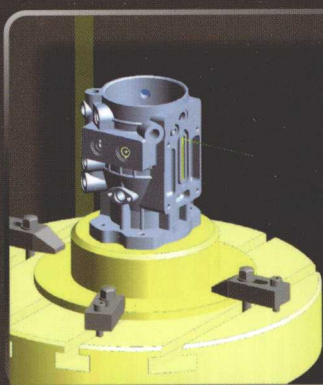


高级数控培训丛书

数控宏程序应用技术 及实例精粹

李 锋 编著

SHUKONG HONGCHENGXU
YINGYONGJISHU
JI SHILI JINGCUI



化学工业出版社

013050754

TG659
476

高级数控培训丛书

数控宏程序应用技术 及实例精粹

李 锋 编著

SHUKONG HONGCHENGXU
YINGYONGJISHU
JI SHILI JINGCUI



北航

C1657479



化学工业出版社

· 北京 ·

TG659
476

图书在版编目 (CIP) 数据

数控宏程序应用技术及实例精粹 / 李锋编著. —北京:
化学工业出版社, 2013.5
高级数控培训丛书
ISBN 978-7-122-16304-2

I. ①数… II. ①李… III. ①数控机床-程序设计
IV. ①TG659

中国版本图书馆 CIP 数据核字 (2013) 第 006905 号

责任编辑: 王 烨
责任校对: 吴 静

文字编辑: 谢蓉蓉
装帧设计: 王晓宇

出版发行: 化学工业出版社 (北京市东城区青年湖南街 13 号 邮政编码 100011)

印 刷: 北京永鑫印刷有限责任公司

装 订: 三河市万龙印装有限公司

787mm×1092mm 1/16 印张 12½ 字数 313 千字 2013 年 7 月北京第 1 版第 1 次印刷

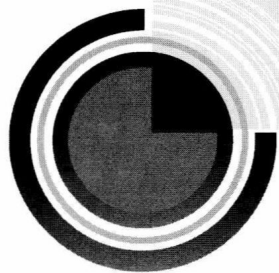
购书咨询: 010-64518888 (传真: 010-64519686) 售后服务: 010-64518899

网 址: <http://www.cip.com.cn>

凡购买本书, 如有缺损质量问题, 本社销售中心负责调换。

定 价: 39.00 元

版权所有 违者必究



FOREWORD

前言

随着科学技术的发展,数控机床在机械制造业中的应用越来越广泛,而在对数控机床的应用中,数控系统厂家所提供的宏程序、参数编程功能,并没有得到广泛应用。同时,市场上介绍此功能的书籍相对较少。鉴于此,编者通过理论梳理,并把亲身实践的、在机床加工中成功运用的实例,奉献给广大读者。

本书力求理论联系实际,全书列举五十多个实例,且对每个实例,作者都用 FANUC、SIEMENS 和华中数控系统分别进行宏程序的编制,旨在帮助读者提高编程的方法及技巧。

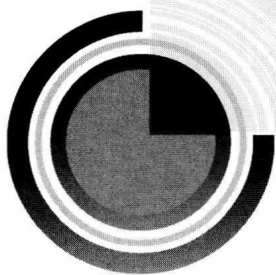
书中详细介绍了 FANUC 宏程序、SIEMENS R 参数和华中数控宏程序编程的基础知识,然后通过实例把数控系统的参数编程知识、技巧以及与每个实例相关的数学知识、如何进行数学建模、加工应注意事项和易犯的错误等都进行了讲述,从而让读者对如何运用宏程序及参数编程有一个更全面的了解。

本书可作为各高职院校、高级技工学校、技术学院、技师学院及参加各类数控大赛人员的培训教材,同时也可作为各高等院校进行数控机床编程系统二次开发的参考用书。

本书由陕西航天职工大学机电系李锋编著,在编写过程中得到西安航空发动机集团有限公司高级工程师李广琦、李晓敏的多次指导,同时也得到校领导的大力支持及教务处张立新、袁秋岐、李文斌、周怀兵、王红珠、李正泽、王海莲老师的帮助,在此深表谢意。

由于作者的知识水平有限,书中疏漏之处敬请同行及读者不吝指正。

编 者

**绪论 变量编程概述**

- 0.1 变量编程的特点 /1
- 0.2 变量编程在数控系统中的运行过程 /2

第1章 宏程序编程基础知识

- 1.1 FANUC 数控系统宏程序编程基础知识 /3
 - 1.1.1 变量概述 /3
 - 1.1.2 系统变量 /5
 - 1.1.3 运算指令 /9
 - 1.1.4 转移与循环指令 /10
 - 1.1.5 参数在编程中的赋值方法 /11
 - 1.1.6 用户宏程序报警及处理方法 /12
 - 1.1.7 有关用户宏程序的参数 /13
 - 1.1.8 用户宏程序的使用限制 /15
- 1.2 SIEMENS 数控系统参数编程基础知识 /16
 - 1.2.1 R 参数 /16
 - 1.2.2 算术运算规律 /17
 - 1.2.3 程序跳转 /18
 - 1.2.4 参数在编程中的赋值方法 /20
- 1.3 华中数控系统宏程序编程基础知识 /20
 - 1.3.1 宏变量及常量 /20
 - 1.3.2 运算符与表达式 /23
 - 1.3.3 赋值语句 /23
 - 1.3.4 条件判别语句 IF ELSE END IF /23
 - 1.3.5 循环语句 WHILE ENDW /23

第2章 宏程序编程技术及实例

- 2.1 常规编程技术及实例 /24
 - 2.1.1 数控车抛物线 /24
 - 2.1.2 数控车余弦曲线 /27

- 2.1.3 数控车内腔去余量 /29
- 2.1.4 G17 平面加工椭圆 /31
- 2.1.5 椭圆均等分层加工 /33
- 2.1.6 椭圆不均等分层加工 /35
- 2.1.7 双曲线 /37
- 2.1.8 抛物线 /40
- 2.1.9 正弦曲线 /42
- 2.1.10 余弦曲线 /45
- 2.1.11 正切曲线 /47
- 2.1.12 五边形 /49
- 2.1.13 六边形 /52
- 2.1.14 铣花边 /54
- 2.1.15 铣削圆周平面 /57
- 2.1.16 铣削大平面 /58
- 2.1.17 内环去余量 /61
- 2.1.18 铣槽 /62
- 2.1.19 凸轮加工 /69
- 2.2 孔加工编程技术及实例 /78
 - 2.2.1 圆周钻孔 /78
 - 2.2.2 带角度排孔 /84
 - 2.2.3 环形钻孔 /88
 - 2.2.4 三角形孔位加工 /90
 - 2.2.5 排孔 /92
 - 2.2.6 矩阵孔加工 /96
- 2.3 深孔加工编程技术及实例 /99
- 2.4 变刀补编程技术 /102
 - 2.4.1 锥体 /102
 - 2.4.2 圆锥体 /105
 - 2.4.3 孔口倒 90° 圆角 /109
 - 2.4.4 孔口倒角 /111
 - 2.4.5 圆柱倒 90° 圆角 /114
 - 2.4.6 圆柱倒直角 /117
 - 2.4.7 等距离外倒角 /119
- 2.5 规则曲面编程技术 /122
 - 2.5.1 孔口倒小于 90° 圆角 /122
 - 2.5.2 圆柱倒小于 90° 圆角 /125
 - 2.5.3 半球体加工 /127
 - 2.5.4 矩形内孔口倒凹圆弧 /131
 - 2.5.5 矩形内孔口倒凸圆弧 /134
 - 2.5.6 孔口倒角 /137

- 2.5.7 等半径外加工 /139
- 2.5.8 等半径内加工 /144
- 2.6 螺纹加工编程技术及实例——在孔为 $\phi 28$ 的孔内铣深30的螺纹 /147
- 2.7 多轴编程技术及实例——在圆环上钻孔 /151

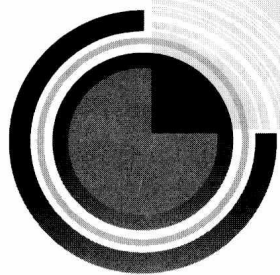
第3章 宏程序仿真加工技术

- 3.1 总体功能 /154
- 3.2 详细功能介绍 /155
 - 3.2.1 VERIFICATION 2轴、3轴基本数控程序加工仿真验证 /155
 - 3.2.2 Machine simulation 机床实体仿真模块 /156
 - 3.2.3 Multi-Axis 4轴、5轴及多轴联动数控加工仿真验证 /156
 - 3.2.4 OptiPath 数控程序优化 /157
 - 3.2.5 AUTO-DIFF 仿真后的零件模型与设计原型自动比较过切和欠切 /157
 - 3.2.6 Unigraphics Interface UG 接口 /158
 - 3.2.7 模型输出 (Model Export) /158
 - 3.2.8 探头仿真 (CNC machine probing) /158
 - 3.2.9 step 模型接口 /158
- 3.3 数控车椭球体加工及仿真 /158
- 3.4 铣槽仿真 /166
- 3.5 数控铣椭圆体加工及仿真 /173

第4章 数控系统功能 G 代码宏开发技术

- 4.1 数控车正弦曲线 /181
- 4.2 半球型腔去余量 /185

参考文献



绪论

变量编程概述

变量编程是指在程序中，用变量表述一个地址的数字值。而在程序中对变量进行赋值，可以使程序在几个相似的地方使用。在本书里，我们把 FANUC 系统的宏程序和 SIEMENS 系统的 R 参数编程统称为变量编程。

变量编程与普通程序的区别在于：在变量编程中，可以使用变量，可以给变量赋值，变量间可以运算，程序运行可以跳转；而在普通程序中，只可指定常量，常量之间不可以运算，程序只能按顺序执行，不能跳转，功能是固定的。

0.1 变量编程的特点

(1) 高效

数控加工中常常会遇到数量少，品种繁多，有规则几何形状的工件，在编程时我们只要稍加分析与总结，找出他们之间的共同点，把这些共同点设为变量应用到程序中，编制出一个程序通过改变其中共性的变量就可以解决一类问题，使我们在加工相类似零件时，只需改变其中几个变量中的赋值，就可以采用此程序进行零件加工，大大节省了编程时间，而且在运用时准确性也大大提高。即使使用 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。

(2) 经济

在实际生产中，经常会出现许多结构相似，但品种多、数量少的零件。这些零件在某些特征上变化不定，按常规加工方法，往往需要用成型刀加工。但零件品种多，所以成型刀需要量很大，定做一把成型铣刀要比普通铣刀费用高出 2~3 倍，为了降低加工成本，采用变量编程此类问题即可解决。同时许多在实际加工中必须依靠球头铣刀加工的零件，我们只需平底刀就可以解决了。

(3) 应用范围广

变量在实际加工中还可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)、G 代码、M 代码等进行设置，大大地提高了加工效率。

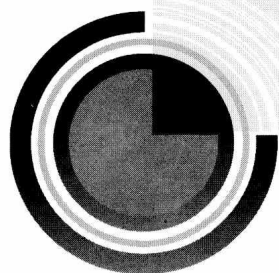
(4) 有利于解决软件编程带来的缺陷

① 对于软件编程来说,一般编制的曲面加工程序的容量比较大。而一般的数控系统的标准配置为 128k 或 256k,当一次传输不完程序时,我们就要考虑在线加工了。当在线加工时,就会出现程序的传输速率往往跟不上机床的节拍。原因在于:常见的数控系统如 FANUC-0i、SIEMENS 802D、810D 等系统所支持的 RS-232 口最大传输波特率为 19200bit/s,而大多数传输软件支持的最大波特率在 19200~38400 bit/s,即使在 19200bit/s 的波特率下工作,当计算精度较高、进给速度值较大时,在实际加工中就可以看到机床的进给运动有明显的断续、迟滞。而使用变量编程时,即使再复杂的数控加工程序都比较精炼,可以说任何一个合理优化的变量编程,极少会超过 60 行,换算成字节数,至多不过 2kB,根本不用在线加工。

② 从用户使用的角度来说,使用 CAD/CAM 软件来生成刀具轨迹及加工程序是非常容易的事,但是剖析 CAD/CAM 软件计算刀具轨迹的原理,就知道它存在一定的弊端。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(或称为曲面的计算精度),使刀具与加工表面接触点(相交点或相切点)逐点移动完成加工,从本质上看,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。而使用变量编程时,为了对复杂的加工运动进行描述,变量编程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(螺旋)插补 G02/G03 指令等。因此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。

0.2 变量编程在数控系统中的运行过程

- (1) 读取数控代码。
- (2) 提取变量和变量定义。
- (3) 将预先保存的全局变量、系统变量和用户自定义变量保存在相关列表中。
- (4) 读取数控代码,提取复杂表达式。
- (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) 按照结果输出实际的数值。



第1章

宏程序编程基础知识

1.1 FANUC 数控系统宏程序编程基础知识

FANUC 公司 1972 年由富士通信机械制造株式会社独立出来, 1976 年研制成功数控系统 5, 1979 年推出数控系统 6, 1980 年在系统 6 的基础上向低档和高档扩展, 研制系统 3 和系统 9。1984 年又推出新型产品数控系统 10、系统 11、系统 12, 该产品在硬件方面做了较大的改进, 采用了大规模集成电路、32 位高速处理器、光导传输、磁泡存储器等技术, 提高了抗干扰性和可靠性。该系统在工厂自动化 DNC 方面能够实现主计算机与机床、工作台、机械手、搬运车等之间各种数据的双向传递, 充实了专用宏程序功能、自动计划功能、自动刀具补偿功能、刀具寿命管理、彩色图形显示 CRT 等。PLC 编程不仅可以使用梯形图语言, 还可以使用 PASCAL 语言, 便于用户自己开发软件; 1985 年推出数控系统 0, 它的主要目标是体积小、价格低, 适用于机电一体化的小型机床, 主要特点有彩色图形显示、会话菜单式编程, 专用宏程序功能、多种语言(汉语、英语、德语、法语)显示、录返(Playback)功能等; 1987 年研制成功数控系统 15, 称之为划时代的智能型 AI-CNC 系统, 它应用了 MMC (Man Machine Control)、CNC、PMC (Programmable Machine Control) 的新概念, 采用了高速度、高精度、高效率加工的数字伺服单元、数字主轴单元和纯电子式绝对位置检测器, 还增加了 MAP、窗口功能等。

宏程序是数控系统生产厂家留给用户在数控系统的平台上进行开发的工具, 当然这里的开放和开发都是有条件和有限制的。通过以下知识的学习, 我们就可以充分利用数控系统所提供的宏程序功能了。

1.1.1 变量概述

(1) 变量表示

#I (I=1, 2, 3, ...) 或#[<式子>]

例: #5, #109, #501, #[#1+#2-12.]

(2) 变量的使用

① 地址字后面指定变量号或公式

格式: <地址字> # I

<地址字> - # I

<地址字> [<式子>]

例: F#103, 设 #103 = 150 则为 F150;

Z-#110, 设 #110 = 250 则为 Z-250;

X [#24 + [#18 * COS [#1]]]。

② 变量号可用变量代替

例: #[#30], 设 #30 = 3 则为 #3

③ 程序号、顺序号和任选程序段跳转号不能使用变量

例: 下述方法不允许

O#1;

/#2G0X100.0;

N#3Z200.0;

④ 变量号所对应的变量, 对每个地址来说, 都有具体数值范围。

例: #30 = 1100 时, 则 M#30 是不允许的。

⑤ #0 为空变量, 没有定义变量值的变量也是空变量。

⑥ 变量值定义: 程序定义时可省略小数点, 例 #123 = 149

(3) 变量的类型

变量根据变量号可以分为四种类型, 功能见表 1-1。

表 1-1 变量的类型及功能

变量号	变量类型	功 能
#0	空变量	该变量总是空, 没有值赋给该度量
#1~#33	局部变量	局部变量只能用在宏程序中存储数据, 例如, 运算结果 当断电时, 局部变量被初始化为空, 调用宏程序时, 自变量对局部变量赋值
#100~#199	公共变量	公共变量在不同的宏程序中的意义相同
#500~#999		当断电时, 变量#100~#199 初始化为空, 变量#500~#999 的数据保存, 即使断电也不丢失
#1000 以上	系统变量	系统变量用于读和写 CNC 的各种数据, 例如, 刀具的当前位置和补偿值等

(4) 变量值的范围

局部变量和公共变量可以为 0 值或以下范围中的值: -10^{47} 到 -10^{-29} 或 10^{-29} 到 10^{47}

(5) 变量的引用

① 在地址后指定变量号即可引用其变量值, 当用表达式指定变量时, 要把表达式放在括号中。

例如: G1X[#1+#2]F#3;

被引用变量的值会依据地址的最小设定单位自动地进行取舍。

例如: 当系统的最小输入增量为 1/1000mm 单位, 指定 G00X#1, 并将 12.3456 赋值给变量#1, 实际指定值为 G00X12.346;

② 改变引用变量值的符号，要把负号（-）放在#的前面。

例如：G00X-#1；

③ 当引用未定义的变量时，变量及地址字都被忽略。

例如：当变量#1 的值是 0，并且变量#2 的值是空时，G00X#1Y#2 的执行结果为 G00X0。

1.1.2 系统变量

系统变量用于读和写 NC 内部数据，同时它也是自动控制和通用程序开发的基础。

(1) 接口信号

它是可编程机床控制器（PMC）和用户宏程序之间交换的信号，功能见表 1-2。

表 1-2 接口信号功能

变量号	功 能
#1000~#1015 #1032	把 16 位信号从 PMC 送到用户宏程序。变量#1000 到#1015 用于按位读取信号。变量 # 1032 用于一次读取一个 16 位信号
# 1100~# 1115 # 1132	把 16 位信号从用户宏程序送到 PMC。变量#1100 到#1115 用于按位写信号。变量 # 1132 用于一次写一个 16 位信号
#1133	变量#1133 用于从用户宏程序一次写一个 32 位的信号到 PMC。注意，#1133 的值为-99999999 到+99999999

(2) 刀具补偿值

用系统变量可以读和写刀具补偿值。

可使用的变量数取决于刀补数，是否区分外形补偿和磨损补偿以及是否区分刀长补偿和刀尖补偿。当偏置组数小于等于 200 时，也可使用#2001~#2400。刀具补偿存储器 C 的系统变量，见表 1-3。

表 1-3 刀具补偿存储器 C 的系统变量

补偿号	刀具长度补偿 (H)		刀具半径补偿 (D)	
	外形补偿	磨损补偿	外形补偿	磨损补偿
1	#11001 (#2201)	#10001 (#2001)	#13001	#12001
⋮	⋮	⋮	⋮	⋮
200	#11201 (#2400)	#10201 (#2200)	⋮	⋮
⋮	⋮	⋮	⋮	⋮
400	#11400	#10400	#13400	#12400

(3) 宏程序报警

宏程序报警的系统变量，功能见表 1-4。

表 1-4 宏程序报警的系统变量功能

变量号	功 能
#3000	当变量#3000 的值为 0~200 时，CNC 停止运行且报警 可在表达式后指定不超过 26 个字符的报警信息 CRT 屏幕上显示报警号和报警信息，其中报警号为变量#3000 的值加上 3000

例：#3000=1 (TOOL NOT FOUND)

→报警屏幕上显示“3001 TOOL NOT FOUND”(刀具未找到)。

(4) 停止和信息显示

程序停止执行并显示信息，功能见表 1-5。

表 1-5 程序停止执行并显示信息功能

报警号	功 能
#3006	在宏程序中指令“3006=1 (MESSAGE);”时，程序在执行完前一程序段后停止 可在同一个程序段中指定最多 26 个字符的信息，由控制输入“(”和控制输出“)”，相应信息显示在外部操作信息画面

(5) 时间信息

时间信息可以读和写，时间信息的系统变量功能见表 1-6。

表 1-6 时间信息的系统变量功能

变量号	功 能
#3001	该变量为一个计时器，以 1 毫秒为计时单位 当电源接通时，该变量值复位为 0 当达到 2147483648 毫秒时，该计时器的值返回到 0
#3002	该变量为一个计时器，以一个小时为单位计时。 该计时器即使在电源断电时也保存数值。 当达到 9544.371767 小时，该计时器的值返回到 0
#3011	该变量用于读取当前的日期（年/月/日）。年/月/日信息转换成十进制数。 例如，2001 年 9 月 28 日表示为 20010928
#3012	该变量用于读取当前的时间（时/分/秒）。时/分/秒信息转换成十进制数。 例如，下午 3 点 34 分 56 秒表示为 153456

(6) 模态信息

模态信息的系统变量见表 1-7。

表 1-7 模态信息的系统变量

变量号	功 能
#4001	G00, G01, G02, G03, G33
#4002	G17, G18, G19
#4003	G90, G91
#4004	
#4005	G94, G95
#4006	G20, G21
#4007	G40, G41, G42
#4008	G43, G44, G49
#4009	G73, G74, G76, G80~G89
#4010	G98, G99
#4011	G50, G51
#4012	G65, G66, G67
#4013	G96, G97
#4014	G54~G59
#4015	G61~G66
#4016	G68, G69
⋮	⋮

续表

变量号	功 能
#4022	
#4102	B 代码
#4107	D 代码
#4109	F 代码
#4111	H 代码
#4113	M 代码
#4114	顺序号
#4115	程序号
#4119	S 代码
#4120	T 代码
#4130	P 代码（现在选择的附加工件坐标系）

(7) 当前位置

位置信息不能写，只能读。信息系统的系统变量见表 1-8。

表 1-8 信息系统的系统变量

变量号	位置信息	坐标系	刀具补偿值	运动时的读操作
#5001~#5003	程序段终点	工件坐标系	不包括	可能
#5021~#5023	当前位置	机床坐标系		
#5041~#5043	当前位置	工件坐标系	包括	不可能
#5061~#5063	跳转信号位置			可能
#5081~#5083	刀具长度补偿值			不可能
#5101~#5103	伺服位置偏差			

第 1 位代表轴号（从 1 到 3）。

变量#5081~#5083 存储的刀具长度补偿值是当前的执行值，不是后面程序段的处理值。

在 G31（跳转功能）程序段中跳转信号接通时的刀具位置储存在变量#5061 到#5063 中。

当 G31 程序段中的跳转信号未接通时，这些变量中储存指定程序段的终点值。

移动期间不能读是指由于缓冲（预读）功能的原因，不能读期望值。

(8) 工件坐标系补偿值

工件零点偏移值可以读和写，工件零点偏移值的系统变量功能见表 1-9，表 1-10。

表 1-9 工件零点偏移值的系统变量功能

变量号	功 能
#5201	第 1 轴外部工件零点偏移
⋮	⋮
#5203	第 3 轴外部工件零点偏移值
#5221	第 1 轴 G54 工件零点偏移值
⋮	⋮
#5223	第 3 轴 G54 工件零点偏移值
#5241	第 1 轴 G55 工件零点偏移值
⋮	⋮
#5243	第 3 轴 G55 工件零点偏移值

续表

变量号	功 能
#5261 ⋮ #5263	第 1 轴 G56 工件零点偏移 ⋮ 第 3 轴 G56 工件零点偏移
#5281 ⋮ #5283	第 1 轴 G57 工件零点偏移 ⋮ 第 3 轴 G57 工件零点偏移
#5301 ⋮ #5303	第 1 轴 G58 工件零点偏移 ⋮ 第 3 轴 G58 工件零点偏移
#5321 ⋮ #5323	第 1 轴 G59 工件零点偏移 ⋮ 第 3 轴 G59 工件零点偏移
#7001 ⋮ #7003	第 1 轴工件零点偏移值 (G54.1 P1) ⋮ 第 3 轴工件零点偏移值 (G54.1 P1)
#7021 ⋮ #7023	第 1 轴工件零点偏移值 (G54.1 P2) ⋮ 第 3 轴工件零点偏移值 (G54.1 P2)
⋮	⋮
#7941 ⋮ #7943	第 1 轴工件零点偏移值 (G54.1 P48) ⋮ 第 3 轴工件零点偏移值 (G54.1 P48)

表 1-10 工件零点偏移值的系统变量号

轴	功 能	变 量 号	
第 一 轴	外部工件零点偏移	#2500	#5201
	G54 工件零点偏移	#2501	#5221
	G55 工件零点偏移	#2502	#5241
	G56 工件零点偏移	#2503	#5261
	G57 工件零点偏移	#2504	#5281
	G58 工件零点偏移	#2505	#5301
	G59 工件零点偏移	#2506	#5321
第 二 轴	外部工件零点偏移	#2600	#5202
	G54 工件零点偏移	#2601	#5222
	G55 工件零点偏移	#2602	#5242
	G56 工件零点偏移	#2603	#5262
	G57 工件零点偏移	#2604	#5282
	G58 工件零点偏移	#2605	#5302
	G59 工件零点偏移	#2606	#5322
第 三 轴	外部工件零点偏移	#2700	#5203
	G54 工件零点偏移	#2701	#5223
	G55 工件零点偏移	#2702	#5243
	G56 工件零点偏移	#2703	#5263
	G57 工件零点偏移	#2704	#5283
	G58 工件零点偏移	#2705	#5303
	G59 工件零点偏移	#2706	#5323

1.1.3 运算指令

运算式的右边可以是常数、变量、函数、式子，式中#J、#k也可以为常量，式子右边为变量号、运算式

(1) 定义

#I = #J

(2) 算术运算

#I = #J + #k

#I = #J - #k

#I = #J * #k

#I = #J / #k

(3) 逻辑运算

#I = #J OR #k

#I = #J XOR #k

#I = #J AND #k

(4) 函数运算

#I = SIN [#J] 正弦

#I = COS [#J] 余弦

#I = TAN [#J] 正切

#I = ATAN [#J] 反正切

#I = SQRT [#J] 平方根

#I = ABS [#J] 绝对值

#I = ROUND [#J] 四舍五入化整

#I = FIX [#J] 上取整

#I = FUP [#J] 下取整

#I = BIN [#J] BCD→BIN (二进制)

#I = BCN [#J] BIN→BCD

说明：

① 角度单位为度。

例：90度30分为90.5°。

② ATAN 函数后的两个边长要用“1”隔开。

例：#1=ATAN 1 / [-1]时，#1 为了 135.0。

③ ROUND 用于语句中的地址，按各地址的最小设定单位进行四舍五入。

例：设#1=1.2345，#2=2.3456，设定单位1μm

G91 X - #1; X - 1.235

X - #2 F300; X - 2.346

X [#1 + #2] ; X3.580

未返回原处，应改为 X ROUND [#1] + ROUND [#2]。

④ 取整后的绝对值比原值大为上取整，反之为下取整。

例：设#1=1.2，#2=-1.2 时

若 #3=FUP#1 时, 则 #3=2.0

若 #3=FIX#1 时, 则 #3=1.0

若 #3=FUP#2 时, 则 #3=-2.0

若 #3=FIX#2 时, 则 #3=-1.0

⑤ 指令函数时, 可只写开头 2 个字母。

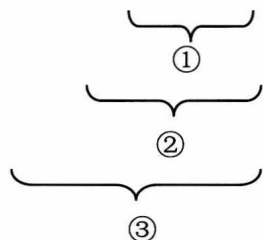
例: ROUND→RO

FIX→FI

⑥ 优先级

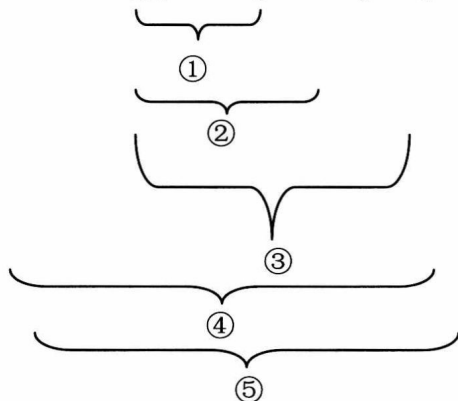
函数→乘除 (*, /, AND)→加减 (+, -, OR, XOR)

例: #1=#2+#3*SIN[#4];



⑦ 括号为中括号, 最多 5 重, 圆括号用于注释语句。

例: #1=SIN [[[#2+#3]*#4+#5]*#6]; (3 重)



1.1.4 转移与循环指令

(1) 无条件的转移

格式: GOTO 1;

GOTO #10;

(2) 条件转移

格式: IF[<条件式>] GOTO n

条件式:

#jEQ#k 表示 =

#jNE#k 表示 ≠

#jGT#k 表示 >

#jLT#k 表示 <