

基于动态运动基元的移动机器人路径规划^①

姜明浩^② 陈 洋^③ 李威凌

(武汉科技大学信息科学与工程学院 武汉 430081)

摘 要 针对二维环境下移动机器人路径规划问题,提出一种基于动态运动基元(DMPs)的路径规划方法。该方法首先用手柄来控制机器人,记录机器人的运动轨迹,然后将记录的轨迹作为 DMPs 算法的学习样本,利用轨迹样本进行训练获得 DMPs 算法的模型参数,从而实现机器人的路径规划。当运动环境中存在障碍物时,在已有学习基础上通过设计耦合因子规划出避障路径,在此基础上,改变机器人运动的目标位置,可以完成对新目标的泛化推广。仿真和实验结果表明,DMPs 算法在移动机器人路径规划上具有可行性。

关键词 动态运动基元(DMPs), 移动机器人, 二维环境, 路径规划, 避障

0 引 言

机器人的路径规划是指在有障碍物的环境中,规划出一条从起点到终点的合适的路径,这条路径可以避开所有的障碍物到达终点,并且在经过某些指定的点时要完成给定的任务,最后在完成任务的前提下尽量优化机器人的运行轨迹^[1]。传统的路径规划方法包括可视图法^[2]、自由空间法^[3]、栅格法^[4]、人工势场法^[5]等。文献[6]提出一种可视图法与人工免疫算法相结合的方法,可视图法用来得到机器人运动的环境信息,人工免疫算法用来求解最优路径,两者相结合应用到机器人路径规划上。文献[7]提出了一种改进的势场栅格算法。文献[8]在蚁群算法搜索过程中加入与全局信息素相关的人工势场局部搜索寻优算法,利用蚁群的全局性避开人工势场局部最优,使两者完善结合。然而以上路径规划算法均在环境建模的基础上添加诸多的约束条件通过各种算法搜索机器人运动路径。搜索到的

路径取决于环境建模的准确性和算法的有效性,因此本文希望从学习的角度进行机器人路径规划,即可以不用在传统的算法上加约束条件实现机器人的路径规划。本文针对二维环境下移动机器人路径规划问题,提出了一种基于动态运动基元(dynamic movement primitives, DMPs)^[9-13]的路径规划算法,该算法的主要思想是从演示中学习。目前 DMPs 算法在机械臂和四旋翼无人机上已经有了很好的应用^[14,15],但国内外对其在移动机器人路径规划上的研究较少。本文将动态运动基元算法应用在移动机器人路径规划问题中,从运动学习的角度将移动机器人一次具体运动看作学习样本,提取其运动基元。对于新的目标点,能够利用学习获得的运动基元泛化出相应的行走路径。在有障碍物的情况下,机器人也能够通过 DMPs 规划出一条可行的避障路径。这种运动学习机制可以有效地实现移动机器人的路径规划。最后,在 MATLAB 仿真环境和基于 ROS 的移动机器人平台上分别验证了该算法解决移动机器人路径规划问题的可行性。

① 国家自然科学基金(61573263,61203331),湖北省科技支撑(2015BAA018)和武汉科技大学科技创新基金(14ZRC147)资助项目。

② 男,1992年生,硕士生;研究方向:机器人路径规划,自主学习;E-mail: jiangmh92@163.com

③ 通讯作者,E-mail: chenya@wust.edu.cn

(收稿日期:2016-08-26)

1 动态系统的运动生成

这一部分主要从动态运动基元框架的组成、基于 DMPs 的多自由度运动的学习和 DMPs 算法的避障方法三个方面进行讨论。

1.1 动态运动基元

动态运动基元一般用来形成离散或有规律的运动,本文主要讨论离散运动。一个一维运动可由下面微分方程表示,也可以表示成一个带有外力扰动的线性弹簧系统^[16]:

$$\tau \ddot{v} = K(g - x) - Dv - K(g - x_0)s + Kf(s) \quad (1)$$

$$\tau \dot{x} = v \quad (2)$$

其中:

- x 和 v 分别是系统的位移和速度;
 - x_0 和 g 分别是初始位置和目标位置;
 - τ 时间伸缩因子;
 - K 是弹簧的弹性系数;
 - D 是系统处于临界状态下的阻尼系数;
 - f 是非线性函数,用于生成任意复杂的运动。
- 方程(1)是转换系统,其中非线性函数 f 定义为

$$f(s) = \frac{\sum_i w_i \psi_i(s)}{\sum_i \psi_i(s)} \quad (3)$$

$\psi_i(s)$ 是基本高斯函数,定义为

$$\psi_i(s) = \exp(-h_i(s - c_i)^2) \quad (4)$$

其中 c_i 为中心, h_i 为高度, w_i 为权重。

由式(3)可知,函数 f 并不直接取决于时间参数,而是取决于相位变量 s , s 的表达形式为

$$\tau \dot{s} = -\alpha s \quad (5)$$

其中 α 为任意常数。由方程(5)可知 s 是由 1 到 0 单调递减的,因此方程(5)称为正则系统。

本文通过手柄遥控机器人运动,并记录其走过路径的位移 x 、速度 v 与加速度 $\dot{v}$,因为正则系统是可积分的,即 s 可以根据参数 τ 计算出来,所以式(1)中的 $f_{\text{demo}}(s)$ 可以表示成

$$f_{\text{demo}}(s) = \frac{\tau \dot{v} + Dv}{K} - (g - x) + (g - x_0)s \quad (6)$$

要找出方程(3)中的最佳权重值 w_i ,可以根据最小误差准则 $J = \sum_s (f_{\text{demo}}(s) - f(s))^2$ 来求解,即

当 J 取最小时的 w_i 值就是最佳的权重值。根据以上求解得到的最佳权重值 w_i ,并且针对不同的轨迹设置其相对应的起点 x_0 和终点 g 值,代入到式(1)和式(2)中就可以算出当前物体运动的位移 x 、速度 v 和加速度 $\dot{v}$,即得到通过 DMPs 算法学习之后的轨迹点。

1.2 基于 DMPs 的多自由度运动的学习

DMPs 算法的流程如图 1 所示。为了将多自由度运动学习应用于图 1 所示的系统,将多自由度的运动共享一个正则系统,对应每个自由度有其相应的动力学系统。同一个正则系统保证了每个自由度运动的时间耦合性,而对应的动力学系统保证了每个自由度有其相对应的运动特性,描述如图 2 所示^[17]。

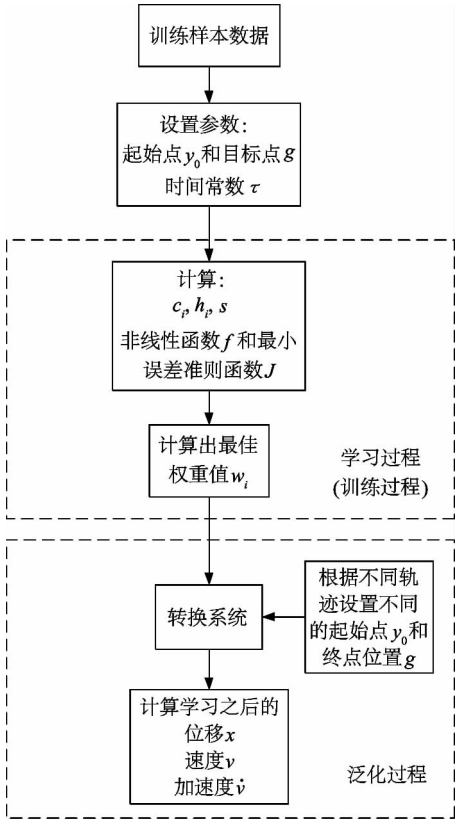


图 1 DMPs 学习算法的结构图

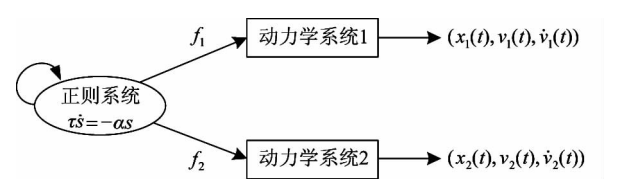


图 2 多自由度运动学习原理图(以两个自由度为例)

由图 1 和图 2 可知,基于对运动的学习可以规划出一条所需要的运动轨迹,完成运动任务。鉴于上述的动态运动基元的运动学习策略特点,将此方法运用到移动机器人的路径规划上。首先将移动机器人的运动分解为两个自由度,分别建立各自自由度的非线性强制项 f 以及相应的动力学系统,如图 2 所示。首先让系统学习一条所需的运动轨迹,再通过图 1 所示的流程,学习和泛化出移动机器人实际所需要的运动轨迹点,本文的实验平台是基于 ROS 的移动机器人,通过设计用户界面 (user interface, UI) 读取机器人实际运动的轨迹点,这样移动机器人在此控制策略和给定的路径轨迹点下能够完成目标任务。

1.3 二维路径避障

在一个二自由度运动系统中,在直角坐标系中建立点到点的运动模型,其中位置向量 $\mathbf{x} = [x_1 \ x_2]^T$ 和对应的速度向量为 $\mathbf{v} = [v_1 \ v_2]^T$ 。运动从起点 $\mathbf{x}_0 = [x_{01} \ x_{02}]^T$ 到目标点 $\mathbf{g} = [g_1 \ g_2]^T$,在从起点到目标点的运动过程中有障碍物,设障碍物圆心为 $\mathbf{o} = [o_1 \ o_2]^T$ 。在式(1)中加入耦合项 C_i 从而构建带避障功能的动力学系统,即下式:

$$\tau \dot{\mathbf{v}} = K(\mathbf{g} - \mathbf{x}) - D\mathbf{v} - K(\mathbf{g} - \mathbf{x}_0)\mathbf{s} + Kf(\mathbf{s}) + C_{i,j} \quad (7)$$

对于两个自由度的运动耦合项向量形式为 $C_i = [c_{i,x} \ c_{i,y}]^T$, $C_{i,j} (j = x, y)$, 分别代表 X 轴方向和 Y 轴方向上的运动耦合因子。 C_i 还表示为

$$C_i = \gamma R \mathbf{v} \theta e^{-\beta \theta} \quad (8)$$

其中 θ 为速度向量 $\mathbf{v}$ 与障碍物中心坐标和当前位置坐标差向量 $(\mathbf{o} - \mathbf{x})$ 的夹角, R 是以 $\mathbf{r} = (\mathbf{o} - \mathbf{x}) \times \mathbf{v}$ 为轴, $\pi/2$ 为旋转角的旋转矩阵^[18], 如式(9)所示:

$$\theta = \arccos\left(\frac{(\mathbf{o} - \mathbf{x})^T \mathbf{v}}{\|\mathbf{o} - \mathbf{x}\| \|\mathbf{v}\|}\right) \quad (9)$$

2 仿真分析

2.1 数据采集

根据 1.1 节可知,当获取到机器人运动时的位移、速度和加速度时,且已知系统对应的最优权重

值,就可以通过 DMPs 算法得到学习之后的运动轨迹点,即完成移动机器人的路径规划。因此采集机器人运动时的数据是必要的。本文用 VS2010 软件搭建了一个微软基础类库 (Microsoft foundation classes, MFC) 界面模拟机器人运动的二维环境,该界面的长和宽都为 1m,在仿真界面上机器人用一个直径为 5mm 的小实心圆来代替,障碍物为各种平面图形 (如矩形、圆形、椭圆形和直线),利用开源计算机视觉库 (open source computer vision library, OPENCV) 来实现手柄对机器人的控制,设定手柄摇杆推动的位移大小为机器人速度的大小,并且设置速度的范围为 $\pm 5\text{mm/s}$,机器人能从起点避免与障碍物碰撞到达目标点,在这个过程中,通过回调函数实现对机器人运动轨迹中的位移、速度和加速度值的采集,其中采集数据的频率为 10Hz。本文搭建了一个 MFC 界面模拟机器人运动环境,通过 OPENCV 实现手柄操控机器人运动,其运动环境如图 3 所示。

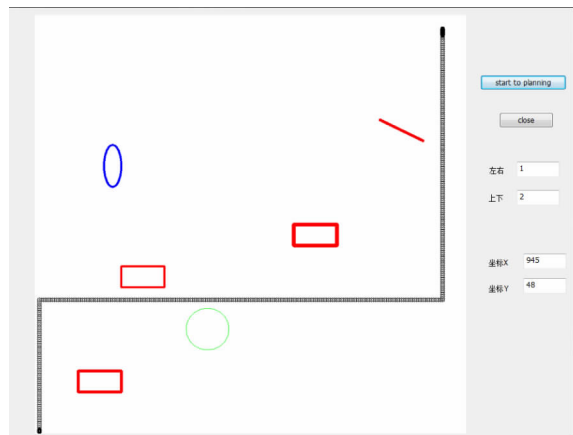


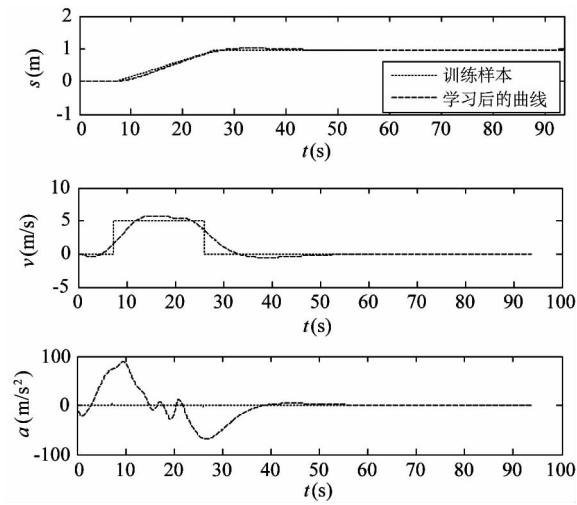
图 3 机器人运动的二维环境

当成功进入如图 3 所示的二维环境里,通过操控手柄控制机器人 (一个点) 运动到目标位置,考虑到机器人一般只能前后和左右运动,因此手柄遥控机器人时只是让机器人前后或者左右运动 (黑色折线为手柄操控机器人运动时的轨迹)。本次仿真采集了 938 个数据,由数据分析可知,在 X 轴方向上,目标位置记为 $X(s) = 0.950\text{m}$,到达目标点所用的时间记为 $X(t) = 26.2\text{s}$;在 Y 轴方向上,目标点位置记为 $Y(s) = 0.972\text{m}$,到达目标点所用的时间记为 $Y(t) = 39\text{s}$ 。

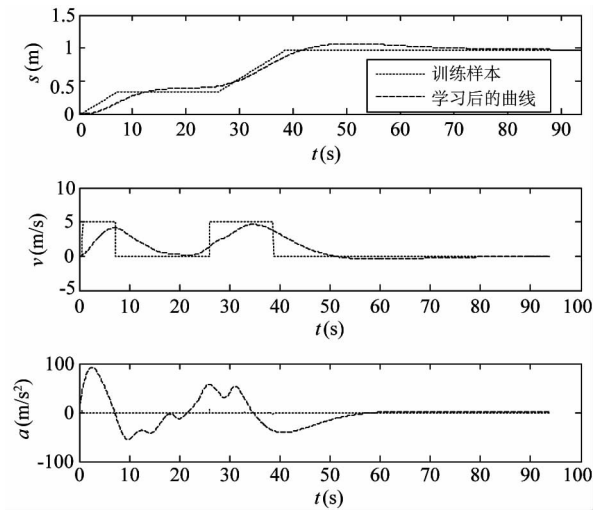
2.2 性能分析

在 2.1 节中采集到的数据,作为 DMPs 算法的学习样本,训练完样本后得到学习之后的运动轨迹,仿真图如图 4 所示。其中图 4(a)为 X 轴方向上的

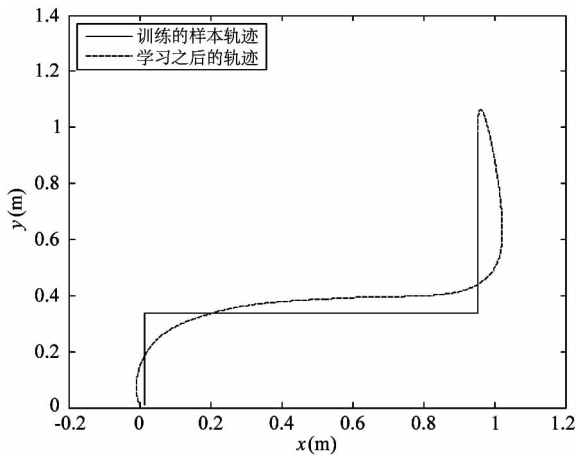
DMPs 学习后的仿真图;图 4(b)为 Y 轴方向上的 DMPs 学习后仿真图;图 4(c)为样本轨迹与学习后的轨迹图;图 4(d)为各自由度的非线性强制项的变化曲线。



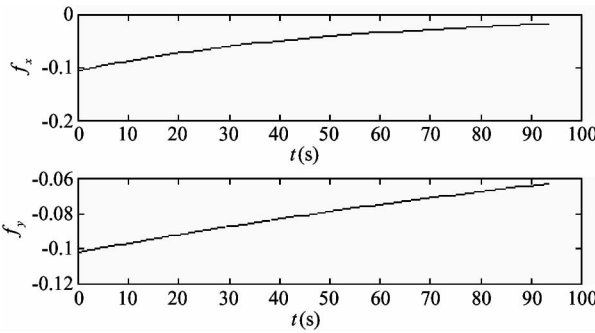
(a) X 轴方向上的仿真图



(b) Y 轴方向上的仿真图



(c) 训练样本的轨迹与学习之后的轨迹对比图



(d) 二自由度的非线性强制项

图 4 通过 DMPs 学习后的机器人轨迹仿真图

由样本数据可知, X 轴方向上到达目标点的时间为 26.2s; Y 轴方向上到达目标点位置的时间为 39.0s。根据图 4 分别记录 $X(t) = 26.2$ s 时的 DMPs 学习后的值和 $Y(t) = 39.0$ s 时的 DMPs 学习后的值,将样本数据与学习之后的数据进行分析,如表 1 所示。表 2 是样本轨迹到达目标位置的时间与通过 DMPs 算法学习后到达目标点位置时间的对比分析。表 3 是各自由度上的最佳权重值,本次仿真采

用的基本高斯函数的个数为 $N = 8$ 。

表 1 样本数据与学习后的数据对比

时间(s)	$X(t) = 26.2$	$Y(t) = 39.0$
样本的值(m)	0.950	0.972
DMP 学习后的值(m)	0.913	0.921
误差百分比(%)	3.863	5.216

表 2 到达目标点所用时间的对比

目标点位置(m)	$X(s) = 0.95$	$Y(s) = 0.972$
样本轨迹所用时间(s)	26.2	39.0
学习所用时间(s)	27.5	42.3

表 3 各自由度上的最佳权重值

权重	X 轴方向	Y 轴方向
w_1	9.41	-18.56
w_2	12.66	-6.67
w_3	12.70	15.64
w_4	6.02	34.88
w_5	-1.42	64.71
w_6	-34.46	3.35
w_7	-138.10	-172.35
w_8	-205.35	-258.68

由表 1 和表 2 可知,通过 DMPs 算法学习后的轨迹到达目标点相对于样本轨迹到达目标点有一些延时,但学习后的轨迹是收敛于目标点位置的。

2.3 DMPs 算法的避障方法

由 1.3 节可知,在 DMPs 算法中增加避障因子 C_i 来实现避开障碍物,设定式(8)中的常数 $\gamma = 8$, $\beta = 60/\pi$ 。设置起点坐标为(0, 0),终点坐标为(1, 1),旋转矩阵 R 表示成:

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

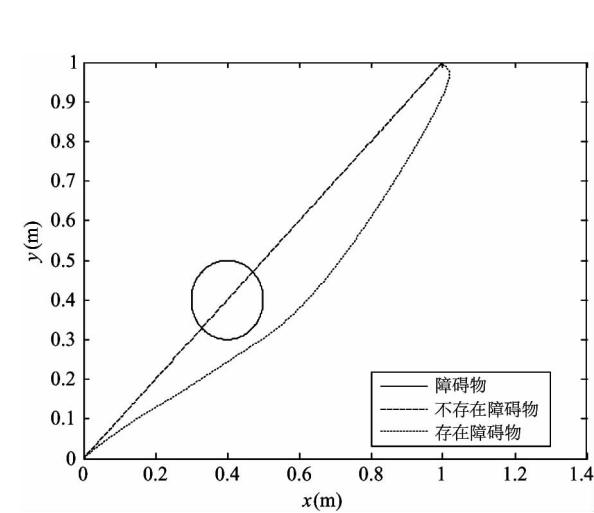
(10)

设定障碍物在直角坐标系表示为圆心坐标为(0.4, 0.4),半径为 0.1 的圆,采用 DMPs 算法避开障碍物的仿真图如图 5(a)所示, X 轴方向和 Y 轴方向上的耦合因子变化曲线如图 5(b)所示。分析图 5(b)可知在 22s 前由于障碍物的阻挡,耦合因子起作用改变机器人运动轨迹,在 22s 左右由于机器人基本避开障碍物耦合因子迅速衰减到 0。

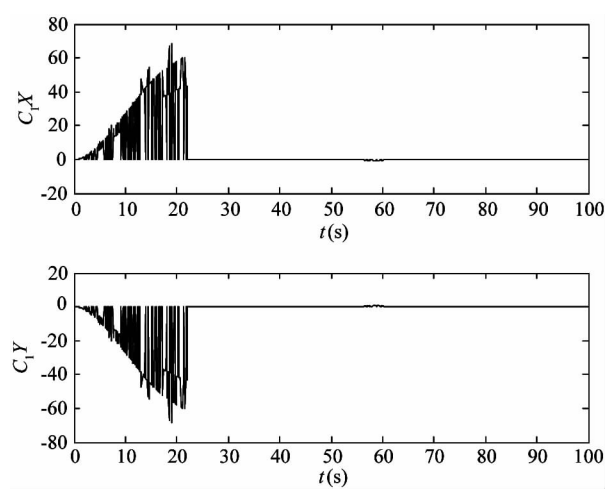
2.4 DMPs 的泛化推广能力

当改变目标点到任意位置时,DMPs 算法能否使机器人自主到达目标点,是决定 DMPs 算法泛化推广性能优劣的重要标准。本文改变目标点在 X 轴方向上位置为0.8m和0.5m,其仿真图如图 6 所示。其中图 6(a)为目标点在 X 轴方向上位置为 0.8m;图 6(b)为目标点在 X 轴方向上的位置为 0.5m;图 6(c)为改变目标点位置时的运动轨迹对

比图。



(a) 不存在障碍物与存在障碍物的轨迹对比图

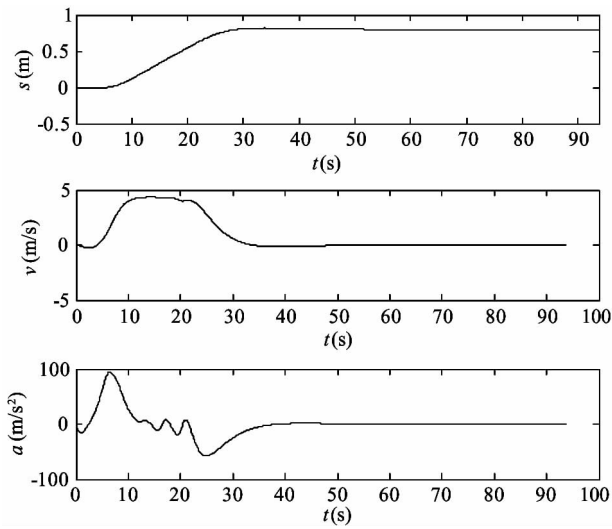


(b) 二自由度耦合因子变化曲线

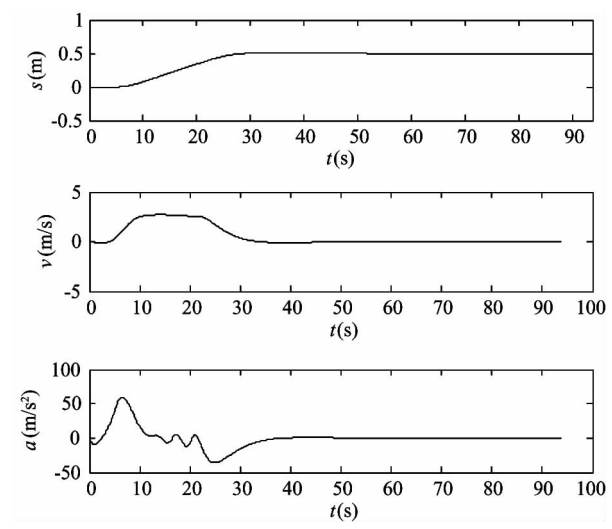
图 5 DMPs 避开障碍物的仿真图

由图 6 可知,当目标点位置改变时,DMPs 算法仍能使机器人到达目标点位置,并且此运动轨迹具有原样本轨迹的运动特性。本文希望了解 DMPs 算法的准确性,因此将目标点位置与通过 DMPs 算法学习后到达的位置进行对比分析,如表 4 所示。

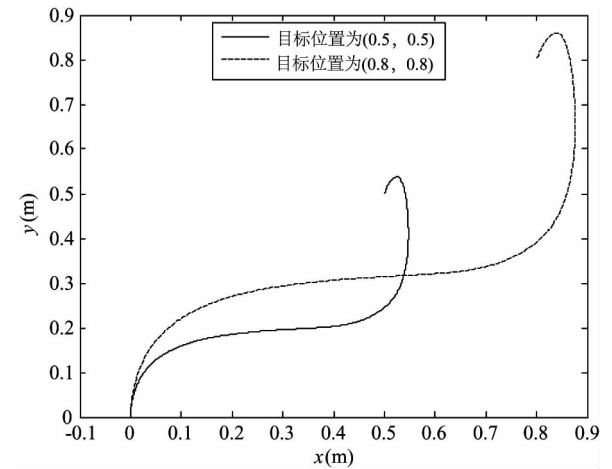
由表 4 可知,当目标点位置改变时,利用 DMPs 算法可以使机器人到达目标位置,而且当时间为样本轨迹到达目标点时的时间($t = 26.2s$)时,其位移误差不超过 10%,由以上分析可知,DMPs 算法具有较好的泛化推广能力。



(a) 目标点在 X 轴上的位置为 0.8m



(b) 目标点在 X 轴上的位置为 0.5m



(c) 改变目标点位置时的轨迹对比图

图 6 改变目标点位置时的轨迹仿真图

表 4 样本数据与学习后的数据对比

目标位置(m)	实际位移(m)	误差百分比(%)
$g = 0.800$	$y = 0.758$	5.25
$g = 0.500$	$y = 0.474$	5.20

3 实验与分析

本文在 JS-R 机器人平台上验证 DMPs 算法的有效性。首先用手柄控制机器人沿着一条可行的路径运动;然后在 Qt 中的 UI 界面上采集机器人运动的位移、速度和加速度,将采集到的数据作为样本并进行训练;将学习后的数据(速度值)通过 UI 界面读入到 JS-R 机器人中,设置机器人运动的起点位置,因此机器人可以按照给定的速度运动。其中 UI 界面如图 7 所示,其中左框用来发布采集到的数据,右框中的 connect 按钮用于给机器人连接用户发布的数据,这样机器人就可以按照用户发布的数据运动;实验结果如图 8 所示,其中图 8(a)描述手柄操控机器人运动的轨迹图,图 8(b)是机器人通过 DMPs 算法自主运动的轨迹图,图 8(c)是机器人运动的平面示意图。其中左图是训练时的示意图;右图是通过 DMPs 算法学习后机器人运动的平面示意图。

从图 8 可知,通过 DMPs 算法训练样本轨迹,设置机器人运动的起始位置和目标位置,机器人可以按照学习的轨迹可以到达指定的目标点位置,在 $t = 28s$ 时,机器人在拐弯处的速度是平滑的; $t = 180s$ 时,机器人的位置相对于样本轨迹中的目标位

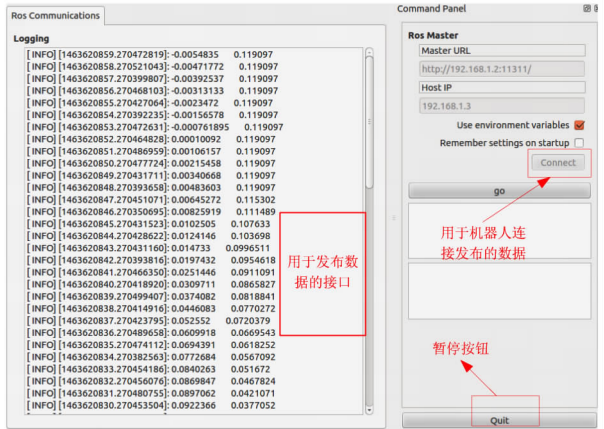


图 7 UI 界面用于发布数据

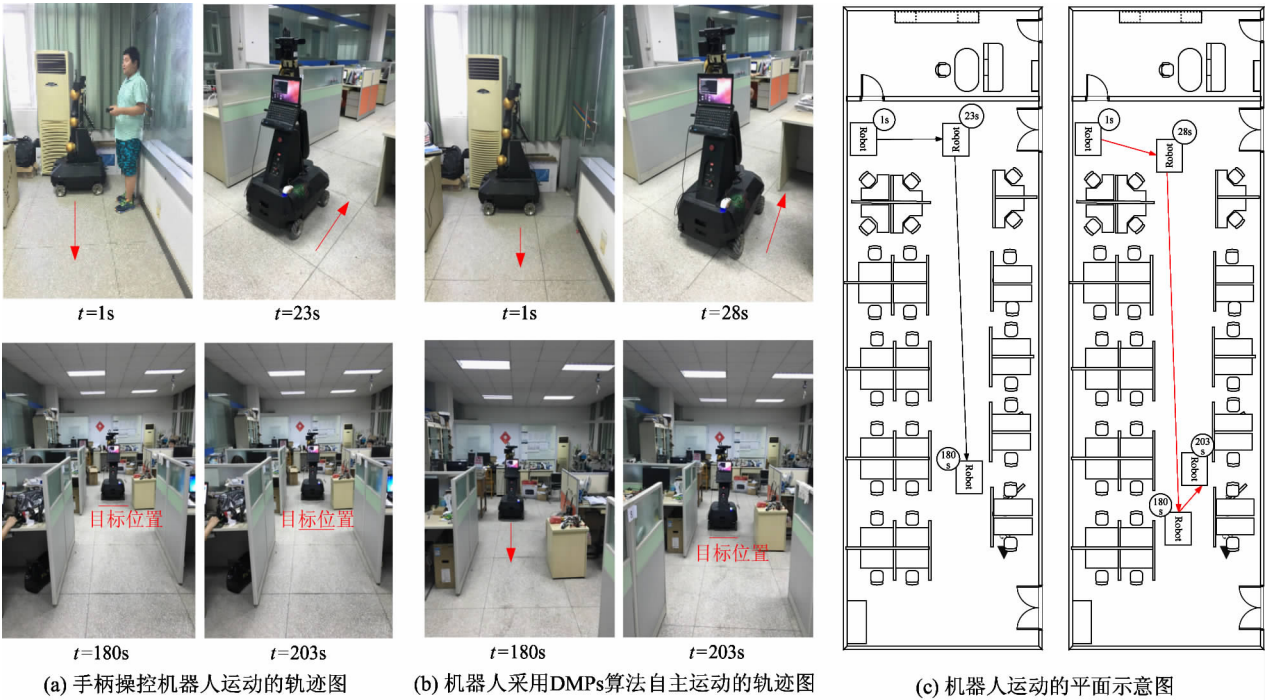


图 8 机器人训练和学习后的实验对比图

置有些超调,所以机器人之后会来回走直到到达目标位置。实验结果表明 DMPs 算法可以很好的应用在移动机器人路径规划上。

为了验证 DMPs 算法的泛化推广能力,本文将目标点位置由 $(3.65, 13.54)m$ 分别改变为 $(3, 5)m$ 和 $(4, 8)m$,移动机器人在不重新训练的前提下可以

自主到达新的目标点位置。实验结果如图 9 所示,其中 9(a) 和 9(b) 分别表示目标点位置为 $(3, 15)m$ 和 $(4, 8)m$ 时的机器人运动轨迹图;9(c) 是机器人运动的平面示意图,其中左图是目标点位置为 $(3, 15)m$ 时的示意图;右图是目标点位置为 $(4, 8)m$ 时的平面示意图。

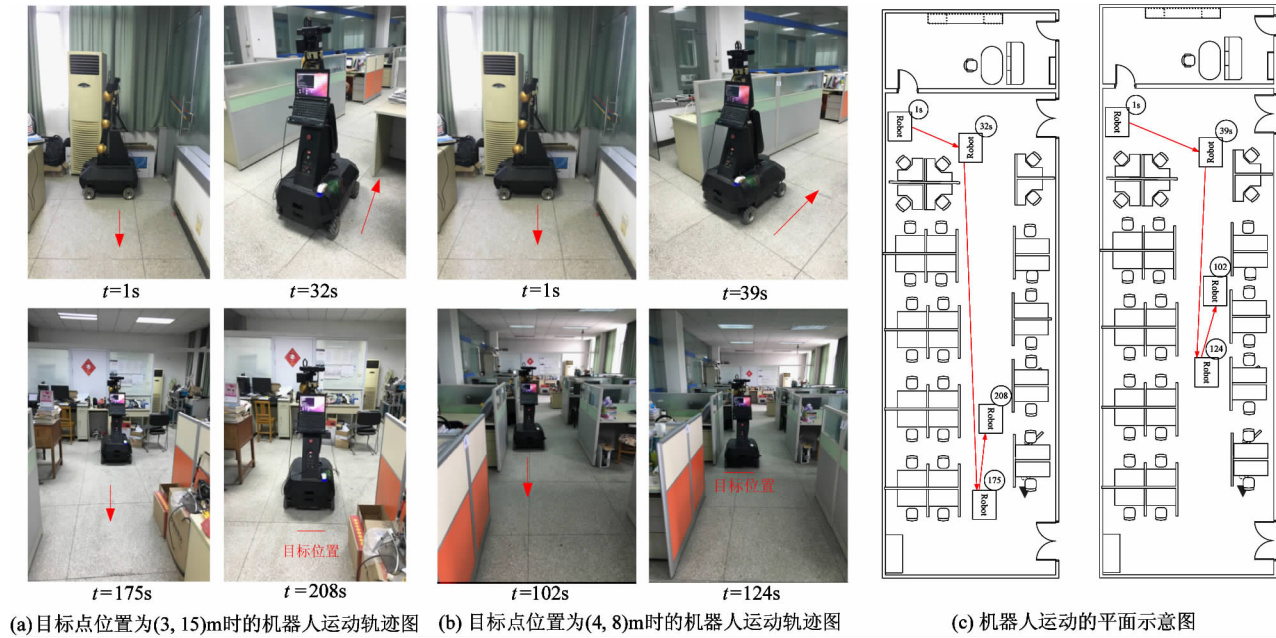


图 9 通过 DMPs 算法泛化推广时的运动轨迹平面示意图

从实验结果可以看出,移动机器人在不重新训练样本轨迹的前提下可以到达新的目标位置,并且保持之前样本轨迹的特性(有超调但是收敛于目标位置)。因此 DMPs 算法的移动机器人路径规划可以不仅仅针对某一个指定任务,对其他任务也有泛化推广的能力。

4 结 论

本文提出了一种基于 DMPs 算法的移动机器人路径规划算法,解决了传统路径规划算法中任务必须特定和实时性差等问题。DMPs 算法通过对样本轨迹进行学习并对新的任务进行泛化来规划出机器人的运动轨迹。仿真和实验结果表明该算法能有效地对样本轨迹进行学习。当目标位置改变时,DMPs 算法能在误差很小的前提下到达新的目标位置,并且 DMPs 算法具有实时性,不用重新训练样本就可以在线的学习,快速地到达目标位置。在实验研究中,由于移动机器人体积较大以及实验室现有环境的限制,本文没有对避障进行实验验证,这也是未来需要开展的工作。总之,仿真和实验结果表明,将动态运动基元方法应用于机器人的路径规划问题具有较好的可行性,并且表现出良好的应用前景。

参考文献

- [1] 宋晖, 张华, 高小明. 智能机器人路径规划与算法研究. 微计算机信息, 2006, 22(11): 244-252
- [2] 张建英, 赵志萍, 刘瞰. 基于人工势场法的机器人路径规划. 哈尔滨工业大学学报, 2006, 8(3): 1306-1309
- [3] 乔俊福, 温泽源. 模糊控制算法在移动机器人中的应用. 吉林工业大学学报, 2012, 31(3): 79-82
- [4] 甘亚辉, 戴先中. 基于遗传算法的多机器人系统最优轨迹规划. 控制理论与应用, 2010, 27(9): 1245-1252
- [5] 于振中, 闫继宏, 赵杰等. 改进人工势场法的移动机器人路径规划. 哈尔滨工业大学学报, 2011, 43(1): 49-55
- [6] 段勇, 王猛, 徐心和. 基于人工免疫与可视图相结合的移动机器人路径规划方法. 高技术通讯, 2014, 24(3): 278-283
- [7] 梁嘉俊, 曾碧, 何元烈. 基于改进势场栅格法的清洁

- 机器人路径规划算法研究. 广东工业大学学报, 2016, 33(4): 30-36 + 43
- [8] 刘建华, 杨建国, 刘华平等. 基于势场蚁群算法的移动机器人全局路径规划方法. 农业机械学报, 2015, 46(9): 18-27
- [9] Maaref M, Rezazadeh A, Shamaei K, et al. A bicycle cranking model for assist-as-needed robotic rehabilitation therapy using learning from demonstration. *IEEE Robotics and Automation Letters*, 2016, 1(2): 653-660
- [10] Cheng H, Chen H. Learning from demonstration enabled robotic small part assembly. In: Proceedings of the 9th IEEE Conference on Industrial Electronics and Applications, Hangzhou, China, 2014. 395-400
- [11] Ijspeert A J, Nakanishi J, Schaal S. Learning rhythmic movements by demonstration using nonlinear oscillators. In: Proceedings of the IEEE Conference on Intelligent Robots and Systems (IROS2002), Lausanne, Switzerland, 2002. 958-963
- [12] Ijspeert A J, Nakanishi J, Shibata T, et al. Nonlinear dynamical systems for imitation with humanoid robots. In: Proceedings of the IEEE International Conference on Humanoid Robots, Washington, USA, 2001. 219-226
- [13] Pastor P, Hoffmann H, Asfour T, et al. Learning and generalization of motor skills by learning from demonstration. In: Proceedings of the IEEE International Conference on Robotics and Automation, Kobe, Japan, 2009. 763-768
- [14] Ijspeert A, Nakanishi J, Pastor P, et al. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural Computation*, 2013, 25(2): 328-373
- [15] Fang Z, Wang G, Li W, et al. Control-oriented modeling of flight demonstrations for quadrotors using higher-order statistics and dynamic movement primitives. In: Proceedings of the 2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE), Istanbul, Turkey, 2014. 1518-1525
- [16] Tadej P, Andrej G, Leon Z, et al. Online learning of task-specific dynamics for periodic tasks. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, USA, 2014. 1790-1795
- [17] Ude A, Gans A, Asfour T, et al. Task-specific generalization of discrete and periodic dynamic movement primitives. *IEEE Transactions on Robotics*, 2010, 26(5): 800-

[18] Park D H, Hoffmann H, Pastor P, et al. Movement re-production and obstacle avoidance with dynamic move-

ment primitives and potential fields. In: Proceedings of the IEEE-RAS International Conference on Humanoid Robots, Daejeon, Korea, 2008. 91-9

Path planning for mobile robots based on dynamic movement primitives

Jiang Minghao, Chen Yang, Li Weiling

(School of Information Science and Engineering, Wuhan University of Science and Technology, Wuhan 430081)

Abstract

A novel path planning algorithm based on dynamic movement primitives (DMPs) for a mobile robot working in 2D environment is proposed. The algorithm can be described below: Firstly, the robot's trajectory is recorded while the robot is controlled by a handle. Then the recording trajectory is used as a learning sample of DMPs, and the autonomous path planning of the robot is realized, which utilizes the model parameters obtained by training the sample trajectory. In the environment with obstacles, the avoidance path can be produced through designing a coupling factor on the basis of the existing study. Based on the autonomous path planning, the generalization comes true to new goals while the original goal is changed. The simulation and experimental results show that the DMPs algorithm is feasible on a mobile robot path planning.

Key words: dynamic movement primitives (DMPs), mobile robot, 2D environment, path planning, obstacle avoidance