CPE476 – Mobile Robotics

Design Assignment #1

Name: Neel Patel

Email: pateln3@unlv.nevada.edu

Github Repository link (root): https://github.com/neelpatel114/submissions.git

Youtube Playlist link (root): https://youtube.com/playlist?list=PLjhbM6__bgV_OnArIwnmPu7-PxkAD8iHn

Overview of Robot:

Parts used:

- 4 TT Motors with encoders
- Adafruit Feather Board Motor Driver
- Teensy 4.0
- Battery Holders
- Batteries
- Buck Converter
- Chassis

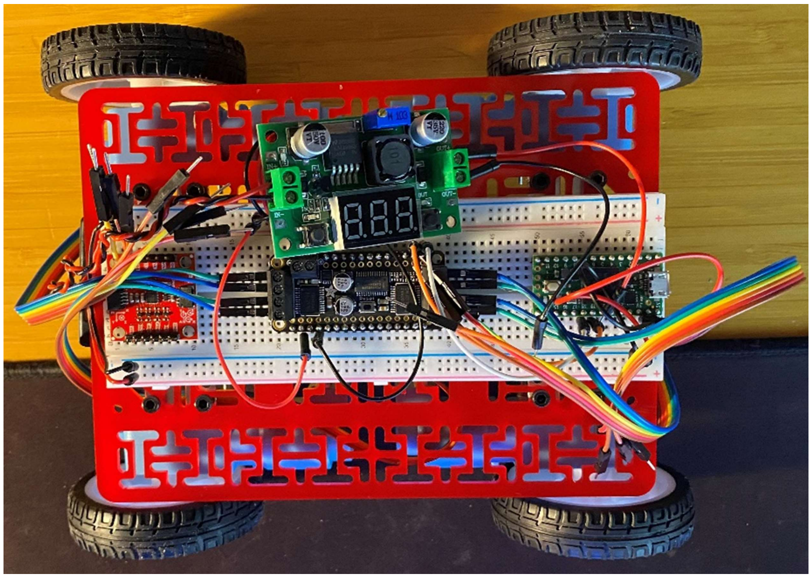

Figure 1 Picture of Assembled Robot

Task 1: Straight Line

This task was to assemble the modular robot and have it move in a straight line for 2-5 meters. My target distance was 2.5 meters and I was able to get the robot to reach near that distance.

Trail	Results (meters)
1	2.8
2	2.6
3	2.7
4	2.4

$$\text{RMSE} = \sqrt{[\ \Sigma(P_i - O_i)^2 / n\ ]} = 0.64226$$

Task 1: Straight Line Code

```
#include <Adafruit_MotorShield.h>

// Create the motor shield object with the default I2C address
Adafruit_MotorShield AFMS = Adafruit_MotorShield();
// Or, create it with a different I2C address (say for stacking)
// Adafruit_MotorShield AFMS = Adafruit_MotorShield(0x61);

// Select which 'port' M1, M2, M3 or M4. In this case, M1
Adafruit_DCMotor *myMotor = AFMS.getMotor(1);
Adafruit_DCMotor *myMotorTwo = AFMS.getMotor(2);
Adafruit_DCMotor *myMotorThree = AFMS.getMotor(3);
Adafruit_DCMotor *myMotorFour = AFMS.getMotor(4);
// You can also make another motor on port M2
//Adafruit_DCMotor *myOtherMotor = AFMS.getMotor(2);

void setup() {
       Serial.begin(9600);           // set up Serial library at 9600 bps
       Serial.println("Adafruit Motorshield v2 - DC Motor test! - Stright");

       if (!AFMS.begin()) {         // create with the default frequency 1.6KHz
              // if (!AFMS.begin(1000)) {  // OR with a different frequency, say 1KHz
              Serial.println("Could not find Motor Shield. Check wiring.");
              while (1);
       }
       Serial.println("Motor Shield found.");

       // Set the speed to start, from 0 (off) to  (max speed)
       myMotor->setSpeed(150);
       myMotorTwo->setSpeed(150);
       myMotorThree->setSpeed(150);
       myMotorFour->setSpeed(150);

       myMotor->run(FORWARD);
       myMotorTwo->run(FORWARD);
```

```
        myMotorThree->run(FORWARD);
        myMotorFour->run(FORWARD);
        // turn on motor
        myMotor->run(RELEASE);
        myMotorTwo->run(RELEASE);
        myMotorThree->run(RELEASE);
        myMotorFour->run(RELEASE);
}

void loop() {
        uint8_t i;

        Serial.print("tick");

        myMotor->run(FORWARD);
        for (i=0; i<; i++) {
                myMotor->setSpeed(i);
                myMotorTwo->setSpeed(i);
                myMotorThree->setSpeed(i);
                myMotorFour->setSpeed(i);
                delay(10);
        }
        for (i=; i!=0; i--) {
                myMotor->setSpeed(i);
                myMotorTwo->setSpeed(i);
                myMotorThree->setSpeed(i);
                myMotorFour->setSpeed(i);
                delay(10);
        }

        Serial.print("tock");

        myMotor->run(BACKWARD);
        myMotorTwo->run(BACKWARD);
        myMotorThree->run(BACKWARD);
        myMotorFour->run(BACKWARD);
        for (i=0; i<255; i++) {
                myMotor->setSpeed(i);
                myMotorTwo->setSpeed(i);
                myMotorThree->setSpeed(i);
                myMotorFour->setSpeed(i);
                delay(10);
        }
        for (i=255; i!=0; i--) {
                myMotor->setSpeed(i);
                myMotorTwo->setSpeed(i);
                myMotorThree->setSpeed(i);
                myMotorFour->setSpeed(i);
                delay(10);
        }

        Serial.print("tech");
        myMotor->run(RELEASE);
        myMotorTwo->run(RELEASE);
        myMotorThree->run(RELEASE);
        myMotorFour->run(RELEASE);
        delay(1000);
}
```

Task 2: Square

This task was more difficult than the first because it required a singular turn. I began to modify the code above to make the two left wheels go in reverse and then make the two right wheels go forward but I ran into an issue of the weight of the robot being too much for the it to turn in opposing directions. I solved this by increasing the motor speed. The size for the square was not consistent to measure but it did stay within a 2 meter by 2 meter box of tape I used to measure the box. I measure by taking a measurement before the circle began from the center and then took a measurement at the end. Below I used the distance at the end.

Trial	Circle
1	2.3
2	2.1
3	1.9
4	1.9

$$\text{RMSE} = \sqrt{[\ \Sigma(P_i - O_i)^2 / n\ ]} = 0.17321$$

Task 2: Square Code

```
#include <Adafruit_MotorShield.h>

// Create the motor shield object with the default I2C address
Adafruit_MotorShield AFMS = Adafruit_MotorShield();
// Or, create it with a different I2C address (say for stacking)
// Adafruit_MotorShield AFMS = Adafruit_MotorShield(0x61);

// Select which 'port' M1, M2, M3 or M4. In this case, M1
Adafruit_DCMotor *myMotor = AFMS.getMotor(1);
Adafruit_DCMotor *myMotorTwo = AFMS.getMotor(2);
Adafruit_DCMotor *myMotorThree = AFMS.getMotor(3);
Adafruit_DCMotor *myMotorFour = AFMS.getMotor(4);
// You can also make another motor on port M2
//Adafruit_DCMotor *myOtherMotor = AFMS.getMotor(2);

void setup() {
  Serial.begin(9600);           // set up Serial library at 9600 bps
  Serial.println("Adafruit Motorshield v2 - DC Motor test! - Stright");

  if (!AFMS.begin()) {         // create with the default frequency 1.6KHz
  // if (!AFMS.begin(1000)) {  // OR with a different frequency, say 1KHz
    Serial.println("Could not find Motor Shield. Check wiring.");
    while (1);
  }
  Serial.println("Motor Shield found.");

  Serial.println("Turn Right.");
  // Set the speed to start, from 0 (off) to 255 (max speed)
  myMotor->setSpeed(255);
  myMotorTwo->setSpeed(255);
  myMotorThree->setSpeed(255);
  myMotorFour->setSpeed(255);
  myMotor->run(FORWARD);
  myMotorTwo->run(FORWARD);
  myMotorThree->run(FORWARD);
  myMotorFour->run(FORWARD);
  // turn on motor
  myMotor->run(RELEASE);
  myMotorTwo->run(RELEASE);
  myMotorThree->run(RELEASE);
  myMotorFour->run(RELEASE);

}

void loop() {
  uint8_t i;

  Serial.println("straight");

  myMotor->run(FORWARD);
  myMotorTwo->run(FORWARD);
  myMotorThree->run(FORWARD);
  myMotorFour->run(FORWARD);

  for (i=0; i<255; i++) {
    myMotor->setSpeed(i);
    myMotorTwo->setSpeed(i);
    myMotorThree->setSpeed(i);
    myMotorFour->setSpeed(i);
    delay(25);
  }
  for (i=255; i!=0; i--) {
    myMotor->setSpeed(i);
    myMotorTwo->setSpeed(i);
    myMotorThree->setSpeed(i);
    myMotorFour->setSpeed(i);
    delay(25);
  }

  Serial.println("turn");

  myMotor->run(BACKWARD);
  myMotorTwo->run(FORWARD);
  myMotorThree->run(BACKWARD);
  myMotorFour->run(FORWARD);

  for (i=0; i<18; i++) {
    myMotor->setSpeed(255);
    myMotorTwo->setSpeed(255);
    myMotorThree->setSpeed(255);
    myMotorFour->setSpeed(255);
    delay(100);
  }

}
```

Task 3: Circle

The circle task was simpler than the right turn due to both set of motors being able to turn in the same direction. For the code for this task I modified the speed of the left side to be slower than the right side so it would result in a constant turn and create a circle. I measured this task by seeing where it fell at every quarter turn within the 2x2 meter circle

Trial	Circle
1	1.6
2	1.4
3	1.7
4	1.9

$$\text{RMSE} = \sqrt{[\ \Sigma(P_i - O_i)^2 / n\ ]} = 0.39370$$

Task 3: Circle Code

```
#include <Adafruit_MotorShield.h>

// Create the motor shield object with the default I2C address
Adafruit_MotorShield AFMS = Adafruit_MotorShield();
// Or, create it with a different I2C address (say for stacking)
// Adafruit_MotorShield AFMS = Adafruit_MotorShield(0x61);

// Select which 'port' M1, M2, M3 or M4. In this case, M1
Adafruit_DCMotor *myMotor = AFMS.getMotor(1);
Adafruit_DCMotor *myMotorTwo = AFMS.getMotor(2);
Adafruit_DCMotor *myMotorThree = AFMS.getMotor(3);
Adafruit_DCMotor *myMotorFour = AFMS.getMotor(4);
// You can also make another motor on port M2
//Adafruit_DCMotor *myOtherMotor = AFMS.getMotor(2);

void setup() {
Serial.begin(9600);           // set up Serial library at 9600 bps
Serial.println("Adafruit Motorshield v2 - DC Motor test! - Stright");

if (!AFMS.begin()) {         // create with the default frequency 1.6KHz
// if (!AFMS.begin(1000)) {  // OR with a different frequency, say 1KHz
Serial.println("Could not find Motor Shield. Check wiring.");
while (1);
}
Serial.println("Motor Shield found.");

Serial.println("Turn Right.");
// Set the speed to start, from 0 (off) to 255 (max speed)
myMotor->setSpeed(255);
myMotorTwo->setSpeed(255);
myMotorThree->setSpeed(255);
myMotorFour->setSpeed(255);
myMotor->run(FORWARD);
myMotorTwo->run(FORWARD);
myMotorThree->run(FORWARD);
myMotorFour->run(FORWARD);
// turn on motor
myMotor->run(RELEASE);
myMotorTwo->run(RELEASE);
myMotorThree->run(RELEASE);
myMotorFour->run(RELEASE);

}

void loop() {
uint8_t i;

Serial.println("Circle");

myMotor->run(BACKWARD);
myMotorTwo->run(FORWARD);
myMotorThree->run(BACKWARD);
myMotorFour->run(FORWARD);

myMotor->setSpeed(100);
myMotorTwo->setSpeed(200);
myMotorThree->setSpeed(100);
myMotorFour->setSpeed(200);

}
```

Conclusion:

This assignment did have a learning curve, but it was still a great experience being able to finally use everything together in one project. I am interested to see how the other components will improve upon this design.

I understand the Student Academic Misconduct Policy -
http://studentconduct.unlv.edu/misconduct/policy.html

“This assignment submission is my own, original work”.
Neel Patel