CPE476 – Mobile Robotics

Midterm Assignment

DO NOT REMOVE THIS PAGE DURING SUBMISSION:

Name: Neel Patel

Email: pateln3@unlv.nevada.edu

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

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

Assignment Description:

Find the position of robot as (x, y, theta) - say for example (you can choose your own maneuvers) you start at (0,0,90), then give a command to go straight at vr and vl = 100 (motor speed) for 3 sec, then give a command to turn at vr = 100 and vl = 150 for 1.5 sec, then a command to turn at vr = 125 and vl = 100 for 2.5 sec? Where is your robot (x_a, y_a, theta_a) from (0,0,90)? By applying the Motion Equations where should be the Robot (x_t, y_t, theta_t)? What is the error at the end of each maneuver (no RMSE)?

Picture of Robot:

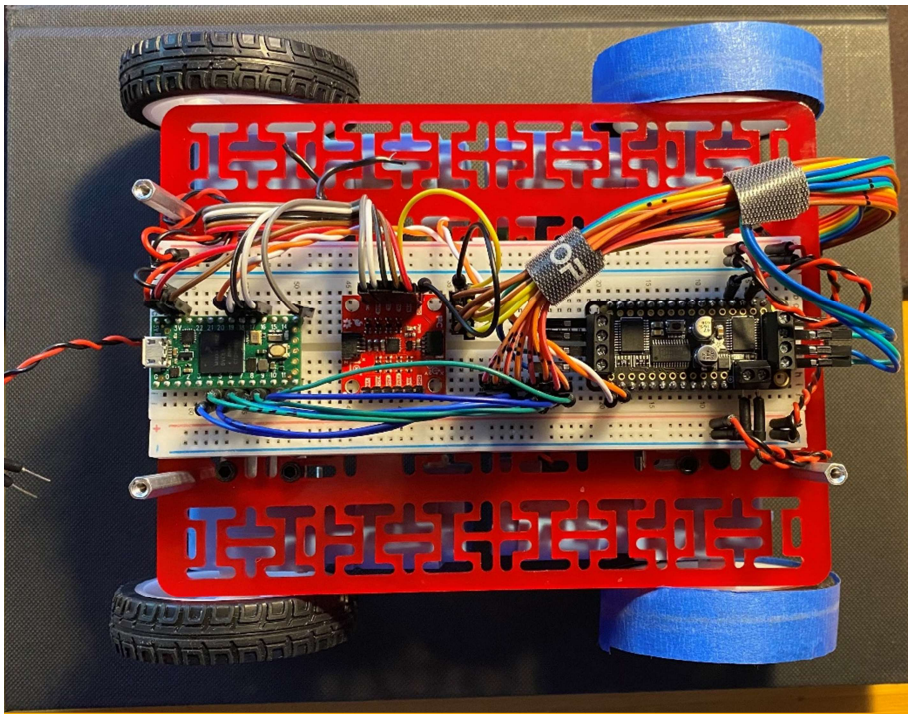
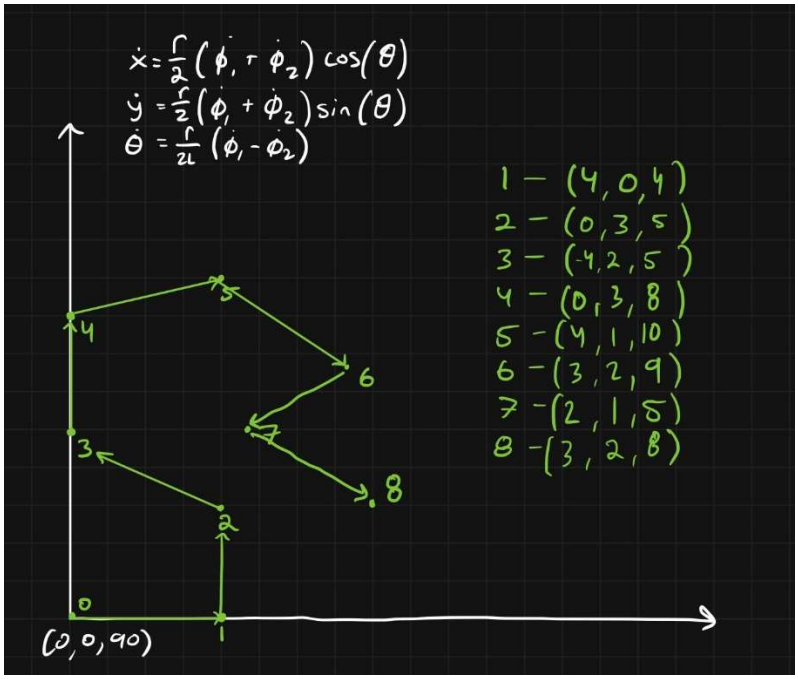


Figure 1 Picture of Robot

Pin Connections:

- Motors
  - Motor Power (Blue and Green) -> Motor Drive Ports M1-4.
  - Motor GND -> GND bus.
  - Motor A -> Teensy Pin 11,9,7, and 5.
  - Motor B -> Teensy Pin 10,8,6, and 4.
    Motor 3V Encoder Power -> 3.3V bus from Teensy
- ICM-20948
  - GND -> Teensy GND
  - VIN -> Teensy 3V output Pin
  - DA -> Teensy SDA1 (Pin 17)
  - Cl -> Teensy SCL1 (Pin16)
- Motor Driver (Adafruit)
  - I2C Data Pins -> SCL0 (Pin 19) and SDA0 (Pin 18)
  - Motor Power Pins -> Power Bus
  - Logic Power Pins -> Power Bus
- Teensy
  - VIN -> Power Bus
  - GND -> Power Bus

Robot Position on Path:



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);

//Global Vars
int pin_1A = 11;
int pin_1B = 10;
int pin_2A = 9;
int pin_2B = 8;
int pin_3A = 7;
int pin_3B = 6;
int pin_4A = 5;
int pin_4B = 4;


void forward(int time,int speed){
	myMotor->run(FORWARD);
	myMotorTwo->run(FORWARD);
	myMotorThree->run(FORWARD);
	myMotorFour->run(FORWARD);

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

void backward(int time,int speed){
	myMotor->run(BACKWARD);
	myMotorTwo->run(BACKWARD);
	myMotorThree->run(BACKWARD);
	myMotorFour->run(BACKWARD);

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

void right(int time,int speed){
```

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

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

}

void left(int time,int speed){
        myMotor->run(BACKWARD);
        myMotorTwo->run(FORWARD);
        myMotorThree->run(BACKWARD);
        myMotorFour->run(FORWARD);

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

}

void stop(){
        myMotor->run(RELEASE);
        myMotorTwo->run(RELEASE);
        myMotorThree->run(RELEASE);
        myMotorFour->run(RELEASE);
}

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 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() {
        delay(5000);
        //F1
        Serial.println("Forward");
        forward(15,255);
        stop();
        //L2
        Serial.println("Left");
        left(10,255);
        stop();
        //F3
        Serial.println("Forward");
        forward(15,255);
        stop();
        //L4
        Serial.println("Left");
        left(10,255);
        stop();
        //F5
        Serial.println("Forward");
        forward(15,255);
        stop();
        //R6
        Serial.println("Right");
        right(12,255);
        stop();
        //F7
        Serial.println("Forward");
        forward(15,255);
        stop();
        //R8
        Serial.println("Right");
        right(12,255);
        stop();
        //F9
        Serial.println("Forward");
        forward(15,255);
        stop();
        //R10
        Serial.println("Right");
        right(12,255);
        stop();
        //F11
        Serial.println("Forward");
        forward(15,255);
        stop();
        //R12
        Serial.println("Right");
```

```
        right(12,255);
        stop();
        //F13
        Serial.println("Forward");
        forward(15,255);
        stop();
        //L14
        Serial.println("Left");
        left(10,255);
        stop();
        //F15
        Serial.println("Forward");
        forward(15,255);
        stop();
        //L16
        Serial.println("Left");
        left(10,255);
        stop();


}
```

Implementation:

During this project I was able to add additional functionality to the robot from previous assignments. I successfully connected the motor encoder to the necessary components and was able to use the output to determine the motors position. I was also able to connect the IMU to the board. The robot was able to follow a path with efficient accuracy. A major issue that occurred was wheel slippage. This was especially impactful on the front two wheels which had a significant impact on the robot's ability to turn. I was able to minimize the impact of the front two wheels on turning by putting tape around the wheels to allow them to slide efficiently. This made turning much more accurate. The IMU also provided significant improvement on the non IMU implementation. I look forward to adding additional functionality using the Jetson Nano.

"This assignment submission is my own, original work".

- Neel Patel