Path-Following Car :EE3 Final Report

Partners: Greyson Beck, Jonathan Goh
Lab Section: Thursday @ 12:00

I. Introduction

"A Self-Driving Uber Car Kills Pedestrian in Arizona ."[1] This phrase ran as a headline across virtually every major news outlet in March of 2018, following an incident where one of Uber's self-driving vehicles was unable to avoid something unexpectedly entering the vehicle's path. In this case, that something was a human being, as opposed to a wall or dummy that may be found at a test site. Occurrences such as this bring to light the importance of both accurate sensor technology and the dynamic responses to sudden changes picked up by these sensors. Uber's vehicle includes top-mounted lidar, front-mounted radar, and short and long range optical cameras (which involves very complex computer vision algorithms) for its sensor system. It is believed that "no less than four different imaging systems should have picked up the victim."[2] On the other hand, Uber engineers have accounted for a variety of scenarios should these imaging systems pick up anything out of the ordinary, to which the responses vary from slowing the vehicle to swerving.[2]

We developed our electric vehicle with those concepts in mind, as they have proven to be crucial not only to the tech industry, but to the safety of the world. While the overall goal for this project was to design a path-following car, we understood, and still do, that the work we do in the lab must hold up to the standards of the real world. Thus, in our design, we prioritized accuracy over the speed and 'smoothness' of our car's movements. In other words, if ensuring that our vehicle stayed on the path the entire time meant a significantly slower time from start to finish as well as some harmonic motion pulling out of turns, then we were satisfied. To summarize the theory, this was implemented first and foremost via a sensor system, which included three pairs of IR LEDs and IR phototransistors, both aimed at the ground. The idea was that the light from the LEDs would be reflected back to the bottom of the vehicle and picked up by the corresponding phototransistor. The location of the central axis of the car relative to the path (black tape on white paper), was determined by the voltages sent to an Arduino Nano by the phototransistors. The second portion of the system was a transistor circuit, with each of the two transistors connected to a DC motor. The signal from the sensors would be outputted to the transistors, and thus to the motors, via the Arduino.

II. Testing Methodology (Focusing on the Path Sensing Subsystem)

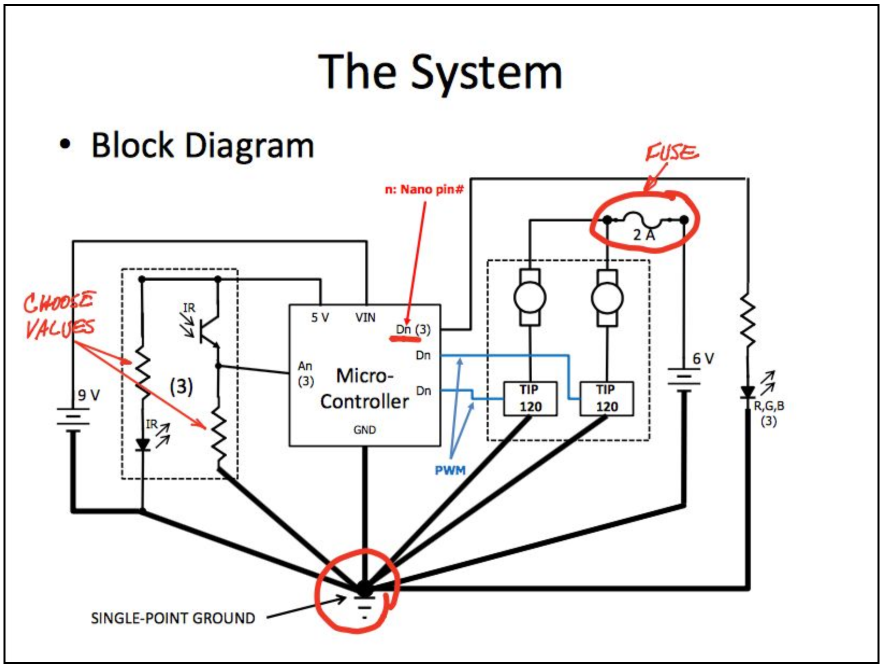

Figure 1.1: Illustration of the Vehicle Circuit. The values chosen for the resistors, denoted by "Choose values" were 10k (left) and 1k (right). Diagram taken from the lecture slides from class.

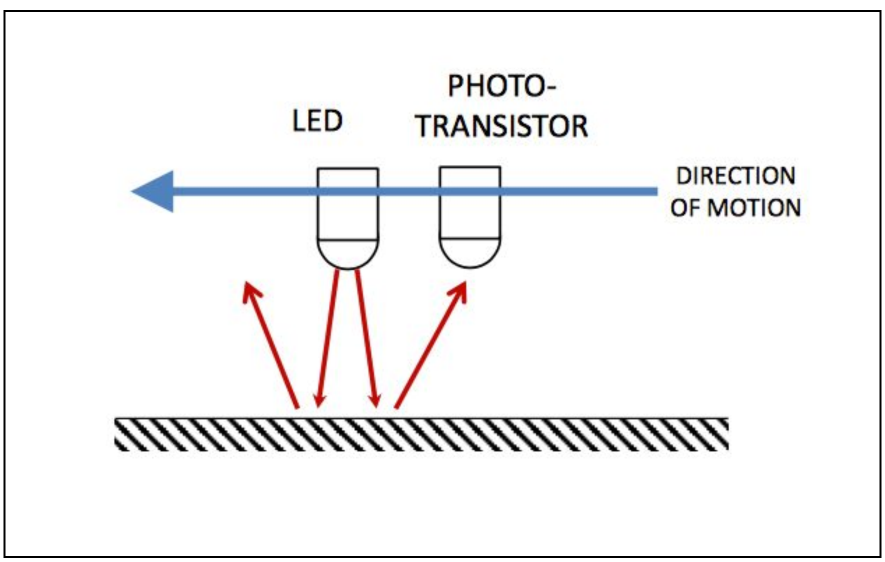

Figure 1.2: Diagram of the IR-LED/Transistor Pairs. The figure illustrates that the IR light is emitted from the LED and is reflected off the surface of the ground back to a Phototransistor. Diagram was taken from the lecture slides used in class.

How we Designed the Test:

To design the test for the Path Sensing Subsystem of our completed car, we took into consideration several the factors that would affect the movement and direction of the vehicle. It was determined that the IR-LED/Transistor pairs that sensed the dark zones and light zones were integral to the creation of the Path Sensing Subsystem. With IR-LEDs and IR Transistors in mind, we needed to ensure that our car would be able to sense both: light areas (representative of the white space of the track) and dark areas (representative of the black path of the track). The tests that were designed made sure that the response from the IR-LED/Transistor pairs were accurate and that the values that were collected from the car showed distinct high/low values to help determine what direction the car should move in.

How We Conducted the Test

The first test on the completed car was to check the IR-LEDs/Transistors, ensuring that each pair correctly sensed values based on the conditions they were set in. To read the values from the sensors, we wrote a basic loop in the Arduino IDE that read values from each IR Transistor's analog pin and printed the value of the sensor onto the console, which allowed for real time sensing values of the sensors. To physically test our values, we placed the car over white paper (representative of the white space of the track) to make sure that the resulting, printed, values were of a high magnitude. A series of other tests were similarly run on each IR-LED/Transistor pair, but with the LED and Transistor placed over a black colored portion of paper (representative of the black path of the track). Four repetitions of trials for light/dark were recorded for each IR-LED/Transistor pair. With the raw data we were now able to analyze the values read by the sensors.

Test #1

IR LED	**High**	**Low**	**Mean**
Left	225	90	160
Middle	260	115	190
Right	240	90	165

Table 1.3(a)

Test #2

IR LED	**High**	**Low**	**Mean**
Left	218	86	152
Middle	251	113	182
Right	243	91	167

Table 1.3(b)

How We Analyzed the Test Data

Firstly, we made sure that there was an evident high/low value sensed from each IR Transistor, if values did not have a distinctively high or low value, then we knew that there was something wrong with the sensing. Based on the recorded data from the sensors we averaged the high and low values for each sensor to find the common point that was approximately between the highest and lowest values of the sensor (midpoint value). The data also made it evident that trace amounts of light was leaking into adjacent IR-LED/Transistor pairs. To fix this issue we positioned the LEDs in a way that isolated light from accidentally being sensed by a different IR Transistor.

How We Interpreted the Data

Using the data that was collected from the sensors, we were able to find the midpoint of each respective IR-LED/Transistor pair. The midpoint value that was calculated was what we used as a base case to test whether the respective sensed value was lower than the midpoint (representative of a dark zone) or whether the respective sensed value was higher than the midpoint (representative of a light zone). This analysis was further broken down by realizing that based on the values from each transistor, we could determine the what direction the car must turn to stay centered on the dark path. When the left sensor and right sensor received high values, and the middle sensor received a low value we knew that the car was centered on the path (the car must continue along the dark path with a straight motion). When the left sensor and middle sensor received high values, and the right sensor received a low value we knew that the car was to the left side of the path (in order to correct this we turned the car in the right direction to center it back onto the dark path). When the right sensor and middle sensor received high values, and the left sensor received a low value we knew that the car was to the right side of the path (in order to correct this we turned the car in the left direction to center it back onto the dark path). Lastly, When the left sensor, right sensor, and middle sensor all received low values, we knew that the car had reached the end of the path (we stopped the movement of the car).

III. Results and Discussion

Test Discussion

As seen in **Table 1.3 a and b**, we collected two samples of data from our sensors on two different days so that we could ensure that we had accurate sensing. By gathering the high and the low readings of data, we were able to extrapolate a midpoint (left-160, middle 190, and right-165) for each IR-LED/Transistor pair. This midpoint was useful with determining values that were read that were significantly higher or lower than the value that we had calculated (High values: 225, 260, 240. Low values: 90, 115, 90). Since the IR-LED/Transistor pairs accurately read either really high or really low values, it was easy to determine what type of material each IR-LED/Transistor pair was sensing; an output of a high value represented the white paper background, while the output of a low value represented the black electrical tape. Using some basic knowledge that the color black absorbs light energy and the color white reflects light energy, we were able to check that our output was accurate to what we should have expected. Using the midpoint we made basic loops that checked whether the sensed values from the car were higher or lower than that of the midpoint value that we had calculated; this served the second purpose of letting us know the orientation of the car, and the direction that the car must turn to center the car back onto the path. When the left sensor and right sensor received high values, and the middle sensor received a low value we knew that the car was centered on the path (the car must continue along the dark path with a straight motion). When the left sensor and middle sensor received high values, and the right sensor received a low value we knew that the car was to the left side of the path (in order to correct this we turned the car in the right direction to center it back onto the dark path). When the right sensor and middle sensor received high values, and the left sensor received a low value we knew that the car was to the right side of the path (in order to correct this we turned the car in the left direction to center it back onto the dark path). Lastly, When the left sensor, right sensor, and middle sensor all received low values, we knew that the car had reached the end of the path (we stopped the movement of the car).

Race Day Results

Fortunately, our car was able to complete the course in its first run, with no major hiccups or adjustments necessary. The total time required for the car to finish was approximately 24 seconds, and judging by other groups that we witnessed, this was an average time. To be more specific regarding our own vehicle's performance, there was excessive weaving (oscillation) that occurred following the first turn. For us, this was an issue that plagued us throughout testing as we sought to make its 'accuracy' the most important aspect (we cared less about how it looked as it was following the track, and put more emphasis on the car's actual completion of the track). This seems reasonable given that makes up a majority of the total score. Another less critical aspect of the car's performance that was not spectacular was speed. Again, this was not a surprise and an aspect of design we were forced to sacrifice. In addition, although our car was able to sense the automatic stop and respond accordingly, its final resting position was at an angle. Overall, however, my partner and I were satisfied with the outcome of race day.

IV. Conclusions and Future Work

Overall, we achieved our ultimate goal with this project, as our car was able to follow the path and sense the automatic stop. However, although we are satisfied with this, we would have liked to implement the wheel speed sensing feature. Simply put, we ran out of time, as I am sure was the case with many groups. Aside from that, all sorts of innovative 'additions' to the project ran through our heads, though not all were feasible. On the simpler side, we would have loved to attach a GoPro to a spot on the top of the vehicle, simply for recording purposes. Similarly, we thought of placing an ultrasonic sensor somewhere toward the front of the car aimed in the direction of travel. Then, we could implement it as a distance measuring tool to check for collisions, making it a path-following, collision detecting car! To test this design, we would simply place solid objects at various distances in front of the sensor to determine the accuracy. After that, we would only need to write some very basic code into our arduino code that checks for the distance each time the void loop() runs, and if this distance goes below a certain value, the car would stop. While this would have been an awesome feature to implement, we clearly did not have the time, but maybe another group will get creative and try a similar concept. At the end of the day, my partner and I were able to gain some incredible hands on experience building a design that, as we look at some of the autonomous vehicles of the day, is not all that different.

V. References

1. Wakabayashi, Daisuke. "Self-Driving Uber Car Kills Pedestrian in Arizona, Where Robots Roam." *The New York Times* 19 Mar. 2018. *nytimes* Web. 12 June 2018.
2. Coldewey, Devin. "Here's how Uber's self-driving cars are supposed to detect pedestrians." *TechCrunch*. Oath Tech Network, 19 Mar. 2018 Web. 12 June 2018.