Advances in Applied Mathematics 应用数学进展, 2025, 14(4), 572-578
Published Online April 2025 in Hans. https://www.hanspub.org/journal/aam
https://doi.org/10.12677/aam.2025.144187

Hans汉斯

基于深度强化学习的路径规划研究

周泰霖

浙江师范大学数学科学学院，浙江 金华

收稿日期：2025年3月18日；录用日期：2025年4月11日；发布日期：2025年4月21日

摘　要

随着自动化和智能系统的发展，高效的路径规划已成为机器人、自动驾驶汽车、无人机导航等领域的关键技术之一。本文主要研究基于深度强化学习的路径规划算法，我们设计了一系列奖励函数提高智能体路径规划能力，最后通过仿真实验验证算法有效性。

关键词

路径规划，深度强化学习，奖励函数

Research on Path Planning Based on Deep Reinforcement Learning

Tailin Zhou

School of Mathematical Sciences, Zhejiang Normal University, Jinhua Zhejiang

Received: Mar. 18th, 2025; accepted: Apr. 11th, 2025; published: Apr. 21st, 2025

Abstract

With the development of automation and intelligent systems, efficient path planning has become one of the key technologies in the fields of robotics, autonomous vehicles, and drone navigation. In this paper, we study the path planning problem based on deep reinforcement learning. We design a series of reward functions to improve the agent's path planning ability, and verify the effectiveness of the algorithm through simulation experiments.

Keywords

Path Planning, Deep Reinforcement Learning, Reward Function

文章引用: 周泰霖. 基于深度强化学习的路径规划研究[J]. 应用数学进展, 2025, 14(4): 572-578.
DOI: 10.12677/aam.2025.144187

1. 引言

路径规划作为自主移动机器人、无人驾驶车辆和智能物流系统等领域的核心技术，其研究水平直接决定着智能体在复杂环境中的自主决策能力。路径规划本质上是一个优化问题，其目标是在环境等多种约束的情况下找到一条最优路径。目前路径规划的算法有很多，传统的路径规划算法比如 A*算法[1]、Dijkstra 算法[2]、快速搜索扩展树(RRT)算法[3]等，这些是全局路径规划算法，全局的环境信息都为已知。同时还有局部路径规划算法，比如动态窗口法[4]、强化学习[5]等。局部路径规划算法并不提前已知全部的环境信息，而是随着运动的进行逐步获得环境信息。

随着研究的深入，路径规划对处理复杂环境能力与实时性提出了更高的要求，而深度强化学习(DRL)能够很好地解决这一点。近年来，关于路径规划的 DRL 方法的研究呈现出显著的增长趋势。深度 Q 网络(DQN) [6]及其改进版本[7]已成功应用于连续状态空间的路径跟踪[8]和避障[9]。基于策略的 DRL 能够解决连续动作空间的路径规划问题。近端策略优化(PPO)在样本效率和学习稳定性之间取得了良好的平衡，并且应用于静态障碍物环境和动态障碍物环境[10]。深度确定性策略梯(DDPG)作为常用的连续空间的强化学习算法，它使用确定性策略函数选择动作，并采用演员－评论家网络结构。[11]通过设计类似 APF 风格的奖励函数增强了 DDPG 的避障性能。[12]使用 DDPG 训练运动规划器以跟踪 A*提供的路径。

虽然 DRL 在路径规划问题的研究上已有很多进展，但仍然存在着收敛速度慢、面对复杂环境处理不佳、泛化能力不足等缺陷。因此更高效的路径规划 DRL 方法仍需要不断的研究与开发。

2. 问题描述

本文所要实现的目标是，在地图上随机分配一个目标位置 $p_g=\left(x_g,y_g\right)$，智能体能够从出发点到目的地自动规划一条安全的路径。智能体的运动看成是二维平面上的运动，智能体的运动学方程为

$$\dot{x}=v\cos\left(\theta\right)$$

$$\dot{y}=v\sin\left(\theta\right)$$

$$\dot{\theta}=\omega$$

3. 状态和动作空间描述

观测向量 $s=\left\{s_g,s_v,s_o\right\}$ 包含三个部分。第一部分 s_g 是表示智能体当前位置与目标位置之间的相对位置差。具体而言位置差在局部坐标系中表示为 $s_g=\left(x_g-x_c,y_g-y_c\right)$，其中 $\left(x_c,y_c\right)$ 是智能体当前的坐标。

第二部分 s_v 表示智能体当前的自身状态，包括智能体的朝向 θ、线速度 v 和角速度 ω。这些值是导航控制的运动反馈。

第三部分 s_0 表示智能体与障碍物的位置关系，包括与每个障碍物的距离。为了能更好应对复杂环境，引入了时间序列信息，因此 s_0 包括最近的两个与障碍物的距离信息。

动作向量 a_n 包括线速度 v_n 和角速度 ω_n，即 $a_n=\left(v_n,\omega_n\right)$。为了符合实际，我们规定线速度和角速度上限为 $v_{\max}$ 和 $\omega_{\max}$。因此，$v_n\in\left[0,v_{\max}\right]$，$\omega_n\in\left[-\omega_{\max},\omega_{\max}\right]$。

4. 奖励函数

奖励函数是强化学习中机器重要的部分，影响着训练结果的好坏。本节设计了一系列奖励函数，用

于提高学习的结果。奖励函数定义如下：

$$r = 100 \cdot \left(r_o + r_g + r_c + r_m \right) \tag{1}$$

r_o 用于检测智能体在导航过程中是否发生碰撞，我们定义 r_s 如下：

$$r_o = \begin{cases} r_{\text{arrival}} & d_{\text{target}}^c < r_{\text{safe}} \\ r_{\text{collision}} & d_{\text{obstacle}}^c < r_{\text{safe}} \end{cases}$$

其中，r_{arrival} 表示成功到达目标时获得的奖励，$r_{\text{collision}}$ 表示与障碍物发生碰撞时受到的惩罚。d_{target}^c 是当前时刻智能体与目标之间的距离，d_{obstacle}^c 是当前时刻智能体与最近障碍物之间的最小距离，r_{safe} 是安全距离。

r_g 用于鼓励智能体往目的地前进。r_g 定义如下：

$$r_g = r_g^d + r_g^a$$

$$\begin{cases} r_g^d = k_1 \cdot \dfrac{d_{\text{target}}^p - d_{\text{target}}^c}{\max\left(d\right)} \\ r_g^a = k_2 \cdot \dfrac{\theta_{\text{target}}^p - \theta_{\text{target}}^c}{\max\left(\theta\right)} \end{cases}$$

其中 d_{target}^p 表示上一时刻与目标的距离，θ_{target}^p 和 θ_{target}^c 分别表示上一时刻和当前时刻与目标方向的偏差。$\max\left(d\right)$ 和 $\max\left(\theta\right)$ 分别表示在时间间隔内距离和方向的最大变化量。

r_c 通过智能体与障碍物的距离变化给予相应惩罚，防止进入危险区域。r_c 定义如下：

$$r_c = r_c^1 + r_c^2$$

$$r_c^1 = \begin{cases} -k_3 \cdot \left[\left(\dfrac{1}{d_{\text{obstacle}}^c} - \dfrac{1}{\rho_0} \right) \Big/ \left(\dfrac{1}{r_{\text{safe}}} - \dfrac{1}{\rho_0} \right) \right]^m & d_{\text{obstacle}}^c < \rho_0 \\ 0 & \text{其他情况} \end{cases}$$

$$r_c^2 = \begin{cases} r_{\text{enter}} & d_{\text{obstacle}}^c < d_{\text{obstacle}}^p < \rho_0 \\ r_{\text{exit}} & d_{\text{obstacle}}^p < d_{\text{obstacle}}^c < \rho_0 \end{cases}$$

其中 d_{obstacle}^p 表示上一时刻智能体与最近障碍物之间的最小距离，r_{enter} 和 r_{exit} 分别为靠近障碍物惩罚和原理障碍物奖励，ρ_0 是障碍物的影响范围，m 是调整惩罚强度的参数。

r_c^1 是根据人工势场法(APF)公式设计的，通过添加 $\dfrac{1}{r_{\text{safe}}} - \dfrac{1}{\rho_0}$ 项进行归一化。

最后，智能体在每个时间步会受到轻微的负奖励 r_m，以鼓励其积极探索，加快学习过程。

5. 算法

本文采用的深度强化学习的基础算法是软演员评论家(SAC)算法，结合第 4 节提出的新的奖励函数实现智能体的路径规划。

SAC 通过引入熵正则化项，在探索和利用之间取得了更好的平衡，从而提高样本的利用效率。同时它使用双重 Q 网络来估计 Q 值，减少了过估计问题，进一步提升了样本质量。此外 SAC 结合最大熵强化学习框架，具有良好的鲁棒性和策略泛化能力，能够避免策略过于贪婪而导致的不稳定。能够缓解深度强化学习在路径规划中存在的样本效率低和训练不稳定的问题。

算法：路径规划的改进 SAC 算法

初始化：初始化 Q 函数参数 δ_1,δ_2、策略网络参数 ϕ、目标 Q 函数参数 $\overline{\delta}_1 \leftarrow \delta_1, \overline{\delta}_2 \leftarrow \delta_2$、温度参数 α、经验回放缓冲区 $\mathcal{D}$、学习率 $\beta_\delta,\beta_\phi,\beta_\alpha$。

迭代：

For 回合 $e=1\rightarrow E$ do

获取初始状态向量 s_1

For 时间步 $t=1\rightarrow T$ do

采样动作 $a_t \sim \pi_\phi(a \mid s_t)$

获取新的状态向量 s_{t+1} 并根据公式(1)计算奖励函数 r_t

将转换 $(s_t a_t, r_t, s_{t+1})$ 存储在回放缓冲区 D

For 训练周期 $k=1\rightarrow K$

从 D 中采样 N 个元组 $\{(s_t,a_i,r_i,s_{t+1})\}_{i=1,\cdots,N}$

更新 Q 函数参数 $\delta_l \leftarrow \delta_l - \beta_\delta \nabla_{\delta_l} J_Q(\delta_l)$ 其中 $l \in \{1,2\}$

更新策略网络参数 $\phi \leftarrow \phi - \beta_\phi \nabla_\phi J_\pi(\phi)$

调整温度 $\alpha \leftarrow \alpha - \beta_\alpha \nabla_\alpha J(\alpha)$

更新目标网络参数 $\overline{\delta}_l \leftarrow \tau\delta_l + (1-\tau)\overline{\delta}_l$ 其中 $l \in 1,2$

End for

End for

End for

6. 实验

在我们的仿真实验中，我们使用了 10×10 的地图用于训练，它被表示为$[0,10]\times[0,10]$的二维平面。在训练过程中，为了保证输入状态向量的多样性，初始位置是固定的，目标位置是在地图范围内随机分配的。

参数设置如下所示：

τ：0.005

折扣率 γ：0.99

全部学习率(Learning rate)：0.001

小批量大小(Mini batch)：256

训练回合数(Episodes)：8000

每回合时间步数(Steps)：300

时间间隔(Δt)：0.1

回放缓冲区容量：100000

最大线速度 $v_{\max}$：0.6

最大角速度 $\omega_{\max}$：$\frac{5}{18}\pi$

$\max(d)$：0.06

$\max(\theta)$：$\frac{5}{180}\pi$

安全半径 r_{safe}：0.6

r_{arrival}：7

$r_{\text{collision}}$：−10
k_1：2
k_2：0.5
k_3：4
m：1.8
ρ_0：0.7
r_{enter}：−1
r_{exit}：0.5
r_m：−0.01

训练完成后，测试的路径规划结果和奖励曲线如图 1 和图 2 所示，图 1 展示了智能体能够对不同目的地给出相应的路径，说明该算法能够实现智能体的路径规划。图 2 展示了训练过程中的奖励曲线。从图中可以看出，随着训练的进行，算法的奖励逐渐增加，并在后期达到了稳定状态，保持了较高水平。由于训练过程中存在随机探索，会存在一些波动但并未不影响算法的整体稳定性。这表明算法已经收敛到一个稳定的策略，并能够持续获得较高的回报，进一步证明了所提出的算法在训练过程中的有效性和稳定性。

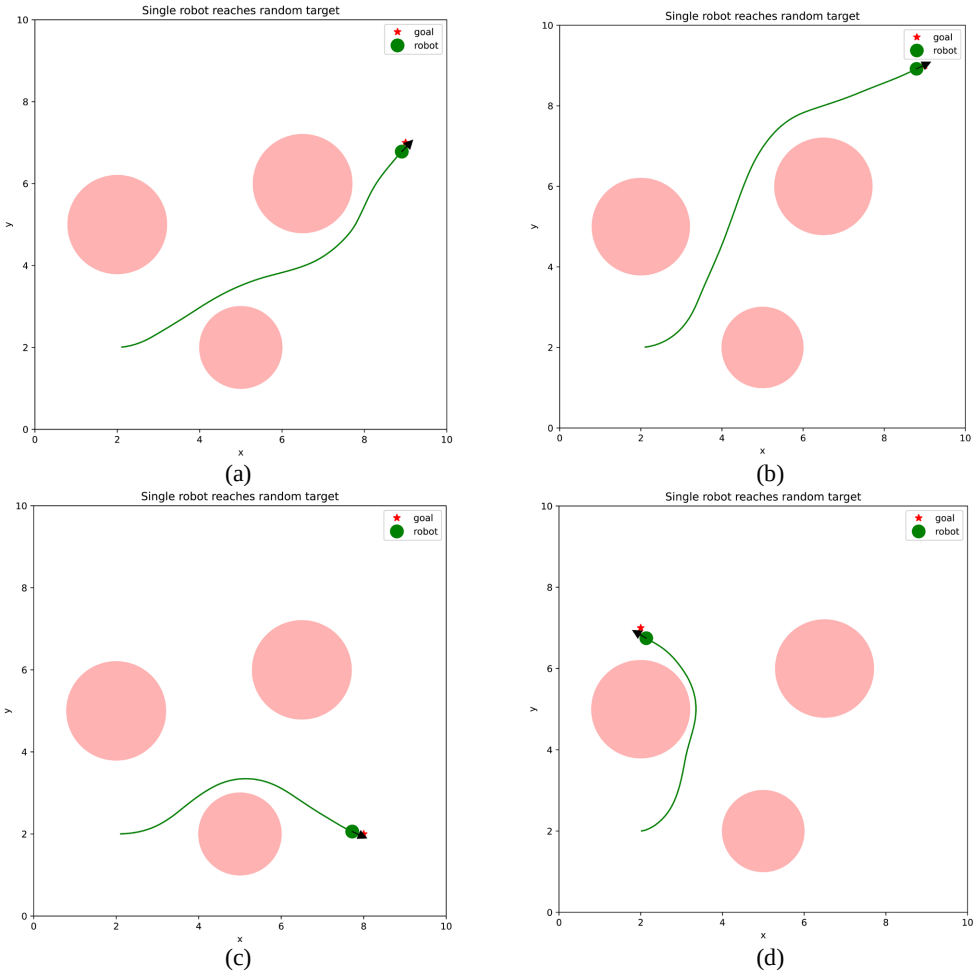

Figure 1. Agent path planning
图 1. 智能体路径规划

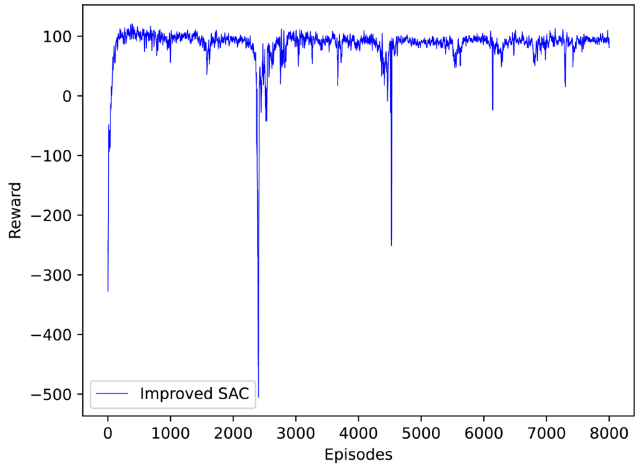

Figure 2. Reward curve
图 2. 奖励曲线

7. 总结

本文设计了一系列奖励函数，包括碰撞检测奖励、目标导航奖励、障碍物避让奖励，其中融合了密集奖励和稀疏奖励，帮助智能体在快速学习和长期规划之间找到平衡，提高其在复杂环境中的适应性和泛化能力。最后结合提出的奖励函数，使用 SAC 算法实现了智能体对不同目标点的路径规划，并给出了训练和测试结果。

参考文献

[1] Bai, X., Jiang, H., Cui, J., Lu, K., Chen, P. and Zhang, M. (2021) UAV Path Planning Based on Improved A∗ and DWA Algorithms. *International Journal of Aerospace Engineering*, **2021**, Article 4511252. https://doi.org/10.1155/2021/4511252

[2] 宋佳. 基于 Dijkstra 算法的 AGV 绿色节能路径规划研究[D]: [硕士学位论文]. 南昌: 南昌大学, 2022.

[3] 向金林, 王鸿东, 欧阳子路, 等. 基于改进双向 RRT 的无人艇局部路径规划算法研究[J]. 中国造船, 2020, 61(1): 157-166.

[4] Ballesteros, J., Urdiales, C., Velasco, A.B.M. and Ramos-Jimenez, G. (2017) A Biomimetical Dynamic Window Approach to Navigation for Collaborative Control. *IEEE Transactions on Human-Machine Systems*, **47**, 1123-1133. https://doi.org/10.1109/thms.2017.2700633

[5] 黄岩松, 姚锡凡, 景轩, 等. 基于深度 Q 网络的多起点多终点 AGV 路径规划[J]. 计算机集成制造系统, 2023, 29(8): 2550-2562.

[6] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., *et al*. (2015) Human-Level Control through Deep Reinforcement Learning. *Nature*, **518**, 529-533. https://doi.org/10.1038/nature14236

[7] Van Hasselt, H., Guez, A. and Silver, D. (2016) Deep Reinforcement Learning with Double Q-Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, **30**, 2094-2100. https://doi.org/10.1609/aaai.v30i1.10295

[8] Kato, Y. and Morioka, K. (2019) Autonomous Robot Navigation System without Grid Maps Based on Double Deep Q-Network and RTK-GNSS Localization in Outdoor Environments. 2019 *IEEE/SICE International Symposium on System Integration* (*SII*), Paris, 14-16 January 2019, 346-351. https://doi.org/10.1109/sii.2019.8700426

[9] Han, S., Choi, H., Benz, P. and Loaiciga, J. (2018) Sensor-Based Mobile Robot Navigation via Deep Reinforcement Learning. 2018 *IEEE International Conference on Big Data and Smart Computing* (*BigComp*), Shanghai, 15-17 January 2018, 147-154. https://doi.org/10.1109/bigcomp.2018.00030

[10] Li, J., Ran, M., Wang, H. and Xie, L. (2021) A Behavior-Based Mobile Robot Navigation Method with Deep Reinforcement Learning. *Unmanned Systems*, **9**, 201-209. https://doi.org/10.1142/s2301385021410041

[11] Sampedro, C., Bavle, H., Rodriguez-Ramos, A., de la Puente, P. and Campoy, P. (2018) Laser-Based Reactive Navigation for Multirotor Aerial Robots Using Deep Reinforcement Learning. 2018 *IEEE/RSJ International Conference on*

Intelligent Robots and Systems (*IROS*), Madrid, 1-5 October 2018, 1024-1031. https://doi.org/10.1109/iros.2018.8593706

[12] Leiva, F. and Ruiz-del-Solar, J. (2020) Robust RL-Based Map-Less Local Planning: Using 2D Point Clouds as Observations. *IEEE Robotics and Automation Letters*, **5**, 5787-5794. https://doi.org/10.1109/lra.2020.3010732