

基于 PBRs-SAC 算法的无人车路径规划研究

杨南禹, 时正华[†]

(河海大学 理学院, 江苏 南京 211100)

摘 要: 针对复杂环境下无人车路径规划问题, 在软演员评论家(SAC)算法的框架下进行改进。通过在奖励函数的设计上融合基于势能的回报塑形(PBRs)思想, 并加入双连帧等训练技巧, 设计了 PBRs-SAC 算法。之后在 Ubuntu 操作系统上搭建基于 Gazebo 的仿真环境, 分别模拟静态与动态实验环境进行训练。最后, 通过消融实验、敏感性测试实验与鲁棒性分析实验验证该算法的有效性。

关键词: 强化学习; 无人车; 势能; 奖励函数; 路径规划

中图分类号: TP18

文献标识码: A

Study on Path Planning for Unmanned Vehicles Using PBRs-SAC Algorithm

YANG Nanyu, SHI Zhenghua[†]

(College of Science, Hohai University, Nanjing, Jiangsu 211100, China)

Abstract: Aiming at the path planning problem of unmanned vehicles in complex environments, an improvement was made within the framework of the soft actor-critic algorithm. The PBRs-SAC algorithm was designed by incorporating the idea of potential-based reward shaping into the reward function design, and by integrating training techniques such as double concatenation frames. Subsequently, a simulation environment based on Gazebo was constructed on the Ubuntu operating system to simulate static and dynamic experimental environments for training. Finally, the effectiveness of the algorithm was assessed through ablation experiments, sensitivity testing experiments, and conducted robustness analysis experiments.

Key words: reinforcement learning; UGV; potential energy; reward function; path planning

自动驾驶技术一直在人工智能领域中扮演着重要角色, 是汽车行业未来的发展方向, 无人车路径规划研究是自动驾驶技术目前面对的核心问题。路径规划是指在给定环境中, 找到从起点到达目标的最优路径的过程。在无人车领域, 路径规划是一个关键问题, 因为它涉及无人车在复杂环境中如何安全、高效地导航。首先, 需要对车辆所在的环境进行地图建模。地图建模可以分为静态地图建模和动态地图建模。在对地图建模之后, 无人车和环境的状态也需要进行适当的表示。无人车状态通

常包括位置、方向、速度等信息, 而环境状态则包括障碍物的位置、速度等。在使用一定的方法找到路径后, 通常还需要对路径进行优化, 以考虑实际应用中的各种因素, 比如如何应对更加复杂的环境中可能出现的动态障碍物等。

面对复杂和不确定的环境, 许多研究人员提出了路径规划问题的解决方案。最常见的方法是一些控制类算法以及智能仿生类算法, 如 A-Star^[1]、遗传算法、蚁群算法^[2]、人工势场法^[3]等。这类单一的智能仿生算法或传统控制类算法对于研究路

径规划问题,虽然取得了一定的研究成果,但仍存在着一些不可忽视的问题,如收敛速度慢、精度低等。

强化学习是一种基于马尔科夫链决策过程的机器学习方法,它迫使智能体不断接收和处理传感器感知的环境状态信息,通过惩罚和奖励来引导智能体进行学习,使智能体实时和快速地对外界环境变化做出响应。强化学习算法具有不依赖环境模型和先验知识的优点,解决了在线路径规划的试错问题。因此,对于路径规划这类决策问题,运用强化学习算法不失为一种强有力的方案。

Q-Learning^[4]算法是目前强化学习中最经典的算法。该算法构建了一个状态和动作的值函数表,表中数值通过误差反向传播不断修正,智能体会根据值函数表选择对应的动作。然而,这种基于值函数表格的强化学习算法只能处理动作空间和状态空间维度较低的环境,无法解决更接近实际情况的高维度问题。深度学习拥有十分强大的感知能力,能够处理包含大量数据的问题,但其欠缺一定的决策能力。深度强化学习结合了强化学习和深度学习的优势,在帮助智能体应对复杂环境和高维度任务时表现突出。DQN (deep Q network)^[5]作为最经典的深度强化学习算法,不再使用值函数表格,取而代之的是一个值函数网络。DQN 通过神经网络计算出值函数后,使用贪心策略来输出动作。Zhou 等^[6]将 DQN 算法用于路径规划问题,与传统的 A* 算法相比,其搜索速度大幅加快。DQN 算法虽然引入神经网络来代替值函数表格,却仍然无法很好解决连续状态环境下的问题。为解决这类问题,有学者提出 A-C(actor-critic)结构强化学习算法,这类算法通常包含 Actor 部分和 Critic 部分,其中 Critic 对 Actor 获得的奖励进行处理和学习,然后把处理后获得的新奖励传递给 Actor 进行学习,从而通过结合 Critic 和 Actor 两部分网络,得到一个更优的算法框架,其中最经典的算法是确定性策略梯度算法(deep deterministic policy gradient, DDPG)。DDPG 算法可以看成是在 DQN 算法的基础上加入了 A-C 结构,其中 Actor 网络负责学习策略,Critic 网络负责学习动作价值函数。这样的改进使得智能体可以在连续动作上更有效地进行学习。Dong 等^[7]将 DDPG 算法应用于路径规划问题,加速了神经网络的收敛。然而上述方法仍没有很好解决在面对环境中有大

量随机因素,如动态障碍物时的路径规划问题。

比起 DDPG 算法,软演员评论家(soft actor-critic, SAC)算法^[8]引入了最大策略熵来学习策略,使得其鲁棒性显著增强。通过将熵引入强化学习算法,尽可能地随机化策略,使智能体可以更充分地探索状态空间,防止其落入局部最优点,同时也使得智能体在面对复杂环境时更容易做出调整,因而该算法刚被提出就受到众多学者关注^[9-11]。刘庆强等^[12]通过引入优先经验回放技术来改进 SAC 算法,提高了智能体训练过程的收敛速度和稳定性。但是,将 SAC 算法用于路径规划的任务仍未被充分研究。

奖励函数的设计直接关乎智能体在每步的动作选择以及收敛结果,是强化学习算法中最关键的一环。稀疏的奖励函数通常会导致智能体难以获得正奖励,从而影响其学习的速度。针对这一缺陷,人工智能领域著名学者吴恩达提出了基于势能的回报塑形(potential-based reward shaping)技术。基于势能的回报塑形技术在原奖励函数的基础上增加了一项基于当前状态与目标之间距离的势能函数。以路径规划任务为例,当前位置距离目标点越近,获得的势能奖励越大,反之则越小。由于任何状态下都有相对应的势能奖励值,原来稀疏的奖励函数因此而变得更加稠密。本文借鉴这一思想,在 SAC 算法的基础上设计了更加合理的奖励函数。本文工作主要有三方面的贡献:(1)在 SAC 算法中加入双连帧技巧,使其更适应动态环境下的问题;(2)借鉴势能的回报塑形技术,设计合理的奖励函数,引导智能体通过自主学习成功到达目标;(3)在 Gazebo 实验平台进行仿真并呈现实验效果。

1 PBRs-SAC 算法设计

1.1 SAC 算法设计

SAC 算法^[13]基于最大熵原理与 A-C 框架,具有较高的探索性,对于存在多个最优或者次优行为的场景,最大熵原理倾向于让智能体学习到一个随机的策略,使其能够探索更多行为^[14]。A-C 框架主要由 Actor 网络和 Critic 网络构成。Actor 网络采用参数化的策略网络学习最优策略,并将状态 s 作为输入,负责生成动作 a 并和环境交互。为评价动作优劣,另一个参数化的 Critic 网络构建状态 s

到动作 a 的 Q 值函数映射。Critic 网络由两套 Q 网络构成,分别为当前 Q 网络与目标 Q 网络,通过延时更新控制它们之间参数的差异化。目标 Q 网络通过经验池计算目标 Q 值,并传递给当前 Q 网络,同时目标 Q 网络也定期从当前 Q 网络复制最新网络参数。

SAC 算法的具体流程如下:SAC 算法将当前智能体的状态 s_t 输入 Actor 网络中,Actor 网络根据输入的状态得到高斯分布的均值和方差,依概率分布采样输出动作 a_t ,智能体执行动作 a_t ,获取奖励 r_t ,状态变为 s_{t+1} ,并将当前产生的经验 (s_t, a_t, s_{t+1}, r_t) 放入经验池中。经验达到一定的数量后,从经验池中采样出一批经验并计算出两个 Critic 目标网络平均价值的最小值,然后根据公式

$$\nabla_{\varphi_i} \frac{1}{B} \sum (Q_{\varphi_i}(s_t, a_t) - Q_s(r, s_{t+1}))^2 \quad (1)$$

更新两个 Critic 网络,根据公式

$$\nabla_{\theta} \left[\frac{1}{B} \sum_{s \in B} (\min_{j=1,2} Q_{\varphi_j}(s_t, a_t') - \alpha \log \pi_{\theta}(a_t | s_t)) \right] \quad (2)$$

更新 Actor 网络,根据公式

$$\varphi_{ti} \leftarrow \rho \varphi_{ti} + (1 - \rho) \varphi_i \quad (3)$$

更新 Critic 目标网络。不断进行以上操作直至收敛。

1.2 状态空间与动作空间设计

在强化学习算法当中,状态空间信息是整个模型进行决策和评估收益的依据,因而状态空间的设计关乎整个模型最终的收敛效果。本次实验借助雷达传感器来获取无人车的状态,无人车的车载雷达通过在 360° 的范围内均匀发射 120 条射线探测无人车与周围范围内障碍物的距离。具体的状态空间定义如下:

$$s = \{l_a, d_o, d_t, v_a, v_o\} \quad (4)$$

其中, l_a 表示状态信息,包括无人车的位置信息; d_o 表示无人车和附近范围内障碍物的距离; d_t 表示无人车与目标点的距离; v_a 表示无人车的速度; v_o 表示动态障碍物的速度。

SAC 算法可以处理连续状态下的决策空间,本文选择无人车的速度和方向作为主要执行动作,并使得智能体可以朝着任意方向行走,以此更加模拟现实情况。具体定义为

$$a \in \{v, w\}, v \in [0, 1], w \in [-\pi, \pi] \quad (5)$$

其中, a 表示动作; v 表示速度大小; w 表示

方向。

在 SAC 算法过程中,输入经验回放池的经验通常为—组 (s_t, a_t, s_{t+1}, r_t) 状态对,然而在动态环境中,近乎—帧的经验显然不如多组连续的状态所提供的价值大,因此将两个连续的状态对 (s_t, a_t, s_{t+1}, r_t) 与 $(s_{t+1}, a_{t+1}, s_{t+2}, r_{t+1})$ 作为—组经验输入到经验池当中,使模型能更好地适应动态环境。

1.3 基于势能的回报塑形奖励函数设计

奖励函数的设计通常直接决定了整个强化学习模型的好坏,为设计更加稠密的奖励函数,引入基于势能的回报塑形技术。在障碍物附近、目标点附近与其余位置分别设立基于势能的回报塑形奖励函数,使得无人车在进入该范围内时获得与平常时刻不同的奖励或惩罚。具体设计如下。

①当智能体与障碍物相撞,即规定智能体与障碍物之间距离小于 0.2 时,得到一个大的惩罚:

$$r_1 = -100 \quad (6)$$

②当智能体到达目标点,即智能体与目标点之间的距离小于 0.2 时,获得一个正奖励:

$$r_2 = 100 \quad (7)$$

③当智能体处于障碍物附近,即规定智能体与障碍物之间距离大于 0.2 同时小于 1.5 时,每步都会获得一个较大的惩罚:

$$r_3 = 0.8 \times (d_{pret} - d_t) - \frac{2}{d_r} - 1 \quad (8)$$

④当智能体处于目标点附近,即规定智能体与目标点之间距离大于 0.2 同时小于 1.5 时,每步都会获得一个较大的奖励:

$$r_4 = 1.2 \times (d_{pret} - d_t) + \frac{1}{d_t} - 1 \quad (9)$$

⑤当智能体处于其他情况时,每步都会获得一个较小的惩罚:

$$r_5 = 0.9 \times (d_{pret} - d_t) - 0.5 \times \frac{d_{pret}}{d_r} - 1 \quad (10)$$

其中, d_{pret} 表示上一时刻智能体与目标点之间的距离; d_t 表示这一时刻智能体目标点的距离; d_r 为智能体与最近障碍物的距离; d_{pret} 表示上一时刻智能体与最近障碍物的距离。

$$R = r_1 + r_2 + r_3 + r_4 + r_5 \quad (11)$$

本实验中实验终止情况分为四种:智能体与障碍物相撞、智能体走出环境范围、回合到达最大步长和智能体到达目标点。具体算法见表 1。

表 1 PBRS-SAC 算法伪代码

算法 1:PBRS-SAC 算法	
输入:环境状态	输出:最优策略
初始化算法参数、各网络参数以及经验池。	
for episode=1 to MAXepisode do	
for step=1 to Y do	
获得 Actor 网络的 a_t 并将状态更新至 s_{t+1} ,	
并根据相对应的状态获得不同的奖励值 r ;	
for step=2 to Y do	
将 $(s_{t-1}, a_{t-1}, s_t, r_{t-1})$ 与	
(s_t, a_t, s_{t+1}, r_t) 作为一组经验存储至经验池;	
end for	
if step mod K == 0 then	
从经验池中采样 B	
计算 Critic 目标网络的 Q 值	
根据公式(1)更新 Critic 权重	
根据公式(2)更新 Actor 权重	
根据公式(3)更新 Critic 目标网络权重;	
end if	
end for	
end for	

2 仿真实验

本节在只包含静态障碍物的简单环境与包含静态及动态障碍物的复杂环境下分别进行对比实验,并加入参数敏感性分析与鲁棒性测试,以验证本文算法的有效性。

2.1 实验仿真平台 Gazebo

机器人操作系统(robot operating system, ROS)是一个适用于机器人编程的操作系统,它提供了一个结构化的通信层,为基于 ROS 的应用程序之间搭建了一个沟通的桥梁,帮助使用者更方便地进行机器人算法开发^[15]。

Gazebo 是 ROS 中的一个实现物理仿真的仿真平台。它支持机器人开发所需的机器人、传感器和环境模型,并且通过基于 ODE 的物理引擎使得在 Gazebo 上进行的实验能够得到逼真的仿真结果,这也使得其在仿真机器人上发挥着重要作用。因此,本节全部工作基于 Gazebo 进行。

2.2 静态训练环境

在 Gazebo 仿真平台中搭建一个如图 1 所示由六个静态障碍物组成的静态训练环境(其中蓝色柱子代表静态障碍物),在环境中对无人车进行约 1400 个回合的训练之后,实验结果如图 2 和图 3 所示。

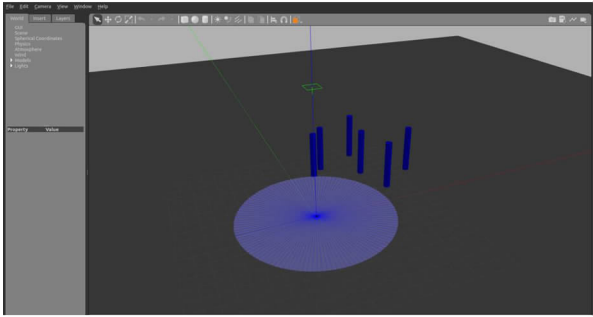


图 1 静态训练环境

在经过约 600 个回合的训练之后,模型就已经获得了较好的训练效果,通过图 2 与图 3 不难看出无人车已经学习到了最优的路径,并在很长一段时间内都有着接近 100% 的实验成功率。

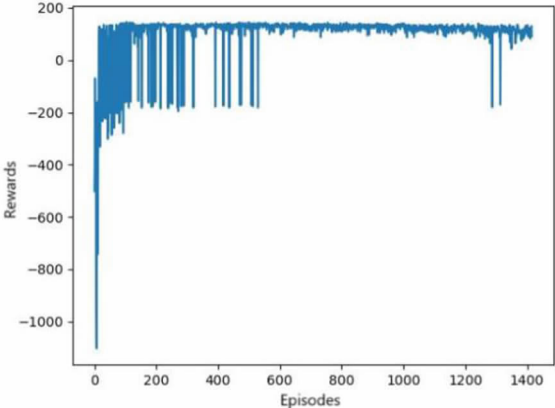


图 2 静态训练环境下的总奖励图

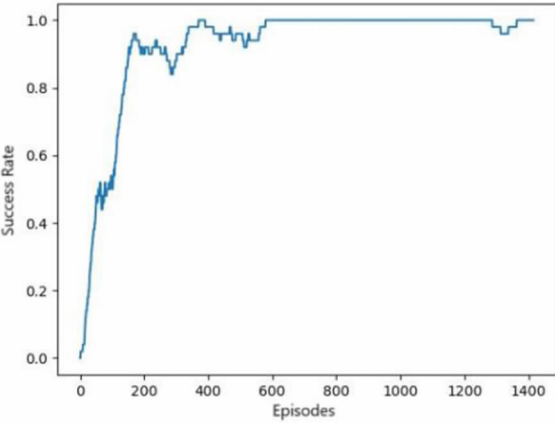


图 3 静态训练环境下的成功率图

2.3 动态训练环境

在静态环境的基础上,加入动态障碍物以构成动态训练环境,如图 4 所示。其中白色柱子根据事先编辑好的路线进行匀速直线运动,蓝色柱子仍为静态障碍物。同时,将去掉奖励函数中③与④的部分作为普通 SAC 算法进行对照分析。实验结果如图 5~图 8 及表 2 所示。

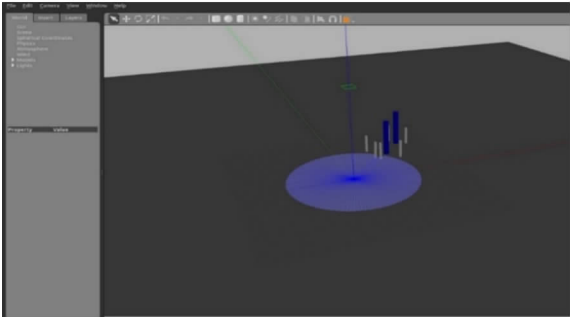


图 4 动态训练环境

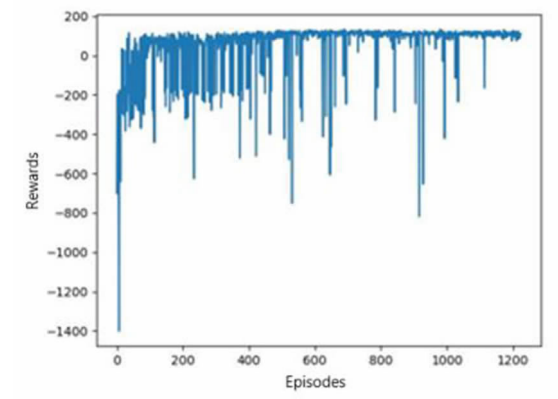


图 5 PBRS-SAC 算法总奖励图

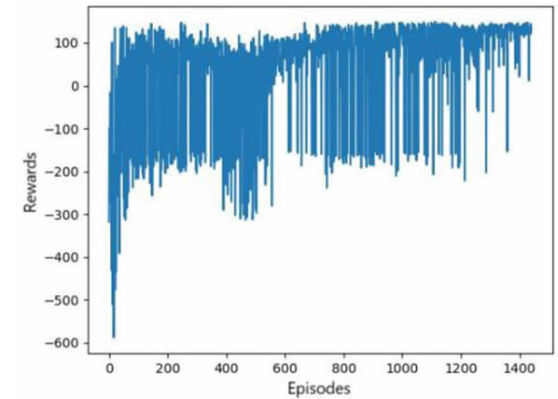


图 6 SAC 算法总奖励图

由图 5 和图 6 可知,在 1200 个回合之后,PBRS-SAC 算法的总奖励值已有明显的收敛趋势,这表明智能体已经学习到了一个较优的策略。而多训练 200 个回合的未加改进的 SAC 算法在总奖励上仍没有一个明显的收敛趋势。图 7 与图 8 是两个算法的成功率图。就成功率而言,PBRS-SAC 算法效果明显更好。尽管在大量训练之后,所提出的算法仍有任务失败的情况,不过其失败次数较少,且波动相对较小。由表 2 可见,PBRS-SAC 算法的成功回合平均用时与成功回合平均搜索路径长度均小于未加改进的 SAC 算法,同时前 1000 个回合的成功率也远高于未加改进的 SAC 算法,这更加体现出所提出的算法效果更好。

2.4 参数敏感性分析

在仿真实验的过程中,大量的参数对模型造成

了一定的不确定性。其中,通过调试其中部分参数并重复进行六个动态障碍物、两个静态障碍物的实验,计算每次实验前 1000 回合的总成功率。通过不断实验,分析得出参数敏感性与参数实际取值,见表 3 与表 4。

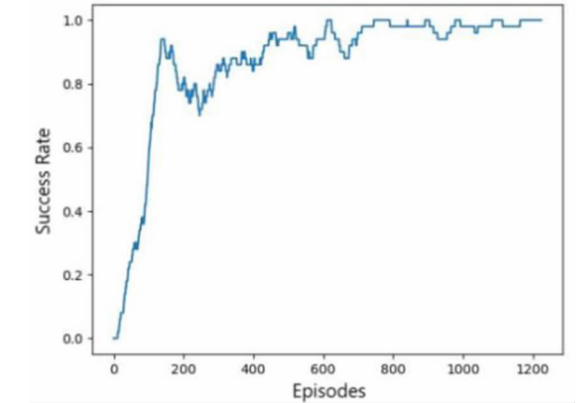


图 7 PBRS-SAC 算法成功率图

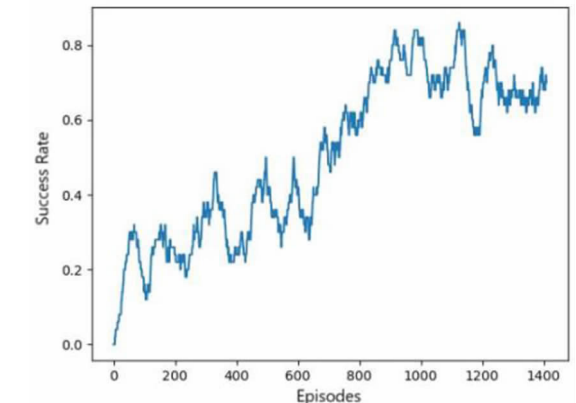


图 8 SAC 算法成功率图

表 2 动态训练环境实验结果

算法	成功回合平均用时	成功回合平均搜索路径长度	成功率
SAC	31.98	15.33	0.58
PBRS-SAC	29.31	14.28	0.81

表 3 参数敏感性分析实验

经验池容量	总成功率	训练批次大小	总成功率
10000	0.74	128	0.77
50000	0.75	256	0.81
100000	0.81	512	0.79
150000	0.80	1024	0.71
学习率	总成功率	每次迭代最大回合数	总成功率
0.0001	0.77	100	0.28
0.0003	0.86	300	0.75
0.0006	0.86	500	0.77
0.001	0.80	700	0.78
每步执行时长	总成功率	折扣因子	总成功率
0.1	0.56	0.99	0.80
0.15	0.71	—	—
0.2	0.84	—	—
0.25	0.83	—	—

表 4 参数实际取值

参数	实际取值
经验池大小	100000
训练批次大小	256
每步动作执行时间	0.2
折扣因子	0.99
学习率	0.0003
每次迭代最大回合数	700

2.5 鲁棒性测试

在原动态环境中再添加两个动态障碍物与一个静态障碍物,形成八个动态障碍物的新环境之后,再将所提算法放入其中进行测试。测试结果显示,多数情况下 PBRs-SAC 算法能在 1000 回合之后获得接近 95%的成功率。因此所提出的算法在更复杂的新环境中也有较好的适用性,具有更好的鲁棒性,实验结果见图 9 与图 10。

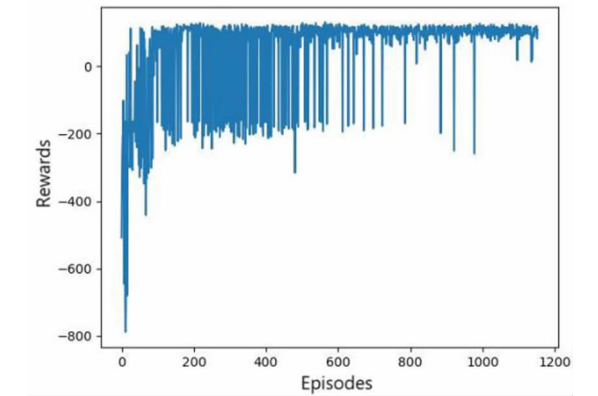


图 9 PBRs-SAC 算法总奖励图

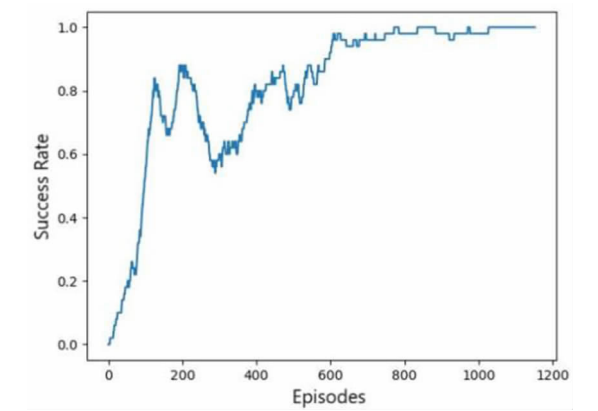


图 10 PBRs-SAC 算法成功率图

3 结 论

在 SAC 算法框架下设计了 PBRs-SAC 算法,以解决具有动态和静态障碍物的复杂环境下无人车路径规划问题。通过使用双连帧实验技巧以及

设计基于势能的回报塑形奖励函数,使训练效率得到提高。最后在 Gazebo 仿真平台上搭建复杂环境,并通过消融实验、敏感性测试实验与鲁棒性分析实验验证该算法的有效性。

参考文献

[1] 陈明,何宁,宏晨,等.改进分层合作 A* 的无人机交通管理中路径规划[J/OL]. 计算机工程与应用,2023;1-10.

[2] RASHID R, PERUMAL N, ELAMVAZUTHI I, et al. Mobile robot path planning using ant colony optimization [C]//2016 2nd IEEE International Symposium on Robotics and Manufacturing Automation (ROMA). IEEE, 2016; 1-6.

[3] 郭明皓,姬鹏,黄海威.基于改进人工势场法的无人车路径规划与跟踪控制[J/OL]. 系统仿真学报,2023;1-10.

[4] WANG S Y, DUAN J J, SHI D, et al. A data-driven multi-agent autonomous voltage control framework using deep reinforcement learning [J]. IEEE Transactions on Power Systems, 2020, 35(6): 4644-4654.

[5] 黄晓冬,苑海涛,毕敬,等.基于 DQN 的海战场船路径规划及仿真[J]. 系统仿真学报,2021, 33(10): 2440-2448.

[6] ZHOU S, LIU X, XU Y, et al. A deep Q-network (DQN) based path planning method for mobile robots [C]//2018 IEEE International Conference on Information and Automation (ICIA). IEEE, 2018; 366-371.

[7] DONG Y, ZOU X. Mobile robot path planning based on improved DDPG reinforcement learning algorithm [C]//2020 IEEE 11th International Conference on Software Engineering and Service Science (ICSESS). IEEE, 2020; 52-56.

[8] 宋紫阳,李军怀,王怀军,等.基于路径模仿和 SAC 的机械臂路径规划算法[J/OL]. 计算机应用,2023;1-7.

[9] 肖硕,黄珍珍,张国鹏,等.基于 SAC 的多智能体深度强化学习算法[J]. 电子学报,2021,49(9): 1675-1681.

[10] 韩金亮,任海菁,吴淞玮,等.基于多层注意力机制-柔性 AC 算法的机器人路径规划[J]. 计算机应用研究,2020,37(12): 3650-3655.

[11] 黄文振,尹奇跃,张俊格,等.基于模型的强化学习中可学习的样本加权机制[J]. 软件学报,2023,34(6): 2765-2775.

[12] 刘庆强,刘鹏云.基于优先级经验回放的 SAC 强化学习算法[J]. 吉林大学学报,2021,39(2): 192-199.

[13] CHEN Y, YING F, LI X, et al. Deep reinforcement learning in maximum entropy framework with automatic adjustment of mixed temperature parameters for path planning [C]//2023 7th International Conference on Robotics, Control and Automation (ICRCA). IEEE, 2023; 78-82.

[14] YU Z, HU H, JIANG M, et al. Optimized SAC deep reinforcement learning control for electro-hydraulic servo systems [C]//2023 IEEE 12th Data Driven Control and Learning Systems Conference (DDCLS). IEEE, 2023; 214-219.

[15] 翁嘉鑫,何坚强,陆群.基于 ROS 平台的移动机器人自主导航技术研究[J]. 自动化技术与应用,2023, 42(4): 5-8.