Sensors and Motor Control Lab

Individual Lab Report – 1

Nithin Subbiah Meganathan

Team D: CuBi

Team Members

Jorge Anton Garcia

Paulo Camasmie

Laavanye Bahl

Changsheng Shen

Carnegie Mellon University

14th February 2019

Sensors and Motor Controls Lab

Individual Progress

My part in this assignment was to control the servo motor and obtain readings from the force-resistance sensor.

Force-sensitive resistor works on the principle of voltage divider configuration. The schematic of the circuit is obtained from the datasheet and the exact circuit was reproduced on a breadboard, powered and controlled through Arduino. The sensor was connected to an analog pin and 5V supply was provided. The value of R_M was chosen to be 47k ohms which was in the operating range of the sensor and was known from the datasheet.

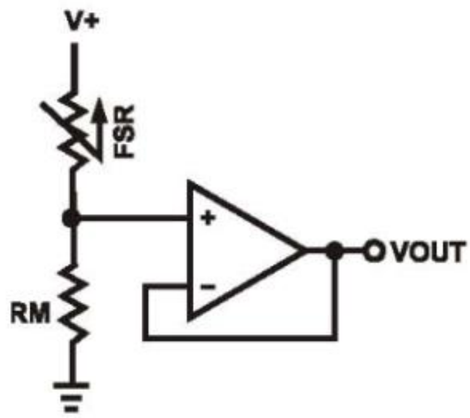


Figure 1. Schematic for Sensor Circuit[1]

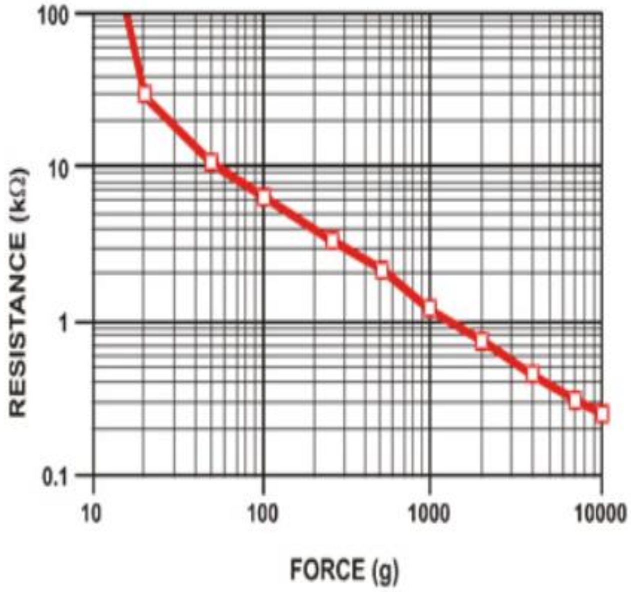


Figure 2. Force Curve[1]

The transfer function was obtained from the force curve in the datasheet. The (x,y) points corresponding to the force and resistance respectively were tabulated and curve-fitting was done. The resulting equation was used to map the resistance to the force. The resistance that maps to the force is calculated from the following equation,

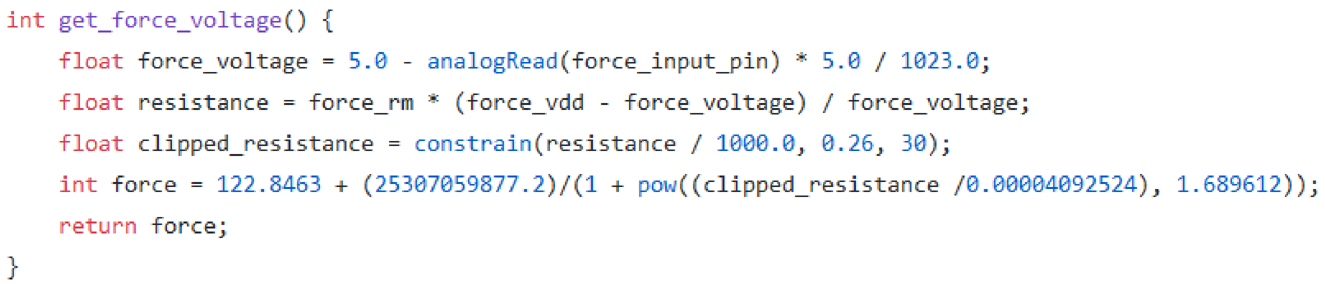
$$V_{OUT} = \frac{R_M V+}{(R_M + R_{FSR})}$$

where V_{OUT} is the measured output, V+ (5V) and R_M (47k ohms) are known. Filters were not implemented for this sensor as the output seemed to be pretty stable, but the range of resistance was provided which approximated the linear line in the force curve (fig 2). Calibration was also done by observing the real time values.

My next task was to control the servo motor. This was achieved by connecting the servo motor to an Arduino, which also powered the motor. The program was such that it receives angle as an input from the sensor interfacing and commands the servo to reach that angle position. A condition was set that the input angles must lie between 0 to 180 degrees otherwise the servo does not respond.

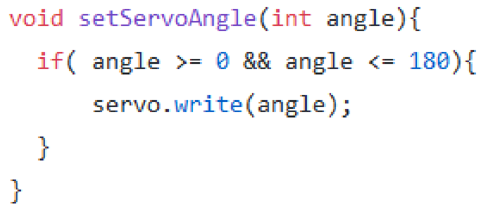
Additionally, I also found the transfer function for IR sensor using graph from the datasheet.

Code Snippets



```
int get_force_voltage() {
    float force_voltage = 5.0 - analogRead(force_input_pin) * 5.0 / 1023.0;
    float resistance = force_rm * (force_vdd - force_voltage) / force_voltage;
    float clipped_resistance = constrain(resistance / 1000.0, 0.26, 30);
    int force = 122.8463 + (25307059877.2)/(1 + pow((clipped_resistance /0.00004092524), 1.689612));
    return force;
}
```

Figure 3. Code Snippet for Force Sensitive Sensor



```
void setServoAngle(int angle){
  if( angle >= 0 && angle <= 180){
      servo.write(angle);
  }
}
```

Figure 4. Code Snippet for Servo Motor Control

Challenges

A challenge that my task involved was the way to obtain the transfer function. My initial plan was to calibrate the sensor manually by reading the voltage from sensor and correlating with the physical quantity but since I got a force sensor, I was unsure how to go about it. However, going through the datasheet gave me an idea to use the graphs provided. It was difficult to get an accurate graph but that was the best that could be obtained. Sample points from the graph were then used to fit a curve and obtain the transfer function. Additionally, integrating the code and sensors with everyone’s work proved to be a herculean task which Jorge completed.

Teamwork

Laavanye was responsible for controlling the stepper motor using ultrasonic sensor and Paulo worked on controlling the DC motor using PID. Jorge interfaced the microsensor and integrated everyone's code and electronics. He also prepared the communication protocol on the Arduino side. Bobby was responsible for the GUI and he also worked on integration with Jorge.

Figure

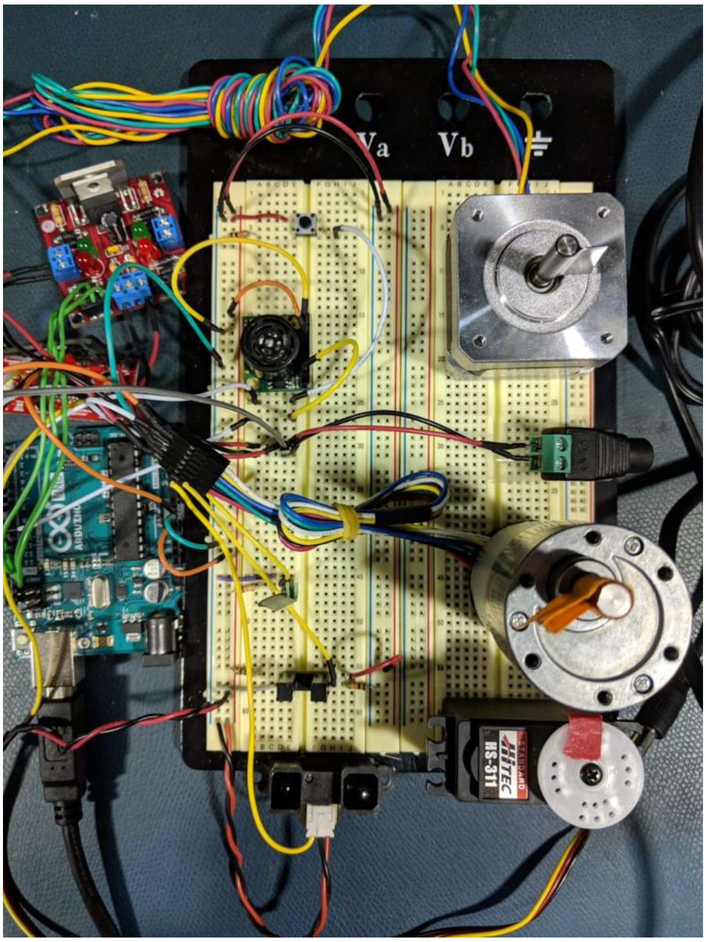


Figure 5. Circuit with all components

Team CuBi

Individual Progress

My contribution was validating different manipulator designs that Paulo proposed or had designed. As the person responsible for team management along with Jorge, I have been involved in going through the system engineering aspects of our project, laying out the system-level objectives that we need to meet as a team, and preparing the schedule. I also started out to control the Dynamixel motors that we are using throughout our robot. I had been partnering with Jorge to do these tasks.

Challenges

Manipulator design is one of the highest risks in our project and I see this as a potential challenge moving forward. However, we are actively working on this currently and hope to prototype the final design soon. Another important challenge is working in tandem with other subsystems. While laying out the system-level objectives, it was clear to me how every task is tied up to other's work. Sensors and Motor Control lab also proved how it's easy to work individualy but putting everyone's work together is a huge challenge. So, it is going to be hard in integrating everyone's work while building the system even at initial stage.

Teamwork

Jorge had been working on the control of Dynamixel motors and preparing a schedule for our team. Laavanye and Bobby finalized the sensors for CuBi. Laavanye then started on image segmentation using the sensors. Bobby had built the our base Turtlebot Waffle Pi 3. Paulo had been designing the manipulator and prototyping using the 3D printer. He has made a couple of design iterations till now.

Plans

My plans are to achieve the control of Dynamixel motors and affix them to the manipulator. Also, I need to create a schedule that the team can follow. Apart from this, I would like to familiarize myself with the vision subsystem in CuBi.

Figures

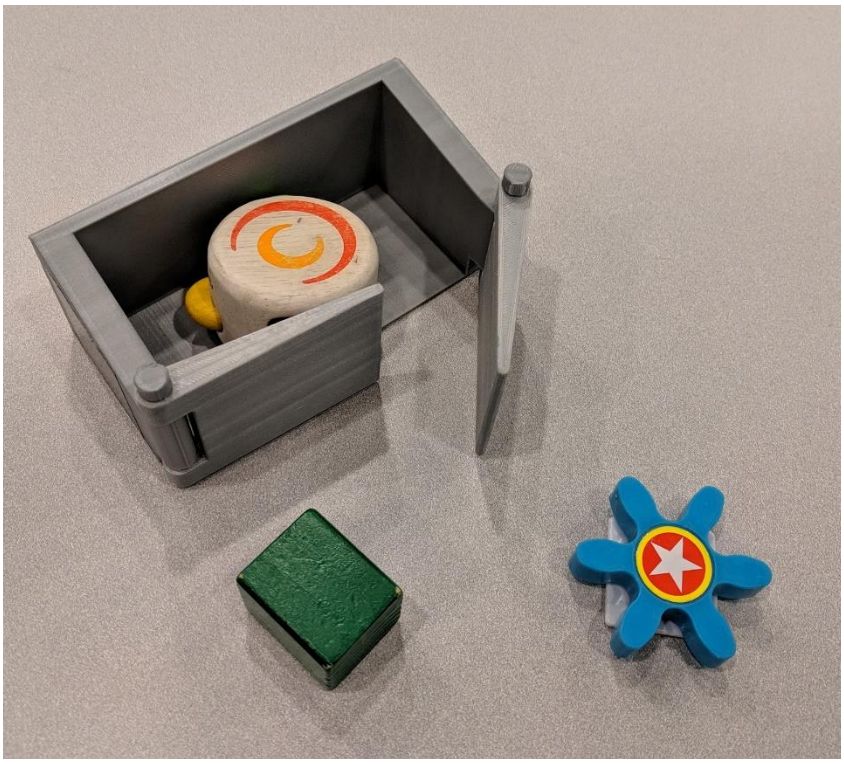


Figure 6. Initial Gripper Prototype

References

1. https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/2010-10-26-DataSheet-FSR400-Layout2.pdf

Sensors and Motors Control Lab Quiz

1.

a. The accelerometer measures in the range of ±3.6g, where g is the acceleration due to gravity ($g=9.81m/s^2$).

b. The dynamic range of the accelerometer is 7.2g, where g is the acceleration due to gravity ($g=9.81m/s^2$).

c. The C_{DC} capacitor provides a steady DC voltage supply. Even though the input voltage is DC, it is susceptible to deviations from the constant voltage. C_{DC} discharges stored charge whenever the voltage across it drops thus maintaining a steady DC supply.

d. Transfer function: $V = 3 + 0.3*a$, a is the acceleration in g.

e. 0.009mV

f. 750 µg

g. I would place the sensor stationary and measure the output. Deviation from the expected zero acceleration output is the noise.

2. Filtering

   In a moving average filter, impulses can drive the output to a high and create an anomaly in the average output.

   A median filter is computationally intensive and the beginning window is usually a problem to compute since we wouldn't have enough values to fit our window size.

   Opamps

   Sensor range: -1.5V to 1.0V

   1. V_2 will be the input voltage and V_1 the reference voltage since we need a non-inverting opamp to calibrate the sensor as required.
   2. $V_{out} = (V_2-V_1)*(R_f/R_i) + V_2$

      $V_{out}=0V$ for $V_2=-1.5V$
      $1.5 = (-1.5-V_1)*(R_f/R_i)$

      $V_{out}=5V$ for $V_2=1.0V$
      $5=(1-V_1)*(R_f/R_i) + 1$

      Solving both equations, $V_1 = -3$ V
   3. $R_f/R_i = 1$

   Sensor range: -2.5V to 2.5V

   1. V_2 will be the input voltage and V_1 the reference voltage since we need a non-inverting opamp to calibrate the sensor as required.
   2. Calibration cannot be done for this case because the gain needs to be one for this case but if the gain is one ($R_f/R_i =0$) then there cannot be any offset as needed through calibration.

      $V_{out} = V_{in} (1+ R_f/R_i) – V_{ref}(R_f/R_i)$

3. a. Encoder gives the current position which can be used to get the error between the current and target position. A PWM can then be applied to the input signal to each of the PID terms.

b. Increase the proportionate gains as this puts a penalty on the position error that drives the system faster to the target.

c. Integral term should be used as this puts a penalty on the sum of errors accumulated over the period of attaining the target position thus eliminating the steady state error.

d. Derivative term should be applied as this controls the rate at which error converges to zero thus controlling the overshoot.