Product: UPLIFT.ED
Team: UPLIFT

Abstract

UPLIFT.ED ("Elevation Device", henceforth referred to as ED) is a robot designed to assist those with dexterity and mobility issues. ED's functionalities are to raise the contents of otherwise inaccessible drawers up to counter-top height and to also act as a platform for the user to place items to be raised up, for example shopping.

In this demonstration, we are outlining the steps that were taken in order to finalise ED as a prototype and have all its functions operating successfully.

1. Project plan update

The goals set for this demonstration are as follows:

- **Produce a revised version of our privacy policy** Achieved
- **Create new testing environments for ED** Achieved
- **Fully automate the navigation** Achieved
- **Complete the integration of the web app with ED** Achieved
- **Fully implement ED's main state machine** Achieved
- **Improve ED's Initialisation Sequence** Partially Achieved

1.1. Deviations from goals

Unfortunately, we were unable to fully implement our planned initialisation sequence, due to unforeseen complications and ultimately running out of time. We have, however, improved upon our previous initialisation sequence, and thus partial progress has still been made.

1.2. Achieved Goals

The goals we've been able to accomplish so far were achieved in good time and to a high degree of quality. The separation of tasks to be undertaken by sub-teams proved excellent, with each team working effectively within their own groups, but not limited to such, as advice was often shared.

The project manager, Ben, had initially created a privacy policy for the team but after consultation with the team of experts, has revised the privacy policy and created an updated version, featuring a more user-friendly and readable approach.

Andrew, a member of the robot sub-team, has worked on creating new environments for ED to be tested in, these new environments contain more intricate layouts for ED to traverse, which will aid in fully testing our system to make sure it is robust. Moreover, on the hardware side, Rafael designed an outer shell for ED which was also imported into Webots.

Sky, Hoffmann and Rafael have been implementing ED's operational sequences. Rafael rewrote ED's main controller, and implemented the marker alignment algorithm and the main lifting sequence. Sky re-implemented the camera calibration and made progress towards completing the initialisation sequence. Sky and Rafael also worked on the integration between ED and the app. Rafael also redesigned the web app, building on the previous design and fully implementing the integration to the main ROS software stack. The app is now able to display ED's current state and the user is able to run ED's full sequence from start to finish, including lowering and closing drawers as well as initialising a new room configuration.

Meiling, a member of the app sub-team, has been laying the groundwork for our team website during the preparation for this demonstration.

All code relevant to the demonstration was uploaded to the team's two GitHub repositories, which are organised as a ROS package and a React app respectively, so as to be easily accessible by the whole team. This allowed team members to work concurrently.

1.3. Future plans for ED

As we have reached the end of our process, there will be no further demonstrations. However, this does not mean that future plans for ED's design have not been considered. Here are a few points outlining what the team would plan to implement should ED have continued development after this project:

- Redesign the custom drawers to allow trays to be taken from the side, allowing easier access for wheelchair users.
- Conceptualise how ED's lifting procedure could be altered to allow for items to not only be raised from low down areas, but to also have items lowered down from high areas, for example to bring high drawers down to an acceptable height for wheelchair users.
- Explore the usage of ED outside of a kitchen environment, for example in a bedroom scenario to fetch medication for a bedridden user.

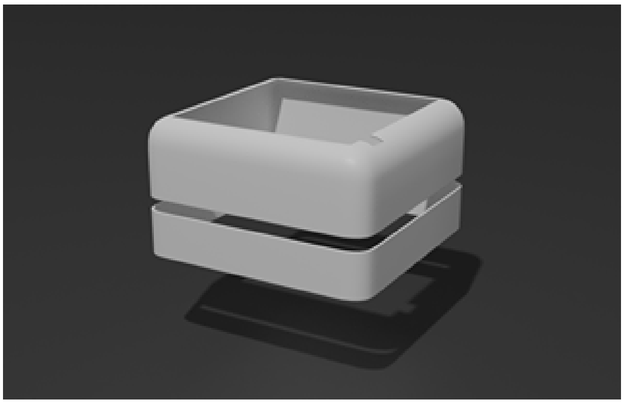

Figure 1. Custom shell casing

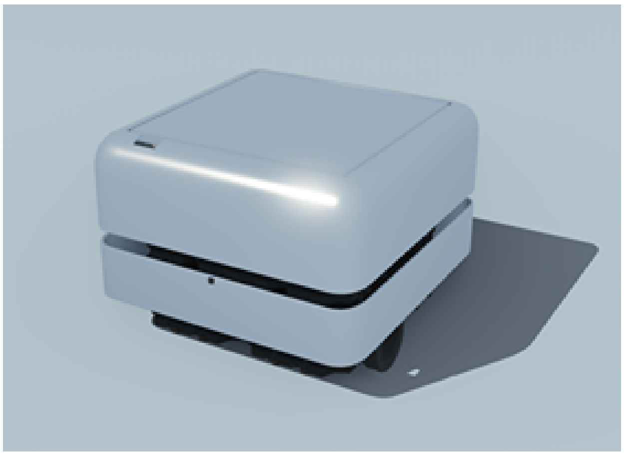

Figure 2. ED with custom shell casing

2. Technical details

2.1. Hardware

ED has been fitted with a new custom shell to surround the base unit and lifting mechanism as seen in Figure 1 and Figure 2. The shell has been designed to allow ED to remain, lightweight, streamlined and aesthetically marketable. The shell is custom fitted so that neither the LIDAR, camera or distance sensors are affected. Furthermore, the existing physical footprint of ED has not increased. By moving the legs of the scissor lift mechanism to the inside of the slider joint, the shell occupies the space previously taken by the lifting legs.

A new motor and linear actuator have been selected for use in ED's lifting mechanism. The force required to lift an average load of 10kg using ED's lift was calculated. The motor was selected taking into account the distance the screw driven actuator needed to travel, the force required and voltage and amperage of the power supply. As shown in Appendix A, the new linear motor far exceeds the required force to lift 10kg given that the minimum lift angle of the leg joints will be (when lifting the bottom drawer) no less than 7 degrees.

2.2. App

The app team and navigation team integration is now fully completed, with the app now fully able to connect with and send messages to ED - implemented using the standard ROS JavaScript library, roslibjs (Torris). The main mode of communication between the app and the ROS stack are the '/ed$_s tate',' /cmd_g oal',' /config', and' /init_m arkers' topics, where' /ed_s tate$

The user interface for the app has been redesigned again with a focus on accurately displaying ED's current state and giving the user full control. As such, when ED is performing a task, the user is able to cancel the action at any time using a "Cancel" button. An additional "Emergency Stop" button allows the user to fully stop ED immediately and is always displayed when ED is carrying out an action.

Upon opening the app, the user is presented with a page populated only by a welcome message and a "Connect to ED" button. Following a successful connection, the connection state is updated, and the user is redirected to the home page if an existing room configuration is found or to an interactive Setup guide otherwise. The home page consists of a visual representation of all present cabinets, including customisable names and icons, and the docking station. Swiping left or right allows the user to change between cabinets to find the one they wish to open. Upon selecing a cabinet, the user is presented with a choice of drawer to lift. The app is fully scalable and can support a theoretically unlimited number of cabinets and drawers, which are rendered based on the recieved configuration sent via '

2.3. Software

2.3.1. The New Environments

In order to more completely test ED's capabilities in real kitchen environments, we have designed two more simulated environments with features that are potentially difficult for ED to deal with, such as a kitchen island with a narrow walkway.

Kitchen 1, which can be seen in Figures 12 to 14, is designed using an L-shaped counter creating a more complex environment for ED to operate in. Drawers have been fitted in the kitchen which can open into the same space, requiring ED to use its distance sensor to ensure no obstacles are present above the lifting platform before opening a drawer. Additionally, obstacles found in a typical kitchen, such as a radiator and a bin, have been inserted into the testing environment creating additional challenges for ED's navigation system.

Kitchen 2, seen in Figures 15 to 17, is a larger environment with an island fitted in the centre. The island creates a challenge for the initialisation sequence and normal running as it blocks any direct path from one side of the kitchen to the other. The limited space on either side of the island creates further challenges for ED's navigation system to operate within a smaller space and still recognise any given marker.

We tested ED in these environments, the details of which can be found in Section 3.2.

2.3.2. The Main State Machine

ED's main state machine has now been fully implemented. He has three main processes: initialisation, lifting a drawer, and returning to the docking station. The execution of each of these processes is controlled by the app.

At any point during ED's operation, the user can cancel the process, in which case ED will return to the docking station, or push an emergency stop button, which will immediately halt all movement.

2.3.3. Initialisation Sequence

Initialisation is an important part of ED's ability to function. Our previous Initialisation Sequence involved moving randomly around the space and recording every estimated value of the position of each marker to average for the final position estimation. This was fairly successful, but could definitely be improved upon.

Our plan was to implement an initialisation sequence as outlined in Demo 2, the state diagram of which can be seen in Figure 7. In order to do this, we utilised the explore_lite package (Hörner), which will direct ED to move into unexplored space. We chose this package because it works with the navigation stack we already have.

Upon seeing a marker, ED would shut down the explore_lite process, execute its lining up sequence and then send the accurate location of the marker to be written to the location file. ED would then restart the explore_lite process until all expected markers had been located.

Unfortunately, we were unable to fully implement this process due to time constraints. Using explore_lite had the unforeseen side effect of shutting down the process if the map is complete, even if there are markers that had not been found. We planned to deal with this by writing a node that would send ED to random unoccupied spaces once the explore_lite node had terminated, but we ran out of time. No existing random walk node exists that can be easily integrated with our navigation stack, so we were unable to move forward with this plan.

Despite this, using the explore_lite node alongside our previous method of averaging all estimations is an improvement upon our old initialisation process. No formal evaluation was undertaken, but it was observed that the obstacle avoidance process was unable to navigate and effectively map smaller kitchens with a lot of obstacles, such as kitchen 1, seen in Figure (fig). The explore_lite process is able to fully map the space with no problems, as seen in Figure (fig).

2.3.4. Normal Running Sequence

The user can instruct ED which cabinet and drawer ED should open and lift. The App communicates with ED via the '/cmd_goal' node, which is read by the main controller that then starts the lifting sequence.

ED first moves to within 0.5m of where he knows the cabinet to be. This information is taken from the YAML file that was written during the initialisation sequence. This initial movement is done using action_lib, as detailed in previous demos.

Once a signal to stop lifting has been received, ED will lower its lift, and close the drawer, using the alignment sequence once more to move into the correct position. ED then returns to its docking station and reverses into position. This is executed much the same as the initial movement and alignment, with an extra step of turning 180 degrees and reversing 0.15m.

3. Evaluation

3.1. Effects of LIDAR Noise on Map Creation

Due to feedback given after our last demonstration, we have conducted tests of the robustness of the map creation system to LIDAR noise. Our usual running of ED features a coefficient of 0.043 for LIDAR noise, so we know that the system is able to cope with some noise, but we wanted to find out how much noise the system could cope with before becoming non functional. We ran the initialisation sequence with noise coefficients detailed below, which generated the maps that can be seen in Appendix C.

Due to the fact that map generation is either successful or not, we were unable to produce a graph of results. As an alternative, success or failure has been recorded in this table. Success is judged on whether the map shows the obstacles in the kitchen with reasonable accuracy, such as the L shaped counter, so that ED is aware of their presence and can move accordingly. Results of these tests can be seen in Table 2.

Test	Coefficient	Result
1	0.0	√
2	0.1	√
3	0.2	√
4	0.5	×

Table 1. Results for tests on the effect of LIDAR noise on map production

From these results it can be seen that ED's map creation system is fully functional up to a coefficient of 0.2. The mapping completely fails with a coefficient of 0.5, as can be seen in Figure 11.We consider this to be a success, because we have determined that a reasonable amount of noise for a system to put up with is 0.1, and this is well within the bounds of successful map creation.

3.2. Runs of Whole System

3.2.1. Initialisation

Unfortunately, since the initialisation sequence was finished late due to the previously described issues, we were unable to test the initialisation sequence to the extent that we would have liked. Our plan was to run the initialisation sequence multiple times in each environment, and evaluate based on the quality of the map produced (with 0.043 LIDAR noise) and the average distance between the observed marker positions and the true marker positions.

We have been unable to complete the marker accuracy tests, because this would have required a lot of data processing which we did not have the time to do. We have however been able to complete multiple runs to generate maps, examples of which can be seen in Appendix E, which were all successful. Our success criteria were that the entire space be mapped, and that obstacles are mapped with a reasonable amount of accuracy.

3.2.2. Normal Running

To test the normal running sequence, we instructed ED to open and lift five different drawers in each environment, and then lower the drawer and return to the docking station. The criteria for success was that ED successfully open, raise, lower and close the drawer without getting stuck or unbalancing, and navigate to and from the drawer successfully and within a reasonable amount of time. We did not set a specific maximum time because the time taken to navigate to the drawer depends on how far the drawer is from the docking station, and what obstacles there are and where.

The success rates for each environment are summarised in the Table below.

#	Kitchen	Success Rate
1	Original Kitchen	60%
2	Kitchen 1	80%
3	Kitchen 2	80%

Table 2. Results for tests on the effect of LIDAR noise on map production

Runs where ED failed to complete the task were usually unsuccessful due to inaccuracies in the navigational system, such as executing the "ED is stuck" process when ED was not stuck. These issues can be fixed by tweaking the parameters that are fed into the navigation system, therefore we are confident that these inaccuracies can be ironed out. On one occasion in our original Demo Kitchen, ED was unsuccessful due to a physics bug in Webots. This of course is not applicable to real life operation.

4. Budget

Our current cost estimation for ED (scissor lift system and TurtleBot) is summarised in Table 3.

Component Name and Amount	Estimated Cost (£)
Washer M5 10mm 1mm (x4)	0.15
M5 Rod 310mm (x5)	3
M5 Rod 30mm (x5)	0.96
M5 Rod 45mm (x2)	0.58
M5 Threaded Insert 7.2mm 12mm (x2)	0.32
Spacer M5 10mm 13mm (x4)	5
MDF 9mm x 1220 x 2440 (x2)	22
Turtlebot 3 Waffle Pi (x1)	1007.22
Consumable items (eg. Glue) (N/A)	10
Miscellaneous items (e.g. Wires) (N/A)	10
Scissor-Lift Actuator (x1)	81.99
Scissor-Lift Controller (x1)	50.20
Hook Motor (x1)	2.17
Total	1191.42

Table 3. The unit cost for ED, not including the custom drawers.

In order to gain an estimate for the cost of the custom drawer units, a unit from the website Better Kitchens was used as it approximately matched the specifications of our custom drawer. The unit chosen was the 500mm 4 Drawer Base Unit, which cost £126.56, not including surplus charges given by the site. To account for necessary custom parts, the estimate for our drawers will be priced at £150 per unit. As discussed in our previous demonstration, the installation of these drawers is eligible for a grant from the Scottish Government.

Lastly, there is the need for ED's charging station. The estimate price for this component will be a Husqvarna automatic lawnmower charging station, which costs approximately £218.11.

Taking into account all of these prices, and that it's expected that user will require an average of three of the custom drawer units, the total price for an implementation of our system would be approximated at £1709.53.

5. Video

The UPLIFT.ED product pitch video can be found on the team's University SharePoint page here.

References

Hörner, Jiří. explore$_l$*ite. URL.*

Torris, Russell. roslibjs. URL http://wiki.ros.org/roslibjs.

A. Lifting Calculations and Statistics

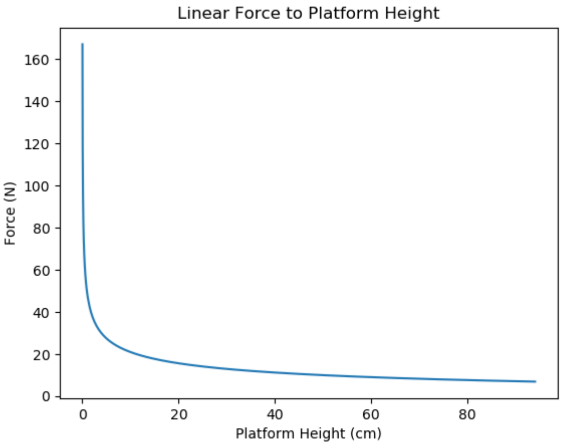

Figure 3. Calculated force to height

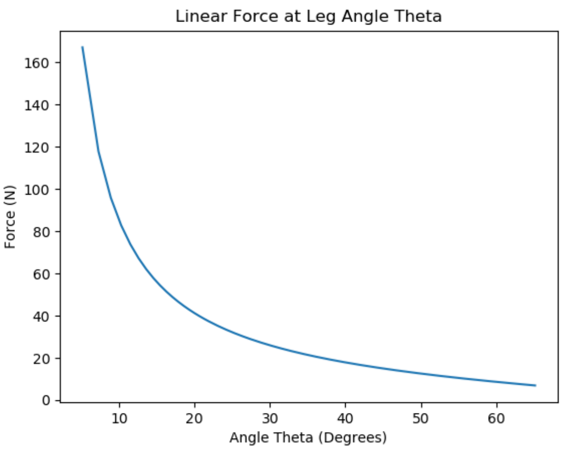

Figure 4. Calculated force to angle, minimum angle of lift while raising basket load is 7 degrees

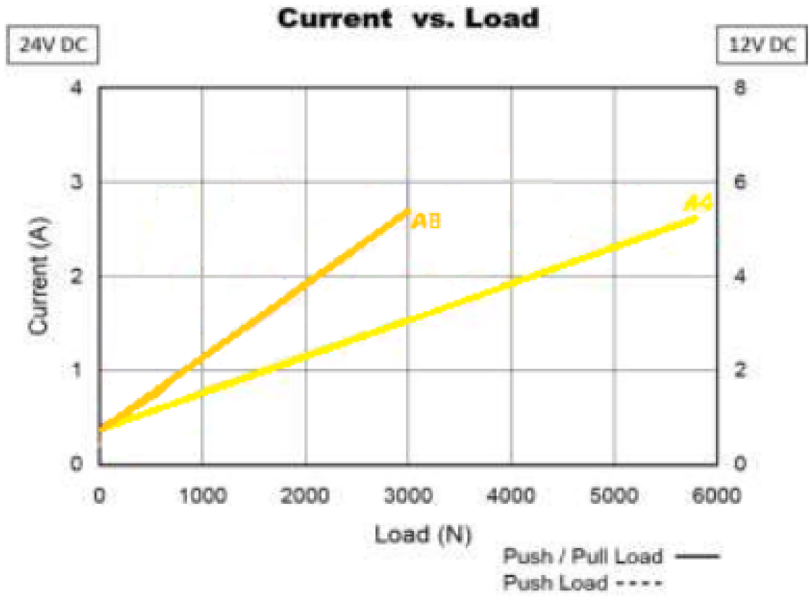

Figure 5. current to load output of linear actuator

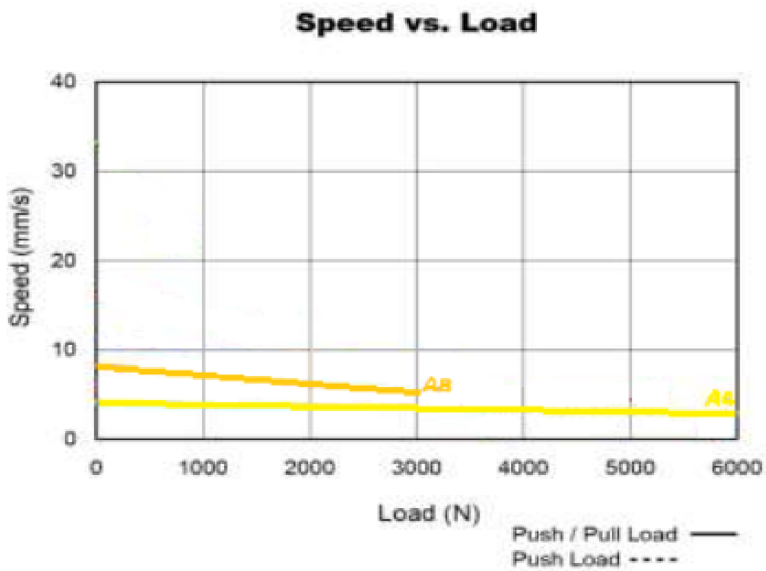

Figure 6. speed to load output of linear actuator

B. Initialisation Sequence Diagram

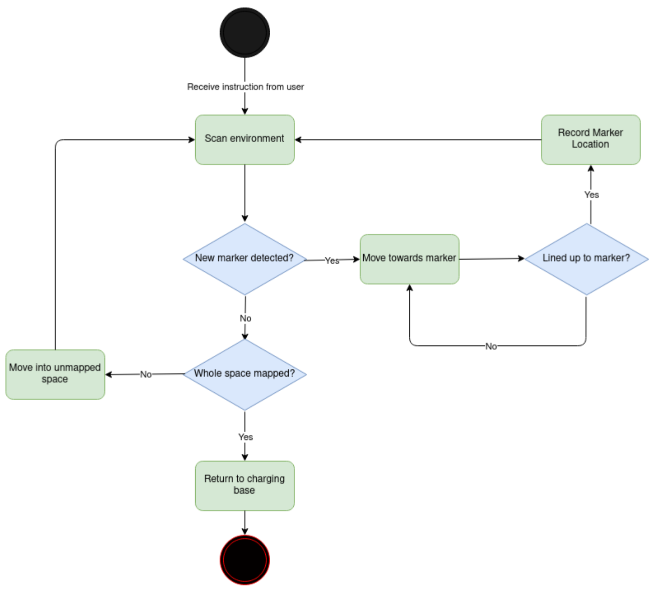

Figure 7. A flow chart describing ED's initialisation sequence.

C. LIDAR Noise Tests

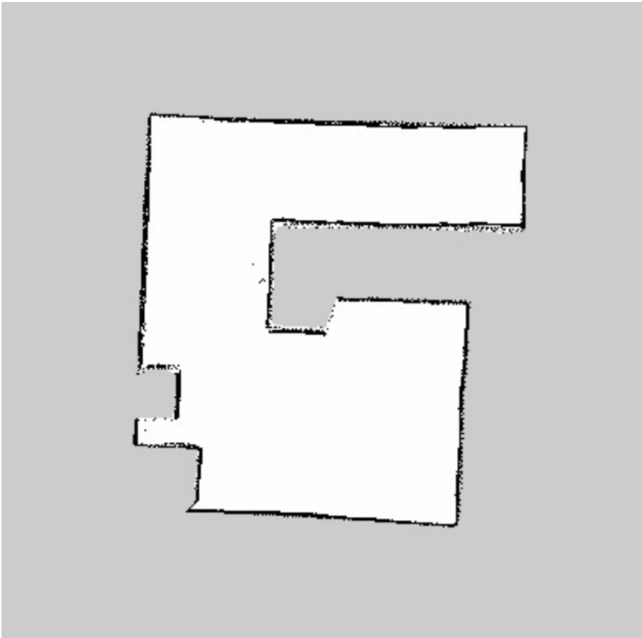

Figure 8. Map generated with 0.0 Noise Coefficient

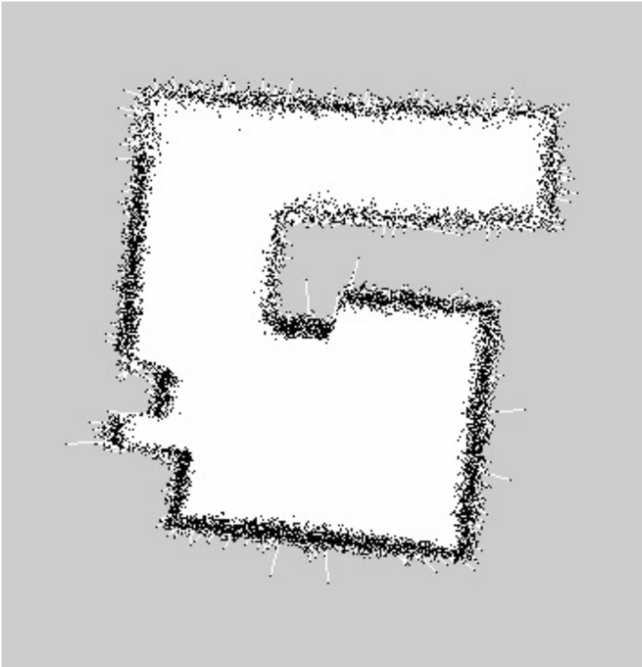

Figure 9. Map generated with 0.1 Noise Coefficient

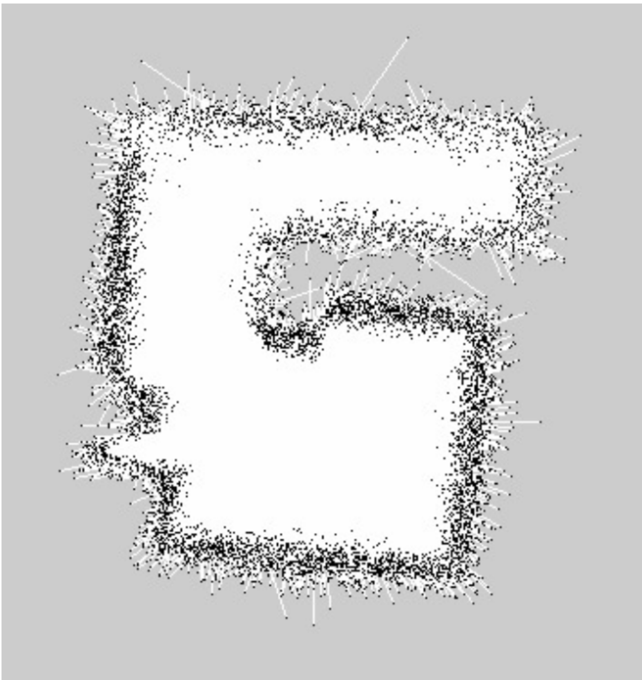

Figure 10. Map generated with 0.2 Noise Coefficient

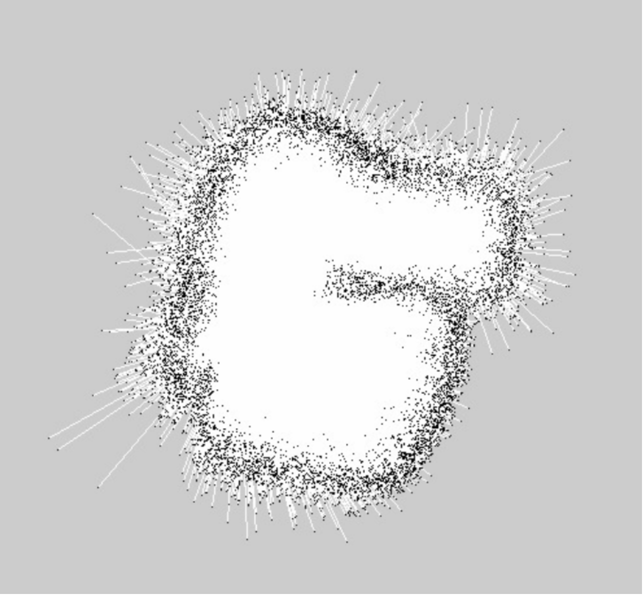

Figure 11. Map generated with 0.5 Noise Coefficient

D. Environment Images

Figure 12. The "Kitchen 1" environment developed for testing.

Figure 13. The "Kitchen 1" environment developed for testing.

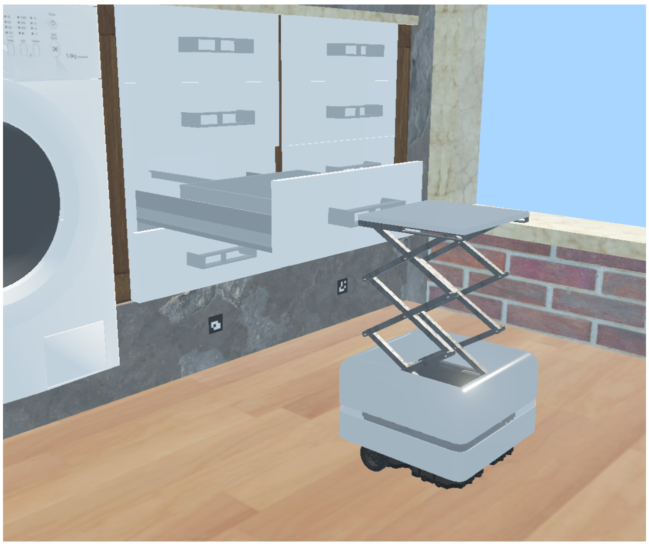

Figure 14. The "Kitchen 1" environment developed for testing.

Figure 15. The "Kitchen 2" environment developed for testing.

Figure 16. The "Kitchen 2" environment developed for testing.

Figure 17. The "Kitchen 2" environment developed for testing.

E. Running Test Maps

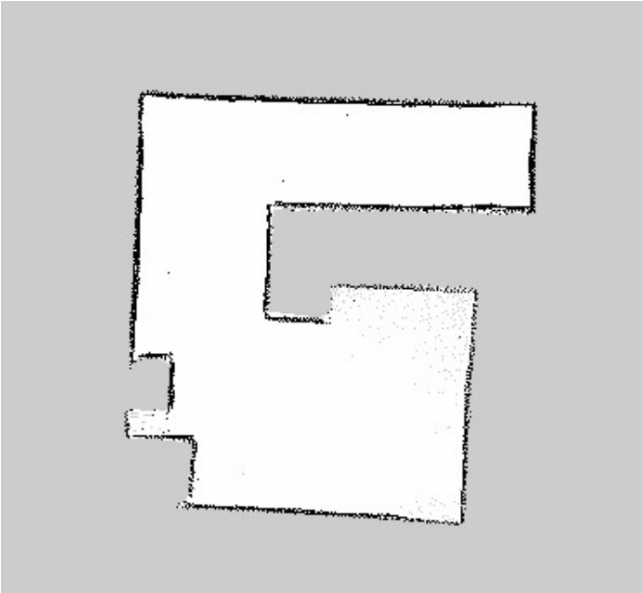

Figure 18. Map result for initialisation test on kitchen 1

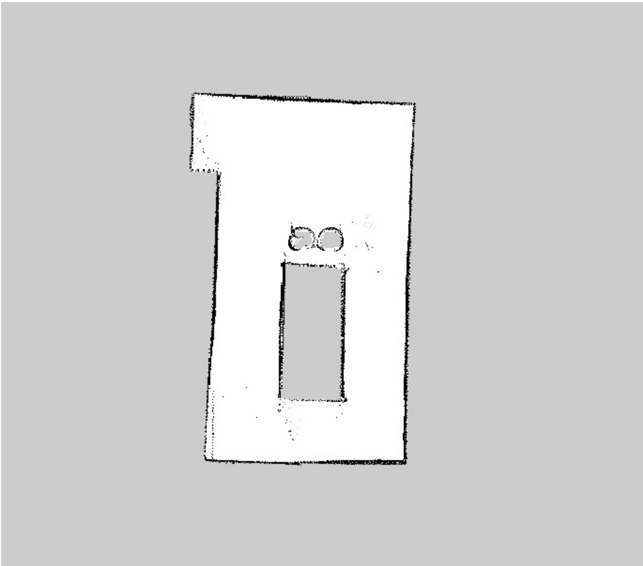

Figure 19. Map result for initialisation test on kitchen 2

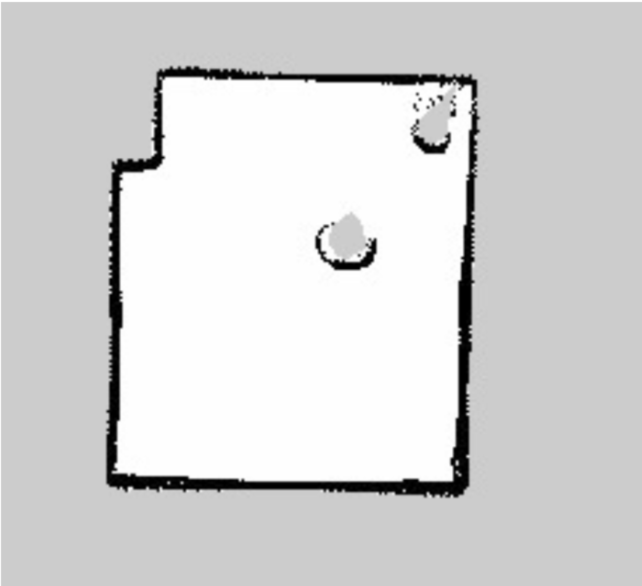

Figure 20. Map result for initialisation test on demo 1 kitchen