An Extended Navigation Framework for Autonomous Mobile Robot in Dynamic Environments using Reinforcement Learning Algorithm

Nguyen Van Dinh[1], Nguyen Hong Viet[1], Lan Anh Nguyen[1], Hong Toan Dinh[1], Nguyen Tran Hiep[1], Pham Trung Dung[1], Trung-Dung Ngo[2] and Xuan-Tung Truong[1]

Abstract— **In this paper, we propose an extended navigation framework for autonomous mobile robots in dynamic environments using a reinforcement learning algorithm. The main idea of the proposed algorithm is to provide the mobile robots the relative position and motion of the surrounding objects to the robots, and the safety constraints such as minimum distance from the robots to the obstacles, and a learning model. We then distribute the mobile robots into a dynamic environment. The mobile robots will automatically learn to adapt to the environment by their own experienced through the trial-and-error interaction with the surrounding environment. When the learning phase is completed, the mobile robots equipped with our proposed framework are able to navigate autonomously and safely in the dynamic environment. The simulation results in a simulated environment shows that, our proposed navigation framework is capable of driving the mobile robots to avoid dynamic obstacles and catch up dynamic targets, providing the safety for the surrounding objects and the mobile robots.**

Keywords—Mobile robot navigation system, autonomous mobile robot, reinforcement learning, q-learning.

I. INTRODUCTION

The ability to autonomously navigate in dynamic environments, such as urban and terrain environments, museums, airports, offices and homes, and shopping malls, is crucial for mobile robots. If we wish to deploy the autonomous mobile robots in such environments, the first and most important issue is that, the mobile robots must avoid obstacles in their vicinity, while navigating safely towards a given goal. In order to archive that, several mobile robot navigation systems have been proposed in the recent years [1], [2], [3], [4].

The conventional mobile robot navigation frameworks can be roughly classified into two categories according to the techniques utilized to develop the motion planning systems: (i) model-based methods and (ii) learning-based approaches. In the first category, the navigation systems utilize available models, such as artificial potential field, vector field histogram, dynamic window approach, velocity obstacles, randomized kinodynamic planning, inevitable collision states, reciprocal velocity obstacles techniques to develop the motion planning system. In the second category of the methods, the machine learning techniques, such as inverse reinforcement learning, are used to enable the mobile robots to navigate autonomously in dynamic environments.

*This work was not supported by any organization

[1]Nguyen Van Dinh, Nguyen Hong Viet, Lan Anh Nguyen, Hong Toan Dinh, Nguyen Tran Hiep, Pham Trung Dung and Xuan-Tung Truong are with the Faculty of Control Engineering, Le Quy Don Technical University, Vietnam xuantung.truong@gmail.com

[2]Trung-Dung Ngo is with the School of Sustainable Design Engineering, University of Prince Edward Island, Canada

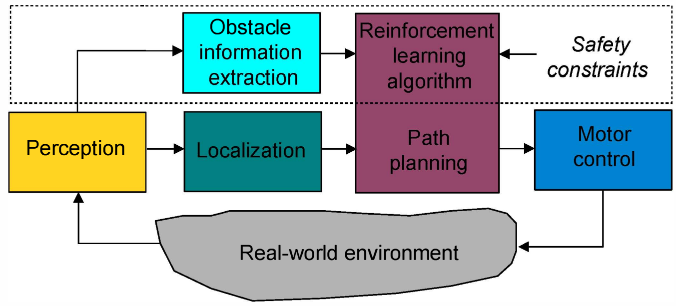

Fig. 1. An extended navigation framework for autonomous mobile robots in dynamic environments using a learning approach.

Although the model-based approaches [5], [6], [7], [8], [9] have been evaluated such that, the navigation systems are capable of driving the mobile robots to navigate autonomously and safely towards a given goal, they still suffer essential weaknesses that seriously hinder the robot capabilities to navigate in dynamic environments. For example, in these papers, the authors have to hand-craft all the features of the surrounding environments and then incorporate them into the robot navigation system. In addition, several parameters are empirical set by the authors experiences for a specific environment. Moreover, this parameter set often need to be tuned individually, and can vary significantly for different environments.

In order to overcome the aforementioned drawbacks, recently, a few machine learning techniques-based navigation systems have been proposed to enable the mobile robots to navigate autonomously and safely in dynamic environments [10], [11], [12]. In these papers, the authors utilized the inverse reinforcement learning technique to develop the navigation systems of the mobile robot. Using the inverse reinforcement learning-based autonomous mobile robot navigation systems, the mobile robots are taught using the demonstrations of the human experts. The mobile robots will learn and get the policy of each action they have learnt. The trained mobile robots then can work in the same situation. Although these methods have been enabled the mobile robots to dealt with dynamic environments, and the mobile robots can learn to adapt to the surrounding environments. However, the environments are very dynamic, unknown, clustered, and unstructured. Therefore, the authors cannot teach the mobile robots by hand in each single situations.

To overcome the above-mentioned drawbacks, we propose an extended navigation framework for autonomous mobile robots in dynamic environments using a reinforcement learning algorithm [13]. Because, the reinforcement learning is an useful way for robotics to learn control policies. In addition, the main advantage of the reinforcement learning is the completed independence from human labeling. In other words, using the reinforcement learning algorithms, the mobile robots can learn without expert supervision. Furthermore, reinforcement learning is an online learning algorithm. Thus it offers to robotics a framework, which enables a mobile robot to autonomously discover an optimal action through trial-and-error interactions with surrounding environment [14].

The remainder of the paper is organized as follows. Section II presents the proposed navigation framework for autonomous mobile robots in dynamic environments. Section III provide the simulation results of the proposed model in a simulated environment. We conclude this paper in Section IV.

II. The Proposed Framework

A. Extended Mobile Robot Navigation Framework

Our primary objective is to develop a navigation system that enable a mobile robot to navigate safely and autonomously in dynamic environments. To achieve that goal, in this paper, we propose an extended mobile robot navigation system, as shown in Fig. 1. The extended navigation system is develop based on the conventional navigation scheme introduced by Siegwart et al. [2], and consists of two major parts: (i) the conventional navigation scheme, and (ii) the extended part (in the dash line box). In the first part, the conventional navigation scheme is based on the composition of four typical functional blocks: perception, localization, motion planning, and motor control. In the extended part, the navigation framework aims at extracting the obstacles information, including their position and the motion in the robots vicinity. These obstacles information, and the safety constraints such as the minimum distance from the mobile robot to the surrounding obstacles, are then used as inputs of the q-learning algorithm. The output of this model is the optimal action of the mobile robot. This optimal action is then combined with a path planning technique to allow the mobile robot to navigate autonomously and safely in the dynamic environments.

B. Q-Learning-Based Mobile Robot Navigation Framework

Reinforcement learning [13], is a type of machine learning technique, which allows agents and machines to automatically determine an optimal behavior within a specific context, to maximize their performance. Figure 2 shows a typical framework of the reinforcement learning algorithm, in which at each step the agent executes an action, receives an observation (new sate), and receives a reward; while the environment receives an action, emits an observation (new sate), and emits a reward.

Reinforcement learning is a useful way for robotics to learn control policies [14]. The main advantages of the reinforcement learning algorithms are the completed independence from human labeling and the potential of automating the design of data representations. Therefore, reinforcement learning abstracted considerable attentions recently. Conventional reinforcement learning methods are normally utilized to improve the controller performances in path-planning of robot-arms [15] and controlling of helicopters [16], but they are rarely applied to mobile robot.

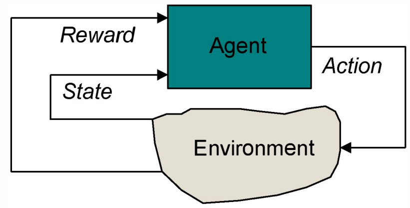

Fig. 2. A typical framework of the reinforcement learning algorithm. An agent takes an action in an environment. The agent then transits to a new state and gets a reward, which are feedbacks into the agent.

$$Q(s,a) = r + \gamma \max_{a'} Q(s',a') \tag{1}$$

A typical model of the reinforcement learning techniques is the q-learning algorithm [17], which is a model-free reinforcement learning technique. The main idea of q-learning algorithm is that, we can iteratively approximate the q-function using the Bellman equation Eq. 1. In the simplest case, the q-function is implemented as a table, with states as columns and actions as rows. A detailed description of the q-learning algorithm was given in [17].

Algorithm 1: Q-learning-based navigation framework for autonomous mobile robot in dynamic environments

input : Laser data $X = (x_1, x_2, \ldots, x_N)$, set of possible actions A, learning rate α, discount factor γ, epsilon-greedy policy ε, robots pose, safety constraints

output: $Q(s,a)$, state $s \in S$, action $a \in A$

```
begin
    Initialize Q(s,a) arbitrarily for s ∈ S, a ∈ A
    for (each episode):
        Randomly set the robots pose in the scenario
        Observe initial state of the robot s
        while (s is not a terminal state):
            Select an action a
                With probability ε select a random action a
                Otherwise select a = max_a' Q(s,a')
            Take action a, observe s', compute reward r
            Q(s,a) += α (r + γ max_a' Q(s',a') - Q(s,a))
            s = s'
        end while
    end for
end
```

In this paper, we utilize q-learning algorithm to develop a navigation framework for autonomous mobile robot in dynamic environments, as presented in Algorithm 1. The inputs of the algorithm are the laser data $X = (x_1, x_2, ..., x_N)$, robots pose (position, orientation and motion of the mobile robot), learning rate α, discount factor γ, epsilon-greedy policy ε, and safety constraints. The output of the algorithm is the $Q(s,a)$ value, which allows the mobile robot to select an optimal action based on the current states of the robot. The possible actions of the mobile robot consist of move forward, left, right, or backward. The safety constraint is the safety distance from the robot to the obstacles, and are used to compute the reward of the q-learning algorithm.

In Algorithm 1, the mobile robot learns a q-function that can be used to determine an optimal action. To accomplish that, there are two things that are useful for the mobile robot to do: (i) *exploit* the knowledge that it has found for the current state s by doing one of the actions a that maximizes $Q(s',a)$ (line 9 of the Algorithm 1); (ii) *explore* in order to build a better estimate of the optimal q-function. That is, it should select a different action from the one that it currently thinks is the best (line 8 of the Algorithm 1). There have been a number of suggested ways to trade off between the exploration and exploitation. In this paper, the $epsilon-greedy$ policy ε is used to select the greedy action (one that maximizes $Q(s,a')$), where $0 < \varepsilon < 1$. It is possible to change ε value through time. Intuitively, early in the life of the robot it should select a more random strategy to encourage initial exploration and, as time progresses, it should act more greedily.

When the training phase is completed, we save the trained model as a brain of the mobile robot. Then, whenever we distribute the trained mobile robot in a dynamic environment, it will load this trained model and utilize it to choose the appropriate actions to navigate autonomously and safely in the environment.

III. Simulation

In order to demonstrate the effectiveness of the proposed mobile robot navigation framework, in this section, we conduct experiments in a simulated environment.

A. Simulation Setup

To narrow the gap between the simulated and real-world environments we have created a simulated office-like scenario with walls, dynamic obstacles (red circles), dynamic targets (blue circles), and a mobile robot for testing the proposed navigation framework, as shown in Fig. 3. The mobile robot is requested to catch up dynamic targets while avoiding dynamic obstacles.

In this paper, we chose a mobile robot model, which can only do four possible actions, including move to the left, right, forward or backward. The obstacles and targets bounce around. The mobile robot is equipped with a simulated laser rangfinder, providing the angular field of view of 360^o, and the resolution of 12^o. In other words, the mobile robot has 30 eye pointing out in all directions. We assume that, in each direction the robot can observes 5 variables in its vicinity: (1) the range from the robot to the walls, obstacles and targets; the type of sensed objects including (2) targets, and

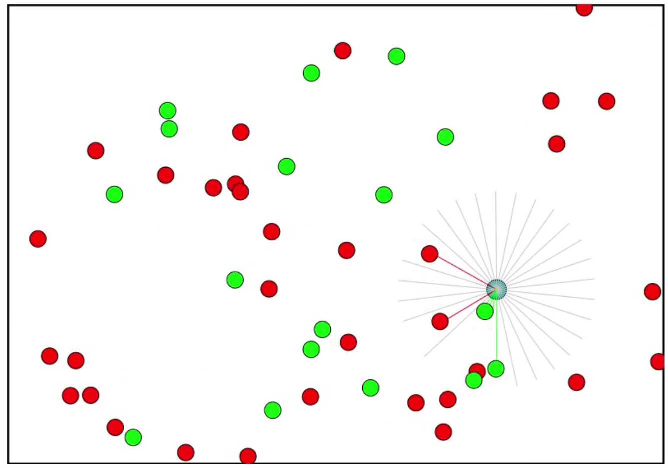

Fig. 3. A simulated office-like scenario with walls, dynamic obstacles (red circles), dynamic targets (blue circles), and a mobile robot. The mobile robot is equipped with a laser rangfinder, providing angular field of view of 360^o. The robot is requested to approach the dynamic targets while avoiding dynamic obstacles.

(3) obstacles; and the velocity of the sensed objects including (4) targets, and (5) obstacles. In addition, the mobile robot's proprioception includes two additional sensors for its own speed in both directions v_x and v_y. Therefore, there is a total of (30x5)+2=152 dimensional state space.

In order to evaluate the performance of the proposed mobile robot navigation framework, the collision index proposed by Truong et al. [9] is used. The immediate reward awarded to the mobile robot is +1 for catching up a target and -1 for making contact with any obstacle. The parameters of the Algorithm 1 are set including the discount factor $\gamma = 0.9$, the epsilon-greedy policy $\varepsilon = 0.2$, and the learning rate $\alpha = 0.005$.

B. Simulation Results

In this section, using the trained model, we conduct two experiments in two different scenarios to verify the performance of the proposed mobile robot navigation framework. The simulation results are shown in Fig. 4.

1) Experiment 1 – Stationary Scenario: A mobile robot and fifty targets and obstacles are randomly distributed in the scenario. The robot is equipped with our proposed navigation framework, but the targets and obstacles are stationary. In this experiment, the mobile robot is able to catch up the static targets (green circles) and avoid the static obstacles (red circles). In addition, Fig. 4(a) indicates that, the mobile robot always keeps a safety distance to the stationary obstacles. In other words, the mobile robot is capable of autonomously and safely navigating in the stationary environment.

2) Experiment 1 – Dynamic Scenario: A mobile robot and fifty targets and obstacles are randomly distributed in the scenario. The robot is equipped with our proposed navigation framework, however the targets and obstacles are randomly move around. In this experiment, the mobile robot is capable of catching up the dynamic targets and avoiding the moving obstacles. Moreover, Fig. 4(b) shows that, the mobile robot

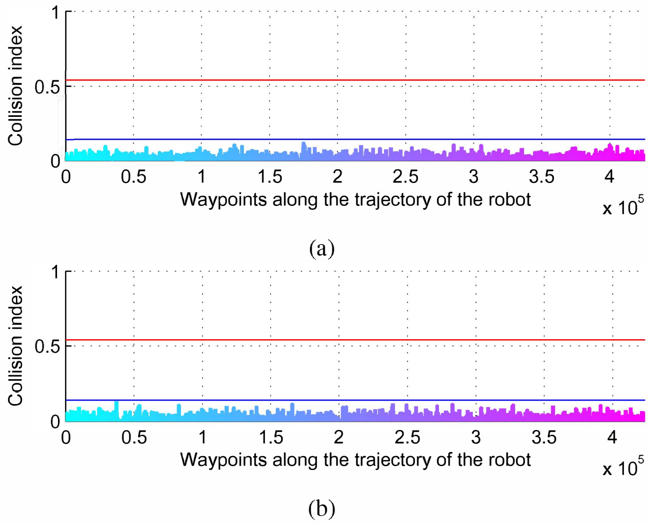

Fig. 4. Simulation results of the proposed mobile robot navigation framework in two different scenarios: (a) stationary scenario, and (b) dynamic scenario.

always keeps a safety distance to the dynamic obstacles in the robots vicinity. In other words, the mobile robot is capable of autonomously and safely navigating in the dynamic environment.

Overall, the proposed extended navigation framework enables an autonomous mobile robot to catch up dynamic targets and avoid moving obstacles in the dynamic environments, providing the safety for the robot and the surrounding objects.

IV. Conclusion

We have presented an extended navigation framework for autonomous mobile robots in dynamic environments using q-learning algorithm. The main idea of the proposed algorithm is to provide the mobile robots the relative position and motion of the obstacles to the robots, the safety constraints such as minimum distance from the robot to the obstacles, and a learning model. We then distribute the robots into a dynamic environment. The robots will automatically learn to adapt to the environment by their own experienced through the trial-and-error interaction with the surrounding environment. The simulation results in a simulated environment shows that, our approach is capable of guiding the mobile robots to navigate autonomously and safely in the dynamic environments.

In the future, we will implement the proposed framework in our mobile robot platform for real-world experiments. In addition, deep reinforcement learning techniques [18] and continuous actions (acceleration and steering) [19] should also be considered to improve the learning efficiency and navigation task of the mobile robots.

References

[1] M. Bennewitz, *Mobile Robot Navigation in Dynamic Environments*. Germany: PhD Thesis, Department of Computer Science, University of Freiburg, 2004.

[2] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*. The MIT Press, February 2011.

[3] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robotics and Autonomous Systems*, vol. 61, pp. 1726–1743, 2013.

[4] J. Rios-Martinez, A. Spalanzani, and C. Laugier, "From proxemics theory to socially-aware navigation: A survey," *International Journal of Social Robotics*, September 2014.

[5] P. Trautman, J. Ma, R. Murray, and A. Krause, "Robot navigation in dense human crowds: the case for cooperation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2153–2160, May 2013.

[6] C. Sungjoon, K. Eunwoo, and O. Songhwai, "Real-time navigation in crowded dynamic environments using gaussian process motion control," in *IEEE International Conference on Robotics and Automation*, May 2014.

[7] M. Shiomi, F. Zanlungo, K. Hayashi, and T. Kanda, "Towards a socially acceptable collision avoidance for a mobile robot navigating among pedestrians using a pedestrian model," *International Journal of Social Robotics*, vol. 6, no. 3, pp. 443–455, 2014.

[8] B. Daman and J. Van den Berg, "Generalized reciprocal collision avoidance," *The International Journal of Robotics Research*, vol. 34, no. 12, pp. 1501–1514, 2015.

[9] X. T. Truong and T. D. Ngo, "Dynamic social zone based mobile robot navigation for human comfortable safety in social environments," *International Journal of Social Robotics*, pp. 663–684, 2016.

[10] D. Vasquez, B. Okal, and K. O. Arras, "Inverse reinforcement learning algorithms and features for robot navigation in crowds: An experimental comparison," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1341–1346, September 2014.

[11] B. Kim and J. Pineau, "Socially adaptive path planning in human environments using inverse reinforcement learning," *International Journal of Social Robotics*, pp. 1–16, 2015.

[12] K. Henrik, S. Markus, S. Christoph, and B. Wolfram, "Socially compliant mobile robot navigation via inverse reinforcement learning," *The International Journal of Robotics Research*, pp. 1289–1307, 2016.

[13] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1st ed., 1998.

[14] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *International Journal of Robotics Research*, vol. 32, pp. 1238–1274, Sept. 2013.

[15] C. Xie, S. Patil, T. M. Moldovan, S. Levine, and P. Abbeel, "Model-based reinforcement learning with parametrized physical models and optimism-driven exploration," *arXiv:abs/1509.06824*, 2015.

[16] A. Y. Ng, A. Coates, M. Diel, V. Ganapathi, J. Schulte, B. Tse, E. Berger, and E. Liang, "Autonomous inverted helicopter flight via reinforcement learning," in *The 9th International Symposium on Experimental Robotics*, pp. 363–372, 2006.

[17] L. P. Kaelbling, M. L. Littman, and A. P. Moore, "Reinforcement learning: A survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996.

[18] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.

[19] S. Gu, T. P. Lillicrap, I. Sutskever, and S. Levine, "Continuous deep q-learning with model-based acceleration," *arXiv:1603.00748v1*, 2 Mar 2016.