DOI:10.11918/202503079

强化学习驱动的移动机器人预测控制参数整定

刘月笙[1],徐中显[2],贺 宁[1],贺利乐[1]

(1.西安建筑科技大学 机电工程学院,西安 710055;2.西安邮电大学 自动化学院,西安 710121)

摘 要:为提升动态环境下全向移动机器人轨迹跟踪预测控制的性能与适应性,并克服现有基于机器学习的参数调优方法存在的数据依赖性强、短期控制精度与长期系统性能难以兼顾等局限,提出了一种融合强化学习理论与事件触发机制的模型预测控制参数在线自整定方法。首先,建立全向移动机器人运动学模型,并构建对应的轨迹跟踪模型预测控制框架。其次,提出了一种融合 Actor-Critic 强化学习的参数动态优化框架,通过构建联合状态误差与动态性能指标的奖励函数,驱动控制器实时优化控制参数。进一步地,将事件触发机制深度协同于参数优化框架,构建自适应控制器,通过减少参数更新频率以降低计算负载,实现高效控制。最后,搭建全向移动机器人实物实验平台,在阶跃轨迹、Lemniscate 曲线追踪,以及动态避障等多场景下开展对比实验。结果表明,相比于采用静态参数的传统模型预测控制方法,所提方法在阶跃轨迹跟踪场景中降低了约70%的超调量和调节时间,在 Lemniscate 轨迹跟踪场景中降低了约65%的状态偏差,在动态避障场景中降低了约30%的状态偏差,从而验证了该方法在复杂动态环境下提升轨迹跟踪性能的有效性及环境适应能力。本研究为解决动态不确定环境下移动机器人的高性能轨迹跟踪控制难题提供了新的思路和途径。

关键词:全向移动机器人;模型预测控制;参数整定;强化学习;Actor-Critic 框架;事件触发

中图分类号:TP242 **文献标志码**:A **文章编号**:0367-6234(2026)04-0212-11

Reinforcement learning-driven parameter tuning for mobile robot's predictive control

LIU Yuesheng[1], XU Zhongxian[2], HE Ning[1], HE Lile[1]

(1. School of Mechanical and Electrical Engineering, Xi'an University of Architecture and Technology, Xi'an 710055, China;
2. School of Automation, Xi'an University of Posts and Telecommunications, Xi'an 710121, China)

Abstract: To enhance the performance and adaptability of trajectory tracking predictive control for omnidirectional mobile robots in dynamic environments and address the limitations of existing machine learning-based parameter tuning methods, such as strong data dependency and the difficulty in balancing short-term control accuracy with long-term system performance, this paper proposed an online self-tuning method for model predictive control (MPC) parameters. This method integrated reinforcement learning theory with an event-triggered mechanism. First, the kinematic model of the omnidirectional mobile robot was established, and a corresponding trajectory tracking MPC framework was constructed. Second, a dynamic parameter optimization framework incorporating the Actor-Critic reinforcement learning was introduced. By designing a reward function that combines state errors and dynamic performance metrics, the controller was driven to optimize control parameters in real time. Furthermore, the event-triggered mechanism was seamlessly integrated into the parameter optimization framework to develop an adaptive controller. This integration reduced the frequency of parameter updates, thereby lowering computational load and enabling efficient control. Finally, a physical experimental platform for omnidirectional mobile robots was developed, and comparative experiments were conducted across multiple scenarios, including step trajectory tracking, Lemniscate curve tracking, and dynamic obstacle avoidance. Experimental results demonstrate that compared to traditional MPC methods using static parameters, the proposed approach reduces overshoot and adjustment time by approximately 70% in step trajectory tracking, decreases state deviation by approximately 65% in Lemniscate trajectory tracking, and reduces state deviation by approximately 30% in dynamic obstacle avoidance scenarios. These results validate the effectiveness and environmental adaptability of the proposed method in enhancing trajectory tracking performance in complex dynamic environments. This research provides novel insights and approaches for addressing the challenges of high-performance trajectory tracking control of mobile robots in

收稿日期:2025-03-29;**录用日期**:2025-07-19;**网络首发日期**:2026-01-29
网络首发地址:https://link.cnki.net/urlid/23.1235.T.20260129.1121.007
基金项目:国家自然科学基金(62473301);陕西省自然科学基础研究计划项目(2024JC-YBQN-0703)
作者简介:刘月笙(1995—),男,博士研究生;贺 宁(1989—),男,教授,博士生导师;贺利乐(1963—),男,教授,博士生导师
通信作者:徐中显,xuzhongxian@xupt.edu.cn

dynamic and uncertain conditions.

Keywords: omnidirectional mobile robot; model predictive control; parameter tuning; reinforcement learning; Actor-Critic network; event triggering

随着自动化技术的飞速发展，机器人在自动驾驶、智能物流、环境监测等领域的应用日益广泛。移动机器人凭借其独特的全向运动能力，在动态复杂环境中展现出卓越的轨迹跟踪性能，成为工业自动化场景中不可或缺的智能载体[1-2]。

模型预测控制（model predictive control，MPC）凭借其多目标优化和约束处理能力，已成为机器人轨迹跟踪领域的核心控制方法[3-4]。MPC 通过滚动时域优化机制，基于当前状态与预测模型生成最优控制序列，实现动态环境下的闭环控制。然而，MPC 控制性能高度依赖于控制参数的合理配置，因此，如何有效整定这些控制参数成为提升性能的关键。

传统 MPC 参数整定方法多依赖经验驱动的启发式调参或基于规则的参数配置[5-7]。这类方法在静态场景中具有一定可行性，但在目标轨迹突变、外部扰动频发的动态环境下，固定参数 MPC 常因超调量过大、响应滞后等问题，造成跟踪精度显著下降。尤其当移动机器人在多障碍物干扰引发的紧急避障与轨迹重规划场景中，传统方法的参数自适应能力不足将进一步加剧控制性能的恶化。因此，如何实现 MPC 参数的在线自适应优化，已成为提升动态环境下机器人控制鲁棒性的关键挑战。

近年来，随着数据驱动技术的迅速发展，机器学习方法为 MPC 参数自整定提供了新思路[8-12]。IRA 等[8]提出一种基于专家反馈与多臂老虎机算法的多变量 MPC 参数整定方法，通过交互式探索实现动态环境下的控制器鲁棒性优化。Moumouh 等[9]设计了一种基于人工神经网络的 MPC 参数自适应整定框架，利用神经网络在线学习预测模型与性能指标的隐式关系，显著降低动态轨迹跟踪的稳态误差。Jardine 等[10]采用学习自动机优化 MPC 预测时域与控制时域参数，在四旋翼无人机轨迹规划中提升实时性，但其离散动作空间设计导致参数优化粒度粗糙。Liu 等[11]提出了一种波浪船舶路径跟踪和横向稳定的控制方法，基于投影神经网络建立动态系统，并优化 MPC 代价函数，从而获得良好的性能。贺宁等[12]提出了一种融合模糊 C 均值聚类、极限学习机与改进踝骨粒子群的 MPC 参数整定算法，通过数据聚类与分层优化策略优化 MPC 控制参数，从而提高鲁棒性并减少整定时间。Liu 等[13]实现了具有曲率自适应参数调优的双层 MPC 架构，可为高速应用实现控制参数的自动调整。Dai 等[14]提出了一种 MPC 自适应预测参数调整策略，通过预测特性自适应优化跟踪性能。Lin 等[15]建立了一个模糊控制参数调谐框架，该框架能够根据速度输入自动调整 MPC 的预测和控制时间时域。上述研究虽然通过机器学习方法学习系统动态特性，实现了自动调整控制参数，然而仍然存在局限性：1）计算复杂度高，难以满足实时控制要求；2）对高质量训练数据依赖性强，增加实际应用难度；3）难以实现短期控制精度与长期性能的动态平衡，易引发策略振荡或系统失稳。

为突破上述局限，本文聚焦强化学习在动态参数优化中的独特优势。相较于其他机器学习方法，强化学习通过在线交互机制能更好适应环境变化，但直接应用于实时控制面临双重挑战：一方面，传统强化学习依赖大量试错训练，难以满足移动机器人控制的高效性要求；另一方面，学习策略的可解释性不足导致其在安全敏感场景应用受限。针对这些特定挑战，本文融合专家经验与在线优化机制：通过专家经验引导的预训练减少初始探索成本，结合事件触发的条件激活策略压缩在线计算负载；同时构建参数整定框架，使强化学习在模型预测控制的安全边界内运行，确保系统约束的刚性继承。该设计在保留强化学习动态适应优势的同时，显著降低了学习成本与行为不确定性。

基于此融合框架，本文提出了一种强化学习驱动的 MPC 参数整定方法。通过构建基于 Actor-Critic 网络的在线优化框架，设计状态误差与性能指标的联合奖励函数，实现控制参数的实时动态调整；进一步引入事件触发机制，仅在跟踪误差超限或控制输入饱和时激活参数更新，显著降低计算负载。

1　运动学建模与模型预测控制器设计

移动机器人的高精度轨迹跟踪控制需要准确的系统模型和高效的控制策略。本文详细介绍移动机器人的运动学建模和 MPC 控制器设计，为后续的 MPC 参数整定策略设计奠定基础。

1.1　全向移动机器人运动学建模

全向移动机器人具有在水平面上实现任意方向自由移动的能力，在狭窄环境中表现出极强的适应性。本文研究的全向移动机器人搭载 4 个 Mecanum 轮（图 1），这种结构使得运动学模型较为复杂，但提供了灵活的运动能力。

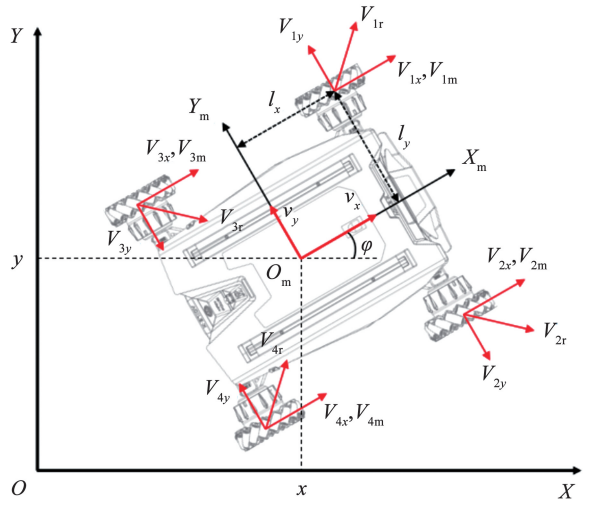


图 1 全向移动机器人运动学模型

Fig. 1 Kinematic model of omnidirectional mobile robot

为了建立在水平面上运动学模型，首先建立全局坐标系 XOY，然后建立机器人坐标系 $X_mO_mY_m$ 原点 O_m 与机器人几何中心重合，X_m 轴为机器人纵向轴，Y_m 轴为机器人横向轴。机器人在 XOY 中的位姿可以用 (x,y,φ) 表示，其中 (x,y) 为全局位置坐标，φ 为位姿角。令 v_x 表示机器人的纵向线速度，v_y 表示横向线速度，ω 表示角速度，则机器人整体的位姿变换表示为

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\varphi}\end{bmatrix}=\begin{bmatrix}\cos\varphi & -\sin\varphi & 0\\ \sin\varphi & \cos\varphi & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}v_x\\ v_y\\ \omega\end{bmatrix} \tag{1}$$

令 l_x 表示纵向轮间距，l_y 表示横向轮间距，r_i 表示轮子半径，θ_i 表示小滚轮的偏置角。假设第 i 个轮子的角速度为 ω_i，则其贡献的速度可以分解为纵向速度 V_{ix} 和横向速度 V_{iy}。对于每个 Mecanum 轮，小滚轮贡献的速度可以分解为旋转产生的滚动速度 V_{im} 和摩擦产生的速度 V_{ir} 两个分量。可得

$$\begin{cases}V_{ix}=V_{ir}\sin\theta_i+V_{im}\cos\theta_i\\ V_{iy}=V_{ir}\cos\theta_i-V_{im}\sin\theta_i\end{cases} \tag{2}$$

假设滚轮在轴线方向无滑动（$V_{im}=0$），各轮子的速度分量与机器人本体速度关联，即

$$\begin{cases}V_{1x}=v_x-\omega l_y,\ V_{1y}=v_y+\omega l_x\\ V_{2x}=v_x+\omega l_y,\ V_{2y}=v_y+\omega l_x\\ V_{3x}=v_x-\omega l_y,\ V_{3y}=v_y-\omega l_x\\ V_{4x}=v_x+\omega l_y,\ V_{4y}=v_y-\omega l_x\end{cases} \tag{3}$$

每个轮子的角速度可表示为

$$\omega_i=\frac{1}{r}(v_x\cos\theta_i+v_y\sin\theta_i+\omega(l_x\sin\theta_i-l_y\cos\theta_i)) \tag{4}$$

对于标准四轮 Mecanum 机器人（$|\theta_i|=\pm\pi/4$ rad），运动学矩阵为

$$\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\\ \omega_4\end{bmatrix}=\frac{1}{r}\begin{bmatrix}1 & -1 & -l_x-l_y\\ 1 & 1 & l_x+l_y\\ 1 & 1 & -l_x-l_y\\ 1 & -1 & l_x+l_y\end{bmatrix}\begin{bmatrix}v_x\\ v_y\\ \omega\end{bmatrix} \tag{5}$$

逆矩阵推导为

$$\begin{bmatrix}v_x\\ v_y\\ \omega\end{bmatrix}=\frac{r}{4}\begin{bmatrix}1 & 1 & 1 & 1\\ -1 & 1 & 1 & -1\\ \dfrac{-1}{l_x+l_y} & \dfrac{1}{l_x+l_y} & \dfrac{-1}{l_x+l_y} & \dfrac{1}{l_x+l_y}\end{bmatrix}\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\\ \omega_4\end{bmatrix} \tag{6}$$

最后，联立式(1)和式(6)，得到机器人在世界坐标系下的运动学模型为

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\varphi}\end{bmatrix}=\frac{r}{4}\begin{bmatrix}\cos\varphi+\sin\varphi & \cos\varphi-\sin\varphi & \cos\varphi-\sin\varphi & \cos\varphi+\sin\varphi\\ \sin\varphi-\cos\varphi & \cos\varphi+\sin\varphi & \cos\varphi+\sin\varphi & \sin\varphi-\cos\varphi\\ \dfrac{-1}{l_x+l_y} & \dfrac{1}{l_x+l_y} & \dfrac{-1}{l_x+l_y} & \dfrac{1}{l_x+l_y}\end{bmatrix}\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\\ \omega_4\end{bmatrix} \tag{7}$$

令 $\boldsymbol{q}=[x,y,\varphi]^{\mathrm{T}}$ 表示机器人系统状态，$\boldsymbol{u}=[\omega_1,\omega_2,\omega_3,\omega_4]^{\mathrm{T}}$ 表示控制输入，状态空间方程简写为

$$\boldsymbol{q}_{k+1}=\boldsymbol{q}_k+\dot{\boldsymbol{q}}_k=f(\boldsymbol{q}_k,\boldsymbol{u}_k) \tag{8}$$

1.2 轨迹跟踪预测控制器设计

轨迹跟踪是机器人沿预定参考轨迹 $\boldsymbol{q}^{\mathrm{ref}}$ 运动。本文中参考轨迹 $\boldsymbol{q}^{\mathrm{ref}}$ 由路径规划算法生成。系统实时计算本体位姿 $\boldsymbol{q}$ 与参考路径 $\boldsymbol{q}^{\mathrm{ref}}$ 之间的偏差，并调整控制策略以实现精确跟踪。

在轨迹跟踪控制中，主要目标是最小化状态偏差并保持控制稳定。为实现这一目标，MPC 控制器通过在线优化控制输入 $\boldsymbol{u}$ 来驱动机器人沿着参考轨迹 $\boldsymbol{q}^{\mathrm{ref}}$ 前进，其中的优化目标函数 J 可表示为

$$J(\boldsymbol{q}_k,\boldsymbol{U}_k)=\sum_{i=1}^{N}[\boldsymbol{q}_{k+i}^{\mathrm{ref}}-\hat{\boldsymbol{q}}_{k+i|k}]^{\mathrm{T}}\boldsymbol{W}_q[\boldsymbol{q}_{k+i}^{\mathrm{ref}}-\hat{\boldsymbol{q}}_{k+i|k}]+\sum_{i=0}^{N-1}\Delta\boldsymbol{u}_{k+i|k}^{\mathrm{T}}\boldsymbol{W}_u\Delta\boldsymbol{u}_{k+i|k} \tag{9}$$

式中：N 为预测时域，$k+i|k$ 为基于 k 时刻对 $k+i$ 时刻的预测，输入增量 $\Delta\boldsymbol{u}_{k+i|k}=\boldsymbol{u}_{k+i|k}-\boldsymbol{u}_{k+i-1|k}$。目标函数包含两个关键部分，状态误差项衡量机器人本体与参考轨迹之间的误差，要求尽可能接近参考轨迹以保证精确跟踪，其权重矩阵为 $\boldsymbol{W}_q$；控制惩罚项用于惩罚控制输入 $\boldsymbol{u}$ 的剧烈变化，确保控制策略平稳，其权重矩阵为 $\boldsymbol{W}_u$。

考虑机器人性能约束，得到优化问题：

$$\boldsymbol{U}_k^* = \underset{u\in\boldsymbol{\mathcal{U}}}{\operatorname{argmin}}\ J(\boldsymbol{q}_k, \boldsymbol{U}_k)$$

$$\text{s.t.}\begin{cases}\hat{\boldsymbol{q}}_{k+i+1|k} = f(\hat{\boldsymbol{q}}_{k+i|k}, \boldsymbol{u}_{k+i|k})\\ \hat{\boldsymbol{q}}_{k|k} = \boldsymbol{q}_k, \Delta\boldsymbol{u}_{k|k} = \boldsymbol{u}_{k|k} - \boldsymbol{u}_{k-1}\\ \hat{\boldsymbol{q}} \in \boldsymbol{\mathcal{Q}}, \boldsymbol{u} \in \boldsymbol{\mathcal{U}}, \Delta\boldsymbol{u} \in \boldsymbol{\mathcal{U}}_\Delta\end{cases} \tag{10}$$

式中，$\boldsymbol{\mathcal{Q}}$，$\boldsymbol{\mathcal{U}}$，$\boldsymbol{\mathcal{U}}_\Delta$ 分别为对应各变量的约束范围。

求解优化问题得到最优控制序列 $\boldsymbol{U}_k^* = [\boldsymbol{u}_{k|k}^*, \boldsymbol{u}_{k+1|k}^*, \cdots, \boldsymbol{u}_{k+N-1|k}^*]^{\mathrm{T}}$。在每个控制时刻 k，MPC 选择最优控制序列中的第 1 个控制输入应用于机器人：$\boldsymbol{u}_k = \boldsymbol{u}_{k|k}^*$。随后，系统根据实际执行结果更新状态，并重新计算目标函数，继续优化控制输入。该过程持续进行，直到任务完成或达到预定停止条件。

在 MPC 控制器中，控制性能精度主要受 3 类参数影响：1）状态偏差权重矩阵 $\boldsymbol{W}_q = \mathrm{diag}(w_x, w_y, w_\varphi)$ 中的参数直接决定各状态维度的跟踪误差惩罚强度，增大权重可提高对应状态跟踪精度，但过度强化会引发控制输入振荡。2）控制输入变化权重矩阵 $\boldsymbol{W}_u = \mathrm{diag}(w_{\Delta\omega_1}, w_{\Delta\omega_2}, w_{\Delta\omega_3}, w_{\Delta\omega_4})$ 中的参数则分别约束 4 个 Mecanum 轮角速度的增量惩罚，其权重与系统平滑性呈正相关关系，高权重抑制超调但可能延长调节时间。3）预测时域 N 作为滚动优化的预测时域，需平衡长期稳定性与计算效率，较小 N 提升响应速度但忽略误差累积效应，较大 N 则反之。这些参数的动态协同优化是提升性能的核心。相较于固定参数或启发式规则，动态参数整定机制显著增强了控制器在轨迹突变、外部扰动等复杂场景的鲁棒性。

2　基于 Actor-Critic 强化学习的 MPC 参数整定

在 MPC 控制器中，控制性能与控制参数密切相关，但二者之间缺乏显式解析关系，使得参数整定困难。本文提出一种融合 Actor-Critic 强化学习框架的 MPC 参数动态整定方法。相较于传统端到端策略网络直接建立“状态-控制参数”映射的调参方式，本方法通过动态交互与自主优化，实现控制性能的全局最优与复杂环境的高效适应。

2.1　Actor-Critic 网络构建

Actor-Critic 网络通过策略网络（Actor）生成参数优化策略，价值网络（Critic）评估策略的长期收益，融合策略优化的自主性与值函数评估的稳定性，实现动态环境下 MPC 参数的高效整定与全局性能均衡，其架构如图 2 所示。

2.1.1　Actor 网络

作为策略网络，Actor 网络的核心功能是通过实时感知系统状态动态调整 MPC 控制器的优化参数。MPC 系统的目标函数的完整映射关系为

$$\boldsymbol{S}, \boldsymbol{W}, N \rightarrow \min_{u\in\boldsymbol{\mathcal{U}}} J \tag{11}$$

式中：该映射关系中，$\boldsymbol{S}$ 表征实时采集的系统观测变量集，$\boldsymbol{W}$ 表征待优化的 MPC 目标函数权重向量。

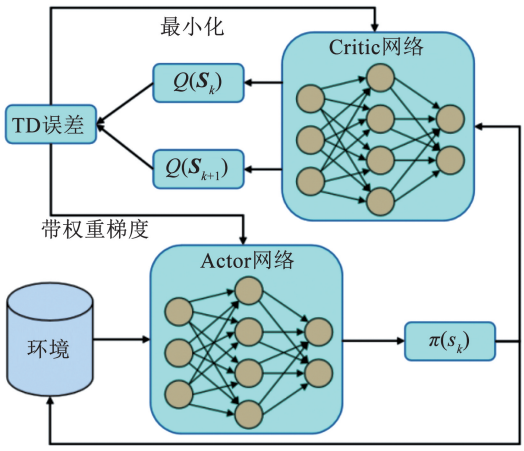


图 2　Actor-Critic 网络架构

Fig. 2　Actor-Critic network architecture

Actor 网络设计采用多维状态观测向量作为输入：

$$\boldsymbol{S}_k = \left[\underbrace{\boldsymbol{q}_k}_{\text{当前状态}}^{\mathbf{R}^3}, \underbrace{\boldsymbol{u}_{k-1}}_{\text{上一时刻的控制输入}}^{\mathbf{R}^4}, \underbrace{\boldsymbol{q}_{k:k+N}^{\mathrm{ref}}}_{\text{参考路径点}}^{\mathbf{R}^{3(N+1)}}\right]^{\mathrm{T}} \in \mathbf{R}^{3N+10} \tag{12}$$

式中：$\boldsymbol{q}_k$、$\boldsymbol{u}_{k-1}$ 为共同表征系统实时运行状态，$\boldsymbol{q}_{k:k+N}^{\mathrm{ref}}$ 为序列定义有限时域内的轨迹跟踪目标。特别需要注意的是，若将预测时域 N 作为网络输入，将导致输入层神经元数量动态变化，违反神经网络的结构固定性要求，并且输出权重参数维度与 N 产生耦合，破坏控制律的稳定性。因此将预测时域 N 设定为超参数。

在参数生成机制方面，Actor 网络通过可微策略函数实现权重向量的端到端映射：

$$\boldsymbol{W}_k = \pi(\boldsymbol{S}_k; \boldsymbol{\theta}_\pi) = \left(\underbrace{w_x, w_y, w_\varphi}_{\text{状态偏差的权重}}, \underbrace{w_{\Delta\omega_1}, w_{\Delta\omega_2}, w_{\Delta\omega_3}, w_{\Delta\omega_4}}_{\text{控制输入变化的权重}}\right) \in \mathbf{R}^7 \tag{13}$$

式中：$\pi(\cdot)$ 为策略函数，$\boldsymbol{\theta}_\pi$ 为 Actor 网络的可训练参数集。

Actor 网络采用 3 层全连接架构（256-128-64 神经元配置），其设计基于高维状态特征提取与实时控制需求（$N=10$）的平衡原则：输入状态向量 $\boldsymbol{S}_k$ 维度为 40，输出为 7 维权重向量。256 神经元层提供充足容量以捕获动态环境特性，且可以避免过参数化导致的过拟合风险；128 神经元层实现特征精炼，64 神经元层生成决策抽象特征，这种递减结构符合深度特征提取的“蒸馏”范式。隐藏层采用 LeakyReLU 激活函数，缓解梯度消失问题；输出层通过 Softmax 函数实现权重归一化，保证参数权重满

足非负约束条件。

2.1.2 Critic 网络

Critic 作为价值评估器,其功能在于量化特定状态$\boldsymbol{S}_k$和参数配置$\pi(\boldsymbol{S}_k;\boldsymbol{\theta}_\pi)$联合作用下的长期预期收益。网络输入采用状态 – 策略拼接向量,输出为标量价值估计值$Q(\boldsymbol{S}_k,\pi(\boldsymbol{S}_k;\boldsymbol{\theta}_\pi);\boldsymbol{\theta}_Q)$,其中$\boldsymbol{\theta}_Q$为 Critic 网络的可训练参数。

Critic 网络采用特征共享机制:前两层(256-128 神经元配置)与 Actor 网络共享权值参数,确保状态表征一致性,减少参数量以提升训练效率;第 3 层独立分支通过线性激活函数输出无约束价值估计,专注学习状态 – 策略的价值映射关系。这种轻量化设计在保证评估准确性的前提下,显著降低计算负载。

2.1.3 奖励函数

在本文中,Actor 网络的输出$\pi(\boldsymbol{S}_k;\boldsymbol{\theta}_\pi)$是 MPC 目标函数 J 的一部分,直接以 J 作为奖励函数会引发“奖励破解”问题。因此设计独立的动态性能指标作为强化学习的奖励函数。奖励函数r_k设计为

$$r_k = -\lambda_\sigma\sigma - \lambda_t t_s - \sum_{i=1}^{N}\gamma^i\left|\hat{\boldsymbol{q}}_{k+i|k} - \boldsymbol{q}_{k+i}^{\mathrm{ref}}\right| \tag{14}$$

式中:预测超调量σ和预测调节时间t_s分别为控制领域普遍采用的性能指标,λ_σ、λ_t为权重系数,γ为缩放因子。相较于现有强化学习调参方法中直接使用 MPC 目标函数作为奖励函数的设计,本文融合预测超调量与调节时间指标,通过权重实现轨迹跟踪中瞬态精度与长期性能的协同优化。具体的,σ、t_s设计为

$$\begin{cases}\sigma = \max\{\underbrace{\max_i[\hat{\boldsymbol{q}}_{k+i|k} - \boldsymbol{q}_{k+i}^{\mathrm{ref}}]_+}_{\text{正向超调}}, \underbrace{\max_i[\boldsymbol{q}_{k+i}^{\mathrm{ref}} - \hat{\boldsymbol{q}}_{k+i|k}]_+}_{\text{负向超调}}\}\\ t_s = \min\{T\in\{1,\cdots,N\}\mid\forall t\geqslant T,\ \|\hat{\boldsymbol{q}}_{k+i|k} - \boldsymbol{q}_{k+i}^{\mathrm{ref}}\|\leqslant\epsilon_s\}\end{cases} \tag{15}$$

式中:$[x]_+ = \max\{0,x\}$表示取正部,仅统计超出参考值的正向偏差;ϵ_s为允许误差带的宽度。

2.1.4 参数更新

基于双时间尺度更新机制,Actor-Critic 框架通过策略评估与策略改进的迭代实现协同优化。

1)Critic 网络参数更新。采用时序差分误差最小化准则,定义贝尔曼误差损失函数为

$$L_C(\boldsymbol{\theta}_Q) = \|Q(\boldsymbol{S}_k,\pi(\boldsymbol{S}_k;\boldsymbol{\theta}_\pi);\boldsymbol{\theta}_Q) - y_k^Q\|^2 \tag{16}$$

目标值y_k^Q通过滞后目标网络计算,$\boldsymbol{\theta}_\pi^-$、$\boldsymbol{\theta}_Q^-$表示目标网络的滞后参数,即

$$y_k^Q = r_k + \gamma Q(\boldsymbol{S}_{k+1},\pi(\boldsymbol{S}_{k+1};\boldsymbol{\theta}_\pi^-);\boldsymbol{\theta}_Q^-) \tag{17}$$

参数更新采用梯度下降法,学习率η_{RL}为

$$\boldsymbol{\theta}_Q \leftarrow \boldsymbol{\theta}_Q + \eta_{\mathrm{RL}}\nabla_{\boldsymbol{\theta}_Q}L_C(\boldsymbol{\theta}_Q) \tag{18}$$

2)Actor 网络参数更新。依据策略梯度定理,通过价值函数最大化进行策略优化,即

$$\begin{cases}L_A = -Q(\boldsymbol{S}_k,\pi(\boldsymbol{S}_k;\boldsymbol{\theta}_\pi);\boldsymbol{\theta}_Q)\\ \boldsymbol{\theta}_\pi \leftarrow \boldsymbol{\theta}_\pi + \eta_{\mathrm{RL}}\nabla_{\boldsymbol{\theta}_\pi}L_A(\boldsymbol{\theta}_\pi)\end{cases} \tag{19}$$

3)目标网络更新。采用软更新策略确保训练稳定性,更新系数$\tau\in(0,1)$为

$$\begin{cases}\boldsymbol{\theta}_Q^- \leftarrow \tau\boldsymbol{\theta}_Q + (1-\tau)\boldsymbol{\theta}_Q^-\\ \boldsymbol{\theta}_\pi^- \leftarrow \tau\boldsymbol{\theta}_\pi + (1-\tau)\boldsymbol{\theta}_\pi^-\end{cases} \tag{20}$$

2.2 离线训练

为构建鲁棒的初始策略并提升样本效率,本文采用两阶段离线训练流程,结合专家演示预训练与价值函数引导的强化学习预优化。具体流程如下:

1)在初始阶段,通过专家 MPC 控制器生成高质量的状态 – 参数数据集 D。首先,利用专家控制器在多样化参考轨迹(如直线、曲线、突变轨迹等)上运行,实时采集系统状态$\boldsymbol{S}_k$、专家 MPC 参数$\boldsymbol{W}_k$,以及对应的控制性能指标(σ,t_s)。其次,基于性能指标对原始数据进行筛选:仅保留满足$(\sigma<\sigma_{\mathrm{th}})\cap(t_s<N)$的样本,以剔除明显的低质量参数配置。最后,构建专家数据集$D=\{(\boldsymbol{S}_k,\boldsymbol{W}_k)\}_{k=1}^n$。

基于筛选后的数据集,采用监督学习框架训练 Actor 网络。定义均方误差损失函数为

$$L_{\mathrm{SL}} = \frac{1}{n}\sum_{k=1}^{n}\|\pi(\boldsymbol{S}_k;\boldsymbol{\theta}_\pi) - \boldsymbol{W}_k\|^2 \tag{21}$$

训练过程中,Critic 网络保持冻结状态,仅更新 Actor 网络的参数$\boldsymbol{\theta}_\pi$。网络采用 Adam 优化器,学习率设置为10^{-3},并引入早停机制,验证集损失连续 10 轮无改善时终止。

2)在监督学习的基础上,进一步引入 Critic 网络的动态评估能力,实现策略从模仿到自主优化的过渡。此阶段复用专家数据集中的状态转移序列$(\boldsymbol{S}_k,\boldsymbol{S}_{k+1})$,并通过离线强化学习框架重新构建马尔可夫决策过程。算法 1 给出了 Actor-Critic 网络的训练流程。

算法 1 Actor-Critic 网络训练

输入:初始网络参数$\boldsymbol{\theta}_\pi$、$\boldsymbol{\theta}_Q$,缩放因子γ,学习率η_{RL},软更新率τ
输出:优化后的网络参数$\boldsymbol{\theta}_\pi^*$,$\boldsymbol{\theta}_Q^*$

```
初始化  目标网络参数,经验回放池
循环训练(回合数 = 1 ~ M):
    获取环境初始状态S_0
    时间步 = 0 ~ T:
        Actor 网络生成控制参数(13)
        执行控制动作,观测奖励 r_k 和下一状态S_{k+1}
        将转移样本(S_k, W_k, r_k, S_{k+1})存入经验池 D
        网络更新(当|D| ≥批次大小时):
            从 D 中随机采样批次{(S_k, W_k, r_k, S_{k+1})}
            计算目标值(17)
            Critic 网络参数更新(18)
            Actor 网络参数更新(19)
            目标网络软更新(20)
```

2.3　基于事件触发的自适应 MPC 控制器设计

为实现动态环境下的高效参数整定，本文提出一种融合事件触发机制的自适应 MPC 控制器。该控制器通过事件触发机制激活参数设定策略。

2.3.1　事件触发机制

定义触发函数 $\Gamma(k)$ 为跟踪误差与控制增量联合判据为

$$\Gamma(k)=\begin{cases}1, & \|\boldsymbol{e}_k\|>e_{\text{th}}\\ 0, & \|\boldsymbol{e}_k\|\leqslant e_{\text{th}}\end{cases} \tag{22}$$

式中：$\boldsymbol{e}_k=[e_x,e_y,e_\varphi]^{\mathrm{T}}=[x_k-x_k^{\text{ref}},y_k-y_k^{\text{ref}},\varphi_k-\varphi_k^{\text{ref}}]^{\mathrm{T}}$ 为位姿跟踪误差，e_{th} 为误差阈值。当 $\Gamma(k)=1$ 时，激活强化学习参数整定模块；否则沿用上一时刻的控制参数 $\boldsymbol{W}_{k-1}$，减少冗余计算。因此，k 时刻的控制参数为

$$\boldsymbol{W}_k=\Gamma(k)\cdot\pi(\boldsymbol{S}_k)+(1-\Gamma(k))\cdot\boldsymbol{W}_{k-1} \tag{23}$$

所设计的事件触发机制深度耦合 Actor-Critic 框架形成动态休眠－激活策略：当 $\|\boldsymbol{e}_k\|\leqslant e_{\text{th}}$ 时参数冻结，仅误差 $\|\boldsymbol{e}_k\|>e_{\text{th}}$ 时更新。该策略突破了强化学习的实时性瓶颈。

2.3.2　控制器架构

控制器如图 3 所示，主要包含 3 个核心模块：事件检测模块、参数生成模块，以及 MPC 求解器。

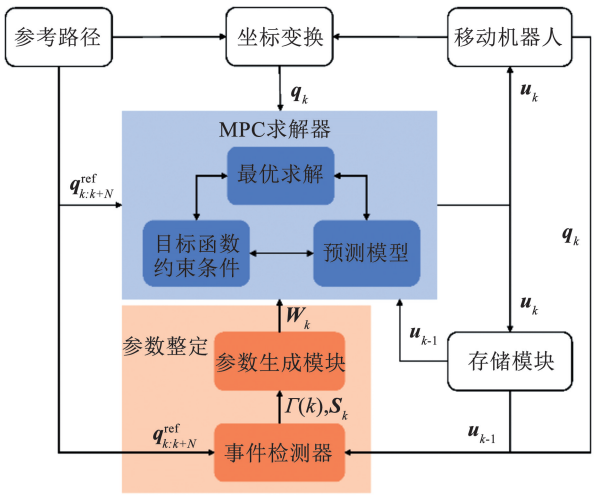


图 3　自适应 MPC 控制器架构

Fig. 3　Adaptive MPC controller architecture

算法 2 给出了自适应 MPC 控制器的实现流程。

2.4　稳定性与收敛性分析

为验证闭环系统的鲁棒性，基于 Lyapunov 稳定性理论与强化学习理论，分析系统的稳定性和收敛性。

2.4.1　系统稳定性分析

定义跟踪误差向量 $\boldsymbol{e}_k=\boldsymbol{q}_k-\boldsymbol{q}_k^{\text{ref}}$，采用 Lyapunov 第二方法，构造对称正定矩阵 $\boldsymbol{P}=\boldsymbol{P}^{\mathrm{T}}>0$，并定义候选 Lyapunov 函数，即

$$V_k=\boldsymbol{e}_k^{\mathrm{T}}\boldsymbol{P}\boldsymbol{e}_k \tag{24}$$

算法 2　事件触发自适应 MPC 控制

输入：参考轨迹 $\boldsymbol{q}_k^{\text{ref}}$，触发阈值 e_{th}
输出：控制序列 $\boldsymbol{u}_k$
1. 初始化 $\boldsymbol{W}_k\leftarrow$默认权重
2. 实时控制循环（时间步 $=0\sim T$）：
3.　　读取当前机器人位姿 $\boldsymbol{q}_k$
4.　　计算跟踪误差范数 $\|\boldsymbol{e}_k\|$
5.　　if　$\|\boldsymbol{e}_k\|>e_{\text{th}}$ then
6.　　通过 Actor 网络生成参数 $\boldsymbol{W}_k=\pi(\boldsymbol{S}_k;\boldsymbol{\theta}_\pi)$
7.　　else
8.　　参数继承 $\boldsymbol{W}_k\leftarrow\boldsymbol{W}_{k-1}$
9.　　求解 MPC 优化问题 $\boldsymbol{U}_k^*=\arg\min_{u\in\boldsymbol{\mathcal{U}}}J(\boldsymbol{q}_k,\boldsymbol{U}_k)$
10.　　实施控制量 $\boldsymbol{u}_k=\boldsymbol{u}_{k|k}^*$

满足 $\lambda_{\min}(\boldsymbol{P})\|\boldsymbol{e}_k\|^2\leqslant V_k\leqslant\lambda_{\max}(\boldsymbol{P})\|\boldsymbol{e}_k\|^2$，其中 $\lambda_{\min}(\boldsymbol{P})$、$\lambda_{\max}(\boldsymbol{P})$ 分别为 $\boldsymbol{P}$ 最小和最大特征值。

分析 Lyapunov 函数差分可得

$$\Delta V_k=V_{k+1}-V_k=\boldsymbol{e}_{k+1}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{e}_{k+1}-\boldsymbol{e}_k^{\mathrm{T}}\boldsymbol{P}\boldsymbol{e}_k \tag{25}$$

代入系统动力学方程和控制律，可得

$$\Delta V_k=[f(\boldsymbol{q}_k,\boldsymbol{u}_k)-\boldsymbol{q}_{k+1}^{\text{ref}}]^{\mathrm{T}}\boldsymbol{P}[f(\boldsymbol{q}_k,\boldsymbol{u}_k)-\boldsymbol{q}_{k+1}^{\text{ref}}]-[\boldsymbol{q}_k-\boldsymbol{q}_k^{\text{ref}}]^{\mathrm{T}}\boldsymbol{P}[\boldsymbol{q}_k-\boldsymbol{q}_k^{\text{ref}}] \tag{26}$$

系统稳定性依赖于以下两个关键假设。

假设 1（MPC 优化可行性）　在训练充分时，对于任意 $\boldsymbol{S}_k$ 和 $\boldsymbol{W}_k$，存在最优控制序列 $\boldsymbol{U}_k^*$ 满足：

$$J(\boldsymbol{q}_k,\boldsymbol{U}_k^*;\boldsymbol{W}_k)\leqslant J(\boldsymbol{q}_k,\boldsymbol{U}_k;\boldsymbol{W}_k) \tag{27}$$

同时 Actor 策略生成的参数变化满足性能非退化条件为

$$J(\boldsymbol{q}_k,\boldsymbol{U}_k^*;\boldsymbol{W}_k)\leqslant J(\boldsymbol{q}_k,\boldsymbol{U}_{k-1}^*;\boldsymbol{W}_k)+\delta_k \tag{28}$$

式中 $\delta_k\geqslant 0$ 为有界容差项。

假设 2（系统动力学约束）　系统运动学模型（8）满足 Lipschitz 连续特性，可得到状态误差传播界限为

$$\boldsymbol{q}_{k+1}-\boldsymbol{q}_{k+1}^{\text{ref}}\leqslant L_f\|\boldsymbol{q}_k-\boldsymbol{q}_k^{\text{ref}}\|+L_u\|\Delta\boldsymbol{u}_k\| \tag{29}$$

式中 L_f、L_u 为模型 Lipschitz 常数。结合 MPC 求解施加的控制输入约束 $\Delta u\in\boldsymbol{\mathcal{U}}_\Delta$（隐含 $\|\Delta\boldsymbol{u}_k\|\leqslant\kappa\|\boldsymbol{e}_k\|$），最终得到：

$$\|\boldsymbol{q}_{k+1}-\boldsymbol{q}_{k+1}^{\text{ref}}\|\leqslant\underbrace{(L_f+\kappa L_u)}_{\eta}\|\boldsymbol{e}_k\| \tag{30}$$

式中 η 为保证误差收缩的核心系数。

Actor 网络输出层采用 Softmax 激活函数，确保权重有界归一，固有满足 $\|\boldsymbol{W}_k-\boldsymbol{W}_{k-1}\|_\infty\leqslant 1$。需要注意的是，所设计的事件触发机制显著限制了参数更新的频率和幅度：当 $\Gamma(k)=0$ 时，参数无变化；当 $\Gamma(k)=1$ 时，$\|\boldsymbol{W}_k-\boldsymbol{W}_{k-1}\|\leqslant\Delta\boldsymbol{W}_{\max}$。Actor 网络自身的 Lipschitz 连续性和机器人运动性能约束共同限制了单步参数变化量为

$$\Delta\boldsymbol{W}_{\max}\leqslant L_\pi\|\boldsymbol{S}_k-\boldsymbol{S}_{k-1}\|\leqslant L_\pi\cdot\Delta\boldsymbol{S}_{\max} \tag{31}$$

式中 L_π 为 Actor 网络的 Lipschitz 常数。

误差传播界限和参数变化影响回代至 ΔV_k，

可得

$$\Delta V_k \leqslant (\eta^2 \lambda_{\max}(\boldsymbol{P}) - \lambda_{\min}(\boldsymbol{P})) \| \boldsymbol{e}_k \|^2 + \xi(\Delta \boldsymbol{W}_k) \tag{32}$$

式中,$\xi(\Delta \boldsymbol{W}_k)$ 项表征参数变化扰动,其幅值受限于:

$$|\xi(\Delta \boldsymbol{W}_k)| \leqslant L_f L_g \Delta \boldsymbol{W}_{\max} \| \boldsymbol{e}_k \| \tag{33}$$

式中 L_g 为控制器映射 $\boldsymbol{u}_k = g(\boldsymbol{W}_{k-1}, \boldsymbol{e}_k)$ 的 Lipschitz 常数。

令 $\beta = \lambda_{\min}(\boldsymbol{P}) - \eta^2 \lambda_{\max}(\boldsymbol{P})$,应用 Young 不等式:

$$L_f L_g \Delta \boldsymbol{W}_{\max} \| \boldsymbol{e}_k \| \leqslant \frac{(L_f L_g \Delta \boldsymbol{W}_{\max})^2}{4\epsilon} + \epsilon \| \boldsymbol{e}_k \|^2 \tag{34}$$

最终的稳定性条件,即

$$\Delta V_k \leqslant -(\beta - \epsilon) \| \boldsymbol{e}_k \|^2 + \frac{(L_f L_g \Delta \boldsymbol{W}_{\max})^2}{4\epsilon} < 0 \tag{35}$$

当 $\| \boldsymbol{e}_k \| > e_{\text{th}} = \dfrac{L_f L_g \Delta \boldsymbol{W}_{\max}}{2\sqrt{\epsilon(\beta - \epsilon)}}$ 时,$\Delta V_k < 0$,系统会快速收敛至 $\| \boldsymbol{e}_k \| \leqslant e_{\text{th}}$,故系统是稳定的。

2.4.2 系统收敛性分析

在满足稳定性条件 $\| \boldsymbol{e}_k \| \leqslant e_{\text{th}}$ 的前提下,系统收敛性可进一步进行论证。

当 $\| \boldsymbol{e}_k \| \leqslant e_{\text{th}}$ 时,事件触发机制进入休眠状态,即 $\boldsymbol{\Gamma}(k) = 0$,系统维持 $\boldsymbol{W}_k = \boldsymbol{W}_{k-1}$ 的参数配置。此时闭环系统退化为固定参数的经典 MPC 控制器,其收敛性满足:

$$\lim_{k \to \infty} \| \boldsymbol{e}_k \| \to 0 \tag{36}$$

这保证在参数冻结期间,系统能自主驱动跟踪误差渐进收敛至零域内。

3 实验验证

3.1 实验设计

本文应用真实的全向移动机器人进行轨迹跟踪实验,如图 4 所示。型号为 SCOUT MINI,采用 Mecanum 四轮驱动全向移动底盘。轮距 $l_x = 0.540$ m,轴距 $l_y = 0.475$ m,最大设定线速度 3 m/s,最大设定角速度 2.523 5 rad/s,控制周期 0.1 s。

设计 3 类实验场景(阶跃轨迹、Lemniscate 轨迹,以及复杂避障),对比传统固定参数 MPC(F-MPC)、基于规则的参数自适应 MPC(A-MPC)与本文方法(RL-MPC)的控制性能。其中,F-MPC 通过人工经验设定控制器参数,A-MPC 基于 $\boldsymbol{W}_k = \text{diag}(10 \cdot \tanh \| \boldsymbol{e}_k \|, 1)$ 的简单规则调整参数。所有实验在相同条件下重复 5 次,取多次实验的均值和标准差,以避免偶然性影响。实验超参数配置见表 1。

图 4 移动机器人系统

Fig. 4 Mobile robot system

表 1 实验参数配置

Tab. 1 Experimental parameter configuration

预测时域	学习率	收缩因子	超调量权重	调节时间权重	误差阈值
N	η	γ	λ_σ	λ_{t_s}	e_{th}
10	0.001	0.95	1	0.03	0.05

3.2 网络结构消融实验

为验证 Actor 网络结构(256-128-64)选择的合理性,设计了消融实验,对比不同网络结构方案。所有模型采用相同的预训练数据和在线学习策略,通过随机生成的 5 条满足硬件约束的连续轨迹进行测试。评估指标包括各状态偏差(e_x, e_y, e_φ)的平均均方误差(average mean squared error, AMSE)和 Actor 网络单次前向推理耗时。实验结果取 5 次运行均值,汇总于表 2。

表 2 Actor 网络结构消融实验结果

Tab. 2 Actor network structure ablation experiment results

网络结构	S_{AMSE}/m		S_{AMSE}/rad	推理耗时/ms
	e_x	e_y	e_φ	
256-128-64	1.77×10^{-3}	1.06×10^{-3}	6.46×10^{-3}	0.08
512-256-128	1.68×10^{-3}	1.16×10^{-3}	5.83×10^{-3}	0.25
128-64-32	1.95×10^{-3}	1.01×10^{-3}	6.62×10^{-3}	0.05
64-32-16	4.42×10^{-3}	3.23×10^{-3}	9.71×10^{-3}	0.02

分析表 2 结果可知:512-256-128 网络的推理耗时显著高于其他网络,其计算效率在更复杂场景或更高控制频率下可能成为瓶颈;64-32-16 网络的性能明显劣于其他网络,说明过小的容量无法有效学

习从高维状态空间到多变量控制参数这一复杂非线性映射关系。

综上所述，实验结果支持选择 256-128-64 结构的合理性，该结构实现了计算效率与实时控制性能的平衡。

3.3　阶跃轨迹跟踪实验

为验证所提出方法在追踪不连续轨迹时系统恢复稳定的能力，本文设计了一组阶跃轨迹跟踪实验。实验采用实测超调量与调节时间作为控制性能的评价指标；采用参数整定平均用时和单步求解平均耗时作为实时性的评价指标。

在阶跃轨迹跟踪实验中，设置机器人的初始位姿$\boldsymbol{q}_0=[0,0,0]^{\mathrm{T}}$。参考点在 X 轴上的位置随时间线性递增 $x^{\mathrm{ref}}(t)=t$，航向角恒定为 $\varphi^{\mathrm{ref}}(t)=0$，$Y$ 轴上的位置在时间 $t<10$ s 时保持 $y^{\mathrm{ref}}(t<10)=0$，当 $t\geqslant 10$ s 时突变为 $y^{\mathrm{ref}}(t\geqslant 10)=2$。

图 5 展示了其中 1 次实验中 3 种方法的跟踪轨迹对比，所有方法在 $t<10$ s 阶段均能稳定跟踪参考轨迹，但阶跃突变后一段时间内（$t\in[10,15]$ s）表现出显著差异，进入误差带后（$t>15$ s）重归稳定。由于阶跃突变只发生在 Y 轴，且所有方法的调节时间都小于 5 s，因此，实验重点关注 $t\in[10,15]$ s 时间窗内发生在 Y 轴上的动态响应特性，通过超调量、调节时间两个指标的综合代价 $\lambda_\sigma\sigma+\lambda_t t_s$ 评估控制性能。

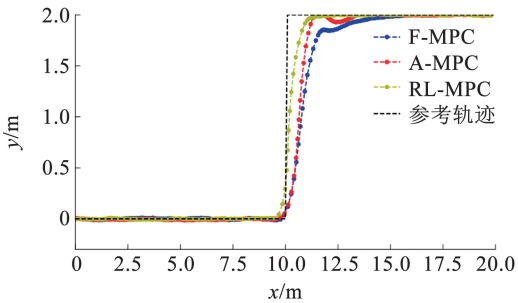
5 次实验结果的均值与标准差见表 3。本文提出的 RL-MPC 的调节时间平均为 1.48 s，远低于其他 2 个对比组，并且都没有产生超调现象。参数整定耗时相比于求解优化问题耗时占比不足 1%，证明了额外的参数整定过程几乎对计算效率没有影响。

表 3　阶跃轨迹跟踪实验对比结果

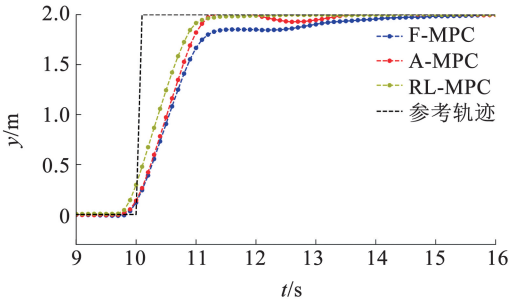
Tab. 3　Comparison results of step trajectory tracking experiments

评估指标	σ 均值/m	t_s 均值/s	t_s 标准差/s	参数整定平均耗时/ms	单步求解平均耗时/ms
F-MPC	0	5.14	0.10	0	34.98
A-MPC	0.031	3.55	0.18	0.02	37.74
RL-MPC	0	1.93	0.06	0.08	35.31

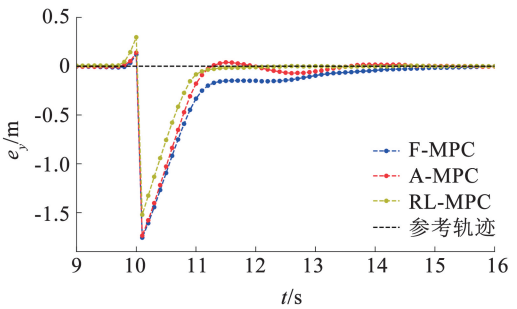
实验结果表明，基于强化学习的参数整定方法有效减少了传统方法在突变场景下的超调量与调节时间。



(a) 跟踪轨迹



(b)　Y轴状态量



(c) Y轴状态偏差

图 5　阶跃轨迹跟踪对比

Fig. 5　Step trajectory tracking comparison

3.4　Lemniscate 轨迹跟踪实验

在 Lemniscate 轨迹跟踪实验中，由于路径是连续且平滑，无法直接观测超调量与调节时间，因此通过比较 3 种方法在连续运动场景下的状态偏差评估其控制性能。机器人初始位姿为$\boldsymbol{q}_0=[0,0,0]^{\mathrm{T}}$，参考轨迹为长轴 3.0 m、短轴 1.5 m 的 Lemniscate 连续曲线。

$$\begin{cases} x^{\mathrm{ref}}(t)=\dfrac{3\cos(t)}{1+\sin^2(t)} \\ y^{\mathrm{ref}}(t)=\dfrac{3\cos(t)\sin(t)}{1+\sin^2(t)} \\ \varphi^{\mathrm{ref}}(t)=\arctan\left(\dfrac{y^{\mathrm{ref}}(t+\Delta t)-y^{\mathrm{ref}}(t)}{x^{\mathrm{ref}}(t+\Delta t)-x^{\mathrm{ref}}(t)}\right) \end{cases} \tag{37}$$

图 6 展示了其中 1 次实验中 3 种方法的跟踪轨迹对比，其中 RL-MPC 的跟踪轨迹与参考路径高度吻合。图 7 对比 3 种方法的状态偏差分量。

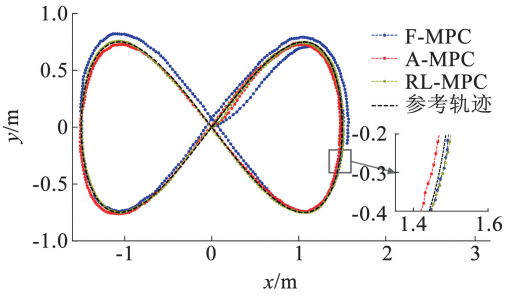
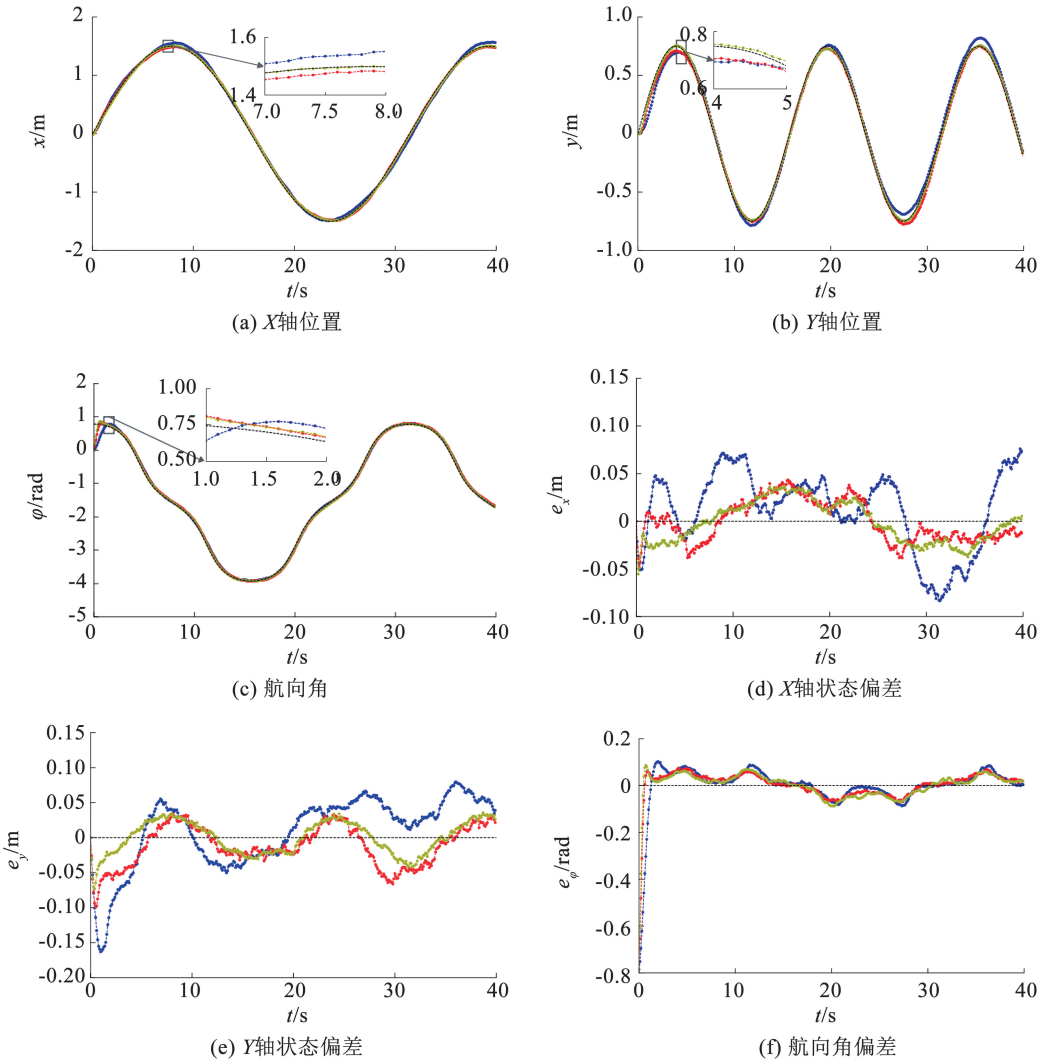


图 6 Lemniscate 跟踪轨迹

Fig. 6 Lemniscate tracking trajectory

5 次实验结果的均值和标准差见表 4。由表 4 可以看出，RL-MPC 各分量偏差的均值为 $1.160\ 8\times10^{-3}$，远低于 A-MPC 的 $-5.459\ 0\times10^{-3}$ 与 F-MPC 的 $1.373\ 9\times10^{-3}$；而 3 种方法的标准差则相差不大。实验结果证明了在连续轨迹跟踪中，RL-MPC 具有更优的稳定性和精度。



(a) X轴位置 (b) Y轴位置

(c) 航向角 (d) X轴状态偏差

(e) Y轴状态偏差 (f) 航向角偏差

图 7 Lemniscate 轨迹跟踪实验的分量对比

Fig. 7 Component comparison of Lemniscate trajectory tracking experiments

表 4 Lemniscate 轨迹跟踪实验结果

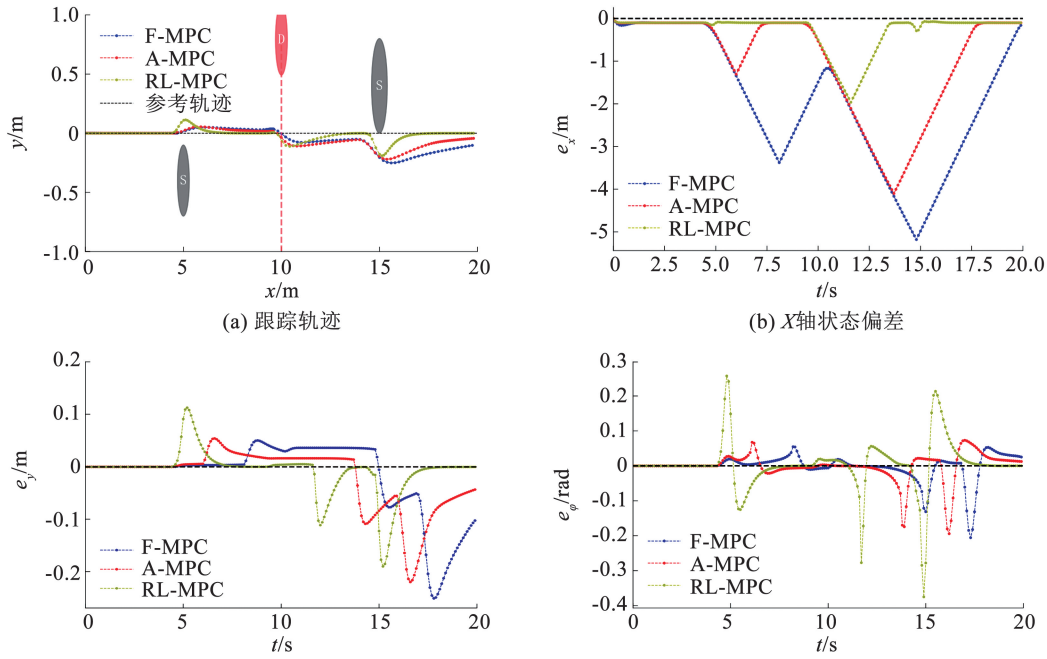
Tab. 4 Lemniscate trajectory tracking experiment results

评估指标	e_x 均值/m	e_x 标准差/m	e_y 均值/m	e_y 标准差/m	e_φ 均值/rad	e_φ 标准差/rad	参数整定平均耗时/ms	单步求解平均耗时/ms
F-MPC	$1.002\ 7\times10^{-2}$	$4.314\ 4\times10^{-2}$	$1.355\ 3\times10^{-2}$	$5.187\ 2\times10^{-2}$	$2.840\ 4\times10^{-3}$	$9.607\ 6\times10^{-2}$	0	46.12
A-MPC	$-8.051\ 1\times10^{-3}$	$1.932\ 7\times10^{-2}$	$-8.380\ 2\times10^{-3}$	$2.710\ 8\times10^{-2}$	$5.423\ 1\times10^{-5}$	$7.051\ 5\times10^{-2}$	0.02	41.06
RL-MPC	$2.225\ 2\times10^{-2}$	$2.218\ 4\times10^{-2}$	$1.160\ 8\times10^{-3}$	$2.272\ 0\times10^{-2}$	$5.186\ 2\times10^{-3}$	$4.370\ 0\times10^{-2}$	0.08	43.22

3.5　动态避障轨迹跟踪实验

为验证所提方法在复杂动态环境中的参数自适应能力，设计动态避障轨迹跟踪实验。实验场景设置：机器人初始位姿为$\boldsymbol{q}_0=[0,0,0]^{\mathrm{T}}$；参考轨迹为一条直线；设置位于(5.0，−0.4)和(15.0,0.4)的静态障碍物；另设有初始位置(10，−1)，以0.1 m/s速度沿 Y 轴正方向运动的动态障碍物；安全距离为0.1 m。通过比较状态偏差评估各方法的控制性能。

具体轨迹跟踪效果如图8所示，灰色和红色块分别表示静态障碍物和动态障碍物。RL-MPC展现出显著的避障自适应能力：在动态障碍物区域，RL-MPC平滑避让路径并快速回归至参考期望路径，F-MPC因参数固定导致紧急避让时产生较大超调，A-MPC虽能平滑避障但轨迹振荡明显。



(a) 跟踪轨迹　(b) X轴状态偏差　(c) Y轴状态偏差　(d) 航向角偏差

图8　动态避障轨迹跟踪实验对比

Fig. 8　Comparison of dynamic obstacle avoidance trajectory tracking experiments

此次实验中控制器求解时间如图9所示。

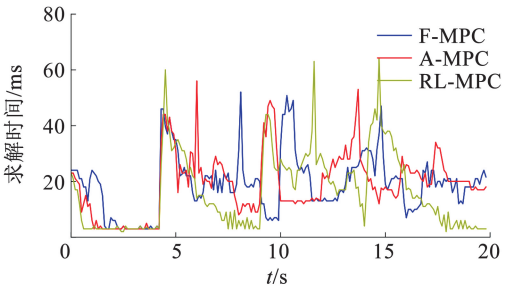


图9　动态避障轨迹跟踪实验控制器求解时间

Fig. 9　Solution time of controller in dynamic obstacle avoidance trajectory tracking experiment

5次实验结果的均值和标准差见表5。由表5可以看出：RL-MPC各分量偏差的均值为$2.809\,2\times10^{-1}$，低于A-MPC的$3.106\,4\times10^{-1}$与F-MPC的$3.592\,0\times10^{-1}$；而3种方法的标准差则相差不大。RL-MPC与障碍物的最近距离为0.103 m，大于安全距离以确保安全；而F-MPC则小于安全距离，为避障增加了风险。实验结果证明了RL-MPC在动态避障中的有效性。

表5　动态避障轨迹跟踪实验结果

Tab. 5　Dynamic obstacle avoidance trajectory tracking experiment results

评估指标	e_x 均值/m	e_x 标准差/m	e_y 均值/m	e_y 标准差/m	e_φ 均值/rad	e_φ 标准差/rad	与障碍物的最近距离/m	参数整定平均耗时/ms	单步求解平均耗时/ms
F-MPC	$-1.054\,5$	$1.236\,9$	$-2.291\,4\times10^{-2}$	$5.738\,9\times10^{-2}$	$-1.710\,4\times10^{-4}$	$3.934\,8\times10^{-2}$	9.70×10^{-2}	0	27.42
A-MPC	$-9.105\,4\times10^{-1}$	$1.138\,0$	$-2.122\,0\times10^{-2}$	$5.761\,6\times10^{-2}$	$-1.693\,4\times10^{-4}$	$4.168\,4\times10^{-2}$	1.21×10^{-1}	0.02	24.11
RL-MPC	$2.665\,9\times10^{-1}$	$4.109\,4\times10^{-1}$	$1.018\,6\times10^{-2}$	$4.427\,6\times10^{-2}$	$4.140\,5\times10^{-3}$	$7.331\,6\times10^{-2}$	1.03×10^{-1}	0.08	25.56

4 结 论

1)为提升动态环境下移动机器人轨迹跟踪预测控制的控制性能与适应性,设计了一种基于 Actor-Critic 网络的实时参数整定框架,通过构建联合状态误差与动态性能指标的奖励函数,实现 MPC 权重矩阵的动态协同优化。

2)提出了一种动态休眠 - 激活策略,将事件触发机制深度集成于强化学习框架。通过事件检测器实时监测跟踪误差和控制输入饱和状态,仅在误差超限时激活 Actor 网络参数更新,减少冗余计算负载。

3)实验表明,所提方法显著提升了轨迹跟踪性能:在阶跃轨迹跟踪中,超调量与调节时间降低约 70%;在 Lemniscate 连续轨迹跟踪中,状态偏差降低约 65%;在动态避障轨迹跟踪中,状态偏差降低约 30%。所提方法基本不影响计算效率:平均参数整定时间仅为 0.08 ms,不足求解优化平均耗时的 1%。

4)未来研究将进一步探索动态调整预训练策略,减少对专家数据集的依赖;开发轻量化 Critic 网络,进一步压缩推理耗时。

参考文献

[1]倪洪杰, 王宏霞, 俞立. 轮式移动机器人快速轨迹跟踪[J]. 哈尔滨工业大学学报, 2020, 52(10): 167
NI Hongjie, WANG Hongxia, YU Li. Fast trajectory tracking of wheeled mobile robots[J]. Journal of Harbin Institute of Technology, 2020, 52(10): 167. DOI: 10.11918/201911148

[2]罗欣, 丁晓军. 地面移动作业机器人运动规划与控制研究综述[J]. 哈尔滨工业大学学报, 2021, 53(1): 1
LUO Xin, DING Xiaojun. Research and prospective on motion planning and control of ground mobile manipulators[J]. Journal of Harbin Institute of Technology, 2021, 53(1): 1. DOI:10.11918/201910067

[3]刘清河, 王泽文, 赵立军. 自适应 LOS 制导结合 MPC 控制的车辆循迹优化[J]. 哈尔滨工业大学学报, 2022, 54(1): 96
LIU Qinghe, WANG Zewen, ZHAO Lijun. Vehicle tracking optimization based on adaptive LOS guidance and MPC control[J]. Journal of Harbin Institute of Technology, 2022, 54(1): 96. DOI: 10.11918/202012053

[4]贺宁, 范昭, 马凯. FDI 攻击下移动机器人弹性预测镇定控制研究[J]. 北京理工大学学报, 2024, 44(7): 722
HE Ning, FAN Zhao, MA Kai. Resilientpredictive stabilization control of mobile robot system under FDI attack[J]. Transactions of Beijing institute of Technology, 2024, 44(7): 722. DOI: 10.15918/j.tbit1001 - 0645.2023.171

[5]席裕庚, 李德伟, 林姝. 模型预测控制——现状与挑战[J]. 自动化学报, 2013, 39(3): 222
XI Yugeng, LI Dewei, LIN Shu. Model predictive control: status and challenges[J], Acta Automatica Sinica, 2013, 39(3): 222. DOI: 10.3724/SP.J.1004.2013.00222

[6]SUZUKI R, KAWAI F, NAKAZAWA C, et al. Parameter optimization of model predictive control by PSO[J]. Electrical Engineering in Japan, 2012, 178(1): 40. DOI: 10.1002/eej.21188

[7]GARRIGA J L, SOROUSH M, SOROUSH H M. On the effects of tunable parameters of model predictive control on the locations of closed-loop eigenvalues[J]. Industrial & Engineering Chemistry Research, 2010, 49(17): 7951. DOI: 10.1021/ie100030e

[8]IRA A S, MANZIE C, SHAMES I, et al. Tuning of multivariable model predictive controllers through expert bandit feedback[J]. International Journal of Control, 2021, 94(10): 2650. DOI: 10.1080/00207179.2020.1727959

[9]MOUMOUH H, LANGLOIS N, HADDAD M. A novel tuning approach for MPC parameters based on artificial neural network[C]//2019 IEEE 15th International Conference on Control and Automation (ICCA). Edinburgh: IEEE, 2019: 1638. DOI: 10.1109/ICCA.2019.8900026

[10]JARDINE P T, GIVIGI S, YOUSEFI S. Parameter tuning for prediction-based quadcopter trajectory planning using learning automata[J]. IFAC-PapersOnLine, 2017, 50(1): 2341. DOI: 10.1016/j.ifacol.2017.08.420

[11]LIU Cheng, WANG Daiyi, ZHANG Yuxi, et al. Model predictive control for path following and roll stabilization of marine vessels based on neurodynamic optimization[J]. Ocean Engineering, 2020, 217: 107524. DOI: 10.1016/j.oceaneng.2020.107524

[12]贺宁, 习坤, 高峰, 等. 基于 FCM-ELM-BBPS 的预测控制参数整定[J]. 湖南大学学报(自然科学版), 2023, 50(12): 168
HE Ning, XI Kun, GAO Feng, et al. Predictive control parameter tuning algorithm based on FCM-ELM-BBPS[J]. Journal of Hunan University (Natural Sciences), 2023, 50(12): 168. DOI: 10.16339/j.cnki.hdxbzkb.2023190

[13]LIU Hebing, SUN Jinhong, CHENG K W E. A two-layer model predictive path-tracking control with curvature adaptive method for high-speed autonomous driving[J]. IEEE Access, 2023, 11: 89228. DOI: 10.1109/ACCESS.2023.3306239

[14]DAI Changhua, ZONG Changfu, CHEN Guoying. Path tracking control based on model predictive control with adaptive preview characteristics and speed-assisted constraint[J]. IEEE Access, 2020, 8: 184697. DOI: 10.1109/ACCESS.2020.3029635

[15]LIN Xinyou, TANG Yunliang, ZHOU Binhao. Improved model predictive control path tracking strategy based an online updating algorithm with cosine similarity and a horizon factor[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(8): 12429. DOI: 10.1109/TITS.2021.3114060

(编辑 张 红)