TEST METHOD

REQUIREMENTS

To validate the system's performance, a robust test method was required. The test method needed to identify the robot, track position and rotation throughout the test, output positional data and a graphical visualisation and output an 'efficiency rating' based on the directness of the route taken (Figure 32). Research was undertaken to find an appropriate test method which met our requirements, however, the test methods found were either: too simple and would not deliver enough information to understand the impact of each component fully (such measuring the start/finish points by hand); or too expensive to implement due to requiring expensive cameras (£3000-£20000) for infrared feature tracking, such as those used for gait analysis [20]. Therefore a bespoke test protocol was designed.

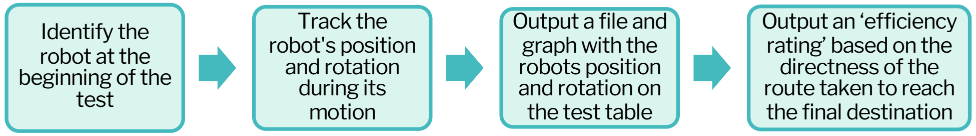


Figure 32: Test method requirement flowchart.

HARDWARE DESIGN

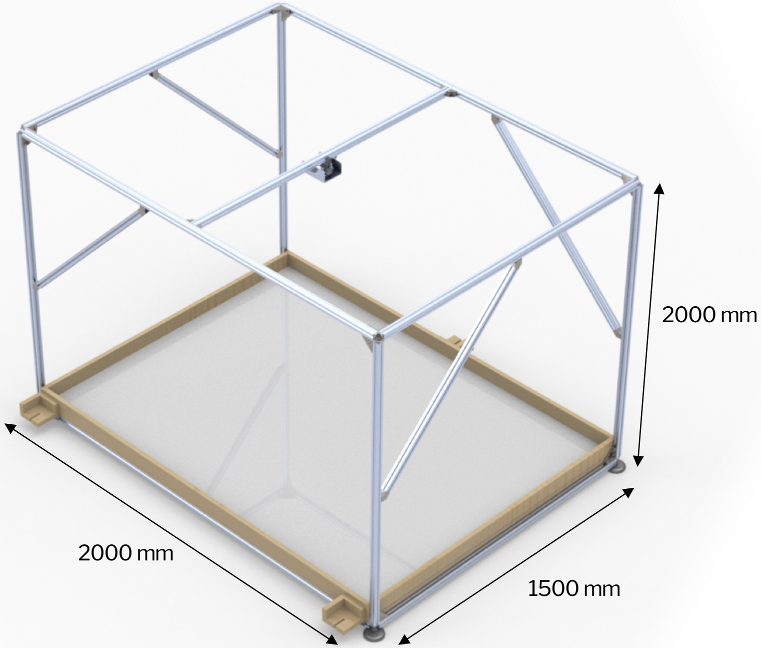
The simplest method to measure the robot accuracy is to use a pen mounted on the robot to mark the start and finish positions. With these positions marked, the error between the robot's actual final position and target final position can be measured by hand.

This technique is laborious and prone to human error and uncertainty caused by manufacturing tolerances. Additionally, information on what may have caused the errors cannot be determined with this basic method.

A test method which offered a higher level of information, such as the path the robot took to reach its final position, was desirable. Using a camera as the main test data recording tool offered the most flexibility for developing a testing protocol suitable for a wide range of applications. Versatility was a key attribute to consider to ensure that the test method was cost effective and offered the best possible value.

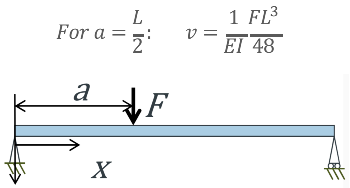
Frame based set-ups for mounting the camera overhead were considered. The frame needed to be robust to ensure repeatable results across different test sessions which may be months apart. A simple four legged frame with a camera mount in the centre of the top frame was chosen to offer the robustness required, while also giving flexibility for later changes. With versatility being a key requirement, it was decided that the test rig frame would be made from modular aluminium extrusion. Using aluminium extrusion allows for quick adjustment during setup and use, while providing the robustness required for repeatable results. The base of the frame features 3 cross struts to ensure the table is flat without risk of sagging. The frame also features four adjustable feet for levelling.

FINAL DESIGN



DEFLECTIONS

The aluminium extrusion profile size options were 20 x 20 mm, 30 x 30 mm or 40 x 40 mm. Basic beam bending theory has been applied to give the following deflection estimations for the 2 m beams, which are the longest in the structure, as well as 1 m beams for comparison. The structure will not have any additional loads applied besides the weight of the structure itself and the camera, which is negligible. The structure must be able to resist loads experienced if 'knocked' by users. The deflection estimates were calculated based on a 100 N reasonable worst-case knocking force being applied at the centre of the beams.



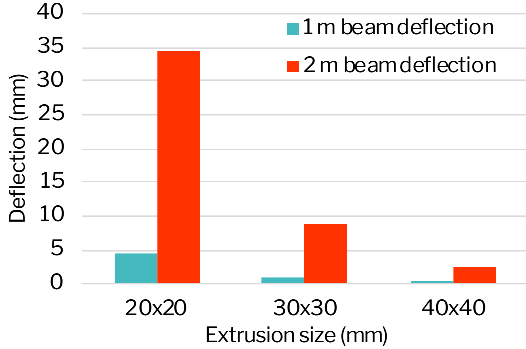


Figure 33: Deflection of aluminium extrusion

The deflection estimates (Figure 33) for the 20 x 20 mm aluminium extrusion were too high and this size was discounted. Deflection estimates of less than 10 mm for the other extrusion sizes for a 100 N force are a positive indication that the structure should be stable. 100 N is equivalent to adding a 10 kg mass to the bar. Based on these calculations and guidance received from faculty members, 30 x 30 mm aluminium extrusion was chosen. Cross bracing was also added at four locations to improve structural rigidity.

The area on which the robot operates is 2000 x 1500 mm, representing a half-sized Eurobot table. The base is water-resistant MDF to minimise the risk of warping, while the side walls are made from 22 mm thick MDF, matching the height and thickness of a Eurobot table. Three beacon mounts were also manufactured to Eurobot specifications and are mounted on the aluminium extrusion to allow for easy adjustability or removal. The table surface is a single piece of 3 mm Forex, which is removable, allowing for different patterns to be printed and used without causing damage to the test table, increasing versatility.

CAMERA SPECIFICATION

A Logitech C930E webcam was chosen due to having the largest field of view (claimed 90˚) of commercially available webcams. This allowed for the test rig frame height to be minimised, whilst including the full table length and width. Using a commercially available webcam was preferred as ensuring the drivers are available increases the usability for other users of the test method besides those involved in this project.

In practice, the Logitech C930E did not have a field of view of 90˚. The image frame was restricted to a 16:9 widescreen perspective, so if the camera had a 90˚ horizontal field of view, the vertical field of view would be 50.6˚.

$$FOV_v = 90 \text{ x } \frac{9}{16} = 50.6^{\circ}$$

The 2000 x 1500 mm test table has an aspect ratio of 4:3, so the vertical field of view is more important. As long as the requirements for the vertical field of view are satisfied, the horizontal view will also be satisfied. The height the camera must be mounted at is:

$$h = \frac{(t_h/2)}{(\tan\frac{FOV_v}{2})}$$

where:
h = camera mount height
FOV_v = camera vertical field of view
t_h = table height (1.5 m for this test rig)

1.59 m would be the correct mounting height if the field of view was within specification. However, during testing the camera needed to be mounted at a height of 1.8 above the table to capture the full height of the table. Therefore, the formula can be rearranged to find the cameras actual field of view.

$$h = \frac{(t_h/2)}{(\tan\frac{FOV_v}{2})} = \frac{(1.5/2)}{(\tan\frac{50.6}{2})} = 1.59 \text{ m}$$

$$FOV_v = 2\tan^{-1}\frac{(\frac{t_h}{2})}{h} = 2\tan^{-1}\frac{(\frac{1.5}{2})}{1.80} = 45^{\circ} \text{ (vertical FOV)}$$

$$FOV_h = 2\tan^{-1}\frac{(\frac{t_w}{2})}{h} = 2\tan^{-1}\frac{(\frac{2}{2})}{1.30} = 75^{\circ} \text{ (horizontal FOV)}$$

Using an additional wide angle lens was tested in an attempt to increase the field of view. While the lens was successful at increasing the field of view, the image clarity was poor and the images were too out of focus to use for testing, so additional lenses were discounted. Due to the camera field of view being lower than expected, the test table frame had to be taller than initially planned (2 m height rather than 1.6 m), causing the structure to be less stable than planned causing vibration when knocked by users. Additional bracing was added to increase the stability and the vibration of the frame is damped more quickly after being knocked as a result.

The Logitech C930E is a 1080p high definition camera, with each pixel representing 1.38 mm on our test table. Using a 4K camera would increase the resolution to 0.69 mm per pixel and could be considered for future work.

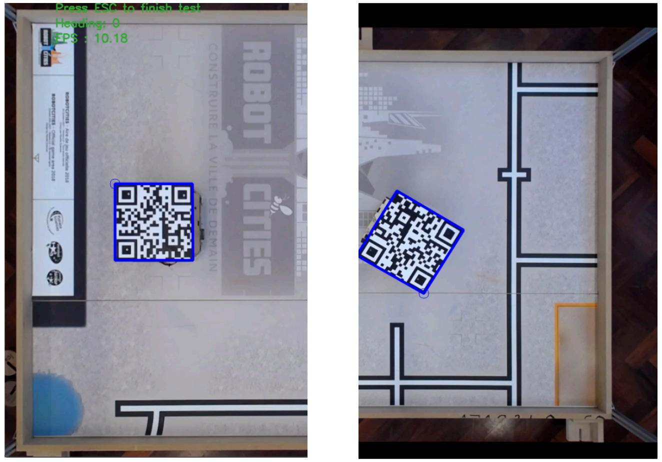


Figure 34: The difference in edge limits for a 19.7 cm high robot (left) and a 43 cm high robot (right)

SOFTWARE DESIGN

To meet the testing requirements for this project, the software needed to perform a number of tasks (Table 10).

Requirement
Lens distortion correction
Identify the robot at the beginning of the recording based on an identifiable pattern mounted on the robot
Track the robot's position and rotation during its motion
Output a file and graph with the robot's position and rotation on the test table
Output an 'efficiency rating' based on the directness of the route taken to reach the final destination

Table 10: Summary of test method requirements

CAMERA CALIBRATION

When using computer vision as a measurement tool, it is important to measure the characteristics of the camera used, the intrinsic properties. The pinhole camera model is a mathematical model commonly used in computer vision applications. The camera is modelled as having an aperture described by just a point and using no lenses to focus the light. This model is used to remodel 2D coordinates from a photograph/video to 3D coordinates for the object world that the image represents using perspective transformation.

To estimate the intrinsic parameters, a known pattern must be used. In this project a dot pattern was used, where the distance between the centres of each dot and the number of dots is known. A series of calibration images were taken where the dot pattern is spread evenly around the image frame, making sure to capture images at the edges of the frame where the distortion is at its worst. The intrinsic camera properties were then used to undistort the video, removing any aspect of camera distortion.

TABLE EDGE LIMIT CALIBRATION

The test method is versatile and can be used for different height robots. However, robots operating at different heights are effectively operating on different planes, which must be accounted for before tests begin to ensure accurate determination of the table edge limits.

The table corner points are determined by placing the robot to be tested in each corner of the table. The corner point positions are recorded and input into the test method code to set the table limits for position calculations. The video is cropped based on the corner points to reduce video frame size and increase the robot tracking speed. Figure 34 highlights the difference between video frame size for a 19.7 cm high robot and a 43 cm high robot.

ALGORITHM

In order to track the robot, its position must be identified from the video stream. Using a clearly identifiable pattern mounted centrally on top of the robot was determined to be the most appropriate method for a computer vision system. Patterns using colour are commonly used in similar applications but it was decided that there was a high risk of error due to varying lighting conditions and patterned table surfaces. Instead, a QR code was chosen. A QR code is well suited to this application because it has enough features to be unique (making it less susceptible to errors caused by other patterns in the video) and a black and white colour scheme means it is less susceptible to variance caused by lighting conditions.

To identify the QR code in the video, a detection algorithm was required. The algorithms tested were: scale-invariant feature transform (SIFT); speeded up robust features (SURF); and Oriented FAST and Rotated BRIEF (ORB). SIFT was published in 1999 [14] and has long been the industry standard for feature detection in images. For SIFT to detect features a sample image must be supplied, the QR code in this case. The frames of the video are then compared to the reference image and potentially matching features are identified based on the Euclidean distance of their feature vectors. The full set of potential matches are then filtered based on object, location, scale and orientation to identify good matches. The image features are generally found on high contrast regions of the images, such as the object edges, making the black and white QR code ideally suited for feature matching.

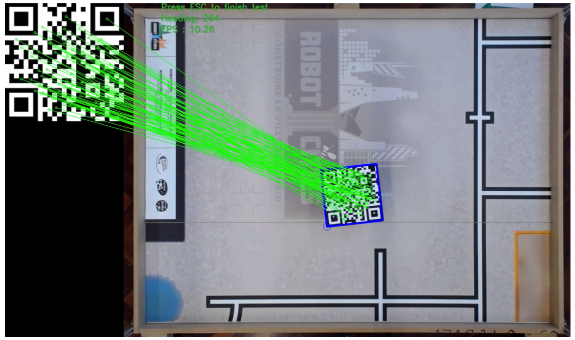


Figure 35: SURF detection in progress. The green lines indicate the features being matched between the sample image (left) and the live video feed (right)

Another method of recognising features is using SURF [15] which is also a feature detector and descriptor. SURF is faster and more robust against image transformations; however, it can identify fewer features than SIFT. El-Gayar et al. ran a comparative study of low level feature extraction algorithms, including SIFT and SURF, in which the performance was assessed when an image was subjected to variations in scale, occlusion, orientation, illumination and blurring. The results show that SURF is consistently faster than SIFT, however SIFT finds more descriptors [16]. ORB was also tested for comparison. ORB was developed by Rublee et al. [17] with an implementation in OpenCV, the open source computer vision package used in this project. The algorithm combines FAST [18] and BRIEF [19] with the primary focus being speed of processing.

All three algorithms were tested using the same Logitech 930E webcam to detect a stationary QR code on the test table which was in the same position for each test. Given the QR code is stationary, the standard deviation of the (x, y) position and the heading (rotation of the QR code) must be zero for a perfectly consistent detection algorithm. The results of this test are summarised in Table 11. SIFT has the lowest standard deviation (σ_x= 0.044 mm; σ_y = 0.044 mm; $\sigma_{heading}$ = 0.038°) of all the detection algorithms, which is evident during testing because for a stationary QR code there is little fluctuation in the box which marks the QR code position and edges. However, at 1.65 frames per second (fps), the processing speed is too slow for our application and the risk of missing parts of the robot's motion is high. ORB offers considerable processing speed improvements achieving 29.97 fps, meaning very few frames are not processed while using the 30 fps Logitech 930E. However, the position and heading standard deviation is significantly higher (σ_x = 6.518 mm; σ_y = 6.672 mm; $\sigma_{heading}$ = 1.827°) than SIFT. SURF offers a suitable compromise between SIFT's accuracy and ORB's processing speed, with low standard deviation (σ_x = 0.095 mm; σ_y = 0.151 mm; $\sigma_{heading}$ = 0.131°) and a reasonable processing speed of 8.181 fps. Therefore, SURF was selected for the feature detection algorithm (Figure 35).

Algorithm	Mean x [mm]	Mean y [mm]	Mean heading [°]	x standard deviation [mm]	y standard deviation [mm]	Heading standard deviation [°]	Mean frames per second processed
SIFT	289.11	1230.72	267.20	0.044	0.044	0.038	1.65
SURF	288.81	1231.21	267.35	0.095	0.151	0.131	8.18
ORB	299.16	1242.66	270.11	6.518	6.672	1.827	29.97

Table 11: The difference in key performance indicators for SIFT, SURF and ORB detection algorithms

SURF: HESSIAN REFINEMENT

SURF determines points of interest by using the determinant of the Hessian matrix as a measure of the local change around points [15]. Points with high differences in contrast such as edges are generally preferred by SURF and therefore the points with the largest Hessian matrix determinant are chosen as key points. In the OpenCV implementation of SURF, it is possible to adjust the Hessian matrix settings to refine the detection results. In summary, the higher the Hessian matrix settings, the lower the number of key points retained for comparison between the sample image and the video frame. In theory, the lower the Hessian matrix threshold, the slower and more accurate the detection will be. To find the optimal Hessian threshold to meet the required speed and accuracy for this application, tests were carried out to detect the same stationary QR code using the same webcam and program, changing only the Hessian threshold between each test. The mean and standard deviation for the (x,y) position and heading were calculated for each Hessian value ($100 \le H \le 16000$), while also recording the mean processing speed, Figure 36. When $H \ge 16000$, the results were too erratic and unusable. The Hessian threshold which offered the best compromise between accuracy and processing speed was H=2700. At H=2700, σ_x = 0.083 mm; σ_y = 0.110 mm; $\sigma_{heading}$ = 0.102°; while maintaining a mean processing speed of 7.815 fps.

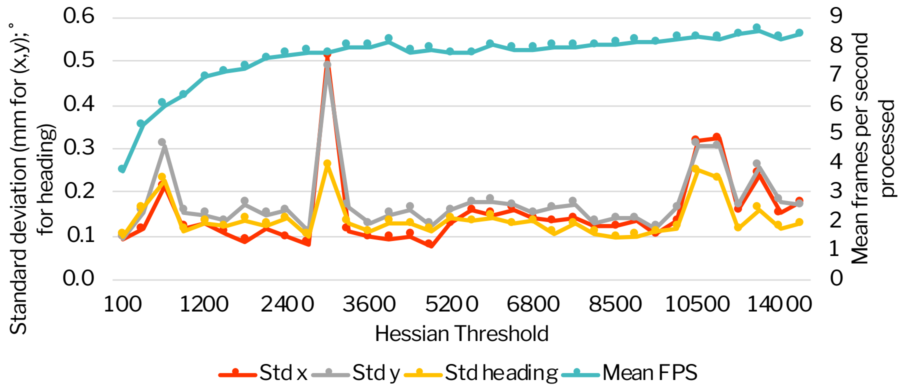


Figure 36: Variation in key performance measures from changing the Hessian threshold

TEST METHOD VALIDATION

PROCESS

Before the test method can be used to assess the robot's positional accuracy, it must be validated to assign measurement tolerances. To validate the test method the following process is completed, Figure 37. First, the test is calibrated, by ensuring the camera is secure, calibrating the camera and determining the table edge limits. Using a 400 x 600 mm laser cut MDF board with holes set in a grid format in 50 mm increments, randomly chosen points are marked on the table surface, using a whiteboard marker. The MDF board is accurately cut by laser and aligned against the table walls to ensure the points are accurately marked. The positions of these marked points are now known. The QR code used for detection is aligned with each marked point using a hole in the middle of the pattern. Now, the QR code is at the known position. The test method is used to determine the position of the QR code and the test method positional results are compared to the known positions. The test method uncertainty is then determined.

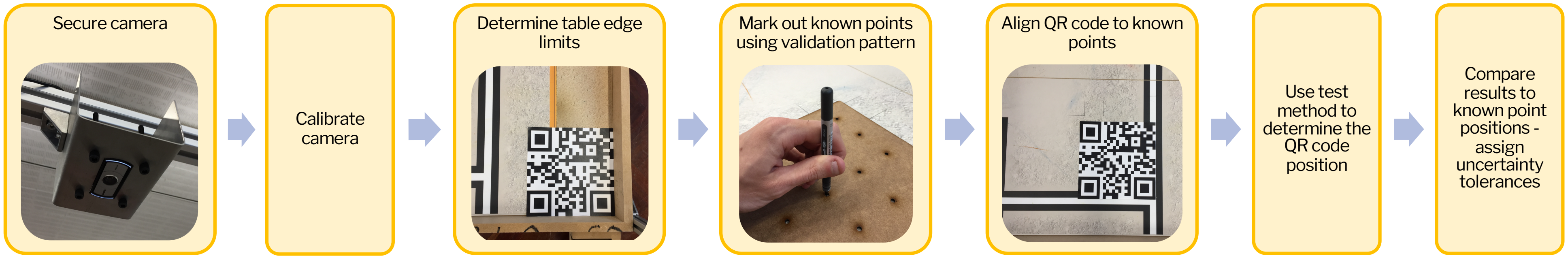


Figure 37: The process completed the validate the test method.

VALIDATION RESULTS

The test method validation results are displayed in Table 13 for 20 positions on the test table, marked using the laser cut validation pattern. The error is the difference between the known position on the table and the position measured by the test method. The results are summarised in Table 12, indicating a mean error of -0.92 mm in the x-direction and 2.03 mm in the y-direction, and a standard deviation of 0.77 mm and 0.79 mm in the x and y directions, respectively. Therefore, the 2σ confidence bounds for the x-direction are [-0.53, 2.42] mm and [0.46, 3.61] mm in the y-direction. The errors in the x-direction are almost exclusively negative, while the errors in the y-direction are almost exclusively positive. Errors in the x-direction are largest on the right of the table (points 3-5 and 8-10 on Figure 38), which may highlight an issue with the camera calibration and image undistortion or the table edge limit definition.

Potentially the uncertainty could be reduced by recalibrating the camera or defining different table edge limits but the best next step to reduce the uncertainty is to use a 4K camera to increase the video resolution.

	x	y
Mean error [mm]	-0.92	2.03
Standard deviation [mm]	0.77	0.79
2σ Lower confidence interval [mm]	-2.46	0.46
2σ Upper confidence interval [mm]	0.62	3.61
Largest absolute error [mm]	-2.64	2.82

Table 12: Test method validation results summary.

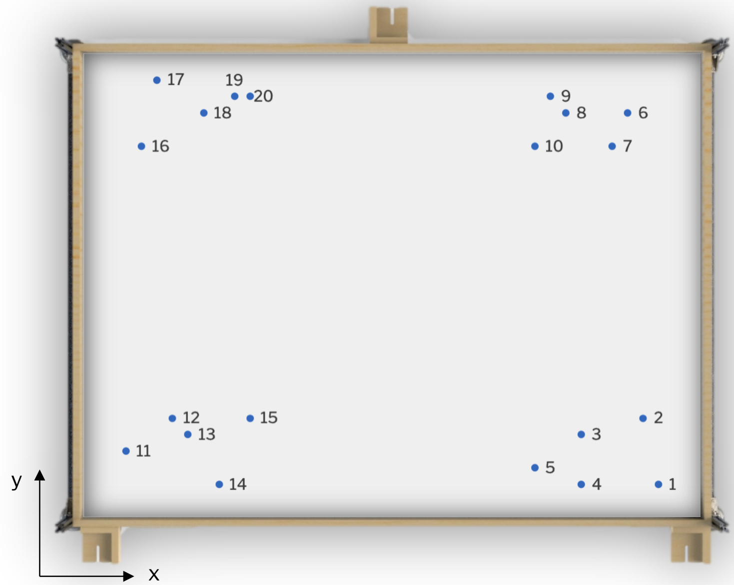


Figure 38: The numbered points are the known positions on the test table which were tested to determine the positional in the test method

Point number	x error [mm]	y error [mm]
1	0.17	-0.17
2	-0.81	0.21
3	-1.63	2.06
4	-1.86	2.80
5	-2.45	2.45
6	-0.52	1.27
7	-1.04	1.76
8	-1.37	2.04
9	-1.43	2.81
10	-2.64	2.10
11	-0.92	1.98
12	-0.40	2.32
13	-0.81	2.52
14	-0.61	2.71
15	-0.35	2.06
16	-0.61	2.40
17	-0.18	2.19
18	0.08	2.82
19	-0.43	2.24
20	-0.59	2.14

Table 13: The (x,y) errors at each point

RESULTS

BEACON PERFORMANCE DROP

Despite promising beacon testing results, once integrated into the robot, the performance dropped dramatically. Not only did the sampling rate decrease, but the latency of the results increased beyond a usable limit. External factors such as ultrasonic echoing, radio frequency and electronic interference were considered as possible culprits; CPU load and Raspberry Pi voltage levels were also considered.

RADIO WAVE INTERFERENCE

The beacon was designed to sit above the top plate of the robot chassis (as seen in Figure 39). However, the frequency of readings was much lower than expected when compared to initial testing completed on the prototype chassis. Testing the beacon performance 1 m away from the robot chassis not only improved the sampling rate, but also the latency of the readings, suggesting the robot design was causing interference. Further investigation highlighted that the radio frequencies between the beacons and modem were being blocked by the aluminium top plate. By replacing the middle and top plates of the robot with acrylic, the beacon was then far enough away from aluminium to not cause interference. This only slightly improved sampling rate, suggesting there was another factor affecting performance.

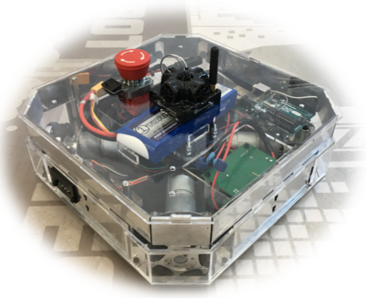


Figure 39: Robot with acrylic top and mid plates.

LATENCY & SAMPLING RATE

Signal latency and low sampling update rates still remained, evidenced in Figure 40. The path taken by the robot (data recorded by the test table marked in blue) has a duration of 5.99 s. The orange data points show the frequency and timing of the beacon readings when connected to the robot. Here, the first beacon reading was received by the system at 6.35 seconds, 0.36 s after the robot had finished its path. Additionally, only four position updates were recorded by the beacons, which is too few to be used in the Kalman filter, requiring multiple readings per second.

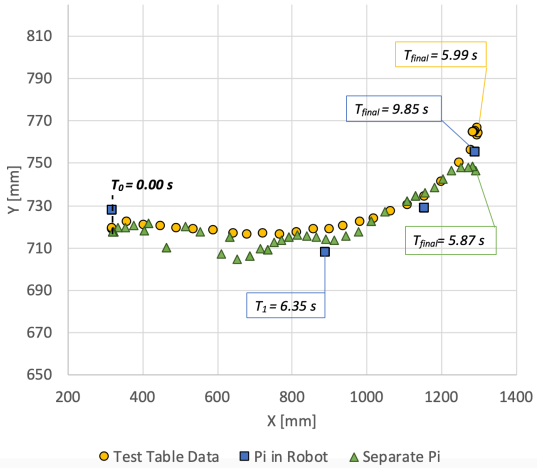


Figure 40: Change in robot position as it drives from one position to another. Beacon readings are taken along this motion for two instances.

COMPUTATIONAL LOAD

It was suspected that high computational load could be the cause of the infrequent readings. Beacon results were also collected on a separate Raspberry Pi to determine if this was the case. As is illustrated in Figure 40, there are significantly more data points (green markers) when using a separate Raspberry Pi. The final beacon reading was recorded 5.87 s after the robot started moving, which closely corresponds to the time at which the robot stopped moving, at 5.99 s.

These results showed that, the current robot is not capable of processing the beacon data fast enough to be useful in this system, but the beacons would be an appropriate sensor if connected to a system that could manage the computational load.

LOCALISATION PERFORMANCE OVER TIME

TESTING ROUTE

A square shaped path was chosen as a route for the robot to traverse. Inspired by the UMBMark, a benchmark test used for odometric systems [21], a square path is ideal for evaluating the error in a system as the path has the same start and end point. Therefore, the error incurred in the system is easily visible. To test system performance over time, the diamond path (Figure 41) of side length 500 mm was traced 5 times. This test was repeated using a Part II student robot. Our robot was also tested using the same method, once with direct optical sensor feedback and once with a Kalman filter state estimation. The duration of this path varied with each setup, but took in the region of 100 seconds for all setups. As the robot reaches each of the four corners, the error in the robot's position (in x, y and angular directions), the robot's estimation of position is also recorded to evaluate how well the robot is perceiving its position.

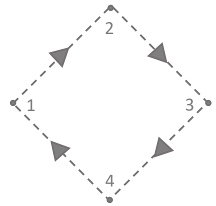


Figure 41: Path traced to determine error in system.

ODOMETRIC SYSTEM

A second year project Eurobot entry was tested to provide a baseline to which the designed systems in the this project can be compared to. The robot is a standard two wheel differential drive that reads the encoders to determine the distance travelled. The robot has no perception of its real position in these systems so only the error in position compared to the desired position.

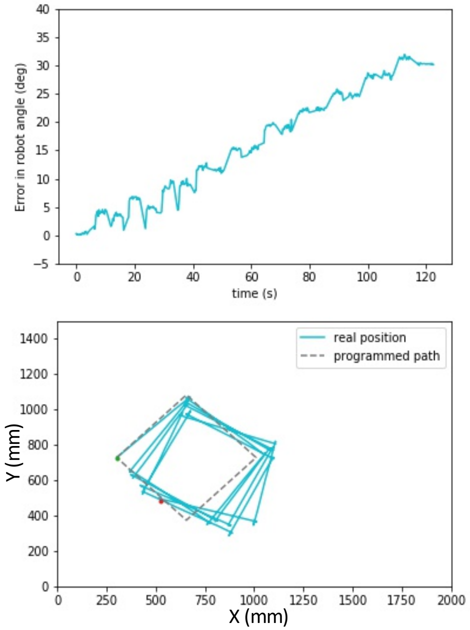


Figure 42: Typical Eurobot entries localising with odometry using encoders to give feedback of position

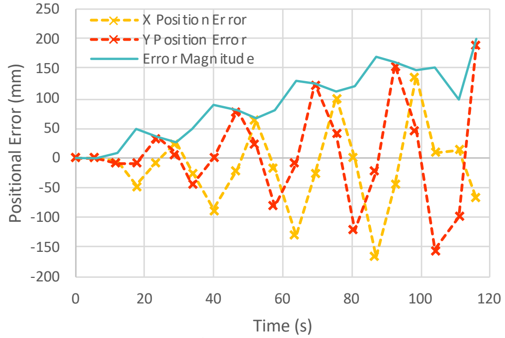


Figure 43: Error between intended position and real position.

The baseline system incurred a position error of 200 mm (Figure 43) after completing five loops of the path. The total angle error built up over the period totalled 30° (Figure 42). The odometric system has an interesting quality that when the error is large in the x direction, it is minimal in the y direction, and the direction of the error oscillates throughout the path. This suggests that the angle error generated by each turn has the largest effect on overall position error.

DIRECT MEASUREMENTS FEEDBACK

The robot using optical measurements as direct feedback shows a lower overall position error of 25 mm after completing the path (Figure 44). As only the optical sensor, which is sensitive to drift, is used in this system the error still builds over time. However, the rate at which the error builds is approximately 8 times slower than in the baseline odometric system, which is a large improvement. The error in the robot's perception of its position shows a correlation with the absolute error, further suggesting that the sensor's drift is the cause of the error increase over time. The overall error in position estimation at the end of the path is 20 mm (Figure 45). Improved performance over the odometric system can be attributed to two reasons. First, the optical sensor measures the overall position of the chassis, compared to the encoders which only measure the rotation of the wheels. Measuring the overall chassis position successfully accounts for slip between the wheels and the surface. Additionally, the robot stores the last position estimate, and uses that to calculate the distances required to the next point, which slows down the rate at which error compounds.

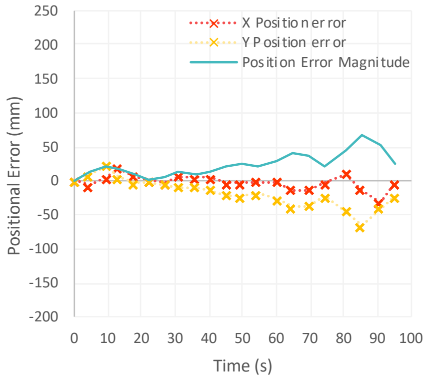


Figure 44: Error between intended position and real position.

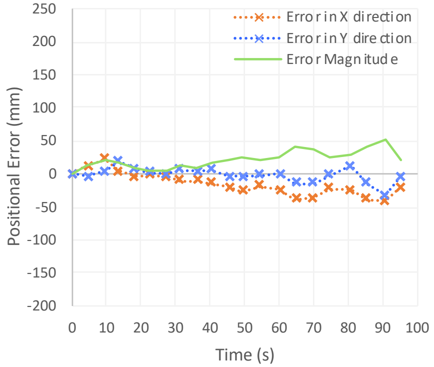


Figure 45: Error between estimated position and real position.

The error in angular position totals 2.9° (Figure 47) after a completion of the path. This is approximately 10 times lower than the error incurred in the baseline odometric system. Because no angle error correction was installed, this improvement can be entirely attributed to the omnidirectional drivetrain, which allows movement in all directions without requiring the robot to make additional movements to turn.

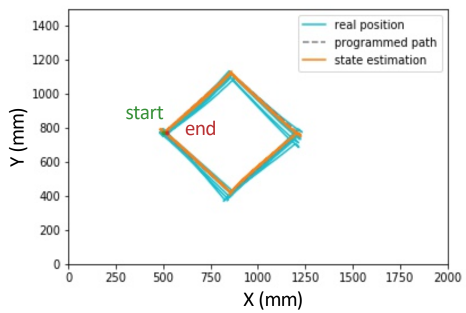


Figure 46: Test route position.

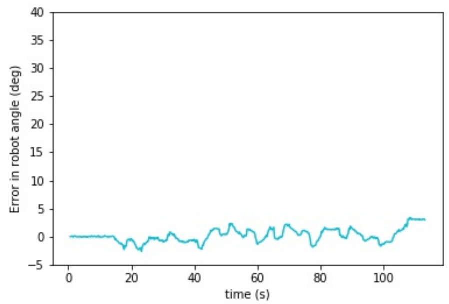


Figure 47: Error in robot angle over test route.

KALMAN FILTERING FEEDBACK

The robot performance appears to be worse with the Kalman filter using optical sensor readings. The overall error build up after the path was completed was 81 mm (Figure 48). The error in the estimation of position was also worse than using optical sensor readings directly, giving a total error of 88 mm (Figure 49). However, both showed an improvement compared to the baseline odometric system.

On further investigation of the filter implementation, there are two possible reasons for the reduced performance: incorrect modelling of the sensor noise or a mismatch in the timestep employed in the state model compared to the real frequency of the sensor readings. The error in the position estimate shows the same shape as the overall error, suggesting that the values modelling the noise in the measurements have too low of a value. Addition of the filter slowed a drive loop down to take 0.61 s. This was not accounted for in the state model of the filter, which may be the cause of the overshoot in the paths (Figure 50).

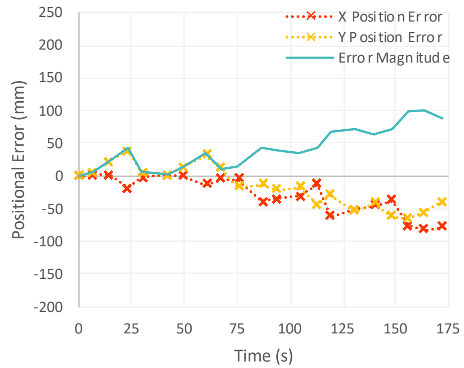


Figure 48: Error between intended position and real position.

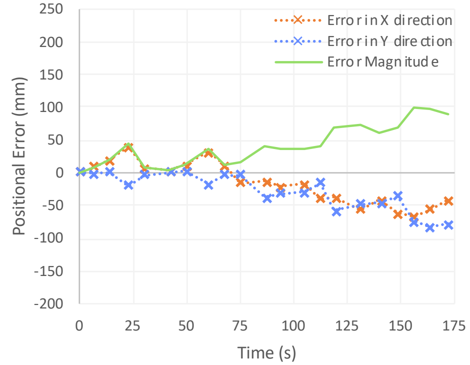


Figure 49: Error between estimated position and real position.

Error in angular position shows the same behaviour as the system using direct optical measurements. There is no difference in angle control between these two systems. On observation of the overall path (Figure 50), the drift in the system when using Kalman filtering is worse than using direct measurements, which may suggest further tuning of the filter is required. Due to time constraints in the project, further investigation into the filter was not possible, but the technique shows promise with further development.

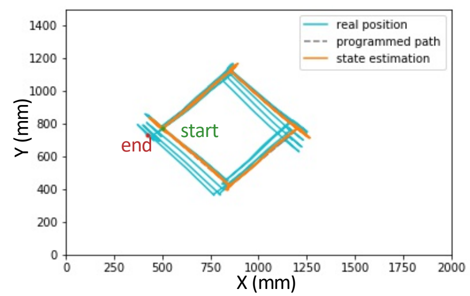


Figure 50: Test route position.

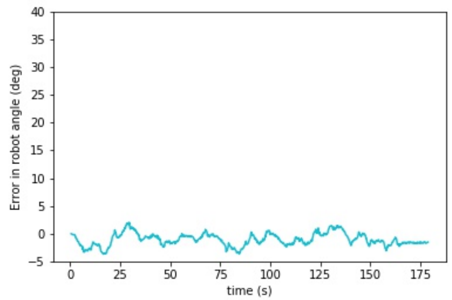


Figure 51: Error in robot angle over test route.

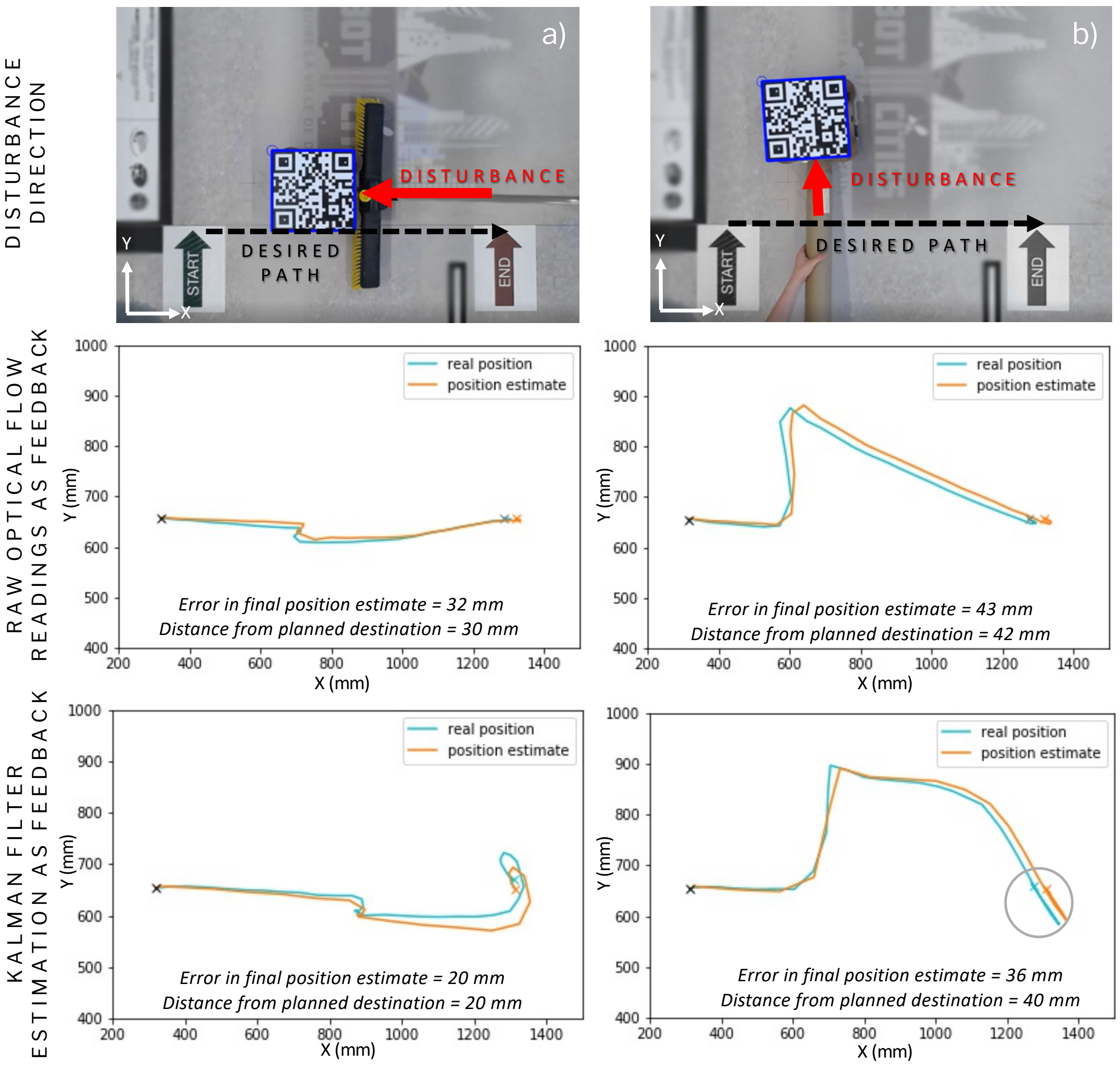


Figure 52: Top: The robot's programmed path and the applied disturbance. Middle: The response of the robot using raw optical feedback. Bottom: The response of the robot using a Kalman Filter to merge optical measurements

EFFECT OF THE PID CONTROLLER

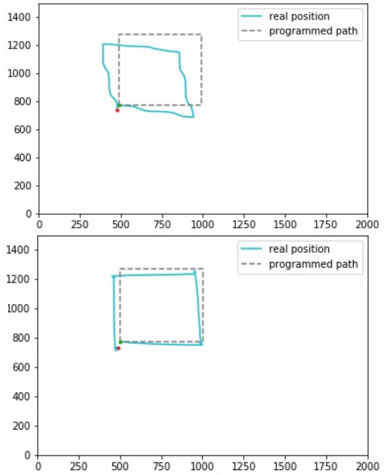


Figure 53: Top: path of motion without PID control. Bottom: Path of motion with PID controller using raw optical measurements as feedback

Only controlling the speed of each motor is insufficient for providing straight, neat paths with an omni directional drivetrain meaning the real path taken diverges from the programmed path (Figure 53). The nonlinear path taken causes systematic error in the system, making it difficult to design a state model for the Kalman filter. The error can be assumed to be systematic as the overall drift of the robot cancels out when as the square path is taken. To attain more predictable motion, PID control was applied to the overall position of the robot so that its position could be corrected as the robot moves. It is evident that the robot follows the designed path more closely.

EFFECT OF DISTURBANCES

A single direction horizontal path of 1000 mm long was programmed into the robot. Two types of disturbances were applied to the robot: a force to oppose the robot's direction of motion (Figure 52a) and a lateral push perpendicular to the direction of motion to offset the robot from its course by 300 mm. (Figure 52b). Each experiment was completed once using only raw measurements for the robot's position feedback to the PID controller and once with the Kalman filter applied to provide feedback to the PID.

Overall, the PID controller performed well, with large disturbances only incurring an error less than 50 mm in all experiments. This falls short of the project aim which specifies accuracy below 10 mm. However, it is a large improvement compared to a simple odometric system that reads encoder values, as such a system would not be able to detect and overcome the above disturbances at all.

Using only raw optical measurements for feedback (middle row, Figure 52) generated a fast response to the disturbance. The robot started to correct its position immediately after the disruptive forces had been removed.

Implementation of a Kalman filter slowed the response of the system. It can be seen by observing the time taken for the robot to start correcting its path, which is much slower (the robot travels further along the table before making corrective motions) suggesting the state error covariance matrices had too small of a value. This may also be a symptom of incorrect timesteps used in the filter. There is also a noticeable overshoot at the end of the path (circled). The filtered system shows marginally less error than the direct measurements at estimating the position when disturbances have been applied.

FINAL DESIGN PROPOSAL

DESIGN BRIEF

The aim of the project was to create a small-scale localisation system with millimetre accuracy that could be used to improve Southampton's success at the Eurobot competition. This has been developed to incorporate the needs of stakeholders, which is reflected in the design of the system, robot and test method. The initial objectives were as follows:

- Design a robotic platform that can determine its position on the Eurobot table with millimetre accuracy over 90 seconds
- Create a low-cost final design with respect to leading Eurobot competition teams
- Design a system that can be reused and built on by future teams entering Eurobot
- Develop a reliable method of verifying the accuracy of the robot and comparing it to previous localisation methods

ROBOT DESIGN

The final design employs the following:

SENSORS
Optical sensor

DRIVE
EMG30 motors

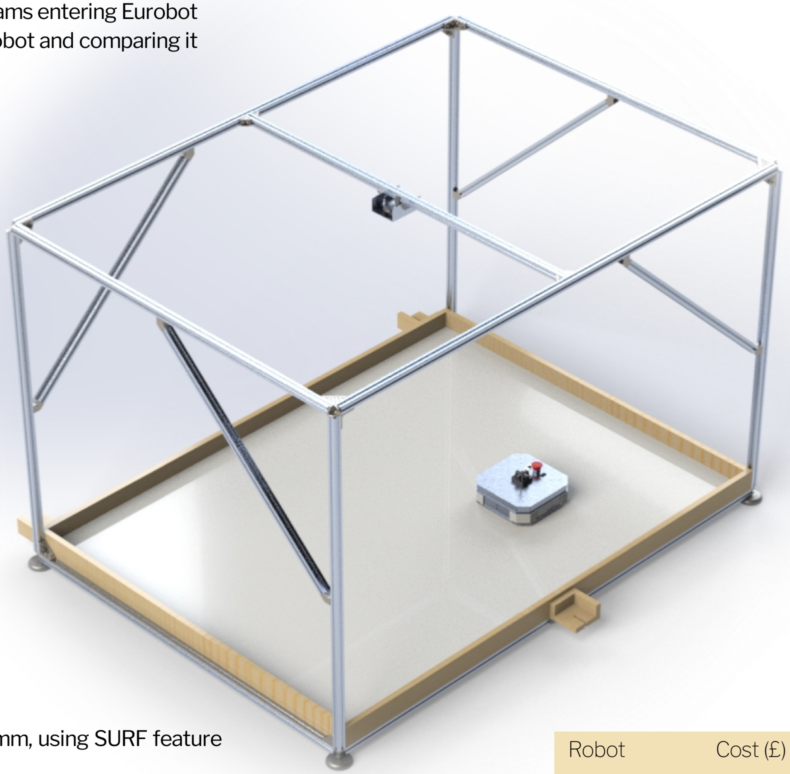
MD25

Omni-wheels

PROCESSING
Raspberry Pi

SYSTEM
Kalman Filter

PID Control



TEST DESIGN

The final design employs the following:

HARDWARE
Aluminium Extrusion (30 x 30 mm)

Water resistant MDF

3 mm Forex board

Logitech C930E webcam

SOFTWARE
SURF Feature Detection

COST BREAKDOWN

The project cost can be broken down into three sections: final system, test method and development.

The cost of the final robot (Table 14) is significantly lower than that spent by top Eurobot teams. This allows for the design to be replicated by future teams, which is reasonable should they receive sponsorship. A more in depth breakdown of the cost is available overleaf.

SUMMARY

The test method measures the location of a robot to within 3 mm, using SURF feature detection in real time.

Because this method uses a mountable QR code to identify the robot, it can easily be applied to other, future projects.

Using a combination of sensor readings and a Kalman filter, the system provides an accurate estimate of the robot's position to within 88 mm.

Using PID control, the robot is capable of being within 25 mm of its target destination, over a 90 second run time. This value is increased to 30 mm on application of a large external disturbance to the system.

The design allows for use in the Eurobot competition through the modular design giving other teams freedom to further develop it to suit their needs.

Robot	Cost (£)	Development	Cost (£)	Test Method	Cost (£)
Chassis	91.24	Accelerometers	17.30	Extrusion	294.79
Optical sensor	18.02	IMU	32.00	Fixings	242.74
Electronics	94.98	Beacon System	410.00	Table	120.80
Drive	126.48	Optical flow sensor	15.62	Camera	102.79
		Battery	21.83		
Total	330.72	Total	651.96	Total	761.12

Table 14: Costing overview.

DETAILED COST BREAKDOWN

The project expenditure can be grouped into three main areas (Table 15). The cost of the final design of the robot, including all manufacturing costs, materials and electronic hardware. The costs incurred from purchasing parts for the test method make up the second area. Lastly, the additional costs from testing the suitability of a variety of sensors are included in the third section. As the project involved trialling a variety of sensors to find the best combination for a localisation application, this is a crucial part of the total spend. The final robot design cost £330.72, which is a feasible cost for a Part II group should they decide to adopt this system.

Funding for the project came from three sources. The GDP core funding of £700 covered the cost of the final robot and some of the sensors that were trialled in the prototype phases of the project. The remaining £107.47 required for purchasing sensors to trial was kindly provided by Boeing, who have historical sponsorship links to the Eurobot Part II module. The team entered the elevator pitch to raise funds for the test method. While the pitch failed to directly get sponsorship, attending the event secured funding for the development of the test method rig through Alastair McDonald.

Robot Total £330.72

	Unit cost	*Number of*	*Total Cost*	*Source*	*Cost to team*
Chassis				Total	£91.24
Aluminium	£25.00	1	£25.00	EDMC	£25.00
WaterJet Cutting	£45.00	1	£45.00	EDMC	£45.00
RS PRO Standoffs	£15.28	1	£15.28	RS Components	£15.28
300 x 300 mm 1 mm Foam	£3.35	1	£3.35	Camthorne Industrial Supplies	£3.35
Acrylic	£17.89	1	£17.89	Sheet Plastics	£17.89

Drives				Total	£126.48
EMG 30 DC Motors	£30.78	4	£123.12	FEEG2001 module	£0.00
MD25 Motor Controllers	£41.00	2	£82.00	FEEG2001 module	£0.00
60 mm Omniwheels	£17.57	4	£70.28	RobotShop	£70.28
Wheel Hubs	£9.80	4	£39.20	RobotShop	£39.20
1 mm Steel	£2.00	1	£2.00	EDMC	£2.00
WaterJet Cutting	£15.00	1	£15.00	EDMC	£15.00

Sensors and Electronics				Total	£113.00
Raspberry Pi Model 3 B +	£32.00	1	£32.00	The Pi Hut	£32.00
Zelotes Gaming Mouse	£13.99	1	£13.99	Amazon	£13.99
PLA for Mouse Sensor Mount	£3.00	1	£3.00	Amazon	£3.00
Suspension for mount	£1.03	1	£1.03	Portswood Hardware	£1.03
USB cables	£5.99	1	£5.99	Amazon	£5.99
T connectors	£0.30	4	£1.20	Amazon	£1.20
Turnigy 5 Ah Lithum Polymer Battery	£21.83	1	£21.83	Hobbyking	£21.83
Strip board	£6.00	1	£6.00	RS Components	£6.00
Sparkfun BOB-12009	£2.25	1	£2.25	Mouser	£2.25
XL4015 DC-DC Buck Converter	£1.10	1	£1.10	Xlsemi	£1.10
SanDisk Ultra 128 GB microSD card	£21.00	1	£21.00	Amazon	£21.00
Jumper cables	£3.61	1	£3.61	Mouser	£3.61

Test Method Total £761.12

	Unit cost	*Number of*	*Total Cost*	*Source*	*Cost to team*
2.044 m Length 30 x 30 mm V-slot Rail	£16.82	4	£67.28	KJN Aluminium Profiles	£67.28
1.044 m Length 30 x 30 mm V-slot Rail	£12.46	8	£99.65	KJN Aluminium Profiles	£99.65
2.000 m Length 30 x 30 mm V-slot Rail	£16.48	6	£98.86	KJN Aluminium Profiles	£98.86
Corner Bracket with Fixings	£4.01	8	£32.06	KJN Aluminium Profiles	£32.06
30 x 30 mm Angle Brackets with fixings	£3.05	24	£73.15	KJN Aluminium Profiles	£73.15
30 x 60 mm Angle Brackets with fixings	£3.96	12	£47.52	KJN Aluminium Profiles	£47.52
30 x 30 mm 45 deg Connector and Fixings	£8.58	8	£68.64	KJN Aluminium Profiles	£68.64
T-nuts	£3.96	2	£7.92	KJN Aluminium Profiles	£7.92
80 mm Diameter Feet and fixings	£13.45	1	£13.45	KJN Aluminium Profiles	£13.45
Extrusion Delivery	£29.00	1	£29.00	KJN Aluminium Profiles	£29.00
MDF Sheets	£70.80	1	£70.80	Totton Timber	£70.80
Camera Mount Steel	£4.80	1	£4.80	EDMC	£4.80
WaterJet Cutting	£15.00	1	£15.00	EDMC	£15.00
1500 x2000 mm 3mm Forex table surface	£50.00	1	£50.00	University Print Centre	£50.00
Logitech 930E Camera	£82.99	1	£82.99	Amazon	£82.99

Additional Development Cost Total £476.75

	Unit cost	*Number of*	*Total Cost*	*Source*	*Cost to team*
BNO055 IMU 9 Axis Orientation Sensor	£32.00	1	£32.00	Mouser	£32.00
MarvelMind Ultrasonic Beacons	£390.00	1	£390.00	MarvelMind	£390.00
Optical Flow Sensor APM2.5 ADNS 3080	£15.62	1	£15.62	Amazon	£15.62
Turnigy 5 Ah Lithum Polymer Battery	£21.83	1	£21.83	Hobbyking	£21.83
3 Axis Accelerometer Adafuit Industries	£6.10	1	£6.10	RS Components	£6.10
Arduino UNO	£20.00	1	£20.00	Arduino	£0.00
3 axis Accelerometer, Magnetometer	£11.20	1	£11.20	RS Components	£11.20

Total Project Cost: £1568.59

Table 15: Cost breakdown of the final robot design, test method and development costs incurred from trialling different sensors to find the most appropriate ones for the final design.

SUSTAINABILITY AND ENGAGEMENT

SUSTAINABILITY

As this was a small-scale project, there were no direct environmental impacts, such as use of fossil fuels or pollutants. However, it is still essential to assess the sustainability and minimise the environmental impact of any modern engineering project.

3D printing produces no waste product during manufacture, and the plastic parts used in the project were designed and orientated to minimise use of support structures during printing. The aluminium robot chassis design was arranged to minimise waste material during waterjet cutting.

Regarding recyclability, PLA plastic is not easily recyclable through regular recycling channels but is possible with plastic composing [22]. Aluminium used in both the robot chassis and the testing table are very easy to recycle – in fact, recycling scrap aluminium requires only 5% of the energy used to make new aluminium from raw ore. As a result, two thirds of all aluminium ever produced is still in use today [23]. For comprehensive sustainability analysis it is also critical to consider the Lithium Polymer battery used. Lithium is a finite resource and global capacity for mining is limited [24]. It is important that the battery is recycled at the end of its life-cycle to mitigate the environmental impact as much as possible.

This project has been also been designed for longevity. The produced outcomes will not simply be disposed of, but instead reused after the project's completion. The extruded aluminium rail used to make the test method frame is used extensively by the university in other projects. The test method is assembled using fastening as opposed to more permanent techniques such as welding. Should the test method no longer be required, the extruded beams can be disassembled and repurposed much more conveniently than if the frame were made from welded aluminium.

ENGAGEMENT

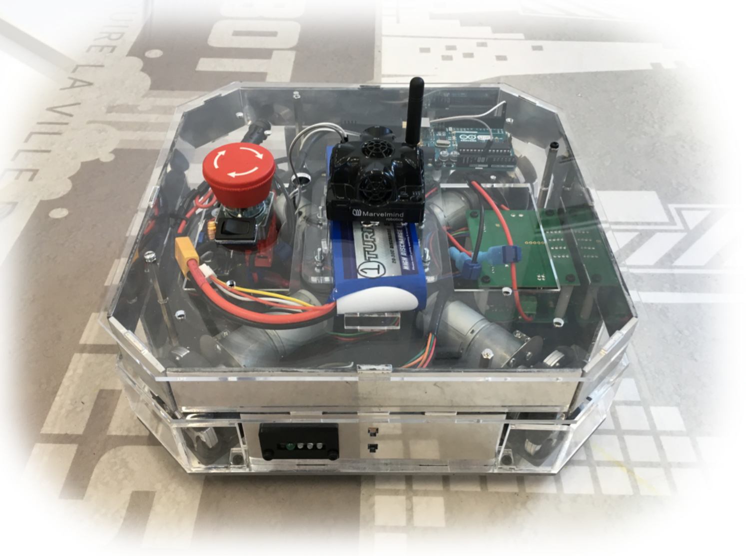
It was important that this project not only satisfied our design requirements, but also accounted for the stakeholder's needs too. Accessibility for future Eurobot teams and RoboSoc was kept in mind from the beginning. All code has been uploaded to GitHub throughout the project and is free, open-source and easily accessible. The front-end GUI is quick and easy to use, and the testing table is ready for use by other members of the university.

Also, in-line with Alastair McDonald's stakeholder requirements, the testing table is versatile for use with other localisation projects, and the interactivity and visual aspects of the project are ideal for the Engineering Design Show 2019.

MARKET VIABILITY

SYSTEM DESIGN

Localisation has many applications in industry, ranging from autonomous vehicles to robotic vacuums. The unique combination of sensors presented in this project are low-cost and could be applied to budget conscious projects which require localisation with centimetre accuracy. However, the target market for this product are students within the university. The robot is predominantly designed to be competitive against external Eurobot teams. Its performance compared to these winning teams is the key driver of its demand within the university.



ROBOT DESIGN

While the robot mechanical design is not the core focus of this project, a platform on which to operate the system was required. The main stakeholders in the robot design were Robosoc and future Part II Eurobot competitors in the university. The robot has been designed to use a versatile platforming design that can be reconfigured for each stakeholder's individual requirements. The final robot design completed here employs an optical sensor and a PID controller for localisation. The positional accuracy is within 25 mm and the total cost is £330.72. Had all of the sensors that were tested been implemented in the final design, the total cost would be approximately £800. This could have increased the positional accuracy further. Even £800 is still significantly lower than the top performing European teams spend, which can be up to £20000.

TEST METHOD

Developing a test method which is accurate to 3 mm and is capable of tracking a robot throughout its motion for less than £800 is an achievement which has commercial viability. There are a number of companies developing products that require localisation, such as robotic vacuums and lawnmowers, that would not have the budget to develop an expensive test method, such as those featuring infra-red cameras costing over £3000 each [20].

PROJECT REVIEW

CRITICAL REVIEW

In this project, there have been many successes. The design was not based on any previous work done at the university, therefore building a robot with a working system for localisation was an achievement. Not only were the objectives largely achieved, additional accomplishments are realised. The implementation of a PID controller that corrects the robot's position throughout the duration of a movement was delivered despite being beyond the initial scope of the project. A bespoke test method employing computer vision techniques was also developed and built, which accounted for one team member's entire workload, which increased the quality of the project but proportionally reduced the number of development hours available for the robot.

The success of achieving the initial objectives, including those introduced from stakeholders are reviewed below:

- *Design a robotic platform that can determine its position on the Eurobot table with millimetre accuracy over 90 seconds*

Using the test method, it has been determined that the robot's estimated position is approximately 20 mm out from its real position after a series of movements lasting 100 s. This is achieved using an optical sensor to measure the distance travelled, with PID control. The error incurred in the system is larger than the 10 mm permissible error stated in the objective, however it is still a significant improvement on the 200 mm error recorded from a typical Eurobot entry. This can be attributed to the drift associated with the optical sensor.

- *Create a low-cost final design with respect to leading Eurobot competition teams*

The final cost of the robot was £330.72. Recreating this design is affordable for Part II Southampton teams with sponsorship. Additional costs were spent in the development phase, allowing for the final design to be of minimal cost. Parts that the university has available were used where possible to reduce the spending required. The cost-benefit of each part was considered allowing the cost of the entire system to be significantly less than top national Eurobot teams.

- *Design a system that can be reused and built on by future teams entering Eurobot*

The chassis is aluminium therefore will survive many years of use. The layered design allows for easy adaption and inclusion of further layers with additional M3 standoffs. The inclusion of a USB port allows for the lower layers to remain unchanged

- *Develop a reliable method of verifying the accuracy of the robot and compare it to previous localisation methods*

Additional funding facilitated the build of a more sophisticated test method. The test method employs computer vision to track a QR code that is mounted above the robot. This is able to identify its position to within 3 mm of the actual position and record the path taken.

- *Incorporate visually appealing, interactive and user-friendly aspects to design, suitable for use in the Engineering Design Show 2019 and University open-days.*

The test method is able to track in real-time with the stated accuracy, allowing for demonstrations to easily be done. Furthermore a GUI was developed, to allow for control of the robot to be demonstrated in an interactive manner.

Although the initial research and development phase enabled the best choice in available sensors, the time this took limited the time available for further development of the final product. With more time, the IMU would have been integrated into the system. Furthermore, if the issues with latency in the beacon system had been discovered sooner, a solution may have been developed so that the beacons could be implemented. This also may have given more time for the Kalman filter development, increasing the frequency above the current 1.6 Hz.

INNOVATION

Throughout the project, options for all aspects of the design were considered. An example of a unique aspect of the design is the inclusion of a repurposed mouse optical sensor. This, alongside research into existing products and methods, allowed for the development of a bespoke robotic system fit for the Eurobot competition.

Meetings with stakeholders ensured that both the robot and test method were developed in the most suitable way for their required purpose. An example of this being minimising the height and perimeter, to maximise the available space for future users. The importance of the financial aspect of the final robot, and the validation of its results were understood from the start of the project, and therefore were recognised in the design brief and objectives.

PROCESS

A prototype was made for the drive system development and other aspects such as the sensors were tested individually before implementing on the robot. This allowed for individuals to work simultaneously on different aspects of the project, and reduce waste from unnecessary iterations. This however had consequences. The time spent on the sensors was longer than initially planned for, which meant postponing the manufacture of the chassis and development of the sensor fusion algorithms.

Research showed there was no suitable existing test method therefore one was specifically developed. Theory was applied for choice of material, and to reduce waste additional bracing was only added where required. The aluminium extrusion has the added benefit of being versatile and reusable in other, future projects.

COMMUNICATION

Communication of the design was key throughout the process, ranging from communication with supervisors to Part II students, each requiring various styles of communication and levels of detail. Technical achievements are detailed within outputs such as this report, and in the accompanying presentation, however a top-level description is used for a broader audience, such as in the video.

CONCLUSION

Although there could be improvements made, the project is a successful solution to a challenging problem. The main successes are a localisation system that can estimate its position to within an accuracy of 20 mm, the completion of a robotic base which can be built upon and used in Eurobot competitions, and a test method that can measure a robot's position and trajectory to within 3mm. Time constraints limited the integration of more sensors, which might have improved the accuracy of the system. Furthermore, the inclusion of a Kalman filter showed promising results as a method, and although it requires further development, may have been more valuable had multiple sensors been included.

FURTHER SCOPE

For all of the achievements of this project, there is still a great deal more that can be done to improve our system. Should this project be carried forward into future years, the following aspects should be considered for further improvement.

PROCESSING POWER

While the testing of each individual sensor with a Raspberry Pi in isolation from the robot yielded promising results, when combined together the performance of the system worsened significantly. Testing highlighted that this issue was attributed to the computational load on the Pi. The reduction in performance was impactful to the point where sensors had to be left from the final design purely because of their computational load. Future work could be aimed to find a suitable alternative to the Raspberry Pi Model 3 B+ that can successfully handle the load of multiple sensors.

SENSORS

Immediate improvements to the system accuracy would most likely come from including readings from the IMU and beacon system into the sensor fusion algorithm. Provided the processing power of the microprocessor is sufficient, the following improvements to the sensors could also be made:

- Beacon System: Using the predictive algorithm mentioned in 'Absolute Positioning System' (or similar). This would allow for increased sampling update rates and, therefore, allow for the Kalman Filter to be run at higher speeds.
- Computer Mouse Sensor: Higher performance gaming mice are available on the market; replacing the current Zelotes gaming mouse could improve the accuracy of the sensor system. Another improvement could come from using two computer mice which would allow for a second angle measurement to be calculated (IMU being the first).

MOTOR CONTROLLER

The MD25 motor controllers proved to be a limiting factor for the sampling rate of the sensor fusion algorithm. Each message sent along the I2C bus must be proceeded by a 50 ms delay when communicating with the MD25 motor controller. This means that the sampling rate for sensor readings was limited to around 4 Hz, as there were four messages to be sent in each drive loop. Further work could be dedicated to reviewing the suitability of different motor controllers that can handle communications more quickly, therefore allowing quicker update rates for the filter.

SENSOR FUSION

The implemented Kalman filter is currently only designed to estimate coordinate position of the robot. Further work on the sensor fusion algorithm would include angular position in the Kalman filter model so that readings from the IMU orientation sensor could be implemented to control the robot's angular position. Additional angle corrective movements would need to be programmed in to the robot's movements. Further investigation into different sensor fusion methods to find the most appropriate would also be a valuable area in which to dedicate work.

TEST METHOD

To improve the test method, future work should focus on increasing the resolution and improving the user interface. Using a 4K camera would increase the number of pixels by a factor of four, effectively quadrupling the resolution. This would allow for reduced uncertainty in the measurements and increased confidence in the results. The main obstacles to this work will be computational power of the laptop processing the data.

Currently the test method software is operated via command line. It would be beneficial to develop a GUI to make the testing easier to operate and more user friendly.

REFERENCES

[1] Planete Sciences, "Eurobot 2019 Rules," 10 October 2018. [Online]. Available: http://www.eurobot.org/images/2019/Eurobot2019_Rules_Cup_OFFICIAL_EN.pdf. [Accessed 8 May 2019].

[2] W. Elmenreich, "An Introduction to Sensor Fusion," Institut fur Technische Informatik, Vienna, 2002.

[3] IRobot, "Roomba e5," IRobot, 2019. [Online]. Available: https://shop.irobot.co.uk/at/roomba-vacuum-cleaning/irobot-roomba-e5/e515840.html?_ga=2.138579187.1721911632.1557341089-756223146.1557341089&lang=en_GB. [Accessed 8 May 2019].

[4] M E Porter. (1980). Competitive Strategy. Free Press ISBN 0-684-84148-7

[5] J. Vander, D. Hagel, M. Blaich, and O. Bittel, "Mobile Robot Localization Using Beacons and Kalman Filter Technique for Eurobot Competition," vol. 33, no. June, 2009.

[6] RobotShop, "RPLidar A1M8 - 360 Degree Laser Scanner Development Kit." [Online]. Available: RPLidar A1M8 - 360 Degree Laser Scanner Development Kit. [Accessed: 05-May-2019].

[7] Robert Holmberg, "Design and Development of Powered-Castor Holonomic Mobile Robots", Stanford University, 2000

[8] Roland Siegwart, Illah R. Nourbakhsh "Introduction to Autonomous Mobile Robots", The MIT Press, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2004

[9] Robu.in, "100 mm Aluminium Mecanum Wheel," MACFOS PVT.LTD, 2019.

[10] Robot Shop, "60 mm Aluminium Omni Wheel," Robot Shop, 2019. [Online]. Available: https://www.robotshop.com/uk/60mm-aluminum-omni-wheel.html. [Accessed 8 May 2019].

[11] A. Birari and e. al, "Velocity control of omni drive robot using PID controller and dual feedback," *IEEE First International Conference on Control, Measurement and Instrumentation,* 2016.

[12] J. Owoyemi, "Kalman Filter: Predict, Measure, Update Repeat," Medium, 2017.

[13] P. Goel and e. al, "Robot Localization Using Relative and Absolute Position Estimates," Institute for Robotics and Intelligent Systems, Los Angeles, 1999.

[14] Lowe, D.G., 1999, September. Object recognition from local scale-invariant features. In *iccv* (Vol. 99, No. 2, pp. 1150-1157).

[15] Bay, H., Ess, A., Tuytelaars, T. and Van Gool, L., 2008. Speeded-up robust features (SURF). *Computer vision and image understanding*, *110*(3), pp.346-359.

[16] M. M. El-gayar, H. Soliman and N. Meky, "A comparative study of image low level feature extraction algorithms," *Egyptian Informatics Journal,* vol. 14, no. 2, pp. 175-181, 2013.

[17] Rublee, E., Rabaud, V., Konolige, K. and Bradski, G.R., 2011, November. ORB: An efficient alternative to SIFT or SURF. In *ICCV* (Vol. 11, No. 1, p. 2).

[18] Rosten, E. and Drummond, T., 2006, May. Machine learning for high-speed corner detection. In *European conference on computer vision* (pp. 430-443). Springer, Berlin, Heidelberg.

[19] Calonder, M., Lepetit, V., Strecha, C. and Fua, P., 2010, September. BRIEF: Binary robust independent elementary features. In *European conference on computer vision* (pp. 778-792). Springer, Berlin, Heidelberg.

REFERENCES

[20] Vicon, *Vicon Camera Systems,* 2019. [Online]. Available: https://www.vicon.com/products/camera-systems

[21] J. B. L. Feng, "UMBmark: A Benchmark Test for Measuring Odometry Errors in Mobile Robots," in *SPIE Conference on Mobile Robots*, Philadelphia, 1995.

[22] "How are Plastics Recycled?," Recycle More, [Online]. Available: https://www.recycle-more.co.uk/how-is-it-recycled-/plastic-recycling. [Accessed 1 May 2019].

[23] "About Aluminium," Constellium, [Online]. Available: https://www.constellium.com/about-constellium/about-aluminium. [Accessed 1 May 2019].

[24] "Lithium Statistics and Information," United States Geological Survey, 2018 [Online]. Available: https://www.usgs.gov/centers/nmic/lithium-statistics-and-information. [Accessed 1 May 2019].