Principles of Integrated Engineering (PIE)

Mini-Project 3 - Section 2: Group 21

Sparsh Gupta, Akshat Jain

Franklin W. Olin College of Engineering

October 15, 2023

Introduction

We built a robot that follows a line path and utilizes DC motors and IR reflectance sensors. The robot is run using Arduino code and the motors are controlled using a motor shield.

Implementation and Functionality

Sensor Calibration

To find the desired R_{sense} value for the IR reflectance sensor, we utilize the following formula:

$$5V = V_{out} + I_c R_{sense}$$
$$V_{out} = 5V - I_c R_{sense}$$

Essentially, we need to obtain the analog voltage values from the IR sensors when they detect the line or the floor, and we need to record the difference in these values for our computations.

The $I_c^{floor} = 0.001A$ and $I_c^{line} = 0.00001A$, so if we choose a value of $100,000\Omega$ for our R_{sense}, we obtain the following values for V_{out}:

$$V_{out}^{floor} = |5V - 0.001A \cdot 100000\Omega| = 95V$$

$$V_{out}^{line} = |5V - 0.00001A \cdot 100000\Omega| = 4V$$

As we can see, the V_{out}^{floor} and V_{out}^{line} vary by a factor of more than 20 and it gives us a reasonable range to observe the differences when the IR sensor detects the floor or the line. Once we tested the IR sensors with this $R_{sense} = 100000\Omega$ value, we obtained analog voltage read values ranging from 200 to 900 depending on if it was detecting the floor or the line, and we decided that this was a perfectly working range for us to do our computations.

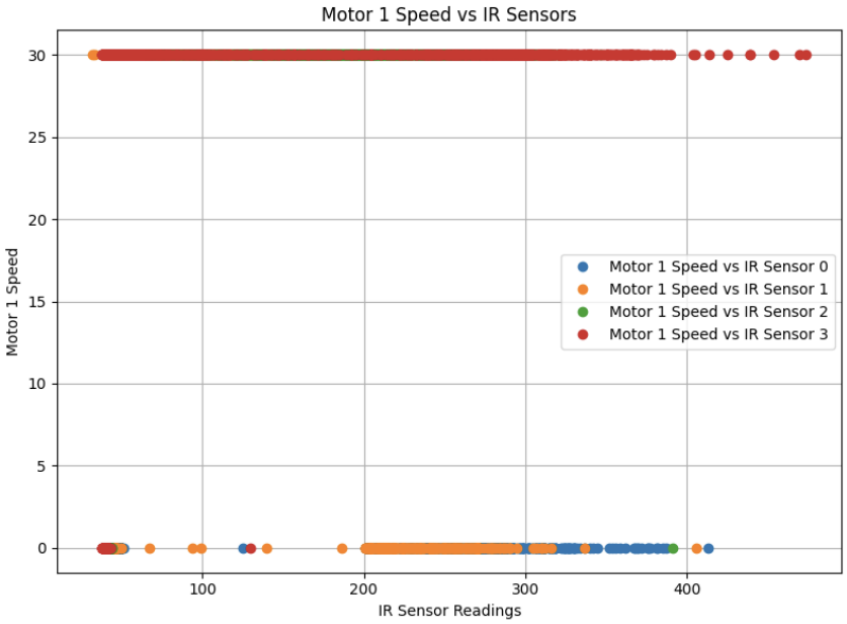
Control Logic

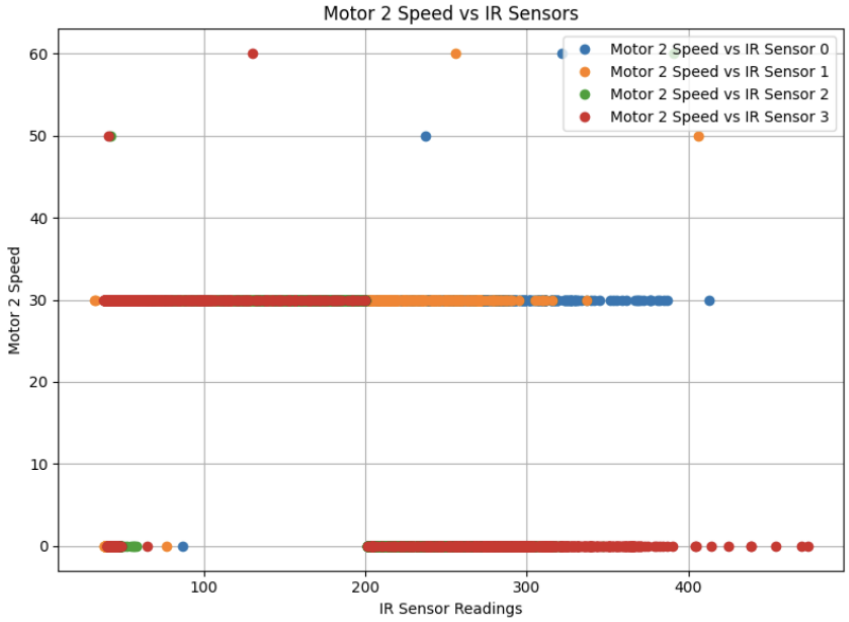
We employ an Adafruit motor shield to govern motor control. The initial motor speed is configured at 30 rpm. The analog sensor output is translated into a boolean result, hinging on a specified cutoff. When all four sensors uniformly report the same value, both motors proceed straight. When a single sensor exclusively detects the line, the opposing motor adjusts to steer the robot. Detection by 2 or 3 sensors implies an imminent sharp turn, prompting the robot to pivot until the leftmost or rightmost off-line sensor registers the line.

Serial Interface

The Serial interface is used to modify motor speeds by sending commands from the Arduino Serial port. For this, we identify if the Serial port is available first and then we can send commands through the Serial monitor command line. The commands we use are "W" to increase the speed by 10 and "S" to decrease the speed by 10.

Plot with motor commands and sensor data





Demo video with line following and behavior change

Line following: https://youtu.be/6WaoDBJH2Vw
Behaviour change: https://youtu.be/TfLmgbF4W9s

Mechanical Design

The chassis of the robotic car is composed of two DC motors attached to Omni-wheels for driving it. The Arduino and the Adafruit Motor Shield is mounted on top of the chassis. A breadboard is mounted on the bottom of the chassis such that the four IR reflectance sensors face the ground and the circuit is built on it. The electrical connections from the breadboard to the motor shield Analog Input pins and 5V/GND pins to obtain the IR sensor V_{out} values and have 5V/GND rails is done using jumper wires.

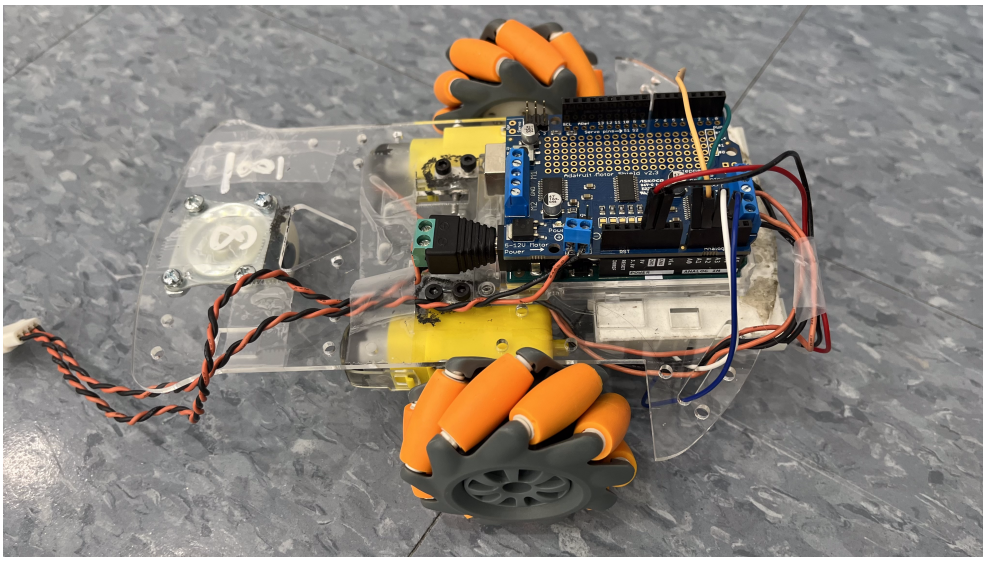


Figure 1: Robot - top view

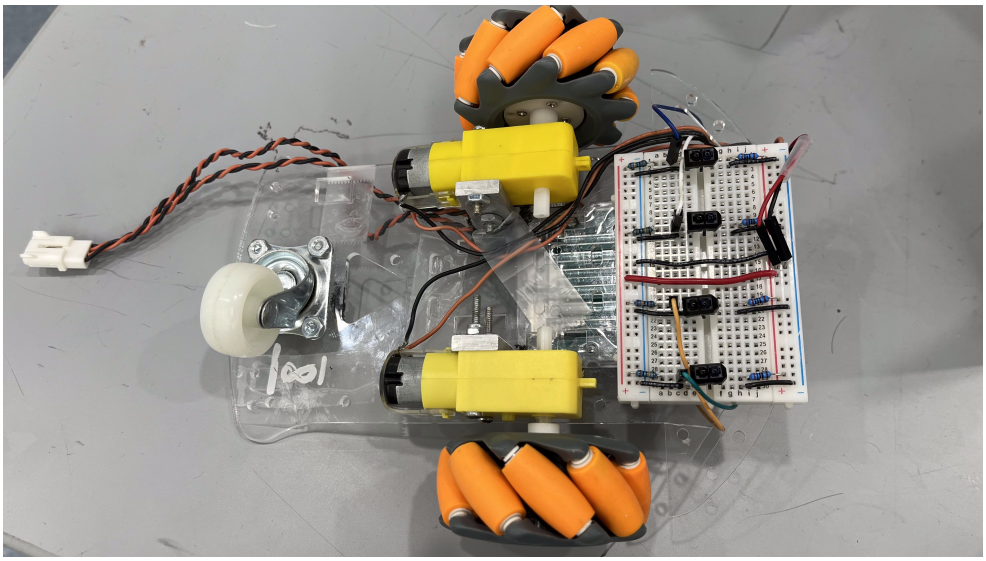


Figure 2: Robot - bottom view

Circuit Diagram

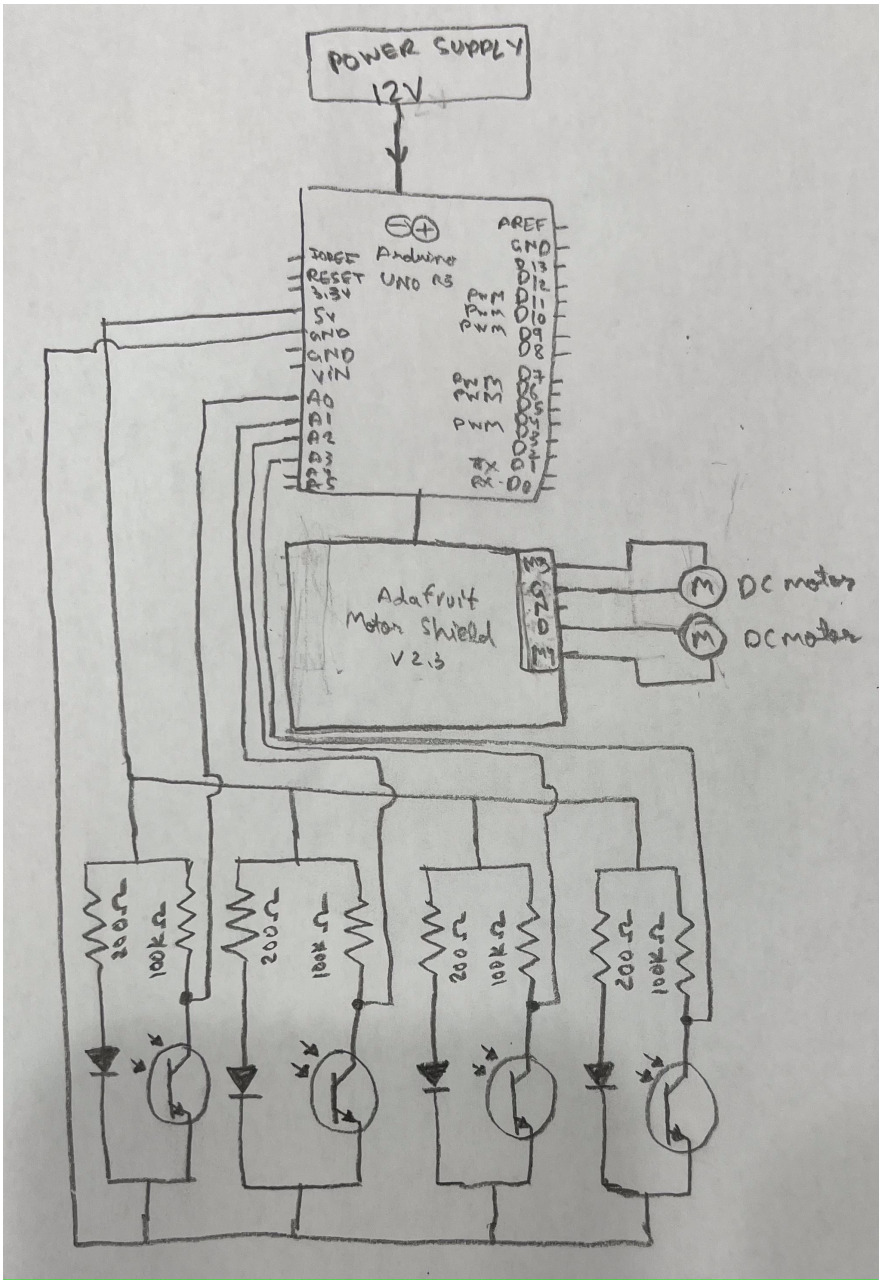


Figure 3: Circuit Diagram

Code

Arduino

```
#include <Adafruit_MotorShield.h>

Adafruit_MotorShield AFMS = Adafruit_MotorShield();
Adafruit_DCMotor *myMotor1 = AFMS.getMotor(3);
Adafruit_DCMotor *myMotor2 = AFMS.getMotor(4);

int IRSensor2 = A0; // IR sensor pin for left-most sensor
int IRSensor3 = A1; // IR sensor pin for middle-left sensor
int IRSensor4 = A2; // IR sensor pin for middle-right sensor
int IRSensor5 = A3; // IR sensor pin for right-most sensor

int cutoffValue = 200;
int speed = 30;

void setup() {

  // Start motor with the default frequency 1.6KHz
  AFMS.begin();

  Serial.begin(9600); // Set Serial Baud Rate

  // Intialize motors
  myMotor1->setSpeed(0);  // Set Motor 1 speed
  myMotor2->setSpeed(0);  // Set Motor 2 speed

  // Initialize IR sensors
  pinMode(IRSensor2, INPUT); // IR Sensor 2 pin INPUT
  pinMode(IRSensor3, INPUT); // IR Sensor 3 pin INPUT
  pinMode(IRSensor4, INPUT); // IR Sensor 4 pin INPUT
  pinMode(IRSensor5, INPUT); // IR Sensor 5 pin INPUT
}

void loop() {
  // IR sensor readings
  int IR2 = analogRead(IRSensor2); // Set the IR Sensor 2 as Input
  int IR3 = analogRead(IRSensor3); // Set the IR Sensor 3 as Input
  int IR4 = analogRead(IRSensor4); // Set the IR Sensor 4 as Input
  int IR5 = analogRead(IRSensor5); // Set the IR Sensor 5 as Input

    // Print sensor readings
  Serial.print(IR2);
  Serial.print(",");
  Serial.print(IR3);
```

```
  Serial.print(",");
  Serial.print(IR4);
  Serial.print(",");
  Serial.print(IR5);
  Serial.print(",");

  // Classify IR sensor readings as 0 or 1 based on cutoffValue
  int IR2state = (IR2 > cutoffValue) ? 1 : 0;
  int IR3state = (IR3 > cutoffValue) ? 1 : 0;
  int IR4state = (IR4 > cutoffValue) ? 1 : 0;
  int IR5state = (IR5 > cutoffValue) ? 1 : 0;

  // Modify motor speeds using serial commands
  if (Serial.available() > 0) {
    char command = Serial.read();
    if (command == 'W') {
      // Increase speed
      speed += 10;
    } else if (command == 'S') {
      // Decrease speed
      speed -= 10;
    }
  }

  if (IR2state==0 & IR3state==0 & IR4state==0 & IR5state==0){
    int motor1Speed = speed;
    int motor2Speed = speed;

    myMotor1->setSpeed(motor1Speed);
    myMotor2->setSpeed(motor2Speed);

    Serial.print(motor1Speed);
    Serial.print(",");
    Serial.println(motor2Speed);
  }
  if (IR2state==1 & IR3state==1 & IR4state==1 & IR5state==1){
    int motor1Speed = speed;
    int motor2Speed = speed;

    myMotor1->setSpeed(motor1Speed);
    myMotor2->setSpeed(motor2Speed);

    Serial.print(motor1Speed);
    Serial.print(",");
    Serial.println(motor2Speed);
  }
```

```
  else if (IR2state==0 & IR3state==0 & IR4state==1 & IR5state==0){
    int motor1Speed = speed;
    int motor2Speed = 0;

    myMotor1->setSpeed(motor1Speed);
    myMotor2->setSpeed(motor2Speed);

    Serial.print(motor1Speed);
    Serial.print(",");
    Serial.println(motor2Speed);
  }

  else if (IR2state==0 & IR3state==0 & IR4state==0 & IR5state==1){
    int motor1Speed = speed;
    int motor2Speed = 0;

    myMotor1->setSpeed(motor1Speed);
    myMotor2->setSpeed(motor2Speed);

    Serial.print(motor1Speed);
    Serial.print(",");
    Serial.println(motor2Speed);
  }

  else if (IR2state==0 & IR3state==1 & IR4state==0 & IR5state==0){
    int motor1Speed = 0;
    int motor2Speed = speed;

    myMotor1->setSpeed(motor1Speed);
    myMotor2->setSpeed(motor2Speed);

    Serial.print(motor1Speed);
    Serial.print(",");
    Serial.println(motor2Speed);
  }

  else if (IR2state==1 & IR3state==0 & IR4state==0 & IR5state==0){
    int motor1Speed = 0;
    int motor2Speed = speed;

    myMotor1->setSpeed(motor1Speed);
    myMotor2->setSpeed(motor2Speed);

    Serial.print(motor1Speed);
    Serial.print(",");
    Serial.println(motor2Speed);
```

```
  }

  else if (IR2state==1 & IR3state==1 & IR4state==1 & IR5state==0){
    int motor1Speed = 0;
    int motor2Speed = speed+30;

    Serial.print(motor1Speed);
    Serial.print(",");
    Serial.println(motor2Speed);

    while (IR5state==0){
      IR5 = analogRead(IRSensor5); // Set the IR Sensor 5 as Input
      IR5state = (IR5 > cutoffValue) ? 1 : 0;
      myMotor1->setSpeed(motor1Speed);
      myMotor2->setSpeed(motor2Speed);
    }
    myMotor1->setSpeed(speed);
    myMotor2->setSpeed(0);
    delay(200);
  }

  else if (IR2state==1 & IR3state==1 & IR4state==0 & IR5state==0){
    int motor1Speed = 0;
    int motor2Speed = speed+20;

    Serial.print(motor1Speed);
    Serial.print(",");
    Serial.println(motor2Speed);

    while (IR5state==0){
      IR5 = analogRead(IRSensor5); // Set the IR Sensor 5 as Input
      IR5state = (IR5 > cutoffValue) ? 1 : 0;
      myMotor1->setSpeed(motor1Speed);
      myMotor2->setSpeed(motor2Speed);
    }

    myMotor1->setSpeed(speed);
    myMotor2->setSpeed(0);
    delay(200);
  }

  // Run motors
  myMotor1->run(FORWARD);
  myMotor2->run(FORWARD);
}
```

Python Plotting code

```
import serial
import matplotlib.pyplot as plt

arduinoComPort = "/dev/cu.usbmodem12101"
baudRate = 9600

serialPort = serial.Serial(arduinoComPort, baudRate, timeout=1)

# Initialize data lists
motor1_speeds = []
motor2_speeds = []
sensor0_readings = []
sensor1_readings = []
sensor2_readings = []
sensor3_readings = []

# Set the maximum number of readings
max_readings = 4000
reading_count = 0

# Main loop to read data from the Arduino, then display it
while reading_count < max_readings:
    lineOfData = serialPort.readline().decode()

    if "end\r\n" in lineOfData:
        break

    elif len(lineOfData) > 0:
        print(lineOfData)
        data = list(map(float, lineOfData.split(',')))

        sensor0_readings.append(data[0])
        sensor1_readings.append(data[1])
        sensor2_readings.append(data[2])
        sensor3_readings.append(data[3])
        motor1_speeds.append(data[4])
        motor2_speeds.append(data[5])

        reading_count += 1

# Create subplots for Motor 1 speed vs all IR sensors and Motor 2 speed vs
    all IR sensors
plt.figure(figsize=(16, 6))

# Motor 1 Speed vs IR Sensor Data
plt.subplot(1, 2, 1)
plt.plot(sensor0_readings, motor1_speeds, 'o', label='Motor 1 Speed vs IR
   Sensor 0')
plt.plot(sensor1_readings, motor1_speeds, 'o', label='Motor 1 Speed vs IR
```

```
    Sensor 1')
plt.plot(sensor2_readings, motor1_speeds, 'o', label='Motor 1 Speed vs IR
    Sensor 2')
plt.plot(sensor3_readings, motor1_speeds, 'o', label='Motor 1 Speed vs IR
    Sensor 3')
plt.xlabel('IR Sensor Readings')
plt.ylabel('Motor 1 Speed')
plt.legend()
plt.title('Motor 1 Speed vs IR Sensors')
plt.grid(True)

# Motor 2 Speed vs IR Sensor Data
plt.subplot(1, 2, 2)
plt.plot(sensor0_readings, motor2_speeds, 'o', label='Motor 2 Speed vs IR
    Sensor 0')
plt.plot(sensor1_readings, motor2_speeds, 'o', label='Motor 2 Speed vs IR
    Sensor 1')
plt.plot(sensor2_readings, motor2_speeds, 'o', label='Motor 2 Speed vs IR
    Sensor 2')
plt.plot(sensor3_readings, motor2_speeds, 'o', label='Motor 2 Speed vs IR
    Sensor 3')
plt.xlabel('IR Sensor Readings')
plt.ylabel('Motor 2 Speed')
plt.legend()
plt.title('Motor 2 Speed vs IR Sensors')
plt.grid(True)

plt.tight_layout()
plt.show()
```

Reflection

Software: We started by using the inbuilt PID library for the project but then faced problems with the library so switched to simple logic and conditional statements for the code.

Electrical: We faced several electrical problems because we unfortunately got defective motors and motor shield. SO, we had to replace the DC motors four times to finally obtain motors which were working. When using the motor shield, our Serial communication was not working because we would stop getting IR sensor data as soon as we would keep the robot on the ground because the sensors were connected to A2 through A5, and we found out that the shield uses the SDA and SCL i2c pins on pins A4 and A5 to control DC and stepper motors and that was causing this problem. So, we switched to using A0 and A1 instead after removing stuff that was blocking those pins in the motor shield.

Mechanical: The mechanical design was pretty simple where we attached a breadboard with sensors on the bottom of the robot, and the motors were attached using plates to the chassis of the robot. Because the old motors that we were using were defective, one of the motors in this process started smoking because it was unable to rotate. But, everything else was very smooth in terms of the mechanical aspect.

Teaming: The teaming was pretty smooth and we contributed equally to the project while achieving our learning goals for the project. We both did software, electrical and mechanical things equally as well.