

基于改进蝴蝶算法的机械臂时间最优轨迹规划

周明月, 周明伟, 刘桂岐, 程超

引用本文

周明月, 周明伟, 刘桂岐, 程超. [基于改进蝴蝶算法的机械臂时间最优轨迹规划](#)[J]. 计算机科学, 2023, 50(11A): 220900284-8.

ZHOU Mingyue, ZHOU Mingwei, LIU Guiqi, CHEN Chao. [Time Optimal Trajectory Planning of Manipulator Based on Improved Butterfly Algorithm](#) [J]. Computer Science, 2023, 50(11A): 220900284-8.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[基于终态神经网络的冗余机械臂重复运动规划](#)

Repeatable Motion Planning of Redundant Manipulators Based on Terminal Neural Networks
计算机科学, 2018, 45(12): 201-205. <https://doi.org/10.11896/j.issn.1002-137X.2018.12.033>

[机械臂最优运动规划问题的混合粒子群算法](#)

Hybrid Particle Swarm Algorithm for Optimal Motion Planning of Robot Arms
计算机科学, 2009, 36(11): 232-234.

基于改进蝴蝶算法的机械臂时间最优轨迹规划

周明月 周明伟 刘桂岐 程超

长春工业大学计算机科学与工程学院 长春 130000

(zmyjlu@ccut.edu.cn)

摘要 机械臂在轨迹规划过程中,为使驱动装置符合实际负载要求,各关节速度和加速度在选取上会相对保守,导致完成一套动作需要的时间过长,从而使基于运动速度和加速度的机械臂的连续性和平稳性未能充分发挥。为解决机械臂各个关节速度和加速度的优化问题,提出一种基于改进蝴蝶算法的机械臂时间最优轨迹规划方法。首先,利用3-5-3多项式插值算法对AUBO-i5六自由度机械臂构造机械臂运动轨迹,然后,利用引入莱维飞行和正弦余弦算法的蝴蝶优化算法对运动轨迹进行时间优化,在满足工作需求的前提下,减少机械臂的运行时间。仿真结果表明,改进的蝴蝶算法与同类算法相比,不易陷入局部最优,且具有更高的寻优精度。将改进的蝴蝶算法应用在轨迹规划中,机械臂的运动时间大大缩短,可保证机械臂在实际生产中平稳高效地完成工作。

关键词: 机械臂; 时间最优轨迹规划; 改进的蝴蝶算法; 3-5-3多项式

中图分类号 TP241

Time Optimal Trajectory Planning of Manipulator Based on Improved Butterfly Algorithm

ZHOU Mingyue, ZHOU Mingwei, LIU Guiqi and CHEN Chao

School of Computer Science and Engineering, Changchun University of Technology, Changchun 130000, China

Abstract Aiming to plan the trajectory of the manipulator, common practice makes the drive device meet the actual load requirements, the velocity and acceleration of each joint will be relatively conservative in the selection. As a result, it takes too long to complete a set of actions, so that the continuity and smoothness of the manipulator based on motion speed and acceleration cannot be fully exerted. In order to solve the problem of optimal selection of the velocity and acceleration of each joint of the manipulator, a time-optimal trajectory planning method of the manipulator based on the improved butterfly algorithm is proposed. Firstly, the 3-5-3 polynomial interpolation algorithm is used to construct the motion trajectory of the robotic arm of AUBO-i5 six-degree-of-freedom manipulator. Then, the butterfly optimization algorithm introducing Levi flight and sine cosine algorithm is used to optimize the time of the motion trajectory, which reduces the running time of the robotic arm on the premise of meeting the work requirements. Simulation results show that, compared to similar algorithms, the improved butterfly algorithm is not easily trapped in local optimization and has higher optimization accuracy. The improved butterfly algorithm is applied to trajectory planning, and the run times of the robotic arm is significantly reduced, which can ensure that the robotic arm can the task smoothly complete and efficiently in actual production.

Keywords Robotic arm, Time optimal trajectory planning, Improved BOA, 3-5-3 polynomial

1 引言

随着人工智能、视觉计算、深空探测等技术的不断发展以及更高层次的市场需求,作为工业机器人产业最新研究方向的机械臂技术正在快速发展^[1]。如今,机械臂已经应用于各个领域的自动化生产中,特别是在装配、焊接、测试、上下料等现代化生产过程中都能看到其身影^[2]。轨迹规划^[3]作为机械臂控制的基础,可以按照作业任务要求,对机械臂运动的时间、速度和路径点进行设计,以完成各项任务。机械臂的工作效率和可靠性决定了工作完成的质量,所以,对机械臂进行轨迹规划研究,可减少机械臂的内部损耗,提高机械臂的工作效率。

近年来,针对机械臂的轨迹规划问题,研究人员做了大量研究^[4-5]。按照规划所在空间不同,轨迹规划可分为关节空间轨迹规划和笛卡尔空间轨迹规划。关节空间是指机械臂关节运动所构成的空间,而笛卡尔空间是指机械臂末端工具平移和旋转运动所构成的空间。这两种规划方式都需要根据逆运动学求解出各个关节的位置信息,从而进行轨迹规划。其中,笛卡尔空间的轨迹规划存在奇异性,会在某些方向失去运动能力,故本文选取关节空间轨迹规划。机械臂在关节空间的轨迹规划^[6-9]对应点到点的运动,即从一个点运动到另一个点,需要将机械臂的关节变量表示为只与时间有关的函数,然后对其角度、角速度和角加速度进行约束。Sabarigirish等采用三次多项式插值算法对五自由度机械臂进行轨迹规划,

基金项目:吉林省教育厅重点项目(JKH20210754KJ)

This work was supported by the Key Projects of Jilin Provincial Department of Education(JKH20210754KJ).

通信作者:刘桂岐(liuguiqiay@126.com)

保证了机械臂的平稳运动,但是存在关节加速度不连续的问题^[7];Fang 等应用五次多项式插值算法对机械臂进行轨迹规划,解决了三次多项式插值算法中关节加速度不连续的问题,得到了连续光滑的运动曲线^[8];Porawagama 等将三次和五次多项式结合,运用 5-3-5 次多项式插值算法,使生成的运动轨迹在位置、速度、加速度上连续可微^[9]。轨迹优化^[10-13]在轨迹规划算法的基础上提升了运动轨迹的质量,改善了运动轨迹的性能,更加符合实际生产要求。Wang 等在鲸鱼算法和差分进化算法的基础上,提出改进的鲸鱼优化算法,通过减小加速度,实现加速度和速度的连续性,并保持机械臂运动的平稳性^[10];Zhao 等提出了一种改进的鲸鱼优化和粒子群优化混合方法,通过优化机械臂的运动轨迹,降低了运动中的加速度,并提高了机械臂的工作效率^[11];Yu 等利用五次多项式函数建立各关节的位置信息,并采用改进的自适应遗传算法对机械臂的各个关节插补点的时间间隔进行优化,实现了时间最优的轨迹规划^[12];Wang 等提出了一种基于改进布谷鸟搜索算法的 3-5-3 多项式插值轨迹规划算法,消除了不满足速度约束的任何插补时间,缩短了机械臂的运行时间^[13]。

改进的智能算法优化了机械臂的运行时间,但存在易陷入局部最优的问题。为此,本文提出了一种基于改进蝴蝶算法的机械臂时间最优轨迹规划算法,将莱维飞行和正弦余弦算法引入蝴蝶优化算法中,改善了轨迹规划算法的寻优能力,优化了机械臂的运行时间。

2 AUBO-i5 机械臂运动学模型

由于 AUBO-i5 型机械臂占用体积小,重量轻,能够快速安装,可与人或其他自动化设备共同协作,所以本文选取 AUBO-i5 型机械臂作为模型来研究机械臂的时间最优轨迹规划算法。图 1 为 AUBO-i5 机械臂的关节示意图。如图 1 所示,AUBO-i5 机械臂有 6 个关节,每一个关节即为一个自由度,其中三个自由度(关节 4、关节 5 和关节 6)可确认机械臂末端的位置信息,另外 3 个自由度(关节 1、关节 2 和关节 3)可确定机械臂的位姿信息,AUBO-i5 机械臂的 6 个自由度可满足机械臂在三维空间内以任意姿势到达可操作范围内的任意位置。根据机械臂出厂初始参数设置,AUBO-i5 机械臂的 DH(Denavit Hartenberg)参数如表 1 所列。

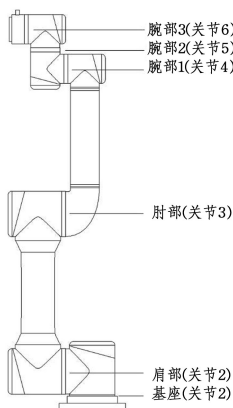


图 1 AUBO-i5 机械臂的关节示意图

Fig. 1 Schematic diagram of joints of AUBO-i5 robotic arm

表 1 AUBO-i5 机械臂的 DH 参数

Table 1 DH parameters of AUBO-i5 robotic arm

连杆 i	关节角 $\theta_i/(^\circ)$	连杆偏移量 d_i/cm	连杆长度 a_i/cm	扭转角 $\alpha_i/(^\circ)$
1	180	9.85	0	0
2	-90	12.15	0	-90
3	0	0	40.8	180
4	-90	0	37.6	180
5	0	10.25	0	-90
6	0	9.4	0	90

AUBO-i5 机械臂具有 6 个自由度,所以可以用 \mathbf{T} 来表示机械臂的末端位姿和位置信息。其中矢量 $\mathbf{n}, \mathbf{o}, \mathbf{a}$ 组成 3 阶矩阵来表示机械臂末端的位姿信息,矢量 \mathbf{p} 则表示机械臂末端的位置信息。利用表 1 中的 DH 参数,可以求得机械臂的连杆变换矩阵,机械臂末端坐标系与基坐标系的变换矩阵 \mathbf{T} 可表示为:

$${}^0\mathbf{T} = \prod_{i=1}^6 {}^{i-1}\mathbf{T} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

其中,矩阵 \mathbf{T} 的前置上标表示被转换的坐标系,前置下标表示目的坐标系, (n_x, n_y, n_z) 表示机械臂末端执行器的法向矢量, (o_x, o_y, o_z) 表示机械臂末端执行器的滑动矢量, (a_x, a_y, a_z) 表示机械臂末端执行器的接近矢量, (p_x, p_y, p_z) 表示机械臂末端执行器的位置矢量,齐次变换矩阵 ${}^{i-1}\mathbf{T}$ 表示连杆 i 的坐标系相对于连杆 $i-1$ 的坐标系的位姿,公式如下:

$${}^{i-1}\mathbf{T} = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

其中, $c = \cos, s = \sin, \alpha_i$ 表示连杆 i 的关节角, d_i 表示连杆 i 的连杆偏移量, a_i 表示连杆 i 的连杆长度。

在机械臂的轨迹规划中,单独使用三次多项式插值算法进行轨迹规划只能保证角度和角速度的连续性,无法约束加速度,可能会导致加速度突变,损耗机械臂的使用寿命。单独使用五次多项式插值算法进行轨迹规划虽然可以约束加速度,解决突变问题,但速度和加速度可能会很大,导致脱离电机的驱动范围。因此,本文选择结合三次和五次多项式插值算法来优化机械臂的运动轨迹。

选取 4 个路径点 x_{i1}, x_{i2}, x_{i3} 和 x_{i4} ,相邻的两点组成 1 个时间段,3 个时间段可表示为: $0-t_1, t_1-t_2, t_2-t_3$,其中 $0-t_1$ 为第一段时间,应用三次多项式插值算法, t_1-t_2 为第二段时间,应用五次多项式插值算法, t_2-t_3 为第三段时间,应用三次多项式插值算法,如图 2 所示。

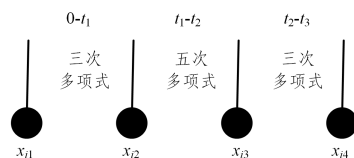


图 2 时间间隔图

Fig. 2 Time interval graph

建立各关节变量对时间 t 的函数表达式。三段时间中关节位置与时间的函数可表示为:

$$S_{i1}(t) = d_{i13}t^3 + d_{i12}t^2 + d_{i11}t + d_{i10} \quad (3)$$

$$S_{i2}(t) = d_{i25}(t-t_1)^5 + d_{i24}(t-t_1)^4 + d_{i23}(t-t_1)^3 + d_{i22}(t-t_1)^2 + d_{i21}(t-t_1) + d_{i20} \quad (4)$$

$$S_{i3}(t) = d_{i33}(t-t_1-t_2)^3 + d_{i32}(t-t_1-t_2)^2 + d_{i31}(t-t_1-t_2) + d_{i30} \quad (5)$$

分别对式(3)一式(5)求一阶导数,可得到关节的速度与时间的函数表达式:

$$\dot{S}_{i1}(t) = 3d_{i13}t^2 + 2d_{i12}t + d_{i11} \quad (6)$$

$$\dot{S}_{i2}(t) = 5d_{i25}(t-t_1)^4 + 4d_{i24}(t-t_1)^3 + 3d_{i23}(t-t_1)^2 + 2d_{i22}(t-t_1) + d_{i21} \quad (7)$$

$$\dot{S}_{i3}(t) = 3d_{i33}(t-t_1-t_2)^2 + 2d_{i32}(t-t_1-t_2) + d_{i31} \quad (8)$$

分别对式(3)一式(5)求二阶导数,可得到关节的加速度与时间的函数表达式:

$$\mathbf{A} = \begin{bmatrix} t_1^3 & t_1^2 & t_1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 3t_1^2 & 2t_1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 6t_1 & 2 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & t_2^5 & t_2^4 & t_2^3 & t_2^2 & t_2 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 5t_2^4 & 4t_2^3 & 3t_2^2 & 2t_2 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 20t_2^3 & 12t_2^2 & 6t_2 & 2 & 0 & 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t_3^3 & t_3^2 & t_3 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3t_3^2 & 2t_3 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6t_3 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (12)$$

$$\mathbf{X} = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ x_{i4} \ 0 \ 0 \ x_{i1} \ 0 \ 0 \ x_{i3} \ x_{i2}]^T \quad (13)$$

$$\mathbf{Y} = [D_1 \ D_2 \ D_3]^T \quad (14)$$

$$D_1 = [d_{i13} \ d_{i12} \ d_{i11} \ d_{i10}] \quad (14a)$$

$$D_2 = [d_{i25} \ d_{i24} \ d_{i23} \ d_{i22} \ d_{i21} \ d_{i20}] \quad (14b)$$

$$D_3 = [d_{i33} \ d_{i32} \ d_{i31} \ d_{i30}] \quad (14c)$$

其中, $(x_{i1}, x_{i2}, x_{i3}, x_{i4})$ 分别表示第 i 个关节的 4 个路径点。

3 基于时间最优的机械臂轨迹优化算法

3.1 蝴蝶优化算法

Arora 等^[14]于 2019 年提出蝴蝶优化算法(Butterfly Optimization Algorithm, BOA),作为一种基于自然规律的优化算法,其主要模拟了蝴蝶搜索食物和交配的行为,利用蝴蝶通过气味的大小来确定食物和交配对象位置的自然规律,以得到优化问题最优解的位置。气味表征的是物理刺激,其公式如下:

$$f = cI^\alpha \quad (15)$$

其中, f 表示气味的大小, α 表示指数因子, c 表示感觉因子, α 和 c 的取值范围为 $[0, 1]$, I 表示根据当前环境下蝴蝶个体计算得到的适应度函数,当某个蝴蝶个体的适应度值较高时,其他个体就会向它移动。当 $\alpha=1$ 时,表示香味在传播时没有损耗,所传出的香味会被其他蝴蝶个体全部接受。 c 可以影响算法的收敛速度, c 值越大,表示蝴蝶会尽可能搜索解空间,同时收敛速度下降。

蝴蝶算法包含两个阶段:全局搜索阶段和局部搜索阶段。搜索方式的选择主要取决于转换概率 P 的取值,搜索方式的

$$\ddot{S}_{i1}(t) = 6d_{i13}t + 2d_{i12} \quad (9)$$

$$\ddot{S}_{i2}(t) = 20d_{i25}(t-t_1)^3 + 12d_{i24}(t-t_1)^2 + 6d_{i23}(t-t_1) + 2d_{i22} \quad (10)$$

$$\ddot{S}_{i3}(t) = 6d_{i33}(t-t_1-t_2) + 2d_{i32} \quad (11)$$

其中, $S_{i1}(t), S_{i2}(t), S_{i3}(t)$ 分别表示机械臂第 i 个关节在第 1, 2, 3 段时间内 t 时刻的关节位置; $\dot{S}_{i1}(t), \dot{S}_{i2}(t), \dot{S}_{i3}(t)$ 分别表示机械臂第 i 个关节在第 1, 2, 3 段时间内 t 时刻的关节速度; $\ddot{S}_{i1}(t), \ddot{S}_{i2}(t), \ddot{S}_{i3}(t)$ 分别表示机械臂第 i 个关节在第 1, 2, 3 段时间内 t 时刻的关节加速度,多项式中的系数可根据 $\mathbf{Y} = \mathbf{X} \cdot \mathbf{A}^{-1}$ 求得, \mathbf{A} 为关于时间 t 的函数矩阵, \mathbf{X} 为插值点矩阵, \mathbf{Y} 表示系数矩阵,如式(12)一式(14)所示。

选取方法如下:

(1) 选取一个 $0 \sim 1$ 的随机数 r , 如果 $r < P$, 则采用全局搜索方式, 蝴蝶感知到气味值较大的蝴蝶并向其靠近, 通过函数可表示为:

$$x_i^{t+1} = x_i^t + (r^2 * g^* - x_i^t) * f_i \quad (16)$$

(2) 选取一个 $0 \sim 1$ 的随机数 r , 如果 $r > P$, 则采用局部搜索方式, 蝴蝶没有感受到气味, 将会在搜索空间内随机移动, 通过函数可表示为:

$$x_i^{t+1} = x_i^t + (r^2 * x_k^t - x_j^t) * f_i \quad (17)$$

其中, x_k^t 和 x_j^t 分别表示在第 t 代时随机选取的第 k 只和第 j 只蝴蝶的位置, r 表示 $[0, 1]$ 的随机数, f_i 表示第 i 只蝴蝶的气味。

蝴蝶个体在觅食的过程中, 局部搜索和全局搜索均可发生, 因此, 蝴蝶算法可利用切换概率 P 在局部搜索和全局搜索之间切换。当未达到停止条件时, 一直进行迭代, 当达到最大迭代次数时, 则输出最优值。迭代结束的标志为最大迭代次数(这里设置的是 500)。

3.2 基于改进蝴蝶算法的机械臂轨迹规划方法

3.2.1 基于莱维飞行的个体位置更新方法

传统蝴蝶算法在机械臂的轨迹优化中, 存在寻优精度低、易陷入局部最优解的问题。针对该问题, 本文提出将莱维飞行算法^[15]引入蝴蝶算法中, 利用其搜索的随机性增强蝴蝶个体的鲁棒性, 提高算法整体的搜索能力。改进后的蝴蝶个体拥有更强的活力, 可以对所在空间进行充分的搜索, 提高了寻优精度, 有效地缩短了机械臂的运行时间。

莱维飞行是法国的数学家 Levy 提出的一种随机搜索路径行为。莱维飞行是一个点在任意空间内进行任意距离的随机游走过程,应用莱维飞行的随机性可以避免蝴蝶算法陷入局部最优。莱维飞行中的方差增长速度较快,所以在搜索过程中具有明显的优势。 $levy(s)$ 表示随机搜索的路径,其表达式如下:

$$levy(s) = \frac{\lambda \Gamma(\lambda) \sin\left(\frac{\pi\lambda}{2}\right)}{\pi} \cdot \frac{1}{s^{\lambda+1}} \quad (18)$$

其中, λ 表示指数参数; α 取值为 1; Γ 表示 gamma 函数; s 表示莱维飞行的步长,其公式如下:

$$s = \frac{u}{|v|^{\frac{1}{\beta}}} \quad (19)$$

其中, β 表示 $[0, 2]$ 的随机数, 实验中设置为 1.5, $u \sim N(0, \sigma_u^2)$, $v \sim N(0, \sigma_v^2)$, σ_v 和 σ_u 表示服从正态分布的标准差, 且 $\sigma_v = 1$ 。

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta) \cdot \sin(\pi\beta/2)}{\Gamma[(1+\beta)/2] \cdot \beta \cdot 2^{(\beta-1)/2}} \right\}^{1/\beta} \quad (20)$$

将莱维飞行应用到蝴蝶算法的全局搜索和局部搜索阶段。该算法的随机性可有效避免蝴蝶算法陷入到局部最优, 最大程度地利用机械臂的速度和加速度, 极大地缩短机械臂的运行时间。

全局搜索的公式更改为:

$$x_i^{t+1} = x_i^t + (r^2 * g^* - x_i^t) * levy(s) * f_i \quad (21)$$

其中, x_i^t 表示第 i 个个体在第 t 次迭代的解向量, r 表示 $[0, 1]$ 的随机数, g^* 表示当前最优的个体, f_i 表示第 i 个个体的香味浓度。

局部搜索的公式更改为:

$$x_i^{t+1} = x_i^t + (r^2 * x_k^t - x_j^t) * levy(s) * f_i \quad (22)$$

其中, x_k^t 和 x_j^t 分别表示在第 t 次迭代的两个随机个体的位置。

3.2.2 基于正余弦算法的个体位置更新方法

传统蝴蝶算法在机械臂的轨迹优化的过程中, 存在搜索能力下降且迭代后期多样性下降, 易陷入局部最优解的问题。针对此类问题, 本文将正余弦算法^[16]引入蝴蝶算法的局部搜索阶段。在搜索过程中, 将正余弦算法看作一个优化算子, 使其能避免算法在迭代后期陷入局部最优解。

正余弦算法是澳大利亚学者 Seyedali Mirjalili 于 2016 年提出的一种优化算法。该算法利用正弦和余弦函数特性, 通过其自适应性改变正余弦函数的振动幅度, 从而平衡算法的搜索能力和开发能力, 并找到最优解。正余弦算法的函数表达式如下:

$$x_i^{t+1} = \begin{cases} x_i^t + R_1 \times \sin(R_2) \times |R_3 \times M_i^t - x_i^t|, & R_4 \leq 0.5 \\ x_i^t + R_1 \times \cos(R_2) \times |R_3 \times M_i^t - x_i^t|, & R_4 > 0.5 \end{cases} \quad (23)$$

其中, x_i^t 表示第 t 次迭代的第 i 个个体的位置, M_i^t 为第 t 次迭代的第 i 个个体的最优解, 即机械臂的运行时间, R_1, R_2, R_3, R_4 这 4 个参数影响着第 i 个个体在下次迭代位置的更新方向。 R_1 表示线性递减函数, 它的取值决定了个体的搜索范围; R_2 表示 $[0, 2\pi]$ 的随机数, 它影响着个体在移动中的步长, 即搜索最优时间的效率; R_3 表示 $[-2, 2]$ 的随机数, 只有当 R_3 的数值大于 1 时, 才会对下次迭代中的位置产生影响; R_4 表示 $[0, 1]$ 的随机数, 它影响着正弦和余弦两种位置更新方式的选择。

3.3 机械臂时间最优轨迹规划

由式(12)可以看出, 搜索最优值仅与时间 t 有关, 因此将 3-5-3 次多项式插值的时间 t 作为寻优变量, 在运行中需要满足其速度、加速度和加加速度的约束条件, 则优化的目标函数可表示为:

$$f(t) = \min \sum_{i=0}^n (t_{i1} + t_{i2} + t_{i3}) \quad (24)$$

$$s. t. |v_i| \leq v_i^s \quad (24a)$$

$$v_i(0) = v_i(t_f) = 0 \quad (24b)$$

$$|r_i| \leq r_i^s \quad (24c)$$

$$r_i(0) = r_i(t_f) = 0 \quad (24d)$$

$$|j_i| \leq j_i^s \quad (24e)$$

$$i = 1, 2, \dots, 6 \quad (24f)$$

其中, 约束条件为式(24a)、式(24c)和式(24e)。 v_i, r_i 和 j_i 分别代表第 i 个关节的实际速度、加速度和加加速度, v_i^s, r_i^s 和 j_i^s 代表第 i 个关节理论可以达到的最大值。式(24b)和式(24d)表示第 i 个关节的始末速度和加速度为 0。总而言之, 机械臂是以时间最短为优化目标, 其数学模型可以描述为式(24), t_{i1}, t_{i2}, t_{i3} 为机械臂关节在 3 段路径的运动时间。

为了保持机械臂运动轨迹的平稳性, 同时基于对变量的控制原则, 设定运动的初、末速度和加速度的数值为 0。计算过程中, 在满足速度、加速度和加加速度约束的前提下, 通过将插值时间与运动轨迹的约束条件相结合, 以运行时间最短作为优化目标, 则可求得机械臂各个关节的最佳运动轨迹。

3.4 算法整体流程

在研究过程中, 本文的主要目的是缩短机械臂的运行时间, 因此采用改进的蝴蝶算法对运行时间进行优化, 得到一个时间最优解。在算法中, 蝴蝶个体在搜索空间内通过释放不同程度的气味, 使个体间相互吸引并进行位置移动, 逐渐形成一个蝴蝶群体网络。最优解就是通过个体的不断搜索而产生, 算法流程如图 3 所示。

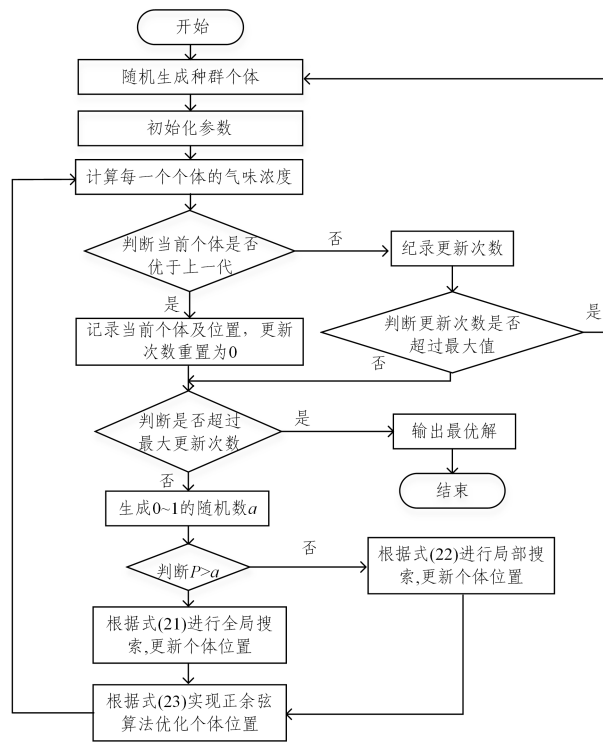


图 3 算法流程图

Fig. 3 Algorithm flow chart

基于改进蝴蝶算法的机械臂时间最优轨迹规划方法的算法仿真流程如算法 1 所示。

算法 1 基于改进蝴蝶算法的机械臂时间最优轨迹规划方法

- 步骤 1 初始化参数,如:蝴蝶个体数量、最大迭代次数、位置及速度;
 步骤 2 计算每只蝴蝶个体在 3-5-3 混合多项式中的时间(t_{t1} , t_{t2} , t_{t3}),并根据式(12)求解出多项式系数 a_{ij} ;
 步骤 3 根据式(6)一式(8)求解出各段时间内的关节速度,并判断是否满足速度约束要求,同时约束速度在规定范围之内;
 步骤 4 利用目标函数计算出每只蝴蝶个体所对应的适应度值,更新蝴蝶个体的位置和最优适应度,并获取最优个体的时间值,即机械臂运行的时间;
 步骤 5 判断是否满足最大迭代次数,若满足,则输出最优解;若不满足,则重复步骤 2—步骤 4。

4 仿真实验与分析

4.1 改进蝴蝶算法的性能分析

为验证改进蝴蝶算法的有效性,选取 6 个不同的基准函数: F_1-F_6 。其中, F_1-F_3 是单峰函数,用于验证算法的收敛

能力和精度, F_4-F_6 是复杂非线性的多峰函数,用于验证算法的探索寻优能力,表 2 列出了这 6 个函数的具体信息。在实验中,蝴蝶的种群大小设置为 30;最大迭代次数设置为 500;感觉因子 c 表示蝴蝶个体对气味的敏感程度,将其设置为 0.01;指数因子 α 表示气味传递的损耗,将其设置为 0.1; α 和 c 是两个常数,决定着机械臂时间最优解的搜索能力;转换概率 P 决定着机械臂最优时间解的搜索方式的切换,也影响着机械臂运行时间的优化程度,这里将 P 值设置为 0.8。

为验证改进蝴蝶算法(改进 BOA)的可行性,选取传统蝴蝶算法(BOA)、鲸鱼算法(Whale Optimization Algorithm, WOA)^[17]、飞蛾算法(Moth Optimization Algorithm, MFO)^[18]在同一轨迹下进行对比研究。在求解数据时,应用 BOA,WOA,MFO 和改进 BOA 算法分别测试每一个基准函数,采集 20 次运行数据,并取数据中的最优解、最差解、平均值和标准差值。最优解和最差解的值越小表示寻优精度越高,平均值和标准差的值越小说明寻优的稳定性越好,测试结果如表 3 所列。

表 2 基准函数
Table 2 Benchmark functions

函数编号	基准函数	公式	维度	范围	理想值
F_1	Sphere	$F_1(x) = \sum_{i=1}^n x_i^2$	30	$[-100, 100]$	0
F_2	Beale	$F_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	2	$[-4.5, 4.5]$	0
F_3	Cigar	$F_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	30	$[-100, 100]$	0
F_4	Step	$F_4(x) = \max_i \{ x_i , 1 \leq i \leq n \}$	30	$[-100, 100]$	0
F_5	Quartic function with noise	$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	$[-1.28, 1.28]$	0
F_6	Bohachevsky	$F_6(x) = \sum_{i=1}^n ([x_i + 0.5])^2$	2	$[-100, 100]$	0

表 3 基准函数测试结果
Table 3 Benchmark function testing results

函数	算法	指标			
		最优解	最差解	平均值	标准差
F_1	改进 BOA	3.26×10^{-212}	9.02×10^{-169}	4.52×10^{-170}	0.00
	BOA	1.13×10^{-11}	1.45×10^{-11}	1.29×10^{-11}	8.15×10^{-13}
	WOA	5.39×10^{-85}	1.52×10^{-71}	1.28×10^{-72}	3.48×10^{-72}
	MFO	3.59×10^{-33}	7.78×10^{-30}	1.63×10^{-30}	2.48×10^{-30}
F_2	改进 BOA	5.47×10^{-107}	3.21×10^{-83}	2.31×10^{-84}	7.17×10^{-84}
	BOA	1.14×10^{-9}	5.90×10^{-9}	3.38×10^{-9}	1.20×10^{-12}
	WOA	4.72×10^{-57}	1.01×10^{-48}	5.46×10^{-50}	2.20×10^{-49}
	MFO	3.19×10^{-20}	4.55×10^{-18}	6.35×10^{-19}	1.02×10^{-18}
F_3	改进 BOA	2.20×10^{-216}	6.22×10^{-146}	3.11×10^{-147}	1.36×10^{-146}
	BOA	1.02×10^{-11}	1.46×10^{-11}	1.25×10^{-11}	4.70×10^{-10}
	WOA	1.27×10^4	6.69×10^4	3.92×10^4	1.53×10^4
	MFO	1.09×10^{-9}	3.51×10^{-6}	6.55×10^{-7}	9.45×10^{-7}
F_4	改进 BOA	1.82×10^{-112}	6.07×10^{-80}	3.04×10^{-81}	1.32×10^{-80}
	BOA	5.64×10^{-9}	7.30×10^{-9}	6.23×10^{-9}	4.70×10^{-10}
	WOA	3.34	5.43×10^1	2.41×10^1	1.62×10^1
	MFO	2.65×10^{-5}	3.50	6.18×10^{-1}	9.86×10^1
F_5	改进 BOA	5.32×10^{-9}	1.61	1.31×10^{-1}	3.66×10^{-1}
	BOA	2.89×10^1	2.90×10^1	2.89×10^1	2.83×10^{-2}
	WOA	2.69×10^1	2.88×10^1	2.80×10^1	5.51×10^{-1}
	MFO	3.61×10^{-1}	1.21×10^2	2.13×10^1	3.19×10^1
F_6	改进 BOA	8.81×10^{-18}	8.65×10^{-8}	1.24×10^{-8}	2.04×10^{-8}
	BOA	4.00	6.66	5.68	6.30×10^{-1}
	WOA	1.32×10^1	1.59	5.08×10^{-1}	3.57×10^{-1}
	MFO	7.31×10^{-16}	3.02×10^{-28}	1.80×10^{-29}	3.16×10^{-13}

对于 6 个基准函数,改进 BOA 最优解均优于对比算法。其中对于 F_1 函数,改进 BOA 的最优值比其他 3 种算法的

最优值高出 100 多个数量级;对于 F_3 函数,改进 BOA 算法的最优值比其他 3 种算法的最优值高出 200 多个数量级;对于

F_4 函数,改进 BOA 的最优值比其他 3 种算法的最优值高出约 100 个数量级。对于 $F_1 - F_4$ 函数,改进 BOA 的标准差最小;对于 F_5 和 F_6 函数,尽管其标准差不是最优,但与最优标准差值相差并不大。在 $F_1 - F_5$ 函数中,改进 BOA 算法的最差解和平均值均优于对比算法。由此证明,本文改进的蝴蝶算法相较于其他 3 种算法具有更好的寻优效果和稳定性。

为了进一步展示改进蝴蝶优化算法的寻优能力,给出了函数 $F_1 - F_6$ 的适应度迭代曲线,如图 4 所示。从图中可以看出,传统 BOA, WOA 和 MFO 算法在函数迭代时的速度

较慢,说明 3 种算法容易陷入局部最优解,而改进 BOA 在函数迭代时呈不断递减的趋势,直到收敛至最优,且具有更高的寻优精度。在图 4(e)和图 4(f)中,可明显看出传统 BOA 和 WOA 算法在近 50 代时可找到最优解,且随着进化代数的增加,结果基本稳定。而改进 BOA 在进化过程中出现多个拐点,说明该算法在更新解时,可以有效地跳出局部最优解。观察 6 个函数曲线可以得出,改进的 BOA 相较于其他 3 种算法可以稳定地搜索到最优解。由表 3 和图 5 结果可以证明,改进的 BOA 具有较好的寻优能力。

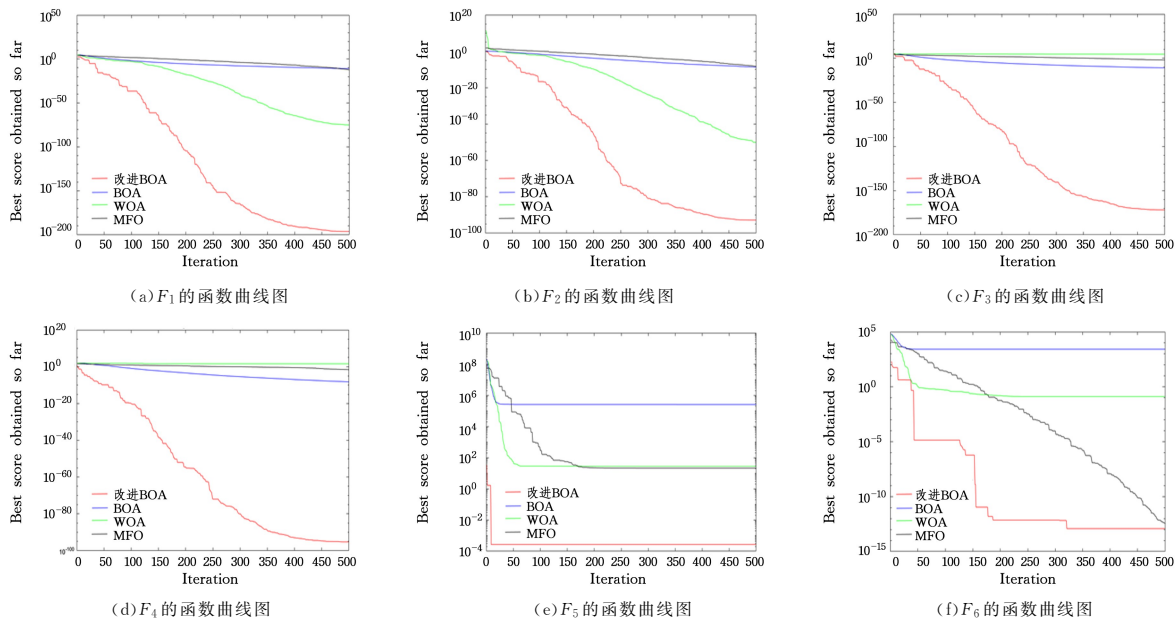


图 4 $F_1 - F_6$ 的函数迭代曲线图

Fig. 4 Function iteration curve of $F_1 - F_6$

4.2 时间最优轨迹规划算法的性能分析

利用 MATLAB2016b 编程软件进行时间最优轨迹规划的仿真实验。首先,给出 4 个路径点作为运动轨迹的插补点。然后,通过逆运动学求解将插补点从笛卡尔空间转化为关节空间,转化后的插补点如表 4 所列。最后,在满足运动学约束的条件下,对运动轨迹的时间进行优化。其中,关节位置、速度及加速度的约束条件如表 5 所列。

表 4 机械臂关节空间的插补点

Table 4 Interpolated points in the joint space of manipulator

关节	x	$\theta_{x1}/(^{\circ})$	$\theta_{x3}/(^{\circ})$	$\theta_{x4}/(^{\circ})$
1	80	-30	-50	
2	-45	-10	10	
3	15	100	50	
4	110	60	-100	
5	-60	50	-40	
6	80	30	20	

表 5 机械臂关节的约束条件

Table 5 Constraints of robot arm joints

关节数	速度/ $(^{\circ}/s^{-1})$	加速度/ $(^{\circ}/s^{-2})$	加加速度/ $(^{\circ}/s^{-3})$
1	100	45	60
2	95	40	60
3	100	75	55
4	150	80	70
5	130	90	75
6	110	80	70

关节的最大速度分别为 $46.7^{\circ}/s, 48.4^{\circ}/s, 48.0^{\circ}/s, 73.9^{\circ}/s, 56.9^{\circ}/s, 55.7^{\circ}/s$, 最大加速度分别为 $12.0^{\circ}/s^2, 13.9^{\circ}/s^2, 21.8^{\circ}/s^2, 29.3^{\circ}/s^2, 28.5^{\circ}/s^2, 30^{\circ}/s^2$ 。可见,未进行优化的机械臂运行速度和加速度是相对保守的,即使都在约束条件范围内,但机械臂的工作效率是极低的。

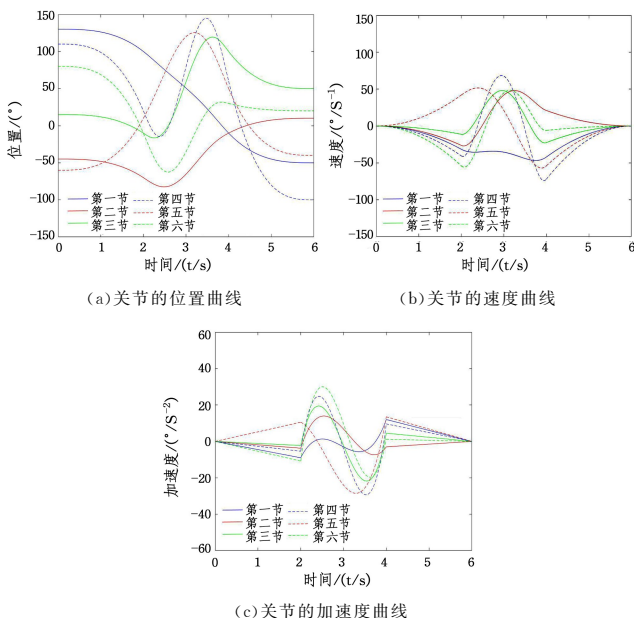


图 5 机械臂关节的运行曲线

Fig. 5 Running curve of joints of robotic arm

图 5 给出了未进行优化的机械臂关节位置、速度和加速度的运动曲线。其中,可以看到机械臂运行时间为 6s, 6 个

在改进的蝴蝶算法中,转换概率 P 是控制蝴蝶个体搜索范围大小的参数。当 P 值较大时,蝴蝶个体容易进行全局搜索,最优解趋于收敛,但是精度较差;当 P 值较小时,蝴蝶个体则容易进行局部搜索,最优解容易陷入局部最优,且搜索消耗较大。由此可见选取一个合适的 P 值非常重要。我们在不同的 P 值下,多次实验计算出机械臂的运行时间并选取最优值,如图 6 所示,当 P 值取 0.8 时,机械臂的运行时间最短。

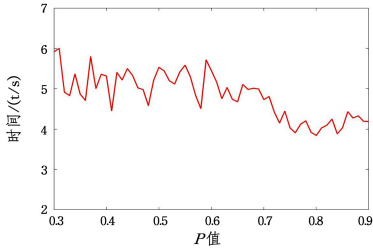


图 6 P 值与机械臂运行时间的关系曲线

Fig. 6 Relation curve between P value and running time of manipulator

可以看出迭代 100 次后,机械臂运行时间的最优解已经趋于稳定。优化后各个关节的位置、速度和加速度关于时间的函数曲线如图 8 所示,机械臂的各个关节可以在较短的时间内准确地到达所设定的路径点,运行时间也由原来的 6 s 缩短为 3.84 s,缩短了近 36%,达到了运行时间最优的效果。优化后的每个关节在运动开始时都有更高的加速度运动,这也保证了机械手能在短时间内达到低于极限速度的高速,保证了机器人的执行效率。整个运行过程连续,没有发生突变,并满足速度和加速度的约束条件,验证了本文方法的有效性。

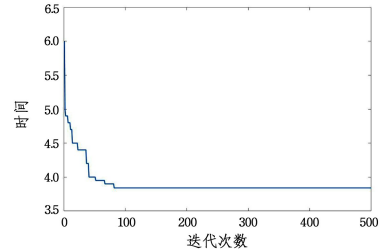
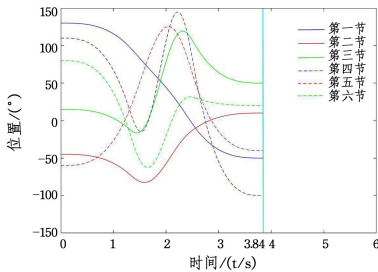


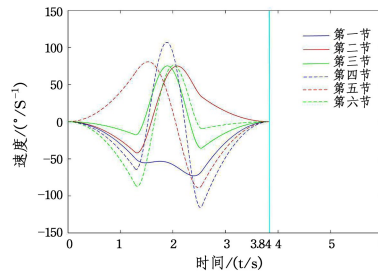
图 7 机械臂运行时间的优化曲线

Fig. 7 Running time optimization curve of manipulator

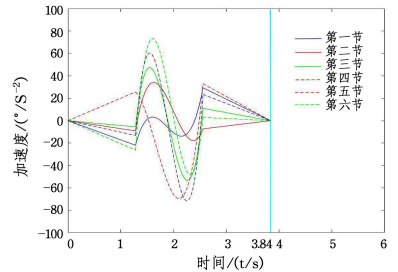
图 7 为改进 BOA 算法针对机械臂运行时间的优化曲线,



(a) 关节的位置曲线



(b) 关节的速度曲线



(c) 关节的加速度曲线

图 8 应用改进 BOA 算法的机械臂关节的运行曲线

Fig. 8 Running curve of joints of robotic arm using improved BOA algorithm

为验证改进 BOA 可以更好地优化机械臂的运行时间,分别选择 BOA, PSO 和 GA 这 3 种时间最优轨迹规划方法进行对比分析。其中 BOA 方法是利用传统蝴蝶算法的时间最优轨迹规划方法; PSO 方法是利用粒子群算法 (PSO) 的时间最优轨迹规划方法^[19]; GA 是利用遗传算法 (GA) 的时间最优轨迹规划方法^[20]。每个算法重复进行 10 次实验,分别取其最优值和平均值。实验结果如表 6 所列,与其他 3 种算法相比,本文提出的时间最优轨迹规划方法的运行时间最短。图 8 和表 6 的实验结果可以说明,利用本文提出的基于改进蝴蝶算法的机械臂时间最优轨迹规划方法,能够在平滑地完成动作的同时,减少机械臂的运行时间,为大规模工业生产中平稳高效的生产方式提供技术支撑。

表 6 优化算法的对比实验结果

Table 6 Comparative experimental results of optimization algorithms

算法	最优时间/s	平均时间/s
改进 BOA	3.84	3.84
BOA	4.18	4.21
PSO	3.94	4.01
GA	3.96	3.96

结束语 本文选择 AUBO-i5 六自由度机械臂作为模型,推导了机械臂在关节空间内应用 3-5-3 次多项式插值算法的轨迹规划方法。通过将莱维飞行和正余弦算法引入蝴蝶算法

中,改进了蝴蝶算法在迭代过程中个体位置的更新方式,以提高轨迹规划算法的寻优精度和速度。实验结果表明,改进的蝴蝶算法相比其他同类算法具有较高的寻优精度,并且能够较快地跳出最优解,且利用提出的轨迹规划算法得到的机械臂运动曲线满足运动学约束条件且连续,极大地缩短了机械臂的运行时间,更符合现代化产业对生产效率的要求。

在接下来的工作中,主要针对机械臂的稳定性和可靠性进行分析,根据机械臂的实际意义研究其内部控制。

参考文献

- [1] VARSHNEY V, KUMAR A, SAXENA T, et al. Design and development of intellectual robotic arm[C]// 2019 6th International Conference on Computing for Sustainable Global Development (INDIACom). 2019:984-987.
- [2] LI L, SHANG J Y, FENG Y L. Review of trajectory planning of articulated industrial robots[J]. Computer Engineering and Applications, 2018, 54(5): 36-50.
- [3] WEI B, SHU S H, ZHANG Y, et al. Kinematics analysis and trajectory planning simulation of manipulator based on AUBO-I5[C]// 2020 4th Annual International Conference on Data Science and Business Analytics (ICDSBA). 2020:72-75.
- [4] YUDHA H M, DEWI T, RISMA P, et al. Arm robot manipulator design and control for trajectory tracking; a review[C]// 2018 5th International Conference on Electrical Engineering, Compu-

- ter Science and Informatics(EECSI). 2018;304-309.
- [5] MA C Y,ZHANG Y,CHENG S. Time optimal trajectory planning based on redundant manipulator[C]// 2021 6th International Conference on Control, Robotics and Cybernetics, 2021: 125-129.
- [6] MEI T X,YANG Y,CHEN J B, et al. Simulation research on motion trajectory of PUMA 560 manipulator based on MATLAB[C]// 2019 Chinese Control And Decision Conference (CCDC). 2019;4857-4862.
- [7] SABARIGIRISH S,MIJA S J. Obstacle avoiding trajectory planning for 5 degree of freedom robot[C]//2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems(ICPEICES). 2016;1-5.
- [8] FANG S,MA X H,ZHAO Y, et al. Trajectory planning for seven-dof robotic arm based on quintic polynomial[C]//2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics(IHMSC),2019;198-201.
- [9] PORAWAGAMA C D,MUNASINGHE S R. Reduced jerk joint space trajectory planning method using 5-3-5 spline for robot manipulators[C]//7th International Conference on Information and Automation for Sustainability. 2014;1-6.
- [10] WANG T,XIN Z J,MIAO H B, et al. Optimal trajectory planning of grinding robot based on improved whale optimization algorithm[J]. Mathematical Problems in Engineering, 2020, 2020(4):1-8.
- [11] ZHAO J,ZHU X J,SONG T J. Serial manipulator time-jerk optimal trajectory planning based on hybrid IWOA-PSO algorithm[J]. IEEE Access, 2022, 10: 6592-6604.
- [12] YU H,MENG Q X,ZHANG J Y, et al. Time-optimal trajectory planning of robot based on improved adaptive genetic algorithm[C]// 2018 Chinese Control And Decision Conference. 2018: 6397-6402.
- [13] WANG W J,TAO Q,CAO Y T, et al. Robot time-optimal trajectory planning based on improved cuckoo search algorithm[J]. IEEE Access, 2020, 8: 86923-86933.
- [14] ARORA S,SINGH S. Butterfly optimization algorithm: a novel approach for global optimization[J]. Soft Computing, 2019, 23(3):715-734.
- [15] ZHENG J F,ZHAN H W, HUANG W, et al. An overview of the development of Lévy Flight and the application of intelligent optimization algorithms[J]. Computer Science, 2021, 48(2): 190-206.
- [16] GAO W X,LIU S,XIAO Z Y, et al. Butterfly optimization algorithm for global optimization[J]. Computer Application Research, 2020, 37(10): 2966-2970.
- [17] MIRJALILI S,LEWIS A. The whale optimization algorithm[J]. Advances in Engineering Software, 2016, 95: 51-67.
- [18] MIRJALILI S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm[J]. Knowledge-Based Systems, 2015, 89(11): 228-249.
- [19] GAO M Y,DING P, YANG Y X. Time-optimal trajectory planning of industrial robots based on particle swarm optimization[C]// 2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control(IMCCC). 2015:1934-1939.
- [20] LI G H,WANG Y L. Industrial robot optimal time trajectory planning based on genetic algorithm[C]// 2019 IEEE International Conference on Mechatronics and Automation(ICMA). 2019;136-140.



ZHOU Mingyue, born in 1980, Ph.D, associate professor. Her main research interests include nonlinear optimization and wireless communication.



LIU Guiqi, born in 1986, Ph. D. Her main research interests include wireless network location technology and so on.