[研究·设计]

DOI:10.3969/j.issn.1005-2895.2017.03.005

基于 RTX 的参数化插补算法的仿真研究

高 甜,张立强,杜金锋

(上海工程技术大学 机械工程学院,上海 201620)

摘 要:在数控机床高速加工过程中,常规采用的增量式递推算法采样插补技术由于后一步的计算都是建立在前一步结果的基础上,容易产生误差累积效应。为了解决这一问题,笔者提出了参数化插补算法,并建立了参数化插补轨迹模型,基于自动加减速控制原理,自主开发设计了 Windows XP SP3 + RTX 实时扩展数控系统软件平台,通过平台进行仿真实验。结果表明提出的参数化插补算法能有效解决累积误差的问题,同时能够很好地减小弓高误差。

关 键 词:数控加工;参数化插补算法;自动加减速控制;累积误差;弓高误差

中图分类号:TG659 **文献标志码**:A **文章编号**:1005-2895(2017)03-0020-06

Simulation Research of Parametric Interpolation Algorithm Based on RTX

GAO Tian, ZHANG Liqiang, DU Jinfeng

(School of Mechanical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China)

Abstract: In the process of high speed machining of CNC machine tools, because the latter step was always based on the results of previous step during the sampling interpolation technique based on the conventional incremental algorithm, it was easy to cause cumulative error effect. In order to solve this problem, this paper proposed a parametric interpolation algorithm, established the parametric interpolation trajectory model, studied on automatic acceleration and deceleration control principle, independent designed Windows XP SP3 + RTX as software platform for real time extended numerical control system, the simulation experiment was carried out on the platform, the results show that the proposed parametric interpolation algorithm can effectively solve the problem of cumulative error, and reduce the chord error at the same time.

Keywords: CNC machining; parametric interpolation algorithm; auto acceleration and deceleration control; cumulative error; chord error

计算机数字控制系统最初由通用系统控制个人计算机的软件和硬件平台,实时性较弱。之后发展到利用总线和运动控制卡实现控制,实时性变强,但运动控制卡型号多样,虽然便于个人计算机软件包的开发,但不利于统一标准管理,不能实现各型号间的相互替代,导致以个人计算机加运动控制器的数控系统结构推广性不好。软件技术的发展,推动了数控系统向全软件开放式体系结构迈进,形成了 PC + I/O 形式的全软件化数控系统的体系结构。充分利用了个人计算机的开放性,由个人计算机完成数控装置的处理任务,实现多任务的强实时性控制。文献[1]提出以 Windows XP SP3 + RTX 实现扩展的数控系统软件平台,克服了 Windows XP 在数控系统控制中的弱实时性缺点。为了给数控系统中的运动控制任务处理提供一个强实时性的执行环境,结合 Windows 平台的丰富应用程序和多种开发工具以及 RTX 软件的实时处理能力,实现全软件数控系统实时性、系统开放性和可扩展性等各项性能的提升。

收稿日期:2016-10-24;**修回日期**:2017-02-06

基金项目:国家自然科学基金项目(51305254);上海市教委科研创新项目(13YZ108)。

第一作者简介:高甜(1993),女,山东枣庄人,硕士研究生,主要研究方向为精密制造与数控技术。E-mail:gtina19930910@163.com

机床加工过程中各个轴的误差综合起来就是总误差,若不妥善处理,工件的误差可能会超过许可误差。目前有许多减少机床误差的方法[2-4],但被证明有效的是参数化自动加减速控制。文献[5]提出通过参数化对数控系统进行调配,解决由多轴联动之间惯量不匹配造成的跟随误差和轮廓误差过大的问题。文献[6-8]提出通过转接的几何约束和运动约束以及转接参数的计算来实现高速插补的方法,同时采用 S 型加减速来规划速度和加速度。文献[10-11]分别介绍了直线加减速和 S 型加减速,并在误差补偿与控制过程对二者进行了对比。

笔者以 VC++6.0 面向对象开发语言为集成开发环境,建立数控软件进程体系结构,并搭建数控系统的硬件结构,实现数控系统的基本功能,通过参数化算法简化高速高精度插补过程中的计算问题,并通过自动加减速控制计算不同插补位置的插补直线段的长度,进而来约束插补过程中的弓高误差,保证插补的精度。最后借由 RTX 实时仿真实验,验证该算法的可行性和有效性。

1　软件体系结构设计

由于 Windows XP 实时性弱但界面强大,设计开放式数控系统软件,图 1 所示为创设软件结构体系。

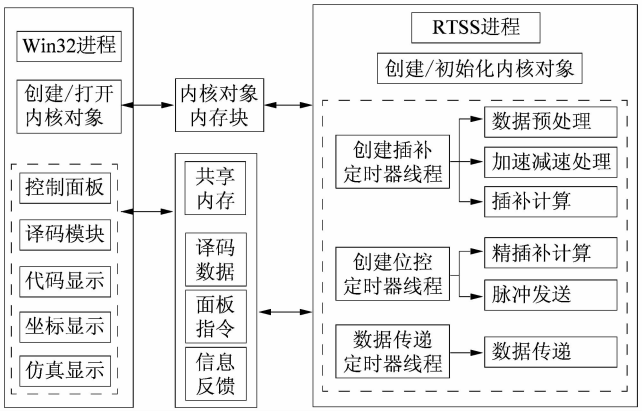

图 1　软件结构体系图

Figure 1　Software architecture diagram

运用 VC++6.0 作为开发工具,首先使用 MFC 建立可执行的文件的程序向导作为 Win32 的进程环境,然后再建立一个 RTX 的程序向导作为 RTSS 进程环境,将两个进程进行相关设置,实现 Win32 利用相应的函数调用 RTSS 进程中的文件,使两个进程之间能够实现内存共享机制,并且能够互相通信。

2　控制算法

常规采用的增量式递推算法采样插补技术虽然每一步的工作量很小,计算较快,但是由于后一步的坐标的确定都是建立在前一步坐标上的,容易造成累积误差。在高速高精度插补过程当中,因为每一插补间断内的插补直线段都十分短,导致插补步数较多,如果采取常规的增量式递推算法采样插补技术将会造成误差积累效应,引起对插补结果不能轻忽的影响。

需要采纳新的想法来设计插补算法,应对高速高精度插补过程的特殊性。新的插补算法的设计要求是:①创立方便规划被插补曲线的参数化模型,使之在进行实时插补运算过程中运算量小,不涉及繁复的函数计算,通过简单的几何运算解决问题。②保障每一个轨迹点坐标的计算基准都在模型的坐标原点,消除累积误差,同时保障在高速高精度加工过程中插补的计算精度和速度。

2.1　参数化插补算法的轨迹计算公式

将插补算法参数化使之符合对于新算法的规定。在插补算法中除轨迹通常表示为

$$\begin{cases} x = f_1(u), \\ y = f_2(u), \\ z = f_3(u)。 \end{cases}$$

式中,u 是参数化公式中的参变量,$u \in [0,1]$。

插补对象不同,轨迹计算参数化公式也不同。对圆弧插补时,其轨迹可表示为

$$[x \quad y] = \frac{1}{1+u^2}[2u \quad 1-u^2]\boldsymbol{M}。$$

式中,当插补点落在第一、二、三、四象限时,$\boldsymbol{M}$ 分别是取值为 $\begin{bmatrix} 0 & R \\ R & 0 \end{bmatrix}$, $\begin{bmatrix} -R & 0 \\ 0 & R \end{bmatrix}$, $\begin{bmatrix} 0 & -R \\ -R & 0 \end{bmatrix}$, $\begin{bmatrix} R & 0 \\ 0 & -R \end{bmatrix}$的对角矩阵;其中 R 为圆弧半径。

而对椭圆进行插补时,其轨迹公式表示为:

$$[x \quad y] = \frac{1}{1+u^2}[2u \quad 1-u^2]\boldsymbol{P}。$$

式中,$\boldsymbol{P}$ 也是常数矩阵,在第一、二、三、四象限分别为$\begin{bmatrix} 0 & B \\ A & 0 \end{bmatrix}$, $\begin{bmatrix} -A & 0 \\ 0 & B \end{bmatrix}$, $\begin{bmatrix} 0 & -B \\ -A & 0 \end{bmatrix}$, $\begin{bmatrix} A & 0 \\ 0 & -B \end{bmatrix}$, A, B 为椭圆长半轴和短半轴。

当插补曲线为位于 $x-y$ 坐标系第一象限的抛物线插补,其轨迹计算公式为:

$$\begin{cases} x = au^2, \\ y = bu。 \end{cases}$$

式中,a, b 是抛物线终点坐标的坐标值。

三次样条曲线插补的轨迹计算公式为:

$$\begin{cases} x = C_{3x}u^3 + C_{2x}u^2 + C_{1x}u + C_{0x}, \\ y = C_{3y}u^3 + C_{2y}u^2 + C_{1y}u + C_{0y}, \\ z = C_{3z}u^3 + C_{2z}u^2 + C_{1z}u + C_{0z}。 \end{cases}$$

式中，C_{ix}，C_{iy}，C_{iz}（$i=0,1,2,3$）是由曲线的型值点所决定的系数。

参数化插补算法所列出的具体的规划轨迹的公式的特点：没有复杂的函数计算，仅需少数的几何计算就可以求取插补轨迹上动点的坐标值，既能保证轨迹插补的速度，又能保证插补的精度。

2.2 实时插补计算

实行实时插补计算的主要目的是在各个插补间段内按照给定的参数化插补轨迹公式实时计算当前插补点的坐标值。因为插补运算并不是单纯的静态的几何运算，插补运算必须使得相邻差补点之间的距离符合插补周期内的进给速度，所以为了保障插补点能够按照给定的进给速度沿着被插补曲线的轨迹进行插补运动，产生符合要求的插补轨迹，参变量 u 的取值需要按照进给速度 F 来协定。以圆弧插补为例，参数化插补的计算原理如图 2 所示，设半径为 R 的圆弧起点为 $D(x_o,y_o)$，终点是 $E(x_e,y_e)$。其中点 P,Q 为邻近的两个插补点，ΔL_j 是插补步长，θ 是 ΔL_j 步距角。图 2 中 θ_j 有

$$u_j = \frac{1-\sin\theta_j}{\cos\theta_j}。$$

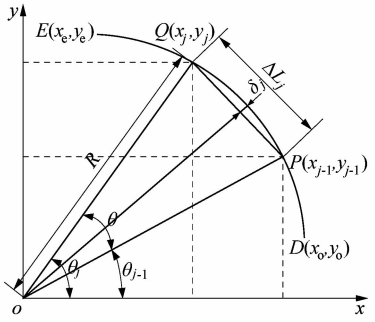

图 2 插补过程示意图

Figure 2 Schematic diagram of interpolation process

为了达到预定的标准，插补算法过程为：

1）由已经给定的每一段插补所用时间（插补周期）ΔT 和该插补时间段内的进给速度 F_j，则插补直线段的长度 ΔL_j 有

$$\Delta L_j = \frac{F_j \Delta T}{60\ 000}。$$

2）计算出每一个插补周期的参变量的变化量 Δu_j

$$\Delta u_j = \frac{\mathrm{d}u}{\mathrm{d}s}\Delta L_j。$$

式中：$\frac{\mathrm{d}u}{\mathrm{d}s}$ 为参变量随着曲线弧长变化的变化速率。

因为高速度、高精确度加工过程中插补频率较高，插补直线段极短，所以插补直线段与被插补曲线十分接近，可采取以直代曲的方法来计算参变量对于曲线弧长的变化率，即有

$$\frac{\mathrm{d}u}{\mathrm{d}s} \approx \frac{\Delta\hat{u}}{\Delta\hat{L}} = \frac{\Delta\hat{u}}{\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}}。$$

式中：$\Delta\hat{u}$，$\Delta\hat{L}$ 为对应变量的摄动量。根据这个近似公式就可以求出参数变化量对曲线弧长的变化率。

计算出 Δu_j，通过简单的迭代求取当前插补周期中的参变量 u_j 即

$$u_j = u_{j-1} + \Delta u_j。$$

式中，u_{j-1} 为上一个插补周期的参变量的值。

将当前插补周期中求得的参变量的取值，带入插补算法的计算轨迹的公式中，就可以计算出当前插补周期下轨迹上插补点的坐标了。从插补起点不间断重复进行上面计算至插补点结束，就可以获得整个的插补曲线的离散后轨迹。

2.3 插补误差控制

高速高精度插补过程中，运动速度的精确性与轨迹精确度相抵，轨迹精确度是主要矛盾，需要保证插补点的坐标精确性。倘若只按照进给速度进行插补运算能够保证有限的插补点在插补曲线上，保证不存在轨迹误差，但不能保证插补直线段与被插补曲线之间的弓高误差满足插补要求，因而必须要单独对弓高误差进行控制。误差关系简图如图 3 所示。

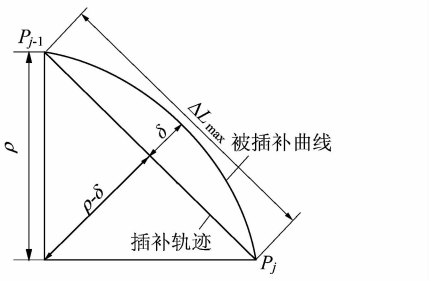

图 3 误差关系简图

Figure 3 Error relationship diagram

根据图 3 中所示的几何关系，求出每一个插补周期中，在满足允许误差 δ 条件下插补直线段的长度

$$\Delta L_{\max} = 2\sqrt{\delta(2\rho-\delta)}。$$

式中:ΔL_{max}作为实际插补过程中的弦长的约束的长度;ρ 为被插补弧线的半径。当根据进给速度计算出来的弦长小于约束弦长的长度时,则根据由进给速度规划出来的弦长进行插补轨迹的拟合,否则用约束弦长作为标准进行插补。这样规定可以避免在被插补曲线曲率比较小的部分刀轴可以自动降速,调节进给速度,实现自适应控制,减小由逼近误差引起的插补误差。

2.4　自动加速控制

根据加减速曲线自动调整每一个插补周期中插补直线段的长度来控制参数化插补中的自动加减速过程。在软件设计部分,需要设计一条单独的通用控制通道,能够完成自动加减速过程中的计算和轨迹规划,这样可以实现加减速的计算和控制过程与加减速曲线的形状没有关系,保证加减速曲线的自行存储。其原理图 4 所示。

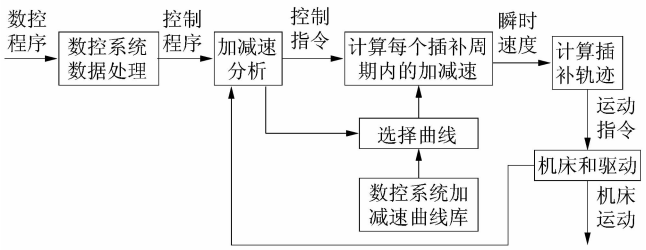

图 4　加减速曲线独立存储原理图

Figure 4　Independent storage schematic diagram acceleration and deceleration curve

假设给定的加速曲线如图 5 所示,其中的 f_d 是自动加速过程中进给速度总的变化量,也叫标样速度差;t_d 是从初始速度加速到标样速度所需要的时长,简称标样加速时间。

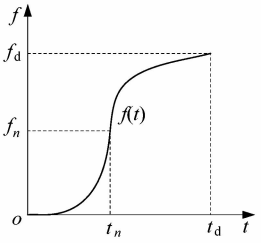

图 5　自动加速过程曲线

Figure 5　Automatic acceleration process curve

通过自动加速控制得到每一插补阶段的进给速度的具体步骤为:

1)计算机床刀轴加速过程中预期进给速度 F_2 和初始进给速度 F_1 的差值 F_D,即 $F_D = F_2 - F_1$。将 F_D 与标样速度进行比较,计算出机床加速过程中实际速度差与标样速度差之间的比值

$$K = F_D / f_D。$$

2)查表时间 t_n 的确定。根据比值 K、插补周期个数 n 和查表时间 t_n 之间的关系确定查表时间,并根据查表时间确定标样速度变化量,然后根据进给速度实际变化量与标样速度变化量之间的比值,计算出当前进给速度实际变化量:

$$t_n = \Delta T \cdot n / K;$$
$$\Delta F_n = f_n K;$$
$$F_i = F_1 + \Delta F_n。$$

式中:ΔT 为采样周期;n 为采样周期个数;F_n 为查表时间的样板速度;ΔF_n 为实际速度改变量;F_i 为当前进给速度。

3)求实际的进给速度,并利用所求得的进给速度计算插补直线段的长度,规划刀具行进的轨迹。

2.5　自动减速控制

自动减速控制的步骤与自动加速过程类似,减速曲线如图 6 所示。

具体步骤:

1)计算机床刀轴自动减速过程前的初始进给速度 F_{11} 和减速过程结束后的预期进给速度 F_{21} 的差值 F_{D1},计算机床实际速度差与标样速度差之间的比值 K_1。

$$F_{D1} = F_{11} - F_{21};$$
$$K_1 = F_{D1} / f_{d1}。$$

2)利用减速到预期进给速度所需要的插补周期的个数 n,计算查表时间,并按照查表时间查找标样速度变化量,根据进给速度实际变化量与标样速度变化量之间的比值,从而计算出当前进给速度实际变化量。其中 F_{D1} 为加工过程速度差。

$$t_{n1} = \Delta T_1 \cdot n / K;$$
$$\Delta F_{n1} = F_{D1} - f_{n1} K;$$
$$F_{i1} = F_{11} - \Delta F_{n1}。$$

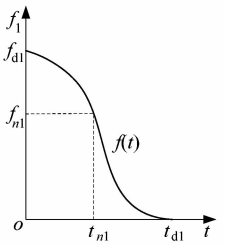

图 6　自动减速曲线

Figure 6　Automatic deceleration curve

3）按照当前周期的实际进给速度求取当前插补周期的插补直线段的长度，并据此来规划插补轨迹。相对于加速过程而言，减速过程还需要预测减速点，方便确定从哪一时间点开始进行减速运动。可以通过下述公式计算减速距离 s，从而规划减速点的位置。

$$s = \left(\frac{F_{11} - F_{21}}{f_{d1}}\right)^2 s_{d1} + \frac{F_{11} - F_{21}}{f_{d1}} t_{d1} F_{21}。$$

式中：F_{11}，F_{21} 分别代表减速过程的初始进给速度和结束时的预期进给速度；f_{d1} 为减速过程中的标样速度差；t_{d1} 为标样减速时间；s_{d1} 为标样减速距离，可通过公式 $s_{d1} = \int_0^{t_{d1}} f(t)\,\mathrm{d}t \approx \sum_{i1=1}^{m} f_{i1}\Delta t$ 近似求取，其中 f_{i1} 为标样曲线离散取值；m 为离散点总数；Δt 为时间增量。

2.6 插补预处理

为保证插补进程顺利，必须进行插补预处理来为实时插补做准备，而参数化插补过程中的预处理的目的是按照输入数控系统的信息值求取相关的待定系数和参变量的取值，为插补中点的判定提供依据。

3 实验案例

实验之前先进行设备的检查与调试，空转电机，观察电机的运转是否平滑过渡，是否有额外的噪音。实验时，首先给实验平台步进电机供电，然后启动各个坐标轴的驱动器，最后连接好计算机并行端口就可以进行模拟试验了。观察仿真系统的显示结果是否正常连续，与原图之间的误差是否控制在允许范围内。图 7 所示是进行系统仿真之后的结果，仿真结果与源代码相符合。

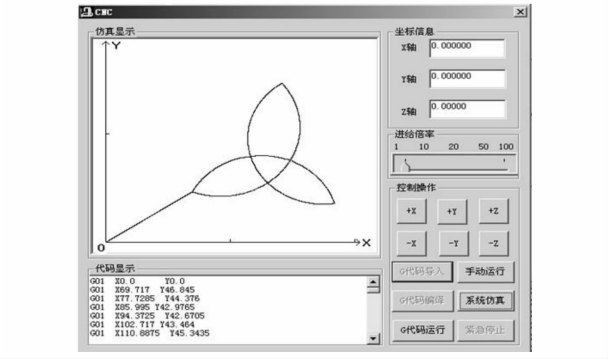

图 7 仿真系统仿真结果

Figure 7 Simulation results of simulation system

Windows XP SP3 + RTX 软件系统通过并行端口将控制信号输出到步进电机驱动器上，由驱动器开环的操控步进电机的运动，得到图 8 所示的实验结果，试验运行过程中无较大的噪声，曲线过渡平稳，仿真与实验均吻合于 G 代码。

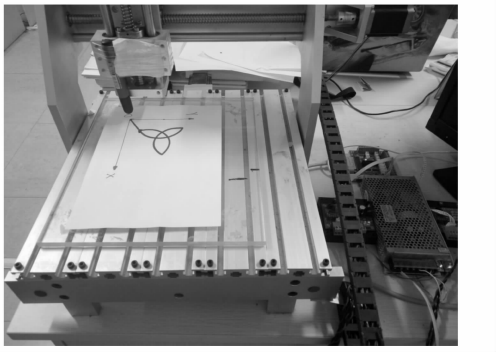

图 8 实验结果

Figure 8 Experimental results

将不同插补方式产生的插补线段产生的误差值比较如图 9 所示。可以看出，参数化插补算法能够通过控制插补直线段的长度很好地控制插补误差的大小，插补误差在许可的范围内，并且，插值结果具有较好的连续性。其他插补算法，插补误差较大，插补结果不光滑。

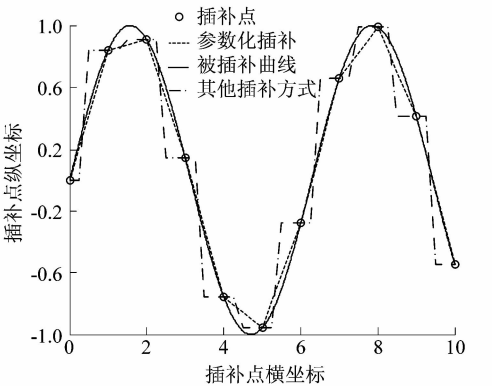

图 9 误差对比图

Figure 9 Error contrast diagram

4 结语

为了提高对于曲率变化较大的曲线的插补精度，笔者应用基于 Windows XP + RTX 为原型的数控软件，设计了参数化自动加减速插补算法，计算最佳的插补直线段的长度，严格控制插补过程中的弓高误差，利用参数化算法降低了整个计算过程的难度。与常规增量式递推算法相比，消除了累积误差，提高了插补速度和精度，大大提高了机床的加工效率和加工精度，满足高速高精度加工的工业要求，具有较高的实用性。实验中，在拐角过渡处机床震动略大的问题尚未得到很好地解决，需要进一步优化算法，改进拐角处机床的过渡的问题。

（下转第 29 页）