Safe Navigation for Mobile Robots with Motion Constraints and Traffic Rules

Yuting Tao, Mingyang Li, Xiao Cao, Peng Lu*

***Abstract*—Deep reinforcement learning (DRL), which integrates neural networks with reinforcement learning algorithms, plays a crucial role in enhancing autonomous robot navigation in various environments. However, existing research has encountered limitations, such as relying on conventional algorithms to ensure safety, failing to consider the motion constraints of robots, and disregarding traffic rules for robots moving in crowds. To address these challenges, we proposed a novel end-to-end method that integrates the mobile robot's motion constraints and traffic rules to ensure safe movements and interactions in crowds. By designing a novel observation space and a Long Short-Term Memory module, our DRL framework can output two continuous actions, linear and angular velocities for collision avoidance in dense environments, only using the depth image from the onboard RGBD camera. We have designed a novel reward function that considers motion constraints and traffic rules for differential drive robots, facilitating efficient and safe obstacle avoidance. We have extensively validated the proposed framework against state-of-the-art methods using the BARN_Dataset map set. Furthermore, various real-world scenarios, including structured, unstructured, and random environments, and moving pedestrians, have further demonstrated the effectiveness and robustness of our proposed framework.**

I. Introduction

In autonomous systems, the ability of mobile robots to navigate efficiently and safely in dynamic and potentially unfamiliar environments stands as a cornerstone for numerous applications. Despite significant advancements in mobile robotics and artificial intelligence, collision avoidance remains challenging, primarily due to the unpredictable nature of environments and complex decision-making. Rule-based approaches, such as the Dynamic Window Approach [1], and the Timed Elastic Band [2] are common in standard navigation scenarios. Moreover, Velocity Obstacle (VO) [3] and Reciprocal Velocity Obstacle (RVO) [4] methods are designed for dynamic obstacle cooperative avoidance. These methods rely on additional perception methods that map environments with obstacle information and estimate their motion. Tuning of multiple modules is highly complex and time-consuming. In contrast, DRL-based methods provide an end-to-end solution from observation to action, which enables efficient navigation in complex environments [5], [6], [7]. Many DRL-based methods cannot work alone and rely on conventional methods for challenging scenarios [6] or global path planning [8]. Other DRL-based methods only cope with simple scenarios [5]. Moreover, to generate a smooth trajectory for the robot to follow, existing DRL-based methods just enforce a penalty on the angular velocity and do not provide a systematical solution on how to guide the agent to generate smooth trajectories [6], [8]. Furthermore, some DRL methods also consider traffic rules [9], [10] to facilitate social behaviors and effective collision avoidance. However, since these rules only apply to USVs, we aim to introduce specific traffic rules for mobile robots to enhance their safety during obstacle avoidance in crowds.

The authors are with the Department of Mechanical Engineering, The University of HongKong, HongKong SAR, China (e-mail: taoyt@connect.hku.hk; myli0823@connect.hku.hk; u3009678@connect.hku.hk; lupeng@hku.hk).

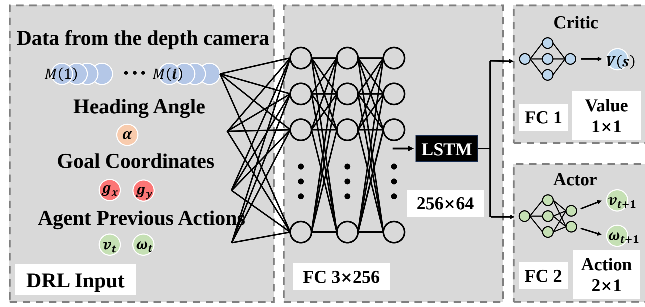

Fig. 1. DRL Neural Network with three FC layers and an LSTM module.

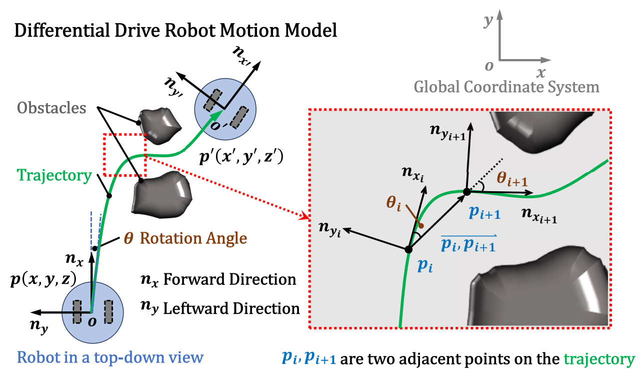

Fig. 2. Differential drive robot trajectory analysis. The left depicts the trajectory of a mobile robot, while the right provides a motion analysis of two adjacent points on any segment of the trajectory. p_i and p_{i+1} are two consecutive points on the trajectory, and n_{x_i} and n_{y_i} indicate the forward and leftward direction vectors in the robot's coordinate system at position p_i.

II. Proposed Method

We model the collision avoidance challenge in mobile robotics as a partially observable Markov decision process (POMDP) [11], defined by the tuple $(\mathcal{S}, \mathcal{A}, \Omega, \mathcal{O}, \mathcal{T}, \mathcal{R})$, the neural network's architecture shown in Fig. 1. The state space vector is designed as $\boldsymbol{o}^t = [\boldsymbol{m}^t, \boldsymbol{h}^t, \boldsymbol{g}^t, \boldsymbol{v}^{t-1}, \boldsymbol{\omega}^{t-1}]$, including minimal distance data from depth camera $\boldsymbol{m}^t$, the heading angle towards the goal $\boldsymbol{h}^t$, the coordinates of the goal $\boldsymbol{g}^t$, and the last linear and angular velocity $\boldsymbol{v}^{t-1}$, $\boldsymbol{\omega}^{t-1}$. Specifically,

TABLE I

PERFORMANCE METRICS EVALUATED FOR DIFFERENT METHODS

Metrics	Method	Success Rate	Average Time (s)	Average Length (m)	Average Speed (m/s)	Decision-making Time (ms)
	DWA [1]	0.72	18.72	9.61 / 0.48	0.51	31.53
	TEB [2]	0.64	18.77	9.55 / 0.49	0.51	33.47
V_max = 1.0 m/s	DRL_VO [8]	0.66	13.04	9.12	0.70	64.55
	Ours w/o MC	0.78	14.07	9.90	0.72	19.81
	Ours w/ MC	**0.82**	**12.17**	**9.05**	**0.75**	**19.61**

Note: 'Ours w/o MC' stands for 'Ours without Motion Constraint', and 'Ours w/ MC' stands for 'Ours with Motion Constraint'. FOV = 90°.

the 2D depth map from the depth camera is processed to the distance information by one-dimensional minimum pooling, aiming to distill the map into a streamlined form. Then action space $\boldsymbol{a}^t = [\boldsymbol{v}^t, \boldsymbol{\omega}^t]$ includes linear and angular velocity.

The first part of the reward function incentivizes the agent to reach the target and penalizes collisions or deviations from the goal: $r_u = +1$(arrived); +0.001(approaching); -1(collision); -0.001(going away). Secondly, $r_{time} = -\frac{1}{Maxstep}$ encourages expedited navigation. Then, the motion constraint reward function specifically applicable to differential drive ground robots is $r_{traj} = 0.001 - \|\mathcal{F}\|$. The trajectory can be defined by arc segments, each bounded by two points $\boldsymbol{p_i}$ and $\boldsymbol{p_{i+1}}$, as depicted in Fig. 2. In accordance with the chord-tangent angle theorem, $\sin\theta_i$ is equal to $\sin\theta_{i+1}$. $\boldsymbol{n_{x_i}}$ and $\boldsymbol{n_{x_{i+1}}}$ represent the absolute attitude of the robot at positions $\boldsymbol{p_i}$ and $\boldsymbol{p_{i+1}}$ in the global coordinate system, the following relationship is established: $\boldsymbol{n_{x_i}} \times \boldsymbol{p_i p_{i+1}} = \boldsymbol{p_i p_{i+1}} \times \boldsymbol{n_{x_{i+1}}}$. This leads to the formulation of the constraint function $\mathcal{F}$:

$$\begin{aligned} \mathcal{F} &= \boldsymbol{n_{x_i}} \times \boldsymbol{p_i p_{i+1}} - \boldsymbol{p_i p_{i+1}} \times \boldsymbol{n_{x_{i+1}}} \\ &= (\boldsymbol{n_{x_i}} + \boldsymbol{n_{x_{i+1}}}) \times \left(x_{i+1} - x_i, y_{i+1} - y_i, 0\right)^T \end{aligned} \tag{1}$$

when the magnitude of the objective function $\|\mathcal{F}\|$ is close to 0, a positive reward is received. Conversely, when deviating from the prescribed trajectory, a negative reward is incurred. Additionally, the reward function of velocity constraints, aiming to balance the agent's travel speed and moderate rapid changes in velocity, is given by:

$$r_{vel} = 0.001 \cdot \left[\frac{1}{1+\exp(-\alpha \cdot (v_t - v_{max}/2))} - \beta \cdot (v_t - v_{t-1})^2\right] \tag{2}$$

The first part of r_{vel} is a formula similar to a Sigmoid function, encouraging the agent to prefer executing higher linear velocity, and the second part prevents excessive changes in speed, constraining the agent's acceleration. Moreover, we also add rules to ensure safe navigation: 1) safe distance from pedestrians, and 2) right-hand traffic rules. Specifically, whether the robot meets a pedestrian head-on or passes from behind, it should always move to the right side. Therefore, we have $r_{traffic} = k_1 \cdot f(d) + k_2 \cdot p$, where $f(d)$ is -0.001 when unsafe distance, and if the robot obeys the right-hand traffic rules, $p = 0.001$; if not, $p = -0.001$. Finally, the total reward function is designed as follows: $r = r_u + r_{time} + r_{traj} + r_{vel} + r_{traffic}$

III. EXPERIMENTS AND RESULTS

In our assessment, we used the BARN_Dataset to test our approach's capability to maintain a considerable speed while ensuring collision avoidance, with the results detailed in Table I. Our method showcased the highest success rate and achieved the greatest average speed among all tested algorithms, suggesting an exceptional generalization capability that significantly reduces the risk of collision.

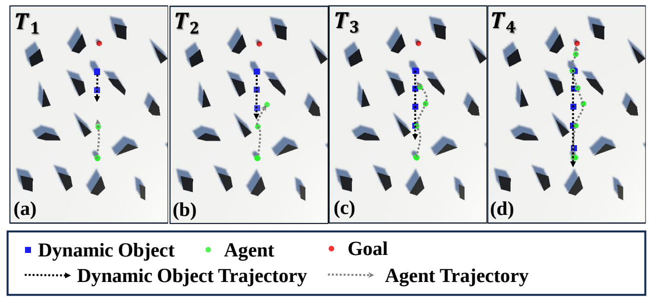

Fig. 3. An example of dynamic collision avoidance.

Furthermore, we examined whether the agent could follow the rules set in the reward function. Its collision avoidance performance is illustrated in Fig. 3, which shows the agent avoiding an oncoming dynamic object. The trajectories indicate that when the pedestrian, represented by a blue block, approaches, the agent prefers to avoid from the right side while maintaining a safe distance. This behavior results from the safe distance and right-hand traffic rules embedded in our reward function for dynamic pedestrians.

IV. CONCLUSION

We introduced a direct and efficient state input integrated with a well-designed DRL neural network framework featuring an LSTM module, extracting limited and sequential environmental information and mapping it to two continuous actions. Then we developed a novel reward function that implicitly incorporates the robot's motion constraints and traffic rules into collision avoidance policy during training. This encourages the agent to choose higher speeds with smooth and safer trajectories, thus improving obstacle avoidance. Simulations and experiments have demonstrated the effectiveness of our method for obstacle avoidance in map-less, unstructured, and complex environments.

REFERENCES

[1] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[2] C. Roesmann, W. Feiten, T. Woesch, F. Hoffmann, and T. Bertram, “Trajectory modification considering dynamic constraints of autonomous robots,” in *ROBOTIK 2012; 7th German Conference on Robotics*, 2012, pp. 1–6.

[3] D. Wilkie, J. Van Den Berg, and D. Manocha, “Generalized velocity obstacles,” in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 5573–5578.

[4] S. H. Arul and D. Manocha, “V-rvo: Decentralized multi-agent collision avoidance using voronoi diagrams and reciprocal velocity obstacles,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 8097–8104.

[5] R. Chai, H. Niu, J. Carrasco, F. Arvin, H. Yin, and B. Lennox, “Design and experimental validation of deep reinforcement learning-based fast trajectory planning and control for mobile robot in unknown environment,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 4, pp. 5778–5792, 2024.

[6] T. Fan, P. Long, W. Liu, and J. Pan, “Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios,” *The International Journal of Robotics Research*, vol. 39, no. 7, pp. 856–892, 2020.

[7] Y. Han, I. H. Zhan, W. Zhao, J. Pan, Z. Zhang, Y. Wang, and Y.-J. Liu, “Deep reinforcement learning for robot collision avoidance with self-state-attention and sensor fusion,” *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6886–6893, 2022.

[8] Z. Xie and P. Dames, “DRL-VO: Learning to navigate through crowded dynamic scenes using velocity obstacles,” *IEEE Transactions on Robotics*, vol. 39, no. 4, pp. 2700–2719, 2023.

[9] Y. Ma, Y. Zhao, Y. Wang, L. Gan, and Y. Zheng, “Collision-avoidance under colregs for unmanned surface vehicles via deep reinforcement learning,” *Maritime Policy & Management*, vol. 47, no. 5, pp. 665–686, 2020.

[10] X. Xu, Y. Cao, P. Cai, W. Zhang, and H. Chen, “Research on real-time collision avoidance and path planning of usvs in multi-obstacle ships environment,” *Ocean Engineering*, vol. 295, p. 116890, 2024.

[11] K. P. Murphy, “A survey of pomdp solution techniques,” *environment*, vol. 2, p. X3, 2000.