2022 International Conference on Image Processing, Computer Vision and Machine Learning (ICICML) | 978-1-6654-6468-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICICML57342.2022.10009821

A Distributed Multi-Robot Path Planning Algorithm for Searching Multiple Hidden Targets

First Author（*Corresponding Author*）: Luxi Zhang
Affiliation: Donghua University
e-mail: 2201839@mail.dhu.edu.cn

Second Author: Jie Qi
Affiliation: Donghua University
e-mail: jieqi@dhu.edu.cn

***Abstract*—This paper proposes a new distributed multi-robot path planning algorithm based on fuzzy logic control and reinforcement learning, which can navigate for robots searching for hidden targets in unknown environment. The algorithm is composed of two controllers, a fuzzy logic controller based on multiple behavior coordination strategy and a policy controller based on deep reinforcement learning. The first controller is used for roaming search to find the hidden targets, and the role of the second controller is to navigate from the current position to the targets. To check the performance of the algorithm, we simulate it in simulation environment built in the CoppeliaSim simulation software and implemented by e-Puck robots. The simulation results demonstrate the effectiveness of the proposed method.**

Keywords- distributed multi-robot path planning, fuzzy logic control, deep reinforcement learning, CoppeliaSim

I. INTRODUCTION

Mobile robots are one of the research hotspots in today's artificial intelligence industry. They can imitate human activities, thereby liberating humans from high-intensity or high-repetition work, and assisting humans in completing some special tasks. Path planning technology [1-5] is one of the key technologies for robots to imitate human behavior to complete tasks. And at present, there are many mature methods, which can be roughly divided into: classical algorithms, intelligent optimization algorithms and artificial intelligence algorithms.

Most of the classical algorithms for mobile robot path planning are oriented to scenarios where the environment is known and the robot is positioned accurately. Dong et al.[6] present skilled-RRT for regular 2-dimensional (2D) building environments. Wang et al.[7] propose a novel learning-based multi-RRTs (LM-RRT) approach for robot path planning in narrow passages. The intelligent optimization algorithm [8] realizes the search or planning of the path by simulating the behavior of animals and plants and various natural phenomena in nature. You et al. [9] introduced a dynamic search induction operator in the ant colony algorithm speeding up the convergence speed and completing the planning task well in a complex environment. Wang et al. [10] proposed an offline path planning method based on the IQPSO algorithm. Wei et al. [11] combined fuzzy control with improved artificial potential field method to solve the trap problem in path planning. The artificial intelligence methods are mainly based on algorithms based on deep learning and reinforcement learning. Chen et al. [12] designed a bidirectional neural network for path planning. Yao et al. [13] proposed a path planning method that combines improved black-hole potential field and reinforcement learning.

This paper proposes a distributed multi-robot search multi-objective path planning method based on fuzzy logic control and reinforcement learning. The method combines the process of searching for hidden targets based on multi-behavior fusion fuzzy logic control and the process of reaching targets based on reinforcement learning algorithm, so that the distributed control of multi-robots realizes real-time path planning for searching hidden targets in unknown environments.

The structure of this paper is as follows: the chapter II introduces the algorithm framework; the design of the fuzzy logic controller and the reinforcement learning-based policy controller are respectively in the chapter III and the chapter IV; the chapter V shows the simulation verification experiments and results; the summary and outlook are in the chapter VI.

II. THE FRAMEWORK OF THE PROPOSED ALGORITHM

The method proposed in this paper consists of two controllers, one is a fuzzy logic controller with multiple behavior coordination strategies, and the other is a policy controller based on reinforcement learning (RL) and trained with the Proximal Policy Optimization (PPO) algorithm. The two controllers act on the target searching process and the target reaching process, respectively. The control flow of the i-th robot is shown in Figure1:

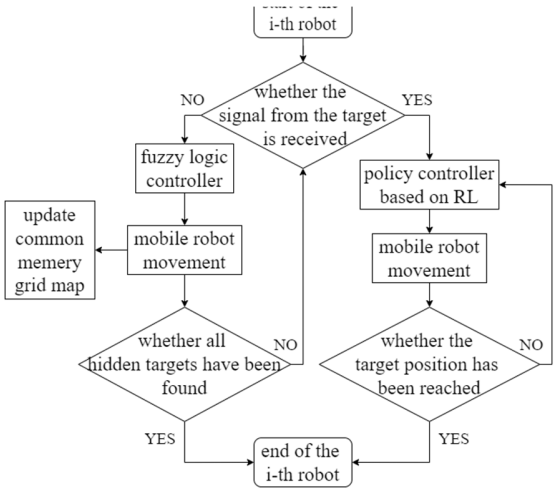

Figure1. The control flow diagram of the i-th robot (where the memory grid map is maintained by all robots)

Authorized licensed use limited to: UNIVERSIDADE DE BRASILIA. Downloaded on November 26,2023 at 18:44:30 UTC from IEEE Xplore. Restrictions apply.

III. FUZZY LOGIC CONTROLLER BASED ON MULTIPLE BEHAVIOR COORDINATION STRATEGY

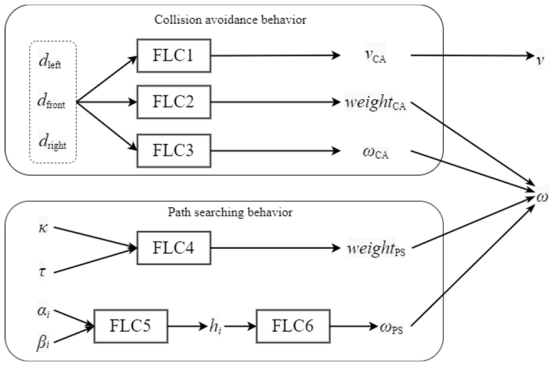

Figure2. Schematic diagram of fuzzy logic controller with multi-behavior coordination strategy

Aiming at the process of searching for targets in an unknown environment, a fuzzy logic path planning algorithm based on memory grids is proposed, and a variety of behavior coordination strategies are used. This paper designs a fuzzy logic framework (as shown in Figure2) which coordinate of collision avoidance behavior and path search behavior.

A. *Quantification of environmental information*

The search neighborhood of the robot in memory grid map is defined as an area centered on the robot with an area of $n*n$ divided into three areas: left, front and right according to the direction of the robot. We set 4 characteristic indices for the robot, which are trajectory strength κ, obstacle strength τ, repetition risk α and collision risk β. The trajectory strength κ and obstacle strength τ are defined as the number of trajectory points and obstacle points in the search neighborhood divided by the neighborhood area, respectively. Repetition risk α_i and collision risk β_i (where $i = 1,2,3$) are defined as the number of track points and the number of obstacle points in the left, front, and right regions, respectively.

Divide robot's proximity sensor readings into three area and take the minimum distance read by the sensor in this area as the obstacle distance d_{left}, d_{front} and d_{right} in the area.

B. *The controllers in collision avoidance behavior*

The collision-avoidance (CA) behavior is a kind of sensor-based behavior. Three fuzzy logic controllers are involved in the collision avoidance behavior: FLC1, FLC2 and FLC3. Their inputs are all preprocessed obstacle distances d_{left}, d_{front} and d_{right}, which are represented by the fuzzy set {very near, near, far}, and the membership function is shown in the Figure3.

The output of FLC1 is the linear velocity v of the mobile robot, which is represented by a fuzzy set {stop, slow, fast}, and the membership function is shown in the Figure3. The output of FLC2 is the weight index $weight_{\text{CA}}$ of the CA behavior, which is represented by the fuzzy set {small, medium, large}, and its membership function is similar to d. The output of FLC3 is the recommended turning angle ω_{CA} of the CA behavior, which is represented by a fuzzy set {LB, LS, FW, RS, RB}, and its membership function is shown in the Figure3, where L(left) means turning left, R(right) means turning right, and B(big) means large angle, S(small) means small angle, FW means no turning.

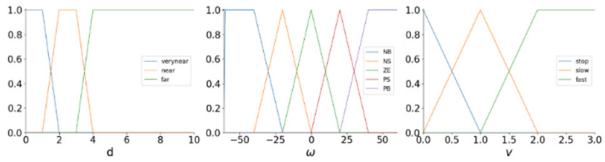

Figure3. The memberships of d, ω_{CA} and v

C. *The controllers in Path Search Behavior*

The path-searching (PS) behavior calculates the minimum risk angle of the mobile robot according to the information extracted from the memory grid to avoid walking through the previous trajectory and obstacles, and improve the efficiency of roaming search. We design three fuzzy logic controllers in the PS behavior: FLC4, FLC5 and FLC6.

The inputs of FLC4 and FLC5 are 4 characteristic indices from the memory grid, and the size of which is converted into a fuzzy set {low, medium, high}. The output of FLC5 is the risk index h_i of each area in the neighborhood, which represents the comprehensive risk of each area in the neighborhood of the mobile robot, represented by a fuzzy set {dangerous, uncertain, safe}. The membership function of each of them is similar to that of d shown in Figure3.And the weight index $weight_{\text{PS}}$ and the recommended angular velocity ω_{PS} of the PS behavior are similar to that of the CA behavior.

According to the walking and collision avoidance experience from human, the fuzzy rules of FLC1-6 are designed. And in this paper, we use the Mamdani inference method as the fuzzy inference engine, and use the center of gravity method for defuzzification [14].

IV. REINFORCEMENT LEARNING BASED POLICY CONTROLLER

Aiming at the task of target-oriented path planning, a path planning scheme based on reinforcement learning is proposed. The agent(robot) uses the data collected by interacting with the environment to update the strategy function through the PPO algorithm. After a period of training, the strategy function can decide the actions that the mobile robot should perform according to the real-time environmental information.

A reinforcement learning problem can be described as a Markov decision process $M = (S, A, P, R, \gamma)$, which consists of a state space S, an action space A, state transition probabilities P, a reward function R, and a discount factor γ, then maximize the return: $G_t = \sum_{k=0} \gamma^k r_{t+k+1}$.

A. *The Design of State Space, Action Space and Reward Function*

Each state value in the state space consists of three parts: $s_{targ} = [d, \theta]$ indicates the relative position of the robot to the target point, where $d \in \mathbb{R}$ and $\theta \in (-\pi, \pi]$; $s_{prox} \in \{0,1\}$ indicates the detection status of the sensor equipped with the robot; $s_{vis} = \{0,1,2,3\}$ indicates the observation status of the target by the vision sensor. "0" means the target leaves the detection range of the robot's vision sensor; "1", "2", and "3"

Authorized licensed use limited to: UNIVERSIDADE DE BRASILIA. Downloaded on November 26,2023 at 18:44:30 UTC from IEEE Xplore. Restrictions apply.

represent the target point appears within the vision sensor range, and is located in front of it to the left, right, or right, respectively.

Define the action space $A = \{\text{LB}, \text{LS}, \text{FW}, \text{RS}, \text{RB}\}$.

According to the needs of the task to reach the goal and avoid obstacles, define the reward as $r_t = r_t^{base} + r_t^{OA} + r_t^{vis}$.The $r_t^{base} = \|d_t\| - \|d_{t+1}\|$ is the basic reward, which guides the robot to move to the target; the r_t^{OA} is the obstacle avoidance reward, if the robot forward with an obstacle ahead, then $r_t^{OA} = -c_0$, otherwise $r_t^{OA} = 0$; the r_t^{vis} is the target coverage reward, which guides the robot to remain within the detection range of the vision sensor to the target and the value is shown in the TABLEI, where the hyperparameters c_0, c_1, c_2 and c_3 satisfy $c_0 > c_1 > c_2 > c_3 > 0$.

TABLEI. Table for calculating goal coverage rewards

s_{vis} \ a	LB	LS	FW	RS	RB
0	0	0	0	0	0
1	c_1	c_2	$-c_3$	$-c_2$	$-c_1$
2	$-c_2$	$-c_3$	c_2	$-c_3$	$-c_2$
3	$-c_1$	$-c_2$	$-c_3$	c_2	c_1

B. *The design and training method of policy controller*

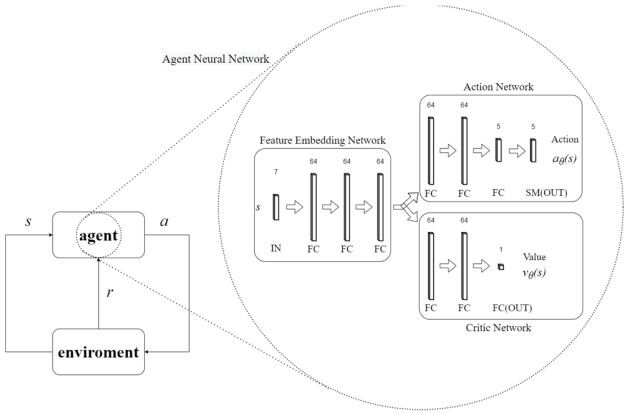

Figure4. The Policy Controller Based on Reinforcement Learning

We parameterize the agent as the agent neural network including: feature embedding network, actor network and critic network, as shown in Figure4. In the feature embedding network, we use three fully connected (FC) layers to compute the state embedding. The actor network consists of three FC layers and one softmax (SM) layer. The last SM layer is the output (OUT) layer and outputs the probability of selecting action a at state s according to policy π_θ. The critic network consists of three FC layers, and the last FC layer is the OUT layer, which outputs the estimated value of the state s.

In order to realize the update optimization of the network, we use PPO algorithm [15] to train the agent neural network.

V. Simulation Experiments

A. *Selection of Simulation Robot and Construction of Simulation Experiment Environment*

The experiments take the e-puck robot as the simulation control object, which is a two-wheel differential mobile robot with simple design and rich sensors. The E-puck robot is 70 mm in diameter, 55 mm in height, and weighs 150 grams. It is equipped with a wealth of sensors, such as proximity sensors, vision sensors, audio sensors, and inertial measurement units.

We build a simulation experiment environment in CoppeliaSim simulation software, a cross-platform (Windows, MacOS, Linux) software. The simulation experiment scene we built is mainly composed of three elements: mobile robot, work space with obstacles, and the target. As shown in Figure 5, the e-puck robot in the CoppeliaSim model library is selected, and a 3*3m^2 floor is used. Then, add cylinders and cubes of different shapes as obstacles, while add collidable, measurable and detectable green cylinders as targets to be searched. In order to verify the search and rescue ability of the proposed algorithm applied to the robots, it is assumed that the pink area shown in Figure5 is the target signal range, the mobile robot can only receive the signal of the target point within this range.

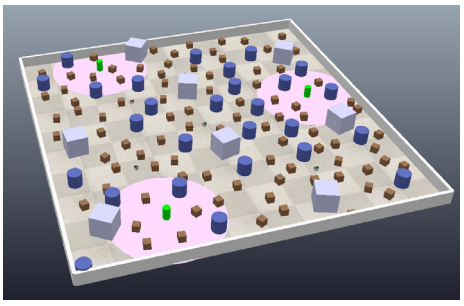

Figure5. The simulation experiment environment in CoppeliaSim

B. *The Training Results of The Agent Neural Network*

According to the PPO algorithm, the agent neural network is trained on the CoppeliaSim simulation experiment environment. The mobile robot received 30 episodes of training. At the end of each episode, the data of the current episodes was used to update the network, and the Adam optimizer was used to update the network parameters, and the learning rate was 0.001. During training, the time step required to reach the target point in each round is shown in Figure6.

Figure6. The time step of the training process varies with episode

C. *The Simulation Verification Experiment*

The proposed distributed multi-robot path planning algorithm is used to search for unknown targets in a simulation environment to verify the effectiveness of the proposed algorithm. Four cars are used to search for three unknown targets at random starting points in space. The memory grid map of the trajectory and search process is shown in Figure7. The

Authorized licensed use limited to: UNIVERSIDADE DE BRASILIA. Downloaded on November 26,2023 at 18:44:30 UTC from IEEE Xplore. Restrictions apply.

experimental results show that the proposed algorithm can make the mobile robot search and reach the target in the unknown environment.

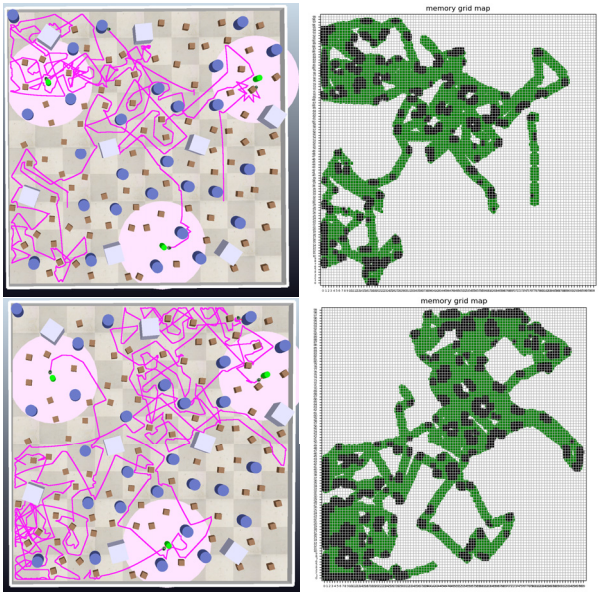

Figure7. Simulation verification experimental results

VI. Conclusion

In order to make the mobile robots complete the task of searching and reaching the hidden target in the unknown environment while avoid the unknown obstacles, we propose a path planning algorithm that combines the fuzzy logic controller based on multiple behavior coordination strategy and the policy controller based on reinforcement learning. Then the experiments in CoppeliaSim simulation software show that the e-puck robot applying the proposed algorithm is able to search for hidden objects in unknown environments.

However, in the search process, since the search area is not divided, the search efficiency is not high enough. This aspect can be studied in future work to improve the performance of the algorithm.

References

[1] Zafar, M.N. and Mohanta, J.C. (2018) Methodology for path planning and optimization of mobile robots: A review. Procedia computer science, 133, pp.141-152.

[2] Zhang, H.Y., Lin, W.M. and Chen, A.X. (2018) Path planning for the mobile robot: A review. Symmetry, 10(10), p.450.

[3] Costa, M.M. and Silva, M.F. (2019) A survey on path planning algorithms for mobile robots. In 2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC) .Portugal. pp. 1-7.

[4] Patle, B.K., Pandey, A., Parhi, D.R.K. and Jagadeesh, A. (2019) A review: On path planning strategies for navigation of mobile robot. Defence Technology, 15(4), pp.582-606.

[5] Karur, K., Sharma, N., Dharmatti, C. and Siegel, J.E. (2021) A survey of path planning algorithms for mobile robots. Vehicles, 3(3), pp.448-468.

[6] Dong, Y., Camci, E. and Kayacan, E. (2018) Faster RRT-based nonholonomic path planning in 2D building environments using skeleton-constrained path biasing. Journal of Intelligent & Robotic Systems, 89(3), pp.387-401.

[7] Wang, W., Zuo, L. and Xu, X. (2018) A learning-based multi-RRT approach for robot path planning in narrow passages. Journal of Intelligent & Robotic Systems, 90(1), pp.81-100.

[8] Mac, T.T., Copot, C., Tran, D.T. and De Keyser, R. (2016) Heuristic approaches in robot path planning: A survey. Robotics and Autonomous Systems, 86, pp.13-28.

[9] Xiaoming, Y., Sheng, L. and Jinqiu, L. (2017) An ant colony algorithm with dynamic search strategy and its application in robot path planning [J]. Control and decision-making, 32(3), pp.552-556.

[10] Wang, L., Liu, L., Qi, J. and Peng, W. (2020) Improved quantum particle swarm optimization algorithm for offline path planning in AUVs. IEEE Access, 8, pp.143397-143411.

[11] Wei, L.X., Wu, S.K., Sun, H. and Zheng, J. (2019) Mobile robot path planning based on multi-behaviours. Control and Decision, 34(12), pp.2721-2726.

[12] Chen, Y.W. and Chiu, W.Y. (2015) November. Optimal robot path planning system by using a neural network-based approach. In 2015 international automatic control conference (CACS). Yilan. pp. 85-90.

[13] Yao, Q., Zheng, Z., Qi, L., Yuan, H., Guo, X., Zhao, M., Liu, Z. and Yang, T. (2020) Path planning method with improved artificial potential field—a reinforcement learning perspective. IEEE Access, 8, pp.135513-135523.

[14] Wang, M. (2005) August. Fuzzy logic based robot path planning in unknown environment. In 2005 International Conference on Machine Learning and Cybernetics. Guangzhou.Vol. 2, pp. 813-818.

[15] Schulman, J., Wolski, F., Dhariwal, P., Radford, A. and Klimov, O., (2017) Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.

Authorized licensed use limited to: UNIVERSIDADE DE BRASILIA. Downloaded on November 26,2023 at 18:44:30 UTC from IEEE Xplore. Restrictions apply.