基于 MPPI 的机械臂安全轨迹优化

Abstract:

Keyword:

1 介绍

机械臂最优轨迹规划[1]。零阶优化技术在机器人领域正变得日益流行，这得益于它们处理不可微函数和逃离局部最小值的能力，这些优势使它们在轨迹优化和策略优化中特别有用[2]。

基于采样的轨迹规划算法，MPPI 的规划。

[3]提出了一种新颖的基于路径积分的约束采样模型预测控制（CSMPC）方法。首先，采用结合策略采样投影和拉格朗日乘数法的分层优化来处理高精度操作任务中的等式硬约束。其次，将避碰和平稳运动建模为不等式软约束，通过碰撞检测和时间序列预测来确保动态交互的安全性和平稳性。

[4]为了在存在移动障碍物的杂乱环境中控制机器人，并在机器人的关节空间中生成可行且反应性强的无碰撞运动，提出了一种利用基于采样的 MPC 来调制关节空间动力系统的方法。具体而言，一个代表无约束的、朝向目标的期望关节空间运动的标称动力系统，会通过障碍物切向速度分量进行局部偏转，这些分量能引导机器人绕过障碍物并避开局部最小值。此类切向速度分量是由基于采样的 MPC 异步生成的滚动时域无碰撞路径构建而成的。值得注意的是，MPC 无需持续运行，仅在检测到局部最小值时才被激活。

[5]结合了无模型方法和基于模型方法的优势，使用学到的面向任务的潜在动态模型在短 horizon 内进行局部轨迹优化，并使用学到的终端价值函数来估计长期回报，这两者都是通过时序差分学习联合学习的，实现了更优的样本效率和渐近性能。

基于安全基础，引入控制障碍函数[6]:控制障碍函数（CBFs）是非线性系统安全控制领域的核心工具，其核心思想是通过构造一个连续可微的函数，将系统的安全约束转化为对控制输入的不等式约束，从而在保证系统完成控制任务的同时，始终避免进入不安全区域。

[7]直接在损失函数中增加控制障碍函数。为软约束，需要平衡目标与安全的损失比重，在理论上无法保证安全成功率。

[8]使用安全滤波（safety filter）。

[9]实现基于安全的重要性采样。对于变化的约束，如动态障碍物，需要重新设计控制器；如果需要提升安全性，需要再多一层迭代，影响计算效率；无法确保 100%安全性；在复杂约束下实时性不足。

[10] 用贝叶斯神经网络（BNN）为每个硬约束构建代理模型，输出约束满足的均值估计和认知不确定性度量，进而计算候选轨迹可行的联合概率。

[11]融入通过时序差分学习离线训练的终端价值函数，以近似长期的未来成本。这使得在大幅缩短滚动轨迹长度的同时，能够进行无限范围的推理，从而提高计算效率和运动性能。其次，我们提出了一种折扣调制策略，根据约束违反情况调整采样轨迹的回报。与传统的成本塑造相比，这为约束实施提供了一种更具可解释性和有效性的机制。

强化学习缺乏模型预测控制的几个关键优势：难以执行硬约束、需要大量训练，且在训练领域之外的泛化能力较差。

本文采用“离线 RL 训练+在线 MPPI 控制”的混合框架，创新点如下：

1.

2.

3

.

2 Preliminaries

2.1 问题定义与符号

对于一个 n 自由度的机械臂，$q \in \mathbb{R}^n$ 表示机器人配置空间中一组关节角度，关节角度与速度作为系统状态向量 $x=[q,\dot{q}]$，关节加速度作为控制向量 u。将机器人系统写成离散系统形式：

$$x_{t+1} = f\left(x_t, u_t\right) \tag{1}$$

无限时域最优控制寻求一系列控制输入，以最小化无限时域累积成本：

$$\min_{\{u_0,u_1,\ldots\}} \sum_{t=0}^{\infty} \ell\left(x_t, u_t\right) \tag{2}$$

式中，$\ell(\cdot)$为过程成本函数。在此基础上，进一步引入三类核心约束以保障轨迹规划的安全性、可行性与物理合规性：其一为避障约束，定义关节配置与障碍物间的最小安全距离阈值，确保机械臂运动过程中连杆、关节与工作空间内静态 / 动态障碍物无碰撞；其二为状态约束，限定关节角度、关节速度的取值边界，避免系统进入物理不可行状态；其三为控制约束，约束关节加速度的幅值上限，匹配执行器输出能力，防止因控制量突变引发机械振动或硬件损伤。

2.2 Model Predictive Path Integral (MPPI)

MPPI 属于采样型模型预测控制方法，在名义控制附近做随机搜索，但不是选最小成本轨迹，而是用指数权重把更新方向集中到一组低成本样本上，从而得到更平滑的在线优化控制。 对于滚动时域内的名义控制序列 $\bar{\mathrm{U}}=\{\bar{u}_0,\ldots,\bar{u}_H\}$，在时刻 t 对 K 条样本采样噪声：

$$\varepsilon_i^k \sim N\left(0,\Sigma\right) \tag{3}$$

式中，$i=0,\ldots,H-1$，$k=1,\ldots,K$。以此构造候选控制量：

$$u_i^k = \bar{u}_i + \varepsilon_i^k \tag{4}$$

对每条候选控制序列执行 rollout 得到轨迹$\left\{x_{t+i}^k\right\}_{t=0}^{H}$，滚动时域 H 上的累计成本为：

$$S_k = S\left(U^k; x_t^k\right) = \sum_{i=0}^{H-1} \ell\left(x_{t+i}^k, u_{t+i}^k\right) + \ell_f\left(x_{t+H}^k\right) \tag{5}$$

式中，$U = \{u_t, ..., u_{t+H-1}\}$ 为控制序列，$\ell(\cdot)$ 为过程成本函数，$\ell_f(\cdot)$ 为终端成本函数。并采用指数型权重强调低成本轨迹：

$$\omega_k = \exp\left(-\frac{1}{\lambda}\left(S_k - S_{\min}\right)\right) \tag{6}$$

使用归一化权重 $\tilde{\omega}_k$ 对噪声做加权平均，更新名义控制序列：

$$\bar{u}_i \leftarrow u_i + \sum_{k=1}^{K} \tilde{\omega}_k \varepsilon_i^k \tag{7}$$

最终控制输入为$u_t^* = \bar{u}_0$，即仅提取优化后控制序列的初始时刻控制信号作用于机械臂执行器，以驱动机器人完成当前时间步的运动。在进入下一时刻 t+1 时，需对控制序列执行滚动操作。

2.3 soft actor-critic(SAC)

强化学习是一种流行的基于采样的方法，用于寻找最大化期望回报的近似最优策略。SAC 在最大熵强化学习框架下，同时最大化期望回报与策略熵，从而在学习过程中保持探索性并提高训练稳定性。其目标函数为：

$$\max_{\pi} \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty} \gamma^t \left(r\left(s_t, \mathrm{a}_t\right) + \alpha \mathcal{H}\left(\pi\left(\cdot \mid s_t\right)\right)\right)\right], \tag{8}$$

其中γ为折扣因子，$\mathcal{H}\left(\pi\left(\cdot \mid s_t\right)\right)$表示策略熵，$\alpha$ 为温度参数。

SAC 采用双Q网络$Q_{\theta_1}(s,a)$，$Q_{\theta_2}(s,a)$以缓解Q值过估计。对于经验回放池$\mathcal{D}$中的样本(s,a,r,s',d)，soft Bellman 备份目标为

$$y = r + \gamma(1-d)\left(\min_{i\in\{1,2\}} Q_{\bar{\theta}_i}\left(s', a'\right) - \alpha \log \pi_\phi\left(a' \mid s'\right)\right), a' \sim \pi_\phi\left(\cdot \mid s'\right), \tag{9}$$

并最小化 TD 误差

$$\min_{\theta_i} \mathbb{E}_{(s,a,r,d',d)\sim\mathcal{D}}\left[\left(Q_{\theta_i}(s,a) - y\right)^2\right], \tag{10}$$

其中，$\bar{\theta}_i$为 target critic 参数，通过软更新维持训练稳定性。

策略通过最大化“高Q+高熵”目标来更新：

$$\min_{\phi} \mathbb{E}_{s\sim\mathcal{D},\ a\sim\pi_\phi}\left[\alpha \log \pi_\phi(a \mid s) - \min_{i\in\{1,2\}} Q_{\theta_i}(s,a)\right]. \tag{11}$$

基于上述原理，SAC 采用神经网络$\pi_\phi(\cdot \mid x)$学习一种策略使得目标函数(8)最大，其中ϕ为待学习的神经网络参数。

3 SAC-MPPI 轨迹规划

MPPI 作为一种在线算法，需在每个时间步计算最优控制输入，以保障从任

意状态启动的控制性能。然而，MPPI 的在线计算过程往往耗时较多，尤其针对具有长预测范围的高维动态系统，因需大量采样与轨迹评估，易导致实时性下降。为解决这一问题，本研究引入 Soft Actor - Critic（SAC）算法，通过离线训练学习名义轨迹，分别实现在线 MPPI 计算量的缩减和预测范围的优化，最终达成“离线学习赋能在线控制” 的高效轨迹规划目标。

3.1 算法框架

本文提出的 RL-MPPI 算法框架如下，分为离线学习和在线规划两部分。

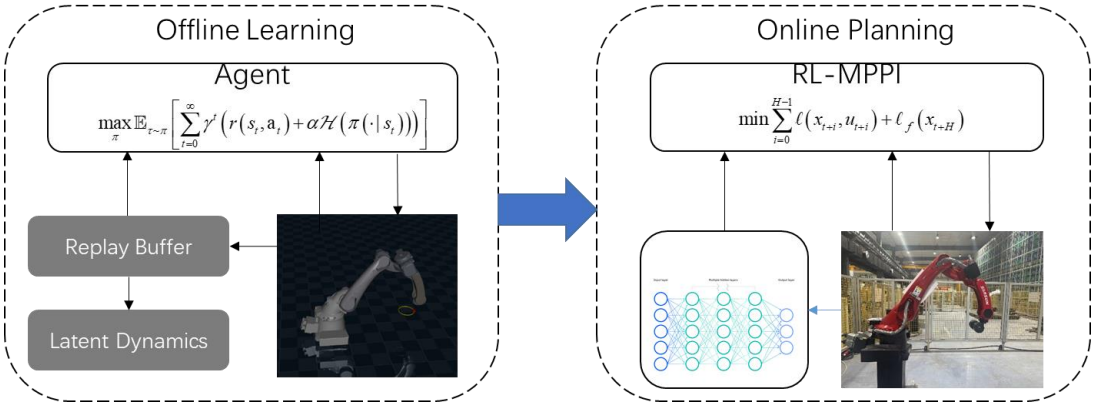


图 1

离线学习阶段，通过仿真环境学习最优控制策略。在线规划阶段，加载学习到的控制策略作为 MPPI 的名义控制序列。

1 采样效率更高、实时性更强：RL 先验将采样集中在更可能低代价 / 可行的区域，重要性采样方差更小；在相同样本数 K 下更容易得到有效轨迹，或在达到同等性能时可减少 K。

2 对分布外场景更鲁棒：SAC 策略在目标变化、障碍物加入等情况下可能出现策略偏差；MPPI 的在线优化可对 RL 输出进行纠偏，避免直接冲入不可行区域。

3 学习先验 + 在线优化互补：RL 提供经验知识（快速给出合理控制方向），MPPI 提供在线最优性提升（针对当前状态与环境即时重算），两者结合通常能在复杂场景下获得更稳定的到达与避障行为。

3.2 基于折扣因子的约束

为实现机械臂轨迹规划中多类约束的统一化、自适应管理，本节首先构建约束通用表达形式，再通过随机终止信号将约束违规程度转化为奖励折扣权重，最后结合指数移动平均（EMA）与有限时域折扣规则，形成兼顾安全性与控制连续性的约束处理方案。

机械臂运动需满足关节位置边界、速度幅值上限、加速度执行器能力等多维度物理约束，这些约束可统一抽象为不等式约束通用形式：

$$c_i(x,u)\le 0 \quad \forall i\in\mathcal{I}\ , \tag{12}$$

为避免传统手工设计惩罚函数存在的阈值依赖、违规叠加风险，本节采用基于约束违规的随机终止信号，直接将违规程度映射为对未来奖励的折扣权重，定义为：

$$\delta_h = \max_{i \in \mathcal{I}} \left(p_i^{\max} \cdot \mathrm{clip}\left(\frac{[c_h]_i}{c_i^{\max}}, 0, 1 \right) \right) \tag{13}$$

式中，超参数 $p_i^{\max} \in [0,1]$ 为约束 i 的最大终止概率超参数，$[c_h]_i$ 为第 h 步中约束 i 的实际违规量，$c_i^{\max}$ 是所有轨迹中第 i 个约束的最大违反值。公式(13)具有如下性质：

1) 约束违规不再依赖手工设计的惩罚函数，而是直接将违规程度转化为对未来奖励的折扣权重，违规越严重，δ_h 越接近 $p_i^{\max}$，未来奖励被折扣的比例越高，引导机器人主动规避违规。

2) 只要有一项约束严重违规，就会触发较高的 δ_h，避免多项轻微违规叠加导致安全风险或运动失效，确保约束的严格性。

3) 无违规时，$[c_h]_i = 0$，$\delta_h = 0$，未来奖励无折扣，鼓励正常探索；轻微违规时，$0 < [c_h]_i < c_h^{\max}$，$\delta_h \in (0, p_i^{\max})$，未来奖励部分折扣，既警示违规又允许机器人学习恢复策略；严重违规时，$[c_h]_i \geq c_h^{\max}$，$\delta_h = p_i^{\max}$，直接终止未来所有奖励，严格禁止严重违规。

为避免单一批次极端违规值导致的波动，$c_i^{\max}$ 采用指数移动平均（EMA）机制，更新方式如下：

$$c_i^{\max} \leftarrow \tau_c \cdot \hat{c}_i^{\max} + (1 - \tau_c) \cdot c_i^{\max} \tag{14}$$

式中，$\hat{c}_i^{\max} = \max_h [c_h]_i$ 是当前训练批次中所有时刻 h，第 i 个约束的最大违反值，$\tau_c \in [0,1]$ 是衰减率超级参数，表示保留历史信息的百分率。无需手动设定固定违规阈值，通过数据自动学习合理的违规参考范围，适配不同任务的约束要求。

3.3 离线策略学习

我们采用 TD-CD(Temporal-Difference Constraint-Discounting)的思想将约束信息以软终止形式注入到 SAC 的时序差分备份中，实现对约束的折扣化处理。训练过程为在线交互式 off-policy 学习：agent 在受约束动力学环境中滚动采样转移 (s_t, u_t, r_t, s_{t+1})，并将每一步的约束违反程度映射为一个软终止强度 δ_t，进而得到时变折扣因子 γ_t。

在学习更新时，我们使用标准 SAC 的双 Q 网络与最大熵策略更新框架，但在 critic 的 TD 目标中使用时变折扣 γ_t。对从回放池采样的 minibatch $\{(s_i, u_i, r_i, s_i', \gamma_i)\}$，先构造软值函数目标：

$$V(s_t') = \mathbb{E}_{a \sim \pi} \left[\min\left(Q_{\bar{\theta}_1}(s_i', a), Q_{\bar{\theta}_2}(s_i', a) \right) - \alpha \log \pi(a \mid s_i') \right] \tag{15}$$

并令

$$y_i = r_i + \gamma_i V(s_i') \tag{16}$$

随后通过最小化均方误差更新两个 critic。Actor 与温度参数 α 的更新保持 SAC 标准形式不变，约束信息仅通过 critic 的时变折扣在价值传播中体现。最后

对目标 Q 网络执行软更新。这一流程使得约束影响以连续、可解释的方式进入 TD 学习，通常比硬终止/硬惩罚更平滑、稳定，也更适合与 off-policy 回放结合。

Algorithm 1 TD-CD-SAC Online Training (cd_sac_ball)

Require: Constrained env. $\mathcal{E}$, discount γ, batch size B
Require: TD-CD params $p^{\max}$, τ_c, switch `use_amount`
Initialize SAC (two critics, actor, targets) and replay buffer $\mathcal{D}$
$c_{\max}^{seen} \leftarrow 0$, $c_{\max}^{ema} \leftarrow 1$; sample $s \sim \mathcal{E}$.reset()
for $t = 0, 1, \ldots, T-1$ **do**
  Sample action $u \sim \pi(\cdot \mid s)$ (or random at start)
  Step env: $(s', r, d, \text{info}) \leftarrow \mathcal{E}$.step($u$)
  ▷ TD-CD: compute per-transition discount γ_t
  if $d = 1$ **then**
    $\gamma_t \leftarrow 0$
  else
    if `use_amount`=1 **then**
      $c \leftarrow$ info.`vel_violation_amount`; $c_{\max}^{seen} \leftarrow \max(c_{\max}^{seen}, |c|)$
      $\delta \leftarrow p^{\max} \,\text{clip}(|c|/c_{\max}^{ema}, 0, 1)$
    else
      $c \leftarrow \mathbb{I}[\text{info.`constraint_violation`} = 1]$; $\delta \leftarrow p^{\max} c$
    end if
    $\gamma_t \leftarrow \gamma(1-\delta)$
  end if
  Store (s, u, r, s', γ_t) into $\mathcal{D}$; set $s \leftarrow s'$
  if $|\mathcal{D}| \geq B$ **then**
    Sample $\{(s_i, u_i, r_i, s_i', \gamma_i)\}_{i=1}^{B} \sim \mathcal{D}$
    ▷ Same as SAC, but TD target uses stored γ_i
    Update SAC with targets $y_i = r_i + \gamma_i V(s_i')$
  end if
  if $t \bmod N_{\text{eval}} = 0$ **then**
    $c_{\max}^{ema} \leftarrow \tau_c c_{\max}^{ema} + (1-\tau_c)\max(c_{\max}^{seen}, \epsilon)$; $c_{\max}^{seen} \leftarrow 0$
  end if
end for

3.4 在线轨迹规划

使用离线学习得到的$\pi_\phi(\cdot \mid x)$作为 nominal 控制输入，将

该滚动操作的核心意义在于，使 MPPI 能基于机器人实时更新的状态，持续在滚动时域内重新采样、评估并优化控制序列，避免因固定时域规划导致的模型失配问题。

当机械臂运动过程中检测到工作空间内新增动态障碍物时，通过序列滚动可快速将新的环境约束纳入下一时刻的控制优化，确保后续运动始终满足避障等安全要求，同时维持控制的连续性与实时性。

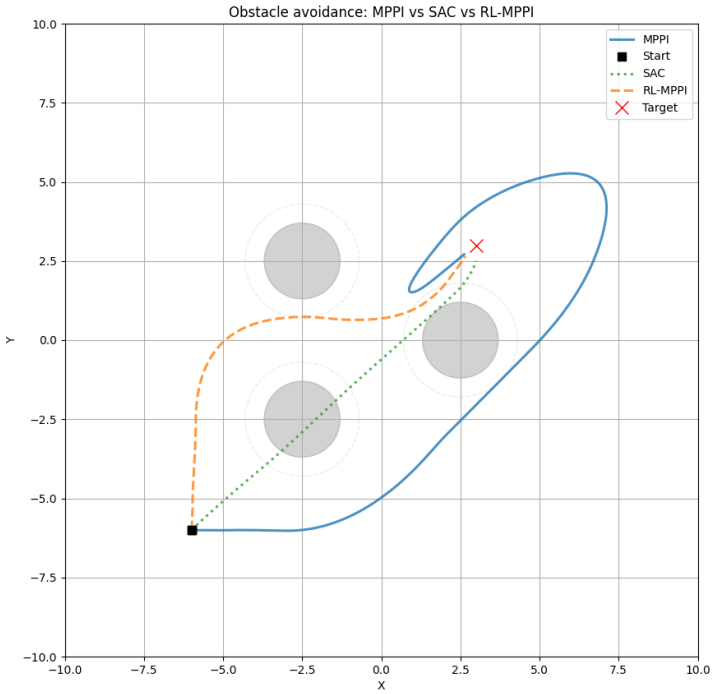
Algorithm 1 Model Predictive Path Integral Control (MPPI)

Require: Current state x_t, dynamics model f, stage cost function c, terminal cost function c_f, number of samples N, control horizon T, noise covariance Σ_u, temperature parameter λ

Ensure: Optimal control sequence $\mathbf{u}^*$

Initialization:
  Control sequence $\mathbf{u}_0 = \mathbf{0} \in R^{T\times m}$ (or previous control sequence)
  Noise sequences $\{\delta\mathbf{u}_i\}_{i=1}^N$, where $\delta u_i^k \sim \mathcal{N}(0, \Sigma_u), \forall k = 1, \ldots, T$
Generate Sampled Trajectories:
for $i = 1$ to N **do**
    $x_i^0 = x_t$
    $\mathbf{u}_i = \mathbf{u}_0 + \delta\mathbf{u}_i$
    $J_i = 0$
  for $k = 0$ to $T - 1$ **do**
      $x_i^{k+1} = f(x_i^k, u_i^k)$
      $J_i += c(x_i^k, u_i^k)$
  end for
    $J_i += c_f(x_i^T)$
end for
Calculate Weights:
  $J_{min} = \min\{J_1, J_2, \ldots, J_N\}$
  $\omega_i = \exp\left(-\frac{J_i - J_{min}}{\lambda}\right), \quad i = 1, \ldots, N$
  $\omega_i = \omega_i / \sum_{j=1}^N \omega_j$
Calculate Optimal Control Sequence:
  $\mathbf{u}^* = \mathbf{u}_0 + \sum_{i=1}^N \omega_i \cdot \delta\mathbf{u}_i$
return $\mathbf{u}^*$

4 Experiments



5 结论

参考文献

[1] Liu J, Yap H J, Khairuddin A S M. Review on motion planning of robotic manipulator in dynamic environments[J]. Journal of Sensors, 2024, 2024(1): 5969512.

[2] Jordana A, Zhang J, Amigo J, et al. An Introduction to Zero-Order Optimization Techniques for Robotics[J]. arXiv preprint arXiv:2506.22087, 2025.

[3] Wang X, Li H, Wang D, et al. Constrained sampling-based MPC using path integral for collision-free robot manipulation[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2025.

[4] Koptev M, Figueroa N, Billard A. Reactive collision-free motion generation in joint space via dynamical systems and sampling-based MPC[J]. The International Journal of Robotics Research, 2024, 43(13): 2049-2069.

[5] Hansen N, Wang X, Su H. Temporal difference learning for model predictive control[J]. arXiv preprint arXiv:2203.04955, 2022.

[6] Ames A D, Coogan S, Egerstedt M, et al. Control barrier functions: Theory and applications[C]//2019 18th European control conference (ECC). Ieee, 2019: 3420-3431.

[7] Yin, Ji, et al. "Shield Model Predictive Path Integral: A Computationally Efficient Robust MPC Approach Using Control Barrier Functions." arXiv preprint arXiv:2302.11719 (2023).

[8] Rabiee P, Hoagg J B. Guaranteed-safe mppi through composite control barrier functions for efficient sampling in multi-constrained robotic systems[J]. arXiv preprint arXiv:2410.02154, 2024.

[9] Gandhi M, Almubarak H, Theodorou E. Safe importance sampling in model predictive path integral control[J]. arXiv preprint arXiv:2303.03441, 2023.

[10] Ezeji O, Ziegltrum M, Turrisi G, et al. BC-MPPI: A Probabilistic Constraint Layer for Safe Model-Predictive Path-Integral Control[C]//Workshop on Agents and Robots for reliable Engineered Autonomy. Cham: Springer Nature Switzerland, 2025: 131-143.

[11] Crestaz P N, De Matteis L, Chane-Sane E, et al. TD-CD-MPPI: Temporal-Difference Constraint-Discounted Model Predictive Path Integral Control[J]. 2025.

[12] Almubarak H, Sadegh N, Theodorou E A. Safety embedded control of nonlinear systems via barrier states[J]. IEEE Control Systems Letters, 2021, 6: 1328-1333.

[13] Chane-Sane E, Leziart P A, Flayols T, et al. Cat: Constraints as terminations for legged locomotion reinforcement learning[C]//2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2024: 13303-13310.

[14]