

数控宏程序编程方法、技巧与实例

第2版

冯志刚 主编

涉及华中、西门子、FANUC三种数控系统

详解车削和铣削数控宏程序编程方法、技巧

提供典型实例的工艺分析和编程设计

实例源自技能等级和数控竞赛实操试题



机械工业出版社
CHINA MACHINE PRESS



数控宏程序编程方法、 技巧与实例

-60

第 2 版

冯志刚 主编



机械工业出版社

本书围绕当前常见的 HNC—21/22M 华中世纪星、SINUMERIK 802D 和 FANUC 0i 三种数控系统的宏程序编程设计展开。全书共分 4 章,第 1 章介绍 HNC—21/22M 华中世纪星、SINUMERIK 802D 和 FANUC 0i 三种数控系统的用户宏程序基础理论知识及宏指令调用格式和特点;第 2 章介绍在数控车床上对典型零件和非圆曲线零件进行宏程序编程的技巧和编程实例;第 3 章介绍在数控铣床和加工中心上对典型零件和非圆曲线零件进行宏程序编程的技巧和大量的编程实例;第 4 章详细讲述了实例零件的工艺分析和程序设计。

本书是一本实用性非常强的数控技术用书,特别适合高技能数控人才使用,并可供数控行业的工程技术人员、从事数控加工编程及操作的人员参考,也可供各类大中专院校、技工学校机电一体化专业、数控专业及相关专业的师生使用。本书可作为各类竞赛和国家职业技能鉴定数控高级工、数控技师、高级技师的参考书。

图书在版编目 (CIP) 数据

数控宏程序编程方法、技巧与实例/冯志刚主编. —2 版.

—北京:机械工业出版社,2011.3

ISBN 978-7-111-33271-8

I. ①数… II. ①冯… III. ①数控机床—程序设计 IV. ①TG659

中国版本图书馆 CIP 数据核字(2011)第 017025 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策划编辑:周国萍 责任编辑:周国萍 责任校对:任秀丽

封面设计:路恩中 责任印制:杨 曦

北京京丰印刷厂印刷

2011 年 5 月第 2 版·第 1 次印刷

184mm×260mm·27.25 印张·672 千字

0 001—4 000 册

标准书号:ISBN 978-7-111-33271-8

定价:53.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

策划编辑:(010) 88379733

社服务中心:(010) 88361066

网络服务

销售一部:(010) 68326294

门户网:<http://www.cmpbook.com>

销售二部:(010) 88379649

教材网:<http://www.cmpedu.com>

读者购书热线:(010) 88379203

封面无防伪标均为盗版

中国版本图书馆 CIP 数据核字(2011)第 017025 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策划编辑:周国萍 责任编辑:周国萍 责任校对:任秀丽

封面设计:路恩中 责任印制:杨 曦

北京京丰印刷厂印刷

2011 年 5 月第 2 版·第 1 次印刷

184mm×260mm·27.25 印张·672 千字

0 001—4 000 册

标准书号:ISBN 978-7-111-33271-8

定价:53.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

策划编辑:(010) 88379733

销 售 二 部:(010) 88379649

教材网:<http://www.cmpedu.com>

读者购书热线:(010) 88379203

封面无防伪标均为盗版

第 1 版前言

数控编程作为数控加工的关键技术之一，其程序的编制效率和质量在很大程度上决定了产品的加工精度和生产率。尤其是随着数控加工不断朝高速、精密方向的发展，提高数控程序的编制质量和效率对于提高制造企业的竞争力有着重要的意义。随着 CAD/CAM 软件的不断普及应用，数控编程的模式逐渐由自动编程取代手工编程。但 CAM 软件编程和手工编程有着各自的特长，且现有的 CAM 软件不能满足所有数控系统的特殊功能。因此，充分结合这两种编程模式，对于提高编程的效率和質量有着重要的意义。

目前，国内企业使用的数控机床大多是普通数控机床和少量的高精密数控机床。对于传统的普通数控机床，存在着内存容量有限，无法实现高速切削加工等缺陷；因此，最大限度地发挥普通数控机床的加工效率，编制简洁合理的小容量数控程序，使加工程序变得简洁，对现实加工来说，有着很重要的实际意义。

在手工编制数控机床程序中，经常把能完成某一功能的一系列指令像子程序那样存入存储器，用一个总指令来代表它们，使用时只需给出这个总指令就能执行其功能。所存入的一系列指令称作用户宏功能主体，这个总指令称作用户宏功能指令。

在编程时，不必记住用户宏功能主体所含的具体指令，只要记住用户宏功能指令即可。用户宏功能的最大特点是在用户宏功能主体中能够使用变量；变量之间还能够进行运算；用户宏功能指令可以把实际值设定为变量，使用户宏功能更具通用性。可见，用户宏功能是提高数控机床性能的一种特殊功能。宏功能主体既可由机床生产厂提供，也可由机床用户自己编制（见编程实例）。使用时，先将用户宏主体像子程序一样存放到内存里，然后用子程序调用指令 M98 调用。

本书作者通过实例介绍了数控铣削加工编程中运用宏程序编程，可以充分提高数控编程的效率。希望能为从事数控加工与编程的读者提供借鉴。

目 录

第 2 版前言

第 1 版前言

第 1 章 用户宏程序	1
1.1 HNC—21/22M 华中世纪星数控系统宏指令编程	1
1.1.1 宏变量及常量	1
1.1.2 运算符与函数	4
1.1.3 语句表达式	4
1.1.4 调用方式	5
1.1.5 用户宏程序的结构及用户宏功能	6
1.2 SINUMERIK 802D 数控系统 R 参数指令编程	8
1.2.1 计算参数 R	8
1.2.2 程序跳转	9
1.2.3 子程序	12
1.2.4 R 参数编程的结构及 R 参数功能	13
1.3 FANUC Oi—MC 数控系统用户宏程序	15
1.3.1 变量	15
1.3.2 系统变量	17
1.3.3 算术和逻辑运算	23
1.3.4 宏程序语句和 NC 语句	27
1.3.5 转移和循环	28
1.3.6 宏程序调用	30
1.3.7 用户宏程序的结构及用户宏功能	40
第 2 章 数控车床的宏程序编程	42
2.1 数控车床宏程序编程特征	42
2.1.1 在宏程序主体中使用变量	42
2.1.2 变量之间的演算	43
2.1.3 用宏程序命令对变量进行赋值	46
2.2 数控车床宏程序编程技巧	46
2.2.1 用宏程序和 R 参数编程实现规格不同的轴加工	46
2.2.2 用宏程序和 R 参数编程实现螺纹的粗、精加工	54
2.2.3 用宏程序和 R 参数编制孔加工钻削循环	57
2.3 非圆锥曲线类零件数控车削的宏程序编程实例	60
2.3.1 椭圆类零件的宏程序和 R 参数编程	60
2.3.2 双曲线过渡类零件的宏程序和 R 参数编程	67
2.3.3 抛物线类零件的宏程序和 R 参数编程	74
2.4 数控车削典型曲面零件的宏程序编程实例分析	80
第 3 章 数控铣床、加工中心的宏程序编程	100

3.1 数控铣床、加工中心宏程序编程特征	100
3.1.1 在宏程序主体中使用变量	100
3.1.2 变量之间的演算	101
3.1.3 用宏程序命令对变量进行赋值	105
3.2 数控铣床、加工中心宏（参数）程序编程技巧	108
3.2.1 根据不同类型的零件进行程序设计及加工方法的选择	108
3.2.2 设计程序流程结构框图	110
3.2.3 合理选择图形的数学处理方法	110
3.2.4 非圆曲线轮廓零件编程实例	115
3.3 数控铣床、加工中心宏程序编程实例	120
3.3.1 零件平面铣削宏程序编程实例	120
3.3.1.1 长方形零件平面同向铣削宏程序编程	120
3.3.1.2 长方形零件平面双向铣削宏程序编程	127
3.3.1.3 圆形零件平面的双向铣削宏程序编程	135
3.3.2 孔系类零件宏程序编程实例	145
3.3.2.1 直线点阵孔群钻削宏程序编程	145
3.3.2.2 矩形框式点阵孔群宏程序编程	150
3.3.2.3 平行四边形框式点阵孔群宏程序编程	158
3.3.2.4 矩形网式点阵孔群宏程序编程	167
3.3.2.5 平行四边形网式点阵孔群宏程序编程	174
3.3.2.6 圆弧点阵孔群宏程序编程	182
3.3.2.7 圆环形点阵孔群宏程序编程	187
3.3.2.8 交错排列的网格点阵孔群宏程序编程	192
3.3.3 外轮廓侧面铣削宏程序编程实例	199
3.3.3.1 圆形零件外轮廓侧面铣削宏程序编程	199
3.3.3.2 长方形外轮廓侧面铣削宏程序编程	204
3.3.3.3 跑道形外轮廓侧面铣削宏程序编程	211
3.3.4 凹槽类零件侧面铣削宏程序编程实例	218
3.3.4.1 圆形凹槽类零件侧面铣削宏程序编程	218
3.3.4.2 方形凹槽类零件侧面铣削宏程序编程	224
3.3.4.3 跑道形凹槽类零件侧面铣削宏程序编程	231
3.3.5 锥台类零件侧面铣削宏程序编程实例	243
3.3.5.1 圆形锥台类零件侧面铣削宏程序编程	243
3.3.5.2 正四棱锥台类零件侧面铣削宏程序编程	251
3.3.5.3 正多棱锥台类零件侧面铣削宏程序编程	259
3.3.6 锥槽类零件侧面铣削宏程序编程实例	267
3.3.6.1 圆锥槽侧面铣削宏程序编程	267
3.3.6.2 四方锥槽类零件侧面铣削宏程序编程	277
3.3.6.3 跑道形锥槽类零件侧面铣削宏程序编程	290
3.3.7 非圆锥曲线类零件的宏程序编程实例	303
3.3.7.1 椭圆类零件曲面的宏程序编程	303
3.3.7.2 双曲线类零件曲面的宏程序编程	311
3.3.7.3 抛物线类零件曲面的宏程序编程	319

3.3.7.4 阿基米德螺旋线类零件曲面的宏程序编程	329
3.3.8 球面类零件的宏程序编程实例	337
3.3.8.1 凸半球面零件类的宏程序编程	337
3.3.8.2 凹半球面零件类的宏程序编程	343
3.3.8.3 相邻面圆角过渡类零件的宏程序编程	352
3.4 数控铣削典型曲面零件的宏程序编程实例分析	363
第4章 典型曲面零件宏程序编程实例分析与加工	399
4.1 烟灰缸的宏程序编程与加工	399
4.2 五角星的宏程序编程与加工	411
4.3 快餐饭盒凹模的宏程序编程与加工	417
参考文献	426

第 1 章 用户宏程序

1.1 HNC—21/22M 华中世纪星数控系统宏指令编程

HNC—21/22M 华中世纪星数控系统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、逻辑运算和函数的混合运算。此外，宏程序还提供了循环语句、分支语句和子程序调用语句，利于编制各种复杂的零件加工程序，减少乃至免除手工编程时进行繁琐的数值计算，以及精简程序量。

1.1.1 宏变量及常量

1. 宏变量

在常规的主程序和子程序内，总是将一个具体的数值赋给一个地址。为了使程序更具通用性、更加灵活，在宏程序中设置了变量。

(1) 变量的表示 变量可以用“#”号和紧跟其后的变量序号来表示：#i (i = 1, 2, 3...)。

例如：#5, #109, #501。

(2) 变量的引用 将跟随在一个地址后的数值用一个变量来代替，即引入了变量。

例如：对于 F [#103]，若 #103 = 50 时，则为 F50；

对于 Z [-#110]，若 #110 = 100 时，则为 Z - 100；

对于 G [#130]，若 #130 = 3 时，则为 G03。

(3) 变量的类型 华中数控系统的变量分为公共变量和系统变量两类。

1) 公共变量：公共变量又分为全局变量和局部变量。全局变量是在主程序和主程序调用的各用户宏程序内都有效的变量。也就是说，在一个宏指令中的#i 与在另一个宏指令中的#i 是相同的。局部变量仅在主程序和当前用户宏程序内有效。也就是说，在一个宏指令中的#i 与在另一个宏指令中的#i 是不一定相同的。

全局变量的序号为：#50 ~ #199

当前局部变量的序号为：#0 ~ #49

0 层局部变量的序号为：#200 ~ #249

1 层局部变量的序号为：#250 ~ #299

2 层局部变量的序号为：#300 ~ #349

3 层局部变量的序号为：#350 ~ #399

4 层局部变量的序号为：#400 ~ #449

5 层局部变量的序号为：#450 ~ #499

6 层局部变量的序号为：#500 ~ #549

7 层局部变量的序号为：#550 ~ #599

华中数控系统可以子程序嵌套调用，调用的深度最多可以有九层。每一层子程序都有自己独立的局部变量，变量个数为 50。如当前局部变量为 #0 ~ #49；第一层局部变量为 #200 ~ #249；第二层局部变量为 #250 ~ #299；第三层局部变量为 #300 ~ #349；依此类推。

2) 系统变量：系统变量定义为有固定用途的变量，它的值决定系统的状态。系统变量包括刀具偏置变量、接口的输入/输出信号变量、位置信号变量等。

例如：#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	#1115 G28 中间点位置 C	#1116 G28 中间点位置 U
#1117 G28 中间点位置 V	#1118 G28 中间点位置 W	#1119 G28 屏蔽字
#1120 镜像点位置 X	#1121 镜像点位置 Y	#1122 镜像点位置 Z
#1123 镜像点位置 A	#1124 镜像点位置 B	#1125 镜像点位置 C
#1126 镜像点位置 U	#1127 镜像点位置 V	#1128 镜像点位置 W
#1129 镜像屏蔽字	#1130 旋转中心 (轴 1)	#1131 旋转中心 (轴 2)
#1132 旋转角度	#1133 旋转轴屏蔽字	#1134 保留
#1135 缩放中心 (轴 1)	#1136 缩放中心 (轴 2)	#1137 缩放中心 (轴 3)
#1138 缩放比例	#1139 缩放轴屏蔽字	#1140 坐标变换代码 1
#1141 坐标变换代码 2	#1142 坐标变换代码 3	#1143 保留
#1144 刀具长度补偿号	#1145 刀具半径补偿号	#1146 当前平面轴 1
#1147 当前平面轴 2	#1148 虚拟轴屏蔽字	#1149 进给速度指定
#1150 G 代码模态值 0	#1151 G 代码模态值 1	#1152 G 代码模态值 2
#1153 G 代码模态值 3	#1154 G 代码模态值 4	#1155 G 代码模态值 5
#1156 G 代码模态值 6	#1157 G 代码模态值 7	#1158 G 代码模态值 8
#1159 G 代码模态值 9	#1160 G 代码模态值 10	#1161 G 代码模态值 11
#1162 G 代码模态值 12	#1163 G 代码模态值 13	#1164 G 代码模态值 14
#1165 G 代码模态值 15	#1166 G 代码模态值 16	#1167 G 代码模态值 17
#1168 G 代码模态值 18	#1169 G 代码模态值 19	#1170 剩余 CACHE
#1171 备用 CACHE	#1172 剩余缓冲区	#1173 备用缓冲区
#1174 保留	#1175 保留	#1176 保留
#1177 保留	#1178 保留	#1179 保留
#1180 保留	#1181 保留	#1182 保留
#1183 保留	#1184 保留	#1185 保留
#1186 保留	#1187 保留	#1188 保留
#1189 保留	#1190 用户自定义输入	#1191 用户自定义输出
#1192 自定义输出屏蔽	#1193 保留	#1194 保留

2. 常量

类似于高级编程语言中的常量，在用户宏程序中也具有常量。在华中数控系统中的常量主要有三个：

PI: 圆周率

TRUE: 条件成立 (真)

FALSE: 条件不成立 (假)

1.1.2 运算符与函数

宏程序中的各运算符、函数将实现丰富的宏功能。华中数控系统的运算符有:

1. 算术运算符

+、-、*、/。

2. 条件运算符

EQ (=)、NE (≠)、GT (>)、GE (≥)、LT (<)、LE (≤)。

3. 逻辑运算符

AND (与)、OR (或)、NOT (非)。

4. 函数

SIN (正弦)、COS (余弦)、TAN (正切)、ATAN (反正切 -90° ~ 90°)、ABS (绝对值)、INT (取整)、ATAN2 (反正切 -180° ~ 180°)、SIGN (取符号)、SQRT (平方根)、EXP (指数)。

1.1.3 语句表达式

用运算符连接起来的常数、宏变量构成表达式。

例如: 175/SQRT [2] * COS [55 * PI/180];

#3 * 6 GT 14;

华中数控系统的语句表达式有三种:

1. 赋值语句

格式: 宏变量 = 常数或表达式

把常数或表达式的值送给一个宏变量称为赋值。

例如: #2 = 175/SQRT [2] * COS [55 * PI/180];

#3 = 124.0;

2. 条件判别语句 IF ELSE ENDIF

格式 (i): IF 条件表达式

...条件成立 (真)

ELSE

...条件不成立 (假)

ENDIF

格式 (ii): IF 条件表达式

...条件成立 (真)

ENDIF

3. 循环语句 WHILE ENDW

格式: WHILE 条件表达式

...条件成立 (真)

ENDW

条件判别语句的使用参见第3章宏程序编程举例。

循环语句的使用参见 1.1.5 节中宏程序编程举例。

1.1.4 调用方式

G 代码在调用宏（子程序或固定循环，下同）时，为保存当前主程序的编程信息，系统会将当前程序段各字段（A ~ Z 共 26 个字段，如果没有定义则为零）的内容复制到宏执行时的局部变量#0 ~ #25，同时复制调用宏时当前通道九个轴的绝对位置（机床绝对坐标）到宏执行时的局部变量#30 ~ #38。

调用一般子程序时，不保存系统模态值，即子程序可修改系统模态，并保持有效；而调用固定循环时，保存系统模态值，即固定循环子程序不修改系统模态参数。

表 1-1 列出了宏当前局部变量#0 ~ #38 所对应的宏调用者传递的字段参数名。

宏程序的调用方式类似于调用子程序，即同样采用 M98 调用，采用 M99 结束。但在宏程序时，应给出所需要的参数值，例如有一个逼近整圆的数控加工程序，在程序中把加工整圆作为宏程序进行调用，在调用时要给出所要求的圆心点和圆半径，见表 1-2。

表 1-1 当前局部变量表

宏当前 局部变量	宏调用时所传递的字段 名或系统变量	宏当前 局部变量	宏调用时所传递的字段 名或系统变量
#0	A	#20	U
#1	B	#21	V
#2	C	#22	W
#3	D	#23	X
#4	E	#24	Y
#5	F	#25	Z
#6	G	#26	固定循环指令初始平面 Z 模态值
#7	H	#27	不用
#8	I	#28	不用
#9	J	#29	不用
#10	K	#30	调用子程序时轴 0 的绝对坐标
#11	L	#31	调用子程序时轴 1 的绝对坐标
#12	M	#32	调用子程序时轴 2 的绝对坐标
#13	N	#33	调用子程序时轴 3 的绝对坐标
#14	O	#34	调用子程序时轴 4 的绝对坐标
#15	P	#35	调用子程序时轴 5 的绝对坐标
#16	Q	#36	调用子程序时轴 6 的绝对坐标
#17	R	#37	调用子程序时轴 7 的绝对坐标
#18	S	#38	调用子程序时轴 8 的绝对坐标
#19	T		

表 1-2 圆的宏程序调用

程 序	说 明
O1234	文件名
%1000	程序号
G54 G00 X0 Y0 Z0	工件零点偏置
M98 P2 X-50 Y0 R50	调用加工整圆的宏程序，并给出圆心点和圆半径
M30	程序结束
%2	加工整圆的宏程序
.....	
M99	宏程序结束，返回主程序

1.1.5 用户宏程序的结构及用户宏功能

HNC—21/22M 华中世纪星数控系统用户宏程序由文件名（O××××）、宏程序号（%××××）、宏程序主体和宏程序指令组成。宏程序结束指令用 M30，结束宏程序的同时，返回主程序开头，同时将本宏程序内所用的局部变量清零。

变量、演算式和转向语句的使用是用户宏功能的核心。它们既可以用在宏程序中，也可以用在主程序中。所以从广义上说，程序使用变量的功能就可以称为用户宏功能，有没有宏程序都是如此。下面举用户宏功能的一个应用实例。

编程举例：加工图 1-1 所示零件，铣削圆台与斜方台，各自加工 3 个循环，要求倾斜 10° 的斜方台与圆台相切，圆台在方台之上。试用变量、演算式和转向语句来编写程序。

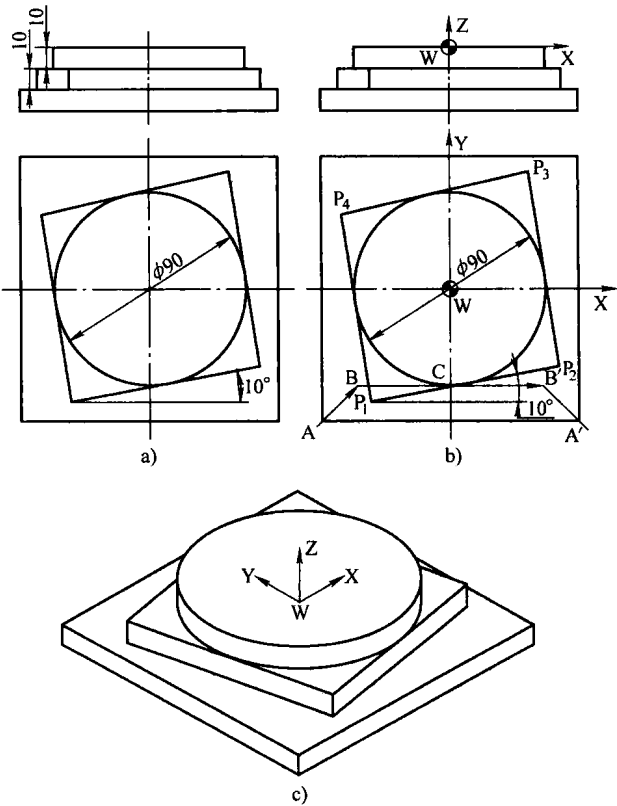


图 1-1 宏程序编制示例图
a) 零件尺寸图 b) 刀具路线图 c) 效果图

O × × × × ; 文件名

% 10 ; 程序号

N005 #10 = 10.0 ; 圆台阶高度

#11 = 10.0 ; 方台阶高度

#12 = 124.0 ; 圆外定点的 X 坐标值

#13 = 124.0 ; 圆外定点的 Y 坐标值

#101 = 12.0 ; 刀具半径(偏大,粗加工)

#102 = 6.5 ; 刀具半径(偏中,半精加工)

#103 = 6.0 ; 刀具半径(实际,精加工)

N010 T01 M06 ; 调用 1 号刀具

N015 G17 G90 G21 G94 G54 G40 G49 G80 ; 工艺加工初始状态设置

N020 G43 G00 Z50.0 H01 S650 M03 ; 建立刀具长度补偿,主轴正转,转速为
650r/min

N025 X0 Y0 M07 ; 刀具快速移动到工件坐标零点,打开切削液

N030 #0 = 0 ; 计数器置初始值

N035 G00 X[- #12] Y[- #13] ; 刀具快速插补至圆外定点 A,坐标为(- 124.0 ,
- 124.0)

N040 Z[- #10] ; 刀具快速下降至工件表面下方 10mm 处,准备加工圆台

N045 WHILE #0 LT 3 ; 如果#0 大于或等于 3,跳转到 N090 程序段(加工圆台循环)

N050 G42 G01 X[- #12/2] Y[- 90/2] F150.0 D[#0 + 101]

; 右补偿,直线插补到圆台切线 B 点上

N055 X0 Y[- 90/2] ; 直线插补到圆台直径切点上 C 点

N060 G03 J[90/2] ; 逆时针整圆插补

N065 G01 X[#12/2] Y[- 90/2] ; 沿圆台直径切点切线切出到 B' 点

N070 G40 X[#12] Y[- #13] ; 取消刀具半径补偿,直线插补到 A' 点

N075 G00 X[- #12] Y[- #13] ; 刀具快速返回到程序起点 A 点

N080 #0 = #0 + 1 ; 计数器加 1,返回 N045 程序段

N085 ENDW ; 循环结束

N090 Z[- #10 - #11] ; 刀具快速下降至斜方台最终加工深度

N095 #2 = 45 * SQRT[2] * COS[55 * PI/180] ; 斜方台左下角 P₁ 点横坐标计算

#3 = 45 * SQRT[2] * SIN[55 * PI/180] ; 斜方台左下角 P₁ 点纵坐标计算

#4 = 90 * COS[10 * PI/180] ; 斜方台右下角 P₂ 点离开左下角 P₁ 点横向相对值计算

#5 = 90 * SIN[10 * PI/180] ; 斜方台右下角 P₂ 点离开左下角 P₁ 点纵向相对值计算

#0 = 0 ; 计数器置初始值

N100 WHILE #0 LT 3 ; (如果#0 大于或等于 3,跳转到 N145 程序段)进入加工斜方台循环

N105 G90 G42 G01 X[- #2] Y[- #3] F150.0 D[#0 + 101]

; 右补偿,直线插补到斜方台左下角 P₁ 点

N110 G91 X[#4] Y[#5] ; 运用相对坐标指令,直线插补到斜方台右下角 P₂ 点

N115 X[- #5] Y[#4] ; 直线插补到斜方台右上角 P₃ 点

N120 X[-#4] Y[-#5];直线插补到斜方台左上角 P₄ 点
 N125 X[#5] Y[-#4];直线插补到斜方台左下角 P₁ 点
 N130 G90 G40 G00 X[-#12] Y[-#13];运用绝对坐标指令,取消刀具半径补偿,刀具快速运行到程序起点 A 点
 N135 #0 = #0 + 1;计数器加 1,返回 N100 程序段
 N140 ENDW;循环结束
 N145 G00 Z100.0 M09;刀具退到工件上表面 100mm 处,切削液关闭
 N150 G49;取消刀具长度补偿
 N155 G00 X0 Y0 M05;刀具退回工件坐标零点,主轴停止
 N160 M30;程序结束并返回程序开头

1.2 SINUMERIK 802D 数控系统 R 参数指令编程

1.2.1 计算参数 R

要使一个 NC 程序不仅仅适用于特定数值下的一次加工,或者必须要计算出数值,这两种情况均可以使用计算参数。在程序运行时,可以由控制器计算或设定所需要的数值,也可以通过操作面板设定参数数值。如果参数已经赋值,则它们可以在程序中对由变量确定的地址进行赋值。

编程的计算参数从 R0 ~ R299,共 300 个计算参数。其中,R0 ~ R99 在编程时可以自由使用;R100 ~ R249 主要用于加工循环传递参数;R250 ~ R299 用于加工循环的内部计算参数。如果编程时没有用到加工循环,R100 ~ R249 同样可以在编程时自由使用。

1. 赋值

1) 可以在以下数值范围内给计算参数赋值:

$$\pm (0.0000001 \sim 99999999) \text{ (8 位,带符号和小数点)}$$

在取整数数值时,可以去除小数点,正号可以省去。

编程举例 1:

$$R0 = 3.5678 \quad R1 = -37.3 \quad R2 = 2 \quad R3 = -7 \quad R4 = -45678.1234$$

2) 用指数表示法可以赋值更大的数值范围:

$$\pm (10^{-300} \sim 10^{+300})$$

指数值写在 EX 符号之后;最大符号数:10 个(包括符号和小数点);EX 值范围: -300 ~ +300。

编程举例 2:

$$R0 = -0.1EX - 5; \text{表示 } R0 \text{ 等于 } -0.0000001$$

$$R1 = 1.874EX8; \text{表示 } R1 \text{ 等于 } 187400000$$

注释:一个程序段中可以有多个赋值语句,也可以用计算表达式赋值。

3) 给其他的地址赋值:通过给其他的 NC 地址分配计算参数或参数表达式,可以增加 NC 程序的通用性。可以用数值、算术表达式或 R 参数对任意 NC 地址赋值,但对地址 N、G 和 L 例外。

赋值时在地址符之后需写入符号“=”；赋值语句也可以赋值一负号。给坐标轴地址（运行指令）赋值时，要求有一独立的程序段。

编程举例 3：

N10 G00 X = R2 ; 给 X 值赋值

2. 参数的计算

在计算参数时也遵循通常的运算规则。圆括号内的运算优先进行。另外，乘法和除法运算优先于加法和减法运算。

角度计算单位为度（°）。

编程举例 4：R 参数编程

N10 R1 = R1 + 1 ; 由原来的 R1 加上 1 后得到新的 R1

N20 R1 = R1 + R2 R2 = R5 - R6

R7 = R8 * R9 R10 = R11/R12;

N30 R13 = SIN (25.3) ; R13 等于正弦 25.3°

N40 R14 = R1 * R2 + R3 ; 乘法和除法运算优先于加法和减法运算

N50 R14 = R3 + R2 * R1 ; 与 N40 一样

N60 R15 = SQRT (R1 * R1 + R2 * R2) ; 表示 R15 等于 R1 的平方与 R2 的平方和开平方根

N70 ...

编程举例 5：坐标值赋值

N10 G01 G91 X = R1 Y = R2 F300 ;

N20 Z = R3 ;

N30 X = - R4 ;

N40 Z = - R5 ;

N50 ... ;

1.2.2 程序跳转

1. 标记符——程序跳转目标

标记符用于标记程序中所跳转的目标程序段，用跳转功能可以实现程序运行分支。标记符可以自由选取，但必须由 2~8 个字母或数字组成，其中开始两个符号必须是字母或下划线。

跳转目标程序段中标记符后面必须为冒号。标记符位于程序段段首。如果程序段有段号，则标记符紧跟着段号。

在一个程序中，标记符不能含有其他意义。

程序举例：

N10 MARKE1: G01 X20 ; MARKE1 为标记符，跳转目标程序段

:

TR789: G00 X10 Z20 ; TR789 为标记符，跳转目标程序段没有段号

N100 ... ; 程序段号可以是跳转目标

2. 绝对跳转

功能：NC 程序在运行时以写入时的顺序执行程序段。程序在运行时可以通过插入程序