

基于IPSO优化的MPC轨迹跟踪控制研究

赵宇涵

上海理工大学健康科学与工程学院, 上海

收稿日期: 2025年2月21日; 录用日期: 2025年3月14日; 发布日期: 2025年3月21日

摘要

针对机械臂轨迹跟踪过程中因外界干扰和固定控制器参数难以应对动态环境变化而导致的跟踪精度不足问题, 本文提出了一种基于改进粒子群算法的机械臂自适应模型预测轨迹跟踪控制策略。通过机械臂动力学方程建立系统模型, 并设计用于轨迹跟踪的模型预测控制器, 引入粒子群优化算法对控制器中的权重系数进行在线调整。采用线性惯性权重下降策略, 克服了传统粒子群算法易陷入局部最优的问题, 同时提高了收敛速度。仿真实验结果表明, 所提出的改进粒子群优化模型预测控制器具有更稳定、精准的跟踪性能, 相较于原版控制器和传统粒子群优化控制器, 关节角均方误差分别降低了33.85%和23.08%。

关键词

轨迹跟踪, 模型预测控制, 粒子群优化算法, 机械臂

Research on MPC Trajectory Tracking Control Based on IPSO Optimization

Yuhan Zhao

School of Health Science and Engineering, University of Shanghai for Science and Technology, Shanghai

Received: Feb. 21st, 2025; accepted: Mar. 14th, 2025; published: Mar. 21st, 2025

Abstract

To address the issue of insufficient tracking accuracy in robotic arm trajectory tracking due to external disturbances and the inability of fixed controller parameters to adapt to dynamic environments, this paper proposes an adaptive model predictive trajectory tracking control strategy based on an improved particle swarm optimization (PSO) algorithm. The system model is established using the dynamic equations of the robotic arm, and a model predictive controller (MPC) is designed

for trajectory tracking. The PSO algorithm is introduced to adjust the weight coefficients of the controller online. A linear inertia weight decay strategy is employed to mitigate the problem of traditional PSO algorithms easily falling into local optima, while also improving convergence speed. Simulation experiments demonstrate that the proposed improved PSO-based MPC achieves more stable and accurate tracking performance, with the mean squared error of joint angles reduced by 33.85% and 23.08% compared to the original controller and the traditional PSO-improved controller, respectively.

Keywords

Trajectory Tracking, Model Predictive Control, Particle Swarm Optimization, Manipulator

Copyright © 2025 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

机器人控制技术在工业自动化、医疗诊断、精密制造等领域扮演着关键角色，特别是在机械臂的轨迹跟踪控制中，其精度和鲁棒性直接决定了系统的整体性能。然而，随着工作环境的复杂性增加和任务需求的多样化，传统的控制方法面临着显著挑战。尽管 PID 控制等方法已被广泛应用，它们在处理高非线性和高耦合的机械臂系统模型时常常表现出局限性[1]。模型预测控制(Model Predictive Control, MPC) 因其在处理多变量约束和优化性能方面的优势，近年来受到广泛关注。

在 2022 年, Jingkun Yan 等人提出了一种基于滚动地平线控制(Receding Horizon Control, RHC)的冗余度机器人轨迹跟踪规划方案。他们通过对机械手非线性模型的线性化, 构造了一个最小化跟踪误差、速度范数和加速度范数的凸二次规划问题, 并利用递归神经网络模型来求解该 RHC 方案[2]。紧接着, 2023 年, Tianqi Zhu 等人研究了一种基于运动学的 MPC 方法, 生成了平滑的轨迹并通过低层 MPC 进行轨迹跟踪和安全保护, 同时采用超扭曲观测器来提高动态障碍物运动的估计精度, 以实现动态避障的安全和平稳[3]。然而, 这些研究基于运动学的模型简化了系统的复杂性, 可能导致对动态系统的不充分逼近, 这限制了其在处理复杂实时动态场景中的应用。为此, Qirong Tang 等人提出了一种基于机械臂动力学模型的 MPC 方案, 专注于解决轨迹跟踪中的输入约束问题。尽管其方法考虑了机械臂的动力学特性, 但在 MPC 中复杂的参数调节问题仍未得到有效解决[4]。

传统的 MPC 方法在实际应用中往往难以实现对系统参数的在线调整, 尤其是在高动态环境下, 固定参数的 MPC 难以适应环境的变化。因此, 发展一种能够自适应调整参数的 MPC 方法具有重要的研究意义。为了解决这一问题, 粒子群优化算法(Particle Swarm Optimization, PSO)作为一种强有力的数值优化工具, 被广泛应用于动态优化控制领域[5]。PSO 因其简单性和高效性受到青睐, 但其在实践中也存在一些显著缺陷。例如, PSO 容易陷入局部极小点, 且优化过程可能耗时较长[6], 这在实时控制 MPC 中尤为突出。因此, 必须探索改进的 PSO 算法或其他更快速高效的优化方法, 以满足实时控制的需求。

本文提出了一种基于自适应权重的 MPC 控制策略, 通过离散化动力学方程构建了一个二次规划问题, 将关节扭矩作为控制量, 关节角作为状态量, 并通过改进的 PSO 算法对控制器进行在线参数调整。实验结果表明, 该方法在提升控制精度和系统鲁棒性方面表现出显著优势。

本文结构安排如下: 第二部分介绍系统模型及其离散化处理, 第三部分讨论控制器设计原理, 第四部分进行仿真实验验证, 第五部分总结。

2. 系统模型

2.1. 机械臂动力学模型描述

机械臂动力学方程描述了机械臂在关节空间中的动力学行为，它反映了关节扭矩与关节位置、速度和加速度之间的关系[7]。基于二阶欧拉-拉格朗日方法，机械臂在连续时间动力学方程可以表示为：

$$M(\theta)\ddot{\theta}(t)+C(\theta,\dot{\theta})\dot{\theta}(t)+G(\theta)=\tau(t) \quad (1)$$

其中 $\theta, \dot{\theta}, \ddot{\theta}$ 分别代表机械臂 n 维关节加速度、速度以及位置向量。 $M(\theta), C(\theta, \dot{\theta}), G(\theta)$ 分别表示惯性矩阵、科里奥利和离心力矩阵与重力向量，三个矩阵会随着关节角的变化而随时间更新。 τ 代表机械臂 n 维关节力矩向量，表示施加在关节处的驱动力矩。

2.2. 状态空间方程

控制器设计需要系统动态的精确数学描述，即系统的状态空间方程。对于 n 自由度机械臂，定义系统状态量为 $x(t)=[x_1(t), x_2(t)]^T=[q(t), \dot{q}(t)]^T \in \mathbb{R}^{2n}$ ，控制量为 $u(t)=\tau(t) \in \mathbb{R}^n$ 。将动力学系统重新排列并带入到状态量的导数中可以得到：

$$\dot{x}(t)=\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix}=\begin{bmatrix} \dot{x}_1(t) \\ M^{-1}(x_1)(u(t)-(C(x_1, x_2)\dot{x}_1(t)+G(x_1))) \end{bmatrix} \quad (2)$$

基于上述推导可将系统的状态空间方程组写为：

$$\begin{cases} \dot{x}_2 = \dot{x}_1 \\ \dot{x}_1 = m(x_1)u(t) + n(x_1, x_2) \\ y = x_1 = Dx \end{cases} \quad (3)$$

其中记 $D=[I_{(n)} \quad O_{(n \times n)}]$ ，矩阵 $I_{(n)}$ 为 n 阶单位矩阵，矩阵 $O_{(n \times n)}$ 为全零矩阵。记 $m(x_1)=M^{-1}(\theta)$ ， $n(x_1, x_2)=-M^{-1}(\theta)(C(\theta, \dot{\theta})x_2+G(\theta))$ 。

2.3. 模型离散化

在实时控制系统中，采用固定的时间步长进行控制对于保证系统的稳定性和性能至关重要。采用泰勒展开法将系统状态量展开到方程所包含的最高阶项[8]，即关节角加速度项：

$$x(t+T)=\begin{bmatrix} x_1(t+T) \\ x_2(t+T) \end{bmatrix}=\begin{bmatrix} x_1(t)+T\dot{x}_1(t)+\frac{T^2}{2}\ddot{x}_1(t) \\ x_2(t)+T\dot{x}_2(t) \end{bmatrix} \quad (4)$$

将公式(3)带入公式(4)，将 $x(t+(k+1)T)$ 记作 $x(k+1)$ ，化简即可得到系统离散状态方程：

$$\begin{cases} x(k+1)=Ax(k)+Bu(k)+C \\ y=x_1(k)=Dx(k) \end{cases} \quad (5)$$

其中

$$A=\begin{bmatrix} I_n & TI_n \\ O_{(n \times n)} & I_n \end{bmatrix} \quad B=\begin{bmatrix} \frac{T^2}{2}m(x_1) \\ Tm(x_1) \end{bmatrix} \quad C=\begin{bmatrix} \frac{T^2}{2}n(x_1, x_2) \\ Tn(x_1, x_2) \end{bmatrix}$$

在MPC中， N_p 代表预测区间。MPC的核心思想是求解一个有约束的二次规划问题[9]，来寻找使得

设计目标函数最小化的最优控制输入。这个目标函数通常包括偏差、控制量变化等方面的约束，从而在未来的多个时间步长内保持系统性能的最优。此外，为了加速优化进程提高算法效率，定义控制范围 $N_c (N_c \leq N_p)$ 为控制区间，在该范围内计算用于优化的控制输入。记 $x(k+i|k)$ 为当前 k 时刻对 $k+i$ 时刻的状态量预测，基于公式(5)，将其迭代写为多步预测的形式。

$$\begin{aligned} y(k+1|k) &= Dx(k+1|k) = D(Ax(k) + Bu(k) + C) \\ y(k+2|k) &= DA^2x(k|k) + DABu(k+1|k) + DAC + DBu(k) + DC \\ &\dots \\ y(k+N_p|k) &= DA^{N_p}x(k|k) + DA^{N_p-1}Bu(k|k) + DA^{N_p-1}C + \dots \\ &\quad + DABu(k+N_p-2|k) + DAC + DBu(k+N_p-1|k) + DC \end{aligned} \quad (6)$$

基于公式(6)，将多步预测方程整理为：

$$\begin{cases} Y_{pre}(k) = Ex(k) + FU(k) + G \\ U(k) = U_{old} + H\Delta U(k) \end{cases} \quad (7)$$

其中

$$\begin{aligned} Y_{pre} &= \begin{bmatrix} y(k+1) \\ y(k+2) \\ \vdots \\ y(k+N_c) \\ \vdots \\ y(k+N_p) \end{bmatrix} & G &= \begin{bmatrix} DC \\ D(AC+C) \\ \vdots \\ D(A^{N_c-1}C + A^{N_c-2}C + \dots AC + C) \\ \vdots \\ D(A^{N_p-1}C + A^{N_p-2}C + \dots AC + C) \end{bmatrix} \\ E &= \begin{bmatrix} DA \\ DA^2 \\ \vdots \\ DA^{N_c} \\ \vdots \\ DA^{N_p} \end{bmatrix} & F &= \begin{bmatrix} DB & 0 & 0 & \dots & 0 \\ DAB & DB & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ DA^{N_c-1}B & DA^{N_c-2}B & DA^{N_c-3}B & \dots & DA^{N_c-N_c}B \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ DA^{N_p-1}B & DA^{N_p-2}B & DA^{N_p-3}B & \dots & DA^{N_p-N_c}B \end{bmatrix} \\ H &= \begin{bmatrix} I_n & \dots & O_{n \times n} \\ \vdots & \ddots & \vdots \\ I_n & \dots & I_n \\ \vdots & \ddots & \vdots \\ I_n & \dots & I_n \end{bmatrix}_{N_p \times N_c} & \Delta U(k) &= \begin{bmatrix} u(k) \\ u(k+1) \\ u(k+2) \\ \vdots \\ u(k+N_c-1) \end{bmatrix} - \begin{bmatrix} u_{old} \\ u_{old} \\ u_{old} \\ \vdots \\ u_{old} \end{bmatrix} = \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \Delta u(k+2) \\ \vdots \\ \Delta u(k+N_c-1) \end{bmatrix} \end{aligned}$$

3. 自适应 MPC 控制器设计

3.1. 基本 MPC 控制器设计

基于 MPC 的轨迹跟踪控制器主要目的是在一定约束条件下优化控制量增量，以减小误差并达到更好的跟踪效果[10]。其中，约束条件分为硬约束和软约束，硬约束为机械臂自身的关节限位，软约束设定为机械臂的关节扭矩需要在合理范围内。基于以上，如公式所示，MPC 中的代价函数可以设为，在满足硬约束条件的前提下，最小化跟踪误差与控制量增量之和，同时将软约束作为代价函数中的惩罚项，以

确保关节扭矩在合理范围内。

$$\begin{aligned} \min_{\Delta U(k)} J(x(k), \Delta U(k)) &= \sum_{k=0}^{N_p-1} \|y_{pre}(k) - y_{ref}(k)\|_Q^2 + \sum_{k=0}^{N_c-1} \|\Delta u(k)\|_R^2 \\ s.t. \quad &u_{\min}(k+i) \leq u(k+i) \leq u_{\max}(k+i) \\ &\Delta u_{\min}(k+i) \leq \Delta u(k+i) \leq \Delta u_{\max}(k+i) \\ &i = 0, 1, \dots, N_c - 1 \end{aligned} \quad (8)$$

其中, $Y_{pre}(k)$ 为 k 时刻系统的多步预测输出, $Y_{ref}(k)$ 为跟踪任务中下发的参考轨迹。 Q 、 R 分别表示误差与控制权重系数矩阵。

在有约束的情况下, 通常无法直接推导出解析的控制律, 因此需要在线求解最优控制问题来获得数值解。线性模型的 MPC 优化问题可以转化为二次规划问题。这里先将误差项定义为:

$$L = Y_{ref} - Ex(k) - FU_{old} - G \quad (9)$$

将公式(9)带入公式(8)可得:

$$\begin{aligned} \min_{\Delta U(k)} J(x(k), \Delta U(k)) &= \|F(U(k) - U_{old}) - L\|_Q^2 + \|\Delta U(k)\|_R^2 \\ &= \|FH\Delta U(k) - L\|_Q^2 + \|\Delta U(k)\|_R^2 \\ &= \Delta U^T(k) H^T F^T \bar{Q}^T \bar{Q} FH \Delta U(k) + L^T \bar{Q}^T \bar{Q} L \\ &\quad - 2L^T \bar{Q}^T \bar{Q} FH \Delta U(k) + \Delta U^T(k) \bar{R}^T \bar{R} \Delta U(k) \end{aligned} \quad (10)$$

其中 $\bar{Q} = \text{diag}\{Q, Q, \dots, Q\}$, $\bar{R} = \text{diag}\{R, R, \dots, R\}$ 。无关项 $L^T \bar{Q}^T \bar{Q} L$ 不是关于 ΔU 的相关量, 对优化结果不会产生影响, 目标函数可进一步简化为:

$$\min_{\Delta U(k)} J(\Delta U(k)) = \Delta U^T(k) \varphi \Delta U(k) + \gamma \Delta U(k) \quad (11)$$

其中 $\varphi = H^T F^T \bar{Q}^T \bar{Q} FH + \bar{R}^T \bar{R}$, $\gamma = -2L^T \bar{Q}^T \bar{Q} FH$, 公式(11)为标准二次型形式。公式(8)中的约束可以写为以下形式:

$$\begin{aligned} \begin{bmatrix} I_n & \dots & 0 \\ \vdots & \ddots & \vdots \\ I_n & \dots & I_n \\ -I_n & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & -I_n \end{bmatrix} \Delta U(k) \leq \begin{bmatrix} \Delta u_{\max}(k) \\ \vdots \\ \Delta u_{\max}(k + N_c - 1) \\ -\Delta u_{\min}(k) \\ \vdots \\ -\Delta u_{\min}(k + N_c - 1) \end{bmatrix} \\ \begin{bmatrix} I_n & \dots & 0 \\ \vdots & \ddots & \vdots \\ I_n & \dots & I_n \\ -I_n & \dots & 0 \\ \vdots & \ddots & \vdots \\ -I_n & \dots & -I_n \end{bmatrix} \Delta U(k) \leq \begin{bmatrix} u_{\max}(k) - u_{old} \\ \vdots \\ u_{\max}(k + N_c - 1) - u_{old} \\ u_{old} - u_{\min}(k) \\ \vdots \\ u_{old} - u_{\min}(k + N_c - 1) \end{bmatrix} \end{aligned} \quad (12)$$

截至目前, 本文已建立了基于控制量及其增量约束(12)的轨迹跟踪问题目标函数(11)。控制器在优化过程中, 通过实时在线求解目标函数, 可获得机械臂关节扭矩的控制增量, 确保高精度的轨迹跟踪和良好的鲁棒性。

3.2. 基于 PSO 的自适应 MPC 控制器设计

由于机械臂系统具有非线性和时变等特性，固定的 MPC 参数可能无法在所有情况下提供最佳控制。此外，外部干扰和系统内部的不确定性也可能影响控制效果。为应对这些挑战，本研究提出了一种通过动态调节关键参数来优化控制性能的方法。具体来说，研究通过实时监测系统性能指标，选取轨迹跟踪误差的均方误差作为主要指标，并采用粒子群优化算法(PSO)对 MPC 的权重系数进行动态调整。

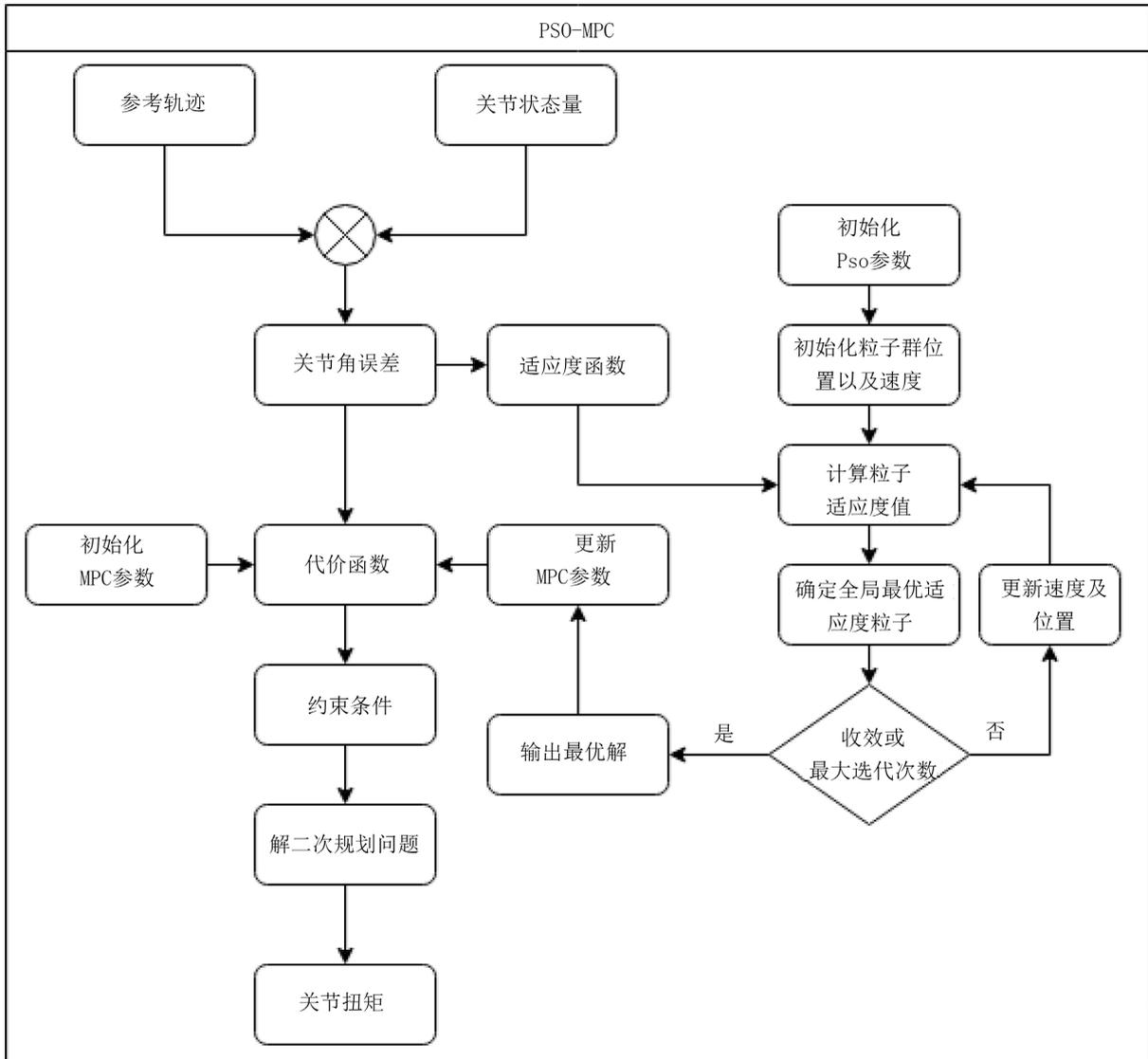


Figure 1. Flowchart of adaptive MPC controller algorithm based on PSO

图 1. 基于 PSO 的自适应 MPC 控制器算法流程图

图 1 展示了基于 PSO 的自适应 MPC 轨迹跟踪控制器算法流程。流程图描述了如何通过 PSO 算法实时调整 MPC 的权重系数，以最小化轨迹误差，并在面对不确定因素时实现快速响应。PSO 算法通过模拟粒子群在解空间中的搜索行为，迭代更新每个粒子的速度和位置，找到使轨迹误差最小的最优权重组合。这种动态调整策略能够有效应对机械臂的非线性和时变特性，提高控制系统的鲁棒性和响应性能。算法

具体步骤如下:

(1) 定义适应度函数

在 PSO-MPC 控制器中, 采用机械臂关节角与参考轨迹的角度之间的均方误差作为适应度函数, 用于评估权重系数对状态量的影响。均方误差即适应度函数定义为:

$$\min_{(q,t)} J = \frac{1}{N} \sum_{t=1}^N \|q(t) - q_{ref}(t)\|^2 \quad (13)$$

(2) 初始化种群

在粒子群优化(PSO)算法中, 初始化粒子群是至关重要的第一步。粒子群由多个粒子组成, 每个粒子对应一个潜在解。初始化阶段, 每个粒子的初始位置和速度通常在定义的搜索空间内随机生成, 并且会选择一个较小的速度以防运动过快错失最优解。这里粒子的位置表示 MPC 参数中的权重系数 Q 和 R。

(3) 更新粒子群位置及速度信息

对于每个粒子, 采用以下公式进行位置及速度的更新:

$$\begin{cases} v_i^{k+1} = \omega v_i^k + c_1 r_1 (p_i^{best} - x_i^k) + c_2 r_2 (g^{best} - x_i^k) \\ x_i^{k+1} = x_i^k + v_i^{k+1} \end{cases} \quad (14)$$

其中 v 表示粒子的速度, x 表示粒子的位置, p_i^{best} 是粒子先前最优解, g^{best} 表示全局最优解, 随机数 r_1, r_2 , 加速度系数 c_1, c_2 。权重系数 ω 在一定程度上决定了运动趋势以及整体收敛速度, 后续会对权重系数进行改进优化。

(4) 更新个体及全局最佳位置信息

根据公式(14), 会不断计算出粒子的适应度值, 通过比较当前适应度与历史、全局最优解, 更新历史以及全局最优解, 公式如下:

$$\begin{cases} p_i^{best} = \arg \min(\xi(x_i)) \\ g^{best} = \arg \min(\xi(x_i)) \end{cases} \quad (15)$$

(5) 迭代确定最佳适应度的解

如果达到最大迭代次数或全局最优位置的适应度值满足预定的精度要求, 停止算法; 否则返回步骤(3), 继续迭代。最终的全局最优位置 g^{best} 即为问题的最优解或最优近似解。

3.3. 惯性权重下降的改进 PSO 算法

传统 PSO 算法在搜索过程中, 惯性权重的固定设置可能导致搜索效率和精度的平衡不佳。为进一步提升 PSO 算法在自适应 MPC 中的性能, 基于等人的惯性下降策略研究, 本研究引入了动态惯性权重调节机制。动态惯性权重的引入旨在通过调整惯性权重值, 改进搜索过程的自适应性和鲁棒性, 从而提升 MPC 的控制效果。线性动态惯性权重根据以下公式更新:

$$\omega(t) = \omega_{max} - \frac{(\omega_{max} - \omega_{min}) \times iter}{iter_{max}} \quad (16)$$

具体来说, 惯性权重 ω 的动态调节机制可以描述为: 在算法初始阶段, 设置较大的惯性权重, 以增强全局搜索能力, 使粒子能够快速遍历整个解空间, 进行广泛的探索; 随着迭代次数的增加, 逐渐减小惯性权重, 以增强局部搜索能力, 使粒子在接近最优解时能够进行精细搜索。这种动态调整机制可以显著提高算法的效率和精度, 避免陷入局部最优解。

4. 实验验证

4.1. 仿真环境

本节通过仿真实验来验证前文提出算法的有效性,采用实验室搭建的开源仿真平台 robopal 验证提出算法有效性[11],平台如图 2 所示。

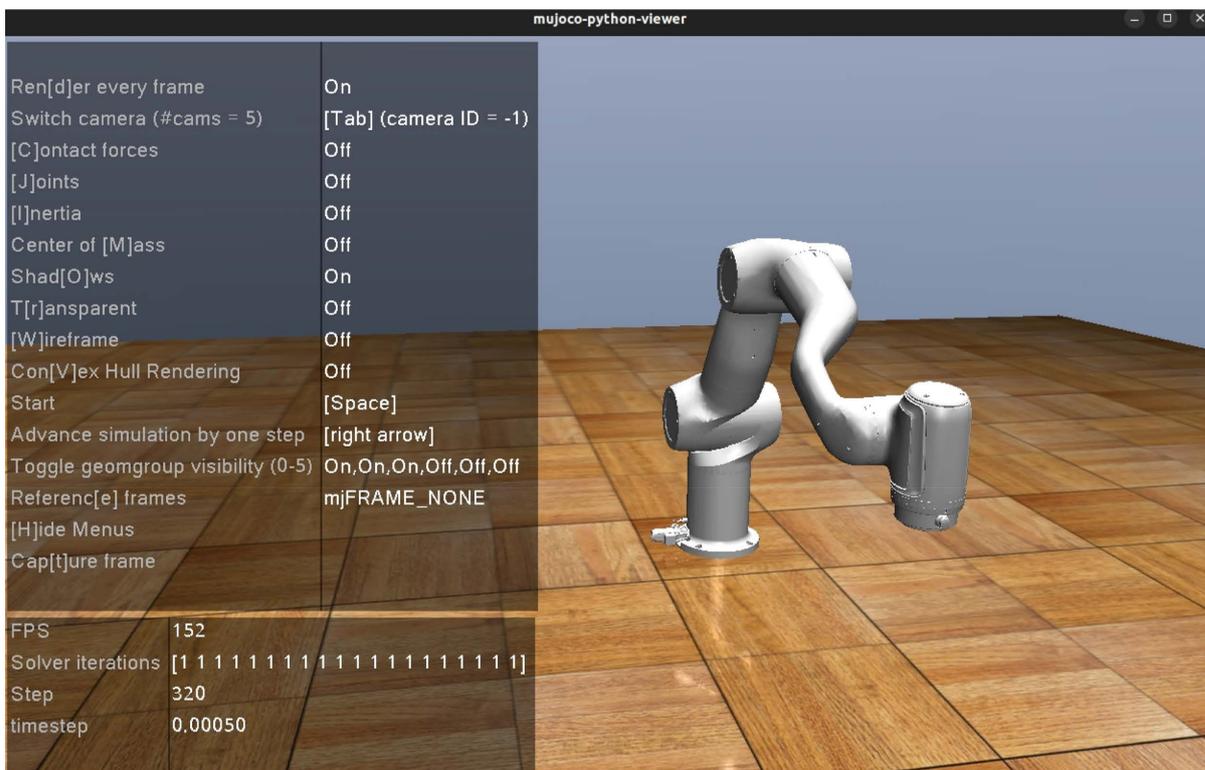


Figure 2. Robopal, a simulation platform built based on MuJoCo

图 2. 基于 MuJoCo 搭建的仿真平台 robopal

robopal 中主要包括基于 kdl 机器人运动学库与 Pinocchio 机器人动力学库机械臂的机械臂正逆运动学与动力学求解、机械臂关节状态量获取以及具备多种基础控制方案,包括关节空间内的位置控制、速度控制和扭矩控制等。这里选择选用思灵 Diana7_med 七自由度机械臂进行实验验证。

4.2. 参数设置

Table 1. Basic MPC parameters and initial values

表 1. 基本 MPC 参数项以及初始值

参数项	设定值
步数	1500
采样时间	0.01
预测时域	15
控制时域	10
误差权重系数	1000
控制权重系数	10

为了验证本文提出的基于改进粒子群优化算法(IPSO)的自适应 MPC 控制器的性能, 设置了三个实验对照组, 分别为原版 MPC、PSO-MPC 和 IPSO-MPC。原版 MPC 中, 权重系数 Q 和 R 的值为固定预设值, 预设值通过实验前对比优化验证得到, 原版 MPC 中参数设置如上表 1 所示。

基于粒子群优化的 PSO-MPC 与 IPSO-MPC 参数初始化设定如表 2 所示。

Table 2. PSO parameters and initial values
表 2. 粒子群参数项以及初始值

参数项	设定值
种群数量	30
迭代次数	20
加速度系数	0.5
最大惯性系数	0.95
最小惯性系数	0.35
误差权重系数搜索范围	[100, 2000]
控制权重系数搜索范围	[0.1, 10]

实验中, 机械臂的初始关节状态为 $q_{start} = [0, -0.641, 0, 1.385, 0, -1.116, 0]^T$ 。实验过程中将参考轨迹直接由关节角增量给出, 每步的各关节增量为 $\Delta q_i = 0.001, i = (1, 2, \dots, 7)$ 。各关节的初始速度默认设置为 0 m/s。公式(12)中, 关节扭矩的最大限度被设为 $20 \text{ N}\cdot\text{m}$, MPC 每步计算出的扭矩增量被限制为 $5 \text{ N}\cdot\text{m}$ 。

4.3. 实验结果与分析

本实验简化了参考轨迹进行逆运动学转化的步骤, 直接对关节进行增量赋值, 笛卡尔空间内轨迹参考价值不大, 这里直接对关节空间内关节角的跟踪误差进行讨论。关节空间内参考轨迹与实际轨迹的误差对比图如图 3 所示。

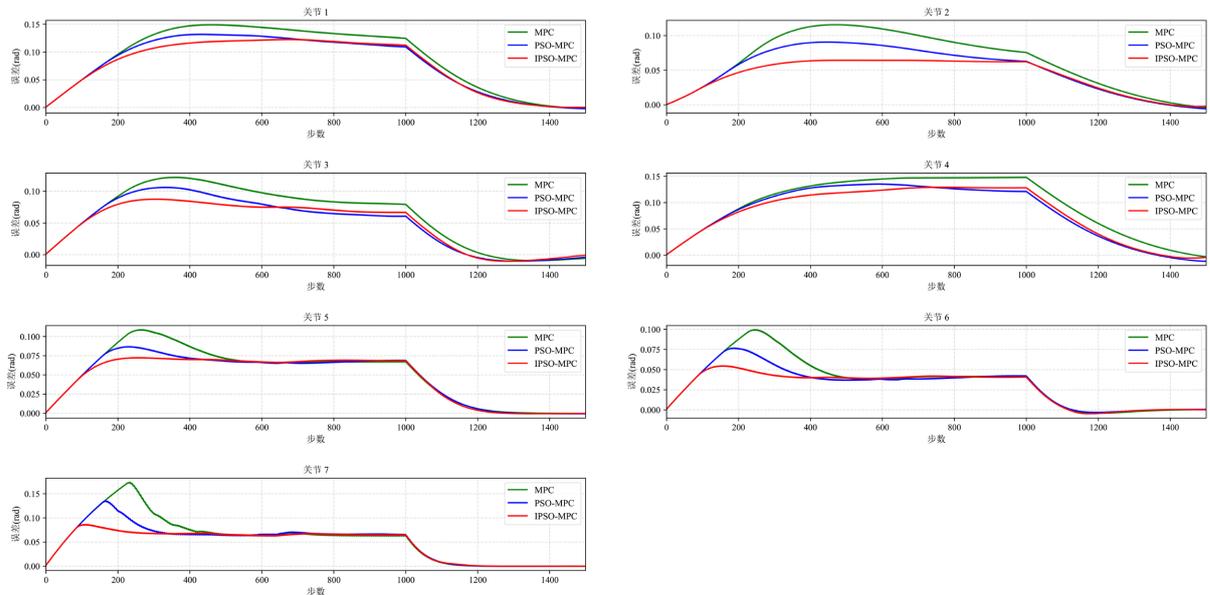


Figure 3. Comparison of trajectory tracking errors in joint space
图 3. 关节空间下轨迹跟踪误差对比图

图 3 展示了机械臂在关节空间内的给定轨迹运动跟随控制，即轨迹跟踪误差情况，纵轴单位为弧度制的关节角度差。图中绿蓝红三种颜色的线条分别表示原版 MPC 控制器、PSO-MPC 控制器以及 IPSO-MPC 控制器下的七个关节的误差。图中可以清晰观察到本文所提出的 IPSO-MPC 在轨迹跟踪过程中表现出了最小的跟踪误差。尤其是在像二关节这种承重关节的误差中表现更为出色，粒子群寻优在该关节的控制中发挥了更显著的作用，其能够更有效地搜索到针对二关节的最优控制参数组合，在面对二关节承重带来的复杂力学变化时，粒子群可以快速调整搜索方向，找到更合适的控制策略，以减少误差积累，相较于原版 MPC 和 PSO-MPC，IPSO-MPC 在第二关节的误差接近降低了一倍。同时，在整体关节误差调节过程中，IPSO-MPC 表现也很出色。IPSO-MPC 在 200 步左右合理的调节了权重系数，避免了另外两组出现的误差峰值，整体跟踪效果更加稳定，以较小的误差完成了整个跟踪任务。

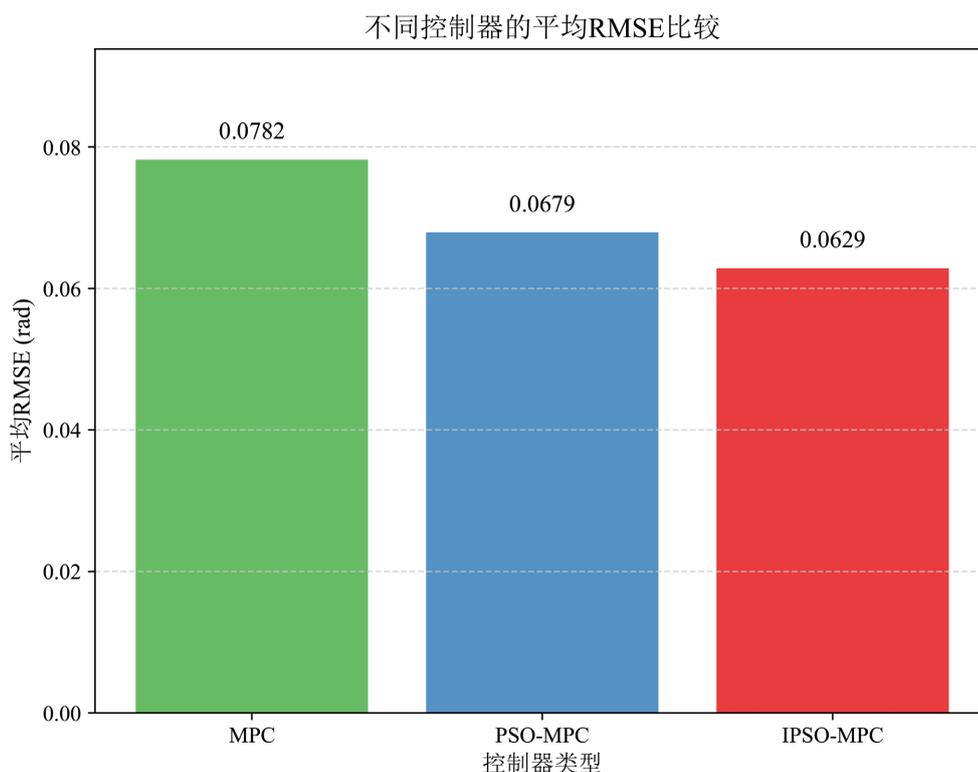


Figure 4. Mean square error comparison histogram

图 4. 均方误差对比柱状图

基于公式(13)计算跟踪任务过程中，各实验组的均方误差也能体现出 IPSO-MPC 的优势。图 4 展示了三个实验组的均方误差可视化柱状图。IPSO-MPC 相较于原版 MPC 与 PSO-MPC 分别降低了 33.85%、23.08%，改进后的 IPSO-MPC 具有更强的抗外界干扰能力以及自适应调节能力，跟踪效果也是得到了显著提升。

5. 结论

本文围绕轨迹跟踪精度问题展开深入研究，创新性地提出了一种基于 MPC 的增量式扭矩控制器，专门用于轨迹跟踪任务。为优化控制器性能，引入 PSO 优化算法对 MPC 控制器参数进行实时调整，以保障系统在动态环境下的控制误差精度。针对传统 PSO 算法易陷入局部最优解及收敛速度慢的固有缺陷，采用线性惯性权重下降策略，有效提升了算法性能。通过在 MuJoCo 仿真平台开展对照实验，获得了具

有说服力的结果。相较于原版 MPC, IPSO-MPC 的关节角度均方误差显著降低, 提升幅度达 33.85%; 与传统 PSO 优化的 MPC 相比, 也有 23.08% 的提升。这充分表明, 所提出的 IPSO-MPC 控制器能够凭借权重系数的灵活调整, 实现对动态环境的实时、精准响应, 展现出卓越的轨迹跟踪精度和强大的抗干扰能力, 极大地提升了算法在实际应用中的有效性。综上所述, 本研究成果不仅在理论层面完善了轨迹跟踪控制算法, 在实践应用中也具有重要价值。未来, 该方法有望在遥操作从端设备控制等复杂场景中广泛应用, 为确保从臂的高精度轨迹跟踪提供可靠解决方案, 推动相关领域技术的进一步发展。

参考文献

- [1] Hewing, L., Wabersich, K.P., Menner, M. and Zeilinger, M.N. (2020) Learning-Based Model Predictive Control: Toward Safe Learning in Control. *Annual Review of Control, Robotics, and Autonomous Systems*, **3**, 269-296. <https://doi.org/10.1146/annurev-control-090419-075625>
- [2] Yan, J., Jin, L., Yuan, Z. and Liu, Z. (2022) RNN for Receding Horizon Control of Redundant Robot Manipulators. *IEEE Transactions on Industrial Electronics*, **69**, 1608-1619. <https://doi.org/10.1109/tie.2021.3062257>
- [3] Zhu, T., Mao, J., Han, L., Zhang, C. and Yang, J. (2024) Real-Time Dynamic Obstacle Avoidance for Robot Manipulators Based on Cascaded Nonlinear MPC with Artificial Potential Field. *IEEE Transactions on Industrial Electronics*, **71**, 7424-7434. <https://doi.org/10.1109/tie.2023.3306405>
- [4] Tang, Q., Chu, Z., Qiang, Y., Wu, S. and Zhou, Z. (2020) Trajectory Tracking of Robotic Manipulators with Constraints Based on Model Predictive Control. 2020 17th International Conference on Ubiquitous Robots (UR), Kyoto, 22-26 June 2020, 23-28. <https://doi.org/10.1109/ur49135.2020.9144943>
- [5] Marini, F. and Walczak, B. (2015) Particle Swarm Optimization (PSO). A Tutorial. *Chemometrics and Intelligent Laboratory Systems*, **149**, 153-165. <https://doi.org/10.1016/j.chemolab.2015.08.020>
- [6] 周燕, 刘培玉, 赵静, 王乾龙. 基于自适应惯性权重的混沌粒子群算法[J]. 山东大学学报(理学版), 2012, 47(3): 27-32.
- [7] 段书用, 李昌洛, 韩旭, 刘桂荣. 机械臂动力学分析及关节非线性摩擦模型建立[J]. 机械工程学报, 2020, 56(9): 18-28.
- [8] 刘帅, 许媛媛, 闫锋刚, 金铭. 泰勒展开与交替投影最大似然结合的离网格 DOA 估计算法[J]. 电子与信息学报, 2024, 46(8): 3219-3227.
- [9] Najafqolian, M.A., Alipour, K., Mousavifard, R. and Tarvirdizadeh, B. (2024) Control of Aerial Robots Using Convex QP LMPC and Learning-Based Explicit-MPC. *IEEE Transactions on Industrial Informatics*, **20**, 10883-10891. <https://doi.org/10.1109/tii.2024.3397353>
- [10] Liu, H., Fan, A., Li, Y., Bucknall, R. and Chen, L. (2024) Hierarchical Distributed MPC Method for Hybrid Energy Management: A Case Study of Ship with Variable Operating Conditions. *Renewable and Sustainable Energy Reviews*, **189**, Article ID: 113894. <https://doi.org/10.1016/j.rser.2023.113894>
- [11] Zhou, H., Huang, Y., Zhao, Y. and Lu, Y. (2024) Robopal: A Simulation Framework Based Mujoco. <https://github.com/NoneJou072/robopal>