Mobile Robot Localisation and Navigation Using LEGO NXT and Ultrasonic Sensor

Yanan Liu[1], *Graduate Student Member, IEEE,* Rui Fan[2], *Member, IEEE,* Bin Yu[1],
M. Junaid Bocus[3], *Graduate Student Member, IEEE,* Ming Liu[2], *Senior Member, IEEE,* Hepeng Ni[4],
Jiahe Fan[5], *Student Member, IEEE,* Shixin Mao[6]

Abstract— Mobile robots are becoming increasingly important both for individuals and industries. Mobile robotic technology is not only utilised by experts in this field but is also very popular among amateurs. However, implementing a mobile robot to perform tasks autonomously can be expensive because of the need for various types of sensors and the high price of robot platforms. Hence, in this paper we present a mobile robot localisation and navigation system which uses a LEGO ultrasonic sensor in an indoor map based on the LEGO MINDSTORM NXT. This provides an affordable and ready-to-use option for most robot fans. In this paper, an effective method is proposed to extract useful information from the distorted readings collected by the ultrasonic sensor. Then, the particle filter is used to localise the robot. After robot's position is estimated, a sampling-based path planning method is proposed for the robot navigation. This method reduces the robot accumulative motion error by minimising robot turning times and covering distances. The robot localisation and navigation algorithms are implemented in MATLAB. Simulation results show an average accuracy between 1 and 3 cm for three different indoor map locations. Furthermore, experiments performed in a real setup show the effectiveness of the proposed methods.

Index terms—mobile robot, LEGO NXT, localisation and navigation, particle filter, path plan.

I. INTRODUCTION

Mobile robot localisation and navigation technologies play an important role in modern-day robots, which are being increasingly used in industry, driver-less cars, assisted living, logistics and domestic applications, especially in an ageing society [1]. The state-of-the-art mobile robot technologies have been significantly enhanced with the furthered development of sensors and algorithms. Consequently, mobile robots are utilised in a wider range of applications than ever before. Companies such as JD.com are testing motorised robots for delivery purposes in universities [2]. Moreover, Google and Baidu are developing a new generation of autonomous driver-less cars [3]. Automatic cleaning robots are being increasingly used in households nowadays [4]. People are fascinated by the rapid development of mobile robots and hence a lot of research is being devoted to this field to make our lives better. Unfortunately, most of the advanced sensors are expensive for individuals. Hence, in this paper, we demonstrate an affordable and easy-to-use option for robot amateurs and allow them to develop a mobile robot system using the LEGO NXT and an ultrasonic sensor.

The LEGO NXT [5] is an off-the-shelf tool-kit to verify mobile robot control algorithms. It encompasses a new generation of programmable, educational [6] and modularised robotics, enabling developers to easily understand, design and program robots [7].

A number of mobile robotic systems have been developed based on the LEGO NXT with various different types of sensors [8]. Lee and Buitrago [9] proposed the map generation and robot localisation system which simply uses a PC camera and an ultrasonic sensor. However, the accuracy of robot positioning can be largely affected by a change in illumination in the environment. Consequently, it is unlikely to achieve robust operation. A Radio-Frequency IDentification (RFID)-based localisation framework was put forward by Chawla and Robins, which was realised using the LEGO NXT kits [10]. However, this solution requires putting tags in the environment for the readers to track. Therefore, the localisation accuracy depends on the position of the tags to some extent. Moreno et al. [11] and Li et al. [12] also achieved robot localisation and navigation with several or several types of sensors. Thus, these solutions, where many sensors are used, would be more expensive.

In this paper we utilise a LEGO ultrasonic sensor as the only perception tool which performs both the localisation and navigation tasks. This method not only makes the robot system inexpensive but also simplifies the software architecture. However, the ultrasonic sensor data is often mixed with propagation phenomena such as noise, echoes, reflections, attenuation and other limiting factors [13]. Consequently, to obtain an accurate estimation of the robot position, the sensor readings should be manipulated in such a way that useful information can be extracted from the noisy and distorted data. Hence, in this paper, a practical method is proposed to process the acoustic data for localisation purposes. The

[1]Bristol Robotics Laboratory, University of Bristol, Bristol, United Kingdom. yanan.liu@bristol.ac.uk

[2]Robotics and Multi-Perception Laboratory, Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong.

[3]Department of Electrical and Electronic Engineering, University of Bristol, Bristol, BS8 1UB, United Kingdom.

[4]School of Mechanical Engineering, Shandong University, Jinan, 250061, China.

[5]School of Communication Engineering, Jilin University, Changchun, 130000, China.

[6]Shenzhen Sanhang Industrial Technology Research Institute, Shenzhen, 518000, China.

Fig. 1: Overview of the proposed robot control process.

Fig. 2: The mechanical structure of the mobile robot.

processed sensor data acts as an input for the particle filter method, which probabilistically estimates the robot position [14]. The particle filter approach is widely applied to solve uncertainty problems based on probability theory and the Bayesian theorem [15]. Once the robot's position is probabilistically estimated, a path to the target position is planned with the sampling-based method [16]. Unlike the popular A* path planning method, which produces a gridded trajectory and usually causes localisation error in a gridded decomposed map [17], this paper uses sampling based path planning method to select optimal sample points and form a short, collision-free trajectory with minimal turning times. A re-localisation and re-navigation mechanism is designed in case the robot is positioned incorrectly.

Finally, all the robot control algorithms are implemented using the RWTH Mindstorms NXT Toolbox for MATLAB and these are run on a computer that connects to the robot through a USB cable. An overview of the proposed robot control process is illustrated in Fig. 1.

The remaining sections of this paper are organised as follows: Section II describes the proposed robot control system including its mechanical structure, the robot's functionalities and the algorithm's description. In Section III, the simulation and experimental results are provided and the proposed system performance is evaluated. Finally, Section IV summarises the paper and provides recommendations for future work.

II. System Description

This section describes the robot's mechanical structure, software functionalities, localisation algorithms and path planning algorithms.

A. Robot Design

In the initial phase, we aimed to design a robot which has a simple, light and easily controlled structure. The differential drive structure is appropriate for indoor robot navigation because it enables the robot to flexibly change direction in a confined place. Furthermore, only two motors are needed to drive the robot, allowing it to be light-weight. The third servo motor can rotate the ultrasonic sensor 360° around the motor axis, enabling the sensor to get adequate readings from the environment. The rotating sensor is installed in such a way that it is homocentric to the robot's centre of motion to simplify the geometrical complexity. Moreover, the position obtained from the particle filter can be directly used

as the robot motion centre without any further coordinate transformations. Finally, the height of the sensor must not be greater than the height of the wall. The robot hardware architecture is shown in Fig. 2.

B. Some Basic Robot Functionalities

Based on the existing NXT library, we developed a number of function modules that can be easily called in MATLAB for the robot's motion control and sensor operation. These basic functions are described as follows:

1) Move: This function has three main features:
- Move the robot forward by a given distance in the same direction it is facing.
- Force the robot to stop when it gets too close to the wall.
- Output a boolean value indicating whether there is a potential collision with the wall.

2) Turn: This function allows the robot to rotate itself around its centre of motion which is the midpoint between its two wheels.

3) Turnback: This function is always running when the robot is performing its tasks. If the distance between the wall and the robot is less than a given threshold, this function will force the robot to stop and then instruct it to move the robot backwards by a certain distance to maintain a safe distance from the wall.

4) Ultrascan: This function instructs the ultrasonic sensor to perform a quick 360° rotation around the servo axis to collect readings. During the rotation, the ultrasonic sensor collects a series of distances to the wall with respect to different angles. Then, the ultrasonic sensor rotates back to its original position and starts to detect potential objects in front of it.

C. Algorithm Description

In this section, we first analyse the characteristics of the ultrasonic sensor by comparing the real data and sensor readings. Then, a data processing method is proposed to handle the sensor data. Finally, we describe the particle filter localisation method and a sampling-based path planning method.

Fig. 3: LEGO ultrasonic sensor.

(a) Robot position in the map, yellow points are cross points of scanning lights and wall. The red line is the boundary of the map.

(b) The ultrasonic sensor scan, red lines are formed by simulated sensor readings, blue lines are formed by real sensor readings.

(c) Intersection points when the robot is at position (86, 85).

(d) Sensor readings when robot is at position (86, 85).

Fig. 4: Scannings of a typical ultrasonic sensor in the arena.

1) Characteristics of the Ultrasonic Sensor: Ultrasonic sensors are among the most popular sensors used for robot indoor localisation because they are cheap, lightweight, compact and have low energy consumption [18]. Consequently, this paper uses a LEGO ultrasonic sensor (Fig. 3) for the purpose of localisation and navigation tasks.

As an active sensor, the ultrasonic sensor emits acoustic waves that propagate through the environment and are then captured by the receiver. The distance travelled by the acoustic waves d, can be calculated as follows:

$$d = c * t/2 \qquad (1)$$

where c is the propagation velocity of the acoustic waves and t is the time taken for the acoustic waves to reach the receiver.

However, the ultrasonic sensor is not accurate enough to measure the time of flight t [13]. Another major limitation of the ultrasonic sensor is that the ultrasonic waves form a cone with a certain opening angle when propagating through the environment [19]. As a result, ultrasonic sensors cannot feedback accurate depth information for some orientations.

Before applying the ultrasonic scan data of the robot to perform particle scoring, we plotted it on a radar chart to verify its accuracy. Fig. 4 illustrates the inaccuracy of the readings when scanning a map with an ultrasonic sensor. As shown in Fig. 4 (a) and (b), there is a notable inconsistency between the scanning data of a simulated particle and real robot. The robot starts scanning at a position (20, 85) with an orientation of 0° in the map. The real sensor readings are distorted because of wave reflections from the wall and the dispersal of the wave cone. Furthermore, sensor reading errors occur due to echoes, attenuation and other uncertain factors. Consequently, the raw scanning data is not suitable to localise the robot directly.

2) Ultrasonic Sensor Data Processing: After further investigation, we found that the inaccuracy increased dramatically as the angle between the wave propagation and the wall increases (Fig. 4 (b) and (d)). Therefore, we take the minimum of the scanning data and turn our robot to that direction so that ultrasonic sensor will be approximately perpendicular to the wall. Then, we keep the scanning data within the range of ±15° at 0°, 90°, 180° and 270° positions so that we can ensure the accuracy while keeping enough

data for particle filter. As shown in Fig. 4 (d), the scanning data in the black dotted box can be used for localisation.

3) Particle Filter Localisation: For the localisation part, we utilise particle filter to probabilistically estimate the location of the robot in the map. The particle filter approach is widely applied to solve uncertainty problems based on probability theory and Bayesian theorem. It has a number of successful applications filed in robotics [20], [21], [22], [23], [24]. In this scenario, the uncertainty of the robot location and orientation (x, y, θ) is represented by particle distributions.

To localise a robot, we first fill the given map with randomly distributed particles where each particle potentially represents the real robot position and orientation. Then, the robot sensor readings and simulated particle reading are compared to calculate each particle's probability to be the true position. After ranking the particles according to their probability, we reallocate the particles around previously generated particles with higher weights. By doing this iteratively, the scope of the robot position can be gradually narrowed down.

Particle filter localisation is developed based on Bayesian theorem (Eq. 2) as follows:

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \qquad (2)$$

where, $P(A|B)$ represents the probability that the particle position is the robot position, given a particle reading B. $P(A)$ is the prior knowledge which equals to $P(A|B)$ in the

previous step. Furthermore, *P(B)* is an independent parameter which can be obtained from the particle scanning. Finally, *P(B|A)* represents the likelihood of the particle reading *B*, given a robot position *A*. Apparently, *P(B|A)* can not be obtained directly. However, it can be known by taking advantage of the actual sensor readings *R* instead of the unknown robot position *A* [25],

$$P(B|A) \propto P(B|R) \qquad (3)$$

Moreover, the ultrasonic sensor model is based on a normal distribution given by Gaussian function [25]. Applying Eq. 3, *P(B|A)* can be obtained as follows:

$$P(B|A) \propto P(B|R) \propto \frac{1}{\sigma\sqrt{2\pi}} e^{-(b-r)^2/2\sigma^2} \qquad (4)$$

4) Sampling-Based Path Planning: Once the robot position is probabilistically estimated, a collision free path needs to be planned. For this scenario, the turning error of the robot impacts on the final position error. Moreover, low energy consumption is favoured for mobile robot. Consequently, we utilise the sampling-based method to plan a path with minimum turning times and short distance. The proposed method is described in Algorithm 1.

Algorithm 1 Sampling based path planning method

1: calculate the configuration space: *C*
2: distribute *N* particles in *C* randomly
3: current path distance $l = \infty$
4: **while** $i \neq N$ **do**
5:     $i++$
6:     select *i* particles
7:     link start point, *i* particles and end point with straight lines
8:     **if** no collisions detected *and* path length $Pi<l$ **then**
9:         $l = Pi$
10:     **if** a path found with *i* particles **then**
11:         break
12: **end**

At first, a number of particles are distributed in the configuration space randomly. With this method, a collision free path can be guaranteed. Particles are selected according to the particle numbers and path length. The lesser the number of particles, the fewer the number of turning times for the robot.

As shown in Fig. 5, the robot starts at (50, 10) and the target position is (90, 90). To prevent collision, we included an offset which is half the mobile length inside the original map to produce a configuration space. Then, 30 intermediate waypoints are distributed in the configuration space randomly. A safe path consisting of several intermediate waypoints(usually 0 - 2 for this type of map) can be found. Finally, in this case, one intermediate point is selected out of these 30 particles according to the length of the whole path.

Fig. 5: Sampling based path planning.

Fig. 6: A comparison between A* path planning (blue) and sampling based path planning (red). Start position A (20, 20), target position B (80, 80). Stars in map represent the grids of A* path planning. We assume there is a particle C distributed on a grid. According to the particle distribution situation, other possible sampling based paths are shown in dotted green lines.

A* path planning is a popular robot path planning method. However, it causes localisation error in a gridded decomposed map. Fig. 5 illustrates the effectiveness of the sampling based path planning method . The sampling based method allows for a shorter path length and minimum turning times, enabling the mobile robot to move towards the desired position with minimal turning errors and running time.

However, when the robot is moving along the planned path, inaccurate robot position estimation can possibly cause robot to get very close to the arena wall, which may lead to a collision. To detect potential collisions, the ultrasonic sensor scanning is kept running at all times during the robot navigation process. If the distance between the robot and the wall is smaller than a given threshold, the robot will be instructed to move backwards and perform re-localisation and re-navigation.

TABLE I: Localisation and navigation simulation results.

Map	Completion time (s)	Distance from target (cm)	Path length (cm)	Detected collision
Map1	15.22	1.79	116.9	0
Map2	16.69	1.83	120.1	0
Map3	23.48	2.11	276.2	0

(a) Map 1. (b) Map 2. (c) Map 3.

Fig. 7: Three different maps for simulation.

III. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of the proposed robot localisation and path planning algorithms both in terms of simulations and real arena setup. Experiments are carried out on a desktop computer having a Core i7 6700 CPU. The desktop PC and the robot controller NXT communicate with each other through a USB cable. The robot control algorithms run on the BotSim Robot simulator which was developed by the University of Bristol.

A. Simulation Results

To evaluate the effectiveness of the proposed algorithms, we simulate the robot localisation and navigation algorithms in three different maps as shown in Fig. 7. For this simulation, sensor scanning and robot motion noises are taken into consideration. The ultrasonic scanning error is set to 1 cm, while the robot motion error is 0.1 cm and the robot turning error is 0.005 radian. These parameters are set according to the actual performance of the sensor and robot in the environment. Tab. I shows the average test results for each map. Both the robot running time and target accuracy are recorded by the robot simulator. The final target accuracy is between 1 cm and 3 cm for a completion time between 15 and 24 seconds. Furthermore, no collision occurred, indicating the robustness and effectiveness of the proposed sampling based path planning method.

B. Experimental Results in the Arena

We conduct further real experiments in the arena (Map 1) to demonstrate the feasibility of our approaches. A detailed flowchart of the proposed robot control system is shown in Fig. 8. A desktop PC running the MATLAB robot simulator is utilised to perform calculation and send instructions to the robot. At first, the robot starting position is randomly set. Then, the ultrasonic sensor is driven to scan the inside of the arena for 360°. An ultrasonic reading is recorded for each unit angle. The estimation of robot position and orientation is obtained using the processed sensor data, which usually takes several seconds. Based on the estimated robot position and orientation, a collision free path is planned by linking

Fig. 8: Robot operation process.

several waypoints which are selected from the randomly-distributed intermediate particles. The mobile robot moves to each intermediate point in sequence until it reaches its final destination. A wrong position estimation would most possibly cause collisions. In this case, localisation and path planning need be performed again. Finally, if the robot reaches its destination, it would carry out re-localisation to ensure it is as close as possible to the correct target position.

The arena is made up of cardboard and it has the same dimensions as Map 1 in the simulator. As observed in Fig. 9, the simulator calculates the start position based on the sensor readings and then selects the best intermediate points for the mobile robot. The vehicle successfully navigates through the map without collision and reaches the final target according to the simulator instructions. Fig. 9 shows the simulator process and the robot navigation process. The robot motion process follows the simulator's instructions precisely. Therefore, the effectiveness and the performance of the robot localisation and path planning system is successfully demonstrated by the experimental results.

Fig. 9: Robot localisation and navigation in the arena and the simulator.

IV. CONCLUSION AND FUTURE WORK

In this paper, we presented a mobile robot localisation and navigation system using the LEGO NXT in a map. The robot navigation task, which is carried out with only one ultrasonic sensor, has been proved to be feasible. Furthermore, an effective method has been proposed to deal with noisy data obtained from the ultrasonic sensor. The robot's position and orientation have been estimated using the particle filter algorithm. In addition, a sampling based robot path was planned with minimum turning times and short path distance. Simulation results show that for the simulation in Map 1, a final target accuracy of 1.79 cm was obtained for a completion time of 15.22 seconds. Finally, the accuracy of our algorithms was backed by tests in a real arena.

One of the downsides of our work is that we could not effectively guarantee the robustness of the system in every real map. In future contributions, we aim to improve the system adaptability for different types of maps.

REFERENCES

[1] C. Wang, L. Meng, S. She, I. M. Mitchell, T. Li, F. Tung, W. Wan, M. Q. H. Meng, and C. W. de Silva, "Autonomous Mobile Robot Navigation in Uneven and Unstructured Indoor Environments," 2017.

[2] G. Q. Huang, M. Z. Chen, and J. Pan, "Robotics in ecommerce logistics," *HKIE Transactions*, vol. 22, no. 2, pp. 68–77, 2015.

[3] H. Lipson and M. Kurman, *Driverless: intelligent cars and the road ahead*. Mit Press, 2016.

[4] J.-S. Gutmann, K. Culp, M. E. Munich, and P. Pirjanian, "The social impact of a systematic floor cleaner," in *Advanced Robotics and its Social Impacts (ARSO), 2012 IEEE Workshop on*, pp. 50–53, IEEE, 2012.

[5] P. Ranganathan, R. Schultz, and M. Mardani, "Use of lego nxt mindstorms brick in engineering education," in *Proceedings of the 2008 ASEE North Midwest Sectional Conference*, pp. 17–19, 2008.

[6] T. Karp, R. Gale, L. A. Lowe, V. Medina, and E. Beutlich, "Generation NXT: Building young engineers with LEGOs," *IEEE Transactions on Education*, vol. 53, no. 1, pp. 80–87, 2010.

[7] W. Grega and A. Pilat, "Real-time control teaching using LEGO® MINDSTORMS® NXT robot," *2008 International Multiconference on Computer Science and Information Technology*, pp. 625–628, 2008.

[8] M. Pinto, A. P. Moreira, and A. Matos, "Localization of mobile robots using an extended kalman filter in a lEGO NXT," *IEEE Transactions on Education*, vol. 55, no. 1, pp. 135–144, 2012.

[9] J. F. M. Lee and J. A. Buitrago, "Map generation and localization for a LEGO NXT robot," *2015 IEEE 2nd Colombian Conference on Automatic Control, CCAC 2015 - Conference Proceedings*, pp. 6–10, 2015.

[10] K. Chawla and G. Robins, "An RFID-based object localisation framework," *International Journal of Radio Frequency Identification Technology and Applications*, vol. 3, no. 1/2, p. 2, 2011.

[11] L. Moreno, J. J. M. Armingol, S. Garrido, A. de la Escalera, and M. M. a. Salichs, "A Genetic Algorithm for Mobile Robot Localization Using Ultrasonic Sensors," *Journal of Intelligent & Robotic Systems*, vol. 34, no. 2, pp. 135–154, 2002.

[12] Y. Lee, B. M. Namee, and J. Kelleher, "Navigating the Corridors of Power : Using RFID and Compass Sensors for Robot Localisation and Navigation," 2010.

[13] J. J. Garc, M. Mazo, and D. Hernanz, "Ultrasonic Ranging Sensor using Simultaneous," vol. 51, no. 12, pp. 1660–1670, 2004.

[14] F. Gustafsson, "Applications Practice with Positioning Particle Filter Theory and," *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, no. 7, pp. 528–82, 2010.

[15] H. Nurminen, A. Ristimäki, S. Ali-Löytty, and R. Piché, "Particle filter and smoother for indoor localization," *2013 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2013*, no. October, pp. 28–31, 2013.

[16] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[17] M. Ganeshmurthy and G. Suresh, "Path planning algorithm for autonomous mobile robot in dynamic environment," *2015 3rd International Conference on Signal Processing, Communication and Networking (ICSCN)*, pp. 1–6, 201502.

[18] H. Zhao and Z. Wang, "Motion measurement using inertial sensors, ultrasonic sensors, and magnetometers with extended kalman filter for data fusion," *IEEE Sensors Journal*, vol. 12, no. 5, pp. 943–953, 2012.

[19] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots*. MIT press, 2011.

[20] R. Fan, Y. Liu, X. Yang, M. J. Bocus, N. Dahnoun, and S. Tancock, "Real-time stereo vision for road surface 3-d reconstruction," *arXiv preprint arXiv:1807.07433*, 2018.

[21] R. Fan, X. Ai, and N. Dahnoun, "Road surface 3d reconstruction based on dense subpixel disparity map estimation," vol. 27, pp. 3025–3035, 2018.

[22] R. Fan, V. Prokhorov, and N. Dahnoun, "Faster-than-real-time linear lane detection implementation using soc dsp tms320c6678," 2016.

[23] U. Ozgunalp, R. Fan, X. Ai, and N. Dahnoun, "Multiple lane detection algorithm based on novel dense vanishing point estimation," vol. 18, pp. 621–632, 2017.

[24] R. Fan and N. Dahnoun, "Real-time stereo vision-based lane detection system," vol. 29, p. 074005, 2018.

[25] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.