doi:10.6041/j.issn.1000-1298.2026.19.033

基于深度确定性策略梯度自适应 DWA 算法的移动机器人路径规划

聂　进[1]　易向贤[1]　罗洋坤[2]　许建民[3]　宋　雷[3,4]　陈尧箸[3]

(1. 娄底职业技术学院汽车学院, 娄底 417000; 2. 湖南汽车工程职业大学车辆工程学院, 株洲 421000;
3. 厦门理工学院机械与汽车工程学院, 厦门 361000; 4. 泉州信息工程学院机械与电气工程学院, 泉州 362000)

摘要: 针对传统动态窗口法在复杂环境中避障与绕行策略适配性不足的问题,提出基于深度确定性策略梯度(Deep deterministic policy gradient, DDPG)的自适应动态窗口(Adaptive dynamic window approach, ADWA)算法。ADWA 在 DWA 框架中引入 DDPG 强化学习机制,从结构层面对 DWA 进行改进,通过策略网络实现代价函数中关键权重参数的动态自调节。具体地,ADWA 利用激光雷达数据以及机器人与目标点之间的相对位置信息构建 DDPG 状态空间,并将 Actor 网络的输出结果作为 DWA 中航向角偏差代价项的可变权重,引导运动方向决策。通过与环境的持续交互,算法基于奖励机制不断优化策略函数,实现对不同场景环境的自适应调整。在不同环境中进行了仿真与实验验证,实验结果表明,相对于传统的 DWA 算法,ADWA 算法平均成功到达率为 94.7%,且平均绕行率仅为 2.7%,显著提高了机器人在应对各种场景时的灵活性和鲁棒性,并且能够有效提高机器人在复杂环境中的通行效率。

关键词: 移动机器人; 路径规划; 动态窗口法; 深度确定性策略梯度

中图分类号: TP242.6　　**文献标识码**: A　　**文章编号**: 1000-1298(2026)19-0355-10　　**OSID**:

Local Path Planning of Mobile Robot Based on DDPG Adaptive DWA Algorithm

Nie Jin[1]　Yi Xiangxian[1]　Luo Yangkun[2]　Xu Jianmin[3]　Song Lei[3,4]　Chen Yaoruo[3]

(1. *Automotive School*, *Loudi Vocational and Technical College*, *Loudi* 417000, *China*
2. *School of Vehicle Engineering*, *Hunan Automotive Engineering Vocational University*, *Zhuzhou* 421000, *China*
3. *School of Mechanical and Automotive Engineering*, *Xiamen University of Technology*, *Xiamen* 361000, *China*
4. *School of Mechanical and Electrical Engineering*, *Quanzhou University of Information Engineering*, *Quanzhou* 362000, *China*)

Abstract: Aiming to address the poor adaptability of the traditional dynamic window approach (DWA) in obstacle avoidance and detour strategy selection within complex and dynamic environments, an adaptive dynamic window approach (ADWA) enhanced by the deep deterministic policy gradient (DDPG) algorithm was proposed. The proposed framework integrated a reinforcement learning mechanism into the DWA architecture, allowing online optimization of the motion evaluation process through a policy network that dynamically adjusted the weighting coefficients of key cost function components. A continuous DDPG state space was constructed by using laser radar perception data and the relative position between the robot and the target, while the output of the actor network served as an adaptive weighting factor for the heading deviation cost term, guiding motion decisions in real time. Through continuous interaction with the environment and reward-driven policy optimization, the proposed method autonomously refined its decision-making strategy to adapt to varying obstacle distributions and environmental conditions without manual parameter tuning. Extensive simulations and real-world experiments conducted across diverse and cluttered environments demonstrated that the ADWA achieved an average goal-reaching success rate of

收稿日期: 2025-06-05　修回日期: 2025-10-29

基金项目: 湖南省自然科学基金项目(2025JJ70067、2023JJ50512)、湖南省教育厅科学研究项目(24C1201、25C1449)和娄底职业技术学院科研项目(2024WZK002)

作者简介: 聂进(1984—),男,教授,博士,主要从事新能源汽车、机器人路径规划研究,E-mail: nie0822@163.com

通信作者: 宋雷(1999—),男,助教,主要从事机器人路径规划研究,E-mail: songlei15926264173@163.com

94.7% and a detour rate of only 2.7%. Compared with the traditional DWA, the proposed method exhibited superior flexibility, robustness, and navigation efficiency, effectively enhancing the adaptability and overall motion performance of mobile robots in complex and uncertain scenarios. These findings confirmed that integrating DDPG into the DWA framework can provide a scalable and intelligent approach to real-time motion planning for autonomous robotic navigation.

Key words: mobile robot; path planning; dynamic window approach; deep deterministic policy gradient

0 引言

近年来移动机器人被广泛应用于家庭服务[1]、生产运输[2-3]、抢险救灾[4]及农业田间作业等场景。局部路径规划是自主导航中一个重要的研究方向,传统的局部路径规划方法有动态窗口法[5]、人工势场法[6-7]和时间弹性带法[8-9]等。其中,DWA 因其对机器人运动约束的良好适应性和较高的实时性,被广泛应用于多种导航系统中。

为提升 DWA 在复杂环境中的性能,研究者对其进行了多种优化改进,包括虚拟机械臂、模糊控制、引入障碍物密度因子等[10-13],但其仍存在易陷入局部最优、避障与绕行策略固定无法自适应调整等问题。不同于传统的路径规划算法,基于强化学习(Reinforcement learning, RL)的方法能够摆脱对全局地图的依赖,通过在环境中不断试错从而学习最优策略。但由于状态空间过大会造成维度灾难[14-15],以及直接映射速度控制量易产生动作不确定性,难以满足路径稳定性的要求,因此机器人难以利用 RL 在真实世界中导航。为此,许多学者提出了不同的改进方法[16-23]。

综上所述,现有研究在一定程度上提升了 DWA 和 RL 规划算法的性能,但在实际应用中,不同情况需要不同成本函数中权重的最优设置才能使 DWA 得到一条较好的路径,基于 RL 的方法以激光雷达数据输出速度旋量,虽能实现端到端决策,具有较好的自适应能力,也因此无法保证网络不会生成会导致碰撞的指令[24],因此本文在 DWA 结构的基础上,引入深度确定性策略梯度机制,提出一种自适应局部路径规划算法。该方法保留 DWA 算法的运动约束建模优势,同时通过 DDPG 算法对机器人状态的实时监测,动态调整 DWA 算法的代价权重,使得 DWA 无须依赖全局路径以及人为调节权重,提高导航性能,从而提升路径规划的适应性与稳定性。

1 深度强化学习算法

1.1 强化学习

强化学习(RL)是机器学习的一个重要分支,不同于依赖大量有标签样本的监督学习,RL 的核心思想是通过智能体(Agent)与环境交互实时生成带有奖励的训练数据,基于环境反馈的奖励信号自动学习最优策略。该过程可形式化为马尔科夫决策过程,其状态-动作-奖励序列可描述为(S_0,A_0,R_0;S_1,A_1,R_1;…;S_{t-1},A_{t-1},R_{t-1};S_t,A_t,R_t)。式中,S_t为时刻 t 时 Agent 的状态,A_t为时刻 t 时 Agent 采取的动作,R_t为智能体在该状态 S_t - 动作 A_t下所获得的即时奖励。

RL 算法旨在通过最大化 Agent 从当前状态起始的期望累计奖励(Reward)来学习最优策略,累计奖励 G_t 的定义为

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots + \gamma^k R_{t+k+1} = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} (\gamma \in (0,1)) \tag{1}$$

式中 γ——折扣因子

k——未来奖励的时间步偏移量

强化学习根据策略更新机制的不同,可分为 3 类:①基于值函数的强化学习,如 Q - learning。②基于策略梯度的直接策略搜索的强化学习。③值函数与策略函数联合优化的强化学习,如 DDPG 算法。

1.2 DDPG 算法模型

深度确定性策略梯度是联合值函数和直接策略搜索的强化学习算法。DDPG 算法属于 Actor - Critic 框架,结合了策略函数和价值函数的联合优化思想,适用于连续动作空间任务。算法中包含 2 个主要神经网络模块 Actor 和 Critic,Actor 网络用于近似策略函数,输入当前状态S_t,输出当前时刻应采取的动作A_t;Critic 网络用于评估动作质量,即计算状态-动作对的 Q 值,反馈给 Actor 网络以指导其策略优化。为提升训练稳定性,DDPG 引入了目标网络机制,即每个主网络均配有一个软更新的目标网络,网络结构如图 1 所示。Actor - Online 网络根据状态S_t选择动作A_t,Agent 执行动作后,状态转移至S_{t+1},环境返回奖励R_t,交互数据被存储于经验回放池中。在网络更新阶段,从经验池中随机采样批量样本,通过最小化 Critic 网络损失函数和最大化 Actor 网络性能指标进行梯度更新,进而实现策略的迭代优化。DDPG 网络的更新过程为:

(1) 计算 Critic - Online 网络更新的目标 Q 值

$$y_t = R_t + \gamma Q'(S_{t+1}, A_{t+1}) \tag{2}$$

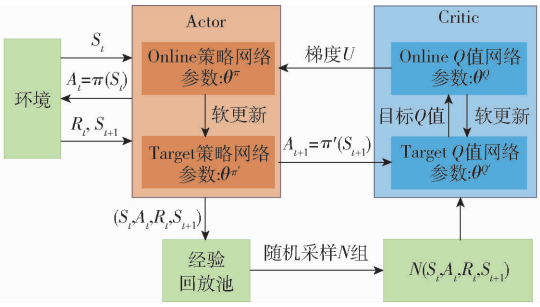


图 1　DDPG 算法网络结构图

Fig. 1　DDPG algorithm network structure

其中　$A_{t+1}=\pi'(S_{t+1})$

式中　y_t——目标 Q 估计值

$Q'(\cdot)$——Critic－Target 网络

$\pi'(\cdot)$——Actor－Target 网络

（2）计算 Critic－Online 网络损失函数 $L(\theta^Q)$

$$L(\theta^Q)=\frac{1}{N}\sum_t (y_t-Q(S_t,A_t))^2 \tag{3}$$

其中　$A_t=\pi(S_t)$

式中　$Q(\cdot)$——Critic－Online 网络

$\pi(\cdot)$——Actor－Online 网络

N——单次从经验回放池中抽取的训练样本批量大小

（3）Critic－Online 网络的 Q 值作为梯度对 Actor－Online 网络进行更新

$$\nabla_{\theta^\pi}U\approx\frac{1}{N}\sum_t \nabla_a Q(S_t,A_t|\theta^Q)\nabla_{\theta^\pi}\pi(S_t|\theta^\pi) \tag{4}$$

式中　$\nabla_{\theta^\pi}U$——策略优化梯度

U——全局累积回报期望

θ^π——在线策略网络 $\pi(\cdot)$ 的参数集合

∇——梯度算子

（4）软更新 Critic－Target 网络和 Actor－Target 网络

$$\begin{cases} Q'\leftarrow\lambda Q+(1-\lambda)Q' \\ \pi'\leftarrow\lambda\pi+(1-\lambda)\pi' \end{cases} \tag{5}$$

式中　λ——软更新参数

2　自适应动态窗口算法

2.1　经典 DWA 算法

DWA 算法首先根据移动机器人的当前速度及其运动学约束建立速度的采样窗口；随后，在该窗口范围内采样多组速度，并基于机器人运动模型生成对应的预测轨迹；最后，借助轨迹评价函数对各候选轨迹进行评分，选取评价值最高的轨迹所对应的速度作为下一时刻的控制输入，从而实现局部路径优化与实时避障。本文中的机器人底盘为差速驱动，如图 2 所示，只考虑它的正向运动和旋转运动。

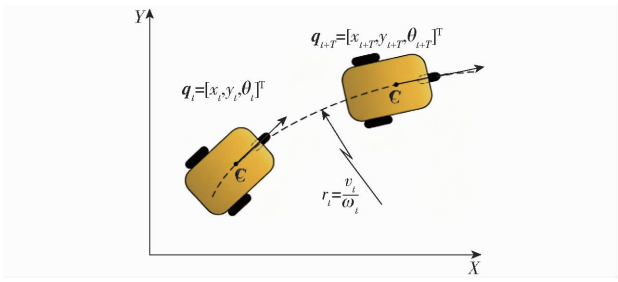


图 2　机器人底盘模型

Fig. 2　Robot chassis model

DWA 算法的主要流程为

（1）建立机器人的运动学模型

$$\begin{cases} \dot{x}=v\cos\theta \\ \dot{y}=v\sin\theta \\ \dot{\theta}=\omega \\ \dot{v}=a \\ \dot{\omega}=\alpha \end{cases} \tag{6}$$

式中　(x,y)——机器人位置坐标

θ——机器人航向角

v、ω——机器人的线速度和角速度

a、α——机器人的线加速度和角加速度

（2）机器人的行驶速度限制

$$\begin{cases} 0\leqslant v\leqslant v_{\max} \\ -\omega_{\max}\leqslant\omega\leqslant\omega_{\max} \end{cases} \tag{7}$$

式中　$v_{\max}$——最大线速度

$\omega_{\max}$——最大角速度

（3）最大加速度限制机器人下一时刻的速度

$$\begin{cases} v_t-a_{\max}\Delta t\leqslant v_{t+1}\leqslant v_t+a_{\max}\Delta t \\ \omega_t-\alpha_{\max}\Delta t\leqslant\omega_{t+1}\leqslant\omega_t+\alpha_{\max}\Delta t \end{cases} \tag{8}$$

式中　Δt——一个预测步的时间步长

$a_{\max}$——最大线加速度

$\alpha_{\max}$——最大角加速度

（4）速度窗口中采样多组 (v,ω)，生成多条预测轨迹，其中每个时间步的预测方法为

$$\begin{bmatrix} x_{t+\Delta t} \\ y_{t+\Delta t} \\ \theta_{t+\Delta t} \end{bmatrix}=\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}+\Delta t\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \tag{9}$$

式中　$[x_t \quad y_t \quad \theta_t]^T$——$t$ 时刻机器人在全局坐标系下的位姿

（5）使用评价函数选出最优轨迹

$$\mathrm{Cost}(v,\omega)=\sigma\begin{bmatrix} C_{\mathrm{heading}} \\ C_{\mathrm{obstacle}} \\ C_{\mathrm{velocity}} \end{bmatrix}[w_1 \quad w_2 \quad w_3] \tag{10}$$

式中　σ——归一化

C_{heading}——预测轨迹终点处机器人的姿态角与目标航向角的偏差代价

$C_{obstacle}$——预测轨迹中的每个坐标点与障碍物的距离代价

$C_{velocity}$——预测轨迹的终点处机器人的速度代价

w_1——机器人航向角偏差代价权重

w_2——障碍物距离代价权重

w_3——机器人速度代价权重

最终使 Cost(v,ω)最小的轨迹即为最优轨迹。

2.2 基于 DDPG 的自适应 DWA 算法

在复杂的真实环境中,使用固定权重的 DWA 算法难兼顾路径跟踪效率与避障安全性,存在一定的适应性局限。即 DWA 算法的成本函数中各因素的最佳权重无法针对所有导航场景进行定义,因为它们取决于机器人周围障碍物的分布,以及成本函数中各因素归一化的细节[24]。例如,当机器人接近目标点且无障碍物遮挡时,应增强目标导向性,即适当增大路径偏差代价$C_{heading}$的权重w_1,以加快机器人收敛至目标点。而当机器人接近障碍物时,则应增强避障能力与提升障碍物距离代价$C_{obstacle}$的权重w_2,以避免碰撞风险。因此,代价函数中的各权重应能随着机器人周围环境以及自身状态的变化而自适应调整,以提升整体决策性能。

然而,仅通过手动设计规则或函数形式,难以穷尽所有可能的环境状态及相应策略,存在泛化能力不足的问题。因此,本文提出一种基于深度确定性策略梯度的自适应 DWA 算法,采用 DDPG 网络结构,通过训练使智能体在不同环境下输出最优的权重值,实现 DWA 算法中代价函数权重的自适应动态调节,从而提升算法在复杂场景中的鲁棒性与灵活性。

2.2.1 DDPG 状态空间的定义

本研究中,机器人搭载二维激光雷达对周围环境进行感知,用于检测障碍物分布情况。状态空间设计综合考虑了机器人周围的环境信息、目标点的相对位置关系以及自身姿态,具体构建方式为

(1)从激光雷达采样数据中选取机器人前方 90°至 270°之间的 40 个方向,以固定角度间隔提取测距结果,组成激光感知部分的状态向量 $\boldsymbol{s}_1$

$$\boldsymbol{s}_1 = [l_1 \quad l_2 \quad \cdots \quad l_{39} \quad l_{40}] \tag{11}$$

式中 l_1、l_2、…、l_{40}——当前激光点与机器人的距离

(2)计算机器人当前位置与目标点之间的欧氏距离,作为状态变量s_2,定义为

$$s_2 = [(x_g - x_c)^2 + (y_g - y_c)^2]^{\frac{1}{2}} \tag{12}$$

式中 (x_c, y_c)——机器人当前的全局坐标

(x_g, y_g)——目标点的全局坐标

(3)引入机器人当前航向角与目标方向之间的偏差角s_3,用于描述姿态与导航方向之间的误差,定义为

$$s_3 = \alpha_c - \arctan\frac{y_g - y_c}{x_g - x_c} \tag{13}$$

式中 α_c——机器人当前时刻在全局坐标中的航向角

因此,状态空间 $S = [\boldsymbol{s}_1, s_2, s_3]$ 共包含 42 个维度的特征量,为提高训练稳定性和收敛速度,所有状态输入在送入 DDPG 网络前均进行了归一化处理,以作为智能体的观测输入。

2.2.2 DDPG 动作空间的定义

在 DWA 算法中,通过计算最小代价 Cost(v,ω)选择当前时刻最优速度,而权重向量[w_1 w_2 w_3]在代价函数的加权计算中起着至关重要的作用。在代价函数中,速度代价的设计目标在于鼓励机器人以尽可能高的速度行驶,防止陷入“低速徘徊”状态,因此本文将机器人速度代价权重w_3设定为常数。

影响轨迹优劣的主要因素为 $C_{heading}$ 与 $C_{obstacle}$。根据代价函数表达式(式(10)),其对应的权重比值w_1/w_2将直接影响策略偏向:$w_1/w_2 > 1$ 时,机器人更倾向于朝向目标点移动,强调路径收敛性;反之,当$w_1/w_2 < 1$ 时,机器人更重视避障安全性,更倾向于远离障碍物区域。

为降低神经网络的训练复杂度,提高收敛速度,并减少推理时的计算负担,本文将使用w_1/w_2作为 DDPG 网络动作空间的输出,通过学习不同环境状态下对w_1/w_2的动态调节,实现代价函数中关键权重的自适应,从而提升 DWA 轨迹选择策略的灵活性与鲁棒性。

2.2.3 DDPG 奖励函数的定义

为引导智能体学习合理的轨迹选择策略,设计复合奖励函数,用于评价在状态空间 S 下执行动作 A 后获得的环境反馈,其值反映该动作对完成任务目标的贡献程度。奖励函数输入包含当前时刻的状态S_{cur}、当前时刻离目标点的距离D_{cur}、上一时刻离目标点的距离D_{pre}、当前时刻的线速度 v 和角速度 ω,输出总奖励 R,完整计算规则为:

(1)初始化总奖励,$R = 0$。

(2)若S_{cur}判定到达目标,执行 $R = R + 300$。

(3)若S_{cur}判定发生碰撞,执行 $R = R - 200$。

(4)若满足停滞条件 $v < 0.1$, $\omega < 0.05$,执行 $R = R - 200$。

(5)若$D_{cur} < D_{pre}$, 执行 $R = R + 0.01$,否则执行 $R = R - 1$。

(6)叠加每步运动基础惩罚 $R = R - 0.5$。

2.2.4　DDPG 神经网络设计

本文 Actor 网络由 4 层神经网络构成，其网络结构如图 3 所示。输入层为 42 维的状态向量，表示机器人当前的环境感知信息与自身状态。输出层为 1 维的动作空间，即 DWA 算法中航向偏差角代价权重与障碍物的距离代价权重的比值w_1/w_2，用于动态调整代价函数中的策略偏好。在输入层与输出层之间设置 2 个隐含层，隐含层 1 由 128 个神经元组成，与输入层使用全连接方式连接，采用 ReLU 激活函数。隐含层 2 由 64 个神经元组成，与隐含层 1 使用全连接方式连接，采用 ReLU 激活函数。输出层与隐含层 2 使用全连接方式连接，采用 sigmoid 激活函数。

Critic 网络与 Actor 网络类似，主要区别是 Critic 网络的输入层包含机器人当前状态和当前动作 2 个部分，共 43 维数据。Critic 网络的输出层是状态-动作对的 Q 值，用于指导 Actor 网络的策略更新。

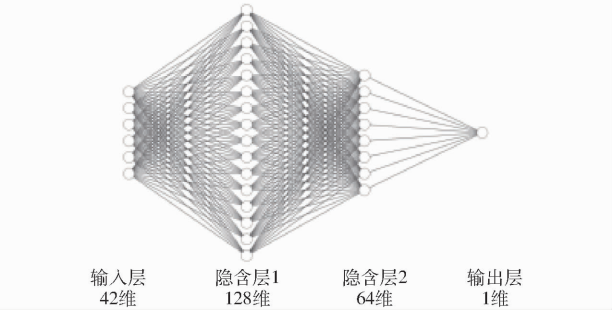


图 3　Actor 网络结构图

Fig. 3　Actor network architecture diagram

3　实验验证

3.1　虚拟仿真实验分析

训练在 Intel Core i7 - 10700F 2.9 GHz CPU、16 GB RAM、Ubuntu 20.04 上进行，开发环境是 VSCode、Python 3.7。DDPG 算法的关键超参数如表 1 所示。

表 1　DDPG 算法参数

Tab. 1　DDPG algorithm parameters

参数	取值
经验池容量	5 000
折扣因子	0.95
软更新参数	0.1
初始探索噪声	0.2
噪声衰减因子	0.999 9
学习率	0.001
一组训练集的大小	150

本文采用融合深度确定性策略梯度机制的 DWA 自适应路径规划算法，并在 3 种不同仿真环境 1、2、3 中进行训练与验证。由于强化学习在训练初期缺乏有效的经验数据，机器人尚未掌握有效策略，因此需要 Actor - Online 网络采取随机选择动作策略，从而提高机器人的探索性能。随着训练次数的增加，机器人在学习到大量数据后，逐渐从探索状态转换到利用状态，此后算法进入收敛阶段。在环境 1 中训练的 Critic - Target 网络的最大 Q 值曲线如图 4 所示，网络在第 780 回合时收敛，说明策略网络在该阶段已学得较为稳定的动作选择策略。

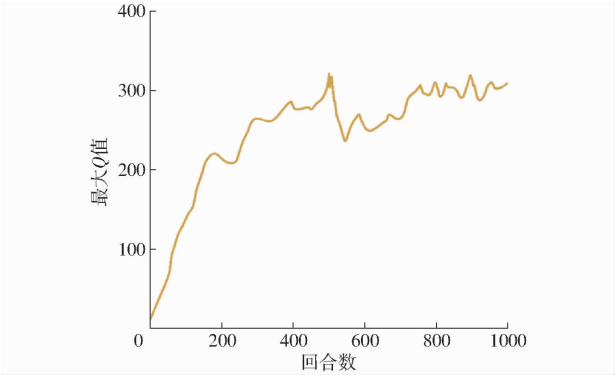


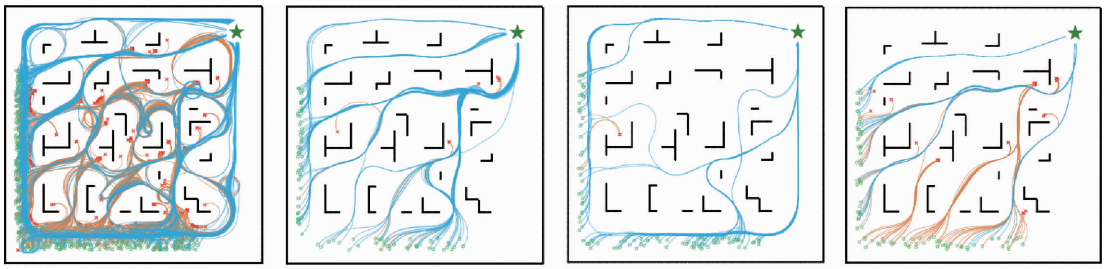
图 4　最大 Q 值曲线

Fig. 4　Maximum Q value curve

图 5a、6a、7a 分别展示了 3 种不同环境下，所提算法的训练过程，即每个训练回合的轨迹，每个训练回合中，系统会在地图左侧与下方构成的 L 形区域内随机选取一个起始点。机器人的初始线速度和初始角速度均为 0，初始航向角为 π/4。每种环境下训练 1 000 回合，当满足以下任意条件之一时，将视为本回合结束，并重新初始化机器人状态进入下一个回合：①机器人成功到达目标点。②与障碍物发生碰撞。③线速度和角速度都低于阈值，进入困境状态。④单次回合步数超过 300 步。

图 5b ~ 5d、图 6b ~ 6d 和图 7b ~ 7d 分别展示了在不同权重配置下以及训练完后机器人在 3 种不同仿真环境中执行 100 回合路径规划任务后的轨迹。其中，图 5b、6b、7b 所示为采用融合 DDPG 策略优化的自适应 DWA 算法所生成的轨迹，其代价权重均为$[w_a,1,1]$，其中w_a为 DDPG 算法动态生成的自适应航向角偏差代价权重。图 5c、6c、7c 为低航向角偏重策略下的轨迹，其 DWA 权重均为[0.2,1,1]。图 5d、6d、7d 则为高航向角偏重策略下的轨迹，其 DWA 对应的权重均为[0.8,1,1]。

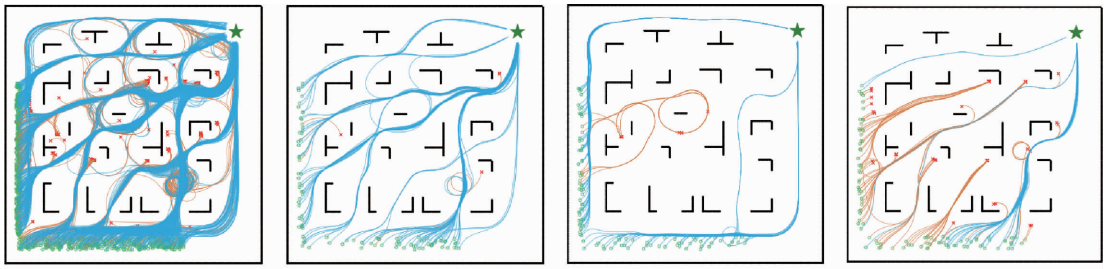
图中路径信息的可视化设计如下：成功到达目标点的路径以蓝色表示，未能成功抵达目标点的路径以橙色表示，绿色圆圈标记每回合的起始位置；机器人在运行过程中出现陷入困境、碰撞障碍物或步



(a) 训练过程 (b) 自适应权重 (c) 低航向角偏差代价权重 (d) 高航向角偏差代价权重

图 5 仿真训练环境 1

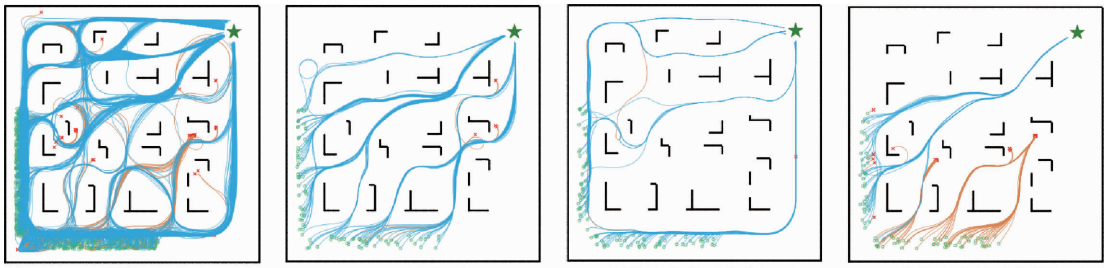
Fig. 5 Simulation training environment 1



(a) 训练过程 (b) 自适应权重 (c) 低航向角偏差代价权重 (d) 高航向角偏差代价权重

图 6 仿真训练环境 2

Fig. 6 Simulation training environment 2



(a) 训练过程 (b) 自适应权重 (c) 低航向角偏差代价权重 (d) 高航向角偏差代价权重

图 7 仿真训练环境 3

Fig. 7 Simulation training environment 3

数超过阈值等失败情况时,其位置均以红色叉号标识。

从图 5b、6b、7b 的观察结果可以得出结论:采用融合 DDPG 策略优化的自适应 DWA 算法后,机器人在大多数回合中能够成功规避障碍物并有效到达目标点,同时实现了较为高效的路径规划,缩短了机器人的行驶路径。相比之下,图 5c、6c、7c 所示为采用固定低航向角偏差代价权重时的路径分布,虽然机器人大多数路径未进入障碍物区域,但其运行路径明显偏长。这主要由于障碍物距离代价权重占主导地位,使机器人更倾向于绕行整个障碍物区域,进而造成路径效率下降和能耗上升。而图 5d、6d、7d 中,采用固定高航向角偏差代价权重的情况下,路径失败率明显上升,多数情况下属于失败路径。此类权重配置中,航向偏差代价在整体代价函数中占据主导地位,使机器人路径选择更强烈地朝向目标点。但对障碍物的规避意图减弱,容易导致机器人在复杂障碍物分布区域内陷入困境,缺乏自主脱困能力,从而影响整体任务完成率。

表 2 展示了在 3 种不同权重配置下,机器人到达目标点的成功率以及绕过整个障碍物区域行驶的绕行率。为客观评估算法自适应性能,本研究对绕行率作如下定义:实验路径起点均随机选取于地图左下方 L 形区域,导致 3 种配置方案累计 300 条有效轨迹的起点存在差异性。考虑到起点非一致性对

表 2 机器人到达率与绕行率

Tab. 2 Robot arrival rate and detour rate %

仿真环境	指标	自适应权重	低航向角偏差代价权重	高航向角偏差代价权重
环境 1	到达率	93	97	35
	绕行率	5	82	0
环境 2	到达率	96	91	28
	绕行率	3	88	0
环境 3	到达率	95	99	37
	绕行率	0	90	0
平均	到达率	94.7	95.7	33.3
	绕行率	2.7	86.7	0

路径长度直接对比的干扰，采用行驶轨迹与起点至终点最短路径的比值作为评价基准，当该比值超过 1 时判定为绕行行为，绕行率即定义为绕行轨迹数量占总轨迹数量的百分比，公式为

$$R_{\mathrm{det}} = \frac{N_{\mathrm{det}}}{N_{\mathrm{all}}} \times 100\% \tag{14}$$

式中　R_{det}——绕行率，%

N_{det}——绕行轨迹总数量

N_{all}——全部实验轨迹总数量

其中，当行驶轨迹长度与起点到终点最短路径长度的比值大于 1.2 时，该轨迹被判定为绕行轨迹。

结果表明，尽管采用低航向角偏差代价权重的 DWA 算法使机器人成功到达目标点的平均概率达到 95.7%，但同时也造成了显著的路径冗余，其绕过整个障碍物区域的轨迹占比也高达 86.7%。在采用上述权重配置的情况下，虽然保证了安全性，但代价是运行路径显著延长，进而导致导航效率下降和能源消耗的增加。与之相反，采用高航向角偏差代价权重的 DWA 算法时，尽管机器人更倾向于直线驶向目标，但对障碍物的规避能力显著下降，最终导致成功到达目标点的平均轨迹数仅占 33.3%，其余多数回合中机器人陷入障碍密集区域，无法完成导航任务，表现出明显的稳定性不足。

相比之下，所提出的采用融合 DDPG 策略优化的自适应 DWA 算法展现出优越的性能表现，该方法在 3 种不同环境下的平均成功率达到 94.7%，同时将平均绕行率控制在 2.7% 以内。这一结果不仅显著优于传统 DWA 固定权重策略，也充分验证了自适应策略对复杂动态环境的高度适应性与良好的鲁棒性。更为关键的是，ADWA 算法生成的路径更加接近最优路径，有效提升了机器人路径规划的整体效率与能耗表现。

3.2　真实环境实验分析

鉴于在真实环境中直接进行强化学习训练存在潜在的安全风险，且实验环境初始化和重置存在较大操作难度与不确定性，为保障训练过程的安全性与效率，本文首先在 Gazebo 仿真平台中构建虚拟环境进行训练，并在训练收敛后将所获得的神经网络模型参数迁移至真实机器人平台 MR500 上进行验证，从而实现算法在真实场景中的可行性评估。

3.2.1　Gazebo 仿真训练

Gazebo 仿真环境部署在 Intel Core i7 − 10700F 2.9 GHz CPU 上，仿真训练中所使用的机器人模型与实际机器人保持一致。在 Gazebo 仿真中，构建了一个与真实环境相同的陷阱地图，如图 8 所示，其尺寸为 560 cm × 165 cm。图 8a 为 Gazebo 仿真环境；图 8b 为 ROS Rviz 的可视化界面，其中蓝色区域表示机器人的初始起点区域，星号标志为导航目标点。在此环境中，如果 DWA 算法的航向角偏差权重较低，则机器人无法通过关卡 1；反之，如果航向角偏差代价权重较高，则机器人会在关卡 2 处陷入困境。这 2 种情况均会导致机器人无法成功到达目标点。此类问题凸显了固定权重策略在面对复杂动态环境时的适应性不足。

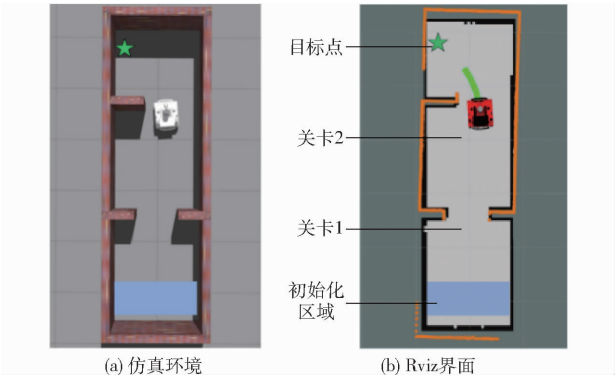


图 8　Gazebo 仿真训练环境

Fig. 8　Gazebo simulation training environment

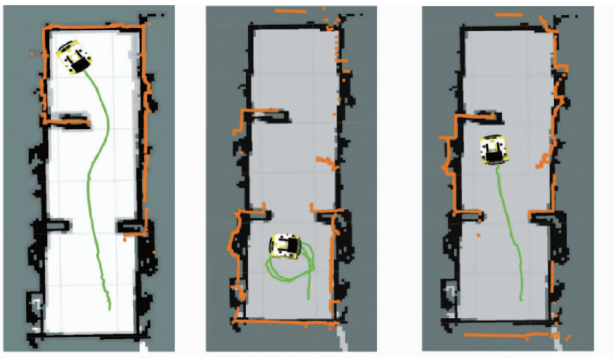
使用 ADWA 算法在仿真环境中进行训练，DDPG 算法遵循表 1 中所列的参数配置。共进行 500 个回合。在训练完成并模型收敛后，将经过训练的模型参数成功地应用于真实机器人，并进行对比实验以评估算法的路径规划性能与适用性。

3.2.2　机器人实验

MR500 机器人的处理器为 Intel Core i7 − 8700T，搭载了 RS16 激光雷达、IMU 和轮式里程计，底盘为四轮差速模型。在真实环境的实验中，系统的输入包含栅格地图、激光雷达获取的实时环境信息、定位算法提供的机器人在地图中的位姿信息，以及目标点的位置坐标；系统的输出为 DWA 算法的航向角偏差代价权重，以便在不同环境条件下进行自适应路径规划，基于上述系统架构，本文在 2 类差异化场景中开展了对比实验 1、2、3，以验证算法的性能。

实验 1 选取相同的起点和目标点，使用本文提出的自适应权重 DWA 算法和低航向角偏差代价权重以及高航向角偏差代价权重的 DWA 算法进行对比实验，不同代价权重的路径规划结果如图 9 所示。

图 9 中，绿色轨迹表示机器人运行轨迹，橙色点云数据为激光雷达探测点，根据图 9b 和图 9c 的观察结果，可以明显看出，DWA 算法使用固定的代价权重存在着特定的局限性。低航向角偏差代价权重的 DWA 算法在面对关卡 1 的狭窄通道时，因其倾向于远离障碍物导致机器人不断绕行而无法通过，耗费了时间和能源。相反，高航向角偏差代价



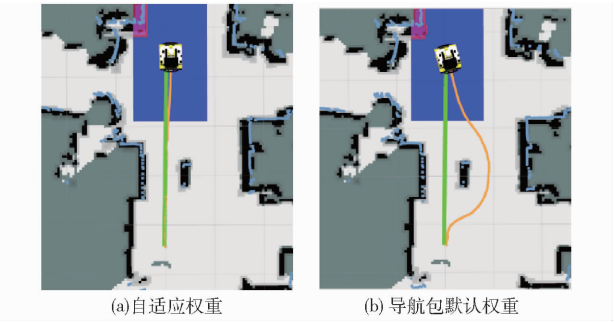
(a) 自适应权重 (b) 低航向角偏差代价权重 (c) 高航向角偏差代价权重

图 9 真实实验 1

Fig. 9 Real experiment 1

权重的 DWA 算法虽然能够通过关卡 1,但因其更趋向于朝着目标点行驶,从而陷入困境无法到达目标点。而基于 DDPG 的自适应 DWA 算法能够很好地解决以上问题,如图 9a 所示,在关卡 1 和关卡 2 处,机器人可以根据不同的环境选择最合适的 DWA 算法的代价权重,使其能够以较短路径高效地到达目标位置。

实验 2 依旧选取相同的起点和目标点,使用本文提出的自适应权重 DWA 算法和 ROS 导航包中默认的权重进行对比实验,不同代价权重的路径规划结果如图 10 所示。



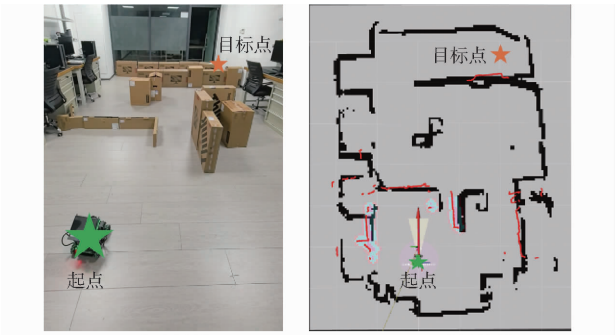
(a)自适应权重 (b) 导航包默认权重

图 10 真实实验 2

Fig. 10 Real experiment 2

图 10 中,绿色轨迹代表起点至终点的最短路径,橙色轨迹为机器人实际运行轨迹,青色点云则对应实时环境探测数据。对比可知,图 10b 中采用 ROS 导航包默认权重配置的 DWA 算法虽能保障机器人安全抵达目标点,但在面对狭窄路口时,因过度保守的避障策略选择绕行,导致路径长度显著增加;而图 10a 中本文提出的自适应权重算法可通过动态调节参数,实现对狭窄路口的精准识别与决策,顺利穿越路口直达目标,有效避免了非必要绕行,充分体现了其在复杂环境下的路径规划优势。

实验 3 依旧选取相同的起点和目标点,使用本文提出的自适应权重 DWA 算法和 ROS 导航包中默认的权重进行对比实验,如图 11 所示。



(a) 实验场景3 (b) SLAM地图结果

图 11 真实实验 3 与 SLAM 地图

Fig. 11 Real experiment 3 and SLAM map

如图 12 ~ 14 所示,左侧部分为小车完整的行驶轨迹(以橙色曲线标记)示意,右侧带有绿色数字标号的为机器人运行过程的连续快照,绿色数字标明对应的时刻,自适应权重算法(图 12)与采用 ROS 导航包默认权重配置 2 的 DWA 算法(图 14),均能使机器人安全抵达目标位置,其中权重为[0.2,1,1]。但采用默认权重配置 1 的 DWA 算法却无法完成该任务(图 13),其中权重为[0.8,1,1]。具体而言,在图 13 中,由于 DWA 算法对航向角赋予了相对较高的权重,机器人倾向于直接朝向目标运动。

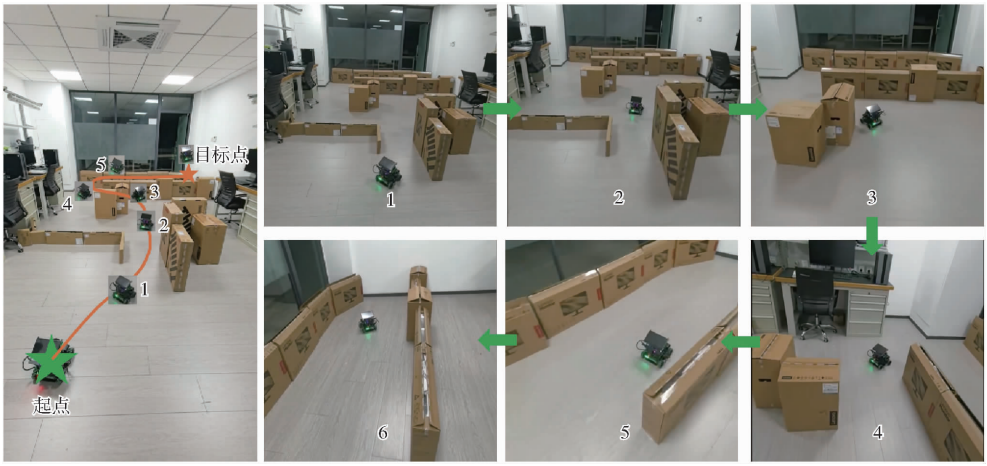


图 12 自适应权重运行实验过程图

Fig. 12 Diagram of adaptive-weight operation experiment process

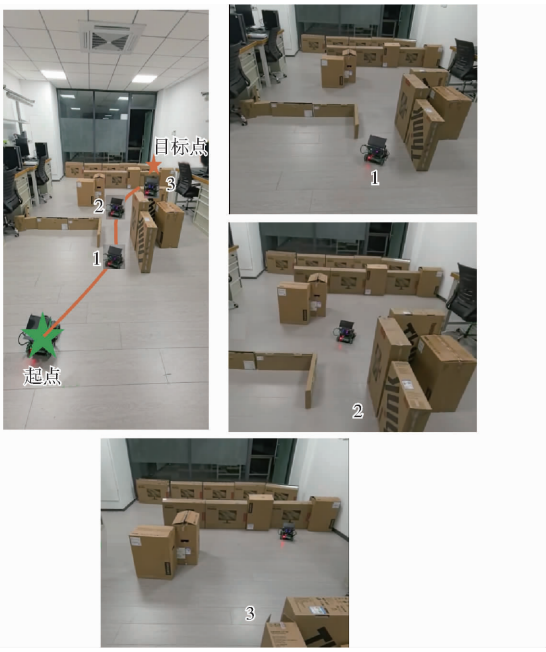


图 13　ROS 导航包默认权重 1 运行实验过程图

Fig. 13　Diagram of ROS navigation package import with default weight-1 operation experiment process

尽管这使得机器人能够穿过狭窄通道，却也导致其无法到达位于墙后的目标位置。相反，在图 14 中，对障碍物距离权重较高的 DWA 算法，会使机器人为避开狭窄通道而绕行。而在图 12 中，本文提出的自适应权重算法在机器人运动过程中动态调整参数。这种动态调整机制让机器人能够成功到达目标位置，同时避免了不必要的绕行，从而验证了所提算法的有效性。

4　结论

（1）本研究针对传统 DWA 算法在机器人路径规划中的固定代价权重无法适应多样化环境和容易陷入困境等问题，提出了一种融合深度强化学习策略的 ADWA。该方法基于 DDPG 算法，通过激光雷达感知周围环境状态，设计了具有代表性的状态空间与奖励函数结构，使得网络能够输出连续动作以动态调整代价函数中关键权重参数，显著提升路径规划策略的灵活性与环境适应能力。

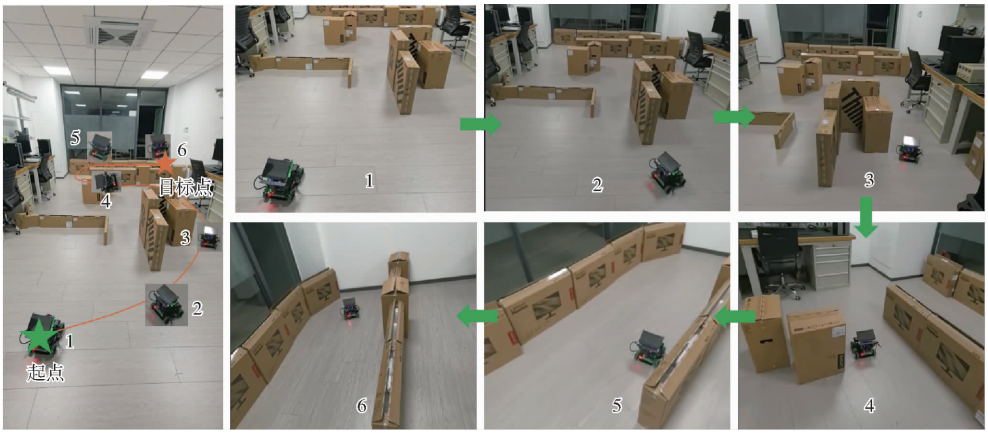


图 14　ROS 导航包默认权重 2 运行实验过程图

Fig. 14　Diagram of ROS navigation package import with default weight-2 operation experiment process

（2）仿真实验结果表明，所提出的 ADWA 算法在多种仿真环境下均展现出优异的避障性能与路径规划效率。其中，平均成功避障率达到 94.7%，显著高于固定权重 DWA 算法，同时平均绕行率仅为 2.7%，有效避免了因保守性过高而导致的路径冗长问题。进一步地，本文将训练完成的模型部署至实际移动机器人平台，实验验证了算法在真实物理环境中具有良好的迁移能力与有效性。

参　考　文　献

[1] Li H, Ding X. Adaptive and intelligent robot task planning for home service: a review[J]. Engineering Applications of Artificial Intelligence, 2023, 117: 105618.

[2] Liu Q, Liu Z, Xiong B, et al. Deep reinforcement learning-based safe interaction for industrial human-robot collaboration using intrinsic reward function[J]. Advanced Engineering Informatics, 2021, 49: 101360.

[3] Kumar S, Sikander A. A novel hybrid framework for single and multi-robot path planning in a complex industrial environment [J]. Journal of Intelligent Manufacturing, 2024, 35(2): 587 – 612.

[4] Ramezani M, Habibi H, Sanchez-Lopez J L, et al. UAV path planning employing MPC-reinforcement learning method considering collision avoidance[C]//2023 International Conference on Unmanned Aircraft Systems (ICUAS). IEEE, 2023: 507 – 514.

[5] 汪小昱，祁子涵，杨震宇，等. 基于 DAV_DWA 算法的农业机器人局部路径规划[J]. 农业机械学报，2025，56(2)：105 – 114.

Wang Xiaochan, Qi Zihan, Yang Zhenyu, et al. Local path planning for agricultural robots based on DAV_DWA[J]. Transactions of the Chinese Society for Agricultural Machinery, 2025,56(2):105 - 114. (in Chinese)

[6] Duhé J F, Victor S, Melchior P. Contributions on artificial potential field method for effective obstacle avoidance[J]. Fractional Calculus and Applied Analysis, 2021, 24: 421 - 446.

[7] 时维国,宁宁,宋存利,等.基于蚁群算法与人工势场法的移动机器人路径规划[J].农业机械学报,2023,54(12):407 - 416.
Shi Weiguo, Ning Ning, Song Cunli, et al. Path planning of mobile robots based on ant colony algorithm and artificial potential field algorithm[J]. Transactions of the Chinese Society for Agricultural Machinery, 2023,54(12):407 - 416. (in Chinese)

[8] Deray J, Magyar B, Solà J, et al. Timed-elastic smooth curve optimization for mobile-base motion planning[C]//2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2019: 3143 - 3149.

[9] Wu J, Ma X, Peng T, et al. An improved timed elastic band (TEB) algorithm of autonomous ground vehicle (AGV) in complex environment[J]. Sensors, 2021, 21(24): 8312.

[10] Kobayashi M, Motoi N. Local path planning: dynamic window approach with virtual manipulators considering dynamic obstacles[J]. IEEE Access, 2022, 10: 17018 - 17029.

[11] Sun Y, Wang W, Xu M, et al. Local path planning for mobile robots based on fuzzy dynamic window algorithm[J]. Sensors, 2023, 23(19): 8260.

[12] Mai X, Li D, Ouyang J, et al. An improved dynamic window approach for local trajectory planning in the environment with dense objects[C]//Journal of Physics: Conference Series. IOP Publishing, 2021: 012003.

[13] Lee D H, Lee S S, Ahn C K, et al. Finite distribution estimation-based dynamic window approach to reliable obstacle avoidance of mobile robot[J]. IEEE Transactions on Industrial Electronics, 2020, 68(10): 9998 - 10006.

[14] Zhou Y, Van Kampen E J, Chu Q. Hybrid hierarchical reinforcement learning for online guidance and navigation with partial observability[J]. Neurocomputing, 2019, 331: 443 - 457.

[15] Alegre L N, Ziemke T, Bazzan A L C. Using reinforcement learning to control traffic signals in a real-world scenario: an approach based on linear function approximation[J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 23(7): 9126 - 9135.

[16] Du Y, Zhang X, Cao Z, et al. An optimized path planning method for coastal ships based on improved DDPG and DP[J]. Journal of Advanced Transportation, 2021, 2021: 1 - 23.

[17] Han H, Wang J, Kuang L, et al. Improved robot path planning method based on deep reinforcement learning[J]. Sensors, 2023, 23(12): 5622.

[18] 田箫源,董秀成.基于改进 DQN 的移动机器人避障路径规划[J].中国惯性技术学报,2024,32(4):406 - 416.
Tian Xiaoyuan, Dong Xiucheng. Obstacle avoidance path planning of mobile robot based on improved DQN[J]. Journal of Chinese Inertial Technology, 2024,32(4):406 - 416. (in Chinese)

[19] Gong H, Wang P, Ni C, et al. Efficient path planning for mobile robot based on deep deterministic policy gradient[J]. Sensors, 2022, 22(9): 3579.

[20] Park M, Lee S Y, Hong J S, et al. Deep deterministic policy gradient-based autonomous driving for mobile robots in sparse reward environments[J]. Sensors, 2022, 22(24): 9574.

[21] Yan C, Chen G, Li Y, et al. Immune deep reinforcement learning-based path planning for mobile robot in unknown environment[J]. Applied Soft Computing, 2023, 145: 110601.

[22] Lobos-Tsunekawa K, Leiva F, Ruiz-Del-Solar J. Visual navigation for biped humanoid robots using deep reinforcement learning[J]. IEEE Robotics and Automation Letters, 2018, 3(4): 3247 - 3254.

[23] Jiang H, Esfahani M A, Wu K, et al. iTD3 - CLN: learn to navigate in dynamic scene through deep reinforcement learning[J]. Neurocomputing, 2022, 503: 118 - 128.

[24] Dobrevski M, Skočaj D. Dynamic adaptive dynamic window approach[J]. IEEE Transactions on Robotics, 2024, 40: 3068 - 3081.