



非圆曲线的宏程序数控铣削编制

江南大学机械工程学院 (214063) 俞 盛

随着计算机辅助设计的广泛运用,在模具加工及凸轮加工过程中常遇到以数学方程描述的有规律的过渡曲线与曲面。如图 1 是在工程实践中遇到的一种非圆曲线柱面示意图。

曲线 I: $Y = 0.0000003 \cdot X^4 + 0.0001 \cdot X^3 - 0.0152 \cdot X^2 + 0.0829 \cdot X + 36.337$
A: (65, 10.323) B: (-24, 24.309)
曲线 II: $Y = 0.0002 \cdot X^3 - 0.0174 \cdot X^2 - 0.5895 \cdot X + 46.223$
C: (24.286, 24.509) D: (65, -10.685)

在精确保证图样设计要求下,如采用计算机辅助制造软件(如 MASTERCAM、UG 等软件)编制这类零件的数控加工程序,其编制过程较长,还需要进行程序输入等操作。采用宏程序,直接将曲线的参数方程以变量、表达式的方式输入数控加工程序中,可以减少乃至免除编程时繁锁的节点数值计算,达到精简程序量,提高编程效率的目的。

1. 非圆曲线数控加工宏程序的编制及注意事项

作为现代数控系统程序编制功能重要组成之一的类似于高级语言的宏程序汇编语言功能,已经广泛运用于各种数控车床、数控铣床及加工中心。江南大学机械工程学院引入的华中 I 型数

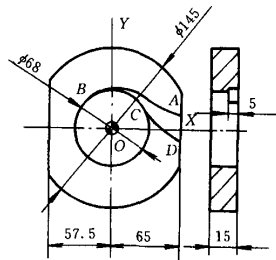


图 1

控铣床就具有宏程序编制功能。宏程序是指由多个宏指令构成并被 CNC 系统所识别的可操作程序简称。宏指令是用一个宏名称和若干个参数来代替一段程序,它分宏变量及常量,运算符与表达式,多种判别语句、循环语句等多种类型。用户可以利用宏程序进行算术运算、逻辑运算和函数混合运

算,方便地编制各种复杂的零件加工程序。宏程序编制方法与规则和我们所熟悉的汇编语言编制方法大体相同。有了宏指令,再加上必要的国际通用 ISO 数控加工代码,宏程序编制就比较简单。

在编制宏程序时,应注意考虑宏变量的种类,其主要分局部变量(#0~#49)、全局变量(#50~#99)、系统变量(如#100~#199表示刀补号100~199的补偿值),应按

表 2

$n = 100 + 10 \times \text{Unit No.}$

I/O 分类	Word 号	Bit (位)															
		15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
输出	n	0				0				0	0	内存 下传	内存 上传	错误 重置	EEP- ROM 写入	原点 调整	操作 指令
	n+1	0				0				组号 00~08: 2 位 BCD 00: 凸轮无输出							
输入	n+2	外部凸轮输出状况															
		16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	n+3	内部凸轮输出状况															
		32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
	n+4	内部凸轮输出状况															
		48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33
	n+5	0				当前凸轮角度 (000°~359° 3 位 BCD 码)											
	$\times 10^2$					$\times 10^1$				$\times 10^0$							
	n+6	完成某组动作分相器旋转的圈数															
		小数点位置数据				圈数数据 (000~999: 3 位 BCD)											
	n+7	0				0				执行组号 (00~08: 2 位 BCD)							
	n+8	状态															
	AL-M2	AL-M1	0	0	0				0	内存 调用 失败	原点 调整 完成	内存 调用 完成	分相 器 故障	错误	RUN 模式	操作	
n+9	0				0				错误代码								

(收稿日期: 20030815)

需要选择不同的宏变量进行赋值。必须指出的是表示刀补的系统变量在编制时应以所选铣刀直径 1/2 输入。

在编制循环语句格式如：

WHILE 条件表达式

.....

ENDW (有些系统使用 END)

时, WHILE、ENDW 在一个程序中必须彼此对应。必须注意的是 WHILE、ENDW 的循环格式不支持多重循环嵌套,而在数控系统编程说明书中并未加说明。

宏指令的算术运算符仅规定了加 (+)、减 (-)、乘 (×)、除 (/),对于三次方及其以上的运算,数控系统编程说明书中并未加说明。在编程中可以利用变量的连乘加以解决,如宏变量 #1 的四次方可写成: #1×#1×#1×#1。

2. 宏程序清单如下 (程序流程如图 2 所示):

```
%9999; (有些系统使用 09999) 零件程序编号;
#1=65; 赋予当前 #1 号局部变量为 65;
#2=0; 赋予当前 #2 号局部变量为 0;
#3=-24; 赋予当前 #3 号局部变量为 -24;
#101=2.5; 采用 φ5mm 的立铣刀, 赋予刀补值 #101 号半径补偿量为 2.5;
N10 G90 G92 X0 Y0 Z0; 在绝对值方式下建立工件坐标系;
N20 G01 Z2 S3821 M03 F343; 刀具运动到 Z=2;
N30 X69.36 Y0; 刀具运动到 X=69.36 Y=0;
N40 Z-5; 刀具沿 Z 轴负向切入 Z=-5;
N50 G41 D101 X69.36 Y9.274; 刀具运动到 X=69.36 Y=9.274 建立调用刀补;
WHILE #1 GE #3; 判别当 #1 ≥ #3 时执行曲线 I 的直线插补反之执行 ENDW 后段程序;
#2=0.0000003*#1*#1*#1*#1+0.0001*#1*#1*#1-0.0152*#1*#1+0.829*#1+36.337; 计算当 #1 确定时 #2 的值;
N60 G01 X[#1] Y[#2]; 刀具运动到曲线上 X=#1 Y=#2;
#1=#1-0.001; 采用等间距直线逼近 X 方向每隔 0.001 (华中 I 型数控系统所能实现的最小位移量) 进行一次数值累减;
ENDW; 曲线 I 插补结束;
N70 G01 Z2; 曲线 I 加工完成后刀具运动到 Z=2;
N80 G40 X0 Y30; 刀具运动到 X=0 Y=30 取消刀补;
#4=24.286; 赋予当前 #4 号局部变量为 24.286;
#5=0; 赋予当前 #5 号局部变量为 0;
#6=65; 赋予当前 #6 号局部变量为 65;
N90 G01 G41 D101 X0 Y30; 刀具运动到 X=0 Y=30 建立刀补;
N100 Z-5; 刀具运动到 Z=-5;
N110 X24.286 Y24.509; 刀具运动到 X=24.286 Y=
```

24.509;

WHILE #4 LE #6; 判别当 #4 ≤ #6 时进行曲线 II 直线插补反之执行 ENDW 后程序;

#5=0.0002*#4*#4*#4-0.0174*#4*#4-0.5895*#4+46.223; 计算 #4 确定时 #5 的值;

N120 G01 X[#4] Y[#5]; 刀具运动到曲线上 X=#4 Y=#5;

#4=#4+0.001; X 方向上每隔 0.001 进行一次数值累加;

ENDW; 曲线 II 插补结束;

N130 G01 Z2; 刀具运动到 Z=2;

N140 G40 X69.36 Y0; 刀具运动到 X=69.36 Y=0 取消刀补;

N150 X0 Y0; 刀具运动到 X=0 Y=0;

N160 Z0; 刀具运动到 Z=0;

N170 M30;

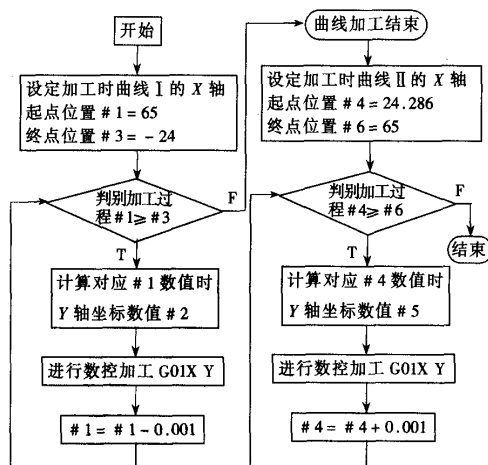


图 2 程序流程图

3. 结束语

数控加工的宏程序相对于计算机辅助制造软件而言具有不可替代的优势。以曲线 I 为例, X 轴在 [-24, 65] 区间内, 如果间隔 0.001mm 计算一次轮廓点坐标值, 传统轮廓节点编程需要超过 90 000 段数控指令, 如采用宏程序编程曲线 I, 曲线 I 全部加工完成只需要 17 段数控指令, 就可达到同等精度的数控加工。由于消除了繁琐的轮廓节点输入, 极大地精简了程序量, 程序检查方便, 大大减少了加工准备时间中程序输入、程序通信时间。此外, 只要修改刀具半径补偿量 (程序清单中以 #101 表示) 和适当修改数值累减、累加量, 就可实现粗、精加工程序互用。此宏程序清单已经运用于实际生产, 它的程序解释执行速度快, 完全满足实时加工控制要求, 且运行可靠。

(收稿日期: 20030616)