ELE494-08

Autonomous Robotic Systems

Project CTE Document 3

Yousif Khaireddin

63618

Dr. Shayok Mukhopadhyay

Part 1

Results I have Achieved

Throughout this project I have learned a lot and achieved a great amount. One of the most important things I learned was choosing the required sensors and interfacing them to properly and accurately complete a required task. The next most important thing I learned is the motivation behind the need for a Kalman filter, and this comes from the lack of accuracy we faced with our sensors. One of the major results I have achieved was adequately selecting all the hardware (with the help of Nasir) and properly piecing together all the parts to build the entire system. Another important result I achieved was learning to properly track the execution of a program throughout time since that is required throughout all the integrations we have used in the code. One last result I achieved was properly calibrating the motors and other hardware used to account for the inaccuracies present in mismatch of design.

Contribution to Project

I would like to preface this section by saying that all sections of the project had significant input from both members since we always worked side by side in the lab. So, while it is me, in fact, who completed the things listed below, Nasir's input was vital throughout every step.

- Choosing the correct hardware for the project
- Developing separate testing mechanisms for each piece of equipment
- Constructing the robot and developing the circuitry
- Developing code to account for reversing directions
- Tracking timestep throughout the code
- Building the code for Ackerman's steering
- Building the code for the complementary filter

One good way to track each of our contributions would be to visit the GitHub repository Nasir has set up here:

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

Part 2

CTE 1

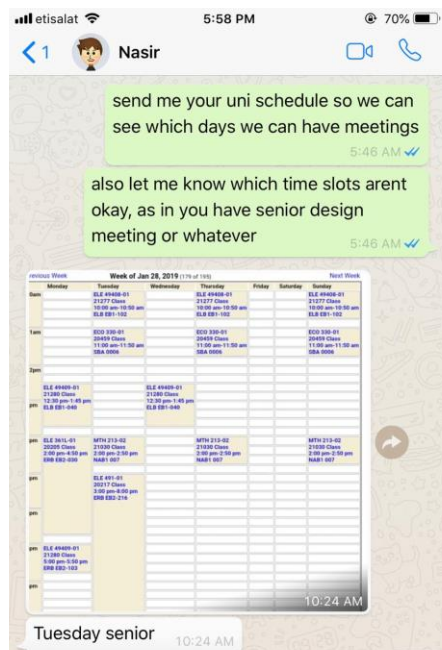
Initial Goal Statement

Throughout our project, we how to create a robot with accurate position estimation that will be able to survey an area and obtain the location of maximum light intensity.

Team Formation

Nasir and I have previously worked on many other projects together, so we already understood the way we both thought and assessed different scenarios. For this, as seen in the picture on the right, as soon as the project was announced in class, I messaged Nasir to ensure he would work with me.

Once we established that we are working together, I set up our first meeting on where we did some initial research regarding different topics we could work on.

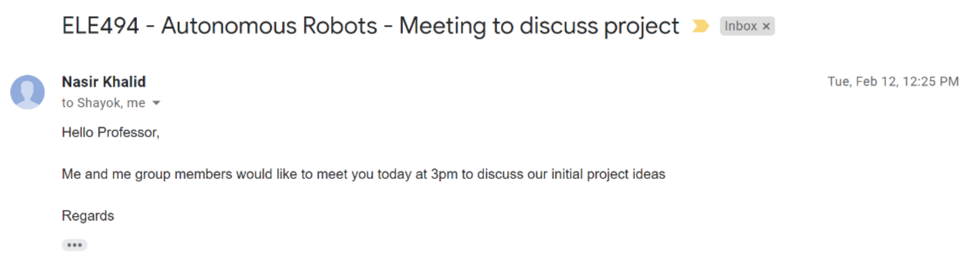


The outcomes of this meeting are listed below.

Meeting 1 - Brainstorming Ideas

- Drink Pouring Machine
- Light Detection (for solar panels)
- Plotter
- Line Follower
- Lane Detection (software)

We then set up a meeting with Dr. Shayok to discuss all our ideas, help us develop the, and help us narrow down our options regarding what is and is not possible though the knowledge we will attain in this class. Eventually, we agreed on locating the area of maximum light intensity within a region.



ELE494 - Autonomous Robots - Meeting to discuss project Inbox ×

Nasir Khalid Tue, Feb 12, 12:25 PM
to Shayok, me

Hello Professor,

Me and me group members would like to meet you today at 3pm to discuss our initial project ideas

Regards

Team Member roles

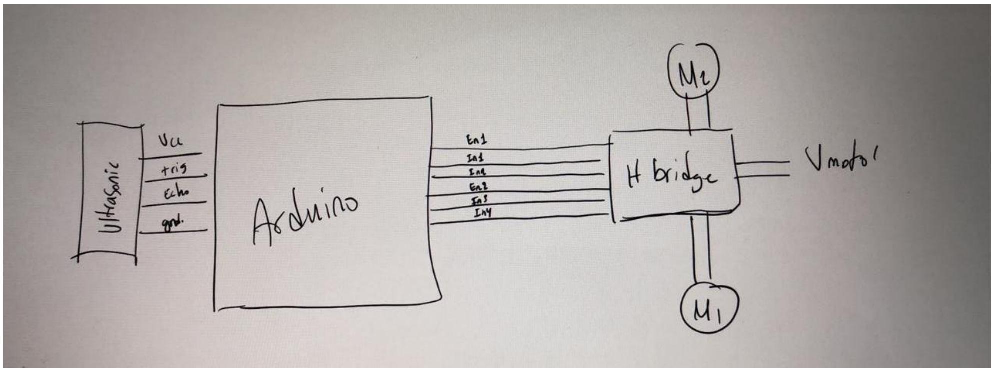
Since Nasir and I are really busy throughout the week, we usually have one meeting on Thursday which lasts around 3-6 hours between research, coding, building, and so on. For this, there hasn't been a very clear division between the roles.

However, so far, Nasir has been the one responsible for developing the code to run the motors, I have worked on the code shown below which is required for utilizing the Ultrasonic sensor in distance measurement, and both of us have worked together on actually constructing the robot which is shown on the next page. When constructing the robot, I was the one responsible in deciding the location of the different components, and both of us worked together in wiring and soldering everything.

The following code was written to test the ultrasonic sensor

```
#include <hcsr04.h>

//pin definitions
#define TRIG_PIN 5
#define ECHO_PIN 4

HCSR04 hcsr04(TRIG_PIN, ECHO_PIN, 20, 4000);

void setup(){
    Serial.begin(9600);
}

void loop() {
    Serial.println(hcsr04.ToString());
    delay(250);
}
```

Regarding hardware, after researching the workings of the different components, I sketched the following circuit diagram to give the both of us a better idea of how we could implement motor control and ultrasonic distance measurement. Once we had agreed on the connections, we worked together to actually build the robot (as seen on the next page) according to this.



Regarding the rest of the project, we require incorporating two encoders and a gyroscope regarding hardware, and actually programming the robot to identify its position in terms of software. So, Nasir will be responsible for writing the programs for these components to communicate with the robot. I will be responsible for wiring and mounting these components onto to the robot, and we will both be responsible on developing the mathematics and the actual program regarding accurate position localization and estimation.

Team Strengths and Weaknesses

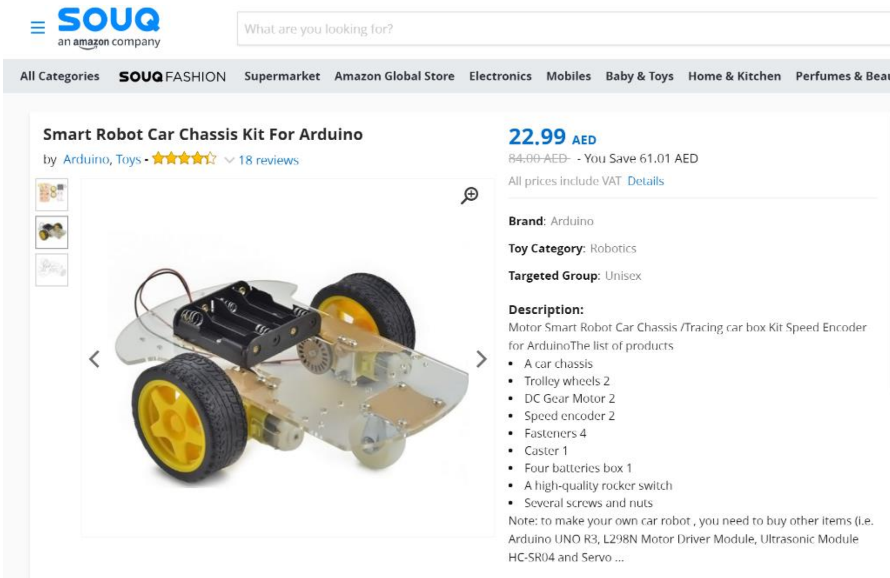
<table>
<tr><th></th><th>Nasir</th><th>Yousif</th></tr>
<tr><td rowspan="2">Strengths</td><td>Experienced w/ Arduino
More free time in schedule</td><td>Experienced w/ RaspberryPi
Better at developing circuitry
Handy work
Lives in dorms</td></tr>
<tr><td colspan="2">Programming
Good team communication
Strong researching capabilities</td></tr>
</table>

<table>
<tr><td rowspan="2">Weaknesses</td><td>Procrastination</td><td>Time management
Doing multiple projects this semester</td></tr>
<tr><td colspan="2">Limited knowledge of mechanical systems
Limited knowledge in accurate positioning</td></tr>
</table>

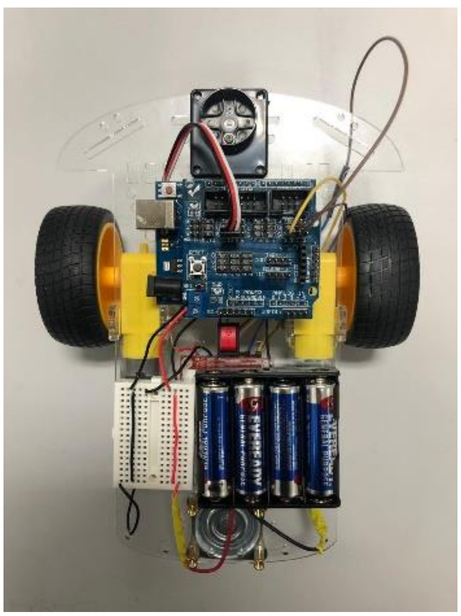
Broad Objectives

As agreed with Dr. Shayok throughout our meetings, we have decomposed our project into simpler and simpler sub projects. Depending on our ability to progress through our ideas and complete the tasks required, we will develop the project further. So, we will initially start with performing accurate position localization within a region. Once this is completed, we could develop the robot further to allow it to detect the light intensity of the area it is in. Should we complete this, we will then move on to detecting the area of maximum light intensity within an area and optimizing the rate at which the robot can reach there. Next, we could work on obstacle avoidance or path planning while performing this task.

To get us started, we purchased a Arduino car kit as can be seen below.



Over the past two weeks, we built and modified this kit, yielding the following.



Moving on from here, as mentioned earlier, we hope to incorporate two encoders and a gyroscope to aid us in positioning. We have already purchased these parts and are waiting on their arrival.

Next, we hope to develop a the Kalman filter required to help us transform these velocity and acceleration readings into usable data for distance measurement.

After developing this program, we can move onto adding the light sensors (LDRs) and interfacing them with the robot to help us detect the light intensity as we travel and use this information to detect the area of maximum light.

CTE 2

Updated Goal Statement

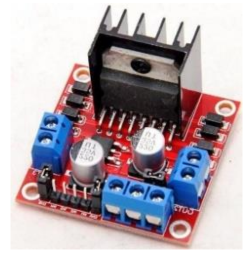
Throughout our project, we aim to build a robot that will survey an area to identify the point of maximum light intensity while simultaneously wirelessly communicating without our phones to send back real-time information regarding its position and light intensity at that area. Should time allow, we hope to also incorporate obstacle avoidance without prematurely hard-coding the location of these obstacles.

Objective

This robot will be placed in a pre-defined area which it should survey to obtain the location of maximum light intensity. We hope to accomplish the above through continuously measuring the position of the robot using two encoders and an accelerometer and feeding this information into a Kalman filter to obtain accurate positioning and localization. During the execution of the continuous loop described above, the robot should also measure the light intensity as it moves. This whole process will occur while the robot wirelessly feeds this information back to us enabling us to plot its movement and progress in real-time along with the light intensity values at these different positions. Once done with surveying the area, the robot should go to the point of maximum light intensity and stay there.

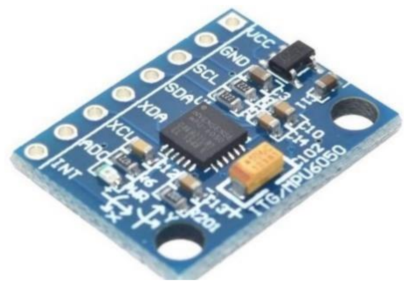
Hardware Specifications

Robot Chassis, Wheels, & Motors

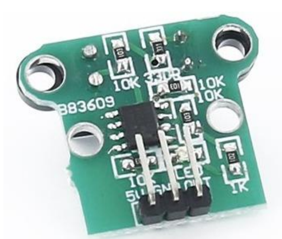


L298N Dual H-Bridge motor controllers

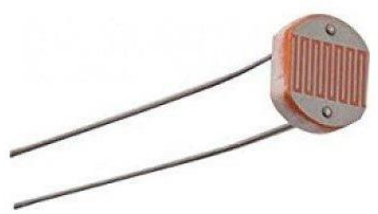
ESP8266 based NODEMCU microcontroller



MPU6050 3 axis accelerometer/gyroscope



HC-020K Speed Measuring Module



Photoresistor LDR CDS 5mm

Huawei 6700 mAH power bank

Plan of Action

Preliminary Tests

Once we have everything required, we hope to start by building the code to run each of the components separately. This will be done through powering each thing alone and watching for its expected operation.

Robot Assembly

Once we have ensured the different systems work separately, we can then mount everything onto the chassis and test that the robot can actually move. We will supply some PWM signal and watch for the robot's motion.

Sensor Data Acquisition Functionality

Next, we will test he ability of the encoders to capture the robots wheel speed by running the robot for a set number of wheel rotations and ensuring that the encoder counter is relatively accurate.

Next, we will test the accelerometer to ensure that it is also producing a reasonably understandable and logical output with changes in the robot's acceleration. To do this, we could increase the PWM signal over time and see that that the acceleration output is also increasing.

Accuracy in Positioning

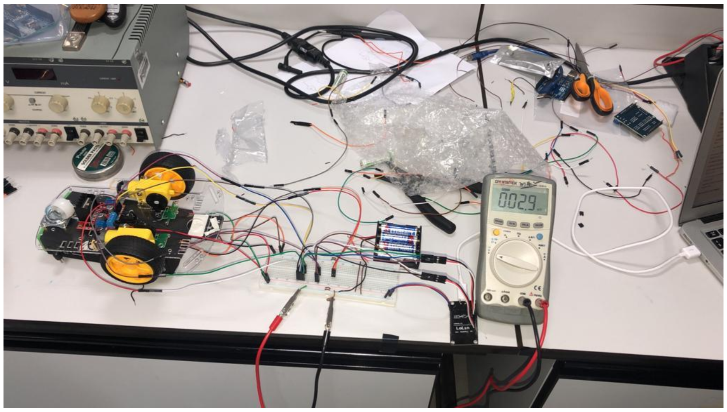
Once done, we will then move on to implementing the Kalman filter code and ensuring its operation. This will be tested by moving the robot 1m and seeing that its measured position with respect to its initial position is 1m (with reasonable allowed error).

Real-Time Communication

Then, we will move on to ensure the ability for the robot to communicate with us in real-time, as required by testing that the measured values are indeed passed over the network created and plotted as expected.

Primitive Testing and Results

Some initial testing of the system can be seen below. Here, we were in the process of testing that the different components do indeed work well together, before we mount everything onto the robot.



Hardware Connections

Through the testing done above, we were able to ensure that the encoders are indeed operating as expected. Their outputs were then fed into the access point to allow us to access them through our phones and see the following.

Display of Results

From the above, as seen as though our scenario isn't overly complex, we expect that the Kalman filter will indeed rectify the errors in positioning that will occur if we just integrated these speeds to obtain a measure of position. It is important to note that although the same PWM signal was applied to both H-Bridge inputs, the second wheel is (according to our encoder alone) spinning 0.3 rev/sec slower than the other one. This means that some calibration and balancing will be required to ensure that both wheels are equally responsive and similar in performance.

Next, we also expect that the wireless transfer of results to be possible since we were able to obtain the above results. Currently, these values are updated whenever this page is refreshed, so actually taking this data to create real-time plots might present some issues.

Team Member Roles

Regarding our project, both members will equally contribute to almost all parts to fulfill the required objects.

I will be responsible for developing the circuitry and Hardware connections.
The circuit in figure 1, for instance, was built by me.

Nasir, on the other hand will be responsible for preparing the code for displaying the results in real-time. This can be seen in figure 2.

Next, we will both be responsible for implementing the Kalman filter into our system and developing the algorithm that the robot should follow to ultimately reach the area of maximum light intensity.

Some of the code we have already written regarding these tasks, along with which member wrote that specific set of code, can be seen in the appendix.

<u>Appendix</u>

Some of the code we have already developed can be seen below:

Here are the initializations (written by both)

```
#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);
```

Server Operation (written by Nasir)

```
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);
}
```

Microcontroller Setup (written by both)

```
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);
}
```

Encoder Counter (written by me)

```
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);
}

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