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与控制方法及仿真研究

刘钊铭[1,2]，刘乃龙[1,2]，魏青[1,2]，崔龙[1]

（1. 中国科学院沈阳自动化研究所，辽宁 沈阳 110016；2. 中国科学院大学， 北京 100049）

摘要：传统的机械臂轨迹生成方法基于轨迹插补理论，主控制器以固定的时间间隔采样关节的轨迹函数，并将位置、速度等信息发送给关节控制器。这种方法本质上是一种离散采样方法，轨迹精度的高低依赖于时间间隔的长短，提高轨迹精度就要减小时间间隔，提高采样频率。但是通信总线存在带宽和实时性等限制，采样频率不能随意提高。针对传统方法的这种特点，本文将提出基于运动描述语言（MDL）的机械臂轨迹生成与控制方法，建立一种具有混杂系统特征的机械臂控制系统架构，应用函数的空间分解理论，摆脱发送离散采样点的方式，直接将参数化的连续轨迹函数发送给机器人的关节控制器，大幅度减少通信总线传输的数据量。机械臂将在轨迹精度不损失的前提下，降低通信频率，减小总线负载。本文还通过仿真软件验证了方法的有效性。

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 **文献标识码：**A

A Research of MDL Method for Manipulator Trajectory Generation and Control Including Simulation

LIU Zhao-ming[1,2]， LIU Nai-long[1,2], WEI Qing[1,2], CUI Long[1]

(1. Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang Liaoning 110016;
2. University of Chinese Academy of Sciences, Beijing 100049)

ABSTRACT： Traditional trajectory generation method based on trajectory interpolation. The joint's trajectory function is sampled at a fixed time interval in the core controller, then the position, velocity and so on are be transformed to the joint controller. Actually, this method a discrete sampling method, and the accuracy of trajectory depends on time interval. That means the higher sampling frequency the more accuracy. However the bandwidth and real-time capability of communication bus is limited, the sampling frequency cannot be increased arbitrarily. In view of the feature of traditional method, this paper will propose a method of manipulator trajectory generation and control based on motion description language (MDL), and establish an architecture for robotic control system with hybrid system features. We will use the function space decomposition theory to get rid of the way of sending discrete sampling points and directly send the parameterized continuous trajectory function to the joint controller, greatly reducing the amount of data transmitted by the communication bus. The manipulator will reduce the communication frequency and bus load on the premise of no loss of track accuracy. This article also verified the validity of the method through simulation software.

KEYWORDS： Robot; MDL; Trajectory Generation; Robotic Simulation

1 引言

运动描述语言（Motion Description Language），简写 MDL，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，可以用一种具有语言的特征依靠字符串驱动的方法进行控制，这就是 MDL 方法[1-3]。Brockett 教授提出 MDL 理论后，又有许多学者对该理论进行了完善和应用研究。1995 年，马里兰大学的 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]。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]。2008 年，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，针对遥操作机器人系统的特点，基于 MDL 方法设计了新的遥操作控制系统架构，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]。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、感知和通信约束的控制系统[9]。2012 年，佐治亚理工的 Gargas 以 MDLe 模型为基础，建立了一套移动机器人运动控制与规划的方法。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，在减小控制系统通信频率的基础上保持了原有运动轨迹的精确性[10]。

MDL 理论的核心思想就是把底层具体的运动细节交给执行机构控制，高层控制系统只负责整体运动规划，不考虑执行单元的具体细节，以前馈执行为主。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，并对轨迹函数进行采样，最后以固定的通信频率将这些采样点通过总线发送到关节驱动器，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下，如果想提高机械臂的轨迹精度，就只能提高采样频率，加快通信速度。基于 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，而是直接传输参数化的连续轨迹函数。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的运动基元上，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关节轨迹函数，驱动关节运动。因此在这种系统架构下，通信频率将大幅降低而轨迹精度却几乎不会损失。

在这篇论文中，我们将首先介绍 MDL 的基本理论，然后基于 MDL 理论建立专用于机械臂轨迹生成与控制的 MDLg 模型并给出它的算法实现。最后，通过 V-REP 机器人仿真平台进行仿真实验。

2 MDL 基本理论

2.1 Brockett 的基本理论

考虑一个具有如下形式的系统：

$$\begin{aligned} \dot{x} &= f(x) + G(x)\big(u(t) + k(x)\big) \\ y &= h(x) \end{aligned} \tag{1}$$

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;k是状态反馈。在MDL理论中，这个系统的模型可以被分解成许多小段，每一段用一个三元组(u,k,t)来表示，这个三元组被称为运动基元（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翻译成如下的分段仿射系统：

$$\begin{aligned} &\dot{x} = f(x) + G(x)(u_1(t) + k(x)), y = h(x); \tau_0 \le t < \tau_1 \\ &\dot{x} = f(x) + G(x)(u_2(t) + k(x)), y = h(x); \tau_1 \le t < \tau_2 \\ &\quad\vdots \\ &\dot{x} = f(x) + G(x)(u_i(t) + k(x)), y = h(x); \tau_{i-1} \le t < \tau_i \end{aligned} \tag{2}$$

关于这个分段仿射系统与原系统的关系，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(x)满足利普希茨连续性条件，那么形如方程组(2)的分段仿射模型可以对系统(1)生成一个足够好的逼近[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。因此只要将一个连续系统分段表示，然后将每一个小段用参数化的运动基元(u,k,t)表示，就可以用一串离散的符号序列来驱动一个连续系统。

2.2 机械臂轨迹生成

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。路径规划是控制的高层次内容，它的任务是在笛卡尔空间生成一条满足要求并且规避碰撞的合理末端路径；轨迹生成任务在路径规划完成之后，它负责在机械臂的关节空间根据已经产生的末端路径和速度、加速度等约束条件生成足够平滑的关节运动轨迹。一般用样条曲线插补的方法实现轨迹的平滑，用优化理论使轨迹满足约束条件[11]。本文关注的就是机械臂的轨迹生成，基于运动描述语言理论建立一种用于机械臂轨迹生成的MDLg模型。

事实上，传统方法在关节空间产生的轨迹都是使用点到点运动方式实现的。所谓点到点运动，就是对关节的运动轨迹以Δt为采用时间进行离散采样，然后在两个离散的采样点之间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会达到期望精度。关于样条曲线，一般根据轨迹的连续性要求进行选取，已经形成了成熟的理论[12]。常用的样条曲线有二次、三次、五次样条曲线等：

$$\begin{aligned} &\mathrm{q(t)} = \mathrm{q}_2 t^2 + a_1 t + a_0 \\ &q(t) = a_3 t^3 + \mathrm{q}_2 t^2 + a_1 t + a_0 \\ &q(t) = a_5 t^5 + a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0 \end{aligned} \tag{3}$$

这些样条曲线就是构建机械臂轨迹生成MDLg模型运动基元的基础。

3. MDLg 模型

本节将介绍用于机械臂轨迹生成的MDLg模型。MDLg模型以MDL理论和加入了中断描述的MDLe模型为基础，结合机械臂轨迹生成与控制的特点建立。基于MDLg模型的机器人控制系统架构如图1所示，是一种主从结构。MDLg生成器位于机器人的主端核心控制器中，它将关节的运动轨迹映射成一系列基于运动基元的字符串序列。MDLg解析器位于机器人的关节控制器，将主端发送来的字符串序列解析为轨迹参考函数，并基于关节的动力学模型对关节运动进行控制。

下面给出MDLg模型的定义。考虑一个具有如下形式的系统：

$$\dot{x} = f(x) + \sum_{i=1}^{m} g_i(x)u_i(t)\,; y = h(x) \in \mathbb{R}^p$$

其中

$$\begin{aligned} x(\cdot):&\quad \mathbb{R}^+ = [0,\infty) \to \mathbb{R}^N, \\ u_i(t):&\quad \mathbb{R}^+ \times \mathbb{R}^p \to \mathbb{R}, \\ &\quad (t, y(t)) \mapsto u_i(t, y(t)) \end{aligned}$$

对于这个系统，首先定义MDLg模型的运动基元。

定义3.1（运动基元）：在MDLg模型中，运动基元（atom）是一个三元组符号，用σ_i来表示，其中

$$\begin{aligned} &\sigma_i = \left(U_i\left(t\right), \xi_i, T_i\right), t \in \left[\tau_{i-1}, \tau_i\right] \\ &T_i = \tau_i - \tau_{i-1} \\ &U_i = (u_1, \cdots, u_m)^T \end{aligned} \tag{4}$$

其中U_i是输入控制率，ξ_i是一个触发中断的布尔函数

$$\begin{aligned} &\xi_i : \mathbb{R}^k \to \{0,1\} \\ &s(t) \mapsto \xi(s(t)) \end{aligned}$$

t是表示时间的正整数，而$s(\cdot): [0,T] \to \mathbb{R}^k$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。

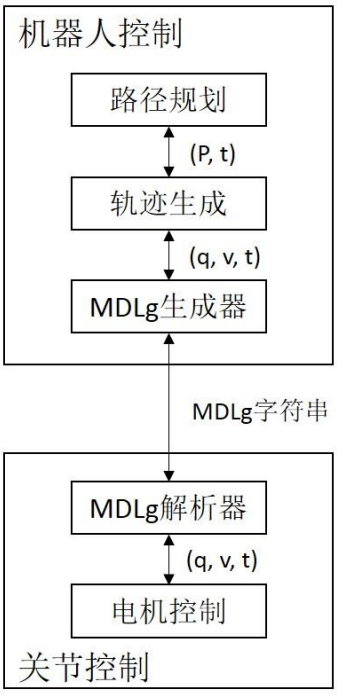

图 1　基于 MDLg 模型的控制器架构

当这一串运动基元被传输到MDLg从端解析器的时候，就可以构成一个运动序列

$$\sigma_1 \cdots \sigma_n = (U_1, \xi_1, T_1) \cdots (U_n, \xi_n, T_n)$$

解析器将这个运动序列重构以后就可以得到原系统的状态方程

$$\begin{aligned} \dot{x} &= f(x) + G(x)U_1,\ \tau_0 \leqslant t < \tau_1 \\ \dot{x} &= f(x) + G(x)U_2,\ \tau_1 \leqslant t < \tau_2 \\ &\vdots \\ \dot{x} &= f(x) + G(x)U_n,\ \tau_{n-1} \leqslant t < \tau_n \end{aligned} \tag{5}$$

定义 3.2（缩放基元）：**缩放基元**是一个进行了标量数乘的运动基元，它对控制信号进行了放大或缩小，对执行时间进行了压缩或扩张

$$\begin{aligned} &(\alpha, \beta)\sigma_i = (\alpha U_i(\beta t), \xi, \frac{T_i}{\beta}), \quad t \in [0, T] \\ &st \quad \beta = \frac{T_i}{T} \end{aligned} \tag{6}$$

等式左边就是一个缩放基元$(\alpha, \beta)\sigma_i$，(α, β)被称为缩放系数。

缩放基元对执行时间不同的运动基元进行时间扩张，使它们的执行时间一致，有利于基元的存储和传输。

定义 3.3（字母表）：字母表用符号Σ表示，是有限个运动基元的集合

$$\Sigma = \{\sigma_1, \sigma_2, \cdots, \sigma_n\}$$

其中对$\forall \sigma_i, \sigma_j \in \Sigma$都满足$\sigma_i \neq (\alpha, \beta)\sigma_j, \alpha \in \mathbb{R}^m, \beta \in \mathbb{R}$。

字母表包含了系统中所有的运动基元。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：

$$(\alpha_1,\beta)\sigma_1 \Vert (\alpha_2,\beta)\sigma_2 = \big(\alpha_1 u_1(\beta t)+\alpha_2 u_2(\beta t)\big), \min\{\xi_1,\xi_2\}$$
$$st \quad T=\frac{T_i}{\beta}$$

在运动基元合并运算的基础上，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Σ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

$$\tilde{\sigma}_i=(\alpha_{i1},\beta_i)\sigma_1 \Vert \cdots \Vert (\alpha_{in},\beta_i)\sigma_n \quad (7)$$

其中n是字母表Σ中基元的个数。另外用符号$\tilde{\rho}_i=(\alpha_{i1},\beta_i)(\alpha_{i2},\beta_i)\cdots(\alpha_{in},\beta_i)$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

$$MDLg=\{\tilde{\rho}_1,\tilde{\rho}_2,\cdots,\tilde{\rho}_m\}$$

如果将一个 MDLg 字符串传输到 MDLg 从端解析器，解析器就得到了一个合并基元序列$\tilde{\sigma}_1\tilde{\sigma}_2\cdots\tilde{\sigma}_m$。解析器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

$$\begin{aligned}\dot{x}&=f(x)+G(x)(\alpha_{11}u_1+\cdots+\alpha_{1n}u_n),\tau_0\leqslant\beta_1 t<\tau_1\\ \dot{x}&=f(x)+G(x)(\alpha_{21}u_1+\cdots+\alpha_{2n}u_n),\tau_1\leqslant\beta_2 t<\tau_2\\ &\vdots\\ \dot{x}&=f(x)+G(x)(\alpha_{m1}u_1+\cdots+\alpha_{mn}u_n),\tau_{m-1}\leqslant\beta_m t<\tau_m\end{aligned}$$

在完成了 MDLg 模型的基本定义之后，就可以运用这个模型构建 MDLg 轨迹生成算法，这将在下一节中展开论述。

4.算法实现

本节将介绍 MDLg 算法的具体实现。MDLg 算法包含两个主要部分：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，这称为**时域分割**；第二步是将已经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的 MDLg 解析器中重构出控制信号的过程。图 2 展示了整个算法的信号流图：

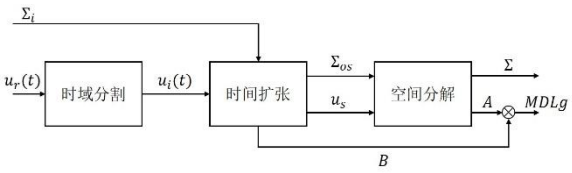

图 2　MDLg 算法信号流图

4.1 时域分割

时域分割是对参考控制信号$u_r(t)$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$u_r(t)$和初始字母表Σ_{init}。给定的控制输入信号是一个在时间域上连续的函数。然而，事实上计算机并不能处理连续信号，要以时间间隔Δt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$u_r(k)\in\mathbb{R}^{1\times d}$。接下来需要在采样信号中找到变化率比较大的点，以完成时域分割。首先计算出输入信号的二阶导数

$$\begin{aligned}\dot{u}_r(k)&=\frac{u_r(k+1)-u_r(k)}{\Delta t},k\in[1,d-1]\\ \ddot{u}_r(k)&=\frac{\dot{u}_r(k+1)-\dot{u}_r(k)}{\Delta t},k\in[1,d-2]\end{aligned} \quad (8)$$

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

$$\begin{aligned}&\Vert \ddot{u}_r(k-1)\Vert>\varepsilon,\Vert \ddot{u}(k)\Vert<\varepsilon\\ \text{or }&\Vert \ddot{u}_r(k-1)\Vert<\varepsilon,\Vert \ddot{u}(k)\Vert>\varepsilon\end{aligned} \quad (9)$$

其中ε是用户给定的控制变量。这一过程将会输出一个矩阵$u_i\in\mathbb{R}^{n\times m}$，它的行数 n 表示分割出的总段数，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，使每一个分割段的执行时间一致。因此，这一过程有两个输入，分别是前面经过分割得到矩阵$u_i\in\mathbb{R}^{n\times m}$和初始字母表$\Sigma_{init}\in\mathbb{R}^{j\times k}$。输出则是完成了时间扩张的分割段矩阵$u_s\in\mathbb{R}^{n\times\max(s,k)}$和$\Sigma_s\in\mathbb{R}^{j\times\max(s,k)}$，以及扩张尺度系数向量$\beta\in\mathbb{R}^n$。扩张以后每一段的时间尺度都变为$T_s=\Delta t\times(max(m,k)-1)$。

首先用下面的方程计算时间扩张系数

$$\beta_{s_i}T_i=T_s$$

其中

$$T_i=(s-1)\cdot\Delta t$$
$$\mathrm{T_s}=(\max(m.k)-1)\cdot\Delta t$$

可以得到

$$\beta_{s_i} = \frac{(\max(m,k)-1)\cdot\Delta t}{(s-1)\cdot\Delta\cdot t} = \frac{\max(m,k)-1}{s-1} \quad (10)$$

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

$$\beta = \begin{bmatrix}\beta_1 \\ \vdots \\ \beta_n\end{bmatrix} \in \mathbb{R}^n \; st \; \beta_i = \frac{1}{\beta_{s_i}}, i \in [1,n] \quad (11)$$

输出的这个向量β用于对扩张后的向量u_s进行压缩，得到分割段的原始执行时间。

4.2 空间分解

时域分割过程完成之后，就要对扩张后的分割段进行空间分解，把它们分解映射到运动基元上。但是这里存在一个的问题，即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$u_s \in \mathbb{R}^{j\times max(m.k)}$和字母表$\Sigma_s \in \mathbb{R}^{j\times max(m,k)}$。输出则是扩展后的新字母表Σ和表示运动序列的 MDLg 字符串。

根据泛函分析理论，参考控制信号$u_r(t)$是L^2空间中的函数。因此可以对这个函数进行正交分解，正交分解的基就是已经得到的字母表Σ_s。根据希尔伯特空间正交分解理论，对每一个分割段$u_s(t)$进行正交分解后，这个函数就可以表示成如下形式：

$$u_s(t) = \alpha_1 u_1(t) + \cdots + \alpha_n u_n(t) \quad (12)$$

即参考信号u_s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$\sigma_i = (u_i, \xi_i, T_i)$。下一步就是计算出线性组合的参数α，根据希尔伯特空间函数内积的性质，可以得到：

$$\begin{aligned}\langle u_s, u_i\rangle &= \langle \alpha_1 u_1 + \alpha_2 u_2 + \cdots + \alpha_n u_n, u_i\rangle \\ &= \langle \alpha_1 u_1, u_i\rangle + \cdots + \langle \alpha_n u_n, u_i\rangle \\ &= \alpha_1\langle u_1, u_i\rangle + \cdots + \alpha_n\langle u_n, u_i\rangle\end{aligned}$$

从u_1到u_n排列成矩阵，可以得到下面的矩阵方程

$$\begin{bmatrix}\langle u_s,u_1\rangle \\ \vdots \\ \langle u_s,u_n\rangle\end{bmatrix} = \begin{bmatrix}\langle u_1,u_1\rangle & \cdots & \langle u_n,u_1\rangle \\ \vdots & \ddots & \vdots \\ \langle u_1,u_n\rangle & \cdots & \langle u_n,u_n\rangle\end{bmatrix}\begin{bmatrix}\alpha_1 \\ \vdots \\ \alpha_n\end{bmatrix}$$

用简化符号表示上面的矩阵方程，令：

$$\mathrm{v} = \begin{bmatrix}\langle u_s,u_1\rangle \\ \vdots \\ \langle u_s,u_n\rangle\end{bmatrix}, \quad \alpha = \begin{bmatrix}\alpha_1 \\ \vdots \\ \alpha_n\end{bmatrix}$$
$$G = \begin{bmatrix}\langle u_1,u_1\rangle & \cdots & \langle u_n,u_1\rangle \\ \vdots & \ddots & \vdots \\ \langle u_1,u_n\rangle & \cdots & \langle u_n,u_n\rangle\end{bmatrix} \quad (13)$$

则方程可以表示为$\mathrm{v} = \mathrm{G}\alpha$，因此参数向量可以表示为

$$\alpha = G^{-1}v \quad (14)$$

如此操作把所有的分割段都进行分解，就得到系数矩阵 A

$$A = \begin{bmatrix}\alpha_1^T \\ \vdots \\ \alpha_n^T\end{bmatrix} \quad (15)$$

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，可以选择一些典型的样条曲线作为运动基元，本文使用了二次、三次、五次、正弦曲线以及傅立叶级数组合曲线作为初始运动基元。第二个问题则需要构建一种扩张运动基元的方式，使对参考信号u_r的分解满足给定的精度要求。

如果参考信号u_r经过初始运动基元分解后，误差不满足给定要求，那么可以令

$$u^* = \alpha_1 u_1 + \alpha_2 u_2 + \cdots + \alpha_n u_n$$
$$st \;\; u_i \in \Sigma_{init}, i \in [1,n]$$

则有

$$u_r = u^* + u^\perp \;\; st \;\; u^\perp \notin \Sigma_{init} \quad (16)$$

这样就得到了新的运动基元σ_{n+1}，其中的控制率为$u_{n+1} = u_r - u^* = u^\perp$。扩展后的字母表是$\Sigma = \Sigma_{init} \oplus \sigma_{n+1}$。这一过程将会被反复执行，直到分解满足给定误差要求为止。

连续函数上的内积一般由如下公式定义

$$\langle u, v\rangle = \int_0^T u(t)v(t)dt \quad (17)$$

但是在实践中，处理的是离散点的内积，需要使用数值内积算法

$$\begin{cases} h(s) = u(s)v(s), s \in [1, max(m,k)] \\ \langle u,v\rangle \approx \frac{dt}{2}\sum_{i=1}^{max(m,k)-1}(h(i+1)+h(i)) \end{cases}$$

整个算法的细节在算法表 1 中展示。

表 1　MDLg 算法

算法：MDLg 字符串产生算法

输入： u_r, Σ_i, Δt

输出： Σ, MDL_g字符串(A, β)

根据输入$u_t(t)$拟合Σ_{init};
分割$u_r(t) \rightarrow \{u_1(t_1), \cdots, u_m(t_m)\}$;
缩放$\{u_1(t_1), \cdots, u_m(t_m)\} \rightarrow \{u_1, \cdots, u_m\}, \Sigma_{\text{init}} \rightarrow \Sigma_s$;
根据方程（10）（11）计算β;
根据方程（13）计算矩阵G;
A = [ ], Σ = Σ_s;
while 映射 **do**
  While !done **do**
    根据方程（13）计算v;
    根据方程（14）计算$\alpha = G^{-1}v$;
    计算$\|u_i\|^2$;
    if $\|u_i\|^2 - \alpha^T G \alpha < \varepsilon$ **then**
      $A = [A \quad \alpha^T]^T$;
      done = TRUE;
    else
      根据方程（13）计算u^T;
      $\Sigma = [\Sigma \quad u^T]^T$;
      根据方程（16）重新计算矩阵G;
      done = FALSE;
    end if;
  end while;
end while;

4.3 信号重构

本过程将在 MDLg 从端解析器中重构参考信号，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Σ和 MDLg 字符串的系数矩阵 A 以及扩张系数向量β。字母表就被预先存储在 MDLg 从端关节中，我们只需要把字符串 A 和扩张系数β实时传输到机器人关节控制器中的 MDLg 解析器，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原始时间。因此，第二步就是应用扩张系数β使所有分割段回归初始的执行时间。然后把所有分割段拼接在一起就得到了初始的参考信号。

所有的合并基元被顺序执行，关节就可以按照参考轨迹运动。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，就会控制关节和机器人停止运动，保证系统安全。

5.仿真研究

为了验证 MDLg 模型的效果，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验证。

本文仿真使用 V-REP 3.4.0 和 Matlab 2016b。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 算法的有效性。

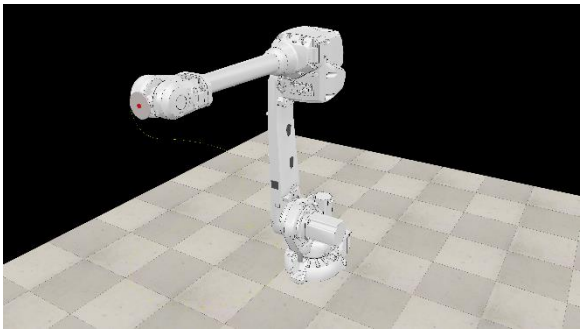

图 3　IRB4600 机械臂模型

本次仿真开启了 V-REP 的实时仿真模式（real-time mode），应用 MDLg 轨迹生成算法，机械臂实现了末端的圆形轨迹运动。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[1]。

[1] 仿真代码及录像
https://github.com/liuzhaoming5954/MDL_paper

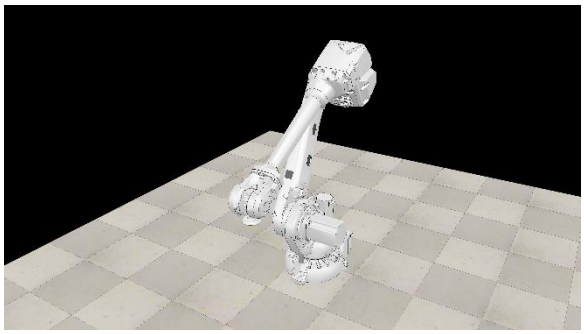

图 4 机械臂末端的圆形轨迹

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 4 所示的圆周，需要转动 1、2、3、5 轴，4 轴不需要运动也不考虑 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到 100 个，与传统方式相比数量大幅减少。

表 2 MDLg 与传统方法的数据传输量

	方法	
	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，机器人控制系统恰好具有这种特征。使用本文提出的 MDLg 模型和轨迹生成算法，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械臂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，还需要在真实的机械臂上进行实验研究。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 算法解析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, 负超. 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.

作者简介：

刘钊铭（1989-），男（汉族），辽宁沈阳人，博士研究生，主要研究领域为机器人动力学建模与控制，机器人关节驱动器，实时以太网通信。

刘乃龙（1989-），男（汉族），山东临沂人，博士研究生，主要研究领域为机器人操作，人工智能，深度强化学习。

魏 青（1988-），男（汉族），河南夏邑人，博士研究生，主要研究领域为遥操作机器人技术，虚拟现实技术。

崔 龙（1980-），男（汉族），辽宁沈阳人，博士，副研究员，硕士研究生导师，主要研究领域为机器人及遥操作技术，智能机构与振动技

术，先进工业机器人及其控制。

联系方式：

第一作者	导师/通讯作者
姓名 刘钊铭 手机 18698868270 邮箱 liuzhaoming@sia.cn	姓名 崔龙 手机 18604045616 邮箱 cuilong@sia.cn
详细通信地址	邮编
辽宁省沈阳市沈河区南塔街114号中国科学院沈阳自动化研究所	110016