|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 录用批号 | 文章分类 | 文章编号 | 审稿费 | 稿件号 | 页面 | 收稿日期 | 20年月日 |
|  |  |  |  | CS |  | 修回日期 |  |

**基于运动描述语言的机械臂轨迹生成方法**

刘钊铭，刘乃龙，魏青，崔龙

（中国科学院沈阳自动化研究所，辽宁 沈阳 110016）

**摘要：**稿件摘要篇幅应在200至300字（5-6行），其中要包含有目的、方法、结果、结论四要素。全文篇幅为五号字体（包括题目和文献,此为送审模板，请勿参照出刊书样）A4纸张4-页以上。传统的机械臂轨迹生成方法

**关键词：** 机器人；运动描述语言；轨迹生成；机器人仿真

**中图分类号：**TP242.2  **文献标识码**：A

**A MDL Method for Robot Manipulator Trajectory Generation**

LIU Zhao-ming，LIU Nai-long, WEI Qing, CUI Long

(Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang Liaoning 110016, China)

**ABSTRACT：**Large

**KEYWORDS：**digital watermark; wavelet packet; adaptive; feature of texture

**1 引言**

运动描述语言（Motion Description Language）是由哈佛大学的Roger Brockett教授提出的用于机器人控制的理论。Brockett教授指出，机器人系统是一种连续信号和离散信号相互混合的系统，具有混杂系统（Hybrid System）的特征[[1-3](#_ENREF_1)]。1995年，Manikonda提出了MDLe模型，这是一种加入了中断描述函数的扩展MDL模型，他把这种方法用于非完整性移动机器人的运动规划与控制[[4](#_ENREF_4)]。2005年，哈佛大学的李洪谊提出了DMDL模型，这种模型应用于包含了惯性的动力系统的控制。他以倒立摆系统为研究对象，运用DMDL模型得到了很好的控制结果[[5](#_ENREF_5)]。随后中科院沈阳自动化研究所的化建宁在机器人遥操作系统中应用了MDL方法，针对网络遥操作机器人系统的特点，基于MDL方法设计了新的系统控制结构，成功解决了网络遥操作系统中存在的通信带宽不足和计算能力较弱的问题[[6-8](#_ENREF_6)]。2010年，佐治亚理工的Patrick J. Martin再次完善了MDL理论的基础框架，丰富了MDL理论的许多细节，并设计了MDL控制系统的软件结构，使它更适用于具有能源、驱动、传感和通信约束的控制系统[[9](#_ENREF_9)]。2012年，佐治亚理工的Gargas 以MDLe为基础，提出了一套移动机器人运动控制的方法。这种方法运用希尔伯特空间理论，把移动机器人的运动控制信号分解到运动描述基元，实现在减少控制系统通信要求的基础上保持了原有运动轨迹的精确性[[10](#_ENREF_10)]。

MDL方法的核心思想就是把底层的控制问题交给执行机构本身，高层控制系统只负责整体规划，不考虑执行单元的具体细节，以前馈功能为主。在传统的机械臂控制系统中，每个关节的轨迹控制信息在机器人核心控制器中根据一定的约束条件生成，并对这条满足约束条件的轨迹进行采样，最后以固定的通信频率将这些采样点通过总线系统发送到关节驱动器，控制关节按照给定参考信号运动。由于系统传输的是轨迹的离散采样点，因此在这种系统架构条件下，如果想提高机械臂的轨迹运动精度，就只能提高采样频率，加快通信速度。然而MDL的控制系统架构则与之不同，在该系统的框架下，不再传输离散的轨迹采样点，而是直接传输连续的轨迹函数。这些函数利用类似傅立叶变换的技术，被映射到一些可以参数化表示函数基元上，通信系统只需要传输几个简单的参数，就可以在关节端重构出轨迹函数，驱动关节运动。因此在这种控制系统架构下，提高机械臂的轨迹精度无需提高通信频率，实现了降低总线负载而不损失轨迹精度的诉求。

在这篇论文中，我们将介绍运动描述语言的基础知识，然后介绍用于机械臂轨迹生成的运动描述语言理论和它的算法实现。最后，通过V-REP机器人仿真平台进行仿真实验。

**2 MDL基础**

2.1 Brockett基础理论

考虑一个具有如下形式的系统：

其中u、x、y都是时间的函数。u是输入的控制率，它是一个m维的连续函数。x是n维的系统状态。y是一个p维的系统输出。在MDL理论中，这个系统的模型可以被分解成许多段，每一段用一个三元组(u,k,t)来表示，我们把它叫做一个运动基元（atom）。从初始时刻开始，MDL编译器接收到一个运动基元的序列，然后编译器可以将这个序列转换成如下的分段仿射模型：

关于这个分段仿射模型与原系统的关系，Brockett教授给出了如下定理：

**定理2.1**：如果关于x的连续函数G满足利普希茨连续性条件，那么这个分段仿射模型可以对任意曲线生成一个足够好的逼近[[2](#_ENREF_2)]。

这个定理构成了整个运动描述语言理论的基础，它使一个仿射系统可以被分段表示。这样我们只要把一个连续系统分段表示，然后将每一段连续的小系统参数化，就可以用一段离散的符号序列来表示和控制一个连续系统

2.2 机械臂轨迹生成理论

机械臂的运动控制分为路径规划（Path Planning）和轨迹生成（Trajectory Generation）两个层次。路径规划的是控制的高层次内容，它的任务是在末端笛卡尔空间生成一条满足要求并且规避碰撞的合理路径；轨迹生成则位于整个系统的低层，它负责在机械臂的关节空间根据末端轨迹和速度、加速度等约束生成足够平滑的运动轨迹，一般用样条曲线插补的方式实现轨迹的平滑，用优化理论使轨迹满足约束条件[[11](#_ENREF_11)]。

事实上，在关节空间产生的轨迹都是使用点到点（PTP）运动方式实现的。所谓点到点运动，就是对关节的运动轨迹以为采用时间进行离散采样，然后在两个离散的采样点直接使用样条曲线进行轨迹插补，只要采样时间足够小，实际的运动轨迹的就一定会达到期望精度，而插补所用的样条曲线可以使每段轨迹的速度、加速度等满足连续性条件，抑制机械臂的震动。关于样条曲线，一般根据轨迹的连续性要求进行选取，目前已经有了成熟的理论[[12](#_ENREF_12)]。常用的样条曲线有二次、三次、五次等样条曲线：

这些样条曲线就是构建机械臂轨迹生成MDL系统运动基元的基础。

**3. MDLg理论**

本节将介绍用于机械臂轨迹生成的MDLg理论的基础。MDLg以传统的MDL理论和加入了中断的改进型MDLe为基础，专门用于机械臂的控制。用于机械臂轨迹生成的MDLg架构具有一个主端和多个从端。MDLg生成器位于机器人的主端核心控制器，将关节的运动轨迹映射成一系列基于运动基元的运动序列字符串。MDLg解析器位于机器人的从端关节控制器，将主端发送来的运动序列解析为运动参考函数，并基于关节的动力学模型对关节运动进行控制。

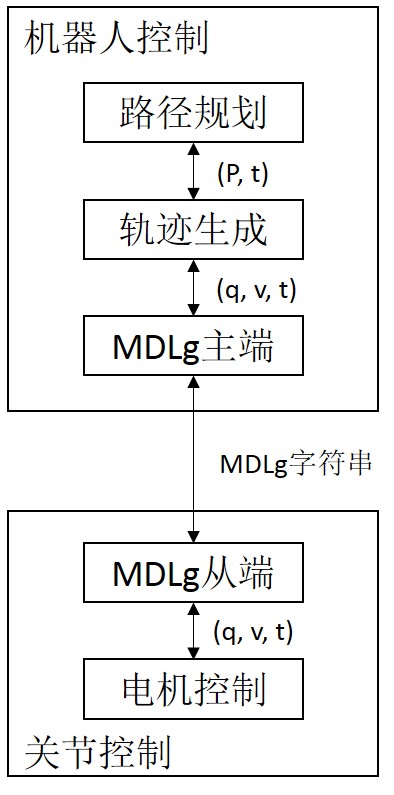


图1 基于MDL的机械臂控制架构

下面给出MDLg模型的定义。考虑一个具有如下形式的系统：

其中

对于这个系统，下面开始定义MDLg体系的运动基元。

**定义3.1**（运动基元）：MDLg理论中，运动基元（atom）是一个三元组符号，用来表示，其中

其中是输入控制率，是一个触发中断的布尔函数

其中是表示时间的正整数，而表示一个来自关节传感器的k维信号反馈。在发生紧急情况的时候，这个函数会触发中断，停止机械臂的运动。

当这一串运动基元被传输到MDLg从端的时候，就可以构成一个运动序列

从端将这个运动序列重构以后就可以得到系统的状态方程

**定义3.2**（缩放基元）：**缩放基元**是对一个运动基元进行标量数乘，对控制信号进行放大或者缩小，对执行时间进行压缩或者扩张。

等式左边就是一个缩放基元，并且把称为缩放系数。

通过这个定义，可以对执行时间不同的各个控制基元进行时间扩张，使它们的时间尺度一致，有利于基元序列的存储和传输。

**定义3.3**（字母表）：字母表用符号表示，是有限个运动基元的集合,

其中对都满足。

字母表的定义表明，字母表中的每一个运动基元都是独立的，它不能使用缩放操作被另外一个基元表示。接下来将要定义基元的合并方法，首先需要引入运动基元的合并运算：

在运动基元合并运算的基础上，就可以定义合并基元：

**定义3.4**（合并基元）：合并基元是对字母表中的所有运动基元进行缩放与合并操作后得到的新的运动基元

其中是字母表中基元的个数。另外用符号表示合并基元中所有缩放系数的组合，称为合并基元的系数。

**定义3.5**（MDLg字符串）：MDLg字符串是一系列合并基元的系数的组合，用符号MDLg表示

如果将一个MDLg字符串传输到到MDLg从端，从端就得到了一个合并基元序列。从端将这个合并基元序列重构以后，就可以得到一个满足如下状态方程组的系统

**4.算法实现**

这个部分将会介绍MDLg算法的具体实现。MDLg算法包含两个部分：控制信号的分解与控制信号的重构。整个算法流程大致包含三个步骤：第一步是对控制信号的分割，根据某种规则将控制信号在时间域上分割成不同时间长度的小段，这步称为**时域分割**；第二步是将已经被分段的信号分解到运动基元上，这一步称为**空间分解**；第三步是**信号重构**，就是通过MDLg字符串信息在机器人关节端重构出控制信号的过程。下图展示了整个算法的信号流程图。

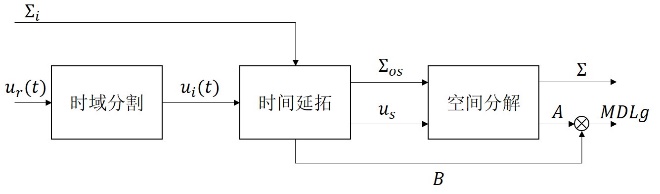


图2 算法实现信号流图

4.1时域分割

时域分解是对参考控制信号的预处理过程，把输入信号根据某种规则分割成许多不同时间长度的小段。这一过程有两个输入，分别是控制信号和初始字母表。给定的控制输入信号是一个在时间域上连续的函数。然而，事实上计算机控制器并不能处理连续信号。因此，要以时间间隔对控制信号进行离散化处理，可以得到控制信号的离散形式为向量。接下来需要在采用信号中找到变化率比较大的点。首先计算出输入信号的二阶导数

根据二阶导数值就可以找到用于分段的过渡点。所谓过渡点就是满足下面条件的点

其中是由用户给定的控制变量。这一过程将会输出一个矩阵，它的行数n表示一共分割出的段数，列数m是采样点最多的那一个分割段的采样点个数，其它不足m个采样点的分割段则用0填充。

完成分割过程之后，就要对分割段进行时间尺度扩张，使每一个分割段的时间尺度一致。因此，这一过程有两个输入，分别是前面经过分割得到矩阵和初始字母表。输出则是完成了时间尺度扩张的分割段矩阵和，以及扩张尺度系数。扩张以后每一段的时间尺度变为。

首先用下面的方程计算时间扩张系数

其中

可以得到

当计算出所有的时间扩张系数以后，可以把它们总结成一个向量

输出的这个向量用于对扩张后的向量进行反扩张，得到原始分割段。

4.2空间分解

时域分割过程完成之后，就要对得到的被扩张的分割段进行空间分解，把它们分解映射到运动基元上。然而这里存在一个棘手的问题，就是初始化的运动基元可能不够，无法满足控制信号分解重构的精度要求，这就需要在分解的同时对运动基元进行必要的扩充。空间分解过程的输入信号为扩张后的分段信号矩阵和字母表矩阵。输出则是经过扩展的字母表和表示运动序列的MDLg字符串。

根据函数空间理论，参考控制信号是空间中的函数。因此我们可以对这个函数进行正交分解，正交分解基就是已经得到的字母表。依据希尔伯特空间正交分解理论，对每一个分割段进行正交分解后，这个函数就可以表示成如下形式：

参考信号就可以表示成一些基函数的线性组合，这些基函数就是运动基元。下一步就是计算出线性组合的参数，这里应用希尔伯特空间理论，根据内积的性质

从到排列成矩阵，可以得到下面的矩阵方程

这个矩阵方程可以简化为，因此参数向量可以表示为

如此操作把所有的分割段都进行分解，得到矩阵A

但是，这里存在两个问题。其一是如何选择初始的运动基元；其二是初始的运动基元可能不足。对初始运动基元的选择，基于轨迹插补理论，选择一些样条曲线作为运动基元。第二个问题则需要构建一种扩张运动基元的方式，使参考信号获得更完美的分解。

如果参考信号经过初始运动基元分解后，误差不满足要求，那么可以令

因此可以得到

这样就得到了新的运动基元，其中的控制率为。扩展后的字母表是。这一过程将会被反复执行，直到分解满足给定精度要求为止。

连续函数上的内积一般由如下公式定义

但是在实践中，需要处理的是离散点的内积。就需要使用数值内积算法

整个算法的细节将在算法表格中展示。

4.3信号重构

本过程将在MDLg从端重构参考信号，完成对机器人关节的控制。通过前面两个过程，可以获得扩展后的字母表和MDLg字符串矩阵A以及扩张系数向量。字母表就被预先存储在MDLg从端也就是关节中，我们只需要把字符串A和扩张系数实时传输给机器人关节控制器，就可以完成机器人轨迹的控制。

信号重构分为两个步骤。首先是重构参考信号，在这个过程中，可以得到每一个分割段。但是每个分割段的执行时间并不是正确的。因此，第二步就是应用扩张系数使所有分割段回归初始的运行时间。这样把所有分割段拼接在一起就得到了初始参考信号。

所有的基元段被顺序执行，这样就可以获得参考轨迹。同时，每一个运动基元都可以接收到来自关节传感器的反馈信息，只要出现异常情况，就可以控制关节和机器人停止运动，保证系统安全。

表1 MDLg算法

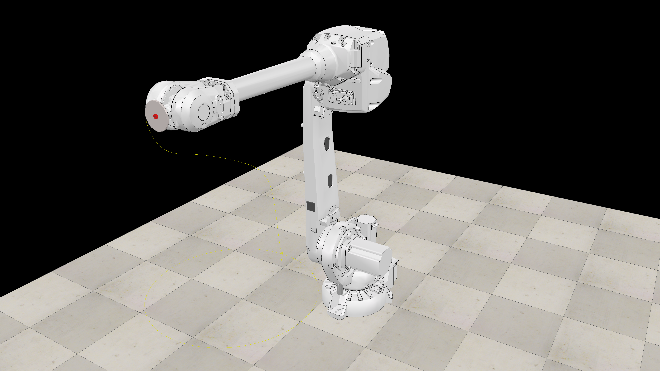
|  |  |
| --- | --- |
| **算法1** MDLg算法实现 | |
| **输入：** |  |
| **输出：** |  |
| 1: | 依据方程（）根据输入; |
| 2: | 分割; |
| 3: | 缩放; |
| 4: | 根据方程（）计算; |
| 5: | 根据方程（）计算矩阵; |
| 6: | ; |
| 7: | **while** 映射 **do** |
| 8: | **While** !done **do** |
| 9: | 根据方程（）计算; |
| 10: | 根据方程（）计算; |
| 11: | 计算; |
| 12: | **if** **then** |
| 13: |  |
| 14: | done = TRUE; |
| 15: | **else** |
| 16: | 根据方程（）计算; |
| 17: | ; |
| 18: | 根据方程（）重新计算矩阵; |
| 19: | done = FALSE; |
| 20: | **end if**; |
| 21: | **end while**; |
| 22: | **end while**; |

**5.仿真研究**

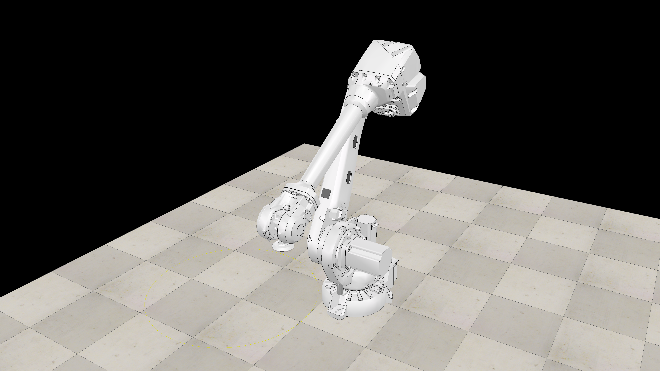
为了验证理论的实际效果，我们用V-REP和Matlab进行了联合仿真实验。V-REP（Virtual Robot Experimentation Platform）是Coppelia Robotics公司开发的一款跨平台机器人仿真软件，它是一个集成开发环境，采用分布式控制架构，可以进行机器人模型创建、动力学仿真、控制器开发，支持C/C++、Python、Java、Lua和Matlab等编程语言。使用V-REP平台可以快速实现机器人控制算法的开发与仿真。

本文仿真使用V-REP 3.4.0和Matlab 2016a。机器人模型在V-REP中建立，利用V-REP自带的物理引擎进行实物仿真。MDLg轨迹生成算法在Matlab中开发实现，通过V-REP提供的Remote API接口与Matlab连接，将关节的运动数据发送给V-REP中的机器人模型，控制机器人运动。

本次选择ABB的IRB4600机器人模型进行仿真。IRB4600是一款传统的6轴机械臂，我们让这个机械臂的末端走一个圆形轨迹以验证MDLg轨迹生成算法的有效性。



在V-REP中将仿真步长设置为1ms，开启实时仿真模式（real-time mode），在MDLg轨迹生成算法下，机械臂实现了末端的圆形轨迹。完整的仿真代码和视频录像已经被上传到网络空间，读者可以下载参考。



使用MDLg算法，主端控制器只需要向关节控制器实时传输包含缩放系数的MDLg字符串。这种构型的六轴机器人，末端画出如图所示的圆周，需要转动1、2、3、5轴，4轴几乎不转动同时也不考虑6轴的转动。通过下面表格的对比可以看到，末端画圆的情况下，MDLg算法需要实时传输的数据量不足100个，与传统方式相比数量大幅减少。

表 2 MDLg算法与传统方法的数据传输量

|  |  |  |
| --- | --- | --- |
|  |  | |
| MDLg | 传统方法 |
| 数据传输量 | 54 | 519 |

**6.结论**

运动描述语言（MDL）是一种分布式的控制体系结构，适用于连续和离散信号相混杂的系统的控制，机器人系统正式具有这种特征的控制系统。使用本文提出的MDLg控制架构和轨迹生成算法，可以在不损失或者损失极少轨迹精度的条件下，大幅减少机器人运动过程中的实时数据传输量，从而实现机器人控制系统的优化和简化。

但是必须指出，本文只是基于运动描述语言的机器人控制系统的初步理论，目前只进行了算法的仿真验证，要想真正实现这种优化的控制系统，还需要在真实的机器人上进行实验研究。这就要进一步考虑机器人的动力学模型、关节控制器设计以及总线时钟同步方法等内容。因此，下一步任务就是设计一款具有MDLg空间重构能力的机器人关节控制器，进行多关节联动实验。

**参考文献：**

[1] Brockett R. Formal languages for motion description and map making[J]. Robotics, 1990, 41: 181-191.

[2] Brockett R W. On the computer control of movement[C]. Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on, 1988: 534-540.

[3] Brockett R W. Hybrid models for motion control systems[M]. Springer, 1993.

[4] Manikonda V, Krishnaprasad P S, Hendler J. A motion description language and a hybrid architecture for motion planning with nonholonomic robots[J]. Proceedings of 1995 Ieee International Conference on Robotics and Automation, Vols 1-3, 1995: 2021-2028.

[5] Li H. Choreographing Dynamical Systems[D]. Harvard University, 2004.

[6] 化建宁. 基于网络的机器人遥操作系统: 运动描述语言方法[D]. 2008.

[7] 化建宁, 崔玉洁, 贾琪, et al. 基于MDL的机器人网络遥操作系统控制方法[J]. 机器人, 2013, 35(5): 615-622.

[8] 化建宁, 符秀辉, 郑伟, et al. 基于运动描述语言的轮式移动机器人控制[J]. 机器人, 2006, 28(3): 316-320.

[9] Martin P J. Motion description languages: from specification to execution[D]. Georgia Institute of Technology, 2010.

[10] Gargas Iii E F. Generation and use of a discrete robotic controls alphabet for high-level tasks[D]. Georgia Institute of Technology, 2012.

[11] Craig J J, 贠超. 机器人学导论[M]. 北京: 机械工业出版社, 2006.

[12] Biagiotti L, Melchiorri C. Trajectory Planning for Automatic Machines and Robots [M]. Springer Science & Business Media, 2008.