



OPEN Intelligent mobile robot navigation in unknown and complex environment using reinforcement learning technique

Ravi Raj[✉] & Andrzej Kos[✉]

The usage of mobile robots (MRs) has expanded dramatically in the last several years across a wide range of industries, including manufacturing, surveillance, healthcare, and warehouse automation. To ensure the efficient and safe operation of these MRs, it is crucial to design effective control strategies that can adapt to changing environments. In this paper, we propose a new technique for controlling MRs using reinforcement learning (RL). Our approach involves mathematical model generation and later training a neural network (NN) to learn a policy for robot control using RL. The policy is learned through trial and error, where MR explores the environment and receives rewards based on its actions. The rewards are designed to encourage the robot to move towards its goal while avoiding obstacles. In this work, a deep Q-learning (QL) agent is used to enable robots to autonomously learn to avoid collisions with obstacles and enhance navigation abilities in an unknown environment. When operating MR independently within an unfamiliar area, a RL model is used to identify the targeted location, and the Deep Q-Network (DQN) is used to navigate to the goal location. We evaluate our approach using a simulation using the Epsilon-Greedy algorithm. The results show that our approach outperforms traditional MR control strategies in terms of both efficiency and safety.

As Artificial Intelligence (AI) technology rapidly advances, MRs are evolving and becoming more proficient at performing tasks that were previously too complex and hard, including space exploration, heavy-weight carriage, security, and many more. By integrating AI and robotics, robots might become intelligent and autonomous, performing tasks without assistance from humans¹. MRs are becoming more proficient by virtue of AI technology and can be employed in a wide range of domains, including defense, freight, infrastructure development and inspection, postal delivery, security, healthcare, customer support, and more. MR utilizes the sensors along with additional technologies to recognize its surroundings and execute its assigned assignment². The diversity of fundamental problems, which extend from difficult computational challenges to others that are exceptionally logical, is certainly among the most significant and fascinating features of MRs that allow them to solve these problems easily³. To complete a predetermined work, the MR generally goes through three phases: perception (sensing), path planning and synthesis (process), and motion (action). Finding the optimal routes to reach the target while avoiding obstacles is an important step in the navigation of MRs⁴. For a navigational system, one of the most important factors is determining how long and how much energy a robotic device needs to travel a given course^{5,6}. Since artificial intelligence (AI) approaches are accurate for generating findings, numerous researchers are employing them to automate the trajectory formulation method⁷. One of the most challenging areas of robot development continues to be path modeling. One of the main driving forces behind the analysis and study of mobile robots is the algorithm's broad range of applications and the complex nature of its design and development⁸. For researchers, the most important part of the development of MRs for implementation in the real world is how they operate in tropical environments.

The capability of autonomous learning is not present in the conventional MR navigation technique. MRs must be able to navigate unknown areas autonomously while avoiding both static and moving obstacles. Classical and heuristic approaches are both utilized for developing the operating technique of MRs. Traditional approaches can become tiresome and may stop producing the best results whenever environments become more complex. Nowadays, researchers choose heuristic methods more and more because of the extent to which they imitate the way humans acquire behavior⁹. Among the most well-known heuristic methods is reinforcement learning (RL), which is often used in MR navigation. The observed rotation angle around obstacles is refined

Faculty of Computer Science, Electronics, and Telecommunications, AGH University of Krakow, Aleja Adama Mickiewicza 30, Krakow 30-059, Poland. ✉email: raj@agh.edu.pl

using the RL-based controller¹⁰. The RL approach allows MR to identify the route based on past behavior. The MR, additionally known to be an agent, recognizes the environment, makes a choice, and gets rewarded or penalized in accordance with the environment, allowing the successful use of the RL approach. Whenever the MR ultimately receives a higher reward, it subsequently modifies its technique, and the agent might acquire various behavioral patterns through the development of an extensive number of rules¹¹.

RL has received interest from researchers for being able to overcome the fundamental navigation challenges faced by MRs, because of its powerful illustration and capacity for learning from experience. Experts have developed two distinct learning techniques, including Monte Carlo and Temporal-Difference (TD) learning because full flexibility of data and substantial use of memory are essential in programming with variables that aren't feasible. The Bellman equations and the Markov decision process (MDP), among other concepts, were integrated with TD learning to create the Q-learning approach in¹². After that, the RL technique made immense advancements, and RL concepts are implemented in a range of real-world problems, especially MR navigation. Due to its strong visualizations and flexibility over experiences-based Q-learning, RL has attracted a lot of attention for its ability to overcome the fundamental navigation difficulty encountered by MRs. When mobile robots are used in crowded circumstances, they must be able to move safely and interact socially amid people and different barriers¹³. Some of the key issues with deep learning in robots are that it does not ensure the reliability or accuracy of a learned policy, which presents an additional barrier to learning-based collision prevention¹⁴. Since it is challenging to produce training data for physical robots in an efficient and secure way, and due to the quantity and quality of the training data acquired has a significant impact on robot learning efficiency, the collision-avoidance policy used in this study is trained via simulations instead of in the real environment.

As seen in Fig. 1, the conventional navigation strategy maps an unknown region using simultaneous localization and mapping (SLAM), determines MRs present location using an orientation localization technique, and employs a path planning mechanism to navigate it to its target. The conventional navigational framework's inability to control an unfamiliar and unpredictable setting stems from its reliance on high-precision global mapping, which is highly sensitive to sensor noise. The designated complexity of multimodal situations constitutes a challenge for the traditional DRL approach, where the computation stress increases substantially as the diversity of inputs rises. Through the training process of a deep neural network (DNN), the deep learning

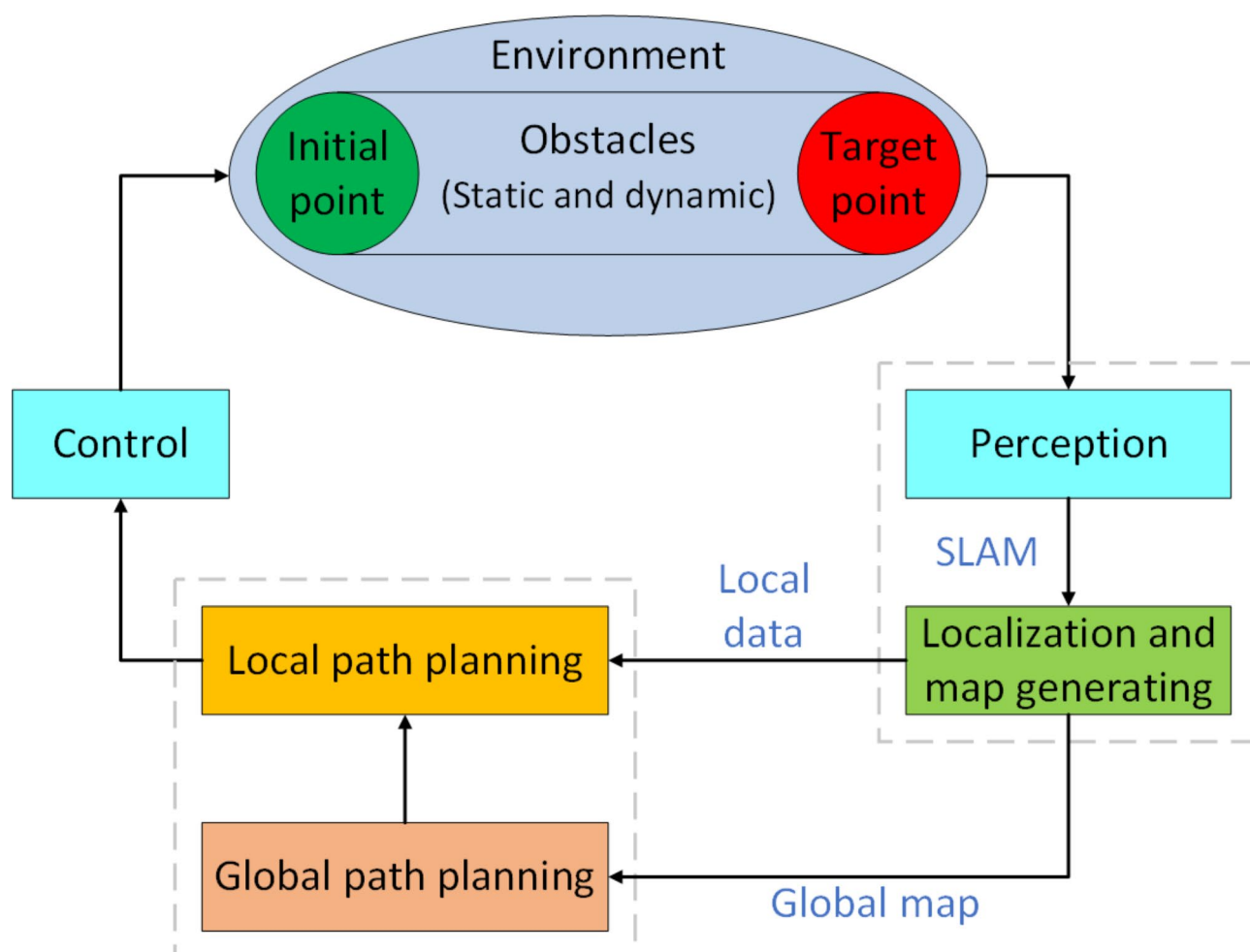


Fig. 1. Illustration of the MR navigation using traditional technique, based on¹⁵.

(DL) approach models functions that are nonlinear while learning the underlying rules and characteristics of the original data. Deep reinforcement learning (DRL) has been generated by fusing RL with a DNN. The traditional navigation method is currently being supplemented or replaced in several experiments by DRL-based MR navigation. Professionals from across the world have worked to find a solution to the navigation challenge for the last 20 years. Integrating many distinct techniques is a widely utilized method.

The phenomenal modeling capabilities of deep neural networks and the robust interaction, exploration, and self-learning features of RL have led to the widespread use of deep learning, particularly DRL-based techniques, enabling map-free navigation in recent times. In DRL, the agent automatically explores the surroundings by trial and error and then adapts the action based on the rewards received. Therefore, regardless of previous experience, including maps, autonomous navigation might be conducted with DRL. Deep Q-network (DQN), introduced by¹⁶, is an early method to integrate DL with DRL, and while it considerably increases the practical skills of DRL, the actions taken by this technique are only discrete. In¹⁷, a novel learning framework for DRL is suggested that drastically reduces the time of training. This study is enhancing the capability of visual navigation with a camera-equipped robot. The navigation difficulties can be separated into multiple manageably simple assignments using a distributed training strategy using the DRL that is proposed in¹⁸ to ensure the robot might successfully navigate in a substantial and complicated interior region without requiring the use of a map.

The remaining article is divided into the following ways: Section “[Literature survey](#)” contains a literature survey; Section “[Background information](#)” describes the contribution of this research; Section “[Theoretical methodology](#)” provides background information and proposed research techniques; Section “[Simulation and experimental analysis](#)” describes the theoretical concept of the proposed approach; Section “[Main contributions of this research](#)” discusses simulation and experimental analysis; and Section “[Conclusion and future perspectives](#)” provides the conclusion and future research trends of this research.

Literature survey

In recent years, many researchers have focused on the development of an effective framework based on the DDRL technique for the navigation of MR in a partially or fully unknown environment. Singla et al.¹⁹ explain a technique for allowing an unmanned aerial vehicle (UAV) quadrotor using a monocular camera to avoid obstacles by itself within unplanned and unexplored indoor environments. This study implements a memory-based DDRL for the avoidance of obstacles for UAVs, considering partial information about the environment. Li et al.²⁰ address the issue of automated exploration in a completely new environment, which is crucial when using a robotic system for various social activities. To improve the functional flexibility of the robot system, this study designs a generic exploring architecture by breaking down the methods of exploration in planning, decision-making, and mapping categories. This research suggests a DDRL-based decision technique that relies on this structure, which makes use of a DNN to learn an exploration approach considering the partial map.

Cai et al.²¹ propose a Linear Temporal Logic (LTL) formula-based deep policy gradient technique for directing a robot with uncertain dynamics working in a crowded environment. This paper presents a unique route-planning-guided reward system that incorporates sampling-based methodologies for successfully completing target-reaching tasks and tackles the sustainability issue of exploration in training. Yan et al.²² suggest a multi-robot navigation architecture that uses option-based hierarchy DDRL, enabling quick and secure navigation. Two control schemes are included in this architecture: a lower-level version that utilizes sub-policies to produce actions and a higher-level version that autonomously develops a reliable and steady behavior selection policy. Ali et al.²³ present a technique to utilize DDRL to enable autonomous exploration of unfamiliar environments. In this study, the environment is explored using external and curiosity-driven reward mechanisms. By anticipating future states, the curiosity-based reward mechanism encourages an agent to search for previously unexplored areas, while the external incentive function helps the agent avoid collisions.

Xue et al.²⁴ focused on the challenge of enabling efficient UAV navigation in large-scale unknown regions involving many obstacles utilizing DDRL techniques. In this study, a UAV through perception-constrained navigation is represented as a partially observable MDP and an actor-critic-based rapid recurrent stochastic valued gradient method has been developed for a real-time solution. Wang et al.²⁵ present a hierarchical navigation system that integrates model-based control with DRL-based perception. A DDRL network built using the Long Short-Term Memory (LSTM) and Soft Actor-Critic (SAC) algorithms can be trained to estimate 2D lidar inputs, goal locations, and the state of robots into a set of local checkpoints that are ideal in terms of avoiding collisions. Zeng et al.²⁶ provide a unique DDRL technique that can maneuver non-holonomic robots into an uncertain, unfamiliar environment containing moving obstacles while maintaining constant control. Simulation tests demonstrate that the suggested strategy, by continuously issuing acceleration instructions, can achieve excellent robotic maneuvering in unfamiliar and difficult settings when compared to traditional approaches.

Zhu et al.²⁷ provide a hierarchical DRL architecture for quick and secure navigation that has an excellent rate of sampling and sim-to-real transferring skill: The robot might follow toward the goal point while maintaining an adequate distance from obstacles thanks to the minimal DRL policy; additional a high degree DRL policy is added for further enhanced navigational reliability. With the goal of minimizing state space and preventing inadequate reward, this method chooses a waypoint that is on a trajectory that leads the robot toward the desired destination as the subgoal. Furthermore, the path is created using a regional or global map, which may greatly enhance the suggested DRL framework's adaptation capacity, security, and sampling accuracy. To increase mobility accuracy and decrease action space, a target-directed illustration of the action space might additionally be developed from the subgoal. Xue et al.²⁸ propose a novel technique for guiding the multi-UAV's navigational activities depending upon the gradient algorithm of the depth deterministic policy, which is known as multi-agent recurrent deterministic policy gradient (MARDPG). To be more precise, every critic network trains centralized, enabling every UAV to determine the policies of different UAVs, and overcoming the delayed

convergence issue brought due to the uncertain surroundings. The requirement for resources to communicate between UAVs is eliminated by distributed operation. In critic's network, sharing variables speeds up training.

Background information

To perform autonomous navigation while navigating through an unfamiliar area, MR acquires sensor data and simultaneously creates a map of its environment. The MR exploration algorithm is required to generate a map that looks more like the real world. In conventional navigation approaches, the MR creates an environment map by itself and finds its own location. DL and RL have progressed throughout the decades, and with their assistance, numerous end-to-end control techniques have been created. These technologies integrate sensor information and the environment into deep neural networks (DNN) as inputs and outputs for MR control. One major benefit of end-to-end approaches is their conciseness²⁹. Despite this, there is a significant lag before using the method on an actual MR. Since end-to-end learning techniques require extensive trial and error, many academics have implemented this method of learning in simulation environments. Since the MR begins by acquiring the ability to navigate to discover a new environment, the algorithm takes a while to converge. It cannot be assured that the mapping that a DNN learns in a simulation setting will have a better generalization for the actual world because the two environments are not exactly similar. The relationship between surroundings and an agent might be represented by the MDP. MDPs apply several variables, such as the surroundings, agent action, and rewards, to decide what the optimum next action of the framework will take. An MDP describes an iterative decision-making problem in which an agent must select an array of actions to maximize reward-based criteria. Equation (1) is the abbreviation for an MDP (M)^{30,31}.

$$M = (S, A, T, R) \quad (1)$$

Where S is the number of finite states, A is a set of possible actions, $T(s, s') = T_r(s_{t+1} = s' | s_t = s, a_t = a)$ is the transition probability that action a takes for state s at time t that will forward to state s' at time $t + 1$, and $R(s, s')$ is the possible reward value after the transition between state s to state s' because of action a .

To enable MRs to avoid obstacles and navigate safely in previously unknown complex environments, we developed an RL-based regional planner that translates acquired sensory information directly to robotic behaviors. Also, we divide the MR exploration strategies into three different parts: planning, mapping, and decision. Thus, as each component acts independently from the others, we can create the planning techniques for various MRs using a single decision segment. The MR's abilities to navigate effectively in a familiar area without requiring any training are guaranteed by the planning algorithms. Utilizing conventional navigation techniques, the decision section concentrates on an effective exploration approach. Figure 2 depicts the interaction between all three parts that compose the exploratory architecture. The decision element gets information from the mapping element and forwards it through the planning element. The complete MR exploration technique is divided into subsections in the following ways:

Planning strategy

Finding a plausible path from the present location to the desired location is the main objective of this section. The planning strategy is abbreviated as Eq. (2)²⁰:

$$\Gamma_{t:t+T} = f_{plan}(l_{0:t}, g, m_t) \quad (2)$$

Where the planning trajectory is denoted by $\Gamma_{t:t+T}$, f_{plan} is the planning function, positions of MR from time 0 to t are denoted by $l_{0:t}$, the target point is denoted by g , and m_t is the map created in the environment at time t .

Local planning and global planning techniques are two of the most extensively studied frameworks for several strategies that carry out the planning function. For turning the route into a trajectory and modifying it according to the real-time data collected from sensors, the objective of this research is to find the shortest route over the map between the current location and the target location.

Mapping strategy

The mapping section, which is additionally referred to as simultaneous localization and mapping (SLAM), generates the MR position and environment mapping at every moment by systematically processing the sensor data. The SLAM function is expressed as in Eq. (3)²⁰:

$$m_t, l_t = f_{map}(l_{t-1}, m_{t-1}, c_t, o_t) \quad (3)$$

Where control and observation are c_t and o_t consecutively, f_{map} is the mapping function, the location of MR is denoted by l_{t-1} at time $t - 1$, and m_{t-1} is the map created in the environment at time $t - 1$.

The two main types of SLAM techniques are filter-based and graph-based. Filter-based techniques modify the predicted value of both landmark and MR positions using a Kalman filter as well as the expanded Kalman filter. The graph-based techniques use nonlinear optimization for predicting the MR location by building a position graph using both control and observation data. We use a graph-based approach in this study because it is easy to apply and performs well in most scenarios. The graph-based technique can minimize errors and reliably provide the obtained answer when compared with the filter-based technique³².

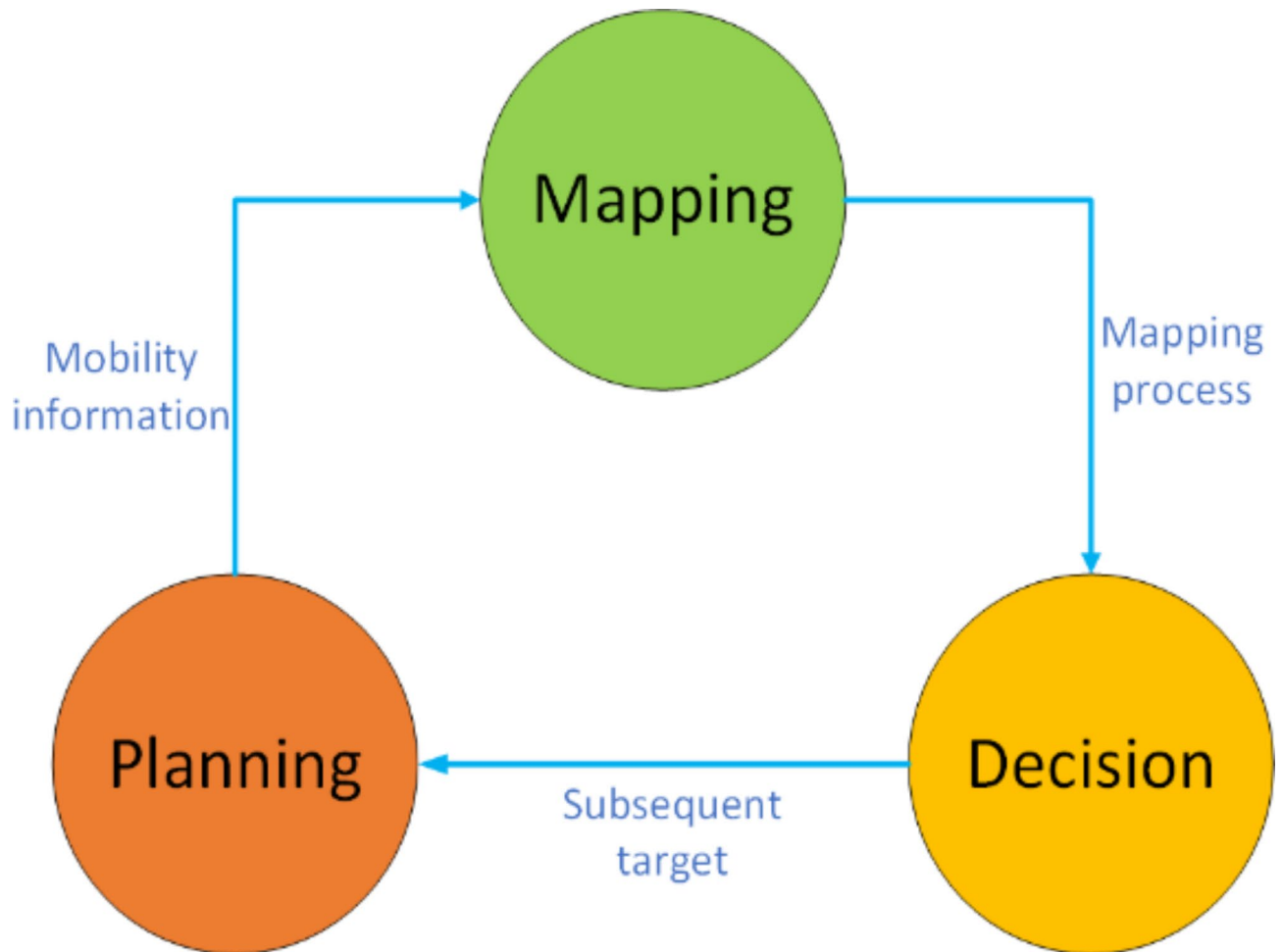


Fig. 2. Relationship between planning, mapping, and decision.

Decision strategy

This component selects the target point, or the location, that the MR needs to reach based on past observations. The map for the surrounding area generated by the mapping section serves as the input for this work. The decision strategy can be expressed as in Eq. (4)²⁰:

$$g_t = f_{dec}(l_{0:t}, m_t) \quad (4)$$

Where the goal point at time t is denoted by g_t , the position of MR from time 0 to t is represented by $l_{0:t}$, the surrounding map generated at time t is m_t , and the decision function is f_{dec} .

These techniques lead to significant reliance on knowledge, as they are necessary to build extensive rules for complicated contexts. The primary goal of this study is to determine the simplest and most common function f_{dec} . Our approach addresses more complicated scenarios and learns the workable technique by trial and error, as compared to stacking the rules for study.

Theoretical methodology

Mobile robots must have the ability to discover how to travel independently in unfamiliar interior scenarios while avoiding both dynamic and static obstacles. By finding the ideal Q-value function that produces the best outcomes across every situation, RL seeks to educate the agent on how to act in a strange setting. After choosing action over every state, the agent adopts rewards from the surroundings to modify Q-values towards optimality convergence. For an RL algorithm, there is a conflict between exploring and exploiting when the surroundings are unknown. By choosing the course of action with the highest assessed value, the agent makes use of its existing understanding. If the agent chooses any of the remaining actions, on the other hand, it suggests that the agent is exploring to refine and enhance its assessment of the act's values.

Developing a surroundings map that is as close to replicating the real-world scenario as possible is the aim of MR Exploring. In Fig. 3, the agent, and surroundings within the DRL-based system of navigation are shown collaborating with each other. The DRL agent drives toward the intended destination while avoiding static and dynamic obstacles, substituting the localizing, map-generating, and planning of local paths that make up the traditional navigation approach. The global route planning element in Fig. 3 establishes a series of landmarks

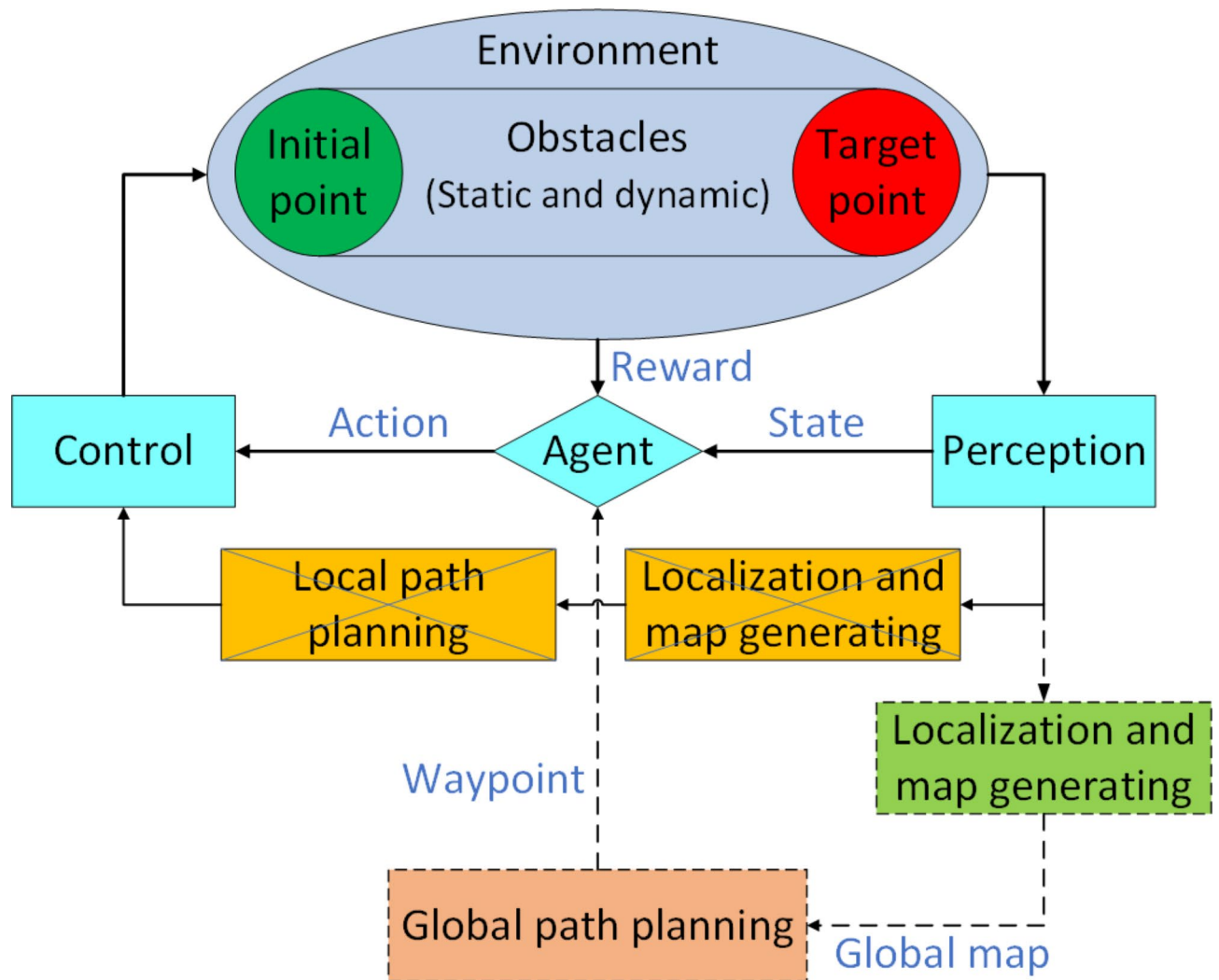


Fig. 3. Illustration of the MR navigation technique using DRL, based on¹⁵.

to act as transitional locations of references throughout DRL-based MR movement, enabling an integrated navigation system for carrying out extended-distance movement through challenging terrain.

The MR can navigate through the environment and avoid obstacles while achieving its goal on time in this study. Now, we explain the objective function and DRL for the proposed technique in detail.

Objective function

When exploring the area, MR is meant to take the shortest route possible to conserve battery life. Next, we provide the subsequent objective function in Eq. (5)²⁰:

$$Z(\hat{m}, y_{t=0:T}) = D(y_{t=0:T}) + \min_{u_{t=0:T}} \|m - \hat{m}\|^2 \quad (5)$$

Where the real map is m , the estimation map is $\hat{m}$, and $D()$ is the distance between the starting point to the target point.

However, it is challenging to regulate the objective function and determine an optimal approach because the real map is unavailable because of the unknown surroundings. Our utilization of the occupancy grid map to depict the environment is motivated by the application of the Shannon entropy during the estimation of the exploration³³. The cross-grid occupied probability is $p(m_{i,j})$ for column i and row j . The mapping entropy is expressed in Eq. (6)²⁰:

$$E(m) = - \sum_i \sum_j p(m_{i,j}) \log p(m_{i,j}) \quad (6)$$

There are three states that can occur on every grid in the map: occupied, free, and uncertain. This objective function assesses the constructed map's uncertainty. The proposed DRL technique used for this study is explained

in subsection A; the strategy of action space and state space for the experiment is explained in subsection B; and the proposed model is illustrated in subsection C of this section.

Mobile robot learning

A standard robotics issue with multispectral applications across many industries is MR navigation. Numerous industries, including healthcare, robotics, smart surveillance systems, human-computer interfaces, and several more, employ human activity recognition extensively^{34,35}, or RL. Rather than attempting an extensive investigation for a space this vast, we just address the issue of indoor navigation. Getting to a target on a collision-free path in the least amount of time is often the goal of navigation. Perhaps the most crucial ability that an autonomous MR must possess is the practice of localization learning, as determining the robot's position is necessary to decide what to do next. In various mapping-based robot systems of navigation, an environment remains available for the robot since it was originally specified in a metric map. Here, MR learns its surroundings using the RL algorithm using DQN and Double Deep Q- Network (DDQN) techniques. RL algorithms enable MR to learn the environment without predefined mapping. Nowadays, most RL-based techniques are used by robotics to perform navigation tasks that establish a mapping link between the control strategy and raw sensor inputs.

To learn how to select either discrete or continuous navigation activities for movement utilizing internal sensory input in an area with either a known^{36–38} or unidentified³⁹ flat surface, DRL has been the primary learning approach used to teach robots to navigate. However, most of these initiatives concentrate on navigation in stationary environments, making them impractical for most real-life applications as different types of moving items need to be considered.

Deep Q-networks

A model-free RL technique called Q-learning is implemented to determine the worth of some action in each situation. It can address issues with stochastic transitioning and reward despite the need for modifications, and it doesn't demand environment modeling. To simulate the impacts of receptive fields, the DeepMind approach uses a deep convolutional neural network (DCNN) comprising layers of convolutional filtering. The trained network might achieve a level comparable to the level of human beings shortly after playing 49 games, according to research done in 2015 and originally published in Nature by Mnih et al.¹⁶. DQN is dependent upon Q-learning, where the action value function is represented by DCNN, and the reward feedback received from the game is responsible for training the network. Gradient descent is used for updating the neural network's parameters. The DQN's loss function is indicated in Eq. (7)¹⁵:

$$L(\theta_i) = F \left[\left(R + \lambda \max_{A'} Q(S', A'; \theta_i^-) - Q(S, A; \theta_i) \right)^2 \right] \quad (7)$$

Where the discount factor is denoted by $\lambda \in [0,1]$, S is the state, R is the reward, A is the action, $S' = S_{t+1}$, $A' = A_{t+1}$, and θ_i is the parameter for the current Q-network $Q(S, A; \theta_i)$ is replicated to θ_i^- for the goal Q-network $Q(S', A'; \theta_i^-)$ at each step of n -time.

Double deep Q-networks

Although DQN's introduction aided in the general adoption of DRL, it has several drawbacks, one of which is that it overestimates the action value function. Van Hasselt et al.⁴⁰ noted that overestimating occurs for the value function whenever the DQN computes the temporal-difference (TD) error and an identical Q network is used to choose actions and compute value functions; therefore, they have suggested the DDQN technique. For the goal Q function, a dual network architecture is used by DDQN, where the goal Q network estimates the specified optimal action and the current Q network is responsible for the selection of optimal action. The action decision and assessment of policy activities are divided into two sets of variables, which lowers the chance of overestimation. The DDQN's loss function is indicated in Eq. (8)¹⁵:

$$L(\theta_i) = F \left[\left(R + \lambda Q(S', \arg \max_{A'} Q(S', A'; \theta_i^-) - Q(S, A; \theta_i) \right)^2 \right] \quad (8)$$

Reinforcement learning-based navigation

DL is basically a set of frameworks enabling non-linear function estimations by using DNNs to abstract characteristics that are buried in inputs, which is now frequently employed in the field of robotics, including indoor navigation and robot manipulation. Aerial vehicles, autonomous vehicles, and two- or three-dimensional moving ships are examples of MRs. Finding the best or least undesirable route between the starting point and a destination while avoiding obstacles is the purpose of MR navigation. We have limited our attention to a two-dimensional (2D) space navigation issue with the objective of simplifying this assignment. The mobile robot navigation problem essentially consists of obstacle avoidance and point-to-point (P2P) motion: (1) The target point's location in relation to the initial location is necessary to complete the P2P assignment⁴¹, and (2) The three types of obstacles that are susceptible to detection by cameras, ultrasonic sensors, laser rangefinders, and other sensors involve those characterized as static, dynamic, and physically consistent. A hallway or wall are examples of natural environmental structures that are referred to as physically consistent obstacles in this study. These obstacles create an interior or labyrinthine setting. With its significant representation and knowledge-based

learning skills, RL has received considerable interest as an alternative solution to the fundamental navigation problem faced by robotics systems⁴².

When utilizing the DRL technique for autonomy in navigation, the goal is to determine the optimal path of action for directing MR toward its destination by interacting with its surroundings. The conventional navigational system might be merged with or replaced by DRL-based navigation. Because RL utilizes a trial-and-error learning system, it is forbidden for MR to crash into surrounding obstacles during the physical training phase. Before being implemented in an actual MR for navigation and making decisions in real-time, the DNN is often trained in a simulated setting. The mechanism of agent-environment interaction in the DRL-based navigation system is depicted in Fig. 3. The DRL agent proceeds in the direction of the destination location while preventing dynamic, static, and basic physically constant obstacles. Subsequently substitutes both the localization and map-building modules along with the local route planning section of the standard navigation system. Nonetheless, the agent could encounter a local trap in a setting with excessively complicated, structurally persistent obstacles⁴³. The process of acquisition is continuous, with an agent gaining knowledge of the policy through interactions with its surroundings. Figure 4 illustrates how the agent determines what to do after determining the current situation. After carrying out the decision-made action, it gets feedback—that is, the reward—and monitors the subsequent state to decide on its subsequent course of action.

The global planning of paths section, as seen in Fig. 3, creates a sequence of touchpoints that serve as middle objective points for DRL-based navigation, which allows the combined navigational system to do long-range movement into a complicated structural domain. Depending on whether RL employs an explicit policy or not, it might be either active or passive. Only the set policy's effectiveness is the focus of passive learning. Considering the surroundings are unknown, the agent's main problem is that it remains unaware of both the transitional model and the reward function. To determine the expected outcome of every state, the agent uses the predefined policy to carry out a series of experiments. Active learning, as opposed to passive learning, also tries to learn how to behave in the environment. The agent needs to comprehend the transition model to be able to identify the best policy because it is ignorant about the surroundings and the policy. The operational scenarios and the RL algorithm's effectiveness are significantly determined by three essential components, including state space, action space, and reward function, of the DRL-based MR navigational framework.

State space

The initial point, objective point, and obstacles are the state-space configurations that are typically implemented. The MR's initial and target coordinates serve as the initial position and target position representations, respectively. Many studies transform global Cartesian positions into regional polar positions and then define the goal point location using the angle and distance with respect to MR. The state of an obstacle can be expressed as the velocity, location, and dimension of the dynamic obstacles (agent leveling), or it can be expressed as a state of sensor-level through treating sensor information, such as depth camera data, lidar/ultrasonic range data, or monocular camera images. The state space is denoted by S .

Action space

There are mainly three types of actions in the DRL based MR navigation, including discrete moving actions (turning left, turning right, moving forward, moving backward, and many more), motor speed commands (the

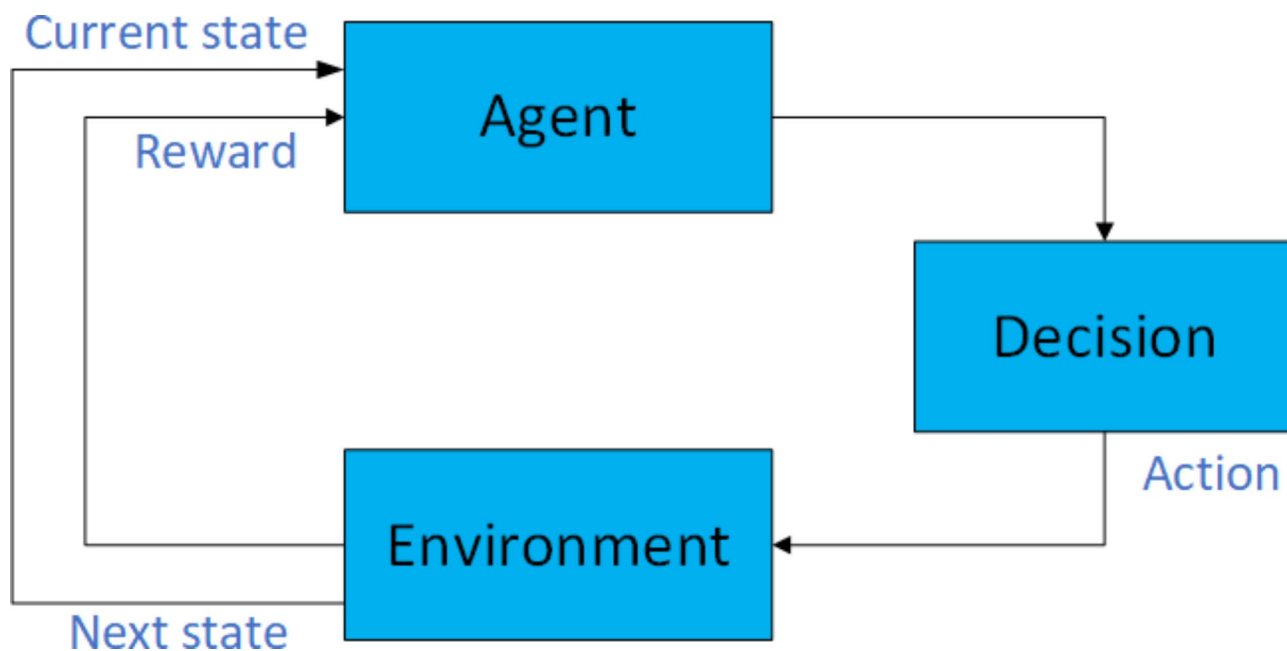


Fig. 4. Relationship between agent, decision, and environment.

required speed of the right and left motor), and continuous velocity commands (the angular and linear velocity of the MR). The sensor-level states might be used to accomplish end-to-end commands through motor velocity commands, although the corresponding training is substantially more challenging. Proportional-integral-derivative (PID) or equivalent lower-level controllers are often needed to generate control of mobility commands for MR and operate the MR to accomplish the required motion for discrete motions and continuous velocity instructions. The action space is denoted by A . Robots move and change directions by using several components including actuators, control algorithms, and sensors.

Reward function

The RL agent is trained to do a task using the reward function. The rewards are given positive or negative only, when MR hitting obstacles or reaching the goal, which means this reward is small. The agent cannot rapidly converge when rewards are few. Most research employs dense reward-shaping techniques to increase training efficiency. A positive reward is always given to MR when it reaches the target and is close to the target point. When an MR reaches or collides with obstacles, it usually gets a negative reward. A time-stepping penalty represents a penalty that MR receives at every step in order to motivate it to travel toward the goal more quickly. The reward function is denoted by R .

Simulation and experimental analysis

Researchers are concentrating on autonomous interaction with surroundings and self-learning methodologies, notably on RL systems, to enhance the ability to generalize while preventing manually gathering supervised information. The following is a description of the intended navigation challenge in this article: A mobile robot with scattered regional distance sensors provides all that is needed to go reliably to a goal controlled by constant guidelines over an initially unfamiliar area with broadly dispersed topography, such as lengthy hallways and dead corners. The MR can build an independent environmental model through exploration because, initially, it does not know anything about its surroundings. The dynamic and unpredictable nature of surroundings with dynamic obstacles adds significantly to their complexity. We implement into operation an RL-based regional plan that interprets observable sensory information into activities for secure navigation and avoiding obstacles in dynamic, previously unexplored areas. We add several frequent patterns to the basic RL approaches to allow navigation in complicated situations and improve performance. We also anticipate better navigation performance in more complicated situations, including alleyways, long hallways, and concave regions, because the memory section can think about several sensor perceptions instead of depending only on the present perception. Figure 5 shows a binary occupancy grid map containing different types of obstacles and the start and target positions for robot path planning, which is used for the simulation to demonstrate the starting and final positions for the navigation of MR.

The ability for autonomous learning is absent in the conventional robot navigation strategy. MRs must be able to navigate unknown areas independently and avoid static and dynamic obstacles. Navigation is the capacity of the robot to make decisions based on its internal skills and sensor data with the objective of achieving its goal locations as quickly and effectively as feasible, providing a partial understanding of its surroundings and a desired location or set of locations. A virtual robot is used for the simulated environment to carry out the chosen actions at every stage of the episodes. The training process for a neural network can be extrapolated into a dataset of accessible training instances, enabling the acquisition of viable answers for comparable conditions that might not have been observed in demonstrations. To assess this proposed approach, simulation tests are performed. The dimensions of an environmental mapping are 250 m by 250 m. The obstacles are dispersed arbitrarily around the surroundings. The robot's starting point coordinate is shown on the map as a red triangle, which is located at (10, 10), and the goal location is shown as a green circle, which is located at (240, 240). The robot's task is to begin at the starting point and choose the best route to take for the purpose of reaching the goal without hitting any obstacles. Table 1 illustrates the list of hyperparameters and their descriptions, which are used in this experiment.

The robot is sent to perform a navigation task within an unfamiliar and novel environment. The simulation's primary goal is to produce an ideal Q-value matrix. In contrast to the tests, the robot is initially unaware of the quantity, dimensions, and locations of the obstacles. It is also noticed that, although the pathways the robot selected may not have been the optimal routes since it did not completely understand its surroundings, they were nevertheless perfectly adequate and met standard reference. The algorithm for the navigation of MR based on RL is abbreviated in Table 2.

The process of training an MR involves placing it in several unpredictable scenarios, referred to as episodes. Every episode assumes a distinct goal direction. Every episode uses a variable quantity of obstacles as well. It is considered that some of these obstacles are stationary while others are considered to be dynamic. The more episodes that are utilized to train the robot, the more skilled it will become at maneuvering its dynamic surroundings⁴⁴. The proposed approach has been developed to operate in simulation environments. We are examining how the pathways differ from one another since they are not the most effective routes. Figure 6 shows four different environments for the navigation of mobile robots, where the starting and goal location is predefined. Blue curved lines represent the normal route for navigation while red curved line represents the optimal or shortest path.

The suggested system generates a state collection with dynamic variables such that, once it successfully navigates a specific situation, it might be effective within another, wherever the barriers and goals are situated at different positions. How to characterize the reward function—the way the environment reacts to the cognitive agent's actions—is another criterion for making decisions.

The number of successful learning episodes for every 50 episodes is shown in Fig. 7. This quickly increasing pattern of successful episodes in Fig. 7 with respect to the number of episodes demonstrated that the proposed

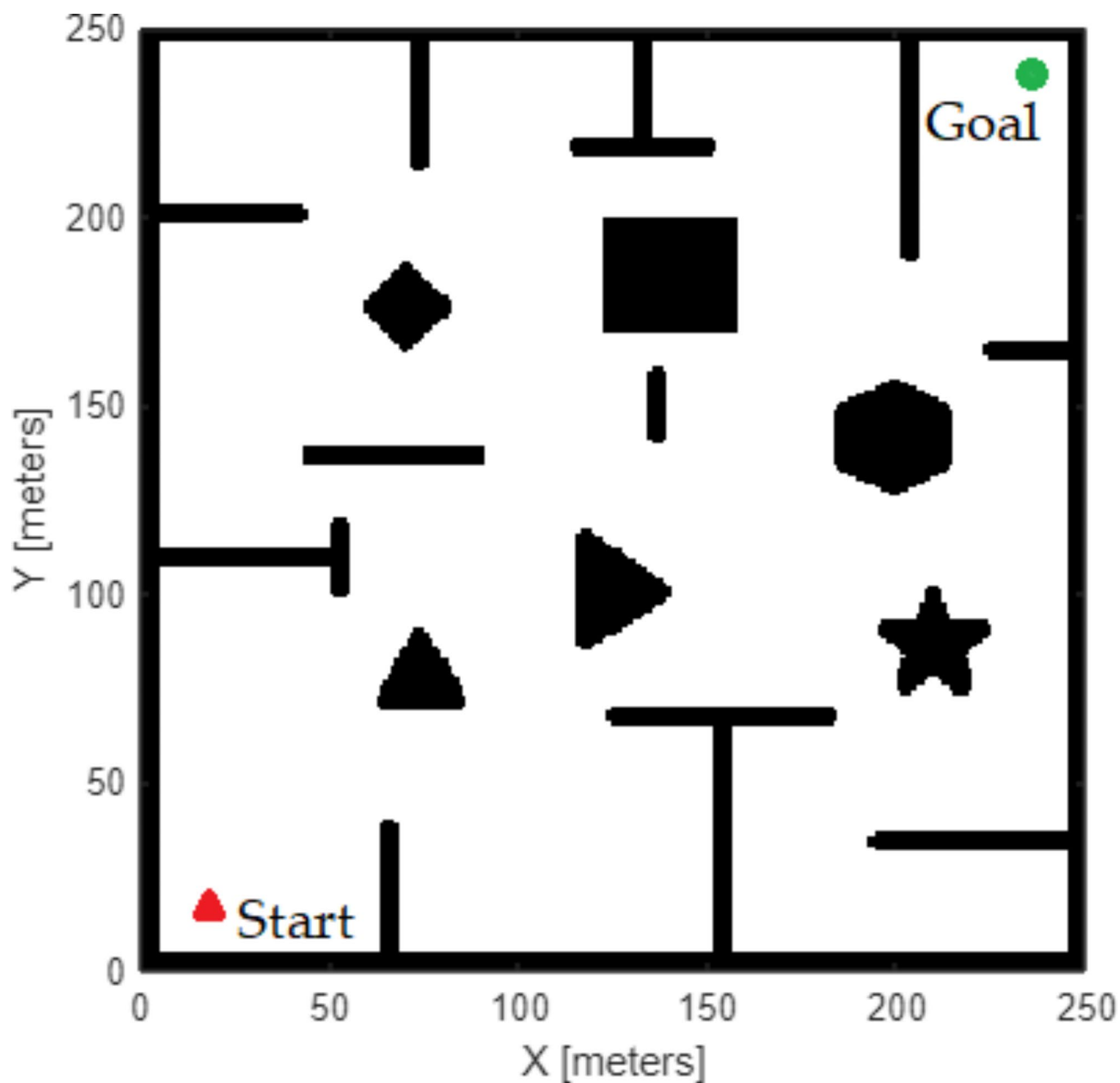


Fig. 5. Binary occupancy grid map for navigation tasks.

	Description
System configuration	
Python version	2.9.1
TensorFlow version	2.9.1
NumPy version	1.24.3
Learning rate	0.001
RAM	8GB
Activation function	ReLu
Gym version	0.26.2
Processor	Intel(R) Core (TM) i3-4005U

Table 1. List of hyperparameters.

Algorithm 1: Mobile robot navigation using RL	
1:	Load the weights of trained NN.
2:	Set the goal and create obstacles randomly
3:	Define the initial position and orientation of the mobile robot
4:	time←1;
5:	for all dynamic steps do
6:	Observe the property of the state and the current state
7:	if the current state is optimal state or non-optimal state then
8:	Stop the navigation;
9:	end if
10:	Estimate all activity Q-values using NN
11:	Pick the optimal action policy
	end for

Table 2. Algorithm for navigation based on RL.

DRL algorithm improved the robot’s intelligence. An epsilon-greedy algorithm is used in simulation. Figure 8 shows the variance graph for reward and epsilon during simulation. The score of reward and epsilon with respect to the number of episodes has been shown in Fig. 8.

Initially, the simulation carried out for the experimental studies only used the Adam optimizer algorithm. We evaluate the three algorithms various performances using the simulation environment. This is necessary to demonstrate that finite state space algorithms might be an effective choice for robotics and to provide a consistent quantitative assessment of the technique. We compared this approach by using three different optimizers including the Adam optimizer algorithm, SGD optimizer algorithm, and RMSprop optimizer algorithm to validate the proposed technique for 100 episodes. Figure 9 shows an illustration of the reward value comparison between the Adam optimizer, SGD optimizer, and RMSprop optimizer. As shown in Fig. 9, the Adam optimizer receives the highest reward during simulation.

We conducted relevant research and compared the findings with other optimization algorithms to validate the produced RL technique’s better sampling performance, speedy mobility, substantial target-reach success rate, and greater extrapolation with regard to diverse motion situations and surroundings. The proposed algorithm is compared with Proximal Policy Optimization (PPO) to analyze the effectiveness of our approach. PPO is an RL technique that trains the cognitive role of a computer agent or robot to complete challenging tasks⁴⁵. Figure 10 shows a comparative reward value assessment between the proposed approach algorithm and the PPO algorithm. The Q-learning algorithm approach outperforms the PPO algorithm as shown in the comparative graph.

Main contributions of this research

The proposed approach has several advantages when compared with the above techniques. First, it does not require explicit modeling of the environment or the robot’s dynamics. Second, it can adapt to changing environments and tasks without requiring manual tuning of control parameters. Third, it can learn from experience and improve its performance over time. This approach has a higher rate of accuracy compared to the traditional MR control strategy. However, most of the such DRL-based movement control strategies are primarily effective in plain indoor environments. To the best of our knowledge, only some research has successfully controlled MR in an unknown environment. We are presenting an end-to-end mobility control technique based on DRL to make it easier for MR to operate effectively in an unknown and complicated environment to get beyond the constraints of the earlier techniques that were discussed. The proposed approach is carried out in a simulation environment, so there is a possibility of variation in the real-world application scenario. The tracking challenge for non-holonomic robots in unpredictable circumstances with constant control of movement is the primary subject of this research. Our goal is to present a DRL method that can quickly and safely navigate through complicated settings while also having excellent training and adaptation capacity across different settings and robotic systems.

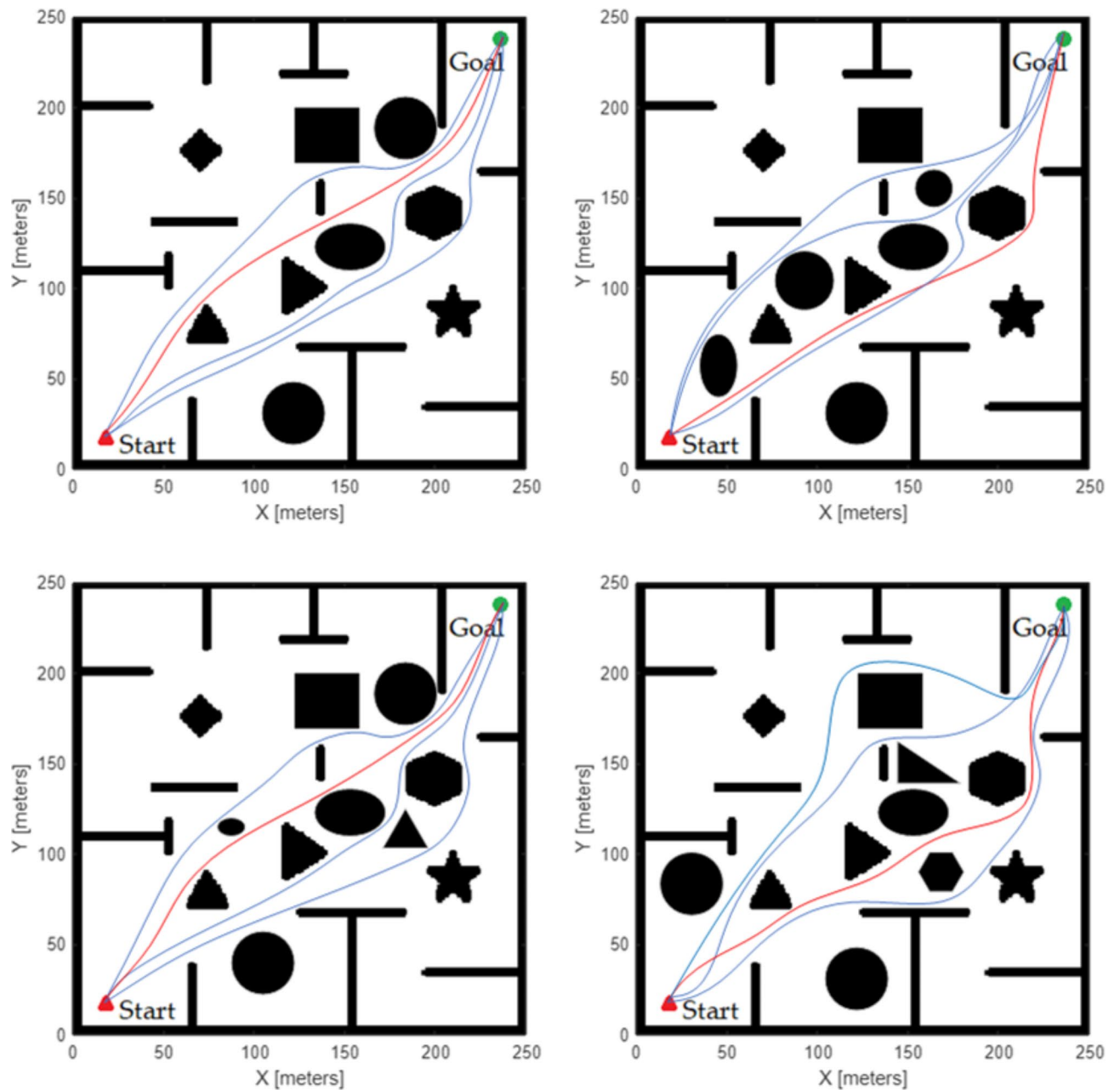


Fig. 6. Illustration of RL-based autonomous navigation results in various scenarios.

Conclusion and future perspectives

For continuous-state problems, sampling every possible solution is impractical, and in the absence of a specific instructor, RL algorithms are the preferred method in supervised learning for the DL domain. This paper presents a comprehensive methodology for DRL-based regional plan training and deployment in extremely complex and dynamic contexts for the navigation of MR. We have tested the proposed autonomous navigation approach for four different environmental settings, where each environment contains different obstacles shapes, and positions. The optimal path while avoiding obstacles between the start point to the goal point has been explored in the simulation. In conclusion, our approach shows promising results for intelligent MR control using DRL. In this approach, to carry out its collision-free navigation, the MR is using the DRL to gather background information about its surroundings from the sensor data. The proposed technique has been validated by using the Adam optimizer and a comparative analysis has been carried out using three different optimizers including Adam, SGD, and RMSprop optimizers. The proposed approach is a very simple but effective simulation system that demonstrated quick and effective training rates as well as the ability to transmit the frameworks to the actual robot. The outcomes demonstrated an apparent increase in the dependability and safety of navigating in complicated surroundings, but at the expense of effectiveness due to an extended exploration phase. Simulations



Fig. 7. Illustration of the number of successful learning episodes in simulation for every 50 episodes.

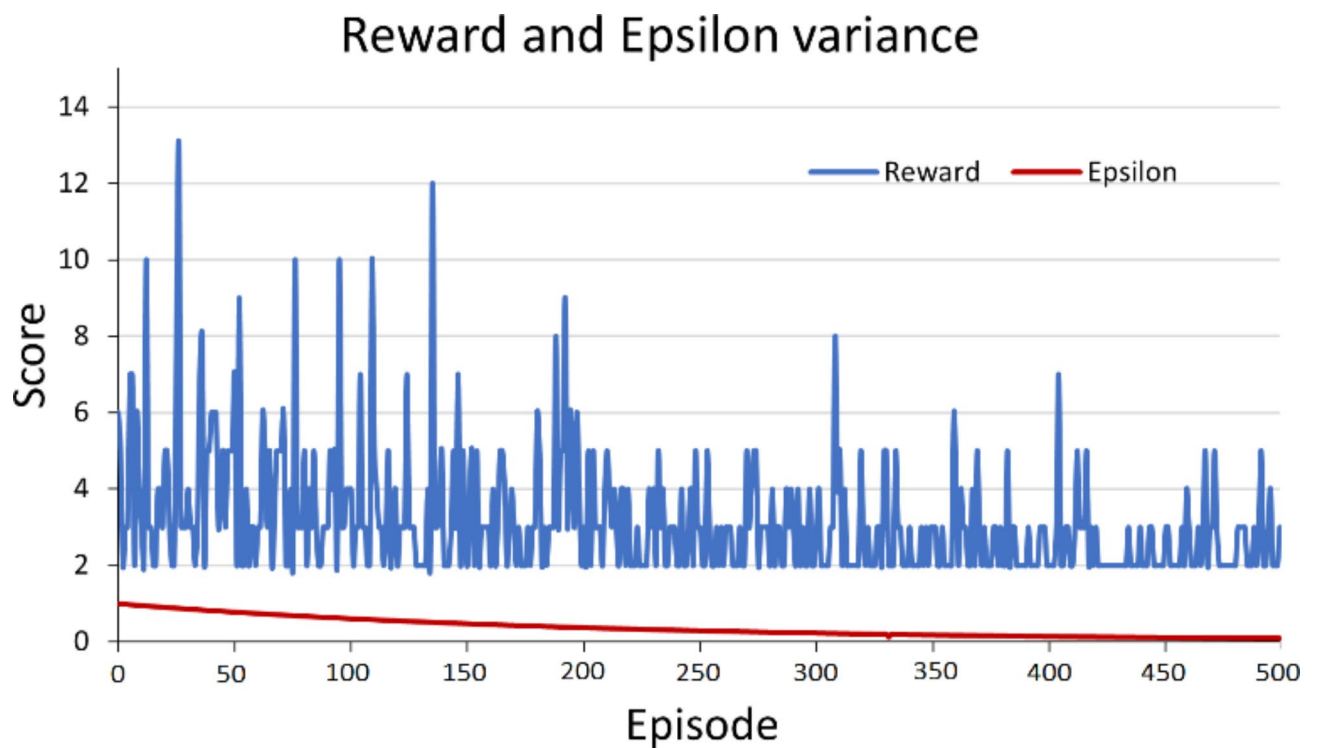


Fig. 8. Illustration of Reward and Epsilon variance score per episode during simulation.

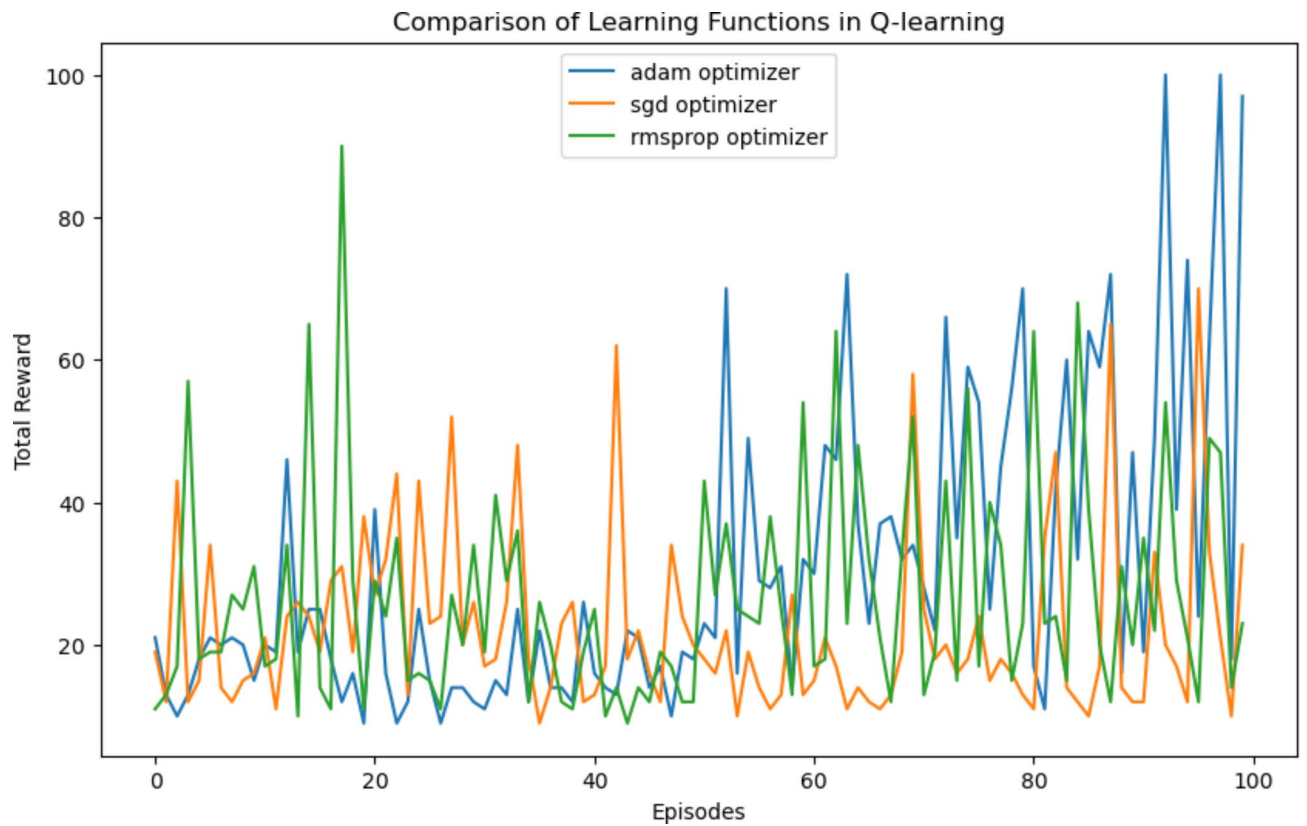


Fig. 9. Illustration of comparative graph for reward value when using Adam, SGD, and RMSprop optimizers in simulation.

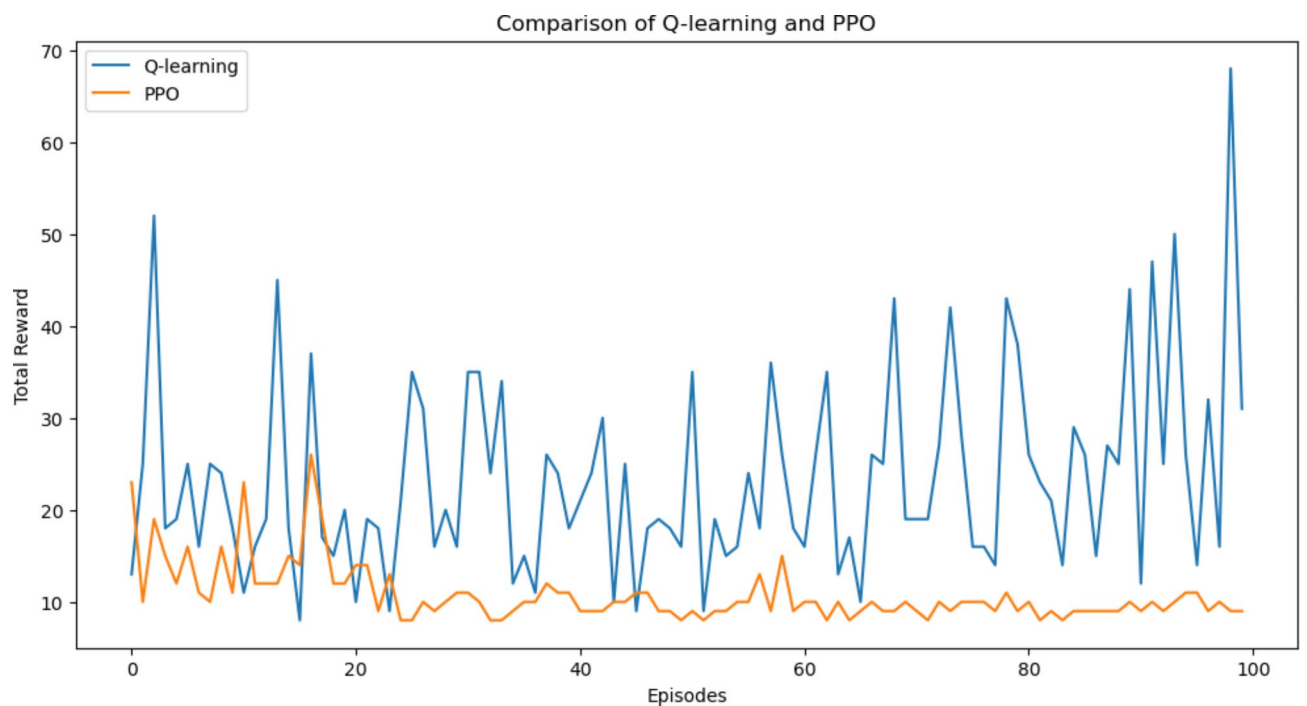


Fig. 10. Comparison between the reward of the proposed algorithm and the PPO algorithm.

demonstrated the effectiveness of our complete learning framework, as there is a significant likelihood that the robot might be able to navigate freely in an unfamiliar area.

The proposed approach has been delivered promising outcomes, it can continue to be improved to improve its learning and performance rate. To enhance the learning approach and raise the rate of accuracy of completely autonomous navigation in an unfamiliar environment, the reward function can be modified in future studies. The experiment carried out in this study is in a simulation environment, so the outcomes can be slightly different in the real-world implementation. Future work will focus on applying our approach to real-world MR systems and evaluating their performance in various applications.

Data availability

Correspondence and requests for materials should be addressed to the corresponding author on reasonable request.

Received: 16 March 2024; Accepted: 11 September 2024

Published online: 01 October 2024

References

- Raj, R. & Kos, A. A. Comprehensive study of mobile robot: History, developments, applications, and future research perspectives. *Appl. Sci.* **12**, 6951. <https://doi.org/10.3390/app12146951> (2022).
- Raj, R. & Kos, A. Artificial intelligence: Evolution, developments, applications, and future scope. *Przegląd Elektrotechniczny*. **2**, 1–13. <https://doi.org/10.15199/48.2023.02.01> (2023).
- Oommen, B., Iyengar, S., Rao, N. & Kashyap, R. Robot navigation in unknown terrains using learned visibility graphs. Part I: The disjoint convex obstacle case. *IEEE J. Robot Autom.* **3**, 672–681. <https://doi.org/10.1109/JRA.1987.1087133> (1987).
- Raj, R. & Kos, A. An optimized energy and time constraints-based path planning for the navigation of mobile robots using an intelligent particle swarm optimization technique. *Appl. Sci.* **13**, 9667. <https://doi.org/10.3390/app13179667> (2023).
- Kashyap, A. K. & Parhi, D. R. Dynamic posture stabilization of humanoid robot NAO using 3D-multilinked dual spring-loaded inverted pendulum model for uneven and inclined floor. *Int. J. Humanoid Robot.* <https://doi.org/10.1142/S021984362350007X> (2023).
- Kashyap, A. K. & Parhi, D. R. Stable locomotion of humanoid robots on uneven terrain employing enhanced DAYANI arc contour intelligent algorithm. *J. Autonomous Vehicles Syst.* **2**, 4, 041002 (2022). <https://doi.org/10.1115/1.4063055>
- Kashyap, A. K. & Parhi, D. R. Dynamic walking of multi-humanoid robots using BFGS Quasi-newton method aided artificial potential field approach for uneven terrain. *Soft Comput.* **27**, 5893–5910. <https://doi.org/10.1007/s00500-022-07606-7> (2023).
- Kashyap, A. K. & Parhi, D. R. Implementation of intelligent navigational techniques for inter-collision avoidance of multiple humanoid robots in complex environment. *Appl. Soft Comput.* **124**, 109001. <https://doi.org/10.1016/j.asoc.2022.109001> (2022).
- M, M. J., Mathew, R. & Hiremath, S. S. Reinforcement learning based approach for mobile robot navigation. in *International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, 523–526 (2019). (2019). <https://doi.org/10.1109/ICCIKE47802.2019.9004256>
- Kashyap, A. K., Parhi, D. R. & Kumar, V. Navigation for multi-humanoid using MFO-aided reinforcement learning approach. *Robotica*. **41**, 1, 346–369. <https://doi.org/10.1017/S0263574722001357> (2023).
- Raj, R. & Kos, A. Dynamic obstacle avoidance technique for mobile robot navigation using deep reinforcement learning. *Int. J. Emerg. Trends Eng. Res.* **11**, 307–314. <https://doi.org/10.30534/ijeter/2023/031192023> (2023).
- Watkins, C. J. C. H. & Dayan, P. Q-Learning. *Mach. Learn.* **8**, 279–292 <https://doi.org/10.1007/bf00992698> (1992).
- Liu, L., Dugas, D., Cesari, G., Siegwart, R. & Dubé, R. Robot navigation in crowded environments using deep reinforcement learning. in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, USA, 5671–5677 (2020). <https://doi.org/10.1109/IROS45743.2020.9341540>
- Fan, T., Long, P., Liu, W. & Pan, J. Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios. *Int. J. Robot Res.* **39**, 7, 856–892. <https://doi.org/10.1177/0278364920916531> (2020).
- Zhu, K. & Zhang, T. Deep reinforcement learning based mobile robot navigation: a review. *Tsinghua Sci. Technol.* **26**, 674–691. <https://doi.org/10.26599/TST.2021.9010012> (2021).
- Mnih, V. et al. Human-level control through deep reinforcement learning. *Nature*. **518**, 529–533. <https://doi.org/10.1038/nature14236> (2015).
- Kulhánek, J., Derner, E., de Bruin, T. & Babuška, R. Vision-based navigation using deep reinforcement learning. in *European Conference on Mobile Robots (ECMR)*, Prague, Czech Republic, 1–8 (2019). (2019). <https://doi.org/10.1109/ECMR.2019.8870964>
- Hsu, S. H., Chan, S. H., Wu, P. T., Xiao, K. & Fu, L. C. Distributed deep reinforcement learning based indoor visual navigation. in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain, 2532–2537 (2018). (2018). <https://doi.org/10.1109/IROS.2018.8594352>
- Singla, A., Padakandla, S. & Bhatnagar, S. Memory-based deep reinforcement learning for obstacle avoidance in UAV with limited environment knowledge. *IEEE Trans. Intell. Transp. Syst.* **22**, 107–118. <https://doi.org/10.1109/TITS.2019.2954952> (2021).
- Li, H., Zhang, Q. & Zhao, D. Deep reinforcement learning-based automatic exploration for navigation in unknown environment. *IEEE Trans. Neural Netw. Learn. Syst.* **31**, 2064–2076. <https://doi.org/10.1109/TNNLS.2019.2927869> (2020).
- Cai, M., Aasi, E., Belta, C. & Vasile, C. I. Overcoming exploration: deep reinforcement learning for continuous control in cluttered environments from temporal logic specifications. *IEEE Robot Automat Lett.* **8**, 2158–2165. <https://doi.org/10.1109/LRA.2023.3246844> (2023).
- Yan, W., Sun, J., Li, Z. & Wang, G. Decentralized multi-robot navigation in unknown environments via hierarchical deep reinforcement learning. in *42nd Chinese Control Conference (CCC)*, Tianjin, China, 4292–4297 (2023). (2023). <https://doi.org/10.23919/CCC58697.2023.10240139>
- Ali, A., Gul, S., Mahmood, T. & Ullah, A. Exploration of unknown environment using deep reinforcement learning. in *International Conference on Robotics and Automation in Industry (ICRAI)*, Peshawar, Pakistan, 1–6 (2023). (2023). <https://doi.org/10.1109/ICRAI57502.2023.10089589>
- Xue, Y. & Chen, W. A UAV navigation approach based on deep reinforcement learning in large cluttered 3D environments. *IEEE Trans. Vehic Technol.* **72**, 3001–3014. <https://doi.org/10.1109/TVT.2022.3218855> (2023).
- Wang, J. & Huang, R. A Mapless navigation method based on deep reinforcement learning and path planning. *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Jinghong, China, 1781–1786 (2022). (2022). <https://doi.org/10.1109/ROBIO55434.2022.10011923>
- Zeng, J. et al. Navigation in unknown dynamic environments based on deep reinforcement learning. *Sensors*. **19**, 3837. <https://doi.org/10.3390/s19183837> (2019).
- Zhu, W. & Hayashibe, M. A hierarchical deep reinforcement learning framework with high efficiency and generalization for fast and safe navigation. *IEEE Trans. Ind. Electron.* **70**, 5, 4962–4971. <https://doi.org/10.1109/TIE.2022.3190850> (2023).

28. Xue, Y. & Chen, W. Multi-agent deep reinforcement learning for UAVs navigation in unknown complex environment. *IEEE Trans. Intell. Veh.* **9**, 1, 2290–2303. <https://doi.org/10.1109/TIV.2023.3298292> (2024).
29. Xiong, H., Ma, T., Zhang, L. & Diao, X. Comparison of end-to-end and hybrid deep reinforcement learning strategies for controlling cable-driven parallel robots. *Neurocomputing*. **377**, 73–84. <https://doi.org/10.1016/j.neucom.2019.10.020> (2020).
30. Sutton, R. S. & Barto, A. G. *Reinforcement Learning: An Introduction*. (MIT Press, 1998). <https://mitpress.mit.edu/9780262039246/reinforcement-learning/> (accessed on June 2, 2024).
31. Puterman, M. L. Chapter 8 Markov decision processes. in *Handbooks in Operations Research and Management Science*, Elsevier **2**, 331–434 ISSN 0927-0507, ISBN 9780444874733. (1990). [https://doi.org/10.1016/S0927-0507\(05\)80172-0](https://doi.org/10.1016/S0927-0507(05)80172-0)
32. Grisetti, G., KümmeDRLe, R., Stachniss, C. & Burgard, W. A tutorial on graph-based SLAM. *IEEE Intell. Transp. Syst. Mag.* **2**, 31–43. <https://doi.org/10.1109/MITS.2010.939925> (2010).
33. Bourgault, F., Makarenko, A. A., Williams, S. B., Grocholsky, B. & Durrant-Whyte, H. F. Information based adaptive robotic exploration. in *IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Switzerland*, 540–545 (2002). <https://doi.org/10.1109/IRDS.2002.1041446>
34. Raj, R. & Kos, A. Different techniques for human activity recognition. *2022 29th Int. Conf. Mixed Des. Integr. Circuits Syst. (MIXDES)*. **171–176** <https://doi.org/10.23919/MIXDES55591.2022.9838050> (2022).
35. Raj, R. & Kos, A. An improved human activity recognition technique based on convolutional neural network. *Sci. Rep.* **13**, 22581. <https://doi.org/10.1038/s41598-023-49739-1> (2023).
36. Zhang, J., Springenberg, J. T., Boedecker, J. & Burgard, W. Deep reinforcement learning with successor features for navigation across similar environments. in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada*, 2371–2378 (2017). <https://doi.org/10.1109/IROS.2017.8206049>
37. Kahn, G., Villafior, A., Ding, B., Abbeel, P. & Levine, S. Self-supervised deep reinforcement learning with generalized computation graphs for robot navigation. in *IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia*, 5129–5136 (2018). <https://doi.org/10.1109/ICRA.2018.8460655>
38. Raj, R. & Kos, A. Study and analysis of discrete event-driven autonomous system with a case study for a robotics task. *Przegląd Elektrotechniczny*. **9**, 50–56. <https://doi.org/10.15199/48.2023.09.09> (2023).
39. Tai, L., Paolo, G. & Liu, M. Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada*, 31–36 (2017). <https://doi.org/10.1109/IROS.2017.8202134>
40. Van Hasselt, H., Guez, A. & Silver, D. Deep reinforcement learning with double Q-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence* **30**, (2016). <https://doi.org/10.1609/aaai.v30i1.10295>
41. Shi, Q., Zhao, S., Cui, X., Lu, M. & Jia, M. Anchor self-localization algorithm based on UWB ranging and inertial measurements. *Tsinghua Sci. Technol.* **24**, 728–737. <https://doi.org/10.26599/TST.2018.9010102> (2019).
42. Raj, R. & Kos, A. Discussion on different controllers used for the navigation of mobile robot. *Int. J. Electron. Telecommun.* **70**, 1, 229–239. <https://doi.org/10.24425/ijet.2024.149535> (2024).
43. Faust, A. et al. PRM-RL: Long-range robotic navigation tasks by combining reinforcement learning and sampling-based planning. in *Proc. 2018 IEEE Int. Conf. Robotics and Automation, Brisbane, Australia*, 5113–5120 (2018). <https://doi.org/10.1109/ICRA.2018.8461096>
44. Jaradat, M. A. K., Al-Rousan, M. & Qadan, L. Reinforcement based mobile robot navigation in dynamic environment. *Robot Computer-integrated Manufact.* **27**, 135–149. <https://doi.org/10.1016/j.rcim.2010.06.019> (2011).
45. Marchesini, E. & Farinelli, A. Discrete deep reinforcement learning for Mapless navigation. in *IEEE International Conference on Robotics and Automation (ICRA), Paris, France*, 10688–10694 (2020). <https://doi.org/10.1109/ICRA40945.2020.9196739>

Acknowledgements

The authors wish to express their thanks for financial support from subvention No. 16.16.230.434 paid for by the AGH University of Krakow, Poland.

Author contributions

Conceptualization, R.R. and A.K.; methodology, R.R.; software, R.R.; validation, R.R.; formal analysis, R.R.; investigation, R.R.; resources, R.R. and A.K.; data curation, R.R.; writing—original draft preparation, R.R.; writing—review and editing, R.R. and A.K.; visualization, R.R. and A.K.; supervision, A.K.; all authors have read and equally contributed to the manuscript.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to R.R.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2024