STM JOURNALS

International Journal of
Advanced Robotics and
Automation Technology

Volume 2, Issue 2, 2024
July—December
DOI (Journal): 10.37591/IJARAT

https://journals.stmjournals.com/ijarat

Review IJARAT

Path Planning using DDPG Algorithm and Univector Field Method for Intelligent Mobile Robot

Jiyon Yun[1], Kangsong Ro[1,*], Jusong Pak[2], Choljin Wang[3]

Abstract

Path planning is one of the most fundamental challenging tasks in robotics and its purpose is to lead the robot from the initial position to the goal without any collision through the optimal route. With the rapid development of artificial intelligence technology, AI has been widely studied for robot path planning and a method by deep reinforcement learning (DRL) was proposed. In general, path planning methods with DRL need discrete action space and use deep Q-network (DQN). Real robot moves in continuous action space, so discontinuity of it may cause impossibility of optimization of path planning and some serious problems in combination with dynamics of the robot. Enhancing the action dimension can be used to work out this problem, but significantly increases the calculation time. In order to dispose of the issue and the optimal path planning, a new method to consider the action space of the robot as an continuous space by combination between univector field method (UVFM) and DDPG which is one of the popular reinforcement learning algorithms was proposed in this paper. Simulation results show possibility and efficiency of the proposed path planning method.

Keywords: DDPG Algorithm•univector field method•continuous action space

INTRODUCTION

The path planning is the core project of robotics. The robot path planning method can be divided into two main categories, one is traditional, while another is reinforcement learning based method.

The traditional methods include univector field method, artificial potential field method [1], genetic algorithm [3], cell decomposition, Dijkstra algorithm, A* algorithm [4] and so on.

Due to the rapid development of the robot technology and artificial intelligence (AI), much more intelligent mobile robots have been developed and researchers are intensifying the study for the robot to carry out more difficult tasks in more complex environment [1]. As traditional path planning methods cannot model the complex or unknown environment and progress the path planning, there have been numerous tries to adopt reinforcement learning (RL) into the development of unmanned mobile robot development [5,6].

Deep reinforcement learning is one section of machine learning. Different from typical machine learning such as supervised or unsupervised learning, RL is operated from interaction between the agent and the external environment. The agent obtains the surrounding environmental information through trial-and-error learning repeatedly, and continuously optimizes its policy to find the best path [7].

***Author for Correspondence**
Kangsong Ro
E-mail: GS.RO@star-co.net.kp

[1-3]Student, Department of Mechanical Engineering, Kim Chaek University of Technology, Pyongyang, DPR Korea

Received Date: July 12, 2024
Accepted Date: October 20, 2024
Published Date: December 10, 2024

Citation: Jiyon Yun, Kangsong Ro, Jusong Pak, Choljin Wang. Path Planning using DDPG Algorithm and Univector Field Method for Intelligent Mobile Robot. International Journal of Advanced Robotics and Automation Technology. 2024; 2(2): 1–9p.

© STM Journals 2024. All Rights Reserved

Nowadays, deep reinforcement learning (DRL) algorithms are extensively used for robot path planning and have become the main method of intelligent robot. Here, use just deep Q-learning [8]. Deep Q-learning is discontinuous and has slow convergence speed and low success rate. So as to get over this weakness, lots of tries to combine traditional path planning method with deep Q-network have been taken and advanced to a certain extent.

To accelerate convergence and raise accuracy of path planning, some methods to combine deep reinforcement learning with Convolutional Neural Networks (CNN) have been proposed [9]. By combining deep Q-learning with prior knowledge, researchers sped up the convergence of path planning [10]. Also, path planning method by deep Q-learning was compared with traditional ones-A* algorithm, Dijkstra algorithm [11]. According to the comparison, deep Q-learning is more accurate than traditional, but have large amount of calculation. Here, they also analyzed the limitation of deep Q-learning: it can only solve discrete path planning, so the necessity of solution for continuous environment was suggested. In discrete path planning, we may lose some important characteristics of each state and this raises some difficult problems in optimization of path planning. So as to solve this issue, a method to increase the dimension of action space, but this increases the scale of deep Q-network and delays the training [12]. A new method to consider the action space of robots in continuous ensvironments was proposed but it was limited in specific robot dynamics [13].

So as to obtain better results, in this paper, an improved path planning method was proposed.

First, DDPG-one of reinforcement learning algorithms was used for new method to study robot path planning in continuous environments.

Next, we configured reward function for DDPG algorithm and combined it with UVFM which is one of typical path planning methods, so as to accelerate convergence.

The rest of the paper is organized as follows. In section 2, background of path planning is presented. In section 3, suggests path planning method combining DDPG algorithm with UVFM. Section 4 gives simulation result and analysis of proposed method. Finally, the paper is concluded in Section 5

RELATED WORKS

Univector Field Method (UVFM)

Univector field method is a modification of the artificial potential field method. Like in potential field method, each point on the field has corresponding unique vector in UVFM. The difference between the methods is the length of the vector. Unlike the APFM, each vector has unit length in UVFM, hence its name univector. Thus, this method only has directional informations.

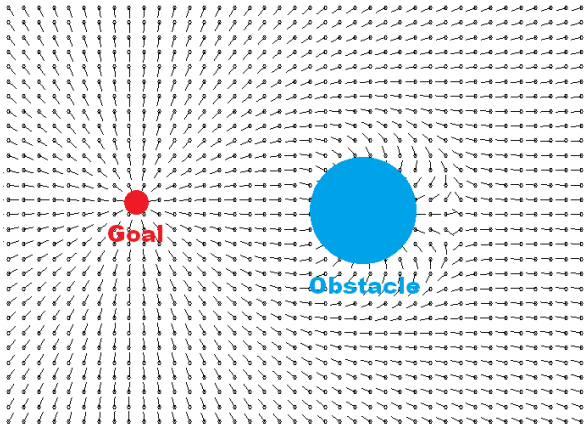

Figure 1. Univector Field.

© STM Journals 2024. All Rights Reserved

The univector field is formed by composition of two subfields. One subfield is attractive field which pull the robot into the goal position and another is repulsive which push the robot away from the obstacle. These two subfields generate the resultant univector field in the environment. After generation of the univector field, real-time control of the robot is possible by corresponding the velocity vector of the robot with the univector at that position. Figure 1 shows a simple univector field. In figure, red circle and blue circle denote attractive point-goal and repulsive point-obstacle, respectively.

However, as UVFM is based on the APFM, it also has difficulty in solution for path planning in complex environment and especially can't avoid local minimum problem.

Path Planning Method by Deep Q-Learning

Deep Q-learning is the most general deep reinforcement learning method and uses deep neural network to choose an action for each state. Thus, train the network to output Q values for all state-action pairs from the current state given as the input. Once the correct Q values are trained, the optimal policy is obtained by selecting the action which gives maximum Q value at that state.

In path planning by deep Q-learning, we discretize the environment and correspond the discrete space to the robot state.

The action space of the robot consists of finite number of actions.

The robot gains the reward after take an action by following reward function:

$$r = \begin{cases} k, & \textit{Robot is in goal position} \\ -k, & \textit{Robot is in obstacle position} \\ 0, & \textit{Robot is in other position} \end{cases} \tag{1}$$

After configuration of the environment, start training with epsilon-greedy.

At this time, training is processed by temporal difference method and the loss of the neural network is:

$$L(\theta) = [R_n + \gamma \cdot Q_\theta(P_{n+1}) - Q_\theta(P_n)] \tag{2}$$

where R_n is a reward of step n, θ is parameter of neural network, $Q_\theta(P_{n+1})$ is predicted Q value of the next state, $Q_\theta(P_n)$ is predicted Q value of the current state, γ is a discount factor.

As we configure the loss function as above, the neural network is trained by Bellman Equation to output correct Q value corresponding to state-action pair.

After training, we can use the deep Q-network to implement the robot path planning.

DDPG Algorithm

DDPG algorithm is one of the deep reinforcement learning algorithms and its best advantage is the continuity of action space. DDPG algorithm is based on actor-critic method. Here, actor network finds the action $a = \mu_\phi(s)$ at each state using policy gradient and critic network outputs Q value according to state-action pair.

During the exploration, store the transition information (current state, action, reward, next state) to the replay buffer. Then, we randomly sample a minibatch of k transitions to train both critic and actor networks. At this time, the loss function of critic network is:

© STM Journals 2024. All Rights Reserved

$$L(\theta)=\frac{1}{k}\sum_{i}\left(y_i-Q_\theta\left(s_i,a_i\right)\right)^2 \tag{3}$$

where y_i is the target value of critic, which conforms to Eq.(4).

$$y_i=r_i+\gamma Q_{\theta'}\left(s_i',\mu_{\phi'}\left(s_i'\right)\right) \tag{4}$$

r_i is reward obtained by action a_i from state s_i, a_i is output of actor network, ϕ,ϕ' are parameters of main actor and target actor, respectively, θ,θ' are parameters of main critic and target critic, respectively; $Q_\theta,Q_{\theta'}$ are Q values predicted by main critic and target critic, respectively.

Parameters of the critic network are updated by gradient descent as follows:

$$\theta=\theta-\alpha_\theta\nabla L\left(\theta\right) \tag{5}$$

where α_θ is learning rate of actor network.

As the actor network aims to maximize the Q value from the critic network so its objective function is configured as:

$$J(\phi)=\frac{1}{k}\sum_{i}Q_\theta(s_i,a) \tag{6}$$

Parameters of the actor network are updated by gradient ascent as follows:

$$\phi=\phi+\alpha_\phi\nabla_\phi J(\phi) \tag{7}$$

where α_ϕ is learning rate of actor network.

After updating main actor and critic networks, update target actor and critic networks by soft replacement.

$$\begin{aligned}\theta'&=\tau\theta+(1-\tau)\theta'\\ \phi'&=\tau\phi+(1-\tau)\phi'\end{aligned} \tag{8}$$

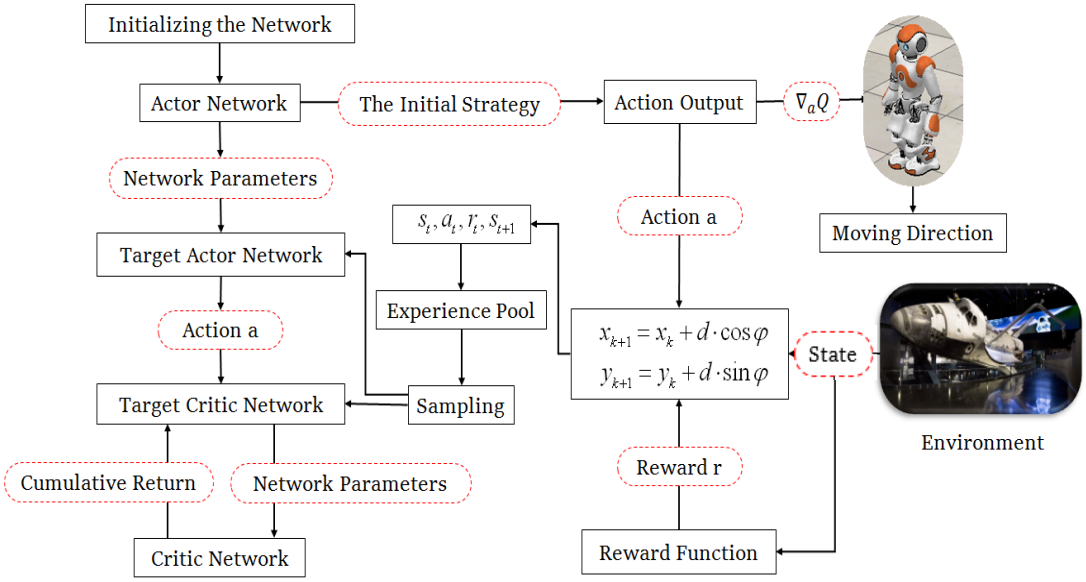

Figure 2. Schematic diagram of agent strategy learning process based on DDPG algorithm.

© STM Journals 2024. All Rights Reserved

where τ is soft replacement ratio for updating the target networks.

Repeat above algorithm for several episodes and update parameters of networks, so as to get the optimal policy.

The schematic diagram of the agent strategy learning process based on DDPG algorithm is shown in Figure 2.

PATH PLANNING BASED ON DDPG COMBINED WITH UVFM

DDPG algorithm is one of reinforcement learning algorithm and is useful as it can deal with the continuous action space. Then, how represent the action space of the robot to be continuous? This problem is settled in this paper by applying the main concept of the univector field method.

In UVFM, a virtual vector field is configured on the whole robot environment and the robot moves towards the direction of the vector corresponding to its point. In other words, each point of the environment has a unique direction for robot motion. Therefore, estimating the position of the robot and moving the robot towards the vector direction corresponding to that point leads the robot to the goal position successfully.

By combining this idea of UVFM with DDPG algorithm, robot path planning can be solved easily. So, the subject is to train the reinforcement learning networks to output desired direction for any given states. After sufficient training, we can accomplish real-time control of robot like in UVFM.

Configuration of Environment Using UVFM

In DDPG algorithm, the agent is the robot and trained DDPG networks to output motion direction of the robot from the input of its state.

The robot moves in continuous environment of specific size. In this environment, the reward function for the robot action is configured as the negative value of the distance between the robot and the target position.

$$R = \begin{cases} 100, & \textit{Reached to the goal} \\ -100, & \textit{Bumped into obstacles} \\ -100, & \textit{Be over the map bound} \\ -d, & \textit{Others} \end{cases} \tag{9}$$

So, the networks are trained to increase the reward, in other words, decrease the distance.
The state of the robot is its position information. Thus,

$$s = \begin{bmatrix} x \\ y \end{bmatrix} \tag{10}$$

The actor network gets this state as an input and outputs the motion direction angle φ. In our case, the range of the robot motion direction is $[-\pi, +\pi]$, so limit the output of the actor in the same range.

After obtaining the motion direction of the robot, move the robot with specific step length d_0 towards that direction, then the robot state changes as follows:

$$\begin{aligned} x_{k+1} &= x_k + d_0 \cos\varphi \\ y_{k+1} &= y_k + d_0 \sin\varphi \end{aligned} \tag{11}$$

where k denotes the step number.

© STM Journals 2024. All Rights Reserved

Pretraining and Acceleration of the Convergence

DDPG algorithm needs to get experience information as many as possible, so initial exploration is inevitable. But this exploration maybe pointless in many cases and causes slow convergence.

In order to accelerate the convergence, in this paper, the actor network is pretrained to initially output the directions towards the goal position from any position. To implement this idea, 100 positions are randomly selected and pretrain the actor network so that its output face to the goal.

By doing this, the ratio of exploration can be reduced in initial steps of the training. In other words, the learning rate and the noise which was added to actor output for exploration can be decreased, pointless exploration was reduced and convergence was significantly accelerated.

The univector field after pretraining of actor network for accelerating convergence is shown in Figure 3. As we can see from the figure, univectors at any position on the environment turned towards the goal.

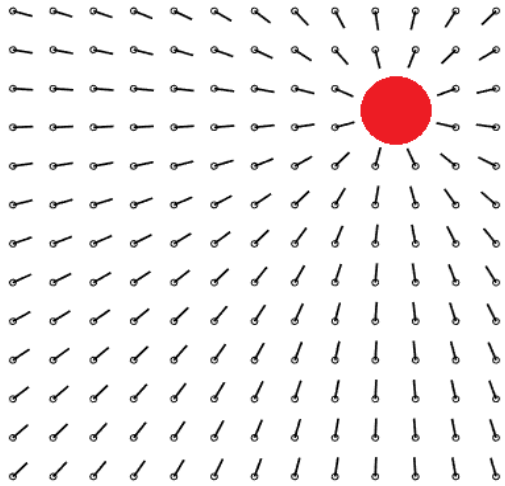

Figure 3. Pretrained environment to face to the goal.

SIMULATION RESULT & ANALYSIS

Configuration of the Environment

Simulation was advanced on the robot experiment platform *CoppeliaSim*, and DDPG algorithm was implemented with deep learning library-*tensorflow*. On the whole map size of 20×20, the initial position of the robot is (-1.3, -0.8), the target position is (1.6, 1.6) and obstacles are placed randomly. Experiments were performed on 8GB RAM, Intel Core-i7-1165G7 processor and NVIDIA RTX 3050 GPU.

The simulation was progressed by result comparison between two algorithms: DQN and proposed algorithm. In path planning by deep Q-learning, the environment was set to 20×20 grid and each cell represents the states. Action space consists of 4 actions. Table 1 shows the action space.

Table 1. Action space.

Action Index	Direction
0	Up
1	Left
2	Down
3	Right

In case of applying DDPG algorithm, as both the action space and environment of the robot are continuous, the agent can't be estimated to exactly reached to specific point. So, we accept 2 concepts:

© STM Journals 2024. All Rights Reserved

target and obstacle range. If our agent went into the target range, we estimate that it reached the goal, and that it bumped into obstacles likewise if our agent went into the obstacle ranges. The diameters of the target range and the obstacle range are 0.5, 0.3m, respectively.

Determination of Simulation Parameter

Structure of the actor and critic network are shown in following Tables. (Table 2, Table 3).

Table 2. Parameter of the actor network.

Layer	Type	Number of parameters	Activation
Input		2	
Layer1	Dense	32	tanh
Layer2	Dense	32	tanh
Output	Dense	1	π*tanh

Table 3. Parameter of the critic network.

Layer	Type	Number of parameters	Activation
Input		3(state-2, action-1)	-
Layer1	Dense	32	-
Output	Dense	1	Linear

Because the output of the actor network is motion direction of the robot, it must be in range of $[-\pi,+\pi]$, so activation function of the output is $\pi*\tanh$.

Parameter values for applying DDPG and DQN algorithm are given as following tables (Table 4, Table 5).

Table 4. Hyperparameter for DDPG algorithm.

Parameter	Definition	Value
P	Initial number of steps	500
M	Size of minibatch	64
N	Exploration noise	$0.5\times0.995^{(\text{num of training})}$
C	Updating frequency of the target network	1
$\lvert D\rvert$	Size of experience pool	10000
α	Learning rate	actor-10^{-4}, critic-10^{-3}
γ	Discount factor	0.9
d_0	Step size	0.1
τ	Soft replacement factor	0.001

Table 5. Hyperparameter for DQN algorithm.

Parameter	Definition	Value
P	Initial number of steps	500
M	Size of minibatch	64
$\mathcal{E}$	Greedy factor	Linearly decrease from 0.9 to 0.1
C	Updating frequency of the target network	2
$\lvert D\rvert$	Size of experience pool	10000
α	Learning rate	10^{-4}
γ	Discount factor	0.9

© STM Journals 2024. All Rights Reserved

Determination of Simulation Parameter

In Figure 4, path planning results by DQN and DDPG algorithm are shown. In the figure, pink and red circles denote obstacles and target range, respectively and yellow tiny circle is the agent.

As we can see from the Figure 4. Path planning result using DQN and DDPG(DQN-Left, DDPG-Right), in path planning by DQN algorithm, the route of the robot is composed some broken lines because of the discontinuous environment and action space. So, the agent lost some important motion characteristics and obtained non-optimal path.

In path planning by DDPG algorithm, robot action space is continuous and obtain smooth path which is good for robot running.

Also, the length of the path in DQN-based path planning is 4.8 while it is 3.9 in DDPG-based path planning. Thus, DDPG remarkably decrease the length of the route than DQN.

This result has demonstrated the effectiveness of the proposed path planning method.

Next, comparison data with pretrained algorithm is given in Table 6.

Table 6. Comparison between pretrained and non-pretrained DDPG algorithm.

Algorithm	Timestep	Episodes	Time
Non-pretrained Algorithm	65482	1734	56.32
Pretrained Algorithm	8594	328	8.63

As we can see from the Table 6, in case of pretraining actor network, the number of steps, episodes and convergence time have been sharply reduced.

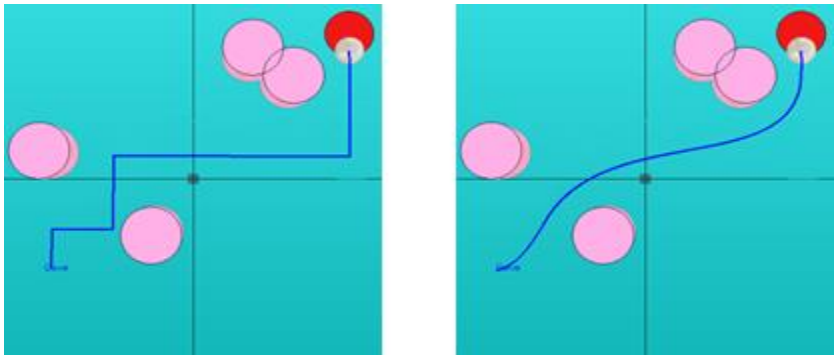

Figure 4. Path planning result using DQN and DDPG(DQN-Left, DDPG-Right).

CONCLUSION

In general, robot path planning by DRL, discretize the robot action space and find optimal path using DQN. In real world, robot action space is continuous and discretizing the action space causes impossibility of optimization of the route and some problems in combination with dynamics of the robot. In this paper, path planning method in continuous action space by combining DDPG algorithm with univector field method and pretraining method to accelerate convergence were proposed. Also, for comparison with simulation result, path planning with DQN algorithm was taken. Results show that proposed path planning method in this paper can consider continuous action space as well as guarantee correctness, efficiency and safety of the route than DQN-based method.

Conflict of Interest

The authors declare no conflicts of interest associated with this manuscript.

© STM Journals 2024. All Rights Reserved

Author Contributions

Jiyon Yun contributed to the study of conception. The study was designed by Jiyon Yun and Kangsong Ro. Literature Review was written by Kangsong Ro. Improved methods are proposed by Jiyon Yun and Kangsong Ro under supervision of Choljin Wang and Jusong Pak. Data analysis was performed by all authors. The first draft of the manuscript was written by Jiyon Yun and all authors commented on previous versions of the manuscript.

REFERENCES

1. Xianxia Liang, Zhaoying Liu, Xueling Song, Yngkun Zhang, “Research on Improved Artificial Potential Field
2. Approach in Local Path Planning for Mobile Robot. Computer Simulation”, 2018
3. Liu Cuodong, Xie Hong-bin, Li Chunguang, “Method of mobile robot path planning in dynamic environment based on genetic algorirthm [J]. Robot”, 2003
4. Bingbing Xu, Rongfei Hao, “Current Situation and Development of Robot Path Planning Technology. Electronic Technology & Software Engineering”, 2019
5. Yifan Guo and Zhiping Liu, “UAV Path Planning Based on Deep Reinforcement Learning”, 2023
6. Jeevan Raajan, Srihari P V, Satya Jayadev P, B Bhikkaji and Ramkrishna Pasumarthy, “Real Time Path Planning of Robot using Deep Reinforcement Learning”, 2020
7. Sudharsan Ravichandiran, “Deep Reinforcement Learning with Python”, 2020
8. Phalgun Chintala, Rolf Dornberger and Thomas Hanne, “Robotic Path Planning by Q Learning and a performance Comparison with Classical Path Finding Algorithms”, 2022
9. Li S., Xin X., and Lei Z., “Dynamic path planning of a mobile robot with improved Q-learning algorithm”, 2015
10. Zhen Shi, Keyin Wang, Jianhui Zhang “Improved reinforcement learning path planning algorithm integrating prior knowledge”, 2023
11. Li S., Xin X., and Lei Z., “Dynamic path planning of a mobile robot with improved Q-learning algorithm”, 2015
12. Noor H. Fallooh, Ahmed T. Sadiq, Eyad I. Abbas & Ivan A. hashim, “Dynamic Path Planning using a modification Q-Learning Algorithm for a Mobile Robot”,2024
13. Xuemei He, Yin Kuang, Ning Song, Fan Liu, “Intelligent Navigation of Indoor Robot Based on Improved DDPG Algorithm”,2023

© STM Journals 2024. All Rights Reserved