

基于 RL-DE 改进算法的移动机械臂逆运动学求解研究

胡德安¹, 孙智¹, 王秋枫², 李明俊^{1†}

(1. 湖南大学 机械与运载工程学院, 湖南 长沙 410082;
2. 中核四〇四有限公司 第三分公司, 甘肃 嘉峪关 732850)

摘要:针对移动平台与机械臂耦合系统因整体关节自由度冗余导致的逆运动学求解难题,传统的基于伪逆矩阵解决方法存在奇异点敏感性缺陷.为解决多自由度冗余移动机械臂逆运动学求解问题,提出一种融合强化学习(reinforcement learning, RL)与差分进化(differential evolution, DE)的改进算法.其中,基于移动机械臂物理模型,以目标位姿与移动机械臂末端实际到达位姿的误差为适应度函数,构建DE算法优化模型.为克服传统DE算法进化参数固定的局限性,进而提升算法的全局搜索能力和求解收敛速度,提出一种调整不同时期进化方向的动态突变多策略,实现全局探索与局部开发策略之间的权衡,并将种群的适应度与多样性的变化作为特征输入,以最优个体适应度的进化率作为奖励函数,采用深度确定性策略梯度(deep deterministic policy gradient, DDPG)强化学习方法优化种群进化参数.仿真结果表明:本文提出的方法无须对矩阵求逆,可有效规避奇异点问题,且其逆运动学求解成功率和定位误差等性能优于人工蜂群、粒子群、自适应差分进化等算法,同时轨迹平滑度及平均误差等轨迹跟踪性能优于自适应差分进化算法.

关键词:移动机械臂;逆运动学;差分进化算法;强化学习;动态突变多策略

中图分类号:TP242

文献标志码:A

Research on inverse kinematics solution of mobile robotic arms based on an improved RL-DE algorithm

HU Dean¹, SUN Zhi¹, WANG Qiufeng², LI Mingjun^{1†}

(1. College of Mechanical and Vehicle Engineering, Hunan University, Changsha 410082, China;
2. The Third Branch, CNNC 404 Co., Ltd., Jiayuguan 732850, China)

Abstract: To address the inverse kinematics challenge arising from the joint redundancy in mobile platform-arm coupled systems, the traditional pseudo-inverse matrix-based methods are sensitive to singularity. This paper proposes an improved algorithm that integrates reinforcement learning (RL) and differential evolution (DE) to solve the inverse kinematics problem for highly redundant mobile robotic arms. The optimization model in the DE algorithm is constructed based on the physical model of the mobile robotic arms, with the fitness function defined as

* 收稿日期:2025-03-07

基金项目:国家自然科学基金资助项目(52402495, U2341269), National Natural Science Foundation of China(52402495, U2341269)

作者简介:胡德安(1977—),男,湖南长沙人,湖南大学教授,博士

† 通信联系人, E-mail: mingjunl@hnu.edu.cn

the error between the target pose and the desired end-effector pose of the mobile robotic arms. To overcome the limitations of fixed evolutionary parameters in conventional DE algorithm and enhance both global search capability and convergence speed, a dynamic mutation multi-strategy mechanism that adaptively adjusts evolutionary directions at different stages to balance global exploration and local exploitation is proposed. Furthermore, the deep deterministic policy gradient (DDPG) algorithm is employed to optimize the evolutionary parameters, in which variation in population fitness and diversity serves as state inputs, and the improvement rate of the best individual's fitness acts as the reward signal. Simulation results demonstrate that the proposed method eliminates the need for matrix inversion, effectively circumventing singularity issues and outperforming artificial bee colony, particle swarm optimization, and self-adaptive differential evolution algorithms in terms of inverse kinematics success rate and positioning accuracy. Additionally, it outperforms the self-adaptive differential evolution algorithm in trajectory smoothness and tracking precision, and exhibits lower average error in trajectory tracking.

Key words: mobile robotic arms; inverse kinematics; differential evolution algorithm; reinforcement learning; dynamic mutation multi-strategy

移动机械臂通过移动平台与多个机械臂协作运动,具有可操作性强、性能高及移动性强等特点,极大地提升了机器人系统的工作空间和运动灵活性,目前广泛应用于空间检测、物体搬运、环境检测、人员辅助等多个领域^[1-2]。然而,机械臂与移动平台的组合导致移动机械臂在工作空间的自由度冗余。移动机械臂的逆运动学研究可转变为多变量、强耦合的高维非线性方程组求解问题,但该过程复杂且存在多组解^[3]。因此,如何高效寻找每个驱动关节的最优控制量,确保末端执行器精确到达指定位姿的空间点,是当前移动机械臂逆运动学研究的难点问题。

近年来,移动机械臂逆运动学研究的方法有解析法、基于雅可比矩阵的方法、数值方法、神经网络方法和群体智能方法^[2]。ZHAO等^[4]提出一种适用于非球腕结构七自由度空间机械臂的闭式逆运动学解析方法,通过固定不同关节识别冗余自由度,基于螺钉理论推导非参数关节角的闭式表达式,并分析多种自运动流形分支获得最佳解析解。解析法基于移动机械臂的构型布置,直接用数学公式表示末端位姿与驱动关节之间的关系。移动机械臂的自由度高且结构复杂,几何分析方法通常不适用。COLAN等^[5]针对串联机器人的逆运动学问题提出基于雅可比矩阵的可变步长迭代方法,并与传统恒定步长方法比较,实验结果表明可变步长策略显著提高基于雅可比矩阵的逆运动学求解效率和收敛性能。基于雅可比矩阵的方法通过移动机械臂的雅可比矩阵在速度层连续迭代求解逆运动学问题,不能有效求解

奇异值。WOLIŃSKI等^[6]提出一种基于二次规划的新型逆运动学求解方法,将关节加速度作为决策变量,并与经典的伪逆方法进行对比,结果表明该方法在约束处理方面具备更高的实用性。采用数值方法将逆运动学问题转化为二次规划问题,其求解过程主要在速度层和加速度层,因此无法准确得到各种约束点和奇异点的解。BAI等^[7]提出一种基于果蝇优化算法的前馈神经网络求解机器人逆运动学问题,通过果蝇优化算法提高神经网络的收敛精度与泛化能力,并与传统神经网络方法和基于粒子群优化的神经网络方法进行对比,结果表明基于果蝇优化算法的神经网络方法在误差控制方面表现更优。神经网络方法通过构造多层神经网络学习移动机械臂关节角度与末端位姿的非线性关系,复杂度高,求解时间长^[8]。由于基于生物群体智能的算法具有良好的全局收敛能力,不需要计算雅可比矩阵及其伪逆,不存在奇异点^[9],已成为移动机械臂逆运动学求解研究的热点。

目前用于求解移动机械臂逆运动学问题的群体智能算法主要有粒子群(particle swarm optimization, PSO)算法、遗传(genetic algorithm, GA)算法、蚁群(ant algorithm, AA)算法、人工蜂群(artificial bee colony, ABC)算法和差分进化(differential evolution, DE)算法等^[10]。PSO算法易于实现,执行效率高,但是全局搜索能力非常依赖对粒子速度上限的设定^[11],不符合全局收敛准则。GA算法在单个染色体上使用交叉操作然后相互比较,拥有较强的全局搜索能力,

但染色体共享效率低^[12].AA算法基于个体协作寻优,具有良好的可扩展性和较强的全局搜索能力,但在面对复杂问题时收敛速度慢,容易陷入局部最优^[13].ABC算法是一种模仿蜜蜂群体觅食行为的优化方法,具有较快的收敛速度,参数对算法收敛性能影响大,因此陷入局部最优的风险较高^[14].DE算法参数少,结构简单,自适应与优化能力强,全局搜索能力突出,可有效求解移动机械臂逆运动学问题^[15].

LÓPEZ-FRANCO等^[15]使用Denavit-Hartenberg法构建移动机械臂的运动学模型,定义优化目标函数,使用标准DE算法研究三种移动机械臂的逆运动学,结果显示DE算法在精度与求解时间上优于PSO、CS(cuckoo search,布谷鸟搜索算法)等算法.HERNANDEZ-BARRAGAN等^[16]提出一种自适应突变策略用于权衡全局搜索与局部开发的方法,将DE算法应用于求解协作移动机械臂的逆运动学问题,多次实验表明其精度CFPSO(constriction factor PSO,收缩因子粒子群优化)、FPA(flower pollination algorithm,花粉授粉)、KABC(knowledge-based artificial bee colony,基于知识的人工蜂群)算法.ZHANG等^[17]提出一种应用于移动机械臂的改进自适应变异差分进化(improved self-adaptive mutation differential evolution, ISAMDE)算法,加入新的自适应突变算子和缩放因子控制变异策略,通过比较该算法在跟踪不同轨迹的性能,表明该算法在解决移动机械臂逆运动学问题的精度与关节平滑性方面具有显著优势.LI等^[2]研究DE算法的变异策略与进化参数对移动机械臂逆运动学的影响,结果显示设置不同变异策略与进化参数,移动机械臂的逆运动学误差有显著变化.

综上所述,国内外研究学者已针对移动机械臂逆运动学求解算法取得了丰富的研究成果,然而仍存在优化和改进空间.已有方法主要关注于通过改进DE算法的变异策略提高算法的全局搜索能力,在求解逆运动学问题时直接根据经验确定种群的进化参数,无法保证算法处理不同目标函数寻优过程的鲁棒性.然而,种群的变异和交叉参数直接影响算法的全局搜索能力与收敛速度^[2].其中,变异参数通过改变搜索步长影响收敛速度和种群多样性,DE算法倾向于在早期阶段进行全局搜索以探索整个可行区域,然后进行局部搜索以加快收敛速度;交叉参数决定亲本传递给后代的信息量,对收敛速度与种群多样性有明显影响^[18].为解决上述问题,本文以一种10自由度移动机械臂为研究对象,建立逆运动学优化模型,提出融合强化学习(RL)与差分进化

(DE)的改进算法,设计动态突变多策略,调整不同进化时期种群全局搜索或局部优化的进化方向,并综合种群适应度与多样性的变化度因素,采用DDPG强化学习方法优化种群进化参数.

1 移动机械臂逆运动学模型构建

本节主要介绍移动机械臂物理模型构建,为逆运动学求解算法研究提供模型基础.

1.1 机械臂物理模型

机械臂由一系列刚体组成,这些刚体称为连杆.连杆由关节连接.相邻两个关节之间的连接定义一个关节变量 q_i ,给定所有关节变量 $\mathbf{q}$,通过正运动学得到机械臂末端的位姿.将所有关节变量定义为:

$$\mathbf{q} = [q_1 \quad q_2 \quad q_3 \quad \cdots \quad q_n]^T \quad (1)$$

式中: q_i 表示机械臂的不同位置的关节, $i \in (1, 2, 3, \cdots, n)$; n 是机械臂的自由度.机械臂的连杆与关节运动模型构成一条运动链,如图1所示.

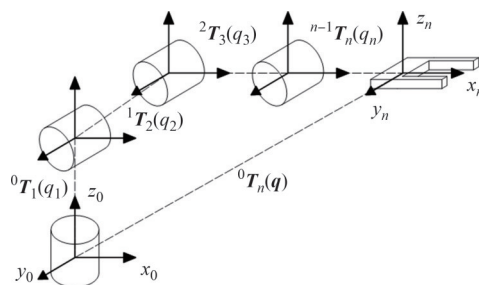


图1 机械臂运动链图

Fig.1 The diagram of robotic arms motion chain

其正运动学方程可表示为^[19]:

$${}^0T_n(\mathbf{q}) = {}^0T_1(q_1) {}^1T_2(q_2) {}^2T_3(q_3) \cdots {}^{n-1}T_n(q_n) = \prod_{i=1}^n {}^i T_i(q_i) \quad (2)$$

式中: ${}^0T_n(\mathbf{q})$ 表示世界参考点到末端执行器位姿的齐次变换矩阵; ${}^{i-1}T_i(q_i)$ 表示连杆 $i-1$ 到连杆 i 的齐次变换矩阵.使用改进的DH(Denavit-Hartenberg)参数法^[19]:

$${}^{i-1}T_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & a_{i-1} \\ \sin(\theta_i)\cos(\alpha_{i-1}) & \cos(\theta_i)\cos(\alpha_{i-1}) & -\sin(\alpha_{i-1}) & -\sin(\alpha_{i-1})d_i \\ \sin(\theta_i)\sin(\alpha_{i-1}) & \cos(\theta_i)\sin(\alpha_{i-1}) & \cos(\alpha_{i-1}) & \cos(\alpha_{i-1})d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

式中: θ_i 是连杆转角; d_i 是连杆偏距; a_{i-1} 是连杆长度; α_{i-1} 是连杆扭转角.若关节为转动关节, q_i 表示

旋转角度, θ_i 是关节变量; 若关节为移动关节, q_i 表示移动距离, 参数 d_i 是关节变量, 如式(4)所示:

$$q_i = \begin{cases} \theta_i, & \text{关节 } i \text{ 为旋转关节} \\ d_i, & \text{关节 } i \text{ 为移动关节} \end{cases} \quad (4)$$

在笛卡儿坐标系下, 连杆 $i-1$ 到连杆 i 的齐次变换矩阵可表示为:

$${}^{i-1}T_i = \begin{bmatrix} \mathbf{R} & \mathbf{P} \\ \mathbf{0} & 1 \end{bmatrix} \quad (5)$$

式中: $\mathbf{R}$ 是 3×3 旋转矩阵, 表示连杆 i 的方向; $\mathbf{P}$ 是 3×1 矩阵, 表示连杆 i 的位置.

1.2 移动底盘物理模型

本文采用前摆臂式履带底盘作为移动机械臂的移动平台, 如图2所示. 其中 θ_f^* 表示虚拟转动关节, d_f^* 表示虚拟移动关节.

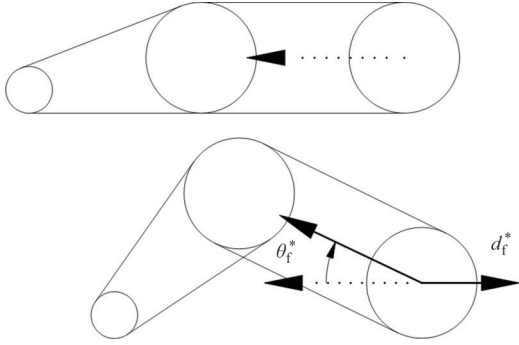


图2 前摆臂运动示意图

Fig.2 The diagram of forerunner motion

若不考虑前摆臂的影响, 将移动平台的位置变化等效为三个虚拟关节的变化量, 如图3所示. 从当前位置到下一个位置, 移动平台的两个虚拟关节 d_x^* 与 d_y^* 代表平台的平移, 虚拟关节 θ_z^* 代表平台的旋转^[15]. 若考虑前摆臂转动, 移动平台整体绕后轮轴旋转, 同时沿平台前进方向偏离一定移动关节量 d_f^* . d_x^* 与 d_y^* 虚拟关节平移一定关节量可以抵消 d_f^* 的偏离影响, 因此前摆臂转动等效于增加一个俯仰转动关节 θ_f^* .

1.3 移动机械臂物理模型

基于连杆模型与虚拟关节方法, 本文研究的移动机械臂关节布置如图4和图5所示. 由图4可知, p_x, p_y 表示平台的虚拟移动关节, θ_z 表示方向垂直于 p_x, p_y 移动平面的虚拟转动关节, θ_f 表示绕后轮轴旋转的俯仰关节. $\theta_1, \theta_2, p_3, \theta_4, p_5, \theta_6$ 表示机械臂的移动关节或转动关节, 其中, p_3, p_5 为移动关节, $\theta_1, \theta_2, \theta_4, \theta_6$ 为转动关节. 图5为移动机械臂改进 DH 参数法坐

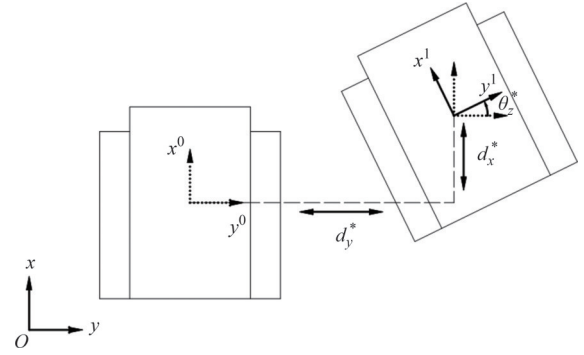


图3 虚拟方法示意图

Fig.3 The diagram of virtual method

标系布局, 其中, 坐标系中下标为 $X, Y, Z, f, 1, 2, 3, 4, 5, 6$ 的参数分别与关节 $p_x, p_y, \theta_z, \theta_f, \theta_1, \theta_2, p_3, \theta_4, p_5, \theta_6$ 对应, G 为世界坐标系.

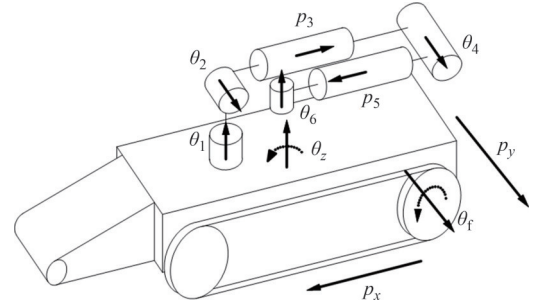


图4 移动机械臂结构示意图

Fig.4 The structural diagram of a mobile robotic arm

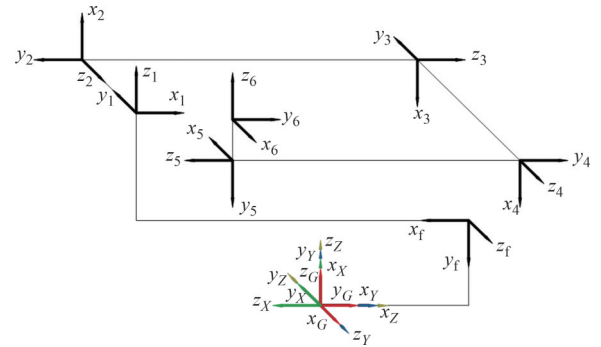


图5 移动机械臂改进 DH 法坐标系布局

Fig.5 The modified DH coordinate system structure of the mobile robotic arm

表1为移动机械臂的改进 DH 参数表. 其中, θ 为连杆转角, α 为连杆扭角, d 为连杆偏距, a 为连杆距离. 将表1中参数代入式(3)~式(5)并结合图5, 构建移动机械臂物理模型.

2 改进 DE 算法

本节主要介绍基于改进 DE 算法的移动机械臂逆

表1 改进DH参数

Tab.1 The modified DH parameters

序号	关节	$\theta/(^\circ)$	d/mm	a/mm	$\alpha/(^\circ)$
1	p_x	90	0	0	90
2	p_y	-90	0	0	90
3	θ_z	0	0	0	90
4	θ_t	180	0	300	90
5	θ_1	180	134	420	90
6	θ_2	90	-23	0	90
7	p_3	180	604	0	90
8	θ_4	0	72.5	0	90
9	p_5	-90	507	0	90
10	θ_6	180	33	0	90

运动学求解方法.通过移动机械臂物理模型构建逆运动学求解问题的适应度函数,结合改进DE算法,将个体基因编码为关节变量,引入均匀初始化操作提升种群多样性,添加动态变异多策略方法与进化监督机制增强算法全局搜索性能,基于种群迭代获得移动机械臂关节在目标位姿点逆运动学的最优解.

2.1 适应度函数构建

定义移动机械臂末端执行器期望位姿为 ${}^0\hat{T}_n$,实际末端到达的位姿为 ${}^0T_n(q)$,两者之间的误差如下式所示:

$$T_{\text{error}} = \| {}^0T_n(q) - {}^0\hat{T}_n \| \quad (6)$$

式中: ${}^0T_n(q), {}^0\hat{T}_n \in \mathbf{R}^{4 \times 4}$, $T_{\text{error}} \in \mathbf{R}$. $q \in \mathbf{R}^n$ 是求解得到的关节估计值, n 表示关节个数, $\|\cdot\|$ 表示弗罗贝尼乌斯范数.

定义关节初始角度 q_{old} 与关节估计角度 q 的误差以最小化移动机械臂的关节运动,运动误差计算为:

$$q_{\text{error}} = \| q_{\text{old}} - q \| \quad (7)$$

式中: $q_{\text{error}} \in \mathbf{R}$. 在求解单个位姿点逆运动学问题时,使用位姿的误差作为适应度函数,即:

$$f_1({}^0\hat{T}_n, q) = T_{\text{error}} \quad (8)$$

在求轨迹跟踪问题时,为降低关节角度的突变,使用位姿误差与关节运动误差的加权和作为适应度函数,适应度函数可定义为:

$$f_2({}^0\hat{T}_n, q, q_{\text{old}}) = \alpha T_{\text{error}} + \beta q_{\text{error}} \quad (9)$$

式中: α, β 是衡量位姿误差与关节运动误差占比的加权因子.

式(8)和式(9)将移动机械臂的逆运动学求解转

化为约束条件下的优化问题,约束为每个关节的运动范围.在差分进化算法中,通过限制个体边界的方式进行关节运动约束.

2.2 均匀初始化

在种群初始化阶段,标准DE算法采用随机方法生成个体数量为 ps 、空间维度为 dim 的初始种群,定义^[2]:

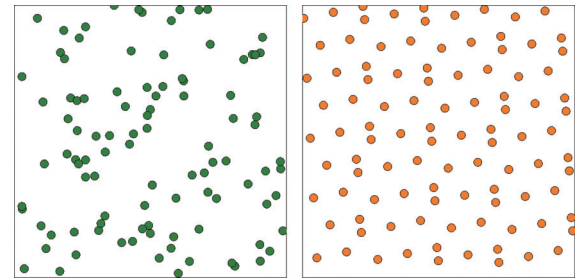
$$X_{i,j} = lb_j + (ub_j - lb_j) \text{rand}(0, 1) \quad (10)$$

式中: $\text{rand}(0, 1)$ 表示 $[0, 1]$ 之间的一个随机数; i 为种群个体, $i = 1, 2, 3, \dots, ps$; j 为空间维度, $j = 1, 2, 3, \dots, dim$; ub_j 和 lb_j 分别表示维度为 j 的上界和下界; $X_{i,j}$ 表示 i 个体第 j 维度信息.由于随机数方法的初始种群在空间上分布不均匀,故采用数理统计方法对种群进行初始化^[20],如下式所示:

$$X_{i,j} = lb_j + (ub_j - lb_j) \bmod \left(2 \cos \left(\frac{2\pi j}{w} \right) i, 1 \right) \quad (11)$$

式中: w 是满足 $(w-3)/2 \geq dim$ 的最小质数.

在二维空间中设置种群个数为100,对式(10)与式(11)的初始化进行对比,如图6所示.结果显示:数理统计方法的初始种群在空间上分布更均匀,有利于增加种群多样性,降低算法陷入局部最优的风险^[20].



(a)随机初始化

(b)均匀初始化

图6 初始化对比示意图

Fig.6 The diagram of initialization comparison

2.3 动态变异多策略方法

改进DE算法在种群多样性丰富的进化前期侧重于全局探索,防止过早陷入局部最优;在进化后期迭代一定次数后种群多样性趋向单一性,侧重在优秀个体周围开发.本文提出一种动态多策略的方法权衡算法在进化过程中探索与开发的占比.采用如下4个算子作为变异策略:

1) DE/rand1:

$$V_i^l = X_{r_1}^l + F_i^l (X_{r_2}^l - X_{r_3}^l) \quad (12)$$

2) DE/best1:

$$V_i^l = X_{\text{best}}^l + F_i^l(X_{r_2}^l - X_{r_3}^l) \quad (13)$$

3) DE/best2:

$$V_i^l = X_{\text{best}}^l + F_i^l(X_{r_1}^l - X_{r_2}^l) + F_i^l(X_{r_3}^l - X_{r_4}^l) \quad (14)$$

4) DE/rand2:

$$V_i^l = X_{r_1}^l + F_i^l(X_{r_2}^l - X_{r_3}^l) + F_i^l(X_{r_4}^l - X_{r_5}^l) \quad (15)$$

式中:rand1 与 rand2 用于全局探索,称为探索算子(exploration);best1 与 best2 用于局部开发,称为开发算子(exploitation); i 表示种群的个体数; l 是迭代次数; V_i^l 表示突变的子代; F_i^l 是突变因子,范围在 $[0, 1]$ 内; r_1, r_2, r_3, r_4, r_5 是 $[1, ps]$ 中互不相同的整数; X_{best}^l 表示当前种群的最优个体。

若突变个体 V_i^l 的某些维度 $V_{i,j}^l$ 超出搜索空间范围,则计算如下:

$$V_{i,j}^l = lb_j + (ub_j - lb_j)\text{rand}(0, 1) \quad (16)$$

动态变异多策略方法首先确定主体算子 mutation, 计算公式如下:

$$\text{mutation} = \begin{cases} \text{exploration, } \text{rand}_l < 1 - \left(\frac{l}{\text{maxIter}}\right)^2 \\ \text{exploitation, 其他} \end{cases} \quad (17)$$

式中:rand_{*l*} 表示 $[0, 1]$ 内的随机数;maxIter 表示最大的迭代次数。

式(17)表明在进化前期种群迭代次数 l 较小,选择探索算子的概率更大;在进化后期迭代次数 l 变大后,选择开发算子的概率更大。rand2 和 best2 算子分别比 rand1 和 best1 算子多一项差分项,相应加强了全局探索能力。确定主体算子后,动态变异多策略方法根据强化学习方法输出的突变因子 F 选择具体的变异算子,如下所示:

$$\text{strategy}^l = \begin{cases} \text{mutation}(1), F^l < F^{l-1} \\ \text{mutation}(2), F^l > F^{l-1} \\ \text{strategy}^{l-1}, \text{其他} \end{cases} \quad (18)$$

式中:strategy^{*l*}、strategy^{*l-1*} 分别表示第 l 、 $l-1$ 代种群的具体变异策略,mutation(1) 对应 rand1 或者 best1, mutation(2) 对应 rand2 或者 best2。若当代的突变因子比上一代大,选择 mutation(2) 增强探索能力;若当代的突变因子比上一代小,选择 mutation(1) 增强开发能力;否则变异策略与上一代保持相同。

2.4 进化监督机制

在进化过程中,将个体多次迭代后仍没有进化的问题视为进化停滞。为避免此问题,将种群个体根据适应度进行排名,排名靠后的个体视为非精英群体。迭代过程中若非精英群体的个体 X_i^l 连续 NS_i

代未进化,且 NS_i 大于停滞界限 NS_{max} , 对个体 X_i^l 的每个维度施加小概率突变措施,以解决个体进化停滞问题,如下式所示:

$$X_{i,j}^l = \begin{cases} lb_j + (ub_j - lb_j)\text{rand}(0, 1), \text{rand}_j \leq \varphi \\ X_{i,j}^l, \text{其他} \end{cases} \quad (19)$$

式中:rand_{*j*} 是 $[0, 1]$ 内的随机数; φ 为控制个体维度突变的调节参数。 φ 过大停滞进化个体容易偏离正确的进化方向,过小不利于解决进化停滞问题,因此设置为 0.5。非精英个体通过式(19)优化后,将 NS_i 重置为 0。

3 基于强化学习的改进 DE 算法

本节主要介绍基于深度确定性策略梯度算法的强化学习方法,以优化改进 DE 算法的进化参数,如图 7 所示。以种群多样性和最佳个体的适应度进化率等种群特征作为 DDPG 方法输入信息,基于当前父代与子代的进化反馈设计奖励函数,最大化种群进化后最优个体适应度的变化率。DDPG 智能体迭代学习后输出当前父代种群的进化参数,以实现快速收敛至全局最优区域。

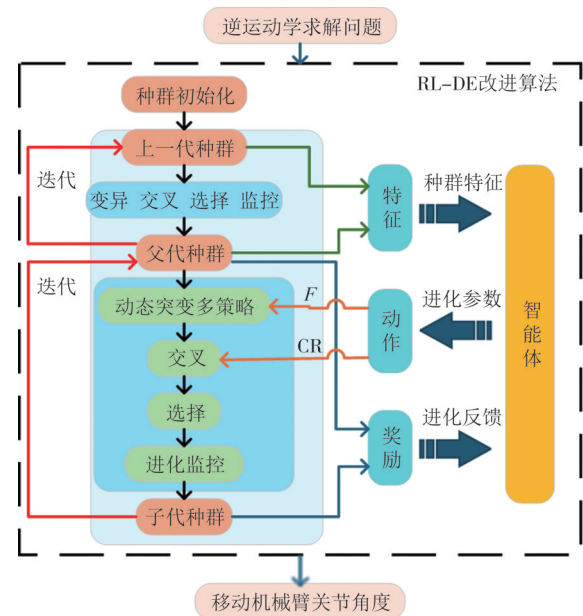


图7 基于RL-DE改进算法的移动机械臂逆运动学求解方法

Fig.7 The inverse kinematics solution method for mobile robotic arm based on the improved RL-DE algorithm

3.1 DDPG 强化学习方法

改进 DE 算法的关键是控制变异与交叉进化过程。交叉操作如式(20)所示:

$$U_{i,j}^l = \begin{cases} V_{i,j}^l, \text{rand}(0, 1) \leq \text{CR}_i^l \text{ 或 } j = j_{\text{rand}} \\ X_{i,j}^l, \text{其他} \end{cases} \quad (20)$$

式中: $U_{i,j}^l$ 表示交叉个体 i 第 j 维度的交叉结果; CR_i^l 表示交叉率, 是范围在 $[0, 1]$ 内的实数; j_{rand} 表示区间在 $[1, \text{ps}]$ 内的随机整数. 进化过程中需要突变因子与交叉率控制个体的进化程度. 突变因子与交叉率越大, 产生的子代个体相对亲本个体的变异程度越大, 反之亦然.

DDPG 是一种无模型、非策略的 RL 算法, 其采用演员-评论家的方法解决连续动作的输出问题, 利用贝尔曼方程存储 Q 函数, 继而用 Q 函数学习动作策略^[21], 如图 8 所示. 其中, $Q(s, a)$ 表示在状态 s 下执行动作 a 获得的预期累积回报, $Q(s', a')$ 表示在下一状态 s' 下执行对应动作 a' 获得的预期累积回报.

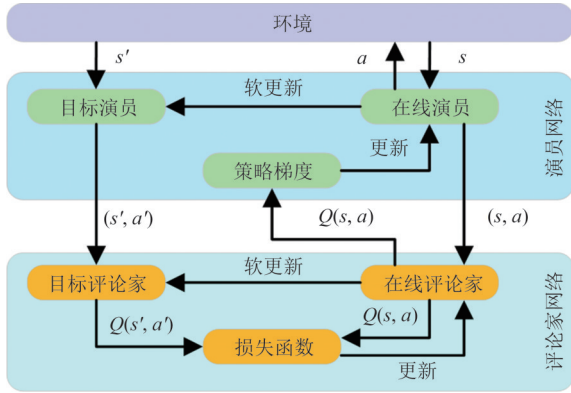


图 8 DDPG 结构图

Fig.8 The structural diagram of DDPG

本文在改进 DE 算法迭代过程中融合 DDPG 强化学习, 将当前种群与上一代种群的适应度与多样性特征作为输入状态, 突变因子与交叉率作为连续动作输出. DDPG 输入状态为: 种群适应度特征 FC 与多样性特征 PC, 如下式计算:

$$\text{state1: PC} = \sqrt{\frac{1}{\text{ps}} \sum_{i=1}^{\text{ps}} \sum_{j=1}^{\text{dim}} \left(\frac{X_{i,j}^{l+1} - X_{i,j}^l}{\text{ub}_j - \text{lb}_j} \right)^2} \quad (21)$$

$$\text{state2: FC} = \sqrt{\frac{1}{\text{ps}} \sum_{i=1}^{\text{ps}} \left(\frac{f(X_i^{l+1}) - f(X_i^l)}{f(X_i^l)} \right)^2}$$

式中: $f(\cdot)$ 表示适应度函数. DDPG 输出动作为: 第 $l+1$ 代种群的突变因子 F^{l+1} 与交叉率 CR^{l+1} , $F^{l+1}, \text{CR}^{l+1} \in [0, 1]$. 若范围超出 $[0, 1]$, 则采用 $\text{rand}(0, 1)$ 随机生成一个 $[0, 1]$ 范围内的实数代替超出的动作.

为优化改进 DE 算法的收敛速度, DDPG 奖励函

数考虑子代种群进化后最优个体适应度的变化率. level 表示第 $l+1$ 代到第 $l+2$ 代最优个体适应度的进化水平, 如:

$$\text{level} = \frac{f(X_{\text{best}}^{l+1}) - f(X_{\text{best}}^{l+2})}{f(X_{\text{best}}^{l+1})} \quad (22)$$

式中: X_{best}^{l+1} 表示第 $l+1$ 代的最优个体, X_{best}^{l+2} 表示第 $l+2$ 代的最优个体. 奖励函数设计如下:

$$r^{l+1} = \begin{cases} 20 \times \text{level}, & \text{level} \geq 0.6 \\ 16 \times \text{level}, & \text{level} \in [0.5, 0.6) \\ 12 \times \text{level}, & \text{level} \in [0.4, 0.5) \\ 8 \times \text{level}, & \text{level} \in [0.3, 0.4) \\ 6 \times \text{level}, & \text{level} \in [0.2, 0.3) \\ 4 \times \text{level}, & \text{level} \in [0.1, 0.2) \\ 2 \times \text{level}, & \text{其他} \end{cases} \quad (23)$$

由式(23)可知, 智能体输出的进化参数使当前种群最佳个体的适应度值到下一代最佳个体的适应度值变化率越大, 获得的奖励越高. 在迭代前期, 种群的搜索空间大, 每一代种群根据智能体的学习效果向奖励最大的方向进化, 实现快速收敛到全局最优区域; 在迭代后期, 种群的搜索空间缩小, 智能体选择使最佳个体的适应度值变化率最大的进化参数, 进一步开发当前区域. 由式(17)与式(18)可知, 输出的突变因子 F 若升高, 则智能体判定算法需要加强探索能力, F 若降低, 则算法需要对优秀个体展开进一步开发, 从而选择对应变异算子. 这种以最优个体适应度进化率为奖励导向的 RL-DE 改进算法建立突变因子与突变策略的联系, 解决 DE 算法进化参数的选取问题.

DDPG 的策略网络与评价网络均采用 2 层全连接层, 每层的神经元个数为 256, 采用 relu 激活层.

3.2 RL-DE 改进算法训练流程

选取式(8)作为适应度函数, 对 RL-DE 改进算法的智能体进行训练, 伪代码如表 2 所示.

4 实验与结果

为验证本文算法的优越性, 首先设计 200 个目标位姿点逆运动学求解实验, 将本文方法与已有的算法进行对比, 测试离散位姿点求解的成功率及误差性能; 其次, 选取上述实验表现最佳的对比算法与本文算法进行 3 种不同轨迹的跟踪仿真对比实验, 验证算法轨迹跟踪误差及关节平滑性等性能. 仿真实验开发环境为 MATLAB R2023b, 其运行平台

表2 RL-DE改进算法伪代码

Tab.2 The pseudo-code of the improved RL-DE algorithm

伪代码:RL-DE改进算法学习过程	
1:	初始化DE的迭代次数 maxIter, $F^0=0.5$, $CR^0=0.5$
2:	初始化DDPG网络权重 θ , 折扣因子 γ , 学习率 α ,
	训练回合 episode, 训练步 step
3:	对于每回合 episode = 1: 2 000
4:	初始化DE种群 X^0
5:	$X^i = X^0$
6:	$F^i = F^0$, $CR^i = CR^0$
7:	X^i 变异、交叉、选择生成子代种群 X^{i+1}
8:	计算种群 X^i 、 X^{i+1} 适应度 f^i 、 f^{i+1}
9:	根据式(21)计算状态 $state^{i+1}=[PC, FC]$
10:	对于每个训练步 step = 1: maxIter
11:	将 $state^{i+1}$ 输入 DDPG 智能体, 得到动作输出
	$action^{i+1}=[F^{i+1}, CR^{i+1}]$
12:	X^{i+1} 变异、交叉、选择生成子代种群 X^{i+2}
13:	根据式(8)计算种群 X^{i+1} 、 X^{i+2} 适应度 f^{i+1} 、 f^{i+2}
14:	根据式(21)计算状态 $state^{i+2}=[PC, FC]$
15:	根据式(22)和式(23)计算奖励 r^{i+1}
16:	存储 $[state^{i+1}, action^{i+1}, r^{i+1}, state^{i+2}]$ 到经验池中
17:	$state^{i+1} \leftarrow state^{i+2}$
18:	$X^{i+1} \leftarrow X^{i+2}$
19:	定期软更新 DDPG 网络权重 θ

为 Windows 10 64 bit 系统的个人电脑 (CPU i7-4790, RAM 16 GB)。

4.1 目标位姿点实验

为保证目标位姿点在移动机械臂的工作范围内, 使用蒙特卡罗方法随机生成 200 组关节角度, 如下所示^[2]:

$$\begin{cases} \hat{q}_j = lb_j + \text{rand}(0, 1)(ub_j - lb_j) \\ \hat{q} = [\hat{q}_1, \hat{q}_2, \dots, \hat{q}_j, \dots, \hat{q}_{10}] \end{cases} \quad (24)$$

式中: ub_j 和 lb_j 分别表示关节 j 运动范围上、下界。据式(3)正运动学得到位姿转化矩阵 ${}^0\hat{T}_n(\hat{q})$, 优化问题表示为:

$$\begin{aligned} \min f_1({}^0\hat{T}_n(\hat{q}), q) \\ \text{s.t. } q \in [lb, ub] \end{aligned} \quad (25)$$

对比算法及其参数设置如下: 基于知识的人工蜂群(KABC)算法的雇佣蜂群设置为 50, 观察蜂群设置为 50, 停滞极限设置为 $(ps \times \dim)/2$ ^[22]; 收缩因子粒子群优化(CFPSO)算法的粒子种群设置为 50, 认知因子 $c_1=2.05$, 社会因子 $c_2=2.05$, 收缩因子 $\phi = 4.1$ ^[23]; 布谷鸟搜索(cuckoo search, CS)算法的发现概率 $Pa=0.25$ ^[24]; DE^[2]与自适应差分进化^[16](self-adaptive differential evolution, SDE)的变异因子与交叉率分别设置

为 0.4 和 0.9; 外部存档自适应差分进化(adaptive differential evolution with optimization external archive, ADE)算法的变异因子均值 $\mu_F=0.6$, 交叉率均值 $\mu_{CR}=0.85$ ^[25]; 自适应调参差分进化(self-adapting control parameters in differential evolution, SADE)算法的变异因子 $F=0.1+0.9\text{rand}(0, 1)$, 交叉率 $CR=\text{rand}(0, 1)$, 种群设置为 50^[26]。本文提出的 RL-DE 改进算法种群设置为 50, 所有算法迭代次数设置为 1 000。利用式(8)计算末端位姿误差小于 1×10^{-10} m 的逆解视为成功解。定义成功率表示如下:

$$\text{suc_rate} = \frac{n_{\text{suc}}}{n_{\text{total}}} \times 100\% \quad (26)$$

式中: n_{suc} 表示获得成功解的目标位姿点个数; n_{total} 表示所有目标位姿点个数。

结果如表 3 和表 4 所示。由表 3 可知, KABC、CFPSO 和 CS 算法的成功率均为 0, DE 和 SDE 算法成功率均大于 10%, 由此可知差分进化算法在求解移动机械臂的逆运动学问题上具有显著优势。本文方法误差均值、误差最大值、成功率求解均为最佳, 误差最小值与最佳的 SDE 算法表现相当, 由表 4 可知, 本文方法通过强化学习获得种群进化参数, 所需计

表3 随机点对比结果

Tab.3 The comparison of random points

算法	误差/m			成功率/%
	最小值	均值	最大值	
KABC ^[22]	8.93×10^{-5}	1.05×10^{-1}	0.52	0
CFPSO ^[23]	9.30×10^{-5}	4.13×10^{-1}	1.19	0
CS ^[24]	3.91×10^{-4}	3.90×10^{-3}	0.03	0
DE ^[2]	1.88×10^{-16}	1.70×10^{-3}	0.17	49.5
SDE ^[16]	9.79×10^{-17}	7.30×10^{-3}	0.80	92.0
ADE ^[25]	2.22×10^{-13}	3.10×10^{-3}	0.44	11.5
SADE ^[26]	1.85×10^{-16}	3.40×10^{-3}	0.67	30.0
本文方法	1.10×10^{-16}	1.72×10^{-4}	0.04	99.5

表4 计算时间对比

Tab.4 The computational time comparison

算法	平均时间/s
KABC ^[22]	37.509 4
CFPSO ^[23]	17.836 4
CS ^[24]	35.809 4
DE ^[2]	17.974 4
SDE ^[16]	34.883 5
ADE ^[25]	18.260 7
SADE ^[26]	18.886 4
本文方法	36.091 4

算资源大,因此目标位姿点逆运动学的平均求解时间较长,为 36.091 4 s.此外,SDE 算法与本文算法成功率较高,分别为 92.0% 和 99.5%,两种算法的误差分布如图 9 所示.由图 9 可知,以 0 作为基准,SDE 算法的误差分布离散,且最大值超过 0.8 m,本文方法的误差分布更集中在 0 附近,最大值不超过 0.05 m,误差的方差更小.如图 10 所示为不同算法在求解 4 个随机目标位姿点逆运动学问题时的迭代过程,可见,本文方法具备更突出的寻优能力与收敛速度.

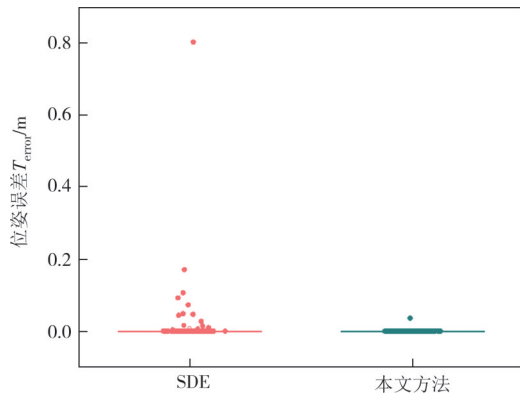


图 9 位姿误差对比箱型图

Fig.9 The box diagram of pose error comparison

4.2 轨迹跟踪实验

为进一步验证本文算法的有效性,对 SDE 算法与 RL-DE 改进算法进行轨迹跟踪实验.选取 3 种复杂程度不同的轨迹进行对比实验,每条轨迹被分段,取 $k=1,2,3,\dots,201$,共 201 个目标点,轨迹方程如下:

轨迹 1:

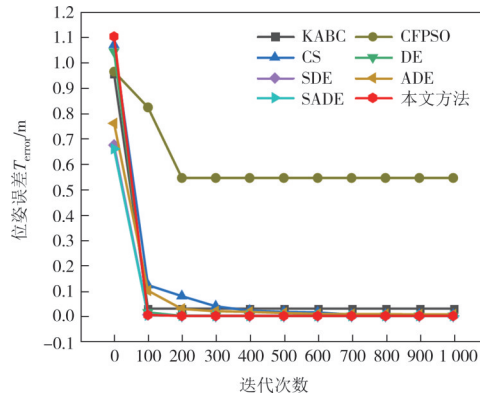
$$\begin{cases} \theta_k = 0: \pi/100: 2\pi \\ x_k = 0.5 \\ y_k = 0.2\cos(\theta_k) \\ z_k = 0.55 + 0.2\sin(\theta_k) \end{cases} \quad (27)$$

轨迹 2:

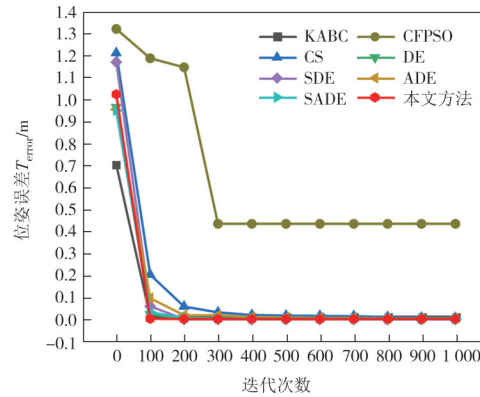
$$\begin{cases} \theta_k = 0: \pi/100: 2\pi \\ r_k = 0.15 + 0.08\cos(3\theta_k) \\ x_k = 0.4 \\ y_k = r_k \cos(r_k) \\ z_k = 0.5 + r_k \sin(\theta_k) \end{cases} \quad (28)$$

轨迹 3:

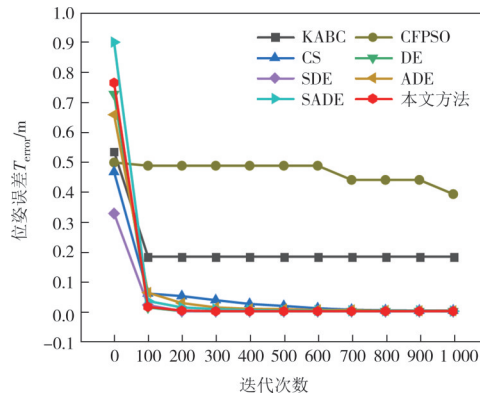
$$\begin{cases} y_k = -0.5: 0.005: 0.5 \\ x_k = 0.4 \\ z_k = 0.5 + 0.1\sin(30y_k) \end{cases} \quad (29)$$



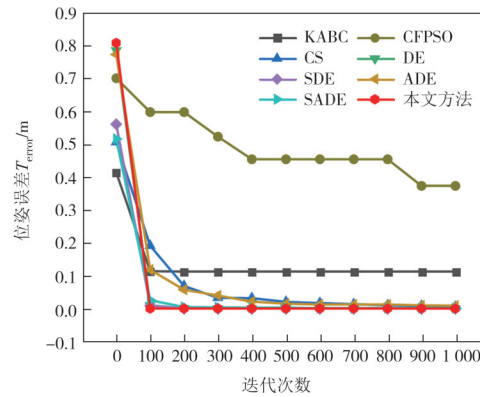
(a) 随机目标点 1



(b) 随机目标点 2



(c) 随机目标点 3



(d) 随机目标点 4

图 10 不同算法迭代曲线

Fig.10 Iterative curves of different algorithms

式中: (x_k, y_k, z_k) 表示第 k 个目标点坐标. 轨迹 1、2、3 分别为圆曲线、玫瑰曲线和正弦曲线, 如图 11 所示.

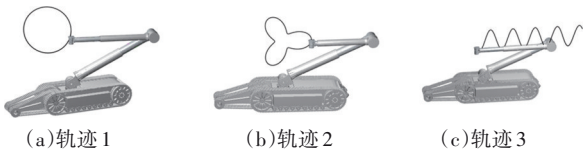


图 11 3 种轨迹曲线示意图
Fig.11 The illustration of three trajectory curves

在对比实验中, 规定所有跟踪点 ${}^0\hat{T}_n$ 的方向矩阵 R 为:

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

将轨迹跟踪转化为最小优化问题, 定义如下:

$$\begin{aligned} \min f_2({}^0\hat{T}_n, \boldsymbol{q}, \boldsymbol{q}_{\text{old}}) \\ \text{s.t. } \boldsymbol{q} \in [\text{lb}, \text{ub}] \end{aligned} \tag{30}$$

式中: $\alpha = 1, \beta = 0.05$. 在求解 k 跟踪点的关节变量 $\boldsymbol{q}_k$ 时, 将 $\boldsymbol{q}_{\text{old}}$ 设置为 $k-1$ 跟踪点的关节变量 $\boldsymbol{q}_{k-1}$. 为增加关节的平滑性, 在轨迹跟踪中进一步降低关节突变, 在求出 $k-1$ 跟踪点的逆解后, 保存其最后一代种群中排名前 10 的优秀个体. 求解 $k(k > 1)$ 跟踪点的逆解时, 在种群初始化步骤加入 $k-1$ 跟踪点保存的优秀个体, 如图 12 所示. 由于相邻两个跟踪点在三维空间的欧氏距离很小, 上述优秀个体的保存过程可以加快求解当前跟踪点逆运动学问题的收敛速度.

本文方法与 SDE 参数设置与目标位姿点实验相

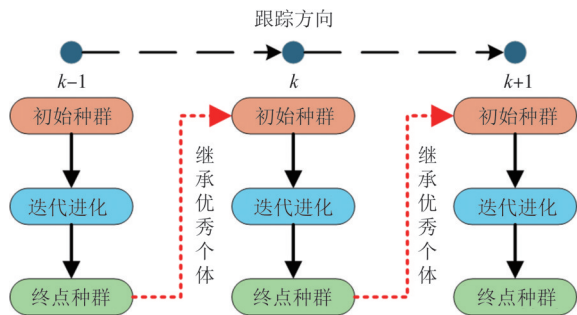


图 12 轨迹跟踪设置
Fig.12 Trajectory tracking settings

同, 两者适应度函数加入关节变化权重. 通过式(8) 计算每个跟踪点的最终位姿误差. 实验结果以位姿误差不超过 $1 \times 10^{-5} \text{ m}$ 的跟踪点视为成功点, 并通过式(26)计算轨迹跟踪的成功率, 误差结果如图 13 所示. 由图 13 可知在 3 种轨迹跟踪实验中, 本文方法的误差波动更小.

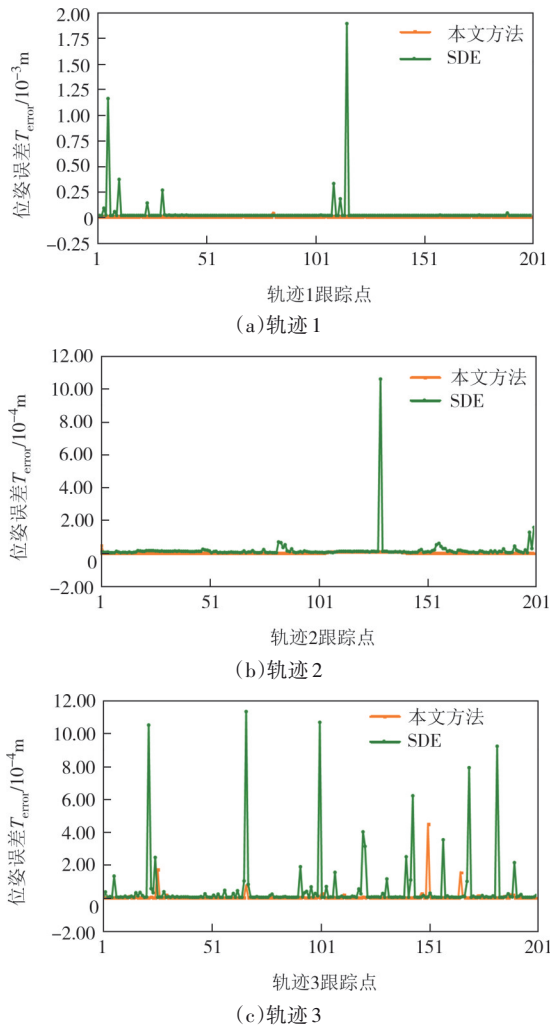


图 13 跟踪点误差
Fig.13 The error of tracking points

表 5 为跟踪点误差对比. 由表 5 可知, 本文方法的轨迹跟踪实验跟踪成功率均在 95% 以上, 轨迹 2 的跟踪成功率为 99.50%, 轨迹 3 的跟踪成功率为

表 5 跟踪点误差对比

Tab.5 The comparison of tracking point error

评估参数	SDE				本文方法			
	误差/m			成功率/%	误差/m			成功率/%
	最大值	均值	最小值		最大值	均值	最小值	
轨迹 1	1.9×10^{-3}	3.4×10^{-5}	1.0×10^{-5}	0	6.4×10^{-5}	8.8×10^{-6}	4.2×10^{-6}	95.02
轨迹 2	1.1×10^{-3}	1.3×10^{-5}	3.7×10^{-16}	80.60	4.6×10^{-5}	1.5×10^{-6}	4.0×10^{-16}	99.50
轨迹 3	1.1×10^{-4}	4.5×10^{-6}	3.3×10^{-16}	91.54	4.4×10^{-5}	5.2×10^{-7}	3.4×10^{-16}	98.51

98.51%. SDE算法各轨迹的跟踪成功率均比本文方法低,且轨迹1的跟踪成功率为0.

图14给出了本文方法和SDE算法轨迹跟踪的三维空间图,绿色点表示成功跟踪点,红色点表示误差超过 $1 \times 10^{-5} \text{ m}$ 的失败跟踪点.图15为3种轨迹跟踪的关节角度变化曲线.为评估关节运动轨迹平滑度,基于角度变化特征提出评价指标 $D_s^{[27]}$:

$$D_s = \frac{\sum_{i=1}^{N-1} |\Delta q(i)|}{N-1} \quad (31)$$

式中: $\Delta q(i)$ 表示相邻跟踪点的关节角度差; N 为轨迹跟踪点个数. D_s 数值越小,表明关节的平滑度性能越好.表6为轨迹跟踪实验移动机械臂关节运动的平滑度评价结果.综合图15和表6可知本文方法的关节轨迹比SDE算法的关节轨迹更平滑.

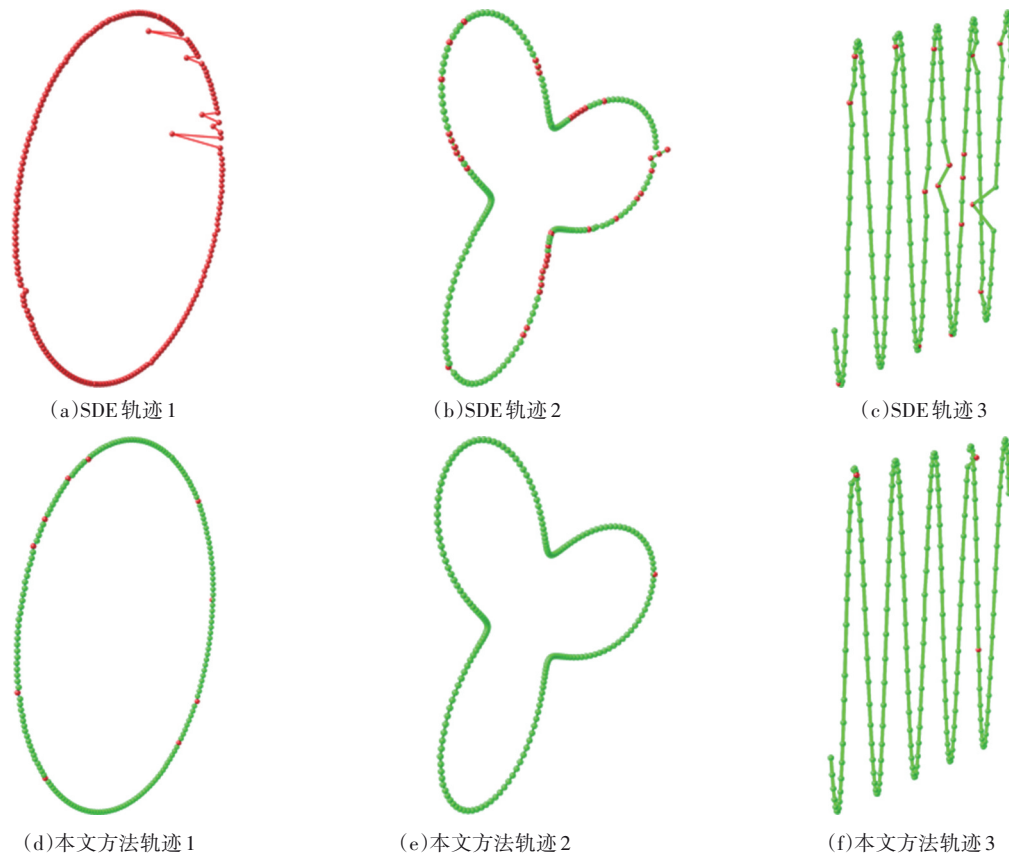


图14 轨迹跟踪对比

Fig.14 The comparison of tracking track

表6 关节运动平滑度

Tab.6 The smoothness of joint motion

关节	轨迹1		轨迹2		轨迹3	
	本文方法	SDE	本文方法	SDE	本文方法	SDE
p_x/mm	2.12	2.65	1.98	2.20	4.17	4.15
p_y/mm	4.71×10^{-1}	2.14×10^{-1}	6.45×10^{-2}	1.87×10^{-1}	8.64×10^{-2}	1.22×10^{-1}
$\theta_z/(\circ)$	2.36×10^{-2}	9.50×10^{-2}	3.90×10^{-3}	4.83×10^{-2}	6.70×10^{-3}	7.27×10^{-2}
$\theta_l/(\circ)$	9.00×10^{-5}	1.41×10^{-4}	4.01×10^{-2}	5.68×10^{-2}	6.41×10^{-2}	1.32×10^{-1}
$\theta_1/(\circ)$	1.30×10^{-2}	1.06×10^{-1}	2.40×10^{-3}	3.69×10^{-2}	6.31×10^{-4}	4.90×10^{-3}
$\theta_2/(\circ)$	3.03×10^{-1}	2.96×10^{-1}	3.32×10^{-1}	3.17×10^{-1}	6.07×10^{-1}	6.60×10^{-1}
p_3/mm	1.98	2.01	2.32	2.71	6.00	4.03
$\theta_4/(\circ)$	3.03×10^{-1}	2.96×10^{-1}	3.01×10^{-1}	2.97×10^{-1}	5.55×10^{-1}	6.11×10^{-1}
p_5/mm	1.04	2.04×10^{-1}	1.95	1.68	1.67	1.73
$\theta_6/(\circ)$	1.53×10^{-2}	5.39×10^{-2}	4.20×10^{-3}	4.61×10^{-2}	6.80×10^{-3}	7.43×10^{-2}

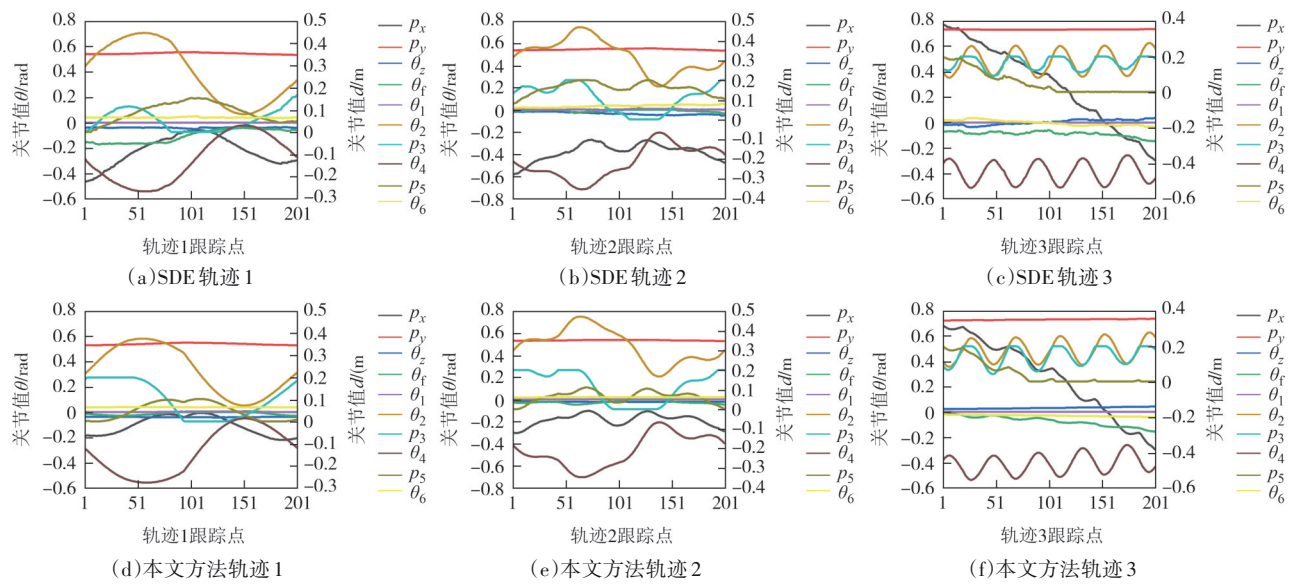


图15 移动机械臂关节变化结果

Fig.15 The joints variation results of mobile robotic arms

5 结论

本文针对移动机械臂逆运动学高效求解问题,提出了一种融合RL与DE算法的改进算法,优化DE算法种群进化方向和收敛速度.基于进化参数动态设定种群突变策略,调整DE算法不同进化阶段的全局搜索与局部优化方向,保证算法全局寻优能力.加入进化监督机制优化停滞进化个体,有助于降低算法陷入局部最优的风险.采用DDPG强化学习方法以种群最优个体适应度变化率为奖励函数优化改进DE算法进化参数,解决进化参数固定的局限性问题以提高DE算法的全局收敛性能.两组对比实验结果表明:相比于KABC、CFPSO、CS、SDE等智能启发算法,本文方法目标点逆运动学求解成功率最高,平均误差最小,算法收敛性能优越;相比于SDE算法,本文方法在不同目标轨迹跟踪场景下逆运动学求解平均误差更小,跟踪成功率更高,关节运动轨迹更为平滑.

综上所述,本文提出的RL-DE改进算法可有效提升移动机械臂逆运动学求解成功率和降低跟踪误差.

参考文献

[1] CHEN F, SELVAGGIO M, CALDWELL D G. Dexterous grasping by manipulability selection for mobile manipulator with visual guidance[J]. IEEE Transactions on Industrial Informatics, 2019,

15(2):1202-1210.

- [2] LI M H, QIAO L J. A review and comparative study of differential evolution algorithms in solving inverse kinematics of mobile manipulator[J]. Symmetry, 2023, 15(5):1080.
- [3] 李翠明, 任柯州, 龚俊. 基于改进遗传算法的清洁移动机械臂运动学逆解[J]. 太阳能学报, 2022, 43(7):180-185.
- LI C M, REN K Z, GONG J. Inverse kinematics solution of cleaning mobile manipulator based on improved genetic algorithm [J]. Acta Energiæ Solaris Sinica, 2022, 43(7):180-185. (in Chinese)
- [4] ZHAO G J, TAO B, JIANG D, et al. A closed-form inverse kinematic analytical method for seven-DOF space manipulator with aspheric wrist structure[J]. Machines, 2024, 12(9):632.
- [5] COLAN J, DAVILA A, HASEGAWA Y. Variable step sizes for iterative Jacobian-based inverse kinematics of robotic manipulators[J]. IEEE Access, 2024, 12:87909-87922.
- [6] WOLIŃSKI Ł, WOJTYRA M. A novel QP-based kinematic redundancy resolution method with joint constraints satisfaction [J]. IEEE Access, 2022, 10:41023-41037.
- [7] BAI Y H, LUO M Z, PANG F L. An algorithm for solving robot inverse kinematics based on FOA optimized BP neural network [J]. Applied Sciences, 2021, 11(15):7129.
- [8] AGGARWAL L, AGGARWAL K, URBANIC R J. Use of artificial neural networks for the development of an inverse kinematic solution and visual identification of singularity zone (s) [J]. Procedia CIRP, 2014, 17:812-817.
- [9] RAM R V, PATHAK P M, JUNCO S J. Inverse kinematics of mobile manipulator using bidirectional particle swarm optimization by manipulator decoupling[J]. Mechanism and Machine Theory, 2019, 131:385-405.
- [10] NGUYEN-VAN S, LIEU Q X, XUAN-MUNG N, et al. A new study on optimization of four-bar mechanisms based on a hybrid-

- combined differential evolution and jaya algorithm[J]. *Symmetry*, 2022, 14(2):381.
- [11] LIU Y Y, XI J L, BAI H F, et al. A general robot inverse kinematics solution method based on improved PSO algorithm[J]. *IEEE Access*, 2021, 9:32341–32350.
- [12] KHALEEL H Z. Inverse kinematics solution for redundant robot manipulator using combination of GA and NN[J]. *Al-Khwarizmi Engineering Journal*, 2018, 14(1):136–144.
- [13] WONG C C, FENG H M, LAI Y C, et al. Ant colony optimization and image model-based robot manipulator system for pick-and-place tasks[J]. *Journal of Intelligent & Fuzzy Systems*, 2019, 36(2): 1083–1098.
- [14] LOPEZ-FRANCO C, HERNANDEZ-BARRAGAN J, ALANIS A Y, et al. A soft computing approach for inverse kinematics of robot manipulators[J]. *Engineering Applications of Artificial Intelligence*, 2018, 74:104–120.
- [15] LÓPEZ-FRANCO C, HERNÁNDEZ-BARRAGÁN J, ALANIS A Y, et al. Inverse kinematics of mobile manipulators based on differential evolution[J]. *International Journal of Advanced Robotic Systems*, 2018, 15:1729881417752738.
- [16] HERNANDEZ-BARRAGAN J, LOPEZ-FRANCO C, ARANA-DANIEL N, et al. Inverse kinematics for cooperative mobile manipulators based on self-adaptive differential evolution[J]. *PeerJ Computer Science*, 2021, 7: e419.
- [17] ZHANG Q Q, WANG D Q, GAO L F. Research on the inverse kinematics of manipulator using an improved self-adaptive mutation differential evolution algorithm[J]. *International Journal of Advanced Robotic Systems*, 2021, 18(3):17298814211014413.
- [18] MIN C, ZHANG M, ZHANG Q X, et al. A two-stage adaptive differential evolution algorithm with accompanying populations[J]. *Mathematics*, 2025, 13(3):440.
- [19] 蔡自兴, 谢斌. 机器人学[M]. 3版. 北京:清华大学出版社, 2015:40–42.
- CAI Z X, XIE B. *Robotics* [M]. 3rd ed. Beijing: Tsinghua University Press, 2015: 40–42. (in Chinese)
- [20] YANG Q Y, CHU S C, PAN J S, et al. Dynamic multi-strategy integrated differential evolution algorithm based on reinforcement learning for optimization problems[J]. *Complex & Intelligent Systems*, 2024, 10(2):1845–1877.
- [21] SUMIEA E H, ABDULKADIR S J, ALHUSSIAN H S, et al. Deep deterministic policy gradient algorithm: a systematic review[J]. *Heliyon*, 2024, 10(9): e30697.
- [22] BAI J H, LI J J, WANG M J. An efficient artificial bee colony algorithm for analytical inverse kinematics of redundant manipulator [C]//7th International Conference on Intelligent Robotics and Control Engineering (IRCE). August 7–9, 2024, Xi'an, China. IEEE, 2024:117–123.
- [23] COLLINSM T J, SHEN W M. Particle Swarm Optimization for high-DOF inverse kinematics [C]//3rd International Conference on Control, Automation and Robotics (ICCAR). April 24–26, 2017, Nagoya, Japan. IEEE, 2017:1–6.
- [24] LI M H, LUO X, QIAO L J. Inverse kinematics of robot manipulator based on BODE-CS algorithm[J]. *Machines*, 2023, 11(6): 648.
- [25] ZHANG J Q, SANDERSON A C. JADE: adaptive differential evolution with optional external archive[J]. *IEEE Transactions on Evolutionary Computation*, 2009, 13(5): 945–958.
- [26] BREST J, GREINER S, BOSKOVIC B, et al. Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems[J]. *IEEE Transactions on Evolutionary Computation*, 2006, 10(6): 646–657.
- [27] 朱特浩, 赵群飞, 夏泽洋. 利用 Kinect 的人体动作视觉感知算法[J]. *机器人*, 2014, 36(6):647–653.
- ZHU T H, ZHAO Q F, XIA Z Y. A visual perception algorithm for human motion by a kinect[J]. *Robot*, 2014, 36(6):647–653. (in Chinese)