

参赛密码 _____

(由组委会填写)



第十二届“中关村青联杯”全国研究生 数学建模竞赛

题 目 数控加工刀具运动的优化控制模型研究

摘 要：

本文针对数控加工刀具运动的优化控制问题，主要完成了以下几个方面的工作：

对于问题一，本文提出了折线转接点速度不为 0 的过渡算法。首先通过查阅文献，综合考虑了折线型加工型线的转接速度限制条件（加速度限制和加工误差限制），并在折线过渡点处采用了一步插补的直接过渡方法。在各直线段的加工阶段，提出了加加速度 J 恒定的 S 型加减速算法对直线进行宏观速度规划。最后建立了基于时间分割法的插补算法模型：插补周期恒定，采用前加减速速度控制，计算出理论插补点的坐标，计算理论插补点与当前位置的偏移值，对偏移值进行向上取整使其为机床分辨率的整数倍，从而得到该插补周期的实际偏移量，由该算法插补得到的插补点的误差通常能控制在一个分辨率内，精度很高。最后对夹角为 90° 和 135° 的折线交点处的坐标速度变化规律进行了讨论。

对于问题二，本文在问题一提出的算法的基础上进行了调整。首先考虑了圆弧段加工时的速度限制条件（加速度限制、圆弧弓高误差限制和多步转接条件限制）。在直线段和圆弧过渡的过程中分别考虑了从直线段过渡到圆弧段、从圆弧段过渡到直线段过渡线段插补点坐标的求解方法。在直线段的加工过程中仍用 S 型加减速算法进行速度规划，圆弧段的插补采用匀速率的方式进行。问题二不考虑瞬时启动加速度及瞬时启动速度对加工过程的影响，速度规划中的速度和加速度均从 0 开始变化，插补算法模型沿用上题中的算法。通过对圆弧半径变化的讨论发现：圆弧半径越大，算法效率越高；圆弧半径越小，算法效

率越低。最后将上述算法应用于加工示例进行仿真。

对于问题三，考虑瞬时启动加速度及瞬时启动速度对加工过程的影响，影响主要体现在对速度规划的影响上。经过对比分析发现，考虑瞬时启动加速度及瞬时启动速度能够使得刀具较快进入匀加速阶段以及匀速阶段，加工效率更高。最后通过加工示例仿真验证了这一点。

对于问题四，基于前三个问题中对 S 型加减速控制算法的使用情况以及与其他常用加减速方法横向对比来总结出其优缺点。在 S 型曲线加减速算法中，为了保证速度光滑，加速度连续，启停时不产生冲击、失步而引入加加速度，在一定程度上增强了机床运行的平稳性，但是加加速度的非连续也限制了增强平稳性的程度。因此，在满足精度和误差要求的条件下，建立出的能够提高机床运行平稳性的优化控制运动模型主要就是能够使加加速度连续变化。而在 $0\sim\pi$ 之间，正弦曲线对应的一次、二次、三次导数均是连续可导，这样光滑连续的曲线对解决加加速度连续变化提供了可能。但是三角函数加减速算法对最大允许加速度利用不高，所以需要对其提出改进以达到优化的目的。

关键字：S 型加减速算法 直接过渡插补法 时间分割法 速度规划 弓高误差 瞬时启动加速度 瞬时启动速度 改进型三角函数加减速算法

目录

1	问题重述.....	5
1.1	问题背景.....	5
1.2	待解决的问题	5
2	问题分析.....	7
3	符号说明及模型假设.....	8
3.1	符号说明.....	8
3.2	模型基本假设	9
4	折线实时加工优化控制算法	10
4.1	转接速度限制条件	10
4.1.1	加速度限制	10
4.1.2	加工误差限制	11
4.2	轨迹段转接处理算法	11
4.3	S 型加减速算法速度规划	12
4.4	S 型加减速插补算法.....	19
4.4.1	理论插补点.....	19
4.4.2	实际插补点及插补算法	20
4.5	实时加工控制算法流程	22
4.6	折线夹角与转接点速度讨论	23
5	直线段和圆弧实时加工控制算法模型	25
5.1	转接速度限制条件	25
5.1.1	加速度限制	25
5.1.2	圆弧弓高误差限制	26
5.1.3	多步转接条件限制	26
5.2	圆弧转接处理算法	26
5.2.1	直线段转接圆弧	26
5.2.2	圆弧转接直线段	27
5.3	直线段和圆弧实时加工控制算法	28
5.3.1	S 型速度规划.....	28
5.3.2	直线和圆弧加工控制算法流程	28
5.4	圆弧半径对算法效率的影响	29

5.5 加工示例仿真验证	29
6 瞬时启动加速度及瞬时启动速度	31
6.1 瞬时启动加速度及瞬时启动速度对速度规划的影响	31
6.2 加工示例仿真验证	32
7 S 型曲线加减速控制方法分析.....	34
8 平稳性优化控制运动模型.....	34
9 模型评价	38
9.1 模型的优点.....	38
9.2 模型的缺点.....	38
参考文献.....	39
附录 I	40
附录 II	41

1 问题重述

1.1 问题背景

随着机械零件对加工质量与精度要求的提高，数控加工技术正朝着高速、高效、高精度的方向发展，要求机床各运动轴都能够在极短的时间内达到高速运行状态并实现高速准停。在数控系统中，为保证机床在启停时不产生冲击、失步、超行程或振荡，必须有专门的加减速控制规律以使机床在各种加工作业情况下能够快速、准确地停留在给定位置上。因此，研究开发高速、高精度、高有效柔性的加工刀具运动加减速控制算法，已成为现代高性能数控系统的研究重点，其中较有代表性的是 S 型曲线加减速控制算法。

而在数控机床的连续切削中，其进给速度不仅直接影响到加工零件的表面粗糙度和精度，也能影响到加工效率。对于不同材料的工件、加工刀具、加工方式和条件，应选择合适的进给速度，而进给速度的控制方法则与采用的插补算法有关。插补的任务就是根据起点、重点、轨迹轮廓、进给速度等，按照数控系统的当量，对轮廓轨迹进行细化，所以加工刀具行走的路线一定是一系列首尾相接的直线段。

对于一般曲线加工，加工控制算法就是在满足误差要求的条件下，通过插补的方法，找出若干小直线段来组成加工刀具的运动轨迹，然后计算各坐标轴上刀具运动的控制信息来合成刀具实际加工轨迹和速度控制，加工出所需的零件。而优化控制则是要在数控机床所提供的精度、速度、加速度等限制条件下，寻求对机床刀具在各坐标轴方向上的运动进行合理控制，今儿优化其加工效率，且还能保证机床运动平稳、速度光滑以及加速度连续等。

1.2 待解决的问题

● 问题一

假设加工型线位折线，在指定加工误差（指在加工型线的法线方向上，加工型线与刀具实际轨迹的差值最大值）的条件下，建立实时加工优化控制算法，当相邻两折线段夹角为 90° 和 135° 时，讨论通过折线交点时对应各坐标运动速度的变化。

● 问题二

假设加工型线是由直线段和圆弧段（相切或不相切）组成的连续曲线，在指定加工误差的条件下，不考虑瞬时启动加速度及瞬时启动速度，建立实时加工优化控制算法，讨论圆弧半径的变化对算法效率的影响；并应用所建立的模型对下方所附的**加工路径示例**进行检验。

● 问题三

在问题二的基础上，考虑瞬时启动加速度及瞬时启动速度，建立相对应的实时加工优化控制算法；并应用所建立的模型对下方所附的**加工路径示例**进行检验。

● 问题四

结合前三个问题，分析 S 型曲线的加减速控制方法的优缺点，在满足精度和速度要求的条件下，建立能提高机床运行平稳性的优化控制运动模型（如刀

具在各坐标轴方向上的运动满足加加速度连续变化等)。

附：加工路径示例

- 1) 图 1.1 所示是一个圆角矩形切割路径的示意图，路径的四个圆角是半径为 0.5cm 整圆的 1/4 圆弧。矩形外围大小为：41×41（单位：cm²）。

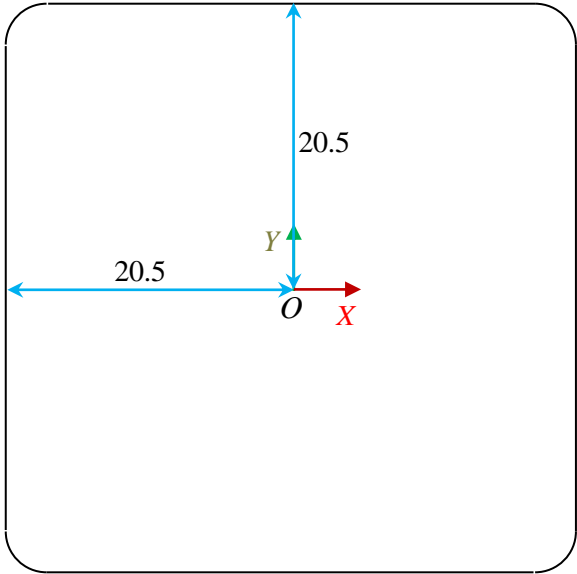


图 1.1 圆角矩形切割路径

- 2) 在实际加工过程中，从坐标为（-20.500， -20.000， 0.000）的节点 1 位置开始下刀，该点处的瞬时速度为 0.13 m/min；从节点 1 到节点 2（-20.500， -17.708， 0.000）的过程中，要求最大速度为 0.19 m/min。从节点 2（-20.500， -17.708， 0.000）出发，顺时针加工，回到该点（-20.500， -17.708， 0.000）（记为节点 11），期间要求最大速度为 1.26，节点 11 处的终止速度为 0.13 m/min。表 1 给出加工过程中对最大速度的要求。

表 1.1 符号含义表

路径节点	节点坐标（X、Y、Z）	用户设定最大频率 F	最大频率 F 对应速度
1	（-20.500， -20.000， 0.000）	2773	0.13
2	（-20.500， -17.708， 0.000）	4029	0.19
3	（-20.500， 20.000， 0.000）	26865	1.26
4	（-20.000， 20.500， 0.000）	26865	1.26
5	（20.000， 20.500， 0.000）	26865	1.26
6	（20.500， 20.000， 0.000）	26865	1.26
7	（20.500， -20.000， 0.000）	26865	1.26
8	（20.000， -20.500， 0.000）	26865	1.26

9	$(-20.000, -20.500, 0.000)$	26865	1.26
10	$(-20.500, -20.000, 0.000)$	26865	1.26
11	$(-20.500, -17.708, 0.000)$	2773	0.13

表中最大频率指的是控制脉冲的最大频率，本题可不予考虑，对应的速度指的是刀具的运动速度，单位是 m/min。

运动过程中，机床对于速度、加速度、加加速度等限制条件如下：

- 进给速度 V 范围 $[V_{min}, V_{max}]$: $[0.13, 6]$ 单位: m/min
- 加速度 A 范围 $[A_{min}, A_{max}]$: $[0.02, 0.6]$ 单位: m/s^2
- 加加速度 J_{const} : 300 单位: mm/s^3
- 瞬时启动速度 V_0 : 0.13 单位: m/min
- 瞬时启动加速度 A_0 : 0.02 单位: m/s^2
- 误差 ε : 1 单位: μm
- 分辨率: $\frac{1}{1280}$ 单位: mm

2 问题分析

本题要求建立数控加工控制优化算法，即宏观上对加工路径的速度、加速度进行合理规划，使得机床运动平稳、速度光滑、加速度连续且时间最优；微观上建立合理的插补算法，找出若干小直线段组成加工刀具的轨迹，使得刀具的轨迹满足误差的要求。我们对本题提出的四个问题作如下分析。

● 问题一

问题一要求针对折线型加工型线建立实时加工控制优化算法。如果要刀具精确走到两条直线段的折点处，必须采用每条直线分别加速再降速为 0 的方法。但该方法不但加工效率低且频繁启停将会对机床造成冲击。因此考虑在转折点处保持一定速度，且保证加工轨迹在轮廓误差允许范围内。通过查阅已有文献获得直线段间转接速度的限制条件。在分别加工每条直线时，考虑采用 S 型加减速算法进行速度规划；直线段的转折点处考虑采用直接过渡法进行过渡；最后建立有效的插补算法模型，使得插补速度在宏观上能够与 S 型加减速规划结果相吻合，微观上满足系统误差条件。在以上算法的基础上分别讨论线段夹角是 90° 和 135° 时各坐标速度的变化。

● 问题二

问题二要求针对直线段和圆弧的加工型线建立实时加工控制优化算法。解题的思想仍与第一题类似。首先考虑在进行圆弧段加工时的速度限制，由于刀具所走的路线均为一段一段微小直线段构成，用微小直线段逼近圆弧，且微小直线间有夹角，因此要考虑由此带来的圆弧弓高误差、加速度限制等，保证满足加工误差要求。在直线段和圆弧过渡的过程中要分别考虑从直线段过渡到圆弧段、从圆弧段过渡到直线段过渡线段插补点坐标的求解方法。在直线段的加工过程中仍考虑用 S 型加减速算法进行速度规划，圆弧段的插补考虑采用匀速率的方式进行，但插补速度方向在不断进行微调。本题中不考虑瞬时启动加速

度及瞬时启动速度对加工过程的影响，因此进行速度规划时加速度和速度均从 0 开始变化。建立满足以上要求的实时加工控制优化算法后，将算法应用于加工示例仿真。

● 问题三

问题三建立在问题二的基础上，考虑瞬时启动加速度及瞬时启动速度对加工过程的影响。初步分析，影响主要体现在对速度规划的影响上。在考虑瞬时启动加速度及瞬时启动速度的情况下，对速度规划的起始速度重新进行定义，建立实时加工算法，并将算法应用于加工示例仿真，并与问题二中的仿真结果对比。

● 问题四

问题四主要是希望通过前三个问题中对 S 型曲线加减速控制方法的使用情况，与其他常用加减速方法横向对比，就响应时间、速度光滑度、方法复杂度等方面总结出其优缺点。并针对 S 型曲线加减速控制方法的缺点，提出一种优化控制运动模型在满足精度和误差要求的条件下，可以弥补其缺点。

3 符号说明及模型假设

3.1 符号说明

表 3.1 符号含义表

符号	含义
d	机床分辨率
T	插补周期
ε	最大加工误差
$a_{x-\max}$	x 轴最大加速度
$a_{y-\max}$	y 轴最大加速度
P_s	加工型线起点
P_e	加工型线终点
P_{se}	加工型线转折点

P_c	转折处插补线段起点
P_n	转折处差不线段终点
$(\Delta x_i, \Delta y_i)$	第 i 个插补周期理论进给量
$(\Delta x_i^*, \Delta y_i^*)$	第 i 个插补周期实际进给量
t_{all}	S 型速度规划总时间
t_r	S 型速度规划总时间整除插补周期 T 后的余数
v_s	x 轴、y 轴速度的合成速率
E_R	圆弧插补弓高误差
L_c	圆弧弧长
v_{cir}	圆弧插补速度
φ	圆弧对应的圆心角
a_0	瞬时启动加速度
v_0	瞬时启动速度

3.2 模型基本假设

对于所研究的问题的参数和范围，本文作出如下几点假设和限定：

- 假设不考虑刀具尺寸大小及刀具磨损，加工刀具抽象为一点；
- 假设 z 轴为竖直方向的进给轴，即加工立体工件时，沿着 z 轴方向将工件分为不同平面，仅仅考虑每个平面上曲线的加工算法，即所建立算法只包含 x 轴与 y 轴坐标。若需要走空间曲线，空间曲线的算法也可由平面算法类推得到；

- 假设不考虑机床电机的实际响应及控制系统通信延时等原因造成的误差。

4 折线实时加工优化控制算法

4.1 转接速度限制条件

数控加工过程中，待加工零件的轮廓轨迹多由直线和圆弧组成，如果对每一条加工曲线都采用从静止加速到目标速度再减速到 0 的方法将会导致平均加工速度低，影响加工效率；此外，频繁加减速会加大电机的负荷，产生运动噪声和降低控制精度，减少电机寿命；

为解决上述问题，需要一种方法使得满足一定条件的相邻待加工曲线之间不必减速到 0，可以保持速度上的连续，而且尽可能地达到编程时设定的进给速度。因此，为了实现相邻线段间转接时速度上的连续，可在保证加工轨迹在轮廓误差允许范围内，通过引入转接段来实现转接时速度的连续。

通常对转接速度起限制作用的条件主要有：加速度限制条件、加工误差限制条件和圆弧半径误差限制条件。

4.1.1 加速度限制

为了满足系统的动态响应要求，机床会有最大加速度的限制，最大加速度为一定值。

以图 4.1 来说明，在 xy 平面内，相邻两段加工轨迹为 L_1 、 L_2 ，转接点为 C ，依照加工方向， L_1 、 L_2 可以看作为矢量线段，以 $+x$ 为始边，逆时针旋转到 L_1 的角度为 α ，逆时针旋转到 L_2 的角度为 β ， $0 \leq \alpha, \beta < 2\pi$ 。根据文献^[1]可知，由加速度限制条件决定的转接速度如下：

$$v_C \leq \min \left\{ \frac{a_{x-\max} T}{|\cos \beta - \cos \alpha|}, \frac{a_{y-\max} T}{|\sin \beta - \sin \alpha|} \right\} \quad (4.1)$$

式中： T 为插补周期， $a_{x-\max}$ 、 $a_{y-\max}$ 为两轴最大允许加速度。

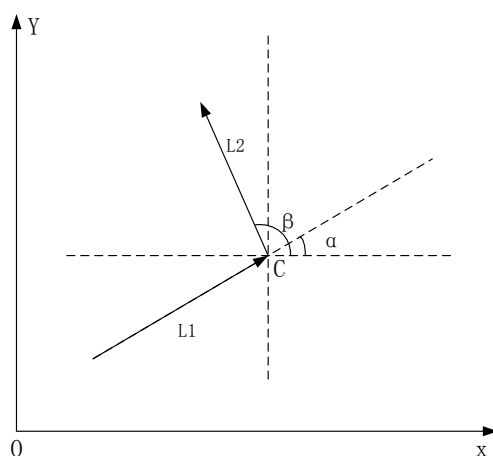


图 4.1 相邻两段轨迹

对式 (4.1) 进行进一步化简可得：

$$v_c \leq \min \left\{ \frac{a_{x-\max} T}{2 \sin \frac{\alpha + \beta}{2} \left| \sin \frac{\beta - \alpha}{2} \right|}, \frac{a_{y-\max} T}{2 \cos \frac{\alpha + \beta}{2} \left| \sin \frac{\beta - \alpha}{2} \right|} \right\} \quad (4.2)$$

式中 $\beta - \alpha$ 可以用两条加工线段矢量的夹角表示, 为已知量, 记为 θ , 即 $\theta = \beta - \alpha$ 。令 v_c 满足:

$$v_c \leq \min \left\{ \frac{a_{x-\max} T}{2 \left| \sin \frac{\theta}{2} \right|}, \frac{a_{y-\max} T}{2 \left| \sin \frac{\theta}{2} \right|} \right\} \quad (4.3)$$

则 v_c 一定满足式 4.2 中的不等式。

4.1.2 加工误差限制

在进行轨迹插补时, 如果轨迹长度不是插补步长的整数倍, 就要对轨迹进行转接处理, 具体处理方法见 4.2 节。这导致两段轨迹的转角处会出现加工误差, 如图 4.2 所示。图中, AB 为实际插补轨迹, h 为轨迹插补带来的加工误差, θ 为两段轨迹的夹角, 设系统允许的最大加工误差 ε , 则相应的转接速度限制为^[2]:

$$v_c \leq \frac{2\varepsilon}{T \tan \frac{\theta}{2}} \quad (4.4)$$

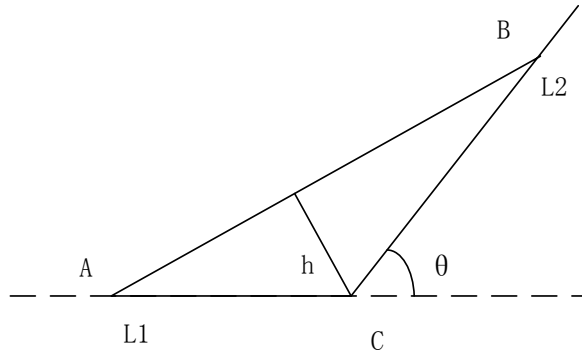


图 4.2 转接点处加工误差

最终的转接点处的速度由 (4.3) 和 (4.4) 中的较小值决定, 即:

$$v_c = \min \left(\frac{a_{x-\max} T}{2 \left| \sin \frac{\theta}{2} \right|}, \frac{a_{y-\max} T}{2 \left| \sin \frac{\theta}{2} \right|}, \frac{2\varepsilon}{T \tan \frac{\theta}{2}} \right) \quad (4.5)$$

4.2 轨迹段转接处理算法

如图 4.3 所示, 插补过程中, 当前段和下一加工线段理论上经 P_{se} 点转接, 若在转接时的速度不为 0, 而以速度 v_e 转接, 会使当前点 P_c 与转节点 P_{se} 之间的距离小于转接速度 v_e 在一个插补周期 T 内的位移 $S = v_e T$ 。如果在当前线段

补插下一步，下一步将超过转接点 P_{se} ，因此要在下一段线段上计算实际的转接点 P_n 。各点坐标已经在图 4.3 中标示出来了，我们考虑一般情况，即所要加工的线段均不是水平线段或竖直线段，则转接点的坐标应满足以下条件^[3]：

$$\begin{cases} (X_n - X_c)^2 + (Y_n - Y_c)^2 = S^2, \\ \frac{Y_n - Y_{se}}{X_n - X_{se}} = k, \end{cases} \quad (4.6)$$

其中， k 为 X_n 线段 $P_{se}P_e$ 的斜率。

解方程组 (4.6) 可以得到点 P_n 的坐标。由于方程组含有二次方程，因此有 2 组解， (X_{n1}, Y_{n1}) 和 (X_{n2}, Y_{n2}) ，由于点 P_n 在点 P_{se} 和点 P_e 之间，从而能确定唯一解使得：

$$(X_n - X_{se})(X_n - X_e) < 0 \quad (4.7)$$

成立。

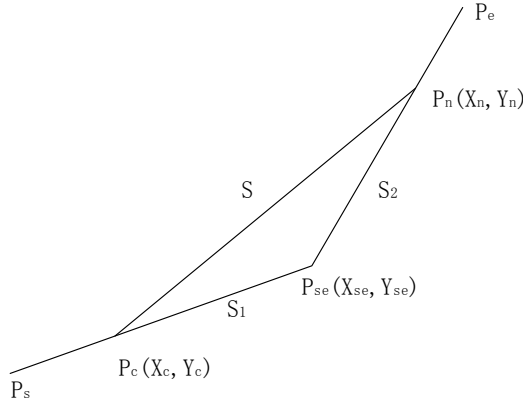


图 4.3 当前段和直线段转接

特别的，若线段 $P_{se}P_e$ 为水平线段时，有：

$$\begin{cases} Y_n = Y_e, \\ X_n = X_c \pm \sqrt{S^2 - (Y_n - Y_c)^2}, \end{cases} \quad (4.8)$$

式 (4.8) 中的 \pm 依然由式 (4.7) 来判定加还是减。 $P_{se}P_e$ 为垂直线段时可以同样进行讨论：

$$\begin{cases} X_n = X_e, \\ Y_n = Y_c \pm \sqrt{S^2 - (X_n - X_c)^2}, \\ (Y_n - Y_{se})(Y_n - Y_e) < 0, \end{cases} \quad (4.9)$$

确定点 $P_n(X_n, Y_n)$ 后，以 P_n 为直线段的起点进行 P_nP_e 段的插补。

4.3 S 型加减速算法速度规划

加减速控制是数控系统的核心技术。按照加减速型线的不同，可分为线性加减速和 S 型加减速等。线性加减速由于在加减速开始处与结束处速度均有突变，易引起抖动和振动，影响加工精度。S 型加减速控制可实现加减速过程中加速度的连续变化，过渡平滑，可有效减小冲击和振荡。故选择使用 S 型加减速

控制对实时加工控制算法进行优化。值得注意的是数控机床对加工刀具运动的控制是在各坐标轴方向上分别控制，然后合成为实际加工轨迹，所以本节中的 S 型曲线速度规划为针对各坐标轴上的速度的规划。

图 4.4 所示为 S 曲线加减速过程，整个过程由加加速、匀加速、减加速、匀速、加减速、匀减速及减减速七段组成。

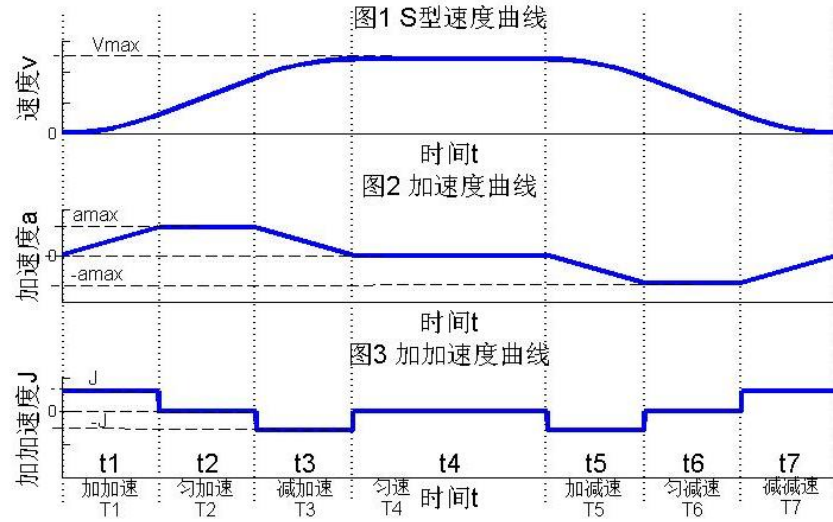


图 4.4 S 型曲线加减速过程

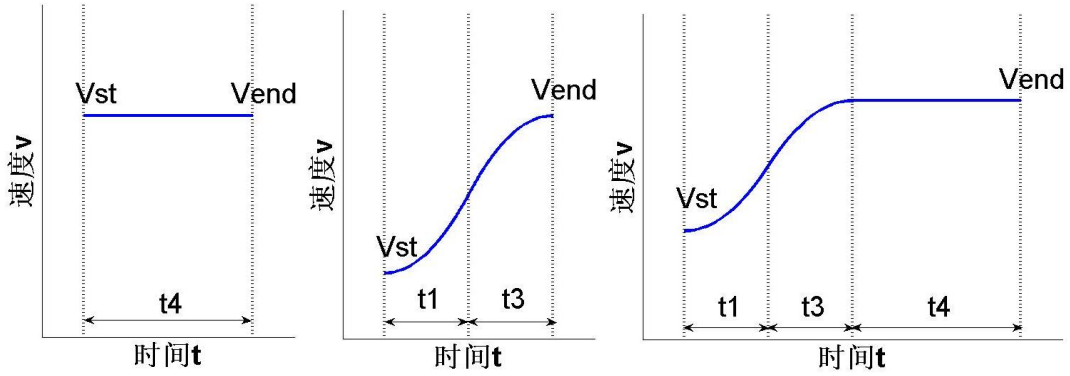
在起点速度与终点速度相等的情况下，其 S 型曲线加加速与减加速的时间相同，加减速与减减速的时间相同 ($t_1 = t_3$, $t_5 = t_7$)，可得出位移公式如下^[4]：

$$s(t) = \begin{cases} v_{st}t + \frac{1}{6}j_{max}t^3 & t \in [0, t_1) \\ s_1 + v_1t + \frac{1}{2}a_{max}t^2 & t \in [t_1, t_1 + t_2) \\ s_2 + v_2t + \frac{1}{2}a_{max}t^2 - \frac{1}{6}j_{max}t^3 & t \in [t_1 + t_2, 2t_1 + t_2) \\ s_3 + v_3t & t \in [2t_1 + t_2, 2t_1 + t_2 + t_4) \\ s_4 + v_4t - \frac{1}{6}j_{max}t^3 & t \in [2t_1 + t_2 + t_4, 2t_1 + t_2 + t_4 + t_5) \\ s_5 + v_5t - \frac{1}{2}a_{max}t^2 & t \in [2t_1 + t_2 + t_4 + t_5, 2t_1 + t_2 + t_4 + t_5 + t_6) \\ s_6 + v_6t - \frac{1}{2}a_{max}t^2 + \frac{1}{6}j_{max}t^3 & t \in [2t_1 + t_2 + t_4 + t_5 + t_6, 2t_1 + t_2 + t_4 + 2t_5 + t_6) \end{cases} \quad (4.10)$$

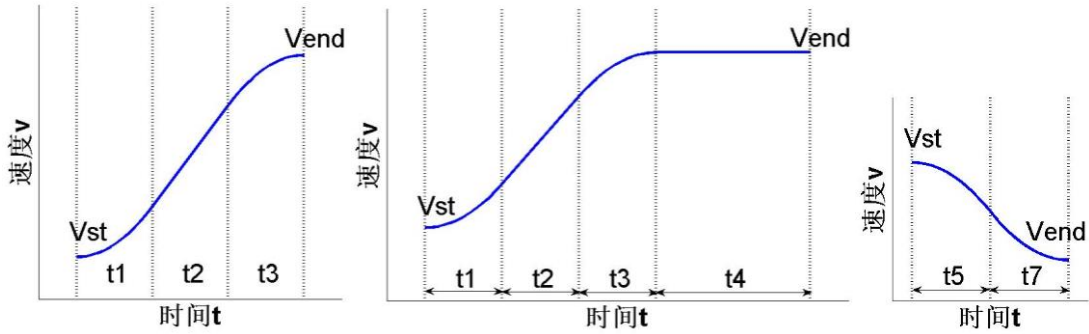
$$\left. \begin{aligned} s_1 &= v_{st}t_1 + \frac{1}{6}j_{max}t_1^3 \\ s_2 &= s_1 + v_1t_2 + \frac{1}{2}a_{max}t_2^2 \\ s_3 &= s_2 + v_2t_3 + \frac{1}{2}a_{max}t_3^2 - \frac{1}{6}j_{max}t_3^3 \\ s_4 &= s_3 + v_3t_4 \\ s_5 &= s_4 + v_4t_5 - \frac{1}{6}j_{max}t_5^3 \\ s_6 &= s_5 + v_5t_6 - \frac{1}{2}a_{max}t_6^2 \end{aligned} \right\} \quad (4.11)$$

式中, s_i 为起始点到各阶段点位移, $i = 1, 2, \dots, 6$; t_i 为图 XX 中 7 段加减速各阶段持续时间, $i = 1, 2, \dots, 7$, 其中 $t_1 = t_3, t_5 = t_7$; v_i 为各阶段起始速度, $i = 1, 2, \dots, 6$; a_{max} 为最大加速度; j_{max} 为最大加加速度。

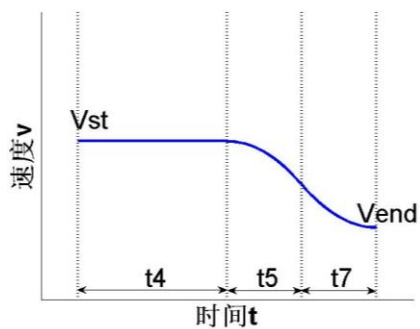
在曲线速度规划中, 需要得到各曲线段的参数, 如曲线段速度最大值 v_{max} 、起点速度 v_{st} 和终点速度 v_{end} , 进一步分析判断其速度曲线类型是属于加速、减速或包含加速和减速, 再采用 S 曲线加减方法对其重新进行插补进给速度规划, 由此可以得到图 4.5 所示的 17 种 S 曲线加减速类型。



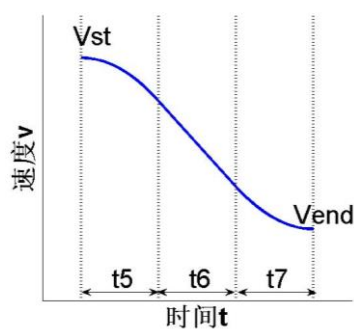
(a) 只包含匀速段 (b) 只包含加加速段和减加速 (c) 包含匀速段, 不含匀加速、减速段



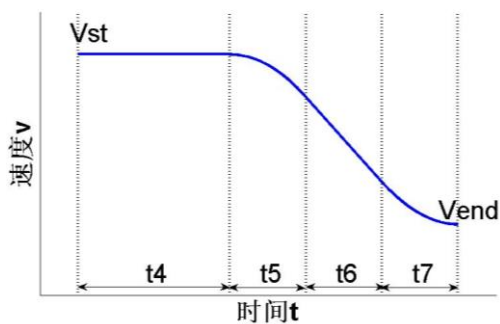
(d) 含匀加速段, 不含匀速、减速段 (e) 含匀加速段, 不含减速段 (f) 只包含加减速段和减减速段



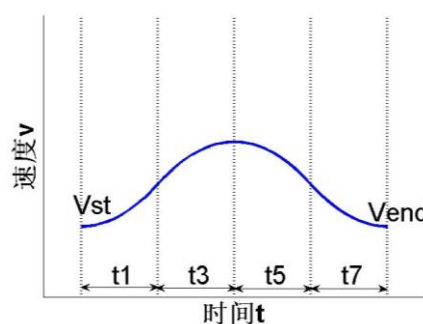
(g) 只包含匀速段、加减速段和减减速段



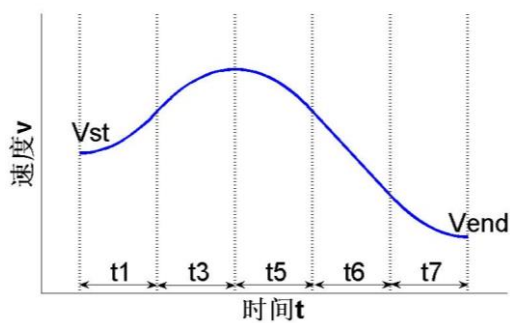
(h) 包含匀减速段, 不含加速段、匀速段



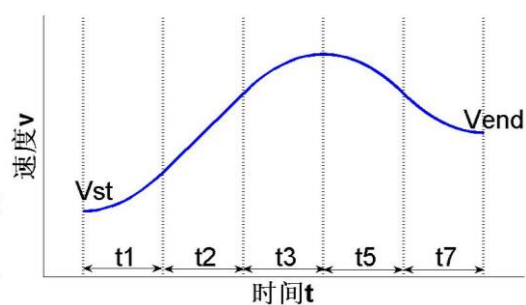
(i) 包含匀速段、匀减速段, 不含加速段



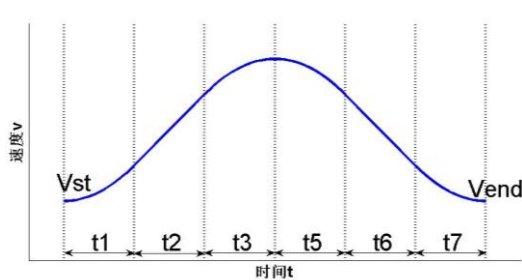
(j) 不含匀加速段、匀速段和匀减速段



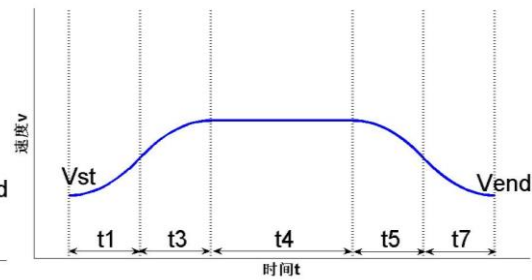
(k) 包含匀减速段, 不含匀速段、匀加速段



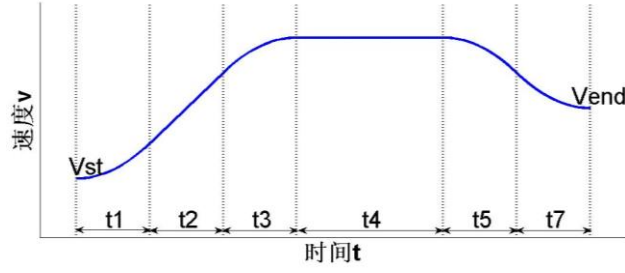
(l) 包含匀加速段, 不含匀速段、匀减速段



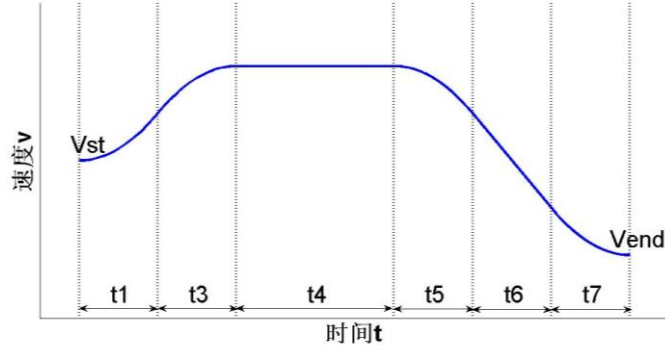
(m) 包含匀加速段、匀减速段, 不含匀速段



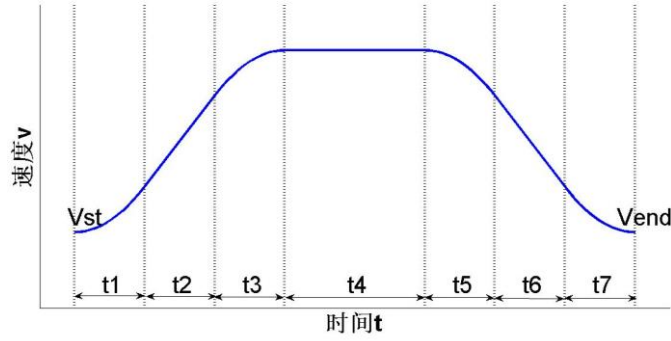
(n) 包含匀速段, 不含匀加速段、匀减速段



(o) 包含匀加速段、匀速段,不含匀减速段



(p) 包含匀速段、匀减速段,不含匀加速段



(q) 包含匀加速段、匀速段、匀减速段

图 4.5 不同曲线段参数条件下的 17 种 S 曲线加减速类型

重新规划插补进给速度的过程主要包括以下两个步骤:

(1) 根据各曲线段长度 L_s 、起点速度 v_{st} 、终点速度 v_{end} 、速度最大值 v_{max} 判断当前曲线段的加减速类型。下面先以图 4.5b~图 4.5e 类型范围为例说明仅包含加速段加减速类型判断过程。将曲线段起点速度 v_{st} 、最大加速度 a_{max} 及最大加加速度 j_{max} 代入位移计算公式 (4.10), 得到位移长度的表达式为:

$$S = v_{st}t_1 + \left(v_{st} + \frac{1}{2}j_{max}t_1^2\right)t_2 + \frac{1}{2}a_{max}(t_2^2 + t_3^2) + (v_{st} + \frac{1}{2}j_{max}t_1^2 + a_{max}t_2)t_3 \quad (4.12)$$

当 $v_{max} - v_{st} \leq a_{max}^2/j_{max}$ 时, 取:

$$a_{max} = \sqrt{(v_{max} - v_{st})/j_{max}} \quad (4.13)$$

此时为加速过程中无匀加速段的临界条件, 加速过程中的时间分别为:

$$\begin{cases} t_1 = \sqrt{(v_{\max} - v_{st})/j_{\max}} \\ t_2 = 0 \\ t_3 = t_1 \end{cases} \quad (4.14)$$

由上式计算可得到 t_1 、 t_2 和 t_3 ，代入位移长度公式计算出 S 的理论值 S_1 。比较 L_s 与 S_1 大小，当 $L_s > S_1$ 时为图 4.5c 类型；当 $L_s \leq S_1$ 时为图 4.5b 类型。

当 $v_{\max} - v_{st} > a_{\max}^2/j_{\max}$ 时，则加速过程中有匀加速段，加速过程中的时间分别为：

$$\begin{cases} t_1 = a_{\max}/j_{\max} \\ t_2 = \frac{v_{\max} - v_{st}}{a_{\max}} - t_1 \\ t_3 = t_1 \end{cases} \quad (4.15)$$

由上式计算可得 t_1 、 t_2 和 t_3 ，代入位移长度公式计算出 S 的理论值 S_{11} ，同时，将不包含匀加速段的临界条件 $t_1 = t_3 = a_{\max}/j_{\max}$ 及 $t_2 = 0$ 代入位移长度公式中，计算出不包含匀加速段的轨迹理论长度极值 S_2 。比较 L_s 与 S_{11} 、 S_2 的大小，当 $L_s > S_{11}$ 时为图 4.5e 类型；当 $S_2 < L_s \leq S_{11}$ 时为图 4.5d 类型；当 $L_s \leq S_2$ 时为图 4.5b 类型。

同理，再以图 4.5f~图 4.5i 类型范围为例说明仅包含减速段加减速类型判断过程。将曲线段起点速度 v_{st} 、最大加速度 a_{\max} 及最大加加速度 j_{\max} 代入 $t_5 \sim t_7$ 时间段的位移计算式，对应图 4.5f~图 4.5i 类型范围的位移长度，得到位移长度的表达式为：

$$\begin{aligned} S' = & v_{\max}t_5 + \left(v_{\max} - \frac{1}{2}j_{\max}t_5^2\right)t_6 - \frac{1}{2}a_{\max}(t_6^2 + t_7^2) + (v_{\max} - \frac{1}{2}j_{\max}t_5^2 \\ & - a_{\max}t_6)t_7 \end{aligned} \quad (4.16)$$

当 $v_{\max} - v_{end} \leq a_{\max}^2/j_{\max}$ 时，取：

$$a_{\max} = \sqrt{(v_{\max} - v_{end})/j_{\max}} \quad (4.17)$$

此时为减速过程中无匀减速段的临界条件，减速过程中的时间分别为：

$$\begin{cases} t_5 = \sqrt{(v_{\max} - v_{end})/j_{\max}} \\ t_6 = 0 \\ t_7 = t_5 \end{cases} \quad (4.18)$$

由上式计算可得到 t_5 、 t_6 和 t_7 ，代入位移长度公式计算出 S' 的理论值 S'_1 。比较 L_s 与 S'_1 大小，当 $L_s > S'_1$ 时为图 4.5g 类型；当 $L_s \leq S'_1$ 时为图 4.5f 类型。

当 $v_{\max} - v_{end} > a_{\max}^2/j_{\max}$ 时，则减速过程中有匀减速段，减速过程中的时间分别为：

$$\begin{cases} t_5 = a_{\max}/j_{\max} \\ t_6 = \frac{v_{\max} - v_{end}}{a_{\max}} - t_5 \\ t_7 = t_5 \end{cases} \quad (4.19)$$

由上式计算可得 t_5 、 t_6 和 t_7 ，代入位移长度公式计算出 S' 的理论值 S'_{11} ，同时，将不包含匀减速段的临界条件 $t_5 = t_7 = a_{\max}/j_{\max}$ 及 $t_6 = 0$ 代入位移长度公式中，计算出不包含匀减速段的轨迹理论长度极值 S'_2 。比较 L_s 与 S'_{11} 、 S'_2 的大小，当 $L_s > S'_{11}$ 时为图 4.5i 类型；当 $S'_2 < L_s \leq S'_{11}$ 时为图 4.5h 类型；当 $L_s \leq S'_2$ 时为图 4.5f 类型。

最后,综合上述两种规划方法,判断任意一段或多段情况下的加减速类型。在已知曲线段长度 L_s 、起点速度 v_{st} 、终点速度 v_{end} 、速度最大值 v_{max} 、最大加速度 a_{max} 及最大加加速度 j_{max} 的情况下,仿照先前两种情况代入位移长度公式分别求出 S_1 、 S_{11} 和 S_2 和 S'_1 、 S'_{11} 和 S'_2 ,若求不出则空缺。然后再根据速度先进行第一轮判断:

若 $v_{st} < v_{end}$,当 $L_s < S_1$ 时,令 $v_{max} = v_{end}$,可求出 t_1 、 t_2 和 t_3 ,并对应图 4.5b 类型;当 $S_2 < L_s \leq S_{11}$ 时,令 $v_{max} = v_{end}$,可求出 t_1 、 t_2 和 t_3 ,并对应图 4.5d 类型。

若 $v_{st} > v_{end}$,当 $L_s < S'_1$ 时,令 $v_{max} = v_{st}$,可求出 t_5 、 t_6 和 t_7 ,并对应图 4.5f 类型;当 $S'_2 < L_s \leq S'_{11}$ 时,令 $v_{max} = v_{st}$,可求出 t_5 、 t_6 和 t_7 ,并对应图 4.5h 类型。

当上述情况都不满足,则进入第二轮判断:

若计算包含匀加速段和匀减速段 S_{11} 、 S'_{11} 、 S_2 和 S'_2 ,当 $L_s > S_{11} + S'_{11}$ 时,对应图 4.5q 类型;当 $S_2 + S'_2 < L_s \leq S_{11} + S'_{11}$ 时,对应图 4.5m 类型;当 $L_s \leq S_2 + S'_2$ 时,对应图 4.5j 类型。

若计算出为不包含匀加速段和匀减速段,当 $L_s > S_1 + S'_1$ 时,对应图 4.5n 类型;当 $L_s \leq S_1 + S'_1$ 时,对应图 4.5j 类型。

若计算出为包含匀加速段,不包含匀减速段,当 $L_s > S_{11} + S'_1$ 时,对应图 4.5o 类型;当 $S_2 + S'_1 < L_s \leq S_{11} + S'_1$ 时,对应图 4.5l 类型;当 $L_s \leq S_2 + S'_1$ 时,对应图 4.5j 类型。若计算出为不包含匀加速段,包含匀减速段,当 $L_s > S_1 + S'_{11}$ 时,对应图 4.5p 类型;当 $S_1 + S'_2 < L_s \leq S_1 + S'_{11}$ 时,对应图 4.5k 类型;当 $L_s \leq S_1 + S'_2$ 时,对应图 4.5j 类型。

至此,列举出的所有类型的判断加减速类型方法已全部提出。

(2) 在根据上述步骤判断出当前段加减速类型后,可根据位移长度公式对时间 t 求一阶和二阶导数,分别得到速度及加速度方程,进而可求出加速、匀速及减速各部分时间。

以图 4.5j 类型为例,根据位移长度公式计算各部分需要时间,经化简得:

$$v_{max} = v_{st} + j_{max}t_1^2 = v_{end} + j_{max}t_5^2 \quad (4.20)$$

$$L_s = (v_{st} + v_{max})t_1 + (v_{end} + v_{max})t_5 \quad (4.21)$$

由公式 (4.20) 得:

$$t_5 = \sqrt{t_1^2 + \frac{v_{st}-v_{end}}{j_{max}}} \quad (4.22)$$

将公式 (4.20)、公式 (4.22) 代入公式 (4.21) 得:

$$L_s = (2v_{st} + j_{max}t_1^2)t_1 + (v_{end} + v_{st} + j_{max}t_1^2)\sqrt{t_1^2 + \frac{v_{st}-v_{end}}{j_{max}}} \quad (4.23)$$

求解公式 (4.23) 可计算得到 t_1 ,将 t_1 代入公式 (4.22) 计算得到 t_5 。

同理,该方法也可推导出其他各种加减速类型的各部分运动时间 t_1 、 t_2 、 t_3 、 t_4 、 t_5 、 t_6 和 t_7 。至此,S 曲线加减速规划全部完成。详细流程图如图 4.6 所示,详情请参见附录 I。

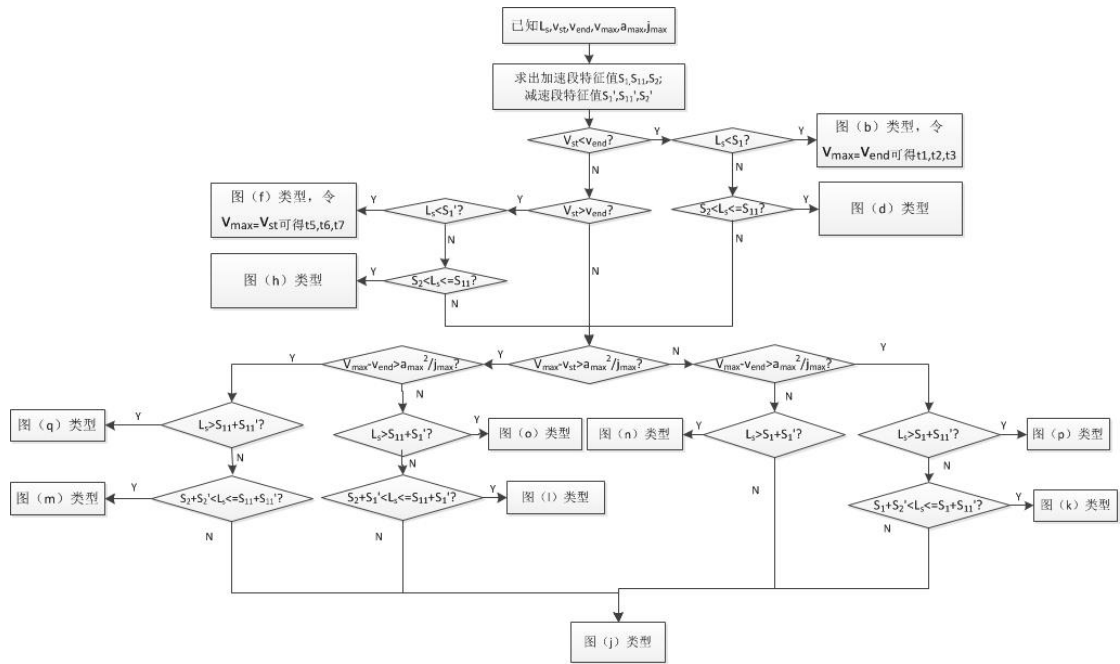


图 4.6 S 曲线加减速类型判断流程图

4.4 S 型加减速插补算法

插补的实质是把连续的曲线离散化的过程，所以插补算法的加减速控制的运动参数是离散的。本文中采用事件分割法进行插补，下面将阐述插补的具体实现过程。

时间分割法的工作方式为：用插补周期为时间单位，将加工过程分割成许多个单位时间内的进给过程。插补计算的主要任务是计算出下一个插补点的坐标从而计算出轮廓速度 f 在各个坐标轴的分量，进而可以得到下一个插补周期内各个坐标轴方向的进给量，在进给过程中，对实际位置进行采样，与插补计算的坐标值进行比较，得出位置误差，位置误差将在下一个采样周期内修正，采样周期可以等于插补周期也可以小于插补周期。

在 4.3 节中详细介绍了 S 型加减速算法的速度规划，根据机床的速度、加速度以及加加速度的限制，用户设定的进给速度和加工线段的长度，我们可以明确得出加工线段 7 个阶段的时间，绘制出 S 型加减速控制曲线。下面我们以某直线为例，详细介绍 S 型加减速插补算法的实现过程。

4.4.1 理论插补点

假设该直线段起点为 $P_s(x_s, y_s)$ ，终点为 $P_e(x_e, y_e)$ ；其与 $+x$ 轴的夹角为 α ， $\alpha < 45^\circ$ ，则其长轴为 x 轴；对该直线在 x 轴的投影进行 S 型加减速规划，即 $|x_e - x_s|$ 段的长度进行速度规划，设 $L = |x_e - x_s|$ 。该直线的起始速度为 v_{st} ，终点速度为 v_{end} ， v_{end} 的值由 (4.5) 式决定。规划后得知该直线包含 S 型加减速插补算法的全部 7 个阶段，时间分别为 $t_1 \sim t_7$ 。

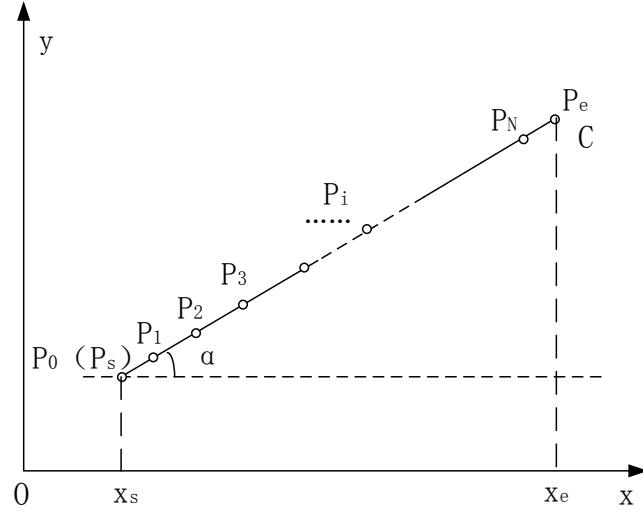


图 4.7 理论插补点

假设插补周期为 T ，速度规划走完整个直线段的时间应为：

$$t_{all} = \sum_{i=1}^7 t_i \quad (4.24)$$

其中 t_{all} 一般不是插补周期为 T 的整数倍，因此可以将 t_{all} 写为以下形式：

$$t_{all} = NT + t_r \quad (t_r < T) \quad (\text{其中 } N \text{ 为正整数})。$$

从上式可以看出，加工改线段的过程中共要进行 N 次插补。

按照速度规划的理想状况进行计算，将直线段的起点 $P_s(x_s, y_s)$ 记为 P_0 ，第 i 次插补的理想目标点为 $P_i(x_i, y_i) (i \leq N)$ (如图 4.7 所示)。由于已知 $|x_e - x_s|$ 段的 S 型加减速度的位移方程 $S(t)$ ，则有：

$$\begin{cases} x_i - x_s = S(iT) \\ \tan \alpha = \frac{y_i - y_s}{x_i - x_s} \end{cases} \quad (4.25)$$

化简后得：

$$\begin{cases} x_i = S(iT) + x_s \\ y_i = \tan \alpha \cdot S(iT) + y_s \end{cases} \quad (4.26)$$

由此可以得到 $P_i(x_i, y_i)$ 的准确坐标。

4.4.2 实际插补点及插补算法

由于每一机床都有对应的分辨率 d ，每个插补周期的进给量 Δx 、 Δy 的长度一定是分辨率的整数倍，也就是说，刀具的运动方向受限制，因此实际的插补点并无法到达理论的插补点 $P_i(x_i, y_i)$ ，两者之间存在误差，设第 i 次插补的实际插补点为 $Q_i(x_i^*, y_i^*)$ 。

起点处不存在误差，即 $Q_0 = P_0$ 。

1) 第一个插补周期坐标增量计算：

第一个插补周期的理论目标点为 $P_1(x_1, y_1)$ ，则第一个插补周期坐标的理论

增量为：

$$\Delta x_1 = x_1 - x_0 = m_1 d, \Delta y_1 = y_1 - y_0 = k_1 d \quad (d \text{ 为机床分辨率}) \quad (4.27)$$

通常来说 m_1 、 k_1 不是整数，将 m_1 、 k_1 分别向上取整得 M_1 、 K_1 ，则第一个插补周期的实际坐标增量为：

$$\Delta x_1^* = M_1 d, \Delta y_1^* = K_1 d \quad (M_1、K_1 \text{ 均为整数}) \quad (4.28)$$

很显然，通过简单的平面几何方法可以验证该插补方法所造成的误差小于一个分辨率 d ，自然满足加工的精度要求。在进行直线段的插补时，误差越小，实际路程则越短，则速度越快越平稳。

2) 第二个插补周期坐标增量计算：

第二个插补周期开始时刻，刀具实际位置位于点 $Q_1(x_1^*, y_1^*)$ ，第二个插补周期的理论目标点为 $P_2(x_2, y_2)$ ，则第二个插补周期坐标的理论增量为：

$$\Delta x_2 = x_2 - x_1^* = m_2 d, \Delta y_2 = y_2 - y_1^* = k_2 d \quad (4.29)$$

进行步骤 1) 中类似的向上取整，得到第二个插补周期的实际坐标增量为：

$$\Delta x_2^* = M_2 d, \Delta y_2^* = K_2 d \quad (M_2、K_2 \text{ 均为整数}) \quad (4.30)$$

3) 第 i 个插补周期坐标增量计算：

第 i 个插补周期开始时刻，刀具实际位置位于点 $Q_{i-1}(x_{i-1}^*, y_{i-1}^*)$ ，第二个插补周期的理论目标点为 $P_i(x_i, y_i)$ ，则第 i 个插补周期坐标的理论增量为：

$$\Delta x_i = x_i - x_{i-1}^* = m_i d, \Delta y_i = y_i - y_{i-1}^* = k_i d \quad (4.31)$$

进行前两步中类似的向上取整，得到第 i 个插补周期的实际坐标增量为：

$$\Delta x_i^* = M_i d, \Delta y_i^* = K_i d \quad (M_i、K_i \text{ 均为整数}) \quad (4.32)$$

4) 循环上述算法直至第 N 个插补周期。

5) 进行轨迹段转接处理算法

由于 t_r 不足一个插补周期 T ，因此若是按照原方向继续插补，则会超过线段终点 $P_e(x_e, y_e)$ ，因此在该点应用 4.2 节中所讨论的轨迹段转接处理算法，求得下一直线段上的插补点 R_0 。已知理论折点 $P_N(x_N, y_N)$ ，线段终点 $P_e(x_e, y_e)$ (即理论转折点) 和下一段直线段的斜率 k ，只需得知转折处插补周期内插补线段的长度 $|P_N R_0|$ ，就可以求得下一线段上插补点 R_0 的坐标。

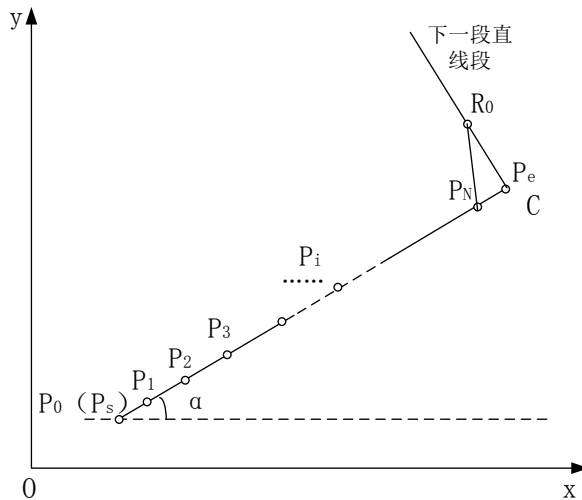


图 4.8 轨迹段转接处插补示意图

为对加工时间进行优化，在转折点处保持合成速率不变。对 x 轴的速度规划终点速度为 v_{end} ，则合成的速率 $v_s = \left| \frac{v_{end}}{\cos \alpha} \right|$ 。插补线段的长度 $|P_N R_0|$ 由下式给出：

$$|P_N R_0| = v_s \cdot T \quad (T \text{ 为一个插补周期}) \quad (4.33)$$

- 6) 以 R_0 点作为下一直线段的起始点重复 1)~5) 步骤进行下一直线段的速度规划与插补。

4.5 实时加工控制算法流程

综上所述，实时加工控制算法流程图绘制如图 4.9 所示：

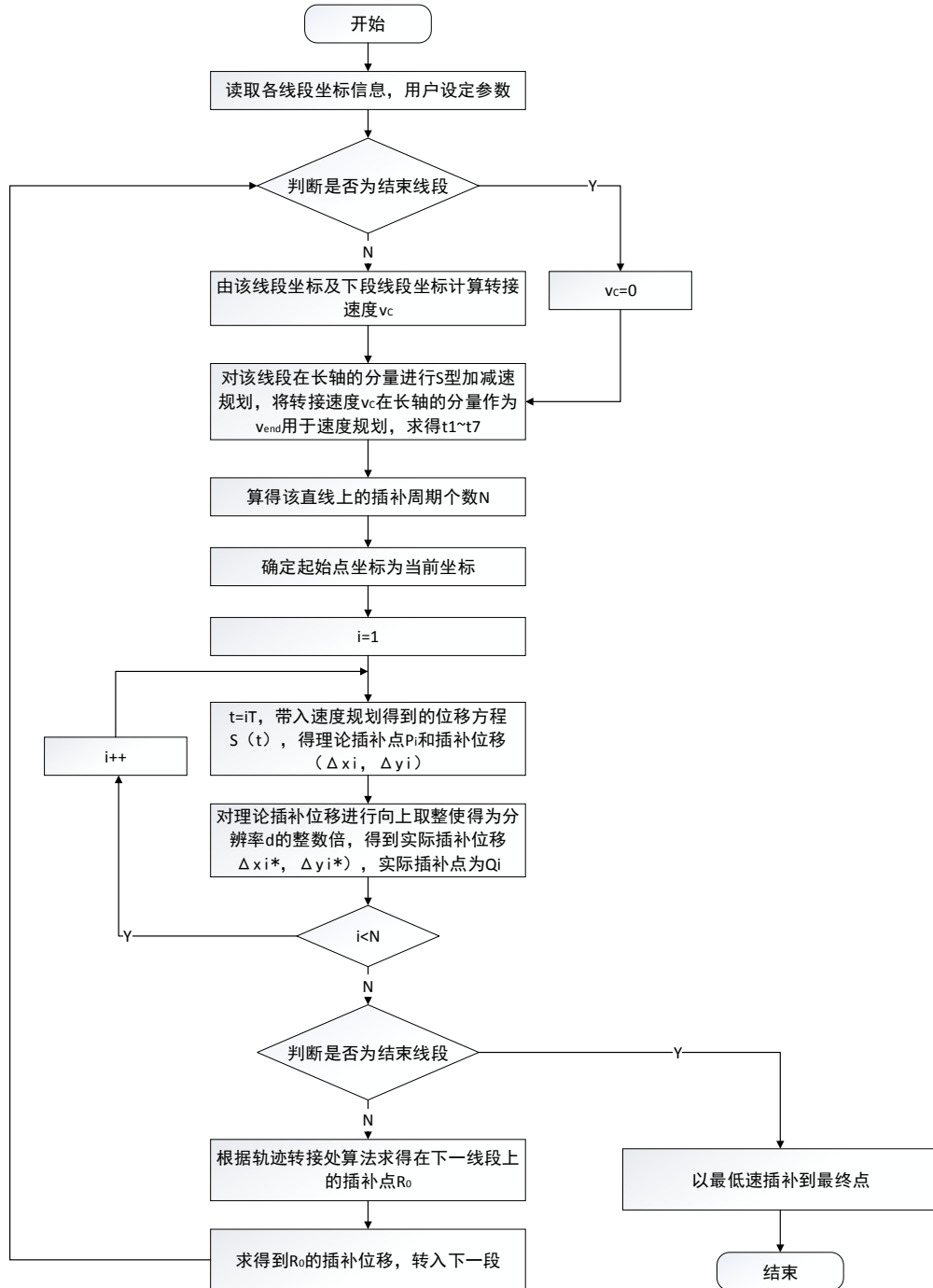


图 4.9 实时加工控制算法流程图

4.6 折线夹角与转接点速度讨论

已知应用上述限制条件计算出的转接速度可以实现轨迹段的连续加工，现针对加工型线为折线时，讨论在其不同夹角的情况下，通过折线交点时对应各坐标运动速度的变化情况。

假设在 xy 平面内，相邻两段加工轨迹 $L1$ 和 $L2$ 如图 4.10 所示，轨迹段转接点为 C 。依照加工方向，将 $L1$ 和 $L2$ 看成是矢量线段，其中以 x 正轴作为初始边，逆时针旋转到 $L1$ 形成夹角 α ，逆时针旋转到 $L2$ 形成夹角 β ，规定 $0 \leq$

$\alpha, \beta \leq 360^\circ$ ，则夹角 $0 \leq \gamma \leq 180^\circ$ 。

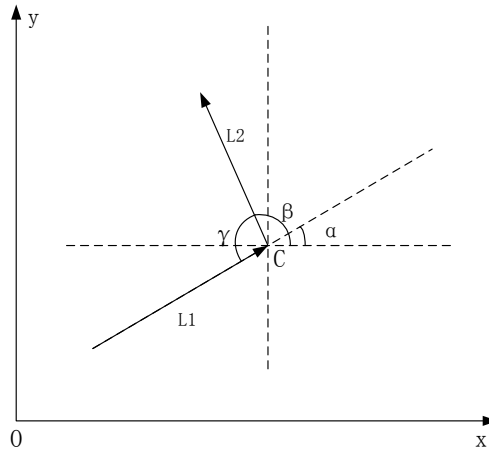


图 4.10 折线加工型线几何关系

因通过折线交点所需时间较短，假设经过交点前后的转接速度 v_c 速率相同，根据图 4.10 所示几何关系，可得出交点前各坐标轴运动速度分量分别为：

$$v_{xp} = v_c \cos \alpha \quad (4.34)$$

$$v_{yp} = v_c \sin \alpha \quad (4.35)$$

交点后各坐标轴运动速度分量分别为：

$$v_{xa} = v_c \cos \beta \quad (4.36)$$

$$v_{ya} = v_c \sin \beta \quad (4.37)$$

联立上述公式可以得出各坐标轴速度变化量分别为：

$$\Delta v_x = |v_c \cos \beta - v_c \cos \alpha| = 2v_c \left| \sin \frac{\beta + \alpha}{2} \sin \frac{\beta - \alpha}{2} \right| \quad (4.38)$$

$$\Delta v_y = |v_c \sin \beta - v_c \sin \alpha| = 2v_c \left| \cos \frac{\beta + \alpha}{2} \sin \frac{\beta - \alpha}{2} \right| \quad (4.39)$$

根据几何关系，折线夹角 $\gamma = 180 - (\beta - \alpha)$ ，故上述公式可转化为：

$$\Delta v_x = 2v_c \cos \frac{\gamma}{2} \left| \cos \left(\alpha - \frac{\gamma}{2} \right) \right| \quad (4.40)$$

$$\Delta v_y = 2v_c \cos \frac{\gamma}{2} \left| \sin \left(\alpha - \frac{\gamma}{2} \right) \right| \quad (4.41)$$

根据题意，当 $\gamma = 90^\circ$ 时，各坐标轴速度变化量为：

$$\Delta v_{x1} = \sqrt{2}v_c |\cos(\alpha - 45^\circ)| \quad (4.42)$$

$$\Delta v_{y1} = \sqrt{2}v_c |\sin(\alpha - 45^\circ)| \quad (4.43)$$

根据 α 的取值范围与 Δv_{x1} 的单调性可知，当 α 处于 $[0, 45^\circ]$ 、 $[135^\circ, 225^\circ]$ 、 $[315^\circ, 360^\circ]$ 区间时， Δv_{x1} 逐渐增大；当 α 处于 $[45^\circ, 135^\circ]$ 、 $[225^\circ, 315^\circ]$ 区间时， Δv_{x1} 变化量逐渐减小。

根据 α 的取值范围与 Δv_{y1} 的单调性可知，当 α 处于 $[0, 45^\circ]$ 、 $[135^\circ, 225^\circ]$ 、 $[315^\circ, 360^\circ]$ 区间时， Δv_{y1} 逐渐减小；当 α 处于 $[45^\circ, 135^\circ]$ 、 $[225^\circ, 315^\circ]$ 区间时， Δv_{y1} 逐渐增大。

同理，当 $\gamma = 135^\circ$ 时，各坐标轴速度变化量为：

$$\Delta v_{x2} = 2v_c \cos 67.5^\circ |\cos(\alpha - 67.5^\circ)| \quad (4.44)$$

$$\Delta v_{y2} = 2v_c \cos 67.5^\circ |\sin(\alpha - 67.5^\circ)| \quad (4.45)$$

根据 α 的取值范围与 Δv_{x2} 的单调性可知，当 α 处于 $[0, 67.5^\circ]$ 、 $[157.5^\circ, 247.5^\circ]$ 、 $[337.5^\circ, 360^\circ]$ 区间时， Δv_{x2} 逐渐增大；当 α 处于 $[67.5^\circ, 157.5^\circ]$ 、

[247.5°, 337.5°]区间时, Δv_{x2} 逐渐减小。

根据 α 的取值范围与 Δv_{y2} 的单调性可知, 当 α 处于 [0, 67.5°]、[157.5°, 247.5°]、[337.5°, 360°] 区间时, Δv_{y2} 逐渐减小; 当 α 处于 [67.5°, 157.5°]、[247.5°, 337.5°] 区间时, Δv_{y2} 逐渐增大。

虽然各坐标轴上的速度分量随着夹角的变化情况有增有减, 但是根据公式 XX 可以得出合成速度的变化量为:

$$\Delta v = \sqrt{\Delta v_x^2 + \Delta v_y^2} = 2v_c \cos \frac{\gamma}{2} \quad (4.46)$$

可见, 随着两条轨迹段夹角的减小, 转接速度的变化量会逐渐增大。因此如何确定两相邻轨迹线的转接轨迹从而使转接角减小, 将成为提高转接速度的关键。

5 直线段和圆弧实时加工控制算法模型

5.1 转接速度限制条件

5.1.1 加速度限制

当被插补圆弧半径较大时, 因为插补步长比圆弧半径的长度小得多, 相邻两插补步长的转角很大, 不需要考虑加速度限制。当被插补圆弧半径较小时, 有必要考虑加速度限制。圆弧插补示意图如下图所示, 图中 DF 、 FE 为两相邻插补步长, 插补速度由第 4 节中的公式 (4.2) 决定。

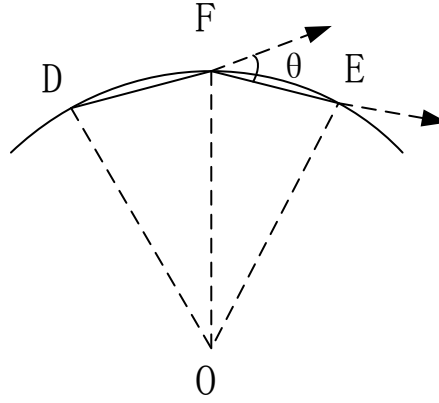


图 5.1 圆弧插补示意图

令加速度 $a = \min(a_{x-\max}, a_{y-\max})$, 由加速度所限制的速度 v_1 满足:

$$v_1 \leq \frac{aT}{2 \left| \sin \frac{\theta}{2} \right|} \quad (5.1)$$

即可。

又因为:

$$\left| \sin \frac{\theta}{2} \right| = \left| \sin \frac{180^\circ - 2\angle DFO}{2} \right| = |\cos \angle DFO| = \frac{v_1 T}{2R} \quad (5.2)$$

故:

$$v_1 \leq \sqrt{aR} \quad (5.3)$$

5.1.2 圆弧弓高误差限制

在圆弧轨迹插补中，用小段直线逼近圆弧，所以加工轨迹存在弓高误差 E_R 。假设进给速度为 v_2 ，插补步长 $|MN| = v_2 T$ ，圆弧半径为 R ，系统允许的最大加工误差为 ε ，则有：

$$\begin{aligned} E_R &= R - \sqrt{R^2 - v_2^2 T^2 / 2} \leq \varepsilon \\ v_2 &\leq \frac{2R}{T} \sqrt{1 - (1 - \frac{\varepsilon}{R})^2} \end{aligned} \quad (5.4)$$

舍去二次项后为：

$$v_2 \leq \frac{\sqrt{8R\varepsilon}}{T} \quad (5.5)$$

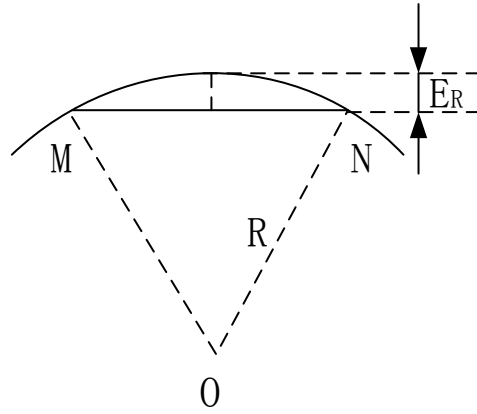


图 5.2 圆弧弓高误差

5.1.3 多步转接条件限制

为了使圆弧过渡有效，将圆弧分散到多个插补周期中，在圆周上至少应有两个以上的插补点，设圆弧的弧长为 $L_c = R\varphi$ (φ 为圆弧对应的圆心角)，即要求圆弧的插补速度 v_3 满足：

$$\begin{aligned} \frac{L_c}{v_3 T} &\geq 2 \\ v_3 &\leq \frac{R\varphi}{2T} \end{aligned} \quad (5.6)$$

综上所述，圆弧的转接速度应取 (5.3)、(5.5)、(5.6) 以及用户设定速度 F 中的最小值，即圆弧的转接速度 v_{cir} 由下式决定：

$$v_{cir} = \min(v_1, v_2, v_3, F) \quad (5.7)$$

5.2 圆弧转接处理算法

5.2.1 直线段转接圆弧

基于和 4.2 节中分析的同样的原因，在直线和圆弧转接点处需要在圆弧 C

上计算实际的转接点 P_n ，如图 5.3 所示。

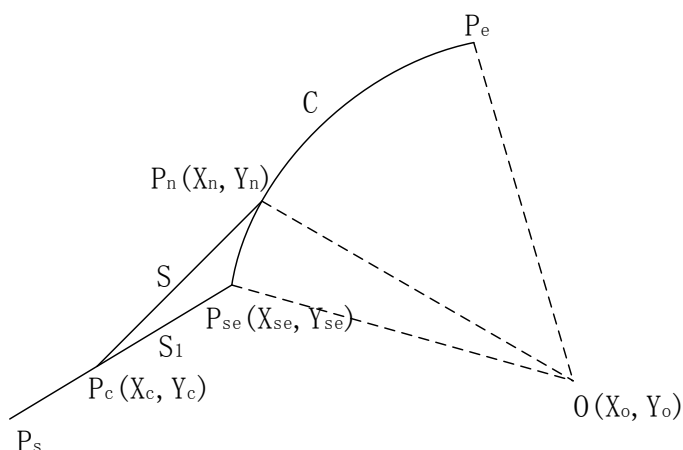


图 5.3 直线段和圆弧段转接处理算法示意图

转接点的横、纵坐标满足以下条件:

$$\begin{cases} (X_n - X_c)^2 + (Y_n - Y_c)^2 = \hat{S} \\ (X_n - X_o)^2 + (Y_n - Y_o)^2 = \hat{R} \end{cases} \quad (5.8)$$

其中： R 为圆弧 C 的半径， (X_o, Y_o) 为圆心坐标。

解上面的方程组得到 P_n 的两组解, 由 $\angle P_n O P_e \leq \angle P_{se} O P_e$ 用作限制条件舍去一组解从而得到唯一解。确定 P_n 后, 以 P_n 为圆弧段 C 的起点进行下一段圆弧的插补。

5.2.2 圆弧转接直线段

圆弧转接直线的处理算法示意图如图 5.4 所示。

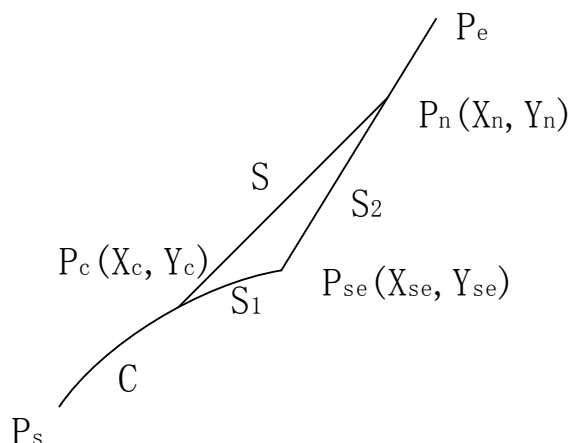


图 5.4 圆弧转接直线段的处理算法示意图

由该图可以看出, 圆弧转接直线段的处理算法和 4.2 节中直线段与直线段转接的算法类似。已知 P_c 点坐标, P_{se} 点坐标, 线段 $P_{se}P_e$ 的斜率和插补直线段的长度 S , 由 4.2 节中的方法即可求得直线段上的插补点 P_n 的坐标。

5.3 直线段和圆弧实时加工控制算法

5.3.1 S 型速度规划

当加工型线为直线段和圆弧连接时，圆弧段的插补速度的值也就是速率保持不变，变化的仅仅是速度的方向，因此圆弧段不需要进行速度规划，仅需要根据 5.1 节中介绍的方法求得圆弧段的插补速度 v_{cir} 。要对直线段部分进行 S 型速度规划，圆弧段的插补速度 v_{cir} 就是直线段部分速度规划的终速度 v_{end} 。规划的方法见 4.3 小节。

当加工型线为圆弧和直线段连接时，圆弧的插补速度 v_{cir} 即为下一段直线段的起始速度，直线段的速度规划过程与 4.3 节中的规划方法完全相同。

本题中不考虑瞬时启动加速度及瞬时启动速度，因此机床在启动时的起始速度和起始加速度是从 0 开始的。

5.3.2 直线和圆弧加工控制算法流程

直线和圆弧加工控制算法流程图如图 5.5 所示

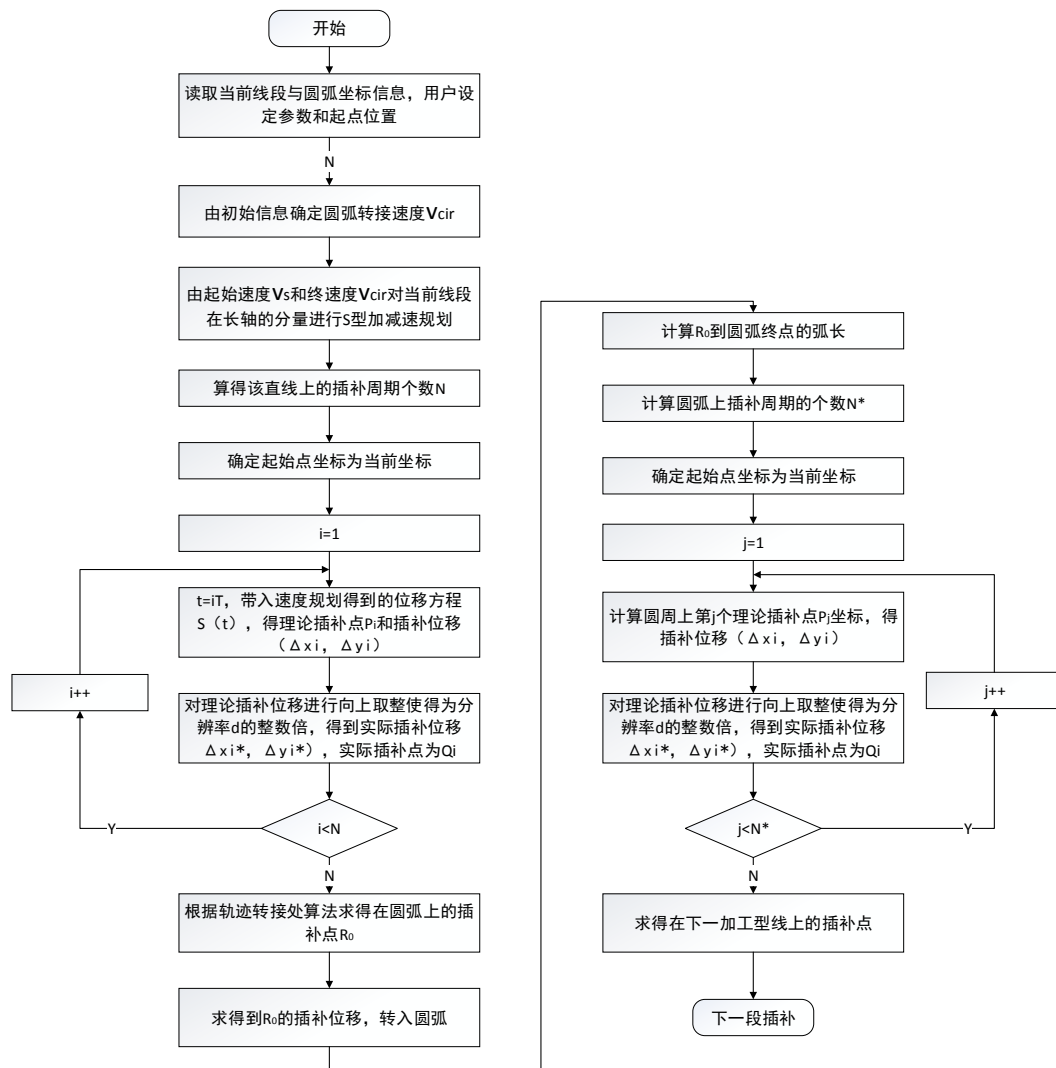


图 5.5 直线和圆弧加工控制算法流程图

5.4 圆弧半径对算法效率的影响

圆弧处的插补以恒定的速率进行，根据 5.1 节中的分析可知，圆弧处的插补速度 v_{cir} 由式 (5.7) 决定：

$$v_{cir} = \min(v_1, v_2, v_3, F) = \min(\sqrt{aR}, \frac{\sqrt{8R\varepsilon}}{T}, \frac{R\varphi}{2T}, F) \quad (5.9)$$

由此可见，式 (5.9) 中 \min 函数的前三项均与圆弧的半径有关。

如果圆弧的半径足够大，使得 $v_{cir} = F$ ，则从直线过渡到圆弧的过程中，不需要进行减速，此时加工的效率最大。

如果圆弧的半径较小， v_{cir} 的值由 \min 函数的前三项决定， $v_{cir} \leq F$ ，则在直线过渡到圆弧的插补过程中需要进行减速，并且圆弧半径越小，则 v_{cir} 的值越小，从直线过渡到圆弧的过程中进行减速的过程也越长，加工效率越低。

综上所述，圆弧半径越大，加工的效率越高；圆弧半径越小，加工的效率越低。

5.5 加工示例仿真验证

所要加工的图形为一个圆角矩形，路径的四个角是半径为 0.5 的整圆的 1/4 圆弧。矩形外围大小是：41×41（单位：cm²）。

在加工过程中不考虑瞬时启动加速度及瞬时启动速度对加工过程的影响，对题目中提供的最大速度的要求进行修改后的表格如表 5.1 所示：

表 5.1 圆角矩形切割路径加工中速度要求

路径节点	节点坐标(X、Y、Z)	用户设定的最大频率 F	最大频率 F 转换后对应的速度
1	(-20.500,-20.000,0.000)	2773	0.0
2	(-20.500,-17.708,0.000)	4029	0.19
3	(-20.500,20.000,0.000)	26865	1.26
4	(-20.000,20.500,0.000)	26865	1.26
5	(20.000,20.500,0.000)	26865	1.26
6	(20.500,20.000,0.000)	26865	1.26
7	(20.500,-20.000,0.000)	26865	1.26
8	(20.000,-20.500,0.000)	26865	1.26
9	(-20.000,-20.500,0.000)	26865	1.26
10	(-20.500,-20.000,0.000)	26865	1.26

11	(-20.500,-17.708,0.000)	2773	0.0
----	-------------------------	------	-----

表中最大频率指的是控制脉冲的最大频率，本题可以不予考虑，对应的速度指的是刀具的运动速度，单位是 m/min

运动过程中，机床对于速度、加速度、加加速度等的限制条件如下：

- 进给速度 V 范围 $[V_{\min}, V_{\max}]$: $[0.13, 6]$ 单位 m/min
- 加速度 a 范围 $[A_{\min}, A_{\max}]$: $[0.02, 0.6]$ 单位 m/s^2
- 加加速度 J_{const} : 300 单位 mm/s^3
- 瞬时启动速度 V_0 : $V_0 = 0.13$ 单位 m/min
- 瞬时启动加速度 a_0 : $a_0 = 0.02$ 单位 m/s^2
- 误差 $\varepsilon = 1\mu\text{m}$
- 分辨率: $\frac{1}{1280}\text{mm}$
- 插补周期: 1ms

本加工示例的直线段均平行于 x 轴或 y 轴，因此具有特殊性，即在直线段上的插补点都落在理论轨迹上，因此不存在误差。圆弧段的插补轨迹如图 5.6 所示，图中蓝色实线为目标加工轨迹，红色虚线为实际插补轨迹：

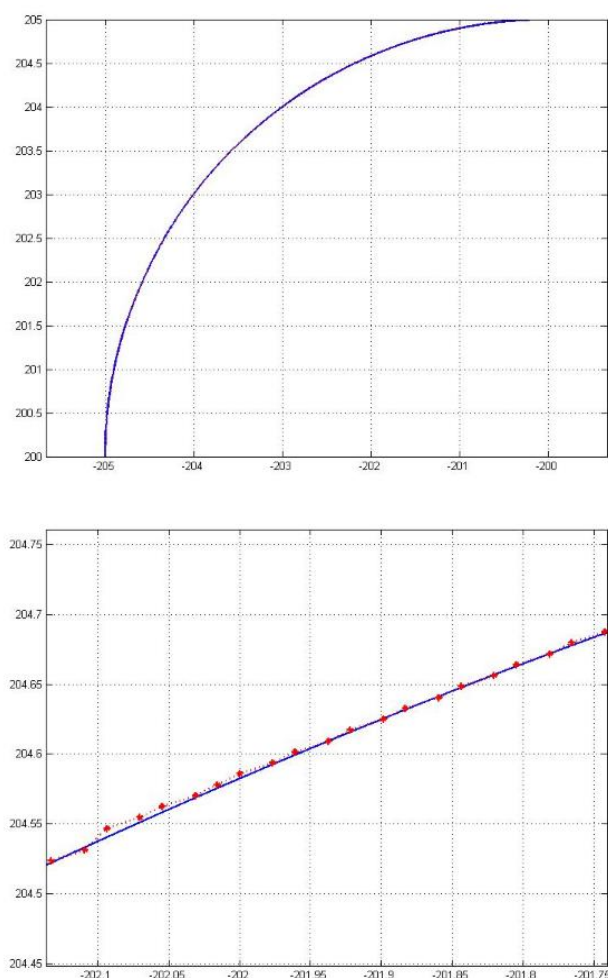


图 5.6 加工示例仿真结果

加工全程的速度曲线如图 5.7 所示：

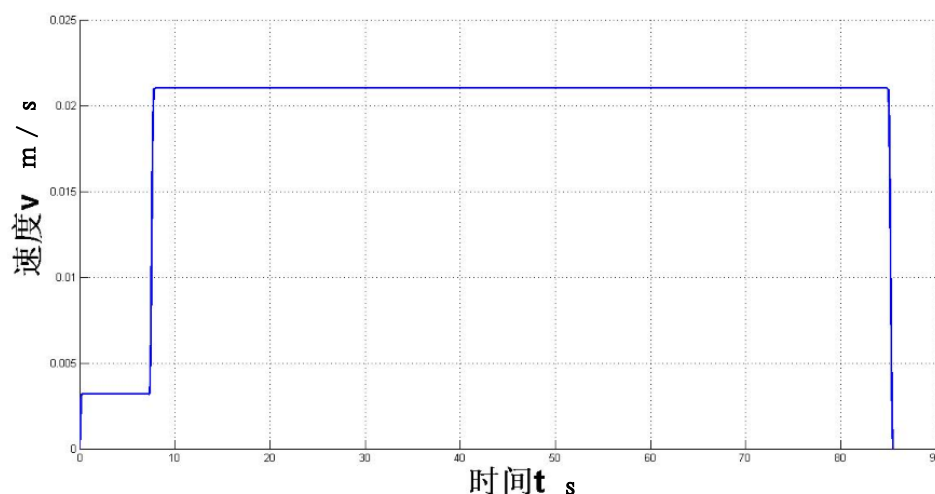


图 5.7 加工全程速度曲线

通过仿真可知，圆弧段的速度限制条件大于用户设定的进给速度，因此在加工的过程中，除了启动段满足 S 型加减速曲线的两段加速段（加工段 1 和 2）以及停止前的减速段（加工段 11）外，加工段 3~10 均以匀速运行。

各加工段时间和总加工时间如表 5.2 中所示：

表 5.2 各加工段仿真时间

时 间/s	T ₁	T ₂	T ₃	T ₄	T ₅	T ₆	T ₇	T ₈	T ₉	T ₁₀	T _总
	7.3406	18.1632	0.3740	19.0476	0.3740	19.0476	0.3740	19.0476	0.3740	1.3560	85.4987

仿真结果显示，加工示例的加工总时间为 85.4987s。

6 瞬时启动加速度及瞬时启动速度

6.1 瞬时启动加速度及瞬时启动速度对速度规划的影响

目前采用的 S 型速度控制曲线中，加速度每次都是从 0 开始增加，最后又降为 0，而实际运动过程中电机启动时允许有一个瞬时启动加速度 a_0 ，即认为加速度可以从 0 瞬时提高到瞬时加速度 a_0 ，或瞬间从 a_0 下降到 0，速度也有类似功能。

考虑瞬时启动加速度及瞬时启动速度对加工过程的影响主要体现在对速度规划的影响上。本文对同一条直线段分别作了有瞬时启动加速度及瞬时启动速度和没有瞬时启动加速度及瞬时启动速度的速度规划，并用 matlab 绘制了其速度与加速度的曲线图用于对比，如图 6.1 所示。

该图中假设起始速度和结束速度一样，因此规划出的速度曲线为对称曲线，加速度曲线为反对称曲线。图中红色曲线代表没有考虑瞬时启动加速度及瞬时启动速度的曲线， $t_1 \sim t_7$ 分别为其七个阶段的时间；蓝色曲线代表考虑了瞬时启动加速度及瞬时启动速度的曲线， $T_1 \sim T_7$ 分别为其七个阶段的时间。

通过对比和分析，主要有以下几点区别：

- 1) 由于加加速度 J 为固定值，当瞬时启动加速度 a_0 存在时，加速度不是从 0 开始增加，因此加速度增加到最大值的时间缩短，也就是加加速度段时间减少。在图中表现为 $T_1 < t_1$ 。
- 2) 由于瞬时启动速度 v_0 的存在，速度不是从 0 开始增加，且加加速度时间段减少，也就是说存在瞬时启动加速度使得速度规划提前进入加速度为最大值的阶段，因此速度达到最大值的时间应该缩短。在图中表现为 $T_1 + T_2 + T_3 < t_1 + t_2 + t_3$ 。
- 3) 由于考虑瞬时启动加速度及瞬时启动速度能够较早进入匀速阶段，即较早以速度的最大值运行。又考虑到减速的几个阶段和加速阶段对称，相比于没有考虑瞬时启动加速度及瞬时启动速度的规划也用时更短，所以其匀速运动的时间更长。体现在图中为 $T_4 > t_4$ 。
- 4) 由于考虑瞬时启动加速度及瞬时启动速度的速度规划运行在最大速度的时间更长，因此走完整条直线的用时应该更短。这体现在图中为：

$$\sum_{i=1}^7 T_i < \sum_{i=1}^7 t_i。$$

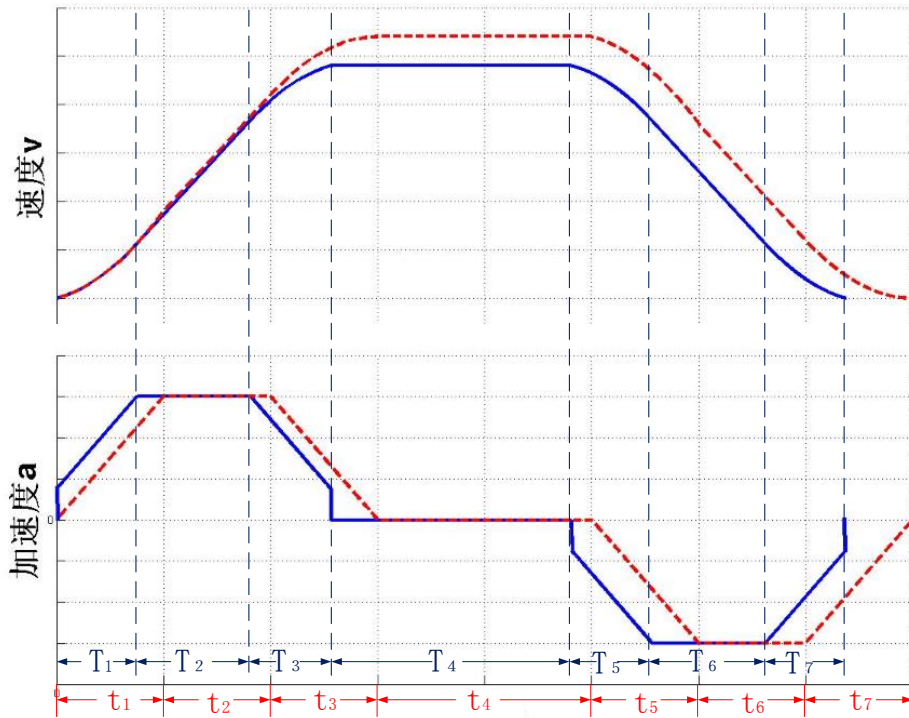


图 6.1 两种速度规划对比图

通过上述讨论，我们可以得出结论，考虑瞬时启动加速度及瞬时启动速度的加工方案效率更高，用时更短。

6.2 加工示例仿真验证

在考虑了瞬时启动加速度及瞬时启动速度的情况下对 5.5 中所述的加工示例进行仿真，切割路径加工中的速度要求调整为表 6.1 中所示，其余参数指标仍沿用 5.5 节中的指标。

表 6.1 圆角矩形切割路径加工中速度要求

路径节点	节点坐标(X、Y、Z)	用户设定的最大频率 F	最大频率 F 转换后对应的速度
1	(-20.500,-20.000,0.000)	2773	0.13
2	(-20.500,-17.708,0.000)	4029	0.19
3	(-20.500,20.000,0.000)	26865	1.26
4	(-20.000,20.500,0.000)	26865	1.26
5	(20.000,20.500,0.000)	26865	1.26
6	(20.500,20.000,0.000)	26865	1.26
7	(20.500,-20.000,0.000)	26865	1.26
8	(20.000,-20.500,0.000)	26865	1.26
9	(-20.000,-20.500,0.000)	26865	1.26
10	(-20.500,-20.000,0.000)	26865	1.26
11	(-20.500,-17.708,0.000)	2773	0.13

考虑了瞬时启动加速度及瞬时启动速度的速度曲线如图 6.2 所示：

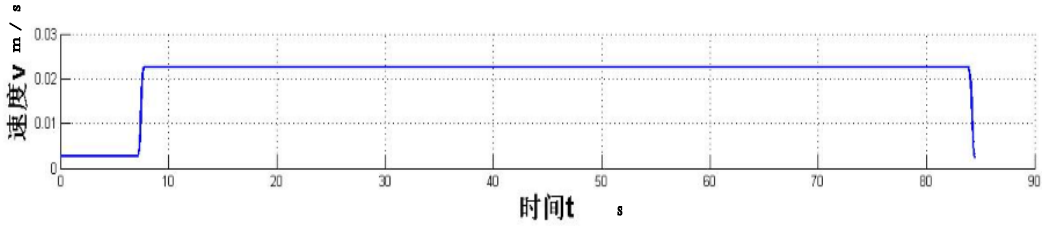


图 6.2 加工全程速度曲线

各加工段的加工时间如表 6.2 所示：

表 6.2 各加工段仿真时间

时间/s	T ₁	T ₂	T ₃	T ₄	T ₅	T ₆	T ₇	T ₈	T ₉	T ₁₀	T _总
	7.2501	18.1183	0.3740	19.0476	0.3740	19.0476	0.3740	19.0476	0.3740	1.3363	85.3435

仿真结果显示，加工示例的加工总时间为 85.3435 s。

考虑了瞬时启动加速度及瞬时启动速度的加工时间小于没有考虑瞬时启动加速度及瞬时启动速度的加工时间，其加工效率更优。

本题中两者的差距很小的原因是本题提供的最大加速度值较大，用户设定的最大进给速度较小，加速段和减速段的时间相对较少，匀速段的时间较长，因此两者的最终加工时间很接近。

综上所述，考虑瞬时启动加速度及瞬时启动速度有利于提高加工效率。

7 S 型曲线加减速控制方法分析

加减速控制一直是数控系统的关键技术之一，也是实现数控系统高实时性的瓶颈。在数控机床实际使用中，为了保证在启动或停止时不产生冲击、失步、超程或振荡，必须对各坐标轴上的速度进行加减速控制。良好的加减速控制算法，除了应保证数控机床运动平稳，在启停和程序段间速度有变化时不产生失步、超程、冲击、振荡外，还应具有算法简单、系统加减速处理时间短、实时性强的特点。

目前常用的加减速控制方法主要有以下几个：直线加减速算法、指数加减速算法、S 型曲线加减速算法。此处综合前三个问题使用 S 型曲线加减速控制方法的体会，通过与其他控制方法的横向比较，对 S 型曲线加减速控制方法优缺点进行分析。相对于直线加减速算法与指数加减速算法，S 型曲线算法的优点在于引入加加速度，通过对加加速度的控制来进行加速度连续的加减速变化，保证了速度启动、停止或进给速度在变迁过程中不会出现速度突变，能够平稳过渡，实现了加减速的柔性控制，使其能够最大限度地减小对机械系统造成的冲击，增强了机床运行的平稳性，适合用于高速进给数控系统。

虽然速度的连续变化有了较大改善，但是 S 型曲线算法的缺点也由此较为明显的体现。S 型曲线算法将变速过程分为 7 个阶段，导致涉及参数较多，且因路径长度的不同而有多种速度规划情况，实现起来较为复杂，占用机时较长，实时性有待提高。此外加加速度的不连续变化也对速度平滑性的程度造成一定阻碍。

8 平稳性优化控制运动模型

在 S 型曲线加减速算法中，加加速度的引用实现了加速度的加减速变化，从而保证了速度的平稳性，将速度变迁过程分为 7 个段，但是加加速度则非连续。文献[5]在 S 型曲线加减速的基础上，利用正余弦曲线的特性，拟合其加速度的过程，使其加加速度连续，进而实现速度的平稳光滑过渡，从而衍生出三角函数加减速算法。但其在整个加减速过程中没有充分发挥机床的性能，没有有效利用最大允许加速度，而是为保证线型对其进行限制，造成速度无法在较短时间、距离内达到期望值，导致与 S 型加减速曲线相比耗费时间较长。基于此，特提出改进型三角函数加减速算法对控制运动模型进行优化。

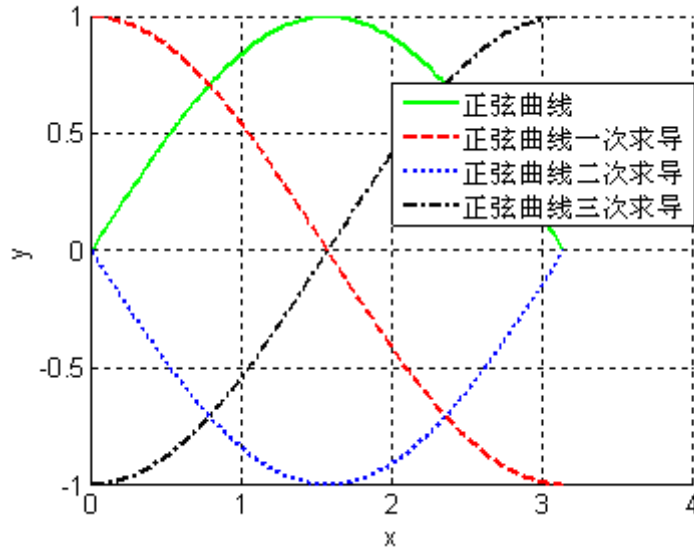


图 8.1 正弦曲线及其三次求导曲线

由图 8.1 可知，在 $0 \sim \pi$ 之间，正弦曲线对应的一次、二次、三次导数均是连续可导。这样光滑连续的曲线对于加减速运动控制而言是非常希望获取的，这样就不仅仅实现速度、加速度的连续，更实现加速、加加速度的导数均为连续的平稳光滑过度，使之可以保证加加速度也是连续平稳光滑过度，中间过程不存在任何冲击。保证加工的柔性，减少机床因为频繁的启停而造成机床的损害和精度损失。

依据正弦曲线及其导数的特性，进行推导可知，在进行加速运动过程中，其相关公式如下所示：

$$j(t) = \frac{\pi}{t_{am}} a_{rmax} \cos \frac{t\pi}{t_{am}} \quad (8.1)$$

$$a(t) = a_{rmax} \sin \frac{t\pi}{t_{am}} \quad (8.2)$$

$$v(t) = v_s + a_{rmax} \frac{t_{am}}{\pi} - a_{rmax} \frac{t_{am}}{\pi} \cos \frac{t\pi}{t_{am}} \quad (8.3)$$

$$s(t) = v_s t + \frac{a_{rmax} t_{am}}{\pi} t - \frac{a_{rmax} t_{am}^2}{\pi^2} \sin \frac{t\pi}{t_{am}} \quad (8.4)$$

其中 $t \in [0, t_{am}]$ 。

式中， t_{am} 表示整个加速运行时间， a_{rmax} 表示实际采用的最大加速度 v_s 表示运动过程中的起点速度。另设 j_{max} 表示最大加加速度，其与机床动力学、切削载荷等均有关系，是有限制值的，一般取定之后不再变化。 v 表示终点速度。

其具体约束关系如下：

$$t_{am} = \frac{\pi a_{max}}{j_{max}} \quad (8.5)$$

其中 $a_{rmax} \in [0, a_{max}]$ 。

同时将 $t = t_m$ 带入公式 (8.5)，得到：

$$v = v_s + a_{rmax} \frac{t_{am}}{\pi} + a_{rmax} \frac{t_{am}}{\pi} \quad (8.6)$$

从而推出：

$$v - v_s = 2a_{rmax} \frac{t_{am}}{\pi} \quad (8.7)$$

又因为公式 (8.5)，故此：

$$a_{rmax} = \sqrt{\frac{(v-v_s)j_{max}}{2}} \leq a_{max} \quad (8.8)$$

$$t_{am} = \pi \sqrt{\frac{(v-v_s)}{2j_{max}}} \quad (8.9)$$

而采用通用 S 型曲线加速方法，运行时间为： $t_{as} = 2\sqrt{\frac{v-v_s}{j_{max}}}$ 。由此可知两种方法耗费时间基本相等。在不明显延长运行时间的情况下，却可以实现加加速度的连续可导。

由上式可知，当 $v - v_s \leq \frac{2a_{max}^2}{j_{max}}$ 时，才可以采用上述的加速方式进行加速运算，故此，当终点速度与初始速度之差大于上述值时，则不能够采用这样的正余弦加速方式。为使公式具有更广的适用性，现在做进一步的改进：

加加速度：

$$j(t) = \begin{cases} \frac{\pi}{t_{am}} a_{max} \cos \frac{t\pi}{t_{am}} & t \in [0, \frac{t_{am}}{2}] \\ 0 & t \in [0, \frac{v-v_s}{a_{max}} - \frac{2t_{am}}{\pi}] \\ \frac{\pi}{t_{am}} a_{max} \cos \frac{t\pi}{t_{am}} & \frac{t_{am}}{2} \leq t \leq t_{am} \end{cases} \quad (8.10)$$

即在其加速过程中，当加速度达到最大时，保持一段时间，而不是立即开始减速，从而使运动具有较大的加速度，减少加速运行时间，提高效率。

加速度：

$$a(t) = \begin{cases} a_{max} \sin \frac{t\pi}{t_{am}} & t \in [0, \frac{t_{am}}{2}] \\ a_{max} & t \in [0, \frac{v-v_s}{a_{max}} - \frac{2t_{am}}{\pi}] \\ a_{max} \sin \frac{t\pi}{t_{am}} & t \in [\frac{t_{am}}{2}, t_{am}] \end{cases} \quad (8.11)$$

速度：

$$v(t) = \begin{cases} v_s - \frac{a_{max}t_{am}}{\pi} \cos \frac{t\pi}{t_{am}} & t \in [0, \frac{t_{am}}{2}] \\ v_s + \frac{a_{max}t_{am}}{\pi} + a_{max}t & t \in [0, \frac{v-v_s}{a_{max}} - \frac{2t_{am}}{\pi}] \\ v - \frac{a_{max}t_{am}}{\pi} - \frac{a_{max}t_{am}}{\pi} \cos \frac{t\pi}{t_{am}} & t \in [\frac{t_{am}}{2}, t_{am}] \end{cases} \quad (8.12)$$

位移：

$$s(t) = \begin{cases} v_s t + \frac{a_{max}t_{am}}{\pi} t - \frac{a_{max}t_{am}^2}{\pi^2} \sin \frac{t\pi}{t_{am}} & t \in [0, \frac{t_{am}}{2}] \\ v_s t + \frac{a_{max}t_{am}}{\pi} t + \frac{1}{2} a_{max} t^2 & t \in [0, \frac{v-v_s}{a_{max}} - \frac{2t_{am}}{\pi}] \\ v \left(t - \frac{t_{am}}{2} \right) - \frac{a_{max}t_{am}}{\pi} \left(t - \frac{t_{am}}{2} \right) + \frac{a_{max}t_{am}^2}{\pi^2} - \frac{a_{max}t_{am}^2}{\pi^2} \sin \frac{t\pi}{t_{am}} & t \in [\frac{t_{am}}{2}, t_{am}] \end{cases} \quad (8.13)$$

总共的运行时间： $t_{a-total} = t_m + \frac{v-v_s}{a} - \frac{2t_m}{\pi}$

将相关公式带入计算为：

$$t_{a-total} = \frac{v-v_s}{a_{max}} + (\pi - 2) \frac{a_{max}}{j_{max}} \quad (8.14)$$

而通用 S 型加速总运行时间： $t_{sa-total} = \frac{v-v_s}{a_{max}} + \frac{a_{max}}{j_{max}}$

则由公式 (8.14) 可知文章提出的加速方法，其总运行时间与通用的 S 型曲线总运行时间相比，可以说基本没有差异。而在运行中不含有匀速加速的过程中时，即： $v - v_s \leq \frac{2a_{max}^2}{j_{max}}$ 时，运行效果更加明显，只需要改变实际采用的最大加速度数值，即可实现加加速度的连续可导。而一般采用的 S 型则只能实现加速度的连续可导。故在机床频繁加减速时，更具有优势，减少对机床自身造成的冲击，保证机床安全与轨迹精度，运行更加平稳。

其减速的运算过程为上述过程的反过程。下式均为数据型，不含方向矢量，具体公式如下：

加加速度：

$$j(t) = \begin{cases} -\frac{\pi}{t_{dm}} a_{max} \cos \frac{t\pi}{t_{dm}} & t \in [0, \frac{t_{dm}}{2}] \\ 0 & t \in [0, \frac{v-v_e}{a_{max}} - \frac{2t_{dm}}{\pi}] \\ -\frac{\pi}{t_{dm}} a_{max} \cos \frac{t\pi}{t_{dm}} & \frac{t_{dm}}{2} \leq t \leq t_{dm} \end{cases} \quad (8.15)$$

加速度：

$$a(t) = \begin{cases} -a_{max} \sin \frac{t\pi}{t_{dm}} & t \in [0, \frac{t_{dm}}{2}] \\ -a_{max} & t \in [0, \frac{v-v_e}{a_{max}} - \frac{2t_{dm}}{\pi}] \\ -a_{max} \sin \frac{t\pi}{t_{dm}} & t \in [\frac{t_{dm}}{2}, t_{dm}] \end{cases} \quad (8.16)$$

速度：

$$v(t) = \begin{cases} v + \frac{a_{max}t_{dm}}{\pi} \cos \frac{t\pi}{t_{dm}} - \frac{a_{max}t_{dm}}{\pi} & t \in [0, \frac{t_{dm}}{2}] \\ v - \frac{a_{max}t_{dm}}{\pi} - a_{max}t & t \in [0, \frac{v-v_e}{a_{max}} - \frac{2t_{dm}}{\pi}] \\ v_e + \frac{a_{max}t_{dm}}{\pi} + \frac{a_{max}t_{dm}}{\pi} \cos \frac{t\pi}{t_{dm}} & t \in [\frac{t_{dm}}{2}, t_{dm}] \end{cases} \quad (8.17)$$

位移：

$$s(t) = \begin{cases} vt - \frac{a_{max}t_{dm}}{\pi} t + \frac{a_{max}t_{dm}^2}{\pi^2} \sin \frac{t\pi}{t_{dm}} & t \in [0, \frac{t_{dm}}{2}] \\ vt - \frac{a_{max}t_{dm}}{\pi} t - \frac{1}{2} a_{max} t^2 & t \in [0, \frac{v-v_s}{a_{max}} - \frac{2t_{dm}}{\pi}] \\ v_e \left(t - \frac{t_{dm}}{2}\right) + \frac{a_{max}t_{dm}}{\pi} \left(t - \frac{t_{dm}}{2}\right) - \frac{a_{max}t_{dm}^2}{\pi^2} + \frac{a_{max}t_{dm}^2}{\pi^2} \sin \frac{t\pi}{t_{dm}} & t \in [\frac{t_{dm}}{2}, t_{dm}] \end{cases} \quad (8.18)$$

式中， t_{dm} 表示整个减速运行时间， j_{max} 表示最大加加速度，， 其与机床动力学、切削载荷等均有关系，是有限制值的，一般取定之后不再变化。 v 、 v_e 表

示运动过程中的起点速度与终点速度。

其具体约束关系如下：

$$t_{dm} = \frac{\pi a_{max}}{j_{max}} \quad (8.19)$$

当其不含匀减速部分，即 $v - v_e \leq \frac{2a_{max}^2}{j_{max}}$ 时，减速算法可以进一步简化为：

$$j(t) = \frac{\pi}{t_{dm}} a_{dmax} \cos \frac{t\pi}{t_{dm}} \quad (8.20)$$

$$a(t) = -a_{dmax} \sin \frac{t\pi}{t_{dm}} \quad (8.21)$$

$$v(t) = v - a_{dmax} \frac{t_{dm}}{\pi} + a_{dmax} \frac{t_{dm}}{\pi} \cos \frac{t\pi}{t_{dm}} \quad (8.22)$$

$$s(t) = vt - \frac{a_{dmax} t_{dm}}{\pi} t + \frac{a_{dmax} t_{dm}^2}{\pi^2} \sin \frac{t\pi}{t_{dm}} \quad (8.23)$$

其中 $t \in [0, t_{dm}]$ 。

a_{dmax} 表示实际应用的最大减速度。由上述减速相关公式可以得出：

$$a_{dmax} = \sqrt{\frac{(v-v_e)j_{max}}{2}} \leq a_{max} \quad (8.24)$$

$$t_{dm} = \frac{\pi a_{dmax}}{j_{max}} = \pi \sqrt{\frac{v-v_e}{2j_{max}}} \quad (8.25)$$

9 模型评价

9.1 模型的优点

- (1) 问题一提出折线转接点速度不为 0 的过渡算法，最大程度对加工效率进行优化；转接速度限制充分考虑加速度、加工误差等条件，确保转接速度的最大值；速度规划时将所有情况的判断方法进行列举；在讨论折线交点处坐标速度变化规律时，补充折线夹角对合成速度的影响。
- (2) 问题二提出圆弧转接点速度不为 0 的过渡算法，最大程度对加工效率进行优化；转接速度限制充分考虑加速度限制、圆弧弓高误差限制和多步转接条件限制，确保准接速度的最大值。
- (3) 针对问题二与问题三提出的算法，加工示例的仿真结果证明实际值与理论值误差较小，时间响应较快。
- (4) 问题四中，针对三角函数加减速算法对最大允许加速度利用不高的情况，进行了改进优化。

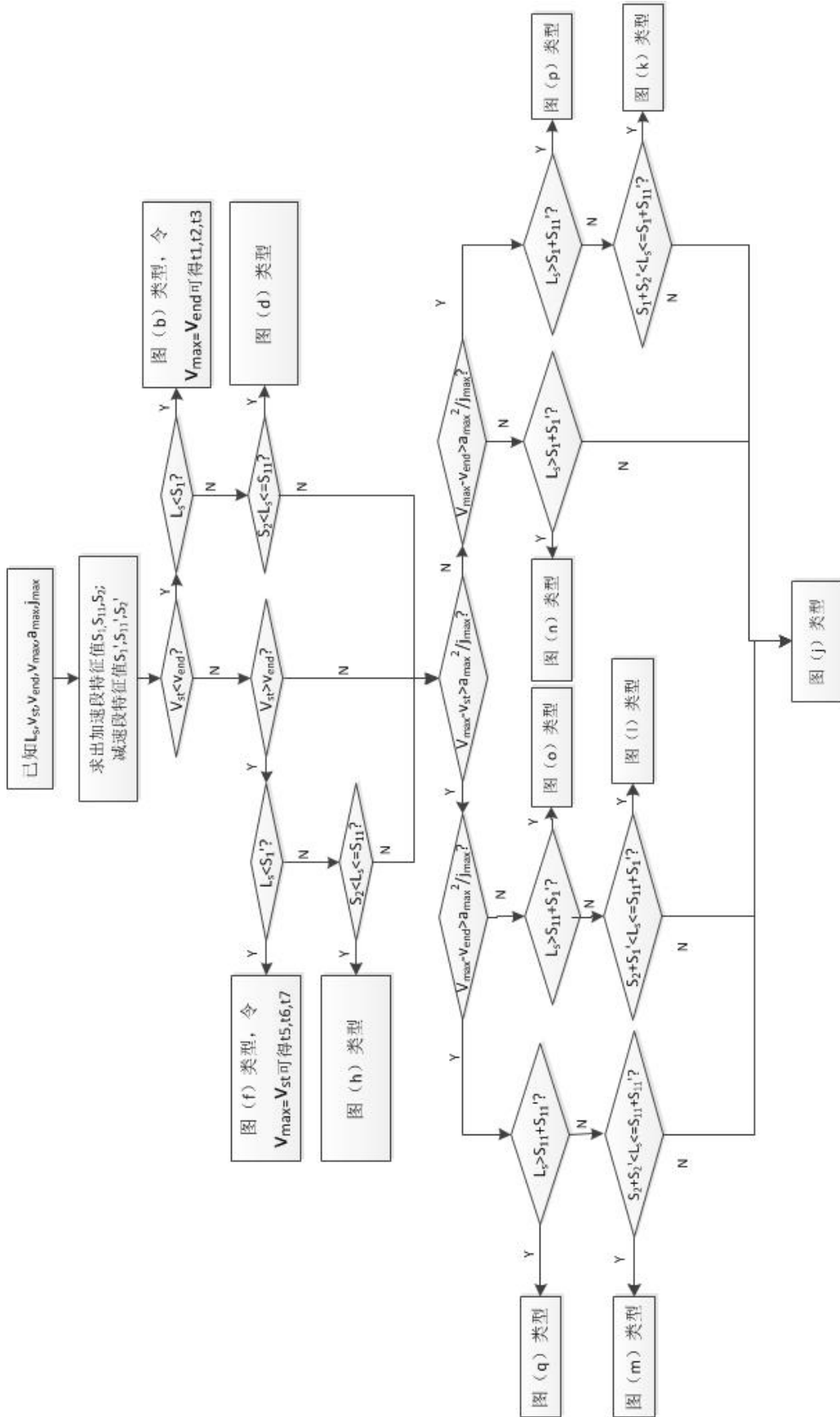
9.2 模型的缺点

加减速控制算法涉及参数较多，且因路径长度的不同而有多种速度规划情况，实现起来较为复杂。

参考文献

- [1] 吕强, 张辉, 杨开明等. 数控连续加工中提高轨迹段转接速度的方法研究[J]. 制造技术与机床, 2008:79-83. DOI:doi:10.3969/j.issn.1005-2402.2008.07.029.
- [2] 杨开明, 石川, 叶佩青等. 数控系统轨迹段光滑转接控制算法[J]. 清华大学学报: 自然科学版, 2007, 47:1295-1299. DOI:doi:10.3321/j.issn:1000-0054.2007.08.012.
- [3] 刘清建. 数控系统运动控制及运行优化[D]. 天津大学, 2010. DOI:doi:10.7666/d.y2081895.
- [4] 潘海鸿, 杨微, 陈琳,等. 全程S曲线加减速控制的自适应分段NURBS曲线插补算法[J]. 中国机械工程, 2010:190-195.
- [5] 胡磊, 林示麟, 徐建明等.S 曲线加减速速度控制新方法[J]. 组合机床与自动化加工技术, 2010:22-26. DOI:doi:10.3969/j.issn.1001-2265.2010.01.006.

附录 I



附录 II

Matlab 编程文件:

一、圆弧段插补 M 文件

```
function [] = yh(T,v,R)
T=0.001;
f=10/1280;
v=21;
R=5;
x0=-200;
y0=200;
beta=pi;
alpha=2*asin(0.5*v*T/R);
N=ceil(0.5*pi/alpha);
alpha1=beta:-alpha:beta-N*alpha;
x=x0+R*cos(alpha1);
y=y0+R*sin(alpha1);
x_(1)=x(1);
y_(1)=y(1);
for i=1:N
    dx(i)=x(i+1)-x_(i);
    dy(i)=y(i+1)-y_(i);
    nx(i)=ceil(dx(i)/f);
    ny(i)=ceil(dy(i)/f);
    x_(i+1)=x_(i)+nx(i)*f;
    y_(i+1)=y_(i)+ny(i)*f;
i=i+1;
end
alpha1=beta:-alpha/100:beta-N*alpha;
x=x0+R*cos(alpha1);
y=y0+R*sin(alpha1);
plot(x,y,'-b','LineWidth',2);
hold on;
plot(x_,y_,':*r','LineWidth',2);
axis('equal');
grid on;
end
```

二、不考虑瞬时启动加速度及瞬时启动速度的加工示例速度规划 M 文件

```
function [] = l2(t,v,s,T1)
j=0.3;
v0=0;
a0=0;
a=0.6;
```

```

s0=0;
t01=(0.19/60/j)^0.5
a=(0.19/60*j)^0.5
t02=0;
t03=t01
t04=(22.92*0.001-(-
j*t03.^3/6+0.5*a*t03^2+(a*t02+0.5*j*t01^2+a0*t01+v0)*t03+0.5*a*t02^2
+(0.5*j*t01^2+a0*t01+v0)*t02+j*t01^3/6+0.5*a0*t01^2+v0*t01+s0))/(0.1
9/60)
i=1;
for t=0:0.001:t01+t03+t04
if t<=t01
v(i)=0.5*j*t.^2+a0*t+v0;
elseif t<=t01+t03
v(i)=-0.5*j*(t-t01).^2+a*(t-t01)+(a*t02+0.5*j*t01^2+a0*t01+v0);
else
v(i)=-0.5*j*t03^2+a*t03+(a*t02+0.5*j*t01^2+a0*t01+v0);
end
i=i+1;
end

v0=0.19/60;
a0=0;
a=0.6;
s0=0;
t2=0;
t1=((1.26-0.19)/60/j)^0.5
a=((1.26-0.19)/60*j)^0.5
t3=t1;
t4=(377.08*0.001-(-
j*t3.^3/6+0.5*a*t3^2+(a*t2+0.5*j*t1^2+a0*t1+v0)*t3+0.5*a*t2^2+(0.5*j
*t1^2+a0*t1+v0)*t2+j*t1^3/6+0.5*a0*t1^2+v0*t1+s0))/1.26*60+4*(0.5*pi
*0.5*0.01/1.26*60)+1.2/1.26*60;
for t=t01+t03+t04+0.001:0.001:t01+t03+t04+t1+t2+t3+t4
if t<=t01+t03+t04+t1
v(i)=0.5*j*(t-(t01+t03+t04)).^2+a0*(t-(t01+t03+t04))+v0;
elseif t<=t01+t03+t04+t1+t2+t3
v(i)=-0.5*j*(t-t1-t2-(t01+t03+t04)).^2+a*(t-t1-t2-
(t01+t03+t04))+(a*t2+0.5*j*t1^2+a0*t1+v0);
else
v(i)=-0.5*j*t3^2+a*t3+(a*t2+0.5*j*t1^2+a0*t1+v0);
end
i=i+1;

```

```

end

v0=1.26/60;
a0=0;
a=0.6;
s0=0;
t26=0;
t25=((1.26-0)/60/j)^0.5
a=((1.26-0)/60*j)^0.5
t27=t25;
t24=((22.92*0.001-(-j*t25.^3/6+v0*t25-0.5*a0*t25^2+j*t27^3/6-
0.5*a*t27^2+(-0.5*j*t25^2+v0-a0*t25)*t25))/1.26*60
for
t=(t01+t03+t04)+t1+t2+t3+t4+0.001:0.001:(t01+t03+t04)+t1+t2+t3+t4+t2
4+t25+t27
if t<=t01+t03+t04+t1+t2+t3+t4+t24
    v(i)=1.26/60;
elseif t<=t01+t03+t04+t1+t2+t3+t4+t24+t25
v(i)=-0.5*j*(t-(t01+t03+t04+t1+t2+t3+t4+t24)).^2-a*(t-
(t01+t03+t04+t1+t2+t3+t4+t24))+v0;
else
    v(i)=0.5*j*(t-(t01+t03+t04+t1+t2+t3+t4+t24+t25)).^2-a*(t-
(t01+t03+t04+t1+t2+t3+t4+t24+t25))+(-0.5*j*t25^2+v0-a0*t25);
end
i=i+1;
end
T1=t01+t03+t04
T2=t1+t2+t3+t4-(4*(0.5*pi*0.5*0.01/1.26*60)+1.2/1.26*60)
T3=0.5*pi*0.5*0.01/1.26*60
T4=0.4/1.26*60
T5=0.5*pi*0.5*0.01/1.26*60
T6=0.4/1.26*60
T7=0.5*pi*0.5*0.01/1.26*60
T8=0.4/1.26*60
T9=0.5*pi*0.5*0.01/1.26*60
T10=t24+t25+t27
T=t01+t03+t04+t1+t2+t3+t4+t24+t25+t27
t =0:0.001:t01+t03+t04+t1+t2+t3+t4+t24+t25+t27;
plot (t,v,'-b','LineWidth',1.5);
%set(gca,'xtick',[],'ytick',[]);
xlabel ('时间t','fontsize',30,'fontweight','bold');
ylabel ('速度v','fontsize',30,'fontweight','bold');
box off;
grid on;

```

```

    %hold on;
end

```

三、考虑瞬时启动加速度及瞬时启动速度的加工示例速度规划 M 文件

```

function [] = l3(t,v)
j=0.3;
v0=0.13/60;
a0=0.015;
a=0.6;
s0=0;
t01=(0.06/60/j)^0.5-a0/j;
a=(0.06/60*j)^0.5;
t02=0;
t03=t01+a0/j;
t04=(22.92*0.001-(-
j*t03.^3/6+0.5*a*t03^2+(a*t02+0.5*j*t01^2+a0*t01+v0)*t03+0.5*a*t02^2
+(0.5*j*t01^2+a0*t01+v0)*t02+j*t01^3/6+0.5*a0*t01^2+v0*t01+s0))/(0.1
9/60);
i=1;
for t=0:0.001:t01+t03+t04
if t<=t01
v(i)=0.5*j*t.^2+a0*t+v0;
elseif t<=t01+t03
v(i)=-0.5*j*(t-t01).^2+a*(t-t01)+(a*t02+0.5*j*t01^2+a0*t01+v0);
else
v(i)=-0.5*j*t03.^2+a*t03+(a*t02+0.5*j*t01^2+a0*t01+v0);
end
v(i)
i=i+1;
end

v0=0.19/60;
a0=0.015;
a=0.6;
s0=0;
t2=0;
t1=((1.26-0.19)/60/j)^0.5-0.5*a0/j;
a=((1.26-0.19)/60*j)^0.5;
t3=t1;
t4=(377.08*0.001-(-
j*t3.^3/6+0.5*a*t3^2+(a*t2+0.5*j*t1^2+a0*t1+v0)*t3+0.5*a*t2^2+(0.5*j
*t1^2+a0*t1+v0)*t2+j*t1^3/6+0.5*a0*t1^2+v0*t1+s0))/1.26*60+4*(0.5*pi
*0.5*0.01/1.26*60)+1.2/1.26*60;
for t=t01+t03+t04+0.001:0.001:t01+t03+t04+t1+t2+t3+t4

```

```

if t<=t01+t03+t04+t1
v(i)=0.5*j*(t-(t01+t03+t04)).^2+a0*(t-(t01+t03+t04))+v0;
elseif t<=t01+t03+t04+t1+t2+t3
    v(i)=-0.5*j*(t-t1-t2-(t01+t03+t04)).^2+a*(t-t1-t2-
    (t01+t03+t04))+(a*t2+0.5*j*t1^2+a0*t1+v0);
else
    v(i)=-0.5*j*t3^2+a*t3+(a*t2+0.5*j*t1^2+a0*t1+v0);
end
i=i+1;
end

v0=1.26/60;
a0=0.015;
a=0.6;
s0=0;
t26=0;
t25=((1.26-0.13)/60/j)^0.5-a0/j;
a=((1.26-0.13)/60*j)^0.5;
t27=t25+a0/j;
t24=(22.92*0.001-(-j*t25.^3/6+v0*t25-0.5*a0*t25^2+j*t27^3/6-
0.5*a*t27^2+(-0.5*j*t25^2+v0-a0*t25)*t25))/1.26*60;
for
t=(t01+t03+t04)+t1+t2+t3+t4+0.001:0.001:(t01+t03+t04)+t1+t2+t3+t4+t2
4+t25+t27
if t<=t01+t03+t04+t1+t2+t3+t4+t24
    v(i)=v0;
elseif t<=t01+t03+t04+t1+t2+t3+t4+t24+t25
v(i)=-0.5*j*(t-(t01+t03+t04+t1+t2+t3+t4+t24)).^2-a0*(t-
(t01+t03+t04+t1+t2+t3+t4+t24))+v0;
else
    v(i)=0.5*j*(t-(t01+t03+t04+t1+t2+t3+t4+t24+t25)).^2-a*(t-
(t01+t03+t04+t1+t2+t3+t4+t24+t25))+(-0.5*j*t25^2+v0-a0*t25);
end
i=i+1;
end

T1=t01+t03+t04
T2=t1+t2+t3+t4-(4*(0.5*pi*0.5*0.01/1.26*60)+1.2/1.26*60)
T3=0.5*pi*0.5*0.01/1.26*60
T4=0.4/1.26*60
T5=0.5*pi*0.5*0.01/1.26*60
T6=0.4/1.26*60
T7=0.5*pi*0.5*0.01/1.26*60
T8=0.4/1.26*60

```

```

T9=0.5*pi*0.5*0.01/1.26*60
T10=t24+t25+t27
T=t01+t03+t04+t1+t2+t3+t4+t24+t25+t27

t =0:0.001:t01+t03+t04+t1+t2+t3+t4+t24+t25+t27;
    plot (t,v,'-b','LineWidth',1.5);
    %set(gca,'xtick',[],'ytick',[]);
    xlabel ('时间t','fontsize',20,'fontweight','bold');
    ylabel ('速度v','fontsize',20,'fontweight','bold');
    box off;
    grid on;
    %hold on;
end

```

四、加工示例节点1—节点2速度曲线

```

function [] = l1(t,v)
j=0.3;
v0=0;
a0=0;
a=0.6;
s0=0;
t01=(0.19/60/j)^0.5
a=(0.19/60*j)^0.5
t02=0;
t03=t01
t04=(22.92*0.001-(-
j*t03.^3/6+0.5*a*t03^2+(a*t02+0.5*j*t01^2+a0*t01+v0)*t03+0.5*a*t02^2
+(0.5*j*t01^2+a0*t01+v0)*t02+j*t01^3/6+0.5*a0*t01^2+v0*t01+s0))/(0.1
9/60)
i=1;
for t=0:0.001:t01+t03+t04
if t<=t01
    v(i)=0.5*j*t.^2+a0*t+v0;
elseif t<=t01+t03
    v(i)=-0.5*j*(t-t01).^2+a*(t-t01)+(a*t02+0.5*j*t01^2+a0*t01+v0);
else
    v(i)=-0.5*j*t03^2+a*t03+(a*t02+0.5*j*t01^2+a0*t01+v0);
end
i=i+1;
end
t =0:0.001:t01+t02+t03+t04;
    plot (t,v,'-b','LineWidth',1.5);
    %set(gca,'xtick',[],'ytick',[]);

```

```

xlabel ('时间t','fontsize',30,'fontweight','bold');
ylabel ('速度v','fontsize',30,'fontweight','bold');
box off;
%axis ([-1,t1+t2+t3+t4+t5+t6+t7+1,0,25]);
grid on;
%hold on;

end

```

五、S型曲线的速度、加速度、加加速度曲线走势

```

function [] = vaj(t,v,a,J)
j=0.3;
v0=0;
i=1;
for t=0:0.001:16
if t<=2
v(i)=0.5*j*t.^2+v0;
a(i)=j*t;
J(i)=j;
elseif t<=4
v(i)=0.6*(t-2)+0.5*j*2^2+v0;
a(i)=0.6;
J(i)=0;
elseif t<=6
v(i)=-0.5*j*(t-4).^2+0.6*(t-4)+0.6*2+0.5*j*2^2+v0;
a(i)=-j*t+1.8;
J(i)=-j;
elseif t<=10
v(i)=2.4;
a(i)=0;
J(i)=0;
elseif t<=12
v(i)=-j*(t-10).^2/2+2.4;
a(i)=-j*t+10*j;
J(i)=-j;
elseif t<=14
v(i)=-0.6*(t-12)-0.5*j*2^2+2.4;
a(i)=-0.6;
J(i)=0;
else
v(i)=0.5*j*(t-14).^2-0.6*(t-14)-0.6*2-
0.5*j*2^2+2.4;
a(i)=j*(t-16);

```

```

        J(i)=j;

end
i=i+1;
end
subplot (3,1,1);
t =0:0.001:16;
plot (t,v,'-b','LineWidth',3);
set(gca,'xtick',[]);
xlabel ('时间t','fontsize',24,'fontweight','bold');
ylabel ('速度v','fontsize',24,'fontweight','bold');
box off;
axis ([0,16,0,4]);
title ('图1 S型速度曲线','fontsize',30);
hold on;
subplot (3,1,2);
t =0:0.001:16;
plot (t,a,'-b','LineWidth',3);
set(gca,'xtick',[]);
xlabel ('时间t','fontsize',24,'fontweight','bold');
ylabel ('加速度a','fontsize',24,'fontweight','bold');
box off;
axis ([0,16,-1,1]);
title ('图2 加速度曲线','fontsize',30);
hold on;
subplot (3,1,3);

t =0:0.001:16;
plot (t,J,'-b','LineWidth',3);
set(gca,'xtick',[]);
xlabel ('时间t','fontsize',24,'fontweight','bold');
ylabel ('加加速度J','fontsize',24,'fontweight','bold');
box off;
axis ([0,16,-0.5,0.5]);
title ('图3 加加速度曲线','fontsize',30);
text (0.8,-0.3,'t1','fontsize',24);
text (2.8,-0.3,'t2','fontsize',24);
text (4.8,-0.3,'t3','fontsize',24);
text (7.8,-0.3,'t4','fontsize',24);
text (10.8,-0.3,'t5','fontsize',24);
text (12.8,-0.3,'t6','fontsize',24);
text (14.8,-0.3,'t7','fontsize',24);
end

```


六、S型曲线的位移曲线走向

```
function [] = s(t,v,s)
j=0.3;
v0=0;
i=1;
t1=2;
t2=2;
t3=2;
t4=4;
t5=2;
t6=2;
t7=2;
a0=0;
a=0.6;
s0=0;
for t=0:0.001:t1+t2+t3+t4+t5+t6+t7
if t<=t1
s(i)=j*t.^3/6+0.5*a0*t^2+v0*t+s0
elseif t<=t1+t2
s(i)=0.5*a*(t-t1)^2+(0.5*j*t1^2+a0*t1+v0)*(t-t1)+j*t1^3/6+0.5*a0*t1^2+v0*t1+s0;
elseif t<=t1+t2+t3
s(i)=-j*(t-t1-t2).^3/6+0.5*a*(t-t1-t2)^2+(a*t2+0.5*j*t1^2+a0*t1+v0)*(t-t1-t2)+0.5*a*t2^2+(0.5*j*t1^2+a0*t1+v0)*t2+j*t1^3/6+0.5*a0*t1^2+v0*t1+s0;
elseif t<=t1+t2+t3+t4
s(i)=(-0.5*j*t3^2+a*t3+(a*t2+0.5*j*t1^2+a0*t1+v0))*(t-t1-t2-t3)+(-j*t3.^3/6+0.5*a*t3^2+(a*t2+0.5*j*t1^2+a0*t1+v0)*t3+0.5*a*t2^2+(0.5*j*t1^2+a0*t1+v0)*t2+j*t1^3/6+0.5*a0*t1^2+v0*t1+s0);
elseif t<=t1+t2+t3+t4+t5
s(i)=-j*(t-t1-t2-t3-t4).^3/6-0.5*a0*(t-t1-t2-t3-t4).^2+(-0.5*j*t3^2+a*t3+(a*t2+0.5*j*t1^2+a0*t1+v0))*(t-t1-t2-t3-t4)+((-0.5*j*t3^2+a*t3+(a*t2+0.5*j*t1^2+a0*t1+v0))*t4+(-j*t3.^3/6+0.5*a*t3^2+(a*t2+0.5*j*t1^2+a0*t1+v0)*t3+0.5*a*t2^2+(0.5*j*t1^2+a0*t1+v0)*t2+j*t1^3/6+0.5*a0*t1^2+v0*t1+s0));
elseif t<=t1+t2+t3+t4+t5+t6
s(i)=-0.5*a*(t-(t1+t2+t3+t4+t5))^2+(-0.5*j*t5^2-a0*t5+(-0.5*j*t3^2+a*t3+(a*t2+0.5*j*t1^2+a0*t1+v0)))*(t-(t1+t2+t3+t4+t5))-j*t5^3/6-0.5*a0*t5^2+(-0.5*j*t3^2+a*t3+(a*t2+0.5*j*t1^2+a0*t1+v0))*t5+((-0.5*j*t3^2+a*t3+(a*t2+0.5*j*t1^2+a0*t1+v0))*t4+(-
```

```

j*t3.^3/6+0.5*a*t3^2+(a*t2+0.5*j*t1^2+a0*t1+v0)*t3+0.5*a*t2^2+(0.5*j
*t1^2+a0*t1+v0)*t2+j*t1^3/6+0.5*a0*t1^2+v0*t1+s0));
else
    s(i)=j*(t-(t1+t2+t3+t4+t5+t6)).^3/6-0.5*a*(t-
(t1+t2+t3+t4+t5+t6)).^2+(-a*t5+(-0.5*j*t5^2-a0*t5+(-
0.5*j*t3^2+a*t3+(a*t2+0.5*j*t1^2+a0*t1+v0))))*(t-
(t1+t2+t3+t4+t5+t6))-0.5*a*t6^2+(-0.5*j*t5^2-a0*t5+(-
0.5*j*t3^2+a*t3+(a*t2+0.5*j*t1^2+a0*t1+v0)))*t6-j*t5^3/6-
0.5*a0*t5^2+(-0.5*j*t3^2+a*t3+(a*t2+0.5*j*t1^2+a0*t1+v0))*t5+((-
0.5*j*t3^2+a*t3+(a*t2+0.5*j*t1^2+a0*t1+v0))*t4+(-
j*t3.^3/6+0.5*a*t3^2+(a*t2+0.5*j*t1^2+a0*t1+v0)*t3+0.5*a*t2^2+(0.5*j
*t1^2+a0*t1+v0)*t2+j*t1^3/6+0.5*a0*t1^2+v0*t1+s0));
end
i=i+1;
end
t =0:0.001:t1+t2+t3+t4+t5+t6+t7;
plot (t,s,'-b','LineWidth',3);
xlabel ('时间t','fontsize',30,'fontweight','bold');
ylabel ('位移s','fontsize',30,'fontweight','bold');
box off;
%axis ([-1,t1+t2+t3+t4+t5+t6+t7+1,0,25]);
grid on;
%hold on;
end

```