“WallE” Autonomous Navigation Robot: Project Postmortem

Baran Usluel

Georgia Institute of Technology
School of Electrical and Computer Engineering

Submitted
Dec 3, 2018

Introduction

This document summarizes the strategy used by team WallE for the ECE 2031 final project. The design problem was to implement the autonomous navigation of a DE2 robot using SCASM, so that it moves from a random start position and orientation in a clear area, to the predefined goal area, as shown in Figure 1.

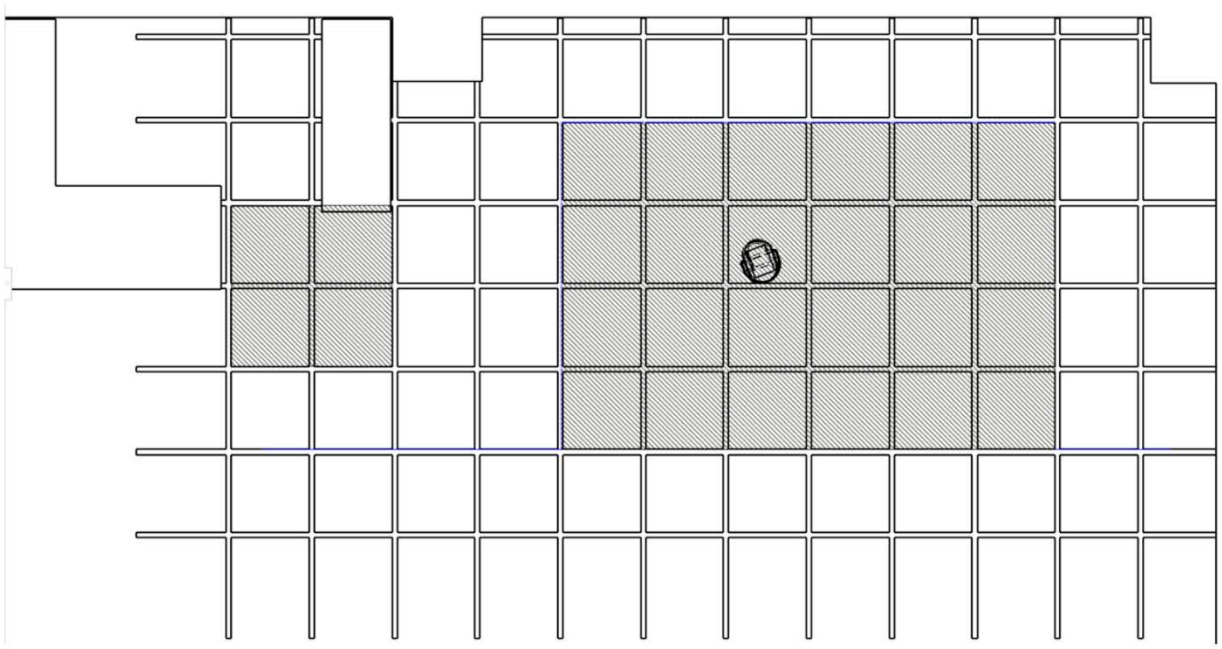

Figure 1. Layout of the robot demo area in the Digital Design Laboratory. The square grid represents the 2x2 feet carpet tiles, with the start (right) and target (left) areas highlighted.

Our proposed solution was to implement a wall-following algorithm to move around the room counterclockwise and stop in front of the goal area using a trigger condition, before finally moving into the goal area. This strategy had several benefits, including its feasibility, robustness due to reduced complexity, and efficiency. Ultimately, it yielded some positive results during the demo, but would have benefitted from more testing to improve the implementation.

General Methodology

Planning

We spent a significant amount of time at the beginning of the design process trying to agree on an strategy. During this period, we went through several different ideas, including implementing a Simultaneous Localization and Mapping (SLAM) algorithm, but decided against them because they were infeasible or unlikely to work. The wall following strategy was embraced because of its relative simplicity.

We also decided to use a Google Docs document to keep a detailed log of our progress and a private Git repository for version control of our code.

Design

The robot strategy that we proposed and implemented consisted of the following steps:

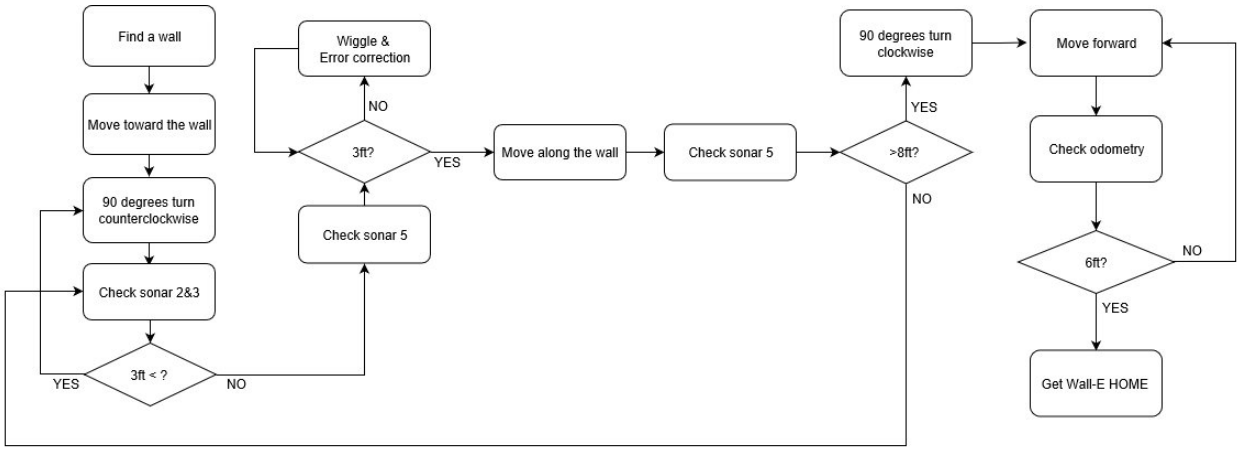

Figure 2. Flowchart illustrating the conditions, decisions and actions involved in the robot strategy.

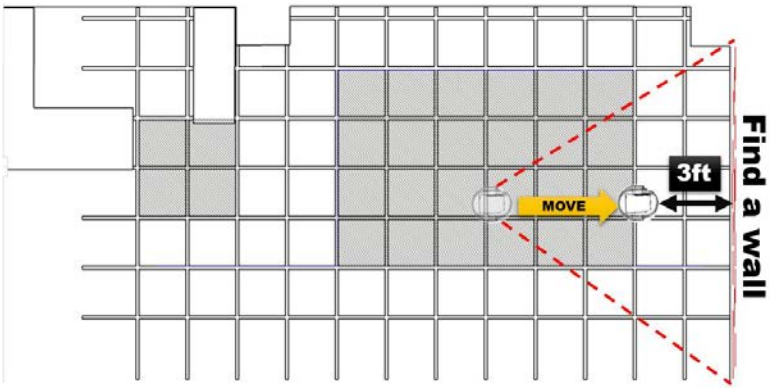

Figure 3. WallE identifying and moving towards the widest measured wall.

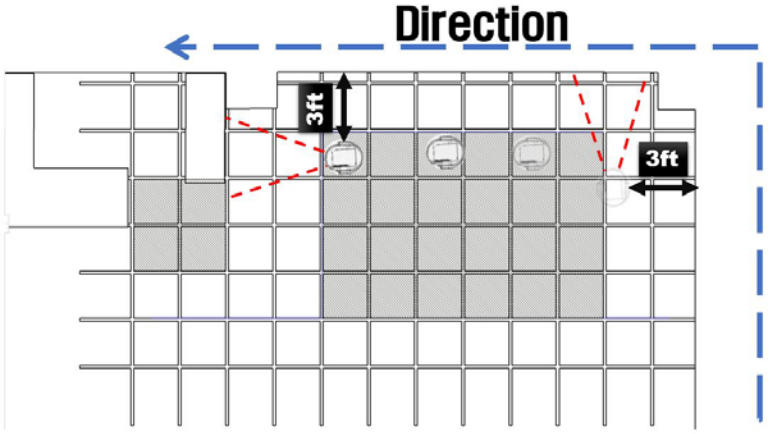

Figure 4. WallE following walls while maintaining distance and turning at corners.

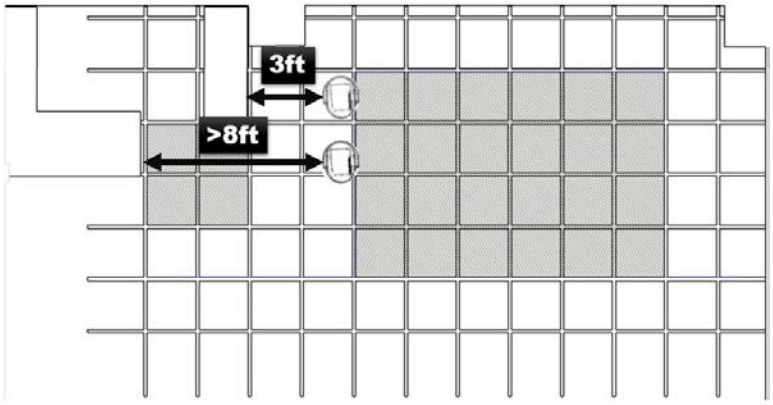

Figure 5. WallE detecting the termination trigger condition.

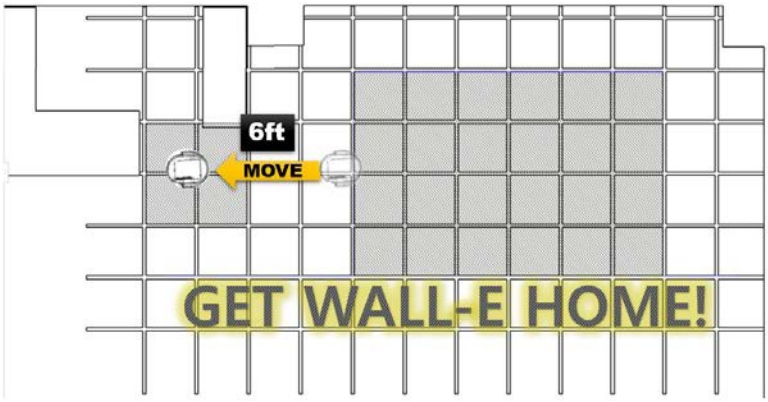

Figure 6. WallE moving to the destination.

Testing

Once we implemented the preliminary version of our algorithm, we tested and updated it. Some changes we made, aside from bug fixes, included:

- Having the robot always monitor its front sonars to avoid collisions.
- Tuning the wall following distance from 3ft.
- Improving and tuning the heading / wall distance error correction subroutine.

Technical Results

During early stages of testing, the test area had a cardboard sheet by a table, and WallE performed very well with it. However, this was later removed and the robot performed more poorly afterwards.

On the final day of testing, the robot had an approximate 60% success rate. One problem was that the front sonars had trouble sensing the table because of the gap beneath it. Another was that the performance of the wall distance error correction suffered, possibly because of code changes made on the last day.

In the demo, WallE lost its heading on the first run, and moved in circles around the room until the time ran out. On the second run, it detected the target trigger condition too late and stopped to the side of the goal area. The last run finished successfully, even though WallE initially appeared to be going off course.

Conclusions

Our project had a modest success during the demos but had potential to do better. If we could revisit the project, we would do more testing in the actual demo scenario. This includes testing the robot with more varied starting positions, because I think we unintentionally placed the robot in the same spots.

We would also improve our wall following algorithm. During testing, there were days where it performed amazingly, but its performance suffered later on.

Finally, we would reconsider how the robot handles unexpected cases. For example, if it gets too close to the table due to lost sonar signals, it could move backwards to the correct distance.