

华中数控铣/加工中心 宏程序编程实例

主 编 杨 旭

副主编 王甫忠 唐双林

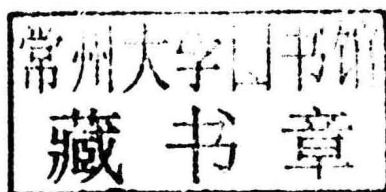
主 审 李世蓉 董祥忠



国防工业出版社
National Defense Industry Press

华中数控铣/加工中心 宏程序编程实例

主 编 杨 旭
副主编 王甫忠 唐双林
主 审 李世蓉 董祥忠



国防工业出版社

· 北京 ·

内 容 简 介

本书以“理论够用,实用性强”为宗旨,以华中数控系统为基础,详细讲述了数控铣床/加工中心宏程序的理论基础知识,内容主要包括宏程序的基础理论知识,平面的铣削、外轮廓零件的铣削、内轮廓的铣削等大量的宏程序加工实例,由浅入深、图文并茂,每句程序都进行了详细、清晰的注释说明。本书所有程序均在华中数控机床上进行校验和实际加工验证过。

本书还可以作为除华中数控系统外的其他数控系统和专业教师的参考书,也可以作为高级数控铣工、数控铣技师、数控铣高级技师、各类数控大赛的培训教材,也适用于各类中职学校、高职院校、高级技工学校、技术学院、技师学院教学教材,也可作为数控爱好者的自学教材。

图书在版编目(CIP)数据

华中数控铣/加工中心宏程序编程实例/杨旭主编. —北京:
国防工业出版社,2013.1
ISBN 978-7-118-08331-6

I. ①华... II. ①杨... III. ①数控机床—铣床—程序设计 ②数控机床加工中心—程序设计 IV. ①TG547
②TG659

中国版本图书馆 CIP 数据核字(2012)第 223188 号

※

国防工业出版社 出版发行

(北京市海淀区紫竹院南路 23 号 邮政编码 100048)

北京嘉恒彩色印刷有限责任公司

新华书店经售

*

开本 710×960 1/16 印张 10½ 字数 181 千字

2013 年 1 月第 1 版第 1 次印刷 印数 1—4000 册 定价 28.00 元

(本书如有印装错误,我社负责调换)

国防书店:(010)88540777

发行邮购:(010)88540776

发行传真:(010)88540755

发行业务:(010)88540717

前 言

随着现代科学技术的发展,数控机床在机械模具制造行业中的应用越来越普及,特别是在塑料成型模和冷冲压成形模具行业的盛行,各种自动编程软件的广泛应用,对数控机床的操作也提出了更高的要求。

目前,我国高职高专等学校目前培养的数控人才大多还是数控操作人员,数控高端技能型人才依然很紧缺,特别是编程人员。由于受各种因素的影响,学生编程过分依赖软件,只能用手工编写一些简单的基本加工程序,对一般规则的曲线、曲面知之甚少,造成学生的基本编程能力得不到应有的训练和提高。CAD/CAM 软件和手工编程都有各自的特长,且现有软件不能满足所有数控系统的特殊功能,所以充分结合这两种编程模式,对于提高编程效率和质量有着重要意义。

为了帮助在校学生和企业工程技术人员能进一步学习数控手工编程,为迈向高技能打下良好的基础,本书例题由浅入深,实用性强,图文并茂,每句程序都进行了详细、清楚的注释说明,所选例题都是常见的加工零件,与 Fanuc 系统大同小异,让读者更容易懂和接受。与此同时,书中每个例题都采用多种编程方法进行编程,这样让学生掌握多种编程思路。现今许多高职高专学校大都使用华中数控系统,市场上专门介绍华中数控宏程序的编程书籍相对很少。本系统编程语言采用国际通用 G 代码编程,具有直线、圆弧、螺旋线插补功能,支持程序的旋转、缩放、镜像、刀具补偿、宏程序、子程序调用等功能,具有小线段高速加工功能(G64)和准确定位功能(G61),S 曲线加减速控制。特别适合加工 CAD/CAM 生成的复杂模具加工程序,并且三维图形实时显示刀具轨迹和零件形状,界面实时加工参数显示,空运行和图形化程序校验功能,方便编程人员对加工代码的编制和检验,具有后台编程功能,故编写了这本书。

本书所有的程序都是经过作者亲自校验和实际加工验证过,对广大读者学习华中数控系统及相关数控系统的编程和操作人员有一定的帮助。现在宏程序业的广泛应用,使编程显得方便,它不但可以提高操作者的编程能力,还可帮助编程者更深入的了解自动编程软件的本质。

本书可作为高级数控铣工、数控铣技师、数控铣高级技师培训教材,各类数控大赛的培训教材,也适用于各类中职学校、高职院校、高级技工学校、技术学院、技师学院教学教材。也可作为数控爱好者的自学教材。

本书由杨旭主编,四川机电职业技术学院王甫忠、唐双林两位老师担任副主编。在编写过程中得到了四川机电职业技术学院李世蓉副教授的指点,同时还得到了四川大学高分子科学与工程学院硕士生导师董祥忠教授的指导与支持,在此深表感谢。

由于作者的知识水平有限,加之编写时间仓促,书中难免会出现不妥和疏漏之处,恳请同行及读者批评指正。

四川科技职业学院 先进材料与现代制造工程系:杨旭

2012年3月28日

目 录

第 1 章 绪论	1
1.1 宏程序和普通程序的对比	1
1.2 宏程序编程的特点	1
1.3 宏程序在数控系统中的运行过程	2
1.4 宏程序与 CAD/CAM 软件生成程序的加工性能对比	3
第 2 章 宏程序基础理论	4
2.1 宏变量	4
2.2 常量	11
2.3 运算符与表达式	11
2.4 语句表达式	12
2.4.1 赋值语句	12
2.4.2 条件判断语句	12
2.4.3 循环语句	13
2.5 宏程序的调用	14
2.5.1 自变量定义	14
2.5.2 自变量定义与调用	16
第 3 章 平面的宏程序铣削	19
3.1 回字形开放区域平面铣削	20
3.2 弓字形开放区域平面铣削	24
3.3 开放区域平面铣削(中心垂直下刀)	28
第 4 章 外轮廓零件的铣削加工	32
4.1 外整圆柱的加工(精加工程序)	32
4.1.1 圆的参数方程编程	32
4.1.2 圆的标准方程编程	34
4.2 外椭圆柱的加工(精加工程序)	35
4.2.1 椭圆参数方程编程	35
4.2.2 椭圆标准方程编程	37

4.3	外圆锥台的加工(从上至下粗加工)	38
4.4	外圆锥台的加工(从上至下精加工)	41
4.4.1	用直线和圆弧指令进行加工	41
4.4.2	根据圆的参数方程进行编程	42
4.5	外椭圆锥台加工(从上至下粗加工)	44
4.6	外椭圆锥台加工(从上至下精加工)	46
4.7	正棱台的加工(从上至下粗加工)	48
4.7.1	根据正切公式计算深度编程	48
4.7.2	根据切削层数编程	50
4.8	正多边形的加工	52
4.8.1	以角度作为变量	52
4.8.2	以边数作为变量	53
4.9	正多棱锥台加工	55
4.9.1	坐标旋转进行编程	55
4.9.2	斜边与X轴之间的夹角进行编程	57
4.10	四棱台的数控铣工艺设计及编程举例	59
第5章	内轮廓零件的铣削加工	62
5.1	整圆内轮廓铣削加工(中心垂直下刀)	62
5.2	整圆内轮廓精加工铣削(螺旋铣削)	64
5.3	椭圆内轮廓铣削加工(中心垂直下刀)	65
5.4	四边形内轮廓铣削加工(中心垂直下刀)	68
5.5	带圆角四边形内轮廓铣削加工(中心垂直下刀)	70
5.6	倒锥四棱台轮廓铣削加工(从下至上粗加工)	73
5.7	内椭圆倒锥台轮廓铣削加工(从下至上粗加工)	75
5.8	圆锥台孔轮廓铣削加工(从下至上粗加工)	77
5.9	正多棱台内轮廓铣削加工(从下至上加工)	79
第6章	孔、圆柱面倒圆角的铣削加工	82
6.1	外圆柱面倒90°圆角	82
6.1.1	勾股定理编程	82
6.1.2	圆的参数方程编程	83
6.2	外圆柱面倒小于90°的圆角	85
6.3	孔口倒90°圆角	87
6.3.1	勾股定理编程	87
6.3.2	圆的参数方程编程	88

6.4	孔口倒小于 90° 圆角	90
6.5	孔口倒角	91
6.6	四方体变化倒角	93
第7章	球面的粗加工与精加工	96
7.1	凸半球面的粗加工(平底立铣刀)	97
7.2	凸半球面的精加工(球头铣刀)	99
7.2.1	根据圆的参数方程编程	99
7.2.2	转化平面在 G18 或 G19 平面编程	101
7.2.3	根据勾股定理编程	102
7.3	球冠的粗加工(平面立铣刀)	103
7.4	球冠的精加工(球头铣刀)	105
7.5	椭球面的粗加工(平底立铣刀)	107
7.6	椭球面的精加工(球头铣刀)	109
7.7	凹半球面的粗加工(键槽铣刀)	112
7.8	凹半球面的精加工(球头铣刀)	113
7.8.1	根据圆的参数方程编程	113
7.8.2	根据勾股定理编程	115
7.9	内椭球面粗加工(中心垂直下刀)	116
7.10	内椭球面精加工	119
第8章	孔加工	122
8.1	圆周钻孔加工	122
8.2	带角度排空加工	124
8.3	矩阵孔加工	125
8.3.1	弓字形路线编程	125
8.3.2	锯齿形路线编程	127
第9章	圆柱孔及螺纹的铣削加工	130
9.1	概述	130
9.2	螺纹铣刀主要类型	130
9.3	螺纹铣削轨迹	132
9.4	单刀刃螺纹铣刀铣削加工内(直)螺纹	133
9.5	单刀刃螺纹铣刀铣削加工外(直)螺纹	134
第10章	实例	137
10.1	倒两个圆角	137
10.2	层孔的加工	143

10.3 椭球面的加工工艺及编程实例	145
10.4 数控大赛改编样题编程方法	147
附录 常用基本数学知识	154
附表 华中数控铣编程指令	157
参考文献	158

第 1 章 绪 论

1.1 宏程序和普通程序的对比

数控编程实际是 ISO 代码编程,即每个代码的功能都是固定的,由系统厂家开发,使用者只需按照编程规则进行编程即可。但有时候这些指令满足不了其加工要求,因此系统为用户提供了宏程序功能,它是用户对数控系统进行一定的功能扩展,实际上是对数控系统进行二次开发,这也是有局限性的。

用户宏程序和普通程序是有区别的,认识和了解这些区别后,有助于对宏程序的学习和掌握,表 1-1 为用户宏程序和普通程序的简要对比。

表 1-1 用户宏程序和普通程序的简要对比

宏 程 序	普 通 程 序
可以使用变量,并给其赋值	只能使用常量
变量间可以运算	常量之间不能运算
程序之间可以跳转	普通程序只能按顺序执行,不能跳转

1.2 宏程序编程的特点

从模块化加工的角度看,宏程序最具有模块化的思想和资质条件,编程人员只需要根据零件几何信息和不同的数学模型即可完成相应模块化加工程序的设计,应用时只需要把零件信息、加工参数等输入到相应模块的调用语句中,就能使编程人员从繁琐的、大量重复性的编程工作中解脱出来。

由于宏程序基本上包含了所有的加工信息(如所使用刀具的几何尺寸信息等),而且非常简明、直观,通过简单的存储和调用,就可以很方便地重现当时的加工状态,给周期性的生产特别是不定期的间隔式生产带来了极大的便利。

1. 高效

数控加工中常常会遇见规则曲面的零件图,也会遇到图形相近而尺寸不一样的零件图。为了提高加工效率,使程序更灵活、通用、智能,通常采用宏程序编

程,即改变前面定义变量就可以进行其加工,并且占内存少。在 CAD/CAM 软件中,无论构造规则或不规则的曲面,都有一个数学运算的过程,也必然存在着计算的误差和处理,而在对其生成三维加工刀具轨迹时,软件是根据操作者所选择的加工方式、设定的加工参数,并结合所设定的加工误差,使刀具与加工表面接触点逐点移动完成加工。从本质上看,其实就是在误差允许的范围内每条路径用直线或者圆弧去逼近曲面的过程。即便是使用 CAD/CAM 软件,对于此类零件也需要重新绘制图形,设置刀具路径、实体切削验证、执行后处理以及向机床中传输程序后才可以进行加工,其占用大量的内存空间和时间。

2. 经济

在实际生产中,经常会出现许多结构相似、种类数量少的零件,按常规加工方法,往往需用成型铣刀加工。但零件品种多时,所用成型刀量大,成型铣刀比普通铣刀费用高出 2~3 倍。为了降低成本,采用宏程序编程解决此类问题即可。许多实际加工中必须用球头铣刀加工的零件,可以用平底铣刀就可以解决了。

3. 应用范围

宏程序在实际加工中还可以应用到数控加工的其他环节,如刀具长度补偿(H)、半径补偿(D)、进给量(F)、主轴转速(S)等进行设置,大大提高了加工效率。变量编程必然会最大限度的使用数控系统内部的各种指令代码,因此数控机床在执行变量编程时,数控系统的工业计算机可以直接进行插补运算,且运算速度极快,再加上伺服电动机和机床的迅速响应,使得加工效率极高。

1.3 宏程序在数控系统中的运行过程

宏程序在数控系统中按如下步骤运行:

- (1) 首先,读取数控代码。
- (2) 提取变量和定义变量。
- (3) 预先保存在相关信息列表中。
- (4) 读取数控代码,提取复杂的表达式并进行计算。
- (5) 解释与执行代码过程中读取变量列表中保存的数值,计算表达式,并给变量赋新值。
- (6) 按照条件语句、循环语句等控制程序的下一步操作。
- (7) 按照结果输出实际数值,控制机床移动。

1.4 宏程序与 CAD/CAM 软件 生成程序的加工性能对比

任何数控加工只要能够用宏程序完整的表达,即使再复杂,其程序篇幅都比较精炼,可以说任何一个合理的、优化的宏程序极少会超过 60 行,换算成字节数至多不超过 2KB。宏程序与 CAD/CAM 软件生成程序的加工性能对比如下:

(1) 宏程序短小精悍,即使是最廉价的数控系统,其内部程序存储空间也会有 10KB 左右,完全容纳得下任何复杂的宏程序,因此根本无需考虑机床与外部电脑的传输速度对实际加工速度的影响。

(2) 为了描述复杂零件的加工,宏程序必然会最大限度地使用数控系统内部的各种指令代码,例如直线插补 G01 指令和圆弧插补 G02/G03 指令等。因此机床在执行宏程序时,数控系统的计算机可以直接进行插补运算,且运算速度极快,再加上伺服电机和机床的速度响应,使得加工效率极高。

首先,CAD/CAM 软件生成的程序通常都比较大,非常容易突破数控系统内部程序存储空间的限制,大部分程序都只能在线传输加工,显然机床与电脑之间的传输速度成为影响加工速度的第一个“瓶颈”因素。

其次,从用户使用的角度说,使用 CAD/CAM 软件来生成刀具轨迹及加工程序是很容易的事。在 CAD/CAM 软件中,无论构造规则或不规则的曲面,都有一个数学运算的过程,也必然存在着计算的误差和处理。而对其生成三维加工刀具轨迹,软件是根据操作者所选择的加工方式,设定的加工参数,并结合所设定的加工误差(或称为曲面的计算精度),使刀具以加工表面接触点逐点移动完成,其实就是在允许的误差值范围内沿每条路径用直线去逼近曲面的过程。

最后,在 CAD/CAM 软件后置上做文章有一个根本的弊端:它并没有改变、改良或优化刀具轨迹本身,只是增加了一个“二次逼近”的计算过程,必然会导致额外的误差积累,也不可能从根本上解决问题。

第2章 宏程序基础理论

HNC-21/22M 华中世纪星为用户配备了强有力的类似于高级语言的宏程序功能,用户可以使用变量进行算术运算、逻辑运算和函数混合运算。此外宏程序还提供了循环语句、分支语句(条件判断语句),这样使程序更加灵活、快捷,从而提高加工效率。

2.1 宏变量

普通加工程序只能描述一个几何形状,直接用数值指定移动距离;数控系统提供了另一种编程方式——宏编程,在程序中使用变量,通过对变量进行赋值、算术运算、逻辑运算和函数的混合运算及使用各种条件转移命令的处理方法达到程序功能,这种有变量的程序叫宏程序。

1. 变量的表示

在华中数控系统中变量用“#”和紧跟其后的变量序号来表示:

$$\#i \ (i = 1, 2, 3, \dots)$$

例如: #10, #5。

注意:变量在执行时,必须用[]把变量、公式或者其他的表达式括起来。

2. 变量的引用

将跟随在一个地址后的数值用一个变量代替,即引入了变量。例如: F[#103], 若 #103 = 150 时,则 F150; Z[-#110], 若 #110 = 600 时,则 Z-600。

引用未定义变量时,变量及地址都被忽略,例如: #20 = 0, #22 为空时, G00 X[#20] Y[#22], 其结果为 G00 X0; 当 #11 = 0, #15 = 0 时,则 G00 X[#11] Y[#15], 其结果为 G00 X0 Y0。改变引用变量的值的符号,要把负号(-)放在“#”号的前面,并且要用[]括起来,例如: Y[-#3], [-#6]。

注意:“变量值是0”与“变量的值是空”是两个完全不同的概念。

3. 变量的类型

变量分为公共变量和系统变量。公共变量,包括局部变量和全局变量,用户可以单独使用,系统作为处理资料的一部分。系统变量,用于系统内部运算时各种数据的存储。

1) 公共变量

局部变量: #0 ~ #49 是在宏程序中局部使用的变量,用于存放宏程序中的数据,断电时丢失为空。

全局变量: #50 ~ #199,它对于由主程序调用的各子程序及各宏程序来说是可以公用的,可以人工赋值,有断电为空与断电记忆两种。

2) 系统变量

#200 ~ #249	0 层局部变量
#250 ~ #299	1 层局部变量
#300 ~ #349	2 层局部变量
#350 ~ #399	3 层局部变量
#400 ~ #449	4 层局部变量
#450 ~ #499	5 层局部变量
#500 ~ #549	6 层局部变量
#550 ~ #599	7 层局部变量
#600 ~ #699	刀具长度寄存器 H0 ~ H99
#700 ~ #799	刀具半径寄存器 D0 ~ D99
#800 ~ #899	刀具寿命寄存器
#1000	机床当前位置 X
#1001	机床当前位置 Y
#1002	机床当前位置 Z
#1003	机床当前位置 A
#1004	机床当前位置 B
#1005	机床当前位置 C
#1006	机床当前位置 U
#1007	机床当前位置 V
#1008	机床当前位置 W
#1009	保留
#1010	程编机床位置 X
#1011	程编机床位置 Y
#1012	程编机床位置 Z

#1013 程编机床位置 A
#1014 程编机床位置 B
#1015 程编机床位置 C
#1016 程编机床位置 U
#1017 程编机床位置 V
#1018 程编机床位置 W
#1019 保留
#1020 程编工件位置 X
#1021 程编工件位置 Y
#1022 程编工件位置 Z
#1023 程编工件位置 A
#1024 程编工件位置 B
#1025 程编工件位置 C
#1026 程编工件位置 U
#1027 程编工件位置 V
#1028 程编工件位置 W
#1029 保留
#1030 当前工件零点 X
#1031 当前工件零点 Y
#1032 当前工件零点 Z
#1033 当前工件零点 A
#1034 当前工件零点 B
#1035 当前工件零点 C
#1036 当前工件零点 U
#1037 当前工件零点 V
#1038 当前工件零点 W
#1039 保留
#1040 G54 零点 X
#1041 G54 零点 Y
#1042 G54 零点 Z
#1043 G54 零点 A
#1044 G54 零点 B
#1045 G54 零点 C
#1046 G54 零点 U

#1047 G54 零点 V
#1048 G54 零点 W
#1049 保留
#1050 G55 零点 X
#1051 G55 零点 Y
#1052 G55 零点 Z
#1053 G55 零点 A
#1054 G55 零点 B
#1055 G55 零点 C
#1056 G55 零点 U
#1057 G55 零点 V
#1058 G55 零点 W
#1059 保留
#1060 G56 零点 X
#1061 G56 零点 Y
#1062 G56 零点 Z
#1063 G56 零点 A
#1064 G56 零点 B
#1065 G56 零点 C
#1066 G56 零点 U
#1067 G56 零点 V
#1068 G56 零点 W
#1069 保留
#1070 G57 零点 X
#1071 G57 零点 Y
#1072 G57 零点 Z
#1073 G57 零点 A
#1074 G57 零点 B
#1075 G57 零点 C
#1076 G57 零点 U
#1077 G57 零点 V
#1078 G57 零点 W
#1079 保留
#1080 G58 零点 X

#1081 G58 零点 Y
#1082 G58 零点 Z
#1083 G58 零点 A
#1084 G58 零点 B
#1085 G58 零点 C
#1086 G58 零点 U
#1087 G58 零点 V
#1088 G58 零点 W
#1089 保留
#1090 G59 零点 X
#1091 G59 零点 Y
#1092 G59 零点 Z
#1093 G59 零点 A
#1094 G59 零点 B
#1095 G59 零点 C
#1096 G59 零点 U
#1097 G59 零点 V
#1098 G59 零点 W
#1099 保留
#1100 中断点位置 X
#1101 中断点位置 Y
#1102 中断点位置 Z
#1103 中断点位置 A
#1104 中断点位置 B
#1105 中断点位置 C
#1106 中断点位置 U
#1107 中断点位置 V
#1108 中断点位置 W
#1109 坐标系建立轴
#1110 G28 中间点位置 X
#1111 G28 中间点位置 Y
#1112 G28 中间点位置 Z
#1113 G28 中间点位置 A
#1114 G28 中间点位置 B