

西 南 科 技 大 学

# 工 程 硕 士 专 业 学 位 论 文

基于三次样条的

自适应可控制轨迹规划技术

作 者 姓 名 雷利强

所 在 学 院 信息工程学院

专 业/领 域 控制工程

学 校 导 师 梁艳阳 副教授

校 外 导 师 田超 高级工程师

论文完成日期 2015 年 4 月 20 日

Classified Index: TP391.9

U. D. C: 621.3

Southwest University of Science and Technology

Master Degree Thesis

The adaptive and controllable trajectory  
planning technology based on cubic spline

Grade: 2012

Candidate: Lei Li Qiang

Academic Degree Appliedfor: Master

Speciality: Control Engineering

Supervisor: Liang Yan Yang

Apr. 20. 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：雷利强

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：雷利强

导师签名：梁新

日期：2015.5.26

## 摘 要

国内工业机器人发展起步较晚，与世界发达国家还有一定距离。轨迹规划技术是工业机器人运动控制系统中的一个核心模块，研究轨迹规划技术对提升工业机器人的性能起着关键性的作用。

本文以通用六自由度工业机器人为平台，分析了机器人运动学，设计了机器人位姿正逆解算法，并将轨迹规划技术抽象成两层（速度规划层和轨迹生成层）进行研究。在速度规划层，以高性能的 S 型速度控制曲线为基础，设计了两种自适应可控 S 型速度规划器：基于给定运行速度的 S 型速度规划器和基于给定运行时间的 S 型速度规划器。在轨迹生成层，研究了几种笛卡尔空间轨迹：直线、圆弧和三次 B 样条；并研究了空间直线-直线、空间直线-圆弧和空间三次非均匀 B 样条-三次非均匀 B 样条轨迹之间的平滑过渡轨迹生成算法；完成对空间轨迹和空间过渡轨迹的 S 型速度规划。最后，使用 C++ 语言对算法进行封装并在 MATLAB 环境下对测试结果进行仿真。

通过大量的实验测试，结果表明，所设计的基于三次样条的轨迹规划技术达到了自适应可控的效果。另外，通过仿真图验证了基于三次样条的轨迹规划算法的正确性。

**关键词：**正逆解算法 S 型速度规划器 三次 B 样条 轨迹规划 过渡

## Abstract

Industrial robot in domestic has a big gap with the developed countries in the world because of the development of industrial robot started behind. Trajectory planning technology is a core module of industrial robot motion control system, the research on trajectory planning technology plays a key role for enhancing the performance of industrial robots.

This paper analyzes the kinematics and designs robot kinematics forward-reverse algorithms at position and orientation based on common industrial robot with six free degrees, research trajectory planning technology that abstractively divided into two layers: speed planning and trajectory generation. In the speed planning layer, two kinds of adaptive controllable S-type speed planners based on high performance S-model speed control curve that are S-type speed planner based on a given running speed and S-type speed planner based on a given running time are designed. In the trajectory generation layer, three trajectories consisting of linear, circular and cubic B-spline are studied in cartesian space; and the smooth transition trajectory generating algorithms including linear to linear, linear to circle and non-uniform cube B-spline to non-uniform cube B-spline in cartesian space are studied; the independent and transition trajectories planning in in cartesian space that based on S-type speed planner are accomplished. Finally, algorithms are packaged in C++ language and simulation on test results are done in MATLAB environment.

Through a large number of experimental tests, the results show that the trajectory planning techniques based on cubic spline reach the effect of adaptation and controlling. Additionally, trajectory planning algorithms based on cubic spline are verified correctly by simulation figures.

**Key words:** Forward and inverse algorithms; S-type speed planner; cube B-spline; trajectory planning; transition

---

# 目 录

1	绪论	1
1.1	课题背景与研究意义	1
1.2	轨迹规划概述及研究现状	2
1.2.1	轨迹规划概述	2
1.2.2	轨迹规划国内外研究现状	2
1.3	论文的主要内容	4
1.3.1	研究内容	4
1.3.2	论文结构	5
2	机器人运动学	6
2.1	机器人空间模型描述	6
2.1.1	机器人位姿	6
2.1.2	坐标变换	8
2.2	机器人正逆解	10
2.2.1	D-H 参数建模	10
2.2.2	机器人正解	12
2.2.3	机器人逆解	14
2.3	本章小结	19
3	速度规划算法研究与设计	20
3.1	速度控制曲线	20
3.1.1	梯形速度曲线	20
3.1.2	正弦型速度曲线	21
3.1.3	标准 S 型速度曲线	23
3.2	自适应可控 S 型速度规划器	25
3.2.1	S 型速度曲线的特殊处理	26
3.2.2	基于给定运行速度的 S 型速度规划器	28
3.2.3	基于给定运行时间的 S 型速度规划器	33

---

3.2.4	S 型速度规划器类接口设计 .....	42
3.3	本章小结 .....	44
4	笛卡尔空间轨迹规划研究 .....	45
4.1	空间轨迹规划 .....	45
4.1.1	直线和圆弧轨迹规划 .....	45
4.1.2	三次 B 样条轨迹规划 .....	50
4.2	基于三次 B 样条的空间轨迹过渡 .....	57
4.2.1	直线-直线过渡 .....	57
4.2.2	直线-圆弧过渡 .....	59
4.2.3	三次非均匀 B 样条-三次非均匀 B 样条过渡 .....	61
4.3	本章小结 .....	63
5	轨迹规划算法仿真与分析 .....	64
5.1	机器人正逆解算法仿真与分析 .....	64
5.2	速度规划算法仿真与分析 .....	64
5.2.1	给定运行速度的 S 型速度规划算法 .....	64
5.2.2	给定运行时间的 S 型速度规划算法 .....	66
5.3	笛卡尔空间轨迹规划算法仿真与分析 .....	69
5.3.1	三次非均匀 B 样条轨迹规划算法 .....	69
5.3.2	直线-直线过渡轨迹规划算法 .....	72
5.3.3	直线-圆弧过渡轨迹规划算法 .....	73
5.3.4	B 样条-B 样条过渡轨迹规划算法 .....	75
5.4	本章小结 .....	77
结 论	.....	78
致 谢	.....	80
参考文献	.....	81
攻读硕士学位期间参与研究科研项目情况	.....	86

---

# 1 绪论

## 1.1 课题背景与研究意义

工业机器人诞生于 20 世纪 60 年代，在 20 世纪 90 年代得到迅速发展。它是综合了计算机、控制论、机构学、信息和传感技术、人工智能、仿生学等多学科而形成的高新技术，是当代研究十分活跃、应用日益广泛的领域。它的出现是为了适应制造业规模化生产，解决单调、重复的体力劳动和提高生产质量而代替人工作业，尤其是在繁重、高危环境下的作业，如焊接、喷涂、搬运、打磨等。工业机器人的优点在于可以提高产品的生产效率、改善产品的质量，并且能够减少产品的成本、提高能源的利用率和减轻生产过程中对环境的污染。

由于工业机器人在国外发展较早，性能优越，成本较低，相比于国内工业机器人，很多工厂更愿意选择国外的产品。据统计，现今我国工业机器人大部分都来自于进口，这里的工业机器人通常指的是工业控制领域中应用的较多的垂直式串联多关节型工业机器人，通过工业机器人技术的水平就能体现出自动化的水平，所以发展工业机器人的先进技术有利于社会生产力的进步和提升国家的综合国力<sup>[1]</sup>。

工业机器人主要由主体、驱动系统和控制系统三大部分组成，轨迹规划技术是工业机器人控制系统的一个非常核心的模块，对提升工业机器人控制系统的性能指标起着关键的作用。轨迹规划研究的本质是研究出一种可靠的算法来满足机器人末端执行器在运动过程中的稳定，防止轨迹、速度和加速度的突变带来的机器人刚体磨损和机器人系统冲击。随着工业机器人产业的发展，用户对产品需求的多样化，在机器人轨迹规划器中，轨迹规划算法应该具有自适应和可控制等特点，来满足用户的不同要求。

在某些特殊场合，对机器人的运动时间要求比较严格，为此，本文提出一种基于给定运行时间的自适应可控速度规划算法，有效地解决了机器人运动过程中涉及同步的问题。本文亦提出了分层的概念，将轨迹规划算法分为两层，底层为速度规划算法，上层为空间轨迹的生成，降低了轨迹规划器内部的耦合性，空间轨迹可以弹性地选择合适的速度规划算法来完成整段运动轨迹空间位置、速度和加速度的规划。通过本文的研究结果，为设计出稳定、可靠、通用的工业机器人运动控制器打下坚实的基础。



## 1.2 轨迹规划概述及研究现状

### 1.2.1 轨迹规划概述

工业机器人的轨迹规划从规划空间上,可分为关节空间的轨迹规划和笛卡儿空间的轨迹规划两种。轨迹规划既可在关节空间也可在笛卡尔空间中进行,但是有一个约束条件是,规划的轨迹函数都必须连续和平滑,使得机器人末端执行器的运动过程中速度平滑,在机器人运动过程中,为了避免刚体冲击造成刚体磨损、加工精度低等问题,甚至还必须满足加速度连续,以保证机器人运动平稳。

关节空间的轨迹规划是以关节角度的函数来描述机器人的轨迹的,也就是说机器人末端执行器的运动轨迹是由关节变量直接确定的,关节角度函数可以选取不同类型,主要有三次多项式插值法、高阶多项式插值法、用抛物线过渡的线性插值法等。在关节空间中进行轨迹规划时,输入量为各个关节角的角度,输出为机器人末端执行器位姿,不需要实时进行运动学逆解,计算量比较少,也容易实现。缺点是机器人末端执行器的运动轨迹是不可预知的,对于那些轨迹、姿态两者的瞬时变化规律严格要求的作业,如连续弧焊作业等,不能准确的实现避障。

针对以上关节空间规划时存在的不足,就必须在笛卡儿坐标空间进行轨迹规划,笛卡儿空间的轨迹规划是预先对期望运行轨迹进行插补,不断地进行运动学逆解,将轨迹插补点逐点地逆解到关节空间,得到各个关节期望转动的关节量,再进行关节变量的控制。笛卡尔空间的轨迹规划方法概念直观,规划的轨迹结果明确,相对于关节空间规划,笛卡尔空间的轨迹规划可以保证规划轨迹的唯一性,但在笛卡尔空间轨迹规划时,需要实时通过逆解将位姿变换为相应的关节坐标,计算量大,导致控制间隔变长。

### 1.2.2 轨迹规划国内外研究现状

在进行轨迹规划时,不仅要考虑机器人的运动学要求,同时还要综合考虑机器人的性能,比如机器人的极限工作速度、加速度、加加速度、自重、负载能力等多种因素。查阅大多数科学文献发现国内外学者在进行轨迹规划算法研究时,主要是基于以下两种原则:运动时间最优和冲击(加加速度)最优。

多项式插值经常用于轨迹规划算法中,由于其具有高阶和缺乏凸包属性等缺点,很难使用传统的优化算法进行优化,Fu R, Ju H 等人在文献<sup>[2]</sup>中提出了一种基于粒子群的时间最优的机械臂轨迹规划算法,解决了基于多项式插值的轨迹规划优化的问题。肖文皓,白瑞林等人在文献<sup>[3]</sup>中提到,在进行关节空间插补时,采用了高次的 B 样条曲线,在插补过程中将关节速度、加速度等约束条件通过映射关系与对 B 样条控制点的约束条件关联起来,最终利用信赖域法求解出了关节空间

插补最优时间。Gasparetto A 和 Zanutto V 在文献<sup>[4, 5]</sup>中分别使用了三次和五次 B 样条曲线, 以执行时间以及冲击的积分最小为约束进行了时间-冲击最优规划。随着智能算法的普及, 国内外研究人员在轨迹规划算法中逐渐地引入了智能算法, 并且取得了不错的效果。机器人末端执行器在连续通过一系列示教点过程中, Menasri R, Oulhadj H 等人在文献<sup>[6, 7]</sup>中使用遗传算法 (Genetic Algorithm, GA) 满足了运动轨迹上位置、速度和加速度平滑的基础上, 并通过论证计算了插补时间最优。文献<sup>[8-10]</sup>都先后引入了遗传算法来进行关节空间的时间最优轨迹规划研究, 并且达到了一定的效果。Costantinescu D 等人在文献<sup>[11, 12]</sup>中提出对机器人的驱动器的冲击做一些约束, 文献中也详细分析了该算法并不是严格的时间最优算法, 但它生成的轨迹是平滑的, 可以有效地被执行。

减小机器人运动过程中的刚体冲击是轨迹规划算法研究的目标之一, 实际做法往往是限制冲击的值, 最小化的冲击可以减小机器人末端执行器运动的绝对误差, 降低机器人刚体结构压力和减少机器人频率共振等, 以保证机器人能够正常稳定的运行。Simon D 和 Isik C<sup>[13]</sup>使用了三角函数曲线轨迹插补满足了加速度的连续, 从而确保冲击可控, 并且可以求取冲击的最小值。Piazzzi A 等人在文献<sup>[14, 15]</sup>中提出了一种基于区间分析的方法, 找出了基于三次样条的关节空间轨迹规划中全局最小冲击值。Macfarlane S, Croft E A 等人在文献<sup>[16, 17]</sup>中提出一种基于五次多项式串联的方法, 获得了两示教点之间平滑、冲击-有界的运动轨迹, 文中还提到这种轨迹近似于直线段与抛物线轨迹的结合。Petrinec 等人在文献<sup>[18]</sup>中通过基于五次多项式和四次多项式相结合的方式来进行轨迹规划, 规划出的空间轨迹插补点上满足空间位置、速度、加速度和加加速度连续, 从轨迹规划算法上直接对加加速度有了很好的限制。Huang P, Xu Y, Liang B 在文献<sup>[19]</sup>中提出一种新的方法, 在关节空间规划中, 基于遗传算法找出了冲击全局最优的一条运动轨迹。

相对于运动时间最优和冲击 (加加速度) 最优的研究, 底层速度规划算法模型和空间运动轨迹插补算法更是轨迹规划研究的重点, 对提升机器人实际应用中的稳定性起着关键作用。

林仕高, 刘晓麟和欧元贤在文献<sup>[20]</sup>中提出在机器人末端轨迹曲线的启停段加入正弦加减速过程, 同时在连续轨迹的两条曲线间运用五次多项式过渡曲线的新算法, 使得机器人末端轨迹速度连续平滑和加速度连续。文献<sup>[21-26]</sup>提到, 相对于正弦加减速控制曲线, S 型速度控制曲线也能够满足机器人末端轨迹速度连续平滑和加速度连续, 由于 S 型速度曲线的加速度呈梯形状, 梯形斜边的斜率为系统冲击, 在系统所承受范围内, 可改变系统冲击的大小来调整加减速性能, 从加减速性能来讲, S 型速度曲线要优于正弦加减速控制曲线, 更能最大限度的发挥机器人

的性能。

在笛卡尔空间规划中，轨迹与轨迹之间的平滑过渡可以提高工业机器人末端执行器的运动效率，而空间直线插补和空间圆弧插补又是笛卡尔空间轨迹规划最常见的两种轨迹插补方式，也是最常用的，那么如何实现空间直线-直线、空间直线-圆弧和空间圆弧-圆弧之间的平滑过渡也是轨迹规划器的一个关键技术点。一些国外厂家生产的工业机器人还提供了空间自由曲线的插补，使用空间自由曲线进行轨迹与轨迹之间的过渡，在此基础上，通过优化空间自由曲线的求解时间，成功应用到实际中。在国内，研究学者也在进行相关的研究，很多学者使用 B 样条进行曲线拟合，B 样条几乎能够拟合各种空间曲线，以其为基础的 NURBS 曲线更是成为了国际标准组织（ISO）指定的工业产品几何形状设计的唯一标准<sup>[27]</sup>。在工业自动控制领域中，文献<sup>[28-34]</sup>中将 B 样条作为空间自由曲线基础进行了相关研究，实现了空间自由轨迹的曲线插补，通过仿真验证了样条曲线的平滑性。文献<sup>[35-37]</sup>提出了 B 样条曲线的求解方式，优化了中间计算过程，提高了求解效率。

经过以上的分析介绍，本文以高性能的 S 型速度曲线为基础，从运行速度和运行时间两方面考虑，研究了两种通用的自适应可控 S 型速度规划算法，实现等周期插补。上层的空间轨迹进行规划时，可以根据不同的需求选择合适的速度规划算法完成空间轨迹位置、速度和加速度的等周期插补。在笛卡尔空间中，研究了基于三次 B 样条的空间自由曲线，通过优化求解运算过程，应用到实际工程中，实现了空间独立轨迹-轨迹之间的平滑过渡。

## 1.3 论文的主要内容

### 1.3.1 研究内容

本课题以通用六自由度工业机器人为平台，主要研究了以下几方面内容：

- (1) 机器人运动学理论，主要包括机器人 D-H（Denavit 和 Hardenberg）建模、运动学正解和逆解。
- (2) 研究分析了几种速度控制曲线的模型及它们的优缺点。重点研究了 S 型速度控制曲线，在此理论基础上，设计了两种自适应可控 S 型速度规划器：基于给定运行速度的 S 型速度规划器和基于给定运行时间的 S 型速度规划器，有效地应用于轨迹规划中要求时间同步的场合。
- (3) 基于自适应可控 S 型速度规划器，研究了笛卡尔空间直线和空间圆弧的轨迹规划、基于三次 B 样条的空间自由曲线轨迹规划。以空间自由曲线（三次准均匀 B 样条曲线）作为过渡轨迹，实现了空间直线-直线、空间直线-圆弧和空间三次

非均匀 B 样条-三次非均匀 B 样条的平滑过渡。

### 1.3.2 论文结构

本文共分为 5 章：

第 1 章绪论，介绍了课题的研究背景和研究意义，以及国内外研究现状，确定研究方向。

第 2 章机器人运动学，分析介绍了机器人模型建立，设计了机器人运动学正解与逆解算法。

第 3 章速度规划算法研究与设计，研究了几种常见的速度曲线模型，着重对 S 型速度控制曲线进行了深入的研究。

第 4 章笛卡尔空间轨迹规划，着重研究了空间自由曲线轨迹规划，轨迹与轨迹之间过渡问题。

第 5 章轨迹规划算法仿真与分析，对研究结果进行仿真验证与分析。

## 2 机器人运动学

机器人运动学研究的主要是机器人末端执行器运动与各关节运动之间的关系。机器人由一系列的关节和连杆组成，用一个关节坐标系来表示一个转动关节，齐次矩阵可以表示坐标系之间的变换，主要包括坐标系的平移和旋转。在已知机器人机械结构的情况下，根据机器人运动学理论对各个关节和末端执行器建立合适的坐标系以及机器人的 D-H 参数模型，求解出机器人的正解与逆解，从而对机器人进行控制和轨迹规划。本章介绍了机器人运动学的数学理论；建立了机器人的 D-H 模型，在此基础上实现了运动学正逆解算法。

### 2.1 机器人空间模型描述

#### 2.1.1 机器人位姿

机器人的位姿是指末端执行器坐标系的位置和姿态，如果位姿和参考坐标系已知，那么就可以确定机器人的状态。末端执行器的位姿通常使用一个  $4 \times 4$  的齐次变换矩阵来表示，这样能够将运动、变换和映射与矩阵运算联系起来<sup>[38]</sup>。

假设坐标系  $A$  的原点与参考坐标系的原点重合， $A$  由三个相互垂直的向量  $s$ ， $n$ ， $a$  表示，通常这三个向量称为单位向量，分别表示法线向量、指向向量和接近向量，如图 2-1 所示。每一个向量都由它们所在参考坐标系中的三个分量表示。那么坐标系  $A$  可以由  $s$ ， $n$ ， $a$  写出，以矩阵的形式表示为：

$$A = \begin{bmatrix} s_x & n_x & a_x \\ s_y & n_y & a_y \\ s_z & n_z & a_z \end{bmatrix} \quad (2-1)$$

其中  $s_x^2 + s_y^2 + s_z^2 = 1$ ， $n_x^2 + n_y^2 + n_z^2 = 1$ ， $a_x^2 + a_y^2 + a_z^2 = 1$ 。

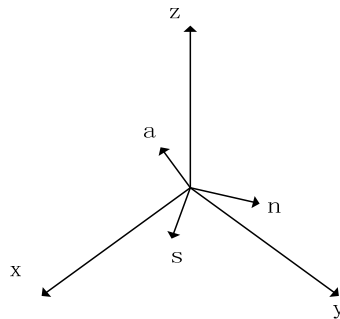


图 2-1 坐标系在参考坐标系原点的表示

Fig. 2-1 Coordinate system at the origin of the reference coordinate system

如图 2-2 所示，如果坐标系  $B$  的原点不与参考坐标系的原点重合，那么坐标系  $B$  的原点相对于参考坐标系的位置也必须表示出来，可以使用一个向量  $p = (p_x, p_y, p_z)$  表示该坐标系原点相对于参考坐标系原点的位置，向量  $p$  表示位置向量<sup>[39]</sup>。当坐标系  $B$  原点与参考坐标系原点重合时，此时向量  $p$  为零向量。这样，就可以使用三个表示方向的向量和一个表示位置的向量来描述坐标系  $B$ ，写成齐次矩阵的形式如下：

$$B = \begin{bmatrix} s_x & n_x & a_x & p_x \\ s_y & n_y & a_y & p_y \\ s_z & n_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2-2)$$

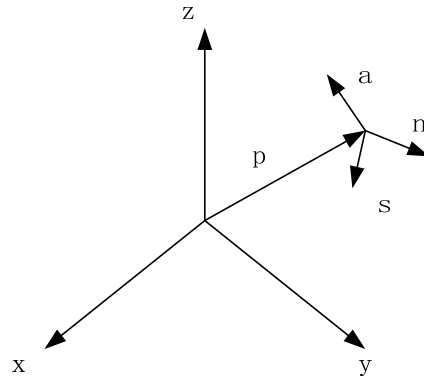


图 2-2 一个坐标系在另一个坐标系中的表示

Fig. 2-2 A coordinate system at another coordinate system

机器人的位姿就可以使用坐标系  $B$  来表示，记为：

$$T = \begin{bmatrix} s_x & n_x & a_x & p_x \\ s_y & n_y & a_y & p_y \\ s_z & n_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2-3)$$

其中机器人的位置矩阵表示为：

$$p = [p_x, p_y, p_z]^T \quad (2-4)$$

机器人的姿态矩阵表示为：

$$A = \begin{bmatrix} s_x & n_x & a_x \\ s_y & n_y & a_y \\ s_z & n_z & a_z \end{bmatrix} \quad (2-5)$$

### 2.1.2 坐标变换

坐标变换称为空间中的一个运动，坐标变换可以用来表示机器人位姿的变化。当空间中的一个坐标系相对于参考坐标系运动时，可以用坐标系变换的方式来表示这一运动，因为变换本身就是坐标系状态的变化，因此变换可以用坐标系来表示，坐标系的变换包括三种形式：纯平移、绕一个轴纯旋转和复合的变换<sup>[40]</sup>。

#### (1) 纯平移

如果一个坐标系以不变的姿态在空间中运动，那么该变换就是纯平移。改变的只是坐标系原点相对于参考坐标系原点的变化。在此情况下，坐标系的单位向量始终保持同一方向。如图 2-3 所示，经过平移后的坐标系相对于参考坐标系的位置可以用原来坐标系的原点位置向量加上位移向量求得。通常使用矩阵形式表示，新的坐标系通过左乘变换矩阵得到。变换矩阵  $T$  表示为：

$$T = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2-6)$$

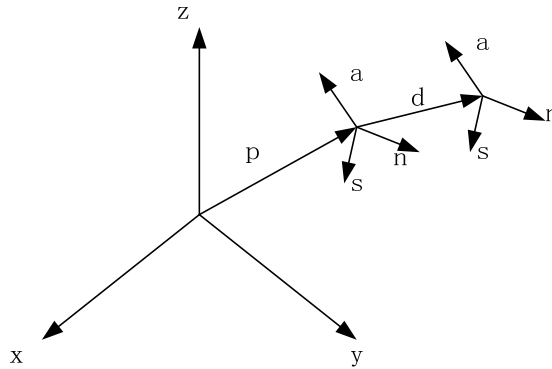


图 2-3 空间纯平移变换的表示

Fig. 2-3 Pure translation transform in space

其中， $d_x$ ， $d_y$ ， $d_z$ 是纯平移向量相对于参考坐标系  $x$ ， $y$  和  $z$  轴的三个分量。新的坐标系位置可以表示为：

$$\begin{aligned}
 B_{new} = T \times B &= \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & n_x & a_x & p_x \\ s_y & n_y & a_y & p_y \\ s_z & n_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} s_x & n_x & a_x & p_x + d_x \\ s_y & n_y & a_y & p_y + d_y \\ s_z & n_z & a_z & p_z + d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{2-7}$$

## (2) 绕一个轴纯旋转

绕任意轴的旋转都可以分解成绕  $x$  轴旋转、绕  $y$  轴旋转和绕  $z$  轴旋转的组合。绕  $x$  轴旋转，假设旋转坐标系位于参考坐标系的原点，旋转坐标系中点  $P$ ，如图 2-4 所示，为了得到在参考坐标系中的坐标，旋转坐标系中点  $P$  必须左乘旋转矩阵  $Rot(x, \theta)$ ，则有：

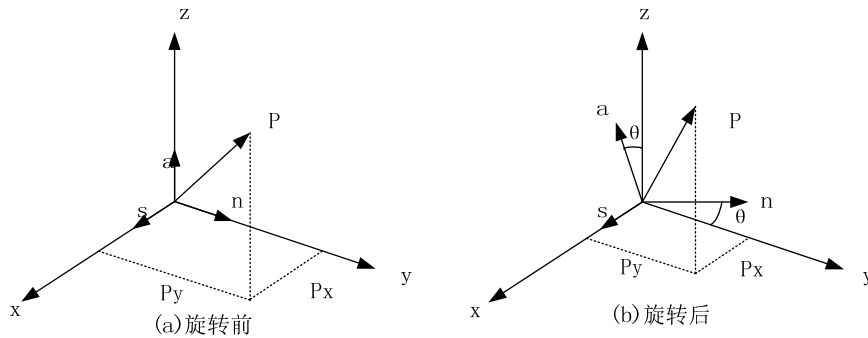


图 2-4 绕  $x$  轴旋转

Fig. 2-4 Rotation around the x axis

$$P_{xyz} = Rot(x, \theta) \times P_{sna} \tag{2-8}$$

$$Rot(x, \theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix} \tag{2-9}$$

其中  $P_{xyz}$  为旋转后相对于参考坐标系的位置， $P_{sna}$  为相对于旋转坐标系的位置（旋转前后不变）； $Rot(x, \theta)$  为绕  $x$  轴旋转的旋转矩阵，为了简化书写，用  $C\theta$  表示  $\cos \theta$ ，用  $S\theta$  表示  $\sin \theta$ 。

绕  $y$  轴、 $z$  轴的旋转与绕  $x$  轴旋转类似。分别写成齐次矩阵的形式，绕  $x$  轴纯旋转表示为：



$$Rot(x, \theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\theta & -S\theta & 0 \\ 0 & S\theta & C\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2-10)$$

绕  $y$  轴纯旋转表示为:

$$Rot(y, \theta) = \begin{bmatrix} C\theta & 0 & S\theta & 0 \\ 0 & 1 & 0 & 0 \\ -S\theta & 0 & C\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2-11)$$

绕  $z$  轴纯旋转表示为:

$$Rot(z, \theta) = \begin{bmatrix} C\theta & -S\theta & 0 & 0 \\ S\theta & C\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2-12)$$

### (3) 复合变换

复合变换是由参考坐标系或当前运动坐标系的一系列沿轴平移和绕轴旋转所组成, 是由坐标平移和坐标旋转共同作用的。任何复合变换都可以分解成平移和旋转按一定顺序的组合。

## 2.2 机器人正逆解

### 2.2.1 D-H 参数建模

D-H 参数建模是由 Denavit 和 Hartenberg 在 1955 年提出的一种机器人的表示和建模方法, 并推导出了运动方程, 这已成为机器人运动建模的标准方法<sup>[41]</sup>。串联机器人由一系列的连杆和关节组成, 一般包括  $n+1$  个连杆和  $n$  个关节, 连杆 0 表示机器人的底座, 连杆  $n$  表示机器人的末端执行器, 如所图 2-5 示。本文中以通用六自由度机器人为研究平台, 那么  $n=6$ 。

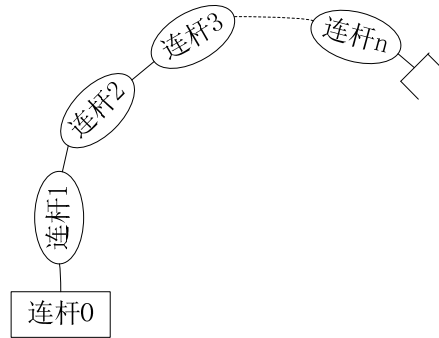


图 2-5 串联机器人结构

Fig. 2-5 Structure of the robot series

连杆之间通过关节连接，连杆  $i-1$  与连杆  $i$  之间通过关节  $i$  连接，为了很好的描述他们之间的关系，在关节  $i$  处建立坐标系  $R_i$ ，那么相邻关节之间的坐标系变换关系如所图 2-6 示。变换矩阵为：

$${}^{i-1}T_i = Rot(x, \alpha_i) Trans(x, a_i) Rot(z, \theta_i) Trans(z, d_i)$$

$$= \begin{bmatrix} C\theta_i & -S\theta_i & 0 & a_i \\ C\alpha_i S\theta_i & C\alpha_i C\theta_i & -S\alpha_i & -d_i S\alpha_i \\ S\alpha_i S\theta_i & S\alpha_i C\theta_i & C\alpha_i & d_i C\alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2-13)$$

其中上式中参数含义： $i$  为关节编号，从 1 开始； $\alpha_i$  为以  $x_{i-1}$  为轴， $z_{i-1}$  旋转到  $z_i$  的角度； $a_j$  为沿  $x_{i-1}$  移动， $z_{i-1}$  到  $z_i$  的距离； $\theta_j$  为以  $z_i$  为轴， $x_{i-1}$  旋转到  $x_i$  的角度； $d_i$  为沿  $z_i$  移动， $x_{i-1}$  到  $x_i$  的距离。

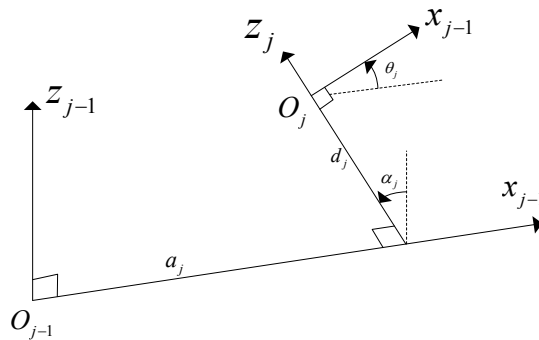


图 2-6 关节坐标系变换

Fig. 2-6 Joint coordinate transformation

根据 D-H 建模方法，建立机器人模型如图 2-7 所示。

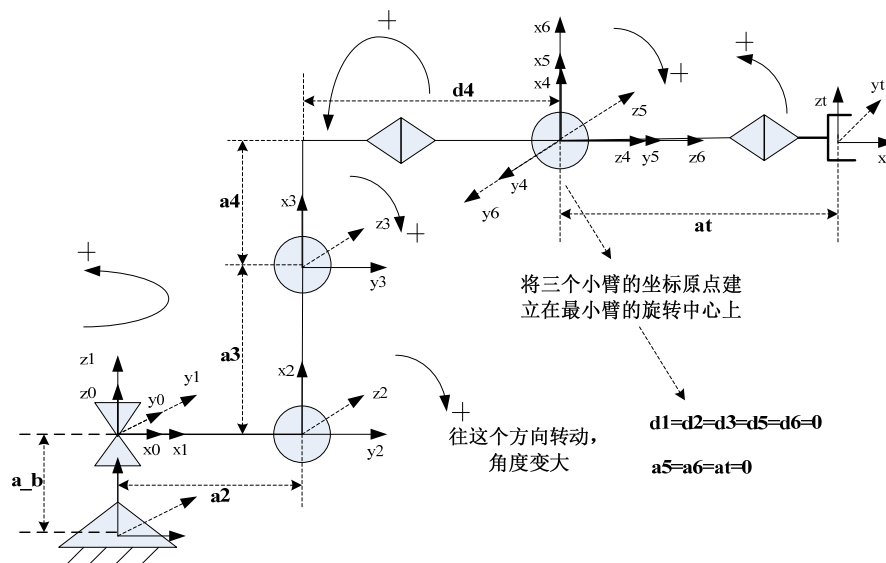


图 2-7 机器人的 D-H 参数模型

Fig. 2-7 D-H parameters model of the robot

D-H 参数表如所示：

表 2-1 机器人 D-H 参数表

Table. 2-1 Robot D-H parameter table

关节 $i$	$a_i(\text{mm})$	$\alpha_i(^{\circ})$	$\theta_i(^{\circ})$	$d_i(\text{mm})$	关节 $\theta_i$ 变量范围( $^{\circ}$ )
1	$a_1 = 0$	$\alpha_1 = 0$	$\theta_1(0)$	$d_1 = 0$	-170—170
2	$a_2 = 85$	$\alpha_2 = -90$	$\theta_2(-90)$	$d_2 = 0$	-92—135
3	$a_3 = 380$	$\alpha_3 = 0$	$\theta_3(0)$	$d_3 = 0$	-129—116
4	$a_4 = 100$	$\alpha_4 = -90$	$\theta_4(0)$	$d_4 = 425$	-160—160
5	$a_5 = 0$	$\alpha_5 = 90$	$\theta_5(0)$	$d_5 = 0$	-120—120
6	$a_6 = 0$	$\alpha_6 = -90$	$\theta_6(0)$	$d_6 = 0$	-360—360
$t$	$a_t = 85$	0	0	0	0

### 2.2.2 机器人正解

已知各个关节角度，求解末端位姿，这个过程称为机器人正解。在此六自由度机器人模型中，则体现为求解工具坐标系的位姿，即  ${}^0T = {}^0T_1T_2T_3T_4T_5T_6T_t$ ，由表 2-1 中的参数，依次计算相邻两关节间的变换矩阵，这些传递矩阵是以  $\theta_i$  为变量的函数，如下所示：

$${}^0_1T = \begin{bmatrix} s_1 & -s_1 & 0 & a_1 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2-14)$$

$${}^1_2T = \begin{bmatrix} c_2 & -s_2 & 0 & a_2 \\ 0 & 0 & 1 & 0 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2-15)$$

$${}^2_3T = \begin{bmatrix} c_3 & -s_3 & 0 & a_3 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2-16)$$

$${}^3_4T = \begin{bmatrix} c_4 & -s_4 & 0 & a_4 \\ 0 & 0 & 1 & d_4 \\ -s_4 & -c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2-17)$$

$${}^4_5T = \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2-18)$$

$${}^5_6T = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_6 & -c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2-19)$$

$${}^6_tT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & a_t \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2-20)$$

为了书写方便，在以上各式中使  $c_i$  和  $s_i$  代表  $\cos \theta_i$  和  $\sin \theta_i$ ，最终可求得末端工具坐标系位姿在参考坐标系中的位姿矩阵，如下所示：

$${}^0T_i = \begin{bmatrix} s_x & n_x & a_x & p_x \\ s_y & n_y & a_y & p_y \\ s_z & n_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2-21)$$

其中：

$$\begin{aligned} s_x &= c_1((c_2c_3 - s_2s_3)(c_4c_5c_6 - s_4s_5) - (c_2c_3 + s_2c_3)s_5c_6) \\ &\quad + s_1(s_4c_5c_6 + c_4s_6) \\ s_y &= s_1((c_2c_3 - s_2s_3)(c_4c_5c_6 - s_4s_6) - (c_2c_3 + s_2c_3)s_5c_6) \\ &\quad - c_1(s_4c_5c_6 + c_4s_6) \\ s_z &= -(s_2c_3 + c_2c_3)(c_4c_5c_6 - s_4s_6) + (s_2s_3 - c_2c_3)s_5s_6 \\ n_x &= -c_1(c_2c_3 - s_2s_3)(c_4c_5c_6 + s_4c_6) \\ &\quad + c_1(c_2s_3 + s_2c_3)s_5s_6 - s_1(s_4c_5s_6 - c_4c_6) \\ n_y &= s_1((-c_2c_3 + s_2s_3)(c_4c_5c_6 + s_4c_6) \\ &\quad + (c_2s_3 + s_2c_3)s_5s_6) + c_1(s_4c_5s_6 - c_4c_6) \\ n_z &= (s_2c_3 + c_2c_3)(c_4c_5c_6 + s_4s_6) - (s_2s_3 - c_2c_3)s_5s_6 \\ a_x &= -c_1(c_4s_5(c_2c_3 - s_2s_3) + (c_2c_3 + s_2s_3)c_5) - s_1s_4s_5 \\ a_y &= -s_1(c_4s_5(c_2c_3 - s_2s_3) + (c_2c_3 + s_2s_3)c_5) - c_1s_4s_5 \\ a_z &= (s_2c_3 + c_2s_3)c_4s_5 + c_5(s_2s_3 - c_2s_3) \\ p_x &= c_1((c_2c_3 - s_2s_3)a_4 - d_4(c_2s_3 + s_2c_3) + c_2a_3 + a_2) \\ &\quad - a_1(c_2(c_4s_5(c_2c_3 - s_2s_3) + (c_2s_3 + s_2c_3)c_5) + s_1s_4s_5) \\ p_y &= s_1(a_4(c_2c_3 - s_2s_3) - d_4(c_2s_3 + s_2c_3) + c_2a_3 + a_2) \\ &\quad - a_1(s_1(c_4s_5(c_2c_3 - s_2s_3) + (c_2s_3 + s_2c_3)c_5) - c_1s_4s_5) \\ p_z &= -a_4(s_2c_3 + c_2s_3) + d_4(s_2s_3 - c_2c_3) - a_3s_2 \\ &\quad + a_1((s_2c_3 + c_2s_3)c_4s_5 + c_5(s_2s_3 - c_2c_3)) \end{aligned}$$

### 2.2.3 机器人逆解

由上一节可知，已知机器人各个关节的角度，通过机器人正解，我们可以得到末端工具坐标系相对于参考坐标系的位姿。机器人逆解是指在给定末端工具坐标系位姿下，求解各个关节角度的过程。为了使机器人末端执行器位姿运动可控，在机器人运动过程中，就必须不断进行逆解，求解出各个关节转动角度。

由  ${}^0T = {}^0T_1{}^1T_2{}^2T_3{}^3T_4{}^4T_5{}^5T_6{}^6T$ ，其中  ${}^6T$  已知，从而可得到：

$${}^0_6T = {}^0_1T {}^1_2T {}^2_3T {}^3_4T {}^4_5T {}^5_6T = {}^0_tT * ({}_t^6T)^{-1} = \begin{bmatrix} s_x & n_x & a_x & p_x \\ s_y & n_y & a_y & p_y \\ s_z & n_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2-22)$$

根据公式 (2-22) 可以求解 6 个关节角度  $\theta_1 \sim \theta_6$ 。

(1) 求解前三个关节角度  $\theta_1$ ,  $\theta_2$  和  $\theta_3$

由于六自由度机械臂的位置是由前三个关节决定, 位姿是由 6 个关节共同决定。可知:

$${}^0_6T \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = {}^0_4T \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (2-23)$$

对公式 (2-23) 两端同时依次左乘矩阵  $({}^0_1T)^{-1}$ , 可知:

$$\begin{bmatrix} c_1p_x + s_1p_y \\ c_1p_y - s_1p_x \\ p_z \\ 1 \end{bmatrix} = \begin{bmatrix} a_2 + a_3c_2 + d_4(-c_3s_2 - c_2s_3) + a_4(c_2c_3 - s_2s_3) \\ 0 \\ -a_2s_2 + a_4(-c_3s_2 - c_2s_3) + d_4(-c_2c_3 + s_2s_3) \\ 1 \end{bmatrix} \quad (2-24)$$

由公式 (2-24) 左右两端的第二行元素相等, 可求得  $\theta_1$ :

$$\theta_1 = \begin{cases} a \tan 2(p_y, p_x) \\ a \tan 2(-p_y, -p_x) \end{cases} \quad (2-25)$$

对公式 (2-24) 两端同时依次左乘矩阵  $({}_2^1T)^{-1}$ , 将可得到以下表达式:

$$\begin{bmatrix} -a_2c_2 + c_1c_2p_x + c_2p_ys_1 - p_zs_2 \\ -c_2p_z + a_2s_2 - c_1p_xs_2 - p_ys_1s_2 \\ c_1p_y - p_xs_1 \\ 1 \end{bmatrix} = \begin{bmatrix} a_3 + a_4c_3 - d_4s_3 \\ c_3d_4 + a_4s_3 \\ 0 \\ 1 \end{bmatrix} \quad (2-26)$$

对公式 (2-26) 的两端前两行元素分别平方求和, 计算可得到:

$m = [a_2 - (c_1p_x + s_1p_y)]c_2 - (-p_z)s_2$ , 令  $u = [a_2 - (c_1p_x + s_1p_y)]$ ,  $v = -p_z$ , 其中

$m = \frac{a_4^2 + d_4^2 - a_3^2 - p_z^2 - [a_2 - (c_1p_x + s_1p_y)]^2}{2a_3}$ , 那么可得:

$$u * \cos \theta_2 - v * \sin \theta_2 = m \Rightarrow \frac{u}{\sqrt{u^2 + v^2}} * \cos \theta_2 + \frac{-v}{\sqrt{u^2 + v^2}} * \sin \theta_2 = \frac{m}{\sqrt{u^2 + v^2}}$$

令  $\alpha = a \tan 2(-v, u)$ ，则上式成为  $\cos(\theta_2 - \alpha) = \frac{m}{\sqrt{u^2 + v^2}}$

进一步采用图解法，如图 2-8 所示，可得： $\theta_2 - \alpha = a \tan 2\left(\pm\sqrt{u^2 + v^2 - m^2}, m\right)$

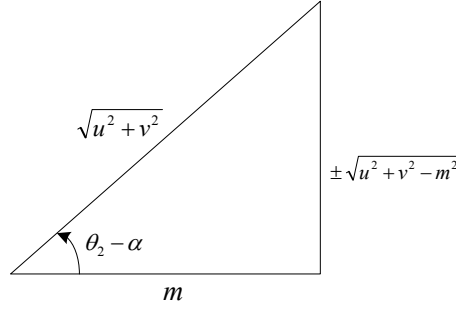


图 2-8 图解法求角度  $\theta_2$

Fig. 2-8 Angle  $\theta_2$  solution by graphic

进而可求得：

$$\theta_2 = a \tan 2(-v, u) \pm a \tan 2\left(\sqrt{u^2 + v^2 - m^2}, m\right) \quad (2-27)$$

由公式 (2-26) 前两行相等可得到方程组：

$$\begin{cases} a_4 c_3 - d_4 s_3 = A \\ a_4 s_3 + d_4 c_3 = B \end{cases} \quad (2-28)$$

其中  $A = -a_3 - a_2 c_2 + c_1 c_2 p_x + c_2 p_y s_1 - p_z s_2$ ， $B = -c_2 p_z + a_2 s_2 - c_1 p_x s_2 - p_y s_1 s_2$ ，由

公式 (2-28) 可得：

$$\begin{cases} s_3 = \frac{Ba_4 - Ad_4}{a_4^2 + d_4^2} \\ c_3 = \frac{Aa_4 + Bd_4}{a_4^2 + d_4^2} \end{cases} \quad (2-29)$$

进而求得  $\theta_3$ ：

$$\theta_3 = a \tan 2(s_3, c_3) \quad (2-30)$$

(2) 求解后三个关节角度  $\theta_4$ ， $\theta_5$  和  $\theta_6$

在上一步中，求得了前三个关节角  $\theta_1$ ， $\theta_2$  和  $\theta_3$ ，那么矩阵  ${}^0_3T$  已知，由  ${}^0_6T = {}^0_3T {}^3_6T$  两边分别乘以  ${}^0_3T$  的逆矩阵可得：

$${}^3_6T = ({}^0_3T)^{-1} ({}^0_6T) \quad (2-31)$$

公式 (2-31) 左边  ${}^3_6T = {}^3_4T {}^4_5T {}^5_6T$ ，写成：

$${}^3_6T = \begin{bmatrix} c_4c_5c_6 - s_4s_6 & -c_4c_5s_6 - s_4c_6 & -c_4s_5 & a_4 \\ s_5c_6 & -s_5s_6 & c_5 & d_4 \\ -s_4c_4c_6 - c_4s_6 & s_4c_5s_6 - c_4c_6 & s_4s_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2-32)$$

右边的结果写成：

$$({}^0_3T)^{-1} ({}^0_6T) = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ r_{41} & r_{42} & r_{43} & r_{44} \end{bmatrix} \quad (2-33)$$

其中：

$$\begin{aligned} r_{11} &= (c_1c_2c_3 - c_1s_2s_3)s_x + (s_1c_2c_3 - s_1s_2s_3)s_y - (s_2c_3 + c_2s_3)s_z \\ r_{12} &= (c_1c_2c_3 - c_1s_2s_3)n_x + (s_1c_2c_3 - s_1s_2s_3)n_y - (s_2c_3 + c_2s_3)n_z \\ r_{13} &= (c_1c_2c_3 - c_1s_2s_3)a_x + (s_1c_2c_3 - s_1s_2s_3)a_y - (s_2c_3 + c_2s_3)a_z \\ r_{21} &= (-c_1c_2s_3 - c_1s_2c_3)s_x - (s_1c_2s_3 + s_1s_2c_3)s_y + (s_2s_3 - c_2c_3)s_z \\ r_{22} &= (-c_1c_2s_3 - c_1s_2c_3)n_x - (s_1c_2s_3 + s_1s_2c_3)n_y + (s_2s_3 - c_2c_3)n_z \\ r_{23} &= (-c_1c_2s_3 - c_1s_2c_3)a_x - (s_1c_2s_3 + s_1s_2c_3)a_y + (s_2s_3 - c_2c_3)a_z \\ r_{33} &= -s_1p_x + c_1p_y \end{aligned}$$

由公式 (2-32) 和公式 (2-33) 的第二行第三列元素相等，即  $c_5 = r_{23}$ ，如图 2-9 所示，采用图解法解得  $\theta_5$ ：

$$\theta_5 = a \tan 2\left(\pm\sqrt{1-r_{23}^2}, r_{23}\right) \quad (2-34)$$

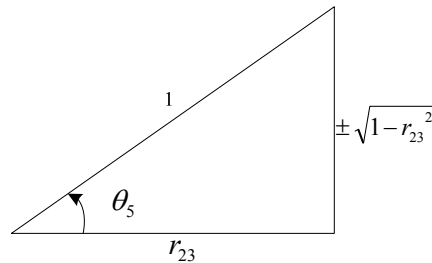


图 2-9 图解法求角度  $\theta_5$

Fig. 2-9 Angle  $\theta_5$  solution by graphic



在  $\theta_5$  已知的基础上,可以继续根据公式(2-32)和公式(2-33)来求解  $\theta_4$  和  $\theta_6$ ,此时要分两种情况进行讨论。

情况一:  $s_5 \neq 0$

当  $s_5 \neq 0$  (即  $\theta_5 \neq 0$ )时,由公式(2-32)和公式(2-33)第一行第三列和第三行第三列元素相等,即:

$$\begin{cases} -c_4 s_5 = r_{13} \\ s_4 s_5 = r_{33} \end{cases} \quad (2-35)$$

由公式(2-35)可解得:

$$\theta_4 = a \tan 2(r_{33}s_5, -r_{13}s_5) \quad (2-36)$$

由公式(2-32)和公式(2-33)第二行第一列和第二行第二列元素相等,即:

$$\begin{cases} s_5 c_6 = r_{21} \\ -s_5 s_6 = r_{22} \end{cases} \quad (2-37)$$

由公式(2-37)可解得:

$$\theta_6 = a \tan 2(-r_{22}s_5, r_{21}s_5) \quad (2-38)$$

情况二:  $s_5 = 0$

当  $s_5 = 0$  (即  $\theta_5 = 0$ )时,由公式(2-32)和公式(2-33)第一行第一列和第一行第二列元素相等,即:

$$\begin{cases} c_4 c_5 c_6 - s_4 s_6 = r_{11} \\ -c_4 c_5 s_6 - s_4 c_6 = r_{12} \end{cases} \quad (2-39)$$

由公式(2-39)可解得:

$$\theta_4 + \theta_6 = a \tan 2(-r_{12}, r_{11}) \quad (2-40)$$

从公式(2-40)可以看出,此时  $\theta_4$  和  $\theta_6$  有无穷多组解,机器人处于奇异位型,连杆 5 和连杆 6 成一直线,实际使用时,在保持  $(\theta_4 + \theta_6)$  不变的前提下,令  $\theta_4$  与  $\theta_6$  朝相反方向转动相同大小的角度,从而避免奇异位型。

进行逆解时,总共存在 8 组解(不发生奇异时),一般通过定义如下的性能函数:

$$f = \sum_{i=1}^6 (\mu_i \Delta \theta_i)^2, \quad 0 \leq \mu_i \leq 1 \quad (2-41)$$

上式中  $\mu_i$  为关节运动影响因子,根据机器人结构设定,选择一组解,使上式最小,能够满足机器人运动过程中能量最优。

## 2.3 本章小结

本章主要分析了机器人运动学相关的理论基础和机器人建模方法与正逆解的推导。机器人末端工具的空间状态使用位姿矩阵来描述，根据机器人机械结构信息，使用 D-H 参数建模方法对六自由度串联机器人建立精确的数学模型。对每个关节建立坐标系，通过坐标系之间的变换，推导出了机器人正解公式（末端位姿矩阵关于六个关节角度的表达式）。机器人逆解的推导与分析，给定机器人末端工具的位姿，推导出了六个关节角求解公式，并分析了逆解推导过程中的关键点。

### 3 速度规划算法研究与设计

为了保证机器人在启动、运行以及停止时不对整个机械系统产生冲击或振荡，这样就必须设计特定的加减速控制策略，控制输入到电机的电压按照这个规律变化，使机器人末端工具在运动中能具有快速性，并且能准确地运行到给定的位姿上，这种控制方法称为加减速控制<sup>[42]</sup>。本章介绍了工业运动控制系统中常用的速度曲线：梯形速度曲线、正弦型速度曲线和 S 型速度曲线<sup>[43-47]</sup>，并分析了它们的优缺点，最终选择 S 型速度曲线作为重点研究对象，并研究和设计了两种自适应可控 S 型速度规划器：基于给定运行速度的 S 型速度规划器和基于给定运行时间的 S 型速度规划器。

#### 3.1 速度控制曲线

##### 3.1.1 梯形速度曲线

梯形速度曲线是工业自动化控制领域应用较多的速度控制方法之一<sup>[48, 49]</sup>，一般多用于低端的控制领域。梯形速度曲线在变速过程中，加速度为一常数，用户可以根据机器人的性能设定加速度的值，假设已知两个示教点的位置，示教点之间的运行速度  $v$ ，梯形速度曲线的加速度  $a$ ，示教点之间的总运行时间  $T$ ，可以得到标准的梯形速度曲线，如图 3-1 所示。

图中曲线分为三段：匀加速段、匀速段和匀减速段，因为匀加速段和匀减速段的加速度  $a$  不变，所以这两段运行时间相等，变速时间  $t_1 = v/a$ ，所以可以很容易计算出梯形速度曲线的速度和位移表达式，如下：

$$v(t) = \begin{cases} at, & 0 < t < t_1 \\ v, & t_1 \leq t < t_2 \\ v - at, & t_2 \leq t < T \end{cases} \quad (3-1)$$

$$L(t) = \begin{cases} 0.5at^2, & 0 \leq t < t_1 \\ 0.5\frac{v^2}{a} + v(t - t_1), & t_1 \leq t < t_2 \\ 0.5\frac{v^2}{a} + v(T - 2t_1) + 0.5a(t - t_2)^2, & t_2 \leq t \leq T \end{cases} \quad (3-2)$$

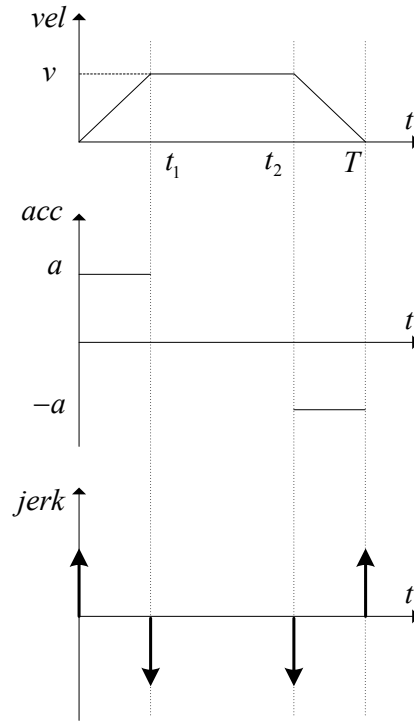


图 3-1 梯形速度曲线

Fig. 3-1 Trapezoidal speed curve

由图中可以看出，在加速度突变处，产生脉冲形式的冲击，给机器人刚体带来冲击。由此可见，梯形速度控制曲线的优点是算法实现容易、效率高效，缺点是速度曲线过渡不平滑、加速度不连续运动精度低且存在刚体冲击。

### 3.1.2 正弦型速度曲线

如图 3-2 所示，正弦型速度曲线与梯形速度曲线相比，速度的控制段数相同，也分为三段，分别为正弦加速度段、匀速段和正弦减速段。其加速度表现为三角函数形式，加速度的导数加加速度（冲击）也表现为三角函数形式。设正弦加速度段的加速度表达式为：

$$a = a_{\max} \sin(kt) \quad (3-3)$$

其中  $k$  为正弦函数的系数，那么可以求出正弦加速度段  $t_1$  为：

$$t_1 = \pi / k \quad (3-4)$$

运行速度  $v$  的求解可对加速度积分得到：

$$v = \int_0^{t_1} a_{\max} \sin(kt) \quad (3-5)$$

由公式 (3-3)、公式 (3-4) 和公式 (3-5) 可求解出：

$$k = \frac{2a_{\max}}{v} \quad (3-6)$$

最终可以推导出正弦型速度曲线的速度和位移的表达式为：

$$v(t) = \begin{cases} \frac{v}{2} \left(1 - \cos \frac{2a_{\max}}{v} t\right), & 0 \leq t < t_1 \\ v, & t_1 \leq t < t_2 \\ \frac{v}{2} - \frac{v}{2} \cos \frac{2a_{\max}}{v} (t - T), & t_2 \leq t \leq T \end{cases} \quad (3-7)$$

$$L(t) = \begin{cases} \frac{v^2}{4a_{\max}} \left(\pi - \sin \frac{2a_{\max}}{v} t\right), & 0 \leq t < t_1 \\ vt - \frac{\pi v^2}{4a_{\max}}, & t_1 \leq t < t_2 \\ \left[\frac{v}{2} t - \frac{v^2}{4a_{\max}} \sin \frac{2a_{\max}}{v} (t - T)\right] + \left(\frac{vT}{2} - \frac{\pi v^2}{4a_{\max}}\right), & t_2 \leq t \leq T \end{cases} \quad (3-8)$$

其中  $T$  为运行总时间。

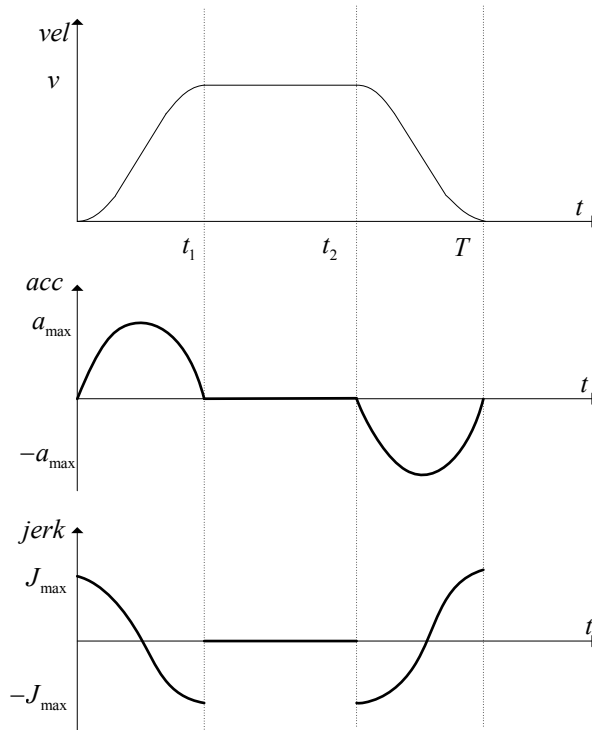


图 3-2 正弦型加速度曲线

Fig. 3-2 Sinusoidal acceleration curve

正弦型速度曲线的计算方式较复杂，但其加减速策略可以实现刚体的平滑运动，一般用在高精度机器人加工作业中<sup>[50]</sup>。

### 3.1.3 标准 S 型速度曲线

标准 S 型速度曲线几乎与正弦型速度曲线的形状一样<sup>[51, 52]</sup>，区别在于其加速度呈梯形形状，一般分成七段加减速处理，加速度段由加加速度段、匀加速段和减加速度段组成；减速度段由加减速段、匀减速段和减减速度段组成；中间为匀速段<sup>[53-55]</sup>。标准 S 型速度曲线相比正弦型速度曲线最大的点在于，标准 S 型速度曲线可以控制加速度的变化率（加加速度），用户可以根据系统的性能设定加速度的变化率，从而在更短的时间内加速到期望运行的速度，提高了机器人的作业效率。

S 型速度曲线参数如图 3-3 所示，初始速度为  $v_s$ ，末速度为  $v_e$ ，匀速段运行速度为  $v$ ，最大加速度为  $a_{\max}$ ，系统能够承受的最大冲击为  $J_{\max}$ ，为了推导简便，在这里假设初始速度等于末速度，并且为零，即  $v_s=v_e=0$ ，利用位移、速度、加速度和加加速度之间的关系，分别推导出加速度、速度和位移的表达式如下：

$$a(t) = \begin{cases} J_{\max} t, & 0 \leq t < t_1 \\ a_{\max}, & t_1 \leq t < t_2 \\ a_{\max} - J_{\max} (t - t_2), & t_2 \leq t < t_3 \\ 0, & t_3 \leq t < t_4 \\ -J_{\max} (t - t_4), & t_4 \leq t < t_5 \\ -a_{\max}, & t_5 \leq t < t_6 \\ -a_{\max} + J_{\max} (t - t_6), & t_6 \leq t < t_7 \end{cases} \quad (3-9)$$

$$v(t) = \begin{cases} \frac{1}{2} J_{\max} t^2, & 0 \leq t < t_1 \\ v_1 + a_{\max} (t - t_1), & t_1 \leq t < t_2 \\ v_2 + a_{\max} (t - t_2) - \frac{1}{2} J_{\max} (t - t_2)^2, & t_2 \leq t < t_3 \\ v_3, & t_3 \leq t < t_4 \\ v_4 - \frac{1}{2} J_{\max} (t - t_4)^2, & t_4 \leq t < t_5 \\ v_5 + a_{\max} (t - t_5), & t_5 \leq t < t_6 \\ v_6 - a_{\max} (t - t_6) + \frac{1}{2} J_{\max} (t - t_6)^2, & t_6 \leq t < t_7 \end{cases} \quad (3-10)$$

$$L(t) = \begin{cases} \frac{1}{6} J_{\max} t^3, & 0 \leq t < t_1 \\ L_1 + v_1(t-t_1) + \frac{1}{2} a_{\max}(t-t_1)^2, & t_1 \leq t < t_2 \\ L_2 + v_2(t-t_2) + \frac{1}{2} a_{\max}(t-t_2)^2 - \frac{1}{6} J_{\max}(t-t_2)^3, & t_2 \leq t < t_3 \\ L_3 + v_3(t-t_3), & t_3 \leq t < t_4 \\ L_4 + v_4(t-t_4) - \frac{1}{6} J_{\max}(t-t_4)^3, & t_4 \leq t < t_5 \\ L_5 + v_5(t-t_5) - \frac{1}{2} a_{\max}(t-t_5)^2, & t_5 \leq t < t_6 \\ L_6 + v_6(t-t_6) - \frac{1}{2} a_{\max}(t-t_6)^2 + \frac{1}{6} J_{\max}(t-t_6)^3, & t_6 \leq t < t_7 \end{cases} \quad (3-11)$$

其中  $v_1$ 、 $v_2$ 、 $v_3$ 、 $v_4$ 、 $v_5$ 、 $v_6$  和  $L_1$ 、 $L_2$ 、 $L_3$ 、 $L_4$ 、 $L_5$ 、 $L_6$  分别为  $t_1$ 、 $t_2$ 、 $t_3$ 、 $t_4$ 、 $t_5$ 、 $t_6$  时刻对应的速度和位移。

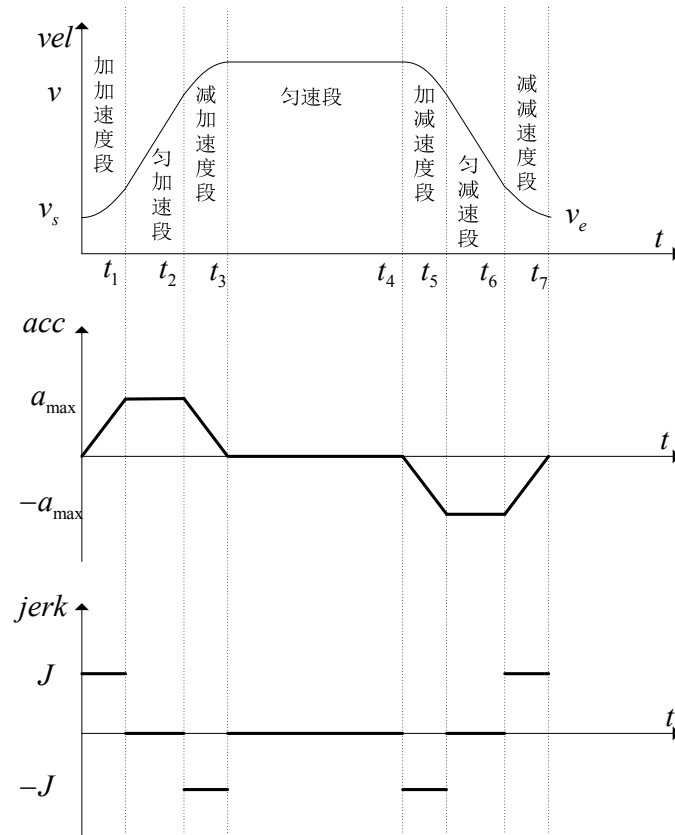


图 3-3 标准 S 型速度曲线

Fig. 3-3 Standard S-curve velocity profile

S 型速度曲线上每一点都是加速度连续的,从而避免了刚体运动的柔性冲击,且速度平滑性很好,运动精度高,由于其加速度变化率可控制的优点,充分发挥了机器人的性能。常用于高速、高精度工业控制中。所以本文以高性能的 S 型速度曲线为研究对象,服务于上层的轨迹规划算法。

### 3.2 自适应可控 S 型速度规划器

本文设计的 S 型速度规划器将机器人末端执行器的运行轨迹抽象成一条虚拟的直线,对于用户来说, S 型速度规划器就类似于一个黑匣子,用户只需要给定一组输入参数,经过 S 型速度规划器进行速度规划后,通过特定的接口输出运行周期  $T$ , 以及位移、速度和加速度关于时间的函数  $L(t)$ 、 $v(t)$  和  $a(t)$ 。S 型速度规划器作为底层接口被上层用户所调用,这里的用户特指轨迹规划算法,输入参数有位移、初始速度和末速度、运行速度、运行时间、加速度和加加速度等。用户可根据需求自行控制输入参数,考虑到初始速度和末速度的特殊性,初始速度由机器人末端执行器上一段运行轨迹的末速度所决定,不能改变,而当前这段轨迹末速度的选择相比就更具有弹性。在进行速度规划时,为了满足 S 型速度曲线的特性,规划器可以弹性地自适应选择合适的末速度,并把规划结果返回给用户,是否采用由用户决定。本节研究和设计了两种自适应可控 S 型速度规划器:基于给定运行速度的自适应可控 S 型速度规划器和基于给定运行时间的自适应可控 S 型速度规划器,其工作原理如图 3-4 所示。

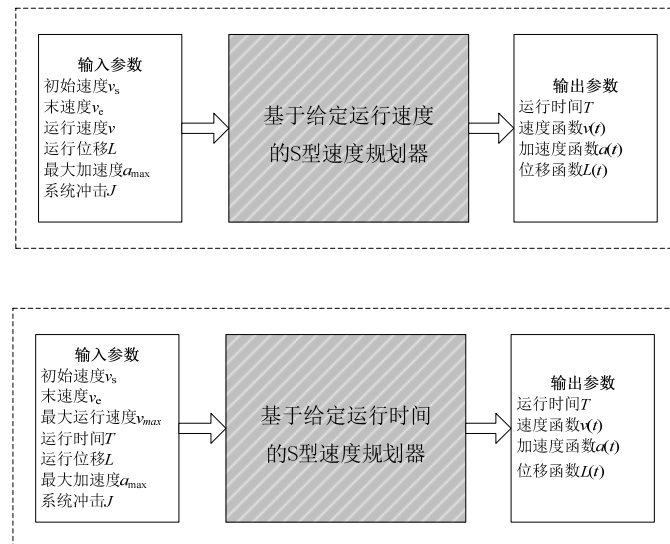


图 3-4 自适应可控 S 型速度规划器工作原理

Fig. 3-4 Working principle of adaptive and controllable S-type velocity planner



### 3.2.1 S 型速度曲线的特殊处理

上一节中, 介绍了标准 S 型速度曲线的特点, 但是在实际应用中, 由于机器人末端执行器在不同的作业场合要求不同的运行速度。当空间轨迹路程长度  $L$  过短, 没有达到给定运行速度的要求, 即不存在匀速段; 又初始速度或末速度与期望的运行速度差值太小, 导致了缺少匀加速段或匀减速段; 更有一些极端情况, 如空间轨迹路程长度  $L$  过短, 初始速度、末速度和运行速度过大, 导致不存在变速段, 此时不能求解出一个 S 型速度规划的解。类似这类情况需要对速度进行重新规划或选取合适的速度规划算法。本文在进行 S 型速度规划时, 没有严格进行分段处理, 而是把速度有变化的时间段进行统一处理, 统称为变速段, 并进行变速-匀速-变速、变速-变速, 甚至只有匀速段或变速段的分段自适应可控 S 型速度规划。

假设变速前的速度为  $v_1$ , 变速后的速度为  $v_2$ , 系统能提供的最大加速度为  $a_{\max}$ , 系统所能承受的最大加加速度为  $J$ 。如果变速前后速度相等, 即  $v_1=v_2$ , 此段规划中加速度和加加速度都为 0, 称为匀速规划。当变速前后速度不相等时, 如图 3-5 所示, 分为两类情况处理: 变速前速度小于变速后速度和变速前速度大于变速后速度。每一类有三种状态: 梯形加速、三角形临界加速和三角形欠加速状态。

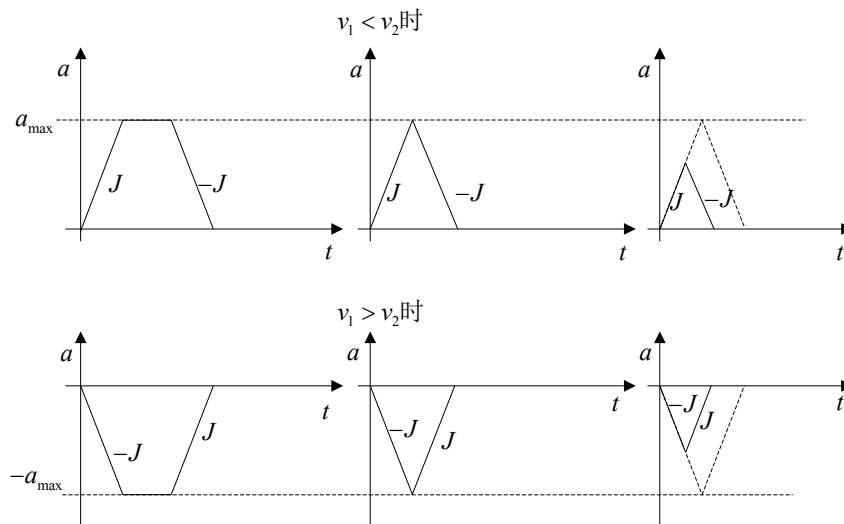


图 3-5 变速类型

Fig. 3-5 Transmission Classification

首先分别求取三种状态下的变速时间, 然后在此基础上推导出加速度、速度和位移关于变速时间的表达式, 以  $t_r$  表示变速段的总运行时间,  $t_{hr}$  表示变速段的匀加速段时间。

### (1) 三角形临界加速状态的变速时间

根据图 3-5 中图所示，三角形底边长度为三角形临界加速状态下总运行时间，写成：

$$t_{thr} = 2 \frac{a_{\max}}{J} \quad (3-12)$$

在三角形临界加速状态下  $t_r = t_{thr}$ ，因为不存在匀加速段，所以  $t_{hr} = 0$ 。

### (2) 三角形欠加速状态的变速时间

欠加速状态下，三角形的面积表示变速前后速度变化大小，记为：

$$|v_1 - v_2| = \frac{1}{2} t_r \times (J \frac{t_r}{2}) \quad (3-13)$$

由公式 (3-14) 可计算出变速段的总运行时间：

$$t_r = 2 \sqrt{\frac{|v_1 - v_2|}{J}} \quad (3-14)$$

在此状态下也不存在匀加速段，所以  $t_{hr} = 0$ 。

### (3) 梯形加速状态下的变速时间

梯形加速状态下的变速总时间等于三角形临界加速状态的变速总时间加上匀加速的时间。假设三角形临界加速状态下的速度变化大小为  $|\Delta v_{thr}|$ ，其计算过程如下所示：

$$|\Delta v_{thr}| = 2 \int_0^T a(t) dt = 2 \int_0^T J t dt = \frac{a_{\max}^2}{J} \quad (3-15)$$

匀加速时间  $t_{hr}$  的求解过程如下所示：

$$t_{hr} = \frac{|v_1 - v_2| - |\Delta v_{thr}|}{a_{\max}} \quad (3-16)$$

所以变速段总的运行时间为：

$$t_r = t_{thr} + t_{hr} = \frac{a_{\max}}{J} + \frac{|v_1 - v_2|}{a_{\max}} \quad (3-17)$$

### (4) 加速度、速度和位移

有了各个状态下的变速时间，进而求得加速度的表达式：

$$a(t) = \begin{cases} \operatorname{sgn}(v_2 - v_1)Jt, & 0 \leq t \leq \frac{(t_r - t_{hr})}{2} \\ \frac{1}{2}\operatorname{sgn}(v_2 - v_1)J(t_r - t_{hr}), & \frac{(t_r - t_{hr})}{2} < t \leq \frac{(t_r + t_{hr})}{2} \\ \operatorname{sgn}(v_2 - v_1)J(t_r - t), & \frac{(t_r + t_{hr})}{2} < t \leq t_r \end{cases} \quad (3-18)$$

对加速度进行积分可得到速度，对速度进行积分又可得到位移，进一步求得速度和位移的表达式如下：

$$v(t) = \begin{cases} v_1 + \frac{1}{2}\operatorname{sgn}(v_2 - v_1)Jt^2, & 0 \leq t \leq \frac{(t_r - t_{hr})}{2} \\ v_1 + \frac{1}{8}\operatorname{sgn}(v_2 - v_1)Jt_{thr}^2 \\ \quad + \frac{1}{2}\operatorname{sgn}(v_2 - v_1)Jt_{thr}(t - \frac{t_{thr}}{2}), & \frac{(t_r - t_{hr})}{2} < t \leq \frac{(t_r + t_{hr})}{2} \\ v_1 + \frac{1}{4}\operatorname{sgn}(v_2 - v_1)Jt_{thr}^2 + \frac{1}{2}\operatorname{sgn}(v_2 - v_1)Jt_{thr}t_{hr} \\ \quad - \frac{1}{2}\operatorname{sgn}(v_2 - v_1)J(t_r - t)^2, & \frac{(t_r + t_{hr})}{2} < t \leq t_r \end{cases} \quad (3-19)$$

$$L(t) = \begin{cases} v_1t + \frac{1}{6}\operatorname{sgn}(v_2 - v_1)Jt^3, & 0 \leq t \leq \frac{(t_r - t_{hr})}{2} \\ v_1t + \frac{1}{48}\operatorname{sgn}(v_2 - v_1)Jt_{thr}^3 + \frac{1}{8}\operatorname{sgn}(v_2 - v_1)Jt_{thr}^2(t - \frac{t_{thr}}{2}) \\ \quad + \frac{1}{4}\operatorname{sgn}(v_2 - v_1)Jt_{thr}(t - \frac{t_{thr}}{2})^2, & \frac{(t_r - t_{hr})}{2} < t \leq \frac{(t_r + t_{hr})}{2} \\ v_1t + \frac{1}{4}\operatorname{sgn}(v_2 - v_1)J(-\frac{1}{2}t_{thr}^3 - \frac{3}{2}t_{thr}^2t_{hr} - t_{thr}^2t + 2t_{thr}t_{hr}t) \\ \quad + \frac{1}{6}\operatorname{sgn}(v_2 - v_1)J(t_r - t)^3, & \frac{(t_r + t_{hr})}{2} < t \leq t_r \end{cases} \quad (3-20)$$

其中  $\operatorname{sgn}(x)$  为符号函数，当  $x > 0$  时，取值为 1； $x < 0$  时，取值为 -1。

### 3.2.2 基于给定运行速度的 S 型速度规划器

已知运行位移为  $L$ 、初始速度为  $v_s$ 、末速度为  $v_e$ 、运行速度为  $v$ 、系统最大加速度为  $a_{\max}$  和系统所能承受的加加速度为  $J$ （冲击）。

#### (1) 速度处理

首先计算临界位置，即 S 型速度曲线中不存在匀速段，由初始速度变速到运行速度，再由运行速度变速到末速度。如图 3-6 (a)、(b) 和 (c) 所示，主要分为三类：运行速度比初始速度和末速度都高、运行速度比初始速度和末速度都低和运行速度在初始速度和末速度之间。根据公式 (3-20) 分别计算初始速度变速到运行速度和运行速度变速到末速度的运动路程之和，记为  $L_1$ 。

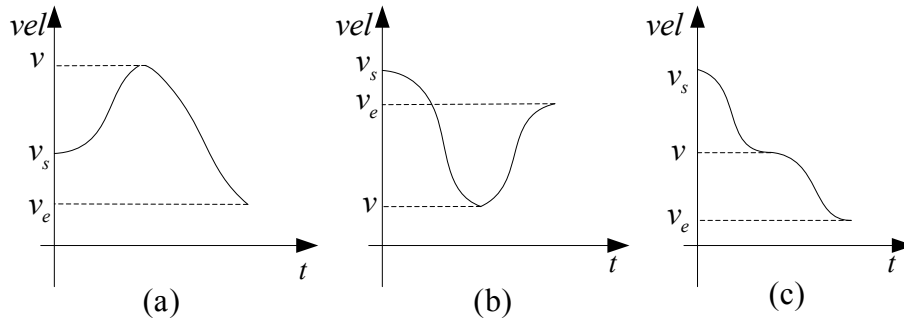


图 3-6 临界位置情况

Fig. 3-6 The case of critical position

如果  $L_1 < L$  (两示教点之间的运行位移，由用户给定)，那么说明用户设定的初始速度、末速度和运行速度合理，可以由初始速度变速到运行速度，然后匀速运动一段时间，再由运行速度变速到末速度，可以得到一个可行解，匀速运动的时间由路程差  $(L - L_1)$  除以运行速度  $v$  得到，规划结果为标准的 S 型速度曲线，分为变速-匀速-变速三段。

如果  $L_1 > L$ ，则先忽略用户设定的运行速度，直接由初始速度变速到末速度，并通过公式 (3-20) 计算变速路程  $L_2$ 。

如果  $L_1 > L$  并且  $L_2 > L$ ，则说明用户给定运行位移太短，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)，这时需对末速度进行重新规划，在区间  $[v_s, v_e]$  中搜索实际的末速度，如图 3-7 所示，实际末速度小于用户设定的末速度，规划结果为只有变速段。

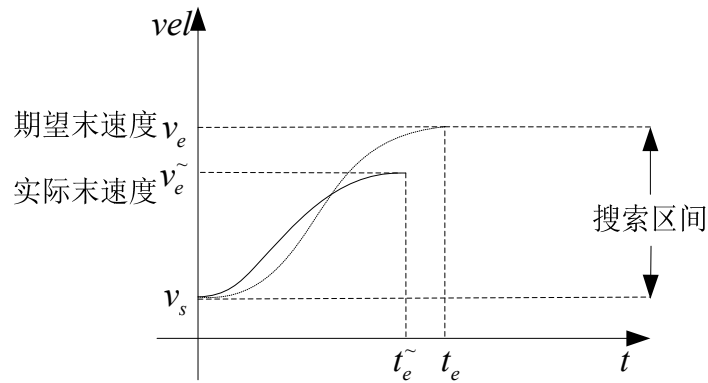


图 3-7 重新规划末速度

Fig. 3-7 Replanning final velocity

如果  $L_1 > L$  并且  $L_2 < L$ ，在这种情况下需要对运行速度进行重新规划，分以下三种情况进行处理。

1) 用户设定的运行速度高于初始速度和末速度

如图 3-8 所示，用户设定的运行速度高于初始速度和末速度时，以  $[\max(v_s, v_e), v]$  为搜索区间，查找实际的运行速度，最终得到可行解，规划结果由两段组成：变速-变速。

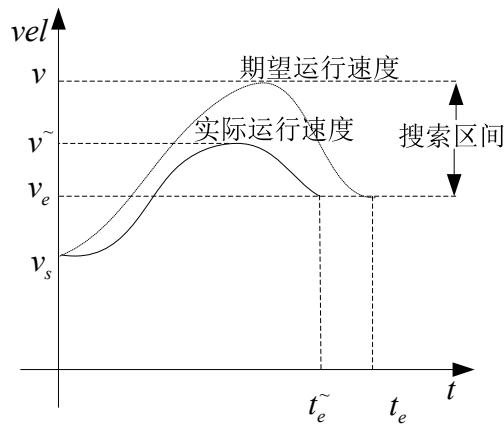


图 3-8 用户设定的运行速度高于初始速度和末速度

Fig. 3-8 Speed set by user is higher than the speed of first and last

2) 用户设定的运行速度低于初始速度和末速度

如图 3-9 所示，用户设定的运行速度低于初始速度和末速度时，以  $[v, \min(v_s, v_e)]$  为搜索区间，查找实际的运行速度，最终得到可行解，规划结果由两段组成：变

速-变速。

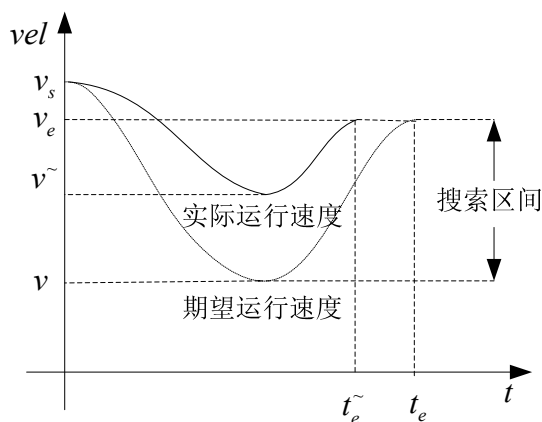


图 3-9 用户设定的运行速度低于初始速度和末速度

Fig. 3-9 Speed set by user is lower than the speed of first and last

### 3) 用户设定的运行速度介于初始速度和末速度之间

用户设定的运行速度介于初始速度和末速度之间时，直接进行初始速度到末速度的变速处理，将剩余的运动路程( $L-L_2$ )作为匀速段插入到变速段之前或之后，为了提高机器人末端执行器的运动效率，当  $v_s > v_e$  时，将匀速段以速度为  $v_s$  运行，插入到变速段之前，如图 3-10 中 (a) 图所示；反之，将匀速段以速度为  $v_e$  运行，插入到变速段之后，如图 3-10 中 (b) 图所示。最终得到可行解，规划结果由两段组成：匀速-变速或变速-匀速。

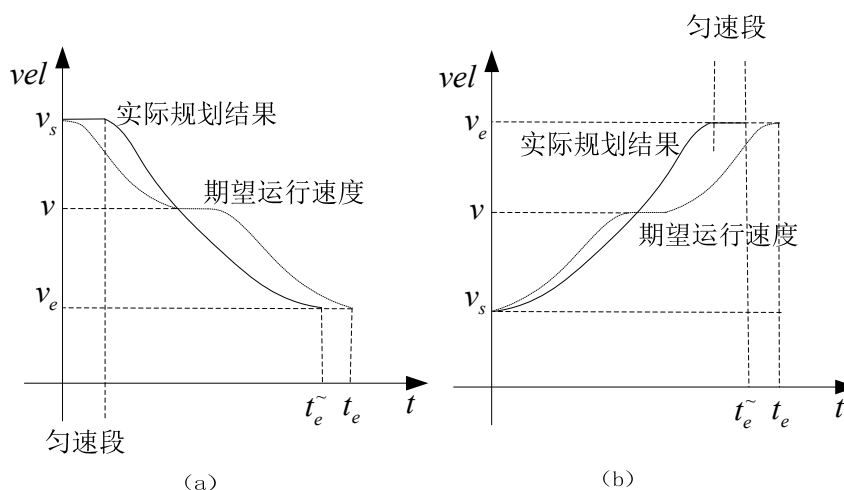


图 3-10 用户设定的运行速度介于初始速度和末速度之间

Fig. 3-10 Speed set by user is among the speed of first and last

## (2) 整体规划流程

首先，检查用户输入参数有效性，其次，处理用户输入速度特殊的情况，如果用户设定的运行速度与初始速度和末速度都相等，那么只需要以运行速度  $v$  匀速运动完位移  $L$  即可；如果用户设定的运行速度与初始速度和末速度其中之一相等，可采用图 3-10 的规划方法，在变速段之前或之后插入一段匀速段，匀速段的速度为用户设定的运行速度  $v$ 。一般情况则按照图 3-11 的流程处理，最终完成速度规划。

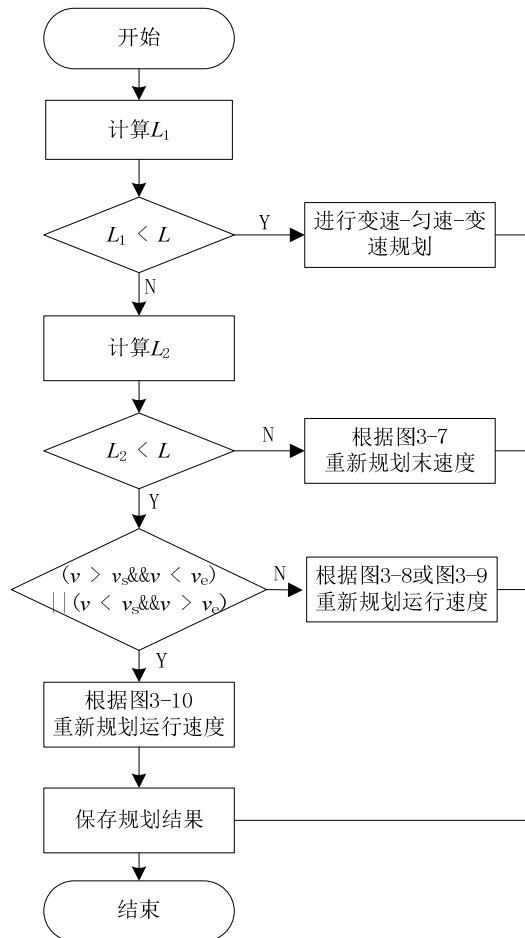


图 3-11 整体规划流程图

Fig. 3-11 The flowchart of the total planning

基于给定运行速度的 S 型速度规划器能保证速度曲线都呈 S 形状，服务于上层轨迹规划算法，使得机器人末端执行器在运行轨迹上速度平滑并且能有效地避免了刚体冲击。但是由于基于给定运行速度的 S 型速度规划器运行时间的不确定性，在一些运行时间要求严格的场合下，就不适用了，因此在下一节提出了基于

给定运行时间的 S 型速度规划。

### 3.2.3 基于给定运行时间的 S 型速度规划器

已知运行位移为  $L$ 、初始速度为  $v_s$ 、末速度为  $v_e$ 、最大运行速度为  $v_{\max}$ 、运行时间  $T$ ，系统最大加速度为  $a_{\max}$  和系统所能承受的加加速度为  $J$ （冲击）。

#### (1) 速度处理

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以下四种情况。

情况一：平均速度  $v$  小于初始速度  $v_s$  和末速度  $v_e$ 。

情况二：平均速度  $v$  大于初始速度  $v_s$  和末速度  $v_e$ 。

情况三：初始速度  $v_s$  大于平均速度  $v$  且平均速度  $v$  大于末速度  $v_e$ 。

情况四：初始速度  $v_s$  小于平均速度  $v$  且平均速度  $v$  小于末速度  $v_e$ 。

关键问题就是如何根据平均速度、初始速度和末速度，确定每种情况的运行速度调整或末速度调整方法，从而进行运行位移  $L$  和运行时间  $T$  的匹配，本文采取的方法是首先匹配运行位移  $L$ ，当运行位移  $L$  匹配成功后再进行运行时间  $T$  的匹配，返回计算得到的最合速度规划结果。

#### (1.1) 平均速度比初始速度和末速度都小

首先，计算初始速度到平均速度的变速，并得到变速位移  $L_1$ ，如果该变速位移  $L_1$  大于给定运行位移  $L$ ，那么说明初始速度太大或用户设定运行时间不合理，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，如图 3-12 (a) 图所示，在这种情况下，给定运行时间的 S 型速度规划失败，并返回标志提醒用户。

其次，计算平均速度到末速度的变速，并得到变速位移  $L_2$ ，如果两段变速位移之和  $(L_1 + L_2)$  大于给定运行位移  $L$ ，那么说明初始速度或末速度太大，亦或给定运行时间不合理，在这种情况下，如图 3-12 (b) 图所示，确定末速度的搜索区间  $[v, v_e]$ ，降低末速度。反之，如果两段变速位移之和  $(L_1 + L_2)$  小于等于给定运行位移  $L$ ，如图 3-12 (c) 图所示，确定运行速度的搜索区间  $[0, v]$ ，查找是否存在合适的运行速度匹配运行位移  $L$  和运行时间  $T$ ，一旦匹配成功，保存规划结果，返回相应标志给用户。



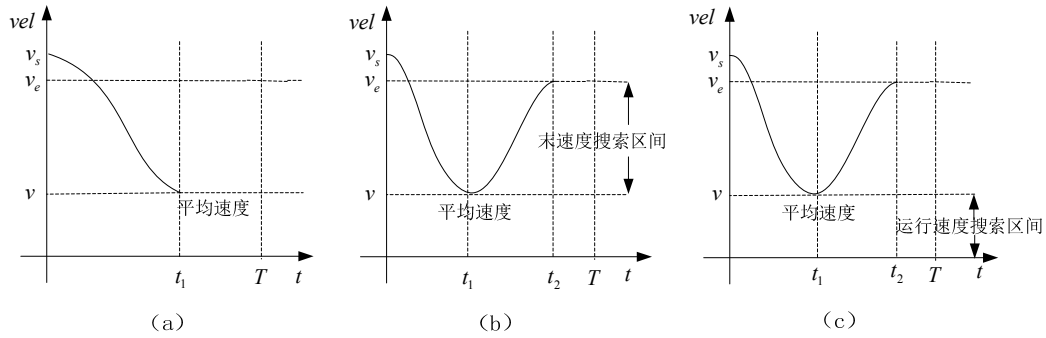


图 3-12 平均速度比初始速度和末速度都小的情况分析

Fig. 3-12 Analysis of average speed lower than the speed of first and last

当平均速度比初始速度和末速度都小时，规划流程如图 3-13 所示，详细规划步骤如下：

(a) 计算变速位移  $L_1$ ，如果  $L_1 > L$ ，规划失败，设置相应标志，执行步骤 e；否则执行步骤 b。

(b) 判断是否达到最大查找步数，如果大于，规划失败，设置相应标志，执行步骤 e；否则执行步骤 c。

(c) 计算变速位移  $L_2$ ，如果  $L_1 + L_2 < L$ ，执行步骤 d；否则根据末速度搜索区间  $[v, v_e]$ ，降低末速度，执行步骤 b。

(d) 根据运行速度搜索区间  $[0, v]$  查找运行速度，并进行运行位移  $L$  和运行时间  $T$  匹配，如果匹配成功，则保存规划结果，设置相应标志，执行步骤 e；否则根据末速度搜索区间  $[v, v_e]$ ，降低末速度，执行步骤 b。

(e) 程序返回。

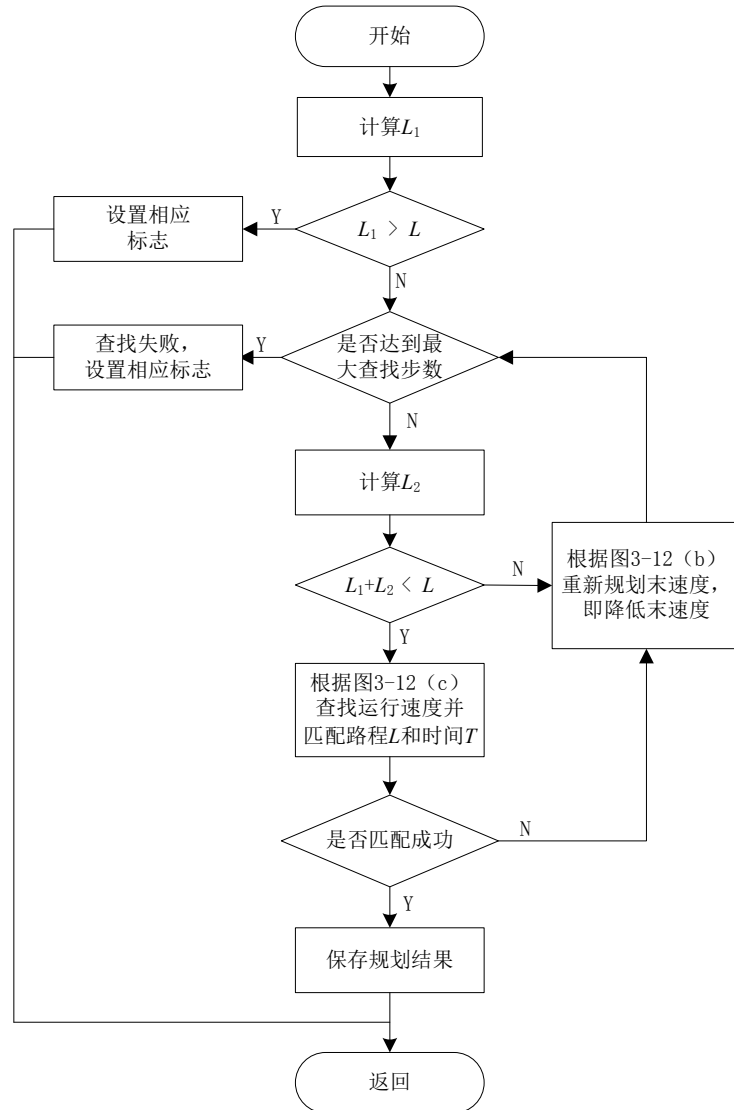


图 3-13 平均速度比初始速度和末速度都小的规划流程图

Fig. 3-13 Planning flowchart of average speed lower than the speed of first and last

### (1.2) 平均速度比初始速度和末速度都大

首先, 计算初始速度到平均速度的变速和平均速度到末速度的变速, 分别得到变速位移  $L_1$  和  $L_2$ 。

如果两段变速位移之和 ( $L_1+L_2$ ) 大于给定运行位移  $L$ , 那么说明初始速度、末速度太小或用户设定运行时间不合理, 如图 3-14 (a) 所示, 在这种情况下, 给定运行时间的 S 型速度规划失败, 返回相应标志给用户。

反之, 如图 3-14 (b) 所示, 如果两段变速位移之和 ( $L_1+L_2$ ) 小于等于给定运行位移  $L$ , 那么可以确定运行速度搜索区间  $[v, v_{\max}]$ , 在此区间内, 必定会存在

一个合适的运行速度匹配运行位移  $L$ ，同时进行运行时间  $T$  的匹配，一旦匹配成功，保存规划结果，返回相应标志给用户。

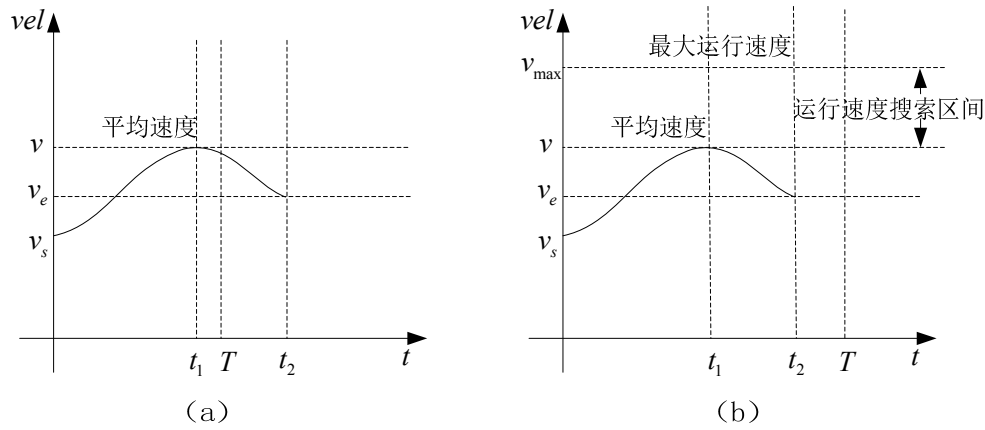


图 3-14 平均速度比初始速度和末速度都大的情况分析

Fig. 3-14 Analysis of average speed higher than the speed of first and last

当平均速度比初始速度和末速度都大时，规划流程如图 3-15 所示，详细规划步骤如下：

(a) 计算变速位移  $L_1$  和  $L_2$ ，如果  $(L_1 + L_2) > L$ ，规划失败，设置相应标志，执行步骤 d；否则执行步骤 b。

(b) 判断是否达到最大查找步数，如果大于，规划失败，设置相应标志，执行步骤 d；否则执行步骤 c。

(c) 根据运行速度搜索区间  $[v, v_{max}]$  查找运行速度，并进行运行位移  $L$  和运行时间  $T$  匹配，如果匹配成功，则保存规划结果，设置相应标志，执行步骤 d；否则根据运行速度搜索区间  $[v, v_{max}]$ ，提升运行速度，执行步骤 b。

(d) 程序返回。

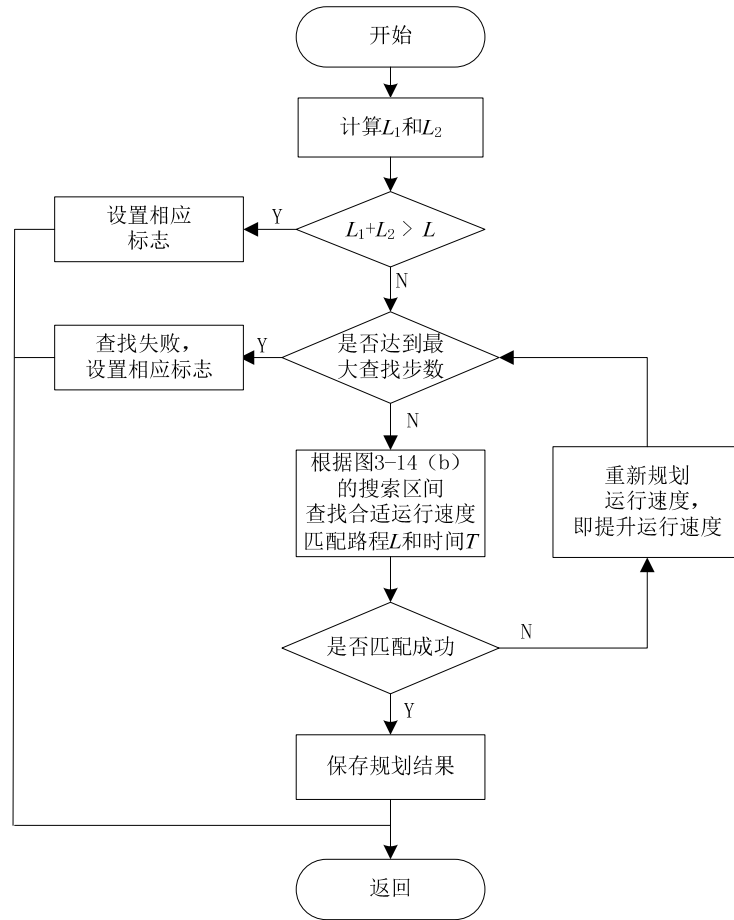


图 3-15 平均速度比初始速度和末速度都大的规划流程图

Fig. 3-15 Planning flowchart of average speed higher than the speed of first and last

### (1.3) 初始速度大于平均速度且平均速度大于末速度

首先, 计算初始速度到平均速度和平均速度到末速度的变速, 分别得到变速位移  $L_1$  和  $L_2$ , 继续计算初始速度  $v_s$  到末速度  $v_e$  的变速位移  $L_3$ 。

如果变速位移之和 ( $L_1 + L_2$ ) 大于运行位移  $L$ , 并且变速位移  $L_3$  也大于运行位移  $L$ , 那么说明初始速度太大或末速度太小, 亦或用户设定运行时间不合理, 如图 3-16 (a) 和 (b) 所示, 在这种情况下, 无法实现给定时间的运动规划, 规划失败, 设置相应标志并返回。反之, 如图 3-16 (c) 和 (d) 所示在区间  $[v_s, v_e]$  之内, 必定存在一个运行速度, 使得初始速度到运行速度和运行速度到末速度的变速位移之和小于等于运行位移  $L$ , 确定运行速度的搜索区间为  $[v_s, v_e]$ , 查找合适的运行速度, 同时进行运行位移  $L$  和运行时间  $T$  的匹配, 一旦匹配成功, 保存规划结果, 返回相应标志给用户。

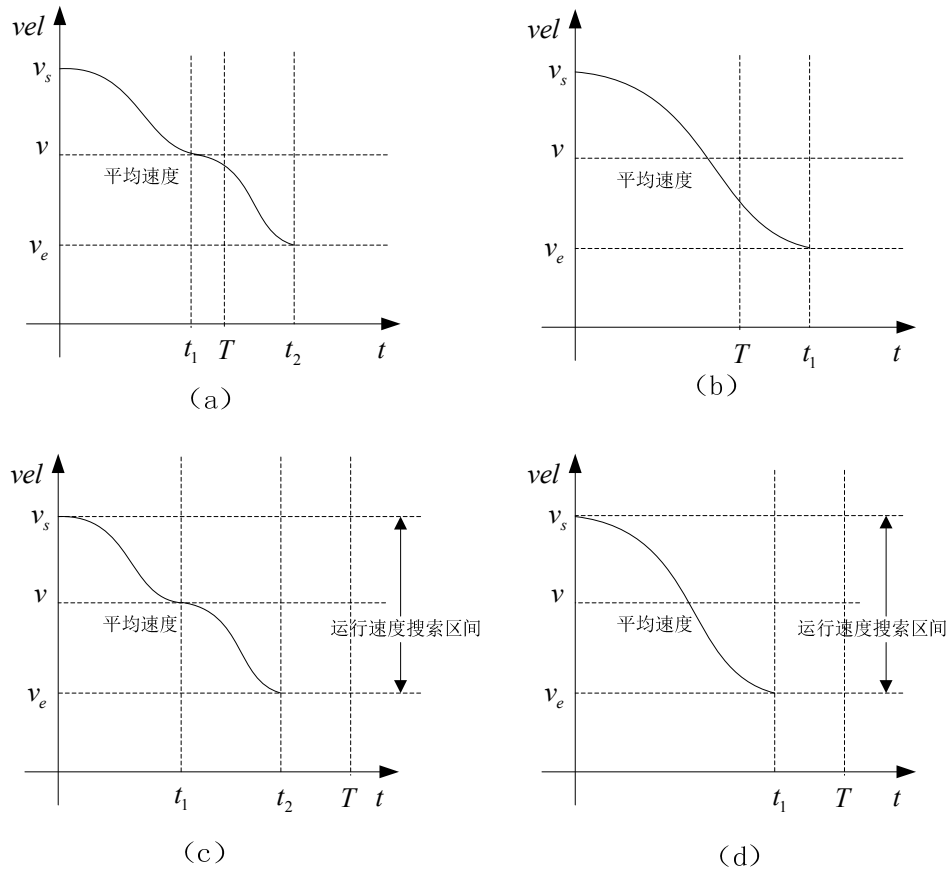


图 3-16 初始速度大于平均速度且平均速度大于末速度的情况分析

Fig. 3-16 Analysis of first speed higher than average speed and average speed higher than last speed

当初速度大于平均速度且平均速度大于末速度  $v_e$  时, 规划流程如图 3-17 所示, 详细规划步骤如下:

- (a) 计算变速位移  $L_1$ 、 $L_2$  和  $L_3$ 。
- (b) 如果  $L_1 + L_2 > L$  且  $L_3 > L$ , 规划失败, 设置相应标志, 执行步骤 e。否则执行步骤 c。
- (c) 判断是否达到最大查找步数, 如果大于, 规划失败, 设置相应标志, 执行步骤 e; 否则执行步骤 d。
- (d) 根据运行速度搜索区间  $[v_s, v_e]$  查找合适的运行速度, 进行运行位移  $L$  和运行时间  $T$  的匹配, 如果匹配成功, 保存规划结果并设置相应标志, 执行步骤 e。否则, 继续调整运行速度, 执行步骤 c。
- (e) 程序返回。

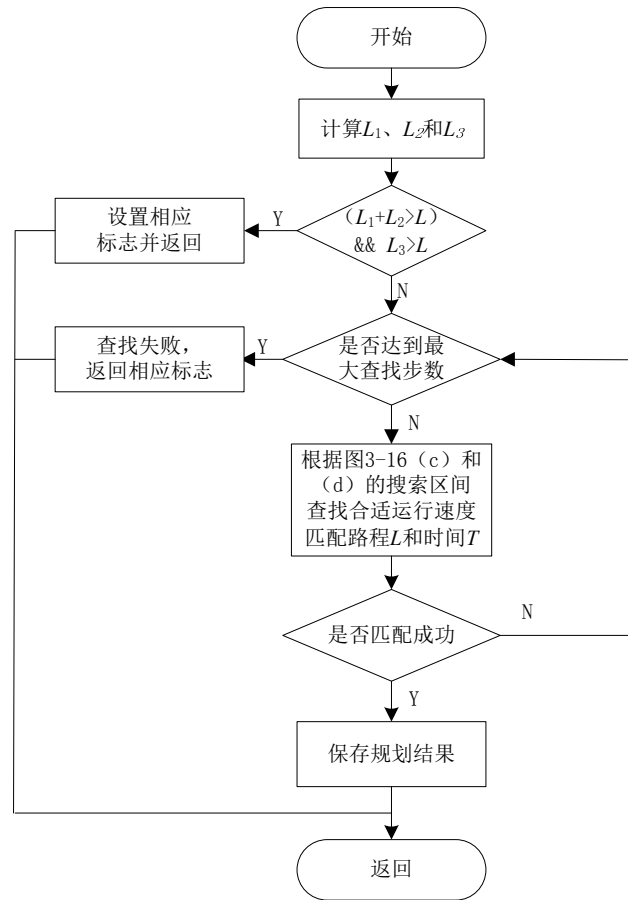


图 3-17 初始速度大于平均速度且平均速度大于末速度的规划流程图

Fig. 3-17 Planning flowchart of first speed higher than average speed and average speed higher than last speed

#### (1.4) 初始速度小于平均速度且平均速度小于末速度

首先, 计算初始速度到平均速度的变速位移  $L_1$ 。

如果变速位移  $L_1$  大于运行位移  $L$ , 那么说明给定的初始速度太小或用户设定的运行时间不合理, 无法满足给定运行时间的 S 型速度规划, 如图 3-18(a) 所示, 在这种情况下, 规划失败, 设置相应标志并返回。

如果变速位移  $L_1$  小于等于运行位移  $L$ , 继续计算平均速度到末速度的变速位移  $L_2$ 。

如果变速位移  $L_1$  小于等于运行位移  $L$  且变速位移  $L_2$  大于运行位移  $L$ , 那么说明用户设定的末速度太大, 如图 3-18(b) 所示, 在这种情况下, 需要重新规划末速度, 即降低末速度, 确定末速度搜索区间  $[v, v_e]$ , 在区间  $[v, v_e]$  内, 查找合适的末速度  $v_e'$  满足变速位移  $L_2$  小于运行位移  $L$ 。进而确定运行速度搜索区间  $[v, v_e']$ ,

如图 3-18 (c) 所示, 查找合适的运行速度进行运行位移  $L$  和运行时间  $T$  的匹配, 一旦匹配成功, 保存规划结果, 设置相应标志后返回。

如果变速位移  $L_1$  小于等于运行位移  $L$  且变速位移  $L_2$  小于等于运行位移  $L$ , 如图 3-18 (d) 所示, 直接确定运行速度的搜索区间为  $[v, v_e]$ , 在区间  $[v, v_e]$  内, 查找合适的运行速度进行运行位移  $L$  和运行时间  $T$  的匹配, 一旦匹配成功, 保存规划结果, 设置相应标志后返回。

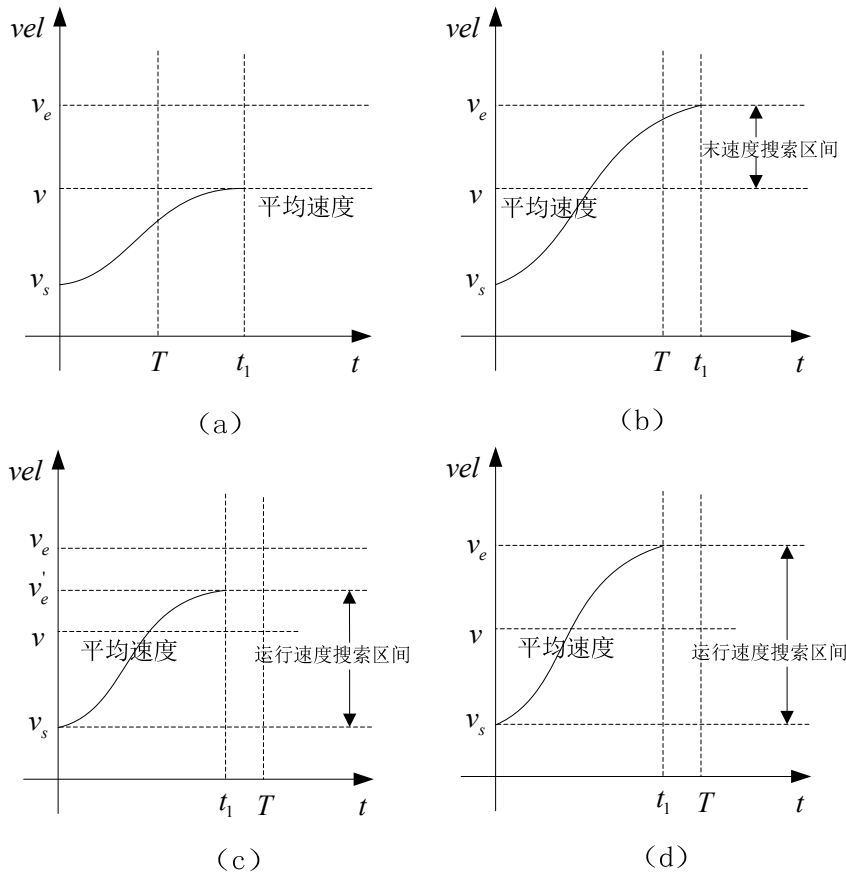


图 3-18 初始速度小于平均速度且平均速度小于末速度的情况分析

Fig. 3-18 Analysis of first speed lower than average speed and average speed lower than last speed

当初始速度大于平均速度且平均速度大于末速度时, 规划流程如图 3-19 所示, 详细规划步骤如下:

- (a) 计算变速位移  $L_1$ 。
- (b) 如果  $L_1 > L$ , 规划失败, 设置相应标志, 执行步骤 g。否则执行步骤 c。
- (c) 判断是否达到最大查找步数, 如果大于, 规划失败, 设置相应标志, 执行步骤 g; 否则执行步骤 d。

(d) 计算变速位移  $L_2$ 。

(e) 如果  $L_2 < L$ ，执行步骤 f，否则根据末速度搜索区间  $[v, v_e]$  或  $[v, v_e']$ ，降低末速度，执行步骤 c。

(f) 根据运行速度搜索区间  $[v_s, v_e]$  或  $[v_s, v_e']$  查找合适的运行速度，进行运行位移  $L$  和运行时间  $T$  的匹配，如果匹配成功，保存规划结果并设置相应标志，执行步骤 g。否则根据末速度搜索区间  $[v, v_e]$  或  $[v, v_e']$ ，降低末速度，执行步骤 c。

(g) 程序返回。

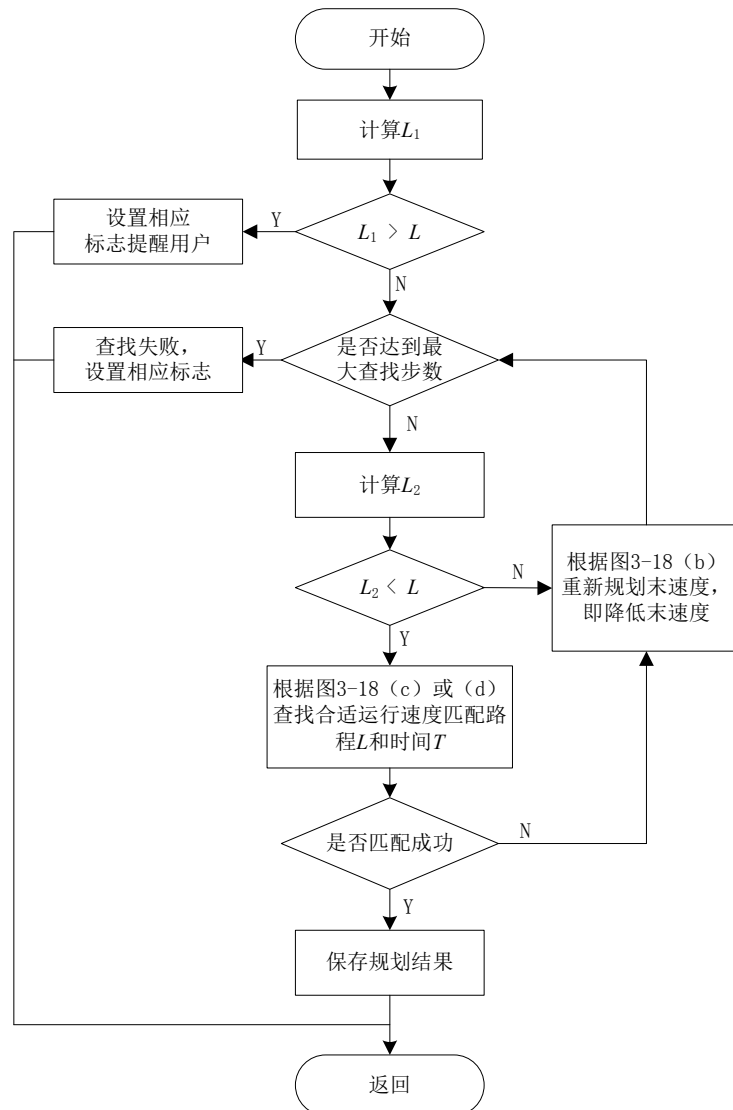


图 3-19 初始速度小于平均速度且平均速度小于末速度的规划流程图

Fig. 3-19 Planning flowchart of first speed lower than average speed and average speed lower than last speed



## (2) 整体规划流程

首先，检查用户输入参数有效性。其次，处理用户输入速度特殊的情况，如果平均速度  $v$  与初始速度  $v_s$  和末速度  $v_e$  都相等，那么只需要以平均速度  $v$  匀速运动完位移  $L$  即可，否则根据平均速度、初始速度和末速度之间的关系，分为四种情况：平均速度小于初始速度和末速度、平均速度大于初始速度和末速度、初始速度大于平均速度且平均速度大于末速度和初始速度小于平均速度且平均速度小于末速度。在上一节中，分析了每一种情况详细的处理方法。整体规划流程如图 3-20 所示。

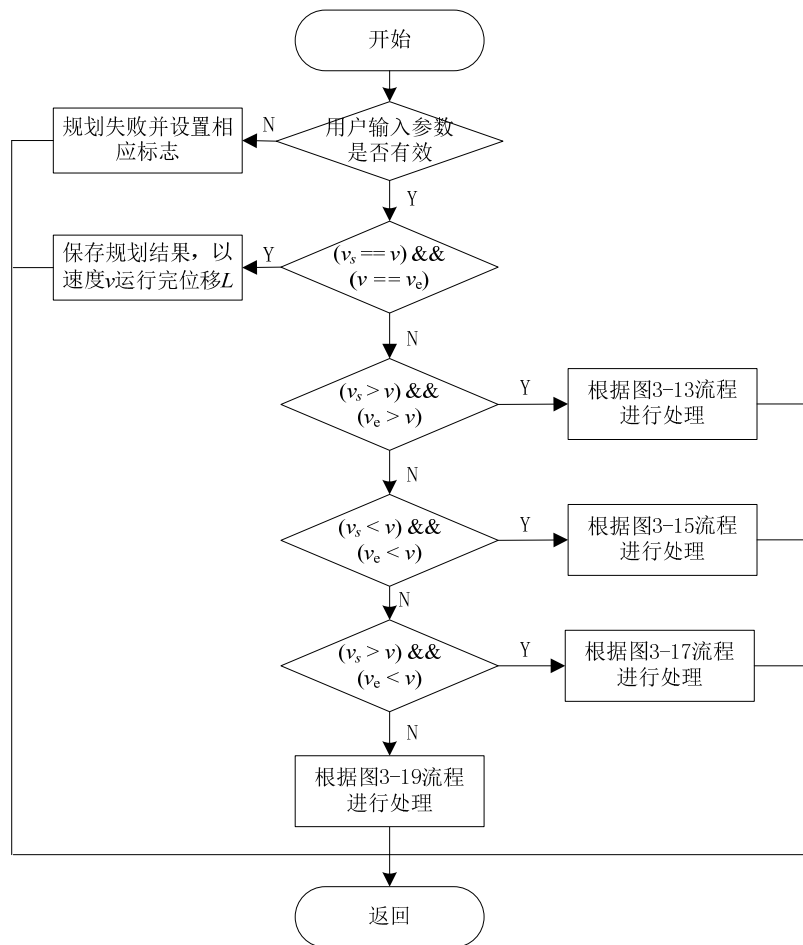


图 3-20 整体规划流程图

Fig. 3-20 The flowchart of the total planning

### 3.2.4 S 型速度规划器类接口设计

S 型速度规划器类 UML 图如图 3-21 所示，数据成员和成员函数描述如表 3-1 所示。

表 3-1 参数描述

Table. 3-1 Parameter description

m_jerk	系统所能承受的最大冲击
m_acc	系统最大加速度
m_vel_s	用户设定的初始速度
m_vel_e	用户设定的末速度
m_vel_max	系统最大运行速度
m_duration	用户设定的运行时间
m_distance	用户设定的运行位移
SetParameters( )	用户通过该接口对数据成员进行初始化
SetProfileVelocity( )	根据用户给定的运行速度进行 S 型速度规划，规划成功返回 0，失败返回-1，其他返回 1
SetProfileDuration( )	根据用户给定的运行时间进行 S 型速度规划，规划成功返回 0，失败返回-1，其他返回 1
Duration( )	规划成功后，通过该接口可获得运动周期
Pos( )	规划成功后，获取对应时刻的位移
Vel( )	规划成功后，获取对应时刻的速度
Acc( )	规划成功后，获取对应时刻的加速度

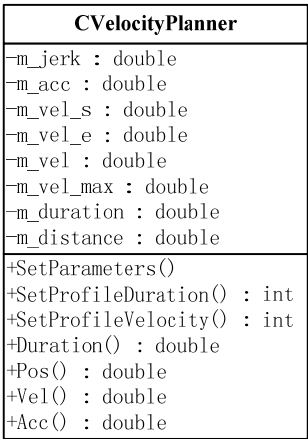


图 3-21 CvelocityPlanner 类的设计

Fig. 3-21 The design of class CvelocityPlanner

### 3.3 本章小结

本章首先介绍了工业运动控制系统中常见的速度控制曲线：梯形速度曲线、正弦型速度曲线和 S 型速度曲线，通过原理上的分析对比，确定了将高性能的 S 型速度控制曲线作为研究对象，将 S 型速度曲线进行特殊处理，把速度有变化的阶段统一成变速段，推导出了位移、速度和加速度的解析表达式，在此基础上，研究和设计了两种自适应可控 S 型速度规划器：基于给定运行速度的 S 型速度规划器和基于给定运行时间的 S 型速度规划器，本章设计的速度规划器是后面笛卡尔空间轨迹规划的重要基础。

## 4 笛卡尔空间轨迹规划研究

笛卡尔空间轨迹规划的目的是生成含机器人末端执行器运行信息的插补点。在轨迹规划前，用户需对机器人进行示教，同时设定速度、加速度和加加速度等信息，系统在此过程中记录下关键的示教点位置以及示教点之间的轨迹类型，经过轨迹插补和速度规划后便生成了含运行信息的插补点。机器人在作业时，通常要求机器人末端执行器在笛卡尔空间中按照严格的轨迹进行运动，通过逆解算法将空间位姿矩阵（含运行信息的插补点）转换成关节角度，进而输入运动控制卡驱动各关节电机转动，使得机器人末端执行器沿着规划的轨迹进行运动。本章结合上一章的速度规划算法进行笛卡尔空间轨迹规划研究。首先研究空间直线、圆弧和三次 B 样条曲线的轨迹规划算法，最后以三次 B 样条曲线为基础，研究空间直线-直线、空间直线-圆弧和三次 B 样条-三次 B 样条连续运动过程中平滑过渡。

### 4.1 空间轨迹规划

#### 4.1.1 直线和圆弧轨迹规划

直线插补和圆弧插补是工业控制中最常用的两种轨迹插补方式，几乎所有的复杂轨迹都可分解成直线插补和圆弧插补的组合，因此空间直线和圆弧的轨迹规划也是机器人运动控制器不可或缺的一部分。

##### (1) 空间直线轨迹规划

已知空间中首末两个示教点，首末两点位置分别为  $p_1 = (x_1, y_1, z_1)$ ， $p_2 = (x_2, y_2, z_2)$ ，可写出直线的参数方程如下：

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} + k \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{bmatrix} \quad (4-1)$$

两个示教点之间的直线段距离，也就是  $p_1$  到  $p_2$  之间的距离为：

$$L_{p_1 p_2} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (4-2)$$

直线上任意一点  $p = (x_p, y_p, z_p)$  到起点  $p_1$  的距离可表示为：

$$L_{pp_1} = \sqrt{(x_p - x_1)^2 + (y_p - y_1)^2 + (z_p - z_1)^2} \quad (4-3)$$

那么由公式（4-1）、公式（4-2）和公式（4-3）可求得直线参数：

$$k = \pm \frac{L_{p_1 p}}{L_{p_1 p_2}} \quad (4-4)$$

考虑到轨迹插补点都在  $p_1$  到  $p_2$  之间, 所以  $k$  只能为正值, 那么  $p_1$  到  $p_2$  之间直线轨迹方程可写成:

$$\begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} + \frac{L_{p_1 p}}{L_{p_1 p_2}} \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{bmatrix} \quad (4-5)$$

假设用户设定过起点  $p_1$  的标量速度为  $v_s$ , 过终点  $p_2$  的标量速度为  $v_e$ , 首末点之间期望运行速度为  $v$ , 最大加速度  $a_{\max}$ , 系统最大冲击  $J$ , 又因为两点间运动距离已由公式 (4-2) 求出, 将这些参数输入 S 型速度规划器进行规划。规划完成后, 调用接口 Duration( ) 获取运行总时间  $T$ , 将  $T$  进行  $N$  等分, 插补时刻  $t_i = (iT/N)$ , 其中  $i = 0, 1, 2, \dots, N$ 。进而通过接口 Pos( )、Vel( ) 和 Acc( ) 分别获取  $t_i$  时刻的标量位移  $L_i$ 、速度  $v_i$  和加速度  $a_i$ 。

还需要将标量位移、速度和加速度插补序列转换为空间位置、空间速度和空间加速度插补点序列, 将  $L_i$  直接代入公式 (4-5) 可得到直线空间位置。求取空间速度和空间加速度, 要先求得位置插补点上的方向向量, 又直线上所有点的方向向量都相同, 由起点指向终点, 如下:

$$\vec{n} = \frac{(x_2 - x_1, y_2 - y_1, z_2 - z_1)}{L_{p_1 p_2}} \quad (4-6)$$

将标量速度  $v_i$  和加速度  $a_i$  乘上公式 (4-6) 就可得到空间速度和空间加速度。

## (2) 空间圆弧轨迹规划

在求解空间圆弧轨迹时, 通常是给定三点  $p_s = (x_s, y_s, z_s)$ 、 $p_m = (x_m, y_m, z_m)$  和  $p_e = (x_e, y_e, z_e)$ , 并且这三点不共线, 两两之间不重合, 那么可以确定唯一一段圆弧使得机器人工具坐标系以  $p_s$  为起点, 经过中间点  $p_m$ , 运行到终点  $p_e$ 。直接求解圆弧轨迹插补点比较复杂, 通常是通过坐标系变换的方式求解, 也就是以这三点建立新的直角坐标系, 求解出圆弧轨迹插补点在新的坐标系中的值, 然后通过坐标系变换关系将这些值传递到原来坐标系中, 最终求解出插补点在原来坐标系中的值。下面给出详细的求解步骤:

### (2.1) 求解圆心位置 $p_o = (x_o, y_o, z_o)$ 和圆弧半径 $r$ <sup>[56]</sup>

根据定义, 在笛卡尔空间中, 不共线的三点可以确定唯一平面, 所以  $p_s$ 、 $p_m$  和  $p_e$  可以确定一个平面, 写出平面方程式如下:

$$\begin{vmatrix} x-x_s & y-y_s & z-z_s \\ x_m-x_s & y_m-y_s & z_m-z_s \\ x_e-x_s & y_e-y_s & z_e-z_s \end{vmatrix} = 0 \quad (4-7)$$

由  $\overrightarrow{p_s p_m} = (x_m - x_s, y_m - y_s, z_m - z_s)$  和点  $p_s$ 、 $p_m$  的中点坐标可以确定一个平面，写出平面方程式如下：

$$\begin{aligned} (x_m - x_s)(x - \frac{x_m + x_s}{2}) + (y_m - y_s)(y - \frac{y_m + y_s}{2}) \\ + (z_m - z_s)(z - \frac{z_m + z_s}{2}) = 0 \end{aligned} \quad (4-8)$$

同理，可写出垂直与  $\overrightarrow{p_m p_e}$ ，过点  $p_m$ 、 $p_e$  的中点坐标的平面方程，如下：

$$\begin{aligned} (x_e - x_m)(x - \frac{x_m + x_e}{2}) + (y_e - y_m)(y - \frac{y_m + y_e}{2}) \\ + (z_e - z_m)(z - \frac{z_m + z_e}{2}) = 0 \end{aligned} \quad (4-9)$$

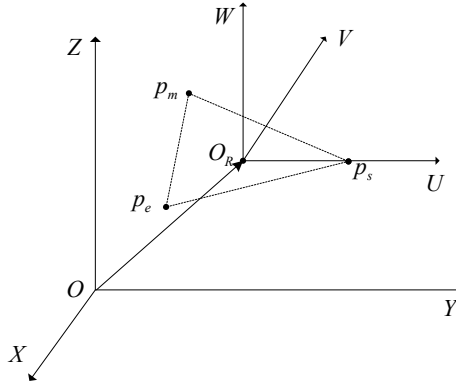
分析可知，这三个平面相交于一点，并且该点为圆心，由公式(4-7)、公式(4-8)和公式(4-9)可求出圆心坐标为  $p_o = (x_o, y_o, z_o)$ ，进而求得圆弧半径：

$$r = \sqrt{(x_s - x_o)^2 + (y_s - y_o)^2 + (z_s - z_o)^2} \quad (4-10)$$

## (2.2) 建立新坐标系 $O_R$ - $UVW$ 并推导坐标系之间的传递矩阵

如图 4-1 所示，新坐标系的坐标原点  $O_R$  建立在圆心  $p_o$  处，以  $\overrightarrow{p_o p_s}$  的指向作为新坐标系的  $U$  轴方向，其单位方向向量为：

$$u = \frac{\overrightarrow{p_o p_s}}{|\overrightarrow{p_o p_s}|} \quad (4-11)$$

图 4-1 新坐标系  $O_R-UVW$ Fig. 4-1 The new coordinate system  $O_R-UVW$ 

取垂直于圆弧所在平面的单位法向量作为  $W$  轴，单位法向量可以由  $\overrightarrow{p_s p_m}$  和  $\overrightarrow{p_m p_e}$  叉乘后归一化得到，写成：

$$w = \frac{\overrightarrow{p_s p_m} \times \overrightarrow{p_m p_e}}{|\overrightarrow{p_s p_m} \times \overrightarrow{p_m p_e}|} \quad (4-12)$$

$V$  轴按照右手法则建立，在  $u$  和  $w$  的叉乘方向上，写成：

$$v = u \times w \quad (4-13)$$

原坐标系  $O-XYZ$  中的点与新坐标系  $O_R-UVW$  中的点变换关系为：

$${}^O P = {}^O T_R {}^R P$$

$$\text{其中 } {}^O T_R = \begin{bmatrix} u_x & v_x & w_x & x_o \\ u_y & v_y & w_y & y_o \\ u_z & v_z & w_z & z_o \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4-14)$$

### (2.3) 在新坐标系 $O_R-UVW$ 中进行圆弧轨迹规划

假设用户设定过起点  $p_s$  的标量速度为  $v_s$ ，过终点  $p_e$  的标量速度为  $v_e$ ，首末点之间期望运行速度为  $v$ ，最大加速度  $a_{\max}$ ，系统最大冲击  $J$ ，又起点到终点的运动弧长为  $\theta r$ 。

同直线轨迹规划类似，首先调用 S 型速度规划器进行规划，规划完成后，将运行总时间  $T$  进行  $N$  等分，插补时刻  $t_i = (iT/N)$ ，其中  $i = 0, 1, 2, \dots, N$ 。然后通过相应接口可得到标量路程  $L_i$ 、速度  $v_i$  和加速度  $a_i$ 。

如图 4-2 (a) 所示，已知弧长  $L_i$ ，那么可以求得弧长所对应的圆心角：

$$\theta_i = \frac{L}{r} \quad (4-15)$$

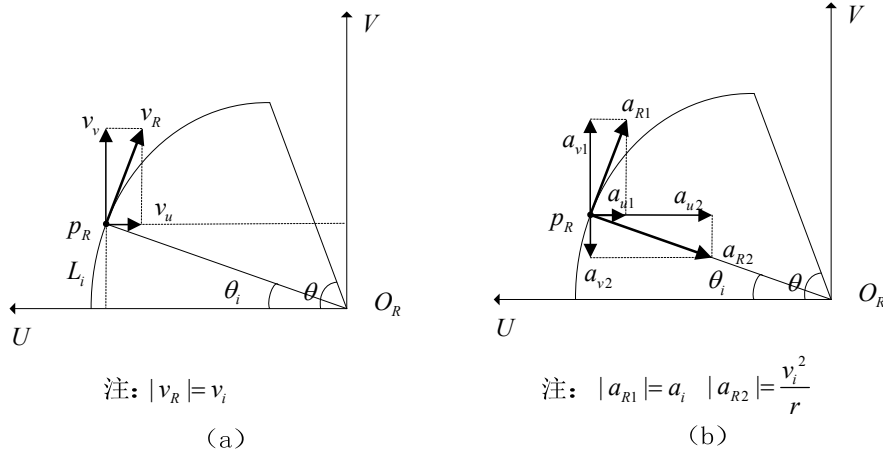


图 4-2 圆弧位置、速度和加速度在新坐标系中的描述

Fig. 4-2 Description of arc position, velocity, acceleration at the new coordinate system

由于圆弧建立在平面  $O_R-UVW$  上, 所以位置、速度和加速度在  $W$  轴上的作用都为 0。空间位置插补点的求取, 机器人末端工具在圆弧上移动弧长  $L_i$  后, 位于图 4-2 (a) 中  $p_R = (p_u, p_v, p_w)$ , 根据公式 (4-15) 可求得:

$$\begin{cases} p_u = r \cos \theta_i \\ p_v = r \sin \theta_i \\ p_w = 0 \end{cases} \quad (4-16)$$

空间速度插补点的求取, 在  $p_R$  点上的速度为  $v_R$ , 根据速度规划结果可知,  $v_R$  大小为  $v_i$ , 方向与圆弧相切, 那么可以很容易写出  $v_R = (-v_u, v_v, 0)$ , 根据公式 (4-15) 可求得:

$$\begin{cases} v_u = v_i \sin \theta_i \\ v_v = v_i \cos \theta_i \\ v_w = 0 \end{cases} \quad (4-17)$$

空间加速度插补点的求取, 如图 4-2 (b) 所示, 圆弧运动加速度分析比速度分析较复杂。在  $p_R$  存在两个加速度, 一个是切线加速度  $a_{R1}$ , 另一个是向心加速度  $a_{R2}$ , 根据速度规划结果可知  $a_{R1}$  的大小为  $a_i$ , 方向与圆弧相切,  $a_{R2}$  的大小为  $v_i^2/r$ , 方向由点  $p_R$  指向圆心。总的加速度  $a_R = (a_u, a_v, 0)$ , 其中  $a_u = -a_{u1} - a_{u2}$ ,  $a_v = a_{v1} - a_{v2}$ , 求得如下:



$$\begin{cases} a_u = -a_i \sin \theta_i - \frac{v_i^2}{r} \cos \theta_i \\ a_v = a_i \cos \theta_i - \frac{v_i^2}{r} \sin \theta_i \\ a_w = 0 \end{cases} \quad (4-18)$$

(2.4) 将新坐标系  $O_R-UVW$  中的轨迹规划结果变换到原坐标系  $O-XYZ$  中

在上一节中，完成了新坐标系中的圆弧的轨迹规划，现根据公式 (4-14) 可将新坐标系中的位置、速度和加速度的规划结果变换到原坐标系  $O-XYZ$  中，写成：

$$\begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = {}^O T_R \begin{bmatrix} p_u \\ p_v \\ p_w \\ 1 \end{bmatrix} \quad (4-19)$$

$$\begin{bmatrix} v_x \\ v_y \\ v_z \\ 1 \end{bmatrix} = {}^O T_R \begin{bmatrix} v_u \\ v_v \\ v_w \\ 1 \end{bmatrix} \quad (4-20)$$

$$\begin{bmatrix} a_x \\ a_y \\ a_z \\ 1 \end{bmatrix} = {}^O T_R \begin{bmatrix} a_u \\ a_v \\ a_w \\ 1 \end{bmatrix} \quad (4-21)$$

#### 4.1.2 三次 B 样条轨迹规划

B 样条曲线的概念最初由里森费尔德 (Riesenfeld) 等人提出，并在贝塞尔曲线的基础上，进行推广从而形成了 B 样条曲线，B 样条曲线具有贝塞尔曲线的优点，同时还具有良好局部修改能力、较好的逼近性及凸包性<sup>[57]</sup>。三次 B 样条曲线具有局部修改特性、凸包性、三阶可微等优点，参数表达形式简单，适合计算机编程，在计算机辅助制造中得到广泛应用。本节研究将三次 B 样条曲线应用于笛卡尔空间轨迹规划算法，目标是：在给定若干控制点或示教点的前提下，生成三次 B 样条曲线，借助上一章研究的 S 型速度规划算法，对位置、速度和加速度进行定周期插补，使机器人末端执行器依次平滑的通过曲线轨迹。

##### (1) B 样条曲线的定义及其重要性

B 样条曲线是由分段 B 样条基函数和控制点所决定的，给定  $n+1$  个控制点  $P_0, P_1, \dots, P_n$  和一个节点向量  $U = u_0, u_1, \dots, u_m$ ，其中  $u_0 \leq u_1 \leq \dots \leq u_m$ ， $p$  次 B 样

条曲线由这些控制点和节点向量定义为：

$$C(u) = \sum_{i=0}^n N_{i,p}(u) P_i, \quad 0 \leq u \leq 1 \quad (4-22)$$

B 样条曲线节点向量最大下标必须满足  $m=n+p+1$ ，上式中  $N_{i,p}(u)$  称为第  $i$  个  $p$  次规范 B 样条基函数，是由节点向量  $U$  所决定的  $p$  次分段多项式，递归定义如下：

$$\begin{cases} N_{i,0}(u) = \begin{cases} 1, & u_i \leq u \leq u_{i+1} \\ 0, & \text{其他} \end{cases} \\ N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \\ \text{规定: } \frac{0}{0} = 0 \end{cases} \quad (4-23)$$

上述公式称为德布尔递归公式， $N_{i,p}(u)$  的两个下标  $i$  和  $p$  分别表示基函数的序列和基函数的次数。如图 4-3 所示，使用三角计算格式更容易理解  $N_{i,p}(u)$  的计算过程，左边第一列为所有节点区间，第二列为零次基函数，根据公式 (4-23) 可知，在零次基函数的已知的情况下，又可以求出一次基函数，以此类推，就可求出  $N_{i,p}(u)$ 。

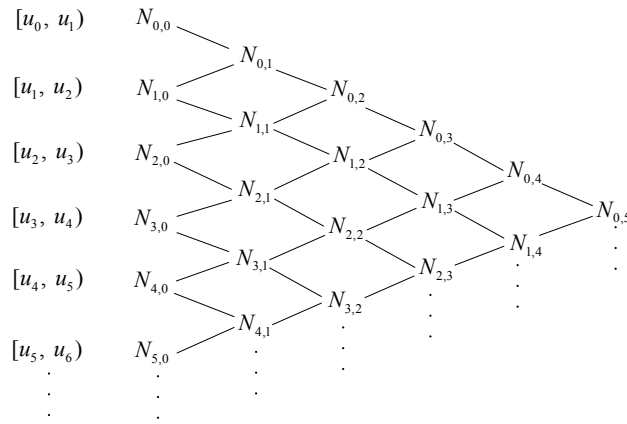


图 4-3 德布尔三角递推描述

Fig. 4-3 Description of Cox-de Boor triangle recursive

B 样条曲线具有以下重要性质：

- 1) B 样条曲线是逐段曲线，每一个分量都是  $p$  次曲线。
- 2) 第一个和最后一个节点重复度为  $p+1$  的 B 样条曲线经过第一个和最后一个控制点且与控制折线的第一边和最后一边相切。
- 3)  $P_i$  只影响在区间  $[u_i, u_{i+p+1})$  上的曲线  $C(u)$ ，即局部修改特性。
- 4)  $C(u)$  在重复度  $k$  的节点上是  $C_{p-k}$  连续的，即可微性。

5) 样条曲线包含在控制折线的凸包内。即凸包性。更特别地, 如果  $u$  在节点区间  $[u_i, u_{i+1})$  里, 那么  $C(u)$  在控制点  $P_{i-p}, P_{i-p+1}, \dots, P_i$  的凸包里。

6)  $p$  次 B 样条曲线的导数是另外一条  $p-1$  次的 B 样条曲线, 在原来的节点向量上移动第一个和最后一个节点向量变为  $p$  次, 控制点为  $Q_0, Q_1, \dots, Q_n$ 。表示如下:

$$C'(u) = \sum_{i=0}^{n-1} N_{i,p-1}(u)Q_i \quad \text{其中 } Q_i = p \frac{P_{i+1} - P_i}{u_{i+p+1} - u_{i+1}} \quad (4-24)$$

## (2) 三次 B 样条曲线的求解

根据 B 样条曲线性质及公式 (4-22) 可知, 三次 B 样条曲线第  $l$  段曲线的表达式为:

$$C_l(u) = \sum_{i=0}^3 N_{i+l,3}(u)P_{i+l}, \quad 0 \leq u \leq 1 \quad (4-25)$$

其中  $P_{i+l}$  为控制点,  $l=0, 1, \dots, n-3$ 。在笛卡尔空间轨迹规划中, 三次 B 样条曲线的应用存在两种情况: 一种是已知一些示教点, 然后生成一条过这些示教点的三次 B 样条曲线, 进而在曲线轨迹上进行速度规划。求解过程是先构造节点向量  $U$  反求出控制点  $P_i$ , 然后通过给定参数  $u$  求解出 B 样条基函数, 最后根据公式 (4-25) 求解出 B 样条曲线上的点; 另一种是将三次 B 样条曲线作为空间过渡轨迹, 从而实现空间位置、速度和加速度的平滑过渡, 在这种情况下, 求解过程相对较容易, 一般是根据轨迹过渡区域信息设定控制点  $P_i$  和构造节点向量  $U$ , 然后通过给定参数  $u$  求解出 B 样条基函数, 进而求出 B 样条曲线位置点。

### (2.1) 给定控制点 $P_i$ , 求解三次准均匀 B 样条曲线

已知  $n$  个控制点  $P_0, P_1, \dots, P_{n-1}$ , 现构造三次 B 样条曲线的节点向量  $U$ , 根据 B 样条曲线的节点向量最大下标必须满足  $m=n+p+1, m$  也称为节点向量的长度。为了满足 B 样条曲线的性质二, 使节点向量首末端点为  $p+1$  阶重点。中间节点均匀分布, 这些节点等间距分布在  $[0, 1]$  的区间范围内, 除去首末  $2p$  个多余重点, 又因为是三次 B 样条曲线,  $p$  的值为 3, 所以中间节点的长度为  $n-p+1=n-2$ , 那么中间节点子区间的间距为  $1/(n-2)$ , 最终构造的节点向量如下:

$$u_i = \begin{cases} 0 & i = 0, 1, 2, 3 \\ 1 & i = n+1, n+2, n+3, n+4 \\ u_{i-1} + 1/(n-2) & 4 \leq i \leq n \end{cases} \quad (4-26)$$

根据节点向量的分布, 将此类 B 样条曲线称为准均匀 B 样条曲线。有了节点向量后, 给定参数  $u$ , 首先判断  $u$  位于的节点区间  $[u_i, u_{i+1}]$ , 进而根据公式 (4-23) 求解出在此区间内的非零 B 样条基函数并且至多有 4 个非零, 最后将控制点  $P_i$  和

非零 B 样条基函数代入公式 (4-25) 可直接求出给定参数  $u$  对应的三次 B 样条曲线上的点。

### (2.2) 给定示教点 $D_i$ , 求解三次非均匀 B 样条曲线

已知  $n$  个示教点  $D_0, D_1, \dots, D_{n-1}$ , 将  $D_0$  和  $D_{n-1}$  作为三次 B 样条曲线的首末点,  $D_1, D_2, \dots, D_{n-2}$  作为曲线的中间点, 可将样条曲线分为  $n-1$  段, 对于三次 B 样条曲线, 每一段由  $p+1=4$  个控制点所影响, 且相邻两段曲线之间共用  $p=3$  个控制点, 那么可以推算出控制点的个数为  $n-1+3=n+2$  个, 也就是说总共有  $n+2$  个控制点需要反求, 进而确定节点向量的最大下标为  $m=n+5$ 。为了满足 B 样条曲线性质二, 构造节点向量首末端点为 4 阶重点, 即  $u_0=u_1=u_2=u_3=0, u_{n+2}=u_{n+3}=u_{n+4}=u_{n+5}=1$ , 节点向量中间点  $u_4, u_5, \dots, u_{n+1}$  采用累积弦长法<sup>[58]</sup>进行构造, 累积弦长法构造节点向量的具体步骤是: 首先分别求取相邻两示教点之间的距离  $|D_{i+1} - D_i|$  并将所有弦长进行累积求和, 然后依次以每段弦长占累积弦长和的比例系数来决定节点向量的分布。最终求得节点向量表达式如下:

$$u_i = \begin{cases} 0 & i = 0, 1, 2, 3 \\ 1 & i = n+2, n+3, n+4, n+5 \\ u_{i-1} + |D_{i-3} - D_{i-4}| / \sum_{j=0}^{n-2} |D_{j+1} - D_j|, & 4 \leq i \leq n+1 \end{cases} \quad (4-27)$$

将节点向量  $u_3, u_4, \dots, u_{n+2}$  和示教点  $D_0, D_1, \dots, D_{n-1}$  代入公式 (4-25) 联立可得  $n$  个方程:

$$C(u_{i+3}) = \sum_{j=i}^{i+3} N_{j,3}(u_{i+3})P_j = D_i \quad i=0,1,2,\dots,n-1 \quad (4-28)$$

而三次非均匀 B 样条  $n$  个示教点对应有  $n+2$  个控制点, 所以还需要构造两个方程, 文献<sup>[58]</sup>给出了三种边界条件来补充剩余的两个递补方程: 切矢条件、自由端点条件和闭曲线条件。本文采用自由端点条件作为边界条件, 在条件的特点在于端点处具有零曲率且不受力矩作用, 递补方程为:

$$\begin{cases} P_0 = P_1 \\ P_{n+1} = P_n \end{cases} \quad (4-29)$$

根据以上分析后, 联立  $n+2$  个方程, 可以很容易写出控制点求解方程组的矩阵形式:

$$M \begin{bmatrix} P_0 \\ P_1 \\ \vdots \\ P_n \\ P_{n+1} \end{bmatrix} = \begin{bmatrix} 0 \\ D_0 \\ \vdots \\ D_{n-1} \\ 0 \end{bmatrix} \quad (4-30)$$

其中  $M$  为  $n+2$  阶系数矩阵，如下所示：

$$M = \begin{bmatrix} 1 & -1 & & & & & \\ & 1 & & & & & \\ N_{1,3} & N_{2,3} & N_{3,3} & N_{4,3} & & & \\ & \ddots & \ddots & \ddots & \ddots & & \\ & & N_{n-2,3} & N_{n-1,3} & N_{n,3} & N_{n+1,3} & \\ & & & & 1 & & \\ & & & & -1 & 1 & \end{bmatrix} \quad (4-31)$$

由公式 (4-30) 和公式 (4-31) 可解出  $n+2$  个控制点。现已知节点向量  $U$  和控制点  $P_i$ ，那么与 (2.1) 中求解过程类似，给定参数  $u$  后，就可以求得三次非均匀 B 样条曲线上的点。

### (3) 曲线积分及参数-弧长反向拟合<sup>[59]</sup>

现在已经求解出了三次 B 样条曲线，要在样条曲线上进行速度规划，还需求解出曲线长度  $L(u)$ ，由于曲线表达式复杂，直接通过定积分方式求解难度较大，文献<sup>[60]</sup>提出了用复合求积法代替定积分方式求解样条曲线长度。为了提高积分误差精度，本文中采用辛普森积分法，辛普森积分可以达到很高的误差精度，也容易用计算机编程实现。假设区间  $[a, b]$  被划分为  $n$  等分，步长  $h=(b-a)/n$ ，分割点为  $x_k = a + kh$ ， $k=0, 1, \dots, n$ 。辛普森积分公式如下所示：

$$L(u) = \int_a^b C'_n(x) dx \approx h \sum_{k=0}^{n-1} \frac{1}{6} [C'_n(x_k) + 4C'_n(x_{k+\frac{1}{2}}) + C'_n(x_{k+1})] \quad (4-32)$$

假设积分收敛误差  $e$ ，当辛普森积分的  $n$  阶和  $2n$  阶差值小于等于积分收敛误差  $e$  时，即满足如下公式：

$$\left| \int_a^b C'_{2n}(x) dx - \int_a^b C'_n(x) dx \right| \leq e \quad (4-33)$$

此时再对区间  $[a, b]$  进行细分一次，将  $4n$  阶的辛普森积分结果作为最终的曲线积分长度，如下所示：

$$L_{[a,b]} = \int_a^b C'_{4n}(x) dx \quad (4-34)$$

因为空间 B 样条曲线是关于参数  $u$  的表达式，在底层速度规划后，根据插补时刻得到的是标量位移，所以首先要通过位移长度求得参数  $u$ ，然后将参数代入公式 (4-28) 求取 B 样条曲线空间位置点，参数-弧长反向拟合其实质是求解  $L(u)$  的反函数  $u(L)$ ，在对曲线区间进行细分的基础上，在第  $j$  个细分区间  $u \in [u_j, u_{j+1}]$  采用三次厄米特多项式拟合函数  $u_j(L)$ ，表达式如下：

$$u_j(L) = a + b(L - L_j) + c(L - L_j)^2 + d(L - L_j)^2(L - L_{j+1}) \quad (4-35)$$

其中  $a, b, c, d$  为待定系数， $L \in [L_j, L_{j+1}]$ ， $L$  和  $L_{j+1}$  为区间  $[0, u_j]$  和  $[0, u_{j+1}]$  的曲线积分长度。

将  $L_j$  和  $L_{j+1}$  代入公式 (4-35) 只能得到两个方程，又  $L(u)$  和  $u(L)$  互为反函数，根据定义反函数的导数为原函数导数的倒数，这样又可得到两个方程，如下所示：

$$\begin{cases} u_j(L_j) = u_j \\ u_j(L_{j+1}) = u_{j+1} \\ u'_j(L_j) = \frac{1}{L'(u_j)} \\ u'_j(L_{j+1}) = \frac{1}{L'(u_{j+1})} \end{cases} \quad (4-36)$$

根据公式 (4-36) 可求得待定系数  $a, b, c, d$ ，如下所示：

$$\begin{cases} a = u_j \\ b = u'_j \\ c = \frac{(u_{j+1} - u_j) - u'_j(L_{j+1} - L_j)}{(L_{j+1} - L_j)^2} \\ d = \frac{u'_{j+1} + u'_j}{(L_{j+1} - L_j)^2} - \frac{2(u_{j+1} - u_j)}{(L_{j+1} - L_j)^3} \end{cases} \quad (4-37)$$

本文在构造三次非均匀 B 样条曲线时，使控制点  $P_0 = P_1$  和  $P_n = P_{n+1}$ ，根据 B 样条曲线导数性质可知，在节点向量 0 和 1 处的一阶导数为 0，在公式 (4-36) 中，曲线导数表达式出现在分母，所以在首末两个细分区间不能采用以上方式进行拟合，直接采用直线拟合的方式。由于进行曲线积分参数细分为  $4n$  阶，因此确定需进行直线拟合的区间分别为  $[0, 1/4n]$  和  $[1-1/4n, 1]$ ，表达式如下所示：

$$\begin{cases} u_1(L) = \frac{1}{4nL_1} L, \quad L \in [0, L_1] \\ u_{4n}(L) = \frac{1}{4n(L_{4n+1} - L_{4n})} L + \frac{(4n-1)L_{4n+1} - 4nL_{4n}}{4n(L_{4n+1} - L_{4n})}, \quad L \in [L_{4n}, L_{4n+1}] \end{cases} \quad (4-38)$$

#### (4) 三次样条自适应可控速度规划

三次非均匀 B 样条曲线是空间自由曲线，在样条曲线上存在曲率极值点，由于系统存在最大法向加速度  $A_{\max}$ ，机器人末端执行器通过这些曲率极值点时必须要进行限速处理<sup>[61]</sup>，空间曲线曲率计算公式如下：

$$g(u) = \frac{|C'(u) \times C''(u)|}{|C'(u)|^3} \quad (4-39)$$

由于 B 样条曲线表达式较复杂，直接通过  $g'(u)=0$  求取比较繁琐，工程上常用的方法是将参数区间 $[0,1]$ 进行离散，分别求得离散点对应曲率，然后查找曲率极值点。求得了曲率极值点后，就可通过曲率限制速度的公式：

$$V_{\text{limit}} = \sqrt{A_{\max} \rho} \quad (4-40)$$

求出曲率极值点的最大速度  $V_{\text{limit}}$ ，上式中  $\rho$  为曲率半径，大小为曲率的倒数  $1/g(u)$ 。

如果用户设定的运行速度  $v$  大于  $V_{\text{limit}}$ ，此时系统自适应忽略用户设定的运行速度，在曲率极值点处进行分段，分成两段分别进行速度规划，以此类推，如果有  $n$  个速度限制点，那么需要分成  $n+1$  段。在进行第  $i$  段速度规划时，由前面介绍的 S 型速度规划器可知，规划后，第  $i$  段的末速度有可能小于末速度，即  $V_{\text{limit}}$ ，那么进行  $i+1$  段规划的时候，初始速度就不能采用通过曲率极值点求出的  $V_{\text{limit}}$ ，而是将上一段规划后的末速度作为当前段规划的初始速度。

#### (5) 三次 B 样条整体规划流程

首先判断用户设定的是示教点还是控制点，如果是控制点，那么根据 (2.1) 小节的求解方式进行三次准均匀 B 样条曲线的构造；反之如果是示教点，则根据 (2.2) 小节的求解步骤构造三次非均匀 B 样条曲线。其次，进行三次 B 样条曲线积分及曲线反向拟合，求出曲线参数关于曲线长度的函数  $L(u)$  及其反函数  $u(L)$ 。最后，根据系统最大法向加速度查找出三次 B 样条曲线上的限速点，如果存在限速点，并在限速点处将曲线进行分段，对每一段进行 S 型速度规划，规划完成后将标量规划结果转换成空间轨迹插补点序列。三次 B 样条轨迹规划整体规划流程如图 4-4 所示。

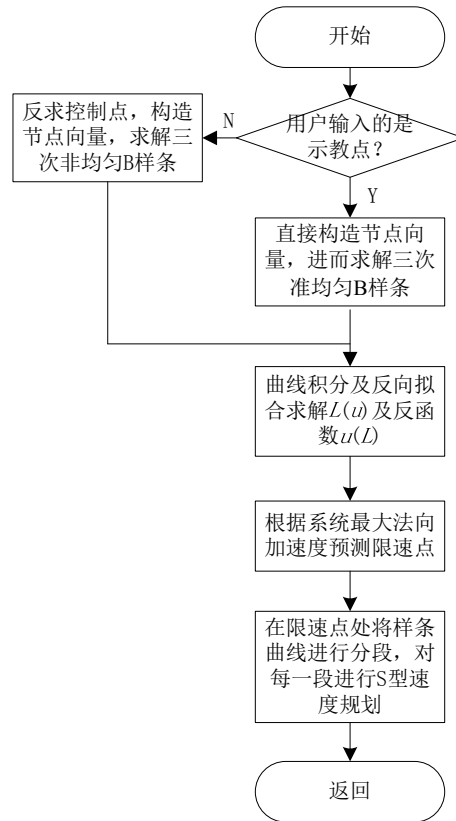


图 4-4 三次 B 样条整体规划流程图

Fig. 4-4 The total planning flowchart of cubic B-spline

## 4.2 基于三次 B 样条的空间轨迹过渡

在笛卡尔空间轨迹规划中, 不同轨迹之间的连接处曲率不连续。传统的方法是机器人末端执行器通过轨迹连接处时降速为零, 从而平滑通过以减少刚体的冲击。由于直线-直线连续轨迹处于同一平面中, 在不影响加工精度的条件下, 可在连接处采用圆弧进行过渡, 从而避免在连接处速度降为零, 大大地提高了机器人的工作效率, 但这种方法并不通用, 如直线-圆弧可能不处于同一平面。本文采用空间三次准 B 样条作为过渡轨迹, 灵活地解决了空间轨迹连续运动平滑过渡的问题, 如空间直线-直线过渡、空间直线-圆弧和空间三次非均匀 B 样条-三次非均匀 B 样条过渡。

### 4.2.1 直线-直线过渡

如图 4-5 所示, 空间两条直线  $OE$  和  $OF$  相交于点  $O$ , 首先确定安全过渡区, 假设安全过渡区是以  $O$  为圆心,  $OD$  为半径的圆弧内, 圆弧半径为  $R$ , 其中  $R = L_{OA}$



$= L_{OD}$ 。  $A$  和  $D$  点的坐标表示如下：

$$\begin{aligned} \begin{bmatrix} x_A \\ y_A \\ z_A \end{bmatrix} &= \begin{bmatrix} x_O \\ y_O \\ z_O \end{bmatrix} + \frac{R}{L_{OE}} \begin{bmatrix} x_E - x_O \\ y_E - y_O \\ z_E - z_O \end{bmatrix} \\ \begin{bmatrix} x_D \\ y_D \\ z_D \end{bmatrix} &= \begin{bmatrix} x_O \\ y_O \\ z_O \end{bmatrix} + \frac{R}{L_{OF}} \begin{bmatrix} x_F - x_O \\ y_F - y_O \\ z_F - z_O \end{bmatrix} \end{aligned} \quad (4-41)$$

在线段  $OA$  和  $OD$  上分别取点  $B$  和  $C$ ，可以是线段上除首末点以外的任意一点，本文直接取线段的中点。 $B$  和  $C$  点的坐标表示如下：

$$\begin{aligned} \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} &= \begin{bmatrix} x_O \\ y_O \\ z_O \end{bmatrix} + \frac{R}{2L_{OE}} \begin{bmatrix} x_E - x_O \\ y_E - y_O \\ z_E - z_O \end{bmatrix} \\ \begin{bmatrix} x_C \\ y_C \\ z_C \end{bmatrix} &= \begin{bmatrix} x_O \\ y_O \\ z_O \end{bmatrix} + \frac{R}{2L_{OF}} \begin{bmatrix} x_F - x_O \\ y_F - y_O \\ z_F - z_O \end{bmatrix} \end{aligned} \quad (4-42)$$

以  $A$ 、 $B$ 、 $C$  和  $D$  四点作为控制点可确定一条三次准均匀 B 样条曲线，首先构造节点向量，由于控制点个数为 4，又样条曲线次数  $p=3$ ，所以节点向量最大下标  $m=7$ ，根据三次准均匀 B 样条性质，节点向量首末端点为  $p+1=4$  阶重点，中间节点均匀分布，那么最终构造出的节点向量  $U=[0, 0, 0, 0, 1, 1, 1, 1]$ 。现已知控制点和节点向量，就可以求解出三次准均匀 B 样条曲线，具体的求解方法参考 4.1.2 小节。

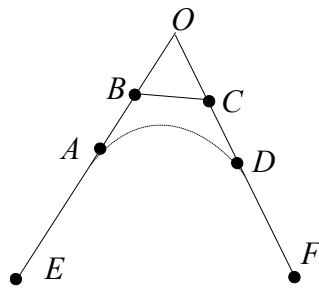


图 4-5 直线-直线过渡

Fig. 4-5 Linear-linear transition

上面得到直线-直线过渡的轨迹，进而在轨迹上进行速度规划，将轨迹分为三段：直线、过渡区轨迹（三次准均匀 B 样条）和直线，通过曲率极值点和最大向心加速度  $A_{\max}$  来判断过渡区轨迹上是否存在限速点  $V_{\text{limit}}$ ，如果存在限速点，还需

要在限速点处将过渡区轨迹进行分段，最后进行分段轨迹 S 型速度规划。如图 4-6 所示，考虑过渡区轨迹上不存在限速点的情况，用户可控制每一段轨迹的初始速度、末速度和运行速度的设定。将过渡区轨迹的初始速度  $v_s$  作为第一段直线的末速度，将过渡区轨迹的末速度  $v_e$  作为第二段直线速度规划的初始速度。由第 3 章介绍的 S 型速度规划器可知，在第  $i$  段规划完成后，末速度有可能降低，在进行第  $i+1$  段规划时，系统自适应地选择第  $i$  段规划后的末速度作为  $i+1$  段的初始速度，直到每一度都规划成功后，再将标量规划结果转换到空间，就可得到直线-直线过渡轨迹的等周期插补点信息。如果过渡区轨迹上存在限速点，只需在限速点处进行分段，处理方法类似。

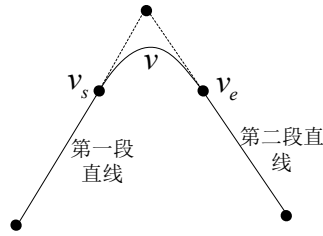


图 4-6 直线-直线过渡速度规划

Fig. 4-6 Speed planning of linear-linear transition

#### 4.2.2 直线-圆弧过渡

如图 4-7 所示，空间直线  $AB$  和圆弧  $O-BCD$  相交于点  $B$ ，圆弧所在平面记为  $M_1$ ，直线有可能不在圆弧所在平面，记直线所在平面为  $M_2$ 。设定安全过渡区是以  $B$  为球心， $r$  为半径的球体，安全过渡区分别与直线和圆弧相交于  $E$  和  $C$  两点，即  $r = L_{BE} = L_{BC}$ ，取线段  $BE$  上的中点  $F$ ， $L_{BF} = r/2$ ，圆弧上  $C$  点处的切线上取点  $G$ ， $L_{CG} = r$ ，以点  $E$ 、 $F$ 、 $G$  和  $C$  点为控制点生成三次准均匀 B 样条，且曲线与控制多边形的  $EF$  和  $CG$  相切，所以直线和圆弧之间平滑过渡。 $E$  和  $F$  的坐标表示如下：

$$\begin{aligned} \begin{bmatrix} x_E \\ y_E \\ z_E \end{bmatrix} &= \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} + \frac{r}{L_{AB}} \begin{bmatrix} x_A - x_B \\ y_A - y_B \\ z_A - z_B \end{bmatrix} \\ \begin{bmatrix} x_F \\ y_F \\ z_F \end{bmatrix} &= \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} + \frac{r}{2L_{AB}} \begin{bmatrix} x_A - x_B \\ y_A - y_B \\ z_A - z_B \end{bmatrix} \end{aligned} \quad (4-43)$$

$C$  和  $G$  点的坐标求取相对复杂，首先要求得圆弧  $O-BCD$  的夹角  $\theta$  和  $OB$  与  $OG$  的夹角  $\theta_1$ ，如下：

$$\theta = 2\arcsin \frac{r}{2R} \quad (4-44)$$

$$\theta_1 = \theta - \arctan \frac{r}{2R} \quad (4-45)$$

在新的圆弧坐标系中，求得  $C$  和  $G$  点的坐标为：

$$\begin{cases} U_C = R \cos \theta \\ V_C = R \sin \theta \\ W_C = 0 \end{cases} \quad (4-46)$$

$$\begin{cases} U_G = \sqrt{r^2/4 + R^2} \cos \theta_1 \\ V_G = \sqrt{r^2/4 + R^2} \sin \theta_1 \\ W_G = 0 \end{cases}$$

圆弧坐标系的建立见 4.1.1 小节，这里就不在介绍，将新坐标系中  $C$  和  $G$  点的坐标乘上转换矩阵就可得到原坐标系中的坐标，分别记为  $(x_C, y_C, z_C)$  和  $(x_G, y_G, z_G)$ 。

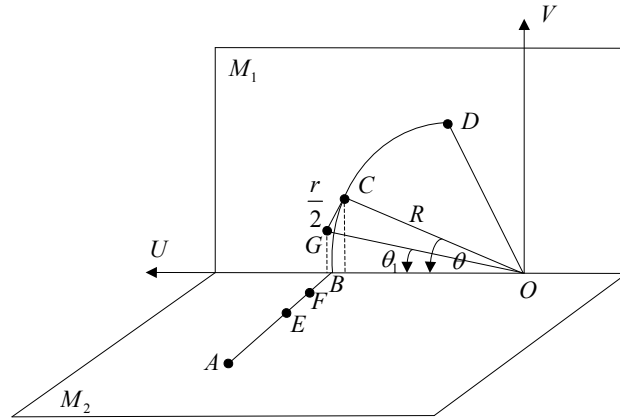


图 4-7 直线-圆弧过渡

Fig. 4-7 Linear-circle transition

上面已经得到了 4 个控制点  $E$ 、 $F$ 、 $G$  和  $C$  的坐标值，节点向量的构造方式与直线-直线过渡中一致，为  $U = [0, 0, 0, 0, 1, 1, 1, 1]$ 。已知控制点和节点向量，就可以构造出三次准均匀 B 样条曲线。

同直线-直线过渡轨迹规划类似，将直线-圆弧过渡的整条轨迹分为三段：直线、过渡区轨迹（三次准均匀 B 样条）和圆弧，如果过渡区轨迹存在限速点，还需在限速点处将过渡区轨迹进行细分，进而对每一段轨迹设置初始速度、运行速度、

末速度和轨迹长度等参数进行 S 型速度规划，规划完成后就可得到直线-圆弧过渡轨迹的等周期插补点信息。

### 4.2.3 三次非均匀 B 样条-三次非均匀 B 样条过渡

如图 4-8 所示，两条独立的三次非均匀 B 样条曲线相交于点  $O$ ，分别记为样条曲线一和样条曲线二，设定安全过渡区是以点  $O$  为球心， $R$  为半径的空间球体，球体分别与样条曲线一和样条曲线二交于点  $A$  和点  $B$ 。空间球体的表达式为：

$$(x-x_o)^2 + (y-y_o)^2 + (z-z_o)^2 = R^2 \quad (4-47)$$

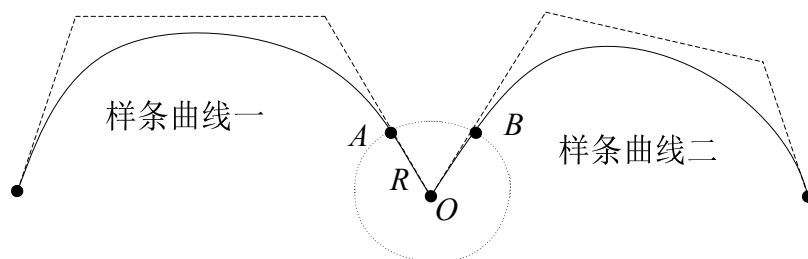


图 4-8 三次非均匀 B 样条-三次非均匀 B 样条过渡

Fig. 4-8 Nonuniform cubic B-spline - nonuniform cubic B-spline transition

安全过渡区放大图如图 4-9 所示，在过渡点  $A$  和  $B$  处分别求出样条曲线一和样条曲线二的切线，进而在切线方向上取点  $C$  和点  $D$ ，满足  $L_{AC} = L_{BD} = R/2$ ，以  $A$ 、 $B$ 、 $C$  和  $D$  作为控制点，生成过渡区曲线（三次准均匀 B 样条曲线），根据三次准均匀 B 样条曲线的性质可知，过渡区曲线分别与  $AC$  和  $BD$  相切，所以过渡区曲线与样条曲线一和样条曲线二在连接点  $A$  和  $B$  处是连续的。

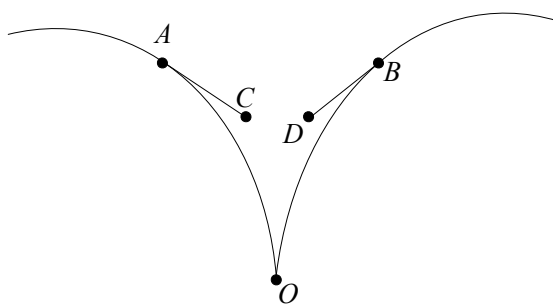


图 4-9 安全过渡区放大图

Fig. 4-9 The zoom figure of safety transition

#### (1) 过渡曲线控制点的求解

由于 B 样条曲线没有确定的解析表达式，所以不能使用联立方程组的方法来

求解  $A$  和  $B$  点的空间坐标，只能采用逼近的方式在三次非均匀 B 样条曲线节点向量区间  $[0, 1]$  内查找近似点。本文采用二分查找算法，该算法查找效率高，完全满足设计要求，设定位置误差  $POS\_ERR$  为 0.01 毫米，以求取样条曲线一上  $A$  点坐标为例，假设  $O$  点为样条曲线一的尾部，具体操作流程如下：

(a) 设置查找区间为  $[u_a, u_b]$ ，初始化  $u_a = 0$ ， $u_b = 1$ ，执行步骤 (b)。

(b) 取  $u_a$  和  $u_b$  的中点  $u_c = (u_a + u_b) / 2$ ，求取节点参数  $u_c$  点对应的 B 样条曲线位置点坐标  $P$ ，代入公式 (4-47) 左边，得到  $Res$ 。首先判断  $Res - R^2$  的绝对值是否小于  $POS\_ERR$ ，如果小于，那么查找完成， $P$  就是  $A$  的坐标，否则执行步骤 (c)。

(c) 更新查找区间，如果  $Res - R^2$  大于 0，那么使  $u_a = u_c$ ；反之  $u_b = u_c$ ，执行步骤 (b)。

点  $O$  为样条曲线二的首部，点  $B$  的坐标求解方式与上面类似，只是二分查找时查找区间更新不同，这里不在赘述。

点  $C$  和  $D$  的空间坐标求取。首先求取点  $A$  和  $B$  处的切线向量，即三次非均匀 B 样条曲线在这两点处的导数，并将其单位化，记为  $\mathbf{n}_A$  和  $\mathbf{n}_B$ ，根据空间直线关系很容易求得  $C$  和  $D$  的坐标，表示如下：

$$\begin{aligned} \begin{bmatrix} x_C \\ y_C \\ z_C \end{bmatrix} &= \begin{bmatrix} x_A \\ y_A \\ z_A \end{bmatrix} + \frac{R}{2} \mathbf{n}_A \\ \begin{bmatrix} x_D \\ y_D \\ z_D \end{bmatrix} &= \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} - \frac{R}{2} \mathbf{n}_B \end{aligned} \quad (4-48)$$

## (2) 过渡曲线节点向量的构造

过渡曲线节点向量的构造方法与直线-直线过渡和直线圆弧过渡中节点向量的构造方法一样，四个控制点确定一段三次 B 样条曲线，节点向量首末 4 阶重点使得过渡曲线（三次准均匀 B 样条曲线）分别与控制多边形的  $AC$  和  $BD$  边相切，最终构造出的节点向量为  $U = [0, 0, 0, 0, 1, 1, 1, 1]$ 。有了控制点和节点向量，给定节点参数，就可求得对应过渡曲线上的空间位置。

## (3) 自适应可控轨迹规划

将三次非均匀 B 样条-三次非均匀 B 样条过渡依次分为三段轨迹：三次非均匀 B 样条、三次准均匀 B 样条和三次非均匀 B 样条，在 4.1.2 小节中，已经详细分析介绍了一条独立的三次 B 样条曲线的轨迹规划方法。首先，通过曲线积分及曲线反向拟合，求出曲线参数关于曲线长度的函数  $L(u)$  及其反函数  $u(L)$ ，从而很容易

求得对应节点区间内样条曲线的长度，用户控制输入每段样条曲线通过示教点的标量速度、及示教点之间的期望的标量运行速度。其次，根据系统所能承受的最大向心加速度  $A_{\max}$  求取每一条轨迹上的限速点，进而在限速点处自适应进行分段，这里为了叙述方便，假设样条曲线一和样条曲线二只有两个示教点且曲线上各存在一个限速点，如图 4-10 所示。

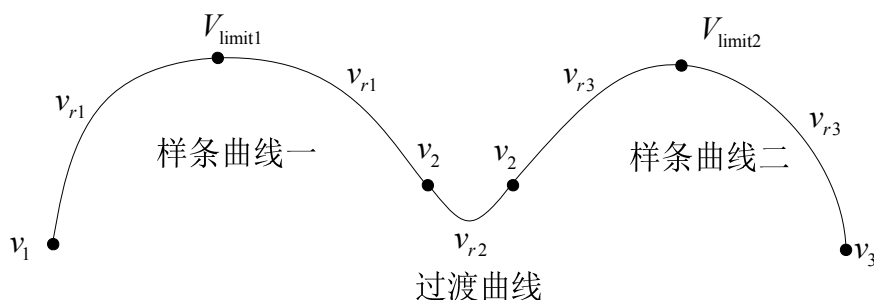


图 4-10 自适应可控轨迹规划

Fig. 4-10 Adaptive and controlled trajectory planning

上图中两个限速点  $V_{\text{limit1}}$  和  $V_{\text{limit2}}$  把整条轨迹分成了五段，分别依次对每一段轨迹调用 S 型速度规划器进行速度规划，在第  $i$  段规划完成后，末速度有可能降低，在进行第  $i+1$  段规划时，系统自适应地选择第  $i$  段规划后的末速度作为  $i+1$  段的初始速度，直到每一段都规划完成后，得到的是标量位移、速度和加速度的规划结果，还需将标量规划结果转换到笛卡尔空间，将标量位移规划结果直接通过反函数  $u(L)$  可得到节点参数，进而可得到曲线空间位置插补点信息，标量速度和加速度规划结果乘上对应位置点上的方向向量可得到位置点上的空间速度和加速度插补点信息，自此完成整条曲线的轨迹规划。

### 4.3 本章小结

在本章中，基于第 3 章的自适应可控 S 型速度规划器，分析介绍了笛卡尔空间直线、圆弧和三次 B 样条曲线的自适应可控轨迹规划，并详细推导了轨迹规划中的算法计算流程，保证了机器人末端执行器在平滑运行。以三次准均匀 B 样条曲线作为空间过渡曲线，完成了空间直线-直线、空间直线-圆弧和三次非均匀 B 样条-三次非均匀 B 样条曲线的平滑过渡，满足曲线轨迹的曲率连续，避免了机器人末端执行器在进行空间多示教点连续运行速度时速度降为零的情况，提高了机器人的作业效率。

## 5 轨迹规划算法仿真与分析

上面研究的笛卡尔空间位置层轨迹规划算法是机器人运动控制系统中不可或缺的一部分，决定了机器人末端执行器的加工精度及效率。在机器人运动控制系统中使用这些算法，首先要用编程语言对算法进行编码。本文采用 C++ 编程语言，C++ 是一种广泛使用的计算机程序设计语言，其运行效率接近 C 语言，广泛用于工业领域的计算机编程，用 C++ 语言设计的程序具有跨平台性质（可运行在不同的操作系统上，如 Windows 和 Unix），根据这一性质，如果将轨迹规划算法使用 C++ 语言实现，那么可应用于不同操作系统平台的机器人运动控制系统。本文将算法抽象成接口，用户可见的只有参数输入和输出接口，算法实现细节对于用户不可见，使用起来非常方便，具有通用性。MATLAB 是一种用于算法开发、数据可视化、数据分析以及数值计算的高级技术计算语言和交互式环境，本文使用 MATLAB 对算法输出结果进行可视化分析与验证。

### 5.1 机器人正逆解算法仿真与分析

本文主要研究的是空间位置，其固定姿态不变，因此只对位置正逆解进行验证。设定位置空间一示教点位置  $pos=[300, 100, 400]$ ，单位为毫米（mm），通过调用运动学逆解算法求得关节空间关节解如表 5-1 所示，进而将这组关节角度作为运动学正解算法的输入，求得示教点位置  $pos1=[300.0032, 100.0012, 400.0039]$ ，根据结果可知  $pos1$  与  $pos$  在允许的误差范围内是相等的，即验证了算法的正确性。

表 5-1 逆解出的关节角度值

Table. 5-1 The joint angle values by inverse algorithm

关节 $i$	1	2	3	4	5	6
弧度	0.4354	-0.7729	0.7065	-1.4292	0.4401	1.4145

### 5.2 速度规划算法仿真与分析

由于梯形速度控制曲线和正弦型速度控制曲线算法实现较简单，且不是本文的研究重点，在本节中只对 S 型速度规划算法进行仿真分析和验证。S 型速度规划算法接口设计在 3.2.4 小节已经给出，这里不在赘述。

#### 5.2.1 给定运行速度的 S 型速度规划算法

如 3.2.2 小节，从给定运行速度的 S 型速度规划器的三种情况进行分析：运行

速度大于初始速度和末速度、运行速度小于初始速度和末速度和运行速度介于初始速度和末速度之间。 $v_s$  为初始速度,  $v_e$  为末速度,  $v$  为运行速度,  $v_{\max}$  为最大运行速度,  $L$  为运行位移,  $t$  为运行时间,  $a$  为系统最大加速度,  $J$  为系统所能承受的最大冲击, 位移和速度以毫米为单位, 时间以秒为单位, 本文后面出现同样的变量均用以上解释。

### (1) 运行速度大于初始速度和末速度

设置输入参数为:  $v_s=20\text{mm/s}$ ,  $v_e=30\text{mm/s}$ ,  $v=100\text{mm/s}$ ,  $L=1000\text{mm}$ ,  $a=300\text{mm/s}^2$ ,  $J=800\text{mm/s}^3$ , 输出结果如图 5-1 (a) 所示, 在运行位移足够长的情况下, 初始速度、运行速度和末速度都能够达到用户的期望值。将运行位移减短, 设置为  $L=50\text{mm}$ , 其它参数不变, 规划结果如图 5-1 (b) 所示, 为了满足 S 型速度曲线的特性, 运行速度要小于用户设定的  $100\text{mm/s}$ , 与 3.2.2 小节分析的一致, 证明了算法的正确性。

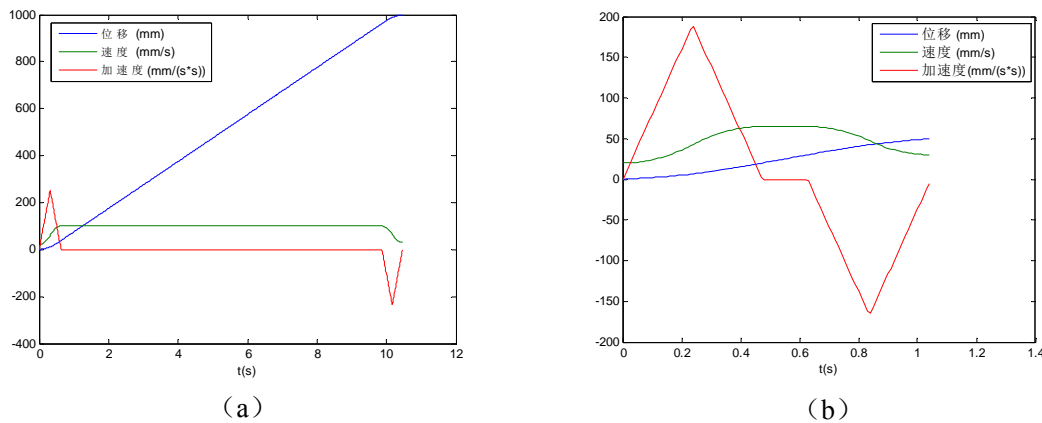


图 5-1 用户设定的运行速度高于初始速度和末速度

Fig. 5-1 Speed set by user is higher than the speed of first and last

### (2) 运行速度小于初始速度和末速度

设置输入参数为:  $v_s=200\text{mm/s}$ ,  $v_e=300\text{mm/s}$ ,  $v=100\text{mm/s}$ ,  $L=1000\text{mm}$ ,  $a=300\text{mm/s}^2$ ,  $J=800\text{mm/s}^3$ , 输出结果如图 5-2 (a) 所示, 同样, 运行位移足够长, 规划结果与期望一致, 将位移设定为  $L=250\text{mm}$ , 其它参数不变, 输出结果如图 5-2 (b) 所示, 规划后的运行速度要大于用户所设定的  $100\text{mm}$ , 与算法设计的原理一致。



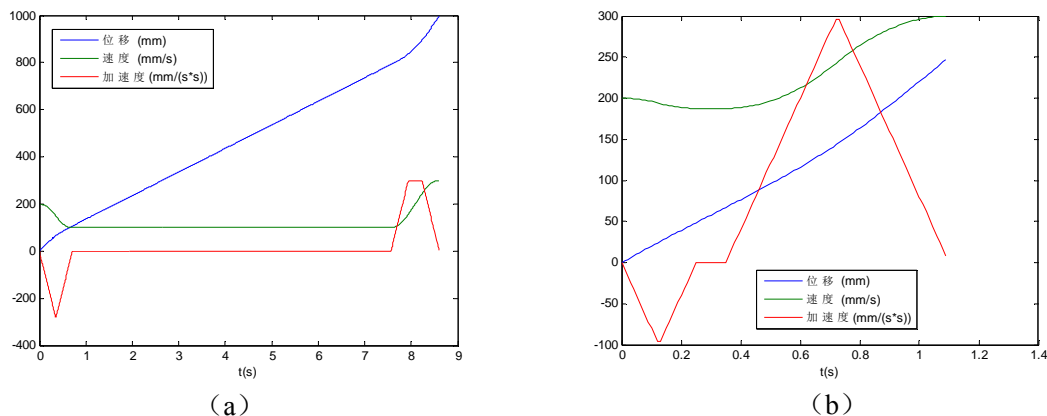


图 5-2 用户设定的运行速度低于初始速度和末速度

Fig. 5-2 Speed set by user is lower than the speed of first and last

### (3) 运行速度介于初始速度和末速度之间

设置输入参数为： $v_s=100\text{mm/s}$ ， $v_e=300\text{mm/s}$ ， $v=200\text{mm/s}$ ， $L=1000\text{mm}$ ， $a=300\text{mm/s}^2$ ， $J=800\text{mm/s}^3$ ，输出结果如图 5-3 (a) 所示，规划结果与期望一致。重新设定运行位移  $L=250\text{mm}$ ，调用速度规划器进行规划，如图 5-3 (b) 所示，由于运行位移太短，无法满足用户设定运行速度，又末速度大于初始速度，最终将剩余的匀速段速度插入到速度规划的末端。

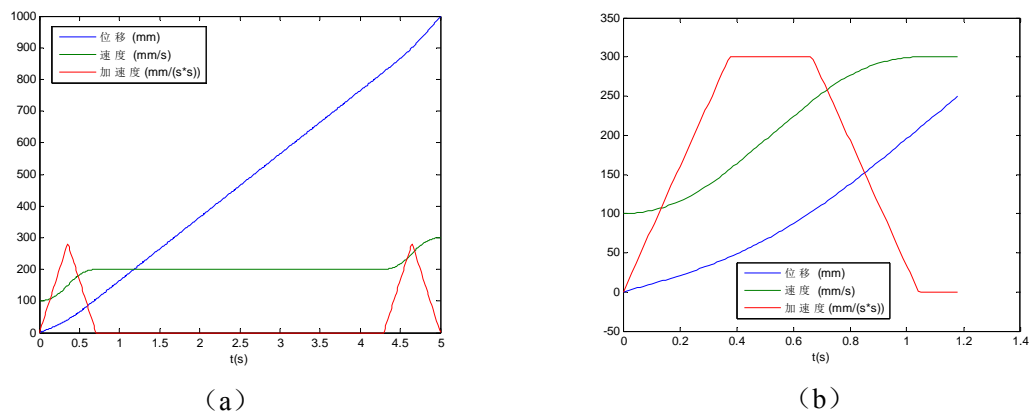


图 5-3 用户设定的运行速度介于初始速度和末速度之间

Fig. 5-3 Speed set by user is among the speed of first and last

## 5.2.2 给定运行时间的 S 型速度规划算法

给定运行时间的 S 型速度规划器输入参数与给定运行速度的 S 型速度规划器稍有区别，增加了用户设定的运行时间  $t$  和系统最大速度  $v_{\max}$ ，速度规划器在时间

层进行匹配, 相比后者, 限定条件更加苛刻。如 3.2.3 小节分析结果, 分为四种情况进行验证: 平均速度小于初始速度和末速度、平均速度大于初始速度和末速度、初始速度大于平均速度且平均速度大于末速度、初始速度小于平均速度且平均速度小于末速度。

#### (1) 平均速度比初始速度和末速度都小

设置输入参数为:  $v_s=150\text{mm/s}$ ,  $v_e=200\text{mm/s}$ ,  $t=10\text{s}$ ,  $v_{\max}=500\text{mm/s}$ ,  $L=1000\text{mm}$ ,  $a=300\text{mm/s}^2$ ,  $J=800\text{mm/s}^3$ , 平均速度  $v_{\text{aver}}=1000/10=100\text{mm/s}$ , 规划结果如图 5-4 所示, 满足 S 型速度控制曲线的特性, 与算法设计原理一致。

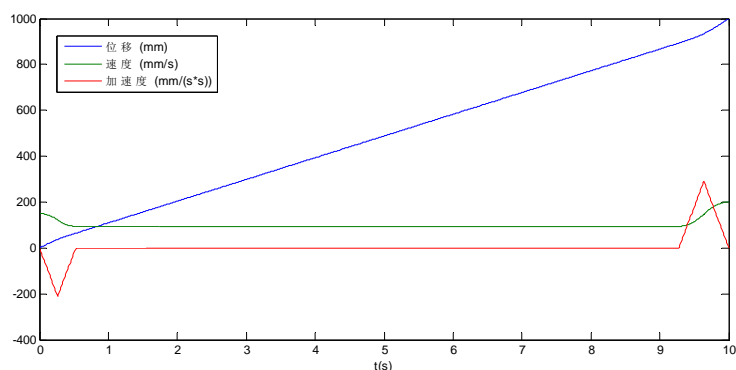


图 5-4 平均速度比初始速度和末速度都小

Fig. 5-4 Average speed lower than the speed of first and last

#### (2) 平均速度比初始速度和末速度都大

设置输入参数为:  $v_s=100\text{mm/s}$ ,  $v_e=200\text{mm/s}$ ,  $t=4\text{s}$ ,  $v_{\max}=500\text{mm/s}$ ,  $L=1000\text{mm}$ ,  $a=300\text{mm/s}^2$ ,  $J=800\text{mm/s}^3$ , 平均速度  $v_{\text{aver}}=1000/4=250\text{mm/s}$ , 规划结果如图 5-5 所示。

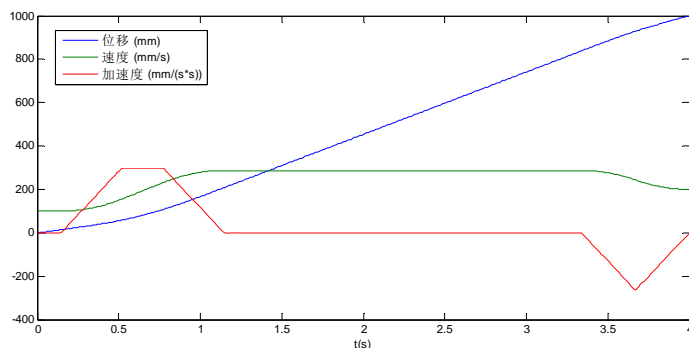


图 5-5 平均速度比初始速度和末速度都大

Fig. 5-5 Average speed higher than the speed of first and last

### (3) 初始速度大于平均速度且平均速度大于末速度

设置输入参数为:  $v_s=200\text{mm/s}$ ,  $v_e=100\text{mm/s}$ ,  $t=6\text{s}$ ,  $v_{\max}=500\text{mm/s}$ ,  $L=1000\text{mm}$ ,  $a=300\text{mm/s}^2$ ,  $J=800\text{mm/s}^3$ , 平均速度  $v_{\text{aver}}=1000/6=166.67\text{mm/s}$ , 规划结果如图 5-6 所示。

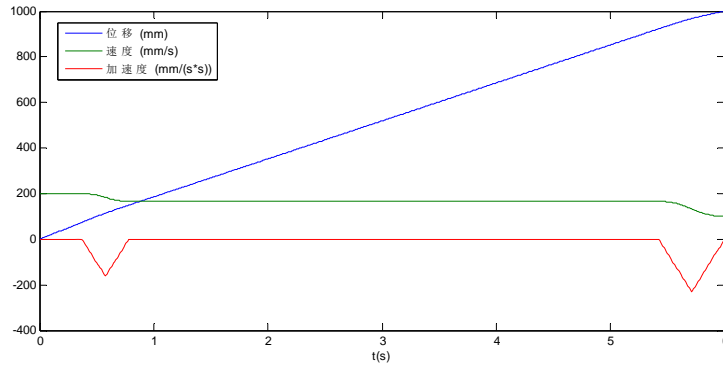


图 5-6 初始速度大于平均速度且平均速度大于末速度

Fig. 5-6 First speed higher than average speed and average speed higher than last speed

### (4) 初始速度小于平均速度且平均速度小于末速度

设置输入参数为:  $v_s=100\text{mm/s}$ ,  $v_e=250\text{mm/s}$ ,  $t=6\text{s}$ ,  $v_{\max}=500\text{mm/s}$ ,  $L=1000\text{mm}$ ,  $a=300\text{mm/s}^2$ ,  $J=800\text{mm/s}^3$ , 平均速度  $v_{\text{aver}}=1000/6=166.67\text{mm/s}$ , 规划结果如图 5-7 所示。

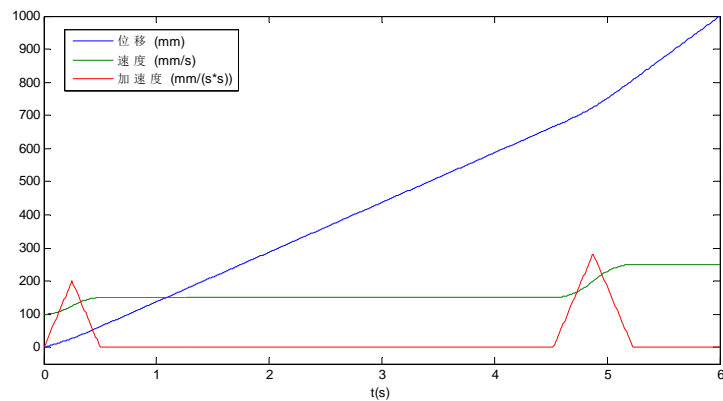


图 5-7 初始速度小于平均速度且平均速度小于末速度

Fig. 5-7 First speed lower than average speed and average speed lower than last speed

### 5.3 笛卡尔空间轨迹规划算法仿真与分析

由于直线-圆弧过渡轨迹规划涵盖了直线轨迹规划和圆弧轨迹规划，这里就不再单独对直线和圆弧进行仿真验证，这一节中主要对三次非均匀 B 样条曲线、直线-直线过渡、直线-圆弧过渡和三次非均匀 B 样条-三次非均匀 B 样条过渡进行仿真与分析。

#### 5.3.1 三次非均匀 B 样条轨迹规划算法

设定 4 个示教点  $A(300, 100, 400)$ ,  $B(400, 200, 450)$ ,  $C(450, 150, 350)$ ,  $D(420, -100, 420)$ ，单位为毫米 (mm)。生成的三次非均匀 B 样条曲线如图 5-8 (a) 所示，样条曲线的曲率如图 5-8 (b) 所示，说明在三次非均匀 B 样条曲线是一条光滑的曲线。

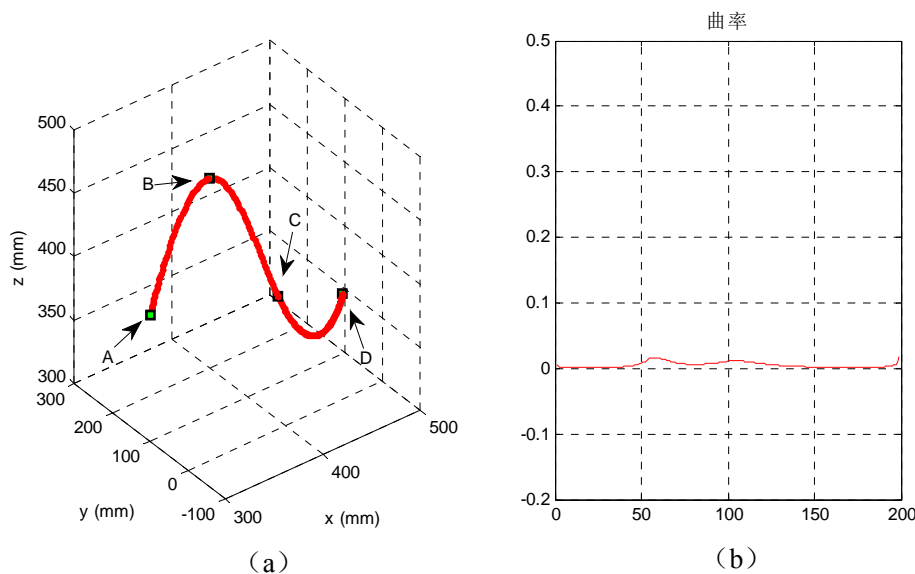


图 5-8 三次非均匀 B 样条曲线

Fig. 5-8 Nonuniform cubic B-spline

设定 4 个控制点  $A(130, 300, 200)$ ,  $B(200, 250, 100)$ ,  $C(300, 300, 350)$ ,  $D(300, 400, 500)$ ,  $E(350, 500, 300)$ ，单位为毫米 (mm)，生成的三次准均匀 B 样条曲线如图 5-9 (a) 所示，样条曲线的曲率如图 5-9 (b) 所示，说明在三次准均匀 B 样条曲线是一条光滑的曲线。

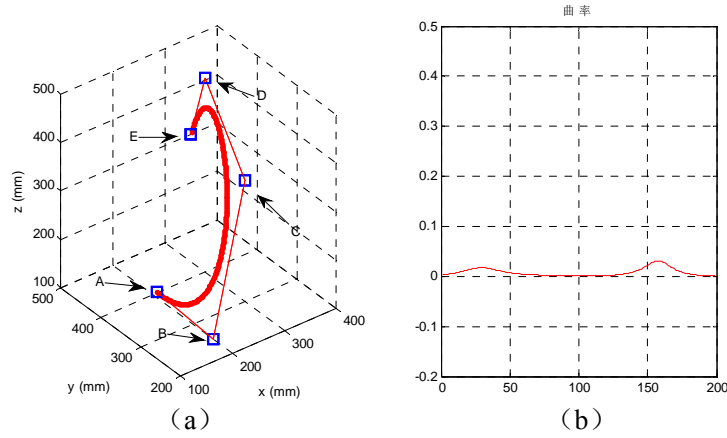


图 5-9 三次准均匀 B 样条曲线

Fig. 5-9 Quasi-uniform cubic B-spline

在图 5-8 (a) 上的三次非均匀 B 样条曲线上进行 S 型速度规划, 依次设定示教点之间的运行速度为  $v_1=120\text{mm/s}$ ,  $v_2=150\text{mm/s}$ ,  $v_3=110\text{mm/s}$ , 设定  $v_s=100\text{mm/s}$ ,  $v_e=110\text{mm/s}$ ,  $a=100\text{mm/s}^2$ ,  $J=200\text{mm/s}^3$ ,  $A_{\max}=500\text{mm/s}^2$ ,  $\text{minlen}=0.01\text{mm}$ ,  $\text{minlen}$  为曲线细分曲线精度。将样条曲线长度作为运行位移, 进行 S 型速度规划后的标量位移、速度和加速度曲线分别如图 5-10 (a)、(b) 和 (c) 所示, 横坐标轴为时间轴, 可以看出速度曲线平滑且呈 S 型。取为采样周期为  $0.01\text{s}$ , 将标量规划结果转换到空间, 如图 5-11 (b) 所示, 为三次非均匀 B 样条轨迹插补点。图 5-11 (a) 轨迹规划前的曲线, 可见插补后的轨迹与插补前的曲线完全重合, 验证了算法的正确性。空间位置、速度和加速度在  $x$ 、 $y$  和  $z$  轴方向上的分量与时间  $t$  的关系如图 5-12 (a) 列、(b) 列和 (c) 列所示, 由于三次 B 样条曲线是  $C^2$  连续的, 所以规划后插补点上的加速度各分量也是连续的。根据位置插补点逆解出的关节角位置与时间  $t$  的关系如图 5-13 所示。

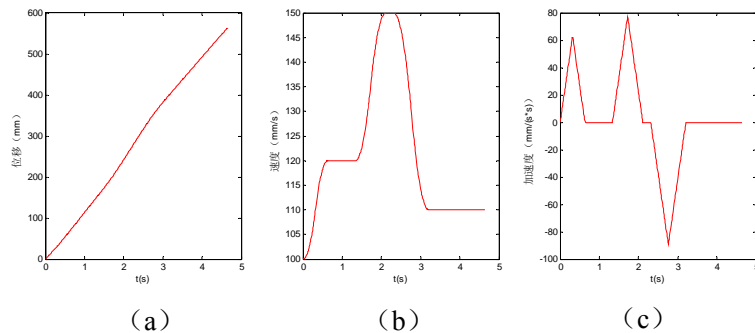


图 5-10 末端位置 S 型速度规划

Fig. 5-10 The S-type velocity planning results at end position

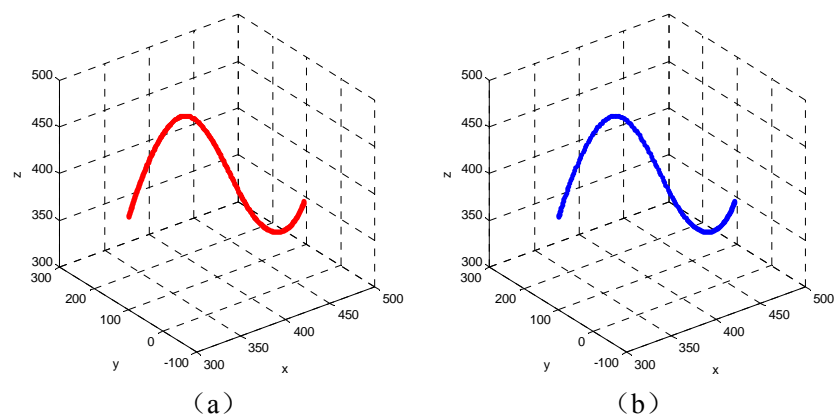


图 5-11 三次非均匀 B 样条轨迹插补点

Fig. 5-11 Nonuniform cubic B-spline trajectory interpolation points

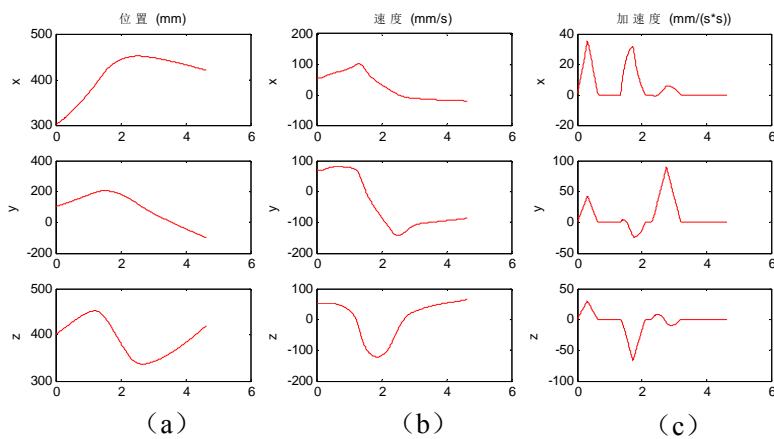
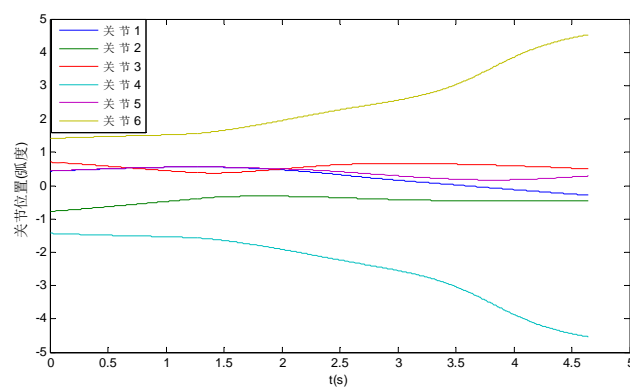
图 5-12 空间位置、速度和加速度的  $x$ 、 $y$  和  $z$  轴分量Fig. 5-12 Space position, speed, acceleration at axis  $x, y, z$ 

图 5-13 关节角位置

Fig. 5-13 The position of joints

### 5.3.2 直线-直线过渡轨迹规划算法

设定空间三点坐标  $E(300, 0, 400)$ ,  $O(400, 200, 450)$ ,  $F(300, 0, 500)$ , 直线  $EO$  和直线  $OF$  交于点  $O$ , 设定安全过渡区圆弧半径  $R$  长度为  $50\text{mm}$ ,  $v_s=0\text{mm/s}$ ,  $v_1=80\text{mm/s}$ ,  $v_{\text{mid}}=30\text{mm/s}$ ,  $v_e=0\text{mm/s}$ ,  $v_2=60\text{mm/s}$ ,  $a=100\text{mm/s}^2$ ,  $J=200\text{mm/s}^3$ ,  $\text{minlen}=0.01\text{mm}$ ,  $\text{minlen}$  为过渡曲线细分精度。  $v_1$  为直线  $EO$  上的运行速度,  $v_2$  为直线  $OF$  上的运行速度,  $v_{\text{mid}}$  为过渡区临界点上的速度, 过渡区内的运行速度设为  $v_t=v_{\text{mid}}+10=40\text{mm/s}$ , 轨迹完成 S 型速度规划后, 设置采样周期为  $0.01\text{s}$ 。整条轨迹插补点如图 5-14 所示, 图 5-14 (b) 为过渡区的放大图。末端位置 S 型速度规划结果如图 5-15 (a)、(b) 和 (c) 所示。空间位置、速度和加速度在  $x$ 、 $y$  和  $z$  轴方向上的分量与时间  $t$  的关系如图 5-16 (a) 列、(b) 列和 (c) 列所示。逆解后的关节角位置时间  $t$  的关系如图 5-17 所示。

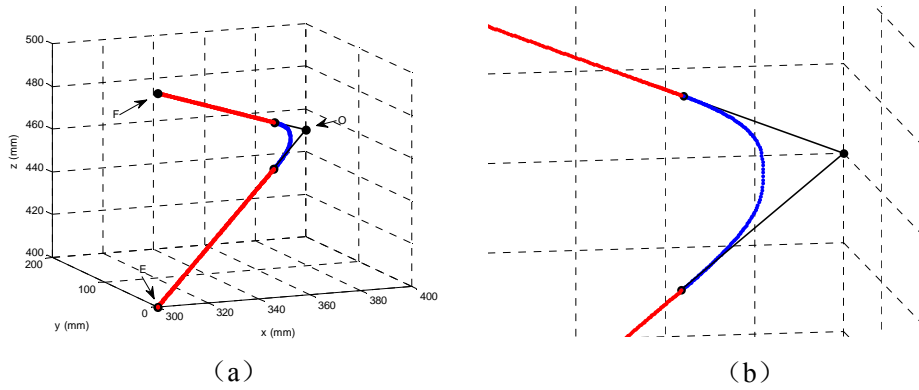


图 5-14 直线-直线过渡轨迹插补点

Fig. 5-14 Line-line transition trajectory interpolation points

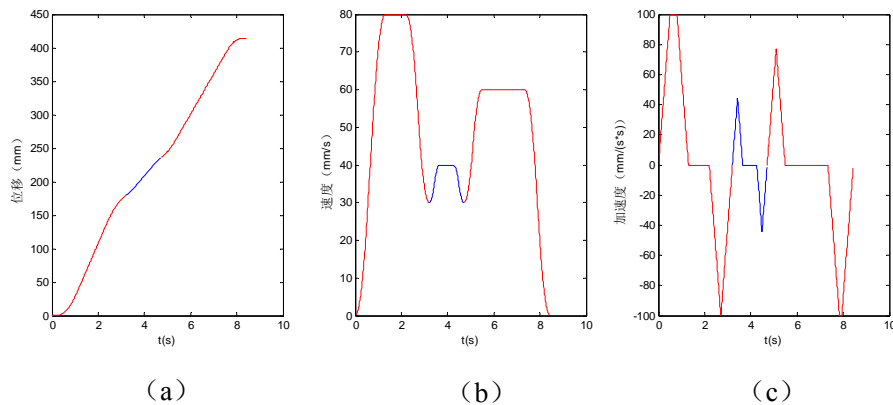


图 5-15 末端位置 S 型速度规划

Fig. 5-15 The S-type velocity planning results at end position

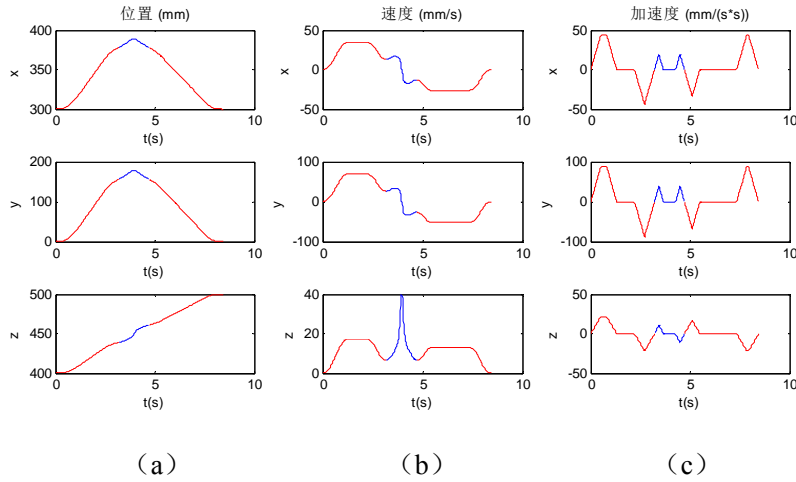
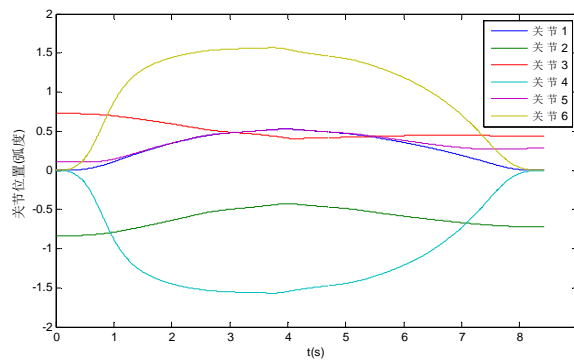
图 5-16 空间位置、速度和加速度的  $x$ 、 $y$  和  $z$  轴分量Fig. 5-16 Space position, speed, acceleration at axis  $x, y, z$ 

图 5-17 关节角位置

Fig. 5-17 The position of joints

### 5.3.3 直线-圆弧过渡轨迹规划算法

设定空间 4 个点  $A(300,100,400), B(300,150,350), H(350,50,360), D(400,150,450)$ , 直线  $AB$  与圆弧  $BHD$  交于点  $B$ , 设定安全过渡区球体半径  $R$  长度为 20mm,  $v_s=80\text{mm/s}$ ,  $v_1=100\text{mm/s}$ ,  $v_{\text{mid}}=50\text{mm/s}$ ,  $v_e=100\text{mm/s}$ ,  $v_2=120\text{mm/s}$ ,  $a=100\text{mm/s}^2$ ,  $J=200\text{mm/s}^3$ ,  $A_{\text{max}}=500\text{mm/s}^2$ ,  $\text{minlen}=0.01\text{mm}$ ,  $\text{minlen}$  为过渡曲线细分精度。 $v_1$  为直线  $AB$  上的运行速度,  $v_2$  为圆弧  $BHD$  上的运行速度,  $v_{\text{mid}}$  为过渡区临界点上的速度, 过渡区内的运行速度设为  $v_t=v_{\text{mid}}+10=60\text{mm/s}$ , 轨迹完成 S 型速度规划后, 设置采样周期为 0.01s。整条轨迹插补点如图 5-18 (a) 所示, 图 5-18 (b) 为过渡区的放大图。末端 S 型速度规划结果如图 5-19 (a)、(b) 和 (c) 所示, 由于



直线段路程太短，并没有加速到用户设定的运行速度  $100\text{mm/s}$ ，自适应调整后由  $v_s$  直接变速到  $v_{\text{mid}}$ ，过渡区的运行速度  $v_t$  也被降低。空间位置、速度和加速度在  $x$ 、 $y$  和  $z$  轴方向上的分量与时间  $t$  的关系如图 5-20 (a) 列、(b) 列和 (c) 列所示。逆解后的关节角位置与时间  $t$  的关系如图 5-21 所示。

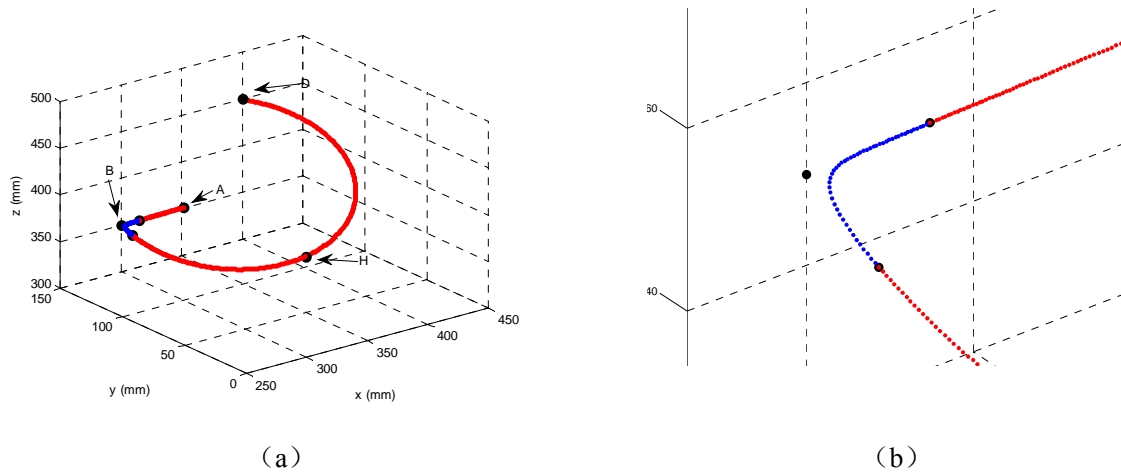


图 5-18 直线-圆弧过渡轨迹插补点

Fig. 5-18 Line-circle transition trajectory interpolation points

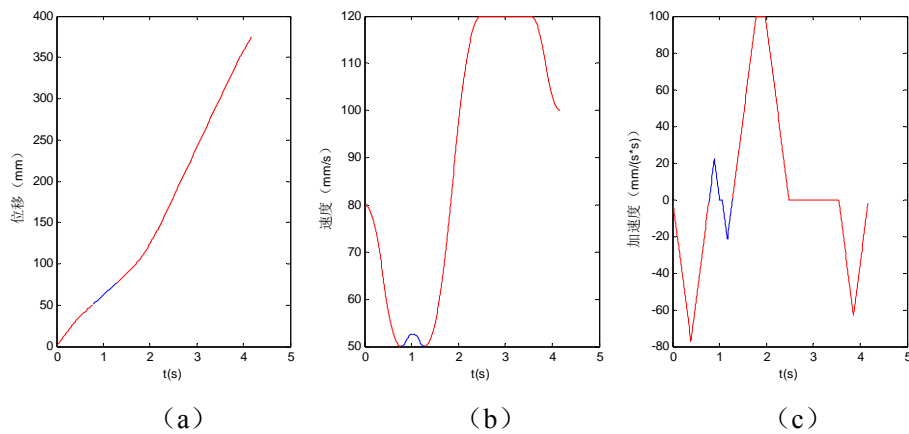


图 5-19 末端位置 S 型速度规划

Fig. 5-19 The S-type velocity planning results at end position

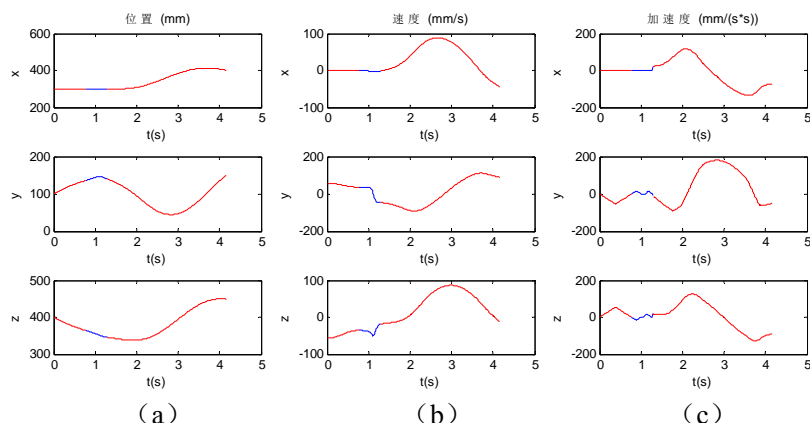
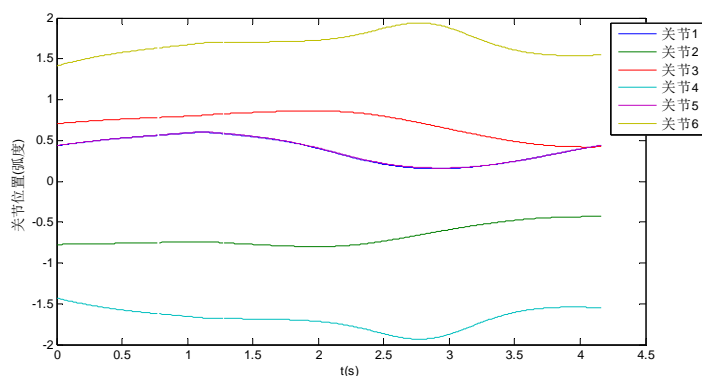
图 5-20 空间位置、速度和加速度的  $x$ 、 $y$  和  $z$  轴分量Fig. 5-20 Space position, speed, acceleration at axis  $x, y, z$ 

图 5-21 关节角位置

Fig. 5-21 The position of joints

### 5.3.4 B 样条-B 样条过渡轨迹规划算法

设置示教点  $T1\_1(300, 100, 400)$ ,  $T1\_2(300, 150, 350)$ ,  $T1\_3(350, 180, 370)$  生成第一条三次非均匀 B 样条, 设置  $T2\_1(350, 180, 370)$ ,  $T2\_2(400, 50, 300)$ ,  $T2\_3(450, 150, 450)$  生成第二条三次非均匀 B 样条, 相交于点  $O(350, 180, 370)$ , 以上单位为毫米(mm)。设定安全过渡区球体半径  $R$  长度为 20mm,  $v_s=100\text{mm/s}$ ,  $v_l=120\text{mm/s}$ ,  $v_{\text{mid}}=50\text{mm/s}$ ,  $v_e=150\text{mm/s}$ ,  $v_2=120\text{mm/s}$ ,  $a=100\text{mm/s}^2$ ,  $J=200\text{mm/s}^3$ ,  $A_{\text{max}}=500\text{mm/s}^3$ ,  $\text{minlen}=0.01\text{mm}$ ,  $\text{minlen}$  为整条曲线细分精度。 $v_l$  为第一条样条曲线上的运行速度,  $v_2$  为第二条样条曲线上的运行速度,  $v_{\text{mid}}$  为过渡区临界点上的速度, 过渡区内的运行速度设为  $v_t=v_{\text{mid}}+10=60\text{mm/s}$ 。轨迹完成 S 型速度规划后, 设置采样周期为 0.01s, 整条轨迹插补点如图 5-22 (a) 所示, 图 5-22 (b) 为过渡区的放大图。末端 S 型

速度规划结果如图 5-23 (a)、(b) 和 (c) 所示, 由于第一条样条曲线路程太短, 并没有加速到用户设定的运行速度 120mm/s, 自适应调整后运行速度为 105mm/s, 过渡区的运行速度  $v_t$  也被降低。空间位置、速度和加速度在  $x$ 、 $y$  和  $z$  轴方向上的分量与时间  $t$  的关系如图 5-24 (a) 列、(b) 列和 (c) 列所示。逆解后的关节角位置与时间  $t$  的关系如图 5-25 所示。

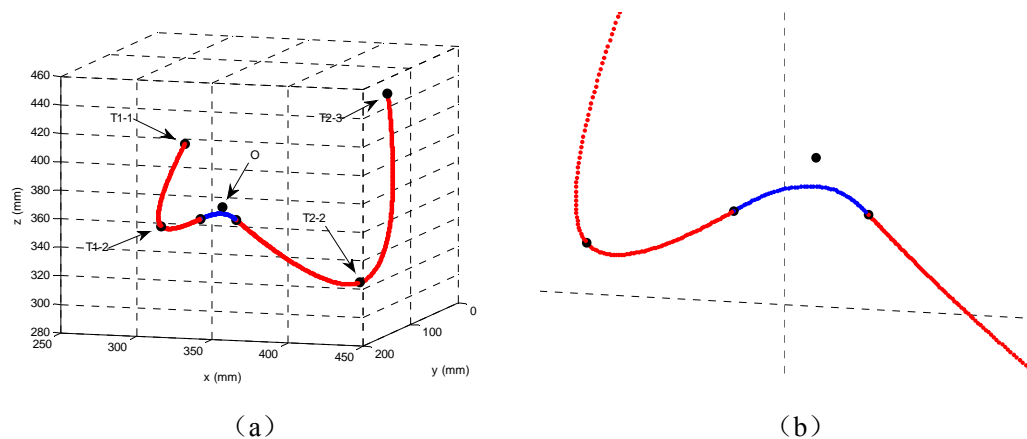


图 5-22 三次非均匀 B 样条-三次非均匀 B 样条过渡轨迹插补点

Fig. 5-22 Interpolation points of Nonuniform cubic B-spline and nonuniform cubic B-spline transition

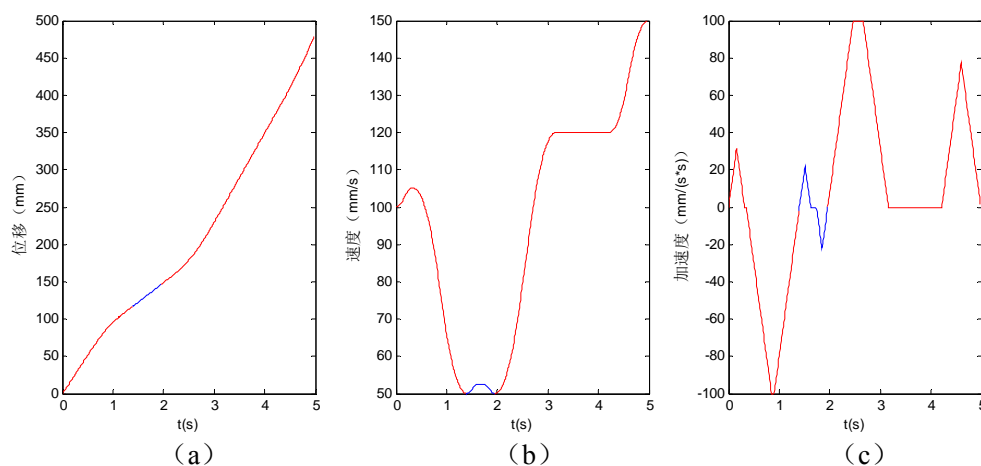


图 5-23 末端位置 S 型速度规划

Fig. 5-23 The S-type velocity planning results at end position

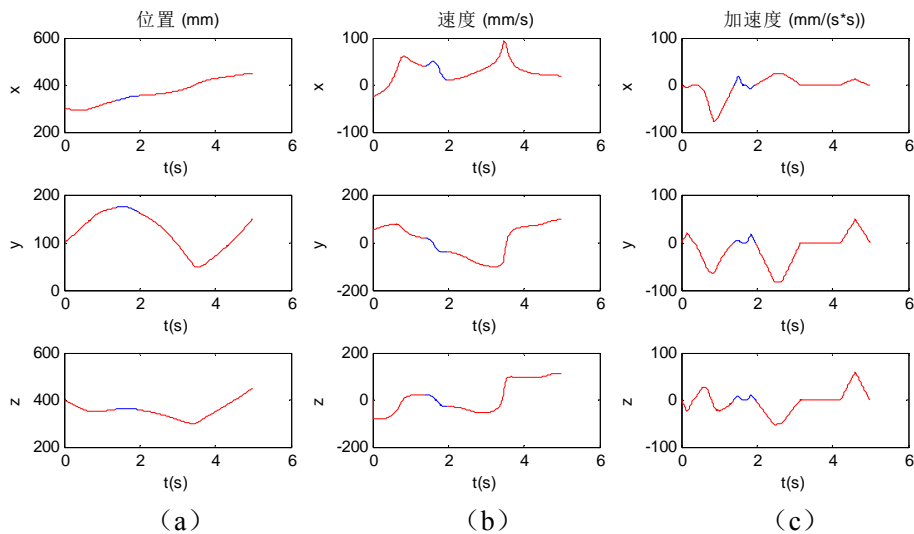
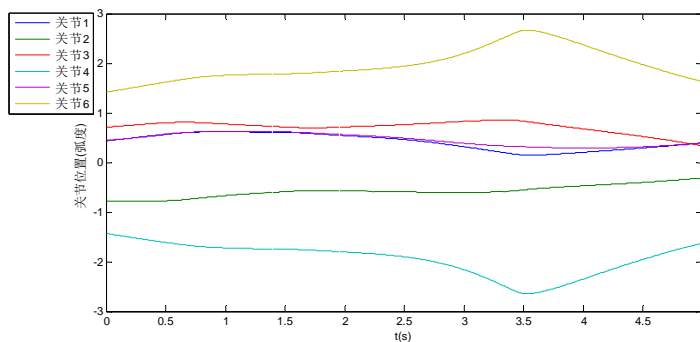
图 5-24 空间位置、速度和加速度的  $x$ 、 $y$  和  $z$  轴分量Fig. 5-24 Space position, speed, acceleration at axis  $x, y, z$ 

图 5-25 关节角位置

Fig. 5-25 The position of joints

## 5.4 本章小结

本章使用 C++ 语言进行编程，对前几章涉及的算法进行了编码与封装，包括六自由度机器人正逆解算法、S 型速度规划算法、笛卡尔空间直线、圆弧、三次 B 样条轨迹规划算法及空间轨迹过渡算法。并通过 MATLAB 对算法结果进行可视化的仿真，对仿真结果进行了详细分析，同时论证了算法的正确性。

## 结 论

轨迹规划技术是工业机器人运动控制器中至关重要的一部分。国内工业机器人发展较国外相比起步晚，发展缓慢，至今国内大部分投入到生产线上的机器人都来源于进口，因此发展工业机器人先进技术也是迫在眉睫，对提升国内社会生产力具有巨大的作用。因此，本文借助于通用六自由度工业机器人平台研究了以下几方面的内容：

(1) 首先分析了机器人运动学方面的理论知识，对机器人平台进行 D-H 建模，研究了机器人笛卡尔空间位置到关节空间的映射关系，并推导和设计了机器人正逆解算法。

(2) 考虑到机器人运动控制器是一个很庞大的系统，在设计时，将轨迹规划分为两层：速度规划层和轨迹生成层。

在速度规划层，对常用的速度控制曲线进行对比分析后，最终选用高性能的 S 型速度控制曲线作为本文的研究目标，并设计了两种自适应可控 S 型速度规划器：基于给定运行速度的 S 型速度规划器和基于给定运行时间的 S 型速度规划器，在标量空间进行规划，用户只需输入位移、速度、时间和加速度等参数，系统经过运算后就可得到 S 型速度规划结果。在一些时间要求同步的场合，基于给定运行时间的 S 型速度规划器更是起着关键的作用。

在轨迹生成层，首先分析了工业运动控制中最常见的两种轨迹直线和圆弧，并推导了空间直线、圆弧和三次 B 样条曲线的计算方式，借助第三章研究设计的自适应可控 S 型速度规划器，完成了这三种空间独立曲线的自适应可控轨迹规划。

(3) 机器人末端执行器在进行连续示教点运动时，为了避免刚体冲击，传统方式是在末端执行器经过中间示教点时降速为零，然后再加速，从而达到平滑过渡的目的。也有通过设置安全过渡区，在过渡区内使用圆弧作为过渡轨迹，此种方式较难控制。本文研究了以空间自由曲线三次 B 样条作为安全过渡区轨迹，实现了空间直线-直线、空间直线-圆弧和空间三次非均匀 B 样条-三次非均匀 B 样条之间的平滑过渡，大大提升了机器人末端执行器的作业效率。

(4) 使用 C++ 语言对以上的研究算法进行编码，充分地考虑算法的通用性，对算法进行封装，将接口与实现分离，算法实现细节对用户不可见，用户只需通过相应接口就可实现对算法的调用。构造相应输入数据对算法进行测试。使用 MATLAB 可视化编程语言对算法输出接口产生的结果进行仿真与分析，验证了算法的正确性和可行性。

本文对设计的自适应可控轨迹规划技术进行了大量的仿真与验证，然而在后

续的开发中，还会进一步进行优化。由于轨迹规划只属于机器人运动控制器中的一部分，要完成整个系统的设计并且最后应用到自主研发的通用六自由度工业机器人平台还需更多的时间，因此对本文所做工作仍然存在需要完善的地方：

- (1) 算法的执行效率有待进一步的提高。
- (2) 更深一层的考虑算法的通用性，对算法更好的封装。
- (3) 充分考虑用户需求，对笛卡尔空间轨迹规划模块进行进一步完善。

## 致 谢

时光飞逝，转眼之间，即将为三年的研究生生涯画上句号。回顾过去的点点滴滴，我永远无法忘记众多的良师益友给予了我科研上的指导和帮助、思想上的鞭策和鼓励、生活上的关怀和照顾以及人生的指引，感谢你们陪伴我度过人生中最美好的时光！

首先，我要感谢我的导师梁艳阳副教授！大四时有幸认识梁老师，在他的指引下我进入工业机器人研究室实验室跟着师兄一起学习，并参与到您的科研项目中。在您细心指导及鼓励下展开了我人生第一个科研项目，您对工作积极热情及细致入微的态度、一切为了学生，对学生的无私奉献的精神让我深受触动，让我对未来的学习生活充满期待和向往，同时也充满力量。在研究生期间，与梁老师密切接触，让我发现梁老师不仅是一位睿智的学者，就像那远洋的指南针，给我指引方向，更是一名积极乐观的学者，给予我前进无限的动力。在此我要向我的导师致以最诚挚的敬意和最衷心的感谢！是您给予了我学习锻炼的机会，让我在研究生期间得到相当大的提升！

感谢各位已经毕业的师兄在进入实验室后对我的帮助！感谢实验室师弟师妹在研究设计中给予的支持！

感谢我的母校——西南科技大学，是你提供了一个良好的学习氛围，优美的校园环境，给我的学习和工作带来了极大的帮助！

感谢我的父母，感谢你们二十多年含辛茹苦的培养，是你们给了我坚强的精神支撑，让我得以顺利完成学业，我永远爱你们！

感谢参加本论文审批与答辩的各位老师百忙中所给予的指导！

---

## 参考文献

- [1] 苗建伟, 杨铁宝, 曾林森. 一种六自由度机械手的智能轨迹规划 [J]. 组合机床与自动化加工技术, 2014 (9): 73-76.
- [2] Fu R, Ju H. Time-optimal trajectory planning algorithm for manipulator based on PSO [J]. Information and Control, 2011, 40(6): 802-808.
- [3] 肖文皓, 白瑞林, 许凡, 等. 基于信赖域算法的机械臂时间最优轨迹规划 [J]. 传感器与微系统, 2013, 32(6): 77-80.
- [4] Gasparetto A, Zanutto V. A technique for time-jerk optimal planning of robot trajectories [J]. Robotics and Computer-Integrated Manufacturing, 2008, 24(3): 415-426.
- [5] Gasparetto A, Zanutto V. Optimal trajectory planning for industrial robots [J]. Advances in Engineering Software, 2010, 41(4): 548-556.
- [6] Menasri R, Oulhadj H, Daachi B, et al. A genetic algorithm designed for robot trajectory planning [C]; Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on. IEEE, 2014: 228-233.
- [7] Menasri R, Nakib A, Oulhadj H, et al. Path planning for redundant manipulators using metaheuristic for bilevel optimization and maximum of manipulability [C]; Robotics and Biomimetics (ROBIO), 2013 IEEE International Conference on. IEEE, 2013: 145-150.
- [8] Nong-Liang S, Yan-Jun W. Time-optimal trajectory planning based on improved adaptive genetic algorithm [J]. 测试科学与仪器, 2012, 03:103-108.
- [9] 居鹤华, 付荣. 基于 GA 的时间最优机械臂轨迹规划算法 [J]. 控制工程, 2012, 19(3): 472-477.
- [10] 李东洁, 邱江艳, 尤波. 一种机器人轨迹规划的优化算法 [J]. 电机与控制学报, 2009, 13(1): 123-127.
- [11] Costantinescu D, Croft E. Smooth and time-optimal trajectory planning for industrial manipulators along specified paths [J]. Journal of robotic systems, 2000, 17(5): 233-249.
- [12] Constantinescu D. Smooth time optimal trajectory planning for industrial manipulators [D]; University of British Columbia, 1998.
- [13] Simon D, Isik C. A trigonometric trajectory generator for robotic arms [J]. International Journal of Control, 1993, 57(3): 505-517.



- 
- [14] Piazzzi A, Visioli A. Global minimum-jerk trajectory planning of robot manipulators [J]. Industrial Electronics, IEEE Transactions on, 2000, 47(1): 140-149.
- [15] Piazzzi A, Visioli A. An interval algorithm for minimum-jerk trajectory planning of robot manipulators [C]; Decision and Control, 1997, the 36th IEEE Conference on. IEEE, 1997, 2: 1924-1927.
- [16] Macfarlane S, Croft E A. Jerk-bounded manipulator trajectory planning: design for real-time applications [J]. Robotics and Automation, IEEE Transactions on, 2003, 19(1): 42-52.
- [17] Macfarlane S, Croft E A. Design of jerk bounded trajectories for online industrial robot applications [C]; Robotics and Automation, 2001. IEEE International Conference on. IEEE, 2001, 1: 979-984.
- [18] Petrínek K, Kovacic Z. Trajectory planning algorithm based on the continuity of jerk [C]; Control & Automation, 2007 MED'07 Mediterranean Conference on. IEEE, 2007: 1-5.
- [19] Huang P, Xu Y, Liang B. Global minimum-jerk trajectory planning of space manipulator [J]. International Journal of Control Automation and Systems, 2006, 4(4): 405-413.
- [20] 林仕高, 刘晓麟, 欧元贤. 机械手笛卡尔空间轨迹规划研究 [J]. 机械设计与制造, 2013(3): 49-52.
- [21] Wu C, Ge Y W, Zhang G, et al. S-Curve Algorithm Based on SCM [J]. Applied Mechanics and Materials, 2013, 416: 626-630.
- [22] Wang X, Wang J-W, Rao Z, et al. Trajectory and Velocity Planning Algorithm Based on NURBS curve [J]. Journal of System Simulation, 2008, 20(15):3973-3980.
- [23] Sano S, Ouyang H, Uchiyama N. Residual load sway suppression for rotary cranes using simple dynamics model and S-curve trajectory [C]; Emerging Technologies & Factory Automation (ETFA), 2012 IEEE 17th Conference on, F, 2012 [C]. IEEE, 2012: 1-5.
- [24] Tsai M-C, Cheng C-W. A real-time predictor-corrector interpolator for CNC machining [J]. Journal of Manufacturing Science and Engineering, 2003, 125(3): 449-460.
- [25] 薛晖. S 型速度曲线在有轨巷道堆垛机速度控制中的应用研究 [D]; 兰州交通大学, 2013.
-

- 
- [26] 于洋, 刘平. S 型速度曲线进给率控制研究 [J]. 制造技术与机床, 2012(9).
- [27] 韩庆瑶, 赵保亚, 谭建鑫, et al. NURBS 曲线曲面重构的方法 [J]. 机械设计与制造, 2006(3): 137-139.
- [28] Lampariello R, Nguyen-Tuong D, Castellini C, et al. Trajectory planning for optimal robot catching in real-time [C]; Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE, 2011: 3719-3726
- [29] 皮佑国, 范德和. 数控加工中连续微线段轨迹的 B 样条曲线拟合 [J]. 华南理工大学学报 (自然科学版), 2012, 202(38):194-234
- [30] Lee A-C, Lin M-T, Pan Y-R, et al. The feedrate scheduling of NURBS interpolator for CNC machine tools [J]. Computer-Aided Design, 2011, 43(6): 612-628.
- [31] 林锋, 汪地. 三次非均匀 B 样条曲线插补算法研究 [J]. 组合机床与自动化加工技术, 2012 (8): 32-35.
- [32] 李林峰, 马蕾. 三次均匀 B 样条在工业机器人轨迹规划中的应用研究 [J]. 科学技术与工程, 2013 (13): 3621-3625.
- [33] 李芳, 刘凯, 任杰青, 等. 一种空间 B 样条曲线的变步长离散插补算法 [J]. 制造业自动化, 2013, 35(18): 37-40.
- [34] 李小霞, 汪木兰, 刘坤, 等. 基于五次 B 样条的机械手关节空间平滑轨迹规划 [J]. 组合机床与自动化加工技术, 2012 (8): 39-42.
- [35] Haron H, Rehman A, Adi D, et al. Parameterization method on B-spline curve [J]. Mathematical Problems in Engineering, 2012.
- [36] 金席卷, 姚杰, 方逵. C~ 1 插值 B 样条曲线 [J]. 科技创新导报, 2012 (14): 30-30.
- [37] 陶淑一. 反求三次均匀 B 样条曲线控制顶点 [J]. 科技通报, 2014, 30(3): 23-25.
- [38] 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.
- [39] 缪东晶, 王国磊, 吴聊, 等. 自由曲面均匀喷涂的机器人轨迹规划方法 [J]. 清华大学学报: 自然科学版, 2013 (10): 1418-1423.
- [40] 蒋新松. 机器人学导论 [M]. 辽宁科学技术出版社, 1994.
- [41] 周芳芳, 樊晓平, 叶榛. DH 参数表生成三维机器人模型仿真系统 [J]. 系统仿真学报, 2006, 18(4): 947-950.
- [42] Jeon J W, Ha Y Y. A generalized approach for the acceleration and deceleration of industrial robots and CNC machine tools [J]. Industrial Electronics, IEEE
-

- Transactions on, 2000, 47(1): 133-139.
- [43] 董玉红, 邵俊鹏. 机床数控技术 [M]. 哈尔滨工业大学出版社, 2005.
- [44] 胡建华, 廖文和. CNC 系统中几种升降速控制曲线的研究与比较 [J]. 南京航空航天大学学报, 1999, 31(6): 706-711.
- [45] 商允舜. CNC 数控系统加减速控制方法研究与实现 [D]; 浙江: 浙江大学机械与能源工程学院, 2006.
- [46] 王润孝, 秦现生. 机床数控原理与系统 [M]. 西北工业大学出版社, 1997.
- [47] 张莉彦. 基于数据采样插补的加减速控制的研究 [J]. 北京化工大学学报, 2002, 29(3): 91-93.
- [48] Bokhorst J A, Slomp J, Suresh N C. An integrated model for part-operation allocation and investments in CNC technology [J]. International Journal of Production Economics, 2002, 75(3): 267-285.
- [49] Chen Z, Guo W, Li C. Research and development on full distribution CNC system [C]; Industrial Electronics and Applications, 2006 1ST IEEE Conference on, F, 2006. IEEE, 2006: 1-6.
- [50] 王允航. 连续轨迹的 S 型速度规划的研究 [D]; 哈尔滨工业大学, 2010.
- [51] 李加文, 陈宗雨, 李从心. 基于函数逼近的三角函数加减速方法 [J]. 机床与液压, 2006 (3): 66-67.
- [52] 郭永忠. 三角函数在数控机床中的应用 [J]. 中国科教创新导刊, 2009 (25): 170-170.
- [53] 朱晓春, 屈波, 孙来业, 等. S 曲线加减速控制方法研究 [J]. 中国制造业信息化: 学术版, 2007, 35(12): 38-40.
- [54] 石川, 赵彤, 叶佩青, 等. 数控系统 S 曲线加减速规划研究 [J]. 中国机械工程, 2007, 18(12): 1421-1425.
- [55] 张碧陶, 高伟强, 沈列, 等. S 曲线加减速控制新算法的研究 [J]. 机床与液压, 2009, 37(10): 27-29.
- [56] 陈国良, 黄心汉, 王敏. 机械手圆周运动的轨迹规划与实现 [J]. 华中科技大学学报: 自然科学版, 2006, 33(11): 63-66.
- [57] Piegl L, Tiller W. The NURBS book (2nd ed.) [M]. Springer-Verlag New York, Inc., 1997.
- [58] 刘新山. NURBS 曲线插补技术研究及其仿真 [D]; 吉林大学, 2007.
- [59] Lei W, Sung M, Lin L, 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] 刘华伟, 陈耀元, 王丹. 三次NURBS曲线积分问题的数值方法及应用研究 [J]. 中国造船, 2006, 78-83.
- [61] 陈晓兵, 廖文和, 孙全平. 一种高速数控加工自适应进给速度生成算法 [J]. 中国机械工程, 2008, 204-207.
-

## 攻读硕士学位期间参与研究科研项目情况

- [1] 国防技术基础研究项目（科工计[2011]869 号）.
- [2] 通用六自由度工业机器人运动控制器研制.