SMART ROBOTIC ARM CONTROL THROUGH AUTONOMOUS OBJECT DETECTION AND RETRIEVAL

A Paper
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Vicky Ramnath Mahodaya

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Electrical and Computer Engineering

April 2010

Fargo, North Dakota

North Dakota State University
Graduate School

Title

SMART ROBOTIC ARM CONTROL THROUGH AUTONOMOUS

OBJECT DETECTION AND RETRIEVAL

By

Vicky Mahodaya

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

ABSTRACT

Mahodaya, Vicky Ramnath, M.S, Department of Electrical and Computer Engineering, College of Engineering and Architecture, North Dakota State University, April 2010. Smart Robotic Arm Control through Autonomous Object Detection and Retrieval. Major Professor: Dr. Mark Schroeder.

This paper outlines the design and development of a robotic arm that can automatically detect and locate an object within its working envelope, reach to the object, grab the object with the aid of force feedback, bring it back to the user and transition back to user-control mode. A smart robotic arm like this can be used to control a brain computer interface (BCI) enabled prosthetic arm. This could be done by controlling the arm's coarse motion using BCI signals to the arm in manual mode while the fine motion control required for object capturing and retrieval could be performed by the smart robotic arm upon automatic detection of the object.

The smart robotic arm was developed by interfacing force, distance and position sensors with a robotic arm possessing three actuated joints. Using feedback control and direct-inverse kinematic equations, the arm was controlled to perform autonomous object detection and retrieval.

ACKNOWLEDGMENTS

I would like to express the deepest appreciation to my advisory committee chair, Professor Mark Schroeder, who has the attitude and the substance of a genius; he continually and convincingly conveyed a spirit of adventure in terms of innovation and research. Without his guidance and persistent help, this paper would not have been possible.

I would also like to thank my advisory committee members, Professor Jake Glower, Professor Chao You and Professor Majura Selekwa, for their concern and guidance on my path to success in writing this paper.

DEDICATION

I would like to dedicate this paper to my parents; without their support and moral teachings I would not have made it to the USA to further my studies and achieve my career goals.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

1. INTRODUCTION

1.1. Background

Robotic arms have been around for last 50 years. Starting from their early design and development in 1954 there has been a vast improvement and variation in the design and uses of robotic arms [1, 2]. Prehensile robotic arms are nowadays used extensively in industry as assembly robots which greatly increase the efficiency of the process [2]. In the field of space exploration, robotic arms serve the purpose of repair and maintenance of satellites, telescopes and shuttles in outer space. The medical field benefits from robotic arm manipulators as tele-operated surgery and diagnosis of patients located at remote areas is now possible for doctors [3, 4]. Also, robotic prosthetics can benefit those who are missing a limb. Now the anthropomorphic robotic arm manipulator can replace their limbs and improve their quality of life. One of the latest advancements in the field of robotics is using robotic arms to help save lives as part of bomb diffusing squads and nuclear hazard management teams [3].

To perform these tasks as a real-time operation the robotic arm has to be sensitive as to not damage the items it is trying to hold. It should have a very accurate sense of position and desired location it wants to reach and should be able to sense the objects in its working envelope. The proposed enhancements to the robotic arm used in this work will be able to meet these constraints.

This introductory chapter includes a background, problem description, system overview and related work to the topic. Chapter 2 includes information about the sensors, their signal conditioning circuits and calibration. Chapter 3 provides details on the data

acquisition technique whereas Chapter 4 focuses on the actuators and their control. Chapter 5 details the approach and control algorithm while Chapter 6 contains a flow chart of the Labview program along with main-vi and sub-vi screenshots. Chapter 7 and Chapter 8 contain the results and conclusions respectively.

1.2. Problem Description

The objectives for the robotic arm are as follows:

- Detect an object that is placed in its periphery.
- Determine the location of the object and whether or not it is within the gripper handling criteria.
- Use a control algorithm to precisely move to the location of interest autonomously using feedback sensor data.
- Finally, it should be able to control the required force of the gripper to grasp an object without causing damage to it.

1.3. System Overview

The system can be basically divided into two sections, hardware and software. Figure 1 shows a system level block diagram. The hardware consists of a robotic arm manipulator, sensors and actuators and the software section consists of Labview software and its interface.

The arm manipulator is a five link planar robotic arm trainer from OWI Inc. The manipulator is instrumented to meet the requirements of the system by installing position, distance and force sensor. Of these three of the actuators were used, including the shoulder joint that provides the rotation up to 120^{o}, the elbow joint that has a maximum deflection

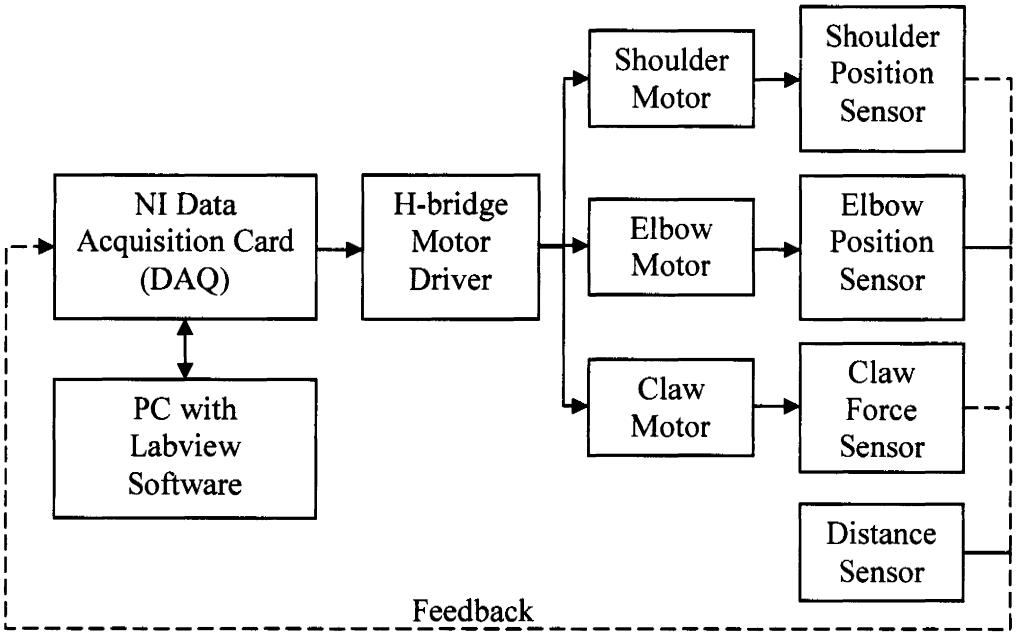


Figure 1 Functional block diagram of closed loop robotic control system

up to 135°, and the end joint that forms the claw of the arm and opens up to 50 mm (2 in) wide. Each joint is operated using a 3V DC motor. The arm is oriented to sweep through a range of 135° at a length of 228.5 mm. Position, force and distance were obtained by interfacing sensors with the robotic arm. Figure 2 shows the OWI Robotic arm trainer used to realize the base, shoulder, elbow, wrist and claw joints.

The rotary position sensor is used to provide the information regarding the angular position of the joints of the arm. Each of the joint is upgraded with a rotary potentiometer which gives the measurement of change in angle in terms of change in resistance.

A flexible piezoresistive sensor FlexiForce A201 from Tekscan Inc. was used to measure force. It can measure force in the range of 0 -1 lb and is used to acquire the information of the force that is being exerted by the claw on the target object. The sensor uses a pressure sensitive ink on the tip of the sensor that decreases the resistance when

subjected to force. Distance and proximity of the target object from the end effector are sensed using an ultrasonic sensor from Parallax Inc. The sensor works by measuring the time of flight of the ultrasonic burst from the transmitter back to the receiver.

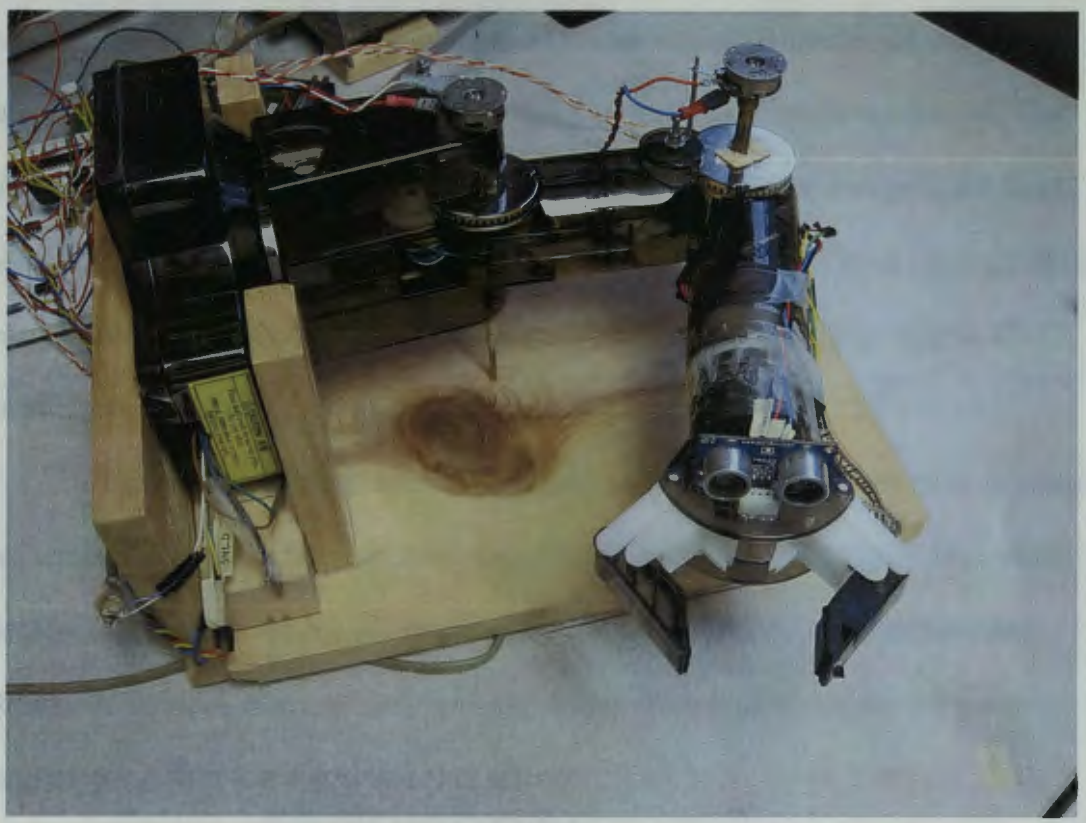


Figure 2 Assembled view of a robotic arm

The software package Labview 7.0; it contributes to the tasks of acquiring sensor data, performing calculations and sending out signals to the joint motors to move the arm. The software interacts with the hardware through the use of a data acquisition (DAQ) card. Kinematic models written in Labview and mathscript are used to control the arm while guiding it towards the target object location. Force, angle and distance sensors are used to provide feedback to the controller during the search, grasp and retrieve phases. Figure 2 shows the assembled view of the arm manipulator.

1.4. Related Work

A robot manipulator is a nonlinear, multivariable system. A wide variety of control techniques can be used to control such a system [2]. Some of the common algorithms and control methods used for robotic arm control are Lagrangian Dynamics [5], adaptive control [6], PID control, fuzzy logic control [7, 8] and neural network models [9]. The primary drawback of these methods for determining acceptable trajectories is the large amount of processing time needed to obtain a solution. Other higher level language based controls have also been used in the past but these methods have a higher degree of execution complexity.

Different kinds of sensors have been previously used with the robotic arms to make them smart and sensible. Some of these sensors are haptic sensors [10] that can identify the object by touching the surface to recognize the form of the object, color sensors that differentiate between colors of the object and vision sensors that compare the present image captured with one that is stored in memory.

Ultrasonic sensors have also been used to collect images [4] [11] that are then decoded using image processing techniques like Kalman filters and other composite image matching techniques [4]. Image guided control has been used based on the information from these sensors. However, these methods have complex algorithms and do not produce very effective results.

2. SENSORS, SIGNAL CONDITIONING AND CALIBRATION

2.1. Distance Sensor

The distance sensor, which also plays the role of proximity sensor, is the primary feedback device for locating the target object. It gives the information regarding the proximity within $\pm 20^{\circ}$ view angle and provides the distance from the object. It can also, to some extent, provide information about the visible surface shape which can be used in future development of the system. Shown below in Figure is the image of the sensor used in the system.

Figure 3 Ultrasound proximity and distance sensor

The sensor works on the principal of high frequency sound echo measurement, much like radar devices. It measures the distance from the object based on the time difference between the transmission of the sound wave and its echo. The sensor emits a short burst of ultrasound wave at 40 KHz that hits the object and returns to the sensor at the speed of sound in air, 340.29 meters/second. A pulse, whose width corresponds to the time

that it took for the wave to travel, is output from sensor. The ultrasound sensor usually comes in the transceiver form as shown in Figure 3 in which one cylindrical transducer acts as a transmitter and the other as a receiver for ultrasound waves.

The signal conditioning circuit [12] in Figure 4 shows the functioning of the sensor. A function generator provides a short excitation pulse at 10 Hz to the sensor to trigger the ultrasound burst from it. At the falling edge of the excitation pulse the ultrasound burst is emitted. As soon as the echo reaches the receiver the output pulse is created. Thus an output from the sensor follows the excitation pulse as shown at the bottom input to the AND gate in Figure 4. Logic gates are used to avoid the excitation pulse from being seen by the pulse width detector of the DAQ card. The pulse width is related to the distance, d, from the object through the equation d = speed of sound (m/sec) x pulse width (sec).

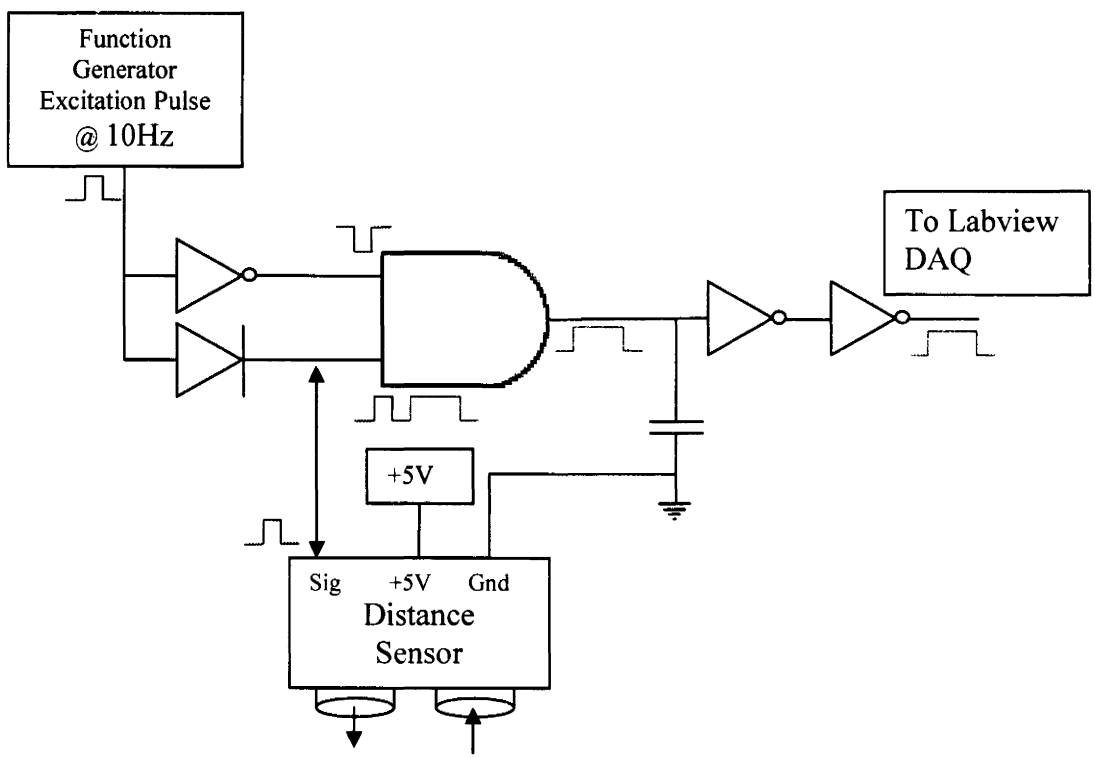


Figure 4 Distance sensor circuit

Calibration of the distance sensor was done using two different 6" tall target objects - a paper cylinder of approximately 1" diameter and a 1" x 1" wooden block. Each target was placed at different known distances from the sensor and distance was calculated from the width of the captured pulse. Results of the sensors calibration are shown in Figure 5 and Figure 6. According to the results the performance of the sensor is independent of the shape of the two target objects and the sensor has a nearly linear response for both objects.

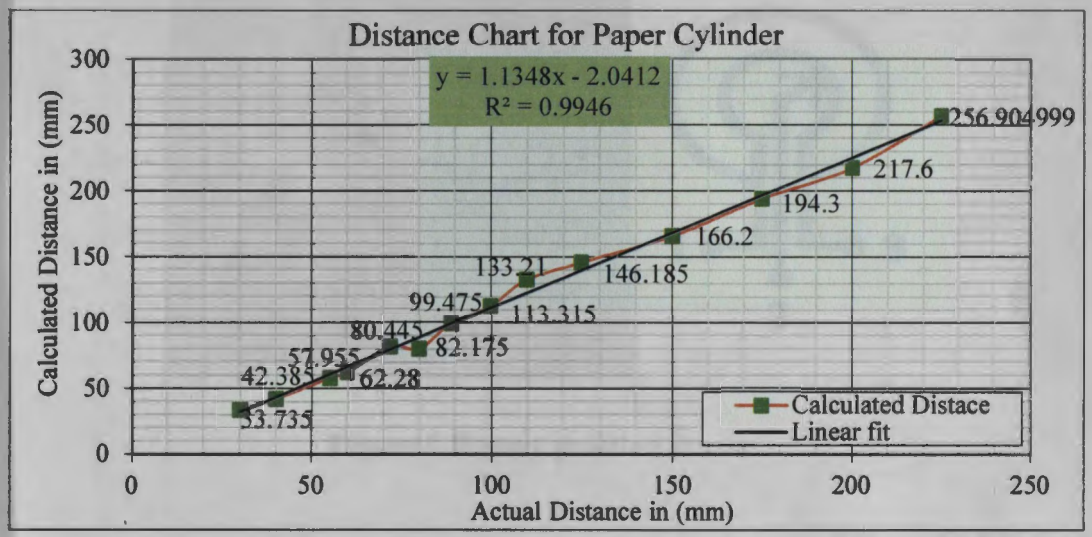


Figure 5 Distance chart for paper cylinder

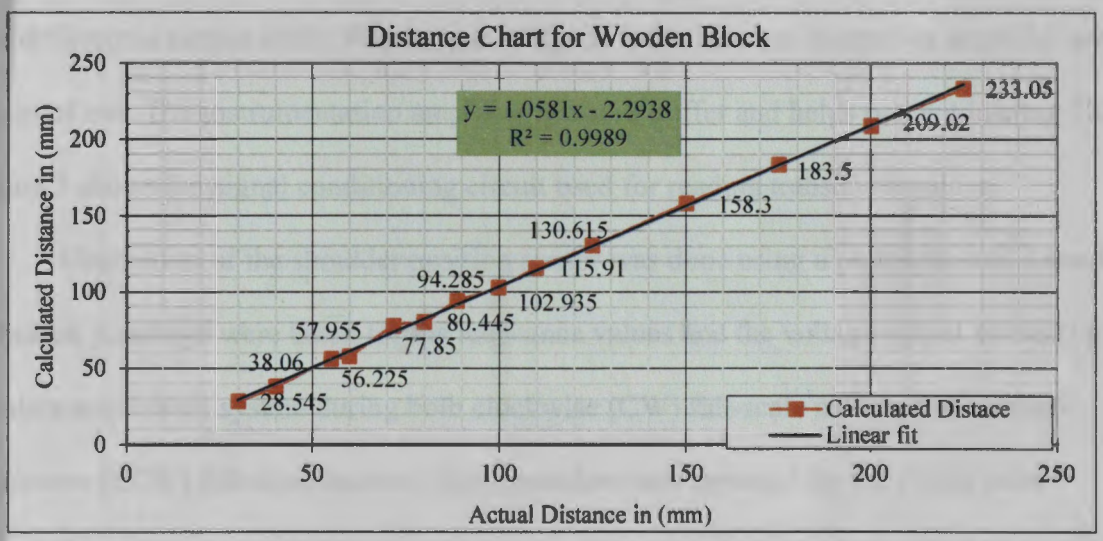


Figure 6 Distance chart for wooden block

2.2. Position Sensor

The potentiometer-based position sensors serve the purpose of feeding back the angular position of the links in order to complete the control loop. Two position transducers are located on the elbow joint and shoulder joint, respectively. Figure 7 shows an image of the potentiometer used in the system.

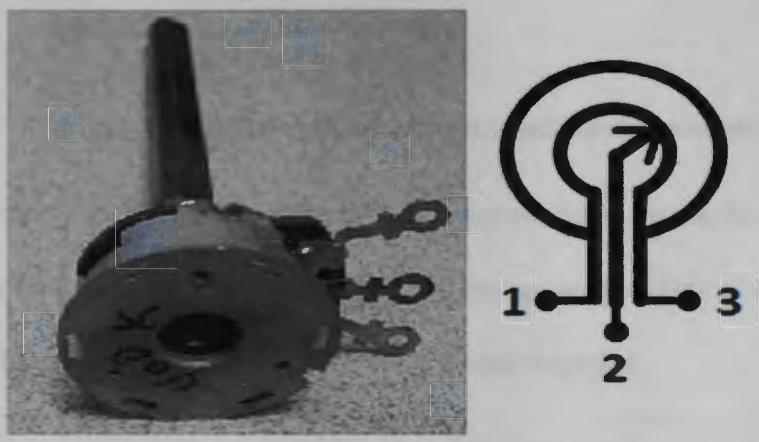


Figure 7 Rotary position sensor

The position transducers are single turn potentiometers of 6 KΩ and 10 KΩ for the shoulder and elbow position, respectively, each making up an arm of a Wheatstone bridge. The differential output of the Wheatstone bridge is fed to an instrumentation amplifier with a gain of one. The instrumentation amplifier acts as a buffer and helps prevent loading [12]. Figure 8 shows the signal conditioning circuit used for reading transducer values.

Calibration of the shoulder position sensor was done using a protractor and a needle indicator. Readings were taken for the resistance values and the voltage output as read by the data acquisition system during both clockwise (CW) full-scale motion and counter-clockwise (CCW) full-scale motion. This procedure was repeated for the elbow joint sensor. The calibration table for the position sensor can be found in Appendix I.

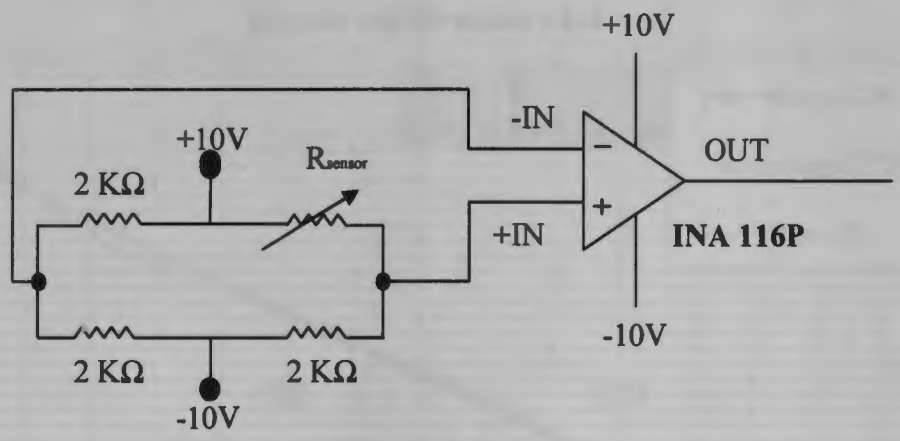


Figure 8 Position sensor output conditioning circuit

Figure 9 and Figure 10 show plots of the sensor readings and the linear fit equation used to calibrate both shoulder and elbow position sensors, respectively. Results from the plots show that both position sensors have a fairly linear response.

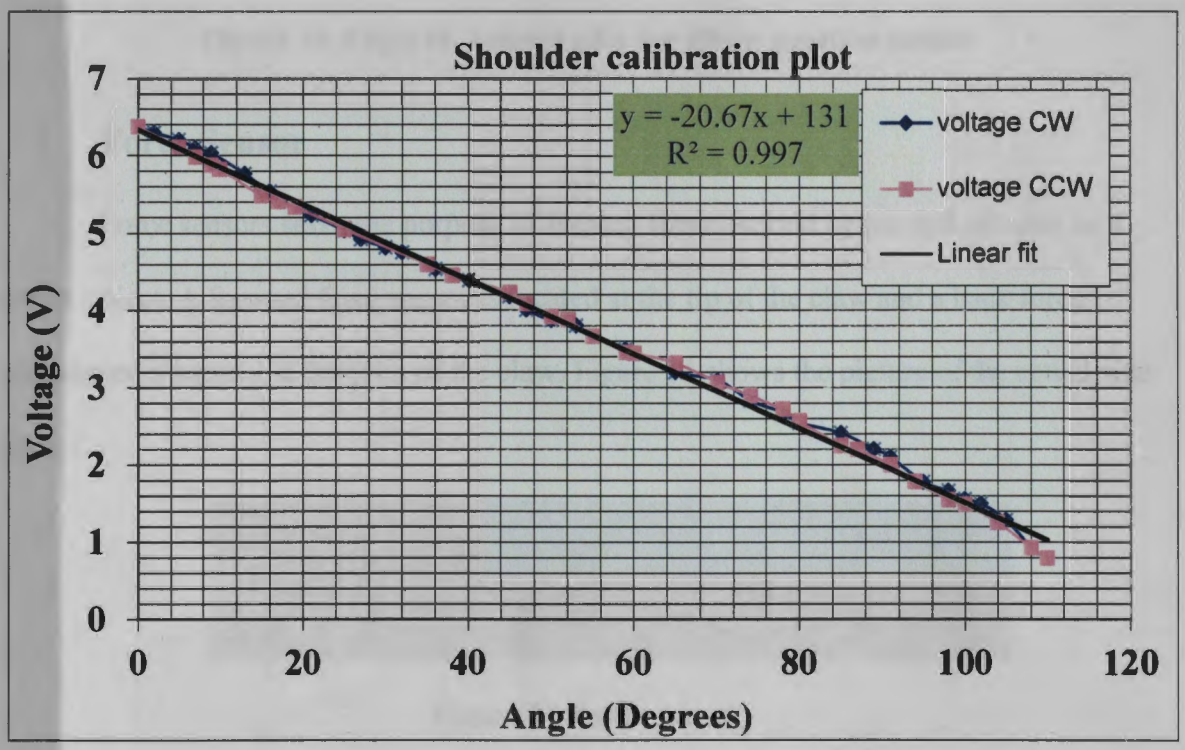


Figure 9 Angle vs. voltage plot for shoulder position sensor

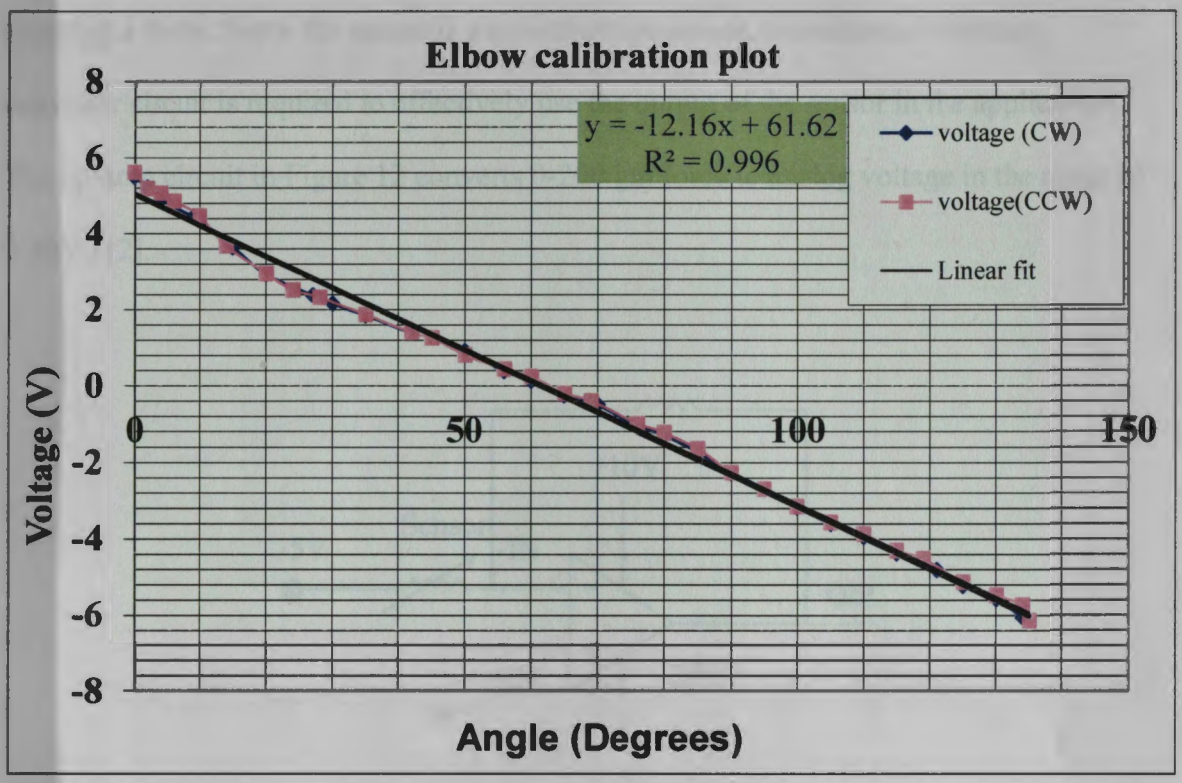


Figure 10 Angle vs. voltage plot for elbow position sensor

2.3. Force Sensor

Force sensors serve the purpose of sensing force exerted by the end effector on a target object. A forward force sensor is located at the tip of the claw and a back force transducer is located at the joint of the claw. Figure 11 shows the picture of the actual size sensor.

Figure 11 Force sensor

Basic construction of the transducer includes two layers laminated together; each consists of pressure sensitive ink. The ink resistance decreases exponentially when

applying a force. Since the sensor is a resistive type sensor, a resistance to voltage converter circuit is required to effectively use the output of the sensor in the application. The op-amp circuit in Figure 12 converts 0-200 gm force to analog voltage in the range of 0-10V [12].

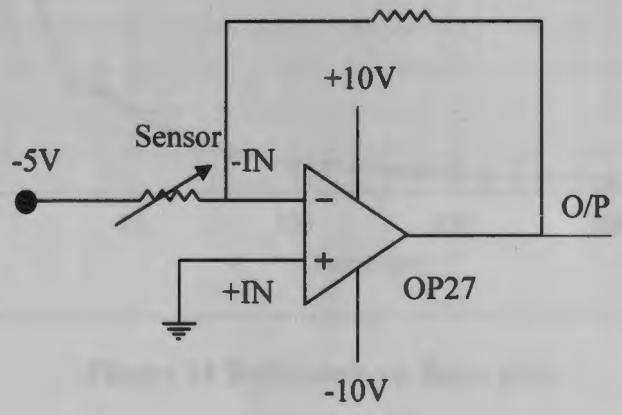


Figure 12 Force sensor circuit

The force sensor was calibrated against a set of standard weights in the range of 10-200 gm. Figure 13 is the plot of Voltage vs. Force that shows fairly linear response of the sensor and Figure 14 shows the plot of resistance vs. force.

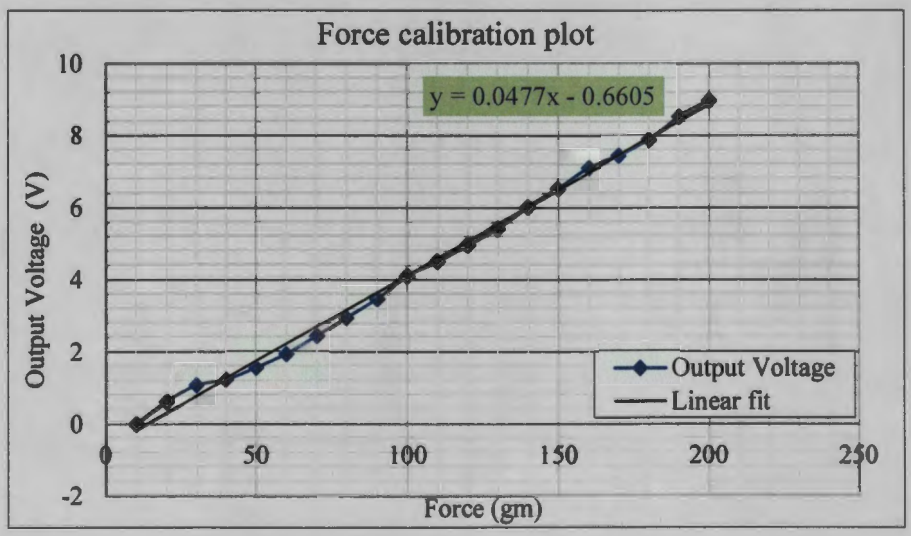


Figure 13 Output voltage vs. force plot for force sensor

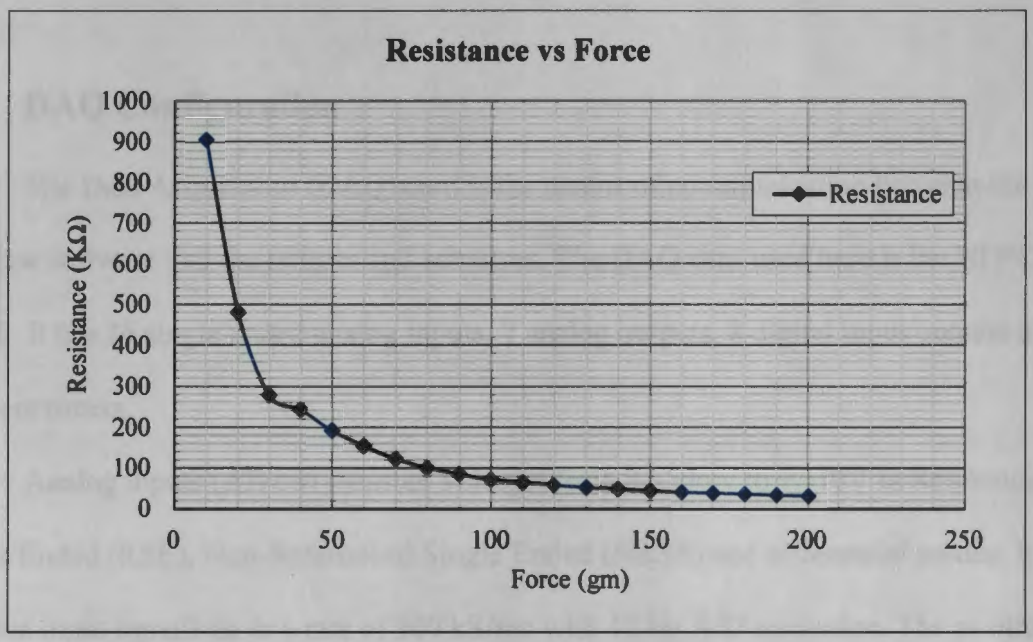


Figure 14 Resistance vs. force plot

3. DATA ACQUISITION

3.1. DAQ Configuration

The Data Acquisition (DAQ) card is the means of communication between the Labview software and the sensors and actuators. The DAQ card used here is the NI PCI-6040E. It has 16 single ended analog inputs, 2 analog outputs, 8 digital input/outputs and 2 counters/timers.

Analog inputs (AI) can measure voltages from +/-50mv to +/-10V in Referenced Single Ended (RSE), Non-Referenced Single Ended (NRSE) and differential modes. It can scan the input signals up to a rate of 500 kS/sec with 12 bit A/D resolution. The position and force sensors are connected to the analog input channels of the DAQ card. Six of the eight digital I/O pins are configured as digital outputs to control the H-bridge actuator circuits.

A 24 bit counter on the board is configured to measure the pulse width of the 5V TTL/CMOS signal generated by the distance sensor. The frequency of the counter can be set at 20 MHz or 100 KHz based on the expected range of the signal.

3.2. I/O Configuration

Table 1 shows the connections of the sensors to the DAQ connector. Table 2 shows the connections of the DAQ card to the three H-bridges. All the sensors are sampled once as per demand based on the loop rate which could be anywhere up to 500Hz except for distance sensor which is sampled at 4 Hz. Counter 1 output is used to sample the pulse

width measurement. Elbow, shoulder and force sensors are sampled through analog input channels.

Table 1 DAQ input distribution

INPUTS	Location on DAQ	Type	Sampling Rate
Distance Sensor	CTR 1 OUT +5V	Counter/Timer 1 5V supply	N Samples 4 Hz
Elbow Position Sensor	AI7 AI GND	Analog Input 7 - RSE Analog Ground	On demand up to 500 Hz
Shoulder Position Sensor	AI6 AI GND	Analog Input 6 - RSE Analog Ground	On demand up to 500 Hz
Back Force Sensor	AI0 AI GND	Analog Input 1 - RSE Analog Ground	On demand up to 500 Hz
Forward Force Sensor	AI8 AI GND	Analog Input 8 - RSE Analog Ground	On demand up to 500 Hz

Table 2 DAQ output distribution

OUTPUTS	Location on DAQ	Type
Elbow motor	P 0.2 P 0.3	TTL logic output to H-bridge TTL logic output to H-bridge
Shoulder motor	P 0.0 P 0.5	TTL logic output to H-bridge TTL logic output to H-bridge
Claw motor	P 0.1 P 0.6	TTL logic output to H-bridge TTL logic output to H-bridge

4. MOTORS, DRIVES AND CONTROL

4.1. Motors

Motors that are used in this design are part of the robotic kit from OWI. The motor is a permanent magnet brush-type, bidirectional, DC 3V motor with approximately 18,000 RPM at no load. The armature resistance of the motor coil is 3Ω and the continuous power consumption of the motor is up to 3W. A gearbox with gear ratio of approximately 18,000:1 is connected to each motor to increase the torque. The gearbox reduces the end effector RPM from 18,000 to 1.

4.2. Motor Control

Motors on the arm are controlled and driven by H-bridge circuits. The H-bridge is a configuration of transistors that enables voltage to be applied to the load in either direction, thus allowing the motors to run in either direction. The H-bridge that was used is the NJM2670 dual H-bridge driver. It is a solid state H-bridge driver consisting of BJTs as switches.

Figure 15 shows the diagram of a typical H-bridge circuit. An H-bridge has a minimum of 6 terminals. Vcc is connected to the positive power input and GND is connected to the power ground. Input 1 and Input 2 are for control logic from the DAQ card and Output 1 and Output 2 are connected to the two terminals of a DC motor. The four cases in Table 3 can be achieved using the DAQ digital I/O lines as inputs. In figure 15 red line shows the direction of the flow of current and clockwise rotation of motor, blue line represents counterclockwise motion of the motor.

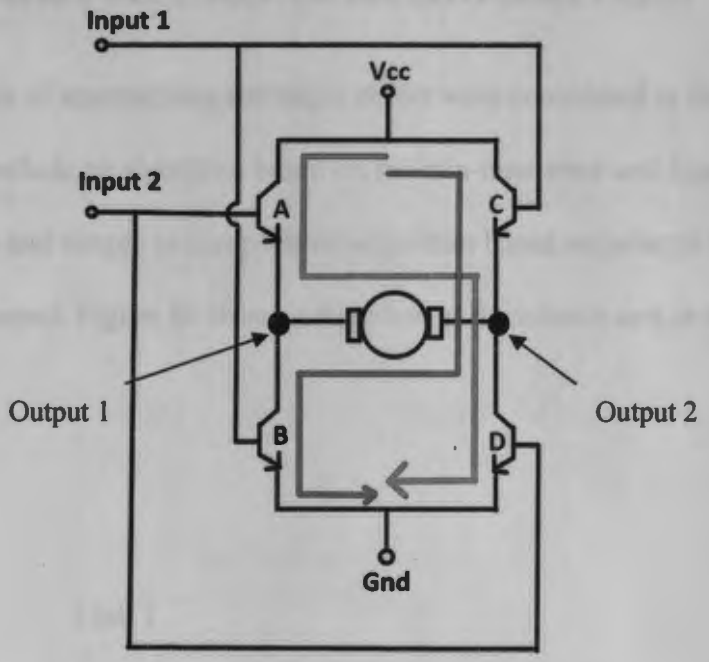


Figure 15 Motor control circuit

Table 3 Motor control logic

Input 1	Input 2	Switch (A)	Switch (B)	Switch (C)	Switch (D)	Motion
0	0	0	0	0	0	Free
0	1	1	0	0	1	CCW (Blue)
1	0	0	1	1	0	CW (Red)
1	1	1	1	1	1	Brake

5. TARGET APPROACH ALGORITHM

Different methods of approaching the target object were considered in this research project. Some of them include an algorithm based on the min-max error and fuzzy logic [8]. Finally, an effective and simple to comprehend algorithm based on polar to Cartesian conversion was implemented. Figure 16 shows a top view of the robotic arm in the Cartesian plane.

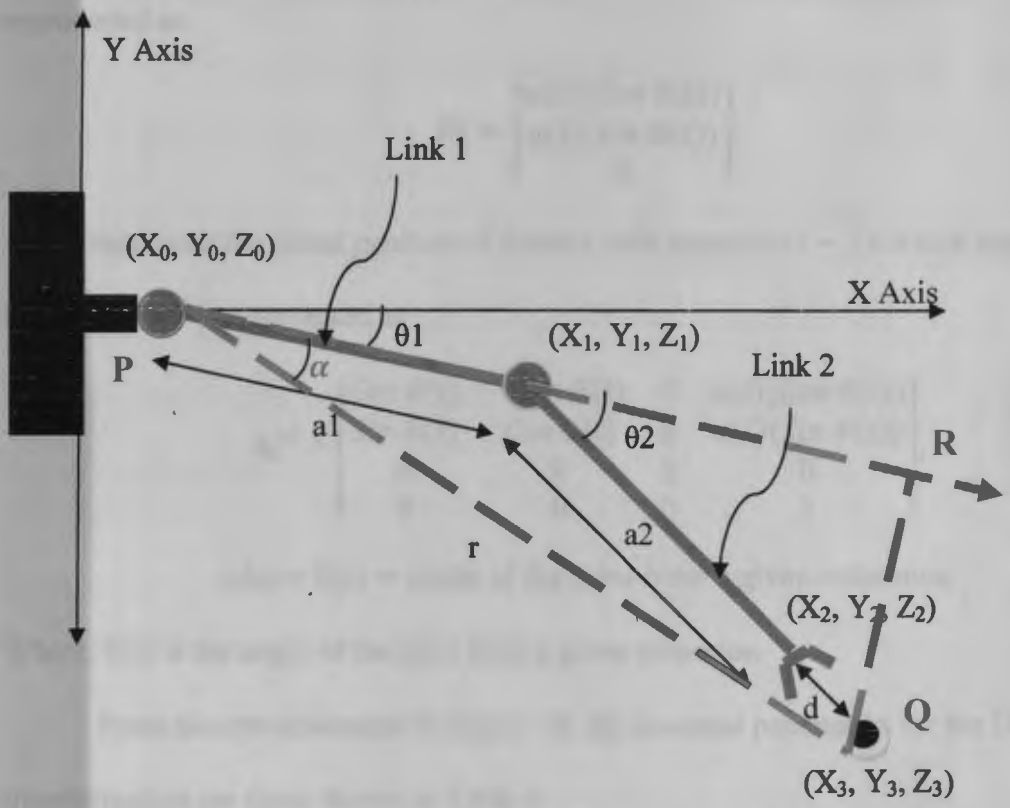


Figure 16 Top view of the robotic arm schematic

5.1. Direct Kinematics

Direct kinematics is the process of solving for the position of the end effector for given link lengths and joint angles. Using the Denavit-Hartenberg (D-H) method [13], the

transformation of joint coordinates to base (Cartesian) coordinates for each link can be represented by a homogeneous transformation matrix of the form

$$\mathbf{A}_i^{i-1} = \begin{bmatrix} \boldsymbol{Ri} & \boldsymbol{Pi} \\ \mathbf{0} & 1 \end{bmatrix}.$$

Here, i is a coordinate frame (X_i, Y_i, Z_i) for a link, $\boldsymbol{Ri}$ is the rotational matrix represented as

$$\boldsymbol{Ri} = \begin{bmatrix} Cos\,\theta(i) & -Sin\,\theta(i) & 0 \\ Sin\,\theta(i) & Cos\,\theta(i) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

$\boldsymbol{Ri}$ represents the orientation of frame $i-1$ in terms of frame i. $\boldsymbol{Pi}$ is a translation vector represented as

$$\boldsymbol{Pi} = \begin{bmatrix} a(i)(Cos\,\theta(i)) \\ a(i)(Sin\,\theta(i)) \\ 0 \end{bmatrix},$$

which represents the initial position of frame i with respect to $i-1$ for link length $a(i)$.

In general $\mathbf{A}_i^{i-1}$ is represented as

$$\mathbf{A}_i^{i-1} = \begin{bmatrix} Cos\,\theta(i) & -Sin\,\theta(i) & 0 & a(i)(Cos\,\theta(i)) \\ Sin\,\theta(i) & Cos\,\theta(i) & 0 & a(i)(Sin\,\theta(i)) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where $\theta(i)$ = angle of the joint from a given reference.

Where, $\theta(i)$ is the angle of the joint from a given reference.

From the arm schematic in Figure 16, the essential parameters for the D-H transformation are those shown in Table 4.

Table 4 D-H coordinate frame

Link	Link Length $a(i)$	Joint Angle $\theta(i)$
L1	a1	θ1
L2	a2	θ2

Therefore,

$$A^0_1 = \begin{bmatrix} Cos\,\theta1 & -Sin\,\theta1 & 0 & (a1)(Cos\,\theta1) \\ Sin\,\theta1 & Cos\,\theta1 & 0 & (a1)(Sin\,\theta1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$A^1_2 = \begin{bmatrix} Cos\,\theta2 & -Sin\,\theta2 & 0 & (a2)(Cos\,\theta2) \\ Sin\,\theta2 & Cos\,\theta2 & 0 & (a2)(Sin\,\theta2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The transformation matrix T represents the position and orientation of (X_2, Y_2, Z_2) with respect to (X_0, Y_0, Z_0) and is the product of the A matrices of each joint [13].

$$T = A^0_1 . A^1_2 = A^0_2$$

$$= \begin{bmatrix} Cos\,(\theta1+\theta2) & -Sin\,(\theta1+\theta2) & 0 & (a2)\,Cos(\theta1+\theta2) + (a1)(Cos\,\theta1) \\ Sin\,(\theta1+\theta2) & Cos\,(\theta1+\theta2) & 0 & (a2)Sin(\theta1+\theta2) + (a1)(Sin\,\theta1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

From this equation, the translation vector provides the position (X_2, Y_2, Z_2) given in base frame coordinates as

$$P_{2(X2,\,Y2,\,Z2)} = \begin{bmatrix} (a2)Cos\,(\theta1+\theta2) + (a1)Cos\,\theta1 \\ (a2)Sin\,(\theta1+\theta2) + (a1)Sin\,\theta1 \\ 0 \end{bmatrix}. \quad (5.0)$$

5.2. Intermediate Calculations

The location of the target object can be described as O(X3, Y3, Z3), coordinates of which can be determined using information about the position of the arm and the distance d_i to the object. Using straight line properties, the distance between the arm end point (X_2, Y_2, Z_2) and object location (X_3, Y_3, Z_3) is given as

$$d = \sqrt{((X3 - X2)^2 + (Y3 - Y2)^2 + (Z3 - Z2)^2)}. \quad (5.1)$$

Since the arm is located at a fixed height and only moves in the x-y plane, the z-coordinates can be ignored. The slope of the line formed by claw and the target object with respect to X-axis can be given as

$$m = \frac{(Y3-Y2)}{(X3-X2)} \equiv \frac{(Y2-Y1)}{(X2-X1)}. \quad (5.2)$$

Using Equations 5.1 and 5.2, the object coordinates are determined to be

$$X3 = X2 + \frac{d}{\sqrt{(m^2+1)}} \quad (5.3)$$

and

$$Y3 = Y2 + m\frac{d}{\sqrt{(m^2+1)}}. \quad (5.4)$$

5.3. Inverse Kinematics

Inverse kinematics [13] is the process of solving for joint angles when the position of the end effector is known. The solutions of such problems are not always unique. From Equations 5.0, 5.3 and 5.4 we know that for the X-Y plane,

$$X_3 = (a2)Cos\,(\theta1 + \theta2) + (a1)(Cos\,\theta1) + \frac{d}{\sqrt{(m^2+1)}}, \quad (5.5)$$

$$Y_3 = (a2)Sin\,(\theta1 + \theta2) + (a1)(Sin\,\theta1) + m\frac{d}{\sqrt{(m^2+1)}}. \quad (5.6)$$

Referring to Figure 16, $a1^2 + a2^2 > r^2 \equiv X3^2 + Y3^2$ implies that there are two solutions. Since the arm has a limited range of motion in each direction, only one solution is possible. Using equation 5.5 and 5.6 for $\theta2$ we get

$$r^2 = X3^2 + Y3^2$$

$$Cos\,\theta2 = \frac{r^2 - a1^2 - a2^2}{2.a1.a2},$$

and

$Sin\,\theta 2 = \pm\sqrt{(1 - Cos^2\theta 2)}$.

Taking the arctangent of $\theta 2$, the angle $\theta 2$ can be calculated,

$$\boxed{\theta 2 = arctan2\left(\frac{\sqrt{\left(1-\frac{r^2-a1^2-a2^2}{2.a1.a2}\right)}}{\frac{r^2-a1^2-a2^2}{2.a1.a2}}\right)}$$

Referring to ΔPQR (dotted line) in Figure 16 and calculating the angle α,

$$\text{Tan}\,\alpha = \frac{QR}{PR} = \frac{a2\ \text{Cos}\ \theta 2}{a1+a2\ \text{Sin}\ \theta 2},$$

$$\text{therefore, } \alpha = \text{arctan2}\left(\frac{a2\ \text{Cos}\ \theta 2}{a1+a2\ \text{Sin}\ \theta 2}\right). \qquad (5.7)$$

Now from Figure 16, we know that,

$$\frac{Y3}{X3} = \text{Tan}\,(\alpha+\theta 1).$$

Taking the arctangent of both sides,

$$\text{arctan2}\left(\frac{Y3}{X3}\right) - \alpha = \theta 1. \qquad (5.8)$$

Thus, from equation 5.7 and 5.8, the angle $\theta 1$ can be calculated as,

$$\boxed{\theta 1 = \text{arctan2}\left(\frac{Y3}{X3}\right) - \text{arctan2}\left(\frac{a2\ \text{Cos}\ \theta 2}{a1+a2\ \text{Sin}\ \theta 2}\right).}$$

6. LABVIEW PROGRAM AND FLOW CHART

6.1. Main VI

The main Virtual Instrument (VI) consists of a front panel and a block diagram. The front panel is a user interface that is used to control the arm motion, select between auto and manual mode and to read sensor data in real time. Individual sliders for the shoulder, elbow and claw are provided to control the motion of the arm manually. The drop down list box is provided to select between manual, auto-manual and object ahead modes. A switch is present to select between auto and manual configuration while in the auto-manual mode. Switches are also provided to configure the arm during the auto-manual mode to hold or release the object after retrieval and to select between the hard or soft target objects. The block diagram consists of a sequence of functions and sub-VI's that run in a continuous loop. The sequences are shown in Appendix B. Figure 17 shows a screenshot of the front panel of the main VI.

Figure 17 Front panel view

6.2. Flow Chart

Figure 18 and Figure 19 shows the flow chart of the main program. This flow chart breaks down the algorithm into steps to achieve the objective based on the user input and feedback from the arm manipulator sensors.

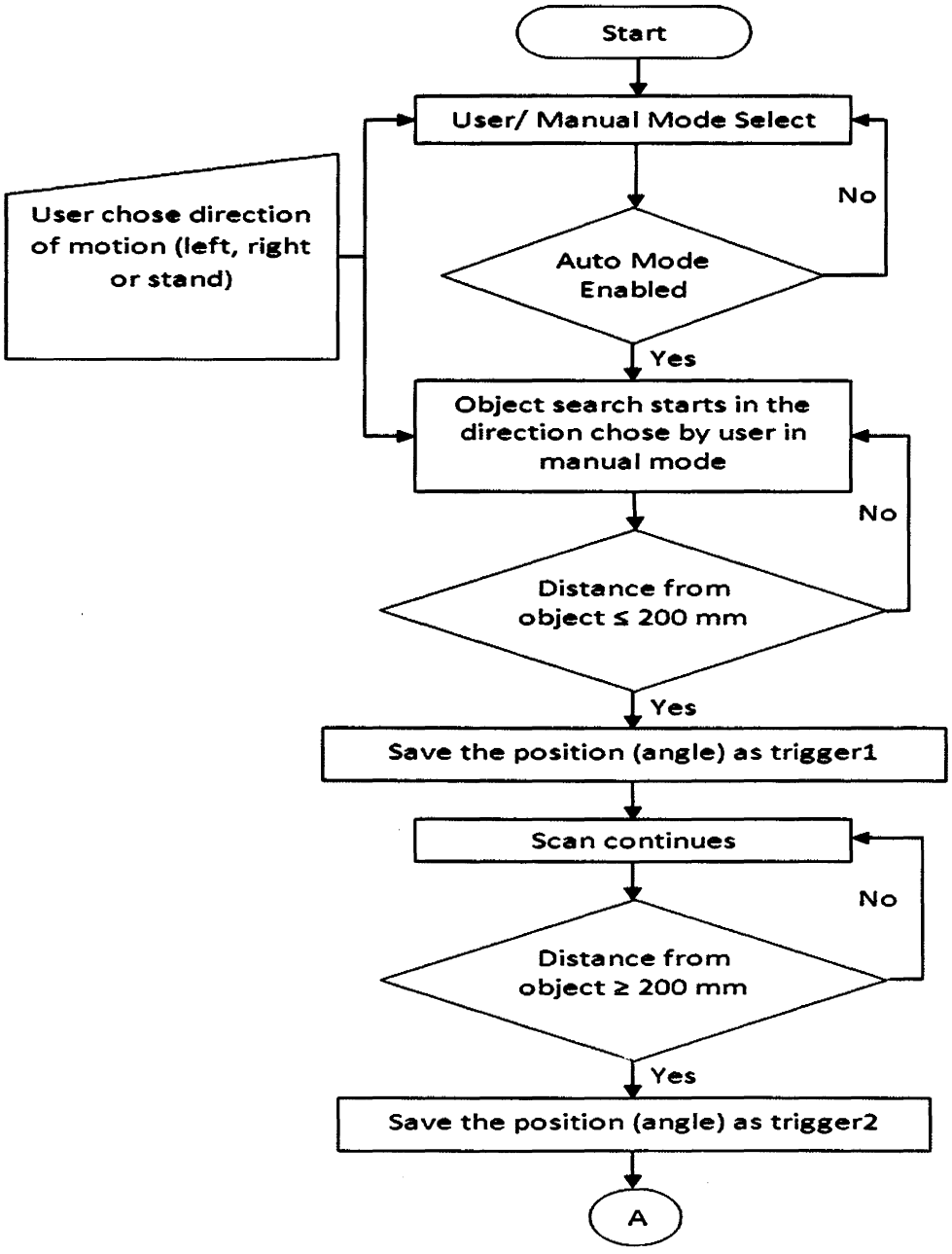


Figure 18 Flow chart part 1

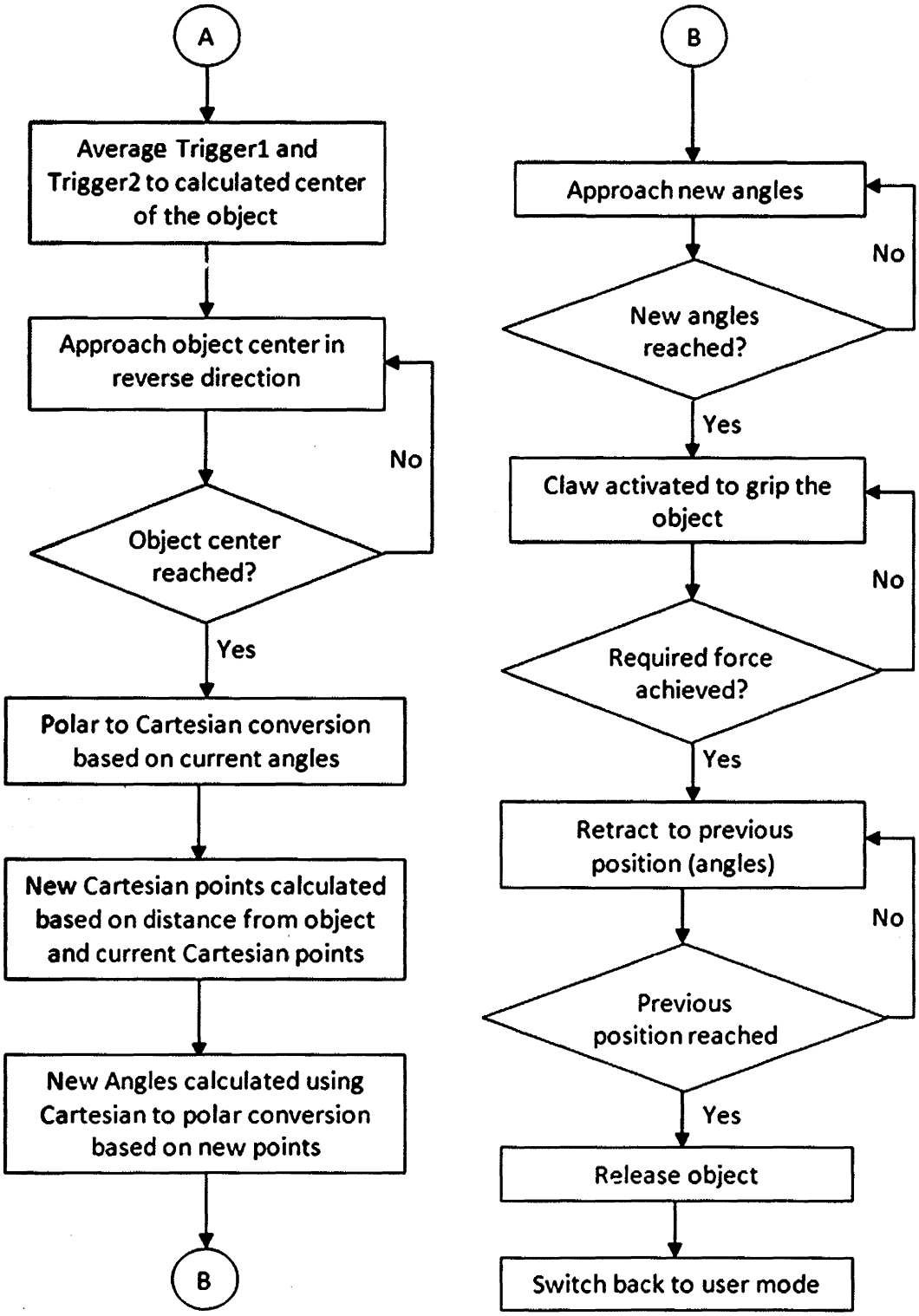


Figure 19 Flow chart part 2

6.3. Sub-VI's

Sub-VI is a subroutine that is called by the main program. Subroutines help to reduce the redundancy of the code and simplify the complex task into small steps.

6.3.1. Motor output

Figure 20 shows the elbow motor output sub-vi which is the same for the shoulder and claw motors.

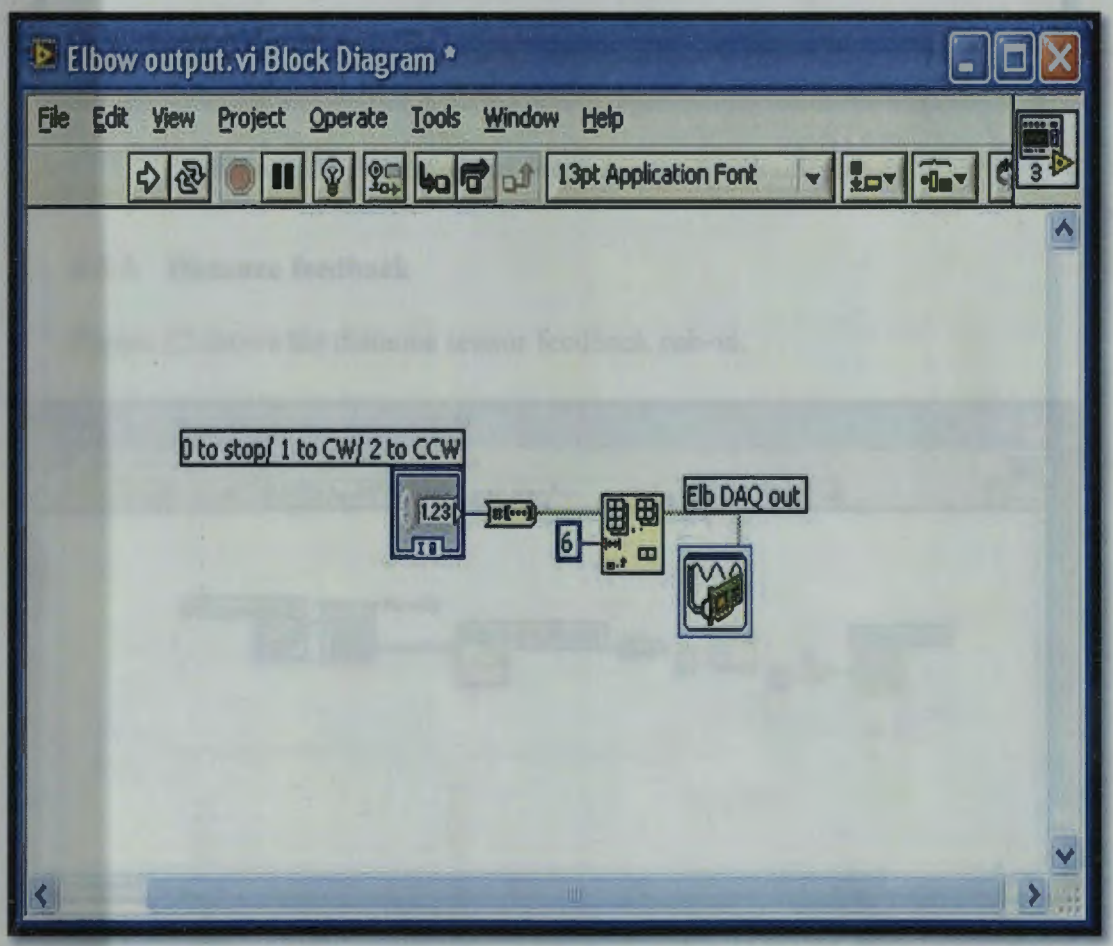


Figure 20 Motor output sub-vi

6.3.2. Position and force feedback

Figure 21 shows the sub-vi for position and force sensor feedback.

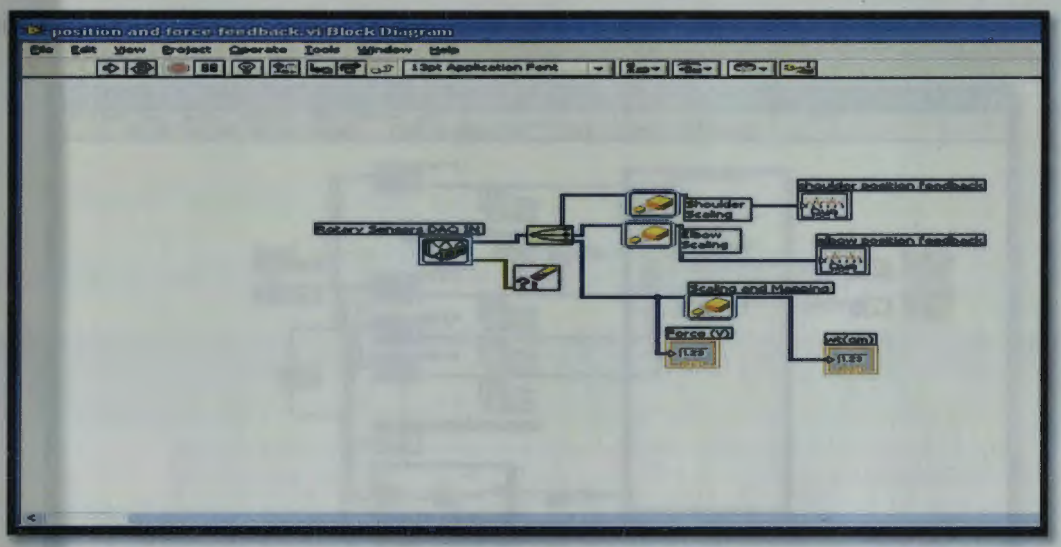


Figure 21 Position and force sensor feedback sub-vi

6.3.3. Distance feedback

Figure 22 shows the distance sensor feedback sub-vi.

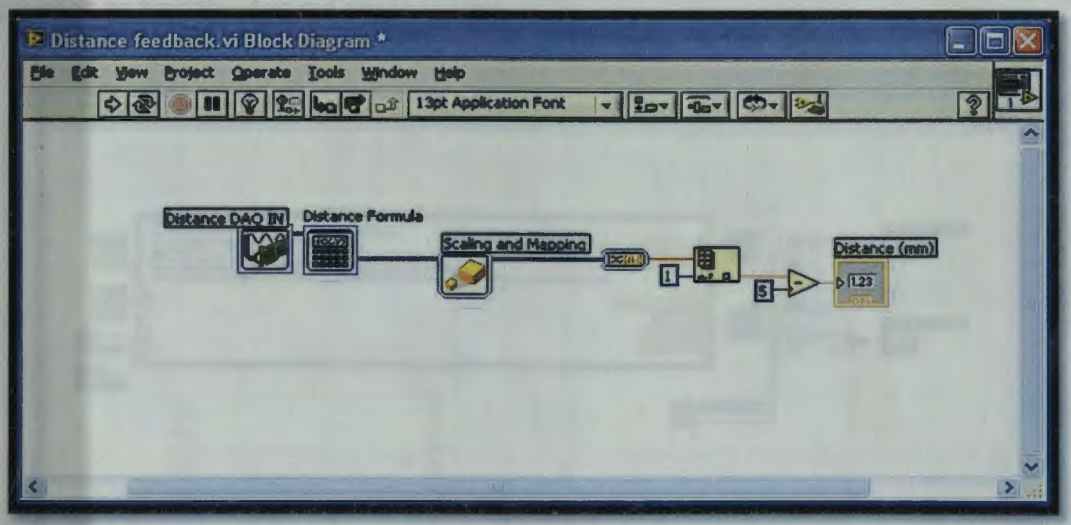


Figure 22 Distance sensor feedback sub-vi

6.3.4. Direct-inverse kinematics algorithm

Figure 23 shows the sub-vi of the forward algorithm that calculates the Cartesian coordinates of the arm location based on polar values of the joint positions.

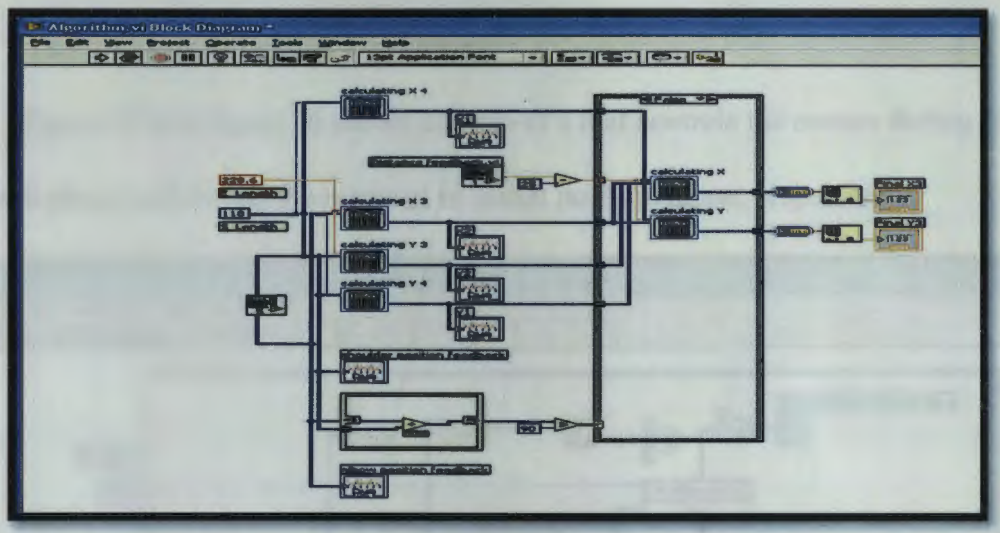


Figure 23 Direct algorithm sub-vi

Figure 24 shows the inverse algorithm sub-vi which performs the Cartesian to polar conversion to find the desired angle values.

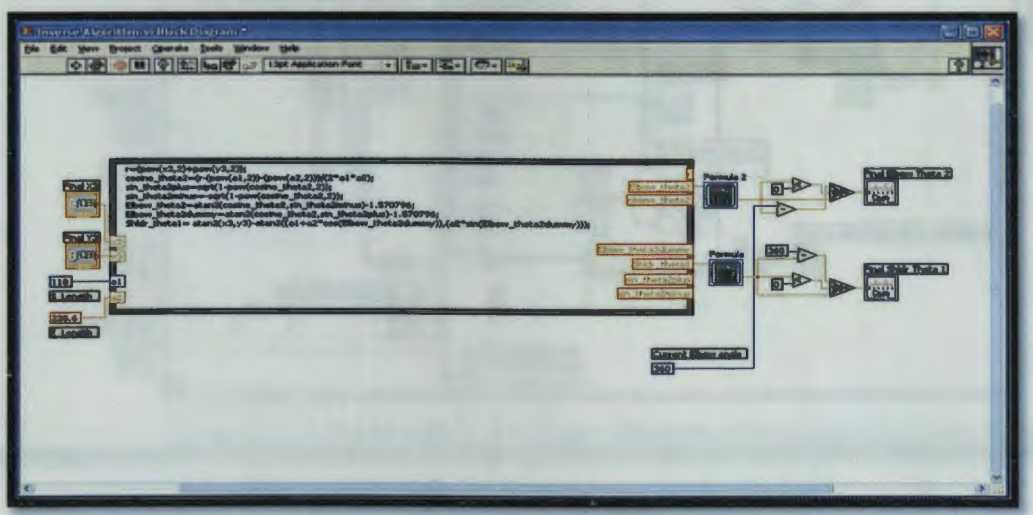


Figure 24 Inverse algorithm sub-vi

Direct algorithm also performs the intermediate calculations. Angles obtained as a result of inverse algorithm are then inserted into the angle approach algorithm which helps the arm manipulator reach the target position.

6.3.5. Angle approach algorithm

Figure 25 and figure 26 shows the sub-vi's that controls the motors during the target approach phase and during the retrieval to initial position phase, respectively.

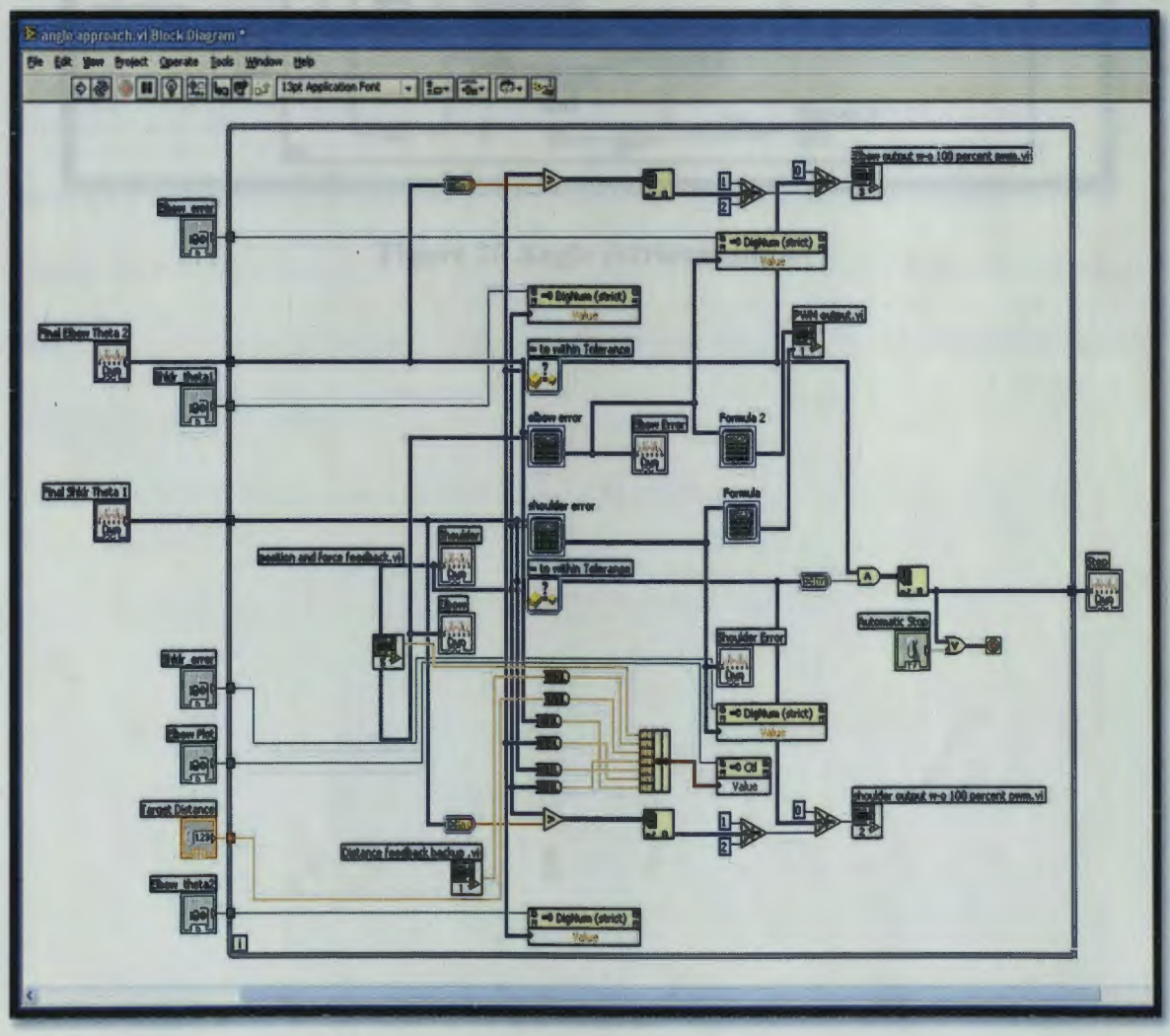


Figure 25 Angle approach sub-vi

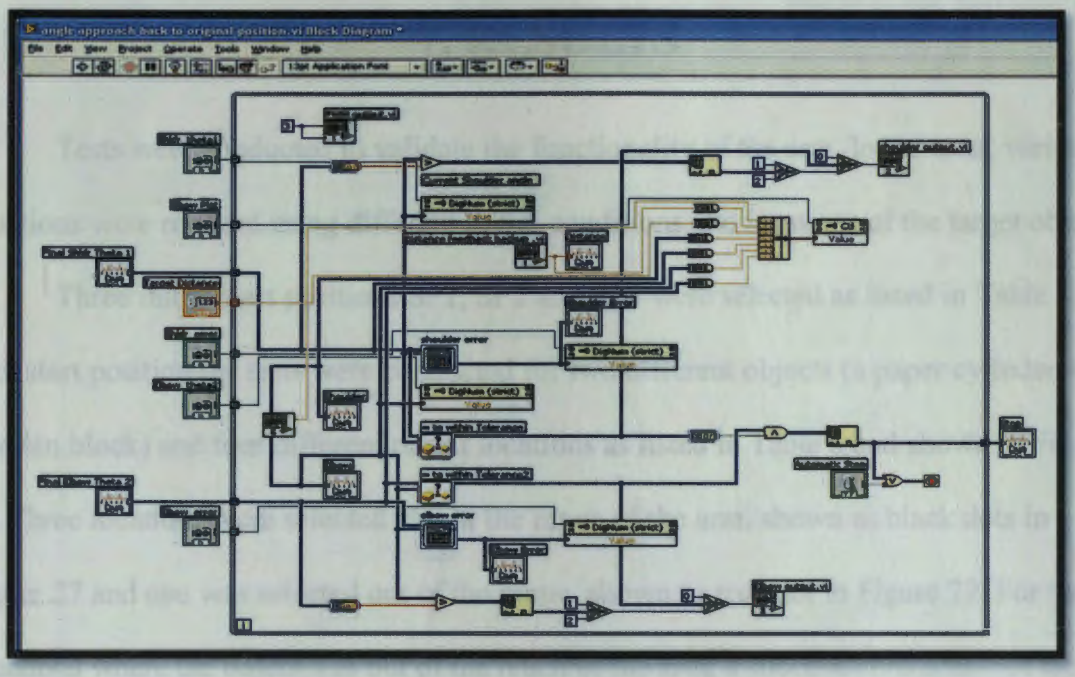


Figure 26 Angle retrieval sub-vi

7. RESULTS

Tests were conducted to validate the functionality of the arm. In the tests, various situations were realized using different initial conditions and locations of the target object.

Three initial start positions SP1, SP2 and SP3 were selected as listed in Table 5. For each start position the tests were conducted for two different objects (a paper cylinder and a wooden block) and four different object locations as listed in Table 6 and shown in Figure 27. Three locations were selected within the range of the arm, shown as black dots in Figure 27 and one was selected out of the range, shown as reds dot in Figure 27. For the situations where the object was out of the reach of the arm, a successful rejection of the object was considered as a pass. Figure 28 shows the results in a real time sensor data plot in Labview. It shows in real time how target values of the position and distance sensors are meet at the end of the plot.

Table 5 Start positions

	Shoulder position	**Elbow position**
SP1	0	0
SP2	15	30
SP3	0	130

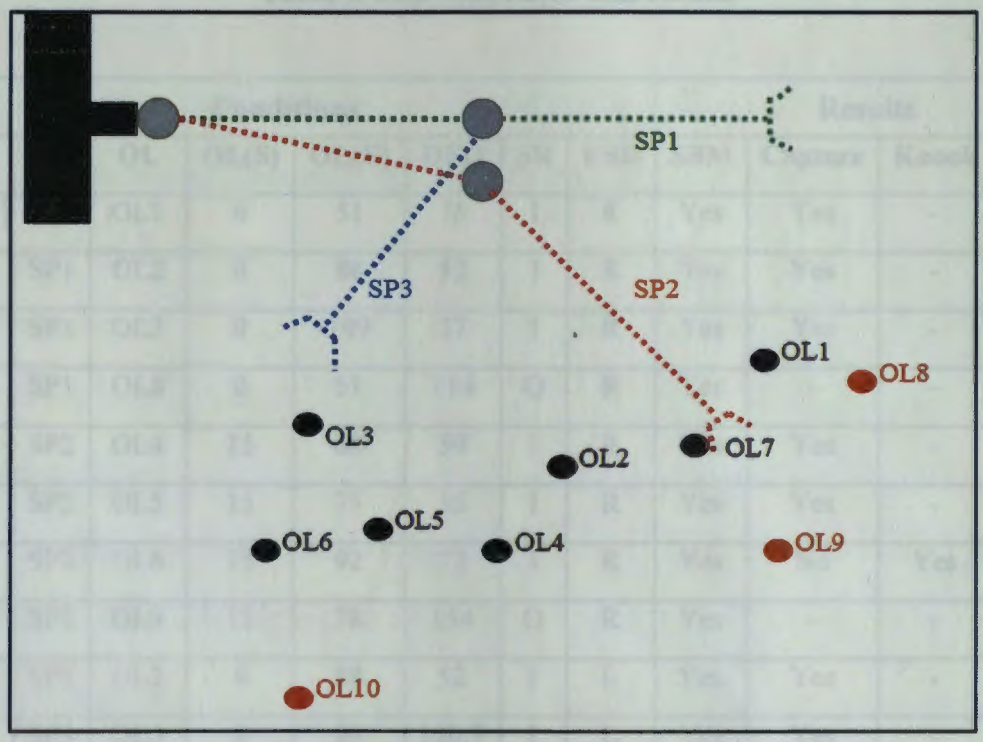


Figure 27 Target object placement

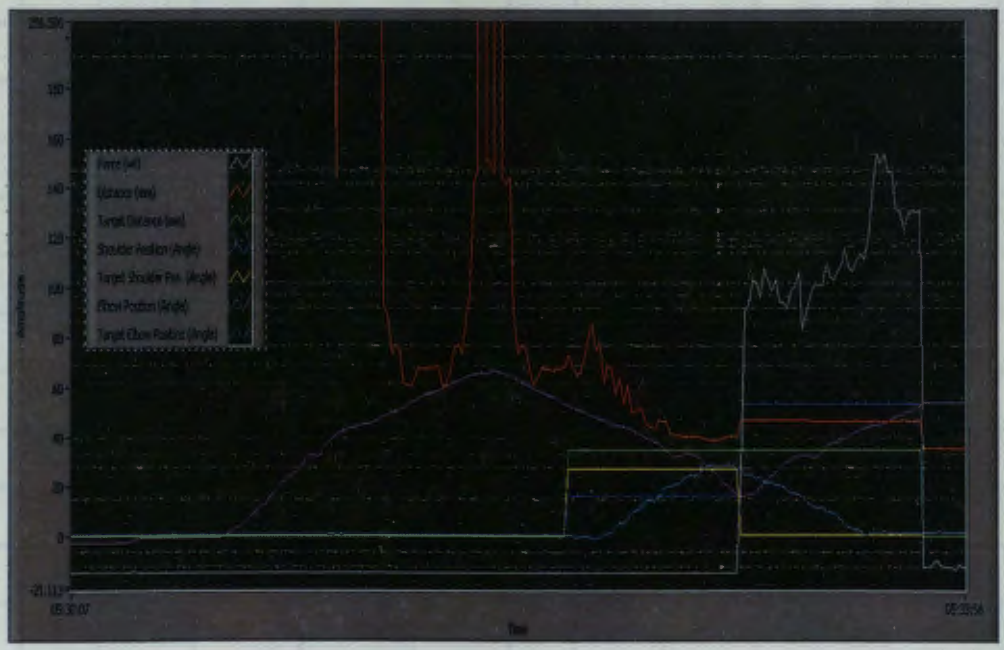


Figure 28 Real time sensor data plot

Table 6 Test conditions and results

Conditions								Results			
Obj.	**SP**	**OL**	**OL(S)**	**OL(E)**	**DFO**	**SR**	**USD**	**SBM**	**Capture**	**Knock**	**ORS**
PC	SP1	OL1	0	51	75	I	R	Yes	Yes	-	-
PC	SP1	OL2	0	88	52	I	R	Yes	Yes	-	-
PC	SP1	OL3	0	109	57	I	R	Yes	Yes	-	-
PC	SP1	OL8	0	51	114	O	R	Yes	-	-	Yes
PC	SP2	OL4	15	60	59	I	R	Yes	Yes	-	-
PC	SP2	OL5	15	75	55	I	R	Yes	Yes	-	-
PC	SP2	OL6	15	92	72	I	R	Yes	No	Yes	-
PC	SP2	OL9	15	78	154	O	R	Yes	-	-	Yes
PC	SP3	OL2	0	88	52	I	L	Yes	Yes	-	-
PC	SP3	OL5	0	99	100.7	I	L	Yes	Yes	-	-
PC	SP3	OL7	0	60	85	I	L	Yes	Yes	-	-
PC	SP3	OL10	0	103	189	O	L	Yes	-	-	Yes
WB	SP1	OL1	0	51	75	I	R	Yes	Yes	-	-
WB	SP1	OL2	0	88	52	I	R	Yes	Yes	-	-
WB	SP1	OL3	0	109	57	I	R	Yes	Yes	-	-
WB	SP1	OL8	0	51	114	O	R	Yes	-	-	Yes
WB	SP2	OL4	15	60	59	I	R	Yes	No	Yes	-
WB	SP2	OL5	15	75	55	I	R	Yes	Yes	-	-
WB	SP2	OL6	15	92	72	I	R	Yes	Yes	-	-
WB	SP2	OL9	15	78	154	O	R	Yes	-	-	Yes
WB	SP3	Ol2	0	88	52	I	L	Yes	Yes	-	-
WB	SP3	OL5	0	99	100.7	I	L	Yes	Yes	-	-
WB	SP3	OL7	0	60	85	I	L	Yes	Yes	-	-
WB	SP3	OL10	0	103	189	O	L	Yes	-	-	Yes

DFO – Distance from object

OL- Object location

OL (S) - Object location with respect to shoulder position

OL (E) - Object location with respect to elbow position

ORS- Out of range success

PC- Paper cylinder

SBM- Switch back to manual

SP – Start position

SR- Set range (In Range (I)/Out of Range (O))

USD- User set direction (Right (R)/Left (L))

WB – Wooden block

7.1. **Result Summary**

Results indicate that the arm was able to successfully switch back from auto-mode to user-control-mode 24 out of 24 times. The arm was able to capture the target object 16 out of 18 times when the object was within range. For the cases where the object was not captured, it was knocked over successfully. The arm successfully avoided the target object 6 out of 6 times when it was out of the range.

8. CONCLUSION

In this work, an ultrasonic sensor, force sensor and two position sensors were used to create an autonomous arm manipulator with three degrees of freedom. Using feedback from the ultrasound sensor, target objects of different shape and size were detected and distance of the end effector from the target object successfully calculated. Algorithms and programs were created to operate and control the arm manipulator to detect, reach and retrieve the target object. Three modes of operation were provided which include 1) manual mode 2) auto-manual mode and 3) object in front mode. The process of a smooth transition from user control to auto arm control was implemented. All the sensors and actuators were successfully integrated to create a smart autonomous robotic arm. This system was successfully tested under different configurations and situations.

In future work related to smart robotic arm control, the following upgrades and features can be integrated. A short range RFID receiver can be placed on the claw of the arm to detect objects with RFID tags. After identification of the object using the attached RFID tag, the arm can be programmed to retrieve the object and then go to a predefined position autonomously [14]. For example, daily use items like a toothbrush, a cell phone or a glass of water can be tagged with an RFID chip. Once the user has a hold of the object using the smart robotic arm, the arm will use RFID to determine which item it is holding. Based on a lookup table for the end position related to that item, the arm can either move close to the user's face in the case of a toothbrush or glass of water or it can reach to the user's ear in the case of a cell phone [15]. Similarly, object identification can be achieved

using a color sensor where a lookup table for each color can be created to correspond to a known end position of the arm.

In the past few years, extensive research has been done on brain-computer interface systems. So far, control of two-dimensional movement has been successfully carried out in the form of matrix-style typing and playing two-dimensional ping pong The arm manipulator presented here can provide a platform for advanced research on three-dimensional robot control using non-invasive brain-computer interface techniques [16].

The current system runs the smart robotic arm using a desktop version of the Labview software. A handheld version of Labview for portable devices such as PDAs can be used thus allowing one to use this arm in ambulatory situations. The claw width can be increased to accommodate a bigger object. The current claw width is a limiting factor in terms of the size of the object that can be grabbed by this manipulator. More degrees of freedom can be added to obtain a better realization of a human limb. Instead of a single ultrasonic sensor, an array of ultrasonic sensors can be used [11]. This will help increase the area of the scanned region as well as its sensitivity. Object collision avoidance [17] can also be implemented using a sensor array.

REFERENCES

1) S. Niku, "Introduction to robotics: analysis, systems, applications," Prentice Hall, 1st edition, 2001.

2) E. Garcia, M. Jimenez, P. Santos and M. Armada, "The Evolution of Robotics Research from Industrial Robotics to Field and Service Robotics," in *Proceedings of IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 99-103, 2007.

3) G. Gupta, S. Mukhopadhyay, C. Messom and S. Demidenko, "Master-Slave Control Of a Teleoperated Anthromorphic Robotic Arm With Gripping Force Sensing," in *Proceedings of IEEE Instrumentation and Measurement,* vol. 55, no. 6, pp. 2136-2145, 2006.

4) P. Abolmaesumi, S. Salcudean, W. Zhu, M. Sirouspour and S. Dimaio, "Image-Guided Control of a Robot for Medical Ultrasound," in *Proceedings of IEEE Robotics and Automation*, vol. 18, no. 1, pp. 11-23, 2002.

5) W. Book, "Recursive Lagrangian Dynamics of Flexible Manipulator Arms," *in proceedings of International Journal of Robotics Research*, vol. 3, no. 3, pp. 87-101 1984.

6) J. Slotine and W. Li, "On the Adaptive Control of Robot Manipulators," in *Proceedings of International Journal of Robotics Research*, vol. 6, no. 3, pp. 49-59 1987.

7) P. Feng and Y. Wan-sheng, "Design and Application of Fuzzy Logic Controller Based on LabVIEW," *Control and Instruments in Chemical Industry,* vol. 2, 2004.

8) A. Saffiotti, "The uses of fuzzy logic in autonomous robot navigation," *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, vol. 1, pp. 180-197, 1997.

9) S. Ge, T. Lee and C. Harris, "Adaptive Neural Network Control of Robotic Manipulators", *World Scientific Series in Robotics and Intelligent Systems*, vol. 19, 1998.

10) G. Heidemann and M. Schöpfer, "Dynamic Tactile Sensing for Object Identification", *in IEEE Proceeding of International Conference on Robotics & Automation*, vol. 1, pp. 813-818, 2004.

11) G. Hueber, T. Ostermann, T. Bauernfeind, R. Raschhofer and R. Hagelauer, "New Approach of Ultrasonic Distance Measurement Technique in Robot Applications", in *Proceedings of International Conference Signal Processing*, vol. 3, pp. 2066-2069, 2000.

12) E. Dobelin and D. Manik, "Measurement Systems: Application and Design," *McGraw Hill*, 4th edition, 1990.

13) F.L Lewis, D.M Dawson and C.T Abdallah, "Robot Manipulator Control- Theory and Practice," *CRC Press*, 2nd edition, 2004.

14) N. Chong and K. Tanie, "Object Directive Manipulation Through RFID", in *Proc. Int. Conf. on Control, Automation and Systems*, vol. 42, pp. 22-25, 2003.

15) P. Chang and H. Park, "Development of a Robotic Arm for Handicapped People: A Task-Oriented Design Approach", *Autonomous Robots*, vol. 15, no. 1, pp. 81-92, 2003.

16) J. Wolpaw and D. McFarland, "Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans," *Proceedings of the National Academy of Sciences*, vol. 101, no. 51, pp. 17849-17854, 2004.

17) O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," *International Journal of Robotics Research*, vol. 5, no. 1, pp. 90-98, 1986.

APPENDIX

A. Sensor Specifications

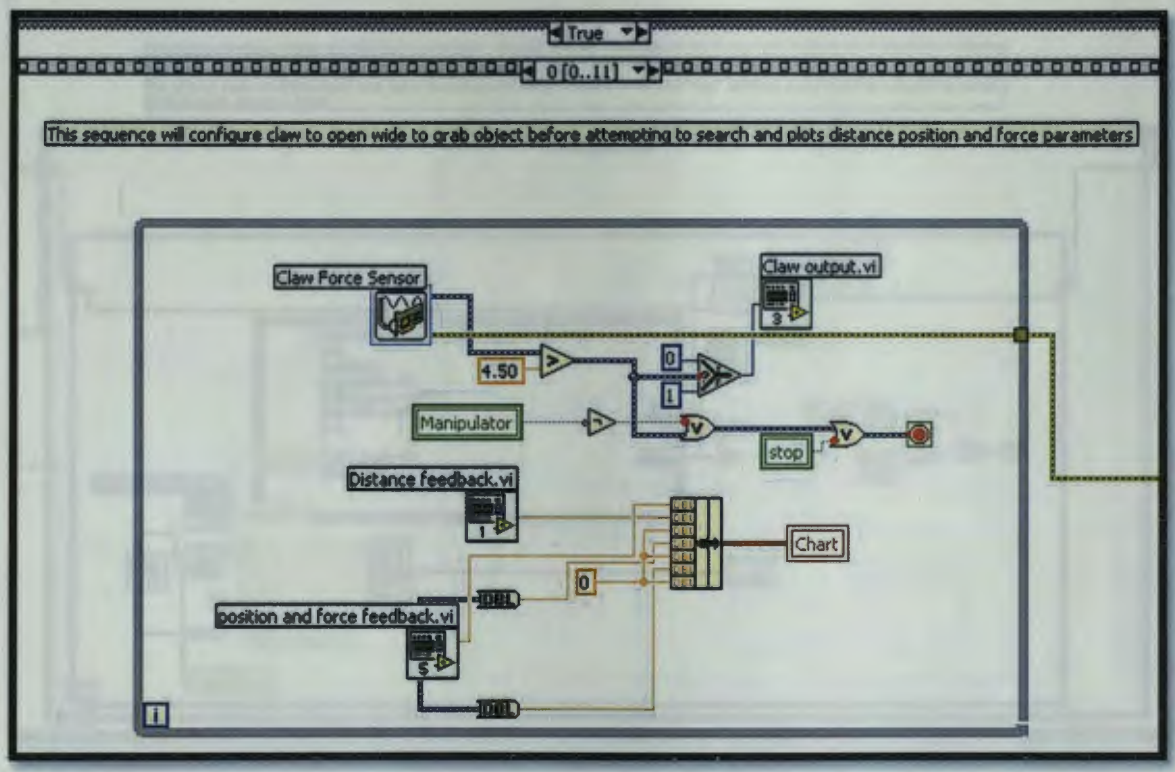
Specifications of the force sensor are as follows:

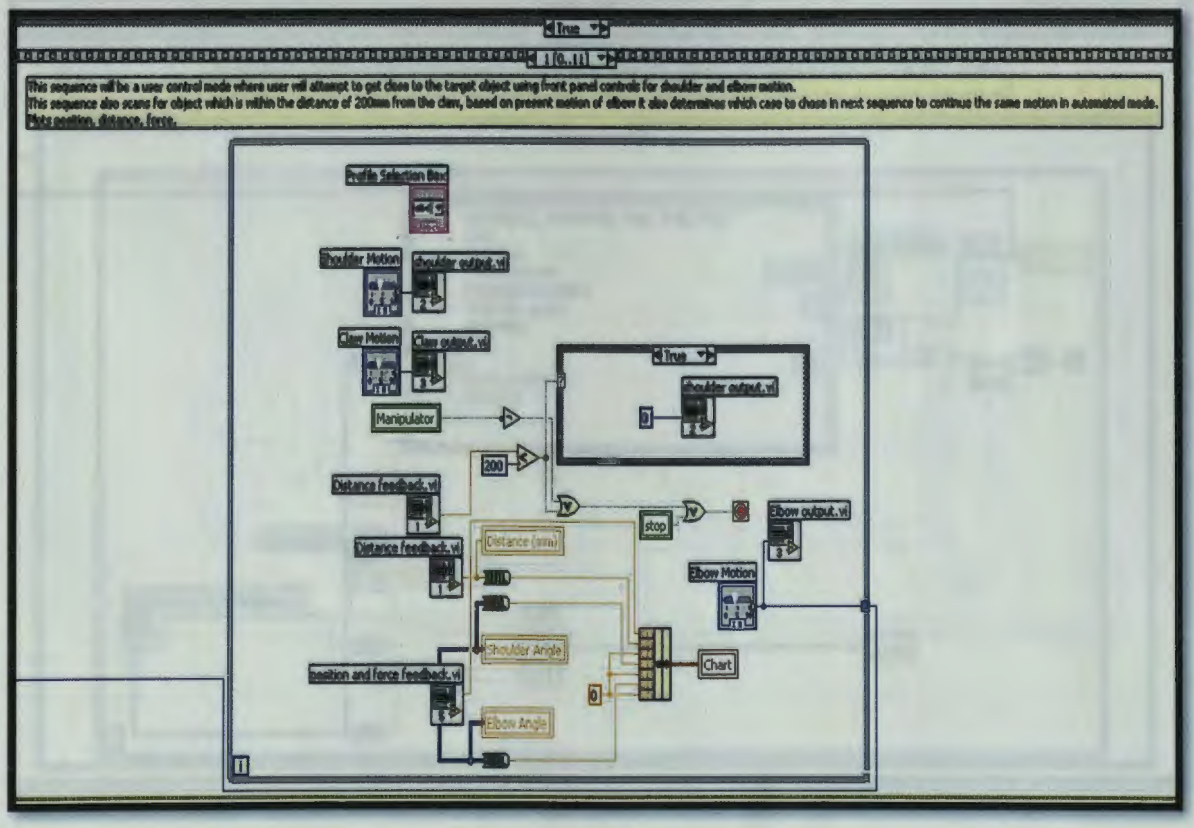
Length	6" (152mm)
Thickness	0.008" (.208mm)
Width	0.55" (14mm)
Sensing Area	0.375" diameter (9.53mm)
Linearity Error	<+/-5%
Repeatability	<+/-2.5% of full scale
Drift	<5% per logarithmic time scale
Response Time	<5 microseconds
Operating Temperatures	15°F to 140°F (-9°C to 60°C)
Force Range	0-1 lb. (4.4 N)
Temperature Sensitivity	0.2% per degree F
Output	Variable Resistance

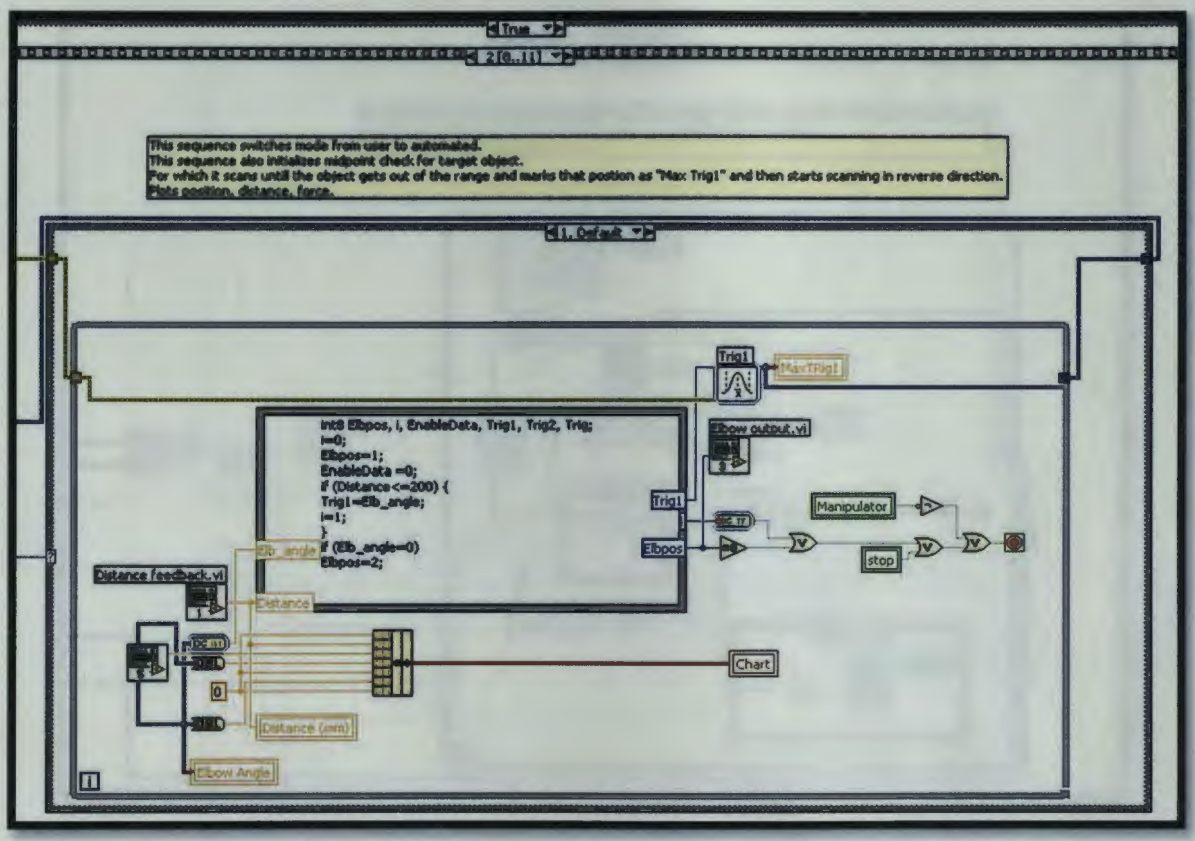
Specifications of distance sensor are as follows:

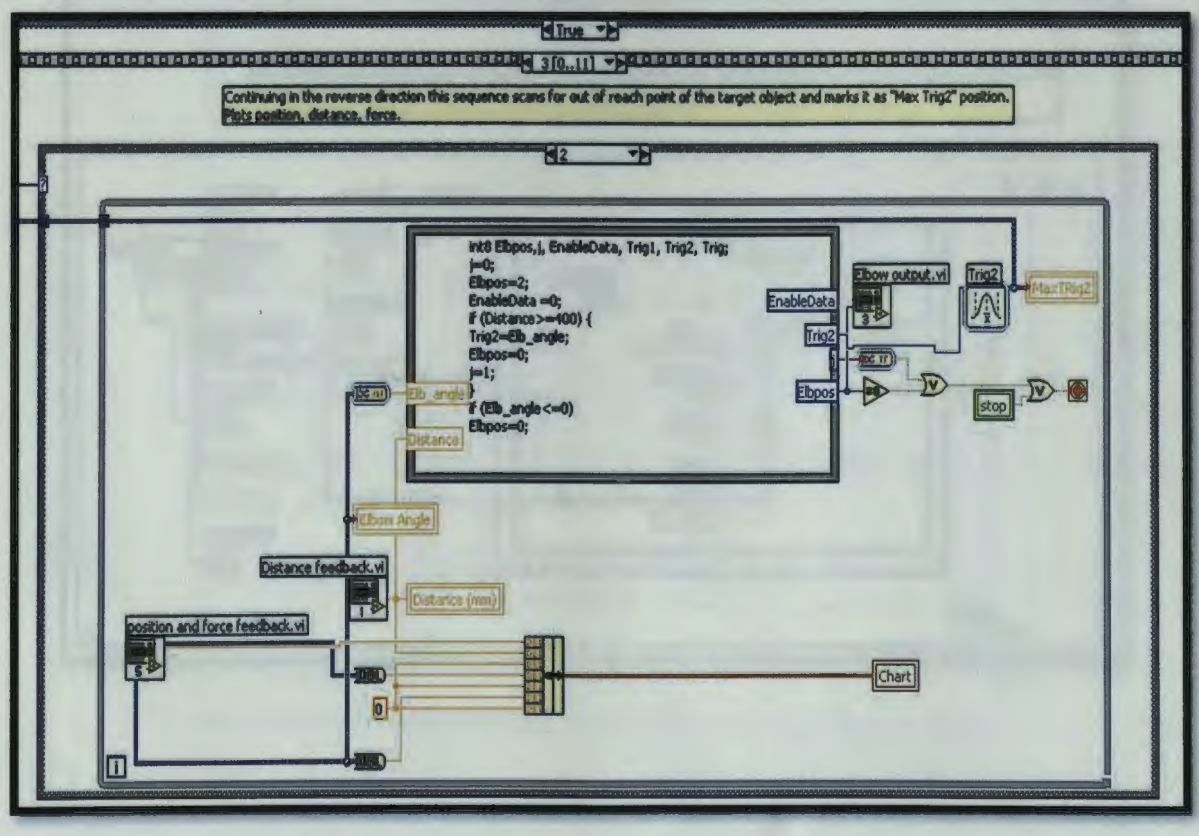
Length	0.84" (21.3mm)
Thickness	0.6" (15.3mm)
Width	1.8" (45.7mm)
Supply Voltage	5VDC
Supply Current	30mA
Range	3 cm to 3 m (1.2 inch to 3.3
Burst Frequency	40 KHz
Burst time	750 us
Excitation pulse	5V TTL

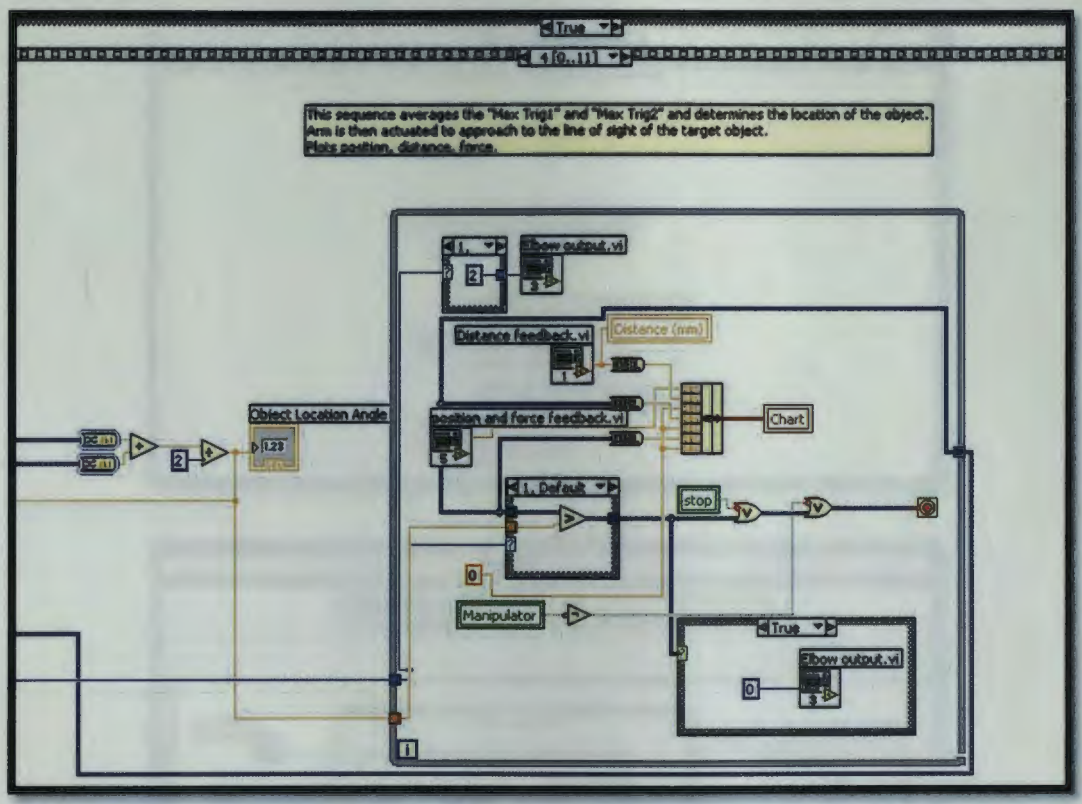
B. Program Screenshots

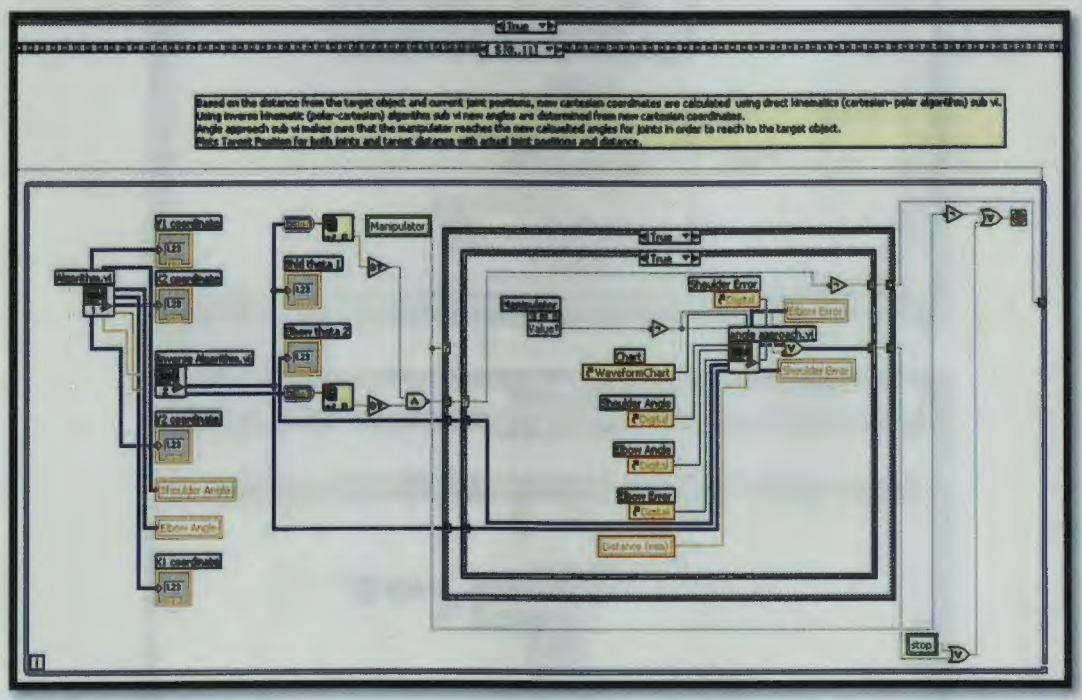


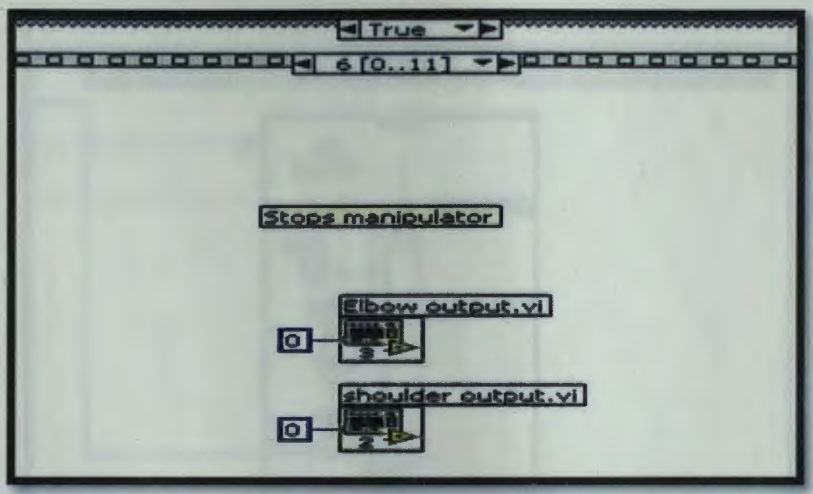


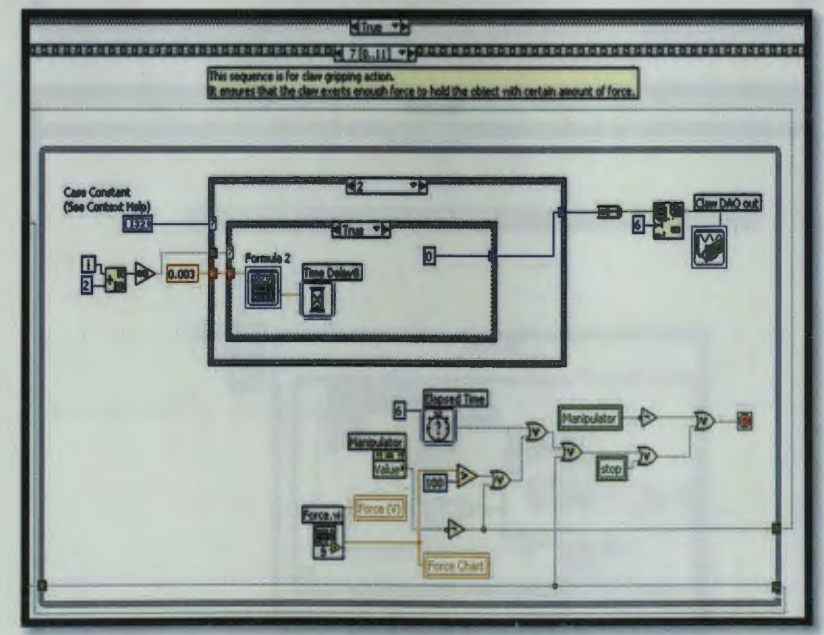

True
2 [0..1]
This sequence switches mode from user to automated.
This sequence also initializes midpoint check for target object.
For which it scans until the object gets out of the range and marks that position as "Max Trig1" and then starts scanning in reverse direction.
Plots position, distance, force.
1, Default
Trig1
MaxTrig1
int8 Elbpos, i, EnableData, Trig1, Trig2, Trig;
i=0;
Elbpos=1;
EnableData =0;
if (Distance<=200) {
Trig1=Elb_angle;
i=1;
}
if (Elb_angle=0)
Elbpos=2;
Elbow output.vi
Manipulator
stop
Elb_angle
Elbpos
Distance feedback.vi
Distance
Chart
Distance (mm)
Elbow Angle

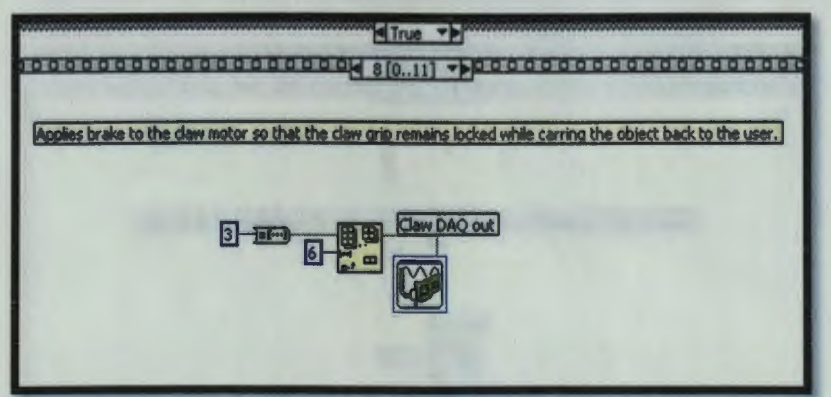

True
3 [0..1]
Continuing in the reverse direction this sequence scans for out of reach point of the target object and marks it as "Max Trig2" position.
Plots position, distance, force.
2
int8 Elbpos,j, EnableData, Trig1, Trig2, Trig;
j=0;
Elbpos=2;
EnableData =0;
if (Distance>=400) {
Trig2=Elb_angle;
Elbpos=0;
j=1;
}
if (Elb_angle<=0)
Elbpos=0;
EnableData
Trig2
MaxTrig2
Elbow output.vi
Elb_angle
Elbpos
stop
Distance
Elbow Angle
Distance feedback.vi
Distance (mm)
position and force feedback.vi
Chart

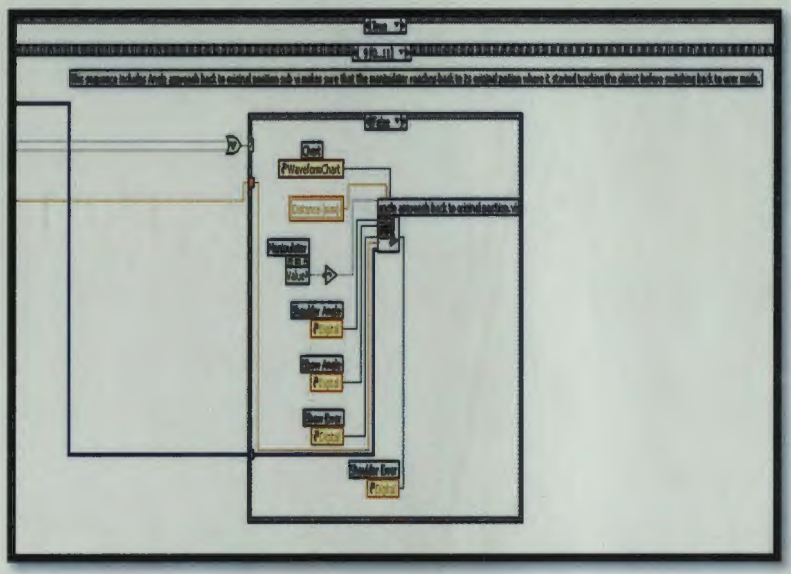

True
This sequence averages the "Max Trig1" and "Max Trig2" and determines the location of the object.
Arm is then actuated to approach to the line of sight of the target object.
Data position, distance, force.
Elbow output.vi
Distance feedback.vi
Distance (mm)
Object Location Angle
Position and force feedback.vi
Chart
stop
Manipulator
True
Elbow output.vi

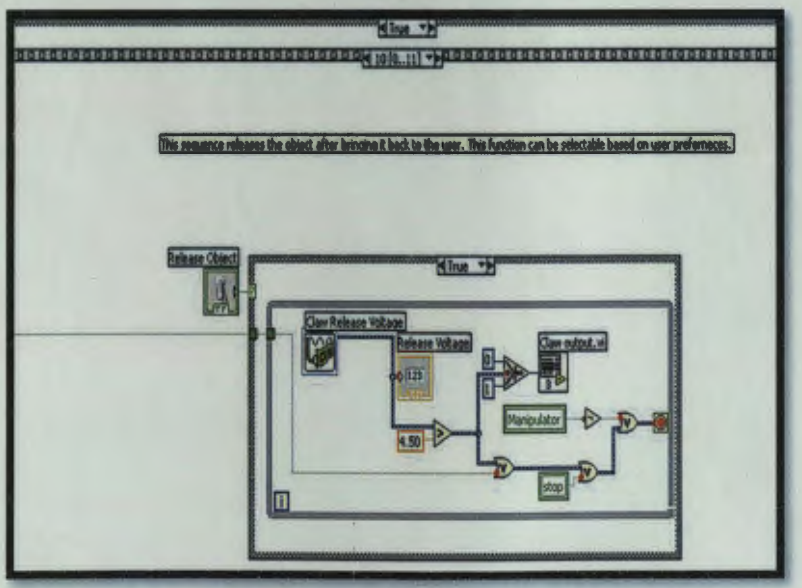

Based on the distance from the target object and current joint positions, new cartesian coordinates are calculated using direct kinematics (cartesian- polar algorithm) sub vi.
Using inverse kinematic (polar-cartesian) algorithm sub vi new angles are determined from new cartesian coordinates.
Angle approach sub vi makes sure that the manipulator reaches the new calculated angles for joints in order to reach to the target object.
Manipulator
Shoulder Error
Elbow Error
Chart
WaveformChart
Shoulder Angle
Elbow Angle
Distance (mm)
stop

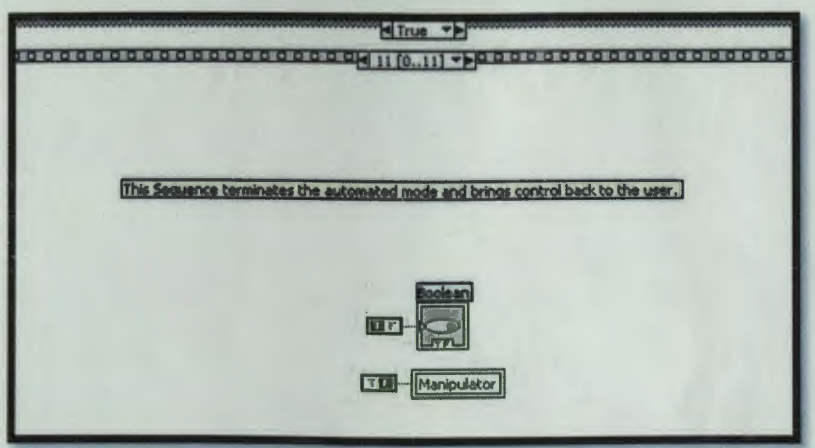

True
6 [0..11]
Stops manipulator
Elbow output.vi
shoulder output.vi


True
7 [0..11]
This sequence is for claw gripping action.
It ensures that the claw exerts enough force to hold the object with certain amount of force.
Case Constant
(See Context Help)
2
True
Formula 2
Time Delay
0.003
0
Claw DAQ out
Elapsed Time
Manipulator
stop
Force.vi
Force (V)
Force Chart


True
8 [0..11]
Applies brake to the claw motor so that the claw grip remains locked while carring the object back to the user.
3
6
Claw DAQ out




This sequence releases the object after bringing it back to the user. This function can be selectable based on user preferences.
Release Object
True
Claw Release Voltage
Release Voltage
Claw output..vi
1.25
Manipulator
4.50
stop


True
This Sequence terminates the automated mode and brings control back to the user.
Boolean
Manipulator