

文章编号: 2095-2163(2022)02-0018-07

中图分类号: TP242.2

文献标志码: A

改进适应度函数的 CMA-ES 算法在机器人逆运动学中的应用

谭薪兴, 李光, 薛晨慷, 易静, 于权伟
(湖南工业大学机械工程学院, 湖南株洲 412007)

摘要: 针对机器人的逆运动学多解问题, 本文利用“最佳柔顺性”规则, 对适应度函数进行改进, 提出了一种自适应协方差矩阵进化策略(CMA-ES)算法求逆运动学解的新方法。在原算法的基础上, 将机器人各关节运动范围作为约束条件, 在工作空间中获得唯一且平稳光滑的路径。以 REBot-V-6R-6500 型 6 自由度机器人作为研究算例, 结果表明: 在点对点运动的逆运动学求解中, 代表位置误差的适应度函数的平均值稳定在 10^{-17} 数量级; 在空间螺旋轨迹连续跟踪的逆运动学求解中, 求解的各关节轨迹平滑且唯一, 代表位置误差的适应度函数的平均值稳定在 10^{-32} 数量级; 在点对点运动和连续轨迹跟踪的逆运动学求解中, 位置平均误差值均稳定在 10^{-16} m 数量级。

关键词: 机器人; 逆运动学; CMA-ES 算法; 最佳柔顺性

Application of CMA-ES algorithm with improved fitness function in solving inverse kinematics of robot

TAN Xinxing, LI Guang, XUE Chenkang, YI Jing, YU Quanwei

(College of Mechanical Engineering, Hunan University of Technology, Zhuzhou Hunan 412007, China)

[Abstract] Aiming at the multi-solution problem of inverse kinematics of robots, we follow the best compliance rule to improve the fitness function and propose a new method of Covariance Matrix Adaptation Evolution Strategy algorithm for inverse kinematics solutions. On the basis of the original algorithm, the motion range of each joint of the robot is taken as a constraint condition, and a unique, stable and smooth path is obtained in the working space. Taking the REBot-V-6R-6500 6-DOF robot as a research example, the results show that in the inverse kinematics solution of point-to-point motion, the average value of the fitness function representing the position error is stable at 10^{-17} level, the spiral trajectory of each joint in space of the inverse kinematics solution of tracking is smooth and unique, and the average value of the fitness function representing the position error is stable at 10^{-32} level. In the inverse kinematics solution of point-to-point motion and continuous trajectory tracking, the average position error value is stable at 10^{-16} m level.

[Key words] robot; inverse kinematics; CMA-ES algorithm; best compliance

0 引言

机器人运动学包括正向运动学和逆向运动学, 正向运动学是由一组关节角度确定机器人末端执行器一个姿态, 逆向运动学是由目标姿态求解各关节角度, 逆向运动学求解复杂且存在多解^[1-2]。

机器人逆向运动学求解是进行机器人轨迹规划、路径规划、避障等研究的基础^[3]。很多学者在该领域做了大量研究, 有许多逆向运动学的求解理论和方法, 传统的方法有代数法、几何法、数值法等。代数法主要以消元的方式将高维方程组转化为低维方程组, 还需要大量的三角代换, 计算过程十分复杂, 文献[1]提出基于分离重构技术, 降低了求解复杂度, 得到了逆运动学全部解; 几何法针对机器人的

特殊结构进行简化再求解, 一般无法单独使用甚至无法使用^[4]; 数值法可以精确搜索到一组逆解, 求解速度与初始值相关, 甚至存在无解, 需要的运算量大, 不适合实时性要求较高的场合^[5]。

近年来, 利用智能算法求解机器人逆向运动学的方法取得一定的成果。神经网络算法的计算代价高, 计算能力很大程度取决于数据的大小, 同时也取决于网络的深度和复杂程度; 遗传算法全局搜索能力强, 局部搜索能力较弱, 容易误入局部最优解; 蚁群算法和鱼群算法参数设置复杂, 如果参数设置不当, 容易偏离最优解; 粒子群算法操作简单, 收敛速度快, 在众多领域得到广泛应用, 但存在提前收敛、维数灾难等问题。

自适应协方差矩阵进化策略(Covariance Matrix

作者简介: 谭薪兴(1997-), 男, 硕士研究生, 主要研究方向: 机器人运动规划; 李光(1963-), 男, 博士, 教授, 硕士生导师, 主要研究方向: 机器人智能控制。

通讯作者: 李光 Email: liguang@hut.edu.cn

收稿日期: 2021-10-22

Adaptation Evolution Strategy, CMA-ES)是在进化策略(Evolution Strategy, ES)算法的基础上发展而来的无约束优化算法,在全局优化、多峰优化、多目标优化、大规模优化和结构性工程等领域得到了大量应用,该算法的搜索性能较高,并且能够达到比较高的搜索精度。文献[6]提出了改进的CMA-ES算法,在保留原算法的优点上,可以实时且高精度地求出逆向运动学解;文献[7]引入1/5成功理论,调整CMA-ES算法的规模和步长,提高了算法求解效率;上述智能算法提高了机器人逆向运动学求解的效率,但是忽略了机器人在实际运动过程中的平稳性和能耗。

本文提出了一种新的适应度函数用于CMA-ES算法求逆运动学解:与原算法相比,该算法将各关节运动范围作为约束条件,通过加权最小二乘法和位姿误差建立新的适应度函数,求解出唯一的一组运动学逆解。将该算法应用于REBot-V-6R-6500型六自由度机器人,仿真结果表明,该方法可以得到机器人高精度和平稳的逆运动学解。

1 CMA-ES 算法

1.1 算法原理

CMA-ES算法是一种进化策略类算法。经典ES算法寻找最优解主要依靠突变,但是如何调整突变的方向没有成熟的理论支撑,需要依据经验设置,因此该方法会导致无效的突变。为克服经典ES的不足,CMA-ES采用正态分布 $N(m, C)$ 生成一定数量的搜索种群, $N(m, C)$ 表示搜索分布均值为 m ,以协方差矩阵为 C 的多元正态分布,且 $C = BDB^T$,其中 $BB^T = I$, B 的列向量由 C 的特征向量正交基组成, D 为对角矩阵,化简多元正态分布 $N(m, C)$ 得式(1)。

$$\begin{aligned} N(m, C) &\sim m + N(0, C) \\ &\sim m + N(0, B D^2 B^T) \\ &\sim m + N(0, (BD)(BD)^T) \\ &\sim m + BDN(0, I) \end{aligned} \quad (1)$$

其中,“ \sim ”表示服从相同的分布。

由式(1)的逆序可知, I 确定等概率密度圆球面,通过 B, D 的变化可以确定 C 的椭球面;球面的分布尺度由 D 中的对角元素决定,主轴的方向由 B 确定,变化 B, D 实现球面的分布旋转。

1.2 算法步骤

CMA-ES算法流程如图1所示。

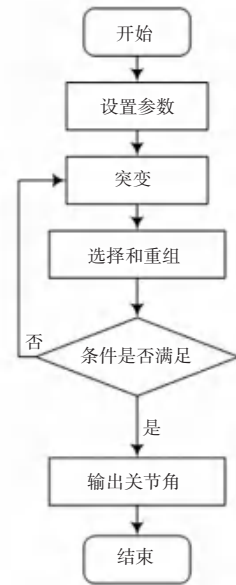


图1 CAM-ES 算法流程图

Fig. 1 CAM-ES algorithm flow chart

Step 1 设置参数并初始化。静态参数:子代个体数 λ ,父代个体数 $\mu < \lambda$,重组权值 $\omega_i = 1, 2, \dots, u$,自适应调整时的常量 $c_\sigma, d_\sigma, c_c, c_1, c_\mu$,最大迭代次数为 G ;动态参数:初始分布均值为 $m^{(0)} \in R^N$,根据实际问题设置维数 N ,初始分布均值可以是随机获得,也可以直接给定,初始全局步长 $\sigma^{(0)} \in R^+$,初始进化路径 $p_\sigma^{(0)} = 0, p_c^{(0)} = 0$,初始协方差矩阵 $C^{(0)} = I \in R^{N \times N}$,初始进化代数 $g^{(0)} = 0$ 。

Step 2 突变。突变的目的是使个体之间产生差异,CMA-ES算法的突变过程是以当代的均值为中心,产生下一代种群,式(2)。

$$\begin{aligned} x_k^{(g+1)} &= m^{(g)} + \sigma^{(g)} N_k(0, C^{(g)}) \\ k &= 1, 2, \dots, \lambda \end{aligned} \quad (2)$$

其中, $x_k^{(g+1)} \in R^N$ 是第 $g+1$ 代种群中生成的第 k 个个体; $m^{(g)}$ 是第 g 代种群中适应度排名前 μ 个个体的加权平均值; $\sigma^{(g)}$ 是第 g 代种群进化的步长; $C^{(g)}$ 是第 g 代种群进化的协方差矩阵。

Step 3 选择和重组。经典ES算法的选择策略有两种:一种是 $(\mu + \lambda)$ 策略,适应度排名由上一代的父代与当前子代一起参与,同时选择优秀的个体;另一种是 (μ, λ) 策略,即父代不参与当前代的适应度排名竞争,优秀个体只从当前子代中选择。CMA-ES采用的是第二种选择策略,对 λ 个子代进行适应度评价,根据适应能力由高到低排序,选取适应能力强的前 μ 个个体,作为下一代种群的父代,更新策略参数 $m^{(g)}, \sigma^{(g)}, C^{(g)}$,以此来传递优良的基因。

(1) 均值移动:下一代起始搜索点,即新的均值

$m^{(g+1)}$ 是从样本 $x_1^{(g+1)}, \dots, x_\lambda^{(g+1)}$ 中选择出来的 μ 个最优子群的加权平均值, 式(3) 和式(4)。

$$m^{(g+1)} = m^{(g)} + \sigma^{(g)} \langle y \rangle_\omega = \sum_{i=1}^{\mu} \omega_i x_{i;\lambda}^{(g+1)} \quad (3)$$

$$\langle y \rangle_\omega = \sum_{i=1}^{\mu} \omega_i y_{i;\lambda}^{(g)}$$

$$y_{i;\lambda}^{(g)} = (x_{i;\lambda}^{(g)} - m^{(g)}) / \sigma^{(g)} \quad (4)$$

其中, 权重 ω_i 满足 $\sum_{i=1}^{\mu} \omega_i = 1$, $x_{i;\lambda}^{(g+1)}$ 是第 $g+1$ 代的 λ 个个体中第 i 个最优的个体, 即突变中心 $m^{(g+1)}$ 为下一代的父代最优个体的加权平均值。

(2) 协方差矩阵自适应调整 (CMA), 式(5) ~ 式(9)。

$$\begin{cases} \uparrow 1, & \text{如果 } \frac{\|p_\sigma^{(g+1)}\|}{\sqrt{1 - (1 - c_\sigma)^{2(g+1)}}} < \frac{\sigma}{e} 1.5 + \frac{1}{n - 0.5} \frac{\ddot{\sigma}}{\sigma} \\ \uparrow h_\sigma^{(g+1)} \\ \uparrow 0 & \text{否则} \end{cases} \quad (5)$$

$$E \|N(0, I)\| \approx \sqrt{N} \frac{\sigma}{e} - \frac{1}{4N} + \frac{1}{21N^2} \frac{\ddot{\sigma}}{\sigma} \quad (6)$$

$$p_\sigma^{(g+1)} = (1 - c_c) p_\sigma^{(g)} + h_\sigma^{(g)} \cdot \sqrt{c_c(2 - c_c)} \mu_{eff} \frac{m^{(g+1)} - m^{(g)}}{\sigma^{(g)}} \quad (7)$$

$$C^{(g+1)} = (1 - c_1 - c_\mu) C^{(g)} + c_1 (p_c^{(g+1)} p_c^{(g+1)T} + \delta(h_\sigma^{(g+1)}) C^{(g)}) + c_\mu \sum_{i=1}^{\mu} \omega_i y_{i;\lambda}^{(g+1)} y_{i;\lambda}^{(g+1)T} \quad (8)$$

$$\delta(h_\sigma^{(g+1)}) = (1 - h_\sigma^{(g+1)}) c_c (2 - c_c) \quad (9)$$

其中, c_c 是 p_c 的更新学习速率; h_σ 为赫维赛德阶跃 (Heaviside) 函数, 用于调整 $\|p_c\|$ 的过大增长; μ_{eff} 为方差有效选择质量, 且 $1 < \mu_{eff} \leq \mu$, c_1 和 c_μ 分别为 C 的“秩 1”和“秩 μ ”的更新学习速率。

(3) 全局步长控制 (CSA), 式(10) 和式(11)。

$$p_\sigma^{(g+1)} = (1 - c_\sigma) p_\sigma^{(g)} + \sqrt{c_\sigma(2 - c_\sigma)} \mu_{eff} \cdot C^{-\frac{1}{2}} \frac{m^{(g+1)} - m^{(g)}}{\sigma^{(g)}} \quad (10)$$

$$\sigma^{(g+1)} = \sigma^{(g)} \exp \left(\frac{\sigma}{e} \frac{\|p_\sigma^{(g+1)}\|}{E(\|N(0, I)\|)} - 1 \right) \frac{\ddot{\sigma}}{\sigma} \quad (11)$$

$$\rho = \exp \left(\frac{\sigma}{e} \frac{\|p_\sigma^{(g+1)}\|}{E(\|N(0, I)\|)} - 1 \right) \frac{\ddot{\sigma}}{\sigma}$$

其中, ρ 可看作步长变化的伸缩因子; c_σ 为 p_σ 的更新学习率; d_σ 为接近于 1 的阻尼系数; $E \|N(0, I)\|$ 为归一化进化路径在随机选择下的期望长度。

Step 4 判断是否达到最大迭代次数或精度, 若是, 则停止, 输出最优解和最优决策向量, 否则返

回 Step 2。

2 逆运动学计算

2.1 正向运动学数学模型

采用标准 D-H 法建立的 REBot-V-6R-6500 型六自由度机器人结构简图, 如图 2 所示。目前, 普遍认为 D-H 法是对机器人运动学建模最好的方法, 可以确定机器人的关节参数和关节变量, 与机器人结构顺序和复杂程度无关, 通过齐次变换矩阵描述两个相邻坐标系间的空间位姿关系。

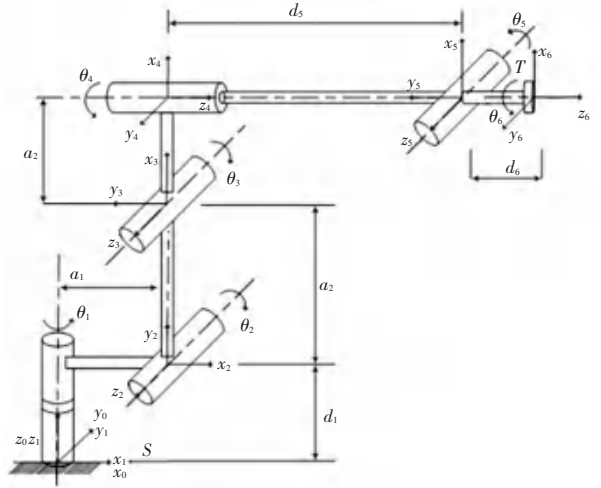


图 2 REBot-V-6R-6500 结构简图

Fig. 2 Structure diagram of REBot-V-6R-6500

首先, 建立连杆坐标系, 通过 4 次旋转平移变换, 根据 DH 参数, 第 i 连杆的参数定义为: a_i 为连杆长度; α_i 为相邻两关节轴线的夹角; d_i 为第 i 根连杆和第 $i-1$ 连杆的偏置距离; θ_i 为第 i 连杆的旋转角。变换过程: 绕 z_{i-1} 轴旋转 θ_i 角, 沿 z_{i-1} 轴平移 d_i , 沿 x_i 轴平移 a_i , 绕 x_i 轴旋转 α_i , 可以得到机器人连杆 i 坐标系相对于连杆 $i-1$ 坐标系的变换矩阵 ($D-H$ 矩阵)^[8] 即式(12), REBot-V-6R-6500 机器人关节参数, 见表 1。

表 1 REBot-V-6R-6500 机器人关节参数

Tab. 1 Joint parameters of REBot-V-6R-6500 robot

连杆 i	关节角 $\theta_i / (^\circ)$	扭角 $\alpha_i / (^\circ)$	长度 a_i / m	偏置 d_i / m
1	θ_1	90	0.100	0.320
2	$\theta_2 + 90$	0	0.290	0
3	θ_3	90	0.121	0
4	θ_4	-90	0	0.310
5	θ_5	90	0	0
6	θ_6	0	0	0.1115

$${}^{i-1}T_i = Rot(\theta_i, z_{i-1})Tran(0,0,d_i)Trans(a_i,0,0)Rot(a_i, x_i) = \begin{pmatrix} \dot{\theta}_i \cos \theta_i & -\sin \theta_i \cos a_i & \sin \theta_i \sin a_i & a_i \cos \theta_i \dot{\theta}_i \\ \dot{\theta}_i \sin \theta_i & \cos \theta_i \cos a_i & -\cos \theta_i \sin a_i & a_i \sin \theta_i \dot{\theta}_i \\ \dot{a}_i & \sin a_i & \cos a_i & d_i \dot{a}_i \\ \dot{\theta}_i & 0 & 0 & 1 \end{pmatrix} \quad (12)$$

根据式(12)和表(1)中的 $D-H$ 参数,得到该型号六自由度机器人正向运动学数学模型式(13),通过正向运动学控制机器人末端执行器到达指定位置。

$${}^0T = {}^0T(\theta_1) {}^1T(\theta_2) {}^2T(\theta_3) {}^3T(\theta_4) {}^4T(\theta_5) {}^5T(\theta_6) = \begin{pmatrix} \dot{\theta}_1 n_x & o_x & a_x & p_x \dot{\theta}_1 \\ \dot{\theta}_1 n_y & o_y & a_y & p_y \dot{\theta}_1 \\ \dot{a}_1 & \sin a_1 & \cos a_1 & d_1 \dot{a}_1 \\ \dot{\theta}_1 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \dot{\theta}_1 & o & a & p \dot{\theta}_1 \\ \dot{a}_1 & 0 & 0 & 1 \end{pmatrix} \quad (13)$$

2.2 适应度函数

用机器人末端执行器的接近矢量 \mathbf{a} 和位置矢量 \mathbf{P} 建立适应度函数 f_1 和 f_2 , 式(14)。适应度函数 f_1 为位置误差范数,表示目标位置 P_d 和实际位置 P_c 之间的误差;适应度函数 f_2 为姿态误差范数,表示目标位置 a_d 和实际位置 a_c 之间的误差;适应度函数 $f_1 + f_2$ 表示位置误差范数和姿态误差范数之和。系数 x 和 y 是为方便调整求解过程中位置和姿态误差而设置的权重,以便求得最合适的逆解,本文中 $x = 1, y = 1$ 。

$$\begin{aligned} f_1 &= x \| p_d - p_c \| \\ f_2 &= y \| a_d - a_c \| \end{aligned} \quad (14)$$

六自由度机器人逆运动学存在多解,为了获得唯一的逆解,本文基于加权最小二乘法,结合“最佳柔顺性”规则,通过附加约束建立目标函数 f_3 , 使用 CMA-ES 算法求解适应度函数 $f = f_1 + f_2 + f_3$ 最小值,获得最优解,最佳柔顺性的含义为式(15):

$$\min f_3 = \sum_{i=1}^{i=6} z \times [w_i(\theta_i(k) - \theta_i(k-1))]^2, i = 1, 2, 3, \dots, 6 \quad (15)$$

其中, $\theta_i(k) - \theta_i(k-1)$ 表示关节角 i 与其自身前一个位置关节角的差值,使得关节运动更平滑,减少机器人运动路径距离;系数 z 是为了保证算法运行时不产生局部收敛甚至发散等情况,平衡“最佳柔顺性”准则对整个适应度函数优化结果的影响^[9]。本文中 $z = 0.001$;系数 w_i 遵循“少运动上臂,多运动下臂”的选取原则。该原则的优点是运动时更为平稳,减少能源消耗,提高工作效率等。在本研究中,加权系数 w_i 设计为 $w = [3 \ 2 \ 1.5 \ 1 \ 1 \ 1]$ 。

3 仿真与结果分析

3.1 求解工作空间

机器人的关节角取值范围参考该型号机器人说明书,见表 2。求解机器人工作空间方法有:解析法、几何法、数值法等,但这几种计算复杂、效率低,本文在机器人各关节允许的角度值限制内,以蒙特卡洛法为基础,用随机函数给各关节角赋值,得到 10 000 组关节角,计算机器人可达工作空间;利用正向运动学公式(13)解出末端执行器在笛卡尔空间中离散点图,从而得到机器人三维工作空间,如图 3 所示。

表 2 关节角取值范围

Tab. 2 Range of joint angle

关节	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
上限	-170°	-45°	-70°	-170°	-120°	0°
下限	170°	135°	130°	170°	120°	360°

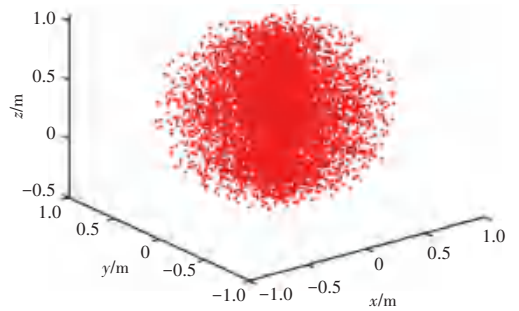


图 3 三维工作空间

Fig. 3 Three-dimensional workspace

3.2 参数设置

CMA-ES 算法的种群数量 $\lambda = 100$, 优秀个体 $\mu = 50$, 初始步长 $\sigma = 0.5$, 初始均值在 0~1 内随机产生,求解过程中,前一个逆向运动学解作为当前初始点均值,初始均值缩放系数 $\varepsilon = 0.65$,算法的停止条件为进化次数 $G = 200$ 或者适应度函数 $f \leq 10^{-9}$ 。为了保证算法稳定运行,两种算法的适应度函数中,权值系数 $x = 1, y = 1, z = 0.001$ 。

3.3 点到点运动仿真

仿真的计算机配置:操作系统为 64 位 Window 10 企业版,处理器为 AMD Ryzen 5 2600X 3.60 GHz,内存 8.0 GB。

机器人初始状态时选取各关节角度为 $q = [0 \ 0 \ 0 \ 0 \ 0 \ 0]$,目标位置选取工作空间内的一组关节角,根据正向运动学公式(13)求出末端执行器的位姿矩阵,分别使用适应度函数 $f_1 + f_2$ 和 f ,在 CMA-ES 算法上独立运行 1 000 次,求逆运动学的解。随机选取机器人的一组关节角度为 $q =$

[0.412 0.365 0.254 0.121 0.454 0.235]。

两种适应度函数的算法单独运行 1 000 次的结果见表 3, 结果表明改进后适应度函数代表位置误差的适应度函数 f_1 的平均值比原适应度函数中 f_1 的平均值小 9 个数量级; 任意选取 1 000 组结果中的一组的位置误差结果见表 4, 可以看出改进适应度函数的位置误差值比未改进的适应度函数求得的位置精度高 10^{-6} 数量级。

表 3 CMA-ES 独立运行 1 000 次的适应度函数 f_1 的值

Tab. 3 The value of the fitness function f_1 for CMA-ES running independently for 1 000 times

适应度函数	最小值	最大值	平均值
$f_1 + f_2$	$4.922 2 \times 10^{-9}$	$4.519 1 \times 10^{-8}$	$3.099 2 \times 10^{-8}$
f	$8.673 9 \times 10^{-19}$	$5.906 3 \times 10^{-17}$	$4.025 2 \times 10^{-17}$

表 4 点对点运动位置误差

Tab. 4 Position error of point-to-point motion

适应度函数	$ x /m$	$ y /m$	$ z /m$
原适应度函数	$5.375 1 \times 10^{-10}$	$1.855 8 \times 10^{-10}$	$2.950 7 \times 10^{-11}$
改进适应度函数	$1.111 0 \times 10^{-16}$	$1.387 8 \times 10^{-17}$	$1.111 0 \times 10^{-16}$

3.4 连续轨迹跟踪仿真

在工作空间内的轨迹方程为式(16):

$$\begin{cases} x = 0.15\cos\theta + 0.51 \\ y = 0.15\sin\theta \\ z = 0.05t + 0.35 \end{cases} \quad (16)$$

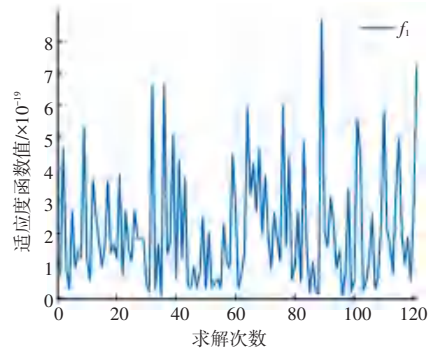
其中, $0 \leq \theta \leq 2\pi$, 步长为 $\pi/60$ (rad), 长度单位为 m, 为了满足实际需求, 保证末端执行器关节轴在工作过程中始终平行于世界坐标系 x 轴正方向, 运动轨迹的每一个点的 α 矢量姿态均为 $[1; 0; 0]$ 。

使用原适应度函数和改进适应度函数对选取的空间内轨迹方程(16)进行连续轨迹跟踪, 适应度函数 $f_1 + f_2$ 和 f 中代表位置误差的适应度函数 f_1 的结果见表 5, 连续求解过程中适应度函数值 f_1 的波动情况如图 4 所示。仿真结果表明, 改进的适应度函数中 f_1 的平均值比原适应度函数中 f_1 的平均值少 13 个数量级, 同时对轨迹上所有点进行轨迹跟踪, 轨迹跟踪以及轨迹仿真如图 5 所示, 可以看到能完全实现对给定曲线的跟踪, 且运动过程平稳。

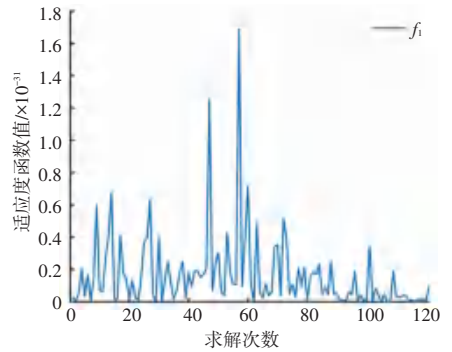
表 5 连续轨迹跟踪的适应度函数 f_1 的值

Tab. 5 The value of the fitness function f_1 for continuous trajectory tracking

适应度函数	最小值	最大值	平均值
$f_1 + f_2$	$4.292 1 \times 10^{-21}$	$8.711 5 \times 10^{-19}$	$2.071 5 \times 10^{-19}$
f	$1.005 3 \times 10^{-35}$	$1.688 3 \times 10^{-31}$	$1.626 9 \times 10^{-32}$



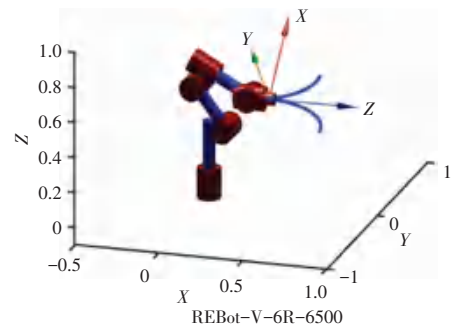
(a) 原适应度函数



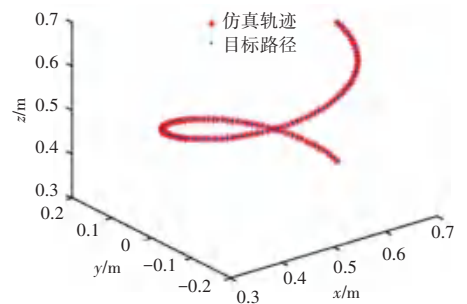
(b) 改进的适应度函数

图 4 适应度函数 f_1 的波动

Fig. 4 Fluctuation of fitness function f_1



(a) 轨迹仿真



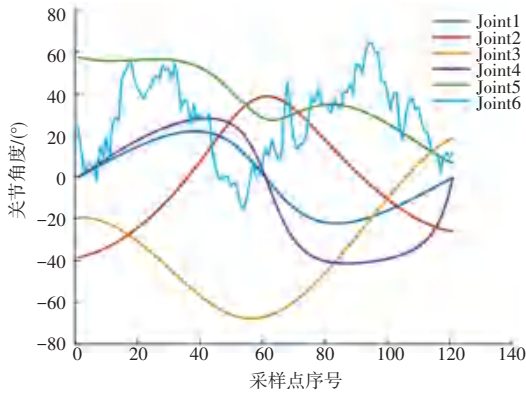
(b) 轨迹跟踪

图 5 轨迹仿真与跟踪

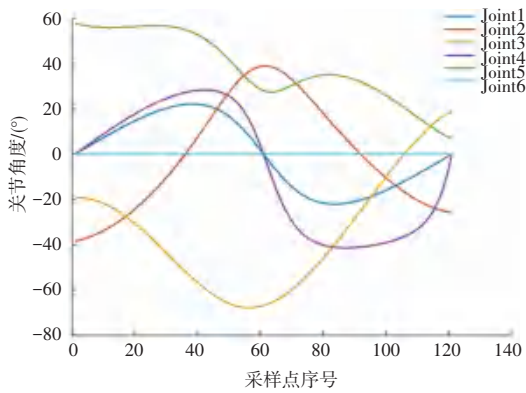
Fig. 5 Trajectory simulation and tracking

两种适应度函数使用 CMA-ES 算法求解轨迹

的关节角度变化如图 6 所示。结果表明,在保持原适应度函数的特点上,基于加权最小二乘法“最佳柔顺性”规则改进的适应度函数求解的结果唯一且更光滑,关节运动距离更小,能源消耗更少,运行过程平稳。



(a) 原适应度函数



(b) 改进的适应度函数

图 6 关节角变化

Fig. 6 Changes of joint angle

在两种适应度函数求解下,笛卡尔空间中目标位置 $P(X,Y,Z)$ 和仿真位置 $p(x,y,z)$ 之间的误差变化,如表 6、表 7 和图 7 表示。可以看出,两者求解的位置精度相差较大,采用新适应度函数求解得到的平均位置误差值稳定在 10^{-17} m 数量级,最小值误差为 0 m,且代表位置误差稳定度的标准差值也稳定在 10^{-17} m 数量级,在轨迹跟踪精度和轨迹跟踪稳定度方面明显优于使用原适应度函数解得的结果。

表 6 原适应度函数轨迹跟踪的位置误差

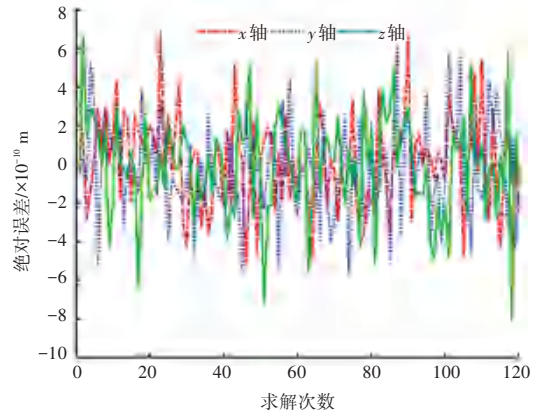
Tab. 6 Position error of original fitness function for trajectory tracking

位置误差/m	x	y	z
最小值	$1.067\ 1 \times 10^{-12}$	$2.727\ 7 \times 10^{-13}$	$5.717\ 6 \times 10^{-15}$
最大值	$6.899\ 7 \times 10^{-10}$	$6.181\ 4 \times 10^{-10}$	$8.064\ 5 \times 10^{-10}$
平均值	$1.876\ 1 \times 10^{-10}$	$2.047\ 2 \times 10^{-10}$	$2.033\ 1 \times 10^{-10}$
标准差	$1.523\ 4 \times 10^{-10}$	$1.571\ 9 \times 10^{-10}$	$1.739\ 4 \times 10^{-10}$

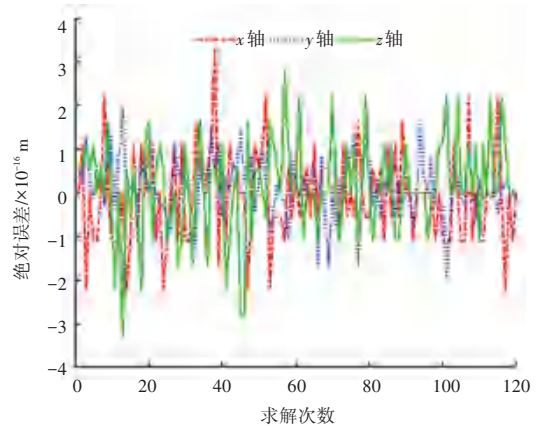
表 7 改进适应度函数轨迹跟踪的位置误差

Tab. 7 Position error of improved fitness function for trajectory tracking

位置误差/m	x	y	z
最小值	0	0	0
最大值	$3.330\ 7 \times 10^{-16}$	$1.942\ 9 \times 10^{-16}$	$3.330\ 7 \times 10^{-16}$
平均值	$8.257\ 9 \times 10^{-17}$	$5.065\ 5 \times 10^{-17}$	$7.936\ 7 \times 10^{-17}$
标准差	$7.025\ 9 \times 10^{-17}$	$4.759\ 4 \times 10^{-17}$	$8.038\ 8 \times 10^{-17}$



(a) 原适应度函数



(b) 改进的适应度函数

图 7 空间位置误差波动

Fig. 7 Fluctuation of spatial position error

4 结束语

本文提出改进的适应度函数,将加权最小二乘法的“最佳柔顺性”规则和位姿误差结合,形成改进的适应度函数,使用 CMA-ES 算法求逆运动学解,使其解具有唯一性;在单点求得的逆解中,改进适应度函数中代表位置误差的适应度函数 f_1 的平均值比原适应度函数中 f_1 的平均值少 10 个数量级;在连续的轨迹求解中,改进适应度函数中 f_1 的平均值比原适应度函数逆解的结果提升 13 个数量级,且使用改进适应度函数求解得到的平均位置误差值稳定在 10^{-16} m。(下转第 31 页)