

西南科技大学  
研究生学位论文

基于四元数样条曲线的姿态轨迹规划研究

年        级        2012

姓        名        王效杰

申请学位级别        硕    士

专        业        控制理论与控制工程

指    导    教    师        梁艳阳 副教授

Classified Index: TP242.2

U. D. C: 621.3

Southwest University of Science and Technology

Master Degree Thesis

# Research on Orientation Trajectory Planning Based on Quaternion Spline Curve

Grade: 2012

Candidate: Wang Xiao Jie

Academic Degree Applied for: Master

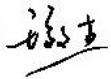
Speciality: Control Science and Control  
Engineering

Supervisor: Liang Yan Yang

Apr. 10. 2015

# 独 创 性 声 明

本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得西南科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

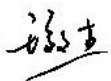
签 名: 

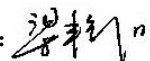
日 期: 2015. 5. 26

## 关于论文使用和授权的说明

本人完全了解西南科技大学有关保留、使用学位论文的规定，即：学校有权保留学位论文的复印件，允许该论文被查阅和借阅；学校可以公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存论文。

(保密的学位论文在解密后应遵守此规定)

签 名: 

导师签名: 

日 期: 2015. 5. 26

## 摘 要

近年来,工业机器人技术快速发展。运动控制是机器人技术的关键技术之一,而其中的轨迹规划直接影响着机器人的精确度及效率。轨迹规划一般分为位置轨迹规划和姿态轨迹规划。机器人姿态的表示方法有多种,如欧拉角、旋转矩阵等,但具有插值困难、奇异等问题,四元数则不存在类似问题。因此,本论文采用四元数样条曲线作为姿态插值曲线对机器人进行姿态轨迹规划。

为了能够进行多姿态示教点连续规划,本文将四元数样条曲线作为姿态插补曲线,并通过采用球面线性插补的方式对样条曲线首尾增加示教点,以使曲线首末与首末示教点重合,并且各示教点与节点矢量对应四元数重合。通过对给定四元数及其一二阶导数等边界值条件求取四元数样条曲线进行研究,以得到多四元数插值曲线平滑过渡曲线的一般形式。

其次,本文对基于四元数曲线的姿态轨迹规划算法进行了研究,主要包括两个问题:曲线正逆向拟合、四元数向姿态空间转换。为了得到姿态角位移,对四元数导数与角速度之间的关系进行研究,进而通过复合柯特斯公式进行数值积分。为了将速度规划得到的标量结果矢量化,首先通过三次厄米特插值反求出节点  $u$  关于角位移  $\theta$  的函数,进而通过中点有限差分法得到节点  $u$  关于时间的一二阶导数,最后通过角速度及加速度与四元数及其一二阶导数之间的关系将其矢量化。

通过建立机器人模型,在 MATLAB 上进行了仿真验证。仿真验证了本文所研究算法的可行性与有效性。

**关键词:** 四元数      样条曲线      姿态轨迹规划      过渡曲线      S 型速度规划

## Abstract

In recent years, industrial robot technology is having much rapid development. Motion control is one of the key technologies of industrial robots, and the trajectory planning of which impacts on its accuracy and efficiency directly. Trajectory planning is generally divided into position trajectory planning and orientation trajectory planning. There are several representing methods of orientation, such as Euler angles, rotation matrix, and so on. Unfortunately, the interpolation difficult or singularity may occur when using those methods. However, quaternion has no similar issues. Thus, this thesis uses quaternion spline curve as orientation interpolation curve for orientation trajectory planning.

For the sake of orientation continuous trajectory planning with multi-teaching-points, quaternion spline curve as the orientation interpolation curve is researched. The control points in the head and the tail are increased through spherical linear interpolation so that the head and the tail of quaternion spline curve coincide with the first and the end teaching point, and every teaching point coincides with the quaternions corresponding to node vector parameter. And a general type of smooth transition curve between multi-interpolation-curves is obtained based on given boundary conditions of quaternions, first and second derivatives.

Further, the orientation trajectory planning algorithms based on quaternion spline curve is researched, mainly including forward and backward curve fitting, transformation from quaternion to orientation space. In order to obtain orientation angular displacement, the relationship between quaternion along with its derivative and angular velocity is researched, and the angular displacement is received through the composite cotes formulas for numerical integration. In order to get the vector-velocity and vector-acceleration, the knot  $u$  about angular displacement  $\theta$  is reversed out by third-order Hermit interpolation firstly, the first and second derivative of knot  $u$  about time is obtained by the mean of middle finite difference method then, so the velocity

and acceleration obtained by S-type velocity planning can be vectorized utilizing the relationship between quaternion derivative and angular velocity.

By establishing the model of robot, simulation is worked in the MATLAB software. The simulation verified the feasibility and effectiveness of the algorithms proposed in this paper.

**Key words:** quaternion; spline curve; orientation trajectory planning; transition curve; S-type velocity planning

---

# 目 录

1 绪论 .....	1
1.1 课题研究背景及意义 .....	1
1.2 姿态轨迹规划研究现状 .....	2
1.2.1 机器人轨迹规划 .....	2
1.2.2 位置轨迹规划 .....	3
1.2.3 姿态轨迹规划 .....	4
1.3 论文主要研究内容 .....	5
1.3.1 主要研究内容 .....	5
1.3.2 论文结构 .....	6
2 机器人数学基础及运动学 .....	7
2.1 引言 .....	7
2.2 机器人数学基础 .....	7
2.2.1 姿态表示方法 .....	7
2.2.2 DH 建模 .....	9
2.3 机器人运动学(位姿) .....	11
2.3.1 运动学(位姿)正解 .....	11
2.3.2 运动学(位姿)逆解 .....	13
2.4 机器人运动学(速度及加速度) .....	17
2.4.1 雅克比矩阵 .....	17
2.4.2 机器人速度及加速度运动学正逆解 .....	18
2.5 本章小结 .....	19
3 基于四元数样条曲线的姿态插补曲线生成 .....	20
3.1 引言 .....	20
3.2 四元数基础 .....	21
3.2.1 四元数基本运算及四元数函数求导 .....	21
3.2.2 四元数与旋转 .....	23
3.2.3 四元数与旋转矩阵 .....	24

---

3.3 四元数样条曲线.....	25
3.3.1 曲线形式.....	25
3.3.2 节点矢量.....	26
3.3.3 控制点反求.....	27
3.3.4 四元数样条曲线导数.....	28
3.4 基于四元数的球面线性插补及其导数.....	29
3.4.1 四元数球面线性插补.....	29
3.4.2 四元数球面线性插补的导数.....	31
3.5 平滑过渡曲线的生成.....	31
3.6 本章小结.....	33
4 基于四元数的姿态轨迹规划算法研究.....	34
4.1 引言.....	34
4.2 四元数曲线拟合.....	34
4.2.1 四元数曲线导数与角速度及角加速度之间的关系.....	34
4.2.2 角位移拟合.....	35
4.3 四元数向姿态空间转换.....	36
4.3.1 曲线反向插值.....	37
4.3.2 四元数向角速度及角加速度转换.....	38
4.4 S 型速度规划.....	39
4.4.1 基本原理.....	40
4.4.2 核心算法设计.....	41
4.5 本章小结.....	46
5 算法仿真、验证及分析.....	47
5.1 机器人运动学仿真.....	47
5.2 基于四元数球面线性插值的姿态轨迹规划.....	49
5.3 基于四元数样条曲线的姿态轨迹规划.....	51
5.4 平滑过渡曲线的姿态轨迹规划.....	53
5.5 本章小结.....	56

---



---

结    论 .....	57
致    谢 .....	59
参考文献 .....	60
附录 1 .....	64
附录 2 .....	65
附录 3 .....	67
攻读硕士学位期间发表的学术论文及研究成果 .....	71

---

# 1 绪论

## 1.1 课题研究背景及意义

目前,随着我国人口老龄化的加重,人口红利逐渐消失,人力资源成本不断攀升,使得传统严重依赖劳动力的劳动密集型产业逐渐失去了竞争优势,产业结构调整 and 升级要求日益迫切。在这种背景下,机器人技术无疑是一个重要的突破口。现今,机器人技术已成为一个国家综合国力的体现,同时也是第四次工业革命(即由德国提出的工业 4.0 时代)的重要构成。

从 20 世纪 60 年代第一台机器人出现开始,机器人的种类越来越多样化,由最初的坦克炮台式机械臂发展到现在的机械臂、轮式机器人、飞行机器人、水下机器人、类人机器人等;应用领域快速扩大,由只能够进行旋转等简单运动到应用在工业、军事、航空航天、服务、医疗等各行各业;数目快速增长,劳动密集型企业的代表——富士康更是提出了百万机器人计划。目前,应用最为广泛、最成熟的机器人是工业机器人。国际标准化组织(International Organization for Standardization, ISO)对机器人的定义为:“机器人是一种自动的、位置可控的、具有编程能力的多功能机械手,这种机械手具有几个轴,能够借助于可编程序操作处理各种材料、零件、工具和专用装置,以执行种种任务”<sup>[1]</sup>。工业机器人是基于任务的多关节驱动末端的可编程机构,可应用于焊接、喷涂、装配、分拣、码垛、搬运等重复性任务<sup>[2]</sup>。工业机器人的应用将有助于我国产业的升级改造,有力缓解人口老龄化所带来的劳动力不足等问题,提升我国装备制造业及其他产业在国际上的竞争力。

近年来,我国工业机器人装备总量急剧上升,但本土工业机器人相关产业发展现状不容乐观。2013 年,中国以 3.7 万台的工业机器人销售量超过日本,成为当今世界最大的工业机器人市场。2014 年前三季度,国内工业机器人销量继续高速增长,达到 3.36 万台,同比增长 32.5%,预计全年达到 4.5 万台<sup>[3]</sup>。相对于美国、日本等发达国家而言,中国发机器人密度还处于较低水平,还有非常大的提升空间。但中国的工业机器人市场 90%的份额被 ABB、库卡、安川、发那科等发展历

史悠久、研究底蕴深厚、所生产机器人性能优越的外国机器人生产厂商所垄断<sup>[4]</sup>。为了改变现状，习近平将机器人比作“制造业皇冠上的明珠”以对其在我国制造业中的地位进行肯定，国家对机器人产业给予了优厚的政策支撑以对其进行扶持，各地政府也兴起了兴建机器人产业园区的热潮<sup>[5]</sup>。近年来，国内也涌现出了一批机器人研发与集成公司，一些实力雄厚的传统行业集团公司也通过设立子公司、并购、与国外机器人公司合资等方式涉足机器人产业。新松机器人是依托中国科学院沈阳自动化所设立的机器人公司，是国内最早、也是目前国内实力最雄厚的机器人公司。此外，实力较强的机器人公司还有广州数控、安徽埃夫特、南京埃斯顿、哈尔滨博实、绵阳福德等。上海新时达依托于自身在工业控制方面的优势成立了新时达机器人子公司，已初具规模。首钢与日本的机器人优势企业安川共同投资注册了首钢—莫托曼机器人有限公司。

虽然国内机器人行业在面对严峻形势下取得了长足的进步，但相对于国外还存在着诸多不足。机器人的本体设计目前不存在专利保护，机器人主要的核心部分有五部分，即：伺服系统、控制器、核心算法、精密减速器及应用和集成技术<sup>[4]</sup>。目前，国内大部分机器人公司仍从事附加值最低的机器人集成，而真正掌握的核心技术非常少。核心算法作为机器人技术五大核心之一，是机器人稳定运行、精准完成预定任务的基础。轨迹规划是上层控制算法的核心，往往又分为位置轨迹规划和姿态轨迹规划，本文将重点对机器人姿态轨迹规划进行研究。本研究的进行将有助于对机器人核心算法的掌握，提高国产机器人的稳定性及精确性，创造我们国家自己的机器人民族品牌，提升国产机器人在国内乃至全球的竞争力，降低对国外机器人的依赖，破除国外机器人在国内的垄断地位，为国内装备制造业升级改造提供强大支撑，降低人口老龄化加深、劳动力供应下降、人力资源成本上升对我国制造业在全球竞争力的不利影响。

## 1.2 姿态轨迹规划研究现状

### 1.2.1 机器人轨迹规划

目前，广泛应用的工业机器人为六自由度机械臂，该机器人由六个旋转关节通过连杆依次连接成链状，末端连接有完成某类任务所需要的执行机构。在工业上，一般采用示教—再现方式来使机器人进行运动<sup>[6]</sup>。示教即轨迹规划：首先确定

机器人完成特定工作所必须经过的关键点(即示教点),并记录各个示教点处的位置和姿态(简称为“位姿”)信息;然后确定各个示教点之间的路径类型(直线、圆弧、样条曲线、自由曲线等);最后确定运动规律,即确定位姿、速度、加速度等与时间的对应关系,以便机器人执行。轨迹规划分为关节空间轨迹规划和任务空间轨迹规划<sup>[7]</sup>。关节空间轨迹规划时,首先将示教点的位姿通过运动学逆解得到相应各个关节变量的值,然后分别对每个关节进行规划后进行同步。关节空间轨迹规划计算量小,并且不存在奇异,可充分发挥电机的性能,但同时也存在末端所经过路径未知的缺点,因此只有在大自由空间中为了获得快速运行时才使用<sup>[8]</sup>。关节空间轨迹规划相对较为简单,且研究较为成熟。根据边界值条件,可采用多项式插值<sup>[9-11]</sup>,或抛物线-线性-抛物线插值<sup>[12, 13]</sup>,也可采用更为灵活的样条曲线插值<sup>[14, 15]</sup>。相对于关节空间轨迹规划而言,任务空间轨迹规划更加安全,但在接近奇异时往往规划失败,而且任务空间中的速度和加速度与机器人的位型关系密切<sup>[16]</sup>。但是,由于工业环境复杂,出于避障或特殊任务的需要,多数采用任务空间轨迹规划。

### 1.2.2 位置轨迹规划

在任务空间中,机器人的运动一般分为平移和旋转。任务空间轨迹规划可分为耦合规划和解耦规划<sup>[17]</sup>。耦合规划即机器人末端的位置进行轨迹规划,姿态由平移路径来确定;解耦规划即将运动分为平移和旋转分别进行规划,然后进行同步。耦合轨迹规划虽然减少了一个规划量,增加了操作的简便性,但同时也丢失了确定姿态的灵活性。位置轨迹规划即给出欧氏空间中连续曲线作为末端的路径,并给出路径上位置点、一二阶导数与时间之间的对应关系。机械臂的路径一般由连接示教点的直线、圆弧等基本单元构成。但直线-直线、直线-圆弧、圆弧-圆弧轨迹连接点处不具有  $C^2$  连续性,故相应规划出来的加速度也是不连续的。为了使路径具有  $C^2$  连续性,往往需要在连接点处用曲线进行连接。陈伟华采用圆弧最为连续直线规划的过渡曲线,但只能保证  $C^1$  连续<sup>[18]</sup>。张升<sup>[19]</sup>提出在关节空间中进行线性插值时用抛物线作为过渡曲线以保证加速度连续。林威等<sup>[20]</sup>将该性质推广到任务空间轨迹规划,即在笛卡尔空间中直线轨迹间用抛物线进行过渡。但对于直线-圆弧、圆弧-圆弧过渡却不能简单地用抛物线或其他平面曲线进行过渡,因为该情况中两段路径很可能不在同一平面。此外,当机械臂末端进行自由曲线运动时,也需要进行空间曲线插补。目前,比较理想的方法是采用样条曲线进行插值。B样条曲线是贝塞尔曲线的一般化,可以简便地对空间任意示教点间进行任意连续

性插值, 并且其导数形式统一, 易于实现。由 B 样条曲线推广而来的 NURBS 曲线 (非均匀有理 B 样条曲线) 因其良好的性质被 ISO 选做工业产品几何性质的重要标准<sup>[21]</sup>。毛征宇等<sup>[22]</sup>运用三次均匀 B 样条曲线进行数控加工中复杂曲线的轨迹规划。张红强等<sup>[23]</sup>将 B 样条曲线用于关节空间轨迹设计。赖永林等<sup>[24]</sup>采用三次非均匀有理 B 样条曲线作为插值曲线进行任务空间轨迹规划, 并给出了曲线积分、曲线反向拟合求解方法, 并利用曲率极大值点为分段点对速度、加速度进行了分段规划。王文莉等<sup>[25]</sup>利用 NURBS 曲线进行自适应实时插补。样条曲线还可以作为直线-圆弧、圆弧-圆弧轨迹的过渡曲线<sup>[26]</sup>。此外, 为了减少参数降低计算量, Biagiotti 等<sup>[17]</sup>提出了基于样条曲线的位置近似轨迹规划和混合轨迹规划方法。

### 1.2.3 姿态轨迹规划

由于缺乏直观性, 并且姿态表示为非欧式空间, 因此, 与平移轨迹规划相比, 姿态轨迹规划方面的研究相对较少, 且研究不够全面、完善。机械臂末端姿态的表示方法一般有欧拉角、RPY、旋转矩阵、四元数等<sup>[16]</sup>。但不能直接利用欧拉角、旋转矩阵进行姿态插补。因为欧拉角存在万向锁等导致插值困难的问题, 直接对旋转矩阵进行插值还会破坏其正交性, 四元数则不存在这些问题<sup>[27]</sup>。RPY 虽然操作简单易行, 但姿态表示不直观, 并且还会存在奇异。陈伟华等<sup>[28]</sup>直接在笛卡尔空间对三维坐标、RPY 三个姿态角共六个分量进行过渡插补, 虽然比关节空间直观, 但直接对坐标分量以及姿态角度分量插补再组合成三维位置轨迹和姿态轨迹的方式, 过渡轨迹中插补点的位置、速度、加速度处于不可控状态。文献[17]将机器人末端姿态运动转化为绕任意轴旋转进行规划, 规划变量为绕任意轴的旋转角度。该方法简单、直观, 计算简便, 且计算量小, 但只适用于两点间姿态规划, 否则在两个示教点间进行规划时必须将首末速度设为 0, 这样必然会降低执行效率。

针对这些问题, 学者们将四元数<sup>[29, 30]</sup>概念引入姿态插补。目前, 四元数已被应用在飞行器姿态控制<sup>[31, 32]</sup>、计算机图形图像及动画<sup>[33, 34]</sup>、机器人末端姿态控制及运动学求解<sup>[31, 35, 36]</sup>等诸多方面。利用四元数描述末端姿态可以有效解决万向锁等问题, 而且姿态描述直观、运算量小。单位四元数可表示为  $q=(\cos(\theta), n*\sin(\theta))$ , 可表示刚体绕轴  $n$  旋转  $2*\theta$  后的姿态。Min-xiu Kong 等<sup>[37]</sup>将姿态规划归结为  $q=(\cos(\theta t/2), n*\sin(\theta t/2))$  进行插值, 并进行轨迹规划。虽然较为简便, 但其原理与

绕任意轴旋转相同，并存在相同的缺陷。四元数插补算法有球面线性插补、球面样条插补等<sup>[38]</sup>。基于四元数的姿态插补同样可以归结为球面插补。一般情况下运用四元数球面线性插值可有效解决两点之间的姿态插补。但球面线性插补只能保证曲线  $C^0$  连续，在示教点处角速度必须为 0，否则会对机械臂造成冲击。对于多点姿态过渡问题，可以采用球形曲线连续点四元数插补算法—Squad，该方法能使四维单位球上的姿态曲线达到一阶连续<sup>[38]</sup>。为了进一步改善姿态平滑度，任秉银等<sup>[39]</sup>在研究圆弧插补中，用四元数及分段三次有理插值保形样条函数使姿态达到连续。季晨<sup>[40]</sup>将单位四元数曲线构造问题转化为在欧式空间中单位球面曲线问题。张小明等<sup>[41]</sup>通过建立球面上的点和四元数空间中的点之间的一一对应关系，从而应用三维欧式空间的 NURBS 曲线对四元数进行插补。但他们均没有证明曲线的  $C^2$  连续，并且也没有给出曲线的导数。Myoung-Jun Kim 等<sup>[42]</sup>由空间 B 样条曲线出发，提出了高阶可导的单位四元数样条曲线，并给出了其导数形式。Myoung-Jun Kim 等<sup>[43]</sup>提出了一种求取四元数样条曲线控制点的数值方法。该方法虽然能得到控制点，但求得的首末控制点不与首末示教点重合，示教点也不完全与节点对应的点相重合，这都使得该方法求出的四元数样条曲线无法应用在机器人末端姿态插补中。Wenbing Ge 等<sup>[44]</sup>提出了一种求解控制点的解析方法。Kenjiro T Miura<sup>[45]</sup>提出了积分四元数积分曲线，该曲线具有简单的曲率表达形式，并且曲线完全位于控制曲线内，但其需要进行复杂的数值积分，计算量大。

## 1.3 论文主要研究内容

### 1.3.1 主要研究内容

虽然四元数样条曲线有非常好的性质，但基于四元数样条曲线的机器人姿态轨迹规划技术研究得还不够完善、系统。因此，本文主要的研究内容如下：

- 1.研究并实现四元数样条曲线节点矢量的确定、控制点的求取，以满足机器人末端姿态插值的需求，从而实现多点连续规划；以四元数样条曲线为基础，对多四元数插补曲线间平滑过渡技术进行研究。

- 2.应用 S 型速度规划对姿态进行轨迹规划；研究并实现由四元数空间(四元数及其一二阶导数)向姿态角空间(姿态角、角速度及角加速度)的转换。

### 1.3.2 论文结构

本论文共分为 5 章。

第 1 章为绪论，主要介绍了本论文的研究背景及所具有的研究价值和意义，并对机器人轨迹规划技术中位置轨迹规划和姿态轨迹规划尤其是四元数样条曲线的发展及其在刚体姿态插值中的应用等领域的国内外研究现状进行了综述。

第 2 章主要建立本文的论证平台——六自由度机械臂的运动学模型。为了实现该目标，首先确定了模型建立所需要的数学基础，包括姿态表示方法及 D-H 建模方法等；然后，给出了六自由度机器人运动学位置正解算法及逆解的几何解法和数值解法；最后，给出了机器人的速度、加速度正逆解所需要的雅克比矩阵及其导数的求解方法，并给出了本文中机械臂的雅克比矩阵及其导数矩阵。

第 3 章的主要内容为基于四元数样条曲线的姿态插补曲线生成，为本文的重点内容之一。首先给出了本文涉及到的四元数相关基础，并说明了四元数与刚体旋转之间的关系；然后给出了四元数样条曲线及其一二阶导数的表达形式、曲线控制点反求算法，并结合机器人姿态插补进行了曲线优化；接着，介绍了基于四元数的姿态球面线性插补，并给出其一二阶导数表达形式；最后，比对位置插补，在单位球面上进行多插补曲线间平滑过渡曲线生成等相关算法的研究。

第 4 章主要对基于四元数的姿态轨迹规划等相关算法进行研究，为本文的重点内容之一。首先对四元数曲线长度拟合及反向拟合的方法进行了研究；然后，对由四元数空间(四元数及其一二阶导数)向姿态角空间(姿态角、角速度及角加速度)的转换相关算法进行研究；最后，给出了基于四元数的姿态轨迹规划方法，包括 S 型速度规划等相关算法的研究。

第 5 章对上述研究的算法进行联合仿真验证。首先随机给定两个示教点，并以直线对位置、以绕固定轴旋转对姿态进行轨迹规划，以验证机器人运动学相关算法的正确性；然后，固定示教点位置坐标，给定两个姿态点进行球面线性插补以验证姿态角位移拟合、速度加速度矢量化及速度规划相关算法；接着，验证了基于四元数样条曲线的姿态轨迹规划的可行性；最后，基于球面线性插补曲线对平滑过渡曲线姿态轨迹规划算法进行了验证。

## 2 机器人数学基础及运动学

### 2.1 引言

机器人运动学是上层规划包括轨迹规划的基础。为了证明所研究算法的正确性，需要有平台进行验证。本文主要对姿态进行轨迹规划，因此验证平台机器人要具备三个旋转自由度。而六自由度机械臂是目前工业应用最广泛也是成熟的一类机器人，本文就以六自由度工业机械臂作为验证平台。为了建立六自由度机械臂的运动模型，本章首先介绍一些必备的数学基础，主要有姿态的表示方法及 DH 建模方法等；然后建立机器人的正逆运动学模型，包括位置正逆解、速度及加速度正逆解。

### 2.2 机器人数学基础

机器人的数学基础的内容非常繁多，不便在此详述。但在应用时，由于应用场合或习惯的不同，相同概念就有可能有多种描述和表达方式，如表示姿态的常用方法欧拉角共有 12 种表示方法<sup>[46]</sup>。为避免歧义，该部分将对本文所用到的重要数学知识进行介绍。

#### 2.2.1 姿态表示方法

常用的机器人姿态表示方法主要有齐次坐标矩阵、欧拉角、四元数等方法。齐次坐标矩阵主要用在坐标系的转换与传递。机器人示教时其姿态一般以欧拉角的方式给出，四元数是本文姿态轨迹规划时姿态的主要表示方法。本部分将对坐标矩阵及欧拉角进行说明与分析，四元数将在第 3 章中进行介绍。在此，将直接对结论进行叙述，详细说明可参考文献<sup>[16,47]</sup>。

##### (1) 齐次坐标矩阵

由于机器人机构相对较为复杂，直接由机器人关节变量求取末端的位姿参数较为困难。在进行位姿正逆解时，一般将问题转换为两个相邻关节之间的转换，即建立在两关节上坐标系的相对关系。对其进行表示的最简便方式为齐次坐标矩阵。如图 2-1 为两个坐标系之间的关系表示。坐标系  $R_j$  的原点在  $R_i$  中的坐标为  $P = [p_x p_y p_z]^T$ ，坐标系  $R_j$  的  $x$ 、 $y$ 、 $z$  轴的方向可用  $R_i$  中的相互正交的单位向量  $s$ 、 $n$ 、 $a$



进行表示，则坐标系  $R_j$  相对  $R_i$  可表示为式(2.1)。

$${}^i T_j = \begin{bmatrix} s & n & a & P \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

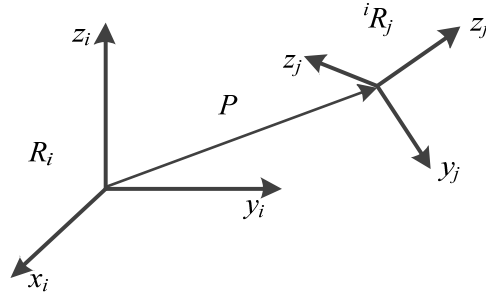


图 2-1 坐标系变换

Figure 2-1 Transformation of frames

其中，姿态可用矩阵形式进行表示，如式(2.2)所示。

$${}^i R_j = [s \quad n \quad a] = \begin{bmatrix} s_x & n_x & a_x \\ s_y & n_y & a_y \\ s_z & n_z & a_z \end{bmatrix} \quad (2.2)$$

## (2) 欧拉角

虽然欧拉角具有奇异和精确度低等缺点，但因为具有直观且简便易用，成为刚体姿态最普遍的表示方法。欧拉角有多种表示方法，在本文中采用 ZXZ 的方式。用欧拉角表示刚体姿态时，需要三个角度进行表示，即  $\varphi$ 、 $\theta$ 、 $\psi$ ，如图 2-2。

参考图 2-2，欧拉角的定义如下：

首先，绕  $z_0$  旋转角度  $\varphi$ ，使得  $x_0$  与  $N$  重合，其中  $0 \leq \varphi \leq 2\pi$ ；

然后，绕新坐标系的  $x$  轴(即  $N$ )旋转角度  $\theta$ ，使得  $z_0$  与  $z_n$  重合，其中  $0 \leq \theta \leq \pi$ ；

最后，绕  $z_n$  旋转角度  $\psi$ ，使得  $N$  与  $x_n$  重合，其中  $0 \leq \psi \leq 2\pi$ 。

经过上述三步变换，坐标系  $x_0 y_0 z_0$  与  $x_n y_n z_n$  重合， $\varphi$ 、 $\theta$ 、 $\psi$  即为欧拉角。根据旋转矩阵的性质，可以得到相应的变化矩阵为式(2.3)。

$$\begin{aligned} {}^0 R_n &= \text{rot}(z, \varphi) \text{rot}(x, \theta) \text{rot}(z, \psi) \\ &= \begin{bmatrix} C\varphi C\psi - S\varphi C\theta S\psi & -C\varphi S\psi - S\varphi C\theta C\psi & S\varphi S\theta \\ S\varphi C\psi + C\varphi C\theta S\psi & -S\varphi S\psi + C\varphi C\theta C\psi & -C\varphi S\theta \\ S\theta S\psi & S\theta C\psi & C\theta \end{bmatrix} \end{aligned} \quad (2.3)$$

其中， $C$  代表  $\cos$ ， $S$  代表  $\sin$ 。

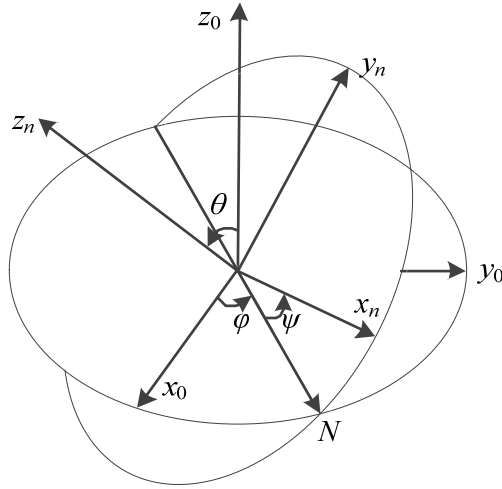


图 2-2 欧拉角

Figure 2-2 Euler angles

在进行运动学逆解时，往往还需要根据已知的旋转矩阵得到相对应的欧拉角。已知旋转矩阵为式(2.2)，与式(2.3)相比较，可以得到相应的结果为：

①当  $a_x$  与  $a_y$  不同时为 0 时， $\theta = \arctan 2\left(a_z, \sqrt{1-a_z^2}\right)$ ， $\varphi = \text{atan2}(a_x, a_y)$ ， $\psi = \text{atan2}(s_z, n_z)$ 。

②当  $a_x = a_y = 0$  时，此时会出现奇异，分两种情况进行分析：

当  $a_z = -1$  时， $\theta = \pi$ ， $\varphi - \psi = \text{atan2}(n_x, -n_y)$ 。即，当绕  $x$  轴旋转  $180^\circ$  时，新的  $z$  轴方向与原  $z$  轴方向相反，两次分别绕  $z$  轴旋转的角度  $\varphi$ 、 $\psi$  等价于一次绕  $z$  轴旋转  $\varphi - \psi$ 。

当  $a_z = 1$  时， $\theta = 0$ ， $\varphi + \psi = \text{atan2}(n_x, n_y)$ 。即，当绕  $x$  轴旋转  $0^\circ$  后，新的  $z$  轴方向与原  $z$  轴方向相同，两次分别绕  $z$  轴旋转的角度  $\varphi$ 、 $\psi$  等价于一次绕  $z$  轴旋转  $\varphi + \psi$ 。

### 2.2.2 DH 建模

在对串联机器人进行运动学分析时，往往把串联机器人转换为链式结构，通过相邻关节之间的相对位姿关系，最终得到末端相对基坐标系的位姿关系。为了得到相邻关节之间的位姿关系，许多学者进行了相关研究，其中最著名的方法为 DH 参数法<sup>[48]</sup>。根据  $x$  轴选取方式的不同，DH 建模有两种方法，在此仅对本文中使用的方法进行介绍。如图 2-3 所示，机器人的基座为连杆 0，关节  $i$  连接连杆  $i-1$  和连杆  $i$ ，坐标轴的选取为：

第  $i$  个关节的转轴为第  $i$  个坐标系的  $z$  轴，当关节为平移关节时，平移方向为  $z$  轴方向；两个相邻坐标系的  $z$  轴  $z_i$  与  $z_{i-1}$  的公垂线作为第  $i-1$  个坐标系的  $x$  轴，即

$x_{i-1}$ ;  $y_{i-1}$  用右手系的方式进行确定。

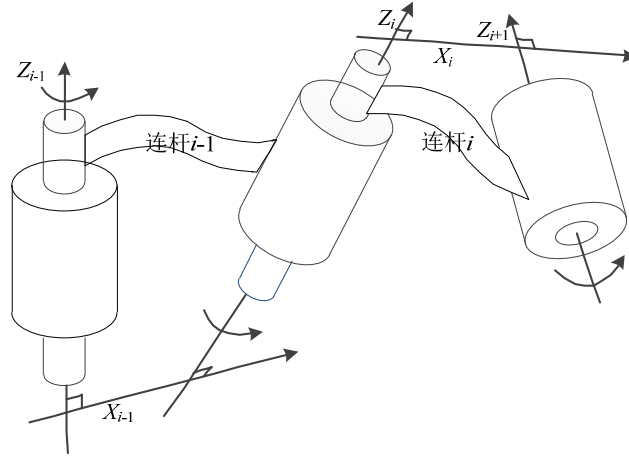


图 2-3 DH 建模

Figure 2-3 DH modeling

为了更加清晰地表示使用 DH 参数法确定两个相邻关节上坐标系之间的关系，将图 2-3 表示为图 2-4 所示。两个  $z$  轴分别沿两个关节的轴向方向，取  $Z_{i-1}$  与  $Z_i$  的公垂线作为第  $i-1$  个坐标系的  $x$  轴，即  $X_{i-1}$ 。两个坐标系的关系可通过以下四步得到：

- ①绕  $X_{i-1}$  轴旋转  $\alpha_{i-1}$ ， $Z_{i-1}$  旋转到  $z_1$  处，此时，两个  $z$  轴平行；
- ②沿  $X_{i-1}$  平移  $d_{i-1}$ ，两个  $z$  轴重合；
- ③沿  $Z_i$  平移  $r_i$ ，两个坐标系的原点重合；
- ④绕  $Z_i$  轴旋转  $\theta_i$ ，则两个坐标系重合。

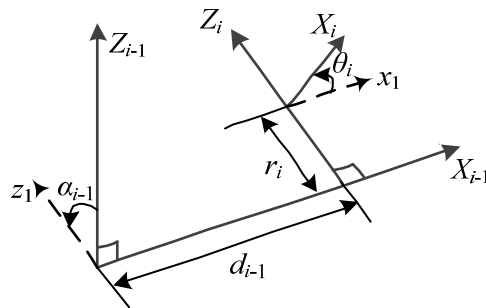


图 2-4 DH 参数的几何表示

Figure 2-4 The geometry representation of DH parameters

以上四步中的  $\alpha_{i-1}$ 、 $d_{i-1}$ 、 $r_i$ 、 $\theta_i$  被称为 DH 参数。根据坐标变换的性质，可以

得到坐标系  $i$  相对于坐标系  $i-1$  的位姿矩阵为式(2.4)。

$${}^{i-1}T_i = Rot(x, \alpha_{i-1}) Trans(x, d_{i-1}) Trans(z, r_i) Rot(z, \theta_i)$$

$$= \begin{bmatrix} C\theta_i & -S\theta_i & 0 & d_{i-1} \\ C\alpha_{i-1}S\theta_i & C\alpha_{i-1}C\theta_i & -S\alpha_{i-1} & -r_iS\alpha_{i-1} \\ S\alpha_{i-1}S\theta_i & S\alpha_{i-1}C\theta_i & C\alpha_{i-1} & r_iC\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

## 2.3 机器人运动学(位姿)

### 2.3.1 运动学(位姿)正解

机器人运动学位置正解是指已知机器人的结构参数及关节变量求得末端的位姿。本论文以实验室开发的 FDR-v6 型六自由度机械臂(图 2-5)为实验平台。根据 2.2.2 中 DH 建模方法可以得到机器人的几何模型如图 2-6 所示,其中标注了各个关节的转动方向及相邻关节的相对位姿关系,其对应 DH 参数表为表 2-1。为了使工具坐标系在初始状态(各个关节角为 0)时,工具坐标系与基坐标系的姿态相同,首先将关节 6 坐标系沿  $z_6$  平移  $r_t$  是的工具坐标系原点与关节 6 坐标系原点重合,然后再进行 DH 变化。

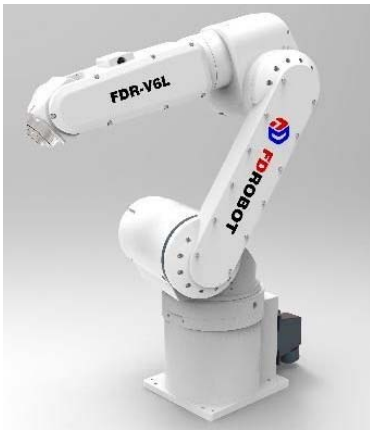


图 2-5 FDR-v6 概念模型

Figure 2-5 the conceptual model of DR-v6

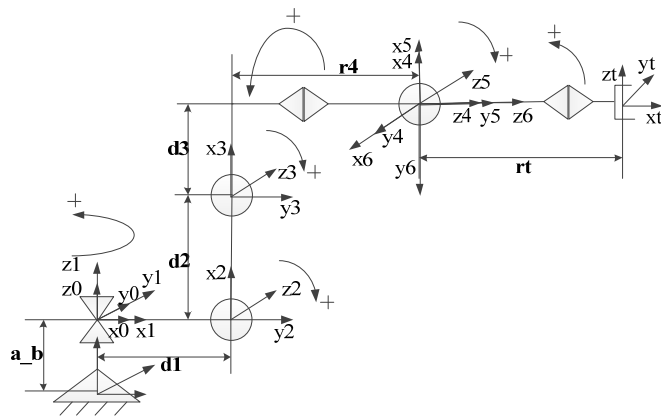


图 2-6 FDR-v6 几何模型

Figure 2-6 the geometric model of FDR-v6

得到机器人 DH 参数后,就可以根据 2.2.2 得到第  $i-1$  个关节坐标系到第  $i$  个关节坐标系的变换矩阵  ${}^{i-1}T_i$ , 其中  $i=1, \dots, 6, t$ ; 根据坐标变换的性质, 将各个变换矩阵顺次相乘, 即可得到末端的位姿矩阵, 其计算公式可表示为式(2.5)。

表 2-1 DH 参数表

Table 2-1 The table of DH parameters

连杆 $i$	$\alpha_{i-1}(^\circ)$	$d_{i-1}(\text{mm})$	$r_i(\text{mm})$	$\theta_i(^\circ)$	范围( $^\circ$ )
1	0	0	0	$\theta_1$	-170~170
2	-90	85	0	$\theta_2(-90)$	-92~135
3	0	380	0	$\theta_3$	-129~116
4	-90	100	425	$\theta_4$	-160~160
5	90	0	0	$\theta_5$	-120~120
6	-90	0	0	$\theta_6(90)$	-360~360
$t$	沿 z6 平移 rt=85				0
	90	0	0	90	

$${}^0T_t = {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \cdot {}^3T_4 \cdot {}^4T_5 \cdot {}^5T_6 \cdot T_t \quad (2.5)$$

其中:

$${}^0T_1 = \begin{bmatrix} c1 & -s1 & 0 & 0 \\ s1 & c1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^1T_2 = \begin{bmatrix} s2 & c2 & 0 & d_1 \\ 0 & 0 & 1 & 0 \\ c2 & -s2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^2T_3 = \begin{bmatrix} c3 & -s3 & 0 & d_2 \\ s3 & c3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$${}^3T_4 = \begin{bmatrix} c4 & -s4 & 0 & d_3 \\ 0 & 0 & 1 & r_4 \\ -s4 & -c4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^4T_5 = \begin{bmatrix} c5 & -s5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s5 & c5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^5T_6 = \begin{bmatrix} -s6 & -c6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -c6 & s6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$${}^6T_t = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & r_t \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

上式中  $c_i$ 、 $s_i$  分别代表  $\cos\theta_i$ 、 $\sin\theta_i$ ，其中  $i=1, \dots, 6$ 。对机器人进行运动控制时，需要将上述算法转化为程序进行执行。由于进行矩阵运算较为困难，故一般将其转化为多项式进行计算，结果见附录 1。

### 2.3.2 运动学(位姿)逆解

要对机器人进行控制，不仅需要根据机器人各个关节变量及其结构得到末端的位姿，还需要根据末端的位姿及其结构得到各个关节变量的值，即运动学(位姿)逆解。串联机器人的运动学(位姿)逆解较正解困难地多，主要方法有数值解法和解析解法。数值解法计算速度快，适合于在线运行，但其难点在于初值的选取。很多方法的初值采用随机法，即在工作空间内任意选取直到迭代到目标值。但随机选取的值有很大的不确定性，有可能位于奇异的工作空间或工作空间外。即使有解，所得到的结果未必最优(逆解有多解，初值直接影响到结果)。解析法有几何解法和代数解法。代数解法用优化函数进行最优解的选取，有可能选择的结果的位形并不是最优的；而几何解法具有非常强的直观性。但解析法的计算速度不如数值解法。因此，本文采用解析法中的几何解法进行数值解法初值的选取，这样就弥补了二者的缺点。

#### (1)几何解法<sup>[49]</sup>

由图 2-6 所示的建模方法可以看出，后面三个关节对应的坐标系原点重合。为了方便描述，将后三个关节合称为腕关节。在进行运动学逆解时，已知末端的位姿。设得到的工具位姿矩阵为式(2.6)。

$${}^0T_t = \begin{bmatrix} {}^0R_t & P_t \\ 0 & 1 \end{bmatrix} \quad (2.6)$$

对式(2.6)右乘  ${}^6T_t$  的逆，即可得到第 6 个关节坐标系的位姿矩阵。

$${}^0T_6 = {}^0T_t \cdot {}^6T_t^{-1} = \begin{bmatrix} R & P_w \\ 0 & 1 \end{bmatrix} \quad (2.7)$$

其中  $P_w$  为后三个坐标系原点 3\*1 位置向量。

将机械臂投影到 xy 平面，可得到如图 2-7(a)所示，其中  $O_c$  表示腕关节坐标系的原点。由图 2-7(a)可以应用三角函数计算得到关节 1 对于关节变量  $\theta_1$  的值，可表示为式(2.8)。

$$\theta_1 = a \tan 2(P_w(2), P_w(1)) \quad (2.8)$$

由于未明确关节 2、3 变量的值，故  $\theta_1$  还有可能为如图 2-7(b)所示。此时， $\theta_1$  可表示为式(2.9)。

$$\theta_1 = \pi + a \tan 2(P_w(2), P_w(1)) \quad (2.9)$$

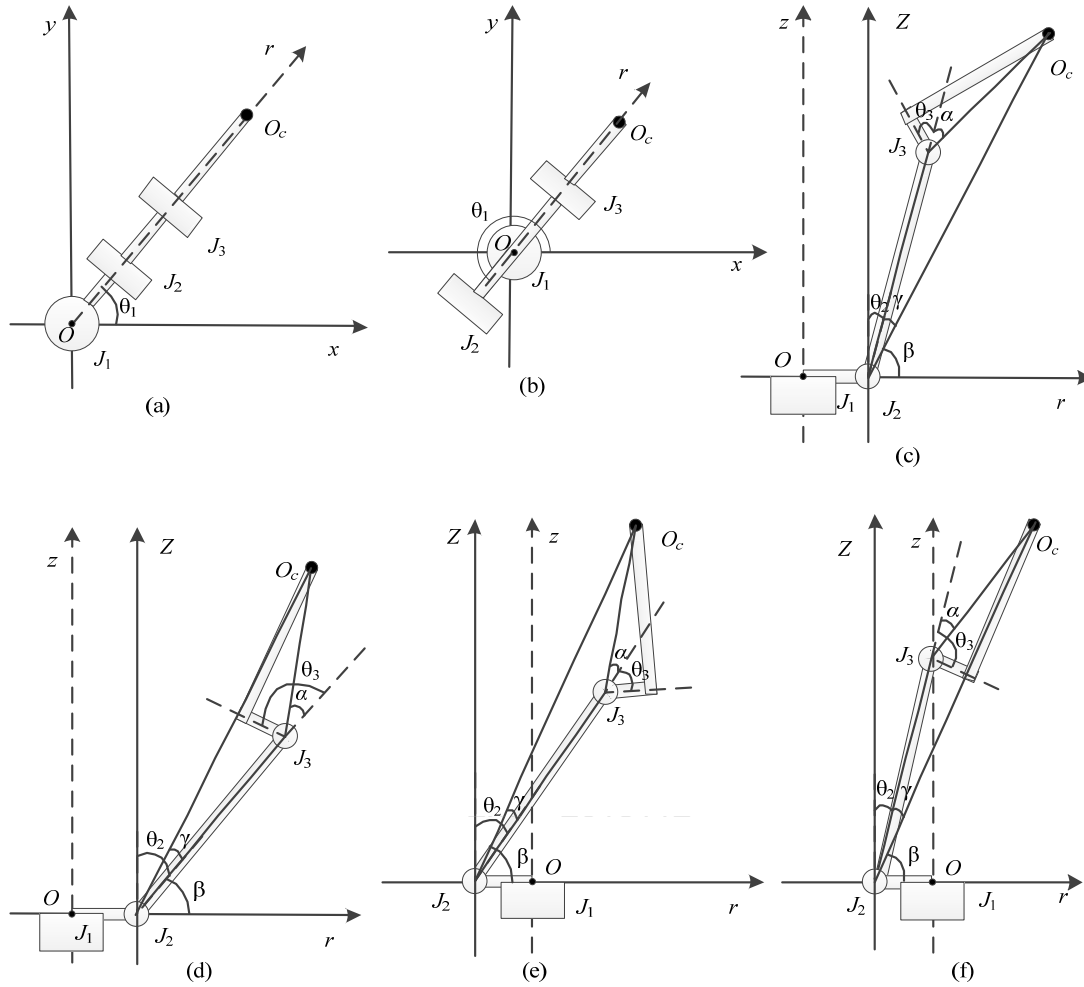


图 2-7 几何解法

Figure 2-7 Geometrical solution

得到关节 1 对应关节变量  $\theta_1$  的值后, 以直线  $OO_c$  为  $r$  轴, 将图 2-7(a)所示位形的机械臂投影到  $zOr$  平面可能会出现两种情况, 分别由图 2-7(c)与图 2-7(d)进行表示; 同样, 将图 2-7(b)所示位形的机械臂投影到  $zOr$  平面也可能出现两种情况, 分别由图 2-7(e)与图 2-7(f)进行表示。为了方便计算, 将  $z$  轴的原点平移到关节 2 坐标系原点处, 得到新坐标系  $ZOr$ 。对于图 2-7(c)(d),  $O_c$  点在  $ZOr$  坐标系内的坐标为  $(R, P_w(3))$ , 其中  $R$  可由式(2.10)得到。

$$R = \sqrt{P_w(1)^2 + P_w(2)^2 - d_1^2} \quad (2.10)$$

由图 2-7(c)中几何关系, 可以得到角  $\alpha$ 、 $\beta$ 、 $\gamma$  的角度值分别如式(2.11)~(2.13)所示。

$$\alpha = a \tan 2\left(\sqrt{1 - \cos \alpha^2}, \cos \alpha\right) \quad (2.11)$$

其中:

$$\cos \alpha = \frac{R^2 + P_w(3)^2 - d_2^2 - d_3^2 - r_4^2}{2 \cdot d_2 \cdot \sqrt{d_3^2 + r_4^2}};$$

$$\beta = a \tan 2(P_w(3), R); \quad (2.12)$$

$$\gamma = a \tan 2\left(\sqrt{1 - \cos \gamma^2}, \cos \gamma\right), \quad (2.13)$$

其中:

$$\cos \gamma = \frac{R^2 + P_w(3)^2 + d_2^2 - d_3^2 - r_4^2}{2 \cdot d_2 \cdot \sqrt{R^2 + P_w(3)^2}}$$

进而可以得到关节 3 的关节变量  $\theta_2$  的值, 如式(2.14)所示。

$$\theta_2 = \frac{\pi}{2} - \beta \mp \gamma \quad (2.14)$$

同理, 可以得到关节 3 的关节变量  $\theta_3$  的值, 如式(2.15)所示。

$$\theta_3 = -a \tan 2(d_3, r_4) \pm \alpha \quad (2.15)$$

其中 $\pm$ 分别对应图 2-7(c)与图 2-7(d)两种情况。根据机器人的常见位形, 在进行多值选择时, 优先选择图 2-7 中(a)(c)对应的情况。

得到关节 1~3 对应的关节角度后, 就可以得到相应的变换矩阵, 即  ${}^0T_3 = {}^0T_1 * {}^1T_2 * {}^2T_3$ 。进而可以得到只含有关节 4~6(即腕关节)对应的关节变量的变换矩阵, 即  ${}^3T_6 = {}^3T_3^{-1} * {}^0T_6$ , 其中旋转矩阵为式(2.16)所示。

$${}^3R_6 = \begin{bmatrix} -c6s4 - c4c5s6 & s4s6 - c4c5c6 & -c4s5 \\ -s5s6 & -c6s5 & c5 \\ c5s4s6 - c4c6 & c4s6 + c5c6s4 & s4s5 \end{bmatrix} \quad (2.16)$$

假设  $R_{ij}(i,j=1,2,3)$ 表示式(2.16)矩阵中第  $i$  行第  $j$  列对应的元素, 则腕关节对应三个关节的关节角如式(2.17)所示, 其中 $\pm$ 对应于关节 5 关节角的正负。同样, 根据六自由度机械臂常见位形, 旋转关节 5 关节角为正值, 其他两个对应选择。

$$\begin{aligned} \theta_5 &= a \tan 2\left(\pm\sqrt{1 - R23^2}, R23\right) \\ \theta_4 &= a \tan 2(\pm R33, \mp R13) \\ \theta_6 &= a \tan 2(\mp R21, \mp R22) \end{aligned} \quad (2.17)$$



(2)数值解法<sup>[16]</sup>

数值解法主要是利用微分的性质，即  $dX=Jdq$ ，其中  $X$  代表末端在笛卡尔空间中的位姿向量， $q$  代表机器人关节变量向量， $J$  为机器人的雅克比矩阵(雅克比矩阵的求解方法将在 2.4 节中进行介绍)。由于在机器人首次运行时，其逆解由几何解法进行确定，故运行过程中以当前关节角进行初始化，其过程可用图 2-8 进行表示。主要步骤如下：

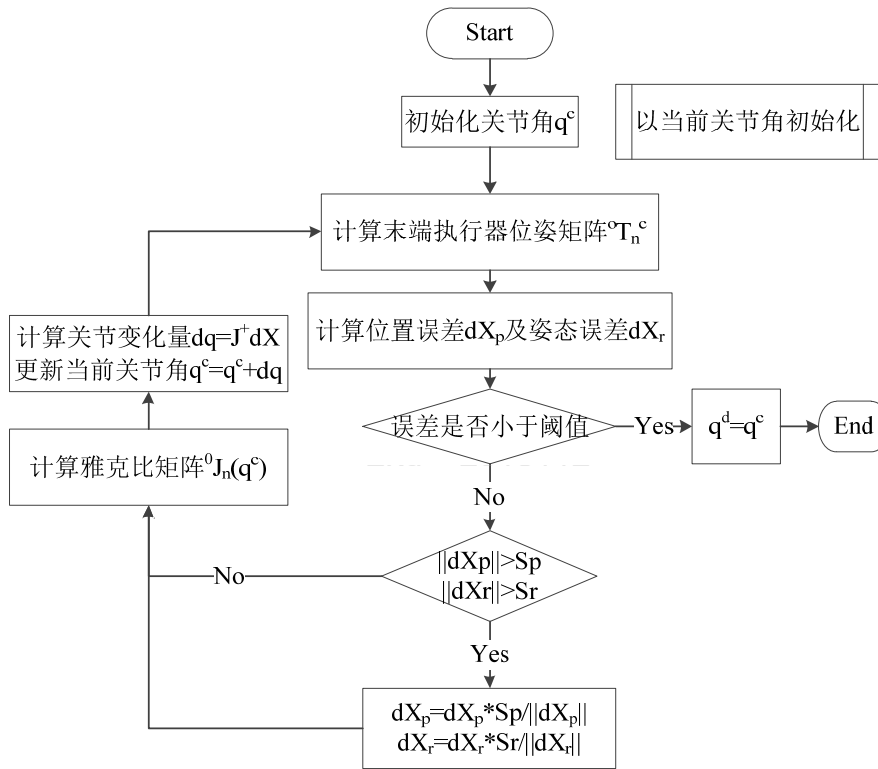


图 2-8 数值解法流程图

Figure 2-8 The flow chart of numerical solution

- (1) 以当前位形对  $q^c$  进行初始化。
- (2) 计算  $q^c$  对应的末端位姿  ${}^0T_t^c$ 。
- (3) 计算当前位姿  ${}^0T_t^c$  与期望位姿  ${}^0T_t^d$  之间的位置误差  $dX_p$  及姿态误差  $dX_r$ ，其中  $dX_p = p_n^d - p_n^c$ ， $dX_r = \mu\alpha$ 。角  $\alpha$  和单位向量  $\mu$  可由绕任意轴旋转得到， $\text{rot}(\mu, \alpha) = {}^0R_t^d * ({}^0R_t^c)^T$ 。
- (4) 如果  $dX_p$  及  $dX_r$  小于给定的阈值，则认为  $q^d = q^c$ ，结束；否则继续。
- (5) 为了使得到的位形位于可行域内，需要对  $dX_p$  及  $dX_r$  进行判断。如果  $dX_p$

及  $dX_r$  分别小于给定阈值  $S_p$  及  $S_r$ ，则直接进行(7)；否则，继续。

(6) 对  $dX_p$  及  $dX_r$  进行赋值，其中  $dX_p = S_p * dX_p / \|dX_p\|$  及  $dX_r = S_r * dX_r / \|dX_r\|$ 。

(7) 计算雅克比矩阵  $J = {}^0J(q^c)$ 。

(8) 计算关节变量变化量  $dq = J^+ dX$  ( $J^+$  为  $J$  的伪逆)，并更新当前关节角  $q^c = q^c + dq$ 。

(9) 返回(2)。

## 2.4 机器人运动学(速度及加速度)

要使机器人平稳运行，还需要了解机器人末端与关节变量之间速度与加速度的变换关系。与位置运动学相似，机器人速度及加速度运动学也存在正逆解，其关键在于雅克比矩阵。

### 2.4.1 雅克比矩阵

雅克比矩阵就是机器人末端与关节变量变化速度之间的变换矩阵。雅克比矩阵的推导有微分法<sup>[8]</sup>和物理法<sup>[16]</sup>，微分法从关节变量的微小运动导致机器人末端的位姿变化出发得到，物理法通过将关节速度所导致的旋转及平移速度逐级传递最终得到末端的速度。在此，只对第二种方法进行简要介绍。因为本文涉及的机器人为六自由度机械臂，由 6 个旋转关节进行驱动，在此将不对具有平移关节的情形进行介绍。

当关节  $i$  的速度为  $dq_i/dt$  时，对末端速度的影响如图 2-9 所示。对末端速度产生的增量可表达为式(2.18)。

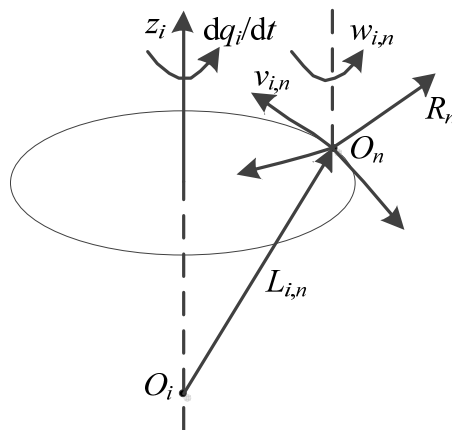


图 2-9 旋转关节的情形

Figure 2-9 The case of revolute joint

$$\begin{cases} v_{i,n} = (a_i \times L_{i,n}) \frac{dq_i}{dt} \\ \omega_{i,n} = a_i \frac{dq_i}{dt} \end{cases} \quad (2.18)$$

其中,  $a_i$  为沿着轴  $z_i$  的单位向量。

当机器人关节均为旋转关节, 末端的速度为式(2.19)。

$$\begin{cases} v_n = \sum_{i=1}^n (a_i \times L_{i,n}) \frac{dq_i}{dt} \\ \omega_n = \sum_{i=1}^n a_i \frac{dq_i}{dt} \end{cases} \quad (2.19)$$

将式(2.19)写为矩阵形式, 即式(2.20)。

$${}^0V_n = {}^0J_n \frac{dq}{dt} \quad (2.20)$$

其中雅克比矩阵为

$${}^0J_n = \begin{bmatrix} a_1 \times L_{1,n} & \cdots & a_n \times L_{n,n} \\ a_1 & \cdots & a_n \end{bmatrix}。$$

## 2.4.2 机器人速度及加速度运动学正逆解

### (1) 速度正逆解

机器人速度运动学正解是指根据机器人关节角的变化速度得到机器人末端的速度, 即式(2.21)。

$$\frac{dX}{dt} = J(q) \frac{dq}{dt} \quad (2.21)$$

其中  $J(q)$  为雅克比矩阵。本文所用实验平台机器人的雅克比矩阵见附录 2。

当计算速度逆解(即由末端速度求解关节速度)时, 可由式(2.22)进行。

$$\frac{dq}{dt} = J(q)^+ \frac{dX}{dt} \quad (2.22)$$

其中  $J(q)^+$  为雅克比矩阵的伪逆。

### (2) 加速度正逆解<sup>[49]</sup>

对式(2.21)进行关于时间  $t$  的求导, 就可以得到加速度正解的求解公式, 即式(2.23)。本文实验平台机器人雅克比矩阵关于时间的一阶导数矩阵结果见附录 3。

$$\frac{d^2 X}{dt^2} = J(q) \frac{d^2 q}{dt^2} + \left( \frac{d}{dt} J(q) \right) \frac{dq}{dt} \quad (2.23)$$

由式(2.23)可以得到加速度逆解的求解公式，即式(2.24)。

$$\frac{d^2 q}{dt^2} = J(q)^+ \left( \frac{d^2 X}{dt^2} - \left( \frac{d}{dt} J(q) \right) \frac{dq}{dt} \right) \quad (2.24)$$

## 2.5 本章小结

本章为本论文的基础章节，主要解决机器人的基本运动学算法问题。首先，介绍了所需要的数学基础，包括姿态的表示方法、DH 建模等；然后结合本论文的实验本体给出了机器人的运动学(位姿)正解及关于运动学(位姿)逆解的几何解法和数值解法；最后给出了雅克比矩阵的求解方法及速度和加速度运动学正逆解相关算法。

### 3 基于四元数样条曲线的姿态插补曲线生成

#### 3.1 引言

机器人末端一般具有平移自由度和旋转自由度。因为两种自由度为两种不同单位的量，不能够应用归一化的方法进行同时规划。因此在进行轨迹规划时一般对位置和姿态分别进行规划，然后再进行同步。机器人轨迹规划所遵循的一般步骤如图 3-1 所示。首先确定所要经过的关键点即示教点；然后根据实际需求确定示教点间路径的类型，并根据相应插补曲线算法计算插补点；接着进行速度规划，即根据系统限制(所能提供的最大速度、加速度等)、用户要求(运行时间、运行速度等)及速度规划方式(S 型、梯形等)得到位姿、速度、加速度等的时间序列；如果上述规划是在笛卡尔空间中进行的，则还需经过逆运动学计算出相应的关节角度、角速度及角加速度等，然后驱动关节电机进行运动。

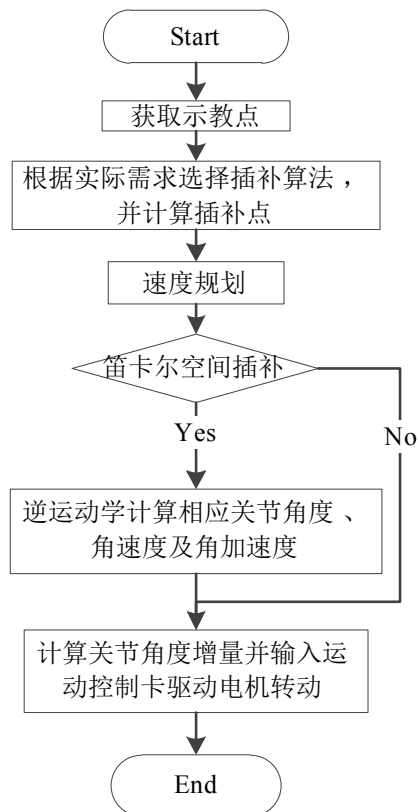


图 3-1 机器人轨迹规划流程

Figure 3-1 The process of robot trajectory planning

本章主要解决机器人末端姿态轨迹规划中的插补曲线生成的问题。机器人末端姿态的表示方法一般有欧拉角、RPY、旋转矩阵、四元数等<sup>[16]</sup>。相对于旋转矩阵和欧拉角，四元数具有某些方面的优势，如速度更快、提供平滑插值、存储空间较小、有效避免万向锁等问题<sup>[50]</sup>。因此，本文在进行姿态轨迹规划时，将以四元数为姿态表示工具，用四元数曲线作为插值曲线。为了实现此目的，本文首先介绍四元数基础，包括四元数较为特殊的基本运算、四元数函数求导及其与旋转的关系；然后进行四元数样条曲线的研究，主要包括曲线的形式、控制点的反求、节点的选取及样条曲线一二阶导数形式等；接着介绍四元数中的球面插补，并给出了球面插补的一二阶导数；最后，对多四元数插补曲线平滑过渡曲线生成相关算法进行了研究。

## 3.2 四元数基础

复数可以用作二维平面内旋转操作。为了将这种性质推广到三维空间，爱尔兰数学家哈密顿于 1843 年发现了四元数的概念。目前，四元数被应用于许多不同的领域来表述旋转和方位，如动画及图像处理<sup>[51]</sup>、控制论、信号处理<sup>[52]</sup>、姿态控制<sup>[53]</sup>、物理和轨道力学<sup>[54]</sup>等。

### 3.2.1 四元数基本运算及四元数函数求导

与复数相同，四元数由实部和虚部组成。设  $q$  为四元数，则可表述为  $q=s+xi+yj+zk$ ，其中  $i$ 、 $j$ 、 $k$  为虚单位，且满足  $i^2=j^2=k^2=ijk=-1$ 。四元数还可以表示为向量形式  $q=[s, \mathbf{u}]$ ，其中  $s$  为实部， $\mathbf{u}=[x \ y \ z]$  为三个实数构成的虚部向量。在本文中四元数只用来表示旋转，乘或除以一个系数之后所代表的旋转不变。故下文所针对的四元数均为单位四元数。单位四元数的向量形式又可表示为  $q=[\cos\theta, \sin\theta*\mathbf{v}]$ ，其中  $\mathbf{v}$  为单位向量表示的虚部向量。四元数的加减与实数向量相同，相同部分进行加减即可，但四元数乘法、除法等基本运算及函数求导方法与实数不尽相同。下面将对四元数较为特殊的运算及四元数函数求导进行介绍<sup>[38]</sup>。

共轭：设  $q=[s, \mathbf{u}]$  为四元数，则其共轭四元数为  $q^*=[s, -\mathbf{u}]$ 。

逆：对于一个不为 0 的四元数  $q=[s, \mathbf{u}]$ ，其逆  $q^{-1}=q^*/\|q\|^2$ ， $\|\bullet\|$  表示 2-范数。四元数范数的计算方法与实数向量的相同。单位四元数的 2-范数为 1，故单位四元数

的逆  $q^{-1} = q^*$ 。

乘法: 对应四元数  $q_1 = [s_1, \mathbf{u}_1]$  和  $q_2 = [s_2, \mathbf{u}_2]$ , 则有  $q_1 q_2 = [s_1 s_2 - \mathbf{u}_1 \bullet \mathbf{u}_2, \mathbf{u}_1 \times \mathbf{u}_2 + s_1 \mathbf{u}_2 + s_2 \mathbf{u}_1]$ , 其中  $\bullet$  和  $\times$  分别表示三维向量的内积和外积。并且四元数的乘法是不可交换的。

数乘: 对于  $q$  为四元数,  $r$  为实数, 则数乘定义为  $rq = [r, 0]q = [rs, r\mathbf{v}]$ 。

对数: 设  $q$  为单位四元数, 且  $q = [\cos\theta, \sin\theta \mathbf{v}]$ , 则其对数为  $\log q = [0, \theta \mathbf{v}]$ 。在此规定,  $\log[1, (0, 0, 0)] = [0, (0, 0, 0)]$ , 并有性质  $\log(q^t) = t \log q$ 。

指数: 对于形式为  $q = [0, \theta \mathbf{v}]$  的四元数, 其指数为  $\exp q = [\cos\theta, \sin\theta \mathbf{v}]$ 。由此可见, 四元数的指数与对数是互逆的, 并有  $q^t = \exp(t \log q)$ 。

四元数指数函数求导: 对于单位四元数  $q = [\cos\theta, \sin\theta \mathbf{v}]$  及  $u \in \mathbf{R}$ , 则有式(3.1)成立。

$$\frac{d}{du} q^u = q^u \log(q) \quad (3.1)$$

对于多个四元数函数相乘所构成的四元数函数求导如下:

设  $f, g \in C^1(\mathbf{R}, \mathbf{H})$ , 其中  $\mathbf{R}$  为实数域,  $\mathbf{H}$  为四元数域, 则有式(3.2)成立。

$$\frac{d}{du} (f(u)g(u)) = \frac{d}{du} (f(u))g(u) + f(u)\frac{d}{du} (g(u)) \quad (3.2)$$

对于复合四元数函数的求导如下:

设  $f \in C^1(\mathbf{H}, \mathbf{H})$ ,  $g \in C^1(\mathbf{R}, \mathbf{H})$ , 则有式(3.3)成立。

$$\frac{d}{du} (f(g(u))) = f'(g(u))g'(u) \quad (3.3)$$

对于四元数函数的指数为实数函数的情况, 其求导公式如下:

对于  $q \in C^1(\mathbf{R}, \mathbf{H}_1)$  (其中  $\mathbf{H}_1$  为单位四元数集),  $r \in C^1(\mathbf{R}, \mathbf{R})$ ,  $u \in \mathbf{R}$ 。因为  $q$  将实数映射为单位四元数, 因此  $q(u)$  可以表示为  $q(u) = [\cos\theta(u), \sin\theta(u)\mathbf{v}]$ , 因此有式(3.4)成立。

$$\frac{d}{du} q(u)^{r(u)} = \begin{bmatrix} -\sin(r(u)\theta(u))(r'(u)\theta(u) + r(u)\theta'(u)), \\ \cos(r(u)\theta(u))(r'(u)\theta(u) + r(u)\theta'(u))\mathbf{v}(u) + \sin(r(u)\theta(u))\mathbf{v}'(u) \end{bmatrix} \quad (3.4)$$

由于四元数的乘法是不可交换的, 因此要注意上述求导所得结果的顺序。

### 3.2.2 四元数与旋转

四元数与旋转的关系可用如下定理进行描述：设单位四元数  $q=[\cos\theta, \sin\theta*\mathbf{n}]$ ,  $p=[0, \mathbf{r}]$ , 其中  $\mathbf{r}=[x, y, z]$  为三维向量，则  $p'=qpq^{-1}$  为  $p$  绕轴  $\mathbf{n}$  旋转  $2\theta$ <sup>[55]</sup>。

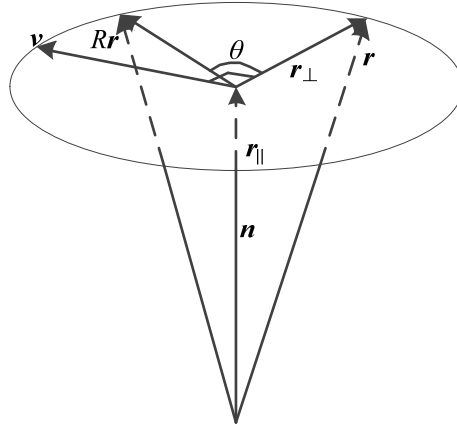


图 3-2 四元数与旋转的关系

Figure 3-2 The relationship of Quaternion and rotation

该定理的证明过程大致如下。

首先， $p$  绕轴  $\mathbf{n}$  旋转  $\theta$  可由图 3-2 进行表示。 $p$  即为图 3-2 中向量  $\mathbf{r}$ ，绕轴  $\mathbf{n}$  旋转  $\theta$  后变为  $R\mathbf{r}$ 。可以将  $\mathbf{r}$  分解为  $\mathbf{r}_\perp$  和  $\mathbf{r}_\parallel$ ，其中  $\mathbf{r}_\parallel = (\mathbf{r} \cdot \mathbf{n})\mathbf{n}$ ， $\mathbf{r}_\perp = \mathbf{r} - \mathbf{r}_\parallel$ 。为了证明，构造一个二维平面，即  $\mathbf{v}$  垂直于  $\mathbf{r}_\perp$  和  $\mathbf{n}$ ，则  $\mathbf{v} = \mathbf{n} \times \mathbf{r}$ 。进而， $(R\mathbf{r})_\perp = \mathbf{r}_\perp \cos\theta + \mathbf{v} \sin\theta$ 。又由图 3-2 可知式(3.5)成立。

$$\begin{aligned} R\mathbf{r} &= (R\mathbf{r})_\perp + (R\mathbf{r})_\parallel \\ &= (1 - \cos\theta)(\mathbf{r} \cdot \mathbf{n})\mathbf{n} + \mathbf{r} \cos\theta + (\mathbf{n} \times \mathbf{r}) \sin\theta \end{aligned} \quad (3.5)$$

然后，应用四元数基本运算计算  $Rq(p) = qpq^{-1}$ 。先记  $q = [\cos\theta, \sin\theta*\mathbf{n}] = [s, \mathbf{v}]$ ，则有式(3.6)。

$$\begin{aligned} Rq(p) &= [s, \mathbf{v}][0, \mathbf{r}][s, \mathbf{v}]^{-1} \\ &= [0, (s^2 - \mathbf{v} \cdot \mathbf{v})\mathbf{r} + 2(\mathbf{v} \cdot \mathbf{r})\mathbf{v} + 2s(\mathbf{v} \times \mathbf{r})] \end{aligned} \quad (3.6)$$

将  $q = [\cos\theta, \sin\theta*\mathbf{n}]$  带入式(3.6)可得式(3.7)。

$$Rq(p) = [0, \mathbf{r} \cos 2\theta + (1 - \cos 2\theta)(\mathbf{n} \cdot \mathbf{r})\mathbf{n} + (\mathbf{n} \times \mathbf{r}) \sin 2\theta] \quad (3.7)$$

对比式(3.5)与(3.7)可以发现，将式(3.5)中的  $\theta$  换为  $2\theta$ ，即为式(3.7)。因此，单位四元数  $[\cos\theta, \sin\theta*\mathbf{n}]$  可将任意向量  $\mathbf{r}$  绕轴  $\mathbf{n}$  旋转  $2\theta$ 。



### 3.2.3 四元数与旋转矩阵

在进行姿态轨迹插补时使用四元数对姿态进行表示，因此还需要了解四元数与旋转矩阵之间的变换。

#### (1) 四元数转换为旋转矩阵

一个绕轴  $\mathbf{n}$  旋转角度  $\theta$  的操作可以用单位四元数  $q=[q_1, q_2, q_3, q_4] = [\cos(\theta/2), \mathbf{n}\sin(\theta/2)]$  进行表示，其中  $\theta \in [0, \pi]$ 。由参考文献[16]第二章式[2.30]可知，绕任意轴  $\mathbf{n}$  旋转角度  $\theta$  的操作可用旋转矩阵如式(3.8)表示。

$$\text{rot}(\mathbf{n}, \theta) = \begin{bmatrix} n_x^2(1-C\theta) + C\theta & n_x n_y(1-C\theta) - n_z S\theta & n_x n_z(1-C\theta) + n_y S\theta \\ n_x n_y(1-C\theta) + n_z S\theta & n_y^2(1-C\theta) + C\theta & n_y n_z(1-C\theta) - n_x S\theta \\ n_x n_z(1-C\theta) - n_y S\theta & n_y n_z(1-C\theta) + n_x S\theta & n_z^2(1-C\theta) + C\theta \end{bmatrix} \quad (3.8)$$

因为  $q$  为单位四元数，故有式(3.9)成立。

$$q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1 \quad (3.9)$$

由四元数与绕任意轴旋转的关系可知式(3.10)成立。

$$C\theta = C^2(\theta/2) - S^2(\theta/2) = 2q_1^2 - 1 \quad (3.10)$$

故有式(3.11)成立。

$$\begin{cases} q_2^2 = \frac{1}{2}n_x^2(1-C\theta), q_3^2 = \frac{1}{2}n_y^2(1-C\theta), q_4^2 = \frac{1}{2}n_z^2(1-C\theta) \\ q_2q_3 = \frac{1}{2}n_x n_y(1-C\theta), q_2q_4 = \frac{1}{2}n_x n_z(1-C\theta), q_3q_4 = \frac{1}{2}n_y n_z(1-C\theta) \\ n_x S\theta = 2q_1q_2, n_y S\theta = 2q_1q_3, n_z S\theta = 2q_1q_4 \end{cases} \quad (3.11)$$

进而可以得到旋转矩阵，如式(3.12)所示。

$$R(q) = \begin{bmatrix} 2(q_1^2 + q_2^2) - 1 & 2(q_2q_3 - q_1q_4) & 2(q_2q_4 + q_1q_3) \\ 2(q_2q_3 + q_1q_4) & 2(q_1^2 + q_3^2) - 1 & 2(q_3q_4 - q_1q_2) \\ 2(q_2q_4 - q_1q_3) & 2(q_3q_4 + q_1q_2) & 2(q_1^2 + q_4^2) - 1 \end{bmatrix} \quad (3.12)$$

#### (2) 旋转矩阵转换为四元数

旋转矩阵转换为四元数问题即为求解式(2.2)相对应的四元数。由式(2.2)、式(3.12)及旋转矩阵和单位四元数的相关性质，可得式(3.13)。

$$q_1 = \frac{1}{2}\sqrt{s_x + n_y + a_z + 1} \quad (3.13)$$

虽然开方有正负，但在此取正。这是因为  $\theta \in [0, \pi]$ ，故  $q_1 = \cos\theta > 0$ 。将式(3.12)中元素(1,1)减去元素(2,2)及元素(3,3)，可得式(3.14)。

$$4q_2^2 = s_x - n_y - s_z + 1 \quad (3.14)$$

式(3.14)给出了  $q_2$  的大小，还需要给出其符号。将(3.12)中的元素(3,2)与元素(2,3)相减可得式(3.15)。

$$4q_1q_2 = n_z - a_y \quad (3.15)$$

元素  $q_1$  始终为正，因此  $q_2$  的符号即为  $n_z - a_y$  的符号。 $q_2$  可用式(3.16)进行求解。

$$q_2 = \frac{1}{2} \text{sign}(n_z - a_y) \sqrt{s_x - n_y - a_z + 1} \quad (3.16)$$

类似地， $q_3$ 、 $q_4$  可由式(3.17)及式(3.18)进行求解。

$$q_3 = \frac{1}{2} \text{sign}(a_x - s_z) \sqrt{-s_x + n_y - a_z + 1} \quad (3.17)$$

$$q_4 = \frac{1}{2} \text{sign}(s_y - n_x) \sqrt{-s_x - n_y + a_z + 1} \quad (3.18)$$

由以上可以看出，用四元数进行表示姿态不会出现奇异。

### 3.3 四元数样条曲线

用四元数表示姿态虽然具有诸多优点，但是不能将四元数当作实数向量那样直接进行实数空间诸如线性、圆弧等的插补，也不能直接将空间 B 样条曲线直接照搬过来应用到四元数样条曲线插补上。这是因为用来表示姿态的四元数为单位四元数，直接将实数空间位置插补的方法平移过来得到的插补结果将不是单位四元数。为了实现像空间位置一样的插补并且保持四元数的单位性，将对四元数在四维单位球面上进行插补。在进行多点位置插补时，往往会利用 B 样条曲线及其衍生的 NURBS 曲线等。这是因为 B 样条具有凸包性、仿射不变性，构造灵活，并且求导简便。Myoung-Jun Kim 等<sup>[42, 43, 56]</sup>将空间样条曲线推广到了四元数空间，得到了四元数样条曲线的表达式。本节正是在其工作的基础上进行的。

#### 3.3.1 曲线形式

本文采用的四元数样条曲线由 Myoung-Jun Kim 等提出，具体推导过程可参见文献[42]。对于给定控制点  $(q_0, q_1, \dots, q_n)$  的  $k$  次四元数曲线的表达式如式(3.19)所示。

$$q(u) = q_0^{\tilde{N}_{0,k}(u)} \prod_{i=1}^n \exp(\tilde{N}_{i,k}(u) \log(q_{i-1}^{-1} q_i)) \quad (3.19)$$

$$\begin{aligned} \text{其中 } \tilde{N}_{i,k}(u) &= \sum_{j=i}^n N_{j,k}(u) \\ &= \begin{cases} \sum_{j=i}^n N_{j,k}(u) & u_i < u < u_{i+k-1} \\ 1 & u \geq u_{i+k-1} \\ 0 & u \leq u_{i+k-1} \end{cases} \circ \end{aligned}$$

$N_{j,k}(t)$ 为空间  $k$  次 B 样条曲线的第  $j$  个基函数。

由空间 B 样条曲线的性质可知，对于  $k$  次空间 B 样条曲线来说，同时不为零的基函数至多有  $k+1$  个。本文中主要研究三次四元数样条曲线，故式(3.19)又可以简化为式(3.20)。

$$q(u) = q_{i-3} \prod_{k=3}^0 \exp(\tilde{N}_{i-k,3}(u) \log(q_{i-k-1}^{-1} q_{i-k})) \quad (3.20)$$

其中：

$$\begin{aligned} \tilde{N}_{i-2,3}(u) &= N_{i-3,3}(u) + N_{i-2,3}(u) + N_{i-1,3}(u) + N_{i,3}(u); \\ \tilde{N}_{i-2,3}(u) &= N_{i-2,3}(u) + N_{i-1,3}(u) + N_{i,3}(u); \\ \tilde{N}_{i-1,3}(u) &= N_{i-1,3}(u) + N_{i,3}(u); \\ \tilde{N}_{i,3}(u) &= N_{i,3}(u). \end{aligned}$$

又知，对于  $k$  次空间 B 样条曲线来说，同时不为零的  $k+1$  个基函数和为 1。故在计算时，对于三次四元数样条曲线，式(3.20)又可以简化为式(3.21)。

$$q(u) = q_{i-3} (q_{i-3}^{-1} q_{i-2})^{\tilde{N}_{i-2,3}} (q_{i-2}^{-1} q_{i-1})^{\tilde{N}_{i-1,3}} (q_{i-1}^{-1} q_i)^{\tilde{N}_{i,3}} \quad (3.21)$$

### 3.3.2 节点矢量

空间 B 样条曲线具有如下性质：对于  $k$  次 B 样条曲线，其控制点数  $m$  与节点数  $n$  之间的关系为  $n=m+k+1$ 。四元数样条曲线是由空间 B 样条曲线得来的，同样具有该性质。当给定  $n$  个示教点  $Q_i(i=0,1,2,\dots,n-1)$  时，对于三次均匀 B 样条四元数曲线，共有  $n+2$  个控制点需要反求，同时节点矢量最大下标为  $n+5$ ，节点矢量长度为  $n+6$ 。此外，空间 B 样条中，若要使首尾控制点与首尾示教点分别重合，则首尾处节点的重数应分别为  $k$ 。因此，三次四元数样条曲线首末各 4 个节点对应相等，构成三重节点，中间节点参数可以利用均匀节点矢量法求得。其中首末重节点为式(3.22)，

$$\begin{aligned} u_0 = u_1 = u_2 = u_3 &= 0 \\ u_{n+2} = u_{n+3} = u_{n+4} = u_{n+5} &= 1 \end{aligned} \quad (3.22)$$

中间节点为式(3.23),

$$u_{i+4} = u_{i+3} + \frac{1}{n-1} \quad (3.23)$$

其中,  $i=0, \dots, n-1$ 。

### 3.3.3 控制点反求

在对机械臂进行示教时, 机械臂末端姿态必须经过给定示教点, 这就使得用于姿态插补的四元数样条曲线要在第一个示教点出开始, 终止于最后一个示教点, 并且经过中间每一个示教点。实际情况中往往只给定示教点, 这就需要由示教点来反求控制点。但由于四元数样条曲线为非线性方程, 不能通过解线性方程组的方式将控制点求解出来。文献[43]给出了一种反求控制点的迭代方法。

给定  $n-1$  个示教点  $Q_i$  ( $i=1, \dots, n-1$ ), 由式(3.20)可以得到  $n-3$  个迭代方程, 如式(3.24)所示。

$$\begin{aligned} q_1 &= Q_1 \\ q_i^* &= q_{i-1} \left[ q_{i-1}^{-1} Q_i (q_i^{-1} q_{i+1})^{-1/6} \right]^{6/5}, i=2, \dots, n-2 \\ q_{n-1} &= Q_{n-1} \end{aligned} \quad (3.24)$$

边界的两个控制点可以通过式(3.25)求得。

$$q_0 = q_1 (q_1^{-1} q_2)^{-1}, q_n = q_{n-1} (q_{n-2}^{-1} q_{n-1})^{-1} \quad (3.25)$$

需要指出的是, 该方法虽然能得到控制点, 但求得的首末控制点不与首末示教点重合, 示教点也不完全与节点对应的点相重合, 这都使得该方法求出的四元数样条曲线无法应用在机械臂末端姿态插补中; 此外首末示教点处曲线的曲率变化非常大, 可能会超过机械臂的运动极限, 并导致机械臂震动。为了能够适应机械臂末端姿态的插补, 本文对上述算法进行了改进。以首末示教点为首末控制点, 并且在第一个与第二个示教点间、最末两个示教点间采用球面线性插补方法各增加一个示教点, 这样就使得曲线的首末端点与首末示教点重合, 并且样条曲线首尾位置的曲率为 1。同时, 由于算法的原因, 该算法只能用于均匀四元数样条曲线的控制点反求, 而不能用来反求解非均匀四元数样条曲线的控制点。

### 3.3.4 四元数样条曲线导数

四元数样条曲线的表达式见式(3.20)所示。根据上述四元数函数的导数中式(3.1)~(3.3)以及四元数乘法的不可交换原则，可得四元数样条曲线的一阶导数如式(3.26)所示。

$$\begin{aligned} \frac{d}{du}q(u) &= q_{i-3} \left( q_{i-3}^{-1} q_{i-2} \right)^{\tilde{N}_{i-2,3}} \tilde{N}_{i-2,3}'(u) \log \left( q_{i-3}^{-1} q_{i-2} \right) \left( q_{i-2}^{-1} q_{i-1} \right)^{\tilde{N}_{i-1,3}} \left( q_{i-1}^{-1} q_i \right)^{\tilde{N}_{i,3}} \\ &\quad + q_{i-3} \left( q_{i-3}^{-1} q_{i-2} \right)^{\tilde{N}_{i-2,3}} \left( q_{i-2}^{-1} q_{i-1} \right)^{\tilde{N}_{i-1,3}} \tilde{N}_{i-1,3}'(u) \log \left( q_{i-2}^{-1} q_{i-1} \right) \left( q_{i-1}^{-1} q_i \right)^{\tilde{N}_{i,3}} \\ &\quad + q_{i-3} \left( q_{i-3}^{-1} q_{i-2} \right)^{\tilde{N}_{i-2,3}} \left( q_{i-2}^{-1} q_{i-1} \right)^{\tilde{N}_{i-1,3}} \left( q_{i-1}^{-1} q_i \right)^{\tilde{N}_{i,3}} \tilde{N}_{i,3}'(u) \log \left( q_{i-1}^{-1} q_i \right) \end{aligned} \quad (3.26)$$

其中， $\tilde{N}_{i-2,3}, \tilde{N}_{i-1,3}, \tilde{N}_{i,3}$  同式(3.20)；

$$\tilde{N}_{i-2,3}'(u) = \frac{3}{u_{i+1} - u_{i-2}} N_{i-2,2}(u),$$

$$\tilde{N}_{i-1,3}'(u) = \frac{3}{u_{i+2} - u_{i-1}} N_{i-1,2}(u),$$

$$\tilde{N}_{i,3}'(u) = \frac{3}{u_{i+3} - u_i} N_{i,2}(u)。$$

类似地，可以得到四元数样条曲线的二阶导数表达式如式(3.27)所示。

$$\begin{aligned} \frac{d^2}{du^2}q(u) &= q_{i-3} \left( q_{i-3}^{-1} q_{i-2} \right)^{\tilde{N}_{i-2,3}} \left( \tilde{N}_{i-2,3}'(u) \log \left( q_{i-3}^{-1} q_{i-2} \right) \right)^2 \left( q_{i-2}^{-1} q_{i-1} \right)^{\tilde{N}_{i-1,3}} \left( q_{i-1}^{-1} q_i \right)^{\tilde{N}_{i,3}} \\ &\quad + q_{i-3} \left( q_{i-3}^{-1} q_{i-2} \right)^{\tilde{N}_{i-2,3}} \left( q_{i-2}^{-1} q_{i-1} \right)^{\tilde{N}_{i-1,3}} \left( \tilde{N}_{i-1,3}'(u) \log \left( q_{i-2}^{-1} q_{i-1} \right) \right)^2 \left( q_{i-1}^{-1} q_i \right)^{\tilde{N}_{i,3}} \\ &\quad + q_{i-3} \left( q_{i-3}^{-1} q_{i-2} \right)^{\tilde{N}_{i-2,3}} \left( q_{i-2}^{-1} q_{i-1} \right)^{\tilde{N}_{i-1,3}} \left( q_{i-1}^{-1} q_i \right)^{\tilde{N}_{i,3}} \left( \tilde{N}_{i,3}'(u) \log \left( q_{i-1}^{-1} q_i \right) \right)^2 \\ &\quad + q_{i-3} \left( q_{i-3}^{-1} q_{i-2} \right)^{\tilde{N}_{i-2,3}} \tilde{N}_{i-2,3}''(u) \log \left( q_{i-3}^{-1} q_{i-2} \right) \left( q_{i-2}^{-1} q_{i-1} \right)^{\tilde{N}_{i-1,3}} \left( q_{i-1}^{-1} q_i \right)^{\tilde{N}_{i,3}} \\ &\quad + q_{i-3} \left( q_{i-3}^{-1} q_{i-2} \right)^{\tilde{N}_{i-2,3}} \left( q_{i-2}^{-1} q_{i-1} \right)^{\tilde{N}_{i-1,3}} \tilde{N}_{i-1,3}''(u) \log \left( q_{i-2}^{-1} q_{i-1} \right) \left( q_{i-1}^{-1} q_i \right)^{\tilde{N}_{i,3}} \\ &\quad + q_{i-3} \left( q_{i-3}^{-1} q_{i-2} \right)^{\tilde{N}_{i-2,3}} \left( q_{i-2}^{-1} q_{i-1} \right)^{\tilde{N}_{i-1,3}} \left( q_{i-1}^{-1} q_i \right)^{\tilde{N}_{i,3}} \tilde{N}_{i,3}''(u) \log \left( q_{i-1}^{-1} q_i \right) \\ &\quad + 2q_{i-3} \left( q_{i-3}^{-1} q_{i-2} \right)^{\tilde{N}_{i-2,3}} \tilde{N}_{i-2,3}'(u) \log \left( q_{i-3}^{-1} q_{i-2} \right) \left( q_{i-2}^{-1} q_{i-1} \right)^{\tilde{N}_{i-1,3}} \tilde{N}_{i-1,3}'(u) \log \left( q_{i-2}^{-1} q_{i-1} \right) \left( q_{i-1}^{-1} q_i \right)^{\tilde{N}_{i,3}} \\ &\quad + 2q_{i-3} \left( q_{i-3}^{-1} q_{i-2} \right)^{\tilde{N}_{i-2,3}} \tilde{N}_{i-2,3}'(u) \log \left( q_{i-3}^{-1} q_{i-2} \right) \left( q_{i-2}^{-1} q_{i-1} \right)^{\tilde{N}_{i-1,3}} \left( q_{i-1}^{-1} q_i \right)^{\tilde{N}_{i,3}} \tilde{N}_{i,3}'(u) \log \left( q_{i-1}^{-1} q_i \right) \\ &\quad + 2q_{i-3} \left( q_{i-3}^{-1} q_{i-2} \right)^{\tilde{N}_{i-2,3}} \left( q_{i-2}^{-1} q_{i-1} \right)^{\tilde{N}_{i-1,3}} \tilde{N}_{i-1,3}'(u) \log \left( q_{i-2}^{-1} q_{i-1} \right) \left( q_{i-1}^{-1} q_i \right)^{\tilde{N}_{i,3}} \tilde{N}_{i,3}'(u) \log \left( q_{i-1}^{-1} q_i \right) \end{aligned} \quad (3.27)$$

其中， $\tilde{N}_{i-2,3}, \tilde{N}_{i-1,3}, \tilde{N}_{i,3}$  同式(3.20)；

$\tilde{N}_{i-2,3}', \tilde{N}_{i-1,3}', \tilde{N}_{i,3}'$  同式(3.26)；

$$N_{i-2,2}''(u) = \frac{2}{u_i - u_{i-2}} N_{i-2,1}(u) - \frac{2}{u_{i+1} - u_{i-1}} N_{i-1,1}(u) = -\frac{2}{u_{i+1} - u_{i-1}} N_{i-1,1}(u),$$

$$N_{i-1,2}''(u) = \frac{2}{u_{i+1} - u_{i-1}} N_{i-1,1}(u) - \frac{2}{u_{i+2} - u_i} N_{i,1}(u),$$

$$N_{i,2}''(u) = \frac{2}{u_{i+2} - u_i} N_{i,1}(u) - \frac{2}{u_{i+3} - u_{i+1}} N_{i+1,1}(u) = \frac{2}{u_{i+2} - u_i} N_{i,1}(u)。$$

### 3.4 基于四元数的球面线性插补及其导数

四元数可以用来表示机器人姿态，并且无奇异。单位四元数与其同方向上的四元数(即乘以不为零的正实数得到的四元数)所具备的旋转功能相同。在 3.3 中，给出了在四维球面上单位四元数插补的四元数样条曲线生成相关算法来适应多示教点并且对路径没有要求时的姿态轨迹规划。类似于空间位置两点间轨迹规划中最简单的情形——线性插补，在两个姿态示教点间也可进行基于四元数的球面线性插值。

#### 3.4.1 四元数球面线性插补

四元数两点间线性插补有两种方法，即 Lerp 和 Slerp<sup>[38]</sup>。四元数可以表示为向量形式，Lerp 即直接利用空间位置的线性插补方法，其表达式可以用式(3.28)进行表示。

$$\text{Lerp}(q_0, q_1, u) = q_0(1-u) + q_1u \quad (3.28)$$

其中， $q_0$ 、 $q_1$  分别为姿态插值的起始点和终止点， $u$  为参数，且  $u \in [0,1]$ 。Lerp 插值可以表示为图 3-3(a)所示。由图 3-3(a)可以看出，虽然该方法非常简便且直观，但所得到的单位四元数并不是等间距的，即在单位四元数空间下不是线性的。为了达到单位四元数空间下的线性插值，得到的结果必须为如图 3-3(b)所示，此即为单位球面线性插补 Slerp。

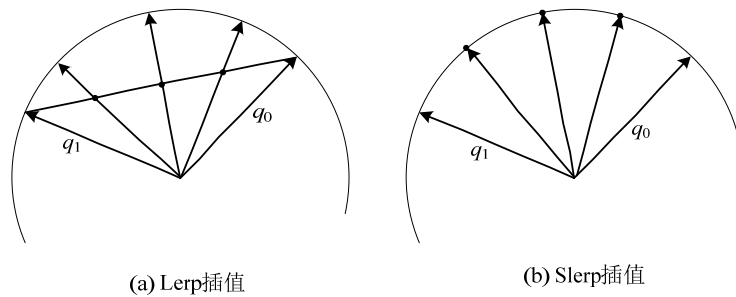


图 3-3 两个姿态示教点之间插补

Figure 3-3 Interpolation between two orientation teach points

类比空间位置的线性插值，得到单位四元数的单位球面线性插补的 Slerp 函数可由式(3.29)进行表示。

$$\text{Slerp}(q_0, q_1, u) = q_0 (q_0^{-1} q_1)^u \quad (3.29)$$

式(3.29)为理论上 Slerp 插值公式，但在实际应用中需要更为高效的计算方法。如图 3-4，插值点  $q$  可用式(3.30)进行表示。

$$q = k_0 v_0 + k_1 v_1 \quad (3.30)$$

由如图 3-4 所示的几何关系可以得式(3.31)，其中  $\theta$  为四元数  $q_0$  与  $q_1$  之间的夹角，求法与向量夹角相同。

$$\sin \theta = \frac{\sin u \theta}{k_1} \quad (3.31)$$

则  $k_1$  可表示为式(3.32)。

$$k_1 = \frac{\sin u \theta}{\sin \theta} \quad (3.32)$$

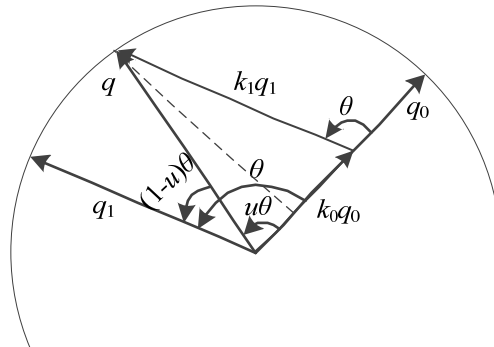


图 3-4 Slerp 插值公式推导

Figure 3-4 Formula derivation of Slerp interpolation

同理，可解出  $k_0$  如式所示。

$$k_0 = \frac{\sin(1-u)\theta}{\sin \theta} \quad (3.33)$$

故 Slerp 插值公式又可写为式(3.34)。

$$\text{Slerp}(q_0, q_1, u) = \frac{\sin(1-u)\theta}{\sin \theta} q_0 + \frac{\sin u \theta}{\sin \theta} q_1 \quad (3.34)$$

相较于式(3.29)，由于式(3.34)不涉及到四元数的幂、逆及乘法，只有数乘及加法，计算量相对较小，可以应用在对计算量要求严格的实时环境中。

### 3.4.2 四元数球面线性插补的导数

球面线性插补的函数表达式如式(3.34)所示。由于  $\text{Slerp} \in C^2(\mathbf{R}, \mathbf{H})$ ，则可得球面线性插补的一阶导数为式(3.35)所示。

$$\frac{d}{du} \text{slerp}(q_0, q_1, u) = -\frac{\cos(1-u)\theta}{\sin \theta} \theta q_0 + \frac{\cos u \theta}{\sin \theta} \theta q_1 \quad (3.35)$$

同理，球面线性插补的二阶导数为式(3.36)所示。

$$\begin{aligned} \frac{d^2}{du^2} \text{slerp}(q_0, q_1, u) &= -\frac{\sin(1-u)\theta}{\sin \theta} \theta^2 q_0 - \frac{\sin u \theta}{\sin \theta} \theta^2 q_1 \\ &= -\theta^2 \text{slerp}(q_0, q_1, u) \end{aligned} \quad (3.36)$$

### 3.5 平滑过渡曲线的生成

类似于空间位置插补，除了研究两点间姿态线性插补、多姿态示教点样条曲线插补，根据具体情况很多时候还需要进行不同插值曲线之间的平滑过渡，如球面线性插补——球面线性插补之间的平滑过渡、球面插补——四元数样条曲线插补之间的平滑过渡及多四元数样条曲线之间的平滑过渡等。但，不论是何种四元数曲线之间的平滑过渡，过渡曲线都要与前后两段插补曲线相接，并且相接处的一二阶导数要与被连接曲线相等。因此，可以将平滑过渡曲线的生成简化为根据起始点及其一二阶导数求四元数样条曲线问题，所得四元数样条曲线即为平滑过渡曲线。

如图 3-5 所示，AB、BC 分别为由示教点生成的相接于 B 的两条四元数曲线。为了生成两条曲线的平滑过渡曲线，首先需要得到过渡曲线的首末点，即与两条四元数曲线的交点 D、E。D、E 的选取可根据实际应用所要求近似程度进行选取。得到 D、E 后，相应点处的一二阶导数可同样求取。

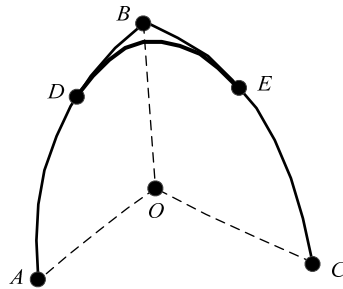


图 3-5 平滑过渡曲线生成示意图

Figure 3-5 schematic diagram of smooth transition curve generation



设 D、E 点处对应单位四元数及其一二阶导数分别为  $Q_{d0}$ 、 $Q_{d1}$ 、 $Q_{d2}$ 、 $Q_{e0}$ 、 $Q_{e1}$ 、 $Q_{e2}$ 。根据上述信息共可以得到六个方程，故可以得到六个控制点，设为  $[q_0, q_1, q_2, q_3, q_4, q_5]$ 。所得四元数样条曲线为三阶曲线，根据控制点数、节点数及次数之间的等式关系，可得节点矢量为  $U=[0, 0, 0, 0, 1/3, 2/3, 1, 1, 1, 1]$ 。

根据四元数样条曲线(3.21)，当节点  $u=0$  时， $\tilde{N}_{1,3}(0)=1$ ， $\tilde{N}_{2,3}(0)=\tilde{N}_{3,3}(0)=0$ ；当节点  $u=1$  时， $\tilde{N}_{3,3}(1)=\tilde{N}_{4,3}(1)=\tilde{N}_{5,3}(1)=1$ ，可得到式(3.37)。

$$q(0)=q_0, q(1)=q_5 \quad (3.37)$$

其中， $q(0)=Q_{d0}$ ， $q(1)=Q_{e0}$ 。

根据四元数样条曲线的一阶导数(3.26)，当节点  $u=0$  时， $\tilde{N}'_{1,3}(0) \neq 0$ ， $\tilde{N}'_{2,3}(0)=\tilde{N}'_{3,3}(0)=0$ ；当节点  $u=1$  时， $\tilde{N}'_{3,3}(0)=\tilde{N}'_{4,3}(0)=0$ ， $\tilde{N}'_{5,3}(0) \neq 0$ ，可得式(3.38)。

$$\begin{aligned} \frac{d}{du}q(0) &= q_0 \tilde{N}'_{1,3}(0) \log(q_0^{-1} q_1) \\ \frac{d}{du}q(1) &= q_4 \tilde{N}'_{5,3}(1) \log(q_4^{-1} q_5) \end{aligned} \quad (3.38)$$

其中，

$$\frac{d}{du}q(0)=Q_{d1}, \quad \frac{d}{du}q(1)=Q_{e1};$$

$$\tilde{N}'_{1,3}(0)=\frac{3}{U(5)-U(2)},$$

$$\tilde{N}'_{5,3}(1)=\frac{3}{U(9)-U(6)}。$$

根据四元数样条曲线的二阶导数(3.27)，当节点  $u=0$  时， $\tilde{N}''_{1,3}(0) \neq 0$ ， $\tilde{N}''_{2,3}(0) \neq 0$ ， $\tilde{N}''_{3,3}(0)=0$ ；当节点  $u=1$  时， $\tilde{N}''_{3,3}(0)=0$ ， $\tilde{N}''_{4,3}(0) \neq 0$ ， $\tilde{N}''_{5,3}(0) \neq 0$ ，可得式(3.39)。

$$\begin{aligned} \frac{d^2}{du^2}q(0) &= q_0 \left( \tilde{N}'_{1,3}(0) \log(q_0^{-1} q_1) \right)^2 + q_0 \tilde{N}''_{1,3}(0) \log(q_0^{-1} q_1) + q_0 \tilde{N}''_{2,3}(0) \log(q_1^{-1} q_2) \\ \frac{d^2}{du^2}q(1) &= q_5 \left( \tilde{N}'_{5,3}(1) \log(q_4^{-1} q_5) \right)^2 + q_5 \tilde{N}''_{5,3}(1) \log(q_4^{-1} q_5) + q_4 \tilde{N}''_{4,3}(1) \log(q_3^{-1} q_4) (q_4^{-1} q_5) \end{aligned} \quad (3.39)$$

其中，

$$\frac{d^2}{du^2}q(0)=Q_{d2}, \quad \frac{d^2}{du^2}q(1)=Q_{e2};$$

$$\tilde{N}'_{1,3}(0)、\tilde{N}'_{5,3}(1) \text{ 同式(3.38);}$$

$$\tilde{N}_{1,3}^*(0) = -\frac{2}{U(5)-U(3)},$$

$$\tilde{N}_{5,3}^*(1) = \frac{2}{U(8)-U(6)},$$

$$\tilde{N}_{2,3}^*(0) = -\tilde{N}_{1,3}^*(0), \quad \tilde{N}_{4,3}^*(1) = -\tilde{N}_{5,3}^*(1)。$$

联立式(3.37)~(3.39)，即可解得六个控制点，进而得到平滑过渡四元数样条曲线。

### 3.6 本章小结

本章为本论文的重点章节之一。本章主要对基于四元数的姿态插补曲线生成等相关算法进行了研究。首先给出了与本文密切相关的四元数相关基础，主要包括四元数的逆、共轭、乘法，单位四元数的对数及指数，四元数函数求导等，给出了四元数与旋转之间的关系及相关证明，并就单位四元与旋转矩阵之间的转换进行了研究；然后，本章重点对四元数样条曲线的生成进行了研究，主要包括四元数样条曲线的表达形式及含义、已知示教点对控制点进行反求及节点矢量的构造，并给出了四元数样条曲线的一二阶导数形式等；接着，给出了四元数球面线性插补及其一二阶导数的相关内容；最后，对四元数插补平滑过渡曲线的求解算法进行了研究，并得到了其一般表达形式。本章为基于四元数的姿态轨迹规划的基础，下一章将基于本章的相关研究成果进行给定示教点的姿态轨迹规划相关技术研究。

## 4 基于四元数的姿态轨迹规划算法研究

### 4.1 引言

在第 3 章中, 已经对基于四元数的姿态轨迹插值曲线生成进行了研究。本章将在上一章的基础上进行姿态轨迹规划算法的相关研究。在本章中, 将对曲线拟合、S 型速度规划等进行研究。速度规划的算法主要包括以下几种: 直线速度规划、指数速度规划、三角函数速度规划、S 曲线速度规划等<sup>[57]</sup>。为了减少末端的震荡, 应使规划的结果具有连续的速度及加速度。典型 S 型曲线速度曲线有加加速、匀加速、减加速、匀速、加减速、匀减速、减减速等七个阶段, 规划出来的结果速度及加速度连续, 并且加减速控制能达到较高的柔性, 给实际应用情况的不同留有很多的规划余地。与空间位置轨迹规划不同, 在四元数空间进行姿态轨迹规划所得到的结果并不能与机器人末端姿态的角度、角速度及角加速度直接对等, 因此还需要研究从四元数空间向姿态角度空间进行转换的相关算法和标量速度及加速度向矢量转换的问题。

### 4.2 四元数曲线拟合

在位置轨迹规划中, 在将位置轨迹曲线(或直线)确定后, 需要将轨迹拟合为路程, 然后进行速度规划。但本文进行姿态轨迹生成是基于四元数及其曲线进行的, 而四元数的导数与机器人末端姿态角速度并非直接对等的, 因此还需要对四元数曲线导数与姿态角速度之间的关系进行研究。在本节中将主要对导数与姿态角速度之间的关系及角位移拟合进行研究。

#### 4.2.1 四元数曲线导数与角速度及角加速度之间的关系

在本文中所采用的速度规划方式为 S 型速度规划, 不仅需要了解四元数与旋转角度的关系, 还需要对四元数导数与姿态角速度及角加速度之间的关系进行研究。四元数及其导数与角速度的关系可表示为式(4.1)所示<sup>[58]</sup>。

$$q'(u) = \frac{1}{2}w(u)q(u) \quad (4.1)$$

其中,  $w(u) = (w_x(u), w_y(u), w_z(u))^T \in \mathbf{R}^3$ ,  $q(u) \in C^1(\mathbf{R}, \mathbf{H})$ 。

进而，由式(4.1)可得式(4.2)。

$$w(u) = 2q'(u)q^{-1}(u) \quad (4.2)$$

因为在本文中用于姿态插值的为单位四元数，且对应单位四元数有式(4.3)成立。

$$\begin{aligned} q(u)^{-1} &= q(u)^* \\ \frac{d}{du}(q(u)^{-1}) &= \left( \frac{d}{du}q(u) \right)^{-1} \end{aligned} \quad (4.3)$$

因此对式(4.2)求导，可得角加速度与四元数及其一二阶导数之间的关系为式(4.4)所示。

$$\begin{aligned} \alpha(u) &= 2q''(u)q^{-1}(u) + 2q'(u)(q^{-1}(u))' \\ &= 2q''(u)q^{-1}(u) + 2 \begin{bmatrix} \|q'(u)\|^2 \\ \mathbf{0} \end{bmatrix} \end{aligned} \quad (4.4)$$

其中， $\mathbf{0} \in \mathbf{R}^3$ 。

#### 4.2.2 角位移拟合

在进行姿态轨迹规划时，需要知道末端姿态随节点的改变量，故需要对曲线进行拟合得到关于节点  $u$  的姿态角位移。然而，与空间 B 样条曲线不同的是，空间 B 样条曲线拟合后得到的是末端平移的路程，对四元数样条曲线进行拟合得到的却不是末端旋转的姿态角位移。根据四元数与旋转之间的关系，单位四元数  $q = (\cos(\theta), \mathbf{v} \sin(\theta))$  (其中  $\mathbf{v}$  表示单位向量) 表示刚体绕转轴  $\mathbf{v}$  旋转了  $2\theta$ 。角速度与四元数及其导数之间的关系式为式(4.2)。又因为  $d\theta = \omega du$ ，故对其进行积分即可得到角位移，可表示为式(4.5)。

$$\theta = \int \omega du \quad (4.5)$$

式(4.5)不存在解析解，故将式(4.5)转化为数值的方法可得表达式为(4.6)。

$$\theta = \sum \omega(u) du \quad (4.6)$$

为了保证控制的准确性，需要确定拟合精度。可采用两次细分拟合做差作为拟合精度。当拟合精度小于阈值时，则认为该结果即为角位移。

上述方法虽然简单，但是计算量较大。为了提高曲线插补的准确度及速度，本文选用精度更高的复合柯特斯公式进行数值积分。因为复合柯特斯公式具有

$o(h^6)$  阶的精度，所以计算时误差收敛速度更快，有利于减少算法迭代时间并提高系统实时性。对于定义在区间  $[a, b]$  上的曲线  $C(u)$ ，设其导数曲线由  $f(u)$  表示，则  $n$  阶复合柯特斯公式的表达式为式(4.7)。

$$C_n = \frac{b-a}{90n} [7f(a) + \sum_{k=0}^{n-1} [32f(x_{k+1/4}) + 12f(x_{k+2/4}) + 32f(x_{k+3/4})] + 14 \sum_{k=1}^{n-1} f(x_k) + 7f(b)] \quad (4.7)$$

式中  $x_k = a + \frac{k}{n}$ ,  $x_{k+\frac{1}{4}} = x_k + \frac{1}{4n}$ ,  $x_{k+\frac{2}{4}} = x_k + \frac{2}{4n}$ ,  $x_{k+\frac{3}{4}} = x_k + \frac{3}{4n}$  分别表示区间  $[a, b]$

进行  $4n$  等分后的细分区间端点,  $k=0, 1, 2, \dots, n$ , 其中  $n$  为非零偶数。

为确保复合柯特斯积分的准确性, 在求解 B 样条曲线长度算法的每步迭代中, 需要分别计算  $2n$  阶和  $n$  阶的复合柯特斯积分值  $C_{2n}$  和  $C_n$ , 然后计算两者之间的差值, 看是否在限制误差  $\varepsilon$  内, 即需要判断式(4.8)不等式是否成立。

$$|C_{2n} - C_n| < \varepsilon \quad (4.8)$$

### 4.3 四元数向姿态空间转换

在位置轨迹规划中, 生成插补曲线后直接对其进行积分即可得到位移, 然后对所得位移进行速度规划后得到标量的位移、速度、加速度与时间的关系, 速度与对应位移处曲线上导数对应的单位向量相乘即可将速度矢量化, 加速度相类似。但本文所采用的姿态轨迹规划方法与位置轨迹规划方法在拟合及矢量化方面有两处不同: 首先, 直接对四元数插补曲线进行求积分得到的并不是姿态角位移; 其次, 不能应用位置规划中矢量化方法将速度规划得到的标量结果角速度及角加速度进行矢量化, 其过程需用图 4-1 进行表示。在 4.2 中对四元数导数与对应的角速度及角加速度进行了研究, 并给出了四元数曲线导数与角速度及角加速度之间的关系。对式(4.2)进行积分即可得到姿态角的角位移。得到的位移结果为标量, 同样该结果经 S 型速度规划得到的角位移、角速度及角加速度均为关于时间的标量函数。而最终得到的需要为矢量化的角速度及角加速度才能够与位置矢量化速度及加速度通过运动学反解得到各个关节的速度及加速度。为了能够将 S 型速度规划所得标量结果转化为矢量的姿态角速度及角加速度, 本节主要解决两个问题: 曲线的反向拟合问题及四元数转换为角速度及角加速度问题。

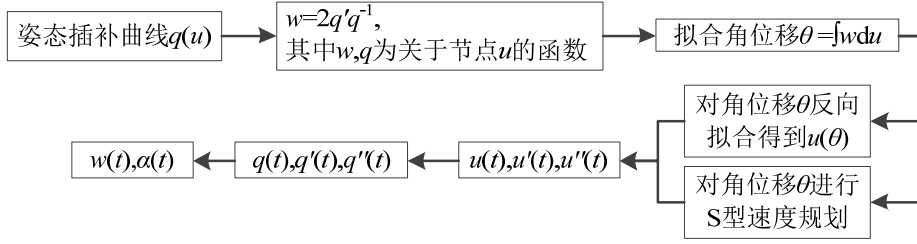


图 4-1 四元数向姿态空间转换

Figure 4-1 Transform quaternion to orientation space

### 4.3.1 曲线反向插值

在速度规划时，能够求出任意插补时刻  $t$  所对应的角位移长度，但需要将该位移值转换为四元数样条曲线的节点才能代入样条方程计算插补点四元数，即需要求出节点  $u$  关于时间  $t$  的函数。通过式(4.6)得到的结果为  $\theta(u)$ ，即角位移  $\theta$  关于节点矢量  $u$  的函数关系；通过式 S 型速度规划得到的结果为  $\theta(t)$ ，即角位移  $\theta$  关于节点矢量  $t$  的函数关系。要得到节点  $u$  关于时间  $t$  的函数只需对  $\theta(u)$  进行反求，即得到  $u(\theta)$ 。

因为在对姿态角位移进行拟合时是分区段分段进行的，因此，在对其进行反向拟合时也进行分段插值，即拟合出反函数  $u(\theta)$  使得给  $\theta \in [\theta_i, \theta_{i+1}]$  求出对应的  $u \in [u_i, u_{i+1}]$ ，其中  $\theta_i$  为节点区间  $[0, u_i]$  上对应的总角位移，如图 4-2 所示。因为在区间两端的值及其导数已知，故可采用三次厄米特插值<sup>[59]</sup>。第  $i$  段曲线  $u_i(\theta)$  可由式(4.9)得到，其中  $\theta \in [\theta_i, \theta_{i+1}]$ 。在式(4.9)中共有四个未知数，共需要四个初始条件，其中两个可由首末点函数值得到，即式(4.10)。

$$u_i(\theta) = A + B(\theta - \theta_i) + C(\theta - \theta_i)^2 + D(\theta - \theta_i)^3 (\theta - \theta_{i+1}) \quad (4.9)$$

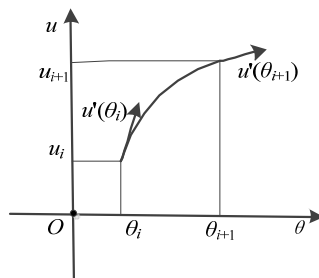


图 4-2 曲线反向拟合

Figure 4-2 Reverse curve fitting

$$\begin{cases} u_i(\theta_i) = u_i \\ u_i(\theta_{i+1}) = u_{i+1} \end{cases} \quad (4.10)$$

其他两个条件可由首末点的导数得到。结合式(4.2)及式(4.5)，对于起始点，对应导数为式(4.11)。

$$u_i'(\theta_i) = \frac{1}{\theta'(u_i)} = \frac{1}{2\|q'(u_i)q^{-1}(u_i)\|} \quad (4.11)$$

同理，对于末端来说，对应导数为式。

$$u_i'(\theta_{i+1}) = \frac{1}{\theta'(u_{i+1})} = \frac{1}{2\|q'(u_{i+1})q^{-1}(u_{i+1})\|} \quad (4.12)$$

综合式(4.10)~(4.12)可求得式(4.9)的系数如式(4.13)所示。

$$\begin{cases} A = u_i \\ B = u_i'(\theta_i) \\ C = \frac{u_{i+1} - u_i - B(\theta_{i+1} - \theta_i)}{(\theta_{i+1} - \theta_i)^2} \\ D = \frac{u_i'(\theta_{i+1}) - B}{(\theta_{i+1} - \theta_i)^2} - 2\frac{C}{\theta_{i+1} - \theta_i} \end{cases} \quad (4.13)$$

通过上述方法得到节点  $u$  关于时间的序列后，应用中点有限差分法得到节点  $u$  关于时间一二阶导数，如式(4.14)所示。

$$\begin{aligned} u'(t_i) &= \frac{u(t_{i+1}) - u(t_{i-1}))}{t_{i+1} - t_{i-1}} \\ u''(t_i) &= 4\frac{u(t_{i+1}) - 2u(t_i) + u(t_{i-1}))}{(t_{i+1} - t_{i-1})^2} \end{aligned} \quad (4.14)$$

#### 4.3.2 四元数向角速度及角加速度转换

虽然在 4.2.1 中介绍了四元数与角速度及角加速度之间关系，但为关于节点  $u$  的角速度和角加速度。本节将基于 4.3.1 对求解机器人末端关于时间的角速度及角加速度进行研究。

由文献[16]知，角速度与四元数的导数可用矩阵表示为式(4.15)所示。

$$\dot{q} = \Omega_q w \quad (4.15)$$

其中：  $\dot{q} = [\dot{q}_1 \quad \dot{q}_2 \quad \dot{q}_3 \quad \dot{q}_4]^T$ ，

$$\Omega_q = \frac{1}{2} \begin{bmatrix} -q_2 & -q_3 & -q_4 \\ q_1 & q_4 & -q_3 \\ -q_4 & q_1 & q_2 \\ q_3 & -q_2 & q_1 \end{bmatrix}。$$

取式(4.15)的逆，可得(4.16)式。

$$w = \Omega_q^+ \dot{q} \quad (4.16)$$

$$\text{其中: } \Omega_q^+ = 4\Omega_q^T = 2 \begin{bmatrix} -q_2 & q_1 & -q_4 & q_3 \\ -q_3 & q_4 & q_1 & -q_2 \\ -q_4 & -q_3 & q_2 & q_1 \end{bmatrix}。$$

本文中四元数曲线为关于节点  $u$  的函数。要得到关于时间  $t$  的角速度，则需要对四元数求关于时间  $t$  的导数，即式(4.17)所示。

$$w(t) = \Omega_q^+ \frac{d}{du} q(u) \cdot \frac{d}{dt} u(t) \quad (4.17)$$

其中:  $q=q(u(t))$ 。

而由式(4.4)可得角加速度公式，即式(4.18)。

$$\begin{bmatrix} 0 \\ \alpha(t) \end{bmatrix} = 2 \left( \frac{d^2}{du^2} q(u) \right) q^{-1}(u) \left( \frac{d}{dt} u(t) \right)^2 + 2 \left( \frac{d}{du} q(u) \right) q^{-1}(u) \frac{d^2}{dt^2} u(t) + 2 \begin{bmatrix} \|q'(u)\|^2 \\ \mathbf{0} \end{bmatrix} \left( \frac{d}{dt} u(t) \right)^2 \quad (4.18)$$

将式(4.18)简化并转化为矩阵方程，可得式(4.19)。

$$\alpha(t) = \Omega_q^+ \left( \left( \frac{d^2}{du^2} q(u) \right) \left( \frac{d}{dt} u(t) \right)^2 + \left( \frac{d}{du} q(u) \right) \frac{d^2}{dt^2} u(t) \right) \quad (4.19)$$

#### 4.4 S 型速度规划

本部分将对 S 型速度规划进行研究。本速度规划要做到给定任意的首末速度和位移、加速度和加加速度以及运行速度（或者运行时间）都能正常规划出 S 型的速度曲线。总体设计思想是把规划分解为两个层面进行处理，即速度规划层和位移（时间）规划层。速度规划层作为最底层的功能，根据给定的首末速度进行变速规划，得到变速的轨迹数据；位移（或时间）规划层通过解析法或者数值法得到满足给定位移或运动时间的整体轨迹数据。



#### 4.4.1 基本原理

##### (1) 速度层

如图 4-3 所示是变速运动时加速度曲线的几种情况，速度增加的变速对应图中上面虚框部分，速度降低的变速对应图中下面虚框部分。其中对速度降低的变速取反就变换为速度增加的变速问题。因此，变速实际上只有两种情况，即存在匀加速阶段和不存在匀加速阶段。

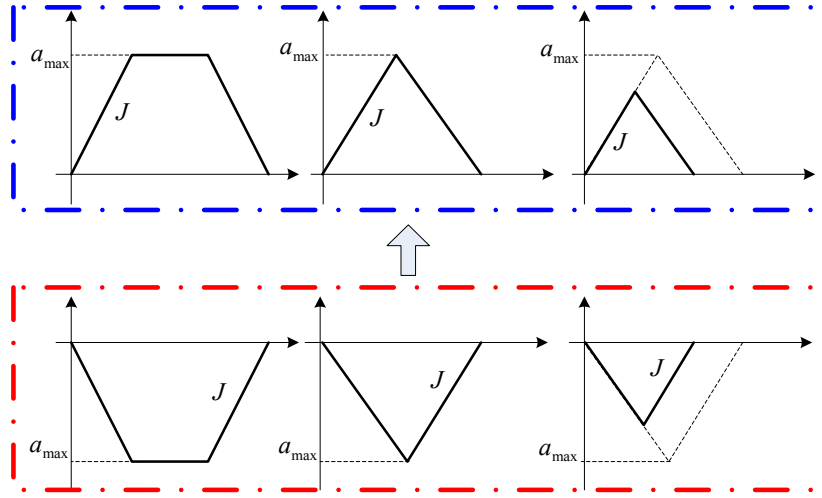


图 4-3 变速情况分析

Figure 4-3 Variable speed situation analysis

给定最大加速度  $a_{\max}$  和加加速度  $J$ ，令  $v_1$  和  $v_2$  分别为变速前后的速度，变速总时间为  $t_r$ ，匀变速时间为  $t_{hr}$ ，变加速度时间为  $t_{thr}=t_r-t_{hr}$ ，那么加速度、速度和位移表达式如(4.20)~(4.22)所示。

$$a(t) = \begin{cases} \operatorname{sgn}(v_2 - v_1)Jt, & t \leq (t_r - t_{hr})/2 \\ \operatorname{sgn}(v_2 - v_1)J(t_r - t_{hr})/2, & (t_r - t_{hr})/2 < t < (t_r + t_{hr})/2 \\ \operatorname{sgn}(v_2 - v_1)J(t_r - t), & (t_r + t_{hr})/2 < t \leq t_r \end{cases} \quad (4.20)$$

$$v(t) = \begin{cases} v_1 + \frac{1}{2} \operatorname{sgn}(v_2 - v_1)Jt^2, & t \leq (t_r - t_{hr})/2 \\ v_1 + \frac{1}{8} \operatorname{sgn}(v_2 - v_1)Jt_{thr}^2 + \frac{1}{2} \operatorname{sgn}(v_2 - v_1)Jt_{thr}(t - t_{thr}/2), & (t_r - t_{hr})/2 < t < (t_r + t_{hr})/2 \\ v_1 + \frac{1}{4} \operatorname{sgn}(v_2 - v_1)Jt_{thr}^2 + \frac{1}{2} \operatorname{sgn}(v_2 - v_1)Jt_{thr}t_{hr} - \frac{1}{2} \operatorname{sgn}(v_2 - v_1)J(t_r - t)^2, & (t_r + t_{hr})/2 < t \leq t_r \end{cases} \quad (4.21)$$

$$L(t) = \begin{cases} v_1 t + 1/6 \operatorname{sgn}(v_2 - v_1) J t^3, & t \leq (t_r - t_{hr})/2 \\ v_1 t + 1/48 \operatorname{sgn}(v_2 - v_1) J t_{thr}^3 + \frac{1}{8} \operatorname{sgn}(v_2 - v_1) J t_{thr}^2 (t - t_{thr}/2) \\ \quad + \frac{1}{4} \operatorname{sgn}(v_2 - v_1) J t_{thr} (t - t_{thr}/2)^2, & (t_r - t_{hr})/2 < t < (t_r + t_{hr})/2 \\ v_1 t + \frac{1}{4} \operatorname{sgn}(v_2 - v_1) J \left( -\frac{1}{2} t_{thr}^3 - \frac{3}{2} t_{thr}^2 t_{hr} - t_{hr}^2 t_{thr} + t_{thr}^2 t + 2 t_{thr} t_{hr} t \right) \\ \quad + \frac{1}{6} \operatorname{sgn}(v_2 - v_1) J (t_r - t)^3, & (t_r + t_{hr})/2 < t \leq t_r \\ v_1 t_r + \frac{1}{4} \operatorname{sgn}(v_2 - v_1) J \left( \frac{1}{2} t_{thr}^3 + \frac{3}{2} t_{thr}^2 t_{hr} + t_{thr} t_{hr}^2 \right), & t > t_r \end{cases} \quad (4.22)$$

## (2) 位移层

位移层的情况主要分为三种：

(a) 满足所有给定参数。由初始速度变速到运行速度  $v$ ，然后运行一段时间（可能时间为 0），然后再由运行速度  $v$  变速到末速度。

(b) 调整运行速度。通过调整运行速度后，满足 (a)。

(c) 首末变速。只能实现（甚至要调整末速度）首速度到末速度的变速。

关键问题就是如何确定属于上述那种情况以及如何调整运行速度或者末速度，返回计算得到的“最合适”速度规划结果。

## (3) 时间层

根据位移  $L$  和给定运行时间  $T$  求平均速度  $v$ ，根据平均速度  $v$  将给定时间的规划分为三种情况：平均速度小于首末速度，平均速度大于首末速度，及平均速度介于首末速度之间。

关键问题就是如何根据平均速度和首末速度，确定每种情况的运行速度调整或末速度调整方法，返回计算得到的“最合适”速度规划结果。

## 4.4.2 核心算法设计

### (1) 速度层

如图 4-3 所示，当处于临界情况时，相应的临界速度计算如式(4.23)所示。只有当变速大于该临界值时，变速过程才存在匀加（减）速阶段。

$$\Delta v_{thr} = 2 \int_0^T a(t) dt = 2 \int_0^T J t dt = \frac{a_{\max}^2}{J} \quad (4.23)$$

当变速值大于临界值时，变速总时间、匀加（减）速时间及变加速(减)时间分

别为式(4.24)、(4.25)及(4.26)。

$$t_r = 2 \frac{a_{\max}}{J} + \frac{|v_2 - v_1| - |\Delta v_{thr}|}{a_{\max}} \quad (4.24)$$

$$t_{hr} = \frac{|v_2 - v_1| - |\Delta v_{thr}|}{a_{\max}} \quad (4.25)$$

$$t_{thr} = 2 \frac{a_{\max}}{J} \quad (4.26)$$

当变速值小于或等于该临界值是，变速时间为式(4.27)。

$$t_r = t_{thr} = 2 \sqrt{\frac{|v_2 - v_1|}{J}}, \quad t_{hr} = 0 \quad (4.27)$$

## (2) 位移层

位移层的临界情况为由首速度变速到运行速度，再由运行速度变速到末速度，如图 4-4 所示。图 4-4 中左图所示为运行速度比首末速度都高，中图所示为运行速度比首末速度都低，右图所示为运行速度介于首末速度之间。

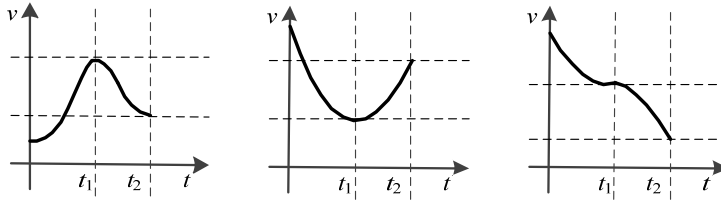


图 4-4 临界位移描述

Figure 4-4 Description of critical displacement

计算两段变速的位移之和  $L_{ref1}$ ，如果  $L_{ref1} < L$ （给定的运动位移），那么说明能达到用户设定的速度和首末速度，系统的运行情况为变速、匀速再变速，如图 4-5 所示。

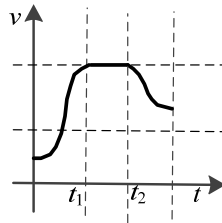


图 4-5 变速-匀速-变速的运动过程

Figure 4-5 movement process of variable - uniform - variable speed

否则，先忽略用户设定的运行速度，直接根据首末速度，系统进行变速，计算此变速过程的路程  $L_{ref2}$ 。如果  $L_{ref2} > L$ ，则说明给定路程太短，以至于无法完成初

始速度到末端速度的变速过程（当然也有可能用户设定了非常小的加加速度或加速度，这也说明给定路程相对过短），则采用二分查找的方法确定与给定末端速度最接近的实际末端速度。如图 4-6 所示为变加速过程，变减速过程类似，编程实现方法基本一样。

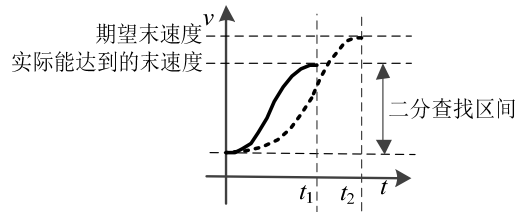


图 4-6 通过二分查找确定实际的末速度

Figure 4-6 Determine the final velocity by the binary search

否则，说明可以通过调整运行速度实现变速-匀速-变速的运动，可以分为以下三种情况进行处理。

(1) 用户设定的运行速度高于首末速度，则以  $v-\max(v_s, v_e)$  为二分区间，进行二分查找，确定实际的运行速度，如图 4-7 所示。

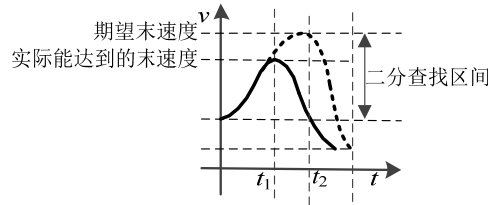


图 4-7 查找所能达到的最高运行速度

Figure 4-7 To find the highest running speed which can be achieved

(2) 如果用户设定的运行速度低于首末速度，则以  $v-\min(v_s, v_e)$  为二分区间，进行二分查找，确定实际的运行速度，如图 4-8 所示。

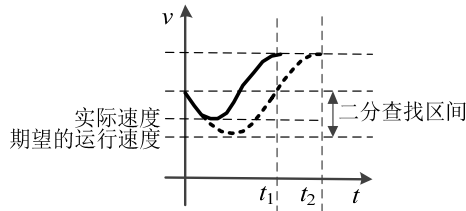


图 4-8 通过二分查找确定实际的运行速度

Figure 4-8 Determine the running velocity by the binary search

(3) 如果用户设定的运行速度介于首末速度之间，那么最大速度（可能是首

速度也可能是末速度) 匀速运行, 其它部分首末变速, 如图 4-9 所示。

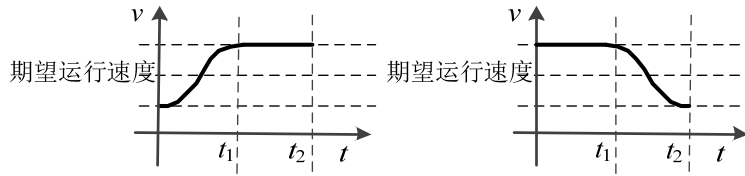


图 4-9 设定的运行速度介于首末速度之间

Figure 4-9 the situation of setted running speed between initial and finnal velocity

### (3) 时间层

用户给定时间而不是运行速度的情况下, 本规划定义为时间层的规划, 时间层的规划 (或者给定时间的规划) 在进行同步处理时是必须功能, 在其它运动时间要求严格的场合下必须采用。

首先根据给定的位移  $L$  和运动时间  $T$  计算平均速度  $v$ , 如果位移  $L$  为负, 那么要进行整体取反处理 (与位移层处理方法相同), 可以分为三种情况进行处理。

#### (a) 平均速度 $v$ 比首末速度都小

首先计算首速度到平均速度的变速, 并得到变速位移  $L_{ref1}$ 。如果该变速位移  $L_{ref1} \geq L$ , 那么说明首速度太大且运动位移过小, 还未来得及减速到平均速度, 在小于给定的运动时间内就已经完成了位移。这种情况下只能直接采用位移层的速度规划, 得到一个可行解, 并返回标志提醒用户, 结果是否采用由用户决定, 情况如图 4-10 中的左图所示。

其次计算平均速度到末速度的变速, 并得到变速位移  $L_{ref2}$ , 如果两段变速位移之和  $(L_{ref1} + L_{ref2}) \geq L$ , 那么说明首速度或末速度太大且运动位移过小, 这种情况下, 确定如图 4-10 中的中图所示的搜索区间, 降低末速度, 继续循环。如果  $(L_{ref1} + L_{ref2}) < L$ , 那么按照如图 4-10 中的右图所示的搜索区间进行搜索处理。

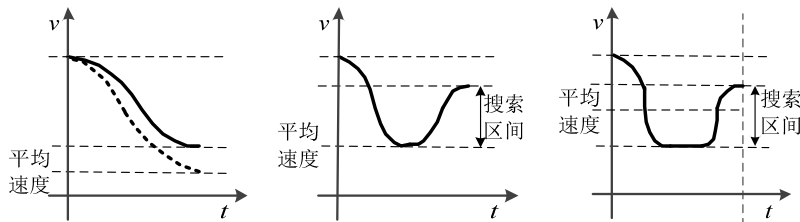


图 4-10 平均速度最小情况下的时间层规划分析

Figure 4-10 Time layer planning analysis when the average velocity is minimum

(b) 平均速度  $v$  比首末速度都大

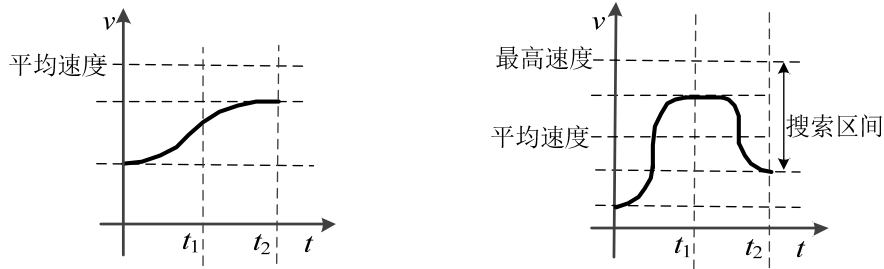


图 4-11 平均速度最大情况下的时间层规划分析

Figure 4-11 Time layer planning analysis when the average velocity is maximum

首先计算首速度  $v_s$  到平均速度  $avgvel$  的变速和平均速度  $avgvel$  到末速度  $v_e$  的变速，并得到变速位移  $L_{ref1}$  和  $L_{ref2}$ 。如果两段变速位移之和  $(L_{ref1} + L_{ref2}) \geq L$ ，那么说明首速度或末速度太小且运动位移过小。这种情况下，为了满足末速度，采用位移层的规划，并返回标志 1 提醒用户，如图 4-11 中的左图所示。如果  $(L_{ref1} + L_{ref2}) < L$ ，按照如图 4-11 中的右图所示的搜索区间进行搜索处理。

(c) 平均速度  $v$  介于首末速度之间

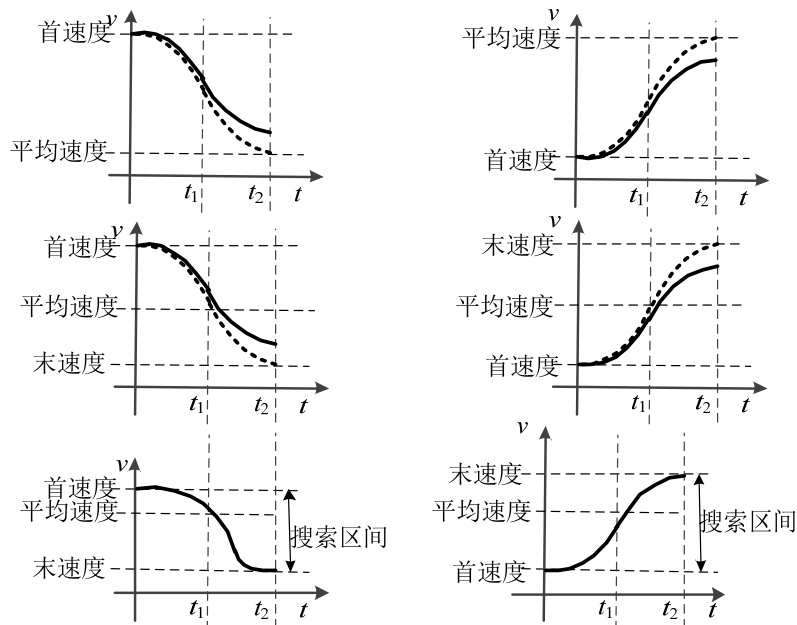


图 4-12 平均速度处于首末速度之间的时间层规划分析

Figure 4-12 Time layer planning analysis when the average velocity is between initial and final velocity

首先计算首速度到平均速度的变速, 得到位移  $L_{ref1}$ , 如果  $L_{ref1} \geq L$ , 说明无法实现给定时间的运动规划, 直接采用位移层规划, 并返回规划结果, 通知用户, 由用户决定如何使用该规划结果, 如图 4-12 中的上图所示。

否则, 进行首末变速, 得到变速位移  $L_{ref2}$ , 如果  $L_{ref2} \geq L$  且末速度比首速度大, 在图 4-12 中的中图所示搜索区间内降低末速度, 然后再进行首末变速得到新的变速位移  $L_{ref2}$ , 直到  $L_{ref2} < L$  且运动时间满足要求时, 调用运行速度查找功能, 如果运行速度查找成功则退出循环。否则直接采用位移层规划, 并返回规划结果 1。

## 4.5 本章小结

本章主要基于第 3 章姿态轨迹插补曲线进行了姿态轨迹规划相关技术的研究, 主要包括四元数样条曲线导数与姿态角速度及角加速度之间的关系、角位移拟合、标量角速度及角加速度矢量化、S 型速度规划等三部分。在第 3 章的基础上, 首先给出了四元数函数导数与角速度之间的关系, 并给出了用数值积分的得到姿态角位移的方法, 给速度规划打下了基础。在四元数向姿态空间转换中, 运用三阶厄米特插值与差分法得到了节点及其导数与时间的关系, 进而可将经 S 型速度规划得到的标量姿态角位移、角速度、角加速度矢量化, 以便通过运动学逆解得到相应的关节位置、速度及加速度。在 S 型速度规划中, 将速度规划分为速度层、位移层和时间层进行研究, 其中速度层为速度规划中的底层规划, 主要充当预规划的角色, 位移(时间)层则是对给定位移(时间)进行规划, 尽可能得到满足要求的规划结果。

## 5 算法仿真、验证及分析

本章的主要工作是对本文研究的基于四元数曲线的姿态轨迹规划相关算法的有效性 & 可行性进行验证。MATLAB 是一款优秀的数值仿真软件，已广泛应用于工程计算、通信与信息处理、控制系统仿真、图像处理、金融分析等领域<sup>[60]</sup>。本章将以 MATLAB 为仿真平台对本文所研究的算法进行验证，主要包括机器人运动学正逆解算法、基于四元数曲线的姿态插补曲线及过渡曲线生成算法、姿态轨迹规划算法等。

### 5.1 机器人运动学仿真

机器人运动学是本文所研究算法验证的基础，其正确性直接影响到算法的验证。本部分首先将在不涉及本文所研究姿态轨迹规划方法的情况下对本文所建立机器人模型的运动学正逆解(本文中主要用到逆解)及 S 型速度规划等相关算法进行测试。在位置上，将采用直线作为机器人的轨迹。直线虽然方程简单，但要使其精确实现却较为困难。因为要使机器人尽可能精准地沿直线运动时需要关节不停地正反转，同时对运动学也是很大的考验。关于姿态，本部分将采用绕固定轴旋转的方法进行两点姿态规划。

本次仿真给定的示教点如表 5-1 所示。根据首末位置可以直接求出运动中平移的位移为 567.8908mm；根据首末示教点姿态欧拉角可以计算出相应的旋转矩阵，进而得到从起始示教点到末尾示教点姿态绕固定轴旋转角度，其中固定转轴方向向量为(0.2474,0.0804,-0.9656)，所需要旋转角度为 1.2901rad；接着分别对平移和旋转进行 S 型速度规划。本次仿真为两点间规划，因此设定首末示教点处平移和旋转的速度均为 0，设定平移的运行速度为 1m/s，最大加速度为 1m/s<sup>2</sup>，最大加加速度为 2m/s<sup>3</sup>；设定旋转的运行角速度为 1rad/s，最大角加速度为 1rad/s<sup>2</sup>，最大角加加速度为 2rad/s<sup>3</sup>。S 型速度规划的结果如图 5-2 所示。将 S 型速度规划结果转化为矢量的速度及加速度，并通过运动学逆解将位姿、速度及加速度转化为对应的关节角度、角速度及角加速度，结果如图 5-3 所示。从图 5-3 可以看出，关节角度、角速度及角加速度都是连续的。将机器人简化为连杆机构，可通过如图 5-1 观察到



其运动状态，位形在可接受范围内。由此可验证本文运动学相关算法是正确的。

表 5-1 运动学仿真示教点信息

Table 5-1 teach points for kinematic simulation

	起点	终点
位置 (mm)	(400,0,400)	(0,400,450)
姿态 (rad)	$(-0.2\pi, 0.2\pi, 0)$	$(-0.6\pi, 0.3\pi, 0)$

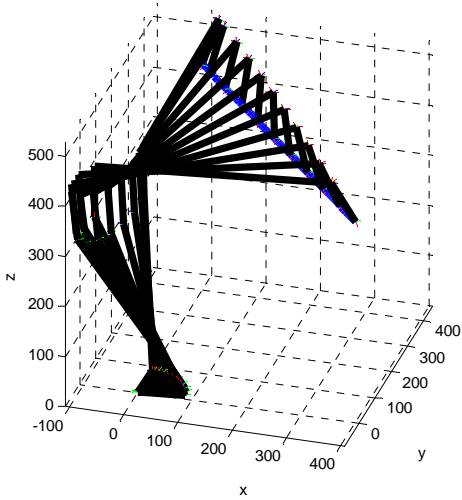


图 5-1 机器人运动情况

Figure 5-1 the movement of manipulator

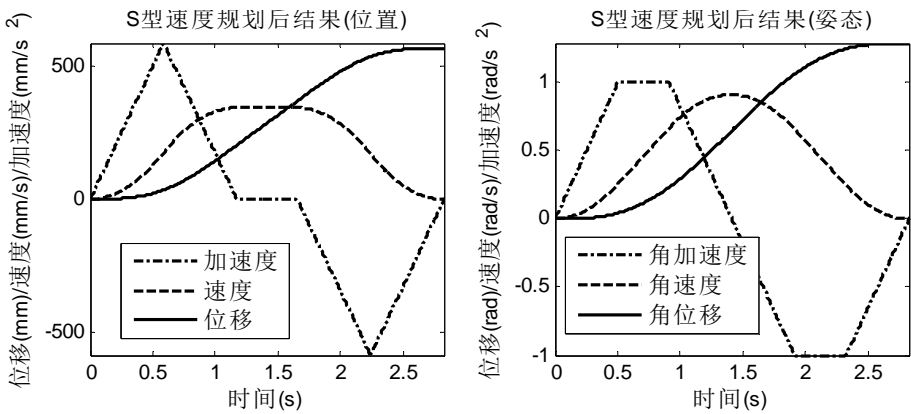


图 5-2 S 型速度规划结果

Figure 5-2 the result of S-type velocity planning

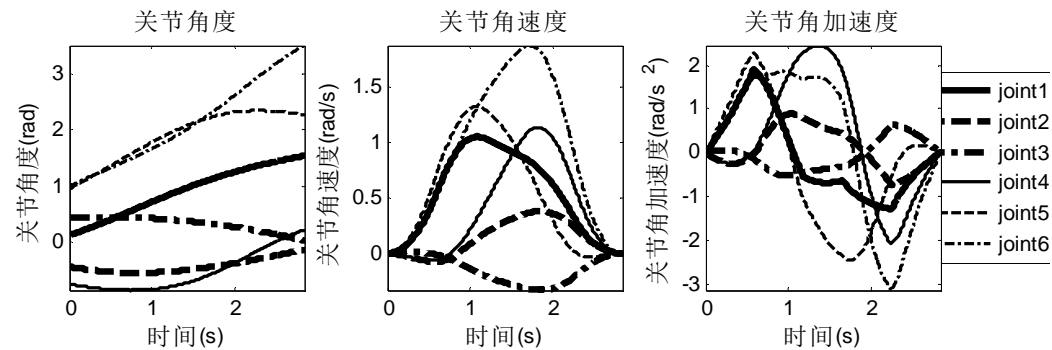


图 5-3 关节随时间的变化情况

Figure 5-3 the movement of joints about time

## 5.2 基于四元数球面线性插值的姿态轨迹规划

四元数球面线性插补类似于位置插补中的直线轨迹，插补曲线构造较为简单。本部分将利用球面线性插补曲线作为姿态插补曲线以验证四元数曲线拟合、反向拟合、节点导数求取及规划速度矢量化等算法。为了减少其他干扰，本部分及后续小节中验证姿态轨迹规划相关算法的部分将固定位置，而只改变其姿态。固定位置坐标为(400, 0, 400)(单位：mm)。

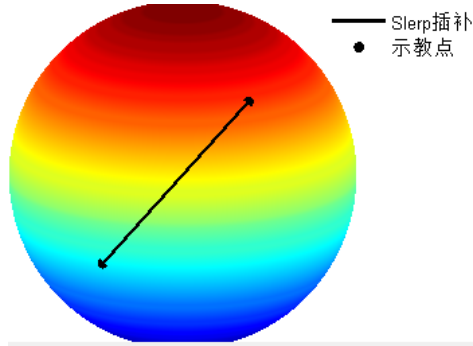


图 5-4 球面线性插补

Figure 5-4 Spherical linear interpolation

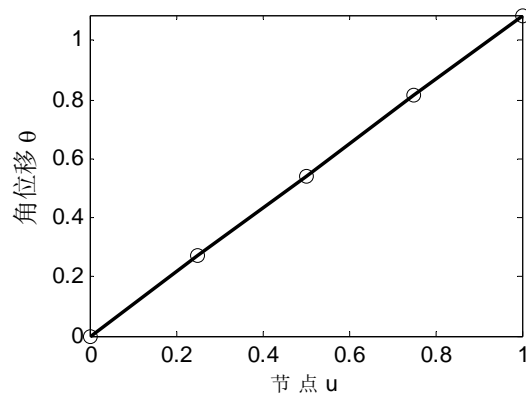


图 5-5 角位移于节点  $u$  的关系

Figure 5-5 the relationship between angle translation and knot  $u$

球面插补需要两个示教点，即起始点和终止点姿态。在此，随机产生两个单位四元数作为该部分仿真验证示教点的姿态，即起点姿态四元数为(0.6228,-0.6070,0.0148,0.4934)，终点姿态四元数为(0.5158,-0.4209,0.7044,-0.2462)。从起点到终点对姿态进行球面线性插补如图 5-4 所示，对角位移进行拟合得到角位移与节点  $u$  之间的函数关系图如图 5-5 所示，可知球面线性插补下角位移与节点  $u$  呈线性关系。将首尾速度设为 0，运行角速度、最大角加速度、最大角加加速度同 5.1，所得到的 S 型速度规划结果如图 5-6 中左图所示。根据角位移与节点  $u$  的关系及 S 型速度规划得到的角速度可以用 4.3.1 中数值方法得到节点  $u$  及其一二阶导数与时间的关系如图 5-7 所示，进而可以将 S 型速度规划得到的角速度、角加速度矢量化，其结果如图 5-6 中右边两幅图所示。通过运动学逆解得到相应关节角度、角速度及角加速度，如图 5-8 所示。图 5-8 表明，做得到的关节角及其速度、加速

度连续，并且在可接受的范围内，达到了预期的目的。此结果表明，本文所研究的针对四元数曲线的姿态位移拟合及角速度矢量化等算法是合理的。

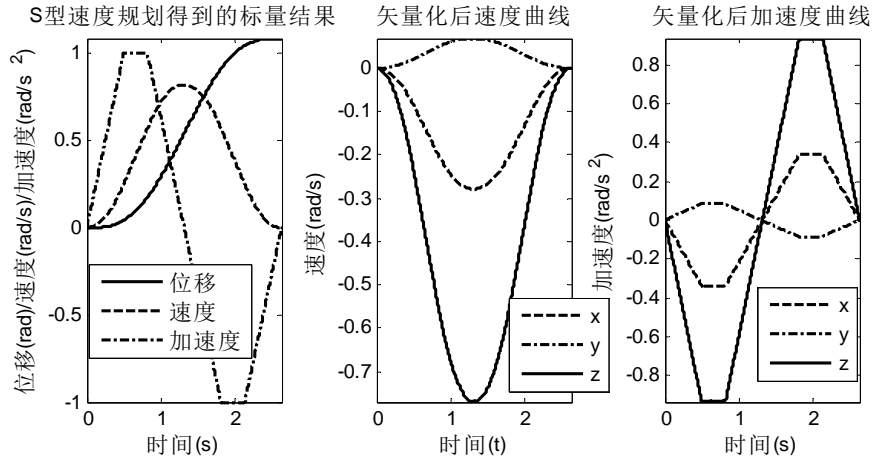


图 5-6 S 型速度规划及矢量化结果

Figure 5-6 the result of S-type velocity planning and vectorization

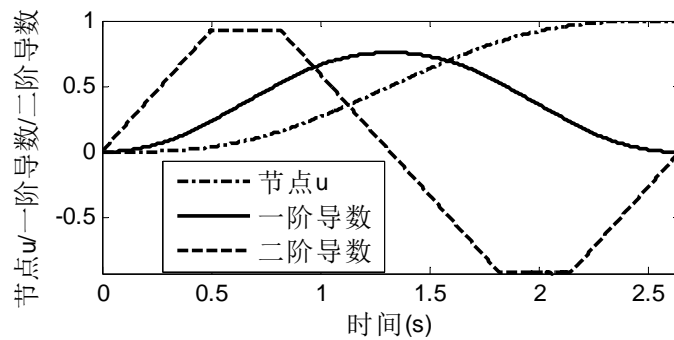


图 5-7 节点及其导数与时间的关系

Figure 5-7 the change of knot and its derivative about time

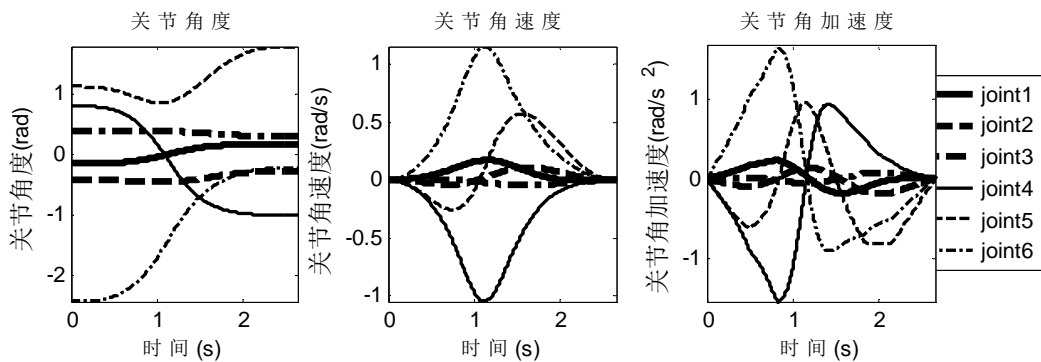


图 5-8 关节角随时间的变化情况

Figure 5-8 the movement of joints angulars about time

### 5.3 基于四元数样条曲线的姿态轨迹规划

本部分将基于经过以上两小节验证过算法的基础上，对本文所研究的基于四元数样条曲线的姿态轨迹规划相关算法进行验证。

首先给定四个姿态示教点，用单位四元数表示依次为 $(0.4711, 0.7322, 0.4668, -0.1551)$ ,  $(0.8133, 0.2562, 0.5164, -0.0794)$ ,  $(0.3947, -0.3320, 0.8521, 0.0898)$ ,  $(0.5000, -0.7871, 0.3588, 0.0418)$ 。利用 3.3 中的相关算法生成四元数样条曲线，将其后三个元素单位化后表示在三维单位球面上如图 5-9 所示，其中实线表示四元数样条曲线；点划线为样条曲线的控制曲线；“\*”为控制点，由相邻控制点经球面线性插补生成；“•”表示教点。求取样条曲线一二阶导数并得到其曲率，曲率与节点  $u$  之间的关系如图 5-10 所示。由图 5-10 可知，曲线首末位置的曲率为 1，表明曲线首尾与控制曲线相切，这也使得四元数样条曲线可以作为曲率始终为 1 的球面线性插补的过渡曲线。

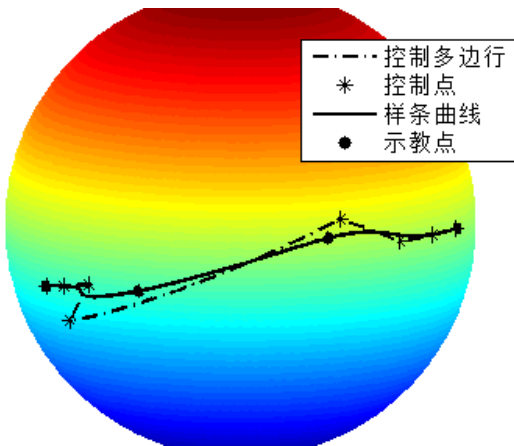


图 5-9 四元数样条曲线

Figure 5-9 the quaternion spline curve

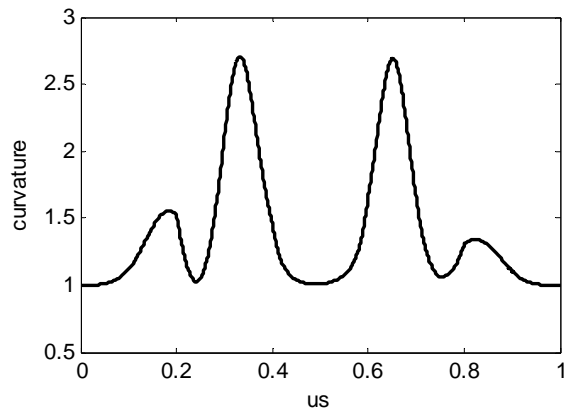


图 5-10 四元数样条曲线曲率

Figure 5-10 the curvature of spline curve

根据 4.2 中相关算法对曲线对应的姿态角位移进行拟合，得到节点  $u$  与角位移之间的关系如图 5-11 中左图所示。限制姿态变化过程中的运行速度为  $0.5\text{rad/s}$ ，最大加速度为  $0.5\text{rad/s}^2$ ，最大加加速度为  $1\text{rad/s}^3$ ，并给定四个示教点处角速度依次为  $(0, 0.0635, 0.4567, 0)$  (单位:  $\text{rad/s}$ )，进行 S 型速度规划得到结果如图 5-11 右图所示，进而反求出节点  $u$  与时间之间的关系，利用差分法得到节点  $u$  一阶导数及二阶导数与时间  $t$  的关系。节点  $u$  及其一二阶导数与时间之间的关系可表示为如图

5-12 所示。利用四元数样条曲线、节点  $u$  及其一二阶导数与时间的关系可用将 S 型速度规划的标量结果矢量化，矢量化后的姿态角速度及加速度在  $xyz$  轴上的分量可表示为如图 5-13 所示。利用机器人运动学逆解可以得到机器人关节空间对应的关节角度、角速度及角加速度，如图 5-14 所示。

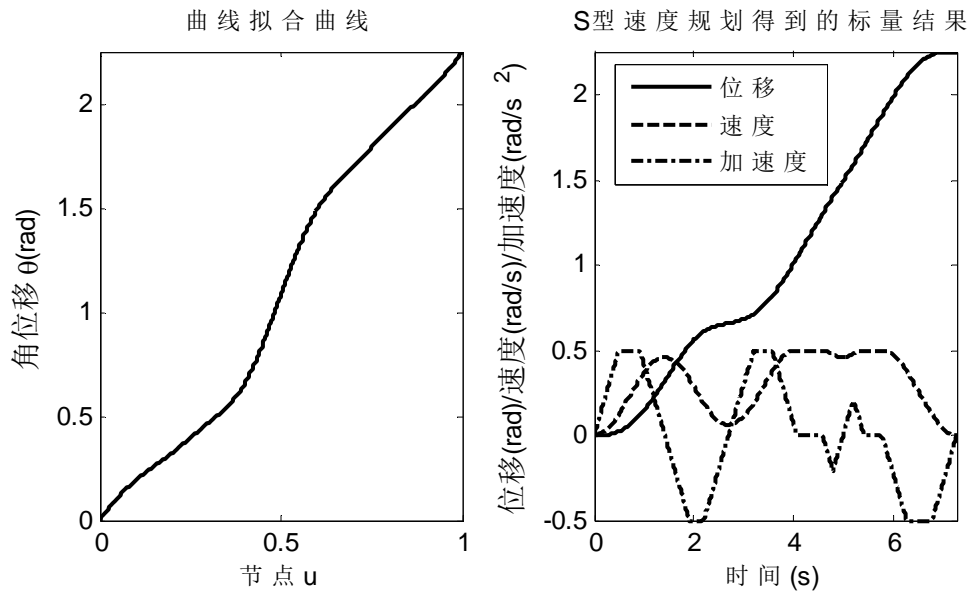


图 5-11 曲线拟合及 S 型速度规划

Figure 5-11 Curves fitting and S-type velocity planning

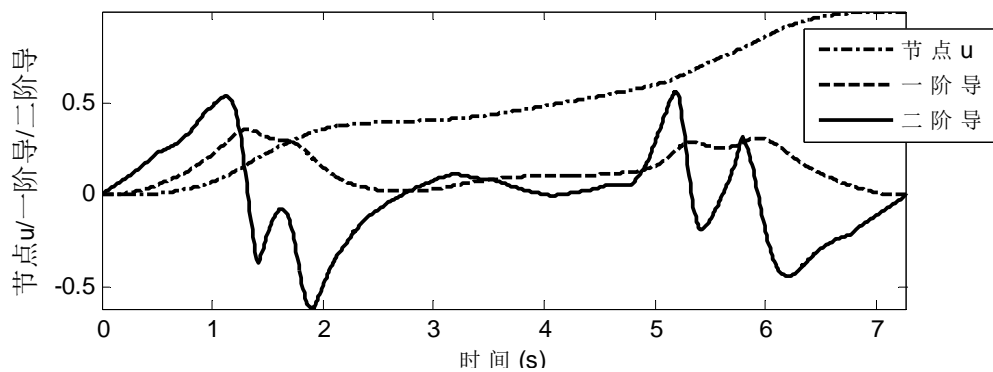


图 5-12 节点及其导数与时间的关系

Figure 5-12 the change of knot and its derivative about time

与四元数球面线性插补相比(图 5-7, 图 5-5), 样条曲线对应的节点及其一二阶导数与时间之间的关系(图 5-12)变化要复杂地多, 并且节点  $u$  与姿态角位移之间也不呈线性关系(图 5-11)。这是由样条曲线的特性造成的。球面线性插补属于单个区

间规划,  $u$  与四元数成线性分布(如式(3.34))。四元数样条曲线为多段规划, 由示教点求得控制点, 控制点间距不同, 但本文所用四元数样条曲线为均匀样条曲线, 这就使得相同节点区间内对应的曲线长度变化较大, 进而使得对应的姿态角位移相差较大。但即便如此, 矢量化后的姿态角速度及角加速度还是连续的, 逆解得到的关节角度、角速度及角加速度都是连续的, 且在安全范围内, 说明算法有效。

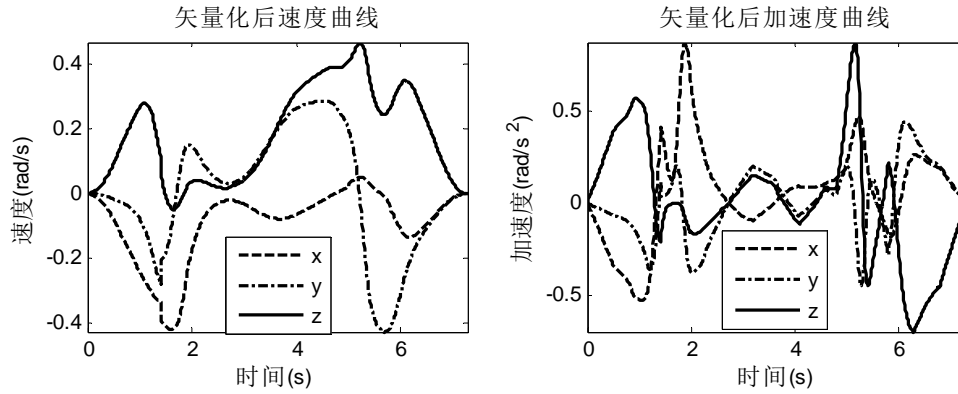


图 5-13 矢量化后速度及加速度在 xyz 轴上的分量

Figure 5-13 the components in xyz axis of velocity and acceleration after vectorization

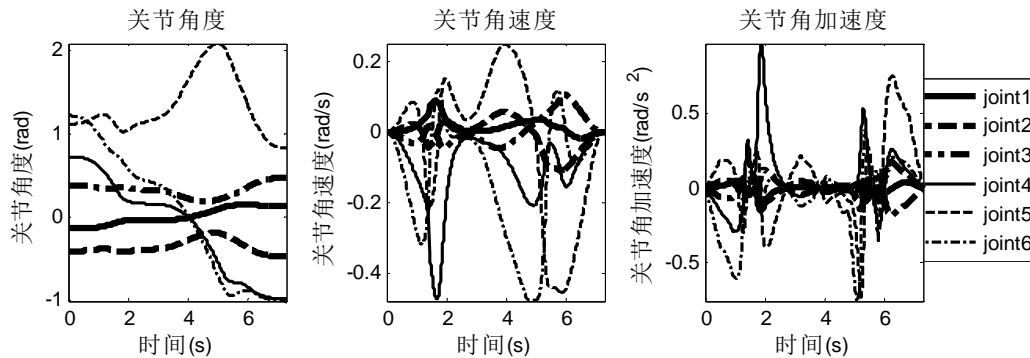


图 5-14 关节随时间的变化情况

Figure 5-14 the movement of joints about time

## 5.4 平滑过渡曲线的姿态轨迹规划

在进行验证时, 本部分采用两条球面线性插补作为需要平滑连接的插值曲线。球面线性插补较四元数样条曲线插补简单, 但其只能用于两点间插补, 多段规划时在两节点处速度要降低为 0, 否则会引起机械臂抖动等。用样条曲线作为四元数球面线性插补的过渡曲线可以将两者结合起来, 算法较为简便, 并且也可进行多段规划。本节将对 3.5 所提出的球面线性插补过渡曲线的姿态轨迹规划相关算法进

行仿真验证。

要进行球面线性插补过渡需要两条相接的球面线性插补曲线，即三个姿态示教点。三个姿态示教点用单位四元数依次表示为(0.4711,0.7322,0.4668,-0.1551),

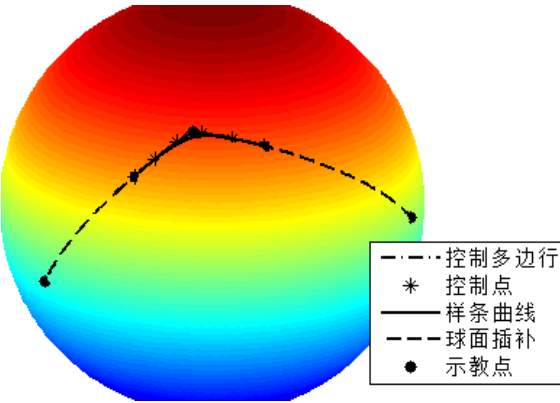


图 5-15 球面插补过渡  
Figure 5-15 transition curve between Slerps

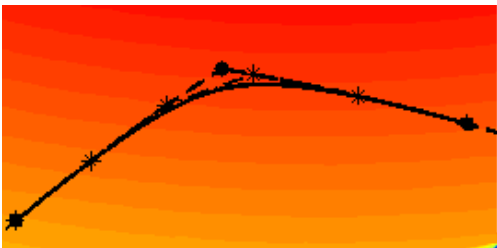


图 5-16 过渡区域  
Figure 5-16 transition region

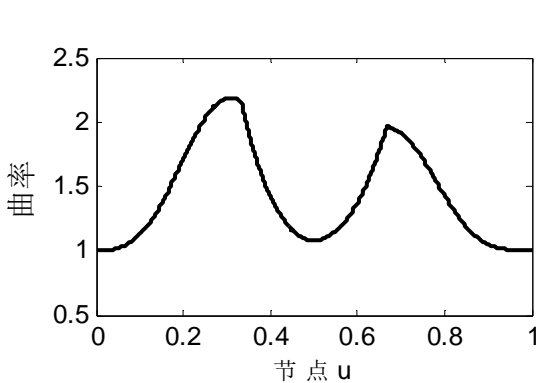


图 5-17 过渡区曲线曲率  
Figure 5-17 curvature in transition region

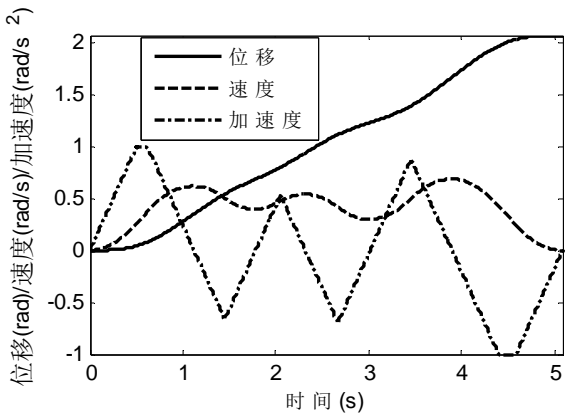


图 5-18 S 型速度规划结果  
Figure 5-18 S-type velocity planning

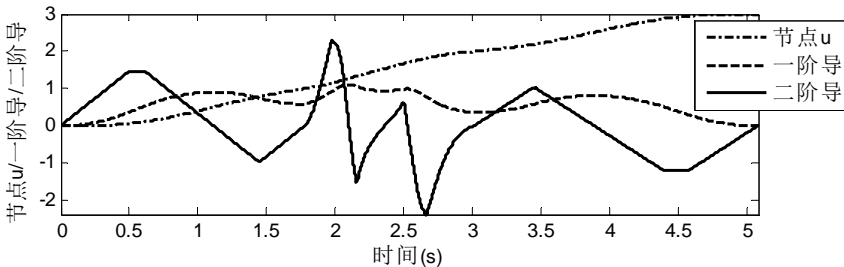


图 5-19 节点及其导数与时间的关系  
Figure 5-19 the change of knot and its derivative about time

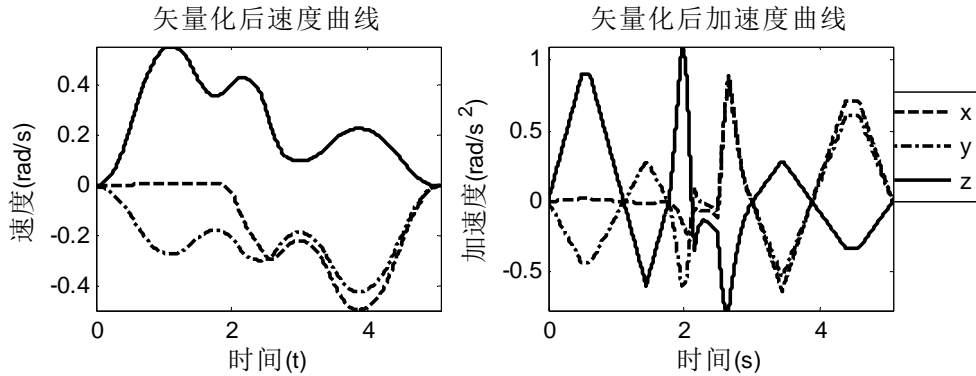


图 5-20 矢量化后速度及加速度在 xyz 轴上的分量

Figure 5-20 the components in xyz axis of velocity and acceleration after vectorization

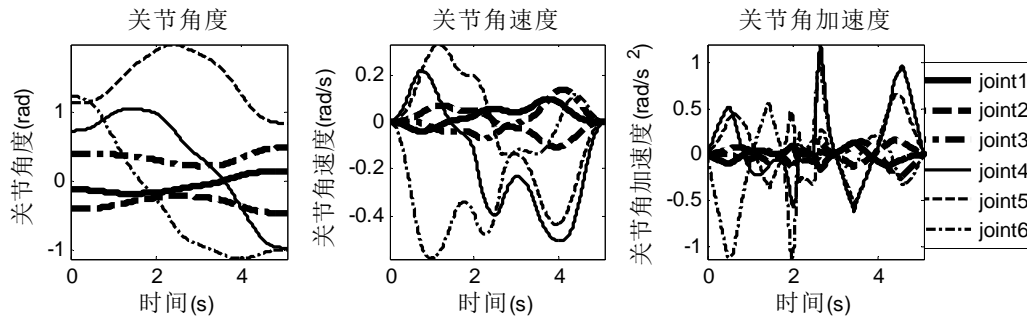


图 5-21 关节随时间的变化情况

Figure 5-21 the movement of joints about time

(0.5364, 0.1262, 0.6388, 0.5370), (0.5000, -0.7871, 0.3588, 0.0418)。将三个示教点依次两相邻间进行球面线性插补，并根据 3.5 中算法得到球面线性插补间过渡曲线。球面线性插补及其过渡曲线在三维单位球面上表示为如图 5-15 所示，其中过渡区域如图 5-16 所示。由过渡区域曲率曲线(图 5-17)可知，过渡曲线首末位置曲率为 1，与前后球面线性插补曲线平滑相接，这是由过渡曲线生成算法所保证的。因为过渡曲线不经过原中间示教点，故在进行速度规划时，将原第二个示教点删除，并增加过渡曲线与球面线性插补曲线相交的两个姿态点为示教点，从而得到新的示教点集。给定示教点处角速度为(0, 0.3, 0.4, 0)(单位: rad/s)对球面线性插补曲线及过渡曲线进行姿态角位移进行拟合，进行 S 型速度规划，得到图 5-18。根据反向拟合得到的节点  $u$  关于角位移的函数关系及角位移与时间的关系，得到节点及其一二阶导数与时间之间的关系如图 5-19 所示。矢量化后笛卡尔空间姿态角速度及加速度如图 5-20 所示。通过运动学逆解将姿态、角速度及加速度转化为关节角度、角



速度及角加速度，如图 5-21 所示。

与图 5-7 相比可知，在图 5-19 中可以看到，节点  $u$  及其一二阶导数明显分为三段，即球面线性插补——样条曲线插补——球面线性插补，并且节点  $u$  及其一二阶导数在过渡四元数样条曲线与前后球面线性插补连接处连续线性。这主要是由于球面线性插补和样条曲线插补与节点之间过渡点处的四元数及其一二阶导数相等。从图 5-20 和图 5-21 可知，规划结果是连续的。

## 5.5 本章小结

本章主要将本文研究的算法进行实现并在 MATLAB 上进行可行性及有效性验证。首先，以直线作为位置插补曲线、以绕固定轴旋转作为姿态插补方法对机器人进行了两点间位姿轨迹规划，验证了本文所建立机器人模型运动学(包括位姿、速度及加速度)正逆解的正确性；然后，以四元数球面线性插补为姿态插补曲线对机器人进行了固定位置的姿态轨迹规划仿真，验证了基于四元数的姿态角位移拟合、标量姿态角速度及角加速度矢量化等算法的正确性；接着，在以上仿真验证的基础上，进行了基于四元数样条曲线的姿态轨迹规划仿真，验证了以四元数样条曲线作为姿态插补曲线进行姿态轨迹规划的可行性；最后，对研究的基于四元数样条曲线的多条四元数插值曲线平滑过渡曲线算法进行了姿态轨迹规划验证。通过以上仿真，得到各个仿真数据及仿真图。通过对比分析，证明了本文所研究算法的正确性及有效性。

## 结 论

本文针对工业机器人传统姿态插值方法存在奇异、效率低等姿态轨迹规划难问题研究了以四元数样条曲线为姿态插补曲线的机器人姿态轨迹规划方法，最终得到了具有二阶连续性的插补曲线，并且规划出来的关节角位移、角速度以及角加速度都是连续的，并且通过了仿真验证。本文所在的主要工作如下：

(1) 建立了实验用机器人的运动学模型及相关算法。首先，基于 DH 建模方法对本文的实验平台 FDR-v6 进行几何建模，并得到了其运动学(位置)正解；然后，分别通过几何解法和数值解法得到机器人运动学(位置)结果，并将两者的优点进行结合得到了本文的机器人运动学(位置)逆解方法；最后，对雅克比矩阵的求解进行了介绍，并得到了本文实验用机器人的雅克比矩阵，及其速度与加速度的运动学正逆解相关算法。

(2) 研究了基于四元数曲线的姿态插补曲线及平滑过渡曲线生成相关算法。首先，结合本文的内容，对四元数的基本知识进行了介绍，主要有比较特殊的运算性质及四元数与旋转的关系，并得到了四元数与旋转矩阵之间的变换关系；然后，对四元数样条曲线进行了研究，给出了四元数样条曲线及其一二阶导数的形式、节点向量形式，尤其是通过改变控制点形式使得曲线始于第一个示教点并终于最末一个示教点，并且曲线首末与控制多边形相切；接着给出了四元数球面线性插补曲线生成及其一二阶导数的相关算法；最后，对基于四元数曲线姿态插补的平滑过渡曲线进行了研究，并给出了一般平滑过渡曲线生成的相关算法。

(3) 基于生成的四元数插补曲线对姿态轨迹规划进行了研究。首先，给出了四元数及其一二阶导数与姿态角速度、角加速度之间的关系，进而通过数值积分方法得到角位移。然后，对由四元数空间向姿态角空间转换方法进行了研究：通过三次厄米特插值得到节点  $u$  关于角位移  $\theta$  的关系，进而可得节点  $u$  关于时间的关系，通过差分法得到节点  $u$  关于时间的一二阶导数，并由此与四元数样条曲线及其一二阶导数得到矢量化的角速度及角加速度。最后，对 S 型速度规划算法进行了研究，针对规划前提给定位移和时间的不同，将规划分为位移层和时间层进行分析研究，同时由于预规划的需要将速度层作为速度规划的底层进行设计。

(4) 基于 MATLAB 仿真软件对上述基于四元数曲线的姿态轨迹规划相关算法的有效性 & 可行性进行仿真验证。主要对六自由度机器人模型的运动学相关算法、四元数样条曲线生成、球面线性插补曲线生成、四元数过渡曲线生成的连续性及平滑性进行了验证，对曲线拟合及反向插值、基于四元数曲线的姿态轨迹规划算法的有效性进行了验证。

由于时间及本人精力有限，本文还有一些问题需要进一步进行优化 & 完善。主要有以下三个方面：

(1) 分段点的确定：在位置轨迹规划中，当曲线曲率超过一定限度时就要对速度进行处理，否则会引起力超过系统所能提供的值或急剧变化，进而会造成机械臂震动。类似于位置轨迹规划，姿态轨迹规划也会出现相类似的情况。当角速度或转轴变化较快时，同样会引起动量矩的变化，因此同样需要进行限制。但不同于位置轨迹规划的是姿态轨迹规划的插补曲线不是直接采用姿态角，而是采用四元数样条曲线间接进行插补。在位置插补中，可以直接通过曲线的曲率半径及速度计算出所受力的大小；但姿态轨迹规划中四元数曲线的曲率、姿态角速度与动量矩之间的关系并没有这么明确。因此，轨迹规划中的分段点较难确定。

(2) 非均匀四元数样条曲线：鉴于在反求四元数样条曲线控制点时的条件方程为非线性方程，当示教点多时，进行求解时只能够采用数值解法。而要采用数值解法，方程的形式要统一，在此情况下只有使得节点是均匀的。因此，解决非均匀样条曲线控制点的反求还需要新方法。

(3) 计算速度问题：本文所研究的姿态轨迹规划方法中，首先需要将姿态示教点转化为四元数，然后构造相应的四元数曲线进行姿态插值，而后还要求解曲线导数，并进行姿态角位移拟合及反向插值，再进行 S 型速度规划，还需要经规划结果转化为姿态角速度及角加速度。虽然该算法所得姿态轨迹规划结果连续，但所涉及算法复杂，并且所涉及四元数运算多为非线性，不能够很好地满足实时性要求。因此，还需要对算法进行简化 & 优化，以适应对实时性要求较高的场合。

## 致 谢

时光荏苒，岁月如梭。不知不觉中在西南科技大学的三年研究生生活即将结束。在这三年中，有很多需要感激和感谢的人和事，其中最需要感谢的人就是我的导师梁艳阳博士。梁艳阳博士知识渊博，思路开阔，总能给人以启迪。虽与我们为师生关系，但同为八零后的他从没以老师看待学生的眼光看待我们，而始终以朋友的心态与我们交流，真正做到了亦师亦友。在指导学生方面，梁艳阳博士更是孜孜不倦，总是教导学生底子薄不可怕，但要肯努力。正是在梁艳阳博士的指导下，把我这样一位数学专业出身的学生引入机器人之门，并体会到了科学研究的快乐。

感谢西南科技大学这个美丽的大学校园给我三年的研究生阶段提供了非常好的学习和生活平台，也给我留下了美好的回忆。同时，还要感谢工业机器人研究室的老师和同学们。与他们在一起就像一个大家庭，一起学习研究，一起运动玩耍。

更应该感谢的还有我的家人及朋友。尤其是我最亲爱的父母，养育了我二十多年，却从未对我有过什么要求。谨以此文献给我的父母！

---

## 参考文献

- [1] 王田苗, 陶永. 我国工业机器人技术现状与产业化发展战略 [J]. 机械工程学报, 2014, 50 (9): 1-13.
- [2] Niku Saeed B, 孙富春, 朱纪洪, et al. 机器人学导论: 分析, 系统及应用 [M]. 电子工业出版社, 2004.
- [3] 反省机器人 30 年历程 中国机器人四大疑问 .  
<http://robot.ofweek.com/2014-12/ART-8321202-8420-28918284.html>, 2014.
- [4] 大话机器人. <http://finance.sina.com.cn/stock/hyyj/20140227/000618351492.shtml>, 2014.
- [5] 徐晖. 政策支持 机器人产业迎良机 [J]. 电器工业, 2014, (008): 58-59.
- [6] 杨进, 李鸿基, 梁殿胜. 基于 PMAC 的工业机器人示教再现程序的研发 [J]. 机床与液压, 2013, 41 (15): 151-154.
- [7] 蔡自兴. 机器人学 [M]. 清华大学出版社, 2000.
- [8] 谭民, 徐福, 侯增广. 先进机器人控制 [M]. 高等教育出版社, 2007.
- [9] 孙亮, 马江, 阮晓钢. 六自由度机械臂轨迹规划与仿真研究 [J]. 控制工程, 2010, 17 (3): 388-392.
- [10] 韩军, 郝立. 机器人关节空间的轨迹规划及仿真 [J]. 南京理工大学学报: 自然科学版, 2000, 24 (6): 540-543.
- [11] 马强. 六自由度机械臂轨迹规划研究 [D]. 哈尔滨: 哈尔滨工程大学, 2007.
- [12] 方菲, 刘平安. 抛物线过渡的并联机器人线性轨迹规划 [J]. 华东交通大学学报, 2006, 22 (4): 113-116.
- [13] Xingguo Zhang, Aihua Yin. The path planning of robot for the assembly of compressor based on the method of Parabola transition [J]. Machine Tool and Hydraulics, 2005, 6 52-54.
- [14] Thompson Stuart E, Patel Rajnikant V. Formulation of joint trajectories for industrial robots using B-splines [J]. IEEE Transactions on Industrial Electronics, 1987, (2): 192-199.
- [15] 孙秀萍. 机器人手臂轨迹规划的研究 [J]. 内蒙古师范大学学报: 自然科学版, 2009, 38 (2): 174-177.
- [16] Khalil Wisama, Dombre Etienne. Modeling, identification and control of robots [M]. Butterworth-Heinemann, 2004.
- [17] Biagiotti Luigi, Melchiorri Claudio. Trajectory planning for automatic machines and robots [M]. Springer, 2008.
- [18] 陈伟华. 工业机器人笛卡尔空间轨迹规划的研究 [D]: 广州: 华南理工大学, 2010.
- [19] 张升. 四自由度机械手轨迹规划及运动控制 [D]: 长安大学, 2012.

- 
- [20] 林威, 江五讲. 工业机器人笛卡尔空间轨迹规划 [J]. 机械工程与自动化, 2014, (5): 141-143.
- [21] Daly Leonard, Brutzman Don. X3D: extensible 3D graphics standard [J]. IEEE Signal Processing Magazine, 2007, 24 (6): 130-135.
- [22] 毛征宇, 刘中坚. 一种三次均匀 B 样条曲线的轨迹规划方法 [J]. 中国机械工程, 2010, (21): 2569-2572.
- [23] 张红强, 章兢, 王耀南, et al. 机器人关节空间 B 样条轨迹设计的混沌优化 [J]. 电机与控制学报, 2007, 11 (2): 174-177.
- [24] 赖永林, 林茂松, 梁艳阳. 基于三次非均匀 B 样条曲线的机器人轨迹规划算法研究 [J]. 科学技术与工程, 2013, 13 (35): 10511-10517.
- [25] 王文莉, 黄祖广, 胡天亮, et al. NURBS 自适应实时插补算法的研究 [J]. 制造技术与机床, 2014, (6): 148-151.
- [26] 郑秀娟. 移动机械臂的运动控制与轨迹规划算法研究 [D]: 武汉科技大学, 2012.
- [27] 刘松国, 朱世强, 王宣银, et al. 基于四元数和 B 样条的机械手平滑姿态规划器 [J]. 浙江大学学报: 工学版, 2009, 43 (7): 1192-1196.
- [28] 陈伟华, 张铁, 崔敏其. 基于五次多项式过渡的机器人轨迹规划的研究 [J]. 煤矿机械, 2012, 32 (12): 49-50.
- [29] 刘爱东, 黄彬, 卢中武. 基于四元数的坐标旋转与插值方法研究 [J]. 计算机与现代化, 2012, (2): 44-47.
- [30] 肖尚彬. 四元数方法及其应用 [J]. 力学进展, 1993, 23 (2): 249-260.
- [31] Yongguo Zhu, Xiang Huang, Wei Fang, et al. Trajectory planning algorithm based on quaternion for 6-DOF aircraft wing automatic position and pose adjustment method [J]. Chinese Journal of Aeronautics, 2010, 23 (6): 707-714.
- [32] 周江华, 苗育红. 四元数在刚体姿态仿真中的应用研究 [J]. 飞行力学, 2000, 18 (4): 28-32.
- [33] 邢燕. 四元数及其在图形图像处理中的应用研究 [D]: 合肥工业大学, 2009.
- [34] Pletinckx Daniel. Quaternion calculus as a basic tool in computer graphics [J]. The Visual Computer, 1989, 5 (1-2): 2-13.
- [35] 刘放, 汪鏊, 胡俊, et al. 基于四元数的空间圆弧插补算法 [J]. 机械科学与技术, 2009, 28 (11): 1425-1428.
- [36] Chiaverini Stefano, Siciliano Bruno. The unit quaternion: a useful tool for inverse kinematics of robot manipulators [J]. Systems Analysis Modelling Simulation, 1999, 35 (1): 45-60.
- [37] Kong Min-xiu, Ji Chen, Chen Zheng-Sheng, et al. Application of Orientation Interpolation of Robot Using Unit Quaternion[C]; proceedings of the the IEEE International Conference on
-

- Information and Automation, Yinchuan, China, F, 2013: 384-389.
- [38] Dam Erik B, Koch Martin, Lillholm Martin. Quaternions, interpolation and animation [M]. Technical Report DIKU-TR-98/5: Datalogisk Institut, Københavns Universitet, 1998.
- [39] 任秉银, 梁兆东, 孔民秀. 机械手空间圆弧位姿轨迹规划算法的实现 [J]. 哈尔滨工业大学学报, 2012, 44 (7): 27-31.
- [40] 季晨. 工业机器人姿态规划及轨迹优化研究 [D]. 哈尔滨: 哈尔滨工业大学, 2013.
- [41] 张小明, 朱利民, 丁汉, et al. 五轴加工刀具姿态球面 NURBS 曲线设计及优化 [J]. 机械工程学报, 2010, 46 (17): 140-144.
- [42] Kim Myoung-Jun, Kim Myung-Soo, Shin Sung Yong. A general construction scheme for unit quaternion curves with simple high order derivatives[C]; proceedings of the Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, New York, F, 1995: 369-376.
- [43] Kim Myoung-Jun, Kim Myung-Soo, Shin Sung Yong. A  $C^2$ -continuous B-spline quaternion curve interpolating a given sequence of solid orientations[C]; proceedings of the Proceedings of Computer Animation, F 19-21 Apr 1995, 1995: 72-81.
- [44] Ge Wenbing, Huang Zhangjin, Wang Guoping. Interpolating Solid Orientations with a  $C^2$ -Continuous B-Spline Quaternion Curve [M]. Technologies for E-Learning and Digital Entertainment. Springer. 2007: 606-615.
- [45] Miura Kenjiro T. Unit quaternion integral curve: a new type of fair free-form curves [J]. Computer Aided Geometric Design, 2000, 17 (1): 39-58.
- [46] Pao Richard L. Euler angle transformations [J]. IEEE Transactions on Automatic Control, 1966, 11 (4): 707-715.
- [47] Diebel James. Representing attitude: Euler angles, unit quaternions, and rotation vectors [J]. Matrix, 2006, 58 15-16.
- [48] Denavit Jacques. A kinematic notation for lower-pair mechanisms based on matrices [J]. Trans of the ASME Journal of Applied Mechanics, 1955, 22 215-221.
- [49] Spong Mark W, Hutchinson Seth, Vidyasagar Mathukumalli. Robot modeling and control [M]. Wiley New York, 2006.
- [50] 帕贝里. 3D 数学基础: 图形与游戏开发 [M]. 清华大学出版社有限公司, 2005.
- [51] Chen BJ, Shu HZ, Zhang Hui, et al. Quaternion Zernike moments and their invariants for color image analysis and object recognition [J]. Signal Processing, 2012, 92 (2): 308-318.
- [52] Ell Todd A, Bihan Nicolas Le, Sangwine Stephen J. Quaternion Fourier Transforms [J]. Quaternion Fourier Transforms for Signal and Image Processing, 2014, 35-66.

- 
- [53] Salmeron-Quiroz BB, Castellanos JFG, Paredes SAR, et al. Global estimation of robot's attitude via quaternion and data fusion[C]; proceedings of the 7th IEEE Conference on Industrial Electronics and Applications, F, 2012: 524-529.
- [54] 范奎武. 描述人造地球卫星轨道的四元数法 [J]. 航天控制, 2011, (06): 14-16+21.
- [55] Watt Alan, Watt Mark. Advanced Animation and Rendering Techniques Theory and Practice, [M]. Wokingham, England: Addison-Wesley, 1994.
- [56] Kim Myoung-Jun, Kim Myung-Soo, Shin Sung Yong. A compact differential formula for the first derivative of a unit quaternion curve [J]. Journal of Visualization and Computer Animation, 1996, 7 (1): 43-57.
- [57] 吴基斌. 运动控制关键算法及嵌入式实现研究 [D]: 华南理工大学, 2010.
- [58] Kim Myung-Soo, Nam Kee-Won. Hermite interpolation of solid orientations with circular blending quaternion curves [J]. Journal of Visualization and Computer Animation, 1996, 7 (2): 95-110.
- [59] Lei W. T., Sung M. P., Lin L. Y., et al. Fast real-time NURBS path interpolation for CNC machine tools [J]. International Journal of Machine Tools and Manufacture, 2007, 47 (10): 1530-1541.
- [60] Lindfield George R, Penny John ET. Numerical methods: using MATLAB(third edition) [M]. Academic Press, 2012.
-



## 附录 1

根据 2.3.1 中机器人运动学(位置)正解算法所得的位姿矩阵为:

$${}^0T_t = \begin{bmatrix} s_x & n_x & a_x & p_x \\ s_y & n_y & a_y & p_y \\ s_z & n_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

其中:

$$\mathbf{s_x} = c5*c1*c23 - s5*(s1*s4 + c4*c1*s23);$$

$$\mathbf{s_y} = s5*(c1*s4 - c4*s1*s23) + c5*s1*c23;$$

$$\mathbf{s_z} = -c4*s5*c23 - c5*s23;$$

$$\mathbf{n_x} = s6*(s5*c1*c23 + c5*(s1*s4 + c4*c1*s23)) - c6*(c4*s1 - s4*c1*s23);$$

$$\mathbf{n_y} = c6*(c1*c4 + s4*s1*s23) + s6*(s5*s1*c23 - c5*(c1*s4 - c4*s1*s23));$$

$$\mathbf{n_z} = -s6*(s5*s23 - c4*c5*c23) + c6*s4*c23;$$

$$\mathbf{a_x} = s6*(c4*s1 - s4*c1*s23) + c6*(s5*c1*c23 + c5*(s1*s4 + c4*c1*s23));$$

$$\mathbf{a_y} = c6*(s5*s1*c23 - c5*(c1*s4 - c4*s1*s23)) - s6*(c1*c4 + s4*s1*s23);$$

$$\mathbf{a_z} = -s4*s6*c23 - c6*(s5*s23 - c4*c5*c23);$$

$$\mathbf{p_x} = c1*d1 + r6*(c5*c1*c23 - s5*(s1*s4 + c4*c1*s23)) + d3*c1*s23 + r4*c1*c23 + c1*d2*s2;$$

$$\mathbf{p_y} = d1*s1 + d3*s1*s23 + r6*(c5*s1*c23 + s5*(c1*s4 - c4*s1*s23)) + r4*s1*c23 + d2*s1*s2;$$

$$\mathbf{p_z} = c2*d2 + d3*c23 - r4*s23 - r6*(c5*s23 + c4*s5*c23);$$

上式中,  $c_i$ 、 $s_i$  分别表示  $\cos\theta_i$ 、 $\sin\theta_i$ ;  $c23$ 、 $s23$  分别表示  $\cos(\theta_2+\theta_3)$ 、 $\sin(\theta_2+\theta_3)$ ;  $r_j$ 、 $d_k$  分别表示相应连杆参数, 参见表2-1。

## 附录 2

根据 2.4 中机器人雅克比矩阵计算所得的雅克比矩阵为:

$$J=[J1 \ J2 \ J3 \ J4 \ J5 \ J6]$$

其中:

$$\mathbf{J1}=[-d1*s1-d3*s1*s23-r6*(c5*s1*c23+s5*(c1*s4-c4*s1*s23))-r4*s1*c23-d2*s1*s2;$$

$$c1*d1+r6*(c5*c1*c23-s5*(s1*s4+c4*c1*s23))+d3*c1*s23+r4*c1*c23+c1*d2*s2;$$

$$0;0;0;1];$$

$$\mathbf{J2}=[c1*(c2*d2+d3*c23-r4*s23-r6*(c5*s23+c4*s5*c23));s1*(c2*d2+d3*c23-r4*s23-r6*(c5*s23+c4*s5*c23));$$

$$-s1*(d3*s1*s23+r6*(c5*s1*c23+s5*(c1*s4-c4*s1*s23))+r4*s1*c23+d2*s1*s2)-c1*(r6*(c5*c1*c23-s5*(s1*s4+c4*c1*s23))+d3*c1*s23+r4*c1*c23+c1*d2*s2);$$

$$-s1;c1;0];$$

$$\mathbf{J3}=[-c1*(r4*s23-d3*c23+r6*(c5*s23+c4*s5*c23));$$

$$-s1*(r4*s23-d3*c23+r6*(c5*s23+c4*s5*c23));$$

$$-c1*(r6*(c5*c1*c23-s5*(s1*s4+c4*c1*s23))+d3*c1*s23+r4*c1*c23)-s1*(d3*s1*s23+r6*(c5*s1*c23+s5*(c1*s4-c4*s1*s23))+r4*s1*c23);$$

$$-s1;c1;0];$$

$$\mathbf{J4}=[r6*(c5*s1*c23+s5*(c1*s4-c4*s1*s23))*s23-r6*s1*c23*(c5*s23+c4*s5*c23);$$

$$r6*c1*c23*(c5*s23+c4*s5*c23)-r6*(c5*c1*c23-s5*(s1*s4+c4*c1*s23))*s23;$$

$$r6*c1*c23*(c5*s1*c23+s5*(c1*s4-c4*s1*s23))-r6*(c5*c1*c23-s5*(s1*s4+c4*c1*s23))*s1*c23;$$

$$c1*c23;s1*c23;-s23];$$

$$\mathbf{J5}=[-r6*(c1*c4+s4*s1*s23)*(c5*s23+c4*s5*c23)-r6*s4*(c5*s1*c23+s5*(c1*s4-c4*s1*s23))*c23;$$

$$r6*s4*(c5*c1*c23-s5*(s1*s4+c4*c1*s23))*c23-$$

---

```

r6*(c5*s23 + c4*s5*c23)*(c4*s1 - s4* c1* s23);
    - r6*(c1*c4 + s4*s1*s23)*(c5* c1* c23 - s5*(s1*s4 + c4* c1*
s23)) - r6*(c5*s1*c23 + s5*(c1*s4 - c4* s1* s23))*(c4*s1 - s4*
c1* s23);
    s4* c1* s23 - c4*s1;c1*c4 + s4*s1*s23;s4*c23];
J6=[0; 0; 0;
    c5* c1* c23 - s5*(s1*s4 + c4* c1* s23);
    c5*s1*c23 + s5*(c1*s4 - c4* s1* s23);
    - c5*s23 - c4*s5*c23];

```

上述雅克比矩阵元素中符合的含义同附录 1。

---

## 附录 3

对附录 2 中机器人雅克比矩阵求关于时间的导数可得雅克比矩阵导数为：

$$dJ = [dJ1 \quad dJ2 \quad dJ3 \quad dJ4 \quad dJ5 \quad dJ6]$$

其中：

$$\begin{aligned} dJ1 = & [v3 * (r6 * (s23 * c5 * s1 + c23 * c4 * s1 * s5) - d3 * c23 * s1 + r4 * s23 * s1) \\ & + v2 * (r6 * (s23 * c5 * s1 + c23 * c4 * s1 * s5) - d3 * c23 * s1 + r4 * s23 * s1 - d2 * c2 \\ & * s1) - v1 * (d1 * c1 - r6 * (s5 * (s1 * s4 + s23 * c1 * c4) - c23 * c1 * c5) + r4 * c23 \\ & * c1 + d3 * s23 * c1 + d2 * c1 * s2) - r6 * v5 * (c5 * (c1 * s4 - s23 * c4 * s1) - c23 * s1 \\ & * s5) - r6 * v4 * s5 * (c1 * c4 + s23 * s1 * s4); \\ & - v1 * (r6 * (s5 * (c1 * s4 - s23 * c4 * s1) + c23 * c5 * s1) + d1 * s1 + r4 * c23 \\ & * s1 + d3 * s23 * s1 + d2 * s1 * s2) - v2 * (r6 * (s23 * c1 * c5 + c23 * c1 * c4 * s5) - d3 \\ & * c23 * c1 + r4 * s23 * c1 - d2 * c1 * c2) - v3 * (r6 * (s23 * c1 * c5 + c23 * c1 * c4 * s5) \\ & - d3 * c23 * c1 + r4 * s23 * c1) - r6 * v5 * (c5 * (s1 * s4 + s23 * c1 * c4) + c23 * c1 * s5) \\ & - r6 * v4 * s5 * (c4 * s1 - s23 * c1 * s4); \\ & 0; 0; 0; 0]; \\ dJ2 = & [r6 * v5 * c1 * (s23 * s5 - c23 * c4 * c5) - v2 * c1 * (r4 * c23 + d3 * s23 + d2 * s2 \\ & + r6 * (c23 * c5 - s23 * c4 * s5)) - v3 * c1 * (r4 * c23 + d3 * s23 + r6 * (c23 * c5 \\ & - s23 * c4 * s5)) - v1 * s1 * (d3 * c23 - r4 * s23 + d2 * c2 - r6 * (s23 * c5 + c23 * c4 \\ & * s5)) + r6 * v4 * c23 * c1 * s4 * s5; \\ & v1 * c1 * (d3 * c23 - r4 * s23 + d2 * c2 - r6 * (s23 * c5 + c23 * c4 * s5)) - v2 \\ & * s1 * (r4 * c23 + d3 * s23 + d2 * s2 + r6 * (c23 * c5 - s23 * c4 * s5)) - v3 * s1 * (r4 \\ & * c23 + d3 * s23 + r6 * (c23 * c5 - s23 * c4 * s5)) + r6 * v5 * s1 * (s23 * s5 - c23 * c4 \\ & * c5) + r6 * v4 * c23 * s1 * s4 * s5; \\ & v3 * (c1 * (r6 * (s23 * c1 * c5 + c23 * c1 * c4 * s5) - d3 * c23 * c1 + r4 * s23 \\ & * c1) + s1 * (r6 * (s23 * c5 * s1 + c23 * c4 * s1 * s5) - d3 * c23 * s1 + r4 * s23 * s1)) \\ & - v5 * (r6 * s1 * (c5 * (c1 * s4 - s23 * c4 * s1) - c23 * s1 * s5) - r6 * c1 * (c5 * (s1 \\ & * s4 + s23 * c1 * c4) + c23 * c1 * s5)) + v4 * (r6 * c1 * s5 * (c4 * s1 - s23 * c1 * s4) \\ & - r6 * s1 * s5 * (c1 * c4 + s23 * s1 * s4)) + v2 * (s1 * (r6 * (s23 * c5 * s1 + c23 * c4 \\ & * s1 * s5) - d3 * c23 * s1 + r4 * s23 * s1 - d2 * c2 * s1) + c1 * (r6 * (s23 * c1 * c5 + c23 \\ & * c1 * c4 * s5) - d3 * c23 * c1 + r4 * s23 * c1 - d2 * c1 * c2)); \\ & -v1 * c1; -v1 * s1; 0]; \end{aligned}$$

$$dJ3=[v1*s1*(r4*s23-d3*c23+r6*(s23*c5+c23*c4*s5))-v3*c1*(r4*c23+d3*s23+r6*(c23*c5-s23*c4*s5))-v2*c1*(r4*c23+d3*s23+r6*(c23*c5-s23*c4*s5))+r6*v5*c1*(s23*s5-c23*c4*c5)+r6*v4*c23*c1*s4*s5;$$

$$r6*v5*s1*(s23*s5-c23*c4*c5)-v2*s1*(r4*c23+d3*s23+r6*(c23*c5-s23*c4*s5))-v3*s1*(r4*c23+d3*s23+r6*(c23*c5-s23*c4*s5))-v1*c1*(r4*s23-d3*c23+r6*(s23*c5+c23*c4*s5))+r6*v4*c23*s1*s4*s5;$$

$$v2*(c1*(r6*(s23*c1*c5+c23*c1*c4*s5)-d3*c23*c1+r4*s23*c1)+s1*(r6*(s23*c5*s1+c23*c4*s1*s5)-d3*c23*s1+r4*s23*s1))+v3*(c1*(r6*(s23*c1*c5+c23*c1*c4*s5)-d3*c23*c1+r4*s23*c1)+s1*(r6*(s23*c5*s1+c23*c4*s1*s5)-d3*c23*s1+r4*s23*s1))-v5*(r6*s1*(c5*(c1*s4-s23*c4*s1)-c23*s1*s5)-r6*c1*(c5*(s1*s4+s23*c1*c4)+c23*c1*s5))+v4*(r6*c1*s5*(c4*s1-s23*c1*s4)-r6*s1*s5*(c1*c4+s23*s1*s4));$$

$$-v1*c1;-v1*s1;0];$$

$$dJ4=[v4*(r6*s1*s4*s5*c23^2+r6*s23*s5*(c1*c4+s23*s1*s4))-v1*(r6*s23*(s5*(s1*s4+s23*c1*c4)-c23*c1*c5)+r6*c23*c1*(s23*c5+c23*c4*s5))+v5*(r6*s23*(c5*(c1*s4-s23*c4*s1)-c23*s1*s5)+r6*c23*s1*(s23*s5-c23*c4*c5))-v2*(r6*s23*(s23*c5*s1+c23*c4*s1*s5)-r6*c23*(s5*(c1*s4-s23*c4*s1)+c23*c5*s1)+r6*c23*s1*(c23*c5-s23*c4*s5)-r6*s23*s1*(s23*c5+c23*c4*s5))-v3*(r6*s23*(s23*c5*s1+c23*c4*s1*s5)-r6*c23*(s5*(c1*s4-s23*c4*s1)+c23*c5*s1)+r6*c23*s1*(c23*c5-s23*c4*s5)-r6*s23*s1*(s23*c5+c23*c4*s5))-v1*(r6*s23*(s5*(c1*s4-s23*c4*s1)+c23*c5*s1)-r6*c23*s1*(s23*c5+c23*c4*s5))];$$

$$v2*(r6*c23*(s5*(s1*s4+s23*c1*c4)-c23*c1*c5)+r6*s23*(s23*c1*c5+c23*c1*c4*s5)+r6*c23*c1*(c23*c5-s23*c4*s5)-r6*s23*c1*(s23*c5+c23*c4*s5))+v3*(r6*c23*(s5*(s1*s4+s23*c1*c4)-c23*c1*c5)+r6*s23*(s23*c1*c5+c23*c1*c4*s5)+r6*c23*c1*(c23*c5-s23*c4*s5)-r6*s23*c1*(s23*c5+c23*c4*s5))+v5*(r6*s23*(c5*(s1*s4+s23*c1*c4)+c23*c1*s5)-r6*c23*c1*(s23*s5-c23*c4*c5))+v1*(r6*s23*(s5*(c1*s4-s23*c4*s1)+c23*c5*s1)-r6*c23*s1*(s23*c5+c23*c4*s5))];$$

$*c5+c23*c4*s5))+v4*(r6*s23*s5*(c4*s1-s23*c1*s4)-r6*c23^2*c1*s4*s5);$

$v5*(r6*c23*s1*(c5*(s1*s4+s23*c1*c4)+c23*c1*s5)+r6*c23*c1*(c5*(c1*s4-s23*c4*s1)-c23*s1*s5))-v3*(r6*s23*c1*(s5*(c1*s4-s23*c4*s1)+c23*c5*s1)+r6*s23*s1*(s5*(s1*s4+s23*c1*c4)-c23*c1*c5)-r6*c23*s1*(s23*c1*c5+c23*c1*c4*s5)+r6*c23*c1*(s23*c5*s1+c23*c4*s1*s5))-v2*(r6*s23*c1*(s5*(c1*s4-s23*c4*s1)+c23*c5*s1)+r6*s23*s1*(s5*(s1*s4+s23*c1*c4)-c23*c1*c5)-r6*c23*s1*(s23*c1*c5+c23*c1*c4*s5)+r6*c23*c1*(s23*c5*s1+c23*c4*s1*s5))+v4*(r6*c23*c1*s5*(c1*c4+s23*s1*s4)+r6*c23*s1*s5*(c4*s1-s23*c1*s4));$

$-v1*c23*s1-v2*s23*c1-v3*s23*c1;$

$v1*c23*c1-v2*s23*s1-v3*s23*s1;-v2*c23-v3*c23];$

$dJ5=[v1*(r6*(c4*s1-s23*c1*s4)*(s23*c5+c23*c4*s5)+r6*c23*s4*(s5*(s1*s4+s23*c1*c4)-c23*c1*c5))-v3*(r6*(c1*c4+s23*s1*s4)*(c23*c5-s23*c4*s5)-r6*s23*s4*(s5*(c1*s4-s23*c4*s1)+c23*c5*s1)-r6*c23*s4*(s23*c5*s1+c23*c4*s1*s5)+r6*c23*s1*s4*(s23*c5+c23*c4*s5))-v2*(r6*(c1*c4+s23*s1*s4)*(c23*c5-s23*c4*s5)-r6*s23*s4*(s5*(c1*s4-s23*c4*s1)+c23*c5*s1)-r6*c23*s4*(s23*c5*s1+c23*c4*s1*s5)+r6*c23*s1*s4*(s23*c5+c23*c4*s5))+v4*(r6*(c1*s4-s23*c4*s1)*(s23*c5+c23*c4*s5)-r6*c23*c4*(s5*(c1*s4-s23*c4*s1)+c23*c5*s1))+v5*(r6*(c1*c4+s23*s1*s4)*(s23*s5-c23*c4*c5)-r6*c23*s4*(c5*(c1*s4-s23*c4*s1)-c23*s1*s5));$

$v4*(r6*(s1*s4+s23*c1*c4)*(s23*c5+c23*c4*s5)-r6*c23*c4*(s5*(s1*s4+s23*c1*c4)-c23*c1*c5))-v3*(r6*(c4*s1-s23*c1*s4)*(c23*c5-s23*c4*s5)-r6*s23*s4*(s5*(s1*s4+s23*c1*c4)-c23*c1*c5)+r6*c23*s4*(s23*c1*c5+c23*c1*c4*s5)-r6*c23*c1*s4*(s23*c5+c23*c4*s5))-v2*(r6*(c4*s1-s23*c1*s4)*(c23*c5-s23*c4*s5)-r6*s23*s4*(s5*(s1*s4+s23*c1*c4)-c23*c1*c5)+r6*c23*s4*(s23*c1*c5+c23*c1*c4*s5)-r6*c23*c1*s4*(s23*c5+c23*c4*s5))+v5*(r6*(c4*s1-s23*c1*s4)*(s23*s5-c23*c4*c5)-r6*c23*s4*(c5*(s1*s4+s23*c1*c4)+c23*c1*s5))-v1*(r6*(c1*c4+s23*s1*s4)*(s23*c5+c23*c4*s5)+r6*c23*s4*(s5*(c1*s4-s23*c4*s1)+c23*c5*s1));$

$$\begin{aligned}
& v5*(r6*(c1*c4 + s23*s1*s4)*(c5*(s1*s4 + s23*c1*c4) + c23*c1*s5) - r6*(c4*s1 - s23*c1*s4)*(c5*(c1*s4 - s23*c4*s1) - c23*s1*s5)) \\
& - v4*(r6*(c1*s4 - s23*c4*s1)*(s5*(s1*s4 + s23*c1*c4) - c23*c1*c5) - r6*(s1*s4 + s23*c1*c4)*(s5*(c1*s4 - s23*c4*s1) + c23*c5*s1)) + v2 \\
& *(r6*(c1*c4 + s23*s1*s4)*(s23*c1*c5 + c23*c1*c4*s5) + r6*(c4*s1 - s23*c1*s4)*(s23*c5*s1 + c23*c4*s1*s5) + r6*c23*c1*s4*(s5*(c1*s4 - s23*c4*s1) + c23*c5*s1) + r6*c23*s1*s4*(s5*(s1*s4 + s23*c1*c4) - c23*c1*c5)) + v3*(r6*(c1*c4 + s23*s1*s4)*(s23*c1*c5 + c23*c1*c4*s5) + r6*(c4*s1 - s23*c1*s4)*(s23*c5*s1 + c23*c4*s1*s5) + r6*c23*c1*s4*(s5*(c1*s4 - s23*c4*s1) + c23*c5*s1) + r6*c23*s1*s4*(s5*(s1*s4 + s23*c1*c4) - c23*c1*c5));
\end{aligned}$$

$$v4*(s1*s4 + s23*c1*c4) - v1*(c1*c4 + s23*s1*s4) + v2*c23*c1*s4 + v3*c23*c1*s4;$$

$$v2*c23*s1*s4 - v4*(c1*s4 - s23*c4*s1) - v1*(c4*s1 - s23*c1*s4) + v3*c23*s1*s4;$$

$$v4*c23*c4 - v2*s23*s4 - v3*s23*s4];$$

$$dJ6=[0;0;0;$$

$$\begin{aligned}
& -v5*(c5*(s1*s4 + s23*c1*c4) + c23*c1*s5) - v1*(s5*(c1*s4 - s23*c4*s1) + c23*c5*s1) - v2*(s23*c1*c5 + c23*c1*c4*s5) - v3*(s23*c1*c5 + c23*c1*c4*s5) - v4*s5*(c4*s1 - s23*c1*s4);
\end{aligned}$$

$$\begin{aligned}
& v5*(c5*(c1*s4 - s23*c4*s1) - c23*s1*s5) - v1*(s5*(s1*s4 + s23*c1*c4) - c23*c1*c5) - v2*(s23*c5*s1 + c23*c4*s1*s5) - v3*(s23*c5*s1 + c23*c4*s1*s5) + v4*s5*(c1*c4 + s23*s1*s4);
\end{aligned}$$

$$v5*(s23*s5 - c23*c4*c5) - v3*(c23*c5 - s23*c4*s5) - v2*(c23*c5 - s23*c4*s5) + v4*c23*s4*s5];$$

其中,  $v_i(i=1,2,3,4,5)$ 为对应关节角速度; 上述雅克比矩阵导数矩阵元素中其他符号的含义同附录 1。

## 攻读硕士学位期间发表的学术论文及研究成果

### 发表的学术论文：

- [1] 王效杰, 梁艳阳, 熊莉英, 张平. Delta 机器人期望工作空间求解算法研究[J]. 机械设计与制造, 2014 (9): 175-177+180.
- [2] 梁艳阳, 王效杰, 刘恒. Delta 机器人结构参数与期望工作空间之间关系研究[J]. 机械设计与制造, 2015, 02: 154-156+160.
- [3] 刘宏伟, 王效杰, 梁艳阳, 张健. 机器人末端执行器姿态轨迹规划研究[J]. 机械设计与制造, 2015, 04: 28-30+34.

### 攻读硕士期间参与科研项目：

- [1] 面向稻田监测的柔索悬吊机器人运动控制关键技术研究(四川省科技计划项目)
- [2] 高速高精度数控系统的样条轨迹规划技术研究(四川省教育厅)