

基于凸规划的机械臂轨迹规划方法<sup>①</sup>

赵建军<sup>②\*\*\*</sup> 魏 毅<sup>\*</sup> 朱登明<sup>\*</sup> 夏时洪<sup>\*</sup> 王兆其<sup>\*</sup>

(<sup>\*</sup>中国科学院计算技术研究所前瞻研究实验室,移动计算与新型终端北京市重点实验室 北京 100190)

(<sup>\*\*</sup>中国科学院大学 北京 100049)

**摘 要** 研究了快速求解具有时间约束的机械臂轨迹规划问题,提出了一种基于凸规划的轨迹规划方法。该方法针对机械臂轨迹规划中动力学约束非线性强、时间约束不易处理的问题,首先通过变量替换,将非线性约束转化为线性约束,然后添加新的约束,将原始非凸优化问题转化为凸规划问题,在此基础上,将其写作二阶锥规划(SOCP)形式,使用SeDuMi等优化工具包近似实时求解。该方法具有以下优点:计算高效,凸规划问题能够在多项式时间内得到求解;算法全局稳定,能收敛到全局最优解,不需要提供优化初值;可扩展性强,工业机器人的多种约束以及性能指标如加速度平滑约束、功率等均可扩充。仿真实验表明,与现有方法相比,该方法能够有效提高轨迹规划的效率,机器人的轨迹规划可以近似实时求解。

**关键词** 轨迹规划,时间约束,凸规划,二阶锥规划(SOCP)

0 引 言

机械臂轨迹规划是机器人学领域中的重要研究课题。机械臂轨迹取决于机械臂在运动过程中的位移、速度和加速度,而轨迹规划则是指根据作业任务要求,计算出关节空间中或者笛卡尔空间中满足机械臂运动学与动力学约束的运动轨迹,目的是使机械臂末端平稳地完成一定的作业。机械臂是一个高度非线性、多变量耦合的动力学系统,而轨迹规划需要同时考虑避障、运动空间约束以及运动学与动力学约束,要同时处理所有约束非常困难<sup>[1,2]</sup>。一种简化轨迹规划问题的策略是解耦的方法<sup>[3]</sup>。现有大多数解耦方法都是基于Hollerbach和Shin的工作<sup>[4,5]</sup>,将原始规划问题解耦为两步:(1)根据用户输入的路径约束点规划一条几何路径,考虑回避障碍、运动空间<sup>[5]</sup>; (2)考虑运动学与动力学约束,在状态空间 $(s, \dot{s})$ ( $s, \dot{s}$ 分别代表既定路径的弧长参数与导数)中计算一个沿着既定路径的时间函数,使得机械臂在满足关节角速度、角加速度、关节力矩约

束的同时,完成规划任务<sup>[6,7]</sup>。解耦方法通过将几何约束与动力学约束分开考虑,这在一定程度上简化了轨迹规划问题。本文不考虑避障、运动空间等几何约束,重点处理第(2)中的运动学约束与动力学约束。

轨迹规划中处理运动学约束与动力学约束主要有基于样条曲线的轨迹规划方法,即根据机械手在约束点的位姿、速度和加速度约束,从一类函数(如 $n$ 次多项式、B样条等)中选取参数化轨迹,在关节空间中进行插值,得到满足约束条件的运动轨迹。在考虑了关节速度、加速度、加加速度约束的情况下,这类方法大都使用遗传算法、神经网络等方法求解样条参数<sup>[8-11]</sup>。由于该类算法缺乏完整的收敛性理论证明,无法保证其求解速度与解的全局最优性,现有方法大都需要进行离线轨迹规划。此外,现有文献中只涉及运动学约束,没有处理关节力矩、功率等动力学约束。另外一种方法是动态规划法<sup>[12-14]</sup>。Vukobratovic等在时域中将运动变量离散化,使用动态规划方法求解轨迹规划问题<sup>[12]</sup>。其对应时域做均匀划分可能导致运动数据尤其是加速度变化剧烈,

① 国家自然科学基金(61173067, 61272322), 863 计划(2007AA01Z320, 2012AA011501)和十二五国家科技支撑课题(2013BAK03B07)资助项目。  
② 男,1983 年生,博士生;研究方向:虚拟现实,人体运动合成,机械臂轨迹规划;联系人, E-mail: zhaojianjun@ict.ac.cn (收稿日期:2014-06-16)

从而导致动力学方程难以满足。Duleba 则通过拉格朗日乘子将时间约束加入到优化目标函数中,使用动态规划在状态空间中求解该问题<sup>[13]</sup>。将原始问题转化为以拉格朗日乘子为参数的递归系列子问题后,采用预测-校正的策略调整拉格朗日乘子得到满足时间约束的解,其算法效率很大程度上取决于初值选取的好坏。本文提出了一种基于凸规划的直接法,用以求解具有时间约束的轨迹规划问题。具有时间约束的轨迹规划问题是一个复杂的非线性优化问题,通过变量替换以及添加新的约束,我们将时间约束的机械臂轨迹规划问题转化为一个凸规划问题。进一步,我们将得到的凸规划问题改写为二阶锥规划(second-order cone program, SOCP)形式,从而能够使用 SeDuMi 等计算工具包进行快速求解<sup>[15]</sup>。此算法继承了凸规划的以下优点:凸规划得到的优化解一定为全局最优解;凸规划求解速度快,能够近似实时求解;无需设置初值与松弛因子等。仿真实验结果表明,与现有的方法相比,本文方法有效提高了具有动力学约束与时间约束的轨迹规划求解效率,能够近似实时地完成规划任务。另外本文提出的方法具有很强的扩展性,工业机器人作业中涉及的大多数性能指标,如加速度平滑、功率等,都能够转化为凸约束或凸目标函数,使用凸规划处理。

## 1 具有时间约束的轨迹规划问题

大多数机器人作业中,用户需约束机械臂在起始点与末端点以及中间关键路径点的位姿,而在弧焊、曲面加工、摄像控制中,用户不仅规定机械臂的起始点、末端点,而且约束机械臂末端沿着特定的路径运动。以上第一种形式的约束可以通过样条或者多项式拟合得到第二种形式的末端轨迹约束。不失一般性,具有时间约束的轨迹规划描述是:在限定关节力矩、关节角度、关节角速度、关节角加速度等参数变化范围的情况下,计算合适的关节力矩,使机器人机械臂在关节力矩的驱动下准确地沿给定的轨迹平稳运动且在指定的时间通过指定的位置(图 1)。

### 1.1 轨迹规划问题性能目标

机械臂规划的末端轨迹表示为  $\mathbf{q}(s)$ , 规划目标为机械臂末端在时刻  $T_l$  到达  $\mathbf{q}(s(T_l)) = \mathbf{q}(s_l) = \mathbf{q}_l, l = 0, \dots, L$ 。其中,  $\mathbf{q} \in \mathbf{R}^n$  表示机械臂的  $n$  个关节角度,  $s$  为机械臂末端轨迹曲线的弧长参数。

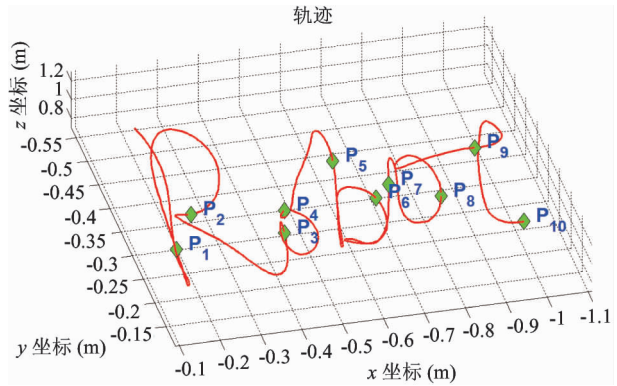


图 1 具有时间约束的轨迹规划问题

这种具有时间约束的轨迹规划问题的目标是使得机械臂末端在  $L$  个轨迹点处的时间误差最小,其目标函数为

$$\min_{s(t)} \sum_{l=0}^L \|t_l - T_l\| \Leftrightarrow \min_{s(t)} \sum_{l=0}^L \|(t_l - t_{l-1}) - (T_l - T_{l-1})\| \quad (1)$$

其中  $t_l$  为机械臂末端运行到  $q_l$  的时间。

关节空间中表示的机械臂末端轨迹  $\mathbf{q}(s)$  可以描述为弧长参数  $s$  的函数。轨迹的弧长参数描述了末端轨迹的几何特征,而末端轨迹的时间信息由弧长参数与时间的函数  $s(t)$  来确定。不失一般性,我们假设机械臂末端轨迹弧长参数的取值范围为  $[0, 1]$ 。在本文中,弧长参数  $s$  与时间  $t$  的关系不再注明。

### 1.2 机器人动力学方程约束

本文轨迹规划基于一台 KUKA-110 机器人(见图 2),该机械臂具有 6 个自由度,其多刚体系统动力学方程可表示如下:

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}^2 + \mathbf{F}_s(\mathbf{q}) \operatorname{sgn}(\dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) \quad (2)$$



图 2 KUKA-110 机器人

其中,  $\mathbf{q} \in \mathbf{R}^n$  表示  $n$  个关节角度,  $\dot{\mathbf{q}}$  表示关节角速度,  $\ddot{\mathbf{q}}$  表示关节角加速度,  $\boldsymbol{\tau} \in \mathbf{R}^n$  表示关节力矩,  $\mathbf{M}(\mathbf{q}) \in \mathbf{R}^{n \times n}$  为质量矩阵,  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbf{R}^{n \times n}$  表示离心力与科里奥利力,  $\mathbf{F}_s(\mathbf{q}) \in \mathbf{R}^{n \times n}$  表示哥伦布摩擦力,  $\mathbf{G}(\mathbf{q}) \in \mathbf{R}^n$  表示重力。实际应用中, 摩擦力可忽略不计, 故在本文中忽略该项。本文实验中,  $n=6$ 。

由微积分中的链式法则可知, 给定机械臂末端轨迹, 那么其关节角速度与关节角加速度可以表示为

$$\begin{aligned}\dot{\mathbf{q}}(s) &= \mathbf{q}'(s)\dot{s} \\ \ddot{\mathbf{q}}(s) &= \mathbf{q}'(s)\ddot{s} + \mathbf{q}''(s)\dot{s}^2\end{aligned}\quad (3)$$

其中  $\dot{s} = ds/dt$ ,  $\ddot{s} = d^2s/dt^2$ ,  $\mathbf{q}'(s) = \partial \mathbf{q}(s)/\partial s$ ,  $\mathbf{q}''(s) = \partial^2 \mathbf{q}(s)/\partial s^2$ 。

将式(3)代入式(2)中得到弧长参数表示的动力学约束如下:

$$\begin{aligned}\boldsymbol{\tau}(s) &= \mathbf{m}(s)\ddot{s} + \mathbf{c}(s)\dot{s}^2 + \mathbf{g}(s) \\ \mathbf{m}(s) &= \mathbf{M}(\mathbf{q}(s))\mathbf{q}'(s) \\ \mathbf{c}(s) &= \mathbf{M}(\mathbf{q}(s))\mathbf{q}''(s) + \mathbf{C}(\mathbf{q}(s), \mathbf{q}'(s))\mathbf{q}'(s) \\ \mathbf{g}(s) &= \mathbf{F}_s(\mathbf{q}(s))\text{sgn}(\mathbf{q}'(s)) + \mathbf{G}(\mathbf{q}(s))\end{aligned}\quad (4)$$

在上式中, 由于在轨迹运行过程中可以假设  $\dot{s} > 0$ , 故  $\text{sgn}(\dot{\mathbf{q}}(s))$  等价替换为  $\text{sgn}(\mathbf{q}'(s))$ 。

### 1.3 运动学约束与动力学约束

不失一般性, 轨迹规划问题分别考虑如下动力学与运动学约束:

$$\underline{\boldsymbol{\tau}}(s) \leq \boldsymbol{\tau}(s) \leq \bar{\boldsymbol{\tau}}(s) \quad (6)$$

$$(\dot{\mathbf{q}}_i(s))^2 \leq (\bar{\dot{\mathbf{q}}}_i(s))^2 \quad (7)$$

$$\ddot{\mathbf{q}}_i(s) \leq \ddot{\mathbf{q}}_i(s) \leq \bar{\ddot{\mathbf{q}}}_i(s) \quad (8)$$

$$s(0) = 0$$

$$s(T) = 1$$

$$\dot{s}(0) = \dot{s}_T^2$$

$$\dot{s}(T) = \dot{s}_T^2$$

其中式(6)表示关节力矩约束, 式(7)表示关节角度范围约束, 式(8)表示关节角加速度范围约束, 式(9)表示初始点与末端点的状态约束。

在运动规划中, 约束机械臂末端沿着给定轨迹始终向前运动:

$$\dot{s}(t) \geq 0 \quad (10)$$

### 1.4 轨迹规划的数学描述

综合上述目标函数(式(1))、动力学方程约束(式(4)、(5))、动力学约束(式(6))、运动学约束(式(7)、(8))、边界约束(式(9))以及末端轨迹运动约束(式(10)), 具有时间约束的轨迹规划问题所

对应的数学描述如下:

$$\begin{aligned}\min_{s(t)} \quad & \sum_{l=0}^L \| (t_l - t_{l-1}) - (T_l - T_{l-1}) \| \\ \text{s. t.} \quad & \boldsymbol{\tau}(s) = \mathbf{m}(s)\ddot{s} + \mathbf{c}(s)\dot{s}^2 + \mathbf{g}(s) \\ & s(0) = 0 \\ & s(T) = 1 \\ & \dot{s}(0) = \dot{s}_T^2 \\ & \dot{s}(T) = \dot{s}_T^2 \\ & \dot{s}(t) \geq 0 \\ & \underline{\boldsymbol{\tau}}(s) \leq \boldsymbol{\tau}(s) \leq \bar{\boldsymbol{\tau}}(s) \\ & (\dot{\mathbf{q}}_i(s))^2 \leq (\bar{\dot{\mathbf{q}}}_i(s))^2 \\ & \ddot{\mathbf{q}}_i(s) \leq \ddot{\mathbf{q}}_i(s) \leq \bar{\ddot{\mathbf{q}}}_i(s) \\ & \text{for } t \in [0, T]\end{aligned}\quad (11)$$

其中,  $\underline{\boldsymbol{\tau}}(s(t))$ 、 $\bar{\boldsymbol{\tau}}(s(t))$  分别表示关节力矩允许的最小值和最大值,  $(\bar{\dot{\mathbf{q}}}_i(s))^2$  表示关节角速度平方允许的最大值,  $\ddot{\mathbf{q}}_i(s)$ 、 $\bar{\ddot{\mathbf{q}}}_i(s)$  分别表示关节角加速度允许的最小值和最大值。

## 2 基于凸规划的轨迹规划方法

轨迹规划的原始问题(式(11))是一个非线性非凸的优化问题, 不易求解。我们通过变量替换以及添加新的约束, 将之转化为凸规划, 然后使用直接法将优化问题离散化, 表示为数值形式的凸规划形式。

### 2.1 原始问题转化为凸规划问题

本小节通过非线性等量代换和一系列公式变换, 将原始优化模型中的目标函数与约束函数进行转换, 从而将原始问题转化为凸规划问题。

#### (1) 目标函数

由于原始问题中目标函数中  $t_l$  与  $s$  的关系并不是简单的初等函数关系, 故引入新的优化变量  $a(s) \equiv \dot{s}$ ,  $b(s) \equiv \ddot{s}$  ( $\dot{s}$  为  $s$  的一阶导数,  $\ddot{s}$  为  $s$  的二阶导数) 并将目标函数中的积分变量由  $t$  变为  $s$ , 则原始优化模型的目标函数变为

$$\begin{aligned}\sum_{l=0}^L \| (t_l - t_{l-1}) - (T_l - T_{l-1}) \| \\ = \sum_{l=0}^L \| \int_{t_{l-1}}^{t_l} 1 dt - (T_l - T_{l-1}) \| \\ = \sum_{l=0}^L \| \int_{s_{l-1}}^{s_l} \frac{1}{\dot{s}} ds - (T_l - T_{l-1}) \| \\ = \sum_{l=0}^L \| \int_{s_{l-1}}^{s_l} \frac{1}{b(s)} ds - (T_l - T_{l-1}) \| \end{aligned}\quad (12)$$

(2) 关节角速度约束

由于  $(\dot{\mathbf{q}}_i(s))^2 = (\dot{\mathbf{q}}_i'(s)\dot{s})^2 = (\dot{\mathbf{q}}_i'(s))^2 b(s) \leq (\bar{\mathbf{q}}_i'(s))^2$ , 故可以将关节速度约束转换为

$$b(s) \leq \bar{b}(s) \tag{13}$$

其中  $\bar{b}(s)$  为与  $t$  无关的值。

(3) 关节角加速度约束

由于  $\ddot{\mathbf{q}}_i(s) \leq \ddot{\mathbf{q}}_i(s) \leq \bar{\ddot{\mathbf{q}}}_i(s) \Leftrightarrow \ddot{\mathbf{q}}_i(s) \leq \mathbf{q}_i'(s)\ddot{s} + \mathbf{q}_i''(s)s^2 \leq \bar{\ddot{\mathbf{q}}}_i(s) \Leftrightarrow \ddot{\mathbf{q}}_i(s) \leq \mathbf{q}_i'(s)a(s) + \mathbf{q}_i''(s)b(s) \leq \bar{\ddot{\mathbf{q}}}_i(s)$ , 故可以将关节角加速度的约束转换为如下形式:

$$\underline{f}(s) \leq f(s)a(s) + h(s)b(s) \leq \bar{f}(s) \tag{14}$$

其中  $\underline{f}(s), f(s), h(s), \bar{f}(s)$  为与  $t$  无关的值, 其取值与末端轨迹类型有关。

(4) 添加新约束

新引入的优化变量  $a(s), b(s)$  之间具有微分关系  $a(s) \equiv \dot{s}, b(s) \equiv \dot{s}^2$ , 由于  $\dot{b}(s) = b'(s)s$  以及  $\dot{b}(s) = d(s^2)/dt = 2\dot{s}\dot{s} = 2a(s)s$ , 故需要添加新的约束

$$b'(s) = 2a(s) \tag{15}$$

这里需要指出的是此时优化模型的目标函数(式(11))是一个复杂的非线性函数, 现有优化方法对该优化模型只能求得局部最优解且计算耗时。

为此, 我们通过添加一个新的约束, 将目标函数转化为一个凸目标函数

$$\begin{aligned} \min_{a(s), b(s), \tau(s)} \sum_{l=0}^L \left\| \int_{s_{l-1}}^{s_l} \frac{1}{\sqrt{b(s)}} ds - (T_l - T_{l-1}) \right\| \\ \Rightarrow \begin{cases} \min_{a(s), b(s), \tau(s)} \int_{s_0}^{s_L} b(s) ds \\ \text{s. t. } \int_{s_{l-1}}^{s_l} \frac{1}{\sqrt{b(s)}} ds - (T_l - T_{l-1}) \leq 0 \end{cases} \end{aligned} \tag{16}$$

从而将原始问题转变成为一个凸规划问题, 则原始轨迹规划问题描述为新的凸规划模型如下:

$$\begin{aligned} \min_{a(s), b(s), \tau(s)} \int_{s_0}^{s_L} b(s) ds \\ \text{s. t. } \tau(s) = \mathbf{m}(s)a(s) + \mathbf{c}(s)b(s) + \mathbf{g}(s) \\ b(0) = \dot{s}_0^2 \\ b(1) = \dot{s}_T^2 \\ b'(s) = 2a(s) \\ b(s) \geq 0 \end{aligned}$$

$$b(s) \leq \bar{b}(s)$$

$$\underline{\tau}(s) \leq \tau(s) \leq \bar{\tau}(s)$$

$$\underline{f}(s) \leq f(s)a(s) + h(s)b(s) \leq \bar{f}(s)$$

$$\int_{s_{l-1}}^{s_l} \frac{1}{\sqrt{b(s)}} ds - (T_l - T_{l-1}) \leq 0, l = 0, \dots, L \tag{17}$$

2.2 凸规划问题离散化为数值形式模型

将弧长参数  $s$  在  $[0, 1]$  上进行均匀采样, 得到  $K+1$  个离散点:  $s^0 = 0 \leq s^k \leq 1 = s^K, k = 0, \dots, K$ , 同时原凸规划的优化变量  $a(s), b(s), \tau(s)$  也取此  $K+1$  个离散点上的值作为优化变量。由于机械臂在力矩驱动下进行运动, 所以假设  $a(s)$  在  $[s^{k-1}, s^k]$  上为常值(图 3)。另外由于  $b'(s) = 2a(s)$ , 故假设  $b(s)$  在  $[s^{k-1}, s^k]$  上为线性函数:

$$\begin{aligned} b(s) = b^k + ((b^{k+1} - b^k)/(s^{k+1} - s^k))(s - s^k) \\ s \in [s^k, s^{k+1}] \end{aligned} \tag{18}$$

将式(18)代入式(16)中的约束函数进行解析积分运算, 得到:

$$\sum_{k=k_{l-1}}^{k_l} \frac{2(s^k - s^{k+1})}{\sqrt{b^{k+1}} + \sqrt{b^k}} \leq (T_l - T_{l-1}), l = 1, \dots, L \tag{19}$$

由式(4)以及  $a(s), b(s)$  在  $[s^{k-1}, s^k]$  上为常值与线性函数, 故  $\tau(s)$  为  $[s^{k-1}, s^k]$  上的非线性函数, 故取中点值为  $[s^{k-1}, s^k]$  的近似值, 即  $\tau^k = \tau(s^{k+1/2})$ 。

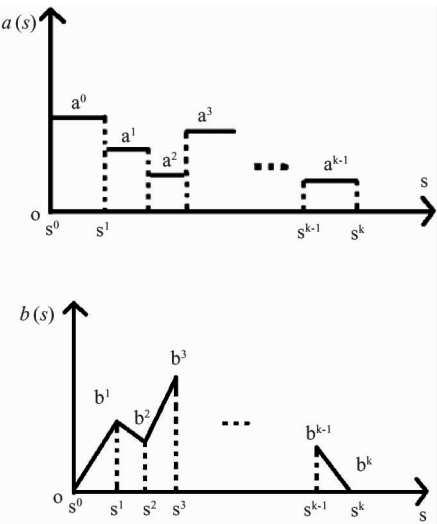


图 3  $a(s), b(s)$  在  $[0, 1]$  的取值

引入  $b^{k+1/2} = (b^k + b^{k+1})/2$ , 则原始凸规划问题离散化为以下形式:

$$\begin{aligned}
& \min_{a^k, b^k, \tau^k} \sum_{k=0}^{K-1} b^k \\
& \text{s. t. } \tau^k = \mathbf{m}(s^{k+1/2})a^k + \mathbf{c}(s^{k+1/2})b^{k+1/2} + \mathbf{g}(s^{k+1/2}) \\
& \quad b^0 = s_0^2 \\
& \quad b^K = s_K^2 \\
& \quad (b^{k+1} - b^k) = 2a^k(s^{k+1} - s^k) \\
& \quad b^k \geq 0, b^K \geq 0 \\
& \quad b^k \leq \dot{b}(s^k), b^K \leq \dot{b}(s^K) \\
& \quad \underline{\tau}(s^{k+1/2}) \leq \tau^k \leq \bar{\tau}(s^{k+1/2}), k = 0, \dots, K-1 \\
& \quad \sum_{k=k_{l-1}}^{k_l} \frac{2(s^{k+1} - s^k)}{\sqrt{b^{k+1}} + \sqrt{b^k}} \leq (T_l - T_{l-1}), l = 1, \dots, L
\end{aligned} \tag{20}$$

### 3 将凸规划转换为二阶锥规划形式

因为二阶锥规划的求解无需提供初值且计算效率高,可使用工具软件 SeDuMi 等求解,本节将凸规划模型(式(20))改写为二阶锥规划(SOCP)模型。

二阶锥规划的标准形式如下:

$$\begin{aligned}
& \min_x \mathbf{f}^T \mathbf{x} \\
& \text{s. t. } \mathbf{F} \mathbf{x} = \mathbf{g} \\
& \quad \|\mathbf{A}_i \mathbf{x} + \mathbf{b}_i\| \leq \mathbf{c}_i^T \mathbf{x} + \mathbf{d}_i, i = 1, \dots, m
\end{aligned} \tag{21}$$

首先,引入新优化变量  $c^k, r^k, k = 0, \dots, K-1$ , 将式(19)改写为以下约束:

$$\sum_{k=k_{l-1}}^{k_l} r^k \leq (T_l - T_{l-1}) \tag{22}$$

$$2(s^{k+1} - s^k)/(c^{k+1} + c^k) \leq r^k \tag{23}$$

$$c^k \leq \sqrt{b^k} \tag{24}$$

式(23)和式(24)能够改写为两个标准的二阶锥约束如下:

$$\left\| \frac{2\sqrt{2(s^{k+1} - s^k)}}{c^{k+1} + c^k - r^k} \right\| \leq c^{k+1} + c^k + r^k \tag{25}$$

$$k = 0, \dots, K-1$$

$$\left\| \frac{2c^k}{b^k - 1} \right\| \leq b^k + 1, k = 0, \dots, K \tag{26}$$

此时,得到一个标准形式的二阶锥规划如下:

$$\begin{aligned}
& \min_{a^k, b^k, c^k, r^k, \tau^k} \sum_{k=0}^{K-1} (b^k) \\
& \text{s. t. } \tau^k = \mathbf{m}(s^{k+1/2})a^k + \mathbf{c}(s^{k+1/2})b^{k+1/2} + \mathbf{g}(s^{k+1/2}) \\
& \quad b^0 = s_0^2, b^K = s_K^2 \\
& \quad (b^{k+1} - b^k) = 2a^k(s^{k+1} - s^k) \\
& \quad \underline{\tau}(s^{k+1/2}) \leq \tau^k \leq \bar{\tau}(s^{k+1/2}), k = 0, \dots, K-1
\end{aligned}$$

$$\left\| \frac{2\sqrt{2(s^{k+1} - s^k)}}{c^{k+1} + c^k - r^k} \right\| \leq c^{k+1} + c^k + r^k$$

$$\left\| \frac{2c^k}{b^k - 1} \right\| \leq b^k + 1 \tag{27}$$

$$b^k \geq 0$$

$$b^k \leq \dot{b}(s^k), k = 0, \dots, K$$

$$\sum_{k=k_{l-1}}^{k_l} r^k \leq (T_l - T_{l-1}), l = 0, \dots, L-1$$

上式中的  $a^k, \tau^k, r^k$  定义在  $k=0, 1, \dots, K-1$  共  $K$  个点上,  $b^k, c^k$  定义在  $k=0, 1, \dots, K$  共  $K+1$  个点上。

### 4 仿真实验

我们在一台 CPU 双核 2.4GHz, 内存 4G 的 PC 机进行仿真实验。首先使用该算法求解 6 自由度机器人(kuka-110 的 D-H 参数, 见表 1)具有时间约束的轨迹规划问题, 在 4.1 中展示了相关结果。此外, 就算法理论与数值计算效率两方面, 我们在 4.2 中将本文算法与其他方法进行了对比。

表 1 KUKA110 机器人的 D-H 坐标参数

| 关节 | $\alpha(^{\circ})$ | $a(\text{cm})$ | $d(\text{cm})$ | $\theta(^{\circ})$ | 运动范围( $^{\circ}$ ) |
|----|--------------------|----------------|----------------|--------------------|--------------------|
| 1  | 90                 | 35             | 0              | 0                  | $[-185, 185]$      |
| 2  | 0                  | 125            | 0              | 90                 | $[-146, 0]$        |
| 3  | -90                | 5.5            | -150           | 0                  | $[-119, 155]$      |
| 4  | 90                 | 0              | 0              | 0                  | $[-350, 350]$      |
| 5  | -90                | 0              | -80.6          | 0                  | $[-125, 125]$      |
| 6  | -90                | 0              | 0              | 90                 | $[-350, 350]$      |

#### 4.1 轨迹规划数值仿真实验

机器人末端的目标轨迹如图 1 所示, 在实验中, 我们设定  $L = 10$ 。末端到达 10 个关键位姿的时刻与弧长参数  $T_l, s_l, l = 0, \dots, 10$  的取值见表 2。我们将目标轨迹离散化为 1000 个点后进行计算, 其优化求解耗时为 71.7449s。使用计算得到的关节力矩驱动机器人运动, 其运行轨迹如图 4 所示, 弧长参数与时间的对应关系见图 5。机械臂末端到达约束点的时刻, 以及相邻两个约束点之间的规划时间误差参见表 2。

机器人运行过程中, 各关节力矩的取值见图 6。从图中可以看出在整个运行过程中, 各关节力矩的取值均在虚线限制的允许范围之内, 这说明在优化求解中, 各关节力矩的约束都得到了满足。另外从

表 2  $T_l, s_l$  的取值以及规划时间误差

| $l$ | $s_l$ | $T_l$ (s) | 误差 ( $\mu$ s) |
|-----|-------|-----------|---------------|
| 1   | 0.1   | 1.2       | 0.16          |
| 2   | 0.2   | 2.3       | 0.13          |
| 3   | 0.3   | 3.5       | 0.13          |
| 4   | 0.4   | 4.5       | 0.12          |
| 5   | 0.5   | 5.5       | 0.09          |
| 6   | 0.6   | 6.7       | 0.14          |
| 7   | 0.7   | 11.2      | 2.80          |
| 8   | 0.8   | 16.4      | 3.25          |
| 9   | 0.9   | 20.7      | 2.43          |
| 10  | 1.0   | 26.7      | 6.52          |

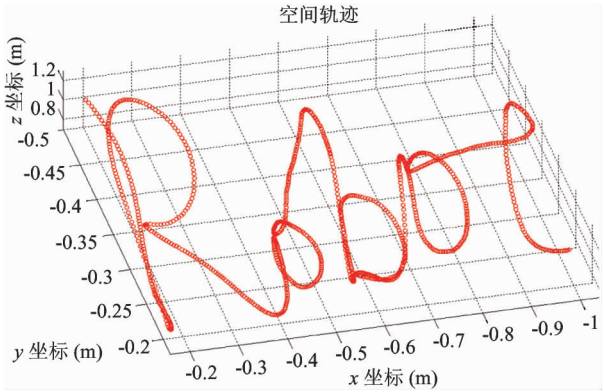


图 4 末端跟踪结果

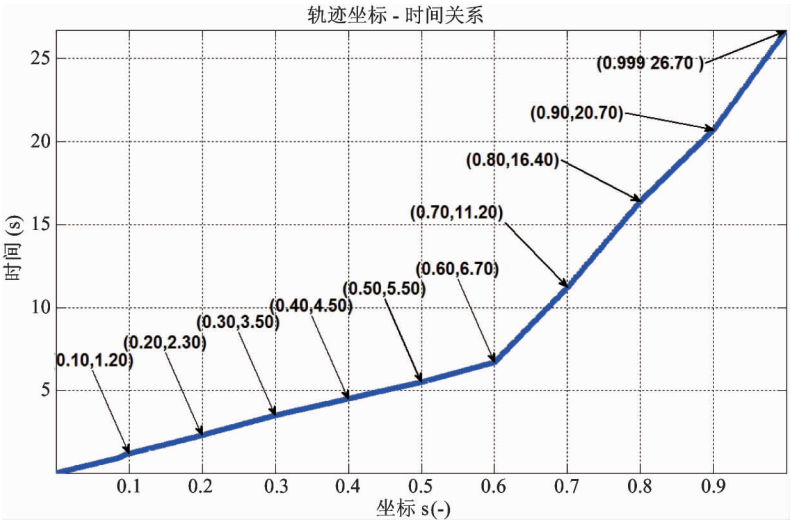


图 5 轨迹-时间关系

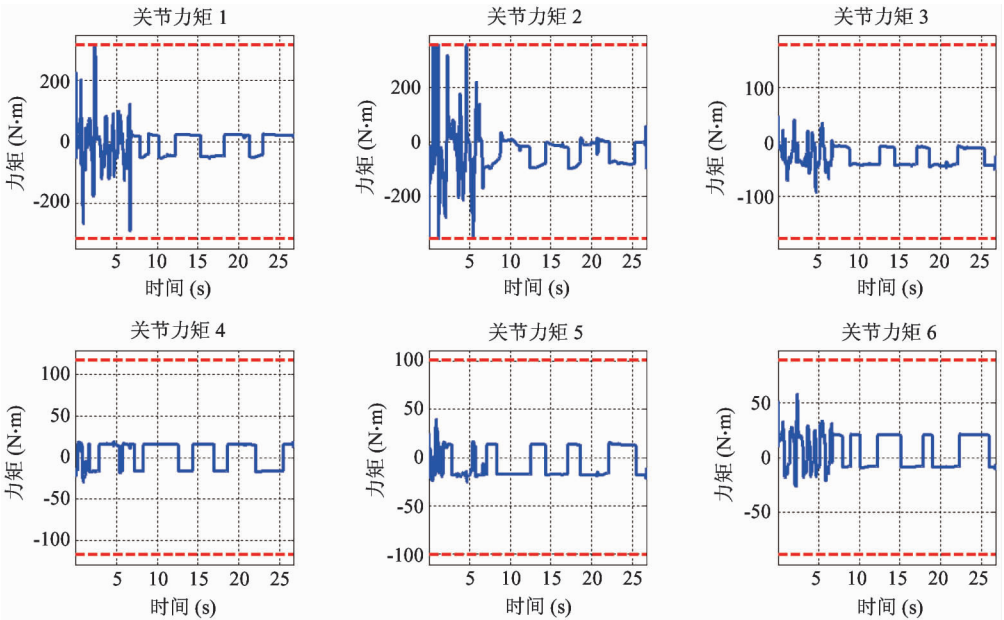


图 6 关节力矩

图中可以看出在机器人运行的前半段时间,各关节力矩比较大,甚至在某些时刻达到了极限值,这是因为机器人在前半段运行较快,而在后半段运行较慢。

实验中, $K$ 的取值对于实验结果的精度以及求解速度有较大影响, $K$ 取值越大,轨迹规划的精度越高,但是其求解也越耗时,在轨迹规划中需要妥善选择。当 $K$ 取100、300、500、1000、1500时,轨迹规划问题的求解时间见表3。从表中可以看出当 $K$ 增加时,其求解时间呈非线性的加速增长趋势。此外,当 $K$ 取值较小时,末端轨迹的跟踪曲线将丢失很多细节信息,与原始曲线相差较大。如 $K=100$ 时,其末端跟踪轨迹见图7,从放大的部分可以看出,由于采样点数量较少,生成轨迹与目标轨迹差异较大。将图4与图7对比,可以看出 $K$ 取值越大,轨迹上采样点越多,整个轨迹的跟踪误差越小。

表3 不同  $K$  取值对应的求解时间

| $K$     | 100   | 300   | 500    | 1000   | 1500    |
|---------|-------|-------|--------|--------|---------|
| 求解时间(s) | 0.887 | 3.228 | 10.452 | 71.745 | 208.937 |

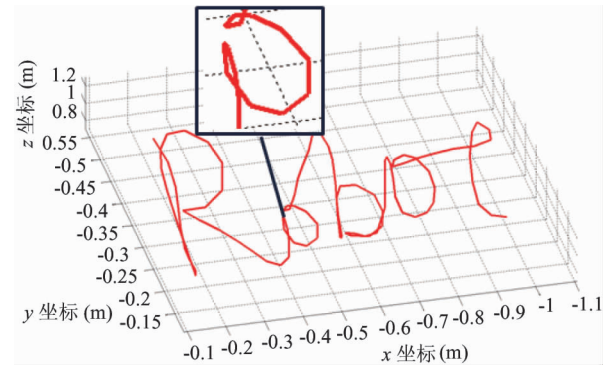


图7  $K=100$  时末端跟踪轨迹

我们将该算法跟踪其他类型的末端轨迹曲线,都得到了类似的结果(图8)。在这两组实验中, $K=1000$ 。Matlab 编辑运行环境见图9所示。

4.2 与其他方法对比实验

与基于样条曲线的神经网络、遗传算法等相比,本文方法有以下优点:(1)能够同时处理动力学约束和运动学约束,而现有的遗传算法以及神经网络方法<sup>[8-11]</sup>无法处理动力学约束;(2)扩展性更强,工业机器人作业中经常涉及的大多数性能指标能够添加到凸规划优化模型中;(3)只涉及离散化参数 $k$ ,而神经网络、遗传算法等需要谨慎设置优化参数,如精度、种群数量、停止条件等。

| 末端轨迹 | 规划时间<br>误差 (ms) | 求解时间<br>(ms) |
|------|-----------------|--------------|
|      | 3.4             | 670.904      |
|      | 2.1             |              |
|      | 4.6             |              |
|      | 3.8             |              |
|      | 7.2             |              |
|      | 9.7             |              |
|      | 2.6             |              |
|      | 5.7             |              |
|      | 3.7             |              |
|      | 3.1             |              |
|      | 5.6             | 652.769      |
|      | 4.5             |              |
|      | 8.6             |              |
|      | 6.2             |              |
|      | 2.2             |              |
|      | 6.3             |              |
|      | 9.1             |              |
|      | 11.3            |              |
|      | 9.4             |              |
|      | 5.3             |              |

图8 其它两组轨迹规划结果(规划时间误差为相邻两个约束点规划后运行时间与目标运行时间之间的差异)

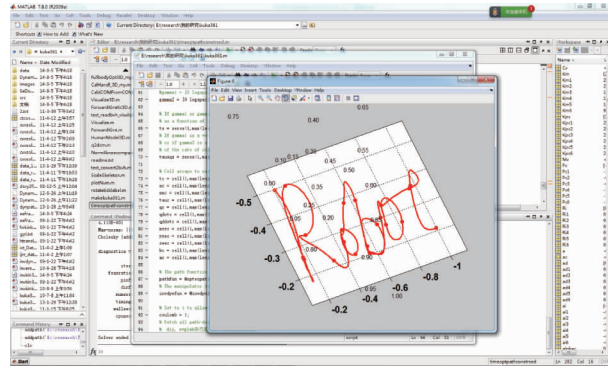


图9 Matlab 仿真实验环境

与动态规划方法相比,本文方法使用的凸规划模型,其收敛性理论能够保证在多项式时间内求得全局最优解。我们对4.1节中轨迹规划问题使用动态规划方法进行求解,并与本文方法进行对比。为了对比算法效率,我们在对比实验中只考虑速度约束,而不考虑其他运动学与动力学约束。在动态规划中,我们设 $L=1$ ,将空间参数( $s$ )离散化为 $k$ 个取值,速度约束 $[-0.4,0.4]$ (m/s)离散化为 $j$ 个区间。本文方法与动态规划的计算效率见表4。从表中可以看出,本文算法计算效率比动态规划算法快一个量级以上。这是由于动态规划法基于预测校正的策略,需要不停地调整拉格朗日因子,以得到满足时间约束的解。从对比实验中可以看出,本文算法能够近似实时求解轨迹规划,在需要频繁进行轨迹

规划应用场景中具有更强的实用性。

表 4 本文算法与动态规划计算时间对比

| $k$  | 本文算法<br>计算时间(s) | $j$ | 动态规划法<br>计算时间(s) |
|------|-----------------|-----|------------------|
| 300  | 12.352          | 30  | 60.521           |
|      |                 | 50  | 86.354           |
|      |                 | 100 | 325.961          |
|      |                 | 200 | 635.365          |
|      |                 | 300 | 1244.975         |
| 600  | 27.861          | 30  | 118.449          |
|      |                 | 50  | 128.268          |
|      |                 | 100 | 586.821          |
|      |                 | 200 | 725.514          |
|      |                 | 300 | 1770.057         |
| 1000 | 38.649          | 30  | 241.612          |
|      |                 | 50  | 359.188          |
|      |                 | 100 | 1256.821         |
|      |                 | 200 | 2055.857         |
|      |                 | 300 | 4041.064         |

5 结 论

本文提出一种基于凸规划的轨迹规划方法,解决了传统轨迹规划方法难以处理动力学约束的问题。首先通过变量替换以及添加新的约束,我们将具有时间约束的机械臂轨迹规划问题转化为一个凸规划问题,然后通过直接法将凸规划问题写为二阶锥规划(SOCP)形式,该优化问题可在多项式时间内求得全局最优解。与现有算法相比,该算法有以下几个优点:优化模型的局部最优解即为全局最优解;可扩展性强,工业机器人的大多数性能指标可以添加到凸规划中;算法高效,可近似实时求解。从仿真实验结果中可以看出,本文算法可在半分钟之内求解一般规模的轨迹规划问题,与现有的动态规划方法相比,数值求解速度快一个数量级。

参考文献

[ 1 ] Piazzi A, Visioli A. Global minimum-jerk trajectory planning of robot manipulators. *IEEE Transactions on Industrial Electronics*, 2000,47(1):140-149

[ 2 ] Diehl M, Bock H, Diedam H, et al. Fast direct multiple

shooting algorithms for optimal robot control. In: *Proceedings of Symposium on Fast Motions in Biomechanics and Robotics Optimization and Feedback Control. Lecture Notes in Control and Information Sciences*, 2006. 65-94

[ 3 ] Siciliano B, Khatib O. *Springer Handbook of Robotics*. Springer. 2008.119-120

[ 4 ] Hollerbach J. Dynamic scaling of manipulator trajectories. *Journal of Dynamic Systems Measurement and Control-Transactions of the Asme*, 1984 ,106(1):102-106

[ 5 ] Shin K, McKay N. Minimum-time control of robot manipulators with geometric path constraints, *IEEE Trans Autom Contr*, 1985,30(6):531-541

[ 6 ] Donald B, Xavier P, Canny J, et al. Kinodynamic motion planning, *J ACM*, 1993, 40(5):1048-1066

[ 7 ] Verscheure D, Demeulenaere B, Swevers J, et al. Time-Optimal Path Tracking for Robots: A Convex Optimization Approach, *IEEE Transactions on Automatic Control*, 2009,54(10): 2318-2327

[ 8 ] Kano H, Fujioka H, Martin C. Optimal smoothing and interpolating splines with constraints. In: *Proceeding of the 46th IEEE Conference on Decision and Control*, New Orleans, USA. 2007. 1796-1801

[ 9 ] Cheng T, Lewis F, Abu-Khalaf M. A neural network solution for fixed-final time optimal control of nonlinear systems. *Automatica*, 2007, 43(3): 482-490

[10] Egerstedt M, Martin C. Optimal trajectory planning and smoothing splines, *Automatica*, 2001, 37(7):1057-1064

[11] 刘松国,朱世强,吴文祥,具有运动时间约束的机械手最优平滑轨迹规划. *电机与控制学报*,2009, 13(6): 897-902

[12] Vukobratovic M, Kircansky M. A method of optimal synthesis of manipulator robot trajectories. *Journal of Dynamic Systems, Measurement, and Control*, 1982, 104(2):188-193

[13] Duleba I. Minimum cost, fixed time trajectory planning in robot manipulators: A suboptimal solution. *Robotica*, 1997,15(5): 555-562

[14] Shin K, McKay N. A dynamic programming approach to trajectory planning of robotic manipulators. *IEEE Trans Autom Contr*, 1986,31(6): 491-500

[15] Sturm J. Using SeDuMi: A Matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 1999,11(1-4): 625-653

# A convex optimization approach for path tracking of robot manipulators

Zhao Jianjun<sup>\*\*\*</sup>, Wei Yi<sup>\*</sup>, Zhu Dengming<sup>\*</sup>, Xia Shihong<sup>\*</sup>, Wang Zhaoqi<sup>\*</sup>

(<sup>\*</sup> Beijing Key Laboratory of Mobile Computing and Pervasive Device, Advanced Computing Research Laboratory, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

(<sup>\*\*</sup> University of Chinese Academy of Sciences, Beijing 100049)

## Abstract

The problem of fast solving a robot manipulator's path tracking with time constraints was studied, and a novel path tracking approach based on convex optimization was presented. To overcome the difficulties in dealing with the strong nonlinear dynamic constraints and time constraints in path tracking, the presented approach converts the nonlinear constraints into linear constraints by replacement of variables, and then adds new constraints to convert the original non-convex optimization problem into a convex optimization problem, furthermore, converts it into a second-order cone program (SOCP), and uses the optimization tools such as the SeDuMi to conduct the real-time solving. This approach has several advantages. Firstly, SOCP problems can be solved in polynomial time by the interior-point methods. Secondly, the convex optimization is globally stable and the solution is globally optimal. Besides, there is no need to provide initial values for the optimization. Thirdly, this approach has great flexibility and can be applied to the more complicated circumstances where some other types of constraints and objective functions can be taken into account, such as acceleration constraints, minimum energy objective function and minimum jerk objective function. The simulations on a six-degrees-of-freedom robot manipulator show the better efficiency and effectiveness of the proposed approach.

**Key words:** path tracking, time constraints, convex optimization, second order cone programming (SOCP)