

三轴数控侧铣空间刀具半径补偿算法*

胡自化^{1,2} 张平² 杨冬香¹ 徐宏¹

(1. 湘潭大学机械工程学院 湘潭 411105;

2. 湘潭大学土木工程与力学学院 湘潭 411105)

摘要: 通过深入分析三轴数控侧铣加工的特点, 利用几何平面投影原理, 采用平面轮廓刀具半径补偿算法推导的类似方法, 建立三轴数控侧铣加工的空间刀具半径补偿算法。将三轴数控侧铣加工的空间刀具半径补偿归纳为三种转接过渡类型处理, 通过加工平面的投影相交求出空间两相邻程序段的过渡转接点坐标分量 x 、 y , 然后通过逆投影计算转接点的第三分量 z , 从而推导出各类型转接点的坐标计算公式。基于 UG 平台编写了三轴数控侧铣刀具补偿仿真软件, 模拟和铣削加工实例结果表明所建立的三轴数控侧铣加工空间刀具半径补偿算法正确有效。基于该算法的空间刀具半径补偿 G 指令将可避免由于刀具磨损和刀具更换导致的三轴侧铣数控代码重生成。

关键词: 三轴数控侧铣 几何平面投影 空间刀具半径补偿 数控加工

中图分类号: TH164

0 前言

一般来说, 侧铣加工是数控铣削加工的一典型工艺。为了简化数控编程以及提高编程效率和质量, 目前大部分数控铣削系统具有一些快捷的编程功能, 如刀具半径和长度补偿指令、复合循环指令、宏指令以及参数化编程、子程序调用等。然而, 传统数控铣削系统的刀具半径补偿指令仅对平面加工轮廓两轴数控侧铣有效, 而对三轴数控侧铣无效。因此三轴数控侧铣编程时, 首先需将零件轮廓沿轮廓法矢方向偏置一刀具半径距离, 偏置结果就是用描述刀具路径的刀位数据, 然后根据刀位数据生成数控代码。当刀具磨损或更换时, 三轴侧铣数控代码需要重新生成, 计算机与数控装置之间需通过串口或 Internet/Intranet 网络不断进行数据通信。尽管在二维刀具半径补偿算法方面已有很多研究^[1-2], 也建立了二维刀具补偿 G 代码, 但由于缺乏空间刀具补偿的 G 代码, 三轴数控侧铣编程仍使用较为低下的方式, 数控编程工作十分繁琐。刀具一旦磨损或换刀, 就需要将零件轮廓线沿其在 Oxy 平面的投影曲线的法矢方向偏置一刀具半径距离以获得新的刀位数据, 数控代码也就应根据新的刀位数据重新生成, 这种传统 CAD 编程方法存在许多弊端。为了彻底改变这种繁琐数控编程, 就必须建立三维刀具半径补偿算法, 尤其是随着高速切削高新技术迅猛发展, 新的加工理念要求快速生成刀具路径, 即需要数控系统在多轴加工中能够实现空间刀具自动

补偿功能^[3]。但零件轮廓的复杂性以及需考虑许多复杂的转接过渡, 使得空间刀具半径补偿算法的建立举步维艰, 对空间刀具半径补偿方面研究的文献很少^[4-7]。刘雄伟等^[4]和刘德福等^[5]仅从理论角度考虑了刀轴平行于曲面法矢的五轴加工偏摆问题计算, 离工程实际应用还有较大差距。MORETON 等^[7]仅研究了三轴车削中心的三维刀具补偿近似算法, 得出了一些有价值的结论, 但至今尚未取得能有效指导实际生产的三维刀具半径补偿理论。

由于刀轴矢量(0 0 1)在三轴加工中始终保持不变, 刀轨曲线与零件轮廓曲线在加工平面的投影互为等距, 因此本文拟通过深入分析三轴数控侧铣加工的特点, 利用几何平面投影原理, 采用平面轮廓刀具半径补偿建立的类似方法, 建立三轴数控侧铣加工的空间刀具半径补偿算法。该算法不仅可突破刀具补偿功能限于平面轮廓加工的理论瓶颈, 而且是刀具补偿向空间轮廓面多轴侧铣加工的有益尝试, 尤其在刀具磨损和刀具更换时可避免数控程序的重新生成, 具有重要的理论研究价值和实际指导意义, 本文将阐述算法的具体建立和推导过程。

1 三轴侧铣空间刀具半径补偿原理

两轴侧铣加工的刀具半径补偿原理已经成为数控铣削系统的一个典型特征, 一般来说平面轮廓插补的 G 代码仅包含平面线性插补指令(G01)和加工平面的顺/逆圆弧插补指令(G02/G03), 因而两轴数控侧铣的刀具半径补偿原理就包含两种简单轮廓类型即平面直线和平面圆弧, 以及三种转接类型即缩短型、伸长型和插入型^[8]。两轴数控侧铣加工的刀具半径补偿算法计算流程如图 1 所示。

* 国家自然科学基金(10372087)、湖南省自然科学基金(04JJ3043)和湖南省教育厅科研基金(06B094)资助项目。20060718 收到初稿, 20070129 收到修改稿

开始							
从基于零件轮廓编程获得的数控代码文件中读出两相邻程序段的坐标和几何类型(直线段或圆弧段)							
计算两相邻程序段(直线段或圆弧段)方向矢量之间的转角,方向为第一段矢量以逆时针方向转向第二段矢量							
判断刀具偏置方向							
刀具左偏				刀具右偏			
判断转角所处象限				判断转角所处象限			
第一、二象限	第三象限	第四象限	转角为 180°	第一象限	第二象限	第三、四象限	转角为 180°
缩短型	插入型	伸长型	两旋向相同的圆弧之间的转接看成缩短型或插入型	伸长型	插入型	缩短型	两旋向相同的圆弧之间的转接看成插入型或缩短型
根据直线与直线、圆弧与圆弧、直线与圆弧、圆弧与直线的各转接类型公式计算转接点的坐标							
得到刀心轨迹数据							
读下一段的几何信息, 循环							
循环直到程序结束							
结束							

图 1 两轴数控侧铣加工的刀具半径补偿算法计算流程

目前 G01/G1 指令是实现三轴以上联动的唯一插补指令, 对三轴数控侧铣空间轮廓, 其刀轨规划结果为若干微小直线段, 即被加工件的轮廓由许多微小直线段来逼近, 因此三轴数控侧铣的刀具半径补偿算法建立就只需考虑轮廓线为直线的情况, 同时意味着转接过渡只需讨论两相邻直线段的情形。此外, 由于三轴侧铣加工中刀轴方向总是与主轴方向平行(z 轴), 因此通过几何平面投影, 三轴侧铣加工的刀具半径补偿就简化为两轴数控侧铣的情形。图 2 表示两轴和三轴数控侧铣加工中刀具半径补偿关系。

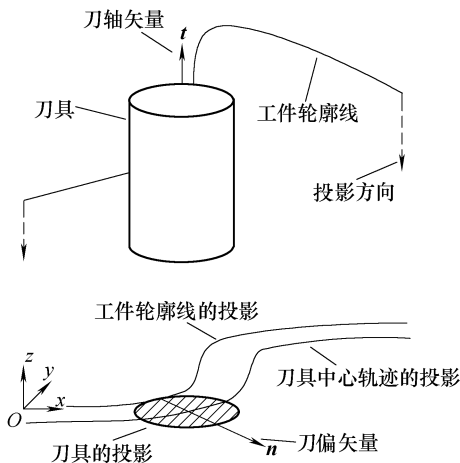


图 2 两轴和三轴数控侧铣加工中刀具半径补偿关系

这里, 三轴数控侧铣的刀轴矢量记为 $t = [0 \ 0 \ 1]^T$, 而刀偏矢量即轮廓法矢总垂直于 z 轴, 记为 $n = [x_n \ y_n \ z_n]^T$, 根据几何平面投影原理, 矢量 n 的计算公式可表示如下

$$n = \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} = \begin{bmatrix} \pm \frac{y_1 - y_2}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \\ \mp \frac{x_2 - x_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \\ 0 \end{bmatrix} \quad (1)$$

式中 x_1, y_1 —— 零件轮廓直线段的起点坐标

x_2, y_2 —— 零件轮廓直线段的终点坐标

因此, 刀具中心坐标 P_1 为

$$P_1 = P_0 + r n \quad (2)$$

式中 P_0 —— 刀具与零件轮廓的接触点, 也是刀偏矢量的始端点

r —— 刀具半径

事实上, 侧铣加工要求相邻线段必须平滑过渡, 因此仅考虑单个空间直线段的半径偏置是很不够的, 分析和解决空间相邻直线段之间的复杂转接问题是成功建立空间刀具半径补偿算法的关键和核心内容。下一节将详细推导三轴数控侧铣加工中空间相邻直线段的过渡转接计算公式。

2 过渡转接公式建立

在两轴和三轴数控侧铣中, 容易发现, 当偏置零件轮廓至刀具位置时, 刀偏矢量的 z 坐标分量总是为 0, 这说明三轴数控侧铣的刀偏矢量和两轴侧铣的刀偏矢量位于 Oxy 同一平面内。因此, 基于这一共同特征, 通过几何平面投影, 三轴数控侧铣的空间刀具半径补偿的转接处理可简化为平面侧铣的刀具半径补偿的转接处理, 于是相邻空间直线段之间的转接可划分为缩短型、伸长型和插入型三种类型。

2.1 缩短型

所谓缩短型指刀心轨迹的总长度比零件轮廓上对应的两相邻直线段的总长度短。如图 3 所示, r 是刀具半径, 零件轮廓的相邻直线段为 AB 、 BC , 相应的刀轨为 DE 、 FG , 于是需建立一条连接 DE 、 FG 的直线段。很明显 DE 、 FG 在平面 Oxy 内的投影线相交于点 $E'(F')$ 。因此只需计算出点 $E'(F')$ 的坐标, 然后将其分别投影至刀轨 DE 、 FG 上得对应点 E 、 F , 从而整个刀具轨迹为 DE 、 EF 和 FG 。点 E 和 F 的坐标推导过程如下所述。



图3 缩短型相邻刀轨

已知点 A 、 B 、 C 的坐标分别为 (x_A, y_A, z_A) 、 (x_B, y_B, z_B) 、 (x_C, y_C, z_C) ，则直线 AB 、 BC 的方向矢量分别为 $\{n_x, n_y, n_z\}$ 、 $\{m_x, m_y, m_z\}$ 。

$$\begin{cases} n_x = x_B - x_A \\ n_y = y_B - y_A \\ n_z = z_B - z_A \end{cases} \quad (3)$$

$$\begin{cases} m_x = x_C - x_B \\ m_y = y_C - y_B \\ m_z = z_C - z_B \end{cases} \quad (4)$$

根据式(1)可分别得到直线 AB 、 BC 的单位刀偏矢量或单位法矢量 $\{a_x, a_y, a_z\}$ 、 $\{b_x, b_y, b_z\}$ 。

根据式(2)可得点 D 的坐标 (x_D, y_D, z_D) 为

$$\begin{cases} x_D = x_A + a_x r \\ y_D = y_A + a_y r \\ z_D = z_A \end{cases} \quad (5)$$

点 H 的坐标 (x_H, y_H, z_H) 为

$$\begin{cases} x_H = x_B + b_x r \\ y_H = y_B + b_y r \\ z_H = z_B \end{cases} \quad (6)$$

同时可分别建立 DE 、 FG 的直线方程如下

$$\frac{x - x_D}{n_x} = \frac{y - y_D}{n_y} = \frac{z - z_D}{n_z} \quad (7)$$

$$\frac{x - x_H}{m_x} = \frac{y - y_H}{m_y} = \frac{z - z_H}{m_z} \quad (8)$$

如图3所示， $D'E'$ 和 $F'G'$ 分别为 DE 和 FG 在平面 Oxy 上的投影，于是直线 $D'E'$ 和 $F'G'$ 的方程如下所示

$$\frac{x - x_D}{n_x} = \frac{y - y_D}{n_y} \quad (9)$$

$$\frac{x - x_H}{m_x} = \frac{y - y_H}{m_y} \quad (10)$$

事实上，直线 $D'E'$ 和 $F'G'$ 在 Oxy 平面相交于点 $E'(F')$ ，因此通过式(9)、(10)可以得出点 $E'(x'_E, y'_E)$ 、 $F'(x'_F, y'_F)$ 的坐标值，点 E 、 E' 的 x 、 y 坐标分量分别

相等，同样点 F 、 F' 也是如此。因此将 x_E 或 y_E 分别代入式(7)中的 x 或 y ，可以得到点 E 的 z 坐标值 z_E 。同样地，用 x_F 或 y_F 分别代入式(8)中的 x 或 y ，可以得到点 F 的 z 坐标值 z_F 。所得结果如下

$$\begin{cases} x_E = x'_E = \frac{y_D n_x m_x - y_H n_x m_x + x_H n_x m_y - x_D n_y m_x}{m_y n_x - m_x n_y} \\ y_E = y'_E = \frac{y_D n_x m_y - y_H n_y m_x + x_H n_y m_y - x_D n_y m_y}{m_y n_x - m_x n_y} \\ z_E = \frac{n_z (x_E - x_D)}{n_x} + z_D \end{cases} \quad (11)$$

$$\begin{cases} x_F = x'_F = \frac{y_D n_x m_x - y_H n_x m_x + x_H n_x m_y - x_D n_y m_x}{m_y n_x - m_x n_y} \\ y_F = y'_F = \frac{y_D n_x m_y - y_H n_y m_x + x_H n_y m_y - x_D n_y m_y}{m_y n_x - m_x n_y} \\ z_F = \frac{m_z (x_F - x_H)}{m_x} + z_H \end{cases} \quad (12)$$

至此已分别计算出了点 D 、 E 、 F 和 G 的坐标，用空间直线段连接这些点就得到整个刀具轨迹 DE 、 EF 和 FG 。由于逼近零件轮廓的直线段很短，因此线段 EF 同样也很短。在这种情况下，就可以取线段 EF 的中点 K ，直接用直线段连接点 D 、 K 和 G ，则 DK 和 KG 就可当作是一完整的刀具轨迹。

2.2 伸长型

伸长型指刀心轨迹的总长度比零件轮廓上对应的两相邻直线段的总长度长。如图4所示，刀具半径为 r ，两编程空间直线段为 AB 、 BC ，刀具轨迹为 DE 、 FG 。与缩短型类似，从图4可以看出， DE 与 FG 不相交，所以需构造一条连接线段。很明显 DE 、 FG 在 Oxy 平面上的投影相互交于点 $E'(F')$ ，所以只需计算出点 $E'(F')$ 的坐标，然后将其分别映射至刀具轨迹 DE 和 FG 上，计算出相应的映射点 E 、 F 的坐标，就可以得出整个刀具轨迹 DE 、 EF 和 FG 。

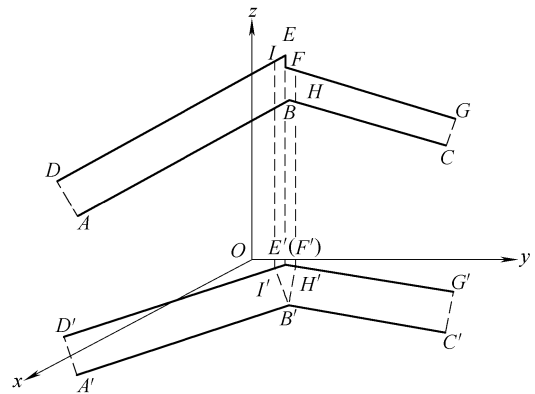


图4 伸长型相邻刀轨

显然，伸长型与缩短型在转接处理上情况非常

类似，所以点 D 、 E 、 F 、 G 的坐标推导与缩短型一样，当然所得的公式与缩短型的公式也完全相同。

2.3 插入型

插入型如图 5 所示，与二维情况一样，当两编程空间直线段在 Oxy 平面上的投影之夹角 $\theta < 90^\circ$ 时，为了提高加工效率，可在刀具轨迹 DE 、 FG 之间插入一条直线段，否则，如图 6 所示，刀具轨迹的长度将远远大于编程轨迹的长度。在传统二维刀具半径补偿情况中，插入直线段常采用如图 6 所示方法来获得。显然，在这种方法中，如图 7 所示，当 $\theta \ll 90^\circ$ 时，刀轨长度将同样远远大于编程轨迹的长度，而当 θ 接近于 90° 时，刀具与工件将发生干涉。为了避免这种缺陷，作者引入另一种处理方

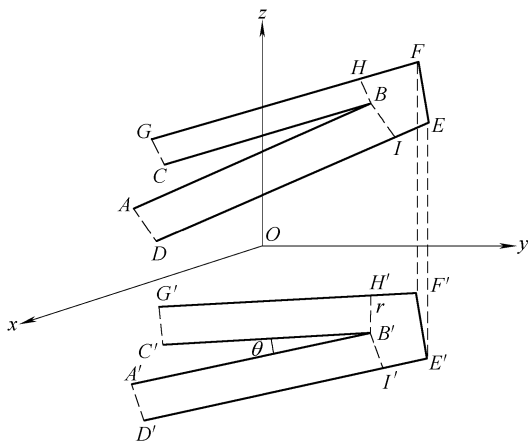


图 5 插入型相邻刀轨

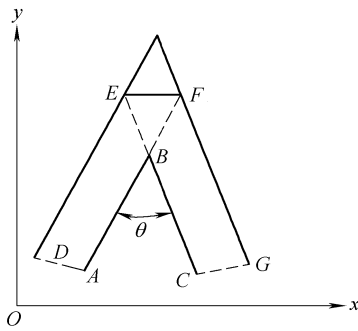
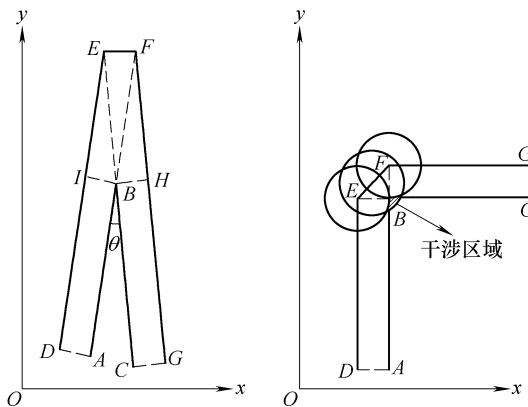


图 6 传统的插入型相邻刀轨



(a) $\theta \leq 90^\circ$

(b) $\theta \geq 90^\circ$

图 7 传统插入型的两种极限情况

法，如图 5 所示， E' 是点 I' 沿 $D'E'$ 方向偏置 r 后所得的点， F' 是点 H' 沿 $F'G'$ 的反方向偏置 r 后得到的点。只须计算出点 E' 、 F' 的坐标，然后将它们分别映射至刀具轨迹 DE 、 FG 上，计算出对应点 E 、 F 的坐标，就可以得到整个刀具轨迹 DE 、 EF 和 FG 。点 E 和 F 的坐标推导如下。

和缩短型情况一样，已知点 A 、 B 和 C 的坐标分别为 (x_A, y_A, z_A) 、 (x_B, y_B, z_B) 、 (x_C, y_C, z_C) ，从而直线 AB 、 BC 的方向矢量分别为 $\{n_x, n_y, n_z\}$ 、 $\{m_x, m_y, m_z\}$ ，直线 AB 、 BC 的单位刀偏矢量或单位法矢量分别为 $\{a_x, a_y, a_z\}$ 、 $\{b_x, b_y, b_z\}$ 。

根据式(2)，点 D 的坐标 (x_D, y_D, z_D) 为

$$\begin{cases} x_D = x_A + a_x r \\ y_D = y_A + a_y r \\ z_D = z_A \end{cases} \quad (13)$$

点 I 的坐标 (x_I, y_I, z_I) 为

$$\begin{cases} x_I = x_B + a_x r \\ y_I = y_B + a_y r \\ z_I = z_B \end{cases} \quad (14)$$

点 H 的坐标 (x_H, y_H, z_H) 为

$$\begin{cases} x_H = x_B + b_x r \\ y_H = y_B + b_y r \\ z_H = z_B \end{cases} \quad (15)$$

至此可以建立 DE 、 FG 的直线方程分别如下

$$\frac{x - x_I}{n_x} = \frac{y - y_I}{n_y} = \frac{z - z_I}{n_z} \quad (16)$$

$$\frac{x - x_H}{m_x} = \frac{y - y_H}{m_y} = \frac{z - z_H}{m_z} \quad (17)$$

因为点 E' 是点 I' 沿 $D'I'$ 方向偏置 r 后所得的点，而 $D'E'$ 的方向矢量与 $A'B'$ 的方向矢量相同，点 E 与点 E' 的 x 、 y 坐标分量分别相等，从而可以计算出点 E' 的坐标如下

$$\begin{cases} x'_E = x_I + \frac{n_x}{\sqrt{n_x^2 + n_y^2}} r \\ y'_E = y_I + \frac{n_y}{\sqrt{n_x^2 + n_y^2}} r \end{cases} \quad (18)$$

与点 E' 类似，点 F' 的坐标为

$$\begin{cases} x'_F = x_H - \frac{m_x}{\sqrt{m_x^2 + m_y^2}} r \\ y'_F = y_H - \frac{m_y}{\sqrt{m_x^2 + m_y^2}} r \end{cases} \quad (19)$$

因为点 E 、 E' 的 x 、 y 坐标分量分别相等，点 F 、 F' 的 x 、 y 坐标分量也分别相等，所以将 x_E 或 y_E 代入 DE 的方程式(14)，可得点 E 的 z 坐标分量。同样，

将 x_F 或 y_F 代入 FG 的方程式(15), 可得点 F 的 z 坐标分量。所得结果如下

$$\begin{cases} x_E = x'_E = x_I + \frac{n_x}{\sqrt{n_x^2 + n_y^2}} r \\ y_E = y'_E = y_I + \frac{n_y}{\sqrt{n_x^2 + n_y^2}} r \\ z_E = z_I + \frac{n_z}{\sqrt{n_x^2 + n_y^2}} r \end{cases} \quad (20)$$

$$\begin{cases} x_F = x'_F = x_H - \frac{m_x}{\sqrt{m_x^2 + m_y^2}} r \\ y_F = y'_F = y_H - \frac{m_y}{\sqrt{m_x^2 + m_y^2}} r \\ z_F = z_H - \frac{m_z}{\sqrt{m_x^2 + m_y^2}} r \end{cases} \quad (21)$$

至此已经求得点 D 、 E 、 F 和 G 的坐标, 分别用空间直线段顺序连接这些点就可得到整个刀具轨迹 DE 、 EF 和 FG 。

3 编程、仿真与切削试验

根据以上所建立推导的过渡转接公式, 三轴数控侧铣加工的刀具半径补偿编程指令格式可定义如下

$$\left. \begin{matrix} G41/G42 \\ G40 \end{matrix} \right\} G01/G00 \quad X_Y_Z_D_$$

其中指令 $G41$ 、 $G42$ 分别表明从编程线段在 Oxy 平面上的投影来看刀具为左偏或右偏, 与传统中的一样, 指令 $G40$ 为取消刀补指令, X 、 Y 、 Z 为存放终点坐标分量的地址, 下划线表示终点的坐标分量值, D 为指定刀具半径补偿值的存储器地址, 其中三轴数控侧铣刀具偏置方向定义如图 8 所示。

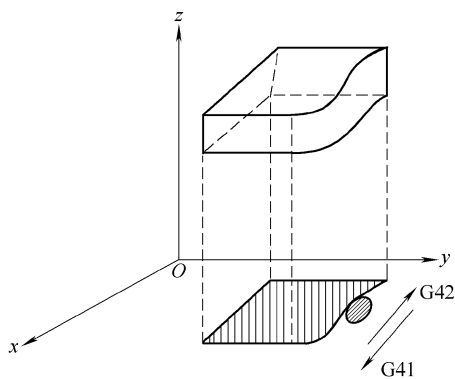


图 8 三轴数控侧铣中 $G41/G42$ 的定义

从工件轮廓线在 Oxy 面上的投影看, 沿着当前

刀具走刀的前进方向观察, 若刀具位于工件的左侧则为左偏置($G41$); 否则如刀具位于工件的右侧则为右偏置($G42$)。与传统的编程格式相比, 这里所定义的格式包含了 z 坐标分量, 同时也不需要受加工平面选择指令 $G17/G18/G19$ 的限制。仿效传统二维半径补偿图表, 本文所建立的三轴数控侧铣加工的刀具半径补偿算法计算流程可归纳如图 9 所示。

开始					
从基于零件轮廓编程获得的数控代码文件中读出两相邻程序段的坐标和几何类型(均为空间直线段)					
计算两相邻程序段(均为空间直线段)方向矢量之间的转角, 方向为第一段矢量以逆时针方向转向第二段矢量					
判断刀具偏置方向					
刀具左偏			刀具右偏		
判断转角所处象限			判断转角所处象限		
第一、二象限	第三象限	第四象限	第一象限	第二象限	第三、四象限
缩短型	插入型	伸长型	伸长型	插入型	缩短型
根据直线与直线转接类型公式(1)~(21)计算转接点的坐标					
获得刀心轨迹数据					
读下一段的几何信息, 循环					
循环直到程序结束					
结束					

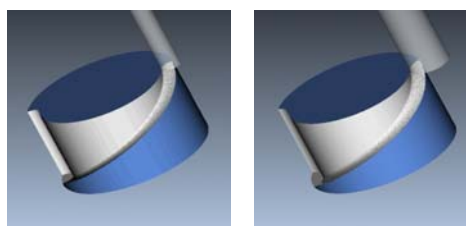
图 9 三轴数控侧铣加工的刀具半径补偿算法计算流程

为了验证三轴数控侧铣加工的刀具半径补偿算法, 首先基于 $UGII/Open Grip$ 对所建立的算法和公式进行了模拟验证。如图 10 所示实例, 为使用相同毛坯, 用半径不同的刀具来模拟三轴数控侧铣的加工过程。侧铣工件轮廓线为一条空间螺旋线, 其曲线方程定义如下

$$\begin{cases} x = 30 \cos \omega t \\ y = 30 \sin \omega t \\ z = 40t \end{cases}$$

式中 t —— 参变量, $t=0\sim 1$

图 10 为不同刀具半径的模拟结果, 表明不同刀具半径的实体加工模拟结果非常吻合。



(a) $r=5$ mm (b) $r=10$ mm

图 10 不同刀具半径模拟结果

图 11 所示为空间自由曲线的三轴数控侧铣建立刀具半径补偿后的实体加工模拟结果,其刀具半径补偿转接包括伸长型、缩短型和插入型三种转接类型。该实例充分说明所建立的三种转接算法是正确有效的。

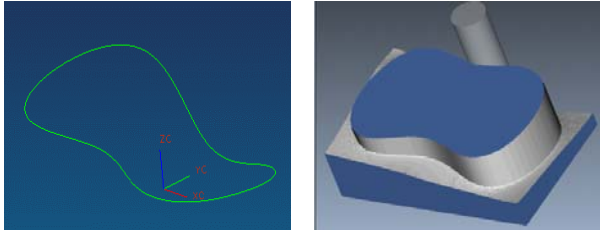
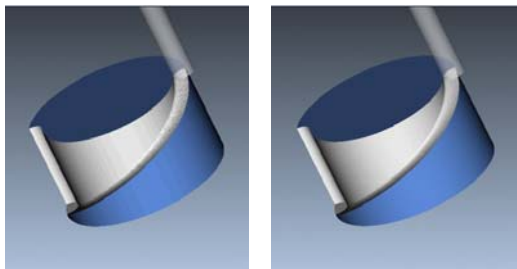


图 11 空间自由曲线模拟结果

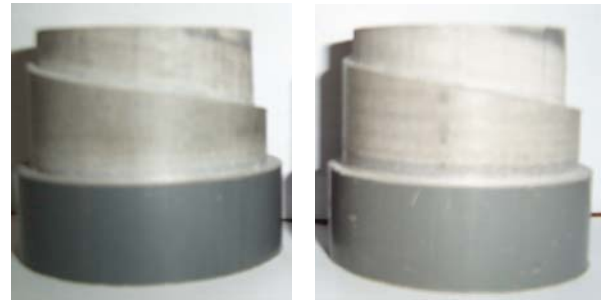
对于缩短型和插入型,转接段 EF 插在相邻两编程线段之间,如图 3、4 所示。当零件轮廓为曲线时,线段 EF 很短,所以对这种情况还提出另一种转接方法:取线段 EF 的中点 K ,然后用直线段连接点 D 、 K 和 G 。 DK 和 KG 就可以近似看成整个刀心轨迹。为了进行比较,采用相同毛坯和相同刀具来模拟两种过渡处理方法的加工过程(图 12)。零件的轮廓仍然为一条螺旋线,图 12 a 是当刀轨为 DE 、 EF 和 FG 时的模拟结果,而图 12 b 是当刀轨为 DK 、 KG 和 KG 时的模拟结果。在仿真过程中两种方法结果非常相似,加工效率却显著不同。所提出的简化方法适合于微米级以下编程直线段,这样编程线段的数量可以大幅度减少,从而可明显提高加工效率。



(a) 刀轨为 DE 、 EF 和 FG (b) 刀轨为 DK 和 KG

图 12 两种过渡处理方法的比较

为了进一步验证所建立的三轴数控侧铣加工的刀具半径补偿算法的正确性,作者为此开发研制了具有三轴侧铣空间刀补功能的开放式数控原形机床,对螺旋面进行了半径补偿切削加工试验。试验实例仍为一空间螺旋线,其参数 $r=20\text{ mm}$,螺距 $h=20\text{ mm}$,切削加工试验结果如图 13 所示。测量结果表明,用半径为 3 mm 与 6 mm 的刀具加工的零件螺旋面尺寸非常接近,相对误差在微米级以下。切削试验进一步验证了所提出的三轴数控侧铣空间刀补算法在实际生产应用可获很好的尺寸与形状的一致性,说明该算法是正确、可靠和有效的。



(a) $r=3\text{ mm}$ (b) $r=6\text{ mm}$

图 13 切削加工实例试验结果

4 结论

(1) 提出了基于几何平面投影原理的空间刀具半径补偿有效算法,即将空间刀具半径补偿建立定义为三种转接过渡类型处理,利用加工平面首先求出空间两相邻程序段的过渡转接点的坐标分量 x 、 y ,然后通过逆投影计算转接点的第三分量 z ,从而推导出各类型转接点的坐标计算公式;同时确定了刀偏类型的判断准则。为填补三轴数控侧铣刀具半径补偿功能奠定了扎实理论基础。

(2) 基于 UG 软件编写了三轴数控侧铣刀具补偿仿真程序,研制了具有三轴侧铣空间刀补功能的开放式数控原形机床等试验装置,对所建立的空间刀具补偿算法进行了实例充分验证。仿真和加工试验结果表明所建立的三轴数控侧铣加工空间刀具半径补偿算法是正确有效的。

(3) 定义了刀具半径补偿编程指令通用格式。该格式既适用于两轴侧铣又适用于三轴数控侧铣,与传统的二维轮廓偏置指令完全一致,从而保证了数控铣削刀具半径补偿指令的真正有效统一。

参 考 文 献

- [1] 朱世春,刘学平,刘光复. 微机数控系统刀具半径补偿的矢量分析和处理[J]. 中国机械工程, 1995, 6 (增刊): 83-86.
- [2] 史旭明,赵万生,狄士春. 数控刀具半径补偿研究[J]. 组合机床与自动化加工技术, 1998(11): 15-19.
- [3] 李沪曾,狄伟,诸乃雄,等. 高速切削加工的计算机数控[J]. 机床与液压, 2002(4): 65-67.
- [4] 刘雄伟,袁哲俊,刘华明. 数控机床三维刀具半径补偿方法探讨[J]. 机床, 1991(8): 48-49.
- [5] 刘德福,陈峰. 计算机数控系统三维刀具半径补偿研究[J]. 组合机床与自动化加工技术, 2001(8): 9-11.
- [6] HU Zihua, QI Rui, ZHANG Ping, et al. An effective approximate tool radius compensation method for 3-axis NC machining free surface[C]//Proc. of the 11th International

- Manufacturing Conference in China, Jinan, 2004. Jinan: Shandong University Press, 2004: 118-119.
- [7] MORETON D, DURNFORD R. Three-dimensional tool compensation for a three-axis turning centre[J]. *Int. J. Adv. Manuf. Technol.*, 1999, 15: 649-654.
- [8] 刘雄伟. 数控加工理论和编程技术[M]. 北京: 机械工业出版社, 2000.

TOOL RADIUS COMPENSATION ALGORITHM FOR THREE-AXIS NC PERIPHERAL MILLING

HU Zihua^{1,2} ZHANG Ping²
YANG Dongxiang¹ XU Hong¹

(1. School of Mechanical Engineering, Xiangtan
University, Xiangtan 411105;

2. Civil Engineering & Mechanics College, Xiangtan
University, Xiangtan 411105)

Abstract: By thoroughly analyzing the characteristic of three-axis NC peripheral milling, based on geometry planar projection, a tool radius compensation algorithm for three-axis NC peripheral milling is established by using the similar method to that for planar contours NC milling. The space tool

radius compensation for three-axis NC peripheral milling is reduced to three types of transitions processing. The first work is to calculate the coordinates (x and y) dimension of the transitions between neighbor linear segments in space by the projections intersection on the work plane, and the next work is to obtain the third coordinate(z) dimension of the transitions by reverse projection, consequently the formulas to calculate all sorts of transitions coordinates are derived. Based on UG platform, the simulation software of tool compensation for three-axis NC peripheral milling is programmed, the results of the simulation and NC milling examples show that the proposed space tool radius compensation algorithm for three-axis NC peripheral milling is highly effective and correct. The space tool radius offset G instructions with the developed approach will avoid NC codes regeneration due to tool wear or tool change during the three-axis NC flank milling process.

Key words: Three-axis NC peripheral milling
Geometry planar projection
Space tool radius compensation
NC machining

作者简介: 胡自化, 男, 1967 年出生, 博士, 副教授。主要研究方向为基于数控技术的柔性制造自动化加工技术。

E-mail: iamtxtu@163.com