ELEC1601_R18_Group_13 Project report

Group Members

1. Zheng (Casto) Tang
   SID: 510413811
2. Ka Ho Cheng
   SID: 510223944
3. Shihao (Scott) Meng
   SID: 500535840
4. Yifu Chang
   SID: 510477592
5. Tushaar Jagannathan
   SID: 510576389

Contents:

PROJECT REPORT

Introduction

The objective of this project was to construct an algorithm that would allow a robot to navigate through mazes/obstacles to reach from a starting point to its respective finish point. Due to covid restrictions, this was done virtually and not with any physical hardware/equipment. The university has provided us with the basic code of both the robot and a maze layout. Each group was required to modify the raw code by working together to solve the issues and build the optimal algorithm possible.

Project Management

Project Management is crucial for any success in group projects. Improving communication between members allows us to save time and operate more efficiently. There were several ways we were able to sustain the level of efficiency.

Tools

The group made use of the software **Trello**. This allowed us to track our progress and observe each member's contributions. Meeting minutes and session reports were recorded and roles were delegated relatively equally among all group members with the help of Trello.

CmapTools was also used to create brief but detailed flow charts to summarize the events that folded during each meeting. The included topics that were discussed, issues that needed to be solved, and many more.

Discord and **Zoom** were frequently used to share files, conduct video calls for our meetings, and discuss the progress of the project.

For each member to access the code, **Github** was used. We already had an existing repository we had used for our previous lab assignments in the same unit and created a new folder containing all the contents required for this project.

Delegation and Task division

All members were assigned a role that they must fulfill for this project.
The roles are as follows:

- Casto - Meeting Facilitator
  In charge of setting up meetings and introducing the tasks at hand

- Yifu - Issue Tracker and Assistant Meeting Facilitator
  Reminding of the tasks completed/unfinished and tracking their progress

- Scott - Idea Curator (Cmap designer)
  Building intuitive and simple but detailed flow charts of our progress in each meeting

- Tushaar - Idea Curator (Session reporter)
  Description of activities done at each Lab session

- Ka ho - Reporter
  Recording of minutes of meeting

Meetings were held every Wednesday from 8 pm to 10 pm AEDT to discuss our code, how much progress was made and issues we are facing. Saturdays from 3 pm to 4:30 pm AEDT were reserved for discussing Managerial tasks on Trello and CMap.

Usual Lab classes were on Thursdays from 6 pm to 9 pm AEDT where we resolved mainly doubts that neither of our members could resolve, requiring assistance from tutors.

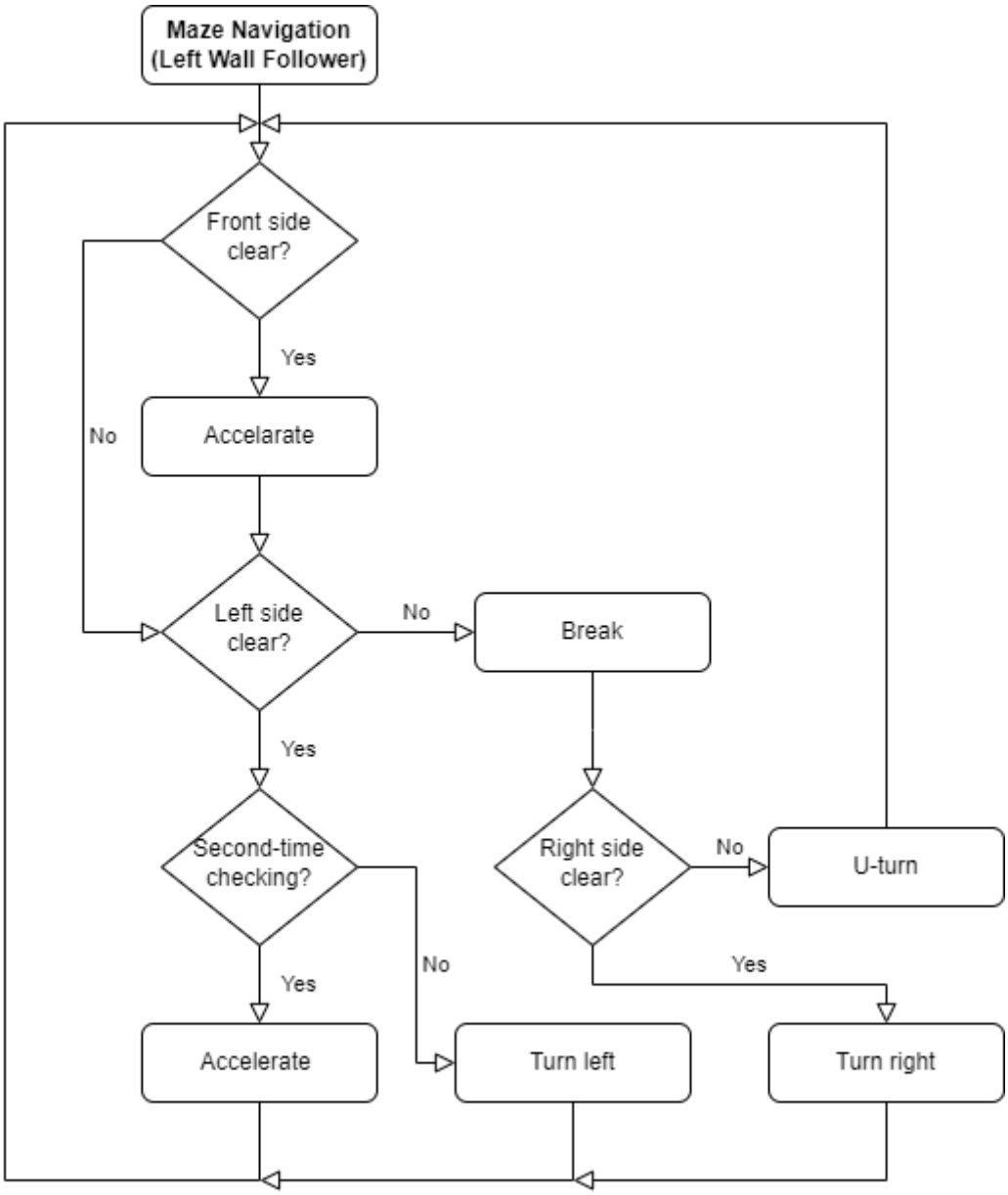
Discussion and Reflection

The team faced several challenges, particularly at the onset:

1) We were struggling to add multiple sensors to the robot and the mathematical operations were also quite challenging to understand. However, through fiddling with the code we were able to overcome this.
2) A considerable amount of time was required to design a maze as well but we were able to build a unique maze, first by drawing the schematic on an iPad and then recording the vector points of each joint, before writing the code.
3) Our robot would merely scratch walls that were diagonal and hence required us to calibrate the angle of our sensors to allow the robot to detect the wall much sooner in diagonal walls.

We mainly struggled with efficiency, often taking too much time for tasks that could have been done much sooner if we had read the project description much sooner. However, we learned from this and our efficiency improved as the weeks passed.

Software Implementation



Description

After several iterations, we eventually settled down on the wall-following algorithm to maximize the success rate when navigating more complex mazes. In short, the robot tries to move towards the closest left wall and keeps following it. While following the wall if the front is blocked, the robot attempts to turn right. If the robot reaches a dead end, it will keep turning right until the front is not blocked.

Pseudocode

- If the front side is clear, the robot will attempt turning left:
    - If the front is clear, the robot will accelerate until it reaches the top speed
    - At any point, if the robot has already turned for more than 360 degrees, it will move towards the wall located at 90 degrees to the left of the starting position
- If the front side is not clear:
    - If the left side is clear, the robot will attempt turning left
    - Else if the left side is not clear, the robot will attempt to break until it reaches a full stop
- If the robot is stopped and the front side is not clear:
    - The robot will attempt turning right until the front side is clear again
- If the robot reaches a dead end:
    - the robot will attempt to break until it reaches a full stop

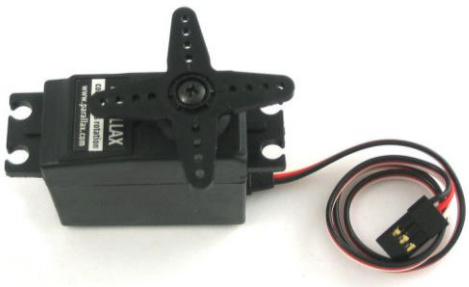
Highlights

- The robot will only attempt to break when both the front sensors detect objects in front, which makes it possible for the robot to maintain its speed while doing minor angle adjustments when it is already following the left wall.
- When the robot is accelerating, if it is attempting to turn left, the speed of which will be decreased by one unit of speed change to make the turn smoother.
- A counter is put in place to prevent the robot from driving in a circle when it is in an open area. If the front side is always clear, the robot will only attempt to turn left if the counter is less than 450 divided by the default angle change to force the robot to move towards the wall located at 90 degrees to the left of the starting position.

Hardware Design

Components

Continuous micro-servo



Continuous rotation servos are standard hobby RC servos that have been modified to offer open-loop speed control instead of their usual closed-loop position control. This effectively turns them into motors with integrated motor drivers in a compact, inexpensive package. A wheel can be attached and used for a drive system for a robot that can be controlled using an RC signal or a connection to a single microcontroller. Two were used in this model but for more stability, a set of four servos would be more suitable.

Ultrasonic Sensor

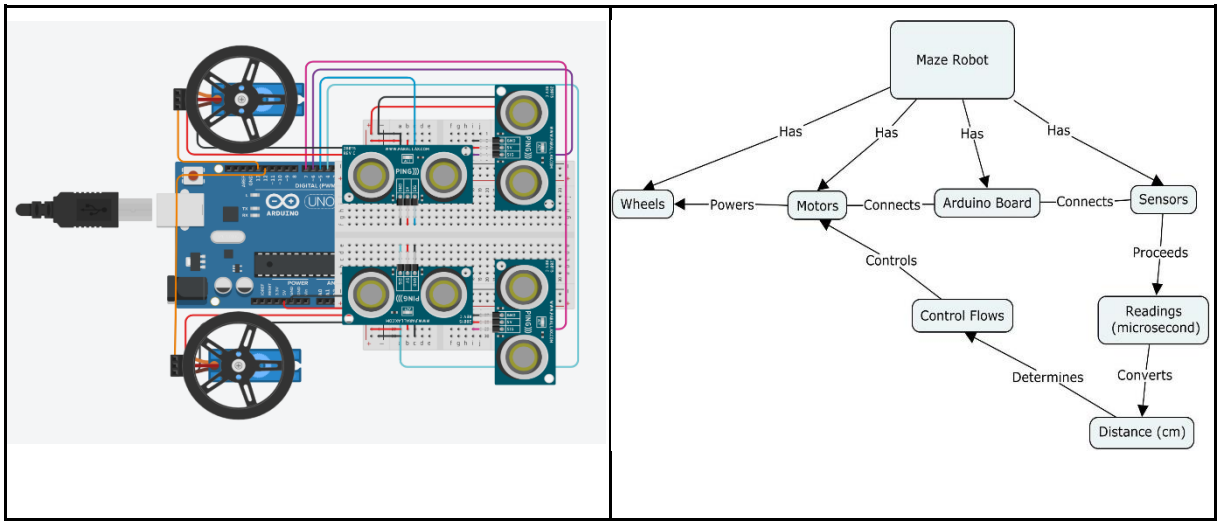
An ultrasonic sensor is a device that measures the distance of a target object by emitting ultrasonic sound waves and converting the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound.

Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver. The specific sensor we found available online would range from 2cm to 500cm with an optimal range of 10cm to 250cm.

Circuit Diagram

Since none of our group members had access to hardware components like an Arduino board, we constructed a 2D model using TinkerCad. We created a model to bring to life the design that we envisioned could mimic the properties of our virtual bot and CMap to explain how each component operates

The model and flowchart are as the followings:



Both the servos and the ultrasonic sensors have 3 pins on Tinkercad:

1) Power: To intake the required electrical power to operate the component
2) Signal: Provides a digital signal for use with hardware components. The output allows a hardware signal to control the pin's output drivers without requiring the CPU to write registers.
3) Ground: A safety measure when there's something wrong with your appliance, the grounding prong creates a new, low-resistance grounding path down to the main electrical panel. This trips the breaker, stopping the electrical current and preventing damage to your appliance

The reason that the side sensors may seem a bit too far inside is due to the optimal range of the sensor being between 10cm - 250cm (20cm to 300cm on TinkerCad). Therefore, bringing the sensors inwards actually produces the best results for detecting walls up close.

Calibration

The components will require the optimum settings to perform the operations we desire. Hence, calibrating them is crucial.

Depending on the model of our **servo**, we can calibrate the speed at which we want our robot to function. Most servos operate on a voltage of 4.8V or 6V with a max rpm of 95 rpm for the latter. Since Arduino is limited to an output of 5V the max Rpm will change to 78 rpm. Using a 100mm wheel rim for our tyres, the max speed achievable is 0.393m/s or 40 cm/s. Hence our model cannot operate at speeds greater than 40 cm/s. The weight of the robot does not affect max speed but the rate of acceleration reduces and should be taken into account.

Ultrasonic sensors produce ultrasonic sound waves to measure the time taken for the soundwave to return. Since we are trying to measure proximity, we must convert the time (in microseconds) to distance (in centimeters). Using the formula $v = d/t$, we can find read it as $d = vt$. The speed of sound is 340 m/s or 29 microseconds (The Arduino board measures readings in this unit for accuracy purposes). Since the sensor measures the echo of the sound wave the distance is doubled, hence to correct this the formula becomes $d = vt/2$. Now the units are consistent and can be used as our logic for control flows

Discussion and Conclusion

We managed to build a paradigmatic and robust algorithm that was able to not just solve the Test cases/Test mazes but also several unique mazes. We were also able to construct a simple maze, granted not as complicated but it represents our basic understanding of the code and our ability to manipulate it. We were also to make our demonstration more intuitive with a title screen and play around with the colours of the layout to make it more user-friendly to those who have little to no coding background

However, there are several ways where we could have done a more admirable job. The rate of progress was especially slow in the first week and project management should have been set during that week but was delayed. Installation of the software took quite some time and issues took longer to solve during the first week due to poor communication and lengthy waiting time for tutor assistance, during which little work was done

Overall, all of us are pleased with the result of our project, improving week by week to present a commendable and satisfactory project.

References

LazyFoo. (2004). *Getting an image on the screen.* Lazy Foo' Productions. Retrieved from https://lazyfoo.net/tutorials/SDL/02_getting_an_image_on_the_screen/index.php

n.a. (2021). *HC-SR04 Ultrasonic Sonar Distance Sensor.* Adafruit. Retrieved from https://www.adafruit.com/product/3942#technical-details

n.a. (2021). *New style mini RC Mecanum wheel omni robot var chassis kit with TT motor for Arduino.* Ali Express. Retrieved from https://www.aliexpress.com/item/4001150670259.html

N.a (2021) *FEETECH Continuous Rotation Servo FS5106R.* Retrieved from https://www.pololu.com/product/3430

SudKul. (2017). *RoboND-PathPlanning*. Github. Retrieved from https://github.com/udacity/RoboND-PathPlanning

Appendices

Trello link (Project management)
https://trello.com/b/pOvi7Dc3/2021elec1601r1813

Github Link (Source code)
https://github.com/Cavinsto/ELEC1601/tree/main/Project

TinkerCad link (Hardware model)
https://www.tinkercad.com/things/0hrCzT97ejp-elec1601project2/editel?sharecode=0l5YwXOnK3eZGjPFJ0dulL9vTpazNBTj6ZWudygq-jU

Google docs link (Project report)
https://docs.google.com/document/d/1yjWni9lEhrVuCN4m9p62mTkky5OXPLvUWLAnpvkyb3M/edit?usp=sharing