4 8.4	0.0 2.9
6-36	5	-1.9 31	0.4 21	0.4 9.0	4.1 17	3.6 6.1	-0.4 3.8
7-49	7	1.1 17	-0.5 9.4	3.1 10	-0.2 10	2.7 7.3	0.2 1.1
8-64	1	7.5 24	-5.9 24	-5.2 17	-20.5 25	4.3 18	0.0 1.5
9- 4	7	12.8 57	19.5 20	14.0 37	0.5 41	-2.8 32	-27.5 53
10- 8	9	-4.5 27	7.0 14	-2.5 19	7.4 18	-2.5 14	-3.0 22
11-16	9	-6.4 20	5.7 16	-14.0 22	3.9 19	-2.6 12	1.1 5.4
12-32	7	-5.6 20	4.6 11	-2.9 15	1.3 12	-0.3 7.4	0.1 1.1
13-48	2	3.0 20	-8.8 28	-10.1 21	-18.0 40	-0.4 15	-0.2 7.0
14-12	16	-4.3 20	-4.8 14	1.1 26	-5.3 12	-0.2 19	1.8 6.9
15-32	7	-4.5 17	3.1 11	0.3 35	-1.5 11	-2.8 10	-0.3 1.3
16- 8	7	-13.8 35	-2.5 11	-7.9 35	-3.9 21	-6.0 26	-9.3 16
17-24	7	-5.0 19	0.7 20	1.4 26	6.1 29	-1.1 21	-0.3 3.0
18- 9	16	-5.7 25	-9.9 24	-3.3 29	-29.7 18	0.8 24	2.7 20
19-12	8	-11.7 16	-11.8 19	-5.9 20	-17.5 24	-5.3 18	1.4 7.3
20-12	10	-3.2 16	-1.9 20	1.5 10	3.0 23	2.2 13	3.7 6.9
21- 8	13	-9.8 20	-3.0 20	0.3 15	-11.6 32	-6.6 19	2.1 21
22-24	9	-7.6 18	-15.8 28	-10.6 15	-8.3 28	-6.1 19	-0.1 1.2
Overall Error		-3.1 14	0.8 3.7	-1.1 6.9	0.3 7.5	0.6 4.4	-0.3 1.0

* Number in parenthesis is the number of training cycle iterations for ANN.

VTM: Variable Threshold Method
 ALG: Algebraic Cross Talk Method
 2x2: 2x2 Edge and Shape Detection Method
 FFT: Deconvolution Method Using FFT
 ANN: Artificial Neural Network Method

Table 10.11 Relative errors for all sensors,
all pressure levels, post processed.

Figure 10.13 Boxplot of relative errors presented in Table 10.11, for all sensors, all pressures combined and post processed.

Shape Number (Area)	Number of Trials	IMPROVEMENT FACTORS (%) (Median : Inter Quartile Range)				
		VTM	ALG	2x2	FFT	ANN [*] (2,400)
1- 1	6	56.1 : 29	-26.3 : 87	56.6 : 30	0.0 : 17	24.7 : 146
2- 4	16	-3.0 : 193	-47.6 : 182	-41.0 : 172	-61.3 : 106	-151.9 : 229
3- 9	15	-142.0 : 222	-57.2 : 318	-142.8 : 98	-66.0 : 162	-12.0 : 106
4-16	15	-121.5 : 105	-13.4 : 218	-60.3 : 93	-2.1 : 228	56.5 : 29
5-25	8	-155.7 : 187	-91.3 : 211	-66.0 : 131	-77.0 : 274	72.5 : 27
6-36	5	-59.0 : 195	-61.6 : 197	-31.7 : 158	-111.2 : 431	74.4 : 35
7-49	7	-102.6 : 196	-116.5 : 186	-18.4 : 301	-349.1 : 794	87.1 : 29
8-64	1	-80.9 : 224	-27.4 : 212	-153.4 : 392	-210.7 : 569	96.3 : 5.9
9- 4	7	-36.2 : 138	-9.1 : 116	-65.6 : 153	-2.2 : 52	-74.9 : 218
10- 8	9	-71.7 : 116	-20.4 : 170	-63.7 : 130	0.6 : 120	-6.1 : 172
11-16	9	-98.8 : 126	-6.8 : 68	-62.0 : 111	0.0 : 85	62.3 : 34
12-32	7	-115.0 : 275	-32.6 : 275	-48.8 : 162	-60.8 : 268	85.2 : 24
13-48	2	-230.1 : 832	-53.9 : 180	-124.7 : 371	-170.8 : 929	78.5 : 98
14-12	16	-216.5 : 295	-50.9 : 264	-117.3 : 295	-65.6 : 363	28.5 : 79
15-32	7	-183.3 : 269	-84.7 : 176	-84.4 : 138	-35.9 : 603	83.1 : 24
16- 8	7	-78.1 : 112	-25.1 : 125	-116.9 : 140	5.3 : 53	25.6 : 102
17-24	7	-173.5 : 221	-31.6 : 228	-86.1 : 178	-31.2 : 601	82.3 : 36
18- 9	16	-127.5 : 282	-7.6 : 94	-191.5 : 338	0.0 : 216	10.6 : 103
19-12	8	-135.9 : 105	-9.4 : 156	-189.3 : 116	0.0 : 123	47.7 : 64
20-12	10	-166.0 : 244	-27.9 : 167	-156.9 : 220	-13.0 : 267	33.9 : 41
21- 8	13	-121.6 : 136	-14.7 : 77	-122.0 : 76	0.0 : 115	-1.9 : 115
22-24	9	-260.8 : 312	-17.2 : 85	-278.2 : 293	-2.4 : 205	88.5 : 27
Overall Impr.		-105.4 : 124	-48.7 : 152	-87.5 : 92	-55.9 : 163	51.7 : 36

* Numbers in parenthesis are the number of training cycle iterations.

VTM: Variable Threshold Method
 ALG: Algebraic Cross Talk Method
 2x2: 2x2 Edge and Shape Detection Method
 FFT: Deconvolution Method Using FFT
 ANN: Artificial Neural Network Method

Table 10.12 Improvement factors for all sensors,
all pressure levels combined.

Figure 10.14 Boxplot of improvement factors presented in Table 10.12, for all sensors, all pressures combined.

Shape Number (Area)	Number of Trials	IMPROVEMENT FACTORS (%) (Median : Inter Quartile Range)				
		VTM	ALG	2x2	FFT	ANN [*] (2-480)
1- 1	6	29.0 : 38	-7.4 : 101	38.4 : 66	67.4 : 49	47.5 : 151
2- 4	16	-35.7 : 101	-50.6 : 68	-7.7 : 157	-4.0 : 92	-165.2 : 230
3- 9	15	-14.1 : 116	-33.2 : 200	-52.8 : 107	-35.9 : 110	-1.8 : 96
4-16	15	-46.1 : 83	-24.0 : 134	19.9 : 75	25.9 : 71	57.9 : 34
5-25	8	-73.9 : 170	-47.3 : 165	-72.0 : 193	41.3 : 42	74.9 : 32
6-36	5	-11.1 : 92	56.8 : 139	-22.3 : 140	31.2 : 69	77.7 : 36
7-49	7	-66.4 : 119	4.1 : 186	22.2 : 290	35.1 : 93	90.1 : 29
8-64	1	8.0 : 201	33.0 : 119	-80.6 : 201	43.3 : 107	93.0 : 21
9- 4	7	-37.9 : 66	-24.8 : 74	-43.7 : 127	9.1 : 61	-74.9 : 218
10- 8	9	-12.8 : 85	4.1 : 58	-37.5 : 109	3.8 : 45	6.9 : 179
11-16	9	-39.9 : 135	-16.0 : 109	-38.7 : 134	17.0 : 59	61.3 : 39
12-32	7	-67.8 : 285	5.1 : 216	14.4 : 130	44.3 : 33	88.3 : 18
13-48	2	-100.1 : 419	9.3 : 334	-104.8 : 518	26.8 : 174	68.3 : 169
14-12	16	-74.5 : 159	-21.2 : 181	-18.6 : 84	17.2 : 82	29.6 : 62
15-32	7	-21.7 : 234	-96.2 : 357	31.5 : 57	39.9 : 55	87.7 : 39
16- 8	7	-4.2 : 92	-15.7 : 78	4.8 : 96	15.3 : 71	24.4 : 101
17-24	7	-44.4 : 131	-92.9 : 140	-72.6 : 148	-13.5 : 136	82.0 : 35
18- 9	16	-59.4 : 63	-9.2 : 61	-135.9 : 316	-15.2 : 90	13.0 : 102
19-12	8	-25.0 : 76	9.6 : 98	-50.0 : 138	7.0 : 40	55.6 : 73
20-12	10	-47.6 : 135	-12.7 : 97	-83.8 : 224	19.0 : 71	32.8 : 44
21- 8	13	-16.8 : 69	11.7 : 62	-54.6 : 114	-0.5 : 64	-1.9 : 117
22-24	9	-100.5 : 189	-28.1 : 126	-101.4 : 192	-2.7 : 66	89.4 : 36
Overall Impr.		-29.4 : 48	-6.9 : 82	-40.1 : 66	11.6 : 29	55.2 : 35

* Numbers in parenthesis are the number of training cycle iterations for ANN.

VTM: Variable Threshold Method
 ALG: Algebraic Cross Talk Method
 2x2: 2x2 Edge and Shape Detection Method
 FFT: Deconvolution Method Using FFT
 ANN: Artificial Neural Network Method

Table 10.13 Improvement factors for all sensors, all pressures combined and post processed.

Figure 10.15 Boxplot of improvement factors presented in Table 10.13, for all sensors, all pressures combined and post processed.

Figure 10.16 Comparative boxplots for improvement factors presented in Tables 10.12 and 10.13, for all sensors, all pressures combined, before and after post proc.

10.5 Error Dependence on Object Area

Tables 10.10 through 10.13 as well as Figures 10.12 through 10.15 also show the dependence of the relative error values and improvement factors on the area of the test objects. Relative error values for the raw data were quite high for objects with smaller sizes. For the majority of the signal processing methods, the improvement factors for small objects were quite high, which was necessary to reduce the high relative errors. However, the sensors in general had lower relative error values for larger objects. These small relative errors decreased the denominator term in equation 10.5 significantly while the numerator was increased only a slight amount by the signal processors. Therefore, the improvement factors became large negative numbers for larger objects. Hence, if the error was large then the signal processors reduced its value. On the other hand, if the error was small, then the signal processors might increase it, but only by a small amount. The only exception to this phenomenon was the artificial neural network signal processor. This method produced high improvement factors for larger objects since larger objects excited more neurons, which is essential for the operation of the neural network.

10.6 Error Dependence on the Applied Pressure

The bottom rows of improvement factor tables such as Tables 10.8 and 10.9 are listed in Tables 10.14 and 10.15 and presented graphically in Figures 10.17 and 10.18. As can be seen from these presentations, there was no apparent dependence of the performance of the signal processors on the sensor tested or on the pressure level used. For example, the variable threshold method provided both positive and negative overall improvement factors for sensors 10 and A6. Dependence of the relative error on the size of the contact object was reduced by the post processing.

Sensor	Number	Pressure	IMPROVEMENT FACTORS (Median : Inter Quartile Range)				
			VTM	ALG	2x2	FFT	ANN
05	25		- 39 : 311	-145 : 192	28 : 180	0 : 7	54 : 84
05	50		- 70 : 59	-184 : 250	- 76 : 85	- 53 : 231	47 : 67
10	25		- 92 : 87	- 4 : 43	-119 : 153	-129 : 257	10 : 161
10	50		-113 : 88	- 18 : 13	- 99 : 113	14 : 91	78 : 49
21	25		-106 : 199	- 14 : 999	- 63 : 199	-542 : 811	42 : 30
21	50		-174 : 269	- 8 : 999	-106 : 110	-256 : 999	57 : 56
22	25		- 8 : 243	-170 : 144	12 : 157	11 : 21	73 : 26
22	50		- 18 : 140	-200 : 234	31 : 77	9 : 16	81 : 26
36	25		-311 : 314	- 7 : 572	-145 : 360	- 7 : 64	-83 : 201
36	50		- 93 : 236	- 4 : 677	- 52 : 261	0 : 6	72 : 91
37	25		-202 : 214	- 92 : 999	-217 : 272	- 17 : 165	1 : 188
37	50		-222 : 210	- 88 : 999	-135 : 199	- 59 : 159	43 : 119
A5	25		-104 : 177	- 80 : 226	-184 : 218	-169 : 321	30 : 97
A5	50		-186 : 212	-271 : 462	-266 : 242	-156 : 380	58 : 79
A6	25		-118 : 98	20 : 999	- 63 : 123	-117 : 673	50 : 74
A6	50		- 43 : 51	38 : 999	- 44 : 57	-291 : 486	83 : 59
Overall			-103 : 123	- 49 : 152	- 87 : 92	- 56 : 162	52 : 36

Table 10.14 Overall improvement factors for all sensors, all pressures.

Overall Improvement Factors

Figure 10.17 Bar graph of overall improvement factors presented in Table 10.14, for all sensors, all pressures.

Sensor Number	Pressure	IMPROVEMENT FACTORS AFTER POST PROCESSING (Median : Inter Quartile Range)				
		VTM	ALG	2x2	FFT	ANN
05	25	- 60 : 128	- 55 : 212	1 : 145	40 : 36	54 : 84
05	50	- 10 : 85	-150 : 155	4 : 138	5 : 44	51 : 91
10	25	- 14 : 95	- 10 : 133	61 : 205	8 : 135	6 : 178
10	50	14 : 100	62 : 60	29 : 123	53 : 55	78 : 49
21	25	- 52 : 66	- 21 : 83	41 : 118	-800 : 690	46 : 39
21	50	- 20 : 104	- 64 : 180	39 : 69	- 8 : 40	56 : 60
22	25	- 14 : 139	27 : 79	44 : 141	52 : 27	74 : 27
22	50	- 8 : 84	-109 : 82	3 : 83	63 : 35	82 : 27
36	25	-291 : 272	- 2 : 111	-130 : 301	1 : 30	- 41 : 194
36	50	- 45 : 85	12 : 125	-12 : 157	19 : 51	73 : 79
37	25	- 62 : 95	- 62 : 130	84 : 119	19 : 72	1 : 195
37	50	-130 : 204	- 97 : 118	56 : 160	12 : 73	54 : 97
A5	25	- 39 : 106	- 4 : 74	67 : 215	4 : 84	30 : 91
A5	50	- 59 : 103	12 : 73	91 : 244	28 : 50	59 : 86
A6	25	- 16 : 123	43 : 82	19 : 82	7 : 82	60 : 71
A6	50	28 : 50	35 : 37	18 : 60	11 : 72	83 : 52
Overall		- 29 : 47	- 15 : 75	- 40 : 65	- 11 : 29	55 : 35

Table 10.15 Overall improvement factors for all sensors, all pressures, post processed.

Figure 10.18 Bar graph of improvement factors presented in Table 10.15, for all sensors, all pressures, post proc.

10.7 Error Dependence on Object Alignment

During the testing of the sensor, one always knows where an object is placed on the sensor array. On the other hand, during the normal operation of the sensor (and the attached signal processing system), the sensor contacts objects in various positions and orientations. Both the sensor and the signal processing system must perform whether the sensor cells are aligned with the edges of the objects or not. To determine if the sensor array or any of the signal processing methods have any dependence object alignment, relative error values were re-sorted according to the orientation of the object. Tables 10.16 and 10.17 show the relative error values for total force estimations with and without the post processing. The first entry in each location of these tables is the median of relative errors where the object edges were aligned with the edges of the sensor elements. The second entry contains the median of the relative errors for the situations where object and sensor element edges were not aligned.² The fact that these two entries are close to each other in every instance shows that

² When the objects were not aligned with the sensor array elements, their corners were 0.5 cell distance away from the corners of the sensor array cells.

Sensor	Pressure	RELATIVE ERROR VALUES (Error With Alignment / Without Alignment)					
		RAW	VTM	ALG	2x2	FFT	ANN (2400)
05	25	26/ 23	-29/-41	49/ 52	3/-10	23/ 23	-1/ -1
05	50	-5/ -8	-16/-21	13/ 13	-19/-27	15/ 10	-1/ -0
10	25	-6/ -5	-24/-29	-11/ -6	-30/-36	30/ 28	2/ -3
10	50	-17/-20	-38/-43	-22/ -24	-34/-41	18/ 14	-1/ -0
21	25	10/ 6	-28/-28	4282/4896	-15/-27	-88/-44	1/ -1
21	50	2/ -5	-31/-32	1747/3853	-13/-25	50/ 32	-0/ -1
22	25	30/ 39	-14/-22	90/ 104	-7/-17	27/ 35	0/ -1
22	50	16/ 22	-22/-16	61/ 62	-8/-17	14/ 20	0/ -0
36	25	4/ 1	-40/-49	119/ 104	-23/-31	11/ 6	-2/ -0
36	50	8/ 6	-26/-27	117/ 114	-14/-20	8/ 5	0/ 0
37	25	-8/ -5	-24/-31	182/ 188	-30/-38	7/ 10	-4/ 2
37	50	3/ -6	-34/-36	195/ 196	-16/-24	15/ 5	0/ 0
A5	25	-7/ -1	-22/-25	20/ 27	-35/-38	35/ 39	-1/ -2
A5	50	-7/ -6	-25/-33	26/ 28	-28/-34	20/ 18	-0/ 1
A6	25	-31/-20	-68/-62	175/ 188	-46/-42	62/ 89	-2/ -2
A6	50	-18/-21	-26/-35	227/ 205	-28/-32	64/ 53	-0/ -0
Overall		-1/ -5	-26/-31	103/ 104	-21/-29	19/ 19	0/ 0

Table 10.16 Relative error values for all sensors, all pressures. First number in each cell is the median value of the relative error when object and the cell edges were aligned. The second number is the median of the relative error without alignment.

Sensor	Pressure	POST PROCESSED RELATIVE ERROR VALUES (Error With / Without Alignment)					
		RAW	VTM	ALG	2x2	FFT	ANN (2400)
05	25	26/ 23	-14/-5	-1/ 6	16/-3	9/ 8	-3/-3
05	50	-5/ -8	0/-0	-3/-3	4/-1	-1/-5	-1/-0
10	25	-6/ -5	4/ 0	4/ 7	4/-0	4/ 4	2/-1
10	50	-17/-20	6/-2	4/ 0	3/-3	1/-1	0/-0
21	25	10/ 6	-2/ 1	1/-3	6/ 2	-43/14	1/-1
21	50	2/ -5	-4/-3	-10/-8	2/-4	-2/-5	0/-0
22	25	30/ 39	7/-1	12/ 7	7/ 3	13/17	1/-1
22	50	16/ 22	-4/ 4	-1/ 8	8/-0	-1/ 1	0/-0
36	25	4/ 1	1/ 0	5/-0	-1/-4	5/ 1	-2/-2
36	50	8/ 6	4/ 4	-1/-4	-2/-0	2/ 2	0/ 0
37	25	-8/ -5	1/-0	8/13	4/-1	-1/ 4	0/ 1
37	50	3/ -6	-2/-5	-3/-0	3/-4	6/ 1	0/ 0
A5	25	-7/ -1	8/ 3	3/ 7	8/ 4	2/ 6	-2/-1
A5	50	-7/ -6	6/ 1	1/ 2	-0/-3	-3/-0	0/ 1
A6	25	-31/-20	-5/11	3/ 1	-1/ 6	-24/14	-2/-3
A6	50	-18/-21	5/-2	-5/10	1/-3	-20/20	-0/-0
Overall		-1/ -5	1/ 0	1/ 1	4/-1	0/ 3	0/ 0

Table 10.17 Relative error values with and without alignment for all sensors, all pressures, post processed.

neither the sensor, nor the signal processors have any dependence on alignment.

10.8 Error Dependence on Object Shape

The dependence of relative error values on the object shape can be deduced from Tables 10.1 and 10.10. (The two numbers in the first column of each table refer to the object number and the size of the object, respectively.) Objects 19 and 20 have the same area (12 cells) but completely different shapes. The former resembles the letter H while the latter resembles the letter V. (See Figure 9.1.) In Table 10.1, the raw error for the former is twice the magnitude of the latter. The results are similar for objects 3 and 18. Inspection of corresponding elements in Table 10.10 shows that this behavior (dependence of relative error on the object shape) was still apparent even after error values from all the sensors were combined, i.e., this dependence was associated with the force sensors used during this study. However, the relationship between signal processor performance and test object shape was not apparent as in the case of sensor A6 alone, i.e., even though the sensors were sensitive to the object shape, the signal processing methods were not.

10.9 Range of Error Values

The interquartile range values listed in the relative error tables are indications of the variation in the error values. Interquartile range values are sometimes larger than the magnitude of the relative error values. This was a result of the wide distribution of the data and showed that the application of the same force using the same object at different locations and orientations yielded different error values. Interquartile ranges rarely decreased after signal processing, meaning that the spread of the data was not reduced by the signal processing. For example, the mean value of interquartile ranges listed in Table 10.1 is 15.4 for the raw data and 13.6 after processing with artificial neural networks. However, as can be seen from Table 10.7, the overall interquartile range values were reduced by superior signal processors such as the artificial neural network.

As mentioned in the beginning of this chapter, a signal processor is expected to work under any input condition instead of just some specific ones. Median and interquartile range are measures for the typical characteristics of the signal processing methods. However, if the performance of a signal processor is to

be determined for a variety of input conditions, then median value and the full range instead of the interquartile range should be considered in order to account for all possibilities. Tables 10.18 and 10.19 show the relative error values and improvement factors previously presented in Tables 10.7 and 10.15 using range instead of interquartile range. As can be seen from Table 10.18, signal processors almost always reduced the range of the error as well as its median value. The range values in Table 10.18 for the artificial neural network were calculated without taking into consideration the outputs produced for object number 1, due to the insufficient training problem of that signal processor, explained in Section 10.2. Doing so does not affect the range values obtained because the errors for object number 1 can be reduced by more extensive training of the neural network method. However, training with 50,000 iterations lasts approximately three weeks using a micro-VAX. Therefore, only one set of data was trained at this level. However, if the neural network signal processors were allowed to train long enough, they would provide the minimal error range as well as the minimal error median among all the signal processors for the total force estimation.

Sensors	Pressure	RELATIVE ERRORS FOR MASS ESTIMATIONS (POST PROCESSED) (Median : Range)					
		RAW	VTM	ALG	2x2	FFT	ANN [*] (2400)
05	25	25 : 372	-2 : 162	-3 : 554	7 : 148	7 : 311	-6 : 119
05	50	-3 : 130	2 : 102	-4 : 360	-1 : 76	1 : 58	-0 : 84
10	25	-3 : 391	3 : 271	6 : 58	-1 : 179	4 : 41	-1 : 165
10	50	-18 : 207	1 : 145	2 : 252	-2 : 165	1 : 41	-0 : 48
21	25	7 : 170	-1 : 174	-1 : 588	6 : 198	-1 : 883	0 : 51
21	50	-4 : 261	-4 : 226	-9 : 297	-2 : 298	-0 : 163	-1 : 31
22	25	26 : 378	3 : 262	0 : 246	6 : 119	11 : 126	-1 : 82
22	50	16 : 281	-1 : 227	-7 : 97	4 : 243	-0 : 31	-0 : 50
36	25	-1 : 91	6 : 213	2 : 72	-6 : 71	-0 : 53	0 : 115
36	50	8 : 55	1 : 58	-1 : 42	-4 : 95	1 : 39	-0 : 22
37	25	-6 : 106	-1 : 73	7 : 58	1 : 99	1 : 57	1 : 95
37	50	-3 : 43	-6 : 60	1 : 86	0 : 61	4 : 27	0 : 22
A5	25	-7 : 229	0 : 209	3 : 102	7 : 123	5 : 69	-2 : 74
A5	50	-4 : 136	3 : 116	-1 : 50	-2 : 108	-1 : 36	0 : 44
A6	25	-30 : 105	6 : 245	-7 : 676	6 : 90	-17 : 100	-3 : 49
A6	50	-17 : 78	-1 : 120	-10 : 81	0 : 65	-16 : 75	-1 : 18
Over- all		-3 : 56	0 : 12	-1 : 17	0 : 13	1 : 28	0 : 7

* Range Values Are Calculated Neglecting Shape Number 1.

Table 10.18 Overall relative error values for all sensors, all pressures, post processed and presented as median and range.

Sensor	Pressure	IMPROVEMENT FACTORS FOR MASS ESTIMATIONS (POST PROCESSED) (Median : Range)				
		VTM	ALG	2x2	FFT	ANN* (2400)
05	25	-60 : 650	-55 : 465	-1 : 249	40 : 200	54 : 319
05	50	-10 : 552	-150 : 1483	4 : 1833	5 : 282	51 : 441
10	25	-14 : 439	-10 : 499	-61 : 2052	8 : 319	6 : 1017
10	50	14 : 452	62 : 612	29 : 2098	53 : 547	78 : 291
21	25	-52 : 296	-21 : 572	-41 : 282	-800 : 2505	46 : 178
21	50	-20 : 459	-64 : 501	-39 : 182	-8 : 155	56 : 274
22	25	-14 : 611	27 : 276	-44 : 614	52 : 223	74 : 177
22	50	-8 : 441	-109 : 381	-3 : 1047	63 : 146	82 : 92
36	25	-291 : 1182	-2 : 549	-130 : 1137	1 : 115	-41 : 822
36	50	-45 : 1153	12 : 498	-12 : 2794	19 : 495	73 : 312
37	25	-62 : 716	-62 : 577	-84 : 1020	19 : 287	1 : 419
37	50	-130 : 595	-97 : 655	-56 : 294	12 : 311	54 : 350
A5	25	-39 : 526	-4 : 419	-67 : 806	4 : 265	30 : 329
A5	50	-59 : 1798	12 : 358	-91 : 963	28 : 173	59 : 4020
A6	25	-16 : 371	43 : 1017	19 : 316	7 : 348	60 : 147
A6	50	28 : 171	35 : 155	18 : 158	11 : 226	83 : 351
Overall		-29 : 219	-15 : 212	-40 : 159	11 : 863	55 : 124

* Improvement Factors are Calculated
Neglecting Shape Number 1.

Table 10.19 Overall improvement factors for
all sensors, all pressures,
post processed and
presented as median and range.

10.10 Error in the Area Estimations

In certain applications, the information about the size (or the area) of the input pattern is as important as total intensity of the input. When object number 14 (see Appendix C) was applied to the sensor array A6 at five different locations using five different pressure levels (10, 25, 50, 75 and 100 grams/cell), the total sensor output was very close to the sensor input even without signal processing (see Figure 10.19). However, the area estimations were satisfactory only for the 2x2 edge and shape detection and the variable threshold methods (see Figure 10.20) since these methods were originally designed to detect the locations of the objects. Since the training of the artificial neural network was done such that the network only provided a single output representing the total sensor output, the area information was not available from this method.

To obtain conclusive results from the area estimation study, an analysis similar to the one used for the mass estimation study was performed. Table 10.21 and Figures 10.21 and 10.22 show the relative error values where Table 10.21 and Figures 10.23 and 10.24 show the corresponding improvement factors for the area estimation study. As can be seen from these two tables, only the

variable threshold method and the 2x2 edge and shape detection methods reduced the area estimation errors and provided improvement factors in the range of 65 to 70 percent.

10.11 Summary

As can be seen from the overall improvement values presented in Tables 10.8 and 10.9, all of the methods did provide improvement of the relative error for at least one of the test objects. All methods other than the artificial neural network provided improvements for only a few of the test objects. Therefore, if these inputs were those with which the sensor array was most likely to come into contact, then these methods would be quite useful. Otherwise, the artificial neural network signal processor would provide the best performance.

Figure 10.19 Object number 14 applied to sensor A6 at five locations, five pressures (10, 25, 50, 75, 100 grams/cell), raw total force data is presented.

Figure 10.20 Raw and processed area data obtained by applying object #14 (area of 12 cells) to sensor A6, at five locations with five pressures.

Sensor	Pressure	RELATIVE ERRORS FOR AREA ESTIMATIONS (Median : Inter Quartile Range)				
		RAW	VTM	ALG	2x2	FFT
05	25	143 : 192	-48 : 33	131 : 253	-29 : 30	143 : 192
05	50	178 : 234	9 : 100	173 : 278	-20 : 38	277 : 316
10	25	148 : 222	-1 : 56	145 : 225	-43 : 39	248 : 313
10	50	164 : 198	-32 : 50	161 : 201	-35 : 37	286 : 359
21	25	87 : 144	-42 : 42	97 : 322	-50 : 33	-55 : 375
21	50	110 : 154	-46 : 42	96 : 292	-33 : 33	205 : 217
22	25	134 : 213	-57 : 35	135 : 257	-68 : 44	132 : 219
22	50	163 : 272	-29 : 47	157 : 278	-42 : 53	160 : 259
36	25	138 : 181	-63 : 42	134 : 234	-44 : 33	156 : 158
36	50	163 : 171	-45 : 50	134 : 253	-41 : 34	163 : 171
37	25	141 : 179	-35 : 38	146 : 278	-57 : 21	234 : 208
37	50	167 : 241	-41 : 44	147 : 291	-38 : 33	281 : 281
A5	25	141 : 178	-7 : 67	183 : 258	-54 : 38	251 : 279
A5	50	161 : 216	-37 : 50	213 : 294	-52 : 29	258 : 324
A6	25	56 : 124	-75 : 24	76 : 200	-55 : 35	150 : 200
A6	50	82 : 129	-24 : 33	118 : 228	-33 : 39	263 : 275
Overall		142 : 41	-39 : 20	140 : 34	-42 : 19	219 : 107

Table 10.20 Relative error values for area estimations from all sensors, all pressures.

Figure 10.21 Boxplot of relative area errors presented in Table 10.20, for all sensors and all pressures.

Figure 10.22 Bargraph of relative area errors presented in Table 10.20, for all sensors and all pressures.

Sensor	Pressure	Improvement Factors for Area Estimations (Median : Inter Quartile Range)			
		VTM	ALG	2x2	FFT
05	25	68 : 83	-3 : 32	74 : 53	0 : 0
05	50	74 : 35	1 : 65	82 : 30	-46 : 19
10	25	75 : 39	0 : 6	69 : 111	-61 : 54
10	50	74 : 54	0 : 6	74 : 52	-61 : 41
21	25	46 : 126	-21 : 92	15 : 177	-179 : 433
21	50	54 : 172	-7 : 96	63 : 66	-83 : 112
22	25	63 : 349	-5 : 21	54 : 336	0 : 10
22	50	74 : 108	4 : 22	64 : 97	0 : 3
36	25	57 : 126	0 : 37	61 : 111	-13 : 27
36	50	73 : 88	12 : 51	71 : 68	0 : 0
37	25	78 : 71	-9 : 46	62 : 136	-51 : 52
37	50	72 : 79	5 : 53	75 : 86	-48 : 56
A5	25	59 : 98	-29 : 29	67 : 157	-82 : 72
A5	50	74 : 77	-30 : 17	72 : 93	-50 : 40
A6	25	-7 : 426	-20 : 122	9 : 230	-139 : 243
A6	50	55 : 103	-66 : 83	55 : 110	-193 : 134
Overall		70 : 18	-4 : 21	65 : 15	-50 : 76

Table 10.21 Improvement factors for all sensors, all pressures for area estimations.

Figure 10.23 Boxplot of improvement factors presented in Table 10.21, for area estimations using all sensors, all pressures.

Figure 10.24 Bar diagram of improvement factors presented in Table 10.21, for area estimations using all sensors and all pressures.

Chapter 11. CONCLUSIONS

This dissertation has demonstrated that it is possible to improve the performance of a sensor array such that its outputs are comparable to those of a sensor with more ideal characteristics. This goal was achieved using digital signal processing methods after analog processing. A set of experiments was performed to gain a quantitative understanding of the performance of the five signal processing methods developed, and a post processing algorithm was introduced to further reduce the error.

The eight sensors used in the evaluation were chosen from different fabrication runs, and had different characteristics. Perhaps the most common property of the sensors was the inconsistency of their characteristics. Their sensitivity, uniformity, cross-talk, and total force estimation error all varied significantly from one to another. As can be seen from the tables in Chapter 10, the total force estimation varied from one pressure level to another, even within the same sensor.

The variable threshold and edge and shape detection methods did not perform well on sensors which

underestimated the total force but they were useful in discrimination of shape and area of the objects forced on the sensor array. The errors in these methods were compensated by post processing of the signal processor output. These two methods were quite easy to train and implement on a digital computer. They were not computationally intensive and could be implemented with only integer arithmetic. Satisfactory experimental results could be obtained for these methods, even without proper training, using parameters determined with synthetic data since their dependence upon sensor characteristics are minimal. These two methods also provided the best results for the area estimations.

The algebraic cross-talk reduction method was more case dependent when compared to the previous two (variable threshold method and the 2x2 edge and shape detection method). Determination of the eigenvalues and eigenvectors of the cross-talk matrices was iterative, where convergence was not guaranteed, and even when convergence occurred, it happened quite slowly. As the matrix size increased, errors in the parameters propagated quickly, making the results quite unsatisfactory. This method provided encouraging results only when the sensors had highly structured cross-talk

between elements. When the cross-talk was reduced by improvements in the sensor manufacturing technology, the residual cross-talk was not further reduced by this method since the eigenvalues and the eigenvectors were very close to each other for the new sensors. Perhaps a singular value decomposition method [Press, 1986] could be used to reduce the cross-talk algebraically for sensors having a moderate amount of cross-talk between the elements. In any case, the algebraic cross-talk reduction method cannot be implemented without floating point arithmetic since matrices with real numbers are needed during the run time.

Deconvolution using FFT yielded good results only if the sensor transfer function was known or could be approximated closely. Once the transfer function was determined, training was quite fast. However, the run time required extensive use of FFT algorithms which were CPU intensive and required floating point arithmetic. Due to the crude approximations of the sensor transfer functions, most results were unsatisfactory with this method. On the other hand, if a better kernel was chosen, then better results could be expected from the deconvolution method. One logical way to choose a better kernel would be to first train an artificial neural

network to learn the input-output relation of the sensor array, and then use the structure formed by the connections of the neural network to develop the convolution kernel.

The neural network signal processor consistently reduced the total sensor input estimation error as well as the range of the error for every sensor at both pressure levels even without post processing. In cases where very few of the cells were excited, the network did not perform well. By extensive training, this drawback was eliminated. Unfortunately, the remarkable results of this method were obtained at the cost of enormous computational time for training. Training of one neural network for one training set consumed approximately 48 hours of CPU time on a micro-VAX II¹. The technique developed to determine appropriate initial values reduced the total training time significantly, however, since this was the first time the network was implemented, all the possible connections were left in the network making its training a huge computational task. Since all connections between neurons which represented

¹ Digital and VAX are trademarks of Digital Equipment Corporation.

non-adjacent sensor elements were found to be negligibly small (about 5000 times weaker than the rest of the connections), a simpler network with almost seven times fewer unknowns could be used in future implementations. This network would take much less time and data to train and would be easier to implement. Since such a network could be implemented with any hardware capable of addition and multiplication, a micro-controller such as the one described in Chapter 3 would be sufficient to carry out the necessary computations. An analog hardware implementation would, however, provide a much faster response.

All the methods described above reduced the estimation error for the total sensor array input. If the characteristics of the sensor array were known but not the sensor input, then a table such as Table 10.7 could be used to choose the method providing the smallest relative error. On the other hand, if the types of inputs that a sensor might receive were known *a priori*, but not the characteristics of the sensor array, then a table such as Table 10.12 could be used to choose the signal processing method which provides the highest improvement factor for those inputs.

If neither the sensor characteristics nor the sensor inputs were known, then the choice of signal processing method depends upon the availability of computational power. If the computational capabilities are moderate, then the variable threshold method would be the simplest solution to implement. If CPU time is unlimited for design, then the artificial neural network would be the best choice for general purpose signal processing.

The five signal processing methods described in this dissertation were developed to compensate for the most severe drawback of the example sensor array, the cross-talk. Since the objective of the signal processing study was to recover the total sensor input, the methods were optimized for this goal. If another quantity, such as the area of contact, is to be measured, then a similar optimization procedure should be used to form the best signal processor for determination of that quantity. This may not necessarily be the same result as obtained for the original variable. The ultimate choice of method, however, must be based upon comparison of the performance of many sensors under many conditions likely to occur during the intended usage of the sensor array.

In summary, it is possible to improve the output signals from sensor arrays, causing them to resemble those from sensors with characteristics closer to ideal. Many techniques for signal processing of sensor arrays can be developed and implemented with either digital or analog circuits so that they can be integrated with sensor arrays. The five signal processing methods described in this dissertation were listed in order of complexity and, as one might expect, the best results were obtained using the most sophisticated method. The neural network signal processor provided the best results for all the sensor arrays studied and for a majority of the input patterns. Since a high degree of computational power is required only during training, and the network can be implemented after training using simple computers, the artificial neural network signal processor should be the method of choice.

REFERENCES

Anderson, Charles W., "Strategy Learning with Multilayer Connectionist Representations," Proceedings of the International Workshop on Machine Learning, University of California, Irvine, June 22-25, 1987.

Bartels, R.H. and G.W. Stewart, "Algorithm 432: Solution of the Matrix Equation $AX + XB = C[F4]$," Communications of the ACM, 15(9):820-826, September 1972.

Berec, Alojz, Fabrication and Evaluation of Microelectronic Capacitive Force Sensor, M.S. Thesis, Case Western Reserve University, Cleveland, Ohio, 1989.

Berec, Alojz, Orhan Soykan and Michael R. Neuman, "A Multielement Capacitive Force Sensor," Proceedings of the 40th ACEMB, Niagara Hilton Hotel, Niagara Falls, New York, p. 101, September 10-12, 1987.

Buttazzo, Giorgio, Paolo Dario and Ruzena Bajcsy, "Finger Based Explorations," SPIE Vol 726 Intelligent Robots and Computer Vision: Fifth in a Series, pp. 338-345, 1986.

Crago, Patrick E., Howard J. Chizeck, Michael R. Neuman and F. Terry Hambrecht, "Sensors for Use with Functional Neuromuscular Stimulation," IEEE Transactions on Biomedical Engineering, Vol. BME-33(2):256-268, February 1986.

Dario, Paolo and Giorgio Buttazzo, "An Anthropomorphic Robot Finger for Investigating Artificial Tactile Perception," The International Journal of Robotics Research, 6(3):25-48, Fall 1987.

Devore, Jay L., Probability and Statistics for Engineering and the Sciences, Second Edition, Brooks/Cole Publishing Company, Monterey, California, 1987.

Driels, M.R., "Pose Estimation using Tactile Sensor Data for Assembly Operations," Proceedings of the IEEE, pp. 1255-1261, 1986.

Fearing, Ronald S. and John M. Hollerbach, "Basic Solid Mechanics for Tactile Sensing," The International Journal of Robotics Research, 4(3):40-54, Fall 1985.

Fearing, R.S., A. Rise and T.O. Binford, "A Tactile Sensing Finger Tip for a Dextrous Hand," SPIE Vol. 726 Intelligent Robots and Computer Vision: Fifth in a Series, pp. 378-387, 1986.

Fearing, R.S., "Some Experiments with Tactile Sensing during Grasping," Proceedings of the IEEE, pp. 1637-1643, 1987.

Frei, Werner and Chung-Ching Chen, "Fast Boundary Detection: A Generalization and a New Algorithm," IEEE Transactions on Computers, Vol C-26(10):988-998, October 1977.

Gerald, Curtis F, Applied Numerical Analysis, Second Edition, Addison-Wesley Publishing Company, Inc., May 1980.

Golub, G.H., S. Nash and C. van Loan, "A Hessenberg-Schur Method for the Problem $AX + XB = C$," IEEE Transactions on Automatic Control, Vol. AC-24(6):909-913, December 1979.

Harmon, Leon D., "Robotic Taction for Industrial Assembly," The International Journal of Robotics Research, 3(1):72-76, Spring 1984.

Haralick, Robert M., "Digital Step Edges from Zero Crossing of Second Directional Derivatives," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-6(1):58-68, January 1984.

Hopfield, John J., "Learning Algorithms and Probability Distributions in Feed-Forward and Feed-Back Networks," Proceedings of the National Academy of Science, USA (Biophysics), 84:8429-8433, December 1987.

Hopfield, John J., "Neurons with Graded Response Have Collective Computational Properties Like Those of Two-State Neurons," Proceedings of the National Academy of Science, USA (Biophysics), 81:3088-3092, May 1984.

Hopfield, John J., "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," Proceedings of the National Academy of Science, USA (Biophysics), 79:2554-2558, April 1982.

Jacobsen, S.C., J.E. Wood, D.F. Knutti and K.B. Biggars, "The UTAH/MIT Dextrous Hand: Work in Progress," The International Journal of Robotics Research, 3(4):21-49, Winter 1984.

Jomppins, C.B. and W.L. Wilson, Elementary Numerical Analysis, Prentice-Hall, New Jersey, 1969.

Kinoshita, Gen-ichiro, Shuhei Aida and Masahiro Mori, "A Pattern Classification by Dynamic Tactile Sense Information Processing," Pattern Recognition, 7:243-251, 1975.

Kinoshita, Gen-ichiro, "Representation and Tactile Sensing of 3-D Objects by a Gripper Finger," Robotica, 1:217-222, 1983.

Luenberger, D.G., Linear and Non-Linear Programming, Addison-Wesley Publishing Company, Taiwan, 1984.

McAlpine, George A., "Tactile Sensing," Sensors, pp. 7-16, April 1986.

Neuman, Michael R., Patrick E. Crago, P. Hunter Peckham, Ronald Riso and Howard J. Chizeck, "Artificial Sensory Transducers," Contract Number NO-1-NS-6-2302, Quarterly Progress Report Number 1, The National Institute of Neurological and Communicative Disorders and Stroke, Bethesda, Maryland, November 11, 1986.

Neuman, Michael R., Patrick E. Crago, P. Hunter Peckham, Ronald Riso and Howard J. Chizeck, "Artificial Sensory Transducers," Contract Number NO-1-NS-6-2302, Quarterly Progress Report Number 2, The National Institute of Neurological and Communicative Disorders and Stroke, Bethesda, Maryland, February 11, 1987.

Neuman, Michael R., Patrick E. Crago, P. Hunter Peckham, Ronald Riso and Howard J. Chizeck, "Artificial Sensory Transducers," Contract Number NO-1-NS-6-2302, Quarterly Progress Report Number 3, The National Institute of Neurological and Communicative Disorders and Stroke, Bethesda, Maryland, May 11, 1987.

Neuman, Michael R., Patrick E. Crago, P. Hunter Peckham, Ronald Riso and Howard J. Chizeck, "Artificial Sensory Transducers," Contract Number NO-1-NS-6-2302, Quarterly Progress Report Number 4, The National Institute of Neurological and Communicative Disorders and Stroke, Bethesda, Maryland, August 11, 1987.

Neuman, Michael R., Patrick E. Crago, P. Hunter Peckham, Ronald Riso and Howard J. Chizeck, "Artificial Sensory Transducers," Contract Number NO-1-NS-6-2302, Quarterly Progress Report Number 5, The National Institute of Neurological and Communicative Disorders and Stroke, Bethesda, Maryland, November 11, 1987.

Neuman, Michael R., Patrick E. Crago, P. Hunter Peckham, Ronald Riso and Howard J. Chizeck, "Artificial Sensory Transducers," Contract Number NO-1-NS-6-2302, Quarterly Progress Report Number 6, The National Institute of Neurological and Communicative Disorders and Stroke, Bethesda, Maryland, February 11, 1988.

Neuman, Michael R., Patrick E. Crago, P. Hunter Peckham, Ronald Riso and Howard J. Chizeck, "Artificial Sensory Transducers," Contract Number NO-1-NS-6-2302, Quarterly Progress Report Number 7, The National Institute of Neurological and Communicative Disorders and Stroke, Bethesda, Maryland, May 11, 1988.

Neuman, Michael R., Patrick E. Crago, P. Hunter Peckham, Ronald Riso and Howard J. Chizeck, "Artificial Sensory Transducers," Contract Number NO-1-NS-6-2302, Quarterly Progress Report Number 8, The National Institute of Neurological and Communicative Disorders and Stroke, Bethesda, Maryland, August 11, 1988.

Neuman, Michael R., Patrick E. Crago, P. Hunter Peckham, Ronald Riso and Howard J. Chizeck, "Artificial Sensory Transducers," Contract Number NO-1-NS-6-2302, Quarterly Progress Report Number 9, The National Institute of Neurological and Communicative Disorders and Stroke, Bethesda, Maryland, November 11, 1988.

Neuman, Michael R., Patrick E. Crago, P. Hunter Peckham, Ronald Riso and Howard J. Chizeck, "Artificial Sensory Transducers," Contract Number NO-1-NS-6-2302, Quarterly Progress Report Number 10, The National Institute of Neurological and Communicative Disorders and Stroke, Bethesda, Maryland, February 11, 1989.

Neuman, Michael R., Patrick E. Crago, P. Hunter Peckham, Ronald Riso and Howard J. Chizeck, "Artificial Sensory Transducers," Contract Number NO-1-NS-6-2302, Quarterly Progress Report Number 11, The National Institute of Neurological and Communicative Disorders and Stroke, Bethesda, Maryland, May 11, 1989.

Pao, Yoh-Han, Adaptive Pattern Recognition and Neural Net Implementations, Addison-Wesley Publishing Company, Inc., 1988.

Pati, Y.C., D. Friedman, P.S. Krishnaprasad, C.T. Yao, M.C. Peckerar, R. Yang and C.R.K. Marrian, "Neural Networks for Tactile Perception," University of Maryland, Systems Research Center, Technical Research Report 87-123, 1987.

Pratt, W.K., Digital Image Processing, John Wiley & Sons, 1978.

Press, W.H., B.P. Flannery, S.A. Teukolsky and W.T. Wetterling, Numerical Recipes: The Art of Scientific Computing, Cambridge University Press, New York, 1986.

Robinson, Güner S., "NOTE: Edge Detection by Compass Gradient Masks," Computer Graphics and Image Processing, 6:492-501, 1977.

Rumelhart, D.E., G.E. Hinton and R.J. Williams, "Learning Internal Representations by Error Propagation," Chapter 8 in D.E. Rumelhart & J.L. McClelland (Eds.), Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations, MIT Press, 1986.

Sayeh, M.R. and J.Y. Han, "Pattern Recognition Using a Neural Network," SPIE Vol. 848 Intelligent Robots and Computer Vision: Sixth in a Series, pp. 281-285, 1987.

Shen, J.T., C.W. Thomas and K.A. Loparo, "An Active Touch Processing System," Proceedings of Society of Manufacturing Engineers Sensors '86 Conference, Detroit, Michigan, Publication Number MS86-940, November 11-13, 1986.

Shiozaki, Akira, "Edge Extraction Using Entropy Operator," Computer Vision, Graphics, and Image Processing, 36:1-9, 1986.

Siegel, David M., "Quantifying the Performance of a Tactile Sensor: A Case Study," SPIE Vol. 726 Intelligent Robots and Computer Vision: Fifth in a Series, pp. 369-377, 1986.

Siegel, David, Iñaki Garabieta and John M. Hollerbach, "An Integrated Tactile and Thermal Sensor," Proceedings of the 1986 IEEE Conference on Robotics and Automation, San Francisco, California, pp. 1286-1291, 1986.

Soykan, Orhan, Alojz Berec and Michael R. Neuman, "Signal Processing for a Capacitive Force Sensor Array," Innovation et Technologie en Biologie et Medecine, 10(1):78-86, 1989.

Soykan, Orhan and Michael R. Neuman, "Signal Processing to Improve Sensor Performance," Sensors Expo Proceedings, O'Hare Exposition Center, Chicago, Illinois, pp. 308B1-2, September 13-15, 1988.

Soykan, Orhan, Howard J. Chizeck and Michael R. Neuman, "Signal Processing for a Force Sensor Array," Proceedings of the 40th ACEMB, Niagara Hilton Hotel, Niagara Falls, New York, p. 223, September 10-12, 1987.

Soykan, Orhan, ESYS 416 Term Project, Case Western Reserve University, unpublished, Fall, 1986.

Stansfield, S.A., "Primitives, Features, and Exploratory Procedures: Building a Robot Tactile Perception System," Proceedings of IEEE Robotics and Automation, pp. 1274-1279, 1986.

Tabatabai, Ali J., and O. Robert Mitchell, "Edge Location to Subpixel Values in Digital Imagery," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-6(2):188-201, March 1984.

Thakoor, A.P., A. Moopenn, John Lambe and S.K. Khanna, "Electronic Hardware Implementations of Neural Networks," Applied Optics, 26(23):5085-5092, 1 December, 1987.

Wozencraft, J.M., I.M. Jacobs, Principles of Communication Engineering, John Wiley and Sons, New York, 1965.

Yang, R, "Tactile Perception for Multifingered Hands," University of Maryland, Systems Research Center, Technical Research Report 87-162, 1987.

Yang, R. and P.S. Krishnaprasad, "Tactile Sensing and Inverse Problems," University of Maryland, Systems Research Center, Technical Research Report 87-126, 1987.

Appendix A. CIRCUIT DIAGRAMS

Nor : Normal Operation
Cal : Calibration

Appendix B. GENERALIZED DELTA RULE
(The Perceptron Algorithm)

The total energy of a neural network at any iteration step,

$$E = \frac{1}{2} \sum_k (t_k - o_k)^2 \quad (1)$$

where, E is the total energy of the system,

t_k is target output for the k -th output neuron,

o_k is the current output of the neuron k .

According to Steepest Descent principle, to minimize the total system error, connection ω_{kj} should be changed by,

$$\Delta \omega_{kj} = -\eta \frac{\partial E}{\partial \omega_{kj}} \quad (2)$$

where, η is user determined step size. The input-output relationship of neuron is given by,

$$o_k = f(\text{net}_k) \quad (3)$$

where net_k stands for the total input of the neuron including the threshold input. Net input of the neuron is given by,

$$\text{net}_k = \sum_j \omega_{kj} o_j \quad (4)$$

where ω_{kj} is the connection from neuron k to neuron j and

o_j is the output of the neuron j . Using chain rule,

$$\frac{\partial E}{\partial \omega_{kj}} = \frac{\partial E}{\partial \text{net}_k} \frac{\partial \text{net}_k}{\partial \omega_{kj}} \quad (5)$$

and,

$$\frac{\partial \text{net}_k}{\partial \omega_{kj}} = \frac{\partial}{\partial \omega_{kj}} \sum_j \omega_{kj} o_j = o_j \quad (6)$$

Defining,

$$\delta_k = - \frac{\partial E}{\partial \text{net}_k} \quad (7)$$

Combining equations 2, 6 and 7,

$$\Delta \omega_{kj} = \eta \delta_k o_j \quad (8)$$

Rewriting equation 7 using chain rule,

$$\delta_k = - \frac{\partial E}{\partial \text{net}_k} = - \frac{\partial E}{\partial o_k} \frac{\partial o_k}{\partial \text{net}_k} \quad (9)$$

Differentiating equation 1 with respect to o_k ,

$$\frac{\partial E}{\partial o_k} = - (t_k - o_k) \quad (10)$$

Differentiating o_k with respect to the neuron input,

$$\frac{\partial o_k}{\partial \text{net}_k} = f'_k (\text{net}_k) \quad (11)$$

Substituting equations 10 and 11 into 9,

$$\delta_k = (t_k - o_k) f'_k (\text{net}_k) \quad (12)$$

Substituting equation 12 into 8,

$$\Delta \omega_{kj} = \eta (t_k - o_k) f'_k (\text{net}_k) o_j \quad (13)$$

Equation 13 can be used to determine the increments needed for the connections between the hidden and the output layers. A similar procedure can be determined for the rest of the connections such as the connections between the hidden layers and connections from the input layers to the first hidden layer. Consider a connection ω_{ji} to hidden layer neuron j to from neuron i which is in the layer below. Rewriting equation 2,

$$\begin{aligned}
\Delta \omega_{j1} &= -\eta \frac{\partial E}{\partial \omega_{j1}} & (14) \\
&= -\eta \frac{\partial E}{\partial \text{net}_j} \frac{\partial \text{net}_j}{\partial \omega_{j1}} \\
&= -\eta \frac{\partial E}{\partial \text{net}_j} o_i \\
&= \eta \left(-\frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial \text{net}_j} \right) o_i \\
&= \eta \left(-\frac{\partial E}{\partial o_j} \right) f'_k(\text{net}_j) o_i \\
&= \eta \delta_j o_i
\end{aligned}$$

$$\text{where } \delta_j = \left(-\frac{\partial E}{\partial o_j} \right) f'_k(\text{net}_j) \quad (15)$$

$-\frac{\partial E}{\partial o}$ can be calculated as

$$\begin{aligned}
-\frac{\partial E}{\partial o_j} &= - \sum_k \frac{\partial E}{\partial \text{net}_k} \frac{\partial \text{net}_k}{\partial o_j} & (16) \\
&= \sum_k \left(- \frac{\partial E}{\partial \text{net}_k} \right) \frac{\partial}{\partial o_j} \sum \omega_{km} o_m \\
&= \sum_k \left(- \frac{\partial E}{\partial \text{net}_k} \right) \omega_{kj} \\
&= \sum_k \delta_k \omega_{kj}
\end{aligned}$$

Substituting equation 16 into equation 15,

$$\delta_j = f'_k(\text{net}_j) \sum_k \delta_k \omega_{kj} \quad (17)$$

In general, the increment needed for any connection ω_{ji} for training pattern p is

$$\Delta_p \omega_{ji} = \eta \delta_{pj} o_{pi} \quad (18)$$

where δ must be calculated according to the location of the receiving neuron. If the receiving neuron is at the output layer, then

$$\delta_{pk} = (t_{pk} - o_{pk}) f'_k(\text{net}_{pk}) \quad (19)$$

otherwise,

$$\delta_{pj} = f'_j(\text{net}_{pj}) \sum_k \delta_{pk} \omega_{kj} \quad (20)$$

The $f'(\text{net}_p)$ can be calculated as follows. Consider the output of any neuron j ,

$$o_j = \frac{1}{1 + \exp [- (\sum_i \omega_{ji} o_i + o_j)]} \quad (21)$$

Differentiating this output with respect to the neuron input,

$$\frac{\partial o_j}{\partial \text{net}_j} = o_j (1 - o_j) \quad (22)$$

Substituting equation 22 into equations 19 and 20,

$$\delta_{pk} = (t_{pk} - o_{pk}) o_{pk} (1 - o_{pk}) \quad (23)$$

$$\delta_{pj} = o_{pj} (1 - o_{pj}) \sum_k \delta_{pk} \omega_{kj} \quad (24)$$

Therefore, the total update equation for ω_{ji} will be

$$\Delta\omega_{ji}(n+1) = \eta(\delta_j o_i) + \alpha\Delta\omega_{ji}(n) \quad (25)$$

where the α is the momentum constant and $\Delta\omega_{ji}$ is

$$\Delta\omega_{ji} = \sum_p \Delta_p \omega_{ji} \quad (26)$$

Appendix C. Test Element Set

Object Number 1

Object Number 2

Object Number 3

Object Number 4

Object Number 5

Object Number 6

Object Number 7

Object Number 8

Object Number 9

Object Number 10

Object Number 11

Object Number 12

Object Number 13

Object Number 14

Object Number 15

Object Number 16

Object Number 17

Object Number 18

Object Number 19

Object Number 20

Object Number 21

Object Number 22

R O T A T I O N S

Object Number 23

Object Number 24

Object Number 25

Object Number 26

Object Number 27

Object Number 28

Object Number 29

Object Number 30

Object Number 31

Object Number 32

Object Number 33

Object Number 34