Deep Reinforcement Learning Based AGV Self-navigation Obstacle Avoidance Method

FENG Na, FAN Fei, XU Guanglin, YU Lianqing

(*School of Mechanical Engineering and Automation*, *Wuhan Textile University*, *Wuhan* 430072, *China*)

Abstract: A local path optimization model and obstacle avoidance strategy based on Actor-Critic algorithm is proposed for the local obstacle avoidance problem of automatic guided vehicles in a complex workshop environment. In the complex working environment of the production workshop, we analyze the automatic obstacle avoidance problem of AGV trolley, establish the front and both sides of the AGV tentacle model and Markov decision process, and describe the local obstacle avoidance path in the form of virtual tentacles. And based on deep reinforcement learning to solve the path obstacle avoidance strategy, it is applied to the AGV self-navigation system. The dynamic obstacle avoidance performance of AGV is tested through simulation experiments, and the effectiveness of the proposed algorithm is verified by completing local obstacle avoidance path planning under global path guidance.

Keywords: Obstacle Avoidance, Virtual Whisker, Deep Reinforcement Learning

1 Introduction

Automated Guided Vehicle (AGV), as an important branch of mobile robots, has been widely used in the storage industry, manufacturing workshops, outdoor hazardous locations and other fields. In manufacturing workshops, the factory layout is complex and variable, and the operating environment of AGV is very complicated, which has higher requirements on their dynamic obstacle avoidance capability [1].

Artificial potential field is a more mature algorithm in local path planning research[2]. The algorithm constructs an artificial potential field in which the obstacles encountered during vehicle motion exert repulsive forces on the vehicle and the target point exerts gravitational forces on the vehicle, thus abstracting the environmental information into repulsive and gravitational fields. However, when the vehicle travels on narrow roads, it tends to sway or oscillate in the passage, which prevents the vehicle from reaching the desired location.

With the rapid development of machine learning, methods that combine deep reinforcement learning with traditional local path planning can show advantages when solving the above problems[3]. Reinforcement learning-based control methods can iteratively optimize the control strategy in interaction with the controlled system without the need to build an accurate mathematical model of the controlled object [4]. However, the existing method takes whether the AGV reaches within a certain range of the target point as the basis for the end of obstacle avoidance, without considering the impact on the subsequent operation, and the trajectory needs to continue to be adjusted to bring the AGV back to the global path after the end of obstacle avoidance, which will affect the overall operation efficiency.

Based on the existing algorithms, this paper proposes a tentacle algorithm combined with deep reinforcement learning[5], establishes the tentacle model and Markov decision process for the front and both sides of the AGV, and analyzes the AGV obstacle avoidance problem while considering the guiding role of the subsequent path direction. Finally, simulation experiments are conducted to verify the effectiveness of the proposed method[6].

2 Modeling Front-end Tentacles

In the production workshop, the workshop aisle is complex and changeable, in order to ensure that the AGV in the safe operation of the premise of the highest possible operating efficiency, then the local obstacle avoidance ability has higher requirements. Driving in the production plant, when the AGV detects obstacles in the front side and left and right sides of the area[7], it starts to carry out local obstacle avoidance path planning. Local obstacle avoidance needs to meet the following requirements: First, the AGV does not collide with obstacles during operation, that is, in every moment of the obstacle avoidance process t , The AGV has for any obstacle $D_\mathrm{r}\cap D_\mathrm{obs}=\varnothing$, D_r indicates the area where the AGV body is located. D_obs is the area where the obstacle is located. Second, AGV do not collide with obstacles during operation. The requirement of no collision is expressed as follows: during obstacle avoidance, the AGV has no collision with any obstacle $D_\mathrm{r}\cap D_\mathrm{edge}=\varnothing$, D_r indicates the geometry of the AGV. D_edge Indicates the edge area on both sides of the aisle.

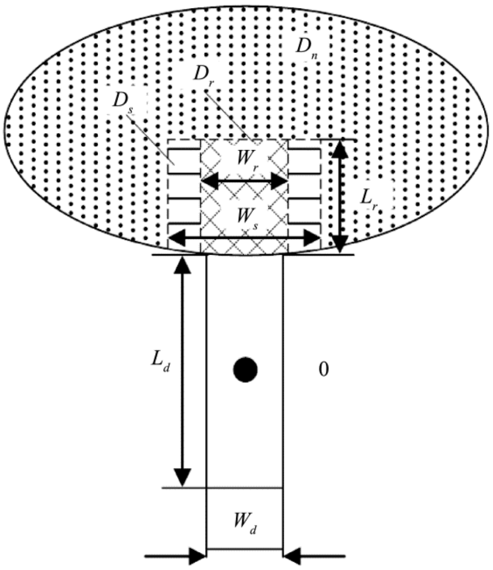

Fig.1 Virtual Touch Design

2.1 Virtual Touch Design

The virtual reach is a part of the area defined within the detection range of the obstacle avoidance sensor, which is designed to facilitate the expression of obstacle information in the front and both sides of the vehicle, and to reduce the amount of computing by processing only the environmental data within the virtual reach[8]. The virtual reach design is shown in Fig.1, where o for the AGV body mass center. D_r (L_r×W_r) indicates the area where the AGV is located.

Where the grid line area D_s (W_s×L_D) It is called the collision zone, which is used to indicate whether the vehicle body is likely to collide with the obstacle when it continues to travel along the current direction, and indicates that a collision will occur when the obstacle exists in this zone when it continues to travel along the current direction. In order to avoid anti-generated collision due to sensor error and other reasons, so W_s the design should be slightly larger than the width of the car body W_r.

2.2 Virtual Tentacle Design

As shown in Fig.2, the tentacle map is circular, starting from the center of mass of the vehicle body, and extending the tentacle map to the two regions on the left and right bounded by the body directly in front and opposite to each other[9]. The tentacle map represents the current candidate planning road, and the gap occupied by the vehicle and virtual tentacles at each point on the current candidate planning road is virtualized at a certain moment, and it is determined whether there is any overlap with the obstacle parts, and the turn path selection is carried out comprehensively.

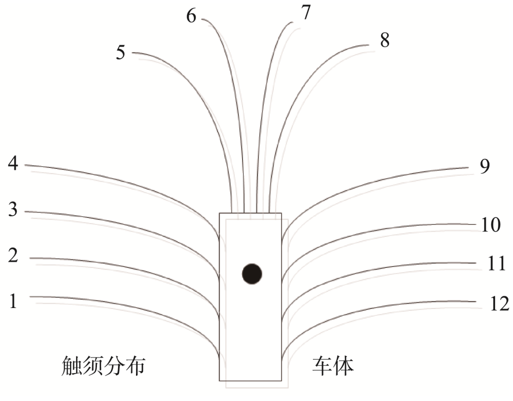

Fig.2 Virtual Tentacle Distribution State Diagram

AGV may encounter situations when performing local obstacle avoidance path planning. First, the collision zone in the virtual reach intersects with the edge area of the aisle, indicating that continuing to move along this route will result in the vehicle intersecting with the aisle, and a negative reward should be given when designing the reward function[10]. Second, both the collision zone and the warning zone in the virtual horn do not intersect with the obstacle and the aisle, indicating that it is safe to continue to move along this route, and a positive reward should be given. Third, the warning area in the virtual tentacle intersects with the obstacle, which indicates that it is safe to continue to move along this route without collision for the time being but cannot guarantee safety from the obstacle[11]. Fourth, the collision zone in the virtual horn intersects with the obstacle, which indicates that continuing to move along this route will lead to a collision between the car and the obstacle, and a negative reward should be given in the design of the reward function.

2.3 Global Path Guidance Constraints

In the local obstacle avoidance path planning, if only bypassing obstacles is planned as a constraint, the AGV may deviate significantly from the global path or travel along the direction opposite to the global path. To solve this problem, this paper adds global path guidance as a constraint to the local obstacle avoidance path planning process.

Select a point from the global path p_{target} as the target point for local path planning, while picking the next adjacent point in the global path p'_{target}. Use these two points to construct the global guidance vector; use the current AGV position point o and the target point p_{target}. Build $\vec{\xi}_{\text{o}} = o - p_{\text{target}}$ is the position vector. At the next moment, the position vector of the AGV body is given by $\vec{\xi}_{\text{o}}$ Changed to $\vec{\xi}_{\text{o}'}$ The amount of change is $\overrightarrow{\Delta\xi_{\text{o}}} = \vec{\xi}_{\text{o}'} - \vec{\xi}_{\text{o}}$.

The selected local path target point may be too close to or overlap with the obstacle, so in determining whether the AGV has completed the obstacle avoidance task, the position of the AGV under the global path guidance should reach or have crossed the local path target point as the basis for determination.

$$\begin{cases} \vec{\xi}_{\text{o}} \cdot \vec{\xi}_{\text{o}'} > 0 \\ \overrightarrow{\Delta\xi} \cdot \vec{\xi}_{\text{o}} > 0 \end{cases} \tag{1}$$

3 Markov Decision Process Construction

When AGV act as intelligent bodies, they need to sense the state of the environment to make decisions. The control process is expressed as a Markovian decision process. MDP can be represented as a quaternion array (*S*, *A*, *P*, *R*), Among them. *S* is the intelligent body state space., *A* is the intelligent body action space., *P* is the state transfer function, *R* is the reward function.

3.1 State Space

The AGV can sense the obstacle information in real time through the laser scanner installed in the front of the vehicle. S_{obs}, The navigation module can obtain its own position information in real time S_o, S_{edge} for road information such as the width and shape of the aisle in which the AGV is currently located, if known. $\vec{\xi}_{\text{target}}$ is the global path information. The state space is $S = [S_{\text{obs}}$、S_o、S_{edge}、$\vec{\xi}_{\text{target}}]$.

3.2 Action Space

The action space is the control command for AGV, in the local obstacle avoidance problem that is the local obstacle avoidance path of dynamic planning, this paper describes the local obstacle avoidance path through the form of virtual tentacles[12]. The turn radius r_k of k the tentacle is expressed as:

$$r_k = \begin{cases} \rho^k R_{\min}, k = 0, \cdots, (n-3)/2 \\ \infty, k = (n-1)/2 \\ -\rho^{k-7} R_{\min}, k = (n+1)/2, \cdots, n-1 \end{cases} \tag{2}$$

Where, $R_{\min}$ is the set minimum limit value of turning radius, ρ is the setting factor, the value of ρ is related to the density of tentacle distribution, the larger the value of ρ, the denser the E tentacles. n is the total number of tentacles, k is the tentacle number.

The action space is A=[$R_{\min}$、ρ、n、k、*stop*], where stop is the stop signal.

3.3 Reward Functions

The goal of local obstacle avoidance path planning is to complete the obstacle avoidance task under the constraints that the vehicle body does not collide with the obstacle, the vehicle body does not intersect with the edge of the aisle, the local obstacle avoidance path is as smooth as possible, and the global path is guided. The reward function in this paper is designed as follows:

$$R = r^q + r^n + r^c + r^D \tag{3}$$

The tangential running reward indicates the reward received by the AGV for moving in the global path direction, and when the AGV is backing up, a negative reward is given.

Normal running reward indicates the reward received by the AGV when it is approaching or deviating from the global path, and the amount of change of the path in the global path normal direction is $\Delta d_t^n = \left\|\vec{\xi}_{o'}^n\right\| - \left\|\vec{\xi}_{o}^n\right\|$, When $\Delta d_t^n > 0$ it means that the AGV deviates from the global path and should be given a negative reward.

Crash Bonus $r_t^c = -100$ It is the penalty that the AGV gets after collision or with an obstacle to guide the AGV to avoid the obstacle.

r^D is the bonus value earned for choices made during the process of making virtual tentacle path selections, calculated as:

$$r^D = \begin{cases} -1.5\ , D_s \cap D_{\text{edge}} \neq \varnothing \\ 1\ , D_b \cap D_{\text{edge}} = \varnothing、 D_{\text{b}} \cap D_{\text{obs}} = \varnothing \\ 0\ , D_{\text{b}} \cap D_{\text{obs}} \neq \varnothing \\ -1.5\ , D_{\text{s}} \cap D_{\text{obs}} \neq \varnothing \\ -0.1\ , D_b \cap D_{\text{edge}} \neq \varnothing、 D_{\text{b}} \cap D_{\text{obs}} \neq \varnothing \end{cases} \tag{4}$$

3.4 Deep Reinforcement Learning Based Obstacle Avoidance Strategy Solving

The local obstacle avoidance path planning problem eventually needs to be solved to obtain the optimal obstacle avoidance policy, and the state space and action space are continuous, and the Actor-Critic algorithm in the deep reinforcement learning algorithm is used to train the policy.

Among them, Actor refers to the policy network μ_θ, that is, obstacle avoidance strategy, according to the AGV current obstacle information, position and other state information S output action A to perform local obstacle avoidance path planning, to complete the interaction between the intelligent body and the environment.

$$A = \mu_\theta(S) + N \tag{5}$$

where θ is the policy neural network parameters, N is the action perturbation, which is used to better explore the environment during the learning process. The perturbation function uses a time-uncorrelated, zero-mean Gaussian noise, and the perturbation term will be removed when testing the strategy network.

The strategy network estimates the future return q of the output action of strategy network μ_θ through the value network, and the strategy neural network parameters are updated by the back propagation algorithm in the input loss function as follows.

$$q = Q_w(S, \mu_\theta(S)) \tag{6}$$

Where w is the value neural network parameter.

Critic refers to the value neural network Q_w. The value network learning unit calculates the value q through the value network based on the training samples, and q_target through the target strategy with the target value network, which is input to the loss function to update the value neural network parameters through ack propagation algorithm.

$$q\_\text{target} = R + \gamma * Q_{\bar{w}}(S', \mu_{\bar{\theta}}(S')) \tag{7}$$

the where γ is the reward discount factor, $\bar{w}$ is the target value neural network parameter, and $\bar{\theta}$ is the target strategy neural network parameter.

The relevant parameters in the algorithm are shown in Table 1.

Table 1 Algorithm Parameter Setting

Parameter Name	Parameter Value
Gamma	0.99
Learning Rate	0.0001
Replay Size	300000
Training Rounds	800

3.5 Obstacle Avoidance Strategy Training

In this paper, we use Python programming language version 3.6 to build a simulation environment to simulate AGV operation, and use TensorFlow to build an AC algorithm neural network model with AMD Ryzen7-4800 processor and Nvidia RTX 2060 graphics card as the system hardware.

In the training of obstacle avoidance strategy, obstacles, the global path of AGV and the initial position of AGV are set randomly in each round, and the AGV interacts with the environment and uses AC algorithm to train the obstacle avoidance strategy. It also serves as a dynamic barrier for other AGV to operate. The relevant parameters of the strategy training simulation environment are shown in Table 2.

Table 2 Parameter Settings of Obstacle Avoidance Strategy Training Environment

Parameter Name	Parameter Value
Training Environment Size	10m x 2m
Global Target Point Location	5M Away from the Start of Obstacle Avoidance.
Obstacle Avoidance Radius	0.5m
Simulation Update Frequency	10Hz

Based on the constructed simulation environment and the algorithm process to train the obstacle avoidance strategy[13], the reward values obtained in each round of the training process are shown in Fig.3. From the figure, it can be seen that a larger reward can be obtained in the late training period and the reward value maintains a smooth trend, indicating that the algorithm has converged. The distribution and state of the obstacles in each round of the strategy training are set randomly, so the reward value will fluctuate.

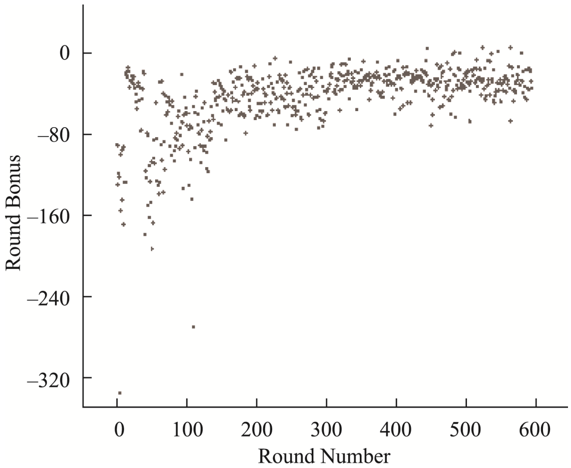

Fig.3 Strategy Training Reward

4 Simulation Experiments and Analysis

In order to investigate the effectiveness of AGV local path optimization and obstacle avoidance, this paper designs a straight section running and turning section running path experiments, in which AGV needs to continuously avoid obstacles to complete the operation. The AGV running path is shown in Fig.4 and Fig.5.

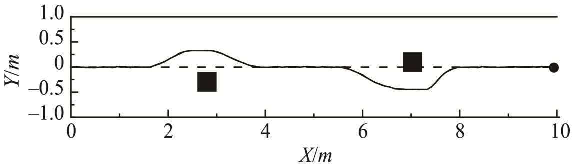

Fig.4 Straight Section Running Path

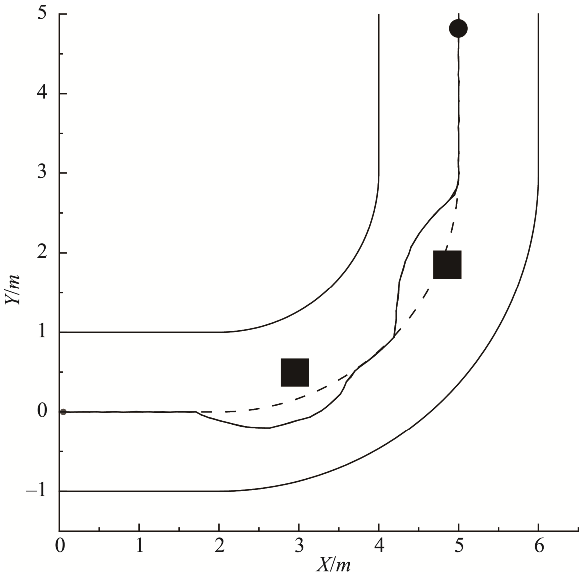

Fig.5 Turning Section Running Path

As shown in Fig.4 and Fig.5, the straight section running path and the turning section running test are shown respectively, where the straight section test has obstacles placed at (x=3, y=-0.58) and (x=6.5, y=0.7), and the turning section test has obstacles placed at (x=3, y=0.5) and (x=4.9, y=1.8).

The experimental results show that the proposed method can plan a safe obstacle avoidance path for the AGV, and the planned obstacle avoidance path deviates from the global path to a small extent[14], and can find a fast path to get away from the obstacle as soon as possible, and the obstacle avoidance process is smooth, and the transition of the obstacle avoidance path is rounded, and the obstacle avoidance task is completed under the premise of ensuring safety.

5 Conclusion

This In this paper, we design a local obstacle avoidance path planning scheme based on deep reinforcement learning for the problem of collision-free operation of AGV in complex and variable aisles in production workshops. The main research contents include: analyzing the AGV obstacle avoidance problem in the production floor environment, establishing the AGV front-end tentacle model and Markov decision process, path avoidance strategy based on deep reinforcement learning, and applying it to the AGV self-navigation system[15]. The dynamic obstacle avoidance performance of the AGV is tested through simulation experiments to verify the effectiveness of the algorithm, and the experimental results show that the proposed method can plan a safe driving path for the AGV.

References

[1] K. Iagnemma, S. Kang, H. Shibly, S. Dubowsky, Online terrain parameter estimation for wheeled mobile robots with application to planetary rovers, IEEE Transactions on Robotics, 20(5)(2004): 921-927.

[2] C. M. Gifford, E. L. Akers, R. S. Stansbury, A. Agah, Mobile robots for polar remote sensing, The Path to Autonomous Robots, Springer US, (2009): 1-22.

[3] Tan ZB, Zhao JX, et al. Smooth path obstacle avoidance algorithm for non-360° detection range four-wheeled navigation vehicles[J]. Robotics, 2013, 35(5): 527-534.

[4] Sun Lixiang, Sun Xiaoxian, Liu Chengju, et al. Deep reinforcement learning-based obstacle avoidance algorithm for mobile robots in crowded environments[J]. Information and Control, 2022, 51(1): 107-118.

[5] Liu Binyan, Ye Xiongbing, Wang Xinbo, et al. An unmanned ground vehicle path avoidance algorithm based on improved artificial potential field[J]. Chinese Journal of Inertial Technology, 2020, 28(06): 769-777.

[6] Wu, X. G., Liu, S. W., Yang, L., et al. Deep reinforcement learning-based ramp gait control method for bipedal robots[J]. Journal of Automation, 2021, 47(8): 1976-1987. Niu Runxin, Mei Tao, Xia Jingting, et al. Intelligent vehicle autonomous driving and obstacle avoidance based on haptic algorithm construction and modification[J]. Automotive Engineering, 2010, 32(12): 1083-1087.

[7] Yuan J, Sun F C, Huang Y L. Trajectory generation and tracking control for double-steering tractor-trailer mobile robots with on-axle hitching[J]. IEEE Transactions on Industrial Electronics, 2015, 62(12): 7665-7677.

[8] Zheng, Silver D, Huang A, Maddison C J, et al. Mastering the game of Go with deep neural networks and tree search[J]. Nature, 2016, 529 (7587): 484–489.

[9] WANG Yuan-xi, YU Ya-li, ZHANG Guo-sheng, et al. Fuzzy Auto-adjust PID Controller Design of Brushless DC Motor[C]//2012 INTERNATIONAL CONFERENCE ON MEDICAL PHYSICS AND BIOMEDICAL ENGINEERING (ICMPBE2012), 33, 2012: 1533-1539.

[10] Zhang Minghuan, Zhang Ke, Zhang Yuchen. Research on whisker algorithm for vehicle autonomous obstacle avoidance [J]. Mechanical Science and Technology, 2012, 31 (12): 1993-1996

[11] L. Ding, H. B. Gao, Z. Q. Deng, Z. J. Li, K. R. Xia, G. R. Duan, Path-following control of wheeled planetary exploration robots moving on deformable rough terrain, The Scientific World Journal, (2014), 2014.

[12] Xu Dan. Design and development of AGV assembly line control system [D]. Shanghai: Shanghai Jiaotong University, 2016

[13] Wu Ningqiang, Li Wenrui, Wang Yanxia, et al. Research on tracking algorithm and motion characteristics of heavy-load AGV vehicles [J]. Journal of Chongqing University of Technology (Natural Science), 2018 (10): 53-57

[14] Ibari B, Benchikh L, Hanifi EAR, et al. Backstepping approach for autonomous mobile robot trajectory tracking[J]. Indonesian Journal of Electrical Engineering and Computer Science, 2016, 2 (3): 478-485.

Author Biography

FENG Na is currently a M.Sc. candidate in Wuhan Textile University. Her main research interests include intelligent robot detection technology and system.

E-mail: 835236757@qq.cm