

SHUKONG XICHUANG
SHIYONG HONGCHENGXU

数控铣床 实用宏程序

胡飞嘉◎编著



```
G54 G90 G0 X0 Y10  
S800 M3 Z5  
G1 Z-5 F500  
G41 D1 Y0 F200  
#1=0  
WHILE[#1 LE 9]DO1  
#2=40-21*[#1 AND 1]  
X[#2*COS[#1*36]]  
Y[#2*SIN[#1*36]]  
#1=#1+1  
X[#2*COS[#1*36]]  
Y[#2*SIN[#1*36]]  
END1  
G40 X0 Y0  
Z200 M5  
M30  
G00 G54 G40 G90 X5.  
S2500 M03
```



金盾出版社

数控铣床实用宏程序

胡飞嘉 编著

金盾出版社

内 容 提 要

本书结合大量实例循序渐进、深入浅出地介绍了三菱、法那科、哈斯系统数控宏程序功能及其差异,主要内容有:宏程序概述、宏程序语法、编程实例、系统变量、宏程序的调用、实用宏程序。

本书注重系统性、实用性、简洁性,实例全部来源于生产实践,并且用尽可能简单的程序解决问题。

本书可作为职业院校和培训机构相关专业的辅助教材,以及数控铣床和加工中心的操作人员与数控工艺人员的参考用书。

图书在版编目(CIP)数据

数控铣床实用宏程序/胡飞嘉编著. — 北京:金盾出版社,2015.1
ISBN 978-7-5082-9702-6

I. ①数… II. ①胡… III. ①数控机床—铣床—程序设计 IV. ①TG547
中国版本图书馆 CIP 数据核字(2014)第 218378 号

金盾出版社出版、总发行

北京太平路 5 号(地铁万寿路站往南)
邮政编码:100036 电话:68214039 83219215
传真:68276683 网址:www.jdcbs.cn
封面印刷:北京精美彩色印刷有限公司
正文印刷:北京万友印刷有限公司
装订:北京万友印刷有限公司
各地新华书店经销

开本:705×1000 1/16 印张:17.5 字数:424 千字

2015 年 1 月第 1 版第 1 次印刷

印数:1~3 000 册 定价:43.00 元

(凡购买金盾出版社的图书,如有缺页、
倒页、脱页者,本社发行部负责调换)

前 言

随着数控机床在机械制造行业的迅速普及,数控宏程序功能日益受到业内人士的重视。数控宏程序编程不仅是数控车工、数控铣工和加工中心操作工职业技能要求中,高级工、技师和高级技师所必须具备的技能,而且在生产实践中也有非常广泛的应用。数控宏程序编程在加工函数曲线、周边圆角、圆柱凸轮、分层铣深、主轴定向等诸多方面有独到的优势,程序的通用性、编程的简易性和调整的方便性是CAM软件所无法比拟的;而编写供调用的宏程序是对数控系统进行二次开发的主要途径。

为了使广大从业人员能够得心应手地使用数控宏程序,本书针对三菱、法那科、哈斯系统的数控宏程序功能进行了系统、全面的阐述。之所以把这三种系统一起讲述,是因为这三种数控系统不仅在国内应用比较广泛,而且编程的格式非常接近,便于读者举一反三、触类旁通。书中还详细说明了这三种系统的细节差别,以避免读者对有差别之处混淆和误用。

作者根据近20年的工作经验(宏程序编程、数控机床操作、计算机软件开发),结合大量的实例对宏程序的格式、用法、注意事项及宏程序的调用进行了深入浅出、全面细致的介绍。本书注重实用性,书中的全部实例均来自于生产实践(部分例子配有实物照片),使读者不仅会使用宏程序编程,而且知道数控宏程序的使用场合。本书注重程序的简练性,尽可能地用简洁的程序解决问题,避免简单的问题复杂化,以引起初学者的兴趣。

河南中光学集团的张宁和李丽参与了本书的校对工作,在此表示感谢。

书中程序虽经过使用,并在此次编写中再经审校,但不足之处难免,欢迎读者不吝指正。作者的电子邮箱为:hufejia@126.com。

作 者

作者简介:胡飞嘉,加工中心操作工,高级技师,全国技术能手,中央企业劳动模范。

目 录

第一章 宏程序概述	1
第一节 宏程序的概念	1
第二节 宏程序与相关领域的联系	3
第三节 使用宏程序的优势	4
第二章 宏程序语法	7
第一节 基本编程指令	7
第二节 变量	17
第三节 运算符和函数	20
第四节 程序流程控制	26
第五节 程序格式在不同系统中的细节差别	44
第六节 程序设计思想和程序结构	48
第二章练习题	52
第三章 编程实例	54
第一节 基本形状编程实例	54
第二节 补偿量用作变量的实例	100
第三节 非圆曲线编程实例	108
第四节 逻辑值参与运算的实例	118
第五节 圆柱面展开编程实例	123
第六节 主轴定向和线速度控制的实例	128
第七节 一次装夹加工多件的实例	145
第八节 无须使用变量编程即可加工的实例	151
第三章练习题	162
第四章 系统变量	168
第一节 检测当前位置的系统变量	168
第二节 控制当前状态的系统变量	171
第三节 有关时间和计时的系统变量	174
第四节 产生停止和报警的系统变量	176
第五节 补偿量	178
第六节 接口变量	179
第七节 系统参数对加工的影响	180
第四章练习题	185
第五章 宏程序的调用	187
第一节 调用宏程序时传递参数的方法	187
第二节 宏程序中当前状态的保存与恢复	190

第三节	设置参数的默认值	193
第四节	宏程序的层次	197
第五节	宏程序的非模态调用和模态调用	199
第六节	通过全局变量传递数据	202
第七节	把宏程序定义成 G 代码的步骤	207
第八节	编写宏程序的意义、思路和程序的改进	209
第九节	调用宏程序和调用子程序的区别	217
第五章练习题		217
第六章 实用宏程序		220
第一节	三菱系统的极坐标定点	220
第二节	FANUC 顺(逆)时针整圆铣削	221
第三节	FANUC 系统钻圆周均布孔	223
第四节	铣削加工的螺旋下刀	225
第五节	换刀并钻孔	226
第六节	铣削内螺纹	230
第七节	铣削外螺纹	232
第八节	FANUC 系统刻制编号	236
第九节	带准备时间的工件计时与计数	248
第六章练习题		249
附录		251
附录一	三菱、FANUC、HAAS 系统铣床 G 代码对照	251
附录二	常见编程工具中关键字的含义及对照	256
附录三	预备知识	258
附录四	数控铣床与加工中心安全生产操作规程	267
附录五	练习题答案	268

第一章 宏程序概述

数控机床目前已经非常普及,数控编程已广泛采用 CAM(计算机辅助制造)类软件自动生成程序。CAM 类软件确实能解决常见零件的编程问题,然而有时也存在一些不便之处。例如,仅为了修改一个圆角半径,或调整一个孔的位置,或调整拔模斜度的大小等,就需要重新生成一遍程序,使工作变得非常烦琐。当需要给一批零件刻制连续的编号,如果采用 CAM 类软件生成程序,则编程工作量非常大。而有些形状事先不能确定(如铸件外形要求见光、接平,分层铣深时每层的深度不同等),更无法生成程序。还有一些情况,如铣齿轮、铣大平面、周边倒角或圆角等,虽然能够用 CAM 类软件生成程序,但往往程序冗长,修改不便。对于这些情况,CAM 类软件已经不太适应,而解决问题的办法就是下面将要介绍的宏程序。

第一节 宏程序的概念

数控编程中的宏程序通常有两个含义:一个是指在程序体中直接使用变量(图 1-1),另一个是指宏程序调用(图 1-2)。



图 1-1 在程序体中直接使用变量



图 1-2 宏程序调用

在程序中直接使用变量就是用变量或计算式来代替数控程序中的数值。例如,G1 X#1,其中#1就是变量。在程序中使用变量有什么意义呢?可使用变量代替一个频繁出现的数值,或者代替某些不能事先确定的尺寸,更常见的是用来进行一些通过计算以实现某种函数关系。现用一个简单的例子来说明用变量代替一个频繁出现的数值的用法。

以下程序段可让铣床走出一个边长是 20mm 的正方形:

```
G91 G1 X20  
Y20  
X-20  
Y-20
```

如果想把这个正方形的边长改成 30mm,那就需要把 4 处 20 都改为 30。如果用一个变量来表示边长,把程序写成以下形式:

```
#1=20
G91 G1 X#1
Y#1
X-#1
Y-#1
```

那么,要把边长改成 30mm,只需要修改 #1 的值即可。这个例子虽然简单,但已经能充分体现出使用变量的方便了。对于用变量来表示函数关系的程序来说,优势更为明显。

宏程序调用类似于计算机程序设计语言中的函数,也就是用一条指令来代替一段程序。例如,铣床上常用的排屑式钻孔指令 G83 X_ Y_ Z_ R_ Q_ F_,就是一个宏指令,只不过它是数控系统原带的,不是用户自定义的。它实际上是调用了一段程序来实现钻孔的动作的。如果不用 G83 指令而用 G0,G1 来编写钻孔程序,将会非常冗长,更让人不能接受的是调整将非常烦琐。假如要更改每次钻削深度,就不仅仅是修改几个数值那么简单的了,很可能还要增加或删除一些程序段,而采用 G83 指令则只需修改 Q 值即可。可见,采用宏程序调用,不仅程序简洁,而且调整方便。以调用的方式使用宏程序还可以让用户对数控系统进行开发,定义自己的 G 代码,用来实现用户所指定的动作。本书的第五章将详细介绍以调用的方式使用宏程序,第六章给出了一些生产中实用的宏程序。

如果读者曾使用过计算机程序设计软件,则有助于理解数控宏程序。例如,要计算 1+2+...+100 之和,计算机程序和数控程序对比见表 1-1。

表 1-1 计算机程序和数控程序对比(计算 1+2+...+100)

计 算 机 程 序			数 控 程 序	
Basic (或 Visual Basic)	Pascal(或 Delphi)	C 语言	三菱,FANUC 或 HAAS	西门子
SUM=0 FOR I=1 TO 100 SUM=SUM+I NEXT I PRINT SUM END	var i,sum;integer; begin sum:=0; for i:=1 to 100 do sum:=sum+i; writeln(sum) end.	main() {int i=1,sum=0; while (i <= 100) sum+=i++; printf ("%d\n", sum);}	#101=1 #102=0 WHILE[#101 LE 100] DO1 #102=#102+ #101 #101=#101+1 END1 M30	DEF INT AA,SUM SUM=0 FOR AA=1 TO 100 SUM=SUM+AA ENDFOR MSG ("SUM="<<< SUM G4 F5 M30

表 1-1 仅是为了让读者对数控宏程序有一个感性认识。即使读者没有接触过计算机程序设计也没有关系,下文将详细介绍数控宏程序。

在不同的数控系统中,数控宏程序的用法也不完全一样。但总的说来,除了西门子系统的程序差别较大外,其他大多数的数控系统宏程序格式大同小异。本书以三菱、FANUC Oi 和 HAAS 系统为例来说明数控宏程序。这三种数控系统在国内应用比较广泛,而且编程的格式非常接近。把这三种系统放在一起介绍,便于读者举一反三、触类旁通。在编程的格式上有差别时,本书会详细说明系统间的差别。如果和这三种系统之外

的数控系统有较大的差异,也会做简略说明。

数控宏程序从发展历史上看,有宏程序 A 和宏程序 B 两种。宏程序 A 不能使用运算符和函数名,只能使用 G65 H $\times\times$ 的格式进行系统预先设定好的几种运算,所表达的含义不直观,现已很少使用。宏程序 B 可以像计算机编程一样使用变量、运算符和函数名,而且运算符、函数名和计算机编程大体上是相同的,程序易于理解。G65 指令在宏程序 B 中表示宏程序的调用,和宏程序 A 中表示一种运算的含义完全不同。本书所介绍的宏程序均指宏程序 B。

有的读者会觉得现在 CAM 类软件功能非常强大,数控宏程序似乎应该淘汰了,实则不然。早期的数控系统是不支持宏程序功能的,只是在发展的过程中为了应用方便,宏功能才越来越强大。CAM 软件编程和数控宏程序编程是两种不同的编程方法,并无优劣之别,只能说某一种零件的加工更适合用其中一种编程方法。好比汉语和英语都可以用来表达和交流,只是使用范围不同。要对数控系统有更深入的了解,进行一些细节的控制,开发更加个性化的功能(从用户角度看,而不是从数控系统设计人员的层面上看),数控宏程序就是一个强有力的工具。从日常工作的角度来看,有些零件使用宏程序编程要比 CAM 软件生成程序更为方便,而且可省去画图的时间。对于形状近似、尺寸不同的小批量多品种零件的加工,数控宏程序可提供良好的程序可控性、可调性。对于一次加工多件的零件来说,数控宏程序可对其中某一个零件进行单独控制。还有些程序(如本书中不找正铣齿轮、刻制连续编号、主轴定向和线速度控制等例子),只能使用宏程序,无法通过 CAM 软件生成程序。

第二节 宏程序与相关领域的联系

一、宏程序与 CAM 软件的联系

宏程序和 CAM 软件的使用场合有明显的不同。宏程序编程灵活、自由,能按照编程者的意愿控制机床的动作,几乎没有空抬刀动作,便于进行程序流程控制,程序可控性、可调性好,能够很方便地控制零件尺寸(尤其是公差较严格的尺寸),多数情况下不需要绘制三维图形,适合于形状规则变化或重复形状的编程,如图 1-3 所示为用数控宏程序加工的旋转抛物面。CAM 类软件可进行复杂形状的编程,不能进行程序流程控制,生成程序必须先绘制三维图形,适合于带有复杂曲面的零件(如模具类零件)的编程。在铣削周边圆角、拔模斜度、分层铣深等形状时,二者结合使用更加方便。

二、宏程序与数学的联系

要用宏程序编制零件的加工程序,必须首先建立数学模型,因此操作者应当具备必要的数学知识。编程中会遇到的有平面几何(如坐标转换)、平面解析几何(如加工抛物线或椭圆)、立体解析几何(如加工圆柱相贯线)、曲线斜率的计算(如不用半径补偿功能编非圆曲线)、弧微分和曲率的计算(如加工非圆曲线内形)、矩阵和行列式的计算(如用三次曲线拟合实测坐标点,无需人工计算)、多项式的化简(如加工非球面透镜模)、逻辑代数式的化简(宏程序中会用到)。不过常用的是平面几何和平面解析几何,以及为数不多的最小曲率半径的计算。



图 1-3 用数控宏程序加工的旋转抛物面

三、宏程序与计算机程序设计的联系

数控加工是计算机领域和机械制造领域的结合点,而数控宏程序编程则是计算机程序设计和机械加工的结合点。数控宏程序可以看作一种程序设计语言。正是有了数控宏程序,数控编程中才能够进行变量的计算和程序流程控制,从而使数控程序成为真正意义上的“程序”。计算机程序设计中的编程思想、程序结构同样适用于数控宏程序编程。数控编程中的运算符、函数名和计算机程序设计语言中的基本相同,掌握一种计算机程序设计语言对深入了解数控宏程序大有裨益。有的操作者之所以不喜欢使用宏程序,正是由于不了解程序设计思想造成的。

四、宏程序与机械加工的联系

数控宏程序是为加工零件而服务的。即使是数控编程,其加工工艺和切削用量的选择等也来自于编程人员的加工经验,因此操作者应当具备必要的机械制造知识,如金属切削原理、刀具的几何角度、切削用量的选择、金属材料与热处理等方面的知识。尽管本书主要从程序设计的角度讲述数控宏程序,但并不否认数控程序必须有一个合理的走刀路径,并选择合适的切削用量。

五、宏程序与数控系统的联系

数控宏程序必然要运行于一个具体的数控系统上。在一个系统上能够运行的程序,不一定可以转化为能在另一个系统上运行的程序。各系统都有大量的系统参数,系统参数设置不同时,程序执行效果也不同。操作者应当熟练掌握所操作的数控系统,了解机床性能、数控系统特点,以及本系统指令格式、指令功能、参数设置等。合理设置系统参数也能节省很多编程的工作量。

第三节 使用宏程序的优势

一、可控性

可控性是指能否对程序进行很方便地控制,例如加工过程中由于刀具折断而将机床

复位,在换好刀具后能否从需要的地方开始加工;多件同时加工时的件数能否设置;在铣外形或内腔轮廓时精铣遍数能否修改,等等。

如图 1-4 所示,有的零件在批量加工时,可在一个工装上安装多个零件,一次加工完成,减少了装夹、换刀等辅助时间。如果在编程时按照最多可加工的工件数考虑,每次执行程序都把工装上的零件全部加工出来,在正常加工时当然没有问题。但有时为了调试程序或测量尺寸,工装上只安装了较少的零件(如能安装 20 件但只装两件),在执行程序时把没有安装零件的位置也都空走一遍,就很浪费时间。这说明编程的可控性不好。如果在编程时能设置变量控制加工哪几件(工装上的某一件或几件,或全部),就可以避免空走刀的现象。这样程序的可控性就较好。

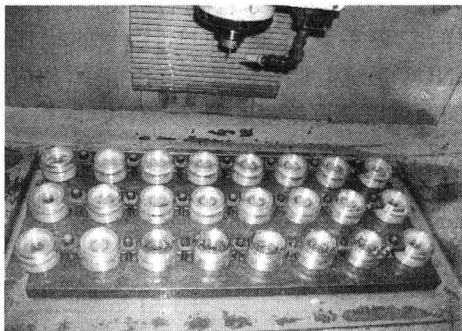


图 1-4 一次加工多个零件

二、可调性

可调性是指能否对加工出的形状、位置、下刀深度等进行调整(包括微调)。

例如,要铣削一个零件的铸件外形,其中一个圆角按图样要求是 $R8\text{mm}$,但有少部分零件由于铸造的原因形成塌角,造成 $R8\text{mm}$ 的圆角无法完全加工出来。为了不致把这些零件按废品处理,考虑把圆角半径加大,使其能够完全铣掉毛坯表面。如果程序这样编写:

```
G1 X0
Y42
G2 X8 Y50 R8
G1 X100
```

当圆角需要改成 $R9\text{mm}$ 时,需修改程序中的 3 个数值($Y42$ 改为 $Y41$, $G2 X8 Y50 R8$ 改为 $G2 X9 Y50 R9$),修改处较多,易出错,程序的可调性不好。而如果这样编写:

```
G1 X0
Y50,R8
X100
```

当圆角需要改成 $R9\text{mm}$ 时,只需要修改一处($R8$ 改为 $R9$),避免了计算两个切点的坐标,调整方便,该程序的可调性比第一个程序好。

三、适应性

适应性有两个含义:第一是指一个程序能否适应形状略有不同的工件,第二是指一个程序能否不加修改地在另一台机床或另一个系统上执行。

例如,要在程序中给 1 号半径补偿量赋值为 5(即 $D1=5$),如果直接给变量赋值,在三菱系统上是 $\#10001=5$,在 FANUC 0i 系统上是 $\#13001=5$,在 HAAS 系统上是 $\#2201=5$,程序适应性(通用性)不好。如果写成 $G10 L12 P1 R5$,则这些系统都可以执行,无须修改

程序就可以在另一台机床上运行。注意:对于不区分半径补偿和长度补偿的三菱系统来说,应把 L12 改为 L10。

再如,在三菱系统上有圆周均布孔指令 G34, HAAS 系统上有圆周均布孔指令 G70, 这样,三菱系统上带有 G34 指令的程序就可以转化为 HAAS 系统的格式。但是, FANUC 系统上没有圆周均布孔指令,相应的程序就不能转化成 FANUC 格式。不过这一问题可以通过编写宏程序来解决,即在 FANUC 系统上按照三菱系统 G34 的格式自定义 G34 指令,让它实现与三菱系统相同的动作。这样,三菱系统带有 G34 指令的程序就可以在 FANUC 系统上运行了。图 1-2 中的程序就是 G34 指令在 FANUC 系统上运行的例子。

四、容错性

容错性是指在一些容易出现错误的地方,程序应当给操作者必要的提示,避免由于人为操作失误而造成零件报废或刀具折断。

例如,在数控机床上大量加工同一种零件时,通常需对尺寸进行微调,操作者把数值输入机床时,一旦由于输入错误(如本来应键入-0.12,由于小数点未按下,变为-12),程序应能发现并产生异常(通常称为报警)。再如,当刀具过长时,应当检测当前 Z 坐标,避免主轴 Z 过低时快速水平移动发生碰撞。

程序的容错性体现在必要时(操作者可能出错时)能够给操作者以提示,而不是每次都提示。如果每次都提示,不仅起不到警示的作用,反而会给操作者带来干扰。

使用宏程序可对机床进行特殊控制,可自定义 G 代码或 M 代码,可进行一些个性化的设置。

第二章 宏程序语法

第一节 基本编程指令

一、代码书写格式

要使用数控宏程序,操作者应当熟知基本的数控编程指令并能够正确操作数控机床。数控机床采用笛卡儿右手三维直角坐标系,通常主轴所在的轴为 Z 轴。无论是工作台移动还是刀具移动,在编程时均按刀具移动对待。围绕 X,Y,Z 轴(或与其平行的轴)转动的旋转轴称为 A,B,C 轴;与 X,Y,Z 轴平行的轴称为 I,J,K 轴。本书附录1列出了三菱、FANUC、HAAS系统铣床G代码对照表可供读者查询。本节主要说明以下几个问题。

①一个程序由若干程序行组成,一个程序行(数控指令程序行)又由若干程序字组成,一个程序字包括地址和数值两部分。例如下列程序:

```
T1 M6  
G54 G90 G0 X0 Y0  
S1600 M3  
G43 H1 Z1  
G1 X100 F500  
.....
```

这个程序中的每一行都是程序行,如“S1600 M3”就是一个程序行。这个程序行由“S1600”和“M3”两个程序字组成,“S1600”由地址S和数值1600两部分组成。在本书中,如果提到“S值是1600”,指的就是地址S后的数值是1600。三菱、FANUC和HAAS数控系统中,地址都是用单字母表示的。绝大多数情况下,地址后面必须跟有数值。“X100”是一个程序字,也可以写成“X[50+50]”,这也是一个程序字。

②每一个程序都必须有一个程序号,程序号以字母O开头,后面跟一个正整数。如O1234表示这个程序是1234号程序。三菱系统可用的程序号最多是8位数,HAAS系统最多5位数,FANUC系统最多4位数。O1和O0001是同一个程序,但在西门子公司系统中程序号前面的O可表示不同的程序。程序号必须写在程序的第一行。程序号后面尽量写上注释,注释会显示在程序列表中。程序号可以任意给定(8000~9999号程序通常不用于主程序)。本书中的例子大都不带程序号,只有在子程序调用和宏程序调用时才带程序号。对于不带程序号的程序例子,读者调试时可任意给定一个程序号。

③本书程序中无小数点时均视同带有小数点,长度单位为mm、角度单位为 $^{\circ}$ 、时间单位为s,数控系统通常可以设置成这样。

例如:G1 X100 Y-20 视同 G1 X100. Y-20.

需要注意,FANUC系统的暂停时间P单位是ms,HAAS系统圆角R、倒角C的数值在无小数点时单位是 μm ,并无法通过系统设置进行修改。

④数值中无用的0可以省去(包括整数的前导0和小数末尾的0),例如 G01 X-30.0 Y0.50 M08 可以写成 G1 X-30 Y.5 M8。无用的标号(N_)可以省去。标号仅在子程序调用时(即三菱系统的 M98 H_或 HAAS 系统的 M97 P_)或程序跳转时(即 GOTO_或 M99 P_)需要写上。

⑤每行程序末尾的分号是程序行的一部分,程序行必须有分号才能执行。但需要注意,大多数情况下程序是在计算机上编写好后传入机床的,而在计算机上输入程序时,回车符和分号的作用是一样的。因此,在计算机上输入程序时,每行程序后由于都要输入回车符,所以就不必再输入分号了。程序传入机床后自然会带上分号。如果程序不是在计算机上编好传入机床的,而是直接在机床的面板上输入的,则程序行末尾的分号一定不能省去。往往程序的最后一行或者在 MDI 模式下只有一行程序时,容易漏打分号。

⑥以“/”开头的程序行是可被跳过的程序行。当控制面板上的“任选段跳跃”按键被按下时,以“/”开头的程序行将被跳过而不被执行。如果“任选段跳跃”按键未按下,则以“/”开头的程序行照常执行。有一些 FANUC 系统支持“/”出现在程序行的中间,如 G1 X20 /Y10,当“任选段跳跃”按键被按下时执行 G1 X20,未被按下则执行 G1 X20 Y10。

⑦计算机上的字母、数字和符号都有半角和全角之分,或者说有英文字符和中文字符之分。在计算机上输入程序时,一定要注意输入英文字符,不能使用中文字符(如 X50 不能打成 X 5 0)。在输入程序时,尽量关闭输入法,即使不关闭也要注意使输入法处于英文状态。如果程序在计算机上查看没有错误而传入机床后出现乱码,首先要检查是否输入了中文字符。此外,在输入程序时还要注意一些字形相近的字符,如字母 O 和数字 0、小写字母 l 和数字 1 等不要混淆。数控系统显示的数字 0 里面有一撇。

⑧计算式中需要使用括号时,不要像数学里那样依次使用小括号、中括号、大括号,数控程序里一律使用中括号(方括号),小括号用来表示注释。西门子系统则使用小括号,而用分号表示注释。

⑨注释一般不要省略。注释就是程序中用小括号(圆括号)括住的部分。注释说明了程序所加工零件的名称、程序的意图等。如果省略注释,虽然不影响程序的运行,但会使人不易理解,也不利于事后查找程序。注释一般使用汉语拼音或英文。除非系统支持中文,否则不要使用中文注释。本书的程序实例为了说明方便,多使用中文注释,读者在调试时不能把中文注释也输入。程序中一般只能用大写字母,但在注释中可以使用小写字母;有的 FANUC 系统在注释中也不能使用小写字母。如果使用的是三菱系统,即使在计算机上编写的程序有空格,传入机床后系统会自动去掉空格。如果确实需要在三菱系统中显示空格,在计算机上编写程序时不要按空格键,而应先按住 Alt 键再依次在小键盘区按下 1,2,7 三个键,然后松开 Alt 键,就会出现一个空白符,该程序传入三菱系统后空格就会显示出来了。

有两种情况下小括号内的内容不是注释:一是给 #3000 或 #3006 赋值后带的括号,表示自定义报警或提示的内容;二是 HAAS 系统 G47 指令后的括号,表示要雕刻的文本。

⑩指令有模态和非模态的区别。模态代码只要不被同组的代码取代,就一直保持有效(G 代码的分组参看本书附录 1),如 G0,G1,G2,G3,G61 等。非模态代码只在当前程序行中起作用,如 G4,G9 等。模态代码有相应的取消功能的代码,而非模态代码不存在续效性,故无须取消。例如,G9(准确停止检查)和 G61(准确停止检查模式)功能一样,只是 G9 是非模态的,而 G61 是模态的,故有取消 G61 的代码 G64。要知道某个 G 代码是

不是模态的,可查阅本书附录 1,根据该代码所在的组号来确定。00 组代码只有 G10, G11 是模态的,其余均是非模态的。非 00 组代码都是模态的。

除了 G 代码外,S,T,F 代码也是模态的。

二、几个主要指令的功能

要使用宏程序编程,应当熟知数控系统已经提供的 G 代码的功能,避免由于对现有 G 代码不了解而走弯路。例如要加工间距相同的一排孔,可以在孔加工指令(G73~G89)的下一行用增量方式 G91 来描述下一个孔的位置,并写上 L_(FANUC 系统用 K_)来表示孔数。如果操作者不知道或不会使用 G91 而又编写宏程序来计算每个孔的坐标,就使问题复杂化了。本节主要说明几个值得重视或容易引起误解的指令的功能。合理地使用这些指令,可在编程时起到事半功倍的作用。一些实用的指令如下。

1. 数据输入指令 G10

G10 指令可以用于输入工件坐标系值或补偿量值。

输入工件坐标系值的格式是 G10 L2 P_ X_ Y_ Z_ (P1 表示 G54, P2 表示 G55,…… P6 表示 G59)。如要把 G54 坐标系设为 X-300 Y-200 Z-150,指令是 G10 L2 P1 X-300 Y-200 Z-150。但通常情况下工件坐标系值是对刀后直接输入的,不是通过程序输入的。

输入长度补偿量值的格式是 G10 L10 P_ R_;输入半径补偿量值的格式是 G10 L12 P_ R_。P 值是补偿号,R 值是补偿量。例如要使半径补偿量 D1=8,指令是 G10 L12 P1 R8。

2. 准确停止检查指令 G9, G61

对于大型数控机床,由于机床工作台质量很大,运动时有很大的惯性,不可能立即停下来。而走刀过程中为了保持恒定的走刀速度,在转角处必然走成圆弧。工作台(连同工件)质量越大、移动速度越快、转角的角度越小,所形成的误差越大。G9 指令就是为了避免造成过大的误差而设的。G9 指令可在上一段走到转角处时,减速到 0,然后再加速到 F 值指定的速度继续走下一段。G9 指令是非模态的,仅当前程序行有效。如果需要持续有效,应使用 G61。取消 G61 模式的指令是 G64。如果仅减速仍不能满足要求而需暂停,则应使用 G4 指令。

3. 转角补偿圆弧插补指令 G39

G39 指令可在铣削工件外形时补偿成圆角,圆角的半径等于补偿量。也就是把转角处看作有一个 R0 的圆弧进行补偿,如图 2-1 所示。

补偿成圆角的意义不仅在于缩短了走刀路径,更可以实现一些特殊的效果,从而简化编程。例如,要铣削一个如图 2-2 所示的槽,铣刀直径 $\phi 10\text{mm}$,槽宽 13mm,按照常规的编程方法,程序中必然有一段圆弧。如果按照如下方式编程,可避免出现圆弧。

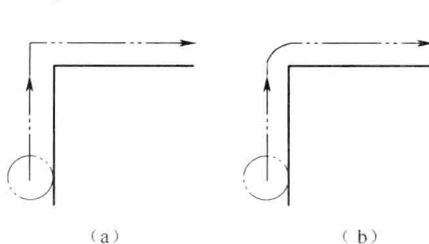


图 2-1 G39 走刀路径

(a)无 G39 时 (b)有 G39 时

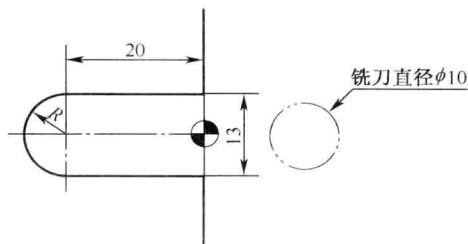


图 2-2 G39 指令例子之一

```

G0 X10 Y0
G42 D1 G1 X6 F_
G39 X-20
X6
G40 G0 X10 Y0

```

半径补偿量 D1 的值不是铣刀的半径,而是槽宽的一半减去铣刀半径的差值,即 $6.5 - 5 = 1.5(\text{mm})$ 。如果觉得这里不好理解,可以这样考虑:假如不带补偿,铣刀走进去后又沿原路返回(不考虑让刀现象),则铣出的槽宽就是铣刀的直径。现在对路径进行了补偿,补偿量是 1.5mm ,即来回走时各偏移 1.5mm ,那么就把槽扩宽了 3mm ,槽宽就铣成了 13mm 。

再如铣削图 2-3 所示的上层形状,按照通常的考虑使用轮廓编程,需考虑两段直线和两段圆弧。如果把这个形状看成一条长 30mm 的线段(两端点就是两个半圆的圆心)并进行补偿,程序就简单多了。使用 $\phi 20\text{mm}$ 的铣刀, X, Y 原点设在左边半圆的圆心处, Z 原点设在上表面。三菱系统的程序如下:

```

G54 G90 G0 X-30 Y25
S800 M3
Z1
G1 Z-10 F1000
G41 D1 Y0 F200
X30
G39 X0
G39 X1
G40 Y25
G0 Z100
M5
M30

```

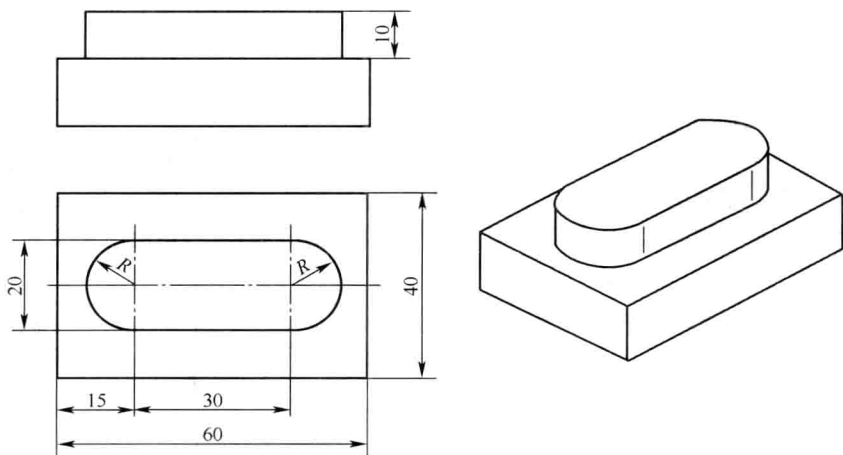


图 2-3 G39 指令例子之二

半径补偿量 D1 的值是凸台宽度的一半加上铣刀的半径,即 $10 + 10 = 20(\text{mm})$ 。如果