Product: TransportED
Team: AutomatEd

[1]

Abstract

TransportED is a service which helps you automate your warehouse and aims to knockout the biggest barrier to entry - the massive infrastructure investment. Instead of a cost prohibitive capital expenditure we aim to provide a swarm of autonomous robots, powerful software and our engineering expertise as a cost-effective and quick to deploy service that integrates seamlessly in a variety of warehouse environments.

In this demo, we have fully integrated the scissor lift with our robot without bugs. We spent a lot of time conducting market research, discovering the advantages of TransportED, meeting with industry experts and further improving the robots to make TransportED more competitive in the automated robot delivery market. The user interface of the warehouse client website has been greatly improved. The map generator interface itself has also been optimised and made far more user-friendly and we've successfully linked our database with the web server. All the subsystems are now joined together and work perfectly in conjunction with the overall system in a Webots simulation. We completed a detailed user guide for our robot, while the product website is still under construction.

1. Project plan update

- Scissor Lift Integration — Achieved
- Market Research — Achieved
- UI design — Achieved
- Map generator interface — Achieved
- Integration of database and simulation — Achieved
- User Guide — Achieved
- Industry day website — Partly achieved

1.1. Deviations from original plan

We have completed all of the tasks as planned and implemented the adjustments mentioned in the previous report. The only thing we have yet to complete is the industry day website

So far, we have added sufficient contents to the *homepage*, *Our system* and *Team* column. However, there are still three parts to fill in, namely **How it works**, **Evaluation** and **Budget**, which includes detailed information about design specifications, software structure, and robot performance in the simulation together with a user guide on how to use our robot. The content has already been generated but still needs to be moved over to the website.

1.2. Group organisation

Modelling & Simulation team Fredrik fixed the scissor lift error and assisted in the integration of front-end, database and simulation, which involved bug fixes and adaptations of the central server. He also wrote large parts of the user guide.

Divy fixed some bugs in the navigation and tested the current code to see if there were any errors left. He did the final evaluation of the system, which includes the creation of the video snippets that can be seen in our video.

Jiaqing has helped writing the user guide and wrote the general part of the demo report. He helped out with editing and design. Reece was the main person behind the user guide. He created illustrations of our robot and wrote main sections for it.

Mike evaluated the costing and mass of the robot breaking it down into different subsystems, using his engineering experience to give estimations for some of the values. The masses were then used to evaluate the mechanical stability of the robot during normal operation.

Software team Ufuk continued to focus on database improvements. He added a page containing packages where users can add and edit existing packages. He also integrated the back-end to the front-end, and extracted the necessary data from the database for the back-end (central server) to use. He added some code to the central server to ensure that the central server started a task, which was immediately added to the database.

Dave completed the market research. He is also responsible for the website that we will show on industry day. He helped with the user guide and produced posters for the demo day. Rohan improved the usability and aesthetics of the warehouse generator and surrounding customer-facing website. He created the website dashboard and helped a little with getting the map generator to work without bugs. Ryan has been working with Rohan and Ufuk to clean up the generator UI and integrate the system with the server so that users can easily use it. Additionally, he made many bug fixes on the back-end.

Arrangements

[1] where you can find our code and simulations: https://github.com/Klausstaler/sdp-21/tree/main

We stuck to the meeting structure described in the previous report. We have checked the codes in the git branch and verified the final version in the main branch, which keeps the code consistent and ensures that users can successfully use our interface after we release the final version of the robot. Each of us looked over the user guide, the website as well as the demo report to make sure there aren't any inconsistencies or misunderstandings. Most of the tasks we do come from the original project plan. After each milestone, the team leader proposed other tasks to make sure everyone can do something. In addition, we also briefly introduced the tasks that have been completed and learned about the time spent by each member and the difficulties they faced. We use this information to redistribute tasks more evenly and to ensure that tasks align with each person's strengths. Everyone accomplished the tasks assigned to them. In the next step, we will go all out to prepare the presentation and website and complete all the work before the industrial day.

1.3. Current budget spent

At this point in time we have not incurred any costs. So far, we have spent 15 minutes of our technician time with Gary to talk about the hardware feasibility of our robot.

2. User Guide

The user guide has been completed, although with our subscription based model it would be expected to be updated regularly and inline with any changes to the system throughout the services life-span.

When working on the user guide we also made sure to have a good mix of people who worked on different sections inform the information that was put into it. This allowed the information to be accurate to the system as it was the people directly involved with that section that were communicated with during its creation.

3. Ethics

Whether or not workers will be replaced with the introduction of robots very much depends on how business leaders who opt to automate their warehouses will ethically approach this issue. The truth is that humans are still better at a large group of tasks, such as packing different objects to fulfill an order.

Ultimately, our goal here at TransportED is to automate the process of picking and stowing of items and bring them to stationary warehouse workers, where they focus solely on getting each order correctly out of the warehouse and into the hands of customers. Furthermore, even the best robots can make mistakes - an item might fall off the robot during transportation. In cases like these, human intervention is still very much needed. Sorting inventory to organize the warehouse and making sure that robot pickers are able to grab them also requires help from humans. We want to unleash the synergies between humans and robots, automating the repetitive and tiring tasks of warehouse fulfilment, leading to higher productivity and satisfaction in the job.

4. Market strategy

The information provided by Graham Allison (Amazon UK Operations' Regional Director) during our meeting in the first report highlighted some key issues of current automation solutions for small and medium-sized warehouses.

- The upfront cost is huge. For warehouses of the size of Amazon, it's in the high eight or low nine figures, with smaller solutions costing anything between $5 million to $25 million. (MMH)
- The payback period of 5 to 10 years is very long, making it difficult to plan for any contingencies. (MMH)
- The time it takes to automate a warehouse can take up to over a year, leaving the warehouse unused in the meantime. If a single warehouse is the source of revenue, you will not be able to automate as you can not cover your fix costs.

To tackle those challenges, we decided to go for rapid deployment, fast return on investment and low capital expenditure. There won't be any upfront investments, instead, we will offer our product as a subscription service, where we take a monthly fee based on the automation benefits of our system. This will allow our customers to profit from automation by day 1 instead of 10 years into the future. As the robots are owned by us, they can simply cancel their contract if they do not need our services anymore and have no liabilities associated with it, allowing for maximum flexibility. As our system works with preexisting shelves and only needs some lines on the floor as well as RFID tags, we minimize the deployment time to a matter of days.

The subscription model will provide us with the needed recurring revenue to supply our customers with steady upgrades to the system. Each of those upgrades will be closely informed by customer feedback, making the customer experience more enjoyable over time. Examples of these improvements could be complex use cases for the robotic arm, such as the management of multiple parcels on one platform or the pickup of more complex objects, improvements to the routing algorithms and speed of the robots.

This should give us a competitive advantage in the automation landscape, enabling many previously sceptic warehouse owners to try automation for themselves and should generate a steady revenue stream for us.

5. Technical details

5.1. Hardware

5.1.1. Robot design

Scissor Lift As described in the previous report, the platform became disengaged from the rest of the robot after the lift has been raised or lowered. After thorough investigation, we found out that this has been caused by missing Physics nodes in the node tree connecting the platform with the robot base. After attaching the required Physics nodes, the platform moved as expected without becoming disengaged.

Choice of wheels: The mecanum wheels have been changed from a custom metal roller design, of mecanum wheel to an off the shelf design (MEC) using polyurethane rollers, capable of handling up to 50kg per wheel. This decision was made after further analysis on the cost of the manufacturing of custom mecanum wheels.

5.2. Software

5.2.1. Server rewrite

While integrating the database code with the central server, we faced issues with the method tasks are sent to the robots. Specifically, we were using asyncio (ASY) for that, which is based on concurrency. To make asyncio work, the I/O has to be non-blocking, meaning that you can still run the program while doing the I/O. Unfortunately, Django (DJA), the framework we use for the website, is using blocking I/O, which blocks the main program until the I/O is finished. To make the central server work with Django, we had to switch from asyncio and non-blocking I/O to blocking I/O using threads. Furthermore, instead of hardcoding the robots in the database, we now write them dynamically into the database whenever they connect with the central server. After connecting, they send their node id, robot id and their size using the established connection. Then, this information is written into the database. Whenever a robot gets new tasks, the scheduler claims the robot by setting a flag for it in the database and writes the task into the database. In case of an outage, this prevents us of losing any information on the executed tasks and claimed robots.

5.2.2. Warehouse Mapper

The warehouse mapper tool has been upgraded to address the issues identified with demo 3's version. The node connections have been made much clearer with the help of arrows and labels which make it more obvious how the connections are laid out. To fix the unappealing look we styled it using the Materialize CSS framework which vastly improved the dropdown and menu options aesthetic. To increase usability, we separated out different stages of the warehouse map construction into individual sections for the user to navigate through, i.e. first the user is presented with the grid setup menu, which is then cleared away to make room for the grid itself.

5.2.3. User Interface

We have converted the home page of the customer-facing website into a dashboard to monitor the state of the warehouse and its robots. The user can now keep track of what tasks are currently being run and the breakdown of how many robots are free/busy directly from the home screen. This means the user does not have to interact with the database or the physical warehouse at all to get a picture of what is happening.

The customer-facing website is split into the dashboard, the package management areas and the warehouse mapper. We made this design decision so functionality is split up logically into different sections that do not overlap too much.

6. Evaluation

6.1. Navigation

To find out the effectiveness of our robot and the automation system, we carry out various tests.

6.1.1. Speed: basic movements and communication

We begin by testing how long our robot takes to complete basic and essential movement. The table 1 was produced by running the given set of tasks 4 times and recording the average time taken to complete them. The time refers to the Webots simulation time. We note that turning on spot takes

N	Task/s to perform	time taken(s)	estimate speed (m/s)
1	T	2.2	0
2	F10	6.1	1.63
3	F5, T, F5	8.4	1.19
4	F5, T, F5, while carrying 7kg parcel	8.5	1.17
5	F5, T, F5, while carrying 7kg parcel + server sending the commands one at a time	8.8	1.13
6	human walk F10	7.14	1.4

Table 1. Time taken to perform certain tasks. Fx = move forward x meters, T= turn 180 degrees.

a significant amount of time. Moreover, the addition of 7kg parcel barely made much difference to the total time. This could also be due to the limitations of the simulation and will be tested once built. Interestingly, performance of our robot is very close to that of a human at 5 km/h walking speed. Hence, by using our system humans could focus on more important and less tedious tasks, like packing.

6.1.2. Speed: Line following and Routing

Next, we test how our system works when integrated with a simple warehouse layout. The warehouse layout used for testing is shown in Figure 1. The distance between each node is roughly 1 meter. The node marked 0, represents the node the robot started on, and displays the time taken to reach every other node. It's interesting that some of the closest nodes take the longest time to reach. However, as this enables the multi agent system, it allows for fast and uncongested movement across the lanes.

6.1.3. Collision detection and Obstacle avoidance

As there will be multiple robots and humans present in the environment, its important for the robot to avoid any collisions. We test this measuring how much the robot travels once it receives the stop command, and how varying speeds affects this. This captured in figure 2. At our top speed, 1.3m/s, the robot travels a distance of 1 meter after the stop command is sent. Hence, we must ensure that in a case where two robots are travelling towards each other at top speed, they are given the stop commands when they are

up lane			down lane			up lane			down lane	
11.6	13.9	14.8	15.6	16.3	16.9	17.5	18.1	18.7	19.6	Right lane
10.7			18			36.6			22.1	
10.1			19			36			23.2	
9.5			0			35.4			24.2	
8.9			0.9			34.8			24.9	
8.2			1.3			34.1			25.6	
5.9	5	4.3	2	33.1	32.5	31.8	30	28.8	26.5	Left lane

Figure 1. Time taken in seconds to reach every node from the starting location. Each gray box represents a shelf. The comment next to each lane represents the direction of allowed movement.

at least 2 meters away. Similarly, we equip our robot with distance sensors capable of capturing the same.

6.2. Robot Stability

As researched in demo 3, the amount of force required for the robot arm to pull a box is half its weight. With our robot having a package limit of 12kg we can conclude that, at least, 60N is required to pull the box. With the robot in its most unstable configuration, scissor lift fully extended and dragging a 12kg package, it is possible. However the package will be moving slowly, due to the low force that can be exerted. For packages closer to the ground much more force can be applied. A breakdown of the calculations can be found in Appendix C .
With this in mind it would be suggested to clients that heavier packages should be placed on the lower shelves, with lighter packages being on higher shelves, where possible.

7. Budget

This is a price estimate of the completed robot: Table 2 shows the estimated total cost of production of a robot as well as the price breakdown of each of the robots sections. The pricing of these sections is also further broke down into its individual components in section A.

Any parts such as mounting brackets will be created using a 3D printer with poly-lactic acid (PLA) filament. This has a much higher impact strength compared to other filaments so would be better in the event of a collision and being

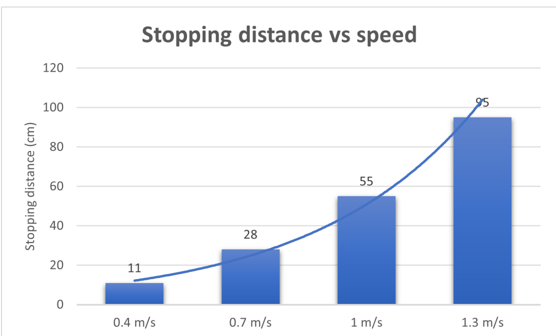

Figure 2. Distance travelled after stop command is sent.

Cost Overview			
Part	Quantity	Price Per	Price Total
Platform	1	£11970	£11970
Chassis	1	£930	£930
Propulsion	4	£344	£1376
Electronics	1	£66	£66
Scissor Lift	1	£270	£270
Labor (per hour)	105	£15	£1575
Total Cost			£16187

Table 2. Price to Manufacture Each Robot (Not Including Equipment and Wiring Required For Manufacturing)

3D printable allows parts to be quickly made. PLA is also well priced with it only being around £44 for 2.3KG (PLA) making it very affordable.

As of right now the arm takes up quite a large part of the budget (£11500). Although it's complexity is not needed in the current iteration of our system with our service model we plan on having future updates being able to take full advantage of the arm's complexity. Allowing multiple packages per robot and shelf stacking capabilities. These kind of future updates justify the large costs here as it will eventually lead to a great increase in productivity and use cases of the robot while also allowing for current users to get these new features in the form of a software update rather than having to wait for the new upgraded robot model to arrive.

8. Video

This is the link to the video.

References

Asyncio python library. URL https://bit.ly/3wj0rtM. Online, accessed 28 March 2021.

Django framework. URL https://bit.ly/3fo9R12. Online, accessed 28 March 2021.

8 inch nexus mecanum wheels. URL https://bit.ly/2PAXl3s. Online, accessed 20 March 2021.

Main obstacles of warehouse automation. URL https://www.mmh.com/article/15_myths_about_warehouse_automation_debunked. Online, accessed 28 March 2021.

Pla fillament 2.3kg pricing and specifications. URL https://bit.ly/3ryxINZ. Online, accessed 28 March 2021.

A. Robot Costing Breakdown

Cost Overview			
Part	Quantity	Price Per	Price Total
Platform	1	£11970	£11970
Chassis	1	£930	£930
Propulsion	4	£344	£1376
Electronics	1	£66	£66
Scissor Lift	1	£270	£270
Labor (per hour)	105	£15	£1575
Total Cost			£16187

Platform			
Part	Quantity	Price Per	Price Total
Shelf	1	£40	£40
Youbot arm	1	£11500	£11500
Arm attachment	1	£150	£150
Linear actuator	1	£245	£245
Vacuum pump	1	£35	£35
Total Cost			£11970

Linear Actuator			
Part	Quantity	Price Per	Price Total
Supports	2	£10	£20
Driving thread	1	£15	£15
Arm shelf	1	£100	£100
Motor	1	£50	£50
Bearing	6	£10	£60
Total Cost			£245

Chassis			
Part	Quantity	Price Per	Price Total
Frame	1	£500	£500
Bodywork	1	£100	£100
Battery	1	£230	£230
Battery case	1	£100	£100
Total Cost			£930

Propulsion			
Part	Quantity	Price Per	Price Total
Motor	1	£80	£80
Motor Gearing	1	£20	£20
Motor mount	1	£30	£30
Motor controller	1	£20	£20
Encoder	1	£20	£20
Mecanum wheel	1	£174	£174
Total Cost			£344

Electronics			
Part	Quantity	Price Per	Price Total
Robot controller	1	£40	£40
Radar module	1	£10	£10
IR sensor	3	£2	£6
RFID reader	1	£10	£10
Total Cost			£66

Scissor Lift			
Part	Quantity	Price Per	Price Total
Lift members	1	£150	£150
Motor	1	£80	£80
Motor controller	1	£20	£20
Encoder	1	£20	£20
Total Cost			£270

Table 3. Detailed costing to manufacture the robot

B. Robot Mass

Cost Overview			
Part	Quantity	Mass Per (kg)	Mass Total (kg)
Platform	1	21	21
Chassis	1	21	21
Propulsion	4	5	20
Electronics	1	3	3
Scissor Lift	1	9	9
Package (max)	1	12	12
Robot			74 kg
Robot & Package			86 kg

Platform			
Part	Quantity	Mass Per (kg)	Mass Total (kg)
Shelf	1	7	7
Youbot arm	1	6	6
Arm attachment	1	1	1
Linear actuator	1	6	6
Vacuum Pump	1	1	1
Total Mass			21 kg

Linear Actuator			
Part	Quantity	Mass Per (kg)	Mass Total (kg)
Supports	2	0.6	0.6
Driving thread	1	0.8	0.8
Arm shelf	1	2	2
Motor	1	1	1
Bearing	6	0.17	1
Total Mass			6 kg

Chassis			
Part	Quantity	Mass Per (kg)	Mass Total (kg)
Frame	1	6	6
Bodywork	1	1	1
Battery	1	6	12
Battery case	1	2	2
Total Mass			21 kg

Propulsion			
Part	Quantity	Mass Per (kg)	Mass Total (kg)
Motor	1	2	2
Motor Gearing	1	0.5	0.5
Motor mount	1	0.5	0.5
Motor controller	1	0.5	0.5
Encoder	1	0.5	0.5
Mecanum wheel	1	1	1
Total Mass			5 kg

Electronics			
Part	Quantity	Mass Per (kg)	Mass Total (kg)
Robot controller	1	0.5	0.5
Radar module	1	0.5	0.5
IR sensor	3	0.5	1.5
RFID reader	1	1	0.5
Total Mass			3 kg

Scissor Lift			
Part	Quantity	Mass Per (kg)	Mass Total (kg)
Lift members	1	6	6
Motor	1	2	2
Motor controller	1	0.5	0.5
Encoder	1	0.5	0.5
Total Mass			9 kg

Table 4. Detailed mass of the robot

C. Robot Stability

Using the masses, from Appendix B, a mass model of the robot can be created. The 2 cases we care about are the best case and the worst case, see figure 3 and figure 4 respectively. Figure 3 gives the model for when the robot is in its safest state. Scissor lift contracted and no package on the platform. Where as figure 4 shows the robot in its most unstable state. Scissor lift fully extended and just attached onto the desired package.

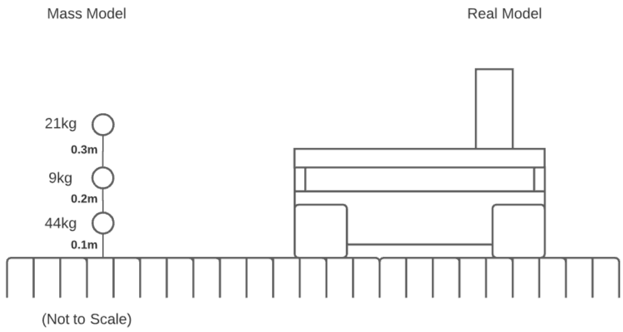

Figure 3. Mass model for stability calculations with robot in its best case

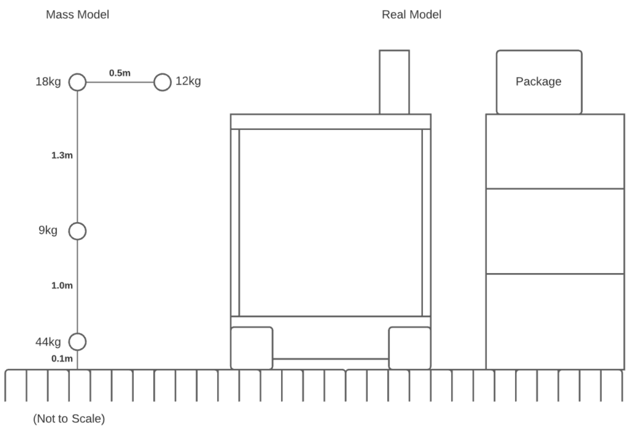

Figure 4. Mass model for stability calculations, with robot in its worst case

The center of mass (CoM) of each model can be created, with the horizontal distance being CoMx and the vertical distance being CoMy, with respect to the center of the wheelbase and the ground respectively, using equation 1.

$$CoMx = \frac{\sum m_i x_i}{\sum m_i} \tag{1}$$

After finding the CoM for the X and Y axis of the model the angle at which the robot would start to fall over can be found, relative to the ground, can be found using equation 2.

$$\alpha = \tan^{-1}\left(\frac{\frac{wb}{2} - CoMx}{CoMy}\right) \tag{2}$$

Where **wb** is the wheelbase

This angle is then used in equation 3 to give the amount of force required, in the horizontal axis, to make the robot unstable.

$$Fh = mg \times sin(\alpha) \times \frac{CoMy}{Y} \tag{3}$$

Where **Y** is the distance between the floor and the shelf

The table below shows the results, using the equations explained above, for the state of the robot during its best and worst case loading conditions.

Position	Low w/o package	High with package
Mass (kg)	74	86
CoMx (m)	0.00	0.07
CoMy (m)	0.27	1.00
α (degree)	52	15
Fh max (N)	257	91

Table 5. Robot Stability calculation results