

密级 _____



中国科学院大学

University of Chinese Academy of Sciences

博士学位论文

时间最优轨迹规划高效算法研究

作者姓名: 赵明勇

指导教师: 高小山 研究员

中国科学院数学与系统科学研究院

学位类别: 博士

学科专业: 应用数学

研究所: 中国科学院数学与系统科学研究院

2016年4月


Y3189745

Algorithm Research on Time-Optimal Trajectory Planning

By

Ming-Yong Zhao

A Dissertation Submitted to
The University of Chinese Academy of Sciences
In partial fulfillment of the requirement
For the degree of
Doctor of Applied Mathematics

Institute of Systems Science
Academy of Mathematics and Systems Science
Chinese Academy of Sciences

May, 2016



摘 要

机器人和数控机床的时间最优轨迹规划是机器人和数控机床领域重要的研究内容,对于提高加工/跟踪效率与加工精度具有重要意义。本论文针对该问题,在三种重要的情形分别提出了新的算法,并且通过仿真和实验验证了提出算法的有效性和实用性。

1. 实际运行中,机器人动力学系统存在着不确定性因素,比如建模过程的误差、材料和工作环境引起的误差、加工和制造过程的扰动等。为了在存在不确定因素情形下实现最优时间运动,本文建立了在给定路径和力矩约束条件下的机器人最优轨迹规划随机模型,并提出了一个有效的求解算法。该算法首先把约束概率不等式简化为确定性条件函数。然后再通过合理的假设,把条件函数简化为一系列的线性函数,最终把该问题转化为了一个线性规划问题。提出算法与已有的其他随机方法相比,与最优解更接近,并且可以处理多个随机变量的情形。与已有的其他严格鲁棒方法相比,提出方法也最准确,因为该算法是严格鲁棒问题的完整解。最后通过仿真实验验证了提出算法的有效性和准确性。

2. 在长加工路径轨迹规划中,由于长加工路径限制点相对较多,数控系统硬件限制等因素,工业中常常分段来加工,并且每段使用零初末速度的条件,这造成了效率的损失。本文基于线性规划,提出了一种速度曲线的高效拼接算法,并且对该算法的鲁棒性和复杂度进行了讨论。最后针对二维猫形路径和三维牙齿形路径分别进行了加工,实验证明该算法得到的解与全局最优解非常的接近($< 0.1\%$),但是计算效率提高了两个数量级。

3. 对机器人在多点打孔、点焊任务中的最优轨迹规划问题进行了研究。这个问题与前面最大的不同是路径并不是固定的,只要求过有限个指定点。本文利用三次Hermite插值多项式以及经过改进的旅行商模型针对这个问题进行建模,并把该问题转化为一个整数规划问题,然后通过遗传算法进行了求解。通过仿真实验将提出方法与最短路径方法进行了对比,证明尽管提出方法的加工路径更长,但加工时间却更短,从而说明了该算法的实用性。

关键词: 机器人, 数控, 随机方法, 长加工路径, 多点加工

Abstract

Time-optimal trajectory planning for robotic manipulators and CNC machines is of great importance for improving the efficiency and manufacture precision in industry. This thesis presents efficient algorithms in three cases from practical applications. Further more, simulations and experiments are implemented to valid the effectiveness and practicability of the proposed algorithms.

1. In industry, there always exist uncertainties in robotic dynamics parameters, such as modeling errors, stochastic variations of material and workspace, environment manufacturing errors and so on. Taking this case into consideration, we construct the stochastic model of optimal trajectory planning for robotic manipulators, with probability inequality constraints under confined torques and path. And then a new stochastic algorithm is proposed to solve it. This method first simplifies the probability inequalities with deterministic conditional functions. After that, by some reasonable assumptions, we reduce the problem into a linear programming. At last we compare the proposed method with other stochastic and robust method respectively. It shows that the proposed method not only has a more close approximation of the original problem, but also equivalent to it in the case of "strict robust". Simulations also valid it.

2. In the trajectory planning problem for long toolpaths, since the long toolpaths always have large number of constraint points but also overload the hardware condition of CNC systems, they usually be processed with sections in zero boundary conditions. This causes loss of efficiency. This thesis presents a new algorithm to solve multiple sections of long toolpaths sequentially, then connects each of them in a loop. The robustness and complexity of the proposed algorithm are also discussed. In the last, the algorithm is tested in a 2D cat-shape toolpath and 3D tooth-shape toolpath respectively, which not only shows that the assembled solution performs close to the global "pseudo-jerk" method($< 0.1\%$), but also the computational load is notably reduced.

3. For multi points manufacturing problems in drilling/spot welding tasks

in which the path is not fixed, only be required to cross specified points, we put forward a new algorithm using cubic Hermite interpolation polynomial and modified TSP which first reduces it into a mixed inter problem and then solves it with genetic algorithm. In the simulation part, we compare the proposed method with minimum distance strategy, the results shows that although the distance of the path we gain is much longer the the minimum distance strategy, the traversal time is much shorter. This phenomenon shows the practicability of the proposed algorithms.

Keywords: Robotic manipulator, CNC, stochastic, long toolpaths, multi-points drilling/spot welding

目 录

摘要	i
Abstract	iii
目录	v
第一章 引言	1
1.1 机器人和数控技术的发展	1
1.1.1 机器人的发展	1
1.1.2 数控技术的发展	2
1.2 机器人和数控机床的轨迹规划问题	3
1.3 机器人最优鲁棒轨迹规划的随机方法	5
1.4 长加工路径的高效轨迹规划算法	6
1.5 机器人在多点打孔和点焊过程中的轨迹规划	7
1.6 本论文概述	8
第二章 机器人最优鲁棒轨迹规划的随机方法	11
2.1 预备知识	11
2.1.1 凸优化问题	11
2.1.2 求解算法的选择及复杂度分析	14
2.2 机器人鲁棒轨迹规划问题	14
2.2.1 机器人时间最优轨迹规划问题	14
2.2.2 鲁棒轨迹规划的随机模型	16
2.3 随机模型的线性化公式	16
2.3.1 将随机模型简化到非凸的确定性模型	16
2.3.2 将非凸模型归纳为线性化模型	19

2.3.3	严格鲁棒模型的线性化等价模型	21
2.4	数值算法	22
2.4.1	B样条离散化	22
2.4.2	如何选择最优的分位数	24
2.4.3	如何计算分位数	25
2.5	仿真实验	26
2.5.1	与其他轨迹规划随机方法的对比	27
2.5.2	与严格鲁棒轨迹规划方法的对比	29
2.6	一般化的随机理论	31
2.7	结论	35
第三章	长加工路径的高效轨迹规划算法	37
3.1	长加工路径轨迹规划问题及线性化方法	37
3.1.1	长加工路径轨迹规划问题	37
3.1.2	线性化方法	38
3.2	长加工路径的高效轨迹规划算法	38
3.2.1	分段高效拼接算法	39
3.2.2	B样条离散化方法	40
3.2.3	复杂度分析	41
3.3	仿真与实验	41
3.3.1	二维猫形路径实验	42
3.3.2	三维牙齿形路径实验	43
3.4	鲁棒性分析	44
3.5	结论	45
第四章	机器人在多点打孔和点焊过程中的轨迹规划算法	49
4.1	问题描述	49
4.2	子问题: 求解点到点的路径以及最短转移时间	51
4.3	主问题: 多点的最短时间轨迹规划	53

4.4 仿真与实验	55
4.4.1 点到点转移路径的测试	55
4.4.2 多点规划最短加工时间的测试	56
4.5 结论	59
第五章 结论与展望	61
5.1 本文总结	61
5.2 未来的研究方向	62
参考文献	63
发表文章目录	73
简历	75
致谢	77

第一章 引言

1.1 机器人和数控技术的发展

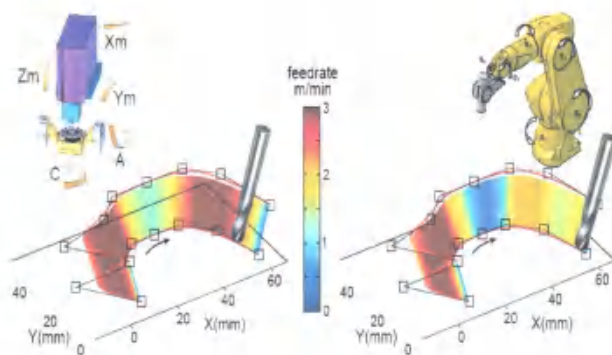


图 1.1: 数控机床和机械手臂示例[8]

1.1.1 机器人的发展

美国机器人协会给现代机器人下的定义是：一种可编程和多功能的操作机，或是为了执行不同的任务而具有可用电脑改变和可编程动作的专门系统。

按照这个定义，机器人的研究和发展历史可能只有几十年。第一台商用工业机器人在1962年由美国Unimation公司生产。在这以后，工业机器人就和计算机辅助设计(CAD)以及计算机辅助制造(CAM)结合在了一起，形成了现在制造自动化的最新发展趋势。到20世纪80年代，日本、美国等国家开始迅速普及工业机器人，国际上因此称1980年为“机器人元年”。到20世纪90年代，由于机器人价格的降低而劳动力成本增高，并且机器人在工业领域的高效、准确等优势，使得越来越多的工业实现了机器人自动化。另一方面，随着机器人作业能力的提高，很多危险的或者人类不可能实现的任务也开始由机器人处理。但直到2000年，在美国78%的机器人仍然仅用于焊接和搬运作业，在对技术要

求更高的专业领域,比如机器人装配,则大概仅有10%的机器人[16]。到2008年底,根据美国电气和电子工程师协会的统计,世界各地已经部署了100万台各种工业机器人。

机器人大致可以分为三代。其中第一代机器人为示教再现型,即人“手把手”的给出指令控制机器人一步一步进行作业;第二代则具备了感知能力:它们通过传感器对外界环境通过听觉、视觉、触觉等获取信息,依次来灵活调节自身的工作状态。比如有触觉的机械手可灵活抓取鸡蛋,具有视觉的机器人能进行精确投篮;第三代为智能机器人,它不仅具备感知能力,而且还具有独立推理和决策、判断和行动的能力,利用人工智能、数据挖掘等算法,甚至在某些领域达到“拟人化”。

目前,智能机器人的应用范围大大增加,除工业生产外,已经渗透到各行各业,正朝着高精度、智能化、拟人化这条没有止境的道路蓬勃发展。

1.1.2 数控技术的发展

数控(Numerical Control)是由存储在介质里面的精确的编程命令进行自动化机床控制的一种技术。目前大多数数控实际上是计算机数控(Computer Numerical Control),在里面计算机是一个完整的组成部分。计算机数控技术的载体便是数控机床(Computer numerical control machine tools),这是一种由计算机程序控制的自动化机床。

数控技术本身的发展可以追溯到20世纪40年代末。1948年,受美国空军委托,美国帕森斯公司在研制直升飞机螺旋桨叶片检验样板加工设备过程中,由于样板形状复杂,要求精度高,之前一般的加工设备难以适应,于是提出采用数字脉冲控制机床的设想。之后一年,该公司与美国麻省理工学院(MIT)开始共同研究,并于1952年成功研制出世界上第一台基于电子管和继电器的机床控制装置,用于三轴立式铣床的控制,这标志着第一台数控机床的诞生。而中国第一台真正意义上的数控机床诞生于1958年9月,由清华大学和铣床研究所合作研发完成并在北京第一机床厂试制成功。

在这之后,数控系统大概又经历了五代的发展历程[1]。从20世纪50年代末,出现了第二代晶体管数控系统,这取代了昂贵的、易损坏的第一代电子管数控系统;之后在1965年,诞生了第三代集成电路数控系统,不仅降低了生产成本,还提高了数控系统的可靠性;1970年第四代小型计算机数控系统出现,

这使得计算机软件控制数控机床成为现实,但当时由于受成本因素的限制,发展较慢;1974年第五代微型计算机数控系统诞生,才使得真正的计算机数控技术得到快速和广泛的发展,因为微处理器的高度集成,带来的高速度、低成本、低价格满足了工业控制系统的需求;最后到20世纪80年代,随着微处理器从16位到32位的过渡,第六代通用化的个人计算机数控系统迅速发展,这也是数控系统最具有深远意义的一次飞跃,因为个人计算机数控系统的硬件平台与软件平台是完全通用的,从而避免了前面五代专用数控系统开发周期长、性能难以达到最佳、开发和生产成本高等问题。

现在数控技术的发展趋势也更加多元化,正在朝着超高速、超精、智能化、网络化、复合化等方向快速跃进。

1.2 机器人和数控机床的轨迹规划问题

机器人和数控机床的轨迹规划问题一般指的是在路径给定或者不定的情况下,设计机器人或者数控系统在路径上相应移动速度的问题。

这类问题目标函数常常为时间最优[23, 79, 81, 82, 86, 87]、能量最低[73]等等,约束条件主要分为两类:一类是运动学的约束,比如速度、加速度、加加速度甚至加加加速度[19, 25, 34, 55, 56, 69, 82]等。另一类为动力学约束,比如力矩、力矩变化率、电压[10, 15, 33, 53, 71, 74, 87]等。

本论文涉及到的轨迹规划的目标函数主要采用的是时间最优。在实际生产中,时间最优的轨迹规划问题对于提高机器人和数控机床的加工效率意义重大。但完整的求解这个问题一般来说是个很复杂很有难度的任务[14, 18, 38, 54, 66, 76]。这之后出现了在系统状态空间内求解轨迹规划问题的直接方法[14]和分解方法[38, 58, 59, 61],它们的计算复杂度均有所降低。

分解方法求解轨迹规划问题一般分为两步:第一步是通过规划产生一条满足任务要求、障碍躲避和符合其它几何限制的路径[38, 58, 59],在这个过程中暂时忽略比如运动学等的约束。在第二步中则是根据生成的路径,在考虑机器人和数控机床的动力学或者运动学限制的条件下,找到最优的速度规划[7, 10, 13, 15, 46, 48, 56, 57, 60-62, 65, 68, 78]。

目前多数的最优时间轨迹规划方法仍是基于分解方法,并且很多主要用在其第二步求解上,即在给定的轨迹上,寻找最大的满足动力学约束和运动学约束的速度。该问题就目前考虑的约束来说,具有一个很好的“bang-bang”的

性质：最优速度总是满足在任意时刻，约束条件有一个取得等号。这意味着速度、加速度或者扭矩等在这一时刻达到饱和[13, 62, 65]。这一类算法中，仅考虑力矩约束的就有[7, 10, 13, 15, 40, 48, 56, 57, 60–62, 64, 65, 68, 78]。这些算法粗略分为三类，其中大多数都是首先通过路径参数 s 和它对时间的导数 \dot{s} [10, 61, 65]把多维的状态空间简化为二维的状态空间。然后求解得到的 (s, \dot{s}) 曲线就大概决定了路径给定情况下时间最优轨迹规划问题的解。

第一类为间接方法，首先在[10, 61, 65]提出，之后又依次有了[60], [57], [48], [78]。为了表述简洁，下面只讨论[65]，这种方法比[10, 61]求解至少快一个数量级。它首先通过在搜索一维情形下的路径参数 s 找到全部的切换点，这些切换点意味着在这些位置，达到饱和的约束条件需要切换。基于这些切换点，利用 s ，从两端开始进行一个数值迭代算法，就可以得到最优的参数速度曲线 (s, \dot{s}) 。

第二类方法则是动态规划方法[62, 64, 65]。第三类方法为直接方法[7, 15]。这三类方法对比，第二类方法和第三类方法能够处理带有更一般约束和复杂目标函数的问题，比如目标函数由时间最优可以变为能量最低，约束条件添加二阶或者二阶以上约束(比如加加加速度、力矩变化率等等)。本论文的求解方法主要基于第三类直接方法。

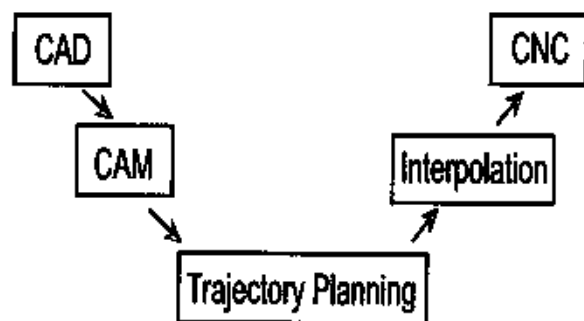


图 1.2: 数控加工过程

图1.2描述了数控加工的一般的过程。加工轨迹首先经过 CAD(Computer Aided Design)进行几何建模，然后，根据 CAD提供的几何信息，CAM(computer-aided manufacturing)会结合机床或者机器人的结构、加工工具、残留高度等信息生成机床或者机器人的移动路径。轨迹规划这里主要是对生成的加工路径结

合数控系统/机器人的动力学、运动学约束生成合适的加工速度。插补则是对生成的加工速度进行加密处理,使得相邻两点加工速度之间的间隔为一个插补周期(比如0.001秒),从而得到移动过程中的刀具或者机器人移动轨迹的插补点,最后则是把插补点按照一定的格式(比如G代码)输入数控系统/机器人中进行处理。

接下来本文对轨迹规划三个重要的情形概况进行综述,并在最后一节简要介绍本人做过的相关工作。

1.3 机器人最优鲁棒轨迹规划的随机方法

机器人轨迹规划对于提高机器人系统的效率至关重要。很多的技术方法被用在了这个基本问题上,包括速度限制曲线和积分方法[10, 15, 47, 61, 79],庞加莱极大值原则和打靶方法[13],最优化方法[17, 74]和在线计算方法[34]等。

在上面的方法中,模型的参数被假定是准确知道的并且等于某些参考值。但在实际中,由于存在模型误差、工作环境和材料的随机误差、加工误差等,机器人系统总是存在不确定性[40]。为了处理这些不确定性,鲁棒的轨迹规划和参数估计方法被提出。一种最基本的鲁棒策略就是最小-最大优化方法[31, 63],这种方法通过在“最坏的干扰”下求得最优解,因此也称为“严格鲁棒”。在[86]中,在给定力矩条件下最小-最大策略的轨迹规划被转化为了一个线性规划问题,由于这个问题特殊的结构,特别是双层的结构,在一般的情况下,这给有效的数值解法带来挑战。还有一些方法注意力集中在不确定参数的特殊分布上面,比如这些不确定的参数在假设噪音是正态分布零均值的条件下通过最小二乘方法被估计和代替[77];在[24]中,在假定噪音是单峰分布并且独立于控制后,使用了参数的期望。除了上面这些,关于这个问题的另一种重要的方法就是随机优化方法[39, 40],这种方法把关于不确定参数的不等式约束用大于等于某个给定概率常数的概率函数来代替,而等式约束和目标函数则变为了它们关于不确定参数的期望的约束和期望的目标函数。这种随机优化方法与“严格鲁棒”相比更一般化(概率常数为1的时候就变为严格鲁棒),因此也更难解决。这种随机优化方法的运用目前主要在专属领域机器人方面、辅助装置方面以及与人互动的机器人方面。在这些例子中,不确定性有时候影响较大,但是机器人或者相关装置并不能因为比如电压超界就关闭运作,因为这样会在一些情况下引起危险。

1.4 长加工路径的高效轨迹规划算法

在轨迹规划中,解的最优性和计算效率是一对相互矛盾的需求,而一个设计好的轨迹规划算法可以大大的减少加工时间,从而会在工业制造的计算机数控系统(CNC)中得到很广泛的应用。

这类问题的算法大体可以分为两类,一类是解析算法。比如[10]等人最先利用相平面分析的方法在带力矩约束的时间最优机器人路径规划问题中找到了解析解。[60]则在该基础上提出了一个更好的寻找速度限制曲线切换点的方法,接着[70]把该算法引入到了数控系统领域。[83]则在上述算法的基础上,给出了一个针对二次B样条路径的简化算法。然而,解析算法由于对于复杂的路径和高维的情况(比如五轴加工),计算复杂度过高,因此另一类计算效率更高的数值算法得到了迅速的发展。

初期这些数值方法也都是非线性规划算法。包括最初的机器人离散时间模型方法[43],后来还加入力矩和加加速度约束[67]。[25]则进一步在目标函数里面加入了加加速度平方积分项,以此来达到光滑轨迹的作用。当优化变量比较多的时候,这些非线性规划方法的复杂度还是很高,于是又出现了一些简化算法。比如[3]提出了窗口法,将整个非线性规划问题分解为一系列的子问题来求解。[19, 20]则是提出了贪心算法的思想并使用一系列的单变量的非线性规划问题来优化最优的速度。[75]则利用对数罚函数的方法求得该问题的一个近似解。真正意义的简化是[74],其把最优时间的轨迹规划问题转化为了一个凸优化问题。更进一步地, [87]提出“pseudo-jerk”的思想,把这个问题近似为了一个线性规划问题。这极大的提高了数值方法的求解效率。

但是,在实际加工中,轨迹规划有时需要处理很长的路径[22]。这带来了两个难题。一是由于长加工路径插补点过多、计算时间的要求、机床存储空间的限制等,最普通的处理算法是分段加工,并且每段都是用零初末速度。这样引起了一定程度的效率的损失。第二是长加工路径数值算法离散的时候,需要的限制点过多,导致即便是最简单的线性规划模型(变量个数可能达到百万级别),个人电脑都无法求解。

在[22]中,使用了一系列可行解拼接的方法,主要思想就是首先找到第一段的可行解,然后通过计算包含第一段但稍长的路径的可行解,找到它们共同的部分作为全局可行解的一部分,进而使长加工路径变“短”,这样就得到了一个相对好一些的可行整体解。这种方法有两个问题,一个是计算的复杂度仍

然较高,因为要不断的计算稍长路径的轨迹规划;另一个是这种方法是基于可行解的拼接方法,最后得到的仍然只是一个可行解,与最优解相差在某些情况下依旧很大。目前具本文所知,还没有文献对最优解进行拼接。

1.5 机器人在多点打孔和点焊过程中的轨迹规划

机器人的时间最优轨迹规划问题在工业中已经被广泛研究并且一些有效的算法已经提出。比如在给定路径,时间最优的条件下,[10]提出了相平面分析法;相似的问题也被[82]通过一种贪婪搜索方法和[84, 85]中的凸优化方法解决。但对于更一般化的点到点的轨迹规划问题,这个问题变得复杂起来。因为这时需要把路径和速度同时进行规划。[9]首先运行相平面分析方法计算给定点到点的时间路径,然后点到点的轨迹规划问题就变为了在可行域内寻找最短时间路径的问题。

与上面提到的简单的轨迹规划问题不同,在工业生产制造中,存在着一类称为“多点加工”的复杂的任务,比如打孔,点焊等等。在这种任务中,加工点的次序是不确定的,因此有必要规划一个策略使得加工过程一次性通过所有的点当且一次,并且同时满足目标的要求,比如最短时间、最短路径、最小能量等等。

通过研究发现,多点打孔、点焊任务均可以被描述为一个旅行商模型(TSP)[21, 45, 80]:机器人的终端行为可以看做一个销售员,它从一点开始,每次穿过一个点一次,同时要完全停下执行加工任务。由于很高的计算复杂度,TSP问题的求解到现在仍然是一个开放性的问题。目前,TSP可行的求解方法可以归类为:枚举方法[72]、动态规划方法[6]、分枝定界方法[36]和智能优化方法(比如遗传算法(GA)[50]、模拟退火算法(SA)[2]、粒子群算法(PSO)[30]等)。

为了简化这个问题,一般的轨迹规划策略常常假设点与点之间的路径是直线,之后这个问题就被描述为一个最短距离的TSP模型[80]。但是根据[9, 21],已经证实由于机器人重力项的扭矩以及运动学方程的非线性性,最短时间路径和最短路径是不一致的。即便对于从点 i 到点 j 的最短时间路径,也有可能与从点 j 到点 i 的最短时间路径不一样,并且除了优化要穿过的点的次序,点与点之间的最优路径也需要优化。

1.6 本论文概述

本论文针对机器人和数控机床最优轨迹规划中的三个子问题：机器人最优鲁棒轨迹规划问题、长加工路径的高效轨迹规划问题、不定路径机器人多点打孔、点焊轨迹规划问题，分别进行了研究并且提出了有效的解决方法。

第二章提出了一种有效的更一般化的随机算法来解决机器人在给定路径条件下的鲁棒轨迹规划问题。

首先，利用随机方法理论，对这个问题建立了随机模型。然后利用分位数和条件概率等统计学的知识，把带概率不等式约束的随机模型简化为了带条件函数不等式约束的随机模型。然后通过合理的假设，进一步地又把条件函数用一系列的确定性的线性函数代替，从而最终把原来的非线性非凸问题转化为一个线性规划问题。这极大的降低了求解的复杂度，并且当随机模型的概率常数变为1时，这个线性规划问题被证明与原来的随机模型还是等价的。仿真实验通过与以前随机方法和严格鲁棒方法的对比也说明了提出方法的有效性。在本章最后一节，基于这个问题，抽象出了解决一般概率不等式约束的随机理论，该理论可以被用于解决带有 n 个随机变量的情形，因此可以被拓展到更多的问题中。

第三章则提出了一种基于线性规划的长加工路径的高效轨迹规划方法。

这种方法基于速度曲线分段拼接的思想，选的拼接位置为相邻两段速度曲线最接近的两个极小值点。拼接结束后还会使用每段改变了的边界条件对每段速度曲线进行一次更新，从而可以进行下一次的拼接。为了保证拼接算法得以有效鲁棒的运行，本章中还提出了避免引起数值误差的两种方法，并且讨论了这种高效计算算法的浮点计算复杂度。最后通过机床对二维和三维图案分别进行了加工，从而验证了算法的有效性和鲁棒性。实验证明，这种算法不仅与全局最优解[23, 86]非常接近，并且计算效率提高大概两个数量级。

在第四章中则是对机器人多点打孔、点焊任务的轨迹规划问题进行了研究。该问题与前者最大的不同是路径是不固定的，因此要把路径的形状和速度一起规划。

本论文提出的方法首先利用三次Hermite插值多项式对路径进行建模，找到点与点之间在该限制下的最优路径，然后计算该路径的最短加工时间，组成时间度量矩阵。利用这个矩阵，建立了改良的旅行商模型，进一步把这个问题最终化为一个整数规划问题。最后利用遗传算法，对这个整数规划问题进行

了解。

最后，第五章总结了本论文，并列出了未来可以研究的方向。



此页不缺内容



第二章 机器人最优鲁棒轨迹规划的随机方法

本章主要研究了在给定路径以及关于力矩的概率不等式约束下的机器人轨迹规划问题。力矩的概率不等式约束指的是力矩不超过给定的界的概率大于或者等于给定的概率常数 κ 。本章提出一种新的随机的方法把概率不等式约束简化为了一系列的确定性的不等式约束,并且证明,在一定的条件下,这些确定性的不等式约束又等价于一系列的线性约束,从而把原来的非凸非线性的随机规划模型转化为一个线性规划模型,在多项式时间内求得最优解。当概率常数 $\kappa = 1$ 时,这个线性规划模型被证明与原来的随机模型还是等价的。总体来说,这是一种普遍的、实用的、有效的算法。仿真实验也用数据说明了这种方法的有效性。

本章的结构如下:首先第一节我们回顾了关于凸优化的基础知识,为后续问题的叙述和理解做铺垫;第二节介绍了机器人鲁棒轨迹规划问题,并提出了在给定路径以及关于力矩的概率不等式约束下的随机模型;第三节对该模型进行了线性化处理;第四节对得到的线性化模型利用B样条离散化的方法进行数值求解;在第五节对随机方法进行了仿真实验,并与其他随机方法和严格鲁棒方法进行了对比;第六节从该模型出发提出了简化一般情形下带有 n 个随机变量的概率不等式约束的随机理论;最后一节对提出的随机方法进行了总结。

2.1 预备知识

在本节,我们介绍一下凸优化的一些基础知识,包括基本概念、凸问题模型、常用求解方法以及凸优化特殊的形式线性规划。因为后面机器人和数控机床最优轨迹规划问题的建立以及求解几乎都是建立在凸优化或者线性规划基础上进行的。了解凸优化的性质、求解方法、求解复杂度等有助于更好地理解后续问题。

2.1.1 凸优化问题

定义 2.1. 集合 $C \in \mathbb{R}^n$ 称为凸集, 如果它满足:

$$\alpha x + (1 - \alpha)y \in C, \forall x, y \in C, \forall \alpha \in [0, 1]$$

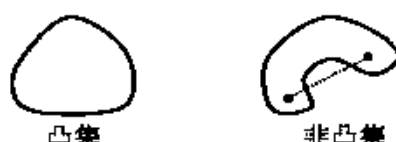


图 2.1: 凸集和非凸集示例

常见的凸集的例子有 n 维实数空间、范数约束形式的集合、仿射子空间、凸集的交集、 n 维半正定矩阵集等。

定义 2.2. 假设 C 为一个凸集，我们说函数 $f: C \rightarrow \mathbb{R}$ 是一个凸函数，如果它满足[5]：

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y), \quad \forall x, y \in C, \forall \alpha \in [0, 1]$$

如图2.2所示， $f(x)$ 即为一个凸函数。常见的凸函数的例子有指数函数、幂函数、负对数函数等。

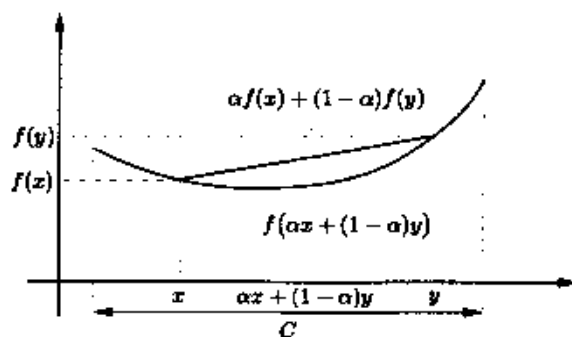


图 2.2: 凸函数示例

定义 2.3. 函数 $f(x)$ 的上镜图定义为: $\{(x, y) | x \in \text{dom} f, y \geq f(x)\}$, 其中 $\text{dom} f$ 指 f 的定义域。

凸函数的其他判定方法:

1. 若函数 $f(x)$ 的上镜图为一个凸集, 则该函数是凸函数。
2. 若函数 $f(x)$ 一阶可微, $\text{dom} f$ 为凸集, 并且对于 $\forall x, y \in \text{dom} f, f(y) > f(x) + \nabla f(x)^T(y - x)$, 则 $f(x)$ 为凸函数。
3. 若函数 $f(x)$ 二阶可微, $\text{dom} f$ 为凸集, 并且 $\nabla^2 f(x) \geq 0$, 则 $f(x)$ 为凸函数。

定义 2.4. 凸优化问题具备如下的标准形式[11]:

$$\begin{aligned} \min_x \quad & f_0(x) \\ \text{s.t.} \quad & f_i(x) \leq 0, i = 1, 2, \dots, m \\ & h_i(x) = 0, i = 1, 2, \dots, p \end{aligned} \quad (2.1)$$

其中 $f_0(x), f_i(x), h_i(x)$ 满足下面三个条件:

1. 目标函数 $f_0(x)$ 是凸函数。
2. 不等式约束函数 $f_i(x), i = 1, 2, \dots, m$ 是凸函数。
3. 等式约束函数 $h_i(x) = a_i^T x - b_i, i = 1, 2, \dots, p$ 是线性函数。

定理 2.1. 凸优化问题具有唯一的全局最优解, 并且其任意的局部最优解都是全局最优解[11]。

注 1. 凸优化问题上良好的性质, 使得许多的数值算法可以求解, 比如内点法[42]、信赖域反射算法[37]、序列二次规划方法[26]、活动集方法[27]等, 并且不必担心求的解落入局部最优解, 因为对于凸优化问题, 局部最优解都是全局最优解。

定义 2.5. 作为凸优化问题的特殊形式, 线性规划问题(LP)具备如下的形式:

$$\begin{aligned} \min_x \quad & c^T x + d \\ \text{s.t.} \quad & Gx \leq h \\ & Ax = b \end{aligned} \quad (2.2)$$

其中 $G \in \mathbb{R}^{m \times n}, A \in \mathbb{R}^{p \times n}$ 。

注 2. 对于线性规划问题, 除适用上面提到的凸优化求解方法外, 还有比如单纯形法[32, 41]、对偶单纯形法[4, 44]等。由于线性规划是凸优化的一种最简单的形式, 因此其求解速度较普通的凸优化问题大大提高。

2.1.2 求解算法的选择及复杂度分析

(1) 凸优化求解算法的选择:

内点法一般用于大规模的稀疏问题,或者小规模稠密问题。这种算法在每一步的迭代中都满足边界条件,并且即便遇到无穷的值也能继续迭代;

序列二次规划方法具备前者的优点,但遗憾的是它不是一种大规模问题计算方法;

信赖域反射算法要求函数的梯度以及约束必须是有界的或者线性的两者满足之一,由于要求较多,这种方法求解大规模稀疏问题和小规模稠密问题效率都非常的高,而且该方法是一种大规模问题计算方法;

单纯形法和对偶单纯形法只用于线性规划,一般计算过程占用内存小,并且尽管理论的算法复杂度是指数级的,但实践效果却常常很好。

(2) 凸优化求解复杂度分析[11]:

假设求解一般的凸优化问题和线性规划问题使用的都是最常用的内点法。

则对于线性规划问题,假设其未知变量个数为 n ,限制条件个数为 m (一般有 $m \geq n$),则最坏情况浮点计算复杂度为 $O(n^2m)$ 。

对于一般的凸问题,假设其未知变量个数为 n ,限制条件个数为 m ,则最坏情况浮点计算复杂度为 $O(\max\{n^2m, n^3, F\})$,其中 F 为计算该问题目标函数和所有限制函数一阶和二阶导数需要的浮点计算次数,一般情况下 F 的浮点计算复杂度要远远大于 n^2m 和 n^3 。

2.2 机器人鲁棒轨迹规划问题

2.2.1 机器人时间最优轨迹规划问题

n 自由度的机器人的动力学方程可以表示成如下形式[15]:

$$\tau = M(q)\ddot{q} + \dot{q}^T C(q)\dot{q} + G(q) \quad (2.3)$$

其中 $\tau \in \mathbb{R}^n$ 是力矩向量, $q \in \mathbb{R}^n$ 是关节的角位移, $M(q) \in \mathbb{R}^{n \times n}$ 是机器人的惯性矩阵, $C(q) \in (\mathbb{R}^n)^{n \times n}$ 作为一个矩阵,其元素属于 \mathbb{R}^n ,代表着科氏力和离心力, $G(q) \in \mathbb{R}^n$ 是重力项。

假设机器人移动的几何路径已经被路径规划算法[35]确定下来。这条路径假设有如下的参数形式:

$$r = r(s) = [x_1(s), \dots, x_n(s)]^T, s \in [0, 1], n = 2, 3 \quad (2.4)$$

举例来说, 当 $n = 3$, $r(s)$ 就是一条空间的曲线。进一步, 假设 $r(s)$ 至少 C^1 光滑并且对于任何的 $s_0 \in [0, 1]$, 总是存在一个 $i \in \{1, \dots, n\}$ 满足 $\frac{dx_i}{ds}(s_0) \neq 0$ 。记 $\dot{r} = \frac{dr}{ds}$, $\dot{\cdot} = \frac{d}{dt}$, 其中 t 代表时间。通过使用前向运动学映射的雅克比矩阵 J , 得到机器人角位移 q 和移动路径 r 的关系如下:

$$\dot{r} = J(q)\dot{q} \quad (2.5)$$

假设机器人不存在奇异位置, 则雅克比矩阵 J 是可逆的, 并且关节的速度和加速度满足如下公式:

$$\dot{q} = q'\dot{s}, \quad (2.6)$$

$$\ddot{q} = q'\ddot{s} + q''\dot{s}^2 \quad (2.7)$$

其中 $q' = J^{-1}r'$, $q'' = J^{-1}r'' + (J^{-1})'r'$ 。通过(2.3), (2.6), 和(2.7), 力矩的动力学模型可以表示成关于 \ddot{s} 和 \dot{s}^2 的线性函数:

$$\tau = m(s)\ddot{s} + c(s)\dot{s}^2 + g(s) \quad (2.8)$$

其中 $m(s) = M(q(s))q'(s) \in \mathbb{R}^n$, $c(s) = M(q(s))q''(s) + q'(s)^T C(q(s))q'(s) \in \mathbb{R}^n$, $g(s) = G(q(s)) \in \mathbb{R}^n$ 。

像[74]中叙述的那样, 在参数确定情形, 力矩有界约束下的轨迹规划问题可以被表示成如下的凸优化控制问题:

$$\begin{aligned} \min_{b(s)} \quad & J = \int_0^1 \frac{1}{\sqrt{a(s)}} ds \\ \text{s.t.} \quad & a'(s) = 2b(s), a(0) = a_0, a(1) = a_1 \\ & a(s) > 0, \forall s \in (0, 1) \\ & \tau_{\min} \leq m(s)b + c(s)a + g(s) \leq \tau_{\max} \end{aligned} \quad (2.9)$$

其中 $a = \dot{s}^2$, $b = \ddot{s}$, $\tau_{\min} \in \mathbb{R}^n$ 和 $\tau_{\max} \in \mathbb{R}^n$ 分别是力矩的上下界, a_0 和 a_1 是 $a(s)$ 的边界条件。最近, 解决这个问题的有效的算法已经在[47, 74, 81, 86]中。

2.2.2 鲁棒轨迹规划的随机模型

由于机器人的动力学系统存在不确定性(比如建模误差、材料和工作环境引起的误差、生产制造过程中的干扰等), 公式(2.3)中参数变量 m, c 和 g 应该看作随机变量, 并且它们代表了几乎机器人系统的所有变化情况。与确定性的轨迹规划模型(2.9)不同, 鲁棒轨迹规划的随机模型有以下形式[28, 39]:

$$\begin{aligned} \min_{b(s)} \quad & J = \int_0^1 \frac{1}{\sqrt{a(s)}} ds \\ \text{s.t.} \quad & a'(s) = 2b(s), a(0) = a_0, a(1) = a_1 \\ & a(s) > 0, \forall s \in (0, 1) \\ & P(\tau_{\min} \leq m(s)b + c(s)a + g(s) \leq \tau_{\max}) \geq \kappa \end{aligned} \quad (2.10)$$

其中 $\kappa \in (0, 1]$ 是给定的概率常数。由于 a 和 b 与随机变量 m, c, g 没有直接的关系, 因此目标函数和等式约束关于 m, c, g 的期望还是它们本身。上面问题的物理意义就是寻找最优的加速度 $b(s)$ (也就可以确定速度 $a(s)$), 使得机器人满足力矩约束 $\tau_{\min} \leq \tau(s) \leq \tau_{\max}$ 的概率大于或者等于 κ , 并且完成给定的运动。

注 3. 在本章中, $m = M(q)q'$ 假设是一个连续的随机变量。

在下面, 将要提出一种解决上述问题(2.10)的有效算法。

2.3 随机模型的线性化公式

在这一节, 我们将证明随机模型(2.10)在适当的条件下能够通过一个线性规划模型来很好的近似。

2.3.1 将随机模型简化到非凸的确定性模型

首先, 引入两个概率的概念: 假设 X 是一个随机变量, 取值范围为 \mathbb{R} , 那么关于 X 的分布函数 F 定义如下:

$$F(x) = P(X \leq x)$$

其中 P 是概率函数。显然 F 是一个单调递增的函数, 并且对于任意一个 $\kappa \in (0, 1)$, 定义 $F^{-1}(\kappa)$ 的值为满足 $F(x) \geq \kappa$ 的最小的 x , 也就是:

$$F^{-1}(\kappa) = \inf_x \{x \in \mathbb{R} \mid F(x) \geq \kappa\}$$

则 $F^{-1}(\kappa)$ 就称为 X 的 κ 分位数。假设 a 是 X 的 κ 分位数, 那么如果 $P(X \leq a) \geq \kappa$, 则对于任意的 $b \in \mathbb{R}$, $P(X \leq b) \geq \kappa$ 意味着 $b \geq a$ 。进一步, 如果 X 是一个连续变量, 那么有 $P(X \leq a) = \kappa$ 。假设 A, B 是两个事件, $P(A|B)$ 表示在事件 B 条件下事件 A 发生的概率, 则有以下两个引理:

引理 2.2. 如果 $P(A) \geq \frac{1+\kappa}{2}$ 并且 $P(B) \geq \frac{1+\kappa}{2}$, 那么 $P(A \cap B) \geq \kappa$ 。

证明: 由于 $P(A \cap B) = P(A) + P(B) - P(A \cup B)$, $0 \leq P(A \cup B) \leq 1$, 因而有 $P(A \cap B) \geq \frac{1+\kappa}{2} + \frac{1+\kappa}{2} - 1 \geq \kappa$ 。

引理 2.3. 如果 $P(A|B) \geq \kappa_1$, $P(B) \geq \kappa_0$, 并且 $\kappa_0 \neq 0$, 那么 $P(A) \geq \kappa_0 \kappa_1$ 。

证明: 注意到 $P(A|B) = \frac{P(A \cap B)}{P(B)} \geq \kappa_1$ 。因此, $P(A) \geq P(A \cap B) \geq P(B) \cdot \kappa_1 \geq \kappa_0 \cdot \kappa_1$ 。

下面的定理把(2.10)中出现的概率不等式简化为了确定性的不等式。

定理 2.4. 假设 m_1, c_1, g_1 分别是随机变量 $m(s), c(s), g(s)$ 的 $\kappa_3, \kappa_2, \kappa_1$ 分位数, m_2, c_2, g_2 分别是随机变量 $-m(s), -c(s), -g(s)$ 的 $\kappa_3, \kappa_2, \kappa_1$ 分位数, 其中 $\kappa_3, \kappa_2, \kappa_1, \kappa_0 \in [0, 1]$ 并且满足 $\kappa_0 \kappa_1 \geq \frac{1+\kappa}{2}$, $\kappa_2 \kappa_3 \geq \kappa_0$, $\kappa_0 \neq 0$, $\kappa_3 \neq 0$ 。假设 a 和 b 与问题(2.10)中一致,

$$f(a, b) = \begin{cases} m_1 b + c_1 a + g_1, & b \geq 0 \\ -m_2 b + c_1 a + g_1, & b < 0 \end{cases},$$

$$g(a, b) = \begin{cases} -m_2 b - c_2 a - g_2, & b \geq 0 \\ m_1 b - c_2 a - g_2, & b < 0 \end{cases}$$

那么, $f(a, b) \leq \tau_{\max}$ 并且 $g(a, b) \geq \tau_{\min}$ 推出 $P(\tau_{\min} \leq m(s)b + c(s)a + g(s) \leq \tau_{\max}) \geq \kappa_0$ 。

在这里 κ 是问题(2.10)中预先确定的概率常数。 $\kappa_3, \kappa_2, \kappa_1$ 是用户事先选择的调节参数, 我们将在本章后面提供一个最优的选择它们的方法。

证明: 考虑如下的事件:

$$A: mb + ca + g \leq \tau_{\max}$$

$$B: mb + ca + g \geq \tau_{\min}$$

$$C: \tau_{\max} - mb - ca \geq g_1$$

$$D: \tau_{\max} - c_1 a - g_1 \geq mb$$

$$E_1: \tau_{\max} - c_1 a - g_1 \geq m_1 b$$

$$E_2: \tau_{\max} - c_1 a - g_1 \geq -m_2 b$$

首先断言 $P(A) \geq \frac{1+\kappa}{2}$ 并且 $P(B) \geq \frac{1+\kappa}{2}$ 。然后利用引理2.2, 我们就有 $P(\tau_{\min} \leq mb + ca + g \leq \tau_{\max}) = P(A \cap B) \geq \kappa$, 从而定理就得到证明。现在只需证明断言 $P(A) \geq \frac{1+\kappa}{2}$ 成立, 断言 $P(B) \geq \frac{1+\kappa}{2}$ 可以通过相似的方法证明。为了证明 $P(A) \geq \frac{1+\kappa}{2}$, 考虑如下三种情况:

首先证明 $P(D) \geq \kappa_3$ 。如果 $b \geq 0$, 那么 $f(a, b) \leq \tau_{\max}$ 就变为了 $m_1 b + c_1 a + g_1 \leq \tau_{\max}$, 从而 $P(E_1) = 1$ 。注意到 D 和 E_1 的左边的项是一样的, 因而 $P(D | E_1) \geq P(mb \leq m_1 b)$ 。如果进一步假设 $b > 0$, 利用分位数的定义, 我们有 $P(mb \leq m_1 b) = P(m \leq m_1) \geq \kappa_3$, 其中 m 是一个随机变量。如果 $b = 0$, $P(mb \leq m_1 b) = 1 \geq \kappa_3$ 也仍然成立。所以, $b \geq 0$ 可以推出 $P(D | E_1) \geq \kappa_3$ 。利用引理2.3, $P(D) \geq P(D | E_1) \cdot P(E_1) \geq \kappa_3$ 。类似地, 如果 $b < 0$, 则有 $P(E_2) = 1$ 和 $P(D) \geq P(D | E_2) \cdot P(E_2) \geq P(mb \leq -m_2 b) \geq P(-m \leq m_2) \geq \kappa_3$ 。从而 $P(D) \geq \kappa_3$ 得到证明。

第二步接着证明 $P(C) \geq \kappa_0$ 。由于 $a = \dot{s}^2 > 0$, $P(C | D) \geq P(ca \leq c_1 a)$, 利用分位数的定义, 我们有 $P(ca \leq c_1 a) = P(c \leq c_1) \geq \kappa_2$ 。因此 $P(C | D) \geq \kappa_2$ 。通过引理2.3, 得到 $P(C) \geq P(C | D) \cdot P(D) \geq \kappa_2 \kappa_3 \geq \kappa_0$ 。

最后证明 $P(A) \geq \frac{1+\kappa}{2}$ 。再次利用分位数的定义, 得到 $P(A | C) \geq P(g \leq g_1) \geq \kappa_1$ 。根据引理2.3, $P(A) \geq P(A | C) \cdot P(C) \geq \kappa_1 \kappa_0 \geq \frac{1+\kappa}{2}$, 从而断言得证。

根据定理2.4, 随机问题(2.10) 可以被简化为确定性非凸问题(2.11), 后者是前者的一个可行解。进一步, 实验结果显示(2.11)还是随机问题(2.10)的一个很好的近似, 特别是当概率常数 κ 很大的时候。

$$\begin{aligned}
 \min_{b(s)} \quad & J = \int_0^1 \frac{1}{\sqrt{a(s)}} ds \\
 \text{s.t.} \quad & a'(s) = 2b(s), a(0) = a_0, a(1) = a_1 \\
 & a(s) > 0 \text{ for } s \in (0, 1) \\
 & f(a, b) \leq \tau_{\max}, g(a, b) \geq \tau_{\min}
 \end{aligned} \tag{2.11}$$

其中 $f(a, b)$ 和 $g(a, b)$ 的定义和定理2.4一致。

2.3.2 将非凸模型归纳为线性化模型

由于 $f(a, b)$ 和 $g(a, b)$ 是变量 b 的一个条件函数, 所以带有约束 $f(a, b) \leq \tau_{\max}$ 和 $g(a, b) \geq \tau_{\min}$ 的问题(2.11)一般来说是一个非凸问题。在这一部分, 我们将会引入一个条件, 并在这个条件下把问题(2.11)简化为一个线性模型, 这意味着它的目标函数和约束条件关于变量 a 和 b 都是线性的。

引理 2.5. 使用定理2.4中的记号。假设 $\kappa_3 \geq \frac{1}{2}$, 那么 $m_1 + m_2 \geq 0$ 。

证明: 利用分位数的定义, 我们有 $P(m \leq m_1) = \kappa_3$, $P(-m \leq m_2) = \kappa_3$ 。从而, 得到 $P(-m \geq -m_1) = \kappa_3$ 和 $P(-m \leq -m_1) = 1 - \kappa_3$ 。由于 $\kappa_3 \geq \frac{1}{2}$, 我们有 $1 - \kappa_3 \leq \kappa_3$ 。再次根据分位数的定义, 由于 $1 - \kappa_3 \leq \kappa_3$, 我们得到 $-m_1 \leq m_2$, 这也就是 $m_1 + m_2 \geq 0$ 。

引理 2.6. 使用定理2.4中的记法。假设 $A = \{(a, b) : f(a, b) \leq \tau_{\max}\}$, $B = \{(a, b) : (m_1 b + c_1 a + g_1 \leq \tau_{\max}) \cap (-m_2 b + c_1 a + g_1 \leq \tau_{\max})\}$, $C = \{(a, b) : g(a, b) \geq \tau_{\min}\}$, $D = \{(a, b) : (-m_2 b - c_2 a - g_2 \geq \tau_{\min}) \cap (m_1 b - c_2 a - g_2 \geq \tau_{\min})\}$ 。如果 $m_1 + m_2 \geq 0$, 那么 $A = B$ 并且 $C = D$ 。

证明: 我们首先证明 $A = B$ 。显然 $B \subseteq A$, 因而我们只需证明 $A \subseteq B$ 。给定任意 $(a, b) \in A$, 如果 $b \geq 0$, 利用 $f(a, b)$ 的定义, 我们有 $m_1 b + c_1 a + g_1 \leq \tau_{\max}$ 。又由于 $m_1 + m_2 \geq 0$, $b \geq 0$, 在 $c_1 a + g_1 \leq \tau_{\max} - m_1 b$ 两边分别加上 $-m_2 b$, 我们得到 $-m_2 b + c_1 a + g_1 \leq \tau_{\max} - (m_1 + m_2)b \leq \tau_{\max}$ 。因此 $(a, b) \in B$ 。如果 $b < 0$, 利用 $f(a, b)$ 的定义, 我们有 $-m_2 b + c_1 a + g_1 \leq \tau_{\max}$ 。类似地, $m_1 b + c_1 a + g_1 \leq \tau_{\max} + (m_1 + m_2)b \leq \tau_{\max}$, 所以有 $(a, b) \in B$ 。 $A = B$ 得证。相同的方式, 我们也能证明 $C = D$ 。

根据引理2.5和2.6, 如果 $\kappa_3 \geq 1/2$, 那么问题(2.11) 与以下的问题等价:

$$\begin{aligned}
 \min_{b(s)} \quad & J = \int_0^1 \frac{1}{\sqrt{a(s)}} ds \\
 \text{s.t.} \quad & a'(s) = 2b(s), a(0) = a_0, a(1) = a_1 \\
 & a(s) > 0 \text{ for } s \in (0, 1) \\
 & m_1 b + c_1 a + g_1 \leq \tau_{\max} \\
 & -m_2 b + c_1 a + g_1 \leq \tau_{\max} \\
 & -m_2 b - c_2 a - g_2 \geq \tau_{\min} \\
 & m_1 b - c_2 a - g_2 \geq \tau_{\min}
 \end{aligned} \tag{2.12}$$

在上式中由于目标函数 $\frac{1}{\sqrt{a(s)}}$ 关于 a 是凸的并且约束条件关于 a 和 b 是线性的, 因此上述问题是一个凸优化问题。

根据[20, p. 385], 对于一个时间最优轨迹规划问题, 如果它的约束条件只有速度和加速度, 那么这个问题的最优解是唯一的并且在任意时刻速度都是最大的。从而我们有如下的性质:

引理 2.7. 问题(2.12) 有唯一的最优解 $\hat{s}_m(s)$, $s \in [0, 1]$, 并且对于任意的参数 s , $\hat{s}_m(s)$ 在该点的值都等于所有可行解在该点取值的最大值。

参考文献[10, 61] 也用不同的方法证明了这个性质。根据引理2.7, 问题(2.12)的目标函数可以用机器人终端关节的移动速度 $F_v(a) = \|r'(s)\|^2 a(s)$ 来代替, 其中 $\|r'(s)\| = \sqrt{x_1'^2(s) + \cdots + x_n'^2(s)}$ 。这个想法就是最大化机器人终端关节的移动速度 $F_v(a)$ 。注意到 $F_v(a)$ 关于 a 是线性的, 因此最终我们得到关于随机优化模型(2.10)的简化模型如下:

$$\begin{aligned}
 \max_{b(s)} \quad & \int_0^1 \|r'(s)\|^2 a(s) ds \\
 \text{s.t.} \quad & a'(s) = 2b(s), a(0) = a_0, a(1) = a_1 \\
 & a(s) > 0 \text{ for } s \in (0, 1) \\
 & m_1 b + c_1 a + g_1 \leq \tau_{\max} \\
 & -m_2 b + c_1 a + g_1 \leq \tau_{\max} \\
 & -m_2 b - c_2 a - g_2 \geq \tau_{\min} \\
 & m_1 b - c_2 a - g_2 \geq \tau_{\min}
 \end{aligned} \tag{2.13}$$

其中 m_1, c_1, g_1 分别是随机变量 m, c, g 的 $(\frac{1+\kappa}{2})^{\frac{1}{3}}$ 分位数, m_2, c_2, g_2 分别是随机变量 $-m, -c, -g$ 的 $(\frac{1+\kappa}{2})^{\frac{1}{3}}$ 分位数。

定理 2.8. 假设 $\kappa_3 \geq \frac{1}{2}$, 那么问题(2.11) 和问题(2.13) 有相同的唯一的最优解。

证明: 根据引理2.7, 问题(2.12)关于 α 和 b 有唯一的最优解。下面只需证明问题(2.11) 和问题(2.13) 都与问题(2.12)等价。

使用定理2.4的记法。我们已经假设 $\kappa_1 = \kappa_2 = \kappa_3 = (\frac{1+\kappa}{2})^{\frac{1}{3}}$ 。由于 $\kappa_3 = (\frac{1+\kappa}{2})^{\frac{1}{3}} \geq \frac{1}{2}$, 根据引理2.5, 我们有 $m_1 + m_2 \geq 0$ 。再根据引理2.6可以得到, 问题(2.11) 和(2.12)等价。

假设 $\dot{s}_m(s), s \in [0, 1]$ 是问题(2.12)的最优解。那么它显然也是问题(2.13)的一个可行解。根据引理2.7, $\alpha_m(s) = \dot{s}_m(s)^2$ 在任意的 $s \in [0, 1]$ 处达到了最大值。由于几何轨迹被假设为非奇异的, 也就是对于任意的 $s \in [0, 1], \|\dot{r}'(s)\|^2 > 0$ 。因此 $\dot{s}_m(s), s \in [0, 1]$ 也是问题(2.13)的唯一的最优解。所以问题(2.12) 和问题(2.13) 是等价的。

2.3.3 严格鲁棒模型的线性化等价模型

当概率常数 $\kappa = 1$, 随机轨迹规划问题(2.10)就变为了严格鲁棒轨迹规划问题, 也就是说, 对于随机变量的任意变化, 得到的最优解都要在约束条件内。

当 $\kappa = 1$, 随机变量 m, c, g 必须是有界的才有意义。因此为了方便起见, 我们可以定义 $F^{-1}(1)$ 为对应随机变量的最小上确界。在这种假设下, 我们有以下更好的结果:

定理 2.9. 如果 $\kappa = 1$, 那么问题(2.10) 和问题(2.13) 有相同的唯一最优解。

证明: 根据定理2.8, 问题(2.11) 和问题(2.13) 是等价的, 因此我们只需要证明问题(2.10) 和问题(2.11) 等价。由于问题(2.10) 和(2.11) 的目标函数是相同的, 我们只需证明它们的约束条件是一样的即可。根据定理2.4, 问题(2.11) 的约束成立可以推出问题(2.10)的约束成立, 因此我们只需证明问题(2.10)的约束成立可以推出问题(2.11)的约束成立即可。

由于 $\kappa = 1$, 只需证明 $\tau_{\min} \leq mb + ca + g \leq \tau_{\max}$ 可以推出 $f(a, b) \leq \tau_{\max}$ 和 $g(a, b) \geq \tau_{\min}$ 。因为对于 m, c, g 的任意的取值, $\tau_{\min} \leq mb + ca + g \leq \tau_{\max}$ 总是成立的, 特别地, 当 $b \geq 0$, m, c, g 分别取值 m_1, c_1, g_1 的时候, 上式仍

然成立。从而我们得到 $m_1 b + c_1 a + g_1 \leq \tau_{\max}$ 。当 $b < 0$, m, c, g 分别取值 $-m_2, c_1, g_1$, 我们得到 $-m_2 b + c_1 a + g_1 \leq \tau_{\max}$ 。这也就是 $f(a, b) \leq \tau_{\max}$ 。相同的方式, 我们也可以证明 $g(a, b) \geq \tau_{\min}$ 。

2.4 数值算法

2.4.1 B样条离散化

在这一节, 通过两步简化, 问题(2.13) 会被离散化为一个线性规划问题。首先, 使用 B样条对函数变量 $a(s)$ 和 $b(s)$ 离散化; 紧接着, 根据一般的优化方法[12], 仅考虑这个问题在有限个点上的约束条件, 从而把问题(2.13)简化为了一个线性规划问题。

假设 K 和 p 是正整数, $s_i = \frac{i}{K-p+2}, i = 1, \dots, K-p+1$ 。基函数 $N_{i,p}(s)$ [49] 是关于路径参数 s 的函数, 定义如下:

$$N_{i,0}(s) = \begin{cases} 1, & s(i) \leq s \leq s(i+1) \\ 0, & \text{其余情形} \end{cases}$$

$$N_{i,k}(s) = \frac{s - s(i)}{s(i+k) - s(i)} N_{i,k-1}(s) + \frac{s(i+k+1) - s}{s(i+k+1) - s(i+1)} N_{i+1,k-1}(s)$$

其中 $i = 0, \dots, K+1, k = 1, \dots, p$, 节点向量 $s \triangleq [s(0), s(1), \dots, s(K+p+2)] = [\underbrace{0, \dots, 0}_{p+1}, s_1, s_2, \dots, s_{K-p+1}, \underbrace{1, \dots, 1}_{p+1}]$ 。基函数 $N_{i,p}(s)$ 的导数计算有如下公式:

$$N_{i,p}(s)' = \frac{p}{s(i+p) - s(i)} N_{i,p-1}(s) - \frac{p}{s(i+p+1) - s(i+1)} N_{i+1,p-1}(s)$$

然后 $a(s)$ 和 $b(s)$ 可以通过基函数 $N_{i,p}(s)$ 在参数域 $s \in [0, 1]$ 中表示为:

$$a(s) \approx \sum_{i=0}^{K+1} \hat{a}_i N_{i,p}(s) \quad (2.14)$$

$$b(s) = \frac{1}{2} a'(s) \approx \frac{1}{2} \sum_{i=0}^{K+1} \hat{a}_i N'_{i,p}(s)$$

其中 $[\hat{a}_1, \dots, \hat{a}_K]^T$ 是要被优化的控制节点, 边界条件是 $\hat{a}_0 = a_0, \hat{a}_{K+1} = a_1$ 。

在[51, 74]中, $a(s)$ 被假设是分段线性的。在本节中我们取 $p = 2$, 这意味着 $a(s)$ 是二次的, $b(s)$ 是线性连续的, 从而可以实现对原始 a 和 b 关于分划 K 更好的近似。

为了保证数值解的稳定性, 我们用如下方式选取限制节点:

$$\bar{s}_j = \begin{cases} \frac{1}{2}s_1, & j = 1 \\ \frac{1}{2}(s_{j-1} + s_j), & j = 2, \dots, K-1 \\ \frac{1}{2}(s_{K-1} + 1), & j = K \end{cases} \quad (2.15)$$

利用前面介绍的知识, 目标函数可以近似为:

$$\int_0^1 \|r'(s)\|^2 a(s) ds \approx \mathbf{B} \mathbf{X} \quad (2.16)$$

其中 $\mathbf{X} = [\hat{a}_1, \dots, \hat{a}_K]$, $\mathbf{B} = [B_1, \dots, B_K]$, $B_i = \sum_{j=1}^K \|r'(\bar{s}_j)\|^2 N_{i,p}(\bar{s}_j) \Delta \bar{s}_j$, $\Delta \bar{s}_j = \frac{1}{K}$ 。

假设第 l 个关节的力矩为 $\tau_l = \mathbf{m}_l b + \mathbf{c}_l a + \mathbf{g}_l$, 记 $\tau_{l,1} = \mathbf{m}_{1l} b + \mathbf{c}_{1l} a + \mathbf{g}_{1l}$, $\tau_{l,2} = -\mathbf{m}_{2l} b + \mathbf{c}_{1l} a + \mathbf{g}_{1l}$, $\tau_{l,3} = -\mathbf{m}_{2l} b - \mathbf{c}_{2l} a - \mathbf{g}_{2l}$, $\tau_{l,4} = \mathbf{m}_{1l} b - \mathbf{c}_{2l} a - \mathbf{g}_{2l}$, 其中下标 l 代表着个向量的第 l 个分量。这些力矩在限制点的值记为 $\hat{\tau}_{l,1}, \hat{\tau}_{l,2}, \hat{\tau}_{l,3}, \hat{\tau}_{l,4}$, 比如, $\hat{\tau}_{l,1} = [\tau_{l,1}(\bar{s}_1), \dots, \tau_{l,1}(\bar{s}_K)]$, 从而我们有:

$$\begin{aligned} \hat{\tau}_{l,1} &= \mathbf{A}_{l1} \mathbf{X} + \mathbf{G}_{l1} \\ \hat{\tau}_{l,2} &= \mathbf{A}_{l2} \mathbf{X} + \mathbf{G}_{l2} \\ \hat{\tau}_{l,3} &= \mathbf{A}_{l3} \mathbf{X} + \mathbf{G}_{l3} \\ \hat{\tau}_{l,4} &= \mathbf{A}_{l4} \mathbf{X} + \mathbf{G}_{l4} \end{aligned} \quad (2.17)$$

其中 $\mathbf{A}_{l1}, \mathbf{A}_{l2}, \mathbf{A}_{l3}, \mathbf{A}_{l4} \in \mathbb{R}^{K \times K}$, $\mathbf{G}_{l1}, \mathbf{G}_{l2}, \mathbf{G}_{l3}, \mathbf{G}_{l4} \in \mathbb{R}^K$, 其具体形式如下:

$$\begin{aligned} \mathbf{A}_{l1j} &= \frac{1}{2} \mathbf{m}_{1l}(\bar{s}_j) N'_{i,p}(\bar{s}_j) + \mathbf{c}_{1l}(\bar{s}_j) N_{i,p}(\bar{s}_j), \\ \mathbf{G}_{l1j} &= \mathbf{g}_{1j}(\bar{s}_j) + \frac{1}{2} \mathbf{m}_{1l}(\bar{s}_j) (a_0 N'_{0,p}(\bar{s}_j) + a_1 N'_{K+1,p}(\bar{s}_j)) + \mathbf{c}_{1l}(\bar{s}_j) (a_0 N_{0,p}(\bar{s}_j) + a_1 N_{K+1,p}(\bar{s}_j)), \\ \mathbf{A}_{l2j} &= -\frac{1}{2} \mathbf{m}_{2l}(\bar{s}_j) N'_{i,p}(\bar{s}_j) + \mathbf{c}_{1l}(\bar{s}_j) N_{i,p}(\bar{s}_j), \\ \mathbf{G}_{l2j} &= \mathbf{g}_{1j}(\bar{s}_j) - \frac{1}{2} \mathbf{m}_{2l}(\bar{s}_j) (a_0 N'_{0,p}(\bar{s}_j) + a_1 N'_{K+1,p}(\bar{s}_j)) + \mathbf{c}_{1l}(\bar{s}_j) (a_0 N_{0,p}(\bar{s}_j) + a_1 N_{K+1,p}(\bar{s}_j)), \end{aligned}$$

$$\begin{aligned}
\mathbf{A}_{l3ji} &= -\frac{1}{2}\mathbf{m}_{2l}(\bar{s}_j)N'_{i,p}(\bar{s}_j) - \mathbf{c}_{2l}(\bar{s}_j)N_{i,p}(\bar{s}_j), \\
\mathbf{G}_{l3j} &= -\mathbf{g}_{2j}(\bar{s}_j) - \frac{1}{2}\mathbf{m}_{2l}(\bar{s}_j)(a_0N'_{0,p}(\bar{s}_j) + a_1N'_{K+1,p}(\bar{s}_j)) - \mathbf{c}_{2l}(\bar{s}_j)(a_0N_{0,p}(\bar{s}_j) + \\
&\quad a_1N_{K+1,p}(\bar{s}_j)), \\
\mathbf{A}_{l4ji} &= \frac{1}{2}\mathbf{m}_{1l}(\bar{s}_j)N'_{i,p}(\bar{s}_j) - \mathbf{c}_{1l}(\bar{s}_j)N_{i,p}(\bar{s}_j), \\
\mathbf{G}_{l4j} &= -\mathbf{g}_{1j}(\bar{s}_j) + \frac{1}{2}\mathbf{m}_{1l}(\bar{s}_j)(a_0N'_{0,p}(\bar{s}_j) + a_1N'_{K+1,p}(\bar{s}_j)) - \mathbf{c}_{1l}(\bar{s}_j)(a_0N_{0,p}(\bar{s}_j) + \\
&\quad a_1N_{K+1,p}(\bar{s}_j)), \\
l &= 1, \dots, n, i, j = 1, \dots, K.
\end{aligned}$$

最终, 问题(2.13) 被简化为如下线性规划问题:

$$\begin{aligned}
&\max_{\mathbf{X}} \mathbf{B}\mathbf{X} & (2.18) \\
&s.t. \quad \hat{a}_0 = a_0, \hat{a}_{K+1} = a_1 \\
&\quad \mathbf{N}_j\mathbf{X} + a_0N_{0,p}(\hat{s}_j) + a_1N_{K+1,p}(\hat{s}_j) \geq 0 & (2.19) \\
&\quad \mathbf{A}_{l1}\mathbf{X} + \mathbf{G}_{l1} \leq \tau_{max,l}\mathbf{I} \\
&\quad \mathbf{A}_{l2}\mathbf{X} + \mathbf{G}_{l2} \leq \tau_{max,l}\mathbf{I} \\
&\quad \mathbf{A}_{l3}\mathbf{X} + \mathbf{G}_{l3} \geq \tau_{min,l}\mathbf{I} \\
&\quad \mathbf{A}_{l4}\mathbf{X} + \mathbf{G}_{l4} \geq \tau_{min,l}\mathbf{I} \\
&\quad j = 1, \dots, K; l = 1, \dots, n
\end{aligned}$$

其中 $\mathbf{I} = [1, \dots, 1]^T \in \mathbb{R}^K$, $\mathbf{N}_j = [N_{1,p}(\hat{s}_j), \dots, N_{K,p}(\hat{s}_j)]$, $\mathbf{B}, \mathbf{A}_{l1}, \mathbf{A}_{l2}, \mathbf{A}_{l3}, \mathbf{A}_{l4}$ 分别来自公式(2.16) 和(2.17)。

定理 2.10. 对于给定的 K , 问题(2.18)在最坏情况下的浮点运算复杂度为 $O(K^{3.5})$ 。

证明: 由于在问题(2.18)的数值算法求解过程中, 我们求解的变量个数和限制数目都为 $O(K)$ 。根据[29]中的算法, 求解这个线性规划问题, 我们需要进行 $O(K)$ 步, 每步需要进行 $O(K^{2.5})$ 次浮点运算, 因此最坏的情况我们需要的浮点运算复杂度为 $O(K^{3.5})$ 。

2.4.2 如何选择最优的分位数

在问题(2.13)中, $\kappa_1, \kappa_2, \kappa_3$ 被假设为 $(\frac{1+\kappa}{2})^{\frac{1}{3}}$ 。在这一节, 我们将解释原因。

我们想选择尽可能小的 $\kappa_1, \kappa_2, \kappa_3$, 这样的话, 当 κ 改变时, 它们才有足够的变化空间去响应 κ 的变化。可以归纳为下面的问题模型:

$$\begin{aligned} \min \max \quad & \{\kappa_1, \kappa_2, \kappa_3\} \\ \text{s.t.} \quad & \kappa_1 \cdot \kappa_0 \geq \frac{1+\kappa}{2} \\ & \kappa_2 \cdot \kappa_3 \geq \kappa_0 \\ & \kappa_0 \kappa_3 \neq 0 \\ & 0 \leq \kappa_0, \kappa_1, \kappa_2, \kappa_3 \leq 1 \end{aligned} \quad (2.20)$$

这个问题中的约束条件来源于定理2.4。

定理 2.11. 问题(2.20) 有唯一的最优解 $(\kappa_1^*, \kappa_2^*, \kappa_3^*) = ((\frac{1+\kappa}{2})^{\frac{1}{3}}, (\frac{1+\kappa}{2})^{\frac{1}{3}}, (\frac{1+\kappa}{2})^{\frac{1}{3}})$ 。

证明: 我们首先证明 $(\kappa_1^*, \kappa_2^*, \kappa_3^*)$ 是最优解。显然 $(\kappa_1^*, \kappa_2^*, \kappa_3^*)$ 满足问题(2.20)的约束($\kappa_0 = (\frac{1+\kappa}{2})^{\frac{2}{3}}$)。如果它不是最优的, 那么必然存在一个解 $(\kappa'_1, \kappa'_2, \kappa'_3)$ 满足 $\min \max\{\kappa'_1, \kappa'_2, \kappa'_3\} < (\frac{1+\kappa}{2})^{\frac{1}{3}}$ 。因而有 $\kappa'_1 < (\frac{1+\kappa}{2})^{\frac{1}{3}}, \kappa'_2 < (\frac{1+\kappa}{2})^{\frac{1}{3}}, \kappa'_3 < (\frac{1+\kappa}{2})^{\frac{1}{3}}, \kappa'_0 \leq \kappa'_2 \kappa'_3 < (\frac{1+\kappa}{2})^{\frac{2}{3}}$, 并且 $\kappa'_0 \kappa'_1 < \frac{1+\kappa}{2}$, 矛盾。

我们现在证明这组解 $(\kappa_1^*, \kappa_2^*, \kappa_3^*)$ 还是唯一的。否则, 存在另一组解 $(\kappa''_1, \kappa''_2, \kappa''_3) \neq (\kappa_1^*, \kappa_2^*, \kappa_3^*)$ 并且 $\min \max\{\kappa''_1, \kappa''_2, \kappa''_3\} = (\frac{1+\kappa}{2})^{\frac{1}{3}}$ 。假设 $\kappa''_1 < (\frac{1+\kappa}{2})^{\frac{1}{3}}$ 。由于 $\kappa''_0 \kappa''_1 \geq \frac{1+\kappa}{2}$, 我们有 $\kappa''_0 > (\frac{1+\kappa}{2})^{\frac{2}{3}}$ 。于是 $\kappa''_2 \kappa''_3 \geq \kappa''_0 > (\frac{1+\kappa}{2})^{\frac{2}{3}}$, 或者 $\kappa''_2 > (\frac{1+\kappa}{2})^{\frac{1}{3}}$ 或者 $\kappa''_3 > (\frac{1+\kappa}{2})^{\frac{1}{3}}$, 矛盾。所以 $\kappa''_1 = (\frac{1+\kappa}{2})^{\frac{1}{3}}$ 并且条件 $\kappa''_0 \kappa''_1 \geq \frac{1+\kappa}{2}$ 就变为 $\kappa''_0 \geq (\frac{1+\kappa}{2})^{\frac{2}{3}}$ 。从而 $\kappa''_2 \kappa''_3 \geq \kappa''_0 \geq (\frac{1+\kappa}{2})^{\frac{2}{3}}$ 。如果 $\kappa''_2 < (\frac{1+\kappa}{2})^{\frac{1}{3}}$, 则 $\kappa''_3 > (\frac{1+\kappa}{2})^{\frac{1}{3}}$, 矛盾。

2.4.3 如何计算分位数

给定一个机器人, 它的随机变量 m, c, g 的分位数可以通过[52]中介绍的方法计算。在这个方法中, 计算分位数仅仅需要有限个随机向量 m, c, g 的样本点。并且, 当增加新的样本点时, 分位数的更新只需要在旧值的基础上利用新增的样本点进行一次迭代即可。下面大概介绍一下这种方法的计算框架:

假设 X 是一个随机变量, F 是概率函数, x^* 是 X 的 κ 分位数。为了计算 x^* , 假设给定了一系列的随机变量 X 的样本点: $Z = [z_1, \dots, z_m]$ 。记 x_1 是我们对 x^* 的最好的估计, 将该值用作迭代算法的初值。接着, 一组对 x^* 进行逼

表 2.1: 机器臂模型参数[46]

部位	质量(kg)	质心到节点的距 离(m)	横轴转动惯 量(kgm ²)	纵轴转动惯 量(kgm ²)	关节间距 离(m)
底座	-	-	5.0	-	0.1
机械臂1	6.6	0.55	0.1	0.6	0.75
机械臂2	4.2	0.6	0.02	0.3	0.75

近的数列计算如下:

$$x_{n+1} = x_n + \frac{1}{n}(\kappa - y_n), n = 1, \dots, m \quad (2.21)$$

其中 $y_n = \begin{cases} 1, & z_n \leq x_n \\ 0, & \text{其余情形} \end{cases}$ 。然后, 我们用 x_{m+1} 作为对 x^* 的最好的估计。这个过程收敛性也在[52]被证明, 这也就是说, $\lim_{m \rightarrow \infty} x_m = x^*$ 。

2.5 仿真实验

为了验证所提方法的有效性, 我们使用一个三自由度的机械臂进行了仿真实验。这个机械臂模型有一个基本的连接和两个臂, 如图2.3所示。该模型在文献[46, 74, 86] 中也被使用。机械臂移动路径是一个在工作空间中的抛物线:

$$\begin{cases} x(s) = 0.5 \\ y(s) = 20s^3 - 30s^2 + 10s \\ z(s) = s - 0.5 \end{cases} \quad (2.22)$$

其中 $s \in [0, 1]$ 是路径的参数。力矩的边界是 $[140, 140, 50] N.m$, 初始条件 a_0 和 a_1 的值为 0, 也就是零初末速度。机械臂模型的参数如表格2.1所示。

在这一节, 所有的数值算法都是在Matlab的环境下运行的, 使用的个人电脑配置为2.8GHz i5 处理器和4GB的内存。

正常情况下该模型的轨迹规划的最优解可以通过求解问题(2.9)得到, 如图2.4所示。我们可以看到问题(2.9)的解显然是“bang-bang-singular”[10]的。

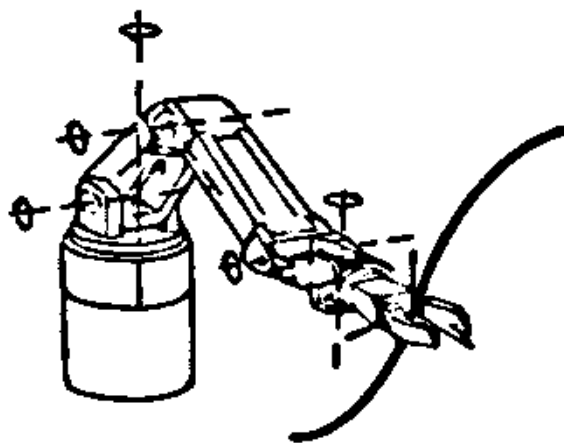
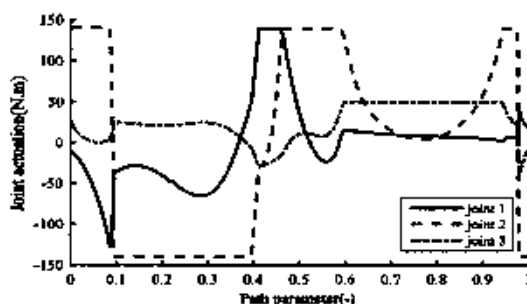


图 2.3: 机械臂及加工路径[46]

也就是说, 对于任意的 $s \in [0, 1]$, 至少有一个力矩在限制条件 $\tau_{\min} \leq \tau(s) \leq \tau_{\max}$ 中达到了它的边界值。这个性质经常被用来验证一个数值解是否是这个问题的最优解以及在(2.14)中的分划 K 是否足够大。在图2.4中, 关节 2, 1, 2, 3和 2 的力矩分别在区间 $[0, 0.4]$, $[0.4, 0.46]$, $[0.46, 0.6]$, $[0.6, 0.94]$, and $[0.94, 1]$ 达到了它们的边界值。这说明我们的数值解法对于 $K = 200$ 是一个有效的解法。

图 2.4: 正常情形下 $K=200$ 时的关节力矩

2.5.1 与其他轨迹规划随机方法的对比

在轨迹规划随机方法[39, 40]中, 作者使用切比雪夫不等式简化在问

表 2.2: 旧方法[39, 40] 与提出的方法遍历时间对比

概率常数 κ		0.6	0.7	0.75	0.8	0.9
遍历时间(s)	旧方法	2.5929	3.5281	7.1461	failed	failed
	提出的方法	1.8734	1.8853	1.8912	1.8970	1.9085

题(2.10)中的概率不等式约束。并且假设 $\tau_{\min} = -\tau_{\max}$ 。记 $E(\tau)$ 是力矩 τ 的期望, $Var(\tau)$ 是其方差。则根据切比雪夫不等式, 概率不等式 $P(\tau_{\min} \leq \tau \leq \tau_{\max}) = P(|\tau| \leq \tau_{\max}) \geq \kappa$ 可以被如下的确定性的不等式代替:

$$1 - \frac{E(\tau^2)}{(\tau_{\max})^2} \geq \kappa, E(\tau^2) \leq (1 - \kappa)\tau_{\max}^2 \quad (2.23)$$

这里记法 τ^2 表示 τ 和它本身的点乘。到这一步, 我们还需要假设 $\kappa < 1$, 因为 $E(\tau^2) > 0$ 。

关于如何计算 $E(\tau^2)$ 的细节在文献[39, 40]没有提及。在下面的对比中, 我们用常规的公式: $E(\tau^2) = E^2(\tau) + Var(\tau)$ 来计算。但即便这样假设, 计算 $E(\tau)$ 和 $Var(\tau)$ 还是非常困难的, 特别是有多个随机变量的时候。因此, 以前的方法看起来只能在一个随机变量的时候有效。为了便于两种方法对比, 考虑如下一种问题:

例子1. 参数 g 是唯一的随机变量, 并且 $g = \bar{g} + \varepsilon$, 其中 \bar{g} 是 g 的参考值, ε 是随机误差, 服从均匀分布, 期望 $E(\varepsilon) = 0$, 方差 $Var(\varepsilon) = \delta \cdot \tau_{\max}^2$, 我们引入 $\delta = 0.01$ 作为一个调节因子。

旧的随机方法[39, 40]和提出方法在 $K = 200$ 条件下加工时间的对比, 如表格2.2所示。从表格中可以看出, 提出方法的加工时间当 $\kappa = 60\%, 70\%, 75\%$ 时, 分别是旧随机方法的 72%, 53%, 26%。当 $\kappa = 80\%$ 或者 90% 时, 由于方法[39, 40]中的放缩太过保守, 以至于计算出来的力矩的界仅仅是 $[61.02 \ 61.02 \ 27.89]N.m$ 和 $[42 \ 42 \ 15]N.m$, 这个界对于移动机器人臂来说太小, 从而没有找到可行解。

例子2. 随机变量 m, c, g 服从均匀分布, 并且自身有 100% 的变化比率。这也就是说 $m \in (0, 2m], c \in (0, 2c], g \in (0, 2g]$ 。变化比率取得这么大是为了放大提出方法的效果。图2.5显示了对应与不同的概率常数 κ 的加工时间。从图中的曲线我们也可以选择合适的概率常数, 从而获得更短的加工时间。比如, 如

果仅仅要求力矩在给定界内的概率大于等于80%，我们的加工时间就仅仅为正常情况下的68%。

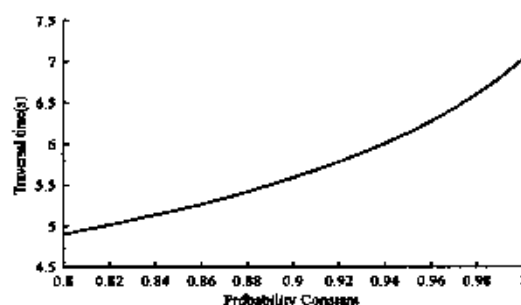


图 2.5: 例2中 $K=200$ 时最短遍历时间与概率常数 κ 的关系

2.5.2 与严格鲁棒轨迹规划方法的对比

在这一节，我们比较了提出的方法（在概率常数 $\kappa = 1$ 的情形）和严格鲁棒轨迹规划方法[86]。

例子3. 假设 $\kappa = 1$ ，随机变量 m, c, g 服从均匀分布并且具有10% 的变化比率。

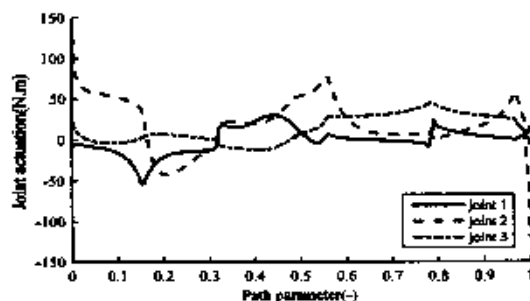


图 2.6: 例3中 $K=200$ 时鲁棒方法的关节力矩

图2.6 显示的是严格鲁棒方法[86]的力矩。图2.7显示的是本文提出的方法的力矩。我们可以发现，提出方法的力矩在绝大多数情况都远远大于前者。这

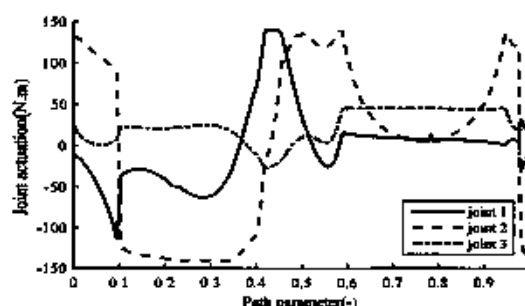
图 2.7: 例2中 $K=200$ 时随机方法的关节力矩

表 2.3: 鲁棒方法[86] 和提出方法最短遍历时间的比较

m, c, g 的变化比率		5%	10%	20%	30%	50%
遍历时间(s)	鲁棒方法[86]	2.1895	2.7573	3.8012	4.7620	6.7039
	提出方法	1.7359	1.8148	1.9822	2.1649	2.5907

意味着加工时间会大大缩短。事实上,通过表格2.3,我们可以发现,提出方法的加工时间(1.8148s)仅仅是严格鲁棒方法加工时间(2.7573s)的66%。

根据定理2.9可知,提出的方法是对严格鲁棒模型的最优解,这也从一方面说明,之前的严格鲁棒方法不是和原问题是等价的,而是一种放缩,带来了加工效率的损失。在表格2.3中,我们也就随机变量不同的变化比率比较了两种方法的加工时间。事实上,正常情况下的加工时间为1.6594s,从表格中可以看出,为了保证力矩在10%干扰的情况下满足约束,加工时间大概是正常情况的1.1倍。

下面我们在例子3中使用“最坏”的随机变量 m, c, g 的值来说明我们提出的方法确实是严格鲁棒的。

图2.8, 图2.9和图2.10显示了上述结果。

以图2.8为例,当计算力矩的时候,我们把随机变量 m, c, g 的取值设为 $1.1m, 1.1c, 1.1g$, 这意味它们达到了它们允许变化的最大值。在这些图片中,我们可以发现,力矩总是严格在边界 $[140, 140, 50]N.m$ 内,并且,在路径的一些区间内,比如在图2.8中的区间 $[0.418, 0.449]$, $[0.4880, 0.512]$, $[0.589, 0.942]$, 在

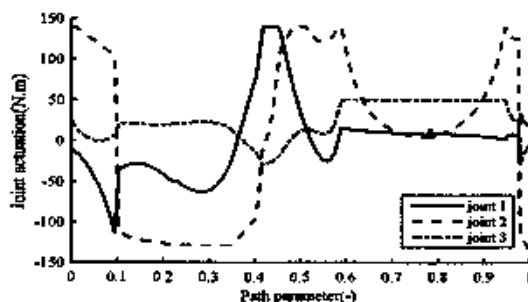


图 2.8: 例3中 $K=200$ 时的关节力矩(m, c, g 分别取值为 $1.1m, 1.1c, 1.1g$)

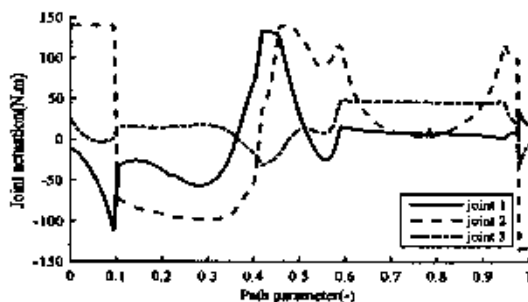


图 2.9: 例3中 $K=200$ 时的关节力矩(m, c, g 分别取值为 $1.1m, 0.9c, 1.1g$)

图2.9中的区间 $[0, 0.095]$, 在图2.10中的区间 $[0.223, 0.343]$ 和 $[0.418, 0.449]$, 某一个力矩正好达到了它的边界值。

2.6 一般化的随机理论

这一节, 我们考虑处理一般化的带 n 个随机变量的概率不等式: $P(B_{\min} \leq m_1x_1 + m_2x_2 + \dots + m_nx_n \leq B_{\max}) \geq \kappa$, 其中 m_1, \dots, m_n 是随机变量, B_{\min}, B_{\max} 是上下界。

为后面叙述方便, 首先定义广义的条件概率如下:

$$P(A|B) = \begin{cases} 0, & \text{如果 } P(B) = 0 \\ \frac{P(A \cap B)}{P(B)}, & \text{如果 } P(B) \neq 0 \end{cases} \quad (2.24)$$

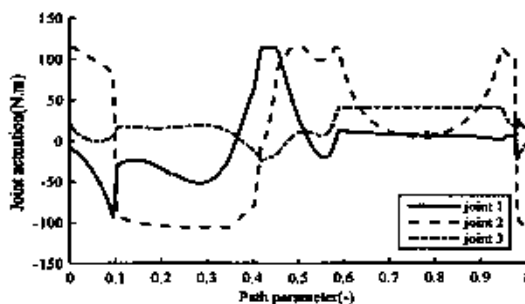


图 2.10: 例3中 $K=200$ 时的关节力矩(m, c, g 分别取值为 $1.1m, 1.1c, 0.9g$)

引理 2.12. 在上面广义条件概率的定义下, 如果 $P(A|B) \geq \kappa_1$, $P(B) \geq \kappa_0$, 那么 $P(A) \geq \kappa_0 \kappa_1$.

证明: 参照引理(2.3), 注意到当 $P(B) = 0$, 我们同样也有 $P(A|B) = 0$, 显然 $P(A) \geq 0$ 也成立.

定理 2.13. 假设 $m_1^{(1)}, m_2^{(1)}, \dots, m_n^{(1)}$ 分别是随机变量 m_1, m_2, \dots, m_n 的 $\kappa_1, \kappa_2, \dots, \kappa_n$ 分位数, $m_1^{(2)}, m_2^{(2)}, \dots, m_n^{(2)}$ 分别是随机变量 $-m_1, -m_2, \dots, -m_n$ 的 $\kappa_1, \kappa_2, \dots, \kappa_n$ 分位数, 其中 $\kappa_1, \kappa_2, \dots, \kappa_n \in [0, 1]$ 并且满足 $\kappa_1 \kappa_2 \dots \kappa_n \geq \frac{1+\delta}{2}$, 记 $f = b_1 x_1 + b_2 x_2 + \dots + b_n x_n, g = a_1 x_1 + a_2 x_2 + \dots + a_n x_n$, 其中

$$b_i = \begin{cases} m_i^{(1)}, & x_i \geq 0 \\ -m_i^{(2)}, & x_i < 0 \end{cases},$$

$$a_i = \begin{cases} -m_i^{(2)}, & x_i \geq 0 \\ m_i^{(1)}, & x_i < 0 \end{cases}$$

那么, $f \leq B_{\max}$ 并且 $g \geq B_{\min}$ 推出 $P(B_{\min} \leq m_1 x_1 + m_2 x_2 + \dots + m_n x_n \leq B_{\max}) \geq \kappa_0$.

证明: 考虑如下的事件:

$$A: m_1 x_1 + m_2 x_2 + \dots + m_n x_n \leq B_{\max}$$

$$B: m_1 x_1 + m_2 x_2 + \dots + m_n x_n \geq B_{\min}$$

首先断言 $P(A) \geq \frac{1+\kappa}{2}$ 并且 $P(B) \geq \frac{1+\kappa}{2}$ 。然后利用引理2.2, 我们就有 $P(B_{\min} \leq m_1x_1 + m_2x_2 + \dots + m_nx_n \leq B_{\max}) \geq \kappa = P(A \cap B) \geq \kappa$, 从而定理就得到证明。现在只需证明断言 $P(A) \geq \frac{1+\kappa}{2}$ 成立, 断言 $P(B) \geq \frac{1+\kappa}{2}$ 可以通过相似的方法证明。下面证明 $f \leq B_{\max}$ 推出 $P(A) \geq \frac{1+\kappa}{2}$ 。

根据定理2.12, 有 $P(b_1x_1 + b_2x_2 + \dots + b_{n-1}x_{n-1} + m_nx_n \leq B_{\max}) \geq P(b_1x_1 + b_2x_2 + \dots + b_{n-1}x_{n-1} + m_nx_n \leq B_{\max} | b_1x_1 + b_2x_2 + \dots + b_{n-1}x_{n-1} + b_nx_n \leq B_{\max}) \cdot P(b_1x_1 + b_2x_2 + \dots + b_{n-1}x_{n-1} + b_nx_n \leq B_{\max}) \geq P(m_nx_n \leq b_nx_n)$, 根据分位数和 b_n 的定义, 当 $x_n > 0$, $P(m_nx_n \leq b_nx_n) = P(m_n \leq m_n^{(1)}) \geq \kappa_n$; 当 $x_n = 0$, $P(m_nx_n \leq b_nx_n) = 1 \geq \kappa_n$; 当 $x_n < 0$, $P(m_nx_n \leq b_nx_n) = P(-m_n \leq m_n^{(2)}) \geq \kappa_n$, 因此, $P(b_1x_1 + b_2x_2 + \dots + b_{n-1}x_{n-1} + m_nx_n \leq B_{\max}) \geq \kappa_n$ 。

同理, $P(b_1x_1 + b_2x_2 + \dots + b_{n-2}x_{n-2} + m_{n-1}x_{n-1} + m_nx_n \leq B_{\max}) \geq P(b_1x_1 + b_2x_2 + \dots + b_{n-2}x_{n-2} + m_{n-1}x_{n-1} + m_nx_n \leq B_{\max} | b_1x_1 + b_2x_2 + \dots + b_{n-1}x_{n-1} + m_nx_n \leq B_{\max}) \cdot P(b_1x_1 + b_2x_2 + \dots + b_{n-1}x_{n-1} + m_nx_n \leq B_{\max}) \geq P(m_{n-1}x_{n-1} \leq b_{n-1}x_{n-1}) \cdot \kappa_n \geq \kappa_{n-1} \cdot \kappa_n$ 。

类似地, 我们就得到 $P(m_1x_1 + m_2x_2 + \dots + m_nx_n \leq B_{\max}) \geq \kappa_1 \cdot \kappa_2 \cdot \dots \cdot \kappa_n \geq \kappa_0$ 。

注4. 在定理2.13中, $f = b_1x_1 + b_2x_2 + \dots + b_nx_n$, $g = a_1x_1 + a_2x_2 + \dots + a_nx_n$ 均为条件函数, 对于给定的 x_1, x_2, \dots, x_n , f, g 有唯一的表达式。对于不定的 x_1, x_2, \dots, x_n , f, g 分别有至多 2^n 种表达式。

注5. 记号同定理2.13, 由引理2.5可知, 如果 $\kappa_i \geq \frac{1}{2}$ 并且 m_i 是连续随机变量, 则 $m_i^{(1)} + m_i^{(2)} \geq 0$ 。

引理2.14. 记号同定理2.13, 如果对于所有的 $i = 1, 2, \dots, n$, m_i 均为连续随机变量并且 $\kappa_i \geq \frac{1}{2}$, 则 $b_ix_i = \max\{m_i^{(1)}x_i, -m_i^{(2)}x_i\}$, $a_ix_i = \min\{m_i^{(1)}x_i, -m_i^{(2)}x_i\}$, 进一步地,

$$\begin{aligned} f(x_1, x_2, \dots, x_n) &= \max_{b'_i \in \{m_i^{(1)}, -m_i^{(2)}\}} \{b'_1x_1 + b'_2x_2 + \dots + b'_nx_n\}, \\ g(x_1, x_2, \dots, x_n) &= \min_{a'_i \in \{m_i^{(1)}, -m_i^{(2)}\}} \{a'_1x_1 + a'_2x_2 + \dots + a'_nx_n\}. \end{aligned}$$

证明: 先证 $b_ix_i = \max\{m_i^{(1)}x_i, -m_i^{(2)}x_i\}$ 。当 $x \geq 0$ 时, $b_ix_i = m_i^{(1)}x_i$, 由于 $m_i^{(1)}x_i - (-m_i^{(2)}x_i) = (m_i^{(1)} + m_i^{(2)}) \cdot x_i \geq 0$, 显然 $b_ix_i = \max\{m_i^{(1)}x_i, -m_i^{(2)}x_i\}$ 。当 $x < 0$ 时, $b_ix_i = -m_i^{(2)}x_i$, 由于 $-m_i^{(2)}x_i - (m_i^{(1)}x_i) = (m_i^{(1)} + m_i^{(2)}) \cdot x_i \leq 0$, 显然 $b_ix_i = \max\{m_i^{(1)}x_i, -m_i^{(2)}x_i\}$ 。

$(-x_i) \geq 0$, 因此也有 $b_i x_i = \max \{m_i^{-1} x_i, -m_i^2 x_i\}$ 。综合上面两种情况, 就有 $b_i x_i = \max \{m_i^{-1} x_i, -m_i^2 x_i\}$ 。类似可以证明 $a_i x_i = \min \{m_i^{-1} x_i, -m_i^2 x_i\}$ 。

由于 $b_i x_i = \max \{m_i^{-1} x_i, -m_i^2 x_i\}$, $f(x_1, x_2, \dots, x_n) = b_1 x_1 + b_2 x_2 + \dots + b_n x_n$, 根据 $f(x_1, x_2, \dots, x_n)$ 定义, $f(x_1, x_2, \dots, x_n)$ 是对每个 $b_i x_i$ 的最大值进行求和, 显然这等价于 $f(x_1, x_2, \dots, x_n) = \max_{b_i' \in \{m_i^{-1}, -m_i^2\}} b_1' x_1 + b_2' x_2 + \dots + b_n' x_n$ 。类似可以证明 $g(x_1, x_2, \dots, x_n) = \min_{a_i' \in \{m_i^{-1}, -m_i^2\}} a_1' x_1 + a_2' x_2 + \dots + a_n' x_n$ 。

定理 2.15. 记号同定理 2.13。假设对于所有的 $i = 1, 2, \dots, n$, m_i 均为连续随机变量并且 $\kappa_i \geq \frac{1}{2}$

则 $f(x_1, x_2, \dots, x_n) \leq B_{\max}$ 等价于 $\bigcap_{b_i' \in \{m_i^{-1}, -m_i^2\}} \{\sum_{i=1}^n b_i' x_i \leq B_{\max}\}$,

$g(x_1, x_2, \dots, x_n) \geq B_{\min}$ 等价于 $\bigcap_{a_i' \in \{m_i^{-1}, -m_i^2\}} \{\sum_{i=1}^n a_i' x_i \geq B_{\min}\}$ 。

证明: 显然 $f(x_1, x_2, \dots, x_n) \leq B_{\max} \Leftrightarrow \bigcap_{b_i' \in \{m_i^{-1}, -m_i^2\}} \{\sum_{i=1}^n b_i' x_i \leq B_{\max}\}$, 只需证明, $f(x_1, x_2, \dots, x_n) \leq B_{\max} \subseteq \bigcap_{b_i' \in \{m_i^{-1}, -m_i^2\}} \{\sum_{i=1}^n b_i' x_i \leq B_{\max}\}$ 。对于任意给定的 x_1, x_2, \dots, x_n , 由于 $f(x_1, x_2, \dots, x_n) = \max_{b_i' \in \{m_i^{-1}, -m_i^2\}} \{b_1' x_1 + b_2' x_2 + \dots + b_n' x_n\} \leq B_{\max}$, 所以显然对于任意的 $b_i' \in \{m_i^{-1}, -m_i^2\}$, 有 $\sum_{i=1}^n b_i' x_i \leq B_{\max}$, 从而得证。类似的方法可以证明 $g(x_1, x_2, \dots, x_n) \geq B_{\min}$ 等价于 $\bigcap_{a_i' \in \{m_i^{-1}, -m_i^2\}} \{\sum_{i=1}^n a_i' x_i \geq B_{\min}\}$ 。

注 6. 上述定理的意义就是把条件函数约束的交 $f(x_1, x_2, \dots, x_n) \leq B_{\max}$, $g(x_1, x_2, \dots, x_n) \geq B_{\min}$ 简化为了一系列的线性函数约束的交:

$$\bigcap_{b_i' \in \{m_i^{-1}, -m_i^2\}} \{\sum_{i=1}^n b_i' x_i \leq B_{\max}\} \text{ 和 } \bigcap_{a_i' \in \{m_i^{-1}, -m_i^2\}} \{\sum_{i=1}^n a_i' x_i \geq B_{\min}\}.$$

注 7. 记号同定理 2.13。我们希望通过如下的方式选取 $\kappa_1, \kappa_2, \dots, \kappa_n$ 使得 $\kappa_1, \kappa_2, \dots, \kappa_n$ 中的最大值最小, 这样当 κ 变化的时候, $\kappa_i, i = 1, 2, \dots, n$ 对它的变化才能够尽量敏感。这也就是求解如下问题:

$$\begin{aligned}
 \min \max \quad & \{\kappa_1, \kappa_2, \dots, \kappa_n\} \\
 \text{s.t.} \quad & \kappa_1 \cdot \kappa_2 \cdot \dots \cdot \kappa_n \geq \frac{1+\kappa}{2} \\
 & \kappa_1, \kappa_2, \dots, \kappa_n \in [0, 1]
 \end{aligned} \tag{2.25}$$

定理 2.16. 上述问题(2.25)有唯一的最优解:

$$(\kappa_1, \kappa_2, \dots, \kappa_n) = ((\frac{1+\kappa}{2})^{\frac{1}{n}}, (\frac{1+\kappa}{2})^{\frac{1}{n}}, \dots, (\frac{1+\kappa}{2})^{\frac{1}{n}}).$$

证明: 反证法: 若存在 $i \in \{1, 2, \dots, n\}$ 满足 $\kappa_i < (\frac{1+\kappa}{2})^{\frac{1}{n}}$, 必存在 $j \in \{1, 2, \dots, n\}, j \neq i$ 满足 $\kappa_j > (\frac{1+\kappa}{2})^{\frac{1}{n}}$. 否则, $\kappa_1 \cdot \kappa_2 \cdot \dots \cdot \kappa_n < \frac{1+\kappa}{2}$, 与约束条件矛盾. 在这种 $\kappa_j > (\frac{1+\kappa}{2})^{\frac{1}{n}}$ 存在的情况下, 显然 $\min \max \{\kappa_1, \kappa_2, \dots, \kappa_n\} > (\frac{1+\kappa}{2})^{\frac{1}{n}}$, 这不是最优解. 从而说明最优解, 必有对 $\forall i \in \{1, 2, \dots, n\}$ 满足 $\kappa_i \geq (\frac{1+\kappa}{2})^{\frac{1}{n}}$. 又 $(\kappa_1, \kappa_2, \dots, \kappa_n) = ((\frac{1+\kappa}{2})^{\frac{1}{n}}, (\frac{1+\kappa}{2})^{\frac{1}{n}}, \dots, (\frac{1+\kappa}{2})^{\frac{1}{n}})$ 是满足条件的一组可行解, 且是最小的, 从而其必是唯一的最优解.

定理 2.17. 记号同定理 2.13. 并且假设对于所有的 $i = 1, 2, \dots, n$, \mathbf{m}_i 均为连续随机变量, $(\kappa_1, \kappa_2, \dots, \kappa_n) = ((\frac{1+\kappa}{2})^{\frac{1}{n}}, (\frac{1+\kappa}{2})^{\frac{1}{n}}, \dots, (\frac{1+\kappa}{2})^{\frac{1}{n}})$, 则

$$\bigcap_{b_i' \in \{m_i^1, -m_i^2\}} \{\sum_{i=1}^n b_i' x_i \leq B_{\max}\} \text{ 和 } \bigcap_{a_i' \in \{m_i^1, -m_i^2\}} \{\sum_{i=1}^n a_i' x_i \geq B_{\min}\} \text{ 推出 } P(B_{\min} \leq \mathbf{m}_1 x_1 + \mathbf{m}_2 x_2 + \dots + \mathbf{m}_n x_n \leq B_{\max}) \geq \kappa_0$$

证明: 显然 $(\kappa_1, \kappa_2, \dots, \kappa_n) = ((\frac{1+\kappa}{2})^{\frac{1}{n}}, (\frac{1+\kappa}{2})^{\frac{1}{n}}, \dots, (\frac{1+\kappa}{2})^{\frac{1}{n}})$ 满足 $\kappa_1 \cdot \kappa_2 \cdot \dots \cdot \kappa_n \geq \frac{1+\kappa}{2}$ 并且对 $\forall i = 1, 2, \dots, n, \kappa_i \geq \frac{1+\kappa}{2}$, 从而满足定理 2.13 和 2.15 的条件, 于是 $f \leq B_{\max}$ 和 $g \geq B_{\min}$ 推出 $P(B_{\min} \leq \mathbf{m}_1 x_1 + \mathbf{m}_2 x_2 + \dots + \mathbf{m}_n x_n \leq B_{\max}) \geq \kappa$, 并且 $f(x_1, x_2, \dots, x_n) \leq B_{\max}$ 等价于 $\bigcap_{b_i' \in \{m_i^1, -m_i^2\}} \{\sum_{i=1}^n b_i' x_i \leq B_{\max}\}$, $g(x_1, x_2, \dots, x_n) \geq$

$$B_{\min} \text{ 等价于 } \bigcap_{a_i' \in \{m_i^1, -m_i^2\}} \{\sum_{i=1}^n a_i' x_i \geq B_{\min}\}, \text{ 所以 } \bigcap_{b_i' \in \{m_i^1, -m_i^2\}} \{\sum_{i=1}^n b_i' x_i \leq B_{\max}\} \text{ 和 } \bigcap_{a_i' \in \{m_i^1, -m_i^2\}} \{\sum_{i=1}^n a_i' x_i \geq B_{\min}\} \text{ 推出 } P(B_{\min} \leq \mathbf{m}_1 x_1 + \mathbf{m}_2 x_2 + \dots + \mathbf{m}_n x_n \leq B_{\max}) \geq \kappa_0$$

2.7 结论

在本章中, 我们考虑了机器人在概率函数约束下的时间最优轨迹规划问题, 并提出一个高效求解算法. 解决的步骤包括三步. 第一步是把这个概率不

等式约束的问题转化为了一个确定性的约束问题,然后第二步对这个确定性问题进行线性化处理,最后一步则是通过 B 样条离散化的方法求解这个问题。

进一步地,当概率函数约束中的概率常数为1的时候,原问题和提出方法简化后的线性规划模型还是等价的,从而说明提出的方法完整的解决了严格鲁棒轨迹规划问题。并且,提出的方法与存在的别的随机方法相比,不仅对原问题更加逼近,并且可以处理多个随机变量的情况,而且在多项式时间内就可以求解,需求仅仅是随机变量的有限个样本点。目前该方法仅仅考虑了力矩的约束。

在第2.6节由该模型出发,把该方法进行了进一步的推广,探讨了处理一般情形下带有 n 个随机变量的概率约束的理论。该理论有很多的应用,比如可以处理带更高阶约束的机器人鲁棒轨迹规划问题,这些高阶约束可以是力矩变化率约束、电压约束[87]等等。

第三章 长加工路径的高效轨迹规划算法

在本章中,提出了一种基于长加工路径速度规划的高效计算方法。该方法通过利用分段速度规划得到的局部最小值点进行循环拼接和更新,从而得到一个与全局最优解很接近的近似解。由于在速度曲线的拼接过程中对精度要求较高,因此本文中讨论了这种方法的鲁棒性,并且讨论了这种快速拼接算法的最坏情况下的浮点计算复杂度。最后本文通过仿真和实验分别对提出的算法进行了验证。实验部分对二维猫形轨迹和三维牙齿形轨迹进行了实际加工。实验证明,提出的方法不仅精度与最优解非常接近,并且极大地提高了计算效率。

本章结构如下:第一节介绍了数控系统轨迹规划的一般模型;第二节对得到的非凸非线性模型进行了线性化处理;第三节提出了长加工路径速度曲线的拼接算法;第四节介绍了进行求解的数值计算方法;仿真和实验放在了第五节,最后一节则对该章进行了总结。

3.1 长加工路径轨迹规划问题及线性化方法

3.1.1 长加工路径轨迹规划问题

假设加工路径为:

$$\mathbf{r}(u) = (x(u), y(u), z(u)) \quad (3.1)$$

其中 u 是路径参数,满足 $0 \leq u \leq 1$ 。

考虑如下问题:

在速度、加速度和加加速度有界的条件下,求速度规划使得加工时间最短,待求变量为参数速度 $q = \dot{u}^2$ 。

下面为该问题的数学模型:

$$\min_q t = \int_0^T 1 dt = \int_0^1 \frac{1}{\sqrt{q}} du \quad (3.2)$$

约束条件为:

$$\begin{aligned} |v_r| = |\dot{\mathbf{r}}| &= |\tau' \sqrt{q}| \leq V_{\max} \\ |a_r| = |\ddot{\mathbf{r}}| &= |\tau'' q + \frac{1}{2} \tau' q'| \leq A_{\max} \\ |j_r| = |\ddot{\mathbf{r}}| &= |\tau''' q + \frac{3}{2} \tau'' q' + \frac{1}{2} \tau' q''| \sqrt{q} \leq J_{\max} \end{aligned} \quad (3.3)$$

其中 $\dot{\tau} = \frac{d(\tau)}{dt}$, $\tau \in \{x, y, z\}, v_\tau, a_\tau, j_\tau$ 分别代表参数速度、参数加速度以及参数加加速度, $V_{\max}, A_{\max}, J_{\max}$ 分别为速度、加速度和加加速度的上界。对于二维情形, 问题模型可以类似推出。

3.1.2 线性化方法

首先是简化约束条件, 上述问题约束中加加速度 j_τ 通过文献[23, 86]中提到的方法用如下的线性函数来简化, 也就变为了所谓的“pseudo-jerk”:

$$|j_\tau| = |\ddot{\tau}| = |\tau'''q + \frac{3}{2}\tau''q' + \frac{1}{2}\tau'q''|\sqrt{q^*} \leq J_{\max} \quad (3.4)$$

其中 q^* 是如下问题的最优解:

$$\min_q t = \int_0^T 1 dt = \int_0^1 \frac{1}{\sqrt{q}} du \quad (3.5)$$

约束条件为:

$$\begin{aligned} |v_\tau| = |\dot{\tau}| = |\tau'\sqrt{q}| &\leq V_{\max} \\ |a_\tau| = |\ddot{\tau}| = |\tau''q + \frac{1}{2}\tau'q'| &\leq A_{\max} \end{aligned} \quad (3.6)$$

其中 $\dot{\tau} = \frac{d(\tau)}{dt}$, $\tau \in \{x, y, z\}, v_\tau, a_\tau$ 分别代表参数速度、参数加速度, V_{\max}, A_{\max} 分别为速度和加速度的上界。

根据[23, 86], 式(3.4)的成立可以保证式(3.3)中 j_τ 约束的成立。

其次是简化目标函数。[23]中已经使用如下最大化参数速度代替原来的目标函数:

$$\max_q Q = \int_0^1 q du \quad (3.7)$$

尽管[23]中没有给出严格的数学证明, 但从多次实验结果上看(随机产生三次 B 样条路径, 使用最小时间和最大速度分别作为目标函数, 约束条件均为速度、加速度以及加加速度约束地情况下比较解的一致性), 式(3.2)产生的解与最大化参数速度情况下产生的解完全相同, 尚未发现不一致的地方, 这说明至少实验上来看是可行的。

3.2 长加工路径的高效轨迹规划算法

在实际的加工过程中, 当加工路径很长时, 由于长加工路径带来的各种问题(比如限制点太多, 插补点存储超过机床硬件限制, 求解复杂度高等), 工业

常常分段加工[22]。在每段的速度规划中，常使用零初末速度初末加速度作为约束条件，这不可避免的造成效率损失。并且频繁的从速度为零启动机床，会导致加工质量的降低。

3.2.1 分段高效拼接算法

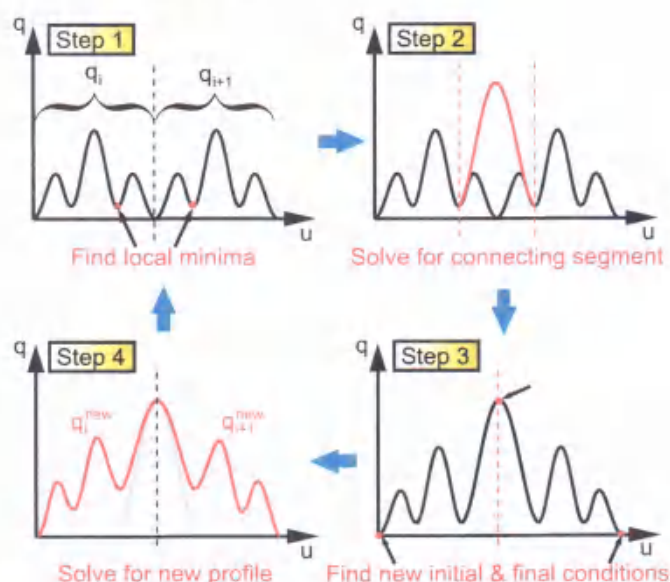


图 3.1: 长加工路径轨迹规划的可视化算法

如图3.1所示，在准备阶段，首先把一条长加工路径分成 N 段，每一段用上面介绍的线性化方法以零初末速度加速度边界条件进行求解，之后循环：

第一步：在段与段之间的连接点附近找到最接近的两个极小值点 (q_i, q_{i+1}) ，如图3.1所示的那样。

第二步：以所得的极小值点 (q_i, q_{i+1}) 作为边界条件，用线性化方法求出它们之间连接部分的速度。

第三步：找到新形成的每段的速度曲线的初末速度加速度，并以它们作为新的边界条件。

第四步：利用第三步中得到的新的边界条件对每一段的速度曲线重新进行线性化方法生成。

上面步骤一直循环,直到第四步得到的速度曲线和上一次循环中得到的速度曲线的差别小于一个阈值停止。这个阈值是由操作者设定的,实验上来看,一般上述步骤只需要循环一次就能得到比较满意的结果。

3.2.2 B样条离散化方法

假设 $0 = u_1 < u_2 < \dots < u_{k-1} < u_k = 1$ 是离散化后的参数域的控制点,并且记

$$s_{i-1} = \frac{u_i + u_{i+1}}{2}, i = 2, 3, \dots, K-2 \quad (3.8)$$

$$du_j = u_{j+1} - u_j, j = 1, 2, \dots, K-1 \quad (3.9)$$

定义 B 样条的节点向量如下:

$$sn = [0, 0, 0, s_1, s_2, \dots, s_{k-3}, 1, 1, 1] \quad (3.10)$$

利用 sn 我们通过 De Boor 算法计算 B 样条函数的基函数:

$$N_{i,0}(s) = \begin{cases} 1, & s_n(i) \leq s \leq s_n(i+1) \\ 0, & \text{其余情形} \end{cases}$$

$$N_{i,p}(s) = \frac{s - sn(i)}{sn(i+p) - sn(i)} N_{i,p-1}(s) + \frac{sn(i+p+1) - s}{sn(i+p+1) - sn(i+1)} N_{i+1,p-1}(s) \quad (3.11)$$

其中 $p = 1, 2, i = 1, 2, \dots, K$ 。

对于我们的问题,我们假设

$$q(u) = \sum_{i=1}^k N_{i,2}(u) a_i \quad (3.12)$$

上面的 a_i 就是我们要求的变量,表示 B 样条函数的控制节点。利用式(3.12),最终的线性离散化问题如下:

$$\max_{a_i} Q = \int_0^1 q du \approx \sum_{i=1}^k \sum_{j=1}^{k-1} N_{i,2}(u_j) du_j a_i$$

约束条件为:

$$\begin{cases} \tau'^2(u_j) \sum_{i=1}^k N_{i,2}(u_j) a_i \leq V_{\max}^2, \\ \pm(\tau''(u_j) \sum_{i=1}^k N_{i,2}(u_j) + \frac{1}{2}\tau'(u_j) \sum_{i=1}^k N_{i,2}'(u_j)) a_i \leq A_{\max}, \\ \pm(\tau'''(u_j) \sum_{i=1}^k N_{i,2}(u_j) + \frac{3}{2}\tau''(u_j) \sum_{i=1}^k N_{i,2}'(u_j) + \frac{1}{2}\tau'(u_j) \sum_{i=1}^k N_{i,2}''(u_j)) \cdot \\ \sqrt{q^*(u_j)} a_i \leq J_{\max} \end{cases} \quad (3.13)$$

其中 $j = 1, 2, \dots, K$ 。

3.2.3 复杂度分析

在本章中, 解决线性规划问题使用的是普遍的内点法。在最坏的情况下, 整体规划方法[23, 86]的浮点计算复杂度为 $O(N^{3.5})$ [29], 其中 N 为限制点的个数。提出的方法在最坏情况下的浮点计算复杂度则为 $O(k(\frac{N}{k})^{3.5})$, 其中 k 为长加工路径分的段数。当 k 与 N 可比较的时候, 比如 $k = \frac{N}{100}$, 则提出方法的计算复杂度就变为了 $O(N)$, 这对于提高计算效率来说意义重大。

3.3 仿真与实验

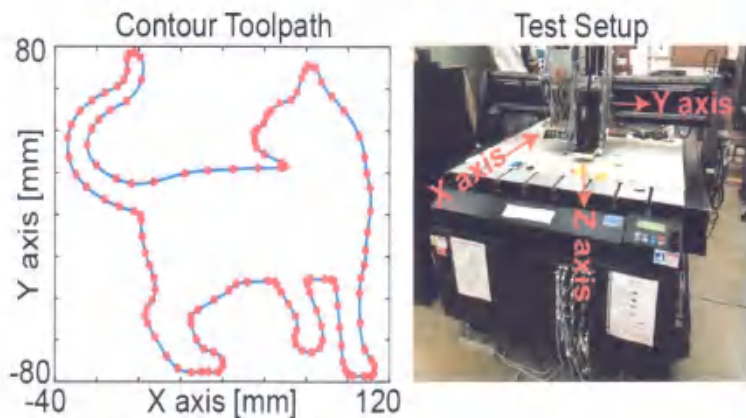


图 3.2: 二维路径和测试机床

本章提出的算法在Matlab环境下进行仿真, 并通过如图3.2的机床进行测试。个人计算机系统Win 10, 处理器为i5 2.6G, 内存为8 GB。测试机床三个轴

的速度加速度以及加加速度的界是相同的, 分别为 $V_{max} = 100\text{mm/s}$, $A_{max} = 500\text{mm/s}^2$ 和 $J_{max} = 10000\text{mm/s}^3$ 。实验分别对二维的猫形路径和三维的牙齿形路径进行了加工。

3.3.1 二维猫形路径实验

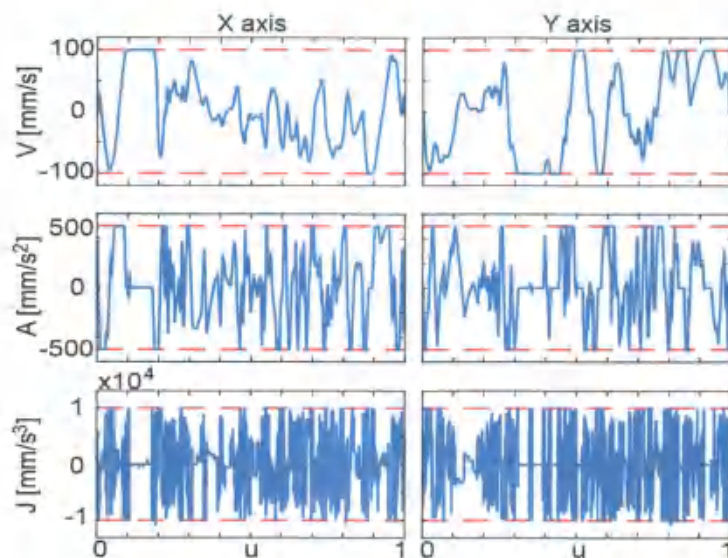


图 3.3: 二维猫形路径的速度、加速度以及加加速度

实验路径如图3.2所示, 轨迹为一个穿过给定离散点的三次 B 样条曲线。图3.3显示了提出算法, 求得的解的速度、加速度以及加加速度的大小。

首先可以观察到, 提出算法解的速度加速度以及加加速度基本都在要求的界内, 是可行解(图中红色的虚线代表上下界)。图3.4显示了提出方法和常数速度方法在跟踪误差和控制信号上的对比。其中常数速度轨迹规划方法是工业常用的一种方法, 就是对整段路径, 在首末位置使用允许的最大加速度加加速度, 中间部分则使用固定的速度(一般很小的速度)进行轨迹规划的方法。可以看到, 两者在跟踪误差和控制信号大小上相差不大。图3.5 则显示了在一次循环中, 提出方法(把整条线段分成10段, 每段300个限制点)和整体规划方法[86](把整段分成相同的3000 个限制点)结果的对比。

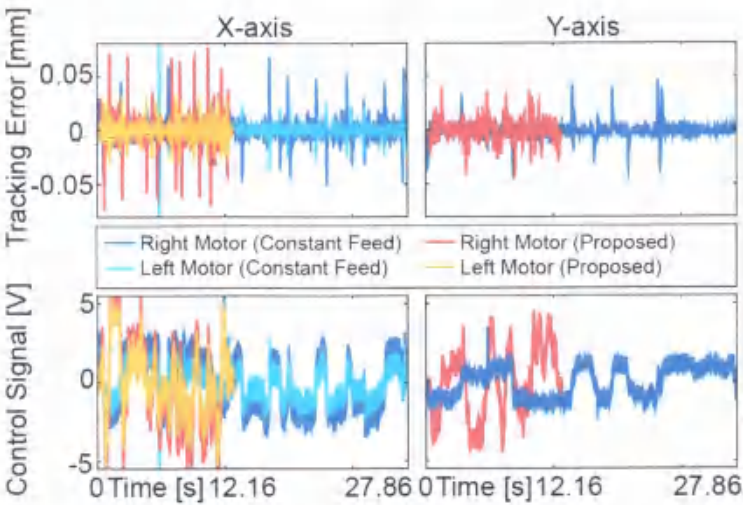


图 3.4: 二维猫形路径的跟踪误差、控制信号及与常数速度方法的对比

表 3.1: 不同算法表现对比

求解策略	计算时间(s)	加工时间(s)
提出方法	15.90	12.16
整体规划方法	27.55	12.15
常数速度方法	2.44	27.86

可以看到两种轨迹规划方法的解非常的近似，最大百分比误差在最大处也没有超过0.5%，表格3.1也验证它们的相似性。提出方法和整体规划方法的加工时间分别为12.16s、12.15s，相差仅为0.08%，但提出方法的计算时间确减少了56%。实际上，当限制点更多的话，两者的差距会更加明显。

3.3.2 三维牙齿形路径实验

在本小节，一个三维牙齿形路径如图3.6所示，被用来测试提出的算法。由于测试机床的限制，为了加工的方便，原来的牙齿被放大了30倍。这条路径被分成了1000段，每段取100个限制点，所以总共该问题的限制点的个数为100000。

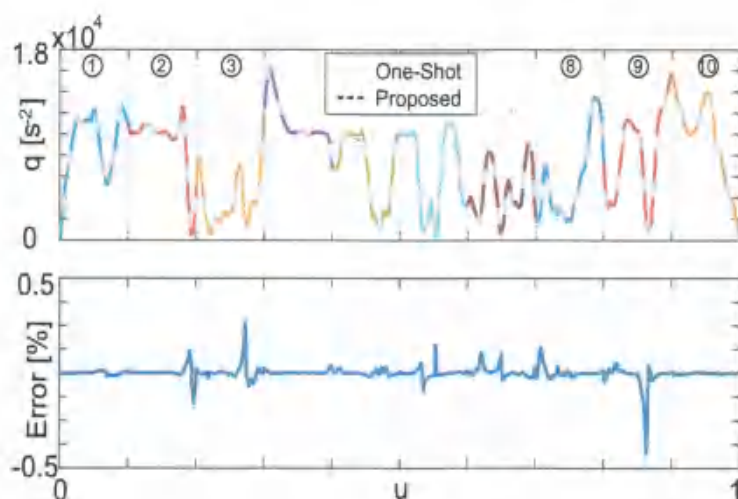


图 3.5: 提出方法与整体规划方法的对比

提出的长加工路径规划方法通过一次循环算法解决这个问题的计算时间为970.36s, 而整体规划方法[86]则无法求解这个问题。这主要由于这样数量的限制点, 整体规划方法最终求解形成的线性规划模型的约束矩阵维数为 $R^{m \times n}$, $m = 16K$, $n = K$, 其中 $K = 100000$ 是这个问题的限制点个数, 这样的线性规划问题一般的个人计算机无法求解(本人计算机计算上限大概为 $K = 5000$ 左右)。图片3.7、图片3.8分别显示了提出方法的跟踪误差、控制信号以及速度、加速度和加加速度。可以看到提出方法的速度、加速度和加加速度都在给定的界内, 并且在很多的地方, 三者之一都达到了边界值, 这从一方面也说明了得到的解和问题最优解应该很接近。而控制信号和跟踪误差与二维猫形路径的控制信号和跟踪误差对比来看, 也是合理的。

3.4 鲁棒性分析

在提出的长加工路径循环算法中, 从理论上讲, 在第二步和第四步的计算中, 总是存在着一个可行解。比如它们上一步的解就分别是它们的可行解。但是在实际中, 由于在求解线性规划问题的过程中总是存在数值误差, 有可能求得的数值解在速度、加速度或者加加速度轻微的超界(比如超出 $1e-8$), 从而会轻微地超过问题的最优解, 如图3.9所示, 但这就可能选择出来的局部极小

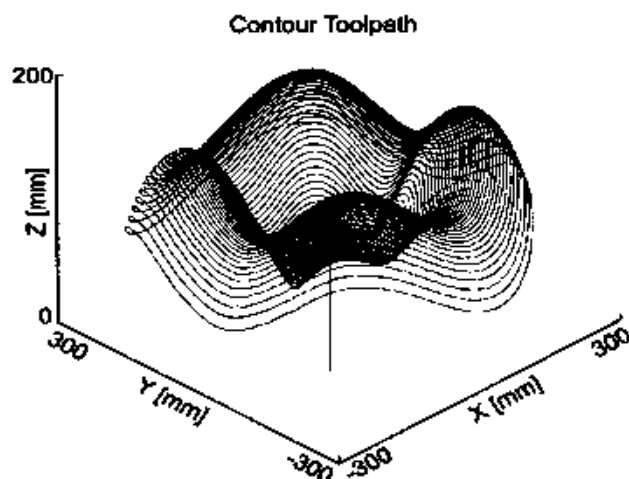


图 3.6: 三维牙齿形路径

值点不在可行域内，从而导致下一步的计算失败。我们提出两个方法来尽量避免这个现象。

(1) 逐步松弛方法

对于长加工路径规划循环算法中的每一步，我们都使在这一步计算中用到的边界条件比上一步大一些。比如，在猫形路径的实验中，准备阶段，第二步和第四步的边界条件分别为：

$$[0.99V_{max}, 0.98A_{max}, 0.96J_{max}], [0.995V_{max}, 0.99A_{max}, 0.98J_{max}], [V_{max}, A_{max}, J_{max}]。$$

(2) 适度加密方法

通过实验结果发现，当限制点取得越多，计算引起的数值误差就会越小。因此，可以在特定的位置，比如选取局部极小值点的地方增加限制点的个数，从而可以降低它们落在非可行域内的概率。

总的来说，逐步松弛方法会损失部分加工时间，但不会影响计算时间。而适度加密的方法会增加计算时间，但对最终的加工时间几乎没有影响。

3.5 结论

本章提出了一种长加工路径高效轨迹规划算法，这种算法的优势就是在保

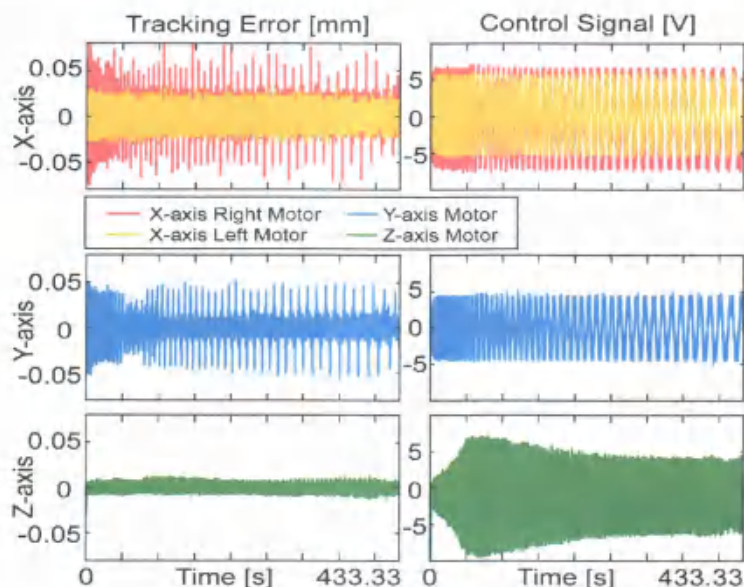


图 3.7: 三维牙齿形路径的跟踪误差以及控制信号

证和“全局最优解”足够接近($< 0.1\%$)的情况下, 计算时间更快(理论浮点算法复杂度减少大概两个数量级), 从而可以在线性时间内求得长加工路径轨迹规划问题的解。

提出的算法是利用分段求解, 然后进行拼接、组合、更新的思想。由于在该过程中对数值精度要求较高, 因此本文还讨论了这种方法的鲁棒性, 并提出逐步松弛和适度加密的方法来提高鲁棒性。

本章最后利用仿真和实验, 在三轴机床上分别对二维和三维路径进行加工, 从加工时间、计算时间、跟踪误差、加工电压等各个方面对提出算法与全局算法进行了对比, 验证了提出算法有效性和极高的效率。

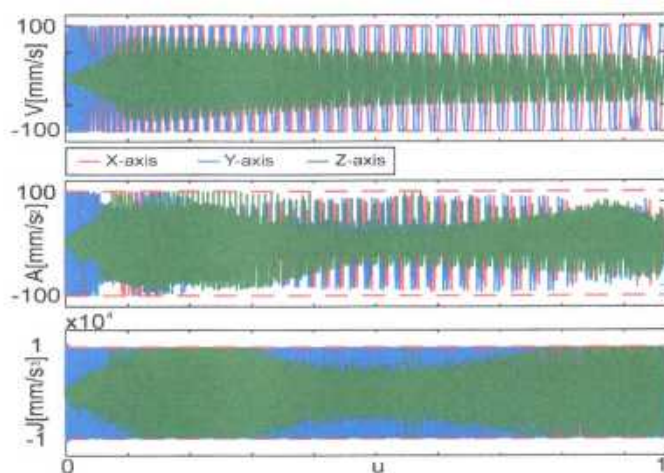


图 3.8: 三维牙齿形路径的速度、加速度以及加加速度

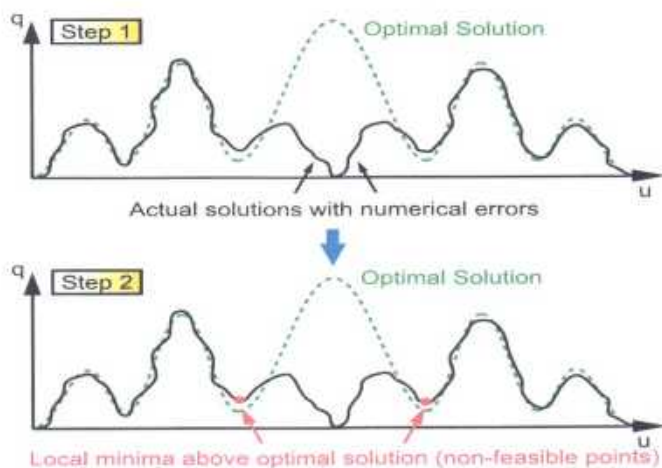


图 3.9: 由数值误差引起的拼接不成功的情形

此页不缺内容



第四章 机器人在多点打孔和点焊过程中的轨迹规划算法

在本章中，提出了一种针对机器人多点打孔、点焊任务的轨迹规划方法。该方法通过优化穿过每个点的次序，以及规划点与点之间的路径，得到最短时间的多点路径规划算法，从而最大化的利用了机器人系统的动力学性能。由于多点打孔、点焊任务是一个走-停的运动，因此可以归结为一个旅行商问题(TSP)和一系列的点到点的路径未知的轨迹规划问题。三次的Hermite插值多项式被用来表示点到点的路径，通过优化点到点的最短时间路径的问题来确定它的参数。然后通过使用点到点的时间路径构建了一种新的TSP模型，并通过经典的遗传算法(GA)进行求解。最后，使用三自由度的机械臂对该方法进行了仿真实验，从而说明了提出方法的有效性。

本章结构如下：第一节建立了该问题的数学模型；第二节对点到点的路径规划子问题利用Hermite差值多项式提出了求解方法，并得到了点到点之间的转移时间(度量矩阵元素)。第三节利用上一节得到的度量矩阵对于多点到多点的轨迹规划问题建立了改良的TSP模型，并利用遗传算法进行了求解。接着在第四节对提出的算法进行了仿真，并与最短路径策略进行了对比。最后一节总结了该章。

4.1 问题描述

在实际生产中，经常使用六自由度的机器人。其中三个自由度用来确定位置，三个自由度用来定向。一般来说，前三个关节用来确定位置，后三个关节通过协同操作用来确定方向。

在本章中，我们主要关注机器人的位置优化，因为方向的确定会在位置确定以后自动控制。因此，下面我们主要讨论前三个关节。假设机器人的前三个关节的动力学模型有如下表示：

$$\tau = M(q)\ddot{q} + \dot{q}^T C(q)\dot{q} + G(q) \quad (4.1)$$

其中 $\tau \in \mathbb{R}^n$ 是力矩向量， $q \in \mathbb{R}^n$ 是关节的角位移， $M(q) \in \mathbb{R}^{n \times n}$ 是机器人的惯性矩阵， $C(q) \in (\mathbb{R}^n)^{n \times n}$ 作为一个矩阵，其元素属于 \mathbb{R}^n ，代表着科氏力和离心力， $G(q) \in \mathbb{R}^n$ 是重力项，这里 $n = 3$ 。

我们的目标就是找到一条最优的路径并规划在上面的速度，沿着这条路径和给定速度，机器人可以在满足运动学和动力学约束的前提下，一次性穿过所有的点仅仅一次，完成打孔、点焊等任务，而总的时间最短。如图4.1所示。

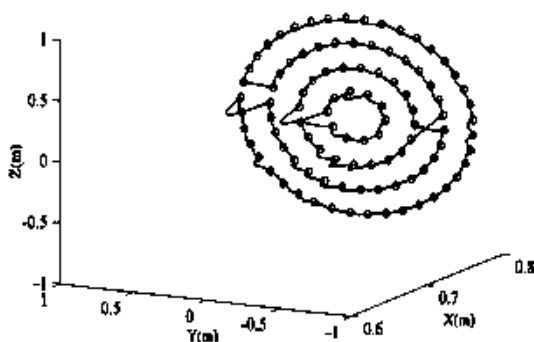


图 4.1: 多点打孔、点焊任务示例

假设加工任务的点的数目用 n_c 表示， $p_1, p_2, \dots, p_{n_c} \in \mathbb{R}^3$ 是对应于加工任务中 n_c 个点的位置。每个关节的力矩约束为：

$$-\tau_B \leq \tau \leq \tau_B \quad (4.2)$$

速度约束为：

$$-\dot{q}_B \leq \dot{q} \leq \dot{q}_B \quad (4.3)$$

其中，位置向量 p 和角位移向量 q 还满足如下关系：

$$\dot{p} = J(q) \dot{q} \quad (4.4)$$

其中， $J(q)$ 表示机器人运动的雅克比矩阵。

由于机器人需要在点所在的位置保持静止，所以我们有 $\dot{q}_i = 0$ ，其中 $i = 1, 2, \dots, n_c$ 。总而言之，机器人在多点打孔和点焊过程中的轨迹规划问题如

下:

$$\begin{aligned} & \min_{\mathbf{q}(t)} T_i \\ \text{s.t.} & \begin{cases} \mathbf{q}(t_i) - \mathbf{Q}\mathbf{W}_i = 0 \\ \dot{\mathbf{q}}(t_i) = 0, i = 1, 2, \dots, n_c \\ \sum_{i=1}^{n_c} \mathbf{W}_i = [1, 1, \dots, 1]_{n_c}^T \\ \tau = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) \\ -\tau_B \leq \tau \leq \tau_B, -\dot{\mathbf{q}}_B \leq \dot{\mathbf{q}} \leq \dot{\mathbf{q}}_B \\ \mathbf{q} \in \Omega_q \end{cases} \quad (4.5) \end{aligned}$$

其中 $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{n_c}]_{n \times n_c}$ 表示所有关节在所有加工点处的角位移信息, $\mathbf{W} \in \mathbb{Z}^{n_c \times 1}$, $\mathbf{W}_i \in \{0, 1\}^{n_c}$ 是单位向量, Ω_q 表示关节角位移的几何限制, $0 = t_1 < t_2 < \dots < t_{n_c} = T_i$.

问题(4.5)是一个典型的混合整数规划问题。根据文献[21], 由于关节的速度要在每一个点处降为零, 因此这个问题可以被分解为一个最短时间TSP问题和一系列的点到点的关于连续变量的未知路径轨迹规划问题。

在本章中, 每一个点到点的未知路径规划问题都会通过一个直接参数化的方法计算出最短的转移时间 T_{ij} , 然后利用全部的最短转移时间构建出最短时间的改良TSP问题, 最后这个问题通过遗传算法(GA)进行求解。

4.2 子问题: 求解点到点的路径以及最短转移时间

点到点的最短时间路径规划问题的模型如下:

$$\begin{aligned} & \min_{\mathbf{q}(t)} T_{ij} \\ \text{s.t.} & \begin{cases} \tau = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) \\ -\tau_B \leq \tau \leq \tau_B, -\dot{\mathbf{q}}_B \leq \dot{\mathbf{q}} \leq \dot{\mathbf{q}}_B \\ \dot{\mathbf{q}}(0) = 0, \dot{\mathbf{q}}(T_{ij}) = 0 \\ \mathbf{q}(0) = \mathbf{q}_i, \mathbf{q}(T_{ij}) = \mathbf{q}_j \\ \mathbf{q} \in \Omega_q \end{cases} \quad (4.6) \end{aligned}$$

为了产生光滑的路径, 我们使用三次Hermite样条来近似加工路径。假设路径参数 $s \in [0, 1]$, 则从点 i 到点 j 的路径可以用如下方式描述:

$$\mathbf{q}(s) = h_{00}(s)\mathbf{q}_i + h_{10}(s)\dot{\mathbf{r}}_i + h_{01}(s)\mathbf{q}_j + h_{11}(s)\dot{\mathbf{r}}_j, s \in [0, 1] \quad (4.7)$$

其中 r_i 表示路径开始的斜率, r_j 表示路径最后的斜率。Hermite 样条的基函数如下:

$$\begin{aligned} h_{00} &= 2s^3 - 3s^2 + 1 \\ h_{01} &= -2s^3 + 3s^2 \\ h_{10} &= s^3 - 2s^2 + s \\ h_{11} &= s^3 - s^2 \end{aligned} \quad (4.8)$$

通过公式(4.7), (4.8), 路径关于参数 s 的导数可以计算如下:

$$\mathbf{q}'(s) = \frac{d\mathbf{q}}{ds} = h'_{00}(s) \mathbf{p}_i + h'_{10}(s) \mathbf{r}_i + h'_{01}(s) \mathbf{p}_j + h'_{11}(s) \mathbf{r}_j \quad (4.9)$$

$$\mathbf{q}''(s) = \frac{d^2\mathbf{q}}{ds^2} = h''_{00}(s) \mathbf{p}_i + h''_{10}(s) \mathbf{r}_i + h''_{01}(s) \mathbf{p}_j + h''_{11}(s) \mathbf{r}_j \quad (4.10)$$

关节的速度为:

$$\dot{\mathbf{q}}(t) = \mathbf{q}'(s(t)) \dot{s}(t) \quad (4.11)$$

关节的力矩为:

$$\tau = \mathbf{m}(s) \ddot{s} + \mathbf{c}(s) \dot{s}^2 + \mathbf{g}(s) \quad (4.12)$$

其中,

$$\begin{aligned} \mathbf{m}(s) &= \mathbf{M}(\mathbf{q}(s)) \mathbf{q}'(s) \in \mathbb{R}^n \\ \mathbf{c}(s) &= \mathbf{M}(\mathbf{q}(s)) \mathbf{q}''(s) + \mathbf{C}(\mathbf{q}(s), \mathbf{q}'(s)) \mathbf{q}'(s) \in \mathbb{R}^n \\ \mathbf{g}(s) &= \mathbf{G}(\mathbf{q}(s)) \in \mathbb{R}^n \end{aligned}$$

通过上述转化, 优化问题(4.6)的优化变量就变为了路径的斜率 r_i, r_j 和参数加速度 $\ddot{s}(t)$ 。记 $a = \ddot{s}, b = \dot{s}, \mathbf{u} = [r_i^T, r_j^T]^T$, 以它们作为未知量, 则优化问题(4.6)就可以写成如下问题:

$$\begin{aligned} \min_{(\mathbf{u}, b)} \quad & T_{ij} = \int_0^1 \frac{1}{\sqrt{a}} ds \\ \text{s.t.} \quad & \begin{cases} \tau(s) = \mathbf{m}(\mathbf{u}, s) b + \mathbf{c}(\mathbf{u}, s) a + \mathbf{g}(\mathbf{u}, s) \\ -\tau_B \leq \tau(s) \leq \tau_B, -\dot{\mathbf{q}}_B \leq \mathbf{q}'(\mathbf{u}, s) \sqrt{a} \leq \dot{\mathbf{q}}_B \\ \dot{\mathbf{q}}(\mathbf{u}, 0) = 0, \dot{\mathbf{q}}(\mathbf{u}, 1) = 0 \\ \mathbf{q}(\mathbf{u}, 0) = \mathbf{p}_i, \mathbf{q}(\mathbf{u}, 1) = \mathbf{p}_j \\ \mathbf{q}(\mathbf{u}, s) \in \Omega_q \end{cases} \end{aligned} \quad (4.13)$$

一般来说,上述问题(4.13)可以近似为一个非线性规划(NLP)问题,并且通过常用的直接参数化方法求解,比如最近向量方法(CVP)、联立方法等等,最终通过序列二次规划(SQP)、拟牛顿法(BFGS)等解出。然而,通过等式(4.12),我们观察到问题(4.13)实际上强非线性的,因为路径参数 u 和优化的目标函数没有清晰的关系。另一方面,运动的变量 (a, b) 在约束条件中是线性的,在目标函数中还是凸的。基于这一点,文献[9, 59]提出了一个合理的嵌套优化策略(Nest Optimization Strategy)。它首先假设优化问题(4.13)中的路径变量 u 开始是固定的,这样该问题就变为了一个凸优化问题。然后对于不同的 u , 都可以分别解出一个最优的优化速度 (a^*, b^*) 和最短的转移时间 T_{ij}^* 。这样,最后的点到点的轨迹规划问题就进一步地被描述为下面的优化问题:

$$\min_u (T_{ij}^*), \text{ s.t. } q(u, s) \in \Omega_q \quad (4.14)$$

图4.2显示了两部嵌套优化策略的计算过程。其中 λ 是在每一步搜索过程中的步长, d 是根据梯度的负方向计算出的搜索方向。在实际计算中,问题(4.14)可以通过存在的NLP问题求解包计算,比如SQP, BFGS等等。

4.3 主问题: 多点的最短时间轨迹规划

多点打孔、点焊过程中的多点最短时间轨迹规划问题可以归结为一个改良TSP问题。定义矩阵 D 是改良TSP问题的度量矩阵。对于一般的最短路径策略,对于任意的 i, j , 有 $D_{ij} = D_{ji}$ 。这也就是说从点 i 到点 j 的距离与从点 j 到点 i 的距离相等。但对于最短时间策略,度量矩阵 D 中的元素代表是转移时间,也就是 $D_{ij} = T_{ij}^*$ 。根据[9, 21], 等式 $D_{ij} = D_{ji}$ 在绝大多数情况下是不成立的。

$$\begin{aligned} \min \quad & T_t = \sum_{i=1}^{n_c} \sum_{j=1}^{n_c} W_{ij} D_{ij} \\ \text{s.t.} \quad & \begin{cases} \sum_{i=1}^{n_c} W_{ij} = 1, j = 1, 2, \dots, n_c \\ \sum_{j=1}^{n_c} W_{ij} = 1, i = 1, 2, \dots, n_c \\ W_{ij} \in \{0, 1\}, D_{ii} = 0 \end{cases} \end{aligned} \quad (4.15)$$

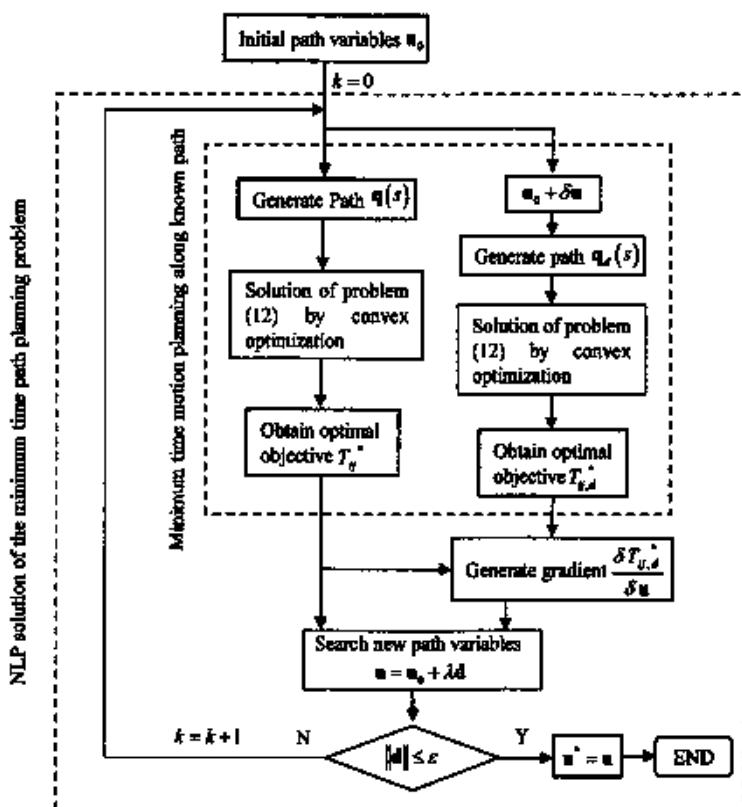


图 4.2: 点到点的最短时间轨迹规划两步嵌套优化策略

在本章中, 使用一种经典的遗传算法, 通过模拟自然选择、遗传和突变的过程来求得上述问题最终的最优路径以及总的加工时间。这个算法的过程如下:

路径选择的遗传算法步骤:

输入: 点的数目 n_c , 以及点与点之间的度量矩阵 D 。

输出: 最优的转移路径以及相应的总的转移时间。

过程:

初始化:

种群规模 pop_size ; $i = 0$;

进化的代数 max_gen ;

初始的种群(这里指初始的转移路径) $Pa_pop(i) = randperm(pop_size, n_c)$ 。

其满足问题(4.15)的约束。

基于度量矩阵 D ，评价每个种群的好坏(每条转移路径转移时间的长短)，找到最优的种群，并记录最短的转移时间为 T_{\min}^* 。

循环：

当 $i \leq \max_gen$ ，

第一步：在问题(4.15)的约束下，利用父种群 $Pa_pop(i)$ ，通过交叉和变异，产生新的子种群 $Ch_pop(i)$ 。

第二步：基于度量矩阵 D ，评价子种群的转移时间，并选出最短的，记为 T_{\min} 。

第三步：如果 $T_{\min}^* > T_{\min}$ ，则更新 $T_{\min}^* = T_{\min}$ 。

第四步：从父种群 $Pa_pop(i)$ 和子种群 $Ch_pop(i)$ 中选择较好的，作为下一代的父种群 $Pa_pop(i+1)$ 。

第五步： $i = i + 1$ 。

结束循环。

路径选择的遗传算法结束。

4.4 仿真与实验

在这一节中，通过加工一个在 $Y-Z$ 平面的多点任务来说明提出算法的有效性。这个实验有两个部分：第一部分是验证点到点轨迹规划的有效性；第二部分是测试在多点任务中，加工路径的最优性。测试用的机器人是一个三自由度的机械臂[46]，三个关节力矩的界均为[140; 140; 50]N.m。 $Y-Z$ 工作平面放在了 $X=1$ 的位置，并且假设机械臂终端关节和工作平面之间没有干涉。八个打孔点放置位置如图4.3所示。基于逆向运动学，对应于空间位置的机械臂的角位移可以相应得到。

4.4.1 点到点转移路径的测试

以点3到点8的转移路径为例，测试了提出算法的有效性。初始的路径是连接这两个点的一条直线段，长度为0.8879m，对应于这条路径的最短加工时间为0.4839s。图4.5显示了机器臂三个关节的最优力矩。运用提出的方法，求得

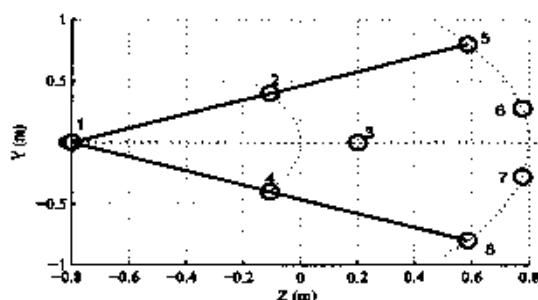


图 4.3: 在 $Y-Z$ 平面的多点加工任务

的点3到点8的最优转移路径如图4.4实线部分所示。它的长度为1.0346m，对应的加工时间为0.4510s，相应的力矩如图4.6所示。

根据图4.5和图4.6中力矩的“bang-bang”结构，说明在这两个路径情况下，得到的速度已经达到了最优值。图4.7显示了直线路径的最优速度和提出方法生成的最优路径的相应速度。可以观察到，尽管生成的最优路径比直线路径要长16.5%，但最终的加工时间却缩短了6.8%。原因可以总结为最优路径导致的速度规划问题解的可行域实际比直线路径更大，因为它至少更好地利用了重力而不是去抵消它。

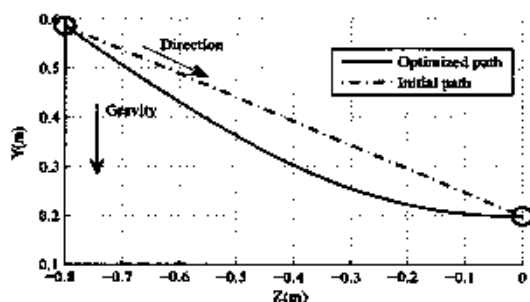


图 4.4: 在 $Y-Z$ 平面的最优路径和直线路径

4.4.2 多点规划最短加工时间的测试

通过运用上一节的结果，我们能够得到度量矩阵 D 每一个元素的值。当使

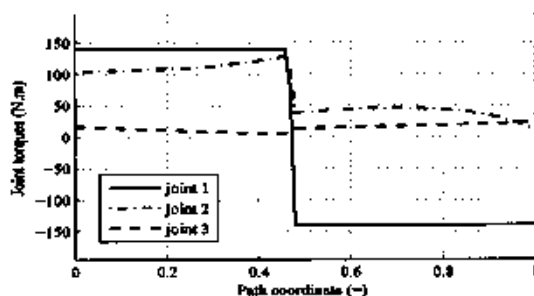


图 4.5: 直线路径每个关节的扭矩

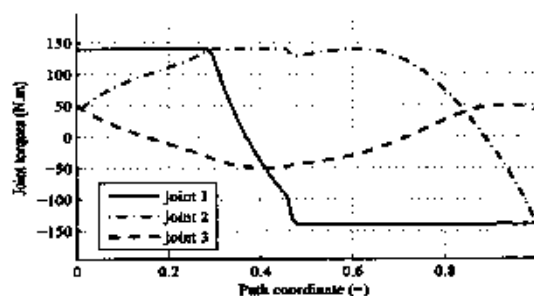


图 4.6: 最优路径每个关节的扭矩

用最短路程策略的时候，相当于使度量矩阵以距离作为度量，其元素取值为如图4.8所示。当使用提出算法也就是最短加工时间策略的时候，相当于度量矩阵以转移时间作为度量，这时候其元素如图4.9所示。这两种策略产生的路径如图4.10所示，可以看到，也是不同的。

对于最短路程策略：

8个点的转移顺序为7-6-5-4-3-2-1-4-8-7，相应的路径长度为5.456m，加工时间为3.25s。

对应的每段转移路径的长度为0.555-0.555-0.887-0.504-0.8-0.8-0.8-0.5559(m)。

对应的每段转移路径的加工时间为0.43-0.38-0.48-0.33-0.42-0.41-0.41-0.39(s)。

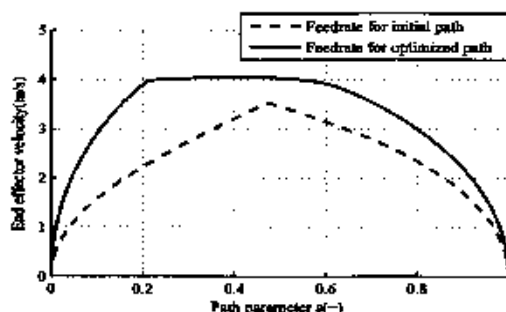


图 4.7: 直线路径和最优路径分别产生的最优速度

表 4.1: 遗传算法阈值

种群规模	40
最大代数	100
变异概率	75%
变异操作	交换25%;反转25%;平移25%

对于最短时间策略(提出的方法):

8个点的转移顺序为5-6-3-4-7-8-4-1-2-5, 相应的路径长度为5.858m, 加工时间为3.04s。

对应的每段转移路径的长度为0.571-0.688-0.682-0.565-0.829-0.835-0.836-0.852(m)。

对应的每段转移路径的加工时间为0.38-0.35-0.35-0.37-0.39-0.41-0.40-0.39(s)。

通过上面的数据, 可以看到尽管提出的方法加工路径的长度比最短路径策略长了7.3%, 但加工时间却减少了6.5%。因此, 对多点打孔、点焊任务中最短时间路径规划是必要的。图表4.2列出了提出方法解决这个问题在各个阶段使用的平均计算时间。计算环境为个人笔记本电脑, 配置为i3 2.5G 处理器和2 GB 内存。由于工业生产中轨迹规划过程是离线的, 因此这种对计算时间的消耗是可以接受的。

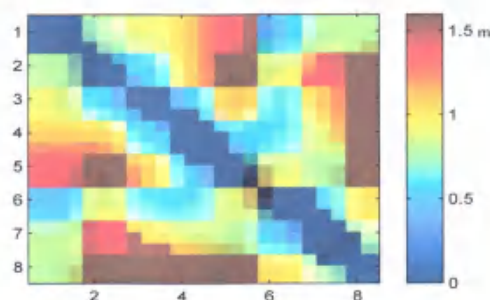


图 4.8: 度量矩阵元素的值(以距离为度量)

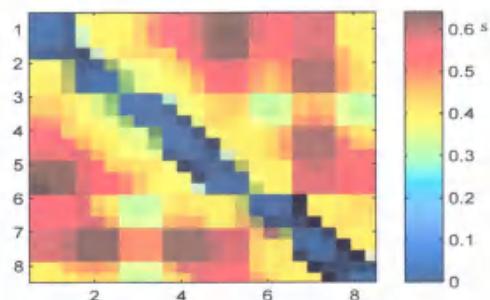


图 4.9: 度量矩阵元素的值(以转移时间为度量)

4.5 结论

在本章中,对于多点打孔、点焊的机器人轨迹规划问题我们提出了一种求解最短时间路径的新方法。提出方法在满足机器人动力学和运动学约束的前提下,使用Hermite插值多项式,对加工路径进行参数化,并利用改良的旅行商和遗传算法,将路径和速度作为一个整体,对该问题进行建模和求解。

本章最后通过仿真实验,对有八个点的空间打孔问题进行规划,结果表明,尽管提出的最短时间路径策略比工业常用的最短路径策略加工路径要长7.3%,但加工时间却减少了6.5%,这充分说明了研究这个问题的必要性和提出算法的有效性。

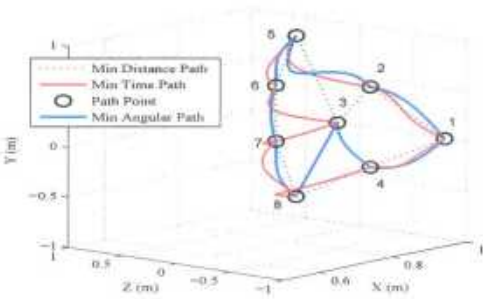


图 4.10: 最短路径策略和最优时间策略路径的对比

表 4.2: 提出算法各步骤计算时间

问题	计算时间[分:秒:毫秒]
度量矩阵	24:32:00
点到点最优路径计算	00:26:00
点到点转移时间求解	00:00:12
TSP求解	00:02:210
总共计算时间	24:34:21



第五章 结论与展望

5.1 本文总结

本论文从机器人和数控机床的最优轨迹规划问题出发, 针对给定路径条件下机器人的鲁棒轨迹规划、数控系统的长加工路径轨迹规划、不定路径机器人多点加工轨迹规划三个轨迹规划中的重要情形进行了研究。本论文的主要贡献总结如下:

在给定路径条件下机器人的鲁棒轨迹规划中, 提出了一种更有效的一般化的随机算法来解决这个问题。这种算法有两个最大的优点。一是: 只需要不确定参数的有限个样本点, 并不需要假定不确定参数的分布或限制其他比如期望、噪音等满足一些约束。二是: 这个概率函数的约束在这种方法中会被放缩为一系列的确定性约束, 并且在合理的条件下, 目标函数也被线性函数代替。因此最后整个问题被化为了一个线性规划模型, 从而可以在多项式时间内求解。

与其他的随机方法[39, 40]相比, 本文的算法也更加普遍和有效。在[39, 40]中, 概率约束通过切比雪夫不等式被简化成确定性的约束, 在这计算的过程中需要变量的期望、方差、协方差等, 而这一般来说对于多个随机变量的情况是很难计算的。本文的方法则使用了分位数的概念, 仅仅通过利用随机变量有限个样本数据就能轻松处理多个随机变量的情况, 并且给出的是一个对于原问题更好的一个近似。

在[86]中, 基于鲁棒优化理论, 提出了一种使用线性规划解决“严格鲁棒”轨迹规划的方法。但其使用的放缩方法并不是等价的, 因此最后得到的解也不是最优的, 而本文提出的方法对“严格鲁棒”问题进行的简化是等价的, 因此得到了这个问题的最优解。最后, 在上述问题的基础上, 提出了求解随机模型的一般化方法, 这种方法可以处理任意 n 个随机变量的情形, 从而可以用于更多的问题中。

针对长加工路径的轨迹规划问题, 本论文提出了一种基于线性规划的高效拼接方法。通过利用分段轨迹规划的局部最小值点进行循环拼接、组合、更新, 最后得到了一个全局解。并且该全局解和“pseudo-jerk”约束条件下的全

局最优解[86]非常接近($< 0.1\%$)。由于这个方法在拼接、组合、更新过程中对数值精度要求较高,因此本文还讨论了这种方法的鲁棒性,并在之后讨论了该算法在最坏情况下的浮点计算复杂度。

这种方法与以前的方法[22]相比,具备两个优势,一是得到的全局解与[86]中的全局最优解非常近似,从而效果远远好于[22]。二是大大提高了计算效率,在线性时间内就可以计算得到。最后本文又通过仿真和实验分别对二维猫形轨迹和三维牙齿形轨迹从加工时间、计算时间、跟踪误差、控制电压等方面对这种方法的有效性和精确性进行了论证。

在不定路径机器人多点加工轨迹规划中,由于要同时规划穿过点的次序以及点与点之间的路径,该问题一直是一个开放性问题。基于打孔、点焊问题是一种走-停运动,本文使用三次Hermite插值多项式来参数化加工路径,并利用改良的旅行商(TSP)对该问题进行建模,将该问题转化为了一个整数规划问题,并利用典型的遗传算法(GA)对该问题进行求解。通过仿真实验,与最短路径方法对比,尽管提出的方法加工路径要长7.3%,但加工时间却减少了6.5%,这充分说明了提出算法的实用性。

5.2 未来的研究方向

在机器人最优鲁棒轨迹规划问题中,本文使用的随机方法已经进行了一般化的推广,能够处理任意 n 个随机变量。从而使这种一般化的随机方法能够解决更多的问题,比如处理在机器人轨迹规划中更高阶的约束,像电压、力矩变化率等,相关的仿真实验也可以对照进行。

在长加工路轨的高效轨迹规划算法中,提出的速度拼接、组合、更新方法只考虑了两轴和三轴的数控系统轨迹规划问题,可以进一步考虑解决五轴数控系统的轨迹规划问题。

在机器人多点打孔和点焊任务的轨迹规划中,其实可以改进的地方至少有两个,一是度量矩阵的计算,由其中提供的计算时间的表格数据可知,这部分占了绝大多数的计算时间。但这其中有很多地方可以加速,比如可以使用上一章中提到的高效算法提速。二是关于遗传算法的选择上,本文只是采用了最普通的一种遗传算法,该算法的收敛性和收敛速度尚未得到证明,如何选择一种更合适的并且计算效率更高的求解算法,仍是一个值得认真考虑的问题。

参考文献

- [1] 周凯. *PC 数控原理, 系统及应用*. 机械工业出版社, 2006.
- [2] Emile Aarts and Jan Korst. Simulated annealing and boltzmann machines. 1988.
- [3] Yusuf Altintas and Kaan Erkorkmaz. Feedrate optimization for spline interpolation in high speed machine tools. *CIRP Annals-Manufacturing Technology*, 52(1):297-302, 2003.
- [4] EML Beale. Cycling in the dual simplex algorithm. *Naval Research Logistics Quarterly*, 2(4):269-275, 1955.
- [5] Dimitri P Bertsekas. *Convex optimization theory*. Athena Scientific Belmont, 2009.
- [6] Dimitri P Bertsekas, Dimitri P Bertsekas, Dimitri P Bertsekas, and Dimitri P Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena Scientific Belmont, MA, 1995.
- [7] John T Betts and William P Huffman. Path-constrained trajectory optimization using sparse sequential quadratic programming. *Journal of Guidance, Control, and Dynamics*, 16(1):59-68, 1993.
- [8] Xavier Bendaert, Sylvain Lavernhe, and Christophe Tournier. Feedrate interpolation with axis jerk constraints on 5-axis nurbs and g1 tool path. *International Journal of Machine Tools and Manufacture*, 57:73-82, 2012.
- [9] James E Bobrow. Optimal robot plant planning using the minimum-time criterion. *Robotics and Automation, IEEE Journal of*, 4(4):443-450, 1988.
- [10] James E Bobrow, Steven Dubowsky, and JS Gibson. Time-optimal control of robotic manipulators along specified paths. *The international journal of robotics research*, 4(3):3-17, 1985.

- [11] Stephen Boyd and Lieven Vandenbergh. *Convex optimization*. Cambridge university press, 2004.
- [12] Theodore WC Chen and Vassilios S Vassiliadis. Inequality path constraints in optimal control: a finite iteration ε -convergent scheme based on pointwise discretization. *Journal of Process Control*, 15(3):353–362, 2005.
- [13] Yaobin Chen and Alan A Desrochers. Structure of minimum-time control law for robotic manipulators with constrained paths. In *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, pages 971–976. IEEE, 1989.
- [14] Howie M Choset. *Principles of robot motion: theory, algorithms, and implementation*. MIT press, 2005.
- [15] D Costantinescu and EA Croft. Smooth and time-optimal trajectory planning for industrial manipulators along specified paths. *Journal of robotic systems*, 17(5):233–249, 2000.
- [16] John J Craig. *Introduction to robotics: mechanics and control*, volume 3. Pearson Prentice Hall Upper Saddle River, 2005.
- [17] Frederik Debrouwere, Wannes Van Loock, Goele Pipeleers, Quoc Tran Dinh, Moritz Diehl, Joris De Schutter, and Jan Swevers. Time-optimal path following for robots with convex-concave constraints using sequential convex programming. *Robotics, IEEE Transactions on*, 29(6):1485–1495, 2013.
- [18] Moritz Diehl, Hans Georg Bock, Holger Diedam, and P-B Wieber. Fast direct multiple shooting algorithms for optimal robot control. In *Fast motions in biomechanics and robotics*, pages 65–93. Springer, 2006.
- [19] Jingyan Dong, PM Ferreira, and JA Stori. Feed-rate optimization with jerk constraints for generating minimum-time trajectories. *International Journal of Machine Tools and Manufacture*, 47(12):1941–1955, 2007.

- [20] Jingyan Dong and JA Stori. A generalized time-optimal bidirectional scan algorithm for constrained feed-rate optimization. *Journal of dynamic systems, measurement, and control*, 128(2):379–390, 2006.
- [21] Steven Dubowsky and TD Blubaugh. Planning time-optimal robotic manipulator motions and work places for point-to-point tasks. *Robotics and Automation, IEEE Transactions on*, 5(3):377–381, 1989.
- [22] Kaan Erkorkmaz. Real-time feed optimization for spline toolpaths employing uninterrupted acceleration profiling. In *American Society for Precision Engineering (ASPE)*, 2012.
- [23] Wei Fan, Xiao-Shan Gao, Chen-Han Lee, Ke Zhang, and Qiang Zhang. Time-optimal interpolation for five-axis cnc machining along parametric tool path based on linear programming. *The International Journal of Advanced Manufacturing Technology*, 69(5-8):1373–1388, 2013.
- [24] Nicola Frigerio and Andrea Matta. Energy-efficient control strategies for machine tools with stochastic arrivals. *Automation Science and Engineering, IEEE Transactions on*, 12(1):50–61, 2015.
- [25] Alessandro Gasparetto and Vanni Zanotto. A technique for time-jerk optimal planning of robot trajectories. *Robotics and Computer-Integrated Manufacturing*, 24(3):415–426, 2008.
- [26] Philip E Gill, Walter Murray, and Michael A Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM review*, 47(1):99–131, 2005.
- [27] Michael Hintermüller, Kazufumi Ito, and Karl Kunisch. The primal-dual active set strategy as a semismooth newton method. *SIAM Journal on Optimization*, 13(3):865–888, 2002.
- [28] Peter Kali and Stein W Wallace. *Stochastic programming*. Springer, 1994.

- [29] Narendra Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 302–311. ACM, 1984.
- [30] James Kennedy. Particle swarm optimization. In *Encyclopedia of machine learning*, pages 760–766. Springer, 2011.
- [31] Jon Kieffer, Aidan J Cahill, and Matthew R James. Robust and accurate time-optimal path-tracking control for robot manipulators. *Robotics and Automation, IEEE Transactions on*, 13(6):880–890, 1997.
- [32] Victor Klee and George J Minty. How good is the simplex algorithm. Technical report, DTIC Document, 1970.
- [33] MH Korayem, M Irani, A Charesaz, AH Korayem, and A Hashemi. Trajectory planning of mobile manipulators using dynamic programming approach. *Robotica*, 31(04):643–656, 2013.
- [34] Torsten Kroger and Friedrich M Wahl. Online trajectory generation: Basic concepts for instantaneous reactions to unforeseen events. *Robotica, IEEE Transactions on*, 26(1):94–111, 2010.
- [35] Steven M LaValle. *Planning algorithms*. Cambridge university press, 2006.
- [36] Eugene L Lawler and David E Wood. Branch-and-bound methods: A survey. *Operations research*, 14(4):699–719, 1966.
- [37] Yuying Li. Centering, trust region, reflective techniques for nonlinear minimization subject to bounds. Technical report, Cornell University, 1993.
- [38] Chun-Shin Lin, Po-Rong Chang, and Js Ye So Luh. Formulation and optimization of cubic polynomial joint trajectories for industrial robots. *Automatic Control, IEEE Transactions on*, 28(12):1066–1074, 1983.
- [39] K Marti and A Aurnhammer. Robust optimal trajectory planning for robots by stochastic optimization. *Mathematical and Computer Modelling of Dynamical Systems*, 8(1):75–116, 2002.

- [40] Kurt Marti, Yuri Ermoliev, and Georg Ch Pflug. *Dynamic stochastic optimization*, volume 532. Springer Science & Business Media, 2012.
- [41] John A Nelder and Roger Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.
- [42] Yurii Nesterov, Arkadii Nemirovskii, and Yinyu Ye. *Interior-point polynomial algorithms in convex programming*, volume 13. SIAM, 1994.
- [43] Charles P Neuman. Discrete dynamic robot models. *IEEE transactions on systems, man, and cybernetics*, 15(2):193–204, 1985.
- [44] Ping-Qi Pan. A dual projective simplex method for linear programming. *Computers & Mathematics with Applications*, 35(6):119–135, 1998.
- [45] Jean-François Petiot, Patrick Chedmail, and Jean-Yves Hascoët. Contribution to the scheduling of trajectories in robotics. *Robotics and Computer-Integrated Manufacturing*, 14(3):237–251, 1998.
- [46] Friedrich Pfeiffer and Rainer Johanni. A concept for manipulator trajectory planning. *Robotics and Automation, IEEE Journal of*, 3(2):115–123, 1987.
- [47] Quang-Cuong Pham. A general, fast, and robust implementation of the time-optimal path parameterization algorithm. *Robotics, IEEE Transactions on*, 30(6):1533–1540, 2014.
- [48] HX Phu, HG Bock, and JP Schölder. Extremal solutions of some constrained control problems? *Optimization*, 35(4):345–355, 1995.
- [49] Les Piegl and Wayne Tiller. *The NURBS book*. Springer Science & Business Media, 2012.
- [50] Liangsheng Qu and Ruixiang Sun. A synergetic approach to genetic algorithms for solving traveling salesman problem. *Information Sciences*, 117(3):267–283, 1999.

- [51] Pedro Reynoso-Mora, Wenjie Chen, and Masayoshi Tomizuka. On the time-optimal trajectory planning and control of robotic manipulators along predefined paths. In *American Control Conference (ACC), 2013*, pages 371–377. IEEE, 2013.
- [52] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.
- [53] Mohammad Jafar Sadigh and Saeed Mansouri. Application of phase-plane method in generating minimum time solution for stable walking of biped robot with specified pattern of motion. *Robotica*, 31(06):837–851, 2013.
- [54] Volker Schulz. *Reduced SQP methods for large scale optimal control problems in DAE with application to path planning problems for satellite mounted robots*. PhD thesis, Citeseer, 1996.
- [55] B Sencer, Y Altintas, and E Croft. Feed optimization for five-axis cnc machine tools with drive constraints. *International Journal of Machine Tools and Manufacture*, 48(7):733–745, 2008.
- [56] Zvi Shiller. On singular time-optimal control along specified paths. *Robotics and Automation, IEEE Transactions on*, 10(4):561–566, 1994.
- [57] Zvi Shiller. Time-energy optimal control of articulated systems with geometric path constraints. *Journal of dynamic systems, measurement, and control*, 118(1):139–143, 1996.
- [58] Zvi Shiller and Steven Dubowsky. Robot path planning with obstacles, actuator, gripper, and payload constraints. *The International Journal of Robotics Research*, 8(6):3–18, 1989.
- [59] Zvi Shiller and Steven Dubowsky. On computing the global time-optimal motions of robotic manipulators in the presence of obstacles. *Robotics and Automation, IEEE Transactions on*, 7(6):785–797, 1991.

- [60] Zvi Shiller and Hsueh-Hen Lu. Computation of path constrained time optimal motions with dynamic singularities. *Journal of dynamic systems, measurement, and control*, 114(1):34–40, 1992.
- [61] Kang G Shin and Neil D McKay. Minimum-time control of robotic manipulators with geometric path constraints. *Automatic Control, IEEE Transactions on*, 30(6):531–541, 1985.
- [62] Kang G Shin and Neil D McKay. A dynamic programming approach to trajectory planning of robotic manipulators. *Automatic Control, IEEE Transactions on*, 31(6):491–500, 1986.
- [63] Kang G Shin and Neil D McKay. Robust trajectory planning for robotic manipulators under payload uncertainties. *Automatic Control, IEEE Transactions on*, 32(12):1044–1054, 1987.
- [64] S Singh and Ming-Chuan Leu. Optimal trajectory generation for robotic manipulators using dynamic programming. *Journal of dynamic systems, measurement, and control*, 109(2):88–96, 1987.
- [65] Jean-Jacques E Slotine and Hyun S Yang. Improving the efficiency of time-optimal path-following algorithms. *Robotics and Automation, IEEE Transactions on*, 5(1):118–124, 1989.
- [66] MC Steinbach, HG Bock, and RW Longman. Time-optimal extension and retraction of robots: Numerical analysis of the switching structure. *Journal of optimization theory and applications*, 84(3):589–616, 1995.
- [67] Hwee Huat Tan and Renfrey B Potts. Minimum time trajectory planner for the discrete dynamic robot model with dynamic constraints. *Robotics and Automation, IEEE Journal of*, 4(2):174–185, 1988.
- [68] Mikko Tarkainen and Zvi Shiller. Time optimal motions of manipulators with actuator dynamics. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pages 725–730. IEEE, 1993.

- [69] Sebastian D Timar and Rida T Farouki. Time-optimal traversal of curved paths by cartesian cnc machines under both constant and speed-dependent axis acceleration bounds. *Robotics and Computer-Integrated Manufacturing*, 23(5):563–579, 2007.
- [70] Sebastian D Timar, Rida T Farouki, Tait S Smith, and Casey L Boyadjieff. Algorithms for time-optimal control of cnc machines along curved tool paths. *Robotics and Computer-Integrated Manufacturing*, 21(1):37–53, 2005.
- [71] Alberto Trevisani. Planning of dynamically feasible trajectories for translational, planar, and underconstrained cable-driven robots. *Journal of Systems Science and Complexity*, 26(5):695–717, 2013.
- [72] Leslie G Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.
- [73] D Verschure, B Demeulenaere, J Swevers, J De Schutter, and M Diehl. Practical time-optimal trajectory planning for robots: a convex optimization approach. *IEEE Transactions on Automatic Control*, 2008.
- [74] Diederik Verschure, Bram Demeulenaere, Jan Swevers, Joris De Schutter, and Moritz Diehl. Time-optimal path tracking for robots: A convex optimization approach. *Automatic Control, IEEE Transactions on*, 54(10):2318–2327, 2009.
- [75] Diederik Verschure, Moritz Diehl, Joris De Schutter, and Jan Swevers. On-line time-optimal path tracking for robots. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 599–605. IEEE, 2009.
- [76] Oskar Von Stryk and Roland Bulirsch. Direct and indirect methods for trajectory optimization. *Annals of operations research*, 37(1):357–373, 1992.
- [77] Andrew D Wilson, Jarvis A Schultz, and Todd D Murphey. Trajectory optimization for well-conditioned parameter estimation. *Automation Science and Engineering, IEEE Transactions on*, 12(1):28–36, 2015.

- [78] Hoang Xuan Phu, Hans Georg Bock, and Johannes P Schlöder. The method of orienting curves and its application to manipulator trajectory planning. *Numerical Functional Analysis and Optimization*, 18(1-2):213–225, 1997.
- [79] Chunming Yuan, Ke Zhang, and Wei Fan. Time-optimal interpolation for cnc machining along curved tool paths with confined chord error. *Journal of Systems Science and Complexity*, 26(5):836–870, 2013.
- [80] P Th Zacharia and NA Aspragathos. Optimal robot task scheduling based on genetic algorithms. *Robotics and Computer-Integrated Manufacturing*, 21(1):67–79, 2005.
- [81] Ke Zhang, Chun-Ming Yuan, and Xiao-Shan Gao. Efficient algorithm for time-optimal feedrate planning and smoothing with confined chord error and acceleration. *The International Journal of Advanced Manufacturing Technology*, 66(9-12):1685–1697, 2013.
- [82] Ke Zhang, Chun-Ming Yuan, Xiao-Shan Gao, and Hongbo Li. A greedy algorithm for feedrate planning of cnc machines along curved tool paths with confined jerk. *Robotics and Computer-Integrated Manufacturing*, 28(4):472–483, 2012.
- [83] Mei Zhang, Wei Yan, ChunMing Yuan, DingKang Wang, and XiaoShan Gao. Curve fitting and optimal interpolation on cnc machines based on quadratic b-splines. *Science China Information Sciences*, 54(7):1407–1418, 2011.
- [84] Qiang Zhang and Shu-Rong Li. Efficient computation of smooth minimum time trajectory for cnc machining. *The International Journal of Advanced Manufacturing Technology*, 68(1-4):683–692, 2013.
- [85] Qiang Zhang, Shu-Rong Li, and Xiao-Shan Gao. Practical smooth minimum time trajectory planning for path following robotic manipulators. In *American Control Conference (ACC), 2013*, pages 2778–2783. IEEE, 2013.
- [86] Qiang Zhang, Shu-Rong Li, Jian-Xin Guo, and Xiao-Shan Gao. Tractable algorithm for robust time-optimal trajectory planning of robotic manipulators

- under confined torque. *International Journal of Computers Communications & Control*, 10(1):123–135, 2014.
- [87] Qiang Zhang, Shurong Li, Jian Xin Guo, and Xiao Shan Gao. Time-optimal path tracking for robots under dynamics constraints based on convex optimization. *Robotica*, pages 1–24, 2015.

发表文章目录

- [1] Mingyong Zhao, Qing-Ge Chen, Kaan Erkorkmaz, Xiao-Shan Gao. Linear Programming Based Feedrate Optimization For Contour Machining of Long Spline Toolpaths. *Spring Meeting: Precision Mechatronic System Design and Control*, 2016.
- [2] Qiang Zhang, Mingyong Zhao. Minimum Time Path Planning for Robotic Manipulator in Drilling/ Spot Welding Tasks. *Journal of Computational Design and Engineering*, 2015.

已完成的文 章

- [3] Mingyong Zhao, Xiao-Shan Gao. An Efficient Stochastic Approach for Robust Time-Optimal Trajectory Planning of Robotic Manipulators under Limited Actuation. Submitted to *Robotica*, 2016.
- [4] Qiang Zhang, Xiao-Shan Gao, Hongbo Li, Mingyong Zhao. Minimum Time Corner Transition Algorithm With Confined Feedrate and Axial Acceleration For NC Machining Along Linear Tool Path. Submitted to *International Journal of Advanced Manufacturing Technology*, 2016.

简 历

基本情况

赵明勇，男，1988年生于山东高密。

教育状况

2007年9月至2011年7月，吉林大学，数学学院，本科，专业：信息与计算科学。

2011年9月至今，中国科学院大学，数学与系统科学研究院，硕博连读研究生，专业：应用数学。

2015年4月至2016年4月，加拿大滑铁卢大学(University of Waterloo)，精确控制实验室(Precision Controls Lab)，国际交流生。

研究兴趣

机器人和数控机床的轨迹规划、各类问题的优化算法。

联系方式

E-mail: 1988zmy@gmail.com



致 谢

在中国科学院大学数学与系统科学研究院五年的学习生活中，首先特别感谢我的导师高小山研究员的悉心指导与支持。他对于本领域渊博的专业知识、对科研精益求精的态度以及敏锐的洞察力对本文的完成起着至关重要的作用；同时高老师勤奋廉洁的生活作风，也深深的影响了我，在此对高老师致以最衷心的感谢和钦佩。

此外，还要衷心感谢课题组的李洪波研究员、袁春明副研究员、贾晓红副研究员、申立勇副研究员、张立先师姐、张强师兄、文勇同学对我的指导和交流、关心与帮助。

在加拿大滑铁卢大学(University of Waterloo)一年的交流中，特别感谢教授Kaan Erkorkmaz在学习和生活上对我的鼓励与指导，感谢Christina Chen等研究生与我共同完成的工作。

最后，谨将本文献给我最亲爱的家人，感谢他们多年以来对我的付出与支持！