

Individual Lab Report – 1

Sensors and Motor Control Lab

By Pratibha Tripathi

Team F – Falcon Eye

Team Members:

Danendra Singh

Pulkit Goyal

Rahul Ramakrishnan

Yuchi wang

October 13,2017

Individual Progress

I was responsible for programming IR sensor and control servo motor based on the obstacle distance from sensor.

IR Sensor

Steps in Programming IR sensor:

- 1) I connected Yellow wire of IR sensor to Analog Input A2, Red wire to VCC and Black to ground of the Arduino Uno.
- 2) I coded IR in Arduino to Serial read the data.
- 3) IR sensor has IR LED (Emitter) and IR Photodiode (Detector). The data from IR sensor depends on the distance of the input, reflectivity of surface and lighting in space.
- 4) The data comes from (0 - 1023) on serial monitor varying with the distance.

Setup for Taking Readings

- 1) I fixed IR Sensor at one end of the bench in Lab and fixed a L shape scale.
- 2) I varied the distance of a paper from IR sensor in increments of 1 inch and took readings from serial monitor.

Interpretation of the Readings

- 1) Readings increased from till 4 inches and then started to decrease as I increased the distance.
- 2) I checked in the data sheet of the IR for validating the relationship, so the valid range is after 4 inches.
- 3) So, for calculating the transfer function, I didn't take readings till 4 inches into consideration.

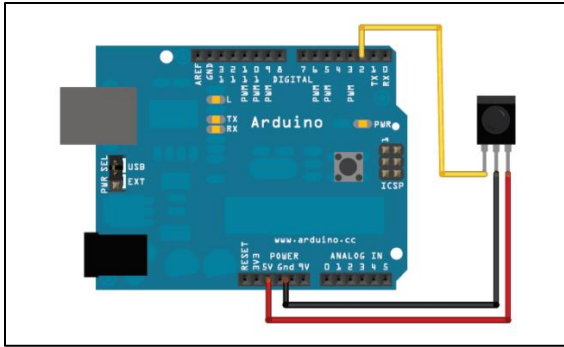


Fig 1: IR connection with Arduino^[3]



Fig 2: IR Sensor^[1]

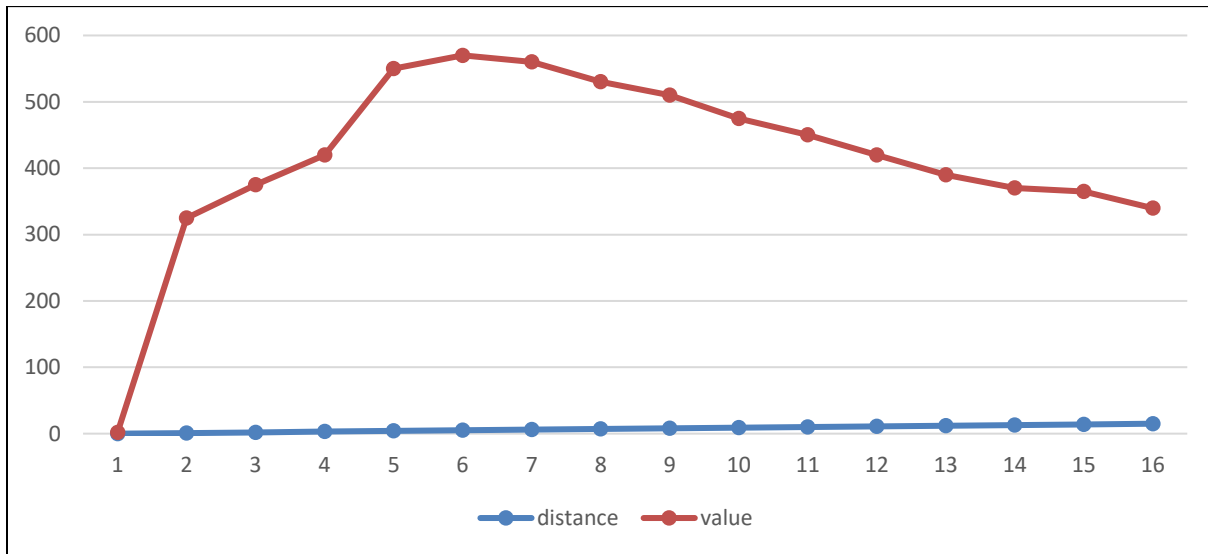


Fig 3: IR values displayed on Serial Monitor in complete range

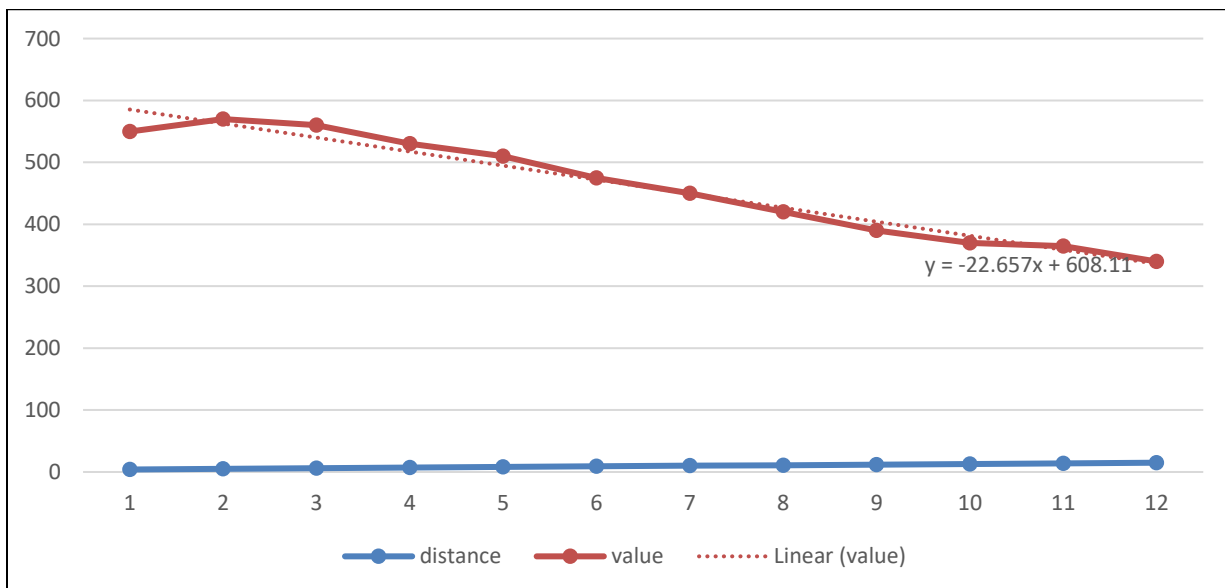


Fig4: IR values displayed on Serial Monitor in effective range and transfer function

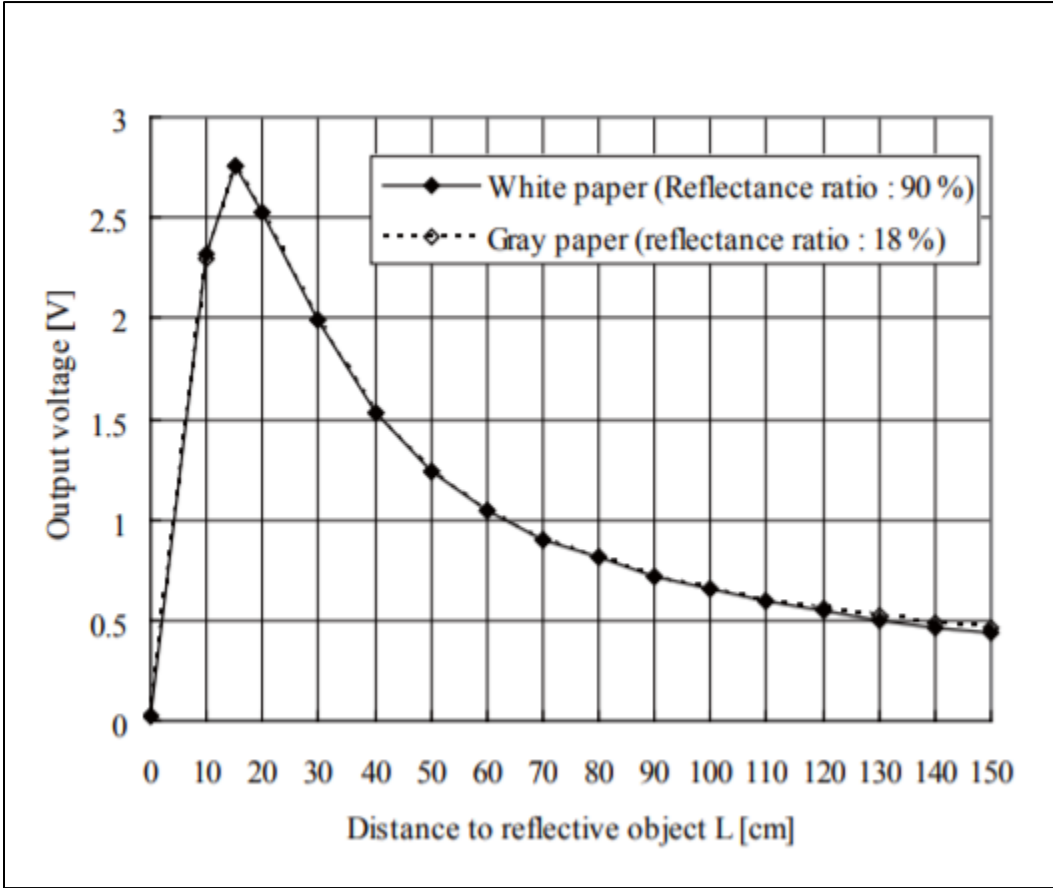


Fig 5: Characteristics of IR sensor as per Datasheet [1]

Servo Motor

Steps in Programming Servo Motor:

- 1) I connected servo to 9th pin, as I used Servo Library for PWM, which disables the analogWrite() (PWM) functionality of pin 9 and 10.
- 2) Servo has three wires, red connected to VCC, Black to ground and Yellow to Digital input (9) in Arduino
- 3) I programmed to map the values in working range (200 – 500) to (0-255).
- 4) These mapped values rotated servo from 0-180 Deg.
- 5) I also calculated distance, using $x = (y-608)/(-23)$, using the linear equation from the graph to display the same in GUI.

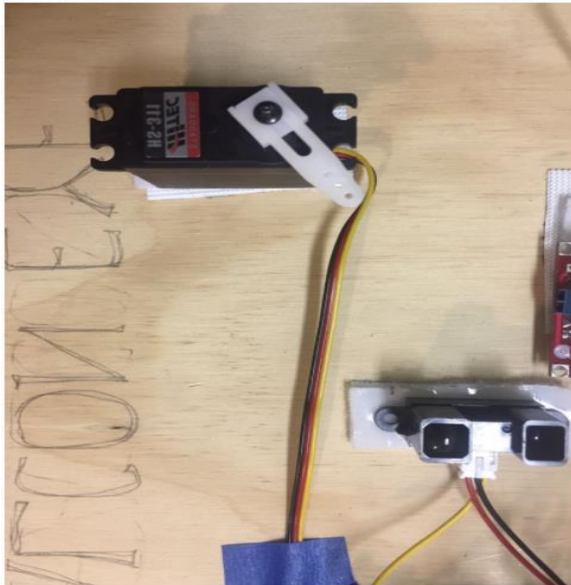


Fig 6: IR Sensor and Servo on Board



Fig 7: Servo Motor^[2]

Hardware Integration

I and Danendra Singh worked together to Place different components together. The components were place so that they don't interfere in each other's working. The wires' length was minimized, so that we don't make wrong connections. Also, we tried to connect red wire to VCC and Black wire for ground on each components' terminals, although due to limited number of connectors we can't do it for all components. The reason for doing it was, initially someone from our team connected black to VCC and red to ground of DC Motor and we connected it to supply like that as we were unaware of that fact, but immediately we realized and corrected it. So, we took care of that while making final board connections.

Also, there is small creative art work done by me at the bottom of the board...it's 4am creativity, when you're done with your left brain and your right brain takes the front seat.

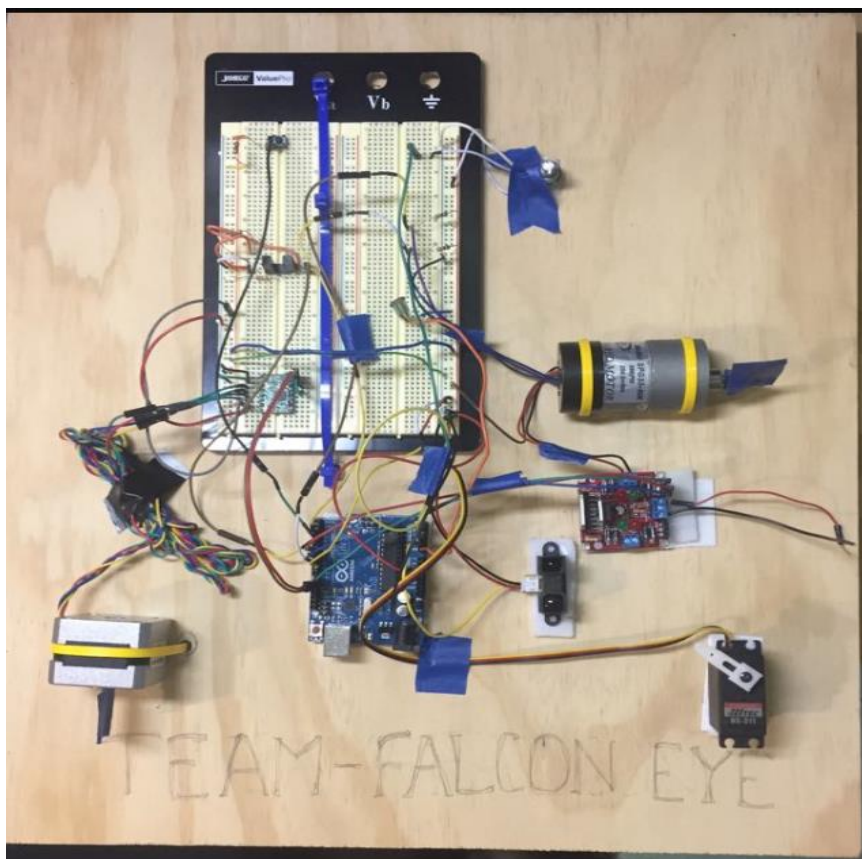


Fig 8: Complete Hardware Integrated Board.

Challenges Faced

I used servo library and as mentioned above, it disabled 9 and 10th pin, but by mistake force sensor to 10th pin. And when we connected whole system together the input from force sensor was coming on serial monitor but the DC Motor was not running. We spend a lot of time in searching whether the use of servo library interferes with the functioning of Force sensor or DC Motor. Because, that seemed most likely reason, as whenever we commented the portion for servo header in code the force sensor and DC Motor combination started to work fine. But later we realized it was a connection problem.

Team Work

Yuchi Wang:	Force sensor, PID control
Pulkit Goyal:	Graphical User Interface
Pratibha Tripathi:	Servo with IR
Danendra Singh:	Stepper and Slot sensor
Rahul Ramakrishnan:	Servo with Potentiometer, PID control

This was the first assignment where whole team had to work together. Everyone working independently on their tasks and then integrating it together. It turned out to be good team work. All the members were working together, so they helped in solving each other's problem, while trying to integrate the hardware. Also, while working on GUI to read data from the system or write the data. We discussed together, what values we want to display, what can be read. We also defined a dictionary, to identify which data is coming from which sensor or motor or it should be written where? Pulkit did an intensive job on making a good GUI and integrating it all together. Yuchi did a great work in integrating all the software. And since Rahul, Danendra and I had completed our part we were supporting them, like in debugging or searching for a function for some piece of task that needs to be done in Qt or just listening to their thought flow as it was helping them to think clearly even after working continuously for long hours. We pulled an all-nighter in the MRSD lab for the first time, with a good morning walk at 5AM. Though it was a bit hectic but it was fun at the same time as we were working together. And above all it ended well, all the pieces ran perfectly during our demonstration.

MRSD Project Progress and Future Plans

Our team is working on two platforms AGV and UAV. For AGV we got a Husky Platform, but the batteries were not working so we ordered them. We have just received the batteries for husky. In the mean time we were working on running the husky in simulation. After getting batteries, we tried to give husky some commands, but we are facing some issues in that. We have a backup option, to get husky from NREC if this Husky doesn't work. So our prime focus is Husky, and we want to make it running first of all.

Also, our sponsor had ordered some drones and they have also arrived. We did some test flights with the drone (Parrot Bepbop 2). We will study about Drone's SDK options. We are also expecting to get a Velodyne Puck but at the same time we are evaluating other options also as back up plan.

References:

- [1] https://www.sparkfun.com/datasheets/Sensors/Infrared/gp2y0a02yk_e.pdf
- [2] <https://www.servocity.com/hs-311-servo>
- [3] <https://learn.adafruit.com/ir-sensor/using-an-ir-sensor>

Task 7 (Sensors and Motor Control Lab) Quiz

1. Reading a datasheet. Refer to the ADXL335 accelerometer datasheet (<https://www.sparkfun.com/datasheets/Components/SMD/adxl335.pdf>) to answer the below questions.
 - What is the sensor's range?
 - $\pm 3g(\text{min})$ and $\pm 3.6g(\text{typ})$
 - What is the sensor's dynamic range?
 - $20\log(0.707*3.6/150\mu\text{g})$
 - What is the purpose of the capacitor C_{DC} on the LHS of the functional block diagram on p. 1? How does it achieve this?
 - A single $0.1\mu\text{F}$ capacitor, C_{DC} , is placed close to the ADXL335 supply pins adequately decouples the accelerometer from noise on the power supply.
 - Write an equation for the sensor's transfer function.
 - $1.5+0.3x$
 - What is the largest expected nonlinearity error in g?
 - $\pm 0.3\%$ of full scale
 - How much noise do you expect in the X- and Y-axis sensor signals when the sensor is excited at 25 Hz?
 - $750\mu\text{g}/\text{rms}$
 - How about at 0 Hz? If you can't get this from the datasheet, how would you determine it experimentally?
 - Set it to 0Hz, calculate the deviation and then convert it to rms.
2. Signal conditioning
 - Filtering
 - What problem(s) might you have in applying a moving average?
 - It does not filter out high outliers. It is bad for frequency encoded signals as it cannot differentiate between two different band of frequencies.
 - What problem(s) might you have in applying a median filter?
 - It is better than moving average filter in removing outliers but the problem is higher computational cost.
 - Opamps
 - In the following questions, you want to calibrate a linear sensor using the circuit in Fig. 1 so that its output range is 0 to 5V. Identify which of V1 and V2 will be the input voltage and which the reference voltage, the value of the reference voltage, and the value of R_f/R_i in each case. If the calibration can't be done with this circuit, explain why.
 - Your uncalibrated sensor has a range of -1.5 to 1.0V.
 - Your uncalibrated sensor has a range of -2.5 to 2.5V.
 - There is no solution. Because it has two equations and three variables.

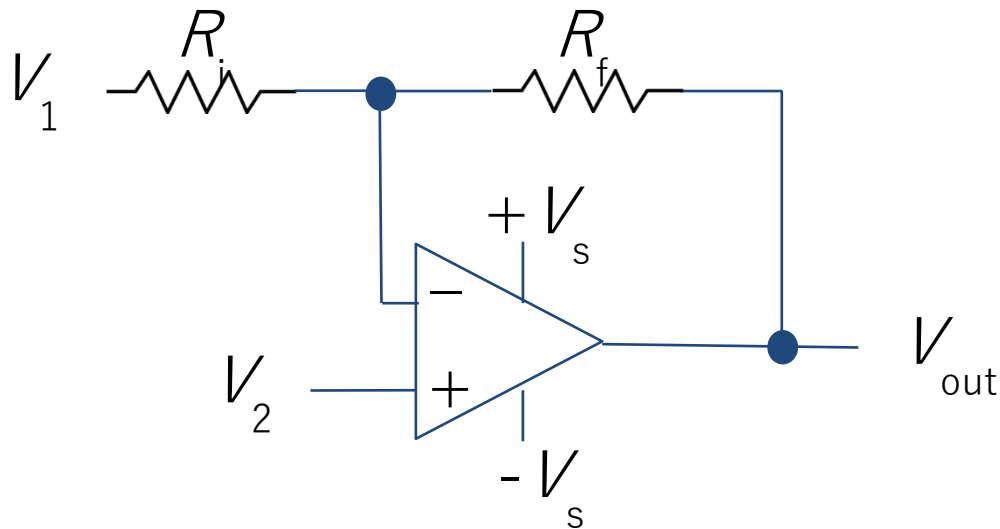


Fig. 1 Opamp gain and offset circuit

3. Control

- If you want to control a DC motor to go to a desired position, describe how to form a digital input for each of the PID (Proportional, Integral, Derivative) terms.
- First we can give proportional term, i.e difference between current location and desired location. Then to further reduce remaining steady state error apply Integral term i.e sum of previous values. Then manage overshoot apply Derivative term, which can be difference of past two error values.
- If the system you want to control is sluggish, which PID term(s) will you use and why?
- In case of a sluggish system, use P term. A high proportional gain term results in high change in output.
- After applying the control in the previous question, if the system still has significant steady-state error, which PID term(s) will you use and why?
- Even after applying P term, if the system still has significant steady state error, apply I term. As integral term accelerates the movement of the process towards setpoint and eliminates remaining steady state error.
- After applying the control in the previous question, if the system still has overshoot, which PID term(s) will you apply and why?
- If the system still overshoot, apply D term. It improves settling time of the system.

Appendix - I

// Snippet of Servo and IR Code

```
void loop_servo() {
  val = analogRead(potpin);          // reads the value of the potentiometer (value between 0 and 1023)
  //Serial.println(val);

  double valr = map(val, 0, 1023, 0, 255);    // scale it to use it with the servo (value between 0 and 180)
  // Serial.print("servo degree: ");
  if (auto_read == 2){
    servo.write(double(val_read)); // sets the servo position according to the scaled value
  }else
  {
    servo.write(double(valr));
  }
  // waits for the servo to get there
}

void loop_ir() {
  val_ir = analogRead(irpin); // reads the value of the potentiometer (value between 0 and 1023)
  distance= ((val_ir-608)/(-23))* 2.54;
  x = (64 *x -10 * (x - val_ir ))/64;
  val_ir2 = map(x, 550, 200, 0, 255); // scale it to use it with the servo (value between 0 and 180)

  if (auto_read == 1)
  {
    if ((val_ir2 < 180) && (val_ir2 > 0)){
      servo.write(val_ir2);
      delay(15);
    }
  }
  else{
    servo.write(val_ir2_read);
    delay(15);
  }
  // sets the servo position according to the scaled value
  // waits for the servo to get there
}}
```