DOI:10.19651/j.cnki.emt.2416813

基于自适应探索 DDQN 的移动机器人路径规划*

冷忠涛[1] 张烈平[2] 彭建盛[3] 王艺霖[4] 张 翠[5]
(1.桂林理工大学广西高校先进制造与自动化技术重点实验室 桂林 541006;2.桂林航天工业学院广西特种工程装备与控制重点实验室 桂林 541004;3.河池学院广西高校人工智能与信息处理重点实验室 河池 546300;4.桂林明富机器人科技有限公司 桂林 541004;5.南宁理工学院信息工程学院 桂林 541006)

摘 要:针对传统双深度 Q 网络算法在路径规划中探索和利用分配不平衡,数据利用不充分等问题,提出了一种改进的 DDQN 路径规划算法。首先,在自适应探索策略中引入探索成功率的概念,将训练过程分为探索环境和利用环境两个阶段,合理分配探索和利用。其次,通过双经验池混合采样机制,将经验数据按照奖励大小进行分区采样,确保有利数据的利用度达到最大。最后,设计了基于人工势场的奖励函数,使机器人能获得更多的单步奖励,有效改善了奖励稀疏的问题。实验结果表明,所提出的算法相比传统 DDQN 算法和基于经验分区和多步引导的 DDQN 算法能获得更高的奖励值,成功率更高,规划时间和步数也更短,算法整体性能更加优越。

关键词:路径规划;DDQN;自适应探索;双经验池;人工势场
中图分类号:TP242;TN711 **文献标识码**:A **国家标准学科分类代码**:520.60

Path planning for mobile robots based on self-adaptive exploration DDQN

Leng Zhongtao[1] Zhang Lieping[2] Peng Jiansheng[3] Wang Yilin[4] Zhang Cui[5]
(1. Key Laboratory of Advanced Manufacturing and Automation Technology, Guilin University of Technology, Education Department of Guangxi Zhuang Autonomous Region, Guilin 541006, China; 2. Guangxi Key Laboratory of Special Engineering Equipment and Control, Guilin University of Aerospace Technology, Guilin 541004, China; 3. Key Laboratory of AI and Information Processing, Education Department of Guangxi Zhuang Autonomous Region, Hechi University, Hechi 546300, China; 4. Guilin Mingfu Robot Technology Company Limited, Guilin 541004, China; 5. School of Information Engineering, Nanning College of Technology, Guilin 541006, China)

Abstract: To address issues such as the imbalanced allocation of exploration and exploitation, as well as insufficient data utilization in traditional double deep Q-Network algorithms for path planning, an improved DDQN path planning algorithm is proposed. Firstly, the concept of exploration success rate is introduced into the adaptive exploration strategy, dividing the training process into exploration and exploitation phases to allocate exploration and exploitation effectively. Secondly, the double experience pool mixed sampling mechanism partitions and samples experience data based on reward size to maximize the utilization of beneficial data. Finally, a reward function based on artificial potential field is designed to enable the robot to receive more single-step rewards, effectively addressing the issue of sparse rewards. Experimental results show that the proposed algorithm achieves higher reward values, greater success rates, and shorter planning times and steps compared to the traditional DDQN algorithm and the DDQN algorithm based on experience classification and multi-steps, demonstrating superior overall performance.

Keywords: path planning; DDQN; self-adaptive exploration; double experience pool; artificial potential field

0 引 言

在人工智能高速发展的背景下,机器人技术在工业自动化,军事,农业,医疗等领域中得以广泛应用。在安全监控,巡逻报警,家庭服务等复杂任务中,移动机器人扮演着非常重要的角色。路径规划是移动机器人领域中的重要组成部分,其目的是让机器人获得一条能够合理避开障碍物,耗时短且可行性大的行进路线[1]。传统的路径规划算法有

收稿日期:2024-09-05
*基金项目:国家自然科学基金(62063006)、广西科技重大专项(2022AA05002)、广西高校人工智能与信息处理重点实验室项目(2022GXZDSY003)资助

人工势场算法[2],DWA 算法[3],A* 算法[4],Dijkstra 算法[5],粒子群算法[6],人工蜂群算法[7]等。虽然这些算法在移动机器人路径规划领域中各自都有着良好的表现,但是在路径规划时非常依赖全局地图的可行性,有很大的局限性。

近年来,针对未知环境下复杂的路径规划问题,越来越多的研究人员将强化学习算法应用到移动机器人的路径规划中。普遍的强化学习算法像 Q-learning 算法[8],SARSA 算法[9]等虽然在路径规划领域中有良好的表现,但这些算法难以处理状态维度和动作维度过大的问题。谷歌旗下的 DeepMind 公司在 2013 年提出了 DQN(deep q-network, DQN)算法[10],通过深度神经网络拟合 Q 值,解决了状态维度和动作维度过大的问题[11],但是 DQN 对 Q 值的估计中容易出现过高的偏差问题,且在训练过程中不稳定。为解决 Q 值过高估计问题,Hasselt 等[12]提出了双深度 Q 网络(double deep q-network, DDQN)算法,将目标中的最大操作分解为动作选择和动作评估来达到减少 Q 值高估的目的,但是 DDQN 算法没有解决有利样本利用率低的问题。

Wang 等[13]为解决算法收敛过慢的问题,结合 DDQN 和优先经验重放(prioritized experience replay, PER)来提高学习效率和稳定性,但是所提出的算法在处理更复杂的任务时有很大的局限性。Chu 等[14]提出一种基于改进 DDQN 的深度强化学习路径规划方法,使用两个输入层的改进卷积神经网络结构来构建 DDQN,但是该方法需要大量的数据和计算资源来进行训练。针对 DDQN 算法进行路径规划时收敛速度慢和精度低的问题,Jiang 等[15]利用二阶时间差分方法评估当前迭代结果的有效性,并引入二叉树结构代替传统的经验池结构来存储迭代动作,提出了基于改进的双深度 Q 网络(improved double deep q network, IDDQN)算法,提高了训练效率和算法精度,但算法没有考虑到在不同类型障碍物环境中的泛化能力。

综上所述,目前基于 DDQN 改进的路径规划算法虽然在一定程度上提高的算法的总体性能,但仍然存在以下 3 个问题:1)传统 DDQN 算法使用的探索策略仍然是 ε-greedy 策略,对于探索和利用两个阶段分配不平衡。2)有效数据利用不足,在已有环境经验的利用上仍然存在改进的空间。3)奖励稀疏问题,在设计奖励函数上还需要更细致的奖励。为此,本文提出了一种基于双经验池(double experience pool, DEP)和人工势场(artificial potential field,APF)的自适应探索 DDQN(self-adaptive exploration double deep q-network, SAE-DDQN)算法。首先,针对 DDQN 算法在训练时对环境信息的探索和利用不平衡问题,提出了自适应探索策略,通过引入探索成功率的概念区分探索率衰减过程,并设置自适应探索因子,使探索和利用能有效分配。其次,设置 DEP 混合采样机制对经验数据进行采样,将经验数据按照奖励大小进行排序,优先采样经验高的数据。最后基于 APF 的思想设计奖励函数,使机器人每一步都能获得相应的奖励,有效解决了最终奖励稀疏的问题。

1 深度强化学习算法

1.1 DQN 算法

Q-learning 在面对环境的复杂程度增加时,Q 表会急剧增加,导致容量维度紧张,为了解决这一问题,DQN 算法利用深度神经网络作为函数逼近器来近似值函数 $Q(s,a)$ 评估在给定状态下执行具体动作的价值,如式(1)所示。

$$Q(s,a,\theta)\approx Q^*(s,a) \tag{1}$$

式中:$Q^*(s,a)$ 是最优 Q 值,表示在状态 s 下执行动作 a 后所能得到的最大回报期望,θ 是最初网络参数,$Q(s,a;\theta)$ 是一个评估 Q 网络,表示在状态 s 下执行动作 a 后所能得到的预期未来回报的估计值。

DQN 算法增加了目标 Q 网络 $\hat{Q}(s_{t+1},a_{t+1};\theta')$ 来产生目标 Q 值,其中 θ' 是更新后的网络参数。评估 Q 网络和目标 Q 网络在网络结构是一样的,但是评估 Q 网络中的网络参数 θ 是实时更新的,而目标 Q 网络中的网络参数 θ' 是通过多次迭代后复制评估 Q 网络中的网络参数 θ 来更新的。DQN 的目标 Q 值计算方式如式(2)所示。

$$Y^{DQN}=r+\gamma\max_{a_{t+1}}\hat{Q}(s_{t+1},a_{t+1};\theta') \tag{2}$$

式中:r 是 t 时刻完成动作 a_t 后的奖励,γ 是奖励折扣因子,s_{t+1} 是 $t+1$ 时刻的状态,a_{t+1} 在状态 s_{t+1} 下选择的最优动作,θ' 是目标 Q 网络的参数。

1.2 DDQN 算法

DDQN 算法是在 DQN 的基础上进行改进的,两者之间的区别在于 DQN 算法是在目标 Q 网络选取最大 Q 值对应的动作,并用目标 Q 网络评价该动作的优劣,而 DDQN 算法是在评估 Q 网络中选取最大 Q 值对应的动作,用目标 Q 网络评价该动作的优劣。DQN 在进行动作选择和动作评估时都是依赖于目标网络的参数,会导致 Q 值的过度估计,而 DDQN 算法将动作选择和动作评估分开进行,有效改善了这个弊端。

DDQN 算法在目标 Q 网络 $\hat{Q}(s_{t+1},a_{t+1};\theta')$ 中对选择的动作进行评估,其更新函数如式(3)所示。DDQN 算法整体框架如图 1 所示。

$$Y^{DDQN}=r+\gamma\hat{Q}(s_{t+1},\operatorname{argmax}_{a_{t+1}}Q(s_{t+1},a_{t+1};\theta);\theta') \tag{3}$$

2 DDQN 算法改进与实现

2.1 自适应探索策略设计

DDQN 算法利用 ε-greedy 策略进行环境探索,通过给贪婪因子 ε 赋予一个小于 1 的初值,以 1−ε 的概率获取当前状态下的最优动作。ε-greedy 策略表达式如式(4)所示。

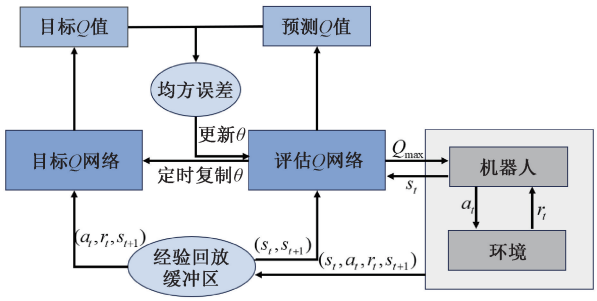


图 1　DDQN 整体框架图

Fig. 1　Overall framework diagram of DDQN

$$\pi(a\mid s)=\begin{cases}1-\dfrac{\varepsilon}{\mid A\mid}(\mid A\mid-1), & a^{*}=\underset{a\in A}{\operatorname{argmax}}\,Q(s,a)\\ \dfrac{\varepsilon}{\mid A\mid}, & \text{其他}\end{cases}\tag{4}$$

式中：$\mid A\mid$ 代表动作集合 A 所有动作的个数，a^{*} 表示当前状态下的最优动作。ε-greedy 策略虽然能在一定程度上合理分配探索和利用，但是由于贪婪因子是一个固定值，在机器人与环境交互后期不利于机器人对已掌握环境信息的利用。本文在 ε-greedy 策略的基础上提出了自适应探索策略，达到合理分配探索和利用的目的。

根据移动机器人与环境交互的程度，自适应探索策略将训练过程分为探索和利用两个阶段，在探索阶段，移动机器人的主要目的是广泛探索未知环境，获得大量的经验数据，因此希望贪婪因子 ε 的衰减速度较慢。具体调整如式(5)所示。

$$\varepsilon_t=1-\sin\left(\left(\frac{t}{T}\right)^{2}\times\frac{\pi}{2}\right),\ t\leqslant\beta T\tag{5}$$

式中：t 为当前迭代的周期数，ε_t 为当前时刻的贪婪因子，T 为整个训练过程的最大训练周期数，β 为自适应探索因子，范围为[0，0.5]。

在算法训练过程中通过比较当前迭代次数和预设迭代次数的大小来判断是否进入利用阶段。在利用阶段移动机器人已掌握了足够的环境信息，此时主要目的是利用已有的环境信息加速算法训练过程。本文提出的自适应探索策略引入了探索成功率这一参数，即在多少次实验之后，智能体达到目标区域的概率。利用阶段贪婪因子 ε 的调整如式(6)所示。

$$\varepsilon_t=\begin{cases}\varepsilon_1+\dfrac{t-\beta T}{T_1-\beta T}(\varepsilon_{\min}-\varepsilon_1)-i\times R, & \varepsilon_t\geqslant\varepsilon_{\min}\\ \varepsilon_{\min}, & \text{其他}\end{cases}\tag{6}$$

式中：T_1 是最小 ε 的目标幕数，ε_1 是探索阶段结束后 ε 的最终数值，i 是探索成功率的加速利用系数，R 为智能体每十次实验中能成功抵达终点的概率，$\varepsilon_{\min}$ 是最小的贪婪因子数值。ε 值的衰减速度和探索成功率之间存在正相关关系，探索成功率越高则表明信息的利用效率更高，不影响算法性能表现的基础上，最大限度的增加收敛速度。

2.2　双经验池混合采样机制设计

在 DDQN 算法中使用了经验回放的方法来降低机器人在训练过程中经验数据的相关性。将机器人与环境交互的数据 (s_t,r_t,a_t,s_{t+1}) 存放在一个容器中，通过随机采样的方式在容器中采样，实现对过去样本数据的复用，但随机采样的方式会导致同时对同一个数据进行多次采样，且数据之间没有优先级之分，也会导致有利数据利用不够充分。

为解决此问题，本文提出了一个 DEP 混合采样机制。将原有容器中的经验数据按照奖励值的大小进行排序，删除奖励值最低的数据，然后设置两个经验池，将奖励值高的数据按比例放进经验池Ⅰ，其余数据放进经验池Ⅱ中。通过设置采样权重 ω，优先采样经验池Ⅰ中的数据，然后使用随机均匀采样对经验池Ⅱ中的数据采样，避免了同时对同一个数据重复采样。改进的混合采样机制结合优先采样和随机均匀采样，优先采取奖励值高的经验数据，淘汰奖励值低的数据，提高了有利数据的利用率，有利于加速算法训练。采样过程如图 2 所示。

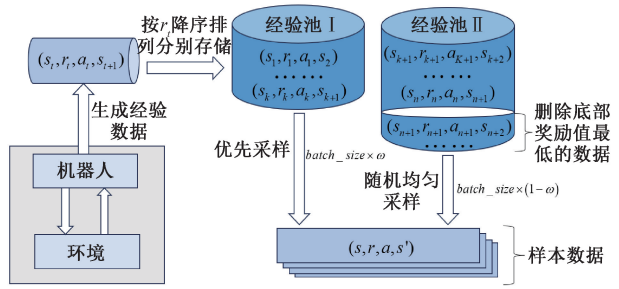


图 2　双经验池混合采样过程

Fig. 2　The process of double experience pool hybrid sampling

2.3　基于 APF 的奖励函数设计

奖励主要影响移动机器人的运动状态，可以让机器人更快学习到环境模型。为提高机器人与环境的交互效率，使其更好的利用奖励来调整行动策略，需要采取更加细化的奖励反馈，为此本文设计了一个基于 APF 的奖励函数。APF 的核心思想是在环境中虚拟一个人工势场，通过虚拟人工势场作用力的思想影响移动机器人每一步的奖励。虚拟人工势场包括引力势场和斥力势场，引力势场一般布置在目标点附近，吸引机器人向目标点移动，引力势场函数如式(7)所示。

$$U_{att}(x)=\frac{1}{2}k_{att}(x-x_{goal})^{2}\tag{7}$$

式中：x 是机器人当前的位置，x_{goal} 是目标点的位置。k_{att} 是吸引力的系数。

斥力势场布置在障碍物附近，使机器人避开障碍物，斥力势场函数如式(8)所示。

$$U_{rep}(x)=\begin{cases}\dfrac{1}{2}k_{rep}\left(\dfrac{1}{d_{obstacle}}-\dfrac{1}{d_0}\right)^{2}, & d_{obstacle}<d_0\\ 0, & \text{其他}\end{cases}\tag{8}$$

式中:$d_{obstacle}$ 是机器人当前位置与障碍物的位置的距离,d_0 是排斥势场的影响距离,k_{rep} 是排斥力的系数。

本文根据机器人和目标点的位置设置了吸引奖励函数,即越接近目标,得到的奖励越多;吸引奖励函数式(9)所示。

$$r_1 = \frac{1}{2}\rho_1 2^{\frac{1}{dis\_current}} \tag{9}$$

式中:ρ_1 是吸引奖励函数常数,dis_current 表示机器人当前状态到目标点的距离。

根据机器人位置和障碍物位置设置排斥奖励函数,越靠近障碍物,惩罚越大,公式如式(10)所示。

$$r_2 = -\frac{1}{2}\rho_2 \frac{1}{dis\_obstacle} \tag{10}$$

式中:ρ_2 是排斥奖励函数常数,dis_obstacle 表示机器人到障碍物的距离。

根据机器人朝向和目标点的角度差设置方向奖励函数,当角度差在 $\left[-\frac{\pi}{2},\frac{\pi}{2}\right]$ 范围内,机器人得到一个正奖励,否则获得一个负奖励,如式(11)所示。

$$r_3 = \begin{cases} \pi - |\alpha|, & -\frac{\pi}{2} \leqslant \alpha \leqslant \frac{\pi}{2} \\ -|\alpha|, & \text{其他} \end{cases} \tag{11}$$

式中:α 表示机器人当前的朝向和目标点的角度差。除上述奖励函数外,本文还设计了 3 个基础奖励:

1)当机器人与目标点距离小于 0.25 m 时,代表机器人到达目标点,此时赋予机器人一个正奖励 $r_4 = 100$。

2)当机器人与障碍物距离大于 0,小于 0.125 m 时,表示机器人与障碍物发生碰撞,此时给机器人一个大的惩罚 $r_5 = -100$。

3)当机器人上一状态到目标点的距离和当前状态到目标点的距离的差值大于 0 时,获得一个正奖励,否则获得一个负奖励,如式(12)所示。

$$r_6 = \begin{cases} 1, & \text{dis\_past} - dis\_current > 0 \\ -1, & \text{其他} \end{cases} \tag{12}$$

式中:dis_past 表示机器人在上一状态时到目标点的距离。

2.4　算法实现

基于上述改进,首先在 DDQN 算法的基础上引入自适应探索策略,平衡探索和利用,然后采用 DEP 混合采样机制对经验数据采样,优化采样机制,最后设计基于 APF 奖励函数解决奖励稀疏问题,提出了基于 SAE-DDQN 的路径规划算法,具体伪代码步骤算法 1 所示。

算法 1:SAE-DDQN 算法

初始化:动作空间 A,状态空间 S,学习率 α,训练回合数 E,衰减因子 γ,每回合步数 T,样本数量 $batch\_size$,经验池Ⅰ,经验池Ⅱ,随机初始化评估 Q 网络参数 θ,将目标 Q 网络的参数初始化为 $\theta' = \theta$,更新参数间隔步数 C。

for $episode = 1$, E **do**

初始化:初始状态

for $t = 1$, T **do**

a) 根据自适应探索策略,利用式(5)和(6)计算贪婪因子 ε,以概率 ε 从动作空间选择随机动作 a_t,否则评估 Q 网络根据 $a_t = \text{argmax}_{a_{t+1}} Q(s_t, a_t; \theta)$ 选择动作

b) 移动机器人执行动作 a_t,并根据 APF 奖励函数计算即时奖励,从环境中获取新的环境状态 s_{t+1} 和即时奖励 r_t

c) 将经验数据 (s_t, a_t, r_t, s_{t+1}) 降序排列,按比例分别存放到经验池Ⅰ和经验池Ⅱ中

d) $s_t \leftarrow s_{t+1}$

e) 从经验池Ⅰ中优先采样 $batch\_size \times \omega$ 个样本数据,经验池Ⅱ中随机均匀采样 $batch\_size \times (1-\omega)$ 个样本数据,结合两个样本数据计算当前目标 Q 值 y:

$$y = r + \gamma \hat{Q}(s_{t+1}, \text{argmax}_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta); \theta')$$

f) 根据均方差损失函数更新 Q 网络的参数 θ

g) 每 C 步更新目标 Q 网络参数 θ',$\theta' \leftarrow \theta$

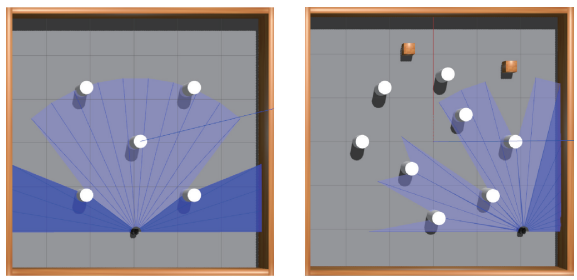
end for

end for

3　基于 ROS 平台的实验设计及结果分析

3.1　仿真实验

为验证本文所提出 SAE-DDQN 路径规划算法的可行性和有效性,本文在 Gazebo 仿真环境中搭建了一个面积为 6 m×6 m 的简单障碍物环境和复杂障碍物环境,如图 3 所示。在简单障碍物环境中,设置移动机器人与障碍物碰撞后回合结束,而到达目标位置后还会继续在这个回合寻找下一个目标位置,如果一直没有碰撞,只有到回合步数最大才结束回合,这样设计有利于移动机器人更快速的学习到好的动作。在复杂障碍物环境中,设置移动机器人与障碍物碰撞或者到达目标点后回合结束。



(a) 简单障碍物环境
(a) Simple obstacle environment

(b) 复杂障碍物环境
(b) Complex obstacle environment

图 3　仿真实验环境

Fig. 3　Simulation experiment environment

本文所用的操作系统为 Linux,CPU 为 i7-13620H,内存为 16 GB,Ubuntu 版本为 20.04,ROS 版本是 Noetic,仿

真平台为 Gazebo。在不同的实验环境中本文设置的部分算法参数参考文献[16]，数值如表 1 所示。

表 1　Gazebo 环境中算法参数设置

Table 1　Algorithm parameter settings in the Gazebo environment

参数	数值
初始探索因子 ε_{init}	1.0
最小探索因子 ε_{min}	0.01
学习率 α	0.000 1
奖励衰减因子 γ	0.9
每回合步数 T	根据环境设置
训练回合数	根据环境设置
成功率利用系数 i	0.05
自适应探索因子 β	0.05
样本 $batch\_size$	256
采样权重 ω	0.7
目标网络更新频率	10

3.2　仿真实验结果分析

为验证本文提出的 DEP 混合采样机制的有效性，在复杂障碍物环境中对传统 DDQN 算法和加入 DEP 混合采样机制的 DDQN 算法进行比较，训练过程中所得出的平均奖励值曲线和成功率曲线如图 4 和 5 所示。

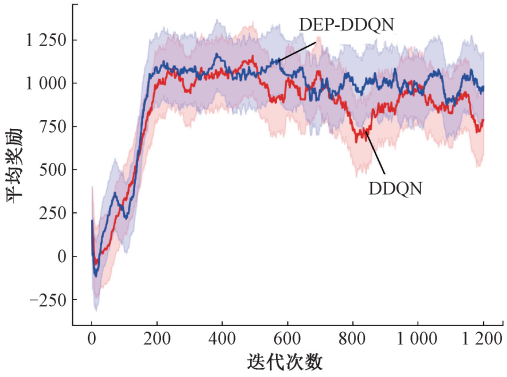


图 4　复杂障碍物环境下 DDQN 算法和 DEP-DDQN 算法的奖励曲线

Fig. 4　Reward curves of the DDQN algorithm and the DEP-DDQN algorithm in a complex obstacle environment

从图 4 和 5 中可以明显看出，DEP-DDQN 算法的平均奖励曲线收敛更快，且波动较小，说明 DEP-DDQN 算法能够采样到更多有利的经验数据，最终的成功率能达到 64.4%，比 DDQN 算法更高。实验验证了 DEP 混合采样机制的有效性。

为验证本文提出的 SAE-DDQN 算法在移动机器人路径规划上的性能，将传统 DDQN 算法，ECMS-DDQN 算法[17]和本文提出的 SAE-DDQN 算法分别利用上述参数搭建的简单障碍物环境和复杂障碍物环境进行训练。

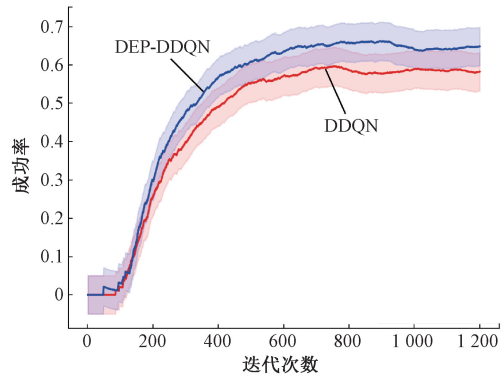


图 5　复杂障碍物环境下 DDQN 算法和 DEP-DDQN 算法的成功率曲线

Fig. 5　Success rate curves of the DDQN algorithm and the DEP-DDQN algorithm in a complex obstacle environment

在简单障碍物环境中，DDQN 算法，ECMS-DDQN 算法和本文提出的算法在训练过程中的平均奖励值曲线，成功率曲线和平均步数曲线分别如图 6～8 所示。

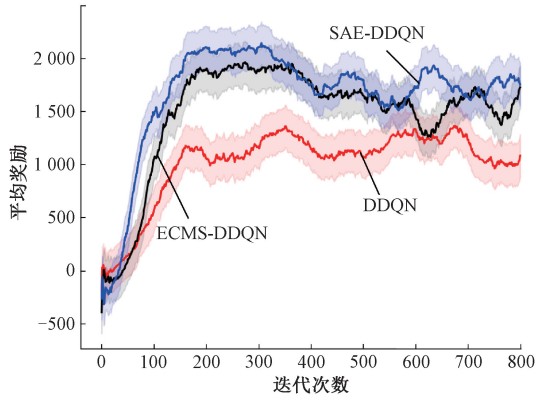


图 6　简单障碍物环境下 3 种算法的奖励曲线

Fig. 6　Reward curves of three algorithms in a simple obstacle environment

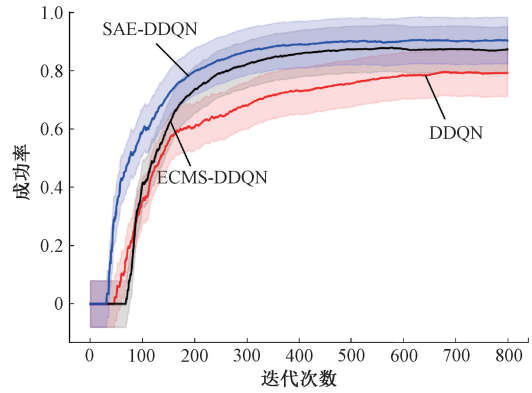


图 7　简单障碍物环境下 3 种算法的成功率曲线

Fig. 7　Success rate curves of three algorithms in a simple obstacle environment

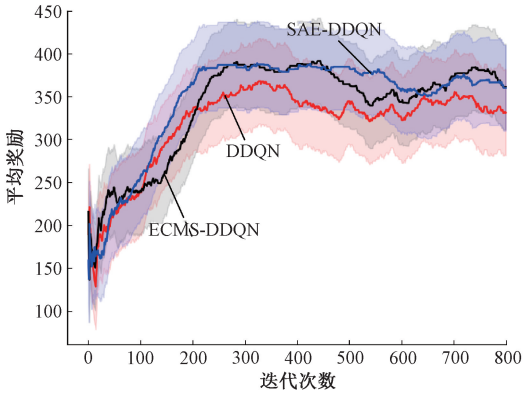


图 8 简单障碍物环境下 3 种算法平均步数曲线

Fig. 8 Average step count curves of three algorithms in a simple obstacle environment

从图 6～8 上可以看出,在简单障碍物环境中,SAE-DDQN 算法和 ECMS-DDQN 算法,DDQN 算法相比能获得更高的奖励值,在前 200 个回合内获得高奖励值的速度更快。从成功率的角度来看,SAE-DDQN 算法成功率能达到 90.4%,ECMS-DDQN 算法能达到了 86.7%,DDQN 算法只有 79.3%,SAE-DDQN 算法比 ECMS-DDQN 算法高 3.7%,比 DDQN 算法高 11.1%。

为避免奖励过大,在简单障碍物环境中的总步数设置为 400,前期 3 种算法在探索环境时由于和障碍物发生碰撞,步数都比较低,但随着迭代次数的增加,SAE-DDQN 算法收敛趋势明显比 DDQN 算法和 ECMS-DDQN 算法更快,SAE-DDQN 算法和 ECMS-DDQN 算法的平均步数都能收敛到 400 左右,但是 ECMS-DDQN 算法波动较大,而 DDQN 算法由于碰撞较多,只能收敛到 328 左右。

表 2 简单障碍物环境中 3 种不同算法性能对比(20 次平均值)

Table 2 Performance comparison of three different algorithms in a simple obstacle environment(average of 20 times)

算法	DDQN	ECMS-DDQN	SAE-DDQN
规划时间/s	33.40	29.55	28.00
路径长度/m	6.25	5.49	5.17
步数	310.15	275.10	261.90

表 2 统计了 3 种算法在简单障碍物环境中性能指标的平均值。将 3 种不同算法的 3 个性能指标分别进行方差分析,计算并统计对应的 F 值和 p 值,如表 3 所示。

设置显著差异水平为 0.05,p 值用于判断结果的显著性,如果 p 值小于 0.05,则说明至少有一组算法存在显著差异。从表 3 中可以看出 3 个性能指标的 p 值均小于 0.05,说明这 3 种算法在对应的性能指标上都存在显著差异。对 3 个性能指标的显著差异进行事后检验,统计对应的均值差异和调整后的 p 值,如表 4 和表 5 所示。

表 3 简单障碍物环境中 3 种不同算法各性能指标的 F 值和 p 值

Table 3 The F-values and p-values of the performance metrics for three different algorithms in a simple obstacle environment

参数	F 值	p 值
规划时间	33.67	2.21×10^{-10}
路径长度	36.53	6.16×10^{-11}
步数	32.14	4.50×10^{-10}

表 4 简单障碍物环境中 DDQN 算法和 SAE-DDQN 算法各性能指标的显著差异对比

Table 4 Comparison of significant differences in performance metrics between the DDQN algorithm and the SAE-DDQN algorithm in a simple obstacle environment

参数	均值差异	调整后的 p 值
规划时间	−5.4 s	0.0
路径长度	−1.08 m	0.0
步数	−48.25	0.0

表 5 简单障碍物环境中 ECMS-DDQN 算法和 SAE-DDQN 算法各性能指标的显著差异对比

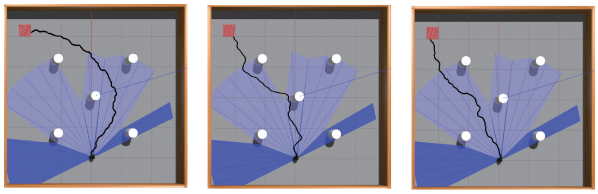
Table 5 Comparison of significant differences in performance metrics between the ECMS-DDQN algorithm and the SAE-DDQN algorithm in a simple obstacle environment

参数	均值差异	调整后的 p 值
规划时间	−1.55 s	0.06
路径长度	−0.32 m	0.03
步数	−13.2	0.09

从表 4 和 5 中可以明显看出,在均值差异上,SAE-DDQN 算法进行路径规划时从起点运动到终点所需的时间比 DDQN 缩短了 5.4 s,比 ECMS-DDQN 缩短了 1.55 s。SAE-DDQN 算法所规划的平均路径长度比 DDQN 短了 1.08 m,比 ECMS-DDQN 缩短了 0.32 m。同时,SAE-DDQN 算法的平均步数也比 DDQN 少了 48.25,比 ECMS-DDQN 少了 13.2。通过调整后的 p 值可以看出,SAE-DDQN 算法与 DDQN 算法相比,各性能指标调整后的 p 值均小于 0.05,说明在 3 个性能指标上 SAE-DDQN 和 DDQN 算法均存在显著差异,性能提升较高,而与 ECMS-DDQN 算法比较,只有路径长度上有显著差异,规划时间和步数上的差异不显著。

简单障碍物环境下 DDQN、ECMS-DDQN 和 SAE-DDQN 算法在移动机器人路径规划测试实验中单次训练给出的规划路径如图 9 所示。3 种算法在规划路径时都选择直接穿过障碍物朝向目标点移动,但 SAE-DDQN 算法规划的路径距离更短且路径较为平滑,DDQN 算法在规划

绕远严重，而 ECMS-DDQN 算法在规划路径时距离障碍物比较近，更容易发生碰撞。



(a) DDQN算法 (a) DDQN algorithm　(b) ECMS-DDQN算法 (b) ECMS-DDQN algorithm　(c) SAE-DDQN算法 (c) SAE-DDQN algorithm

图 9　简单障碍物环境下 3 种算法路径规划轨迹

Fig. 9　Path planning trajectories of three algorithms in a simple obstacle environment

在复杂障碍物环境中，DDQN 算法，ECMS-DDQN 算法和本文提出的算法在训练过程中的平均奖励值曲线、成功率曲线和平均步数曲线如图 10～12 所示。

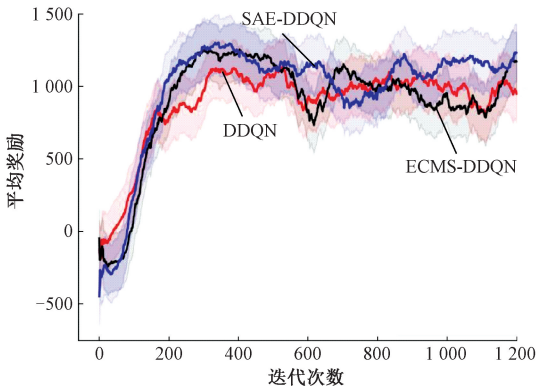


图 10　复杂障碍物环境下 3 种算法的奖励曲线

Fig. 10　Reward curves of three algorithms in a complex obstacle environment

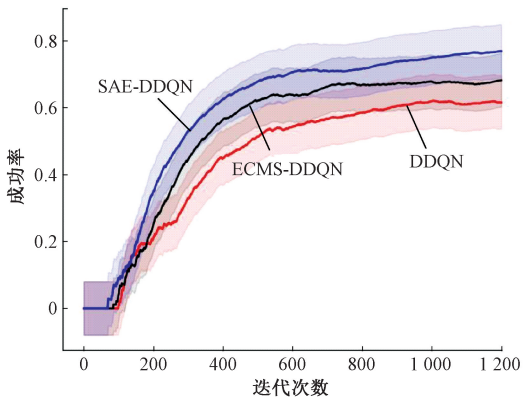


图 11　复杂障碍物环境下 3 种算法的成功率曲线

Fig. 11　Success rate curves of three algorithms in a complex obstacle environment

从图 10～12 中可以看出，3 种算法的奖励值随着迭代次数的增加而变化。SAE-DDQN 算法大约在 255 回合时有明显的收敛趋势，ECMS-DDQN 算法在大约 308 个回合时才有收敛趋势，DDQN 算法大约在 320 回合才开始收敛。且 SAE-DDQN 算法获得的平均奖励值也比 ECMS-DDQN 算法和 DDQN 算法更高，说明 SAE-DDQN 算法对样本利用率高，能更快找到有效样本，获得奖励也更加细致，在路径规划上具有更优越的表现。

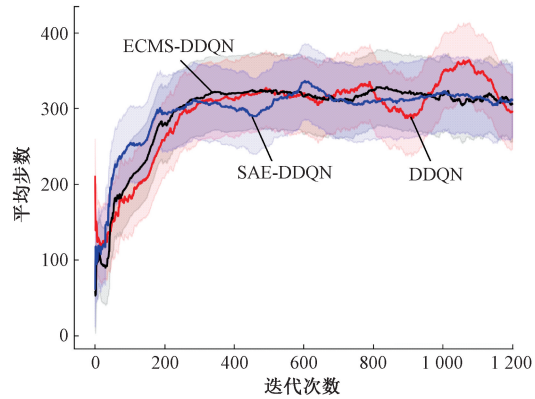


图 12　复杂障碍物环境 3 种算法的平均步数曲线

Fig. 12　Average step count curves of three algorithms in a complex obstacle environment

在复杂障碍物环境中，SAE-DDQN 算法成功率能达到 76.8%，ECMS-DDQN 算法达到了 68.2%，DDQN 算法只有 61.3%，SAE-DDQN 算法比 ECMS-DDQN 算法高 8.6%，比 DDQN 算法高 15.5%，SAE-DDQN 算法的成功率在 69 个回合时成功率开始上升，前期上升速度也比 DDQN 算法和 ECMS-DDQN 算法更快，说明 SAE-DDQN 算法对有利样本的利用率高，探索环境和利用环境的过程更加均衡，能率先找到目标点。前期 3 种算法在探索环境时由于和障碍物发生碰撞，步数都比较低，但随着迭代次数的增加，SAE-DDQN 算法的收敛趋势明显比 DDQN 算法和 ECMS-DDQN 算法更快。

表 6　复杂障碍物环境中 3 种不同算法性能对比（20 次平均值）

Table 6　Performance comparison of three different algorithms in a complex obstacle environment (average of 20 times)

算法	规划时间/s	路径长度/m	步数
DDQN	41.30	38.40	35.75
ECMS-DDQN	7.73	7.24	6.70
SAE-DDQN	380.20	353.55	330.75

表 6 中统计了 3 种算法在复杂障碍物环境中性能指标的平均值，将 3 个性能指标分别进行方差分析，计算并统计对应的 F 值和 p 值，如表 7 所示。

设置显著差异水平为 0.05，从表 7 中可以看出 3 个性能指标的 p 值均小于 0.05，说明这 3 种算法在对应的性能指标上都存在显著差异。对 3 个性能指标的显著差异进行

表 7　复杂障碍物环境中 3 种不同算法各性能指标的 F 值和 p 值

Table 7　The F-values and p-values of the performance metrics for three different algorithms in a complex obstacle environment

参数	F 值	p 值
规划时间	36.49	6.27×10^{-11}
路径长度	30.33	1.07×10^{-9}
步数	38.01	3.25×10^{-11}

事后检验,统计对应的均值差异和调整后的 p 值,如表 8 和 9 所示。

表 8　复杂障碍物环境中 DDQN 算法和 SAE-DDQN 算法各性能指标的显著差异对比

Table 8　Comparison of significant differences in performance metrics between the DDQN algorithm and the SAE-DDQN algorithm in a complex obstacle environment

参数	均值差异	调整后的 p 值
规划时间	−5.55 s	0.0
路径长度	−1.03 m	0.0
步数	−49.45	0.0

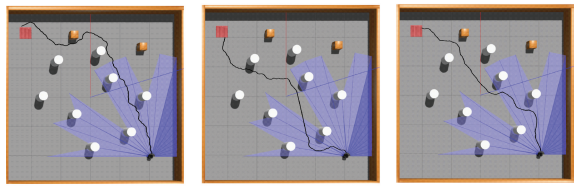
表 9　复杂障碍物环境中 ECMS-DDQN 算法和 SAE-DDQN 算法各性能指标的显著差异对比

Table 9　Comparison of significant differences in performance metrics between the ECMS-DDQN algorithm and the SAE-DDQN algorithm in a complex obstacle environment

参数	均值差异	调整后的 p 值
规划时间	−2.65 s	0.000 4
路径长度	−0.54 m	0.000 4
步数	−22.8	0.000 5

从表 8 和 9 中可以明显看出,在均值差异上,本文提出的算法进行路径规划时从起点运动到终点所需的时间比 DDQN 缩短了 5.55 s,比 ECMS-DDQN 缩短了 2.65 s。SAE-DDQN 算法的平均路径长度比 DDQN 短了 1.03 m,比 ECMS-DDQN 短了 0.54 m。同时,SAE-DDQN 算法的平均步数也比 DDQN 少了 49.45,比 ECMS-DDQN 少了 22.8。通过调整后的 p 值可以看出,SAE-DDQN 算法分别与 DDQN 算法,ECMS-DDQN 算法相比,各性能指标调整后的 p 值均小于 0.05,说明在 3 个性能指标上 SAE-DDQN 和 DDQN 算法存在显著差异,和 ECMS-DDQN 算法也存在显著差异,性能提升较高。从两次仿真实验的数据来看,本文提出的 SAE-DDQN 算法在整体性能上明显优于 DDQN 算法和 ECMS-DDQN 算法。

在复杂障碍物环境下,DDQN、ECMS-DDQN 和 SAE-DDQN 算法在移动机器人路径规划测试实验中单次训练给出的规划路径如图 13 所示。可以明显看出,SAE-DDQN 算法规划的路径距离更短且路径更加平滑,DDQN 算法和 ECMS-DDQN 算法在规划路径时距离障碍物比较近,更容易发生碰撞。



(a) DDQN算法 (a) DDQN algorithm　(b) ECMS-DDQN算法 (b) ECMS-DDQN algorithm　(c) SAE-DDQN算法 (c) SAE-DDQN algorithm

图 13　复杂障碍物环境下 3 种算法路径规划轨迹

Fig. 13　Path planning trajectories of three algorithms in a complex obstacle environment

3.3　实际环境实验验证

为验证本文提出的 SAE-DDQN 算法在实际环境中的可行性和适用性,本文利用 ROS 分布式控制系统,采用了基于麦克纳姆轮的全向移动机器人,以雷达数据作为状态输入,如图 14 所示。电脑端使用的操作系统为 Ubuntu20.04,CPU 为 Intel(R) Core(TM) i7-13620H,内存为 16GB,ROS 版本为 noetic。移动机器人实体尺寸为 0.25 m×0.22 m×0.2 m,激光雷达最大测量距离为 12 m。由于空旷场地和道具有限,并考虑到机器人电池工作时间和无线通信信号的强弱等因数,本次实验在室内搭建了一个面积为 4 m×6 m 的实际实验环境场景,实验环境中利用 5 个圆柱桶来充当障碍物,将移动机器人的起点和目标点分别设置在五个障碍物的外围,设置距离目标点 0.25 m 范围内表示到达终点,如果机器人能顺利避开所有障碍物到达终点范围内则表示成功,否则程序停止运行。

(a) 实验场地 (a) Experimental site　(b) 基于麦克纳姆轮的移动机器人 (b) Mobile robot based on Mecanum wheels

图 14　实际实验环境

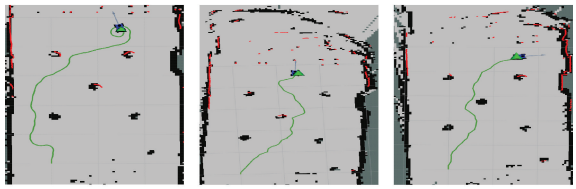
Fig. 14　Actual experimental environment

将仿真环境中训练好的模型部署在移动机器人上,然后在所搭建的实际环境中进行测试,机器人在实际环境中的运动如图 15 所示。利用 Rviz 显示 3 种算法所规划的路径如图 16 所示,从图中显示的路径图不难看出,机器人利

用三种不同算法所规划的路径都能避开所有障碍物，并最终达到目标位置。但 SAE-DDQN 算法规划的路径明显更短，并且也是最安全的。证明了本文提出的 SAE-DDQN 算法在实际环境中的可行性和适用性。

图 15　移动机器人在实际环境中的运动

Fig. 15　Motion of mobile robots in real-world environment



(a) DDQN算法 (a) DDQN algorithm　(b) ECMS-DDQN算法 (b) ECMS-DDQN algorithm　(c) SAE-DDQN算法 (c) SAE-DDQN algorithm

图 16　3 种算法在实际环境中的运动路径

Fig. 16　Motion paths of three algorithms in a practical environment

为比较 3 种算法的性能，本文记录并对比了实际环境中移动机器人所规划的路径长度，规划时间以及步数 3 个指标，如图 17 所示，从图中可以明显看出，SAE-DDQN 算法所规划的路径长度最短，规划时间和步数最少，路径长度比 DDQN 少了 14.6%，比 ECMS-DDQN 少了 10.6%，规划时间比 DDQN 少了 29.4%，比 ECMS-DDQN 少了 17.2%，步数比 DDQN 少了 21.6%，比 ECMS-DDQN 少了 14.0%，说明本文所提出的 SAE-DDQN 算法在实际应用上优于 DDQN 算法和 ECMS-DDQN 算法。

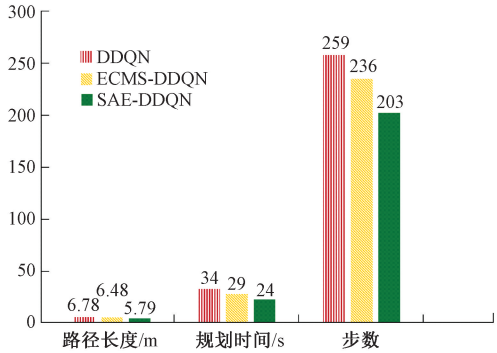


图 17　实际环境中 3 种不同算法的指标对比

Fig. 17　Comparison of performance metrics for three different algorithms in a practical environment

4　结　　论

针对传统 DDQN 算法在路径规划中探索和利用分配不合理，有利数据利用率低和奖励稀疏的问题，提出了自适应探索策略，在其中引入探索成功率平衡探索和利用，使算法收敛更快，利用 DEP 混合采样机制对经验数据进行采样，减小样本之间的相关性，提高了有利样本的利用率，基于 APF 的思想设计奖励函数，使机器人能够获得更多的单步奖励，改善了传统 DDQN 算法奖励稀疏的问题。仿真实验结果表明，在不同的环境下，所提出的 SAE-DDQN 算法规划的路径长度最短，规划时间和步数最少，有效解决了 DDQN 算法对有利样本利用度低，奖励稀疏等问题，最后在实际环境中验证了所提算法的可行性和适用性，实验结果表明该算法路径规划效率更高。但 SAE-DDQN 算法目前只是针对静态环境中的路径规划问题，没有考虑动态障碍物的情况，因此，后续的研究将会针对动态环境下的路径规划问题来开展。

参考文献

[1]　QIN H, QIAO B, WU W J, et al. A path planning algorithm based on deep reinforcement learning for mobile robots in unknown environment[C]. 2022 IEEE 5th Advanced Information Management, Communicates, Electronic and Automation Control Conference(IMCEC). IEEE, 2022, 5: 1661-1666.

[2]　林俊志，席万强，周俊，等. 基于改进 PRM 和 APF 的移动机器人路径规划[J]. 国外电子测量技术，2022，41(12):1-6.
LIN J ZH, XI W Q, ZHOU J, et al. Path planning for mobile robots based on improved PRM and APF[J]. Foreign Electronic Measurement Technology, 2022, 41(12): 1-6.

[3]　王旭扬，梁志伟，高翔，等. 基于改进 DWA 算法的足球机器人局部轨迹规划[J]. 国外电子测量技术，2023，42(8):1-9.
WANG X Y, LIANG ZH W, GAO X, et al. Local trajectory planning for soccer robots based on an improved DWA algorithm[J]. Foreign Electronic Measurement Technology, 2023, 42(8): 1-9.

[4]　张建光，张方，陈良港，等. 基于改进 A* 算法的自动引导车的路径规划[J]. 国外电子测量技术，2022，41(1):123-128.
ZHANG J G, ZHANG F, CHEN L G, et al. Path planning for automatic guided vehicles based on an improved A* algorithm[J]. Foreign Electronic Measurement Technology, 2022, 41(1): 123-128.

[5]　郑弈，谢亚琴. 基于 Dijkstra 算法改进的飞行器航迹快速规划算法[J]. 电子测量技术，2022，45(12):73-79.

ZHENG Y, XIE Y Q. A fast trajectory planning algorithm for aircraft based on improved Dijkstra algorithm[J]. Electronic Measurement Technology, 2022, 45(12): 73-79.

[6] 杨教,陆安江,彭熙舜,等.基于改进粒子群算法的三维路径规划研究[J].电子测量技术,2023,46(12):92-97.
YANG J, LU AN J, PENG X SH, et al. Research on 3D path planning based on improved particle swarm algorithm[J]. Electronic Measurement Technology, 2023, 46(12): 92-97.

[7] KUMAR S, SIKANDER A. Optimum mobile robot path planning using improved artificial bee colony algorithm and evolutionary programming[J]. Arabian Journal for Science and Engineering, 2022, 47(3): 3519-3539.

[8] 王立勇,王弘轩,苏清华,等.基于改进 Q-Learning 的移动机器人路径规划算法[J].电子测量技术,2024,47(9):85-92.
WANG L Y, WANG H X, SU Q H, et al. Path planning algorithm for mobile robots based on improved Q-learning [J]. Electronic Measurement Technology, 2024, 47(9): 85-92.

[9] DARANDA A, DZEMYDA G. Reinforcement learning strategies for vessel navigation[J]. Integrated Computer-Aided Engineering, 2023, 30(1): 53-66.

[10] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Playing atari with deep reinforcement learning[C]. The Workshops at the 26th Neural Information Processing Systems, 2013: 201-220.

[11] 李子怡,胡祥涛,张勇乐,等.基于虚拟目标制导的自适应 Q 学习路径规划算法[J].计算机集成制造系统,2024,30(2):553-568.
LI Z Y, HU X T, ZHANG Y L, et al. Adaptive Q-learning path planning algorithm based on virtual target guidance [J]. Computer Integrated Manufacturing Systems, 2024, 30(2): 553-568.

[12] HASSELT H V, GUEZ A, SILVER D. Deep reinforcement learning with double Q-learning[C]. Proceedings of the AAAI Conference on Artificial Intelligence, 2016, 30(1): 2094-2100.

[13] WANG Y, FANG Y L, LOU P, et al. Deep reinforcement learning based path planning for mobile robot in unknown environment [C]. Journal of Physics: Conference Series. IOP Publishing, 2020, 1576(1): 012009.

[14] CHU ZH ZH, WANG F L, LEI T J, et al. Path planning based on deep reinforcement learning for autonomous underwater vehicles under ocean current disturbance[J]. IEEE Transactions on Intelligent Vehicles, 2022, 8(1): 108-120.

[15] JIANG SH H, SUN SH J, LI C. Path planning for outdoor mobile robots based on IDDQN[J]. IEEE Access, 2024, 12: 51012-51025.

[16] 张磊,母亚双,潘泉.基于改进深度双 Q 网络的移动机器人路径规划算法[J].信息与控制,2024,53(3):365-376.
ZHANG L, MU Y SH, PAN Q. Path planning algorithm for mobile robots based on improved deep double Q-network[J]. Information and Control, 2024, 53(3): 365-376.

[17] ZHANG X, SHI X X, ZHANG Z Q, et al. A DDQN path planning algorithm based on experience classification and multi steps for mobile robots[J]. Electronics, 2022, 11(14): 2120.

作者简介

冷忠涛,硕士研究生,主要研究方向为机器人控制技术。
E-mail:zhongtao_leng@163.com

张烈平,博士,教授,主要研究方向为机器人控制技术,传感器与检测技术。
E-mail:zlp@guat.edu.cn

彭建盛,博士,教授,主要研究方向为智能控制与智能自动化,嵌入式开发与应用。
E-mail:sheng120410@163.com

王艺霖,硕士,主要研究方向为机器人控制技术。
E-mail:YilinWang112233@163.com

张翠(通信作者),硕士,副教授,主要研究方向为传感器与智能信息处理技术。
E-mail:361745092@qq.com