

基于ROS平台的六自由度机械臂 轨迹规划及仿真研究

Research on trajectory planning and simulation of 6-DOF
manipulator based on ROS

刘宇程, 姜振华, 胡俊*

LIU Yu-cheng, JIANG Zhen-hua, HU Jun*

(东华大学 机械工程学院, 上海 201620)

摘要: 针对机械臂轨迹规划算法复杂且运动过程中易产生冲击的问题, 利用开源平台机器人操作系统ROS, 提出一种在该平台下的轨迹规划方法。本课题以埃夫特六自由度 (Degrees of freedom, DOF) 机械臂为实验对象, 在SolidWorks中建立机器人模型并对其进行正逆运动学分析; 利用sw2urdf插件导出ROS识别的URDF模型文件; 通过MoveIt!功能包创建机械臂轨迹规划需要的配置和启动文件, 在Rviz中显示三维模型; 利用MoveIt!提供的C++相关API, 在该平台上对六自由度机械臂进行笛卡尔空间的直线、圆弧、NURBS曲线轨迹规划。通过分析其各关节运动信息, 验证了轨迹规划的合理性。

关键词: 机械臂; ROS; 轨迹规划; Rviz; NURBS

中图分类号: TP242

文献标识码: A

文章编号: 1009-0134(2022)01-038-04

0 引言

近年来, 机械臂在工业生产中所起的作用越来越大, 逐渐改变生产和生活方式, 虽然当前机器人技术不断提高, 也很成熟, 但随着机器人功能的日益丰富, 机器人编程技术会更加复杂且繁琐^[1], 在这种情况下, 将机器人轨迹规划与机器人操作系统及仿真平台结合起来, 这能给机器人技术研究提供新的方法。目前大多数商业机器人仿真平台不开源, 且成本较高, 不利于研究, 而开源机器人操作系统ROS具有编程语言丰富特点, 用户可根据自己的要求在该平台上开发出自己功能算法, 完全符合轨迹规划的要求。

基于ROS平台, 利用其所有的功能包方便快速地对埃夫特机械臂进行建模, 控制, 仿真。根据ROS的功能优势, 搭建了六自由度机械臂的仿真平台, 并在该平台下对机械臂的三种轨迹进行了实验, 通过其中的运动规划库完成了机械臂在笛卡尔空间下的运动规划仿真。

1 ROS及其模型配置

1.1 三维可视化平台Rviz

Rviz是一款三维可视化工具, 很好的兼容了各种基于ROS软件框架的机器人平台。通过该工具, 能够快速方便地进行机器人的相关开发, 而且ROS是一款通用的分布式机器人软件开发框架, 各功能模块通过松耦合的

方式组合, 互不干涉, 许多应用功能代码都能够重复利用, 可以有效提高工作效率^[2]。在Rviz中, 可以使用XML对机器人、周围物体等任何实物进行尺寸、质量、位置、材质、关节等属性的描述, 并且在界面中呈现出来。同时Rviz显示机器人有关信息如机器人的运动状态、位置变化, 以及周围环境的变化等^[3]。Rviz集成了许多接口, 利用其中的点云库将要规划的轨迹信息显示出来, 通过MotionPlanning插件将机械臂运动过程的轨迹与轨迹信息比较, 可以观察机械臂运动是否准确。由于Rviz是基于Qt开发的, 且很多功能插件都是集成在该工具上的, 开发者可在Rviz上开发自己的功能插件以及人机交互界面。

1.2 机器人模型建立

URDF(统一机器人描述格式)是一种基于XML规范、用于描述机器人结构的格式。以ROS中Rviz可视化仿真界面为基础, 利用该三维仿真平台的特性, 引入话题通信机制, 实现机器人模型和轨迹信息的导入。根据六自由度机器人的结构特点, 通过SolidWorks三维软件建立机器人仿真模型, 选择该软件的SW2URDF插件进行加载, 同时设定每个连杆和关节的参数如惯性矩, 转动惯量, 直角坐标以及坐标原点, 这对于后面仿真和轨迹规划分析至关重要。通过导入获得URDF文件, 建立了机器人和周围的可视化虚拟环境。如图1所示。

收稿日期: 2020-06-10

作者简介: 刘宇程 (1994-), 男, 湖南怀化人, 硕士研究生, 研究方向为智能机器人技术。

通讯作者: 胡俊 (1975-), 男, 安徽宣城人, 博士, 研究方向为数控技术与智能制造装备。

1.3 机械臂控制功能包MoveIt! 及其配置

move_group是MoveIt!中的核心^[4], move_group架构图如图2所示。开发者可以通过move_group与其他接口连接,如C++、python接口, MotionPlanning插件。通过MotionPlanning插件,开发者可设计有关运动规划算法驱动机械臂。在验证URDF文件正确之后运用MoveIt!工具包对该文件进行相应的配置,通过MoveIt!配置助手获得一个完整的运动配置功能包。经过配置后生成的功能包包含launch文件和运动规划所需的配置文件。利用运动规划器,运动学求解器找到一条到达目标姿态的较优路径,在没有机器人实体的情况下同时添加ArbotiX关节控制器,从而完成模型的运动仿真。

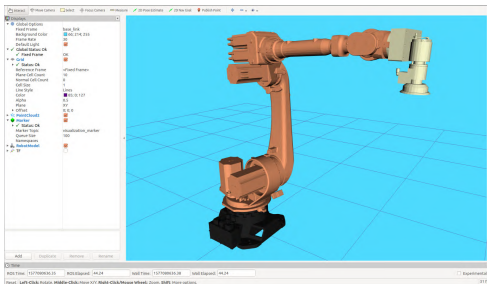


图1 可视化虚拟环境

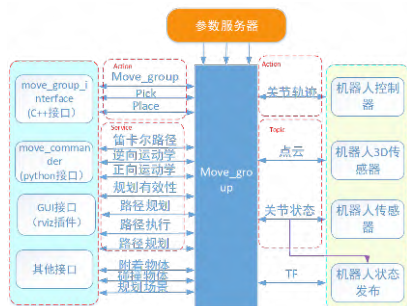


图2 架构图

2 机械臂轨迹规划

2.1 直线轨迹

机械臂末端从初始点 P_0 运动到终点 P_1 ,在笛卡尔坐标系下定义该直线路径为:

$$p(s) = P_0 + s(P_1 - P_0) \quad (1)$$

$s \in [0, 1]$ 保证机械臂末端从 P_0 经过中间一系列点到 P_1 ,且是以直线方式运动。这里的 s 可以看作是时间 t 的函数,随时间的增加完成规划。同时也可看作步距,随步距增加,得到下一个空间点直至达到终点^[5]。在ROS中利用Rviz进行直线规划的仿真,首先给定起始点 P_0 和目标点 P_1 ,利用Rviz中的功能模块进行仿真,仿真如图3所示。

直线规划是在笛卡尔空间下进行的,将其转换为关节空间下的各关节变化,关节状态信息可通过订阅/joint_states话题,再通过rqt_plot 插件将关节随时间变化的曲线

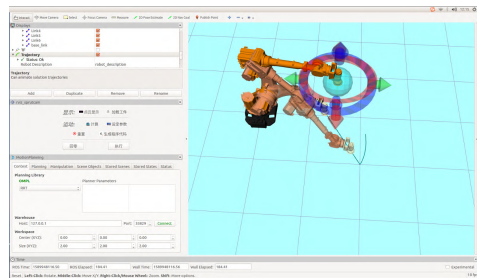


图3 直线插补

绘制出来,如图4所示。将生成的末端位置和理论位置相比较,得到误差曲线,如图5所示。

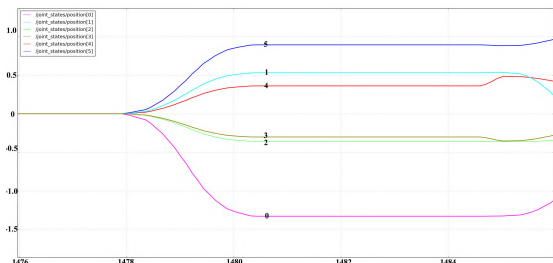


图4 关节位置变化

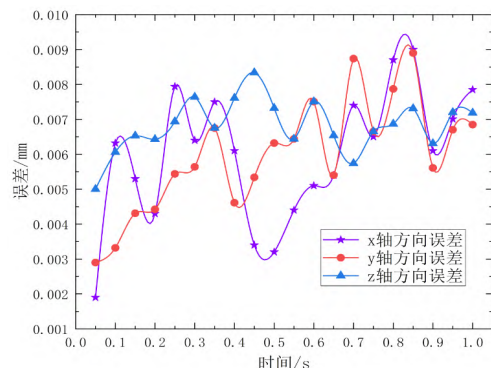


图5 误差曲线

2.2 圆弧轨迹

圆弧插补的实现流程为,将空间点转换到三个点形成的平面,将三维问题转换为二维。然后计算圆弧角,并在该平面上进行插补,最后通过变换矩阵,将插补点从二维坐标转换为三维坐标^[6]。给定空间中3点 P_1, P_2, P_3 构造新的坐标系 P_1-UVW ,以 P_1 为原点, $\overrightarrow{P_1P_2}$ 为U轴,以该三点平面法向量为W轴,则根据叉乘得到V轴表达式,三轴表达式如下得到:

$$\begin{cases} u = P_2 - P_1 \\ w = (P_3 - P_1) \times u \\ u = \frac{u}{\|u\|}, w = \frac{w}{\|w\|} \\ v = u \times w \end{cases} \quad (2)$$

得到新坐标后则 P_2, P_3 的坐标位置分别为 $(P_{2x}, P_{2y}) = ((P_2 - P_1) \cdot u, 0)$, $(P_{3x}, P_{3y}) = ((P_3 - P_1) \cdot u, (P_3 - P_1) \cdot v)$,由

三点位置可知圆心坐标为:

$$\left(\frac{P_{2x}}{2}, \frac{(P_{3x} - P_{2x})^2 + P_{3y}^2 - P_{2y}^2}{4P_{3y}} \right)$$

从而得到圆弧半径,以圆心 P_0 为原点,得到平面M新坐标系 P_0-UVW ,根据平面方程可得法向量:

$$\begin{cases} A = (y_2 - y_1)(z_3 - z_2) - (z_2 - z_1)(y_3 - y_1) \\ B = (z_2 - z_1)(x_3 - x_2) - (x_2 - x_1)(z_3 - z_1) \\ C = (x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1) \\ m = \sqrt{A^2 + B^2 + C^2} \end{cases} \quad (3)$$

以法向量作为新坐标系W轴,则W轴的方向余弦为 $\left(\frac{A}{m}, \frac{B}{m}, \frac{C}{m}\right)$ 以 $\overrightarrow{P_0P_1}$ 方向为U轴,得到该方向的方向余弦为

$\left(\frac{x_1 - x_0}{r}, \frac{y_1 - y_0}{r}, \frac{z_1 - z_0}{r}\right)$ 第三轴的方向余弦由两轴得到,为

$V = U \times W$ 得到新坐标三轴表达式后,便能得到该坐标在世界坐标的变换矩阵:

$$T = \begin{pmatrix} R & p \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} u_x & v_x & w_x & p_{0x} \\ u_y & v_y & w_y & p_{0y} \\ u_z & v_z & w_z & p_{0z} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4)$$

根据变换矩阵,得到 P_1, P_2, P_3 在世界坐标系中的表达式:

$$\begin{pmatrix} P_i \\ 1 \end{pmatrix} = T^{-1} \begin{pmatrix} P_i \\ 1 \end{pmatrix} \quad (i=1,2,3) \quad (5)$$

按照逆时针插补顺序,在新坐标系下由 P_1, P_3 计算出该两点的夹角 θ_{13} 得到插补过程中插补点的变化坐标:

$$\begin{cases} X_i = r * \cos\left(\frac{i * \theta_{13}}{N}\right) \\ Y_i = r * \sin\left(\frac{i * \theta_{13}}{N}\right), i = (0, 1, \dots, N) \\ Z_i = 0 \end{cases} \quad (6)$$

此处N为插补点个数,由上述公式得到各点的坐标还要通过变换矩阵得到在世界坐标系中的表达。结合末端执行器的姿态欧拉角,可得到末端执行器的姿态变化,在ROS中实现的圆弧插补仿真如图6所示。

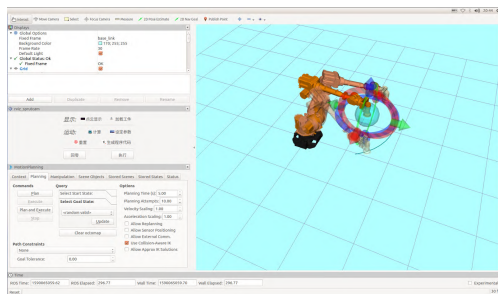


图6 空间圆弧插补

通过rqt_plot插件绘制出的关节角度变化曲线如图7所示,圆弧轨迹规划中,6个关节角度均出现变化,从图中可以看出,关节变化平滑。将ROS中生成的末端位置和理论插值位置相比较,得到误差曲线,如图8所示。

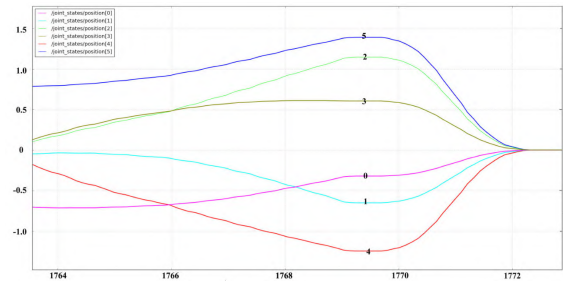


图7 圆弧插补关节位置变化

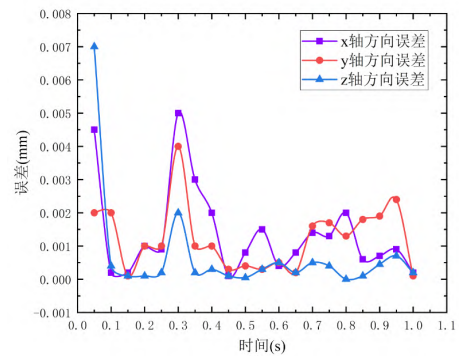


图8 误差曲线

2.3 非均匀有理B样条轨迹

非均匀有理B样条(NURBS)公式如下:

$$C(u) = \frac{\sum_{i=0}^n N_{i,p}(u)w_i P_i}{\sum_{i=0}^n N_{i,p}(u)w_i}, a \leq u \leq b \quad (7)$$

式中 P_i 为控制点, i 为序号,为 p 次规范B样条基函数,且满足以下条件:

$$N_{i,0}(u) = \begin{cases} 1, u_i \leq u \leq u_{i+1} \\ 0, else \end{cases}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+1} - u}{u_{i+1} - u_{i+1}} N_{i+1,p-1}(u) \quad (8)$$

非均匀有理B样条由 $(m=n+p+1)$ 个节点 $U=\{u_0, u_1, \dots, u_m\}$, $(n+1)$ 个控制节点 $P=\{P_0, P_1, \dots, P_m\}$ 组成,通过给定相应节点矢量,根据公式得到 $C(u)$,从而获得曲线方程。对于求 $C(u)$ 的 k 阶导数,令 $w(u) = \sum_{i=0}^n N_{i,p}(u)w_i$, $A(u) = \sum_{i=0}^n N_{i,p}(u)w_i P_i$,由公式:

$$C'(u) = \frac{w(u)A'(u) - w'(u)A(u)}{w^2(u)} = \frac{A'(u) - w'(u)C(u)}{w(u)} \quad (9)$$

由莱布尼兹求导公式知 $A(u)$ 的 k 阶导数,得到 $C(u)$ k 阶导:

$$A^k(u) = (w(u)C(u))^k = w(u)C^k(u) + \sum_{i=1}^k \binom{k}{i} w^i(u)C^{(k-i)}(u)$$

$$C^k(u) = \frac{A^k(u) - \sum_{i=1}^k \binom{k}{i} w^i(u)C^{(k-i)}(u)}{w(u)} \quad (10)$$

根据指定控制点Pi得到NURBS曲线轨迹后, 对其进行轨迹规划, 这就要求根据指定速度控制规划周期性选取插补点, 这里涉及到时间t的函数, 但是NURBS曲线是关于u的函数, 所以要计算出插补点对应的u值, 从而得到插补点对应的坐标值。

结合前人的工作, 选取泰勒级数二阶展开式进行插补, MiChing Tsai等人对这种方法进一步做了改进, 提出了一种基于“预测-校正”的算法, 并在算法的收敛性上给出了证明^[7]。该法通过给定u的一个初值预测下个周期的u值, 再通过当前速度与期望速度间的误差来校正u值, 从而满足速度规划要求。

2.3.1 NURBS曲线长度求取

得到NURBS曲线长度是进行速度规划的前提, 根据Simpson自适应积分方法^[8]求得该曲线长度, 从C(u₁)到C(u₂)的曲线长度为:

$$s = \int_{u_1}^{u_2} \left\| \frac{dC(u)}{du} \right\| du, \left\| \frac{dC(u)}{du} \right\| = \sqrt{\left(\frac{dx(u)}{du} \right)^2 + \left(\frac{dy(u)}{du} \right)^2 + \left(\frac{dz(u)}{du} \right)^2} \quad (11)$$

$$S = \sum_{i=p}^n s_i \quad (12)$$

得到每个区间的曲线长度, 然后求得总和便是整个曲线的长度。

3.3.2 速度规划

在进行速度规划算法时, 要求具有良好的平滑性, 所以通常用多项式函数法, T曲线, 指数曲线和S曲线加速算法等, 以T曲线为例, 设Ts为插补周期, Ta为加速时间, n=S/(vmax*Ts)则其速度方程为:

$$V(t) = \begin{cases} V = a_{\max} t, 0 \leq t \leq T_a \\ V = V_{\max}, T_a < t \leq nT \\ V = V_{\max} - a_{\max} t, nT < t \leq nT + T_a \end{cases} \quad (13)$$

给定曲线长度, 最大速度, 最大加速度即能实现速度规划。

2.3.3 预测-校正

根据前述的泰勒展开式对下一个插补周期的u值预测:

$$V(t) = \left\| \frac{dC(u)}{dt} \right\| = \left\| \frac{dC(u)}{du} \right\| \frac{du}{dt}, \frac{du}{dt} = \frac{V(t)}{\left\| \frac{dC(u)}{du} \right\|}$$

$$u_{k+1} \approx u_k + T \dot{u}_k + \frac{T^2}{2} \ddot{u}_{k+1}$$

$$u_{k+1} = u_k + T \frac{V(t)}{\left\| \frac{dC(u)}{du} \right\|} + \frac{T^2}{2} \left(\frac{\frac{dV(t)}{dt}}{\left\| \frac{dC(u)}{du} \right\|} - \frac{V^2(t) \left(\frac{dC(u)}{du} \frac{d^2C(u)}{du^2} \right)}{\left\| \frac{dC(u)}{du} \right\|^4} \right) \quad (14)$$

泰勒展开式对u_{k+1}进行预测, 但是计算量是比较大的^[9], 为了减少计算量, 采用向后差分方程来计算:

$$\dot{u}_k = \frac{3u_k - 4u_{k-1} + u_{k-2}}{T}$$

$$\ddot{u}_{k+1} = \frac{u_k - 2u_{k-1} + u_{k-2}}{T^2} \quad (15)$$

代入泰勒展开式可得u_{k+1}=3u_k-3u_{k-1}+u_{k-2}, 给定u₀值, 通过泰勒展开式得到u₁, u₂, 在利用上式进行迭代。但通过上式仅仅是预测出下一个插补周期的u_{k+1}还需要通过速度去校正该预测点。通过求得当前速度V_k, 与期望速度相比较, 得到较为准确的u_{k+1}^[10]。

$$\begin{cases} \alpha = \frac{V_{ke}}{-\beta(V_{ke} - V_k^{(n-1)})} \\ u_{k+1}^n = -\alpha(u_k - u_{k+1}^{(n-1)}) + u_k, \forall n \geq 1 \\ V_k^{(n-1)} = \frac{\|C^{(n-1)}(u_{k-1}) - C(u_k)\|}{T} \end{cases} \quad (16)$$

其中β为校正系数, 取[0,1]之间, V_{ke}为该点的期望速度, 也即速度规划的速度, V_k⁽ⁿ⁻¹⁾为迭代(n-1)次得到的当前速度, 迭代通过下式终止^[11]:

$$\frac{\|v_{ke} - v_k^{(n-1)}\|}{v_{ke}} \leq 0.01 \quad (17)$$

通过ROS中C++API对上述算法进行仿真, 给定控制点P₁(0,150), P₂(-100,100), P₃(100,100), P₄(0,150), P₅(-100,200), P₆(100,200), P₇(0,150)给定权重[1,30,10,1,10,30,1], 节点[0,0,0,0.25,0.5,0.5,0.75,1,1,1], 插补周期0.01s, 以及最大速度100mm/s, 最大加速度40mm/s², 根据自适应Simpson积分法, 可估算出曲线的长度为696.619mm。根据程序得到该曲线在ROS中实现的NURBS曲线仿真如图9所示, 通过话题机制以及rqt_plot插件, 绘制出的各关节角度随时间变化曲线如图10所示。

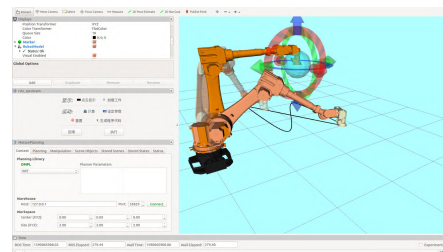


图9 仿真

根据速度公式迭代得到u值, 从而得到速度规划。由速度规划图得知该算法有效(速度误差小于0.01), 如图11所示。

【下转第50页】

其优化设计的可行性和正确性。

5 结语

本文研制了一种可在曲面爬行的大型钢结构巡检爬壁机器人,重点开展永磁轮的磁场仿真,对其结构参数进行了优化。

1) 对爬壁机器人三种失稳形式进行了静力学分析,得到永磁轮产生的吸附力应大于304.65N,为其优化设计提供理论目标。

2) 利用COMSOL软件对永磁轮进行仿真分析,确定了结构参数。结果显示:优化后永磁轮吸附力为357.8N,吸附效率提高了24.2%,机器人能在曲率半径大于0.87m的钢结构上可靠吸附。

3) 进行样机实验,验证了永磁轮能提供稳定可靠的吸附力,且爬壁机器人能够以任意姿态在不同角度壁面上灵活爬行。

参考文献:

[1] Han I H, Yi H, Song C W, et al. A miniaturized wall-climbing segment robot inspired by caterpillar locomotion[J]. Bioinspiration & Biomimetics,

2017, 12(4).

- [2] Alkalla M G, Fanni M, Mohamed A F, et al. EJBOT-II: an optimized skid-steering propeller-type climbing robot with transition mechanism[J]. Advanced Robotics, 2019, 33(20):1042-1059.
- [3] 崔宗伟, 孙振国, 陈强, 等. 两端吸附式焊缝修爬壁机器人研制[J]. 机器人, 2016, 38(01):122-128.
- [4] Li P, Duan X, Sun G, et al. Design and control of a climbing robot for inspection of high mast lighting[J]. Assembly Automation, 2019, 39(1):77-85.
- [5] 潘柏松, 张晋, 魏凯, 等. 基于Halbach阵列爬壁机器人永磁轮吸附单元的设计与优化[J]. 浙江工业大学学报, 2015, 43(04):393-397+474.
- [6] Howlader MD OF, Sattar TP. Design and optimization of permanent magnet based adhesion module for robots climbing on reinforced concrete surfaces[C]//SAI Intelligent Systems Conference. London, England: IEEE, 2016, 650:153-171.
- [7] 袁硕, 刘天羽. 基于Ansoft的电力机器人永磁吸附单元有限元分析[J]. 电机与控制应用, 2018, 45(02):71-74, 127.
- [8] 陈彦臻, 胡以怀, 袁春旺, 等. 爬壁机器人永磁吸附轮的磁路及结构设计分析[J]. 制造业自动化, 2019, 41(06):83-87.
- [9] 侯嘉瑞, 万熠, 孙立新. 风电塔筒爬壁机器人吸附结构设计分析[J]. 机电工程, 2018, 35(09):939-943.
- [10] 李赵春, 李思豪, 董帅. 磁流变弹性体的制备及其摩擦磨损特性测试[J]. 功能材料, 2019, 50(12):12148-12156.



【上接第41页】

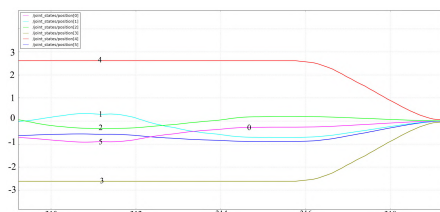


图10 关节位置变化

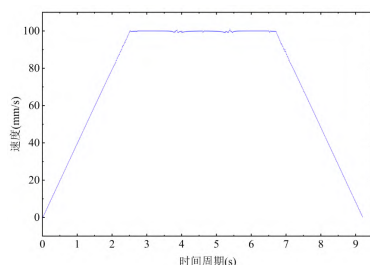


图11 速度规划

3 结语

本文通过ROS机器人平台对六轴工业机器人进行轨迹规划算法研究和仿真实验,完成了在笛卡尔空间的直线和圆弧轨迹, NURBS曲线的规划,并用Rviz进行了仿真;通过预测-矫正环节,对NURBS曲线进行速度规划,使得速度具有平滑性,避免机器震动引起故障。结合在ROS平台上的仿真,以及关节状态信息变化验证了该方法的正确性,对ROS机械臂开发具有一定的指导意义。同时利用

ROS该平台的功能优势,有效的减小了开发难度,提高了开发效率。

参考文献:

- [1] 刘磊, 宁伟. 基于ROS的六自由度机械臂轨迹规划[J]. 自动化与仪表, 2018, 33(03):22-25, 51.
- [2] Lentin Joseph. Mastering ROS for Robotics Programming [M]. Birmingham: Packt Publishing Ltd, 2015.
- [3] 胡春旭. ROS机器人开发实践[M]. 北京:机械工业出版社, 2018.
- [4] Coleman D, Sucan I, Chitta S, et al. Reducing the barrier to entry of complex robotic software: a moveit! case study[J]. Eprint Arxiv, 2014.
- [5] 布鲁诺, 西西里安诺, 洛伦索, 夏维科, 路易吉, 维尼拉, 等. 机器人学建模、规划与控制[M]. 西安:西安交通大学出版社, 2015.
- [6] 李浩, 郑智贞, 黄顺舟, 等. 基于ROS的工业机器人轨迹规划和仿真[J]. 组合机床与自动化加工技术, 2018(12):59-62.
- [7] Tsai M C, Cheng C W. A Real-Time Predictor-Corrector Interpolator for CNC Machining [J]. Journal of Manufacturing Science and Engineering (S1087-1357). 2003, 125(3):449-460.
- [8] 姜健飞, 吴笑千, 胡良剑. 数值分析及其MATLAB实验(第二版)[M]. 北京:清华大学出版社, 2015.
- [9] Nakamura S. Applied Numerical Methods in C[M]. 3rd. USA: Prentice-Hall International, 1993.
- [10] 王超. 基于LinuxCNC的工业机器人B样条插补算法的研究与实现[D]. 华南理工大学, 2017.
- [11] 王昕, 王均伟, 饶志, 等. 基于NURBS曲线轨迹规划与速度规划的研究[J]. 系统仿真学报, 2008, 20(15):3973-3980.