Product: ParkED
Team: Back Benchers

Abstract

Our team has developed a mobile park bench which can move to and from the bench owner's desired locations. Our prototype is a custom built bench suitable for outdoor use. It has a custom navigation stack with global and local planning for obstacle avoidance, automated PID controllers for smoothness, and filtered odometry for robustness. It uses centralized planning to control multiple agents at once. The system uses simulated GPS coordinates and pressure sensors to track benches and collect data on bench usage for park optimisation. We also have a functional frontend which can assign configurations to multiple benches and visualise the collected data.

1. Project management update

1.1. Unachieved and partially achieved goals

- Ultrasonic obstacle detection - *Partially Achieved* - We implemented ultrasonic obstacle detection and local path planning based on the Edge Detection algorithm(Borenstein & Koren, 1988), but due to hardware complications with Raspberry pi 3b+ we could not attach more than 2 ultrasonic sensors; therefore, our demonstration of dynamic obstacle avoidance relied on the global planner (2) for the robot to plan an alternate route in case of any unmapped obstacle(4.4.3).

- Track mobility - *Partially Achieved* - Despite efforts, and many printed models, we were not able to attach tracks to our robot and were forced to use wheels. The tracks we printed would not hold in the screws properly. This has had knock-on effects as the wheels caused a significant change in the dimensions of the bench, and so the case we made does not fit. For this reason, we were forced to make last minute adjustments to the case and its elegant design has been compromised.

- Weatherproofing - *Unachieved* - The changing robot dimensions prevented the case we had made for the robot from fitting.

- Safety Features - *Partially Achieved* - We were not able to create an emergency stop button as we had wished. However, we have implemented obstacle avoidance and the bench will not move if it is occupied, as detected by simulated pressure sensors.

- Authentication - *Partially Achieved* - We managed to finish an individual login authentication design before the final demo, yet we ran out of time integrating it with the current front-end design. We will work on it and have it ready before the industry day.

1.2. Deviations

In the lead-up to the final demo, we ran into unforeseen hardware issues, solving which took up a lot of time. This meant that we did not have time to fully implement all of the features we had planned to include. These are explained in Section 1.1

1.3. Group and work organisation

Soon after starting the project, we organised ourselves in to subgroups. The number and makeup of these groups changed over time, but having smaller organised groups helped us be more efficient and focused.

Throughout the project we had in-person meetings three times a week, which we found to be important for organisation, work allocation and communication. Subgroups also met frequently among themselves to work on more specific plans and tasks.

We also used MS Teams, which proved to be very useful for file sharing and communication both among the whole group and within subgroups. It also allowed team members to join meetings remotely, which was especially important when some of our members were suffering from Covid-19.

To ensure all of the code written by different subgroups worked together, we found version control software to be very important. This is why we used git, specifically github.com, from early on in the project. We organized all of our code into a Github organization to ensure effective collaboration.

1.4. Work allocation

1.4.1. Global planner

- Rory worked with Emily on the design and implementation of the path finding algorithm. Specifically programming the functionality that reads map data and processes it into format that is suitable for the A* algorithm, whilst adhering to the constraints of the domain. Much time was spent altering the domain of the planner to fit the subtle, nevertheless crucial, changes in direction that our project has undergone.

- Emily worked with Rory on the global planner by implementing the A* search algorithm, testing and debugging it. She also worked on integrating the frontend with the navigation stack so that the flow of control to the robot worked.

1.4.2. Local planner

- Ziqian designed the obstacle avoidance algorithm and did some testing on simulation while the robot was not available. Later he researched multi robot conflict avoidance and integrated a multi-agent path finding algorithm with our own system, with help from Abdul.
- Muiz spent most of the time during the project working on the local planner. Initially, the robot was not moving very well due to some hardware issues, so while waiting for the hardware team to work on the robot, he did a lot of research on how to further improve the local planner and obstacle avoidance algorithm. He implement some changes and tested it in simulation and decided which one worked best for our current hardware. After that, he worked on integrating and modifying the local planner on the actual robot with the GPS simulation system. Lastly, he spent some time collaborating with the global planner team for the dynamic obstacle avoidance algorithm.
- Suvi worked as part of the local planner team, specifically writing code for the main flow of the local navigation algorithm.

1.4.3. ROS

- Abdul designed and developed the ROS architecture for the proprietary navigation stack for ParkED. This included defining custom ROS messages, actions, and services that would be used for communication between all sub modules of the system and defining a node-level architecture inspired by ROS 2's(2, 2020)(Hansen, 2019) and ROS 1's(Gill, 2018) navigation stack and some latest research in the field(Marin-Plaza & Hussein, 2018). He also wrote ROS nodes for the hardware to interface with the software system, and worked with almost all subteams to ensure seamless integration and decent abstraction between different components of the system.
- Muiz worked on establishing connection between ROS and our front-end server which is by using rosbridge_suite and roslib.js, at the early stages of the project. He implemented a JavaScript class specifically for connection purposes to be used by the Application team. He also implemented a secure protocol for the connection using rosbridge_server, so that only the our website could connect to ROS master that are running on our robots.
- Jack helped research ROS features and functionality we could use to help our robot talk to the rosbridge server as well as develop nodes for much of the physical devices so that our custom robot could be controlled with our own custom architecture.

1.4.4. Hardware

- Jack was tasked with seeing the bench Hardware, Design, and drivers implemented. performing many tasks such as wiring, soldering, programming drivers, creating firmware, programming sensors data capture, creating ROS nodes and researching protocols that different hardware devices would use. The most difficult task was keeping methods and classes use-able, well documented while protecting data critical data from being altered.
- Abdul assisted Jack throughout hardware development. He also implemented the PID controllers for the robot in order to ensure smooth and jitter-free movement of the robot even at slower speeds. This was particularly challenging as he had to interface with the robot at low-level where he had to control the current and voltage to set the speed of the robot.
- Youwei worked on the 3D modelling of the robot. He designed and 3D printed the wheels, as well as sourced the tracks and added wheel sleeves. He also designed, modelled and laser cut the outer casing, back rest and arm rests, and created the enclosing boxes for both robot models.

1.4.5. Application

- Emily spent a large amount of time working on the front-end react application. She created a responsive website with multiple pages to tell the story of the product by including research about ethics and commercialisation. She implemented a live map using mapbox, which interacts with ROS to show the park and all of the contained benches, updating their positions and states in real-time. She enabled sending instructions from the frontend to ROS, and receiving data from the back-end including live heatmap data and bench statuses.
- Ziqian designed an authentication system in react that provides privacy to the robot controlling page.
- Rory - Worked on the back-end of the heat map, handling the data passed from the pressure sensors and GPS and processing it into a form that can be displayed on the application as a heat map.

1.4.6. Research

- Rory - Reached out to the wider bench community to gain insight into what park councils (our main market) look for and are likely to criticize in a product like ours. Specifically, he arranged a meeting with Steven Cuthill of Edinburgh council who gave us his welcome and qualified perspective on our idea.
- Suvi spoke with an expert about the possible ethical implications of the project. They also carried out a survey identifying user concerns and benefits of the project.

- Emily researched the ethical implications of our bench, including how it might safely traverse parks and how to prevent malicious use of data collection. She also researched the potential for advertising on the bench as a means of increasing its commercial value.

1.4.7. GPS System

- Jack and Abdul created our own proprietary GPS system simulation using technical hardware already situated in the SDP Demo space. They developed a system which would capture 4 camera feeds, take their image contents, undistort the fish-eye and then stitch together to create a 90% accurate representation of the demo area at 30cm from ground level. This system would allow us to track and calculate the angle of the robot around the space using colour dots. This system was then later adapted to capture occupancy data for part of our smart bench usage metrics.

1.5. Total Project Hours

- Abdul **Total: 200hrs**
- Jack **Total: 200hrs**
- Muiz **Total: 180hrs**
- Emily **Total: 170hrs**
- Rory **Total: 150hrs**
- Youwei **Total: 150hrs**
- Ziqian **Total: 130hrs**
- Suvi **Total: 120hrs**

1.6. Budget allocation

- MDF sheets (4 @ £7.50 Each)
- Wooden sheet for box, straight gear, armrest and backrest (£9.5)
- Plastic for 3D printing of 10 track wheels and 6 hollow cylinders (£10)
- 3D printing technician time for 10 track wheels and 6 hollow cylinders (£5)
- Laser cutter technician time (0.5hr @ £15 per hr) **Total: £62**

1.7. Overview of achievements and problems

Throughout this long process, particularly at points of hopeless failure, our team has learned many valuable lessons. At the start of our journey, we suffered from a lack of direction. Though we had a product in mind, we lacked, crucially, an understanding of how our product would assist our target customer. Learning from these experiences and from interviews and feedback, we reorganised our team, setting tangible goals and deadlines which related to the value that our bench contributed to the customer.

We had to redesign our whole robot after demo 1 to make it more suitable for outdoor use. This required building a custom robot with a custom navigation stack (Hansen, 2019). While this set us back, it also allowed us to customise our robot to our needs. We are proud of the system we produced and we learnt a lot by having to get into the nitty-gritty details of navigation.

We had problems achieving initial goals of making our system weatherproof with safety features, due to the unexpected changes in robot size.

We considered the feedback we have received throughout the semester and focused our efforts primarily on functionality that communicates to our client how our idea fits in with the big picture. Real-time data analysis of bench-use density and exemplification of multi-agent bench systems.

Though some of the goals we had set for ourselves were not achieved, we feel satisfied as a team that we have delivered an effective prototype that addresses our target market, and demonstrates how ParkED benches can assist them in improving the comfort of outdoor green spaces.

2. Quantitative analysis and testing

Test	Success
Nodes in path never cross graph constraint	√
Path returns 'None' if start node is on the boundary	√
All nodes in path are within boundary	√
All consecutive nodes in path don't intersect boundary	√
Path returns 'None' if given end node close to obstacle	√
Path returns 'None' if given start node close to boundary	√
Path is found in under 10 milliseconds	√
Form user information collection	√
Data storage and management in database	√
Persistent login state	√

Table 1. Results for unit tests run on the global planner software.

Global Planner For the global planner, which finds a global path through the premapped area, we have designed a test suite using python's `unittest` library. As the system is large with many moving parts, it is important that the code fails in predictable and readable ways if provided with invalid input. Tests like `test_reject_out_of_bounds` and `reject_if_end_in_obstacle` assert this, ensuring that an appropriate error message is given and the path returns 'None'. This None value is interpreted by the node that calls the global planner, which sends the message concisely to the user.

It is difficult to directly test for correctness in a pathfinding algorithm, as finding the optimal path by hand is very resource intensive and error prone. Instead, to test this functionality, we have ensured that the path provided was valid under all input conditions including many edge cases (e.g. constraints such as empty list and various start and end positions). We have defined a valid path as a path that does not enter the buffer zone, the nodes of the path are all within the boundary, and the path never crosses a constraint that it has found. Results for these tests can be found in Table 1.

GPS and Heat Map We ran tests to ensure that the information taken from the GPS system, that contains the bench state and location, was accurate and reliable. See Table 3.

Authentication We built tests that provide user information and ran them against the system to ensure its safety, since the main objective is to avoid malicious usage. We also included functional testing for the validity of user information.

Local Planner To evaluate the feasibility of our local planner, we ran a series of tests in simulation. Though this helped us develop our algorithm, we found it important to run tests on the actual robot so that we could tune our values such as the speed and distance tolerance to integrate it with the whole system.

We ran various paths with obstacles coded manually by referring to the GPS simulation system, and ensured that they followed the expected path. Running manual paths allowed us to test the planner in isolation from any potential confounding data from the rest of the system. Running these tests uncovered a huge delay in the reading for the GPS simulation, which varies around 1-2 seconds. This was one of the biggest

PID Tuning To create smooth and continuous motion in the robot, we implemented a PID controller with Terrain difficulty memory. To tune the PID controller to any weight of robot and battery voltage we first run a PID Calibration loop where the robot will use the P, I and D yields to move the robot at a slow but stable speed. Once the 4 elementary motions have been executed. we would test the robot on different terrain and different resistance to movement. The Accumulated Integral error for each of the elementary motions would be adjusted until the robot drives over the terrain. Due to the nature of the domain, the robot can experience linear and angular drift where the robot would move more or less than the amount specified which would accumulate over many small movements. Below is the table for the tests conducted to analyse and reduce this drift.**??**

3. Budget

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

£987-£1832 in total

Test	Test Runs	Drift (*Degrees)	Pass?
10m Straight Line	20	5cm	√
30s Drive	4	15m	√
10m Reverse	20	5cm	√
30s Reverse	4	12m	√
180 Degree Right	10	3*	√
30s Turn Right	4	518*	√
180 Degree Left	10	5*	√
30s Turn Left	4	473*	√

Table 2. Robot angular and linear drift over a course of tests to measure suitability within domain

Test	Success
'sitdown' message received in server when red circle covered	√
'sitdown' message received in server when yellow circle covered	√
'sitdown' message received in server when green circle covered	√
Location of bench recieved on frontend accurate to 3cm	√
Bench state displayed on UI consistent with actual state 5 times in different positions close to obstacle	√
heat map displayed on UI consistent with trial bench sittings 5 times	√
UI updates bench state when it changes	√

Table 3. Results for quantitative analysis tests run on the heat map system. This ensures that the information taken from the GPS system is relayed reliably and accurately to the frontend.

4. Miscellaneous

4.1. Ethical research

In the lead-up to demo 3, we decided to do some ethical research which we had initially planned to do earlier on. We wanted to find out if there were any ethical concerns we should take into account that we had not yet thought of ourselves.

To do this research, we first met up with the expert on AI Ethics and Society, Benedetta Catanzariti. We got some good insights from this meeting, as well as advice on how to conduct a user survey.

4.1.1. Survey

We carried out our survey through Office Forms, and found our participants through the SDP students mailing list. The survey was anonymous and asked general questions about experience and opinions on park seating, as well as asking for feedback and any concerns relating to our project.

4.1.2. Findings

Many respondents seemed excited by the idea of mobile benches, particularly because they valued location as well as comfort, weather and amenities when finding seating in parks. It also showed us that people had a number of safety concerns, such as being bumped into or the bench moving while being seated - concerns which we addressed using pressure sensors and obstacle avoidance measures. Through talking to Benedetta we identified some privacy concerns regarding data collection and ensured that we used no cameras, collected only anonymous data, and avoided malicious or antisocial use. We also believe that live updating bench occupancy data could pose safety issues, so we would publish this data at intervals instead.

4.2. Commercial Concerns

In order to develop our understanding of our target customer, we reached out to park councils and campus managers. We were fortunate enough to get a reply from Steven Cuthill of Edinburgh City Council who gave us valuable perspective. Steven was excited by our idea, offering us development space to test the product in full scale. He told us of Edinburgh council's difficulty with pedestrian counting in green spaces - they would like accurate data on how the park is being used but sensors for public services are usually mounted on doors; parks don't have doors. We noticed that our benches could satisfy this demand providing information that would inform not only bench positioning but also funding allocation. He reassured us that Edinburgh is currently investing in smart technologies, for example self driving lawn mowers that would be delivered to grassy areas by drone. This gave us confidence in the marketability of our system.

4.3. Data Collection

4.3.1. Heat Map

From the outset of our project, we knew that these benches could be used to collect rich information about how the parks that they inhabit are used. For demo 3, we have developed a way to collect and display this information in a way that is integrated both with our project architecture and our user interface. When a bench is sat on, the on-board pressure sensors send a message to our a server which processes and (temporarily) stores information about this interaction - namely, the time and duration. This is then passed to the frontend which displays over the map of the park, a bench-use-density heat map to the user at the push of a button. We identified a heat map as the most appropriate way to show this information, as it is instantly readable and integrates seamlessly the a map that we were already using. Once the heat map is overlaid, the user is free to use the normal map controls, scaling, selecting benches and moving their positions without any change to the user experience. This elegant feature is a powerful tool that does not rely on instructions or clog up the user experience.

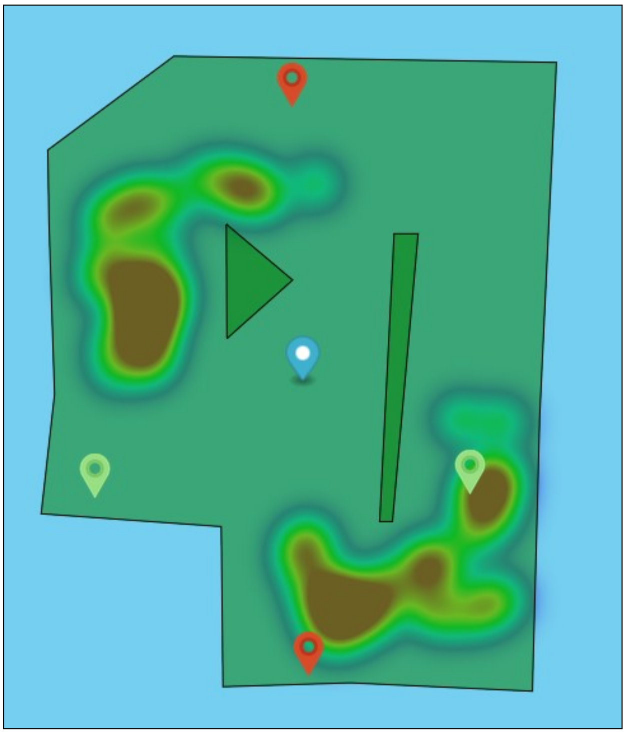

Figure 1. Heat map visualisation of the distribution of bench use. This is an example of what the user will see on the website. The pins represent bench positions and their colour represents their state.

4.3.2. Implementation and presentation

Implementing the the heat map system, such that it could be presented within all of the constraints of a demo, required some creativity. Our first challenge was that we only have one robot - the heat map functionality depends on having multiple benches whose collected data contributes to a larger picture of park usage; collecting data from only one bench wouldn't satisfactorily communicate this use case. Addressing this problem, we represent a series of dummy benches as a circular sheets of coloured paper - a different colour for each bench. The overhead cameras, which mimic the effect of a GPS, detect the position of each uniquely coloured circle. We cover up a circle to mimic sitting down on the bench. The camera system recognises this and sends the appropriate signal to the server which is ultimately displayed as a heat map to the user.

4.4. Navigation

4.4.1. Obstacle Avoidance Algorithm

In order to move the robot accurately and safely, without colliding with moving obstacles such as people and animals, we required a complex and effective algorithm. Such

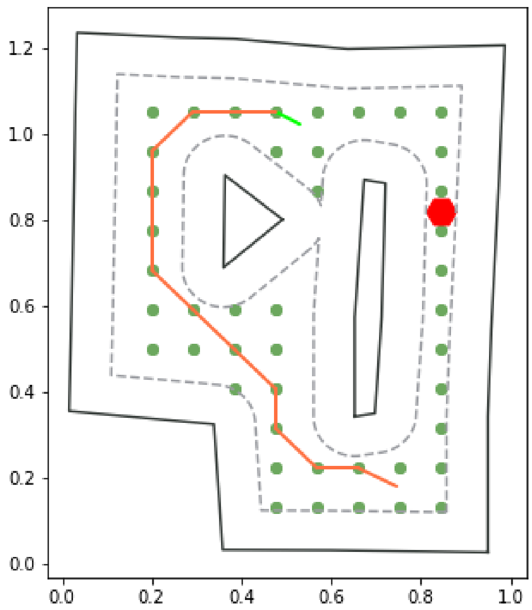

Figure 2. Figure showing the path planned by a robot through a constrained graph. The constraint would be passed to the server through ultrasonic sensors (such sensors are missing on our robot, see Section 4.4.2 so we are simulating this by manual command). Notice how the path avoids the obstacle plotted in red

algorithms have been discussed extensively in robotics literature. However most of them, for example the Potential Field and Vector Field Histogram method (Borenstein & Koren, 1991), are implemented and rigorously tested with a lot of sensors to get an almost 360 degree field of view around the robot. Our bench is designed with only a small number of sensors, so we identified a variation of the Edge Detection algorithm (Borenstein & Koren, 1988) as the most appropriate. It can be implemented with at least 3 ultrasonic sensors, at the front, left and right, to avoid a dynamic obstacle. We implemented this algorithm, and although it ran in simulation, we encountered issues with the integration of ultrasonic sensors (4.4.2), so we could not integrate it into our system. Despite this, we still found a way to navigate around unknown obstacles, relying on the global planner to re-plan the path if it detects an object in its path. The details of this process are explained in Section 4.4.3.

4.4.2. Sensor Limitations

Another important aspect of implementing robot movement is the accuracy of the sensors, such as the ultrasonic sensors for detecting obstacles, or GPS sensors which are used to keep track of the robot's current position. Ultrasonic sensors are well known for their inaccuracies and inconsistencies (Borenstein & Koren, 1991). For example, the sensor might not give out accurate readings in front of an irregularly shaped object, as the sound wave from the ultrasonic sensor would be reflected poorly into its receiver. A GPS system may also not be accurate when there's an object (e.g. a tree) blocking the radio waves. We considered this problem, and implemented a local planner algorithm that tries to reduce the effect of these sensor limitations by giving the bench a sort of autonomy that can handle these outages, making use of two sources of odometry.

4.4.3. Contingency Planning

The global planner receives the start and goal positions of the bench from the frontend and, using a representation of the park, calculates the most efficient path through the space (Zeyad Abd Algfoor, 2015) (Hart et al., 1968). see Figure **??**. It sends this list of coordinates to the local planner. If the local planner detects an object blocking the path that it cannot move around, the global planner is called again, this time with a constraint representing this obstacle. The global planner, if called with a constraint, modifies its graphical representation of the space and returns, as before, a list of coordinates representing the new most efficient path to the goal, see Figure 2. We have slightly updated the graphical representation of the demo space. This was to account for the increase in our robot's size - it can no longer fit through the little gaps that it used to.

4.4.4. GPS Simulation Delay

One of the major challenges we faced while implementing accurate movement of the robot was the 1-2 second delay from our GPS simulation using the overhead cameras. We discovered that this was caused by the intensive calculations being done to detect the robot's location with computer vision. We rely heavily on the GPS to give out the robot's current heading and position to plan out the next movement. Initially, when the robot is spinning, we stop it exactly when the GPS passes in the target heading value to the local planner, but due to the delay, the robot is always off by around 40-50 degrees. To overcome this, we calculated the robot's angular velocity manually, based on the time required to spin to a particular heading. We check with the GPS readings after a few seconds in accordance with the delay, checking if it is at the correct heading. If not, we try to spin again to the target heading until we reach the desired angle.

References

2, Ros. Navigation concepts, 2020. URL https://navigation.ros.org/concepts/index.html.

Borenstein, J. and Koren, Y. Obstacle avoidance with ultrasonic sensors. *IEEE Journal on Robotics and Automation*, 4(2):213–218, 1988. doi: 10.1109/56.2085.

Borenstein, J. and Koren, Y. The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Transactions* *on Robotics and Automation*, 7(3):278–288, 1991. doi: 10.1109/70.88137.

Gill, Jasprit S. Setup and configuration of the navigation stack on a robot, 2018. URL http://wiki.ros.org/navigation/Tutorials/RobotSetup.

Hansen, Matt. Navigation2 overview, 2019. URL https://roscon.ros.org/2019/talks/roscon2019_navigation2_overview_final.pdf.

Hart, Peter E., Nilsson, Nils J., and Raphael, Bertram. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968. doi: 10.1109/TSSC.1968.300136.

Marin-Plaza, Pablo and Hussein, Ahmed. Global and local path planning study in a ros-based research platform for autonomous vehicles. 2018. URL https://www.hindawi.com/journals/jat/2018/6392697/.

Zeyad Abd Algfoor, Mohd Shahrizal Sunar, Hoshang Kolivand. A comprehensive study on pathfinding techniques for robotics and video games. *International Journal of Computer Games Technology*, 2015. URL https://doi.org/10.1155/2015/736138.