Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2025 SCPE. Volume 26, Issues 1, pp. 423–431, DOI 10.12694/scpe.v26i1.3841

RESEARCH ON AUTONOMOUS NAVIGATION AND CONTROL ALGORITHM OF INTELLIGENT ROBOT BASED ON REINFORCEMENT LEARNING

YUNLONG YI* AND YING GUAN†

Abstract. The last few decades have seen impressive developments in the field of robotics, especially in the areas of autonomous navigation and control. Robust algorithms that can facilitate effective decision-making in real-time settings are needed as the need for intelligent robots that can function in complex and dynamic contexts grows. Through trial-and-error interactions with their surroundings, reinforcement learning (RL) has become a promising method for teaching intelligent agents to navigate and control robots independently. The purpose of this study is to look at the creation and use of reinforcement learning algorithms for intelligent robot control and autonomous navigation. With an emphasis on methods like deep Q-learning, policy gradients, and actor-critic approaches, the research delves into the theoretical underpinnings of reinforcement learning and how it has been applied to the field of robotics. This study assesses how well RL algorithms work to help robots acquire the best navigational strategies in challenging surroundings through an extensive literature review and empirical investigation. In addition, the study suggests new improvements and optimizations for current reinforcement learning algorithms to tackle problems unique to robot navigation, such as avoiding obstacles, routing, and interactions with dynamic environments. These improvements increase the effectiveness, flexibility, and security of independent robot navigation systems by utilizing knowledge from cognitive science and neuroscience. The suggested methods are experimentally evaluated through both real-world applications on physical robotic platforms and simulation-based research. Performance measures including navigation speed, success rate, and collision avoidance ability are used to evaluate how well the suggested algorithms operate in different scenarios and circumstances.

Key words: autonomous navigation, control algorithm, intelligent robot system, reinforcement learning

1. Introduction. The demands of the modern logistics and warehousing industries can no longer be met by traditional manual sorting and transit efficiency due to the quick development of intelligent manufacturing and e-commerce [9]. In an assembly shop, front-line personnel can be replaced with indoor robots, enabling automation and intelligent delivery. Their automated transportation system is safer and more dependable, and their independent transit is more effective. The fast expansion of modern industry has led to increasingly complex application situations for robots [11], making the research of autonomous intelligent navigation decision-making algorithms crucial.

Several techniques and technologies must be combined to create a drone navigation system based on reinforcement learning. Sensing and understanding its surroundings: Using sensors to give the drone situational awareness is essential [5]. Drones may gather information about their environment through sensors including proximity detectors, GPS, LIDAR, and cameras. This information can be utilized to guide the aircraft and avoid obstacles. To enable continuous tracking and emergency intervention, swift and dependable connectivity is also necessary for remote control from the ground [6]. To allow the drone to make judgments depending on how it perceives its surroundings and its current condition, an effective autonomous navigation algorithm is also required.

The path planning issue has steadily grown in importance as a study topic in recent years. Conventional path planning techniques consist of the rapidly expanding random tree method [12], the artificial potential field method [2], the Dijkstra algorithm [12], the A∗algorithm, and the D∗ algorithm. But even in path planning, the conventional path planning algorithm is unable to completely comprehend the ever-more-complex and unknown external environment data. The environment's complexity makes it challenging to represent, and the prior algorithm was prone to an unstable condition of convergence. In addition, it struggles with inadequate data processing capability in large-scale areas [13]. One new sophisticated learning algorithm is called reinforcement

*School of Information, Shenyang Institute of Engineering, Shenyang 110136, Liaoning, China (yunlongyieanreas@outlook.com)
†School of Information, Shenyang Institute of Engineering, Shenyang 110136, Liaoning, China

learning.

Robotics is a field that is always changing, and autonomous navigation and control in particular has increased the need for reliable algorithms that can make decisions quickly and effectively in complex, dynamic settings. Reinforcement learning (RL) has become a powerful technique that allows intelligent agents to autonomously move and manipulate robots by means of trial-and-error interactions with their surroundings. The goal of this research is to improve reinforcement learning algorithms in order to improve their safety, flexibility, and efficiency when guiding robots through challenging situations. It seeks to investigate the integration of advanced methodologies that are essential to creating self-governing systems that are capable of learning and adapting on their own without human assistance, such as policy gradients, deep Q-learning, and actor-critic approaches.

Mobile robots' capacity to navigate autonomously is crucial because it can guarantee that the platform will get at the destination from the starting point without running into any of the many impediments in its path. Trajectory planning [7], tracking control [19], and simultaneous localization and mapping (SLAM) [1] are common steps in classical navigation techniques. But SLAM takes a long time and needs a lot of LIDAR density and precision. For mobile robots, autonomous navigation remains difficult in the absence of an obstacle map and poor range information. Consequently, scholarly interest in the unique navigation approach of end-to-end online learning based on deep reinforcement learning (DRL) has been substantial. The main contribution of the proposed method is given below:

1. We provide a novel DRL algorithm that combines the most recent methods for reward structuring, exploitation, and exploration to improve autonomous robots' decision-making and learning capabilities.
2. Our study offers a hybrid architecture that combines the best features of deliberative and reactive architectures to ensure strategic planning and real-time response in dynamic contexts.
3. We illustrate our approach's scalability by implementing it on several robotic platforms and situations. Furthermore, the learnt rules' transferability is assessed, demonstrating the algorithm's flexibility to new workloads without requiring a significant amount of retraining.
4. A thorough examination of the DRL algorithm's constituent parts is part of the study, and ablation tests are used to determine how each part contributes to overall performance, guaranteeing the results' consistency and openness.

The main research question relies on,

1. What are some ways to improve the effectiveness and adaptability of reinforcement learning algorithms in the dynamic environment of autonomous robot navigation?
2. What specific roles can cutting-edge methods like policy gradients, actor-critic methods, and deep Q-learning play in improving robot decision-making in real-time scenarios?
3. How can RL algorithms be made more contextually adaptive and efficient at learning by the integration of insights from cognitive science and neuroscience?
4. What are the shortcomings of existing reinforcement learning models when it comes to addressing the intricacies of robotic navigation in the real world, like avoiding obstacles, interacting dynamically with the surroundings, and ensuring safety?

The rest of our research article is written as follows: Section 2 discusses the related work on various autonomous navigation, control algorithm of intelligent robot,and Deep Learning Algorithms. Section 3 shows the algorithm process and general working methodology of proposed work. Section 4 evaluates the implementation and results of the proposed method. Section 5 concludes the work and discusses the result evaluation.

2. Related Works. Numerous fields, including home services, space exploration, automated industrial environments, and rescues, have made extensive use of robots. The main need for these applications is collision-free path planning. Consequently, path planning skills are critical to the completion of robot navigation tasks. A$*$ (A-start), RRT (rapidly explored random tree), and Dijkstra are examples of traditional path planning algorithms. To achieve the planned path, they must first comprehend all available environmental data, construct an environment model [20], and use the path-searching algorithm in accordance with predetermined optimization criteria. Since environment modeling, the basis of traditional path planning approaches, is weak and only provides local optimal solutions, it is very inaccurate when handling complicated situations.

Planning a route and identifying objects and mitigation are the two main subtasks involved in the difficult task of drone navigation. The work must be divided into smaller, optimally solved tasks because it is not always feasible to plan the entire approach ahead of time in a foreign setting [3, 10]. In unrestricted contexts where landmark placements frequently vary, simultaneous localization and mapping (SLAM) based methods are perfect. These techniques draw data from sensors such as LiDAR and IMU, although they typically add to the computing load [4]. When operating in dynamic situations, the drone's path plan needs to be revised on a regular basis to account for impediments that are identified while in flight and come into its path.

Afterwards, intelligent bionic path planning techniques with some autonomies were developed; these mostly consisted of particle swarm optimization [14], optimizing ant colonies [18], and genetic algorithms [21]. The intelligent bionic algorithm can perform planning path tasks in a dynamic space; however, when the computational load is high, path planning efficiency is low and real-time path planning efficiency is not ensured [8]. Furthermore, the intended path is not the best option when the robot does not know enough about its working surroundings. These conventional path planning methods typically have trouble digesting highly dimensional, complex data on the environment in challenging situations, or they are prone to local optimal performance.

Tasks involving path planning are often solved by adapting artificial intelligence techniques. It is simple to use brute-force or exhaustive search methods for UAV path planning jobs [15]. Although they can be quite slow, the breadth-first and depth-first search for space techniques are thorough and always locate a path if one is available, or the shortest of all accessible paths. To prevent dead ends, they can also be used in conjunction with backtracking [16]. Although they can also be used to speed up search, greedy techniques always run the danger of piling into neighbourhood minima. Because more than 50 targets make brute-force search impractical, these approaches begin with one or more quick fixes and work their way up to the best answer by applying local modifications and, if necessary, random restarts [17].

Although reinforcement learning (RL) has shown great promise for robotic control, current algorithms frequently encounter difficulties related to the dynamic and unpredictable nature of real-world situations. Managing high-dimensional sensory inputs, learning optimal policies quickly, and reacting in real time to environmental changes are some of these issues. Moreover, a lot of training data and computer power are needed for most RL techniques, and they could not scale well in real-world applications or generalize well in other contexts. The development of RL algorithms that can dependably function in a variety of operational contexts without sacrificing performance or safety, interpret complex sensory data fast, and adapt to new situations are still far from being fully developed.

3. Proposed Methodology. The proposed methodology for autonomous navigation and control algorithm of intelligent robot is evaluated by using Deep Reinforcement Method (DRL). At first, build a virtual world in which the robot can function. This could be a fully virtual environment created to test scenarios, or it could be a digital duplicate of an actual location. Next, based on the specifications of the problem, select an appropriate Deep Reinforcement Learning (DRL) algorithm, such as Deep Q-Networks (DQN). To better meet the navigation and control problems unique to the intelligent robot, train the selected DRL algorithm. Create a system of rewards that incentivizes the desired behavior. Assessing the robot's ability to navigate and control itself in real-world settings and gathering performance statistics. Return the design, adjusting the robot control systems and DRL model in response to evaluation results. In figure 3.1 shows the architecture of proposed method.

The indoor robot's autonomous navigation system is a noteworthy technological development in robotic control, providing several technical advantages essential for dependable and effective operations in changing interior environments. The robot can navigate difficult spaces with great accuracy thanks to the integration of GPS-INS (Global Positioning System-Inertial Navigation System) assistance, AHRS (Attitude and Heading Reference Systems), and high-precision dynamic 3D processing. This degree of accuracy is necessary for operations like material handling and distribution in warehouses and manufacturing plants, which call for precise movement and placement.

Perpetual motion formulas, which take into account ongoing changes in position and velocity, provide the foundation for the robot's sophisticated navigational abilities. This allows the robot to move smoothly and respond quickly to changes in its surroundings. The robot's radar system, which gathers vital information such as target spectrum, bearings, and velocity to enable reliable collision avoidance and accurate docking procedures, significantly improves these capabilities. In addition to helping with navigation, this radar technology enables

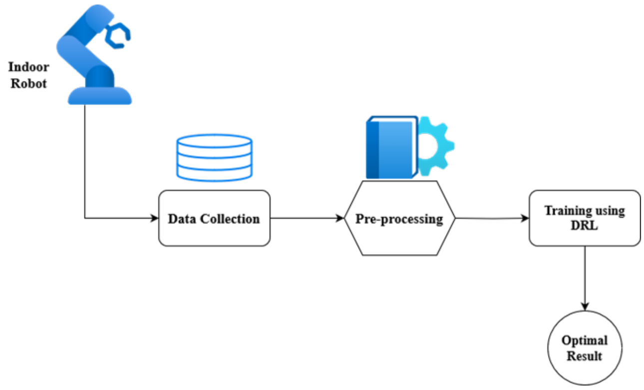

Fig. 3.1: Architecture of Proposed Method

the robot to keep an eye on its surroundings and take proactive measures to avoid potential obstructions or modify its course as needed.

Moreover, the robot's intelligent control mechanism heavily relies on its array of distance sensors. These sensors continuously scan the surroundings to offer real-time information on the location and size of impediments, allowing the robot to quickly alter its course. In highly populated or unpredictable areas, this sensor-based method to obstacle identification is essential for protecting the robot's operational integrity and guaranteeing safety.

3.1. Indoor Robot for Intelligent Control Mechanism. The indoor robot has sophisticated automated navigational capabilities, including high-precision dynamic 3D processing data, AHRS, and GPS-INS inertial navigation support systems, to fulfill the distribution duty. The robot's perpetual motion formulas are determined using the following equations to guarantee brevity and universality in the 2D plane:

$$\begin{bmatrix} \dot{x}\,(t) \\ \dot{y}\,(t) \\ \dot{\varphi}\,(t) \\ \dot{v}\,(t) \end{bmatrix} = \begin{bmatrix} v(t)\cos\phi(t) \\ v(t)\sin\phi(t) \\ \omega\,(t) \\ a(t) \end{bmatrix} \tag{3.1}$$

If ϕ denotes the robot's motion guidance, v is its velocity, and x and y stand for the robot's 2D coordinates in relation to its surroundings. The equations that follow can be used to characterize the status report of time t during the interval:

$$\begin{cases} x\,(t) = x\,(t-1) + v\,(t-1)\,\Delta t\cos\phi\,(t-1) \\ y\,(t) = y\,(t-1) + v\,(t-1)\,\Delta t\sin\phi\,(t-1) \\ v\,(t) = v\,(t-1) + a\,(t-1)\,\Delta t \\ \phi\,(t) = \phi\,(t-1) + \omega(t-1)\Delta t \end{cases} \tag{3.2}$$

The indoor robot has a radar attached to pick up signals. Target spectrum, bearings, velocity, and other data can be obtained by comparing the broadcast signal with the received target echo. This gives fundamental information for navigating, avoiding collisions, parking spaces, and other tasks. The relative azimuth angle φ and the distance in relation D among the indoor robots and the point of interest can be determined at any time t. Furthermore, assume that the goal position vector S_d and the robot position vector S_{uav} are

$$S^{uav} = [x_t, y_t, z_t]^T \; S_d = [x_t, y_t, z_t]^T \tag{3.3}$$

Furthermore, the robot's observation equation to the target point in a two-dimensional coordinate system at a specific time is defined as

$$z = \begin{bmatrix} D \\ \varphi \end{bmatrix} = \begin{bmatrix} \|\,(x_t, y_t) - (x_d, y_d)\,\|_2 \\ \text{actan}\frac{y_t - y_d}{x_t - x_d} \end{bmatrix} \tag{3.4}$$

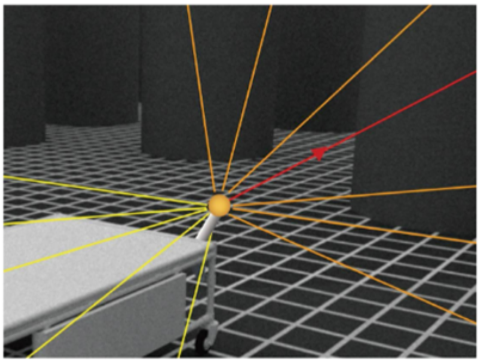

Fig. 3.2: Structure of Agent Sensor

The primary obstacle to autonomous navigation and intelligent control of the robot is a complex and dynamic environment. The robot must identify environmental hazards to navigate autonomously. Therefore, the robot is equipped with a dozen distance sensors to aid in its detection of any obstructions that may be within its detectable range across the front. The robot's ability to detect obstacles at any given time is defined as follows:

$$O_o = [d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9, d_{10}, d_{11}, d_{12}] \tag{3.5}$$

wherein the data sent by the respective sensor is represented by the numbers $d_1, d_2, \ldots, d_3$. The sensor information in front of the agent is represented by the red line in Figure 3.2, the sensor signal in advance of the agent (which includes the robot's right and left sides) is represented by the orange-colored lines, and the sensor signal coming from behind the agent is represented by the yellow lines. If the sensor finds no obstacles, its maximum detectable range is set to L; if it detects obstacles, the distance between the agent and the obstacle is represented by $d_n \in [0, L]$.

3.2. Autonomous Navigation for Indoor Robot using DRL. Large-scale and complicated situations are challenging to handle using the classic path planning algorithm due to its poor convergence speed and high processing requirements. Deep reinforcement learning can be implemented in an end-to-end observation and management systems with strong flexibility through the combination of the perceptual capacity of deep learning with the decision-making power of reinforcement learning. This can significantly increase the effectiveness of path planning.

The agent engages with the surroundings at each instant to acquire a high-dimensional observation, and the deep learning approach may be used to perceive the state attributes. The action's value function is assessed by considering the anticipated return, and the action that corresponds to the current state is mapped to it. By repeatedly performing the procedures, the environment reacts to this action and receives the subsequent observation, allowing the best possible approach to be determined.

In particular, the Markov decision process, symbolized by a quadruple (S, A, R, c), can be used to show the entire process of learning of an agent. The agent's observations and state are represented by the quad S; the tasks that the agent can perform are represented by A; the reward function, R, represents the agent's rewards upon completion of an action in a particular state; and the discount coefficient, c, balances immediate and accumulative rewards during the learning process. In figure 3.3 shows the structure of DRL.

4. Result Analysis. Simulation tests are put up to confirm the efficacy of the DRL algorithm in smart navigation and autonomous control of indoor robots. The Gym-agent-master system, Python 3.6, TensorFlow 1.14.0, and PyCharm are used to execute the environment that has been simulated. The cylinder in the simulated environment is a barrier, and it is generated in the Northeast geodetic coordinate system using the VTK third-party software. The barriers in the scenario being simulated have a radius of one meter and a centre-to-centre distance of three meters. The robot's maximum running speed is set at 2.0 m/s. To guarantee the

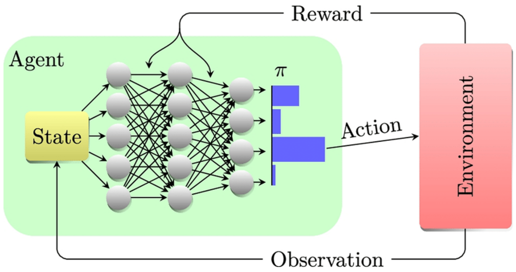

Fig. 3.3: Structure of DRL

efficacy of the navigational task, the robot and destination must be at least 50 meters apart at the beginning, and the simulated phase should last for one second.

In the test of simulation, a simulator is built to enable the robot to handle itself intelligently and autonomously in expansive, complicated settings. The robot's dynamical physical restrictions are disregarded, and its shape is abstracted as a sphere to facilitate the experiment of simulation. To guarantee scarce prizes, there must be a minimum separation of more than 30 meters between the starting point and the destination. The robot has a rangefinder attached so it can be observed. The goals of self-navigating and intelligent operation are considered accomplished when the robot approaches the destination and the distance between it and its intended location is less than one meter.

The AGV's training process begins after the pertinent parameters are specified. This round will be considered ended if the robot cannot finish the training assignment or encounters a barrier within the allotted time. After ten, the experiment will restart, and the subsequent round will start. Simulating the real world, the situation's update rules are established as follows: the robot's setting, its destination, and the number of barriers in every round are all randomly determined.

The AGV's training process begins after the pertinent parameters are specified. This round will be considered ended if the robot cannot finish the training task or collides with an obstacle within the allotted time. At ten, the game will restart and the subsequent round will start. Simulating the real world, the scenario's update rules are established as follows: the robot's setting, its final destination, and the amount of obstacles in every round are all arbitrarily determined.

In the simulation experiment, the robot is trained using the DRL, DDPG, and TD3 algorithms, respectively, to confirm the effectiveness of the proposed DRL algorithm in automatic navigation and intelligent control. As seen in Figure 4.1, the robot's reward value at the end of each training round is recorded.

The DRL algorithm exhibits the most pronounced increasing trend, as seen in Figure 4.2, and it takes the lead to reach the high of 240 after roughly 4000 rounds. The TD3 algorithm exhibits severe fluctuations and the lowest return performance. With a high fluctuation, the classic DDPG method does not begin to rise until around 2000 rounds, and it peaks later than the optimized DRL algorithm. However, the DRL algorithm's reward value declined erratically after about 6600 training episodes. However, after about 7100 rounds, it swiftly recovered to a higher, steady level. This demonstrates how the DRL algorithm suggested in this research might enhance training effects by assisting the robot in adapting to the noisy training environment.

The achievement rate for 0–10000 rounds during the training of the suggested DRL, DDPG, and TD3 algorithms is displayed in Figure 4.2. It is evident that under the DDPG and TD3 computations, the robot's job completion rate is less than 80%, and its learnt methods perform poorly. The DRL algorithm training success rate has the highest growing trend in comparing. The success rate is consistently above 80% after 3000 rounds, with a peak value approaching 90%. When compared to the other two algorithms, the DRL algorithm offers the best learning method and the highest success rate.

We conducted 1000 rounds of comparative tests in each of the three scenarios mentioned above to confirm the effectiveness of the robot autonomous navigation strategy under the DRL algorithm. The success rates of indoor robot navigation are displayed in Table 4.1.

In the testing procedure, we simultaneously logged the data of every successful round and calculated the

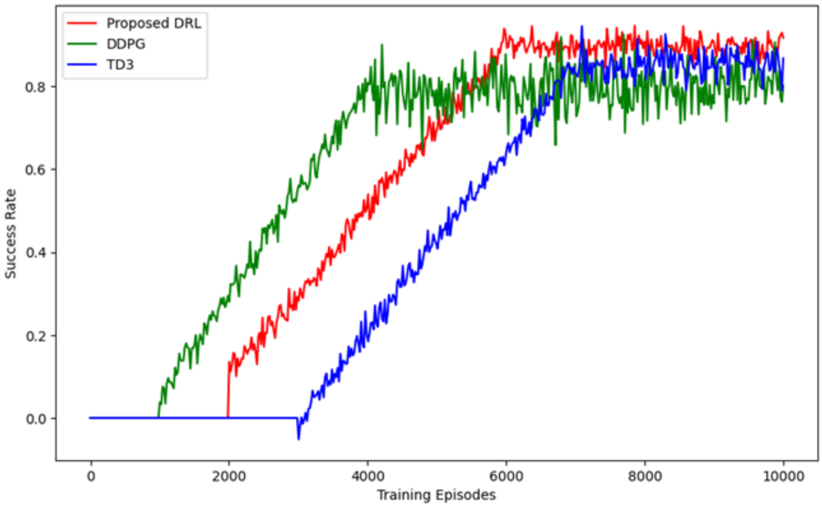

Fig. 4.1: Evaluation of Success rate

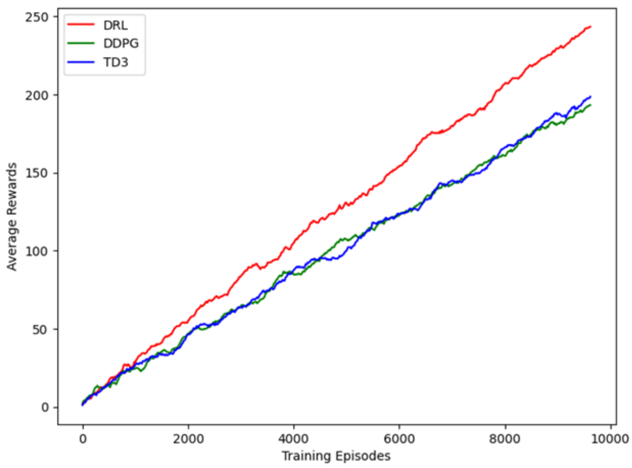

Fig. 4.2: Success rate of Completion of Robots

Table 4.1: Success Rate of Robot capturing autonomous navigation

Methods Used	**100 obstacles**	**150 obstacles**	**200 obstacles**
Proposed DRL	92.35%	83.905%	77%
DDPG	80%	65%	61.32%
TD3	85%	72%	65%

average job completion time under each of the three algorithms, as indicated in Table 4.1. The three algorithms do not significantly differ in how long navigation tasks take in simple settings.

Following training, the intelligent system of control will be evaluated in three different contexts to confirm the efficacy of the indoor robot system navigation approach. The environmental barriers are numbered 100, 150, and 200, in that order. Figure 4.3 displays the outcomes of the simulation. Based on the simulation experiment

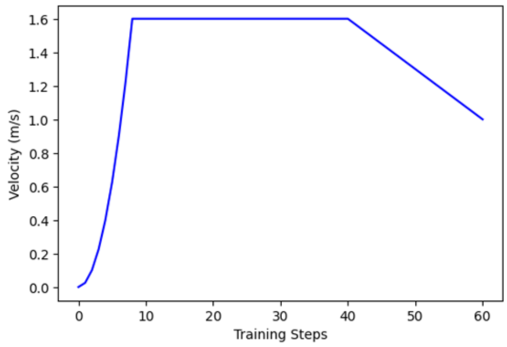

Fig. 4.3: Evaluation of Training steps and Velocity of Robot

results, it may be inferred that an experienced robot is capable of intelligent self-navigating in an environment with varying numbers of obstacles, allowing it to avoid obstacles and reach its target. Additionally, the robot can raise its speed gradually and keep it inside the limit of maximum speed until it reaches its destination, based on the trend of robot speed change.

From the first step to about step 10, the velocity climbs rapidly until it reaches a plateau. From step 10 until about step 40, the plateau has a constant speed just over 1.4 m/s. The velocity gradually decreases after step 40 and stays that way until step 60, the last step shown. This kind of graph could be used to illustrate a simulation or experiment in which the velocity of a vehicle or robot is tracked over time or via repeated training phases. The plateau is a time when the velocity is sustained at its highest level. There are several possible explanations for the decrease: the introduction of a deceleration protocol, the commencement of a limiting factor (such as energy depletion), or changes in the surroundings that could impact velocity.

5. Conclusion. In summary, the study of reinforcement learning-based autonomous navigation and control algorithms for intelligent robots marks a substantial breakthrough in the discipline of robotics. Intelligent robots may learn and change their navigation and control methods in changing circumstances with explicitly programming them by utilizing reinforcement learning methods. The effectiveness and adaptability of algorithms that use reinforcement learning in empowering robots to move around and carry out activities in complex and unexpected environments have been proven by this study. Robots can effectively investigate their surroundings, pick up knowledge from encounters, and gradually improve their decision-making abilities by using reward signals to direct learning. Additionally, the research's conclusions have ramifications for several practical uses, such as automation in industries, service robotics, and driverless cars. Intelligent robots' capacity to navigate and adjust to shifting conditions on their own offers the potential to improve production, safety, and efficiency in a variety of settings. To fully realize the potential for intelligent machines in ever-more complex and dynamic environments, further study and development in this field will be necessary to enhance and maximize self-navigating and control computations, solve issues with adaptability and generalizations, and more.

Subsequent investigations could concentrate on creating increasingly complex sensor fusion algorithms that more successfully combine data from diverse sources including radar, LiDAR, and visual cameras. The robot's ability to see and make decisions in congested or dynamically changing settings may be enhanced by this integration. Navigational judgments could be greatly improved by incorporating machine learning algorithms that use historical data to forecast future environmental situations. Deep learning techniques for predicting possible impediments and human movement patterns in indoor environments could be investigated further.

6. Project information. Basic research project of Liaoning Provincial Department of Education in 2023: Design and research of intelligent positioning and control system for quadruped robots based on big data clustering (JYTMS20230321).

REFERENCES

[1] K. Almazrouei, I. Kamel, and T. Rabie, *Dynamic obstacle avoidance and path planning through reinforcement learning*, Applied Sciences, 13 (2023), p. 8174.

[2] M. Caruso, E. Regolin, F. J. Camerota Verdù, S. A. Russo, L. Bortolussi, and S. Seriani, *Robot navigation in crowded environments: A reinforcement learning approach*, Machines, 11 (2023), p. 268.

[3] X. Chen, S. Liu, J. Zhao, H. Wu, J. Xian, and J. Montewka, *Autonomous port management based agv path planning and optimization via an ensemble reinforcement learning framework*, Ocean & Coastal Management, 251 (2024), p. 107087.

[4] V. D. Cong et al., *Path following and avoiding obstacle for mobile robot under dynamic environments using reinforcement learning*, Journal of Robotics and Control (JRC), 4 (2023), pp. 157–164.

[5] J. Escobar-Naranjo, G. Caiza, P. Ayala, E. Jordan, C. A. Garcia, and M. V. Garcia, *Autonomous navigation of robots: Optimization with dqn*, Applied Sciences, 13 (2023), p. 7202.

[6] W. Fei, Z. Xiaoping, Z. Zhou, and T. Yang, *Deep-reinforcement-learning-based uav autonomous navigation and collision avoidance in unknown environments*, Chinese Journal of Aeronautics, 37 (2024), pp. 237–257.

[7] W. Huang, Y. Zhou, X. He, and C. Lv, *Goal-guided transformer-enabled reinforcement learning for efficient autonomous navigation*, IEEE Transactions on Intelligent Transportation Systems, (2023).

[8] S.-L. Jeng and C. Chiang, *End-to-end autonomous navigation based on deep reinforcement learning with a survival penalty function*, Sensors, 23 (2023), p. 8651.

[9] W. Li, M. Yue, J. Shangguan, and Y. Jin, *Navigation of mobile robots based on deep reinforcement learning: Reward function optimization and knowledge transfer*, International Journal of Control, Automation and Systems, 21 (2023), pp. 563–574.

[10] D. Ma, X. Chen, W. Ma, H. Zheng, and F. Qu, *Neural network model-based reinforcement learning control for auv 3-d path following*, IEEE Transactions on Intelligent Vehicles, (2023).

[11] E. E. Montero, H. Mutahira, N. Pico, and M. S. Muhammad, *Dynamic warning zone and a short-distance goal for autonomous robot navigation using deep reinforcement learning*, Complex & Intelligent Systems, 10 (2024), pp. 1149–1166.

[12] S. Na, T. Rouček, J. Ulrich, J. Pikman, T. Krajník, B. Lennox, and F. Arvin, *Federated reinforcement learning for collective navigation of robotic swarms*, IEEE Transactions on cognitive and developmental systems, 15 (2023), pp. 2122–2131.

[13] F. Naseer, M. N. Khan, and A. Altalbe, *Intelligent time delay control of telepresence robots using novel deep reinforcement learning algorithm to interact with patients*, Applied Sciences, 13 (2023), p. 2462.

[14] J. E. Sierra-Garcia and M. Santos, *Combining reinforcement learning and conventional control to improve automatic guided vehicles tracking of complex trajectories*, Expert Systems, 41 (2024), p. e13076.

[15] Q. Sun, L. Zhang, H. Yu, W. Zhang, Y. Mei, and H. Xiong, *Hierarchical reinforcement learning for dynamic autonomous vehicle navigation at intelligent intersections*, in Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023, pp. 4852–4861.

[16] Z. Sun, Y. Fan, and G. Wang, *An intelligent algorithm for usvs collision avoidance based on deep reinforcement learning approach with navigation characteristics*, Journal of Marine Science and Engineering, 11 (2023), p. 812.

[17] T. Wang, V. Dhiman, and N. Atanasov, *Inverse reinforcement learning for autonomous navigation via differentiable semantic mapping and planning*, Autonomous Robots, 47 (2023), pp. 809–830.

[18] Z. Xu, B. Liu, X. Xiao, A. Nair, and P. Stone, *Benchmarking reinforcement learning techniques for autonomous navigation*, in 2023 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2023, pp. 9224–9230.

[19] Y. Xue and W. Chen, *Multi-agent deep reinforcement learning for uavs navigation in unknown complex environment*, IEEE Transactions on Intelligent Vehicles, (2023).

[20] C. Yan, G. Chen, Y. Li, F. Sun, and Y. Wu, *Immune deep reinforcement learning-based path planning for mobile robot in unknown environment*, Applied Soft Computing, 145 (2023), p. 110601.

[21] Y. Yin, Z. Chen, G. Liu, and J. Guo, *A mapless local path planning approach using deep reinforcement learning framework*, Sensors, 23 (2023), p. 2036.

Edited by: Rajkumar Rajavel
Special issue on: Cognitive Computing for Distributed Data Processing and Decision-Making in Large-Scale Environments
Received: Jan 5, 2024
Accepted: Jul 6, 2024