Deep Reinforcement Learning For Mobile Robot Navigation: A Review

Xiang Fei*
School of Data Science
The Chinese University of Hong Kong, Shenzhen
Shenzhen, Guangdong 518172
xiangfei@link.cuhk.edu.cn

Abstract

Mobile robot navigation is the fundamental groundwork for enabling advanced functionalities in autonomous robot systems. Traditional navigation methods with simultaneous localization and mapping (SLAM) suffers from four limitations: (1) The process of constructing the obstacle map and keeping it up-to-date is resource-intensive, (2) A significant dependency exists on precise sensors for accurately map construction, (3) Primarily limited to functioning within uncomplicated and comparatively unchanging environments, (4) Accumulative error was produced because of the modularization. Deep Reinforcement Learning (DRL) algorithms are based on end-to-end input and output mechanisms, efficiently extracting observation features from environments, which has the potential to overcome the limitations of traditional navigation methods. This paper reviews the prior research of DRL in various tasks (path planning, trajectory tracking and control, localization and mapping) of mobile robot navigation. The purpose is to assist researchers interested in this field in comprehending the current exploration of DRL advancements in navigation.

1 Introduction

Mobile robot navigation is the fundamental groundwork for enabling advanced functionalities in autonomous robot systems [8]. Traditional navigation methods are typically based on prior environmental maps [4]. However, the map-based navigation methods are required to construct the maps in advance. To solve this problem, simultaneous localization and mapping (SLAM) [13] is combined with traditional navigation methods to perform real-time operation and enabling exploration in novel and cluttered environment, which integrates path planning [6] and motion planning and control [15]. SLAM can be divided into two main categories based on sensor type: Laser SLAM and Visual SLAM.

Laser SLAM systems are based on laser sensors. It involves the use of a laser range finder or LIDAR (Light Detection and Ranging) device mounted on a robot. LIDAR emits laser beams and measures the time it takes for the laser pulses to return after hitting objects in the surroundings, thereby creating a 3D point cloud of the environment to extract information. Mainstream algorithms for Laser SLAM include Grid Mapping, Karto SLAM and Lidar odometry and mapping (LOAM), SegMap algorithm, etc [11]. Visual SLAM, on the other hand, involves using cameras (visual sensors) to achieve simultaneous mapping and localization. Instead of using laser range finders, Visual SLAM relies on computer vision techniques to extract information from images or video sequences captured by cameras. Visual SLAM algorithms work by analyzing the images to identify and track distinctive features or landmarks in the environment, such as corners, edges, or patterns. These features are

[*] Personal Webpage: https://edgarfx.github.io/

used to create a map of the surroundings while also determining the robot's position and orientation relative to this map. Notable Visual SLAM algorithms include ORB-SLAM [10], PTAM (Parallel Tracking and Mapping) [7], and LSD-SLAM [3]. Both Laser SLAM and Visual SLAM have their own advantages and limitations. Laser SLAM tends to be more accurate in certain environments and lighting conditions, while Visual SLAM might be more versatile but can be affected by changes in lighting, texture, or visual obstructions. In addition, SLAM-based navigation suffers from four limitations: (1) The process of constructing the obstacle map and keeping it up-to-date is resource-intensive, (2) A significant dependency exists on precise sensors for accurately map construction, (3) Primarily limited to functioning within uncomplicated and comparatively unchanging environments, (4) Accumulative error was produced because of the modularization.

With the development and application of deep reinforcement learninig (DRL) in various fields, adapting DRL techniques to robot navigation has become a potential way to overcome the traditional limitations. The DRL algorithm relies on end-to-end input and output mechanisms, efficiently extracting observation features from either discrete or continuous environments [1]. The training parameters of DRL networks are trained jointly and thereby reduce the dependency of precise sensors and avoid accumulative errors. Besides, several DRL-based approaches have shown effectiveness in deriving control actions efficiently from primary sensor inputs. In the context of mobile robots, intricate environments significantly enlarge the sample space. To deal with this problem, DRL methods typically extract actions from discrete spaces to simplify the issue. When employing the DRL method for navigation tasks, a mobile robot doesn't depend on pre-collected diverse environmental data. Instead, it directly uses raw sensor input to generate control actions, which eliminates the necessity of creating a prior environment map, leading to reduced computation and the ability to effectively handle unforeseen scenarios. Therefore, this paper aims to comprehensively review the utilization of DRL in the navigation of mobile robots. The purpose is to assist researchers interested in this field in comprehending the current exploration of DRL advancements in navigation.

The rest of the paper is orgnized as follows. Section II briefly outlines the background and preliminaries of DRL. Section III discusses the research in the field of applying DRL to robot navigation. Finally, conclusions are given in Section IV.

2 Deep Reinforcement Learning

2.1 Reinforcement Learning Framework

Through engaging with the surrounding environment, agents in Reinforcement Learning (RL) fundamentally acquire the association between an environmental state and an action, constituting what is termed as a policy. Reinforcement Learning operates within the framework of a Markov Decision Process (MDP), which can be formally characterized by a quaternion [12].

$$(S, A, R, P) \tag{1}$$

where S and A represent the state information characterizing the environment (state space) and the action space; $s_t \in S$ and $a_t \in A$ represent the agent's state and action at time t; R is the rewards function; P is the state transition probability distribution function.

The MDP has the following characteristics:

$$P(s_{t+1}, r_{t+1}|s_1, a_1, r_1, \dots, s_t, a_t, r_t) = P(s_{t+1}, r_{t+1}|s_t, a_t) \tag{2}$$

During the learning process, the purpose is to maximize cumulative rewards:

$$R_t = r_1 + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{3}$$

where $\gamma \in [0, 1]$ is the discount factor that reflects the importance of current feedback.

The value function is defined as follows:

$$V_\pi(s) = E_\pi[R_t|S_t = s] = E_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}|S_t = s\right] \tag{4}$$

where V_π represents the expected return function under policy π. The optimal value function:

$$V_\pi^* = \max E_\pi[R_t|S_t = s] \tag{5}$$

The action-value function is defined as follows:

$$\begin{aligned} Q_\pi(s,a) =& E_\pi[R_t|S_t = s, A_t = a] \\ =& E_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}|S_t = s, A_t = a\right] \end{aligned} \tag{6}$$

where $Q_\pi(s,a)$ represents the expected return starting from s and taking the action a under policy π. The optimal action-value function:

$$Q_\pi^*(s,a) = \max E_\pi[R_t|S_t = s, A_t = a] \tag{7}$$

Reinforcement Learning (RL) categorization commonly delineates three primary branches: value-based approaches, policy-based approaches, and actor-critic approaches. Actor-critic methods present a hybridization of value-based and policy-based algorithms within the RL framework.

2.2 Value-Based Methods

Value-based methods aim to approximate the value, often represented as the expected return, associated with a specific state within the environment. The optimal policy aligns with the action yielding the most favorable action-value function. Notably, Q-learning [14] stands out as one of the extensively utilized RL algorithms, employing updates to the Q-value via the application of the Bellman equation:

$$Q_{i+1}(s,a) = E_\pi[R_t + \gamma \max Q_i(s_{t+1}, a_{t+1})|S_t = s, A_t = a] \tag{8}$$

where Q_i moves towards the optimal action-value function Q_i^* when $i \to \infty$. The optimal policy:

$$\pi^* = \arg\max_{a \in A} Q^*(s,a) \tag{9}$$

The limitation of value-based methods is that it is difficult to be performed in continuous and high-dimensional spaces.

2.3 Policy-Based Methods

In contrast to the indirect policy derivation characteristic of value-based methods, policy-based methods directly seek the optimal policy. Policy gradient is the most popular methods. This technique entails the computation of an estimation for the agent's policy gradient, achieved through a stochastic gradient ascent algorithm.

Policy gradient algorithms update parameters to adjust the policy as follows:

$$\hat{g} = \hat{\mathbb{E}}_t[\nabla_\theta log\pi(a_t|s_t;\theta)R_t] \tag{11}$$

The limitation of policy-based methods is that the policy evaluation is inefficient and the convergence is time-consuming.

2.4 Actor-Critic Methods

The Actor-Critic (AC) Methods combine the value-based and policy-based methods together to overcome the limitation of each of the two methods. The policy-based methods serve as the actor, contributing to choose action, while the value-based methods serve as the critic, contributing to evaluate the value and advantages of the adopted action. The AC method can be expressed as follows:

$$\nabla \bar{R}_\theta = \widehat{E}_\tau[\nabla_\theta log\pi(a_t|s_t,\theta)(Q(s_t,a_t) - V(s_t))]$$

Therefore, AC methods possess the capability to address both continuous and discrete problem domains while incorporating a single-step update frequency that enhances the efficiency of learning and training processes.

3 Deep Reinforcement Learning in Various Tasks of Robot Navigation

To perform mobile robot navigation, various tasks are required to implemented, such as path planning, trajectory tracking and control, localization, mapping, etc. In this section, the applications of DRL in these tasks are summarized.

3.1 Path Planning

Path planning in mobile robot navigation is the process of determining the best route for a robot to travel from its starting position to a desired destination while avoiding obstacles or adhering to specific constraints. Mainstream conventional path planning methods include Dijkstra's Algorithm, A-star Algorithm, Visibility Graphs, Voronoi Diagrams, Rapidly-exploring Random Trees (RRT), Artificial Potential Fields, etc. However, these methods suffer from huge resource consumption, complex and dynamic environment, dependence of precise sensors, and accumulative errors. DRL algorithms have been applied to solve these issues.

Xin et al. [16] proposed a novel mobile robot path planning method, utilizing end-to-end deep reinforcement learning algorithms. Firstly, They designed and trained a deep Q-network (DQN) estimate the mobile robot state-action value function. Then, they determined the Q value corresponding to each possible mobile robot action (i.e., turn left, turn right, forward) based on the well trained DQN, where the original RGB images captured from the environment without any hand-crafted features and features matching are used as the input data. After that, the current optimal action is selected by the action selection strategy and the mobile robot is able to reach the goal destination while avoiding obstacles. They conducted 30 times path planning experiments in the seekavoid_arena_01 environment on DeepMind Lab platform. The experimental results show that their deep reinforcement learning based robot path planning method is able to reach more goal points while avoiding obstacles and have better path planning performance.

Zhang et al. [19] proposed a new reinforcement learning algorithm called Geometric Reinforcement Learning (GRL) to perform path planning for Unmanned Aerial Vehicles (UAVs). Previous work on path planning for a single UAV such as Voronoi Graph Search and Visibility Graph Search are not real-time, and may also fail to operate when facing some complex and dynamic scenarios. Besides, traditional Q-Learning algorithm fails to use the geometric distance information which is of great significance for path planning when only partial information of the map is available. Therefore, Zhang et al. divided the map into a series of lattice, path planning for UAVs is formulated as the problem of the optimal path planning. They finely modulated the parameter to control the size of the map to reduce the complexity of calculation of the continuous threat function. In addition, they generalized the algorithm to multiple UAVs by using the information shared from other UAVs, and thereby provided an effective solution for the path planning and avoids local optimums. Experiments showed that their method performed very well regarding of the path length and the integral risk measure.

Bae et al. [2] proposed a novel multi-robot path planning algorithm using Deep Q-Learning conbined with Convolutional Neural Network (CNN). For traditional path planning methods, robots suffers from huge time-consumption caused by wide search area and predesigned formation under a given environment. Besides, these methods can't actively deal with various complex and dynamic scenarios since robots have difficulty in recognize obstacles and cooperative robots. Therefore, Bae's team utilized CNN to analyze the situation using image information of the environment and adopt Deep Q-Learning to perform robot navigation. Simulation experiments showed that their algorithm performed more flexible and efficient compared with conventional methods.

3.2 Trajectory Tracking and Control

Trajectory tracking and control in robot navigation involve the execution of a planned path or trajectory by a robot to accurately follow a predefined trajectory while accounting for uncertainties, disturbances, and environmental changes. This task primarily focuses on controlling the robot's motion to adhere as closely as possible to the desired path generated by the path planning algorithms, which encompasses the design and implementation of control algorithms to regulate the robot's movements (velocity, acceleration, and orientation). Traditional methods include Proportional-Integral-Derivative (PID)

Control, Model Predictive Control (MPC), Linear Quadratic Regulator (LQR), Computed-Torque Control, etc. DRL algothms are also applied to achieve better performance.

Yu et al. [18] proposed a novel deep reinforcement learning method to solve the trajectory tracking and control problem of Autonomous Underwater Vehicles (AUVs). Their proposed deep reinforcement learning of an underwater motion control system consists of two neural networks made up of multiple fully connected layers for different purposes: one network selects action and the other evaluates whether the selected action is accurate. These modules can modify themselves through a deep deterministic policy gradient(DDPG). Both theoretic proof and simulation results show that the method is more accurate than traditional PID control in solving the trajectory tracking of AUV in complex curves to a certain precision.

Lou et al. [9] proposed a novel adaptive control algorithm based on reinforcement learning framework to stablize a quadrotor helicopter and overcome the negative effects of inaccurate system parameters and unpredicted external disturbances. They utilized policy search methods based on a commend-filtered non-linear control algorithm to add and learn adaptive elements. In addition, they provided a new kernel-based regression learning method to predict the inaccurate system parameters. Moreover, they used Policy learning by Weighting Exploration with the Returns (PoWER) and Return Weighted Regression (RWR) to learn the appropriate parameters for adaptive elements, thereby cancelling the effect of external disturbances. Finally, Lou et al. conducted simulation experiments under several conditions and demonstrated the efficiency of the adaptive trajectory-tracking algorithm with reinforcement-learning techniques.

3.3 Localization and Mapping

Localization in robot navigation is the process of determining a robot pose (position and orientation) within an environment, while mapping is to creates a representation of the exploring environment. Mainstream conventional methods are focused on simultaneous localization and mapping, which perform localization and mapping tasks simultaneously and enable real-time operation. Challenges in SLAM include dealing with sensor noise, uncertainties, dynamic environments, loop closures (recognizing previously visited locations), and computational complexity. DRL algorithms have the potential to overcome the traditional limitations.

Garrote et al. [5] proposed to combine a Particle-Filter based Localization (PFL) module and a Reinforcement Learning-based map updating module to build a novel mobile robot localization solution, which integrates relative measurements and absolute indoor positioning sensor (A-IPS) data. Besides, to overcome the problem of 2D-LiDARs' failure in featureless areas, they modified the classic PFL approach and incorporated A-IPS position measurements in the prediction and update stages. The localization method can update the map whenever major modifications are detected. In addition, to deal with the erroreous map association issue caused by the randomness of the PFL and the inconsistencies in the estimated pose, they proposed to use reinforcement learning to assign higher rewards the greater is the overlap between the map and the 2DLIDAR scans, and then a proper update of the map is achieved. Finally, Garrote's team validated the proposed pipeline in a differential drive platform with algorithms developed in ROS. Tests were performed in two scenarios in order to assess the performance of both the localization module and the map update stage. The experiment results show that their localization method offers improvements in relation to prior approaches, and consequently suggest promising perspectives for the proposed map update decision framework.

Yoshimura et al. [17] proposed a novel type of map called highlighted map for mobile robot localization, on which the landmarks in monotonous environments are highlighted. They also introduced the generation method of highlighted map. By using this map, robots can use such landmarks as clues for localization, and thereby, the localization performance can be improved without the requirements of updating their sensors or online computation. In addition, they formulated the problem of making a highlighted map and proposed a new numerical optimization method based on reinforcement learning, which automatically identifies and highlighted the vital landmarks on the map. They proved that the optimization converges under certain technical assumptions. Finally, they conducted both simulation and real-world experiments. Results showed that the highlighted map provided better localization accuracy than a conventional map representation.

4 Conclusion

Mobile robot navigation is the fundamental basis for enabling advanced functionalities in autonomous robot systems. Deep Reinforcement Learning (DRL) algorithms have been applied to overcome the limitations of traditional navigation methods: (1) The process of constructing the obstacle map and keeping it up-to-date is resource-intensive, (2) A significant dependency exists on precise sensors for accurately map construction, (3) Primarily limited to functioning within uncomplicated and comparatively unchanging environments, (4) Accumulative error was produced because of the modularization. This paper reviews the prior research of DRL in three main tasks of mobile robot navigation: path planning, trajectory tracking and control, localization and mapping. However, current applications of DRL methods in the field of robot navigation also have some limitations, such as low sample efficiency; the gap from simulation to real-world cases; and lack of proper evaluation benchmarks. Future work can be carried out by addressing these issues

References

[1] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38, 2017.

[2] H. Bae, G. Kim, J. Kim, D. Qian, and S. Lee. Multi-robot path planning method using reinforcement learning. *Applied Sciences*, 9(15), 2019.

[3] J. Engel, T. Schöps, and D. Cremers. Lsd-slam: Large-scale direct monocular slam. In *European conference on computer vision*, pages 834–849. Springer, 2014.

[4] D. Filliat and J.-A. Meyer. Map-based navigation in mobile robots:: I. a review of localization strategies. *Cognitive systems research*, 4(4):243–282, 2003.

[5] L. Garrote, M. Torres, T. Barros, J. Perdiz, C. Premebida, and U. J. Nunes. Mobile robot localization with reinforcement learning map update decision aided by an absolute indoor positioning system. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1620–1626, 2019.

[6] S. S. Ge and Y. J. Cui. New potential functions for mobile robot path planning. *IEEE Transactions on robotics and automation*, 16(5):615–620, 2000.

[7] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *2007 6th IEEE and ACM international symposium on mixed and augmented reality*, pages 225–234. IEEE, 2007.

[8] H. Li. Mobile robot navigation based on deep reinforcement learning: A brief review. *Journal of Physics: Conference Series*, 2649(1):012027, nov 2023.

[9] W. Lou and X. Guo. Adaptive trajectory tracking control using reinforcement learning for quadrotor. *International Journal of Advanced Robotic Systems*, 13(1):38, 2016.

[10] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[11] B. Schreck, C. Victorri-Vigneau, M. Guerlais, E. Laforgue, and M. Grall-Bronnec. Slam practice: a review of the literature. *European addiction research*, 27(3):161–178, 2021.

[12] R. S. Sutton and A. G. Barto. Reinforcement learning: An introduction. *Robotica*, 17(2):229–235, 1999.

[13] H. Temeltas and D. Kayak. Slam for robot navigation. *IEEE Aerospace and Electronic Systems Magazine*, 23(12):16–19, 2008.

[14] C. J. Watkins and P. Dayan. Q-learning. *Machine learning*, 8:279–292, 1992.

[15] X. Xiao, B. Liu, G. Warnell, and P. Stone. Motion planning and control for mobile robot navigation using machine learning: a survey. *Autonomous Robots*, 46(5):569–597, 2022.

[16] J. Xin, H. Zhao, D. Liu, and M. Li. Application of deep reinforcement learning in mobile robot path planning. In *2017 Chinese Automation Congress (CAC)*, pages 7112–7116, 2017.

[17] R. Yoshimura, I. Maruta, K. Fujimoto, K. Sato, and Y. Kobayashi. Highlighted map for mobile robot localization and its generation based on reinforcement learning. *IEEE Access*, 8:201527–201544, 2020.

[18] R. Yu, Z. Shi, C. Huang, T. Li, and Q. Ma. Deep reinforcement learning based optimal trajectory tracking control of autonomous underwater vehicle. In *2017 36th Chinese Control Conference (CCC)*, pages 4958–4965, 2017.

[19] B. Zhang, Z. Mao, W. Liu, and J. Liu. Geometric reinforcement learning for path planning of uavs. *Journal of Intelligent & Robotic Systems*, 77(2):391–409, Feb 2015.