Final Project Report

Anshita Saini, Victor Wang, William Yue

December 2023

1 Problem Description

Upon completion of milestone 3, our robots were only able to autonomously navigate on a single floor of the GDC. This greatly limits its mobility as there exist many floors in the GDC that the robot is not able to reach. Our goal was to create a robot that can autonomously navigate between floors in the GDC using the elevators and prompt humans to press the elevator button for the robot (robot has no arm). The robot should be able to locate the elevator door and sit at a distance to detect humans passing by. The robot should be able to prompt humans to aid in button pressing only when humans are in the vicinity. Upon the elevator arriving, the robot should be able to detect the door opening and proceed to enter inside the elevator even in the presence of other humans waiting for the elevator in front of it. Finally, the robot should be able to leave the elevator on the right floor even in a crowded elevator.

2 Theory/Solution

2.1 Jackal

We used the Jackal robot for our project. We first reworked our code to be compatible with the Jackal. In particular, we had to identify the ROS topics for the Jackal's sensors and control, and process the message types since they were not always the same as on the F1/10 car.

The Jackal provided us with several hardware components that were useful for our project: speaker, touchscreen, camera. We used the speaker to prompt the human for help. We used the touchscreen for the human to confirm that certain steps had been completed, such as pressing the elevator up/down button, which we could not detect without human assistance. We used the camera to run object detection with YOLO[1] to detect humans nearby.

[1] https://pjreddie.com/darknet/yolo/

2.2 State Machine

We implemented the procedure of using an elevator with a state machine. We keep track of the state of the elevator, the human (present or absent), and the robot. The states are updated in response to robot actions (e.g. entered the elevator), sensor input (e.g. door opened), and human input (e.g. button pressed). Based on the current state, we choose how to prompt to the human. We outline our program logic below.

- Once the robot is facing the elevator, it samples the average laser range r_i in a sector in front of it and uses this to represent the baseline state where the elevator door is closed. As the robot waits for the elevator, it samples the same average laser range r now, and if $r - r_i > h_r$ for some threshold h_r, the robot decides that the door has opened.
- Before the door has opened, if the robot detects a human, it prompts them to press the elevator up/down button and confirm they have on the touchscreen. Once the robot has received confirmation, it stops speaking even if a human is present.
- When the robot detects the door open, it drives forward, stopping at obstacles. Once it has stopped for 3 consecutive seconds, it assumes this is because it has reached the back of the elevator (or stationary passengers). Then it spins around and asks the human to press the button for the target floor and confirm. (Initially, we only spoke when a human was visible, but we removed this condition after considering the case where the human was out of sight behind the robot.) When the human has confirmed, the robot asks the human to notify it on its touchscreen when they reach the target floor.
- When the robot reaches the target floor, it drives forward, and once it has stopped for 3 consecutive seconds, it assumes it has reached a wall outside the elevator. It may then proceed to its location on the target floor as in milestone 3.

3 Challenges Faced

One of the most challenging parts of our project was floor detection. Based on our initial plan, we wanted to read the floor number displayed on the screens inside the GDC elevators to detect when the robot should exit using OCR software specifically for digit detection. We were also planning on using a similar method to read the numbers next to elevator buttons to confirm when a human had pressed the button. However, after much experimentation, we faced two challenges with this. One challenge was that the screen displaying the current floor was not within the scope of the Jackal's camera. Second, although the elevator buttons and numbers were within the Jackal's field of view, the images we were able to retrieve from the camera were not as high quality as what was necessary for the OCR software to work properly.

Due to the challenges faced with OCR, we decided to try a different method of detecting how many floors the robot traversed. Our next idea was to record using the Jackal's microphone and count the number of floors traveled based on the elevator beeping sound at each floor. After recording several tests, we applied several signal processing techniques (including autocorrelation and a Butterworth bandpass filter) to try and extract the beeping sounds from our test recordings. Unfortunately, the microphone picked up far too much noise and we were not able to reliably extract the beeping sounds from the audio file, especially if many people were speaking or moving around in the elevator.

After our challenges with the microphone, we moved on to our last floor detection approach with the Jackal's accelerometer. After running a few tests with this, we also found that it was difficult to accurately compute the distance traveled because the accelerometer reported a changing base value when the robot was not moving in the z-axis on different days of testing (this was simply its measure of gravity). Because we had to subtract gravitational acceleration to compute the distance traveled, the changing base gravitational measurement resulted in compounding errors that impacted our calculations up to a few meters. Thus, we eventually decided to use our robot's GUI and human assistance to ensure the most reliable performance.

One last challenge we faced was losing the laptop's internet connection to the robot when in the elevator. Because we were using Google's text-to-speech library (which required Internet) to relay messages for human assistance, this impeded our robot's ability to speak. We eventually shifted to playing pre-recorded audio files so that we did not need this connection anymore and ran all of our scripts with `nohup` so that losing SSH connection did not affect our robot.

4 Code Link

Our code can be found here: https://github.com/WilliamYue37/cs378_starter/tree/jackal