

机器人泛化轨迹的分段动态运动基元平滑避障方法研究

苏永彬, 刘瞰东[†]

(厦门大学 萨本栋微米纳米科学技术研究院, 福建 厦门 361102)

摘要: 针对现有的基于动态运动基元的轨迹避障算法无法同时满足轨迹的特征保留性和平滑性的问题, 本文提出了一种基于分段动态运动基元的轨迹平滑避障方法. 该方法将障碍物区域的轨迹进行分段处理, 分别构建动态运动基元模型, 并在障碍物之外寻找一个融合点进行分段泛化, 从而得到保留原轨迹特征的避障轨迹. 在此基础上, 采用基于虚拟目标点的轨迹实时平滑方法来处理过渡点的拐角, 得到平滑的避障轨迹. 为验证方法有效性, 设计了仿真实验和实际六轴机器人的轨迹避障实验, 并与现有的改进动态运动基元算法进行对比. 实验结果表明, 本文方法生成的避障轨迹不仅能更好地保留原轨迹的特征, 而且在轨迹的平滑性上也更有优势, 从而验证了该方法的有效性.

关键词: 机器人; 动态运动基元; 平滑避障; 演示学习; 轨迹规划

中图分类号: TP241 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2025.0001

引用格式: 苏永彬, 刘瞰东. 机器人泛化轨迹的分段动态运动基元平滑避障方法研究 [J]. 控制与决策, xxxx, x(x): xxxx-xxxx.

The research on the piecewise dynamic movement primitive smoothing obstacle avoidance method for robot generalized trajectories

SU Yong-bin, LIU Tun-dong[†]

(Pen-Tung Sah Institute of Micro-Nano Science and Technology, Xiamen University, Fujian Xiamen 361102, China)

Abstract: In response to the issue that existing trajectory obstacle avoidance algorithms based on dynamic movement primitives (DMPs) cannot simultaneously preserve trajectory characteristics and ensure smoothness, this paper proposes a trajectory smoothing and obstacle avoidance method based on piecewise dynamic movement primitives. The method segments the trajectory in the obstacle regions, constructs dynamic movement primitive models for each segment, and then finds a fusion point outside the obstacle to perform piecewise generalization, thereby obtaining an obstacle-free trajectory that preserves the characteristics of the original trajectory. On this basis, a real-time trajectory smoothing method based on virtual target points is employed to handle sharp corners at critical points, resulting in a smooth obstacle avoidance trajectory. To validate the effectiveness of the proposed method, both simulation experiments and real-world six-axis robot trajectory avoidance experiments are designed and compared with existing improved dynamic movement primitive algorithms. Experimental results demonstrate that the obstacle avoidance trajectory generated by the proposed method not only better preserves the features of the original trajectory but also has advantages in terms of trajectory smoothness, thereby confirming the effectiveness of the method.

Keywords: robot; dynamic movement primitives; smooth obstacle avoidance; imitation learning; trajectory planning

0 引言

随着工业机器人的应用场景逐渐从流水线式的标准化转向小批量多品种的定制化, 传统的示教再现编程方法已无法满足频繁切换任务轨迹的需求^[1,2]. 相比之下, 机器人的演示学习技术能够灵活地泛化

轨迹, 在应对复杂任务场景、提高编程效率上有着天然的优势, 因此逐渐成为工业领域的重要研究方向之一^[3]. 然而在将演示轨迹泛化到更加复杂的新场景时, 泛化轨迹可能会受到新环境中的障碍物阻挡, 导致安全问题频发^[4]. 人类的演示轨迹通常包含了机器

收稿日期: xxxx-xx-xx; 录用日期: xxxx-xx-xx.

责任编辑: 齐霖.

[†]通信作者. E-mail: ltd@xmu.edu.cn

本文附带电子附录文件, 可登录本刊官网该文“资源附件”区自行下载阅览.

人执行任务的一些优化信息,因此在轨迹避障阶段应尽可能地跟踪原泛化轨迹^[5],并保留原轨迹的特征^[6-9]. 轨迹特征是指为了使机器人顺利完成某些任务而设计的具有特殊形状的路线,例如装配路线^[6]、抓取路线^[10,11]、人机协作路线^[12]等. 同时,在机器人运动过程中,轨迹方向的突变容易导致机器人的抖动,造成系统不稳定,因此在满足上述条件的前提下,应尽可能地保证轨迹的平滑性^[7].

针对泛化轨迹的避障问题,现有方法主要分为两大类:一类是在泛化轨迹的基础上加入经典避障算法进行二次规划;第二类是在轨迹泛化阶段根据障碍物的信息来调整模型参数. 第一类方法通常是在障碍物区域使用传统避障算法,如人工势场法^[13]、RRT 算法^[14,15]、A*算法^[16]等进行二次规划. 然而这类方法仅关注避障的成功率,无法实现对原轨迹特征的保留,并且在规划的起点和终点容易产生拐点,造成轨迹不平滑^[17]. 第二类方法主要包括基于动态运动基元 (DMP)^[18]的方法和基于高斯混合模型 (GMM)^[19]的方法,这类方法在轨迹泛化阶段将障碍物信息输入到模型中,通过调整模型参数来使得生成的泛化轨迹避开障碍物区域,而在不受障碍物影响的区域则保留原轨迹的特征. 例如 Hu 等^[6]提出的基于改进 GMM 的避障算法,通过将受到障碍物影响的高斯簇进行移动,结合轨迹平滑算法生成新的避障轨迹. 然而 GMM 属于多轨迹模型,每次避障都需要通过修改模型参数来重新生成轨迹,计算量大. 相比之下,基于 DMP 的方案由于是单轨迹建模,效率会高很多^[3]. 此外, DMP 能够通过改进模型的非线性项公式来修改特定区域的轨迹走向,因此被广泛应用于泛化轨迹的避障中.

近年来,许多学者研究通过改进 DMP 来实现泛化轨迹的避障,同时实现对原轨迹的形状和特征的保留. 例如 Rai 等人^[20]在 DMP 的公式中增加一个耦合项,这个耦合项可以通过人类演示的避障轨迹来学习和优化,使得机器人能在保留轨迹特征的前提下避开障碍物,然而该方法生成的避障轨迹与原轨迹的误差太大. 为了能够在满足轨迹的特征保留的前提下降低轨迹误差, Zhai 等人^[7]在 DMP 耦合项的基础上增加距离因子和动态逼近算法,使得轨迹在绕过障碍物后尽快回到原轨迹上,降低跟踪误差. 然而这也带来了另一个问题,轨迹绕过障碍物后方向往往需要较大的变化才能回到原轨迹上,造成轨迹平滑性差. 为了提高轨迹的平滑性, Ginesi 等人^[8]在 DMP 的基础上融入了动态势场函数,以实现在障

碍物附近的平滑过渡,然而这种方法常常会导致轨迹特征的丢失,轨迹的跟踪精度也不高. 此外,刘等人^[9]和 Jia 等人^[21]也提出了基于改进 DMP 的机械臂轨迹学习与避障方法,利用 DMP 进行轨迹学习和建模,通过结合经典避障算法或添加耦合项来实现避障. 然而这类方法为了拟合轨迹特征,算法的规划过于保守,因此在避障的同时容易带来较大的跟踪误差.

综上所述,基于二次规划的方法无法保留轨迹的特征,且平滑性差,不适用于实际工业场景中. 而基于调整模型参数的方法无法同时满足特征保留和平滑性,尤其是特征保留^[22],这在机器人执行任务中非常重要. 为此,本文提出了一种基于分段 DMP 的轨迹平滑避障算法. 该方法将障碍物区域的轨迹进行分段处理,分别构建 DMP 模型,并在障碍物之外寻找一个融合点进行分段泛化,以得到保留原轨迹特征的避障轨迹. 在此基础上,采用基于虚拟目标点的轨迹实时平滑方法来处理过渡点的拐角,得到平滑的避障轨迹. 通过仿真实验和实际六轴机器人的轨迹避障实验,验证了本文方法相比于现有的改进 DMP 算法,在轨迹的特征保留性和避障平滑性上都有更强的优势.

1 基于动态运动基元的轨迹泛化方法

1.1 动态运动基元模型概述

DMP 是一种用于机器人运动轨迹学习和控制的模型,能够学习目标轨迹的运动特征,在终点变化的新场景中生成具有原轨迹运动趋势的泛化轨迹. 该方法的核心思想是在简单且稳定的弹簧-阻尼系统中引入非线性函数强迫项,由该强迫项来控制系统的运动过程,使得生成的轨迹保持目标轨迹的形状. 因此 DMP 可以看成是一个 PD 控制器与一个轨迹学习器的叠加,如下公式所示:

$$\ddot{y} = \alpha_y(\beta_y(g - y) - \dot{y}) + \dot{g}. \quad (1)$$

其中 y 表示机器人的位置, \dot{y} 、 \ddot{y} 为机器人对应的速度、加速度, g 为目标点的位置. α_y 和 β_y 为系统参数,通过设置 $\beta_y = \alpha_y/4$ 可以使系统达到临界阻尼,保证系统稳定. 机器人的位置 y 随时间逐渐变化,最终系统收敛于目标位置 g . 最后叠加的一项 \dot{g} 表示轨迹学习器,目的是在收敛过程中改变轨迹的形状为期望的形状.

可通过多个非线性基函数的归一化线性叠加来拟合各种形状的轨迹,因此非线性函数 f 可以通过如下公式进行表示:

$$f(g, x) = \frac{\sum_{i=1}^N \psi_i \omega_i}{\sum_{i=1}^N \psi_i} x(g - y_0). \quad (2)$$

其中 y_0 表示机器人的起始位置, ω_i 表示权重项, ψ_i 是高斯基函数, N 为基函数的个数. 其中高斯基函数定义为:

$$\psi_i = \exp(-h_i(x - c_i)^2). \quad (3)$$

c_i 和 h_i 分别表示高斯基函数的中心位置和方差,如图1所示.

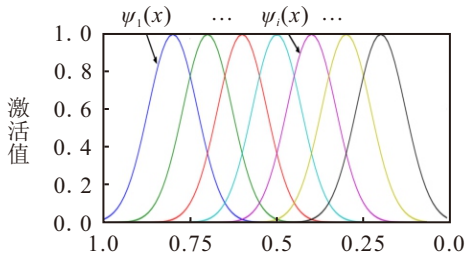


图1 高斯基函数

对每个基函数通过公式(2)来融入权重参数 ω_i 后,可以拟合任意轨迹的形状,如图2所示.

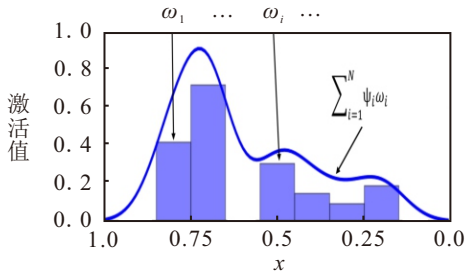


图2 融合权重项后的非线性项

x 为相位变量,用于消除 $f(x, g)$ 对时间的依赖,用一个正则系统表示:

$$\dot{x} = -\alpha_x x. \quad (4)$$

其中 α_x 是一个大于0的常数,在 x 从1逐渐衰减到0的过程中,轨迹也从起始位置收敛到目标点.

1.2 基于动态运动基元的轨迹泛化

为了使演示学习模型生成的作业轨迹能够在新场景中准确地泛化,使用DMP中的强迫项 f 来对目标轨迹进行学习,并通过改变公式(2)中的起点 y_0 和目标点 g 的位置来实现轨迹的泛化.假设目标轨迹 y_d 由一系列的轨迹点组成,定义该轨迹的加速度为 \ddot{y}_d ,则有如下关系:

$$\ddot{y}_d = \frac{\partial y_d}{\partial t} = \frac{\partial}{\partial t} \frac{\partial}{\partial t} y_d. \quad (5)$$

在此基础上,根据公式(1)可以得到生成该目标轨迹所需要的强迫项 f_d :

$$f_d = \ddot{y}_d - \alpha_y(\beta_y(g - y) - \dot{y}) \quad (6)$$

由公式(2)可知,强迫项是通过一系列的高斯基函数加权得到的,因此可以使用局部加权回归来选择基函数的权重,使得强迫项函数与期望轨迹匹配.

假设目标轨迹是一条螺旋向上的曲线,起点坐标为(1,0,0),目标点为(1,0,1),如图3(a)中的蓝色轨迹所示.对该轨迹使用DMP进行学习,随后起点不变,改变目标点位置分别为(0.9,-0.1,0.9)和(1.1,0.0,1.1),得到的两条泛化轨迹如图3(a)的绿色和红色轨迹所示.除了对目标点进行泛化,也可以将整段轨迹在新场景中进行泛化,如图3(b)所示,将起点位置和目标点位置分别改为(1,-1,0)和(1,-2,0),得到的泛化轨迹如图中绿色轨迹所示.

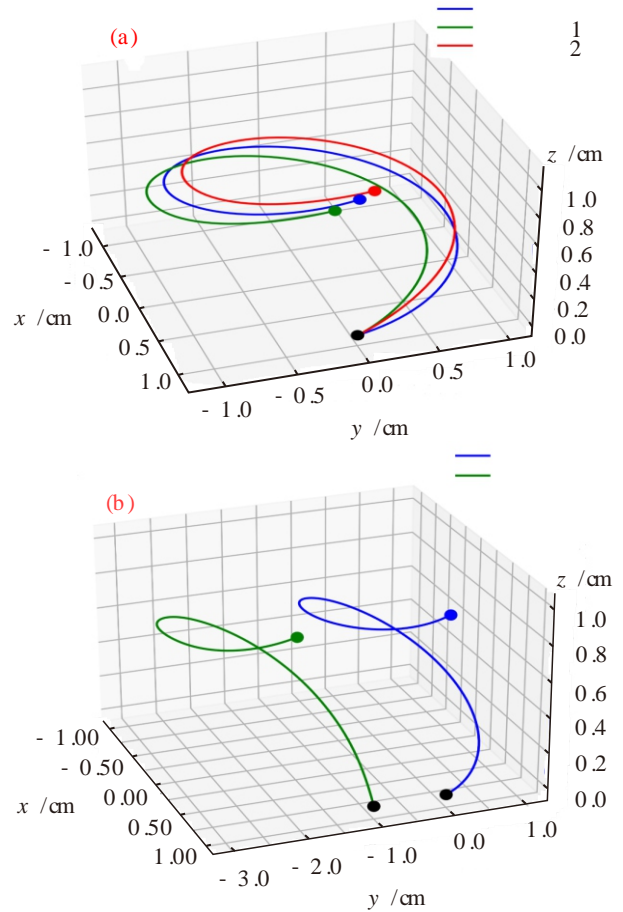


图3 基于动态运动基元的轨迹泛化

2 机器人泛化轨迹的分段避障算法

2.1 分段避障算法的原理

在新的作业场景中,通过DMP算法得到的泛化轨迹可能出现障碍物的干扰,此时要求机器人不仅能够平滑地避开障碍物,还应尽可能地保留泛化轨迹的形状,以满足对演示特征的保留^[6].如下图4所示,假设泛化轨迹在避障临界点A和B之间的一个峰状特征处受到障碍物的阻挡.传统的避障算法如

人工势场法更倾向于生成图中的绿色短虚线避障轨迹, 因为这条路径更短也更容易生成. 然而这条路径却丢失了泛化轨迹的部分特征, 导致机器人无法完成某些任务. 而图中的红色长虚线轨迹既能避开障碍物, 又能保持原泛化轨迹的形状, 因此是更加合适的.

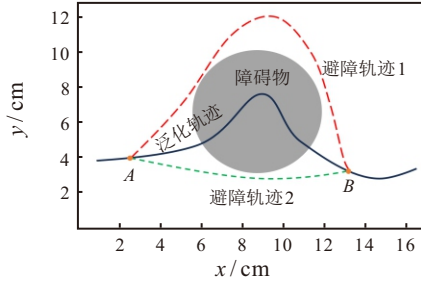


图4 避障轨迹的特征保留度对比

由于 DMP 能够学习轨迹的特征, 并泛化到新的目标点, 本文将采用分段 DMP 来在障碍物区域生成避障轨迹. 首先, 在障碍物边缘选择两个避障临界点 A 和 B, 并在障碍物区域的轨迹上选择一个分段点 C, 然后利用 DMP 对 AC 之间的轨迹进行建模. 得到模型后, 在 C 点的垂直方向上寻找合适的融合点作为新的泛化目标点进行轨迹泛化. 找到泛化轨迹与障碍物没有碰撞的融合点 D, 将得到的泛化轨迹作为前段避障轨迹, 如图 5(a) 的红色轨迹所示. 此避障轨迹保留了 AC 之间的轨迹的特征. 类似地, 同样对 BC 之间的轨迹使用 DMP 进行建模, 并将其泛化到融合点 D, 如图 5(b) 的红色虚线轨迹所示, 将此轨迹作为后段避障轨迹.

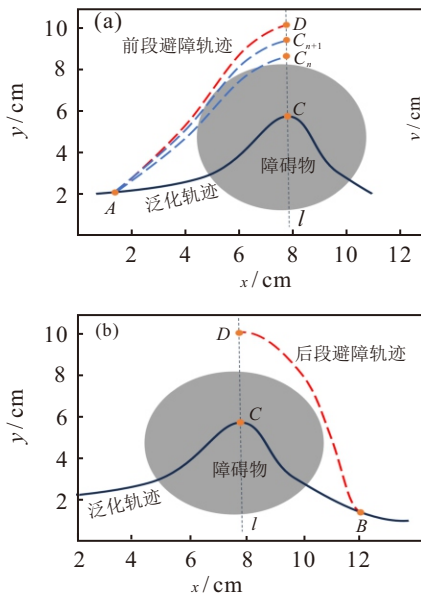


图5 分段避障算法的原理

然而, 在得到无碰撞的前段避障轨迹后, 并不能保证后段的泛化轨迹不会与障碍物发生碰撞. 如果

此时后段轨迹发生了碰撞, 则继续在 C 点(坐标为 (x_C, y_C))的垂直方向上以 c_{step} 为步长寻找无碰撞的融合点, 找到后再进行前段避障轨迹的构建. 假设轨迹 y_{AC} 在 C 点处的垂线方程为:

$$y = -\frac{1}{k}(x - x_C) + y_C. \quad (7)$$

其中 $k = \dot{y}_{AC}(x_C, y_C)$, 表示轨迹在该点的斜率. 此时可以计算出 c_{step} 在 x 和 y 方向上的映射长度:

$$c_x = c_{step} \cdot \frac{k}{\sqrt{1+k^2}}, \quad (8)$$

$$c_y = c_{step} \cdot \frac{-1}{\sqrt{1+k^2}}. \quad (9)$$

定义避障临界点坐标分别为 $A(x_A, y_A)$ 和 $B(x_B, y_B)$, 分段点为 $C(x_C, y_C)$, AC 段轨迹 y_{AC} 及对应的模型 DMP_{AC} , BC 段轨迹 y_{BC} 及对应的模型 DMP_{BC} , 最大迭代次数为 N , 障碍物区域为 O . 计算融合点位置的伪代码如下:

algorithm 1.

input: $A(x_A, y_A), B(x_B, y_B), C(x_C, y_C), y_{AC}, DMP_{AC}, y_{BC}, DMP_{BC}, N, O$

output: $D(x_D, y_D)$

initialize $D = \emptyset$

1: for $i \leftarrow 1$ to N do;

2: $x_{C_{\pm n}} \leftarrow x_C \pm n \cdot c_x, y_{C_{\pm n}} \leftarrow y_C \pm n \cdot c_y$;

3: if $C_{\pm n} \in O$: continue;

4: else:

5: $\{DMP_{AC} \leftarrow C_{\pm n}\} \rightarrow y_{AC_{\pm n}}$;

6: if $y_{AC_{\pm n}} \cap O \neq \emptyset$ or generalized fail:

continue;

7: else: $D = C_{\pm n}$

8: endif;

9: endif;

10: endfor;

11: $\{DMP_{BC} \leftarrow D\} \rightarrow y_{BD}$

12: if $y_{BD} \cap O \neq \emptyset$: repeat 1 – 10 from D;

13: until if $y_{BD} \cap O = \emptyset$ and $y_{AD} \cap O = \emptyset$

得到融合点 D 以及两段避障轨迹 y_{AD} 和 y_{BD} 后, 将两段避障轨迹合在一起, 就能得到保持原泛化轨迹形状的避障轨迹.

2.2 分段点和避障临界点的位置选择

由上一小节可知, 通过合理地设计避障临界点 A 和 B 以及分段点 C 点的位置, 结合 DMP 的强大泛化能力, 可以实现对原轨迹特征的准确拟合. 在这个过程中, 分段点和避障临界点的选择至关重要, 如

果选择不合理, 将导致无法找到合适的融合点使得前后段轨迹均满足避障要求. 假设泛化轨迹与障碍物的两个交点为 M 和 N, 其坐标分别为 (x_M, y_M) 和 (x_N, y_N) , M 和 N 之间的轨迹点为 $[(x_k, y_k), (x_{k+1}, y_{k+1}), \dots, (x_{k+n_{MN}}, y_{k+n_{MN}})]$, 其中 n_{MN} 表示轨迹点总数. 分别计算每个点到 MN 所在直线的距离, 计算公式如下:

$$d_k = \frac{|x_k(y_{N-M}) - y_k(x_{N-M}) + (x_N y_M - x_M y_N)|}{\sqrt{(y_N - y_M)^2 + (x_N - x_M)^2}} \quad (10)$$

其中 y_{N-M} 表示 $y_N - y_M$, x_{N-M} 表示 $x_N - x_M$.

遍历所有 n_{MN} 个点, 找到距离最大的点所在的位置即为分段点 C 的位置, 如图 6 所示. 如果 M 和 N 点之间的泛化轨迹是一条直线, 则直接取中点作为分段点 C 的位置.

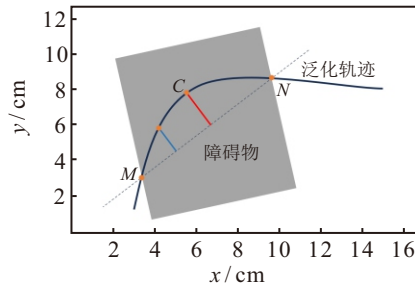


图6 分段点的选择

得到 C 点的位置之后, 需要根据情况调整避障临界点 A 和 B 的位置, 确保算法 1 能够顺利找到融合点. 假设原泛化轨迹 y_d 在 A 点处的垂线方程为:

$$y = k_1(x - x_A) + y_A \quad (11)$$

其中 $k = \dot{y}_d(x_A, y_A)$, 表示轨迹在该点的斜率. 此时可以计算出搜索步长 S_{step} 在 x 和 y 方向上的映射长度:

$$c_{Ax} = S_{step} \cdot \frac{1}{\sqrt{1 + k^2}} \quad (12)$$

$$c_{Ay} = S_{step} \cdot \frac{k}{\sqrt{1 + k^2}} \quad (13)$$

同理可以得到 c_{Bx} 和 c_{By} . 假设初始避障临界点为 $A_0(x_{A_0}, y_{A_0})$ 和 $B_0(x_{B_0}, y_{B_0})$, 最大迭代次数为 M . 确定避障临界点位置算法的伪代码如下:

algorithm 2.

input: $A_0(x_{A_0}, y_{A_0}), B_0(x_{B_0}, y_{B_0}), S_{step}, M$

output: $A(x_A, y_A), B(x_B, y_B)$

initialize $A = (x_{A_0}, y_{A_0}), B = (x_{B_0}, y_{B_0})$

1: for $i \leftarrow 1$ to M do;

2: {algorithm 1 $\leftarrow A$ and B } $\rightarrow D$;

3: if $D = \emptyset$:

4: $x_A \leftarrow x_A - c_{Ax}, y_A \leftarrow y_A - c_{Ay}$

5: $x_B \leftarrow x_B + c_{Bx}, y_B \leftarrow y_B + c_{By}$

6: else : output A and B

7: endif

8: endfor

得到融合点和避障临界点后, 使用 DMP 模型分别进行泛化并将两段泛化轨迹结合在一起, 就可以得到与原泛化轨迹形状相似的避障轨迹, 然而由于 A、B、D 三点处的左右两侧轨迹是单独规划的, 在此处容易出现拐角, 导致轨迹平滑性不高, 因此需要对这些位置进行平滑处理.

3 基于虚拟目标点的轨迹实时平滑算法

3.1 虚拟目标点的设计

轨迹在各分段点和避障临界点处存在拐点导致轨迹不平滑, 常规方案是使用贝塞尔曲线在这些拐点处进行平滑处理, 然而由于这种平滑算法没有考虑障碍物的存在, 生成的轨迹可能会与障碍物有交集, 如图 7 所示的情况.

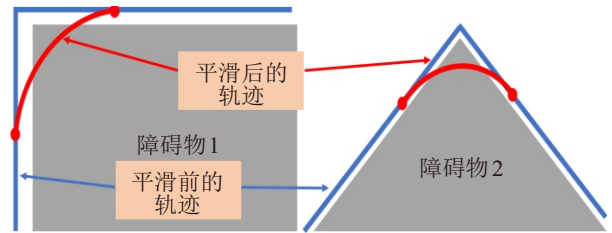


图7 常规平滑算法可能出现碰撞情况

为此, 借鉴人工势场法的思想, 本文提出基于虚拟目标点引导的轨迹实时平滑算法, 通过设置一个吸引子来吸引从起点出发的机器人沿着避障轨迹运动, 并在障碍物附近建立排斥力以避开障碍物. 假设由分段避障算法得到的轨迹为 $y_o = y_1, y_2, \dots, y_N$, 位于机器人前方的轨迹点为 y_k 和 y_{k+1} , 位于机器人后方的轨迹点为 y_{k-1} 和 y_{k-2} , 分别对机器人有引力和推力的作用, 如图 8(a) 所示. 假设机器人当前位置为 q , 则这些轨迹点对机器人产生的力可以通过如下公式计算:

$$F_{att} = k_1(y_k - q) + k_2(y_{k+1} - q) - k_3(y_{k-1} - q) - k_4(y_{k-2} - q) \quad (14)$$

其中 k_1 和 k_2 表示引力系数, k_3 和 k_4 表示推力系数. y_k 定义为主吸引子, 这些轨迹点与机器人的距离越远, 产生的力也越大. 此外, 这些点并非静止的, 而是从 0 时刻开始沿着 y_o 中的轨迹点向前更新, 如图 8(b) 所示, 因此机器人在吸引子的引力的作用下也开始向前运动.

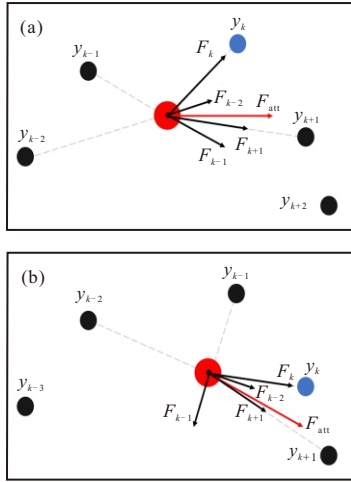


图8 轨迹点对机器人的作用力分析

然而如果只是单纯地通过吸引子的引力引导来确定轨迹的走向,那么生成的轨迹和原始轨迹在没有障碍物的情况下可能是完全重合的,达不到平滑的效果.因此定义一个最大跟踪误差参数 E_{max} 来使吸引子与机器人之间的距离保持在一定范围内,并始终位于机器人前方.当经过转角时,吸引子会比机器人提前通过转角,由于前方有两个轨迹点提供引力,相当于将轨迹转角的角度分摊到不同时刻的机器人运动角度中,因此图8中的两个红色箭头角度差会远小于轨迹转角的度数,从而实现平滑过渡.当吸引子与机器人之间的距离太大时,吸引子会停止向前延伸,直到它们的距离小于最大跟踪误差,起到一个等待作用.此外,定义一个最大步长参数 s_{max} 来限制机器人每次的前进步长,有利于轨迹的平滑过渡.当公式(14)计算出来的进给量太大时,根据最大步长的限制对每个分量进行等比缩小.

3.2 障碍物的斥力分析

当拐角处出现障碍物时,如图9所示的情况,如果没有对障碍物进行处理,将会发生碰撞.对于这种不规则的障碍物,采用文献^[17]中提到的处理方法,对每个角点使用圆形来覆盖它.然而这样处理的话,这个圆可能会挡住之前的避障轨迹,成为一个障碍物,需要对其进行处理.当吸引子进入障碍物区域时,它严格遵循原轨迹的走向,不会受到障碍物的影响.参考人工势场法中对障碍物的处理,此时机器人将受到障碍物的排斥力,导致其运动方向发生变化以避开障碍物.假设障碍物位置为 q_{obs} ,影响范围为 ρ_0 .障碍物的排斥力通过如下公式进行计算:

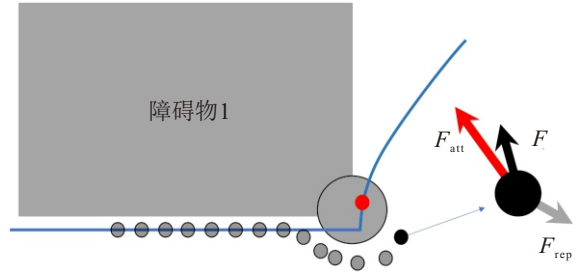


图9 拐点处的避障情况受力分析

$$F_{rep} = \begin{cases} k_{rep} \left(\frac{1}{|\rho_q|} - \frac{1}{\rho_0} \right) \frac{\nabla \rho_q}{\rho_q^2} & \text{if } \rho_q \leq \rho_0, \\ 0 & \text{if } \rho_q > \rho_0. \end{cases} \quad (15)$$

其中, k_{rep} 表示斥力系数, ρ_q 表示方向从障碍物指向机器人的矢量,大小为 $|q - q_{obs}|$. $\nabla \rho_q$ 表示障碍物指向机器人的单位向量.

在两个力的共同作用下,机器人将沿着合理且安全的路径运动,并且平滑地绕过障碍物,如图9所示.具体的轨迹实时平滑和避障效果已录制视频并上传于优酷平台,网址为:<https://v.youku.com/video?vid=XNjQ2OTQ0MDEyMA>

4 实验分析与讨论

4.1 定量评价指标

采用三点曲率来计算轨迹中每个点的曲率,以定量评估轨迹的平滑性,并使用跟踪误差和平均斜率误差作为衡量轨迹特征保留情况的定量指标.其中曲率采用文献^[6]中的公式计算得到.对于跟踪误差,本文仅考虑轨迹单向延伸的情况,如图10所示.

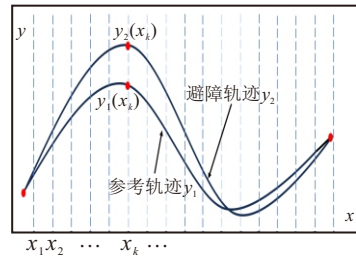


图10 轨迹跟踪误差计算方法

假设参考轨迹为 y_1 ,算法生成的避障轨迹为 y_2 .等间隔地在 x 轴上选择 N 个采样点,计算两条轨迹在该点处的 y 的差值,作为该点的位置跟踪误差.定义误差平均值和最大值为:

$$E_{max} = \max(\{|y_1(x_k) - y_2(x_k)|\}_{k=1}^N), \quad (16)$$

$$E_{mean} = \frac{\sum_{k=1}^N |y_1(x_k) - y_2(x_k)|}{N}. \quad (17)$$

而平均斜率误差是指两条轨迹的对应点之间斜率差值的绝对值的平均值,用于衡量两条轨迹之间

的形状相似性, 即优化后的轨迹的特征保留程度. 平均斜率误差可通过如下公式计算:

$$k_{mean} = \frac{\sum_{k=1}^N \left| \frac{dy_1(x_k)}{dx} - \frac{dy_2(x_k)}{dx} \right|}{N}. \quad (18)$$

4.2 仿真实验

为验证本文方法在跟踪误差, 轨迹相似性和平滑性上的优势, 在仿真环境下设计二维轨迹的避障实验. 在此之前, 许多学者针对 DMP 的避障问题提出了许多改进方法, 例如 Ginesi 等人^[8]在避障耦合项中融入了动态势场函数, 使得轨迹能够更好地保留原轨迹的形状和特征; Zhai 等人^[7]对传统避障耦合项进行改进, 融入了距离因子和动态逼近方法, 使得避障轨迹尽可能地与原轨迹重合. 最近, Jia 等人^[21]进一步改进了 DMP 的公式, 并结合耦合项来实现机械臂的避障, 该方法在轨迹的平滑性和避障成功率上表现良好.

本文将使用这三种方法作为对比方法进行实验. 其中文献 [8] 提出的方法命名为 DMP-DPF(DMP with Dynamic Potential Functions), 文献 [7] 提出的方法命名为 DMP-DAM(DMP with Dynamic Approximation Method), 文献 [21] 提出的方法命名为 iDMP-CT(improved DMP with Coupling Terms). 代码实现上, DMP-DAM 和 iDMP-CT 均在 DMP 的基础上加入原文中提出的耦合项公式进行优化, 而 DMP-DPF 则采用作者提供的开源代码实现. 仿真平台为 Pycharm Community 2022.2, python 版本为 3.6. 实验关键参数设置如下: DMP 系数分别为 $\alpha_y = [25, 25]$, $\beta_y = [6.25, 6.25]$, DMP 基函数个数为 20, $s_{step} = c_{step} = 0.1$, $E_{max} = 0.8$, $s_{max} = 0.12$, $k_1 = 1$, $k_2 = 0.5$, $k_3 = k_4 = 0.1$, $k_{rep} = 0.8$.

在仿真环境下定义一条运动轨迹, 并在轨迹中放置两个静态障碍物, 如图 11 所示.

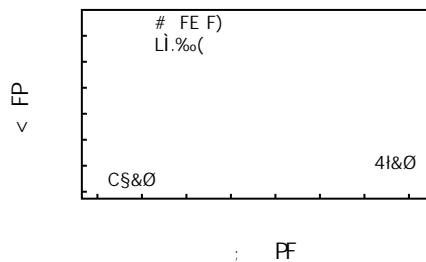
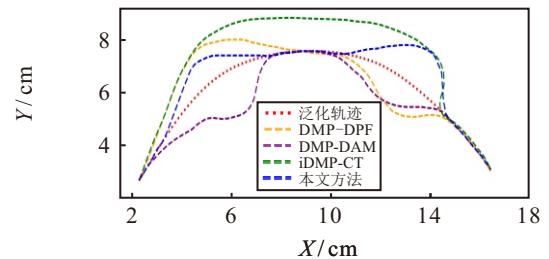


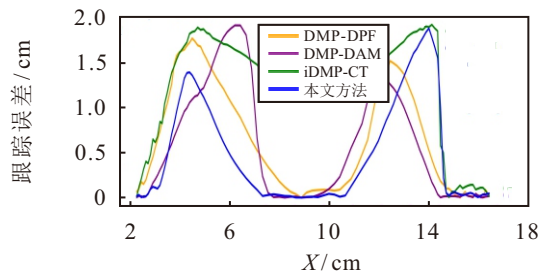
图11 仿真场景 1 的轨迹设置

分别使用本文方法和三种对比方法进行避障轨迹的生成. 下图 12 展示了四种方法生成的避障轨迹, 以及对应的轨迹跟踪误差和曲率值. 对于 DMP-

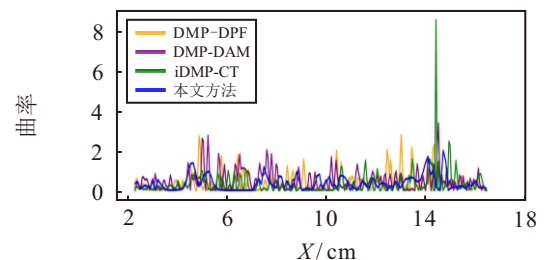
DPF 算法, 由于引入势场, 轨迹的平滑性容易得到保证, 但该方法为了保证轨迹的平滑, 在跟踪误差和特征保留上作了牺牲. 如黄色虚线所示, 轨迹刚开始就受第一个障碍物影响而发生了偏移, 且在绕过第二个障碍物时轨迹往下凹, 而原轨迹在此处的趋势是往上凸的, 这对原泛化轨迹的特征产生了破坏; 对于 DMP-DAM 算法, 由于引入了动态逼近算法, 轨迹在绕过障碍物后能够快速回到原泛化轨迹上, 然而该算法倾向于以最短路径, 最快时间靠近障碍物, 因此生成的轨迹无法保留原轨迹特征, 如绿色虚线所示; 对于 iDMP-CT 算法, 其原理是通过公式中的耦合项来将轨迹整体往远离障碍物的方向牵引, 轨迹倾向于一次性绕过两个障碍物, 随后快速回到原轨迹上, 并保持后段轨迹的趋势向终点运动. 该方法的缺点是轨迹跟踪误差大, 且绕过最后一个障碍物后容易产生轨迹方向突变, 造成轨迹不平滑, 如图 12 顶上的绿色虚线所示; 而本文方法由于对每个障碍物都是单独用分段动态运动基元去拟合原轨迹的特征, 且在拟合过程中确保了轨迹与障碍物没有交集, 因此可以在保留特征情况下避开障碍物. 此外, 本文还采用基于虚拟目标点引导的轨迹实时平滑算法来处



(a) 不同方法的避障轨迹



(b) 位置跟踪误差曲线



(c) 轨迹曲率大小

图12 仿真实验 1 的轨迹跟踪误差和曲率值

理轨迹的过渡点,保障了轨迹的平滑性.图中蓝色虚线展示了本文方法得到的避障轨迹,可以看出,在非障碍物影响区域,轨迹能够很好地对原泛化轨迹进行跟踪,而在遇到障碍物时能平滑地绕过障碍物,且保留原轨迹的特征.

表1详细展示了四种方法生成轨迹的定量指标,可以看出本文方法除了最大跟踪误差之外,其他指标都是最优的.之所以最大跟踪误差不及DMP-DPF和DMP-DAM,是为保留轨迹特征而做的牺牲,因为如果第二个障碍物的融合点设在下方,那么后端轨迹的泛化会失败,这不利于特征保留.

表1 不同方法下仿真轨迹1的各项性能指标对比

方法	E_{max}	E_{mean}	C_{max}	C_{mean}	k_{mean}
DMP-DPF	1.7674	0.6827	2.7957	0.3935	0.1325
DMP-DAM	1.9081	0.5647	3.3935	0.4965	0.1758
iDMP-CT	1.9153	1.1636	8.6175	0.3763	0.0862
本文方法	1.8748	0.5456	2.0127	0.3548	0.0845

为验证更加复杂的场景下本文方法的有效性,参考文献[6]中的轨迹设置,设计一条包含3个演示特征的轨迹,并在每个特征处放置一个障碍物来阻挡轨迹,如下图13所示.

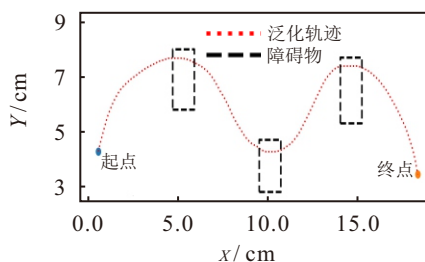


图13 仿真场景2的轨迹设置

同样采用四种方法分别对轨迹进行避障处理,下图14展示了四种方法生成的避障轨迹,以及对应的轨迹跟踪误差和曲率值.此外,表2详细展示了四种方法生成轨迹的定量指标,通过对比容易看出,本文方法得到的各项指标都是最优的.

4.3 六轴机器人轨迹避障实验

为验证该方法在实际机器人上的有效性,设计了一个六轴机器人末端轨迹的避障实验.实验场景如图15所示,采用自主研发的总线式六轴机器人控制系统控制机器人运动,上位机使用EtherCAT总线 and 伺服驱动器进行通信,控制频率为500 Hz,具有基于位置控制的直接示教模式和基于力矩控制的动觉示教模式,可在控制软件中自由增加算法.采用Intel RealSense D435工业相机来获取障碍物的位置和尺寸信息,并通过TCP协议与上位机通信.

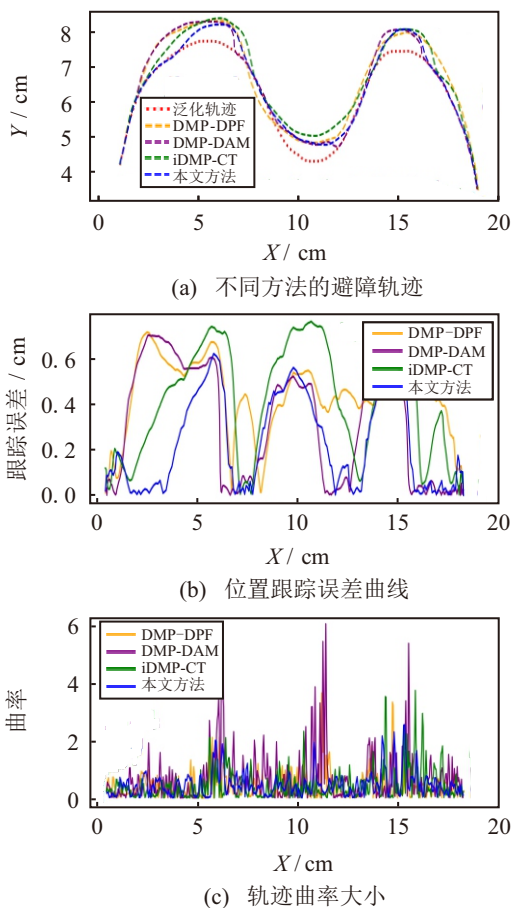


图14 仿真实验2的轨迹跟踪误差和曲率值

表2 不同方法下仿真轨迹2的各项性能指标对比

方法	E_{max}	E_{mean}	C_{max}	C_{mean}	k_{mean}
DMP-DPF	0.7108	0.4322	3.5857	0.4007	0.0498
DMP-DAM	0.6980	0.3175	5.9660	0.6468	0.0541
iDMP-CT	0.7577	0.4093	3.6744	0.4237	0.0475
本文方法	0.6243	0.1824	2.4953	0.3973	0.0436



图15 实验场景设置

在机器人前方放置物料,用于标记特征的位置,随后操作员通过动觉示教的方式拖动机器人末端,生成机器人的模拟作业轨迹(如图16(a)所示).分别设计包含1个特征(图16(b))和2个特征(图16(c))的运动轨迹,随后操作员大致沿着设计的轨迹拖动出机器人.在这个过程中,控制器每隔50 ms记录一次机器人末端位置,最终得到一系列点的集合.使用DMP对该轨迹进行建模,然后在轨迹的特征处分别放入障碍物来阻挡原轨迹,模拟机器人在新的场景中执行该任务时遇到障碍物的干扰的情况.

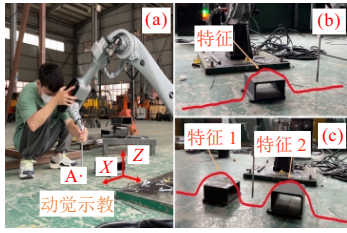


图16 轨迹的设置和采集

图中 A 点表示机器人的末端位置, 将 A 点作为参考点来记录机器人运行轨迹. 实验关键参数设置如下: DMP 系数分别为 $\alpha_y = [25, 25, 25]$, $\beta_y = [6.25, 6.25, 6.25]$, DMP 基函数个数为 100, $s_{step} =$

$c_{step} = 1$, $E_{max} = 2$, $s_{max} = 0.24$, $k_1 = 1$, $k_2 = 0.5$, $k_3 = k_4 = 0.1$, $k_{rep} = 0.8$. 随后分别使用上节提到的 4 种方法进行该场景下的避障轨迹生成. 机器人开始运行后每隔 50ms 记录 A 点的位置坐标, 由于轨迹在 y 方向上的变化很小, 为了更直观地对比, 将轨迹映射到 xoz 平面上, 对轨迹的各项参数进行统计. 下图 17 和 18 分别展示了两种轨迹设置下不同方法的避障轨迹, 以及对应的跟踪误差和曲率值. 此外, 下表 3 和 4 详细展示了四种方法生成轨迹的最大跟踪误差、平均跟踪误差、最大曲率值、平均

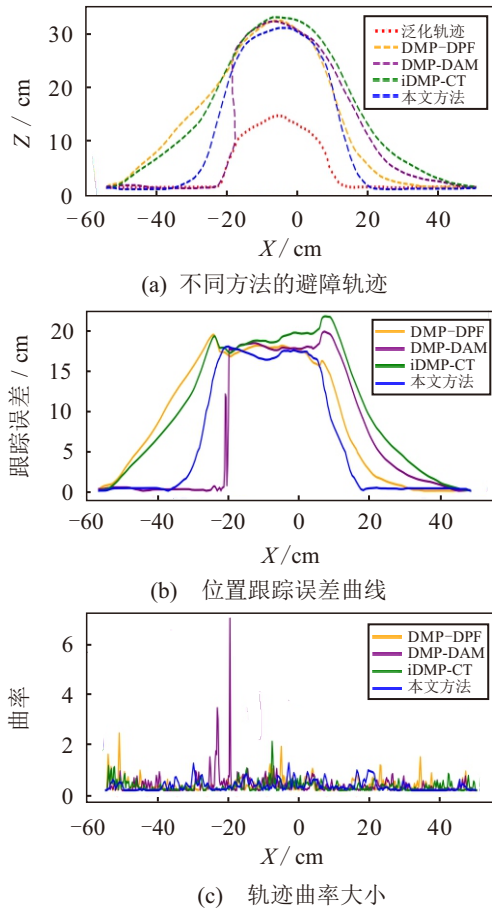


图17 单特征轨迹的跟踪误差和曲率值

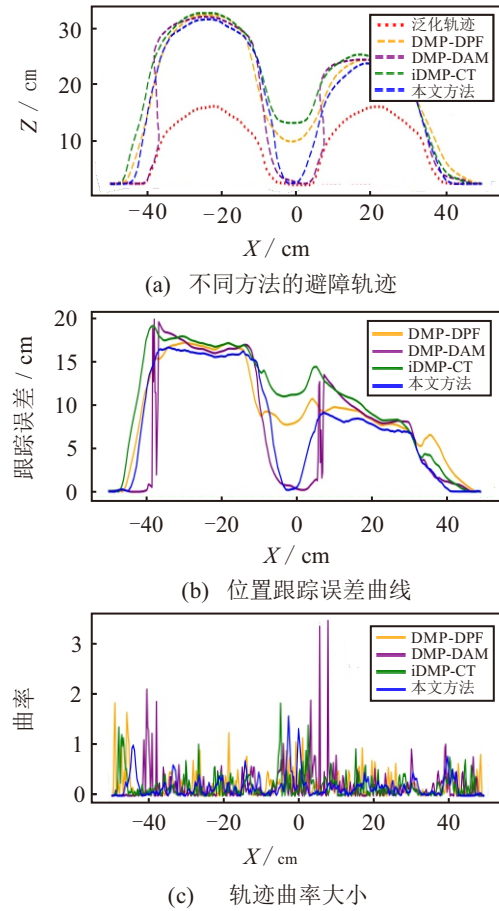


图18 双特征轨迹的跟踪误差和曲率值

表3 不同方法下单特征轨迹的各项性能指标对比

方法	E_{max}	E_{mean}	C_{max}	C_{mean}	k_{mean}
DMP-DPF	19.3184	8.9537	2.2747	0.1909	0.1175
DMP-DAM	19.6915	7.7731	6.8562	0.2020	0.1749
iDMP-CT	21.5838	10.5699	1.9424	0.1978	0.1126
本文方法	17.8614	7.7192	1.0874	0.1806	0.1085

表4 不同方法下双特征轨迹的各项性能指标对比

方法	E_{max}	E_{mean}	C_{max}	C_{mean}	k_{mean}
DMP-DPF	17.1142	9.3029	1.8133	0.1964	0.1848
DMP-DAM	19.7837	8.6865	3.4300	0.1898	0.2413
iDMP-CT	19.0315	10.5629	1.8099	0.1872	0.1803
本文方法	16.5578	8.6118	1.5563	0.1711	0.1754

曲率值以及平均斜率误差. 可以看出, 本文方法在各项指标上均是最优的, 从而验证了本文方法在轨迹特征保留, 轨迹跟踪精度和轨迹平滑性上的综合优势.

本实验的具体实现过程已制作成视频, 并上传于优酷平台, 网址为: <https://v.youku.com/video?vid=XNjQ3Njc3MjEwOA>

5 结论

本文提出了一种基于分段 DMP 的轨迹平滑避障算法, 用于研究机器人作业轨迹泛化到新场景过程中的特征保留和平滑避障问题. 首先, 该方法使用 DMP 来对轨迹进行建模, 以学习原轨迹的特征并实现泛化功能; 其次, 在泛化轨迹被障碍物阻挡时, 使用分段 DMP 来拟合避障轨迹, 以最大程度地保留原泛化轨迹特征; 再次, 本文提出了一种基于虚拟目标点的轨迹实时平滑算法, 能够在跟踪避障轨迹的同时, 对过渡区域的拐角进行平滑处理, 有效地保障了避障轨迹的平滑性; 最后, 在实验阶段, 设计了仿真实验和六轴机器人的轨迹避障实验, 与 3 种现有的基于 DMP 的优化算法进行对比. 实验结果表明, 本文方法不仅能够在障碍物区域生成高相似性的避障轨迹, 而且轨迹的跟踪误差和平滑性指标都更加优秀, 从而验证了本文方法的有效性.

参考文献 (References)

- [1] Li J Z, Cong M, Liu D, et al. Enhanced task parameterized dynamic movement primitives by GMM to solve manipulation tasks[J]. *Robotic Intelligence and Automation*, 2023, 43(2): 85-95.
- [2] Wu X M, He M, Liu T, et al. Study of robot demonstration learning based on the Dirichlet process clustering[J]. *Chinese Journal of Scientific Instrument*, 2023, 44(1): 265-274.
- [3] Huang Y L, Xu D, Tan M. On imitation learning of robot movement trajectories: A survey[J]. *Acta Automatica Sinica*, 2022, 48(2): 315-334.
- [4] Wang Y Q, Hu Y D, El Zaatari S, et al. Optimised learning from demonstrations for collaborative robots[J]. *Robotics and Computer-Integrated Manufacturing*, 2021, 71: 102169.
- [5] Kim B S, Lee M C. A novel approach to coupling terms to avoid obstacles in a manipulator movement reproduction[J]. *International Journal of Control, Automation and Systems*, 2024, 22(4): 1289-1300.
- [6] Hu Y D, Wang Y Q, Hu K X, et al. Adaptive obstacle avoidance in path planning of collaborative robots for dynamic manufacturing[J]. *Journal of Intelligent Manufacturing*, 2023, 34(2): 789-807.
- [7] Zhai D H, Xia Z Q, Wu H C, et al. A motion planning method for robots based on DMPs and modified obstacle-avoiding algorithm[J]. *IEEE Transactions on Automation Science and Engineering*, 2022, 20(4): 2678-2688.
- [8] Ginesi M, Meli D, Roberti A, et al. Dynamic movement primitives: Volumetric obstacle avoidance using dynamic potential functions[J]. *Journal of Intelligent & Robotic Systems*, 2021, 101(4): 79.
- [9] Liu T, Zhang X Y, Lin C Y, et al. Robotic arm trajectory learning and obstacle avoidance method based on segmented dynamic movement primitive[J]. *Robot*, 2024, 46(3): 275-283.
- [10] Zhang S S, Huang H S, Huang D, et al. Subtask-learning based for robot self-assembly in flexible collaborative assembly in manufacturing[J]. *The International Journal of Advanced Manufacturing Technology*, 2022, 120(9): 6807-6819.
- [11] El Zaatari S, Wang Y Q, Li W D, et al. iTP-LfD: Improved task parametrised learning from demonstration for adaptive path generation of cobot[J]. *Robotics and Computer-Integrated Manufacturing*, 2021, 69: 102109.
- [12] Qian K, Xu X, Liu H, et al. Environment-adaptive learning from demonstration for proactive assistance in human-robot collaborative tasks[J]. *Robotics and Autonomous Systems*, 2022, 151: 104046.
- [13] Li W, Zhang Y C, Ning J, et al. Collision avoidance of under-actuated unmanned surface vehicles formation with improved artificial potential field method[J]. *Control and Decision*, 2025, 40(1): 252-260.
- [14] Liu L, Li B, Fang Y M, et al. AGTB-RRT*: A target bidirectional 3D RRT* path planning algorithm based on approximate geodesic obstacle avoidance strategy[J]. *Control and Decision*, 2024, 39(11): 3763-3771.
- [15] Chen D, Tan Q, Xu Z Z. Robotic arm path planning based on sampling point optimization RRT algorithm[J]. *Control and Decision*, 2024, 39(8): 2597-2604.
- [16] Li P Y, Hao L J, Zhao Y J, et al. Robot obstacle avoidance optimization by A* and DWA fusion algorithm[J]. *PLoS One*, 2024, 19(4): e0302026.
- [17] Su Y B, Lin C Y, Liu T. Real-time trajectory smoothing and obstacle avoidance: A method based on virtual force guidance[J]. *Sensors*, 2024, 24(12): 3935.
- [18] Saveriano M, Abu-Dakka F J, Kramberger A, et al. Dynamic movement primitives in robotics: A tutorial survey[J]. *The International Journal of Robotics Research*, 2023, 42(13): 1133-1184.
- [19] Dong J L, Si W Y, Wang N, et al. Robot learning from multiple demonstrations based on generalized Gaussian mixture model[C]. 2023 28th International Conference on Automation and Computing. Birmingham, 2023: 1-6.
- [20] Rai A, Meier F, Ijspeert A, et al. Learning coupling terms for obstacle avoidance[C]. 2014 IEEE-RAS International Conference on Humanoid Robots. Madrid, 2014: 512-518.
- [21] Jia X H, Zhao B, Liu J Y, et al. A trajectory planning method for robotic arms based on improved dynamic motion primitives[J]. *Industrial Robot: the International Journal of Robotics Research and Application*, 2024, 51(5): 847-856.
- [22] Ye B, Yu P, Hu C, et al. GRF-GMM: A trajectory optimization framework for obstacle avoidance in learning from demonstration[M]. *Neural Information Processing*. Singapore: Springer Nature Singapore, 2023: 18-30.

作者简介

苏永彬 (1996-), 男, 博士研究生, 主要研究方向为工业机器人控制, E-mail: suyongbin@stu.xmu.edu.cn;

刘瞰东 (1970-), 男, 教授, 博士, 博士生导师, 主要研究方向为工业机器人、机器视觉, E-mail: ltd@xmu.edu.cn.