Product: ParkEd
Team: Back Benchers

Abstract

Our team is developing a mobile park bench which can move to and from the bench owner's desired locations. For this demo we have created a scale model of the bench which can perform basic motion. We have also created a basic user interface which can be used to control the bench movement.

1. Project management update

1.1. Demo 1 goals:

- A basic scale-model bench structure - achieved
- Basic motion carried out by the bench - partially achieved
- A mock-up of the user interface - achieved
- A map of the park and park boundaries - achieved
- Navigation methods research - achieved

1.2. Deviations from goal implementation:

One we ran into was that the rear motors weren't connecting to the front motors. We will have to add them to the turtle-bot 3 control table as this is how motors can be given instructions of when and how to move.

The shell of the bench structure took longer to commission than expected, as a result we implemented and tested basic motion of the underlying turtle-bot without the bench casing. However, we weren't held up by waiting for this.

We allocated a little too much time for researching, and didn't account for the fact that lots of research happens at the same time as development in our plan. As a result map research happened largely concurrent with park map implementation. Research on navigation and bench safety and security features had to be started earlier than expected, while the bench frame was being built as it would impact the shape of it. For example, the bench frame had to have a window so that we could include a LIDAR sensor in the model without disrupting its vision.

A number of our members had to self-isolate due to the Covid-19 pandemic. This required us to rethink our implementation plan. Two members were isolating for more than half of the time taken to prepare for demo 1. As a result, work was divided between that which could be done remotely and that which had to be done in person.

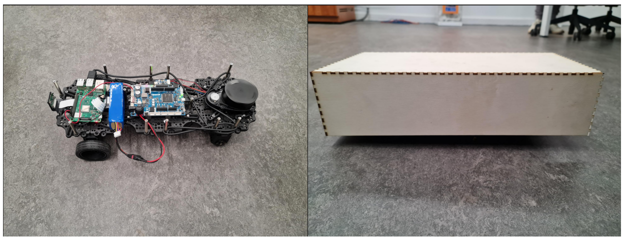

Figure 1. Bench robot with and without the casing.

1.3. Group and work organisation:

Our main method of communication was through meetings held in person. Meetings with the entire team were useful for distributing work and creating broad plans but as our focus shifted to implementation more time was spent on specific meetings with our respective subgroups where we shared newly learned skills and created focused plans.

The front-end team, tasked with designing the UI for controlling the bench, had a slow start. We would complete our individual tasks but encounter thorny merge conflicts wasting time and motivation. We quickly adopted a development protocol which took advantage of React's modular framework, structuring directories such that each component had its own directory with a corresponding .scss file with the same name. This approach greatly streamlined our development experience allowing us to parallellise our tasks.

The use of MS Teams was very useful and enabled subteams to collaborate remotely especially as two of our teammates suffered from covid during this phase. The 'channels' aspect of teams aligned well with our subteams and enabled efficient inter-subteam communication.

The back-end team, in charge of making ROS communicate with the robot, collaborated very closely with the hardware team. We first researched communication protocols within ROS and agreed upon the topics and messages we thought were most appropriate. To verify our choices, we made ROS simulations using these protocols. This allowed us to construct the robot, safe in the knowledge that the system would run. Because of this strict system, once the physical prototype was realised, getting the robot to move was a matter of plug and play.

1.4. Work allocation:

1.4.1. Front-End

- Rory (Leader): Being new to React, the development process involved a lot of learning, research and false starts. Many hours were spent communicating with other subteams, particularly regarding the exchange of information between the front and back end. Rory is the leader of his subgroup as he is a confident and personable communicator who is comfortable in such a role. **40 hours**

- Emily: Research and learning via online tutorials and LinkedIn Learning made up the first few days' work. That was followed by front-end implementation, park-map generation, debugging compilation, and attending team and group meetings. Remote working combined with fever, tiredness and other symptoms because of catching covid-19 made it difficult for her to participate fully. **29 hours**

- Ziqian: He spent a large proportion of time on learning the basics of web design and react framework and watching tutorials since he had no prior experience to them. He then designed the prototype user control web page. He worked collaboratively with other members of the team on the web page and made a basic model of the product web page. **34 hours**

1.4.2. Back-End

- Muiz (Leader): Initially, Muiz had no experience and knowledge in using ROS, so he spent quite a lot of time at first to learn the basics of ROS. After that, he managed to find a way to establish connection between ROS and our front-end server which is by using rosbridge_suite and roslib.js. He spends some time explaining the core concepts of how the communication protocols would work with the rest of the team. After that, he worked together with the hardware team for the actual implementation of the protocol on the robot. **38 hours**

- Abdul: He assisted Muiz in researching and implementing communication protocols that would make the backend (implemented a s ROS node) interface with the robot and the frontend. As the project manager, he hopped between all sub-teams to assist them, bridge communication gaps between sub-teams, and to ensure coherence between them. **40 hours**

1.4.3. Hardware

- Jack (Leader): Spent the first week building and gathering information about the robot requirements for the system. Initially configuring the robot to meet our blueprinted design followed by working with Muiz on agreeing on communication protocols. A significant proportion of time was taken up on adjusting and debugging the issues with the additional motors not appearing on the hardware control table. Still not fixed, its a primary goal and is proving very time consuming. Ordering the additionally parts for the robot were also included in his tasks. **43 hours**

- Suvi: Had difficulty contributing as much as they would have liked due to being ill with Covid-19 and having to self-isolate. Remote working was challenging, especially being part of the hardware team. They did some research the mechanisation of the case, and are starting work on getting fish eye distortion out of camera footage. **25 hours**

- Youwei: Investigated a potential mechanisation system for the robots outer shell. However without a case to mount any system on, he switched to designing the shell, working closely with Jack to make sure it would fit over the robot and its body. Once designed, he commissioned the cutting of the design with the technicians and assembled the case. Now working on the fitting of the case so that it is no longer loose round the body of the robot, he will later assist Suvi in the development of the mechanisation to raise case.**34 hours**

1.5. Budget allocation:

- Turtlebot Technician time (2 hours)
- MDF sheet (£7.5)
- Transparent acrylic sheet (£1.99)
- Laser cutter technician time (£1 @ £15 per hour)

  Total: £10.49

1.6. Plan modifications:

We have not made many modifications to our plan. The largest changes will be finishing the partially achieved goals from demo 1, which should be achievable in line with the original plan. We hope to achieve the following aims for demo 2:

We aim to have completed thorough ethical research and market research for the project by demo day 2. We would like the bench structure to include safety features and navigation sensors. We hope that the bench will be able to detect and avoid collisions while moving. We hope to have implemented a path-finding algorithm to determine how a bench should travel through the park. By this point we will also have a better understanding of how much power the bench will require so we can start researching methods of powering the robot.

2. Quantitative analysis and testing

Using React as the framework for our front-end, we implemented a console.log() to identify interactive component failures. As this is a large-scale project and we want finding faults in the system by means of testing to be easier. We introduced naive usability tests and we discovered instances in which we had to modify our own activity to suit the needs of the system and inconveniences that require added efforts from the users. We thought it would be useful to test the user-interface for readability, so the front-end team invited their flatmates to test the website. They all made

Test	Success
Robot base moves linearly and angularly	√
Hardware assembly matches blueprints	√
Robot Firmware controls all hardware elements	X
Website UI Displays all designed elements	√
Website posting data to movement data to rosbridge	√
Robot moves based on website data	√
Sensor data can be observed by rosbridge	√
Zoomable map shown on the web page	√
Joystick controller shown on the web page	√

Table 1. Results for 9 quantitative tests of the system.

Test Item	Time(mins)	√	X
Front Motors	10	5	0
Back Motors	60	0	7
React Joystick	15	3	2
Website Map Displays	15	3	1
Rosbridge connects	10	2	3
Robot control from Web	30	4	2
LiDar data reads	5	2	0

Table 2. Results from specific test elements. The table includes our test failures from previous testing and corrective testing to ensure robustness

a good point that the park needs to be highlighted with a clear outline in the map next to joystick so that the users can see where the boundary is. Once implemented this was also tested for clarity and robustness.

Our first and second tests of the robot system was to ensure our base frame robot could move and was assembled correctly. We did not succeed in getting full functionality since our rear motors are missing from the hardware control table, a firmware issue. Our assembled robot was still able to move with required precision. While the setback has affected directly the progress made with hardware development, it will not affect the progress that will be made on software development of the system at this stage.

Our third and fourth tests of the robot and system were to ensure the robot would be controllable through the full integration of the systems. Our website joystick and keyboard controls were robustly tested many times prior to integration and the website UI was naively tested to make sure it would display all key elements. This was a serious milestone as the team all understand how to integrate our system and develop it further. We repeatedly run the setup of the system and tested the control of the robot by communicating through the running web-socket. The testing also proved the we were still able to retrieve sensor data from the robot while it was being operated from the rosbridge back-end.

The testing performed does not cover all edge cases however the aim to this point has been to provide functionality and control over the robot in a limited fashion. However with the extent of the quantitative testing and the limiting of options to the user, the system will function as designed. Further development and testing will be executed in later stages to ensure that a naive user can perform the designed system tasks free of bugs and without a need for developer support.

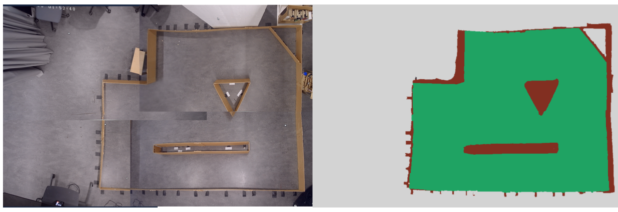

Figure 2. Lens-corrected pictures stitched to produce a map.

3. Budget

- RTK-GPS transceiver (£93-157) (rtk)
- Casing (£234 - £410 | Metal and Designation fee)
- Encoder Board (£35-40)
- Motor Encoder (£25)
- Onboard computer (£30-50)
- Battery (£50-£200)
- Track wheels (£40-£150 | 4 Wheels)
- Track links (£400-700)
- LiDar (£80-100) (lid)

£987-£1832 in total

4. Miscellaneous

In theory, a bench in our life-sized system will localise itself using RTK (Real Time Kinematic) GPS (Morales & Tsubouchi, 2007). Though the actual bench unit will operate outside, we are confined to testing it in the SDP space which, being indoors, is not an appropriate environment for GPS. Still determined to demonstrate the localisation functionality, we have devised a solution: We can simulate a stream of GPS coordinates using the overhead cameras that calculate the bench's position and send to the robot. This requires some non-trivial robot vision work but this is an area in which our team is confident. We hope to have this implemented by the next demo.

References

Lidar, laser scanners rangefinders. URL https://www.robotshop.com/uk/lidar.html.

What is gps rtk? URL https://learn.sparkfun.com/tutorials/what-is-gps-rtk/all.

Morales, Yoichi and Tsubouchi, Takashi. Dgps, rtk-gps and starfire dgps performance under tree shading environments. pp. 519–524, Shenzhen, China, 2007.