Project CTE Document for Midterm 2

ELE494-08
April 14, 2019

Nasir Khalid
B00065082

I. Goal Statement

We aim to develop an autonomous robot that can survey a given location and return the point of maximum light intensity within that location. As it moves it will be sending back data in real time to a web browser on a phone/computer.

II. Objective

The robot will be given a predefined map and it will be given it's starting position within that map. After this it will begin planning the path it will take around this location and during it's journey it will read the light intensity of the points. As it does this it will be sending back data of it's current position on the map & the light intensity at its position. All of this will be visualized so we can watch it in real time. Once it's journey is complete it will go back and remain idle at the point which it determined to be the brightest.

III. Hardware

The main components of this device are:

1) Robot Body

Figure 1: Chassis, wheels and motors of robot [1]

2) NodeMCU Microcontroller

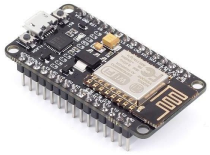

Figure 2: ESP8266 based NODEMCU microcontroller [2]

3) H-Bridge

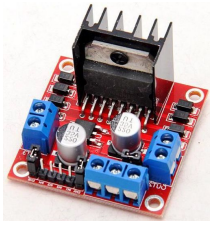

Figure 3: Dual H-Bridge motor controllers [3]

4) Speed Encoders

Figure 4: HC-020K Speed Measuring Module [4]

5) Power Bank

Figure 5: Huawei 6700 mAH power bank [5]

The final components (6 and 7) we currently do not have but we plan to purchase them soon

6) Accelerometer

Figure 6: MPU6050 3 axis accelerometer/gyroscope [6]

7) Light Dependent Resistor

Figure 7: Photoresistor LDR CDS 5mm [7]

IV. Plan of Action

A. Individual Tests

We will begin by testing each sensor independently with the NODEMCU microcontroller and try to get their output to display on a web server that can be accessed by phone or mobile. This will help us ensure that each sensor works properly and that we are using the right functions to interface with them

B. Joint Tests

After the individual tests are complete we will wire all the components together on a breadboard and create a circuit diagram for the entire system. Through this test it will be clear that all the components are working together and during this time we will be writing the preliminary code for the system.

C. Assembly

Now we will assemble all the components on to the chassis of the robot and execute the same preliminary code written in the previous section to ensure that it is operational and that wiring was correctly done.

D. Early Implementation

Now that the robot is ready we will modify it's code heavily to implement the following:

1) Path planning based on a given map
2) Kalman filtration for position
3) Visualization of position
4) Real time graphing of filtered and unfiltered data
5) All movement and robot control functions

E. Final Implementation

After the early testing stage we will implement the final part of the robot which is all the logic and control to identify the brightest spot in a given location.

V. Results

A. Current Results

As of now we have individually tested and jointly tested the sensors we already own Currently the testing includes interfacing the microcontroller with the encoder, h-bridge, and activating the webserver to see data in real time. Shown below is the current set up for joint testing and also the results we obtained from the web server based on testing of the encoders.

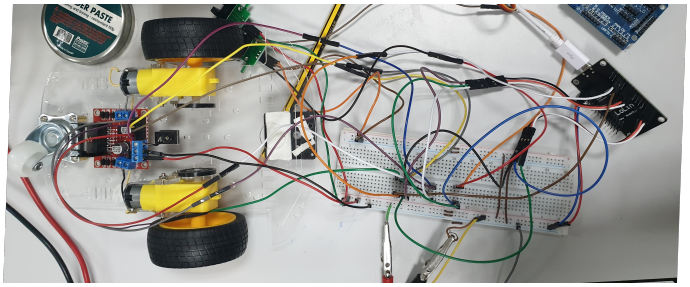

Figure 8: Joint Testing setup

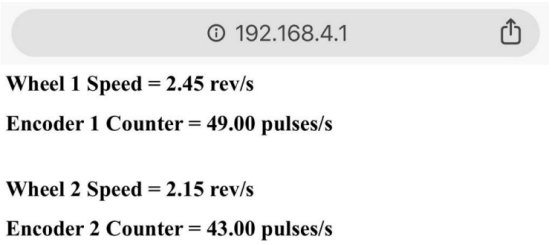

Figure 9: Encoder results from initial joint setup

Initial results from the encoders show a discrepancy between wheel 1 and wheel 2 even though the same speed is expected from both. This shows that we need to perform some filtration on the encoder results and also highlights the importance of the accelerometer because without it we would be unable to do position estimation.

B. Future Results

For the next few stages we expect the robot to display the position more accurately and also provide a live graph highlighting the filtered and unfiltered data so we can see the effects. If possible we would also like to visualize certain other state variables and create a model of the robot that moves in realtime within a 3D map

VI. Team Member Roles

Up until this stage of the project me and Youssef worked interchange on programming the robot and testing the various sensors. However the circuitry was handled by Youssef and the code for the web server and visualization was handled by me. Through this we understood our strong points and form now onwards we are dividing specific sections of the project between the two of us.

Both of us will work on programming the microcontroller, interfacing with the sensors and code for filtering data.

Youssef will work on circuitry and specific logic needed to convert sensor data to understandable readings

I will work on visualization and robot functions for movement and control

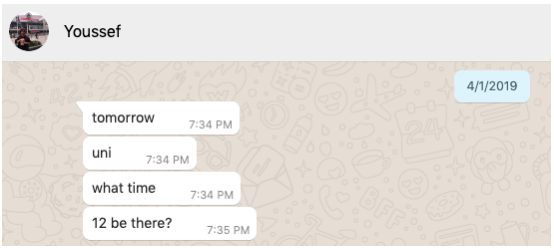

Figure 10: Planning first meeting during spring break

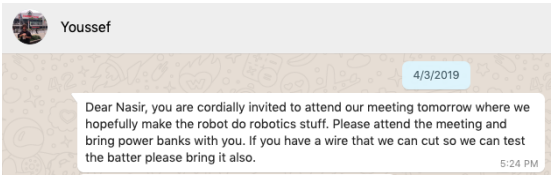

Figure 11: Planning second meeting during spring break

VII. Current State

A. Hardware

Currently the circuit is system is set up for joint testing as shown in Figure 8. The hardware is connected and are using a lab bench top power supply for testing. The set up is currently in the microwave lab and can be accessed at any time if one has lab access.

B. Software

All software is listed in the appendix. A.3 and A.4 were written by Youssef. A.5 was written by me. The remaining code was written by both of us.

References

[1] Souq. Smart robot car chassis kit for arduino. [Online]. Available: https://uae.souq.com/ae-en/smart-robot-car-chassis-kit-for-arduino-10745855/i/

[2] ——. Nodemcu v2 - lua based esp8266 development kit. [Online]. Available: https://uae.souq.com/ae-en/nodemcu-v2-lua-based-esp8266-development-kit-37195389/i/

[3] ——. Dual h bridge dc stepper motor controller l298n. [Online]. Available: https://uae.souq.com/ae-en/dual-h-bridge-dc-stepper-motor-controller-l298n-module-for-arduino-due-and-raspberry-pi-8424632/i/

[4] Amazon. Hc-020k double speed measuring module with photoelectric encoders. [Online]. Available: https://www.amazon.com/HC-020K-Measuring-Photoelectric-Encoders-Experiment/dp/B00EERJDY4/

[5] Souq. Huawei cp- 07 compact 6700 mah quick charging power bank. [Online]. Available: https://uae.souq.com/ae-en/huawei-cp-07-compact-6700-mah-quick-charging-power-bank-black-37071419/i/

[6] ——. Mpu6050 module 3 axis analog gyro sensors and 3 axis accelerometer. [Online]. Available: https://uae.souq.com/ae-en/mpu6050-module-3-axis-analog-gyro-sensors-and-3-axis-accelerometer-33451690/i/

[7] ——. Photoresistor ldr cds 5mm light-dependent sensor gl5516. [Online]. Available: https://uae.souq.com/ae-en/6-pcs-arduino-photoresistor-ldr-cds-5mm-light-dependent-sensor-gl5516-27385793/i/

Appendix

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>

//Pin Definitions

#define ENCODER1 5 //[D1]
#define ENCODER2 12 //[D6]

#define EN1  4   //[D2] 44 ON BREADBOARD
#define IN1  3   //[rx]
#define IN2  1 //[tx]

#define EN2  14 //[D5] 35 on BREADBOARD
#define IN3  16 //[D0] 43 ON BREADBOARD
#define IN4  13 //[D7] 42 ON BREADBOARD


const char* ssid = "Robot";

float count1;
float count2;
float rev1;
float rev2;
float rev1_f;
float rev2_f;
String message;

ESP8266WebServer server(80);
```

Listing 1: Initializaion Code

```
void setup() {
  delay(1000);

  //Defining PIN directions

  pinMode(EN1, OUTPUT);
```

```
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(EN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);

  delay(1000);
  WiFi.softAP(ssid);

  IPAddress myIP = WiFi.softAPIP();

  analogWrite(EN1, 512);
  analogWrite(EN2, 512);
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);

  server.on("/", handleRoot);
  server.begin();

  pinMode(ENCODER1, INPUT);
  pinMode(ENCODER2, INPUT);

  attachInterrupt(
      digitalPinToInterrupt(ENCODER1),
      High_Callback,
      RISING
    );
  attachInterrupt(
      digitalPinToInterrupt(ENCODER2),
      Low_Callback,
      RISING
    );
}
```

Listing 2: Setup Function

```
void loop(){
    rev1 = 0;
    rev2 = 0;
    for(int j=1; j<11;j++){
        count1 = 0;
        count2 = 0;
        delay(100);

        rev1 += count1 / 20; //number of revolutions
        rev2 += count2 / 20; //number of revolutions

    }
    rev1_f = rev1;
    rev2_f = rev2;
    server.handleClient();
    delay(100);
}
```

Listing 3: Encoder Counter Math

```
void High_Callback(){
  count1 += 1;
}
void Low_Callback(){
  count2 += 1;
}
```

Listing 4: Callback Functions for Encoders

```
void handleRoot(){
    message = "<h1>Wheel 1 Speed = ";
    message += String(rev1_f);
    message += " rev/s";
    message +=  "</h1>";

    message += "<h1>Encoder 1 Counter = ";
    message += String(count1);
    message += " pulses/s";
    message +=  "</h1>";
    message +=  "<br>";

    message += "<h1>Wheel 2 Speed = ";
    message += String(rev2_f);
    message += " rev/s";
    message +=  "</h1>";

    message += "<h1>Encoder 2 Counter = ";
    message += String(count2);
    message += " pulses/s";
    message +=  "</h1>";
    message +=  "<br>";

    server.send(200, "text/html",  message);}
```

Listing 5: Server Code to display Encoder Data