Spring, 2012 Haptics Class Project Paper presented at
the University of South Florida, June 27, 2012.

Simulation of Mechanoreceptors and Skin for Gentler Robotic Gripping

Louis Melgar, Mechanical Engineering, University of South Florida, Tampa, Florida

Abstract— **This paper describes the hardware and system used in a 6 bar linkage mechanism that employs force sensing resistors and silicone molded strips to simulate skin and Merkel Disk mechanoreceptors. The purpose of this mechanism is to successfully grip objects without crushing them, using only necessary effort for action. Testing was conducted to quantify values of sensitivity that would give effective response when gripping a range of objects. The results revealed a proper value, along with some points of improvement in the system.**

I. Introduction

When gripping objects, it is apparent that one does not aim to crush the object one is gripping. The operational purpose is to safely envelop an object, acquire grip perception and prepare for the next sequence-be it lifting, turning or moving an object. Only necessary force to prevent slip is required. Howe [8] mentions that experiments have been conducted for touch sensing function in the performance of manipulation tasks, and concluded that grasp force was always near the minimum to avoid slipping, despite geometrical variations and weight differences. Is it necessary then to design robotic gripping mechanisms to apply maximum allowable force to an object in order to perform a spatial displacement task?

Merkel receptors, which sit in between the dermis and epidermis layers of the skin, serve to continuously fire off and send cutaneous sensing data back to the brain. These disk shaped mechanoreceptors [1] are slow adapting and slow pushing, sensing pressure and fine details. Through the simulation of these receptors, a system can be incorporated into a gripping mechanism, allowing for sensory feedback to the mechanism controller.

Sensory information that is relayed from these receptors to the dorsal root, and then to the thalamus serves as tactile (touch) feedback when gripping, feeling or manipulating an object, and gives the human a sense for what he is gripping, what it feels like and if it will slip off his hands. Grip strength however, as tested in Air Force personnel in a NASA experiment was measured at 134 lbs [2]. Humans have the ability to crush objects, but sensory information allows them to minimize effort and use frictional force to their advantage.

By retrofitting a 6 bar power-screw actuated acrylic gripping mechanism (Fig. 1) with a small array of force sensing resistors (FSR), an analog signal can be generated and interpreted by a controller as a digital signal, much like mechanoreceptors interpret tactile data in the brain. The mechanism has the capability to apply large amounts of torque to an object due to the actuation method (power screw) and linkage system, but with the retrofit it should "sense" when there is a normal/lateral force applied from the object to its links. Upon this signal, sufficient grip force will be assumed, and all gripping operations will be halted, safely securing the object within the mechanism. If slip occurs, or is instigated, there will be a lateral-force drop, and if this is enough to cross below the set bias, the mechanism adjusts slightly, until grip is obtained again (sensor reading is above bias) and stops actuation.

Weiss et al. [9] used five high resolution tactile sensors based on resistive principles arranged in two arrays, and also included a higher resolution array for the palm. Through high miniaturization they were able to fit the entire system within the fingertips. They concluded that the sensitivity of the sensors were high enough to act as "artificial skin."

Silicone strips in the form of ridged tabs will also be retrofitted to the interior of the linkages in order to transmit forces to the sensors accordingly, and to behave as skin for the mechanism. Friction forces from the silicone will aid in grip, skin simulation and averting damage to the object being gripped (Fig. 2).

In the context of this system, an experiment will comprise of gripping objects that fit a certain range of geometry, which will output a specific range of normal force against the linkages of the mechanism, therefore testing if the applied bias is then ideal for safe gripping of that range of object sizes and shapes.

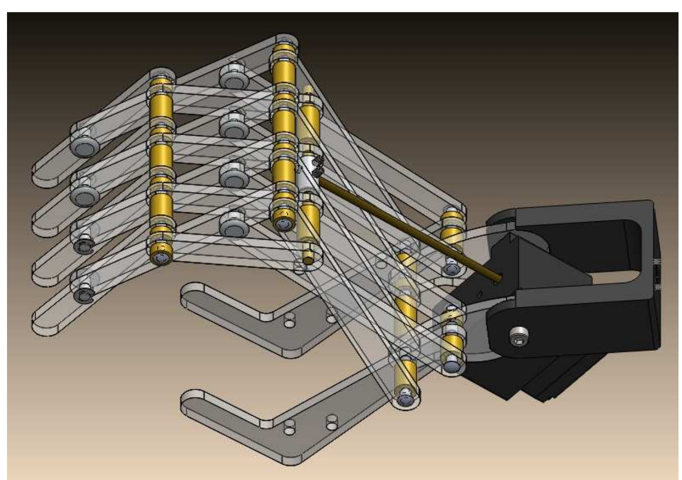

Figure 1: 6-bar linkage gripping mechanism actuated via power-screw

II. Background

Force sensing resistors change resistance with applied lateral force. They are comprised of a flexible substrate layer of semi-conductor material bonded to a layer of printed interdigitating material. When un-actuated they act as resistors of infinite resistance (open circuits), and upon actuation they can range anywhere from 100 kilo-ohms (light pressure) to 250 ohms (high pressure.) [3]

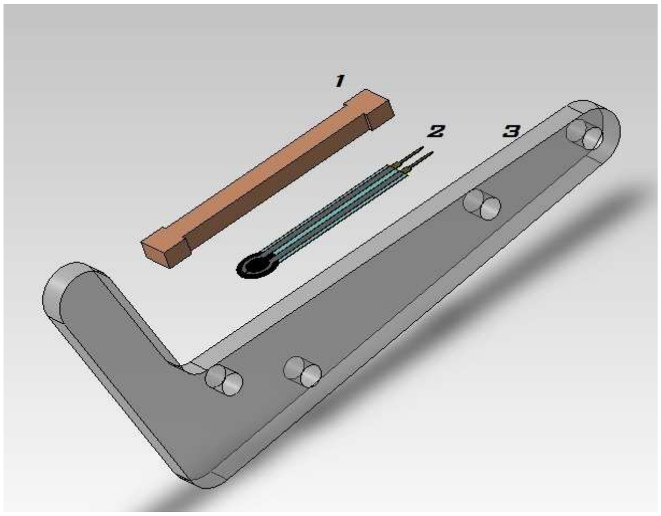

Figure 2: (1) Silicone tab, (2) FSR, (3) Linkage

FSRs have been used before in robotics either in single instances or in arrays for multimodal sensory systems [4] where they must serve to relay data regarding shape and orientation. They are not known to have much resolution and are known to vary about 10% from the specified manufacturer's range, but have proven to be useful in even a simple Haptics experiment in which data received from the FSRs was relayed as touch feedback trough a haptic glove [5].

The FSR sensors used in this system do not aim to accurately measure force and interpret it as "too strong" or "too weak," but rather are used as a means to sense when a certain range of general grip has been obtained. The combination of these and the silicone tabs will work in unison to simulate a simple human process in which a mechanoreceptor sends a signal to an afferent (sensory) neuron, within the somatosensory cortex, and in turn is interpreted and carried by efferent (motor) neurons toward peripheral organs [6].

III. System Design

A. *Linkage System and Actuation*

In terms of kinematics, the mechanism is a 6- bar grounded mechanism designed to close concentrically around objects which are cylindrical in shape. Although the mechanism is most effective for cylinders, it can also hold large rectangular, spherical and odd shaped objects. The radial range is between 100mm for fully open to 25mm for fully closed positions (capability range). Linkages were individually laser cut out of high-impact acrylic sheets, and pinned using brass and aluminum rods. The mechanism was designed to be sturdy, and to support the high lateral loading transmitted to the pin joints when gripping, crushing or holding objects while suspended. Consideration was taken to minimize the aluminum rod diameter to reduce bearing stresses at the linkages, while avoiding pin deformation anywhere. Individual hardened steel pressure pins were used at the outermost pin-joints to increase overall rigidity as well.

The mechanism is actuated through the use of a high torque-low speed DC motor and a direct power-screw drive. Brass was used for the screw and aluminum for the nut, as the nut should, by rule of thumb [7], always be made of a softer material. The drive system is designed to be fully independent from the system upon the removal of two large 8mm screws, for situations in which a nut may need replacement, or a safety-fuse bushing is destroyed. The system is depicted in Fig. 3 and Fig. 4

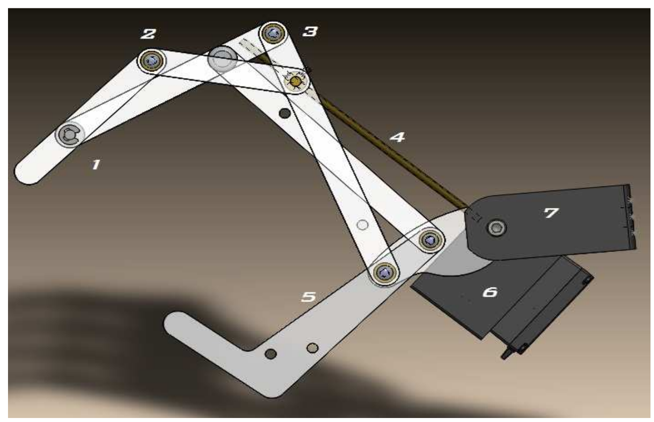

Figure 3: Linkage assembly:
(1) Hardened steel pins, (2) Aluminum pin joints, (3) Aluminum drive nut, (4) Brass power screw, (5) Ground link, (6) DC motor and case containing Safety-fuse, (7) U-bracket and drive tie-down screws

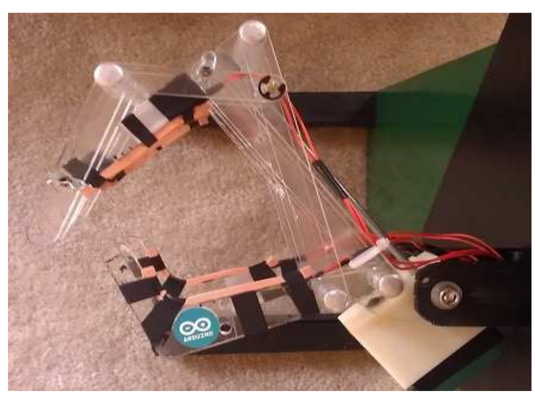

Figure 4: Full assembly including FSR sensors and silicone tabs

B. *Electrical System and Control Logic*

The motor is powered by a 9-volt power source and controlled with a 3 pin PWM activated motor controller through a microcontroller board, Fig. 5. The FSR sensors are attached to 4 analog input pins (1 pin each), and a single output pin acts as the PWM gate for the motor-controller. The system is equipped with manual control and robotic autonomous control. A normally open flip switch starts the actuated control loop, and can be used also as an emergency cutoff in case of sensor failure.

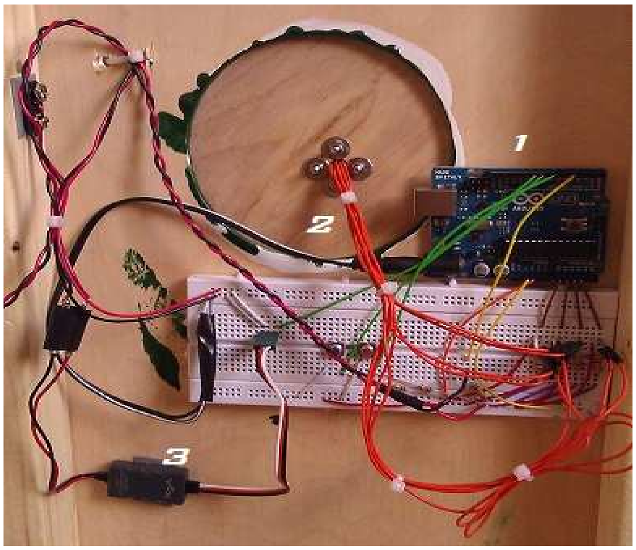

Figure 5: Control System: (1) Micro-controller board, (2) FSR wiring, (3) Motor controller

Given the simple nature of the simulation for the mechanism, the control logic is actually quite simple. In the same manner that our sensory cortex system is always in constant looping, reading interactions with the surrounding environment, the algorithm is constantly sensing for the threshold bias to be met. Upon the robotic control switch being actuated, the motor is started, and the control loop begins. The system waits for an input response from the sensors in order to cancel further actuation to the motor. In the case that slip of the object from the mechanism's grasp were to occur, and the threshold bias no longer met, the loop adjusts slightly or largely to correct this error, until the reading is above the threshold bias. In order to quantify this bias, the analog value received by the controller from the sensor is mapped to a 0-100 range, with 100 being the least sensitive (most force applied before bias is crossed.)

C. *Bias Force Testing*

In order to be able to grip various objects through the use of a single threshold bias, various values had to be experimented with. A value that met the following conditions would suffice for good operation of the control system:

- Gripping large or small objects within capability range without deforming them
- Gripping large or small objects capability range without destroying them
- Sensitive enough to detect when slip has caused a drop in pressure against the sensor

The experiment tested two values of bias: 55 and 22. These values represent just under quarter sensitivity and just over half sensitivity. The mechanism was allowed to grip objects of various sizes and geometries (Fig. 6) and the result of the gripping operation was recorded. Forces were measured and recorded if present on the sensors once the system stabilized and the mechanism had the object safely in its grasp. In order to calculate an integer force value present on the sensor, a simple calculation is carried out which maps the analog value received by the controller from the sensor to an input voltage range of 0-5 volts. This value is then used to find resistance through a voltage-divider formula, considering that each FSR is connected to the board along with a 10K resistor. An inductance value then reveals the approximate force integer value through the use of conditional relationships found experimentally from Force vs. Conductance graphs [3]. The results of the experiments are recorded and presented in the appendix.

Figure 6: Mechanism gripping a large, heavy ceramic

IV. RESULTS

Force values recorded ranged between 0 and 1 N, and revealed that not all the fingers touch an object when gripping it (not all sensors are tripped). For the bias value of 22, all objects were safely gripped that were cylindrical (Fig. 7) or elliptical (egg) in nature. For the case of "failure," this means that sensing the object was not successful, and the result of this was overexertion of force, as well as deformation of the mechanism and object being gripped. A rectangular large object and small spherical object were recorded as a failure. All other objects gripped yielded lateral force readings against the linkages of less than 1 N. This is not strange considering the mechanism is designed to grip lightly, and uses friction to advantage. In the two heaviest objects, slip occurred and was compensated for by the motor. For the heaviest object in the test set, this compensation was large, meaning that actuation continued considerably until the bias was crossed and stability was reached. In the case of the large ceramic mug, there was slight slip and adjustment, but ultimately stabilization was successful.

The bias value of 55 revealed more inconsistent results and higher lateral forces upon grip. This meant that it gripped objects with much more force than was necessary. Failure occurred in four instances, almost damaging the mechanism. Deformation of an object was present for the first time in the case of the cardboard tube. Slip occurred for the screw driver, and adjustment was very large, resulting in a long dampened wave-like response from the motor until stability was reached. Overall, setting the bias this high caused several inconsistencies in the system.

Figure 7: Mechanism gripping a small cylindrical object

V. Conclusions and Future Work

Regardless of the fact that force measurements might be inaccurate, the bias number testing scheme revealed that for most objects being gripped, which fit into the capability range, the system response is effective. A threshold bias of 22 yields just enough sensitivity for the objects to be safely gripped within the concentric circle that the linkages close around. The force required to grip the objects was generally very low, conserving electrical and mechanical energy efficiently during gripping. Further testing using a paper cup filled to the brim during a demonstration also revealed that the mechanism is able to carefully hold deformable objects, and adjust slightly without spilling or deforming the cup; much like a human would hold a coffee cup without a lid carefully. The silicone tabs made to simulate skin effectively transfer the lateral loading from the normal force of the object being gripped to the sensors, and these respond immediately, at the same time that the object is gripped effectively. This might be due largely to the fact that the coefficient of static friction for silicone against other objects is higher than most materials, creating a shear force that overtakes gravitational pull easily, and with much less force than most gripper devices without any rubber aid, or with aids of less sticky materials.

This system could be improved by the addition of other FSR sensors, considering that during testing, failures were attributed to no sensor contact during operation. These could be arrayed in the same manner that the present ones are arrayed. The system currently has sensors on thumbs, index finger, and pinky fingers.

The addition of more silicone tabs might also be of benefit, as well as a way of sandwiching the acrylic plate, FSR and silicone tab permanently. During testing and demoing, the silicone tabs were slipping off, most likely due to being attached by being taped to the mechanism. Permanent attachment would maximize grip, and would protect the sensors from damage.

On-off control is sufficient for the current scope of experimentation and operation, but future work with the mechanism could also lead to improvements to the algorithm, perhaps including the addition of Proportional-Integral-Derivative control for better response to error caused from slip. This would lead to more energy conservation in operation, and the ability to maximize grip while reducing sensitivity, due to faster operational response.

VI. Acknowledgements

The author wishes to thank David Araque, Amy Mclarty and Marcus Rodriguez for their help in the initial revision design and construction of the mechanism, Dr. Kyle Reed for professional and academic input.

References

[1] 2007 Thomson Higher education, *The cutaneous senses* (PowerPoint Presentation), 2007
[2] A. Jeeverajan, *Human Performance Capabilities.* NASA 2008, Section 4.9
[3] A. Limor, *Force Sensing Resistors Are for Squeezing!,* Ladyada.net 2011
[4] P. Payeur, "Intelligent Haptic Sensor System for Robotic Manipulation, " *IEEE Transactions On Instrumentations and Measurement,* pp. 1583-1592
[5] *How to build a hand with haptic feedback,* instructables.com 2012
[6] Ashwini K. Sule, *Afferent VS Efferent, www.buzzle.com/articles 2011*
[7] *Power Screw and Nut Wear, www.roton.com/power_nut.aspx 2005*
[8] *R. Howe, "Tactile sensing and control of robotic manipulation" Journal of Advanced Robotics*
[9] *K. Weiss, Heinz Woern, "Tactile Sensor System for an Anthropomorphic Robotic Hand" Institute of Process Control and Robotics*

VII. APPENDIX

A. *Experimental Data*

Threshold Bias = 22					Force Read N			
Object gripped	Failure	Deformation	Slip	Adjustment	pinky	r. thumb	index	left thumb
long cardboard tube	no	none	none	none	0	0	<1	<1
large plastic case	yes							
large ceramic mug	no	none	yes	slight	<1	0	<1	0
egg	no	none	none	none	0	0	<1	<1
large cylindrical candle	no	none	none	none	0	0	<1	0
screw driver	no	none	none	none	0	0	<1	<1
heavy odd shaped vase	no	none	yes	large	0	0	<1	<1
baseball	yes							

Table 1-Testing for bias of 22

Threshold Bias = 55					Force Readings-N			
Object gripped	Failure	Deformation	Slip	Adjustment	pinky	r.thumb	index	l. thumb
long cardboard tube	no	yes	none	none	<1	<1	1	1
large plastic case	no	no	yes	large	0	<1	0	1
large ceramic mug	no	no	none	none	0	0	1	0
egg	yes							
large cylindrical candle	yes							
screw driver	no	no	yes	wave-like	0	0	<1	<1
heavy odd shaped vase	yes							
baseball	yes							

Table 2-Testing for bias of 55

B. *Arduino Code*

```
// Using FSR sensors to control grip on objects
// adjust bias to set threshold for how sensitive sensors are
// adjust force reading gains to match real forces
// Louis Melgar
// April 28, 2012

#include <Servo.h>


Servo myservo;  // create servo object to control a servo
// Attaching FSRs to pins and values
int fsr1 = 0;  // analog pin used to connect the first FSR
int val1;    // variable to read the value from the analog pin
int fsr1Reading; // variable for force calculations for pinky
int fsr2 = 1;  // analog pin used to connect the second FSR
int val2;    // variable to read the value from the analog pin
int fsr2Reading; // variable for force calculations for right
thumb
int fsr3 = 2;  // analog pin used to connect the third FSR
int val3;    // variable to read the value from the analog pin
int fsr3Reading; //variable for force calculations for index
int fsr4 = 3;  // analog pin used to connect the third FSR
int val4;    // variable to read the value from the analog pin
int fsr4Reading; // variable for force calculations for left
thumb

// Manual ovveride switches
int openswitchpin = 8; // pin for opening hand manually
int closeswitchpin = 7; // pin for closing hand manually
int programswitchpin = 4; // pin for controlling program

//Force calculation memory space
int fsr1Voltage;     // the analog reading converted to voltage
unsigned long fsr1Resistance;  // The voltage converted to
resistance, can be very big so make "long"
unsigned long fsr1Conductance;
long fsr1Force;       // Finally, the resistance converted to
force

int fsr2Voltage;     // the analog reading converted to voltage
unsigned long fsr2Resistance;  // The voltage converted to
resistance, can be very big so make "long"
unsigned long fsr2Conductance;
long fsr2Force;       // Finally, the resistance converted to
force

int fsr3Voltage;     // the analog reading converted to voltage
unsigned long fsr3Resistance;  // The voltage converted to
resistance, can be very big so make "long"
unsigned long fsr3Conductance;
long fsr3Force;       // Finally, the resistance converted to
force

int fsr4Voltage;     // the analog reading converted to voltage
unsigned long fsr4Resistance;  // The voltage converted to
resistance, can be very big so make "long"
unsigned long fsr4Conductance;
long fsr4Force;       // Finally, the resistance converted to
force

// Sensitivity threshhold Bias
int sens = 22; // adjust using 0-100 gain, 0-highest sensitivity

void setup()
{
 Serial.begin(9600);
 myservo.attach(9);  // attaches the servo on pin 9 to the
servo object
 pinMode(openswitchpin, INPUT);
 pinMode(closeswitchpin, INPUT);
 pinMode(programswitchpin,INPUT);
 }

void loop()
{


 //....................Motor Info... Vex 393 w/ Motor controller
29.............
 // These are speed values for the motor drive open/close
operations
 // full open is 65
 // partial open is 85
 // neutral is 95
 // partial close  is 105
 // full close is 115
```

```
  //......................Main Execution routine................................

  // If kill-switch on, begin test sequence program
  if (digitalRead(programswitchpin) == HIGH)
  {
  //Read and scale FSR1
  val1 = analogRead(fsr1);// reads the value of the
potentiometer (value between 0 and 1023)
  val1 = map(val1, 0, 1023, 0, 100);// scale it to use with
sensitivity
  fsr1Reading = analogRead(fsr1); // Reading for force
calculations
  //Serial.println(val1);
  //delay(1500);
  // Test FSR1 pressure values
  // high pressure yields val1 of 125-145
  // extremely high pressure yields va11 of 145-155
  // silly putty tends to dampen signal, and mild contact with
acrylic surface yields
  // value of about 90 TO 115
  // analog voltage reading ranges from about 0 to 1023
which maps to 0V to 5V (= 5000mV)
  fsr1Voltage = map(fsr1Reading, 0, 1023, 0, 5000);
  if (fsr1Voltage == 0) {
    Serial.println("No pressure on pinky");
  } else {
    // The voltage = Vcc * R / (R + FSR) where R = 10K and
Vcc = 5V
    // so FSR = ((Vcc - V) * R) / V
    fsr1Resistance = 5000 - fsr1Voltage;     // fsrVoltage is in
millivolts so 5V = 5000mV
    fsr1Resistance *= 10000;                // 10K resistor
    fsr1Resistance /= fsr1Voltage;
    fsr1Conductance = 1000000;           // we measure in
microohms
    fsr1Conductance /= fsr1Resistance;

    // Use the two FSR guide graphs to approximate the force
    if (fsr1Conductance <= 1000) {
      fsr1Force = fsr1Conductance / 80;
      Serial.print("Force in Newtons on pinky: ");
      Serial.println(fsr1Force);
    } else {
      fsr1Force = fsr1Conductance - 1000;
      fsr1Force /= 30;
      Serial.print("Force in Newtons on pinky: ");
      Serial.println(fsr1Force);
    }
  }

  //Read and scale FSR2
  val2 = analogRead(fsr2);// reads the value of the
potentiometer (value between 0 and 1023)
  val2 = map(val2, 0, 1023, 0, 100);// scale it to use it with
the servo (value between 0 and 180)
  fsr2Reading = analogRead(fsr2); // Reading for force
calculations
  //Serial.println(val2);
  //delay(1500);
  // Test FSR2 pressure values
  // high pressure yields val1 of 130-140
  // extremely high pressure yields va11 of 140-150
  // silly putty tends to dampen signal, and mild contact with
acrylic surface yields
  // value of about 80 TO 90
  fsr2Voltage = map(fsr2Reading, 0, 1023, 0, 5000);
  if (fsr2Voltage == 0) {
    Serial.println("No pressure on right thumb");
  } else {

    fsr2Resistance = 5000 - fsr2Voltage;
    fsr2Resistance *= 10000;
    fsr2Resistance /= fsr2Voltage;
    fsr2Conductance = 1000000;
    fsr2Conductance /= fsr2Resistance;


    if (fsr2Conductance <= 1000) {
      fsr2Force = fsr2Conductance / 80;
      Serial.print("Force in Newtons on right thumb: ");
      Serial.println(fsr2Force);
    } else {
      fsr2Force = fsr2Conductance - 1000;
      fsr2Force /= 30;
      Serial.print("Force in Newtons on thumb: ");
      Serial.println(fsr2Force);
    }
  }

  //Read and scale FSR3
  val3 = analogRead(fsr3);// reads the value of the
potentiometer (value between 0 and 1023)
  val3 = map(val3, 0, 1023, 0, 100);// scale it to use it with
the servo (value between 0 and 180)
  fsr3Reading = analogRead(fsr3); // Reading for force
calculations
  //Serial.println(val3);
  //delay(1500);
  // Test FSR3 pressure values
  // high pressure yields val1 of 125-135
  // extremely high pressure yields va11 of 145-155
  // silly putty tends to dampen signal, and mild contact with
acrylic surface yields
  // value of about 90 TO 110
  fsr3Voltage = map(fsr3Reading, 0, 1023, 0, 5000);
  if (fsr3Voltage == 0) {
    Serial.println("No pressure on index");
  } else {
    /
    fsr3Resistance = 5000 - fsr3Voltage;
    fsr3Resistance *= 10000;
    fsr3Resistance /= fsr3Voltage;
    fsr3Conductance = 1000000;
```

```
    fsr3Conductance /= fsr3Resistance;


    if (fsr3Conductance <= 1000) {
      fsr3Force = fsr3Conductance / 80;
      Serial.print("Force in Newtons on index: ");
      Serial.println(fsr3Force);
    } else {
      fsr3Force = fsr3Conductance - 1000;
      fsr3Force /= 30;
      Serial.print("Force in Newtons on index: ");
      Serial.println(fsr3Force);
    }
  }

  //Read and scale FSR3
  val4 = analogRead(fsr4);// reads the value of the
potentiometer (value between 0 and 1023)
  val4 = map(val4, 0, 1023, 0, 100);
  fsr4Reading = analogRead(fsr4); // Reading for force
calculations
  //Serial.println(val4);
  //delay(1500);
  // Test FSR3 pressure values
  // high pressure yields val1 of 125-135
  // extremely high pressure yields va11 of 145-155
  // silly putty tends to dampen signal, and mild contact with
acrylic surface yields
  // value of about 90 TO 110

  fsr4Voltage = map(fsr4Reading, 0, 1023, 0, 5000);
  if (fsr4Voltage == 0) {
    Serial.println("No pressure on left thumb");
  } else {

    fsr4Resistance = 5000 - fsr4Voltage;
    fsr4Resistance *= 10000;
    fsr4Resistance /= fsr4Voltage;
    fsr4Conductance = 1000000;
    fsr4Conductance /= fsr4Resistance;


    if (fsr4Conductance <= 1000) {
      fsr4Force = fsr4Conductance / 80;
      Serial.print("Force in Newtons on left thumb: ");
      Serial.println(fsr4Force);
    } else {
      fsr4Force = fsr4Conductance - 1000;
      fsr4Force /= 30;
      Serial.print("Force in Newtons on left thumb: ");
      Serial.println(fsr4Force);
    }
  }
  Serial.println("--------------------");
  //delay(1000);
    //If no sensor is tripped, continue closing motor
    if (val1 < sens && val2 < sens && val3 < sens && val4 <
sens)
      {
        myservo.write(65); // closing value

      }
      // If sensor tripped, stop running motor
      else if (val1 > sens || val2 > sens || val3 > sens || val4 >
sens)
      {
        myservo.write(0); // zero pulse value
      }
  }
  // If kill-switch off, human controls
  else if (digitalRead(programswitchpin) == LOW)
  {
    // If pushbutton open true
    if (digitalRead(openswitchpin) == HIGH)
    {
      myservo.write(120);// Open hand
    }
    // If pushbutton two true
    else if (digitalRead(closeswitchpin) == HIGH)
    {
      myservo.write(65);// Close hand
    }
    else if (digitalRead(openswitchpin)== LOW &&
digitalRead(closeswitchpin)== LOW)
    {
      myservo.write(0);// Close hand
    }
  }
  // End of program
}
```