GRASP CONTROL USING TACTILE SENSORY FEEDBACK

A Thesis
Submitted to the College of Graduate Studies and Research
in Partial Fulfilment of the Requirements
for the Degree of
Master of Science
in the
Department of Electrical Engineering
University of Saskatchewan

by

DONALD UWEMEDIMO EKONG
Saskatoon, Saskatchewan
August 1991

Dedicated to

my parents,

Donald and Afiong Ekong,

for their love, encouragement and support.

UNIVERSITY OF SASKATCHEWAN

DEPARTMENT OF ELECTRICAL ENGINEERING

CERTIFICATION OF THESIS WORK

We, the undersigned, certify that Mr. Donald Ekong, candidate for the Degree of Master of Science in Electrical Engineering, has presented his thesis on the subject Grasp Control Using Tactile Sensory Feedback that this thesis is acceptable in form and content, and that this student demonstrated a satisfactory knowledge of the field covered by his thesis in an oral examination held on August 29, 1991.

External Examiner: [signature]

of Control Engineering

Internal Examiners:

[signature]

[signature]

[signature]

[signature]

Date: August 29, 1991

COPYRIGHT

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Dr. Hugh C. Wood, for his guidance and continuous encouragement throughout the course of this work.

I also thank Dr. J. N. Wilson, Dr. R. J. Bolton, Dr. R. E. Gander, and Dr. K. Takaya, for serving on the examining committee, and providing valuable comments on the thesis.

I am grateful to Professor A. E. Krause for his comments on the data acquisition system.

I would also like to thank the LORD, my parents, sisters and friends, for their support and encouragement.

Financial support was gratefully received from Dr. Hugh C. Wood, through NSERC grant 0004522, my father, Dr. D.E.U. Ekong, my uncle, Dr. C.E.U. Ekong, and the Department of Electrical Engineering.

UNIVERSITY OF SASKATCHEWAN

Electrical Engineering Abstract 91A348

GRASP CONTROL USING TACTILE SENSORY FEEDBACK

Student: DONALD U. EKONG Supervisor: Dr. H.C. WOOD

M. Sc. Thesis Submitted to the
College of Graduate Studies and Research
August 1991

ABSTRACT

The design of a grasp control system for an existing three-fingered robot gripper is presented with the results of laboratory testing on a prototype system built for this thesis project.

The system can detect when an object has been grasped by the gripper, and when the object starts to slip. It does this by sampling the sensors of the gripper fingers. Sampling can be performed on the sensors of three fingers, two fingers, or one finger. When slip has been detected, the system takes control action to stop further slip by increasing the grip on the object.

Related work (University of Saskatchewan, Mechanical Engineering Abstract 89A001) covers detailed analysis of the gripper.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

In order for manufacturers to remain competitive, both locally and internationally, new techniques of manufacturing have to be adopted to cope with the changing demands of consumers. In areas where there is either labour shortage or high cost of labour, the new techniques include the adoption of flexible automated manufacturing systems which make use of industrial robots.

An industrial robot is essentially a mechanical device that can be programmed to automatically move objects through different configurations in space [1]. It can also be described as an automatically controlled, reprogrammable, multipurpose machine, with or without locomotion, which is capable of manipulating parts or tools for use in industrial automation.

There are many ways of classifying robots. One form of classification is according to increasing levels of capability [2, 3]. Under this form of classification, robots are divided into the following groups : Operating, Sequence controlled, Playback, Numerically controlled and Intelligent robots. An Operating robot is one that is remotely controlled by an operator such as a teleoperated robot. Sequence controlled robots are stand-alone units that operate in sequence with preset data that is easily modified. A Playback robot is one that is trained by a human operator and then repeatedly performs the required steps in sequence without further operator interaction. With a Numerically controlled robot, the human operator controls the robot through changing a program or entering numbers, rather than through a training mode. An Intelligent robot is one that has the ability to sense its environment and adapt to changing conditions as it completes its task.

Industrial robots have an advantage over conventional machines in that a change in design or model of a product means major changes in the machine or the need for a new machine. When robots are used, all that would be needed is a change in the operating program, thus introducing flexibililty into the manufacturing system.

1.1. Robot System

A typical robot system can be divided into several parts [3], as shown in Figure 1.1:

1. Robot arm - includes the links, joints and wrist
2. End-effector - performs the work by serving as a gripper or tool.
3. Actuator - provides the mechanical power to move the robot arm. Actuators are primarily electric but they could also be hydraulic or pneumatic.
4. Transmission system - couples the actuator to the joints through connections such as cables, belts or gears.
5. Internal sensors - monitor the motion of the robot.
6. External sensors - collect information about the surrounding environment and provide this data to the robot.
7. Control system - computer directing overall activity and the necessary interface with an operator for monitoring, reprogramming or training.

1.2. Areas Of Application Of Robots

Robots are used in many industrial applications. Some of these applications are parts handling, part modification, spray painting, assembly, agriculture and food industry and construction [3].

One of the earliest uses for robots was in parts handling. This included loading and unloading of die casting machines. In parts handling, the robot simply moves an item from point A to point B. The robot can be used for unloading incoming supplies or loading finished goods, which is known as palletising. Parts handling robots are usually pick-and-place types. Vision systems have been incorporated into some of the more complex bin-picking tasks.

Robots that perform part modification functions make use of special tools at their end effectors. They are characterised by very high accuracy requirements. In many cases, specialised sensors are included to help the robot locate the position to be worked on. Part modification applications include drilling, cutting and welding.

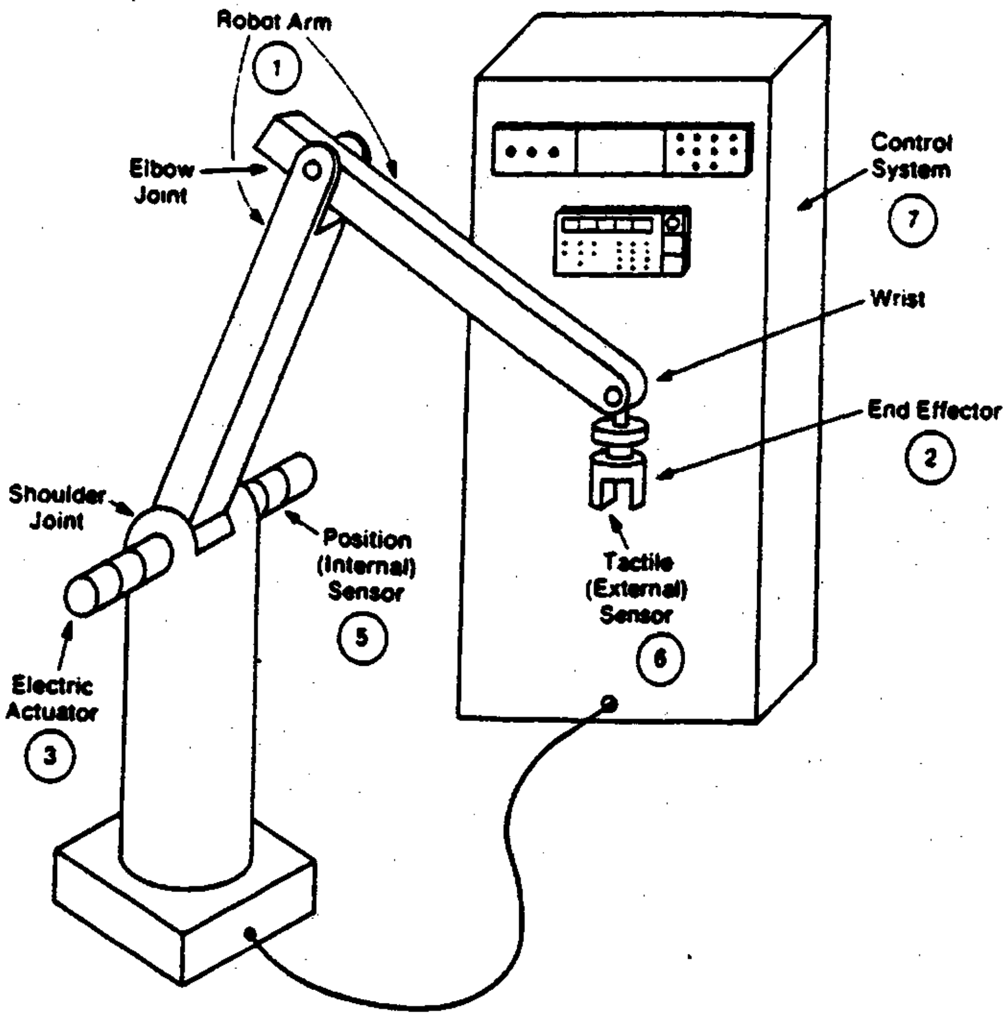


Figure 1.1: Typical robot system (From Poole [3], Fig. 1-7, pp 25)

Spray painting is an excellent task to automate, because the paint spray is hazadous and the job is monotonous yet it requires reasonable repeatability to ensure a quality finish. A characteristic of a spray-painting robot is its need for manual teach-control programming. Contrary to most other robot applications, spray painting works best if the robot is shown what to do (via teach mode) rather than being programmed via offline programming.

Assembly operations include parts insertion, parts fastening, and labelling. An example of parts insertion is the picking up of electronic components and inserting them into a printed circuit board. Robots that are used to assemble printed circuit wafers are used in a clean room, and special care is given to the external robot design to avoid contaminating the air.

Robots which are used in the agriculture and food industry are mostly used for picking and sorting. One example is the citrus grading machine developed by Sunkist Corporation which handles about eight pieces of fruit per second and can grade oranges or lemons according to size, colour, blemishes and frost injury. Another application combines a conveyer belt system with machine vision for handling and sorting fruits and vegetables [3].

At the University of Western Australia, a sheep-shearing robot has been developed and tested [3]. After the sheep is placed in a holding pen, the robot is able to shear the entire sheep. The robot has a collision-avoidance system that controls the angle of the cutter and its closeness to the skin.

Robots have been used in only a few applications in the construction field, but it is one of the key areas expected to expand greatly over the next few years. Concrete finishing robots are a current example [3]. These robots are placed on wet cement to trowl and finish the poured concrete floor. Specially designed wheels ensure that they evenly spread their weight and do not sink into the concrete.

The introduction of robots into the manufacturing system has the effect of increasing the number of products that are made, lowering production cost and increasing production efficiency which leads to a rise in the competitiveness of the company [2]. Robots can replace human beings in dangerous, repetitive and monotonous tasks that workers dislike.

The development of the microprocessor has played an important role in the rapid growth of industrial robots. It has also lead to a considerable amount of work in the areas of computer vision, tactile sensing, proximity sensing and the development of control systems to process sensory information. Present day research in the field of robotics involves the design of "intelligent" robots with artificial senses of sight and touch which make them able to adapt to changes in the environment.

1.3. Objective Of The Project

Many robot tasks require the robot to react to changes at the gripper-object contact. These changes include making and breaking of contacts, and motions of the object. To achieve this requirement, sensors are incorporated into the gripper design to provide information on changes at the gripper-object contact, and algorithms are developed to interpret the tactile information, and to control the grasp.

The objective of this project is to design, for a robot gripper, a system that uses tactile sensory feedback to control grasping, and to test the system for a variety of grasping situations. The system should be able to detect when an object has been gripped by the gripper, and when the object starts to slip. Once slip has been detected, the gripper will adjust its grip on the object. The gripper that is to be used for the testing and analysis in this project was designed and built in the Departments of Electrical and Mechanical Engineering, University of Saskatchewan.

1.4. Outline Of The Thesis

A definition of tactile sensing and previous work done in that area are discussed in Chapter 2. Chapter 3 contains a description of the experimental apparatus. The experiments and results are presented in Chapters 4 and 5. The summary, conclusion and recommendations for further work are given in Chapter 6.

2. TACTILE SENSING FOR GRASP CONTROL

2.1. Introduction

Tactile sensing, or touch sensing, is the ability to sense conditions at the finger-object contact. Human dexterity is a marvelous thing; our hands can grasp a wide variety of shapes and sizes, perform complex tasks and switch between grasps in response to changing task requirements. This is due in part to the physical structure of our hands and in part to our sophisticated control capabilities. In large measure, this control capability is due to tactile sensing. People become clumsy when deprived of reliable tactile information through numbness of anaesthetized or cold fingers even though their other capabilities are intact [4].

The need for tactile sensing occurs in many robotic applications such as in the general problem of locating, identifying, and organizing parts that need to be assembled. Computer vision systems can be used to perform these functions but many cases require positional information that cannot be provided because of deficiencies inherent in computer vision systems. Some of these deficiencies [3, 5] are listed below :

1. The first deficiency is accuracy. In a typical parts handling operation, the vision system could position the robot to within 6mm while maintaining a field view of 1.5m horizontally. Such positioning accuracy is quite good for a vision system, but it may not be adequate for the precise positioning needed to insert a part into a machine tool. This is because without touch, last minute minor corrections to the robot's movement are not practical because the gripper can block the camera's view.

2. The second deficiency of computer vision arises from the fact that a computer vision system cannot see behind the part. If gripping relies on somehow reaching around the part, blind gripping may result in damage of the parts.

3. The third deficiency is that data from vision systems generally require more processing than information from tactile systems.

Tactile sensing gives important information about object shape and slippage, which are very useful in handling objects.

A disadvantage of tactile sensing is that it cannot give information about a part of the object that the robot is not able to reach.

2.2. Tactile Sensing Systems

The major components of a tactile sensing system [5] are

1. a touch surface
2. a transduction medium which converts local forces or moments into electrical signals
3. structure, and
4. a control/interface

The model of a complete tactile sensing system is shown in Figure 2.1. The functional divisions fall into a pyramidal form, with the more computationally intensive processes occurring at the upper end of the pyramid and the more functionally rigid processes at the bottom. The vertical connections permit data flow downwards for addressing purposes and upwards for signal use by higher subsystems.

2.2.1. Transduction

Transduction techniques for tactile data vary from very simple transducers such as switches, which can be used to indicate simple contact with an object, to complex transducers which measure normal and shear forces, and their spatial derivatives, at the contact surface. Many phenomena have been investigated for use in tactile transducers. Some specific examples of tactile transducers include pneumatic sensors, strain gauges, capacitive coupling devices, optical devices, silicon micromechanical structures, piezoelectric devices, piezoresistive devices and magnetostrictive devices [6, 7]. Transducers should be sufficient in number to permit an accurate reconstruction of contact detail.

Fearing [8] describes two different approaches to obtaining contact information

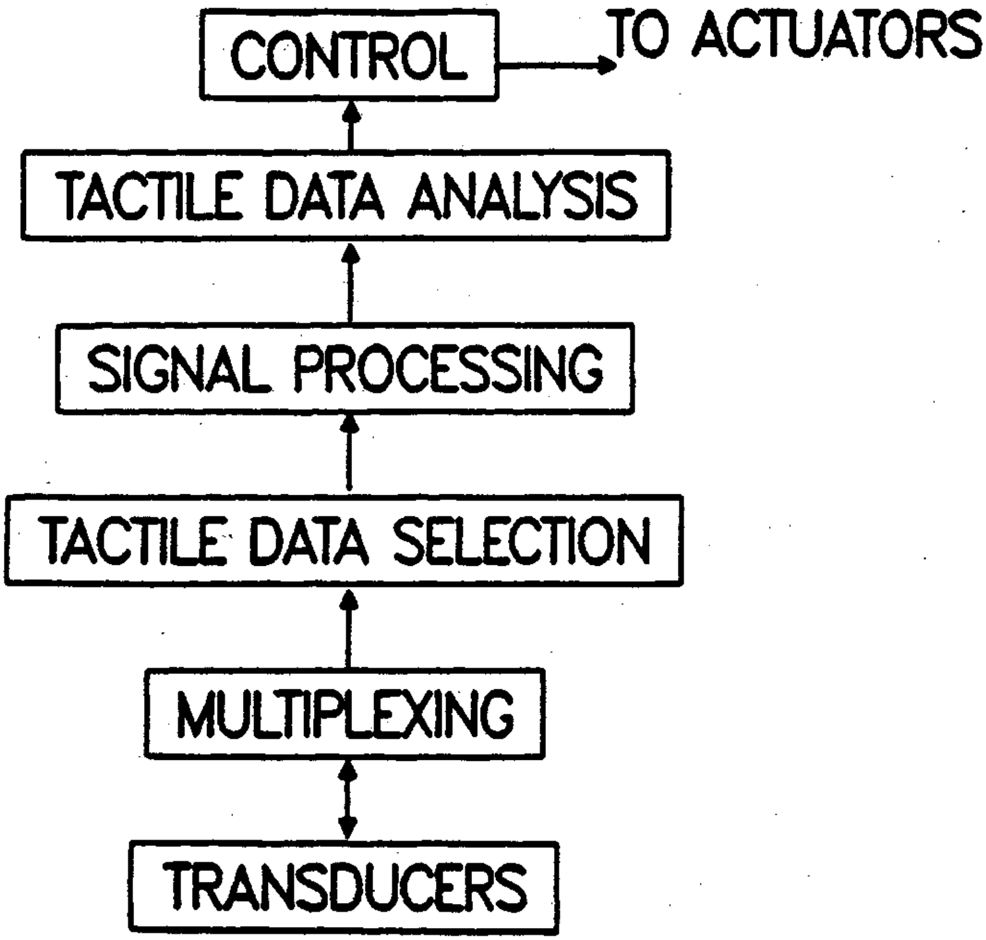


Figure 2.1: General Model of a Tactile Sensing System

from a finger tip sensor. The first approach involves a sensor that gives location and resultant force using a strain gauge structure, and the second approach involves the use of arrays of deflection transducers. The array sensor approach has an advantage for determining contact shape from a single measurement. In one sense, N by M measurements with a single element tactile sensor can be equivalent to one measurement at each of N by M sensors. However, objects may move while being manipulated with the fingers. Object motion makes it necessary to have simultaneous measurement over the whole finger surface. It is difficult and time consuming to explore the object using a single tactile sensor to determine contact type if the object must remain grasped.

2.2.2. Multiplexing

Multiplexing involves two stages: the query stage and the access stage [6]. In the query stage the controller specifies the specific sensory data required for an immediate task and correlates the selected data with its known address. In the access stage the controller outputs this address to a multiplexer or series of multiplexers that respond to their corresponding addresses by enabling the appropiate sensory data for output.

2.2.3. Tactile Data Selection

During a manipulation sequence, the controller must make use of tactile data in a variety of forms, and it must be available at data rates that allow for accurate manipulation. Not all the tactile data may be meaningful, and processing time can be greatly reduced by internally updating only the data elements that have been changed or only the sites of interest. Methods of data management that permit the system user to specify (in software) the nature of tactile data selection and to influence the speed at which data processing can be performed have been identified. Some of these methods are Full scan, Reactive scan and Anticipatory scan [6].

During Full scan, the entire sensing array is scanned during each cycle of data acquisition. This method is sufficient for relatively small sensing arrays or for low operating speeds. On a much larger scale, the sensory system can easily have a size that prohibits the use of a full scan of all sensor elements within the necessary time interval. In this case, another method of accessing data is imperative.

Reactive scan is used in cases where the manipulation algorithm may only require information that has changed since the last sensor system scan, for example, when detecting slip during an object transportation sequence or in the early stages of object acquisition. In order to facilitate this process, only the sets of sensory patches having elements that have changed since the last update cycle are refreshed in the sensor state table. The monitoring of each sensor patch can be performed continuously by a local processor. Although this approach does not completely eliminate the possibility that the data from an inactivated sensor would be processed, it will increase the speed at which the sensor state table may be updated.

Anticipatory scan is used when in the time interval immediately prior to the acquisition of an object of known geometry and location, it may be desirable to monitor specific sensor patches at very high speeds in order to determine the exact instant of rendezvous. In this case, only the sensor patches of interest are monitored, regardless of whether or not sensor activation has taken place. Again, if only some of the sensory patches have been selected, sensor data update times will decrease in a manner proportional to the number of patches selected.

It is possible to combine reactive and anticipatory scanning and update only those sensory patches that have been selected and have changed since the last update cycle. Such an approach would result in the highest possible operating speed.

2.2.4. Signal Processing

Signal processing puts the signals from the sensors into a form that is compatible with the rest of the system. The steps required for processing of the raw sensor data include filtering, amplification, attenuation, buffering and digitization. Filtering is required to reduce the noise in the sensor data. Weak sensor signals are amplified while signals that are too high are scaled down or attenuated to match the capabilities of electronics systems. Buffering is a means of temporarily storing data until they can be used for their intended purpose. Sample-and hold circuits are used to store analog signals, while digital storage registers are used to store discrete signals. Digitization is the conversion of data from analog to digital form using analog-to-digital converters.

2.2.5. Tactile Data Analysis

Tactile data analysis, or data processing, involves the extraction, from the tactile data, of information on contact interactions. Computations may also be performed to enhance or identify contact features.

2.2.6. Control

Tactile information from the lower levels is used to control grasp and manipulation. This can be information on the force that is applied to the grasped object, and/or information on whether the object has started to slip. Based on the information received, the control action could be a decrease of the grasping force if the object has been gripped too hard, or an increase in the grasping force if the object has been gripped too lightly. Control action can occur while the object is moved from one place to another.

2.2.7. Computer System

The functions of control, tactile data selection and analysis, and some aspects of processing can be accomplished by use of digital computer(s). The earliest robot systems did not use computers to control the robot. Instead they relied on plugboard connections, relay logic, limit switches, and position stops. Machine sequence control used a relay logic system based on a ladder-type programming enviroment in which each step implied the next. Unexpected jumps in the program, such as might be necessary to cope with input/output from sensors, could not be handled. Because the robots were operations oriented, they could not deal with the complex decisions required with sensors. Control technology has continued to evolve and now uses the latest microprocessors and intelligent sensors to control robot action [3].

2.3. Use Of Slip Sensing In Grasp Control

Slip sensing plays an important role in the ability to successfully grasp and manipulate objects. When an object has been grasped, it is necessary to know if the force applied to the object is sufficient. One way of doing this is to check if the object has started to slip. Once it has been discovered that the object has started to slip, the force on the object can be increased to stop further slip.

Slip sensing in humans has been investigated by Johansson and Westling [4]. They monitored the grasp force and the object acceleration as subjects lifted small objects. Most people use a grip force which is slightly greater than the minimum required for lifting the object. After a disturbance, or when an incorrect estimate of minimum grip force is made, which results in the application of a grip force which is less than the required minimum, the object begins to slip from the hand. This small movement of the object is quickly followed by an increase in the grasp forces until the object stops moving. The increases in the grasp force occur unconsciously and in a time period that is less than 100 milliseconds. By anaesthetizing the surface of the fingertip skin, Johansson and Westling have shown that corrections to the grasp force are prompted by information from cutaneous tactile sensors.

Tactile sensing research has resulted in the development of various methods of sensing slip. Some of these methods have been incorporated into robot end-effectors, such as grippers, for use in grasp control. Howe and Cutkosky [9] have developed a slip sensing device called the Skin Acceleration Sensor. This sensor has an inner core that is made of hard plastic. Around this core is a layer of soft polyurethane foam. The outer covering or skin of the sensor is made of textured silicone rubber, which is relatively stiff. An accelerometer is attached to the inner surface of the skin. This accelerometer measures the local skin accelerations produced when areas of the skin catch and snap back as the sensor moves against a surface. Figure 2.2 shows the skin acceleration sensor and the apparatus for testing its ability to detect slip. The sliding platform, shown in the figure, is covered with photocopy paper. Figure 2.3 (a) and (b) show some results of tests carried out on the sensor. Figure 2.3 (a) shows the position of the platform, with respect to time, while Figure 2.3 (b) shows the sensor output with respect to time. This sensor has been incorporated into a two-fingered manipulator for a simple grasp-lift-replace experiment [10].

Bicchi et al. [11] have applied Intrinsic Tactile sensing to grasp control. An Intrinsic Tactile sensor, or IT sensor, comprises a force/torque sensor built in the interior part of the fingertip of the robot end-effector. The fingertip surface is not equipped with sensors and can therefore be realized in whatever shape and material the hand designer prefers. The IT sensor is able to find the position of the contact centroid on the fingertip surface, the intensity of the normal component of the contact force, the intensity and direction of the tangential (friction) component of the contact force, and the intensity of the torque generated by friction forces. In an experiment, an object was held in a grasp comprising

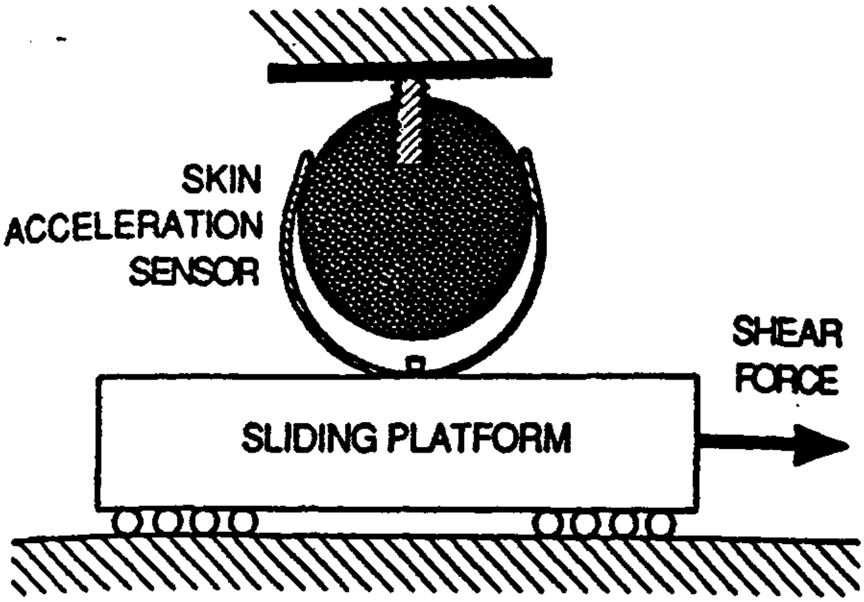
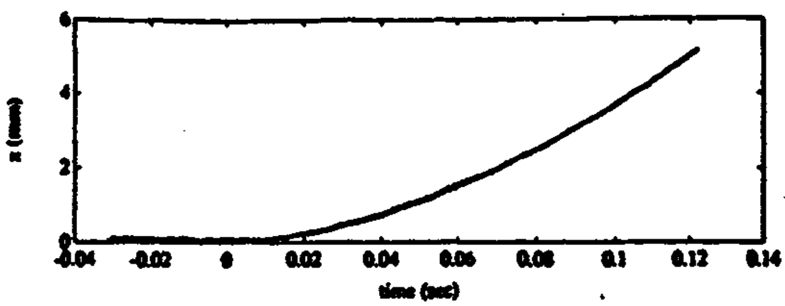
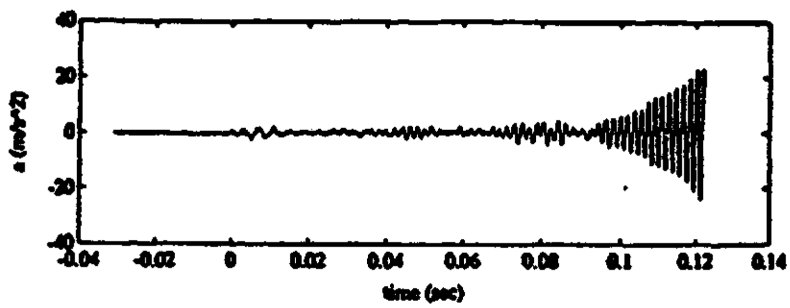


Figure 2.2: Test Apparatus for Skin Acceleration Sensor (From Howe [9], Fig. 5)



(a) Position of sliding platform Vs Time



(b) Sensor output Vs Time

Figure 2.3: Test Results For Skin Acceleration Sensor (From Howe [9], Fig. 6)

three fingers. Only one finger was active, i.e., it was equipped with an IT sensor. The other two fingers, without IT sensing capabilities, had fingertip covers with much higher friction than the active one, and acted only passively to counteract the external disturbances. Figure 2.4 shows a sketch of the finger-object system. Tangential and normal components of the active contact force are represented by f_t and f_n, respectively. Figure 2.5 shows the acceptable and unacceptable contact conditions in the f_n-f_t plane. Angle ϕ = arc tan μ_s, where μ_s is the coefficient of static friction relative to the finger-object contact. μ_s is known with some approximation. The portions of the plane external to angle ϕ correspond to friction ratios higher than μ_s, i.e. to slippage. Disturbances were generated by making the grasped object slip. The system proved to be able to discern and promptly signal when the friction limit was passed. This corresponded to real slip motions.

Bao and van Brussell [12] have incorporated a tactile sensor, developed at the Catholic University of Leuven in Belgium, into the inside surfaces of an off-the-shelf two jaw gripper. In their experiment, each jaw was comprised of a matrix of 16 by 16 cells capable of measuring pressure levels. Slip was detected by looking for changes in (a) pressure levels of the most loaded cell, (b) contact area, and (c) computed center of gravity. A simple digital filter is also used to abate the effect of noise. Figure 2.6 outlines the algorithm for slip detection in pseudocode. Experiments have been carried out, using this system, with different object sizes and satisfactory results were obtained.

At the University of Saskatchewan, a three-fingered robotic gripper has been designed and built by Muench [7, 13]. The tactile sensing system uses arrays of four thick film force sensing resistors at each sensor sight. The signals from each array are analysed and are used to calculate tangential and normal forces at the finger-object contact point. A cylindrical object was grasped by the fingers of the gripper and slip was caused by applying a vertical force on the cylinder. Figure 2.7 shows a plot of the tangential component of the gripping force. As the vertical force applied to the object increases, the tangential force, at the finger-object contact, also increases until the point, labelled A, where the tangential force exceeds the maximum frictional force. At this point, slippage occurs and this is indicated by a momentary decrease in the tangential force due to the difference between the static and dynamic coefficients of friction.

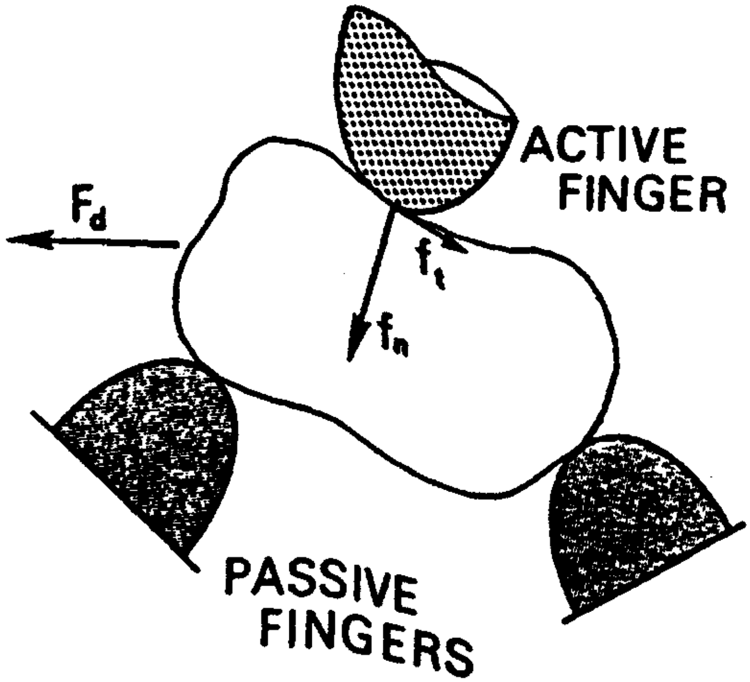


Figure 2.4: IT sensor finger-object system (From Bicchi [11], Fig. 1)

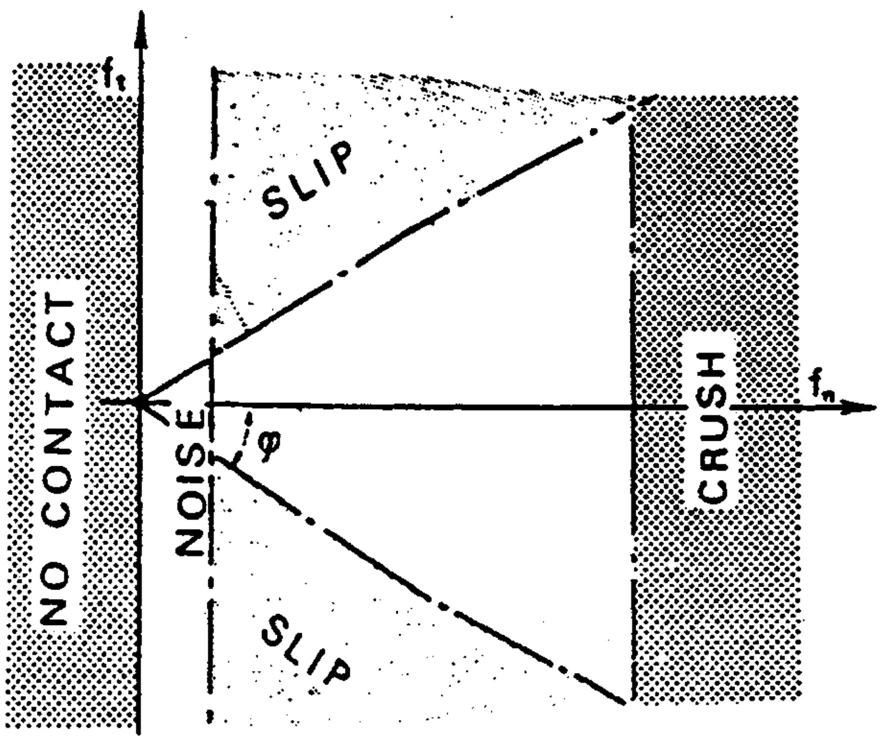


Figure 2.5: Contact conditions for IT sensor (From Bicchi [11], Fig. 2)

```
procedure  slip_detection


   close gripper until contact object

   find out the most loaded cell and record its
     location (I,J)

   calculate the mean pressure P(I,J),  the mean contact
     area A and the mean gravity center (Xc,Yc) over a
       certain sample times

   move gripper up

   repeat while no slip detected

        measure the pressure of cell(I,J) to get a new
          Pnew(I,J)

        calculate a new Anew and a new (Xc,Yc)new

        if  P(I,J)-P(I,J)new > THRP  OR  A-Anew > THRA  OR
            |(Xc,Yc)-(Xc,Yc)new| > THRC
              then slip detected

          else no slip detected
          update P(I,J), A and (Xc,Yc) by using new
            measurements /* filtering */

             (Xc,Yc) = (1-a1)·(Xc,Yc) + a1·(Xc,Yc)new
             P(I,J) = (1-a2)·P(I,J) + a2·Pnew(I,J)
             A = (1-a3)·A + a3·Anew
             /* (0 <) a1, a2 and a3 (< 1) are determined
                by testing */


end
```

Figure 2.6: Slip Detection Algorithm (From Bao [12], Fig. 13)

Patil *, in another experiment that was carried out at the University of Sakatchewan, has done a frequency analysis of the sensor signals during slippage. This was done using a robot arm having a parallel jaw gripper equipped with force sensing resistors. When the grasped object started to slip, there were oscillations in the sensor signals. The frequency of the oscillations was measured and found to be less than 100 Hz. The oscillations in-

---

*Unpublished M.Sc. Thesis

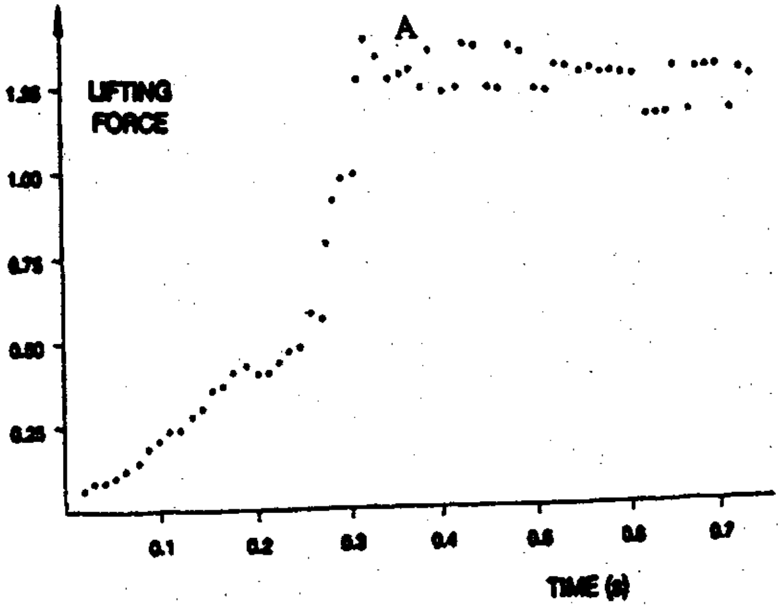


Figure 2.7: Tangential Component of Gripping Force (From Wood [13], Fig. 5)

dicated that the grasped object had started to slip, and control commands were issued to stop further slip.

Dornfeld et al. [14, 15] have used acoustic emission signal analysis for detection of slip-related motion between a workpiece and a robot gripper. Acoustic emission (or AE) is the stress wave generated by a solid undergoing phase transitions, plastic deformations, or fracture. These stress waves travel to the surface of the medium and cause minute displacements detectable by sensitive piezoelectric crystals which convert displacement pulses into a charge or voltage signal, called the acoustic emission signal [14, 15]. Acoustic emission is a high frequency signal ranging from 50 kHz to 1 MHz. The plastic deformation due to relative motion of two materials in contact is a good source of acoustic emission. The experimental setup used for the AE analysis [15] is shown in Figure 2.8. It consists of a two-finger gripper with rectangular fingers. The fingers can close and open by rotating about independent pivots located inside the gripper body. The gripper is pneumatic and the gripping force can be changed by varying the inlet air pressure. The work specimen used in the experiment was a cylindrical steel object polished with

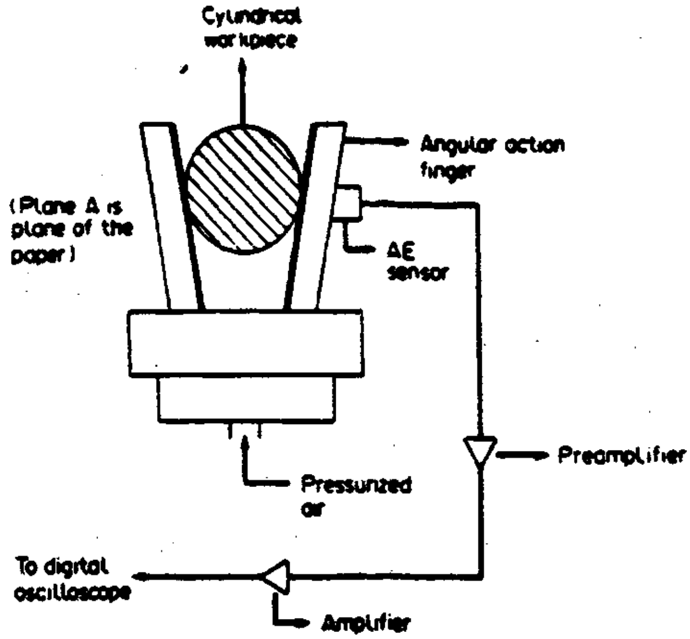
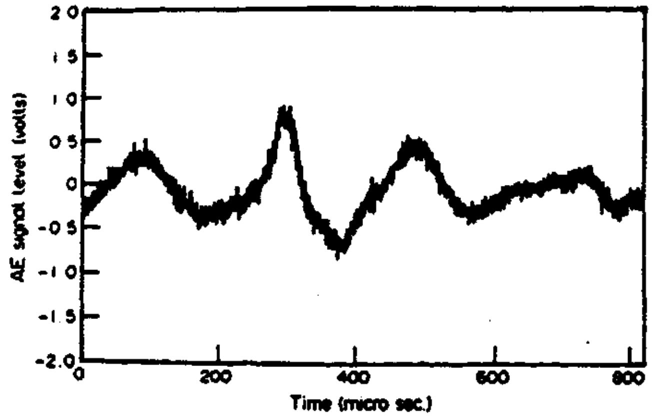
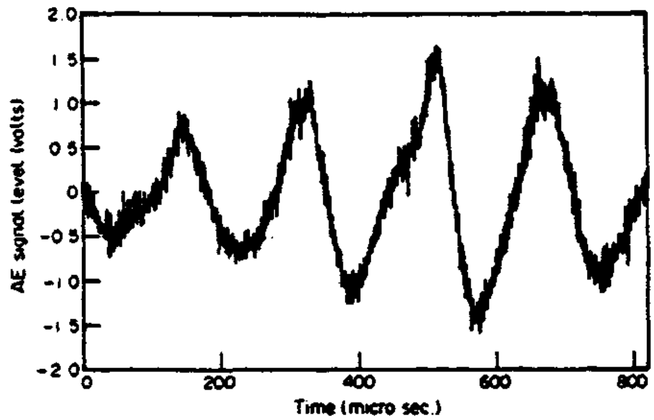


Figure 2.8: Acoustic Emission experimental setup (From Dornfeld [15], Fig. 7)

sandpaper. The transducer output was passed through a preamplifier (40 dB gain, 50 kHz pass filter) and an amplifier (20 dB gain). The amplified AE signal generated by slip was recorded on a digital oscilloscope. Translational slip is characterized by the fact that the cylinder axis always remains normal to the plane defined by the arc of motion of the two fingers (plane A in Figure 2.8). During rotational slip, this axis tilts relative to plane A. The AE generated during translational slip is shown in Figure 2.9 (a) and (b), for gripping pressures of 410 kPa and 830 kPa respectively. In both cases, the signal is periodic, with superimposed high frequency components. The periodic portion of the signal represents the stick-slip related component. The frequency of this component is about 2.5 kHz. The superimposed higher frequency components are due to asperity deformation (deformation occurring due to contact and interaction with a rough surface) and fracture during the slip period of the stick-slip motion. The effect of a higher gripping pressure is to increase the amplitude of the stick-slip related component of the signal, with almost no change in its frequency. Rotational slip produces AE of a distinctly different nature. In the case of rotational slip, no stick-slip motion is present. A typical signal generated during rotational slip at a gripping pressure of 830 kPa is shown in Figure 2.10. The lower frequency stick-slip related component is absent in this case. Only a high frequency burst associated



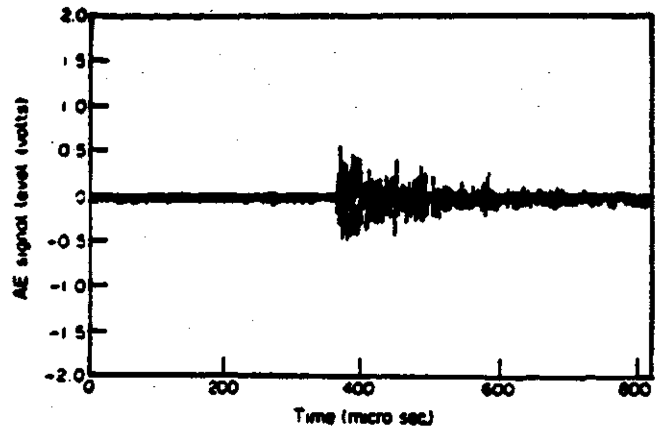
(a) Gripping pressure = 410 kPa (From Dornfeld [15], Fig. 9)



(b) Gripping pressure = 830 kPa (From Dornfeld [15], Fig. 10)

Figure 2.9: AE During Translational Slip

with contact mode transition and subsequent slip is observed. It was also noted during the course of the experiment that rotational slip generated detectable AE only for gripping pressures greater than 690 kPa [15]. The AE information can be used to provide control commands for recovering from unstable configurations.



Gripping pressure = 830 kPa (From Dornfeld [15], Fig. 11)

Figure 2.10: AE During Rotational Slip

2.4. Concluding Remarks

Substantial research efforts have been devoted to the construction of compact, high-resolution tactile sensors that employ sophisticated transduction and processing techniques. Despite the progress in sensor fabrication, relatively few multidetector systems have been used in real manipulation systems. Those designs that have been applied in automated environments have been used in static circumstances for simple contact imaging rather than for active manipulation, and they suffer from being too cumbersome, fragile, or slow [6].

The slow progress in the development of comprehensive tactile sensing systems indicates that the fundamental problem is not only one of tactile sensor design and fabrication. There is limited understanding on how to sense and control the physical phenomena which characterize slip and on the making and breaking of contacts. Advancements in tactile system design will require further understanding of the ways in which contact information can be used to control grasp and aid in task planning.

3. EXPERIMENTAL APPARATUS

For a robot manipulator to perform operations in its environment which are comparable in complexity to those accomplished by human hands, the forces exerted on an object being manipulated must be maintained within certain bounds. Grasping forces exerted on the object must not be so high that the object is damaged, or so low that slip occurs and the object is dropped. Slip detection plays a vital role in the ability to manipulate objects successfully. If the grasped object starts to slip during manipulation, it would be desirable to detect slippage, and make control corrections in time to stop the object from slipping further.

This project presents a grasp control method which uses sensory feedback to improve grasping stability during object manipulation. The grasp control method closely resembles, in principle, that of a human being. When a person grasps an object, the magnitude of the gripping force exerted on the object is not known. As the object is manipulated, the tactile sensors on the hands give information on the position and motion of the object, with respect to the fingers [4]. So, if the object is slipping, the gripping force on the object can be increased to recover stable grasp.

A three-fingered gripper equipped with tactile sensors was used for the experiment. The gripper was calibrated to study the response of the tactile sensors to forces applied on the pad. Then, the gripper gripped an object, and the object was made to slip. During slippage, the tactile sensors were sampled to obtain slip information. This information was used, in real-time, to recover stable grasp.

The experimental setup, shown in Figure 3.1, can be divided into two parts, namely hardware and software. The hardware consists of the gripper and gripper motor, the computer hardware, a Das-8 interface board, a digital-to-analog converter (D/A), a parallel

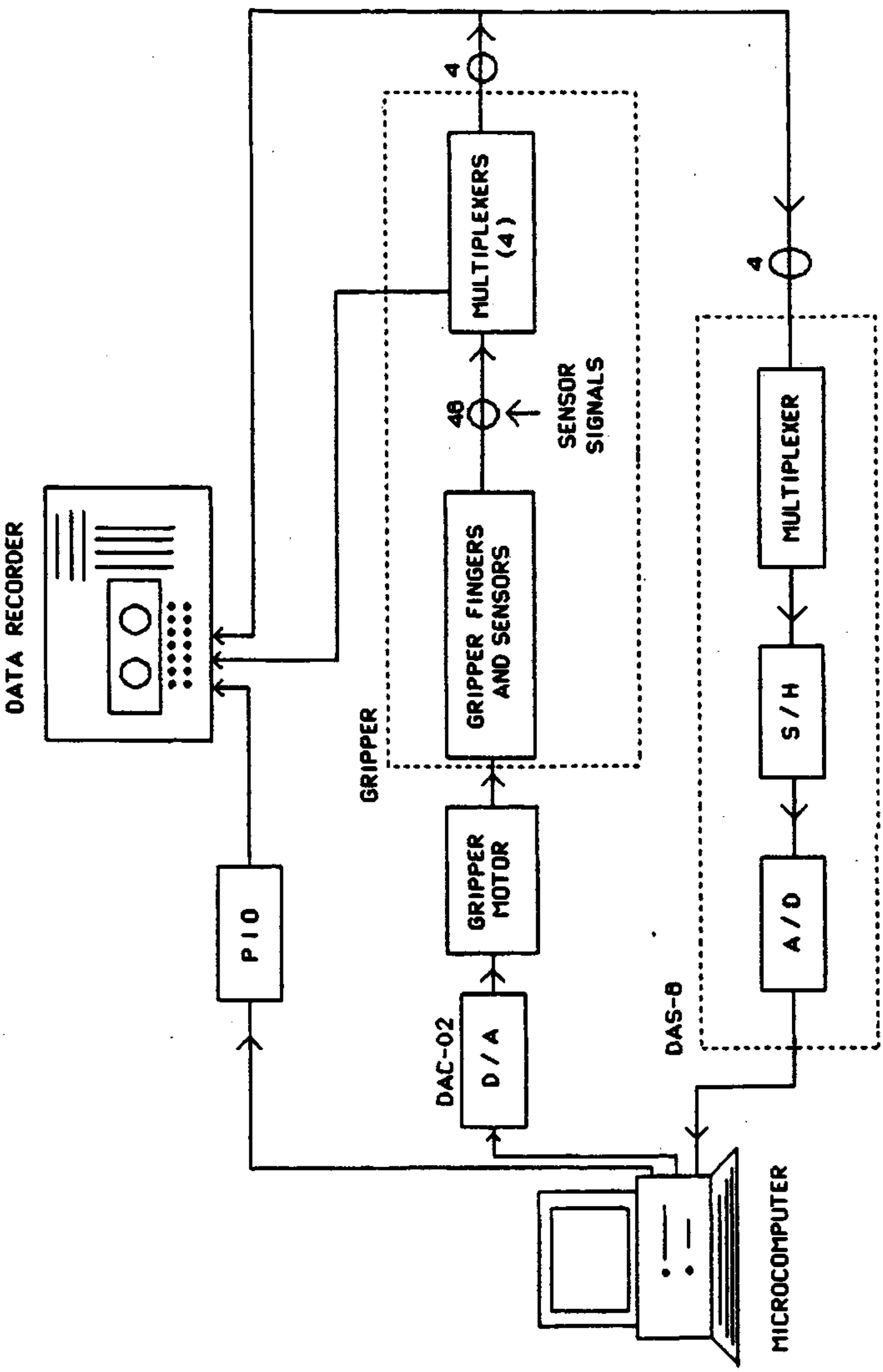


Figure 3.1: Experimental Setup

digital input-output (I/O) interface, a data recorder, and weights. The software consists of programs to perform supervisory control, data acquisition and processing.

3.1. Hardware

3.1.1. Gripper

The parallel jaw gripper and similar mechanisms have been the dominant end-effectors used by research and industrial robots. Owing their success to mechanical simplicity, these grippers have repeatedly proven their usefulness in parts handling and simple assembly operations. For such tasks, there can be no doubt as to their continued effectiveness. As robotic technologies expand into unstructured environments and automated tasks become more complex, it is unlikely that these grippers will be appropriate in all situations. Comprehensive assembly tasks will require refined manipulation capabilities made possible only by grippers that can adapt rapidly to objects of complex geometries and to changes in the position of the grasped object.

The gripper that is used for this project was designed and built at the University of Saskatchewan by Muench [7, 13]. It is a three-fingered gripper with tactile sensing capabilities. Tactile sensing is accomplished with the aid of force sensing resistors which are placed underneath the gripping pads of the gripper. The gripper was designed to be used in a medium industrial environment, such as a metal working shop or small assembly shop, for jobs such as positioning of workpieces into machine tools or component assembly. A sketch of the gripper is shown in Figure 3.2.

3.1.1.1. Tactile Sensing

The tactile sensing technique makes use of force sensitive resistors. The resistors are manufactured by Interlink Electronics, and are sold under the trade name Force Sensing Resistor [16]. A Force Sensing Resistor, or FSR, consists of a sandwich formed from electrically conductive polymer layers. The components are formed by silk-screening an electrically conductive polymer layer onto Mylar sheets [16]. It can be trimmed into virtually any shape and size that is required. There are two basic arrangements of the FSR, Shunt-Mode construction, as shown in Figure 3.3 (a), and Through-Mode construc-

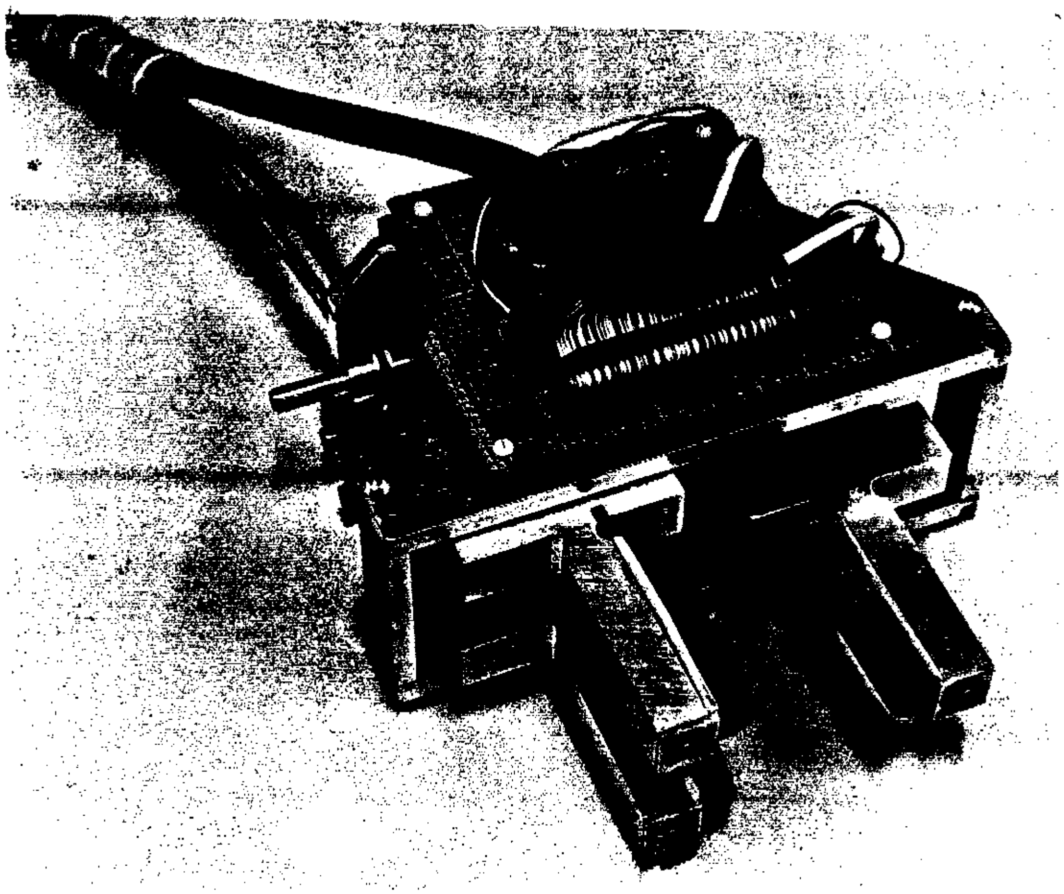
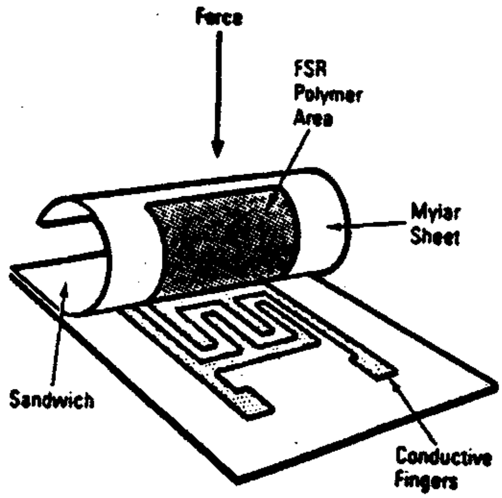


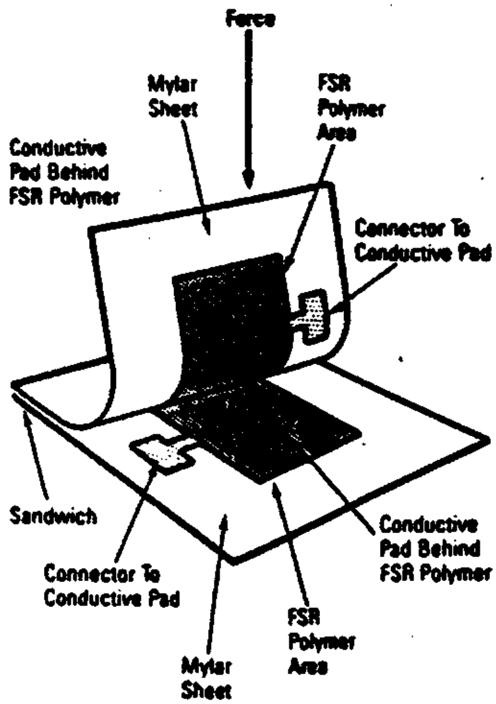
Figure 3.2: Gripper

tion, as shown in Figure 3.3 (b). According to the manufacturer, there are no real advantages of one mode over the other; the choice of one mode for a specific operation depends primarily upon economics and convenience for each particular application.

With the Shunt-mode, the FSR surface is laid in contact with a second surface containing a pair of interlinking conductive fingers. These fingers can be foil overlaid on



(a) Shunt-Mode (from FSR manual [16], Fig. 1)



(b) Through-Mode (from FSR manual [16], Fig. 2)

Figure 3.3: Force Sensing Resistor Modes

mylar, or traces etched on a printed circuit board. When a force is applied perpendicular to the surface area, a shunt circuit is formed between the fingers. The circuit resistance drops as the force is increased.

The Through-mode consists of two opposing FSRs, each of which has been printed over a conductive pad. The two sheets are sandwiched together with the two FSR polymer areas in contact. The FSR elements conduct between each other, and the conductive pads provide connections from which the output resistance can be measured. A force applied perpendicular to the polymer areas lowers the resistance between the two conductive pads.

The style of application that was used for this project was the shunt-mode construction, since this method was found satisfactory in the original gripper design. The mounting substrate for the sensors were produced by etching interlinked patterns onto the surface of a printed circuit board. Each printed circuit board had sixteen 5mm by 5mm patterns, and connections were run from the patterns to solder pads, as shown in Figure 3.4. An FSR was placed on each pattern to form a sensor site.

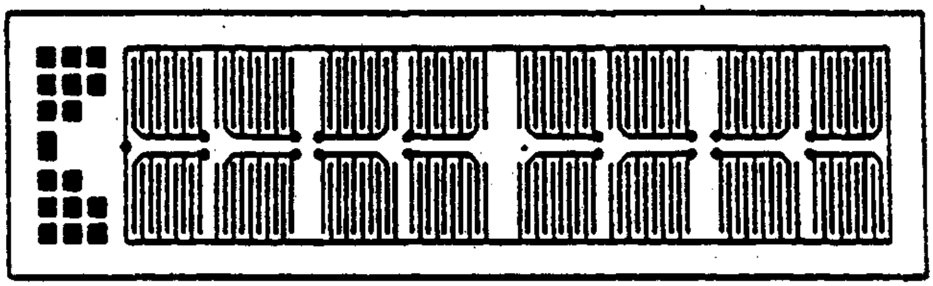
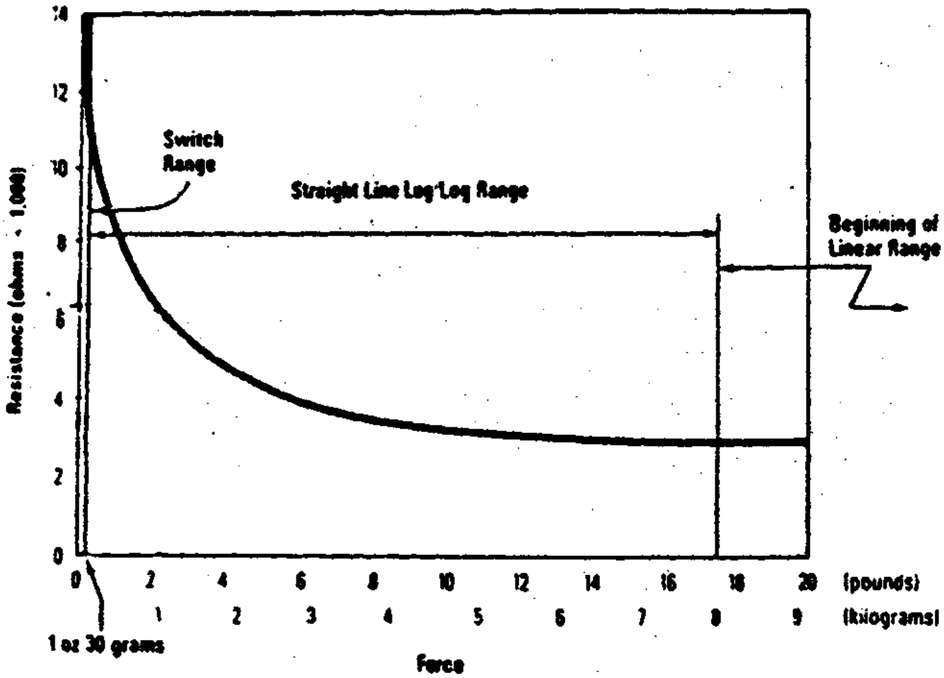


Figure 3.4: Sensor Printed Circuit Board (From Wood [13], Fig. 2)

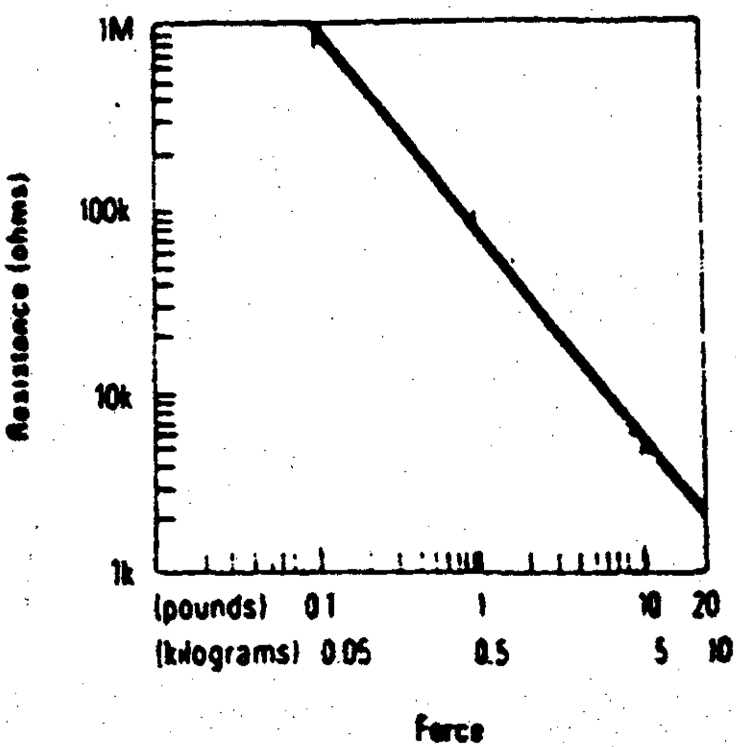
The operating resistance of FSRs can range from ten megohms to a few hundred ohms. There are three useful working ranges: the Switch Range, the Linear Log/Log Range, and the Linear Range. The following characteristics are typical of a sample of about 1 centimeter in diameter [16]. The switched range occurs with applied forces between zero and 30 grams. When no force is applied to the sensor, the resistance is very high (10 to 100) megohms. A slight force (less than 30 grams) causes an abrupt change in resistance of about 100:1. The linear log/log range starts after the switched range. This is a more stable and predictable range, and is the primary area for many applications. The relationship between the applied force and sensor resistance is approximately a Log/Log straight line. This range is from about 30 grams to 8 kilograms. The linear range begins near the end of the linear log/log range. In this range, the change of resistivity with applied force becomes smaller and approximately linear. The maximum useful load can be in excess of 455 kilograms. Figure 3.5 (a) and 3.5 (b) show the Resistance vs. Force relationship.

3.1.1.2. Gripper Pad / Sensor Configuration

As with most tactile sensors, the FSR elements themselves can only sense forces acting normal to its surface, and not forces acting in the shear plane. Measuring slippage on an object usually requires that forces in the shear plane be measured. To obtain the desired measurement, it is necessary to remove the point of contact of the grasped object to a point above the sensors, allowing the shear force at the surface to be translated into a bending moment that is resisted by forces normal to the sensor array [7]. The gripper pad / sensor construction is shown in Figure 3.6. The gripper pad (or pad) is made from polyurethane material. Polyurethane has been used successfully in robotic gripper applications for a number of years, because of its smooth, yet rigid, surface and high coefficient of friction to aid gripping. Also important are the excellent wear characteristics and the fact that it does not damage or scratch the surface of the object being grasped. Each finger has four pads and each pad lies above a sensing array that is made up of four separate sensors. Two layers of flexible materials lie between the gripper and sensors. These materials help transfer the force from the pad to the sensor array by acting as a suspension on which the gripper pad can rest and distribute its load [7]. The first material, called the compliant membrane, is made from neoprene. The second material was added to increase the sensitivity of the gripper. It is made from latex rubber and cut into the size of the sensors. The whole assembly is held together with adhesives.



(a) Resistance vs. Force for an FSR (From FSR manual [16], Fig. 3)



(b) Log/Log Plot of Resistance vs. Force (From FSR manual [16], Fig. 4)

Figure 3.5: Resistance vs. Force Plots For FSR

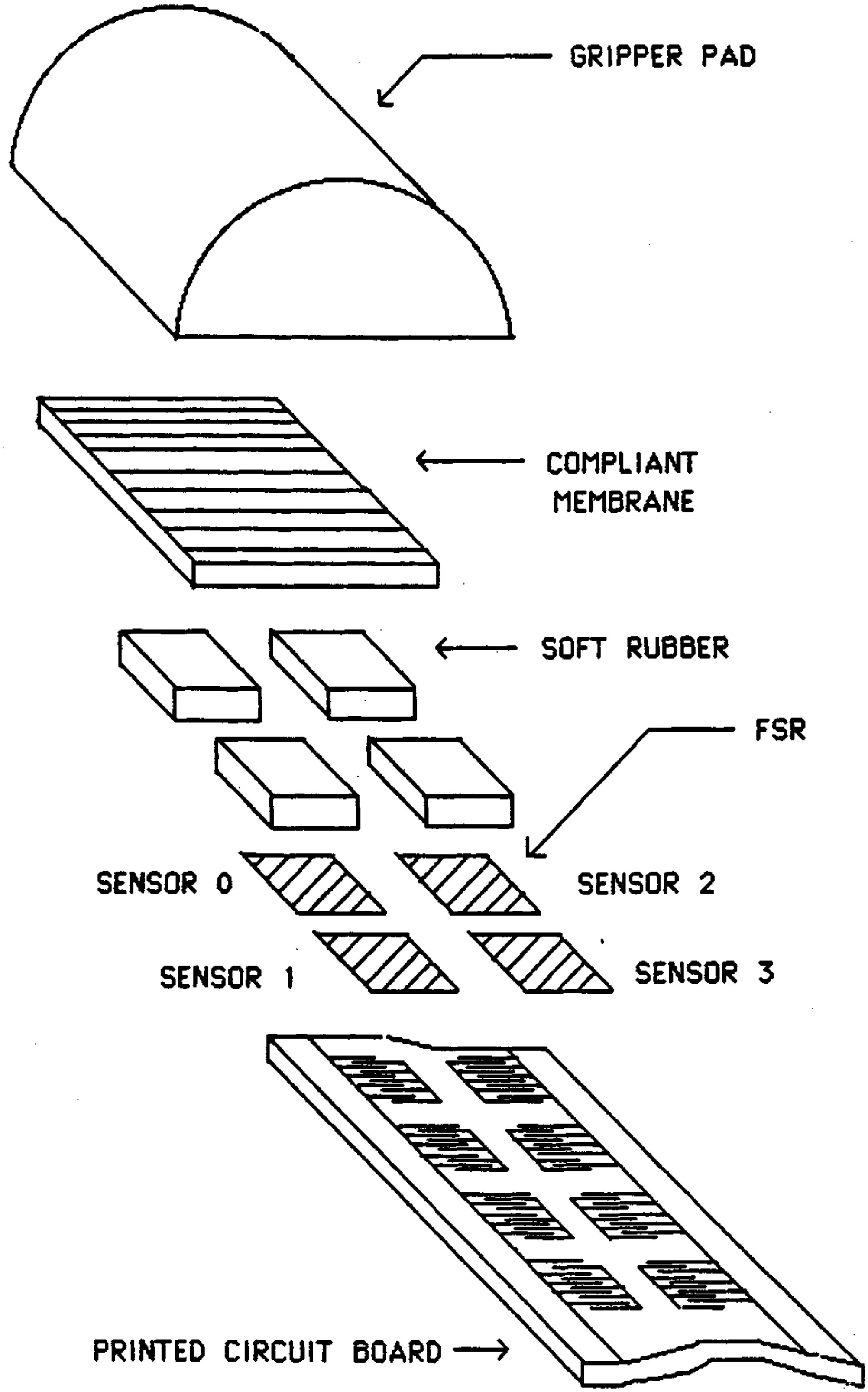


Figure 3.6: Gripper Pad / Sensor configuration

3.1.1.3. Tactile Data Acquisition

Acquisition and conversion of the sensor signal into usable form involves conversion of the output resistance of the FSR to a voltage. This is done by using a constant current source as an input to the sensor as shown in Figure 3.7. The constant current source is switched rapidly from sensor to sensor. By providing a constant current source, the output voltage would be directly proportional to the resistance of the FSR, and inversely proportional to the force being applied at that sensor site.

The constant current source that is used is a voltage-controlled constant current source [7, 18], which is shown in Figure 3.8. The voltage at the noninverting input of the operational amplifier, V_{in}, controls the current flowing through the load resistor, Z_L. The operational amplifier maintains the transistor emitter voltage at V_{in}, and biases the transistor base to the value needed to hold the load current constant. The load current, i_L, is approximately equal to the current through the resistor, R, and is given by the equation below.

$$i_L \approx \frac{V_{cc} - V_{in}}{R}$$

There is a limit on the output voltage. It must not exceed V_{in}, otherwise the transistor would be driven into saturation. Therefore, the load current multiplied by the maximum load resistance should not be greater than V_{in}.

The tactile data acquisition system has four multiplexers. The sensors are multiplexed by connecting all the sensors with the same label to the same mutiplexer. For example, all sensors labelled "sensor 0" are connected to the same multiplexer. The sensors underneath the same pad have the same address on their respective multiplexers, so sensor 0 of pad x would have the same address on its multiplexer as sensor 1 of the same pad or any other sensor of pad x. Therefore, the sensors of a pad are selected simultaneously by selecting the same address on all the multiplexers. Signals from the four sensor elements of a pad are obtained simultaneously using four separate current sources. The constant current source was modified and updated from the original design (Figure 3.8) with additional features to accommodate the multiplexing circuit required to access

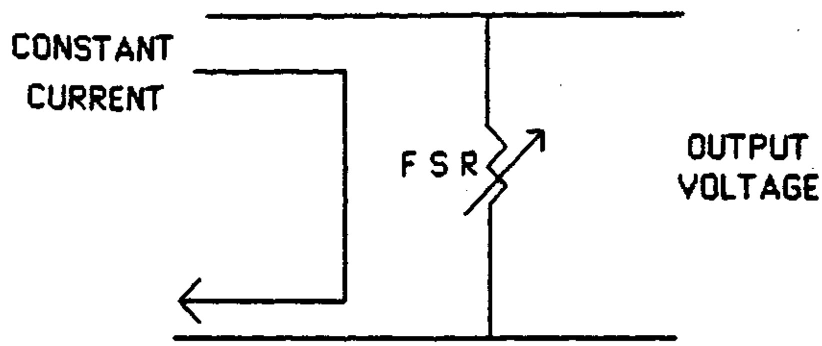


Figure 3.7: Typical Circuit for FSR (From FSR manual [16], Fig. 5)

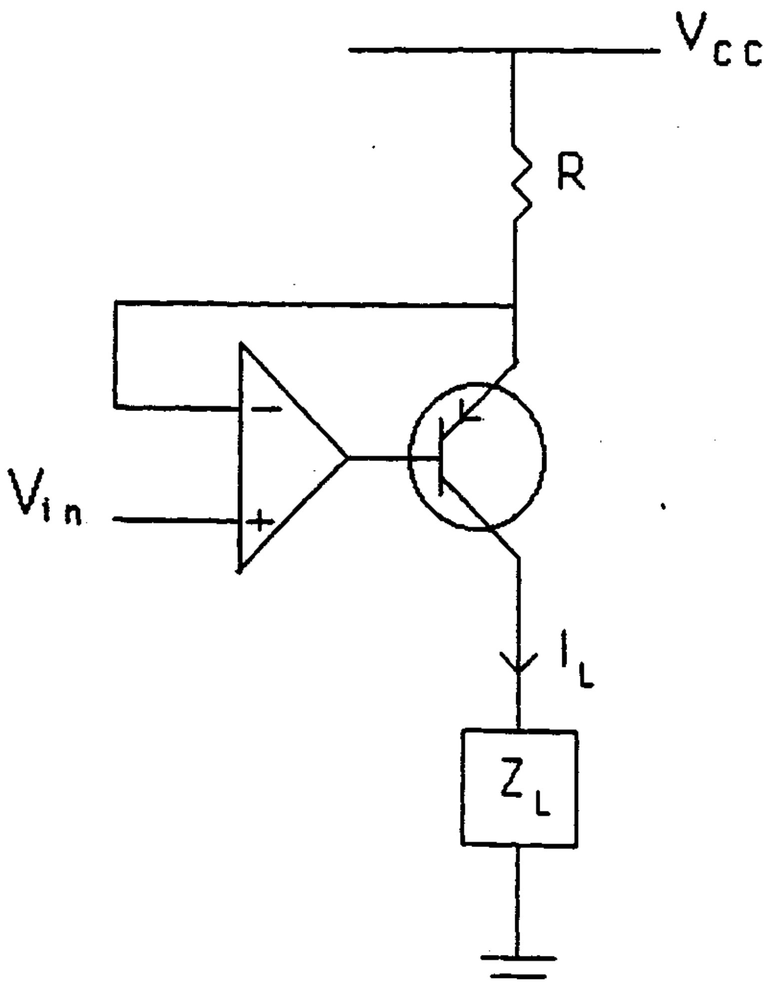


Figure 3.8: Constant Current Source (From Malvino [17], Fig. 17-20, Pg. 531)

the large number of FSR elements, and the signal conditioning circuit. The modified circuit is shown in Figure 3.9. In the modified circuit, V_{in} has been replaced by a voltage

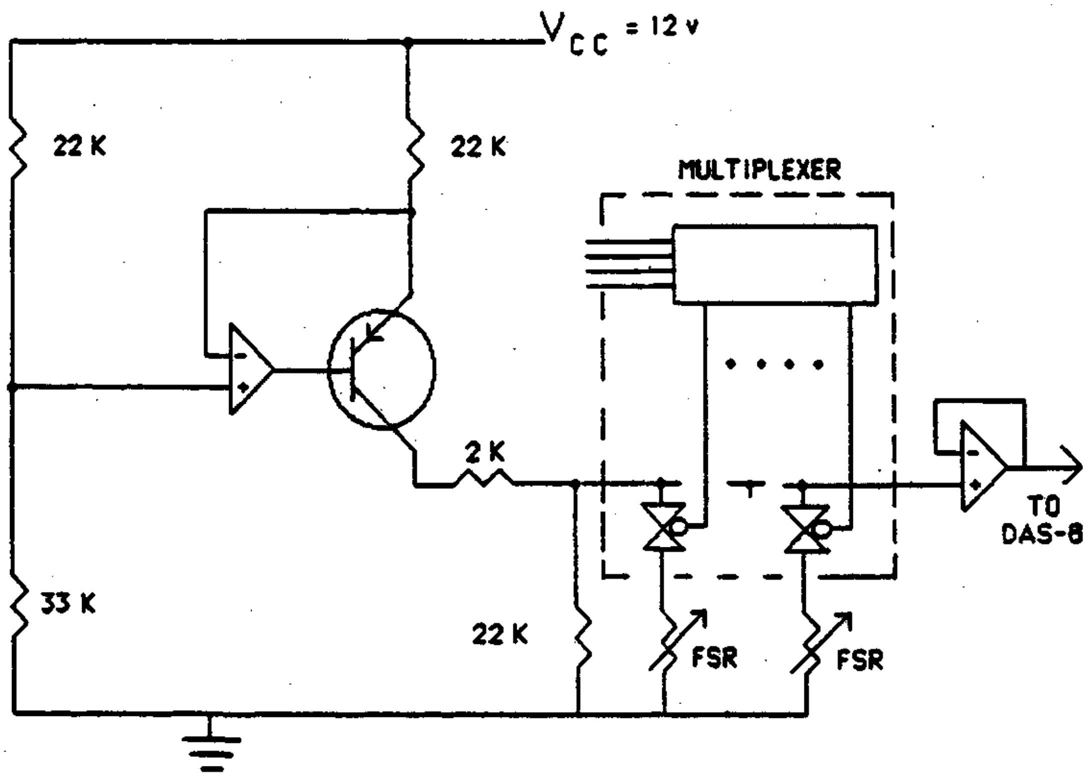


Figure 3.9: Tactile Data Acquisition Circuit (1 of 4)

divider, and a 22 KΩ resistor has been connected in parallel with the FSR elements to provide a path for the current when all the FSR elements are disconnected during switching. Also, when a FSR sensor is connected to the circuit, if there is no force on the sensor, the resistance would be in the megohm range [16]. If there was no 22 KΩ resistor in parallel with the FSR, the transistor would be driven into saturation, and it would no longer act as a constant current source. Hence the 22 KΩ parallel resistor keeps the transistor in the active region at all times. The output voltage from each current source is connected to a signal conditioning unit, which consists of a buffer stage. The output of each buffer stage is connected to one of the channels of the DAS-8.

3.1.1.4. Finger Design

The three fingers of the gripper provide a grasp which is more stable than that of two fingers. An analogy is the comparison of the stability of a three-legged stool with that of a two-legged stool. The finger configuration enables the gripper to grip objects of different sizes and shapes including flat, round, curved or irregular objects. The fingers, which approach each other in a parallel motion, can grip the external or internal surfaces of an object, as shown in Figure 3.10. Gripping is possible on the internal surface because the fingers can reverse their position by passing by each other. This provides a great amount of flexibility in the applications for which the gripper can be used. The fingers of the gripper are driven by the gripper motor.

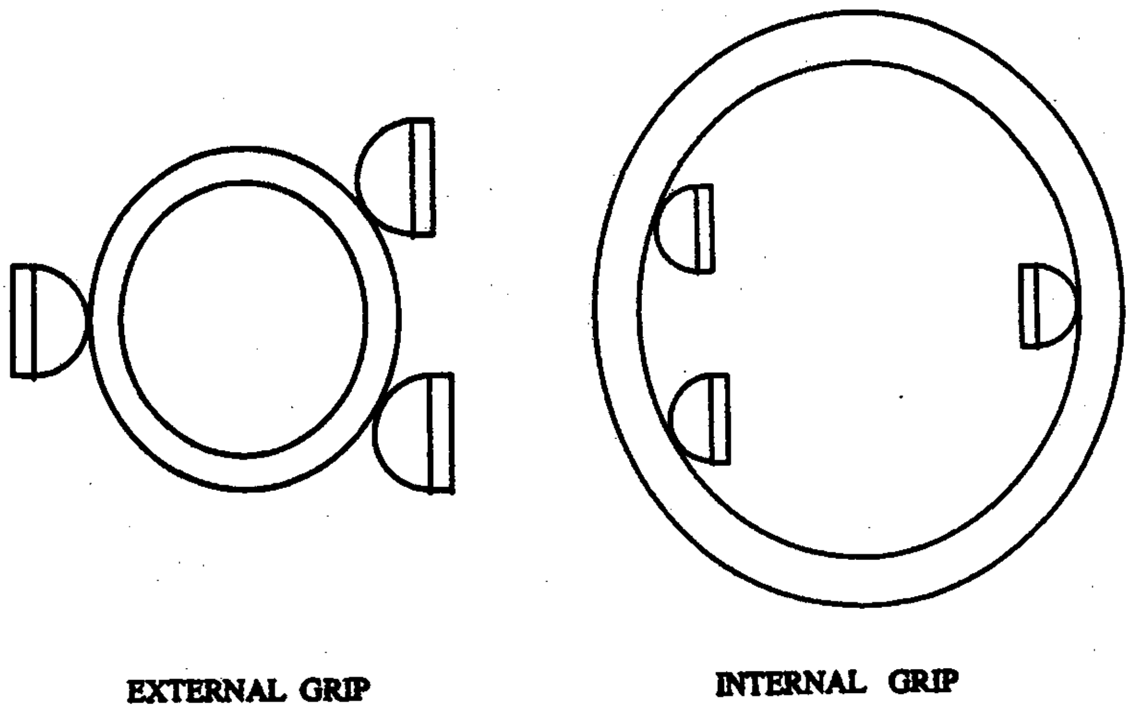


Figure 3.10: External and Internal Gripping

3.1.2. Gripper Motor

The gripper motor is a dc voltage servo motor, which is remotely located in order to minimise the size and the weight of the gripper. It is connected with a high strength flexible steel cable to the finger actuator. Drive actuation of the fingers is accomplished by a worm gear drive which maintains the holding torque supplied by the motor even when the motor power is removed. Until the motor is reversed, the worm mechanism will not release the holding torque from the gripper. As a result, the motor power can be completely removed once the gripper has obtained the desired gripping force or position. More information on the worm drive technique is given in [7].

A precision multi-turn potentiometer is attached to the worm shaft to provide position information. This information is used, when the gripper opens or closes, to find out if the gripper has reached its travel limit.

The drive motor is driven under computer control through a digital-to-analog converter and a bi-directional power amplifier.

3.1.3. Computer Hardware

The computer hardware needed by robot systems is essentially the same as that of other data processing systems. Many of the functions required can be performed by microprocessors and minicomputers used for commercial and industrial applications [19]. Computer programs written to carry out these functions are discussed in the section on software.

An AT&T 6300 WGS (Work Group System) microcomputer was used in the experiment. This microcomputer uses an Intel 8086 microprocessor as its CPU (central processing unit). It is also equipped with an Intel 8087 math coprocessor. The 8086 and 8087 microprocessors have clock speeds of 10 MHz. The 8087 processor, which is designed to perform arithmetic operations efficiently, provides a simple and effective way to enhance the performance of the 8086 processor particularly when an application is computational in nature. As an example of its computing power, the 8087 can multiply two 64-bit real numbers in about 27 microseconds and calculate a square root in about 36 microseconds. If performed by the 8086 through emulation, the same operations would require 2 milliseconds and 20 milliseconds respectively [20]. There are 640 KBytes of RAM on the system board and two disk drives on the system. The microcomputer has interfaces for parallel and serial ports, and A/D and D/A converters [21].

3.1.4. DAS-8

The DAS-8 board has an analog-to-digital converter, eight analog input channels, three digital input lines and four digital output lines, as shown in Figure 3.11. Conversion of analog input voltage to a digital code was done by the analog-to-digital (A/D) converter. It can digitise voltages within the range of -5 to +5 volts. A multiplexer is provided to enable the A/D to select any one of the analog input channels for conversion. The A/D derives its clock cycles from the microcomputer system clock, with conversion times of 35 microseconds for 12-bit conversion and 25 microseconds for 8-bit conversion [22]. A/D conversion is done by successive approximation method, and a sample-and-hold (S/H) device keeps the voltage of the selected channel constant until conversion is completed.

Five analog input channels were used in the experiment. Out of these, four channels were used for the sensor signals and one channel was used to digitise the signal from the gripper position potentiometer.

3.1.5. Digital Output

Digital output was done using the digital output lines of the DAS-8, and a digital I/O interface called PIO12 [23], or PIO for short. The four digital output lines of the DAS-8 were used to select the pads of the robot gripper; the PIO was used because an extra output line was required as a slip detection signal for subsequent processing and display. There are three 8-bit I/O ports in the PIO. One output line of Port A, of the PIO, was used for the slip detection signal.

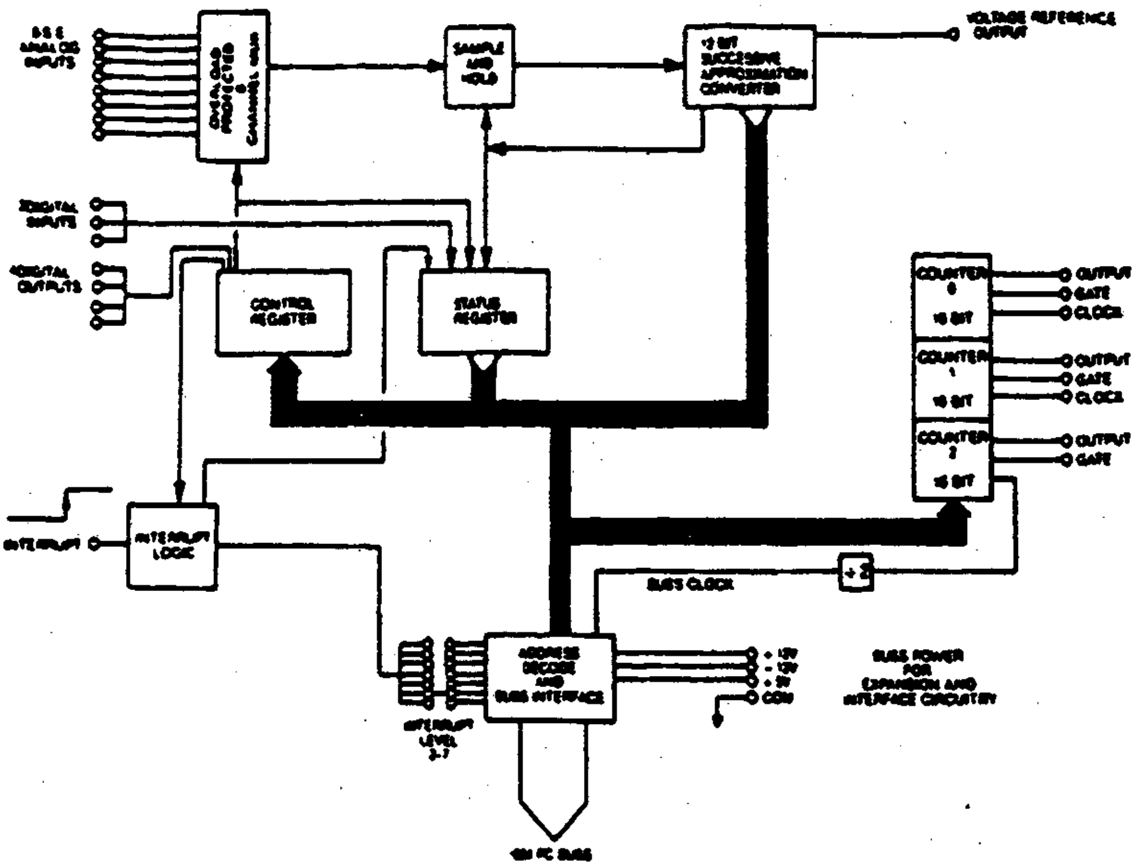


Figure 3.11: Block Diagram for DAS-8 Board (From DAS-8 Manual)

3.1.6. Other Hardware

A D/A converter was required to drive the gripper motor. The converter used was a DAC-02. It has two channels with each one having an output voltage range of -5 to 5 volts [24]. One channel was used for the experiment.

The test object used in the experiment was a hollow cylindrical aluminum tube of length 133mm, outer diameter 28mm, thickness 1mm and weight 45.6g. This tube had a sandblasted surface which provided a random surface texture. Figure 3.12 shows a sketch of the gripper and the test object.

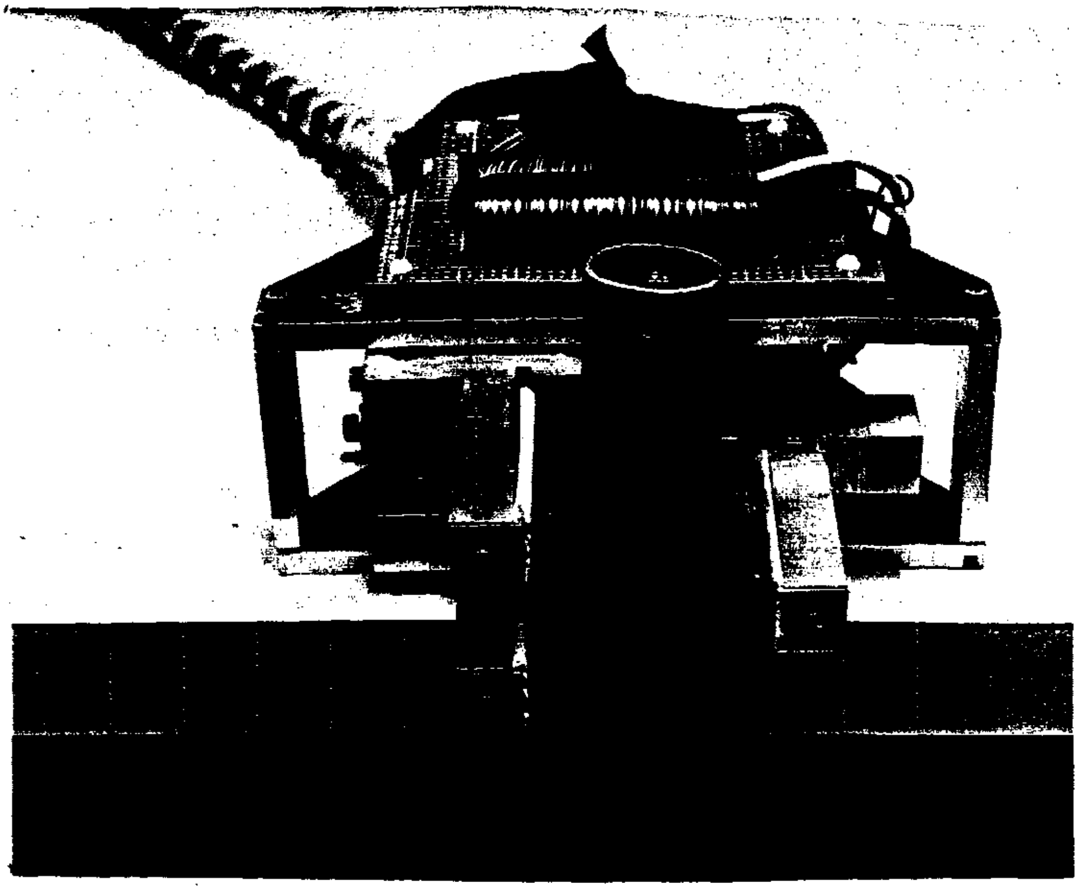


Figure 3.12: Gripper and Test Object

A data recorder was used to record pad selection signals, sensor signals at the A/D input channels, and the slip detection signal from the digital I/O. The data recorder used was a seven channel TEAC XR-310 cassette data recorder. Four of the channels are frequency modulated, while the rest are direct recording channels. The frequency modulated channels were used to record the sensor signals while two of the direct channels were used to record the slip detection signal, and the least significant bit of the pad-select lines. The recorded signals where later digitised and stored in the VAX computer for plotting.

Weights were used to calibrate the sensors.

3.2. Software

The software is used by the computer system. It consists of the system software and the user software. System software is the collecton of programs which are needed in the creation, preparation, and execution of other programs, while user software consists of those programs that are written by the users of the computer system for data acquisition and/or processing [20].

Examples of system software programs include the operating system and software developmental support programs. The operating system of the microcomputer that was used in the experiment was MS-DOS 3.30a, release 1.01. The operating system starts up the computer system every time it is turned on, and provides an interface between the user and the computer system. It does this with the aid of the Resident Monitor which is in the computer memory at all times while the computer is on. The operating system also includes programs called I/O Drivers and File Management Routines. The I/O drivers handle input and output operations to I/O ports, and storage devices such as disk drives. Whenever a user program or other system program needs to use an I/O device, it requests the operating system to use an I/O driver to perform the task. This gives the operating system better control of the computer and alleviates the need to include I/O subroutines within user programs. File management routines are used in conjunction with the I/O drivers for formatting disks to specific layout of sectors and tracks, reading and writing to files to storage devices, manipulating files, and allocating disk file spaces [20].

The user software was developed with the aid of Turbo C, version 2.0. It consists of a text editor for writing programs, a compiler to convert the programs to a form (machine language) which can be used by the microcomputer, and libraries containing subroutines or subprograms, that can be attached to any user program. Turbo C is C language with all the "Turbo" features including a fast and efficient compiler. C is a programming language which can be used for almost any programming task. It encourages the writing of well structured programs and has access to assembly language programming. C language is very portable, so application programs written in C for one system can be easily transferred to another system. It has been implemented on many machines from mainframes to personal computers [25, 26].

Programs were written to perform functions such as data acquisition and processing, and grasp control. Data acquisition involved scanning the sensors periodically, by sending commands to the A/D converter, for information on the state of the sensors. Data processing involved using the digitised data in calculations to determine changes in the forces applied to the sensors. The results of the data acquisiton and processing were used to calibrate the sensors, and in the grasp control algorithm to determine appropriate responses that the gripper should make to changes in its environment. The next chapter describes the sensor calibration experiment. Details of the computer code are included in Appendices A and B.

4. SENSOR CALIBRATION

The sensor calibration experiment was carried out to study the effect on the sensors, of forces applied on the pads. A computer program, named CLB.C, was written to perform sensor calibration. The sensors were calibrated by suspending weights above the gripper pads, and using the sensor calibration program to sample the sensors underneath the respective pads.

4.1. Sensor Calibration Program

In this program, the user has the option of doing either 8-bit short cycle or 12-bit conversion. The user also selects the pad to be calibrated. After the pad has been selected, the four sensors of the pad are sampled sequentially, and the results are displayed on the screen. The user can either resample the four sensors, or sample one of the sensors of the selected pad, or change the initial parameters, for example, select another pad. If the choice is made to sample individual sensors, the user is asked to select one of the four sensors. The selected sensor is sampled a number of times, and the arithmetic mean is calculated. The results, which include the frequency of occurrence of the samples, and arithmetic mean, are displayed on the screen. The flow chart of the calibration program is shown in Figure 4.1, while the program is given in Appendix A.

4.2. Calibration Experiment

The sensor calibration experiment began with the selection of a pad. A hanger was placed on the pad and weights were suspended from the hanger as shown in Figure 4.2. Each sensor under the selected pad was sampled separately and the results were recorded. The weights on the pads were increased and the sensors were sampled after each increase. The weights were then decreased and the sensors were sampled again. Hysteresis in the readings was accounted for by averaging the results. The experiment was carried

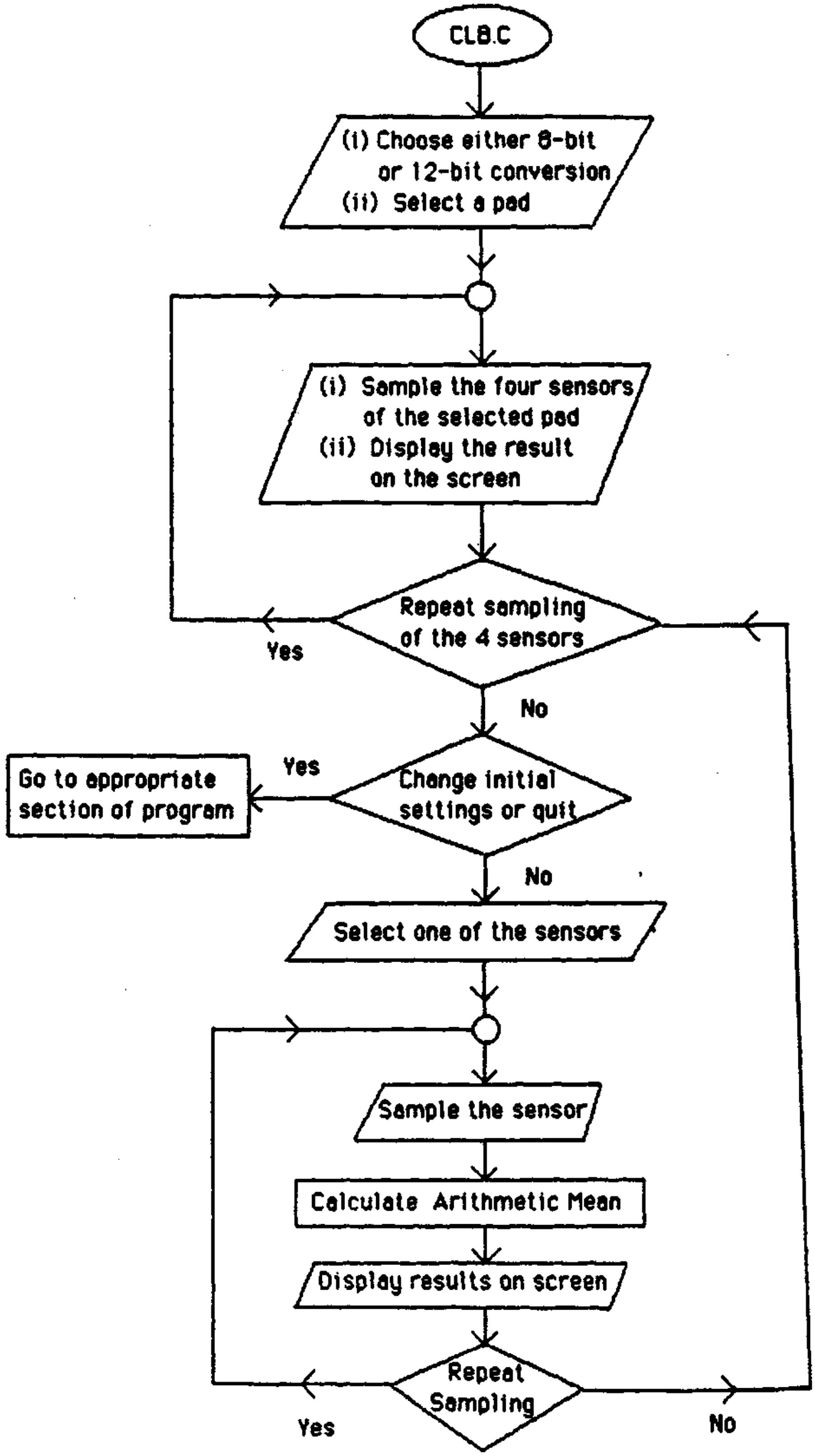


Figure 4.1: Flow Chart for Calibration Program

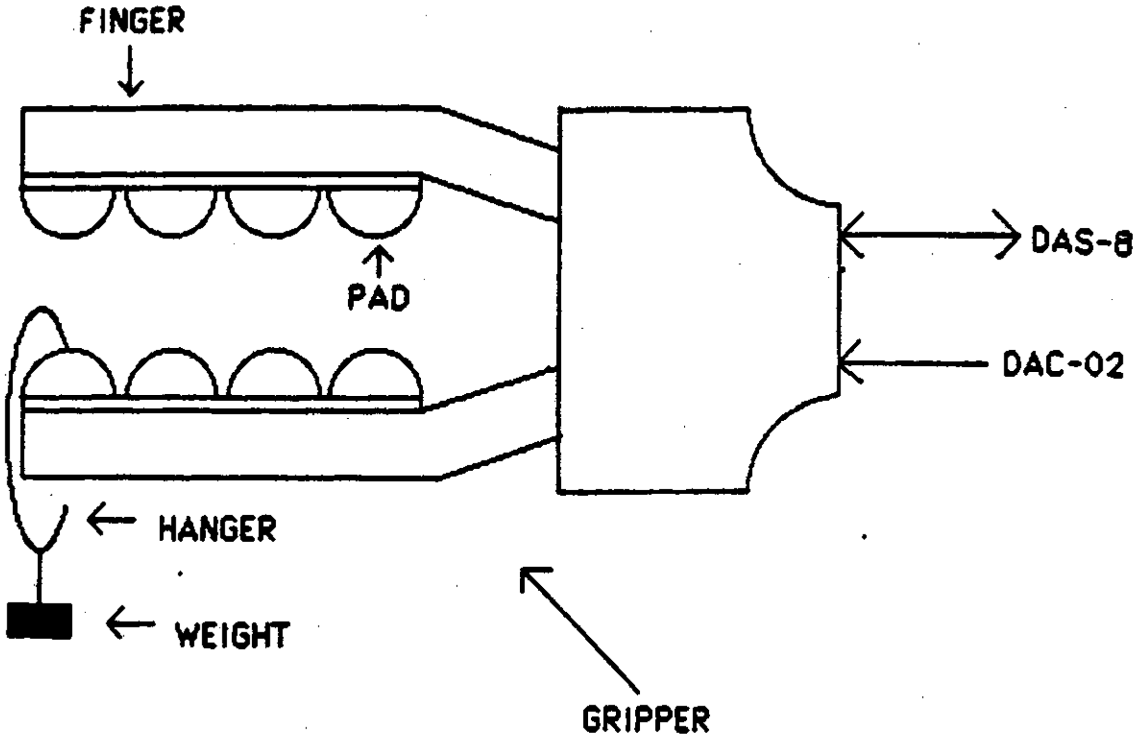


Figure 4.2: Gripper and Weight

out with the hanger placed at three different points on the pad. The points, shown in Figure 4.3 were named "FRONT OF PAD", "CENTER OF PAD", and "BACK OF PAD". The "FRONT OF PAD" point is 3mm from the front of the pad, the "CENTER OF PAD" point is 6mm from the front (and also 6mm from the back) of the pad and the "BACK OF PAD" point is 3mm from the back of the pad.

4.3. Calibration Results

The results of the calibration of pads 1 and 9 are tabulated in Tables 4.1 and 4.2, respectively, and graphically presented in Figures 4.4 and 4.5, respectively. Each figure has four graphs representing the sensors underneath a pad. The horizontal axis on each of the graphs represents the force due to the weights that were suspended from the hanger, while the vertical axis represents the measurement by the A/D of the output voltage from the respective sensors. The horizontal axis annotation for all the graphs are the

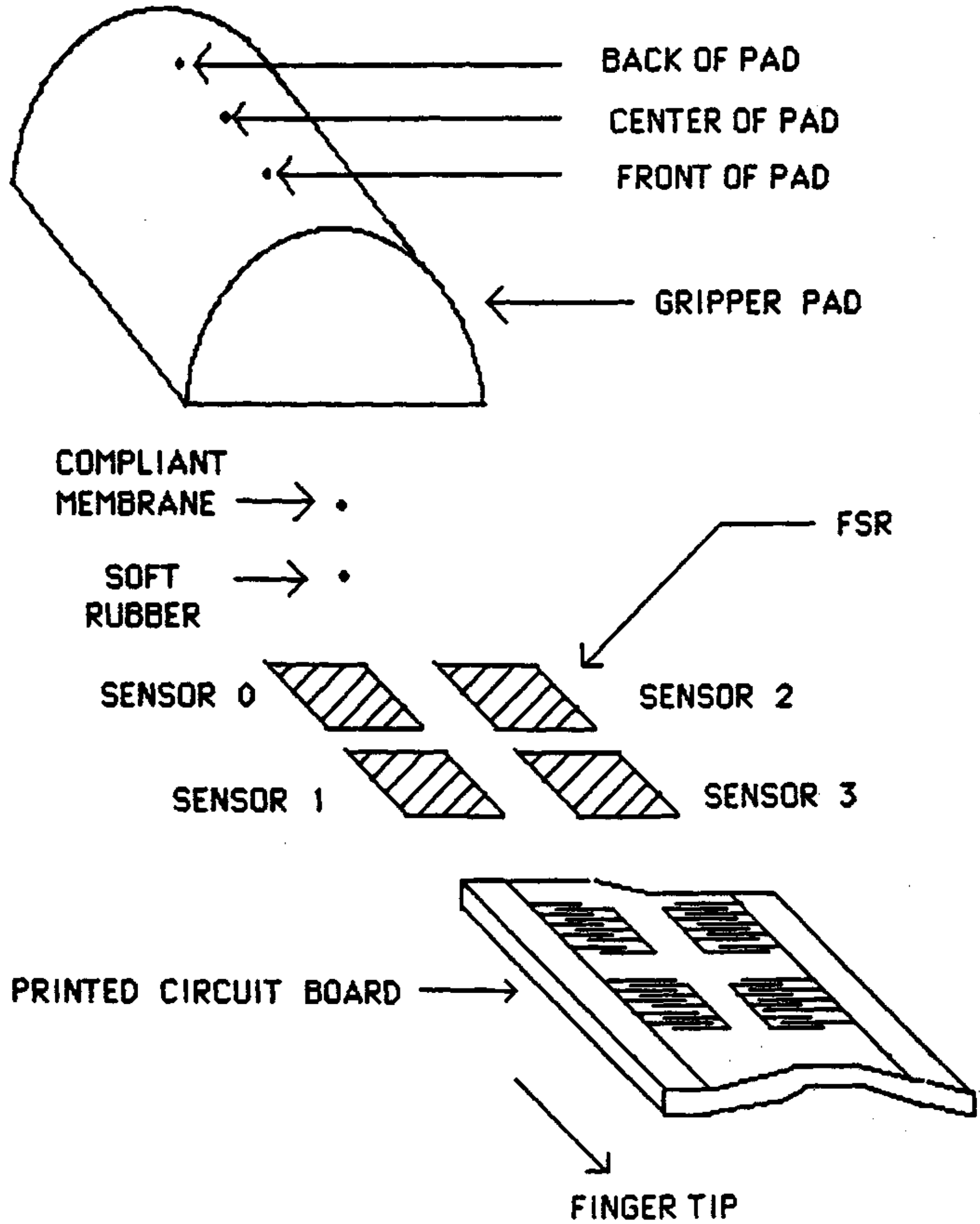


Figure 4.3: Positions of Applied Forces

same, so only the horizontal axis for sensor 3 has been labelled. Each graph has results for sensor output voltages due to forces applied at the front, center and back of the pad.

4.3.1. Calibration Results for Pad 1

The results for pad 1 show that sensors 0 and 2 responded to forces applied at the back of the pad. However, when the weights were moved to the front of the pad, sensors 0 and 2 gave no response other than no-load voltage (4.8 volts). The results also show that sensors 1 and 3 were more responsive to forces applied at the front of pad 1 than to forces applied at any other point on the pad. All the sensors of pad 1 showed some response to forces applied at the center of the pad, with the largest voltage change occurring on sensor 2.

4.3.2. Calibration Results for Pad 9

The results for pad 9 are quite similar to the results from pad 1; with sensors 0 and 2 showing relatively large voltage changes when forces were applied at the back of the pad, and sensors 1 and 3 showing similar changes for forces applied at the front of the pad. However, when the weights were suspended from the center of the pad, only sensor 3 showed a remarkable voltage change. Sensor 3 also responded to forces applied at the back of the pad.

4.4. Analysis Of The Calibration Results

When no force is applied to a sensor, the resistance of the sensor is in the megohm range [16]. From the tactile data acquisition circuit shown in Figure 3.9, all the load current would flow in the 22 KΩ parallel resistor, resulting in an output voltage of 4.8 volts. As the force on the sensor increases, the resistance of the sensor decreases, leading to a decrease in the output voltage from that sensor.

Since sensors 0 and 2 lie underneath the back of the pads, it is expected that these sensors should respond more to forces applied at the back of their respective pads than to forces applied at any other point on their pads. Similarly, sensors 1 and 3, which lie at the front of the pads should show the highest response when forces are applied at the front of their respective pads.

SENSOR 0

FRONT OF PAD

Force (g)	Output (v)
19	4.80
109	4.80
209	4.80
309	4.80
409	4.80
509	4.80
609	4.80
709	4.80

CENTER OF PAD

Force (g)	Output (v)
19	4.80
209	4.80
309	4.73
409	4.71
509	4.69
609	4.67
709	4.65

BACK OF PAD

Force (g)	Output (v)
19	4.80
109	4.73
159	4.67
209	4.51
259	4.36
359	4.30
459	4.25
559	4.04
709	3.85

SENSOR 1

FRONT OF PAD

Force (g)	Output (v)
19	4.80
159	4.80
259	4.73
309	4.68
409	4.61
509	4.60
609	4.57
709	4.53

CENTER OF PAD

Force (g)	Output (v)
19	4.80
209	4.80
309	4.79
409	4.77
509	4.73
609	4.71
709	4.69

BACK OF PAD

Force (g)	Output (v)
19	4.80
109	4.80
209	4.80
309	4.80
409	4.80
509	4.80
609	4.80
709	4.80

Table 4.1: Results for Pad 1

SENSOR 2

FRONT OF PAD

Force (g)	Output (v)
19	4.80
109	4.80
209	4.80
309	4.80
409	4.80
509	4.80
609	4.80
709	4.80

CENTER OF PAD

Force (g)	Output (v)
19	4.80
209	4.67
309	4.57
409	4.43
509	4.28
609	4.12
709	4.06

BACK OF PAD

Force (g)	Output (v)
19	4.80
59	4.77
109	4.65
159	4.45
209	4.26
259	4.02
359	3.87
459	3.65
559	3.40
709	3.18

SENSOR 3

FRONT OF PAD

Force (g)	Output (v)
19	4.80
109	4.73
159	4.61
259	4.49
309	4.32
409	4.28
509	4.06
609	3.90
709	3.87

CENTER OF PAD

Force (g)	Output (v)
19	4.80
209	4.77
309	4.77
409	4.69
509	4.61
609	4.61
709	4.61

BACK OF PAD

Force (g)	Output (v)
19	4.80
109	4.80
209	4.80
309	4.80
409	4.80
509	4.80
609	4.80
709	4.80

Table 4.1 contd.

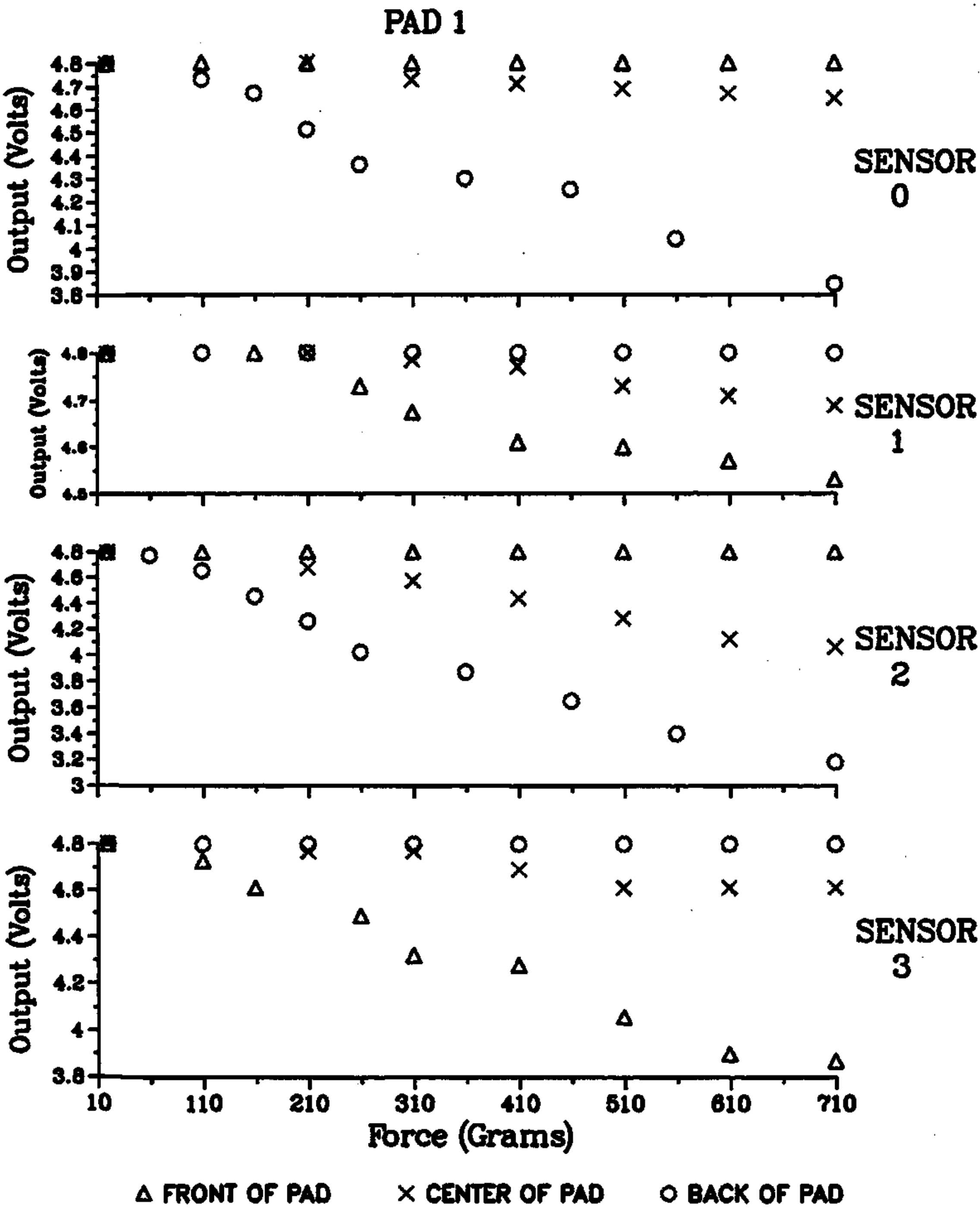


Figure 4.4: Calibration Graphs for Pad 1

SENSOR 0

FRONT OF PAD

Force (g)	Output (v)
19	4.80
109	4.80
209	4.80
309	4.80
409	4.80
509	4.80
609	4.80
709	4.80

CENTER OF PAD

Force (g)	Output (v)
19	4.80
209	4.80
309	4.80
409	4.80
509	4.80
609	4.77
709	4.77

BACK OF PAD

Force (g)	Output (v)
19	4.80
109	4.80
209	4.77
309	4.73
359	4.69
409	4.57
509	4.49
609	4.41
709	4.26

SENSOR 1

FRONT OF PAD

Force (g)	Output (v)
19	4.80
159	4.73
259	4.67
359	4.57
509	4.49
709	4.45

CENTER OF PAD

Force (g)	Output (v)
19	4.80
109	4.80
209	4.80
309	4.80
409	4.74
509	4.74
609	4.74
709	4.73

BACK OF PAD

Force (g)	Output (v)
19	4.80
109	4.80
209	4.80
309	4.80
409	4.80
509	4.80
609	4.80
709	4.80

Table 4.2: Results for Pad 9

SENSOR 2

FRONT OF PAD

Force (g)	Output (v)
19	4.80
109	4.80
209	4.80
309	4.80
409	4.80
509	4.80
609	4.80
709	4.80

CENTER OF PAD

Force (g)	Output (v)
19	4.80
109	4.80
209	4.77
309	4.69
409	4.65
509	4.61
609	4.57
709	4.53

BACK OF PAD

Force (g)	Output (v)
19	4.80
109	4.77
159	4.53
209	4.45
309	4.30
359	4.06
409	3.67
509	3.44
609	3.16
709	2.77

SENSOR 3

FRONT OF PAD

Force (g)	Output (v)
19	4.80
59	4.69
109	4.53
159	4.38
259	4.10
359	3.94
509	3.79
709	3.67

CENTER OF PAD

Force (g)	Output (v)
19	4.80
109	4.65
209	4.61
309	4.53
409	4.34
509	4.26
609	4.20
709	4.14

BACK OF PAD

Force (g)	Output (v)
19	4.80
109	4.80
209	4.80
309	4.80
359	4.69
409	4.65
509	4.65
709	4.53

Table 4.2 contd.

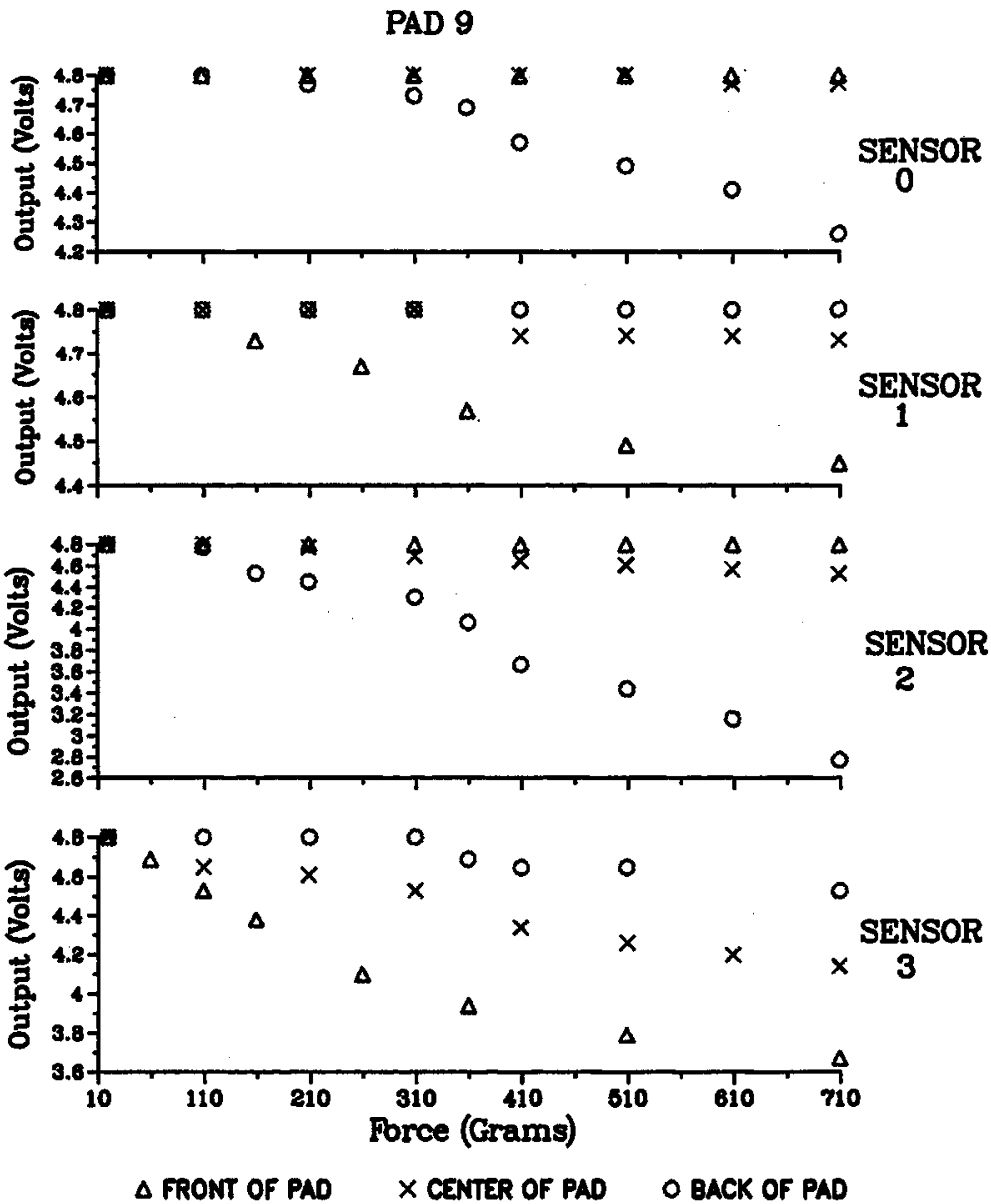


Figure 4.5: Calibration Graphs for Pad 9

Variations in the magnitude of the voltage changes of the sensors can be attributed to slight differences in sensor construction and assembly. The calibration results were applied in the grasp control experiment, which is described in the next chapter.

5. GRASP CONTROL

Information from the calibration experiment was used, in the grasp control experiment, to control the gripper during grasping. The calibration results show that the no-load output voltage of each sensor is 4.8 volts. When an applied force causes the sensor resistance to decrease, the output voltage of the sensor also decreases. Therefore, when the gripper makes contact with an object, a force is applied on at least one of the sensors which causes the resistance and hence the voltage of the sensor to decrease. The decrease in voltage is then used to indicate that the gripper has made contact with the object. This contact information was used, as the gripper closed on the object, to determine if the object had been grasped. The gripper stopped when it was detected that the object was grasped.

The calibration results also show that when a force is applied at different points of a pad, there are different values of output voltage from the affected sensors. When an object, which has been grasped by the gripper, starts to slip, it changes its position on the gripper. This change would be registered as a change in the voltage of any of the sensors. Therefore a change in the voltage of any sensor, with the gripper fingers held fixed, can be used as an indication of slippage. This slip information was used to determine if the grasped object had started to slip. When the object started to slip, the gripper increased its grip. The grasp control experiment was carried out, using different numbers of gripper fingers, and for different modes of slip (translational and rotational).

The setup for this experiment is as shown in Figure 3.1. Communication between the gripper and the microcomputer is bi-directional. In one direction, the microcomputer uses the digital output lines of the DAS-8 to select a pad. In the other direction, the gripper presents the microcomputer, via the DAS-8 analog input channels, the output voltages from the sensors of the selected pad. A computer program, named PROJ.C was written, in C language, to perform grasp control.

5.1. Grasp Control Program

The program utilises the results of the calibration experiment for contact and slip detection. In this program, the user can choose to write the experimental results to the screen and/or a file, select the pads to be sampled, and open or close the fingers of the gripper. The experiment starts when the user chooses to close the gripper. Program control is then transferred to a subroutine named CLOSE_GRIPPER. A positive voltage is applied, through the DAC-02, to the gripper motor to enable it to close in on the test object. While the motor is on, the sensors underneath the selected pads are sampled sequentially. When the A/D performs a conversion, the result is stored as an integer in an A/D register. Each integer value represents a voltage, given by the equation

$$\text{voltage} = ((\text{A/D reading}) * 10 / 256) - 5 .$$

So, an A/D reading of 146 translates to an analog voltage of approximately 0.703 volt. In this program, the A/D readings are used in their integer forms, without transforming them back to the analog voltage values. Results from the calibration experiments show that a sensor reading below no-load voltage is an indication that there is a force acting on the sensor. In this program, as the sensors are sampled, a sensor reading that is less than no-load voltage indicates that the gripper has made contact with the object.

In addition to sampling the sensors, the microcomputer checks to see if the user wants to interrupt the program. It also samples the potentiometer attached to the gripper to determine if the gripper has exceeded its limit. There are three ways of stopping the motor. One way is through a user interrupt. The user interrupts the program by pressing either ctrl-c or ctrl-break. User interrupt is recognised when the microcomputer performs input/ouput operations such as writing to screen, or when the microcomputer checks the status of its peripherals, such as the disk drive. In this program, the microcomputer is made to periodically check the status of the disk drive. When a user interrupt is recognised, program control is passed on to a subroutine named C_BREAK, which stops the motor. This subroutine also allows the user to change the parameters that were set at the beginning of the program, or to exit the program. The motor is also stopped when the gripper approaches its travel limit. When this happens, program control is passed on to subroutine C_BREAK. The third way in which the motor is stopped is when a sensor indicates that the gripper has made contact with an object. The motor is turned off, and control is returned to the main program for the next stage of the experiment, which is slip detection.

The calibration experiment showed that a change in the position of the applied force on a pad results in a change in the output voltage of the sensors of that pad. The grasp control program utilises this information to check for slippage of the grasped object by sampling the sensors repeatedly until a change of output voltage is detected from one of the sensors. The sensors of the selected pads are sampled and the results are stored in part of a ring buffer. The sensors are sampled again and the results are stored in another part of the ring buffer. The data in the ring buffer is processed by subtracting the previous sample value from the latest value, for each sensor. Slip is detected when a sensor value shows a change of magnitude greater than the preset threshold. The threshold value used in this experiment is 3. This means that when subtracting the two samples of a sensor, if the subtraction yields a result that is greater than 3 (0.117 volt), or less than -3 (-0.117 volt), then the object has started to slip. If slip is not detected, the oldest set of sensor readings are discarded, the sensors are sampled again, and the new samples are compared with the results of the previous sampling interval. This process of sampling, checking for slip, and discarding old samples continues until either slip is detected, or the user interrupts the program.

When slip is detected, program control is transferred to a subroutine named CONTROL_ACTION. In this subroutine, the microcomputer puts out a high signal on port A of the PIO, and turns the motor on, for an arbitrary time of 50 milliseconds, to enable the gripper to increase its grip on the test object. If previously requested by the user, the result indicating the sensor that slip was detected is written to file and/or screen. The motor is then turned off, the signal on Port A goes low, and control is returned to the main program. The microcomputer then discards all samples in the ring buffer, and proceeds to resample the sensors.

The microcomputer is able to recognise user interrupts because it also checks the status of the disk drive in addition to sampling of the sensors. An interrupt request from the user causes program control to be transferred to subroutine C_BREAK. This subroutine allows the user to exit the program, or modify the parameters that were set at the beginning of the program. These parameters include the number of pads to be sampled, the write operations of the program, and openning/closing of the gripper. If the user chooses to modify the initial parameters, control is transferred to the appropriate part of the main program. If the user does not choose to do any of these options, the contents of ring buffers are emptied and sampling is repeated.

Opening the gripper fingers transfers control to subroutine OPEN_GRIPPER. In this subroutine, the microcomputer applies a negative voltage to the gripper and the gripper potentiometer is sampled. When the gripper approaches its limit, the motor is stopped and program control is returned to subroutine C_BREAK.

The only way that the user can gracefully exit the program is when the program is interrupted through ctrl-c or ctrl-break. This transfers program control to subroutine C_BREAK. The user then chooses the exit option that is presented in this subroutine to quit the program. The flow chart for the program is shown in Figure 5.1, while the program code is in Appendix B.

5.2. Slip Modes For The Test Object

Two slip modes were identified for the test object, translational slip and rotational slip. Figure 5.2, shows the gripper, the test object and the test object axis. Translational slip occurs when the test object moves in a direction which is parallel to the test object axis, while Rotational slip motion is as a result of tilting of the test object axis.

5.3. Grasp Control During Translational Slip

The grasp control experiment was carried out as the test object was first made to slip in the translational direction. Three tests were carried out for translational slip. In the first test, the pads on the three gripper fingers were sampled; in the second test, the pads on two fingers were sampled; and in the third test, the pads on one finger were sampled.

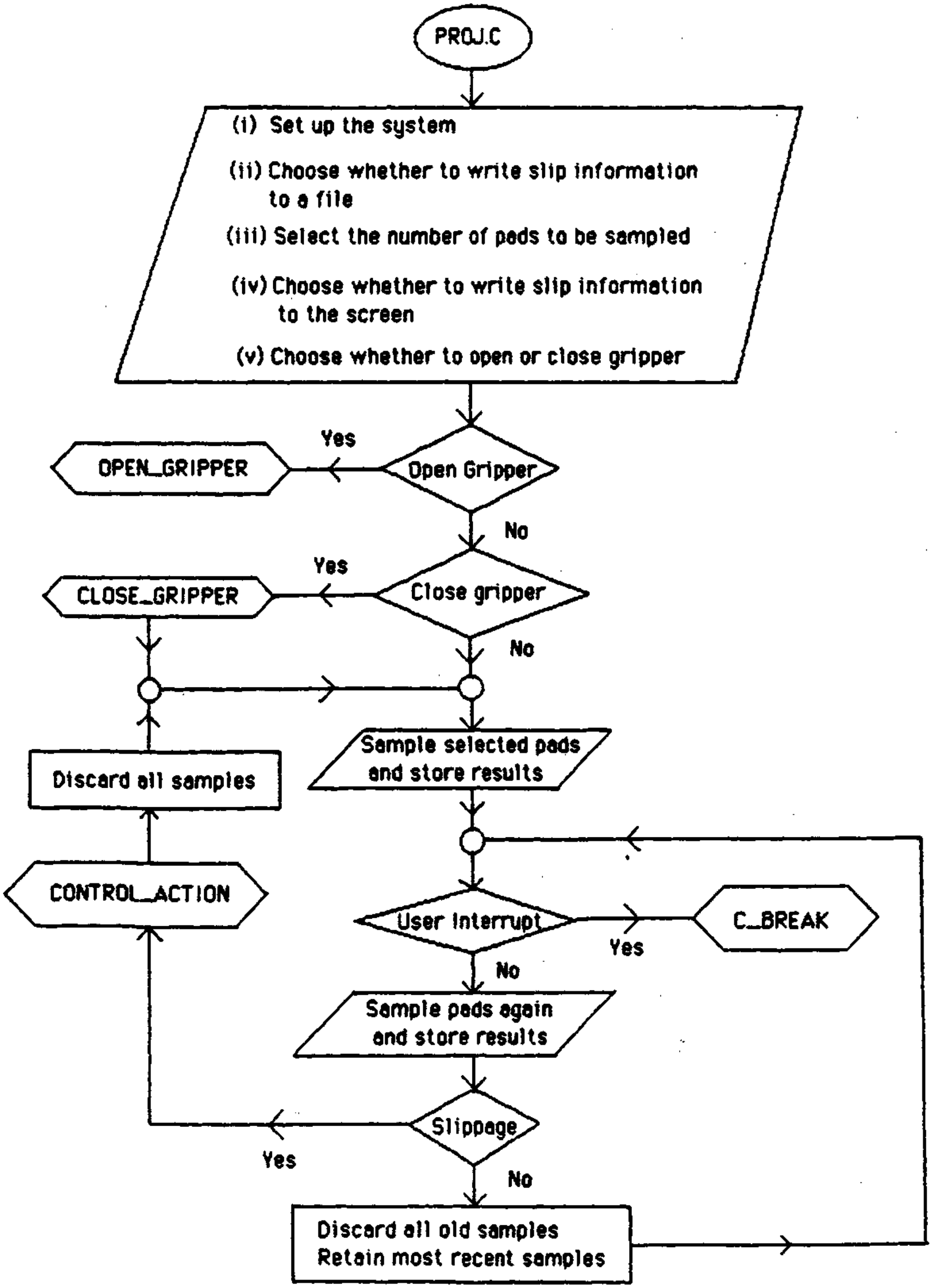
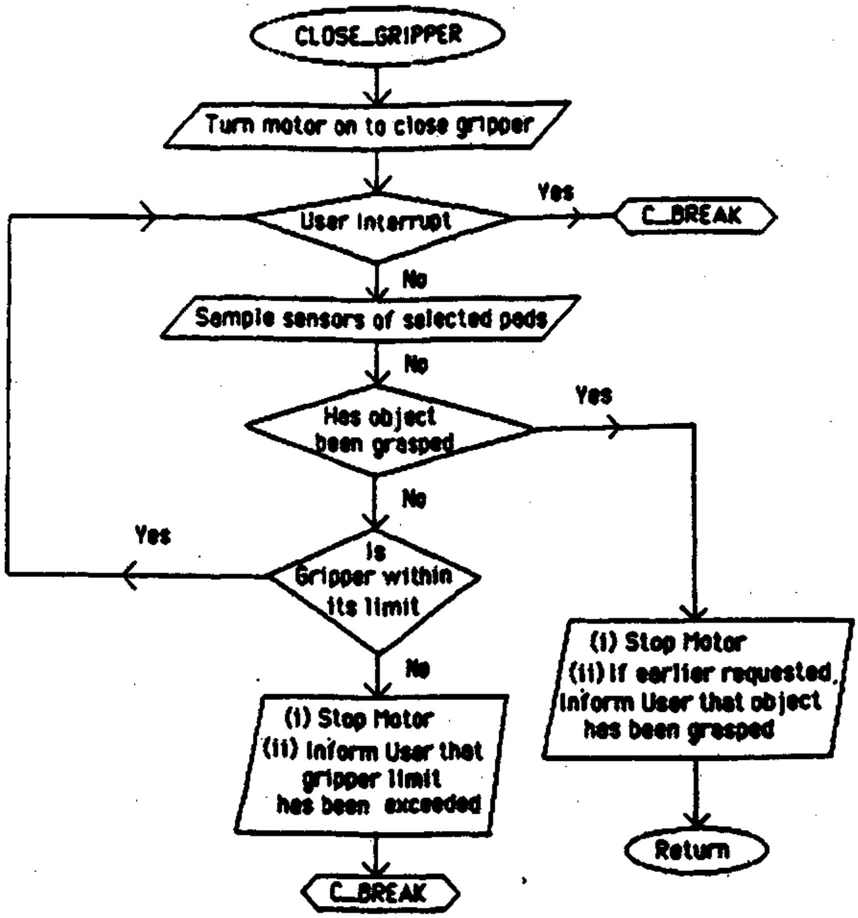


Figure 5.1: Flow Chart for Grasp Control Program



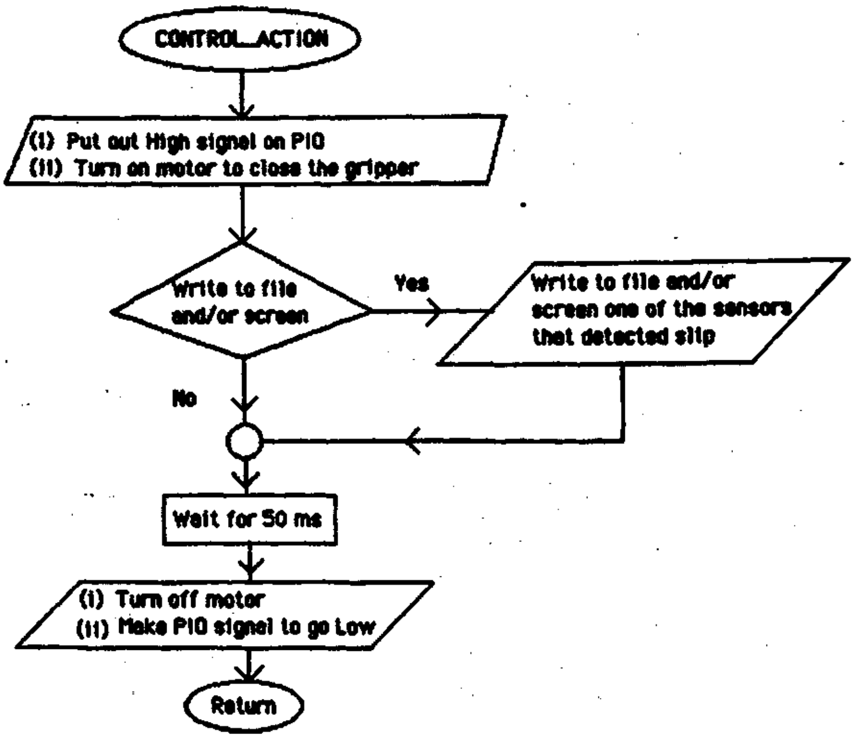


Figure 5.1 Contd.

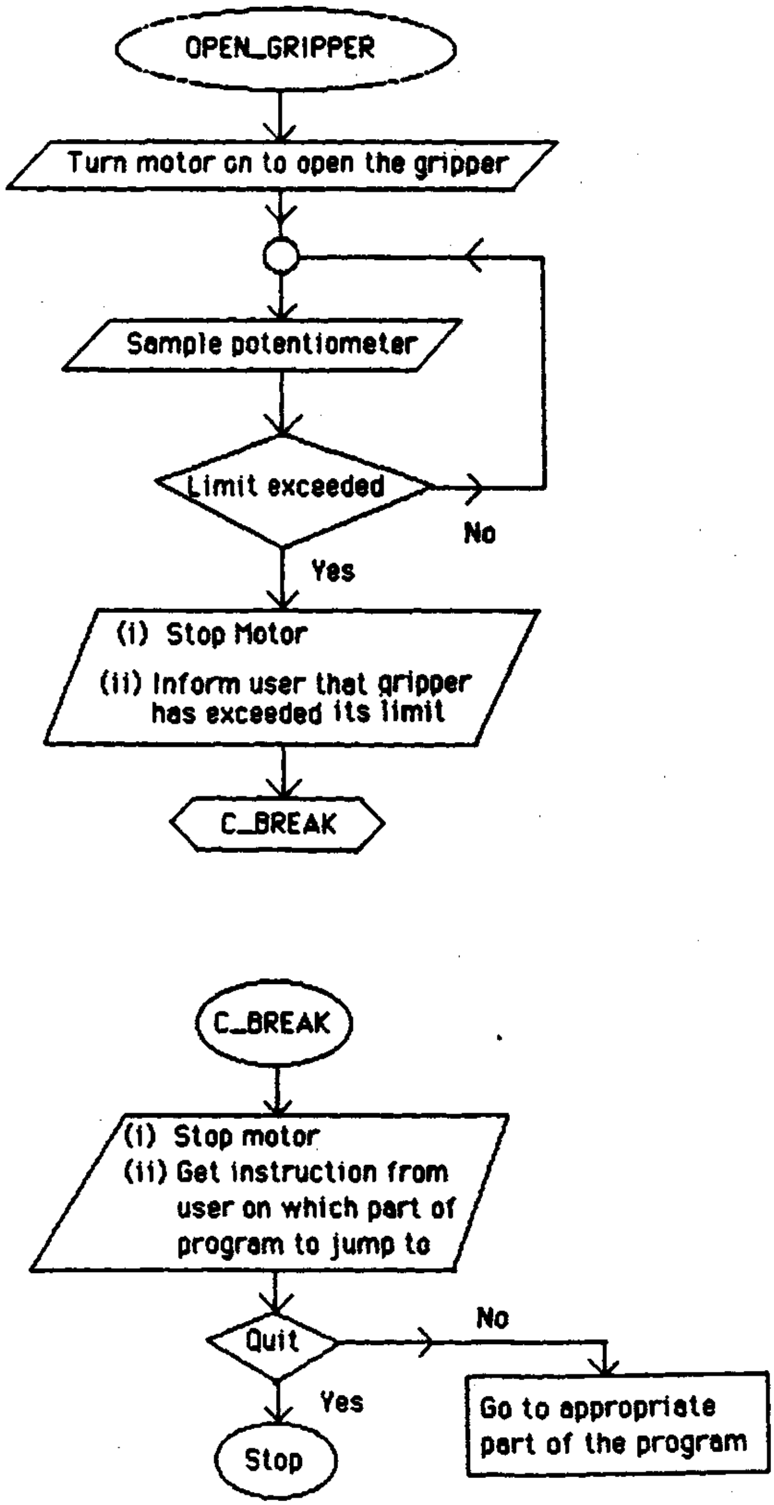


Figure 5.1 Contd.

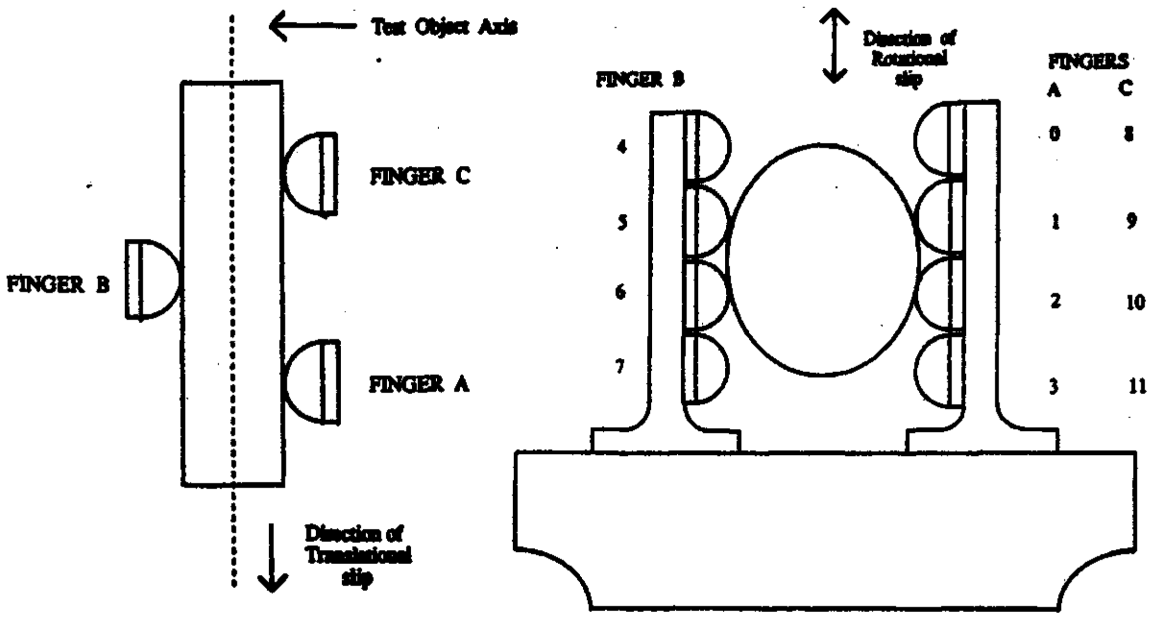
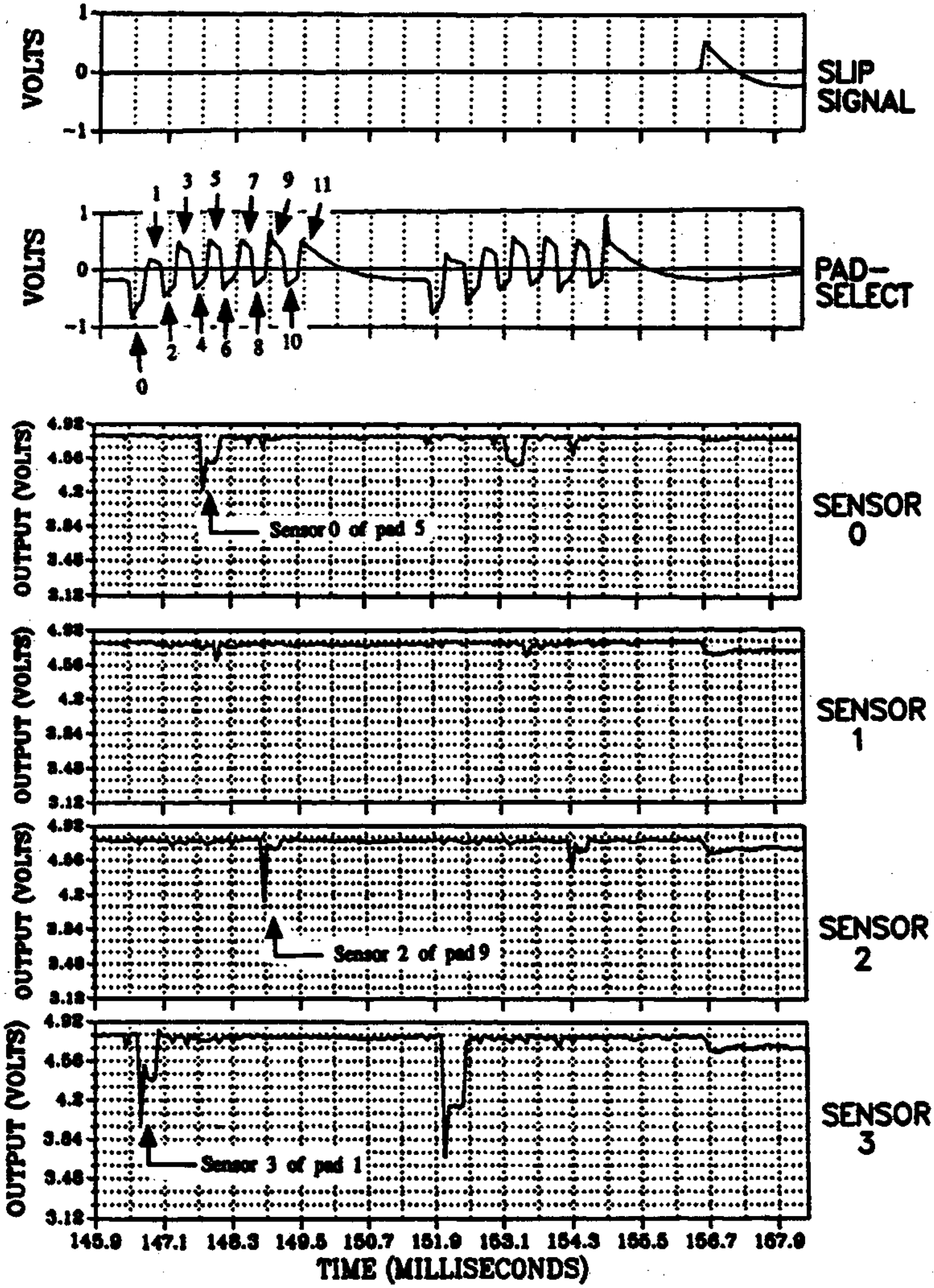


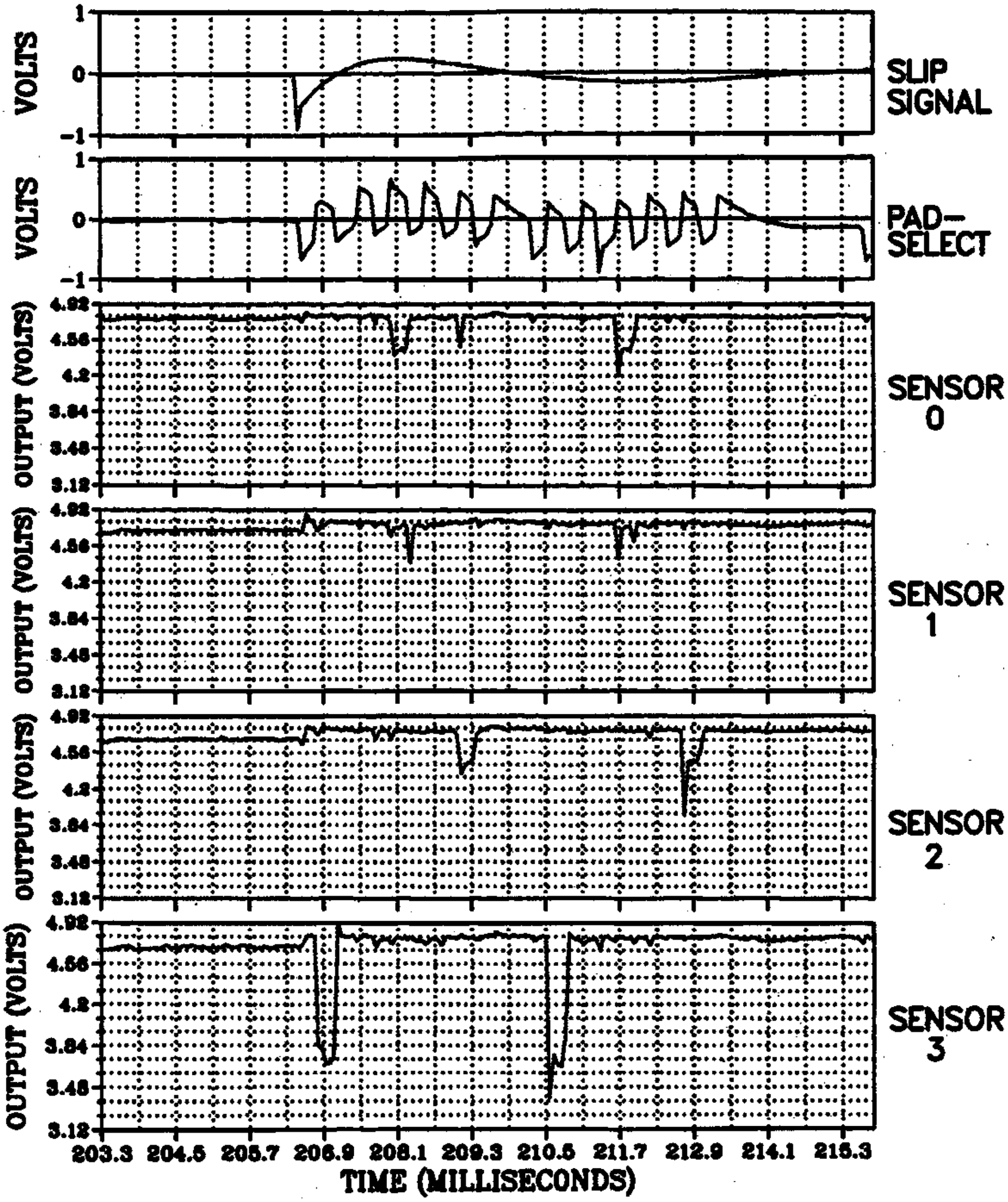
Figure 5.2: Gripper and Slip Directions

Each gripper finger has a total of 4 pads. Pads 0 to 3 lie on finger A, pads 4 to 7 lie on finger B, and pads 8 to 11 lie on finger C. The results are shown in Figures 5.3 to 5.5. There are six graphs in each figure. The horizontal axis for each graph is the time axis, while the vertical axis is the voltage axis. The divisions on the horizontal axis of each graph are similar, so the horizontal axis for the bottom graph is the only one that has been labelled. The slip-signal and pad-select graphs in each figure represent the signals that were recorded by the direct recording channels of the data recorder. These channels can only record ac voltages. The slip-signal graph represents the signal that the microcomputer puts out on the PIO when slip has been detected, and the signal after the gripper has increased its grip on the test object. The pad-select graph represents the least significant bit of the digital output lines that select the pads, and therefore shows the switching times



(a)

Figure 5.3: Translational Slip While Sampling Three Fingers



(b)

Figure 5.3 Contd.

for each pad. The remaining four graphs represent the signals that were recorded by the frequency modulated channels of the data recorder. These voltage signals are from the sensors that lie underneath the respective pads. The ordinates of the graphs represent forces on each sensor as it is selected. The frequency modulated channels of the data recorder can record both constant and changing voltages.

5.3.1. Grasp Control with Sampling of Three Fingers

Figures 5.3(a) and 5.3(b) show the results for the test carried out with sampling done on three fingers. The selection sequence of the pads was from pad 0 to pad 11. Figure 5.3(a) shows when slip occurs. In this figure, the pad-select graph shows the pads being sampled during two time intervals. During the time interval between 145.9 and 149.6 ms, all the pads were sampled. The sensor 0 graph shows that the object was gripped by pad 5; the sensor 2 graph shows that the object was also gripped by pad 9, and the sensor 3 graph shows that the object was gripped by pad 1. From the calibration graphs of Figure 4.4, for pad 1, and Figure 4.5, for pad 9, sensor 2 graph indicates that the force on pad 9 was about 135 grams, while sensor 3 graph indicates that the force on pad 1 was 300 grams.

As the sensors were sampled continuously, spikes were observed in the output signals of the sensors. These spikes were most likely due to switching transients in the circuitry driving the pads. In order to ensure that the A/D conversion started after the spikes had died down, a 0.07 ms delay was introduced after each pad was switched on or selected. Each pad was switched on for a total of 0.27 ms. This time included the delay time and the time it took to sample all the sensors underneath the selected pad.

After all the pads were sampled, the results were compared with the results from the previous sampling. The comparison is shown in the pad-select graph of Figure 5.3(a) between 149.6 and 151.8 ms. The scan time is determined from the time it took to sample the sensors, and to check for slippage (or process the data) by comparing the new data with data that was acquired in the preceeding sampling interval. When measured on the oscilloscope, the scan time was 5.35 ms, and the sensor data processing time was 2.11 ms. As there were no voltage changes above the threshold value, sampling was repeated.

This is shown in the time interval between 151.8 and 155 ms. During this sampling interval, sensor 3 was the only sensor which measured a voltage change above threshold. The magnitude of the voltage change was 0.24 volt. This change, which is shown after the time of 151.9 ms, caused the microcomputer, while comparing the samples, to put out a low-to-high signal on the PIO. This low-to-high signal, which is shown on the slip-signal graph at time 158.6 ms, also indicates that the motor was turned on, thereby causing the gripper to increase its grip on the test object. The motor was turned on for a period of 50 ms. Figure 5.3(b) shows the results for the time periods just before, and after the motor was turned off. When the motor was turned off, the microcomputer put out a high-to-low signal on the PIO. This signal appears after 206.3 ms on the slip-signal graph of Figure 5.3(b). The pad-select graph shows that the pads were sampled immediately after the motor stopped. Two sampling intervals are shown in this graph. The first interval was between 206.3 and 209.9 ms. The samples that were obtained from this sampling interval were not compared with previous samples because sampling began after the gripper increased its grip on the test object. Comparison of samples occurred at the end of the second sampling interval, at about 213.5 ms. The increase in grip of the test object resulted in changes in the voltages, as shown on the graphs for sensors 0, 2 and 3. These changes show increases in the forces that were applied to sensor 0 of pad 5, sensor 2 of pad 9, and sensor 3 of pad 1. From the calibration graphs of Figures 4.4 and 4.5, the force on pad 9 was increased to about 300 grams, while the force on pad 1 was increased to about 720 grams.

5.3.2. Grasp Control with Sampling of Two Fingers

The grasp control experiment was also carried out with sampling restricted to the pads on two fingers. The results are shown in Figure 5.4. Eight pads were sampled. The sampled pads were from pad 4 to pad 11. Figure 5.4 shows two sampling periods. In the period that is shown between 164.6 ms and 167 ms, the sensor 0 graph indicates that the object was gripped by pad 5, the sensor 1 graph indicates that the object was gripped by pad 6, and the sensor 3 graph indicates that the object was gripped by pad 10. The results show that the object was sensed by the sensors on pads 5, 6, and 10. Since pads 5 and 6 lie adjacent to each other, the results indicate that when the object lies between two pads, it can still be sensed by the sensors of both pads. Comparison of results for slip, or data

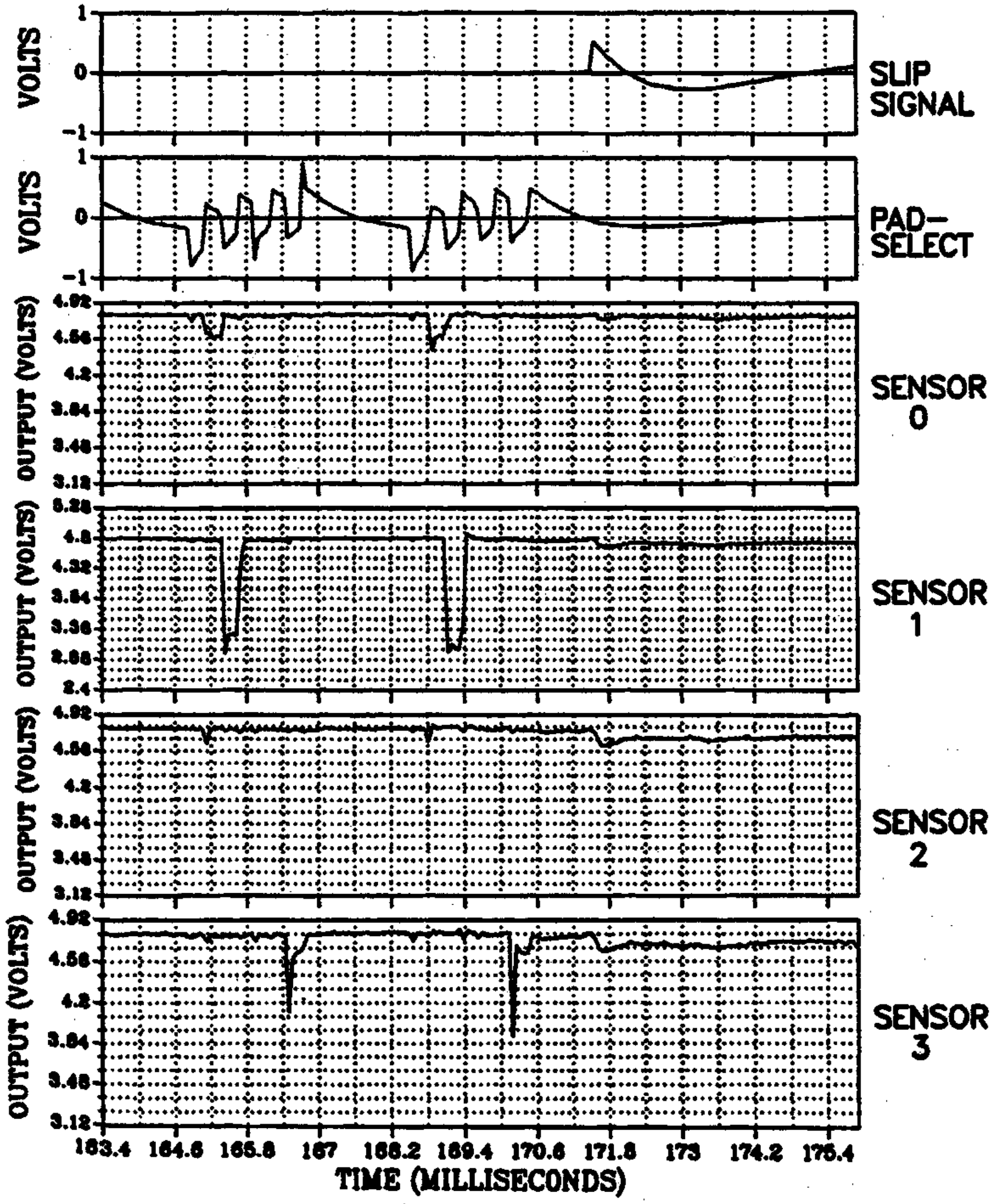


Figure 5.4: Translational Slip While Sampling Two Fingers

processing time, took 1.53ms, while the scan time was 3.7ms. The comparison of results is shown in Figure 5.4, on the pad-select graph between 166.7 ms and 168.5 ms. Another sampling period occurred after the comparison. During this period, the object was made to slip. This caused a voltage change above threshold on sensor 1 of pad 6. As a result of this measured change, a low-to-high slip signal was given by the microcomputer, and the gripper motor was turned on, as shown in the slip signal graph.

5.3.2.1. Grasp Control with Sampling of One Finger

The grasp control experiment was also carried out with sampling performed on one finger. The pads sampled were pads 0 to 3, which are on finger A. The results, which appear in Figure 5.5, show that the object was sensed by sensors 2 and 3 of pad 1. When the object started to slip, changes occurred on both sensors. Sensor 2 measured an increase in voltage (which meant a decrease in applied force), and sensor 3 measured a decrease in voltage (which meant an increase in applied force). The two changes were above the threshold value. This was recognised by the microcomputer and the grip on the test object was increased. The data processing time was 0.97 ms, and the scan time was 2.05 ms.

5.4. Grasp Control During Rotational Slip

The grasp control experiment was carried out for the case of rotational slip. The experiment was carried out with sampling of three fingers, two fingers, and one finger respectively. In addition to grasp control, the microcomputer stored in a file, information on the sensor from which it had recognised that changes above threshold had occurred.

5.4.1. Grasp Control with Sampling of Three Fingers

The results of the experiment in which sampling was carried out on three fingers are shown in Figure 5.6. In the sampling interval that is shown between 77.5 ms and 80.8 ms, the object was sensed by sensor 0 of pad 5, sensor 1 of pad 6, sensor 2 of pads 1 and 5, and sensor 3 of pad 1. Changes in some of the voltage readings appeared in the next sampling period. These changes were a decrease in voltage of sensor 2 of pad 1, and an increase in voltage of sensor 3 of pad 1. The changes show that the object had slipped and caused a decrease in the force on sensor 3 of pad 1, and an increase in the force on

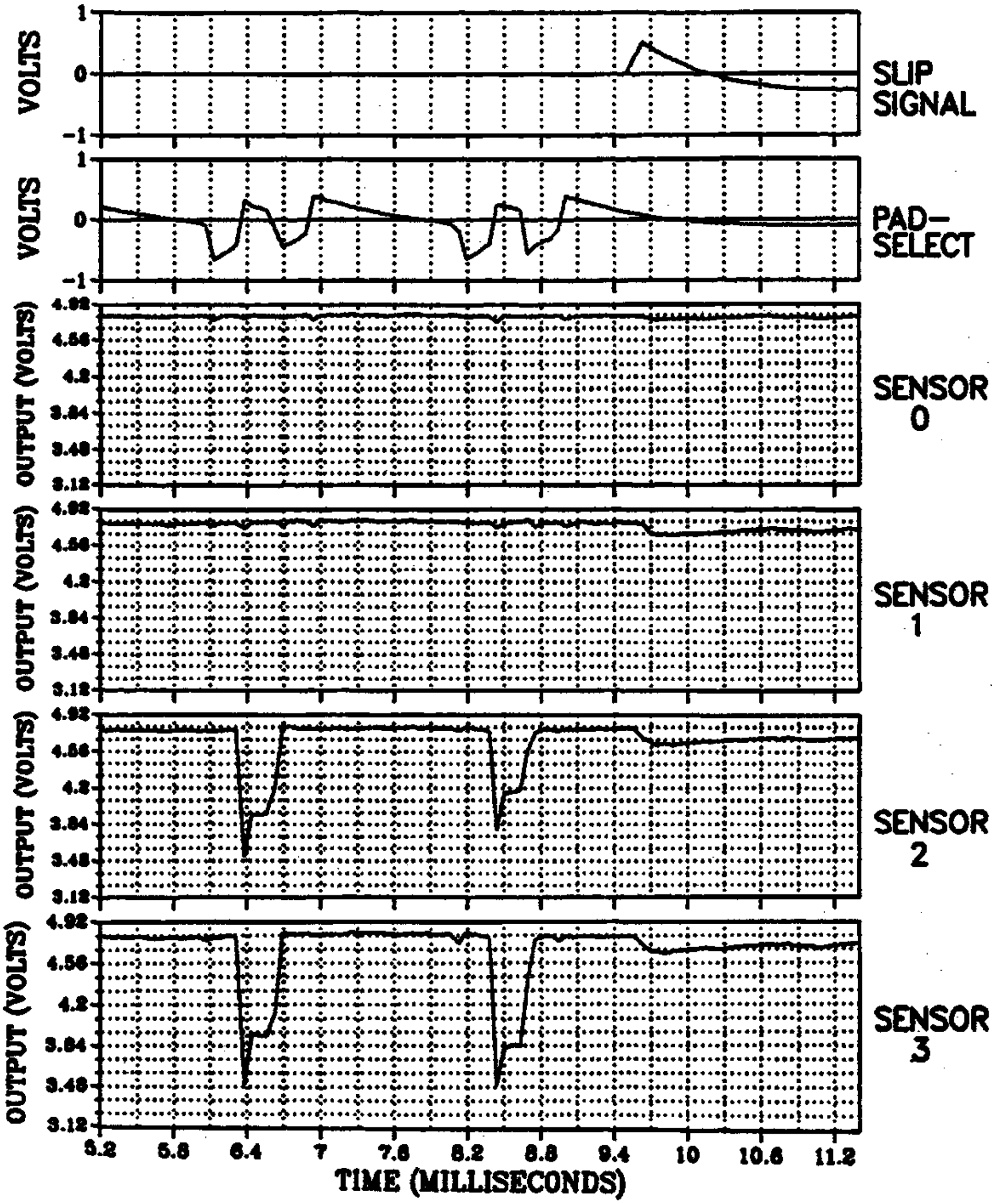


Figure 5.5: Translational Slip While Sampling One Finger

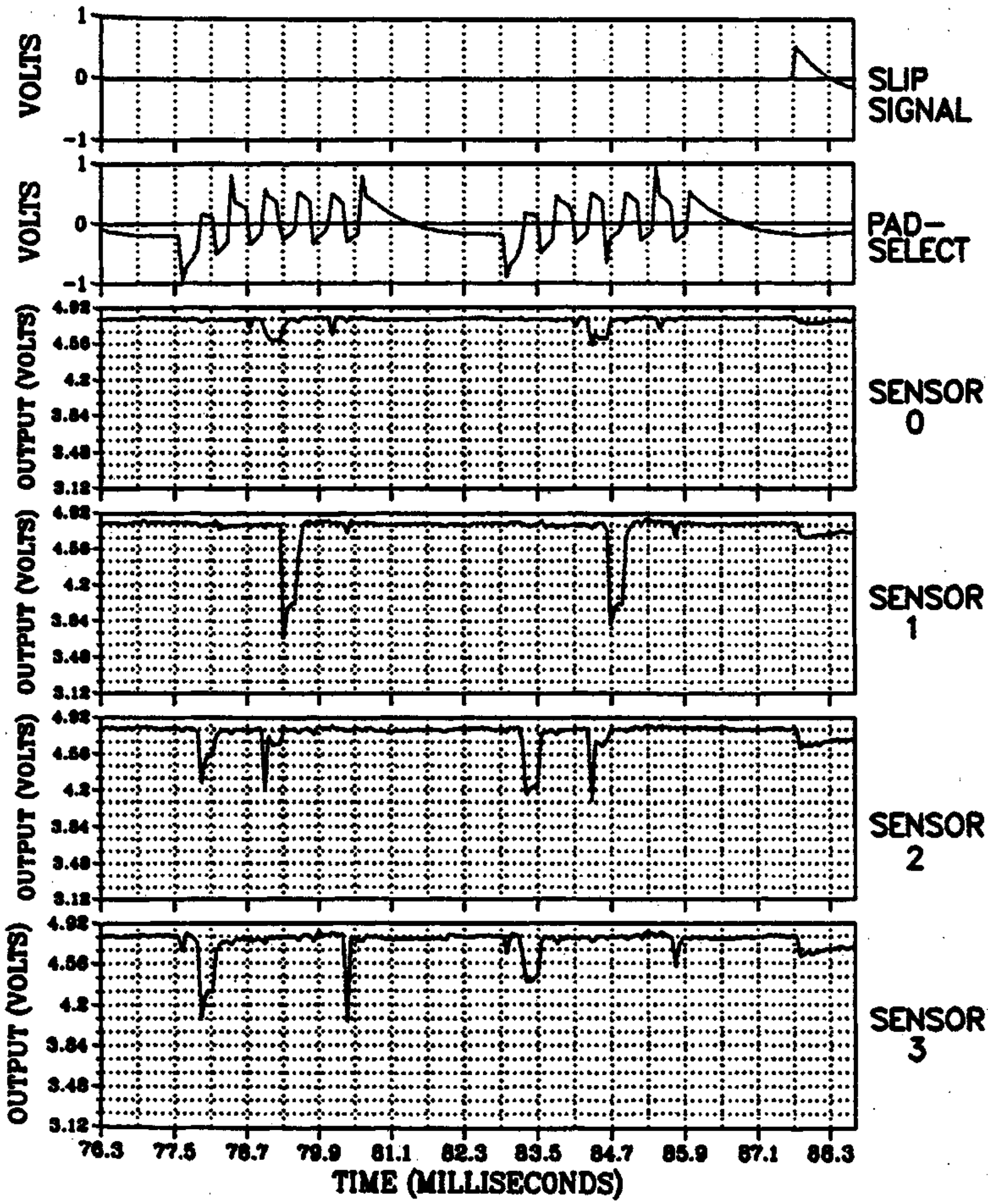


Figure 5.6: Grasp Control During Rotational Slip

sensor 2 of pad 1. Table 5.1 shows the results that were written to the disk file by the microcomputer. Although all the sensors were sampled, sensor 2 of pad 1 was the first sensor on which slip was detected, so this was the only result that was written to the file. The results from Table 5.1 indicate that during the sampling interval before slip, a reading of value 245 (= 4.57 v) was recorded from sensor 2 of pad 1. During the following

Table 5.1: Slip detected by the Microcomputer

```
PAD  1   SENSOR  2

1st val = 245  = 4.570 v
2nd val = 238  = 4.297 v
change = -7
```

sampling interval, the new value for the same sensor was 238 (= 4.297 v). While comparing the values from all the sensors, the microcomputer recognised a change of -7 (= -0.273 v) on sensor 2 of pad 1. The change, which had a magnitude that was greater than the threshold, indicated that slip had occurred. This caused the gripper to increase its grip on the test object.

5.4.2. Grasp Control with Sampling of Fewer Fingers

The grasp control experiment was carried out with sampling of two fingers, and after that, one finger. The microcomputer was able to detect slip for both cases. The scan times were similar to those of the translational slip experiment.

5.5. Analysis Of Grasp Control Results

The results from the grasp control experiments show that when the test object was grasped with a pad of the gripper, not all the sensors of the pad indicated that the object has been grasped. This was due to the magnitude, and points of contact of the forces that were exerted on the gripper pads. The results also showed that the scan time improved when fewer pads/fingers were sampled. An improvement of the scan time means a reduction in the cycle time for sampling each of the sensors. However, an observation of the results show that the more sensors that are sampled, the better the chances of sensing when the grasped object starts to slip because when the object starts to slip, some of the sensors do not indicate that the object has changed its position.

6. CONCLUSION

6.1. Summary

For a robot system to be able to successfully manipulate objects, it should be able to detect when the object has been grasped, and when the object starts to slip. Once slip has been detected, it should be able to apply enough force on the object to stop further slipping. This project presents the design of a system which uses sensory feedback from the tactile sensors of a robot gripper to control grasping.

The gripper, which has three fingers had been designed and built in the Departments of Electrical and Mechanical Engineering, at the University of Saskatchewan. The three fingers provide a stable gripping configuration that enables the gripper to handle many shapes and sizes of objects or workpieces. An object is gripped with gripping pads on the gripper. The object can be gripped on its internal surface (where applicable), and on its external surface. The pads, which are made from polyurethane material, have a high coefficient of friction, and do not damage or scratch the surface of the workpiece. Underneath the pads are compliant material, tactile sensors and circuits.

The tactile sensors of the gripper are made from electrically conductive polymer material, and sold under the trade name Force Sensing Resistor (FSR). The resistivity of each sensor drops as an increasing force is applied to its surface. There are four sensors underneath each pad, and four pads per finger, leading to a total of 48 sensors for the gripper. The sensors are connected to multiplexers, and are selected sequentially. When a sensor is selected, it is connected to the data acquisition circuit and driven by a constant current source. The output voltage from the circuit is digitized for use by a microcomputer. When a force is applied to the selected sensor, the output voltage of the circuit decreases.

Calibration experiments were carried out to study the response of the sensors to changes in force applied on the pads. Experiments were also carried out to control the grasping process of the gripper. A computer program, named CLB.C, was written to perform calibration of the sensors, while another program, called PROJ.C, was written for grasp control. Both programs were written in C language.

The sensors under a pad were calibrated by selecting the pad, hanging weights at various points on the pad, and measuring the output voltage from each sensor underneath the pad. The calibration results showed that all the sensors had the same voltage output when there was no applied force. The results of the calibration also showed that the output voltage from the sensors depended on the magnitude and the point of application of the force. The sensors that were at the back of a pad responded more to forces applied at the back of the pad than to forces applied at any other point on the pad. The sensors at the front of the pad responded more to forces applied at the front of the pad than to forces applied at other points on the pad. The information from the sensor calibration experiments was used in the grasp control experiments to determine when the gripper had grasped an object, and when the object had started to slip. The test object was a hollow cylindrical workpiece with sandblasted surface.

The grasp control experiments were carried out with the object made to slip in translational and rotational directions. As the gripper closed on the object, contact with the object was determined by sampling the sensors of the gripper to check if the output voltage of any sensor was less than the no-load voltage. After the object had been grasped, slip detection was determined by sampling the sensors of the gripper to check if the magnitude of the change in the output voltage of any sensor was above a preset threshold. When slip was detected, the gripper increased its grip on the test object. The scan time is determined from the time it took to sample the sensors, and to check for slippage (or process the data) by comparing the new data with data that was acquired in the preceeding sampling interval. Sampling was carried out on the sensors underneath the pads of three fingers, two fingers, and one finger, respectively. When the pads on the three fingers were sampled, the scan time for each sensor was 5.35 ms; when the pads on two fingers were sampled, the scan time was 3.7 ms; and when the pads on one finger were sampled, the scan time was 2.05 ms.

6.2. Conclusion

A system has been designed, for an existing three-fingered robot gripper, which uses tactile sensory feedback to control grasping. The sensors of the gripper were calibrated to determine how the sensor readings varied with changes of the applied force. The calibration results showed that the system was able to detect changes in the magnitude and position of the force applied on individual pads. Information from the sensor calibration experiment was used in the system for grasp control.

The system is capable of sampling the sensors of the gripper to detect when the gripper has gripped an object, and when the object starts to slip. Sampling can be performed on the sensors of three fingers, two fingers, or one finger. It takes 3.24 ms to sample all the sensors on three fingers, and slip can be detected within 2.11 ms after sampling. It takes 2.17 ms to sample all the sensors on two fingers, and slip is detected within 1.53 ms after sampling is completed. In the case of one finger, the sampling time is 1.08 ms, and slip is detected within 0.97 ms after sampling. When slip is detected, the system puts out a signal and a motor is turned on to increase the grip and stabilize the grasp.

6.3. Future Research

The scan time for each sensor, which improved as the number of sampled fingers decreased, was limited by the software, the system delays, and the cycle time of the microcomputer. The scan time could be further improved by using a faster microcomputer, and by writing some critical code in assembly language.

The results have shown that not all the sensors under a pad responded when an object was gripped at the pad. Also, among the sensors that responded, when the object started to slip, not all the sensors showed changes above threshold. Therefore an improvement in the scan time meant that the probability of detecting slip decreased because fewer sensors were sampled. A new gripper could be designed with more care taken to ensure uniform response of each sensor. Sensors of a different size and shape could be used to examine these effects on gripper response. Smaller sizes could result in greater spatial resolution as well.

Grasp control is an aspect of force control. In the grasp control experiments, the actual force applied to the test object was not measured. Additional work could be carried out to incorporate the grasp control algorithm in a force control algorithm.

Eventually the gripper should be attached to a robot arm and the whole system used in a pick-move-and-place operation, to study how it behaves under these conditions.

REFERENCES

1. Wolovich, W. A. , *Robotics: Basic Analysis and Design*, CBS College Publishing, New York, NY 10017, 1987, pp. 13.
2. Blumenthal, Tuvia, " Economic effects of robotization in Japan", *Robotics and Autonomous Systems*, Vol. 6, October 1990, pp. 323 - 326.
3. Poole, Harry H., *Fundamentals of Robotics Engineering*, Van Nostrand Reinhold, New York, NY 10003 , 1989.
4. Johansson, R. and Westling, G., "Roles of glabrous skin receptors and sensorimotor memory in automatic control of precision grip when lifting rougher or more slippery objects.", *Experimental Brain Research*, Vol. 56, 1984, pp. 550-564.
5. Snyder, Wesley E. , *Industrial Robots: Computer Interfacing and Control*, PRENTICE-HALL INC., Englewood Cliffs, NJ 07632, 1985, pp. 236.
6. Jacobsen, S.C., McCammon, I.D., Biggers, K.B. and Phillips, R. P., "Design of Tactile Sensing Systems for Dextrous Manipulators", *IEEE Control Systems Magazine*, Vol. 8, No. 1, February 1988, pp. 3-13.
7. Muench, T. C. R., "Development of a Low Cost Robotic Grasping Mechanism with Tactile Sensing", M.Sc. Thesis, University of Saskatchewan, Saskatoon, Canada, 1989
8. Fearing, R. S., "Tactile Sensing Mechanisms", *The International Journal of Robotics Research*, Vol. 9, No. 3, June 1990, pp. 3-23.
9. Howe, R. D. and Cutkosky, M. R., "Sensing Skin Acceleration for Slip and Texture Perception", *Proceedings of 1989 IEEE International Conference on Robotics and Automation*, 1989, pp. 145-150.
10. Howe Robert D., Popp, N., Akella, P., Kao, I. and Cutkosky, Mark R., "Grasping, Manipulation, and Control with Tactile Sensing", *IEEE International Conference on Robotics and Automation*, 1990, pp. 1258-1263.
11. Bicchi, A., Salisbury, J. K. and Dario, P., "Augmentation of Grasp Robustness using Intrinsic Tactile Sensing", *Proceedings of 1989 IEEE International Conference on Robotics and Automation*, 1989, pp. 302-307.
12. Bao, C. and van Brussel, H., "A Sensory Controlled Gripper System", *Robotics and Autonomous Systems*, Vol. 6, 1990, pp. 283-295.
13. Wood, H. C., Vaidyanathan, C. S., Muench, T. C., Wilson, J. N., "An Industrial

Robotic Gripper with Tactile Sensing", *North Saskatchewan IEEE Symposium*, November 1989, SEDCO Center, Innovation Place, Univ. of Sask. Research Park.

14. Dornfeld, D., Handy, C., "Slip Detection Using Acoustic Emission Signal Analysis", *Proceedings 1987 IEEE International Conference on Robotics and Automation*, 1987, pp. 1868-1875.

15. Rangwala, S., Forouher, F., Dornfeld, D., "Application of Acoustic Emission Sensing to Slip Detection in Robot Grippers", *International Journal of Machine Tools of Manufacture*, Vol. 28, No. 3, 1988, pp. 207-215.

16. Interlink Electronics, *Force Sensing Resistor*, product technical specifications, Interlink Electronics, Santa Barbara, Califonia, 1986.

17. Malvino, A. P., *Electronic Principles*, McGraw Hill Book Company, 1984.

18. Vaidyanathan, C. S., "Tactile Sensing System for an Industrial Robot", M. Sc. Thesis, University of Saskatchewan, Saskatoon, Canada, September 1987

19. Critchlow, Arthur J., *Introduction to Robotics*, Macmillan Publishing Company, New York, NY 10022, 1985.

20. Liu, Yu-cheng, Gibson, Glenn A., *Microcomputer Systems: The 8086/8088 Family, Architecture, Programming, and Design*, Prentice-Hall of India Private Limited, New Delhi - 110001, 1986.

21. AT&T, *6300 WGS Processor User's Guide*, 1988.

22. MetraByte Corporation, *DAS-8 and DAS-8PGA User's Manual*, 440 Myles Standish Blvd., Tauton, MA 02780, January 1988.

23. MetraByte Corporation, *Data Acquisition and Control for IBM PC/XT/AT and Compatible Computers*, 440 Myles Standish Blvd., Tauton, MA 02780, 1988.

24. MetraByte Corporation, *DAC-02 Manual*, 440 Myles Standish Blvd., Tauton, MA 02780, July 1988.

25. *Turbo C User's Guide*, Borland International, 1800 Green Hills Road, P.O. Box 660001, Scots Valley, CA 95066-0001, 1988.

26. Veklerov, E., Pekelny, O., *Computer Language C*, Harcourt Brace Jovanovich, Publishers, Orlando, Florida 32887, 1989.

A. PROGRAM CLB.C

This appendix contains the algorithm and computer program that was used to calibrate the sensors of the gripper. The program is called CLB.C, and it was written in C language.

ALGORITHM FOR PROGRAM CLB.C

```
select_bit:

        Ask user to choose between 8 or 12 bit conversion;

        Set Das-8 register, no-load and maximum values;


select_pad:

        Ask user to select pad;

        Initialize data storage array sens[];


rpt_samp:

        Initialize counters, and
        transfer the contents of sens[ ] to pre_sens[ ];

        Sample the sensors of the selected pad, and
        store the results in array sens[ ];

        Calculate the output voltages of the sensors;

        Calculate changes in the sensor readings
        i.e. sens[ ] - pre_sens[ ];

        Display results on the screen;

way:
        If user wants to change pads, goto select_pad;

        else if user wants to calibrate a sensor
        goto cal_sens;

        else if user wants to change from 8 to 12 bit
        conversion or vice-versa goto select_bit;

        else if user wants to change the number of times a
        selected sensor is sampled goto change_samps;

        else if user wants to quit, goto end;

        else if user wants to repeat display of all
        4 sensor readings goto rpt_samp;
```

```
cal_sens:

        Ask user to select a sensor;

        Initialize counters;


samp_chn:

        Initialize sensor data storage array;

        Take samples of sensor data ;

        Call Subroutine AVE to calculate
        the arithmetic mean ;

        Display results on the screen;

        If user wants to repeat sampling of the
        selected sensor, goto samp_chn;

        else goto way;


change_samps:

        Print number of times sensor is sampled;

        Input the new number from the user;

        goto way;
```

<u>Subroutine AVE</u>

```
        Display sensor voltages and frequency of occurence;

        Calculate arithmetic mean of the new samples;

        Calculate arithmetic mean of
        new samples below no-load voltage;

        Calculate arithmetic mean of
        previous and present samples;

        Display results ;

        return;
```

```
/* PROGRAM CLB.C */

/* Program to calibrate the robot gripper */

#define     D8          0x330               /* base address of DAS8 */
#define     HIGH        4095  /* Highest no. expected from a/d */
#define     NO_OF_SENSORS  4
#include    <stdio.h>
#include    <math.h>
#include    <dos.h>

int         n, no_load , NO_OF_SAMPLES, dv[ HIGH + 1 ] ;
float       vlt,  fsd ;
double              psum , pcount , pave ;
double              cum_ave, last_count , last_sum ;
double              pj, sqrt( ) ;

main( )
    {
      int  pad, eoc, sens[ NO_OF_SENSORS ],
                 pre_sens[ NO_OF_SENSORS ], bit_len,
        a_d, tot_chng ;

      void        AVE( ) ,
         outportb( int opt, unsigned char val ) ;

      char  wd ;

      int   getche( ), count ;
```

```
        float  volt[ NO_OF_SENSORS ] , all_ave ;


        int    op, chn, cc ;

        unsigned char   inportb( int ptid ) ;

            NO_OF_SAMPLES = 2048 ;

select_bit :

    printf(" (Type 8 for 8 bit a/d conv.");
       printf(" or any other key for 12 bit)\n");

        bit_len = getche( ) ; printf("\n");

        bit_len = bit_len - '0' ;

        if ( bit_len == 8 )

          {
             a_d = D8 ;

             fsd = 256. ;

             no_load = 251 ;

          }

        else

          {

             bit_len = 12 ;

             a_d = D8 + 1 ;

             fsd = 4096. ;

             no_load = 4016 ;

          }
```

```
select_pad :

 printf(" Which pad do you want to calibrate (0 -> 11)?\n");
        printf(" (note:  type a for 10  and b for 11)\n");
        pad = getche( ) ; printf("\n");

        if      ( pad == 'a' || pad == 'A' ) pad = 10 ;
        else if ( pad == 'b' || pad == 'B' ) pad = 11 ;
        else                                 pad = pad - '0';


        pad = pad * 16 ;

        outportb (D8 + 2 , pad) ;


    for ( n = 0 ; n <= NO_OF_SENSORS -1 ; n++) sens[ n ] = 0 ;

rpt_samp :

        tot_chng = count = 0 ;

         for ( n = 0 ; n <= NO_OF_SENSORS -1 ; n++)

              pre_sens[ n ] = sens[ n ] ;


        for ( n = 0 ; n <= NO_OF_SENSORS - 1 ; n++ )
          {
             outportb (D8 + 2 , pad + n) ;  /* select pad + sensor */

             outportb ( a_d  , 0 ) ;       /*  start a/d conv. */

             while  (  (  inportb(D8+2)  &  128  )  !=  0  );  /*  end  of
                                                               conv.? */
             sens[ n ] = inportb(D8+1) ;

             if ( bit_len != 8 ) sens[n] = sens[n] * 16 + inportb(D8) / 16 ;

             if ( sens[n] < no_load )  count  =  count + 1 ;

             volt[n] = (float) (sens[ n ]) * 10. / fsd - 5. ;

          }
```

```
printf("\n PAD  %d\n", pad/16);

for ( n = 0 ;  n <= 1 ; n++ )
  {
     printf("\n SENSOR %d              SENSOR %d\n", n, n + 2 );

     printf(" %d = % .4fv    %d = % .4fv\n",
                 sens[n], volt[n], sens[n+2], volt[n+2]);

   if( (sens[n] - pre_sens[n]) != 0 ) tot_chng = tot_chng + 1 ;

   if( (sens[n+2] - pre_sens[n+2]) != 0 ) tot_chng = tot_chng + 1;

   printf("\n change = %d      change = %d\n", sens[n] - pre_sens[n],
                                  sens[n+2] - pre_sens[n+2 ] );

  }

    printf("\n No. of sensors in which ") ;
    printf("there are changes = %d\n ",tot_chng) ;


way :

printf("\n");
printf(" type p to change Pad, c to calibrate a sensor\n");
printf(" b to change Bit length,\n");
printf(" v to change no of samples per sensor, q to quit");
printf(" or any other key to show all sensor readings\n");

wd = getche( ) ; printf("\n"); printf("\n");

if ( wd == 'p' || wd == 'P' ) goto select_pad ;

else if ( wd == 'c' || wd == 'C' ) goto cal_sens ;

else if ( wd == 'b' || wd == 'B' ) goto select_bit ;

else if ( wd == 'v' || wd == 'V' ) goto change_samps ;

else if ( wd == 'q' || wd == 'Q' ) goto end ;

else                         goto  rpt_samp ;
```

cal_sens :

```
printf("\n");
printf(" Which sensor do you want to sample ?  ");

chn = getche( );  chn = chn - '0' ;
printf("\n");

op = pad + chn ;
outportb(D8 + 2, op);   /* select channel */

last_sum = last_count = cum_ave = 0 ;
```

samp_chn :

```
for ( n = 0 ; n  <= HIGH ; n++ ) dv[n] = 0 ;

for ( n = 0 ; n <= NO_OF_SAMPLES - 1 ; n++ )
   {

        outportb( a_d , 0);  /* start a/d conv. */

        while( ( inportb(D8+2) & 128 ) != 0 ) ; /* end of conv. ? */

        cc = inportb(D8+1);

        if ( bit_len != 8 ) cc = cc * 16 + inportb(D8) / 16 ;

        dv[cc] = dv[cc] + 1 ;

}
```

```
/* process results */

printf("\n");
printf(" PAD  %d    SENSOR   %d\n", pad / 16 , chn );

      AVE( ) ; /* Calculate average */

printf("\n");
printf(" No. of bits used for conversion = %d\n", bit_len );

printf(" Do you want to repeat sampling on this channel (y/n) ? ");

wd = getche( ); printf("\n");

if ( (wd == 'n') || (wd == 'N' ) ) goto way ;

goto samp_chn ;


change_samps :

      printf("\n");
      printf(" No. of samples per sensor = %d\n", NO_OF_SAMPLES);

  printf(" Total no. of samples = %d\n",
                                NO_OF_SAMPLES * NO_OF_SENSORS);

printf("Input new value for no. of samps. per sensor ");
      scanf("%d", &NO_OF_SAMPLES);

      goto  way ;

end : ;

  }
```

```
/***** SUBROUTINE AVE ****/

/**** Subroutine to calculate average ****/

void AVE( )
{

double              all_sum, all_ave , all_count ;

printf("\n ___v___       ___voltage___       ___freq of occurence___ \n ");

    for (n = 0 ; n <= HIGH ; n++ )
      {
         if ( dv[n] != 0 )
           {

             vlt  = ( (float) (n) * 10. / fsd ) - 5. ;

         printf(" %d          % .4f          %d\n", n , vlt, dv[n] );

           }

      }


   psum = pcount = pave = 0 ;


 for ( n = 0 ; n < no_load ; n++ )
   {

         if   ( dv[n] != 0 )

           {

              pcount  = pcount + (double) ( dv[n] )  ;

              psum = psum + (double) (n) * (double) (dv[n]) ;

           }

   }
```

```
    all_sum = psum ; all_count = pcount ;

  for ( n = no_load ; n <= HIGH ; n++ )
    {

        if  ( dv[n] != 0 )
          {

            all_count  = all_count + (double) ( dv[n] ) ;

            all_sum = all_sum +
                          (double) (n) * (double) (dv[n])  ;

          }

    }

     if ( all_count != 0 ) all_ave = all_sum / all_count ;

     printf("\n AVERAGE OF SUM OF ALL SENSOR READINGS = ") ;
     printf(" % .4f\n", all_ave ) ;

     printf("\n FOR ALL SENSOR READINGS ") ;
     printf(" BELOW NO-LOAD VALUE \n");

      if ( pcount != 0 ) pave = psum / pcount ;

      vlt  = ( pave * 10. / fsd ) - 5. ;

     printf("\n    AVE. = % .4f = % .4f v\n", pave, vlt );

     last_sum   = last_sum  +  psum ;

     last_count = last_count + pcount ;

     if ( last_count != 0. ) cum_ave   =  last_sum / last_count ;

     vlt  = ( cum_ave * 10. / fsd ) - 5. ;  printf("\n");

     printf("    CUMMULATIVE AVE. = % .4f = % .4f v\n",  cum_ave, vlt) ;

     printf("\n    TOTAL COUNT = %.0f\n", last_count);

}
```

B. PROGRAM PROJ.C

This appendix contains the algorithm and computer program that was used to perform grasp control. The program is called PROJ.C, and it was written in C language.

ALGORITHM FOR PROGRAM PROJ.C

```
        Setup user interrupt i.e.
        ctrl-c / ctrl-break subroutine;

        Initialise registers;

st_nd :

        Select pads to be sampled;

op_cls :


        If user wants to close the gripper fingers

            {
                call CLOSE_GRIPPER subroutine ;
                goto sec_phase ;
            }

        else if user wants to open gripper fingers,

                call OPEN_GRIPPER subroutine ;


sec_phase :

        Sample the sensors of the selected pads
        and store the results;

rpt_samp :

        Check for user interrupt ;

        Sample the sensors again, and store the results ;

        Check both set of results for slip ;

        If results indicate that the object has slipped

            {
                call CONTROL_ACTION subroutine ;

                discard all samples ;

                goto sec_phase ;
            }
```

```
        else
            {
                discard oldest samples ;
                goto rpt_samp ;
            }
```

<u>Subroutine OPEN_GRIPPER</u>

```
            /*  Subroutine to open the gripper */

        Put out a high signal on the PIO port ;
        Turn the motor on to open the gripper ;

Limit_check :

        If gripper limit is exceeded
            {
                stop motor ;
                sound bell ;
                call C_BREAK subroutine
            }

        goto limit_check ;
```

Subroutine CLOSE GRIPPER

```
    /*  Subroutine to close the gripper */

    Turn the motor on to close the gripper ;

sample_pads :

    Check for user interrupt ;

    Sample the pads and store results ;

    If object has been gripped

        {
            put a high (5v) signal on PIO port ;

            stop motor ;

            If earlier requested, write, to screen,
            the sensor which grip was detected ;

            put out a low signal on the PIO port ;

            return ;

        }

Limit_check :

    Sample potentiometer channel ;

    If gripper limit is exceeded

        {
            stop motor ;

            sound bell ;

            call C_BREAK subroutine
        }

    goto sample_pads ;
```

Subroutine C_BREAK

/* Subroutine to handle user interrupts */

Stop motor ;

Get prompt from user to change
any of the previous settings ;

Return to appropriate portions of the main program ;

Subroutine CONTROL_ACTION

Put out a high signal on PIO port ;

If earlier requested, write to screen, the sensor
which the microcomputer first detected
changes above threshold;

Also, if earlier requested,
write the same information to disk file ;

Turn on the motor to increase the grip on object ;

Wait for 50 milliseconds ;

Stop motor ;

Put out a low signal on PIO ;

Return ;

```
/* PROGRAM PROJ.C */

/* Program to control slipping */

#define D8      0x330       /* base address of DAS8 */
#define D8_1    0x331
#define D8_2    0x332
#define D2      0x340       /* base address of DAC2 */
#define D2_1    0x341
#define PIO     0x350       /* base address of PIO */
#define         BELL    7
#define         NO_OF_SENSORS 4
#define         SAMPS_PER_SENSOR 2
#define         BUFF_SIZE (SAMPS_PER_SENSOR * 12 )
#define         OP_LIM      118
#define         CL_LIM   80
#define         SLIP_DETECT   3
#include <stdio.h>
#include <setjmp.h>
#include <math.h>
#include <dos.h>
int     op_cL ;
jmp_buf     skp, jumper , ch_pd , w_wrt ;
char        c_dir, skip_info, want_wrt ;
```

```
int         getche( ) ;

int     START_PAD, END_PAD , NO_OF_PADS ;

int      cc, j, k, n, p, Lb , Hb , pad,

         last_spc, indx ;

int     minus_SLPD ;
void     outportb( int opt , unsigned char  val ) , exit( );

unsigned char   inportb(int ptid ), lmt_chk,

        sens[ BUFF_SIZE ] [ NO_OF_SENSORS ];

int          C_BREAK( ) , setcbrk ( ), getverify( ), getcbrk( ) ;

void     CLOSE_GRIPPER( ) , OPEN_GRIPPER( ), CONTROL_ACTION( ) ;


void     delay ( );

unsigned  mtr_pulse = 50 ;    /* millisecs */

unsigned char  pad_del, pad_count = 10 ;

FILE     *WRT ;



 main( )
{

      char   wd ;


      setcbrk( 1 ) ;  /* set control - break setting */

      ctrlbrk(C_BREAK) ;

      minus_SLPD = -1. * SLIP_DETECT ;
```

```
/*  set up the system   */

        outportb( PIO + 3 , 0x80 );/* PA of PIO is made an O/P port */

        Lb = 0 ; Hb = 128 ;

        outportb( D2, Lb); outportb( D2_1, Hb); /* Dac2 OP0 = 0v */

        setjmp(w_wrt) ;

        printf("\n Do you want to write results ") ;
        printf("to a data file ? (y/n)\n");

        want_wrt = getche( ) ; printf("\n");

        if ( (want_wrt == 'n') || (want_wrt == 'N') )
          {
            want_wrt = 'n' ;

            goto st_nd ;
          }

          want_wrt = 'y' ;

          WRT = fopen ( "wrt.dat", "wt" ) ; /* Open data file */

          printf("\n Data file name is WRT.DAT \n") ;

st_nd :       START_PAD = 0 ;  END_PAD = 11 ;

        printf("\n") ;
        putchar(BELL);
        printf(" Ready to sample from pad %d  to  pad %d\n",
                        START_PAD, END_PAD  ) ;

        printf("\n");
        printf(" Do you want to sample these pads ? (y/n)\n");
        c_dir = getche( ) ; printf("\n");

       setjmp(ch_pd) ;

        if      ( (c_dir == 'n') || (c_dir == 'N') ) goto strt_pd ;
        else if ( (c_dir == 'p') || (c_dir == 'P') ) goto strt_pd ;
        else                              goto npds    ;
```

```
strt_pd :

      printf("\n PADS ARE NUMBERED FROM PAD 0 TO PAD 11\n") ;

      printf("\n");
      printf(" Which pad do you want to start sampling ?\n");
      printf("  (note: type a for pad 10 and b for pad 11)\n");
      wd = getche( ) ; printf("\n");

      if     ( (wd == 'a') || (wd == 'A') )  START_PAD = 10 ;
      else if( (wd == 'b') || (wd == 'B') )  START_PAD = 11 ;
      else                                   START_PAD = wd - '0' ;

      if ( (START_PAD < 0 ) || (START_PAD > 11 ) ) goto strt_pd ;
end_pd :

      printf("\n");
      printf(" Which pad do you want to end sampling ?\n");
      printf("  (note: type a for pad 10 and b for pad 11)\n");
      wd = getche( ) ; printf("\n") ;

      if     ( (wd == 'a') || (wd == 'A') )  END_PAD = 10 ;
      else if( (wd == 'b') || (wd == 'B') )  END_PAD = 11 ;
      else                                   END_PAD = wd - '0' ;


      if ( (END_PAD < 0 ) || (END_PAD > 11 ) ) END_PAD = START_PAD ;

      if ( START_PAD > END_PAD )
        {
        cc = START_PAD ;  START_PAD = END_PAD ; END_PAD = cc ;
        }

npds   : NO_OF_PADS = END_PAD - START_PAD + 1 ;

      last_spc = NO_OF_PADS * 2 - 1 ;

        setjmp(skp) ;

      printf("\n Do you want to skip information on ");
      printf(" which pads slip occurs (y/n) ?\n");
      skip_info = getche( );

      if ( skip_info == 'N' ) skip_info = 'n' ;
```

```
opn_cls :

        printf("\n") ;
        printf(" Do you want to (o)pen or (c)lose the gripper ?\n");
        op_cL = getche( ) ; printf("\n");

          setjmp(jumper) ;

        if ( (op_cL == 'o') || (op_cL == 'O') )
          {

          printf("\n");
          printf(" Openning Gripper and sampling pads \n") ;
          printf(" To stop process : \n") ;
          printf(" Press ctrl-c or ctrl-Break \n");

          OPEN_GRIPPER ( ) ;
          }

        else if ( (op_cL == 'c') || (op_cL == 'C') )
          {
          printf("\n");
          printf(" Closing Gripper and sampling pads \n") ;
          printf(" To stop process : \n") ;
          printf(" Press ctrl-c or ctrl-Break \n");

          CLOSE_GRIPPER ( ) ;
          }


        printf("\n");
        printf("**** Checking for slippage from PAD %d  to  PAD %d ****\n",
                        START_PAD,    END_PAD   );
        printf("\n");
        printf(" Press ctrl-c or ctrl-Break  ") ;
        printf("to suspend sampling process \n" );
```

```
sec_phase :

     k = -1 ;

    for ( j = START_PAD ; j <= END_PAD ; j++ )
    {

      pad = j * 16 ;

      outportb (D8_2 , pad) ;  /* select pad */

      for ( pad_del = 0 ; pad_del <= pad_count ; pad_del++) ; /* delay */

       k  =  k + 1 ;


      for ( n = NO_OF_SENSORS - 1 ; n >= 0 ; n-- )
        {
         outportb (D8_2 , pad + n) ;  /* select sensor */
         outportb ( D8 , 0 ) ;        /*  start a/d conv. */

         while ( ( inportb(D8_2) & 128 ) != 0 )  ;/* end of conv.? */

         sens [ k ] [ n ] = inportb(D8_1) ;

        }

    }
```

```
rpt_samp :

      getverify( );  /* Check for user interrupt */

      if ( k == last_spc ) k = -1 ;

      for ( j = START_PAD ; j <= END_PAD ; j++ )

    {

       pad = j * 16 ;

       outportb (D8_2 , pad) ; /* select pad */

       for ( pad_del = 0 ; pad_del <= pad_count ; pad_del++) ; /* delay */

       k  =  k + 1 ;

      for  ( n = NO_OF_SENSORS - 1 ; n >= 0 ; n-- )
          {
           outportb (D8_2 , pad + n) ;  /* select sensor */
           outportb ( D8 , 0 ) ;        /*  start a/d conv. */

           while ( ( inportb(D8_2) & 128 ) != 0 )  ;/* end of conv.? */

           sens [ k ] [ n ] = inportb(D8_1) ;

          }

    }
```

```
    /*   compare results  */

   p = END_PAD ;

  indx = k - NO_OF_PADS ;

     if ( indx < 0 ) indx = last_spc ;

  for (j = k ; j >= k - NO_OF_PADS + 1 ; j-- )
    {

      for (n = NO_OF_SENSORS - 1 ; n >= 0 ; n-- )
      {

        cc = sens[j][n] - sens[indx][n] ;

        if ( ( cc > SLIP_DETECT ) || ( cc < minus_SLPD ) )
          {
            CONTROL_ACTION( ) ;

            goto sec_phase ;
          }

      }

     indx = indx - 1 ;  p = p - 1 ;

    }


  goto rpt_samp ;


}
```

```
/************ SUBROUTINE CLOSE_GRIPPER ******************/

      /* Subroutine to close the gripper */

      /* and to check if object has been grasped */

   void CLOSE_GRIPPER ( )

   {

       op_cL = 'c' ;

       /* Close Gripper by making DAC2  OP0 = 4v */

       outportb( D2, 0 );  outportb( D2_1 , 25 );

       k = -1 ;

sample_pads :

     getverify( );  /* Check for user interrupt */

     if ( k == last_spc ) k = -1 ;

     for ( j = START_PAD ; j <= END_PAD ; j++ )

      {

        pad = j * 16 ;

        outportb (D8_2 , pad) ;  /* select pad */

        for ( pad_del = 0 ; pad_del <= pad_count ; pad_del++) ; /* delay */

         k =  k + 1 ;
```

```
for ( n = NO_OF_SENSORS - 1 ; n >= 0 ; n-- )
  {
   outportb (D8_2 , pad + n) ;  /* select sensor */
   outportb ( D8, 0 ) ;         /*  start 8 bit a/d conv. */

   while ( (inportb(D8_2) & 128 ) != 0 ); /* end of conv.? */

   sens [ k ] [ n ] = inportb(D8_1) ;

  }

}


  /*  Check to find out if the object has been Grasped */


p = END_PAD ;

for (j = k ; j >= k - NO_OF_PADS + 1 ; j-- )
 {

   for (n = NO_OF_SENSORS - 1 ; n >= 0 ; n-- )

   {

     if ( sens[j][n] < 249 )
      {
        outportb( PIO, 1) ; /* Put out High sig. on PA */

        Lb = 0 ; Hb = 128 ; /* DAC2  OP0 = 0v */

        outportb( D2, Lb );
        outportb( D2_1 , Hb ); /* stop motor */

      if ( skip_info == 'n' )
       {
        putchar(BELL); printf("\n");

        printf("\n OBJECT GRIPPED AT ");
        printf(" PAD  %d   SENSOR  %d\n", p , n ) ;
        printf("\n");
        printf(" SENSOR READING = %d\n", sens[j][n] );
       }
```

```
            outportb( PIO, 0) ; /* Put out Low sig. on PA */

            return ;

          }

       }

     p = p - 1 ;

    }

Limit_check  :


     outportb (D8_2 , 4 ) ;    /* pot. sensor */
     outportb ( D8 , 0 ) ;            /*  start a/d conv. */

     while ( (inportb(D8_2) & 128 ) != 0 ); /* end of conv.? */

     lmt_chk = inportb(D8_1) ;

     if ( lmt_chk < CL_LIM  )

       {

         /* If Gripper exceeds limit */
         /* Stop motor by making DAC2  OP0 = 0v */

        outportb( D2, 0 );
        outportb( D2_1 , 128 );

        putchar(BELL);
        printf("\n") ;
        printf(" Gripper Limit exceeded \n") ;

        C_BREAK( );

       }


      goto sample_pads ;

   }
```

```
/*****************  SUBROUTINE  OPEN_GRIPPER  *********************/

    /*   Subroutine for openning the Gripper  */

void    OPEN_GRIPPER ( )
  {
      op_cL = 'o' ;

      outportb( PIO, 1) ; /* Put out High sig. on PA */

      Lb = 0 ; Hb = 250 ;        /* DAC2  OP0 = -3.7v */

      outportb( D2, Lb );  outportb( D2_1 , Hb ); /* open pad */

Limit_check  :

      outportb (D8_2 , 4 ) ;    /* pot. sensor */

      outportb ( D8 , 0 ) ;    /*  start a/d conv. */

      while ( (inportb(D8_2) & 128 ) != 0 ); /* end of conv.? */

      lmt_chk = inportb(D8_1) ;

      if ( lmt_chk > OP_LIM )
         {

           Lb = 0 ; Hb = 128 ; /* DAC2  OP0 = 0v */

           outportb( D2, Lb );
           outportb( D2_1 , Hb ); /* stop motor */

           outportb( PIO, 0) ; /* Put out Low sig. on PA */

           putchar(BELL); printf("\n") ;
           printf(" Gripper Limit exceeded \n") ;

           C_BREAK( ) ;

         }

       goto Limit_check ;

  }
```

```
/******************* SUBROUTINE C_BREAK ***********************/

    /*  Subroutine to handle ctrl-break and ctrl-c interrupts */

int     C_BREAK ( )

  {

     outportb( D2, 0 );
     outportb( D2_1 ,128 ); /* stop motor */


     printf("\n");
     putchar(BELL); putchar(BELL);
     printf("\n");
     printf("~~~~ Sampled from PAD %d  to  PAD %d ~~~~\n",
                START_PAD,    END_PAD   );

     printf("\n");
     printf("Type c to close gripper , o to open gripper\n");
     printf(" p to change pads  , q to terminate program\n");
     printf(" s to change information status on slippage\n");
     printf(" w to (or not) write to a data file\n");
     printf(" or any other key to sample pads\n");

     c_dir = getche( ) ; printf("\n");

     if      ( (c_dir == 'p') || (c_dir == 'P') ) longjmp(ch_pd, 1) ;
     else if ( (c_dir == 'c') || (c_dir == 'C') ) op_cL = 'c' ;
     else if ( (c_dir == 'o') || (c_dir == 'O') ) op_cL = 'o' ;
     else if ( (c_dir == 'q') || (c_dir == 'Q') )  exit( ) ;
     else if ( (c_dir == 's') || (c_dir == 'S') )  longjmp(skp,1) ;
     else if ( (c_dir == 'w') || (c_dir == 'W') )  longjmp(w_wrt,1) ;
     else                                  op_cL = 'z' ;

     longjmp(jumper, 1);
```

```
/**** SUBROUTINE CONTROL_ACTION ****/

/*    Subroutine to take control action when object is slipping */

void    CONTROL_ACTION ( )

 {

  float V1, V2 ;


     outportb( PIO, 1) ; /* Put out High sig. on PA */


     if ( skip_info == 'n' )
      {
        putchar(BELL);
        printf("\n");

        printf("\n");
        printf(" PAD  %d   SENSOR  %d\n", p , n ) ;
        printf("\n 1st val = %d   2nd val = %d\n",
                sens[indx][n], sens[j][n] );
        printf("change = %d\n", cc );

      }

     outportb( D2, 20 );
     outportb( D2_1 , 10 ); /* close pad : DAC2 OP0 = 5 v */

     if ( want_wrt == 'y' )
      {
        V1 = sens[indx][n] * 10. / 256. - 5. ;

        V2 = sens[j][n] * 10. / 256. - 5. ;

        fprintf(WRT, "\n PAD  %d   SENSOR  %d\n", p , n ) ;

        fprintf( WRT,"\n 1st val = %d  = % .3f v", sens[indx][n], V1 );
        fprintf( WRT,"\n 2nd val = %d  = % .3f v", sens[j][n], V2 );

        fprintf(WRT, "\n change = %d\n", cc );

      }
```

```
    delay ( mtr_pulse ) ;

    outportb( D2, 0 );
    outportb( D2_1 , 128 ); /* stop motor : DAC2 OP0 = 0 v */

    outportb( PIO, 0) ; /* Put out Low sig. on PA */

    return ;

}
```