PoE Lab 3 Report

Fankie Devanbu, Cecilia Diehl, Mackenzie Frackleton

October 13, 2015

1 Introduction

During the first two labs, we learned about understanding and using signals and inputs to gain information. The goal of this lab was to use that new knowledge and integrate it with an actuated physical system, in this case a chassis connected to DC motors. Collecting the readings from two infrared reflection (IR) sensors we were able to detect when the sensors came into contact with the black tape and develop a motor control loop that allowed the chassis to follow the black line. We then developed a system to integrate our Arduino and electronics with the chassis in a simple and removable manor.

2 Procedure

2.1 Circuitry

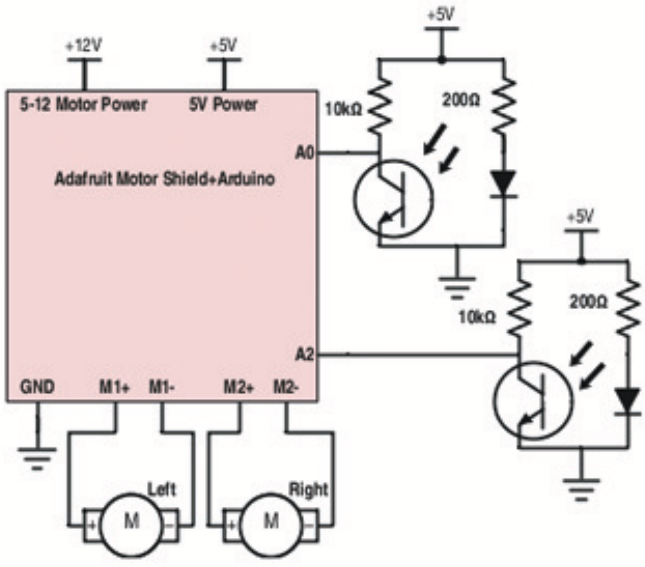

Figure 1: Full Experimental Circuit

We constructed our circuit to mimic the template provided within the Lab 3 protocol. We only needed to calculate the required resistance of the resistor immediately preceding the phototransistor. To do so, we referenced the ideal condition curves on the sensor datasheet.

According to the datasheet, practically any forward current across the LED results in a forward voltage of 1V. Since we put 5V into the system, the voltage drop across the LED's resistor is 4V (as 5V-1V=4V). We substituted voltage and resistance values into $V = iR$, giving us $4 = i200$, and $i = 20mA$ when we solved for current through the LED.

We then referenced the datasheet again and found that, in an ideal system, a forward current of 20mA results in a collector current of 2mA.

Referencing the datasheet one more time, we found that a forward current of 20mA and a collector current of 2mA ceases to be effective at any collector emitter voltage below 1V. Therefore, we assumed the voltage drop across the phototransistor to be 1V and the drop across the associated resistor to be 4V (again, because 5V-1V=4V and we put in 5V). Finally, we substituted 4V and 2mA into $V = iR$ to solve for an ideal resistance of 2,000Ω.However, we empirically determined that a 10kΩ resistor was necessary for a sufficient voltage drop. This process is outlined in the next section.

2.2 Testing and Calibrating Sensor

After we developed an ideal circuit for the sensor, we replicated the ideal test conditions (a 5mm distance from a mirror) as closely as possible. In practice, this was a 1cm distance from the tile floor of the classroom because of our materials limitations.

We decided to test our sensors on the tile floor and the black electrical tape by printing the voltage output of the sensors to our serial port. Unfortunately, the 2kΩ resistor value didn't give any noticeable drop in resistance between the two floorings. We decided to test 3 larger resistance values we found in the stockroom: a 4.53KΩ resistor, a 7.32kΩ resistor, and a 10kΩ resistor.

We finally settled on the 10KΩ resistor, which gave an average output of 613mV on the tile floor and 989mV on the black electrical tape.

2.3 Controller

In order to keep our chassis following the black line, we created a control loop that used input from the IR sensors to determine motor speed. For our loop we used the fairly simple "Bang-Bang" approach implemented on line 39 of our code with the *check* function. Our chassis is always turning left or right, with the exception of the end of the loop, at which point both motors stop completely.

We placed the sensors on either side of the black line and decided to have them continuously report back their voltage output. We found that any output value greater than 900 meant the sensor was on the black electrical tape, so we programmed the chassis to turn away from line as soon as the sensor read over 900. For example, if the left sensor returned a value over 900, the left wheel slows 5 and the right wheel speeds up (this case is in line 46 of our code). The chassis would continue to turn until the condition of another *if* statement within *check* was met. If the right sensor returned and output of over 900, the chassis would begin to turn right, and if both sensors returned 900, both motors would stop completely.

The below figure shows the relationship between sensor output value and motor speed. This data was taken over a short run around one of the black tape course's larger, looping curves. Note that the left y axis describes sensor output in mV while the right y axis describes the speed setting of the dc motors. This speed is a number out of 255 that was input from the arduino environment.

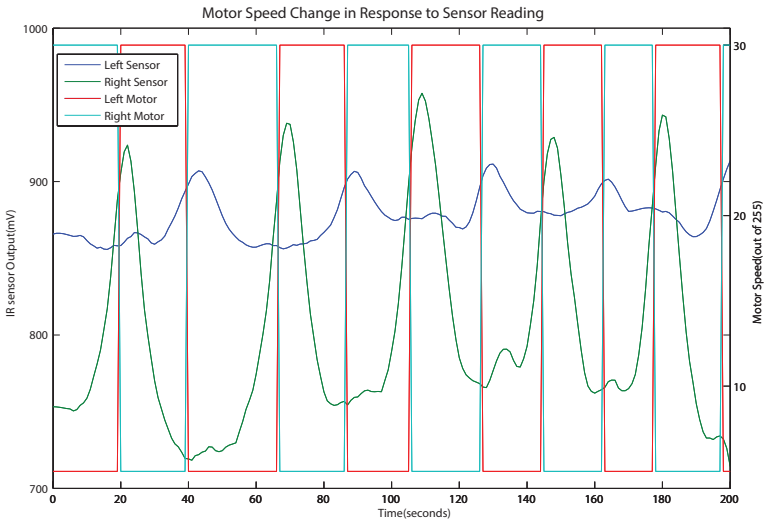

Figure 2: Data Sample taken from 200 seconds of a continuous run of our car

2.4 Chassis Integration

In our design, the chassis was integrated with the Arduino and sensors through tree 3D printed parts, nuts, and bolts. We designed a belly pan, than screwed into the chassis and seated the Arduino in the middle with space for the wires to run between components. Our sensors were attached on the underside with two custom brackets, designed for their shape and to keep them the correct distance from the ground (1 cm). These keep the sensors pointed at a spot close to the center of rotation of the robot, in an effort to reduce error.

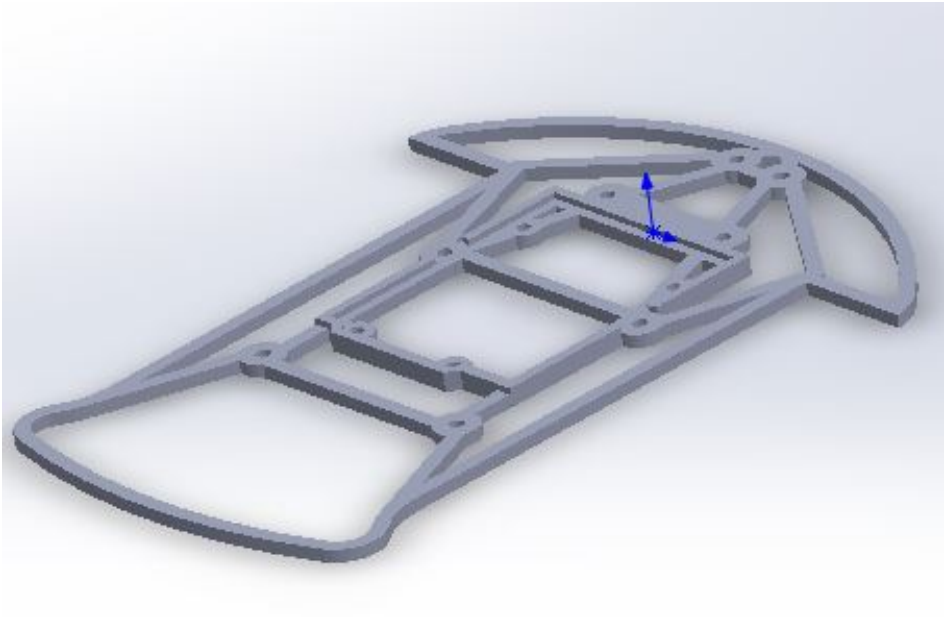

Figure 3: Belly pan that integrates the chassis to the Arduino and motor controller. Designed. to screw into both, with little material. Holes were overlarge and kept still with nuts and bolts. And a ridge helps to keep the Arduino seated

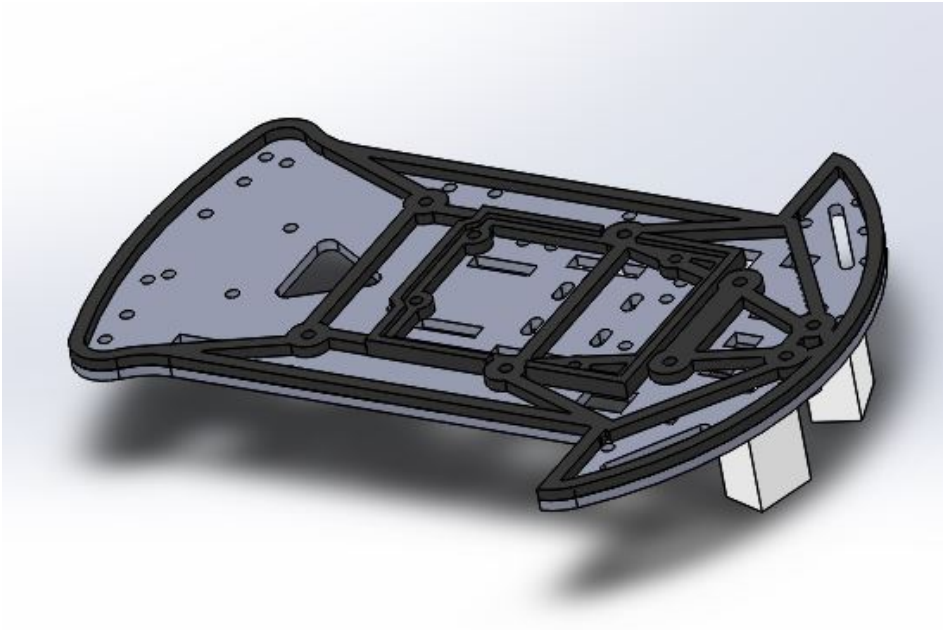

Figure 4: Whole integrated system minus electronics and fasteners. Grey represents the chassis, black our belly pan, and white the sensor brackets.

3 Final Reflection

We found this lab particularly challenging, not necessarily because of the content in the lab, but because of the need to work around set deadlines while dealing with impediments to integrating our system with the chassis. While it is common in industry to have limited time and access to certain components of a project, this is quite challenging to implement as extremely busy students with other classes and commitments to work around. Finding an hour when our entire team was free was a miracle, then to have that hour be useless to us because the chassis was unavailable was incredibly frustrating. Our ability to work together as a team suffered because of this.

That being said, in this assignment we gained experience working in a real world environment where the only prior knowledge given was "ideal conditions" data. We definitely had to think about adapting our sensors to the actual environment we were working in. While experimenting with how the electrical system would react to our changing environment we implemented a simple guess and check method for selecting resistors. As the resistance increased we were able to look at the changes in voltage output and compare it to the sensitivity of our signal readings until we were content with the difference in values between being on and off the tape. Overall this lab was a good way to pull together what we had learned in the earlier two labs while also integrating with a physical system.

4 Source Code

4.1 Arduino Source Cot

```
//calling motor shield from library
#include <Wire.h>
#include <Adafruit_MotorShield.h>
#include "utility/Adafruit_PWMServoDriver.h"

//creating motor shield object
Adafruit_MotorShield AFMS = Adafruit_MotorShield();

//creating DC motor object
Adafruit_DCMotor *MotorR = AFMS.getMotor(1); //Left motor
Adafruit_DCMotor *MotorL = AFMS.getMotor(2); //Right motor

//Setup IRS
const int outputPinR= A2;   // will use L and R to differentiate btwn 2 IRS
const int outputPinL = A0;
const int DefaultSpeed = 20;

int valL = 0;
int valR=0;

void setup() {
    AFMS.begin();
    Serial.begin(9600);
    //default motor speeds to switch if needed
    MotorL->setSpeed(DefaultSpeed+1);
    MotorR->setSpeed(DefaultSpeed);
}

void loop() {
   MotorL->run(BACKWARD);
   MotorR->run(BACKWARD);

   valR= analogRead(outputPinR);
   valL = analogRead(outputPinL);

   check();
}

void check(){
  //if left sensor sees tape, turn left
  if (valL >= 900){ //this sensor value corresponds to black tape
    MotorR->setSpeed(30);
    MotorL ->setSpeed(5);
  }
  //if right sensor sees tape, turn right
    if (valR >= 900){ //this sensor value corresponds to black tape
```

```
      MotorL->setSpeed(30);
      MotorR-> setSpeed(5);
     }
  //if bot sensors see tape, stop
  //this stops the chassis and the endline
   if (valR >= 900 && valL >= 900 ){
     MotorL->setSpeed(0);
     MotorR-> setSpeed(0);
     }
  }
```

4.2 Matlab Plotting Code

```
load snailCurve.csv
dom = 600:800;

%Variables needed for the sgolay filtering technique
windowSize = 11;
b = (1/windowSize)*ones(1,windowSize)
filt2 = @(data) sgolayfilt(data, 3, windowSize);

%breaking down file components into correct vectors
x = filt2(snailCurve(dom,1)); %irL
y = filt2(snailCurve(dom,2));%irR
z= snailCurve (dom,3); %motorL
q = snailCurve (dom,4); %motorR
%create the time vector that corresponds to our 4 vectors of data
s = 0:length(x)-1;

 hold on

 %this plot has 2 axis and a filtered, less noisy reprersentation of IR
 %sensor data
 ax = plotyy([s' s'], [x y], [s' s'], [z q])
 legend('Left Sensor','Right Sensor','Left Motor','Right Motor')
title('Motor Speed Change in Response to Sensor Reading ','fontsize',13)
xlabel('Time')
ylabel(ax(1),'IR sensor Output(mV)')
ylabel(ax(2),'Motor Speed(out of 255)')
ylim(ax(2), [4, 31])
```

5 Data Sheets Cited

Vishay, "Reflective Optical Sensor with Transistor Output," TCRT5000 datasheet, Aug, 17, 2009.