Robotic Hand Control by Flex Sensor & Arduino

Pratik Tarale
Department of Electronic and Telecommunication Engineering
Prof.Ram Meghe college of Engineering and Management, Badnera, India

Abstract— The present manuscript deals with designing a haptic robotic are, which can be used to pick and place the object. In the paper, a robotic are with four degree of freedom is designed and is able to pick the object with a specific weight and place them in a desire location. Animatronics is the use of mechatronics to create machines which seem animate rather than robotic. Animatronic figures are most often powered by pneumatics (compressed air), and, in special instances, hydraulics (pressurized oil), or by electrical means. The figures are precisely customized with the exact dimensions and proportions of living creatures. Motion actuators are often used to imitate "muscle" movements, such as limbs to create realistic motions. The project idea came to us after watching the movie named "Real Steel". We wanted to make a shadow robot from our curiosity. As the whole body of the robot would have been of much cost, we decided to make a shadow hand instead. Approximating the kinematics of the human hand was our top priority when developing this animatronic hand. Each joint of this hand has a movement range again the same as or very close to that of a human hand, including the thumb and even the flex of the palm for the little finger.

Key words: Arduino, Servo Motors, Flex Sensors, Power IC, Arduino Programming

I. Introduction

This paper mainly analysis about different topologies and designs regarding the construction of this Arduino based animatronic hand. Although more complicated and precise(more expensive) versions of this concept have been developed, this is a fun project with many potential applications. Interactive robot control of this level, I think, has many uses in industrial manufacturing, medical research, and anything you want to be able to do with precision that is unsafe to touch. The basic components of the hand and glove are the hand itself, the servos, the Arduino, the glove, and the flex sensors. The glove is mounted with flex sensors: variable resistors that change their value when bent. They're attached to one side of a voltage divider with resistors of a constant value on the other side. The Arduino reads the voltage change when the sensors are bent, and triggers the servos to move a proportional amount. The servo pull strings that act as tendons, allowing the fingers to move This project has been completed successfully, and my goal of integrating all of the underlying technologies has been met. This animatronic hand is able to be controlled according to the controller's wish. It's capable of moving at the required degrees of freedom. It can also pick up things upto minimum desired weight . Now, we can use it as a shadow hand of ours which is of various use. With the help of sensors, this hand can now provide detailed telemetry, which can be exploited to generate innovative manipulation control systems or to provide detailed understanding of the external environment. In the sections that follow, I will describe various outstanding areas that could stand to see some further refinement.

II. System Methodology

A. Robotic glove

Robotic glove houses the circuitry which controls the robotic arm. It consists of Arduino Mega 2560 which is programmed in such a way that it transfers the required data with the help of APC – 220 Module as well as it receives the data transmitted by the robotic arm.

The Gyroscope and Accelerometer installed takes the angles (alpha, beta and gamma) and acceleration in all three directions of the hand respectively, sends the signals to the Arduino Mega via wires where the values are combined and processed simultaneously. At the same time the Flex sensor is doing its job by sending the degree of movement of the finger to the Arduino Mega. The processed values are then transmitted from the Module (Trans-receiver) to the robotic arm. The module takes the feedback from the arm and sends the new processed signals to it.

B. Robotic arm

It is the main instrument where implementation of the program from the robotic glove takes place. It consists of total of 6 nos. of Servos, connected in such a way that it provides 3 DOF's (Degrees of Freedom) to the system. A micro controller Arduino UNO which inputs the values from the module and sends the data accordingly to the servos. Both the circuitry and base are clipped upon the common base, made of acrylic to improve the stability. The whole chassis is made of Acrylic; the lowest point servo is attached in such a way that it moves the upper base horizontally from 0-180 degree depending upon the values from the APC-220 Module. 4 more servos are attached for controlling the direction of the robot in the vertical direction mounted on the base servo. At the apex, a mini servo is connected with the help of gears to control the Grabber of the robot. As mentioned above, the robotic arm mimics the movement of glove worn by the user, when the glove is tilted in the forward direction or any such direction, the arm spontaneously follows suit.

The Grabber functions with the use of flex sensors attached to center most finger, the degree of movement of the finger determines the angle of the mini servo which is attached to the gears controlling the movement of the grabber.

C. Functioning of robotic arm:

A small object of low weight is placed near the robotic arm at a distance within the approach of arm. The system is made on. The operator stands at a distance from the robot and moves the finger/hand up, down, left or right. The robotic arm follows the direction. The arm is brought over the object and then lowered. The grabber is fully opened to pick up the object. The robotic arm then is moved up and rotated to another desired position, then lowered. When the arm reaches the ground floor, the grabber is given a command to release the object, which places it at the desired location. This way the robotic arm can be operated and controlled in any manner as deemed necessary by the operator from a distance, usually up to 200 meters.

III. WORKING PRINCIPAL OF ARDUINO HARDWARE

The Arduino microcontroller is an easy to use yet powerful single board computer that has gained considerable traction in the hobby and professional market. The Duemilanove board features an Atmel ATmega328 microcontroller operating at 5 V with 2 Kb of RAM, 32 Kb of flash memory for storing programs and 1 Kb of EEPROM for storing Parameters. The clock speed is 16 MHz, which translates to about executing about 300,000 lines of C source code per second. The board has 14 digital I/O pins and 6 analog input pins. There is a USB connector for talking to the host computer and a DC power jack for connecting an external 6-20 V power source, for example a 11.1 V battery, when running a program while not connected to the host computer. Headers are provided for interfacing to the I/O pins using 22 g solid wire or header connectors. The Arduino programming language is a simplified version of C/C++. If you know C, programming the Arduino will be familiar. If you do not know C, no need to worry as only a few commands are needed to perform useful functions. An important feature of the Arduino is that we can create a control program on the host PC, download it to the Arduino and it will run automatically. Remove the USB cable connection to the PC, and the program will still run from the top each time you push the reset button. Remove the battery and put the Arduino board in a closet for six months. When you reconnect the battery, the last program you stored will run. This means that you connect the board to the host PC to develop and debug your program, but once that is done, you no longer need the PC to run the program.

IV. WORKING PRINCIPAL OF FLEX SENSOR

Flex sensors are sensors that change in resistance depending how much the sensor is bend. Sensors convert the change in bend to electrical resistance - the more the sensor bend, the higher the resistance value. Using the Flex Sensor is very easy. There are couple of different manufacturers in the market. Datasheet instructs you to use operational amplifier (opamps). That may be useful if you plan to use flex sensor as stand-alone device (without any microcontroller).Because We are using arduino, We skipped all OpAmps and made a very simple circuit with only one additional resistor.

Varying the value of the resistor will results different readings. With 22k Ohm resistor I will get values between 300-700. This works fine for us. In our code we assumed that all values under 400 mean that the sensor is bend. All values above 600 mean that sensor is nor bend. Note that Flex sensor give reliable readings only if you bend it on the specific direction (usually towards on the text side of the sensor).

V. SPECIFICATION HARDWARE DETAILS

A. Servo Motor

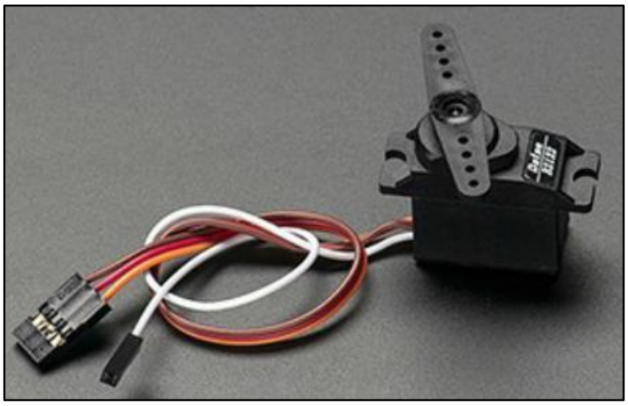


Fig. 1: Servo Motor

B. Servos Moving Erratically Causes with Trouble Shooting:

As with any animatronic project, many times this will come down to a complex system interaction between our microcontroller / pc and the servo controller. Usually the servos are just fine. Here are some things that have been found with testing done on multiple PC's and servo controllers:-

1) Cause- :

Mechanisms may have been subject to dirt or adverse conditions. This can be confirmed by disconnecting the mechanism from the servo.

a) Trouble Shoot:

We disconnected the mechanism from the servo then again connected the mechanism. If there were still issues the hand would have to be carefully disassembled, cleaned, re-assembled and tuned as above. But it didn't work either.

2) Cause-2:

The of cause of this can be almost always a "noisy" power supply. Even though we may had carefully selected an AC to DC 5VDC output supply that can handle 1 A pull, it may not be producing a quiet voltage.

a) Trouble Shoot:

Any control system can be riddled with connection issues. Servos are usually the most trouble-free devices in the entire loop.,So. We backtracked into our servo controller settings (rate mis-match, etc), cable connections, ARDUINO interference, and our code. 2. As the above did not solve the issue,we tried plugging in a simple battery supply of about 11.1 VDC and then converted the voltage of the supply into 3VDC which is the threshold voltage of the servo by adjusting the voltage of a power IC. In all cases this has corrected this issue in our testing.

3) Cause-3:

If your powering the servos with a different power source than the Arduino, they need to share the same ground or the PWM signals will corrupt between the Arduino and servos. Fingers / thumb are moving but erratically.

a) Trouble Shoot:

We checked all our ground connections properly as the above trouble shooting process couldn't resist the servos form moving erratically in total. After that we reconnected those thus we managed to solve this problem finally. .

4) Servos Being So Noisy

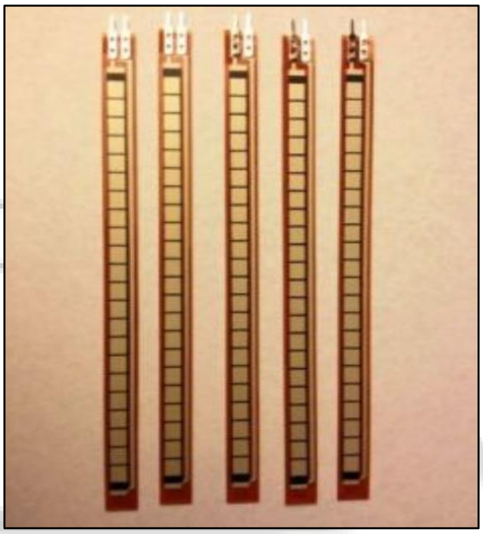
a) Causes with Trouble Shooting:

The word servo refers solely to a device that uses negative feedback for control. One major drawback to working with servos is the large amounts of electrical noise they produce. This noise can interfere with your sensors and can even impair your microcontroller by causing voltage dips on your regulated power line. Large enough voltage dips can corrupt the data in microcontroller registers or cause the microcontroller to reset.

(1) Cause-1:

Cheap brushed motors can be noisy. Cheap hobby grade servos can sometimes chatter if they do not settle in a stable state. This is normal and is caused by poor tuning, a lack of a dead band, and backlash between the motor and the encoder (potentiometer).

C. Flex Sensor



The flex sensors require a circuit in order for them to be compatible with Arduino. It's a voltage divider: the flex sensors are variable resistors, and when paired with resistors of a static value, change in resistance (in this case bending the sensor) can be sensed through the change in voltage between the resistors. This can be measured by the Arduino through its analog inputs. The schematic is attached (red is positive voltage, black is negative, and blue goes to the Arduino). The resistors in the photo are 22K. I color-coded the wires we used in the same way as the schematic, so we can see more easily. The main GND wire, which is connected to all the individual GND wires from the sensors, gets plugged into the Arduino's GND. The +5V from the Arduino goes to the main positive voltage wire, and each blue wire gets

VI. APPLICATIONS

This Haptic Robot arm is used for various applications, of them few are as follows:

- To lift heavier objects
- To lift nuclear wastes without harming the humans
- Used as external limbs of a surgeon during a complex retinal or hear surgeries Prototype for a Bomb disposal robot

VII. FUTURE SCOPE

The robotic arm so far designed is able to lift the objects. It is able to lift the objects of medium weight. In order to extend it to some extent, more advanced tools and material with the capacity to withhold the heavy weight objects are to be used, which is then applicable in warfront and used as a rescuer at several places where there is a need and also in industrial areas, military, and so on.

VIII. CONCLUSION

We are glad that we chose to complete this project on the Arduino. It was our first real coding experience on this platform, and we can say that compared to writing C++, writing Wiring libraries for Arduino makes for a much more fun and Productive experience We are grateful that our time on the C++ taught us a lot about what is happening behind the scenes, but quiet honestly it is nice to not have to worry about it so much. One thing we learned from this project is that servos and flex sensors in positioning, timing and environmental texture can lead to all sorts of undesirable readings. We were a bit disappointed with the performance of the SG90 servos in this particular use case, It required a lot of the fine-tuning to get readings accurate as the servo rotated. As stated previously, another area I need to look into is battery power. This project is a poor use case for a 11.1V battery. A better long term portable power supply would include a higher efficiency in the output. That's why we also used a PC for the long term power supply. Although the Animatronic hand did not operate with no errors, it is a great success overall. The Animatronic hand met all safety restrictions, easy to operate and energy efficient. This types of animatronic hand can be used for various puposes. The Animatronic Hand can be implemented in all the sectors where human interaction is needed, like-Handling of the explosive objects, performing various sophisticated operational jobs in the medical sectors, Industrial manufacturing etc. With more time and resources put for things like motors and base design we can carry a much larger payload and have a sturdier platform to carry things in. Much of this project could be used or improved upon by future EEE students.

REFERENCES

[1] Servo Motor | Servo Mechanism | Theory and Working Principle http://www.electrical4u.com/servo-motor-servo-mechanism-theory-and-working-principle/
[2] DC Servo Motors | Theory of DC Servo Motor http://www.electrical4u.com/dc-servo-motors-theory-and-working-principle/
[3] Introduction to Servo Motors http://mechatronics.mech.northwestern.edu/design_ref/actuators/servo_motor_intro.html
[4] DIY Robotic Hand Controlled by a Glove and Arduino by dschurman http://www.instructables.com/id/DIY-Robotic-Hand-Controlled-by-a-Glove-and-Arduino/
[5] Animatronics http://en.wikipedia.org/wiki/Animatronics
[6] Simple Servo Bed for InMoov http://www.thingiverse.com/thing:65274

[7] AniHand - Animatronic Hand http://letsmakerobots.com/node/34639
[8] Calibrator: An Arduino library to calibrate sensors hooked to analog inputs http://julianvidal.com/blog/calibrator-an-arduino-library-to-calibrate-sensors-hooked-to-analog-inputs/
[9] Arduino - Begin. http://arduino.cc/en/Serial/begin
[10] Arduino learning pdf http://www.ele.uri.edu/courses/ele205/Arduino%20-%20Learning.pdf
[11] Arduino Microcontroller Guide http://www.me.umn.edu/courses/me2011/arduino/arduinoGuide.pdf
[12] Arduino functions http://playground.arduino.cc/Code/Function
[13] Structuring Your Code into Functional Blocks https://www.inkling.com/read/arduino-cookbook-michael-margolis-2nd/chapter-2/recipe-2-10
[14] Arduino programming Problem solution http://stackoverflow.com/questions/12975895/arduino-argument-of-type-void-classname-does-not-match-void-whi
[15] Blink Without Delay http://arduino.cc/en/Tutorial/BlinkWithoutDelay
[16] Operating Two Servos with the Arduino http://www.robotoid.com/appnotes/arduino-operating-two-servos.html
[17] Servo Problems With Arduino http://rcarduino.blogspot.com/2012/04/servo-problems-with-arduino-part-1.html
[18] Problem with arduino servo http://electronics.stackexchange.com/questions/93746/problem-with-arduino-servo
[19] Arduino Servo control problem, external servo power source https://forum.sparkfun.com/viewtopic.php?f=32&t=24263
[20] Problem with multi servos on Arduino http://www.youtube.com/watch?v=iZFO_8hH7QY
[21] Multiple servo control http://www.instructables.com/id/Serial-Servo-Controller-wAduino-Control-Up-To-1/
[22] Flex sensor http://mech207.engr.scu.edu/SensorPresentations/Jan%20-%20Flex%20Sensor%20Combined.pdf
[23] Arduino map() method http://stackoverflow.com/questions/9024124/arduino-map-method-why
[24] Arduino programming tutorial http://opensourcehardwaregroup.com/tutorial-09-reading-analog-pins-and-converting-the-input-to-a-voltage/