G17 - PH222 Final Report

Automated Solar Panel Positioning System For Enhanced Energy Collection

Akshay Chaturvedi - 22B2432
Kartik Vuthoo - 22B4234
Shorya Sethia - 22B2725

April 14, 2024

Abstract

The report details the process leading up to the target of building a sun tracking solar panel to increase efficiency of solar power harvesting. A working model with both the rotating apparatus and brightness tracking line follower component has been built, as proposed at the beginning of the project. The idea of an LCD display being attached to the apparatus for showing efficiency in real time was deemed superficial and discarded.

1. **What was the contribution of each team member to bringing the implementation of the concept to fruition ? What was the time commitment the project ended up requiring ?**
   Although each member of the group gave their best to actively participate in each stage of the project and gave roughly 15 hours each, there were some unique contributions each team member made which also deserve to be outlined :

   1. **Kartik Vuthoo** :
      Came up with the idea for the project and put together the initial stages by discussing and deciding what we should accomplish and how, roughly. Also helped with electronic circuits component by soldering various components to help set them in place.
   2. **Shorya Sethia** :
      Dealt with majority of the technicalities of Arduino coding and also did the initial prototyping of the rotational apparatus.
   3. **Akshay Chaturvedi** :
      Majorly helped with integrating the line following and rotational components together and making them work in unison. Also helped in carving out the logic of the code in some places.

2. **Recapitulate the original proposal of the project.**

   1. The project proposed to build a device to move a solar panel such that it optimally harvests solar power.
   2. To achieve this, the position of the sun would be tracked through the day using a rotational mechanism (with 2 degrees of freedom).
   3. Additionally, a bot-car following a set path (line follower) was to be integrated to station the solar panel on the spot of maximum brightness on the roof. This would be done by sampling the roof once using some representative points and then settling at the spot of maximum recorded brightness.
   4. It was proposed, for demonstration purposes, to add an LCD display, showing the efficiency in real time (by comparing the spot of lowest and highest intensity).

3. **What are the major milestones achieved that can be marked as steps towards the completion of the project? Include workarounds/hacks used to fix unforeseen problems.**

    1. **3D Printing the rotation apparatus** (Credits - Design Studio and IDC) :
    The carefully designed base and additional support structures to house the servo motors and mount the solar panel on, provided a strong start to the project after being acquired and 3D printed as required.

    2. **Building the rotation apparatus prototype** :
    The rotation apparatus, after being appropriately soldered, wired and coded, worked smooth enough to qualify as a working prototype. The relevant code looks like (updated from last time to improve readability) :

```
#include <Servo.h>

Servo servo1; // Define first servo
Servo servo2; // Define second servo

int pos1 = 90; // Initial position for servo 1
int pos2 = 90; // Initial position for servo 2

int topleft;
int topright;
int downleft;
int downright;
float waittime = 500; // Time in milliseconds (0.5 seconds)


void setup() {
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);

  // Attach servos to respective pins
  servo1.attach(9);
  servo2.attach(10);


  // Your existing setup for Timer1
}

void loop() {
  // Read LDR values
  topleft = analogRead(A0);
  topright = analogRead(A1);
  downleft = analogRead(A2);
  downright = analogRead(A3);

  // Read the solar panel voltage
  float solarVoltage = analogRead(solarPanelPin);

  // Print the solar panel voltage value
  Serial.print("Solar Panel Voltage: ");
  Serial.println(solarVoltage/255);
```

```
  Serial.println(solarVoltage/255);

  // Servo control based on LDR reading
  if (topleft > topright) {
    pos1 += 1;
  }
  if (downleft > downright) {
    pos1 += 1;
  }
  if (topleft < topright) {
    pos1 -= 1;
  }
  if (downleft < downright) {
    pos1 -= 1;
  }
  if (pos1 > 180) {
    pos1 = 180;
  }
  if (pos1 < 0) {
    pos1 = 0;
  }
  if (topleft > downleft) {
    pos2 -= 1;
  }
  if (topright > downright) {
    pos2 -= 1;
  }
  if (topleft < downleft) {
    pos2 += 1;
  }
  if (topright < downright) {
    pos2 += 1;
  }
  if (pos2 > 180) {
    pos2 = 180;
  }
  if (pos2 < 0) {
    pos2 = 0;
  }

  // Set servo positions
```

```
  servo1.write(pos1);
  servo2.write(pos2);

  // Delay for smooth movement
  delay(waittime);
}
```

Although there was an issue with the extent of rotation. Rotation all the way to make the panel flat wasn't being achieved. After taking into consideration the movement of the sun over the day at various latitudes, it was deemed that rotation in the opposite direction was excessive, and an offset was used as a workaround.

Click/Tap here to see a demonstration of the rotation apparatus at this stage, along with the workaround.

Click **this** and **this** to see the working after connecting it to the same Li-ion battery as used for the line follower for autonomous functioning instead of powering it with the laptop itself.

3. **Acquiring and assembling line follower components (laser cut or otherwise)** :
The line follower chassis was laser cut. Wheels, DC motors and IR sensors were available in the lab. The motor mounts were separately acquired. The assembly followed along the lines of standard assembly of any line follower.

4. **Running of the line follower** :
The code for the line follower was made ready, connections to an Li-ion battery were made and IR sensors were calibrated for black insulation tape (low reflectivity) acting as the line and whiteboard (high reflectivity) acting as the surface and the line follower was running.

   Click/Tap here to see the working of line follower on its own (different calibration).

5. **Code for learning and going to the brightest spot** :
Finally, the code for the final demonstration of the line follower sampling a 2D plane and settling at the spot of highest brightness was developed and fed into the Arduino. The code used indexing to identify the location of highest intensity. The functions for recording at all locations and moving ahead instead of stopping at the inappropriate locations was separately written and after the hardware was in tested to be in working condition, the code did its job, leading to the completion of the project.

   Click here to see the final working model.

   The following is the code used finally :

```
//including the libraries
#include <AFMotor.h>

//defining pins and variables
#define left A1
#define right A4

//defining motors
AF_DCMotor motor1(1, MOTOR12_1KHZ);
AF_DCMotor motor2(2, MOTOR12_1KHZ);
AF_DCMotor motor3(3, MOTOR34_1KHZ);
AF_DCMotor motor4(4, MOTOR34_1KHZ);


const int n = 5;
float eff[n];
int i = 1;
int k=1;
int maxm=0;
int max_index=0;
// Define the pin for solar panel voltage input
const int solarPanelPin = A0;

float solarVoltage ()
 {
   // Read the solar panel voltage
    float voltage = analogRead(solarPanelPin);

   // Print the solar panel voltage value
    // Serial.print("Solar Panel Voltage: ");
    // Serial.println(voltage/255);

    return voltage/255;
 }

void record ()
 {
    if(i<=n)
     {
      eff[i] = solarVoltage();
```

```
      maxm=max(eff[i], maxm);
      i++;
      motor1.run(FORWARD);
      motor1.setSpeed(200);
      motor2.run(FORWARD);
      motor2.setSpeed(200);
      motor3.run(FORWARD);
      motor3.setSpeed(200);
      motor4.run(FORWARD);
      motor4.setSpeed(200);

      if (i==n)
      {

        for(int j=0;j<n;j++)
         {
           if(maxm==eff[j])
            {
              max_index=j+1;

              break;
            }
         }
      }
     }



 }

void move()
 {
      if(k<max_index)
       {
        motor1.run(FORWARD);
        motor1.setSpeed(115);
        motor2.run(FORWARD);
        motor2.setSpeed(115);
        motor3.run(FORWARD);
```

```
        motor3.setSpeed(115);
        motor4.run(FORWARD);
        motor4.setSpeed(115);
        k++;
       }

       if(k==max_index)
       {
        motor1.run(RELEASE);
        motor1.setSpeed(0);
        motor2.run(RELEASE);
        motor2.setSpeed(0);
        motor3.run(RELEASE);
        motor3.setSpeed(0);
        motor4.run(RELEASE);
        motor4.setSpeed(0);

       }

 }

void setup() {
  //declaring pin types
  pinMode(left,INPUT);
  pinMode(right,INPUT);
  pinMode(solarPanelPin,INPUT);
  //pinMode(frequencyReceivePin, INPUT);
  //begin serial communication
  Serial.begin(9600);

}

void loop(){
  int line;   // set it to 1 if line is black, and 0 if line is white
  line = 1;

  //printing values of the sensors to the serial monitor
  // Serial.println(digitalRead(left));
```

```
// Serial.println(digitalRead(right));

//line not detected by both
if(!digitalRead(left)==line && !digitalRead(right)==line){
  //Forward
  motor1.run(FORWARD);
  motor1.setSpeed(115);
  motor2.run(FORWARD);
  motor2.setSpeed(115);
  motor3.run(FORWARD);
  motor3.setSpeed(115);
  motor4.run(FORWARD);
  motor4.setSpeed(115);
}
//line detected by right sensor
else if(!digitalRead(left)==line && digitalRead(right)==line){
  //turn right
  motor1.run(FORWARD);
  motor1.setSpeed(200);
  motor2.run(FORWARD);
  motor2.setSpeed(200);
  motor3.run(BACKWARD);
  motor3.setSpeed(200);
  motor4.run(BACKWARD);
  motor4.setSpeed(200);

}
//line detected by left sensor
else if(digitalRead(left)==line && !digitalRead(right)==line){
  //turn left
  motor1.run(BACKWARD);
  motor1.setSpeed(200);
  motor2.run(BACKWARD);
  motor2.setSpeed(200);
  motor3.run(FORWARD);
  motor3.setSpeed(200);
  motor4.run(FORWARD);
  motor4.setSpeed(200);

}
```

```
    }
    //line detected by none
    else if(digitalRead(left)==line && digitalRead(right)==line){
      //stop
      motor1.run(RELEASE);
      motor1.setSpeed(0);
      motor2.run(RELEASE);
      motor2.setSpeed(0);
      motor3.run(RELEASE);
      motor3.setSpeed(0);
      motor4.run(RELEASE);
      motor4.setSpeed(0);

      if(i<=n)
       {
        record();
       }

      else if (i>n)
       {
        move();
       }

    }

  }
```

4. **If the project had to be split into its major modules which lead to its working, what would you identify them as ?**

   1. **Rotation Apparatus** :
      The rotation apparatus was one of the two mobile parts which formed the essence of the project. The apparatus included two degrees of rotational freedom (up-down and left-right) implemented using two 180 degree servo motors. The sense of direction was implemented using the cylindrical four sector divider. This created division among the left-right and up-down directions simultaneously.
   2. **Line Follower** :
      The translational degree of freedom was implemented using the line follower which followed a set path that mapped enough points on the 2D plane.
   3. **Solar voltage comparison and navigation mechanism** :
      This module is not a hardware component. It exists in the code and its hardware implementation is through the line follower, but it is logically separate from the line follower mechanism. The key difference is the thrust provided to move ahead from locations when both IR sensors read the black line but brightness isn't the highest. The voltage recording is done in the first round where all values are stored in an array and the same function also sorts the voltage values and finds the index (and hence the location) corresponding to the highest value (and thus the maximum brightness).

5. **How much of the proposed portion of the project was achieved in the final model?**
   The final model included both the features of optimal rotational solar panel positioning and translational positioning. The only component not added was an LCD display to show the efficiency in real time. This was due to overcrowding on the line follower apparatus (which is an issue we dealt with a lot) and the fact that it was for mere aesthetics anyway, so it was deemed unnecessary.

6. **What was the role of Arduino in the project ?**
   All the coding to run the hardware, as outlined above, was done in 2 Arduino UNOs (one for the rotation apparatus and one for the line follower and voltage recording) apparatus.

7. **What was the testing methodology?**
   That is outlined clearly enough by the working of the bot and what was intended to be achieved. Due to one of the challenges listed above, the functioning wasn't achieved in the desired environment (rooftop during daytime), but the demonstrations are based on the same logic and hopefully count as a sufficient proof of concept.

8. **What were the major challenges faced and how were they tackled?**

   1. **3D Printed Components Density Issue** :
      Although all the 3D printing was done in IDC, the first set of 3D printed components had a lower density than would be required to provide them the strength to withstand drilling, as they had to be connected and mounted properly. Since 3D printing takes time, this was a significant initial impedance. The next time, 3D printing was done with the highest possible infill density.
   2. **Interference in IR Sensors and Calibration** :
      The demonstration could not be done on a rooftop because of interference issues with the current IR sensor. The IR sensors would have to be of much higher quality if they were to be used for an outdoor demonstration than the ones we had, which would face a lot of interference due to IR rays from the sun during the day.

      IR sensors were also very sensitive to the background on which their detection threshold was being calibrated. Even similar looking surfaces sometimes showed very different results. This was combated by trial and error, ensuring the IR sensors were new and the black tape used to make the line to be followed, was insulation tape with low reflectivity.
   3. **Wiring issues due to mobility of components** :
      The wiring of the entire set up had to ultimately fit on the line follower chassis, which caused major issues because of wire crowding. The non soldered wires were repeatedly getting disconnected because some pins had an inherently loose connection and sometimes, overcrowding would also lead to some wires getting shorted. Ultimately, everything had to be minutely taped.
   4. **Alignment issue of rotation apparatus** :
      The rotation apparatus had restricted rotation due to the particular arm length chosen for the mounts of the arms of the servo motors. After identification of how much rotation was needed in which direction, the rotation was offset by using an isosceles triangle shaped mount.
   5. **Issue of components wearing out** :
      After testing the apparatus enough to be sure, the components (especially the IR sensors) started wearing out due to overuse and even with everything set and coded properly, the model wasn't working. After appropriate replacement and thus rewiring, the model worked.
   6. **Code Issues** :
      These form a relatively small subset of problems, but initially it was difficult to label the location of maximum intensity in the code to let the bot know where to ultimately stop but this was fixed by writing separate functions for recording and moving and calling them immediately when the "halt" instruction of the regular line follower had been reached, when both IR sensors detect the line (black).

9. **What were the major things learned from doing the project ?**

   1. Perhaps the biggest thing learned was the importance of discarding sub-optimal hardware, because even if all the things work fine on a conceptual level, sensitivities of the hardware at hand have to be taken into account while practical implementation.
   2. Another important thing was the lesson of securing already built circuits in place. One might expect their model to work after building the circuit, but unless there is a way to make sure that the circuital connections remain intact, the replication of important results cannot be guaranteed.

10. **Credits and Resources :**

    1. 3D Printing and Laser Cutting : Design Studio and IDC.
    2. Soldering and acquisition of components : Electronics Lab for the course.
    3. Nice illustration of basic circuit and concept(link).
    4. Illustration of an idea for coding the set-up(link).