

Autonomous Mobile Robot for Remote Location Inspection

Eric Pairet Artau

University of Edinburgh

E.Pairet-Artau@sms.ed.ac.uk

Abstract— Robotics is a multidisciplinary science; solving a particular problem involves many areas of research. A clear example is the challenge motivating this work, in which a robot has to autonomously navigate through a priori known arena, look for features to localise itself and successfully align an antenna. This report presents the design of a LEGO robot which uses a decoupled 2 proportional controller for solving a point-to-point navigation, uses a modified k-nearest neighbours classifier to recognise point of interest in an image acquired with a low-cost camera, and solves the inverse kinematics problem to align the antenna. The robot is localised using an extended Kalman filter, which merges the dead-reckoning of an on-board Hall effect sensor with some feature lines extracted with two looking-forward infra-red sensors. The robot was tested in eight-time trials. The antenna alignment success rate was of 8, 5 and 3 out of 8 at the first, second and third point of interest, respectively. The main limitation of our proposal is that the localisation algorithm is unlikely to recover the robot's correct position once it has misassociated some features.

I. INTRODUCTION

Recent advances in artificial intelligence (AI) are revolutionising the field of robotics at the level of cars autonomously driving in the roads [1] and the Da Vinci robot being used in real laparoscopy surgery [2]. Ironically, some of the challenges faced by these cutting-edge robots are similar to the problems confronted by less impressive machines, such as the Roomba vacuum cleaner [3]; perception, mapping, navigation, planning and control are just some of the fields where any robot struggles at.

Many of the aforementioned fields need to be considered to solve the challenge motivating this work. In summary, it consists of creating a robot with LEGO able to inspect a remote location (see Figure 1), i.e. (i) move through a priori known arena, (ii) detect three point of interest (PoI) characterised with reflective tape and (iv) correctly align its antenna to a satellite, the location of which is known.

This work approaches the challenge designing the car-like robot depicted in Figure 2. The vehicle is equipped with a set of proprioceptive and exteroceptive sensors which allow to (i) autonomously drive the vehicle to a desired waypoint using a decoupled proportional control, (ii) detect the PoIs on the arena using a modified k-nearest neighbours (KNN) classifier [4] and (iii) localise itself in the arena by means of an extended Kalman filter (EKF) [5] and (iv) precisely align the antenna towards the satellite. These tasks are controlled

The author wants to thank the RSS laboratory assistants for their priceless suggestions through the development of this work, and the teammate Paola Ardón for enormously contributing to this project.

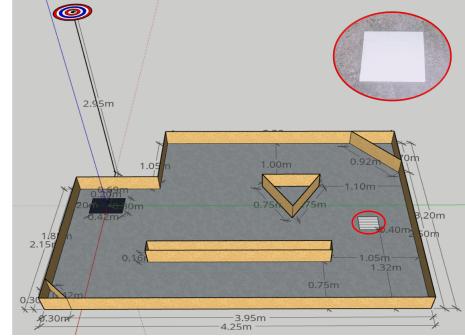


Fig. 1: Schematic of the arena. The black square is the world's reference frame, i.e. where the x-axis (red), the y-axis (green) and the z-axis (blue) cross. On the top left corner, the satellite. On the top right corner, an example of a PoI.

by a generic framework which allows the robot to operate autonomously under different challenges setups.

The remainder of this report is organised as follows. The physical platform design and modelisation are described in Section II. Our proposed framework and used algorithms are detailed in Section III. Some pieces of the proposal and the overall framework have been evaluated; the obtained results are presented and discussed in Section IV. Finally, remarks and proposals for future work are given in Section V.

II. PHYSICAL PLATFORM

Building a robot (even if it is made of LEGO) is not a trivial task. Thus, this section describes the designed LEGO robot and the modelisation of its sensors in Section II-A and Section II-B, respectively.

A. Design

Aiming to build a compact robot which is easy to deploy and recover, stable, rotates with respect to its centre of mass (CM) and allows to easily integrate new sensors, this work designed and built the robot depicted in Figure 2. Its xyz-dimensions are $261 \times 248 \times 287\text{mm}$, and the location of its most important sensors is reported in Table I.

The vehicle is driven by two independent active wheels of 82mm of diameter. Each wheel is commanded with a LEGO 43362 motor [6]; the gear train depicted in Figure 3 has been used to get the wheels turning with an adequate torque and rpm rate. These two wheels and a castor wheel located at the front of the vehicle constitute a stable base where

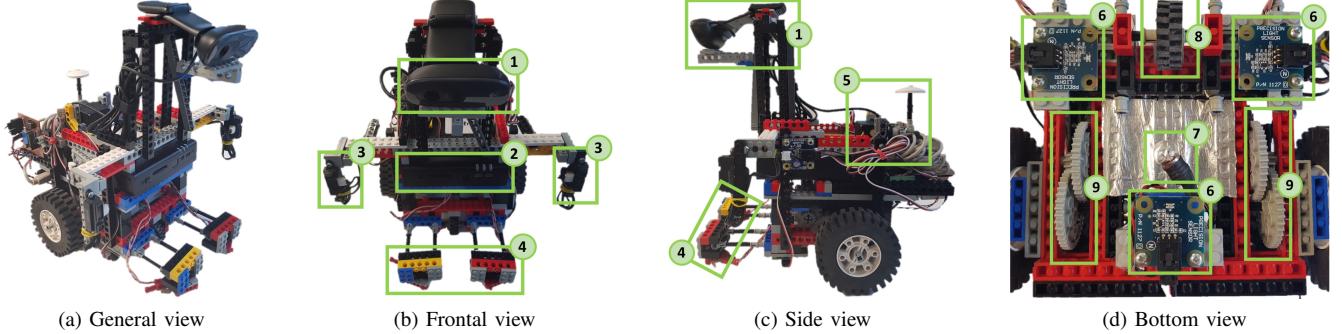


Fig. 2: Different views of the designed LEGO robot. Labels: (1) camera, (2) Fit-PC2, (3) IR sensors, (4) bumper with switchers, (5) antenna commanded by the servomotor, (6) light sensors, (7) light bulb, (8) castor wheel and (9) gear trains.

all electronics, the antenna commanded with a HS-322HD servomotor [7] and the sensors (a low-cost camera [8], a Hall effect sensor, three P/N 1127 light sensors [9], two SHARP GP2D12 IR sensors [10] and two micro switches) are stacked on. All actuators and sensors are centralised in the robot's brain; a Fit-PC2 [11].

Element	Location $[x, y, z]$
Wheel (left)	$[0, 88.5, 0]$
Wheel (right)	$[0, -88.5, 0]$
Castor wheel	$[83, 0, -27]$
IR (left)	$[126, 124, 94]$
IR (right)	$[126, -124, 94]$
Camera	$[151, 0, 208]$
Servomotor	$[-23, 0, 111]$

TABLE I: Location of the most important elements with respect to the robot's local frame, which is estimated to be at the height of the main wheels and between them.

B. Modelisation

In order to have an accurate understanding of the robot's dynamics and its sensors, this work has modelled the Hall effect, the IR and the light sensors.

1) *Hall effect sensor*: is attached to the gear train that connects a motor with its corresponding wheel. Figure 3 shows that the gear ratio between the sensor and the output shaft is 25:1. This lets us compute that a pulse of the Hall sensor can either mean a displacement $\Delta x = 10.3\text{mm}$ along the robot's x-axis or a rotation $\Delta\theta = 6.28^\circ$ with respect to the robot's z-axis, as formulated below:

$$\Delta x = \frac{1}{25} 2\pi R_w, \quad (1)$$

$$\Delta\theta = \frac{360}{2\pi R_w \Delta x}, \quad (2)$$

where $R_w = 41\text{mm}$ is the wheel radius and $R_r = 88.5\text{mm}$ is the distance from a wheel to the robot's reference frame.

2) *IR sensors*: are placed looking forward to measuring the distance from the vehicle to any obstacle located in front of it. For that purpose, a model is needed to convert the IR readings to a metric measurement. Figure 4 reports the mean and variance of the measurements obtained with

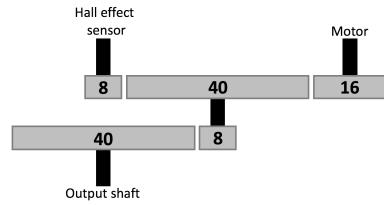


Fig. 3: Schematic of the gear train.

the left and right IR sensors when placing an obstacle at different distances within the range from 100mm to 800mm according to the manufacturer specifications [10].

The gathered data depicts that the measurements from the IR sensors do not have a large variance, thus being a reliable source for measuring distances. In fact, a model with an R-square goodness of 0.9965 and 0.9988 for the left and right sensor, respectively, have been achieved by fitting the data with the following power model:

$$distance_i = a_i * IR_i^{b_i}, \quad (3)$$

where the subscript i is left (L) or right (R), IR_i is the i sensor's measurement, and $[a_L, b_L] = [104.2, -1.104]$ and $[a_R, b_R] = [159.2, -1.196]$ are the model parameters.

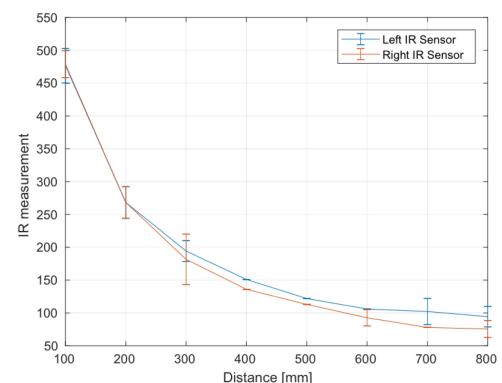


Fig. 4: Measurements from the left and right IR sensors when placing an obstacle at each 100mm. The mean and variance at each point are computed using 13,000 samples.

3) *Light sensor*: is placed underneath the robot (see Figure 2d) to detect the illumination changes on the floor. As reported in Table II, setting a threshold of 50 would allow detecting when the robot is on top of a PoI. To get a more robust method (a) a light bulb is placed near the light sensors to get more contrasted readings (now the threshold should be set at 225) and (b) acknowledging that the ambient light might change, the threshold is dynamically set as the initial reading of the light sensor plus the previously computed 225.

	On floor	On tape
Without bulb	4	95
With bulb	83	368

TABLE II: Average measurements from the light sensors under different scenarios (4,000 samples per scenario).

III. FRAMEWORK AND METHODS

Figure 5 depicts the framework which allows the LEGO robot to solve the challenge stated in Section I. Such architecture consists of four parts: (i) the perception module analyses the surrounding environment, (ii) a localisation module estimates the robot’s position, (iii) a control module operates the robot’s actuators, and (iv) a framework manager which coordinates all the modules to achieve the predefined mission “explore the arena, find and locate yourself on top of the PoIs, and orient your antenna to the satellite”.

Apart from the mission tasks, the robot needs to keep localised and safe. The framework hierarchically coordinates all requirements as follows: the vehicle will normally explore the arena as explained in Section III-A and if it finds a PoI with the technique disclosed in Section III-B, the robot will activate the antenna alignment behaviour formulated in Section III-C. Concurrently, the algorithms described in Section III-D and Section III-E estimates the robot’s position and guarantees the robot safety, respectively¹.

A. Arena exploration

The exploration of the arena is guided by a manually planned trajectory (see Figure 6). This approach aims to

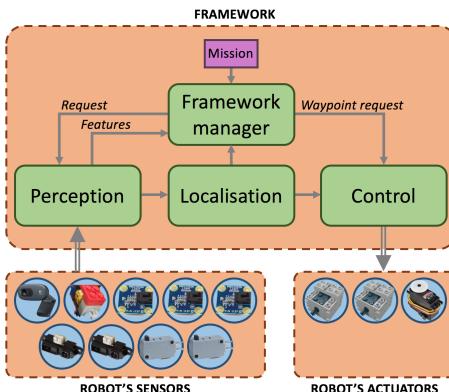


Fig. 5: Framework designed for the LEGO robot.

¹All code for the reported framework and methods is available at https://ericpairet@bitbucket.org/ericpairet/rss_notropis.git. Please contact the author for access permission.

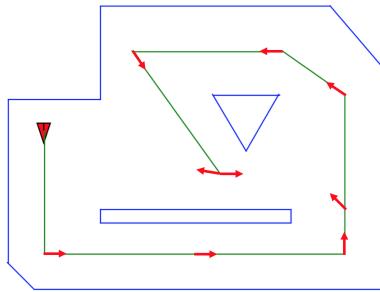


Fig. 6: Manually planned trajectory (green line) which leads to the strategic angle (red arrow) to look for PoIs.

speed up the PoIs localisation by (a) driving the robot around the arena through safe paths and (b) placing the robot where its camera sees the most of the remaining unexplored arena. Since acquiring a frame in low resolution implies approximately 2'', image acquisition and the processing in Section III-B is only executed in these strategic points.

To drive the robot a specific xy-point with respect to the world’s local frame, a decoupled proportional controller which prioritises the heading error is set for each degree of freedom (DoF) of the robot. Because the robot has not an infinitesimal precision, some tolerance is allowed to the linear and angular movements. Independently moving each DoF avoids having to take into account the non-holonomic constraints of a classic car-like system [12].

B. PoI detection

Different approaches have been undertaken to detect a possible PoI in an image: colour segmentation in the RGB and the HSV spaces (not robust under different light conditions) lines extraction using the Hough transform [13] (implies the non-trivial task of selecting the lines of interest), and pixel segmentation using the KNN [4] and the Extra Trees [14] classifiers (both have a satisfactory performance but takes around 90'' and 10'' per image, respectively).

The techniques that satisfactorily generalise the segmentation task have a large computation time. Aiming to speed up the KNN classification, the values for each possible combination of RGB vectors is pre-calculated and stored in a $256 \times 256 \times 256$ -sized lookup table (see Figure 7). This pre-processing step allows to obtain the segmentation quality of the KNN classifier with only 20ms per image, but at the cost of storing the aforementioned lookup table.

The resulting mask might contain undesired wall areas that are reflective due to varnish. Leveraging that they are easy to

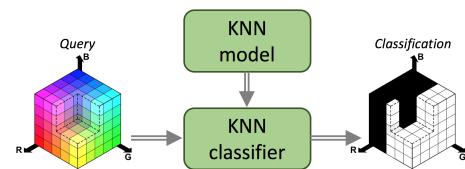


Fig. 7: Schematic of how the KNN classifier is extracted to a binary lookup table.

segment in the colour space, the wall mask is subtracted from the PoI mask. The enhanced segmentation let us compute the rotation needed to head to the PoI as:

$$\theta_{PoI} = \left(0.5 - \frac{CM_x}{width} \right) * FoV, \quad (4)$$

where CM_x is the x-location of the mask's CM, $width = 160$ is the low resolution image width and $FoV = 60$ [8]. After correcting the heading, the robot moves towards the PoI until the light sensors determine that it got on top of it; then, the antenna is aligned as formulated in Section III-C.

C. Antenna alignment

Determining the satellite's location with respect to the robot's antenna (${}^a p_s$) is indispensable for succeeding on the antenna alignment task. This relative position is computed applying inverse kinematics (IK) as:

$${}^a p_s = {}^r H_a^{-1} * {}^w H_r^{-1} * {}^w p_s, \quad (5)$$

where ${}^r H_a$ is the homogenous transformation describing the antenna's location with respect to the robot's frame, ${}^w H_r$ is the homogenous transformation describing the robot's location with respect to the world's frame, and ${}^w p_s$ is the satellite's known position with respect to the world's frame.

The obtained ${}^a p_s$ let us compute by means of trigonometry the angle that both the robot and the antenna have to turn to get the antenna pointing to the satellite.

D. Robot localisation

The robot is localised in the arena using the EKF [5], [15], which has been adapted to merge (i) the Hall effect sensor-based dead-reckoning and (ii) the pose estimation extracted from the two looking-forward IR sensors. An overview of the threefold sequence of such filter is given below.

1) *Prediction*: formulating the dynamics in function of time turned to be an unreliable approach because the motors' performance depends on the battery's voltage. Thus, the robot's position is predicted using the Hall effect sensor. For each obtained pulse, the current robot's motion is used to quantify a local $xy\theta$ -movement: $[\Delta x, 0, 0]$ (forward), $[-\Delta x, 0, 0]$ (backward), $[0, 0, \Delta\theta]$ (left turn) or $[0, 0, -\Delta\theta]$ (right turn). Finally, the robot's new position is predicted by compounding (non-linear operation) the local motion to the previous robot's location.

2) *Data association*: due to the fixed location of the IR sensors, this step starts by stopping the vehicle and turning it towards the expected nearby features, i.e. any wall within the sensors range. This routine is done (i) periodically each 75cm, (ii) before aligning the antenna and (iii) when an unexpected obstacle is found on the way.

Regarding the EKF algorithm itself, any observed wall is encoded as a polar line. The Mahalanobis distance between the observed feature and each line composing the a priori known map is computed. Among all the comparisons, the smallest Mahalanobis distance is kept and if it is smaller than a chi-square $\chi^2_{\rho,\varphi}$ test threshold, the line is considered associated. This work has set the chi-square threshold for 2 DoF and a 95% of confidence, i.e. $\chi^2_{2,0.95} = 0.103$.

3) *Update*: this step has not been adapted for this work, so it makes use of the standard linear equations of the EKF to update the robot's state estimation.

E. Obstacle avoidance

The robot is not meant to collide because (i) follows a predefined collision-free trajectory and (ii) moves towards an obstacle which is in its field of view (FoV). However, this reasoning only holds if the robot's localisation is constantly accurate. Thus, aiming to guarantee the robot's safety, a two-layered obstacle avoidance is included in the framework.

The first layer is conservationist; uses the IRs sensors to detect nearby obstacles. Because the IR sensors have a blind spot [10], the second layer uses the switches located at the front bumper of the robot. In both cases, the vehicle is immediately stopped to localise itself back. Then, the vehicle gets to the desired point, if needed, surrounding the obstacle using the Bug tangent algorithm [16].

IV. EXPERIMENTAL EVALUATION

This section evaluates the most important parts of the proposed framework and the overall performance of it. Specifically, the PoI detection approach is analysed in Section IV-A, the accuracy of the localisation system is discussed in Section IV-B and finally, the overall framework is quantitatively evaluated in Section IV-C.

A. PoI detection

Figure 8 illustrates all the methods that have been tested to detect PoIs in an image. As reported in Section III-B, the selected approach is a modified KNN classifier able to segment a low resolution frame in approximately 20ms. Such method has been exhaustively checked, showing a high level of robustness under different environmental conditions. However, (i) the camera has a blind spot on the imminent area in front of the robot and (ii) the undertaken approach does not succeed on segmenting PoIs located further than approximately 1m from the camera. This might be due to the lack of training samples at this distance.

B. Robot localisation

According to the Hall effect sensor modelisation (see Section II-B.1), the dead-reckoning-based localisation has an accuracy of 10.3mm and 6.28° for linear and angular movements, respectively. Many efforts were made to get a higher localisation accuracy out of this sensor by placing it nearer the motor shaft, but the Phidget board only handles 125 samples per second [17].

A graphical user interface (GUI) has been set up to verify that the programmed EKF performs as expected. Because the Fit-PC2 has a lack of plotting libraries, such GUI runs in local while taking the information of interest through secure shell (SSH). The normally observed EKF's behaviour is an uncertain estimation of the robot's true position when only predicting (see Figure 9a), which is bounded and updated after associating the observations (see Figure 9b and Figure 9c). In each association the uncertainty only gets bounded

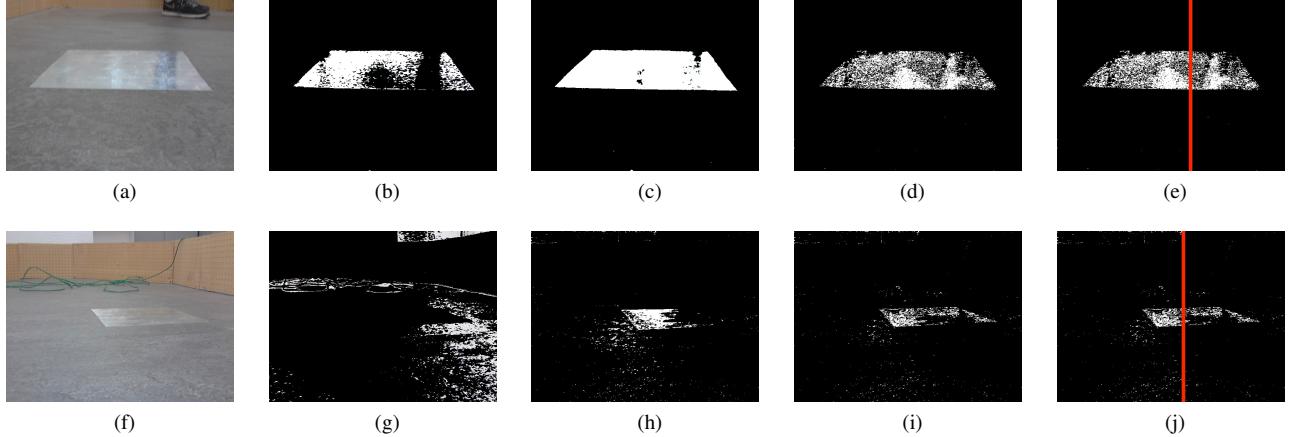


Fig. 8: Automatic detection of PoIs. First row: (a) training image, (b) segmented image using a colour space, (c) segmented image using Extra Trees classifier, (d) segmented image using the modified KNN classifier, and (e) computed mask’s CM. The second row: (f) validation image, (g) segmented image using the same thresholds as in image (b), (h) segmented image using the Extra Trees classifier, (i) segmented image using the modified KNN classifier and (j) computed mask’s CM.

to the perpendicular axis of the wall, thus needing at least two associations in different headings for accurately estimating the robot’s true position. Even though the performance of the EKF is generally satisfactory, sometimes fails the data association; it misassociates the observed feature with one of the walls at 45° . Then, the robot’s heading gets an error of approximately 45° , from which the EKF is unlikely to be recovered. Being able to monitor the robot’s behaviour has been crucial for understanding this fact.

C. Overall framework

The overall framework has been tested in eight-time trials. At each trial, one of the fixed PoI was hidden and some reflective tape was placed in a new location. The robot got to the first PoI 8 out of 8 (8 well localised), to the second PoI 6 out of 8 (5 well localised) and to the third PoI 5 out of 8 (3 well localised). When the robot localisation is accurate, the antenna alignment behaviour always succeeds. The average time to complete an entire mission is of 4’20”.

In the day of the challenge, the robot scored 12 points out of 16; it found and got well localised to 2 of the existent PoIs on the arena. While looking for the third PoI, the robot got lost and the images were not gathered in the correct orientation. Thus, the robot never got to the third PoI.

V. DISCUSSION AND FUTURE WORK

This paper has presented a LEGO robot prototype and a framework which jointly addresses the proposed challenges on autonomous navigation, feature detection, IK and localisation. The proposal was tested in eight-time trials, in which the antenna alignment succeeded 8, 5 and 3 out of 8 at the first, second and third PoIs, respectively. The results demonstrated the suitability of both the robot’s design and the framework to approach different challenge setups, but further work needs to be done to increase the robustness of the system.

Building a lookup table for rapidly knowing the KNN classifier decision has been crucial for accomplishing the challenge within the given time. However, exploring ways

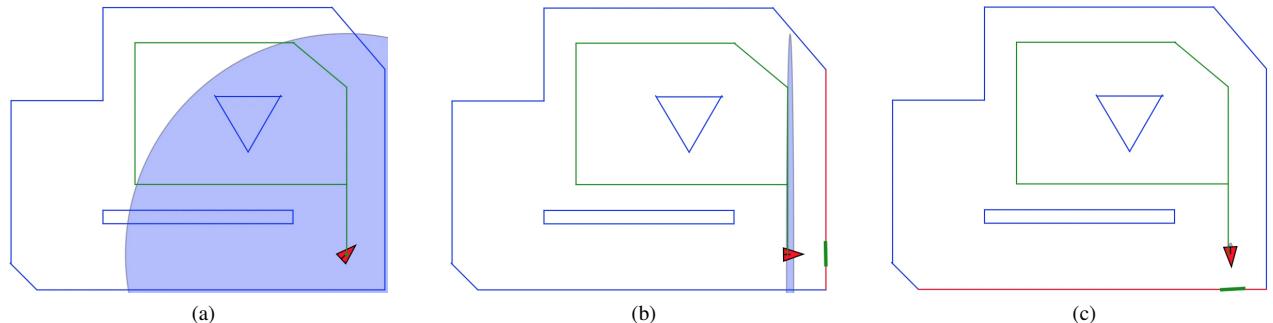


Fig. 9: Robot’s pose estimation using an EKF-based localisation. Robot’s pose estimation uncertainty (purple), perceived wall (thick green line) and associated line from the a priori known map (red). (a) Predicted robot’s position after many iterations, (b) data association and robot’s position update (note the angle change), and (c) data association and robot’s position update. Note that the uncertainty has been exaggerated for visualisation purposes.

of acquiring images in real time would allow using the camera on a regular basis. A more robust localisation system was expected when choosing the EKF. The main limitation is the need of rotating the whole robot to detect nearby features. Assembling the sensors on a rotating platform would drastically reduce the number of robot manoeuvres. Moreover, because there are cases where the EKF does not recover the correct robot's localisation, a particle filter approach would be worth to explore.

REFERENCES

- [1] R. Shanker, A. Jonas, S. Devitt, K. Huberty, S. Flannery, W. Greene, B. Swinburne, G. Locraft, A. Wood, K. Weiss, *et al.*, "Autonomous cars: Self-driving the new auto industry paradigm," *Morgan Stanley Blue Paper*, November, 2013.
- [2] I. Broeders and J. Ruurda, "Robotics revolutionizing surgery: The intuitive surgical "da vinci" system," *Industrial Robot: An International Journal*, vol. 28, pp. 387–392, 10 2001.
- [3] J. Forlizzi and C. DiSalvo, "Service robots in the domestic environment: a study of the roomba vacuum in the home," in *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pp. 258–265, ACM, 2006.
- [4] "MathWorks Classification KNN Class." <https://www.mathworks.com/help/stats/classificationknn-class.html>. Online; accessed 30 October 2017.
- [5] R. E. Kalman *et al.*, "A new approach to linear filtering and prediction problems," 1960.
- [6] Philippe Hurbain, "LEGO 9V Technic Motors compared characteristics." <http://www.philohome.com/motors/motorcomp.htm>, 2008. Online; accessed 26 September 2017.
- [7] Hitec company, "HS-322HD Standard Heavy Duty Servo." <http://hitecrcd.com/products/servos/sport-servos/analog-sport-servos/hs-322hd-standard-heavy-duty-servo/product>, 2017. Online; accessed 11 October 2017.
- [8] Logitech company, "Logitech HD webcam C270 technical specifications." http://support.logitech.com/en_us/article/17556, 2014. Online; accessed 30 October 2017.
- [9] Phidgets, "P/N 1127 Precision Light Sensor." <https://www.phidgets.com/?tier=3&catid=8&pcid=6&prodid=99>, 2010. Online; accessed 26 September 2017.
- [10] SHARP Corporation, "GP2D12 Optoelectronic Device." https://engineering.purdue.edu/ME588/SpecSheets/sharp_gp2d12.pdf, 2005. Online; accessed 26 September 2017.
- [11] fitPC company, "fit-PC2 specifications." <http://www.fit-pc.com/web/products/fit-pc2/fit-pc2-2i-specifications/fit-pc2-specifications/>, 2017. Online; accessed 26 September 2017.
- [12] B. Donald, P. Xavier, J. Canny, and J. Reif, "Kinodynamic motion planning," *Journal of the ACM*, vol. 40, no. 5, pp. 1048–1066, 1993.
- [13] R. O. Duda and P. E. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.
- [14] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine learning*, vol. 63, no. 1, pp. 3–42, 2006.
- [15] M. I. Ribeiro, *Kalman and Extended Kalman Filters: Concept, Derivation and Properties*. Institute for Systems and Robotics. Instituto Superior Tecnico, February 2004.
- [16] I. Kamon, E. Rivlin, and E. Rimon, "New range-sensor based globally convergent navigation algorithm for mobile robots," in *Proceedings - IEEE International Conference on Robotics and Automation.*, vol. 1, pp. 429–435, April 1996.
- [17] Phidgets, "PhidgetInterfaceKit 8/8/8." <https://www.phidgets.com/?tier=3&catid=2&pcid=1&prodid=18>, 2010. Online; accessed 30 October 2017.