

ELE494-08

Autonomous Robotic Systems

Project CTE Document 3

Yousif Khaireddin

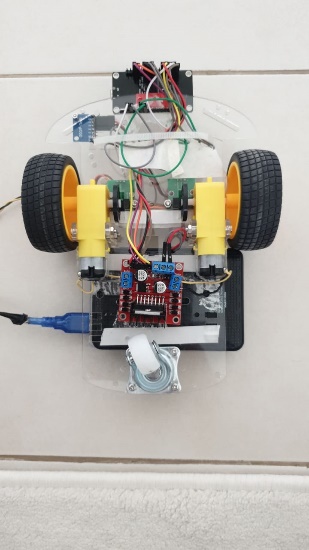
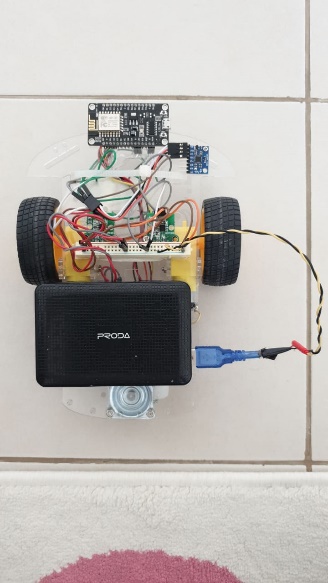
63618

Dr. Shayok Mukhopadhyay

**Results I have Achieved**

Although we have not been able to actually complete the required objective throughout our project, there are still a number of subtasks that we have completed that were mandatory for the project. One of the most important things I have managed to complete is properly track the execution of the program in time rather than depend on delays. This essentially means that I was able to track the exact time take by the execution of each iteration of our main loop. This is necessary since it enabled us to perform the integrations required using the exact timesteps. This was also necessary as it enabled us to place a timestamp on the obtained values for later plotting.

Another major result I achieved was properly building the robot and ensuring the proper functionality of all parts together as required. It is important to note that this was not done entirely by me, Nasir had a major contribution here as well.

Next major result I have achieved is properly calibrating the motors by altering the applied voltages such that I account for mismatches in design. Before this, applying the same voltage to both motors would make the car tilt to one side. Through this I was able to get the car to move straight when required.

This is about it for the actually perfectly working hardware results of the project; however, I also did implement Ackerman’s steering to try and move the car autonomously. This part of the project was not successful since the encoders and accelerometer used were not accurate by any means. It is actually for this same reason that applying a complementary for state estimation was also not successful.

**Contribution to Project**

Throughout this project I was mainly involved in interfacing all the hardware together, and I took car of a good chunk of the code required for the operation of the robot. For a good understanding of each of our contributions, it might be useful to go through our GitHub repository that Nasir created for our project stay coordinated and organized.

<https://github.com/NasirKhalid24/ELE494-08-Project>

* Tracking time

As discussed previously, one of the major things I have done throughout this project was properly track time of execution of each loop. The code for this can be seen below. This would execute at the start of each loop.

//obtaining the timestep between loops

ts = (millis() - tremove)/1000;

tremove = millis();

* Ackerman’s Steering

The next major contribution I had in this project was building the code required for Ackerman’s steering. Once again, this code is not perfectly functional due to many hardware issues including noisy sensors, an extremely rigid wheel and a faulty motor. Nevertheless, this code can be seen below.

//implementing ackremans steering

xh = x + h\*cos(theta);

yh = y + h\*sin(theta);

e1 = xh - xr;

e2 = yh - yr;

v = (k1\*cos(theta) + k2\*sin(theta))\*e1;

w = ((-k1\*sin(theta)/h) + (-k2\*cos(theta)/h))\*e2;

* Complementary Filter

Of course, as can be seen in the code above, Ackerman’s steering depends on an the position and orientation of the robot (x , y , theta). To attain this, I was responsible for building the complementary filter. This section of the code is split into many major subparts which include the following:

Reading encoder speed values and integrating it for position

//obtain the speed of wheels in revolutions/sec   
//using the count values from the interrupt functions

//20 is the number of slits in the encoder

rev1 = count1/(20\*ts);

rev2 = count2/(20\*ts);

//change these speeds into m/s for each wheel

v1 = rev1\*2\*PI\*radius;

v2 = rev2\*2\*PI\*radius;

//obtain actual robot speed

v\_actual = (v1 + v2)/2;

//obtaining positions using integration of encoder velocity

x\_1= x\_1+ ts\*v\_actual\*cos(theta);

y\_1= y\_1+ ts\*v\_actual\*sin(theta);

theta = theta + ts\*w;

That would provide our first estimate for position, the next estimate will come from the accelerometer. Reading the accelerometer values was done by nasir, so he will be coving that in his CTE. However, combining these values together using a complementary filter was done by me, and that can be seen in the code below.

//obtaining an estimate of position using complementary filter

x = x\_1 \* w1 + x\_2 \* w2;

y = y\_1 \* w3 + y\_2 \* w4;

* Applying motor voltages

Once the above has been completed, It is necessary to apply these required voltage values on each motor, and the code for this can be seen below. During this, it was necessary to find the exact pwm value that will shut off the motors since that will serve as our 0 velocity. It was also very important to consider the fact that switching motor directions immediately will burn the h-bridge used. All these considerations can be seen below.

//obtain required speed for each motor

vl = v + (l\*w/2);

vr = v - (l\*w/2);

//scaling the voltage values between 650 - 1024

pwm\_l = ((1024 - 750) \* (vl/25)) + 750;

pwm\_r = ((1024 - 750) \* (vr/25)) + 750;

//apply voltages to wheels

**if** (pwm\_l > 0){

//must switch off mototrs before switching direction

**if** (flag\_p\_l != 1){

analogWrite(EN1, 0);

flag\_p\_l = 1;

delay(100);

}

digitalWrite(IN1, LOW);

digitalWrite(IN2, HIGH);

}**else**{

//must switch off mototrs before switching direction

**if** (flag\_p\_r == 1){

analogWrite(EN1, 0);

flag\_p\_r = 0;

delay(100);

}

digitalWrite(IN1, HIGH);

digitalWrite(IN2, LOW);

pwm\_l = abs(pwm\_l);

}

**if** (pwm\_r > 0){

//must switch off mototrs before switching direction

**if** (flag\_p\_r != 1){

analogWrite(EN2, 0);

flag\_p\_r = 1;

delay(100);

}

digitalWrite(IN3, HIGH);

digitalWrite(IN4, LOW);

}**else**{

//must switch off mototrs before switching direction

**if** (flag\_p\_r == 1){

analogWrite(EN2, 0);

flag\_p\_r = 0;

delay(100);

}

digitalWrite(IN3, LOW);

digitalWrite(IN4, HIGH);

pwm\_r = abs(pwm\_r);

}

//apply required speed for each motor

//note: the 0.9 factor is just used for calibration since the motors aren't identical

analogWrite(EN1, pwm\_l\*0.9);

analogWrite(EN2, pwm\_r);

**2. Save your Midterm 1, Midterm 2, and Final CTE documents as separate .pdf documents. Combine them, and upload one .pdf document - This is your final CTE document. If your CTE documents along the term were on online repositories, save screen shots of the main page of each as a .pdf document, and then combine them with your final CTE.**