



数控加工中的二次曲线拟合与最优插补控制算法

张梅^①, 闫伟^②, 袁春明^{①*}, 王定康^①, 高小山^①

^① 中国科学院数学与系统科学研究院数学机械化重点实验室, 北京 100190

^② 中国石油勘探开发研究院, 北京 100083

* 通信作者. E-mail: cmyuan@mmrc.iss.ac.cn

收稿日期: 2009-12-30; 接受日期: 2010-09-16

中国科学院知识创新工程重要方向项目 (批准号: KG CX2-YW-119-1) 和基金委创新群体项目 (批准号: NSFC 608210021/F02) 资助

摘要 本文提出一种基于二次 B 样条曲线对 G01 代码的拟合及插补方法, 先通过自适应方法选取 G01 代码所描述的加工轨迹形状特征各个特征点, 再用通过所有特征点的二次 B 样条曲线拟合待加工路径. 由于 Timar 等人提出的最优速度规划算法对于三次及三次以上的高阶次样条曲线的计算极为复杂, 本文提出了一种改进的基于二次 B 样条曲线的时间最优插补计算方法. 最后, 将所提出的算法应用到实际的图案加工仿真与实际数控加工中.

关键词 二次 B 样条 速度规划 G01 代码 最优插补

1 引言

在高速高精度的数控加工中, 采用 G01 代码描述的加工路径往往具有数据量大、加工方向变化剧烈且频繁的特点. 直接对这种以 G01 代码描述的小线段进行插补, 存在加工速度低、加工表面的光洁度不够高等缺陷. Lv 等人^[1] 采用圆弧过渡的插补方式进行加工路径的光滑化. 但这些方法得到的加工路径光滑度不够高, 并且产生了两倍的 G 代码, 不能达到数据压缩的目的. Zhang 等人^[2] 采用拐角多周期过渡的方法给出小线段的过渡插补方法进行光滑化加工, 但这一方法同样不能达到数据压缩的目的. 解决上述问题的一种方案是将待加工的产品形状用样条曲线重新进行描述. 在此过程中, 用一段光滑曲线来逼近多个小线段, 称为数据压缩, 每段光滑曲线所逼近的小线段的个数被称为压缩比.

基于样条曲线直接插补是数控机床完成插补加工的趋势. 具体实施时, 对样条曲线和插补加工方式的选择非常关键. 在用样条曲线描述加工产品的形状时, 通常采用的办法是样条插值. Yau 等人^[3,4] 给出了一个基于小线段的 CSB (continuous short blocks) 判别法, 对小线段分段后再进行样条拟合与插补, 但这一方法要求拟合曲线通过所有数据点, 在处理稠密 G01 代码时, 会出现曲线段数过多, 且每段多项式曲线过短的问题. 因此, 应在拟合前选出能够描述曲线形状特征的点, 而后再进行插值计算^[5]. 本文将给出一种新的自适应特征点选取方法, 并针对特征点序列进行加工路径拟合. 文献 [6, 7] 中介绍了多种样条曲线拟合方法, 但是对三次及三次以上的多项式曲线和有理多项式曲线进行速度规划时, 其计算就变得非常复杂, 很难做到实时处理. 因此, 本文采用二次 B 样条曲线拟合待加工路径.

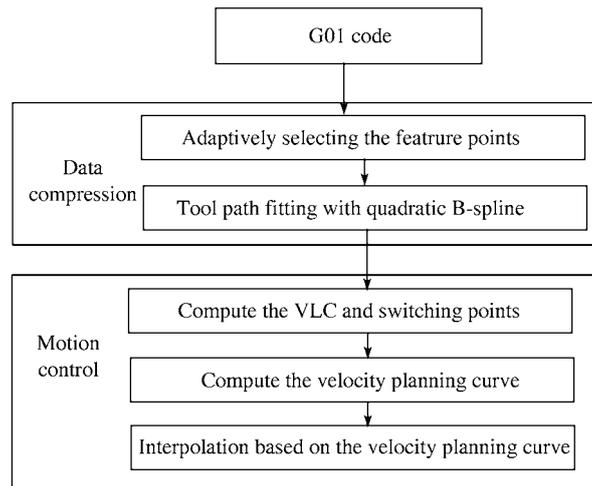


图 1 曲线拟合和插补流程图

Figure 1 Flow chart for curve fitting and interpolation

在曲线插补加工控制方面, Ye 等人^[8]采用了匀速过渡的加工模式. Altintas 等人^[9]给出了一个基于线加速度与线加加速度控制的速度模型. Yu 等人^[10,11]给出了基于平均滤波技术的新的 S 曲线加减速控制方法, 达到加速度和加加速度的连续控制. 在文献 [12–16] 中介绍了不同的速度控制方法, 但上述方法均采用切向 (线) 加速度、加加速度的控制方式, 而没有考虑各个驱动轴的加速能力, 不能充分利用各个驱动轴的最大加速度. 基于上述问题, 要想充分利用各加工轴的加工能力, 就需要进行分轴的速度规划^[17–21]. 在文献 [17] 中, Timar 等人提出一种基于各驱动轴最大加速能力的一般样条曲线的最优插补控制方法. 这种方法对于三次及三次以上的高阶样条曲线所执行的相应运动规划的计算极为复杂, 在现有的软件和硬件环境下无法做到实时计算.

如何克服现有技术的局限, 提出一种新的对 G01 代码的拟合及插补方法, 是数控技术中的重要问题. 有鉴于此, 本文提出了一种基于二次 B 样条曲线对 G01 代码的拟合及插补方法, 能够在一定程度上克服上述现有技术中的难点和缺陷, 具有计算速度快、加工精度高、工作性能稳定, 且适用范围广泛的特点. 该方法能在满足系统预设精度要求的前提下, 用二次 B 样条曲线描述待加工路径, 并实时完成样条曲线的插补计算, 能在整体上保证针对二次 B 样条曲线描述的待加工路径实现加工时间最优, 满足高速、高精度的数控加工的需要.

本文的结构安排如下, 第 2 节给出了曲线的拟合和速度规划流程; 第 3 节探讨了利用二次 B 样条曲线对 G01 代码进行拟合的方法; 第 4 节对于二次样条曲线, 提出了一种改进的基于 Bang-bang 控制思想的速度规划算法; 第 5 节针对 3 种实际的加工模型进行了仿真测试; 第 6 节总结全文.

2 基于二次样条曲线的拟合和速度规划总体思路

我们的加工流程主要分为数据压缩和运动控制两部分, 整体的流程如图 1 所示.

输入代码是由 CAM 对加工样件生成的 G01 代码, 它表示空间中一组离散、稠密、有序点集. 由于 G01 代码的数据量庞大, 不仅给数据的传输和存储带来很大的困难和负担, 并且, 直接对 G01 代码所确定的小折线段进行插补, 需要刀具进行频繁的加减速, 这样不仅严重影响了机床的加工效率, 由此

而产生的震动也会影响加工表面的光洁度, 使之很难达到高精加工的要求. 因此, 本文设计的算法将首先在给定的精度要求下, 将 G01 代码拟合成二次 B 样条曲线, 而后, 针对该样条曲线进行速度规划和插补.

对 G01 代码拟合的过程中, 本文令每段样条曲线在精度范围内逼近尽可能多的数据点, 以达到数据压缩的目的, 主要包括以下 3 个步骤:

1) 首先, 将 G01 代码进行分段: 由于 G01 代码数据量庞大, 在大多数情况下不能只用一条 B 样条曲线进行描述. 因此, 本文将参照机床的机械特性及模型的形状特征对数据点进行分段, 并在每两个分段点间用一条 B 样条曲线进行拟合.

2) 利用二次 Bézier 曲线自适应选取 G01 代码中能够描述待加工轨迹形状特征的特征点.

3) 最后, 利用二次 B 样条曲线插值所有特征点, 得到描述待加工路径的拟合曲线.

在得到描述样件加工路径的拟合曲线后, 针对这些样条曲线进行速度规划和实时插补, 从而得到时间最优的插补过程. 算法主要分为 3 个步骤:

1) 根据机床的机械特性 (主要是加工能力) 及描述加工路径的拟合曲线的性质计算最大允许加工速度曲线 (VLC 曲线) 和 VLC (velocity limit curve) 曲线上的各速度关键点 — 切换点;

2) 根据各个速度关键点、每个关键点的控制轴及 VLC 曲线, 计算出实际加工速度曲线.

3) 根据实际加工速度曲线与插补误差计算插补点, 得到实时插补的数据点.

3 二次 B 样条曲线的 G01 代码拟合

本节, 我们详细讨论 G01 代码的分段、特征点选取和二次 B 样条曲线拟合待加工路径的方法.

3.1 G01 代码的分段

G01 代码所表示的空间点集稠密、数据量庞大, 一般情况下不能只用一条 B 样条曲线进行拟合. 因此, 首先需要将 G01 代码按照一定规则进行分段, 使得每两个分段点间可以用一条 B 样条曲线来逼近.

在分段过程中, 如果两个相邻 G01 代码所表示的折线段长度大于一个预先给定的阈值, 则将该段保留为一个折线段, 不做样条拟合. 对非长折线段的 G01 代码, 还需根据其曲率特征进行分段.

在数控加工过程中, 速度和精度是相互制衡的两个因素, 因此当刀具运行至转角处时, 若曲率较小, 则刀具可以较高速度通过; 否则, 为了保证加工的精度, 刀具需减速运行. 特别是, 当曲率值很大, 即角度非常尖锐时, 应将该位置所对应数据点作为样条曲线的分段点, 以保证加工误差能够满足系统预设精度要求. 因此, 本文将 G01 代码所对应的离散曲率值作为分段的标准, 并根据机床的机械特性, 确定曲率的阈值, 将离散曲率值大于阈值的数据点作为样条曲线的分段点.

按照文献 [22] 所述方法, 可利用相邻三点计算出中间点的离散曲率值. 在得到所有 G01 代码所对应的离散曲率值后, 找出曲率的极值点, 作为初步的特征点.

刀具加工的法向加速度 a_N 、进给速度 v 、加工点所对应的曲率 κ 及曲率半径 r 满足 $r = \frac{v^2}{a_N} = \frac{1}{\kappa}$, 因此, 要保证刀具能够进行平滑加工, 要求加工轨迹的最大曲率为

$$\kappa_{\max} = \frac{a_{N\max}}{v^2}. \quad (1)$$

可将该值作为曲率阈值. 如果上述初步特征点中, 存在曲率超过阈值 κ_{\max} 的点 p_i , 则将 p_i 作为样条曲线的分段点.

值得注意的是, 该曲率阈值仅适用于匀速加工的情形, 其中 v 即为匀速加工的进给速度. 在实际加工中, 采用本文所提供的速度规划方法, 在曲线的每点处速度大小都会有变化. 并且, 在加工轨迹曲率较大处, 速度会有所下降, 所以, 并不适合直接由公式 (1) 给出曲率阈值的计算公式. 为此, 我们根据机床的机械特性及加工方式给出了曲率阈值的一个估计值. 即当数据点所对应离散曲率值满足下述条件时,

$$\kappa > \kappa_{\max} = \alpha \frac{a_{\max}}{F^2}, \quad (2)$$

即将该点作为样条曲线的分段点. 其中, a_{\max} 为各个加工轴最大加速度计算而得的合成最大加速度, F 为系统预设的编程速度, α 为一有理系数, 由经验值给出, 本文采用 $\alpha = 4$ 或 9 .

3.2 自适应选取特征点

通过上述方法对 G01 代码进行分段后, 在每两个分段点间的加工路径可用一条二次 B 样条曲线来描述. 为了提高计算效率, 本文采取样条插值的方法. 一般情况下, 在每两个分段点间仍存在大量数据点, 如果按照文献 [3, 4, 23] 中所述方法, 令 B 样条曲线穿过所有数据点, 则在加工复杂形状时会出现样条曲线中多项式段数过多, 且每段多项式曲线过短的问题, 并且, 在数据点非常稠密的情况下, 曲线可能会出现不规则形状 [24]. 这样不仅不能达到数据压缩的目的, 过短的曲线段也会影响速度规划的计算效率.

因此, 我们在利用 B 样条曲线进行插值前, 要先自适应地提取原始数据点中能够描述待加工路径形状特征的特征点. 该步骤的操作对象为每两个分段点间的所有数据点. 本文提出的自适应选取特征点分为以下两步:

首先, 在上述每两个分段点间, 根据 3.1 小节中所计算出的各点离散曲率值, 找出曲率的极值点, 作为初步特征点保留.

之后, 利用二次 Bézier 曲线添加特征点. 设输入的一组初步特征点为 $\{P_{i1}, \dots, P_{ir}\}$, 其中 P_{i1}, P_{ir} 为两个分段点. 已知二次 Bézier 曲线具有如下表达式:

$$B(u) = (1-u)^2 Q_0 + 2u(1-u) Q_1 + u^2 Q_2, \quad 0 \leq u \leq 1,$$

其中, u 为参数, Q_0, Q_1, Q_2 为控制点.

用二次 Bézier 曲线插值每相邻 3 个特征点 $\{P_{ij-1}, P_{ij}, P_{ij+1}\}$, 有 $Q_0 = P_{ij-1}, Q_2 = P_{ij+1}, u_0 = 0, u_2 = 1$, 并利用累加弦长法计算 u_1 的值, 便可得关于 Q_1 的线性方程, 进而得到二次 Bézier 插值曲线 $B(u)$.

在得到二次 Bézier 插值曲线后, 找到 P_{ij-1}, P_{ij+1} 在原 G01 代码序列中的位置, 并计算上述两个对应位置之间除 P_{ij} 之外的所有数据点到插值曲线 $B(u)$ 的距离. 若上述距离值均小于系统给定的精度要求, 则不需要添加特征点, 且下一次插值曲线的起点为 P_{ij} ; 否则, 将其中最大距离值所对应的数据点 P_k 作为新的特征点添加到原有的特征点序列中.

重复上述过程, 直至分段点 $P_{ir} = Q_2$ 且该段插值曲线所对应数据点到该曲线距离均满足系统预设精度要求. 从而, 得到了一组新的特征点序列 $\{P_{i1}, \dots, P_{ir'}\}$.

如图 2 所示, 其中图 (a) 为根据曲率阈值选取的一段 G01 代码, 用 “o” 表示, 两端用 “+” 表示的点即为满足条件 (2) 的样条曲线分段点; 图 (b) 中标记 “□” 的点为曲率极值点, 即初步特征点;

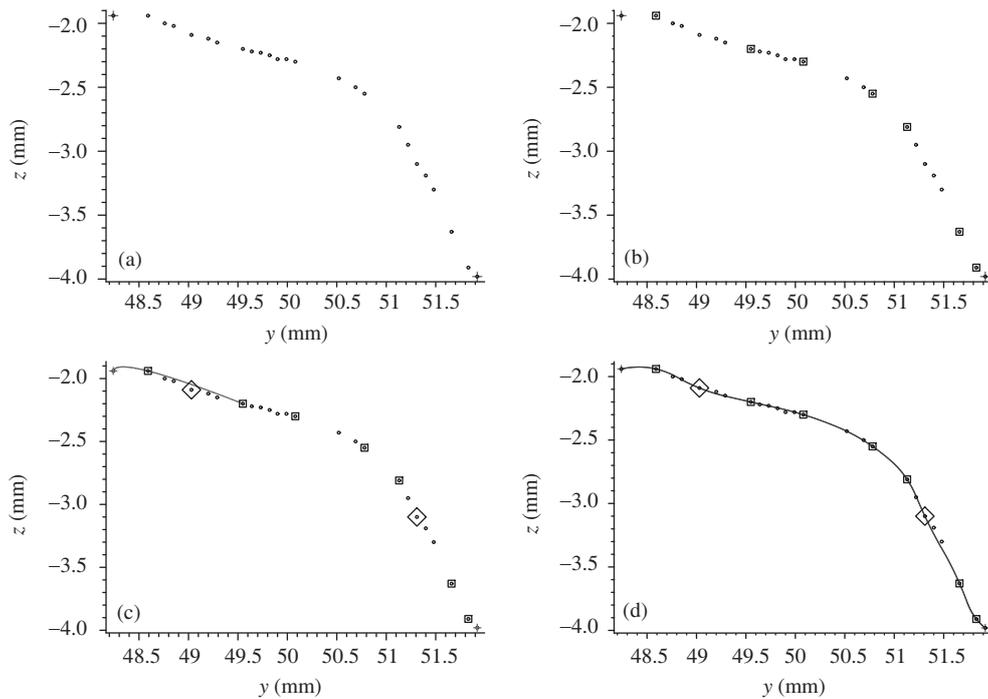


图 2 自适应二次 B 样条拟合

Figure 2 Adaptive quadratic B-spline fitting (a) A section of G01 codes between two breaking points; (b) curvature extreme points; (c) adaptively adding new feature points; (d) quadratic B-spline fitting

图 (c) 中的一段曲线即为插值相邻 3 个初步特征点的二次 Bézier 曲线 (只显示了需要加入新特征点的其中一段), “◇” 所标注的点为到曲线距离不能满足精度要求的点, 即需要作为新的特征点加入原有特征点序列; 图 (d) 为二次 B 样条曲线插值所有特征点得到的加工路径图. 从图中可以看出, 我们提出的自适应添加特征点的方法, 可以很好地捕捉到加工路径的形状特征.

算法 1 自适应特征点添加算法.

输入: 两分段点间的一组特征点序列 $\{P_{i1}, \dots, P_{ir}\}$, 其中 P_{i1}, P_{ir} 为分段点.

输出: 新的特征点序列 $NewP_i = \{P_{i1}, \dots, P_{ir'}\}$.

设集合 $\{Q_1, Q_2, Q_3\}$ 为用二次 Bézier 曲线插值的相邻特征点, 且令 $Q_1 = P_{i1}, Q_2 = P_{i2}, Q_3 = P_{i3}$.

设 $NewP_i$ 为新的特征点序列, 且初值为 $NewP_i = \{P_{i1}, \dots, P_{ir}\}$.

- 1) 用二次 Bézier 曲线插值 $\{Q_1, Q_2, Q_3\}$ 三点, 得到曲线 $B(u)$.
- 2) 找到 $\{Q_1, Q_2, Q_3\}$ 三点在原 G01 代码中的对应位置, 并计算 Q_1, Q_3 间除 Q_2 外的所有数据点到二次插值曲线 $B(u)$ 的距离.
- 3) 选择上述所有距离中的最大值, 如果该值大于系统预设精度要求, 则选出最大距离值所对应数据点 P_k , 如果该点在原 G01 代码中位置在 Q_2 之前, 则令 $Q_3 = Q_2, Q_2 = P_k$; 否则, $Q_3 = P_k$; 返回 1).
- 4) 若 Q_2, Q_3 不在 $NewP_i$ 中, 则将其依次加入特征点序列 $NewP_i$.
- 5) 如果 $Q_3 \neq P_{ir}$, 则令 Q_3 在 $NewP_i$ 的下一个相邻点为 P_{next} , 且令 $Q_1 = Q_2, Q_2 = Q_3, Q_3 = P_{next}$, 返回 1).

3.3 二次 B 样条曲线拟合待加工路径

二次 B 样条曲线可表示为

$$C(u) = \sum_{i=0}^n Q_i N_2^i(u), \quad 0 \leq u \leq 1, \quad (3)$$

其中, $Q_i (i = 0, 1, \dots, n)$ 为 $n + 1$ 个控制点, $N_2^i(u)$ 为定义在结点向量 $\mathbf{T} = \{t_0, t_1, \dots, t_{n+3}\}$ 上的二次 B 样条基函数, u 为参数.

按照样条插值法, 利用上述得到的一组特征点序列, 可构造关于待求控制点序列的一个线性方程组. 并且, 由二次 B 样条曲线得到的该方程组的系数矩阵是一个三对角矩阵, 可用追赶法快速求解.

在对 G01 代码进行拟合时, 我们选取了二次 B 样条曲线作为工具, 不仅可以极大的降低速度规划算法的计算规模, 并且使得求解控制点序列的线性方程组的系数矩阵简化为一个三对角形式, 避免了文献 [4, 25] 中需要使用 Gauss 消去法求解一般线性方程组的计算, 提高了计算效率.

3.4 总体算法

总结上述, 我们得到对 G01 代码的自适应二次 B 样条的拟合算法.

算法 2 自适应二次 B 样条拟合算法.

输入: G01 代码 $\{P_0, \dots, P_n\}$.

输出: 二次 B 样条曲线集合 $\{C_1(u), \dots, C_m(u)\}$.

1) 计算曲率阈值 κ_{\max} .

2) 计算 G01 代码所对应离散曲率值 $\kappa_i, (i = 0, \dots, n)$, 选取曲率极值点作为初步特征点, 并将曲率值大于 κ_{\max} 的数据点作为样条曲线分段点.

3) 对每两个分段点间的特征点集合 $\{P_{i1}, \dots, P_{ir}\}$, 根据系统预设的精度要求, 按照算法 1 所述方法自适应添加特征点, 得到新的特征点集合 $\{P_{i1}, \dots, P_{ir'}\}$.

4) 用二次 B 样条曲线插值特征点集合 $\{P_{i1}, \dots, P_{ir'}\}$, 得到描述待加工路径的拟合曲线 $C_i(u)$.

图 3 是对数控加工工件中“花瓶”的一段较复杂 G01 代码拟合的效果图. 图中共有 486 个数据点, 拟合为 34 条二次 B 样条曲线, 共包含 173 个二次多项式. 为了使图像形状特征更加清晰, 显示时将纵坐标做了拉伸处理.

4 基于二次样条曲线的最优加工方法

以下我们考虑的曲线是三维空间中的样条曲线. 在对样条曲线进行插补之前, 需要知道对应于每一参数值的速度值. 求解这一速度曲线的过程称为速度规划. 为此, 我们首先要分析一下曲线的局部性质. 下文中我们记对参数 u 的导数为“'”.

我们针对第 3.3 小节中的公式 (3) 所描述的参数曲线 $C(u)$, 记这一曲线的参数速度为 $\sigma(u) = \frac{ds}{du} = |C'(u)|$. 因为 $v(u) = \frac{ds(u)}{dt}$, $\frac{d}{dt} = \frac{ds}{dt} \frac{du}{ds} \frac{d}{du} = \frac{v}{\sigma} \frac{d}{du}$, 我们希望求得在满足各轴加速度条件下的沿曲线的加工时间最短的速度规划 $v(u) = \frac{ds}{dt}$. 或者说是下面的最优化问题

$$\min_{v(u)} T = \int_0^T dt = \int_0^1 \frac{\sigma}{v} du, \quad (4)$$

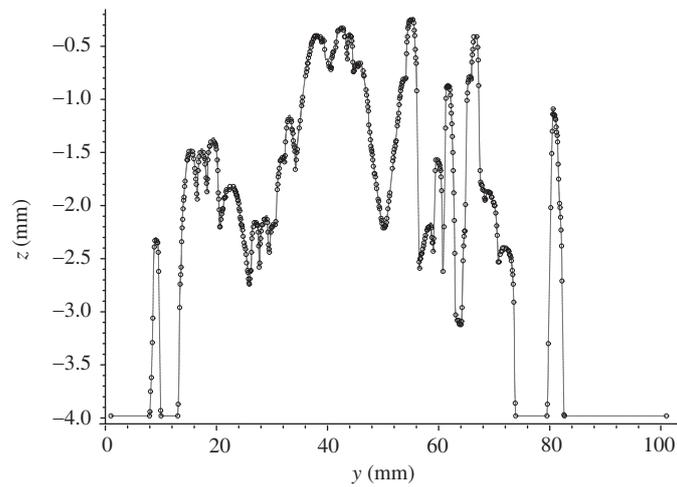


图 3 拟合效果图

Figure 3 Fitting result

满足

$$|a_x(u)| \leq A_x, |a_y(u)| \leq A_y, |a_z(u)| \leq A_z, \forall u \in [0, 1], \quad (5)$$

其中, (a_x, a_y, a_z) 表示加速度在各个坐标轴上的分量, (A_x, A_y, A_z) 表示各轴的最大加速能力.

4.1 基于二次 B 样条的改进速度规划方法

文献 [17] 针对上述的问题 (4), 提出了一种基于“Bang-bang”最优控制思想的计算方法, 即总有一个分量的加速度达到最大. 该算法对于高次曲线需要求解较为复杂的代数方程, 运用数值的方法求解将带来极大计算量, 很难达到实时的要求. 本小节, 我们对于二次曲线, 在文献 [17] 算法的基础上提出了实质性的改进, 大大减少了计算量.

如果我们使用“Bang-bang”的最优控制方法, 即在任一时刻, 公式 (5) 中至少有一个等号成立. 那么不妨设 $a_x = \pm A_x$, 或者说 x 轴为控制轴, 则 $\frac{q}{2\sigma^2}x' + \frac{q}{\sigma^3}(\sigma x'' - \sigma'x') = \pm A_x$, 其中 $q = v(u)^2$. 解这一微分方程得到

$$q = \left(\frac{\sigma}{x'}\right)^2 (n \pm 2A_x x), \quad (6)$$

其中, n 是某个常数, 具体计算时可以由在特定点上的值求得: $n = \left(\frac{x'(u_*)}{\sigma(u_*)}\right)^2 q(u_*) \mp 2A_x x(u_*)$.

因此, 在 Bang-bang 控制下, 我们只要求得上述积分速度曲线的起点终点 (分段点)、起点终点处的速度值和相应的控制轴就可以了. 为了求得速度曲线的分段点, 我们需要引入最大速度曲线 (velocity limit curve, 简称 VLC 曲线), 也就是对于每一个参数值 u , 它所对应的速度最大允许值连成的曲线.

根据文献 [17] 的方法, 容易求得 VLC 曲线的解有如下的形式:

$$v_{\text{lim},P,R}^2 = \frac{\sigma^2(\alpha_R A_R P' - \alpha_P A_P R')}{P'R'' - P''R'}, \quad (7)$$

其中 $v_{\text{lim},P,R}$ 表示由 P 轴和 R 轴确定的 VLC 曲线, $\alpha_P = \pm 1$, $\alpha_R = \pm 1$, P, R 是三轴 x, y, z 中的任意两轴.

注 1 为表示方便, 本小节中 P, R, M, N, K 和 H 均表示 x, y, z 加工轴的其中之一轴. 例如当 $\alpha_P = 1$ 时, 对应的 P 轴表示 $+P$ 轴, 当 $\alpha_P = -1$ 时, 对应的 P 轴表示 $-P$ 轴; 对应的 A_P 表示 P 轴的最大加速度值 ($A_P > 0$). 下文中令 P 轴方向的二次样条曲线具有如下的形式: $P(u) = a_P u^2 + b_P u + c_P$, 其中 a_P, b_P 和 c_P 分别为二次项、一次项和常数项系数.

假定此时速度规划的控制轴为 $+P$ 轴, 那就要求解方程 (6) 和 (7) 的交点, 而且是第一个交点, 那么我们势必要求解一个高次方程的根, 一般情况下就必须用数值算法, 这就带来了一个计算量大并且还要考虑算法鲁棒性和快速性的问题, 因而也就增大了计算的负担, 很难达到实时性. 因此针对这一问题我们在计算过程中尽量避免求解速度规划曲线同 VLC 曲线相交的信息.

下面我们看一下, 当我们采用二次样条曲线时, VLC 曲线上三类切换点^[17]的计算.

(i) 切点: 由于加工曲线最多是二次的, 根据计算我们不必求解切点, 这样减少了大量计算.

(ii) 临界点和不连续点.

首先我们可以获得分段点的参数值, 而这些分段点又是可能的临界点和不连续点. 另外当 P 轴和 R 轴控制时, 计算控制轴发生改变的点可以得到另一部分临界点, 具体的参数值可通过求解线性方程得到: $\alpha_1 A_R P' - \alpha_2 A_P R' = \alpha'_1 A_R P' - \alpha'_2 A_P R'$, 其中, $\alpha_i, \alpha'_i, i = 1, 2$ 的取值为 ± 1 .

由于 P 轴和 R 轴可以是 x 轴、 y 轴和 z 轴中的任意两个轴, 因此最终的 VLC 曲线是三轴中任意两轴确定的 VLC 曲线中最小的一个, 因此

$$v_{\text{lim}}^2 = \min_{P, R \in \{x, y, z\}} (v_{\text{lim}, P, R}^2). \quad (8)$$

通过求解不同控制轴 (P, R) 和 (M, N) ($P, R, M, N \in \{x, y, z\}$) 确定的 VLC 曲线的交点, 可得交点参数有明确的显示表达式

$$u = \frac{(\alpha_N A_N b_M - \alpha_M A_M b_N)(2a_R b_P - 2a_P b_R) - (\alpha_R A_R b_P - \alpha_P A_P b_R)(2a_N b_M - 2a_M b_N)}{(2\alpha_R A_R a_P - 2\alpha_P A_P a_R)(2a_N b_M - 2a_M b_N) - (2\alpha_N A_N a_M - 2\alpha_M A_M a_N)(2a_R b_P - 2a_P b_R)}. \quad (9)$$

以上得出的参数值就是可能的临界点和不连续点的参数值, 根据公式 (8) 可得出最终 VLC 曲线. 最后分别判断以上得到的参数值就可以求出最终的临界点和不连续点.

我们现在在求解三类切换点时, 若采用二次样条曲线, 那么切点不必求, 临界点和不连续点的求解过程可以直接计算出显示解, 这些都大大降低了计算量.

我们在求具体速度规划曲线时, 需要求解加工控制轴的转换点, 如果当前的加工控制轴为 $+P$ 轴, 由它计算得到的速度规划曲线由公式 (6) 可得 $v^2 = (\frac{\sigma}{\sigma'})^2 (n_P + 2A_P P)$, 此时通过求解 R 轴的加速度方程: $\frac{q'}{2\sigma^2} R' + \frac{q}{\sigma^3} (\sigma R'' - \sigma' R') = \pm A_R$, 可求得最终的转换点参数为一个三次方程的根, 并且该方程的实根只有一个

$$u = \frac{\sqrt[3]{(a_R b_P - a_P b_R)(4a_P A_P c_P + 2a_P n_P - A_P b_P^2)(a_R A_P - A_R a_P)^2 + b_P a_R A_P - b_P A_R a_P}}{2(-a_R A_P + A_R a_P) a_P}.$$

因此, 在求解加工轴的转换处的参数时, 由于采用了二次的样条曲线, 转换处的参数具有显示的表达式.

基于上述的讨论, 我们提出一种改进的速度规划算法, 此处仍以三轴加工为例.

算法 3 基于 Bang-bang 控制的二次样条曲线最优速度规划算法.

输入: 分段的二次样条曲线 $C(u) = (x(u), y(u), z(u)) (0 \leq u \leq 1)$, 它们在连接点处可以达到 C^1 连续.

输出: 分段速度规划曲线 $v_{sd}(u)(0 \leq u \leq 1)$.

1) 求解出 VLC 曲线的表达式以及临界点和不连续点的信息.

2) 从 $u = 0$, 初速度 $v(0) = v_0$ 开始, 计算加工控制轴 H , 根据该控制轴求出当前速度加工曲线. 随着 u 的增加, 计算任意非加工轴的加速度达到约束边界时的信息 (换轴), 也就是 $\frac{vv'}{\sigma^2} H' + \frac{v^2}{\sigma^3} (\sigma H'' - \sigma' H') = \alpha_K A_K$, 其中 K 为当前加工轴 H 之外的其他任意一轴. 求得换轴的参数 u 及对应的速度值. 检验在到达该参数前是否在切换点处存在超出各轴加速度的点, 若存在, 将 u 改成满足上述条件的最小的切换点处的参数值. 重新计算加工轴并获得新的速度规划曲线一直到参数 $u = 1$, 得到速度规划函数 v_F ;

从 $u = 1$, 初速度 $v(1) = v_e$ 开始, 计算加工控制轴 H , 根据该控制轴求出当前速度加工曲线. 随着 u 的递减, 当任意非加工轴的加速度达到约束边界时, 求得换轴的参数 u 及对应的速度值, 重新计算加工轴并获得新的速度规划曲线一直到参数 $u = 0$, 得到速度规划函数 v_B ;

3) 计算正向和逆向速度规划曲线 v_F 和 v_B 的交点为 $(u_{in}, v(u_{in}))$, 其中 $v(u_{in}) = v_F(u_{in}) = v_B(u_{in})$, 并获得速度规划曲线

$$v_{sd}(u) = \begin{cases} v_F, & 0 \leq u < u_{in}, \\ v_B, & u_{in} \leq u \leq 1. \end{cases}$$

4) 检验是否所有的 VLC 曲线上的临界点和不连续点都位于当前速度规划曲线 v_{sd} 的上方, 如果是, 那么我们就获得了最终的速度规划曲线 v_{sd} , 并且计算终止; 否则的话, 继续下面的步骤.

5) 选择当前速度规划曲线 v_{sd} 下方, 参数值最小的切换点 $(u_{sp}, v_{lim}(u_{sp}))$; 计算加工控制轴, 根据加工控制轴, 分别得到从 $(u_{sp}, v_{lim}(u_{sp}))$ 开始, 逆向 (u 递减) 和正向 (u 递增) 获得两组新的速度规划曲线 v_{sp-} 和 v_{sp+} , 再分别与当前的整体的速度规划曲线 v_{sd} 相交于 $(u_{sp-}, v_{sp-}(u_{sp-}))$ 和 $(u_{sp+}, v_{sp+}(u_{sp+}))$. 最后修正整体的速度规划曲线, 得到最新的的速度规划曲线 v_{sd} ,

$$v_{sd}(u) = \begin{cases} v_{sd}, & 0 \leq u < u_{sp-}, \\ v_{sp-}, & u_{sp-} \leq u < u_{in}, \\ v_{sp+}, & u_{in} \leq u < u_{sp+}, \\ v_{sd}, & u_{sp+} \leq u \leq 1. \end{cases}$$

返回到步骤 4).

注 2 同文献 [17] 描述的算法相比, 我们虽然避免了求解速度规划曲线同 VLC 曲线的交点, 但要求解速度规划曲线超越 VLC 曲线后的速度规划曲线, 因此在我们的具体算法设计中, 当速度规划曲线超越 VLC 曲线后, 我们用一个很大的速度曲线来取代超越部分, 这是因为最终的速度规划曲线是位于 VLC 曲线下方的, 而超越 VLC 曲线部分会在以后的计算当中被取代, 采用这种方法是为了得到完整的速度规划曲线, 从而简化计算. 最终的实际加工速度曲线是上述曲线求最小得到的.

4.2 二次样条曲线插补方法

在求取曲线的插补点时, 对给定参数值 u_i , 根据速度规划曲线 v_{sd} 和 VLC 曲线 v_{lim} 得到速度值, 即令 $v(u_i) = \min(v_{sd}(u_i), v_{lim}(u_i))$, 计算步长 $\Delta L = v(u_i) \cdot T$ (T 是机床的插补周期), 再根据 ΔL 通过一阶近似求得近似插补点, 用 Newton 迭代算法 [26] 逼近插补点的参数值 u_{i+1} .

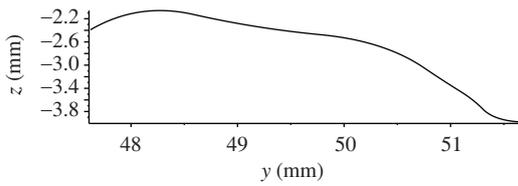


图 4 一段拟合曲线
Figure 4 A fitting curve

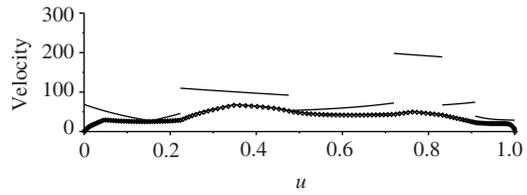


图 5 VLC 曲线、速度规划曲线和插补点速度
Figure 5 VLC, velocity planning curve, and velocities at the interpolation points. The unit for the velocities are mm/s

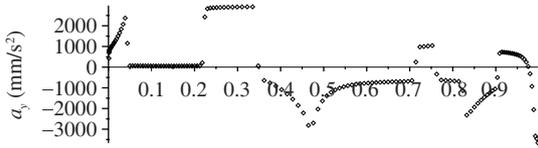


图 6 计算的 y 轴加速度图
Figure 6 Computed acceleration chart of y-axis. The unit for the acceleration is mm/s²

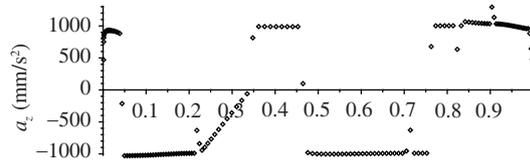


图 7 计算的 z 轴加速度图
Figure 7 Computed acceleration chart of z-axis. The unit for the acceleration is mm/s²

4.3 曲线仿真

拟合曲线是加工工件花瓶中的一段样条曲线,如图 4 所示,由 6 条分段的二次 B 样条曲线连结成的,在连结处达到 C^1 连续,参数变化范围是 $0 \leq u \leq 1$. 加工时用的是截平面的方法,即相当于二轴 (y 轴和 z 轴) 控制加工,加工控制轴的最大速度限制为 $A_y = 3000 \text{ mm/s}^2$ 和 $A_z = 1000 \text{ mm/s}^2$,经计算后,VLC 曲线和速度规划曲线如图 5 所示,加工轴 y 轴方向和 z 轴方向的实际计算加速度值如图 6 和 7 所示. 可以看到,实际加工控制时基本达到了“Bang-bang”最优控制.

5 实例仿真

本节,我们将对 3 个用于实际加工的模型进行仿真测试,这 3 个实例分别是“花瓶”、“北京奥运标志”和“头像”(如图 8). 其中“北京奥运标志”图案是比较简单的图案,“花瓶”是纹路较多较为复杂的两轴加工的典型图案,“头像”是三轴控制的非常有代表性的实际加工例子,加工变化较多,较为复杂. 所有的测试都是在 Windows XP 的操作平台下的 VC++6.0 开发环境中进行的,其中的硬件配置是 Intel Core 2 的 CPU(主频是 2.13 GHz), 1.98 GB 的内存.

首先运用第 3 节提出的二次曲线拟合的算法 2,得到的花瓶、奥运、头像图案的样条压缩比(平均每个样条对应的小线段数目)分别为 27.3, 156.3, 74, 多项式压缩比(平均每个多项式对应的小线段数目)分别为 3.6, 15.5, 8.1. 然后,我们采用第 4 节提出的速度规划算法 3,对拟合的二次曲线进行速度规划并进行插补得到的数据结果与匀速过渡插补算法^[7]的结果比较效率提高了 65%–130%,与传统的圆弧过渡插补算法 1 比较效率提高了 30%–110%.

在上述 3 个例子的计算时间上,拟合算法的计算用时相对于后面的速度规划与插补可以忽略,速度规划改进算法的速度规划曲线计算用时与插补计算用时大体相当,但远小于加工所用的时间,因此我们的算法具有很好的实时性.



图 8 待加工样件

Figure 8 Workpieces

(a) "Vase" workpiece; (b) "Beijing Olympic Games" workpiece; (c) "Head-sculpture" workpiece



图 9 “花瓶”的实际数控加工图

Figure 9 Machine picture

(a) "Vase" NC machining picture; (b) Machine picture

从上述 3 个例子的仿真测试结果来看, 我们的算法对于非常复杂的例子 (“花瓶” 和 “头像” 的图案) 和较简单的例子 (“北京奥运标志” 图案) 都取得了较好的结果.

图 9 给出了图案 “花瓶” 的加工实例, 这些加工实例表明, 我们的算法是鲁棒并且实用的.

6 结论

本文考虑的数控加工方案是基于二次 B 样条曲线上的时间最优速度规划和插补. 对 G01 代码先通过自适应方法选取 G01 代码所描述的每组小线段的各个特征点, 再用通过所有特征点的二次 B 样条曲线拟合待加工路径. 从仿真的实例来看, 我们提出的数据压缩方法具有计算速度快、加工精度高、工作性能稳定, 且适用范围广泛的特点.

根据所得的二次 B 样条曲线的性质和数控机床的加工性能, 采用了最优控制意义下的 Bang-bang 控制方式求取实际加工速度曲线, 在求解过程充分利用二次曲线的性质, 我们所采用的方法主要解决了以下几个问题:

- 1) 避免求解加工速度曲线同 VLC 曲线相交的交点信息, 避免求解高次方程, 减少了大量的计算;
- 2) 采用二次样条曲线, 在求解过程中减少了计算步骤和高次方程的出现, 基本上都可以直接求得显示解, 避免了求解过程中因采用数值方法而带来的适用性差和计算复杂的问题.

然而, 我们的方法是对一条曲线整体考虑的, 对于待加工曲线的起始速度和结束速度都必须有明确的要求, 要想再进一步提高加工速度, 就需要对这两个速度进行规划, 而整体的考虑势必涉及大量的回溯计算, 这样无形中增加了计算量, 会降低实时性. 因此我们今后的工作重点将致力于速度整体规划变为局部规划, 使得速度规划在保证整体最优的情况下, 也同时可以规划加工曲线的起始速度和结束速度, 仍在满足系统预设精度前提下, 实时完成样条曲线的插补计算, 满足高速、高精度数控加工需求.

致谢 感谢中科院沈阳计算所协助在其数控机床与数控系统中实现本文算法并进行实物加工.

参考文献

- 1 Lv Q, Zhang H, Yang K M, et al. Study on the method of increasing turning velocity during CNC continuous machining (in Chinese). *Manuf Tech Mach Tool*, 2008, 3: 79–83
- 2 Zhang L X, Sun R Y, Gao X S, et al. An optimal solution for high-speed interpolation of consecutive micro-line segments and adaptive real-time look-ahead scheme in CNC machining (in Chinese). *MM-Preprints*, 2010, 29: 206–227
- 3 Yau H T, Wang J B. Fast Bezier interpolator with real-time look-ahead function for high-accuracy machining. *Int J MachTools Manuf*, 2007, 47: 1518–1529
- 4 Wang J B, Yau H T. Real-time NURBS interpolator: application to short linear segments. *Int J Adv Manuf Tech*, 2009, 41: 1169–1185
- 5 Zhao X, Yin Y, Yang B. Dominant point detecting based non-uniform B-spline approximation for grain contour. *Sci China Ser E-Tech Sci*, 2007, 50: 90–96
- 6 Farin G, Hoschek J. *Handbook of Computer Aided Geometric Design*. Elsevier Science, 2002
- 7 Wang R H. *Numerical Rational Approximation* (in Chinese). Shanghai: Shanghai Science and Technology Press, 1980
- 8 Ye P Q, Zhao S L. Study on control algorithm for micro-line continuous interpolation (in Chinese). *China Mech Eng*, 2004, 15: 1354–1356
- 9 Erkorkmaz K, Altintas Y. High speed CNC system design. Part I: Jerk limited trajectory generation and quintic spline interpolation. *Int J Mach Tools Manuf*, 2001, 41: 1323–1345
- 10 Yu D, Hu S H, Gai R L, et al. Research on acceleration and deceleration for CNC machine tools based on filtering (in Chinese). *China Mech Eng*, 2008, 19: 804–807
- 11 Gai R L, Lin H, Zheng L M, et al. Design and implementation of velocity planning algorithm for high speed machining (in Chinese). *J Chin Comput Syst*, 2009, 30: 1067–1071
- 12 Shi C, Zhao T, Ye P Q, et al. Study on S-shape curve acceleration and deceleration control on NC system (in Chinese). *China Mech Eng*, 2007, 18: 1421–1425
- 13 Du D S, Yan C L, Li C X. An adaptive NURBS interpolator with real-time look-ahead function (in Chinese). *J Shanghai Jiaotong Univ*, 2006, 40: 843–847
- 14 Lin M T, Tsai M S, Yau H T. Development of a dynamics-based NURBS interpolator with real-time look-ahead algorithm. *Int J Mach Tools Manuf*, 2007, 47: 2246–2262
- 15 Liu X S, Jia Q X, Yuan X H. A study of NURBS interpolation algorithm and ACC-DEC control (in Chinese). *Modul Mach Tool Automat Manuf Tech*, 2007, 2: 60–64
- 16 Yuan C M, Gao X S. Time-optimal interpolation of CNC machines along parametric path with chord error and tangential acceleration bounds. *MM-Preprints*, 2010, 29: 165–188
- 17 Sebastian D T, Farouki R T, Smith T S, et al. Algorithms for time-optimal control of CNC machines along curved tool paths. *Robot Com-Int Manuf*, 2005, 21: 37–53
- 18 Zhang K, Gao X S, Li H B, et al. A Greedy Algorithm for feed-rate planning of CNC machines along curved tool paths with confined jerk for each axis. *MM-Preprints*, 2010, 29: 189–205
- 19 Bobrow J E, Dubowsky S, Gibson J S. Time-optimal control of robotic manipulators along specified paths. *Int J Robot Automat*, 1985, 4: 3–17

- 20 Shiller Z, Lu H H. Robust computation of path constrained time optimal motions. In: Proceedings, IEEE International Conference on Robotics and Automation, Cincinnati, OH, 1990. 144–149
- 21 Timar S D, Farouki R T. Time-optimal traversal of curved paths by Cartesian CNC machines under both constant and speed-dependent axis acceleration bounds. *Robot Com-Int Manuf*, 2007, 23: 563–579
- 22 Coeurjolly D, Svensson S. Estimation of curvature along curves with application to fibres in 3D images of paper. *Lect Notes Comput Sci*, 2003, 2749: 247–254
- 23 Lin H, Wang G, Dong C. Constructing iterative non-uniform B-spline curve and surface to fit data points. *Sci China Ser F-Inf Sci*, 2004, 47: 315–331
- 24 Piegl L, Tiller W. *The NURBS Books*. Berlin: Springer, 1997
- 25 Kang S J, Zhou L S, Zhang D L. Fitting cutter location points with B-spline curves accurately (in Chinese). *Mech Eng Automat*, 2007, 26: 95–97
- 26 Shi R M. *Numerical Computation* (in Chinese). Beijing: Advanced Education Press, 2004

Curve fitting and optimal interpolation on CNC machines based on quadratic B-splines

ZHANG Mei¹, YAN Wei², YUAN ChunMing^{1*}, WANG DingKang¹ & GAO XiaoShan¹

1 Key Laboratory of Mathematics Mechanization, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China;

2 Research Institute of Petroleum Exploration and Development, Beijing 100083, China

*E-mail: cmyuan@mmrc.iss.ac.cn

Abstract In this paper, curve fitting of 3-D points generated by G01 codes and interpolation based on quadratic B-splines are studied. Feature points of G01 codes are selected using an adaptive method. Next, quadratic B-splines are obtained as the fitting curve by interpolating these feature points. Computations required in implementing the velocity planning algorithm mentioned by Timer et al. are very complicated because of the appearance of high-order curves. An improved time-optimal method for the quadratic B-spline curves is presented to circumvent this issue. The algorithms are verified with simulations as well as on real CNC machines.

Keywords quadratic B-spline, velocity planning, G01 codes, interpolation