Study of LiDAR specifications in automated guided vehicle path planning by deep reinforcement learning

Reza Sheikhbahaei, Mohammad Hossein Fallahi, Seyed Abdolmajid Yousefsani[*]
Department of Mechanical Engineering, Faculty of Engineering, Ferdowsi University of Mashhad, Mashhad, Iran

Abstract

The automated guided vehicle path planning performance is highly dependent on the collected LiDAR data. This research centered on studying the LiDAR specifications for automated guided vehicle path planning problem using deep reinforcement learning. Three data collection approaches with differences in the data sample size and scanning range were considered to obtain 15 LiDAR specifications, and the training quality and the path planning performance were evaluated in order to find the optimal LiDAR specification. The most effective results were achieved using the approach of collecting the "distance and angle" pairs of the nearest obstacles. This method consistently reached all the defined targets in both seen and unseen static environments, with no collisions. Additionally, only limited collisions happened in an unseen dynamic environment containing moving obstacles. Notably, increasing the number of data samples can extend the training time, and the excessive data may also complicate decision-making with no significant benefits in terms of the path planning performance.

Keywords: Path planning, LiDAR specifications, Deep reinforcement learning, Collision avoidance, static and dynamic environment

1. Introduction

Using the smart mobile robots in industrial production has made the manufacturing processes more agile, reduced the manufacturing costs, and effectively improved the production speed, endurance, and precision. Among various industrial robotic facilities, the automated guided vehicles (AGVs) are mainly used to work in dangerous

[*] Corresponding author (S.A. Yousefsani): yousefsani@um.ac.ir
Department of Mechanical Engineering, Faculty of Engineering, Ferdowsi University of Mashhad, P.O. Box: 9177948974, Mashhad, Iran

environments, explore unreachable areas, and carry sensitive objects and materials [1]. Reliable performance of such mobile robots substantially depends on the proper specification of the path planning strategies defining the best routes from the origin towards different destinations without encountering the obstacles [2].

Path planning strategies are classified in the following categories: graph-based search, heuristic intelligence, local obstacle avoidance, artificial intelligence, sampling-based, planner-based, constraint problem satisfaction-based, and other algorithms [3]. Considering the complication of path planning in unknown complex environments with a dense distribution of static and dynamic obstacles, implementation of the artificial intelligence (AI) assisted strategies has received special attention in recent years (for further information see Ref. [4]). Deep Learning (DL) and Reinforcement Learning (RL) are among the widely used powerful AI methods to solve the path planning problems (see for example [5]).

More specifically, the DL is a machine learning method that generates a multilayered (deep) neural network to make an appropriate connection between the input and output vectors [6]. The RL operates based on the Markov Decision Process (MDP) [7], where the interaction between the agent and environment generates a reward value at each state. This is used as a feedback signal to train the agent. At every step of this process, the environment is informed by the action of the agent at the current state and generates a new state. The reward function evaluates this action considering the current and the new states. At the training stage, a so-called Q-Table is generated to predict the reward summation (or the Q-Value) of the agent for every state-action pair, and hence, the agent is able to decide to take the appropriate action for each state to attain the higher accumulated rewards.

Fusion of the DL and RL strategies has brought about the deep reinforcement learning (DRL) method [8]. DRL finds a deep neural network (or the Q-Network) which receives the state as an input vector and calculates the Q-Values for all possible actions. This approach, which is also referred to as the deep Q-Network (DQN), estimates the state-action Q-Value by employing the DL. As an example, Guo et al. [9] have introduced a pioneering decentralized path planning model built upon the foundations of DRL.

Application of the DRL in path planning problems has been reported in several studies, as summarized in Table 1. In most of them, the specified problem was to find a path to lead the AGV from its current position to a predefined destination avoiding the obstacle collision. In most cases, the environmental awareness of the AGV was achieved using the light detection and ranging sensor (the LiDAR); however, the selection of the sensor data acquisition approach, the sensor scanning range, and the number of the data samples was quite different. Another approach is indirect use of the LiDAR data for path planning. For instance, as reported in Ref. [10], the LiDAR data is used to organize a binary occupancy grid needed by the proposed Reinforcement Learning Particle Swarm Optimization (RLPSO) algorithm. Table 1 compares the details of the LiDAR data acquisition proposed in previous studies. In some of these studies [11, 27] the sonar sensors were alternatively used, which are operationally similar to the LiDAR sensors.

An open question in the LiDAR-assisted AGV path planning is how to adjust the LiDAR specifications to achieve higher performance and lower cost. Olayemi et al. [30] has addressed this issue only by three scenarios with one sampling approach in static environments. To find a more comprehensive answer to this query, we developed a detailed study to identify the optimum operational specifications of the LiDAR in a DQN-assisted AGV path planning problem. More specifically, we proposed a potential-field-based reward function able to further penalize the AGV when it gets closer to multiple obstacles, rather than obstacle collision. Moreover, we implemented three sampling approaches with different LiDAR scanning ranges and numbers of LiDAR data samples to find the optimum sensor specification among 15 scenarios, further examined in both static and dynamic environments. Several metrics including the training quality and the path planning performance were also evaluated.

The rest of the paper is organized as follows. In Section 2 the problem is defined and the RL assumptions (i.e., the environment, the agent, the states, the action space, and the reward function) as well as the LiDAR specifications are introduced. In the next section, the simulation results including the effects of different LiDAR specifications on the training quality and the path planning performance are presented and a comprehensive discussion is provided. Finally, the last section highlights the concluding remarks.

2. Problem definition and method

As stated in the previous section, this paper studies the effects of different LiDAR specifications on the path planning of a LiDAR-assisted AGV using the reinforcement learning strategy. For this, a DQN training framework was developed to train an agent trying to discover its way towards a target within a predefined environment in presence of a number of fixed and moving obstacles. In brief, at each state, the agent, which is a LiDAR-assisted AGV, takes an action from the action space, and receives a reward value according to the specified reward function. Given that the LiDAR specifications can affect the training quality and the path planning performance, it is essential to find the best LiDAR specifications from different points of view. In what follows, the methodology including the problem assumptions and formulations as well as the implementation platform are defined in detail. To simulate the LiDAR-assisted path planning problem and to implement the proposed DQN training framework different Python libraries were utilized.

2.1 Agent and environment

The simulation environment considered in this study is a two-dimensional 7×7 m^2 square room with twelve fixed obstacles, as shown in Fig. 1. Seven obstacles are circular with a diameter of 30 cm, and the remainder are square shaped with an edge length of 30 cm. The working environment was visualized using the python Matplotlib library.

The agent is an AGV equipped with a 360-degree LiDAR sensor, which is considered to move with a constant linear velocity v and a variable rotational velocity ω. Therefore, the instantaneous position (x_t, y_t) and orientation θ_t of the agent can be calculated as follows:

$$x_t = x_{t-1} + v\delta t \cos\theta_t \tag{1}$$

$$y_t = y_{t-1} + v\delta t \sin\theta_t \tag{2}$$

$$\theta_t = \theta_{t-1} + \omega\delta t \tag{3}$$

where the simulation time step is $\delta t = 1\,\mathrm{ms}$.

2.2 State and Action Space

In this study, the state vector is a $(m+2)\times 1$ vector consisting of m LiDAR data in addition to the relative distance and orientation of the agent with respect to the target point. Moreover, five discrete angular velocities, as listed in Table 2, were assumed as the possible actions to be selected by the agent at every step; while it constantly moves with a linear velocity of 0.5 m/s. Each step, during which the agent action is constant, was set at 50 ms.

2.3 Reward Function

The reward function *R* should be designed to encourage the agent to move toward the goal as fast as possible while avoiding the obstacles. Furthermore, to investigate the effects of various LiDAR specifications, the reward function must be able to take several LiDAR data into account. Therefore, the reward value at the time *t* is calculated as follows:

$$R = R_d + R_g - \sum_{i=1}^{m} R_{o,i} \tag{4}$$

In this equation, the first term on the right-hand side (R_d) stimulates the agent to continuously move toward the goal. That is,

$$R_d = 20\left(d_{t-1}^g - d_t^g\right) \tag{5}$$

where d_{t-1}^g and d_t^g are the distance of the agent to the target at the previous and current steps, respectively. Indeed, R_d prepares instantaneous feedback for the agent action in order to decrease the convergence time of the algorithm. The term R_g in Eq. (4) rewards the agent intensively when it reaches the target point. That is,

$$R_g = 200, \text{if } d_t^g \le d_{reach}^g \tag{6}$$

in which $d_{reach}^g = 0.2^m$ is the reaching target distance. Finally, the terms $R_{o,i}$ in the right-hand side of Eq. (4) penalize the agent, which is moving toward the *i*-th obstacle, if its relative distance is less than $d_{col}^o = 2^m$. That is,

$$R_{o,i}=\begin{cases}\frac{1}{2}\left[\left(\frac{1}{d_t^{o,i}}-\frac{1}{d_{col}^{o}}\right)^2-\left(\frac{1}{d_{t-1}^{o,i}}-\frac{1}{d_{col}^{o}}\right)^2\right], \text{if } d_t^{o,i}\le d_{col}^{o} \text{ and } d_{t-1}^{o,i}\le d_{col}^{o}\\ \frac{1}{2}\left(\frac{1}{d_t^{o,i}}-\frac{1}{d_{col}^{o}}\right)^2, \text{if } d_t^{o,i}\le d_{col}^{o} \text{ and } d_{t-1}^{o,i}> d_{col}^{o}\end{cases} \quad (i=1,...,m) \tag{7}$$

where $d_{t-1}^{o,i}$ and $d_t^{o,i}$ represent the agent distances to the *i*-th obstacle at the previous and current steps, respectively.

2.4 Training Assumptions

The training process was performed by employing the DQN algorithm prepared in the Stable-Baselines3 [31]. In this algorithm a neural network with three hidden layers of 256 nodes was developed. Each training episode was considered as completed when either the agent hit an obstacle or the trial timed out after 500 steps. Other training parameters were set as presented in Table 3.

2.5 LiDAR Specifications

In order to find the best LiDAR specifications, the following data collection approaches were analyzed during the optimization process. It is worth to note that since the obstacles behind the agent are not used for path planning, the corresponding LiDAR data samples were excluded.

2.5.1 Approach A

In this data collection approach, the “distance and angle” pairs of the nearest obstacles (i.e., those with less distances to the agent) were collected. For this, 23 LiDAR data (i.e., distances from the obstacles) at identical angular sectors in the front field of view of the agent, say in range of -90° to 90°, were collected. Afterwards, *n* minimum distances as well as their corresponding angles were considered as the *m* LiDAR data delivered to the neural network. Five cases were assumed with $n=1,2,3,4,$ and 5, which corresponds to $m=2,4,6,8,$ and 10.

In this approach, the LiDAR data were collected at fixed angular distances of $\varphi=180/22=8.\overline{18}°$. Since the minimum visible dimension of the obstacles, say the diameter of the circular obstacles $\left(0.3^m\right)$ and the edge of the square obstacles $\left(0.3^m\right)$, is bigger than parameter $b=2\left(d_{col}^{o}\sin\left(\varphi/2\right)\right)=0.289^m$ as demonstrated in Fig. 2, this LiDAR data distribution guarantees that all obstacles closer than d_{col}^{o} can be detected. Therefore, the reward value is certainly affected by all the surrounding obstacles.

2.5.2 Approach B

In the second approach, the LiDAR data acquisition was conducted in angular range -90° to 90°, while *m* LiDAR data were collected at the following predefined angles:

$$m = 1: \varphi = 0$$
$$m = 3: \varphi = \{-60°, 0, +60°\}$$
$$m = 5: \varphi = \{-72°, -36°, 0, +36°, +72°\}$$
$$m = 7: \varphi = \{-77.14°, -51.43°, -25.71°, 0, +25.71°, +51.43°, +77.14°\}$$
$$m = 9: \varphi = \{-80°, -60°, -40°, -20°, 0, +20°, +40°, +60°, +80°\}$$

Accordingly, this approach picks only the distances of *m* obstacles lying in the middle angles of equal intervals in the fixed range of -90° to 90°.

2.5.3 Approach C

In the third approach, the LiDAR data were collected at equal intervals of 20° and different acquisition ranges as follows:

$$m = 1: \varphi = 0$$
$$m = 3: \varphi = \{-20°, 0, +20°\}$$
$$m = 5: \varphi = \{-40°, -20°, 0, +20°, +40°\}$$
$$m = 7: \varphi = \{-60°, -40°, -20°, 0, +20°, +40°, +60°\}$$
$$m = 9: \varphi = \{-80°, -60°, -40°, -20°, 0, +20°, +40°, +60°, +80°\}$$

This approach is similar to the second one, except that the data collection range changes with *m*.

Considering these assumptions, 15 LiDAR specifications are possible, among which two specifications are repeated in the approaches B and C. Consequently, 13 distinct specifications were considered, and the RL training process was executed for each specification for 180,000 steps.

3. Results and Discussion

The proposed approaches were evaluated and compared from two aspects: the RL training quality and the path planning performance. The RL training quality was assessed by studying the training time required for saturation of the mean episode length or the mean episode reward. In order to evaluate the path planning performance, the behavior of the agent was evaluated in response to a specific number of randomly predefined targets. For this, three different metrics including the success rate, the total score, and the agent's trajectory were tested.

As stated in the previous section, 15 LiDAR specifications were considered to be studied. The corresponding simulation results are labeled as shown in Table 4.

3.1 Training Quality

As mentioned in the previous section, each training episode is considered completed when either an obstacle collision has occurred or the trial has timed out after 500 steps. At the beginning of the RL training process the obstacle collision is unavoidable due to random or untrained decision making. Gradually, as the training evolves, fewer number of obstacle collisions is expected, and consequently, the time-out condition would complete the episodes. It is obvious that a faster decrease in the number of collisions shows a better training performance.

As depicted in Fig. 3, the training performance of the first approach (i.e., A1 to A5) is significantly higher than the other two approaches. Among the studies listed in Table 1, this approach was only employed in Refs. [19, 23, 24]. The saturation is realized if almost all of the episodes end after 500 steps, where the agent touches the goals with nearly no collisions. With this in mind, the mean episode length in each one of these five cases has been saturated to 500 steps after about 60,000 steps, whereas it has not been saturated in the other approaches, even after 270,000 steps.

Figure 3 shows more interesting results. The worst case is B1/C1, where only one LiDAR data at $\varphi = 0$ was collected. This case has not been reported in previous studies listed in Table1. The next is C2, for which three LiDAR data were collected from the angular range of [-20° 20°]. This indicates that a narrow scanning range does not give the AGV a good view to avoid collision. B2 with three LiDAR data collected from the angular ranges of [-60° 60°] is the next in line, for which the inappropriate training performance highlights deficiency of three LiDAR data for covering the AGV's 120° field of view.

B3 and C3 with five LiDAR data are in the next places. More specifically, B3 with five LiDAR data in the range of [-72° 72°] shows a better training performance in comparison with C3 with the same amount of data in the range of [-40° 40°]. Moreover, reviewing the results of B4 and C4 with the same number of LiDAR data indicates that the scanning range of [-77.1° 77.1°] as defined in B4 is slightly more effective compared to [-60° 60°] as defined in C4.

Finally, the best results among the cases of approaches 2 and 3 were obtained for B4, C4, and B5/C5. As also confirmed in Ref. [30], widening the scanning range and increasing the number of data samples can improve the training quality. Remarkably, the mean episode length for C4 and B5/C5 increases with quite similar trends for about three quarters of the steps. However, in the remaining steps it further grows for B5/C5. This can be justified considering that two more LiDAR data have been collected at $\varphi = \{-80°, 80°\}$ in B5/C5.

Figure 4 illustrates that the greatest mean episode rewards are also achieved in cases with the first approach. It shows that, besides the obstacle avoidance, the agent in these cases moves appropriately toward the goals. Moreover, there is no significant difference between the overall trends of the mean reward in the approaches B and C.

3.2 Path Planning Performance

The path planning performance of different LiDAR specifications was also studied. For this, a large number of random targets were first generated, and then, the agent's path planning with different LiDAR specifications was simulated. The ultimate goal for success was set to touch 100 out of the predefined targets. It is noticeable that as soon as the agent either collides with an obstacle or does not reach the target after 600 steps, it restarts from the origin, and a new target is relocated.

As depicted in Fig. 5, in all cases of the approach A the agent was successful to touch the first 100 predefined targets, while it requires much more attempts in the other two approaches. Similar to what was observed in the subsection 3.1, B4 and B5/C5, which also presented a good training quality as shown in Fig. 3, here are in the next places of the success rate. Subsequently, as shown in Fig. 6, the cases of the approach A experienced less collisions and attained higher scores.

The paths traveled by the trained agent with different LiDAR specifications are depicted in Figures 6 to 8. In these figures, the agent's path to reach two, 10, and 100 successful targets are shown, respectively. Moreover, red crosses show the predefined targets.

It can be observed that the path planning performance is higher for all cases in the approach A that experienced no collisions and planned suitable paths to touch all targets, one after another. This approach was only employed in Refs. [19, 23, 24]. For a better comparison of these cases, the associated paths generated to reach 10 successful targets are shown in Fig. 9.

3.3 Testing in unseen environments

As indicated in the previous subsections, the trained agents using the first approach LiDAR specifications showed the best path planning performance. However, the next question is how they perform in some unknown environments. To answer this query, two different unseen environments with circular boundaries, including a static one with 12 fixed obstacles and a more complex dynamic one with four fixed and four moving obstacles were designed for the testing phase, as shown in Fig. 10. Notably, the trained agents are completely unfamiliar with both of these new environments. Despite similar testing in static environments, no evaluation has been performed in a dynamic environment in Ref. [30].

Similar to the subsection 3.2, the path planning performance of the trained agents was evaluated here by successfully touching 100 out of the predefined random targets. The simulation results, including the success rate and the total score, of the cases A1 to A5 are compared for both environments, as shown in Table 5.

Results of testing in the new static environment indicate an excellent path planning performance with no collisions for all cases. Unlike the previous tests, a few obstacle collisions occurred in the new dynamic environment. The high success rates achieved in this test also reveals a remarkable path planning performance. However, one can conclude that whether the proposed training process or the underlying assumptions, including the DQN learning method, the static training environment, the proposed reward function, or the considered LiDAR specification, may be improved for a better path planning in an unseen dynamic environment.

As an interesting result, the simulations in the three proposed environments indicate that the case A1 always achieved the highest total score with no obstacle collisions. This can be justified considering that the decision making would be much simpler when using only one incoming LiDAR data. In other words, a better connection is achieved between the LiDAR data and the action space in the trained DQN of case A1.

4. Conclusion

This paper was focused on the adjustment of the LiDAR specifications in an AGV path planning problem using deep reinforcement learning. The ultimate goal was to identify a set of specifications required to achieve higher performances at lower costs. Using three data collection approaches, 15 LiDAR specifications were defined with different numbers of data samples and scanning ranges, and the training quality and path planning performance were assessed. The best path planning performance and training quality were achieved in one of these approaches, where a number of "distance and angle" pairs of the nearest obstacles were collected. All cases of this approach were completely successful to achieve the defined ultimate goal in both previously seen and unseen static environments with no collisions. Furthermore, no or only a few collisions occurred when tested in an unseen dynamic environment with moving obstacles. It is of particular interest that increasing the number of data samples might require more training time, while adding too many data samples only complicates the decision-making process with no additional advantages.

Funding Statement

This research was funded by Ferdowsi University of Mashhad [grant number 7863].

Conflict of Interest

The authors declare that they have no conflict of interest concerning the contents of this article.

References

[1] Phuyal, S., Bista, D., and Bista, R., "Challenges, opportunities and future directions of smart manufacturing: a state of art review." *Sustainable Futures* 2 (2020): 100023. https://doi.org/10.1016/j.sftr.2020.100023

[2] Wang, C., and Mao, J., "Summary of agv path planning." In *2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE)*, pp. 332-335. IEEE, 2019. https://doi.org/10.1109/EITCE47263.2019.9094825

[3] Qin, H., Shao, S., Wang, T., et al. "Review of autonomous path planning algorithms for mobile robots." *Drones* 7, no. 3 (2023): 211. https://doi.org/10.3390/drones7030211

[4] Lin, S., Liu, A., Wang, J., et al. "A review of path-planning approaches for multiple mobile robots." *Machines* 10, no. 9 (2022): 773. https://doi.org/10.3390/machines10090773

[5] Zhang, C., Cheng, P., Du, B., et al. "AUV path tracking with real-time obstacle avoidance via reinforcement learning under adaptive constraints." *Ocean Engineering* 256 (2022): 111453. https://doi.org/10.1016/j.oceaneng.2022.111453

[6] Janiesch, C., Zschech, P., and Heinrich, K., "Machine learning and deep learning." *Electronic Markets* 31, no. 3 (2021): 685-695. https://doi.org/10.1007/s12525-021-00475-2

[7] Ernst, D., and Louette., A., "Introduction to reinforcement learning." *Feuerriegel, S., Hartmann, J., Janiesch, C., and Zschech, P* (2024): 111-126. https://damien-ernst.be/wp-content/uploads/2024/03/f90de-introduction_to_reinforcement_learning.pdf

[8] François-Lavet, V., Henderson, P., Islam, R., et al. "An introduction to deep reinforcement learning." *Foundations and Trends® in Machine Learning* 11, no. 3-4 (2018): 219-354. https://doi.org/10.1561/2200000071

[9] Guo, D., Ji, S., Yao, Y., et al. "A decentralized path planning model based on deep reinforcement learning." *Computers and Electrical Engineering* 117 (2024): 109276. https://doi.org/10.1016/j.compeleceng.2024.109276

[10] Nagar, H., Paul, A., Machavaram, R., et al. "Reinforcement learning particle swarm optimization based trajectory planning of autonomous ground vehicle using 2D LiDAR point cloud." *Robotics and Autonomous Systems* 178 (2024): 104723. https://doi.org/10.1016/j.robot.2024.104723

[11] Li, Y., Li, C. and Zhang, Z., "Q-learning based method of adaptive path planning for mobile robot." In *2006 IEEE international conference on information acquisition*, pp. 983-987. IEEE, 2006. https://doi.org/10.1109/ICIA.2006.305871

[12] Choi, J , Lee, G., and Lee, C., "Reinforcement learning-based dynamic obstacle avoidance and integration of path planning." *Intelligent Service Robotics* 14 (2021): 663-677. https://doi.org/10.1007/s11370-021-00387-2

[13] Wang, W., Wu, Z., Luo, H., et al. "Path planning method of mobile robot using improved deep reinforcement learning." *Journal of Electrical and Computer Engineering* 2022, no. 1 (2022): 5433988. https://doi.org/10.1155/2022/5433988

[14] Tai, L., Paolo, G., and Liu, M., "Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation." In 2017 IEEE/*RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 31-36. https://doi.org/10.1109/IROS.2017.8202134

[15] Gao, J., Ye, W., Guo, J., et al. "Deep reinforcement learning for indoor mobile robot path planning." *Sensors* 20, no. 19 (2020): 5493. https://doi.org/10.3390/s20195493

[16] Rummery, G.A., and Niranjan, M., *On-line Q-learning using connectionist systems*. Vol. 37. Cambridge, UK: University of Cambridge, Department of Engineering, 1994.

[17] Xiang, J., Li, Q., Dong, X., et al. "Continuous control with deep reinforcement learning for mobile robot navigation." In *2019 Chinese Automation Congress (CAC)*, pp. 1501-1506. IEEE, 2019. https://doi.org/10.1109/CAC48633.2019.8996652

[18] Botteghi, N., Sirmacek, B., Mustafa, K.A.A., et al. "On reward shaping for mobile robot navigation: A reinforcement learning and SLAM based approach." *arXiv preprint arXiv:2002.04109* (2020). https://doi.org/10.48550/arXiv.2002.04109

[19] Guo, S., Zhang, X. Du, Y., et al. "Path planning of coastal ships based on optimized DQN reward function." *Journal of Marine Science and Engineering* 9, no. 2 (2021): 210. https://doi.org/10.3390/jmse9020210

[20] Wang, Y., Fang, Y., Lou, P., et al. "Deep reinforcement learning based path planning for mobile robot in unknown environment." In *Journal of Physics: Conference Series*, vol. 1576, no. 1, p. 012009. IOP Publishing, 2020. https://doi.org/10.1088/1742-6596/1576/1/012009

[21] Li, P., Ding, X., Sun, H., et al. "Research on dynamic path planning of mobile robot based on improved DDPG algorithm." *Mobile Information Systems* 2021, no. 1 (2021): 5169460. https://doi.org/10.1155/2021/5169460

[22] Qiang, L., Nanxun, D., Huican, L., et al. "A model-free mapless navigation method for mobile robot using reinforcement learning." In *2018 Chinese Control And Decision Conference (CCDC)*, pp 3410-3415 IEEE, 2018. https://doi.org/10.1109/CCDC.2018.8407713

[23] Dayal, A., Cenkeramaddi, L.R., and Jha, A., "Reward criteria impact on the performance of reinforcement learning agent for autonomous navigation." *Applied Soft Computing* 126 (2022): 109241. https://doi.org/10.1016/j.asoc.2022.109241

[24] Sivashangaran, S., and Zheng, M., "Intelligent autonomous navigation of car-like unmanned ground vehicle via deep reinforcement learning." *IFAC-PapersOnLine* 54, no. 20 (2021): 218-225. https://doi.org/10.1016/j.ifacol.2021.11.178

[25] Kim, I., Nengroo, S.H., and Har, D., "Reinforcement learning for navigation of mobile robot with LiDAR." In *2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pp. 148-154. IEEE, 2021. https://doi.org/10.1109/ICECA52323.2021.9675949

[26] Manh, T.N., Manh, C.N., Tien, D.P., et al. "Autonomous navigation for omnidirectional robot based on deep reinforcement learning." *International Journal of Mechanical Engineering and Robotics Research* 9, no. 8 (2020): 1134-1139. https://doi.org/10.18178/ijmerr.9.8.1134-1139

[27] Li, C., Zhang, J., and Li, Y., "Application of artificial neural network based on q-learning for mobile robot path planning." In *2006 IEEE International Conference on Information Acquisition*, pp. 978-982. IEEE, 2006. https://doi.org/10.1109/ICIA.2006.305870

[28] Bai, Y., Ding, X., Hu, D., et al. "Research on dynamic path planning of multi-AGVs based on reinforcement learning." *Applied Sciences* 12, no. 16 (2022): 8166. https://doi.org/10.3390/app12168166

[29] Luong, M., and Pham, C., "Incremental learning for autonomous navigation of mobile robots based on deep reinforcement learning." *Journal of Intelligent & Robotic Systems* 101, no. 1 (2021): 1. https://doi.org/10.1007/s10846-020-01262-5

[30] Olayemi, K.B., Van, M., McLoone, S., et al. "The impact of LiDAR configuration on goal-based navigation within a deep reinforcement learning framework." *Sensors* 23, no. 24 (2023): 9732. https://doi.org/10.3390/s23249732

[31] Raffin, A., Hill, A., Gleave, A., et al. "Stable-Baselines3: Reliable Reinforcement Learning Implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-1364.html

List of captions:

Table 1 Summary of the LiDAR data acquisition approaches used in previous studies

Table 2 The agent's action list

Table 3 Training parameters

Table 4 The labels of 15 LiDAR specifications

Table 5 The success rate and the total score of cases A1 to A5 in the new environments

Fig. 1 The proposed simulation environment

Fig. 2 The minimum detectable obstacle dimension *b* in Approach A

Fig. 3 Evolution of the mean episode length with the training progress for 15 LiDAR specifications

Fig. 4 Evolution of the mean episode reward with the training progress for 15 LiDAR specifications

Fig. 5 Comparison of 15 LiDAR specifications (a) success rate and (b) total score

Fig. 6 The agent's path to reach two successful targets

Fig. 7 The agent's path to reach 10 successful targets

Fig. 8 The agent's path to reach 100 successful targets

Fig. 9 The agent's paths to reach 10 successful targets for the approach A cases

Fig. 10 The unseen circular environments designed for testing, (a) the static environment with fixed obstacles and (b) the dynamic environment with both fixed and moving obstacles

Table 1 Summary of the LiDAR data acquisition approaches used in previous studies

Reference	LiDAR Range	Number of LiDAR Data Samples	LiDAR Data Type	RL Method
[11]	[-90° 90°]	7	Predefined angles*	Q-learning
[12]	[-180° 180°]	512	Predefined angles	SAC
[13]	[-90° 90°]	9	Predefined angles	Improved DQN + ε-greedy search
[14]	[-90° 90°]	10	Predefined angles	Asynchronous DDPG
[15]	[-57.3° 57.3°]	10	Predefined angles	PRM+TD3
[16]	[-30° 30°]	5	Predefined angles	Three Q-learning based algorithms
[17]	[-180° 180°]	10	Predefined angles	SAC
[18]	[-90° 90°]	40	Predefined angles	DDPG
[19]	[-180° 180°]	2	Nearest Obstacle**	Optimized DQN
[20]	[-60° 60°]	3	Predefined angles	DDQN
[21]	[-90° 90°]	20	Predefined angles	Improved DDPG
[22]	[-135° 135°]	10	Predefined angles	Improved Q-learning
[23]	[-180° 180°]	24+2	Predefined angles + Nearest Obstacle	DQN
[24]	[-67.5° 67.5°]	1	Nearest Obstacle	DDPG
[25]	[-180° 180°]	24	Predefined angles	DQN
[26]	[-180° 180°]	26	Predefined angles	DQN
[27]	[-90° 90°]	8	Predefined angles	Q-learning
[28]	[-90° 90°]	5	Predefined angles	Kohonen Q-learning
[29]	[-90° 90°]	10	Predefined angles	An incremental learning method

* The distance data collected at some specific angles
** The angle-distance pair of data collected for the nearest obstacle

Table 2 The agent's action list

Action	Angular velocity (rad/s)
0	-4.0
1	-2.0
2	0
3	2.0
4	4.0

Table 3 Training parameters

Parameter Name	Parameter Value
Action Space Size	5
Learning Rate (α)	0.0003
Discount Factor (γ)	0.995
Activation Function	ReLU
Hidden Layers	3
Number of Neurons	256
Batch Size	256
Target update interval	512 steps

Table 4 The labels of 15 LiDAR specifications

Label	Approach	Data Samples (*m*)	Scan Range (degree)
A1	1	2	[-90° 90°]
A2	1	4	[-90° 90°]
A3	1	6	[-90° 90°]
A4	1	8	[-90° 90°]
A5	1	10	[-90° 90°]
B1/C1	2	1	[0 0]
B2	2	3	[-60° 60°]
B3	2	5	[-72° 72°]
B4	2	7	[-77.1° 77.1°]
B5/C5	2	9	[-80° 80°]
C1/B1	3	1	[0 0]
C2	3	3	[-20° 20°]
C3	3	5	[-40° 40°]
C4	3	7	[-60° 60°]
C5/B5	3	9	[-80° 80°]

Table 5 The success rate and the total score of cases A1 to A5 in the new environments

Environment	Evaluation Parameter	LiDAR Specification				
		A1	A2	A3	A4	A5
Static	Success Rate (%)	100	100	100	100	100
	Total Score	20063	20035	19797	19681	19769
Dynamic	Success Rate (%)	100	97.1	96.2	99.0	94.3
	Total Score	19719	16449	15778	17153	16859

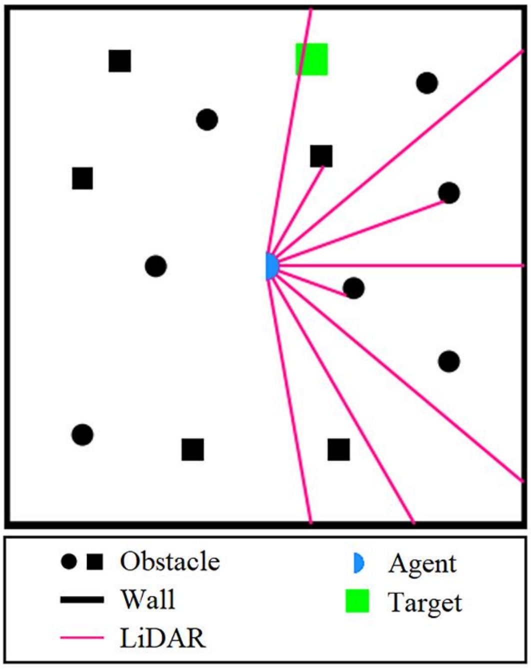


Fig. 1 The proposed simulation environment

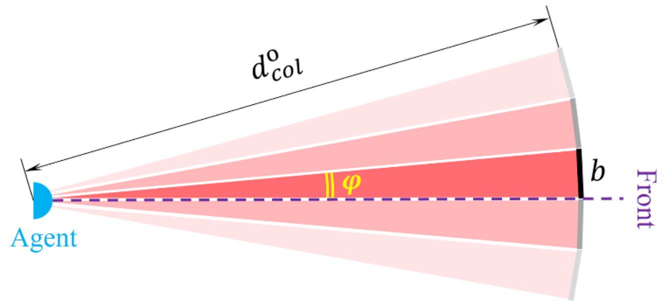


Fig. 2 The minimum detectable obstacle dimension *b* in Approach A

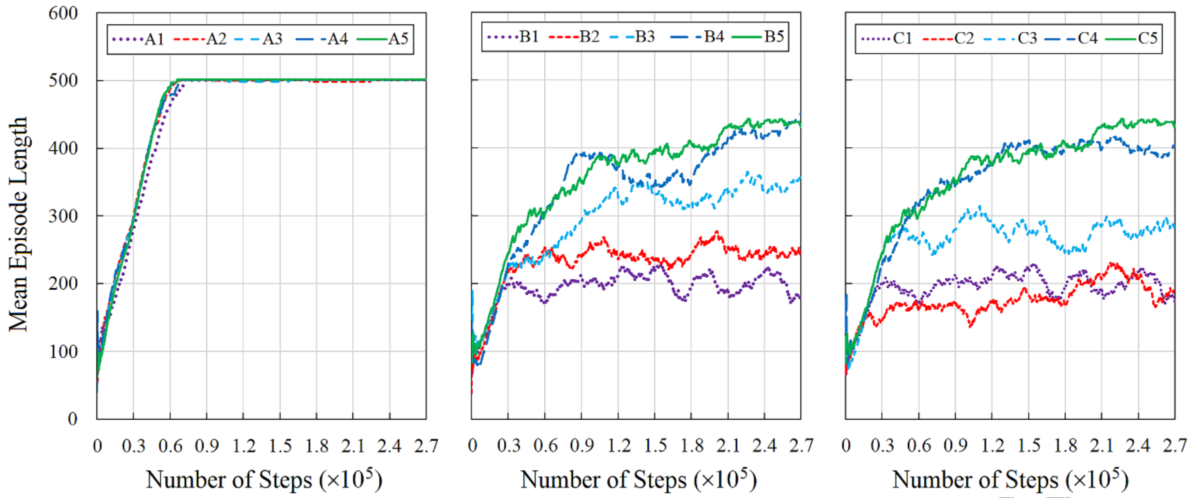


Fig. 3 Evolution of the mean episode length with the training progress for 15 LiDAR specifications

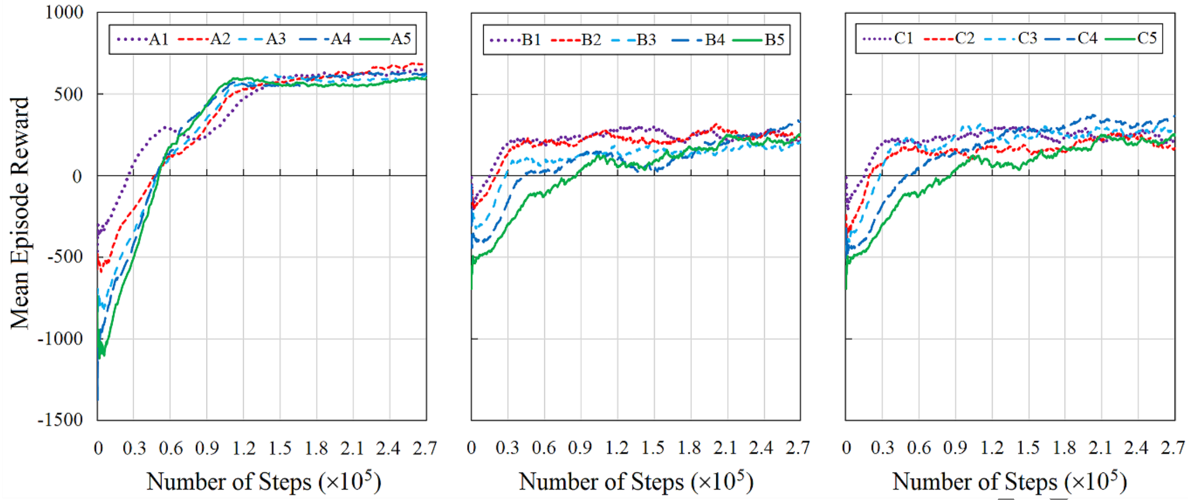


Fig. 4 Evolution of the mean episode reward with the training progress for 15 LiDAR specifications

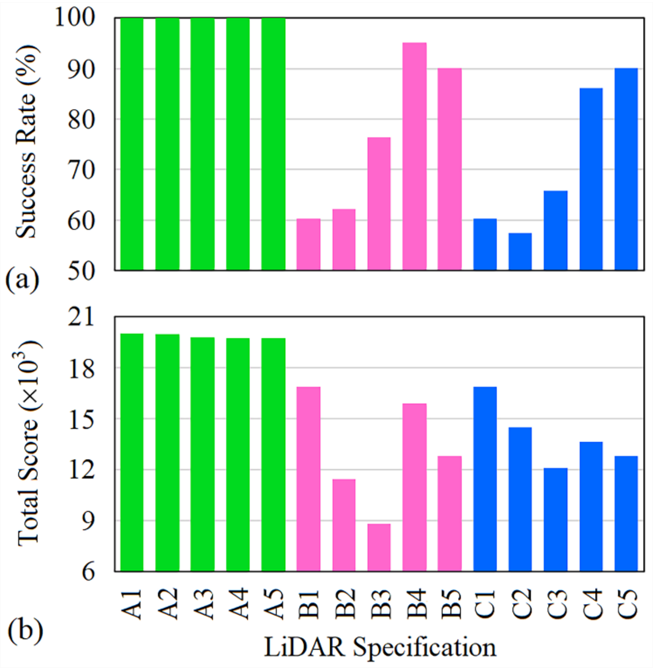


Fig. 5 Comparison of 15 LiDAR specifications (a) success rate and (b) total score

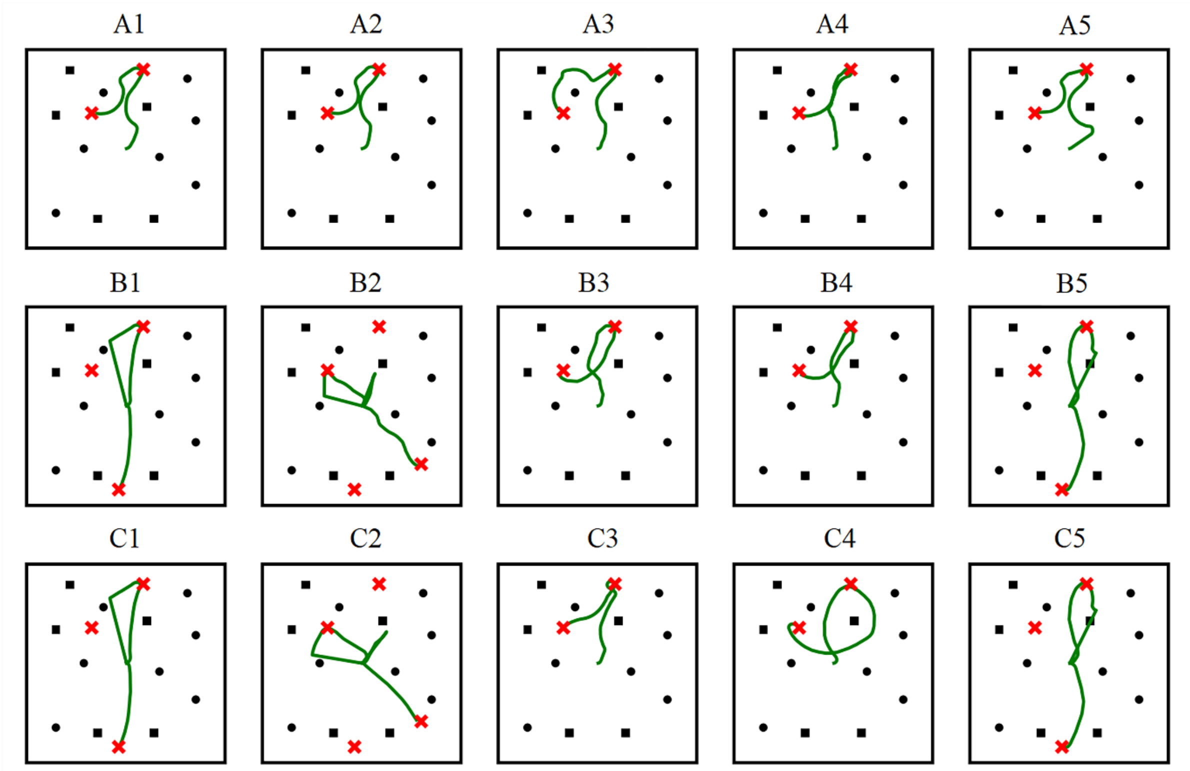


Fig. 6 The agent's path to reach two successful targets

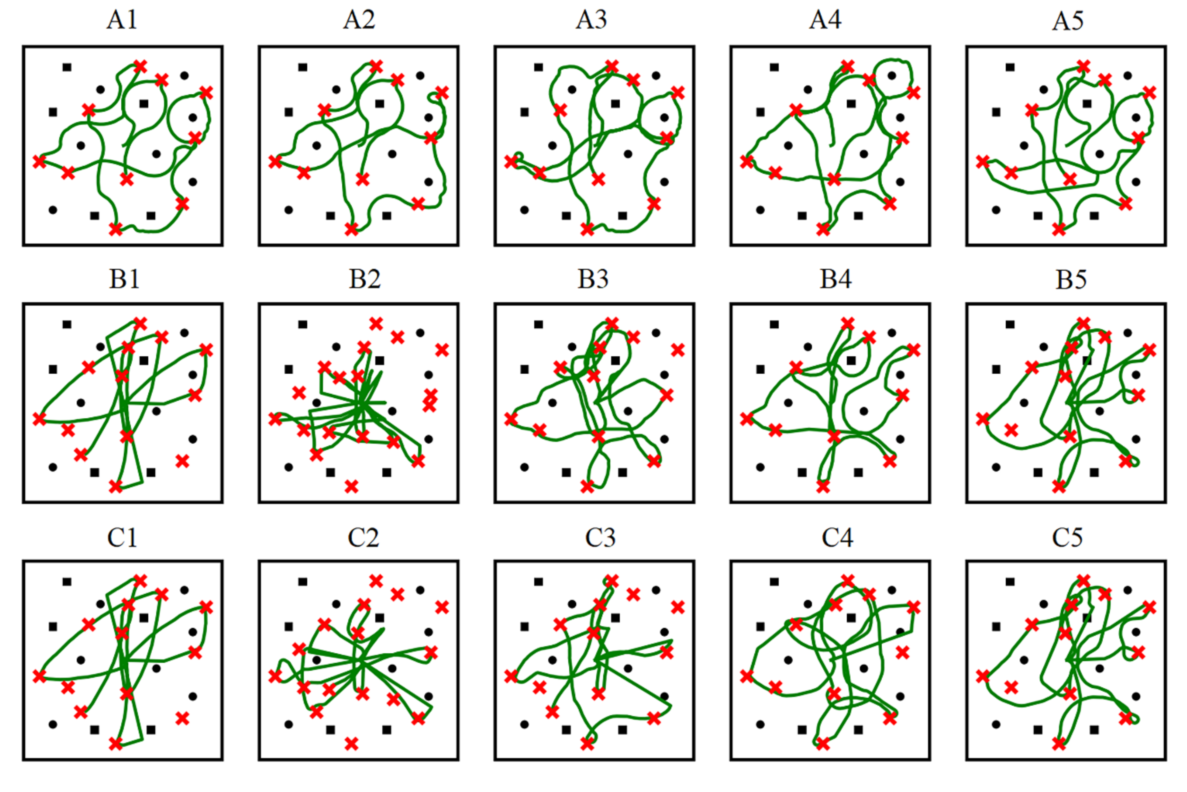


Fig. 7 The agent's path to reach 10 successful targets

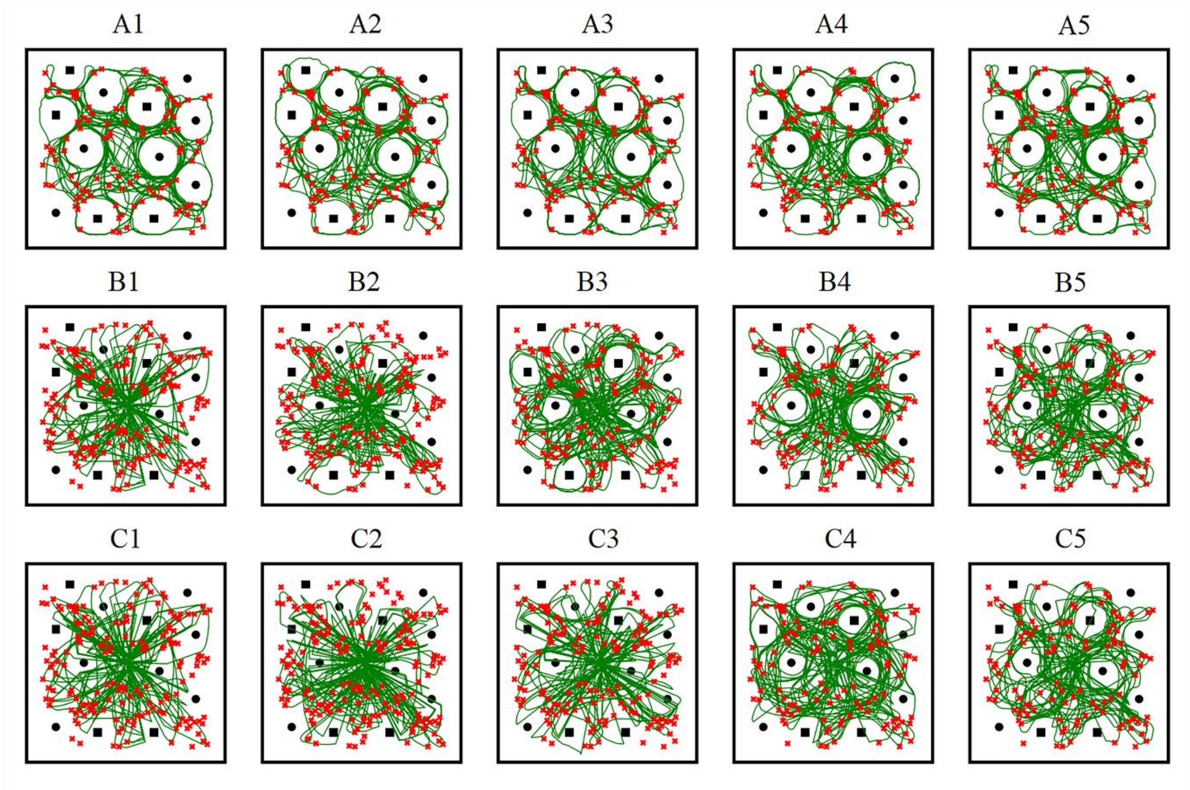
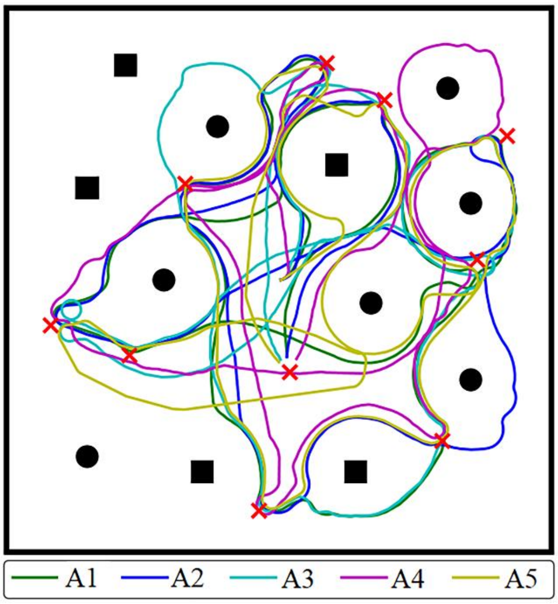


Fig. 8 The agent's path to reach 100 successful targets



٤٦٠
٤٦١ **Fig. 9** The agent's paths to reach 10 successful targets for the approach A cases

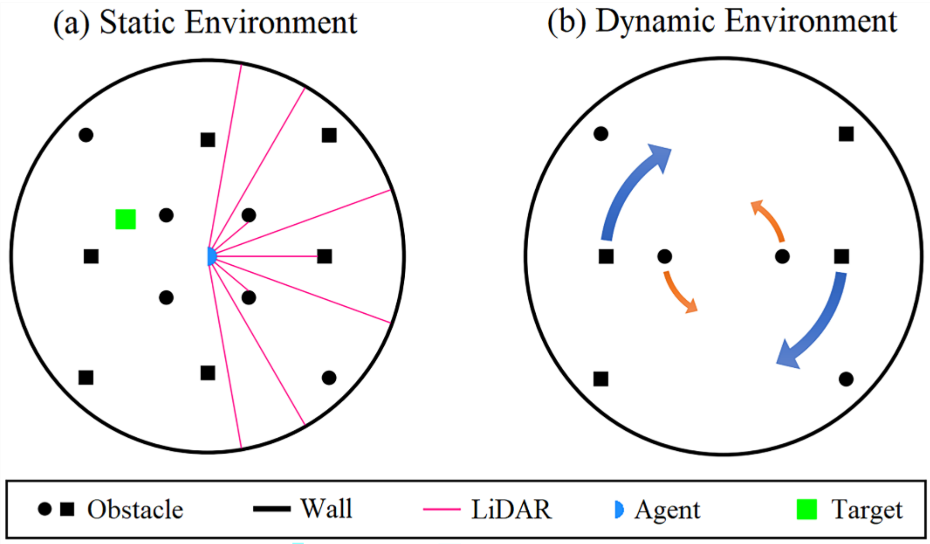
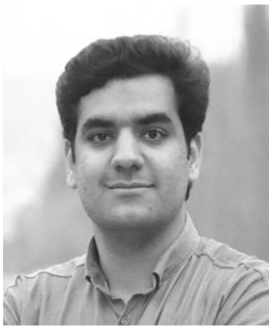


Fig. 10 The unseen circular environments designed for testing, (a) the static environment with fixed obstacles and (b) the dynamic environment with both fixed and moving obstacles

Biographies

	Dr. Reza Sheikhbahaei received his M.Sc. and Ph.D. degrees in Mechanical engineering from Sharif University of Technology (SUT), Tehran, Iran in 2012 and 2019. He is currently working as a postdoctoral researcher in Ferdowsi University of Mashhad (FUM). His research interests are Artificial Intelligence, Robotics, Model Predictive Control, Robust Control, Fault Tolerant Control, and System Identification.
	Mohammad Hossein Fallahi is pursuing his Master's degree in Mechanical Engineering at Sharif University of Technology. His research interests lie in the application of artificial intelligence and optimization methods to industrial and medical systems. His academic interests include Robotics, Control, Reinforcement Learning, and Computer Vision. He is particularly motivated by the integration of intelligent algorithms and engineering principles to develop innovative solutions for complex human and industrial challenges.
	Dr. Seyed Abdolmajid Yousefsani is an Assistant Professor of Mechanical Engineering at Ferdowsi University of Mashhad (FUM). He received his Ph.D. from Sharif University of Technology, Tehran, Iran in 2019, and then pursued two postdoctoral fellowships at EPFL, Switzerland and Stanford University, California, USA. His research interests include Applications of Artificial Intelligence in Applied Mechanics, Smart Materials and Structures, Biomechanics of Biological Tissues, and Mechanics of Biomaterials.