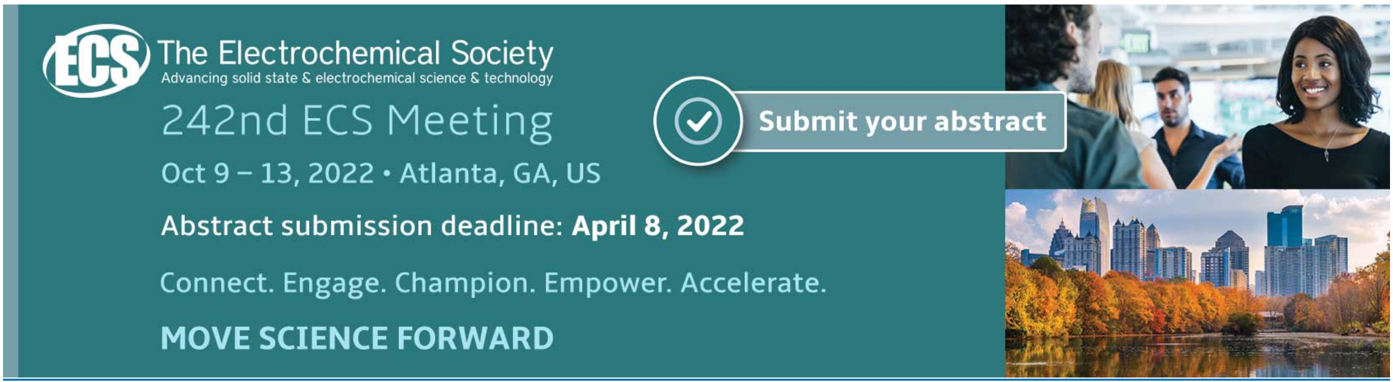
IOP Conference Series: Materials Science and Engineering

PAPER • OPEN ACCESS

Autonomous navigation of a mobile robot in dynamic indoor environments using SLAM and reinforcement learning

To cite this article: C C E Chewu and V Manoj Kumar 2018 *IOP Conf. Ser.: Mater. Sci. Eng.* **402** 012022



Autonomous navigation of a mobile robot in dynamic indoor environments using SLAM and reinforcement learning

C C E Chewu[1]and V Manoj Kumar[1]*

Department of Mechanical Engineering, SRM Institute of Science and Technology, Kattankulathur

*Corresponding Author: vmanojkumar19@gmail.com

Abstract. The recent advances in robotics has resulted in a more convenient use of mobile robots in alications such as assisting the disabled, deliveries and domestic purposes. The main challenge faced by mobile robots is navigation in a dynamic environment, which is path planning for dynamic obstacle avoidance. This paper proposes a novelty method for solving the path planning problem for mobile robots posed by dynamic obstacles based on SLAM (Simultaneous Localisation and Maing) algorithm and Reinforcement Learning. The algorithms implemented relied on the Kinect sensor for maing and rotary encoder for localisation of the robot in the map. The SLAM algorithm implemented resulted in a mean error metric of 4.07%. The modified Q-learning algorithm implemented in this paper allowed the mobile robot to avoid dynamic obstacles by re-planning the path to find another optimal path different from the previously set global optimal path. From the investigation, it was shown that it is possible for a robot to navigate in a dynamic using the Reinforcement Learning technique.

1. Introduction

The ability of mobile robots to autonomously navigate in an environment static or dynamic, has initiated a number of inventions which are very crucial to a number of different sectors. To aid the mobile robot to successfully maneuver through a number of a static and dynamic obstacles to its goal, it needs to have a good perception of the environment. In this paper the Microsoft Kinect v2 was used to create a 3d depth map of the environment by utilizing its point cloud data output. Point cloud enables the robot to determine how far it is from obstacles which also govern the robot's trajectory planning process. SLAM algorithms enable the robot to continuously update its location and map as it navigates to the goal. [1] [2]
The navigation of a mobile robot relies on its perception of the environment. In a recent study focusing on the comparison of cameras for robot perception in which Laser camera, Roland Picza LPX-600, and Microsoft Kinect v2 output was analyzed, the results from the 2 sensors in-terms of accuracy, did not differ much when compared with the actual ground data [3]. The Microsoft Kinect v2 was chosen over Laser camera because of its high accuracy over short distances and also being available in the market cheaply (less than USD200) when compared to the laser camera which is at least 10 times more expensive. [4] [5]

The Kinect camera is mainly useful for Local navigation that is in detection of static and dynamic obstacles and also in the global navigation of the robot which is dependent on the accuracy of the 3D map created using the Kinect. Several algorithms have been recently evaluated by researchers but of interest the Iterative Cost Point (ICP) algorithm used in the map registration process of map creation for alignment of consecutive point cloud frames from the Kinect [6] [7]. This algorithm has high accuracy which is of importance in the localization of the robot as it navigates in the environment. For even much more higher accuracy, the Random Sample Consensus (RANSAC) algorithm was also suggested to be implemented in the registration process, acting as an initial or pre- processing step before the implementation of the ICP algorithm [8] [9] [10].

The Adaptive Monte-Carlo Localization (AMCL) particle filter algorithm which is based on Bayesian probability, allows for a more accurate navigation (localization) of the robot in its environment when co pared to the traditional Monte-Carlo Localization algorithm. The AMCL algorithm was adopted for use in this investigation. [11] [12]

Due to the increasing number of uses of mobile robots, many researchers have now directed their attention to dynamic obstacles rather than the static obstacles only. The popular A* and D* algorithms cannot adapt to the rapid changing parameters of a dynamic environment. Reinforcement Learning performs better and adapt to the changes in the environment [13]. The Q-Learning (QL) algorithm, which is one of the most popular Reinforcement algorithms was able to create a policy to model the environment with no initial input. This investigation is mainly focusing on adapting the modified QL to establish policy that can be used for an environment characterized by dynamic obstacles [14] [15]. The Regime Switching Partial Observable Markov Decision Process (RSPOMDP) based QL algorithm was adopted in this investigation for the navigation of the robot in dynamic environment. This is the first time this algorithm has been implemented for such an alication. To speed up the process of the QL algorithm in obtaining an optimal policy, a greedy search algorithm used in conjunction with QL algorithm has also been found to give satisfactory results in this process. [16] [17]

2. Methodology

In this chapter, the milestones involved in this investigation have been outlined in order of execution. This involves the 3 major categories for SLAM, Reinforcement Learning and evaluation of these algorithms.

The following flowchart shows the order of processes involved in the formulation of this investigation.

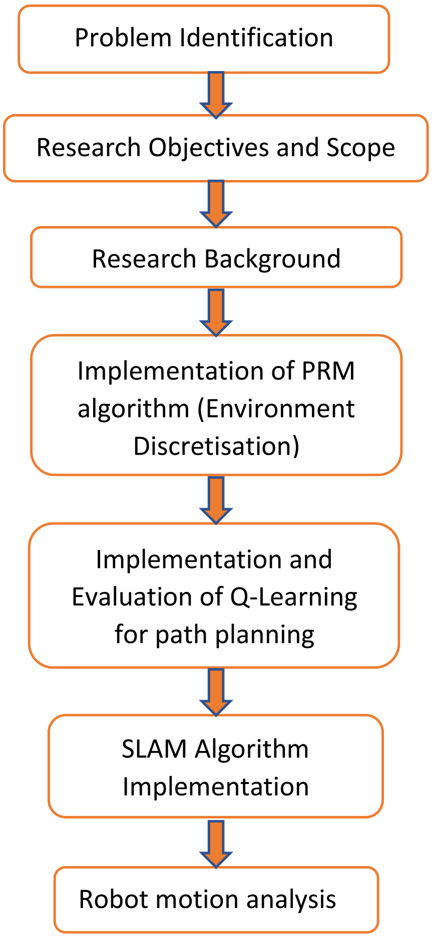

Problem Identification
Research Objectives and Scope
Research Background
Implementation of PRM algorithm (Environment Discretisation)
Implementation and Evaluation of Q-Learning for path planning
SLAM Algorithm Implementation
Robot motion analysis

Figure. 1 Flowchart showing methodology

2.1 Slam

This is the major step which will enable the robot to accurately navigate in its environment. The complication posed by the SLAM technique is which process to start with maing or localisation since these

2 processes depend on each other. This problem is normally referred to as the egg-chicken problem, when 2 interdependent processes have to be started simultaneously. The Kinect sensor captures the visual information of the environment in frames which are converted to point cloud data which involves the depth - colour (RGB-D) association of every point in each frame. The alignment of consecutive frames is done using the ICP algorithm. The process of map creation juxtaposed with odometry data enable to create an accurate 3-D map for robot navigation [10].

The Extended Kalmann Filter algorithm has been suggested to account for the cumulative errors in the rotary encoder sensor.

The Adaptive Monte Carlo Localisation (AMCL) is an important algorithm for accurate robot localization which is based Bayesian probability. AMCL makes use of weighted samples and Bayesian Particle Filter to predict the pose of the mobile robot from the encoder output. Initially, with no previously known information of the mobile robot pose, it is assumed to be a uniform random distribution over the entire environment. The estimates or assumed states of the mobile robot as it navigates are shifted and resampled in accordance to the information from the rotary encoders. [12]

2.2. Reinforcement learning for dynamic obstacle avoidance

Hierarchical Partially Observable Markov Decision Process based Reinforcement Learning (modified Q-Learning) has been suggested for robot's Global and Local motion planning. This algorithm is not only for obtaining time-distance optimal paths but is also useful in predicting which paths can be avoided by the robot due to blockages.

Long-term and short-term motion prediction is also possible by calculating the tangent vector of the obstacle trajectory hence aiding the robot path planning process for dynamic obstacle avoidance. Kalman Filter has also been integrated with this algorithm for predicting motion of a previously identified dynamic obstacle. [13]

3. Implementation of the system

The algorithms used in this investigation were implemented using the Robot Operating System (ROS) which is a popular framework for developing robot software. R.O.S. allow researchers to easily evaluate their algorithms and publish them to the robotics community for further improvement. Perhaps the most important advantages of R.O.S. is that it enables researchers not to reinvent the wheel by the reuse of previously designed algorithms. Open AI-gym software was used to train the Q-Learning algorithm used in this investigation and test for its convergence. an environment similar to the real environment where the robot is to operate was created in Open AI-gym software and a turtlebot was allowed to learn the environment basing on the modified Q-Learning algorithm used. The results of simulation of the modified Q-Learning algorithm used are shown in Figure 5.

The system implemented mainly comprises of 4 components namely: i) Global Navigation ii) Local Navigation iii) Path planning iv) Motion execution.

Global Navigation: this whereby the identifies its obstacles and its position relative to the previously constructed map.

Local Navigation: the robot detects and identifies obstacles in relative to its current position. In this paper, local navigation is done with the use of the depth sensor, the Kinect camera.

Path planning: this involves the finding the best way possible for navigation to avoid static and dynamic obstacles. Global path plan is required to first find the best path that allows the robot to reach the goal avoiding static object. Path re-planning is crucial to determine a different optimal path when a dynamic obstacle id detected.

Motion execution refers to actuation of the dc motors that drives the mobile robot to accurately move the robot in its environment. This involves determining correct input current to the DC motors to give the mobile robot a proper acceleration and velocity.

Reinforcement is used in this section using the Q-Learning algorithm for dynamic obstacles avoidance.

The modified Q-Learning algorithm used in this investigation can be mathematically represented as:

$$Q'_{\Psi_k}(s_t, a_t) = Q'_{\Psi_k}(s_t, a_t,) + \gamma_{\Psi_{k,t}}[\, r_{\Psi_k}(s_t, a_t) + \alpha_{\Psi_k} \max Q'_{\Psi_k}(s_{t+1}, a_{t+1}) - Q'_{\Psi_k}(s_t, a_t)] \quad (1)$$

The optimal action policy for the algorithm is given as:

$$\pi^{*}_{\Psi_k}(s) = (argmax\ Q^{*}_{\Psi_k}(s, a)) \quad (2)$$

Where *s* is the observed environment state, *a* is action to be taken, *r* is the reward factor, γ is the discount factor, *s*' is the predicted next state of the robot and π is the action policy

The Q value for this algorithm is obtained by taking into account the cost function which is expressed as follows:

$$r_{\Psi_{k,\ t}}(s_t, a_t) = wf(\, s_t, s_{t+1}) + (1 - w)g(s_t) \quad (3)$$

Where *w* is a weight adjusted to balance the equation, *w* was set to 10/100, $f(\, s_t, s_{t+1})$ is a Euclidean distance function between 2 nodes in the robot environment based on the Probabilistic Road Map algorithm used in discretizing the environment, $g(s_t)$ is the reward function based on dynamic obstacles.

The dynamic obstacle reward function was set as follows:

$$g(s_t) = \begin{cases} R = 10; & when\ the\ robot\ reaches\ the\ goal \\ R = -10; & when\ the\ robot\ touches\ a\ dynamic\ obstacle \\ R = 0; & in\ any\ other\ situation \end{cases} \quad (4)$$

The cost function in equation (3) allows the Q-Learning algorithm to cater for dynamic obstacle avoidance as it is based on the shortest path and dynamic obstacle reward function. When a dynamic obstacle is detected to block the current optimal path, a new regime or step setting is initiated and the path planner is able to quickly choose another optimal path, using the previous experience about the map, as determined by the Q function value.

A diagram showing the interconnection of the software modules of the navigation processes in this investigation is shown in Figure 2 below:

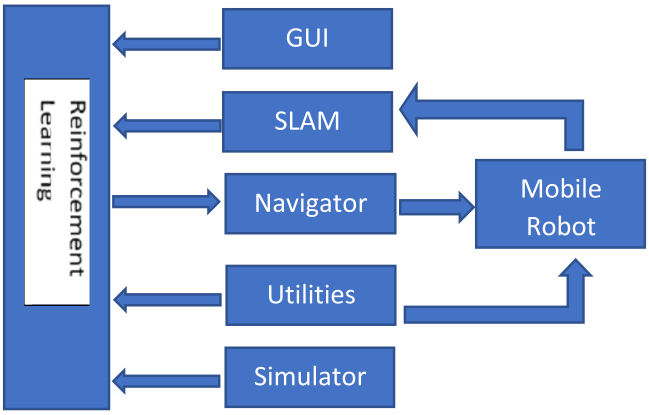
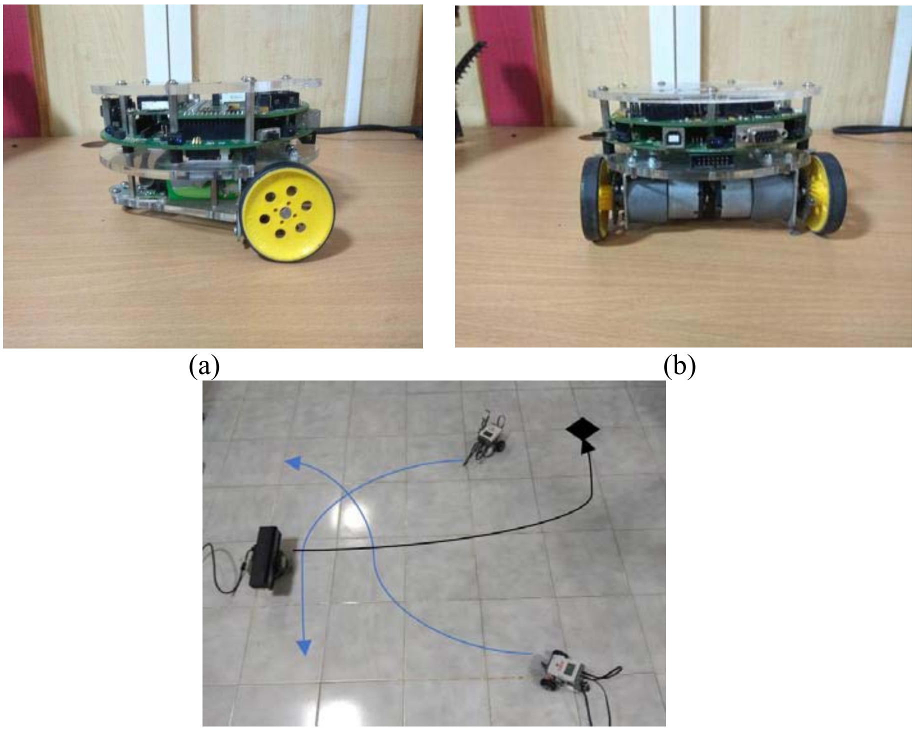


Figure 2. System Software Configuration


(a)

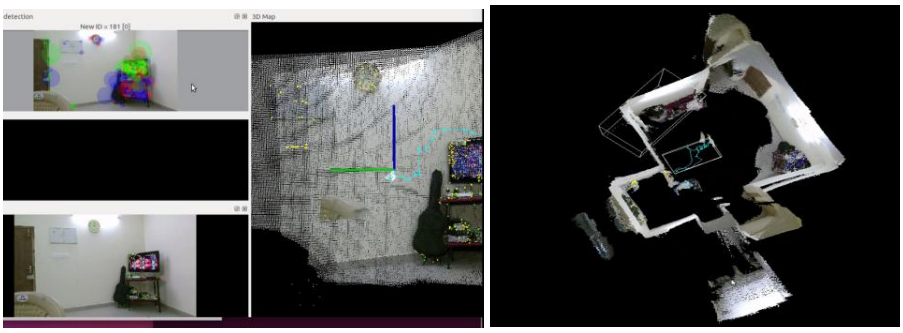
(b)

(c)

Figure 3. (a) Side view (b) Back view of the robot (c) experimental set-up of robot with 2 moving Lego-mind robots

4. Experimental results and discussion

In this section, the results of the investigation have been presented and analysed. The results range from Map building to path planning for dynamic obstacle avoidance algorithms. Figure 4 below shows the results of the map created using Real Time Aearance Based Maing algorithm (RTABMap).



(a) (b)

Figure 4. (a) Map building process using RTABMap algorithm (b) Map result using RTABMap algorithm

To evaluate the accuracy of the generated map, the map is first binarised and aligned respectively to the ground truth map. The distance between the cells of the ground truth map and the generated is obtained using the K-nearest neighbor (K-nn) algorithm. The summation of all these distances is then divided by the number of occupied cells in the ground truth map. This mean error metric was found to be **4.07%.**

The results from Open AI-gym software for evaluation of the implemented Q-Learning algorithm have been shown in Figure 5 below, and they show that the algorithm was able to converge after aroximately 2000 episodes.

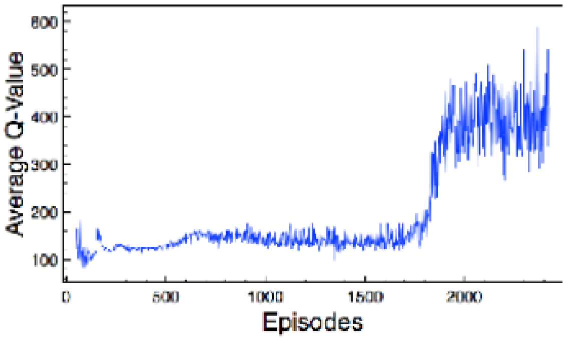


Figure 5. Q-value vs episode graph of implemented algorithm

5. Conclusion and Recommendation

In this paper, a Reinforcement Learning algorithm was proposed for the autonomous robot navigation problem. The Q-Learning algorithm used in this investigation was able to learn the dynamic environment and upon implementation the robot was able to avoid 2 dynamic Lego-mind robots by re-planning and choosing another optimal path based on the past experience of the robot learning the environment. The learning experience of the robot also got better with more experience acquired by the robot showing the ability of this algorithm to be used for robot dynamic obstacle avoidance.

Predictive navigation framework, for long-term and short-term trajectory prediction of the robot has been suggested for further improvement of the autonomous navigation of a robot in dynamic environments for future research. This framework can have the capacity to increase robot learning and avoid these dynamic obstacles more effectively.

6. References

[1] Javier H A, Arnay R, Jonay T and Acosta L O 2016, "Using Kinect on an Autonomous Vehicle for Outdoors Obstacle Detection" IEEE Sensors J. **16** 3603-10

[2] Endres F, Hess J, Sturm J, Cremers D, and Burgard W 2014 3-D maing with an RGB-D camera *IEEE Trans. on Robotics* **30** 177-87

[3] Reinaldo J, Maia R, Souza A 2015 Adaptive navigation for mobile robots with object recognition and ontologies, *Brazilian Conf. on Intelligent Systems* (Brazil: IEEE) 210-15

[4] Lv Q, Lin H, Wang G, Wei H, Wang Y 2017 ORB-SLAM based tracing and 3d reconstruction for robot using Kinect 2.0, *Proc. of the29th Chinese Control And Decision Conf.* (China: IEEE) 3319-23

[5] Jing X, Xiao-liang J, Yin Y and Ding L 2016 visual navigation for mobile robot with Kinect camera in dynamic environment, *Proc. of the 35th Chinese Control Conf.* (China: IEEE*)* 4757-64

[6] Feng X and Zhang Y 2015 An embedded visual SLAM algorithm based on Kinect and ORB feature algorithm*, Proc. of the 34th Chinese Control Conf.* (China: IEEE) 6026-30

[7] Mardiyanto R, Anggoro J and Budiman F 2015 2-D map creator for robot navigation by utilizing kinect and rotary encoder, *Int. Seminar on Intelligent Technology and Its Alications* (Indonesia: IEEE) 81-84

[8] Altuntas N, Imal E, Emanet N, Nur C and Ozturk C N 2016 Reinforcement learning-based mobile robot navigation, *Turkish J. of Electrical Engineering & Computer Sciences* **24** 1747-67

[9] Tanaka R, Kurabe K, Kihal M E, Ichimi M, Tatsuno K 2015 Improvement on an obstacle avoidance in telepresence robot, *Int. Symp. on System Integration* (Japan: IEEE/SICE) 634-39,

[10] Valdinei F, Costa A H R 2009 ," Compulsory Flow Q-Learning: an RL algorithm for robot navigation based on partial-policy and macro-states",J. Braz. Comp. Soc. **15** 65-75

[11] Zhang Y**,** Wang S**,** Yang X and Ma Y 2016 Robotic vehicle navigation based on image processing using Kinec*t, Int. Conf. on Smart City and Systems Engineering* (China: IEEE) 18-22

[12] Costa D S E and Gouvea M M 2010 *9th Int. Conf. on Machine Learning and Applications* (Brazil: IEEE Computer Society) 37-42

[13] Jaradat M A K, Al-Rousan M and Quadan L 2010 Reinforcement based mobile robot navigation in dynamic environment, *Elsevier Robotics and Computer-Integrated Manufacturing* **27** 135–49

[14] Khriji L, Touati F, Benhmed F and Al-Yahmedi A 2011 *Int. J. of Advanced Robotic Systems* **8** 45-51

[15] Foka A F and Trahanias P E 2010 Probabilistic autonomous robot navigation in dynamic environments with human motion prediction, *Int J. Soc. Robot* **01** 79–94

[16] Voisan E, Paulis B, Precup R E and Dragan F 2015 ROS-based robot navigation and human interaction in indoor environment, *IEEE 10th Jubilee Int. Symp. on Alied Computational Intelligence and Informatics* 31-36

[17] Cruz D and Yu W 2016 Path planning of multi-agent systems in unknown environment with neural kernel smoothing and reinforcement learning, *Elsevier Journal of Neurocomputing,* **233** 34-42