

SINUMERIK

SINUMERIK 840D sl / 828D 扩展功能

功能手册

适用于

控制系统 SINUMERIK 840D sl/840DE sl SINUMERIK
828D

软件
4.5 SP2

CNC 软件版本

03/2013
6FC5397-1BP40-3RA1

前言

A4: SINUMERIK 840D sl 的 NCK 数字量/模拟量输入/输出	1
B3: 分布式系统 - 仅适用于 840D sl	2
H1: 手动运行和手轮运行	3
K3: 补偿	4
K5: 运行方式组、通道、跨通 道取轴	5
M1: 坐标转换	6
M5: 测量	7
N3: 软件挡块, 行程开关信 号 - 仅适用于 840D sl	8
N4: 冲裁和步冲 - 仅适用于 840D sl	9
P2: 定位轴	10
P5: 往复运动 - 仅限 840D sl	11
R2: 回转轴	12
S3: 主轴同步	13
S7: 存储器配置	14
T1: 分度轴	15
W3: 换刀	16
W4: 磨削专用的刀具补偿和监 控	17
Z2: NC/PLC 接口信号	18
附录	A

法律资讯

警告提示系统

为了您的人身安全以及避免财产损失，必须注意本手册中的提示。人身安全的提示用一个警告三角表示，仅与财产损失有关的提示不带警告三角。警告提示根据危险等级由高到低如下表示。

 危险
表示如果不采取相应的小心措施， 将会 导致死亡或者严重的人身伤害。

 警告
表示如果不采取相应的小心措施， 可能 导致死亡或者严重的人身伤害。

 小心
表示如果不采取相应的小心措施，可能导致轻微的人身伤害。

注意
表示如果不采取相应的小心措施，可能导致财产损失。

当出现多个危险等级的情况下，每次总是使用最高等级的警告提示。如果在某个警告提示中带有警告可能导致人身伤害的警告三角，则可能在该警告提示中另外还附带有可能导致财产损失的警告。

合格的专业人员

本文件所属的产品/系统只允许由符合各项工作要求的**合格人员**进行操作。其操作必须遵照各自附带的文件说明，特别是其中的安全及警告提示。由于具备相关培训及经验，合格人员可以察觉本产品/系统的风险，并避免可能的危险。

按规定使用 Siemens 产品

请注意下列说明：

 警告
Siemens 产品只允许用于目录和相关技术文件中规定的使用情况。如果要使用其他公司的产品和组件，必须得到 Siemens 推荐和允许。正确的运输、储存、组装、装配、安装、调试、操作和维护是产品安全、正常运行的前提。必须保证允许的环境条件。必须注意相关文件中的提示。

商标

所有带有标记符号 ® 的都是西门子股份有限公司的注册商标。本印刷品中的其他符号可能是一些其他商标。若第三方出于自身目的使用这些商标，将侵害其所有者的权利。

责任免除

我们已对印刷品中所述内容与硬件和软件的一致性作过检查。然而不排除存在偏差的可能性，因此我们不保证印刷品中所述内容与硬件和软件完全一致。印刷品中的数据都按规定经过检测，必要的修正值包含在下一版本中。

前言

SINUMERIK 文献

SINUMERIK 文档分为以下几个类别：

- 一般文献
- 用户文献
- 制造商/维修文档

其它信息

访问链接 www.siemens.com/motioncontrol/docu 可获取关于以下主题的信息：

- 订购文档/查看印刷品一览
- 进入下载文档的链接
- 使用在线文档（查找搜索手册/信息）

如果您对技术文档有疑问（例如：建议或修改），请发送一份电子邮件到下列地址：

docu.motioncontrol@siemens.com

我的文档管理器（MDM）

点击下面的链接，您可以在西门子文档内容的基础上创建自己的机床文档。

www.siemens.com/mdm

培训

提供的培训课程有：

- www.siemens.com/sitrain
SITRAIN - 西门子自动化产品、系统以及解决方案的培训
- www.siemens.com/sinutrain
SinuTrain - SINUMERIK 培训软件

常见问题

常见问题（FAQ）请点击“产品支持”，然后点击右侧的“支持”。<http://support.automation.siemens.com>

SINUMERIK

SINUMERIK 的信息请点击：
www.siemens.com/sinumerik

目标使用人群

本手册供以下人员使用：

- 设计人员
- 机床制造商的技术人员
- 控制系统/机床的调试人员
- 编程人员

用途

本功能手册旨在介绍各项系统功能，使相关人员可以了解、选择功能，并掌握功能的调试步骤。

标准功能范畴

在本文档中描述了标准功能范畴。机床制造商增添或者更改的功能，由机床制造商资料进行说明。

控制系统有可能执行本文档中未描述的某些功能。但这并不表示在交付系统时必须提供这些功能以及相关的维修服务。

同样，因为只是概要，所以该文档不包括全部类型产品的所有详细信息，也无法考虑到安装、运行和维修中可能出现的所有情况。

技术支持

各个国家的技术支持电话请访问以下网址 <http://www.siemens.com/automation/service&support>

结构与内容的相关信息

结构

本手册由以下部分组成：

- 扉页（第 3 页），包含了本功能手册的标题、SINUMERIK 控制系统名称、系统软件、本手册版本针对的软件版本以及各功能说明一览。
- 功能说明，按字母和数字排序（例如 A2、A3、B1 等）。
- 附录，包含：
 - 缩写一览表
 - 手册一览
- 术语一览

说明

详细的数据和报警说明请见：

- 机床数据和设定数据：
详细的机床数据说明（仅提供电子版本，通过 DOConCD 或 DOConWEB 获取）
 - NC/PLC 接口信号：
 - 功能手册“基本功能”；NC/PLC 接口信号（Z1）
 - 功能手册“扩展功能”；NC/PLC 接口信号（Z2）
 - 功能手册“特殊功能”；NC/PLC 接口信号（Z3）
 - 报警：
诊断手册
-

系统数据的表示方法

本文档中，系统数据采用以下表示方法：

信号/数据	表示方法	示例
NC/PLC 接口信号	... NC/PLC 接口信号： <信号地址> (<信号名称>)	若切换了齿轮档，则会由 PLC 程序置位以下 NC/PLC 接口信号： DB31, ... DBX16.0-2（实际齿轮档 A 至 C） DB31, ... DBX16.3（齿轮档已切换）
机床数据	... 机床数据： <类型><编号> <完整名称> (<含义>)	主主轴为机床数据中指定的主轴： MD20090 \$MC_SPIND_DEF_MASTER_SPIND（通道中主主轴的缺省设置）
设定数据	... 设定数据： <类型><编号> <完整名称> (<含义>)	逻辑主主轴包含在设定数据中： SD42800 \$SC_SPIND_ASSIGN_TAB[0]（主轴编号转换器）

说明

信号地址

功能说明中指出的 NC/PLC 接口信号的地址仅仅是适用于 SINUMERIK 840D sl 的地址。适用于 SINUMERIK 828D 的信号地址请见相应功能说明末尾处的数据列表“信号发送至/来自...”。

组态范围

NC/PLC 接口的相关说明以下列特性的最大绝对数量为基础：

- 运行方式组（DB11）
- 通道（DB21, ...）
- 加工轴/主轴（DB31, ...）

数据类型

在控制系统中，以下数据类型用于在零件程序中进行编程：

类型	含义	取值范围
INT	整数，带符号	-2.147.483.648 ... +2.147.483.647
REAL	带小数点的实数	$\approx \pm 5,0 \cdot 10^{-324} \dots \approx \pm 1,7 \cdot 10^{+308}$

类型	含义	取值范围
BOOL	逻辑值	TRUE(≠0) , FALSE (0)
CHAR	ASCII 字符或字节	0 ... 255 或 -128 ... 127
STRING	字符串，以零结束	最多 400 个字符 + /0 (无特殊字符)
AXIS	轴名称	所有在控制系统中存在的轴名称
FRAME	用于平移、旋转、比例缩放和镜像的几何参数	---
提示 数组只可由相同的基本数据类型构成。 最多可以为 3 维数组。		

目录

前言.....	3
1 A4: SINUMERIK 840D sl 的 NCK 数字量/模拟量输入/输出.....	31
1.1 引言.....	31
1.2 通过 PLC 访问.....	33
1.2.1 简要说明.....	33
1.2.2 NCK 输入/输出的配置.....	34
1.2.3 NCK 的数字量输入/输出.....	37
1.2.3.1 NCK 的数字量输入.....	37
1.2.3.2 NCK 的数字量输出.....	39
1.2.3.3 NCK 高速数字量输入/输出之间的互联（直连和逻辑运算）.....	42
1.2.4 NCK 的模拟量输入/输出.....	44
1.2.4.1 NCK 的模拟量输入.....	44
1.2.4.2 NCK 的模拟量输出.....	47
1.2.4.3 NCK 模拟量输入/输出值的表示法.....	51
1.2.5 比较器输入.....	53
1.2.6 可由 NC 直接访问的 PLC 输入/输出.....	57
1.2.6.1 功能.....	57
1.2.6.2 边界条件.....	60
1.2.6.3 示例.....	60
1.3 通过 PROFIBUS 访问.....	62
1.3.1 简要说明.....	62
1.3.2 I/O 区域的配置.....	64
1.3.3 数据交换.....	66
1.3.3.1 通过零件程序/同步动作访问.....	66
1.3.3.2 通过编译循环进行读/写.....	69
1.3.4 边界条件.....	71
1.3.5 示例.....	71
1.3.5.1 写访问中的 PROFIBUS I/O.....	71
1.3.5.2 读访问中的 PROFIBUS I/O.....	73
1.3.5.3 “写访问中 PROFIBUS I/O” 中区域下标的查询.....	74
1.4 数据表.....	76
1.4.1 机床数据.....	76
1.4.1.1 通用机床数据.....	76
1.4.1.2 通道专用机床数据.....	77

1.4.2	设定数据.....	77
1.4.2.1	通用设定数据.....	77
1.4.3	信号.....	78
1.4.3.1	发送至 NC 的信号.....	78
1.4.3.2	从 NC 发出的信号.....	78
2	B3: 分布式系统 - 仅适用于 840D sl	79
2.1	简要说明.....	79
2.1.1	多操作面板配备多 NCU (T:M:N)	79
2.1.2	NCU-Link.....	82
2.1.2.1	Link 通讯.....	82
2.1.2.2	Link 变量.....	83
2.1.2.3	Link 轴.....	83
2.1.2.4	Lead-Link 轴.....	84
2.1.2.5	关联性.....	84
2.1.2.6	应用示例: 回转加工中心.....	85
2.2	NCU-Link.....	88
2.2.1	Link 通讯.....	88
2.2.1.1	简介.....	88
2.2.1.2	Link 模块.....	92
2.2.1.3	参数设置: NC 系统周期.....	92
2.2.1.4	参数设置: Link 通讯.....	94
2.2.1.5	配置.....	95
2.2.1.6	NCU 之间的接线.....	95
2.2.1.7	激活.....	95
2.2.2	Link 变量.....	96
2.2.2.1	Link 变量存储器的属性.....	97
2.2.2.2	Link 变量的属性.....	97
2.2.2.3	写单元.....	99
2.2.2.4	写访问的时间特性.....	99
2.2.2.5	系统变量.....	99
2.2.2.6	写任务的同步.....	100
2.2.2.7	示例: Link 变量存储器的划分.....	100
2.2.2.8	示例: 读取驱动数据.....	102
2.2.3	Link 轴.....	103
2.2.3.1	简介.....	103
2.2.3.2	Link 轴的名称.....	105
2.2.3.3	参数设置.....	105
2.2.3.4	主轴上的辅助功能输出.....	106

2.2.3.5	边界条件.....	107
2.2.4	轴容器.....	109
2.2.4.1	简介.....	109
2.2.4.2	参数设置.....	112
2.2.4.3	编程.....	118
2.2.4.4	系统变量.....	120
2.2.4.5	使用轴容器的加工（图示）.....	121
2.2.4.6	各种运行状态下的特性.....	121
2.2.4.7	撤销轴容器旋转使能时的特性.....	122
2.2.4.8	边界条件.....	124
2.2.5	Lead-Link 轴.....	126
2.2.5.1	简介.....	126
2.2.5.2	参数设置.....	128
2.2.5.3	用于设定主值的系统变量.....	129
2.2.5.4	边界条件.....	130
2.2.5.5	示例.....	130
2.2.6	一个 Link 组内的单位制.....	130
2.3	示例.....	131
2.3.1	Link 轴.....	131
2.3.2	轴容器协调.....	133
2.3.2.1	轴容器旋转，零件程序不等待旋转结束.....	133
2.3.2.2	轴容器旋转，零件程序隐形等待旋转结束.....	134
2.3.2.3	仅在一个通道中执行轴容器旋转（例如启动时）.....	134
2.3.3	分析轴容器系统变量.....	134
2.3.3.1	有条件的跳转.....	134
2.3.3.2	通过 \$AN_AXCTSWA 静态同步.....	135
2.3.3.3	安全等待轴容器旋转结束.....	135
2.3.4	多主轴车床的配置.....	136
2.3.5	Lead-Link 轴.....	146
2.3.5.1	配置.....	146
2.3.5.2	编程.....	148
2.4	数据表.....	149
2.4.1	机床数据.....	149
2.4.1.1	通用机床数据.....	149
2.4.1.2	通道专用机床数据.....	150
2.4.1.3	进给轴/主轴专用机床数据.....	150
2.4.2	设定数据.....	150
2.4.2.1	通用设定数据.....	150
2.4.2.2	进给轴/主轴专用设定数据.....	150

2.4.3	信号.....	151
2.4.3.1	从 NC 发出的信号.....	151
2.4.3.2	HMI/PLC 信号.....	151
2.4.3.3	通用在线接口.....	152
2.4.3.4	来自进给轴/主轴的信号.....	154
2.4.4	系统变量.....	154
3	H1: 手动运行和手轮运行.....	155
3.1	引言.....	155
3.1.1	概述.....	155
3.1.2	JOG 运行方式的一般特性.....	155
3.1.3	通过 PLC 接口控制手动运行.....	159
3.2	连续运行 (JOG-CONT).....	160
3.2.1	一般功能.....	160
3.2.2	点按模式/长按模式的区别.....	161
3.2.3	边界条件.....	162
3.3	断续运行 (JOG-INC).....	162
3.3.1	一般功能.....	162
3.3.2	点按模式/长按模式的区别.....	163
3.3.3	边界条件.....	165
3.4	JOG 下的手轮运行.....	165
3.4.1	功能.....	165
3.4.2	设置.....	173
3.4.3	运行请求.....	178
3.4.4	手轮的重复使用.....	181
3.5	自动方式下的手轮叠加.....	183
3.5.1	一般功能.....	183
3.5.2	手轮叠加的编程和激活.....	188
3.5.3	自动方式下的手轮叠加的特殊性.....	190
3.6	轮廓手轮/通过手轮设定行程（选件）.....	191
3.7	DRF 偏移.....	193
3.8	手动运行至固定点.....	197
3.8.1	引言.....	197
3.8.2	功能.....	198
3.8.3	参数设置.....	200
3.8.4	编程.....	203
3.8.5	边界条件.....	203
3.8.6	应用示例.....	204

3.9	沿刀具中心轴的退刀（手动退刀）	205
3.9.1	概述	205
3.9.2	参数设置	206
3.9.3	选择	207
3.9.4	退刀	209
3.9.5	取消	210
3.9.6	重新选择	211
3.9.7	继续程序执行	211
3.9.8	状态图	212
3.9.9	系统数据	213
3.9.10	边界条件	213
3.10	调试：手轮	214
3.10.1	简介	214
3.10.2	通过 PPU 连接（仅针对 828D）	215
3.10.3	通过 PROFIBUS 连接 (828D)	216
3.10.4	通过 PROFIBUS 连接 (840D sl)	218
3.10.5	通过以太网连接（仅针对 840D sl）	221
3.11	手动运行和手轮运行时的特点	225
3.11.1	几何轴/定向轴的手动运行	225
3.11.2	主轴的手动运行	227
3.11.3	监控	229
3.11.4	其它	230
3.12	数据表	231
3.12.1	机床数据	231
3.12.1.1	通用机床数据	231
3.12.1.2	通道专用机床数据	232
3.12.1.3	进给轴/主轴专用机床数据	233
3.12.2	设定数据	234
3.12.2.1	通用设定数据	234
3.12.3	信号	234
3.12.3.1	从 NC 发出的信号	234
3.12.3.2	发送至 BAG 的信号	234
3.12.3.3	从 BAG 发出的信号	235
3.12.3.4	发送至通道的信号	235
3.12.3.5	来自通道的信号	237
3.12.3.6	到进给轴/主轴的信号	238
3.12.3.7	来自进给轴/主轴的信号	239
3.12.4	系统变量	239

3.12.4.1	系统变量.....	239
3.12.5	OPI 变量.....	240
3.12.5.1	OPI 变量.....	240
4	K3: 补偿.....	241
4.1	引言.....	241
4.2	温度补偿.....	242
4.2.1	功能说明.....	242
4.2.2	调试.....	245
4.2.3	示例.....	246
4.2.3.1	为车床的 Z 轴调试温度补偿.....	246
4.3	间隙补偿.....	249
4.3.1	机械间隙补偿.....	249
4.3.1.1	功能说明.....	249
4.3.1.2	调试.....	250
4.3.2	动态间隙补偿.....	251
4.3.2.1	功能说明.....	251
4.3.2.2	调试.....	252
4.4	插补补偿.....	253
4.4.1	一般特性.....	253
4.4.2	对丝杠螺距误差和测量系统误差的补偿.....	255
4.4.2.1	功能说明.....	255
4.4.2.2	调试.....	257
4.4.2.3	示例.....	260
4.4.3	对垂度和角度误差的补偿.....	261
4.4.3.1	功能说明.....	261
4.4.3.2	调试.....	265
4.4.3.3	示例.....	268
4.4.4	双向丝杠螺距误差补偿.....	277
4.4.4.1	功能说明.....	277
4.4.4.2	调试.....	278
4.4.4.3	示例.....	281
4.4.5	垂度补偿和 NCU-Link 的综合应用 - 仅适用于 840D sl.....	285
4.4.6	插补补偿的特殊性.....	295
4.5	动态前馈控制（跟随误差补偿）.....	296
4.5.1	一般特性.....	296
4.5.2	转速前馈控制.....	298
4.5.3	转矩前馈控制.....	300
4.5.4	动态响应自适应功能.....	302

4.5.5	指令轴和 PLC 轴的前馈控制.....	303
4.5.6	边界条件.....	304
4.6	摩擦补偿（象限误差补偿）.....	305
4.6.1	一般功能说明.....	305
4.6.2	边界条件.....	306
4.6.3	补偿值恒定的摩擦补偿.....	306
4.6.3.1	功能激活.....	306
4.6.3.2	调试.....	307
4.6.4	补偿值和加速度相关的摩擦补偿.....	312
4.6.4.1	功能说明.....	312
4.6.4.2	功能激活.....	313
4.6.4.3	调试.....	313
4.6.5	适用于短程序段的补偿值.....	314
4.7	悬挂上的附加措施.....	315
4.7.1	电子配重.....	315
4.7.2	重启延时.....	316
4.8	数据表.....	318
4.8.1	机床数据.....	318
4.8.1.1	通用机床数据.....	318
4.8.1.2	通道专用机床数据.....	319
4.8.1.3	进给轴/主轴专用机床数据.....	319
4.8.2	设定数据.....	320
4.8.2.1	通用设定数据.....	320
4.8.2.2	进给轴/主轴专用设定数据.....	320
4.8.3	信号.....	321
4.8.3.1	从 NC 发出的信号.....	321
4.8.3.2	从 BAG 发出的信号.....	321
4.8.3.3	来自通道的信号.....	321
4.8.3.4	到进给轴/主轴的信号.....	321
4.8.3.5	来自进给轴/主轴的信号.....	321
5	K5: 运行方式组、通道、跨通道取轴.....	323
5.1	简要说明.....	323
5.2	运行方式组（BAG）- 仅用于 840D sl.....	324
5.3	通道 - 仅用于 840D sl.....	325
5.3.1	通道同步（程序协调）.....	326
5.3.2	通道同步：连续路径运行中的有条件停止.....	328
5.3.3	单通道式试车.....	333

5.4	跨通道取进给轴/主轴.....	338
5.4.1	引言.....	338
5.4.2	跨通道取轴示例.....	340
5.4.3	跨通道取轴方式.....	341
5.4.4	NC 程序跨通道取轴特性.....	343
5.4.5	将轴转换至中立轴（RELEASE）.....	344
5.4.6	在零件程序中接收进给轴或主轴（GET, GETD）.....	345
5.4.7	自动取轴.....	346
5.4.8	通过 PLC 取轴.....	347
5.4.9	可修改的跨通道取轴特性.....	350
5.4.10	通过轴容器旋转取轴.....	351
5.4.11	触发/不触发预处理停止的取轴.....	352
5.4.12	仅由 PLC 控制的轴.....	353
5.4.13	固定指定给 PLC 的轴.....	354
5.4.14	跨通道取轴和经过旋转的 WCS 中的几何轴的综合应用.....	356
5.4.15	通过同步动作取轴.....	357
5.4.16	跨通道取轴和龙门轴的综合应用.....	360
5.5	边界条件.....	360
5.6	数据表.....	362
5.6.1	机床数据.....	362
5.6.1.1	通用机床数据.....	362
5.6.1.2	通道专用机床数据.....	362
5.6.1.3	进给轴/主轴专用机床数据.....	365
5.6.2	设定数据.....	366
5.6.2.1	通道专用设定数据.....	366
5.6.3	信号.....	366
5.6.3.1	发送至/来自 BAG 的信号.....	366
5.6.3.2	发送至/来自通道的信号.....	366
6	M1: 坐标转换.....	367
6.1	简要说明.....	367
6.1.1	TRANSMIT（选件）.....	367
6.1.2	TRACYL（选件）.....	368
6.1.3	TRAANG（选件）.....	369
6.1.4	坐标转换级联.....	369
6.1.5	通过零件程序/软键激活坐标转换机床数据.....	370
6.2	端面转换 TRANSMIT（选件）.....	370
6.2.1	专用设置.....	371
6.2.2	激活.....	374

6.2.3	取消.....	375
6.2.4	应用.....	375
6.2.5	工作区域限制.....	381
6.2.6	TRANSMIT 下的叠加运动.....	382
6.2.7	回转轴旋转超 360 度时的检查.....	383
6.2.8	边界条件.....	383
6.2.9	示例：轴配置.....	384
6.3	柱面转换 TRACYL（选件）	387
6.3.1	前提条件.....	389
6.3.2	专用设置.....	393
6.3.3	激活.....	397
6.3.4	取消.....	398
6.3.5	边界条件.....	398
6.4	斜角转换 TRAANG（选件）	400
6.4.1	前提条件.....	401
6.4.2	专用设置.....	403
6.4.3	激活.....	404
6.4.4	取消.....	405
6.4.5	边界条件.....	405
6.4.6	编程 (G05, G07).....	407
6.5	坐标转换级联.....	408
6.5.1	激活坐标转换级联.....	410
6.5.2	取消坐标转换级联.....	411
6.5.3	坐标转换级联的特殊性.....	411
6.5.4	固定生效的坐标转换.....	411
6.5.5	坐标转换级联中的轴位置.....	417
6.6	坐标 PTP 运动.....	420
6.6.1	编写位置.....	423
6.6.2	轴角度的叠加范围.....	424
6.6.3	位置不唯一性示例.....	425
6.6.4	回转轴位置不唯一性示例.....	426
6.6.5	JOG 模式下的 PTP/CP 切换.....	426
6.7	坐标手动运动（选件）	427
6.8	通过零件程序/软键激活坐标转换机床数据.....	436
6.8.1	功能.....	436
6.8.2	边界条件.....	436
6.8.3	上电、运行方式切换、复位、程序段搜索、REPOS 时的控制系统特性.....	438
6.8.4	相关机床数据列表.....	439

6.9	边界条件.....	442
6.9.1	坐标转换级联.....	442
6.10	示例.....	443
6.10.1	TRANSMIT.....	443
6.10.2	TRACYL.....	445
6.10.3	TRAANG.....	450
6.10.4	坐标转换级联.....	451
6.10.5	通过零件程序激活坐标转换涉及的机床数据.....	456
6.10.6	坐标转换级联中的轴位置.....	457
6.11	数据表.....	460
6.11.1	机床数据.....	460
6.11.1.1	TRANSMIT.....	460
6.11.1.2	TRACYL.....	462
6.11.1.3	TRAANG.....	464
6.11.1.4	坐标转换级联.....	465
6.11.1.5	非坐标转换专用机床数据.....	465
6.11.2	信号.....	466
6.11.2.1	来自通道的信号.....	466
7	M5: 测量.....	467
7.1	简要说明.....	467
7.2	硬件条件.....	468
7.2.1	可使用的测头.....	468
7.3	单通道测量.....	469
7.3.1	测量.....	469
7.3.2	测量结果.....	470
7.4	单轴测量.....	471
7.4.1	测量.....	471
7.4.2	报文选择.....	475
7.4.3	测量结果.....	475
7.5	设置零点、工件测量和刀具测量.....	477
7.5.1	实际值设定和对刀.....	477
7.5.2	工件测量.....	478
7.5.2.1	输入值.....	478
7.5.2.2	选择测量.....	486
7.5.2.3	输出值.....	488
7.5.2.4	计算方法.....	488
7.5.2.5	计算时采用的单位和测量变量.....	492

7.5.2.6	诊断.....	494
7.5.3	工件测量类型.....	494
7.5.3.1	测量棱边（测量类型 1、2、3）	494
7.5.3.2	测量角（测量类型 4、5、6、7）	499
7.5.3.3	测量钻孔（测量类型 8）	503
7.5.3.4	测量轴（测量类型 9）	506
7.5.3.5	测量槽（测量类型 12）	507
7.5.3.6	测量隔断（测量类型 13）	510
7.5.3.7	测量几何轴和辅助轴（测量类型 14、15）	511
7.5.3.8	测量斜边（测量类型 16）	514
7.5.3.9	测量倾斜平面中的角度（测量类型 17）	516
7.5.3.10	通过测量重新定义 WCS 参考系统（测量类型 18）	520
7.5.3.11	1 维、2 维、3 维设定值给定测量（测量类型 19、20、21）	523
7.5.3.12	任意坐标系中的测量点测量（测量类型 24）	528
7.5.3.13	测量矩形（测量类型 25）	532
7.5.3.14	用于备份数据管理框架的测量（测量类型 26）	533
7.5.3.15	用于恢复备份的数据管理框架的测量（测量类型 27）	534
7.5.3.16	用于为锥面车削设定叠加旋转的测量（测量类型 28）	535
7.5.4	刀具测量.....	536
7.5.5	刀具测量的测量类型.....	537
7.5.5.1	测量刀具长度（测量类型 10）	537
7.5.5.2	测量刀具直径（测量类型 11）	540
7.5.5.3	通过放大镜功能测量刀具长度（测量类型 22）	541
7.5.5.4	通过标记位置或当前位置测量刀具长度（测量类型 23）	542
7.5.5.5	通过定向测量两把刀具的长度.....	544
7.6	测量精度和测试.....	556
7.6.1	测量精度.....	556
7.6.2	测头功能测试.....	557
7.7	仿真测量.....	557
7.7.1	一般功能.....	557
7.7.2	位置相关切换请求.....	558
7.7.3	外部切换请求.....	560
7.7.4	系统变量.....	561
7.8	示例 - 仅用于 840D sl.....	561
7.8.1	测量模式 1.....	561
7.8.2	测量模式 2.....	562
7.8.3	连续测量.....	563
7.8.4	功能测试和重复精度.....	564

7.9	数据表.....	566
7.9.1	机床数据.....	566
7.9.1.1	通用机床数据.....	566
7.9.1.2	通道专用机床数据.....	566
7.9.2	系统变量.....	566
8	N3: 软件挡块, 行程开关信号 - 仅适用于 840D sl.....	569
8.1	简要说明.....	569
8.2	挡块信号和挡块位置.....	570
8.2.1	挡块信号单独输出.....	570
8.2.2	挡块信号关联输出.....	575
8.2.3	挡块位置.....	579
8.2.4	提前时间/延迟时间 (动态挡块)	581
8.3	挡块信号输出.....	582
8.3.1	激活.....	582
8.3.2	挡块信号输出到 PLC.....	583
8.3.3	挡块信号以位置控制周期输出到 NCK I/O.....	584
8.3.4	挡块信号定时输出.....	585
8.3.5	挡块信号单独定时输出.....	587
8.4	“行程-时间”挡块.....	588
8.5	边界条件.....	589
8.6	数据表.....	590
8.6.1	机床数据.....	590
8.6.1.1	通用机床数据.....	590
8.6.2	设定数据.....	590
8.6.2.1	通用设定数据.....	590
8.6.3	信号.....	591
8.6.3.1	到进给轴/主轴的信号.....	591
8.6.3.2	来自进给轴/主轴的信号.....	591
9	N4: 冲裁和步冲 - 仅适用于 840D sl.....	593
9.1	简要描述.....	593
9.2	冲程控制.....	593
9.2.1	概述.....	593
9.2.2	快速信号.....	594
9.2.3	冲程释放条件.....	595
9.2.4	冲裁后的轴启动.....	598
9.2.5	冲裁和步冲专用的 PLC 信号.....	599

9.2.6	仲裁和步冲对标准 PLC 信号的特定反应.....	600
9.2.7	信号监控.....	600
9.3	激活和撤销.....	601
9.3.1	语言指令.....	601
9.3.2	功能扩展.....	605
9.3.3	与老系统的兼容性.....	610
9.4	自动划分行程.....	612
9.4.1	简介.....	612
9.4.2	针对轨迹轴的特性.....	613
9.4.3	针对单根轴的特性.....	618
9.5	可旋转刀具.....	623
9.5.1	概述.....	623
9.5.2	主刀具和子刀具的“耦合”.....	624
9.5.3	切向控制.....	625
9.6	保护区.....	629
9.7	边界条件.....	630
9.8	示例.....	630
9.8.1	定义步冲开始的示例.....	630
9.9	数据表.....	635
9.9.1	机床数据.....	635
9.9.1.1	通用机床数据.....	635
9.9.1.2	通道专用机床数据.....	635
9.9.2	设定数据.....	636
9.9.2.1	通道专用设定数据.....	636
9.9.3	信号.....	636
9.9.3.1	发送至通道的信号.....	636
9.9.3.2	来自通道的信号.....	637
9.9.4	语言指令.....	637
10	P2: 定位轴.....	639
10.1	简要说明.....	639
10.2	独立通道、定位轴或并行定位轴.....	641
10.2.1	独立通道 - 仅用于 840D sl.....	641
10.2.2	定位轴.....	642
10.2.3	并行定位轴.....	645
10.3	运动特性和插补过程.....	646
10.3.1	轨迹插补器和轴插补器.....	646

10.3.2	G0 下轨迹轴的插补特性.....	646
10.3.3	自控单轴运动.....	649
10.3.4	包含 NC 控制 ESR 的自控单轴运动.....	653
10.4	定位轴动态响应.....	655
10.5	编程.....	657
10.5.1	简介.....	657
10.5.2	外部编程时的旋转进给率.....	660
10.6	程序段切换.....	661
10.6.1	可设置的程序段切换时间点.....	663
10.6.2	程序段搜索时的运动结束条件.....	668
10.7	通过 PLC 控制.....	669
10.7.1	由 PLC 启动并行定位轴.....	671
10.7.2	由 PLC 控制的轴.....	672
10.7.3	PLC 轴的控制特性.....	674
10.8	特殊功能下的特性.....	675
10.8.1	空运行进给率 (DRY RUN)	675
10.8.2	单程序段.....	675
10.9	示例.....	676
10.9.1	运动特性和插补过程.....	676
10.9.1.1	G0 下轨迹轴无插补运行.....	677
10.10	数据表.....	677
10.10.1	机床数据.....	677
10.10.1.1	NC 专用机床数据.....	677
10.10.1.2	通道专用机床数据.....	677
10.10.1.3	进给轴/主轴专用机床数据.....	678
10.10.2	设定数据.....	678
10.10.2.1	进给轴/主轴专用设定数据.....	678
10.10.3	信号.....	678
10.10.3.1	发送至通道的信号.....	678
10.10.3.2	来自通道的信号.....	679
10.10.3.3	到进给轴/主轴的信号.....	679
10.10.3.4	来自进给轴/主轴的信号.....	679
11	P5: 往复运动 - 仅限 840D sl.....	681
11.1	简要说明.....	681
11.2	异步往复运动.....	682
11.2.1	异步往复运动的设置选项.....	683
11.2.2	由 PLC 控制的异步往复运动.....	689

11.2.3	异步往复运动中的特殊响应.....	690
11.3	通过同步动作控制的往复运动.....	693
11.3.1	在换向点 1 或 2 上作进给运动.....	696
11.3.2	在换向区内作进给运动.....	697
11.3.3	在两个换向点上作进给运动.....	699
11.3.4	在换向点上停止往复运动.....	699
11.3.5	继续往复运动.....	701
11.3.6	阻止下一个进给行程过早开始.....	701
11.3.7	OSCILL 指令：为往复轴指定进给轴.....	702
11.3.8	POSP 指令：指定进给运动方式.....	703
11.3.9	从外部触发往复轴换向.....	703
11.4	边界条件.....	705
11.5	示例.....	705
11.5.1	异步往复运动程序示例.....	705
11.5.2	示例 1：带同步动作的往复运动.....	706
11.5.3	示例 2：带同步动作的往复运动.....	709
11.5.4	示例：起点.....	711
11.5.4.1	通过语言指令来定义起点.....	711
11.5.4.2	通过设定数据启动往复运动.....	712
11.5.4.3	起点和换向点重合时的往复运动（起点 = 换向点 1）.....	713
11.5.5	示例：从外部触发往复轴换向.....	715
11.5.5.1	由外部触发往复轴换向时通过同步动作修改换向点.....	715
11.6	数据表.....	716
11.6.1	机床数据.....	716
11.6.1.1	通用机床数据.....	716
11.6.2	设定数据.....	716
11.6.2.1	进给轴/主轴专用设定数据.....	716
11.6.3	信号.....	717
11.6.3.1	到进给轴/主轴的信号.....	717
11.6.3.2	来自进给轴/主轴的信号.....	717
11.6.4	系统变量.....	717
11.6.4.1	用于同步动作的主处理变量.....	717
12	R2:回转轴.....	721
12.1	简要说明.....	721
12.2	360 °模数.....	726
12.3	回转轴的编程.....	728
12.3.1	简介.....	728

12.3.2	激活模数功能的回转轴（不受限制的回转轴）	729
12.3.3	没有激活模数功能的回转轴	734
12.3.4	回转轴的其他编程特殊性	736
12.4	调试回转轴	737
12.5	回转轴的特殊性	738
12.6	示例	739
12.7	数据表	740
12.7.1	机床数据	740
12.7.1.1	通用机床数据	740
12.7.1.2	进给轴/主轴专用机床数据	740
12.7.2	设定数据	740
12.7.2.1	通用设定数据	740
12.7.2.2	进给轴/主轴专用设定数据	741
12.7.3	信号	741
12.7.3.1	到进给轴/主轴的信号	741
12.7.3.2	来自进给轴/主轴的信号	741
13	S3:主轴同步	743
13.1	简要说明	743
13.1.1	功能	743
13.1.2	主轴同步	745
13.1.3	主轴同步的前提条件	751
13.1.4	通过零件程序来激活主轴同步	752
13.1.5	通过零件程序来撤销主轴同步	754
13.1.6	通过 PLC 控制主轴同步	755
13.1.7	监控主轴同步	758
13.2	编程	760
13.2.1	定义 (COUPDEF)	761
13.2.2	激活和撤销同步 (COUPON、COUPONC、COUPOF)	764
13.2.3	主轴同步的轴系统变量	765
13.2.4	自动激活和撤销位置闭环控制	767
13.3	定义	768
13.3.1	NC 启动时同步主轴对的特性	769
13.3.2	复位时同步主轴对的特性	769
13.4	特点	770
13.4.1	主轴同步的一般特点	770
13.4.2	再次恢复副主轴和主主轴的同步	772
13.4.3	主轴同步和 NC/PLC 接口信号	774

13.4.4	主主轴和副主轴之间的差速.....	778
13.4.5	同步补偿中同步信号的特性.....	783
13.4.6	删除同步补偿及 NC 复位.....	783
13.4.7	主轴同步功能调试的特殊性.....	784
13.5	边界条件.....	789
13.6	示例.....	789
13.7	数据表.....	790
13.7.1	机床数据.....	790
13.7.1.1	NC 专用机床数据.....	790
13.7.1.2	通道专用机床数据.....	790
13.7.1.3	进给轴/主轴专用机床数据.....	791
13.7.2	设定数据.....	792
13.7.2.1	通道专用设定数据.....	792
13.7.3	信号.....	792
13.7.3.1	发送至通道的信号.....	792
13.7.3.2	来自通道的信号.....	792
13.7.3.3	到进给轴/主轴的信号.....	792
13.7.3.4	来自进给轴/主轴的信号.....	793
13.7.4	系统变量.....	793
14	S7:存储器配置.....	795
14.1	简要说明.....	795
14.2	存储器结构.....	795
14.2.1	主动和被动文件系统.....	795
14.2.2	重新配置.....	797
14.3	配置静态用户存储器.....	798
14.3.1	静态 NC 存储器的结构.....	798
14.3.2	调试.....	800
14.4	配置动态用户存储器.....	802
14.4.1	动态 NC 存储器的结构.....	802
14.4.2	调试.....	803
14.5	数据表.....	804
14.5.1	机床数据.....	804
14.5.1.1	通用机床数据.....	804
14.5.1.2	通道专用机床数据.....	808
14.5.1.3	进给轴/主轴专用机床数据.....	810

15	T1: 分度轴	811
15.1	简要说明	811
15.2	运行分度轴	811
15.2.1	在 JOG 模式下运行分度轴	812
15.2.2	在 AUTO 模式下运行分度轴	814
15.2.3	PLC 运行分度轴	815
15.3	设置分度轴	815
15.4	编程分度轴	818
15.5	等距分度	822
15.5.1	属性	822
15.5.2	端面齿	824
15.5.3	特殊情况下端面齿轴的特性	825
15.5.4	限制	826
15.5.5	机床数据修改的生效方式	827
15.6	调试分度轴	827
15.7	分度轴的特点	831
15.8	示例	832
15.8.1	等距分度示例	832
15.9	数据表	834
15.9.1	机床数据	834
15.9.1.1	通用机床数据	834
15.9.1.2	进给轴/主轴专用机床数据	834
15.9.2	设定数据	835
15.9.2.1	通用设定数据	835
15.9.3	信号	835
15.9.3.1	来自进给轴/主轴的信号	835
15.9.4	系统变量	835
16	W3:换刀	837
16.1	简要说明	837
16.2	刀库和换刀装置	838
16.3	换刀时间	838
16.4	换刀导致的加工中断时间	838
16.5	换刀过程的控制	839
16.6	换刀点	840
16.7	边界条件	840

16.8	示例.....	840
16.9	数据表.....	843
16.9.1	机床数据.....	843
16.9.1.1	通用机床数据.....	843
16.9.1.2	通道专用机床数据.....	843
16.9.1.3	进给轴/主轴专用机床数据.....	843
16.9.2	信号.....	844
16.9.2.1	来自通道的信号.....	844
17	W4: 磨削专用的刀具补偿和监控.....	845
17.1	磨削专用的刀具补偿.....	845
17.1.1	刀具数据的结构.....	845
17.1.2	刀沿补偿数据.....	847
17.1.3	刀具的磨削数据.....	851
17.1.4	磨削刀具示例.....	857
17.2	在线刀具补偿.....	861
17.2.1	简介.....	861
17.2.2	连续写入在线刀具补偿.....	863
17.2.3	激活/撤销在线刀具补偿.....	866
17.2.4	连续写入在线刀具补偿示例.....	867
17.2.5	不连续写入在线刀具补偿.....	868
17.2.6	关于在线补偿的提示.....	869
17.3	在线刀具半径补偿.....	870
17.4	磨削专用的刀具监控.....	871
17.4.1	简介.....	871
17.4.2	几何尺寸监控.....	872
17.4.3	转速监控.....	873
17.4.4	激活/撤销刀具监控.....	874
17.5	恒定砂轮圆周速度(GWPS).....	875
17.5.1	简介.....	875
17.5.2	激活/撤销以及编程 GWPS，系统变量.....	876
17.5.3	GWPS 在所有运行方式中生效.....	877
17.5.4	GWPS 编程举例.....	878
17.6	边界条件.....	880
17.6.1	换刀和在线刀具补偿组合使用.....	880
17.7	数据表.....	880
17.7.1	机床数据.....	880
17.7.1.1	通用机床数据.....	880

17.7.1.2	通道专用机床数据.....	881
17.7.1.3	进给轴/主轴专用机床数据.....	881
17.7.2	信号.....	881
17.7.2.1	来自进给轴/主轴的信号.....	881
18	Z2: NC/PLC 接口信号.....	883
18.1	NCK 数字量/模拟量输入/输出(A4).....	883
18.1.1	发送至 NC 信号(DB10).....	883
18.1.2	从 NC 发出的信号(DB10).....	893
18.2	分布式系统(B3).....	897
18.2.1	已定义的逻辑功能/ Defines.....	897
18.2.2	多操作面板配多 NC: DB19 中的接口.....	900
18.2.3	从 NC 发出的信号(DB10).....	908
18.2.4	从进给轴/主轴发出的信号(DB31, ...).....	909
18.3	手动运行和手轮运行(H1).....	910
18.3.1	从 NC 发出的信号(DB10).....	910
18.3.2	发送至通道的信号 (DB21, ...)	914
18.3.3	从通道发出的信号 (DB21, ...)	921
18.3.4	轮廓手轮相关信号.....	929
18.3.5	发送到进给轴/主轴的信号(DB31, ...).....	932
18.3.6	从进给轴/主轴发出的信号(DB31, ...).....	938
18.4	补偿(K3).....	942
18.5	运行方式组、通道和跨通道取轴(K5).....	942
18.5.1	发送到进给轴/主轴的信号(DB31, ...).....	942
18.5.2	从进给轴/主轴发出的信号(DB31, ...).....	943
18.6	坐标转换 (M1).....	944
18.6.1	从通道发出的信号 (DB21, ...)	944
18.7	测量(M5).....	945
18.7.1	从 NC 发出的信号(DB10).....	945
18.7.2	从进给轴/主轴发出的信号(DB31, ...).....	945
18.8	软件挡块和行程开关信号(N3).....	946
18.8.1	信号一览.....	946
18.8.2	从 NC 发出的信号(DB10).....	947
18.8.3	发送到进给轴/主轴的信号(DB31, ...).....	948
18.8.4	从进给轴/主轴发出的信号(DB31, ...).....	949
18.9	仲裁和步冲(N4).....	949
18.9.1	信号一览.....	949
18.9.2	发送至通道的信号 (DB21, ...)	950

18.9.3	从通道发出的信号 (DB21, ...)	952
18.10	定位轴(P2)	952
18.10.1	发送到进给轴/主轴的信号(DB31, ...)	953
18.10.2	Function Call - 仅限 840D sl	956
18.11	往复运动(P5)	957
18.11.1	发送给进给轴/主轴信号(DB31, ...)	957
18.11.2	从进给轴/主轴发出的信号(DB31, ...)	959
18.12	回转轴(R2)	961
18.12.1	发送到进给轴/主轴的信号(DB31, ...)	961
18.12.2	从进给轴/主轴发出的信号(DB31, ...)	961
18.13	同步主轴(S3)	962
18.13.1	发送到进给轴/主轴的信号(DB31, ...)	962
18.13.2	从进给轴/主轴发出的信号(DB31, ...)	962
18.14	存储器配置(S7)	965
18.15	分度轴(T1)	966
18.15.1	从进给轴/主轴发出的信号(DB31, ...)	966
18.16	换刀(W3)	967
18.17	磨削专用的刀具补偿和监控(W4)	967
18.17.1	从进给轴/主轴发出的信号(DB31, ...)	967
A	附录	969
A.1	缩略语列表	969
A.2	手册一览	981
	词汇表	983
	索引	1005

A4: SINUMERIK 840D sl 的 NCK 数字量/模拟量输入/输出

1.1 引言

功能

通过 NCK 的数字量/模拟量输入/输出可以按插补周期读取或输出信号。这些信号例如可用于执行下列功能：

- 在一个程序段内编写多个进给率值
- 在程序段内编写多个辅助功能
- 达到成品尺寸后快速退刀
- 针对特定轴的剩余行程删除
- 程序跳转
- 快速 NC 启动
- 模拟量测量钳
- 行程开关信号
- 冲裁/步冲功能
- 模拟量值控制

硬件

在 SINUMERIK 840D sl 上，NCU 上有三个板载 14 针硬件 I/O：X122、X132 和 X142。其中只有 X142 可用作 NCK 高速 I/O。

X142 上有四路高速数字量输入和四路输出，可通过首个地址字节以及系统变量 \$A_IN[1...4] 和 \$A_OUT[1...4] 访问。

在 PROFIBUS DP/MPI 接口 X126 和 X136 上还可连接附加的 ET 200 模块。NCK 数字量输入/输出因此可各增加 32 路，模拟量输入/输出各增加 8 路。下文中将把此类 NCK 输入/输出称为**外部 NCK 输入/输出**。

1.1 引言

表格 1-1 NCK 数字量和模拟量输入/输出的最大数量

	总计	NCU 板载输入/输出	外部 NCK 输入/输出
数字量输入	36	4	32
数字量输出	36	4	32
模拟量输入	8	-	8
模拟量输出	8	-	8

文档:

硬件说明的更多相关信息请见:

- SINUMERIK 840D sl NCU 设备手册
- SIMATIC ET 200S FC 操作说明

访问 NCK 输入/输出

可通过以下方式访问 NCK 高速数字量/模拟量输入/输出:

- 通过 PLC 访问 (页 33) (适用于板载输入/输出和 PROFIBUS 输入/输出)
- 通过 PROFIBUS 访问 (页 62) (适用于 PROFIBUS 输入/输出)

监控

下列监控对外部 NCK 输入/输出生效:

- 启动时:
 - 检查 ET 200 模块的 I/O 模块配备是否与机床数据设置一致。
- 循环运行中:
 - 按插补周期进行生命符号监控
 - 按插补周期进行模块监控
 - 温度监控

出现故障时, 系统会撤销“NC 就绪”信号并输出一条报警。

出现故障时的特性

出现故障 (例如“NC 就绪”置 0)、NCU 出错以及掉电时, NCK 数字量/模拟量输出会切换至安全状态 (输出端上为 0 V)。

1.2 通过 PLC 访问

1.2.1 简要说明

配置 NCK 输入/输出

调试时，可访问的 NCK 数字量和模拟量输入/输出的数量、对应的硬件口以及执行的 NC 功能是通过机床数据定义的。

更多信息参见“NCK 输入/输出的配置 (页 34)”。

数据交换

数字量和模拟量输入/输出的信号及模拟值在 NC 中（零件程序、ASUP、同步动作等）以系统变量的形式提供：

- \$A_IN [<n>] 读取 NCK 数字量输入 <n>（其中 <n> = 1...4 和 9...40）
更多信息参见“NCK 的数字量输入 (页 37)”。
- \$A_OUT [<n>] 写入/读取 NCK 数字量输出 <n>（其中 <n> = 1...4 和 9...40）
更多信息参见“NCK 的数字量输出 (页 39)”。
- \$A_INA [<n>] 读取 NCK 模拟量输入 <n>（其中 <n> = 1...8）
更多信息参见“NCK 的模拟量输入 (页 44)”。
- \$A_OUTA [<n>] 写入/读取 NCK 模拟量输出 <n>（其中 <n> = 1...8）
更多信息参见“NCK 的模拟量输出 (页 47)”。

<n> = NCK 输入/输出的编号

说明

在通过零件程序读取这些系统变量时，控制系统内部会触发一次预处理停止（STOPRE）。

比较器输入

除了 NCK 数字量/模拟量输入外，系统还提供 16 个内部比较器输入（比较器输入字节 1 和 2）。

比较器输入的信号状态是通过对比模拟量输入信号与设定数据中的阈值得出的。

1.2 通过 PLC 访问

更多信息参见“比较器输入 (页 53)”。

PLC 输入/输出可直接通过 NCK 访问

通过零件程序不仅可以直接访问数字量输入信号和模拟量输入值（最长达 32 个字节），也可以直接访问数字量输出信号和模拟量输出值（最长达 32 个字节）。在 PLC 配置中须考虑到这些字节。这些字节是直接由 PLC 操作系统处理的。这样一来，NC 和 PLC I/O 模块之间的信号传输时间便可缩短到 0.5 毫秒这一量级。

说明

由 NCK 直接访问的输出字节不允许由 PLC 用户程序写入，否则会使 NCK 和 PLC 之间的访问出现冲突。

更多信息参见“可由 NC 直接访问的 PLC 输入/输出 (页 57)”。

1.2.2 NCK 输入/输出的配置

有效 NCK 输入/输出的数量

可访问的 NCK 数字量输入/输出字节的数量以及模拟量输入/输出的数量是通过以下机床数据定义的：

• MD10350 \$MN_FASTIO_DIG_NUM_INPUTS	有效 NCK 数字量输入字节的数量（最大为 5）
• MD10360 \$MN_FASTIO_DIG_NUM_OUTPUTS	有效 NCK 数字量输出字节的数量（最大为 5）
• MD10300 \$MN_FASTIO_ANA_NUM_INPUTS	有效 NCK 模拟量输入的数量（最大为 8）
• MD10310 \$MN_FASTIO_ANA_NUM_OUTPUTS	有效 NCK 模拟量输出的数量（最大为 8）

若零件程序访问了上述机床数据中未定义的输入/输出，系统会输出相应报警。

NCK 输入/输出不需要是实际存在的硬件口。此时 NCK 内部会将信号状态或二进制模拟量值置为零。但这些值无法通过 PLC 修改。

指定外部 NCK 输入/输出的硬件

通过以下机床数据为外部 NCK 输入/输出指定 I/O 信号模块及 I/O 模块：

- MD10366 \$MN_HW_ASSIGN_DIG_FASTIN[<n>] 外部数字量输入对应的硬件口
 - MD10368 \$MN_HW_ASSIGN_DIG_FASTOUT[<n>] 外部数字量输出对应的硬件口
 - MD10362 \$MN_HW_ASSIGN_ANA_FASTIN[<n>] 外部模拟量输入对应的硬件口
 - MD10364 \$MN_HW_ASSIGN_ANA_FASTOUT[<n>] 外部模拟量输出对应的硬件口
- <n> = 下标，用于标明各个外部数字量 I/O 字节（0 ... 3）以及各个外部模拟量输入/输出（0 ... 7）

示例：

为了与外部 PROFIBUS 模块的数字量输入/输出进行数据交换，需要在 NC 中配置两个额外的输入字节和一个额外的输出字节。

硬件指定：

MD10366 \$MN_HW_ASSIGN_DIG_FASTIN[0]='H5000200'；用于 \$A_IN[9] ... [16]

MD10366 \$MN_HW_ASSIGN_DIG_FASTIN[1]='H5000201'；用于 \$A_IN[17] ... [20]

MD10368 \$MN_HW_ASSIGN_DIG_FASTOUT[0]='H5000200'；用于 \$A_OUT[9] ... [16]

定义“H5000000”时，输入/输出位于 PROFIBUS 上。

其中的低位以十六进制指定了模块的起始地址（逻辑 PROFIBUS 地址）。

在 SIMATIC Manager 中进行配置时，该逻辑地址以十进制表示（例如值“H202”对应逻辑地址“514”）。

有效 NCK 输入/输出字节的数量：

MD10350 \$MN_FASTIO_DIG_NUM_INPUTS = 3

MD10360 \$MN_FASTIO_DIG_NUM_OUTPUTS = 2

说明

在配置 MD10350 及 MD10360 时必须将板载字节一同计算在内！

模拟量 NCK 输入/输出的权重系数

通过权重系数可对各 NCK 模拟量输入/输出进行调整，使其与所使用的模拟量 I/O 模块的模数或数模转换器相匹配。

- MD10320 \$MN_FASTIO_ANA_INPUT_WEIGHT[<n>] NCK 模拟量输入的权重系数
更多信息参见“NCK 的模拟量输入 (页 44)”。
- MD10330 \$MN_FASTIO_ANA_OUTPUT_WEIGHT[<n>] NCK 模拟量输出的权重系数
更多信息参见“NCK 的模拟量输出 (页 47)”。

指定 NC 功能

NCK 输入/输出是多个 NC 功能生效的前提条件。

此类 NC 功能是通过机床数据来指定所需的 NCK 输入/输出的。功能不同，机床数据也有所不同。比如，功能“一个程序段中编写多个进给率”便是通过以下机床数据指定 NCK 输入/输出的：

MD21220 \$MC_MULTFEED_ASSIGN_FASTIN

如果选择的是数字量输入/输出，必须在机床数据中指定一个字节地址。此处总是以字节方式进行指定。

字节地址	NCK 数字量输入/输出			
0	无			
1	1 至 4（板载 I/O）	与	5 至 8	（NCK 输出，非硬件口）
2	9	到	16	（外部 NCK 输入/输出）
3	17	到	24	（外部 NCK 输入/输出）
4	25	到	32	（外部 NCK 输入/输出）
5	33	到	40	（外部 NCK 输入/输出）
128	比较器字节 1 的输入 1 至 8			
129	比较器字节 2 的输入 9 至 16			

说明**重复指定**

系统不会检查输入是否重复指定。

但系统会在启动时检查输出是否重复指定，如发现这一错误，会显示报警。

1.2.3 NCK 的数字量输入/输出

1.2.3.1 NCK 的数字量输入

功能

通过 NCK 数字量输入外部信号可以控制工件加工程序的执行过程。

数字量输入 <n> 的信号状态可直接在零件程序中通过系统变量 \$A_IN [<n>] 查询。

硬输入上的信号状态可通过 PLC 用户程序修改。

应用

NCK 数字量输入例如可用于下列功能：

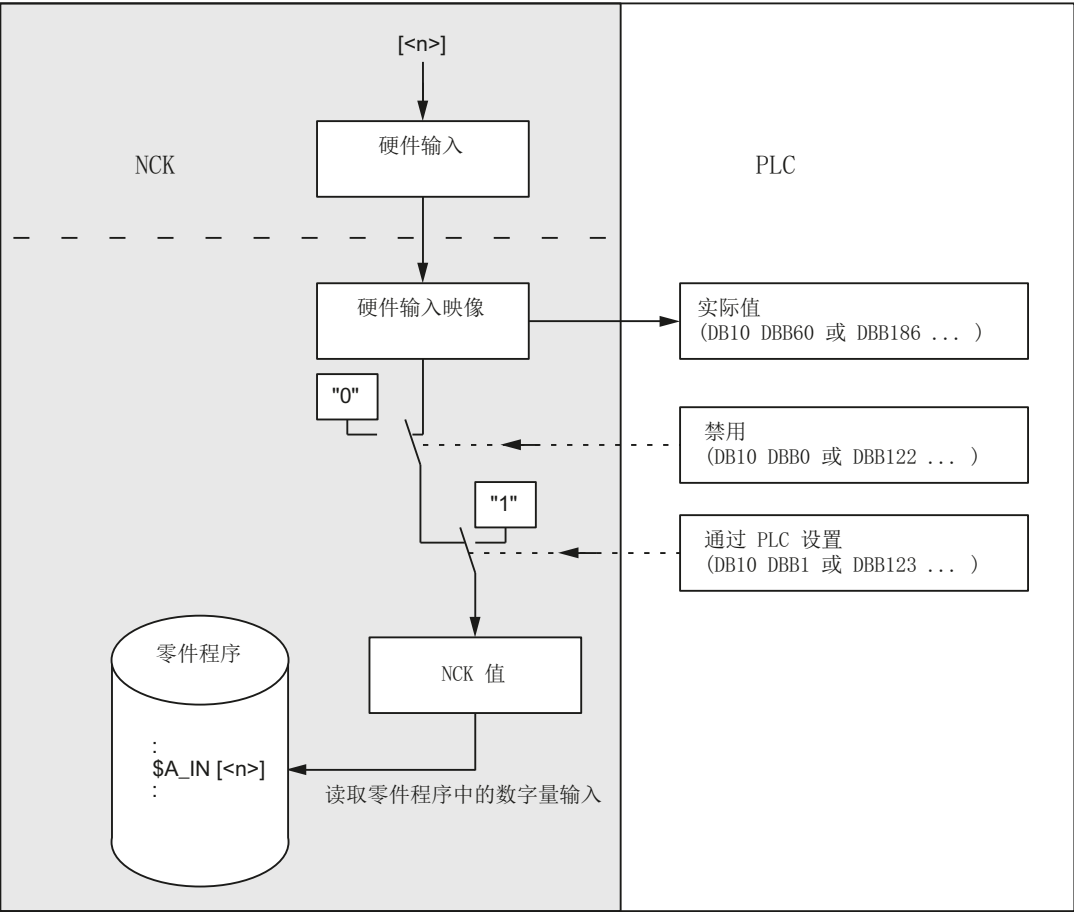
- 删除定位轴的剩余行程
- 在程序段末尾进行快速程序跳转
- 编写读取禁止
- 在一个程序段内编写多个进给率值

文档：

功能手册之同步动作分册

信号流

下图显示了 NCK 数字量输入的信号流：



读取实际值

NCK 将其数字量输入的信号状态发送至 PLC：

DB10 DBB60 或 DBB186 ... （NCK 数字量输入的实际值）

实际值反映了硬输入上的真实信号状态。“实际值”不考虑 PLC 的影响。

禁用输入

可通过 PLC 用户程序禁用各个 NCK 数字量输入：

DB10 DBB0 或 DBB122 ... （禁用 NCK 数字量输入）

此时控制系统内部会将对应输入置为 0。

通过 PLC 置位输入

此外 PLC 还可将每个数字量输入置为 1：

DB10 DBB1 或 DBB123 ... (通过 PLC 置位 NCK 数字量输入)

此接口信号置为 1 时，硬输入上的信号状态及输入禁用立即失效。

重新上电/复位后的特性

重新上电和复位后各个输入上原有的信号电平继续保持。必要时可如上文所述通过 PLC 用户程序将各输入禁用或置为 1。

1.2.3.2 NCK 的数字量输出

功能

通过 NCK 数字量输出可以针对具体加工非常快地触发一些要求快速处理的开关操作，而无需经过 PLC 周期时间。这一过程是由程序控制的，比如：通过程序段切换。

数字量输出 <n> 的信号状态可直接在零件程序中通过系统变量 \$A_OUT[<n>] 设置或重新读取。

此外还可采用多种方式通过 PLC 修改该设置的信号状态。

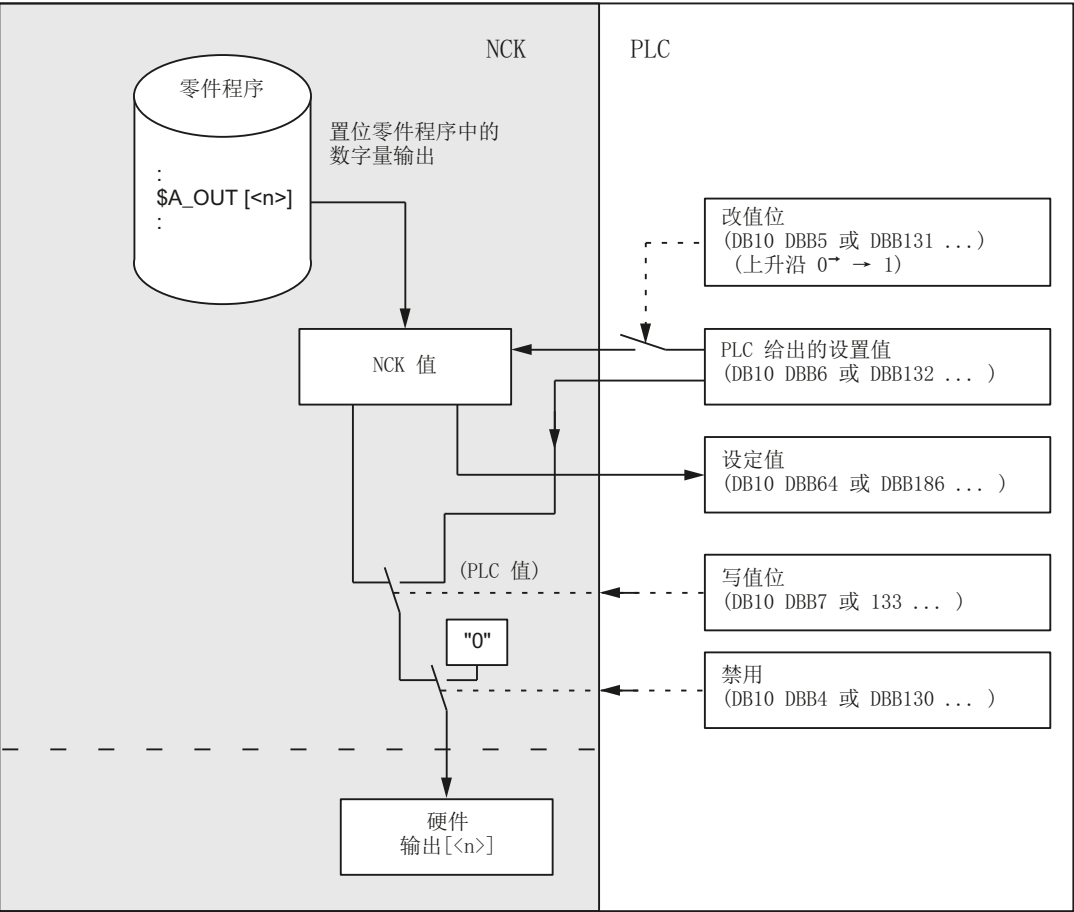
应用

NCK 数字量输出例如可用于下列功能：

- 行程开关信号（参见章节“N3：软件挡块，行程开关信号 - 仅适用于 840D sl (页 569)”）
- 比较器信号的输出

信号流

下图显示了 NCK 数字量输出的信号流：



改值位

每个可由零件程序设置的输出均可由 PLC 借助改值位改写。此时之前的“NCK 值”将丢失。

若需通过 PLC 改写 NCK 值，执行以下步骤：

1. 在 PLC 接口上将对应输出预设为所需的信号状态：
DB10 DBB6 或 DBB132 ... （PLC 给出的 NCK 数字量输出的设置值）
2. 置位该输出对应的改值位（上升沿 0 → 1），使“设置值”变为新的“NCK 值”：
DB10 DBB5 或 DBB131 ...
该值一直生效，直到下一次 PLC 或零件程序给出新的值。

写值位

PLC 既可以使每个输出保持当前比如来自 NC 零件程序的“NCK 值”，也可以通过写值位将“PLC 值”传送给硬件输出。

若需设定“PLC 值”，执行以下步骤：

1. 在 PLC 接口上将对应输出预设为所需的信号状态：
DB10 DBB6 或 DBB132 ...（PLC 给出的 NCK 数字量输出的设置值）
2. 将该输出对应的写值位置为 1：
DB10 DBB7 或 DBB133 ...（NCK 数字量输出的写值位）

和改值位不同的是，写值位不会使原先的 NCK 值丢失。一旦 PLC 将该写值位置为 0 后，原先的 NCK 值恢复生效。

说明

无论是改值位还是写值位，PLC 接口上使用的始终是相同的设置值。因此同时通过改值位和写值位修改信号状态只会产生相同的输出信号状态！

禁用输出

可通过 PLC 用户程序禁用各个 NCK 数字量输出：

DB10 DBB4 或 DBB130 ...（禁用 NCK 数字量输出）

此时在硬输出上输出“0 信号”。

读取设定值

可通过 PLC 用户程序读取数字量输出的当前“NCK 值”：

DB10 DBB64 或 DBB186 ...（NCK 数字量输出的设定值）

此时须注意，此设定值不考虑 PLC 的写值位以及输出的禁用状态。这样一来，设定值可能会与硬输出上的实际信号状态有别。

程序结束/复位后的特性

在程序结束或复位后，每个数字量输出都可在 PLC 用户程序中利用改值位、写值位或禁用信号根据需求设置。

重新上电后的特性

重新上电后数字量输出都会置为 0。这一信号状态可针对具体应用在 PLC 用户程序中利用改值位和写值位改写。

非硬件 NCK 数字量输出

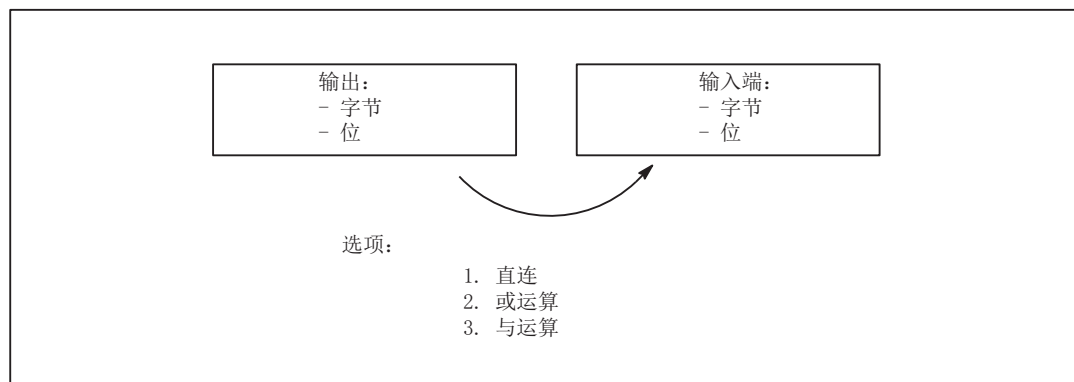
若在零件程序中写入了 MD10360 定义的、但不作为真实硬件存在的 NCK 数字量输出，系统不会发出报警。NCK 值可通过 PLC 读取（DB10 DBB64 或 DBB186 ...）。

1.2.3.3 NCK 高速数字量输入/输出之间的互联（直连和逻辑运算）

功能

NCK 输入/输出中的高速输入可根据高速输出的信号状态通过软件加以设置：

一览：



直连

NCK 输入/输出中的高速输入直接设为高速输出的信号状态。

逻辑或运算

NCK 输入/输出中高速输入的信号状态是输出信号和对应输出信号经过逻辑或运算后的结果。

逻辑和运算

NCK 输入/输出中高速输入的信号状态是输出信号和对应输出信号经过逻辑和运算后的结果。

特殊情况

- 若向同一个输入位指定了多个输出位，则机床数据下标最高的设定生效。
- 若指定了不存在或未激活的输入/输出，系统会忽略此指定且不发出报警。可查看下面两个机床数据的值来检查有效 NCK 输入/输出字节的数量。

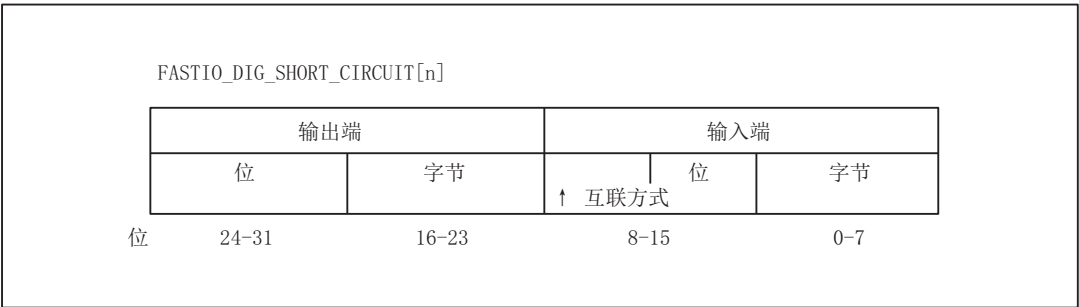
MD10350 \$MN_FASTIO_DIG_NUM_INPUTS

MD10360 \$MN_FASTIO_DIG_NUM_OUTPUTS。

定义输入/输出之间的互联

输入/输出之间的互联是通过以下机床数据定义的：
MD10361 \$MN_FASTIO_DIG_SHORT_CIRCUIT[n]
n：取值范围为 0 至 9，因此最多可指定 10 组互联。
有 2 个十六进制字符用于指定一个输入的字节和位以及一个输出的字节和位。
互联方式在输入的位 12 至 15 中指定，选项有：0、A 和 B：

- 0 直连
- A 逻辑和运算
- B 逻辑或运算



示例

直连：
MD10361 \$MN_FASTIO_DIG_SHORT_CIRCUIT = '04010302H'
输出 4，字节 1 和
输入 3，字节 2 直连

逻辑和运算：
MD10361 \$MN_FASTIO_DIG_SHORT_CIRCUIT = '0705A201H'
输出 7，字节 5 和
输入 2，字节 1 逻辑和运算

逻辑或运算：
MD10361 \$MN_FASTIO_DIG_SHORT_CIRCUIT = '0103B502H'
输出 1，字节 3 和

1.2 通过 PLC 访问

输入 5，字节 2 逻辑或运算

1.2.4 NCK 的模拟量输入/输出

1.2.4.1 NCK 的模拟量输入

功能

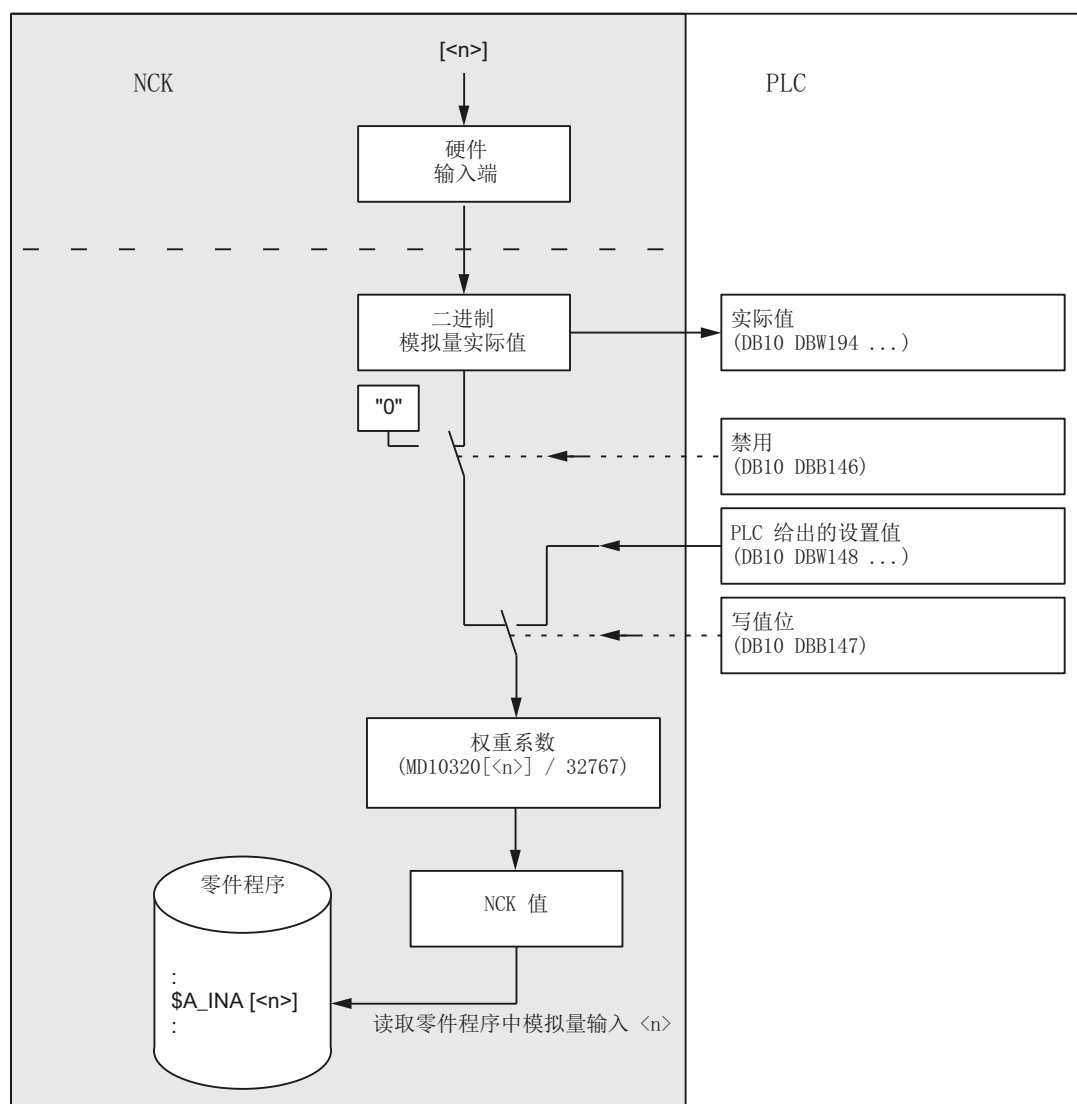
NCK 模拟量输入 <n> 的值可直接在零件程序中通过系统变量 \$A_INA[<n>] 访问。

硬输入上的模拟量值可通过 PLC 用户程序调整。

应用

NCK 模拟量输入尤其适用于磨床和激光加工机床，例如用于 NC 功能“模拟量测量钳”。

信号流



读取实际值

NCK 将硬输入上的实际模拟量值发送至 PLC:

DB10 DBW194 ... 208 (NCK 模拟量输入的实际值)

“实际值”不考虑 PLC 可能造成的影响。

禁用输入

可通过 PLC 用户程序禁用各个 NCK 模拟量输入:

DB10 DBB146 (禁用 NCK 模拟量输入)

此时控制系统内部会将对应输入置为 0。

1.2 通过 PLC 访问

通过 PLC 置位输入

此外还可通过 PLC 为每个 NCK 模拟量输入设定一个值：

DB10 DBB147（通过 PLC 设置 NCK 的模拟量值）

该接口信号置为 1 后，通过 PLC 为相应模拟量输入设定的值立即生效：

DB10 DBW148-162（通过 PLC 设置的 NCK 模拟量输入的设定值）

硬输入上的模拟量值及模拟量输入的禁用状态因而失效。

权重系数

权重系数可按照各种模数转换器（取决于 I/O 模块）调整每个 NCK 模拟量输入，以便在零件程序中读取该输入。

MD10320 \$MN_FASTIO_ANA_INPUT_WEIGHT[<n>]

如果模拟量输入 <n> 已达到最大值或者在 PLC 接口中为该输入设定了 32767，便应该机床数据中输入权重系数，随后在零件程序中编写系统变量 $x = \$A\_INA[<n>]$ 便可读取该输入。之后系统变量 $\$A\_INA[<n>]$ 便会将模拟量输入上的电压值作为单位为微伏的值读取。

说明

非硬件 NCK 模拟量输入上的应用：

权重系数为 32767 时，无论是通过零件程序访问，还是通过 PLC 访问，模拟量值经过模数处理后得出的数值是相同的。这样便可将相应的输入字用于零件程序和 PLC 间的 1:1 通讯。

二进制模拟量值显示

参见“NCK 模拟量输入/输出值的表示法 (页 51)”。

重新上电/复位后的特性

重新上电和复位后各个输入上原有的模拟量值继续保持。必要时可如上文所述通过 PLC 用户程序将各输入禁用或设置设定值。

非硬件 NCK 模拟量值输入

若在零件程序中访问了 MD10300 定义的、但不作为真实硬件存在的 NCK 模拟量输入，系统会读取以下值：

- 通过 PLC 设置的设定值（若接口信号“通过 PLC 设置 NCK 的模拟量值”置为 1 信号）
- 0 V（若接口信号“通过 PLC 设置 NCK 的模拟量值”未置位）

这样便可在无硬件 I/O 的情况下通过 PLC 用户程序使用 NCK 模拟量输入的功能。

1.2.4.2 NCK 的模拟量输出

功能

通过 NCK 模拟量输出可非常快速地输出模拟量值，无须经过 PLC 周期时间。

模拟量输出 <n> 的值可直接在零件程序中通过系统变量 \$A_OUTA[<n>] 设定。

在由 NCK 设定的模拟量值输出到硬件 I/O 上前，该值可由 PLC 修改。

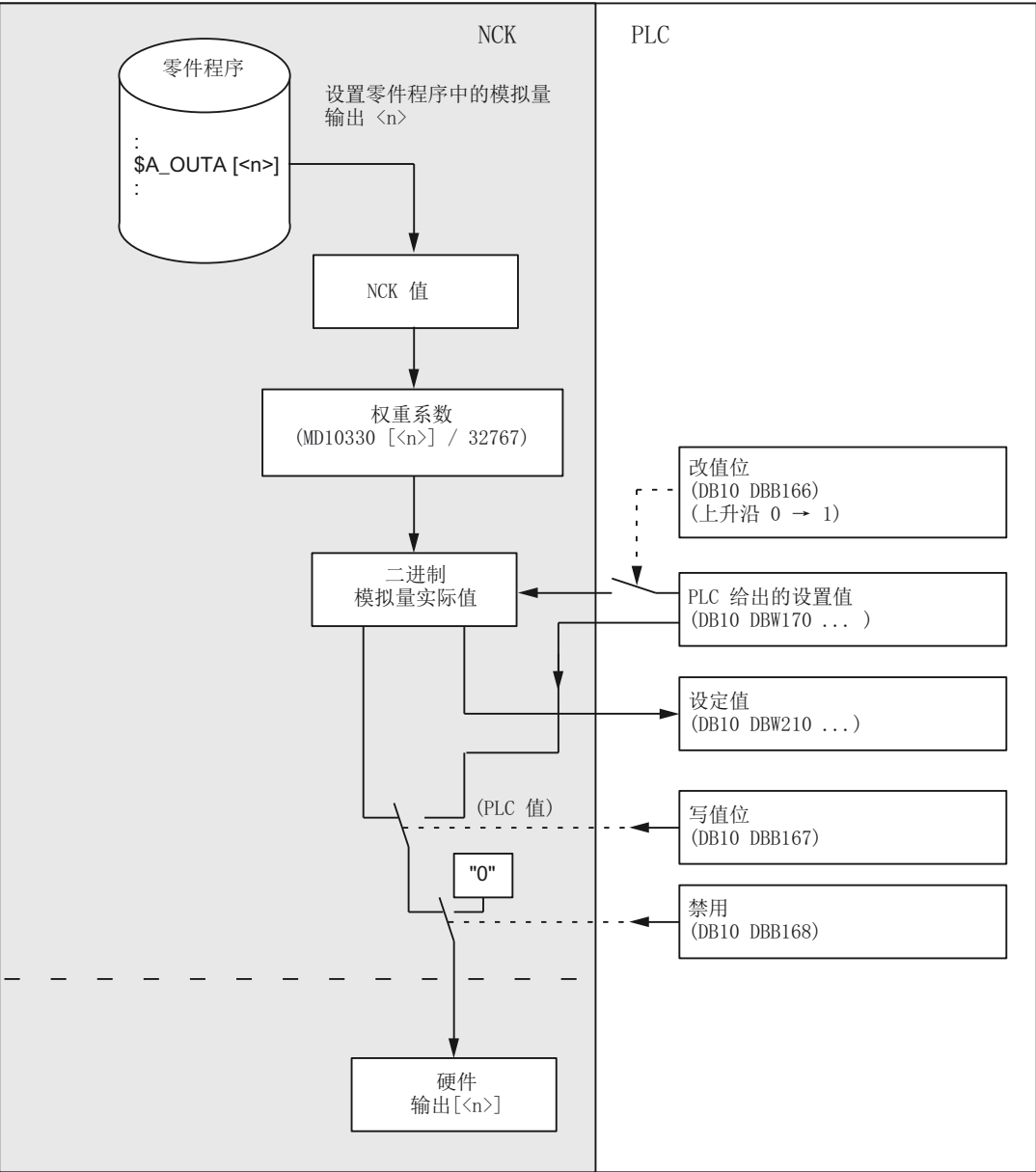
应用

NCK 模拟量输出尤其适用于磨床和激光加工机床。

1.2 通过 PLC 访问

信号流

下图显示了 NCK 模拟量输出的信号流：



改值位

每个已由零件程序设定的 NCK 模拟量值均可由 PLC 借助改值位改写。此时之前的“NCK 值”将丢失。

若需通过 PLC 改写 NCK 值，执行以下步骤：

1. 在 PLC 接口上将相关输出 <n> 预设所需的模拟量值：
DB10 DBW170 ... (通过 PLC 设置的 NCK 模拟量输出 <n> 的设定值)
2. 置位该模拟量输出对应的改值位 (上升沿 0 → 1)，使“PLC 设定值”变为新的“NCK 值”：
DB10 DBB166 (NCK 模拟量输出的改值位)

该值一直生效，直到下一次 PLC 或零件程序给出新的值。

写值位

PLC 既可以使每个输出保持当前比如来自 NC 零件程序的“NCK 值”，也可以通过写值位将“PLC 值”传送给硬件模拟量输出。

若需设定“PLC 值”，执行以下步骤：

1. 在 PLC 接口上将相关输出预设所需的模拟量值：
DB10 DBW170 ... (通过 PLC 设置的 NCK 模拟量输出 <n> 的设定值)
2. 将该模拟量输出对应的写值位置为 1：
DB10 DBB167 (NCK 模拟量输出的写值位)

和改值位不同的是，写值位不会使原先的 NCK 值丢失。一旦 PLC 将该写值位置为 0 后，原先的 NCK 值恢复生效。

说明

无论是改值位还是写值位，PLC 接口上使用的始终是相同的设定值。

禁用输出

可通过 PLC 用户程序禁用各个 NCK 模拟量输出：

DB10 DBB168 (禁用 NCK 模拟量输出)

此时在硬输出上输出“0 信号”。

此时在模拟量输出上输出 0 V

读取设定值

可通过 PLC 用户程序读取模拟量输出的当前“NCK 值”：

DB10 DBW210 ... (NCK 模拟量输出 <n> 的设定值)

此时须注意，此设定值不考虑 PLC 的写值位以及输出的禁用状态。这样一来，设定值可能会与硬输出上的实际模拟量值有别。

权重系数

权重系数可按照各种数模转换器 (取决于 I/O 模块) 调整每个 NCK 模拟量输出，以便在零件程序中编程该输出。

1.2 通过 PLC 访问

MD10330 \$MN_FASTIO_ANA_OUTPUT_WEIGHT[<n>]

在该机床数据中输入一个合适的权重系数，在零件程序中编写 $\$A\_OUTA[n] = x$ 后便可以使对应的模拟量输出 <n> 达到最大值或在 PLC 接口中为该输出设定 32767。之后该模拟量输出上通过系统变量 $\$A\_OUTA[<n>]$ 设定的值会变为单位为微伏的相同电压值。

示例：

模拟量值范围为 10 V（最大范围）：

MD10330 \$MN_FASTIO_ANA_OUTPUT_WEIGHT[<n>] = 10000（缺省值）

$\$A\_OUTA[1] = 9500$ ；在 NCK 模拟量输出 1 上输出 9.5 V

$\$A\_OUTA[3] = -4120$ ；在 NCK 模拟量输出 3 上输出 -4.12 V

说明

非硬件 NCK 模拟量输出上的应用：

权重系数为 32767 时，无论是通过零件程序访问，还是通过 PLC 访问，模拟量值经过模数处理后得出的数值是相同的。这样便可将相应的输出字用于零件程序和 PLC 间的 1:1 通讯。

二进制模拟量值显示

参见“NCK 模拟量输入/输出值的表示法 (页 51)”。

程序结束/复位后的特性

在程序结束或复位后，每个模拟量输出都可在 PLC 用户程序中根据需求利用改值位、写值位或禁用信号设置。

重新上电后的特性

重新上电后模拟量输出都会置为 0。这一信号状态可针对具体应用在 PLC 用户程序中利用改值位和写值位改写。

非硬件 NCK 模拟量输出

若在零件程序中写入了 MD10310 定义的、但不作为真实硬件存在的 NCK 模拟量输出，系统不会发出报警。NCK 值可通过 PLC 读取（DB10 DBB210 ...）。

1.2.4.3 NCK 模拟量输入/输出值的表示法

模拟量值经过模数转换后在 NC/PLC 接口上作为定点数（16 位，含符号）以补码表示。

位编号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
含义	VZ	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

VZ: 符号

最小值	最大值
-32768_D	32767_D
8000_H	$7FFF_H$

步长

模数转换精度为 16 位、额定范围为 $\pm 10\text{ V}$ 的情况下，步长为：

$$20\text{ V} / 2^{16} = 20\text{ V} / 65536 \approx 0.305\text{ mV}$$

模数转换精度 < 16 位

若模拟量模块的模数转换精度小于 16 位（包含符号），那么模拟量值经过模数转换后的数值将会从位 14 起进入接口。未占用的低位将填 0。

14 位模数转换精度

模数转换精度为 14 位（包含符号），额定范围为 $\pm 10\text{ V}$ 的情况下，步长为：

$$20\text{ V} / 2^{14} = 20\text{ V} / 16384 \approx 1.22\text{ mV}$$

位 0 和位 1 总是为 0。

12 位模数转换精度

模数转换精度为 12 位（包含符号），额定范围为 $\pm 10\text{ V}$ 的情况下，步长为：

$$20\text{ V} / 2^{12} = 20\text{ V} / 4096 \approx 4.88\text{ mV}$$

位 0 到 3 总是为 0。

不同模数转换精度下最大值的表示

位编号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
含义	VZ	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

1.2 通过 PLC 访问

16 位模数转换精度: $32767_D = 7FFF_H$	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14 位模数转换精度: $8191_D = 1FFF_H$	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
12 位模数转换精度: $2047_D = 7FF_H$	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0

说明
所使用的模拟量输入/输出模块的数据（模数转换精度、额定范围）请见相应模块的文档。

示例

模数转换精度为 14 位（包含符号），额定范围为 $\pm 10\text{ V}$ 时模拟量值经过模数转换后的数值表方法：

示例 1：模拟量值 = 9.5 V

经过模数转换后的十进制表示：	$9.5\text{ V} / 20\text{ V} * 16384 = 7782$
经过 14 位模数处理后的二进制表示：	01 1110 0110 0110
经过 16 位模数处理后的二进制表示：	0111 1001 1001 1000
经过 16 位模数处理后的十六进制表示：	7998 _H

示例 2：模拟量值 = -4.12 V

经过模数转换后的十进制表示：	$-4.12\text{ V} / 20\text{ V} * 16384 = -3375$
经过 14 位模数处理后的二进制表示：	11 0010 1101 0001
经过 16 位模数处理后的二进制表示：	1100 1011 0100 0100
经过 16 位模数处理后的十六进制表示：	CB44 _H

1.2.5 比较器输入

功能

除了 NCK 数字量/模拟量输入外，系统还提供 2 个内部比较器输入字节，每个字节各带有 8 个比较器输入。比较器输入的信号状态是通过对比高速模拟量输入信号与设定数据中可调阈值得出的。

比较器输入 $\langle n \rangle$ 的信号状态（即比较结果）可直接在零件程序中通过系统变量 $\$A\_INCO[\langle n \rangle]$ 查询。

下标 $\langle n \rangle$ ：

$\langle n \rangle = 1 \dots 8$ 用于比较器字节 1

$\langle n \rangle = 9 \dots 16$ 用于比较器字节 2

术语

在本段说明中涉及到了**比较器输入**（带下标 $\langle n \rangle$ ； $\langle n \rangle$ 的取值范围为 1 ... 8 或 9 ... 16）和**比较器输入位**（带下标 $\langle b \rangle$ ； $\langle b \rangle$ 的取值范围为 0 ... 7）这两个术语。

这两个术语有以下关联：

$\langle n \rangle = 1 \dots 8$ ： 比较器输入 $\langle n \rangle$ 与比较器输入位的对应关系： $\langle b \rangle = \langle n \rangle - 1$

$\langle n \rangle = 9 \dots 16$ ： 比较器输入 $\langle n \rangle$ 与比较器输入位的对应关系： $\langle b \rangle = \langle n \rangle - 9$

示例： 比较器输入 1 对应比较器输入位 0。

指定模拟量输入

通过以下机床数据为比较器字节 1 的输入位 $\langle b \rangle$ 指定一个模拟量输入：

MD10530 \$MN_COMPAR_ASSIGN_ANA_INPUT_1[$\langle b \rangle$]

示例：

MD10530 \$MN_COMPAR_ASSIGN_ANA_INPUT_1[0] = 1

MD10530 \$MN_COMPAR_ASSIGN_ANA_INPUT_1[1] = 1

MD10530 \$MN_COMPAR_ASSIGN_ANA_INPUT_1[7] = 7

模拟量输入 1 作用于比较器字节 1 的输入位 0 和 1。

模拟量输入 7 作用于比较器字节 1 的输入位 7。

1.2 通过 PLC 访问

为比较器字节 2 指定模拟量输入则通过以下机床数据进行：

MD10531 \$MN_COMPAR_ASSIGN_ANA_INPUT_2[]

比较器设置

比较器字节 1 或 2 的各个位（0 到 7）通过以下机床数据设置：

MD10540 \$MN_COMPAR_TYPE_1（比较器字节 1 的设置）

或者

MD10541 \$MN_COMPAR_TYPE_2（比较器字节 2 的参数）

可以进行下列设置：

- 比较类型位（位 0 ... 7）

为每个比较器输入位定义比较条件的类型：

位 = 1: 满足以下条件时，对应的比较器输入位置 1：

模拟量值 $\geq$ 阈值

位 = 0: 满足以下条件时，对应的比较器输入位置 0：

模拟量值 $<$ 阈值

- 通过 NCK 数字量输出输出比较器输入字节（位 16 ... 23）

比较器位还可通过 NCK 数字量输出以字节方式直接输出。为此，必须在此字节中（位 16 ... 23）设定通过哪个 NCK 数字量输出字节输出。

字节 = 0: 不通过 NCK 数字量输出输出

字节 = 1: 通过 NCK 板载的数字量输出 9 ... 16 输出

字节 = 2: 通过外部 NCK 数字量输出 17 ... 24 输出

字节 = 3: 通过外部 NCK 数字量输出 25 ... 32 输出

字节 = 4: 通过外部 NCK 数字量输出 33 ... 40 输出

- 用于输出比较器输入字节的取反位（位 24 ... 31）

对每个比较器信号可额外定义，是否对在 NCK 数字量输出上输出的信号状态进行取反。

位 = 1: 不取反对应的比较器输入位。

位 = 0: 取反对应的比较器输入位。

阈值

在比较器字节 1 或 2 上用于比较的阈值必须作为设定数据保存。必须为每个比较器输入位 ****（其中 ** = 0 ... 7**）输入一个独立的阈值：

SD41600 \$SN_COMPAR_THRESHOLD_1[****]

或者

SD41601 \$SN_COMPAR_THRESHOLD_2[****]

比较器信号作为 NCK 数字量输入

所有执行顺序取决于 NCK 数字量输入的 NC 功能均可通过比较器的信号状态进行控制。此时须在 NC 功能对应的机床数据（“指定使用的硬件字节”）中为比较器字节 1（硬件字节 128）或 2（硬件字节 129）输入字节地址。

示例：

NC 功能“在一个程序段内编写多个进给率”。

通道专用机床数据中的输入：

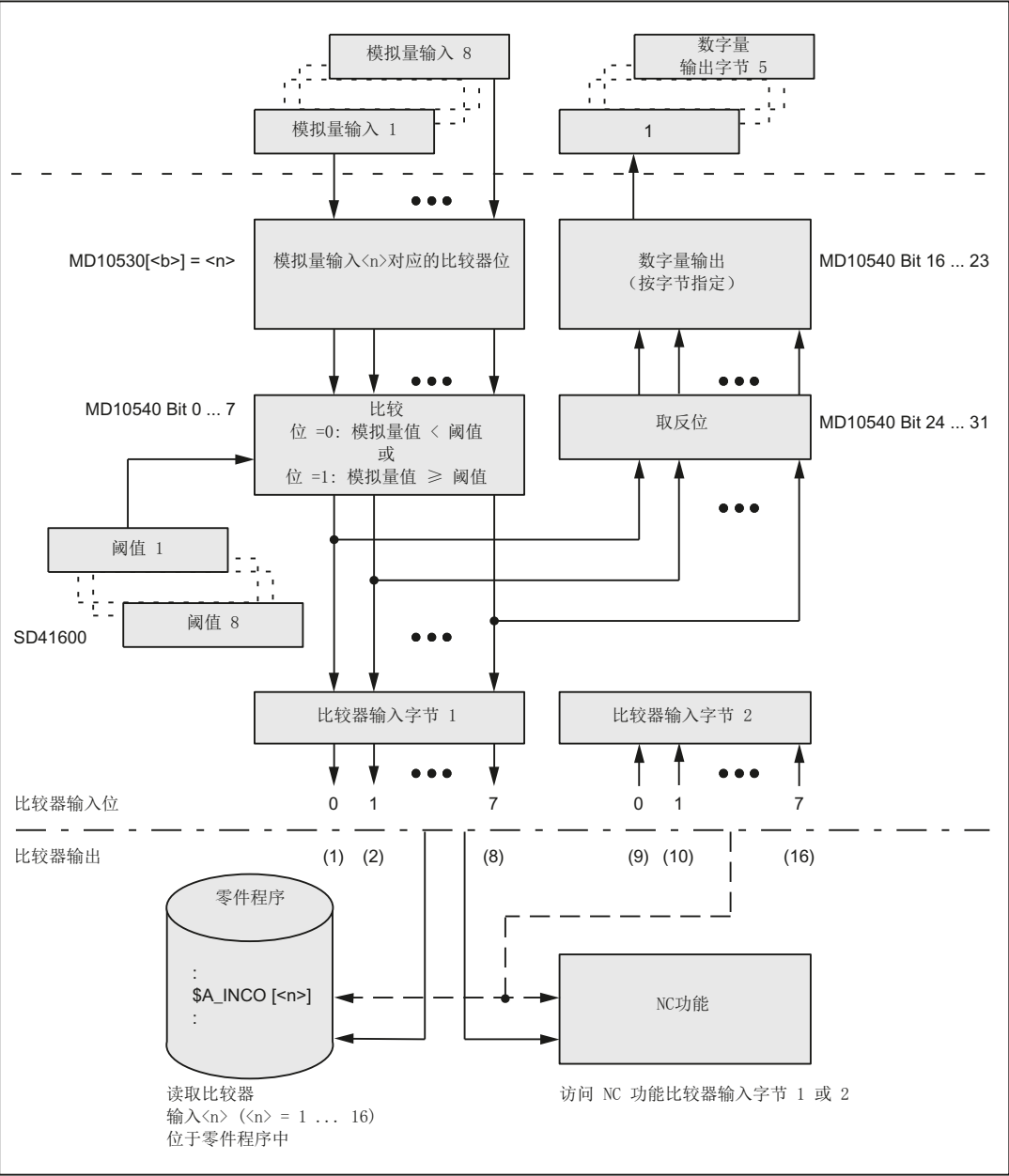
MD21220 \$MC_MULTFEED_ASSIGN_FASTIN = 129

这样便会根据比较器字节 2 的状态激活不同的进给率值。

1.2 通过 PLC 访问

工作流程

下图显示了比较器输入字节 1 的工作流程。



1.2.6 可由 NC 直接访问的 PLC 输入/输出

1.2.6.1 功能

此处预设的 NCK 和 PLC 输入/输出之间的高速数据通道是直接由 PLC 操作系统处理的。

PLC 基本程序 and 用户程序无法对其进行控制。NCK 和 PLC 对同一 PLC 输入/输出的并行访问无意义，且可能导致故障。

数据交换

NC 是通过零件程序和同步动作中的系统变量访问 PLC 的。

表格 1-2 读 PLC:

- \$A_PBB_IN[<n>] 读取字节 (Byte, 8 位)
- \$A_PBW_IN[<n>] 读取字 (Word, 16 位)
- \$A_PBD_IN[<n>] 读取双字 (DWord, 32 位)
- \$A_PBR_IN[<n>] 读取实数 (Real, 32 位浮点值)

<n> = PLC 输入范围内的字节偏移

在从零件程序读取时会触发预处理停止。

表格 1-3 写 PLC:

- \$A_PBB_OUT[<n>] ; 写入字节 (Byte, 8 位)
- \$A_PBW_OUT[<n>] ; 写入字 (Word, 16 位)
- \$A_PBD_OUT[<n>] ; 写入双字 (DWord, 32 位)
- \$A_PBR_OUT[<n>] ; 写入实数 (Real, 32 位浮点数)

<n> = PLC 输出范围内的字节偏移

输出数据也可从零件程序和同步动作读取。在从零件程序读取时会自动触发一次预处理停止 (以实现与实时环境的同步)。

1.2 通过 PLC 访问

取值范围

系统变量可取以下范围内的值：

系统变量	取值范围
\$A_PBB_OUT[<n>]:	(-128 ... +127) 或 (0 ... 255)
\$A_PBW_OUT[<n>]:	(-32768 ... +32767) 或 (0 ... 65535)
\$A_PBD_OUT[<n>]:	(-2147483648 ... +2147483647) 或 (0 ... 4294967295)
\$A_PBR_OUT[<n>]:	(-3.402823466E+38 ... +3.402823466E+38)

传输时间点

只要 NCK 写入了一个数据，该数据便会在一个插补周期结束时便传输给 PLC。

读 PLC 时，NCK 在一个插补周期末尾发出请求“Request”。

读出的新数据最早在下一个插补周期中提供给 NCK。

可通过机床数据设定 NCK 向 PLC 发出请求持续的时间：

MD10398 \$MN_PLCIO_IN_UPDATE_TIME

输入的时间会在系统内部向上取整为插补周期的整倍数值。若此机床数据的值设为 0，那么会在每个插补周期中 NCK 都会向 PLC 发出请求。

配置

为了激活功能，必须在 NC 中配置以下机床数据（上电生效）：

- MD10394 \$MN_PLCIO_NUM_BYTES_IN
直接通过 NC 读取的 PLC I/O 输入字节的数量。
- MD10395 \$MN_PLCIO_LOGIC_ADDRESS_IN
PLC 输入的逻辑初始地址，从该地址起读取数据。
- MD10396 \$MN_PLCIO_NUM_BYTES_OUT
直接通过 NC 写入的 PLC I/O 输出字节的数量。
- MD10397 \$MN_PLCIO_LOGIC_ADDRESS_OUT
PLC 输出的逻辑初始地址，从该地址起写入数据。
- MD10398 \$MN_PLCIO_IN_UPDATE_TIME
可通过 \$A_PBx_IN 读取的输出更新的间隔（参见“传输时间点”一段）。
- MD10399 \$MN_PLCIO_TYPE_REPRESENTATION
系统变量 \$A_PBx_OUT 和 \$A_PBx_IN 的表示法（参见“存储顺序”）。

说明

机床数据中输入的 PLC I/O 的逻辑地址以及待传输字节的数量必须与 PLC 硬件配置一致。
在配置的 PLC I/O 区中不允许出现“地址缺口”。

存储顺序

系统为 NCK 和 PLC 之间两个方向的数据交换分别提供 16 个字节（跨所有通道）。这些区域须由用户自行管理（即变量不可重叠，也不可超出通道界限！）。

这些区域内的变量表示法取决于以下机床数据中的设置：

MD10399 \$MN_PLCIO_TYPE_REPRESENTATION

值	含义
0	（缺省设置） 用 Little-Endian 小端字节序来表示系统变量 （即低序字节存储在起始地址）
1	（PLC 标准格式，推荐使用） 用 Big-Endian 大端字节序来表示系统变量 （即高序字节存储在起始地址）

Big-Endian 格式是 PLC 中常用的表示方式（即也适用于 PLC 输入/输出），因此被广泛使用。

注意：

- PLC I/O 输入和输出区域分配必须满足以下条件：

\$A_PBB_IN[<j>]	; j < ([MD10394 \$MN_PLCIO_NUM_BYTES_IN])
\$A_PBW_IN[<j>]	; j < ([MD10394 \$MN_PLCIO_NUM_BYTES_IN] - 1)
\$A_PBD_IN[<j>]	; j < ([MD10394 \$MN_PLCIO_NUM_BYTES_IN] - 3)
\$A_PBR_IN[<j>]	; j < ([MD10394 \$MN_PLCIO_NUM_BYTES_IN] - 3)
\$A_PBB_OUT[<k>]	; k < ([MD10396 \$MN_PLCIO_NUM_BYTES_OUT])
\$A_PBW_OUT[<k>]	; k < ([MD10396 \$MN_PLCIO_NUM_BYTES_OUT] - 1)
\$A_PBD_OUT[<k>]	; k < ([MD10396 \$MN_PLCIO_NUM_BYTES_OUT] - 3)
\$A_PBR_OUT[<k>]	; k < ([MD10396 \$MN_PLCIO_NUM_BYTES_OUT] - 3)

1.2 通过 PLC 访问

- 不允许超出用于数据交换的最大可用字节数。

1.2.6.2 边界条件

配置

- 需要通过高速数据通道来读写 PLC 输入/输出时，输入/输出必须配置为一个连续的地址块，即其中没有地址缺口。
- 待传输字节的数量必须在 PLC 输入/输出上无缝映射。

时间特性

从 PLC 输入/输出读数据的时间点和通过系统变量向零件程序提供数据的时间点不同步！

数据传输 (NCK ↔ PLC)

- 即使在数据缓存内只有一个系统变量，数据缓存也总是会完整输出给 PLC 输入/输出。
- 若同时对多个系统变量进行了赋值（例如用于初始化 PLC 输入/输出），则无法确保这些数值能够在同一个插补周期内传输。

1.2.6.3 示例

写 PLC 输入/输出

此示例假定以下条件：

- 需要将数据直接输出至以下 PLC 输入/输出：
 - 逻辑地址 521: 8 位数字量输出模块
 - 逻辑地址 522: 16 位数字量输出模块
- 由同步动作通过 \$A_PBx_OUT 输出。

参数设置

必须如下设置机床数据：

MD10397 \$MN_PLCIO_LOGIC_ADRESS_OUT = 521 ; 从逻辑地址 521 起输出数据
 MD10396 \$MN_PLCIO_NUM_BYTES_OUT = 3 ; 总共须输出 3 个字节
 MD10399 \$MN_PLCIO_TYPE_REPRESENTATION = 1; 数据以 Big-Endian 格式表示

启动 NCK 和 PLC

NCK 和 PLC 启动后, NCK 不会向 PLC I/O 循环传送数据以执行写访问。

编程

载入和启动包含以下内容的零件程序:

程序代码	注释
...	
ID=1 WHENEVER TRUE DO \$A_PBB_OUT[0]=123	; 循环输出 (每个插补周期)
...	
ID=2 WHEN \$AA_IW[x] >= 5 DO \$A_PBW_OUT[1]='Habcd'	; 输出一个十六进制值
...	

读 PLC 输入/输出

此示例假定以下条件:

- PLC 输入/输出:

- 逻辑地址 420: 16 位模拟量输入模块
- 逻辑地址 422: 32 位数字量输入模块
- 逻辑地址 426: 32 位输入 DP 从站
- 逻辑地址 430: 8 位数字量输入模块

- PLC 输入/输出是在零件程序中通过 R 参数中的 \$A_PBx_IN 读取的。
- 为了避免 PLC 用户程序 (OB1) 的执行速度减慢, 之前已通过 MD10398 \$MN_PLCIO_IN_UPDATE_TIME 选择了合适的更新时间 (用于读访问), 确保每三个插补周期更新一次。

参数设置

必须如下设置机床数据:

MD10395 \$MN_PLCIO_LOGIC_ADRESS_IN = 420 ; 从逻辑地址 420 起读取数据
 MD10394 \$MN_PLCIO_NUM_BYTES_IN = 11 ; 总共须读取 11 个字节

1.3 通过 PROFIBUS 访问

MD10398 \$MN_PLCIO_IN_UPDATE_TIME = 0.03 ; 更新时间 = 30 ms
(插补周期 = 12 ms)

MD10399 \$MN_PLCIO_TYPE_REPRESENTATION = 1 ; 数据以 Big-Endian 格式表示

启动 NCK 和 PLC:

现在 NCK 和 PLC 启动后将每三个插补周期更新一次, 以便进行读访问。

编程

载入和启动包含以下内容的零件程序:

程序代码	注释
...	
R1=\$A_PBW_IN[0]	; 读取 16 位整数
R2=\$A_PBD_IN[2]	; 读取 32 位整数
R3=\$A_PBR_IN[6]	; 读取 32 位浮点数
R4=\$A_PBB_IN[10]	; 读取 8 位整数
...	

1.3 通过 PROFIBUS 访问

1.3.1 简要说明

功能

“通过 PROFIBUS 访问 I/O” 功能可实现 NCK 和 PROFIBUS I/O 间的直接数据交换。

可用性

此功能适用于等时和非等时配置的 PROFIBUS I/O。

硬件前提条件

- 所需的 PROFIBUS I/O 存在且运行就绪。
- 在 PLC 侧正确完成了 S7 硬件配置, 其中包含了所需的 PROFIBUS I/O, 并且该配置已经载入到 PLC 中,
- 相同的 PROFIBUS 从站上必须有一个 I/O 区域。

I/O 区域

在下文中涉及的 I/O 区域是指：由在 S7-HW-Config 中（PLC）配置的 PROFIBUS 从站的各个有效数据插槽组成的一个连续的、没有缺口的 PROFIBUS I/O 区域，该区域有一个逻辑起始地址，且地址依次上升。

I/O 区域的特征如下：

- 有一个逻辑起始地址（即该 I/O 区域第一个有效数据插槽的起始地址）
- 有一段经过配置的长度（即用于访问连续 PROFIBUS I/O 的字节长度）

NCK 必须获知 I/O 区域的逻辑起始地址，这样 NCK 才能通过其内部的一个 PROFIBUS 通讯接口读/写对应 PROFIBUS I/O 的数据。I/O 区域的配置通过机床数据进行（参见“I/O 区域的配置 (页 64)”）。因此 NCK 与 PROFIBUS I/O 的通讯只能以 I/O 区域为基准。

数据交换

NCK 与 PROFIBUS 输入/输出的数据交换是通过 NCK 内部的一个 PROFIBUS 通讯接口实现的。NCK 和 PROFIBUS I/O 之间的数据交换方式有：

- 零件程序/同步动作按插补周期读/写 NCK 系统变量（\$A_DPx_IN[n,m] 或 \$A_DPx_OUT[n,m]）（数据一致性）。对应的插补周期结束后才会将待写入的 PROFIBUS I/O 数据输出至 PROFIBUS I/O。
参见“通过零件程序/同步动作访问 (页 66)”。
- 编译循环接口读/写数据块（持续一个伺服周期的数据一致性）
参见“通过编译循环进行读/写 (页 69)”。

并行数据访问

在对 I/O 区域进行了相应配置的前提下，编译循环和零件程序/同步动作可并行读同一个 I/O 区域的数据。但是需要注意的是，此时会读取 PROFIBUS I/O 数据的不同映射区。一个映射区内的数据一致性可以确保。但是一个插补周期内映射区之间的数据一致性无法确保。

编译循环和零件程序/同步动作不可并行向同一个 I/O 区域写入数据。在配置 NCK 时，须定义将某个 PROFIBUS I/O 区域指定给系统变量还是编译循环。

激活

此功能在 NCK 启动期间激活。

1.3 通过 PROFIBUS 访问

1.3.2 I/O 区域的配置

I/O 区域通过机床数据进行配置。配置的数据无法再在 NCK 正常工作模式中修改。

系统在读方向提供 16 个 I/O 区域，在写方向提供 16 个 I/O 区域。I/O 区域的最大长度由 NCK 限制为 128 字节。

若 I/O 区域激活过一次，系统会在每个插补周期开头检查该区域是否仍可用。为此系统会分析 I/O 区域中一个有效数据插槽的生命符号。若该生命符号在插补周期开始时未置位，则会发出报警（9050 或 9052）。此报警不会导致零件程序暂停，仅用于提示，直至对应的 I/O 区域重新提供有效的生命符号。

逻辑起始地址

NCK 必须获知 I/O 区域的逻辑起始地址，才能使用该区域。为此须设置下列机床数据：

MD10500 \$MN_DPIO_LOGIC_ADDRESS_IN[<n>]

MD10510 \$MN_DPIO_LOGIC_ADDRESS_OUT[<n>]

<n> = 表示 I/O 区域的下标

I/O 区域的长度

为了使 NCK 能够检查 I/O 区域是否完整配置，必须输入对应 I/O 区域所需的长度（字节）。为此须设置下列机床数据：

MD10501 \$MN_DPIO_RANGE_LENGTH_IN[<n>]

MD10511 \$MN_DPIO_RANGE_LENGTH_OUT[<n>]

若输入了长度 0，则系统只会将对应逻辑起始地址下的有效数据插槽配置为 I/O 区域。此时 I/O 区域的长度将设为找到的有效数据插槽的长度。

更多属性

通过下列机床数据可为每个 I/O 区域指定更多属性：

MD10502 \$MN_DPIO_RANGE_ATTRIBUTE_IN[<n>]

位	值	含义
0		系统变量 \$A_DPx_IN[<n>,<m>] 的表示方式
	0	Little-Endian 格式
	1	Big-Endian 格式
1		预留

位	值	含义
2		读取输入数据
	0	可通过系统变量和 CC-Binding 读取
	1	只可通过 CC-Binding 读取
3		输出槽生命符号报警
	0	输出槽生命符号报警
	1	抑制槽生命符号报警

MD10512 \$MN_DPIO_RANGE_ATTRIBUTE_OUT[<n>]

位	值	含义
0		系统变量 \$A_DPx_OUT[<n>,<m>] 的表示方式
	0	Little-Endian 格式
	1	Big-Endian 格式
1		写入输出数据
	0	仅通过系统变量写入
	1	仅通过 CC-Binding 写入
2		预留
3		输出槽生命符号报警
	0	输出槽生命符号报警
	1	抑制槽生命符号报警

注意:

- 通过机床数据配置 I/O 区域不需要保持连贯。即可为相应的机床数据下标任意指定 I/O 区域。
- 若 NCK 启动阶段无法成功找到一个/多个 I/O 区域，则会中断查找并发出报警（4700 或 4702）。

1.3 通过 PROFIBUS 访问

1.3.3 数据交换

若需通过 NCK 内部的 PROFIBUS 通讯接口实现与 PROFIBUS I/O 的直接数据交换，必须满足以下前提条件：

- 对应 I/O 区域的配置正确。
- PLC 实际上也具备所请求的 I/O 区域（有效数据插槽）。
- PROFIBUS 通讯接口第一次和对应 PROFIBUS I/O 成功进行数据交换后，配置的 I/O 区域才会投入使用。

1.3.3.1 通过零件程序/同步动作访问

简介

借助 NCK 全局系统变量可从零件程序/同步动作访问 PROFIBUS I/O，无论是读访问还是写访问。

- \$A_DPx_IN [<n>,<m>]
- \$A_DPx_OUT [<n>,<m>]

此时须注意：

- 在从零件程序读/写这些变量时会触发预处理停止。
- 为了在从零件程序和同步动作编程时确保数据一致性，系统会访问在对应的插补周期保持恒定的 PROFIBUS I/O 数据。
- 若在一个插补周期内多次对同一个 PROFIBUS I/O 进行写访问（例如同步动作中通过不同的通道访问），则最后一次写访问的数据生效。
- 对应的插补周期结束后才会将待写入的 PROFIBUS I/O 数据输出至 PROFIBUS I/O。

访问 I/O 区域

下列系统变量用于访问 I/O 区域:

表格 1-4 NCK → PROFIBUS I/O

系统变量	值	含义
\$A_DPB_OUT[<n>,<m>]]	8 位, 无符号 (unsigned)	将一个数据字节 (8 位) 写入 PROFIBUS-IO
\$A_DPW_OUT[<n>,<m>] >]	16 位, 无符号 (unsigned)	将一个数据字 (16 位) 写入 PROFIBUS-IO
\$A_DPSB_OUT[<n>,<m>]]	8 位, 有符号 (signed)	将一个数据字节 (8 位) 写入 PROFIBUS-IO
\$A_DPSW_OUT[<n>,<m>]]	16 位, 有符号 (signed)	将一个数据字 (16 位) 写入 PROFIBUS-IO
\$A_DPSD_OUT[<n>,<m>]]	32 位, 有符号 (signed)	将一个数据双字 (32 位) 写入 PROFIBUS-IO
\$A_DPR_OUT[<n>,<m>]]	32 位实数	将输出数据 (32 位实数) 写入 PROFIBUS-IO

<n> = 表示输出数据区域的下标; <m> = 表示数据字节的下标

表格 1-5 PROFIBUS I/O → NCK

系统变量	值	含义
\$A_DPB_IN[<n>,<m>]]	8 位, 无符号 (unsigned)	从 PROFIBUS-IO 读取一个数据字节 (8 位)
\$A_DPW_IN[<n>,<m>]]	16 位, 无符号 (unsigned)	从 PROFIBUS-IO 读取一个数据字 (16 位)
\$A_DPSB_IN[<n>,<m>]]	8 位, 有符号 (signed)	从 PROFIBUS-IO 读取一个数据字节 (8 位)
\$A_DPSW_IN[<n>,<m>]]	16 位, 有符号 (signed)	从 PROFIBUS-IO 读取一个数据字 (16 位)
\$A_DPSD_IN[<n>,<m>]]	32 位, 有符号 (signed)	从 PROFIBUS-IO 读取一个数据双字 (32 位)
\$A_DPR_IN[<n>,<m>]]	32 位实数	从 PROFIBUS-IO 读取输出数据 (32 位实数)

<n> = 表示输入数据区域的下标; <m> = 表示数据字节的下标

1.3 通过 PROFIBUS 访问

检查 I/O 区域的配置

可通过下列系统变量检查 I/O 区域的配置。此位域的每个位均对应一个 I/O 区域。若为零件程序/同步动作访问配置了 I/O 区域，则对应位会置位。

系统变量	值	含义
\$A_DP_IN_CONF	32 位位域	读取所有配置的 PROFIBUS 输入区
\$A_DP_OUT_CONF	32 位位域	读取所有配置的 PROFIBUS 输出区

检查 I/O 区域的可用性

可通过下列系统变量检查 I/O 区域的可用性。此位域的每个位均对应一个 I/O 区域。若 I/O 区域可用于零件程序/同步动作访问，则对应位会置位。

系统变量	值	含义
\$A_DP_IN_VALID	32 位位域	读取所有有效的 PROFIBUS 输入区
\$A_DP_OUT_VALID	32 位位域	读取所有有效的 PROFIBUS 输出区

查询 I/O 区域的状态

通过下列系统变量可查询一个 I/O 区域的准确状态。

系统变量	值	含义
\$A_DP_IN_STATE[<n>] <n> = 表示输入数据区域的下标	0: 数据区域未经过配置 1: 数据区域尚且无法激活 2: 数据区域可用 3: 数据区域当前不可用	读取输入数据区域的状态
\$A_DP_OUT_STATE[<n>] <n> = 表示输出数据区域的下标		读取输出数据区域的状态

查询 I/O 区域的长度

通过下列系统变量可查询一个 I/O 区域配置的长度。

系统变量	含义
\$A_DP_IN_LENGTH[<n> <n> = 表示输入数据区域的下标	读取输入数据区域的长度
\$A_DP_OUT_LENGTH[<n> <n> = 表示输出数据区域的下标	读取输出数据区域的长度

注意：

- 通过 <n>（区域下标）可选择相应的 NCK 配置的 I/O 区域。若请求的 I/O 区域未经过配置，则会发出报警（17020）来显示。
- 通过 <m>（区域偏移）可指定在一个 I/O 区域中从哪个位置开始访问数据（字节偏移）。数据类型可在 I/O 区域内的任意字节偏移上读/写。若读/写访问超出了对应 I/O 区域的界限，则会被系统拒绝并发出报警（17030）。
- 通过机床数据 MD10502 \$MN_DPIO_RANGE_ATTRIBUTE_IN 和 MD10512 \$MN_DPIO_RANGE_ATTRIBUTE_OUT（参见“I/O 区域的配置 (页 64)”），可针对单独的读/写方向或任意一个 I/O 区域确定系统变量 \$A_DPx_IN[<n>,<m>] 或 \$A_DPx_OUT[<n>,<m>] 的表示方式（Little-/Big-Endian）。

1.3.3.2 通过编译循环进行读/写

简介

CC-Binding 用于通过编译循环接口读取/输出数据块。对 I/O 区域数据的访问在伺服任务级进行。数据在每个伺服周期中更新一次。因此只在一个伺服周期中能确保数据一致性。

在 NCK 配置中必须使能对应的 I/O 区域，才能在编译循环中通过 CC-Binding 对 I/O 区域的数据进行写访问：

MD10512 \$MN_DPIO_RANGE_ATTRIBUTE_OUT[<n>], 位 1 = 1

若零件程序/同步动作中同时编写了此 I/O 区域，系统会抑制该编程并发出报警（17020）。

需要注意的是，PLC 中数据通常采用 Big-Endian 格式表示。PROFIBUS I/O 也同样如此。由于 Binding 只能以字节为基础访问一个 I/O 区域（字节偏移、待传输字节的数量），因此必须确保数据类型（16 位、32 位等）的正确表示。

CC-Binding

下列 CC-Binding 可供使用:

- CCDataOpi: getDploRangeConfiguration()
- CCDataOpi: getDploRangeValid()
- CCDataOpi: getDploRangeInInformation()
- CCDataOpi: getDploRangeOutInformation()
- CCDataOpi: getDploRangeInState()
- CCDataOpi: getDploRangeOutState()
- CCDataOpi: getDataFromDploRangeIn()
- CCDataOpi: putDataToDploRangeOut()

注意:

- 进行读/写访问时, Binding
CCDataOpi: getDataFromDploRangeIn() 或
CCDataOpi: putDataToDploRangeOut()
会监控是否遵循了 NCK 侧和 PLC 侧配置的对应 I/O 区域的界限。若访问的数据/数据区域不完全位于配置的 I/O 区域界限内, 那么访问将被拒绝, 并输出枚举器反馈 CCDataSTATUS_RANGE_LENGTH_LIMIT。
- 若尝试访问未经配置 (或未配置用于编译循环) 的 I/O 区域, 则会输出枚举器反馈 CCDataSTATUS_RANGE_NOT_AVAILABLE。

注意
可延缓系统运行速度 编译循环的编写人员须自行确保 CC-Binding 的正确使用! 注意, 在伺服任务级提供配置 I/O 区域的数据会额外占用性能, 须确保该占用不会造成伺服级计算时间上溢。更多有关 Binding 应用的详细信息请见 OEM 文档。

1.3.4 边界条件

硬件

- 仅支持 PLC 的第一条真实 PROFIBUS 支路的 PROFIBUS 从站（连接器标记有 DP1）。

软件

- 不允许再通过 PLC 用户程序并行写 PROFIBUS I/O，不过目前从技术上还无法阻止此操作。
- 设定用于写 PROFIBUS I/O 的 I/O 区域（MD10510 \$MN_DPIO_LOGIC_ADDRESS_OUT[]）不允许位于 PLC 的 I/O 映射区域内（例如 PLC 317，地址 0 - 255），因为此区域由 PLC 操作系统占用。

1.3.5 示例

1.3.5.1 写访问中的 PROFIBUS I/O

前提条件

已进行 S7 硬件配置。

通过零件程序/同步动作访问的配置

- 区域下标 = 5（NCK 内部配置）
- 根据 S7 硬件配置：
 - 逻辑起始地址 = 334
 - 槽长度 = 8 个字节
 - 以 Little-Endian 格式表示

对应的机床数据配置如下：

MD10510 \$MN_DPIO_LOGIC_ADDRESS_OUT[5] = 334（I/O 区域的逻辑起始地址）

MD10511 \$MN_DPIO_RANGE_LENGTH_OUT[5] = 8（I/O 区域的长度）

MD10512 \$MN_DPIO_RANGE_ATTRIBUTE_OUT[5]

位 0 = 0（Little-Endian 格式）

1.3 通过 PROFIBUS 访问

- 位 1 = 0 （只通过系统变量写入）
- 位 3 = 0 （输出槽生命符号报警）

通过编译循环访问的配置

- 区域下标 = 6（NCK 内部配置）
- 根据 S7 硬件配置：
 - 逻辑起始地址 = 444
 - 槽长度 = 10 个字节
 - 以 Little-Endian 格式表示

对应的机床数据配置如下：

```
MD10510 $MN_DPIO_LOGIC_ADDRESS_OUT[6] = 444 （I/O 区域的逻辑起始地址）

MD10511 $MN_DPIO_RANGE_LENGTH_OUT[6] = 0
（使用一个单独的有效数据插槽）

MD10512 $MN_DPIO_RANGE_ATTRIBUTE_OUT[6]
位 0 = 0 （Little-Endian 格式）
位 1 = 1 （仅通过 CC-Binding 写入）
位 3 = 1 （抑制槽生命符号报警）
```

编程

程序代码	注释
\$A_DPB_OUT[5,6]=128	; 从区域 5 偏移 6 开始写入 8 位
\$A_DPW_OUT[5,5]='B0110'	; 从区域 5 偏移 5 开始写入 16 位
	; Little-Endian 格式
	; 注意：区域数据字节 6 被改写
\$A_DPSD_OUT[5,3]='8FHex'	; 从区域 5 偏移 3 开始写入 32 位
	; Little-Endian 格式
	; 注意：区域数据字节 5、6 被改写
\$AC_MARKER[0]=5	
\$AC_MARKER[1]=3	
\$A_DPSD_OUT[\$AC_MARKER[0],\$AC_MARKER[1]]='8FHex'	
	; 从区域 5 偏移 3 开始写入 32 位
	; Little-Endian 格式
	; 间接编程
R1=\$A_DPB_OUT[5,6]	; 从区域 5 偏移 6 开始读 8 位
	; Little-Endian 格式
	; 结果："0xFF"
ID=1 WHENEVER TRUE DO \$A_DPB_OUT[5,0]=123	

程序代码	注释
	; 循环输出（每个插补周期）
\$A_DPB_OUT[5.255]=128	; 区域偏移无效，发出报警 17030。
\$A_DPB_OUT[6.10]=128	; 该区域已由零件程序编程，发出报警 17020
	; 无法写入。
\$A_DPB_OUT[7.10]=128	; 未定义该区域，发出报警 17020。
\$A_DPB_OUT[16.6]=128	; 区域下标 >= 最大可用区域数量，发出报警 17020。
	; 区域数量。

1.3.5.2 读访问中的 PROFIBUS I/O

前提条件

已进行 S7 硬件配置。

通过零件程序/同步动作访问的配置

- 区域下标 = 0（NCK 内部配置）
- 根据 S7 硬件配置：
 - 逻辑起始地址 = 456
 - 槽长度 = 32 个字节
 - 以 Big-Endian 格式表示

对应的机床数据配置如下：

MD10500 \$MN_DPIO_LOGIC_ADDRESS_IN[0] = 456（I/O 区域的逻辑起始地址）

MD10501 \$MN_DPIO_RANGE_LENGTH_IN[0] = 32（I/O 区域的长度）

MD10502 \$MN_DPIO_RANGE_ATTRIBUTE_IN[0]

位 0 = 1（Big-Endian 格式）

位 2 = 0（可通过系统变量和 CC-Binding 读取）

位 3 = 0（输出槽生命符号报警）

通过编译循环访问的配置

- 区域下标 = 1（NCK 内部配置）
- 根据 S7 硬件配置：
 - 逻辑起始地址 = 312
 - 槽长度 = 32 个字节
 - 以 Little-Endian 格式表示

1.3 通过 PROFIBUS 访问

对应的机床数据配置如下：

MD10500 \$MN_DPIO_LOGIC_ADDRESS_IN[1] = 312（I/O 区域的逻辑起始地址）

MD10501 \$MN_DPIO_RANGE_LENGTH_IN[1] = 32（I/O 区域的长度）

MD10502 \$MN_DPIO_RANGE_ATTRIBUTE_IN[1]

位 0 = 1（Big-Endian 格式）

位 2 = 1（只可通过 CC-Binding 读取）

位 3 = 1（抑制槽生命符号报警）

编程

程序代码	注释
\$AC_MARKER[0]=\$A_DPW_IN[0,0]	; 从区域 0 偏移 0 开始读 16 位
	; Big-Endian 格式
\$AC_MARKER[1]=\$A_DPSD_IN[0,1]	; 从区域 0 偏移 1 开始读 32 位
	; Big-Endian 格式
\$AC_MARKER[1]=\$A_DPSD_IN[0,8]	; 从区域 0 偏移 8 开始读 32 位
	; Big-Endian 格式
\$AC_MARKER[2]=0	
\$AC_MARKER[3]=8	
\$AC_MARKER[1]=\$A_DPSD_IN[\$AC_MARKER[2],\$AC_MARKER[3]]	
	; 从区域 0 偏移 8 开始读 32 位
	; Big-Endian 格式
	; 间接编程
ID=2 WHEN \$A_DPB_IN[0,11]>=5 DO \$AC_MARKER[2]='ABCDHex'	
	; 循环读取（每个插补周期）
R1=\$A_DPB_IN[0,255]	; 区域偏移无效，发出报警 17030。
R1=\$A_DPB_IN[2.6]	; 未定义该区域，发出报警 17020。
R1=\$A_DPB_IN[1.10]	; 该区域已由零件程序编程，发出报警 17020
	; 无法写入。
R1=\$A_DPB_IN[16.6]	; 区域下标 >= 最大可用区域数量，发出报警 17020。
	; 区域数量。

1.3.5.3 “写访问中 PROFIBUS I/O” 中区域下标的查询

前提条件

已进行 S7 硬件配置。

通过零件程序/同步动作访问的配置

- 区域下标 = 5 (NCK 内部配置)
- 根据 S7 硬件配置：
 - 逻辑起始地址 = 1200
 - 槽长度 = 32 个字节
 - 以 Big-Endian 格式表示

对应的机床数据配置如下：

MD10510 \$MN_DPIO_LOGIC_ADDRESS_OUT[5] = 1200 (I/O 区域的逻辑起始地址)

MD10511 \$MN_DPIO_RANGE_LENGTH_OUT[5] = 0

(使用一个单独的有效数据插槽)

MD10512 \$MN_DPIO_RANGE_ATTRIBUTE_OUT[5]

位 0 = 1 (Big-Endian 格式)

位 1 = 0 (只通过系统变量写入)

位 3 = 0 (输出槽生命符号报警)

编程

访问前查询区域 5 的状态		
N3	check:	; 跳转标记
N5	IF \$A_DP_OUT_STATE[5]==2 GOTOF write	; 数据区域有效时
		; => 跳转至 N15
N10	GOTOB check	; 返回 check
N15	write:	; 跳转标记
N20	\$A_DPB_OUT[5,6]=128	; 写入数据字节

查询配置的区域/槽是否都有效		
N3	check:	; 跳转标记
N5	IF \$A_DP_OUT_CONF==\$A_DP_OUT_VALID GOTOF write	; 数据区域有效时
		; => 跳转至 N15
N10	SETAL(61000)	; 设置报警号 61000
N15	write:	; 跳转标记
N20	\$A_DPB_OUT[5,6]=128	; 写入数据字节

查询配置的区域 5 是否有效		
N3	check:	; 跳转标记
N5	IF \$A_DP_OUT_VALID B_AND 'B100000' GOTOF write	; 数据区域有效时
		; => 跳转至 N15
N10	SETAL(61000)	; 设置报警号 61000
N15	write:	; 跳转标记

1.4 数据表

查询配置的区域 5 是否有效		
N20	\$A_DPB_OUT[5,6]=128	; 写入数据字节
查询有效 I/O 区域 5 配置的长度		
N100	R1=\$A_DP_OUT_LENGTH[5]	; I/O 区域（槽）的长度，以字节为单位
		; 结果: R1 = 32

1.4 数据表

1.4.1 机床数据

1.4.1.1 通用机床数据

编号	名称: \$MN_	描述
10300	FASTIO_ANA_NUM_INPUTS	有效 NCK 模拟量输入的数量
10310	FASTIO_ANA_NUM_OUTPUTS	有效 NCK 模拟量输出的数量
10320	FASTIO_ANA_INPUT_WEIGHT	NCK 模拟量输入的权重系数
10330	FASTIO_ANA_OUTPUT_WEIGHT	NCK 模拟量输出的权重系数
10350	FASTIO_DIG_NUM_INPUTS	有效 NCK 数字量输入字节的数量
10360	FASTIO_DIG_NUM_OUTPUTS	有效 NCK 数字量输出字节的数量
10362	HW_ASSIGN_ANA_FASTIN	外部 NCK 模拟量输入的硬件指定
10364	HW_ASSIGN_ANA_FASTOUT	外部 NCK 模拟量输出的硬件指定
10366	HW_ASSIGN_DIG_FASTIN	外部 NCK 数字量输入的硬件指定
10368	HW_ASSIGN_DIG_FASTOUT	外部 NCK 数字量输出的硬件指定
10394	PLCIO_NUM_BYTES_IN	可直接读取的 PLC I/O 输入字节的数量
10395	PLCIO_LOGIC_ADDRESS_IN	可直接读取的 PLC I/O 输入字节的起始地址
10396	PLCIO_NUM_BYTES_OUT	可直接写入的 PLC I/O 输出字节的数量
10397	PLCIO_LOGIC_ADDRESS_OUT	可直接写入的 PLC I/O 输出字节的起始地址
10398	PLCIO_IN_UPDATE_TIME	PLCIO 输入循环的更新时间
10399	PLCIO_TYPE_REPRESENTATION	PLC IO 数据的表示方式: Little-Endian 还是 Big-Endian
10500	DPIO_LOGIC_ADDRESS_IN	PROFIBUS I/O 区域的逻辑起始地址
10501	DPIO_RANGE_LENGTH_IN	PROFIBUS I/O 区域的长度

编号	名称: \$MN_	描述
10502	DPIO_RANGE_ATTRIBUTE_IN	PROFIBUS I/O 的属性
10510	DPIO_LOGIC_ADDRESS_OUT	PROFIBUS I/O 区域的逻辑起始地址
10511	DPIO_RANGE_LENGTH_OUT	PROFIBUS I/O 区域的长度
10512	DPIO_RANGE_ATTRIBUTE_OUT	PROFIBUS I/O 的属性
10530	COMPAR_ASSIGN_ANA_INPUT_1	比较器字节 1 的硬件 NCK 模拟量输入指定
10531	COMPAR_ASSIGN_ANA_INPUT_2	比较器字节 2 的硬件 NCK 模拟量输入指定
10540	COMPAR_TYPE_1	比较器字节 1 的设置
10541	COMPAR_TYPE_2	比较器字节 2 的设置

1.4.1.2 通道专用机床数据

编号	名称: \$MC_	描述
21220	MULTFEED_ASSIGN_FASTIN	“在一个程序段内编写多个进给率值”功能的 NCK I/O 输入字节指定

1.4.2 设定数据

1.4.2.1 通用设定数据

编号	名称: \$SN_	描述
41600	COMPAR_THRESHOLD_1	比较器字节 1 的阈值
41601	COMPAR_THRESHOLD_2	比较器字节 2 的阈值

1.4 数据表

1.4.3 信号

1.4.3.1 发送至 NC 的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
禁用 NCK 数字量输入	DB10.DBB0/122/124/126/ 128	DB2800.DBB0/1000
通过 PLC 置位 NCK 数字量输入	DB10.DBB1/123/125/127/ 129	DB2800.DBB1/1001
禁用 NCK 数字量输出	DB10.DBB4/130/134/138/ 142	DB2800.DBB4/1008
NCK 数字量输出的改值位	DB10.DBB5/131/135/139/ 143	DB2800.DBB5/1009
PLC 给出的 NCK 数字量输出的设置值	DB10.DBB6/132/136/140/ 144	DB2800.DBB6/1010
NCK 数字量输出的写值位	DB10.DBB7/133/137/141/ 145	DB2800.DBB7/1011
禁用 NCK 模拟量输入	DB10.DBB146	-
NCK 模拟量输入的写值位	DB10.DBB147	-
PLC 给出的 NCK 数字量输入的设置值	DB10.DBB148-163	-
NCK 模拟量输出的改值位	DB10.DBB166	-
NCK 模拟量输出的写值位	DB10.DBB167	-
禁用 NCK 模拟量输出	DB10.DBB168	-
PLC 给出的 NCK 数字量输出的设置值	DB10.DBB170-185	-

1.4.3.2 从 NC 发出的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
NCK 数字量输入的实际值	DB10.DBB60/186-189	DB2900.DBB0/1000
NCK 数字量输出的设定值	DB10.DBB64/190-193	DB2900.DBB4/1004
NCK 模拟量输入的实际值	DB10.DBB194-209	-
NCK 模拟量输出的设定值	DB10.DBB210-225	-

B3: 分布式系统 - 仅适用于 840D sl

2.1 简要说明

2.1.1 多操作面板配备多 NCU (T:M:N)

单一的操作监控站有时无法胜任对复杂机械设备的操作和监控。因此，在 SINUMERIK 设备网络（以太网）中可通过一个 PCU 将多个操作监控站与多个数控系统（NCU）互联在一起，从而实现对整体设备分布式的灵活操作及监控。

下图为当前可用于在设备网络中组成 T:1:N 操作监控系统的组件一览。其中：

- T 表示薄型客户单元（Thin Client Unit, TCU）或 HT8 手持操作设备（和 PCU 连接在一起）
- 1 (M) 表示人机控制设备（Man Machine Control, MMC）：配备 SINUMERIK Operate 的 PCU 50.x
- N 表示数控单元（Numeric Control Unit, NCU）：NCU 7x0.3 PN

2.1 简要说明

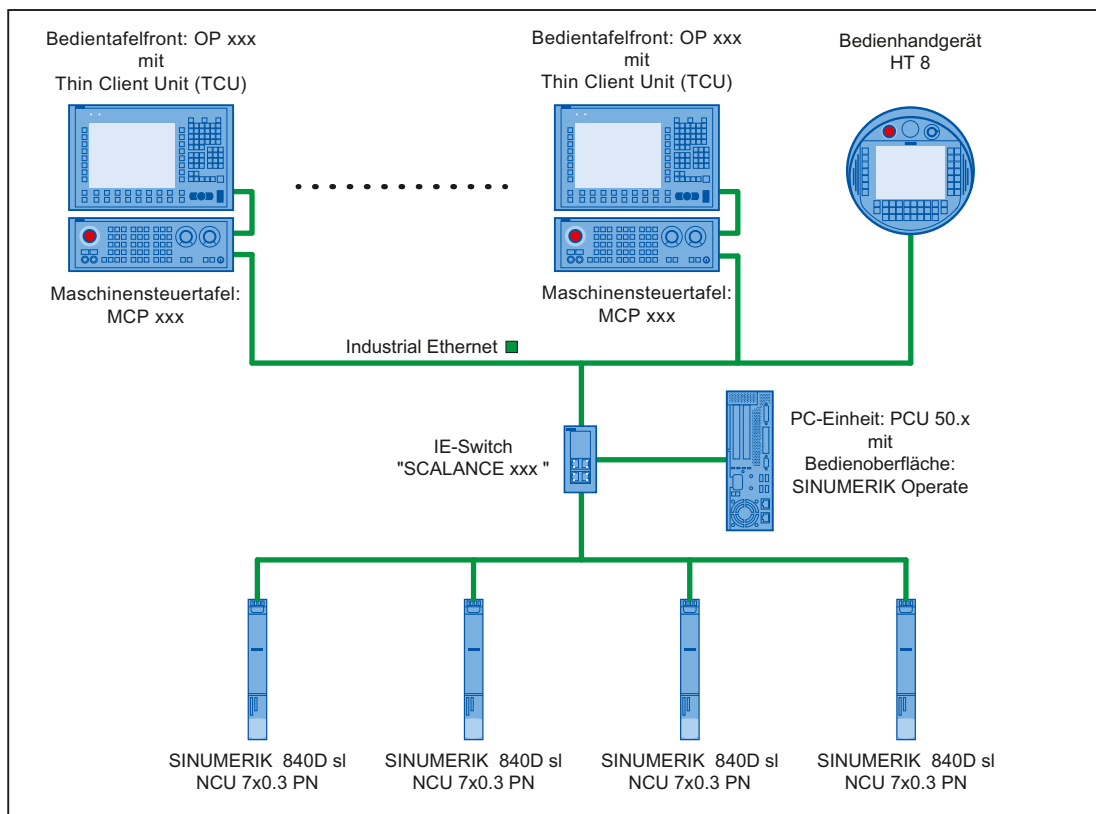


图 2-1 T:1:N 连接的示例

组态范围

T:1:N 操作监控系统须遵循以下组态范围：

T: 薄型客户单元 (TCU) 或 HT8 手持操作设备

TCU 负责将操作界面的图形信息从 PCU 50.x 传输至操作面板前端（OP）并在 OP 上显示这些图形信息。TCU 和操作面板前端构成一个操作站。

- 最大数量
在设备网络中理论上可运行任意数量的操作站，但实际上该数量受到可用地址空间的限制。
- 在线
每个 PCU 可支持同时最多四个操作站的在线。

- 操作焦点

在和 PCU 在线连接的多个操作站中，一个时间点上只能有一个操作站能获得操作焦点。只能通过此操作站进行与操作界面相关的输入操作。其他在线操作站的操作界面会变暗，表示其无操作焦点。不在线的操作站上只显示提示画面。操作焦点可在在线操作站之间来回切换。

- HT8 手持操作设备

HT8 手持操作设备将一个 TCU、一个操作面板前端和一个机床控制面板融合在一台设备中。

M: 配备 INUMERIK Operate 的 PCU 50.x

在一台配备了 SINUMERIK Operate 操作界面系统的 PCU 50.x 上可连接多个操作站和多个 NCU 7x0.3 PN。通过对通道菜单进行相应配置，可在操作站上显示不同 NCU 的通道数据。

说明

当前每个 T:1:N 连接组中只可配置一个 PCU $\Rightarrow M = 1$

N: SINUMERIK 840D sl, NCU 7x0.3 PN

- 内部 SINUMERIK Operate 操作界面系统

由于设备网络中 PCU 50.x 上配备有外部 SINUMERIK Operate 操作界面系统，因此必须将 NCU 集成的内部 SINUMERIK Operate 操作界面系统关闭。唯一的例外情况是该内部 SINUMERIK Operate 操作界面系统只用于刀具管理，此时可不关闭该系统。详细信息请见调试手册“操作单元和联网 (IM5)”，“网络配置”章节 > “应用示例”

- 机床控制面板 (MCP)

为了在 MCP 之间来回灵活切换，一个 NCU 上只可连接一个 MCP。MCP 的切换由用户通过 PLC 用户程序 (功能块 FB 9) 进行。

安装和连接

文档

- TCU, MCP, PCU:

设备手册“SINUMERIK 840D sl 操作单元和联网”

- NCU:

设备手册“SINUMERIK 840D sl NCU 7x0.3 PN”

2.1 简要说明

- 机床控制面板 (MCP)
FB 9: M:N 操作单元切换
- SCALANCE 交换机:
系统手册 “与 SIMATIC 的通讯”

配置、调试和参数设置

文档

- 设备网络的构建和 TCU 调试:
调试手册 “操作单元和联网 (IM5)”
- 通道菜单的配置
调试手册 “SINUMERIK Operate (IM9)”, “通道菜单” 章节

编程

文档

- MCP:
功能手册之基本功能分册, 章节“P3: 用于 SINUMERIK 840D sl 的 PLC 基本程序” > “模块说明” > “FB 9: M:N 操作单元切换”

2.1.2 NCU-Link

2.1.2.1 Link 通讯

NCU-Link 通讯可实现与插补周期同步的、跨 NCU 的数据交换, 用于以下任务:

- Link 变量 \$A_DLx 进行跨 NCU 的数据交换
- Link 轴和 Container-Link 轴跨 NCU 运行进给轴和主轴
- Lead-Link 轴跨 NCU 对进给轴和主轴的主值进行耦合

Safety Integrated

若 Safety Integrated 功能对 Link 轴生效, 则本地安全轴 (NCUx 上的 Link 轴) 和待运行通道 (NCUy) 之间的 NCU-Link 交换的是和安全功能不相关的数据。因此 Link 轴也可以

实现 **Safety Integrated** 支持的功能，例如：取决于速度/转速的设定值限制以及更高级的停止功能 (STOP D/E)。

说明

目前尚不支持 **Safety Integrated** 功能“安全软件限位开关”（SE）和“安全软件挡块，安全挡块信号”（SN）。

Link 模块

Link 通讯需要使用 Link 模块选件。Link 模块是一个 PROFINET 模块选件，支持以太网的等时同步实时通讯（IRT）。Link 模块需占用 NCU 的选件槽。

组态范围

缺省设置下，最多可将 3 个 NCU 组成一个 Link 组。

说明

若项目要求将更多 NCU 加入 Link 组，请咨询当地的西门子办事处。

2.1.2.2 Link 变量

Link 变量为系统全局用户变量。在配置了 Link 通讯后，Link 组中的所有 NCU 都可以使用这些变量。Link 变量可以在零件程序、循环和同步动作中读/写。

Link 变量的读/写在主处理中进行。因此 Link 变量会实现隐性的通道同步，使所有通道在同一个插补周期内可读到的相同的值。全局用户变量（GUD）也可像 Link 变量一样跨通道使用。不过 GUD 是在预处理中处理的，不会产生隐性的通道同步。因此，用户须在必要的情形下通过编程对 GUD 进行同步。

说明

在无 NCU-Link 的设备上，除了全局用户变量（GUD）外，Link 变量也可用作附加的全局用户变量。

2.1.2.3 Link 轴

Link 轴用于构建分布式控制系统，此时机床轴的设定值不是由该轴物理连接的 NCU 通道生成，而是由 Link 组中任何一个别的 NCU 的通道生成。

2.1 简要说明

Link 轴上机床轴和 NCU 的固定关系

在 Link 轴应用中，在哪个 NCU 上生成哪个机床轴的设定值是通过机床数据固定设置的。

Container-Link 轴上机床轴和 NCU 的变化关系

在 Container-Link 轴应用中，NCU 与机床轴的对应关系不是固定的。可选择通过 Link 组中的任意一个 NCU 来运行 Container-Link 轴。参见章节“轴容器 (页 109)”。

无论 NCU 和机床轴的关系是否固定，在生成设定值的 NCU 上都可将 Link 轴对应的逻辑机床轴信息通知给多个通道，从而像本地逻辑机床轴一样处理 Link 轴，例如用于位置/进给率耦合的主值、NCU 本地的跨通道取轴。

2.1.2.4 Lead-Link 轴

在与通用耦合功能组合使用时，若所有插补器在同一个 NCU 上，即生成设定值的通道和处理设定值的通道、主动轴和从动轴、主轴和副主轴在同一个 NCU 上，则无需使用 Lead-Link 轴。此时主动轴、从动轴、主轴或副主轴对应的机床轴也可作为 Link 轴连接至其他 NCU。

应用场合

若一个耦合组的所有插补器不位于同一个 NCU 上，则总是需要使用 Lead-Link 轴。通过 Lead-Link 轴可向另一个 NCU 提供主动轴/主轴的映射或本地插补器的主值。该 NCU 上的插补器将基于此主值生成从动轴/副主轴的设定值。

其它应用

当参与耦合的机床轴数多于 NCU 上仍可用于耦合的绝对或相对轴数时，也可使用 Lead-Link 轴。

文档

功能手册之特殊功能分册；章节“轴耦合(M3)”>“通用耦合”

2.1.2.5 关联性

为了插补运行多根轴，须确保将由正在运行的通道内插补器生成的设定值在相同的时间点传输至位置环。

和设定值传输给本地轴相比，在通过 NCU-Link 传输设定值时（例如：传送到 Link 轴）会产生一个插补周期的延时。必须相应增大插补器和位置控制器之间的缓存，以对此延时进行补偿：

MD18720 \$MN_MM_SERVO_FIFO_SIZE = <值> = <缺省值 + 补偿值>

功能	补偿值
Link 轴	如果 NCU 上有和 Link 轴或 Container-Link 轴进行插补的轴，在此类 NCU 上都必须补偿一个插补周期。 补偿值 = 1
Container-Link 轴	
Lead-Link 轴	在生成 Lead-Link 轴设定值的 NCU 上，必须补偿两个插补周期。 补偿值 = 2

说明

与 Lead-Link 轴不同，Link 轴和 Container-Link 轴还需要设置另一个补偿值（MD18720 \$MN_MM_SERVO_FIFO_SIZE）。在一个 Link 组中同时使用 Link 轴 或 Container-Link 轴和 Lead-Link 轴时，可不对延时进行补偿。

2.1.2.6 应用示例： 回转加工中心

下面以配备两个 NCU 的回转工作台为例介绍“NCU-Link” 功能的使用。此回转工作台的重要单元包括：

- 回转轴 MTR
- 主轴 MS1 - MS4
- 加工站 1 和 2，分别配备线性轴 X1 和 Z1
- 装料站和卸料站

所有机床轴均被指定给固定的 NCU。在各 NCU 的加工程序中都使用相同的进给轴名（X 和 Z）和主轴名（S1）。

每完成一个加工步骤，示意图便显示工件的状态。

执行每个加工步骤时，回转台（MTR）都会逆时针旋转一个位置。这样一来，在每个加工步骤中主轴都会分到一个新站。通道中定义的主轴与机床轴之间不断变化的对应关系在 NCU 中通过一个轴容器映射。

若主轴对应的机床轴不位于本地 NCU 上，主轴设定值会通过 Link 通讯传输至相应的 NCU，并通过该 NCU 输出至机床轴。

2.1 简要说明

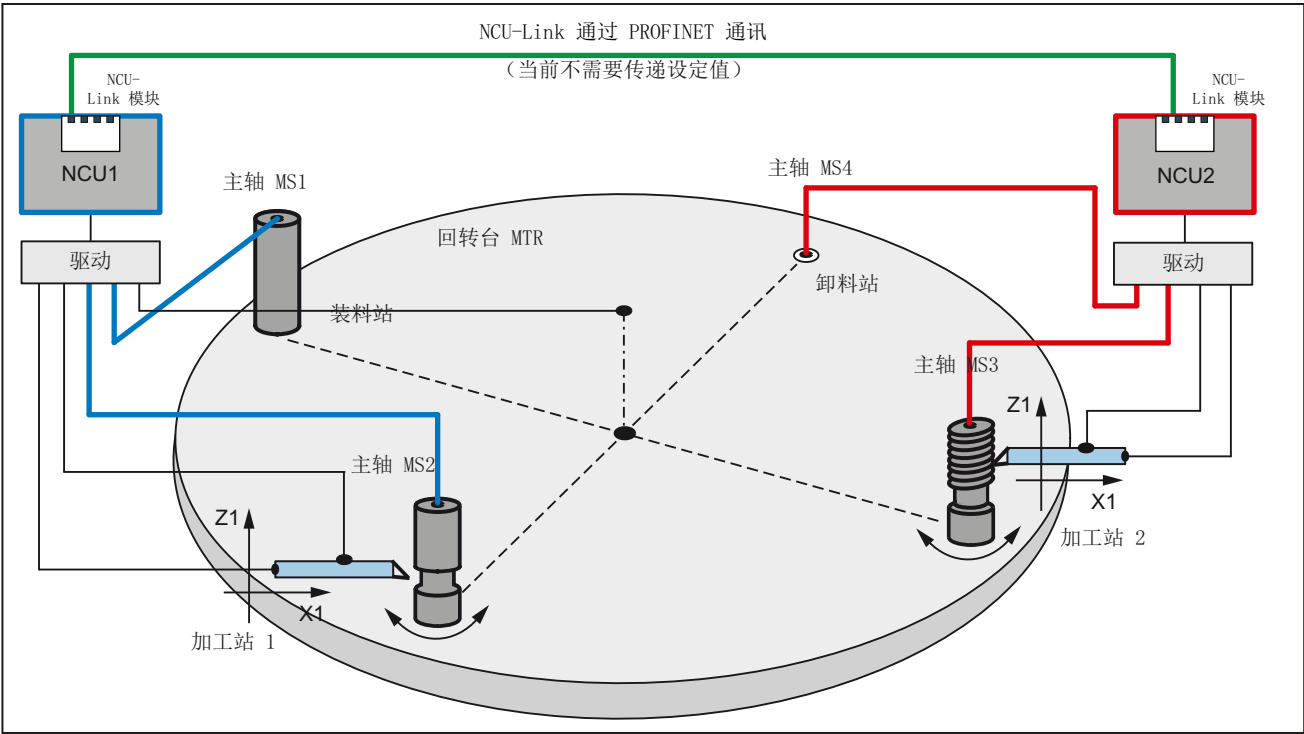


图 2-2 图 1: 初始状态

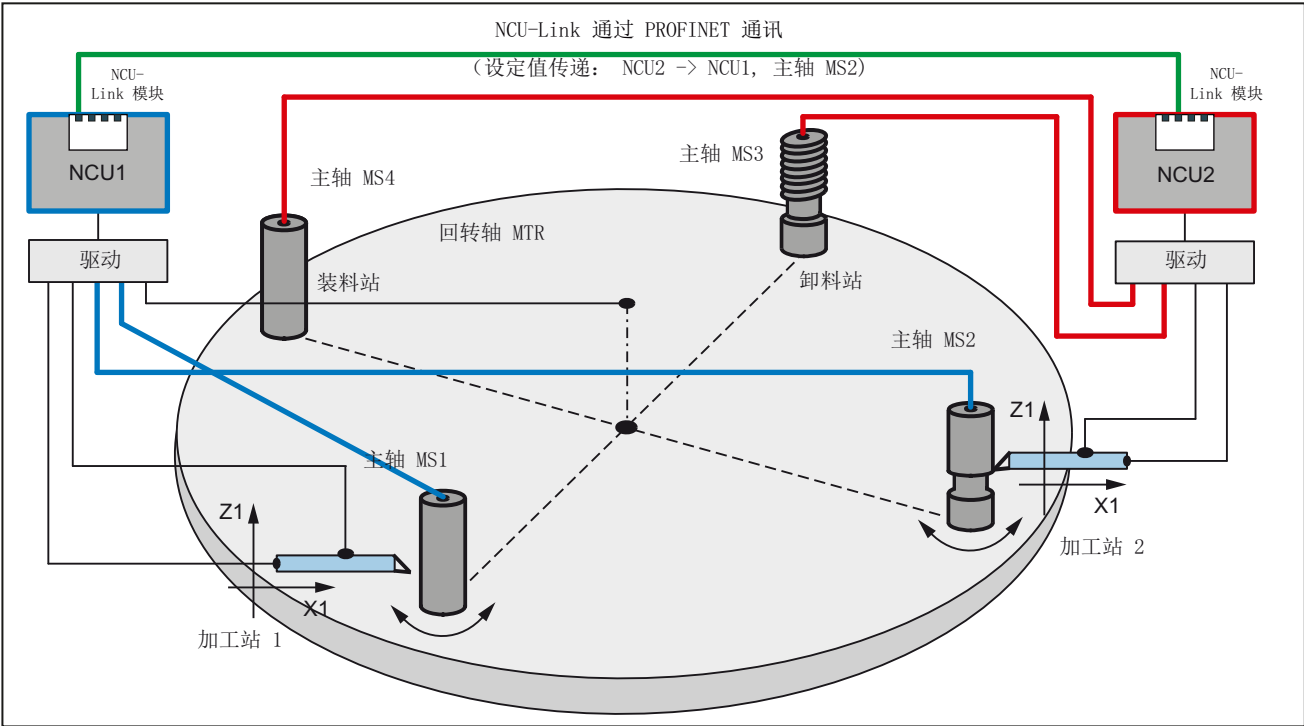


图 2-3 图 2: 旋转一个“位置”后

参数设置 (图表)

简介		
两个 NCU 的零件程序中均编写的通道轴: X, Z, S1		
NCU1	NCU 中定义的机床轴:	
	本地:	X1, Z1
	轴容器:	MS1, MS2
NCU 2	NCU 中定义的机床轴:	
	本地:	X1, Z1
	轴容器:	MS3, MS4

	初始位置 (图 1)	回转轴 MTR (回转台) 旋转一个位置 (图 2)
NCU1	加工站 1: X1, Z1, MS2	
	零件程序中编写的通道轴的映射:	
	X → X1、Z → Z1	X → X1、Z → Z1
	S1 → MS2	S1 → MS1
NCU 2	加工站 2: X1, Z1, MS2	
	零件程序中编写的通道轴的映射:	
	X → X1、Z → Z1	X → X1、Z → Z1
	S1 → MS3	S1 → MS2 (Link 轴)

2.2 NCU-Link

2.2.1 Link 通讯

2.2.1.1 简介

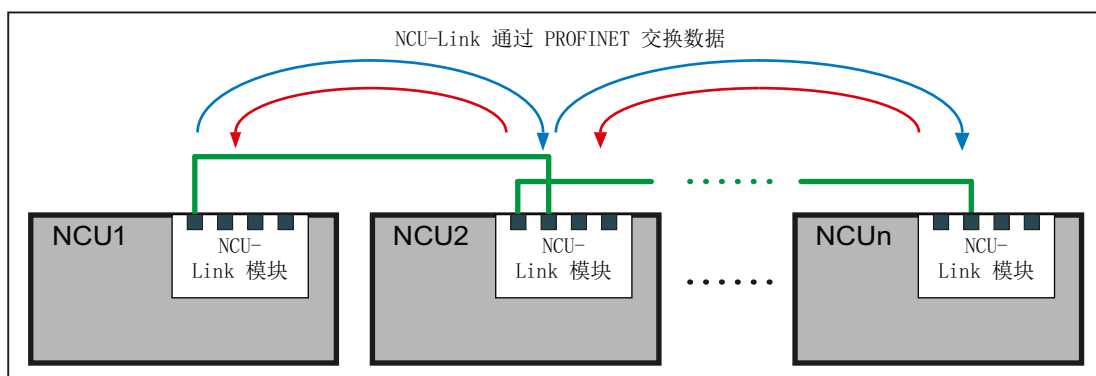


图 2-4 Link 通讯（原理）

NCU-Link 通讯可实现与插补周期同步的、跨 NCU 的数据交换，用于以下任务：

- 跨 NCU 的 Link 变量 $\$A\_DLx$
所有参与 NCU-Link 通讯的 NCU 都可以访问 Link 变量，因为这些变量是通过 NCK Link 与插补周期同步地在 Link 组中的各个 NCU 之间交换的。
- Link 轴和 Container-Link 轴跨 NCU 运行进给轴和主轴
若需在 NCU 的一个通道内运行单轴或插补运行多轴，通过 NCU-Link 也可将设定值传送给 Link 组中其他 NCU 上物理连接的轴。这类轴称为 Link 轴。
- Lead-Link 轴跨 NCU 对进给轴和主轴的主值进行耦合
NCU1 运行轴 X1（主动轴），它的设定值通过 NCU-Link 传输至 NCU2 的 Link 轴（Lead-Link 轴）。NCU2 的轴 X2 以该 Lead-Link 轴为耦合对象。这样一来，轴 X2 便间接成为了 X1 的从动轴。

数据传输

取决于生效的功能，参与 Link 通讯的 NCU 之间会进行下列循环和非循环数据传输：

- 高优先级的循环数据传输：
 - Link 轴的设定值、实际值和状态信号
 - NCU 状态信号
- 高优先级、非循环、不可排挤的数据传输：
 - Safety Integrated 功能范畴内的非安全相关数据
 - 轴容器旋转时容器轴的状态
- 低优先级、非循环、可推迟的数据传输（按优先级从高到低的顺序排列）：
 - Link 变量
 - 热启动请求
 - 轴容器旋转激活
 - NCU 全局机床数据和设定数据的修改
 - Link 轴机床数据的激活
 - 报警

排挤

在进行低优先级、非循环、可排挤的数据传输时，若在一个插补周期内无法发送所有请求，那么高优先级的请求将挤掉低优先级的请求。被排挤的请求会于之后发送。

组态范围：NCU 7x0.3 PN

缺省设置下，最多可将 3 个 NCU 组成一个 Link 组。

说明

如项目要求使用 3 个以上的 NCU，请咨询当地的西门子办事处。若非项目要求的增配，NCU 超过 3 个时 Link 会被系统拒绝，并发出报警 380020。

NCU-Link 和 Safety Integrated

下图显示了包含两个 NCU 和两个机床轴的结构，其中 NCU2 的机床轴 MA2 作为 Link 轴由 NCU1 运行。两根轴均通过 Safety Integrated 功能进行安全监控。

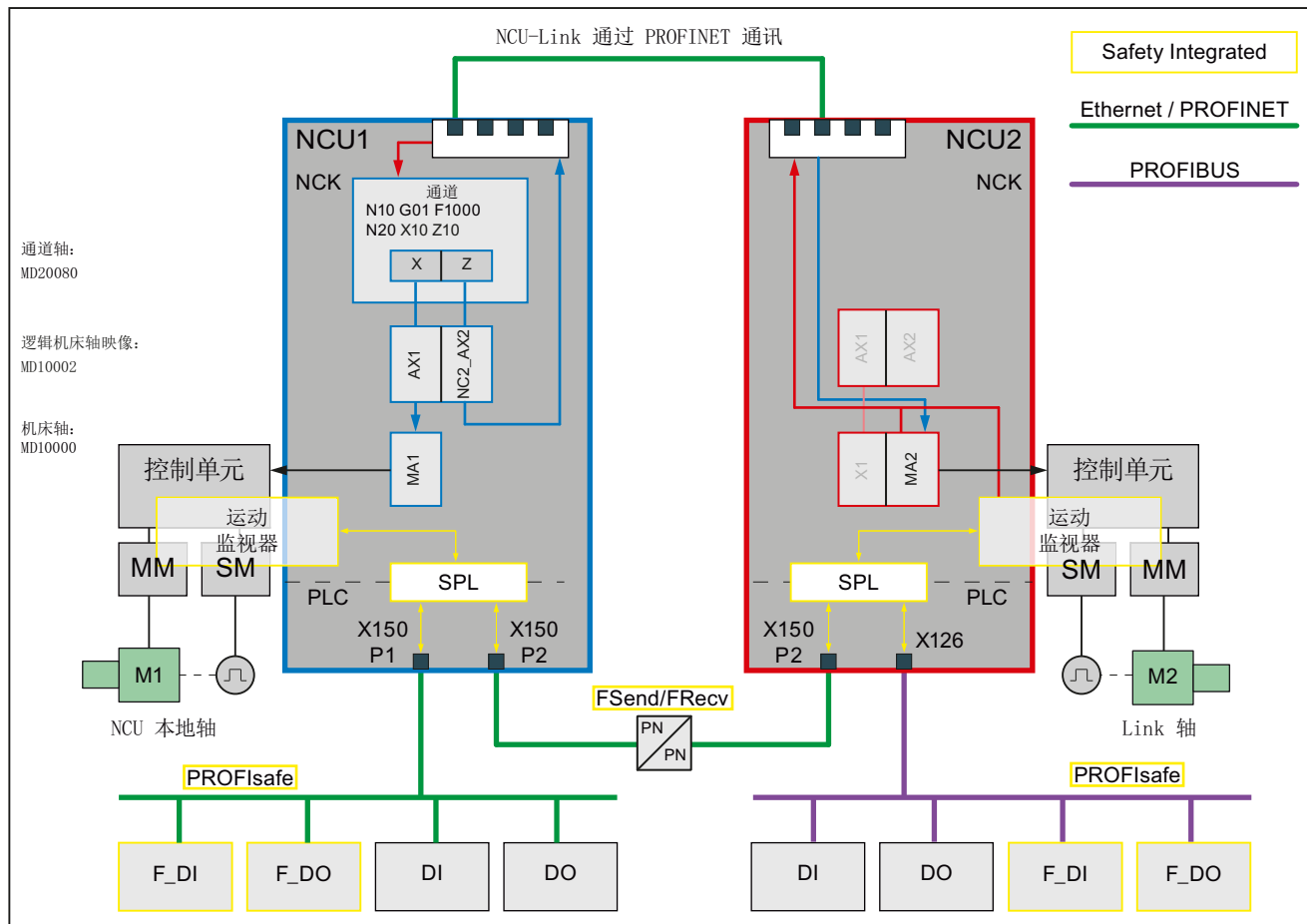


图 2-5 NCU-Link 和 Safety Integrated

故障安全输入信号 (F_DI) 可由两个 NCU 采集, 通过安全可编程逻辑 (SPL) 进行互联, 并通过安全 CPU-CPU 通讯 (FSend/FRecv) 交换。

在 Safety Integrated 功能范畴内，NCU-Link 通讯实现的是对 Safety Integrated 所支持功能的单通道式、和安全不相关的 Link 轴数据的传输。

示例：

- 在执行隐性的 **Stop D** 时，通道中所有正在运行的轴（含 **Link** 轴）以插补方式制动。
- 切换安全速度（**SG**）时，通道中所有正在运行的轴（含 **Link** 轴）减速。

- 切换安全速度（SG）时，通道中所有正在运行的轴（含 Link 轴）减速。

Safety Integrated 验收测试和 NCU-Link

原则上需通过验收测试向导（ATW, Acceptance Test Wizzard）为每个 NCU 单独执行验收测试。但是 Link 轴的报警只能显示在物理连接的本地 NCU 上。为了使 ATW 也能检查 Link 轴的报警，必须将通过 NCU-Link 连接的安全 NCU 的信息提供给 ATW。为此，必须将该 NCU 的信息输入到 PG/PCU 上用于运行 ATW 的 NETNAMES.INI 文件中。

示例：NETNAMES.INI 文件 的相关段落，指出 NCU-Link 中包含的两个安全 NCU 的信息

文件：NETNAMES.INI（段落）

```
[own]
owner= HMI_1
[conn HMI_1]
conn_1= NCU_1
conn_2= NCU_2
[param HMI_1]
mmc_address= 1
[param network]
[param NCU_1]
nck_address= 192.168.214.2,LINE=10,NAME=/NC,SAP=030d,PROFILE=CLT1__CP_L4_INT
plc_address= 192.168.214.2,LINE=10,NAME=/PLC,SAP=0201,PROFILE=CLT1__CP_L4_INT
ADDRESS2=    192.168.214.2,LINE=10,NAME=/CP,SAP=0501,PROFILE=CLT2__CP_L4_INT
ADDRESS10=   192.168.214.2,LINE=10,NAME=/
DRIVE_00_000,SAP=0201,SUBNET=0000-00000000:000,PROFILE=CLT2__CP_L4_INT
ADDRESS11=   192.168.214.2,LINE=10,NAME=/DRIVE_03_003,SAP=0900,PROFILE=CLT2__CP_L4_INT
ADDRESS12=   192.168.214.2,LINE=10,NAME=/DRIVE_03_011,SAP=0B00,PROFILE=CLT2__CP_L4_INT
ADDRESS13=   192.168.214.2,LINE=10,NAME=/DRIVE_03_012,SAP=0C00,PROFILE=CLT2__CP_L4_INT
ADDRESS14=   192.168.214.2,LINE=10,NAME=/DRIVE_03_013,SAP=0D00,PROFILE=CLT2__CP_L4_INT
ADDRESS15=   192.168.214.2,LINE=10,NAME=/DRIVE_03_014,SAP=0E00,PROFILE=CLT2__CP_L4_INT
ADDRESS16=   192.168.214.2,LINE=10,NAME=/DRIVE_03_015,SAP=0F00,PROFILE=CLT2__CP_L4_INT
name=Machine_1
[param NCU_2]
nck_address= 192.168.214.1,LINE=10,NAME=/NC,SAP=030d,PROFILE=CLT1__CP_L4_INT
plc_address= 192.168.214.1,LINE=10,NAME=/PLC,SAP=0201,PROFILE=CLT1__CP_L4_INT
ADDRESS2=    192.168.214.1,LINE=10,NAME=/CP,SAP=0501,PROFILE=CLT2__CP_L4_INT
ADDRESS10=   192.168.214.1,LINE=10,NAME=/
DRIVE_00_000,SAP=0201,SUBNET=0000-00000000:000,PROFILE=CLT2__CP_L4_INT
ADDRESS11=   192.168.214.1,LINE=10,NAME=/DRIVE_03_003,SAP=0900,PROFILE=CLT2__CP_L4_INT
ADDRESS12=   192.168.214.1,LINE=10,NAME=/DRIVE_03_011,SAP=0B00,PROFILE=CLT2__CP_L4_INT
ADDRESS13=   192.168.214.1,LINE=10,NAME=/DRIVE_03_012,SAP=0C00,PROFILE=CLT2__CP_L4_INT
ADDRESS14=   192.168.214.1,LINE=10,NAME=/DRIVE_03_013,SAP=0D00,PROFILE=CLT2__CP_L4_INT
ADDRESS15=   192.168.214.1,LINE=10,NAME=/DRIVE_03_014,SAP=0E00,PROFILE=CLT2__CP_L4_INT
ADDRESS16=   192.168.214.1,LINE=10,NAME=/DRIVE_03_015,SAP=0F00,PROFILE=CLT2__CP_L4_INT
name=Machine_2
```

文档

对“Safety Integrated”的详细说明请见：

功能手册“Safety Integrated”

2.2 NCU-Link

2.2.1.2 Link 模块

NCU-Link 通讯通过 Link 模块进行。Link 模块是一个 PROFINET 模块选件，支持以太网的等时同步实时通讯（IRT）。Link 模块只可用于 NCU-Link 通讯。Link 模块不可用于常规的 PROFINET 通讯。

Link 模块需要占用 NCU 模块上的选件槽。

说明

而一个 NCU 模块上只有一个选件槽。因此 NCU-Link 模块无法与其他选件模块一同使用。

Link 模块和 NCU 模块

NCU 模块 NCU710.3 PN、NCU720.3 PN 和 NCU730.3 PN 的 Link 模块可以为以太网通讯板 CBE30-2。Link 模块需要占用 NCU 模块上的选件槽。

文档

设备手册“NCU7x0.3 PN”；“可连接组件”章节 > “CBE30-2”

2.2.1.3 参数设置：NC 系统周期

Link 通讯的基本前提条件是 NCU 组中的**所有** NCU 的下列系统周期都采用相同设置：

- 系统基本周期
- 位置控制器周期
- 插补周期

系统基本周期

系统基本周期采用 STEP7 项目中为等时通讯设置的 DP 周期。当前系统基本周期在以下机床数据中显示：

MD10050 \$MN_SYSCLOCK_CYCLE_TIME

注意
针对多通讯总线的手动调整 若在一个 NCU 上配置了多条通讯总线（PROFIBUS 1 ... n, PROFINET 1 ... m），必须在 STEP7 HW-Config 中为每条通讯总线设置相同的周期时间。 与位置控制器周期的关联性 在 SINUMERIK 840D sl 上，系统基本周期和位置控制器周期之间的比例固定（1:1），无法修改。由于在 NCU-Link 中只能设置特定的位置控制器周期，因此只允许将此位置控制器周期设置为系统基本周期或 DP 循环周期。参见下文“位置控制器周期”。

位置控制器周期

位置控制器周期和系统基本周期成比例设置。在 SINUMERIK 840D sl 上，该比例固定（1:1），无法修改。当前位置控制器周期在以下机床数据中显示：

MD10061 \$MN_POSCTRL_CYCLE_TIME

注意
允许采用的位置控制器周期 取决于 Link 组中 NCU 的数量，只可为 NCU-Link 设置下列位置控制器周期 [ms]： • 2 个 NCU：2.0、2.5、3.0、3.5 和 4.0 • 3 个 NCU：3.0、3.5 和 4.0 相关内容参见章节“配置 (页 95)”。

插补周期

插补周期和系统基本周期成比例设置。此周期通过以下机床数据设置：

MD10070 \$MN_IPO_SYSCLOCK_TIME_RATIO

当前插补周期在以下机床数据中显示：

MD10071 \$MN_IPO_CYCLE_TIME

设置说明

周期设置

建议采用以下设置：

2.2 NCU-Link

- 保留 90% NCK 计算时间负载率这一缺省设置：
MD10185 \$MN_NCK_PCOS_TIME_RATIO
- 设置系统周期时，须确保正常程序运行下插补器和位置控制器造成的**平均**系统负载率不超过 60%。**最大值**应不超过 90%。

提示

在**特例情况**下显示的最大值可能会超过 100%。

最低循环通讯负载率

在 Link 通讯中，插补周期和位置控制器周期比例为 1 : 1 时能够实现最低的循环通讯负载率。在驱动功能“动态伺服控制（DSC, Dynamic Servo Control）”激活时，该周期设置通常是最高效的设置。

缺点是在 Link 通讯中无法重复报文。

NC/PLC 接口的更新时间

Link 组中的所有 NCU 启动时下列设置生效：

MD18000 \$MN_VDI_UPDATE_IN_ONE_IPO_CYCLE = 1

这样便可在一个插补周期内实现对 NC/PLC 接口的完全读写。

2.2.1.4 参数设置：Link 通讯

NC 专用机床数据

编号	名称 \$MN_	含义
MD12510	NCU_LINKNO	Link 组内 NCU 的唯一数字标识。标识必须从 1 开始以升序给定，且必须连贯。 取值范围：1、2 到最大 NCU 编号 提示 NCU 标识为 1 的 NCU 是 Link 组中的 主 NCU 。Link 轴和轴容器的参数设置只允许通过主 NCU 的机床数据和设定数据进行。
MD18780	MM_NCU_LINK_MASK.位 0	激活 Link 通讯
MD18781	NCU_LINK_CONNECTIONS	内部 Link 连接的数量 提示 建议保留缺省值 0（由 NC 测定数量）。
MD18782	MM_LINK_NUM_OF_MODULES	通过 NCU-Link 相互连接的 NCU 的数量。

2.2.1.5 配置

NC 系统软件为每种支持的“NCU 数量+位置控制器周期”的组合提供专门的配置（参见章节“参数设置：NC 系统周期 (页 92)”）。

根据机床数据中设置的值，系统启动时会载入相应配置：

- MD18782 \$MN_MM_LINK_NUM_OF_MODULES（Link 组的 NCU 数量）
- MD10061 \$MN_POSCTRL_CYCLE_TIME（位置控制器周期）

说明

若无法使用软件提供的缺省配置，请咨询当地的西门子办事处。

2.2.1.6 NCU 之间的接线

Link 组内各个 NCU 的数字顺序在每个 NCU 内通过以下机床数据定义：

MD12510 \$MN_NCU_LINKNO = <NCU 编号>，NCU 编号 = 1 到最大 NCU 编号

接线

NCU-Link 模块须按照 NCU 编号升序接线，即从 NCU1 起开始接线，参见下图：NCU(n)，端口 0 → NCU(n+1)，端口 1

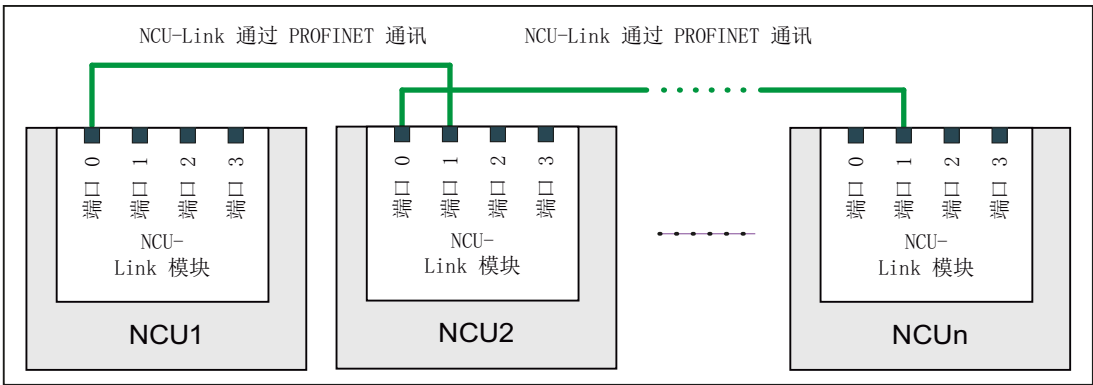


图 2-6 NCU-Link 的接线图

2.2.1.7 激活

Link 通讯通过以下机床数据激活：

2.2 NCU-Link

MD18780 \$MN_MM_NCU_LINK_MASK, 位 0 = 1

说明

激活时间点

建议在对参与 Link 通讯的所有 NCU 执行完整的功能调试后再激活 Link 通讯。

2.2.2 Link 变量

配有多台 NCU 的复杂设备需要在各个 NCU 之间循环交换用户数据，以便在整个系统范围内协调生产过程。这种数据交换是通过 Link 通讯和一个专用的存储区（即每个 NCU 都有的 LINK 变量存储区）实现的。

Link 变量存储器的大小和结构均可由用户定义。Link 变量存储器中保存的数据通过特殊的 Link 变量 \$A_DLx 访问。

Link 变量因此为系统全局用户变量。在配置了 Link 通讯后，Link 组中的所有 NCU 都可以在零件程序和循环中读/写这些变量。与全局用户数据（GUD）不同，Link 变量也可用在同步动作中使用。

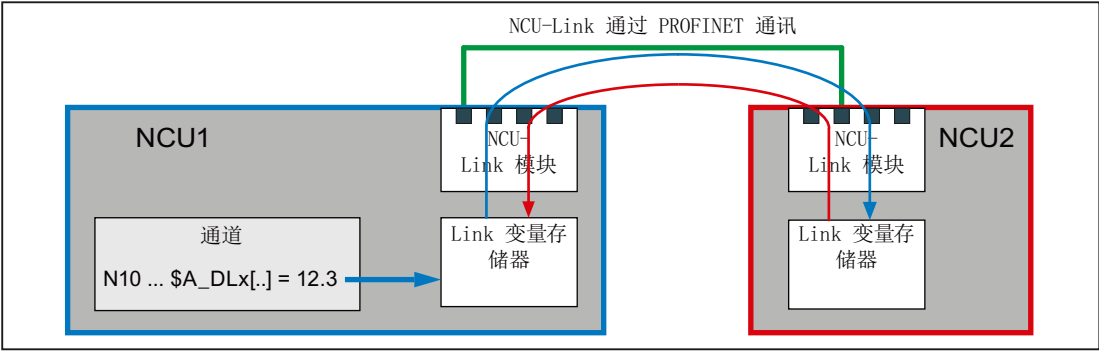


图 2-7 Link 变量

前提条件

使用 Link 轴的前提条件是按照“Link 通讯 (页 88)”章节的内容建立了 Link 通讯。

Link 变量用作 NCU 全局用户变量

说明
 若一个 NCU 不属于某个 Link 点，或者不需要将 Link 变量用于 NCU-Link 通讯，可将 Link 变量用作 NCU 全局用户变量。

2.2.2.1 Link 变量存储器的属性

存储器容量的设置

Link 变量存储器容量通过以下机床数据设置，以字节为单位：
 MD18700 \$MN_MM_SIZEOF_LINKVAR_DATA（Link 变量存储器容量）
 建议将 Link 组中所有 NCU 的 Link 变量存储器设为相同容量。各个存储器容量有所不同时，系统会采用设置的最大容量。

初始化

NCU 启动后，Link 变量存储器通过 0 初始化。

结构

从系统的角度来看，Link 变量存储器是一块没有特定结构的、用于 Link 通讯的存储区域。Link 变量存储器的结构只能由用户/机床制造商定义。根据该定义，对 Link 变量存储器的访问通过特定数据格式的 Link 变量进行。

系统范围内的调整

在对一个 Link 变量存储器执行写访问后，数据更改会传输至 Link 组中的所有其他 NCU 的 Link 变量存储器。正常情况下，Link 通讯会使 Link 变量存储器的更新延迟两个插补周期。

2.2.2.2 Link 变量的属性

对 Link 变量存储器的访问是通过下列特定数据格式的 Link 变量实现的：

数据类型 ¹⁾	名称	数据格式 ²⁾	字节 ²⁾	索引 i ³⁾	取值范围
UINT	\$A_DLB[i]	BYTE	1	i = n * 1	0 ... 255
INT	\$A_DLW[i]	WORD	2	i = n * 2	-32768 ... 32767
INT	\$A_DLD[i]	DWORD	4	i = n * 4	-2147483648 ... 2147483647

2.2 NCU-Link

数据类型 ¹⁾	名称	数据格式 ²⁾	字节 ²⁾	索引 i ³⁾	取值范围
实数	\$A_DLR[i]	实数	8	$i = n * 8$	$\pm(2,2*10^{-308} \dots 1,8*10^{+308})$
1) 零件程序/循环中 Link 变量的数据类型 2) Link 变量存储器中 Link 变量的数据类型或字节数量 3) 下标 i 须注意以下事项: - 下标 i 是一个字节下标, 指出 Link 变量存储器的开头。 - 选择下标时, 须确保 Link 变量存储器中访问的字节符合数据格式限制 $\Rightarrow$ 下标 $i = n * 8$ 字节, 其中 $n = 0, 1, 2, \dots$ - \$A_DLB[i]: $i = 0, 1, 2, \dots$ - \$A_DLW[i]: $i = 0, 2, 4, \dots$ - \$A_DLD[i]: $i = 0, 4, 8, \dots$ - \$A_DLR[i]: $i = 0, 8, 16, \dots$					

写入

写 Link 变量是和主处理同步的。

读取

读 Link 变量时会触发预处理停止。

检查

对 Link 变量和 Link 变量存储器系统会进行下列检查:

- 是否遵循了取值范围限制
- 访问是否遵循格式限制
- 是否未超出定义的 Link 变量存储器的容量

用户/机床制造商须自行避免以下错误:

- 采用错误的数据格式访问
- 访问错误的地址 (下标 i)
- 通过一个 NCU 的多个通道或不同的 NCU 并行改写相同的数据
- 在本地 NCU 或其他 NCU 的一个通道更新前就读数据

说明

数据一致性

用户/机床制造商须自行确保 Link 变量存储器内的数据一致性, 不论是本地 NCU 还是 Link 组中的其他 NCU。

2.2.2.3 写单元

在对 Link 变量存储器进行写访问时（例如 $\$A\_DLB[4] = 21$ ），需要一个所谓的“Link 变量写单元”用于系统内部的写任务管理。每个插补周期可用的最大写单元数量通过下列机床数据设置：

MD28160 $\$MC\_MM\_NUM\_LINKVAR\_ELEMENTS$

最大写单元数量因此可限制每个插补周期能够写入的 Link 变量的数量。

2.2.2.4 写访问的时间特性

Link 变量的写入是和主处理同步的。本地 NCU 的其他通道最迟可在下一个插补周期读取新值。而本地通道在下一个程序段中即可读取新值。

Link 组中的其他 NCU 将于两个插补周期后读取新值。但由于带宽受限，在将写任务传输至 Link 组中的其他 NCU 时可能会出现延迟（Message-Delay：信息延迟）。信息延迟的原因可能为：

- 在一个插补周期中写入大量 Link 变量
- 在同一个插补周期中写入 Link 变量和请求轴容器旋转
- 在同一个插补周期中写入 Link 变量和传输报警

2.2.2.5 系统变量

NC 专用系统变量

名称	含义
$\$AN\_LINK\_TRANS\_RATE\_LAST$	上一个插补周期中仍未传输的写任务的数量。
$\$AN\_LINK\_TRANS\_RATE\_LAST\_SUM[<n>]$	上一个插补周期中仍未向指定 NCU<n>（n 为 NCU 编号）传输的写任务的数量。
$\$AN\_LINK\_CONN\_SIZE\_LINKVAR$	在一个写任务中为一个 Link 变量传输的字节数量。
$\$AN\_LINK\_CONN\_SND[<n>]$	每个插补周期可从当前 NCU 传输至指定 NCU 的最大字节数
$\$AN\_LINK\_CONN\_RCV[<n>]$	每个插补周期可从指定 NCU 传输至当前 NCU 的最大字节数
<n>: <n>: NCU 编号，为各个 NCU 中 MD12510 $\$MN\_NCU\_LINKNO$ 的值	

2.2 NCU-Link

对 \$AN_LINK_CONN_SIZE_LINKVAR 的提示

写入一个 Link 变量后，便会通过 Link 通讯非循环传输由 \$AN_LINK_CONN_SIZE_LINKVAR 指定的字节数量。此数量与 Link 变量的格式无关。

每个插补周期可向指定 NCU 传输的最大写任务数量如下计算：

最大写任务数量 = \$AN_LINK_CONN_SND[<n>] / \$AN_LINK_CONN_SIZE_LINKVAR

通道专用系统变量

名称	含义
\$A_LINK_TRANS_RATE ¹⁾	当前插补周期中还可传输的写任务的数量。
1) 应用示例参见章节：“写任务的同步 (页 100)”	

2.2.2.6 写任务的同步

在特定应用中，如需将新写入的 Link 变量值精准地在两个插补周期内传输至 Link 组中的其他 NCU，则必须在一个同步动作中 Link 变量的写任务。 仅在当前插补周期中尚可执行写任务时，同步动作中才会写入 Link 变量。为此，系统变量 \$A_LINK_TRANS_RATE 中包含了当前插补周期中尚可执行的写任务数。

下面的示例中将写入一个数据类型为单字（WORD，2 个字节）的 Link 变量和一个数据类型为双字（DWORD，4 个字节）的 Link 变量：

程序示例

```
N120 WHEN $A_LINK_TRANS_RATE > 0 DO $A_DLW[0] = 9
N125 WHEN $A_LINK_TRANS_RATE > 0 DO $A_DLD[2] = 7
N130 G4 F1
```

仅在写任务能在同一个插补周期中传输至 Link 组的其他 NCU 时，系统才会执行 N120 中的同步动作。 此时系统变量 \$A_LINK_TRANS_RATE 也将 在同一个插补周期内减一，为后续程序段 N125 中的同步动作提供新值。

2.2.2.7 示例：Link 变量存储器的划分

创建下列数据用于 Link 通讯：

数据格式	数量	每个数据的字节	所需字节
BYTE	2	1	2
WORD	1	2	2

数据格式	数量	每个数据的字节	所需字节
DWORD	3	4	12
REAL	1	8	8
所需的 Link 变量存储器容量:			24

存储器结构

考虑到 Link 变量存储器中的数据格式限制，存储器结构如下：

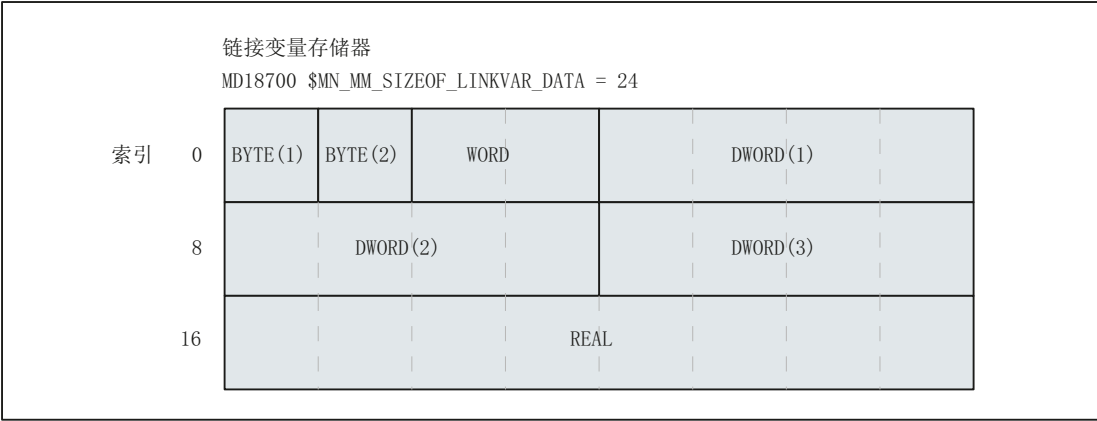


图 2-8 示例：Link 变量存储器结构

说明

存储器结构

理论上 Link 变量存储器可以为任意结构，考虑到数据格式限制也可以为其他结构。

必须根据所定义的存储器结构如下编写 Link 变量的访问：

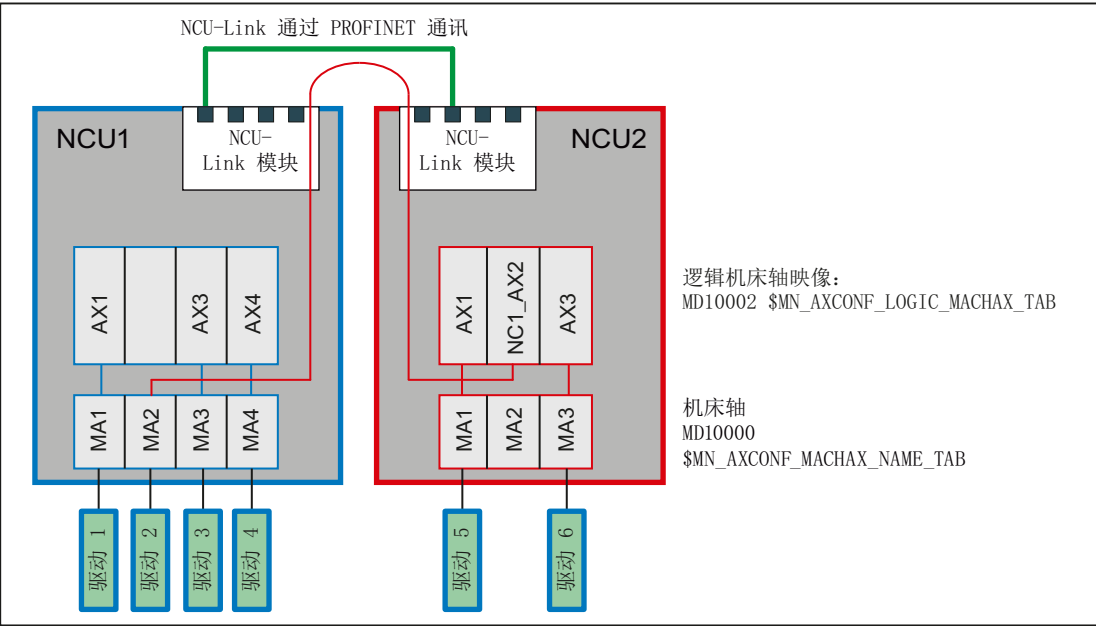
程序代码	描述
\$A_DLB[0]	; BYTE (1)
\$A_DLB[1]	; BYTE (2)
\$A_DLW[2]	; WORD
\$A_DLD[4]	; DWORD (1)
\$A_DLD[8]	; DWORD (2)
\$A_DLD[12]	; DWORD (3)
\$A_DLR[16]	; REAL

2.2 NCU-Link

2.2.2.8 示例：读取驱动数据

任务

设备中包含两个 NCU（NCU1 / NCU2）。NCU 之间通过 NCU-Link 相连。NCU1 的机床轴 MA2（驱动 2）作为 Link 轴由 NCU2 以插补方式运行。驱动 2 的电流实际值需由 NCU1 传输至 NCU2 用于分析。下图为基本的系统结构图。



前提条件

驱动 2（NCU1/MA2）的电流实际值可通过系统变量 \$VA_CURR 读取。若驱动基于 PROFIdrive，必须设置下列机床数据：

MD36730 \$MA_DRIVE_SIGNAL_TRACKING = 1（采集额外的驱动实际值）

通过置位机床数据提供以下驱动实际值：

- \$AA_LOAD, \$VA_LOAD（驱动负载率，单位 %）
- \$AA_POWER, \$VA_POWER（驱动有功功率，单位 W）
- \$AA_TORQUE, \$VA_TORQUE（驱动力矩设定值，单位 Nm）
- \$AA_CURR, \$VA_CURR（进给轴或主轴的电流实际值，单位 A）

编程

NCU1

编写一个静态同步动作，将驱动 2（NCU1/MA2）的电流实际值 \$VA_CURR 通过 Link 变量 \$A_DLR[0]（实数值）按插补周期循环写入 Link 变量存储器的前 8 个字节。

程序代码
IDS=1 WHENEVER TRUE DO \$A_DLR[0]=\$VA_CURR[MA2]

NCU2

编写一个静态同步动作，通过 Link 变量 \$A_DLR[0] 按插补周期循环读取传输过来的电流实际值。如果电流实际值大于 23 A，则显示报警 61000。

程序代码
IDS=1 WHEN \$A_DLR[0] > 23.0 DO SETAL(61000)

2.2.3 Link 轴

2.2.3.1 简介

Link 轴指的是不由物理连接的 NCU、而由另一个 NCU 生成设定值的机床轴。因此在一些复杂设备上（诸如配备多个 NCU 的回转工作台），Link 轴和轴容器（参见章节“轴容器（页 109）”）组合使用便可以轮流运行 Link 组中各 NCU 的机床轴。

下图显示的示例中，机床轴 MA1 连接在 NCU1 上。机床轴 MA2 连接在 NCU2 上。通过在 NCU1 的通道中执行的零件程序通道轴 X 和 Z 插补运行。设定值在 NCU1 的插补器中生成。该设定值传输至 NCU1 的位置控制回路用于控制机床轴 MA1。设定值通过 NCU-Link 传输至 NCU2 的位置控制回路用于机床轴 MA2，之后再输出至驱动。

2.2 NCU-Link

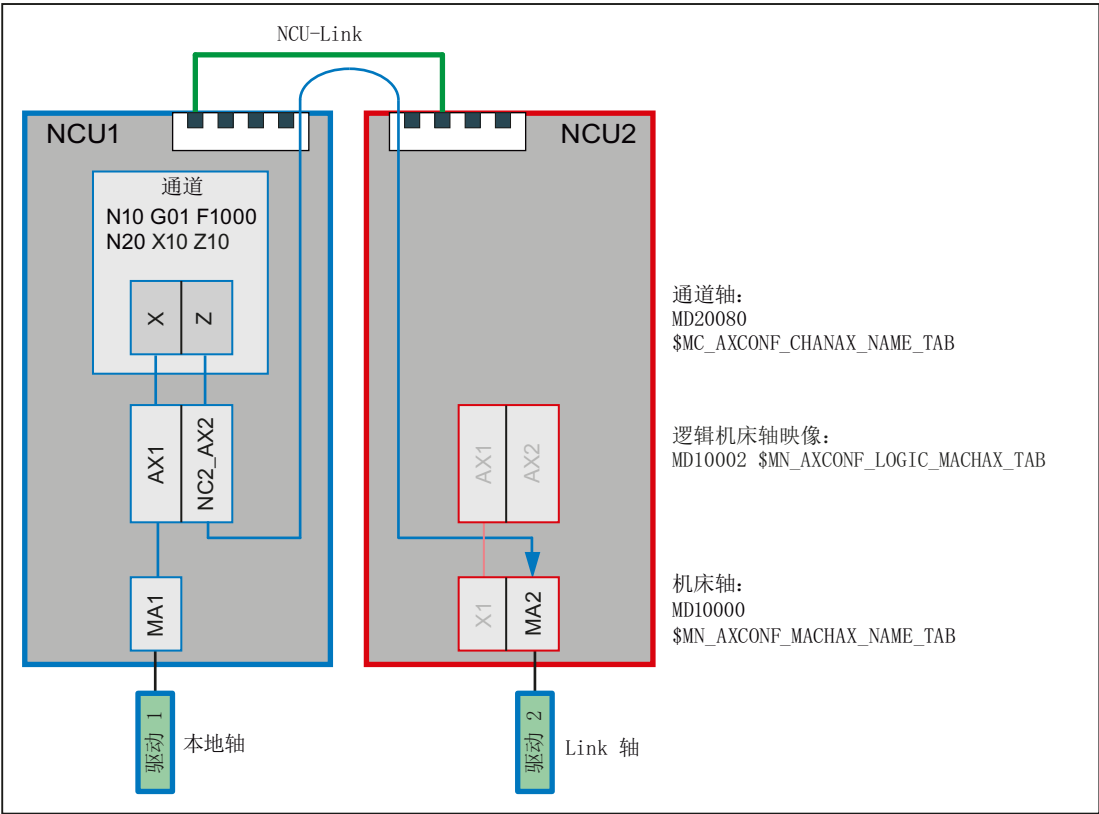


图 2-9 Link 轴

前提条件

使用 Link 轴的前提条件是按照“Link 通讯 (页 88)”章节的内容建立了 Link 通讯。

本地 NCU

Link 轴的本地 NCU 指的是其作为机床轴物理连接的 NCU。Link 轴的位置控制和 NC/PLC 接口信号的交换总是在本地 NCU 上进行。原则上可在 Link 组中的任意 NCU 上生成设定值。

上图：

- NCU1: 机床轴 MA1 的本地 NCU
- NCU2: 机床轴 MA2 的本地 NCU

2.2.3.2 Link 轴的名称

Link 轴的名称由机床轴物理连接的本地 NCU 的标识和机床轴通用名称 AXn 组成：
NC<ID>_<轴>

- <ID>: Link 组中的 NCU 的标识：
 MD12510 \$MN_NCU_LINKNO
 参见章节“参数设置：Link 通讯 (页 94)”
- <轴>: 机床轴通用名称：AX1, AX2, AX3, ...

2.2.3.3 参数设置

将几何轴或辅助轴指定为 Link 轴

直接指定

可在逻辑机床轴映射区中直接指定将几何轴或辅助轴指定为 Link 轴：

MD10002 \$MA_AXCONF_LOGIC_MACHAX_TAB[<轴>] = <Link 轴>

参数	含义
<轴>:	机床轴下标：0, 1, 2, ...（机床轴最大数量 - 1）
<Link 轴>:	Link 轴的名称：NCx_AXy, 参见“Link 轴的名称 (页 105)”一章

间接指定

可在逻辑机床轴映射区中间接指定容器槽来将几何轴或辅助轴指定为 Link 轴。之后容器槽将包含上文介绍的真正的 Link 轴名称。该轴即所谓的 Container-Link 轴。相关内容另见章节“轴容器 (页 109)”。

同步设定值输出

在生成设定值的 NCU 将 Link 轴设定值传输到本地 NCU 时，会产生一个插补周期的延时。为了使本地轴和 Link 轴的插补器在同一时间点将设定值输出至驱动，必须补偿此延时。为此，须在生成设定值的 NCU 上增加插补器和位置控制器间的缓存单元数量，使其比本地 NCU 的缓存单元多出一个。

MD18720 \$MN_MM_SERVO_FIFO_SIZE = 3

2.2 NCU-Link

示例

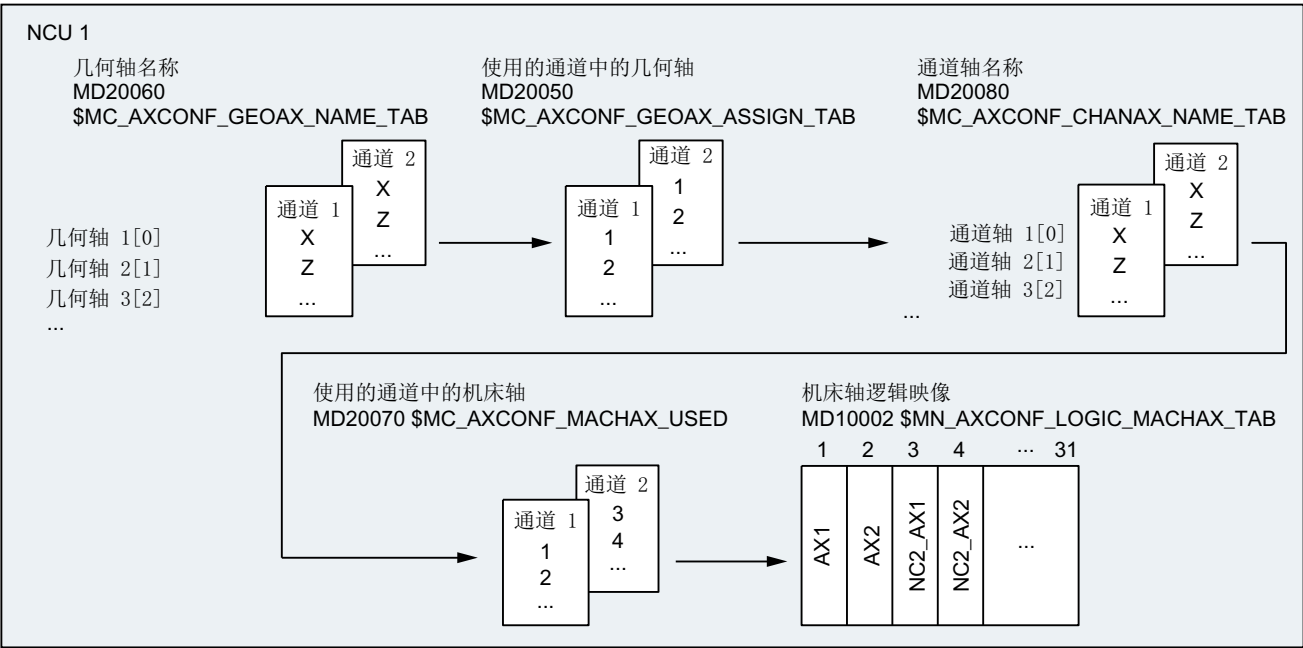


图 2-10 示例：Link 轴的参数设置

通道 1

几何轴 X / Z 指定为 NCU1 的本地机床轴 AX1 / AX2

通道 2

几何轴 X / Z 指定为 NCU2 的 Link 轴：NC2_AX1 / NC2_AX2

2.2.3.4 主轴上的辅助功能输出

在执行程序期间和“通过程序测试搜索”（SERUPRO）进行程序段搜索后，预定义的辅助功能 S、M3、M4、M5、M19 和 M70 会在编程了主轴的 NCU 上的某个通道中输出，另外还会在本地 NCU 上的某根轴上输出。

通道专用辅助功能的输出

- DB21, ... DBW68 (M 功能的扩展地址)
- DB21, ... DBD70 (M 功能 1)
- DB21, ... DBW98 (S 功能 1 的扩展地址)
- DB21, ... DBD100 (S 功能 1)

轴专用辅助功能的输出

- DB31, ... DBD78 (用于进给轴的 F 功能)
- DB31, ... DBW86 (用于主轴的 M 功能)
- DB31, ... DBD88 (用于主轴的 S 功能)

文档

有关辅助功能输出的详细信息请见:

功能手册“基本功能”，“向 PLC 输出辅助功能 (H2)”章节。

2.2.3.5 边界条件

机床轴的最大数量

使用 Link 轴时，每个 NCU 类型可同时使用的几何轴和辅助轴以及机床轴的最大数量仍然受限。

“Lead-Link 轴”和“Link 轴”功能

“Lead-Link 轴”和“Link 轴”功能所需的机床数据 MD18720

\$MN_MM_SERVO_FIFO_SIZE 中的设置不同，因此在一个 Link 组内无法同时使用这两个功能。

报警：一般特性

若 Link 轴本地 NCU 上的位置控制器存在故障，且对应报警的响应不为“NC 未就绪”，那么报警会通过 NCU-Link 传输至生成设定值的 NCU 并在该 NCU 上输出。

报警：急停时的特性

若一个 NCU 通过 NC/PLC 接口激活了急停请求，那么所有和该 NCU 物理连接的轴都切换至“跟踪”状态。其中也包括设定值当前由其他 NCU 生成的 Link 轴。假定这些 NCU 从该时间点起无法再进行有效加工，此时系统会额外生成一条报警来停止所有关联的轴运动。

报警应答

额外生成的这条报警必须通过 NC 复位应答。但是如果在应答这条额外报警前没有应答触发急停的原始报警，系统会生成另一条自行清除的报警，该报警会阻止轴运行或程序启动，直至触发急停的原始报警被应答。

2.2 NCU-Link

报警：报警响应“NC 未就绪”时的特性

若 Link 轴本地 NCU 上的位置控制器存在故障，且对应报警的响应为“NC 未就绪”，那么报警会通过 NCU-Link 传输至生成设定值的 NCU 并在该 NCU 上输出。此外还会在本地 NCU 上输出报警。

假定从该时间点生成设定值的 NCU 无法再进行有效加工，此时系统会额外生成一条报警来停止所有关联的轴运动。

报警应答

参见“报警：急停时的特性”一段的报警应答。

报警：报警响应“方式组未就绪”时的特性

若在包含多个通道的方式组内出现故障，且对应报警的响应为“方式组未就绪”，那么方式组所有通道中的运行将停止。若这些运行彼此之间基本无关联，可通过以下机床数据将响应改配为“通道未就绪”：

`MD11412 $MN_ALARM_REACTION_CHAN_NOREADY = TRUE`

结果

在 NC/PLC 接口中，信号 DB11 DBX26.3（方式组就绪）不复位，而是信号 DB21, ... DBX36.5（通道就绪）复位。

优点

报警响应只限于识别出故障的通道。根据需求，可通过 PLC 用户程序触发其他响应。

前提条件

未出现优先级高于“方式组未就绪”的报警响应。

补偿

不提供下列补偿功能：

- Link 轴：象限误差补偿（QEC）
- Container-Link 轴：垂度补偿（CEC）

Link 组中的一个 NCU 关闭

若 Link 组中的一个 NCU 关闭，那么 Link 组的其他所有 NCU 上的加工将终止，并显示报警。

NCU 组的启动

若在 Link 组中的一个 NCU 上触发了 NCK 复位，则该复位命令会传输至 Link 组中的其他所有 NCU，因此 Link 组的所有 NCU 都会执行一次热启动。

冲裁和步冲工艺

冲裁和步冲所需的高速输入/输出必须连到执行零件程序和轴插补的 NCU 上，并在该 NCU 上设置。PONS、SONS 等“快速冲裁和步冲”命令不适用于 Link 轴。

框架

在框架命令中编写的 Link 轴必须是几何轴。框架命令只作用于 Link 轴当前所在的通道。

主从转速/转矩耦合，主/从连接

主/从连接中所有进给轴/主轴的驱动必须连接至同一个 NCU。不过主主轴可作为 Link 轴由另一个 NCU 的通道运行。

2.2.4 轴容器

2.2.4.1 简介

轴容器(axis container)是一个环形数据结构，包含的单元的数目可设置。在轴容器中，这些单元也被称作槽（槽 1，槽 2，... 槽 n）。通过这些槽可将几何轴和/或辅助轴灵活指定为机床轴。槽中可以是一根 NCU 本地的机床轴（容器轴）或一根 Link 轴（Container-Link 轴）。

下图显示了一个包含四个槽的轴容器。在当前轴容器位置中，容器轴指下列机床轴：

容器轴	机床轴
CT1_SL1	NCU 1: AX1
CT1_SL2	NCU 1: AX2
CT1_SL3	NCU 2: AX1
CT1_SL4	NCU 2: AX2

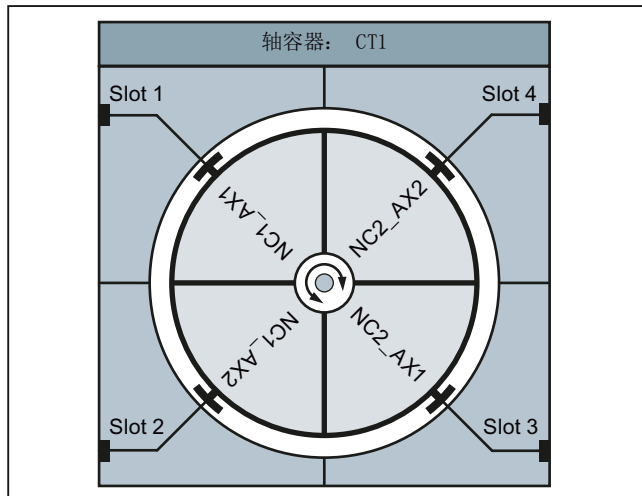


图 2-11 示例：包含四个槽的轴容器 CT1

规则

使用轴容器时须遵循下列规则：

- 在一个时间点上，只允许一根通道轴为轴容器中的机床轴。
- 不允许将轴容器的多个槽指向同一机床轴。
- 在一个时间点上，只能有一个通道具备直接或通过容器轴写入机床轴的权限。
- 一个通道的多根几何轴和/或辅助轴可指定为轴容器中的容器轴。

指定：几何轴或辅助轴 → 容器轴

在逻辑机床轴映射区 MD10002 \$MN_AXCONF_LOGIC_MACHAX_TAB 中可输入“容器”和“槽”，将一根几何轴或辅助轴指定为容器轴（机床轴），例如：输入容器“CT1”、槽“1”：

MD10002 \$MN_AXCONF_LOGIC_MACHAX_TAB[n] = CT1_SL1

因此在运行几何轴或辅助轴时，会运行该时间点上分给槽 1 的机床轴。

轴容器旋转

一旦轴容器涉及的所有通道通过程序命令获得使能，便开始执行轴容器旋转。轴容器旋转后，通道中的几何轴或辅助轴会指定为新的机床轴。

轴容器旋转的步长通过设定数据设置。

参见章节“参数设置 (页 112)”。

轴容器名称

通过轴容器名称 (<轴容器>) 可编写以下程序命令:

- 程序命令:
 - AXCTSWE (<轴容器>)
 - AXCTSWED (<轴容器>)
 - AXCTSWEC (<轴容器>)

名称可采用下列形式:

- CT<容器编号>: 在 CT 上加上轴容器编号。
示例: CT3
- <容器名称>: 通过 MD12750 \$MN_AXCT_NAME_TAB 自定义的轴容器名称。
示例: A_CONT3
- <轴名称>: 各通道中已知容器轴的名称。

隐性等待

若处理下列某个命令时, 系统会隐性等待请求的轴容器旋转完成:

- 语言命令, 会在此通道中触发轴容器包含的容器轴的运动
- GET (<通道轴名称>) 命令, 会获取对应的容器轴
- 下一个 AXCTSWE (<轴容器>) 命令, 写入了该轴容器

说明

IC (0) 命令也会触发等待, 必要时会导致同步 (虽然整体采用绝对尺寸, 但仍通过程序段改为增量尺寸)。

与轴位置的同步

轴容器旋转后, 若通道中新容器轴的绝对机床位置与之前不同, 则会按新位置同步 (内部 REORG)。

说明

SD41700 \$SN_AXCT_SWWIDTH[<轴容器>] 仅在采用新配置时更新。在 RTM/MS 的分步旋转后, 若容器位于初始位置后, 通常可将其继续**向前**旋转, 以达到容器的初始位置。而主轴鼓或回转工作台则须**向后**转, 防止测量电缆和电源电缆扭绞。

2.2 NCU-Link

参见

系统变量 (页 120)

编程 (页 118)

2.2.4.2 参数设置

机床数据

NC 专用机床数据

编号	名称 \$MN_	含义
MD12750	AXCT_NAME_TAB	轴容器名称
MD12760	AXCT_FUNCTION_MASK.位 x	轴容器专用功能
MD1270x	AXCT_AXCONF_ASSIGN_TABx	为轴容器的插槽指定机床轴
MD18720	MM_SERVO_FIFO_SIZE	IPO/SERVO 数据缓存的大小 提示 轴容器跨 NCU 时，涉及的所有 NCU 上都必须设置值 3。

轴容器名称

MD12750 \$MN_AXCT_NAME_TAB[<下标>] = “<名称>”:

参数	含义
<下标>:	0、1 到最大轴容器号
<名称>:	轴容器的名称（例如 CT1）

轴容器专用功能

MD12760 \$MN_AXCT_FUNCTION_MASK.位 x = <值>

参数	<值>	含义
位 0:	0	在直接旋转轴容器时（AXCTSWED），所有其他通道都必须处于复位状态。
	1	在直接旋转轴容器时（AXCTSWED），只有对容器轴具有插补权的通道须处于复位状态。

通过此机床数据激活轴容器专用功能。

为轴容器的插槽指定机床轴

MD1270x \$MN_AXCT_AXCONF_ASSIGN_TABx[<下标>] = <轴>

参数	含义
x:	1 到最大轴容器号
<下标>:	0、1 到最大槽号
<轴>:	本地机床轴的名称（例如 AX1）
	Link 轴的名称。参见章节“简介 (页 103)”。

在一个轴容器内，必须从槽 0 开始连续为各个槽指定机床轴，不能遗漏某个槽。

设定数据

轴容器旋转的步长

SD41700 \$SN_AXCT_SWWIDTH[<下标>] = <步长>

参数	含义
<下标>:	0、1 到最大轴容器号
<步长>:	轴容器转过的槽数

轴容器旋转的示意图

轴容器旋转通过程序命令使能。参见章节“编程 (页 118)”。

以图 1 的左图为例，当轴容器处于**初始位置**时，槽 1 上是 Link 轴 NCU1_AX1。

转过 1 步后（图 1 右图），槽 1 上现在是 Link 轴 NCU2_AX2。

2.2 NCU-Link

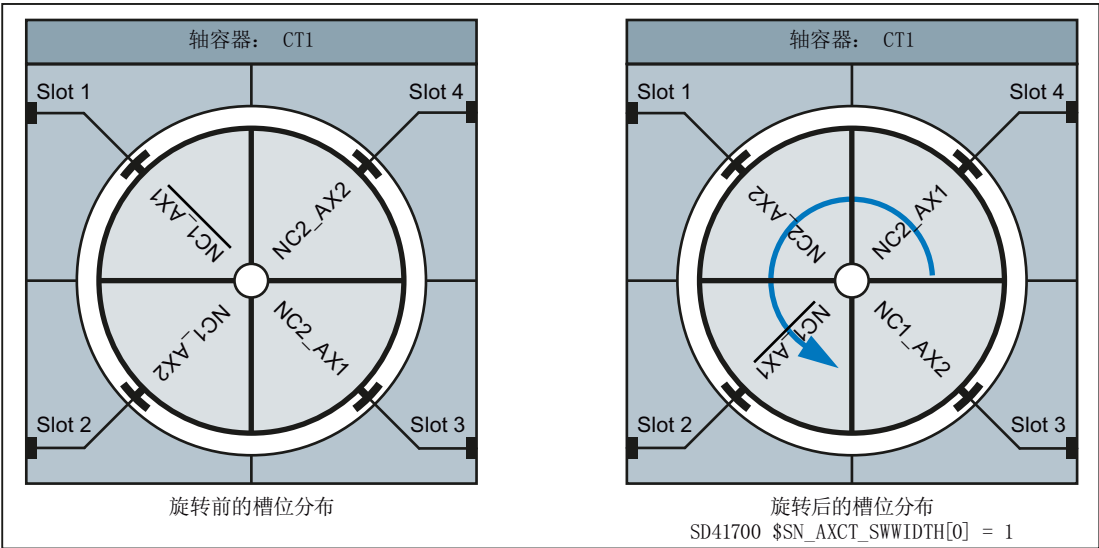


图 2-12 轴容器旋转，图 1

从初始位置起转过 2 步后（图 2 左图），槽 1 上是 Link 轴 NCU2_AX1。

从初始位置起转过 -1 步后（图 2 右图），槽 1 上是 Link 轴 NCU1_AX2。

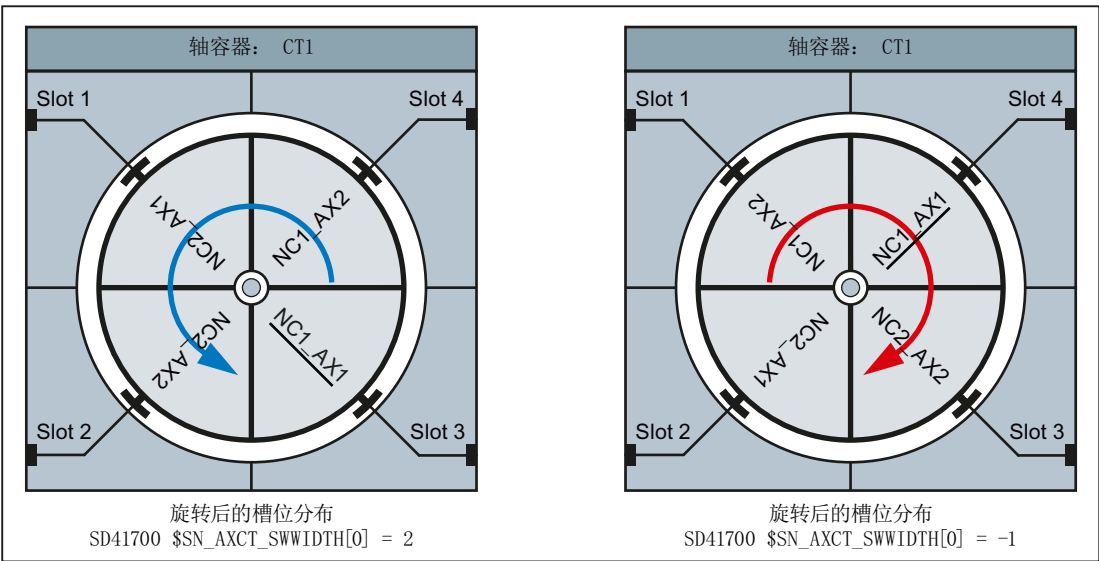


图 2-13 轴容器旋转，图 2

含 Container-Link 轴的轴容器

若轴容器包含 Container-Link 轴，则必须在 Link 组中的主 NCU 上设置轴容器（MD12510 \$MN_NCU_LINKNO == 1）。

轴机床数据的调整

所有容器轴上所有具有“CTEQ”（ConTainer EQual）属性的机床数据必须采用相同的赋值。若赋值不同，则会由系统自动调整。

控制系统启动

控制系统启动时，所有机床数据都会调整为第一个槽的容器轴的值。若此时修改机床数据的赋值，则会显示以下信息：“轴容器 <n> 中的轴机床数据已经过调整”。

机床数据的修改

在修改了任意一根容器轴的机床数据后，新值会立即传输至所有其他的容器轴。此时会显示以下信息：“请注意，正在为所有容器轴设置此机床数据”。

槽的修改

若为轴容器槽指定了新的一根机床轴（MD127xx AXCT_AXCONF_ASSIGN_TAB<x>），则会显示以下提示：“下次启动时将调整轴容器 <n> 中的轴的机床数据”。

说明

Container-Link 轴

如果容器轴同时是 Link 轴，系统会为 Link 组中所有轴容器涉及的 NCU 调整机床数据。

参数设置示例

假设

NCU	组件
NCU1 :	通道 1，几何轴 X / Z → 第 1 通道轴/第 2 通道轴 通道 2，几何轴 X / Z → 第 1 通道轴/第 2 通道轴 机床轴：AX1，AX2 轴容器 CT1，包含 4 个槽
NCU2 :	机床轴： AX1, AX2

2.2 NCU-Link

参数设置：NCU1

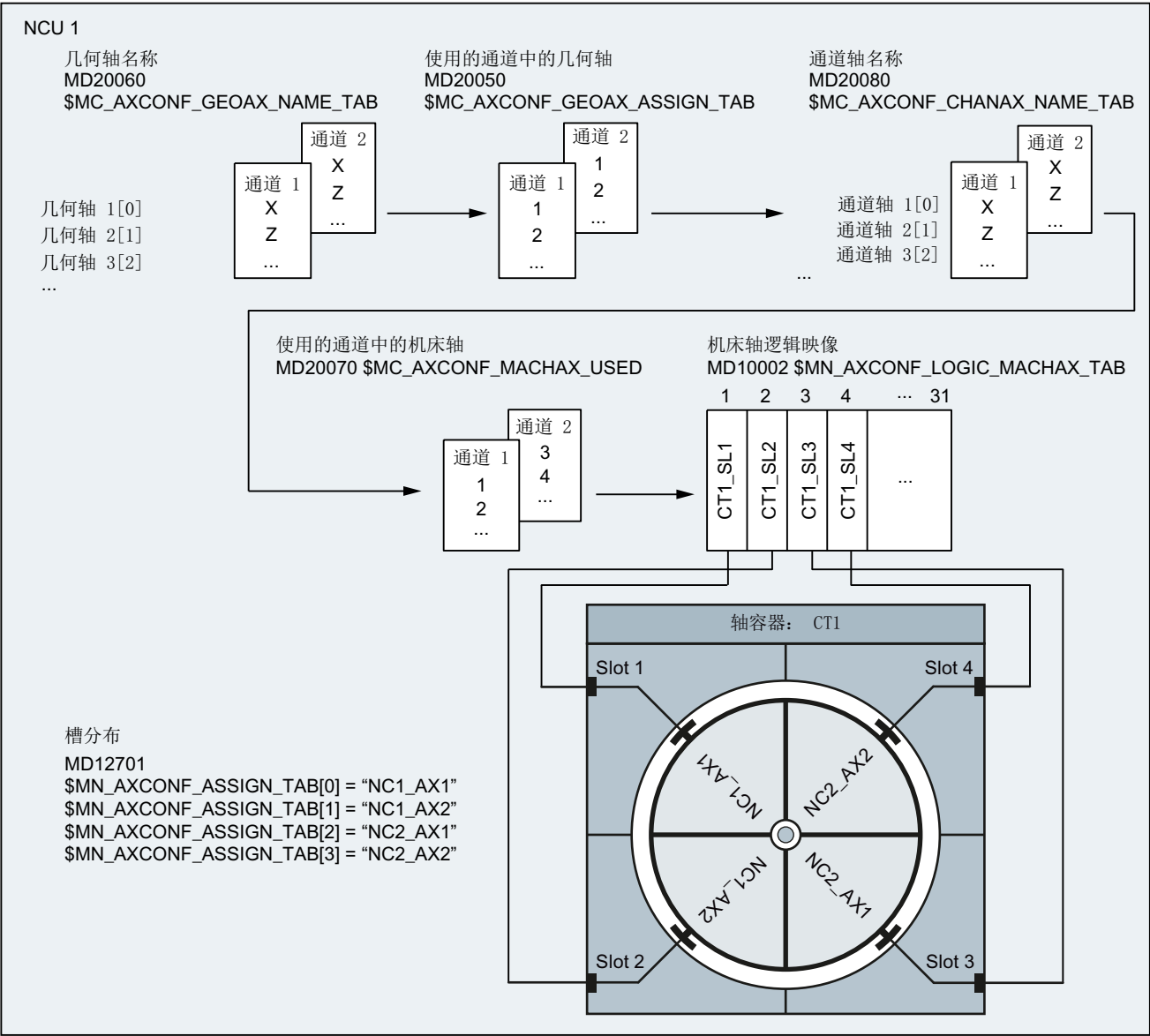


图 2-14 示例：通道轴和轴容器的参数设置

结果

编程 NCU1 第 1 通道和第 2 通道中的几何轴 X 和 Z 后，轴容器中从当前位置开始运行的轴有：

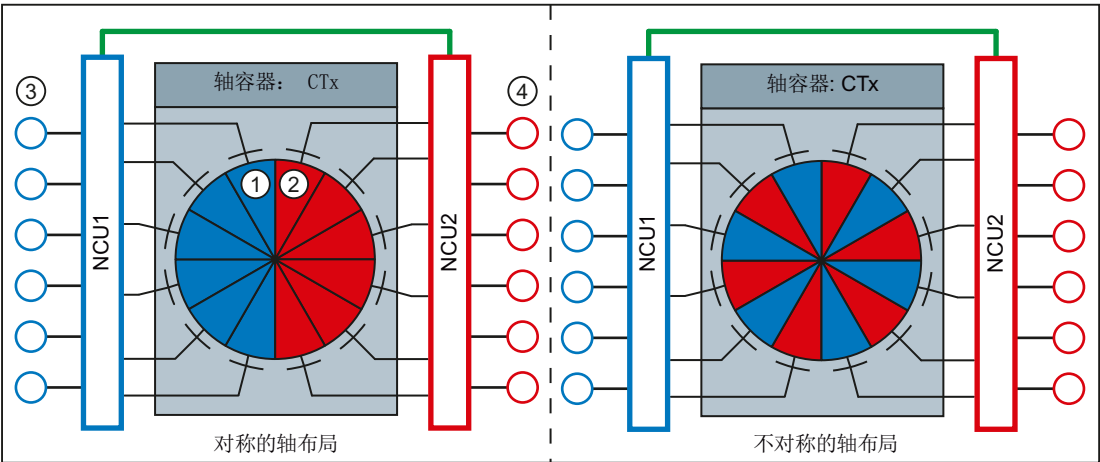
- NCU1 的本地机床轴 AX1 和 AX2。
- NCU2 的机床轴 AX1 和 AX2（容器轴兼 Link 轴）。

轴容器旋转的相关内容请见“编程 (页 118)”章节。

参数设置提示

容器轴布局和通讯负载率

在配备多个 NCU、与轴容器组合使用来交替运行不同 NCU 的轴（Link 轴）的设备上，轴容器内 Link 轴的布局决定了 Link 通讯的负载率。



- ① 蓝色槽指明 NCU1 上连接的一个驱动
- ② 红色槽指明 NCU2 上连接的一个驱动
- ③ NCU1 上连接的驱动
- ④ NCU2 上连接的驱动

图 2-15 轴布局

- 对称的轴布局
采用对称的轴布局时，每个 NCU 一开始时只运行本地轴。因此此时不会发生 Link 通讯。但轴容器每旋转一次，Link 通讯的负载率便有所增加，在所有 NCU 只运行 Link 轴时达到最大值。
- 不对称的轴布局
采用不对称的轴布局时，每个 NCU 一开始便会运行本地轴和 Link 轴。与对称的轴布局相比，此方案能够实现“恒定”保持在中等水平的 Link 通讯负载率。

驱动布局和通讯负载率

在配备多个 NCU、与轴容器组合使用来交替运行不同 NCU 的轴（Link 轴）的设备上，NCU 上连接的驱动的布局决定了 Link 通讯的负载率。

2.2 NCU-Link

- 对称的驱动布局
采用对称的驱动布局时，通过轴容器访问的驱动对称地连接至两个 NCU。采用此布局时，可通过这两个 NCU 的逻辑机床轴映射区（LAI）确定最大驱动数量。

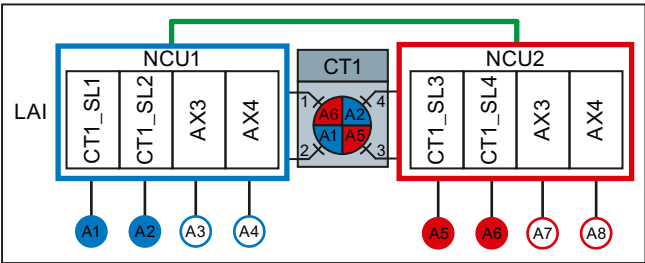


图 2-16 对称的驱动布局

- 不对称的驱动布局
采用不对称的驱动布局时，通过轴容器访问的驱动全部连接至 NCU1。采用此布局时，只可通过 NCU2 上的逻辑机床轴映射区（LAI）确定最大驱动数量。NCU1 的 LAI 中只能输入最大数量减去 NCU2 连接的驱动数量的差。为了在 NCU1 上也能使用最大数量的驱动，必须将驱动连接至 NCU2，然后从 NCU1 通过 NCU Link 访问驱动。这会导致较高的循环 Link 通讯负载。

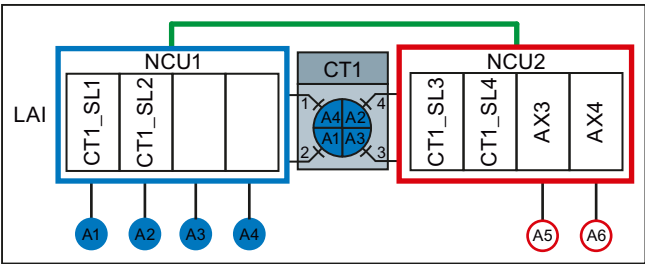


图 2-17 不对称的驱动布局

2.2.4.3 编程

功能

命令 AXCTSWE 或 AXCTSWED 可使能指定轴容器的旋转。

命令 AXCTSWEC 可撤销轴容器的旋转使能。

句法

AXCTSWE (<ID>)
AXCTSWED (<ID>)
AXCTSWEC (<ID>)

含义

AXCTSWE:	使能轴容器旋转 命令 AXCTSWE 不会中断程序处理。 一旦轴容器涉及的所有通道都获得使能，轴容器便立即开始旋转。 轴容器旋转的步长通过设定数据 SD41700 \$SN_AXCT_SWWIDTH 设置 (参见章节“参数设置 (页 112)”)
AXCTSWED:	使能轴容器旋转，不管参与轴容器的其他通道是否获得使能 轴容器旋转的步长通过设定数据 SD41700 \$SN_AXCT_SWWIDTH 设置 (参见章节“参数设置 (页 112)”) 提示 - 此命令用于简化零件程序或同步动作的调试。 - 轴容器涉及的其它通道的相关特性可通过以下机床数据设定： MD12760 \$MN_AXCT_FUNCTION_MASK, 位 0 参见章节“参数设置 (页 112)”。
AXCTSWEC:	撤销轴容器旋转使能 提示 只有在轴容器还没有开始旋转时，才能撤销旋转使能： \$AN_AXCTSWA[<轴容器>] == 0 参见“系统变量 (页 120)”
<ID>:	轴容器或容器轴的名称： CT<编号>: 轴容器的缺省名称： MD12750 \$MN_AXCT_NAME_TAB 示例: CT1 <容器>: 轴容器的用户定义名称： MD12750 \$MN_AXCT_NAME_TAB 示例: CONTAINER_1 <轴>: 通道中已知的容器轴的名称

文档

命令 AXCTSWEC 在同步动作中的应用详见：
功能手册“同步动作”，“详细说明”章节 > “同步动作中的动作” > “撤销轴容器旋转使能 (AXCTSWEC)”

2.2 NCU-Link

2.2.4.4 系统变量

容器专用系统变量

系统变量	描述
\$AC_AXCTSWA[<ID>]	轴容器旋转时特定通道状态
\$AN_AXCTSWA[<ID>]	轴容器旋转时特定 NCU 状态
\$AN_AXCTSWE[<ID>]	轴容器旋转时特定槽状态
\$AN_AXCTAS[<ID>]	轴容器当前已转过的槽数
ID: 轴容器名称或容器轴的名称	

NC 专用系统变量

系统变量	描述
\$AN_LAI_AX_IS_AXCTAX ¹⁾	状态: LAI 轴 == 容器轴 逻辑机床轴映射 (MD10002 \$MN_AXCONF_LOGIC_MACHAX_TAB) 中轴容器相关的机床轴。
\$AN_LAI_AX_IS_LEADLINKAX ¹⁾	状态: LAI 轴 == Lead-Link 轴
\$AN_LAI_AX_IS_LINKAX ¹⁾	状态: LAI 轴 == Link 轴
\$AN_LAI_AX_TO_IPO_NC_CHANAX[<ID>]	通道编号和通道轴编号, 或 NCU ID 与全局轴编号
\$AN_LAI_AX_TO_MACHAX[<ID>]	NCU ID 和机床轴编号
¹⁾ 位标记: 位 n $\triangleq$ MD10002 \$MN_AXCONF_LOGIC_MACHAX_TAB 中的 LAI 轴 (n+1) ID: LAI 轴编号 NCU ID: MD12510 \$MN_NCU_LINKNO 中的值	

文档

对系统变量的详细说明请见:
参数手册 “系统变量”

参见

分析轴容器系统变量 (页 134)

2.2.4.5 使用轴容器的加工（图示）

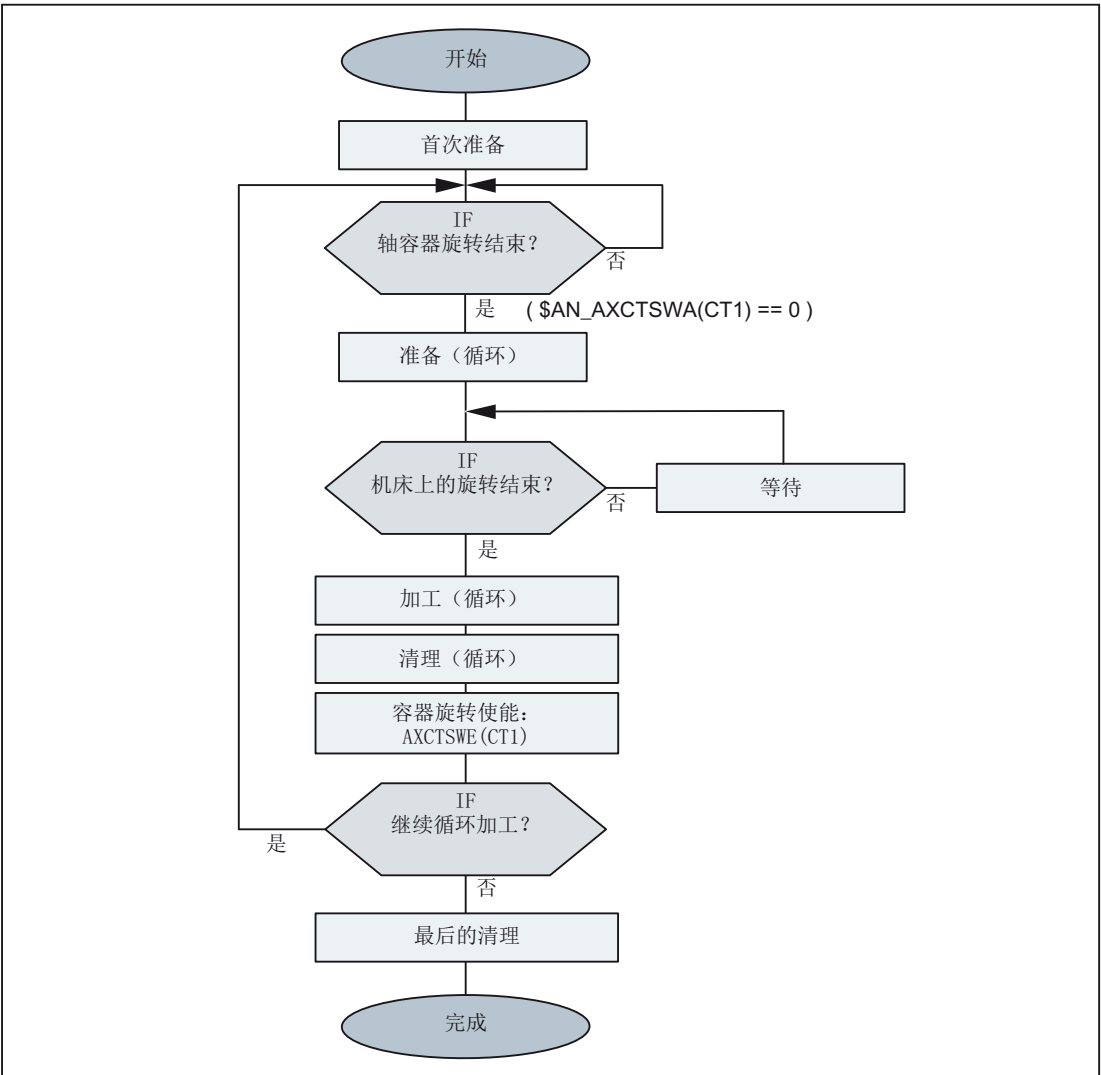


图 2-18 示例：回转工作台上一个工作站的加工步骤示意图

2.2.4.6 各种运行状态下的特性

启动（上电）

控制系统启动时，总是会读取机床数据中确定的槽和轴的对应关系，而不考虑上次关闭时轴容器的实际状态：

MD1270x \$MN_AXCT_AXCONF_ASSIGN_TABx

注意
设定状态和实际状态的一致 控制系统启动后，用户/机床制造商须自行确定轴容器设定状态和机床实际状态之间的差别，并通过适当旋转轴容器来消除差别。

运行方式切换

如果在 **AUTO** 运行方式 中使能了轴容器的旋转，而现在又切换到 **JOG** 方式，则该轴容器中包含的容器轴将无法运行。

特定通道的复位状态

一旦轴容器涉及的通道处于复位状态，那么此通道便不需要轴容器旋转的使能。只要剩下的其他通道获得使能即可旋转轴容器。

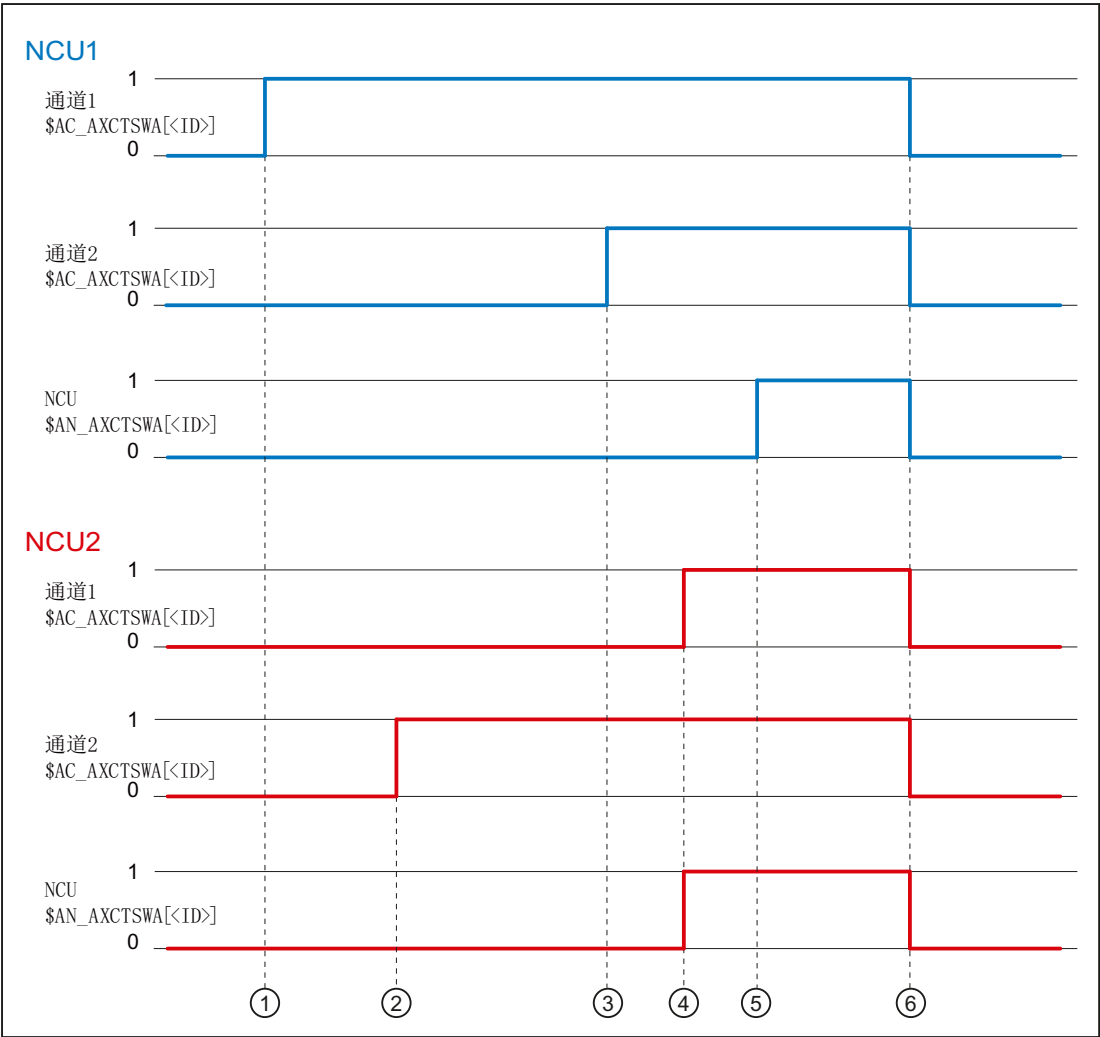
程序段搜索

轴容器旋转的使能和激活（AXCTSWE）无法在一条程序段中输出，而必须分别在一条动作程序段中输出，即：轴容器的状态会随其他通道的状态变化，对每个旋转命令都不尽相同。

2.2.4.7 撤销轴容器旋转使能时的特性

命令 AXCTSWE 可使能特定通道内的轴容器旋转。命令 AXCTSWEC 可再次撤销使能。

下图是一个示例，利用轴容器专用的系统变量来展示轴容器旋转的流程。该轴容器涉及两个 NCU，每个 NCU 各有两个通道。



- ① NCU1, 通道 1: 通过 AXCTSWE 命令使能
- ② NCU2, 通道 2: 通过 AXCTSWE 命令使能
- ③ NCU1, 通道 2: 通过 AXCTSWE 命令使能 → 现在 NCU1 中的所有通道均获得使能 → NCU1 的整体使能状态通过 Link 通讯传输至 NCU2
- ④ NCU2, 通道 1: 通过 AXCTSWE 命令使能 → 现在 NCU2 中的所有通道均获得使能 → NCU2 的整体使能状态通过 Link 通讯传输至 NCU1
NCU2 中现具备所有 NCU (NCU2 和 NCU1) 所有通道的使能 → NCU2 中执行轴容器旋转
- ⑤ NCU1: NCU1 中现具备所有 NCU (NCU1 和 NCU2) 所有通道的使能 → NCU1 中执行轴容器旋转
- ⑥ NCU1 / NCU2: 轴容器旋转完成

图 2-19 跨 NCU 的轴容器旋转使能

2.2 NCU-Link

若需撤销已作出的使能，必须起码有一个轴容器涉及的通道（不管是 NCU1 还是 NCU2 的）尚未获得使能。因此必须在时间点 ④ 之前执行撤销。

一旦所有 NCU 的所有通道都获得使能（时间点 ④），将无法再撤销使能。在此情形下 AXCTSWEC 命令保持无效。此时不会向用户发送反馈信息。

参见

编程 (页 118)

2.2.4.8 边界条件

进给轴方式

无论容器轴是作为进给轴运行还是作为定位主轴(POSA, SPOSA)运行，轴容器的旋转都要等轴到达编程位置后才执行。

主轴

- 作为主轴运行的容器轴在轴容器旋转期间继续旋转。
- 主轴的闭环控制方式（转速控制/位置控制）作用于对应的机床轴。轴容器旋转时，所设置的闭环控制方式会随机床轴“游移”。
- 在编写了一些涉及通道内主主轴的命令时，必须保证在执行该命令前通道中的机床轴已经具有了对应的主轴号：

```
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[<轴>] == 主主轴的编号
```

说明

在作为主轴运行容器轴时，用户/机床制造商须自行确保进行轴容器旋转后通道中仍存在对应的机床轴。

零点偏移

说明

轴容器旋转后，用户/机床制造商须自行确保通道中生效的零点偏移和新的机床轴彼此协调。

连续路径运行

若通道中连续路径运行生效且进行了轴容器旋转，那么后续的容器轴编程将使连续路径运行中断。若容器轴不是路径轴，连续路径运行也会中断。

PLC 轴

若将使能了轴容器旋转的容器轴设为 PLC 轴，在轴容器旋转完成后才能进行状态切换。

命令轴

若将使能了轴容器旋转的容器轴设为命令轴，命令轴在轴容器旋转完成后才能运行。

往复轴

若将使能了轴容器旋转的容器轴设为往复轴，在轴容器旋转完成后才能进行状态切换。

外部零点偏移

“外部零点偏移”以机床坐标系（MCS）为参照。因此，若其中一根容器轴上设有一个“外部零点偏移”，轴容器旋转将被系统拒绝，并发出报警 4022。

轴框架

容器轴为通道轴时，轴容器旋转后其框架将不会保留。轴容器旋转后通道轴会指定为一根新的机床轴，而其框架则是基于机床轴的，因此旋转后通道轴的框架将变化。若旋转前后的框架不一致，系统会进行同步（内部 REORG）。

说明

通道轴与机床轴之间的关系随轴容器旋转变化的。旋转后将保留当前框架。用户须自行确保旋转后的框架选择正确。这例如可通过编程基本框架位来实现。

坐标转换

若作为主轴运行的容器轴涉及坐标转换，那么在使能轴容器旋转前必须撤销坐标转换。

轴耦合

若容器轴处于轴耦合中，在使能轴容器旋转前必须通过 COUPOF 撤销耦合。旋转完成后，可立即通过 COUPON 重新选择耦合。不需要重新定义耦合。

2.2 NCU-Link

龙门轴

龙门轴不允许用作容器轴。

运行到固定挡块

若容器轴位于固定挡块上，则不可执行轴容器旋转。

驱动报警

若存在针对容器轴驱动的报警，则不会执行轴容器旋转。

2.2.5 Lead-Link 轴

2.2.5.1 简介

在轴耦合中，若主动轴和从动轴不连接在同一个 NCU 上，则必须由从动轴所在的 NCU 控制的 Link 轴实现耦合。此时该 Link 轴被称为 Lead-Link 轴。

主动轴的设定值将在与插补周期同步地通过 NCU-Link 传输至 Lead-Link 轴。与之对应，Lead-Link 轴和从动轴的实际值及状态数据也会以相反的方向传输至主动轴。

Lead-Link 轴设为从动轴的本地主动轴。

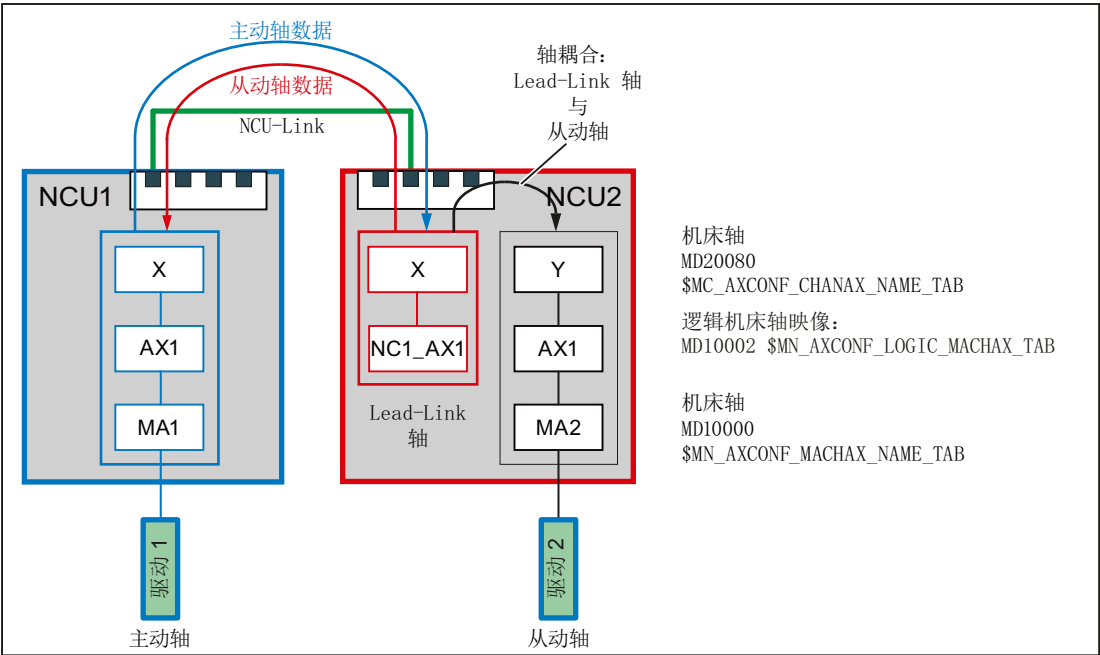


图 2-20 Lead-Link 轴

轴耦合

Lead-Link 轴可与下列轴耦合组合使用:

- 引导值耦合
- 跟随
- 切向跟踪
- 电子齿轮箱 (ELG)
- 同步主轴

前提条件

NCU 必须通过 NCU Link 通讯。参见章节 “Link 通讯 (页 88)”

2.2 NCU-Link

2.2.5.2 参数设置

Link 通讯

NC 专用机床数据

编号	名称 \$MN_	含义
MD12510	NCU_LINKNO	Link 组内 NCU 的唯一数字标识。标识必须从 1 开始以升序给定，且必须连贯。 取值范围：1、2 到最大 NCU 编号 提示 NCU 标识为 1 的 NCU 是 Link 组中的主 NCU。Link 轴和轴容器的参数设置只允许通过主 NCU 的机床数据和设定数据进行。
MD18780	MM_NCU_LINK_MASK.位 0	激活 Link 通讯
MD18781	NCU_LINK_CONNECTIONS	内部 Link 连接的数量 提示 建议保留缺省值 0（由 NC 测定数量）。
MD18782	MM_LINK_NUM_OF_MODULES	通过 NCU-Link 相互连接的 NCU 的数量。

设定值同步

NC 专用机床数据

编号	名称 \$MN_	含义
MD18720	MM_SERVO_FIFO_SIZE	IPO/SERVO 数据缓存的大小 主动轴的设定值通过 NCU-Link 传送到 Lead-Link 轴的 NCU 会产生两个插补周期的延时。该延时可通过在主动轴的 NCU 上和 Lead-Link 轴的 NCU 上设置 IPO/SERVO 数据缓存大小来补偿： - 主动轴的 NCU：4 - Lead-Link 轴的 NCU：2

注意

同时使用 **Lead-Link 轴**和**跨 NCU 的轴容器**时，由于轴容器的限制，必须设置机床数据 MD18720 \$MN_MM_SERVO_FIFO_SIZE = 3。这样一来便无法同步输出主动轴和从动轴的设定值。输出时间点相差一个插补周期。

主动轴、Lead-Link 轴和从动轴**NC 专用机床数据**

编号	名称 \$MN_	含义
MD10000	AXCONF_MACHAX_NAME_TAB	机床轴名称
MD10002	AXCONF_LOGIC_MACHAX_TAB	机床轴逻辑映像

通道专用机床数据

编号	名称 \$MC_	含义
MD20070	AXCONF_MACHAX_USED	所使用的机床轴

轴专用机床数据

编号	名称 \$MA_	含义
MD30554	AXCONF_ASSIGN_MASTER_NCU	主 NCU 若一根机床轴可由多个 NCU 运行，则必须将其中的一个 NCU 定义为主 NCU。控制系统启动后将由此 NCU 生成设定值。

2.2.5.3 用于设定主值的系统变量

通过下列系统变量可在主动轴的 NCU 上设定主值：

- 位置主值：\$AA_LEAD_SP[<主动轴>]
- 速度主值：\$AA_LEAD_SV[<主动轴>]

2.2 NCU-Link

这些变量被修改时，新值也会通过 NCU-Link 传输至从动轴的 NCU。

说明

系统变量的传输优先级低于 Link 变量。

2.2.5.4 边界条件

必须遵循以下边界条件：

- 主动轴不允许为 Link 轴
- 主动轴不允许为容器轴
- 主动轴不允许为龙门轴
- 主动轴不允许在自身的 NCU 内交换（参见章节“跨通道取进给轴/主轴 (页 338)”）
- 与 Lead-Link 轴的耦合不允许级联运行
- Lead-Link 轴不允许脱离主动轴单独运行

说明

“Lead-Link 轴”和“Link 轴”功能

“Lead-Link 轴”和“Link 轴”功能所需的机床数据 MD18720

\$MN_MM_SERVO_FIFO_SIZE 中的设置不同，因此在一个 Link 组内无法同时使用这两个功能。

2.2.5.5 示例

包含 Lead-Link 轴的轴耦合的详细设置及编程示例请见：“示例”章节 > “Lead-Link 轴 (页 146)”。

2.2.6 一个 Link 组内的单位制

为进行跨 NCU 的插补，一个 Link 组中所有 NCU 必须激活相同的单位制。

通过 HMI 实现全局单位制转换

- 只有一个 Link 组中的所有 NCU 满足下列条件后，才可统一转换单位制，即在其中一个 NCU 的 HMI 操作界面上可统一转换所有 NCU 的单位制：
- MD10260 \$MN_CONVERT_SCALING_SYSTEM = 1
 - 对所有通道：
MD20110 \$MC_RESET_MODE_MASK，位 0 = 1
 - 所有通道处于复位状态
 - 没有轴在运行方式 JOG 或 DRF 下运行或通过 PLC 运行
 - “恒定砂轮圆周速度（GWPS）”功能未生效
- 若 Link 组中的一个 NCU 上未满足上述条件，则不会在 Link 组中的任何 NCU 上转换单位制。

不同的单位制

虽然 Link 组激活，但只要不进行跨 NCU 的插补，各 NCU 仍可采用不同的单位制。为此，必须针对 NCU 在零件程序或同步动作中通过 G 命令（G70、G71、G700、G710）设置各个 NCU 的单位制。

文档

功能手册之基本功能分册；章节“速度，设定值/实际值系统，闭环控制（G2）”

2.3 示例

2.3.1 Link 轴

编程示例： 2 个 NCU，各配备一根 Link 轴

NCU 1

机床数据	注释
通用 Link 数据:	
\$MN_NCU_LINKNO = 1	主 NCU
\$MN_MM_NCU_LINK_MASK = 1	激活 NCU-Link

2.3 示例

机床数据	注释
\$MN_MM_SERVO_FIFO_SIZE = 3	插补器和位置控制环之间的数据缓存的大小
\$MN_MM_LINK_NUM_OF_MODULES = 2	Link 模块的数量
机床轴逻辑映像 (LAI) :	
\$MN_AXCONF_LOGIC_MACHAX_TAB[0] = "AX1"	本地机床轴
\$MN_AXCONF_LOGIC_MACHAX_TAB[1] = "AX2"	本地机床轴
\$MN_AXCONF_LOGIC_MACHAX_TAB[2] = "NC2_AX3"	Link 轴
NCU 标识 + 机床轴名称, 组成系统范围内唯一的轴名:	
\$MN_AXCONF_MACHAX_NAME_TAB[0] = "NC1_A1"	
\$MN_AXCONF_MACHAX_NAME_TAB[1] = "NC1_A2"	
\$MN_AXCONF_MACHAX_NAME_TAB[2] = "NC1_A3"	
将通道轴指定为机床轴:	
\$MC_AXCONF_MACHAX_USED[0] = 1	第 1 通道轴指定为 LAI[0] 的机床轴
\$MC_AXCONF_MACHAX_USED[1] = 2	第 2 通道轴指定为 LAI[1] 的机床轴
\$MC_AXCONF_MACHAX_USED[2] = 3	第 3 通道轴指定为 LAI[2] 的机床轴

NCU 2

机床数据	注释
通用 Link 数据:	
\$MN_NCU_LINKNO = 2	从 NCU
\$MN_MM_NCU_LINK_MASK = 1	激活 NCU-Link
\$MN_MM_SERVO_FIFO_SIZE = 3	插补器和位置控制环之间的数据缓存的大小
\$MN_MM_LINK_NUM_OF_MODULES = 2	Link 模块的数量
机床轴逻辑映像 (LAI) :	
\$MN_AXCONF_LOGIC_MACHAX_TAB[0] = "AX1"	本地机床轴
\$MN_AXCONF_LOGIC_MACHAX_TAB[1] = "AX2"	本地机床轴

机床数据	注释
\$MN_AXCONF_LOGIC_MACHAX_TAB[2] = "NC1_AX3"	Link 轴
NCU 标识 + 机床轴名称，组成系统范围内唯一的轴名：	
\$MN_AXCONF_MACHAX_NAME_TAB[0] = "NC2_A1"	
\$MN_AXCONF_MACHAX_NAME_TAB[1] = "NC2_A2"	
\$MN_AXCONF_MACHAX_NAME_TAB[2] = "NC2_A3"	
将通道轴指定为机床轴：	
\$MC_AXCONF_MACHAX_USED[0] = 1	第 1 通道轴指定为 LAI[0] 的机床轴
\$MC_AXCONF_MACHAX_USED[1] = 2	第 2 通道轴指定为 LAI[1] 的机床轴
\$MC_AXCONF_MACHAX_USED[2] = 3	第 3 通道轴指定为 LAI[2] 的机床轴

2.3.2 轴容器协调

下表按时间顺序列明，此处假定轴容器中只有两根通道轴。

2.3.2.1 轴容器旋转，零件程序不等待旋转结束

通道 1	通道 2	注释
AXCTWE(C1)	零件程序 ...	为通道 1 使能轴容器旋转
零件程序，不含容器轴运动	零件程序 ...	
	AXCTSWE(C1)	为通道 2 使能轴容器旋转；由于两个通道均已获得旋转使能，因此执行旋转
零件程序，含容器轴运动	零件程序，含容器轴运动	无等待

2.3 示例

2.3.2.2 轴容器旋转，零件程序隐形等待旋转结束

通道 1	通道 2	注释
AXCTWE(C1)	零件程序 ...	为通道 1 使能轴容器旋转
零件程序，含容器轴运动	零件程序 ...	通道 1 隐性等待轴容器旋转
	AXCTSWE(C1)	为通道 2 使能轴容器旋转；执行旋转。 继续通道 1。

2.3.2.3 仅在一个通道中执行轴容器旋转（例如启动时）

通道 1	通道 2	注释
AXCTWED(C1)	处于复位状态	直接执行旋转

2.3.3 分析轴容器系统变量

2.3.3.1 有条件的跳转

通道 1	注释
N100 AXCTWE(CT1)	通道 1: 使能轴容器 CT1 的旋转
MARKE1:	
N200 ...	零件程序，不含容器轴运动
IF \$AC_AXCTSWA[CT1] == 1 GOTOB MARKE1	如果轴容器 CT1 的旋转仍生效 那么便跳转到 MARKE1 ELSE（轴容器 CT1 的旋转完成）
N300 ...	零件程序，含容器轴运动

2.3.3.2 通过 \$AN_AXCTSWA 静态同步

通道 1	注释
IDS =1 EVERY \$AN_AXCTSWA[CT1] == 1 DO M99	静态同步动作： 在轴容器旋转开始时总是输出辅助功能 M99。
	文档：功能手册之同步动作分册

2.3.3.3 安全等待轴容器旋转结束

若需安全等待轴容器旋转结束，可根据具体情形采用以下示例方案。

示例 1

rl = \$AN_AXCTAS[ctl]; 读取当前轴容器位置
AXCTSWE(ctl); 使能轴容器旋转
WHILE (rl == \$AN_AXCTAS[ctl]); 等待到达新轴容器位置
ENDWHILE; 完成旋转

示例 2，用于第 1 通道

CLEARM(9) ; 删除同步标记 9
AXCTSWE(ctl); 使能轴容器旋转
; 通过同步动作等待
; 轴容器旋转完成
WHEN \$AN_AXCTSWA[ctl] == TRUE DO SETM(9) ; 设置标记 9 并
WAITMC(9, 1); 等待同步标记 9
; 在第 1 通道中

示例 3.1 使用内部等待

M3 S100; 重新编程轴容器主轴
; 内部等待轴容器旋转
; 结束

2.3 示例

示例 3.2 使用内部等待

```
x=IC(0); 重新编程容器轴 x
; 内部等待轴容器旋转
; 结束
```

示例 3.3 使用内部等待

```
AXCTSWE(CTL); 若再次使能轴容器旋转,
; 则会内部等待前一个
; 轴容器旋转结束
N2150 WHILE (rl == $AN_AXCTAS[ctl])
```

说明

NC 程序中的编程:

```
WHILE ($AN_AXCTSWA[n] == 0)
ENDWHILE
```

无法用于确定先前的轴容器旋转是否结束。尽管从 SW 7.x 起 \$AN_AXCTSWA 会执行隐性的预处理停止，但仍不可采用此编程，因为此程序段可能会因重组而中断，之后由于轴容器旋转结束系统变量会重新输出 0。

2.3.4 多主轴车床的配置

引言

下面的示例将采用:

- NCU-Link 组中的多个 NCU
- 通过轴容器实现的灵活配置

机床说明

- 此机床在主轴鼓 A（正面加工）一侧有：
 - 4 根主主轴：HS1 至 HS4
每个主主轴均可进料（液压进料机构，轴：STN1-STN4）。
 - 4 个十字滑块
 - 每个滑块有两根轴。
 - 可选择在每个滑块上安装一把动力刀具 S1-S4。
- 此机床在主轴鼓 B（背面加工）一侧有：
 - 4 根副主轴：GS1 至 GS4
 - 4 个十字滑块
 - 每个滑块有两根轴。
 - 可选择在每个滑块上安装一把动力刀具 S5-S8。
 - 每根副主轴都可通过一根线性轴移动其位置，例如：将主主轴中的棒料接收到主轴鼓 B 中进行背面加工（传料轴：ZG1-ZG4）。
- 耦合：
 - 主轴鼓 A 一旋转，该主轴鼓的**所有**主主轴将和新的滑块组配合工作。
 - 主轴鼓 B 一旋转，该主轴鼓的**所有**副主轴和所有传料轴将和新的滑块组配合工作。
 - 主轴鼓 A 和 B 的旋转为自控。
 - 主轴鼓 A 和 B 的旋转范围在 270 Γραδ 内。
（供电电缆的活动和扭转范围）

概念：位置

主主轴 HS_i 和副主轴 GS_i 及其滑块构成一个“位置”。

NCU 指定

一个“位置”中的进给轴和主轴总是指定给一个 NCU 控制（本示例中的方案）。其中一个作为主 NCU 工作的 NCU 还要控制主轴鼓 A 和 B 的旋转。因此一共需要 4 个 NCU，可管理的最大轴数如下：

轴数

必须为每个 NCU_i 配置下列进给轴/主轴：

滑块 1: X_{i1}, Z_{i1}

2: X_{i2}, Z_{i2}

主轴: HS_i, GS_i, 动力刀具: S1, S2

2.3 示例

传料轴: ZG_i

进料轴: STN_i 。

除上述轴外，还须为主 NCU 另外配置两根用于旋转主轴鼓 A 和 B 的轴。经过计算后表明，用于 4 个“位置”的总轴数无法通过一个 NCU 配置。（最多 31 根轴，需要使用的为 $4 + 10 + 2$ 根）。

轴容器

主轴鼓 A/B 一旋转， HS_i 、 GS_i 、 ZG_i 和 STN_i 便需由另一个 NCU 管理，因此必须将这些轴配置为轴容器中的 Link 轴。

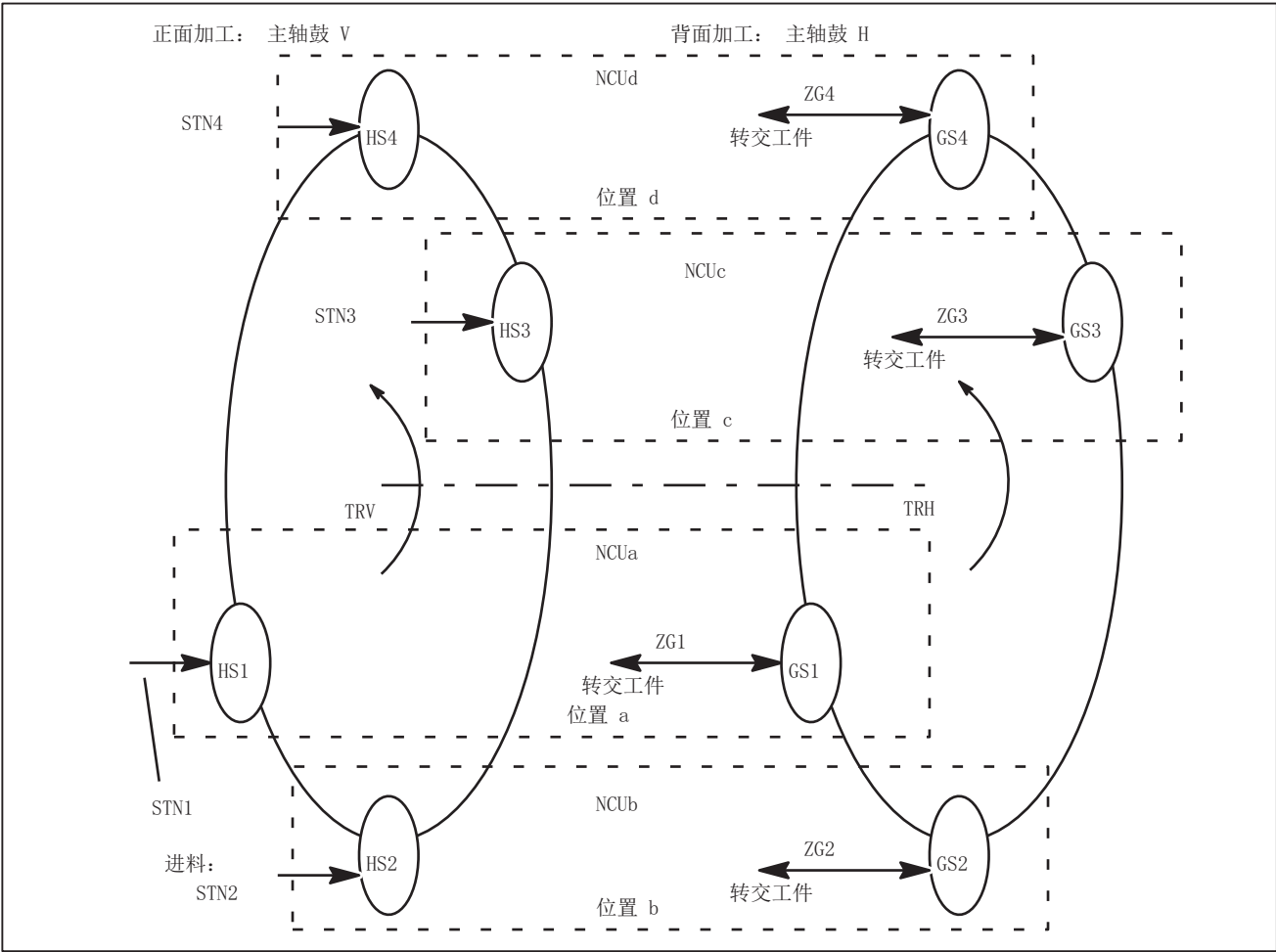


图 2-21 主轴 HS_i 、副主轴 GS_i 、进料轴 STN_i 和传料轴 ZG_i (图示)

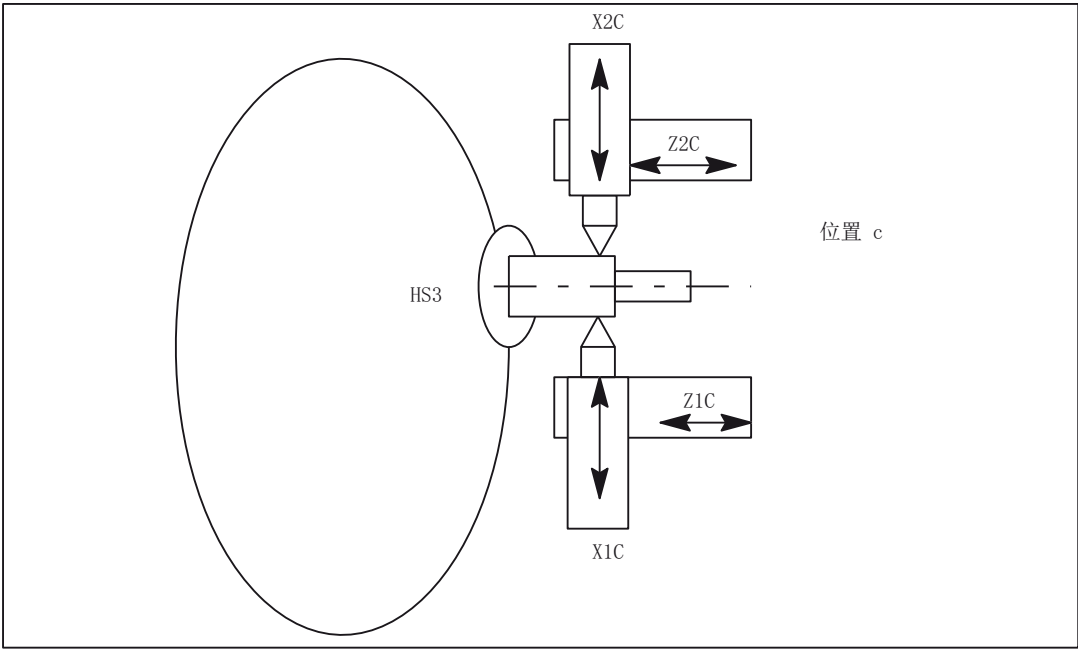


图 2-22 每个位置的两个滑块也可在同一根主轴上工作。

说明

为了清楚地显示轴和滑块、位置之间的关系，轴采用下列命名：

Xij，其中 i 指明滑块（1，2），j 指明位置（A-D）

Zij，其中 i 指明滑块（1，2），j 指明位置（A-D）

位置及滑块保持固定，而主主轴、副主轴、进料轴 **STN** 和传料轴 **ZG** 则会随着主轴鼓 **V** 或 **H** 的旋转运动到新位置。

根据上面的图示，将滑块考虑在内时，每个 **NCU** 须管理下列轴（举例）：

主 **NCU** 的轴

表格 2-1 主 **NCU** 的轴：NCUa

共用轴	本地轴	注释
	TRV（主轴鼓 V）	仅针对主 NCU
	TRH（主轴鼓 H）	仅针对主 NCU
	X1A	滑块 1

2.3 示例

共用轴	本地轴	注释
	Z1A	滑块 1
	X2A	滑块 2
	Z2A	滑块 2
	S1	滑块 1
	S2	滑块 2
HS1		轴容器必需
GS1		轴容器必需
ZG1		轴容器必需
STN1		轴容器必需
4	8	

NCUb 至 NCUd 的轴

除了用于旋转主轴鼓 TRV 和 TRH 的轴以外，其他 NCU 的轴与主 NCU 的相同。代表位置的字母须在 NCU 和轴名称中进行相应替换（a、A → b、B 至 d、D）。

配置规则

下文给出的配置以将下列规则纳入考量：

- 根据上面的“主主轴...”一图，主轴鼓旋转时指定给不同 NCU 管理的主主轴、副主轴和各进给轴必须配置在一个轴容器中。
(HS_i、GS_i、ZG_i、STN_i)。
- 主轴鼓 A 的所有主主轴位于同一容器中（1 号轴容器）。
- 主轴鼓 A 的所有进料轴位于同一容器中（2 号轴容器）。
- 主轴鼓 B 的所有副主轴位于同一容器中（3 号轴容器）。
- 主轴鼓 B 的所有传料轴位于同一容器中（4 号轴容器）。
- 为了确保 NCU 负载的均匀分布，主主轴 HS_i、副主轴 GS_i、副主轴的传料轴 ZG_i 和进主主轴的进料轴 STN_i 如下指定：
NCUa HS1-STN1，
NCUb HS2-STN2，依此类推。
- 滑块轴 X_{ij}、Z_{ij} 为纯粹的本地轴，由固定 NCU 管理。
- 滑块各占用 NCU 的一个单独通道，
从而可自控运动。

配置方案

- 为主主轴或副主轴灵活指定滑块。
- 在每个位置上可单独确定主主轴和副主轴的转速。

特例：

在将完成正面加工的棒料从主轴鼓 V 中传送到主轴鼓 H 中进行背面加工时，主主轴和副主轴必须设为相同的转速（同步主轴耦合）。

若为了“支持”滑块 1 滑块 2 也参与了正面加工，则此时主主轴转速也会影响滑块 2。与之对应，滑块 1 也参与背面加工时，副主轴转速也会影响滑块 1。

小幅转速变化

在跨 NCU 的加工中，采集各个实际值会有一定延时，因此应避免剧烈的转速变化。参见“轴数据和信号”。

NCU1 的配置

零件程序中采用统一的通道轴名称：

S4: 主主轴

S3: 副主轴

X1: 横给轴

Z1: 纵轴

S1: 动力刀具

Z3: 传料轴

TRV: 主主轴鼓 V

TRH: 副主轴鼓 H

STN: 液压进料轴

加粗轴是跨通道取轴时轴的当前通道（本地通道）。

2.3 示例

表格 2-2 NCUa, 位置: a, 通道: 1, 滑块: 1

通道轴名称	..._MACH AX_USED	\$MN_ AXCONF_LOGIC_MA CHAX_TAB	容器, 槽 输入 (String)	机床轴名称
S4	1	AX1: CT1_SL1	1 1 NC1_AX1	HS1
S3	2	AX2: CT3_SL1	3 1 NC1_AX2	GS1
X1	3	AX3:		X1A
Z1	4	AX4:		Z1A
Z3	5	AX5: CT4_SL1	4 1 NC1_AX5	ZG1
<i>S1</i>	<i>6</i>	<i>AX6:</i>		<i>WZ1A</i>
STN	7	AX7: CT2_SL1	2 1 NC1_AX7	STN1
TRV	11	AX11:		TRV
TRH	12	AX12:		TRH
x2 *				
z2 *				

表格 2-3 NCUa, 位置: a, 通道: 2, 滑块: 2

通道轴名称	..._MACH AX_USED	\$MN_ AXCONF_LOGIC_MA CHAX_TAB	容器, 槽 输入 (String)	机床轴名称
S4	1	AX1: CT1_SL1	1 1 NC1_AX1	HS1
S3	2	AX2: CT3_SL1	3 1 NC1_AX2	GS1
Z3	5	AX5: CT4_SL1	4 1 NC1_AX5	ZG1
STN	7	AX7: CT2_SL1	2 1 NC1_AX7	STN1

通道轴名称	..._MACH AX_USED	\$MN_ AXCONF_LOGIC_MA CHAX_TAB	容器，槽 输入 (String)	机床轴名称
X2	8	AX8:		X2A
Z2	9	AX9:		Z2A
S1	10	AX10:		WZ2A
x1 *				
z1 *				

说明

* 一个“位置”中的轴位置和 4 轴加工所需的程序协调。

指定轴在哪个轴容器中的哪个槽上需采用“NC1_AX.”的形式，其中 NC1 = NCU 1。上表中 NCUa 表示 NC1_...，NCUb 表示 NC2_...，以此类推。

其他 NCU

在 NCUb 至 NCUd 上也须设定上表中的配置数据。此时须注意：

- 轴 TRA 和 TRB 仅针对 NCUa，通道 1。
- 其他 NCU 的容器编号保留之前为各轴设定的编号
- 对应的槽编号分别为：
NCUb → 2
NCUc → 3
NCUd → 4。
- 对应的机床轴名称分别为：
NCUb → HS2, GS2, ZG2, STN2
NCUc → HS3, GS3, ZG3, STN3
NCUd → HS4, GS4, ZG4, STN4.

轴容器

下面的表格中按照容器和槽的顺序显示表格 7-17 中的相关信息，其中列明了所有以同种方式配置的 NCU（从 NCUb 到 NCUd）的容器和槽数据，这些配置是在以下机床数据中完成的：

MD12701 \$MN_AXCT_AXCONF_ASSIGN_TAB1[槽]

2.3 示例

...

MD12716 \$MN_AXCT_AXCONF_ASSIGN_TAB16[槽]

其中必须设置槽 1-4，对应多主轴车床的 4 个位置：

说明

机床数据

\$MN_AXC_AXCONF_ASSIGN_TAB_i[槽]

必须设置表中“初始位置”下给出的值（不带逗号和机床轴名称）。

表格 2-4 主轴鼓 A 的轴容器以及和位置相关的配置

容器	槽	初始位置 (TRA 0°)	开关 1 (TRA 90°)	开关 2 (TRA 180°)	开关 3 (TRA 270°)	开关 4 (TRA 0°)
1	1	NC1_AX1, HS1	NC2_AX1, HS2	NC3_AX1, HS3	NC4_AX1, HS4	NC1_AX1, HS1
	2	NC2_AX1, HS2	NC3_AX1, HS3	NC4_AX1, HS4	NC1_AX1, HS1	NC2_AX1, HS2
	3	NC3_AX1, HS3	NC4_AX1, HS4	NC1_AX1, HS1	NC2_AX1, HS2	NC3_AX1, HS3
	4	NC4_AX1, HS4	NC1_AX1, HS1	NC2_AX1, HS2	NC3_AX1, HS3	NC4_AX1, HS4
2	1	NC1_AX7, STN1	NC2_AX7, STN2	NC3_AX7, STN3	NC4_AX7, STN4	NC1_AX7, STN1
	2	NC2_AX7, STN2	NC3_AX7, STN3	NC4_AX7, STN4	NC1_AX7, STN1	NC2_AX7, STN2
	3	NC3_AX7, STN3	NC4_AX7, STN4	NC1_AX7, STN1	NC2_AX7, STN2	NC3_AX7, STN3
	4	NC4_AX7, STN4	NC1_AX7, STN1	NC2_AX7, STN2	NC3_AX7, STN3	NC4_AX7, STN4
主轴鼓运动		0°	+ 90°	+ 90°	+ 90°	- 270°

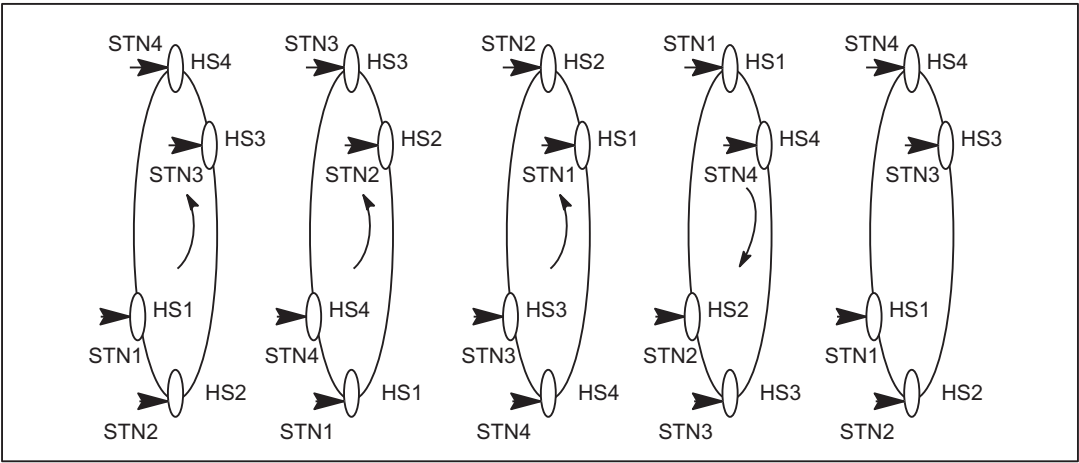


图 2-23 主轴鼓 A 的位置

表格 2-5 主轴鼓 B 的轴容器以及和位置相关的配置

容器	槽	初始位置 (TRB 0°)	开关 1 (TRB 90°)	开关 2 (TRB 180°)	开关 3 (TRB 270°)	开关 4 (TRB 0°)
3	1	NC1_AX2, GS1	NC2_AX2, GS2	NC3_AX2, GS3	NC4_AX2, GS4	NC1_AX2, GS1
	2	NC2_AX2, GS2	NC3_AX2, GS3	NC4_AX2, GS4	NC1_AX2, GS1	NC2_AX2, GS2
	3	NC3_AX2, GS3	NC4_AX2, GS4	NC1_AX2, GS1	NC2_AX2, GS2	NC3_AX2, GS3
	4	NC4_AX2, GS4	NC1_AX2, GS1	NC2_AX2, GS2	NC3_AX2, GS3	NC4_AX2, GS4
4	1	NC1_AX5, ZG1	NC2_AX5, ZG2	NC3_AX5, ZG3	NC4_AX5, ZG4	NC1_AX5, ZG1
	2	NC2_AX5, ZG2	NC3_AX5, ZG3	NC4_AX5, ZG4	NC1_AX5, ZG1	NC2_AX5, ZG2
	3	NC3_AX5, ZG3	NC4_AX5, ZG4	NC1_AX5, ZG1	NC2_AX5, ZG2	NC3_AX5, ZG3
	4	NC4_AX5, ZG4	NC1_AX5, ZG1	NC2_AX5, ZG2	NC3_AX5, ZG3	NC4_AX5, ZG4

2.3 示例

2.3.5 Lead-Link 轴

2.3.5.1 配置

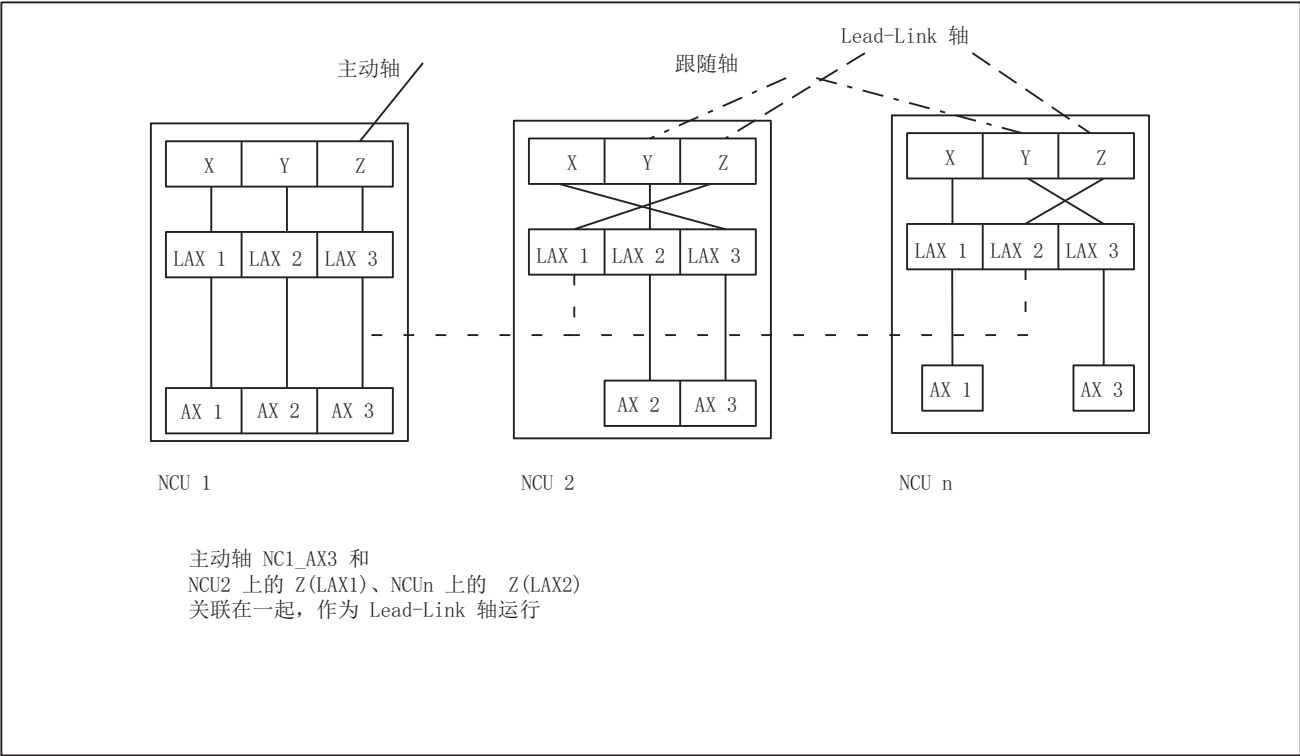


图 2-24 NCU2 至 NCUn 采用一根 Lead-Link 轴，用于实现与 NCU1 所连接的机床轴（NCU1-AX3）的耦合。

下面以一个耦合段为例进行说明，其中 Y 轴（LAX2，AX2）是 NCU2 上的从动轴，Z 轴（LAX3，NC1_AX3）是 Lead-Link 轴。

机床数据

- 只允许在本地 NCU 上载入主动轴的机床数据。相关机床数据会从该 NCU 传输至其他定义了 Lead-Link 轴的 NCU。
- 每采用一条 Lead-Link 轴，该 NCU 上可运行的最大轴数便减少一。

NCU1 的机床数据（主动轴）

机床数据	含义
\$MN_NCU_LINKNO = 1	NCU1 或主 NCU
\$MN_MM_NCU_LINK_MASK = 1	NCU-Link 生效
\$MN_MM_LINK_NUM_OF_MODULES= 2	Link 模块的数量
\$MN_MM_SERVO_FIFO_SIZE = 4	将插补器和位置控制环之间的数据缓存增大到 4
\$MN_AXCONF_LOGIC_MACHAX_TAB[0] = "AX1"	
\$MN_AXCONF_LOGIC_MACHAX_TAB[1] = "AX2"	
\$MN_AXCONF_LOGIC_MACHAX_TAB[2] = "AX3"	
\$MN_AXCONF_MACHAX_NAME_TAB[0] = "XM1"	
\$MN_AXCONF_MACHAX_NAME_TAB[2] = "YM1"	
\$MA_AXCONF_ASSIGN_MASTER_NCK[AX3] = 1	
\$MC_AXCONF_MACHAX_USED[0]=1 ; X	
\$MC_AXCONF_MACHAX_USED[1]=2 ; Y	
\$MC_AXCONF_MACHAX_USED[2]=3 ; Z	

NCU2 的机床数据（从动轴）

机床数据	含义
\$MN_NCU_LINKNO = 2	NCU2
\$MN_MM_NCU_LINK_MASK = 1	激活 Link
\$MN_MM_NUM_CURVE_TABS = 5	曲线表数量
\$MN_MM_LINK_NUM_OF_MODULES= 2	Link 模块的数量
\$MN_MM_NUM_CURVE_SEGMENTS= 50	
\$MN_MM_NUM_CURVE_POLYNOMS = 100	
\$MN_MM_SERVO_FIFO_SIZE = 2	插补器和位置控制环之间的数据缓存的大小（缺省值）
\$MN_AXCONF_LOGIC_MACHAX_TAB[0] = "NC1_AX3"	Lead-Link 轴在 NCU1/AX3 上
\$MN_AXCONF_LOGIC_MACHAX_TAB[1] = "AX2"	

2.3 示例

机床数据	含义
\$MN_AXCONF_LOGIC_MACHAX_TAB[2] = "AX3"	
\$MC_AXCONF_MACHAX_USED[0]=3	AX3 是第 1 通道轴的机床轴
\$MC_AXCONF_MACHAX_USED[1]=2	AX2 是第 2 通道轴的机床轴
\$MC_AXCONF_MACHAX_USED[2]=1	NCU1 上的 AX3 是第 3 通道轴的机床轴

2.3.5.2 编程

NCU1 的程序（主动轴）

NCU1 运行主动轴 Z。

向 NCU2 表明主动轴已被 NCU1 占用：Link 变量 \$A_DLB[0] = 1

向 NCU2 表明主动轴不再被 NCU1 占用：Link 变量 \$A_DLB[0] = 0

程序代码	注释
N1000 R1 = 0	; 初始化循环计数器
N1004 G1 Z0 F1000	; 将 z 轴运行至初始位置
N1005 \$A_DLB[0] = 1	; 向 NCU2 表明 z 轴已被占用
LOOP10:	;
N1005 R1=R1+1	; 提升循环计数器
N1006 Z0.01 G91	; 运行主动轴 z 轴
N1008 Z0.02	; 运行主动轴 z 轴
N1010 Z0.03	; 运行主动轴 z 轴
N1012 IF R1 < 10 GOTOB LOOP10	;
N1098 \$A_DLB[0] = 0	; 向 NCU2 表明 z 轴未被占用

NCU2 的程序（从动轴）

程序通过曲线表建立 NCU1 上的主动轴运动和 NCU2 上的从动轴运动之间的关联。若定义了表格，那么 NCU2 会进入等待状态（N2006），直至 NCU1 占用了主动轴 Z 轴（N1005）。一旦主动轴 Z 轴被占用，便立即激活耦合（N2010）。此耦合保持生效，直至 NCU1 不再占用主动轴 Z 轴。

程序代码	注释
N2000 CTABDEL(1)	; 初始化表 1
N2001 G04 F.1	;
N2003 G0 Y0 Z0	; 将 y 轴和 z 轴运行至初始位置
N2002 CTABDEF(Y, Z, 1, 0)	; 激活表格定义
N2003 G1 X0 Y0	; 控制点 1

程序代码	注释
N2004 G1 X100 Y200	; 控制点 2
N2005 CTABEND	; 取消表格定义
LOOP20:	;
N2006 IF (\$A_DLB[0] == 0) GOTOB LOOP20	; 等待 NCU1
N2010 LEADON(Y,Z,1)	; => 激活耦合
LOOP25:	;
N2030 IF (\$A_DLB[0] > 0) GOTOB LOOP25	; 保持耦合, 直至 NCU1 不再运行主动轴
N2090 LEADOF(Y,Z)	; => 撤销耦合

2.4 数据表

2.4.1 机床数据

2.4.1.1 通用机床数据

编号	名称: \$MN_	描述
10002	AXCONF_LOGIC_MACHAX_TAB	逻辑 NCU 机床轴映射
10065	POSCTRL_DESVAL_DELAY	位置设定值延时
10134	MM_NUM_MMC_UNITS	可同时支持的 MMC 通讯节点的数量
12510	NCU_LINKNO	NCU 组中的 NCU 编号
12520	LINK_TERMINATION	总线终端电阻为激活状态的 NCU 的编号
12530	LINK_NUM_OF_MODULES	Link 模块的数量
12701	AXCT_AXCONF_ASSIGN_TAB1	轴容器中的轴列表
...	...	
12716	AXCT_AXCONF_ASSIGN_TAB16	
12750	AXCT_NAME_TAB	轴容器名称列表
12760	AXCT_FUNCTION_MASK	轴容器专用功能
18700	MM_SIZEOF_LINKVAR_DATA	Link 变量存储器的大小
18720	MM_SERVO_FIFO_SIZE	插补器和位置控制器间的数据缓存的大小
18780	MM_NCU_LINK_MASK, 位 0	Link 通讯激活

2.4 数据表

2.4.1.2 通道专用机床数据

编号	名称: \$MC_	描述
20000	CHAN_NAME	通道名称
20070	AXCONF_MACHAX_USED	通道内有效的机床轴编号
28160	MM_NUM_LINKVAR_ELEMENTS	NCU-Link 变量的写单元数目

2.4.1.3 进给轴/主轴专用机床数据

编号	名称: \$MA_	描述
30550	AXCONF_ASSIGN_MASTER_CHAN	轴和通道的缺省指定
30554	AXCONF_ASSIGN_MASTER_NCU	通道中主主轴的缺省设置, 该 NCU 用于生成轴的设定值。
30560	IS_LOCAL_LINK_AXIS	轴是本地 Link 轴
32990	POCTRL_DESVAL_DELAY_INFO	当前位置设定值延时

2.4.2 设定数据

2.4.2.1 通用设定数据

编号	名称 \$SA	描述
41700	AXCT_SWWIDTH[容器编号]	轴容器旋转设定

2.4.2.2 进给轴/主轴专用设定数据

编号	名称: \$SA_	描述
43300	ASSIGN_FEED_PER_REV_SOURCE	定位轴/主轴旋转进给率

2.4.3 信号

2.4.3.1 从 NC 发出的信号

信号名称	SINUMERIK 840D sl
MCP1 就绪	DB10.DBX104.0
MCP2 就绪	DB10.DBX104.1
手动操作设备就绪	DB10.DBX104.2
NCU-Link 生效	DB10.DBX107.6
HMI2-CPU 就绪（操作面板或 MPI 上连接的 HMI）	DB10.DBX108.1
MPI 连接的 HMI1-CPU 就绪	DB10.DBX108.2
操作面板连接的 HMI1-CPU 就绪（标准连接）	DB10.DBX108.3

2.4.3.2 HMI/PLC 信号

信号名称	SINUMERIK 840D sl
ONL_REQUEST HMI 发出的在线请求	DB19.DBB100
ONL_CONFIRM PLC 对在线请求的应答	DB19.DBB102
PAR_CLIENT_IDENT HMI 写入客户端 ID（总线类型，HMI 总线地址）	DB19.DBB104
PAR_MMC_TYP HMI 类型，根据 NETNAMES.INI：主操作区域/副操作区域/报警服务器	DB19.DBB106
PAR_MSTT_ADR HMI 写入待激活 MCP 的地址	DB19.DBB107
PAR_STATUS PLC 写入对 HMI 的在线使能（连接状态）	DB19.DBB108

2.4 数据表

信号名称	SINUMERIK 840D sl
PAR_Z_INFO	DB19.DBB109
PLC 写入连接状态的附加信息	
M_TO_N_ALIVE	DB19.DBB110
从 PLC 发送至 HMI 的生命符号, 通过 M:N 模块	

2.4.3.3 通用在线接口

信号名称	SINUMERIK 840D sl
MMC1_CLIENT_IDENT	DB19.DBB120
HMI 进入在线模式时, PLC 将 PAR_CLIENT_IDENT 写入 MMCx_CLIENT_IDENT。	
MMC1_TYP	DB19.DBB122
HMI 进入在线模式时, PLC 将 PAR_MMC_TYP 写入 MMCx_TYP。	
MMC1_MSTT_ADR	DB19.DBB123
HMI 进入在线模式时, PLC 将 PAR_MSTT_ADR 写入 MMCx_MSTT_ADR。	
MMC1_STATUS	DB19.DBB124
连接状态, HMI 和 PLC 交替写入请求/应答。	
MMC1_Z_INFO	DB19.DBB125
连接状态的附加信息 (正/负应答、故障信息等)	
MMC1_SHIFT_LOCK	DB19.DBX126.0
HMI 切换禁止	
MMC1_MSTT_SHIFT_LOCK	DB19.DBX126.1
MCP 切换禁止	
MMC1_ACTIVE_REQ	DB19.DBX126.2
HMI 请求有效的操作模式	
MMC1_ACTIVE_PERM	DB19.DBX126.3
通过 PLC 使能操作模式切换	
MMC1_ACTIVE_CHANGED	DB19.DBX126.4
HMI 已切换操作模式	

信号名称	SINUMERIK 840D sl
MMC1_CHANGE_DENIED HMI 主动/被动模式切换被拒绝	DB19.DBX126.5
MMC2_CLIENT_IDENT HMI 进入在线模式时, PLC 将 PAR_CLIENT_IDENT 写入 MMCx_CLIENT_IDENT。	DB19.DBB130
MMC2_TYP HMI 进入在线模式时, PLC 将 PAR_MMC_TYP 写 入 MMCx_TYP。	DB19.DBB132
MMC2_MSTT_ADR HMI 进入在线模式时, PLC 将 PAR_MSTT_ADR 写入 MMCx_MSTT_ADR。	DB19.DBB133
MMC2_STATUS 连接状态, HMI 和 PLC 交替写入请求/应答。	DB19.DBB134
MMC2_Z_INFO 连接状态的附加信息 (正/负应答、故障信息等)	DB19.DBB135
MMC2_SHIFT_LOCK HMI 切换禁止	DB19.DBX136.0
MMC2_MSTT_SHIFT_LOCK MCP 切换禁止	DB19.DBX136.1
MMC2_ACTIVE_REQ HMI 请求有效的操作模式	DB19.DBX136.2
MMC2_ACTIVE_PERM 通过 PLC 使能操作模式切换	DB19.DBX136.3
MMC2_ACTIVE_CHANGED HMI 已切换操作模式	DB19.DBX136.4
MMC2_CHANGE_DENIED HMI 主动/被动模式切换被拒绝	DB19.DBX136.5

2.4 数据表

2.4.3.4 来自进给轴/主轴的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
NCU-Link 轴生效	DB31, ...DBX60.1	-
轴报警	DB31, ...DBX61.1	DB390x, DBX1.1
轴运行就绪	DB31, ...DBX61.2	DB390x, DBX1.2
轴容器旋转生效	DB31, ...DBX62.7	-

2.4.4 系统变量

系统变量	描述
\$AN_AXCTSWE[轴]	指明使能了轴容器旋转的指定轴的槽状态。
\$AN_LAI_AX_IS_AXCTAX	指明逻辑机床轴映射的容器轴（位域）
\$AN_LAI_AX_IS_LINKAX	指明逻辑机床轴映射的 Link 轴（位域）
\$AN_LAI_AX_IS_LEADLINKAX	指明逻辑机床轴映射的 Lead-Link 轴（位域）
\$AN_LAI_AX_TO_MACHAX[轴]	指明指定轴的 NCU-ID 和机床轴号。
\$AN_LAI_AX_TO_IPO_NC_CHANAX[轴]]	指明指定轴的通道号和通道轴号或 NCU 专用轴号和全局轴号。
\$AN_IPO_CHANAX[全局轴编号]	指明指定的全局轴号的通道号和通道轴号
\$AA_MACHAX[轴]	指明指定轴的 NCU-ID 和机床轴号
\$AA_IPO_NC_CHANAX[轴]	指明指定的全局轴号的通道号和通道轴号或 NCU-ID 和全局轴号
\$VA_IPO_NC_CHANAX[轴]	指明指定的机床轴的通道号和通道轴号或 NCU-ID 和全局轴号

对系统变量的详细说明请见：

文档：

参数手册“系统变量”

H1: 手动运行和手轮运行

3.1 引言

3.1.1 概述

即使是先进的数控机床，也需要具备由操作人员手动运行轴的功能：

- 调试机床
在新建了一个加工程序后，尤其需要通过机床操作面板上的移动键或电子手轮来移动机床轴。在选择了坐标平移或旋转时，也可在经过转换的工件坐标系中进行手动运行。
- 程序中中断后退刀
掉电或复位操作造成程序中中断后，操作人员必须手动将刀具从当前加工位置退出来。退刀通常需要在 **JOG** 运行方式（手动方式）中按下移动键来完成。此时须继续采用加工中生效的转换和坐标系。

功能

下列功能用于轴的手动运行：

- 连续运行，分点按模式和长按模式
- 断续运行，分点按模式和长按模式
- 通过电子手轮运行轴
- 在 **AUTO** 方式中将手轮命令叠加到原有的行程命令和/或速度命令上
- 在 **AUTO** 方式中使用手轮来设置附加的增量零点偏移，以补偿刀具磨损（DRF）
- 运行至通过机床数据设定的固定档快
- 掉电或复位操作造成程序中中断后使刀具沿其中心轴方向退回

3.1.2 JOG 运行方式的一般特性

下面将介绍在 **JOG** 方式中手动运行轴的一般特性（与所选择的具体子方式无关）。

3.1 引言

JOG 运行方式

下文所指的“手动运行”指的是在 JOG 运行方式下操作员通过机床控制面板上的移动键来手动移动各轴。

若在当前运行方式组中 JOG 运行模式生效，NC 会通过对应的接口信号向 PLC 报告这一状态：

DB11 DBX6.2 (BAG1: 运行方式 JOG)

DB11 DBX26.2 (BAG2: 运行方式 JOG)

...

DB11 DBX186.2 (BAG10: 运行方式 JOG)

文档：

功能手册之基本功能分册；章节“BAG、通道、程序运行、复位特性 (K1)”

机床功能

JOG 运行方式可细分为多种 JOG 子方式（即所谓的机床功能）：

- 连续运行 (JOG-CONT)
- 断续运行 (JOG-INC)
- 手轮运行

手轮运行

手轮还可以实现以下功能：

- 机床功能 JOG-REPOS，用于运行几何轴和机床轴
- 在 AUTO 运行方式中使轴移动 DRF 偏移量
- 行程叠加
- 往复运动的转向点偏移

生效的机床功能通过 PLC 接口选择。此时机床轴（轴特定）和几何轴（通道特定）分别有专用的 PLC 接口。

运行多根轴

在 JOG 运行方式下可同时运行多根轴。不过这些轴之间不进行插补。

速度

JOG 方式下的运行速度取决于进给率类型，通过下列参数定义：

- 直线进给率 (G94) 生效时 (SD41100 \$SN_JOG_REV_IS_ACTIVE = 0):
 - 通过通用设定数据定义：
SD41110 \$SN_JOG_SET_VELO (JOG 轴速度)
回转轴通过以下通用设定数据定义：
SD41130 \$SN_JOG_ROT_AX_SET_VELO
(回转轴 JOG 速度)
 - 或 (仅在 SD41110 = 0 时) 通过轴专用机床数据定义：
MD32020 \$MA_JOG_VELO (基准 JOG 轴速度)
- 旋转进给率 (G95) 生效时 (SD41100 \$SN_JOG_REV_IS_ACTIVE = 1):
 - 通过通用设定数据定义：
SD41120 \$SN_JOG_REV_SET_VELO (JOG 旋转进给率)
 - 或 (仅在 SD41120 = 0 时) 通过轴专用机床数据定义：
MD32050 \$MD_JOG_REV_VELO (JOG 旋转进给率)

进给速度的默认单位为 mm/min，而旋转进给率或回转轴为 rev/min。

说明

由于进给率受限，轴运行可能无法和手轮旋转同步，而是滞后于手轮，尤其是当手轮每转一刻度发出的脉冲数较多时。

快速移动叠加

若在按下移动键的同时还按下了快速移动键，则轴会以配置的快速移动速度运行：

- MD32010 \$MA_JOG_VELO_RAPID (基准 JOG 快速移动速度)
- MD32040 \$MA_JOG_REV_VELO_RAPID (快速移动速度，旋转进给率)

进给修调

若下列 NC/PLC 接口信号置位，则可通过进给修调开关对 JOG 运行速度进行调整：

DB31, ... DBX1.7 (进给修调开关生效)

进给修调开关各位置对应的百分比通过机床数据指定。若对应机床数据中输入了 0 值，则当开关转到位置 0 % 时不运行轴。

接口信号 DB31, ... DBX1.7 (进给修调生效) 对开关位置 0 % 无作用。

3.1 引言

也可选择通过 PLC 直接设定百分比值（0 % 至 200 %），以代替进给修调开关（格雷码）。此时同样通过机床数据进行选择。

文档:

功能手册之基本功能；章节“进给率 (V1)”

加速度

在手动运行和手轮运行中，轴同样按照设定的特性曲线加速。在 JOG 方式下对各轴生效的加速度特性曲线通过以下轴专用机床数据定义：

MD32420 \$MA_JOG_AND_POS_JERK_ENABLE（轴急动限制的缺省设置）

可为 JOG 方式下的手动运行设定独立的轴加速度限值和急动限值：

MD32301 \$MA_JOG_MAX_ACCEL（JOG 中的最大轴加速度）

MD32436 \$MA_JOG_MAX_JERK（JOG 中的最大轴急动度）

文档:

功能手册之基本功能分册；章节“加速度 (B2)”

显示

选择 JOG 运行方式时会显示 JOG 基本画面。此基本画面中包含了位置值、进给率值、主轴值和刀具值。

坐标系

在 JOG 运行方式下，操作人员可在不同的坐标系中运行轴。

可采用的坐标系如下：

- 基本坐标系
每根轴均可手动运行。
- 工件坐标系
只可手动运行几何轴（通道特定）。

几何轴

在手动运行时须区分，需要将相关轴作为机床轴（轴特定）运行还是作为几何轴（通道特定）运行。

下文首先介绍手动运行机床轴的特性。手动运行几何轴时的特殊之处请见“几何轴/定向轴的手动运行 (页 225)”。

主轴的手动运行

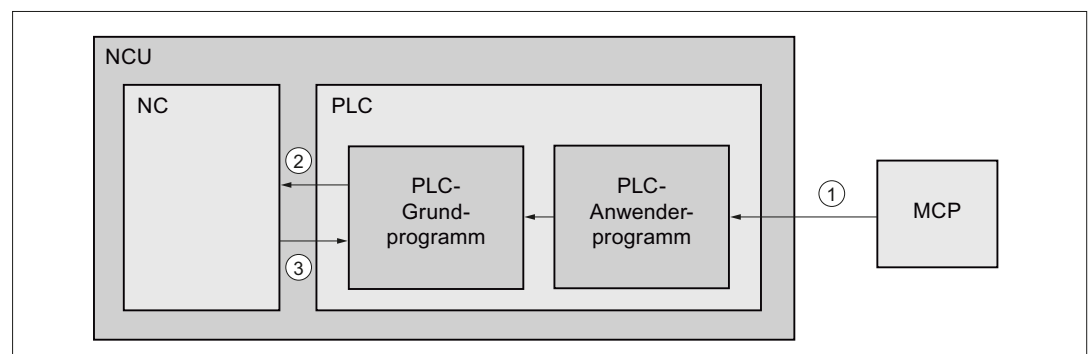
在 JOG 方式下也可手动运行主轴。运行条件与手动运行进给轴时基本相同。在 JOG 方式下，不仅可通过点按或者长按移动键来运行主轴，也可以通过手轮来运行主轴。与进给轴类似，此功能的选择和激活同样通过进给轴/主轴专用的 PLC 接口进行。进给轴专用机床数据也适用于主轴。手动运行主轴时的特殊之处请见“主轴的手动运行 (页 227)”。

3.1.3 通过 PLC 接口控制手动运行

在通过机床控制面板 (MCP) 选择手动运行功能时，NC 是通过 NC/PLC 接口激活该功能的。为此，必须将 MCP 的输入信号通过 PLC 传输至 NC/PLC 接口中 NC 的输入接口。这样一来，机床制造商便可通过 PLC 用户程序根据机床具体条件方便地调整手动运行功能。比如：机床制造商可修改 MCP 上的移动键和发送给 NC 的运行请求中的机床轴/几何轴的对应关系。

基本流程

下图显示了通过机床控制面板 (MCP) 为 NC 选择 JOG 运行方式的基本流程。



- ① 操作人员在机床控制面板上为某根机床轴选择机床功能“JOG-CONT”（手动连续模式）。

MCP 的输入信号从 PLC 基本程序循环传送到 MCP 输入接口中。

- ② PLC 用户程序读取 MCP 的输入信号。根据当前的机床或加工情况，可将输入信号与其他任意信号连在一起。最后，PLC 用户程序将发给 NC 的请求信号写入至对应的轴 NC/PLC 接口中。

PLC 基本程序将请求信号循环传输至内部对 NC 的输入接口。

- ③ 所请求的功能激活后，NC 将反馈信息写入内部对 PLC 的输出接口。PLC 基本程序将输出信号循环传输至对应的 NC/PLC 接口。

3.2 连续运行 (JOG-CONT)

文档

关于如何配置 MCP 以及如何将 MCP 加入 PLC 用户程序的详细信息请见功能手册之基本功能:

- SINUMERIK 840D sl: “P3: 用于 SINUMERIK 840D sl 的 PLC 基本程序”
- SINUMERIK 828D: “P4: 用于 SINUMERIK 828D 的 PLC”

3.2 连续运行 (JOG-CONT)

3.2.1 一般功能

选择

在 JOG 运行方式下的“连续运行”子方式必须通过 PLC 接口激活:

DB21, ... DBX13.6 (机床功能: 连续运行)

手轮叠加生效后, 以下接口信号会反馈给 PLC:

DB21, ... DBX41.6 (有效机床功能: 连续运行)

运行键

按下正/负移动键, 对应轴便会朝对应的方向运行。若同时按下两个移动键, 将不会产生运行动作, 或将使运行中的轴停止。

说明

在接通控制系统后, 轴可能会一直运行到机床限制区域, 因为尚未执行回参考点。此时可能触发硬件限位开关。软件限位开关和工作区限制不生效。

运行指令 +/-

一旦发出轴运行请求 (例如通过按下移动键), NC 便会根据运动方向向 PLC 输出接口信号

DB21, ... DBX40.7 (运行指令 +)

或者

DB21, ... DBX40.6 (运行指令 -)

3.2.2 点按模式/长按模式的区别

JOG 中的连续运行可分为点按模式和长按模式。

连续运行中的点按模式

采用点按模式时，按下移动键多长时间，轴就运行多长时间。松开移动键后轴将制动直至静止。达到设置的轴准停标准后，运动结束。

若松开移动键前轴已经到达某个限位（工作区域限制、软限位开关等），那么该轴将停在该限位上。

连续运行中的长按模式

采用长按模式时，按下移动键后轴就开始运行。松开移动键后，轴仍将继续运行。

 警告
碰撞危险 在长按模式中，按下相应的移动键一次可能会启动多根轴的运行。用户/机床制造商必须通过 PLC 用户程序实现必要的互锁。

操作人员可在任意时间点中断、继续执行或终止轴的运行。若松开移动键前轴已经到达某个限位（工作区域限制、软限位开关等），那么该轴将停在该限位上。

中断运行

操作人员可使用下列方法通过机床控制面板（MCP）的操作界面中断运行：

- 将进给倍率设为 0%
- 进给停止
- NC-Stop 或 NC-Stop 进给轴/主轴

中断原因消除后，轴将继续运行。

终止运行

操作人员可使用下列方法通过机床控制面板（MCP）的操作界面终止运行：

- 再次按下同一移动键
- 按下反向的移动键
- 复位
- 将运行方式切换成 AUTO 或 MDI，从而取消 JOG

3.3 断续运行 (JOG-INC)

下列情形下，控制系统会自动终止运行：

- 达到某个生效的限位（工作区域限制、软限位开关等）

 小心

工作区域限制无效

软限位开关和工作区域限制在轴回参考点后才生效。

- 触发含运行终止的报警

参数设置

点按/长按模式的选择针对一个 NC 中的所有轴生效，设定数据为：

SD41050 \$SN_JOG_CONT_MODE_LEVELTRIGGRD（JOG 连续运行中的点按/长按模式）

3.2.3 边界条件

分度轴

不论是采用点按模式还是长按模式，分度轴总是停止在一个分度位置。例如在点按模式中，松开移动键后分度轴将运行至运动方向上的下一个分度位置（参见章节“T1：分度轴 (页 811)”）。

3.3 断续运行 (JOG-INC)

3.3.1 一般功能

功能

在断续运行中，轴待运行的步数是由操作人员通过机床控制面板设定的。

除了五个固定步数（缺省设置：INC1、INC10、INC100、INC1000 和 INC10000），还有可由设定数据设置的可变步数 (INCvar)。

参数设置

固定步数

固定步数通过以下 NC 专用机床数据设置:

MD11330 \$MN_JOG_INCR_SIZE_TAB[1 ... 5] = <步数 1 ... 5>

可变步数

可变步数通过以下 NC 专用设定数据设置:

SD41010 \$SN_JOG_VAR_INCR_SIZE = <步数>

每一步的长度

固定步数和可变步数的每一步的长度通过以下轴专用机床数据设置:

MD31090 \$MA_JOG_INCR_WEIGHT = <行程>

说明

方向反转

输入负值时, 移动键的方向或手轮旋转方向会取反。

NC/PLC 接口

轴相关选择

DB31, ... DBX5.0 - 5.5 (机床功能: INC1 至 INCvar)

轴相关反馈信息

DB31, ... DBX65.0 - 65.5 (有效机床功能: INC1 至 INCvar)

3.3.2 点按模式/长按模式的区别

与连续运行类似的是: JOG 中的断续运行也可分为点按模式和长按模式。

断续运行中的点按模式

功能

按下某移动键(例如: 正向)后, 轴开始朝着对应的方向运行设置的规定步数。若在还没有走完规定步数前便松开了移动键, 运动将中断, 轴停止。再次按下同一移动键后, 轴将继续运行, 直至剩余行程为零。在此之前可再次松开移动键来中断运动。

3.3 断续运行 (JOG-INC)

只要还没有走完规定步数或运动已终止，反向的移动键就一直无效。

终止运行

如果步数未走完，则可按如下方式终止运行：

- 复位
- DB31, ... DBX2.2（删除剩余行程）

断续运行中的长按模式

功能

在该模式中，只要按下移动键一次（发出首个上升沿），轴将走完规定步数。若在走完规定步数前再次按下同一移动键（发出第二个上升沿），则会终止运行（即不再继续剩下的步数）。

中断运行

中断方式和手动连续运行一样。

终止运行

通过下列操作或监控停止和终止运行：

- 再次按下同一移动键
- 按下反向的移动键
- 复位
- DB31, ... DBX2.2（删除剩余行程）
- 达到首个生效的限位



小心

工作区域限制无效

软限位开关和工作区域限制在回参考点后才生效。

- 取消或修改当前步数（例如从 INC100 切换至 INC10）
- 出现故障（例如撤销了伺服使能）



警告

碰撞危险

在长按模式中，按下移动键一次可能会启动多根轴的运行。必要时须通过 PLC 实现互锁！

说明

轴运行期间，控制系统内部会禁止从 JOG 运行方式切换至 AUTO 或 MDI。

设置

点按/长按模式的选择针对一个 NC 中的所有轴生效，机床数据为：

MD11300 \$MN_JOG_INC_MODE_LEVELTRIGGRD (INC/REF 点按模式)

长按模式为缺省设置。

3.3.3 边界条件

分度轴

按下移动键后，分度轴将运行至运动方向上的下一个分度位置，不管设置的步数是多少（参见章节“T1: 分度轴 (页 811)”）。

3.4 JOG 下的手轮运行

3.4.1 功能

使用电子手轮（附件）可同时手动运行选择的多根轴。手轮每转一个刻度轴走完的行程用“步长”来确定。在选择了坐标平移或旋转时，也可在经过转换的工件坐标系中进行手动运行。

选择

JOG 运行方式必须生效。操作人员还须设置在手轮运行时生效的步数 INC1, INC10, ...。另外，还须在 PLC 接口上设置所需的机床功能，设置方式如同上文所述的断续运行。

3.4 JOG 下的手轮运行

运行

旋转电子手轮时，对应轴将根据手轮旋转方向朝正向或负向运行。

说明

若已通过移动键运行了轴，则无法再用手轮运行轴。

运行行程

手轮旋转所产生的运行行程取决于下列因素：

- 接口处接收到的手轮脉冲数
- 生效的步数（机床功能 INC1，INC10，INC100，... INCvar）
- 手轮每转一个刻度发出的脉冲数：
MD11320 \$MN_HANDWH_IMP_PER_LATCH（手轮每个刻度的脉冲数）
参见章节“设置 (页 173)”。
- 步长：
MD31090 \$MA_JOG_INCR_WEIGHT（手动断续运行/手轮运行中的步长）
参见章节“设置 (页 173)”。

手轮连接

系统上最多可同时连接 6 个手轮。因此同时最多可运行 6 根进给轴。

NC/PLC 接口信号中手轮号的表示方式

根据 MD11324 的设置，在 NC/PLC 接口信号中采用**位编码**（可显示 3 个手轮）或**二进制编码**（可显示 6 个手轮）表示手轮号。

参见“设置 (页 173)”。

手轮指定

可采用以下方式指定通过旋转手轮运行的轴：

- 通过 PLC 用户接口或
- 通过操作界面（HMI）。

手轮和 NC/PLC 接口的关联通过 PLC 用户程序进行。此时可为一个手轮同时指定多根轴。

通过 PLC 用户接口设置

通过下列接口信号中的一个来为手轮指定轴：

- 机床轴：
 - DB31, ... DBX4.0-2（激活手轮 <n> (1, 2, 3)）
- 几何轴：
 - DB21, ... DBX12.0-2（几何轴 1：激活手轮 <n>）
 - DB21, ... DBX16.0-2（几何轴 2：激活手轮 <n>）
 - DB21, ... DBX20.0-2（几何轴 3：激活手轮 <n>）
- 定向轴：
 - DB21, ... DBX320.0-2（定向轴 1：激活手轮 <n>）
 - DB21, ... DBX324.0-2（定向轴 2：激活手轮 <n>）
 - DB21, ... DBX328.0-2（定向轴 3：激活手轮 <n>）

通过操作界面（HMI）设置

按下 JOG 方式下主菜单中的软键“手轮”，显示“手轮”窗口。此处可为每个手轮指定一根轴、使能或禁用手轮。

说明

在连接的手轮超过 3 个，且在 NC/PLC 接口信号中采用二进制编码表示时，**无法**通过操作界面 (HMI) 为手轮指定轴。

3.4 JOG 下的手轮运行

通过 HMI 选择手轮

HMI 和 PLC 之间有一个独立的用户接口，用于通过操作界面激活手轮。由 PLC 基本程序提供的用于手轮 1、2 和 3 的接口包括以下信息：

- 由手轮运行的轴：
 - 轴编号（若选择手轮时选择了机床轴）：
DB10 DBX100.0-4（由手轮 1 运行的轴的编号）
DB10 DBX101.0-4（由手轮 2 运行的轴的编号）
DB10 DBX102.0-4（由手轮 3 运行的轴的编号）
 - 通道编号（若选择手轮时选择了几何轴）：
DB10 DBX97.0-3（由手轮 1 运行的通道的编号）
DB10 DBX98.0-3（由手轮 2 运行的通道的编号）
DB10 DBX99.0-3（由手轮 3 运行的通道的编号）
- 机床轴或几何轴的附加信息：
DB10 DBX100.7（手轮 1：机床轴）
DB10 DBX101.7（手轮 2：机床轴）
DB10 DBX102.7（手轮 3：机床轴）
- 手轮的使能或禁用信息：
DB10 DBX100.6（手轮 1 已选择）
DB10 DBX101.6（手轮 2 已选择）
DB10 DBX102.6（手轮 3 已选择）

通过 PLC 基本程序将指定轴的对应接口信号设置为“0”（禁用）或“1”（使能）：

- 机床轴：
 - DB31, ... DBX4.0-2（激活手轮 <n>）
- 几何轴：
 - DB21, ... DBX12.0-2（几何轴 1：激活手轮 <n>）
 - DB21, ... DBX16.0-2（几何轴 2：激活手轮 <n>）
 - DB21, ... DBX20.0-2（几何轴 3：激活手轮 <n>）

说明

定向轴只能通过对应的 PLC 用户接口信号激活：

- DB21, ... DBX320.0-2（定向轴 1：激活手轮 <n>）
 - DB21, ... DBX324.0-2（定向轴 2：激活手轮 <n>）
 - DB21, ... DBX328.0-2（定向轴 3：激活手轮 <n>）
-

运行请求

借助以下 NC/PLC 接口信号 PLC 可获知轴将要运行或正在运行：

- 机床轴：
 - DB31, ... DBX64.4（运行请求 -）或
 - DB31, ... DBX64.5（运行请求 +）
- 几何轴 1：
 - DB21, ... DBX40.4（几何轴 1：运行请求 -）或
 - DB21, ... DBX40.5（几何轴 1：运行请求 +）
- 几何轴 2：
 - DB21, ... DBX46.4（几何轴 2：运行请求 -）或
 - DB21, ... DBX46.5（几何轴 2：运行请求 +）
- 几何轴 3：
 - DB21, ... DBX52.4（几何轴 3：运行请求 -）或
 - DB21, ... DBX52.5（几何轴 3：运行请求 +）
- 定向轴 1：
 - DB21, ... DBX332.4（定向轴 1：运行请求 -）或
 - DB21, ... DBX332.5（定向轴 1：运行请求 +）
- 定向轴 2：
 - DB21, ... DBX336.4（定向轴 2：运行请求 -）或
 - DB21, ... DBX336.5（定向轴 2：运行请求 +）
- 定向轴 3：
 - DB21, ... DBX340.4（定向轴 3：运行请求 -）或
 - DB21, ... DBX340.5（定向轴 3：运行请求 +）

说明

运行请求是所有子运动的总和，即耦合分量和补偿值也会被考虑在内。

有关信号“运行请求”的生效方式参见“运行请求 (页 178)”一章。

3.4 JOG 下的手轮运行

运行指令

根据机床数据 MD11324 \$MN_HANDWH_VDI_REPRESENTATION 中的设置（参见章节“设置 (页 173)”），在出现运行请求后或在轴运行期间，以下接口信号便已经输出给 PLC:

- 机床轴:
 - DB31, ... DBX64.6（运行指令 -）或
 - DB31, ... DBX64.7（运行指令 +）
- 几何轴 1:
 - DB21, ... DBX40.6（几何轴 1: 运行指令 -）或
 - DB21, ... DBX40.7（几何轴 1: 运行指令 +）
- 几何轴 2:
 - DB21, ... DBX46.6（几何轴 2: 运行指令 -）或
 - DB21, ... DBX46.7（几何轴 2: 运行指令 +）
- 几何轴 3:
 - DB21, ... DBX52.6（几何轴 3: 运行指令 -）或
 - DB21, ... DBX52.7（几何轴 3: 运行指令 +）
- 定向轴 1:
 - DB21, ... DBX332.6（定向轴 1: 运行指令 -）或
 - DB21, ... DBX332.7（定向轴 1: 运行指令 +）
- 定向轴 2:
 - DB21, ... DBX336.6（定向轴 2: 运行指令 -）或
 - DB21, ... DBX336.7（定向轴 2: 运行指令 +）
- 定向轴 3:
 - DB21, ... DBX340.6（定向轴 3: 运行指令 -）或
 - DB21, ... DBX340.7（定向轴 3: 运行指令 +）

取反手轮旋转方向

手轮旋转方向与所需的轴运行方向不一致时，可取反手轮的旋转方向。在由一个手轮（HT2, HT8）运行多根轴时，尤其需要进行方向的调整。

除了设置机床数据外，也可通过设置各轴对应的接口信号“取反手轮旋转方向”来激活取反:

- 机床轴：
 - DB31, ... DBX7.0（取反手轮旋转方向）
- 几何轴：
 - DB21, ... DBX15.0（几何轴 1：取反手轮旋转方向）
 - DB21, ... DBX19.0（几何轴 2：取反手轮旋转方向）
 - DB21, ... DBX23.0（几何轴 3：取反手轮旋转方向）
- 定向轴：
 - DB21, ... DBX323.0（定向轴 1：取反手轮旋转方向）
 - DB21, ... DBX327.0（定向轴 2：取反手轮旋转方向）
 - DB21, ... DBX331.0（定向轴 3：取反手轮旋转方向）
- 轮廓手轮：
 - DB21, ... DBX31.5（取反轮廓手轮旋转方向）

说明

在 PLC 用户程序中，取反信号的设置应和手轮选择（接口“激活手轮”）同时进行。

NC 对手轮旋转方向取反的应答是通过各轴对应的接口信号“手轮旋转方向取反生效”发出的。

- 机床轴：
 - DB31, ... DBX67.0（手轮旋转方向取反生效）
- 几何轴：
 - DB21, ... DBX43.0（几何轴 1：手轮旋转方向取反生效）
 - DB21, ... DBX49.0（几何轴 2：手轮旋转方向取反生效）
 - DB21, ... DBX55.0（几何轴 3：手轮旋转方向取反生效）
- 定向轴：
 - DB21, ... DBX335.0（定向轴 1：手轮旋转方向取反生效）
 - DB21, ... DBX339.0（定向轴 2：手轮旋转方向取反生效）
 - DB21, ... DBX343.0（定向轴 3：手轮旋转方向取反生效）
- 轮廓手轮：
 - DB21, ... DBX39.5（轮廓手轮旋转方向取反生效）

说明

只允许在静止状态下修改取反信号。若在输出运行设定值期间由插补器修改了取反信号，系统会拒绝该修改并发出报警，根据当前的加速/减速值停止轴运行。

3.4 JOG 下的手轮运行

终止/中断运行

以下 NC/PLC 接口信号可使运行终止。给定值/实际值之差(即剩余行程)将被清除。

NC/PLC 接口信号	有效范围	结果	
DB21, ... DBX7.7 (NC 复位)	几何轴/机床轴	终止	
DB21, ... DBX12.0-2 (几何轴 1: 激活手轮 <n>)	几何轴	1 → 0	终止
DB21, ... DBX16.0-2 (几何轴 2: 激活手轮 <n>)			
DB21, ... DBX20.0-2 (几何轴 3: 激活手轮 <n>)			
DB21, ... DBX6.2 (删除剩余行程)	几何轴/机床轴	0 → 1	终止
DB31, ... DBX2.2 (删除剩余行程/主轴复位)	几何轴/机床轴	0 → 1	终止
DB31, ... DBX4.2 (激活手轮 <n>)	机床轴	1 → 0	终止
DB31, ... DBX1.3 (进给轴/主轴禁止)	几何轴/机床轴	0 → 1	终止
DB31, ... DBX1.5 (位置测量系统 1)	几何轴/机床轴	0 → 1	终止
DB31, ... DBX1.6 (位置测量系统 2)	几何轴/机床轴	0 → 1	终止

其他 NC/PLC 接口信号（停止信号）对手轮的作用（终止或中断运行）可进行设置（参见章节“设置 (页 173)”）。

用 NC 停止键仅能中断轴运行。设定值/实际值之差仍保留。按下 NC-START 后，轴将继续运行完剩余的行程。

限制

各种限位同样对手轮运行生效。

更多信息参见章节“监控 (页 229)”。

3.4.2 设置

行程或速度设定

通过手轮可设定行程或速度：

- 行程设定（缺省设置）
但轴会走完手轮设定的行程。脉冲不会丢失。
由于轴最大速度的限制，轴运行可能会滞后于手轮。
- 速度设定
通过手轮只设定所需的运行速度。一旦手轮停止，轴也立即停止。若在一个插补周期中无手轮脉冲，则会立即进行制动。这样便不会导致轴运行滞后于手轮。手轮脉冲不提供行程设定。

设定方式通过以下机床数据设置：

MD11346 \$MN_HANDWH_TRUE_DISTANCE（手轮工作方式：行程设定/速度设定）

手轮每转一个刻度发出的脉冲数

须确定每个手轮每转一个刻度发出的脉冲数：

MD11320 \$MN_HANDWH_IMP_PER_LATCH [<n>]（手轮每个刻度的脉冲数）

说明

输入负值时，手轮反向旋转。

步数

每个手轮脉冲下轴待运行的步数是由所选步数决定的。

除了五个固定步数（缺省设置：INC1、INC10、INC100、INC1000 和 INC10000），还有可由设定数据设置的可变步数 (INCvar)。

固定步数

固定步数通过以下 NC 专用机床数据设置：

MD11330 \$MN_JOG_INCR_SIZE_TAB[1 ... 5] = <步数 1 ... 5>

可变步数

可变步数通过以下 NC 专用设定数据设置：

SD41010 \$SN_JOG_VAR_INCR_SIZE = <步数>

3.4 JOG 下的手轮运行

每一步的长度

固定步数和可变步数的每一步的长度通过以下轴专用机床数据设置:

MD31090 \$MA_JOG_INCR_WEIGHT = <行程>

说明

输入负值时，手轮反向旋转。

步数的限制

机床操作人员可对所选择的步数进行限制:

- 机床轴通过以下轴专用机床数据设置:
MD32080 \$MA_HANDWH_MAX_INCR_SIZE (所选步数的限制)
- 几何轴通过以下通道专用机床数据设置:
MD20620 \$MC_HANDWH_GEOAX_MAX_INCR_SIZE (对几何轴手轮步数的限制)
- 定向轴通过以下通道专用机床数据设置:
MD20621 \$MC_HANDWH_ORIAX_MAX_INCR_SIZE (对定向轴手轮步数的限制)

NC/PLC 接口信号中手轮号的表示方式

NC/PLC 接口信号中手轮号的表示方式通过以下机床数据定义:

MD11324 \$MN_HANDWH_VDI_REPRESENTATION

值	含义
0	位编码式表示 (缺省设置) → 可表示 3 个手轮。
1	二进制编码式表示 → 可表示 6 个手轮。

输出 NC/PLC 接口信号 “运行指令 +” 或 “运行指令 -”

NC/PLC 接口信号 “运行指令 +” 或 “运行指令 -” 的输出特性是通过以下机床数据确定的:

MD17900 \$MN_VDI_FUNCTION_MASK

值	含义
0	出现一个运行请求时（缺省设置），系统就会输出 NC/PLC 接口信号“运行指令 +”或“运行指令 -”。
1	仅当轴发生实际运行时（即设定值输出到伺服上时），系统才会输出 NC/PLC 接口信号“运行指令 +”或“运行指令 -”。

速度

手轮运行中也会使用下列在 JOG 下生效的轴速度：

- SD41110 \$SN_JOG_SET_VELO（JOG 轴速度）
- SD41130 \$SN_JOG_ROT_AX_SET_VELO（JOG 回转轴速度）
- MD32020 \$MA_JOG_VELO（基准 JOG 轴速度）

由于进给率受限，轴运行可能无法和手轮旋转同步，而是滞后于手轮，尤其是当手轮每转一刻度发出的脉冲数较多时。

加速度

如手动运行部分（参见“JOG 运行方式的一般特性 (页 155)”）。

反向运行

反向运行（反向旋转手轮）时的特性可在以下机床数据中设置：

MD11310 \$MN_HANDWH_REVERSE（手轮反向的阈值）

值	含义
= 0	一旦反向转动手轮，系统就计算出轴行程，并且使轴尽可能快地运行到计算出的终点。 在此分两种情况：如果该终点位于轴在当前运行方向上可达到制动状态的点之前，则轴首先朝当前方向制动，然后朝相反方向运行到终点。其他情况下轴立即运行到计算出的终点。
> 0	如果手轮在相反方向上输出的脉冲数超出了机床数据中给定的脉冲数，系统就会尽快使轴制动，忽略插补结束前手轮后续输出的所有脉冲。。 即在轴静止（设定值侧）后才会重新运行。

3.4 JOG 下的手轮运行

达到软限位开关/工作区域限制的特性

在 JOG 方式下，轴达到第一个生效限位时停止运行，并发出相应的报警。

根据机床数据中的设置：

MD11310 \$MN_HANDWH_REVERSE（手轮反向的阈值）

，系统之后的特性如下（只要轴在设定值侧未达到终点）：

- 由手轮脉冲计算出的行程构成一个虚拟终点，用于后续计算。
如果该虚拟终点超出限位（例如超出 10 毫米），那么轴会首先朝反方向上移动 10 毫米，然后再开始朝规定方向移动。若需轴在达到限位后立即再次朝反方向移动，则可以通过删除剩余行程或撤销手轮来清除虚拟的剩余行程。
- 所有导致轴超过限位的手轮脉冲都会被忽略。手轮反向旋转会立即使轴反向移动，即：离开限位。

进给特性

在 JOG 运行方式下，进给轴/主轴的进给特性取决于以下设定数据中的设置：

SD41100 \$SN_JOG_REV_IS_ACTIVE（JOG：旋转进给率/直线进给率）

SD41100 \$SN_JOG_REV_IS_ACTIVE	
生效	进给轴/主轴总是视主主轴的情况而定采用以下旋转进给率： MD32050 \$MA_JOG_REV_VELO （JOG 旋转进给率） 或 MD32040 \$MA_JOG_REV_VELO_RAPID （JOG 旋转进给率叠加快速移动速度）。
未生效	进给轴/ 主轴的特性取决于设定数据： SD43300 \$SA_ASSIGN_FEED_PER_REV_SOURCE （定位轴/主轴的旋转进给率）
	若几何轴上有含旋转的有效框架命令，则该几何轴的特性取决于以下通道专用设定数据： SD42600 \$SC_JOG_FEED_PER_REV_SOURCE （JOG 旋转进给率调整）

NC/PLC 接口停止信号的作用

NC/PLC 接口停止信号对手轮的作用（终止或中断运行）可通过以下机床数据进行设置：

MD20624 \$MC_HANDWH_CHAN_STOP_COND（确定通道专用的手轮运行性能）

MD32084 \$MA_HANDWH_STOP_COND (手轮运行性能)

NC/PLC 接口信号	有效范围	MD20624 \$MC_HANDWH_CHAN_STOP_COND	
		位 == 0	位 == 1
DB11 DBX0.5 (BAG 停止)	几何轴/机床轴	中断, 至 NC 启动	终止
DB11 DBX0.6 (BAG 停止, 进给轴和主轴)	几何轴/机床轴	中断, 至 NC 启动	终止
DB21, ... DBX7.3 (NC 停止)	几何轴/机床轴	中断, 至 NC 启动	终止
DB21, ... DBX7.4 (NC 停止, 进给轴和主轴)	几何轴/机床轴	中断, 至 NC 启动	终止
DB21, ...DBX6.0 (进给禁用)	几何轴/机床轴	中断	终止
DB21, ... DBB4 (进给修调) DB31, ...DBX4.5 (快速移动速度叠加)	几何轴	倍率 == 0: 中断, 如果快速移动叠加未生效	倍率 == 0: 终止, 如果快速移动叠加未生效
DB21, ... DBB5 (快速移动修调) DB31, ...DBX4.5 (快速移动速度叠加)	几何轴	快速移动倍率 == 0: 中断, 如果快速移动叠加已生效	快速移动倍率 == 0: 终止, 如果快速移动叠加已生效
DB21, ... DBX12/16/20.3 (几何轴 1/2/3: 进给停止)	几何轴	中断	终止

NC/PLC 接口信号	有效范围	MD32084 \$MC_HANDWH_CHAN_STOP_COND	
		位 == 0	位 == 1
DB31, ... DBB0 (轴进给修调)	机床轴, 非主轴	倍率 == 0: 中断	倍率 == 0: 终止
DB31, ... DBB19 (主轴倍率)	主轴	倍率 == 0: 中断	倍率 == 0: 终止
DB31, ... DBX4.3 (进给停止/主轴停止)	几何轴/机床轴	中断	终止
DB31, ... DBX2.3 (正在进行夹紧)	几何轴/机床轴	无作用	终止

3.4 JOG 下的手轮运行

NC/PLC 接口信号	有效范围	MD32084 \$MC_HANDWH_CHAN_STOP_COND	
		位 == 0	位 == 1
DB31, ... DBX2.1 (伺服使能)	几何轴/机床轴	中断	终止
DB31, ... DBX21.7 (脉冲使能)	几何轴/机床轴	中断	终止

说明

在轴运行期间撤销“伺服使能”和“脉冲使能”信号会使轴快速停止并发出复位报警。

中断运行

出现停止指令时，系统会保存剩余行程并累加手轮脉冲。停止条件消除后，系统会继续运行算出的行程。

终止运行

出现停止指令时，系统会删除剩余行程且在停止条件消除前会忽略（即：不累加）手轮脉冲。

3.4.3 运行请求

下面几个示例阐明了 NC/PLC 接口信号“运行请求”的生效方式。

示例 1：设定行程的手轮运行，停止条件为非终止标准

若设定行程的手轮（MD11346 \$MN_HANDWH_TRUE_DISTANCE == 1 或 == 3）在运行中出现一条非终止标准的停止条件（参见 MD32084 \$MA_HANDWH_STOP_COND 或

MD20624 \$MC_HANDWH_CHAN_STOP_COND)，则按下图所示方式输出 NC/PLC 接口信号“运行请求”和“运行指令”。

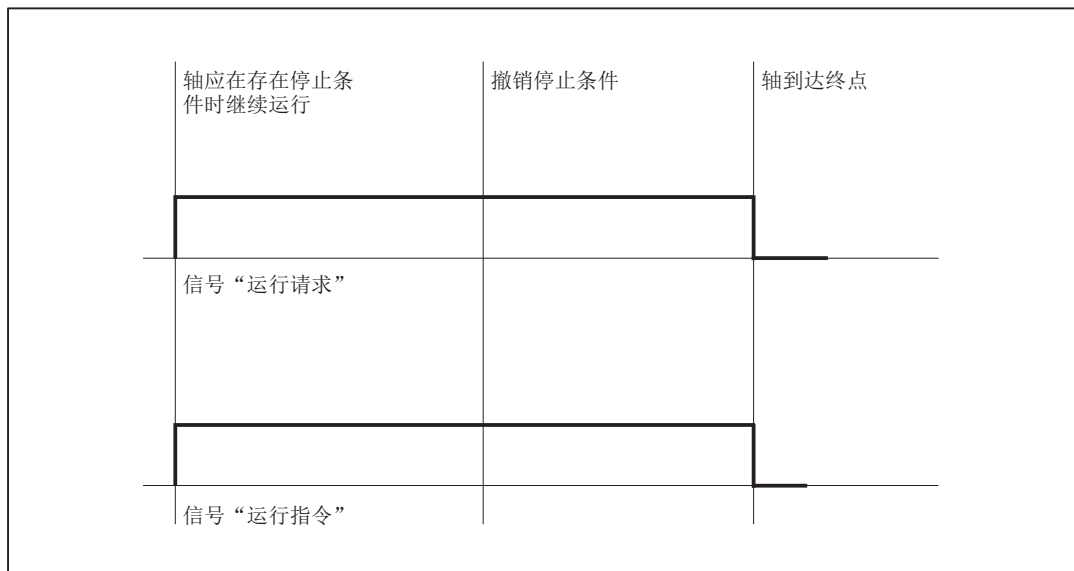


图 3-1 “信号-时间”图:设定行程的手轮运行，停止条件为非终止标准 MD17900
\$MN_VDI_FUNCTION_MASK 位 0 = 0

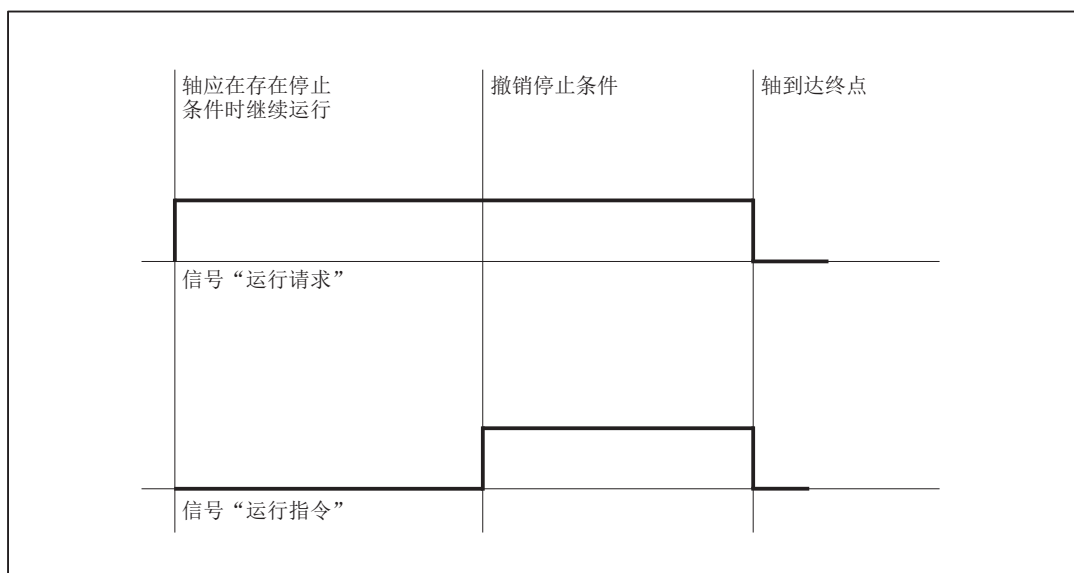


图 3-2 “信号-时间”图:设定行程的手轮运行，停止条件为非终止标准 MD17900
\$MN_VDI_FUNCTION_MASK 位 0 = 1

示例 2: 手轮运行，停止条件为终止标准

在手轮运行中，若出现一条由机床数据 MD32084 \$MA_HANDWH_STOP_COND 或 MD20624 \$MC_HANDWH_CHAN_STOP_COND 设为终止标准的停止条件，不输出运行指令，而是输出对应的运行请求。

停止条件消除后，轴运动终止，对应的 NC/PLC 接口信号“运行请求”复位。此时无停止条件生效，但停止条件先前造成了终止，因此无法运行轴。

此外，此时手轮可设定行程（MD11346 \$MN_HANDWH_TRUE_DISTANCE == 1 或 == 3）或连续旋转，即提供脉冲。

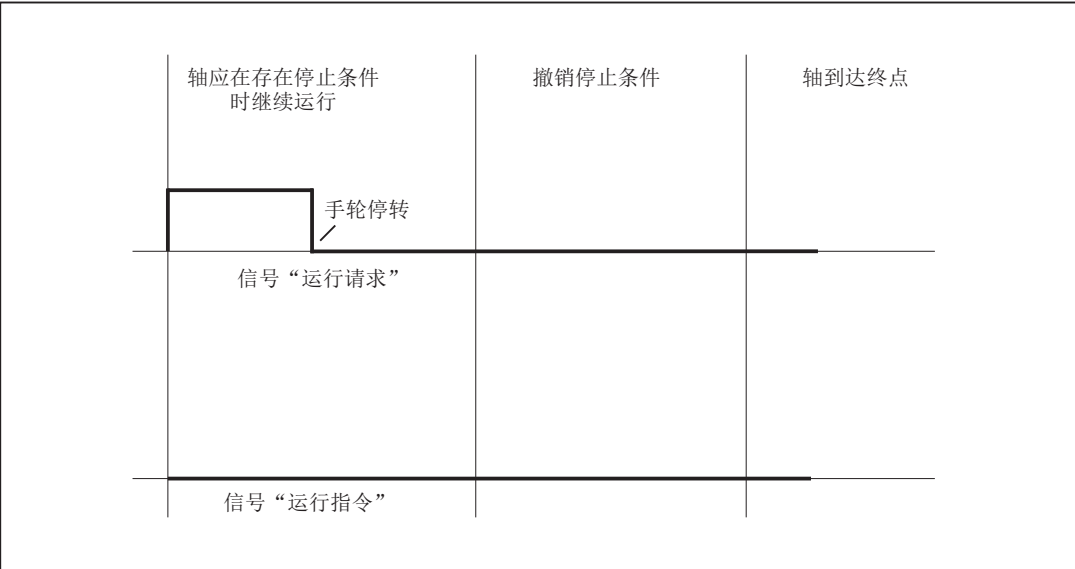


图 3-3 “信号-时间”图手轮运行，停止条件为终止标准

若在手轮运行期间激活了停止条件，运动会终止，NC/PLC 接口信号“运行请求”和“运行指令”复位。

示例 3: 设定速度的手轮运行，停止条件为终止标准

若设定速度的手轮（MD11346 \$MN_HANDWH_TRUE_DISTANCE == 0 或 == 2）不再旋转，则 NC/PLC 接口信号“运行请求”会复位。

同样，在取消手轮时 PLC 信号“运行请求”也会复位。

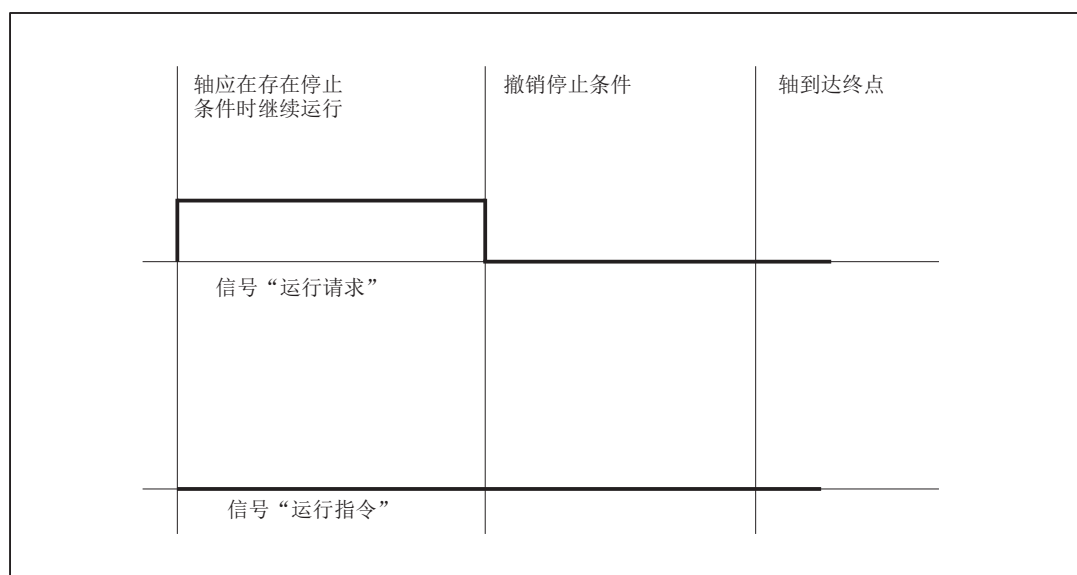


图 3-4 “信号-时间”图设定速度的手轮运行，停止条件为终止标准

边界条件

NC 停止

存在 NC 停止时不会输出运行指令，因此也不会输出运行请求。特例为 DRF 运行：若通过机床数据：MD20624 \$MC_HANDWH_CHAN_STOP_COND（位 13 == 1）在 NC 停止状态下使能了 DRF 运行，则其特性和手轮运行一样。

3.4.4 手轮的重复使用

报警 14320

若将同一个手轮（含轮廓手轮）同时指定用于 DRF 和用于给出叠加量（速度叠加、行程叠加），而该手轮可能对同一根轴造成不同影响，则系统会拒绝手轮的重复指定，并发出报警 14320“在通道 %3 轴 %4 上重复使用手轮 %1(%2)”指明这一错误。该报警会自行清除。

这表示，仅当基本坐标系中参与运行的轴上没有由同一个手轮触发的 DRF 生效时，系统才能执行手轮叠加，即此时 DRF 运动必须结束。

叠加运动开始后，参与运动的轴将无法由同一手轮执行 DRF。仅在该叠加运动达到终点或使用剩余行程删除或 RESET 终止后，该手轮才会执行 DRF。

若需使手轮叠加和 DRF 偏移同时生效，需要由两个单独的手轮来分别激活这两个功能。

示例：轨迹的行程叠加

假设：

通道 1 中的几何轴 X 对应机床轴 3；几何轴 Y 对应机床轴 5；为几何轴 1 选择了手轮 2。

若在主处理中执行程序段 X10 Y10 FD=0，那么无论是机床轴 3，还是机床轴 5 均不可通过手轮 2 执行 DRF。若在通道专用信号 DRF 生效时将手轮 2 指定给机床轴 3，则会发出报警 14320“在通道 1 轴 X 上重复使用手轮 2（8）”。

若使用手轮 2 对机床轴 3 或机床轴 5 执行 DRF，则无法执行 X10 Y10 FD=0，并会触发报警 14320“在通道 1 轴 X 上中重复使用手轮 2（3）”或报警 14320“在通道 1 轴 Y 上重复使用手轮 2（3）”。

示例：定位轴上的速度叠加

假设：

通道 1 中的通道轴 A 对应机床轴 4，且为该轴指定了手轮 1。

若在主处理中执行程序段 POS[A]=100 FDA[A]=0，则无法对机床轴 4 执行 DRF，也就是说，如果通道专用信号 DRF 生效，系统则会发出报警 14320“在通道 1 轴 A 上重复使用手轮 1（6）”。

若对机床轴 4 执行 DRF，那么在当前的 DRF 运动期间无法执行 POS[A]=100 FDA[A]=0。此时会发出报警 14320“在通道 1 轴 A 上重复使用手轮 1（1）”。

示例：PLC 轴上的行程叠加（840D sl）

假设：

通道 1 中的机床轴 4 指定了手轮 2。

若在主处理中执行了一个由 FC18 触发、包含机床轴 4 行程叠加的运动，则无法对机床轴 4 执行 DRF，也就是说，如果通道专用信号 DRF 生效，系统则会发出报警 14320“在通道 1 轴 A 上重复使用手轮 2（9）”。

若对机床轴 4 执行 DRF，那么在当前的 DRF 运动期间无法执行由 FC18 触发、含行程叠加的轴运动。此时会发出报警 14320“在通道 2 轴 A 上重复使用手轮 4（1）”。

3.5 自动方式下的手轮叠加

3.5.1 一般功能

功能

使用此功能可在自动方式（AUTO，MDI）中通过手轮直接运行轴以及修改轴速度。

手轮叠加在 NC 零件程序中通过 NC 语言指令 FD（用于轨迹轴）和 FDA（用于定位轴）激活，且为非模态生效。

在定位轴上还可通过运行指令 POSA 使手轮叠加在所在程序段结束后继续生效。在达到程序中写入的目标位置后，手轮叠加失效。

在同一个 NC 程序段中还可插补或并行运行其他轴。

此外还可通过 PLC 用户程序为并行定位轴激活此功能。

区分

根据程序写入的进给速度，自动方式下的手轮可用于：

- 行程设定
轴进给速度 = 0（FDA = 0）
- 速度叠加
轴进给速度 > 0（FD 或 FDA > 0）

下表中显示了该功能所适用的轴类型。

“自动方式下的手轮叠加”可调整的轴		
轴类型	速度叠加	行程设定
定位轴	FDA[AXi] > 0; 对单轴速度生效	FDA[AXi] = 0
并行定位轴	参数“手轮叠加有效”= 1，且 FC18 的单轴进给速度 > 0	参数“手轮叠加有效”= 1，且 FC18 的单轴进给速度 = 0
轨迹轴	FD > 0 ; 对轨迹速度生效	无法使用

行程设定

在单轴进给速度 = 0 时（例如编写了 $FDA[AXi] = 0$ ），只能由操作人员旋转对应手轮来将定位轴运行至程序写入的目标位置。

轴的运行方向取决于手轮的旋转方向。使用手轮叠加时，不可超出程序编写的目标位置。轴也可背离目标位置朝反向运行，此时运行只受限位的约束。

程序段在以下情形下切换：

- 轴到达了编写的目标位置

或者

- 通过轴接口信号：
DB31, ... DBX2.2（删除剩余行程）
删除了剩余行程。

行程设定从此时间点起失效，之后的手轮脉冲失效。

之后，编程的增量位置将相对于断点，而不是相对于最后编程的位置。

速度叠加

速度叠加分为两种类型：单轴速度的叠加和轨迹速度的叠加。

- **单轴速度叠加**（ $FDA[AXi] > 0$ ）：
定位轴运按编写的单轴进给速度将行至目标位置。随着手轮正转或反转，轴速度相应地提升或降低（最多可降低至零）。叠加后得出的单轴进给速度受最大速度限制。但是轴的运行方向不可与编写的目标方向相反。
轴一旦到达编写的目标位置，系统便进行程序段切换。速度叠加因此自动取消，之后的手轮脉冲失效。
对于通过 PLC 设定目标位置和速度的并行定位轴来说，速度叠加的方式与定位轴相似。
- **轨迹速度的叠加**（ $FD > 0$ ）：
NC 程序段中编写的轨迹轴按编写的轨迹速度运行至目标位置。手轮速度叠加生效时，该轨迹速度上将叠加由**第 1 几何轴的手轮**产生的速度。一旦轴到达编写的目标位置，便进行程序段切换。
随着手轮正转或反转，轨迹速度不断增大或下降，最多可降至零。但是无法通过手轮反转运动方向。

应用示例

“自动方式下的手轮叠加”功能常用于磨床。例如：操作人员使用手轮将来回摆动的砂轮移动到工件上（行程设定）。到达后运行结束，并触发程序段切换（通过激活 DB31, ... DBX2.2（删除剩余行程））。

前提条件

激活“自动方式下的手轮叠加”前须满足下列前提条件：

- 必须为轴指定了手轮
- 为对应手轮指定了每转脉冲数

手轮指定

与“JOG 下的手轮运行 (页 165)”类似，可通过操作界面为轴指定手轮或通过 PLC 接口信号指定：

- 机床轴：
 - DB31, ... DBX4.0-2（激活手轮（1, 2, 3））
- 几何轴：
 - DB21, ... DBX12.0-2（几何轴 1：激活手轮（1, 2, 3））
 - DB21, ... DBX16.0-2（几何轴 2：激活手轮（1, 2, 3））
 - DB21, ... DBX20.0-2（几何轴 3：激活手轮（1, 2, 3））

如果为没有指定手轮的轴编写了手轮叠加，在会出现两种结果：

- **速度叠加中：**

轴仍旧以编写的进给速度运行。
系统会发出一个自删除报警（无响应）。
- **行程设定中：**

由于速度为零，因此轴不会运行。
系统会发出一个自删除报警（无响应）。

说明

在针对轨迹轴的速度叠加中，只有**几何轴 1 的手轮**作用于轨迹速度。

手轮权重

手轮旋转一个刻度所产生的轴行程取决于多种因素（参见章节“JOG 下的手轮运行 (页 165)”）：

- 选择的步数：
MD11330 \$MN_JOG_INCR_SIZE_TAB[5]（手动断续运行/手轮运行中的步数）
或
SD41010 \$SN_JOG_VAR_INCR_SIZE（JOG 方式下的可变步数）
- 步长：
MD31090 \$MA_JOG_INCR_WEIGHT
- 手轮每转一个刻度发出的脉冲数：
MD11320 \$MN_HANDWH_IMP_PER_LATCH

例如：在上述机床数据采用缺省设置时，若选择机床功能 INC1，那么手轮每转一个刻度对应的轴行程为 0.001 mm。

在速度叠加中，速度由手轮在一段时间内产生的行程换算得出。

示例

假设：

操作人员以 100 脉冲/秒的速度旋转手轮。

选择的机床功能为 INC100。

上述用于手轮权重的机床数据采用缺省设置。

- ⇒ 每秒的手轮运行行程：10 mm
- ⇒ 速度叠加：0.6 m/min

PLC 接口信号

手轮叠加生效后，下列发送至 PLC 的接口信号立即置为 1 信号：

- 针对定位轴/并行定位轴/指令轴/往复轴：
DB31, ... DBX62.1（手轮叠加生效）
- 针对轨迹轴：
DB21, ... DBX33.3（手轮叠加生效）

在行程设定中，对应的接口信号根据运行方向输出至 PLC：

- 机床轴:
 - DB31, ... DBX64.6/7 (运行指令 -/+)
- 几何轴:
 - DB21, ... DBX40.6/7 (几何轴 1: 运行指令 -/+)
 - DB21, ... DBX46.6/7 (几何轴 2: 运行指令 -/+)
 - DB21, ... DBX52.6/7 (几何轴 3: 运行指令 -/+)

限制

在手轮叠加中，各种轴限位（软件限位开关、硬件限位开关、工作区域限制）都生效。在手轮设定行程时，手轮可使轴朝着编写的运行方向运行到目标位置。

由此得出的速度受轴机床数据的限制：

MD32000 \$MA_MAX_AX_VELO (最大轴速度)

NC Stop/倍率 = 0

手轮叠加生效时，若将进给倍率设为 0 % 或按下了“CYCLE STOP”，那么

- 行程设定中：

累加和保存在此期间发出的手轮脉冲。按下“CYCLE START”或进给倍率设为大于 0 % 的值后，保存的手轮脉冲将生效（即：使轴运行）。
若此前将手轮切换为无效，比如：通过接口信号 DB21, ... DBX12/16/20.0-2“几何轴 1/2/3: 激活手轮（1, 2, 3）”，则会删除保存的手轮脉冲。
- 在速度叠加中：

不累加在此期间发出的手轮脉冲，手轮脉冲因此无效。

3.5.2 手轮叠加的编程和激活

简介

在通过 **NC** 语言指令 **FD**（用于轨迹轴）和 **FDA**（用于定位轴）编程手轮叠加时，须注意以下几点：

- **FDA** 及 **FD** 为**非模态**生效。
针对定位轴的特例情形：编程运行指令 **POSA** 时，手轮叠加可超出程序段生效，因为定位轴不会影响程序段切换。
- 通过 **FDA** 或 **FD** 激活手轮叠加时，必须在 **NC** 程序段中为定位轴或轨迹轴编写一个**目标位置**。在达到程序中写入的目标位置后，手轮叠加失效。
- 在同一 **NC** 程序段中不可同时编写 **FDA** 和 **FD**，或者同时编写 **FA** 和 **F**。
- 定位轴不允许为分度轴。

定位轴

手轮叠加句法：**FDA**[AXi]=[进给速度值]

示例 1:

激活速度叠加

```
N10 POS[U]=10 FDA[U]=100 POSA[V]=20 FDA[V]=150 ...
```

POS [U]=10	定位轴 U 的目标位置
FDA [U]=100	为定位轴 U 激活速度叠加； U 轴速度为 100 mm/min
POSA [V]=20	定位轴 V 的目标位置（超出程序段生效）
FDA [V]=150	为定位轴 V 激活速度叠加； V 轴速度为 150 mm/min

示例 2:

在同一个 **NC** 程序段中激活行程叠加和速度叠加

```
N20 POS[U]=100 FDA[U]= 0 POS[V]=200 FDA[V]=150 ...
```

POS [U]=100	定位轴 U 的目标位置
FDA [U]=0	为定位轴 U 激活行程叠加；

POS [V]=200 定位轴 V 的目标位置
 FDA [V]=150 为定位轴 V 激活速度叠加；V 轴速度为 150 mm/min

轨迹轴

手轮叠加句法: $FD = [\text{进给速度值}]$

为轨迹轴编程“自动方式下的手轮叠加”时，需满足下列前提条件：

- G 功能组 1 的运动指令生效：G01，G02，G03，CIP
- 准停生效（G60）
- 线性进给率生效，单位为 mm/min 或 inch/min（G94）

控制系统会检查这些前提条件，若不符合将会发出报警。

示例 3:

激活速度叠加

```
N10 G01 X10 Y100 Z200 FD=1500 . . .
```

X10 Y100 Z200 轨迹轴 X、Y 和 Z 的目标位置
 FD=1500 为轨迹轴激活速度叠加，轨迹速度为 1500 mm/min

并行定位轴

并行定位轴的手轮叠加由 PLC 通过 FC18 置位对应的接口信号来激活：

DB31, ... DBX62.1（手轮叠加生效）

若速度参数值 0（F_value）为，那么激活的手轮叠加将作为行程叠加生效，即此情形下不通过轴机床数据计算出进给速度（另见“P2: 定位轴 (页 639)”章节）：

MD32060 \$MA_POS_AX_VELO（定位轴速度的初始设置）

文档：

功能手册之基本功能分册；章节“PLC 基本程序（P3）”

3.5.3 自动方式下的手轮叠加的特殊性

速度显示

使用手轮叠加时，速度显示区域会显示下列数值：

- **设定速度**
= 编写的速度
- **实际速度**
= 加入手轮叠加后的得出的速度

应用在车床 X 轴上

若轴被定义为车床 X 轴且 DIAMON 生效，那么在手轮叠加生效时，手轮脉冲被换算为直径坐标位置。

空运行进给率

空运行生效时，

即 DB21, ... DBX0.6（激活空运行进给率）= 1 时，

空运行进给率：

SD42100 \$SC_DRY_RUN_FEED 总是生效。

因此，即便是在手轮行程叠加（FDA[AXi] = 0）生效的情况下，轴仍会按空运行进给率运行至编写的目标位置，而不受手轮影响（即行程设定无效）。

DRF 生效

“自动方式下的手轮叠加”激活时，必须检查“DRF”功能是否生效（DB21, ... DBX0.3 = 1）。

若该功能激活，那么手轮脉冲也将影响 DRF 偏移。因此操作人员应事先取消 DRF 功能。

进给倍率

进给倍率对手轮生成的速度无效（特例情形：0%）。该倍率只作用于程序编写的进给速度。

若激活了手轮的行程设定，而手轮快速旋转时，轴运行可能无法和手轮旋转同步，而是滞后于手轮，尤其是当手轮每转一刻度发出的脉冲数较多时。

3.6 轮廓手轮/通过手轮设定行程（选件）

功能

此功能激活时，可在 AUTO 和 MDI 运行方式下通过手轮控制轨迹轴和同步轴的进给速度。

可用性

“轮廓手轮”在 SINUMERIK 840D sl 和 SINUMERIK 828D 系统中作为需要授权的选件提供。

设定方式（行程设定或速度设定）

通过手轮可设定行程或速度：

- **行程设定**

由于轴最大速度的限制，轴运行可能会滞后于手轮。但轴会走完手轮设定的行程，**脉冲不会丢失**。

- **速度设定**

通过手轮只设定所需的运行速度。一旦手轮停止，轴也立即停止。若在一个插补周期中无手轮脉冲，则会立即进行制动。这样便**不会导致轴运行滞后于手轮**。。手轮脉冲不提供行程设定。

设定方式通过以下机床数据设置：

MD11346 \$MN_HANDWH_TRUE_DISTANCE（手轮工作方式：行程设定/速度设定）

进给率

进给速度单位为 mm/min，大小**取决于**下列因素：

- 所选手轮在一段时间内发出的脉冲数
- 手轮每转一个刻度发出的脉冲数通过以下机床数据设置：
MD11322 \$MN_CONTOURHANDWH_IMP_PER_LATCH（轮廓手轮每转一个刻度发出的脉冲数）
- 选中的步数（INC1, 10, 100, ...）
- 可用第一几何轴的步长：
MD31090 \$MA_JOG_INCR_WEIGHT（手动断续运行/手轮运行中的步长）

进给速度的大小与下列因素**无关**：

3.6 轮廓手轮/通过手轮设定行程（选件）

- 程序中编写的进给速度单位（mm/min，mm/rev）
- 程序中编写的进给速度（叠加后得出的速度可能更高）
- G0 快速移动速度
- 倍率（仅 0 % 会影响速度设定，即轴静止）

运行方向

运行方向取决于旋转方向：

- **顺时针方向**
→ 朝程序中编写的方向运行。
达到程序段切换标准（插补结束）时，才切换至下一程序段（特性类似 G60）。
- **逆时针方向**
→ 朝程序中编写的方向运行。
此时最多只能运行至程序段开始处。手轮继续旋转时脉冲不会累加。

激活功能

此功能可通过接口信号或 NC 程序激活：

- 通过接口信号激活
通过接口信号激活/取消功能：
DB21, ... DBX30.0-2（激活轮廓手轮（1, 2, 3））
- 通过 NC 程序激活
可在 NC 程序中编写 FD=0 非模态式激活轮廓手轮，也就是说，轮廓手轮前程序段中的进给速度 F... 在轮廓手轮后的程序段中继续有效，**无需**重新编程。

说明

若轮廓手轮前的程序段中未编写进给速度，系统会发出相应报警。
不允许在一个程序段中同时编写 FD 和 F，否则系统会发出报警。

轮廓手轮的仿真

轮廓手轮激活时，也可对其进行仿真。

接口信号

DB21, ... DBX30.3（激活轮廓手轮仿真）激活后，
进给速度不再由轮廓手轮给定，而是采用程序中写入的进给速度。

方向也通过接口信号设定：

DB21, ... DBX30.4 (轮廓手轮仿真负方向)

仿真取消或方向切换时，轴会沿制动斜坡制动。

说明

倍率像执行 NC 程序时一样生效。

边界条件

- **前提条件**
不允许选择固定进给率、空运行进给率、螺纹切削或攻丝。
- **限值**
轴速度和加速度受机床数据中定义的限值的约束。
- **运行中断**
按下“CYCLE STOP”时，此功能保持激活，但是手轮脉冲不会累加，因此变为无效。
前提条件：MD32084 \$MA_HANDWH_CHAN_STOP_COND 位 2 = 1
- **DRF**
选中的 DRF 会设定行程叠加。
- **针对特定通道的剩余行程删除**
此操作会使通过轮廓手轮触发的运行中断，轴制动，程序从下一个程序段重新启动。之后轮廓手轮重新生效。

3.7 DRF 偏移

功能

通过“DRF 偏移”（Differential Resolver Function，手轮干预功能），可在 AUTO 运行方式下通过电子手轮在基本坐标系中设置几何轴和辅助轴的附加增量零偏。

推导出 DRF 偏移的手轮和由手轮移动的几何轴或辅助轴之间的对应关系必须借助机床轴来定义。机床轴指的是几何轴或辅助轴对应的机床上的真实存在的轴。

DRF 偏移不显示在轴的实际值显示区中。

3.7 DRF 偏移

应用

DRF 偏移例如可用于以下应用：

- 在一个 NC 程序段内进行刀具磨损补偿
对于加工时间非常长的 NC 程序段来说，有必要在一个程序段内对刀具磨损执行手动补偿（例如大型的平面铣床）。
- 磨削时的精补
- 简单的温度补偿（热误差补偿）

说明

通过 DRF 偏移产生的零偏在所有运行方式中均生效，且在 RESET 后继续生效。但也可程序中非模态式抑制该偏移。

减速

DRF 激活时，可与 JOG 速度成比例降低通过手轮产生的速度。

MD32090 \$MA_HANDWH_VELO_OVERLAY_FACTOR（JOG 速度与手轮速度（DRF）之比）

DRF 生效

为了能够通过手轮修改 DRF 偏移，DRF 必须生效。为此须满足下列前提条件：

- AUTO 运行方式
- DB21, ... DBX0.3（激活 DRF）= 1

通过 HMI 操作界面的“程序控制”功能可针对某个通道激活或取消 DRF 偏移。

HMI 软件之后会置位接口信号：

DB21, ... DBX24.3（DRF 已选择）= 1

PLC 程序（PLC 基本程序或用户程序）将经过一定逻辑运算的该信号传输至对应的接口信号：

DB21, ... DBX0.3（激活 DRF）

对 DRF 偏移的控制

可修改、删除或读取 DRF 偏移：

操作人员：	通过旋转手轮
零件程序：	- 通过轴专用系统变量 <code>\$AC_DRF[<轴>]</code> 读取 - 通过零件程序指令（DRFOF）在通道的所有轴上删除 - 通过零件程序指令（SUPA）非模态式抑制 文档： 编程手册之基本原理分册
PLC 用户程序：	读取 DRF 偏移（针对特定轴） 文档： 基本机床功能说明；PLC 基本程序（P3）
HMI 操作界面：	显示 DRF 偏移（针对特定轴）

说明

删除 DRF 偏移时，轴不会运行！

3.7 DRF 偏移

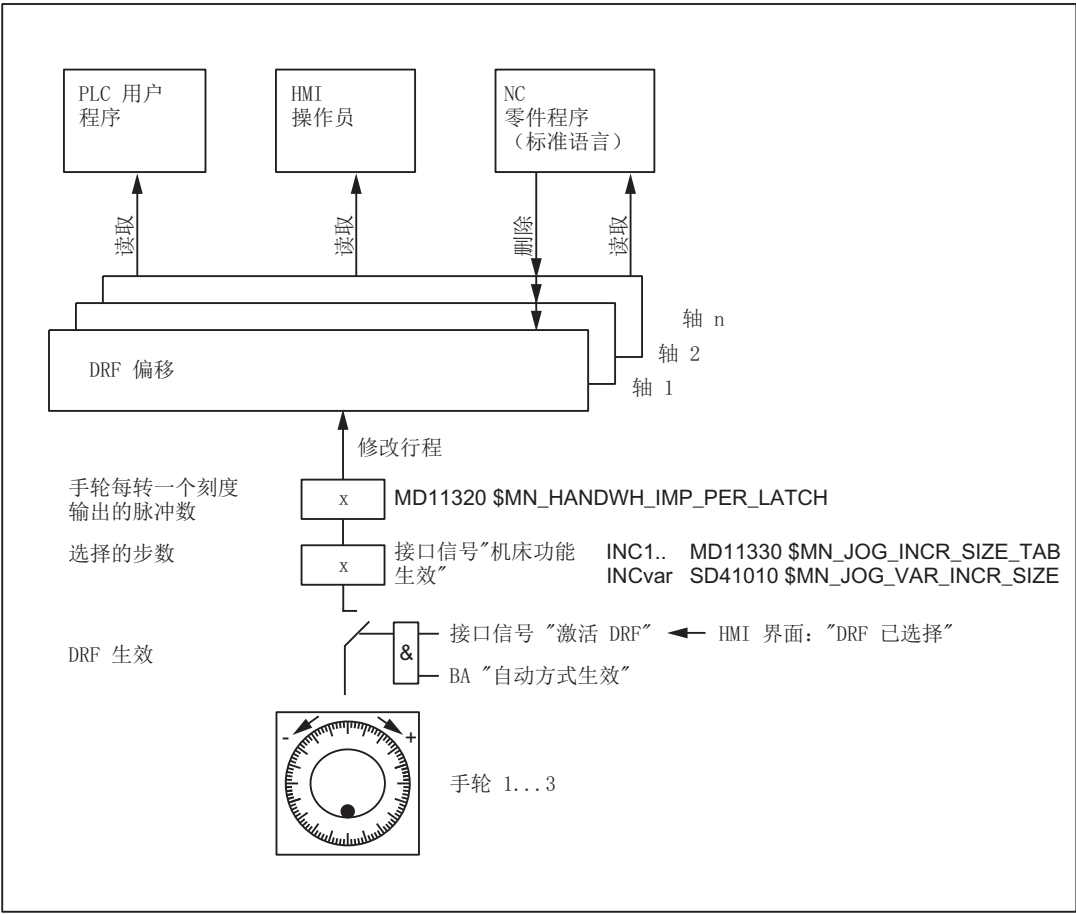


图 3-5 对 DRF 偏移的控制

显示

手轮执行 DRF 时，轴实际位置显示值（ACTUAL POSITION）不会变化。当前的 DRF 偏移可在 DRF 窗口中显示。

回参考点

在机床轴回参考点的阶段 1 中，系统会删除对应的几何轴或辅助轴的 DRF 偏移。
机床轴回参考点过程中，不允许同时执行几何轴或辅助轴的 DRF 偏移。

复位特性

上电复位：删除 DRF 偏移

3.8 手动运行至固定点

3.8.1 引言

功能

“手动运行至固定点”功能可使机床操作员利用机床控制面板上的移动键或手轮将轴运行至机床数据中定义的位置。轴到达指定的固定点后自动停止。

应用

典型应用示例：

- 启动 NC 程序前将轴运行至初始位置。
- 将轴运行至换刀点、装料点、托盘更换点。

前提条件

- “手动运行至固定点”功能只能在 JOG 方式中激活。
在子运行方式 JOG-REPOS、JOG-REF 以及 AUTO 方式下的 JOG 中无法激活此功能。
- 待运行的轴必须已执行回参考点。
- 不允许激活运动转换。
- 待运行的轴不允许与从动轴之间存在有效耦合。

使用 G75 运行至固定点

也可通过指令 G75 从零件程序中调用运行至已定义固定点功能。

有关“使用 G75 运行至固定点”的更多信息请参见：

文档：

编程手册之基本原理分册；章节：“补充指令”>“运行至固定点 (G75)”

3.8 手动运行至固定点

3.8.2 功能

步骤

“手动运行至固定点”的步骤:

- 选择 JOG 运行方式
- 激活“手动运行至固定点”功能
- 通过移动键或手轮将机床轴运行至固定点

激活

选择“手动运行至固定点”功能后, PLC 会通过下列位将目标固定点的编号以二进制编码方式输出至 NC:

DB31, ... DBX13.0-2 (JOG 方式下运行至固定点)

此功能一旦生效, NC 便会通过以下接口信号确认激活:

DB31, ... DBX75.0-2 (JOG 方式下运行至固定点生效)

说明

下列情形下无法激活此功能:

- NCK 复位期间
- 有未经应答的急停
- 执行异步子程序时

这些情形下不会发出报警。但功能会延迟一段时间激活, 直至相应功能完成或经过应答。

过程

真正的运行通过通过移动键或手轮启动, 运行方向为朝着指定固定点的方向。

选中的机床轴运行至指定固定点, 到达后自动停止。

采用“精准停”时, 到达固定点后会发出相应的 NC/PLC 接口信号:

DB31, ... DBX75.3-5 (已手动运行至固定点)

当轴通过 NC 程序、FC18 (适用于 840D sl) 或同步动作等其他方法到达机床坐标系中的目标固定点位置, 且实际位置位于“精准停”公差窗口 (MD36010 \$MA_STOP_LIMIT_FINE) 内时, 也会发出上述信号。

反向运行

反向运行（即朝着指定固定点的反向运行）的特性取决于以下机床数据中位 2 的设置：

MD10735 \$MN_JOG_MODE_MASK（JOG 运行方式的设置）

仅当该位置位时，才能执行反向运行。

若该位未置位，那么反向运行被禁用；尝试使用移动键或手轮使轴反向运行时，会输出如下通道状态消息：

“JOG: 方向已禁用 <轴>”

运行至其他固定点

若在运行至固定点期间选择了另一个固定点，轴运行将终止并发出以下报警：

报警 17812 “通道 %1 轴 %2 手动运行至固定点：固定点已被更改”

反馈信号 DB31, ... DBX75.0-2（手动运行至固定点生效）会显示新选择的固定点的编号。若需继续运行，必须重新启动 JOG 运行。

说明

若想避免触发报警信息，机床操作人员可如下操作：

1. 通过删除剩余行程终止当前运行。
 2. 轴停止后激活到其他固定点的运行并启动该运行。
-

离开固定点/取消

若需离开固定点位置，则必须取消“手动运行至固定点”功能。这通过将激活信号复位为“0”来进行：

DB31, ... DBX13.0-2 = 0

反馈信号 DB31, ... DBX75.0-2（手动运行至固定点生效）和 DB31, ... DBX75.3-5（已手动运行至固定点）在轴离开固定点时被删除。

特殊情况：轴已经位于固定点

若启动该功能时轴已位于指定的固定点，则轴将不再运行。此时会显示以下通道状态信息：

“JOG: 已到达位置 <轴>”

若需离开该固定点，则必须取消“手动运行至固定点”功能。

3.8 手动运行至固定点

应用在增量运行中的特殊性

在增量运行方式中，若在轴走完全部增量前已到达固定点，则视为轴已走完全部增量。此种情况也适用于仅运行整数增量的情形：

```
MD11346 $MN_HANDWH_TRUE_DISTANCE = 2 或 3
```

应用在模数回转轴上的特殊性

模数回转轴朝两个方向运行均可到达固定点。此时不会考虑采用最短行程 (DC)。

应用在主轴上的特殊性

激活“手动运行至固定点”功能后，主轴切换至定位方式。位置控制功能因此激活，主轴可以运行至固定点。

若未识别出零脉冲，则会发出报警，如同在进给轴方式中：

```
报警 17810“通道 %1 轴 %2 未执行回参考点”
```

由于主轴同时也必须是一个模数回转轴，所以在运行方向方面同模数回转轴相同（参见“应用在模数回转轴上的特殊性”一段）。

3.8.3 参数设置

反向运行

反向运行（即朝着指定固定点的反向运行）的特性取决于以下设置：

```
MD10735 $MN_JOG_MODE_MASK, 位 2（JOG 运行方式的设置）
```

位	值	含义
2	0	不能朝反向运行（缺省设置）。
	1	可以朝反向运行。

固定点位置

可为一根轴最多定义 4 个固定点位置：

```
MD30600 $MA_FIX_POINT_POS[0...3] = <固定点位置 1...4>
```


有效固定点位置的数量

MD30600 \$MA_FIX_POINT_POS 中输入的固定点位置有多少是有效位置，可在以下机床数据中设置：

MD30610 \$MA_NUM_FIX_POINT_POS = <有效固定点位置的数量>

说明

例外: G75

为确保兼容，在 G75 中即使：

MD30610 \$MA_NUM_FIX_POINT_POS = 0（无有效固定点位置）

MD30600 \$MA_FIX_POINT_POS[0] 和 [1] 中仍有两个固定点位置有效。

固定点位置 1 和 2 可通过 NC/PLC 接口激活，但只通过 G75 逼近。

3.8 手动运行至固定点

轴动态响应

“手动运行至固定点”中，单轴加速度和单轴加加速度通过下列机床数据定义：

- 通过**移动键或手轮**运行时：

- MD32301 \$MA_JOG_MAX_ACCEL (JOG 中的最大轴加速度)
- MD32436 \$MA_JOG_MAX_JERK (JOG 中的最大轴急动度)

说明

只有在为待运行的机床轴使能了单轴加加速度限制时，MD32436 \$MA_JOG_MAX_JERK 才生效：

MD32420 \$MA_JOG_AND_POS_JERK_ENABLE [<轴>] == TRUE

- 通过**零件程序指令 G75** 运行时：

- MD32300 \$MA_MAX_AX_ACCEL [0/1] (动态模式 DYNNORM/DYNPOS 中轨迹运动中的最大单轴加速度)

说明

定位轴的动态类型 (DYNNORM 或 DYNPOS) 通过以下机床数据定义：

MD18960 \$MN_POS_DYN_MODE = <模式>

- MD32431 \$MA_MAX_AX_JERK [0] (动态模式 DYNNORM 中轨迹运动中的最大单轴加加速度)

说明

只有在为待运行的机床轴使能了单轴加加速度限制时，MD32431 \$MA_MAX_AX_JERK 才生效：

MD32420 \$MA_JOG_AND_POS_JERK_ENABLE [<轴>] == TRUE

文档：

功能手册之基本功能分册；章节“加速度 (B2)”

3.8.4 编程

系统变量

“运行至固定点”功能可使用下列系统变量（这些变量可在零件程序和同步动作中读取）：

系统变量	含义
\$AA_FIX_POINT_SELECTED [<轴>]	目标固定点的编号
\$AA_FIX_POINT_ACT [<轴>]	当前轴所处的固定点的编号

3.8.5 边界条件

轴是分度轴

当待运行的轴是一个分度轴，且目标固定点与分度位置不一致时，轴不会运行，并会发出报警。

框架生效

系统会忽略所有生效的框架。运行在机床坐标系中进行。

补偿值生效

系统会一同执行补偿值（DRF、外部零点偏移、同步动作偏移 \$AA_OFF、在线刀具补偿）。固定点是机床坐标系中的一个位置。

若手动运行至固定点期间激活了补偿运行（DRF、外部零点偏移、同步动作偏移 \$AA_OFF、在线刀具补偿），则会发出报警。轴将不会运行至机床坐标系中指定固定点的位置，而是运行至假设未激活补偿运行时应达到的位置。不会发出固定点对应的 NC/PLC 接口信号 DB31, ... DBX75.3-5。

工作区域限制

轴运行时会考虑到工作区域限制（在 BCS 和 WCS 中），当到达限位时即终止运行。

3.8 手动运行至固定点

3.8.6 应用示例

目标

需要通过“手动运行至固定点”功能将回转轴（机床轴 4 [AX4]）运行至固定点 2（90 度）。

参数设置

机床轴 4 的“运行至固定点”机床数据参数如下设置：

MD30610 \$MA_NUM_FIX_POINT_POS[AX4] = 4 为机床轴 4 定义 4 个固定点。
MD30600 \$MA_FIX_POINT_POS[0,AX4] = 0 AX4 的第 1 个固定点 = 0 度
MD30600 \$MA_FIX_POINT_POS[1,AX4] = 90 AX4 的第 2 个固定点 = 90 度
MD30600 \$MA_FIX_POINT_POS[2,AX4] = 180 AX4 的第 3 个固定点 = 180 度
MD30600 \$MA_FIX_POINT_POS[3,AX4] = 270 AX4 的第 4 个固定点 = 270 度

初始状态

机床轴 4 已回参考点且位于 0 度位置上。即该轴已位于第 1 个固定点上，这一状态通过 NC/PLC 接口信号反馈：

DB31 DBX75.0 = 1（位 0-2 = 1）

运行至固定点 2

控制系统切换至 JOG 运行方式。

通过 NC/PLC 接口信号激活向固定点 2 的运行：

DB31 DBX13.1 = 1（位 0-2 = 2）

通过 NC/PLC 接口信号确认激活：

DB31 DB75.1 = 1（位 0-2 = 2）

通过机床控制面板上的正向移动键连续运行，即可到达固定点 2。

机床轴 4 停止在 90 度位置。这一状态通过 NC/PLC 接口信号反馈：

DB31 DBX75.4 = 1（位 3-5 = 2）

3.9 沿刀具中心轴的退刀（手动退刀）

3.9.1 概述

功能

在 AUTO 或 MDI 运行方式下由于控制系统断电或通道复位造成程序终止时，可通过子运行方式“手动退刀”在机床坐标系中手动使刀具沿其中心轴退回。

此时尤其需要注意下列功能的特殊性：

- 带补偿夹具的攻丝和带编码器的转速控制主轴（G33）
- 不带补偿夹具的攻丝（刚性攻丝）和位置控制主轴（G331/G332）
- CYCLE800：带刀具定向的加工或刀具定向转换

可用性

手动退刀功能可被选中，仅当：

- 使用 G33/G331/G332 攻丝

或通常：

- 使用钻孔工具加工（工具型号 200 至 299）

通过通道复位或断电中断。

需要恢复的数据

为了能在程序终止后执行沿刀具中心轴的退刀，必须恢复程序终止前通道中生效的下列数据：

- 生效的刀具补偿
- 生效的加工平面
- 生效的刀架
- 生效的坐标转换数据组，包含转换参数
- G33 或 G331/G332 中的螺纹组数据
- 参与坐标转换的轴的位置

3.9 沿刀具中心轴的退刀（手动退刀）

若程序终止后这些数据完整存在，在选择子运行方式“手动退刀”后系统即自动在通道中恢复这些数据。此时控制系统会对工件坐标系（WCS）进行调整，使其中一条几何轴与刀具中心轴方向吻合。之后便可通过运行几何轴来手动退刀。

说明

数据备份

备份的不是上述数据本身，而是这些数据的参照。若在选择手动退刀前修改了数据，则会基于改动过的数据执行功能。

限制

在手动退刀方式下不执行可通过中断信号启动的程序。

3.9.2 参数设置

NC 专用机床数据

上电后自动选择手动退刀

控制系统启动（上电）后，通道处于设置的缺省运行方式下：

MD10720 \$MN_OPERATING_MODE_DEFAULT[<BAG>] = <缺省运行方式>

可通过以下机床数据设置，在控制系统启动后自动选择 JOG 方式，而不考虑设置的缺省运行方式。但是仅在一条 BAG 通道中存在退刀数据时才能执行该自动选择。

MD10721 \$MN_OPERATING_MODE_EXTENDED[<BAG>] = <值>	
值	含义
0	按照 MD10720 \$MN_OPERATING_MODE_DEFAULT 选择运行方式
1	DB21, ... DBX377.5 == 1 时（“退刀数据存在”），选择 JOG 运行方式

运行方向使能

可将退刀方向限制为正向或可将两个方向都使能：

MD10735 \$MN_JOG_MODE_MASK, 位 8	
值	含义
0	运行使能：仅正向
1	运行使能：正向和负向

轴机床数据

测量系统的状态

测量系统的状态通过机床数据显示或者必须事先将其设置为所需的状态：

MD34210 \$MA_ENC_REFP_STATE[<轴>]	
值 ¹⁾	含义
绝对值编码器	
2	编码器已校准
增量编码器	
1	自动回参考点：使能，但编码器尚未回参考点
2	自动回参考点：编码器已回参考点并在准停位置，自动回参考点功能将在下一次激活编码器时生效
3	实际位置恢复：恢复关闭前保存在缓存中的最终轴位置，不进行自动回参考点
1) 仅手动退刀涉及的值	

3.9.3 选择

功能

前提条件

只有在通道中有退刀数据且通道处于 JOG 运行方式、为“复位”状态时，才能选择手动退刀。

- DB21, ... DBX377.5 == 1（退刀数据存在）
- DB11, ... DBX(n*20+6).2 == 1（生效的运行方式：JOG，其中 n=0, 1, 2 ... 对应 BAG 1, 2, ...）
- DB21, ... DBX35.7 == 1（通道状态：复位）

3.9 沿刀具中心轴的退刀（手动退刀）

进给轴和主轴

控制系统重新接通（上电）后，参与手动退刀的机床轴的所有有效测量系统都必须处于“已回参考点”或“已恢复”状态。

关于如何在控制系统重新接通（上电）后自动恢复实际位置的详细说明请参见：

文档

功能手册之基本功能分册，章节“R1 回参考点”>“机床参照的自动恢复”

跨通道取轴

如果在选择该功能前通道还没有获得所有必须参与退刀的进给轴和主轴，则系统会隐形地跨通道获取缺少的轴。

坐标系

选择手动退刀时，系统会设置工件坐标系（WCS）用于运行通道轴。参与退刀的轴只能在 WCS 中运行。所有不参与退刀的轴也可在机床坐标系（MCS）中运行。

选择方法

可将 JOG 自动选为上电后的运行方式来方便手动退刀的选择。手动退刀的真正选择则需通过操作界面或 PLC 用户程序手动实现。

自动选择 JOG 运行方式

若通道中存在退刀数据，控制系统启动后会自动选择运行方式 JOG。该设置通过机床数据 MD10721 \$MN_OPERATING_MODE_EXTENDED 进行。参见章节“参数设置 (页 206)”。

通过操作界面选择

在操作界面上通过“退刀”软键选择手动退刀：

操作区“加工”>“ETC 键（“>”）>“退刀”

通过 PLC 用户程序选择

通过 PLC 用户程序选择手动退刀须进行以下操作：

- 针对特定通道询问退刀数据是否存在
DB21, ... DBX377.5 == 1（退刀数据存在）
- 针对 BAG 选择 JOG 运行方式：
DB11, ... DBX(n*20+6).2 == 1（生效的运行方式：JOG，其中 n=0, 1, 2 ... 对应 BAG 1, 2, ...）

- 通过 PI 服务“RETRAC”针对特定通道选择手动退刀。
关于如何通过功能块 FB 4 激活 PI 服务“RETRAC”的详细说明请见：
文档
功能手册之基本功能分册，章节“P3: 用于 SINUMERIK 840D sl 的 PLC 基本程序”>“PI 服务”
- 在确认选择了手动退刀（见下）后，选择工件坐标系：
 - DB19.DBX20.7（MCS/WCS 切换）
 - DB19.DBX0.7（实际值，1=WCS，0=MCS）

选择确认

成功选择后，以下 NC/PLC 接口信号置位：

DB21, ... DBX377.4 = 1（手动退刀生效）

螺纹加工（G33 或 G331/G332）

若螺纹加工（G33 或 G331/G332）期间程序处理终止，那么在选择手动退刀后系统将恢复刀具轴和主轴组成的轴组。此时，相关轴的控制器参数和参数组也按照零件程序中编写的螺纹加工方式进行设置。

3.9.4 退刀

一般退刀方式

退刀是通过在工件坐标系（WCS）中手动运行选择“手动退刀”时定义的几何轴来实现的。退刀可通过机床控制面板（MCP）上的移动键或手轮控制。退刀受到各限位（工作区域限制、软件限位开关等等）的约束。

退刀可通过按下“CYCLE STOP”和“CYCLE START”来暂停和继续。

未参与退刀的进给轴和主轴可任意手动运行。

运行方向

缺省设置下只使能正向退刀。若也需要沿负向退刀，则必须通过以下数据明确使能负向：

MD10735 \$MN_JOG_MODE_MASK, 位 8

螺纹切削（G33）或攻丝（G331, G332）时的退刀方式

进行螺纹加工时（G33 或 G331/G332），除几何轴外主轴也参与退刀。运行所选择的几何轴或主轴时便可执行退刀。

3.9 沿刀具中心轴的退刀（手动退刀）

通过手轮退刀

参与退刀的轴中若有一根轴通过移动键运行，那么作用于其他轴的手轮脉冲将被忽略。

若为多根轴选择了手轮并运行，则会按以下顺序计算手轮脉冲：

1. 退刀轴
2. 主轴
3. 未参与退刀的进给轴/主轴

其他手轮的脉冲会被忽略，直到高优先级手轮停止转动后才会被计算。

禁用功能

在手动退刀期间系统会忽略对以下功能的请求：

- 通过 DB31, ... DBX30 启动主轴
- 通过功能块 FC18 运行主轴或参与退刀的进给轴
- 将主轴或参与退刀的进给轴设为由 PLC 控制的轴
- 将主轴或参与退刀的进给轴切换到另一个通道
- 将主轴或参与退刀的进给轴设为主处理轴（指令轴、摆动轴、FC18/并行轴）

3.9.5 取消

采用以下方法可针对特定通道取消手动退刀：

- 通道复位
 - 机床控制面板：“复位”键
 - PLC 基本程序：DB21, ... DBX7.7 = 1（复位）
- 操作界面：软键“返回”（“<<”）

此时下列机床数据的缺省设置生效：

- MD20110 \$MC_RESET_MODE_MASK[<通道>] ()
- MD20150 \$MC_GCODE_RESET_VALUES[<通道>] ()
- MD20151 \$MC_GCODE_RESET_MODE[<通道>] ()

取消此功能后，通道处于 JOG 运行方式、“复位”状态。

这样便可在 JOG 运行方式下手动运行所有通道轴。退刀数据将保留，以便再次选择手动退刀。

NC/PLC 接口信号

取消手动退刀后，以下 NC/PLC 接口信号复位：

DB21, ... DBX377.4 = 0（手动退刀生效）

3.9.6 重新选择

只要退刀数据存在，便可以随时重新选择手动退刀：

DB21, ... DBX377.5 == 1（退刀数据存在）

重新选择此功能时将恢复原始的退刀数据。每次可将不同的几何轴设为退刀轴。

轴位置变动

在手动退刀取消期间，通道轴可在 JOG 方式下运行。重新选择手动退刀时，退刀将依据新的轴位置进行。

3.9.7 继续程序执行

AUTO 运行方式

在 AUTO 运行方式下按下“CYCLE START”键以继续执行中断的零件程序前，所有有效测量系统为“已恢复”或“未回参考点”状态的机床轴均必须执行回参考点。

注意

可出现轴位置偏移

控制系统掉电重启后，如果轴采用的是增量测量系统，系统会根据机床数据的设置同步或恢复轴位置。在子运行方式“手动退刀”下退刀后，位置恢复后的轴必须执行回参考点。

MDI 运行方式和溢出转存

在 MDI 运行方式和溢出转存中，也可采用恢复后的位置进行加工，而无需执行轴回参考点。为此，必须针对特定通道明确使能系统以恢复后的位置启动：

MD20700 \$MC_REFP_NC_START_LOCK = 2

搜索程序断点

通过搜索程序断点，可从断点起继续执行零件程序。程序断点是程序中断前主程序级上的最后一个程序段。

对程序段搜索功能和操作的详细说明请参见：

3.9 沿刀具中心轴的退刀（手动退刀）

文档

功能手册之基本功能分册；章节“K1: BAG，通道，程序运行，复位特性”> “程序段搜索”
或 “程序段搜索类型 5 SERUPRO”

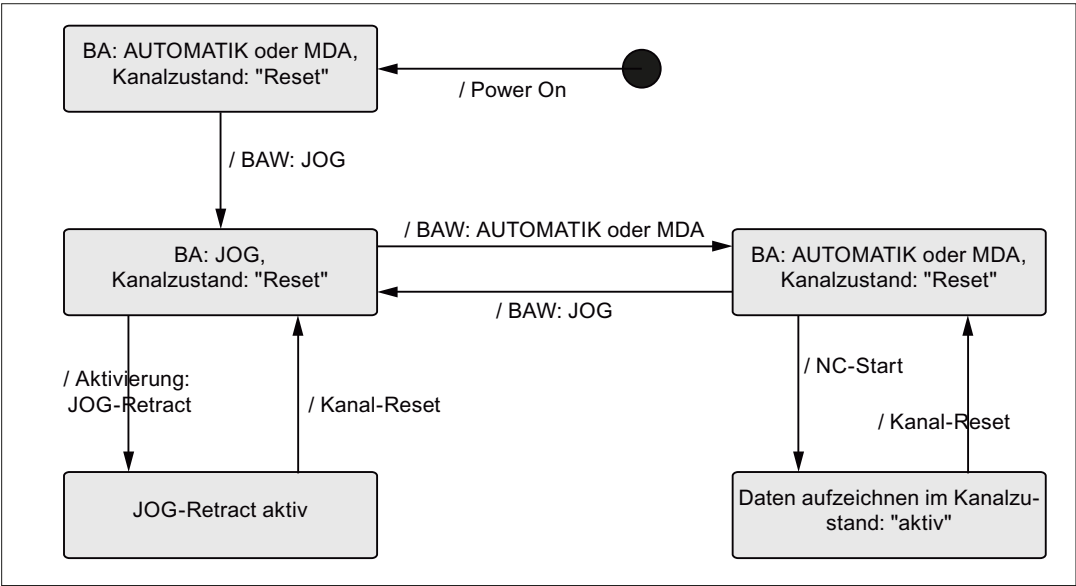
操作手册 “车削版” 或操作手册 “铣削版”； 章节 “加工工件”>“在特定位置开始加工”

通过“CYCLE START” 键继续执行

在 AUTO 或 MDI 运行方式中，按下“CYCLE START” 键可从选择的位置起继续执行零件程序。此时系统会执行下列与手动退刀相关的任务：

- 删除退刀数据。
- 复位 NC/PLC 接口信号：
DB21, ... DBX377.5 = 0（退刀数据存在）
- 备份当前的数据环境。

3.9.8 状态图



BA 运行方式
BA 运行方式切换
W

图 3-6 状态图：手动退刀

3.9.9 系统数据

下列系统数据用于手动退刀：

含义	系统变量 \$VA_	NC/PLC 接口	OPI 变量
退刀数据存在	-	DB21, ... DBX377.5	retractState, 位 0
手动退刀生效	-	DB21, ... DBX377.4	retractState, 位 1
退刀轴	-	-	retractState, 位 2 - 3
恢复后的位置, 第 1 测量系统	\$AA_POSRES	DB31, ... DBX71.4	aaPosRes
恢复后的位置, 第 2 测量系统	\$AA_POSRES	DB31, ... DBX71.5	aaPosRes

文档

OPI 变量和系统变量

参数手册“参数手册 2”，章节“变量”

3.9.10 边界条件

增量测量系统

机床轴采用增量测量系统时，用户须确保掉电后机床轴能足够迅速地制动、不再移动，以防止控制系统最后保存的位置发生变化。否则控制系统重启后恢复的位置将大大偏离机床轴的实际位置。此类机床轴上不允许激活**驱动自动控制的退回**。

关于如何在控制系统重新接通（上电）后自动恢复实际位置的详细说明请参见：

文档

功能手册之基本功能分册，章节“R1 回参考点”>“机床参照的自动恢复”

驱动自动控制的退回

参与退刀的机床轴上不允许激活“驱动自动控制的退回”功能。

轴耦合

选择手动退刀不会恢复轴耦合。

使用 G63 攻丝

手动退刀无法用于带补偿夹具的攻丝和无编码器的转速控制主轴（G63）。

3.10 调试: 手轮

转换

若通过手动退刀选择了坐标转换, 则所有参与坐标转换的机床轴的有效测量系统必须为“已回参考点”或“已恢复”。

OEM 坐标转换, 例如针对六足机器人的平行运动, **只能通过已回参考点的测量系统运行**。
无法采用恢复后的轴位置运行。

通过直接编写向轴进行刀具定向

当刀具定向不通过 NC 功能, 而是通过直接编写定位轴进行时, 手动退刀无法生成沿刀具中心轴的退刀。

NCU-Link

手动退刀也可以和 NCU-Link 组合使用 (参见章节“NCU-Link (页 88)”)。但是轴容器的状态不会因手动退刀改变。进入手动退刀方式前, 用户必须在现有方式 (例如在 MDI) 下为沿刀具中心轴的退刀进行必要的调整。

3.10 调试: 手轮

3.10.1 简介

为了在 SINUMERIK 控制系统上使用手轮, 必须通过 NCK 机床数据对其进行设置。

若手轮不直接连至控制系统, 则还须采取其他措施, 例如: 在手轮通过 PROFIBUS- 或 Ethernet MCP/手轮模块连至控制系统时还需要使用 SIMATIC STEP 7 HW-Config 添加和配置模块。

说明

SINUMERIK 控制系统上最多可设置 6 个手轮。

可连接的控制系统

SINUMERIK 840D sl

SINUMERIK 840D sl 上通过下列组件连接手轮:

- PROFIBUS (页 218) 模块
- Ethernet (页 221) 模块

说明

在 SINUMERIK 840D sl 控制系统上可同时运行多个通过不同模块接入的手轮。

SINUMERIK 828D

SINUMERIK 828D 上通过下列组件连接手轮:

- PPU (页 215)
- PROFIBUS (页 216) 机床控制面板 (MCP)

说明

在 SINUMERIK 828D 控制系统上可同时运行多个通过不同模块接入的手轮。

3.10.2 通过 PPU 连接 (仅针对 828D)

参数设置

直接连至 PPU X143 端子的手轮通过下列 NCK 机床数据设置:

- MD11350 \$MN_HANDWHEEL_SEGMENT[< NCK 中手轮编号 - 1 >] = 2
直接连至 PPU 时, 手轮段赋值必须为 2 (8xxD_HW)。
- MD11351 \$MN_HANDWHEEL_MODULE[< NCK 中手轮编号 - 1 >] = 1
直接连至 PPU 时, 赋值必须为 1。
- MD11352 \$MN_HANDWHEEL_INPUT[< NCK 中手轮编号 - 1 >] = <手轮连接>
必须输入手轮连接的编号: 1 或 2

说明

一个 PPU (端子 X143) 上最多可连接 2 个手轮。

3.10 调试: 手轮

示例

设置通过 PPU 端子 X143 连接的两个手轮。

机床数据	值	含义
		第 1 个手轮:
MD11350 \$MN_HANDWHEEL_SEGMENT[0]	2	硬件段: 8xxD_HW
MD11351 \$MN_HANDWHEEL_MODULE[0]	1	---
MD11352 \$MN_HANDWHEEL_INPUT[0]	1	1. X143 的手轮连接
		第 2 个手轮:
MD11350 \$MN_HANDWHEEL_SEGMENT[1]	2	硬件段: 8xxD_HW
MD11351 \$MN_HANDWHEEL_MODULE[1]	1	---
MD11352 \$MN_HANDWHEEL_INPUT[1]	2	2. X143 的手轮连接

3.10.3 通过 PROFIBUS 连接 (828D)

参数设置

在 SINUMERIK 828D 上, 除了能将两个手轮连接到 PPU 接口, 端子 X143 上外, 还可将第三个手轮通过机床操作面板 (如 MCP 483C PN) 连接到接口 X60 上。

在以下 NCK 机床数据中设置**第三个**手轮:

- MD11350 \$MN_HANDWHEEL_SEGMENT[2] = 5
- MD11351 \$MN_HANDWHEEL_MODULE[2] = 1
- MD11352 \$MN_HANDWHEEL_INPUT[2] = 1

前提条件

运行带有标准数据 (机床数据、STEP 7 配置) 的控制系统。

示例

设置 3 个手轮，通过 PPU 和机床控制面板 “MCP 483C PN” 连接。

NCK 中的手轮编号	机床数据组 (下标)	连接
1	0	PPU, 手轮插槽中的第 1 个手轮
2	1	PPU, 手轮插槽中的第 2 个手轮
3	2	MCP, 手轮插槽中的第 1 个手轮

NCK 机床数据中的设置:

机床数据	值	含义
		NCK 中的第 1 个手轮
MD11350 \$MN_HANDWHEEL_SEGMENT[0]	2	硬件段: 8xxD_HW
MD11351 \$MN_HANDWHEEL_MODULE[0]	1	---
MD11352 \$MN_HANDWHEEL_INPUT[0]	1	手轮插槽中连接第 1 个手轮
		NCK 中的第 2 个手轮
MD11350 \$MN_HANDWHEEL_SEGMENT[1]	2	硬件段: 8xxD_HW
MD11351 \$MN_HANDWHEEL_MODULE[1]	1	---
MD11352 \$MN_HANDWHEEL_INPUT[1]	2	手轮插槽中连接第 2 个手轮
		NCK 中的第 3 个手轮
MD11350 \$MN_HANDWHEEL_SEGMENT[2]	5	硬件段: PROFIBUS
MD11351 \$MN_HANDWHEEL_MODULE[2]	1	参照 MCP 的手轮插槽的逻辑基本地址
MD11352 \$MN_HANDWHEEL_INPUT[2]	1	手轮插槽中的第 1 个手轮

3.10.4 通过 PROFIBUS 连接 (840D sl)

设置

通过 PROFIBUS 模块（例如机床控制面板“MCP 483”）连接的手轮在下列 NCK 机床数据中设置：

- MD11350 \$MN_HANDWHEEL_SEGMENT[< NCK 中手轮编号 - 1>] = 5
通过 PROFIBUS 模块连接时，硬件段赋值必须为 5（PROFIBUS）。
- MD11351 \$MN_HANDWHEEL_MODULE[< NCK 中手轮编号 - 1>] = < 下标 + 1>
必须输入对 MD11353 \$MN_HANDWHEEL_LOGIC_ADDRESS [< 下标>] 的参照，其包含手轮插槽的逻辑基本地址。
- MD11352 \$MN_HANDWHEEL_INPUT[< NCK 中手轮编号 - 1>] = < 手轮插槽中编号 >
一个手轮插槽可能对应多个手轮。 必须输入插槽内手轮的编号：1, 2, ...
- MD11353 \$MN_HANDWHEEL_LOGIC_ADDRESS[< 下标>] = < 逻辑基本地址>
必须输入 SIMATIC STEP 7 W-Config 中定义的手轮插槽逻辑基本地址。

手轮插槽

除了在 NCK 机床数据中设置手轮外，还必须在 STEP 7 中配置 PROFIBUS 模块。其中包括手轮插槽逻辑地址的定义。

手轮插槽位于各 PROFIBUS 模块上的下列插槽上：

PROFIBUS 模块	插槽
机床控制面板 MCP 483	2
机床控制面板 MCP 310	2
手轮连接模块	1

示例

设置 5 个手轮，通过 4 个机床控制面板“MCP 483” 连接。一个“MCP 483” 机床控制面板上最多可连接 2 个手轮。

NCK 中的手轮编号	机床数据组（下标）	连接
1	0	第 1 个 MCP 上的手轮插槽中连接第 1 个手轮
2	1	第 1 个 MCP 上的手轮插槽中连接第 2 个手轮

NCK 中的手轮编号	机床数据组 (下标)	连接
3	2	第 2 个 MCP 上的手轮插槽中连接第 1 个手轮
5	4	第 3 个 MCP 上的手轮插槽中连接第 1 个手轮
6	5	第 4 个 MCP 上的手轮插槽中连接第 2 个手轮

第 4 个手轮在 NCK 中不使用 (机床数据空位)。

说明

在 NCK 机床数据中设置手轮时, 允许存在机床数据空位。

机床控制面板已在 SIMATIC STEP 7 W-Config 中进行如下配置:

	插槽	DP 标识	订货号/名称	输入地址	输出地址
第 1 个 MCP	1	55	标准+手轮	0 ... 7	0 ... 7
	2	2AE	→ 标准+手轮	288 ... 291	
	3	1	→ 标准+手轮		
第 2 个 MCP	1	55	标准+手轮	8 ... 15	8 ... 15
	2	2AE	→ 标准+手轮	304 ... 307	
	3	1	→ 标准+手轮		
第 3 个 MCP	1	55	标准+手轮	16 ... 23	16 ... 23
	2	2AE	→ 标准+手轮	320 ... 323	
	3	1	→ 标准+手轮		
第 4 个 MCP	1	55	标准+手轮	24 ... 29	24 ... 29
	2	2AE	→ 标准+手轮	330 ... 333	
	3	1	→ 标准+手轮		

NCK 机床数据中的设置:

机床数据	值	含义
		NCK 中的第 1 个手轮
MD11350 \$MN_HANDWHEEL_SEGMENT[0]	5	硬件段: PROFIBUS

3.10 调试: 手轮

机床数据	值	含义
MD11351 \$MN_HANDWHEEL_MODULE[0]	1	参照第 1 个 MCP 的手轮插槽的逻辑基本地址
MD11352 \$MN_HANDWHEEL_INPUT[0]	1	手轮插槽中的第 1 个手轮
		NCK 中的第 2 个手轮
MD11350 \$MN_HANDWHEEL_SEGMENT[1]	5	硬件段: PROFIBUS
MD11351 \$MN_HANDWHEEL_MODULE[1]	1	参照第 1 个 MCP 的手轮插槽的逻辑基本地址
MD11352 \$MN_HANDWHEEL_INPUT[1]	2	手轮插槽中的第 2 个手轮
		NCK 中的第 3 个手轮
MD11350 \$MN_HANDWHEEL_SEGMENT[2]	5	硬件段: PROFIBUS
MD11351 \$MN_HANDWHEEL_MODULE[2]	2	参照第 2 个 MCP 的手轮插槽的逻辑基本地址
MD11352 \$MN_HANDWHEEL_INPUT[2]	1	手轮插槽中的第 1 个手轮
		NCK 中的第 4 个手轮
MD11350 \$MN_HANDWHEEL_SEGMENT[3]	0	不设置手轮
MD11351 \$MN_HANDWHEEL_MODULE[3]	0	不设置手轮
MD11352 \$MN_HANDWHEEL_INPUT[3]	0	不设置手轮
		NCK 中的第 5 个手轮
MD11350 \$MN_HANDWHEEL_SEGMENT[4]	5	硬件段: PROFIBUS
MD11351 \$MN_HANDWHEEL_MODULE[4]	6	参照第 3 个 MCP 的手轮插槽的逻辑基本地址
MD11352 \$MN_HANDWHEEL_INPUT[4]	1	手轮插槽中的第 1 个手轮
		NCK 中的第 6 个手轮

机床数据	值	含义
MD11350 \$MN_HANDWHEEL_SEGMENT[5]	5	硬件段: PROFIBUS
MD11351 \$MN_HANDWHEEL_MODULE[5]	5	参照第 4 个 MCP 的手轮插槽的逻辑基本地址
MD11352 \$MN_HANDWHEEL_INPUT[5]	2	手轮插槽中的第 2 个手轮

逻辑基本地址:

机床数据	值	含义
MD11353 \$MN_HANDWHEEL_LOGIC_ADDRESS [0]	288	第 1 个 MCP 的手轮插槽的逻辑基本地址
MD11353 \$MN_HANDWHEEL_LOGIC_ADDRESS [1]	304	第 2 个 MCP 的手轮插槽的逻辑基本地址
MD11353 \$MN_HANDWHEEL_LOGIC_ADDRESS [4]	330	第 4 个 MCP 的手轮插槽的逻辑基本地址
MD11353 \$MN_HANDWHEEL_LOGIC_ADDRESS [5]	320	第 3 个 MCP 的手轮插槽的逻辑基本地址

3.10.5 通过以太网连接（仅针对 840D sl）

参数设置

通过 Ethernet 模块（例如机床控制面板“MCP 483C IE”，“HT 8”或“HT 2”）连接的手轮在下列 NC 机床数据中设置:

- MD11350 \$MN_HANDWHEEL_SEGMENT[< x - 1 >] = 7
通过以太网模块连接手轮时，硬件段赋值必须为 7（Ethernet）。
- MD11351 \$MN_HANDWHEEL_MODULE[< x - 1 >] = 1
通过以太网模块连接手轮时，模块赋值必须为 1。
- MD11352 \$MN_HANDWHEEL_INPUT[< x - 1 >] = y
其中 y = 1, 2, 3, ...（以太网总线上的手轮接口）

其中 x = 1, 2, 3, ...（NC 中的手轮编号）

3.10 调试: 手轮

以太网总线上的手轮接口

以太网总线上手轮接口的编号通过下列定义得出:

- 操作组件接口的顺序为: MCP1, MCP2, BHG
- 每个操作组件接口提供两个手轮接口
- 操作组件: MCP 483C IE
 在一个 MCP 483C IE 上可通过 X60 和 X61 接口最多连接两个手轮。操作组件接口和手轮的对应关系为:
 - X60 接口: 操作组件接口 MCP1 / MCP2 上连接第 1 个手轮
 - X61 接口: 操作组件接口 MCP1 / MCP2 上连接第 2 个手轮
- 操作组件: HT 8
 手轮 HT 8 总是操作组件接口 MCP1 / MCP2 上的第 1 个手轮。
- 操作组件: HT 2
 手轮 HT 2 总是操作组件接口 BHG 上的第 1 个手轮。

操作组件接口 ->		MCP1		MCP2		BHG	
手轮接口 ¹⁾		1	2	1	2	1	2
FB1 参数 ²⁾		MCP1BusAdr		MCP2BusAdr		BHGRecGDNo	
手轮指定 ³⁾							
	MCP 483C IE	X60	X61	X60	X61	-	-
	HT 8	x	-	x	-	-	-
	HT 2	-	-	-	-	x	-
以太网总线上的手轮接口 (y) ⁴⁾ ->		1	2	3	4	5	6
1) 在一个操作组件接口内手轮接口的编号							
2) 操作组件和接口的对应关系通过对应 FB1 参数指定							
3) 各操作组件接口上的手轮接口和手轮的对应关系							
4) 以太网总线上手轮接口的编号 -> MD11352 \$MN_HANDWHEEL_INPUT[< x - 1 >] = y							

示例

设置 3 个手轮，通过下列操作组件连接：

操作组件接口 ->	MCP1		MCP2		BHG	
操作组件	HT 8		MCP 483C		HT 2	
FB1 参数	MCP1BusAdr := 39		MCP2BusAdr := 192		BHGRcGDNo := 40	
手轮接口	x	-	-	X61	x	-
以太网总线上的手轮接口 ->	1	2	3	4	5	6

表格 3-1 用于指定手轮的 NCK 机床数据

机床数据	值	描述
		HT 8: NC 中手轮编号 = 1
MD11350 \$MN_HANDWHEEL_SEGMENT[0]	7	段: 以太网
MD11350 \$MN_HANDWHEEL_MODULE[0]	1	模块: 以太网
MD11350 \$MN_HANDWHEEL_INPUT[0]	1	以太网总线上的手轮接口
		MCP 483C IE: NC 中手轮编号 = 2
MD11350 \$MN_HANDWHEEL_SEGMENT[1]	7	段: 以太网
MD11350 \$MN_HANDWHEEL_MODULE [1]	1	模块: 以太网
MD11350 \$MN_HANDWHEEL_INPUT [1]	4	以太网总线上的手轮接口
		HT 2: NC 中手轮编号 = 3
MD11350 \$MN_HANDWHEEL_SEGMENT[2]	7	段: 以太网
MD11350 \$MN_HANDWHEEL_MODULE [2]	1	模块: 以太网
MD11350 \$MN_HANDWHEEL_INPUT [2]	5	以太网总线上的手轮接口

3.10 调试：手轮

表格 3-2 FB1 参数（选段）

参数	值	注释
MCPNum	:= 2	// 连接的 MCP 的数量 // MCP1 = HT 8
MCP1In	...	// MCP1 参数 ...
...	...	
MCP1BusAdr	:= 39	// 通过连接设备上的开关 S1 和 S2 // 设置的“IP 地址” // MCP2 = MCP 483C IE
MCP2In	...	// MCP2 参数 ...
...	...	
MCP2BusAdr	:= 192	// 通过 MCP 483C 上的开关 S2 // 设置的“IP 地址”
MCPBusType	:= b#16#55	// 总线类型：以太网 // BHG = HT 2
BHG	:= 5	// 总线类型：以太网
BHGIn	...	// BHG 参数 ...
...	...	
BHGRecGDNo	:= 40	// 通过连接设备上的开关 S1 和 S2 // 设置的“IP 地址”

滤波时间

手轮脉冲在以太网总线上的传输具有不确定性，因此在高动态响应的传动设备上可能需要对手轮脉冲传输进行滤波（平滑）。滤波时间通过以下机床数据设置：

- MD11354 \$MN_HANDWHEEL_FILTER_TIME[< x - 1 >] = <滤波时间>
其中 x = 1, 2, 3, ...（NC 中的手轮编号），且滤波时间 = 0.0 ... 2.0 s

滤波时间用于设定何时将发送到控制系统的手轮脉冲传输至插补器。滤波时间为 0.0 s 时，手轮脉冲将在一个单独的插补周期内传输至插补器。这可能会导致轴急动。

建议采用的滤波时间：0.2 – 0.5 s

静态辨识

静态辨识通过连接手轮的以太网模块进行。若一个手轮在定义的时间内未传输脉冲，以太网模块会将该状态辨识为手轮静止并将该状态传输至 NC/PLC 接口：

NC/PLC 接口信号	值	含义
DB10, DBX245.0	0	手轮 1 运行
	1	手轮 1 静止

NC/PLC 接口信号	值	含义
DB10, DBX245.1	0	手轮 2 运行
	1	手轮 2 静止
DB10, DBX245.2	0	手轮 3 运行
	1	手轮 3 静止

通过计算这些信号，可缩短由于累积在控制系统中、尚未传输给插补器的手轮脉冲导致的轴滞后行程。为此，一旦辨识出静态，必须触发轴或通道内的剩余行程删除操作：

- DB31,... DBX2.2 = 1（针对特定轴的剩余行程删除）
- DB21,... DBX6.2 = 1（针对特定通道的剩余行程删除）

3.11 手动运行和手轮运行时的特点

3.11.1 几何轴/定向轴的手动运行

JOG 方式下的坐标系

在 JOG 运行方式下，操作人员也可在工件坐标系（WCS）中手动运行几何轴。此时选中的坐标平移或旋转在手动运行中保持生效。

说明

在 SINUMERIK 840D sl 上使用“Handling 坐标转换软件包”可在 JOG 方式下在多种有效参考系中单独设置几何轴的坐标转换。

文档：

功能手册之特殊功能分册；多轴转换（F2）章节：“坐标手动运行”

应用

坐标转换和框架必须生效的手动运行。几何轴采用最后生效的坐标系运行。下文会对几何轴手动运行的特殊之处进行说明。

并行运行

在通过移动键连续或断续运行几何轴时请注意，在一个时间点上只可运行一根几何轴。若尝试运行超过一根几何轴，则会触发报警 20062“轴已生效”。但是如果使用手轮，则可

3.11 手动运行和手轮运行时的特点

通过手轮 1 至 3 同时运行 3 根几何轴。如果其中有一根轴未定义为几何轴，则会发出对应报警 20060。

PLC 接口

几何轴/定向轴有独立的 PLC 接口，其包含的信号与轴专用 PLC 接口相同：

- 几何轴：
DB21, ... DBB12-23 和
DB21, ... DBB40-56
- 定向轴：
DB21, ... DBB320-331 和
DB21, ... DBB332-343

进给修调/快速移动修调

手动运行几何轴时，针对特定通道的进给修调开关或快速移动修调开关生效。

加速度和加加速度

手动运行几何轴/定向轴时，可针对通道对加速度和加加速度进行限制。这样可提升坐标转换功能的易操作性，并可完全由回转轴产生坐标运动（机器人）。

几何轴

手动运行几何轴时，可通过以下机床数据针对通道设定加速度上限：

MD21166 \$MC_JOG_ACCEL_GEO [<几何轴>]

其中 <几何轴> = 0, 1, 2

在加速模式 SOFT（限制急动的加速）中手动运行几何轴时，可通过以下机床数据针对通道设定加加速度上限：

MD21168 \$MC_JOG_JERK_GEO [<几何轴>]

其中 <几何轴> = 0, 1, 2

定向轴

手动运行定向轴时，可通过以下机床数据针对通道设定加加速度上限：

MD21158 \$MC_JOG_JERK_ORI [<定向轴>]

为了使 MD21158 生效，必须通过以下机床数据为定向轴手动运行使能针对通道的加加速度限制：

MD21159 \$MC_JOG_JERK_ORI_ENABLE == TRUE

文档:

功能手册之基本功能分册; 章节“加速度(B2)”

报警信息

手动运行几何轴/定向轴时, 下列情形下会触发报警 20062“轴已生效”:

- 轴已通过轴专用 PLC 接口在 JOG 方式下运行。
- 为旋转过的坐标系激活了框架, 且通过移动键在 JOG 方式下运行了框架涉及的另一根几何轴。

若尝试在 JOG 方式下运行未定义为几何轴的轴, 则会发出报警 20060“轴无法作为几何轴运行”。

3.11.2 主轴的手动运行

主轴的手动运行

在 JOG 方式下也可手动运行主轴。运行条件与手动运行进给轴时基本相同。在 JOG 方式下, 不仅可通过点按或者长按移动键来运行主轴, 也可以通过手轮来运行主轴。与机床轴类似, 此功能的选择和激活同样通过进给轴/主轴专用的 PLC 接口进行。进给轴专用机床数据也适用于主轴。

主轴运行方式

在定位方式(主轴处于位置闭环控制中)和开环方式下均可手动运行主轴。

JOG 速度

手动运行主轴时采用的速度可如下定义:

- 通过对所有主轴生效的通用设定数据:
SD41200 \$SN_JOG_SPIND_SET_VELO (主轴 JOG 转速)

或者

3.11 手动运行和手轮运行时的特点

- 通过以下机床数据:

MD32020 \$ _MA_JOG_VELO (基准 JOG 轴速度)

此机床数据仅在满足以下条件时才生效:

SD41110 \$SN_JOG_SET_VELO (JOG 轴速度) = 0

主轴的手动运行转速也考虑了当前齿轮档允许的最大转速。

文档:

功能手册之基本功能分册; 章节“主轴 (S1)”

速度修调

主轴的手动运行中, 主轴修调开关有效。

JOG 加速度

在转速开环控制方式和位置闭环控制方式下, 主轴通常具有多个齿轮档。主轴手动运行时的转速会遵循当前齿轮档允许的最大加速度。

文档:

功能手册之基本功能分册; 章节“主轴 (S1)”

PLC 接口信号

主轴手动运行中, NCK 和 PLC 间的 PLC 接口信号生效, 与机床轴的情形相似。

仅在主轴处于位置闭环控制中时, 接口信号:

DB31, ... DBX60.7 或 DBX60.6 (使用精准停或粗准停到达位置)
才置位。

主轴手动运行中要注意以下主轴专用的接口信号:

- 下列发送至主轴的 PLC 接口信号无效:
 - DB31, ... DBX17.6 (M3/M4 反转)
 - DB31, ... DBX18.6/7 (往复方向: 正转/反转)
 - DB31, ... DBX18.5 (往复使能)
 - DB31, ... DBX16.7 (删除 S 值)
- 下列由主轴发出的 PLC 接口信号不置位:
 - DB31, ... DBX83.7 (实际旋转方向: 正转)
 - DB31, ... DBX83.5 (主轴位于设定区域内)

3.11.3 监控

限制

下列限制对手动运行和手轮运行有效：

- 工作区域限制（轴必须已回参考点）
- 软件限位开关 1 和 2（轴必须已回参考点）
- 硬件限位开关

一旦轴到达首个有效限位，控制系统内部便会确保轴运行安全终止。其中的速度控制可以保证及时触发制动过程，使轴准确停止在限位上（例如软件限位开关）。只有在硬件限位开关响应时才会通过“快速停止”使轴制动。

一旦到达某个限位，系统便会发出报警（报警 16016，16017，16020，16021）之后控制系统内部会阻止轴继续越过限位。移动键和手轮在此方向上无效。

说明

为了确保软件限位开关和工作区域限制生效，轴必须事先执行回参考点。

若激活了手轮的 DRF 偏移功能，则会在 JOG 方式下的主处理中对该轴是否到达软件限位开关进行监控。也就是说，在到达软件限位开关前急动限制无效。轴到达软件限位开关后，按照 MD32300 \$MA_MAX_AX_ACCEL（最大轴加速度）进行减速。

有关工作区域限制、硬件和软件限位开关的更多说明请参见：

文档：

功能手册之基本功能分册；章节“轴监控，保护区（A3）”

轴的退回

轴可离开限位、回到允许的范围内运行。

说明

机床制造商

到达限位后轴如何退回由机床制造商决定。请注意机床制造商的文档！

最大速度和最大加速度

手动运行时采用的速度和加速度由调试人员针对各轴通过机床数据定义。控制系统会将生效的轴速度和加速度限制在定义的最大值以内。

3.11 手动运行和手轮运行时的特点

文档:

功能手册之基本功能分册; 章节“速度, 设定值/实际值系统, 闭环控制 (G2)”

功能手册之基本功能分册; 章节“加速度 (B2)”

3.11.4 其它

运行方式从 JOG 切换到 AUTO 或从 JOG 切换到 MDI

仅当通道的所有轴都达到“粗准停”状态时, 运行方式才可从 JOG 切换到 AUTO/MDI。

文档:

功能手册之基本功能分册; 章节“BAG, 通道, 程序运行, 复位特性 (K1)”

JOG 方式下的旋转进给率

在 JOG 运行方式下, 也可通过旋转进给率 (与采用 G95 的情形相似) 手动运行当前主轴。

旋转进给率通过以下设定数据激活:

SD41100 \$SN_JOG_REV_IS_ACTIVE (JOG: 旋转进给率/线性进给率)

采用的进给率 (单位 mm/rev) 通过以下方式设定:

- 通过通用设定数据定义:

SD41120 \$SN_JOG_REV_SET_VELO (JOG 旋转进给率)

- 通过轴专用机床数据定义:

MD32050 \$MA_JOG_REV_VELO (JOG 旋转进给率)

或在快速移动叠加中:

MD32040 \$MA_JOG_REV_VELO_RAPID (采用快速移动叠加时 JOG 旋转进给率),

SD41120 = 0 的条件下。

若未定义主主轴, 同时尝试在 JOG 方式下通过旋转进给率运行轴, 则会触发报警 20055 或针对几何轴触发报警 20065。

车床 X 轴

若将一根几何轴定义为“车床 X 轴”：

MD20100 \$MC_DIAMETER_AX_DEF（带“车床 X 轴”功能的几何轴）

且选择了半径编程，那么在 JOG 方式下运行时须注意下列特性：

- 连续运行：
和其他轴的连续手动运行没有区别。
- 断续运行：
车床 X 轴只运行选中步数的一半。例如在选中 INC10 时按下移动键，轴只运行 5 步。
- 通过手轮运行：
和断续运行一样，车床 X 轴只运行一半手轮脉冲换算出的行程。

文档：

功能手册之基本功能分册；章节“车床 X 轴（P1）”

3.12 数据表

3.12.1 机床数据

3.12.1.1 通用机床数据

编号	名称：\$MN_	描述
10000	AXCONF_MACHAX_NAME_TAB[n]	机床轴名称
10720	OPERATING_MODE_DEFAULT	上电后的缺省运行方式
10721	OPERATING_MODE_EXTENDED	上电后缺省运行方式的扩展设置
10735	JOG_MODE_MASK	JOG 运行方式设置
11300	JOG_INC_MODE_LEVELTRIGGRD	INC 和 REF 长按模式
11310	HANDWH_REVERSE	手轮反向阈值
11320	HANDWH_IMP_PER_LATCH[n]	手轮每转一个刻度输出的脉冲数
11324	HANDWH_VDI_REPRESENTATION	手轮编号（NC/PLC 接口）
11330	JOG_INCR_SIZE_TAB[n]	手动运行/手轮运行中的步数
11340	ENC_HANDWHEEL_SEGMENT_NR	第三个手轮：总线段
11342	ENC_HANDWHEEL_MODULE_NR	第三个手轮：逻辑驱动编号

3.12 数据表

编号	名称: \$MN_	描述
11344	ENC_HANDWHEEL_INPUT_NR	第三个手轮: 编码器接口
11346	HANDWH_TRUE_DISTANCE	手轮设定行程还是设定速度
11350	HANDWHEEL_SEGMENT[n]	手轮段
11351	HANDWHEEL_MODULE[n]	手轮模块
11352	HANDWHEEL_INPUT[n]	手轮连接
11353	HANDWHEEL_LOGIC_ADDRESS[n]	手轮插槽的逻辑地址 (STEP 7)
17900	VDI_FUNCTION_MASK	VDI 信号的功能位

3.12.1.2 通道专用机床数据

编号	名称: \$MC_	描述
20060	AXCONF_GEOAX_NAME_TAB	通道中的几何轴
20100	DIAMETER_AX_DEF	具有机床 X 轴功能的几何轴
20110	RESET_MODE_MASK	复位/程序结束后的缺省设置
20150	GCODE_RESET_VALUES	G 功能组的初始设置
20151	GCODE_RESET_MODE	G 功能组的复位特性
20360	TOOL_PARAMETER_DEF_MASK	刀具参数的定义
20620	HANDWH_GEOAX_MAX_INCR_SIZE	几何轴的限制
20622	HANDWH_GEOAX_MAX_INCR_VSIZE	轨迹速度叠加
20624	HANDWH_CHAN_STOP_COND	确定通道专用的手轮运行性能
20700	REFP_NC_START_LOCK	不带参考点的 NC 启动禁用
21150	JOG_VELO_RAPID_ORI	定向轴的 JOG 快速移动速度
21158	JOG_JERK_ORI	定向轴手动运行时的加加速度上限
21159	JOG_JERK_ORI_ENABLE	使能定向轴手动运行时的加加速度上限
21165	JOG_VELO_GEO	几何轴的基准 JOG 速度
21166	JOG_ACCEL_GEO	几何轴手动运行时的加速度上限
21168	JOG_JERK_GEO	几何轴手动运行时的加加速度上限

3.12.1.3 进给轴/主轴专用机床数据

编号	名称: \$MA_	描述
30450	IS_CONCURRENT_POS_AX	复位时的缺省设置: 中立轴或通道轴
30600	FIX_POINT_POS[n]	轴的固定点位置
30610	NUM_FIX_POINT_POS	轴固定点位置的个数
31090	JOG_INCR_WEIGHT	手动运行/手轮运行中的步长)
32000	MAX_AX_VELO	最大轴速度
32010	JOG_VELO_RAPID	JOG 快速移动速度
32020	JOG_VELO	JOG 轴速度
32040	JOG_REV_VELO_RAPID	JOG 带快速移动速度的旋转进给率
32050	JOG_REV_VELO	JOG 旋转进给率
32060	POS_AX_VELO	定位轴速度的初始设置
32080	HANDWH_MAX_INCR_SIZE	倍数的上限
32082	HANDWH_MAX_INCR_VELO_SIZE	倍数上限, 用于速度叠加
32084	HANDWH_STOP_COND	手轮运行性能
32090	HANDWH_VELO_OVERLAY_FACTOR	JOG 速度与手轮速度 (DRF) 之比
32300	MAX_AX_ACCEL	最大轴加速度
32301	JOG_MAX_ACCEL	JOG 最大轴加速度
32420	JOG_AND_POS_JERK_ENABLE	单轴加加速度上限的缺省设置
32430	JOG_AND_POS_MAX_JERK	单轴运动中的单轴加加速度上限
32431	MAX_AX_JERK	轨迹运动中的单轴加加速度上限
32436	JOG_MAX_JERK	JOG 单轴加加速度上限
34210	ENC_REFP_STATE	测量系统的状态
35130	GEAR_STEP_MAX_VELO_LIMIT[n]	齿轮档允许的最大转速

3.12 数据表

3.12.2 设定数据

3.12.2.1 通用设定数据

编号	名称: \$SN_	描述
41010	JOG_VAR_INCR_SIZE	INC/手轮可变增量大小
41050	JOG_CONT_MODE_LEVELTRIGGRD	连续手动运行采取长按模式
41100	JOG_REV_IS_ACTIVE	JOG 激活时的旋转进给率
41110	JOG_SET_VELO	线性轴 JOG 速度 (G94 时)
41120	JOG_REV_SET_VELO	JOG 速度 (G95 时)
41130	JOG_ROT_AX_SET_VELO	回转轴的 JOG 速度
41200	JOG_SPIND_SET_VELO	主轴的 JOG 速度

3.12.3 信号

3.12.3.1 从 NC 发出的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
手轮 1 运行	DB10.DBB68	DB2700.DBB12
手轮 2 运行	DB10.DBB69	DB2700.DBB13
手轮 3 运行	DB10.DBB70	-
手轮 4 运行	DB10.DBB242	-
手轮 5 运行	DB10.DBB243	-
手轮 6 运行	DB10.DBB244	-
以太网手轮静止	DB10.DBB245	-

3.12.3.2 发送至 BAG 的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
BAG 1: JOG 运行方式	DB11.DBX0.2	DB3000.DBX0.2
BAG2: JOG 运行方式	DB11.DBX20.2	-

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
BAG3: JOG 运行方式	DB11.DBX40.2	-
BAG4: JOG 运行方式	DB11.DBX60.2	-
BAG5: JOG 运行方式	DB11.DBX80.2	-
BAG6: JOG 运行方式	DB11.DBX100.2	-
BAG7: JOG 运行方式	DB11.DBX120.2	-
BAG8: JOG 运行方式	DB11.DBX140.2	-
BAG9: JOG 运行方式	DB11.DBX160.2	-
BAG10: JOG 运行方式	DB11.DBX180.2	-

3.12.3.3 从 BAG 发出的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
BAG 1: JOG 运行方式生效	DB11.DBX6.2	DB3100.DBX0.2
BAG2: JOG 运行方式生效	DB11.DBX26.2	-
BAG3: JOG 运行方式生效	DB11.DBX46.2	-
BAG4: JOG 运行方式生效	DB11.DBX66.2	-
BAG5: JOG 运行方式生效	DB11.DBX86.2	-
BAG6: JOG 运行方式生效	DB11.DBX106.2	-
BAG7: JOG 运行方式生效	DB11.DBX126.2	-
BAG8: JOG 运行方式生效	DB11.DBX146.2	-
BAG9: JOG 运行方式生效	DB11.DBX166.2	-
BAG10: JOG 运行方式生效	DB11.DBX186.2	-

3.12.3.4 发送至通道的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
激活 DRF	DB21, ...DBX0.3	DB3200.DBX0.3
几何轴 1: 激活手轮	DB21, ...DBX12.0-2	DB3200.DBX1000.0-2
几何轴 1: 移动键禁用	DB21, ...DBX12.4	DB3200.DBX1000.4
几何轴 1: 快速移动速度叠加	DB21, ...DBX12.5	DB3200.DBX1000.5
几何轴 1: 移动键 -/+	DB21, ...DBX12.6-7	DB3200.DBX1000.6-7

3.12 数据表

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
几何轴 1: 机床功能 1 INC ... 可变 INC	DB21, ...DBX13.0-5	DB3200.DBX1001.0-5
几何轴 1: 取反手轮旋转方向	DB21, ...DBX15.0	DB3200.DBX1003.0
几何轴 2: 激活手轮	DB21, ...DBX16.0-2	DB3200.DBX1004.0-2
几何轴 2: 移动键禁用	DB21, ...DBX16.4	DB3200.DBX1004.4
几何轴 2: 快速移动速度叠加	DB21, ...DBX16.5	DB3200.DBX1004.5
几何轴 2: 移动键 -/+	DB21, ...DBX16.6-7	DB3200.DBX1004.6-7
几何轴 2: 机床功能 1 INC ... 可变 INC	DB21, ...DBX17.0-5	DB3200.DBX1005.0-5
几何轴 2: 取反手轮旋转方向	DB21, ...DBX19.0	DB3200.DBX1007.0
几何轴 3: 激活手轮	DB21, ...DBX20.0-2	DB3200.DBX1008.0-2
几何轴 3: 移动键禁用	DB21, ...DBX20.4	DB3200.DBX1008.4
几何轴 3: 快速移动速度叠加	DB21, ...DBX20.5	DB3200.DBX1008.5
几何轴 3: 移动键 -/+	DB21, ...DBX20.6-7	DB3200.DBX1008.6-7
几何轴 3: 机床功能 1 INC ... 可变 INC	DB21, ...DBX21.0-5	DB3200.DBX1009.0-5
几何轴 3: 取反手轮旋转方向	DB21, ...DBX23.0	DB3200.DBX1011.0
激活轮廓手轮	DB21, ...DBX30.0-2	DB3200.DBX14.0-2
激活轮廓手轮仿真	DB21, ...DBX30.3	DB3200.DBX14.3
轮廓手轮仿真负方向	DB21, ...DBX30.4	DB3200.DBX14.4
取反轮廓手轮旋转方向	DB21, ...DBX31.5	DB3200.DBX15.5
定向轴 1: 激活手轮	DB21, ...DBX320.0-2	-
定向轴 1: 移动键禁用	DB21, ...DBX320.4	-
定向轴 1: 快速移动速度叠加	DB21, ...DBX320.5	-
定向轴 1: 移动键 -/+	DB21, ...DBX320.6-7	-
定向轴 1:		
机床功能 1 INC ... 可变 INC	DB21, ...DBX321.0-5	-
定向轴 1: 取反手轮旋转方向	DB21, ...DBX323.0	-
定向轴 2: 激活手轮	DB21, ...DBX324.0-2	-
定向轴 2: 移动键禁用	DB21, ...DBX324.4	-
定向轴 2: 快速移动速度叠加	DB21, ...DBX324.5	-
定向轴 2: 移动键 -/+	DB21, ...DBX324.6-7	-
定向轴 2:		
机床功能 1 INC ... 可变 INC	DB21, ...DBX325.0-5	-

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
定向轴 2: 取反手轮旋转方向	DB21, ...DBX327.0	-
定向轴 3: 激活手轮	DB21, ...DBX328.0-2	-
定向轴 3: 移动键禁用	DB21, ...DBX328.4	-
定向轴 3: 快速移动速度叠加	DB21, ...DBX328.5	-
定向轴 3: 移动键 -/+	DB21, ...DBX328.6-7	-
定向轴 3: 机床功能 1 INC ... 可变 INC	DB21, ...DBX329.0-5	-
定向轴 3: 取反手轮旋转方向	DB21, ...DBX331.0	-

3.12.3.5 来自通道的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
DRF 已选择	DB21, ...DBX24.3	DB1700.DBX0.3
手轮叠加生效 (轨迹轴)	DB21, ...DBX33.3	DB3300.DBX1.3
轮廓手轮生效	DB21, ...DBX37.0-2	DB3300.DBX5.0-2
取反轮廓手轮旋转方向生效	DB21, ...DBX39.5	DB3300.DBX7.5
几何轴 1: 手轮生效	DB21, ...DBX40.0-2	DB3300.DBX1000.0-1
运行请求 -/+	DB21, ...DBX40.4-5	DB3300.DBX1000.4-5
运行指令 -/+	DB21, ...DBX40.6-7	DB3300.DBX1000.6-7
机床功能 1 INC ... 可变 INC 生效	DB21, ...DBX41.0-5	DB3300.DBX1001.0-5
取反手轮旋转方向生效	DB21, ...DBX43.0	DB3300.DBX1003.0
几何轴 2: 手轮生效	DB21, ...DBX46.0-2	DB3300.DBX1004.0-1
运行请求 -/+	DB21, ...DBX46.4-5	DB3300.DBX1004.4-5
运行指令 -/+	DB21, ...DBX46.6-7	DB3300.DBX1004.6-7
机床功能 1 INC ... 可变 INC 生效	DB21, ...DBX47.0-5	DB3300.DBX1005.0-5
取反手轮旋转方向生效	DB21, ...DBX49.0	DB3300.DBX1007.0
几何轴 3: 手轮生效	DB21, ...DBX52.0-2	DB3300.DBX1008.0-1
运行请求 -/+	DB21, ...DBX52.4-5	DB3300.DBX1008.4-5

3.12 数据表

信号名称		SINUMERIK 840D sl	SINUMERIK 828D
	运行指令 -/+	DB21, ...DBX52.6-7	DB3300.DBX1008.6-7
	机床功能 1 INC ... 可变 INC 生效	DB21, ...DBX53.0-5	DB3300.DBX1009.0-5
	取反手轮旋转方向生效	DB21, ...DBX55.0	DB3300.DBX1011.0
定向轴 1	手轮生效	DB21, ...DBX332.0-2	-
	运行请求 -/+	DB21, ...DBX332.4-5	-
	运行指令 -/+	DB21, ...DBX332.6-7	-
	取反手轮旋转方向生效	DB21, ...DBX335.0	-
定向轴 2	手轮生效	DB21, ...DBX336.0-2	-
	运行请求 -/+	DB21, ...DBX336.4-5	-
	运行指令 -/+	DB21, ...DBX336.6-7	-
	取反手轮旋转方向生效	DB21, ...DBX339.0	-
定向轴 3	手轮生效	DB21, ...DBX340.0-2	-
	运行请求 -/+	DB21, ...DBX340.4-5	-
	运行指令 -/+	DB21, ...DBX340.6-7	-
	取反手轮旋转方向生效	DB21, ...DBX343.0	-
手动退刀生效		DB21, ...DBX377.4	DB3300, DBX4005.4
退刀数据存在		DB21, ...DBX377.5	DB3300, DBX4005.5

3.12.3.6 到进给轴/主轴的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
进给修调	DB31, ...DBB0	DB380x.DBB0
补偿有效	DB31, ...DBX1.7	DB380x.DBX1.7
删除剩余行程/主轴复位	DB31, ...DBX2.2	DB380x.DBX2.2
激活手轮	DB31, ...DBX4.0-2	DB380x.DBX4.0-2
移动键禁用	DB31, ...DBX4.4	DB380x.DBX4.4
快速移动速度叠加	DB31, ...DBX4.5	DB380x.DBX4.5

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
移动键 -/+	DB31, ...DBX4.6-7	DB380x.DBX4.6-7
机床功能 1 INC ... 可变 INC	DB31, ...DBX5.0-5	DB380x.DBX5.0-5
取反手轮旋转方向	DB31, ...DBX7.0	DB380x.DBX7.0
手动运行至固定点	DB31, ...DBX13.0-2	DB380x.DBX1001.0-2

3.12.3.7 来自进给轴/主轴的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
粗/精准停到达位置	DB31, ...DBX60.6-7	DB390x.DBX0.6-7
手轮叠加生效	DB31, ...DBX62.1	DB390x.DBX2.1
手轮生效	DB31, ...DBX64.0-2	DB390x.DBX4.0-2
运行请求 -/+	DB31, ...DBX64.4-5	DB390x.DBX4.4-5
运行指令 -/+	DB31, ...DBX64.6-7	DB390x.DBX4.6-7
机床功能 1 INC ... 可变 INC 生效	DB31, ...DBX65.0-5	DB390x.DBX5.0-5
测量系统 1/2 的位置已恢复	DB31, ...DBX71.4-5	DB390x.DBX11.4-5
手动运行至固定点生效	DB31, ...DBX75.0-2	DB390x.DBX1001.0-2
已手动运行至固定点	DB31, ...DBX75.3-5	DB390x.DBX1001.3-5
取反手轮旋转方向生效	DB31, ...DBX67.0	DB390x.DBX7.0

3.12.4 系统变量

3.12.4.1 系统变量

名称	描述
\$AA_POSRES	轴状态“位置已恢复”

3.12 数据表

3.12.5 OPI 变量

3.12.5.1 OPI 变量

名称	描述
retractState, 位 0	退刀数据存在
retractState, 位 1	手动退刀生效
retractState, 位 2 - 3	退刀轴
aaPosRes	轴状态 “位置已恢复”

K3: 补偿

4.1 引言

精度误差

机床的加工精度会受到几何误差、切削力误差、测量系统误差的影响，尤其是在加工大型工件时，温度变化和切削力通常会大大降低加工精度。

补偿功能

其中的一部分误差通常可以在调试机床和运行机床期间借助位移编码器和辅助传感器测出并由系统进行补偿。为此，现代化的数控系统会为各轴提供各种有效的补偿功能。

可供使用的补偿功能如下：

- 温度补偿
- 间隙补偿
 - 机械间隙补偿
 - 动态间隙补偿
- 插补补偿
 - 丝杠螺距误差补偿和测量系统误差补偿
 - 垂度补偿和角度误差补偿
- 动态前馈控制（跟随误差补偿）
- 摩擦补偿（象限误差补偿）
 - 常规摩擦补偿
- 电子配重

参数设置

通过轴专用机床数据可为每台机床设置个性化的补偿功能。

生效方式

一旦输入数据可用，补偿便会在控制系统的所有运行方式中生效。但所有需要获得实际绝对位置信息的补偿在轴回参考点后才激活。

4.2 温度补偿

位置显示

正常的实际值和设定值位置显示区不考虑补偿值，显示的是“理想机床”的位置值。补偿值在操作区域“诊断”的窗口“进给轴/主轴信息”中显示。

4.2 温度补偿

4.2.1 功能说明

温度影响造成的变形

传动装置发热或环境影响（例如太阳照射、气流）可能会导致机床床身和其他部件受热膨胀。膨胀幅度取决于温度、机床部件的导热性等因素。

影响

由于机床部件会受热膨胀，因此各轴的实际位置会随温度变化。这种热误差会对工件的加工精度产生不利影响。

温度补偿

激活“温度补偿”功能，可针对特定轴对温度变化造成的实际值变化进行补偿。

传感设备

除了现有位移编码器外，通常还需要多个温度传感器来检测温度，以确定热误差曲线。由于热误差形成较为缓慢，因此 PLC 对热误差曲线的测定和处理可能要持续数分钟。

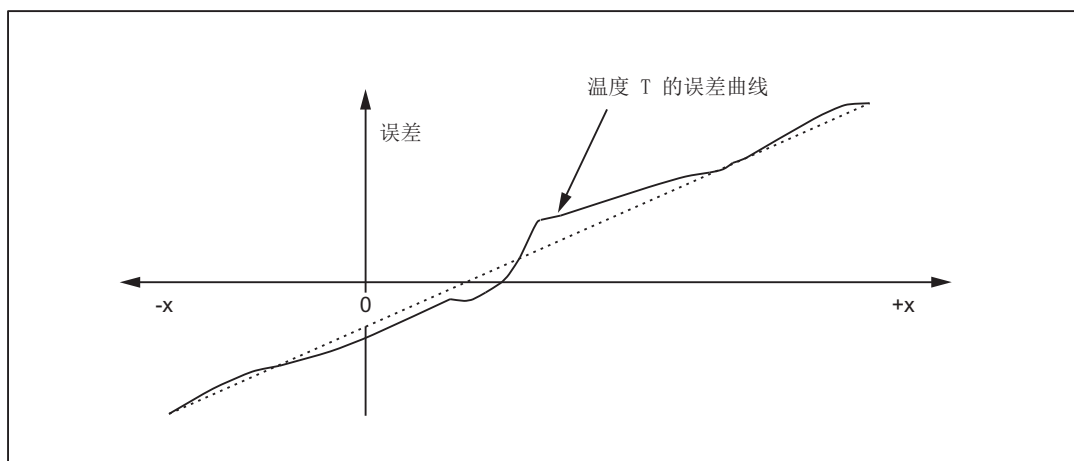
误差曲线

为进行温度补偿，首先需要实测指定温度下（T）轴行程内各个位置上的误差，这样可得出一条该温度的热误差曲线。然后以相同方式获得其他温度下的误差曲线。

误差曲线走势

选择一个轴位置参考点 P_0 后，便可观察温度变化时该参考点的位移情况，即“温度补偿中的位置无关分量”；由于热膨，其他位置点上会发生额外的位移，该位移随着与参考点的间距变大而变大，即“温度补偿中的位置相关分量”。

通常情况下一条直线便可足够精确地表示指定温度 T 下的误差曲线，直线的斜率和参考点位置取决于温度：



补偿公式

补偿值 ΔK_x 由该轴的实际位置 P_x 和温度 T 按以下公式计算得出：

$$\Delta K_x = K_0(T) + \tan\beta(T) * (P_x - P_0)$$

含义：

- ΔK_x : 位置 P_x 上的温度补偿值
- K_0 : 与位置无关的温度补偿值
- P_x : 轴的实际位置
- P_0 : 轴参考点位置
- $\tan\beta$: 位置相关温度补偿的系数（等同于误差曲线的斜率）

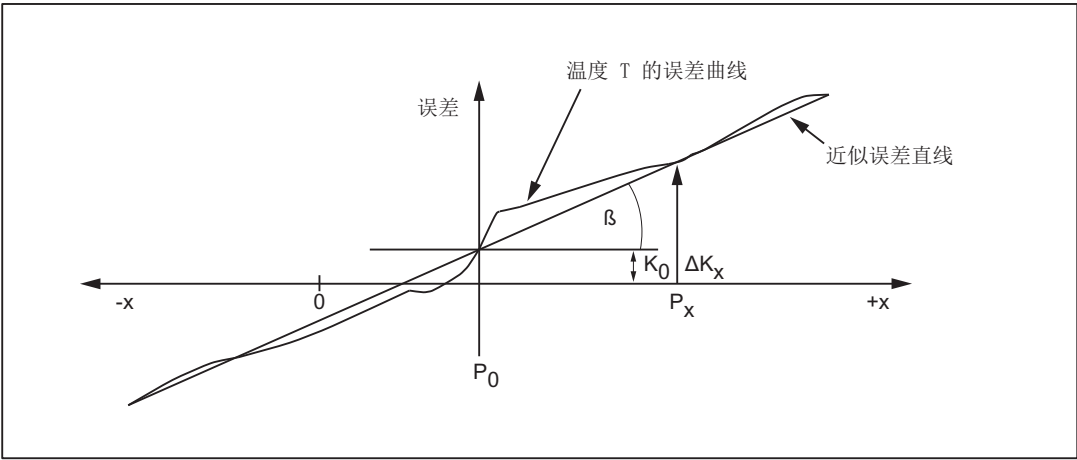


图 4-1 温度补偿的近似误差直线

生效方式

为了使温度补偿生效，必须满足下列前提条件：

- 1. 已选择补偿类型（MD32750，参见“调试 (页 245)”）。
- 2. 补偿类型参数已经过设定（参见“调试 (页 245)”）。
- 3. 轴已执行回参考点：
DB31, ... DBX60.4 或 60.5 = 1（已回参考点/已同步 1 或 2）

一旦满足这些条件，在所有运行方式中系统都会将当前位置实际值对应的温度补偿值附加在设定值上，然后运行机床轴。补偿值 ΔK_x 为正值时，轴朝负向运行。

之后的运行中，若参考点位置由于超出编码器频率等原因丢失（DB31, ... DBX60.4 或 60.5 = 0），则会取消补偿。

周期

补偿值以插补周期计算。

显示

当前位置对应的总补偿值（温度补偿值和垂度补偿值之和）在操作区域“诊断”下的“进给轴/主轴信息”窗口中显示。

温度变化时的参数调整

近似误差直线仅适用于当前的温度值，温度上升或下降时须重新测定误差曲线，并将参数重新传输至 NCK。只有这样才能保证对热误差补偿的准确性。

温度 T 变化时，和它关联的参数值 (K_0 、 $\tan\beta$ 和 P_0) 也会跟着变化，因此始终可通过 PLC 或同步动作对这些数值进行改写。

机床制造商因而可通过 PLC 用户程序反映轴位置和温度值之间的数学关联和工艺关联，并计算出对应的温度补偿参数。温度参数通过服务 FB2 (GET) “读取数据” 和 FB3 (PUT) “写入数据” 传输至 NCK。

有关处理和提供 FB2 和 FB3 的更多详细信息请见：

文档：

功能手册之基本功能分册：章节“PLC 基本程序 (P3)”

平滑补偿值

当补偿值超出每个插补周期的上限 (MD32760, 参见“调试 (页 245)”) 时，控制系统内部会自动将补偿值分配到多个插补周期中进行计算，以防止温度补偿参数的跃变导致机床过载或监控功能响应。

4.2.2 调试

温度相关参数

可为每根轴定义各种温度下的误差曲线。对每条误差曲线都必须确定下列参数，并作为设定数据保存：

- 与位置无关的温度补偿值 K_0 ：
SD43900 \$SA_TEMP_COMP_ABS_VALUE
- 参考点位置 P_0 ，用于位置相关的温度补偿：
SD43920 \$SA_TEMP_COMP_REF_POSITION
- 斜率 $\tan\beta$ ，用于位置相关的温度补偿：
SD43910 \$SA_TEMP_COMP_SLOPE

温度补偿类型和激活方式

温度补偿类型的选择及激活通过以下轴专用机床数据进行：

MD32750 \$MA_TEMP_COMP_TYPE (温度补偿类型)

位	值	含义	对应参数
0		与位置无关的温度补偿	SD43900
	0	未生效	
	1	生效	
1		位置相关的温度补偿	SD43920, SD43910
	0	未生效	
	1	生效	
2		刀具方向上的温度补偿	MD20390 \$MC_TOOL_TEMP_COMP_ON (激活刀具方向上的温度补偿)
	0	未生效	
	1	生效	

每个插补周期的最大补偿值

每个插补周期的补偿值上限，即一个插补周期内可通过温度补偿运行的最大行程，通过以下机床数据限制：

MD32760 \$MA_COMP_ADD_VELO_FACTOR（补偿引起的过速）

输入值是系数，是相对于最大轴速度（MD32000 \$MA_MAX_AX_VELO）的百分比值。

MD32760 还用作温度补偿误差直线斜率（tan β）的上限。

4.2.3 示例

4.2.3.1 为车床的 Z 轴调试温度补偿

下面以车床的 Z 轴为例介绍温度补偿的调试步骤。

测定 Z 轴的误差曲线

为测定 Z 轴的热误差曲线，执行下列步骤：

- 使 Z 轴在整个行程（例如：从 500 mm 到 1500 mm）内移动，使其均匀受热
- 以 100 mm 为步距测量轴位置
- 在滚珠丝杠处测量当前温度
- 每 20 分钟执行一次测量循环

根据测出的数据得出位置和温度值之间的数学关联和工艺关联，并推导出温度补偿参数。
下图是不同温度下的误差曲线，其以 NC 显示的 Z 轴实际位置为基准。

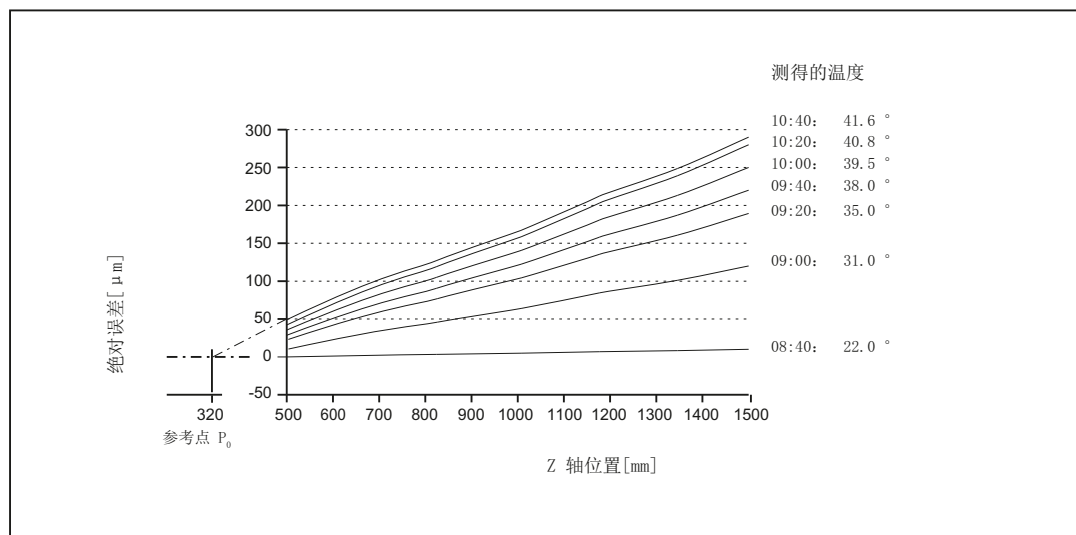


图 4-2 测定的 Z 轴误差曲线

确定参数

下面根据测量结果（见上图）来确定温度补偿参数。

参考点位置 P_0

如上图所示，参考点位置 P_0 的设置基本分为两种：

1. $P_0 = 0$ ，与位置无关的温度补偿值 $K_0 \neq 0$
2. $P_0 \neq 0$ ，位置相关的温度补偿值 $K_0 = 0$

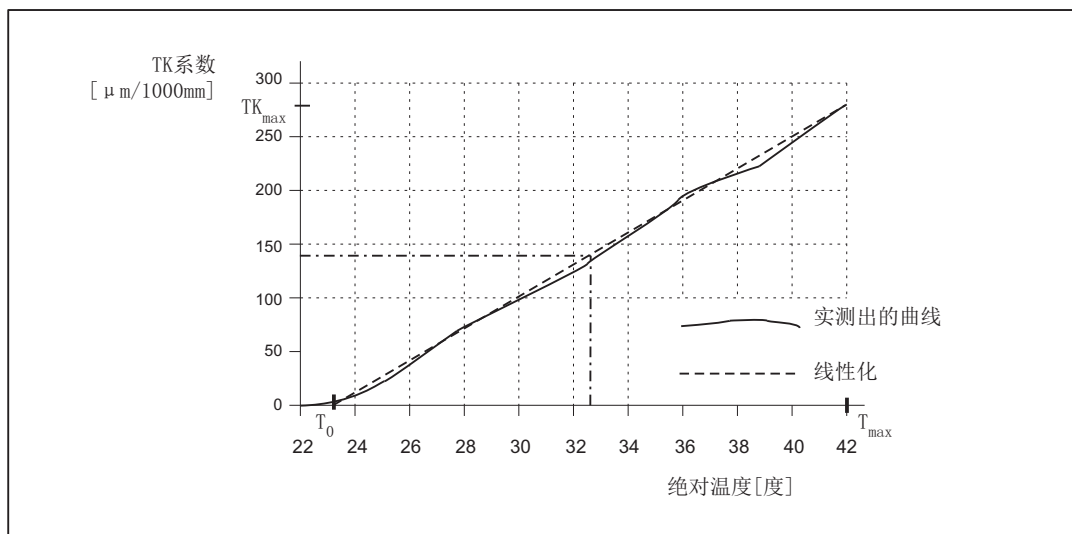
本例选择第 2 种，此时与位置无关的温度补偿值总是为 0。这样一来，温度补偿值只包含位置相关分量。

得到以下参数：

- MD32750 \$MA_TEMP_COMP_TYPE = 2
(仅位置相关温度补偿生效)
- $P_0 = 320 \text{ mm} \rightarrow \text{SD43920 } \$\text{SA\_TEMP\_COMP\_REF\_POSITION} = 320$
(温度相关温度补偿的参考点位置)

系数 $\tan\beta$ (T)

为了测定位置相关温度补偿系数 $\tan\beta$ 和温度的关联，误差曲线斜率和实测温度之间的关系用函数图来表示：

图 4-3 实测温度 T 和系数 $\tan\beta$ 的函数图

经过线性处理后，系数 $\tan\beta$ 与温度 T 的关联如下：

$$\tan\beta(T) = (T - T_0) * TK_{\max} * 10^{-6} / (T_{\max} - T_0)$$

其中

T_0 = 位置相关误差为 0 时的温度；[度]

$T_{\max}$ = 测得的最高温度；[度]

$TK_{\max}$ = $T_{\max}$ 下的温度系数；[$\mu\text{m}/1000\text{ mm}$]

根据上图中的数值便可得出：

$$T_0 = 23^\circ$$

$$T_{\max} = 42^\circ$$

$$TK_{\max} = 270\ \mu\text{m}/1000\text{ mm}$$

从而可推导出 $\tan\beta$ (T)：

$$\begin{aligned}\tan\beta(T) &= (T - 23)\text{ [度]} * 270\text{ [}\mu\text{m}/1000\text{ mm]} * 10^{-6} / (42 - 23)\text{ [度]} \\ &= (T - 23)\text{ [度]} * 14.21\text{ [}\mu\text{m}/1000\text{ mm]} * 10^{-6}\end{aligned}$$

示例：

温度 $T = 32.3$ 度时， $\tan\beta = 0.000132$

PLC 用户程序

在 PLC 用户程序中，必须按照上面的公式计算出实测温度对应的系数 $\tan\beta(T)$ ，然后将计算结果写入 NCK 的以下设定数据：

SD43910 \$SA_TEMP_COMP_SLOPE（位置相关温度补偿的斜率）

根据上面的示例：

SD43910 \$SA_TEMP_COMP_SLOPE = 0.000132

4.3 间隙补偿

4.3.1 机械间隙补偿

4.3.1.1 功能说明

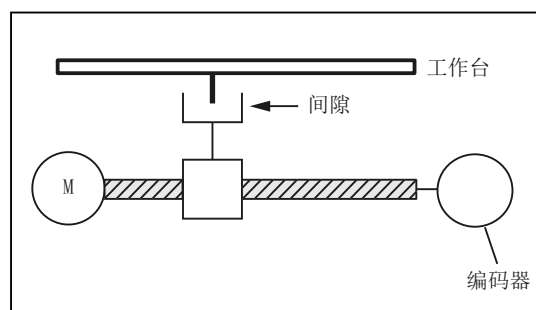
机械间隙

通常，在运动的机床部件和其传动部件（例如滚珠丝杠）之间传递力时会出现较小的间隙，因为完全无间隙的机械会导致过高的磨损。此外机床部件和测量系统之间也存在间隙。

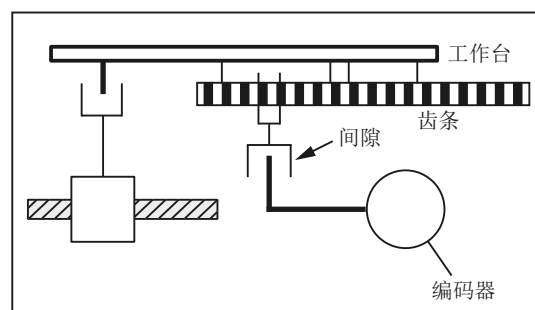
影响

进给轴/主轴采用间接测量系统的情况下，机械间隙会导致运行行程出错。比如，一进给轴在换向时少走或多走了一个间隙值：

正向间隙(正常情况)



负向间隙



编码器快于机床部件(比如工作台)。这样一来编码器滞后于机床部件（例如工作台）。这编码器检测出的实际位置也会先于工作台的实样一来工作台的实际位置会先于编码器检测出实际位置，所以工作台少走一个间隙值。 的实际位置，所以工作台多走一个间隙值。

补偿

为了实现对这种机械间隙的补偿，每次进给轴/主轴换向时系统会在轴实际值基础上增加/减少一个调试时设定的轴间隙值（MD32450，参见“调试 (页 250)”）。

生效方式

在回参考点后，机械间隙补偿在所有的运行模式下有效。

显示

当前位置对应的总补偿值（丝杠螺距误差补偿值、机械间隙补偿值和动态间隙补偿值之和）在操作区域“诊断”下的“进给轴/主轴信息”窗口中显示。

4.3.1.2 调试

反向间隙

机械反向间隙在高速下通过加工台上安装的测量系统测量。

所测得的补偿值会记录在以下轴专用机床数据中：

MD32450 \$MA_BACKLASH（反向间隙）

正向间隙（正常情况）必须输入正的补偿值，正向间隙则须输入负值。

在机床数据设置生效（NewConfig）后，系统会重复测量并检查机械间隙补偿的效果。

说明

可使用 HMI 中集成的圆度测试显示故障。

第 2 测量系统

若存在进给轴/主轴的第 2 测量系统，则同样需要为该系统输入间隙补偿值。第 2 测量系统安装地点不同，因此有可能会产生有别于第 1 测量系统的反向间隙。

在切换测量系统时对应的补偿值会自动激活。

反向间隙的权重系数

反向间隙可按照参数组设置一个权重系数。

MD32452 \$MA_BACKLASH_FACTOR（反向间隙的权重系数）

取值范围: 0.01 ... 100.0

缺省值: 1.0

应用: 用于补偿齿轮间隙。

位置实际值切换时的最大公差

用户可以将间隙补偿值分成多部分来补偿对应轴。这样可避免进给轴上的设定值跃变导致相应故障。

间隙补偿值 (MD32450) 的划分可通过以下轴专用机床数据定义:

MD36500 \$MA_ENC_CHANGE_TOL (位置实际值切换时的最大公差)

需要注意的是, 在 $\langle n \rangle$ 个伺服周期后系统才会计算间隙补偿值 ($\langle n \rangle = \text{MD32450} / \text{MD36500}$)。间隔过长可能会导致静态监控报警。若 $\text{MD36500} > \text{MD32450}$, 则会在一个伺服周期内执行补偿。

4.3.2 动态间隙补偿

4.3.2.1 功能说明

动态间隙

在采用滑动导轨的机床上可能会存在动态间隙。根据轴运行到终点的动态特性 (速度、加加速度等), 机床滑块可能会到达所编写的终点位置, 或在此之前便由于静摩擦停止。此时产生的位置误差为方向对称。

补偿

为补偿这种动态间隙, 系统会根据轴的运行方向激活一半大小的、有符号的补偿值 (MD32456, 参见“调试 (页 252)”)。补偿值以斜坡形式附加。

激活

动态间隙补偿只在必要情形下由 PLC 激活:

DB31, ... DBX25.0 (激活动态间隙补偿)

说明

要求激活动态间隙补偿的情形由机床制造商在 PLC 用户程序中定义。在通过 G1、JOG 模式下或通过手轮运行轴时可能会存在这些情形。

成功激活后，NC 会通过以下接口信号向 PLC 发送反馈信息：

DB31, ... DBX102.0 (动态间隙补偿生效)

前提条件

待补偿的轴必须已执行回参考点。

显示

当前位置对应的总补偿值（丝杠螺距误差补偿值、机械间隙补偿值和动态间隙补偿值之和）在操作区域“诊断”下的“进给轴/主轴信息”窗口中显示。

4.3.2.2 调试

动态间隙补偿的补偿值

在高速条件下测得机械反向间隙并通过 MD32450 \$MA_BACKLASH 输入补偿值后（参见机械间隙补偿的“调试 (页 250)”），系统会在低速条件下重复相同的测量步骤。此时测得的间隙即为动态间隙补偿的补偿值。该值记录在以下机床数据中：

MD32456 \$MA_BACKLASH_DYN (动态间隙补偿的补偿值)

对补偿值变化的限制

动态间隙补偿的补偿值以斜坡形式附加。斜坡通过以下机床数据定义：

MD32457 \$MA_BACKLASH_DYN_MAX_VELO (最大动态间隙补偿值变化速率)

值是相对于最大进给轴速度 (MD32000 \$MA_MAX_AX_VELO) 的百分比值。缺省值为 1%。

4.4 插补补偿

4.4.1 一般特性

功能

通过“插补补偿”可对位置相关的几何误差（丝杠螺距误差和测量系统误差、垂度误差和角度误差）进行修正。

和位置相关的补偿值在调试时通过测量系统确定，保存到一张补偿表中。轴运行期间，系统会利用控制点进行线性插补运算，从而修正实际位置。

方法

“插补补偿”细分为以下两种补偿方法：

- 丝杠螺距误差和测量系统误差的补偿
- 垂度和角度误差的补偿

这两种补偿方法的很多特性都相同，因此下文中将一同描述。

术语

“插补补偿”涉及的重要术语包括：

- 补偿值
位置编码器实测的轴位置和编程的轴位置（= 理想机床的轴位置）间的差值。补偿值通常也称为修正值。
- 参照轴
在计算补偿值时需要参照该轴的设定位置 and 实际位置。
- 补偿轴
设定位置或实际位置可通过补偿值进行修改的轴。
- 控制点
参照轴的位置和补偿轴对应的补偿值。
- 补偿表
包含控制点和补偿值的表格（见下文）
- 补偿关联
参照轴和以其为基准的补偿轴的对应关系，以及与对应补偿表的关联。

补偿表

几何误差会对工件的加工精度产生直接影响，因此必须通过对应的与位置相关的补偿值来进行补偿。补偿值根据测出的误差曲线来确定，并在调试时以补偿表的形式输入控制系统。每个补偿关联必须建立一个单独的补偿表。

补偿表的输入

首先需要通过机床数据定义补偿表的大小，即控制点的数量。下一次上电后 NC 会生成补偿表，补偿值为 0。

补偿值以及附加的表参数通过特殊的系统变量输入补偿表。可通过两种不同的方式执行载入补偿表：

- 通过启动包含参数值的 NC 程序。
- 通过将补偿表从一台外部计算机传输至控制系统。

说明

仅在相应的补偿功能未生效时，才可载入补偿表：

- MD32700 \$MA_ENC_COMP_ENABLE（插补补偿）= 0
- MD32710 \$MA_CEC_ENABLE（垂度补偿使能）= 0

补偿数据保存在静态用户存储器中，因此这些值在控制系统关闭后仍得以保留。根据需求可对这些数值执行更新（例如因为机床老化重新测量后）。

说明

以下机床数据变化时：

- MD18342 \$MN_MM_CEC_MAX_POINTS
- MD38000 \$MA_MM_ENC_COMP_MAX_POINTS

会在下一次系统启动时格式化静态用户存储器（参见“S7:存储器配置 (页 795)”章节）。

控制点之间的线性插补

由起点和终点确定的补偿范围可依据误差曲线分为多个相同大小的子段（见下图）。在下图中，子段分界的实际位置称为“控制点”。调试时必须为每个控制点输入对应的补偿值。

一个子段内各个位置的补偿值是通过**线性插补**该子段的 2 个控制点的补偿值得出（即通过一条直线连接相邻控制点）。

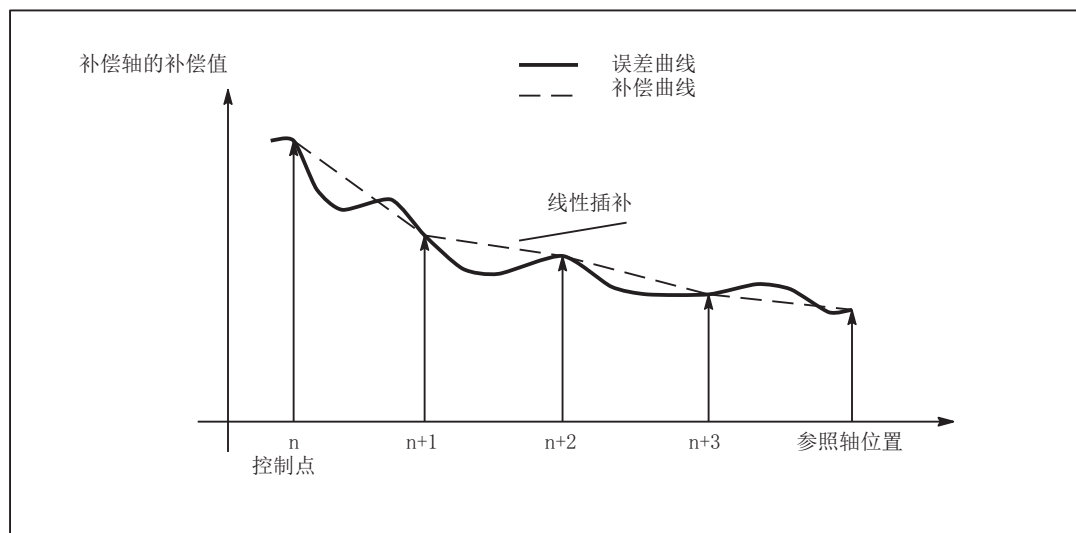


图 4-4 控制点之间的线性插补

参考点上的补偿值

建立补偿表时，参考点上的补偿值须为“0”。

4.4.2 对丝杠螺距误差和测量系统误差的补偿

4.4.2.1 功能说明

丝杠螺距误差和测量系统误差

在数控机床上，“间接测量”的工作原理要求在轴行程范围内的任何一个位置上，滚珠丝杠的螺距都是保持恒定的，从而可以从丝杠的位置推导出轴的实际位置，这是最理想的情况。但是丝杠的制造工艺会导致或多或少的几何误差（即所谓的丝杠螺距误差）。

。另外，还有使用的测量系统类型（不同的刻度盘）、测量系统的安装位置（统称为测量系统误差）以及其他各种机床误差。

补偿

在“测量系统误差补偿”（下文称为“MSEC”）时，参照轴和补偿轴总是相同。因此该补偿为**单轴补偿**，不需要在补偿表内定义参照轴和补偿轴。

说明

丝杠螺距误差补偿（LEC）是测量系统误差补偿的一部分。

在进行 MSEC 时，系统按插补周期用对应的轴补偿值对实际位置进行调整，然后立即运行机床轴。补偿值为正时，轴会在反向上移动该补偿值。

补偿值的大小不受限，也不受监控。因此，为了避免补偿使机床轴的速度或加速度超过极限值，应选取较小的补偿值。否则其他进给轴监控会触发报警信息（例如轮廓监控、转速设定值限制）。

若待补偿的轴还配备了第 2 个位置测量系统，则须为每个测量系统创建和激活独立的补偿表。在两个测量系统间切换时系统会自动使用对应的表格。

前提条件/生效方式

MSEC 仅在满足下列前提条件时才生效：

- 补偿值存储在静态用户存储器中并在上电后生效。
- 相关机床轴上激活了该功能：
MD32700 \$MA_ENC_COMP_ENABLE [**<e>**] = 1

其 **<e>** = 位置测量系统
中： **<e> = 0** 测量系统 1
 <e> = 1 测量系统 2

- 轴已执行回参考点：
DB31, ... DBX60.4 或 60.5 = 1（已回参考点/已同步 1 或 2）

一旦满足上述条件，不管当前处于哪种运行方式，系统都会将用对应补偿值来修正位置实际值，并由机床轴立即移动补偿值。

若接着参考位置由于超出编码器频率等原因丢失（DB31, ... DBX60.4 或 60.5 = 0），则会取消补偿。

4.4.2.2 调试

补偿控制点的数量 (MD38000)

必须为每根机床轴和每个测量系统（若存在第 2 个测量系统）定义预留的补偿表控制点的数量，并通过以下机床数据预留出所需的存储空间：

MD38000 \$MA_MM_ENC_COMP_MAX_POINTS[<e>,<AXi>]

其 <e> = 位置测量系统
中： <e> = 0 测量系统 1
 <e> = 1 测量系统 2

<AXi> = 轴名称（例如 X1、Y1、Z1）

所需的补偿控制点的数量如下计算：

$$\text{MM\_ENC\_COMP\_MAX\_POINTS [e,AXi]} = \frac{\$AA\_ENC\_COMP\_MAX[e,AXi] - \$AA\_ENC\_COMP\_MIN[e,AXi]}{\$AA\_ENC\_COMP\_STEP[e,AXi]} + 1$$

测量系统专用的补偿表参数

必须为每根机床轴和每个测量系统（若存在第 2 个测量系统）以系统变量的形式定义位置相关补偿以及附加的表参数：

- **\$AA_ENC_COMP[<e>,<N>,<AXi>]**
（补偿表中控制点 N 的补偿值）
 <N> = 控制点（轴位置）
 在表中为每个控制点输入对应的补偿值。
 <N> 受相应补偿表的控制点数量上限（MD38000
 \$MA_MM_ENC_COMP_MAX_POINTS）的限制：
 $0 \leq N \leq \text{MD38000} - 1$
 补偿值的大小不受限制。

说明

起点补偿值和终点补偿值在整个补偿范围内有效，即如果补偿表无法覆盖整个运行范围，这两个补偿值应设为“0”。

- **\$AA_ENC_COMP_STEP[<e>,<AXi>]**（控制点间距）
控制点间距定义了相应补偿表的补偿值之间的距离。

4.4 插补补偿

- **\$AA_ENC_COMP_MIN[<e>,<AXi>] (起点)**

起点指相应轴在补偿表中开始补偿的位置 (△ 控制点 0)。

起点的补偿值为 **\$AA_ENC_COMP[<e>,0,<AXi>]**。

小于起点的所有其它位置均使用控制点 0 的补偿值 (但不适用于带模数功能的补偿表)。

- **\$AA_ENC_COMP_MAX[<e>,<AXi>] (终点)**

终点指相应轴在补偿表中结束补偿的位置 (△ 控制点 <k>)。

终点的补偿值为 \$AA_ENC_COMP[<e>,<k>,<AXi>]。

所有大于终点的其它位置均使用控制点 <k> 的补偿值 (不适用于带模数功能的补偿表)。

下列边界条件适用于控制点 <k>:

- k = MD38000 - 1 时:
补偿表被完全占用!
- k < MD38000 - 1 时:
补偿表未被完全占用。表中输入的大于 k 的补偿值无效。
- k > MD38000 - 1 时:
控制系统内部会减小终点, 以限制补偿表。大于 k 的补偿值无效。

- **\$AA_ENC_COMP_IS_MODULO[<e>,<AXi>] (带模数功能的补偿)**

用于激活/取消带模数功能补偿的系统变量:

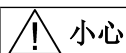
- \$AA_ENC_COMP_IS_MODULO[<e>,<AXi>] = 0: 无模数功能的补偿
- \$AA_ENC_COMP_IS_MODULO[<e>,<AXi>] = 1: 带模数功能的补偿

激活带模数功能的补偿后, 补偿表格会周期性重复, 也就是说: 位置 \$AN_CEC_MAX[t] (△ 控制点 \$AA_ENC_COMP[<e>,<k>,<AXi>]) 上的补偿值后直接紧跟位置

\$AA_ENC_COMP_MIN (△ 控制点 \$AA_ENC_COMP[<e>,<0>,<AXi>]) 上的补偿值。

对于模数为 360° 的回转轴, 建议使用 0 度作为起点 (\$AA_ENC_COMP_MIN), 使用 360° 作为终点 (\$AA_ENC_COMP_MAX)。

此时须在这两个补偿值中输入相同的数值, 否则补偿值会从最大值跳到最小值然后又从最小值跳到最大值。



小心

在输入补偿值时, 须确保定义的范围内的所有控制点均有对应的补偿值, 不能有遗漏。否则这些控制点会采用先前在这些位置上留下的数值。

说明

切换单位制 (修改 MD10240 \$MN_SCALING_SYSTEM_IS_METRIC) 时, 系统会自动对包含位置数据的表参数进行换算。

位置数据总是采用当前生效的单位制。换算必须于外部执行。

可如下配置位置数据的自动换算:

MD10260 \$MN_CONVERT_SCALING_SYSTEM = 1

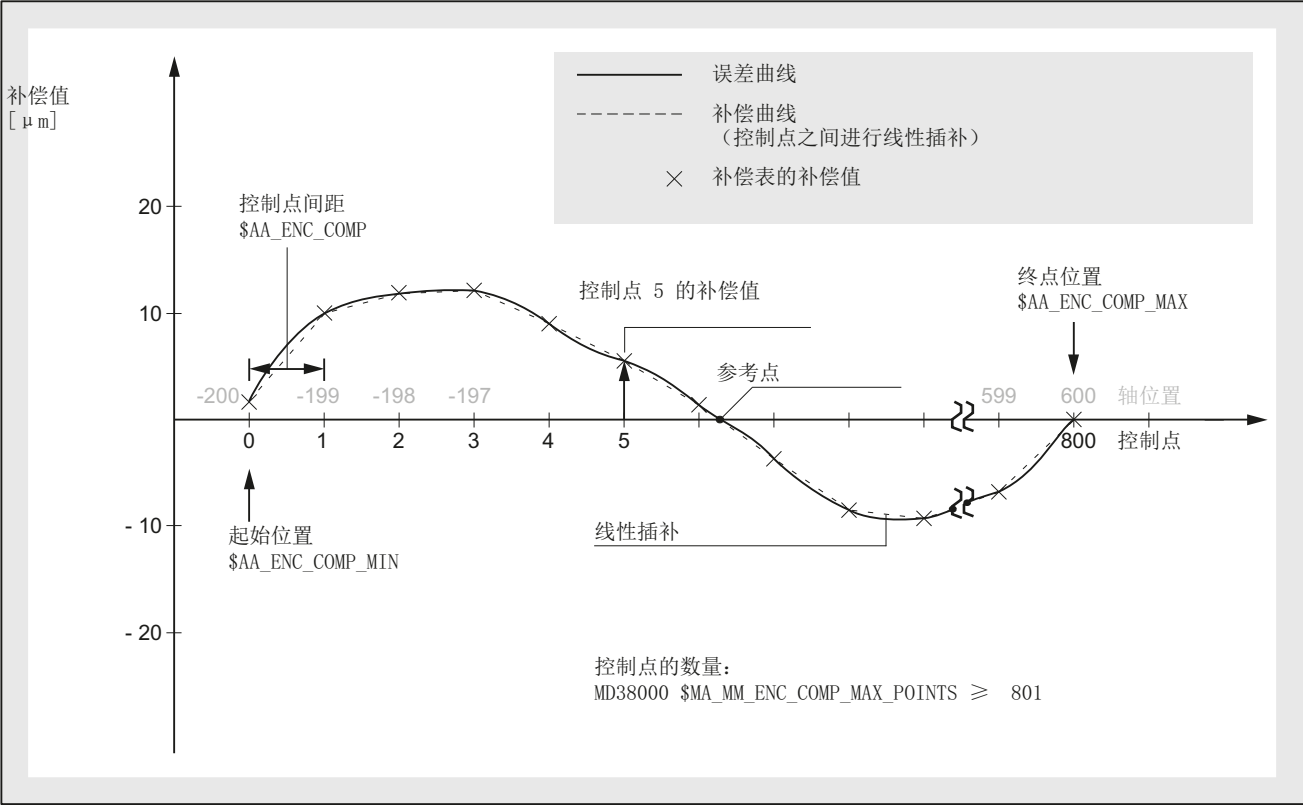
这样便可省去外部换算。

文档:

功能手册之基本功能分册; 章节“速度, 设定值/实际值系统, 闭环控制 (G2)”

4.4.2.3 示例

下面的示例介绍了如何设定机床轴 X1 的补偿值。



程序代码	注释
<pre>%_N_AX_EEC_INI CHANDATA(1) \$AA_ENC_COMP[0,0,X1]=0.003 \$AA_ENC_COMP[0,1,X1]=0.01 \$AA_ENC_COMP[0,2,X1]=0.012 ... \$AA_ENC_COMP[0,800,X1]=-0.0 \$AA_ENC_COMP_STEP[0,X1]=1.0 \$AA_ENC_COMP_MIN[0,X1]=-200.0 \$AA_ENC_COMP_MAX[0,X1]= 600.0 \$AA_ENC_COMP_IS_MODULO[0,X1] = 0 M17</pre>	; 第 1 补偿值 (控制点 0) : +3μm ; 第 2 补偿值 (控制点 1) : +10μm ; 第 3 补偿值 (控制点 2) : +12μm ; 最终补偿值 (控制点 800) : 0μm ; 控制点间距 1.0 mm ; 补偿起点 -200.0 mm ; 补偿终点 +600.0 mm ; 无模数功能的补偿

本例中控制点数量必须 ≥ 801:

MD38000 \$MA_MM_ENC_COMP_MAX_POINTS ≥ 801

补偿表会占用静态用户存储器 6.4 kB（每个补偿值 8 字节）。

4.4.3 对垂度和角度误差的补偿

4.4.3.1 功能说明

垂度误差

重力作用会引起机床部件（包含导轨）的弯曲，从而导致运动部件位移和弯曲。
同样，大型工件（例如滚轮）也会由于自重而下垂。

角度误差

若运动轴之间的夹角（例如相互垂直）偏离目标角度，那么越偏离零点，位置误差也将越大。

补偿

与 MSEC 不同，“垂度补偿”和“角度误差补偿”功能中的参照轴和补偿轴不能相同，因此需要在每个补偿表中指定轴。

若需对自重造成的轴垂度（参照轴）进行修正，必须调整另一根轴（补偿轴）的绝对位置。因此，“垂度补偿”是一种跨轴补偿。

如下图中所示，随着加工头朝 Y1 负向移动，主轴箱在 Z1 负向的垂度也就越大。

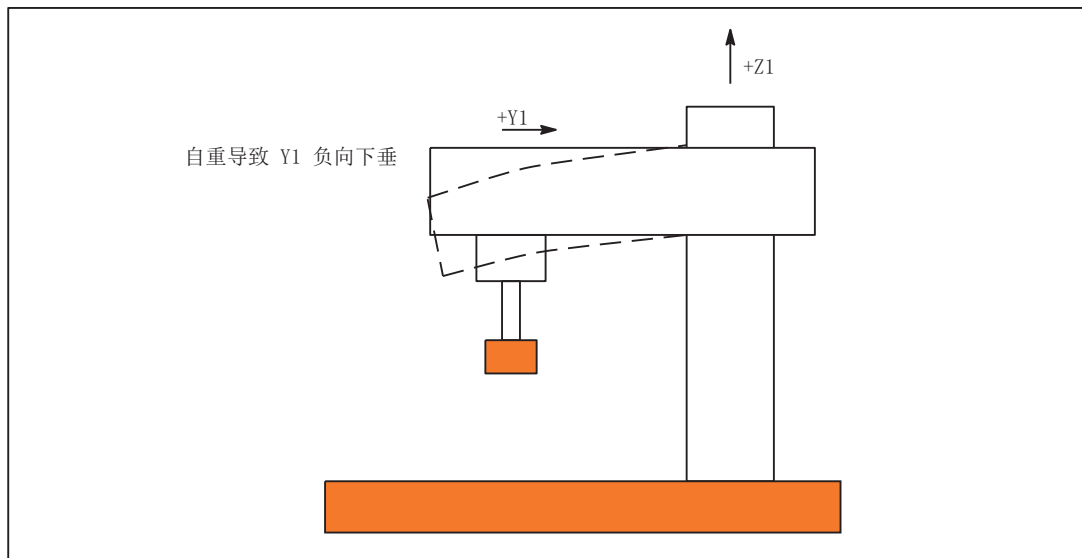


图 4-5 示例：自重引起的下垂

误差必须以补偿表的形式输入，该表包含了每个 **Y1** 轴实际位置上 **Z1** 轴的补偿值。表格中只输入控制点的补偿值。

Y1 轴运行时，控制系统在插补周期内对控制点间的位置执行线性插补，从而计算出对应的 **Z1** 轴补偿值。此补偿值将作为附加设定值发送至位置环。补偿值为正时，轴会在反向上移动该补偿值。

根据具体应用，也可为一根轴定义多个补偿关系。此时总补偿值为该轴各项补偿值的总和。

设置方法

下面介绍垂度补偿中计算补偿值的方式和影响补偿值的各种方法（见下图）。

1. 可以将一根轴定义为**多个**补偿表的输入值（参照轴）（可通过系统变量进行设置）。
2. 可以将一根轴定义为**多个**补偿表的输出值（补偿轴）（可通过系统变量进行设置）。总补偿值为各单个补偿值的总和。
对补偿表数量上限的规定如下：
 - 所有轴可用的表格总数上限：
2 * 系统轴数上限
 - 对一根补偿轴生效的表格数上限：
1 * 系统轴数上限
3. 一根轴可同时为参照轴和补偿轴。计算补偿值时始终会使用已编程的（理想的）位置设定值。
4. 可为每个补偿表定义补偿的生效范围（参照轴的起点和终点）以及控制点间距（可通过系统变量设置）。
5. 补偿可根据方向生效（可通过系统变量设置）。

6. 每个补偿表均具备模数功能，用于循环分析（可通过系统变量设置）。
7. 每个补偿表可以使用一个加权系数，用于和表格值相乘（可作为设定数据给定，这样便可通过零件程序、PLC 或由操作者随时更改）。
8. 使用“表格相乘”功能，每个补偿表可与任意补偿表（自身亦可）相乘。乘法运算通过系统变量进行。乘积会叠加到补偿轴的总补偿值上。
9. 可通过以下几种方法激活补偿：
 - 通过机床数据：
MD32710 \$MA_CEC_ENABLE[<AXi>]（垂度补偿使能）
使能机床轴 <AXi> 所有补偿关系的总和。
 - 通过设定数据：
SD41300 \$SN_CEC_TABLE_ENABLE[<t>]（补偿表的预设）
使能补偿表 [<t>] 的计算。
借助该数据可根据加工情况由零件程序或 PLC 用户程序修改补偿关系，比如：切换补偿表。

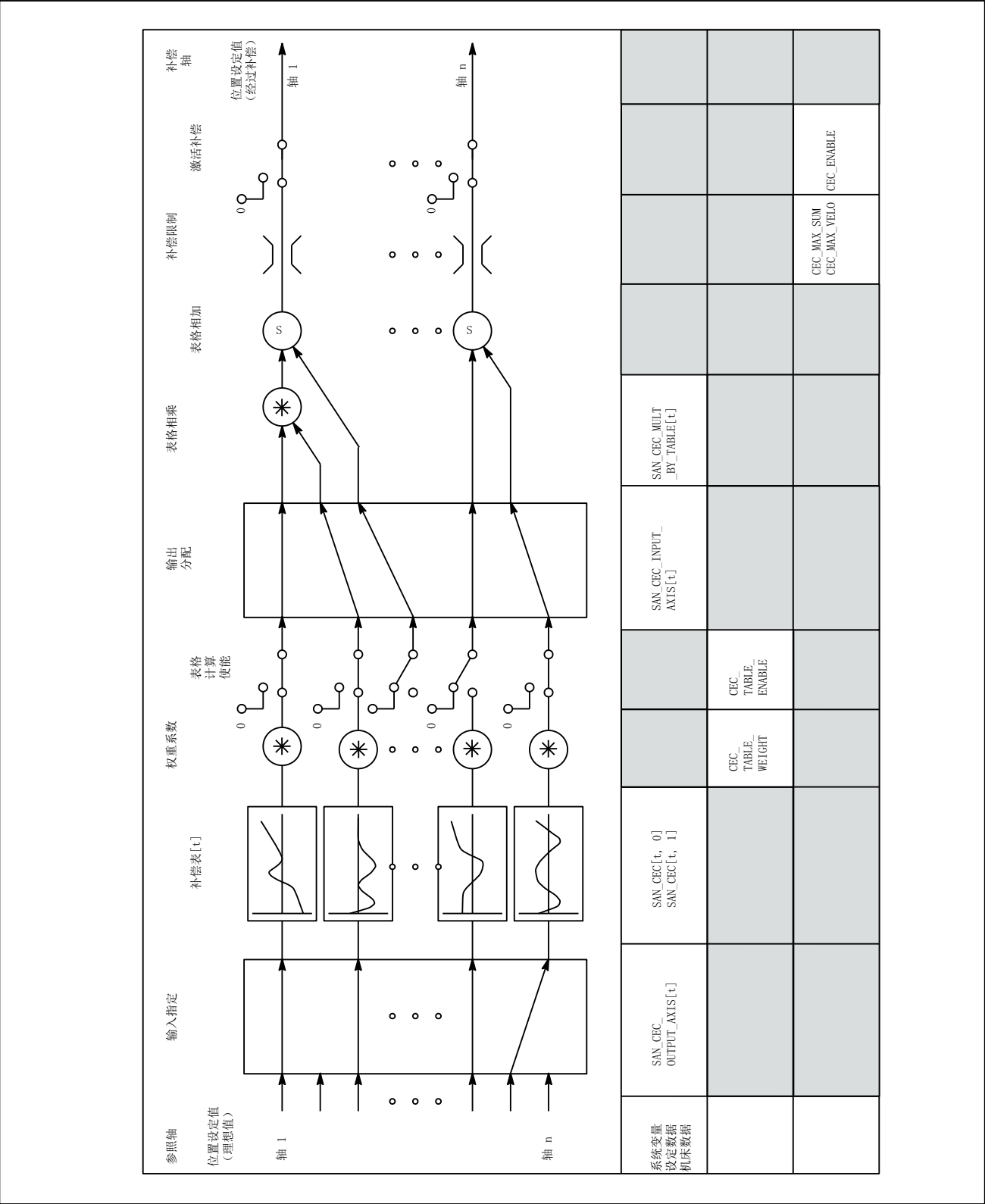


图 4-6 垂直补偿中补偿值的构成

综合补偿

由于一根轴的位置可作为多个表格的输入值（参照轴）、一根轴的总补偿值可以由多个补偿关系（表格）推导出且可以进行表格相乘，因此可以实现高效综合的垂度补偿和角度补偿。

另外，可有效地清除不同种类的误差。例如：一张带模数功能的表用于修正循环出现的误差，而不带模数的表用于修正偶尔出现的误差。

也可通过此功能补偿丝杠螺距误差：只需将参照轴和补偿轴设为相同轴即可。但是，与 MSEC 不同，该功能不会自动考虑测量系统的切换。

前提条件/生效方式

仅在满足下列前提条件时，“垂度补偿”才生效：

- 已为相关机床轴（补偿轴）激活了此功能：
MD32710 \$MA_CEC_ENABLE[<AXi>] = 1
- 补偿值存储在静态用户存储器中并在上电后生效。
- 已使能相应补偿表：
SD41300 \$SN_CEC_TABLE_ENABLE[<t>] = 1
- 参照轴和补偿轴的当前测量系统已回参考点：
DB31, ... DBX60.4 或 60.5 = 1（已回参考点/已同步 1 或 2）

一旦满足上述条件，不管当前处于哪种运行方式，系统都会将根据参照轴的设定位置来按补偿值调整补偿轴的设定位置，并立即使补偿轴移动补偿值。

若接着参考位置由于超出编码器频率等原因丢失（DB31, ... DBX60.4 或 60.5 = 0），则会取消补偿。

4.4.3.2 调试

补偿控制点的数量

必须为每个补偿关系定义所需的控制点的数量，并通过以下机床数据预留出必要的存储空间：

MD18342 \$MN_MM_CEC_MAX_POINTS[<索引>]（垂度补偿中控制点的最大数量），其中下标 = 0, 1, 2, ... (2 * 最大轴数) - 1

所需的补偿表控制点数量如下计算：

$$\text{MM\_CEC\_MAX\_POINTS [t]} = \frac{\text{\$AN\_CEC\_MAX [t]} - \text{\$AN\_CEC\_MIN [t]}}{\text{\$AN\_CEC\_STEP [t]}} + 1$$

表格参数

必须为每个补偿关系以系统变量的形式定义位置相关补偿和附加表参数：

- **\$AN_CEC[<下标>,<N>]** (补偿表 [<t>] 的控制点 <N> 的补偿值)
 <N> = 控制点 (参照轴的位置)，其中 $0 \leq N \leq (\text{MD18342 的值}) - 1$
 在表中为每个控制点输入对应的补偿轴的补偿值。
- **\$AN_CEC_INPUT_AXIS[<下标>]** (参照轴)
 机床轴的名称，其设定值作为补偿表 [<t>] 的输入值使用。
- **\$AN_CEC_OUTPUT_AXIS[<下标>]** (补偿轴)
 机床轴名称，补偿表 [<t>] 的输出值的作用于该轴。

说明

在多通道系统中，若机床轴的名称与通道轴相同，则必须设定“通用轴名称”AX1 ...。

- **\$AN_CEC_STEP[<下标>]** (控制点间距)
 控制点间距定义了补偿表 [<t>] 中输入值的距离：
- **\$AN_CEC_MIN[<下标>]** (起点)
 起点是补偿表 [<t>] 开始的参照轴位置 (≙ 控制点 0)。
 起点的补偿值为 **\$AN_CEC [<t>,0]**。
 所有小于起点的其它位置均使用控制点 0 的补偿值 (带模数功能的补偿表除外)。
- **\$AN_CEC_MAX[<下标>]** (终点)
 终点是补偿表结束的参照轴位置，= 控制点 <k>。
 终点的补偿值为 **\$AN_CEC [<下标>,<k>]**。
 所有大于终点的其它位置均使用控制点 <k> 的补偿值 (带模数功能的补偿表除外)。
 下列边界条件适用于控制点 k：
 - k = MD18342 - 1 时：
 补偿表被完全占用！
 - k < MD18342 - 1 时：
 补偿表未被完全占用。所输入的大于 k 的补偿值无效。
 - k > MD18342 - 1 时：
 控制系统内部会减小终点，以限制补偿表。大于 k 的补偿值无效。

- **\$AN_CEC_DIRECTION[<下标>]**（双向补偿）
可通过此系统变量设置补偿表 [<t>] 是对参照轴的两个运行方向生效，还是只在正向或负向上生效：
 - **\$AN_CEC_DIRECTION[<下标>] = 0:**
补偿表对参照轴的两个运行方向均有效
 - **\$AN_CEC_DIRECTION[<下标>] = 1:**
补偿表只对参照轴的正运行方向有效
 - **\$AN_CEC_DIRECTION[<下标>] = -1:**
补偿表只对参照轴的负运行方向有效应用：
使用两个表格：一个对参照轴的正运行方向生效、另一个对同一轴的负运行方向生效，从而实现和位置相关的间隙补偿。
- **\$AN_CEC_MULT_BY_TABLE [<下标>]**（表格相乘）
使用此功能，各表格的补偿值可以乘以任意表格（本身也可）的补偿值。乘积作为附加补偿值叠加到补偿表的总补偿值上。
句法：
\$AN_CEC_MULT_BY_TABLE[<下标>] = <编号>
其中：
<下标> = 补偿轴的表 1 的下标
<编号> = 补偿轴的表 2 的编号，其中
表编号 = 表下标 + 1
- **\$AN_CEC_IS_MODULO[<下标>]**（带模数功能的补偿）
用于激活/取消带模数功能补偿的系统变量：
 - **\$AA_CEC_COMP_IS_MODULO[<下标>] = 0:** 无模数功能的补偿
 - **\$AA_CEC_COMP_IS_MODULO[<下标>] = 1:** 带模数功能的补偿激活带模数功能的补偿后，补偿表会周期性重复，也就是说：位置 **\$AN_CEC_MAX[<下标>]**（控制点 **\$AN_CEC[<下标>,<k>]**）上的补偿值后直接紧跟位置 **\$AN_CEC_MIN[<下标>]**（控制点 **\$AN_CEC[<下标>,0]**）的补偿值。
必须为这两个补偿值输入相同的值，否则补偿值会从最大值跳到最小值然后又从最小值跳到最大值。
若需将模数回转轴作为参照轴来实现模数补偿，那么使用的补偿表也须采用模数计算。
示例：
MD30300 \$MA_IS_ROT_AX[AX1] = 1; 回转轴
MD30310 \$MA_ROT_IS_MODULO[AX1] = 1; 模数 360°
\$AN_CEC_INPUT_AXIS[0] = AX1
\$AN_CEC_MIN[0] = 0.0
\$AN_CEC_MAX[0] = 360.0
\$AN_CEC_IS_MODULO[0] = 1

单位制

切换单位制（修改 MD10240 \$MN_SCALING_SYSTEM_IS_METRIC）时，系统会自动对包含位置数据的表参数进行换算。

位置数据总是采用当前生效的单位制。换算必须于外部执行。

可如下配置位置数据的自动换算：

MD10260 \$MN_CONVERT_SCALING_SYSTEM = 1

采用此设置时，以下轴机床数据生效：

MD32711 \$MA_CEC_SCALING_SYSTEM_METRIC（垂度补偿的单位制）

可通过此机床数据为此轴上生效的所有表格确定单位制。所有位置数据和总补偿值一起都可以被换算为设置的单位制。单位制切换时不再需要外部换算位置数据。

监控

为了避免垂度补偿使机床轴的速度和加速度过大，系统会对总补偿值进行监控，并将其限制在最大值以内。

采用垂度补偿时的总补偿值上限通过以下机床数据逐轴定义：

MD32720 \$MA_CEC_MAX_SUM（垂度补偿的最大补偿值）

若所得到的总补偿值大于最大值，则会发出相应报警。程序处理不会因此而中断。作为附加设定值输出的补偿值被限制在最大值以内。

此外，也可以逐轴限制总补偿值的变化速率：

MD32730 \$MA_CEC_MAX_VELO（垂度补偿时的速度变化率）

输入值是系数，是相对于最大轴速度（MD32000 \$MA_MAX_AX_VELO）的百分比值。

超出限值时会发出相应报警。程序处理不会因此而中断。一旦补偿值回到限值范围内，系统便会追加运行由于限制而未运行的行程。

4.4.3.3 示例

Y1 轴垂度补偿表

下例中显示了 Y1 轴的垂度补偿表。Y1 轴是参照轴，Z1 轴上的补偿值随该其位置变化而变化。本例中使用补偿表 1，其下标为 0。

程序代码	注释
%_N_NC_CEC_INI	;

程序代码	注释
CHANDATA(1)	;
\$AN_CEC[0,0]=0	; 第 1 补偿值 (控制点 0)
	; 对于 Z1: $\pm 0\mu\text{m}$
\$AN_CEC[0,1]=0.01	; 第 2 补偿值 (控制点 1)
	; 对于 Z1: $+10\mu\text{m}$
\$AN_CEC[0,2]=0.012	; 第 3 补偿值 (控制点 2)
	; 对于 Z1: $+12\mu\text{m}$
...	;
\$AN_CEC[0.100]=0	; 终点补偿值 (控制点 101)
	; 对于 Z1: $\pm 0\mu\text{m}$
\$AN_CEC_INPUT_AXIS[0]=(AX2)	; 基准轴 Y1
\$AN_CEC_OUTPUT_AXIS[0]=(AX3)	; 补偿轴 Z1
\$AN_CEC_STEP[0]=8	; 插补点间距 8.0mm
\$AN_CEC_MIN[0]=-400.0	; 补偿起点 Y1=-400mm
\$AN_CEC_MAX[0] = 400.0	; 补偿终点 Y1=+400mm
\$AN_CEC_DIRECTION[0]=0	; 补偿表对 Y1 的两个运行方向均有效。
\$AN_CEC_MULT_BY_TABLE[0]=0	;
\$AN_CEC_IS_MODULO[0]=0	; 无模数功能的补偿
M17	;

本例中控制点数量必须 ≥ 101 ：

MD18342 \$MN_MM_CEC_MAX_POINTS[0] ≥ 101

补偿表会占用静态用户存储器 **8 Kb**（每个补偿值 **808 字节**）。

表格相乘的应用情形

下面举例说明如何用表格相乘来补偿立柱的弯曲。

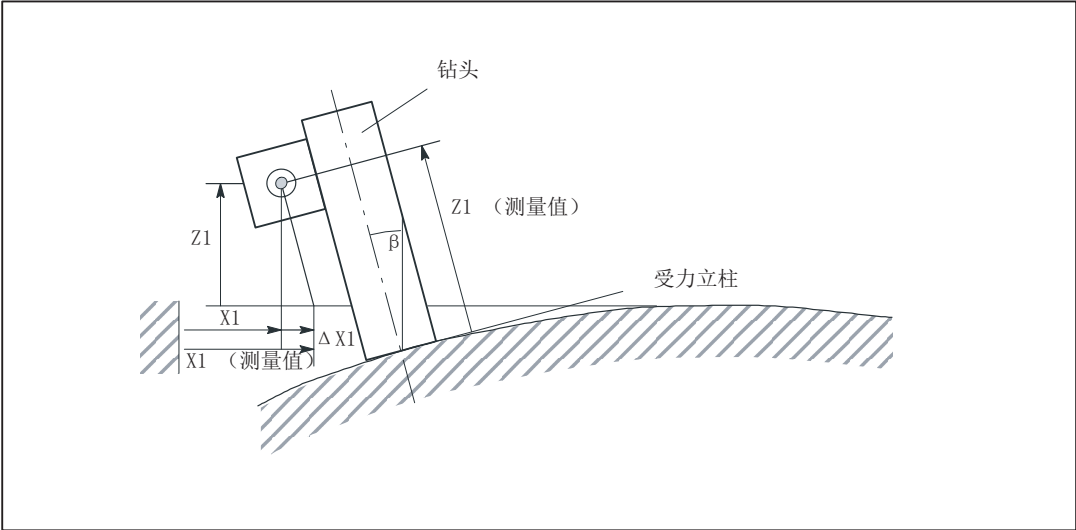


图 4-7 立柱弯曲补偿

在大型机床中，立柱的弯曲可能导致整个机床的倾斜。以上图显示的钻头为例，X1 轴的补偿不仅取决于 X1 轴自身的位置（因为该位置决定了倾角 β ），也取决于钻头所处的 Z1 的高度。

补偿时，X1 轴和 Z1 轴的补偿值如下运算：

$$\Delta X1 = Z1 * \sin\beta(X1) \approx Z1 * \beta(X1)$$

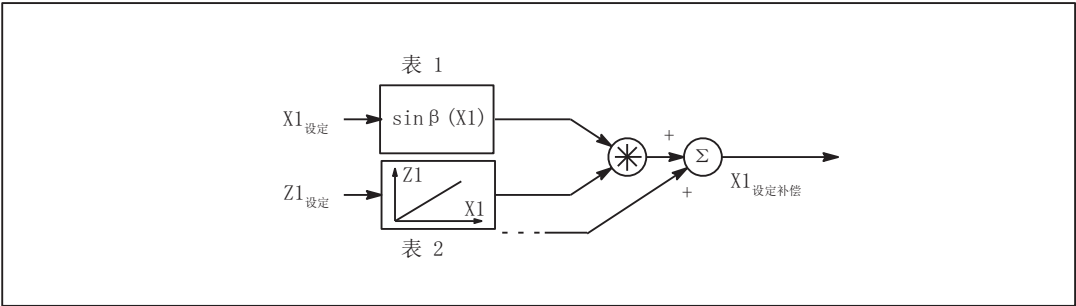


图 4-8 表格相乘

补偿表 1（下标 = 0）反映了轴 X1 对轴 X1 的影响（位置倾斜角 $\beta(X1)$ 的正弦值）。

补偿表 2（下标 = 1）反映了轴 Z1 对轴 X1 的影响（直线）。

在表 1 中选择将表 1（下标 = 0）与表 2 相乘：

$$\$AN\_CEC\_MULT\_BY\_TABLE[0] = 2$$

以网格结构输入补偿值

在实践中，一些平床身机床上 Z 轴的垂度补偿值是根据 X 轴和 Y 轴的不同位置测得的。在此前提条件下，系统提供一种网格格式补偿值输入方式。在网点（XY 平面）上是控制点及其对应的补偿值。控制点间的补偿值由控制系统线性插补得出。

下面以 4 行 x 5 列的网格为例说明如何进行垂度补偿和角度补偿。整个网格的尺寸为 2000 x 900 mm²。补偿值在 X 轴上以 500 mm 为步距，在 Y 轴上以 300 mm 为步距测得。

说明

网格的最大尺寸（行和列的数量）受以下因素限制：

- 行数取决于系统中轴的数量（取决于 NCU 类型）。
- 列数取决于一个补偿表中可输入值的数量上限（最多 2000 个值）。

行数和列数通过以下机床数据设置：

MD18342 \$ _MN_MM_CEC_MAX_POINTS（垂度补偿中控制点的最大数量）

 小心
用于配置存储器的机床数据 机床数据 MD18342 上的设置可在设备启动时使缓冲的 NC 用户存储器自动重置。此时，NC 用户存储器上的所有用户数据被删除，如驱动数据、HMI 机床数据、刀具补偿、零件程序等。 设置机床数据前应先备份用户数据。

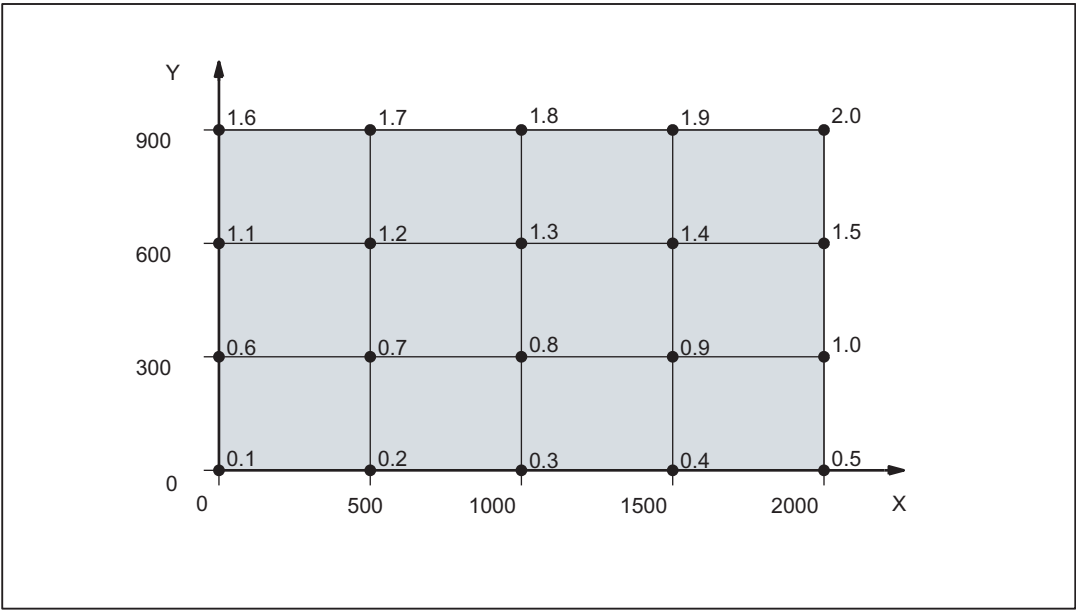


图 4-9 xy 平面内 Z 轴补偿值的网格式分布

本应用示例可通过以下零件程序代码实现：

```
$MA_CEC_ENABLE[Z1]= FALSE           ; 取消补偿
                                     ; 通过置位为 FALSE。
                                     ; 这样便可
                                     ; 修改表值，且不会
                                     ; 触发报警 17070。
NEWCONF                             ; 激活 $MA_CEC_ENABLE
; 在 f 表中定义值 f_i(x)
; 函数值 f_1(x)，用于下标为 [0] 的补偿表
$AN_CEC[0,0]=0.1
$AN_CEC[0,1]=0.2
$AN_CEC[0,2]=0.3
$AN_CEC[0,3]=0.4
$AN_CEC[0,4]=0.5

; 函数值 f_2(x)，用于下标为 [1] 的补偿表
$AN_CEC[1,0]=0.6
$AN_CEC[1,1]=0.7
$AN_CEC[1,2]=0.8
$AN_CEC[1,3]=0.9
$AN_CEC[1,4]=1.0

; 函数值 f_3(x)，用于下标为 [2] 的补偿表
$AN_CEC[2,0]=1.1
```



```
$AN_CEC[2,1]=1.2
$AN_CEC[2,2]=1.3
$AN_CEC[2,3]=1.4
$AN_CEC[2,4]=1.5

; 函数值 f_4(x), 用于下标为 [3] 的补偿表
$AN_CEC[3,0]=1.6
$AN_CEC[3,1]=1.7
$AN_CEC[3,2]=1.8
$AN_CEC[3,3]=1.9
$AN_CEC[3,4]=2.0

; 使能对包含补偿值的 f 表
$SN_CEC_TABLE_ENABLE[0]=TRUE
$SN_CEC_TABLE_ENABLE[1]=TRUE
$SN_CEC_TABLE_ENABLE[2]=TRUE
$SN_CEC_TABLE_ENABLE[3]=TRUE

; 定义 f 表的权重系数
$SN_CEC_TABLE_WEIGHT[0]=1.0
$SN_CEC_TABLE_WEIGHT[1]=1.0
$SN_CEC_TABLE_WEIGHT[2]=1.0
$SN_CEC_TABLE_WEIGHT[3]=1.0

; 下列表格参数的修改在
; 上电后才生效
; 定义 x1 为参照轴
$AN_CEC_INPUT_AXIS[0]=(X1)
$AN_CEC_INPUT_AXIS[1]=(X1)
$AN_CEC_INPUT_AXIS[2]=(X1)
$AN_CEC_INPUT_AXIS[3]=(X1)

; 定义 z1 为补偿轴
$AN_CEC_OUTPUT_AXIS[0]=(Z1)
$AN_CEC_OUTPUT_AXIS[1]=(Z1)
$AN_CEC_OUTPUT_AXIS[2]=(Z1)
$AN_CEC_OUTPUT_AXIS[3]=(Z1)

; 定义 f 表控制点间距
$AN_CEC_STEP[0]=500.0
$AN_CEC_STEP[1]=500.0
$AN_CEC_STEP[2]=500.0
$AN_CEC_STEP[3]=500.0

; 补偿起点 X1=0
```

4.4 插补补偿

```

$AN_CEC_MIN[0]=0.0
$AN_CEC_MIN[1]=0.0
$AN_CEC_MIN[2]=0.0
$AN_CEC_MIN[3]=0.0

; 补偿终点 x1=2000
$AN_CEC_MAX[0]=2000.0
$AN_CEC_MAX[1]=2000.0
$AN_CEC_MAX[2]=2000.0
$AN_CEC_MAX[3]=2000.0

; 下标为 [t1] 的 f 表格值与编号为
; [t2] 的 g 表格值相乘
; 符合上文提到的计算规定
$AN_CEC_MULT_BY_TABLE[0]=5
$AN_CEC_MULT_BY_TABLE[1]=6
$AN_CEC_MULT_BY_TABLE[2]=7
$AN_CEC_MULT_BY_TABLE[3]=8

; 确定 gi(y) 的 g 表值:
; 函数值 g1(x), 用于下标为 [4] 的补偿表
$AN_CEC[4,0]=1.0
$AN_CEC[4,1]=0.0
$AN_CEC[4,2]=0.0
$AN_CEC[4,3]=0.0

; 函数值 g2(x), 用于下标为 [5] 的补偿表
$AN_CEC[5,0]=0.0
$AN_CEC[5,1]=1.0
$AN_CEC[5,2]=0.0
$AN_CEC[5,3]=0.0

; 函数值 g3(x), 用于下标为 [6] 的补偿表
$AN_CEC[6,0]=0.0
$AN_CEC[6,1]=0.0
$AN_CEC[6,2]=1.0
$AN_CEC[6,3]=0.0

; 函数值 g4(x), 用于下标为 [7] 的补偿表
$AN_CEC[7,0]=0.0
$AN_CEC[7,1]=0.0
$AN_CEC[7,2]=0.0
$AN_CEC[7,3]=1.0

; 使能对包含补偿值的 g 表
$SN_CEC_TABLE_ENABLE[4]=TRUE

```

```
$SN_CEC_TABLE_ENABLE[5]=TRUE
$SN_CEC_TABLE_ENABLE[6]=TRUE
$SN_CEC_TABLE_ENABLE[7]=TRUE

; 确定 g 表的权重系数
$SN_CEC_TABLE_WEIGHT[4]=1.0
$SN_CEC_TABLE_WEIGHT[5]=1.0
$SN_CEC_TABLE_WEIGHT[6]=1.0
$SN_CEC_TABLE_WEIGHT[7]=1.0

; 下列表格参数的修改在
; 上电后才生效
; 定义 Y1 为参照轴
$AN_CEC_INPUT_AXIS[4]=(Y1)
$AN_CEC_INPUT_AXIS[5]=(Y1)
$AN_CEC_INPUT_AXIS[6]=(Y1)
$AN_CEC_INPUT_AXIS[7]=(Y1)

; 定义 Z1 为补偿轴
$AN_CEC_OUTPUT_AXIS[4]=(Z1)
$AN_CEC_OUTPUT_AXIS[5]=(Z1)
$AN_CEC_OUTPUT_AXIS[6]=(Z1)
$AN_CEC_OUTPUT_AXIS[7]=(Z1)

; 定义 g 表控制点间距
$AN_CEC_STEP[4]=300.0
$AN_CEC_STEP[5]=300.0
$AN_CEC_STEP[6]=300.0
$AN_CEC_STEP[7]=300.0

; 补偿起点 Y1=0
$AN_CEC_MIN[4]=0.0
$AN_CEC_MIN[5]=0.0
$AN_CEC_MIN[6]=0.0
$AN_CEC_MIN[7]=0.0

; 补偿终点 Y1=900
$AN_CEC_MAX[4]=900.0
$AN_CEC_MAX[5]=900.0
$AN_CEC_MAX[6]=900.0
$AN_CEC_MAX[7]=900.0
$MA_CEC_ENABLE[Z1]=TRUE          ; 重新激活补偿
NEWCONF

; 执行程序测试,
; 以检验补偿的有效性
```

4.4 插补补偿

```
G01 F1000 X0 X0 Z0 G90
R1=0 R2=0
LOOP_Y:
LOOP_X:
STOPRE
X=R1 Y=R2
M0 ; 等待，以检查 CEC 值
R1=R1+500
IF R1 <=2000 GOTOB LOOP_X
R1=0
R2=R2+300
IF R2<=900 GOTOB LOOP_Y
```

说明

您可在操作界面上的“补偿：垂度 + 温度”下查看补偿值。为此先选择软键“诊断”，然后选择“轴信息”。您可在变量“补偿：垂度 + 温度”旁读取当前生效的补偿值。

```
; 为了提供表配置，上电时
; 会设置机床数据
; cec.md:
; 设置选项数据用于调试
; 定义补偿表中控制点的数量
; 此机床数据会重新配置存储器
$MN_MM_CEC_MAX_POINTS[0]=5
$MN_MM_CEC_MAX_POINTS[1]=5
$MN_MM_CEC_MAX_POINTS[2]=5
$MN_MM_CEC_MAX_POINTS[3]=5
$MN_MM_CEC_MAX_POINTS[4]=4
$MN_MM_CEC_MAX_POINTS[5]=4
$MN_MM_CEC_MAX_POINTS[6]=4
$MN_MM_CEC_MAX_POINTS[7]=4
$MA_CEC_MAX_SUM[AX3]=10.0 ; 定义最大
; 总补偿值
$MA_CEC_MAX_VELO[AX3]=100.0 ; 限制总补偿值的
; 总补偿值
M17
```

说明

补偿值无法直接作为 2 维网格输入。必须创建补偿表，并在其中输入补偿值。

一个补偿表包含一行的补偿值（例如四行，即四个补偿表）。在示例的第一个表中输入补偿值 0.1 至 0.5，第二个表中输入补偿值 0.6 至 1.0，以此类推。这些补偿表也被称为 f 表，其表值被称为 f_i(x)（i=表的编号）。

为了分析 f 表的补偿值，需要将其与其他补偿表相乘。这些表在下文称为 g 表，其表值称为 $g_i(y)$ 。 f 表和 g 表的数量相同（示例中为四）。

在 g 表中，每个表中均有一个补偿值设置为 1，其他值均设置为 0。表中补偿值 1 的位置以表编号为准。在第一个 g 表中，补偿值 1 位于第一个控制点位置上，在第二个 g 表中补偿值 1 位于第二个控制点位置上，以此类推。通过将 g 表和 f 表相乘，正确的 f 表补偿值将通过乘以 1 被选中。不相关的补偿值则会通过乘以 0 被隐藏。

此时位置 (x/y) 上的补偿值 D_z 按照以下公式计算：

$$D_z(x/y) = f_1(x) * g_1(y) + f_2(x) * g_2(y) + \dots$$

在计算机床主轴当前位置的补偿值时，按照此公式则将 f 表的值和 g 表的值相乘。

在本示例中，为计算出补偿值 $D_z(500/300)$ ，则需要将 f 表的函数值 $f_i(500)$ 与 g 表的函数值 $g_i(300)$ 相乘。

$$D_z(500/300) = f_1(1000) * g_1(300) + f_2(1000) * g_2(300) + f_3(1000) * g_3(300) + f_4(1000) * g_4(300)$$

$$D_z(500/300) = 0.2 * 0 + 0.7 * 1 + 1.2 * 0 + 1.7 * 0 = 0.7$$

4.4.4 双向丝杠螺距误差补偿

4.4.4.1 功能说明

如果在两个方向上补偿值相差很大、间隙不恒定或对精度要求很高，可能需要采用双向丝杠螺距误差补偿或测量系统误差补偿（采用直接位置测量系统时）。

双向丝杠螺距误差补偿

采用“双向丝杠螺距误差补偿”（“双向 LEC”）时，每根轴会使用两个补偿表。一个补偿表用于正运行方向，另一个用于负运行方向。补偿表中记录了相应补偿点上的偏差（理想设定值和所测得的实际值之间的差值）。中间值的补偿值则由控制系统通过线性插补自动计算。

前提条件/生效方式

“双向 LEC”在 SINUMERIK 控制系统中作为“垂度补偿”的特例实现。因此，“垂度补偿”的前提条件和生效条件也适用于此功能（参见“对垂度和角度误差的补偿 (页 261)”）。

可通过比照测量（例如使用激光干涉仪）检验补偿效果，或者在最简单的情形下查看各轴的信息显示对话框。

注意
如果双向 LEC 同时用于垂度补偿和角度补偿，则必须注意这些功能的边界条件，例如表 <t> 与各功能的对应关系。

4.4.4.2 调试

误差值或补偿值的测量

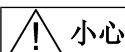
同调试“单向 **LEC**”时一样，调试“双向 **LEC**”时也需要采用适宜的测量设备（例如激光干涉仪）测定每根轴的误差曲线（参见章节“对丝杠螺距误差和测量系统误差的补偿 (页 255)”）。为了执行测量，必须创建一个包含测量点和等待时间的零件程序（参见章节“示例 (页 281)”：程序“**BI_SSFK_MESS_AX1_X.MPF**”）。

在实践中，不同的测量仪器在和 **SINUMERIK** 配用时提供的功能也不同，因此本手册仅就本系统上常规的测量步骤进行说明。

说明
只有在机床数据中正确设置了 MCS 中轴的运行方向后，才能在首次调试时执行测量，以测定丝杠螺距误差。

执行调试

1. 定义控制点的数量（另见章节“垂度和角度误差的补偿：调试”（页 265））
必须为每根轴的正运行方向和负运行方向分别指定一个补偿表。必须通过以下机床数据定义控制点的数量：
MD18342 \$MN_MM_CEC_MAX_POINTS[<t>]（垂度补偿中控制点的最大数量）
其中：<t> = 补偿表的下标
允许范围： $0 \leq t < 2 * \text{最大轴数}$



小心

用户数据丢失

在 MD18342 更改时会发出报警 ALARM 4400:

“缓存重组！”

为确保自动存储器配置不会使所有输入的数据丢失，在未进行批量调试前**请勿**重启系统。

示例:

MD18342 [0] = 11; 11 个控制点用于补偿表 1，例如 X 轴正运行方向

MD18342 [1] = 11; 11 个控制点用于补偿表 2，例如 X 轴负运行方向

MD18342 [2] = 21; 21 个控制点用于补偿表 3，例如 Y 轴正运行方向

MD18342 [3] = 21; 21 个控制点用于补偿表 4，例如 Y 轴正运行方向

...

MD18342 [61] = ...; 补偿表 62 的控制点数量

2. 执行批量调试:
 - 创建包含 MD18342 [<t>] 输入的 NC 存档。
 - 回读已创建的 NC 存档。
提示: 此时系统会配置 NC 存储器。
补偿表现在可用。
3. 以零件程序的形式为各轴及各运行方向创建含补偿值的表格（参见章节“示例 (页 281)”：程序“BI_SSFk_MESS_AX1_X.MPF”）。
4. 在控制系统中执行带补偿值的零件程序。
运行模式 AUTO > 选择程序 > CYCLE 启动

说明

每次读取补偿表前，以下数据必须总是先置 0，然后再置 1 用于激活补偿:

MD32710 \$MA_CEC_ENABLE[<AXi>]（垂度补偿使能）= 0 → 1

SD41300 \$SN_CEC_TABLE_ENABLE[<t>]（补偿表使能）= 0 → 1

反向间隙必须始终置 0:

MD32450 \$MA_BACKLASH [<e>]（反向间隙）= 0

其中：<e> = 位置测量系统

使用程序模版“BI_SSFk_TAB_AX1_X.MPF”（参见章节“示例 (页 281)”）可自动实现这些步骤。在手动输入机床数据时须注意“激活 MD”或“复位”。

5. 执行上电（热启动）。

4.4 插补补偿

- 6. 现在可使用激光干涉仪进行比照测量。
- 7. 为了进一步改善补偿结果，可在程序中对各补偿值进行修正。重新读取表格后无需再次上电。

说明

在 SINUMERIK 828D 上的操作步骤

在 SINUMERIK 828D 上可省去第 2 步和第 3 步，因为一旦使能“垂度补偿，多维”选项，系统立即提供 8 张补偿表，每张表包含 200 个控制点。但无法再手动补充表格！

说明

如第 5 步中所述，补偿表作为可执行程序载入至程序存储器中，在按下 CYCLE START 后传输至先前配置好的控制系统存储区域。该过程可能需要为每张表格重复。但也可以在一个初始化步骤中一次性载入所有表格。补偿值将在 MD32710[<AXi>] = 1 和强制上电后生效。

说明

NC_CEC.INI

通过“调试”>“系统数据”（文件夹“NC 存档数据”>“垂度角度补偿”）复制的文件“NC_CEC.INI”包含了系统建立的所有垂度补偿/角度补偿和和双向 LEC 表。

表格参数

以系统变量的形式为每个方向定义位置相关补偿和附加表格参数：

- \$AN_CEC[<t>,<N>]（补偿表 [<t>] 的控制点 <N> 的补偿值）
- \$AN_CEC_INPUT_AXIS[<t>]（参照轴）
- \$AN_CEC_OUTPUT_AXIS[<t>]（补偿轴）

说明

在“双向 LEC”中，参照轴和补偿轴相同。

- \$AN_CEC_STEP[<t>]（控制点间距）
- \$AN_CEC_MIN[<t>]（起点）
- \$AN_CEC_MAX[<t>]（终点）

- **\$AN_CEC_DIRECTION[<t>]**（双向补偿）
可通过此系统变量设置补偿表 [<t>] 是否只在正方向或负方向上生效：
 - **\$AN_CEC_DIRECTION[<t>] = 1:**
补偿表只对参照轴的正运行方向有效
 - **\$AN_CEC_DIRECTION[<t>] = -1:**
补偿表只对参照轴的负运行方向有效

说明

设置 **\$AN_CEC_DIRECTION[<t>] = 0**（补偿表对参照轴的两个运行方向均有效）在“双向 LEC”中无意义。

- **\$AN_CEC_IS_MODULO[<t>]**（带模数功能的补偿）

说明

此系统变量的详细说明（参见章节“垂度和角度误差的补偿：调试 (页 265)”）。

单位制

参见章节“垂度和角度误差的补偿：调试 (页 265)”。

监控

参见章节“垂度和角度误差的补偿：调试 (页 265)”。

4.4.4.3 示例

下面以一台 3 轴机床为例详细介绍 X 轴和位置相关的补偿表。

配置

补偿控制点的数量：

MD18342 \$MN_MM_CEC_MAX_POINTS[0] = 11（表 1：轴 X，正运行方向）

MD18342 \$MN_MM_CEC_MAX_POINTS[1] = 11（表 2：轴 X，负运行方向）

说明

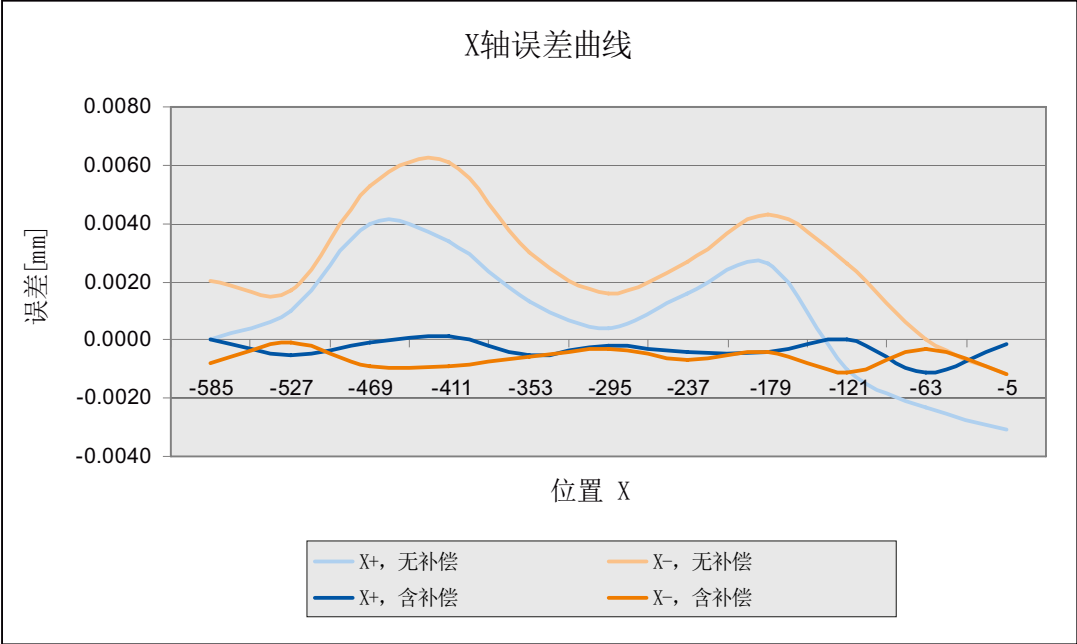
使用 SINUMERIK 828D 时无需定义控制点数量，因为一旦使能“垂度补偿，多维”选项，系统立即提供 8 张补偿表，每张表包含 200 个控制点。但无法再手动补充表格！

控制点

表 <t>	[0,<N>]										
控制点数量	MD18342 \$MN_MM_CEC_MAX_POINTS[0] = 11										
控制点 <N>	0	1	2	3	4	5	6	7	8	9	10
位置 X	-585	-527	-469	-411	-353	-295	-237	-179	-121	-63	-5

测量

	位置	补偿编号	设定位置	偏差		经过补偿后的对比测量	
			测量位置 [mm]	方向 + [mm]	方向 - [mm]	方向 + [mm]	方向 - [mm]
\$AC_CEC_MIN[<t>]	-585	0	-585	0,0000	0,0020	0,0000	-0,0008
		1	-527	0,0010	0,0017	-0,0005	-0,0001
		2	-469	0,0040	0,0053	-0,0001	-0,0009
		3	-411	0,0034	0,0061	0,0001	-0,0009
		4	-353	0,0013	0,0030	-0,0005	-0,0006
		5	-295	0,0004	0,0016	-0,0002	-0,0003
		6	-237	0,0016	0,0027	-0,0004	-0,0007
		7	-179	0,0026	0,0043	-0,0004	-0,0004
		8	-121	-0,0010	0,0026	0,0000	-0,0011
		9	-63	-0,0023	0,0000	-0,0011	-0,0003
\$AC_CEC_MAX[<t>]	-5	10	-5	-0,0031	-0,0012	-0,0001	-0,0012



编程

下面的“BI_SSFk_TAB_AX1_X.MPF” 程序中包含了 X 轴的两个补偿表（正运行方向和负运行方向）的参数赋值：

```

; 双向 LEC
; 第 1 轴 MX1
; 表 1, 正运行方向
; 表 2, 负运行方向
; -----
CHANDATA(1)
$MA_CEC_ENABLE[AX1]=0           ; 补偿关闭
$SN_CEC_TABLE_ENABLE[0]=0       ; 禁用表 1
$SN_CEC_TABLE_ENABLE[1]=0       ; 禁用表 2
NEWCONF
; -----
$AN_CEC[0,0]=0                   ; 第 1 补偿值 (控制点 0)
$AN_CEC[0,1]=0.001               ; 第 2 补偿值 (控制点 1)
$AN_CEC[0,2]=0.004               ; 第 3 补偿值 (控制点 2)
$AN_CEC[0,3]=0.0034              ; 第 4 补偿值 (控制点 3)
$AN_CEC[0,4]=0.0013              ; 第 5 补偿值 (控制点 4)
$AN_CEC[0,5]=0.0004              ; 第 6 补偿值 (控制点 5)
$AN_CEC[0,6]=0.0016              ; 第 7 补偿值 (控制点 6)
```

K3: 补偿

4.4 插补补偿

```
$AN_CEC[0,7]=0.0026          ; 第 8 补偿值 (控制点 7)
$AN_CEC[0,8]=-0.001          ; 第 9 补偿值 (控制点 8)
$AN_CEC[0,9]=-0.0023         ; 第 10 补偿值 (控制点 9)
$AN_CEC[0,10]=-0.0031        ; 终点补偿值 (控制点 10)
$AN_CEC_INPUT_AXIS[0]=(AX1)   ; 参照轴
$AN_CEC_OUTPUT_AXIS[0]=(AX1)  ; 补偿轴
$AN_CEC_STEP[0]=58.0          ; 控制点间距
$AN_CEC_MIN[0]=-585.0         ; 补偿开始
$AN_CEC_MAX[0]=-5.0           ; 补偿结束
$AN_CEC_DIRECTION[0]=1        ; 补偿表对正运行方向生效
$AN_CEC_MULT_BY_TABLE[0]=0    ; 不相乘 (此处无意义)
$AN_CEC_IS_MODULO[0]=0        ; 无模数功能的补偿
;-----
$AN_CEC[1,0]=0.002            ; (控制点 0)
$AN_CEC[1,1]=0.0017           ; (控制点 1)
$AN_CEC[1,2]=0.0053           ; (控制点 2)
$AN_CEC[1,3]=0.0061           ; (控制点 3)
$AN_CEC[1,4]=0.003            ; (控制点 4)
$AN_CEC[1,5]=0.0016           ; (控制点 5)
$AN_CEC[1,6]=0.0027           ; (控制点 6)
$AN_CEC[1,7]=0.0043           ; (控制点 7)
$AN_CEC[1,8]=0.0026           ; (控制点 8)
$AN_CEC[1,9]=0.000            ; (控制点 9)
$AN_CEC[1,10]=-0.0012         ; (控制点 10)
$AN_CEC_INPUT_AXIS[1]=(AX1)   ; 参照轴
$AN_CEC_OUTPUT_AXIS[1]=(AX1)  ; 补偿轴
$AN_CEC_STEP[1]=58.           ; 控制点间距
$AN_CEC_MIN[1]=-585.0         ; 补偿开始
$AN_CEC_MAX[1]=-5.0           ; 补偿结束
$AN_CEC_DIRECTION[1]=-1       ; 补偿表对负运行方向生效
$AN_CEC_MULT_BY_TABLE[1]=0    ; 不相乘 (此处无意义)
$AN_CEC_IS_MODULO[1]=0        ; 无模数功能的补偿 (仅针对回转轴)
;-----
$MA_CEC_ENABLE[AX1]=1         ; 补偿激活
$SN_CEC_TABLE_ENABLE[0]=1     ; 使能表 1
$SN_CEC_TABLE_ENABLE[1]=1     ; 使能表 2
NEWCONF
M17
```

还可创建其他补偿表，例如用于 Y 轴和 Z 轴：

MD18342 \$MN_MM_CEC_MAX_POINTS[2] = 90 (表 3: Y 轴，正运行方向)

MD18342 \$MN_MM_CEC_MAX_POINTS[3] = 90 (表 4: Y 轴，负运行方向)

MD18342 \$MN_MM_CEC_MAX_POINTS[4] = 50 (表 5: Z 轴, 正运行方向)

MD18342 \$MN_MM_CEC_MAX_POINTS[5] = 50 (表 6: Z 轴, 负运行方向)

4.4.5 垂度补偿和 NCU-Link 的综合应用 - 仅适用于 840D sl

应用

设备带 NCU-Link 时, NCU-Link 组中的任意轴可相互进行补偿。通过垂度补偿关联在一起的两根轴必须在一个 NCU 上插补。

另见

- NCU-Link (页 88)
- 轴容器 (页 109)

功能

垂度补偿功能的设置通过“\$AN_CEC ...”形式的系统变量进行。

这些系统变量通常通过 NCK 在特定通道中执行的零件程序设置。在变量 \$AN_CEC_OUTPUT_AXIS 或 \$AN_CEC_INPUT_AXIS 中可使用通道轴名称。因此可访问该通道的每根轴, 即使该轴在另一个 NCU 上。

在通道 2 的程序中写入以下命令, 可以使轴 ZZ 和 XX 关联在一起。该命令针对的是编号为 0 的补偿表:

```
$AN_CEC_INPUT_AXIS[0] = (XX)
```

```
$AN_CEC_OUTPUT_AXIS[0] = (ZZ)
```

现在 NCU-1 上的 AX3 和 NCU-2 上的 AX2 便“关联”在一起 (参见配置 1)。

若待关联的轴位于两个不同的通道上, 则可通过以下方案进行参数设置:

- **方案 1：“采用通道轴名称编程”：**
此时会创建 2 个不同的零件程序 TP1 和 TP2，这两个程序在不同的通道中执行。
轴“ZZ” 和“XR” 关联：
通道 1 中零件程序 TP1 的程序段：
\$AN_CEC_INPUT_AXIS[0] = (XR)
通道 2 中零件程序 TP2 的程序段：
\$AN_CEC_OUTPUT_AXIS[0] = (ZZ)
通道 1 中的 TP1 和通道 2 中的 TP2 执行完毕后，重启后 NCU-2 上的轴 AX2 便和 NCU-1 上的轴 AX1 关联在一起。
- **方案 2：“采用机床轴名称编程”：**
创建一个在 NCU-1 的任意通道中运行的零件程序，在程序中指定机床轴名称和 NCU 编号。
轴“ZZ” 和“XR” 关联：
\$AN_CEC_INPUT_NCU[0]=1
\$AN_CEC_INPUT_AXIS[0] = (AX1)
\$AN_CEC_OUTPUT_NCU[0]=2
\$AN_CEC_OUTPUT_AXIS[0] = (AX2)
NCK 会监控本地 NCU 上的轴是否真的进行插补，即是否存在可编程这些轴的通道。
“本地 NCU” 总是为运行零件程序的 NCU。
在 NCU-1 上，可将以下轴作为配置 1 中的输入轴或输出轴：NC1_AX1、NC1_AX3、NC1_AX4、NC1_AX5、NC2_AX2、NC2_AX6
从 NCK 进行数据备份时，补偿数据总是以“机床轴名称”的方式显示。

说明

在轴名称前必须编写 NCU 编号。无法进行 NC1_AX1 和 NC1_AX2 之间的垂度补偿。

轴指定

输入轴和输出轴通过以下系统变量指定：

\$AN_CEC_INPUT_NCU	与	\$AN_CEC_INPUT_AXIS
\$AN_CEC_OUTPUT_NCU	与	\$AN_CEC_OUTPUT_AXIS

这些系统变量在重启后才生效。

数据备份总是采用机床轴名称的形式。

说明

垂度补偿只能将 NCU 上的一根轴和另一根同样可由该 NCU 运行的轴关联在一起。

轴（输入或输出）不位于本地 NCU 上时，这些变量为选项。若在编程 \$AN_CEC_INPUT_AXIS 和 \$AN_CEC_OUTPUT_AXIS 时使用了通道轴名称，则系统变量 \$AN_CEC_INPUT_NCU 和 \$AN_CEC_OUTPUT_NCU 无意义。

控制系统会检查能否通过该 NCU 插补这两根轴，即能否通过该 NCU 上的一个程序运行这些轴。这些轴可被指定给不同的通道。属于不同 NCU 的两根轴同样可被补偿。否则控制系统会拒绝请求，并输出报警 17040。

补偿中的两根轴必须在同一 NCU 上进行插补，即可存在一个或两个零件程序来运行 NCU 上的输入轴和输出轴。

轴容器

轴容器是一个同类轴构成的轴组。可将该组中的一根轴指定给一根通道轴。此指定关系可变，即运行过程中容器轴和通道轴的对应关系总是会不断变化。因此可编程包含一根轴的零件程序，并在运行过程中通过该程序运行不同的轴。

示例：

主轴鼓上设置了 4 根主轴。每根主轴装载一个工件，主轴鼓每个周期旋转 90 度。这样工件便可从一个加工站传送到另一个加工站。尽管不断换入新的主轴，加工站的通道只允许编程一根主轴。这就是轴容器旋转。

若轴容器为缺省设置（即 \$AN_AXCTAS == 0），可将轴容器与垂度补偿组合使用，否则编程会被拒绝，并触发报警 17040。

需要将“YY”和“XX”关联在一起（参见配置 2）：

1. 用通道轴名称编写指令
\$AN_CEC_INPUT_AXIS[0] = (XX)
\$AN_CEC_OUTPUT_AXIS[0] = (YY)
2. 用机床轴名称编写指令
\$AN_CEC_INPUT_NCU[0]=1 ; 可选
\$AN_CEC_INPUT_AXIS[0] = (AX2)
\$AN_CEC_OUTPUT_NCU[0]=2
\$AN_CEC_OUTPUT_AXIS[0] = (AX2)

现在 NCU-1 的轴 AX2 便和 NCU-2 的轴 2 关联在一起。

注意

跨通道取轴

每次容器旋转时，YY 都保持与 XX 的关联，但是其实际为 YY 之后的另一根轴：在配置 3 中 YY 对应“NCU-1 的 AX5”。

因此每次旋转后它都会关联其他真实轴：在此示例中 NCU-1 的 AX-5 与 NCU-1 的 AX-2 关联。

通常来说：

在 LAI 层的两根轴之间建立关联，这样一来每次轴容器旋转后所关联的轴都会变化。为了实现两根真实轴间的精确关联，必须在每次轴容器旋转时激活一个新的表。

配置示例

下图显示了由两个 NCU 组成的 NCU-Link 的不同轴配置（配置 1、配置 2 和配置 3）。

在配置 1 中显示了 NCU-1 的两个通道：CHAN-1 和 CHAN-2。通道轴名称已经通过机床数据 \$MC_AXCONF_CHANAX_NAME_TAB 定义。第二个 NCU 的通道配置不予以显示。

所有可通过此 NCU 插补的轴均汇总在“逻辑 NCK 机床轴映射”（LAI 层）”中。通道和 MCS 轴层的对应关系通过 \$MC_AXCONF_MACHAX_USED 指定。

“逻辑 NCK 机床轴映射”和真实轴之间的对应关系通过机床数据 \$MN_AXCONF_LOGICMACHAX_TAB 指定。观察从通道轴 ZZ 处开始的连线，会发现该连线在 NCU-2 上的轴 AX-2 处结束，即为了运行 NCU-2 的第 2 轴，必须在 NCU-1 的第 2 通道中编程以下指令： "N2040 POS[ZZ]=10 FA[ZZ]=1000"

和配置 1 相比，配置 2 和配置 3 增加了一个轴容器（CT1），其通过机床数据 \$MN_AXCT_AXCONF_ASSIGN_TAB1 设置。轴容器是一个跨 NCU 对象，即每个轴容器对于整个 NCU 群仅存在一次。

在 NCU-1 上，通道轴 YR 和 YY 是轴容器内的参与轴，NCU-2 中的两根通道轴不予以显示。轴容器包含了真实轴 NC1_AX5、NC1_AX6、NC2_AX1 和 NC2_AX2。启动时轴容器将 YR 与 NC2_AX1 相关联，将 YY 与 NC2_AX2 相关联。配置 3 中的轴容器已经过旋转，即关联结构已变更。此时 YR 与 NC2_AX2 相关联，YY 与 NC1_AX5 相关联。

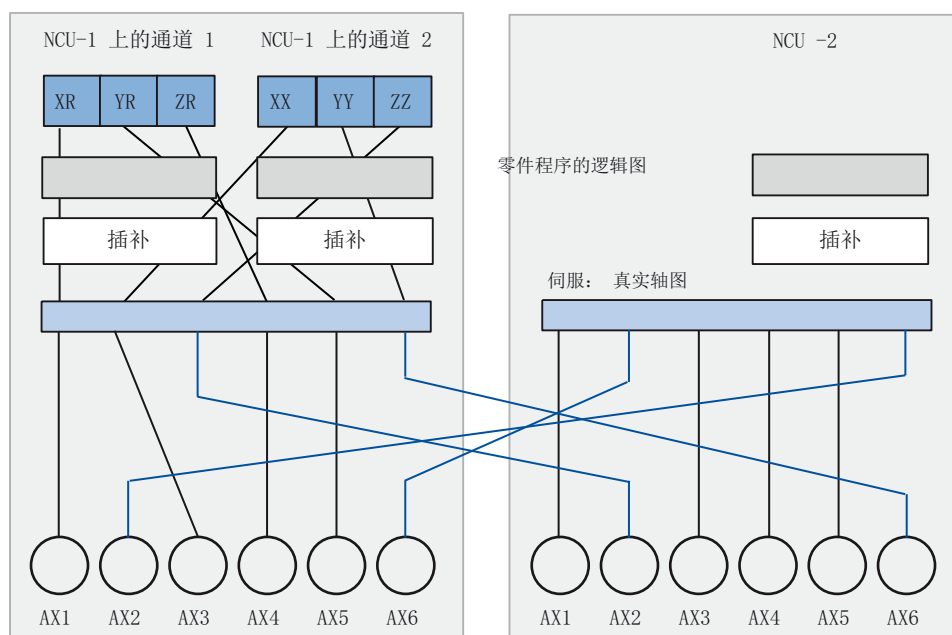


图 4-10 配置 1: 从通道到真实轴的 NCU-Link

配置 1 的机床数据

```
; ##### NCU1 #####
$MN_NCU_LINKNO = 1
$MN_MM_NCU_LINK_MASK = 1
$MN_MM_LINK_NUM_OF_MODULES= 2
$MN_MM_SERVO_FIFO_SIZE = 3

$MN_ASSIGN_CHAN_TO_MODE_GROUP[1]=1

$MN_AXCONF_LOGIC_MACHAX_TAB[0] = "NC1_AX1"
$MN_AXCONF_LOGIC_MACHAX_TAB[1] = "NC1_AX3"
$MN_AXCONF_LOGIC_MACHAX_TAB[2] = "NC2_AX2"
$MN_AXCONF_LOGIC_MACHAX_TAB[3] = "NC1_AX4"
$MN_AXCONF_LOGIC_MACHAX_TAB[4] = "NC1_AX5"
$MN_AXCONF_LOGIC_MACHAX_TAB[5] = "NC2_AX6"
```

```

CHANDATA (1)

$MC_AXCONF_MACHAX_USED[0]=1
$MC_AXCONF_MACHAX_USED[1]=5
$MC_AXCONF_MACHAX_USED[2]=4
$MC_AXCONF_MACHAX_USED[3]=0
$MC_AXCONF_MACHAX_USED[4]=0
$MC_AXCONF_MACHAX_USED[5]=0
$MC_AXCONF_CHANAX_NAME_TAB[0] = "XR"
$MC_AXCONF_CHANAX_NAME_TAB[1] = "YR"
$MC_AXCONF_CHANAX_NAME_TAB[2] = "ZR"

```

```

CHANDATA (2)

$MC_REFP_NC_START_LOCK=0
$MC_AXCONF_MACHAX_USED[0]=2
$MC_AXCONF_MACHAX_USED[1]=6
$MC_AXCONF_MACHAX_USED[2]=3
$MC_AXCONF_MACHAX_USED[3]=0
$MC_AXCONF_MACHAX_USED[4]=0
$MC_AXCONF_MACHAX_USED[5]=0
$MC_AXCONF_CHANAX_NAME_TAB[0] = "XX"
$MC_AXCONF_CHANAX_NAME_TAB[1] = "YY"
$MC_AXCONF_CHANAX_NAME_TAB[2] = "ZZ"

M30

```

```

; ##### NCU-2 #####

$MN_NCU_LINKNO = 2

$MN_MM_NCU_LINK_MASK = 1

$MN_MM_LINK_NUM_OF_MODULES= 2

```

```
$MN_MM_SERVO_FIFO_SIZE = 3  
  
;  
  
;  
  
$MN_AXCONF_LOGIC_MACHAX_TAB[0] = "NC2_AX1"  
$MN_AXCONF_LOGIC_MACHAX_TAB[1] = "NC1_AX6"  
$MN_AXCONF_LOGIC_MACHAX_TAB[2] = "NC2_AX3"  
$MN_AXCONF_LOGIC_MACHAX_TAB[3] = "NC2_AX4"  
$MN_AXCONF_LOGIC_MACHAX_TAB[4] = "NC2_AX5"  
$MN_AXCONF_LOGIC_MACHAX_TAB[5] = "NC1_AX2"  
  
CHANDATA(1)  
$MC_AXCONF_MACHAX_USED[0]=1  
$MC_AXCONF_MACHAX_USED[1]=2  
$MC_AXCONF_MACHAX_USED[2]=3  
$MC_AXCONF_MACHAX_USED[3]=4  
$MC_AXCONF_MACHAX_USED[4]=5  
$MC_AXCONF_MACHAX_USED[5]=6  
$MC_AXCONF_MACHAX_USED[6]=0  
M30
```

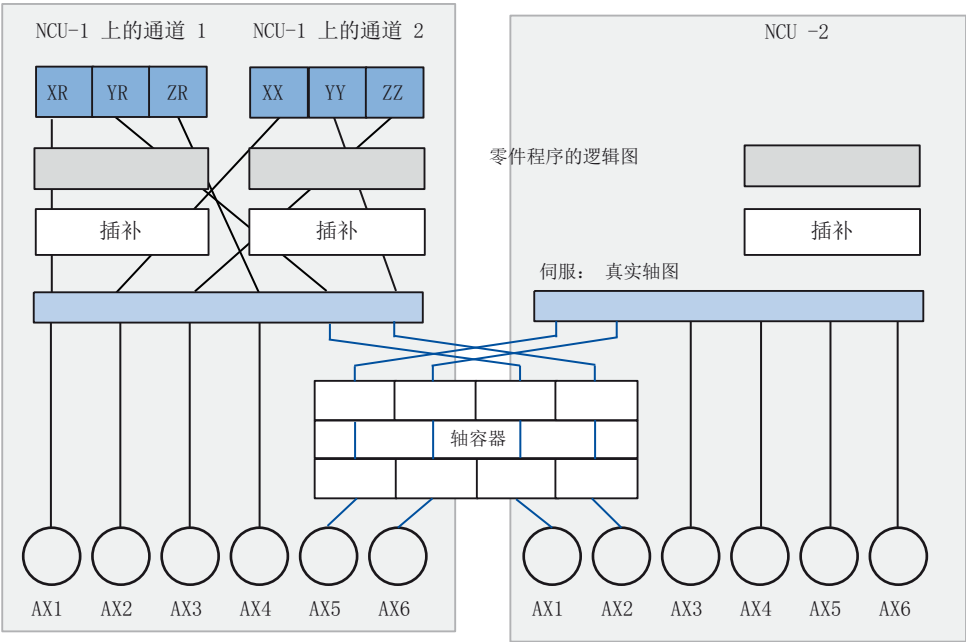


图 4-11 配置 2：轴容器处于初始状态的 NCU-Link

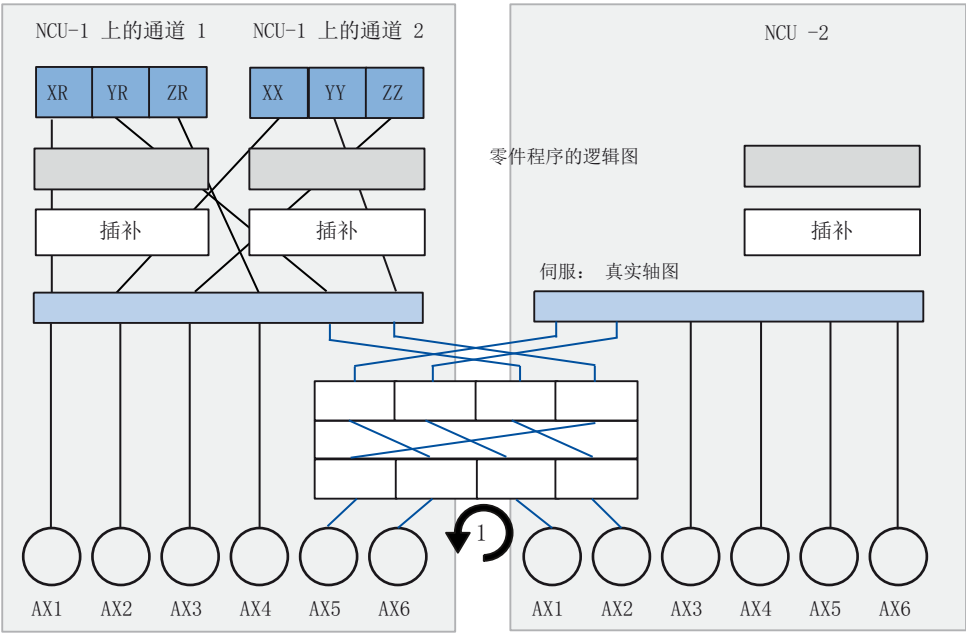


图 4-12 配置 3：轴容器经过旋转的 NCU-Link

配置 2 的机床数据

; ##### NCU1 #####

```
$MN_NCU_LINKNO = 1

$MN_MM_NCU_LINK_MASK = 1

$MN_MM_LINK_NUM_OF_MODULES= 2

$MN_MM_SERVO_FIFO_SIZE = 3


$MN_ASSIGN_CHAN_TO_MODE_GROUP[1]=1


$MN_AXCONF_LOGIC_MACHAX_TAB[0] = "NC1_AX1"
$MN_AXCONF_LOGIC_MACHAX_TAB[1] = "NC1_AX3"
$MN_AXCONF_LOGIC_MACHAX_TAB[2] = "NC2_AX2"
$MN_AXCONF_LOGIC_MACHAX_TAB[3] = "NC1_AX4"
$MN_AXCONF_LOGIC_MACHAX_TAB[4] = "CT1_SL3"
$MN_AXCONF_LOGIC_MACHAX_TAB[5] = "CT1_SL4"


$MN_AXCT_AXCONF_ASSIGN_TAB1[0] = "NC1_AX5"
$MN_AXCT_AXCONF_ASSIGN_TAB1[1] = "NC1_AX6"
$MN_AXCT_AXCONF_ASSIGN_TAB1[2] = "NC2_AX1"
$MN_AXCT_AXCONF_ASSIGN_TAB1[3] = "NC2_AX2"


$SN_AXCT_SWWIDTH[0] = 1


CHANDATA(1)

$MC_AXCONF_MACHAX_USED[0]=1
$MC_AXCONF_MACHAX_USED[1]=5
$MC_AXCONF_MACHAX_USED[2]=4
$MC_AXCONF_MACHAX_USED[3]=0
$MC_AXCONF_MACHAX_USED[4]=0
$MC_AXCONF_MACHAX_USED[5]=0

$MC_AXCONF_CHANAX_NAME_TAB[0] = "XR"
```

4.4 插补补偿

```

$MC_AXCONF_CHANAX_NAME_TAB[1] = "YR"
$MC_AXCONF_CHANAX_NAME_TAB[2] = "ZR"

CHANDATA(2)

$MC_REFP_NC_START_LOCK=0
$MC_AXCONF_MACHAX_USED[0]=2
$MC_AXCONF_MACHAX_USED[1]=6
$MC_AXCONF_MACHAX_USED[2]=3
$MC_AXCONF_MACHAX_USED[3]=0
$MC_AXCONF_MACHAX_USED[4]=0
$MC_AXCONF_MACHAX_USED[5]=0
$MC_AXCONF_CHANAX_NAME_TAB[0] = "XX"
$MC_AXCONF_CHANAX_NAME_TAB[1] = "YY"
$MC_AXCONF_CHANAX_NAME_TAB[2] = "ZZ"
M30

; ##### NCU-2 #####
$MN_NCU_LINKNO = 2
$MN_MM_NCU_LINK_MASK = 1
$MN_MM_LINK_NUM_OF_MODULES= 2
$MN_MM_SERVO_FIFO_SIZE = 3

$MN_AXCONF_LOGIC_MACHAX_TAB[0] = "CT1_SL1"
$MN_AXCONF_LOGIC_MACHAX_TAB[1] = "CT1_SL2"
$MN_AXCONF_LOGIC_MACHAX_TAB[2] = "NC2_AX3"
$MN_AXCONF_LOGIC_MACHAX_TAB[3] = "NC2_AX4"
$MN_AXCONF_LOGIC_MACHAX_TAB[4] = "NC2_AX5"
$MN_AXCONF_LOGIC_MACHAX_TAB[5] = "NC2_AX6"

```

```
CHANDATA(1)

$MC_AXCONF_MACHAX_USED[0]=1
$MC_AXCONF_MACHAX_USED[1]=2
$MC_AXCONF_MACHAX_USED[2]=3
$MC_AXCONF_MACHAX_USED[3]=4
$MC_AXCONF_MACHAX_USED[4]=5
$MC_AXCONF_MACHAX_USED[5]=6
$MC_AXCONF_MACHAX_USED[6]=0

M30
```

4.4.6 插补补偿的特殊性

测量

“测量”功能向操作人员或编程人员提供经过补偿的实际位置（理想机床）。

TEACH IN

即使在“TEACH IN”功能中，待保存的实际位置也根据经过补偿的位置值确定。

软件限位开关

软件限位开关同样监控的是理想的位置值（即经 MSEC 和间隙补偿修正过的位置实际值）。

位置显示

机床坐标系中的位置实际值显示是轴（理想机床）的理想（编程）位置实际值。

在“进给轴/主轴”信息画面（操作区域“诊断”）中显示的是测量系统实测的位置实际值加上 MSEC 和间隙补偿后所得到的值（= 测量系统 1/2 的位置实际值）。

4.5 动态前馈控制（跟随误差补偿）

补偿值显示

在“进给轴/主轴”信息画面（操作区域“诊断”）中会显示以下补偿值：

- 测量系统 1 或 2 的绝对补偿值
显示数值为对应轴当前实际位置（测量系统 1 或 2）的 MSEC 和间隙补偿值的总和。
- 垂度补偿 + 温度补偿
显示值为对应轴当前位置的垂度补偿和温度补偿值的总和。

参考点丢失

若参照轴的参考点丢失（DB31, ... DBX60.4 或 60.5 = 0），则相关轴上的 MSEC 和垂度补偿关闭。重新到达参考点后，这些补偿将自动重新激活。

存取保护

目前对于补偿表尚无存取保护功能。

伺服使能设置

由于补偿关系的存在，参照轴的运行可能也会导致补偿轴的运行，因此同样需要为这些轴设置伺服使能（PLC 用户程序）。否则，补偿功能会受到一定限制。

运行信号的输出

激活或关闭补偿功能、以及每次更改有效补偿表格数量时，系统都会输出补偿轴的运行信号。

由于参照轴运行致使补偿值发生变化时，系统不会输出补偿轴的运行信号。

4.5 动态前馈控制（跟随误差补偿）

4.5.1 一般特性

轴跟随误差

轴跟随误差是机床轴运行时位置控制器的控制偏差。换句话说，轴跟随误差是机床轴设定位置 and 实际位置间的差值。

影响

跟随误差容易导致轮廓误差（误差大小取决于速度），特别在曲面轮廓（例如圆弧和拐角）上加速时。

补偿

使用“动态前馈控制”可以将轴跟随误差几乎降到零。因此此功能也称作“跟随误差补偿”。

方法

系统提供两种“动态前馈控制”方式：

- 转速前馈控制（取决于速度）
- 转矩前馈控制（取决于加速度）

激活

前馈控制方式通过以下机床数据选择和激活：

MD32620 \$MA_FFW_MODE（前馈控制方式）

值	含义
0	无前馈控制
1	采用 PT1 对称的转速前馈控制
2	采用 PT1 对称的转矩前馈控制
3	采用 Tt 对称的转速前馈控制
4	采用 Tt 对称的转矩前馈控制

在零件程序中激活/取消前馈控制

可通过以下轴专用机床数据定义，是否能够通过零件程序为某进给轴/主轴激活和取消前馈控制：

MD32630 \$MA_FFW_ACTIVATION_MODE（通过程序激活前馈控制）

4.5 动态前馈控制（跟随误差补偿）

值	含义
0	无法通过零件程序激活或取消前馈控制。进给轴/主轴处于 MD32620 \$MA_FFW_MODE 设定的状态。
1	可通过零件程序激活或取消前馈控制。指令立即生效。
2	可通过零件程序激活或取消前馈控制。指令在轴下一次达到静止状态时才生效。

通过零件程序激活/取消前馈控制须使用以下指令：

FFWON：前馈控制 ON

FFWOF：前馈控制 OFF

缺省设置（即复位或 M30 后的设置）通过以下通道专用机床数据设定：

MD20150 \$MC_GCODE_RESET_VALUES（G 组的缺省设置）

采用以下设置时，FFWON/FFWOF 对所有进给轴/处于进给轴模式的主轴生效：

MD32630 \$MA_FFW_ACTIVATION_MODE = 1（或 2）

与

MD32620 \$MA_FFW_MODE = 1、2、3 或 4

对于相互插补的轴，MD32630 应设置为相同的值。

为了防止产生急动，只能在进给轴/处于进给轴模式的主轴静止时激活或取消前馈控制。因此切换会通过预处理停止自动延迟至下一次静止状态时执行。

注意
对于和零件程序执行不同步的指令轴或 PLC 轴而言，预处理停止无效。为了确保 FFWON/FFWOF 在进给轴/处于进给轴模式的主轴静止时才生效，必须为每根进给轴/处于进给轴模式的主轴显性设置 MD32630 = 2（另见“指令轴和 PLC 轴的前馈控制 (页 303)”）。

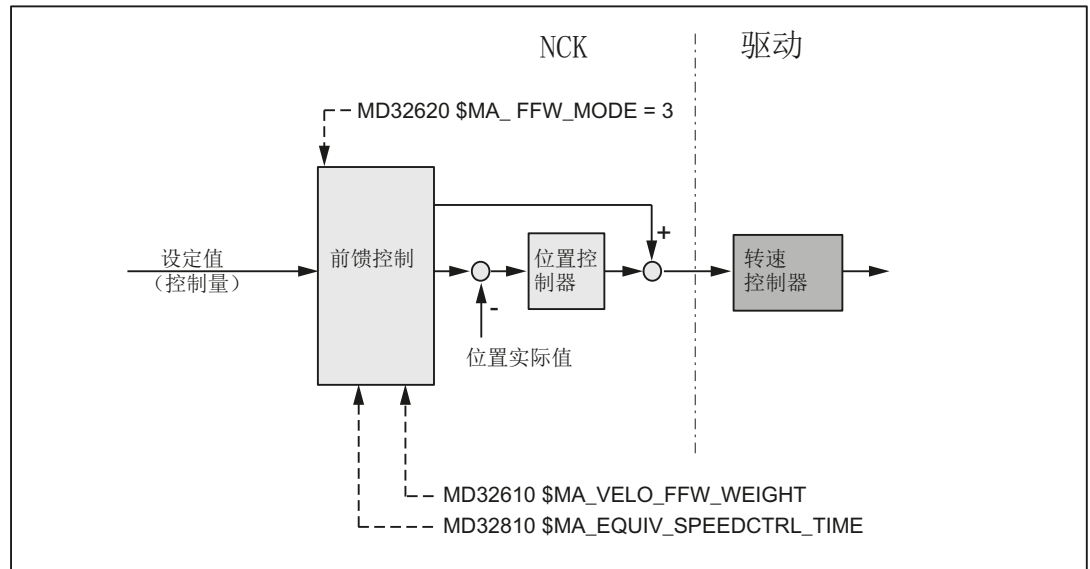
4.5.2 转速前馈控制

功能

使用转速前馈控制时，需要在转速控制器的输入端附加一个速度设定值。因此在恒速条件下可几乎完全消除跟随误差（即控制偏差为 0）。

调试

调试时，必须为转速前馈控制定义下列轴专用参数：



转速环的等效时间常数（MD32810）

为正确设置转速前馈，需要准确确定转速环的等效时间常数，例如通过观察转速设定值阶跃响应的曲线图，然后将其输入到以下机床数据中：

MD32810 \$MA_EQUIV_SPEEDCTRL_TIME（用于前馈控制的转速环等效时间常数）

转速前馈控制系数（MD32610）

附加速度设定值可设置一个权重系数：

MD32610 \$MA_VELO_FFW_WEIGHT

取值范围：0 ... 1

“0”表示无前馈控制。缺省设置下此系数为 1（ $\pm 100\%$ ）。

建议保留 100% 的设置，因为在进给轴/主轴的控制环经过最佳设置以及精确确定了转速环等效时间常数的条件下，此数值为最优设置。

微调

通过微调转速环的等效时间常数（MD32810）可优化对应进给轴/主轴的转速前馈控制。

如需检查前馈效果，可恒速运行运行进给轴/主轴，然后在“进给轴/主轴”信息画面中检查“控制偏差”。

建议选择低加速度和高进给率，以便更好地检查前馈效果。该设置可延长加速阶段，更加清晰地看出控制偏差的变化。

4.5 动态前馈控制（跟随误差补偿）

示例：

MD32300 \$MA_MAX_AX_ACCEL = 0.1 ； 最大轴加速度 = 0.1 m/s²
MD32000 \$MA_MAX_AX_VELO = 20000.0 ； 最大轴速度
 = 20000.0 mm/min

```
； 用于设置等效时间常数的零件程序
G1 F20000
FWON
LOOP:
X1000
X0
GOTOB LOOP
M30
```

文档

转速环等效时间常数（MD32810）的详细设置说明请见：

- 功能手册之基本功能分册；章节“速度，设定值/实际值系统，闭环控制（G2）”：“控制环优化”

4.5.3 转矩前馈控制

功能

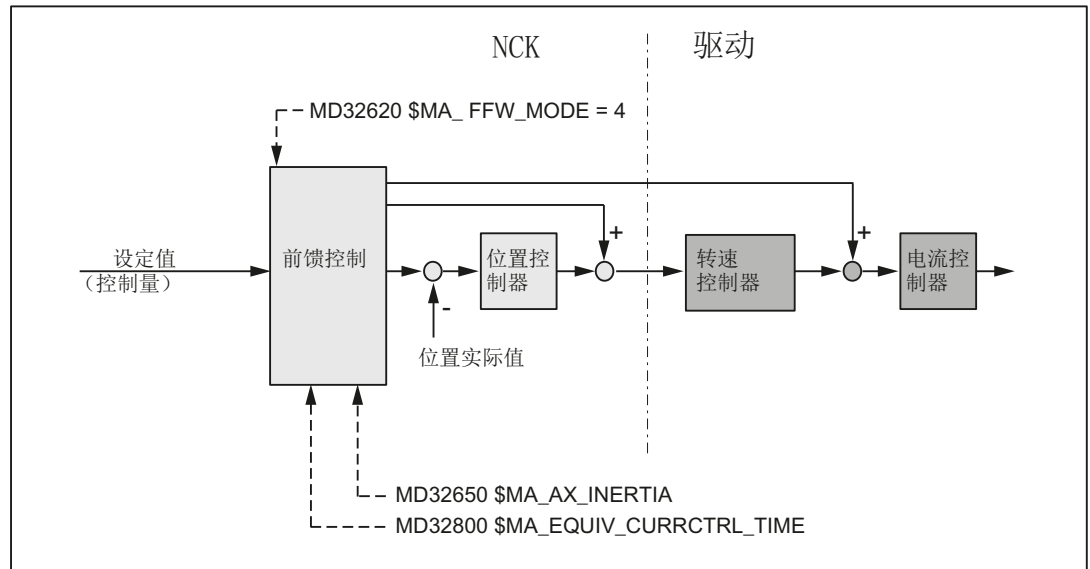
使用转矩前馈控制时，和转矩成正比的附加电流设定值会附加到电流控制器的输入上。此值由加速度和转动惯量计算得出。

应用

转矩前馈控制可满足高轮廓精度要求和最高的动态响应要求。因此当该功能设置正确时，即使在加速过程中也可几乎完全消除掉跟随误差。

调试

调试时，必须为转矩前馈控制定义下列轴专用参数：



电流环的等效时间常数（MD32800）

为正确设置转矩前馈，需要准确确定转速环的等效时间常数，例如通过观察转速设定值阶跃响应的曲线图，然后将其输入到以下机床数据中：

MD32800 \$MA_EQUIV_CURRCTRL_TIME（用于前馈控制的电流环等效时间常数）

轴的总转动惯量（MD32650）

采用转矩前馈控制时，必须测定轴的总转动惯量（驱动转动惯量和负载相对于电机轴的转动惯量），并将其输入以下机床数据：

MD32650 \$MA_AX_INERTIA（转矩前馈控制的转动惯量）

微调

通过微调 MD32800 和 MD32650 的值可优化对应进给轴/主轴的转矩前馈控制。

建议使用跟踪功能记录跟随误差以备检查优化效果。除了观察进给轴/主轴的恒速运行外，此时尤其需要观察加速过程中的跟随误差。

说明

由于加速太快，在调试转矩前馈时无法使用信息画面来检查微调的效果。

4.5 动态前馈控制（跟随误差补偿）

文档

电流环等效时间常数（MD32810）的详细设置说明请见：

- 功能手册之基本功能分册；章节“速度，设定值/实际值系统，闭环控制（G2）”：“控制环优化”

4.5.4 动态响应自适应功能

功能

如果相互插补的轴的控制环的时间特性不同，可通过动态响应自适应获得相同的时间特性，以便达到最佳的轮廓精度并保持良好的控制性能。

调试

动态响应自适应的时间常数（MD32910）

动态响应自适应的时间常数是各轴转速环/电流环的等效时间常数与“响应最慢”的控制环等效时间之间的差值，此值必须输入以下机床数据中。

MD32910 \$MA_DYN_MATCH_TIME（动态响应自适应的时间常数）

示例：

转速前馈控制生效时，轴 1、2、3 转速环的等效时间常数（MD32810）：

- 轴 1：2 ms
- 轴 2：4 ms（为动态响应最慢的轴）
- 轴 3：1 ms

由此得出动态响应自适应的时间常数（MD32910），如下所示：

- 轴 1：2 ms
- 轴 2：0 ms
- 轴 3：3 ms

激活（MD32900）

仅在设置了以下机床数据时，动态响应自适应才生效：

MD32900 \$MA_DYN_MATCH_ENABLE = 1

文档

功能手册之基本功能分册；章节“速度，设定值/实际值系统，闭环控制（G2）”：“控制环优化”

4.5.5 指令轴和 PLC 轴的前馈控制

功能

使用指令轴和 PLC 轴时须采用以下设置，防止前馈控制在高速条件下被激活/取消：

MD32630 \$MA_FFW_ACTIVATION_MODE = 2

采用此设置时，FFWON/FFWOF 仅在低于为此轴配置的静态速度（MD36060 \$MA_STANDSTILL_VELO_TOL）的情况下生效。

如果在轴运行期间发出激活/取消命令，则只有在轴接着停止后，该命令才会生效，以防止跟随误差急剧形成/消减。

注意
若静态速度设置得过高，可能会导致运行中便激活/取消前馈控制。取决于当前跟随误差的大小，监控功能可能会响应。

调试

为指令轴和 PLC 轴检查前馈控制时，建议采用以下步骤：

1. 检查 MD36060 中的静态速度。
2. 检查静止状态下的轴跟随误差。
3. 设置切换条件并使能：
MD32630 = 2
4. 通过零件程序中的 POSA 指令运行轴。
5. 在轴运行期间执行 FFWON。
6. 此时“进给轴/主轴”信息画面中显示的 K_v 系数和跟随误差不允许出现阶跃。
7. 高 K_v 系数和低跟随误差只有在静止状态之后的运行中才出现。前馈控制从静止状态起才生效。

取消前馈控制的步骤与激活类似：

1. 通过零件程序中的 POSA 指令运行轴。
2. 在轴运行期间执行 FFWOF。

4.5 动态前馈控制（跟随误差补偿）

- 3. 此时“进给轴/主轴”信息画面中显示的 K_v 系数和跟随误差不允许出现阶跃。
- 4. 低 K_v 系数和高跟随误差只有在静止状态之后的运行中才出现。前馈控制从静止状态起才无效。

示例

在下面的程序示例中轴 **A** 和轨迹不同步。运行期间尝试在通道中激活前馈控制。与几何轴 **X**、**Y**、**Z** 不同，前馈控制不立即对轴 **A** 生效。此时会等待程序段 **N60** 后的静止状态。在程序段 **N70** 中轴 **A** 将采用前馈控制运行。

程序代码
N10 FFWOF
N20 POSA[A]=1000 FA[A]=10000
N30 G4 F1
N40 FFWON
N50 G0 X10 Y10 Z10
N60 WAITP(A)
N70 POSA[A]=1500 FA[A]=10000
N80 WAITP(A)
M30

4.5.6 边界条件

相互插补的轴

在相互插补的轴上同样需要为**每根轴**设置最佳的前馈控制参数，即相互插补的轴也可具有不同的前馈控制参数。

检查轮廓监控

由于两个等效时间常数：

- MD32810 \$MA_EQUIV_SPEEDCTRL_TIME（用于前馈控制的转速环等效时间常数）
- MD32800 \$MA_EQUIV_CURRCTRL_TIME（用于前馈控制的电流环等效时间常数）

也会影响轮廓监控，必须随后检查这两个数据。

文档：

功能手册之基本功能分册；章节“轴监控，保护区 (A3)”

对 K_v 系数的影响

在前馈控制设置正确时，被控对象对控制量的响应与转速环的动态响应（转速前馈控制时）或电流环的动态响应（转矩前馈控制时）相同，即 MD32200 \$MA_POS_CTRLGAIN 中输入的 K_v 系数对控制特性（例如拐角误差、超调、圆弧/半径误差）几乎无影响。

另一方面，前馈控制对抗干扰性（同步运行）也无影响。此时 MD32200 中设定的 K_v 系数生效。

“ K_v 系数”信息画面

前馈控制生效时，“进给轴/主轴”信息画面中显示的轴 K_v 系数（对控制性能产生影响的 K_v 系数）非常大。

4.6 摩擦补偿（象限误差补偿）

4.6.1 一般功能说明

除了转动惯量和切削力以外，齿轮和机床导轨中的摩擦力也会影响机床轴的运行。轴从静止开始加速时，尤其是从静摩擦转换为明显降低的滑动摩擦时，会对轮廓精度产生不良影响。

此时出现的摩擦力跃变会引起短暂的跟随误差增大。在插补轴上，这会导致明显的轮廓误差。在加工圆形轮廓时，由于在转换方向时一根轴静止，轮廓误差特别突出，尤其在过象限处。

激活摩擦补偿或象限补偿后，当轴从静止状态开始加速，即从静摩擦转换为滑动摩擦时，系统会作为补偿值加入附加的设定值脉冲。由此可以几乎完全消除圆形轮廓上过象限处的轮廓误差。

和加速度相关的摩擦补偿

在多数情形下，通过一个恒定幅值的、与轴加速度无关的补偿值即可实现象限误差补偿。但是，若补偿值与加速度相关，可通过“带自适应的摩擦补偿”激活一条自适应特性曲线，为补偿建模型。

圆度测试

调试摩擦补偿时，最简单的方法是使用操作界面中集成的圆度测试功能。此时系统会采集加工圆弧期间各个机床轴实际位置上产生的圆轮廓，然后将该轮廓和编程的轮廓对比，用示意图展示两者的差别，尤其是过象限处的差别。

4.6 摩擦补偿（象限误差补偿）

圆度测试在操作界面上通过以下菜单调用：

SINUMERIK Operate

操作区域 “调试” > “优化/测试” > “圆度测试”

4.6.2 边界条件

注意

取消设定值相关补偿

以下补偿对位置设定值生效，因此要关闭圆度测试中涉及的轴的这些补偿，然后再开始测试。

- 垂度补偿和角度补偿（CEC）
MD32710 \$MA_CEC_ENABLE[<轴>] = 0
- 双向丝杠螺距误差补偿：
MD32710 \$MA_CEC_ENABLE[<轴>] = 0
- 温度补偿：
MD32750 \$MA_TEMP_COMP_TYPE[<轴>] = 0

4.6.3 补偿值恒定的摩擦补偿

4.6.3.1 功能激活

使能

摩擦补偿通过以下机床数据激活：

MD32490 \$MA_FRICT_COMP_MODE[<轴>] = 1

激活

补偿值恒定的摩擦补偿通过以下机床数据激活：

- MD32500 FRICT_COMP_ENABLE[<轴>] = 1（摩擦补偿激活）
- MD32510 \$MA_FRICT_COMP_ADAPT_ENABLE[<轴>] = 0（自适应关闭）

参数

补偿值恒定的摩擦补偿会考量以下参数:

- MD32520 \$MA_FRICT_COMP_CONST_MAX (最大补偿值)
采用补偿值恒定的摩擦补偿时, 参数设置的值作为补偿值激活。
- MD32540 \$MA_FRICT_COMP_TIME (摩擦补偿时间常数)
补偿值通过一个 DT1 滤波器激活。补偿值在设置的时间段后变为零。

4.6.3.2 调试

圆度测试

如上文所述, 调试补偿值恒定的摩擦补偿时建议使用圆度测试。调试的过程可分为如下步骤:

1. 执行**无**摩擦补偿的圆度测试
2. 执行**带**摩擦补偿的、采用低参数值的圆度测试
3. 执行**带**摩擦补偿的、采用调高的参数值的圆度测试
4. 完成**带**摩擦补偿的、采用最佳参数值的圆度测试

无摩擦补偿的圆度测试

为了测定过象限处圆弧轮廓的初始质量, 必须执行无摩擦补偿的圆度测试。为此必须事先取消摩擦补偿:

MD32500 FRICT_COMP_ENABLE[<轴>] = 0

下图显示了无摩擦补偿时过象限处的典型情形:

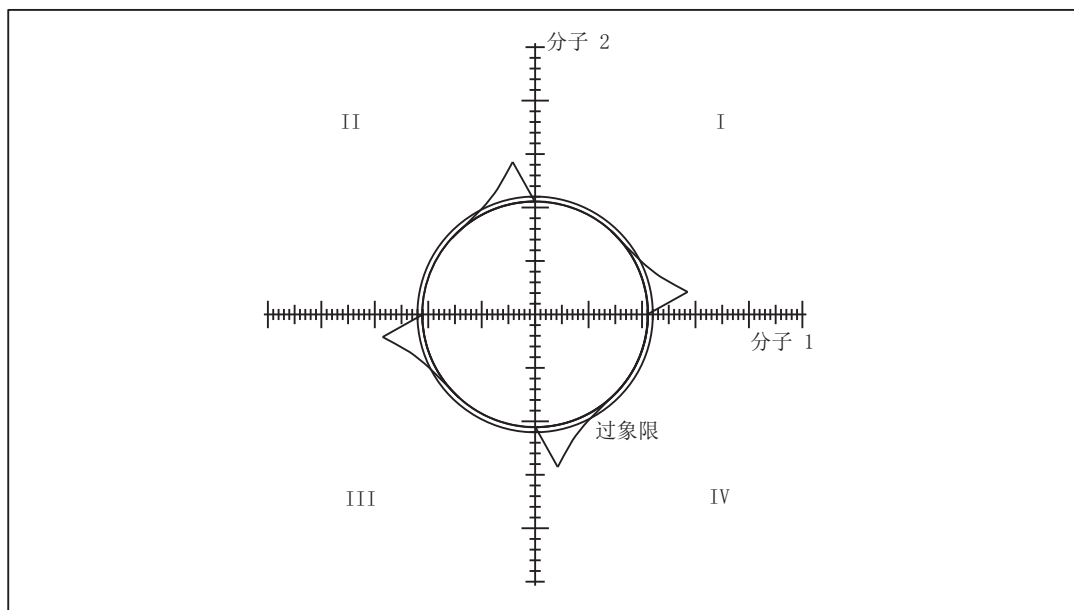


图 4-13 无摩擦补偿时的过象限

之后必须重新启用补偿值恒定的摩擦补偿：

MD32500 FRICT_COMP_ENABLE[<轴>] = 1

带摩擦补偿的、采用低参数值的圆度测试

建议使用低补偿值和低时间常数（持续仅几个位置控制器周期）开展测试，例如：

- MD32520 \$MA_FRICT_COMP_CONST_MAX[<轴>] = 10 [mm/min]
- MD32540 \$FRICT_COMP_TIME[<轴>] = 0.008 [ms]

使用这些参数值进行圆度测试后，可对摩擦补偿的效果进行初步评估。

补偿值过小

在圆度测试中，若过象限处的轮廓偏差得到的补偿不足，则表示补偿值（MD32520）过小。

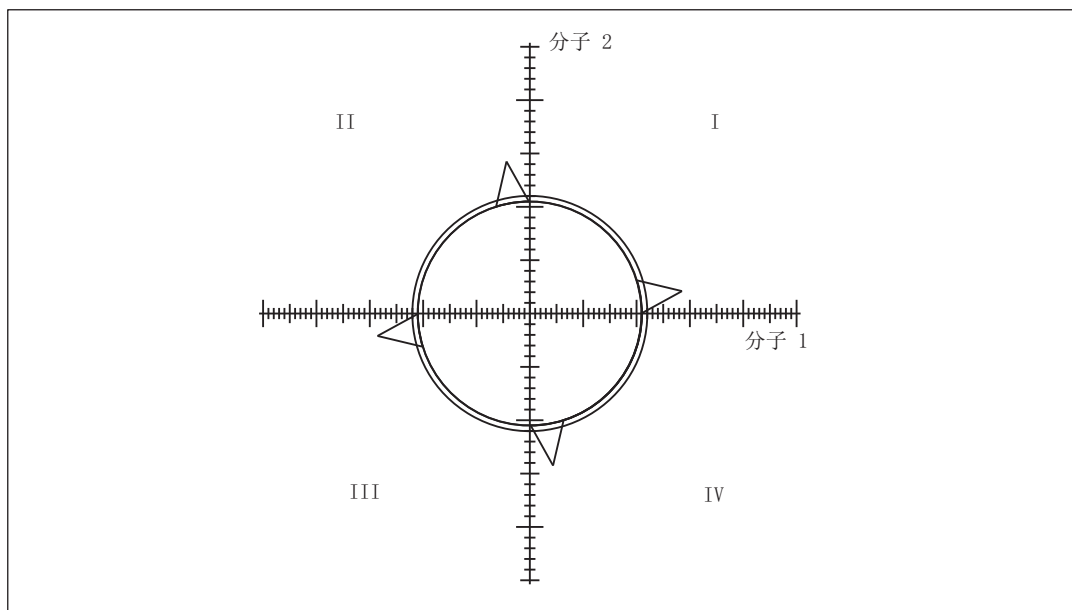


图 4-14 补偿值设置过小

补偿值过大

在圆度测试中，若过象限处的轮廓偏差过补偿，则表示补偿值（MD32520）过大。

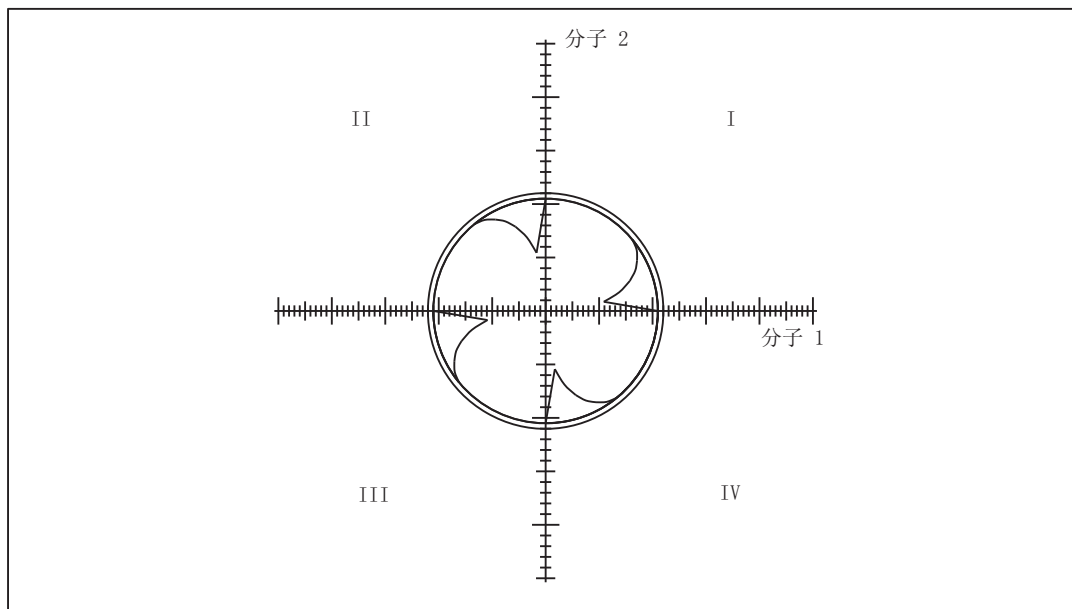


图 4-15 补偿值设置过大

时间常数过小

在圆度测试中，若过象限处的轮廓偏差得到短暂补偿，之后又立即增大，则表示时间常数（MD32540）过小。

4.6 摩擦补偿（象限误差补偿）

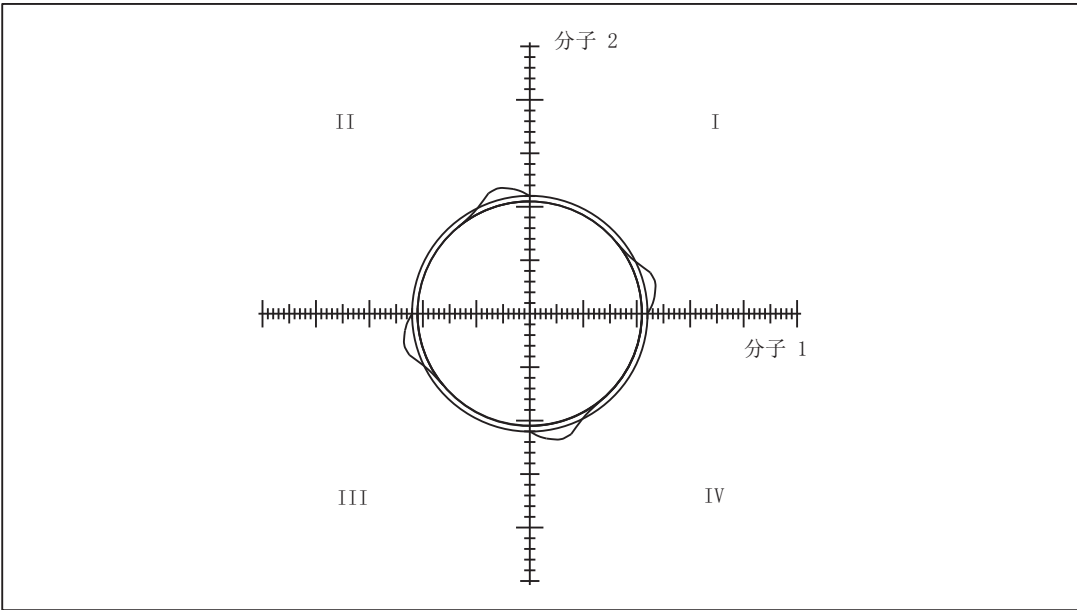


图 4-16 补偿时间常数设置过小

时间常数过大

在圆度测试中，如果一开始误差得以补偿，但后续轮廓上出现过度补偿，而补偿值用的是最佳补偿值，则表明时间常数（MD32540）设置过大。它导致补偿值继续作用，从而出现过度补偿。

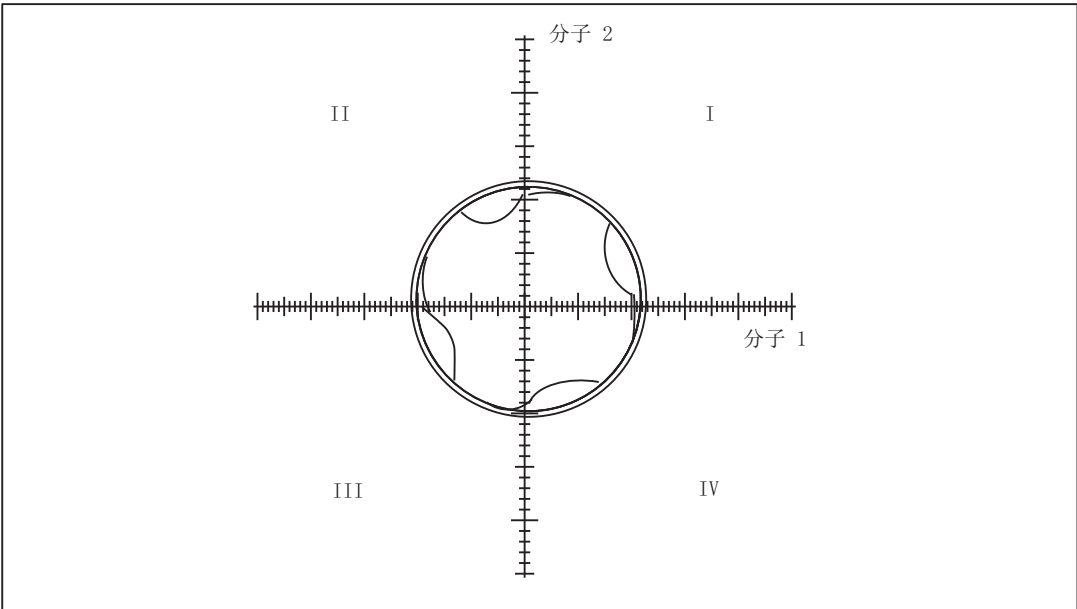


图 4-17 时间常数设置过大

补偿参数的优化

必须通过微调参数值、多次重复圆度测试来获得最佳的补偿参数。建议用机床加工中各种典型的半径和轨迹速度来进行测试。

每次调整参数后，都要接着进行圆度测试，并检查和记录测试结果。

求平均值

若采用不同的半径和轨迹速度时所得到的参数值不同，则必须通过求平均值确定最适宜的值。

正确设置的摩擦补偿

摩擦补偿正确设置后，过象限处“不”再观察到轮廓误差。

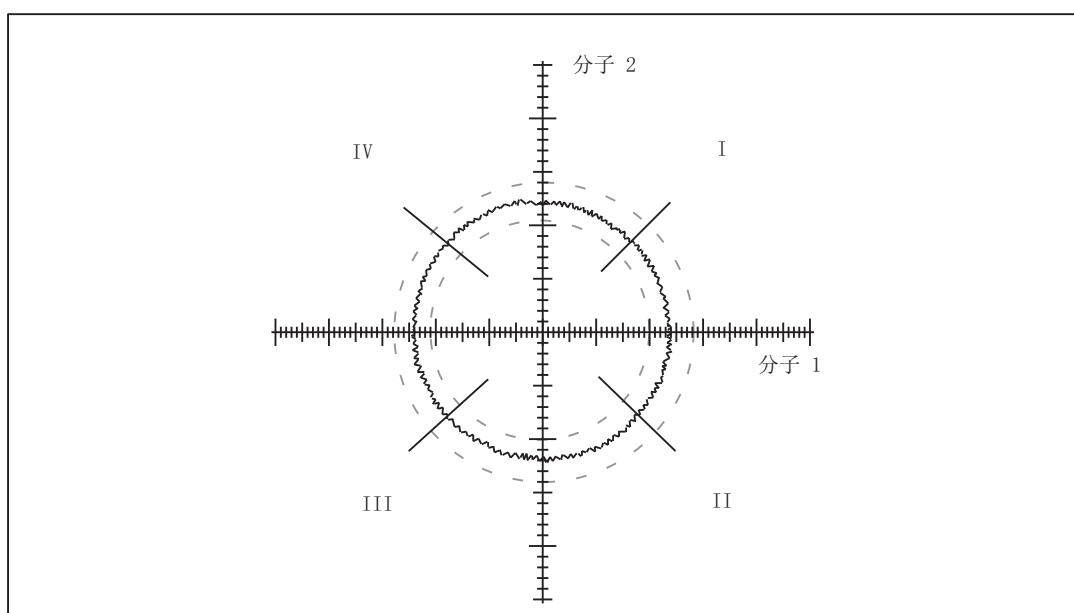


图 4-18 正确设置的摩擦补偿

和加速度相关的补偿值

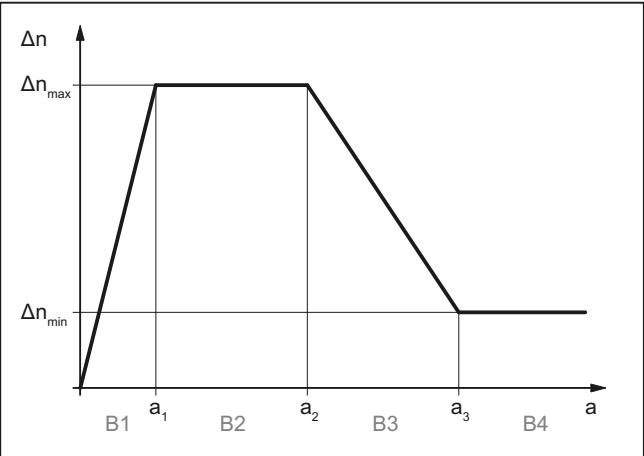
若补偿值为加速度相关，可于随后的步骤中启用下文所述的“带自适应的摩擦补偿”。

4.6 摩擦补偿（象限误差补偿）

4.6.4 补偿值和加速度相关的摩擦补偿

4.6.4.1 功能说明

当补偿值受到加速度的显著影响时，通常情况下高加速度条件下最佳的补偿值对于低加速度条件来说必然太低。 这种相关性可以通过下面的自适应曲线来建模。



- $\Delta n_{\max}$ 最大补偿值
- $\Delta n_{\min}$ 最小补偿值
- a_1 加速度值 1
- a_2 加速度值 2
- a_3 加速度值 3
- B_n 加速阶段，其中 $n = 1, 2, \dots 4$
- 其 加速度: $a_1 < a_2 < a_3$
- 中: 补偿值: $\Delta n_{\max} > \Delta n_{\min}$ ，在特殊情形下也可能 $\Delta n_{\max} < \Delta n_{\min}$

根据对应的加速阶段 B1 至 B4，补偿值 Δn 如下计算：

区域	其中加速度 a	$\Rightarrow$ 补偿值 Δn
B1	$a < a_1$	$\Delta n = \Delta n_{\max} * a / a_1$
B2	$a_1 \leq a \leq a_2$	$\Delta n = \Delta n_{\max}$
B3	$a_2 < a < a_3$	$\Delta n = \Delta n_{\max} + [(\Delta n_{\min} - \Delta n_{\max}) / (a_3 - a_2)] * (a - a_2)$
B4	$a \geq a_3$	$\Delta n = \Delta n_{\min}$

4.6.4.2 功能激活

使能

摩擦补偿通过以下机床数据激活:

MD32490 \$MA_FRICT_COMP_MODE[<轴>] = 1

激活

带自适应特性曲线的摩擦补偿通过以下机床数据激活:

- MD32500 FRICT_COMP_ENABLE[<轴>] = 1 (摩擦补偿激活)
- MD32510 \$MA_FRICT_COMP_ADAPT_ENABLE[<轴>] = 1 (自适应特性曲线激活)

4.6.4.3 调试

为了测定特性曲线参数, 需要在指定加速阶段的多个工作点上确定最优补偿值 Δn_{opt} 。相关内容参见章节“调试 (页 307)”。尤其是在高轨迹速度和小半径的条件下, 须确保测量值足够多。

建议用示意图来表示得出的值对用于分析:

$\Delta n_{opt} = f(a)$, 其中 Δn_{opt} = 最优补偿值, a = 过象限处的加速度。

之后必须将由测量结果确定的自适应特性曲线的参数输入机床数据。

特性曲线参数

加速度值

换向轴过象限处的加速度如下计算:

$a = v^2 / r$, 其中 v = 轨迹速度, r = 圆弧半径

说明

通过进给倍率开关可以简单的方式调整轨迹速度和轴加速度 a 。

作为特性曲线参数测定的加速度值 a_1 、 a_2 和 a_3 必须输入以下机床数据。此时须遵循以下条件: $a_1 < a_2 < a_3$

4.6 摩擦补偿 (象限误差补偿)

- MD32550 \$MA_FRICT_COMP_ACCEL1 (加速度值 1)
- MD32560 \$MA_FRICT_COMP_ACCEL2 (加速度值 2)
- MD32570 \$MA_FRICT_COMP_ACCEL3 (加速度值 3)

补偿值

作为特性曲线参数测定的补偿值 $\Delta n_{\min}$ 、 $\Delta n_{\max}$ 必须输入以下机床数据:

- MD32520 \$MA_FRICT_COMP_CONST_MAX (最大补偿值)
- MD32530 \$MA_FRICT_COMP_CONST_MIN (最小补偿值)

说明

若在轨迹速度极小的情况下未获得满意的测量结果, 可能须提高计算精度:

- MD10200 \$MA_INT_INCR_PER_MM (线性插补的计算精度)
 - MD10210 \$MA_INT_INCR_PER_DEG (角度位置的计算精度)
-

4.6.5 适用于短程序段的补偿值

在运行程序段较短时, 测定的象限误差补偿值可能会导致过补偿。可为此类在一个插补周期内执行完毕的短程序段降低补偿值, 以避免过度补偿。但是每根轴上的降低幅度主要取决于具体的机床工况, 因此需根据经验设定。此值通过以下机床数据以圆度测试中测定的补偿值的百分比设置:

MD32580 \$MA_FRICT_COMP_INC_FACTOR (适用于短程序段的补偿值)

4.7 悬挂上的附加措施

4.7.1 电子配重

无配重的轴

在悬挂了重物的轴上，抱闸松开后，轴可能会下移，出现以下特性：

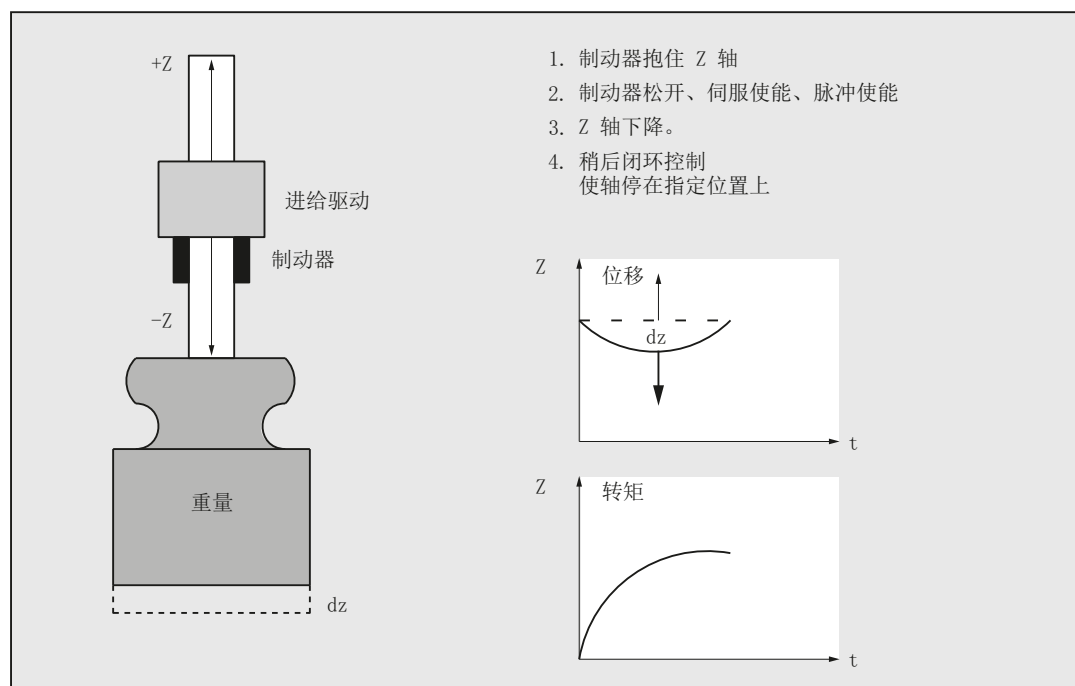


图 4-19 无配重的悬挂轴的下沉

“电子配重”功能

使用“电子配重”功能可几乎完全避免悬挂轴的下沉。

电子配重能够在闭环控制激活时避免负重轴的下沉。制动松开后，恒定的配重转矩将保持悬挂轴的位置：

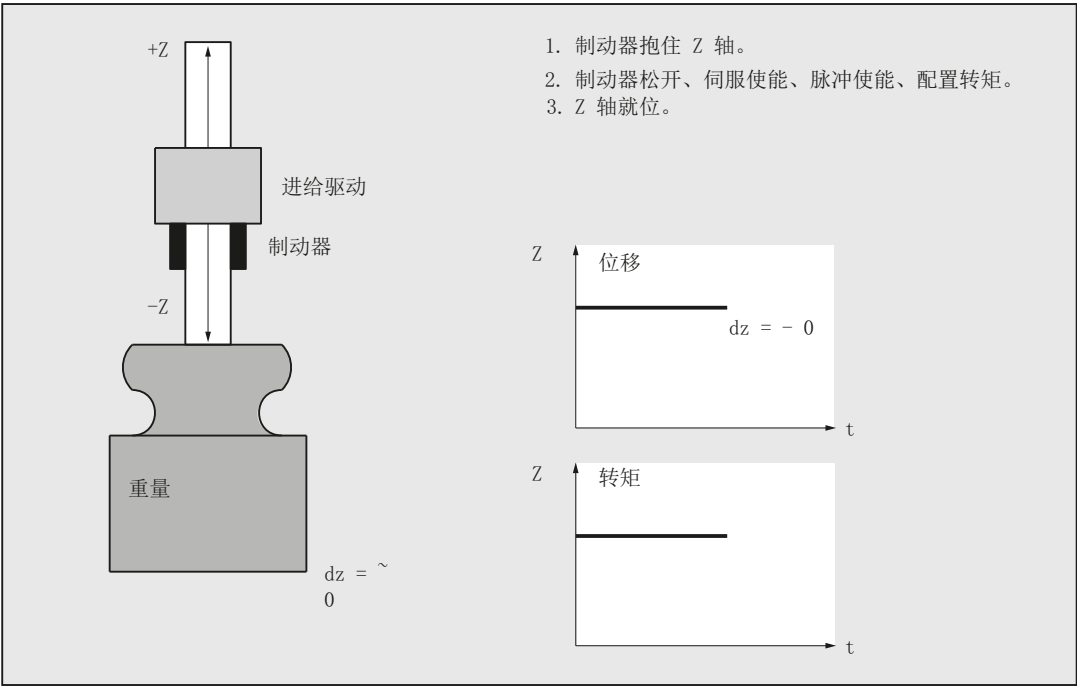


图 4-20 悬挂轴的下沉，电子配重激活

调试

说明

“电子配重”的调试通过驱动进行！

文档

更多信息参见：

SINAMICS S120 功能手册“驱动功能”

4.7.2 重启延时

通过操作界面重启的副作用

HMI 上的机床数据修改生效需要重启 NCK。此时有可能导致悬挂轴下移一段。为避免该现象，需要使用“重启延时”功能。

重启延时

通过重启延时功能可推迟 NCK 和 PLC 的关闭，并提示即将关闭的信息，从而可采取措施防止悬挂轴下移。

说明

该功能只对通过操作界面的手动重新上电生效。

掉电或硬件重启不会激活重启延时。

重启过程

操作软件通过 PI 服务“_N_IBN_SS”触发 NCK 和 PLC 的重启。

NCK 直接通过 PI 服务触发报警 2900。

在 NCK 执行 PI 服务到重启这段时间内（重启延时：MD10088 \$MN_REBOOT_DELAY_TIME），可激活机械轴制动。

报警 2900 的响应

- 以下 NC/PLC 接口信号会被清除，即置零：
 - DB11 DBX 6.3（BAG 运行就绪）；所有 BAG
 - DB21, ... DBX 36.5（通道运行就绪）；所有通道
 - DB31, ... DBX 61.2（轴运行就绪）；所有轴
- NCK 在电流限值处制动。
参见机床数据：
 - MD36610 \$MA_AX_EMERGENCY_STOP_TIME（故障状态下制动斜坡的持续时间）
 - MD36620 \$MA_SERVO_DISABLE_DELAY_TIME（伺服使能封锁延时）

说明

NCK 会在伺服使能封锁延时（MD36620）后取消位置闭环控制。

- 以下 NC/PLC 接口信号保持为 1：
DB10 DBX108.7（NC 就绪）

通过以下机床数据：

MD11410 \$MN_SUPPRESS_ALARM_MASK（特殊报警的支持位）（BIT20）

可抑制报警 2900，但是 NCK 会触发相同的响应。

由于报警 2900 会取消轴的位置闭环控制，因此必然要由 PLC 闭合机械制动。PLC 重启后所有 PLC 输出端被强制置零。因此，在 PLC 中要合理连接制动：即制动要在零信号保持闭合，也就是说 PLC 的 1 信号会打开制动。

说明
此报警的响应类似急停报警（3000）。出于内部原因，NCK 可能会略微延长重启延时。

激活

重启延时如下激活：
MD10088 \$MN_REBOOT_DELAY_TIME（重启延时）> 0
输入的数值即重启延时，单位为秒。

通过系统变量分析

系统变量 \$AN_REBOOT_DELAY_TIME 可在一个同步动作中读取。大于零的值表示操作软件已触发重启请求，指出 NCK 离成功重启（断电再上电）还剩多少时间，单位为秒。用户可以在同步动作中看出即将要进行的重启，并做出相应的操作（例如在 Safety Integrated 应用中进行“安全停止”）。操作软件未发出重启请求的情况下，\$AN_REBOOT_DELAY_TIME 始终为 0.0。

4.8 数据表

4.8.1 机床数据

4.8.1.1 通用机床数据

编号	名称: \$MN_	描述
10050	SYSCLOCK_CYCLE_TIME	系统基本周期
10070	IPO_SYSCLOCK_TIME_RATIO	插补周期的系数
10082	CTRLOUT_LEAD_TIME	设定值接收时间点的推移
10083	CTRLOUT_LEAD_TIME_MAX	可设置的最大设定值接收时间点推移

编号	名称: \$MN_	描述
10088	REBOOT_DELAY_TIME	重启延时
18342	MM_CEC_MAX_POINTS[t]	垂度补偿的控制点数量上限

4.8.1.2 通道专用机床数据

编号	名称: \$MC_	描述
20150	GCODE_RESET_VALUES	G 功能组的初始设置

4.8.1.3 进给轴/主轴专用机床数据

编号	名称: \$MA_	描述
32450	BACKLASH	反向间隙
32452	BACKLASH_FACTOR	反向间隙的权重系数
32456	BACKLASH_DYN	动态间隙补偿的补偿值
32457	BACKLASH_DYN_MAX_VELO	动态间隙补偿变化速率上限
32490	FRICT_COMP_MODE	摩擦补偿类型
32500	FRICT_COMP_ENABLE	摩擦补偿生效
32510	FRICT_COMP_ADAPT_ENABLE	摩擦补偿自适应生效
32520	FRICT_COMP_CONST_MAX	最大摩擦补偿值
32530	FRICT_COMP_CONST_MIN	最小摩擦补偿值
32540	FRICT_COMP_TIME	摩擦补偿时间常数
32550	FRICT_COMP_ACCEL1	自适应加速度值 1
32560	FRICT_COMP_ACCEL2	自适应加速度值 2
32570	FRICT_COMP_ACCEL3	自适应加速度值 3
32580	FRICT_COMP_INC_FACTOR	短暂运动中摩擦补偿值的权重系数
32610	VELO_FFW_WEIGHT	速度/转速前馈控制系数
32620	FFW_MODE	前馈控制类型
32630	FFW_ACTIVATION_MODE	从程序激活前馈控制
32650	AX_INERTIA	转矩前馈控制的转动惯量
32700	ENC_COMP_ENABLE	插补补偿
32710	CEC_ENABLE	垂度补偿使能

4.8 数据表

编号	名称: \$MA_	描述
32711	CEC_SCALING_SYSTEM_METRIC	垂度补偿的单位制
32720	CEC_MAX_SUM	垂度补偿的最大补偿值
32730	CEC_MAX_VELO	垂度补偿导致的最大速度变化率
32750	TEMP_COMP_TYPE	温度补偿类型
32760	COMP_ADD_VELO_FACTOR	补偿导致的过速
32711	CEC_SCALING_SYSTEM_METRIC	垂度补偿的单位制
32800	EQUIV_CURRCTRL_TIME	电流环前馈控制的等效时间常数
32810	EQUIV_SPEEDCTRL_TIME	转速环前馈控制的等效时间常数
32910	DYN_MATCH_TIME	动态响应自适应的时间常数
36500	ENC_CHANGE_TOL	位置实际值切换时的最大公差
38000	MM_ENC_COMP_MAX_POINTS	插补补偿中的控制点数量

4.8.2 设定数据

4.8.2.1 通用设定数据

编号	名称: \$SN_	描述
41300	CEC_TABLE_ENABLE[t]	使能垂度补偿表
41310	CEC_TABLE_WEIGHT[t]	垂度补偿表的权重系数

4.8.2.2 进给轴/主轴专用设定数据

编号	名称: \$SA_	描述
43900	TEMP_COMP_ABS_VALUE	与位置无关的温度补偿数值
43910	TEMP_COMP_SLOPE	位置相关的温度补偿的斜率
43920	TEMP_COMP_REF_POSITION	温度相关温度补偿的参考位置

4.8.3 信号

4.8.3.1 从 NC 发出的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
NC 就绪	DB10.DBX108.7	DB2700.DBX2.7

4.8.3.2 从 BAG 发出的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
BAG 就绪	DB11.DBX6.3	DB3100.DBX0.3

4.8.3.3 来自通道的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
通道就绪	DB21, ...DBX36.5	DB3300.DBX4.5

4.8.3.4 到进给轴/主轴的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
激活动态间隙补偿	DB31, ...DBX25.0	DB380x.DBX5001.0

4.8.3.5 来自进给轴/主轴的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
已回参考点/已同步 1	DB31, ...DBX60.4	DB390x.DBX0.4
已回参考点/已同步 2	DB31, ...DBX60.5	DB390x.DBX0.5
轴运行就绪	DB31, ...DBX61.2	DB390x.DBX1.2
动态间隙补偿生效	DB31, ...DBX102.0	DB390x.DBX5006.0

K5：运行方式组、通道、跨通道取轴

5.1 简要说明

运行方式组

运行方式组是机床轴、主轴和通道构成的一个整体。原则上，每个运行方式组都相当于一台独立的、有多个通道的数控系统。它包含的通道是在运行顺序上始终需要同时在一个运行方式中工作的多个通道。

说明
一般情况下系统配备一个运行方式组（英文简称 BAG）。
文档：
功能手册之基本功能分册；章节“BAG、通道、程序运行”(K1)
说明
SINUMERIK 828D 仅提供 1 个 BAG。

通道

每个通道均独立执行程序解码、程序段处理和插补。在一个通道内可执行一个独立的零件程序。

说明
一般情况下系统配备一个通道。
文档：
功能手册之基本功能分册；章节“BAG、通道、程序运行”(K1)
一个运行方式组在多个通道内的运行可在零件程序中进行同步。
说明
SINUMERIK 828D 仅提供 1 个通道。

跨通道取进给轴/主轴

控制系统接通后，进给轴/主轴被指定给特定通道，并且只能在此通道中使用。

使用“跨通道取进给轴/主轴”功能可释放进给轴/主轴并将其指定给另一个通道，即跨通道获取进给轴/主轴。

跨通道取进给轴/主轴功能既可通过零件程序激活，也可通过 PLC 程序由运动同步动作激活。

跨通道取进给轴/主轴功能可通过以下方式激活：

- 在零件程序中编写 GET/GETD 指令。
- 在零件程序中编写轴名称来自动激活。
- 不触发预处理停止，并在必要时执行主处理和预处理的同步。
- 通过 PLC 发送 VDI 信号给 NCK 来激活。

跨通道取轴功能的新特性

- 跨通道取轴特性可设置。
- 可在轴容器旋转中生成隐性的 GET/GETD 指令触发跨通道取轴。
- 对不参与轮廓加工的轴不会引起预处理停止。
- 支持经过旋转的 WCS (ROT) 中的几何轴，支持 JOG 模式。
- 可通过同步动作 GET(轴)，AXTOCHAN 实现。

说明

SINUMERIK 828D 不支持跨通道取进给轴/主轴功能。

5.2 运行方式组 (BAG) - 仅用于 840D sl

运行方式组

运行方式组是 NC 通道、进给轴和主轴组合构成的加工单元。

一个运行方式组所包含的通道是在运行顺序上始终需要同时在一个运行方式中工作的多个通道。

在一个运行方式组内可在任意通道中编写任意轴。因此一个运行方式组也可视为一台单独的多通道数控系统。

示例

在大型机床（加工中心）上，可能会需要在机床的一部分执行一个零件程序，同时在另一部分夹紧和安装待加工的新工件。 此类任务通常需要两个独立的数控系统。

若借助运行方式组功能，一台包含两个运行方式组的数控系统即可实现这两个任务，因为每个运行方式组可设置不同的运行方式：**AUTO** 用于执行程序，**JOG** 用于安装工件。

运行方式组的指定

配置运行方式组时，需定义 **BAG** 由哪些通道、几何轴、机床轴和主轴构成。

一个运行方式组中可以有一个或多个通道，这些通道不能再由其他 **BAG** 使用。通道中又包含指定的机床轴、几何轴和辅助轴。一根机床轴只能指定给一个 **BAG** 中的通道，而且只能在指定通道内运行。

运行方式组通过以下数据进行配置：

- 通道专用机床数据：
MD10010 \$MN_ASSIGN_CHAN_TO_MODE_GROUP（**BAG** 中生效的通道）
- 通道的配置数据

说明

以第一个 **BAG** 为例的详细说明请见：

文档：

功能手册之基本功能分册；章节“**BAG、通道、程序运行**”(K1)

5.3 通道 - 仅用于 840D sl

说明

通道概念、通道配置、通道状态、指令/信号的作用等信息在以下文档中以第一通道为例说明：

文档：

功能手册之基本功能分册；章节“**BAG、通道、程序运行**”(K1)

针对第一通道的这些说明同样适用于所有其他通道。

5.3 通道 - 仅用于 840D sl

5.3.1 通道同步（程序协调）

功能

在双滑块加工中或处理实时任务时，必须要实现多个通道之间加工的同步。这样才能确保特定工序在多个通道中有序运行。

为了实现这一点，必须将相关通道整合到一个同步组（BAG）中。

通道同步只可通过 NC 语言指令实现。

前提条件

所涉及通道必须属于同一运行方式组（BAG）。

编程

通道同步需要使用特殊的指令（命令）。这些指令均需要占用一个单独的程序段。

表格 5-1 程序协调的指令

指令	含义	
INIT (<通道号>, <路径>, <应答模式>)	选择某个程序以便在特定通道中执行	
	<通道号>:	通道的编号
	<路径>:	NC 程序的绝对或相对路径
	<应答模式>:	应答模式: N（无）或 S（同步）
CLEAR (<程序名称>)	删除指定名称的程序	
START (<通道号>, <通道号>, ...)	在其他通道中启动所选的程序	
	<通道号>, ...:	升序排列的通道号
WAITM (<标记号>, <通道号>, <通道号>, ...)	无条件等待：一旦当前通道执行到 WAITM() 指令，便停止当前通道的所有轴，同时等待其他需要同步的通道，直至其执行到指定的等待标记号。如果这些通道执行到 WAITM() 指令时轴也停止，则达到通道同步的状态。同步后的通道会继续运行。	
	<标记号>:	所有通道中的标记号必须相同。
	<通道号>, ...:	升序排列的通道号（可以指定当前通道，当不强制指定）
WAITE (<通道号>, <通道号>, ...)	等待指定通道的程序结束（不指定当前通道）	

指令	含义
WAITMC (<标记号>, <通道号>, <通道号>, ...)	有条件等待指令：在连续轨迹模式中等待指定通道执行到指定的等待标记号。可以指定当前通道，当不强制指定。 在等到该标记号、轴再次加速后，指定通道的标记被删除。
SETM (<标记号>, <标记号>, ...)	为输出本指令的通道的有条件等待设置等待标记 WAITMC()。通道利用该标记可向其伙伴通道表明“满足等待结束条件”。 可在同步动作中激活此指令。一个指令最多可设置 10 个标记（0 - 9）。
CLEARM (<标记号>, <标记号>, ...)	为输出本指令的通道的有条件等待删除等待标记 WAITMC()。通道利用该标记可向其伙伴通道表明“不/不再满足等待结束条件”。 可在同步动作中激活此指令。一个指令最多可清除 10 个标记（0 - 9）。

说明

WAITMC 和 SETM 的更多详细信息请见“通道同步：连续路径运行中的有条件停止 (页 328)”章节。

说明

一个多通道系统中最多可设置 100 个标记（标记 0 ... 99）。

一个单通道系统中则只有标记 0。

示例：无条件等待 WAITM

通道 1：已选择程序 /_N_MPF_DIR/_N_MPF100_MPF。

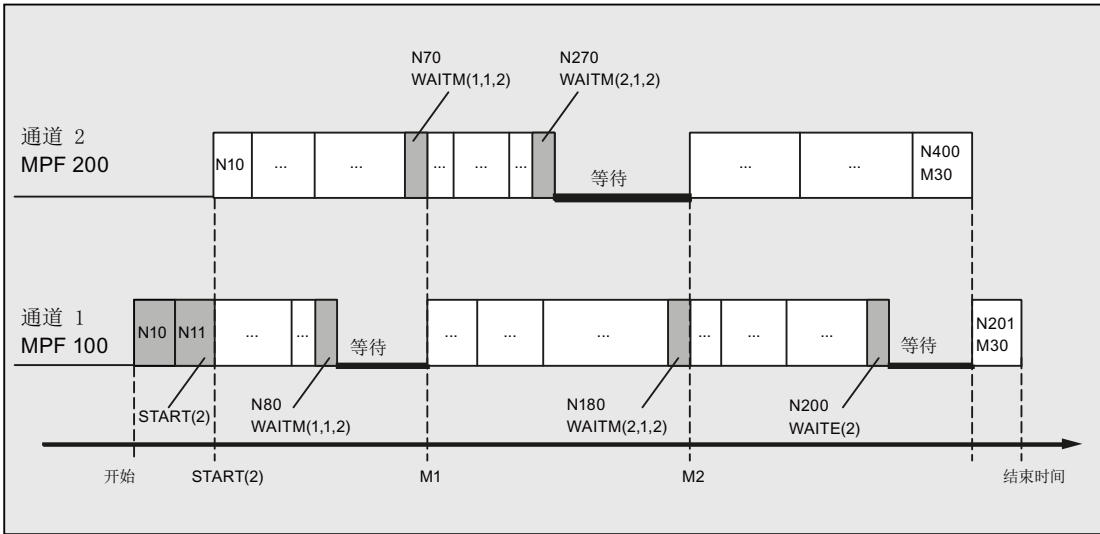
程序代码	注释
N10 INIT (2, "MPF200", "N")	
N11 START (2)	
...	; 在通道 1 中加工
N80 WAITM (1,1,2)	; 等待通道 1 和 2 执行到等待标记 1。
...	; 通道 1 中继续加工。
N180 WAITM (2,1,2)	; 等待通道 1 和 2 执行到等待标记 2。
...	; 通道 1 中继续加工。
N200 WAITE (2)	; 等候通道 2 程序结束。

5.3 通道 - 仅用于 840D sl

程序代码	注释
N201 M30	; 通道 1 程序结束，全部结束。

通道 2: INIT 指令（参见 _N_MPF100_MPF 中的 N10）在通道 2 中执行程序 _N_MPF200_MPF。

程序代码	注释
; \$PATH=/_N_MPF_DIR	
...	; 在通道 2 中加工
N70 WAITM(1,1,2)	; 等待通道 1 和 2 执行到等待标记 1。
...	; 通道 2 中继续加工。
N270 WAITM(2,1,2)	; 等待通道 1 和 2 执行到等待标记 2。
...	; 通道 2 中继续加工。
N400 M30	; 通道 2 程序结束。



5.3.2 通道同步：连续路径运行中的有条件停止

功能

在有条件等待 WAITMC 中，只要有一个需要同步的通道没有执行到指定的标记，轴就会减速并等待该标记。

生成等待标记的时间点与有条件等待调用的时间点无关联。

说明

即便无等待或制动需求（无 WAITMC 指令），也可设置标记用于通道间的交流。在此情形下，通道标记在复位和 NC 启动后仍将保留数值。

前提条件

为缩短 WAITMC 的等待时间，必须满足以下条件：

- 连续路径运行 G64 已设置。
- “LookAhead” 功能生效。

说明

若选择了准停（G60，G09），那么 WAITMC() 指令的作用等同于 WAITM()。

制动方式

从调用 WAITMC() 前的一条运动程序段起，NC 就开始检查其他通道是否都执行到等待标记。如果是，轴会不减速继续运行。

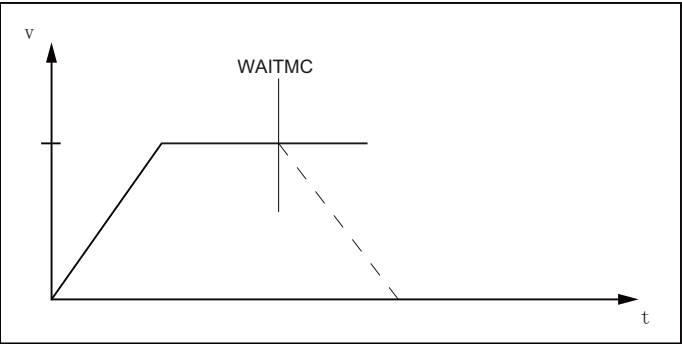


图 5-1 有条件等待 WAITC 下轨迹速度的特性曲线：所有通道均已执行到等待标记

如果还有通道没有执行到等待标记，轴会开始减速。在减速期间，NC 会按插补周期检查该通道是否执行到等待标记。如果是，NC 会使轴再次加速到轨迹速度继续工作。

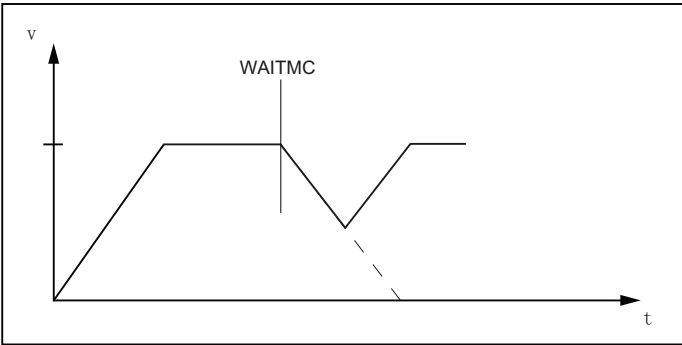


图 5-2 有条件等待 WAITC 下轨迹速度的特性曲线：最后的等待标记在制动期间出现

若待同步通道尚未执行到标记前轨迹速度已降至零，那么加工将停止，直至该通道执行到等待标记。在到达最后一个标记后，将从静止状态起重新加速至轨迹速度。

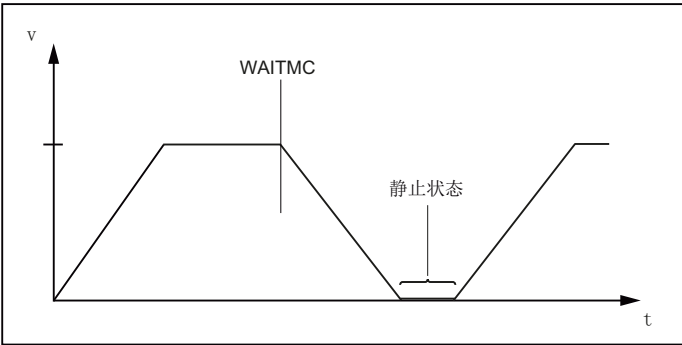


图 5-3 有条件等待 WAITC 下轨迹速度的特性曲线：最后的等待标记在制动后出现

制动斜坡上的程序段切换

运动结束条件 IPOBRKA（允许在制动斜坡上切换程序段）生效时，到达等待标记后将暂时切换至下一个程序段，并启动进给轴。

若尚未到达标记，或有另一个运动停止条件阻止程序段切换，则轴会继续减速。

示例：连续路径运行中的有条件停止

此示例只包含对于通道同步来说非常重要的指令，仅作参考。

通道 1:

程序代码	注释
%100	
N10 INIT (2, "_N_200_MPF", "n")	; 选择通道 2 同步程序。
N11 INIT (3, "_N_300_MPF", "n")	; 选择通道 3 同步程序。

程序代码	注释
N15 START (2,3)	; 启动通道 2 和通道 3 中的程序。
...	; 通道 1 中执行加工。
N20 WAITMC (7,2,3)	; 有条件等待通道 2 和通道 3 执行到标记 7。
...	; 通道 1 中继续加工。
N40 WAITMC (8,2)	; 有条件等待通道 2 执行到标记 8。
...	; 通道 1 中继续加工。
N70 M30	; 通道 1 结束。

通道 2:

程序代码	注释
%200	
N200	; 通道 2 中执行加工。
N210 SETM (7)	; 通道 2 设置等待标记 7。
...	; 通道 2 中继续加工。
N250 SETM (8)	; 通道 2 设置等待标记 8。
N260 M30	; 通道 2 结束。

通道 3:

程序代码	注释
%300	
N300 ...	; 通道 3 中执行加工。
...	
N350 WHEN <条件> DO SETM (7)	; 在同步动作中设置等待标记。
...	; 通道 3 中继续加工。
N360 M30	; 通道 3 结束。

5.3 通道 - 仅用于 840D sl

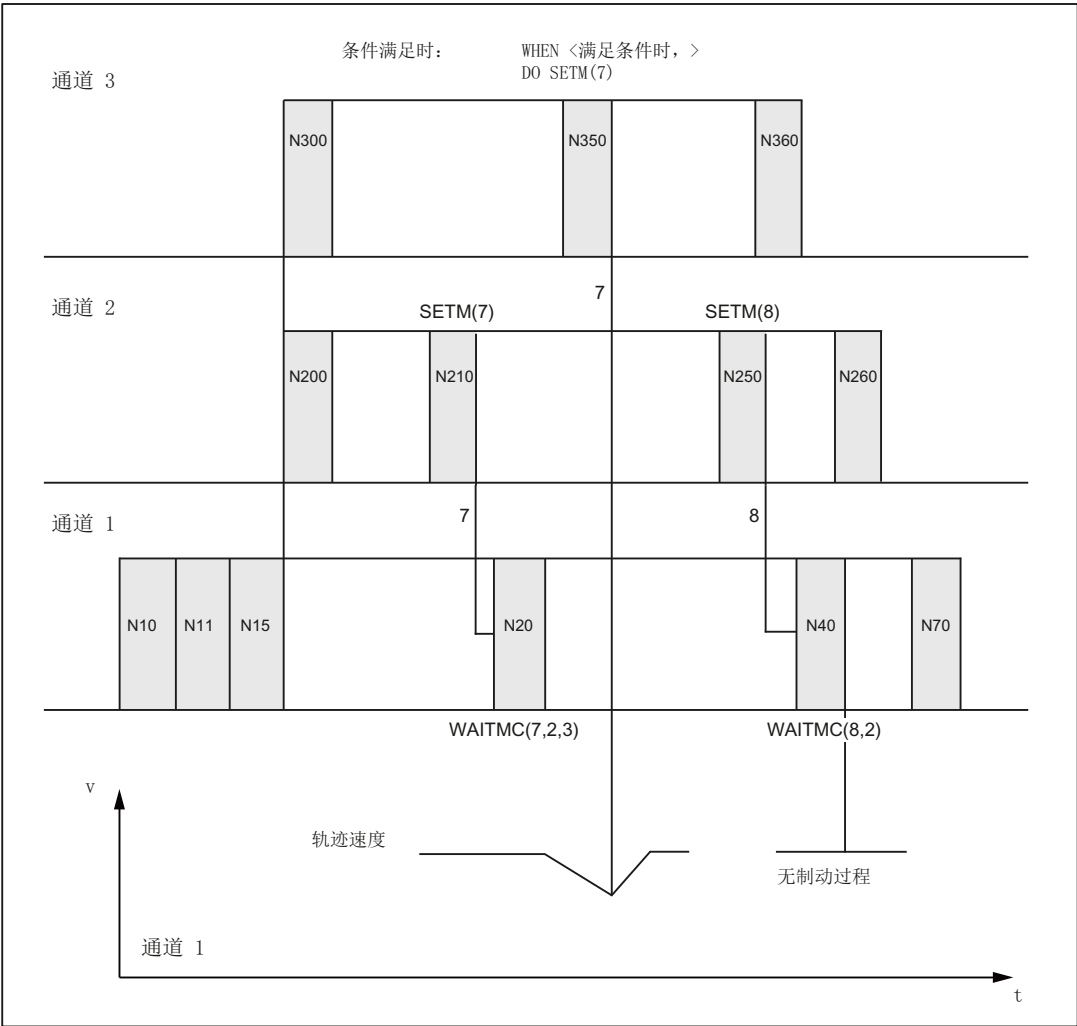


图 5-4 连续路径运行中有条件停止，涉及三个通道（示意图）

示例：WAITMC 和读取禁止

运行期间在通道 3 中输出 M555 并生成读取禁止（RID）。由于 WAITMC 被添加至程序段 N312，因此设置等待标记，通道 2 继续运行。通道 3 中的程序处理则由于读取禁止而停止。

说明

G64 生效时，WAITMC(...) 程序段不生成单独的程序段，而是添加到前一程序段中。连续路径运行生效时应不会出现速度跃变。这样一来，在前一程序段因读取禁止停止时，实际上已实现 WAITMC。

通道 2:

程序代码	注释
N112 G18 G64 X200 Z200 F567	; 通道 2 中执行加工。
N120 WAITMC(1,2,3)	; 有条件等待通道 2 和通道 3 执行到标记 1。
...	; 由于 WAITMC 被添加至程序段 N312, 通道 2 中继续加工。
...	; 通道 2 中继续加工。
N170 M30	; 通道 2 结束。

通道 3:

	注释
	; 运行期间读取禁止 M555。
N300 ...	; 通道 3 中执行加工。
N312 G18 G64 D1 X180 Z300 M555	
N320 WAITMC(1,2,3)	; RID 导致等待。

5.3.3 单通道式试车

功能

在多通道系统中，通过“单通道式试车”功能可在机床上对选择的一个通道的零件程序进行实测。其他通道此时则处于“程序测试”状态。也就是说，启动所有通道时仅所选通道的轴会实际运行。

此外，对于“程序测试”状态下的进给轴和主轴，用户也可抑制这些轴的“程序测试”状态，使其与所选择的通道一起运行。

优点

在多通道系统中创建零件程序对编程人员的抽象思维能力有较高要求。而“单通道式试车”功能可降低此类零件程序的测试风险，并可选择特定通道。

应用

- “单通道式试车”功能适用于：
- 多通道系统
 - 采用 POSA 或指令轴运动的机床

过程

多通道系统可同时启动，也可逐个通道地分时启动。或者可由 PLC 启动一个通道，通过该通道的零件程序初始化和启动其他通道。这两种方案“单通道式试车”均可支持。由此产生一个共同运作的通道组。通常情况下，该组由 NCK 使用的所有通道组成。

一个通道通常需要在工作区域内运行一个刀架，即一把刀具。每个通道在同一工作区域内都需要运行一把刀具，因此必须确保通道间的同步，以避免发生碰撞，并实现通道间的有序协作。可能需要的同步动作如下：

- 通过零件程序指令 WAITM、WAITMC、WAITE、START 实现的通道同步。
- 通过 PLC 实现的通道同步。

示例：

通道 1 中通过读取禁止停止在 M107，直至通道 2 到达 M207；相反的情形下也是如此。

- 跨通道取轴，即一个通道等待直至另一个通道交出轴。
- 零件程序中通过全局变量自由编写的同步动作。
- 跨通道同步
- 轴容器旋转
- 程序测试，包括主处理中的并行同步动作，以及同步动作与通道的同步。

在这些边界条件下，几乎不可能只启动一个通道，否则通道会停留在第一个同步位置。

通过“单通道式试车”功能可启动通道组中的所有通道，其中仅少量通道（通常为一个通道）会发生实际的轴运行。其他通道则处于“程序测试”状态。

因此用户须定义不期望发生运动的通道。这通过操作界面中的“程序控制”菜单实现。选择此功能时，HMI/PLC 接口中会置位以下通道专用信号：

DB21, ... DBX25.7（程序测试已选择）

之后通过以下 NC/PLC 接口信号激活：

DB21, ... DBX1.7（激活程序测试）

PLC 中会置位以下接口信号作为反馈信息：

DB21, ... DBX33.7（程序测试生效）

此外，为了确保试车成功，可能需要实际运行某些进给轴/主轴（特别是主轴），尽管这些轴处于“程序测试”状态下。为此可使用以下 NC/PLC 接口信号：

DB31, ... DBX14.0（抑制程序测试）

示例：

设备包含主主轴和副主轴。两个滑块在主主轴和副主轴上均可运行。每个滑块由一个通道控制。主主轴位于通道 1 中，副主轴（GS）位于通道 2 中。对通道 1 进行试车，并通过通道专用 NC/PLC 接口信号 DB21, ... DBX1.7（激活程序测试）停止通道 2。某种程度上而言，两个工件主轴（主主轴和副主轴）均担当“特殊角色”。为了在工件上开展实际加工，但又无须使工件主轴所处的通道实际运行，便需要单独使两个工件主轴（必要时包含工件上的轴）实际运行。

说明

“程序测试”状态只能在通道停止状态下激活/取消，而轴专用 NC/PLC 接口信号“抑制程序测试”则可随时切换。

诊断

“程序测试”状态可通过系统变量查询：

- 在操作界面、同步动作或带有预处理停止的零件程序中显示，通过系统变量：

\$AC_ISTEST	通道的“程序测试”状态 通道处于“程序测试”状态下时，输出 TRUE (1)。
\$AA_ISTEST[<n>]	轴 <n> 的“程序测试”状态 轴 <n> 处于“程序测试”状态下时，输出 TRUE (1)。

- 在不带预处理停止的零件程序中，通过系统变量：

\$P_ISTEST	通道处于“程序测试”状态下时，输出 TRUE (1)。

示例：

通道处于“程序测试”状态下，且轴“C”已通过“抑制程序测试”脱离该状态。通过系统变量查询时会得到以下结果：

\$AC_ISTEST = TRUE
\$P_ISTEST = TRUE
\$AA_ISTEST[C] = FALSE

边界条件

跨通道取轴

“跨通道取轴”功能能够将进给轴/主轴的信息提供给多个通道，并可由这些通道交替运行（参见“跨通道取进给轴/主轴 (页 338)”章节）。

在与“程序测试”和“单通道式试车”功能一起使用时，跨通道取轴时须注意：

- 如果只有一个通道处于“程序测试”状态，则从该通道取出轴，然后放入到不处于该状态的通道中。如果需要取出的轴处于“禁用状态”，在放入另一个通道后，不管该通道是否处于“程序测试”状态，该轴本身的状态不变（参见示例 3）。
- 程序测试中，零件程序结束/复位时所有未插补的进给轴/主轴将同步至当前的伺服位置。在一些跨通道取轴功能中，如果轴只有在程序结束时才能从原通道中释放，那接收通道只能在程序结束后才能获得该轴。这会导致经过仿真后到达的位置不会传送给接收通道。

说明

程序末尾处应设置一个 WAIT 标记，以确保同时结束。

示例

示例 1：在一台 3 通道设备中对通道 2 进行试车。

可采用以下程序测试步骤：

a. 无 SERUPRO 的程序测试

1. 用户考虑需要实际运行哪些进给轴/主轴。为这些轴置位“抑制程序测试”。
2. 为通道 1 和通道 3 选择“程序测试”状态。
3. 通过 PLC 启动通道 1、通道 2、通道 3。
4. 程序结束后可重新取消“程序测试”。
5. 若当前的“抑制程序测试”设置也适用于其他情形（需要对通道 1 或通道 3 进行试车），可保持此信号的置位状态。很多场合都会用到此设置。

b. 带 SERUPRO 的程序测试

1. 用户考虑需要实际运行哪些进给轴/主轴。为这些轴置位“抑制程序测试”。
2. 为通道 1 和通道 3 选择“程序测试”状态。
3. 通过 PLC 启动通道 1、通道 2、通道 3。
4. 出现故障或报警时，按下 RESET 终止。
5. 对 3 个通道的断点执行 SERUPRO。

- 6. 3 个通道中均到达搜索目标。
- 7. 启动全部 3 个通道。
- 8. 通道 1 和通道 3 重新进入“程序测试”状态，“单通道式试车”继续进行。

示例 2：启用“抑制程序测试”

一个通道处于“程序测试”状态下。需要在运行中为“Y”轴触发“抑制程序测试”（程序段 N1010 处）。

程序代码	注释
N1000 G0 Y1000	
N1010 G4 F10	
N1020 G0 G91 Y=10	; 断续运行。
N1030 M30	

程序运行至位置 1010，即在“抑制程序测试”启用后运行该轴的仿真分量“1000”。

示例 3：程序测试和跨通道取轴

在下面的示例中，通道 1 中的轴“X”和通道 2 中的“X1”是 NCK 的第 1 轴。开始时所有轴处于位置 0。

通道 1 进行“程序测试”	通道 2 不进行“程序测试”
N10010 G0 X0 Y0	
N10020 X100	
N10030 WAITM(91,1,2)	
N10040 WAITM(92,1,2)	
N10050 M0	
N10060 M30	
	N20010 WAITM(91,1,2)
	N20020 G91 G0 X1=10
	N20030 WAITM(92,1,2)
	N20040 M0
	N20050 M30

程序段 N20040（M0）处，第 1 轴（X 或 X1）位于 110 位置上。也就是说，通道 1 中仿真到达的“100”位置在通道 2 中由程序段 N20020 接收。

文档

程序测试的相关信息请见：

功能手册之基本功能分册；章节“BAG、通道、程序运行、复位特性 (K1)”

5.4 跨通道取进给轴/主轴

5.4.1 引言

简介

进给轴/主轴由机床数据固定指定给一个特定通道，且只能在此通道中使用。

定义

使用“跨通道取进给轴/主轴”功能可释放进给轴/主轴并将其指定给另一个通道，即跨通道获取进给轴/主轴。

主轴功能隶属于轴功能，因此下文将统一使用“跨通道取轴”这一术语。

轴类型

按通道划分时轴类型分为四种。跨通道取轴时的响应取决于以下机床数据：

MD30552 \$MA_AUTO_GET_TYPE

通道轴

通道轴可在零件程序中编写，并可在所有运行模式下运行。

PLC 轴

PLC 轴只可通过 PLC 定位。

若在零件程序中写入了一根 PLC 轴，那么：

- 若 MD AUTO_GET_TYPE = 0，则会触发报警。
- 若 MD AUTO_GET_TYPE = 1，则会生成自动 GET。
- 若 MD AUTO_GET_TYPE = 2，则会生成自动 GETD。

中立轴

若在零件程序中写入了一根中立轴，那么：

- 若 MD AUTO_GET_TYPE = 0，则会触发报警。
- 若 MD AUTO_GET_TYPE = 1，则会生成自动 GET。
- 若 MD AUTO_GET_TYPE = 2，则会生成自动 GETD。

另一个通道中的轴

这其实并非真正的轴类型，而是可取轴在另一个通道中生效时的内部状态（作为通道轴、PLC 轴或中立轴）。

若在零件程序中写入了另一个通道中的轴，那么：

- 若 MD AUTO_GET_TYPE = 0，则会触发报警。
- 若 MD AUTO_GET_TYPE = 1，则会生成自动 GET。
- 若 MD AUTO_GET_TYPE = 2，则会生成自动 GETD。

说明

以下两个机床数据：

MD20110 \$MC_RESET_MODE_MASK

MD20112 \$MC_START_MODE_MASK

用于控制复位、启动和零件程序启动时的轴设置。要为跨通道取轴涉及的通道设置合理值，避免结合以下机床数据使用时产生不兼容配置（报警）：

MD30552 \$MA_AUTO_GET_TYPE

文档：

功能手册之基本功能分册；章节“轴、坐标系、框架（K2）”：工件相关的实际值系统

前提条件

为了能执行跨通道取轴，必须通过以下**通道专用**机床数据定义机床轴

MD20070 \$MC_AXCONF_MACHAX_USED（通道中生效的机床轴编号）

并通过以下**轴专用**机床数据定义用于跨通道取轴的通道缺省设置

MD30550 \$MA_AXCONF_ASSIGN_MASTER_CHAN（跨通道取轴的通道缺省设置）。

这两个设置可回答以下问题：

1. 在哪个通道使用轴和取轴？
2. 上电时轴指定给哪个通道？

示例：跨通道取轴

现有 6 根轴 和 2 个通道，通道 1 有轴 1、轴 2、轴 3、轴 4，通道 2 有轴 5 和轴 6。现在需要释放轴 1，并在上电后指定给通道 2。

通道专用机床数据必须如下赋值：

5.4 跨通道取进给轴/主轴

CHANDATA(1)

MD20070 \$MC_AXCONF_MACHAX_USED=(1, 2, 3, 4, 0, 0, 0, 0)

CHANDATA(2)

MD20070 \$MC_AXCONF_MACHAX_USED=(5, 6, 1, 0, 0, 0, 0, 0)

轴专用机床数据必须如下赋值：

MD30550 \$MA_AXCONF_ASSIGN_MASTER_CHAN[AX1]=2

显示

当前轴类型以及该轴当前归属的通道在一个轴 PLC 接口字节显示（参见“通过 PLC 取轴”小节）。

说明

若轴在所选择的通道中无效，则会在操作界面上通过轴名称反转加以显示。

5.4.2 跨通道取轴示例

假设

现有 6 根轴 和 2 个通道，通道 1 有轴 1、轴 2、轴 3、轴 4，通道 2 有轴 5 和轴 6。轴 2 要跨通道转交，在上电后转交给通道 1。

任务

此任务分为以下部分：

- 为机床数据赋值，确保满足跨通道取轴的前提条件。
- 编写通道 1 和通道 2 间的取轴。

前提条件的实现

通道专用机床数据 MD20070 的赋值

\$MC_AXCONF_MACHAX_USED[1]=(1, 2, 3, 4, 0, 0, 0, 0)

\$MC_AXCONF_MACHAX_USED[2]=(5, 6, 2, 0, 0, 0, 0, 0)

轴专用机床数据的赋值：

MD30550 \$MA_AXCONF_ASSIGN_MASTER_CHAN[AX2]=1

通道 1 中的程序	通道 2 中的程序 TAUSH2
... RELEASE (AX2) ; 轴 AX2 使能 INIT (2, " _N_MPF_DIR\N_TAUSH2_MPF" , " S") ; 选择通道 2 中的程序 TAUSH2 START (2) ; 启动通道 2 中的程序 WAITM (1,1,2) ; 等待通道 1 和 2 执行到等待标记 1 ... ; 取轴后的后续程序 ... ; 取轴后的后续程序 ... ; 取轴后的后续程序 M30	... WAITM (1,1,2) ; 等待通道 1 和 2 执行到等待标记 1 GET (AX2) ... ; 接收 AX2 ... ; 取轴后的后续程序 ... ; 取轴后的后续程序 RELEASE (AX2) ... ; 使能后续取轴 ... ; 使能后续取轴 M30

5.4.3 跨通道取轴方式

可通过零件程序或同步动激活跨通道取轴功能（一根轴或多根轴）。也可由 PLC 通过 VDI 接口请求和使能跨通道取轴。轴必须在本通道中用 RELEASE 指令释放，然才能通过 GET 指令传送到另一个通道中。

满足规定的前提条件时，跨通道取轴通过以下方式激活：

- 在零件程序中编写 GET/GETD 指令
通过 GET 从另一个通道接收进给轴或主轴，或者通过 GETD 直接从另一个通道取轴。
使用 GETD 指令时不需要配合使用 RELEASE 指令。
- 满足 MD30552 \$MA_AUTO_GET_TYPE > 0 的前提条件时，通过编写轴名称自动激活。
- 不触发预处理停止，并在必要时执行主处理和预处理的同步。
- 通过 PLC 发送 VDI 信号给 NCK 来激活。

在接收由 PLC 控制的轴时，可通过一个接口信号解除与 NC 程序触发的通道特性的关联。这样一来例如可以脱离 NC 程序对单个 PLC 轴进行插补（另见“自控单轴运动 (页 649)”章节）。

跨通道取轴功能的新特性

- 跨通道取轴特性可通过机床数据 MD10722 \$MN_AXCHANGE_MASK 设置。
- 可在轴容器旋转中生成隐性的 GET/GETD 指令触发跨通道取轴。

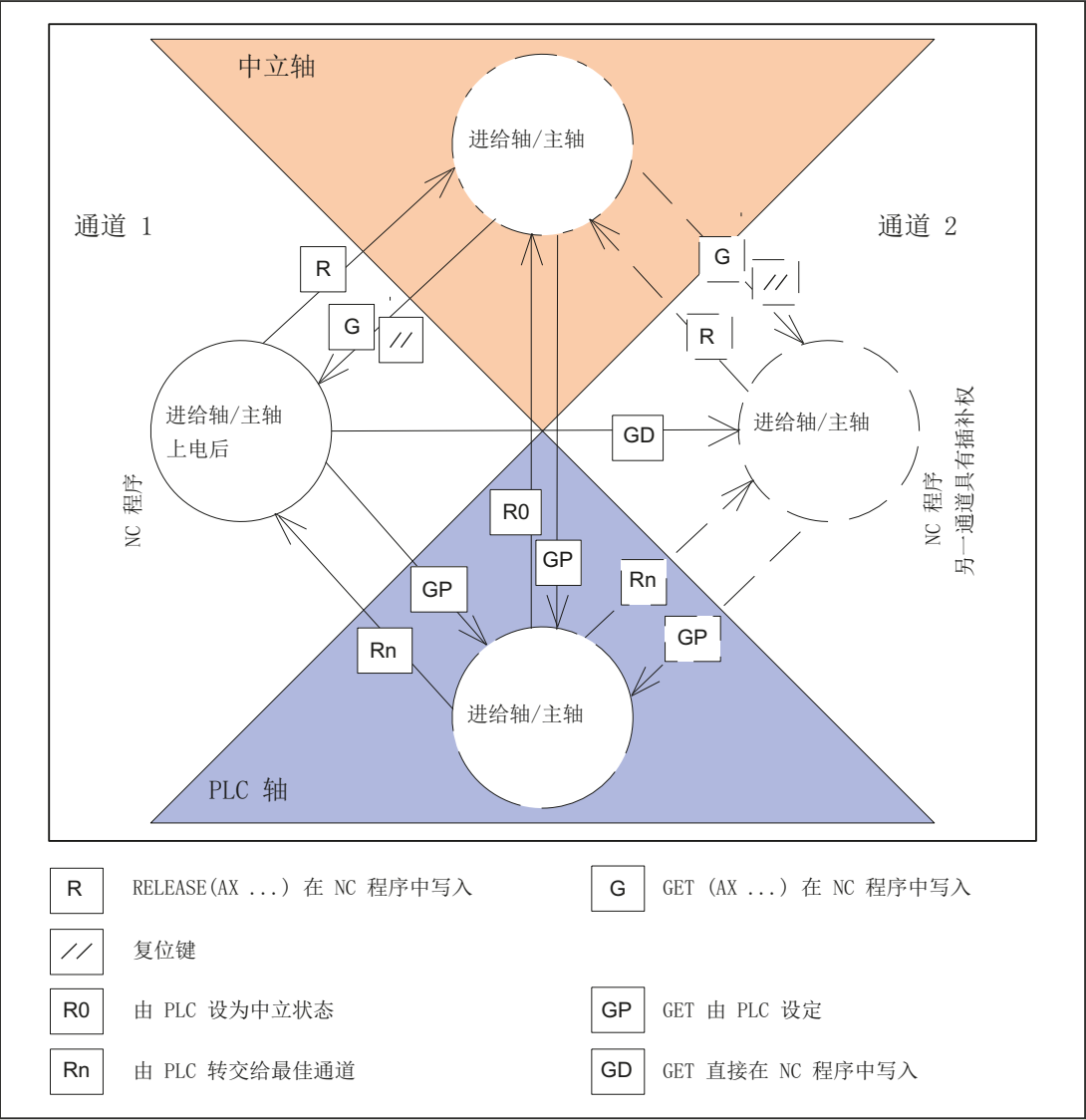
5.4 跨通道取进给轴/主轴

- 对不参与轮廓加工的轴不会引起预处理停止。
- 可通过机床数据 MD32074 \$MA_FRAME_OR_CORRPOS_NOTALLOWED 激活经过旋转的 WCS 中的几何轴以及 JOG 运行模式下的取轴。
- 可通过同步动作 GET(轴)、RELEASE(轴)、AXTOCHAN、\$AA_AXCHANGE_TYP(轴) 跨通道取轴。

5.4.4 NC 程序跨通道取轴特性

可能的过渡状态

下图显示了可通过哪些方式进行跨通道取轴。



5.4 跨通道取进给轴/主轴

5.4.5 将轴转换至中立轴（RELEASE）

功能

通过预定义程序 RELEASE () 使进给轴/主轴进入“中立轴”状态。

句法

```
RELEASE (<MA>, SPI (<SPNR>) [, ...])
```

含义

RELEASE:	使进给轴/主轴进入“中立轴”状态 边界条件：必须位于单独的程序段中。
MA:	机床轴名称
SPI (...):	用于将主轴号转换为相应轴名称的功能
SPNR:	主轴号

其他信息

通用终止条件

- 不满足跨通道取轴的所有前提条件。参见机床数据
 - MD20070 \$MC_AXCONF_MACHAX_USED
 - MD30550 \$MA_AXCONF_ASSIGN_MASTER_CHAN
- 轴参与了转换。
- 轴位于轴组中。

使用 Reset 键终止

如果轴已通过 RELEASE () 使能，但尚未通过 GET () 指令接收，此时执行复位，轴会被重新指定给原先的通道。

龙门轴组中的主动轴

使能龙门轴组中的主动轴时也会同时使能所有的从动轴。

5.4.6 在零件程序中接收进给轴或主轴（GET，GETD）

方式

跨通道取轴的释放时间点及特性在零件程序中的控制方式如下：

- 在同一通道中编写 GET 指令。
- 通过编写 GETD 指令直接从另一通道取轴。

文档：

编程手册“工作准备”；灵活 NC 编程，

章节：跨通道取进给轴/主轴（RELEASE，GET，GETD）

使用 GET 指令

GET (轴名称, 轴名称, SPI (主轴编号), ...)

以下情形下，轴的接收会延迟

- 该轴正在切换测量系统。
- 正在为该轴处理伺服使能（从控制切换至跟踪/停止，或相反）。
- 进给轴/主轴禁用生效。
- 另一个通道中尚未通过 RELEASE 释放该轴。
- 该轴的插补尚未完成（转速控制主轴除外）。

GET (轴名称, ...) 指令总是会生成一个独立的程序段，并会触发预处理停止。

特例：

- 该轴已经是通道轴，则不会生成程序段。
- 轴尚处于同步状态时（即此时尚未换入另一个通道，或者从 PLC 接收信号）同样不会生成程序段。

使用 GETD 指令

使用 **GETD** (GET Directly) 指令可从另一个通道直接取轴。也就是说，在另一个通道中不需要编写 RELEASE 来配合此 GETD 指令。不过此时必须建立另一种通道通讯（例如

5.4 跨通道取进给轴/主轴

等待标记），因为 GETD 会使提供轴的通道中断。若该轴为 PLC 轴，那么取轴会一直延迟，直至 PLC 释放该轴。

 小心

无功能轴
GETD 指令会使待取轴所处通道中的程序运行中断！（REORG）
特例情形：轴当前为中性轴。

说明

若在编写了 GET 或 GETD 指令的情形下取轴被延迟，之后通道中又进行了 RESET，那么通道将不再尝试接收此轴。

通过 GET 完成取轴后，按键复位或程序复位后此轴仍被指定给该通道。可重新编写 RELEASE 和 GET 来取轴，或在上电时将其指定给以下机床数据中定义的通道：

MD30550 \$MA_AXCONF_ASSIGN_MASTER_CHAN

5.4.7 自动取轴

通过编写轴名称自动取轴

重新编写了中立轴或将轴指定给另一个通道的情形下，系统会根据机床数据 MD30552 \$MA_AUTO_GET_TYPE 自动生成一个 GET 或 GETD 指令。

自动取轴的前提条件

MD30552 \$MA_AUTO_GET_TYPE > 0 是自动取轴的前提条件。

自动 GETD

说明

若设置了自动 GETD，则须注意以下事项。

- 通道可能会相互影响。（移除轴时 REORG。）
- 多个通道取一根轴时，无法预知最后到底哪个通道取到了轴。

示例 1

程序代码	注释
N1 M3 S1000	
N2 RELEASE (SPI(1))	; => 切换至中立轴
N3 S3000	; 待取主轴的新转速
	; MD AUTO_GET_TYPE =
	; 0 =>输出报警“轴类型错误”
	; 1 => 生成 GET (SPI(1))。
	; 2 => 生成 GETD (SPI(1))。

示例 2

程序代码	注释
	; (轴 1 = X)
N1 RELEASE (AX1	; => 切换至中立轴
N2 G04 F2	
N3 G0 X100 Y100	; 待取轴的运行
	; MD AUTO_GET_TYPE =
	; 0 =>输出报警“轴类型错误”
	; 1 => 生成 GET (AX1)。
	; 2 => 生成 GETD (AX1)。

示例 3

程序代码	注释
	; (轴 1 = X)
N1 RELEASE (AX1)	; => 切换至中立轴
N2 G04 F2	
N3 POS (X) = 100	; 待取轴的定位
	; MD AUTO_GET_TYPE =
	; 0 =>输出报警“轴类型错误”
	; 1 => 生成 GET (AX1)。 *)
	; 2 => 生成 GETD (AX1)。 *)

*) 轴尚处于同步状态时不会生成独立的程序段。

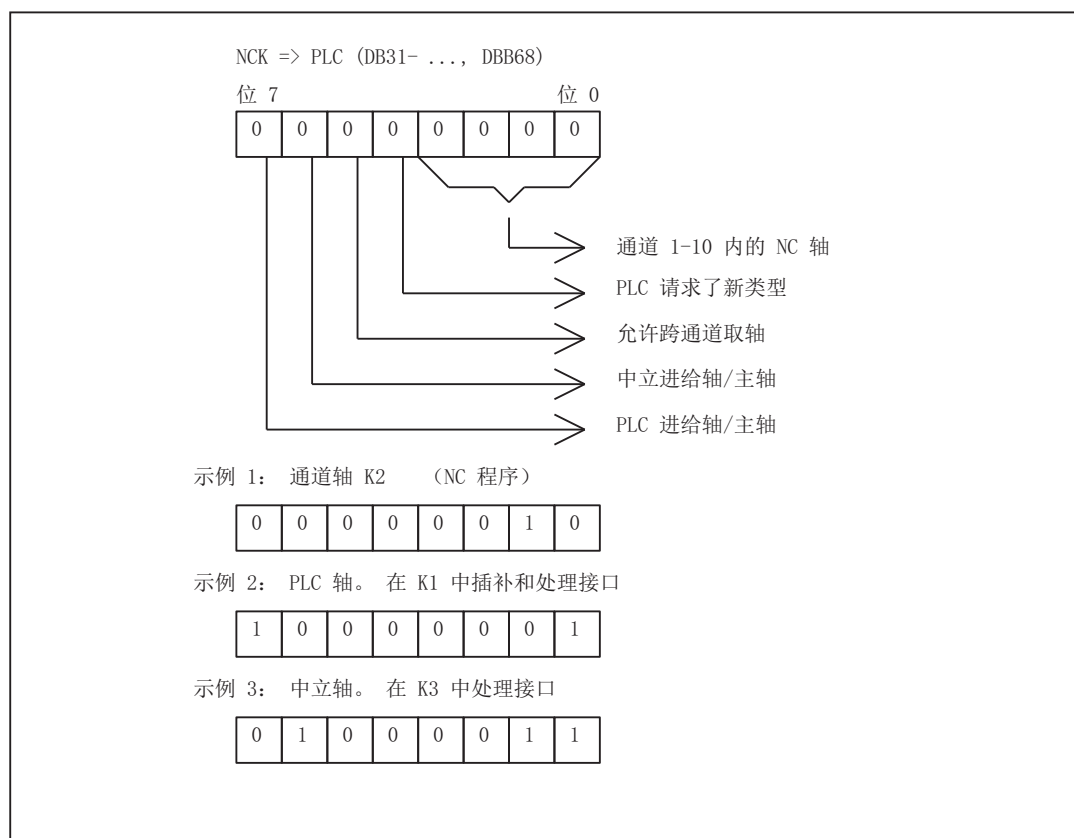
5.4.8 通过 PLC 取轴

PLC 可在任意运行模式下随时请求轴并运行。

PLC 可将一根轴从一个通道切换至另一个通道（仅适用于 840D sl）。

类型显示

轴类型（PLC 轴、通道轴、中立轴）可随时通过一个接口字节确定。



* 中立轴/主轴也包括指令轴/摆动轴

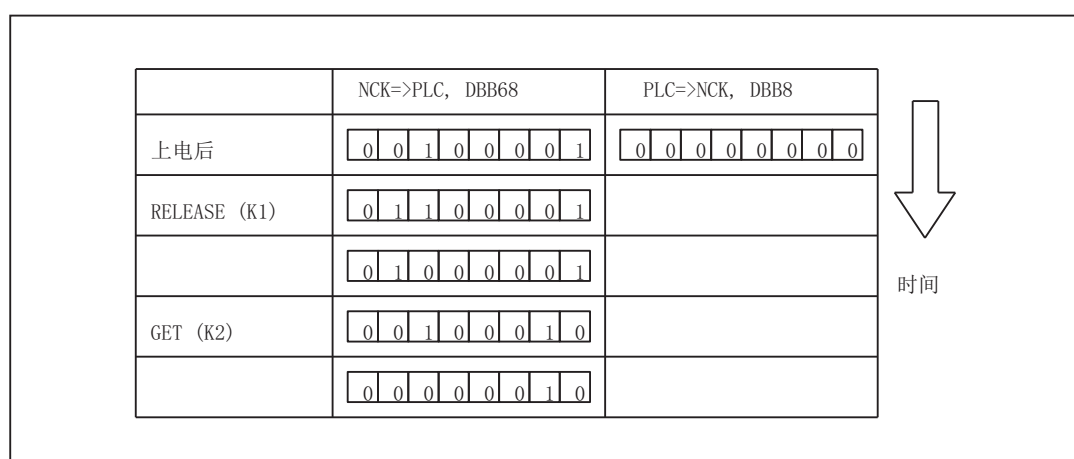
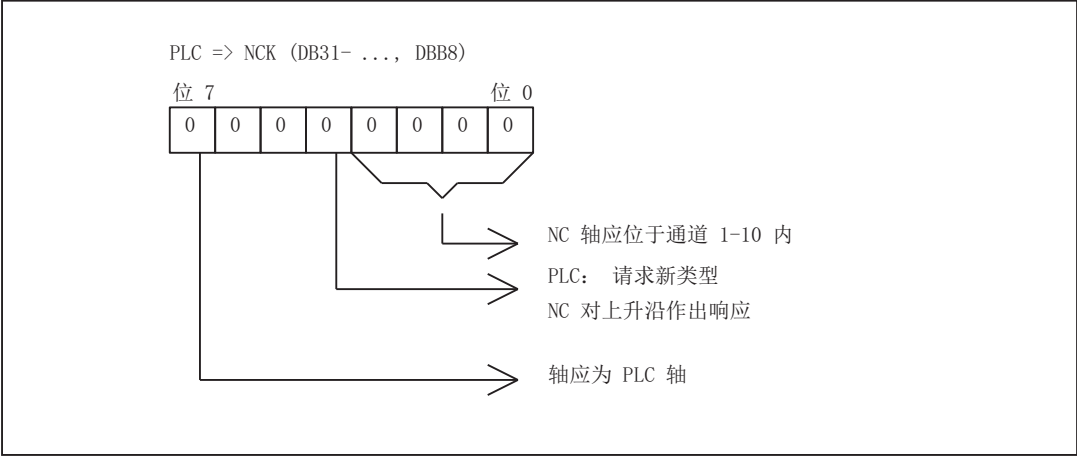


图 5-6 通过零件程序从 K1 取轴到 K2

类型设定



原则上须由 PLC 设置“请求新类型”信号。取轴后此信号再次被删除。使用 GET 和 RELEASE 取轴时也是如此（仅适用于 840D sl）。

840D sl 中 PLC 轴/主轴的控制

PLC 轴和 PLC 主轴通过 PLC 基本程序中的功能块 FC18 运行

FC18: SpinCtrl 主轴控制

示例

下图显示了将 NC 轴切换为 PLC 轴，以及通过 PLC 将 NC 轴转换为中立轴时 NC/PLC 接口信号的顺序。

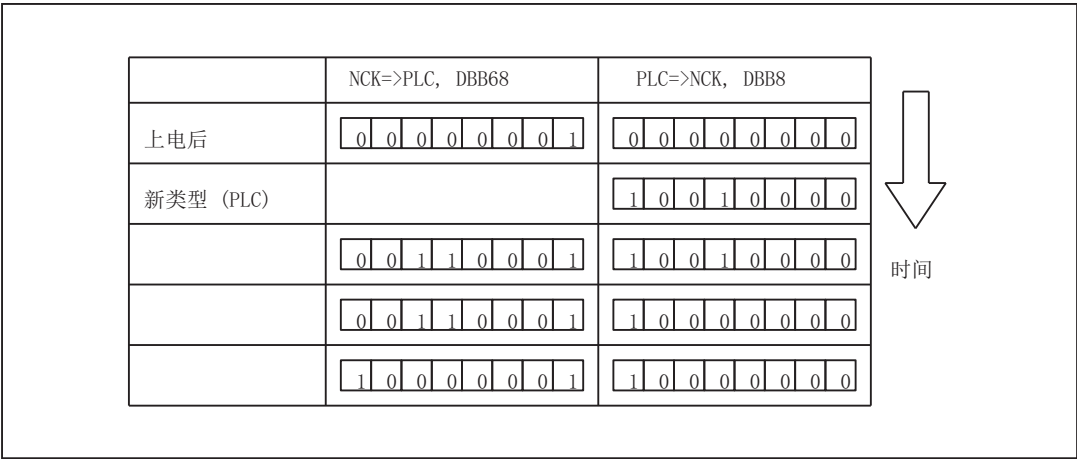


图 5-7 从 NC 轴切换为 PLC 轴

5.4 跨通道取进给轴/主轴

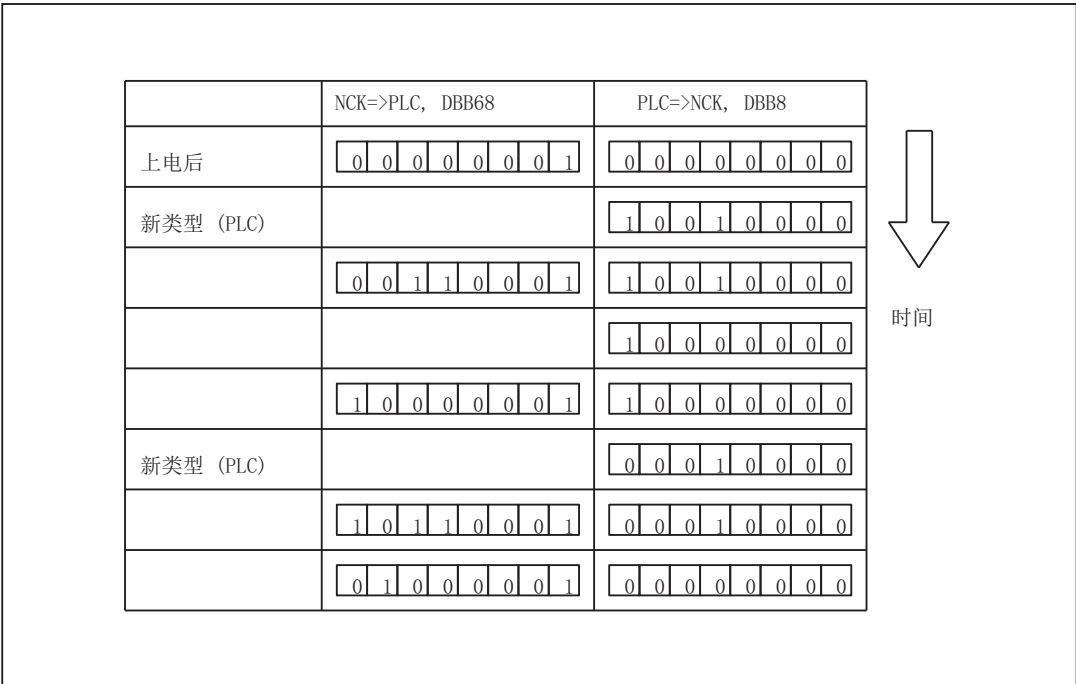


图 5-8 从 NC 轴通过 PLC 切换为中立轴

5.4.9 可修改的跨通道取轴特性

取轴在当前通道中进行，且可依据相应的轴类型通过机床数据 MD10722 \$MN_AXCHANGE_MASK 调整取轴特性。

表格 5-2 跨通道取轴时的释放时间点

MD10722	跨通道取轴特性
位 0 = 1	轴通过 WAITP 进入中立状态后，仍可执行两个通道间的自动取轴。
位 1 = 1	允许由轴容器旋转功能生成的隐性 GET/GETD 来取轴 一旦使能轴容器旋转，便会自动生成隐性的 GET 或 GETD，使可运行通道获取所有轴容器中的轴。只有在旋转结束后，这些轴才可以继续传送给其他通道。
位 2 = 1	跨通道取轴不触发预处理停止，必要时对不参与轮廓加工的轴进行强制重组。 在主处理中插入一个临时程序段之后，系统会检查是否需要重组。只有当该程序段的轴状态与当前的轴状态不一致时，才需要进行重组。

MD10722	跨通道取轴特性
位 3 = 0	由 PLC 请求获取 PLC 固指定轴 每根轴均可由 PLC 请求获取。固定指定的 PLC 轴只能从中立轴到 PLC 轴或相反。
位 3 = 1	通过 VDI 接口请求取轴 通过 VDI 接口发出的取轴请求只适用于以下轴： MD30460 \$MA_BASE_FUNCTION_MASK 位 4=1 条件下 只由 PLC 控制的轴 MD30460 \$MA_BASE_FUNCTION_MASK 位 5=1 条件下 固定指定的并行定位轴（PLC 轴） 对于此类轴，接口信号 NCK → PLC DB31, ... DBX68.5（可跨通道取轴）必须总是置一。对于所有其他轴，此位均置零。固定指定的 PLC 轴只能从中立轴到 PLC 轴或相反。

5.4.10 通过轴容器旋转取轴

使能轴容器旋转

当轴容器旋转使能时，便会自动生成隐性的 GET 或 GETD，将所有能够指定给通道的容器轴指定给该通道。轴容器旋转后才可将这些轴释放给另一个通道。

说明

轴处于“主处理轴”（例如 PLC 轴）状态时，**无法**将其隐性指定给通道。必须先退出该状态，轴才能参与轴容器旋转。

容器轴取轴的更多说明请见“B3: 分布式系统 - 仅适用于 840D sl (页 79)”章节。

示例：隐性生成 GET 或 GETD 的轴容器旋转

通道 1 动作

AXCTSWE (CT 1)

假设:

通道 2 动作

SPOS = 180 已定位
; 取通道 1 中的主轴
; 且允许轴容器旋转

5.4 跨通道取进给轴/主轴

主轴在两个通道中均使用，且为轴容器 CT 1 中的轴。

激活

使用机床数据 MD10722 \$MN_AXCHANGE_MASK，位 1=1 激活轴容器旋转隐含的 GET/GETD。

5.4.11 触发/不触发预处理停止的取轴

跨通道取轴功能的新增特性：不触发预处理停止

使用此 GET 指令只会生成一个中间程序段，而不是触发预处理停止的 GET 程序段。在主处理中执行该程序段时，系统会检查程序段中的轴状态是否与当前轴状态一致。若不一致，则可能会触发强制重组。

系统检查**进给轴或定位主轴**的以下状态：

- 是否是进给轴或定位主轴
- 设定位置

检查**处于转速模式的主轴**的以下状态：

- 主轴运行方式：转速模式
- 主轴转速 S
- 旋转方向 M3, M4
- 齿轮档 M40, M41, M42, M43, M44, M45
- 主轴是否为恒定切削速度

必要时可执行强制重组。从动轴则必须进行强制重组。

激活

不触发预处理停止的取轴以及对当前状态的检查通过以下机床数据激活：MD10722 \$MN_AXCHANGE_MASK，位 2=1。

示例

激活取轴，不触发预处理停止

表格 5-3

N010	M4	S1000	
N011	G4	F2	
N020	M5		
N021	SPOS=0		
N022	POS[B]=1		
N023	WAITP[B]		; 轴 b 转换为中立轴
N030	X1	F10	
N031	X100	F500	
N032	X200		
N040	M3	S500	
N041	G4	F2	
N050	M5		
N099	M30		

若在程序段 N023 后将主轴（轴 B）直接转换为 PLC 轴，使轴运行至 180°再返回 1°，然后再转换为中立轴，那么程序段 N040 不会触发预处理停止和重组。

特殊情况：触发预处理停止的取轴

只要 GET 或 GETD 指令还没有进入主处理，就可以用 RELEASE(轴) 或 WAIT(轴) 再次释放轴。接着的 GET 指令会导致预处理停止。

5.4.12 仅由 PLC 控制的轴

功能

控制系统启动后，轴处于“中立轴”状态，并由 PLC 控制。为了将其作为并行定位轴运行（由 PLC 通过功能块 FC18 运行），必须先通过 PLC 显性请求这些轴。

说明
可通过机床数据将 PLC 取轴范围限制为“仅由 PLC 控制的轴”。MD10722 \$MN_AXCHANGE_MASK, 位 3 = 1

该轴无法通过 NC 零件程序运行。

5.4 跨通道取进给轴/主轴

参数设置

通过以下轴专用机床数据将轴设置为仅由 PLC 控制的轴:

MD30460 \$MA_BASE_FUNCTION_MASK, 位 4 = 1

由 PLC 控制

仅由 PLC 控制的轴的运行特性只可通过轴 NC/PLC 接口信号调整:

- DB31, ... DBX28.1 (复位)
- DB31, ... DBX28.2 (继续)
- DB31, ... DBX28.6 (沿制动斜坡停止)

可采用的运行功能

仅由 PLC 控制的轴可使用以下运行功能:

1. 通过运行键和手轮在 JOG 模式下运行
2. 轴回参考点
3. 通过静态同步动作作为指令轴运行
4. 作为异步往复轴运行
5. 由 PLC 通过 FC18 作为并行定位轴运行

运行功能 1 至 4 结束后, 该轴自动恢复为“中立轴”状态。由 PLC 结束功能 5 后, 轴仍为 PLC 轴。只有由 PLC 明确释放后, 轴才变为中立轴。

5.4.13 固定指定给 PLC 的轴

功能

控制系统启动后, 轴处于“中立轴”状态, 且由 NC 通道控制。若要将其作为并行定位轴运行 (由 PLC 通过功能块 FC18 运行), 无须先通过 PLC 显性请求这些轴。通过 FC18 发出运行请求时 PLC 会自动取轴。通过 FC18 请求的运行结束后, 轴自动恢复为“中立轴”状态。

取轴完成后，借助 PLC 请求也可通过 PLC 控制该轴：“PLC 轴”状态。

说明

可通过机床数据将 PLC 取轴范围限制为“固定指定给 PLC 的轴”。MD10722 \$MN_AXCHANGE_MASK, 位 3 = 1

参数设置

通过以下轴专用机床数据将轴设置为固定指定给 PLC 的轴：

MD30460 \$MA_BASE_FUNCTION_MASK, 位 5 = 1

由 PLC 或 NC 通道控制

固定指定给 PLC 的轴的运行特性可通过 NC 通道或 PLC 调整：

NC 通道：通道专用 NC/PLC 接口信号（选择）

- DB21, ... DBXDBX7.1（NC 启动）
- DB21, ... DBXDBX7.3（NC 停止）
- DB21, ... DBXDBX7.7（复位）

PLC：轴 NC/PLC 接口信号

- DB31, ... DBX28.1（复位）
- DB31, ... DBX28.2（继续）
- DB31, ... DBX28.6（沿制动斜坡停止）

可采用的运行功能

固定指定给 PLC 的轴可使用以下运行功能：

1. 通过运行键和手轮在 JOG 模式下运行
2. 轴回参考点
3. 由 PLC 通过 FC18 作为并行定位轴运行

运行功能 1 至 3 结束后，该轴自动恢复为“中立轴”状态。

5.4.14 跨通道取轴和经过旋转的 WCS 中的几何轴的综合应用

跨通道取轴功能的新增特性：支持含旋转分量的框架指令

在 JOG 模式下，可通过静态同步动作将经过旋转的 WCS 中的几何轴作为 PLC 轴或指令轴运行。为此必须设置机床数据：MD32074

\$MA_FRAME_OR_CORRPOS_NOTALLOWED，位 10=1。该轴的再定位特性通过位 11 控制。

说明

JOG 模式下切换运行方式前

在 JOG 模式下，经过旋转的 WCS 中所有作为几何轴相互关联的 PLC 轴和指令轴必须结束运行，方可切换至其他运行方式。这些轴至少需要恢复为中立轴状态，否则运行方式切换时会发出报警 16908。此外，若在经过旋转的坐标系中只有一根几何轴作为 PLC 轴或指令轴运行，也会触发此报警。

此类轴只能在通道内转换为 PLC 轴或指令轴，不允许由其他通道取轴。

从 JOG 切换至 AUTO 的前提条件

在从 JOG 切换至 AUTO 运行方式时，仅当 MD 32074:

FRAME_OR_CORRPOS_NOTALLOWED 位 11=1 的情况下，程序中断时几何轴运动的终点才会被接收。此时会根据 WCS 中的旋转量定位 PLC 轴或指令轴。

受 WCS 旋转影响的**所有**轴均被归入一个几何轴轴组，并一同处理。因此，该组的所有轴均

- 被指定给 NC 程序，或者
- 所有轴为中立轴，或者
- 所有轴为主处理轴（PLC 轴、指令轴或往复轴）。

例如，几何轴组中的一根轴写入了 WAITP 时，系统会等待该组中的所有其他轴，从而将其一同转换为中立轴。若其中一根轴在主处理中转换为 PLC 轴，那么该组中的所有其他轴将转换为中立轴。

边界条件

若 MD32074 \$MA_FRAME_OR_CORRPOS_NOTALLOWED 位 10 == 0，且在 NC 程序中编写了 ROT Z45，则**无法**跨通道取 X 轴和 Y 轴。对于 Z 轴也存在类似情形，例如编写了

ROT X45 或 ROT Y45，或者 JOG 模式下程序段由于此类编程中断时。此时会为 X 轴和 Y 轴置位 NC/PLC 接口信号：

- DB31, ...DBX68.5（可跨通道取轴）= 1
- DB32, ...DBX68.5（可跨通道取轴）= 1

但是这些信号会被复位。

仅在 MD32074 \$MA_FRAME_OR_CORRPOS_NOTALLOWED 位 10 == 1，且当前执行的程序段不包含此类指令时，才可在 JOG 模式下跨通道取这些轴。

5.4.15 通过同步动作取轴

功能

通过同步动作，可使用 GET(轴) 请求一根轴，并使用 RELEASE(轴) 释放该轴用于取轴。

说明

相关轴必须通过机床数据作为通道轴指定给通道。

使用 NC 语言指令 AXTOCHAN，可通过同步动作或零件程序将轴直接传递给特定通道。此时不必是其自身通道，该通道也无需具备该轴的当前插补权限。

当前状态和轴的插补权限

跨通道取轴所涉及的轴类型和插补权限可通过系统变量 \$AA_AXCHANGE_TYP[轴] 读取：

- 0: 轴被指定给 NC 程序。
- 1: 轴被指定给 PLC，或者作为指令轴/往复轴生效。
- 2: 另一个通道具有插补权限。
- 3: 轴为中立轴。
- 4: 中立轴由 PLC 控制。
- 5: 另一个通道具有插补权限，已为 NC 程序请求该轴。
- 6: 另一个通道具有插补权限，已请求该轴作为中立轴。
- 7: 轴为 PLC 轴，或作为指令轴/往复轴生效，已为 NC 程序请求该轴。
- 8: 轴为 PLC 轴，或作为指令轴/往复轴生效，已请求该轴作为中立轴。

5.4 跨通道取进给轴/主轴

9: 固定指定的 PLC 轴，中立轴状态。

10: 固定指定的 PLC 轴，由 PLC 控制，中立轴状态。

固定指定的 PLC 轴

中立轴状态: `$AA_AXCHANGE_TYP = 9`。

由 PLC 控制，中立轴状态: `$AA_AXCHANGE_TYP = 10`

固定指定给 PLC，不受 **GET** 和 **RELEASE** 指令影响。

能否对该轴执行取轴通过系统变量 `$AA_AXCHANGE_STAT[轴]` 显示。

Figure 1: Configuration of the NC program for the main axis. The diagram illustrates the relationship between the PLC and the NC program for the main axis, showing the allocation of axes to different channels and the resulting states.

The diagram is divided into two main regions: a blue region for "请求用于 NC 程序" (Requested for NC program) and an orange region for "请求用作中立轴" (Requested for use as Neutral Axis).

Channel 1 (通道 1): Associated with $\$AA\_AXCHANGE\_TYP=3$ (Neutral Axis). It shows the "进给轴/主轴 =3" (Feed Axis/Main Axis =3) and "进给轴/主轴 =4" (Feed Axis/Main Axis =4) states. The "进给轴/主轴 上电后" (After power-up of Feed Axis/Main Axis) state is also shown.

Channel 2 (通道 2): Associated with $\$AA\_AXCHANGE\_TYP=4$ (Neutral Axis controlled by PLC). It shows the "进给轴/主轴 =2" (Feed Axis/Main Axis =2) and "进给轴/主轴 =6" (Feed Axis/Main Axis =6) states.

PLC 进给轴/主轴 =8 请求用作中立轴: This central state is associated with $\$AA\_AXCHANGE\_TYP=1$ (PLC =1) and $\$AA\_AXCHANGE\_TYP=7$ (PLC =7). It is labeled "指定给 PLC 或者作为主处理轴生效" (Specified to PLC or as the main processing axis becomes effective).

进给轴/主轴 =5: This state is associated with $\$AA\_AXCHANGE\_TYP=5$ and is labeled "请求用于 NC 程序" (Requested for NC program).

进给轴/主轴 =6: This state is associated with $\$AA\_AXCHANGE\_TYP=6$ and is labeled "请求用作中立轴" (Requested for use as Neutral Axis).

Legend:

- RSy: RELEASE (AX ...) 在同步动作中写入 $\$AA\_AXCHANGE\_TYP=0$
- //: 复位键
- R0: 由 PLC 设为中立状态
- Rn: 由 PLC 转交给最佳通道
- PLC: PLC 轴 指定给 PLC 或者作为主处理轴生效
- GSy: GET (AX ...) 在同步动作中写入 $\$AA\_AXCHANGE\_TYP=n$
- GSf: 取轴, 当在同步动作中满足条件时
- GP: GET 由 PLC 设定
- GD: GET 直接在 NC 程序中写入

功能手册之同步动作分册：章节“同步动作中的动作”

5.5 边界条件

5.4.16 跨通道取轴和龙门轴的综合应用

功能

跨通道取轴时，一个封闭的龙门轴组被当作一个单元处理。因此跨通道取龙门轴组中主动轴的操作也会同时获取轴组中的从动轴。此时除了要满足之前的章节中描述的适用于主动轴的前提条件外，还须满足该组所有从动轴的前提条件。

轴机床数据

在取龙门轴时，必须将该组所有轴的以下机床数据设为相同的值：

- MD30460 \$MA_BASE_FUNCTION_MASK, 位 4（控制分量）
- MD30460 \$MA_BASE_FUNCTION_MASK, 位 5（指定分量）

轴专用 NC/PLC 接口信号

在跨通道取轴功能范畴内，龙门轴组内所有轴的以下 NC/PLC 接口信号总是显示相同的值：

- DB31, ... DBX63.0（复位已执行）
- DB31, ... DBX63.1（PLC 控制轴）
- DB31, ... DBX63.2（轴停止生效）

轴专用系统变量

在跨通道取轴功能范畴内，龙门轴组内所有轴的以下系统变量总是显示相同的值：

- \$AA_AXCHANGE_TYP（跨通道取轴相关的轴类型）
- \$AA_AXCHANGE_STAT（跨通道取轴相关的轴状态）
- \$AA_SNGLAX_STAT（单轴类型）

5.5 边界条件

运行方式组

SINUMERIK 840D sl 可提供最多 10 个 运行方式组（BAG）。

SINUMERIK 828D 仅提供 1 个 BAG。

通道

SINUMERIK 840D sl 可提供最多 10 个 通道。

SINUMERIK 828D 仅提供 1 个通道。

跨通道取进给轴/主轴

SINUMERIK 828D 不支持跨通道取进给轴/主轴功能。

切换至通道轴

在将一根 PLC 轴、中立轴或其他通道中的轴转换为通道轴时，必须进行同步。

同步时会

- 接收当前位置
- 接收主轴当前转速和当前齿轮档。

因此系统需要强制执行预处理停止，这会使当前生效的轨迹运行中断。

若通过 GET 接收轴，则此切换通过零件程序明确定义。

但是若轴由 PLC 分配，则无法明确预见执行切换的程序部分。

（除了通过用户专用 NC <-> PLC 逻辑）

出于此原因，以下状态下轴将延迟切换至通道轴：

- 路径运行生效（已编写 G64+轴）
- 螺纹切削/攻丝生效（G33/G331/G332）

从通道轴切换到通道轴

路径运行生效期间无法将通道轴切换为中立轴或 PLC 轴。

因此在通过 RELEASE 释放轴时，RELEASE 必须占用一个单独的 NC 程序段。

PLC 转换轴类型时，系统内部会触发重组。因此上述程序状态下会出现延迟。

程序段搜索

执行带计算的程序段搜索时，系统会存储所有 GET、GETD 或 RELEASE 程序段，并在 NC 重新启动后输出。

5.6 数据表

例外：
相互冲突的程序段会被删除。
示例：

N10	RELEASE (AX1)	程序段被删除
N40	GET (AX1)	"
N70	目标	

5.6 数据表

5.6.1 机床数据

5.6.1.1 通用机床数据

编号	名称: \$MN_	描述
10010	ASSIGN_CHAN_TO_MODE_GROUP[n]	BAG 中生效的通道 [通道编号]: 0, 1
10722	AXCHANGE_MASK	跨通道取轴特性

5.6.1.2 通道专用机床数据

通道的基础机床数据

编号	名称: \$MC_	描述
20000	CHAN_NAME	通道名称
20050	AXCONF_GEOAX_ASSIGN_TAB[n]	几何轴指定为通道轴 [几何轴编号]: 0...2
20060	AXCONF_GEOAX_NAME_TAB[n]	通道中的几何轴名称 [几何轴编号]: 0...2
20070	AXCONF_MACHAX_USED[n]	通道内有效的机床轴编号 [通道轴编号]: 0...7

编号	名称: \$MC_	描述
20080	AXCONF_CHANAX_NAME_TAB[n]	通道内的通道轴名称 [通道轴编号]: 0...7
20090	SPIND_DEF_MASTER_SPIND	通道内主主轴的初始设置
20100	DIAMETER_AX_DEF	带端面轴功能的几何轴
20110	RESET_MODE_MASK	确定复位/零件程序结束后控制系统的初始设置
20112	START_MODE_MASK	确定 NC 启动时控制系统的初始设置
20150	GCODE_RESET_VALUES[n]	G 功能组的初始设置 [G 组编号]: 0...59
20160	CUBIC_SPLINE_BLOCKS	C 样条下程序段的数量
20170	COMPRESS_BLOCK_PATH_LIMIT	压缩时 NC 程序段的最大运行长度
20200	CHFRND_MAXNUM_DUMMY_BLOCKS	相位/半径处的空程序段
20210	CUTCOM_CORNER_LIMIT	TRC 中平衡程序段的最大角度
20220	CUTCOM_MAX_DISC	DISC 的最大值
20230	CUTCOM_CURVE_INSERT_LIMIT	TRC 中交点计算的最大角度
20240	CUTCOM_MAXNUM_CHECK_BLOCKS	TRC 中预读轮廓计算的程序段
20250	CUTCOM_MAXNUM_DUMMY_BLOCKS	TRC 中不含运行指令的程序段数量
20270	CUTTING_EDGE_DEFAULT	未作设定时刀沿的初始设置
20400	LOOKAH_USE_VELO_NEXT_BLOCK	预读 (Lookahead) 后续程序段的数量
20430	LOOKAH_NUM_OVR_POINTS	预读 (Lookahead) 倍率开关值
20440	LOOKAH_OVR_POINTS[n]	预读的倍率开关值 [开关值编号]: 0...1
20500	CONST_VELO_MIN_TIME	恒定速度下的最小时间
20600	MAX_PATH_JERK	最大轨迹加加速度
20610	ADD_MOVE_ACCEL_RESERVE	为叠加运动保留的加速裕量
20650	THREAD_START_IS_HARD	攻丝中轴的加速性能
20700	REFP_NC_START_LOCK	不带参考点的 NC 启动禁用
20750	ALLOW_GO_IN_G96	G96 时的 G0 逻辑
20800	SPF_END_TO_VDI	子程序结束发送至 PLC
21000	CIRCLE_ERROR_CONST	圆弧终点监控常量
21010	CIRCLE_ERROR_FACTOR	圆弧终点监控系数
21100	ORIENTATION_IS_EULER	定向编程时的角度定义

5.6 数据表

编号	名称: \$MC_	描述
21110	X_AXIS_IN_OLD_X_Z_PLANE	自动框架定义中的坐标系
21200	LIFTFAST_DIST	从轮廓快速退回时的运行行程
21250	START_INDEX_R_PARAM	第一个通道专用 R 参数的编号

通道的辅助功能设置

编号	名称: \$MC_	描述
22000	AUXFU_ASSIGN_GROUP[n]	辅助功能组 [通道中辅助功能编号]: 0...49
22010	AUXFU_ASSIGN_TYPE[n]	辅助功能类型 [通道中辅助功能编号]: 0...49
22020	AUXFU_ASSIGN_EXTENSION[n]	辅助功能扩展 [通道中辅助功能编号]: 0...49
22030	AUXFU_ASSIGN_VALUE[n]	辅助功能数值 [通道中辅助功能编号]: 0...49
22200	AUXFU_M_SYNC_TYPE	M 功能的输出时间
22210	AUXFU_S_SYNC_TYPE	S 功能的输出时间
22220	AUXFU_T_SYNC_TYPE	T 功能的输出时间
22230	AUXFU_H_SYNC_TYPE	H 功能的输出时间
22240	AUXFU_F_SYNC_TYPE	F 功能的输出时间
22250	AUXFU_D_SYNC_TYPE	D 功能的输出时间
22260	AUXFU_E_SYNC_TYPE (准备中)	E 功能的输出时间
22400	S_VALUES_ACTIVE_AFTER_RESE T	S 功能复位后保持生效
22410	F_VALUES_ACTIVE_AFTER_RESE T	F 功能复位后保持生效
22500	GCODE_OUTPUT_TO_PLC	到 PLC 的 G 功能
22550	TOOL_CHANGE_MODE	使用 M 功能时的新刀具补偿
22560	TOOL_CHANGE_M_CODE	用于换刀的 M 功能

通道专用存储器设置

编号	名称: \$MC_	描述
25000	REORG_LOG_LIMIT	用于使能日志文件的插补缓冲器大小, %
28000	MM_REORG_LOG_FILE_MEM	用于重组的存储空间 (DRAM)
28010	MM_NUM_REORG_LUD_MODULES	重组时用于本地用户变量的模块数量 (DRAM)
28020	MM_NUM_LUD_NAMES_TOTAL	本地用户变量的数量 (DRAM)
28030	MM_NUM_LUD_NAMES_PER_PRO G	每个程序的本地用户变量的数量 (DRAM)
28040	MM_LUD_VALUES_MEM	本地用户变量的存储空间 (DRAM)
28050	MM_NUM_R_PARAM	通道专用 R 参数的数量 (SRAM)
28060	MM_IPO_BUFFER_SIZE	插补缓存中 NC 程序段的数量 (DRAM)
28070	MM_NUM_BLOCKS_IN_PREP	预处理的程序段数量(DRAM)
28080	MM_NUM_USER_FRAMES	可设置框架的数量 (SRAM)
28090	MM_NUM_CC_BLOCK_ELEMENTS	用于编译循环的程序段单元的数量 (DRAM)
28100	MM_NUM_CC_BLOCK_USER_MEM	用于编译循环的程序段存储空间 (DRAM)
28500	MM_PREP_TASK_STACK_SIZE	准备任务的栈大小 (DRAM)
28510	MM_IPO_TASK_STACK_SIZE	插补任务的栈大小(DRAM)

5.6.1.3 进给轴/主轴专用机床数据

编号	名称: \$MA_	描述
30460	BASE_FUNCTION_MASK	轴功能
30550	AXCONF_ASSIGN_MASTER_CHAN	用于跨通道取轴的通道缺省设置
30552	AUTO_GET_TYPE	自动 GET 的定义
30600	FIX_POINT_POS	G75 中轴的固定值位置
32074	FRAME_OR_CORRPOS_NOTALLOWED	不允许使用框架或 HL 补偿
33100	COMPRESS_POS_TOL	压缩时的最大偏差

5.6 数据表

5.6.2 设定数据

5.6.2.1 通道专用设定数据

编号	名称: \$SC_	描述
42000	THREAD_START_ANGLE	螺纹加工起始角
42100	DRY_RUN_FEED	空运行进给率

5.6.3 信号

5.6.3.1 发送至/来自 BAG 的信号

从 PLC 发送至 NCK、从 NCK 发送至 PLC 的 BAG 信号包含在数据块 11 中。

这些信号的说明参见：

文档：

功能手册之基本功能分册；章节“NC/PLC 接口信号（Z1）”；

“BAG, 通道, 程序运行(K1)”

5.6.3.2 发送至/来自通道的信号

从 PLC 发送至 NCK、从 NCK 发送至 PLC 的通道信号包含在数据块 21、22、... 中，分别对应第一个通道、第二个通道。

这些信号的说明请见：

文档：

功能手册 “基本功能”；章节“NC/PLC 接口信号（Z1）”；

“BAG, 通道, 程序运行(K1)”

M1: 坐标转换

6.1 简要说明

6.1.1 TRANSMIT（选件）

说明

“TRANSMIT”功能需要使用“TRANSMIT 和柱面转换”选件，该选件需授权。

“TRANSMIT”功能有以下作用：

- 对工件端面进行铣削/钻削
 - 钻孔
 - 轮廓
- 可使用一个直角坐标系来编程这些加工。
- 控制系统将编写的直角坐标系运行转换成真实的机床轴运行（标准情况）：
 - 回转轴 (1)
 - 垂直于回转轴的横轴 (2)
 - 平行于回转轴的纵轴 (3)直线轴 (2) 和 (3) 相互垂直
- 允许刀具中心相对于旋转中心发生偏移。
- 速度控制功能会考虑为旋转运动定义的限制范围。
- 直角坐标系中的运动轨迹不允许穿过旋转中心（此限制针对软件版本 2 和 3）

其他可用设置

- 刀具中心点轨迹可穿过回转轴的旋转中心。
- 回转轴不必为模数轴。

6.1 简要说明

6.1.2 TRACYL (选件)

说明

“柱面转换 (TRACYL)” 功能需要使用“TRANSMIT 和柱面转换” 选件，该选件需获得授权。

“柱面转换 (TRACYL)” 功能有以下作用：

加工：

- 圆柱体上的纵向槽，
- 圆柱体上的横向槽，
- 圆柱体上任意形状的槽。

槽轮廓的编程基于展开的圆柱外表面。

该加工支持

- “X+C+Z” 车床和
- “X+Y+C+Z”

车床。

- 控制系统将编写的柱面坐标系运行转换成真实的机床轴运行（第一种车床的缺省设置，TRAFO_TYPE_n = 512）：
 - 回转轴 (1)
 - 垂直于回转轴的横轴 (2)
 - 平行于回转轴的纵轴 (3)

说明

直线轴 (2) 和 (3) 相互垂直 横轴 (2) 与回转轴相交。此配置不支持槽壁补偿。

- 槽壁补偿需要包含下列轴的“X+Y+C+Z” 车床（TRAFO_TYPE_n = 513）：
 - 回转轴 (1)
 - 垂直于回转轴的横轴 (2)
 - 平行于回转轴的纵轴 (3)
 - 纵轴 (4)和轴 (2)、(3) 一同构成右手直角坐标系。

说明

直线轴 (2)、(3) 和 (4) 相互垂直。此配置支持槽壁补偿。

- 速度控制功能会考虑为旋转运动定义的限制范围。

TRACYL 转换，无槽壁补偿，有附加纵轴（无槽壁补偿的柱面转换 TRAFO_TYPE_n=514）

- 无槽壁补偿的坐标转换只需要一根回转轴以及一根垂直于该轴的直线轴即可。
- 若机床配备了另一根直线轴垂直于回转轴和第一根直线轴，则可将其用于改进刀具补偿。

6.1.3 TRAANG（选件）

说明

“斜轴（TRAANG）”功能需使用“斜轴”选件，该选件需获得授权。

“斜轴（TRAANG）”功能设计用于磨削工艺。此功能有以下作用：

- 斜轴加工。
- 可使用一个直角坐标系来编程。
- 控制系统将编写的直角坐标系运行转换成真实的机床轴运行（标准情况）：斜轴。

6.1.4 坐标转换级联

引言

可将两个坐标转换互联在一起（级联），这样便可将第一坐标转换中的轴运动分量作为相连的第二坐标转换的输入数据。第二坐标转换的运动分量由机床轴执行。

级联设置

- 级联允许包含**两个**坐标转换。
- **第二个**坐标转换必须为“斜轴”（TRAANG）。
- 第一坐标转换可以为：
 - 定向转换（TRAORI），包括万向铣头
 - TRANSMIT
 - TRACYL
 - TRAANG

6.2 端面转换 TRANSMIT (选件)

关于坐标转换的相关信息请见:

文档:

功能手册之特殊功能分册; 章节“3 轴转换到 5 轴转换 (F2)”

6.1.5 通过零件程序/软键激活坐标转换机床数据

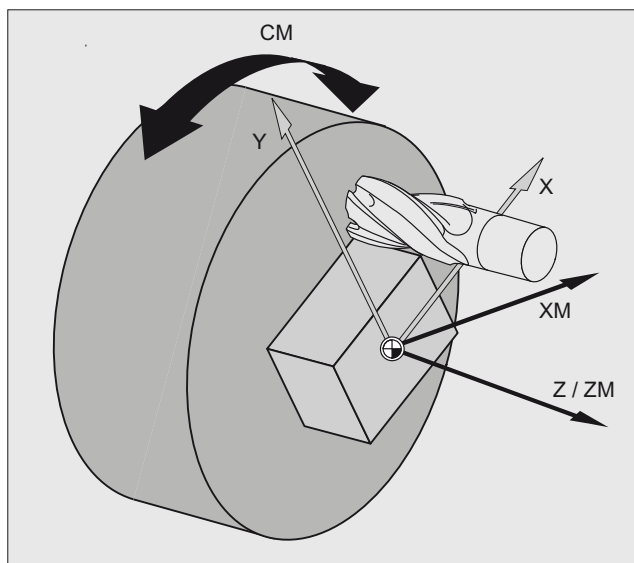
运动转换所涉及的机床数据目前多数为上电生效。

也可通过零件程序/软键激活这些机床数据, 这样便不用重启控制系统。

6.2 端面转换 TRANSMIT (选件)

核心内容

TRANSMIT 可实现在车床上进行端面加工。



X 几何轴

Y 几何轴

Z 几何轴

XM 机床轴

ZM 机床轴

CM 机床轴

说明

转换激活时，参与转换的机床轴、通道轴和几何轴的名称须不同：

- MD10000 \$MN_AXCONF_MACHAX_NAME_TAB (机床轴名称)
 - MD20080 \$MC_AXCONF_CHANAX_NAME_TAB (通道轴名称)
 - MD20060 \$MC_AXCONF_GEOAX_NAME_TAB (几何轴名称)
-

机床数据**通用转换数据**

- \$MC_TRAFO_GEOAX_ASSIGN_TAB_<n>
- \$MC_TRAFO_TYPE_<n>
- \$MC_TRAFO_AXES_IN_<n>

其中 n = 1、2、3 ... 最大坐标转换数据组数

说明

每个通道最多可以定义两个 TRANSMIT 转换

每个通道最多可以定义两个 TRANSMIT 转换：

\$MC_TRAFO_TYPE_<n> = 256 或 257

TRANSMIT 专用机床数据

- \$MC_TRANSMIT_ROT_AX_OFFSET_<t>
- \$MC_TRANSMIT_ROT_SIGN_IS_PLUS_<t>
- \$MC_TRANSMIT_BASE_TOOL_<t>
- \$MC_TRANSMIT_POLE_SIDE_FIX_<t>

其中 t = 1、2

6.2.1 专用设置**一根回转轴和一根直线轴：TRAFO_TYPE = 256**

带一根回转轴和一根直线轴的 TRANSMIT 须设置坐标转换类型 256：

6.2 端面转换 TRANSMIT (选件)

\$MC_TRAFO_TYPE_<n> = 256

其中 n = 1、2、... 最大坐标转换数

坐标转换中的输入轴: \$MC_TRAFO_AXES_IN_<n>

机床数据中会指定通道轴，转换直角坐标系中的轴须映射在这些通道轴上。

\$MC_TRAFO_AXES_IN_<n>[<下标>] = <通道轴编号>

其中 n = 1、2、... 最大坐标转换数

<下标>	含义
0	垂直于回转轴的直线轴
1	回转轴
2	平行于回转轴的直线轴

通道轴编号须对应通过 **\$MC_TRAFO_GEOAX_ASSIGN_TAB_<n>** 定义的轴顺序。

一根回转轴和两根直线轴: TRAFO_TYPE = 257

带一根回转轴和两根直线轴的 TRANSMIT 须设置坐标转换类型 257:

\$MC_TRAFO_TYPE_<n> = 257

其中 n = 1、2、... 最大坐标转换数

此时，第二根直线轴必须垂直于平面，由回转轴和直线轴夹紧。TRANSMIT 中的第二根直线轴只可用于刀具补偿。

坐标转换中的输入轴: \$MC_TRAFO_AXES_IN_<n>

机床数据中会指定通道轴，转换直角坐标系中的轴须映射在这些通道轴上。

\$MC_TRAFO_AXES_IN_<n>[<下标>] = <通道轴编号>

其中 n = 1、2、... 最大坐标转换数

<下标>	含义
0	垂直于回转轴的直线轴
1	回转轴
2	平行于回转轴的直线轴
3	垂直于下标 0 和 1 的直线轴

通道轴编号须对应通过 **\$MC_TRAFO_GEOAX_ASSIGN_TAB_<n>** 定义的轴顺序。

回转轴偏移: TRANSMIT_ROT_AX_OFFSET

如果回转轴的零点与 TRANSMIT 激活时回转轴的零点不重合，则须将角度差作为偏移写入机床数据中：

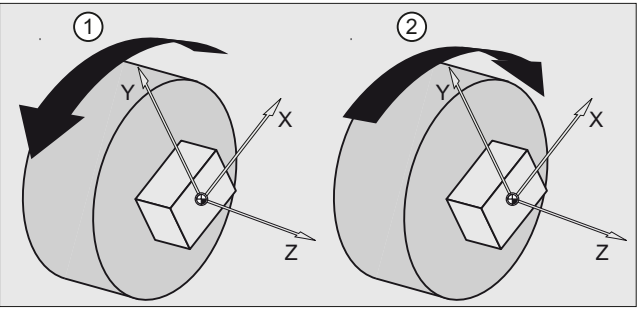
\$MC_TRANSMIT_ROT_AX_OFFSET_<t> = <角度差>

其中 t = 1、2

旋转方向: TRANSMIT_ROT_SIGN_IS_PLUS

借助以下机床数据，TRANSMIT 须能识别出回转轴的旋转方向：

- 当回转轴顺时针旋转时，TRANSMIT 为正向。而面朝 Z 轴观察时，X/Y 平面中回转轴为逆时针旋转。
- 当回转轴逆时针旋转时，TRANSMIT 为负向。



① 顺时针
② 逆时针

图 6-1 回转轴的旋转方向

\$MC_TRANSMIT_ROT_SIGN_IS_PLUS_<t> = <旋转方向>

其中 t = 1、2

<旋转方向>	含义
0	回转轴逆时针旋转 ⇒ 内部符号换向
1	回转轴顺时针旋转 ⇒ 无内部符号换向

6.2 端面转换 TRANSMIT (选件)

刀具零点位置: TRANSMIT_BASE_TOOL

须根据 TRANSMIT 生效直角坐标系原点来指定刀具零点的位置:

- MD24920 \$MC_TRANSMIT_BASE_TOOL_<t>[0] = <X 轴中的偏移>
- MD24920 \$MC_TRANSMIT_BASE_TOOL_<t>[1] = <Y 轴中的偏移>
- MD24920 \$MC_TRANSMIT_BASE_TOOL_<t>[2] = <Z 轴中的偏移>

其中 t = 1、2

可切换的几何轴

切换几何轴 GEOAX () 时, 系统会向 NC/PLC 接口输出编程的 M 功能。

- MD22534 \$MC_TRAFO_CHANGE_M_CODE = <M 功能>

说明

不输出值 0 - 6、17 和 30。

文档:

功能手册之基本功能分册; 章节“坐标系、轴类型、轴配置、工件相关的实际值系统、外部零点偏移”(K2)

6.2.2 激活

转换 TRANSMIT 是在零件程序或同步动作中通过指令 TRANSMIT 激活的。

句法

TRANSMIT
TRANSMIT (n)
TRAFOOF

含义

TRANSMIT: 激活含第一个 TRANSMIT 数据组的 TRANSMIT 功能
TRANSMIT (n): 激活含第 n 个 TRANSMIT 数据组的 TRANSMIT 功能

6.2.3 取消

一个在通道中生效的转换 TRANSMIT 是通过以下指令取消的:

- 取消转换: TRAFOOF
- 激活另一个转换, 如: TRACYL、TRAANG、TRAORI

6.2.4 应用

引言

TRANSMIT 的极点为 TRANSMIT 平面的零点 (图 2-1: $x = 0$, $Y = 0$)。极点位于直线轴和回转轴的交点处 (示例中为 **X** 和 **CM**)。在极点附近, 几何轴位置的微小变化通常会引起机床回转轴位置的巨大变化。但逼近极点的线性运动或过极点不会引起该变化。

过极点的刀具中心点轨迹不会引起零件程序终止。在可编程位移指令或有效刀具半径补偿方面无限制。但是仍不建议在极点附近加工工件, 因为有些条件下可能需要大幅降低进给率来避免回转轴过载。

新特性

定义:

极点是指刀具中心点轨迹直线与回转轴的旋转中心的交点。

涉及此定义的情形如下:

- 过极点运行的前提条件和方法
- 极点附近的特性
- 工作区域限制的特性
- 回转轴旋转超 360° 时的检查

过极点运行

过极点有两种方案:

- 直线轴单独运行
- 直线轴运行至极点, 回转轴在极点处旋转

直线轴单独运行

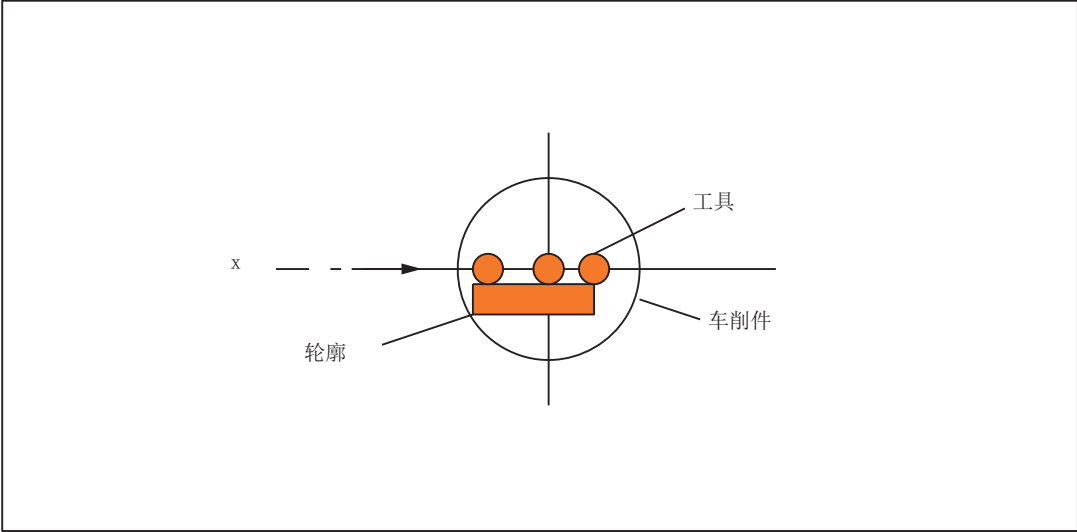


图 6-2 X 轴过极点

在极点处旋转

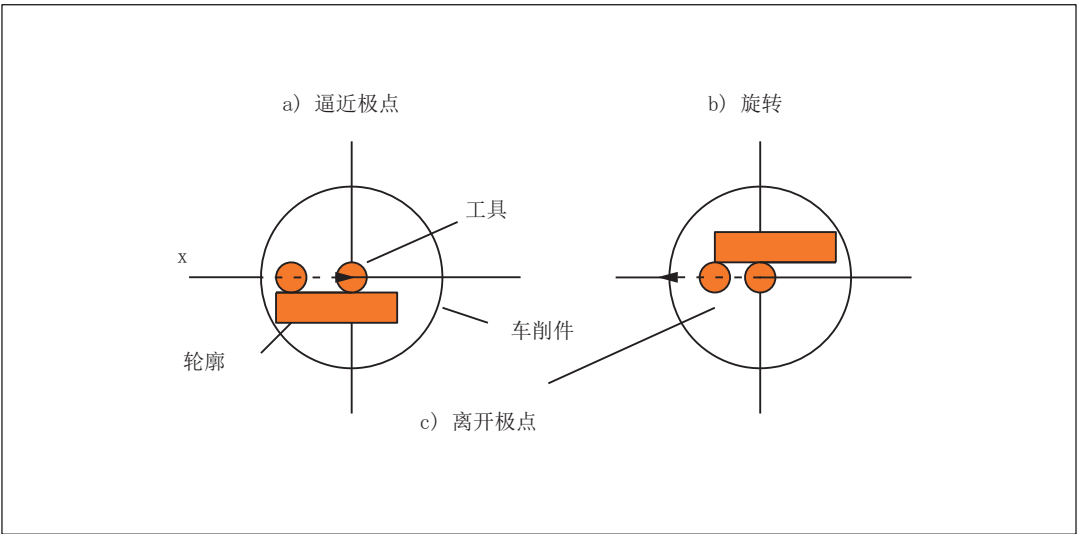


图 6-3 X 轴向极点移动(a)，旋转 (b)，离开极点 (c)

方案选择

选择方案时，须将机床性能和对待加工零件的要求作为依据。通过下列机床数据进行选择：

MD24911 \$MC_TRANSMIT_POLE_SIDE_FIX_1

MD24951 \$MC_TRANSMIT_POLE_SIDE_FIX_2

第一个机床数据对应通道中的第一个 TRANSMIT 转换，第二个则对应通道中的第二个 TRANSMIT 转换。

值	含义
0	过极点运行 刀具中心点轨迹（直线轴）始终穿过极点。
1	绕极点旋转。 刀具中心点轨迹只位于直线轴的正运行范围内（旋转中心前）。
2	绕极点旋转。 刀具中心点轨迹只位于直线轴的负运行范围内（旋转中心后）。

过极点的特性

AUTO 和 JOG 模式下均允许直线轴单独过极点。

特性：

表格 6-1 直线轴单独过极点

运行方式	状态	响应
自动方式	参与坐标转换的所有轴同步运行。TRANSMIT 生效。	快速过极点
	不是参与坐标转换的所有轴都同步运行。（例如定位轴） TRANSMIT 未生效。	缓慢过极点
	启用的 DRF（外部零点偏移）不起作用。启用 DRF 时，极点附近可能出现伺服故障。	加工终止，发出报警
JOG	-	缓慢过极点

在极点处旋转的特性

前提条件：此方案仅在 AUTO 模式下生效。

MD24911 \$MC_TRANSMIT_POLE_SIDE_FIX_1 = 1 或 2

6.2 端面转换 TRANSMIT (选件)

MD24951 \$MC_TRANSMIT_POLE_SIDE_FIX_2 = 1 或 2

值 1: 直线轴保持在正运行区域

值 2: 直线轴保持在负运行区域

若轮廓需要刀具中心点轨迹穿过极点，可通过下面三个步骤防止直线轴运行到远离旋转中心的区域。

步骤	操作
1	将直线轴运行至极点
2	将回转轴旋转 180°，参与坐标转换的其他轴保持静止。
3	执行剩余程序段。此时直线轴会再次离开极点。

在 JOG 模式下，运动会在极点处停止。在此模式下只能沿轨迹切线运行至极点，再沿此切线离开极点。所有其他运动指令都会导致回转轴位置急剧变化或机床急剧运动，即使运动指令为最小位移指令。系统会拒绝这些指令，并发出报警 21619。

在极点附近运行

若刀具中心点轨迹从极点旁经过，控制系统会自动降低进给率和轨迹加速度，以确保不会超出机床数据（MD32000 \$MA_MAX_AX_VELO[AX*] 和 MD32300 \$MA_MAX_AX_ACCEL[AX*]）的赋值。轨迹距极点越近，进给率的削减幅度也就越大。

拐角位于极点处的刀具中心点轨迹

刀具中心点轨迹的拐角位于极点处时，不仅意味着轴速度急剧变化，还意味着回转轴的位置急剧变化。这种急剧变化无法通过制动来降低。

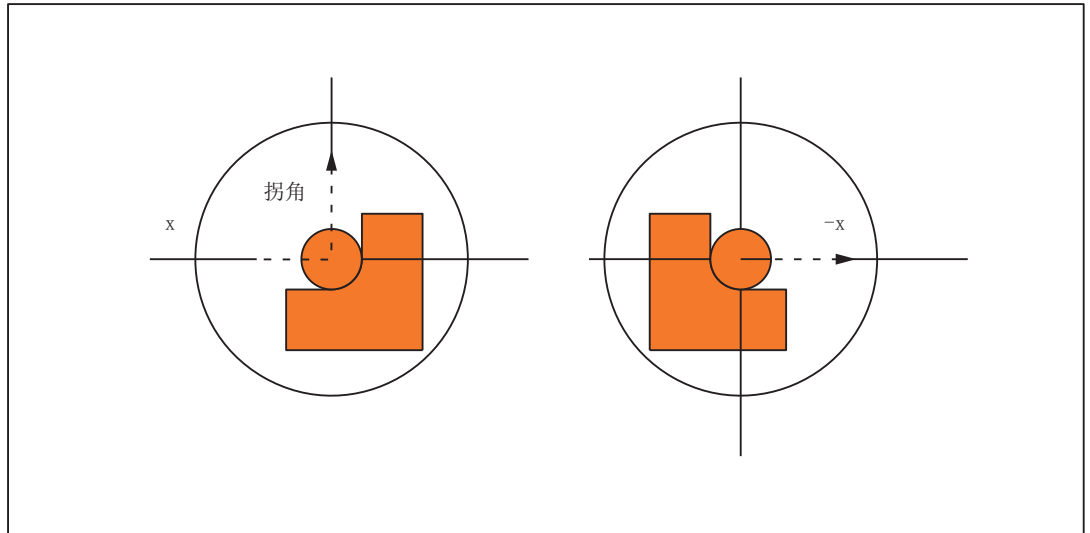


图 6-4 过极点

前提条件:

AUTO 运行方式,

MD24911 \$MC_TRANSMIT_POLE_SIDE_FIX_1 = 0

或者

MD24951 \$MC_TRANSMIT_POLE_SIDE_FIX_2 = 0

控制系统会在阶跃位置插入一个运行程序段，借此产生一个尽可能小的旋转量，从而能继续加工轮廓。

拐角不过极点

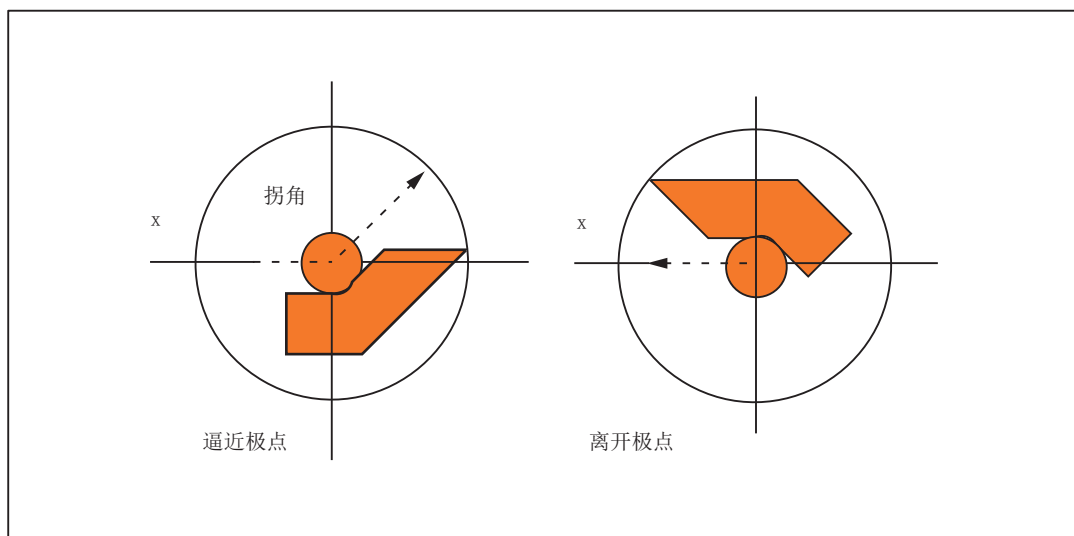


图 6-5 在极点一侧加工

前提条件:

AUTO 运行方式,

MD24911 \$MC_TRANSMIT_POLE_SIDE_FIX_1 = 1 或 2

或者

MD24951 \$MC_TRANSMIT_POLE_SIDE_FIX_2 = 1 或 2

控制系统会在阶跃位置插入一个运行程序段, 借此产生**所需的旋转量**, 从而能在极点的一侧继续加工轮廓。

极点处的坐标转换选择

若需从刀具中心点轨迹的一个位置开始继续加工, 而该位置是激活的坐标转换的极点, 则需要为新转换设定离开极点的运行。

如果设置了

MD24911 \$MC_TRANSMIT_POLE_SIDE_FIX_1= 0

或者

MD24951 \$MC_TRANSMIT_POLE_SIDE_FIX_2 = 0

(过极点), 则会在离开极点的程序段开始处产生一个**尽可能小**的旋转量。根据此旋转量, 轴在旋转中心前或旋转中心后运行。

如果

MD24911 \$MC_TRANSMIT_POLE_SIDE_FIX_1= 1

或者

MD24951 \$MC_TRANSMIT_POLE_SIDE_FIX_2 = 1

则轴在旋转中心前进行加工（直线轴处于正方向运行范围）；如果

MD24911 \$MC_TRANSMIT_POLE_SIDE_FIX_1= 2

或者

MD24951 \$MC_TRANSMIT_POLE_SIDE_FIX_2 = 2

轴在旋转中心后进行加工（直线轴处于负方向运行范围）

极点外的坐标转换选择

控制系统运行参与坐标转换的轴，不分析机床数据 MD24911

\$MC_TRANSMIT_POLE_SIDE_FIX_t。此时 t = 1 对应通道中的第一个 TRANSMIT 坐标转换，t = 2 对应第二个。

6.2.5 工作区域限制

初始情形

TRANSMIT 生效时，若无法将刀具中心点定位至参与转换的回转轴的旋转中心，则会启用工作区域限制取代极点。当垂直于回转轴（考虑刀具补偿）的轴与回转轴不处于同一径向平面，或者两根轴相互倾斜时会出现此情形。两根轴的间距构成了一个 BCS 中轴无法接近的柱形区域。

软件限位开关监控无法涵盖此区域，因为这不是机床轴的运行范围。

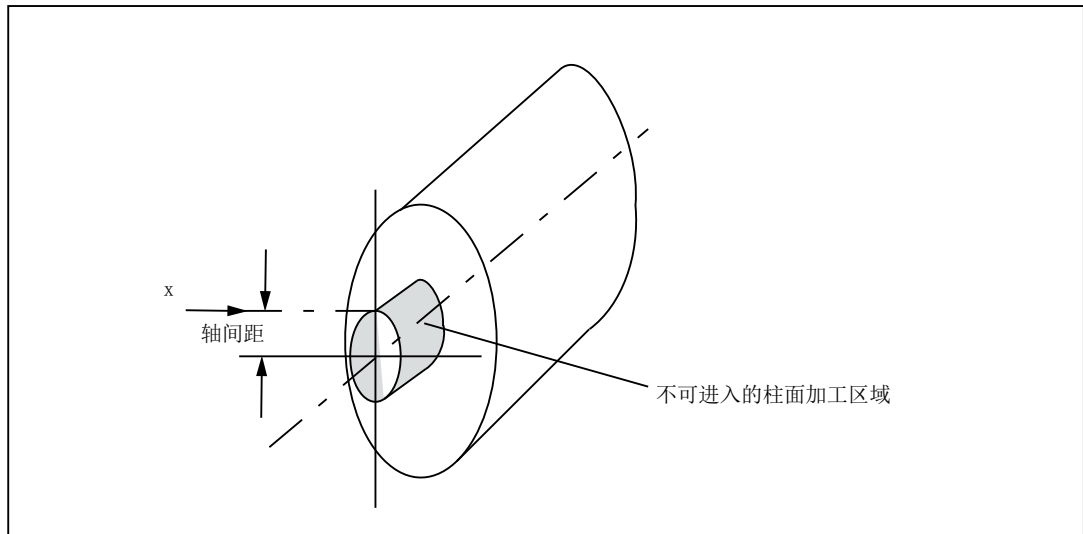


图 6-6 直线轴偏移产生的工作区域限制

运行至限制区域

进入限制区域的运行会被系统拒绝，并会触发报警 21619。对应的零件程序段不会被执行。控制系统会停留在前一程序段结束处。

无法充分预见此类超限运行时（JOG 模式，定位轴），控制系统会停止在限制区域边界处。

限制区域附近的特性

若刀具中心点轨迹从限制区域旁经过，控制系统会自动降低进给率和轨迹加速度，以确保不会超出机床数据（MD32000 \$MA_MAX_AX_VELO[AX*] 和 MD32300 \$MA_MAX_AX_ACCEL[AX*]）的赋值。轨迹距限制区域越近，进给率的削减幅度就可能越大。

6.2.6 TRANSMIT 下的叠加运动

控制系统无法预见叠加运动。叠加运动相对于和极点（或工作区域限制）的距离非常微小（例如刀具精补偿），不会影响功能。坐标转换功能会监控关键轴的叠加运动，并在出现临界值时发出报警 21618。此报警显示，程序段相关的速度控制不再能准确监控机床上的实际条件。因此触发此报警的同时会激活传统的、未经优化的在线速度控制。控制系统内部会生成 REORG，以将预处理和主处理重新同步。

报警表明可能会导致轴过载，从而使零件程序处理终止，因此用户应避免触发此报警。

6.2.7 回转轴旋转超 360 度时的检查

回转轴位置的不唯一性

转数使得回转轴的位置具有不唯一性。控制系统会将包含多次绕极点旋转的程序段划分为子程序段。

在同步动作（例如输出辅助功能、程序段同步的定位轴运行）中须注意此划分，因为同步不再以所编写的程序段结束为准，而是以第一个子程序段的结束为准。

文档：

功能手册之基本功能分册：“H2: 输出给 PLC 的辅助功能”

功能手册之同步动作分册

在单步方式中，控制系统会精确处理各程序段。在其他方式下，子程序段会通过 Look Ahead 像单个程序段一样处理。回转轴的运行区域通过软件限位开关监控。

6.2.8 边界条件

预读

所有需要预读的功能（过极点运行，Look Ahead）仅在能预先精确计算出轴运行的情况下才能充分发挥作用。TRANSMIT 中涉及预读的是回转轴以及与其垂直的直线轴。若这些轴中的一根是定位轴，则会发出报警 10912，取消预读功能并激活传统的在线速度控制。

方案选择

用户须自行在“过极点运行”和“绕极点旋转”中选择一个最佳方案。

多次过极点

一个程序段中可编写任意多次的过极点运行（例如通过编写一个多圈螺线 [Helix]）。该程序段会相应划分为较小的子程序段。与之类似，多次绕极点旋转的程序段也会被划分为子程序段。

回转轴是模数轴

回转轴可以设为一根模数回转轴。与软件版本 2 和 3 不同，此条件不是前提条件。软件版本 2 和 3 中的限制条件失效。

6.2 端面转换 TRANSMIT (选件)

回转轴是主轴

若在无坐标转换的情况下将回转轴用作主轴，在选择坐标转换前必须通过 SPOS 将主轴切换至位置控制模式。

包含附加直线轴的 TRANSMIT

在零件程序中，TRANSMIT 生效时 posBCS[ax[3]] 的通道标识须采用另一个名称，就像几何轴一样。如果轴已指定为几何轴，那只在 TRANSMIT 外写入 posBCS[ax[3]] 时无此限制。TRANSMIT 生效时，系统不会处理 ax[3] 相关的轮廓信息。

REPOS

该指令可用于重新定位到软件版本 4 中 TRANSMIT 产生的子程序段。此时控制系统会使用 BSC 中距目标位置最近的子程序段。

程序段搜索

若通过软件版本 4 的扩展功能生成了中间程序段，在带计算功能的程序段搜索中，系统会搜索最后一个子程序段的终点。

6.2.9 示例：轴配置

带两根直线轴和一根回转轴的 TRANSMIT 轴配置示例。

通道轴

几何轴名称

- MD20060 \$MC_AXCONF_GEOAX_NAME_TAB[0]="X"
- MD20060 \$MC_AXCONF_GEOAX_NAME_TAB[1]="Y"
- MD20060 \$MC_AXCONF_GEOAX_NAME_TAB[2]="Z"

为几何轴指定对应的通道轴

- TRANSMIT 未生效
 - MD20050 \$MC_AXCONF_GEOAX_ASSIGN_TAB[0] = 1 (第 1 几何轴 ⇒ 第 1 通道轴 “XC”)
 - MD20050 \$MC_AXCONF_GEOAX_ASSIGN_TAB[1] = 0 (-)
 - MD20050 \$MC_AXCONF_GEOAX_ASSIGN_TAB[2] = 2 (第 3 几何轴 ⇒ 第 2 通道轴 “ZC”)
- TRANSMIT 生效
 - MD24120 \$MC_TRAFO_GEOAX_ASSIGN_TAB_1[0] = 1 (第 1 几何轴 ⇒ 第 1 通道轴 “XC”)
 - MD24120 \$MC_TRAFO_GEOAX_ASSIGN_TAB_1[1] = 3 (第 2 几何轴 ⇒ 第 3 通道轴 “CC”)
 - MD24120 \$MC_TRAFO_GEOAX_ASSIGN_TAB_1[2] = 2 (第 3 几何轴 ⇒ 第 2 通道轴 “ZC”)

通道轴名称

- MD20080 \$MC_AXCONF_CHANAX_NAME_TAB[0] = "XC"
- MD20080 \$MC_AXCONF_CHANAX_NAME_TAB[1] = "ZC"
- MD20080 \$MC_AXCONF_CHANAX_NAME_TAB[2] = "CC"
- MD20080 \$MC_AXCONF_CHANAX_NAME_TAB[3] = "ASC"

为通道轴指定对应的机床轴

- MD20070 \$MC_AXCONF_MACHAX_USED[0] = 2 (第 2 机床轴 “XM”)
- MD20070 \$MC_AXCONF_MACHAX_USED[1] = 3 (第 3 机床轴 “ZM”)
- MD20070 \$MC_AXCONF_MACHAX_USED[2] = 1 (第 1 机床轴/主轴 “CM”)
- MD20070 \$MC_AXCONF_MACHAX_USED[3] = 4 (第 4 机床轴/主轴 “ASM”)

机床轴

主轴

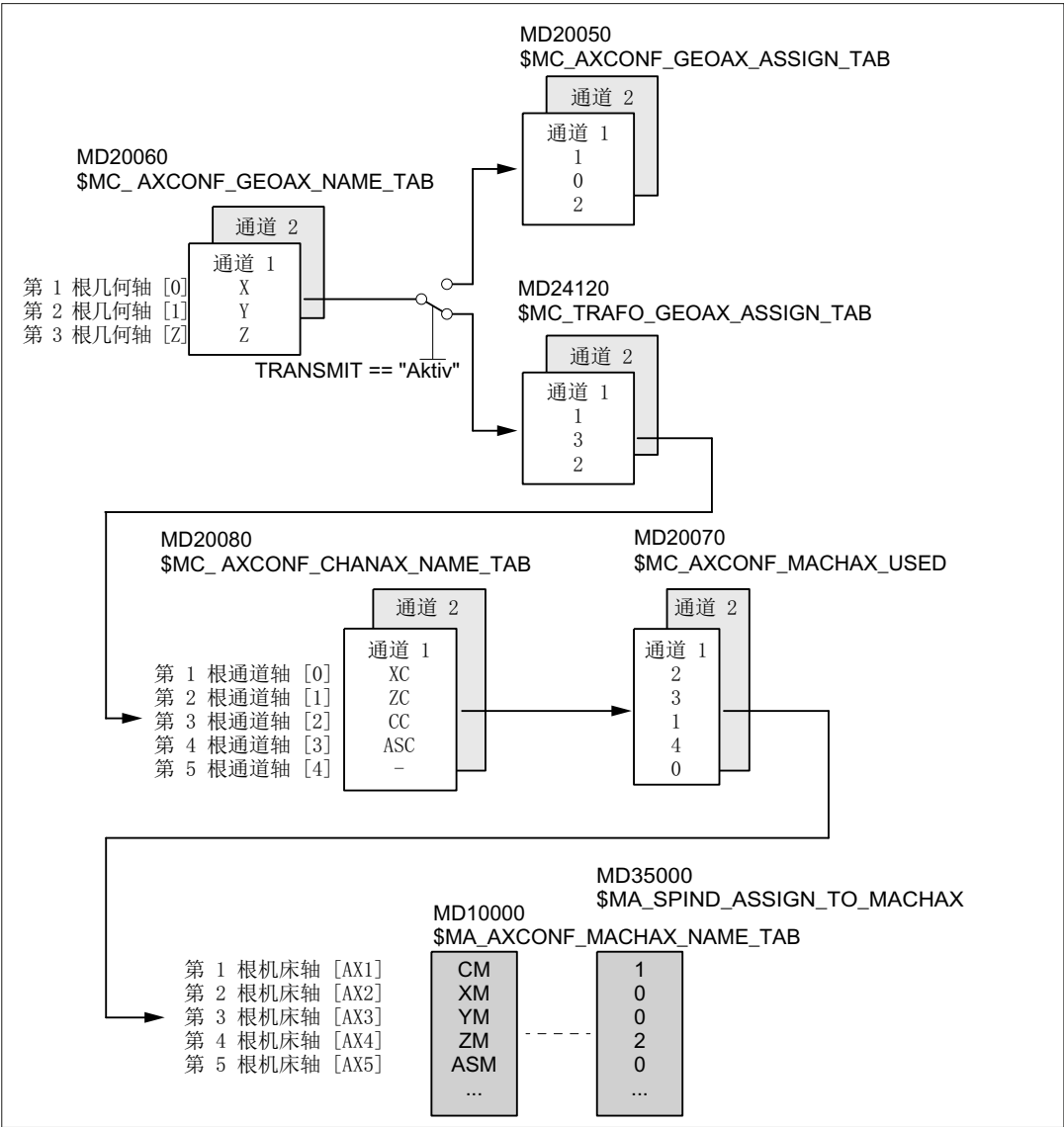
- MD35000 \$MA_SPIND_ASSIGN_TO_MACHAX[0] = 1 (第 1 机床轴 ⇒ 第 1 主轴)
- MD35000 \$MA_SPIND_ASSIGN_TO_MACHAX[1] = 0 (第 2 机床轴 ⇒ 进给轴)
- MD35000 \$MA_SPIND_ASSIGN_TO_MACHAX[2] = 0 (第 3 机床轴 ⇒ 进给轴)
- MD35000 \$MA_SPIND_ASSIGN_TO_MACHAX[3] = 2 (第 4 机床轴 ⇒ 第 2 主轴)

6.2 端面转换 TRANSMIT (选件)

机床轴名称

- MD10000 \$MN_AXCONF_MACHAX_NAME_TAB[0] = "CM"
- MD10000 \$MN_AXCONF_MACHAX_NAME_TAB[1] = "XM"
- MD10000 \$MN_AXCONF_MACHAX_NAME_TAB[2] = "ZM"
- MD10000 \$MN_AXCONF_MACHAX_NAME_TAB[3] = "ASM"

轴配置简图



6.3 柱面转换 TRACYL (选件)

TRACYL 可实现在车床上加工柱面转换曲线 (槽)。机床必须与柱面坐标系相匹配:

- 一根、两根或三根直线轴及一根回转轴
- 直线轴必须相互垂直
- 回转轴必须平行于其中一根直线轴

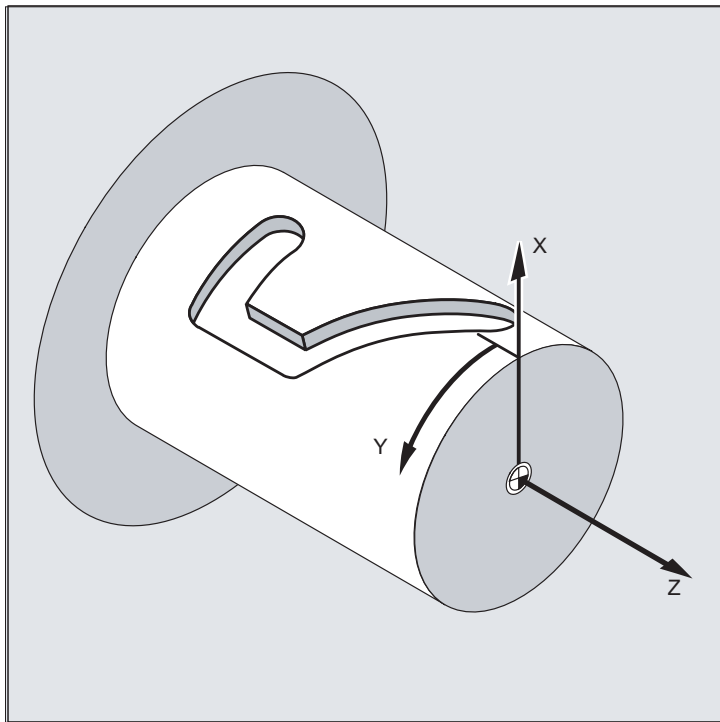


图 6-7 柱面转换 TRACYL

说明

转换激活时, 参与转换的机床轴、通道轴和几何轴的名称须不同:

- MD10000 \$MN_AXCONF_MACHAX_NAME_TAB (机床轴名称)
- MD20080 \$MC_AXCONF_CHANAX_NAME_TAB (通道轴名称)
- MD20060 \$MC_AXCONF_GEOAX_NAME_TAB (几何轴名称)

6.3 柱面转换 TRACYL (选件)

一根或两根直线轴 (轴配置 1)

一根直线轴

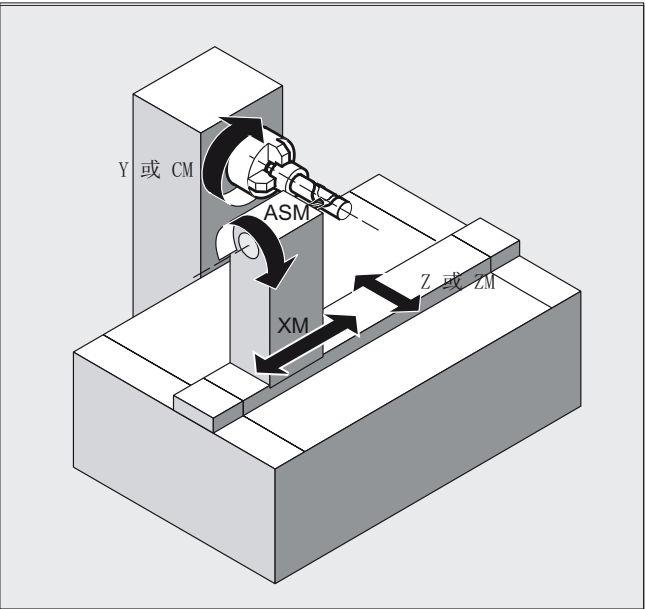
只有一根直线轴 (X) 时，只能加工出平行于柱面圆周的槽。

两根直线轴

有两根直线轴 (X、Z) 时，可在柱面上加工出任意形状的槽。

槽宽大于刀具直径时的槽面角度：

- 平行于圆周的槽 ⇒ 相互平行的槽面
- 平行于回转轴的槽 ⇒ 不相互平行的槽面



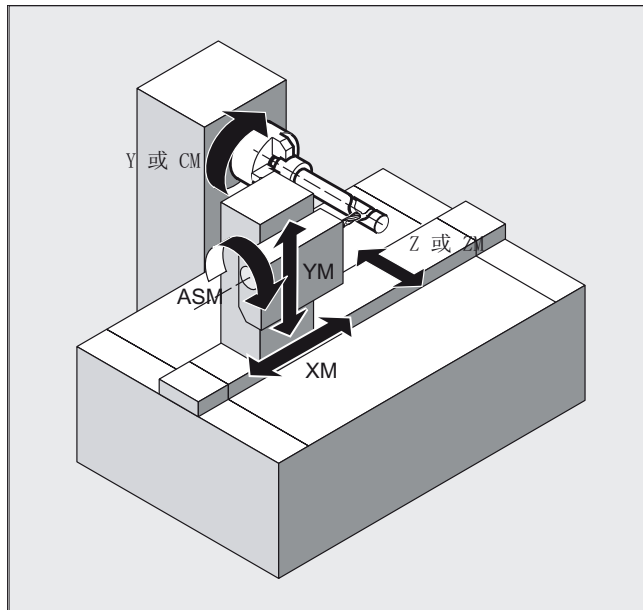
- XM 垂直于旋转中心的横向进给轴
- ZM 平行于旋转中心的直线轴
- Y / CM 坐标转换 Y 轴/回转轴
- ASM 工作主轴

图 6-8 没有槽壁补偿

三根直线轴 (轴配置 2)

有三根直线轴 (X、Y、Z) 时，可在柱面上加工出任意形状的槽。

由于 Y 轴垂直于旋转中心，因此，当槽宽大于刀具直径时，也可加工出接近平行的槽面。



- XM 垂直于旋转中心的横向进给轴
- YM 垂直于 X/Z 平面的辅助轴
- ZM 平行于旋转中心的直线轴
- Y / CM 坐标转换 Y 轴/回转轴
- ASM 工作主轴

图 6-9 有槽壁补偿

6.3.1 前提条件

每个通道可定义的最大 TRACYL 数

可通过下列机床数据定义一个通道中的坐标转换数据组：

- \$MC_TRAFO_GEOAX_ASSIGN_TAB_<n>
- \$MC_TRAFO_TYPE_<n>
- \$MC_TRAFO_AXES_IN_<n>

其中 n = 1、2、3 ... 最大坐标转换数据组数

为此，一个通道中最多允许两个坐标转换数据组用于转换 TRACYL，
\$MC_TRAFO_TYPE_<n> = 512、513 或 514。

6.3 柱面转换 TRACYL (选件)

TRACYL 的机床数据

转换 TRACYL 通过下列机床数据设置:

- \$MC_TRACYL_ROT_AX_OFFSET_<t> (回转轴偏移)
- \$MC_TRACYL_ROT_AX_FRAME_<t> (回转轴平移)
- \$MC_TRACYL_DEFAULT_MODE_<t> (TRACYL 模式选择)
- \$MC_TRACYL_ROT_SIGN_IS_PLUS_<t> (回转轴的符号)
- \$MC_TRACYL_BASE_TOOL_<t> (基本刀具的矢量)

其中 n = 1、2 ... 最大可定义 TRACYL 数

轴配置示例

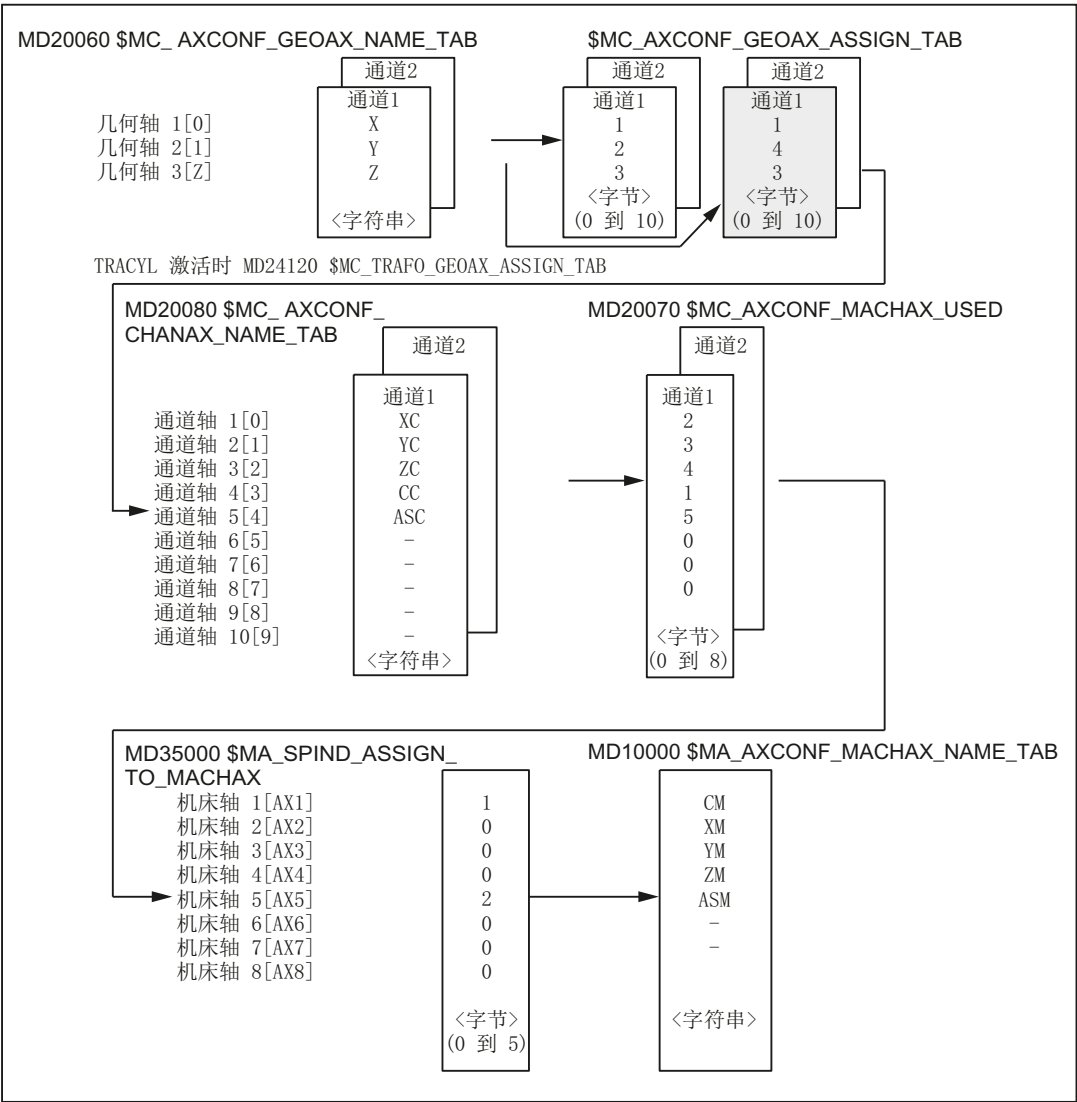


图 6-10 “柱面槽加工，“X+Y+C+Z”轴配置”一图对应的轴配置

上图中高亮显示的配置适用于“TRACYL 生效”的情形。

几何轴名称

- MD20050 \$MC_AXCONF_GEOAX_NAME_TAB[0] = "X"
- MD20050 \$MC_AXCONF_GEOAX_NAME_TAB[1] = "Y"
- MD20050 \$MC_AXCONF_GEOAX_NAME_TAB[2] = "Z"

6.3 柱面转换 TRACYL (选件)

为几何轴指定对应的通道轴

TRACYL 未生效

- MD20050 \$MC_AXCONF_GEOAX_ASSIGN_TAB[0] = 1
- MD20050 \$MC_AXCONF_GEOAX_ASSIGN_TAB[1] = 2
- MD20050 \$MC_AXCONF_GEOAX_ASSIGN_TAB[2] = 3

TRACYL 生效

- \$MC_TRAFO_GEOAX_ASSIGN_TAB_<n>[0] = 1
- \$MC_TRAFO_GEOAX_ASSIGN_TAB_<n>[1] = 4
- \$MC_TRAFO_GEOAX_ASSIGN_TAB_<n>[2] = 3

通道轴名称

MD20080 \$MC_AXCONF_CHANAX_NAME_TAB[0] = "XC"

MD20080 \$MC_AXCONF_CHANAX_NAME_TAB[1] = "YC"

MD20080 \$MC_AXCONF_CHANAX_NAME_TAB[2] = "ZC"

MD20080 \$MC_AXCONF_CHANAX_NAME_TAB[3] = "CC"

MD20080 \$MC_AXCONF_CHANAX_NAME_TAB[4] = "ASC"

为通道轴指定对应的机床轴

MD20070 \$MC_AXCONF_MACHAX_USED[0] = 2

MD20070 \$MC_AXCONF_MACHAX_USED[1] = 3

MD20070 \$MC_AXCONF_MACHAX_USED[2] = 4

MD20070 \$MC_AXCONF_MACHAX_USED[3] = 1

MD20070 \$MC_AXCONF_MACHAX_USED[4] = 5

标记主轴

MD35000 \$MA_SPIND_ASSIGN_TO_MACHAX[0] = 1 (主轴)

MD35000 \$MA_SPIND_ASSIGN_TO_MACHAX[1] = 0 (进给轴)

MD35000 \$MA_SPIND_ASSIGN_TO_MACHAX[2] = 0 (进给轴)

MD35000 \$MA_SPIND_ASSIGN_TO_MACHAX[3] = 0 (进给轴)

MD35000 \$MA_SPIND_ASSIGN_TO_MACHAX[4] = 2 (主轴)

机床轴名称

MD10000 \$MN_AXCONF_MACHAX_NAME_TAB[0] = "CM"

MD10000 \$MN_AXCONF_MACHAX_NAME_TAB[1] = "XM"

MD10000 \$MN_AXCONF_MACHAX_NAME_TAB[2] = "YM"

MD10000 \$MN_AXCONF_MACHAX_NAME_TAB[3] = "ZM"

MD10000 \$MN_AXCONF_MACHAX_NAME_TAB[4] = "ASM"

参见

柱面转换 TRACYL (选件) (页 387)

6.3.2 专用设置

设置坐标转换类型

通过坐标转换数据组中的以下专用机床数据设置坐标转换类型:

- \$MC_TRAFO_TYPE_<n> = <坐标转换类型>

用户须为坐标转换数据组指定坐标转换类型。对于 TRACYL 而言, 值 512 对应轴配置 1; 值 513 对应轴配置 2; 值 514 对应有附加直线轴的坐标转换。

激活有附加直线轴、无/有槽壁补偿的坐标转换 (参见章节“激活 (页 397)”)。

无槽壁补偿的坐标转换类型 514

柱面转换 TRAFO_TYPE_n = 514

若机床有另一根直线轴垂直于回转轴和第一根直线轴, 则可采用坐标转换类型 514, 以便在真实的 Y 轴上移动刀具补偿量。此时系统会假定, 第二根直线轴的工作区较窄, 无法执行零件程序。

之前的设置均适用于 MD10000 \$MC_TRAFO_GEOAX_ASSSIGN_TAB_n。

有槽壁补偿的槽

带槽壁补偿的 TRACYL 坐标转换已将刀具补偿考虑在内。

轴映射

下文将介绍如何设定坐标转换的轴映射。

坐标转换中的几何轴

使用 TRACYL 时，必须为坐标转换数据组指定三个（或 4 个）通道轴编号：

- MD24110 \$MC_TRAFO_AXES_IN_1[0]=垂直于回转轴的直线轴对应的通道轴编号。
- MD24110 \$MC_TRAFO_AXES_IN_1[1] = 回转轴对应的通道轴编号
- MD24110 \$MC_TRAFO_AXES_IN_1[2] = 平行于回转轴的直线轴对应的通道轴编号
- MD24110 \$MC_TRAFO_AXES_IN_1[3]=平行于柱面、垂直于回转轴的附加轴对应的通道轴编号（采用轴配置 2 的情况下）

轴编号须以通过以下机床数据定义的通道轴序列为基准：

MD24120 \$MC_TRAFO_GEOAX_ASSIGN_TAB_n

无槽壁补偿的槽

对于坐标转换类型 514，MD24110 \$MC_TRAFO_AXES_IN_n[] 的下标的含义如下。

基于基本坐标系（BCS）的下标含义：

- [0]: 垂直于回转轴的坐标轴（若存在）
- [1]: 圆柱外表面上垂直于回转轴的轴
- [2]: 平行于回转轴的坐标轴
- [3]: 平行于机床缺省设置中下标 2 的直线轴

基于机床坐标系（MCS）的下标含义：

- [0]: 垂直于回转轴的直线轴（若存在）
- [1]: 回转轴
- [2]: 平行于回转轴的直线轴
- [3]: 垂直于下标 [0] 和 [1] 的直线轴

旋转角度

垂直于回转轴的直线轴的旋转角度须如下定义：

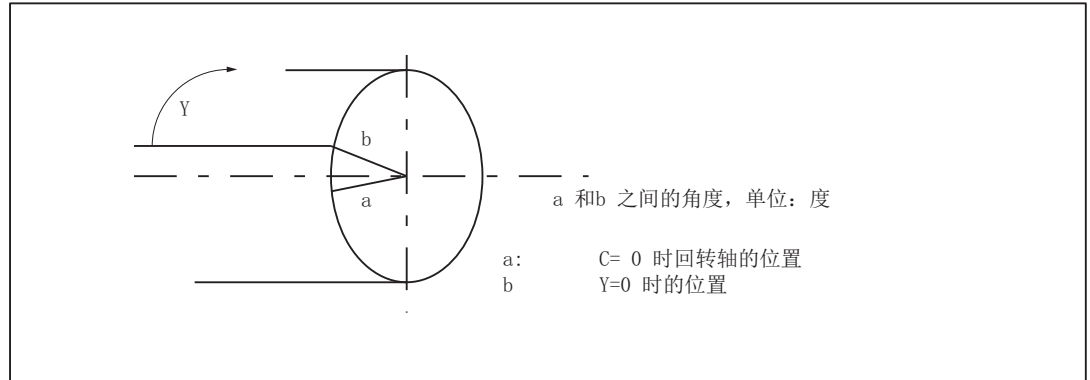


图 6-11 圆柱面中轴的旋转角度

MD24800 TRACYL_ROT_AX_OFFSET_<t>

外表面相对定义的回转轴零点的旋转角度通过以下机床数据设定：

MD24800 \$MC_TRACYL_ROT_AX_OFFSET_t=...°

其中 t 为坐标转换数据组中设定的 TRACYL 编号 (t 最大为 2)。

旋转方向

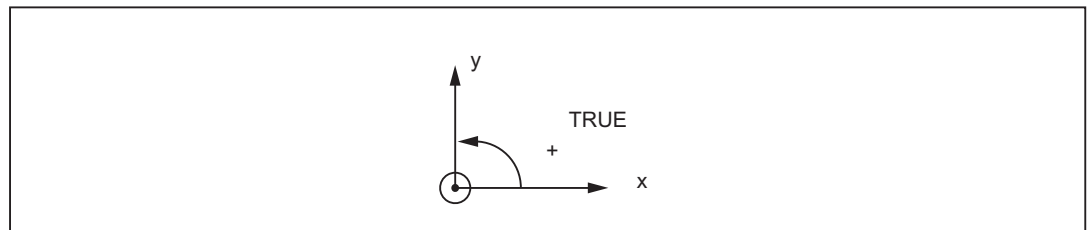
下文将介绍如何通过机床数据设定回转轴的旋转方向。

TRACYL_ROT_SIGN_IS_PLUS_<t>

面朝 z 轴观察时，若 xy

平面中回转轴为逆时针旋转，

则将此机床数据设为 TRUE，反之则设为 FALSE。



MD24810 \$MC_TRACYL_ROT_SIGN_IS_PLUS_t=TRUE

其中 t 为坐标转换数据组中设定的 TRACYL 编号 (t 最大为 2)。

6.3 柱面转换 TRACYL (选件)

可切换的几何轴

可通过机床数据设置一个 M 代码，将 GEOAX 进行的几何轴切换告知 PLC。

- MD22534 \$MC_TRAFO_CHANGE_M_CODE
切换坐标转换时 VDI 接口上输出的 M 代码。

说明

如果此机床数据的值为 0 到 6、17、30 中的某个值，则不输出 M 代码。

文档:

功能手册之基本功能分册；“坐标系、轴类型、轴配置、工件相关的实际值系统、外部零点偏移 (K2)”

刀具零点位置

下文将介绍如何通过机床数据设定刀具零点相对于直角坐标系原点的位置。

MD24820 TRACYL_BASE_TOOL_<t>

控制系统因此可获知刀具零点相对于 TRACYL 坐标系原点的位置。此机床数据有三个分量，分别对应机床坐标系的 X、Y、Z 轴。

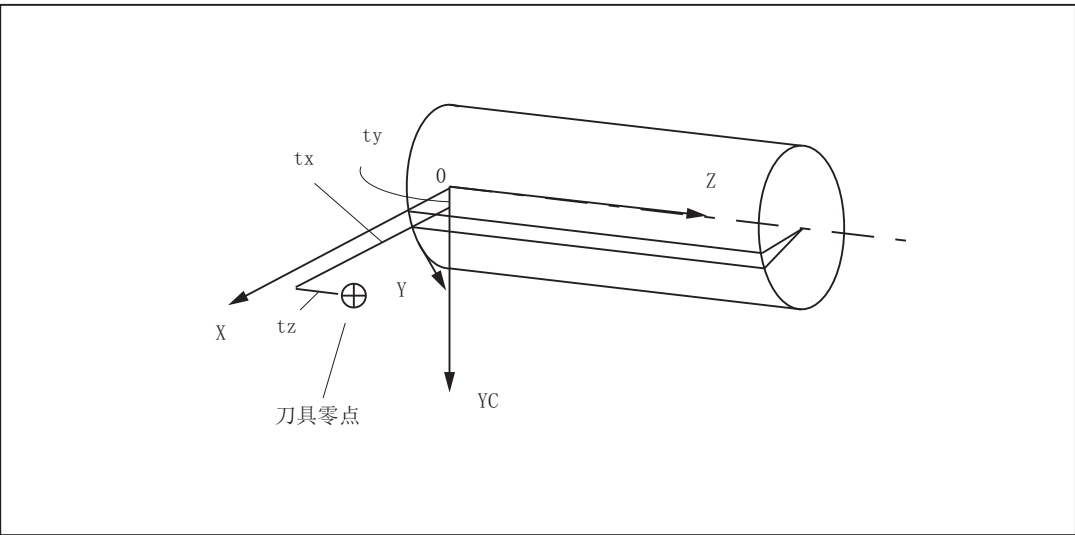


图 6-12 相对于机床零点的刀具零点位置

示例

MD24820 \$MC_TRACYL_BASE_TOOL_<t>[0] = tx

MD24820 \$MC_TRACYL_BASE_TOOL_<t>[1] = ty

MD24820 \$MC_TRACYL_BASE_TOOL_<t>[2] = tz

其中 <t> 为坐标转换数据组中设定的 TRACYL 编号

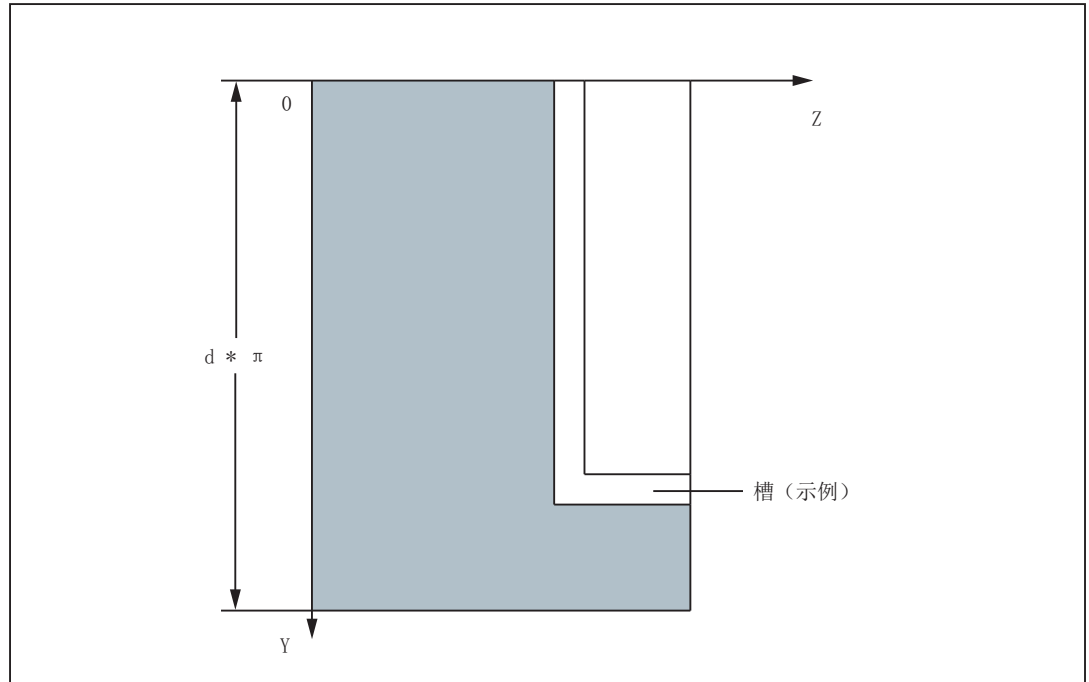


图 6-13 柱面坐标系

参见

柱面转换 TRACYL (选件) (页 387)

6.3.3 激活

转换 TRACYL 是在零件程序或同步动作中通过指令 TRACYL 激活的。

句法

TRACYL (<d>[, <n>] [, <k>])

6.3 柱面转换 TRACYL (选件)

含义

TRACYL (<d>) :	激活含 TRACYL 数据组 1 和加工直径 d 的 TRACYL 功能	
TRACYL (<d> , <n>) :	激活含 TRACYL 数据组 n 和加工直径 d 的 TRACYL 功能	
<d> :	基准直径或加工直径	
	取值范围:	>1
<n> :	TRACYL 数据组编号 (可选)	
	取值范围:	1, 2
<k> :	该参数只与转换类型 514 相关 (可选)。 • k = 0: 无槽壁补偿 • k = 1: 有槽壁补偿 如未给定参数，则机床数据 \$MC_TRACYL_DEFAULT_MODE_<n> 中设定的基本设置生效	

6.3.4 取消

一个在通道中生效的转换 TRACYL 是通过以下指令取消的:

- 取消转换: TRAFOOF
- 激活另一个转换，如: TRAANG、TRANSMIT、TRAORI

6.3.5 边界条件

选择/取消

- 不会插入中间运行程序段（倒角/倒圆）。
- 样条程序段序列必须结束。
- 确保刀具半径补偿已取消。
- TRACYL 前生效的框架由控制系统取消（与 G500 类似）。
- 作用于参与坐标转换的轴的工作区域限制被取消（与 WALIMOF 类似）。
- 连续路径运行和精磨功能中断。

- DRF 偏移必须已由操作人员删除。
- 在选择及槽壁补偿激活时，应将用于补偿的 Y 轴设为零。

换刀

只有在刀具半径补偿取消时才可以换刀。

框架

但 G91（增量尺寸）下的框架转换不作特别处理，这与未激活坐标转换时不同。待执行的增量在新框架的工件坐标系中进行计算，与前一个程序段中哪个框架生效无关。

回转轴平移可看作是通过一个框架对工件倾斜角度进行补偿或看作是回转轴偏移。

为确保回转轴的总框架能在坐标转换中生效，需进行如下设置：

`$MC_TRACYL_ROT_AX_FRAME_<t> = 1`

说明

每次选择和取消框架时都会进行轴对应关系的变更。

文档：

功能手册之基本功能分册：“坐标系、框架”（K2）”

坐标转换轴的功能限制

参与柱面转换的轴不允许用于以下功能：

- 定位轴
- 往复轴
- 预置轴
- 运行至固定点 G75
- 回参考点 G74

JOG 中的手动运行

在 JOG 模式下使用带槽壁补偿的柱面转换（`$MC_TRAFO_TYPE = 513`）时须注意，轴的运行取决于此前 AUTO 模式下的状态。因此，槽壁补偿生效时轴的运行有别于补偿未激活时的情形。这样便可在零件程序中断后重新继续执行（REPOS）。

零件程序中断

- 从 JOG 模式切换至 AUTO 模式
如果在坐标转换激活时中断零件程序处理并且在 JOG 模式中手动运行，那么在 AUTO 模式下继续执行零件程序时须注意：坐标转换也对走刀程序段中从当前位置到中断位置的部分生效。

!

警告

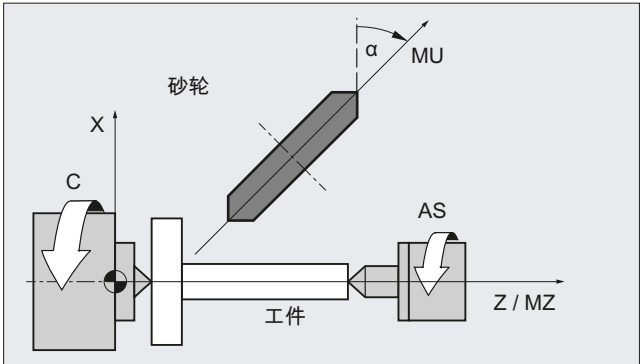
碰撞危险

此时不会执行碰撞监控。
操作人员须自行确保刀具能够顺利实现断点定位。

- 从启动到复位
若通过 RESET 终止零件程序处理，然后通过 START 重新启动，则须注意：只有在零件程序开头通过一个线性程序段（G0 或 G1）将所有轴运行至指定位置后，才能执行剩余的程序段。必要时系统会忽略 RESET 时生效的刀具（可通过机床数据定义）。

6.4 斜角转换 TRAANG (选件)

斜角转换或“斜轴”转换可实现在带有斜置机床轴的机床（如磨床）的直角工件坐标系上进行编程。



- X 几何轴
- Z 几何轴
- MZ 机床轴
- MU 机床轴
- α 轴的倾斜角

说明

转换激活时，参与转换的机床轴、通道轴和几何轴的名称须不同：

- MD10000 \$MN_AXCONF_MACHAX_NAME_TAB (机床轴名称)
 - MD20080 \$MC_AXCONF_CHANAX_NAME_TAB (通道轴名称)
 - MD20060 \$MC_AXCONF_GEOAX_NAME_TAB (几何轴名称)
-

6.4.1 前提条件

每个通道可定义的最大 TRAANG 数

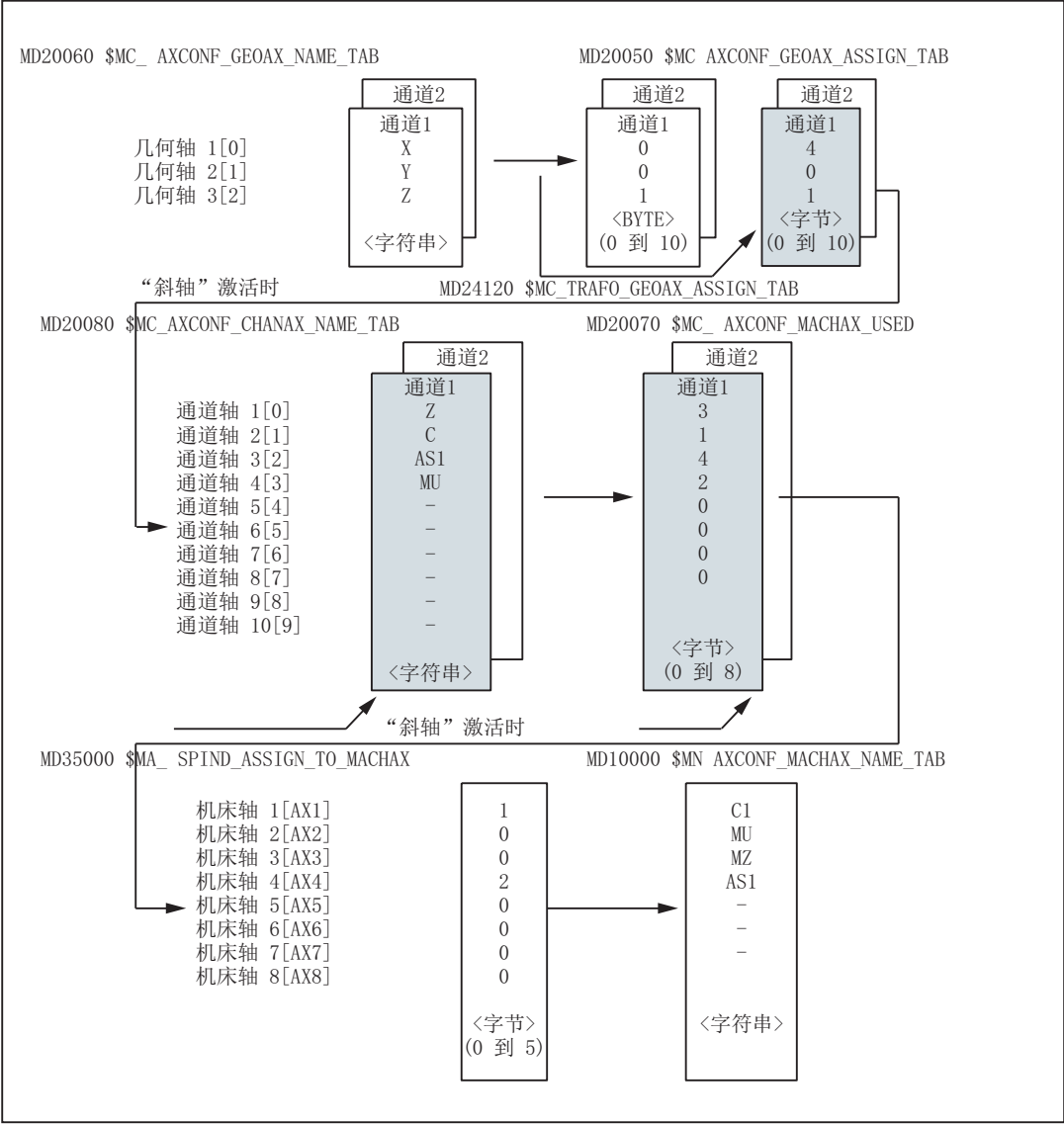
通过下列机床数据定义通道专用的坐标转换数据组：

- \$MC_TRAFO_GEOAX_ASSIGN_TAB_<n>
- \$MC_TRAFO_TYPE_<n>
- \$MC_TRAFO_AXES_IN_<n>

其中 n = 1、2、3 ... 最大坐标转换数据组数

为此，最多允许两个坐标转换数据组用于转换 TRAANG，\$MC_TRAFO_TYPE_<n> = 1024。

轴配置示例



上图中高亮显示的配置适用于“TRAANG 生效”的情形。

坐标转换中的通道轴

- MD24110 \$MC_TRAFO_AXES_IN_1[0]=4; 斜轴对应的通道轴编号
- MD24110 \$MC_TRAFO_AXES_IN_1[1]=1; 纵轴对应的通道轴编号
- MD24110 \$MC_TRAFO_AXES_IN_1[2]=0 ; ---

为几何轴指定对应的通道轴

TRAANG 生效

- MD24434 \$MC_TRAFO_GEOAX_ASSIGN_TAB_<n>[0] = 1
- MD24434 \$MC_TRAFO_GEOAX_ASSIGN_TAB_<n>[1] = 6
- MD24434 \$MC_TRAFO_GEOAX_ASSIGN_TAB_<n>[2] = 3

参见

斜角转换 TRAANG (选件) (页 400)

6.4.2 专用设置

纵轴和斜轴的夹角

- MD24700 \$MC_TRAANG_ANGLE_<m> = <角度>

其中 $-90^\circ < \text{角度} < 90^\circ$ ，不包含 0°

此角度从 X 开始顺时针方向取正值（参见“斜角转换 TRAANG (选件) (页 400)”，角度 α ）。

刀具零点的基本偏移

刀具零点的基本偏移是指在坐标转换激活时指定有效的几何轴。基本偏移中包括是否选择刀具长度补偿。编程的刀具长度补偿对基本刀具起附加作用。

- \$MC_TRAANG_BASE_TOOL_<n>[k] = <基本偏移>

其中 $k = 0、1、2$ （第 1 - 3 根几何轴）

说明

角度变化时系统不会对刀具零点进行换算。

速度控制的优化

用于优化 JOG 模式下以及定位或往复运行中的速度控制的机床数据：

速度裕量

借助以下机床数据设置纵轴的补偿运动所需的速度裕量：

6.4 斜角转换 TRAANG (选件)

MD24720 \$MC_TRAANG_PARALLEL_VELO_RES_<n> = <值>

<值>	含义
0.0	速度裕量是由 NC 根据斜轴角度及斜轴和纵轴的速度特性确定的：纵轴和垂直（虚拟）轴方向上具有相同的速度裕量。
> 0.0	速度裕量 = <值> *（纵轴的 MD32000 \$MA_MAX_AX_VELO）

加速度裕量

借助以下机床数据设置纵轴的补偿运动所需的加速度裕量：

\$MC_TRAANG_PARALLEL_ACCEL_RES_<n> = <值>

<值>	含义
0.0	加速度裕量是由 NC 根据斜轴角度及斜轴和纵轴的加速度特性确定的：纵轴和垂直（虚拟）轴方向上具有相同的加速度裕量。
> 0.0	加速度裕量 = <值> *（纵轴的 MD32300 \$MA_MAX_AX_ACCEL）

转换切换时的 M 代码

切换转换时系统会输出指定的 M 代码：

- MD22534 \$MC_TRAFO_CHANGE_M_CODE = <值>

 警告
不输出 M 代码。 不向 PLC 输出值 0 - 6、17、30。

6.4.3 激活

转换 TRAANG 是在零件程序或同步动作中通过指令 TRAANG 激活的。

句法

TRAANG
TRAANG ([<a>] [, <n>])

含义

TRAANG:	激活带有数据组 1 和最后生效的角度 α 的 TRAANG 功能	
TRAANG (, <n>):	激活带有数据组 n 和最后生效的角度 α 的 TRAANG 功能	
TRAANG (< α >):	激活带有数据组 1 和角度 α 的 TRAANG 功能	
TRAANG (< α >, <n>):	激活带有数据组 n 和角度 α 的 TRAANG 功能	
< α >:	斜置轴的角度 (可选)	
	提示 未给定角度时适用机床数据中设置的值: \$MC_TRAANG_ANGLE_<n>, n = 数据组编号	
<n>:	取值范围:	$-90^{\circ} < \alpha < +90^{\circ}$
	数据组编号 (可选)	
<n>:	取值范围:	1, 2
	数据组编号 (可选)	

6.4.4 取消

一个在通道中生效的转换 TRAANG 是通过以下指令取消的:

- 取消转换 TRAFOOF
- 激活另一个转换, 如: TRACYL、TRANSMIT、TRAORI

6.4.5 边界条件

可通过零件程序或 MDA 选择及取消坐标转换。

选择和取消

- 不会插入中间运行程序段 (倒角/倒圆)。
- 样条程序段序列必须结束。
- 确保刀具半径补偿已取消。
- 当前生效的框架由控制系统取消 (相当于编写 G500)。
- 作用于参与坐标转换的轴的工作区域限制被取消 (相当于编写了 WALIMOF)。
- 激活的刀具半径补偿由控制系统传输到坐标转换。

6.4 斜角转换 TRAANG (选件)

- 连续路径运行和精磨功能中断。
- DRF 偏移必须已由操作人员删除。
- 机床数据 MD24110 \$MC_TRAFO_AXES_IN_n 中指定的所有轴必须基于程序段实现同步（例如：无包含 POSA... 的运行指令）。

换刀

只有在刀具半径补偿取消时才允许换刀。

框架

所有仅以工件坐标系为基准的指令均可使用（FRAME，刀具半径补偿）。但 G91（增量尺寸）下的框架转换不作特别处理，这与未激活坐标转换时不同。待执行的增量在新框架的工件坐标系中进行计算，与前一个程序段中哪个框架生效无关。

扩展

选择和取消 TRAANG 时，几何轴和通道轴间的对应关系可能会变化。用户可将这些几何轮廓分量附加到轴框架上，例如：斜轴相对 x/z 平面的平移、旋转、比例和镜像。

关于坐标转换生效条件下这些框架补偿的更多相关信息请见：

文档：

功能手册之基本功能分册：“轴、坐标系,框架(K2)”

特例情形

参与坐标转换的轴不可：

- 用作预置轴（报警），
- 运行到固定点（报警）
- 回参考点（报警）。

速度控制

TRAANG 下的速度监控默认在预处理中执行。

下列情形下监控和限制在主处理中执行：

- AUTO 模式下，编写的定位轴或往复轴参与了坐标转换。
- 切换至 JOG 模式时

参与坐标转换的轴作为轨迹轴运行时，监控功能将从主处理回到预处理阶段执行（例如从 JOG 切换至 AUTO 模式时）。

在预处理中进行的速度监控充分使用机床最大动态响应，比在主处理中更佳。

进行倾斜加工的机床也是如此。

6.4.6 编程 (G05, G07)

功能

下列功能可供使用：

- 以直角坐标系编写和显示位置
- 刀具补偿和零点偏移的直角坐标计算
- 在 NC 程序中编写斜轴的角度
- 运行至斜向切入的起点 (G07)
- 斜向切入 (G05)

编程

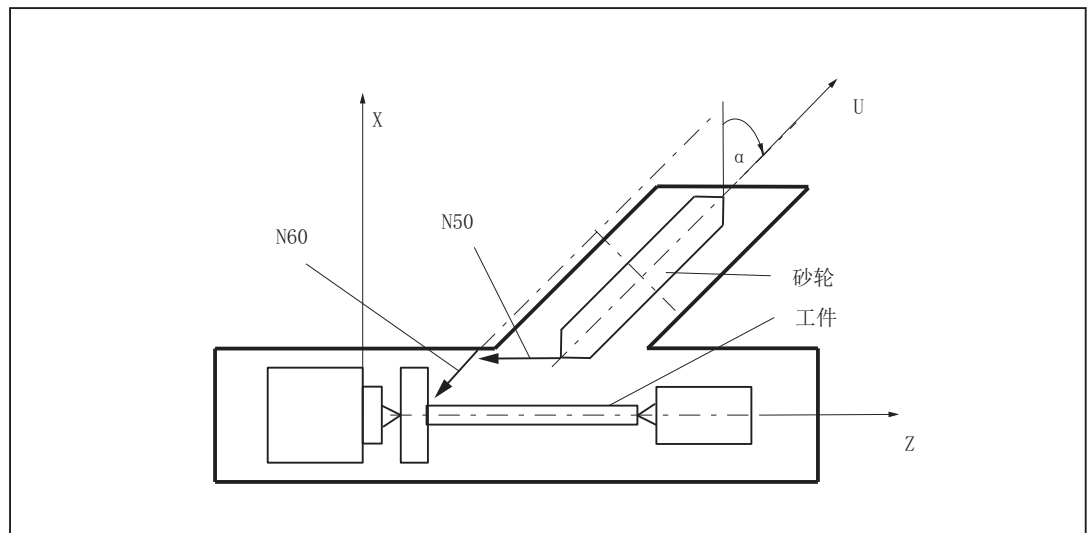


图 6-14 进给轴倾斜一定角度的机床

示例：

程序代码	注释
N...	; 编写斜轴的角度
N50 G07 X70 Z40 F4000	; 运行至起始位置
N60 G05 X70 F100	; 斜向切入
N...	

边界条件

- 仅当 TRAANG 生效时，在 JOG 模式下选择“坐标 PTP 运动”功能（等同于 G05）才有意义。 必须注意 MD20140 \$MC_TRAFO_RESET_VALUE 中的设置。
- “PTP 运动”未生效时，必须在 JOG 模式下以直角坐标退回 REPOS 平移量。
- JOG 模式下“PTP 运动”生效时会监控工作区域。 轴进入限制区域前，系统会对其进行制动。 如果“PTP 运动”未生效，那么轴可精确运行到限制区域边界处。

参见

坐标 PTP 运动 (页 420)

6.5 坐标转换级联

引言

本文介绍的各种坐标转换可以和“斜轴”类型的转换互联在一起：

- TRANSMIT
- TRACYL
- TRAANG（斜轴）

以及

文档：

功能手册之特殊功能分册；“3 轴转换到 5 轴转换（F2）”

中介绍的

- 定向转换
- 万向铣头

应用

下面是从坐标转换级联的丰富应用中选出的几种：

- 使用一个斜置砂轮磨削轮廓，该轮廓位于圆柱体外表面上（TRACYL），例如模具磨削。
- 使用斜置砂轮对一个通过 TRANSMIT 生成的非圆轮廓进行精磨。

说明

下文说明的坐标转换生效时机床轴名称、通道轴名称和几何轴名称须不同。参见机床数据：

MD10000:AXCONF_MACHAX_NAME_TAB

MD20080:AXCONF_CHANAX_NAME_TAB

MD20060:AXCONF_GEOAX_NAME_TAB

否则无法保证轴的唯一性。

轴配置

坐标转换级联需要采用下列配置措施：

- 命名几何轴
- 命名通道轴
- 为几何轴指定对应的通道轴
 - 一般情况（无坐标转换生效）
- 为通道轴指定对应的机床轴
- 标记主轴、旋转、轴模数
- 命名机床轴
- 坐标转换特定名称（每个坐标转换和每个坐标转换级联各有）
 - 坐标转换的类型
 - 参与坐标转换的轴
 - 坐标转换生效时为几何轴指定对应的通道轴
 - 根据具体坐标转换，还需定义坐标系旋转角度、旋转方向、刀具零点相对于坐标系原点的位置、斜轴角度等

坐标转换的数量

在系统中可为每个通道定义最多十个坐标转换数据组。这些坐标转换的机床数据名称以“\$MC_TRAFO_...”开头，以“..._n”结尾，其中 n 为数字 1 至 10：

坐标转换级联数量

一个通道的最多 10 个坐标转换中，最多允许定义两个坐标转换级联。

坐标转换的顺序

配置机床数据时，设定坐标转换级联前必须先进行级联包含的单个坐标转换的设定。

级联顺序

在坐标转换级联中，第二个必须为“斜轴”（TRAANG）。

级联方向

坐标转换级联中的第一个转换的输入为 BCS，第二个转换的输出为“斜轴”。

边界条件

坐标转换级联同样需要注意各个转换本身的边界条件和特殊情形。

6.5.1 激活坐标转换级联

TRACON

坐标转换级联通过以下指令激活：

TRACON(trf, par)

- **trf:**
坐标转换级联的编号：0 或者 1 表示第一个/唯一的级联。如果该位置上没有编写任何内容，则含义等同于赋值 0 或 1，即激活第一个/唯一的级联。2 表示第二个级联。（值不等于 0—2 时会触发故障报警）。
- **par**
为一个或者多个通过逗号分隔开的参数，级联中的转换需要该参数，例如斜轴的角度。如果未设置该参数，那么缺省设置或者上一次所使用的参数将生效。在设置逗号时要注意按正确的顺序输入参数，确保某些参数使用缺省设置。特别是在设置不止一个参数时，即便不必设定 **trf**，该参数前也必须设置逗号，例如 **TRACON(, 3.7)**。

TRACON() 会触发之前生效的另一个坐标转换自动取消。

6.5.2 取消坐标转换级联

TRAFOOF

和其他转换一样，坐标转换级联通过 TRAFOOF 取消。

6.5.3 坐标转换级联的特殊性

刀具数据

刀具总是被指定给级联中的第一个转换。后续转换中的有效刀具长度似乎为零。只有通过机床数据设置的刀具基本长度（_BASE_TOOL_）才会对级联中的**第一个**转换有效。

示例

单个坐标转换及由其构成的级联的配置示例参见“坐标转换级联”章节。

参见

坐标转换级联 (页 408)

6.5.4 固定生效的坐标转换

功能

固定生效的坐标转换总是保持生效状态。选择的其他坐标转换会被作为“第一个转换”和固定生效的坐标转换级联在一起。

选择的其他坐标转换（如 TRANSMIT 等）必须通过 TRACON 与固定生效的坐标转换构成级联。此时在零件程序中不编写 TRACON，而是级联中的第一个转换。

编程相关的更多提示请见

文档：

编程手册之工作准备分册：“坐标转换级联”

选择和取消

固定生效的坐标转换通过以下机床数据选择：

MD20144 \$MC_TRAFO_MODE_MASK, 位 0 = 1

MD20144 \$MC_TRAFO_RESET_VALUE 用于定义固定生效的坐标转换。

MD20140 \$MC_TRAFO_RESET_VALUE=固定生效的坐标转换的数据组编号

此外还须设置（即需要注意）：

MD20110 \$MC_RESET_MODE_MASK

位 0 = 1（计算位 7）

位 7 =0（MD20140 \$MC_TRAFO_RESET_VALUE 确定坐标转换数据组）

MD20112 \$MC_START_MODE_MASK（MD20140 \$MC_TRAFO_RESET_VALUE）

MD20118 \$MC_GEOAX_CHANGE_RESET= TRUE（即几何轴复位）

若这些附加参数设置不当，则会

发出报警 14404。

TRAFOOF 指令会取消生效的 TRACON，并自动选择固定生效的坐标转换。

对 HMI 操作的影响

由于选择了固定生效的坐标转换后总是有转换生效，因此 HMI 操作界面上转换的选择和取消方式经过调整：

HMI 上的 TRACON

HMI 操作界面上不再显示 TRACON，而是显示 TRACON 级联的第 1 转换，例如 TRANSMIT。相应的，第 1 转换的类型通过系统变量 \$P_TRAFO 及 \$AC_TRAFO 输出。TRANSMIT 中写入的循环可直接使用。

HMI 上的 TRAFOOF

HMI 操作界面上，编程指令 TRAFOOF 处不显示 G 代码列表中的转换。相应的，系统变量 \$P_TRAFO 及 \$AC_TRAFO 输出 0 值。固定生效的坐标转换生效，BCS 和 MCS 坐标系不再相同。显示的 MCS 位置总是针对真实机床轴。

系统变量

生效的坐标转换级联的类型由新系统变量输出。

描述	NCK 变量
无坐标转换生效：0 一个坐标转换生效：TRACON 级联中第 1 转换的类型，或者 TRACON 未激活情况下生效的转换类型	\$P_TRAFO_CHAIN[0]
无坐标转换生效：0 一个坐标转换生效：TRACON 级联中第 2 转换的类型	\$P_TRAFO_CHAIN[1] \$AC_TRAFO_CHAIN[1]
级联了超过 2 个转换时才使用。 这些变量当前总是输出 0。	\$P_TRAFO_CHAIN[2] \$AC_TRAFO_CHAIN[2] 和 \$P_TRAFO_CHAIN[3] \$AC_TRAFO_CHAIN[3]

固定生效的坐标转换同样显示：

\$P_TRAFO_CHAIN[0]，\$AC_TRAFO_CHAI[0]

借此可在零件程序或循环中可靠地显示生效的坐标转换。

TRACON 和其他转换间的区别：

无坐标转换生效时查询 \$P_TRAFO、\$AC_TRAFO，不为零时查询 \$P_TRAFO_CHAIN[1]、\$AC_TRAFO_CHAI[1]。

框架

选择和取消 TRACON 时，框架会假定只存在级联的第 1 转换。选择 TRAANG 时不会保留虚拟轴的转换。

JOG

在 JOG 模式下运行时，固定生效的坐标转换保持生效。

边界条件

固定生效的坐标转换不会改变 NCK 中的基本运行顺序。转换生效时存在的所有限制将继续生效。

RESET 仍然会完全取消当前生效的转换，因此固定生效的坐标转换将重新选择。故障状态下不会执行重新选择。此时会通过报警指示故障。

将 TRAANG 选为固定生效的坐标转换时，可能会触发报警 14401 或 14404。固定生效的坐标转换生效时，转换类型将决定故障情形下能否触发其他转换报警。

回参考点时会自动取消转换，之后需要执行 RESET 或 START，以重新选择固定生效的坐标转换。

示例

在配备了附加的斜置 Y 轴的车床上，斜轴转换功能属于机床配置，无需编程人员重新编写。在该车床上，TRACYL 或 TRANSMIT 选中的是内含 TRAANG 的转换。取消编写的转换时，TRAANG 自动重新生效。HMI 操作界面中对应显示 TRACYL 或 TRANSMIT。

机床数据，用于配备斜置 Y1 轴的车床，该轴相对 X1 斜置，同时与 Z1 相垂直。

CANDATA (1)

; 无转换

MD20080 \$MC_AXCONF_CHANAX_NAME_TAB[1] = "Y2"

MD20050 \$MC_AXCONF_GEOAX_ASSIGN_TAB[0]=1

MD20050 \$MC_AXCONF_GEOAX_ASSIGN_TAB[1] = 0

MD20050 \$MC_AXCONF_GEOAX_ASSIGN_TAB[2]=3

; TRAANG 的数据

MD24100 \$MC_TRAFO_TYP_1 = 1024; TRAANG, Y1 轴相对 X1 斜置，垂直于 Z1

MD24110 \$MC_TRAFO_AXES_IN_1[0]=2

MD24110 \$MC_TRAFO_AXES_IN_1[1]=1

MD24110 \$MC_TRAFO_AXES_IN_1[2]=3

MD24110 \$MC_TRAFO_AXES_IN_1[3] = 0

MD24110 \$MC_TRAFO_AXES_IN_1[4] = 0

MD24120 \$MC_TRAFO_GEOAX_ASSIGN_TAB_1[0]=1

MD24120 \$MC_TRAFO_GEOAX_ASSIGN_TAB_1[1]=2

MD24120 \$MC_TRAFO_GEOAX_ASSIGN_TAB_1[2]=3

MD24700 \$MC_TRAANG_ANGLE_1 = 60

MD24720 \$MC_TRAANG_PARALLEL_VELO_RES_1 = 0,2

MD24721 \$MC_TRAANG_PARALLEL_ACCEL_RES_1 = 0,2

; 定义固定生效的坐标转换

MD20144 \$MC_TRAFO_MODE_MASK = 1

MD20140 \$MC_TRAFO_RESET_VALVUE= 1

MD20110 \$MC_RESET_MODE_MASK = 'H01'

MD20112 \$MC_START_MODE_MASK = 'H80'

MD20140 \$MC_TRAFO_RESET_VALUE

MD20118 \$MC_GEOAX_CHANGE_RESET= TRUE

; TRANSMIT、TRACYL 的数据

MD24911 \$MC_TRANSMIT_POLE_SIDE_FIX_1 = 1 ; 也可为 2, 触发报警 21617

MD24200 \$MC_TRAFO_TYP_2 = 257

MD24210 \$MC_TRAFO_AXES_IN_2[0] = 1

MD24210 \$MC_TRAFO_AXES_IN_2[1] = 4

MD24210 \$MC_TRAFO_AXES_IN_2[2] = 3

MD24210 \$MC_TRAFO_AXES_IN_2[3] = 0

MD24210 \$MC_TRAFO_AXES_IN_2[4] = 0

MD24220 \$MC_TRAFO_GEOAX_ASSIGN_TAB_2[0] =1

MD24220 \$MC_TRAFO_GEOAX_ASSIGN_TAB_2[1] =4

MD24220 \$MC_TRAFO_GEOAX_ASSIGN_TAB_2[2] =3

MD24300 \$MC_TRAFO_TYP_3 = 514

MD24310 \$MC_TRAFO_AXES_IN_3[0] = 1

MD24310 \$MC_TRAFO_AXES_IN_3[1] = 4

MD24310 \$MC_TRAFO_AXES_IN_3[2] = 3

MD24310 \$MC_TRAFO_AXES_IN_3[3] = 0

MD24310 \$MC_TRAFO_AXES_IN_3[4] = 0

MD24320 \$MC_TRAFO_GEOAX_ASSIGN_TAB_3[0] =1

MD24320 \$MC_TRAFO_GEOAX_ASSIGN_TAB_3[1] =4

MD24320 \$MC_TRAFO_GEOAX_ASSIGN_TAB_3[2] =3

; TRACON 的数据

; TRACON 级联 TRANSMIT 514/TRAANG (Y1 轴相对 X1 斜置)

MD24400 \$MC_TRAFO_TYP_4 = 8192

MD24995 \$MC_TRACON_CHAIN_1[0] = 3

MD24995 \$MC_TRACON_CHAIN_1[1] = 1

MD24995 \$MC_TRACON_CHAIN_1[2] = 0

MD24420 \$MC_TRAFO_GEOAX_ASSIGN_TAB_4[0] = 1

MD24420 \$MC_TRAFO_GEOAX_ASSIGN_TAB_4[1] = 4

MD24420 \$MC_TRAFO_GEOAX_ASSIGN_TAB_4[2] = 3

; TRACON 级联 TRANSMIT 257/TRAANG (Y1 轴相对 X1 斜置)

MD24430 \$MC_TRAFO_TYP_5 = 8192

MD24996 \$MC-TRACON_CHAIN_2[0] = 2

MD24996 \$MC-TRACON_CHAIN_2[1] = 1

MD24996 \$MC_TRACON_CHAIN_2[2] = 0

MD24434 \$MC_TRAFO_GEOAX_ASSIGN_TAB_5[0] = 1

MD24434 \$MC_TRAFO_GEOAX_ASSIGN_TAB_5[1] = 4

MD24434 \$MC_TRAFO_GEOAX_ASSIGN_TAB_5[2] = 3

M17

; 对应的零件程序:

\$TC_DP1[1,1]=120 ; 刀具类型

\$TC_DP2[1,1]=0

\$TC_DP3[1,1]=3 ; 长度补偿矢量

\$TC_DP4[1,1]=25

\$TC_DP5[1,1]=5

\$TC_DP6[1,1]=2 ; 半径; 刀具半径

; 切换坐标转换:

N1000 G0 X0 Y=0 Z0 A80 G603 SOFT G64

N1010 N1020 X10 Y20 Z30 ; TRAANG(,1) 非必需设定, 因为会自动选择

N1110 TRANSMIT(1) N1120 X10 Y20 Z30 N1130 Y2=0 ; TRACON(2) 非必需设定, 因为会自动执行

N1210 TRAFOOF ; TRAANG(,1) 非必需设定，因为会自动执行

N1220 X10 Y20 Z30

M30

6.5.5 坐标转换级联中的轴位置

功能

下列系统变量可用于带系统坐标转换或 OEM 坐标转换的机床，特别可用于坐标转换级联（TRACON）：

类型	系统变量	含义
实数	\$AA_ITR[ax,n]	第 n 个转换输出处的当前设定值
实数	\$AA_IBC[ax]	坐标轴的当前设定值
实数	\$VA_ITR[ax,n]	第 n 个转换输出处的当前实际值
实数	\$VA_IBC[ax]	轴的当前坐标 BCS 编码器位置
实数	\$VA_IW[ax]	轴的当前 WCS 实际值
实数	\$VA_IB[ax]	轴的当前 BCS 编码器位置

注意以下控制系统上电/复位时特性：

- 上电
轴未执行回参考点时，编码器位置值为 0。\$VA 变量中输出的编码器实际值经过转换。
- 复位
复位会使生效的转换有所变化，从而对系统变量的值产生直接影响。当前生效的转换若在复位后继续生效，复位时该转换会短暂取消，然后重新启用。这会对位置变量产生直接影响。变量值可能出现阶跃。
通过以下变量：
\$AC_STAT == 0
可在同步动作中查询此状态。

\$AA_ITR[<轴>, <转换层>]

\$AA_ITR[ax,n] 变量输出级联的第 n 个转换输出处的轴设定位置。

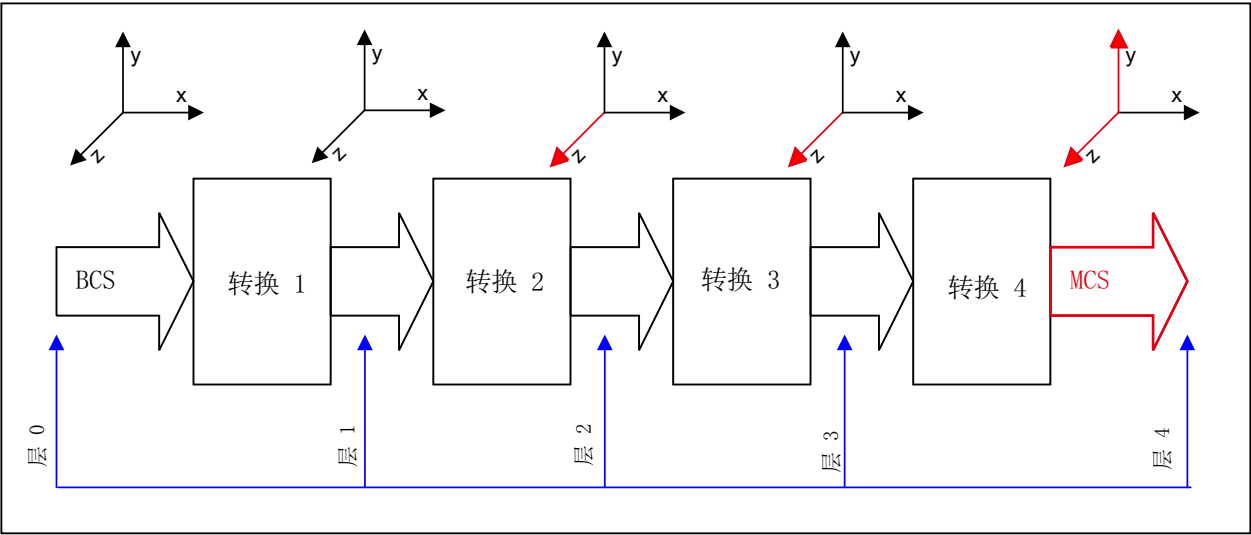


图 6-15 转换层

转换层

此变量的第 2 个下标对应转换层，即获取位置的层：

- 转换层 0：位置为 BCS 位置，即：
\$AA_ITR[x,0] == \$AA_IB[x]
- 转换层 1：第 1 转换输出处的设定位置
- 转换层 2：第 2 转换输出处的设定位置
- 转换层 3：第 3 转换输出处的设定位置
- 转换层 4：第 4 转换输出处的设定位置，即：
\$AA_ITR[x,4] == \$AA_IM[x]

若少一个或少多个转换层，那么最高层将输出相同的值。例如缺少转换层 3 和 4，那么：
\$AA_ITR[x,2] = \$AA_ITR[x,3] = \$AA_ITR[x,4] = \$AA_IM[x]

通过 TRAFOOF 取消转换或系统复位时，转换层 0 到 4 合并，此时变量总是输出 BCS 值（层 0）。

轴

变量的第 1 个下标可为几何轴、通道轴或机床轴名称。编写几何轴名称时，每个转换层中都会启用第 0 层中通道轴和几何轴的对应关系。也就是说，BCS 侧的几何轴对应关系在所

有层中生效。只有在几何轴未经过切换时，才建议使用几何轴名称。否则最好使用通道轴名称。

\$AA_IBC[<轴>]

\$AA_IBC[ax] 变量用于输出坐标轴在 BCS 和 MCS 间的设定位置。若第 n 个转换输出处的轴为坐标轴，则会输出此值。若所有转换输出处的对应轴均非坐标轴，则会输出轴的 BCS 值（包含所有 BCS 补偿）。

若 TRACON 涉及一根坐标轴，则会输出其 MCS 值。轴名称可使用几何轴、通道轴或机床轴的名称。

\$VA_ITR[<轴>, <转换层>]

\$VA_ITR[ax,n] 变量用于输出级联的第 n 个转换输出处的轴的编码器位置

\$VA_IBC[<轴>]

\$VA_IBC[ax] 变量用于输出坐标轴在 BCS 和 MCS 间的编码器位置。轴名称可使用几何轴、通道轴或机床轴的名称。

若第 n 个转换输出处的轴为坐标轴，则会输出此值。若所有转换输出处的对应轴均非坐标轴，则会输出轴的 BCS 值。

\$VA_IW[<轴>]

\$VA_IW[ax] 变量用于输出转换回 WCS 值的轴编码器位置。WCS 值包含所有轴叠加分量（DRF、AA_OFF、外部零点偏移等）和补偿值（CEC 等）。

\$VA_IB[<轴>]

\$VA_IB[ax] 变量用于输出转换回 BCS 值的轴编码器位置。BCS 值包含所有轴叠加分量（DRF、AA_OFF、外部零点偏移等）和补偿值（CEC 等）。

说明

\$VA_ITR\$, VA_IBC, \$VA_IW, \$VA_IB

在一个插补周期内读取变量时，变量值不会改变，即使实际值可能已发生变化。

转换生效时须注意，在插补周期内将实际值转换至 BCS 可能会非常耗时。此时必须设置足够长的插补周期。

6.6 坐标 PTP 运动

功能

使用此功能可通过同步轴运动逼近某个坐标位置。

此功能例如可用于以下情形：关节位置切换导致运行穿过奇点。

过奇点运行会引起进给速度降低或轴过载。

说明

MD24100 \$MC_TRAFO_TYPE_1 中必须输入 TE4 中说明的坐标转换类型。

该功能只能和某个坐标转换功能组合使用。此外，该功能只支持指令 G0 和 G1。否则会发出报警 14144“无法执行 PTP 运动”。

PTP 生效时，参与坐标转换的轴（例如通过 POS 运行的轴）不允许同时为定位轴。此情形会被系统拒绝，并触发报警 17610。

激活

此功能通过编写 PTP 指令来激活。

编写 CP 指令时此功能会被取消。这两个指令均属于 G 功能组 49。

- PTP 指令：通过同步轴运行逼近编写的直角坐标（PTP=point to point，点到点）。
- CP 指令：通过轨迹运行逼近编写的直角坐标（缺省设置，CP=continuous path，连续轨迹）。
- PTPG0 指令：在每个 G0 程序中自动执行编写的 PTP 运动。之后重新设置为 CP。

上电

缺省设置下，上电后坐标转换运动设为 CP 方式。可通过 MD20152 \$MC_GCODE_RESET_VALUES[48] 将缺省设置修改为 PTP 方式。

复位

通过 MD20152 \$MC_GCODE_RESET_MODE[48] (G 功能组 49) 定义在复位/零件程序结束后启用哪种设置。

- MD=0: 设置符合以下机床数据
MD20150 \$MC_GCODE_RESET_VALUES[48]
- MD=1: 保留当前设置

选择

MD20152 \$MC_GCODE_RESET_MODE[48] =0 时, 可通过 MD20150 \$MC_GCODE_RESET_VALUES[48] 激活以下功能:

- MD=2:
PTP, 和之前一样
- MD=3:
PTPG0, 仅在 G0 程序段中自动执行 PTP, 之后切换回 CP

边界条件

请注意下列刀具运动和碰撞相关信息:

- PTP 指令产生的刀具运动和 CP 指令有很大差异, 因此须对独立于生效坐标转换写入的现有子程序进行调整, 确保 TRANSMIT 下无碰撞风险。特别是使用 PTPG0 指令时。
- 在 TRANSMIT 和 PTP 激活时, 机床轴原则上以最短行程运行。微小的程序段终点移动可能会导致回转轴在旋转指令为 + 179.99° 的情况下实际旋转 -179.99°, 尽管程序段终点几乎未改变。

不支持与下列 NC 功能的组合使用:

- PTP 不能和刀具半径补偿 (TRC) 组合使用。
G0 和 G41 本质上并不冲突。但 PTP 生效时所产生的轮廓与 TRC 计算出的轮廓有差别, 并会触发 TRC 报警信息。
- PTPG0 和刀具半径补偿 (TRC) 组合使用时 CP 生效。
由于 G0 和 G41 并不冲突, 因此刀具半径补偿生效时会自动切换至 CP。这样一来半径补偿便可作用于明确定义的轮廓。
- PTP 不能和平滑逼近和回退 (SAR) 组合使用。
SAR 需要一个轮廓用于设计逼近和回退运动。PTP 则无法提供此信息。

- PTPG0 和平滑逼近和回退 (SAR) 组合使用时 CP 激活
SAR 需要一个轮廓用于设计逼近和回退运动, 以及用于切向降低或提升。CP 可为该指令提供所需的程序段。但是 G0 程序段到真正的逼近轮廓之间采用 PTP, 因此更快。回退时也采用相同的方案。
- PTP 和切削循环 (如 CONTPRON、CONTDCON) 不能组合使用
切削循环需要一个轮廓用于设计切削分段。PTP 则无法提供此信息。此时会发出报警 10931“切削补偿错误”。
- PTPG0 和切削循环 (如 CONTPRON、CONTDCON) 组合使用时, CP 激活 切削循环需要一个轮廓用于设计切削分段。CP 可为该指令提供所需的程序段。
- 倒角和倒圆会被忽略。
- PTP 时插补中的轴叠加不允许改变。这例如涉及到快速提刀 LIFTFAST、刀具精补偿、跟随运行 TRAILON 以及切向跟踪 TANGON 功能。

在 PTP 程序段中

- 压缩器会自动取消, 因为其与 PTP 不兼容。
- 写入的 G643 会自动转换至 G642。
- 参与坐标转换的轴不可同时作为定位轴。

特殊性

基准坐标系方面须注意以下特殊性:

- 精磨 G642 总是以机床坐标系为准, 而不是像通常情况下以基本坐标系为准。
- G641 依据由机床轴坐标计算出的虚拟轨迹来确定精磨。
- G1 下的 F 值设定基于由机床轴坐标计算出的虚拟轨迹。

程序段搜索

运行一段包含程序段搜索的程序时, 程序段搜索时的 TRANSMIT 可能会在相同的直角坐标下产生不同的机床轴位置。

报警

有可能导致冲突的动作会被系统拒绝, 并触发下列报警:

报警 14144, 在 PTP 运动中选择或激活了 TRC 时触发。同样在 PTP 运动中选择或激活了平滑逼近和回退 (SAR) 或 PTP 中无所需 G0 和 G1 程序段时, 也会触发此报警。

报警 10753: 在 PTPG0 运动中激活了 TRC 时系统切换至 CP, 从而能正确地进行刀具补偿。

报警 10754: 有冲突。

报警 10778: 有冲突。

报警 10744: 在 PTPG0 运动中激活了 SAR 时系统切换至 CP, 从而能正确地执行平滑逼近和回退。

报警 10746: 有冲突。

报警 17610: 参与坐标转换的轴不允许同时为通过 POS 运行的定位轴。

说明

更多编程相关信息及示例请见:

文档:

编程手册之工作准备分册; 章节“坐标转换”。“坐标 PTP 运动”

6.6.1 编写位置

通常情况下, 光指定直角坐标和刀具方向不足以确定唯一的机床位置。根据具体运动系统, 可存在最多 8 个不同的关节位置。这些不同的关节位置针对特定坐标转换。

STAT 地址

一个直角坐标必须能够换算为唯一的轴角度。为此必须在 STAT 地址下输入关节位置。

STAT 的一个位对应一个可能的位置, 采用二进制值。位的含义取决于相应的坐标转换。

对于印刷手册“Handling 坐标转换软件包 (TE4)”中介绍的坐标转换, 这些位对应不同的位置, 如上图所示。

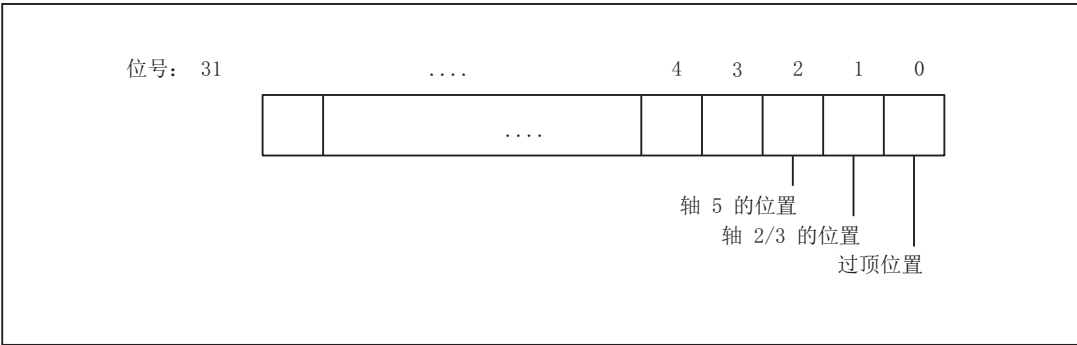


图 6-16 “Handling 坐标转换软件包” 的位

说明

STAT 地址只允许和 PTP 指令组合使用，因为通常坐标转换生效时无法进行位置切换。使用 CP 指令时，系统将起点用作终点。

6.6.2 轴角度的叠加范围

TU 地址

为了确定唯一的大于 $\pm 180^\circ$ 轴角度，必须在 TU (Turn) 地址下编写此信息。此时 TU 地址会一同显示轴角度的符号。这样便可确定 $|\theta| < 360^\circ$ 内唯一的轴角度。

TU 变量的每个位分别对应参与坐标转换的每根轴的运行方向。

- TU 位 = 0: $0^\circ \leq \theta < 360^\circ$
- TU 位 = 1: $360^\circ < \theta < 0^\circ$

如果是直线轴，该 TU 位置 0。

运行范围 $> \pm 360^\circ$ 的轴总是会沿最短轨迹运行，因为无法通过 TU 信息确定唯一的轴位置。

若在一个位置上没有编写 TU，则总是沿最短轨迹运行。

6.6.3 位置不唯一性示例

本文以一个 6 轴关节系统为例说明不同的关节位置如何导致位置的不唯一性。

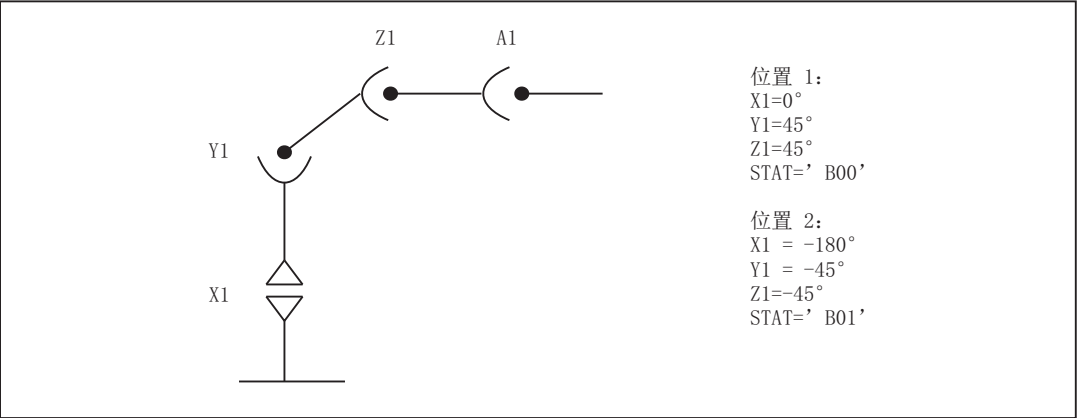


图 6-17 过顶区域的不唯一性

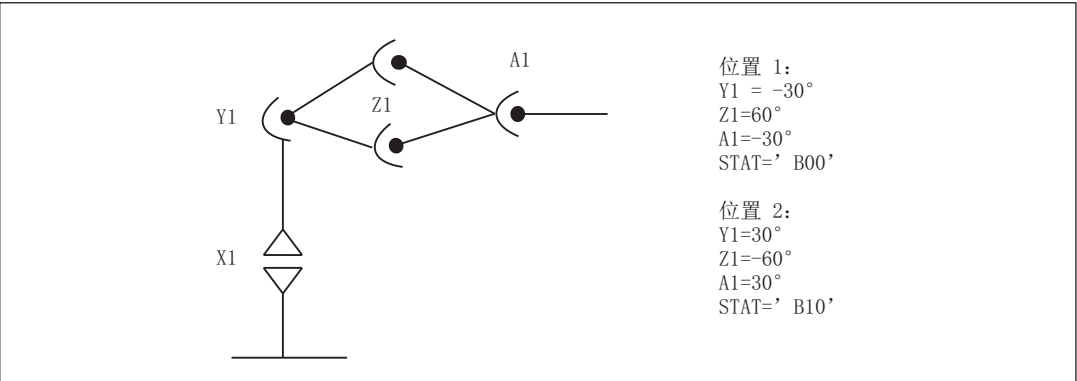


图 6-18 弯肘上方或下方的不唯一性

6.6 坐标 PTP 运动

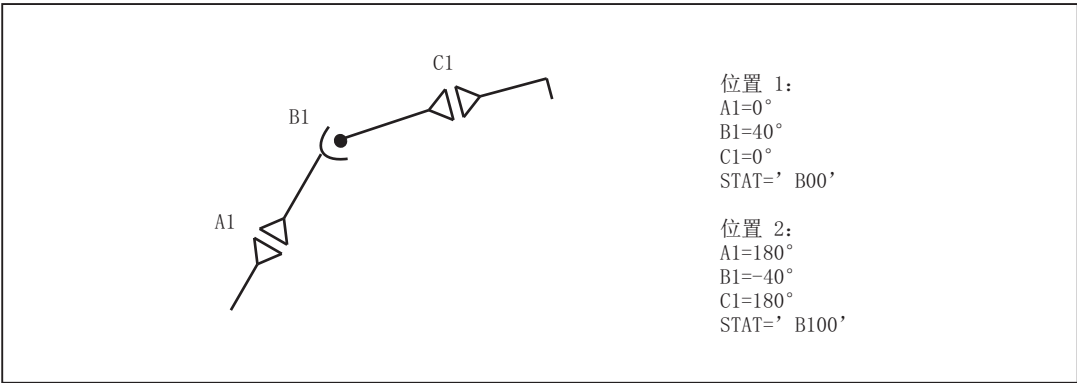


图 6-19 轴 B1 的不唯一性

6.6.4 回转轴位置不唯一性示例

下图中标出的回转轴位置可沿正方向或负方向达到。旋转方向在地址 A1 下编写。

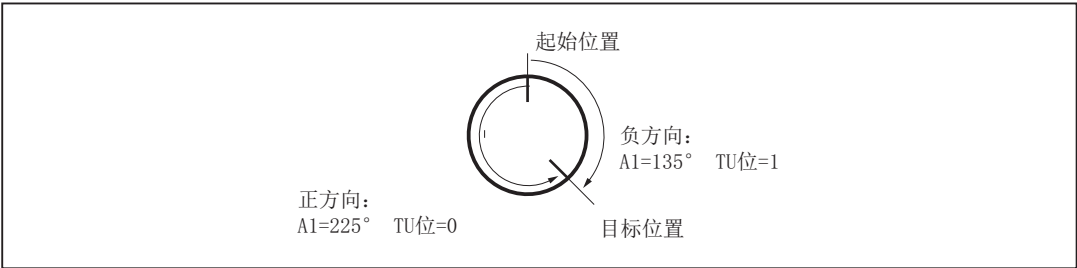


图 6-20 回转轴位置的不唯一性

6.6.5 JOG 模式下的 PTP/CP 切换

在 JOG 模式下可通过通道专用 NC/PLC 接口信号激活和取消坐标转换功能。仅当 JOG 模式下坐标转换已生效时，此控制信号才生效。

返回 AUTO 模式后将恢复为切换前生效的状态。

NC/PLC 接口信号

- 运行方式切换请求:
DB21, ... DBX29.4 (激活 PTP 运动)
- 运行方式生效反馈:
DB21, ... DBX317.6 (PTP 运动生效)

运行方式切换

“坐标 PTP 运动”功能只适用于 AUTO 和 MDA 模式。切换到 JOG 模式时会激活 CP 运动。再次切换回 AUTO 或 MDA 时，这些运行方式下上次设置的模式将重新生效。

REPOS

重新定位期间不会修改“坐标 PTP 运动”的设置。若之前在中断程序段中设置了 PTP，则会通过 PTP 重新定位。使用斜轴“TRAANG”功能时，重新定位运行方式下只有 CP 运动生效。

6.7 坐标手动运动（选件）

说明

使用“坐标手动运动”功能需要获得“Handling 坐标转换软件包”选件。

功能

通过“坐标手动运动”功能可在 JOG 方式中在以下直角坐标系中相互独立地设置轴。

- 基本坐标系（BCS）
- 工件坐标系（WCS）
- 刀具坐标系（TCS）

通过以下机床数据设置和激活坐标系：

MD21106 \$MC_CART_JOG_SYSTEM（JOG 方式下的坐标系）

位	含义
0	基本坐标系
1	工件坐标系
2	刀具坐标系

说明

工件坐标系通过框架指令参照基本坐标系进行了平移和旋转。

文档:

功能手册之基本功能分册; “轴、坐标系,框架(K2)”

参照系的显示:

	WCS = 工件零点		TCS = 刀具参考点

选择参照系

可通过以下设定数据为手动运动分别设定上述某个参照系, 几何轴的**平移** (粗平移) 和定向轴的**定向**以该参照系为参照物。

SD42650 \$SC_CART_JOG_MODE

若该数据不止一个位置位, 以平移坐标系或旋转坐标系, 或尝试设置未通过 MD21106 \$MC_CART_JOG_SYSTEM 使能的坐标系, 则会触发报警 14148“为坐标手动运动选择的坐标系非法”。

平移

通过平移运动, 刀尖 (TCP) 可 3 维平行于参照系的轴运动。该运动通过几何轴的 VDI 信号触发。

通过机床数据 MD24120\$MC_TRAFO_GEOAX_ASSIGN_TAB_x[n] 指定几何轴。多方向同时运动也可实现与参照系的轴成对角线的运动。

BCS 中的平移

基本坐标系 (BCS) 的零点是机床的坐标零点。

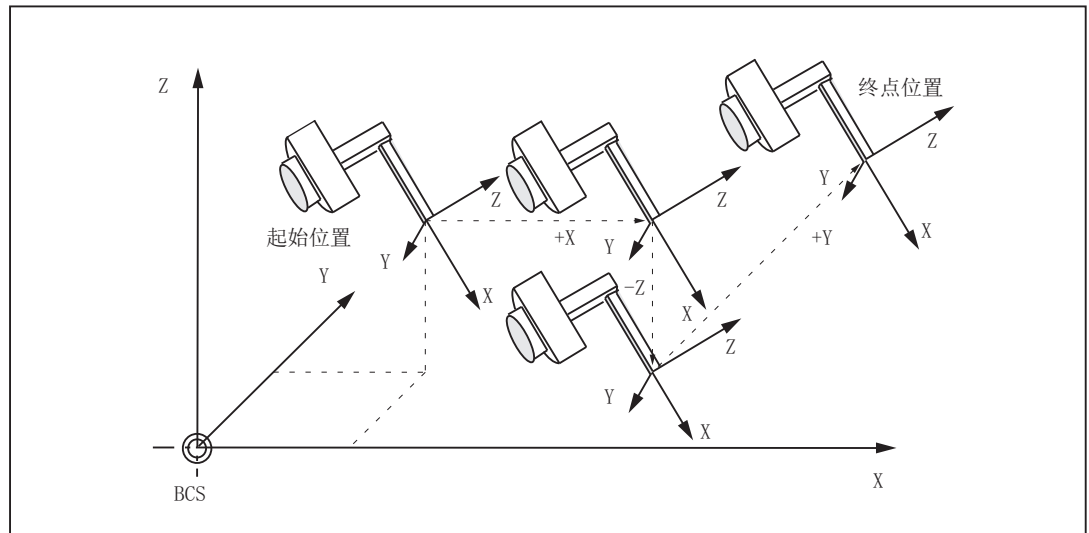


图 6-21 基本坐标系中的手动运动 (平移)

WCS 中的平移

工件坐标系 (WCS) 的零点位于工件零点。工件坐标系可通过框架指令参照基本坐标系进行平移和旋转。无旋转指令生效的情况下，平移的运动方式与基本坐标系中的相同。

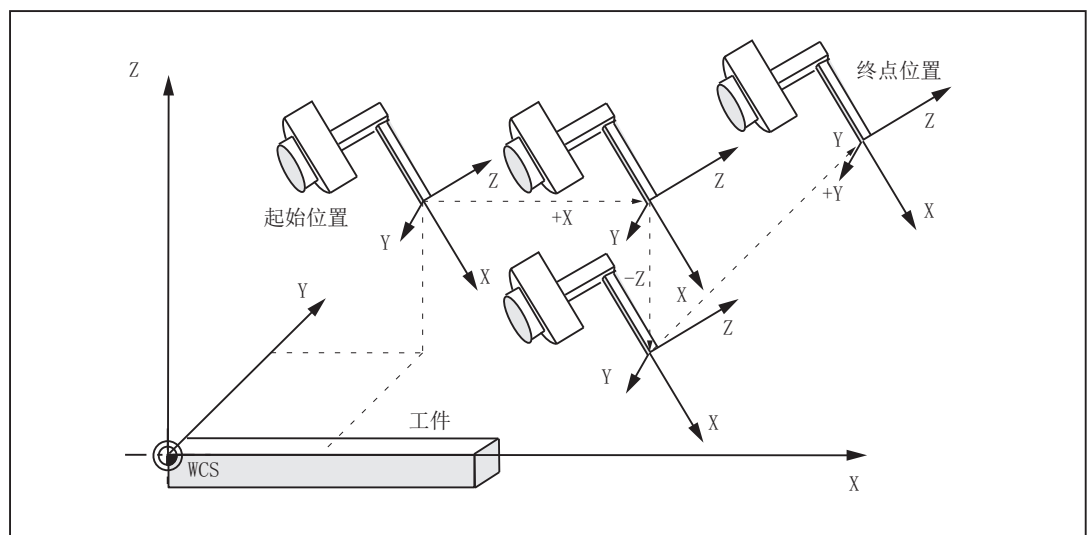


图 6-22 工件坐标系中的手动运动 (平移)

TCS 中的平移

刀具坐标系 (TCS) 的零点位于刀尖。刀具坐标系的方向取决于机床当前的位置，因为运行期间该坐标系也会跟着一起运行。

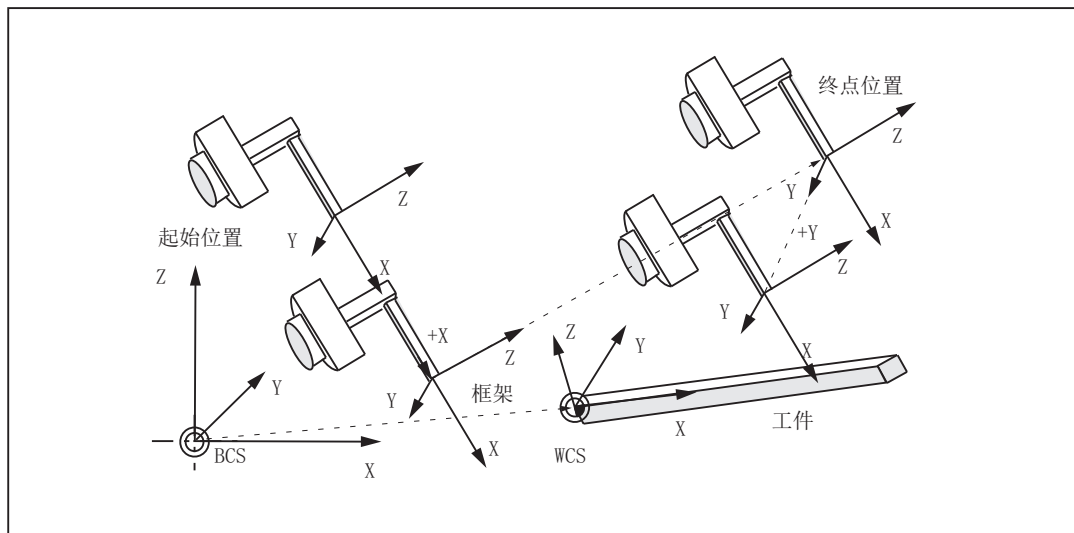


图 6-23 刀具坐标系中的手动运动 (平移)

TCS 中同时的平移与定向

若同时执行平移和定向运动，平移运动总是基于最新的刀具定向。这样可实现刀具方向的进给运动或者垂直于刀具方向的运动。

定向

通过定向运动可将刀具对准零件表面。定向运动由 PLC 通过定向轴的 VDI 信号 (DB21, ... DBB321) 提供给控制系统。

可同时运行多个定向轴。此时虚拟定向轴会围绕相应参照系中的固定轴旋转。

这些旋转根据 RPY 角命名。

- A 角: 绕 Z 轴旋转
- B 角: 绕 Y 轴旋转
- C 角: 绕 X 轴旋转

编写旋转:

用户可通过 G 代码组 50 定义如何执行旋转:

ORIEULER、ORIRPY、ORIVIRT1 和 ORIVIRT2

ORIVIRT1 指令会依据 MD21120 \$MC_ORIAX_TURN_TAB_1 执行旋转。通过以下机床数据将定向轴指定给通道轴: MD24585 \$MC_TRAFO5_ORIAX_ASSIGN_TAB_1。

旋转方向依据“右手准则”确定。此时拇指对准旋转轴。食指和中指便可指出正旋转方向。

WCS 中的定向

旋转是围绕工件坐标系的固定方向进行的。若无旋转指令，那么运动相当于基本坐标系中的旋转。

BCS 中的定向

旋转是围绕基本坐标系的固定方向进行的。

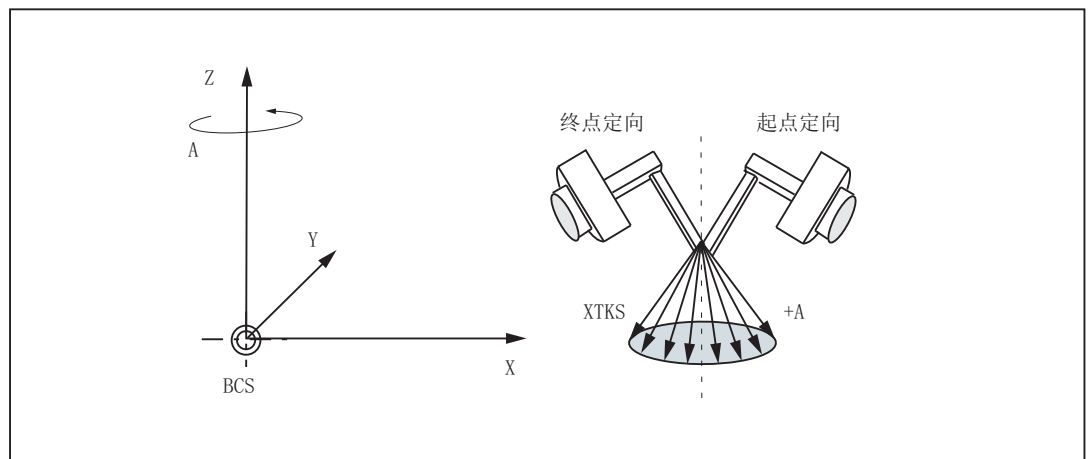


图 6-24 基本坐标系中的手动运动，定向角 A

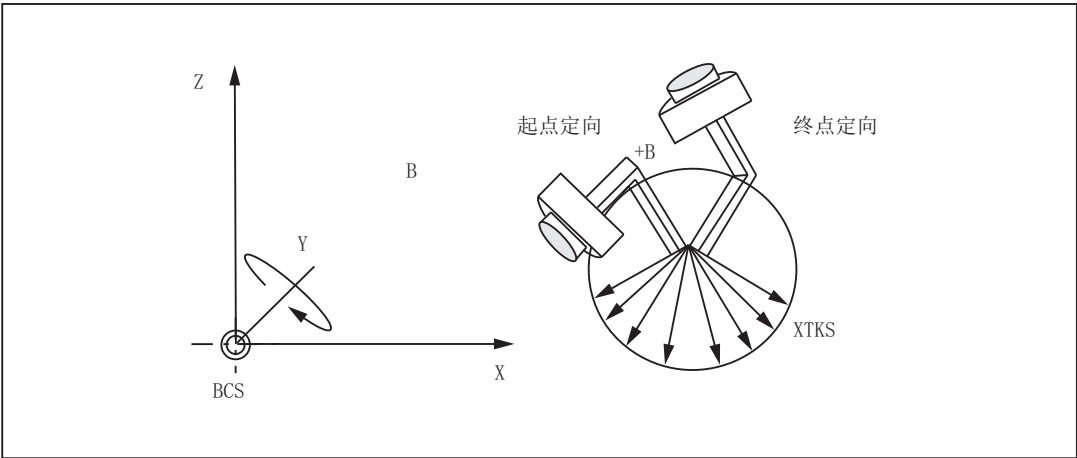


图 6-25 基本坐标系中的手动运动，定向角 B

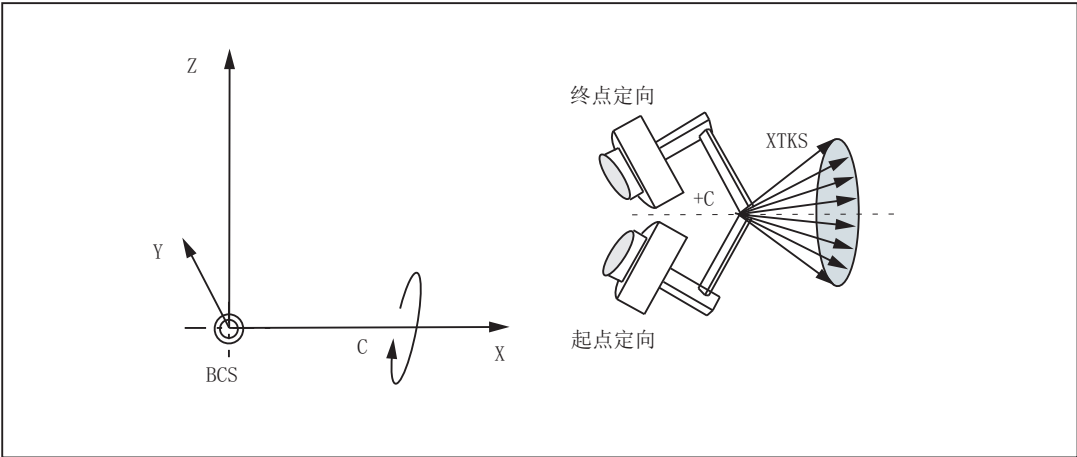


图 6-26 基本坐标系中的手动运动，定向角 C

TCS 中的定向

旋转是围绕刀具坐标系的变化方向进行的。此时总是将刀具的当前参考方向用作旋转轴。

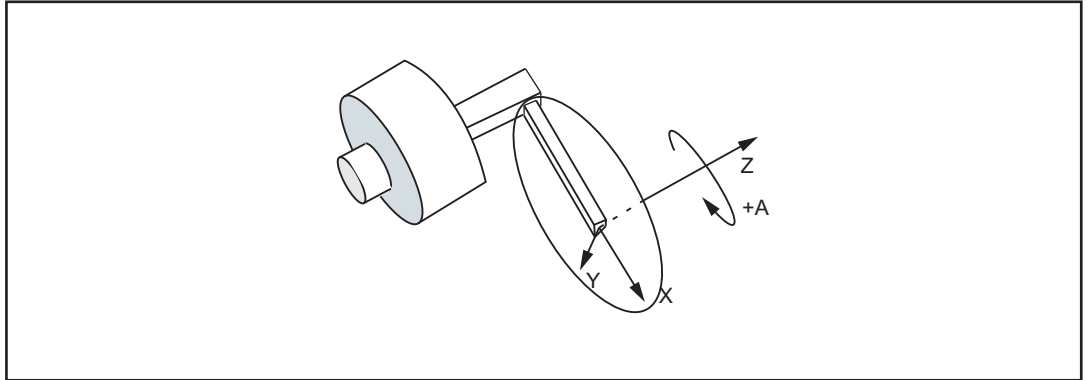


图 6-27 刀具坐标系中的手动运动，定向角 A

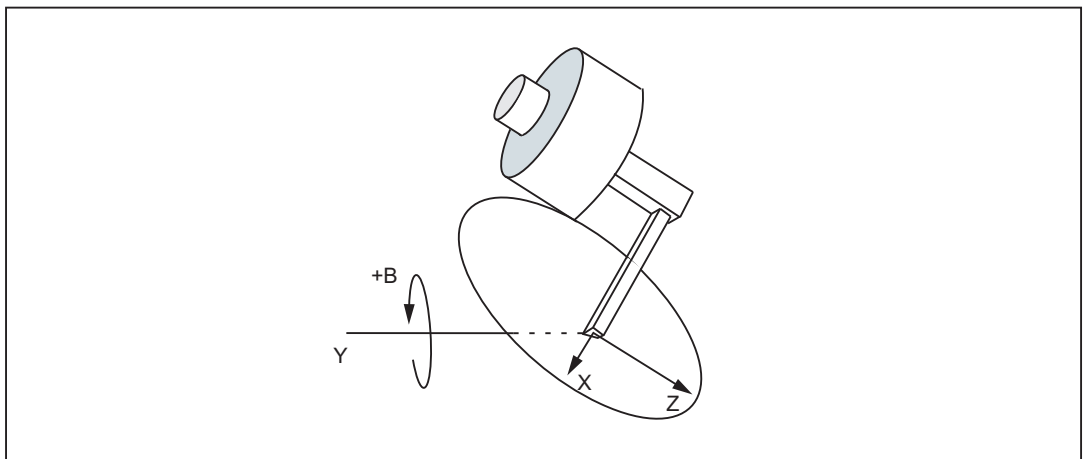


图 6-28 刀具坐标系中的手动运动，定向角 B

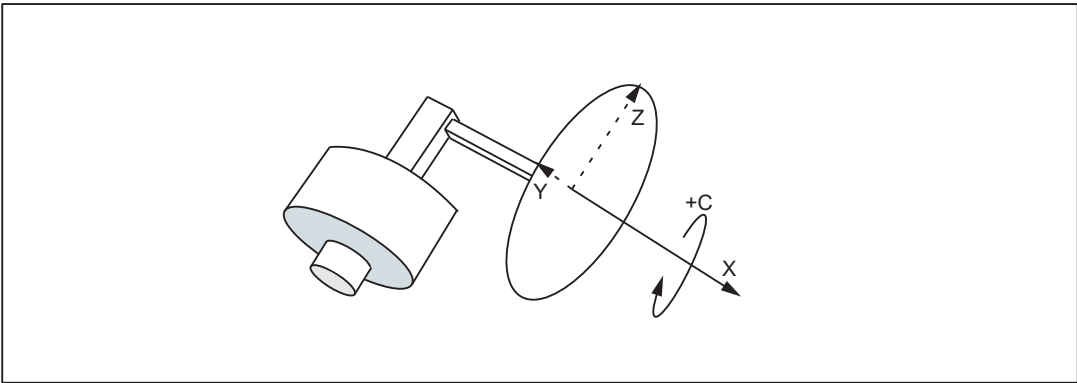


图 6-29 刀具坐标系中的手动运动，定向角 C

边界条件

仅当 NC 中激活了坐标转换时，才可执行“坐标手动运动”功能：DB21, ... DBX33.6 == 1（“坐标转换生效”）

此外须遵循以下边界条件：

- 设置了包含 5 轴或 6 轴转换的“Handling 坐标转换软件包”选件
- 必须已通过以下机床数据定义虚拟定向轴：
MD24585 \$MC_TRAFO5_ORIAX_ASSIGN_TAB_1[n]
- DB21, ... DBX29.4 == 0（激活 PTP 运动）
- MD21106 \$MC_CART_JOG_SYSTEM > 0

表格 6-2 坐标手动运动的条件

程序中坐标转换 (TRAORI..)	编写的运行方式	DB21, ... DBX29.4 “激活 PTP 运动”	DB21, ... DBX33.6 “坐标转换生效”
FALSE	未生效	未生效	0
TRUE	CP	0	1
TRUE	CP	1	0
TRUE	PTP	0	1
TRUE	PTP	1	0

程序中当前生效的 PTP/CP G 代码不会影响坐标手动运动。NC/PLC 接口信号在几何轴和定向轴的通道数据块中计算。

激活

如下设置坐标手动运动的坐标系：

- 通过以下机床数据激活“坐标手动运动”功能：
MD21106 \$MC_CART_JOG_SYSTEM > 0
通过 MD21106 \$MC_CART_JOG_SYSTEM 使能 BCS、WCS 或 TCS 参照系。
- 通过 SD42650 \$SC_CART_JOG_MODE 设置 JOG 运行
缺省值和之前一样：位 0 - 位 2 = 0，位 8 - 位 10 = 0。
平移运动的参标系通过位 0 - 位 2 定义，定向运动的参照系通过位 8 - 位 10 定义。
仅当不是所有位都为 0 时，才能通过新功能运行。平移运动和定向运动的参照系可相互独立地设置。

SD42650 \$SC_CART_JOG_MODE (只允许置位一个位)：

SD42650 \$SC_CART_JOG_MODE							
位 11 - 位 15	位 10	位 9	位 8	位 7 - 位 3	位 2	位 1	位 0
预留	TCS 中的 定向	WCS 中的 定向	BCS 中的 定向	预留	TCS 中的 平移	WCS 中的 平移	BCS 中的 平移

组合坐标系

下面中显示了所有可采用的参照系组合。

SD42650 \$SC_CART_JOG_MODE						运动参照系	
位 10	位 9	位 8	位 2	位 1	位 0	定向	平移
0	0	0	0/1	0/1	0/1	标准	标准
标准	标准	标准	0	0	0	标准	标准
0	0	1	0	0	1	BCS	BCS
0	0	1	0	1	0	BCS	WCS
0	0	1	1	0	0	BCS	TCS
0	1	0	0	0	1	WCS	BCS
0	1	0	0	1	0	WCS	WCS
0	1	0	1	0	0	WCS	TCS
1	0	0	0	0	1	TCS	BCS

6.8 通过零件程序/软键激活坐标转换机床数据

SD42650 \$SC_CART_JOG_MODE						运动参照系	
1	0	0	0	1	0	TCS	WCS
1	0	0	1	0	0	TCS	TCS

6.8 通过零件程序/软键激活坐标转换机床数据

6.8.1 功能

坐标转换涉及的机床数据可通过零件程序/软键激活，也就是说可通过零件程序写入这些数据，借此可完全修改坐标转换的配置。

控制系统中可设置十个不同的坐标转换功能。其中坐标转换的类型通过以下机床数据设置：

MD24100 \$MC_TRAFO_TYPE_1
到
MD24460 \$MC_TRAFO_TYPE_10

属性

坐标转换涉及的机床数据通过 **NEWCONFIG** 指令生效。

保护级为 7/7（KEYSWITCH_0），不需要特殊权限便可通过 **NC** 程序修改。

NEWCONFIG 指令生效时（可通过 **NC** 程序指令 **NEWCONF** 激活、通过 **HMI** 操作激活或在复位或程序结束时隐性激活）若未选择坐标转换，则可毫无限制地修改上述机床数据并激活数据。

修改不仅限于对现有坐标转换重新进行配置，还可以配置新的坐标转换，或将现有坐标转换替换成其他类型的坐标转换或删除。

6.8.2 边界条件

修改机床数据

不允许修改生效的坐标转换涉及的机床数据，否则会触发报警。

通常其中包含了所有和坐标转换数据组相关的机床数据。生效的坐标转换中包含但未使用的机床数据可进行修改（尽管此操作基本无意义）。例如，在通过 MD24100 \$MC_TRAFO_TYPE = 16（5 轴转换，配备可旋转刀具和两个相互垂直的回转轴）激活的坐标转换中，可修改机床数据 MD24564 \$MC_TRAFO5_NUTATOR_AX_ANGLE_n，因为该数据和该坐标转换无关。

此外，定向转换生效时不允许修改机床数据 MD21110 \$MC_X_AXIS_IN_OLD_X_Z_PLANE。

说明

程序中断（重新定位、删除剩余行程、异步子程序）时，控制系统内部需要多个已执行过的程序段用于重新定位。“禁止修改生效坐标转换涉及的机床数据”的规定也适用于这些程序段。

示例:

通过机床数据设置了两个定向转换，例如 MD24100 \$MC_TRAFO_TYPE_1 = 16, MD24200 \$MC_TRAFO_TYPE_2 = 18。

假设第二个转换在执行 NEWCONFIG 命令时生效。之后可修改所有只涉及第一个转换的机床数据，例如：

MD24500 \$MC_TRAFO5_PART_OFFSET_1

但是不可修改的机床数据例如有：

MD24650 \$MC_TRAFO5_BASE_TOOL_2

或者

MD21110 \$MC_X_AXIS_IN_OLD_X_Z_PLANE

此外例如还可通过 MD24300 \$MC_TRAFO_TYPE_3 = 256 设置另一个坐标转换（TRANSMIT）以及设置其他机床数据。

定义几何轴

必须在控制系统启动前通过以下机床数据定义几何轴：

MD24120 \$MC_TRAFO_GEOAX_ASSIGN_TAB_X[n]

或者

MD20050 \$MC_AXCONF_GEOAX_ASSIGN_TAB[n]

更改对应关系

坐标转换数据组和坐标转换间的对应关系取决于 MD24100 \$MC_TRAFO_TYPE_X 中条目的顺序。表中的第一条条目针对第一个坐标转换数据组，第二条条目指定给第二个坐标转换数据组。坐标转换生效后，该对应关系不允许且无法更改。

示例:

设置了三个坐标转换：两个定向转换和一个 TRANSMIT 坐标转换，例如

MD24100 \$MC_TRAFO_TYPE_1 = 16

; 定向转换，第 1 个定向转换数据组

MD24200 \$MC_TRAFO_TYPE_2 = 256: TRANSMIT 坐标转换

MD24300 \$MC_TRAFO_TYPE_3 = 18

; 定向转换，第 2 个定向转换数据组

第一个定向转换数据组指定给第一个转换（即第一个定向转换），第二个转换数据组指定给第三个转换（即第二个定向转换）。

若现在执行 NEWCONFIG 指令时第三个转换生效，则不允许将第一个转换切换为另一组的转换（例如 TRACYL），因为此时第三个转换不再是第二个定向转换，而是第一个。

不过此示例中允许为第一个转换设置另一个定向转换（例如通过 MD24100 \$MC_TRAFO_TYPE_1 = 32）；或在将第二个转换更改为定向转换（例如通过 MD24200 \$MC_TRAFO_TYPE_2 = 48）的同时将另一组中的转换设置为第一个转换（例如通过 MD24100 \$MC_TRAFO_TYPE_1 = 1024, TRAANG）。

6.8.3 上电、运行方式切换、复位、程序段搜索、REPOS 时的控制系统特性

设置以下机床数据，可在系统复位（包含了程序结束）和/或程序开始时自动选择坐标转换：

MD20110 \$MC_RESET_MODE_MASK

MD20112 \$MC_START_MODE_MASK

与

MD20140 \$MC_TRAFO_RESET_VALUE

如果修改了生效的坐标转换涉及的机床数据，这可能会导致程序开始或程序结束时发出报警。

为了避免在 NC 程序中重新配置坐标转换后输出此报警，建议如下编写 NC 程序：

6.8 通过零件程序/软键激活坐标转换机床数据

程序代码	注释
N10 TRAFOOF()	; 取消可能还在生效的坐标转换
N20\$MC_TRAFO5_BASE_TOOL_1[0]=0	; 写入机床数据
N30\$MC_TRAFO5_BASE_TOOL_1[0]=3	
N40\$MC_TRAFO5_BASE_TOOL_1[0]=200	
N130 NEWCONF	; 新写入的机床数据
	; 接收
N140 M30	

6.8.4 相关机床数据列表

可通过 NEWCONFIG 激活的机床数据列表见下。

所有坐标转换

所有坐标转换涉及的机床数据：

- MD24100 \$MC_TRAFO_TYPE_1 至 MD24480 \$MC_TRAFO_TYPE_10
- MD24110 \$MC_TRAFO_AXES_IN_1 至 MD24482 \$MC_TRAFO_AXES_IN_10
- MD24120 \$MC_TRAFO_GEOAX_ASSIGN_TAB_1 至 MD24484 \$MC_TRAFO_GEOAX_ASSIGN_TAB_10

定向转换

定向转换涉及的机床数据：

- MD24550 \$MC_TRAFO5_BASE_TOOL_1 和 MD24650 \$MC_TRAFO5_BASE_TOOL_2
- MD24558 \$MC_TRAFO5_JOINT_OFFSET_1 和 MD24658 \$MC_TRAFO5_JOINT_OFFSET_2
- MD24500 \$MC_TRAFO5_PART_OFFSET_1 和 MD24600 \$MC_TRAFO5_PART_OFFSET_2
- MD24510 \$MC_TRAFO5_ROT_AX_OFFSET_1 和 MD24610 \$MC_TRAFO5_ROT_AX_OFFSET_2
- MD24520 \$MC_TRAFO5_ROT_SIGN_IS_PLUS_1 和 MD24620 \$MC_TRAFO5_ROT_SIGN_IS_PLUS_2
- MD 24530: TRAFO5_NON_POLE_LIMIT_1 和 MD24630 \$MC_TRAFO5_NON_POLE_LIMIT_2

6.8 通过零件程序/软键激活坐标转换机床数据

- MD24540 \$MC_TRAFO5_POLE_LIMIT_1 和 MD24640 \$MC_TRAFO5_POLE_LIMIT_2
- MD24570 \$MC_TRAFO5_AXIS1_1 和 MD24670 \$MC_TRAFO5_AXIS1_2
- MD24572 \$MC_TRAFO5_AXIS2_1 和 MD24672 \$MC_TRAFO5_AXIS2_2
- MD24574 \$MC_TRAFO5_BASE_ORIENT_1 和 MD24674 \$MC_TRAFO5_BASE_ORIENT_2
- MD24562 \$MC_TRAFO5_TOOL_ROT_AX_OFFSET_1 和 MD24662 \$MC_TRAFO5_TOOL_ROT_AX_OFFSET_2
- MD24564 \$MC_TRAFO5_NUTATOR_AX_ANGLE_1 和 MD24664 \$MC_TRAFO5_NUTATOR_AX_ANGLE_2
- MD24566 \$MC_TRAFO5_NUTATOR_VIRT_ORIAX_1 和 MD24666 \$MC_TRAFO5_NUTATOR_VIRT_ORIAX_2

TRANSMIT 坐标转换

TRANSMIT 坐标转换涉及的机床数据:

- MD24920 \$MC_TRANSMIT_BASE_TOOL_1 和 MD24970 \$MC_TRANSMIT_BASE_TOOL_2
- MD24900 \$MC_TRANSMIT_ROT_AX_OFFSET_1 和 MD24950 \$MC_TRANSMIT_ROT_AX_OFFSET_2
- MD24910 \$MC_TRANSMIT_ROT_SIGN_IS_PLUS_1 和 MD24960 \$MC_TRANSMIT_ROT_SIGN_IS_PLUS_2
- MD24911 \$MC_TRANSMIT_POLE_SIDE_FIX_1 和 MD24961 \$MC_TRANSMIT_POLE_SIDE_FIX_2

TRACYL 坐标转换

TRACYL 坐标转换涉及的机床数据:

- MD24820 \$MC_TRACYL_BASE_TOOL_1 和 MD24870 \$MC_TRACYL_BASE_TOOL_2
- MD24800 \$MC_TRACYL_ROT_AX_OFFSET_1 和 MD24850 \$MC_TRACYL_ROT_AX_OFFSET_2

- MD24810 \$MC_TRACYL_ROT_SIGN_IS_PLUS_1 和 MD24870 \$MC_TRACYL_ROT_SIGN_IS_PLUS_2
- MD24808 \$MC_TRACYL_DEFAULT_MODE_1 和 MD24858 \$MC_TRACYL_DEFAULT_MODE_2

TRAANG 坐标转换

TRAANG 坐标转换涉及的机床数据:

- MD24710 \$MC_TRAANG_BASE_TOOL_1 和 MD24760 \$MC_TRAANG_BASE_TOOL_2
- MD24700 \$MC_TRAANG_ANGLE_1 和 MD24750 \$MC_TRAANG_ANGLE_2
- MD24720 \$MC_TRAANG_PARALLEL_VELO_RES_1 和 MD24770 \$MC_TRAANG_PARALLEL_VELO_RES_2
- MD24721 \$MC_TRAANG_PARALLEL_ACCEL_RES_1 和 MD24771 \$MC_TRAANG_PARALLEL_ACCEL_RES_2

坐标转换级联

坐标转换级联涉及的机床数据:

- MD24995 \$MC_TRACON_CHAIN_1 和 MD24996 \$MC_TRACON_CHAIN_2
- MD24997 \$MC_TRACON_CHAIN_3 和 MD24998 \$MC_TRACON_CHAIN_4

固定生效的坐标转换

固定生效的坐标转换涉及的机床数据:

- MD20144 \$MC_TRAFO_MODE_MASK
- MD20140 \$MC_TRAFO_RESET_VALUE
- MD20110 \$MC_RESET_MODE_MASK 和 MD20112 \$MC_START_MODE_MASK

6.9 边界条件

非坐标转换专用

非坐标转换专用的机床数据。这些数据不专用于某个坐标转换数据组，或者在坐标转换未生效时就有效：

- MD21110 \$MC_X_AXIS_IN_OLD_X_Z_PLANE
- MD21090 \$MC_MAX_LEAD_ANGLE
- MD21092 \$MC_MAX_TILT_ANGLE
- MD21100 \$MC_ORIENTATION_IS_EULER

6.9 边界条件

6.9.1 坐标转换级联

此功能可级联**两个**坐标转换。

但是并非任意坐标转换均可级联。

存在以下限制：

- 级联中的**第一个**转换必须为：
 - 定向转换（3 轴/4 轴/5 轴转换，万向铣头）
 - TRANSMIT
 - 柱面转换
 - 斜轴
- **第二个**转换必须为**斜轴**转换。
- 只允许级联两个转换。

允许在级联列表中只输入一个转换（例如用于测试）。

6.10 示例

6.10.1 TRANSMIT

此示例对应“图 6-30 有槽壁补偿的槽，柱面坐标 (页 447)”中图示的配置，其中包含了轴配置乃至激活 TRANSMIT 的主要步骤。

; 车削的通用轴配置

```
MD20060 $MC_AXCONF_GEOAX_NAME_TAB[0] = "X" ; 几何轴
MD20060 $MC_AXCONF_GEOAX_NAME_TAB[1] = "Y" ; 几何轴
MD20060 $MC_AXCONF_GEOAX_NAME_TAB[2] = "Z" ; 几何轴
MD20060 $MC_AXCONF_GEOAX_ASSIGN_TAB[0] = 1 ; X 轴为通道轴 1
MD20060 $MC_AXCONF_GEOAX_ASSIGN_TAB[1] = 0 ; Y 轴不是通道轴
MD20060 $MC_AXCONF_GEOAX_ASSIGN_TAB[2] = 2 ; Z 轴为通道轴 2
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[0]="XC"
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[1]="ZC"
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[2]="CC"
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[3]="ASC"
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[4] = " "
MD20070 $MC_AXCONF_MACHAX_USED[0] = 2 ; XC 轴为机床轴 2
MD20070 $MC_AXCONF_MACHAX_USED[1] = 3 ; ZC 轴为机床轴 3
MD20070 $MC_AXCONF_MACHAX_USED[2]=1 ; CC 轴为机床轴 1
MD20070 $MC_AXCONF_MACHAX_USED[3]=4 ; ASC 轴为机床轴 4
MD20070 $MC_AXCONF_MACHAX_USED[3] = 0 ; 空
MD20070 $MA_SPIND_ASSIGN_TO_MACHAX[AX1]= 1 ; C 轴为主轴 1
MD20070 $MA_SPIND_ASSIGN_TO_MACHAX[AX2]= 0 ; X 轴不是主轴
MD20070 $MA_SPIND_ASSIGN_TO_MACHAX[AX3]= 0 ; Z 轴不是主轴
MD20070 $MA_SPIND_ASSIGN_TO_MACHAX[AX4]= 2 ; AS 轴为主轴 2
MD10000 $MN_AXCONF_MACHAX_NAME_TAB[0]= "CM" ; 机床轴 1
MD10000 $MN_AXCONF_MACHAX_NAME_TAB[1]= "XM" ; 机床轴 2
MD10000 $MN_AXCONF_MACHAX_NAME_TAB[2]="ZM" ; 机床轴 3
MD10000 $MN_AXCONF_MACHAX_NAME_TAB[3]="ASM"; 机床轴 4
```

6.10 示例

```

; TRANSMIT 专用轴配置 (作为第一个且唯一的一个转换)
$MA_ROT_IS_MODULO[3] = TRUE           ; C 轴为模数轴
MD24100 $MC_TRAFO_TYPE_1 = 256        ; TRANSMIT 转换
MD24110 $MC_TRAFO_AXES_IN_1[0]=1      ; 垂直于回转轴的通道轴
MD24110 $MC_TRAFO_AXES_IN_1[1] = 3    ; 回转轴对应的通道轴
MD24110 $MC_TRAFO_AXES_IN_1[2]=2      ; 平行于回转轴的通道轴
MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1[0]=1 ; 通道轴 1 对应几何轴 X 轴
MD24120                               ; 通道轴 2 对应几何轴 Y 轴
$MC_TRAFO_GEOAX_ASSIGN_TAB_1[1]=3
MD24120                               ; 通道轴 3 对应几何轴 Z 轴
$MC_TRAFO_GEOAX_ASSIGN_TAB_1[2]=2
MD24900                               ; XY 平面相对于回转轴零位置的旋转
$MC_TRANSMIT_ROT_AX_OFFSET_1=0        角度
MD24910                               ; 回转轴旋转
$MC_TRANSMIT_ROT_SIGN_IS_PLUS_1=FALSE
E
MD24920 $MC_TRANSMIT_BASE_TOOL_1[0]=0.0 ; 刀具 X 轴偏移量
MD24920 $MC_TRANSMIT_BASE_TOOL_1[1]=0.0 ; 刀具 Y 轴偏移量
MD24920 $MC_TRANSMIT_BASE_TOOL_1[2]=0.0 ; 刀具 Z 轴偏移量
; 激活 TRANSMIT
; 编程 X、Y、Z
; 返回车削运行
TRAFOOF

```

参见

TRACYL (页 445)

6.10.2 TRACYL

此示例对应下图中显示的配置，其中包含了轴配置乃至激活 TRACYL 的主要步骤。

; 车削的通用轴配置

```

MD20060 $MC_AXCONF_GEOAX_NAME_TAB[0] = "X"      ; 几何轴
MD20060 $MC_AXCONF_GEOAX_NAME_TAB[1] = "Y"      ; 几何轴
MD20060 $MC_AXCONF_GEOAX_NAME_TAB[2] = "Z"      ; 几何轴
MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB[0] = 1      ; X 轴为通道轴 1
MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB[1] = 2      ; Y 轴不是通道轴
MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB[2] = 3      ; Z 轴为通道轴 2
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[0] = "XC"
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[1] = "YC"
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[2] = "ZC"
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[3] = "CC"
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[4] = "ASC"
MD20070 $MC_AXCONF_MACHAX_USED[0] = 2           ; X 轴为机床轴 2
MD20070 $MC_AXCONF_MACHAX_USED[1] = 3           ; Y 轴为机床轴 3
MD20070 $MC_AXCONF_MACHAX_USED[2] = 4           ; Z 轴为机床轴 4
MD20070 $MC_AXCONF_MACHAX_USED[3] = 1           ; C 轴为机床轴 1
MD20070 $MC_AXCONF_MACHAX_USED[4] = 5           ; AS 轴为机床轴 5
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[AX1]= 1      ; C 轴为主轴 1
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[AX2]= 0      ; X 轴不是主轴
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[AX3]= 0      ; Y 轴不是主轴
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[AX4]= 0      ; Z 轴不是主轴
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[AX5]= 2      ; AS 轴为主轴 2
MD10000 $MN_AXCONF_MACHAX_NAME_TAB[0]= "CM"     ; 机床轴 1
MD10000 $MN_AXCONF_MACHAX_NAME_TAB[1]= "XM"     ; 机床轴 2
MD10000 $MN_AXCONF_MACHAX_NAME_TAB[2]= "YM"     ; 机床轴 3
MD10000 $MN_AXCONF_MACHAX_NAME_TAB[3]= "ZM"     ; 机床轴 4
MD10000 $MN_AXCONF_MACHAX_NAME_TAB[4]= "ASM"    ; 机床轴 5

```

6.10 示例

```

; TRACYL 专用轴配置 (第一个且唯一一个转换)
MD24100 $MC_TRAFO_TYPE_1 = 513 ; 带槽壁补偿的 TRACYL 转换
MD24110 $MC_TRAFO_AXES_IN_1[0] = 1 ; 垂直于回转轴的通道轴
MD24110 $MC_TRAFO_AXES_IN_1[1] = 4 ; 圆柱外表面上垂直于回转轴的通道轴
MD24110 $MC_TRAFO_AXES_IN_1[2] = 3 ; 平行于回转轴的通道轴
MD24110 $MC_TRAFO_AXES_IN_1[3] = 2 ; 下标 [0] 附加轴对应的通道轴
MD24120 ; 通道轴 1 对应几何轴 X 轴
$MC_TRAFO_GEOAX_ASSIGN_TAB_1 [0]
= 1
MD24120 ; 通道轴 2 对应几何轴 Y 轴
$MC_TRAFO_GEOAX_ASSIGN_TAB_1[1]=
4
MD24120 ; 通道轴 3 对应几何轴 Z 轴
$MC_TRAFO_GEOAX_ASSIGN_TAB_1 [2]
= 3
MD24800 ; XY 平面相对于回转轴零位置的旋转角度
$MC_TRACYL_ROT_AX_OFFSET_1 = 0
MD24810 ; 回转轴旋转
$MC_TRACYL_ROT_SIGN_IS_PLUS_1 =
FALSE
MD24820 $MC_TRACYL_BASE_TOOL_1 ; 刀具 X 轴偏移量
[0] = 0.0
MD24820 $MC_TRACYL_BASE_TOOL_1 ; 刀具 Y 轴偏移量
[1] = 0.0
MD24820 $MC_TRACYL_BASE_TOOL_1 ; 刀具 Z 轴偏移量
[2] = 0.0
; 激活 TRACYL(40.0)
; Y 和 Z 编程参见以下示例
; 返回车削运行
TRAFOOF

```

带槽壁补偿的编程

```
(TRAFO_TYPE_n=513)
```

轮廓

如果需要加工的槽的宽度大于刀具宽度，可以相对于编写的参考轮廓通过 **G41/G42** 写入补偿方向以及通过 **OFFN** 写入槽壁距离（见下图）。

刀具半径

刀具半径自动考虑槽壁（见简化图）。平面刀具半径补偿的所有功能均可供使用（外角和内角的连续过渡以及瓶颈问题的解决方案）。

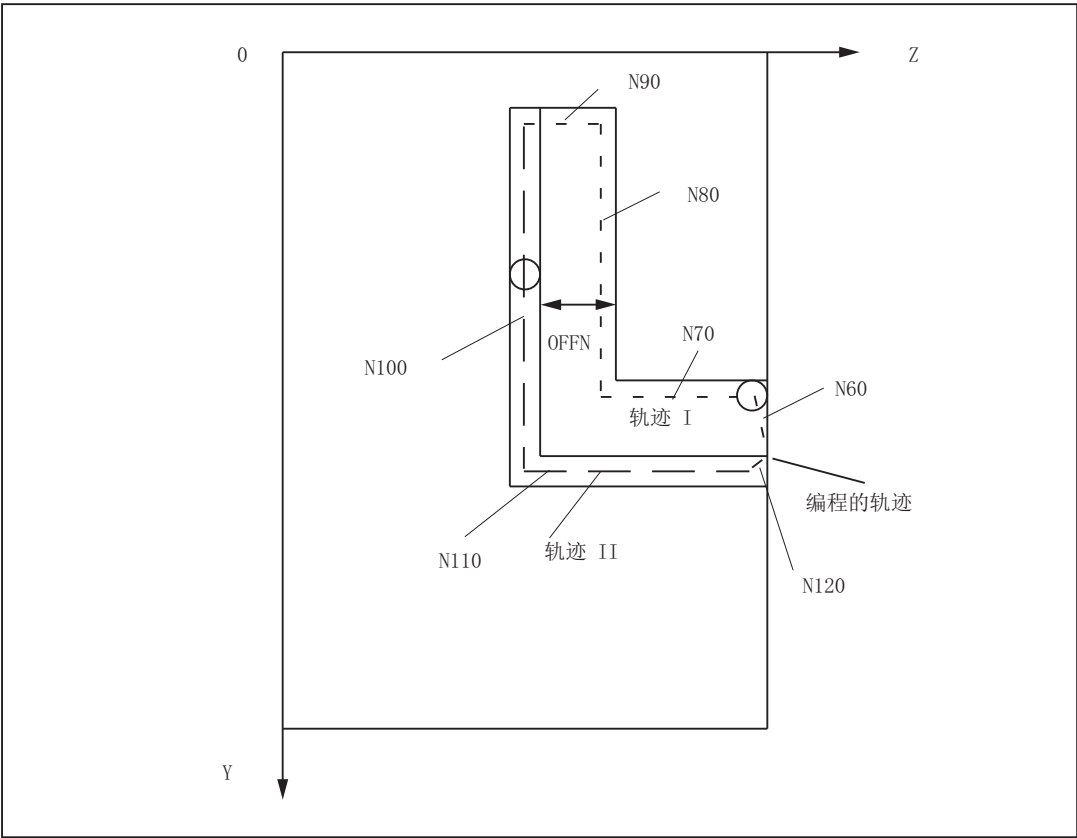


图 6-30 有槽壁补偿的槽，柱面坐标

示例程序，选择坐标转换后轨迹 I 上的刀具沿着轨迹 II 返回出发点

程序代码	注释
N1 SPOS=0;	; 主轴进入回转轴模式
N5 G0 X25 Y0 Z105 CC=200 F5000 G64	; 将主轴定位到槽中心上方
N10 TRACYL(40.)	; 选择坐标转换，参考直径：40 mm
N20 G19 G90	; 加工平面为圆柱外表面
N30 T1 D1	; 刀具选择，也可位于 TRACY (...) 前
N30 T1 D1	; 将刀具向槽底进给

6.10 示例

程序代码	注释
N50 OFFN=12	; 定义槽壁间距，不必单独占用一行

程序代码	注释
; 逼近槽壁	
N60 G1 Z100 G42	; ; 选择用于逼近槽壁的 TRC
加工槽段，轨迹 I	
N70 G1 Z50	; 平行于柱面的槽段
N80 G1 Y10	; 平行于圆周的槽段
; 逼近槽壁，轨迹 II	
N90 OFFN=4 G42	; 定义槽壁间距，并选择 TRC 用于逼近槽壁
; 加工槽段，轨迹 II	
N100 G1 Y70	; 对应 CC=200 度
N110 G1 Z100	; 返回起点
; 从槽壁回退	
N120 G1 Z105 G40	; ; 取消 TRC，用于离开槽壁
N130 G0 X25	; 从槽提刀
N140 TRAFOOF N150 G0 X25 Y0 Z105 CC=200 D0	; 返回出发点，并取消刀具补偿

无槽壁补偿的编程

采用附加直线轴、无槽壁补偿的 TRACYL（TRAFO_TYPE_n = 514）

; 以下机床数据是之后的零件程序的前提条件：

MD20070 \$MC_AXCONF_MACHAX_USED[0] = 1	; X 轴为机床轴 1
MD20070 \$MC_AXCONF_MACHAX_USED[1] = 2	; Y 轴为机床轴 2
MD20070 \$MC_AXCONF_MACHAX_USED[2] = 3	; Z 轴为机床轴 3
MD20070 \$MC_AXCONF_MACHAX_USED[3] = 4	; C 轴为机床轴 4
MD20070 \$MC_AXCONF_CHANAX_NAME_TAB[1] = "Y2"	
MD24120 \$MC_TRAFO_GEOAX_ASSIGN_TAB_1 [0] = 1;	X 轴为通道轴 1
MD24120 \$MC_TRAFO_GEOAX_ASSIGN_TAB_1 [1] = 2;	Y 轴不是通道轴
MD24120 \$MC_TRAFO_GEOAX_ASSIGN_TAB_1 [2] = 3;	Z 轴为通道轴 2
MD24100 \$MC_TRAFO_TYPE_1 = 514	; TRACYL，无槽壁补偿，带扩展刀具长度补偿
MD24110 \$MC_TRAFO_AXES_IN_1[0] = 1	; 垂直于回转轴的通道轴
MD24110 \$MC_TRAFO_AXES_IN_1[1] = 4	; 柱面上的通道轴
	; 垂直于回转轴

MD24110 \$MC_TRAFO_AXES_IN_1[2] = 3 ; 平行于回转轴的通道轴
MD24110 \$MC_TRAFO_AXES_IN_1[3] = 2 ; 下标 [0] 附加轴对应的通道轴
MD24120 \$MC_TRAFO_GEOAX_ASSIGN_TAB_1 [0] = 1; 通道轴 1 对应几何轴 X 轴
MD24120 \$MC_TRAFO_GEOAX_ASSIGN_TAB_1 [1] = 4; 通道轴 2 对应几何轴 Y 轴
MD24110 \$MC_TRAFO_GEOAX_ASSIGN_TAB_1 [2] = 3; 通道轴 3 对应几何轴 Z 轴
MD24808 \$MC_TRACYL_DEFAULT_MODE_1 =0 ; 或不设置

刀具数据:

\$TC_DP1[1,1]=120 ; 刀具类型为立铣刀
\$TC_DP2[1,1]=0
\$TC_DP3[1,1]=0 ; 长度补偿矢量
\$TC_DP4[1,1]=25
\$TC_DP5[1,1]=5
\$TC_DP6[1,1]=4 ; 半径, 刀具半径

零件程序:

程序代码	注释
N1001 T1 D1 G54 G19 G90 F5000 G64	; 选择第 1 个无槽壁补偿的 TRACYL
N1005 G0 X25 Y0 Z105 A=200	
N1010 TRACYL(40.)	; 选择坐标转换
N1040 G1 X20	
N1060 G1 Z100	
N1070 G1 Z50	
N1080 G1 Y10	
N1140 TROFOOF	
N1150 G0 X25 Y0 Z105 A=200	; 选择第 1 个带槽壁补偿的 TRACYL
N2010 G0 TRACYL(40.,1,1)	; 也可编写 TRCYL(40., ,1)
N2040 G1 X20	
N2060 G1 Z100	
N2070 G1 Z50	
N2080 G1 Y10	
N2140 TROFOOF	

6.10.3 TRAANG

此示例对应“有槽壁补偿的槽，柱面坐标”中显示的配置，其中包含了轴配置乃至激活 TRAANG 的主要步骤。

; 磨削的通用轴配置

```
MD20060 $MC_AXCONF_GEOAX_NAME_TAB[0] = "X" ; 几何轴
MD20060 $MC_AXCONF_GEOAX_NAME_TAB[1] = " " ; 几何轴
MD20060 $MC_AXCONF_GEOAX_NAME_TAB[2] = "Z" ; 几何轴
MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB[0] = 0 ; X 非通道轴
MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB[1] = 0 ; Y 轴不是通道轴
MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB[2] = 1 ; Z 轴为通道轴 1
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[0] = "Z"
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[1] = "C"
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[2] = "AS1"
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[3] = "MU"
MD20070 $MC_AXCONF_MACHAX_USED[0] = 3 ; Z 轴为机床轴 3
MD20070 $MC_AXCONF_MACHAX_USED[1] = 1 ; C 轴为机床轴 1
MD20070 $MC_AXCONF_MACHAX_USED[2] = 4 ; AS 轴为机床轴 4
MD20070 $MC_AXCONF_MACHAX_USED[3] = 2 ; MU 轴为机床轴 2
MD20070 $MC_AXCONF_MACHAX_USED[3] = 0 ; 空
MD20070 $MC_AXCONF_MACHAX_USED[3] = 0 ; 空
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[AX1]= 1 ; C 轴为主轴 1
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[AX2]= 0 ; X 轴不是主轴
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[AX3]= 0 ; Z 轴不是主轴
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[AX4]= 2 ; AS 轴为主轴 2
MD10000 $MN_AXCONF_MACHAX_NAME_TAB[0]= "C1" ; 机床轴 1
MD10000 $MN_AXCONF_MACHAX_NAME_TAB[1]= "MU" ; 机床轴 2
MD10000 $MN_AXCONF_MACHAX_NAME_TAB[2]= "MZ" ; 机床轴 3
MD10000 $MN_AXCONF_MACHAX_NAME_TAB[3]= "AS1" ; 机床轴 4
```

; TRAANG 专用轴配置（第一个且唯一一个转换）

```
MD24100 $MC_TRAFO_TYPE_1 = 1024 ; TRAANG 转换
```

```

MD24110 $MC_TRAFO_AXES_IN_1[0] = 4           ; 斜轴对应的通道轴
MD24110 $MC_TRAFO_AXES_IN_1[1] = 1           ; 平行于 Z 轴的通道轴
MD24110 $MC_TRAFO_AXES_IN_1[2] = 0           ; 通道轴未生效
MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1 [0]; X 轴为通道轴 1
= 4
MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1 [1]; Y 轴为通道轴 2
= 0
MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1 [2]; Z 轴为通道轴 3
= 1
MD24700 $MC_TRAANG_ANGLE_1 = 30.             ; 斜轴倾斜角度
MD24710 $MC_TRAANG_BASE_TOOL_1 [0] = 0       ; 刀具 X 轴偏移量
MD24710 $MC_TRAANG_BASE_TOOL_1 [1] = 0       ; 刀具 Y 轴偏移量
MD24710 $MC_TRAANG_BASE_TOOL_1 [2] = 0       ; 刀具 Z 轴偏移量
TRAANG                                         ; 激活
                                              ; 编程 X、Y、Z
TRAFOOF                                       ; 返回车削运行

```

6.10.4 坐标转换级联

示例

下面的章节将介绍：

- 一般通道配置
- 单个坐标转换
- 由上文介绍的单个坐标转换构成的级联
- 单个坐标转换的激活
- 坐标转换级联的激活

示例涵盖以下坐标转换：

- 5 轴转换，配备可旋转刀具，轴序 AB（转换类型 16）
- TRANSMIT（转换类型 256）
- 斜轴（转换类型 1024）

6.10 示例

- 第 1 转换和第 3 转换的级联（转换类型 8192）
- 第 2 转换和第 3 转换的级联（转换类型 8192）

一般通道配置

```
CHANDATA(1) ; 通道 1 中的通道数据
MD20070 $MC_AXCONF_MACHAX_USED[0] = 1
MD20070 $MC_AXCONF_MACHAX_USED[1] = 2
MD20070 $MC_AXCONF_MACHAX_USED[2] = 3
MD20070 $MC_AXCONF_MACHAX_USED[3] = 4
MD20070 $MC_AXCONF_MACHAX_USED[4] = 5
MD20070 $MC_AXCONF_MACHAX_USED[5] = 6
MD20070 $MC_AXCONF_MACHAX_USED[6] = 7
MD20070 $MC_AXCONF_MACHAX_USED[7] = 0
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[3]="A"
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[4]="B"
MD20080 $MC_AXCONF_CHANAX_NAME_TAB[5]="C"
MD36902 $MA_IS_ROT_AX[ AX4 ] = TRUE
MD36902 $MA_IS_ROT_AX[ AX5 ] = TRUE
MD36902 $MA_IS_ROT_AX[ AX6 ] = TRUE
MD36902 $MA_IS_ROT_AX[ AX7 ] = TRUE
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[ AX5 ] = 0
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[AX7] = 1
MD35000 $MA_ROT_IS_MODULO[AX7] = TRUE
```

单个坐标转换

```
; 第 1 个转换 TRAORI
MD24470 $MC_TRAFO_TYPE_1= 16 ; TRAORI: A-B 运动系统
MD24410 $MC_TRAFO_AXES_IN_1[0]=1
MD24410 $MC_TRAFO_AXES_IN_1[1]=2
MD24410 $MC_TRAFO_AXES_IN_1[2]=3
MD24410 $MC_TRAFO_AXES_IN_1[3]=4
```

```

MD24410 $MC_TRAFO_AXES_IN_1[4]=5
MD24410 $MC_TRAFO_AXES_IN_1[5]=0
MD24120$MC_TRAFO_GEOAX_ASSIGN_TAB_1[0]=1
MD24120$MC_TRAFO_GEOAX_ASSIGN_TAB_1[1]=2
MD24120$MC_TRAFO_GEOAX_ASSIGN_TAB_1[2]=3
MD24550$MC_TRAFO5_BASE_TOOL_1[0]=0
MD24550$MC_TRAFO5_BASE_TOOL_1[1]=0
MD24550$MC_TRAFO5_BASE_TOOL_1[2]=0

```

;第 2 个转换 TRANSMIT

```

MD24200 $MC_TRAFO_TYPE_2 = 256      ;TRANSMIT
MD24210 $MC_TRAFO_AXES_IN_2[0]=1
MD24210 $MC_TRAFO_AXES_IN_2[1]=6
MD24210 $MC_TRAFO_AXES_IN_2[2]=3
MD24210 $MC_TRAFO_AXES_IN_2[3]=0
MD24210 $MC_TRAFO_AXES_IN_2[4]=0
MD24210 $MC_TRAFO_AXES_IN_2[5]=0
MD24210 $MC_TRAFO_AXES_IN_2[6]=0
MD24220 $MC_TRAFO_GEOAX_ASSIGN_TAB_2[0] =1
MD24220 $MC_TRAFO_GEOAX_ASSIGN_TAB_2[1] =6
MD24220 $MC_TRAFO_GEOAX_ASSIGN_TAB_2[2] =3

```

;第 3 个转换 TRAANG

```

MD24300 $MC_TRAFO_TYPE_3 = 1024    ;TRAANG
MD24310 $MC_TRAFO_AXES_IN_3[0] = 1
MD24310 $MC_TRAFO_AXES_IN_3[1] = 3
MD24310 $MC_TRAFO_AXES_IN_3[2] = 2
MD24310 $MC_TRAFO_AXES_IN_3[3] = 0
MD24310 $MC_TRAFO_AXES_IN_3[4] = 0
MD24320 $MC_TRAFO_GEOAX_ASSIGN_TAB_3[0] = 1
MD24320 $MC_TRAFO_GEOAX_ASSIGN_TAB_3[1] = 3
MD24320 $MC_TRAFO_GEOAX_ASSIGN_TAB_3[2] = 2
MD24700 $MC_TRAANG_ANGLE_1 = 45.

```

6.10 示例

```
MD24720 $MC_TRAANG_PARALLEL_VELO_RES_1 = 0.2
MD24721 $MC_TRAANG_PARALLEL_ACCEL_RES_1 = 0.2
MD24710 $MC_TRAANG_BASE_TOOL_1 [0] = 0.0
MD24710 $MC_TRAANG_BASE_TOOL_1 [1] = 0.0
MD24710 $MC_TRAANG_BASE_TOOL_1 [2] = 0.0
```

坐标转换级联

```
; 第 4 个转换 TRACON (级联 TRAORI/TRAANG)
MD24400 $MC_TRAFO_TYPE_4 = 8192
MD24420 $MC_TRAFO_GEOAX_ASSIGN_TAB_4[0] =2
MD24420 $MC_TRAFO_GEOAX_ASSIGN_TAB_4[1] =1
MD24420 $MC_TRAFO_GEOAX_ASSIGN_TAB_4[2] =3
MD24995 $MC_TRACON_CHAIN_1[0] = 1
MD24995 $MC_TRACON_CHAIN_1[1] = 3
MD24995 $MC_TRACON_CHAIN_1[2] = 0
; 第 5 个转换 TRACON (级联 TRANSMIT/TRAANG)
MD24430 $MC_TRAFO_TYPE_5 = 8192
MD24434 $MC_TRAFO_GEOAX_ASSIGN_TAB_5[0] =1
MD24434 $MC_TRAFO_GEOAX_ASSIGN_TAB_5[1] =6
MD24434 $MC_TRAFO_GEOAX_ASSIGN_TAB_5[2] =3
MD24996 $MC_TRACON_CHAIN_2[0] = 2
MD24996 $MC_TRACON_CHAIN_2[1] = 3
MD24996 $MC_TRACON_CHAIN_2[2] = 0
```

零件程序（节选）

NC 程序示例，含设置的坐标转换：

程序代码	注释
; 调用单个坐标转换	
\$TC_DP1[1,1]=120	; 定义刀具
\$TC_DP3[1,1]= 10	; 刀具类型
n2 x0 y0 z0 a0 b0 f20000 t1 d1n4 x20	; 刀具长度
n30 TRANSMIT	; 激活 TRANSMIT

程序代码	注释
n40 x0 y20	
n50 x-20 y0	
n60 x0 y-20	
n70 x20 y0	
n80 TRAFOOF	; 取消 TRANSMIT
n130 TRAANG(45.)	; 激活斜轴转换, 参数: 45°角
n140 x0 y0 z20	
n150 x-20 z0	
n160 x0 z-20	
n170 x20 z0	

说明

上面的示例假定机床上“斜轴”的角度可设置, 且在激活单个坐标转换时被设为 0°。

程序代码	注释
; 调用第 1 个坐标转换级联	
; TRAORI + TRAANG	
n230 TRACON(1, 45.)	; 激活 2 个坐标转换级联中的第 1 个级联
	; 之前生效的 TRAANG 自动取消
	; 斜轴的参数为 45°
n240 x10 y0 z0 a3=-1 C3 =1 oriwks	
n250 x10 y20 b3 = 1 c3 = 1	

程序代码	注释
; 调用第 2 个坐标转换级联	
; TRANSMIT + TRAANG	
n330 TRACON(2, 40.)	; 激活第 2 个坐标转换级联
	; 斜轴的参数为 40°
n335 x20 y0 z0	
n340 x0 y20 z10	
n350 x-20 y0 z0	
n360 x0 y-20 z0	
n370 x20 y0 z0	
n380 TRAFOOF	取消第 2 个坐标转换级联
n1000 M30	

6.10 示例

6.10.5 通过零件程序激活坐标转换涉及的机床数据

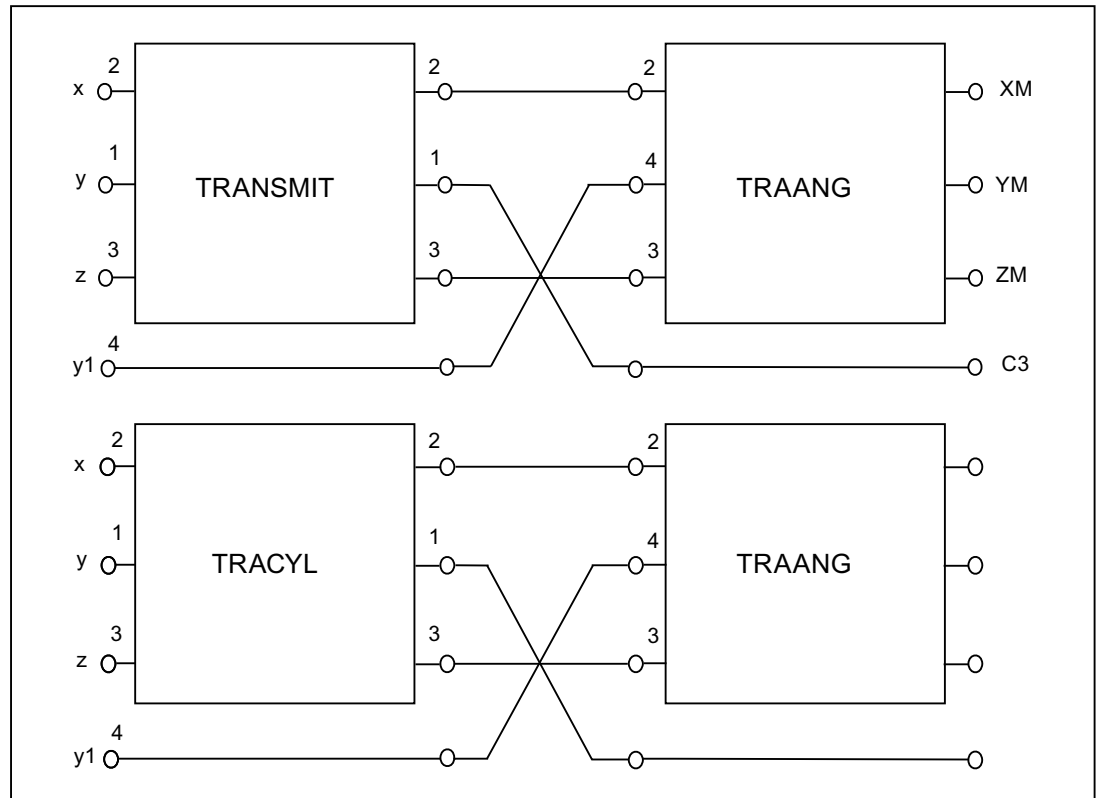
以下示例中允许在程序段 N90 中另外编写一个涉及第二个转换的机床数据（例如 MD24650 \$MC_TRAFO5_BASE_TOOL_2[2]），因为单独编写时该数据不会生效。但如果程序段 N130 中的程序未经修改，则将触发报警，因为之后系统会尝试修改生效的转换。

程序示例：

程序代码	注释
N40 TRAORI (2)	； 选择第 2 个定向转换
N50 X0 Y0 Z0 F20000 T1 T1	
N60 A50 B50	
N70 A0 B0	
N80 X10	
N90 \$MC_TRAFO5_BASE_TOOL_1[2] = 50	； 改写第 1 个定向转换的机床数据
N100 A20	
N110 X20	
N120 X0	
N130 NEWCONF	； 接收新写入的机床数据
N140 TRAORI (1)	； 第 1 个定向转换机床数据的选择生效
N150 G19 X0 Y0 Z0	
N160 A50 B50	
N170 A0 B0	
N180 TRAFOOF	
N190 M30	

6.10.6 坐标转换级联中的轴位置

下面的示例中配置了两个坐标转换级联，并且在零件程序中通过同步动作循环读取系统变量，用于确定轴位置。



机床数据

CHANDATA(1)

```
MD24100 $MC_TRAFO_TYPE_1=256 ; TRANSMIT
MD24110 $MC_TRAFO_AXES_IN_1[0]=2
MD24110 $MC_TRAFO_AXES_IN_1[1]=1
MD24110 $MC_TRAFO_AXES_IN_1[2]=3
MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1[0]=2
MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1[1]=1
MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1[2]=3
```

6.10 示例

MD24200 \$MC_TRAFO_TYPE_2=512 ; TRACYL
 MD24210 \$MC_TRAFO_AXES_IN_2[0]=2
 MD24210 \$MC_TRAFO_AXES_IN_2[1]=1
 MD24210 \$MC_TRAFO_AXES_IN_2[2]=3
 MD24220 \$MC_TRAFO_GEOAX_ASSIGN_TAB_2[0]=2
 MD24220 \$MC_TRAFO_GEOAX_ASSIGN_TAB_2[1]=1
 MD24220 \$MC_TRAFO_GEOAX_ASSIGN_TAB_2[2]=3

MD24300 \$MC_TRAFO_TYPE_3=1024 ; TRAANG
 MD24310 \$MC_TRAFO_AXES_IN_3[0]=2
 MD24310 \$MC_TRAFO_AXES_IN_3[1]=4
 MD24310 \$MC_TRAFO_AXES_IN_3[2]=3
 MD24320 \$MC_TRAFO_GEOAX_ASSIGN_TAB_3[0]=2
 MD24320 \$MC_TRAFO_GEOAX_ASSIGN_TAB_3[1]=4
 MD24320 \$MC_TRAFO_GEOAX_ASSIGN_TAB_3[2]=3
 MD24700 \$MC_TRAANG_ANGLE_1 = 45.
 MD24720 \$MC_TRAANG_PARALLEL_VELO_RES_1 = 0.2
 MD24721 \$MC_TRAANG_PARALLEL_ACCEL_RES_1 =
 0.2
 MD24710 \$MC_TRAANG_BASE_TOOL_1[0] = 0.0
 MD24710 \$MC_TRAANG_BASE_TOOL_1[1] = 0.0
 MD24710 \$MC_TRAANG_BASE_TOOL_1[2] = 0.0

第 1 级联: TRANSMIT / TRAANG

MD24400 \$MC_TRAFO_TYPE_4=8192 ; TRACON (1)
 MD24995 \$MC_TRACON_CHAIN_1[0]=1
 MD24995 \$MC_TRACON_CHAIN_1[1]=3
 MD24995 \$MC_TRACON_CHAIN_1[2]=0
 MD24995 \$MC_TRACON_CHAIN_1[3]=0
 MD24410 \$MC_TRAFO_AXES_IN_4[0]=1
 MD24410 \$MC_TRAFO_AXES_IN_4[1]=2
 MD24410 \$MC_TRAFO_AXES_IN_4[2]=3
 MD24420 \$MC_TRAFO_GEOAX_ASSIGN_TAB_4[0]=2

MD24420 \$MC_TRAFO_GEOAX_ASSIGN_TAB_4[1]=1

MD24420 \$MC_TRAFO_GEOAX_ASSIGN_TAB_4[2]=3

第 2 级联 TRACYL / TRAANG

MD24430 \$MC_TRAFO_TYPE_5=8192 ; TRACON (2)

MD24996 \$MC_TRACON_CHAIN_2[0]=2

MD24996 \$MC_TRACON_CHAIN_2[1]=3

MD24996 \$MC_TRACON_CHAIN_2[2]=0

MD24996 \$MC_TRACON_CHAIN_2[3]=0

MD24432 \$MC_TRAFO_AXES_IN_5[0]=1

MD24432 \$MC_TRAFO_AXES_IN_5[1]=2

MD24432 \$MC_TRAFO_AXES_IN_5[2]=3

MD24434 \$MC_TRAFO_GEOAX_ASSIGN_TAB_5[0]=2

MD24434 \$MC_TRAFO_GEOAX_ASSIGN_TAB_5[1]=1

MD24434 \$MC_TRAFO_GEOAX_ASSIGN_TAB_5[2]=3

M17

零件程序

程序代码	注释
N10 \$TC_DP1[1,1]=120	
N20 \$TC_DP3[1,1]=20	
N30 \$TC_DP4[1,1]=0	
N40 \$TC_DP5[1,1]=0	
N50	
N60 X0 Y0 Z0 F20000 T1 D1	
N70	
N80	; 在同步动作中循环读取变量
N90 ID=1 WHENEVER TRUE DO \$R0=\$AA_ITR[X,0] \$R1=\$AA_ITR[X,1] \$R2=\$AA_ITR[X,2]	
N100 ID=2 WHENEVER TRUE DO \$R3=\$AA_IBC[X] \$R4=\$AA_IBC[Y] \$R5=\$AA_IBC[Z]	
N110 ID=3 WHENEVER TRUE DO \$R6=\$VA_IW[X]-\$AA_IW[X]	
N120 ID=4 WHENEVER TRUE DO \$R7=\$VA_IB[X]-\$AA_IB[X]	
N130 ID=5 WHENEVER TRUE DO \$R8=\$VA_IBC[X]-\$AA_IBC[X]	
N140 ID=6 WHENEVER TRUE DO \$R9=\$VA_ITR[X,1]-\$AA_ITR[X,1]	
N150	
N160	; 第 1 级联: TRANSMIT / TRAANG
N170 TRACON(1,)	

M1: 坐标转换

6.11 数据表

程序代码	注释
N180 X20 Y0 Z0	
N190 X0 Y20 Z10	
N200 X-20 Y0 Z0	
N210 X0 Y-20 Z0	
N220 X20 Y0 Z0	
N230 TRAFOOF	
N240	
N250	; 第 2 级联 TRACYL / TRAANG
N260 TRACON (2, 40.)	
N270 X20 Y0 Z0	
N280 X0 Y20 Z10	
N290 X-20 Y0 Z0	
N300 X0 Y-20 Z0	
N310 X20 Y0 Z0	
N320 TRAFOOF	
N330	
N340 M30	

6.11 数据表

6.11.1 机床数据

6.11.1.1 TRANSMIT

通道专用机床数据

编号	名称: \$MC_	描述
20110	RESET_MODE_MASK	启动和复位/零件程序结束后控制系统的初始设置
20140	TRAFO_RESET_VALUE	坐标转换的初始设置
22534	TRAFO_CHANGE_M_CODE	坐标转换中切换几何轴时输出的 M 代码
24100	TRAFO_TYPE_1	通道中第 1 转换的定义
24110	TRAFO_AXES_IN_1	第 1 转换的轴指定
24120	TRAFO_GEOAX_ASSIGN_TAB_1	第 1 转换的几何轴指定
24200	TRAFO_TYPE_2	通道中第 2 转换的定义
24210	TRAFO_AXES_IN_2	第 2 转换的轴指定

编号	名称: \$MC_	描述
24220	TRAFO_GEOAX_ASSIGN_TAB_2	第 2 转换的几何轴指定
24300	TRAFO_TYPE_3	通道中第 3 转换的定义
24310	TRAFO_AXES_IN_3	第 3 转换的轴指定
24320	TRAFO_GEOAX_ASSIGN_TAB_3	第 3 转换的几何轴指定
24400	TRAFO_TYPE_4	通道中第 4 转换的定义
24410	TRAFO_AXES_IN_4	第 4 转换的轴指定
24420	TRAFO_GEOAX_ASSIGN_TAB_4	第 4 转换的几何轴指定
24430	TRAFO_TYPE_5	通道中第 5 转换的定义
24432	TRAFO_AXES_IN_5	第 5 转换的轴指定
24434	TRAFO_GEOAX_ASSIGN_TAB_5	第 5 转换的几何轴指定
24440	TRAFO_TYPE_6	通道中第 6 转换的定义
24442	TRAFO_AXES_IN_6	第 6 转换的轴指定
24444	TRAFO_GEOAX_ASSIGN_TAB_6	第 6 转换的几何轴指定
24450	TRAFO_TYPE_7	通道中第 7 转换的定义
24452	TRAFO_AXES_IN_7	第 7 转换的轴指定
24454	TRAFO_GEOAX_ASSIGN_TAB_7	第 7 转换的几何轴指定
24460	TRAFO_TYPE_8	通道中第 8 转换的定义
24462	TRAFO_AXES_IN_8	第 8 转换的轴指定
24464	TRAFO_GEOAX_ASSIGN_TAB_8	第 8 转换的几何轴指定
24900	TRANSMIT_ROT_AX_OFFSET_1	回转轴偏离零点的角度 (第 1 TRANSMIT)
24910	TRANSMIT_ROT_SIGN_IS_PLUS_1	TRANSMIT 下回转轴符号 (第 1 TRANSMIT)
24911	TRANSMIT_POLE_SIDE_FIX_1	极点前/后的工作区域限制, 第 1 转换
24920	TRANSMIT_BASE_TOOL_1	刀具零点到几何轴零点的距离 (第 1 TRANSMIT)
24950	TRANSMIT_ROT_AX_OFFSET_2	回转轴偏离零点的角度 (第 2 TRANSMIT)
24960	TRANSMIT_ROT_SIGN_IS_PLUS_2	TRANSMIT 下回转轴符号 (第 2 TRANSMIT)
24961	TRANSMIT_POLE_SIDE_FIX_2	极点前/后的工作区域限制, 第 2 转换
24970	TRANSMIT_BASE_TOOL_2	刀具零点到几何轴零点的距离 (第 2 TRANSMIT)

6.11.1.2 TRACYL

通道专用机床数据

编号	名称: \$MC_	描述
20110	RESET_MODE_MASK	启动和复位/零件程序结束后控制系统的初始设置
20140	TRAFO_RESET_VALUE	坐标转换的初始设置
20144	TRAFO_MODE_MASK	坐标转换功能的选择
24100	TRAFO_TYPE_1	通道中第 1 转换的定义
24110	TRAFO_AXES_IN_1	第 1 转换的轴指定
24120	TRAFO_GEOAX_ASSIGN_TAB_1	第 1 转换的几何轴指定
24130	TRAFO_INCLUDES_TOOL_1	第 1 转换生效时的刀具处理
24200	TRAFO_TYPE_2	通道中第 2 转换的定义
24210	TRAFO_AXES_IN_2	第 2 转换的轴指定
24220	TRAFO_GEOAX_ASSIGN_TAB_2	第 2 转换的几何轴指定
24230	TRAFO_INCLUDES_TOOL_2	第 2 转换生效时的刀具处理
24300	TRAFO_TYPE_3	通道中第 3 转换的定义
24310	TRAFO_AXES_IN_3	第 3 转换的轴指定
24320	TRAFO_GEOAX_ASSIGN_TAB_3	第 3 转换的几何轴指定
24330	TRAFO_INCLUDES_TOOL_3	第 3 转换生效时的刀具处理
24400	TRAFO_TYPE_4	通道中第 4 转换的定义
24410	TRAFO_AXES_IN_4	第 4 转换的轴指定
24420	TRAFO_GEOAX_ASSIGN_TAB_4	第 4 转换的几何轴指定
24426	TRAFO_INCLUDES_TOOL_4	第 4 转换生效时的刀具处理
24430	TRAFO_TYPE_5	通道中第 5 转换的定义
24432	TRAFO_AXES_IN_5	第 5 转换的轴指定
24434	TRAFO_GEOAX_ASSIGN_TAB_5	第 5 转换的几何轴指定
24436	TRAFO_INCLUDES_TOOL_5	第 5 转换生效时的刀具处理
24440	TRAFO_TYPE_6	通道中第 6 转换的定义
24442	TRAFO_AXES_IN_6	第 6 转换的轴指定
24444	TRAFO_GEOAX_ASSIGN_TAB_6	第 6 转换的几何轴指定
24446	TRAFO_INCLUDES_TOOL_6	第 6 转换生效时的刀具处理

编号	名称: \$MC_	描述
24450	TRAFO_TYPE_7	通道中第 7 转换的定义
24452	TRAFO_AXES_IN_7	第 7 转换的轴指定
24454	TRAFO_GEOAX_ASSIGN_TAB_7	第 7 转换的几何轴指定
24456	TRAFO_INCLUDES_TOOL_7	第 7 转换生效时的刀具处理
24460	TRAFO_TYPE_8	通道中第 8 转换的定义
24462	TRAFO_AXES_IN_8	第 8 转换的轴指定
24464	TRAFO_GEOAX_ASSIGN_TAB_8	第 8 转换的几何轴指定
24466	TRAFO_INCLUDES_TOOL_8	第 8 转换生效时的刀具处理
24470	TRAFO_TYPE_9	通道中第 9 转换的定义
24472	TRAFO_AXES_IN_9	第 9 转换的轴指定
24474	TRAFO_GEOAX_ASSIGN_TAB_9	第 9 转换的几何轴指定
24476	TRAFO_INCLUDES_TOOL_9	第 9 转换生效时的刀具处理
24480	TRAFO_TYPE_10	通道中第 10 转换的定义
24482	TRAFO_AXES_IN_10	第 10 转换的轴指定
24484	TRAFO_GEOAX_ASSIGN_TAB_10	第 10 转换的几何轴指定
24486	TRAFO_INCLUDES_TOOL_10	第 10 转换生效时的刀具处理
24800	TRACYL_ROT_AX_OFFSET_1	回转轴偏离零点位置的角度(第 1 TRACYL)
24808	TRACYL_DEFAULT_MODE_1	TRACYL 模式选择 (第 1 TRACYL)
24810	TRACYL_ROT_SIGN_IS_PLUS_1	TRACYL 下回转轴符号 (第 1 TRACYL)
24820	TRACYL_BASE_TOOL_1	刀具零点到几何轴零点的距离 (第 1 TRACYL)
24850	TRACYL_ROT_AX_OFFSET_2	回转轴偏离零点的角度 (第 2 TRACYL)
24858	TRACYL_DEFAULT_MODE_2	TRACYL 模式选择 (第 2 TRACYL)
24860	TRACYL_ROT_SIGN_IS_PLUS_2	TRACYL 下回转轴符号 (第 2 TRACYL)
24870	TRACYL_BASE_TOOL_2	刀具零点到几何轴零点的距离 (第 2 TRACYL)
22534	TRAFO_CHANGE_M_CODE	坐标转换中切换几何轴时输出的 M 代码

6.11.1.3 TRAANG

通道专用机床数据

编号	名称: \$MC_	描述
20110	RESET_MODE_MASK	启动和复位/零件程序结束后控制系统的初始设置
20140	TRAFO_RESET_VALUE	坐标转换的初始设置
20144	RAFO_MODE_MASK	坐标转换功能的选择
20534	TRAFO_CHANGE_M_CODE	坐标转换中切换几何轴时输出的 M 代码
24100	TRAFO_TYPE_1	通道中第 1 转换的定义
24110	TRAFO_AXES_IN_1	第 1 转换的轴指定
24120	TRAFO_GEOAX_ASSIGN_TAB_1	第 1 转换的几何轴指定
24200	TRAFO_TYPE_2	通道中第 2 转换的定义
24210	TRAFO_AXES_IN_2	第 2 转换的轴指定
24220	TRAFO_GEOAX_ASSIGN_TAB_2	第 2 转换的几何轴指定
24300	TRAFO_TYPE_3	通道中第 3 转换的定义
24310	TRAFO_AXES_IN_3	第 3 转换的轴指定
24320	TRAFO_GEOAX_ASSIGN_TAB_3	第 3 转换的几何轴指定
24400	TRAFO_TYPE_4	通道中第 4 转换的定义
24410	TRAFO_AXES_IN_4	第 4 转换的轴指定
24420	TRAFO_GEOAX_ASSIGN_TAB_4	第 4 转换的几何轴指定
24430	TRAFO_TYPE_5	通道中第 5 转换的定义
24432	TRAFO_AXES_IN_5	第 5 转换的轴指定
24434	TRAFO_GEOAX_ASSIGN_TAB_5	第 5 转换的几何轴指定
24440	TRAFO_TYPE_6	通道中第 6 转换的定义
24442	TRAFO_AXES_IN_6	第 6 转换的轴指定
24444	TRAFO_GEOAX_ASSIGN_TAB_6	第 6 转换的几何轴指定
24450	TRAFO_TYPE_7	通道中第 7 转换的定义
24452	TRAFO_AXES_IN_7	第 7 转换的轴指定
24454	TRAFO_GEOAX_ASSIGN_TAB_7	第 7 转换的几何轴指定
24460	TRAFO_TYPE_8	通道中第 8 转换的定义
24462	TRAFO_AXES_IN_8	第 8 转换的轴指定

编号	名称: \$MC_	描述
24464	TRAFO_GEOAX_ASSIGN_TAB_8	第 8 转换的几何轴指定
24700	TRAANG_ANGLE_1	斜轴角度, 单位度 (第 1 TRAANG)
24710	TRAANG_BASE_TOOL_1	刀具零点到几何轴零点的距离 (第 1 TRAANG)
24720	TRAANG_PARALLEL_VELO_RES_1	用于补偿运动的平行轴速度裕量 (第 1 TRAANG)
24721	TRAANG_PARALLEL_VELO_RES_2	用于补偿运动的平行轴速度裕量 (第 2 TRAANG)
24750	TRAANG_ANGLE_2	斜轴角度, 单位度 (第 2 TRAANG)
24760	TRAANG_BASE_TOOL_2	刀具零点到几何轴零点的距离 (第 2 TRAANG)
24770	TRAANG_PARALLEL_ACCEL_RES_1	用于补偿运动的平行轴加速度裕量 (第 1 TRAANG)
24771	TRAANG_PARALLEL_ACCEL_RES_2	用于补偿运动的平行轴加速度裕量 (第 2 TRAANG)

6.11.1.4 坐标转换级联

通道专用机床数据

编号	名称: \$MC_	描述
24995	TRACON_CHAIN_1	第一坐标转换级联
24996	TRACON_CHAIN_2	第二坐标转换级联
24997	TRACON_CHAIN_3	第三坐标转换级联
24998	TRACON_CHAIN_4	第四坐标转换级联

6.11.1.5 非坐标转换专用机床数据

通道专用机床数据

编号	名称: \$MC_	描述
21110	X_AXIS_IN_OLD_X_Z_PLANE	自动框架定义中的坐标系
21090	MAX_LEAD_ANGLE	定向编程中允许的最大提前角
21092	MAX_TILT_ANGLE	定向编程中允许的最大倾斜角
21100	ORIENTATION_IS_EULER	定向编程时的角度定义

6.11 数据表

6.11.2 信号

6.11.2.1 来自通道的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
转换生效	DB21, ...DBX33.6	DB3300.DBX1.6

测量循环

测量循环的使用说明请见：

文档：

编程手册之测量循环分册

7.2 硬件条件

7.2.1 可使用的测头

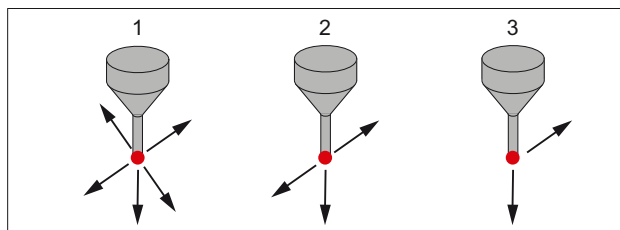
一览

在 SINUMERIK 控制系统中执行“测量”功能时必须使用一个在偏转时提供恒定的无冲撞信号的切换式测头。**不要**使用在偏转时只能提供一个**脉冲**的测头。

测头类型

以下所示测头类型的区别在于可偏转方向的数量：

- 多向测头 (1)
- 双向测头 (2)
- 单向测头 (3)



测头类型

多向测头 (3D)

多向测头可不受限制地用于刀具测量和工件测量。

双向测头

车床需使用双向测头测量工件，而铣床和加工中心则既可使用双向测头，也可使用单向测头。

单向测头

单向测头只能用于铣削中心和加工中心上的工件测量，其限制较低。

主轴位置和单向测头

在铣床和加工中心上使用单向测头的前提条件是：主轴必须能通过 SPOS 功能定位，且能够传输超过 360°的测头切换信号。

测头必须在主轴中经过适当的机械校准，使主轴处于 0°时，可以在以下方向开展测量：

平面	测量方向
G17 (X-Y)	X 轴正向
G18 (Z-X)	Z 轴正向
G19 (Y-Z)	Y 轴正向

说明

由于需要在测量循环中多次通过 SPOS 定位主轴，整个测量过程持续时间较长。

7.3 单通道测量

7.3.1 测量

激活

测量通过零件程序激活。为此需要在程序中编写一个触发事件和一种测量方法。

测量方法分为以下两种：

- **MEAS**：删除剩余行程的测量
示例：
N10 G01 F300 X300 Z200 MEAS=-2
触发事件是第二个测头 (2) 的下降沿 (-)。
- **MEAW**：不删除剩余行程的测量
示例：
N20 G01 F300 X300 Y100 MEAW=1
触发事件是第一个测头 (1) 的上升沿。

系统复位或切换至新程序段时，测量任务终止。

说明

如果在测量程序段中编写了一根几何轴，则会为所有几何轴保存测量结果。

如果在测量程序段中编写了某个参与坐标转换的轴，则会为所有参与该转换的轴保存测量结果。

测头状态

可直接在零件程序和同步动作中查询测头状态。

\$A_PROBE[n]，其中 n = 测头号

\$A_PROBE[n]==1: 测头偏转

\$A_PROBE[n]==0: 测头未偏转

7.3.2 测量结果

读取测量结果

测量结果保存在 NCK 的系统数据中，并可在零件程序中通过系统变量读取。

- **系统变量 \$AC_MEA[编号]**

查询测量任务的状态信号。

[编号]为测头号（1 或 2）

测量开始时此变量会被清除。一旦测头到达触发条件（上升沿或下降沿），此变量会立即置位。这样便可在零件程序中检查测量任务的执行情况。

- **系统变量 \$AA_MM[轴]**

存取机床坐标系中的测量结果。

在零件程序和同步动作中读取。

[轴] 表示被测轴的名称（X、Y、...）。

- **系统变量 \$AA_MW[轴]**

存取工件坐标系中的测量结果。

在零件程序和同步动作中读取。

[轴] 表示被测轴的名称（X、Y、...）。

PLC 信息显示

可通过一个 NC 程序测试测头的功能。

通过诊断菜单“PLC 状态”可在程序结束后检查测量信号。

表格 7-1 测量信号的状态显示

	状态显示
测头 1 偏转	DB10, ...DBX107.0
测头 2 偏转	DB10, ...DBX107.1

接口信号 DB31, ... DBX62.3 可显示轴当前的测量状态：

位 3=1：测量生效

位 3=0：测量未生效

此信号可针对所有测量功能显示，且可在同步动作中通过

- 系统变量 \$AA_MEACTION[轴] 读取。

文档：

功能手册之同步动作分册

7.4 单轴测量

7.4.1 测量

激活

编写单轴测量时，可设定删除剩余行程或不删除剩余行程。测量通过零件程序或同步动作激活。为此须编写测量方法以及最多四个触发事件。通过测量模式可指定触发事件的时序或编写顺序。此外还需针对连续测量编写环形存储器（FIFO）的编号。

测量方法分为以下三种：

- MEASA: 删除剩余行程的测量

示例:

N10 MEASA[X]=(1,1,-1) G01 X100 F100

使用激活的测量系统在模式 1 中测量。触发事件为朝向 X=100 的运行路径上第一个测头 (1) 的上升沿和下降沿。

说明

MEASA 不可在同步动作中编写。

- MEAWA: 不删除剩余行程的测量

示例:

N20 MEAWA[X]=(1,-1,1,-2,2) G01 X100 F100

使用激活的测量系统在模式 1 中测量。触发事件为朝向 X=100 的运行路径上第一个测头 (1) 和第二个测头 (2) 的下降沿及上升沿。

- MEAC (选项): 不删除剩余行程的连续测量

示例:

N30 MEAC[X]=(1,1,-2,2) G01 X100 F100

使用激活的测量系统在模式 1 中测量。测量值保存在第一个 FIFO (1) 中。触发事件为朝向 X=100 的运行路径上第二个测头 (2) 的下降沿和上升沿。

系统复位或切换至新程序段时，测量任务终止。

说明

可在一个程序段中编写 MEASA 和 MEAWA。但如果在写入了 MEAS/MEAW 的程序段中再写入 MEASA/MEAWA，则会触发故障信息。

测量模式

测量模式用于设定触发事件是同时激活还是依次激活，此外还用于定义测量次数。

- 个位（测量模式）
 - 0 = 终止测量任务（例如用于同步动作）
 - 1 = 最多 4 个触发事件可同时激活
 - 2 = 最多 4 个触发事件可依次激活。
第一个触发事件已存在时，发出故障信息
 - 3 = 最多 4 个触发事件可依次激活。
第一个触发事件已存在时，不发出故障信息
- 十位（测量系统）
 - 0/未设置 = 使用激活的测量系统
 - 1 = 使用第一个测量系统
 - 2 = 使用第二个测量系统（前提是存在，否则使用第一个测量系统。此时不会发出故障信息）
 - 3 = 使用第一个和第二个测量系统

说明

测量任务包含两个测量系统时，最多可编写两个触发事件。其中的每个触发事件都会触发两个测量系统的实际值采集。

MEAC 不支持模式 3。

触发事件

触发事件由测头编号和测量信号的触发条件（上升或者下降沿）组成。

- 1 = 测头 1 的上升沿
- 1 = 测头 1 的下降沿
- 2 = 测头 2 的上升沿
- 2 = 测头 2 的下降沿

使用两个测量系统执行测量时，可为每个测头编写最多两个触发事件（上升沿或下降沿）。其中的每个触发事件都会触发两个测头的实际值采集。采用缺省设置 PROFIBUS 报文 391 时，每个触发事件、每个位置控制周期可采集一个测量值。

使用 MEAC 时，可选择 PROFIBUS 报文 395 将该值提高到每个测头、每个脉冲沿、每个位置控制周期 8 个测量值（上升沿和下降沿各 8 个）。相比使用 PROFIBUS 报文 391，此时可实现更高的进给率和转速。

报文选择的更多信息参见章节“报文选择 (页 475)”。

FIFO 变量

单轴测量值是机床坐标系值（MCS）。这些值通过 MEAC 设定的环形存储器（FIFO）保存。在 FIFO 中，测量值按照先进先出原则作为 FIFO 变量输入，例如 \$AC_FIFO1。为测量配置了两个测头时，第二个测头的测量值会单独在之后的 FIFO 中输入。

测量值的数量受到限制。FIFO 的长度可在以下机床数据中配置：

MD28264 LEN_AC_FIFO（FIFO 变量 \$AC_FIFO1 - \$AC_FIFO10 的长度）

可通过零件程序或同步动作从 FIFO 读取测量值。

测量一直生效，直至：

- 编写 MEAC[轴]=(0)
- 一个 FIFO 填满
- 按下 RESET，或到达程序段末尾 M02/M30

文档：

功能手册之同步动作分册，详细说明，

章节：参数 (\$AC_FIFO)

测头状态

可直接在零件程序和同步动作中查询测头状态。

\$A_PROBE[n]，其中 n = 测头号

\$A_PROBE[n]==1：测头已偏转

\$A_PROBE[n]==0：测头未偏转

测头限制

可通过以下用于 PROFIBUS 报文 395 的系统变量启用测头限制：

\$A_PROBE_LIMITED[n]，其中 n = 测头号

\$A_PROBE_LIMITED[n]==1：测头限制生效

\$A_PROBE_LIMITED[n]==0：测头限制无效/复位

系统变量的更多相关信息请见：

文档：

参数手册“系统变量”

7.4.2 报文选择

为 MEAC 单轴测量选择报文

缺省设置下，单轴测量通过 PROFIBUS 报文 391 实现。需要每个触发事件、每个位置控制周期获取更多测量值时，可使用 PROFIBUS 报文 395。

报文选择在 STEP 7 HW Config 中的“DP 从站属性”>“配置”对话框中进行。

此外还需进行以下设置：

- 驱动参数：
CU: p0922 = 395; 报文选择设置
CU: p0684 = 16; 测量方案设置
CU: p0680; 中央测头端子的配置
- PROFIBUS 连接：
MD13211 \$MN_MEAS_CENTRAL_SOURCE = 2（不使用握手的报文连接）

7.4.3 测量结果

读取 MEASA、MEAWA 的测量结果

测量结果保存在 NCK 的系统数据中，并可在零件程序中通过系统变量读取。

- 系统变量 \$AC_MEA[编号]
查询测量任务的状态信号。
[编号]为测头号（1 或 2）
测量开始时此变量会被清除。一旦测头到达触发条件（上升沿或下降沿），此变量会立即置位。这样便可在零件程序中检查测量任务的执行情况。
- 系统变量 \$AA_MM1[轴] 至 \$AA_MM4[轴]
读取机床坐标系中触发信号的测量结果。在零件程序和同步动作中读取。
[轴] 表示被测轴的名称（X、Y、...）。
- 系统变量 \$AA_MW1[轴] 至 \$AA_MW4[轴]
读取工件坐标系中触发信号的测量结果。在零件程序和同步动作中读取。
[轴] 表示被测轴的名称（X、Y、...）。

双测量系统

测量任务包含两个测量系统时，最多可编写两个触发事件。其中的每个触发事件都会触发两个编码器的实际值采集。

一个触发事件

\$AA_MM1[轴] = 触发事件 1, 编码器 1 的测量值

\$AA_MM2[轴] = 触发事件 1, 编码器 2 的测量值

两个触发事件

\$AA_MM1[轴] = 触发事件 1, 编码器 1 的测量值

\$AA_MM2[轴] = 触发事件 1, 编码器 2 的测量值

\$AA_MM3[轴] = 触发事件 2, 编码器 1 的测量值

\$AA_MM4[轴] = 触发事件 2, 编码器 2 的测量值

PLC 信息显示

测头功能测试的更多信息参见章节“测量结果 (页 470)”。

读取 MEAC 的测量结果

所有测量结果均写入预定义的 FIFO。测量值的数量通过机床数据定义（参见“测量 (页 471)”章节）。

- 使用 PROFIBUS 报文 391 且插补周期和位置控制周期的比例 $\leq 8:1$ 时，测量可靠执行。
- 使用 PROFIBUS 报文 395 且插补周期和位置控制周期的比例 $\leq 4:1$ 时，测量可靠执行。
- FIFO 存储器的内容只可读取一次。若需重复使用测量结果，必须将读出的值临时保存在用户数据中。

连续测量

为了实现连续测量，必须通过零件程序循环读取 FIFO 值。从 FIFO 存储器读取和处理测量值的频率必须与 NC 的写入频率相匹配。

存储器中有效条目的数量可在 FIFO 变量中读取。

若希望测量在达到定义的测量值最大数量后结束，必须在程序中显性取消测量任务。

7.5 设置零点、工件测量和刀具测量

编程手册“工作准备”；工艺循环，章节
“回转 – CYCLE800”

7.5.2 工件测量

工件测量

进行工件测量时，测头像刀具一样向夹具中的工件移动。用户可在丰富的测量类型中进行选择，从而以最简单的方式完成车床或铣床上最常见的测量任务。

进行工件测量时，可以测量工件一条边沿、一个拐角或一个钻孔的位置。

为确定工件或钻孔的零点（W），可将实测位置和 WCS 中的设定位置相比。由此得出的偏移可输入到选中的框架中。

变量接口

变量接口包含多种系统变量。

其中分为：

- 输入值
- 输出值

文档：

参数手册“系统变量”

输入值必须通过 HMI 或由循环写入。输出值为计算结果。

文档：

编程手册之测量循环分册

7.5.2.1 输入值

测量类型的有效位

为了确定哪些系统变量对当前测量生效，需要在开展每个测量前将所有变量设置为无效。这通过以下设置实现：

`$AC_MEAS_VALID = 0。`

写入每个输入变量时都会隐性置位 \$AC_MEAS_VALID 中的相应位。若未复位这些有效位，那么这些值在下一次计算时仍保持生效。

说明

机床控制面板复位或 M30（程序结束时复位）后，这些接口不会被复位。

表格 7-2 变量 \$AC_MEAS_VALID 的输入值有效位

位	输入值	含义
0	\$AA_MEAS_POINT1[轴]	所有通道轴的第 1 测量点
1	\$AA_MEAS_POINT2[轴]	所有通道轴的第 2 测量点
2	\$AA_MEAS_POINT3[轴]	所有通道轴的第 3 测量点
3	\$AA_MEAS_POINT4[轴]	所有通道轴的第 4 测量点
4	\$AA_MEAS_SETPPOINT[轴]	边沿、拐角、钻孔的设定位置
5	\$AC_MEAS_WP_SETANGLE	设定工件位置角 α ; $-90 < \varphi < 180$
6	\$AC_MEAS_CORNER_SETANGLE	拐角的设定交角 φ , $0 < \varphi < 180$
7	\$AC_MEAS_T_NUMBER	选择的刀具
7	\$AC_MEAS_D_NUMBER	选择的刀沿
9	\$AC_MEAS_DIR_APPROCH	逼近方向，仅针对边沿测量、槽测量、隔断测量和刀具测量
10	\$AC_MEAS_ACT_PLANE	设置工作平面和进给方向
11	\$AC_MEAS_FRAME_SELECT	计算出的框架进入指定框架
12	\$AC_MEAS_TYPE	工件测量类型
13	\$AC_MEAS_FINE_TRANS	输入平移
14	\$AA_MEAS_SETANGEL[轴]	轴的设定角度
15	\$AA_MEAS_SCALEUNIT	输入值和输出值的单位
16	\$AA_MEAS_TOOL_MASK	刀具设置
17	\$AA_MEAS_P1_COORD	第 1 测量点的坐标系
18	\$AA_MEAS_P2_COORD	第 2 测量点的坐标系
19	\$AA_MEAS_P3_COORD	第 3 测量点的坐标系
20	\$AA_MEAS_P4_COORD	第 4 测量点的坐标系
21	\$AA_MEAS_SET_COORD	设定点的坐标系
22	\$AA_MEAS_CHSFR	系统框架位

位	输入值	含义
23	\$AA_MEAS_NCBFR	全局基本框架位
24	\$AA_MEAS_CHBFR	通道基本框架位
25	\$AA_MEAS_UIFR	数据管理中的可设定框架
26	\$AA_MEAS_PFRAME	不计算可编程框架
27	\$AC_MEAS_INPUT[n]	测量输入参数，长度 n

说明

启用以下设置时，相应测量点的所有轴实际值无效：

\$AC_MEAS_LATCH = 0

测量点

所有通道轴最多有四个测量点可供使用：

类型	输入变量	含义
REAL	\$AA_MEAS_POINT1[轴]	所有通道轴的第 1 测量点
REAL	\$AA_MEAS_POINT2[轴]	所有通道轴的第 2 测量点
REAL	\$AA_MEAS_POINT3[轴]	所有通道轴的第 3 测量点
REAL	\$AA_MEAS_POINT4[轴]	所有通道轴的第 4 测量点

测定的点位置通常作为 WCS 中的实际值 (= 设定值) 提供。一旦为测量点写入轴值，该测量点立即标记为“有效”。每个单独的测量点均可写入或解锁。

有几种测量类型也支持其他坐标系 (BCS、MCS) 中的测量点。可通过以下变量输入在哪个坐标系中测量测量点：

类型	输入变量	含义	数值
INT	\$AA_MEAS_P1_COORD	第 1 测量点的坐标系	0: WCS 为缺省设置 1: BCS 2: MCS 3: ENS 4: WCS_REL 5: ENS_REL
INT	\$AA_MEAS_P2_COORD	第 2 测量点的坐标系	
INT	\$AA_MEAS_P3_COORD	第 3 测量点的坐标系	
INT	\$AA_MEAS_P4_COORD	第 4 测量点的坐标系	
INT	\$AA_MEAS_SET_COORD	设定点的坐标系	

实际值

可为所有轴将测量点设为当前的轴实际值。这些位置基于所选择的坐标系解锁。若未设定坐标系，则会锁存 WCS 中的位置。为此须使用以下变量：

\$AC_MEAS_LATCH[0..3]

下标为 0 至 3，分别对应测量点 1 至 4。此变量赋值为 0 时，对应测量点的所有轴实际值无效。赋值为 1 时，对应测量点的所有轴实际值有效。此变量只可写入。

可通过以下变量写入测量点的单个轴实际值：

类型	系统变量	含义	数值
REAL	\$AA_MEAS_P1_VALID[ax]]	解锁轴的第 1 测量点	0: 轴的测量点无效 1: 获取轴的测量点
REAL	\$AA_MEAS_P2_VALID[ax]]	解锁轴的第 2 测量点	
REAL	\$AA_MEAS_P3_VALID[ax]]	解锁轴的第 3 测量点	
REAL	\$AA_MEAS_P4_VALID[ax]]	解锁轴的第 4 测量点	

可交互使用变量 **\$AC_MEAS_LATCH[0..3]** 和 **\$AA_MEAS_P[1..4]_VALID**。直径编程时会相应地将平面轴考虑在内。

设定值

框架的计算会遵循用户给出的设定值。

表格 7-3 用户设定值的输入值

类型	系统变量	含义
REAL	\$AA_MEAS_SETPoint[ax]	轴的设定位置
REAL	\$AA_MEAS_SETANGLE[ax]	轴的设定角度
INT	\$AA_MEAS_SP_VALID[ax]	1: 轴的设定位置有效/0: 无效
REAL	\$AC_MEAS_WP_SETANGLE	设定工件位置角 α : $-90 < \alpha < 180$

类型	系统变量	含义
REAL	\$AC_MEAS_CORNER_SETANGLE	拐角的设定交角 φ : $0 < \varphi < 180$
INT	\$AC_MEAS_DIR_APPROACH *)	逼近方向 0: +x, 1: -x, 2: +y, 3: -y, 4: +z, 5: -z

*) 仅在边沿测量、槽测量、隔断测量和刀具测量中需要逼近方向。

以下测量点不相关且不会被分析：

- 输入设定工件位置角 α 时：第 2 测量点。
- 输入设定交角 φ 时：第 4 测量点

平面设置

用于定义刀具定向的平面设置。若未设定平面，则会采用生效的平面。

类型	系统变量	数值
INT	\$AC_MEAS_ACT_PLANE	0: G17 工作平面 xy, 进给方向 z 1: G18 工作平面 zx, 进给方向 y 2: G19 工作平面 yz, 进给方向 x

平移

测量工件时，可将平移输入所选框架的精偏分量。为此需要使用变量 \$AC_MEAS_FINE_TRANS。

类型	系统变量	数值
INT	\$AC_MEAS_FINE_TRANS	0: 平移补偿输入粗偏 1: 平移补偿输入精偏

若未对变量 \$AC_MEAS_FINE_TRANS 进行赋值，那么：

- 补偿值输入粗偏并转换至目标框架。通过转换也可能得到平移中的精分量。
- 以下机床数据未预设为 1 时：
MD18600 \$MN_MM_FRAME_FINE_TRANS
总是会将补偿输入粗偏。

计算出的框架

进行工件测量时，计算出的框架会被输入指定框架。

表格 7-4

类型	系统变量	含义
INT	\$AC_MEAS_FRAME_SELECT	工件测量时的框架选择

变量 \$AC_MEAS_FRAME_SELECT 可取以下值

值		含义
0	\$P_SETFRAME	生效的系统框架
1	\$P_PARTFRAME	生效的系统框架
2	\$P_EXTFRAME	生效的系统框架
10..25	\$P_CHBFRAME[0..15]	生效的通道专用基本框架
50..65	\$P_NCBFRAME[0..15]	生效的 NCU 全局基本框架
100..199	\$P_IFFRAME	选择了相应框架时，通过生效的可设定框架进行计算。若选择的框架未生效，则将对应的数据管理框架纳入计算。
500	\$P_TOOLFRAME	生效的系统框架
501	\$P_WPFRAME	生效的系统框架
502	\$P_TRAFRAME	生效的系统框架
503	\$P_PFRAME	生效的当前可编程框架
504	\$P_CYCFRAME	生效的系统框架
505	\$P_RELFRAME (WCS)	生效的系统框架
506	\$P_RELFRAME (ENS)	生效的系统框架
1010..1025	\$P_CHBFRAME[0..15]	生效的通道专用基本框架，包含生效的 G500
1050..1065	\$P_NCBFRAME[0..15]	生效的 NCU 全局基本框架，包含生效的 G500
2000	\$P_SETFR	数据管理中的系统框架
2001	\$P_PARTFR	数据管理中的系统框架
2002	\$P_EXTFR	数据管理中的系统框架
2010..2025	\$P_CHBFR[0..15]	数据管理中的通道专用框架
2050..2065	\$P_NCBFR[0..15]	数据管理中的 NCU 全局基本框架

值		含义
2100..2199	\$P_UIFR[0..99]	数据管理中的可设定框架
2500	\$P_TOOLFR	数据管理中的系统框架
2501	\$P_WPFR	数据管理中的系统框架
2502	\$P_TRAFR	数据管理中的系统框架
2504	\$P_CYCFR	数据管理中的系统框架
2505	\$P_RELFR (WCS)	数据管理中的系统框架
2506	\$P_RELFR (ENS)	数据管理中的系统框架
3010..3025	\$P_CHBFR[0..15]	数据管理中带生效 G500 的通道专用基本框架
3050..3065	\$P_NCBFR[0..15]	数据管理中带生效 G500 的 NCU 全局基本框架

使用 MEASURE() 功能可根据指定的框架计算框架 \$AC_MEAS_FRAME。

值为

0 至 1065 时，借助生效的框架执行计算。

2000 至 3065 时，基于所选择的数据管理中的框架执行计算。测量类型 **14** 和 **15** 不支持选择数据管理中的框架。选择数据管理中的框架时，此框架必须未生效。在此情形下内部计算会假定其为生效状态。

测量点会被转换至所选择的系统，所选择的框架借助整体框架（包括所选择的框架）确定。补偿和激活框架后，实际值设定才生效。

值为

G500 生效时的值（1010..1025，1050..1065，3010..3025，3050..3065），计算框架时会确保 **G500** 在选择此框架后保持生效，从而能到达设定位置。

换算至另一坐标系

若需将一个位置换算为另一个坐标系中的位置，可通过以下变量设定所需框架链的组合：

类型	系统变量	含义	数值
INT	\$AC_MEAS_CHSFR	选择系统框架	位标记对应 MD28082 \$MC_MM_SYSTEM_FRAME_MASK
INT	\$AC_MEAS_NCBFR	选择全局基本框架	位标记（0 ... FFFF）
INT	\$AC_MEAS_CHBFR	选择通道基本框架	位标记（0 ... FFFF）

类型	系统变量	含义	数值
INT	\$AC_MEAS_UIFR	选择可设定框架	0 ... 99
INT	\$AC_MEAS_PFRAME	可编程的框架	0: 包含在内 1: 不包含在内

针对变量中的相应值，系统会读取数据管理框架和建立新的框架链。

说明

若这些变量未置位，则会保留生效的框架。

只须对需要将数据管理框架加入新框架链的变量进行赋值。对于基本框架只可更换所有框架，而不是一个特定框架。通过 \$P_NCBFRMASK 和 \$P_CHBFRMASK 进行的有效修改不计入。

工件和刀具测量的数组变量

以下长度为 n 的数组变量可用于各种测量循环中使用的其他输入参数

类型	系统变量	含义	数值
REAL	\$AC_MEAS_INPUT[n]	测量输入参数	n = 0 ... 9

测量输入参数的控制作用请见测量方案部分。

选择刀具或刀沿

生效刀具的刀具号和刀沿号必须与所选择的刀具一致。选择 T0、D0 时会采用当前生效的刀具。若无刀具生效，则采用通过 T 和 D 选择的刀具。但是只允许选中的刀具为生效状态。

类型	系统变量	含义
INT	\$AC_MEAS_T_NUMBER	选择的刀具
INT	\$AC_MEAS_D_NUMBER	选择的刀沿

使用 3D 测头测量

使用 3D 测头进行测量时，刀具半径已通过测量点补偿，因此不允许将此半径再次引入各种测量的计算。此属性可通过以下机床数据设定：

类型	系统变量	含义
INT	\$AC_MEAS_TOOL_MASK	刀具位置

变量 \$AC_MEAS_TOOL_MASK 可取以下值：

值	含义
0x0:	所有刀具长度均计入（缺省设置）
0x1	计算时不计入刀具半径
0x2	刀具位置处于 x 轴正向（G19）
0x4	刀具位置处于 y 轴正向（G18）
0x8	刀具位置处于 z 轴正向（G17）
0x10	计算时不计入刀具长度
0x20	生效刀具的长度计入位置的坐标系转换
0x40	刀具位置处于 x 轴负向
0x80	刀具位置处于 y 轴负向
0x100	刀具位置处于 z 轴负向
0x200	刀具长度差值取反计入

通过刀具位置和逼近方向可识别出铣刀半径是否计入。若未显性设定逼近方向，可通过选择的平面得出：

工作平面	逼近方向
G17	z 轴负向
G18	y 轴负向
G19	x 轴负向

7.5.2.2 选择测量

通过以下变量选择测量：

类型	系统变量	含义
INT	\$AC_MEAS_TYPE	选测测量类型

变量 \$AC_MEAS_TYPE 可取以下值:

值		含义
0		预设
1	Edge_x	测量 x 棱边
2	Edge_y	测量 y 棱边
3	Edge_z	测量 z 棱边
4	Corner_1	测量拐角 1
5	Corner_2	测量拐角 2
6	Corner_3	测量拐角 3
7	Corner_4	测量拐角 4
8	Hole	测量钻孔
9	Stud	测量轴
10 *	ToolLength	测量刀具长度
11 *	ToolDiameter	测量刀具直径
12	Slot	测量槽
13	Plate	测量隔断
14	Set_Pos	针对几何轴和辅助轴的实际值设定
15	Set_AuxPos	只针对辅助轴的实际值设定
16	Edge_2P	测量斜边
17	Plane_Angles	平面角
18	Plane_Normal	给定设定值的平面角
19	Dimension_1	1 维设定值给定
20	Dimension_2	2 维设定值给定
21	Dimension_3	3 维设定值给定
22 *	ToolMagnifier	ShopTurn: 通过放大镜功能测量刀具长度
23 *	ToolMarkedPos	ShopTurn: 通过标记位置测量刀具长度
24	坐标转换	位置的坐标转换
25	Rectangle	测量矩形
26	Save	备份数据管理框架
27	Restore	恢复数据管理框架
28	锥面车削	平面的叠加旋转

7.5 设置零点、工件测量和刀具测量

* 刀具测量类型

各种测量方法的详细说明请见“工件测量的测量类型”或“刀具测量的测量类型”章节及编程示例。

7.5.2.3 输出值

结果计算

指定了设定位置的情况下，得出的框架会输出到结果框架 \$AC_MEAS_FRAME 中。可在零件程序中读写此框架。结果框架根据所选择的框架计算。

若未选择框架，那么结果框架将决定 WCS 中产生的平移和旋转。通过 PI 服务 _N_SETUDT 和参数类型 7 可将此框架输入到选择的框架中。输入此框架后，结果框架将被清除。

表格 7-5 计算结果的输出值

类型	系统变量	含义
FRAME	\$AC_MEAS_FRAME	结果框架
REAL	\$AC_MEAS_WP_ANGLE	计算出的工件位置角 α
REAL	\$AC_MEAS_CORNER_ANGLE	计算出的交角 ϕ
REAL	\$AC_MEAS_DIAMETER	计算出的直径
REAL	\$AC_MEAS_TOOL_LENGTH	计算出的刀具长度
REAL	\$AC_MEAS_RESULTS[10]	计算结果（取决于 \$AC_MEAS_TYPE）

7.5.2.4 计算方法

激活计算

计算通过 HMI 的 PI 服务 _N_SETUDT 激活。此 PI 服务支持以下参数类型：

类型	含义
1	生效的刀具补偿
2	生效的基本框架
3	生效的可设定框架
4	全局基本框架

类型	含义
5	全局可设定框架
6	计算工件零点或刀具长度
7	激活工件零点（写入对刀）
8	激活外部零点偏移
9	生效的刀架，激活 TCOABS 和 PAROT

复位状态下修改立即可见，停止状态下下一次启动时才会启用框架。

说明

PI 服务只可在复位或停止状态下执行。工件测量计算出的框架会通过参数类型 7 立即激活。测量刀具时不允许通过参数类型 7 发出 PI 服务，因为不必激活零点。

在停止状态下激活

新的 WCS 位置会在停止状态下更新。继续执行零件程序段时，中断程序段的剩余行程会被删除，并会从当前位置逼近下一个程序段的终点位置。

这样一来，在停止状态下也可在 MDA 模式或零件程序中启动主轴，并通过 M0 执行实际值设定、对刀或另一个测量。

测量循环

测量循环中的计算通过预定义功能进行：

INT MEASURE()

MEASURE() 会输出一个结果框架，其可通过 \$AC_MEAS_FRAME 读取：

- 结果是由设定值得出的平移和旋转，换算为所选择的框架。
- 结果框架如下计算：
级联总框架为（测量前）总框架与计算出的平移和旋转级联后的结果。

注意
无预处理限制 MEASURE() 不会触发隐性的预处理停止。 MEASURE() 通过预处理程序段框架运行，因此其须自行决定计算前是否需要预处理停止。

说明

若未选择框架，则不会对计算出的框架进行转换。也就是说，平移和旋转根据设定值和计算出的棱边、拐角、槽位置等得出。多次使用此功能产生的结果总是会叠加至结果框架。

必要时须提前删除结果框架。

信号量变量

测量变量在每个通道中仅出现一次。可通过停止状态和复位状态下的操作执行测量。在停止状态下，操作进程可能与测量循环重叠。以下变量用于防止相互改写：

\$AC_MEAS_SEMA（至测量接口的信号量）

信号量变量 **\$AC_MEAS_SEMA** 由循环

- 在开始处赋值为 1
- 在循环结束处变量重新复位为 0。

此变量赋值为 1 时，HMI 不使用测量接口。

故障信息

若客户未登录，则总是生成汇总故障编号 0xD003。通过 **DIAGN:errCodeSetNrGent** 或 **DIAGN:errCodeSetNrPi** 登录时，**PI_SETUDT** 提供采用以下句法的故障代码：

EX_ERR_PI_REJ_<返回值>，例如：**EX_ERR_PI_REJ_ MEASNOTYPE**

下列返回值通过预定义功能 **MEASURE()** 输出：

表格 7-6 预定义故障信息

号	返回值	含义
0	MEAS_OK	计算正确
1	MEAS_NO_TYPE	未指定类型
2	MEAS_TOOL_ERROR	刀具测定故障
3	MEAS_NO_POINT1	测量点 1 不存在
4	MEAS_NO_POINT2	测量点 2 不存在
5	MEAS_NO_POINT3	测量点 3 不存在
6	MEAS_NO_POINT4	测量点 4 不存在
7	MEAS_NO_SPECPOINT	无参考点存在
8	MEAS_NO_DIR	无逼近方向
9	MEAS_EQUAL_POINTS	测量点相同
10	MEAS_WRONG_ALPHA	Alpha α 错误
11	MEAS_WRONG_PHI	Phi ϕ 错误
12	MEAS_WRONG_DIR	逼近方向错误
13	MEAS_NO_CROSSING	直线不相交
14	MEAS_NO_PLANE	平面不存在
15	MEAS_WRONG_FRAME	未选择框架或选择了错误框架
16	MEAS_NO_MEMORY	存储空间不足
17	MEAS_INTERNAL_ERROR	内部错误

刀具测定错误

出现故障代码 MEAS_TOOL_ERROR 或 EX_ERR_PI_REJ_MEASTOOLERROR 时，系统会将更精确的故障信息连同以下值保存至输出变量 \$AC_MEAS_TOOL_LENGTH

表格 7-7 MEAS_TOOL_ERROR 的预定义故障信息

号	返回值	含义
1	TOOL_NO_BLOCK	无程序段可用于刀具计算
2	TOOL_WRONG_T_NUMBER	T 号错误
3	TOOL_WRONG_D_NUMBER	D 号错误
4	TOOL_EVAL_WRONG_TYPE	刀具不存在

号	返回值	含义
5	TOOL_NO_TOOLCORR_BODY	存储器问题
6	TOOL_DATA_READ_ERROR	读取刀具数据时出错
7	TOOL_NO_TOOL_WITH_TRAFO	转换生效时未选择刀具

7.5.2.5 计算时采用的单位和测量变量

INCH 或 METRISCH 单位

下列输入和输出变量通过英制（INCH）或公制（METRISCH）单位评估：

\$AA_MEAS_POINT1[轴]	第 1 测量点的输入变量
\$AA_MEAS_POINT2[轴]	第 2 测量点的输入变量
\$AA_MEAS_POINT3[轴]	第 3 测量点的输入变量
\$AA_MEAS_POINT4[轴]	第 4 测量点的输入变量
\$AA_MEAS_SETPPOINT[轴]	设定位置的输入变量
\$AC_MEAS_DIAMETER	计算出的直径的输出变量
\$AC_MEAS_TOOL_LENGTH	计算出的刀具长度的输出变量
\$AC_MEAS_RESULTS[n]	计算结果的输出变量

读取或写入输入值和输出值时采用的单位可通过以下输入变量设置

INT \$AC_MEAS_SCALEUNIT	输入和输出变量的单位
0: 单位基于	生效的 G 代码 G70/G700, 采用 INCH 生效的 G 代码 G71/G701, 采用 METRISCH
1: 单位对应	配置, 单位制可通过 OPI 设置 (缺省设置)

不写入变量时, 值 1 总是作为缺省设置。

示例:

缺省单位制为公制:

程序代码	注释
G70	
\$AC_MEAS_POINT1[x] = \$AA_IW[x]	; \$AA_IW[x] 提供缺省单位制
\$AC_MEAS_POINT1[x] = 10	; 10 mm

程序代码	注释
G71	
\$AC_MEAS_POINT1[x] = \$AA_IW[x]	; \$AA_IW[x] 提供缺省单位制
\$AC_MEAS_POINT1[x] = 10	; 10 mm
G700	
\$AC_MEAS_POINT1[x] = \$AA_IW[x]	; \$AA_IW[x] 提供英制值
\$AC_MEAS_POINT1[x] = 10	; 10 英寸
G710	
\$AC_MEAS_POINT1[x] = \$AA_IW[x]	; \$AA_IW[x] 提供公制值
\$AC_MEAS_POINT1[x] = 10	; 10 mm

直径编程

直径编程通过以下机床数据设置：

```
MD20100 $MC_DIAMETER_AX_DEF = "X"      ; 平面轴为 x
MD20150 $MC_GCODE_RESET_VALUES[28] = ; DIAMON
2
MD20360                                ; 刀具长度、框架和
$MC_TOOL_PARAMETER_DEF_MASK =          ; 直径中的实际值
'B1001010'
```

注意以下事项：

- MCS 中的轴位置不会被作为直径值。
- 计算出的刀具补偿和框架分量与生效的 G 代码 DIAMON 或 DIAMOF 无关。
- 测量位置和设定位置根据 DIAMON 读写。
- 框架中的平移被作为平面轴中的直径值。

计算和显示精度

单位为毫米、英寸或度的位置值以小数点后 6 位精确计算和显示。

7.5.2.6 诊断

可为测量接口使用以下诊断功能：

- 一个问题重复出现时，将一条记录写入 `/_N_MPF_DIR/_N_MEAS_DUMP_MPF` 文件。
- 用户首先需要在 `/_N_MPF_DIR` 目录中创建名称为 `_N_MEAS_DUMP_MPF` 的空文件，才能开始记录。
- 文件的内容一直保留，直至通过 `$AC_MEAS_VALID = 0` 清除。

出于运行时间方面的原因，应只对已出现过一次的问题启用跟踪功能。

7.5.3 工件测量类型

7.5.3.1 测量棱边（测量类型 1、2、3）

测量 x 棱边（`$AC_MEAS_TYPE = 1`）

夹紧工件的棱边如下测量：逼近此棱边，然后通过一把已知的刀具执行测量。

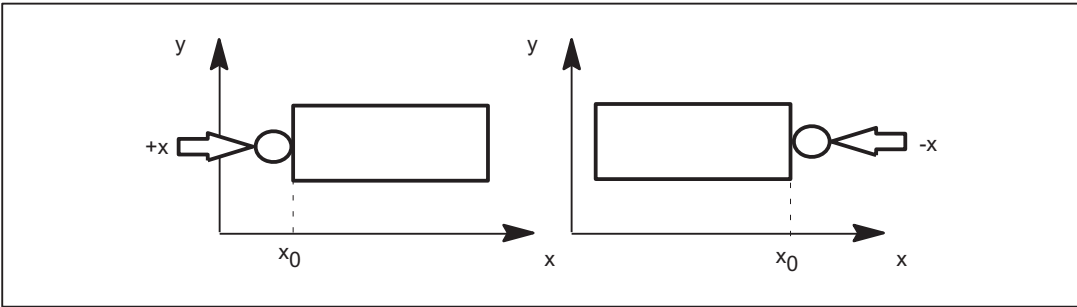


图 7-1 x 棱边

针对测量类型 1，系统会分析下列变量值：

输入变量	含义
<code>\$AC_MEAS_VALID</code>	输入变量的有效位
<code>\$AA_MEAS_POINT1[轴]</code>	所有通道轴的第 1 测量点
<code>\$AA_MEAS_SETPOINT[轴]</code>	x 棱边的设定位置 *
<code>\$AC_MEAS_DIR_APPROACH</code>	0: +x, 1: -x
<code>\$AC_MEAS_ACT_PLANE</code>	未设定时将采用当前生效的平面， 仅在 G17 和 G18 中才考虑刀具半径 *

输入变量	含义
\$AC_MEAS_FINE_TRANS	0: 粗偏移, 1: 精偏移 '
\$AC_MEAS_FRAME_SELECT	未设定时将计算叠加框架 *
\$AC_MEAS_T_NUMBER	未设定时将采用当前生效的 T (T0) *
\$AC_MEAS_D_NUMBER	未设定时将采用当前生效的 D (D0) *
\$AC_MEAS_TYPE	1

* 可选

针对测量类型 1, 系统会写入以下输出变量:

输出变量	含义
\$AC_MEAS_FRAME	结果框架, 包含平移
\$AC_MEAS_RESULTS[0]	被测棱边的位置

示例

测量 x 棱边

程序代码	注释
DEF INT RETVAL	
DEF FRAME TMP	
 \$TC_DP1[1,1]=120	 ; 类型
\$TC_DP2[1,1]=20	; 0
\$TC_DP3[1,1]= 10	; (z) 长度补偿矢量
\$TC_DP4[1,1]=0	; (y)
\$TC_DP5[1,1]= 0	; (x)
\$TC_DP6[1,1]= 2	; 半径
 T1 D1	
g0 x0 y0 z0 f10000	
G54	
 	 ; 测量 x 棱边
\$AC_MEAS_VALID = 0	; 将所有输入值设置为无效
 g1 x-1 y-3	 ; 逼近第 1 测量点
 \$AA_MEAS_POINT1[x] = \$AA_IW[x]	
\$AA_MEAS_POINT1[y] = \$AA_IW[y]	
\$AA_MEAS_POINT1[z] = \$AA_IW[z]	

程序代码	注释
\$AC_MEAS_DIR_APPROACH = 0	; 设置逼近方向 +x
\$AA_MEAS_SETPOINT[x] = 0	; 设置棱边的设定位置
\$AA_MEAS_SETPOINT[y] = 0	
\$AA_MEAS_SETPOINT[z] = 0	
\$AC_MEAS_ACT_PLANE = 0	; 测量平面为 G17
\$AC_MEAS_FRAME_SELECT = 101	; 选择框架 - IFRAME
\$AC_MEAS_T_NUMBER = 1	; 选择刀具
\$AC_MEAS_D_NUMBER = 1	
\$AC_MEAS_TYPE = 1	; 设置“x 棱边”测量类型
RETVAL = MEASURE()	; 开始测量
if RETVAL <> 0	
setal(61000 + RETVAL)	
endif	
\$P_IFRAME = \$AC_MEAS_FRAME	
\$P_UIFR[1] = \$P_IFRAME	; 写入数据管理中的系统框架
g1 x0 y0	; 逼近棱边
m30	

测量 y 棱边 (\$AC_MEAS_TYPE = 2)

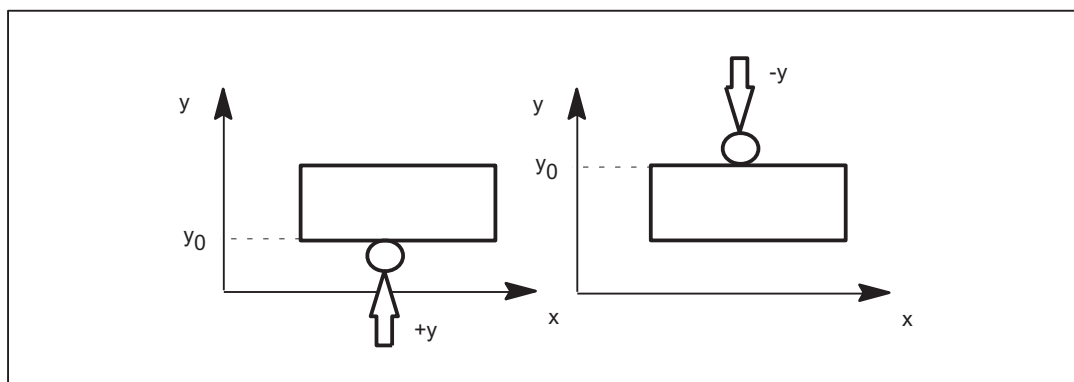


图 7-2 y 棱边

针对测量类型 2，系统会分析下列变量值：

输入变量	含义
\$AC_MEAS_VALID	输入变量的有效位
\$AA_MEAS_POINT1[轴]	所有通道轴的第 1 测量点
\$AA_MEAS_SETPPOINT[轴]	y 棱边的设定位置 *
\$AC_MEAS_DIR_APPROACH	2: +y, 3: -y
\$AC_MEAS_ACT_PLANE	未设定时将采用当前生效的平面， 仅在 G17 和 G19 中才考虑刀具半径 *
\$AC_MEAS_FINE_TRANS	0: 粗偏移, 1: 精偏移 *
\$AC_MEAS_FRAME_SELECT	未设定时将计算叠加框架 *
\$AC_MEAS_T_NUMBER	未设定时将采用当前生效的 T (T0) *
\$AC_MEAS_D_NUMBER	未设定时将采用当前生效的 D (D0) *
\$AC_MEAS_TYPE	2

* 可选

针对测量类型 2，系统会写入以下输出变量：

输出变量	含义
\$AC_MEAS_FRAME	结果框架，包含平移
\$AC_MEAS_RESULTS[0]	被测棱边的位置

测量 z 棱边 (\$AC_MEAS_TYPE = 3)

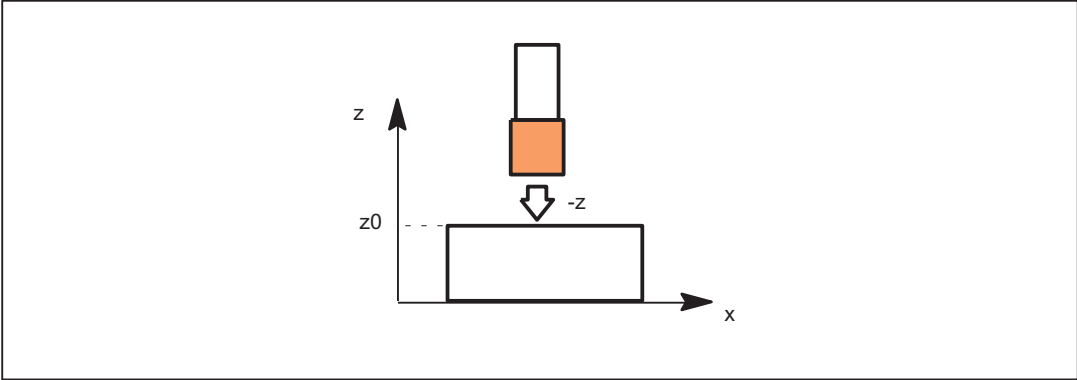


图 7-3 z 棱边

针对测量类型 3，系统会分析下列变量值：

输入变量	含义
\$AC_MEAS_VALID	输入变量的有效位
\$AA_MEAS_POINT1[轴]	所有通道轴的第 1 测量点
\$AA_MEAS_SETPPOINT[轴]	z 棱边的设定位置 *
\$AC_MEAS_DIR_APPROACH	4: +z, 5: -z
\$AC_MEAS_ACT_PLANE	未设定时将采用当前生效的平面， 仅在 G18 和 G19 中才考虑刀具半径 *
\$AC_MEAS_FINE_TRANS	0: 粗偏移, 1: 精偏移 *
\$AC_MEAS_FRAME_SELECT	未设定时将计算叠加框架 *
\$AC_MEAS_T_NUMBER	未设定时将采用当前生效的 T (T0) *
\$AC_MEAS_D_NUMBER	未设定时将采用当前生效的 D (D0) *
\$AC_MEAS_TYPE	3

* 可选

针对测量类型 3，系统会写入以下输出变量：

输出变量	含义
\$AC_MEAS_FRAME	结果框架，包含平移
\$AC_MEAS_RESULTS[0]	被测棱边的位置

7.5.3.2 测量角（测量类型 4、5、6、7）

测量拐角 C1 - C4 (\$AC_MEAS_TYPE = 4, 5, 6, 7)

通过逼近 4 个测量点 P1 至 P4 来测定唯一的拐角。交角 ϕ 已知时只需要 3 个测量点。

若交角 ϕ 和工件位置角 α 已知，则只需要 P1 和 P3 这 2 个测量点。

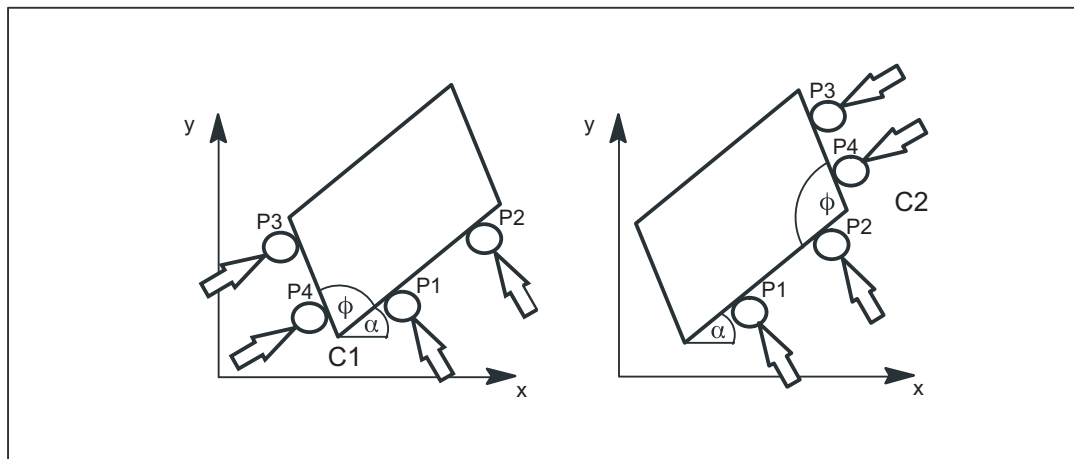


图 7-4 拐角 C1、拐角 C2

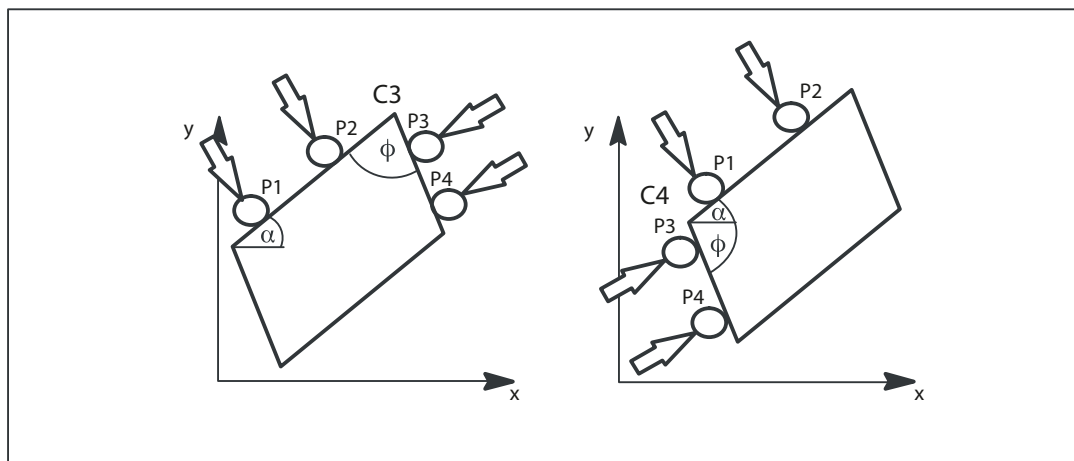


图 7-5 拐角 C3、拐角 C4

针对测量类型 4 至 7，系统会分析下列变量值：

输入变量	含义
\$AC_MEAS_VALID	输入变量的有效位
\$AA_MEAS_POINT1[轴]	测量点 1

输入变量	含义
\$AA_MEAS_POINT2[轴]	测量点 2，采用 \$AC_MEAS_WP_SETANGLE 时不相关
\$AA_MEAS_POINT3[轴]	测量点 3
\$AA_MEAS_POINT4[轴]	测量点 4，采用 \$AC_MEAS_CORNER_SETANGLE 时不相关
\$AA_MEAS_WP_SETANGLE	设定工件位置角 *
\$AA_MEAS_CORNER_SETANGLE	设定交角 *
\$AA_MEAS_SETPoint[轴]	拐角的设定位置 *
\$AC_MEAS_ACT_PLANE	未设定时将采用当前生效的平面 *
\$AC_MEAS_FINE_TRANS	0: 粗偏移, 1: 精偏移 *
\$AC_MEAS_FRAME_SELECT	未设定时将计算叠加框架 *
\$AC_MEAS_T_NUMBER	未设定时将采用当前生效的 T (T0) *
\$AC_MEAS_D_NUMBER	未设定时将采用当前生效的 D (D0) *
\$AC_MEAS_INPUT[0]	未设定时测量外角 * =0: 外角测量 =1: 内角测量
\$AC_MEAS_TYPE	4, 5, 6, 7

* 可选

针对测量类型 4 至 7，系统会写入下列变量值：

输出变量	含义
\$AC_MEAS_FRAME	结果框架，包含平移和旋转
\$AC_MEAS_WP_ANGLE	计算出的工件位置角
\$AC_MEAS_CORNER_ANGLE	计算出的交角
\$AC_MEAS_RESULTS[0]	计算出的拐角点的横坐标
\$AC_MEAS_RESULTS[1]	计算出的拐角点的纵坐标
\$AC_MEAS_RESULTS[2]	计算出的拐角点的垂直坐标

示例

测量拐角 C1，采用 3 个测量点（P1、P3 和 P4），交角 ϕ 已知（90°），工件位置角 α 未知。

程序代码	注释
DEF INT RETVAL	
DEF FRAME TMP	
\$TC_DP1[1,1]=120	; 类型
\$TC_DP2[1,1]=20	; 0
\$TC_DP3[1,1]= 10	; (z) 长度补偿矢量
\$TC_DP4[1,1]=0	; (y)
\$TC_DP5[1,1]= 0	; (x)
\$TC_DP6[1,1]= 2	; 半径
T1 D1	
g0 x0 y0 z0 f10000	
G54	
\$P_CHBFRAME[0] = crot(z,45)	
\$P_IFRAME[x,tr] = -sin(45)	
\$P_IFRAME[y,tr] = -sin(45)	
\$P_PFRAME[z,tr] = -45	
	; 通过 3 个测量点测量拐角
\$AC_MEAS_VALID = 0	; 将所有输入值设置为无效
g1 x-1 y-3	; 逼近第 1 测量点
\$AC_MEAS_LATCH[0] = 1	; 解锁测量点 P1
g1 x-4 y4	; 逼近第 3 测量点
\$AC_MEAS_LATCH[2] = 1	; 解锁测量点 P3
g1 x-4 y1	; 逼近第 4 测量点
\$AC_MEAS_LATCH[3] = 1	; 解锁测量点 P4
\$AA_MEAS_SETPOINT[x] = 0	; 将拐角设定位置设为 (0, 0, 0)
\$AA_MEAS_SETPOINT[y] = 0	
\$AA_MEAS_SETPOINT[z] = 0	
\$AC_MEAS_CORNER_SETANGLE = 90	; 给定设定交角 ϕ
\$AC_MEAS_ACT_PLANE = 0	; 测量平面为 G17
\$AC_MEAS_FRAME_SELECT = 0	; 选择框架 - SETFRAME

程序代码	注释
\$AC_MEAS_T_NUMBER = 1	； 选择刀具
\$AC_MEAS_D_NUMBER = 1	
\$AC_MEAS_TYPE = 4	； 将测量类型设置为“拐角 1”
RETVAL = MEASURE()	； 开始测量
if RETVAL <> 0	
setal(61000 + RETVAL)	
endif	
if \$AC_MEAS_CORNER_ANGLE <> 90	； 查询已知设定交角 ϕ
setal(61000 + \$AC_MEAS_CORNER_ANGLE)	
endif	
\$P_SETFRAME = \$AC_MEAS_FRAME	
\$P_SETFR = \$P_SETFRAME	； 写入数据管理中的系统框架
g1 x0 y0	； 逼近拐角
g1 x10	； 沿直角运行
y10	
x0	
y0	
m30	

7.5.3.3 测量钻孔（测量类型 8）

确定一个钻孔（\$AC_MEAS_TYPE = 8）所需的测量点

确定钻孔的圆心和直径需要 3 个测量点。这三个点必须为不同的点。设定了 4 个点时，系统会根据最小误差平方法对圆弧进行调整。圆弧的确定依据以下原则：使圆弧的点距离平方总和最小化。可读取调整质量。

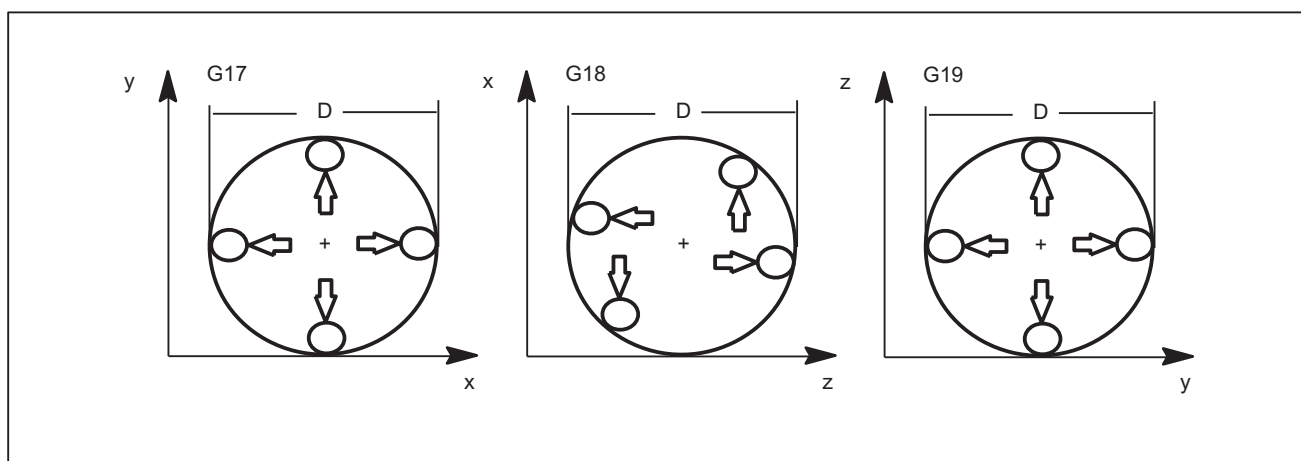


图 7-6 钻孔

针对测量类型 8，系统会分析下列变量值：

输入变量	含义
\$AC_MEAS_VALID	输入变量的有效位
\$AA_MEAS_POINT1[轴]	测量点 1
\$AA_MEAS_POINT2[轴]	测量点 2
\$AA_MEAS_POINT3[轴]	测量点 3
\$AA_MEAS_POINT4[轴]	设定此变量时将通过 4 个点确定中心 *
\$AA_MEAS_SETPPOINT[轴]	钻孔中心的设定位置 *
\$AC_MEAS_ACT_PLANE	未设定时将采用当前生效的平面 *
\$AC_MEAS_FINE_TRANS	0: 粗偏移, 1: 精偏移 *
\$AC_MEAS_FRAME_SELECT	未设定时将计算叠加框架 *
\$AC_MEAS_T_NUMBER	未设定时将采用当前生效的 T (T0) *

7.5 设置零点、工件测量和刀具测量

输入变量	含义
\$AC_MEAS_D_NUMBER	未设定时将采用当前生效的 D（D0）*
\$AC_MEAS_TYPE	8

* 可选

针对测量类型 8，系统会写入以下输出变量：

输出变量	含义
\$AC_MEAS_FRAME	结果框架，包含平移
\$AC_MEAS_DIAMETER	钻孔直径
\$AC_MEAS_RESULTS[0]	计算出的中心的横坐标
\$AC_MEAS_RESULTS[1]	计算出的中心的纵坐标
\$AC_MEAS_RESULTS[2]	计算出的中心的垂直坐标
\$AC_MEAS_RESULTS[3]	圆弧调整质量：距离平方和

示例

测量钻孔

程序代码	注释
DEF INT RETVAL DEF FRAME TMP \$TC_DP1[1,1]=120 \$TC_DP2[1,1]=20 \$TC_DP3[1,1]= 10 \$TC_DP4[1,1]=0 \$TC_DP5[1,1]= 0 \$TC_DP6[1,1]= 2 T1 D1 g0 x0 y0 z0 f10000 G54 \$AC_MEAS_VALID = 0 g1 x-3 y0 \$AA_MEAS_POINT1[x] = \$AA_IW[x] \$AA_MEAS_POINT1[y] = \$AA_IW[y]	 ; 类型 ; 0 ; (z) 长度补偿矢量 ; (y) ; (x) ; 半径 : 测量钻孔 ; 将所有输入值设置为无效 ; 逼近第 1 测量点

7.5 设置零点、工件测量和刀具测量

程序代码	注释
\$AA_MEAS_POINT1[z] = \$AA_IW[z]	
g1 x0 y3	; 逼近第 2 测量点
\$AA_MEAS_POINT2[x] = \$AA_IW[x]	
\$AA_MEAS_POINT2[y] = \$AA_IW[y]	
\$AA_MEAS_POINT2[z] = \$AA_IW[z]	
g1 x3 y0	; 逼近第 3 测量点
\$AA_MEAS_POINT3[x] = \$AA_IW[x]	
\$AA_MEAS_POINT3[y] = \$AA_IW[y]	
\$AA_MEAS_POINT3[z] = \$AA_IW[z]	
\$AA_MEAS_SETPPOINT[x] = 0	; 设置中心的设定位置
\$AA_MEAS_SETPPOINT[y] = 0	
\$AA_MEAS_SETPPOINT[z] = 0	
\$AC_MEAS_ACT_PLANE = 0	; 测量平面为 G17
\$AC_MEAS_FRAME_SELECT = 0	; 选择框架 - SETFRAME
\$AC_MEAS_T_NUMBER = 1	; 选择刀具
\$AC_MEAS_D_NUMBER = 1	
\$AC_MEAS_TYPE = 8	; 将测量类型设置为“钻孔”
RETVAL = MEASURE()	; 开始测量
if RETVAL <> 0	
setal(61000 + RETVAL)	
endif	
if \$AC_MEAS_DIAMETER <> 10	; 查询已知直径
setal(61000 + \$AC_MEAS_WP_ANGLE)	
endif	
\$P_SETFRAME = \$AC_MEAS_FRAME	
\$P_SETFR = \$P_SETFRAME	; 写入数据管理中的系统框架
g1 x-3 y0	; 逼近 P1
g2 I = \$AC_MEAS_DIAMETER / 2	; 基于圆心沿钻孔运行
m30	

7.5.3.4 测量轴（测量类型 9）

确定一根轴（\$AC_MEAS_TYPE = 9）所需的测量点

确定轴的圆心和直径需要 3 个测量点。这三个点必须为不同的点。设定了 4 个点时，系统会根据最小误差平方法对圆弧进行调整。圆弧的确定依据以下原则：使圆弧的点距离平方总和最小化。可读取调整质量。

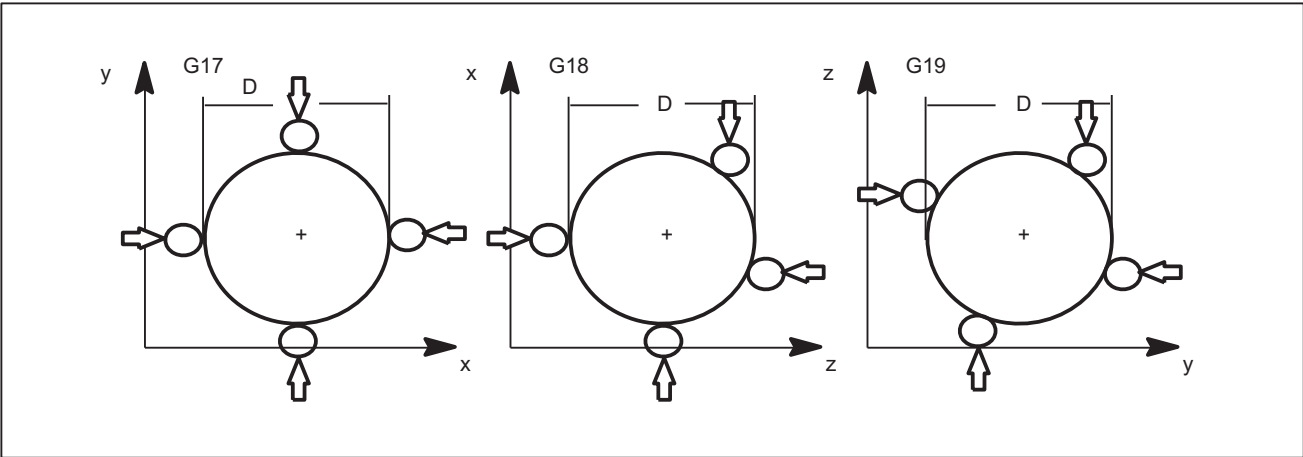


图 7-7 轴

针对测量类型 9，系统会分析下列变量值：

输入变量	含义
\$AC_MEAS_VALID	输入变量的有效位
\$AA_MEAS_POINT1[轴]	测量点 1
\$AA_MEAS_POINT2[轴]	测量点 2
\$AA_MEAS_POINT3[轴]	测量点 3
\$AA_MEAS_POINT4[轴]	设定此变量时将通过 4 个点确定中心 *
\$AA_MEAS_SETPoint[轴]	轴圆心的设定位置 *
\$AC_MEAS_ACT_PLANE	未设定时将采用当前生效的平面 *
\$AC_MEAS_FINE_TRANS	0: 粗偏移, 1: 精偏移 *
\$AC_MEAS_FRAME_SELECT	未设定时将计算叠加框架 *
\$AC_MEAS_T_NUMBER	未设定时将采用当前生效的 T (T0) *

输入变量	含义
\$AC_MEAS_D_NUMBER	未设定时将采用当前生效的 D (D0) *
\$AC_MEAS_TYPE	9

* 可选

针对测量类型 9，系统会写入以下输出变量：

输出变量	含义
\$AC_MEAS_FRAME	结果框架，包含平移
\$AC_MEAS_DIAMETER	轴直径
\$AC_MEAS_RESULTS[0]	计算出的中心的横坐标
\$AC_MEAS_RESULTS[1]	计算出的中心的纵坐标
\$AC_MEAS_RESULTS[2]	计算出的中心的垂直坐标
\$AC_MEAS_RESULTS[3]	圆弧调整质量：距离平方和

7.5.3.5 测量槽（测量类型 12）

确定槽位置（\$AC_MEAS_TYPE = 12）所需的测量点

槽通过逼近两个外边沿或内边沿测量。可将槽中心设置到设定位置。通过逼近方向的分量确定槽位置。

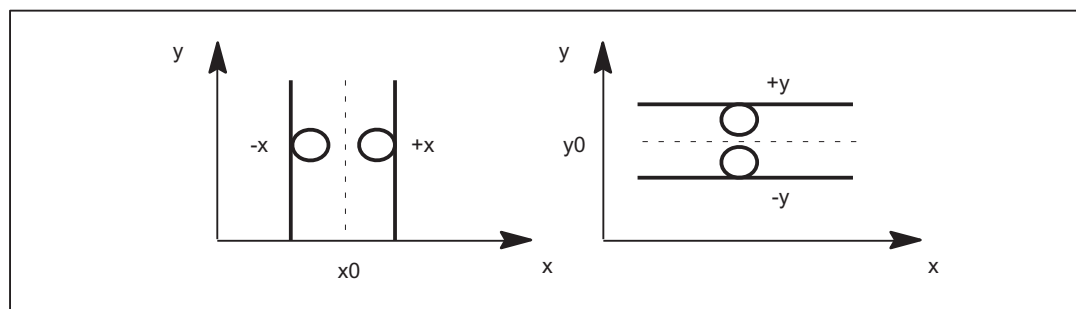


图 7-8 槽

针对测量类型 12，系统会分析下列变量值：

输入变量	含义
\$AC_MEAS_VALID	输入变量的有效位
\$AA_MEAS_POINT1[轴]	测量点 1

输入变量	含义
\$AA_MEAS_POINT2[轴]	测量点 2
\$AA_MEAS_SETPoint[轴]	槽中心的设定位置 *
\$AC_MEAS_DIR_APPROACH	0: +x, 1: -x, 2: +y, 3: -y, 4: +z, 5: -z
\$AC_MEAS_ACT_PLANE	未设定时将采用当前生效的平面 *
\$AC_MEAS_FINE_TRANS	0: 粗偏移, 1: 精偏移 *
\$AC_MEAS_FRAME_SELECT	未设定时将计算叠加框架 *
\$AC_MEAS_T_NUMBER	未设定时将采用当前生效的 T (T0) *
\$AC_MEAS_D_NUMBER	未设定时将采用当前生效的 D (D0) *
\$AC_MEAS_INPUT[0]	测量凹陷时第 2 测量点的逼近方向。坐标必须与第 1 测量点的逼近方向相同。 * 0: +x, 1: -x, 2: +y, 3: -y, 4: +z, 5: -z
\$AC_MEAS_TYPE	12

* 可选

针对测量类型 12，系统会写入以下输出变量：

输出变量	含义
\$AC_MEAS_FRAME	结果框架，包含平移
\$AC_MEAS_RESULTS[0]	计算出的槽中心的位置 (x0, y0 或 z0)
\$AC_MEAS_RESULTS[1]	逼近方向上的槽宽

示例

测量槽，逼近方向 x

程序代码	注释
DEF INT RETVAL	
DEF FRAME TMP	
 \$TC_DP1[1,1]=120	 ; 类型
\$TC_DP2[1,1]=20	; 0
\$TC_DP3[1,1]= 10	; (z) 长度补偿矢量
\$TC_DP4[1,1]=0	; (y)
\$TC_DP5[1,1]= 0	; (x)
\$TC_DP6[1,1]= 2	; 半径
 T1 D1	

7.5 设置零点、工件测量和刀具测量

程序代码	注释
g0 x0 y0 z0 f10000 G54	
\$P_CHBFRAME[0] = crot(z,45) \$P_IFRAME[x,tr] = -sin(45) \$P_IFRAME[y,tr] = -sin(45) \$P_PFRAME[z,rt] = -45	
\$AC_MEAS_VALID = 0	; 测量槽 ; 将所有输入值设置为无效
g1 x-2 \$AA_MEAS_POINT1[x] = \$AA_IW[x] \$AA_MEAS_POINT1[y] = \$AA_IW[y] \$AA_MEAS_POINT1[z] = \$AA_IW[z]	; 逼近第 1 测量点
g1 x4 \$AA_MEAS_POINT2[x] = \$AA_IW[x] \$AA_MEAS_POINT2[y] = \$AA_IW[y] \$AA_MEAS_POINT2[z] = \$AA_IW[z]	; 逼近第 2 测量点
\$AA_MEAS_SETPOINT[x] = 0 \$AA_MEAS_SETPOINT[y] = 0 \$AA_MEAS_SETPOINT[z] = 0	; 设置槽中心的设定位置
\$AC_MEAS_DIR_APPROACH = 0 \$AC_MEAS_ACT_PLANE = 0 \$AC_MEAS_FRAME_SELECT = 0	; 设置逼近方向 +x ; 测量平面为 G17 ; 选择框架 - SETFRAME
\$AC_MEAS_T_NUMBER = 1 \$AC_MEAS_D_NUMBER = 1	; 选择刀具
\$AC_MEAS_TYPE = 12	; 将测量类型设置为“槽”
RETVAL = MEASURE()	; 开始测量
if RETVAL <> 0 setal(61000 + RETVAL) endif	
\$P_SETFRAME = \$AC_MEAS_FRAME \$P_SETFR = \$P_SETFRAME	; 写入数据管理中的系统框架
g1 x0 y0	; 逼近槽中心

程序代码	注释
m30	

7.5.3.6 测量隔断（测量类型 13）

确定隔断位置（\$AC_MEAS_TYPE = 13）所需的测量点

隔断通过逼近两个外边沿或内边沿测量。 可将隔断中心设置到设定位置。 通过逼近方向的分量确定隔断位置。

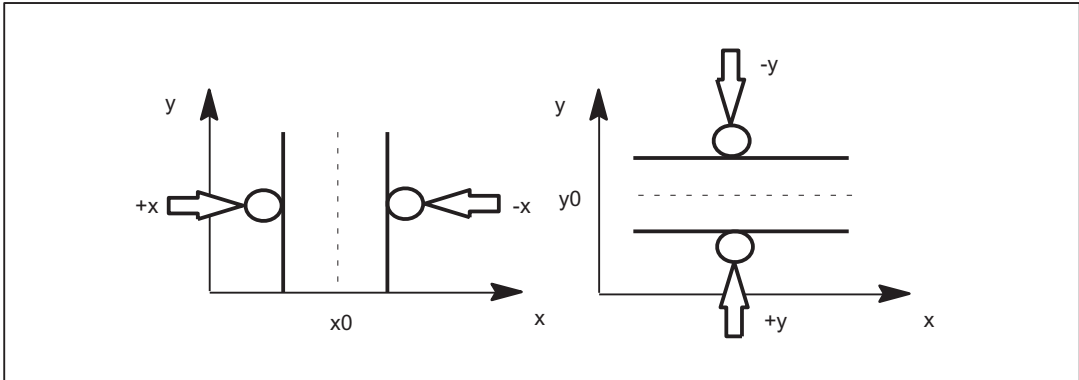


图 7-9 隔断

针对测量类型 13，系统会分析下列变量值：

输入变量	含义
\$AC_MEAS_VALID	输入变量的有效位
\$AA_MEAS_POINT1[轴]	测量点 1
\$AA_MEAS_POINT2[轴]	测量点 2
\$AA_MEAS_SETPPOINT[轴]	隔断中心的设定位置 *
\$AC_MEAS_DIR_APPROACH	0: +x, 1: -x, 2: +y, 3: -y, 4: +z, 5: -z
\$AC_MEAS_ACT_PLANE	未设定时将采用当前生效的平面 *
\$AC_MEAS_FINE_TRANS	0: 粗偏移, 1: 精偏移 *
\$AC_MEAS_FRAME_SELECT	未设定时将计算叠加框架 *
\$AC_MEAS_T_NUMBER	未设定时将采用当前生效的 T (T0) *
\$AC_MEAS_D_NUMBER	未设定时将采用当前生效的 D (D0) *

输入变量	含义
\$AC_MEAS_INPUT[0]	测量凹陷时第 2 测量点的逼近方向。坐标必须与第 1 测量点的逼近方向相同。* 0: +x, 1: -x, 2: +y, 3: -y, 4: +z, 5: -z
\$AC_MEAS_TYPE	13

* 可选

针对测量类型 13，系统会写入以下输出变量：

输出变量	含义
\$AC_MEAS_FRAME	结果框架，包含平移
\$AC_MEAS_RESULTS[0]	计算出的隔断中心的位置 (x0, y0 或 z0)
\$AC_MEAS_RESULTS[1]	逼近方向上的隔断宽度

7.5.3.7 测量几何轴和辅助轴（测量类型 14、15）

几何轴和辅助轴的实际值设定（\$AC MEAS TYPE = 14）

此测量类型通过 HMI 操作界面使用。

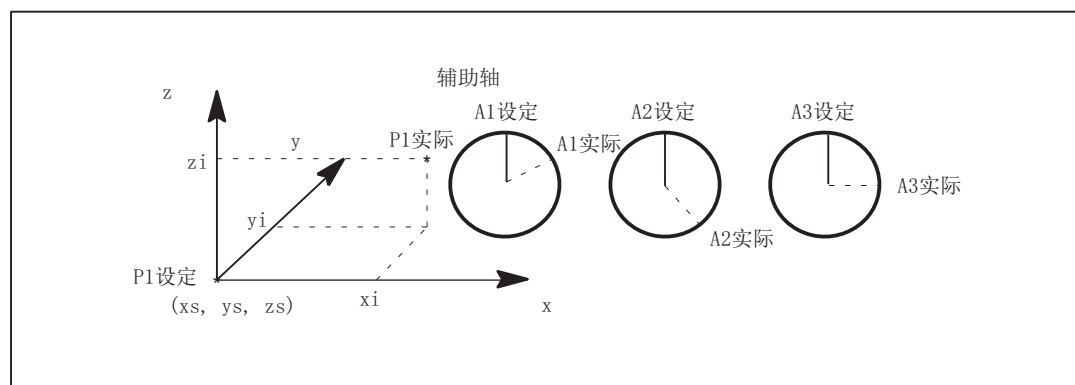


图 7-10 实际值设定

针对测量类型 14，系统会分析下列变量值：

输入变量	含义
\$AC_MEAS_VALID	输入变量的有效位
\$AA_MEAS_POINT1[轴]	轴的实际值

输入变量	含义
\$AA_MEAS_SETPOINT[轴]	单个轴的设定位置 *
\$AC_MEAS_ACT_PLANE	未设定时将采用当前生效的平面 *
\$AC_MEAS_FINE_TRANS	0: 粗偏移, 1: 精偏移 *
\$AC_MEAS_FRAME_SELECT	未设定时将计算叠加框架 *
\$AC_MEAS_T_NUMBER	未设定时将采用当前生效的 T (T0) *
\$AC_MEAS_D_NUMBER	未设定时将采用当前生效的 D (D0) *
\$AC_MEAS_TYPE	14

* 可选

针对测量类型 14，系统会写入以下输出变量：

输出变量	含义
\$AC_MEAS_FRAME	结果框架，包含平移

示例

相对坐标系中的参考点设置。

程序代码	注释
DEF INT RETVAL	
 T1 D1	 ; 激活测头
G54	; 激活所有框架和 G54
 TRANS x=10	 ; WCS 和 ENS 间的平移
G0 x0 f10000	; WKS(x) = 0; ENS(x) = 10
 \$AC_MEAS_VALID = 0	 ; 将所有输入变量设置为无效
\$AC_MEAS_TYPE = 14	; 用于实际值设定的测量类型
\$AC_MEAS_ACT_PLANE = 0	; 测量平面为 G17
 \$AC_MEAS_P1_COORD = 5	 ; ENS_REL 针对第 1 测量点
\$AC_MEAS_LATCH[0] = 1	; 解锁所有轴位置
 \$AC_MEAS_SET_COORD = 5	 ; 设定位置相对于 ENS
\$AA_MEAS_SETPOINT[x] = 0	; 相对 ENS 坐标系中的设定位置
\$AC_MEAS_FRAME_SELECT = 2505	; \$P_RELFR
 RETVAL = MEASURE()	 ; 计算 \$P_RELFR; PI SETUDT(6)
IF RETVAL <> 0 GOTOF ERROR	

程序代码	注释
ENDIF \$ P_RELFR = \$AC_MEAS_FRAME	; 激活; PI SETUDT(7)

仅为辅助轴设定实际值 (\$AC MEAS TYPE = 15)

此测量类型通过 HMI 操作界面使用。

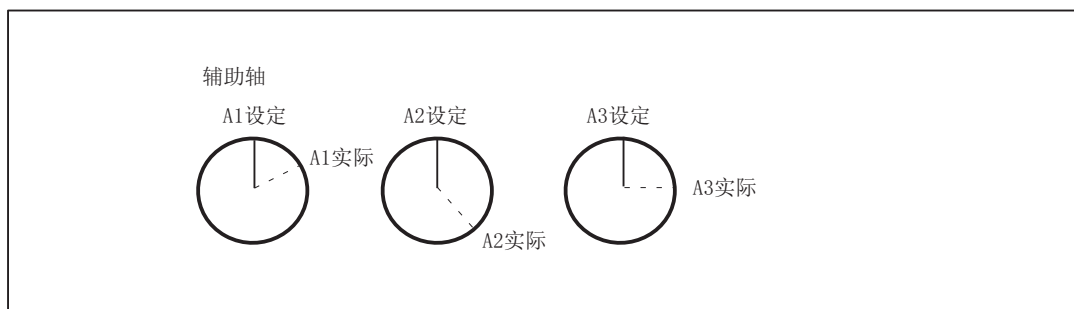


图 7-11 只针对辅助轴的实际值设定

针对测量类型 15，系统会分析下列变量值：

输入变量	含义
\$AC_MEAS_VALID	输入变量的有效位
\$AA_MEAS_POINT1[轴]	轴的实际值
\$AA_MEAS_SETPPOINT[轴]	单个轴的设定位置 *
\$AC_MEAS_FINE_TRANS	0: 粗偏移, 1: 精偏移 *
\$AC_MEAS_FRAME_SELECT	未设定时将计算叠加框架 *
\$AC_MEAS_TYPE	15

* 可选

针对测量类型 15，系统会写入以下输出变量：

输出变量	含义
\$AC_MEAS_FRAME	结果框架，包含平移

7.5.3.8 测量斜边（测量类型 16）

测量斜边（\$AC_MEAS_TYPE = 16）

通过此测量可确定工件位置角，并将其输入框架。可输入一个范围为 +/- 90 度的设定角度，其可作为激活相对生效 WCS 的结果框架后产生的旋转。

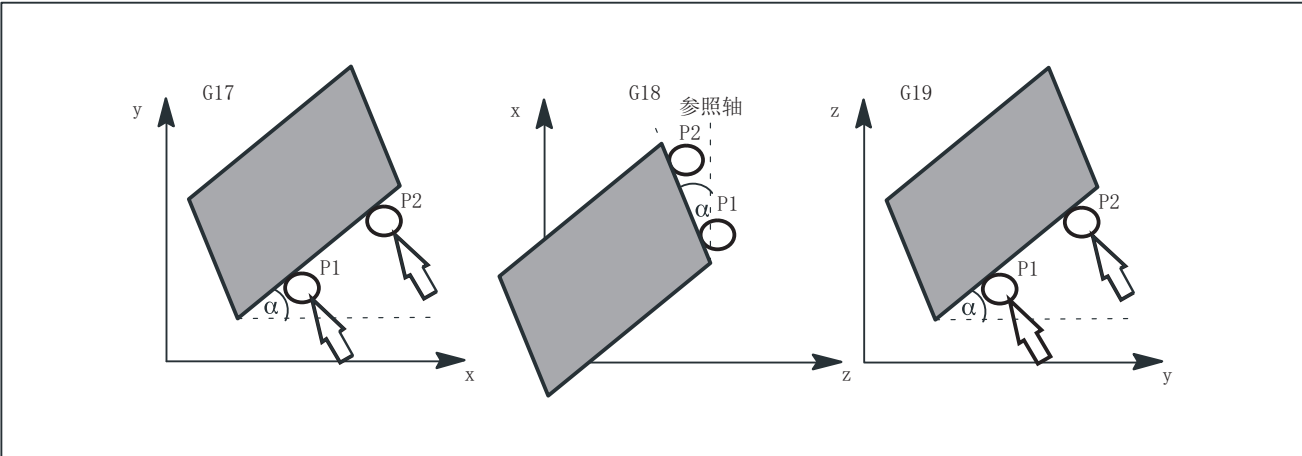


图 7-12 G17、G18 和 G19 平面中的斜边

针对测量类型 16，系统会分析下列变量值：

输入变量	含义
\$AC_MEAS_VALID	输入变量的有效位
\$AA_MEAS_POINT1[轴]	测量点 1
\$AA_MEAS_POINT2[轴]	测量点 2
\$AA_MEAS_SETANGLE	设定角度 *
\$AC_MEAS_ACT_PLANE	未设定时将采用当前生效的平面 *
\$AC_MEAS_FINE_TRANS	0: 粗偏移, 1: 精偏移 *
\$AC_MEAS_FRAME_SELECT	未设定时将计算叠加框架 *
\$AC_MEAS_T_NUMBER	未设定时将采用当前生效的 T (T0) *
\$AC_MEAS_D_NUMBER	未设定时将采用当前生效的 D (D0) *

输入变量	含义
\$AC_MEAS_INPUT[0]	若未设定，则总是将工件应对准的参考坐标作为所选平面的横坐标。 * =0: 参考坐标为横坐标 =1: 参考坐标为纵坐标
\$AC_MEAS_INPUT[1]	若未设定，则将工件位置角作为旋转输入框架。 否则可为回转轴指定一个通道轴索引，将其平移与计算出的旋转一起设置到当前的回转轴位置中。 这样工件便在回转轴位置 = 0 时对准。 当前回转轴值必须在 \$AA_MEAS_POINT[轴] 中设置。 *
\$AC_MEAS_TYPE	16

* 可选

针对测量类型 16，系统会写入以下输出变量：

输出变量	含义
\$AC_MEAS_FRAME	结果框架，包含旋转
\$AC_MEAS_WP_ANGLE	计算出的工件位置角

7.5.3.9 测量倾斜平面中的角度（测量类型 17）

测量倾斜平面中的角度（\$AC_MEAS_TYPE = 17）

倾斜平面通过 P1、P2 和 P3 三个测量点确定。

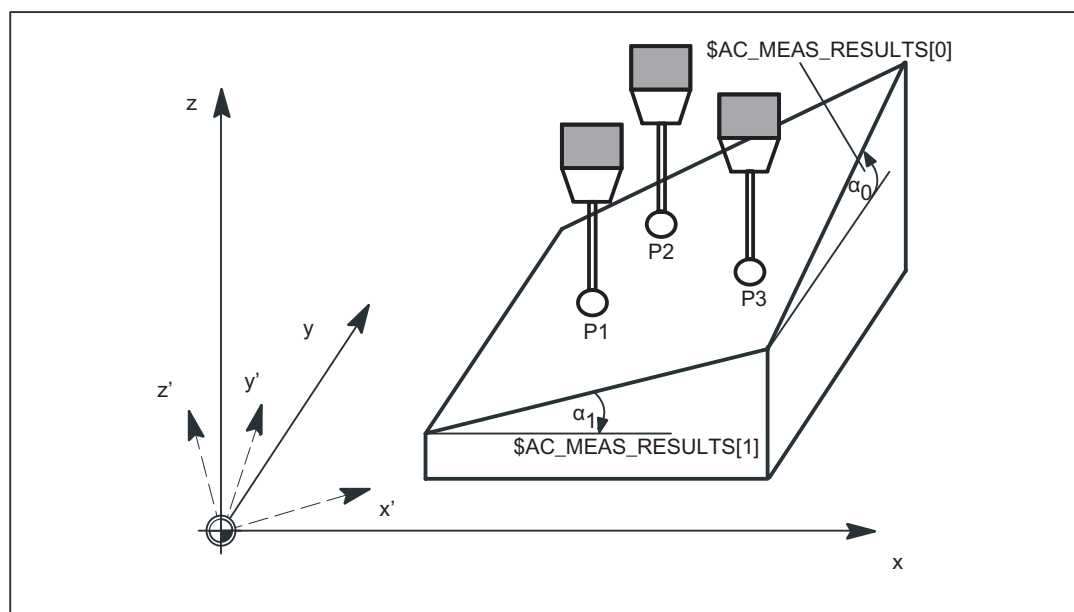


图 7-13 G17 中的倾斜平面

通过 \$AC_MEAS_TYPE = 17 确定两个用于平面倾斜位置的角度 α_0 和 α_1 ，并输入 \$AC_MEAS_RESULTS[0..1]:

- \$AC_MEAS_RESULTS[0] → 绕横坐标旋转
- \$AC_MEAS_RESULTS[1] → 绕纵坐标旋转

这些角度借助三个测量点 P1、P2 和 P3 计算。采用此测量类型时，垂直坐标的角度（\$AC_MEAS_RESULTS[2]）总是预设为 0。

可为横坐标和/或纵坐标设定一个设定旋转，此旋转输入至结果框架中。若只通过设定值设定了一个角度，计算第二个角度时会假定三个测量点在倾斜平面上构成设定角度。结果框架中只输入旋转，WCS 参考点保留。WCS 会被旋转，从而使 z' 垂直于倾斜平面。

针对测量类型 17 定义以下平面设置：

轴名称	G17	G18	G19
横坐标	x 轴	z 轴	y 轴
纵坐标	y 轴	x 轴	z 轴
垂直坐标（进给轴）	z 轴	y 轴	x 轴

针对测量类型 17，系统会分析下列变量值：

输入变量	含义
\$AC_MEAS_VALID	输入变量的有效位
\$AA_MEAS_POINT1[轴]	测量点 1
\$AA_MEAS_POINT2[轴]	测量点 2
\$AA_MEAS_POINT3[轴]	测量点 3
\$AA_MEAS_SETANGLE[轴]	围绕横坐标和纵坐标的设定旋转 *
\$AC_MEAS_ACT_PLANE	未设定时将采用当前生效的平面 *
\$AC_MEAS_FRAME_SELECTION	未设定时将计算叠加框架 *
\$AC_MEAS_T_NUMBER	未设定时将采用当前生效的 T（T0） *
\$AC_MEAS_D_NUMBER	未设定时将采用当前生效的 D（D0） *
\$AC_MEAS_INPUT[0]	未设定时， 点不投影至同一平面 * 0：点不投影至同一平面 1：点投影至生效的平面或选择的平面
\$AC_MEAS_TYPE	17

* 可选

针对测量类型 17，系统会写入以下输出变量：

输出变量	含义
\$AC_MEAS_FRAME	结果框架
\$AC_MEAS_RESULTS[0]	通过三个测量点计算出的、围绕横坐标的角度
\$AC_MEAS_RESULTS[1]	通过三个测量点计算出的、围绕纵坐标的角度
\$AC_MEAS_RESULTS[2]	通过三个测量点计算出的、围绕垂直坐标的角度
\$AC_MEAS_RESULTS[3]	结果框架中输入的、围绕横坐标的角度
\$AC_MEAS_RESULTS[4]	结果框架中输入的、围绕纵坐标的角度
\$AC_MEAS_RESULTS[5]	结果框架中输入的、围绕垂直坐标的角度

示例

测量平面角

程序代码	注释
DEF INT RETVAL	
DEF AXIS _XX, _YY, _ZZ	
T1 D1	; 激活测头
G54	; 激活所有框架和 G54
\$AC_MEAS_VALID = 0	; 将所有输入值设置为无效
\$AC_MEAS_TYPE = 17	; 将测量类型设置为“倾斜平面”
\$AC_MEAS_ACT_PLANE = 0	; 测量平面为 G17
_XX=\$P_AXN1	; 根据平面定义轴
_YY=\$P_AXN2	
_ZZ=\$P_AXN3	
G17 G1 _XX=10 _YY=10 F1000	; 逼近第 1 测量点
MEAS = 1 _ZZ=...	
\$AA_MEAS_POINT1[_xx] = \$AA_MW[_xx]	; 将测量值指定给横坐标
\$AA_MEAS_POINT1[_yy] = \$AA_MW[_yy]	; 将测量值指定给纵坐标
\$AA_MEAS_POINT1[_zz] = \$AA_MW[_zz]	; 将测量值指定给垂直坐标
G1 _XX=20 _YY=10 F1000	; 逼近第 2 测量点
MEAS = 1 _ZZ=...	
\$AA_MEAS_POINT2[_xx] = \$AA_MW[_xx]	; 将测量值指定给横坐标
\$AA_MEAS_POINT2[_yy] = \$AA_MW[_yy]	; 将测量值指定给纵坐标
\$AA_MEAS_POINT2[_zz] = \$AA_MW[_zz]	; 将测量值指定给垂直坐标
G1 _XX=20 _YY=20 F1000	; 逼近第 3 测量点
MEAS = 1 _ZZ=...	
\$AA_MEAS_POINT3[_xx] = \$AA_MW[_xx]	; 将测量值指定给横坐标
\$AA_MEAS_POINT3[_yy] = \$AA_MW[_yy]	; 将测量值指定给纵坐标
\$AA_MEAS_POINT3[_zz] = \$AA_MW[_zz]	; 将测量值指定给垂直坐标
	; 设定角度设定值
\$AA_MEAS_SETANGLE[_xx] = 12	; 绕横坐标旋转
\$AA_MEAS_SETANGLE[_yy] = 4	; 绕纵坐标旋转
\$AC_MEAS_FRAME_SELECT = 102	; 选择目标框架 - G55
\$AC_MEAS_T_NUMBER = 1	; 选择刀具

7.5 设置零点、工件测量和刀具测量

程序代码	注释
\$AC_MEAS_D_NUMBER = 1	
RETVAL = MEASURE()	; 启动测量计算
if RETVAL <> 0	
setal(61000 + RETVAL)	
endif	
if \$AC_MEAS_RESULTS[0] <> 12	
setal(61000 + \$AC_MEAS_RESULTS[0])	
endif	
if \$AC_MEAS_RESULTS[1] <> 4	
setal(61000 + \$AC_MEAS_RESULTS[1])	
endif	
\$P_UIFR[2] = \$AC_MEAS_FRAME	; 将测量框架写入数据管理 (G55)
G55 G0 AX[_xx]=10 AX[_yy]=10	; 激活框架并运行
m30	

7.5.3.10 通过测量重新定义 WCS 参考系统（测量类型 18）

重新定义 WCS' 坐标系（\$AC_MEAS_TYPE = 18）

新 WCS' 坐标系的零点通过倾斜平面法线上的测量点 P1 确定。

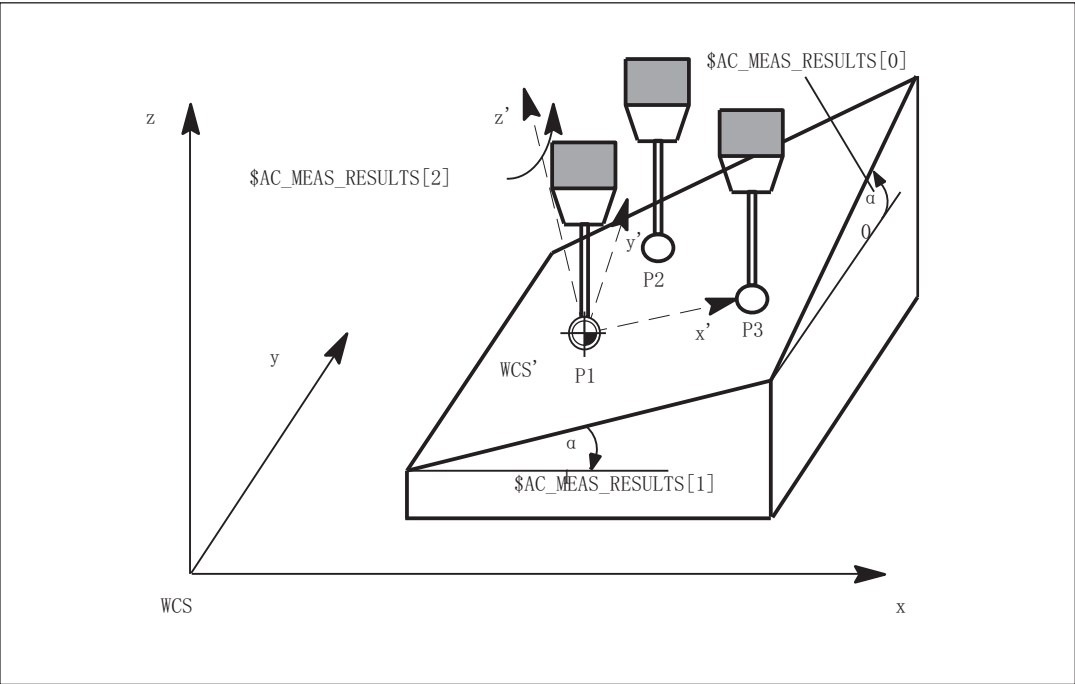


图 7-14 G17 中的倾斜平面

平面测量

平面测量在测量循环中执行。此循环记录三个测量点，并向测量接口提供必要的数值。借助 MEASURE() 功能可依据输入值计算出空间角以及新 WCS' 的平移。

测量框架转换

计算的结果（即空间角和平移）会被输入至测量框架 \$AC_MEAS_FRAME 中。根据框架链中选择的框架，系统会对测量框架进行转换。此外空间角还会被保存至输出值 \$AC_MEAS_RESULTS[0..2] 中。其中

- \$AC_MEAS_RESULTS[0] 为围绕旧 WCS 横坐标的角度，
- \$AC_MEAS_RESULTS[1] 为围绕纵坐标的角度
- \$AC_MEAS_RESULTS[2] 为围绕垂直坐标的角度

确定新 WCS' 的零点

执行完测量后，测量循环可通过测量框架描述和激活框架链中选择的框架。激活后，新的 WCS' 坐标系位于倾斜平面的法线上，测量点 P1 作为新 WCS' 的零点。

之后编写的位置将以倾斜平面为基准。

应用

CAD 系统常通过设定一个平面上的 P1、P2 和 P3 三个点来定义空间中的斜置平面。此时

- 第 1 测量点 P1 用作新 WCS' 参考点
- 第 2 测量点 P2 用于设定经过旋转的新 WCS' 的横坐标 x' 的方向
- 第 3 测量点 P3 用于确定空间角。

针对测量类型 18，系统会分析下列变量值：

输入变量	含义
\$AC_MEAS_VALID	输入变量的有效位
\$AA_MEAS_POINT1[轴]	测量点 1
\$AA_MEAS_POINT2[轴]	测量点 2
\$AA_MEAS_POINT3[轴]	测量点 3
\$AA_MEAS_SETPPOINT[轴]	P1 的设定位置 *
\$AC_MEAS_ACT_PLANE	未设定时将采用当前生效的平面 *
\$AC_MEAS_FRAME_SELECT	未设定时将计算叠加框架 *
\$AC_MEAS_T_NUMBER	未设定时将采用当前生效的 T (T0) *
\$AC_MEAS_D_NUMBER	未设定时将采用当前生效的 D (D0) *
\$AC_MEAS_INPUT[0]	未设定时， 点不投影至同一平面 * 0: 点不投影至同一平面 1: 点投影至生效的平面或选择的平面
\$AC_MEAS_TYPE	18

* 可选

针对测量类型 18，系统会写入以下输出变量：

输出变量	含义
\$AC_MEAS_FRAME	结果框架，包含旋转和转换
\$AC_MEAS_RESULTS[0]	计算出的围绕横坐标的角度
\$AC_MEAS_RESULTS[1]	计算出的围绕纵坐标的角度
\$AC_MEAS_RESULTS[2]	计算出的围绕垂直坐标的角度

示例

倾斜平面上的工件坐标系

程序代码	注释
DEF INT RETVAL DEF AXIS _XX, _YY, _ZZ	
T1 D1 G54	; 激活测头 ; 激活所有框架和 G54
\$AC_MEAS_VALID = 0 \$AC_MEAS_TYPE = 18 \$AC_MEAS_ACT_PLANE = 0	; 将所有输入值设置为无效 ; 将测量类型设置为“倾斜平面” ; 测量平面为 G17
_XX=\$P_AXN1 _YY=\$P_AXN2 _ZZ=\$P_AXN3	; 根据平面定义轴
G17 G1 _XX=10 _YY=10 F1000 MEAS = 1 _ZZ=...	; 逼近第 1 测量点
\$AA_MEAS_POINT1[_xx] = \$AA_MW[_xx] \$AA_MEAS_POINT1[_yy] = \$AA_MW[_yy] \$AA_MEAS_POINT1[_zz] = \$AA_MW[_zz]	; 将测量值指定给横坐标 ; 将测量值指定给纵坐标 ; 将测量值指定给垂直坐标
G1 _XX=20 _YY=10 F1000 MEAS = 1 _ZZ=...	; 逼近第 2 测量点
\$AA_MEAS_POINT2[_xx] = \$AA_MW[_xx] \$AA_MEAS_POINT2[_yy] = \$AA_MW[_yy] \$AA_MEAS_POINT2[_zz] = \$AA_MW[_zz]	; 将测量值指定给横坐标 ; 将测量值指定给纵坐标 ; 将测量值指定给垂直坐标
G1 _XX=20 _YY=20 F1000 MEAS = 1 _ZZ=...	; 逼近第 3 测量点
\$AA_MEAS_POINT3[_xx] = \$AA_MW[_xx]	; 将测量值指定给横坐标

7.5 设置零点、工件测量和刀具测量

程序代码	注释
\$AA_MEAS_POINT3[_yy] = \$AA_MW[_yy]	; 将测量值指定给纵坐标
\$AA_MEAS_POINT3[_zz] = \$AA_MW[_zz]	; 将测量值指定给垂直坐标
\$AA_MEAS_SETPOINT[_xx] = 10	; 设定 P1 的设定值
\$AA_MEAS_SETPOINT[_yy] = 10	
\$AA_MEAS_SETPOINT[_zz] = 10	
\$AC_MEAS_FRAME_SELECT = 102	; 选择目标框架 - G55
\$AC_MEAS_T_NUMBER = 1	; 选择刀具
\$AC_MEAS_D_NUMBER = 1	
RETVAL = MEASURE()	; 启动测量计算
if RETVAL <> 0	
setal(61000 + RETVAL)	
endif	
	; 空间角的计算结果
	; 角度围绕
R0 = \$AC_MEAS_RESULTS[0]	; 旧 WCS 的横坐标
R1 = \$AC_MEAS_RESULTS[1]	; 纵坐标
R2 = \$AC_MEAS_RESULTS[2]	; 垂直坐标
\$P_UIFR[2] = \$AC_MEAS_FRAME	; 将测量框架写入数据管理 (G55)
G55 G0 AX[_xx]=10 AX[_yy]=10	; 激活框架并运行
m30	

7.5.3.11 1 维、2 维、3 维设定值给定测量（测量类型 19、20、21）

1 维设定值给定（\$AC_MEAS_TYPE = 19）

使用此测量方法可为横坐标、纵坐标或垂直坐标给定一个设定值。若给定了两个或三个设定值，则只会按照“横坐标 - 纵坐标 - 垂直坐标”的顺序接收第一个值。此时刀具不被考虑在内。

这是一个完全意义上的用于横坐标、纵坐标或垂直坐标的实际值设定功能。

针对测量类型 19，系统会分析下列变量值：

输入变量	含义
\$AC_MEAS_VALID	输入变量的有效位
\$AA_MEAS_POINT1[轴]	横坐标的测量点 1

输入变量	含义
\$AA_MEAS_POINT1[轴]	纵坐标的测量点 1
\$AA_MEAS_POINT1[轴]	垂直坐标的测量点 1
\$AA_MEAS_SETPPOINT[轴]	横坐标、纵坐标或垂直坐标的设定位置
\$AC_MEAS_ACT_PLANE	未设定时将采用当前生效的平面 *
\$AC_MEAS_FRAME_SELECT	未设定时将计算叠加框架 *
\$AC_MEAS_FINE_TRANS	未设定时将写入粗偏移 *
\$AC_MEAS_TYPE	19

* 可选

针对测量类型 19，系统会写入以下输出变量：

输出变量	含义
\$AC_MEAS_FRAME	结果框架，包含旋转和平移

示例

1 维设定值给定

程序代码	注释
DEF INT RETVAL	
DEF REAL _CORMW_XX,	
_CORMW_YY,	
_CORMW_ZZ	
DEF AXIS _XX, _YY, _ZZ	
T1 D1	; 激活测头
G54	; 激活所有框架和 G54
\$AC_MEAS_VALID = 0	; 将所有输入值设置为无效
\$AC_MEAS_TYPE = 19	; 将测量类型设为“1 维设定值给定”
\$AC_MEAS_ACT_PLANE = 0	; 测量平面为 G17
_XX=\$P_AXN1	; 根据平面定义轴
_YY=\$P_AXN2	
_ZZ=\$P_AXN3	
	; 指定测量值
\$AA_MEAS_POINT1[_xx] = \$AA_MW[_xx]	; 将测量值指定给横坐标
\$AA_MEAS_POINT1[_yy] = \$AA_MW[_yy]	; 将测量值指定给纵坐标
\$AA_MEAS_POINT1[_zz] = \$AA_MW[_zz]	; 将测量值指定给垂直坐标
\$AA_MEAS_SETPPOINT[_xx] = 10	; 设定横坐标的设定值
\$AC_MEAS_FRAME_SELECT = 102	; 选择目标框架 - G55

程序代码	注释
RETVAL = MEASURE()	; 启动测量计算
if RETVAL <> 0	
setal(61000 + RETVAL)	
endif	
\$P_UIFR[2] = \$AC_MEAS_FRAME	; 将测量框架写入数据管理 (G55)
G55 G0 AX[_xx]=10 AX[_yy]=10	; 激活框架并运行
m30	

2 维设定值给定 (\$AC_MEAS_TYPE = 20)

采用此测量方法时可设定两个维度的设定值。此时可从 3 个维度中任意选择 2 个。若给定了三个设定值，则只会接收横坐标和纵坐标的值。此时刀具不被考虑在内。

这是一个完全意义上的实际值设定功能。

针对测量类型 20，系统会分析下列变量值：

输入变量	含义
\$AC_MEAS_VALID	输入变量的有效位
\$AA_MEAS_POINT1[轴]	横坐标的测量点 1
\$AA_MEAS_POINT1[轴]	纵坐标的测量点 1
\$AA_MEAS_POINT1[轴]	垂直坐标的测量点 1
\$AA_MEAS_SETPPOINT[轴]	第 1 维的设定位置
\$AA_MEAS_SETPPOINT[轴]	第 2 维的设定位置
\$AC_MEAS_ACT_PLANE	未设定时将采用当前生效的平面 *
\$AC_MEAS_FRAME_SELECT	未设定时将计算叠加框架 *
\$AC_MEAS_FINE_TRANS	未设定时将写入粗偏移 *
\$AC_MEAS_TYPE	20

* 可选

针对测量类型 20，系统会写入以下输出变量：

输出变量	含义
\$AC_MEAS_FRAME	结果框架，包含旋转和平移

示例

2 维设定值给定

程序代码	注释
DEF INT RETVAL	
DEF REAL _CORMW_XX,	
_CORMW_YY,	
_CORMW_ZZ	
DEF AXIS _XX, _YY, _ZZ	
T1 D1	; 激活测头
G54	; 激活所有框架和 G54
\$AC_MEAS_VALID = 0	; 将所有输入值设置为无效
\$AC_MEAS_TYPE = 20	; 将测量类型设为“2 维设定值给定”
\$AC_MEAS_ACT_PLANE = 0	; 测量平面为 G17
_XX=\$P_AXN1	; 根据平面定义轴
_YY=\$P_AXN2	
_ZZ=\$P_AXN3	
	; 指定测量值
\$AA_MEAS_POINT1[_xx] = \$AA_MW[_xx]	; 将测量值指定给横坐标
\$AA_MEAS_POINT1[_yy] = \$AA_MW[_yy]	; 将测量值指定给纵坐标
\$AA_MEAS_POINT1[_zz] = \$AA_MW[_zz]	; 将测量值指定给垂直坐标
\$AA_MEAS_SETPPOINT[_xx] = 10	; 设定横坐标和纵坐标的设定值
\$AA_MEAS_SETPPOINT[_yy] = 10	
\$AC_MEAS_FRAME_SELECT = 102	; 选择目标框架 - G55
RETVAL = MEASURE()	; 启动测量计算
if RETVAL <> 0	
setal(61000 + RETVAL)	
endif	
\$P_UIFR[2] = \$AC_MEAS_FRAME	; 将测量框架写入数据管理 (G55)
G55 G0 AX[_xx]=10 AX[_yy]=10	; 激活框架并运行
m30	

3 维设定值给定 (\$AC_MEAS_TYPE = 21)

使用此测量方法可为横坐标、纵坐标和垂直坐标分别给定一个设定值。此时刀具不被考虑在内。

这是一个完全意义上的用于横坐标、纵坐标和垂直坐标的实际值设定功能。

针对测量类型 21，系统会分析下列变量值：

输入变量	含义
\$AC_MEAS_VALID	输入变量的有效位
\$AA_MEAS_POINT1[轴]	横坐标的测量点 1
\$AA_MEAS_POINT1[轴]	纵坐标的测量点 1
\$AA_MEAS_POINT1[轴]	垂直坐标的测量点 1
\$AA_MEAS_SETPPOINT[轴]	横坐标的设定位置
\$AA_MEAS_SETPPOINT[轴]	纵坐标的设定位置
\$AA_MEAS_SETPPOINT[轴]	垂直坐标的设定位置
\$AC_MEAS_ACT_PLANE	未设定时将采用当前生效的平面 *
\$AC_MEAS_FRAME_SELECT	未设定时将计算叠加框架 *
\$AC_MEAS_FINE_TRANS	未设定时将写入粗偏移 *
\$AC_MEAS_TYPE	21

* 可选

针对测量类型 21，系统会写入以下输出变量：

输出变量	含义
\$AC_MEAS_FRAME	结果框架，包含旋转和平移

示例

3 维设定值给定

程序代码	注释
DEF INT RETVAL	
DEF REAL _CORMW_XX,	
_CORMW_YY,	
_CORMW_ZZ	
DEF AXIS _XX, _YY, _ZZ	
T1 D1	; 激活测头
G54	; 激活所有框架和 G54
\$AC_MEAS_VALID = 0	; 将所有输入值设置为无效
\$AC_MEAS_TYPE = 21	; 将测量类型设为“3 维设定值给定”
\$AC_MEAS_ACT_PLANE = 0	; 测量平面为 G17
_XX=\$P_AXN1	; 根据平面定义轴
_YY=\$P_AXN2	
_ZZ=\$P_AXN3	

7.5 设置零点、工件测量和刀具测量

程序代码	注释
	; 指定测量值
\$AA_MEAS_POINT1[_xx] = \$AA_MW[_xx]	; 将测量值指定给横坐标
\$AA_MEAS_POINT1[_yy] = \$AA_MW[_yy]	; 将测量值指定给纵坐标
\$AA_MEAS_POINT1[_zz] = \$AA_MW[_zz]	; 将测量值指定给垂直坐标
\$AA_MEAS_SETPPOINT[_xx] = 10	; 给定横坐标、纵坐标和垂直坐标的设定值
\$AA_MEAS_SETPPOINT[_yy] = 10	;
\$AA_MEAS_SETPPOINT[_zz] = 10	
\$AC_MEAS_FRAME_SELECT = 102	; 选择目标框架 - G55
\$AA_MEAS_SETPPOINT[_yy] = 10	
RETVAL = MEASURE()	; 启动测量计算
if RETVAL <> 0	
setal(61000 + RETVAL)	
endif	
\$P_UIFR[2] = \$AC_MEAS_FRAME	; 将测量框架写入数据管理 (G55)
G55 G0 AX[_xx]=10 AX[_yy]=10	; 激活框架并运行
m30	

7.5.3.12 任意坐标系中的测量点测量（测量类型 24）

将测量点换算至任意坐标系（\$AC_MEAS_TYPE = 24）

使用此测量方法可将任意坐标系（WCS、BCS、MCS）中的测量点通过坐标转换换算，使其以新的坐标系为基准。

新坐标系通过指定所需的框架链生成。

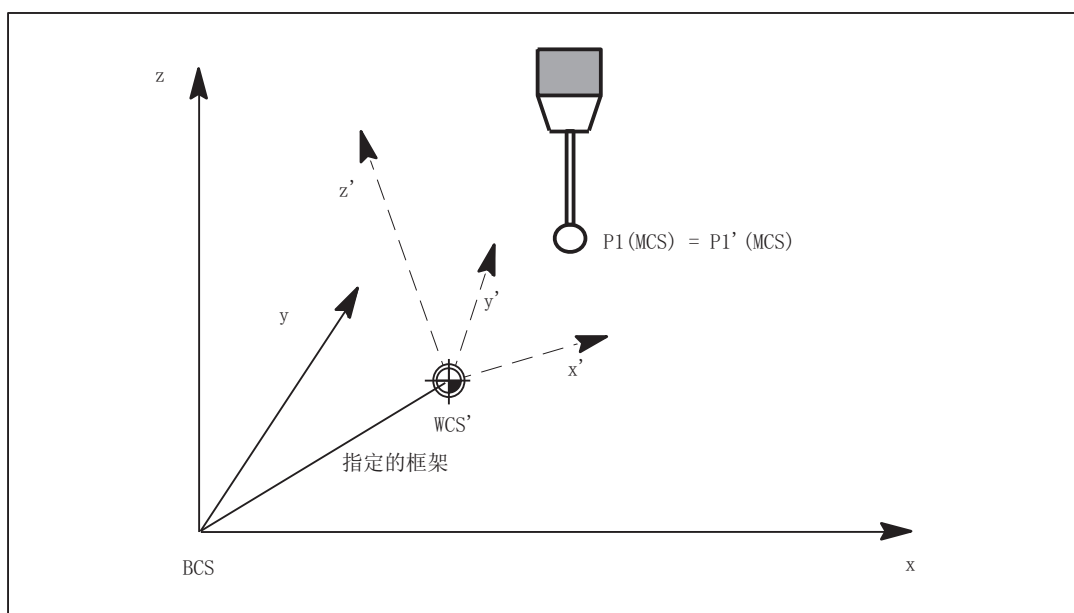


图 7-15 位置的坐标转换

针对测量类型 24，系统会分析下列变量值：

输入变量	含义
\$AC_MEAS_VALID	输入变量的有效位
\$AA_MEAS_POINT1[轴]	需要转换的位置
\$AC_MEAS_P1:COORD	0: WCS（缺省设置），1: BCS，2: MCS *
\$AC_MEAS_P2_COORD	目标坐标系 *
\$AC_MEAS_TOOL_MASK	0x20；在位置坐标转换中将生效刀具的长度考虑在内 *
\$AC_MEAS_CHSFR	数据管理中的系统框架 *
\$AC_MEAS_NCBFR	数据管理中的全局基本框架 *
\$AC_MEAS_CHBFR	数据管理中的通道基本框架 *
\$AC_MEAS_UIFR	数据管理中的可设定框架 *
\$AC_MEAS_PFRAME	1: 不结算可编程框架 *
\$AC_MEAS_TYPE	24

* 可选

针对测量类型 24，系统会写入以下输出变量：

输出变量	含义
\$AC_MEAS_POINT2[轴]	经过换算的轴位置

示例

被测位置的 WCS 坐标转换

程序代码	注释
DEF INT RETVAL	
DEF INT LAUF	
DEF REAL_CORMW_xx, _CORMW_yy, _CORMW_zz	
DEF AXIS _XX, _YY, _ZZ	
 \$TC_DP1[1,1]=120	 ; 刀具类型为轴铣刀
\$TC_DP2[1,1]=20	
\$TC_DP3[1,1]=0	; (z) 长度补偿矢量
\$TC_DP4[1,1]=0	; (y) 长度补偿矢量
\$TC_DP5[1,1]= 0	; (x) 长度补偿矢量
\$TC_DP6[1,1]=2	; 半径
 T1 D1	 ; 激活测头
G17	; 倾斜平面 G17
_xx=\$P_AXN1 _yy=\$P_AXN2 _zz=\$P_AXN3	; 根据平面定义轴
 	 ; 总框架通过 CTRANS(_xx,10,_yy,-1,_zz,5,A,6,B,7) 生成
\$P_CHBFR[0]=CTRANS(_zz,5,A,6) : CROT(_zz,45)	
\$P_UIFR[1]=CTRANS()	
\$P_UIFR[1,_xx,TR]=--SIN(45)	
\$P_UIFR[1,_yy,TR]=--SIN(45)	
\$P_UIFR[2]=CTRANS()	
\$P_PFRAME=CROT(_zz,-45)	
\$P_CYCFR=CTRANS(_xx,10,B,7)	
 G54	 ; 激活所有框架和 G54
G0 X0 Y0 Z0 A0 B0 F1000	
 \$AC_MEAS_VALID = 0	 ; 将所有输入值设置为无效
\$AC_MEAS_TYPE = 24	; 将测量类型设置为“坐标转换”
\$AC_MEAS_ACT_PLANE = 0	; 测量平面为 G17
 	 ; 指定测量值
\$AA_MEAS_POINT1[_xx] = \$AA_IW[_xx]	; 将测量值指定给横坐标
\$AA_MEAS_POINT1[_yy] = \$AA_IW[_yy]	; 将测量值指定给纵坐标

7.5 设置零点、工件测量和刀具测量

程序代码	注释
\$AA_MEAS_POINT1[_zz] = \$AA_IW[_zz]	; 将测量值指定给垂直坐标
\$AA_MEAS_POINT1[A] = \$AA_IW[A]	
\$AA_MEAS_POINT1[B] = \$AA_IW[B]	
\$AC_MEAS_P1_COORD=0	; WCS 位置换算至 WCS'
\$AC_MEAS_P2_COORD=0	
	; 设置 WCS
	; 总框架通过 CTRANS(_xx,0,_yy,0,_zz,5,A,6,B,0) 生成
	; 取消循环框架
\$AC_MEAS_CHSER=\$MC_MM_SYSTEM_FRAME_MASK B_AND 'B1011111'	
\$AC_MEAS__NCBFR='B0'	; 取消全局基本框架
\$AC_MEAS__CHBFR='B1'	; 数据管理中的通道基本框架 1
\$AC_MEAS__UIFR=2	; 数据管理中的可设定框架 G55
\$AA_MEAS_PFRAME=1	; 可设定框架不计入
RETVAL = MEASURE()	; 启动测量计算
if RETVAL <> 0	
setal(61000 + RETVAL)	
endif	
if \$AA_MEAS_PIONT2[_xx] <> 10	
setal(61000)	
M0	
stopre	
endif	
if \$AA_MEAS_PIONT2[_yy] <> -1	
setal(61000)	
M0	
stopre	
if \$AA_MEAS_PIONT2[_zz] <> 0	
setal(61000)	
M0	
stopre	
if \$AA_MEAS_PIONT2[A] <> 0	
setal(61000)	
M0	
stopre	
if \$AA_MEAS_PIONT2[B] <> 7	
setal(61000)	
M0	
stopre	
m30	

7.5.3.13 测量矩形（测量类型 25）

测定矩形（\$AC_MEAS_TYPE = 25）

测定矩形在以下平面中借助刀具进行

- G17 工作平面 xy，进给方向 z
- G18 工作平面 zx，进给方向 y
- G19 工作平面 yz，进给方向 x

每个矩形需要 4 个测量点

这些测量点可按任意顺序设定。纵坐标间距最大的测量点为 P3 和 P4。

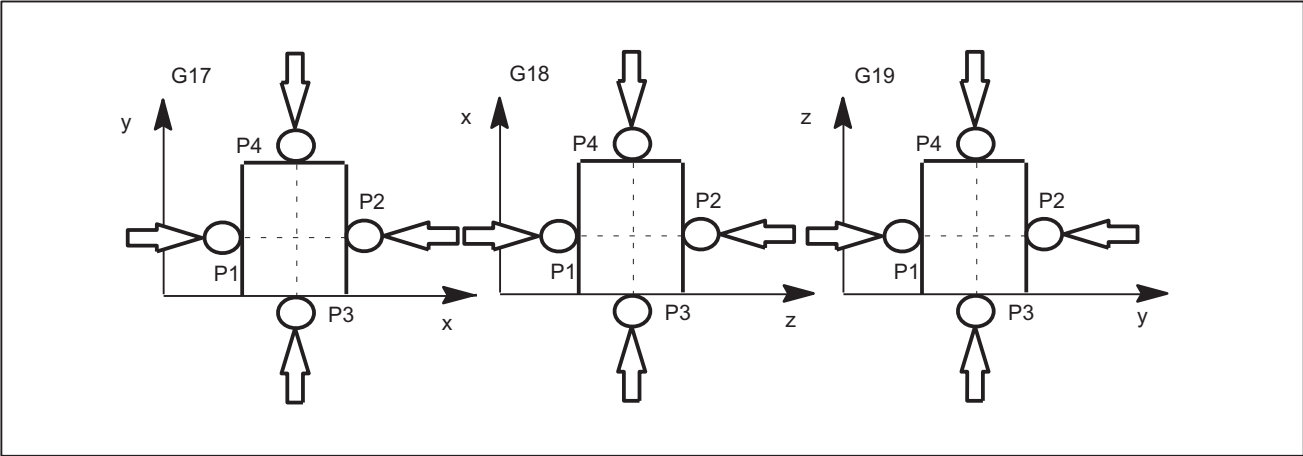


图 7-16 通过工作平面 G17、G18 和 G19 中的进给测定矩形

针对测量类型 25，系统会分析下列变量值：

输入变量	含义
\$AC_MEAS_VALID	输入变量的有效位
\$AA_MEAS_POINT1[轴]	测量点 1
\$AA_MEAS_POINT2[轴]	测量点 2
\$AA_MEAS_POINT3[轴]	测量点 3
\$AA_MEAS_POINT4[轴]	测量点 4
\$AA_MEAS_SETPPOINT[轴]	隔断中心的设定位置 *
\$AC_MEAS_ACT_PLANE	未设定时将采用当前生效的平面 *
\$AC_MEAS_FINE_TRANS	0: 粗偏移，1: 精偏移 *

输入变量	含义
\$AC_MEAS_FRAME_SELECT	未设定时将计算叠加框架 *
\$AC_MEAS_T_NUMBER	未设定时将采用当前生效的 T (T0) *
\$AC_MEAS_D_NUMBER	未设定时将采用当前生效的 D (D0) *
\$AC_MEAS_INPUT[0]	未设定时测量外角 * =0: 外角测量 =1: 内角测量
\$AC_MEAS_TYPE	25

* 可选

针对测量类型 25，系统会写入以下输出变量：

输出变量	含义
\$AC_MEAS_FRAME	结果框架，包含平移
\$AC_MEAS_RESULTS[0]	计算出的中心的横坐标
\$AC_MEAS_RESULTS[1]	计算出的中心的纵坐标
\$AC_MEAS_RESULTS[2]	计算出的中心的垂直坐标
\$AC_MEAS_RESULTS[3]	矩形宽度 P1/P2
\$AC_MEAS_RESULTS[4]	矩形长度 P3/P4

7.5.3.14 用于备份数据管理框架的测量（测量类型 26）

备份数据管理框架（\$AC_MEAS_TYPE = 26）

使用此测量类型可将全部或选定的部分数据管理框架以当前的赋值备份至一个文件。这可通过执行命令或通过零件程序进行。此功能也可从不同的通道激活。备份文件在 _N_SYF_DIR 目录下创建。

恢复操作会删除数据备份，重新备份会覆盖此前的备份。可通过以下设置

- \$AC_MEAS_CHSFR = 0 系统框架；
- \$AC_MEAS_NCBFR = 0 全局基本框架；
- \$AC_MEAS_CHBFR = 0 通道基本框架；
- \$AC_MEAS_UIFR = 0 可设定框架数量

重新备份框架，从而将最近一次备份的数据从数据管理中删除。

说明

备份所有数据管理框架时须注意，每个框架要占用约 1 kB 的存储空间。若存储空间不足，那么备份会被终止，并发出故障信息 MEAS_NO_MEMORY。可通过以下机床数据调整静态存储器的大小：

MD18351 \$MM_DRAM_FILE_MEM_SIZE

针对测量类型 26，系统会分析下列变量值：

输入变量	含义
\$AC_MEAS_VALID	输入变量的有效位
\$AC_MEAS_CHSFR	数据管理中的系统框架的位标记。 * 若未写入此变量，则会备份所有系统框架。
\$AC_MEAS_NCBFR	数据管理中的全局基本框架的位标记。 * 若未写入此变量，则会备份所有全局基本框架。
\$AC_MEAS_CHBFR	数据管理中的通道基本框架的位标记。 * 若未写入此变量，则会备份所有通道基本框架。
\$AC_MEAS_UIFR	数据管理中的可设定框架的数量。 * 0..100: 1: G500 2: G500, G54 若未写入此变量，则会备份所有可设定框架。
\$AC_MEAS_TYPE	26

* 可选

7.5.3.15 用于恢复备份的数据管理框架的测量（测量类型 27）

恢复最近一次备份的数据管理框架（\$AC_MEAS_TYPE = 27）

使用此测量类型可将通过测量类型 26 备份的数据管理框架重新恢复至 SRAM。

可恢复最近一次备份的所有框架，或只选择恢复其中的一部分。若选择了未备份的框架，此设定会被忽略，且不会使恢复进程终止。

针对测量类型 27，系统会分析下列变量值：

输入变量	含义
\$AC_MEAS_VALID	输入变量的有效位
\$AC_MEAS_CHSFR	数据管理中的系统框架的位标记。* 若未写入此变量，则会恢复所有系统框架。
\$AC_MEAS_NCBFR	数据管理中的全局基本框架的位标记。* 若未写入此变量，则会恢复所有全局基本框架。
\$AC_MEAS_CHBFR	数据管理中的通道基本框架的位标记。* 若未写入此变量，则会恢复所有通道基本框架。
\$AC_MEAS_UIFR	数据管理中的可设定框架的数量。* 范围从 1: G54 至 G99: G599。 若未写入此变量，则会恢复所有可设定框架。
\$AC_MEAS_TYPE	27

* 可选

7.5.3.16 用于为锥面车削设定叠加旋转的测量（测量类型 28）

锥面车削的平面叠加旋转（\$AC_MEAS_TYPE = 28）

使用测量类型 28 可为生效平面或特定平面设定一个叠加旋转，旋转角度范围为 $\alpha = \pm 90^\circ$ 。旋转围绕与平面相垂直的坐标轴进行。

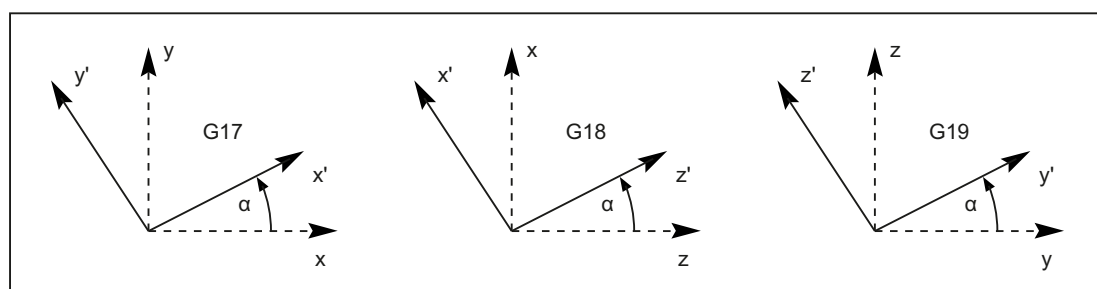


图 7-17 G17、G18 和 G19 平面旋转，角度 $\alpha = +30^\circ$

应用

在锥面车削中，生效平面会旋转特定锥角，此时旋转会被写入生效的循环框架。RESET 时循环框架会被清除。必要时可重新激活。循环框架的选择取决于 ENS 位置显示。激活旋转后，例如会在生效的 G18 平面中向 z' 方向运行， x 和 z 对应的轴的实际位置也同时随之变化。

7.5 设置零点、工件测量和刀具测量

G17 和 G19 平面中的旋转特性与之类似，见上图。针对测量类型 28，系统会分析下列变量值：

输入变量	含义
\$AC_MEAS_VALID	输入变量的有效位
\$AC_MEAS_WP_SETANGLE	设定角度
\$AC_MEAS_ACT_PLANE	未设定时将围绕生效的平面旋转 *
\$AC_MEAS_FRAME_SELECT	未设定时将计算叠加框架 *
\$AC_MEAS_INPUT[0]	1: 锥面车削生效。 *
\$AC_MEAS_TYPE	28

* 可选

针对测量类型 28，系统会写入以下输出变量：

输出变量	含义
\$AC_MEAS_FRAME	结果，包含旋转

7.5.4 刀具测量

通过输入刀具长度，控制系统可以计算出刀尖至刀架参考点 T 的距离。

为了在车床或铣床上对换入的刀具执行测量，可使用以下测量类型：

测量类型	刀具测量
\$AC_MEAS_TYPE = 10	在已经过测量的参考件上测量刀具长度
\$AC_MEAS_TYPE = 11	在已经过测量的参考件上测量刀具直径

测量类型	刀具测量
\$AC_MEAS_TYPE = 22	在机床上通过放大镜功能测量刀具直径 (ShopTurn)
\$AC_MEAS_TYPE = 23	通过标记位置或当前位置测量刀具长度 (ShopTurn) 通过定向测量两把刀具的长度: 两把车刀 启用各自参考点, 刀具定向沿逼近方向 启用一个参考点, 刀具位置与逼近方向及刀具定向相反。 两把铣刀 启用各自参考点, 刀具定向沿 -y 方向 启用一个参考点, 刀具定向沿 -y 方向, 刀具位置与逼近方向相反。 两把旋转 90 度的铣刀 启用各自参考点, 刀具定向沿逼近方向 启用一个参考点, 刀具位置与逼近方向及刀具定向相反。

7.5.5 刀具测量的测量类型

7.5.5.1 测量刀具长度 (测量类型 10)

在已经过测量的参考件上测量刀具长度 (\$AC_MEAS_TYPE = 10)

可在一个已经过测量的参考件上进行刀具长度测量。

平面选择取决于刀具位置:

7.5 设置零点、工件测量和刀具测量

- G17 用于 z 轴方向的刀具位置
- G18 用于 y 轴方向的刀具位置
- G19 用于 x 轴方向的刀具位置

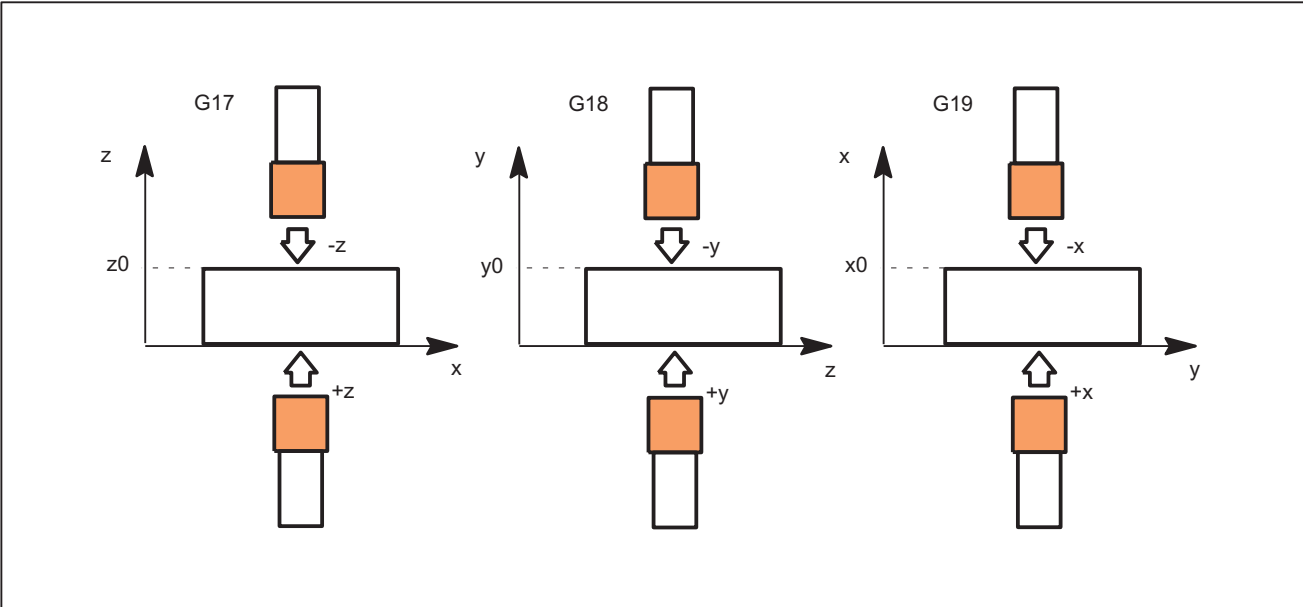


图 7-18 为选择的 G17、G18 和 G19 平面测量刀具长度

针对测量类型 10，系统会分析下列变量值：

输入变量	含义
\$AC_MEAS_VALID	输入变量的有效位
\$AA_MEAS_POINT1[轴]	测量点 1
\$AC_MEAS_P1_COORD	测量点的坐标系 *
\$AA_MEAS_SETPOINT[轴]	设定位置 z0
\$AC_MEAS_SET_COORD	设定点的坐标系 *
\$AC_MEAS_DIR_APPROACH	0: +x, 1: -x, 2: +y, 3: -y, 4: +z, 5: -z
\$AC_MEAS_ACT_PLANE	未设定时将采用当前生效的平面 *
\$AC_MEAS_TYPE	10

* 可选

针对测量类型 10，系统会写入以下输出变量：

输出变量	含义
\$AC_MEAS_TOOL_LENGTH	刀具长度
\$AC_MEAS_RESULTS[0]	刀具长度, x 方向
\$AC_MEAS_RESULTS[1]	刀具长度, y 方向
\$AC_MEAS_RESULTS[2]	刀具长度, z 方向
\$AC_MEAS_RESULTS[3]	刀具长度 L1
\$AC_MEAS_RESULTS[4]	刀具长度 L2
\$AC_MEAS_RESULTS[5]	刀具长度 L3

示例

测量刀具长度

程序代码	注释
DEF INT RETVAL	
T0 D0	
G0 X0 Y0 Z0 F10000	
	; 测量刀具长度
\$AC_MEAS_VALID = 0	; 将所有输入值设置为无效
G1 Z10	; 将刀具运行至参考件
\$AC_MEAS_LATCH[0] = 1	; 解锁测量点 1
\$AC_MEAS_DIR_APPROACH = 5	; 设置逼近方向 -z
\$AA_MEAS_SETPPOINT[X] = 0	; 设置参考位置
\$AA_MEAS_SETPPOINT[Y] = 0	
\$AA_MEAS_SETPPOINT[Z] = 0	
\$AC_MEAS_ACT_PLANE = 0	; 测量平面为 G17
\$AC_MEAS_T_NUMBER = 0	; 未选择刀具
\$AC_MEAS_D_NUMBER = 0	
\$AC_MEAS_TYPE = 10	; 将测量类型设置为“刀具长度”
RETVAL = MEASURE()	, 开始测量
if RETVAL <> 0 setal(61000 + RETVAL)	
endif	
if \$AC_MEAS_TOOL_LENGTH <> 10	; 查询已知刀具长度
setal(61000 + \$AC_MEAS_TOOL_LENGTH)	
endif	
M30	

7.5.5.2 测量刀具直径（测量类型 11）

在已经过测量的参考件上测量刀具直径（\$AC_MEAS_TYPE = 11）

可在一个已经过测量的参考件上进行刀具直径测量。此时须根据刀具位置选择平面，G17 用于 z 轴方向的刀具位置，G18 用于 y 轴方向的刀具位置，G19 用于 x 轴方向的刀具位置。

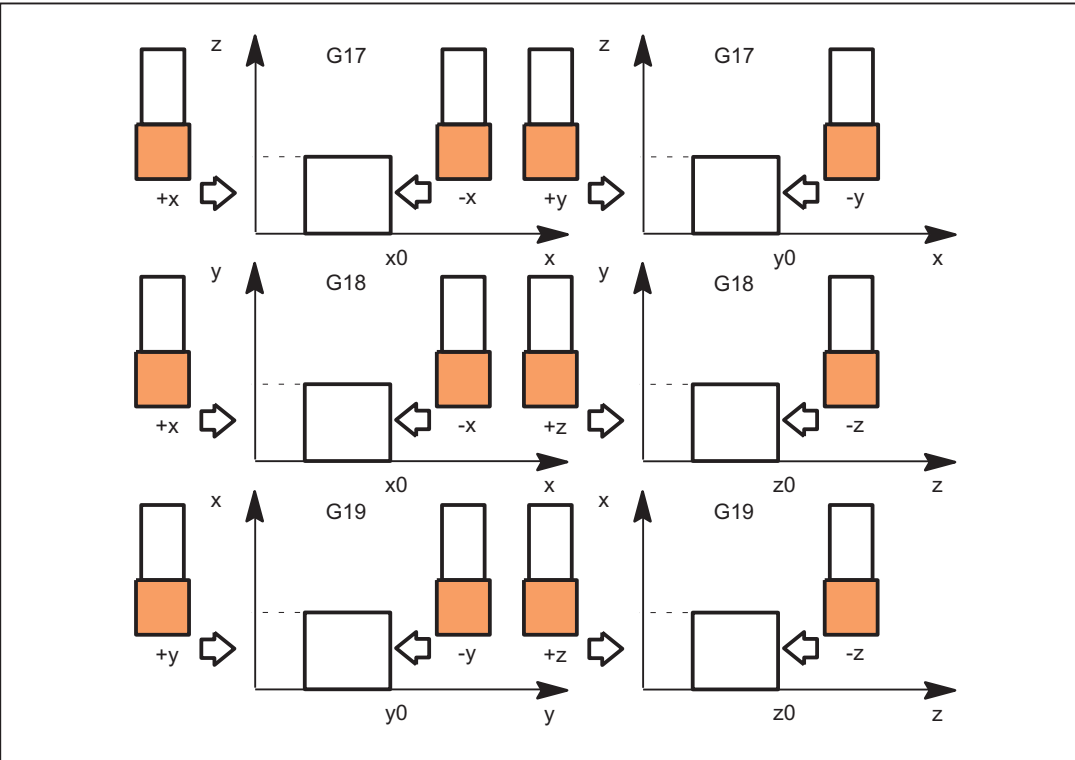


图 7-19 为选择的 G17、G18 和 G19 平面测量刀具直径

针对测量类型 11，系统会分析下列变量值

输入变量	含义
\$AC_MEAS_VALID	输入变量的有效位
\$AA_MEAS_POINT1[轴]	测量点 1
\$AA_MEAS_SETPPOINT[轴]	设定位置 x0
\$AC_MEAS_DIR_APPROACH	0: +x, 1: -x, 2: +y, 3: -y, 4: +z, 5: -z
\$AC_MEAS_ACT_PLANE	未设定时将采用当前生效的平面 *
\$AC_MEAS_TYPE	11

* 可选

针对测量类型 11，系统会写入以下输出变量

输出变量	含义
\$AC_MEAS_TOOL_DIAMETER	刀具直径

7.5.5.3 通过放大镜功能测量刀具长度（测量类型 22）

通过放大镜功能测量刀具长度

通过放大镜功能测量刀具长度（\$AC_MEAS_TYPE = 22）

如果机床配备了放大镜功能，可通过其测定刀具尺寸。

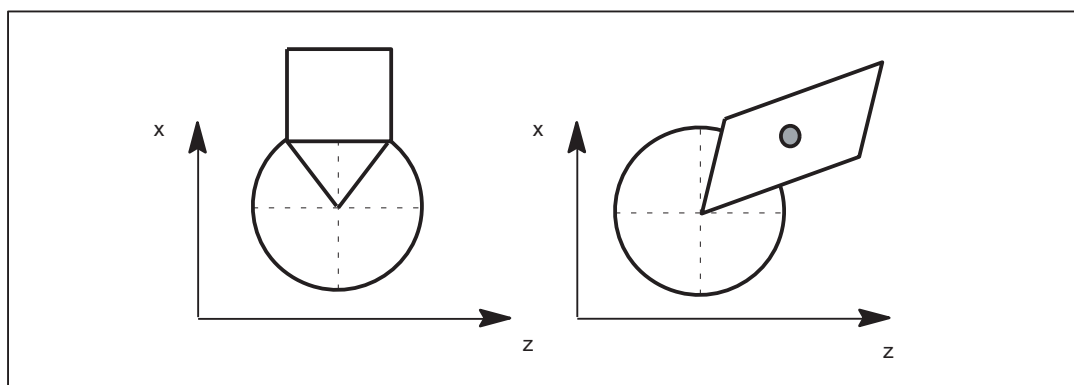


图 7-20 通过放大镜功能测量刀具长度

针对测量类型 22，系统会分析下列变量值

输入变量	含义
\$AC_MEAS_VALID	输入变量的有效位
\$AA_MEAS_POINT1[轴]	所有通道轴的第 1 测量点
\$AC_MEAS_P1_COORD	测量点的坐标系 *
\$AA_MEAS_SETPPOINT[轴]	必须设定放大镜位置 x 和 z
\$AC_MEAS_SET_COORD	设定点的坐标系 *
\$AC_MEAS_ACT_PLANE	未设定时将采用当前生效的平面 *

7.5 设置零点、工件测量和刀具测量

输入变量	含义
\$AC_MEAS_FRAME_SELECT	未设定时将计算叠加框架 *
\$AC_MEAS_T_NUMBER	未设定时将采用当前生效的 T (T0) *
\$AC_MEAS_D_NUMBER	未设定时将采用当前生效的 D (D0) *
\$AC_MEAS_TYPE	22

* 可选

针对测量类型 22，系统会写入以下输出变量

输出变量	含义
\$AC_MEAS_RESULT[0]	刀具长度, x 方向
\$AC_MEAS_RESULT[1]	刀具长度, y 方向
\$AC_MEAS_RESULT[2]	刀具长度, z 方向
\$AC_MEAS_RESULT[3]	刀具长度 L1
\$AC_MEAS_RESULT[4]	刀具长度 L2
\$AC_MEAS_RESULT[5]	刀具长度 L3

7.5.5.4 通过标记位置或当前位置测量刀具长度（测量类型 23）

通过标记位置或当前位置测量刀具长度

通过标记位置或当前位置测量刀具长度（\$AC_MEAS_TYPE = 23）

手动测量中可测定 X 轴方向和 Z 轴方向的刀具尺寸。ShopTurn 能够通过已知的

- 刀架参考点位置和
- 工件尺寸

ShopTurn 计算出刀具补偿数据。

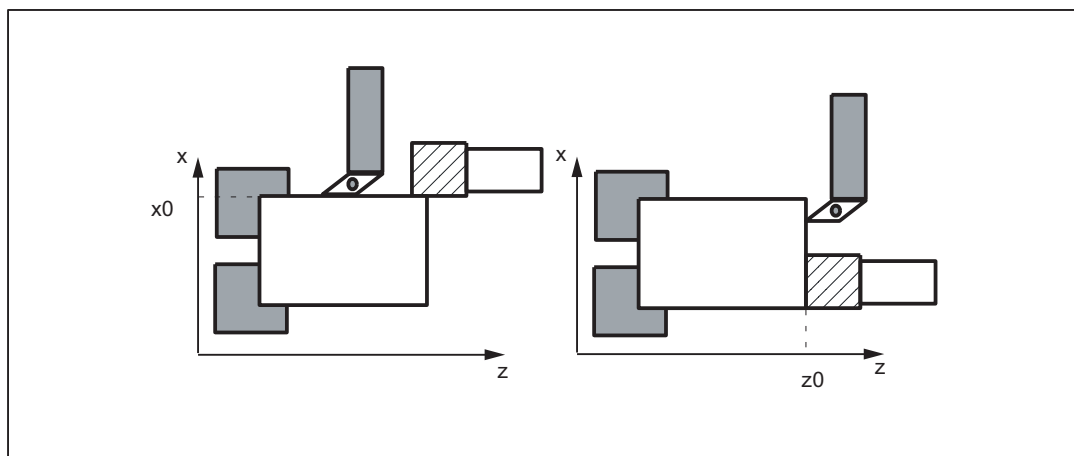


图 7-21 通过标记位置或当前位置测量刀具长度

针对测量类型 23，系统会分析下列输入变量值

输入变量	含义
\$AC_MEAS_VALID	输入变量的有效位
\$AA_MEAS_POINT1[轴]	当前位置或标记位置
\$AC_MEAS_P1_COORD	测量点的坐标系 *
\$AA_MEAS_SETPPOINT[轴]	设定位置（至少须设定一根几何轴）
\$AC_MEAS_SET_COORD	设定点的坐标系 *
\$AC_MEAS_ACT_PLANE	未设定时将采用当前生效的平面 *
\$AC_MEAS_T_NUMBER	未设定时将采用当前生效的 T（T0） *
\$AC_MEAS_D_NUMBER	未设定时将采用当前生效的 D（D0） *
\$AC_MEAS_TOOL_MASK	刀具长度，半径 *
\$AC_MEAS_DIR_APPROACH	逼近方向 *
\$AC_MEAS_INPUT[0] = 1	计算出的刀具长度写入数据管理 *
\$AC_MEAS_TYPE	23

* 可选

针对测量类型 23，系统会写入以下输出变量

输出变量	含义
\$AC_MEAS_RESULT[0]	刀具长度，x 方向
\$AC_MEAS_RESULT[1]	刀具长度，y 方向

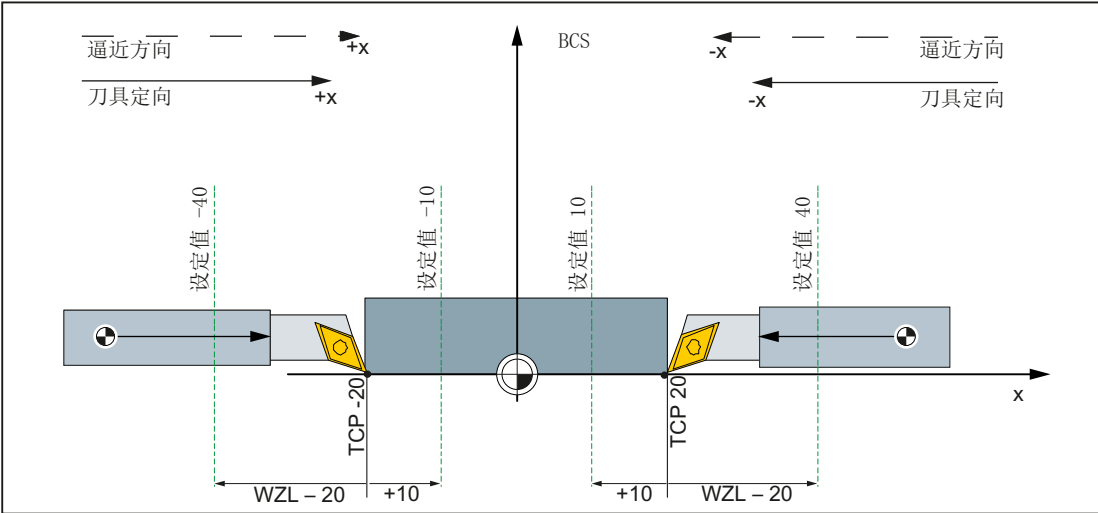
输出变量	含义
\$AC_MEAS_RESULT[2]	刀具长度，z 方向
\$AC_MEAS_RESULT[3]	刀具长度 L1
\$AC_MEAS_RESULT[4]	刀具长度 L2
\$AC_MEAS_RESULT[5]	刀具长度 L3

7.5.5.5 通过定向测量两把刀具的长度

刀具定向

对于向刀架定向的刀具，系统变量 \$AC_MEAS_TOOL_MASK 位 9 必须置 1（0x200）。之后计算出的刀具长度将取反计入。

启用各自参考点的两把车刀，刀具定向沿逼近方向



系统数据中的设置:

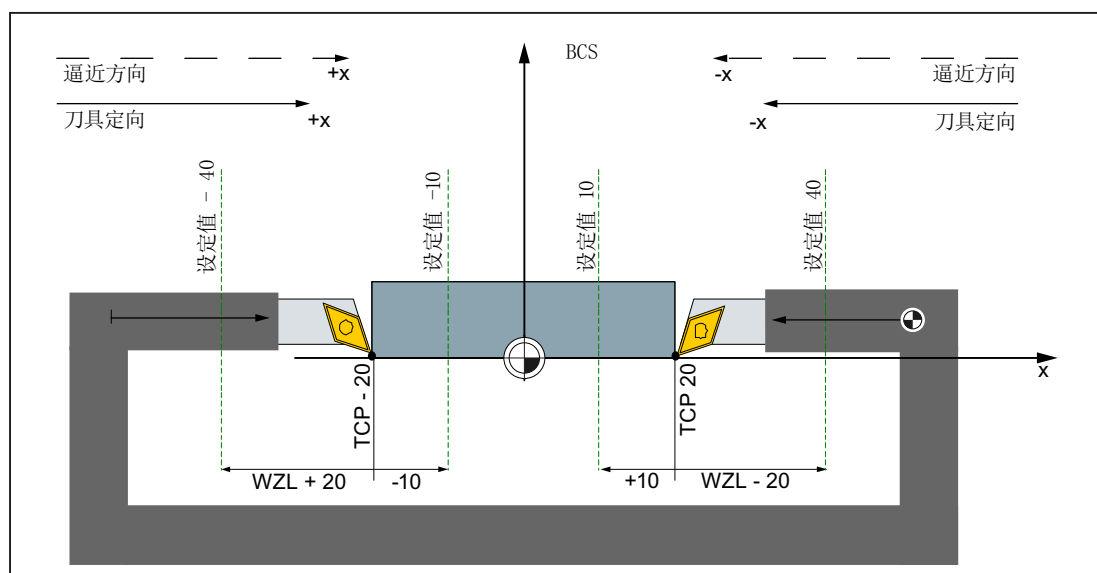
左侧刀具：逼近方向和刀具定向 +x	
系统变量	含义
\$AC_MEAS_TOOL_MASK = 0x2	刀具位置处于 x 轴正向（G19）
\$AC_MEAS_DIR_APPROACH = 0	逼近方向为 x 轴正向

7.5 设置零点、工件测量和刀具测量

右侧刀具：逼近方向和刀具定向 -x	
\$AC_MEAS_TOOL_MASK = 0x40	刀具位置处于 x 轴负向
\$AC_MEAS_DIR_APPROACH = 1	逼近方向为 x 轴负向

针对两把刀具	
\$AC_MEAS_Px_COORD = 1	第 x 个测量点的坐标系 = BCS
\$AC_MEAS_SET_COORD = 1	设定点的坐标系 = BCS

启用一个参考点的两把车刀，刀具定向与逼近方向相反

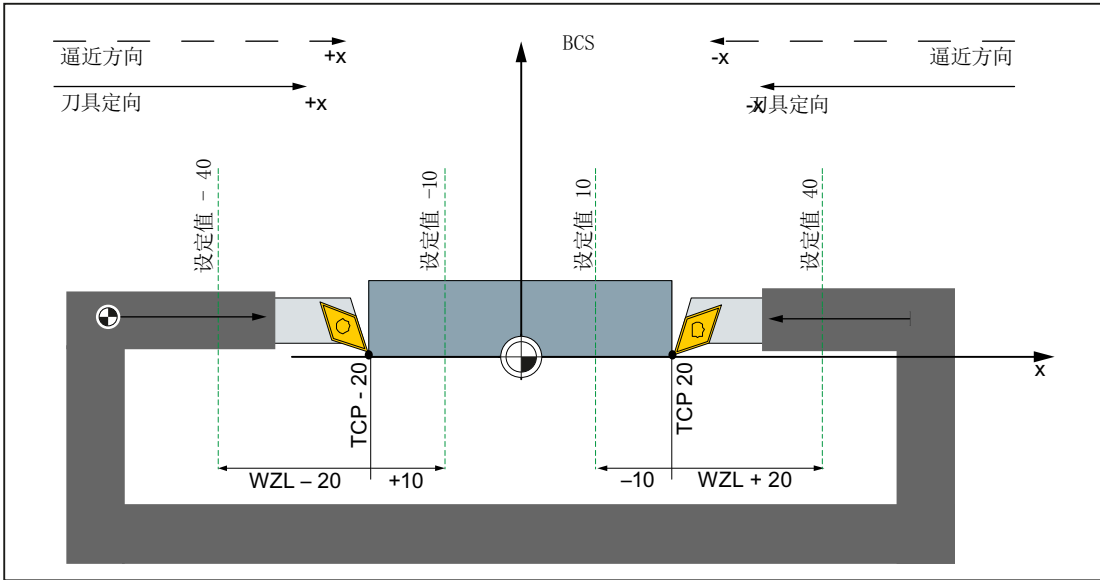


系统数据中的设置：

左侧刀具：逼近方向和刀具定向 +x	
系统变量	含义
\$AC_MEAS_TOOL_MASK = 0x2 + 0x200	刀具位置处于 x 轴正向 (G19) + 刀具长度差值取反计入
\$AC_MEAS_DIR_APPROACH = 0	逼近方向为 x 轴正向

右侧刀具：逼近方向和刀具定向 -x	
\$AC_MEAS_TOOL_MASK = 0x40	刀具位置处于 x 轴负向
\$AC_MEAS_DIR_APPROACH = 1	逼近方向为 x 轴负向

针对两把刀具	
\$AC_MEAS_Px_COORD = 1	第 x 个测量点的坐标系 = BCS
\$AC_MEAS_SET_COORD = 1	设定点的坐标系 = BCS



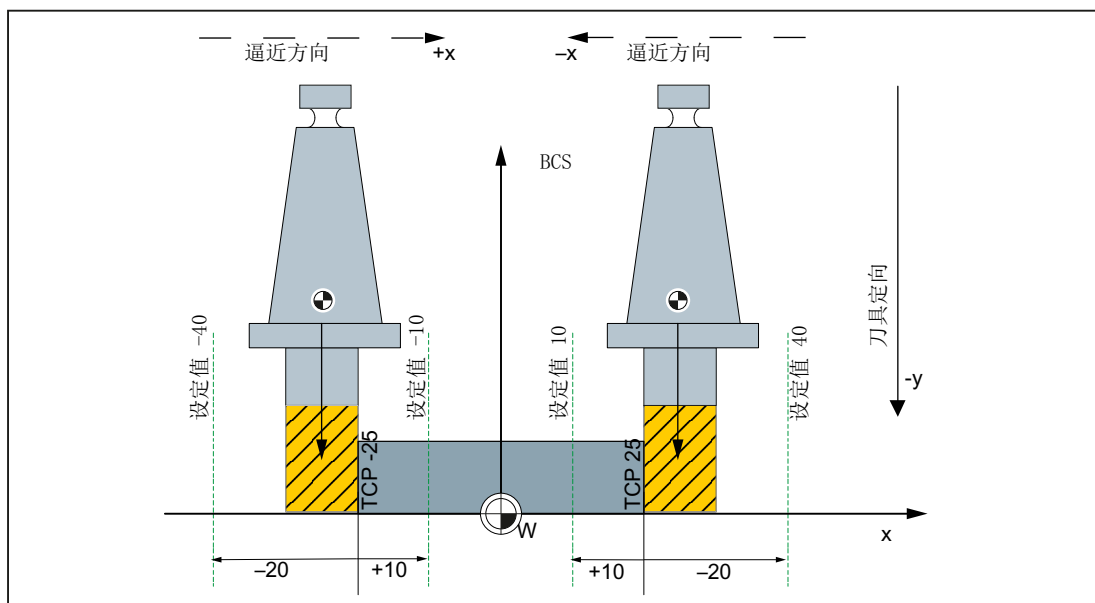
系统数据中的设置：

左侧刀具：逼近方向和刀具定向 +x	
系统变量	含义
\$AC_MEAS_TOOL_MASK = 0x2	刀具位置处于 x 轴正向（G19）
\$AC_MEAS_DIR_APPROACH = 0	逼近方向为 x 轴正向

右侧刀具：逼近方向和刀具定向 -x	
\$AC_MEAS_TOOL_MASK = 0x40 + 0x200	刀具位置处于 x 轴负向 + 刀具长度差值取反计入
\$AC_MEAS_DIR_APPROACH = 1	逼近方向为 x 轴负向

针对两把刀具	
\$AC_MEAS_Px_COORD = 1	第 x 个测量点的坐标系 = BCS
\$AC_MEAS_SET_COORD = 1	设定点的坐标系 = BCS

启用各自参考点的两把铣刀，刀具定向垂直于逼近方向



系统数据中的设置：

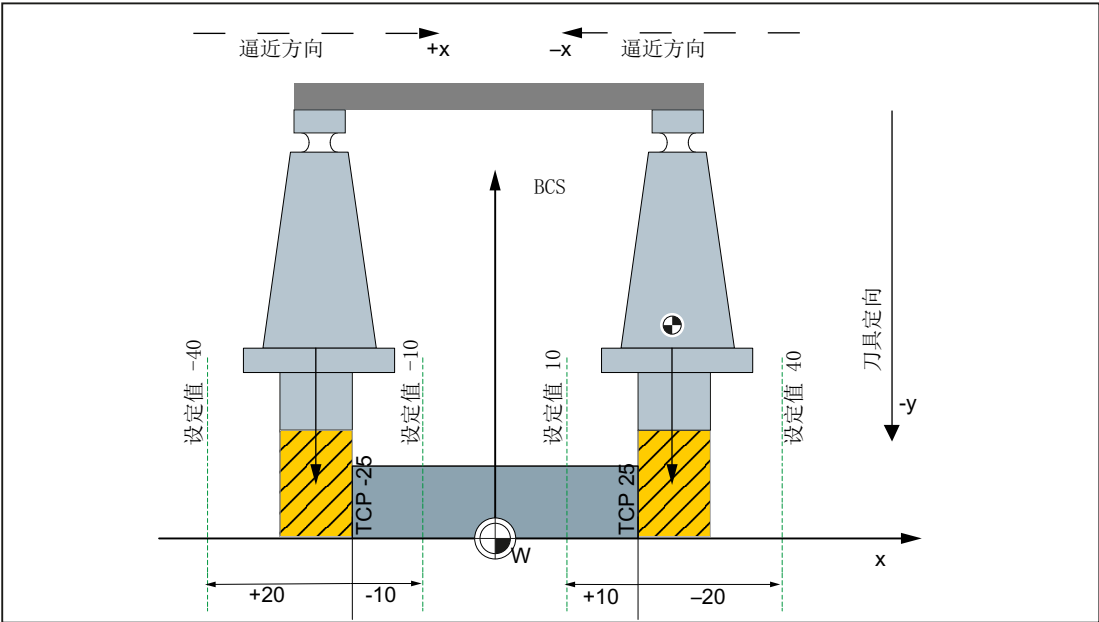
左侧刀具：逼近方向为 x 轴正向和刀具定向为 y 轴负向	
系统变量	含义
\$AC_MEAS_TOOL_MASK = 0x80	刀具位置处于 y 轴负向
\$AC_MEAS_DIR_APPROACH = 0	逼近方向为 x 轴正向

右侧刀具：逼近方向为 x 轴负向和刀具定向为 y 轴负向	
\$AC_MEAS_TOOL_MASK = 0x80	刀具位置处于 y 轴负向
\$AC_MEAS_DIR_APPROACH = 1	逼近方向为 x 轴负向

针对两把刀具	
\$AC_MEAS_Px_COORD = 1	第 x 个测量点的坐标系 = BCS
\$AC_MEAS_SET_COORD = 1	设定点的坐标系 = BCS

启用一个参考点的两把铣刀，刀具定向垂直于逼近方向

启用一个参考点的两把铣刀，刀具定向为 -y

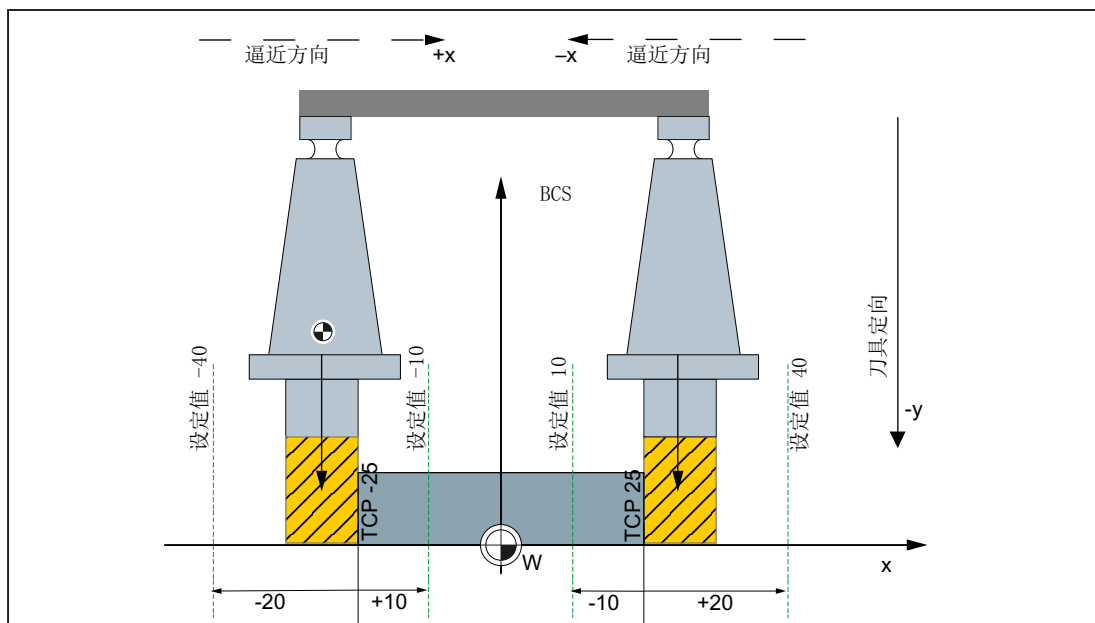


使用现有配置时，刀具位置 `$AC_MEAS_TOOL_MASK` 和向工件的逼近方向 `$AC_MEAS_DIR_APPROACH` 必须如下设置：

左侧刀具：逼近方向为 x 轴正向和刀具定向为 y 轴负向	
系统变量	含义
<code>\$AC_MEAS_TOOL_MASK = 0x80 + 0x200</code>	刀具位置处于 y 轴负向 + 刀具长度差值取反计入
<code>\$AC_MEAS_DIR_APPROACH = 0</code>	逼近方向为 x 轴正向

右侧刀具：逼近方向为 x 轴负向和刀具定向为 y 轴负向	
<code>\$AC_MEAS_TOOL_MASK = 0x80</code>	刀具位置处于 y 轴负向
<code>\$AC_MEAS_DIR_APPROACH = 1</code>	逼近方向为 x 轴负向

针对两把刀具	
<code>\$AC_MEAS_Px_COORD = 1</code>	第 x 个测量点的坐标系 = BCS
<code>\$AC_MEAS_SET_COORD = 1</code>	设定点的坐标系 = BCS



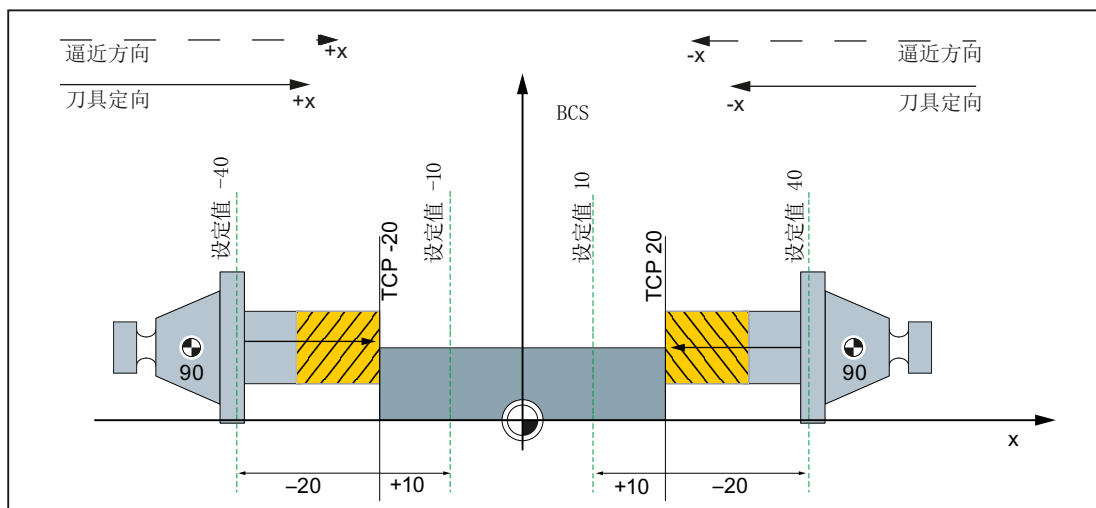
使用现有配置时，刀具位置 `$AC_MEAS_TOOL_MASK` 和向工件的逼近方向 `$AC_MEAS_DIR_APPROACH` 必须如下设置：

左侧刀具：逼近方向为 x 轴正向和刀具定向为 y 轴负向	
系统变量	含义
<code>\$AC_MEAS_TOOL_MASK = 0x80</code>	刀具位置处于 y 轴负向
<code>\$AC_MEAS_DIR_APPROACH = 0</code>	逼近方向为 x 轴正向

右侧刀具：逼近方向为 x 轴负向和刀具定向为 y 轴负向	
<code>\$AC_MEAS_TOOL_MASK = 0x80 + 0x200</code>	刀具位置处于 y 轴负向 + 刀具长度差值取反计入
<code>\$AC_MEAS_DIR_APPROACH = 1</code>	逼近方向为 x 轴负向

针对两把刀具	
<code>\$AC_MEAS_Px_COORD = 1</code>	第 x 个测量点的坐标系 = BCS
<code>\$AC_MEAS_SET_COORD = 1</code>	设定点的坐标系 = BCS

启用各自参考点的两把铣刀，刀具定向沿逼近方向

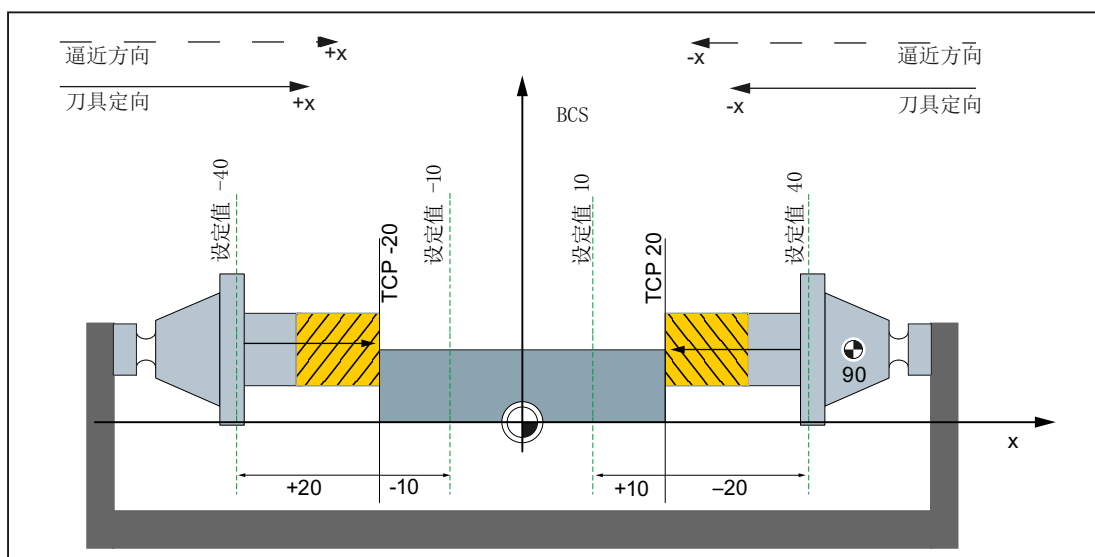


左侧刀具：逼近方向和刀具定向 +x	
系统变量	含义
\$AC_MEAS_TOOL_MASK = 0x2	刀具位置处于 x 轴正向（G19）
\$AC_MEAS_DIR_APPROACH = 0	逼近方向为 x 轴正向

右侧刀具：逼近方向和刀具定向 -x	
\$AC_MEAS_TOOL_MASK = 0x40	刀具位置处于 x 轴负向
\$AC_MEAS_DIR_APPROACH = 1	逼近方向为 x 轴负向

针对两把刀具	
\$AC_MEAS_Px_COORD = 1	第 x 个测量点的坐标系 = BCS
\$AC_MEAS_SET_COORD = 1	设定点的坐标系 = BCS

启用一个参考点的两把铣刀，刀具位置与定向相反

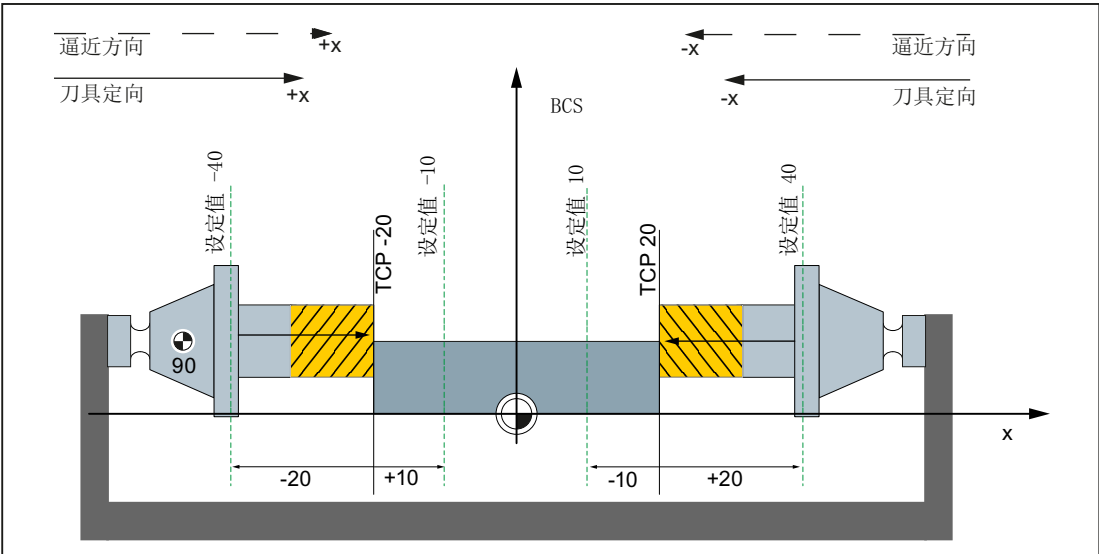


使用现有配置时，刀具位置 `$AC_MEAS_TOOL_MASK` 和向工件的逼近方向 `$AC_MEAS_DIR_APPROACH` 必须如下设置：

左侧刀具：逼近方向和刀具定向 +x	
系统变量	含义
\$AC_MEAS_TOOL_MASK = 0x2 + 0x200	刀具位置处于 x 轴正向（G19）+ 刀具长度差值取反计入
\$AC_MEAS_DIR_APPROACH = 0	逼近方向为 x 轴正向

右侧刀具：逼近方向和刀具定向为 x 轴负向	
\$AC_MEAS_TOOL_MASK = 0x40	刀具位置处于 x 轴负向
\$AC_MEAS_DIR_APPROACH = 1	逼近方向为 x 轴负向

针对两把刀具	
\$AC_MEAS_Px_COORD = 1	第 x 个测量点的坐标系 = BCS
\$AC_MEAS_SET_COORD = 1	设定点的坐标系 = BCS



使用现有配置时，刀具位置 `$AC_MEAS_TOOL_MASK` 和向工件的逼近方向 `$AC_MEAS_DIR_APPROACH` 必须如下设置：

左侧刀具：逼近方向和刀具定向 +x	
系统变量	含义
<code>\$AC_MEAS_TOOL_MASK = 0x2</code>	刀具位置处于 x 轴正向（G19）
<code>\$AC_MEAS_DIR_APPROACH = 0</code>	逼近方向为 x 轴正向

右侧刀具：逼近方向和刀具定向为 x 轴负向	
<code>\$AC_MEAS_TOOL_MASK = 0x40 + 0x200</code>	刀具位置处于 x 轴负向 + 刀具长度差值取反计入
<code>\$AC_MEAS_DIR_APPROACH = 1</code>	逼近方向为 x 轴负向

针对两把刀具	
<code>\$AC_MEAS_Px_COORD = 1</code>	第 x 个测量点的坐标系 = BCS
<code>\$AC_MEAS_SET_COORD = 1</code>	设定点的坐标系 = BCS

WCS 中的各种刀具

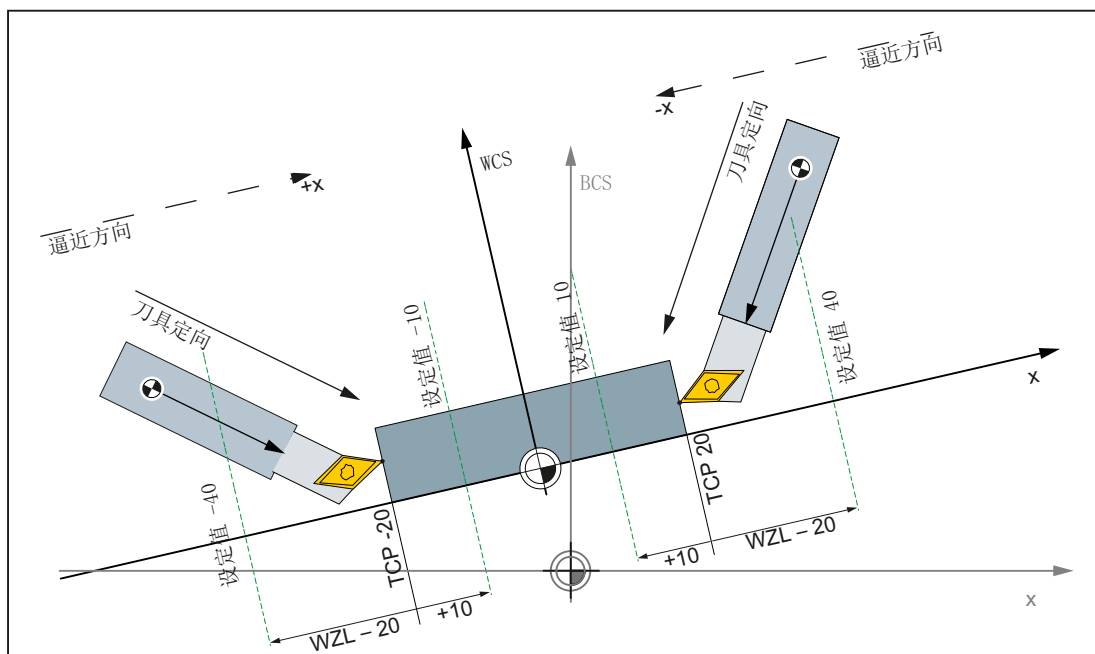


图 7-22 启用各自参考点的两把车刀

系统数据中的设置：

左侧刀具：逼近方向为 x 轴正向和刀具定向为 y 轴负向	
系统变量	含义
\$AC_MEAS_TOOL_MASK = 0x0	所有刀具长度均计入 (缺省设置)
\$AC_MEAS_DIR_APPROACH = 0	逼近方向为 x 轴正向

右侧刀具：逼近方向为 x 轴负向和刀具定向为 y 轴负向	
系统变量	含义
\$AC_MEAS_TOOL_MASK = 0x0	所有刀具长度均计入 (缺省设置)
\$AC_MEAS_DIR_APPROACH = 1	逼近方向为 x 轴负向

针对两把刀具	
\$AC_MEAS_Px_COORD = 0	第 x 个测量点的坐标系 = MKS (缺省设置)
\$AC_MEAS_SET_COORD = 0	设定点的坐标系 = WCS (缺省设置)

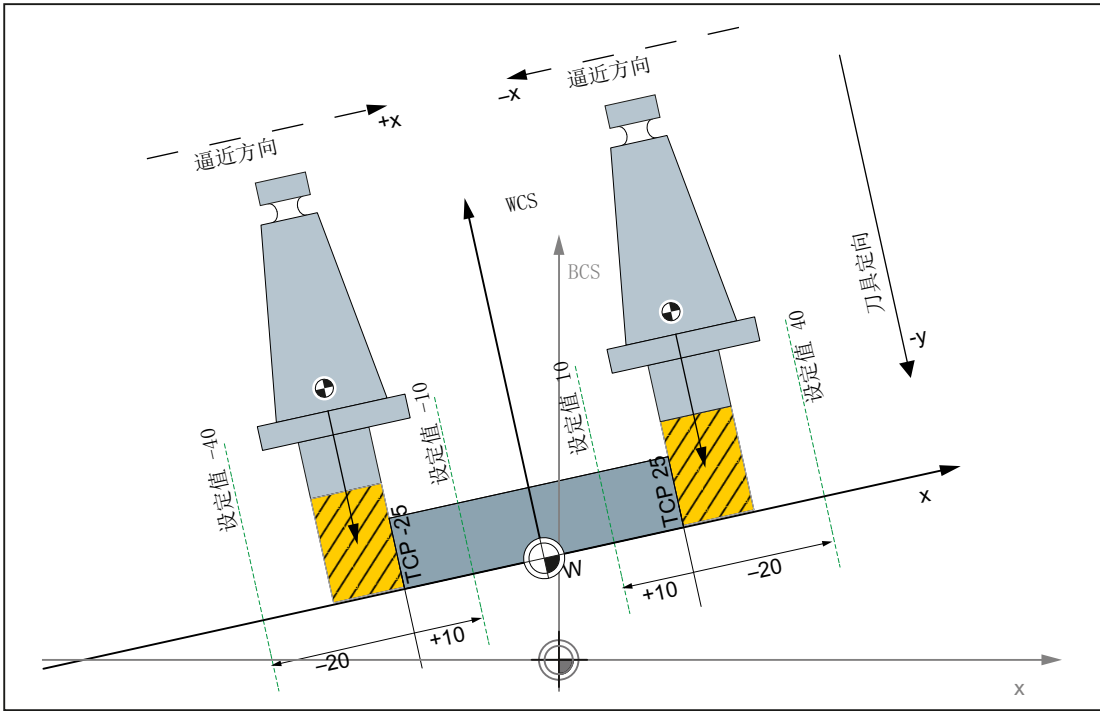


图 7-23 启用各自参考点的两把铣刀

系统数据中的设置:

左侧刀具: 逼近方向为 x 轴正向和刀具定向为 y 轴负向	
系统变量	含义
\$AC_MEAS_TOOL_MASK = 0x80	刀具位置处于 y 轴负向
\$AC_MEAS_DIR_APPROACH = 0	逼近方向为 x 轴正向

右侧刀具: 逼近方向为 x 轴负向和刀具定向为 y 轴负向	
\$AC_MEAS_TOOL_MASK = 0x80	刀具位置处于 y 轴负向
\$AC_MEAS_DIR_APPROACH = 1	逼近方向为 x 轴负向

针对两把刀具	
\$AC_MEAS_Px_COORD = 0	第 x 个测量点的坐标系 = MKS (缺省设置)
\$AC_MEAS_SET_COORD = 0	设定点的坐标系 = WCS (缺省设置)

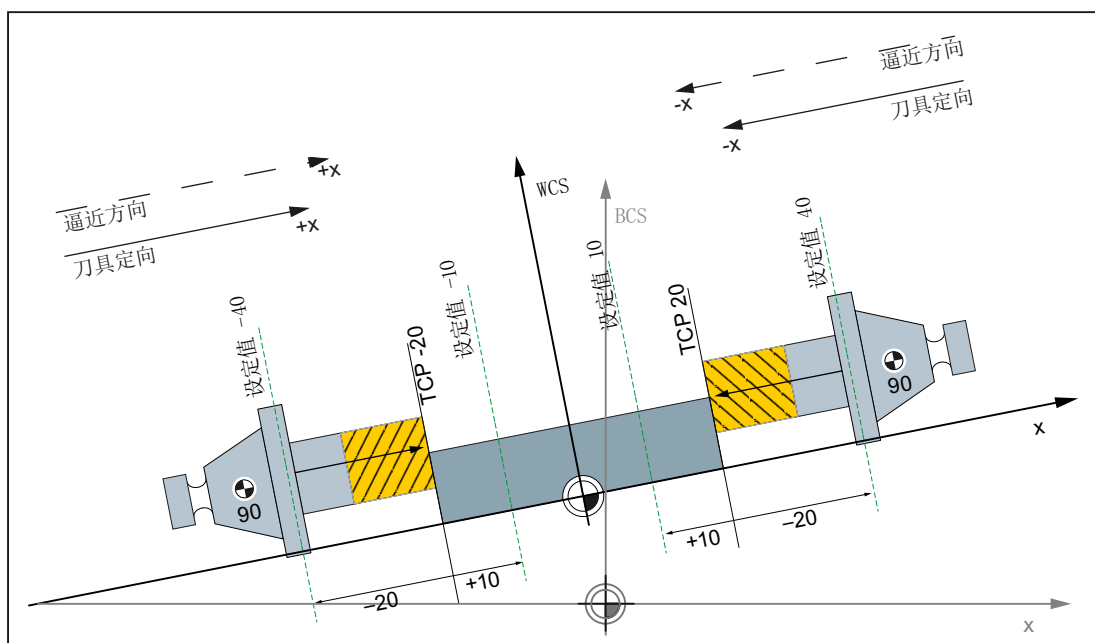


图 7-24 旋转 90 度、启用各自参考点的两把铣刀

系统数据中的设置:

左侧刀具: 逼近方向为 x 轴正向和刀具定向为 y 轴负向	
系统变量	含义
\$AC_MEAS_TOOL_MASK = 0x2	刀具位置处于 x 轴正向 (G19)
\$AC_MEAS_DIR_APPROACH = 0	逼近方向为 x 轴正向
右侧刀具: 逼近方向为 x 轴负向和刀具定向为 y 轴负向	
\$AC_MEAS_TOOL_MASK = 0x40	刀具位置处于 x 轴负向
\$AC_MEAS_DIR_APPROACH = 1	逼近方向为 x 轴负向

针对两把刀具	
\$AC_MEAS_Px_COORD = 0	第 x 个测量点的坐标系 = MKS (缺省设置)
\$AC_MEAS_SET_COORD = 0	设定点的坐标系 = WCS (缺省设置)

7.6 测量精度和测试

7.6.1 测量精度

测量精度受下列参数影响:

- 测量信号延时 (T_{Delay})
- 测量过程中的运行速度 (v_M)

测量信号延时补偿 (T_{Delay})

测量信号延时 (即从触发测头到将测量值保存到控制系统中的时间) 取决于测头的响应时间及控制系统硬件的信号响应时间。测量时会对控制系统延时进行补偿。为此, 须算出延时并将结果输入到以下机床数据中:

MD13220 \$MN_MEAS_PROBE_DELAY_TIME = <算出的延时>

说明

最大补偿延时 $T_{MaxDelay}$

$T_{MaxDelay} = 15 * \text{位置控制周期或 DP 周期}$

对超过 $T_{MaxDelay}$ 的延时进行补偿没有任何的意义, 因此, 超限的延时会被限制为 $T_{MaxDelay}$ 。

测量过程中的最大运行速度 (v_M)

测量过程中允许的最大运行速度取决于编写的测量脉冲沿的数量以及位置控制周期与 DP 周期之间的比例。

为得到正确的结果, 测量过程中必须选择满足以下条件 (每两个位置控制周期或 DP 周期) 的运行速度:

- 最多一个相同的触发信号, 即: 一个测头有一个正脉冲沿或一个负脉冲沿。
- 最多四个不同的触发信号, 即: 两个测头各有一个正脉冲沿和一个负脉冲沿。

7.6.2 测头功能测试

功能测试示例

表格 7-8

程序代码	注释
%_N_PRUEF_MESSTASTER_MPF	
;\$PATH=/_N_MPF_DIR	
; 测头连接的检查程序	
N05 DEF INT MTSIGNAL	; 控制状态的标记
N10 DEF INT ME_NR=1	; 测量输入编号
N20 DEF REAL MESSWERT_IN_X	
N30 G17 T1 D1	; 为测头预先选择
	; 刀具补偿
N40 _ANF: G0 G90 X0 F150	; 起始位置和
	; 测量速度
N50 MEAS=ME_NR G1 X100	; 在 x 轴中的测量输入 1 上
	; 进行测量
N60 STOPRE	
N70 MTSIGNAL=\$AC_MEA[1]	; 读取测量输入 1 上的
	; 软件切换信号
N80 IF MTSIGNAL == 0 GOTO _FEHL1	; 信号分析
N90 MESSWERT_IN_X=\$AA_MW[X]	; 读取工件坐标系中的
	; 测量值
N95 M0	
N100 M02	
N110 _FEHL1: MSG ("测头未切换!")	
N120 M0	
N130 M02	

7.7 仿真测量

7.7.1 一般功能

简要说明

在实际机床上进行测量时，需要连接测头以在特定位置提供切换信号。仿真环境中的测量则不需要使用测头，此时切换位置通过其他方式给定。

- 仿真测量支持两种切换位置设定方式：
- 位置相关的切换请求：切换位置通过测量程序段中编写的轴结束位置推导得出。
 - 外部切换请求：切换位置通过控制数字量输出来确定。

前提条件

若需执行仿真测量，系统中的所有机床轴都必须编写作为仿真轴：

- MD30130 \$MA_CTRLOUT_TYPE[轴] = 0（仿真设定值）
- MD30240 \$MA_ENC_TYPE[轴] = 0（仿真编码器）

7.7.2 位置相关切换请求

功能

“位置相关切换请求”通过以下 NCK 专用机床数据选择：

- MD13230 \$MN_MEAS_PROBE_SOURCE = 0
- MD13231 \$MN_MEAS_PROBE_OFFSET = <位置偏移>

轴切换位置通过测量程序段中编写的轴结束位置和参数设置的位置偏移计算得出：

切换位置[轴] = 结束位置[轴] - 位置偏移

测量程序段中会循环检查是否到达轴的切换位置：

设定位置[轴] ≥ 切换位置[轴]

到达切换位置时，系统会为测头 1 和测头 2 生成切换信号上升沿。一个位置控制周期后将生成下降沿。

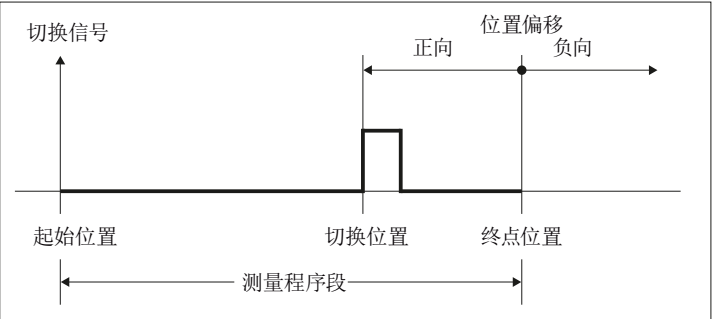


图 7-25 位置相关切换请求

测量值为测量程序段中编写的切换信号（上升沿/下降沿）出现时的轴实际值。

若在一个测量程序段中编写了多根轴，则会通过针对特定轴的位置偏移为每根轴生成一个独立的切换位置。测头信号在到达第一个轴切换位置时生成。

说明

测头信号

为测头 1 和测头 2 生成的测头信号总是相同。

负偏移值

位置偏移输入负值时，切换位置将推移到结束位置后。在此情形下不会生成测头信号。

示例

位置偏移设为 0.1 mm: MD13231 \$MN_MEAS_PROBE_OFFSET = 0.1

示例 1: 2 根轴中的通道专用测量

程序代码	注释
N10 G01 G90	
N20 MEAS=1 X100 Y10 F100	; 上升沿, 测头 1
	; 切换位置[X] = 99.9
	; 切换位置[Y] = 9.9

示例 2: 通过同步动作进行轴测量

程序代码	注释
N10 G01 G90	
N15 WHEN TRUE DO MEASA[X]=(1,1)	; 上升沿, 测头 1
N20 X10 F100	; 切换位置[X] = 9.9

7.7.3 外部切换请求

功能

在以下 NCK 专用机床数据中输入使用的数字量输出的编号（1...8），从而选择“外部切换请求”：

- MD13230 \$MN_MEAS_PROBE_SOURCE = <数字量输出的编号>

测头信号通过控制配置过的数字量输出来触发。测量输入和数字量输出之间不需要硬件接线。

系统会通过置位数字量输出来为测头 1 和测头 2 生成切换信号上升沿。下降沿则通过数字量输出复位生成。

测量值为测量程序段中编写的切换信号（上升沿/下降沿）出现时的轴实际值。

数字量输出：配置

为了能将数字量输出用于仿真测量，必须设置以下机床数据：

- MD10360 \$MN_FASTIO_DIG_NUM_OUTPUTS = 1（生效的 NCK 数字量输出字节的数量）
- MD13120 \$MN_CONTROL_UNIT_LOGIC_ADDRESS = 0（SINAMICS-CU 逻辑地址）

数字量输出：置位

配置的数字量输出可在同步动作中置位：

WHEN <条件> DO \$A_OUT[<数字量输出的编号>] = 1

示例

使用的数字量输出：MD13230 \$MN_MEAS_PROBE_SOURCE = 1

示例 1：2 根轴中的通道专用测量

程序代码	注释
N10 G01 G90 \$A_OUT[1]=0	; 预占用数字量输出 1
N15 WHEN \$AC_DETW<=10 DO \$A_OUT[1]=1	; 轨迹剩余行程 <= 10 => 数字量输出 1 = 1
N20 MEAS=1 X100 Y10 F100	; 上升沿，测头 1

示例 2：轴测量

程序代码	注释
N10 G01 G90 \$A_OUT[1]=0	; 预占用数字量输出 1
N15 WHEN \$AA_IW[X]>=80 DO \$A_OUT[1]=1	; 轴设定值 >= 80 => 数字量输出 1 = 1
N20 MEASA[X]=(1,1) X100 F100	; 上升沿, 测头 1

7.7.4 系统变量

使用仿真测量时，下列系统变量的功能与实际测量时相同：

- \$AC_MEA（测头已切换）
- \$AA_MEAACT（轴测量生效）
- \$AA_MM（获取的测头位置（MCS））
- \$AA_MM1...4（测头位置，第 1 次 - 第 4 次触发（MCS））
- \$AA_MW（获取的测头位置（WCS））
- \$AA_MW1...4（测头位置，第 1 次触发（WCS））

以下系统变量提供的值无意义：

- \$A_PROBE（测头状态）

7.8 示例 - 仅用于 840D sl

7.8.1 测量模式 1

边界条件

- 一次性测量
- 1 测头
- 触发信号为上升沿和下降沿

使用 1 个编码器测量 - 当前编码器的实际值

程序代码
N2 MEASA[X] = (1, 1, -1) G01 X100 F100
N3 STOPRE
N4 IF \$AC_MEA[1]==FALSE gotof ENDE
N5 R10=\$AA_MM1[X]

程序代码	
N6	R11=\$AA_MM2[X]
N7	END:

使用 2 个编码器测量 - 2 个编码器的实际值

程序代码	
N2	MEASA[X]=(31, 1, -1) G01 X100 F100
N3	STOPRE
N4	IF \$AC_MEA[1]==FALSE gotof ENDE
N5	R10=\$AA_MM1[X]
N6	R11=\$AA_MM2[X]
N7	R12=\$AA_MM3[X]
N8	R13=\$AA_MM4[X]
N9	END:

7.8.2 测量模式 2

边界条件

- 2 测头
- 触发信号为上升沿和下降沿
- 当前编码器的实际值

程序代码	
N2	MEASA[X] = (2, 1, -1, 2, -2) G01 X100 F100
N3	STOPRE
N4	IF \$AC_MEA[1]==FALSE gotof MESSTASTER2
N5	R10=\$AA_MM1[X]
N6	R11=\$AA_MM2[X]
N7	MESSTASTER2
N8	IF \$AC_MEA[2]==FALSE gotof ENDE
N9	R12=\$AA_MM3[X]
N10	R13=\$AA_MM4[X]
N11	END:

7.8.3 连续测量

边界条件

- 在测量模式 1 中进行测量
- 测量 100 个值
- 1 测头
- 触发信号为下降沿
- 当前编码器的实际值

在编写的运行结束后连续测量

程序代码	注释
N1 DEF REAL MESSWERT[100]	
N2 DEF INT INDEX=0	
N3 MEAC[X]=(1, 1, -1) G01 X1000 F100	
N4 MEAC[X]=(0)	; 终止
N5 R1=\$AC_FIFO1[4]	; 测量值的数量
N6 FOR INDEX=0 TO R1	
N7 MESSWERT[INDEX]=\$AC_FIFO1[0]	; 读取测量值
N8 ENDFOR:	

删除剩余行程的连续测量

在最后一次测量后删除剩余行程。

程序代码	注释
N1 DEF INT ANZAHL=100	
N2 DEF REAL MESSWERT[ANZAHL]	
N3 DEF INT INDEX=0	
N4 WHEN \$AC_FIFO1[4]==ANZAHL DO DELDTG (X) MEAC[X] =(0)	
N5 MEAC[X]=(1, 1, -1) G01 X1000 F100	; 开始测量
N6 R1=\$AC_FIFO1[4]	; 测量值的数量
N7 FOR INDEX=0 TO R1	
N8 MESSWERT[INDEX]=\$AC_FIFO1[0]	; 读取测量值
N9 ENDFOR:	

通过多个程序段以模态方式连续测量

程序代码	注释
N1 DEF INT ANZAHL=100	
N2 DEF REAL MESSWERT[ANZAHL]	
N3 DEF INT INDEX=0	
N4 ID=1 MEAC[X]=(1, 1, -1)	; 开始测量
N5 ID=2 WHEN \$AC_FIFO1[4]==ANZAHL DO MEAC[X]=(0)	
CANCEL(2)	
N6 G01 X1000 Y100	
N7 X100 Y100	
N8 R1=\$AC_FIFO1[4]	; 测量值的数量
N9 FOR INDEX=0 TO R1	
N10 MESSWERT[INDEX]=\$AC_FIFO1[0]	; 读取测量值
N11 ENDFOR:	

7.8.4 功能测试和重复精度

功能测试

程序代码	注释
%_N_PRUEF_MESSTASTER_MPF	
;\$PATH=/_N_MPF_DIR	
; 测头连接的检查程序	
N05 DEF INT MTSIGNAL	; 控制状态的标记
N10 DEF INT ME_NR=1	; 测量输入编号
N20 DEF REAL MESSWERT_IN_X	
N30 G17 T1 D1	; 为测头预先选择
	; 刀具补偿
N40 _ANF: G0 G90 X0 F150	; 起始位置和
	; 测量速度
N50 MEAS=ME_NR G1 X100	; 在 x 轴中的测量输入 1 上
	; 进行测量
N60 STOPRE	
N70 MTSIGNAL=\$AC_MEA[1]	; 读取测量输入 1 上的
	; 软件切换信号
N80 IF MTSIGNAL == 0 GOTOF _FEHL1	; 信号分析
N90 MESSWERT_IN_X=\$AA_MW[X]	; 读取工件坐标系中的
	; 测量值
N95 M0	
N100 M02	

程序代码	注释
N110 _FEHL1: MSG (“测头未切换!”)	
N120 M0	
N130 M02	

重复精度

可通过此程序获取整个测量系统（机床-测头-对 NC 的信号传输）的测量偏差（重复精度）。

在此示例中会在 X 轴上进行 10 次测量，并将测量值以工件坐标接收。

这样可确定非趋势性的、偶然条件下的尺寸偏差。

程序代码	注释
% _N_PRUEF_GENAU_MPF;	
\$PATH=/_N_MPF_DIR	
N05 DEF INT SIGNAL, II	; 变量定义
N10 DEF REAL MESSWERT_IN_X[10]	
N15 G17 T1 D1	; 初始条件,
	; 刀具补偿
	; 为测头预先选择
N20 _ANF: G0 X0 F150 ←	; 在测量轴上预定位
N25 MEAS=+1 G1 X100 ←	; 在测量输入 1 上进行测量, 此时
	; 切换信号未偏转,
	; 在 x 轴中偏转
N30 STOPRE ←	; 在之后的结果分析后
	; 停止解码
N35 SIGNAL= \$AC_MEA[1]	; 读取测量输入 1 上的
	; 1. ; 软件切换信号
N37 IF SIGNAL == 0 GOTO _FEHL1	; 检查接通信号
N40 MESSWERT_IN_X[II]=\$AA_MW[X]	; 以工件坐标读取测量值
N50 II=II+1	
N60 IF II<10 GOTO _ANF	; 重复 10 次
N65 M0	
N70 M02	
N80 _FEHL1: MSG (“测头未切换!”)	
N90 M0	
N95 M02	

选择了参数显示（用户自定义变量）后，程序执行过程中都可在 MESSWERT_IN_X[10] 区域读取测量结果。

7.9 数据表

7.9.1 机床数据

7.9.1.1 通用机床数据

编号	名称: \$MN_	含义
13200	MEAS_PROBE_LOW_ACTIVE	测头切换性能
13201	MEAS_PROBE_SOURCE	通过数字量输出进行测量脉冲仿真
13210	MEAS_TYPE	PROFIBUS DP 驱动上的测量方式
13211	MEAS_CENTRAL_SOURCE	使用 PROFIBUS DP 驱动集中测量时的数据源

7.9.1.2 通道专用机床数据

编号	名称: \$MC_	含义
20360	TOOL_PARAMETER_DEF_MASK	刀具参数的定义
28264	LEN_AC_FIFO	FIFO 变量 \$AC_FIFO ... 的长度

7.9.2 系统变量

所有输入值列表

名称	含义
\$AC_FIFO1...10	FIFO 变量 1 至 10
\$AC_MEAS_SEMA	接口占用
\$AC_MEAS_VALID	输入值的有效位
\$AA_MEAS_POINT1	所有通道轴的第 1 测量点
\$AA_MEAS_POINT2	所有通道轴的第 2 测量点
\$AA_MEAS_POINT3	所有通道轴的第 3 测量点
\$AA_MEAS_POINT4	所有通道轴的第 4 测量点
\$AA_MEAS_SETPPOINT	所有通道轴的设定位置

名称	含义
\$AA_MEAS_SETANGLE	所有通道轴的设定角度
\$AC_MEAS_P1_COORD	第 1 测量点的坐标系
\$AC_MEAS_P2_COORD	第 2 测量点的坐标系
\$AC_MEAS_P3_COORD	第 3 测量点的坐标系
\$AC_MEAS_P4_COORD	第 4 测量点的坐标系
\$AC_MEAS_SET_COORD	设定点的坐标系
\$AC_MEAS_LATCH[0...3]	解锁 WCS 中的测量点
\$AA_MEAS_P1_VALID	解锁 WCS 中的第 1 测量点
\$AA_MEAS_P2_VALID	解锁 WCS 中的第 2 测量点
\$AA_MEAS_P3_VALID	解锁 WCS 中的第 3 测量点
\$AA_MEAS_P4_VALID	解锁 WCS 中的第 4 测量点
\$AA_MEAS_SP_VALID	将轴设定位置设置为有效
\$AC_MEAS_WP_SETANGLE	工件位置设定角度
\$AC_MEAS_CORNER_SETANGLE	拐角的设定交角
\$AC_MEAS_DIR_APPROACH	逼近方向
\$AC_MEAS_ACT_PLANE	针对工件的工作平面
\$AC_MEAS_SCALEUNIT	单位制 INCH / METRISCH
\$AC_MEAS_FINE_TRANS	补偿计入精偏
\$AC_MEAS_FRAME_SELECT	工件测量时的框架选择
\$AC_MEAS_CHSFR	框架链设置：系统框架
\$AC_MEAS_NCBFR	框架链设置：全局基本框架
\$AC_MEAS_CHBFR	框架链设置：通道基本框架
\$AC_MEAS_UIFR	框架链设置：可设定框架
\$AC_MEAS_PFRAME	框架链设置：程序框架
\$AC_MEAS_T_NUMBER	刀具选择
\$AC_MEAS_D_NUMBER	刀沿选择
\$AC_MEAS_TOOL_MASK	刀具设置
\$AC_MEAS_TYPE	测量类型
\$AC_MEAS_INPUT	测量输入参数

所有输出值列表

名称	含义
\$A_PROBE[1,2]	测头状态
\$A_PROBE_LIMITED[1,2]	超出测量速度
\$AC_MEAS[1,2]	测头已切换
\$AA_MM	获取的测头位置（MCS）
\$AA_MM1...4	测头位置，触发事件 1 至 4（MCS）
\$AA_MW	获取的测头位置（WCS）
\$AA_MW1...4	测头位置，触发事件 1 至 4（WCS）
\$AC_MEAS_FRAME	结果框架
\$AC_MEAS_WP_ANGLE	计算出的工件位置角
\$AC_MEAS_CORNER_ANGLE	计算出的交角
\$AC_MEAS_DIAMETER	计算出的直径
\$AC_MEAS_TOOL_LENGTH	计算出的刀具长度
\$AC_MEAS_RESULTS	测量结果（取决于测量类型）

N3: 软件挡块, 行程开关信号 - 仅适用于 840D sl

8.1 简要说明

功能

“软件挡块”功能可为提供位置实际值的真实轴和仿真轴生成位置相关的开关信号。这些挡块信号可输出到 PLC，也可另外输出到 NCK I/O。

设置信号输出的挡块位置可以通过设定数据定义和修改。设定数据可通过 HMI、PLC 和零件程序读写。

激活

“软件挡块”功能在所有运行方式下均可激活和生效。执行复位或急停后此功能仍保持生效。

应用范围

挡块信号例如可用于：

- 激活保护区域
- 触发和位置相关的辅助运动
- 控制液压往复轴运动

轴类型

软件挡块可在定义为机床轴的线性轴或模数轴上使用。

挡块范围/挡块对

挡块总是以挡块对（包含一个正挡块和一个负挡块）的形式指定给轴。系统提供 32 个挡块对供使用。

正挡块和负挡块均模拟一个机械挡块：轴在特定运行方向上压到挡块时，挡块会动作。

正负挡块和挡块范围的对应关系如下：

8.2 挡块信号和挡块位置

- 正挡块范围: $\geq$ 正挡块的所有位置
- 负挡块范围: $\leq$ 负挡块的所有位置

8.2 挡块信号和挡块位置

8.2.1 挡块信号单独输出

单独输出正挡块信号和负挡块信号, 可方便地辨别轴是否处于正挡块范围或负挡块范围内。

线性轴

挡块信号的脉冲沿输出取决于轴运行方向:

8.2 挡块信号和挡块位置

- 当轴朝正向移动越过负挡块时，负挡块信号会从 1 变 0。
- 当轴朝正向移动越过正挡块时，正挡块信号会从 0 变 1。

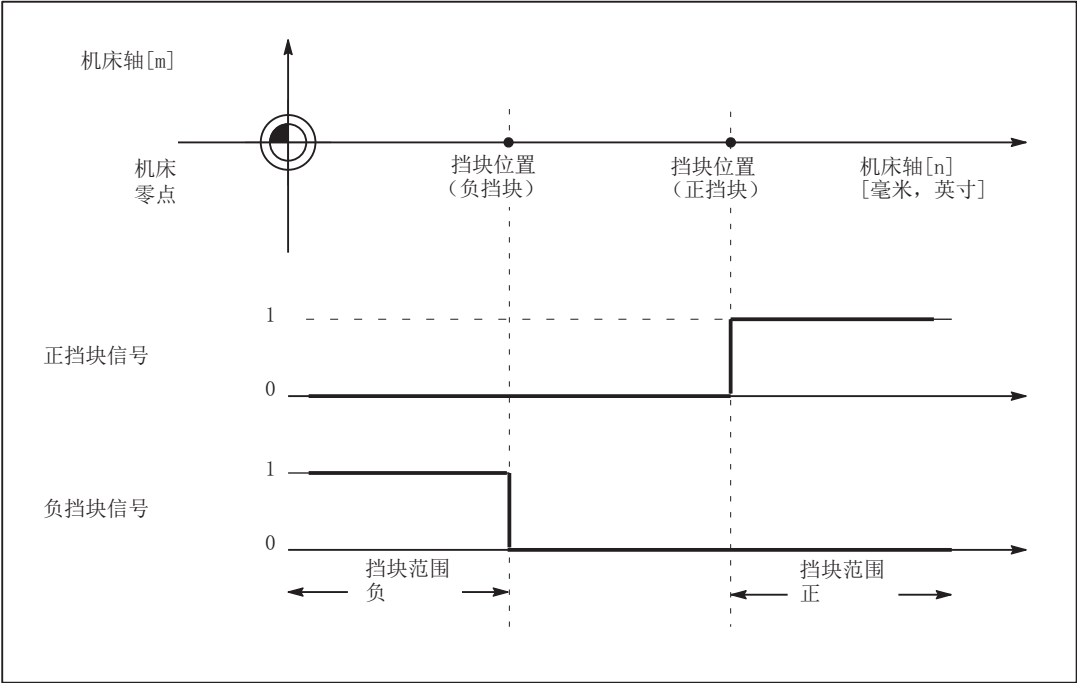


图 8-1 线性轴的软件挡块（负挡块 < 正挡块）

说明

若轴正好处于挡块位置上（正挡块或负挡块），那么定义的输出会出现闪变。此时若继续运行一个步长，该输出将置零或置一。
信号闪变由实际位置的闪变造成。
系统会分析实际位置。

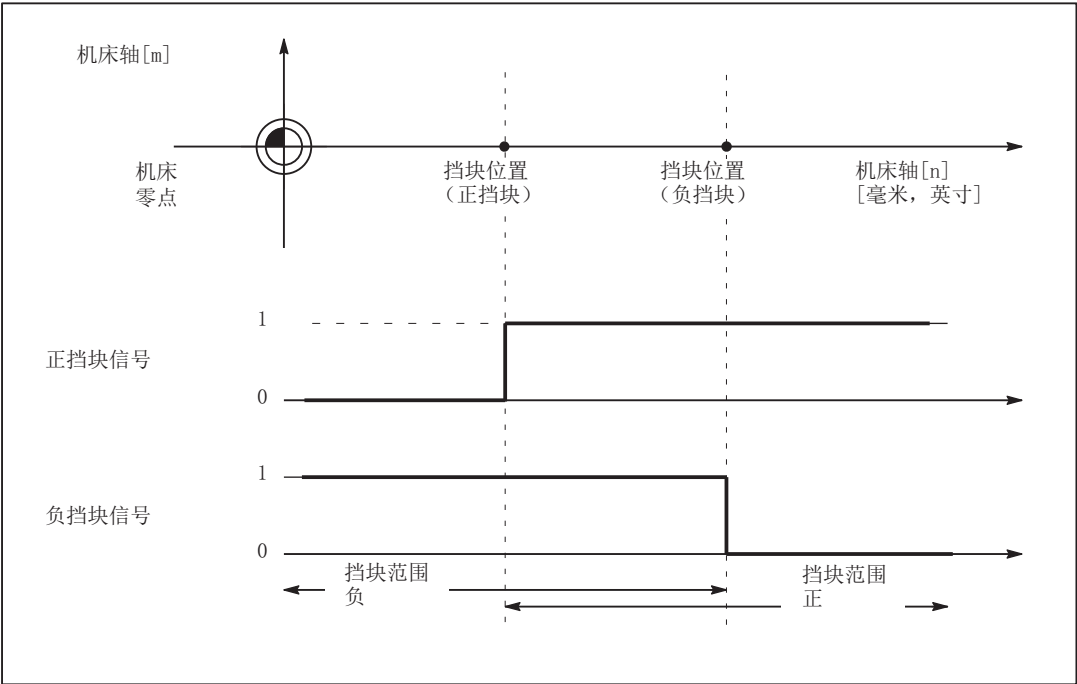


图 8-2 线性轴的软件挡块（正挡块 < 负挡块）

8.2 挡块信号和挡块位置

模数回转轴

挡块信号的脉冲沿输出取决于回转轴的旋转方向：

- 当回转轴正转越过正挡块时，正挡块信号会从 1 变 0；当回转轴正转越过负挡块时，正挡块信号会从 0 变 1。
- 每次正挡块信号输出上升沿时，负挡块信号便切换电平。

说明

此处描述的正挡块特性适用于以下条件：

正挡块 - 负挡块 < 180 度

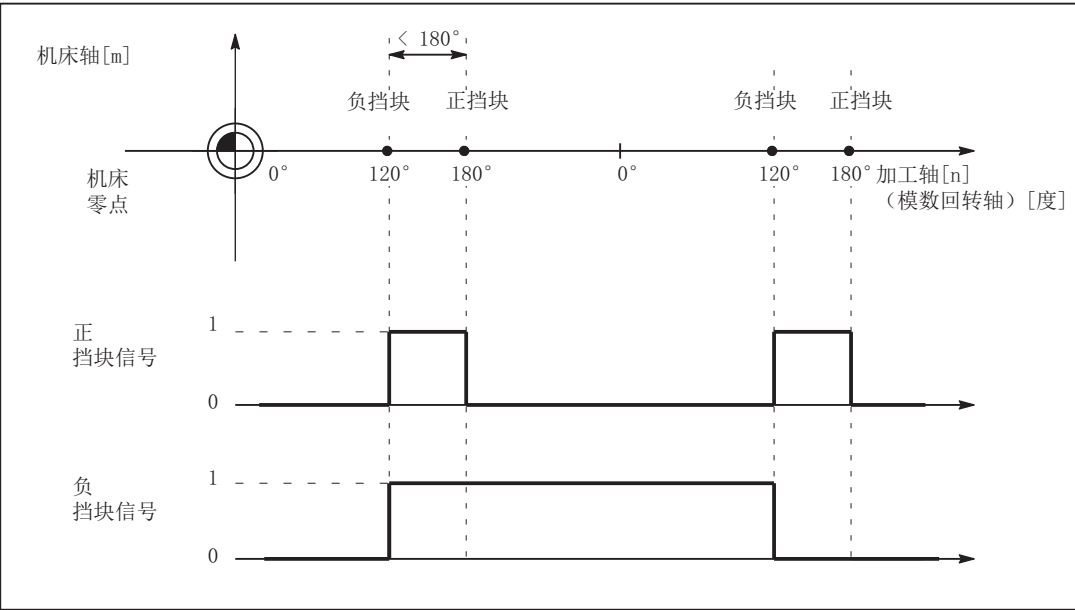


图 8-3 模数回转轴的软件挡块（正挡块 - 负挡块 < 180 度）

在挡块范围设置较小并无法由 PLC 可靠获取的情况下，仍可通过负挡块的信号切换识别出过挡块运行。

两个挡块信号均可输出至 PLC 和 NCK I/O。单独输出正挡块信号和负挡块信号，可方便地辨别轴是否处于正挡块范围或负挡块范围内。

如果不满足此条件（正挡块 - 负挡块 < 180 度）或者所选的负挡块大于正挡块，那么正挡块信号的特性将反转。负挡块信号的响应保持不变。

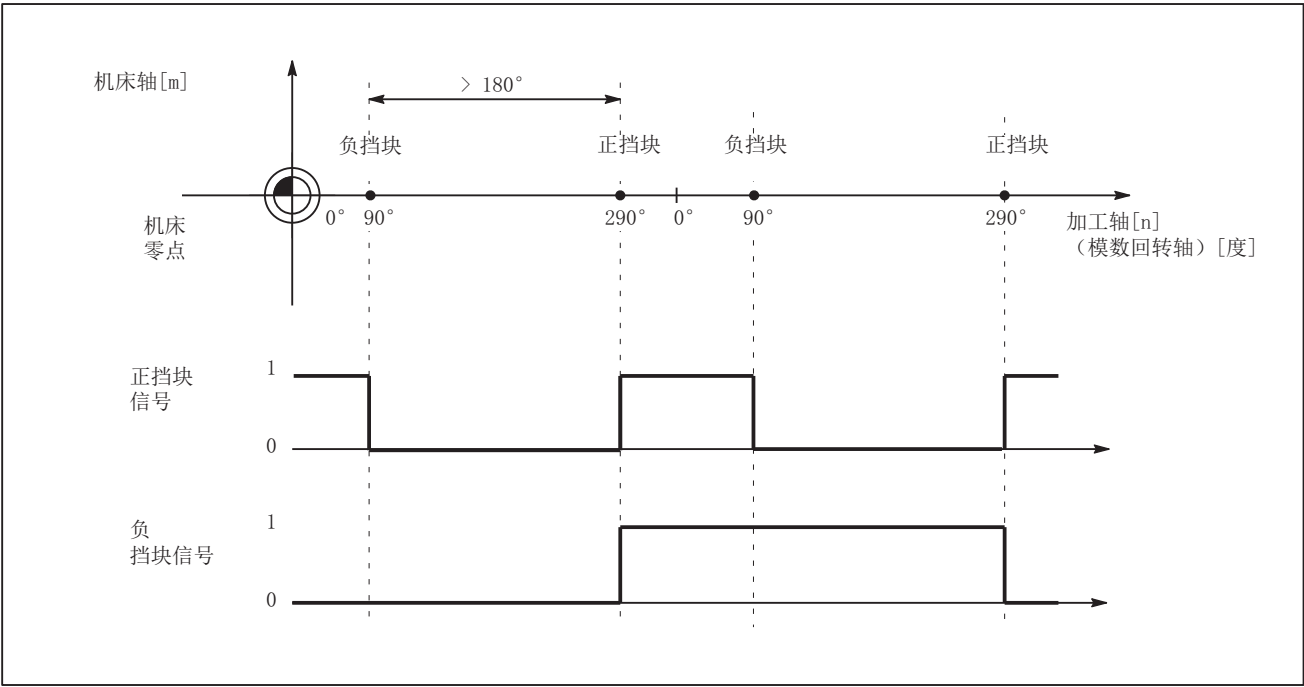


图 8-4 模数回转轴的软件挡块（正挡块 - 负挡块 > 180 度）

8.2.2 挡块信号关联输出

正挡块信号和负挡块信号的关联输出用于：

- 将挡块信号定时输出至 NCU 的 4 个板载输出
- 输出至 NCK I/O，当以下机床数据中未指定第 2 个字节时（=“0”）：
MD10470 SW_CAM_ASSIGN_FASTOUT_2
...
MD10473 SW_CAM_ASSIGN_FASTOUT_4

8.2 挡块信号和挡块位置

线性轴

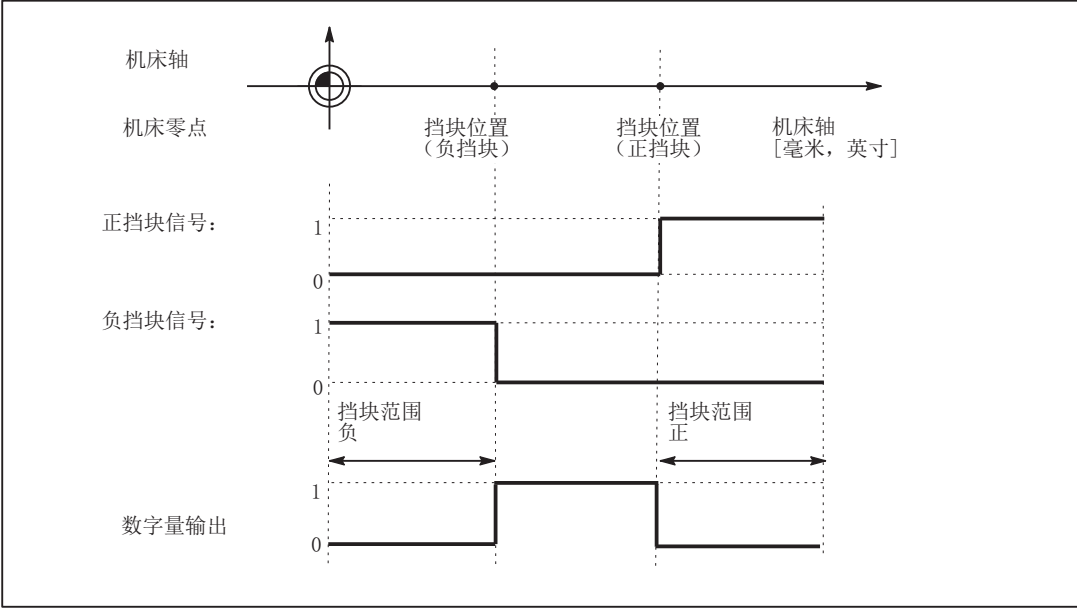


图 8-5 线性轴的行程开关信号（负挡块 < 正挡块）

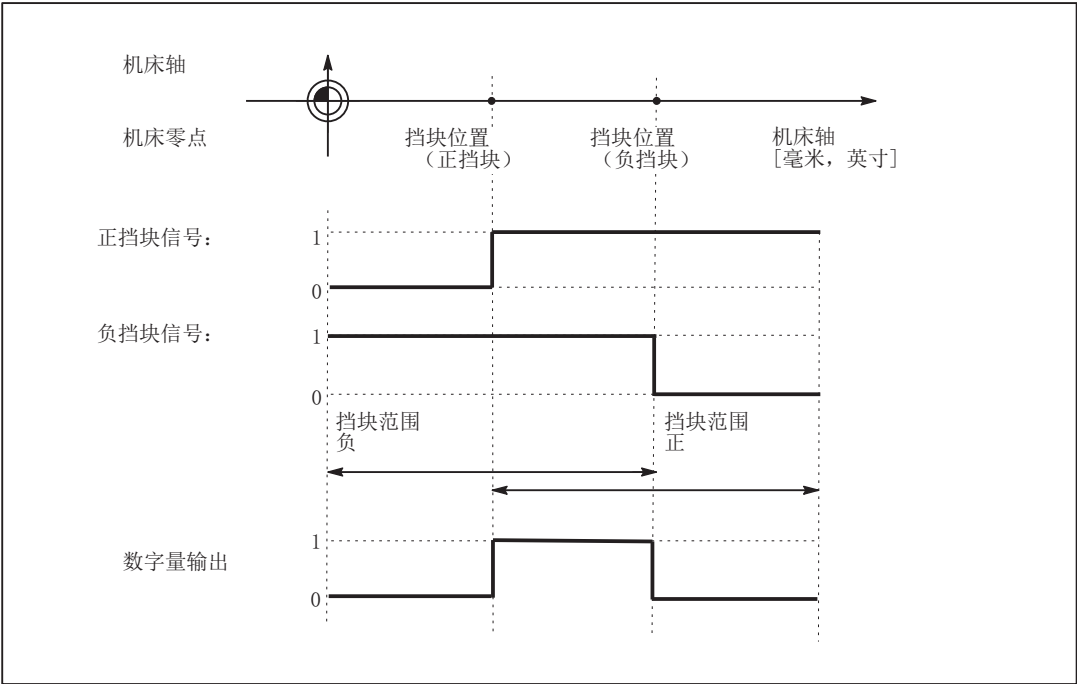


图 8-6 线性轴的行程开关信号（正挡块 < 负挡块）

模数回转轴

对于模数回转轴，缺省情形下信号特性取决于挡块宽度：

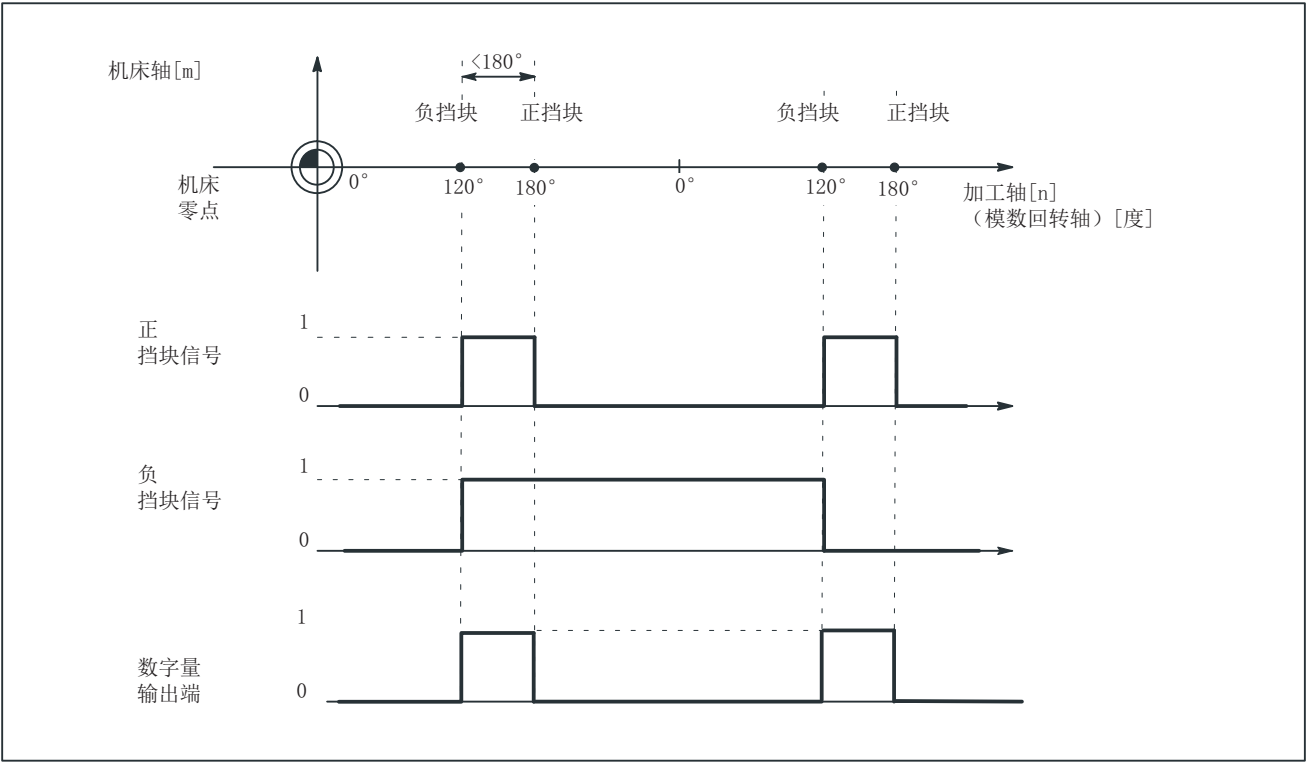


图 8-7 模数回转轴的软件挡块（正挡块 - 负挡块 < 180 度）

8.2 挡块信号和挡块位置

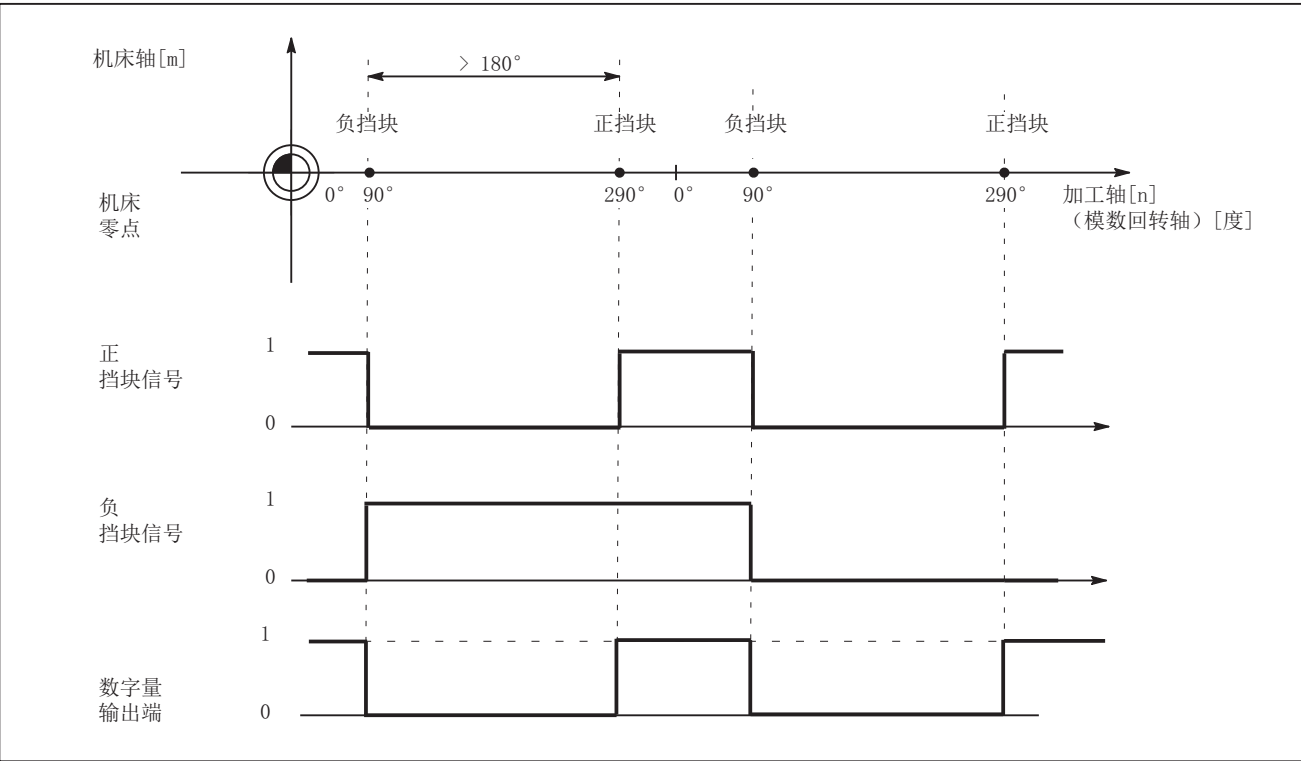


图 8-8 模数回转轴的软件挡块（正挡块 - 负挡块 > 180 度）

抑制信号取反

可通过以下设置抑制“正挡块 - 负挡块 > 180 度”时
正挡块信号特性的反转：

MD10485 SW_CAM_MODE 位 1=1

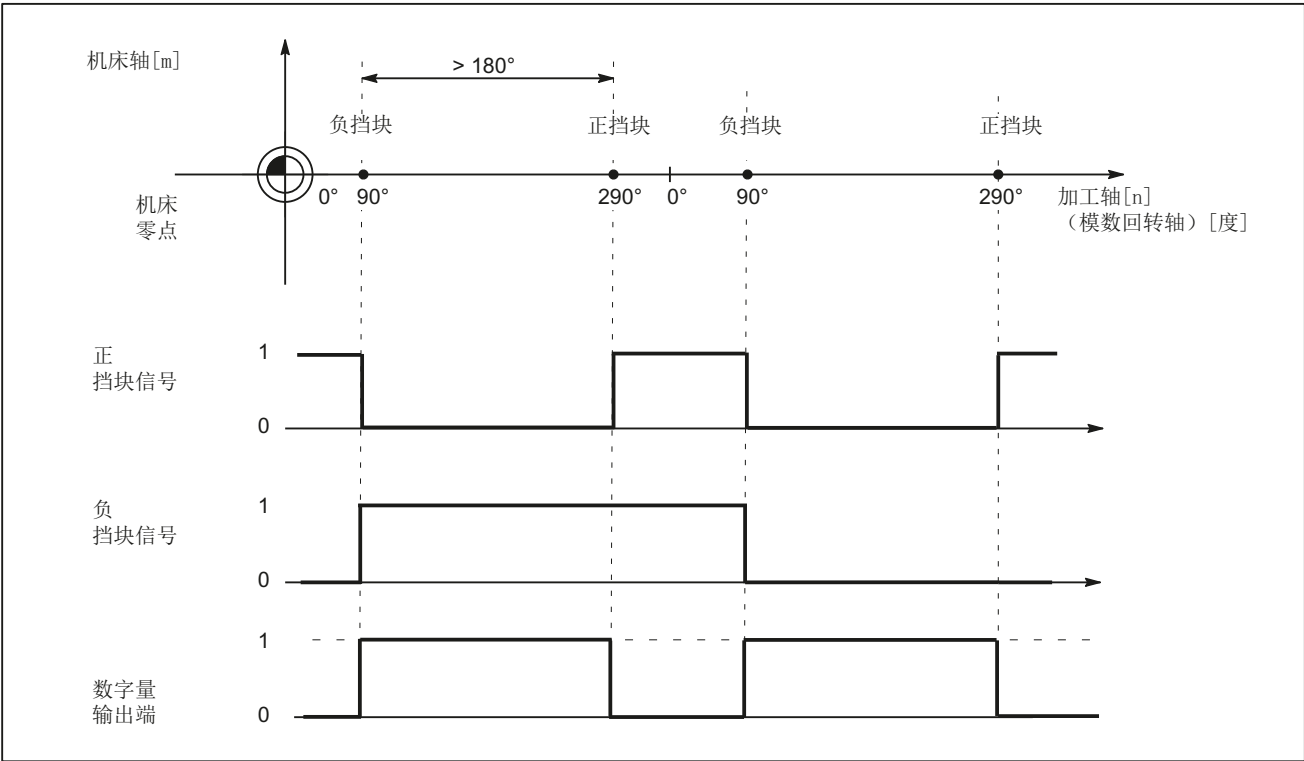


图 8-9 模数回转轴的软件挡块（正挡块 - 负挡块 > 180 度）；正挡块信号特性的反转被抑制

8.2.3 挡块位置

设置挡块位置

正/负挡块位置通过以下通用设定数据定义：

• SD41500 SW_CAM_MINUS_POS_TAB_1[n]	负挡块 1 - 8 的位置
• SD41501 SW_CAM_PLUS_POS_TAB_1[n]	正挡块 1 - 8 的位置
• SD41502 SW_CAM_MINUS_POS_TAB_2[n]	负挡块 9 - 16 的位置
• SD41503 SW_CAM_PLUS_POS_TAB_2[n]	正挡块 9 - 16 的位置
• SD41504 SW_CAM_MINUS_POS_TAB_3[n]	负挡块 17 - 24 的位置
• SD41505 SW_CAM_PLUS_POS_TAB_3[n]	正挡块 17 - 24 的位置
• SD41506 SW_CAM_MINUS_POS_TAB_4[n]	负挡块 25 - 32 的位置
• SD41507 SW_CAM_PLUS_POS_TAB_4[n]	正挡块 25 - 32 的位置

8.2 挡块信号和挡块位置

说明

将每八个挡块对归为一组后，可方便地为各组设定不同的访问保护级（例如：用于修改 WCS 挡块位置和 MCS 挡块位置）。这些位置在机床坐标系中输入。此时系统不会检查挡块位置是否超出最大运行范围。

公制/英制单位制

采用以下设置时：
MD10260 CONVERT_SCALING_SYSTEM = 1
挡块位置不再基于定义的基本单位制，而是基于以下机床数据中设置的单位制：
MD10270 POS_TAB_SCALING_SYSTEM（位置表的单位制）

值	含义
0	公制
1	英制

采用以上设置时，设定数据 SD41500 ... SD41507 中的位置数据也采用 MD10270 中定义的单位制。
G70/G71 或 G700/G710 进行的切换此时无效。

采集挡块位置

系统会比较轴实际值与挡块位置，相应地设置挡块信号。

读写挡块位置

可通过 HMI、PLC 和零件程序对设定数据进行读写访问。
通过零件程序进行的访问与加工不同步。
此时只能通过编写程序段预处理停止（STOPRE 指令）来实现同步。
在 PLC 用户程序中可通过 FB2 和 FB3 读写挡块位置。

将挡块指定给轴

通过以下通用机床数据将一个挡块对指定给机床轴：

MD10450 SW_CAM_ASSIGN_TAB[n] (软件挡块所属的机床轴)

说明

该机床数据的修改在下一次 NCK 启动后生效。
未指定给轴的挡块对将不生效。
一个挡块对总是只能指定给一根机床轴。
一根机床轴则可以有多多个挡块对。

8.2.4 提前时间/延迟时间 (动态挡块)

功能

为了补偿延迟时间, 可为每个正挡块/负挡块指定两个附加的提前时间或延迟时间, 提前或延迟信号的输出。
这两个提前时间或延迟时间需要分别在一个机床数据和一个设定数据中输入。

说明

输入负的时间值会使挡块信号的输出延迟。

在机床数据中输入提前时间或延迟时间

第一个提前时间或延迟时间在以下通用机床数据中输入:

- MD10460 SW_CAM_MINUS_LEAD_TIME[n] (负挡块信号输出的提前时间或延迟时间)
- MD10461 SW_CAM_PLUS_LEAD_TIME[n] (正挡块信号输出的提前时间或延迟时间)

此机床数据中例如可输入:

- 实际值测定和挡块信号输出之间的恒定内部延迟时间 (例如通过示波器测定)
- 恒定外部延迟时间

8.3 挡块信号输出

在设定数据中输入提前时间或延迟时间

第二个提前时间或延迟时间在以下通用设定数据中输入:

- SD41520 SW_CAM_MINUS_TIME_TAB_1[n] (负挡块 1 – 8 上的提前时间或延迟时间)
- SD41521 SW_CAM_PLUS_TIME_TAB_1[n] (正挡块 1 – 8 上的提前时间或延迟时间)
- SD41522 SW_CAM_MINUS_TIME_TAB_2[n] (负挡块 9 – 16 上的提前时间或延迟时间)
- SD41523 SW_CAM_PLUS_TIME_TAB_2[n] (正挡块 9 – 16 上的提前时间或延迟时间)
- SD41524 SW_CAM_MINUS_TIME_TAB_3[n] (负挡块 17 – 24 上的提前时间或延迟时间)
- SD41525 SW_CAM_PLUS_TIME_TAB_3[n] (正挡块 17 – 24 上的提前时间或延迟时间)
- SD41526 SW_CAM_MINUS_TIME_TAB_4[n] (负挡块 25 – 32 上的提前时间或延迟时间)
- SD41527 SW_CAM_PLUS_TIME_TAB_4[n] (正挡块 25 – 32 上的提前时间或延迟时间)

比如: 该设定数据中可输入加工期间可能发生变化的延迟时间。

8.3 挡块信号输出

8.3.1 激活

挡块的状态 (挡块信号) 可输出至 PLC, 也可另外输出至 NCK I/O。

激活挡块信号输出

通过以下 NC/PLC 接口信号激活一根轴的挡块信号输出:

DB31, ... DBX2.0 (激活挡块)

输出给 PLC 的反馈

一根轴的所有挡块成功激活后, NC 会通过以下 NC/PLC 接口信号反馈 PLC:

DB31, ... DBX62.0 (挡块生效)

说明

激活信号可以通过 PLC 用户程序和其他条件关联在一起 (例如轴回参考点, 复位生效)。

8.3.2 挡块信号输出到 PLC

输出到 PLC

对于所有激活了软件挡块的机床轴, 挡块信号的状态均会输出至 PLC。

输出按插补周期进行, 并由 PLC 异步接收。

负挡块信号

负挡块信号的状态由以下 NC/PLC 接口信号输出:

DB10 DBX110.0 至 113.7 (负挡块 1 至 32 的信号)

正挡块信号

正挡块信号的状态由以下 NC/PLC 接口信号输出:

DB10 DBX114.0 至 117.7 (正挡块 1 至 32 信号)

说明

若未选择测量系统或 NC/PLC 接口信号 DB31, ... DBX2.0 (激活挡块) 置 0, 那么下列 NC/PLC 接口信号同样会置 0:

- DB10 DBX110.0 - -113.7 (负挡块 1-32 的信号)
 - DB10 DBX114.0 - -117.7 (正挡块 1-32 的信号)
 - DB31, ... DBX62.0 (挡块生效)
-

8.3 挡块信号输出

8.3.3 挡块信号以位置控制周期输出到 NCK I/O

按位置控制周期输出信号

当通过机床数据 MD10470 至 MD10473 为挡块指定了一个硬件字节后（参见“硬件指定”部分），该挡块的信号按位置控制周期输出。

NCK 数字量输出可以为 NCU 上的 4 个板载输出以及最多 32 个外部 NCK 输出（参见“A4: SINUMERIK 840D sl 的 NCK 数字量/模拟量输入/输出 (页 31)”章节）。

硬件指定

每 8 个挡块对作为一组，通过以下通用机床数据指定给使用的硬件字节：

- MD10470 SW_CAM_ASSIGN_FASTOUT_1 硬件指定，用于将挡块 1 - 8 的信号输出至 NCK I/O
- MD10471 SW_CAM_ASSIGN_FASTOUT_2 硬件指定，用于将挡块 9 - 16 的信号输出至 NCK I/O
- MD10472 SW_CAM_ASSIGN_FASTOUT_3 硬件指定，用于将挡块 17 - 24 的信号输出至 NCK I/O
- MD10473 SW_CAM_ASSIGN_FASTOUT_4 硬件指定，用于将挡块 25 - 32 的信号输出至 NCK I/O

说明

在每个机床数据中可定义一个硬件字节用于输出 8 个负挡块信号，以及一个硬件字节用于输出 8 个正挡块信号。

此外可通过这两个机床数据对挡块信号输出进行取反。

若未指定第 2 个字节（=“0”），则 8 个正/负挡块信号经过第 1 硬件字节的第 1 取反位的取反后输出。

在零件程序中查询状态

可在零件程序中通过主处理变量 \$A_OUT[n] 读取硬件输出的状态（n = 输出位的编号）。

切换精度

挡块信号按位置控制周期输出到 NCK I/O 或板载输出。

切换精度受位置控制器周期和速度的限制。

此时以下关系适用：

$$\Delta \text{ 位置} = V_{\text{当前}} * \text{位置控制周期}$$

其中： $\Delta \text{ 位置}$ = 切换精度（受位置控制周期限制）

$V_{\text{当前}}$ = 当前轴速度

示例：

$$V_{\text{当前}} = 20 \text{ m/min}, \text{ 位置控制周期} = 4 \text{ ms} \quad \Delta \text{ 位置} = 1.33 \text{ mm}$$

$$V_{\text{当前}} = 2000 \text{ rpm}, \text{ 位置控制周期} = 2 \text{ ms} \quad \Delta \text{ 位置} = 24 \text{ 度}$$

8.3.4 挡块信号定时输出

挡块信号定时输出

挡块信号在定时器的控制下定时输出而不是按某周期输出，能够显著提升精度。

可通过以下机床数据为 4 个挡块对选择定时输出到 NCU 的 4 个板载端子：

MD10480 SW_CAM_TIMER_FASTOUT_MASK（挡块信号通过定时器中断事件输出给 NCU）

此时负挡块信号和正挡块信号会经过逻辑运算后作为一个信号输出。

信号计算

首先须定义两个信号的逻辑运算方式。这通过以下机床数据的位 1 进行：

MD10485 SW_CAM_MODE（软件挡块特性）

8.3 挡块信号输出

位	值	信号计算
1	0	正挡块 - 负挡块 ≥ 180 度时 反转正挡块信号的特性
	1	正挡块 - 负挡块 ≥ 180 度时 不反转正挡块信号的特性

说明

此功能不受 MD10470 ... MD10473 中的硬件指定影响。
板载字节不允许多次使用。

限制

对于挡块位置相对的位置：
每个插补周期只进行一次定时输出。
若一个插补周期中有多个挡块对的信号切换，那么输出时的优先级为：
编号（1...32）最低的挡块对决定所有信号的输出时间，即其他挡块对的信号在同一时间输出。

PLC 接口

在 PLC 接口上可查看 NCK 板载输出的映射以及正负挡块的状态。
但是这些信号在定时输出时不相关或不够精确，参见下面两段中的说明。正负挡块的信号按插补周期同步（一次性）生成，且一同传输至 PLC。
因此短于一个插补周期的脉冲在 PLC 中不可见。
板载输出通过中断事件异步于插补周期设置和复位。板载输出的状态同步于 PLC 接口的更新时间测定，并传输至 PLC。
PLC 接口进行更新时，根据当前状态，短于一个插补周期的脉冲可能在 PLC 中不可见，也可能延长一个或多个插补周期后才可见。

其他设置

若需使此处描述的特性生效，必须将以下位置“0”：
MD10485 SW_CAM_MODE 位 0 = 0

8.3.5 挡块信号单独定时输出

挡块信号单独定时输出

挡块信号由定时器控制时, 每个中断事件可输出一个脉冲沿, 而不受插补周期的影响。

挡块信号的相互影响通过以下因素消除:

- 每个插补周期输出一次
- 输出时间点由优先级最高 (编号最低) 的挡块对确定。

输出对象为 4 个板载输出时, 每个插补周期最多可输出 8 个由定时器控制的挡块信号, 用于设置/复位板载输出。缺省设置下这些挡块信号也一同输出到 PLC 接口上供计算。但这些信号对于定时输出无效或者不够准确。

信号计算

首先须定义两个信号的逻辑运算方式。这通过以下机床数据的位 1 进行:

MD10485 SW_CAM_MODE (软件挡块特性)

位	值	信号计算
1	0	正挡块 - 负挡块 ≥ 180 度时 反转正挡块信号的特性
	1	正挡块 - 负挡块 ≥ 180 度时 不反转正挡块信号的特性

设置

挡块对通过以下机床数据指定给板载输出:

MD10480 SW_CAM_TIMER_FASTOUT_MASK (挡块信号通过定时器中断事件输出给 NCU)

此外须显性选择此计算方式:

MD10485 SW_CAM_MODE 位 0 = 1

说明

此功能不受 MD10470 ... MD10473 中的硬件指定影响。

板载字节不允许多次使用。

8.4 “行程-时间”挡块

8.4 “行程-时间”挡块

“行程-时间”挡块

“行程-时间”挡块是一个软件挡块对, 可在指定的轴输出持续指定时间的脉冲。

解决方案

位置通过一个软件挡块对定义。

脉冲持续时间通过正挡块的提前时间/延迟时间定义。

通过机床数据可将“负挡块位置 = 正挡块位置”的挡块对定义为“行程-时间”挡块。

“行程-时间”挡块的属性

- 脉冲持续时间不受轴速度和运行方向反转的影响。
- 脉冲持续时间不受轴位置变化的影响（缺省）。
- 只有在越过挡块位置后该挡块激活（激活脉冲沿）。
轴位置的偏移（例如预设）不会导致挡块激活。
- 负挡块的提前时间/延迟时间仍保持生效, 使脉冲的输出时间提前/延迟。
- 激活脉冲沿和脉冲持续时间不受运行方向的影响。
- 挡块生效时, 再次反向越过挡块位置时不会撤销挡块。
- 挡块时间（脉冲宽度）不会因为重新越过挡块位置而中断, 且不会重新开始。
在模数轴上尤其需要注意此特性: 脉冲宽度大于过模数区的时间时, 将不会在每次旋转时进行切换。

设置

为了确定一个“行程-时间”挡块, 必须进行以下设置:

- **位置**

定义一个“负挡块位置 = 正挡块位置”的挡块对。

通过下列设定数据进行定义:

SD41500 SW_CAM_MINUS_POS_TAB_1

...

SD41507 SW_CAM_PLUS_POS_TAB_4。

- **脉冲持续时间**

脉冲持续时间由下列数据中各个挡块对条目相加得出:

MD10461 SW_CAM_PLUS_LEAD_TIME[n]

SD41521 SW_CAM_PLUS_TIME_TAB_1[n]...

SD41527 SW_CAM_PLUS_TIME_TAB_4[n]

- **偏移**

“行程-时间”挡块的时间偏移由下列数据中挡块对各个条目相加得出:

MD10460 SW_CAM_MINUS_LEAD_TIME[n]

SD41520 SW_CAM_MINUS_TIME_TAB_1[n]...

SD41526 SW_CAM_MINUS_TIME_TAB_4[n]

- **模式**

MD10485 SW_CAM_MODE

在此机床数据中必须设置位 2 = 1, 从而将所有正负挡块位置相同的挡块对作为“行程-时间”挡块处理。

8.5 边界条件

“软件挡块, 行程开关信号”功能的可用性

此功能为选件 (“行程开关信号/挡块机构”), 授权后方可和硬件组合使用。

8.6 数据表

8.6 数据表

8.6.1 机床数据

8.6.1.1 通用机床数据

编号	名称: \$MN_	描述
10260	CONVERT_SCALING_SYSTEM	基本单位制切换有效
10270	POS_TAB_SCALING_SYSTEM	位置表的单位制
10450	SW_CAM_ASSIGN_TAB[n]	软件挡块所属的机床轴
10460	SW_CAM_MINUS_LEAD_TIME[n]	负挡块 1 – 16 上的提前时间或延迟时间
10461	SW_CAM_PLUS_LEAD_TIME[n]	正挡块 1 – 16 上的提前时间或延迟时间
10470	SW_CAM_ASSIGN_FASTOUT_1	硬件指定, 用于将挡块 1 - 8 输出至 NCK I/O
10471	SW_CAM_ASSIGN_FASTOUT_2	硬件指定, 用于将挡块 9-16 输出至 NCK I/O
10472	SW_CAM_ASSIGN_FASTOUT_3	硬件指定, 用于将挡块 17-24 输出至 NCK I/O
10473	SW_CAM_ASSIGN_FASTOUT_4	硬件指定, 用于将挡块 25-32 输出至 NCK I/O
10480	SW_CAM_TIMER_FASTOUT_MASK	挡块信号通过定时器中断事件输出给 NCU
10485	SW_CAM_MODE	软件挡块的特性

8.6.2 设定数据

8.6.2.1 通用设定数据

编号	名称: \$SN_	描述
41500	SW_CAM_MINUS_POS_TAB_1[n]	负挡块 1 - 8 的位置
41501	SW_CAM_PLUS_POS_TAB_1[n]	正挡块 1 - 8 的位置
41502	SW_CAM_MINUS_POS_TAB_2[n]	负挡块 9 - 16 的位置
41503	SW_CAM_PLUS_POS_TAB_2[n]	正挡块 9 - 16 的位置
41504	SW_CAM_MINUS_POS_TAB_3[n]	负挡块 17 - 24 的位置
41505	SW_CAM_PLUS_POS_TAB_3[n]	正挡块 17 - 24 的位置
41506	SW_CAM_MINUS_POS_TAB_4[n]	负挡块 25 - 32 的位置

编号	名称: \$SN_	描述
41507	SW_CAM_PLUS_POS_TAB_4[n]	正挡块 25 - 32 的位置
41520	SW_CAM_MINUS_TIME_TAB_1[n]	负挡块 1 - 8 上的提前时间或延迟时间
41521	SW_CAM_PLUS_TIME_TAB_1[n]	正挡块 1 - 8 上的提前时间或延迟时间
41522	SW_CAM_MINUS_TIME_TAB_2[n]	负挡块 9 - 16 上的提前时间或延迟时间
41523	SW_CAM_PLUS_TIME_TAB_2[n]	正挡块 9 - 16 上的提前时间或延迟时间
41524	SW_CAM_MINUS_TIME_TAB_3[n]	负挡块 17 - 24 上的提前时间或延迟时间
41525	SW_CAM_PLUS_TIME_TAB_3[n]	正挡块 17 - 24 上的提前时间或延迟时间
41526	SW_CAM_MINUS_TIME_TAB_4[n]	负挡块 25 - 32 上的提前时间或延迟时间
41527	SW_CAM_PLUS_TIME_TAB_4[n]	正挡块 25 - 32 上的提前时间或延迟时间

8.6.3 信号

8.6.3.1 到进给轴/主轴的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
激活挡块	DB31, ...DBX2.0	-

8.6.3.2 来自进给轴/主轴的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
挡块生效	DB31, ...DBX62.0	-

N4: 冲裁和步冲 - 仅适用于 840D sl

9.1 简要描述

子功能

专用于冲裁和步冲的功能包含:

- 冲程控制
- 自动位移划分
- 可旋转的主刀具和子刀具
- 夹钳保护

通过语言指令激活或取消功能。

9.2 冲程控制

9.2.1 概述

功能

冲程控制用于真正的工件加工。达到位置后通过 **NC** 的输出信号激活冲头。冲裁单元通过 **NC** 上的输入信号应答冲杆运行。在该时间段内不允许任何轴运行。结束冲裁后重新定位。

快速信号

为了使 **NC** 和冲裁单元间可以直接通讯, 可以使用“快速信号”。该信号和冲头一起作用可以冲出更多数量的孔(每分钟), 因为定位过程被视为加工“时滞”。

PLC 信号

不受时间限制的信号, 例如: 监控或者使能信号通过 **PLC** 接口信号实现。

9.2 冲程控制

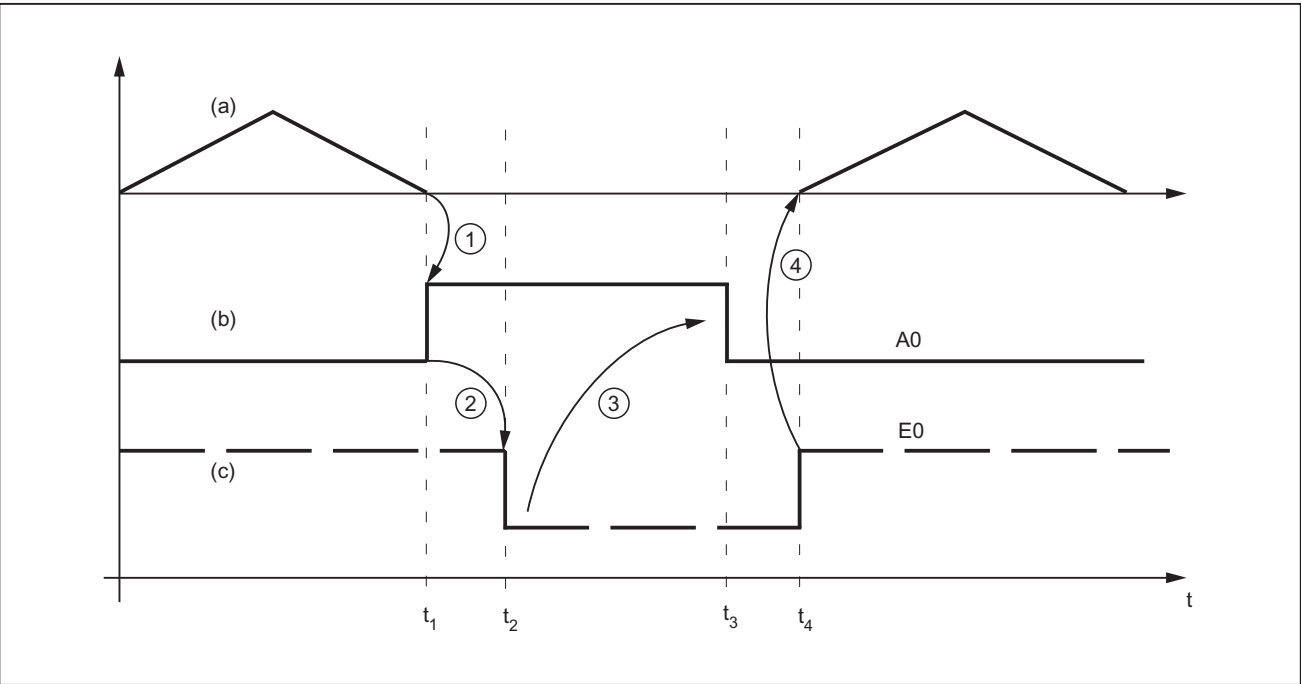
9.2.2 快速信号

功能

NC 和冲裁单元间的同步可使用高速信号。这些信号作用在于一方面，只有当板材停止时才通过高速输出释放冲程。另一方面，当冲模和板材接触时通过高速输入保持板材处于停止状态。

控制系统的高速输入/输出可用于控制冲裁单元。

下图粗略地展示了信号时序。



- (a) 机床轴“时间-速度”图
- (b) 信号“冲程释放”
- (c) 信号“冲程有效”

图 9-1 信号图

说明
由于断线监控，信号“冲程有效”为高位生效。

冲裁和步冲的时序通过两个信号 A_0 和 E_0 控制：

A_0	由 NCK 置位，相当于信号“冲程释放”。
E_0	指出冲裁单元状态，相当于信号“冲程有效”。

信号在 t_1 到 t_4 内的各个时间段内产生的影响如下：

t_1	在时间点 t_1 上，工件（板材）相对于冲模的运动结束。 根据定义的冲程释放条件（参见“冲程释放条件 (页 595)” 章节），释放冲程的高速输出 A_0 置位 ①。
t_2	在时间点 t_2 上，冲裁单元通过高速输入 E_0 报告冲模运动。 该运动通过信号 A_0 触发 ②。 出于安全原因， E_0 高位有效（在断线时，“冲程有效”始终置位，而不进行任何轴运行）。 当冲模离开板材后(t_4)，信号“冲程有效”才再次复位。
t_3	NC 在时间点 t_3 上复位信号“冲程释放”，作为对信号“冲程有效”的应答 ③。从该时间开始，NC 处于等待状态。只有在信号“冲程有效”复位后，才采取允许后续轴运动。 只有在复位了信号 A_0 后，才可以释放下一个冲程。
t_4	在时间点 t_4 上，冲裁结束，即：冲模再次离开板材。 当信号 E_0 的脉冲沿转换时，NC 启动轴运行，作为应答 ④。 脉冲沿转换时的 NC 应答④在“冲裁后的轴启动”中描述。

说明

冲程持续时间由时间 $\Delta t_h = t_4 - t_1$ 决定。
其中还要加上 t_4 上 E_0 脉冲沿转换和轴开始运动之间的响应时间。

9.2.3 冲程释放条件

释放冲程

只有在板材静止后，才能释放冲程。冲程不能提前释放，从而可以确保在冲裁时冲模和加工平面中的板材不会发生相对运动。

9.2 冲程控制

下图展示了不同的冲程释放条件。

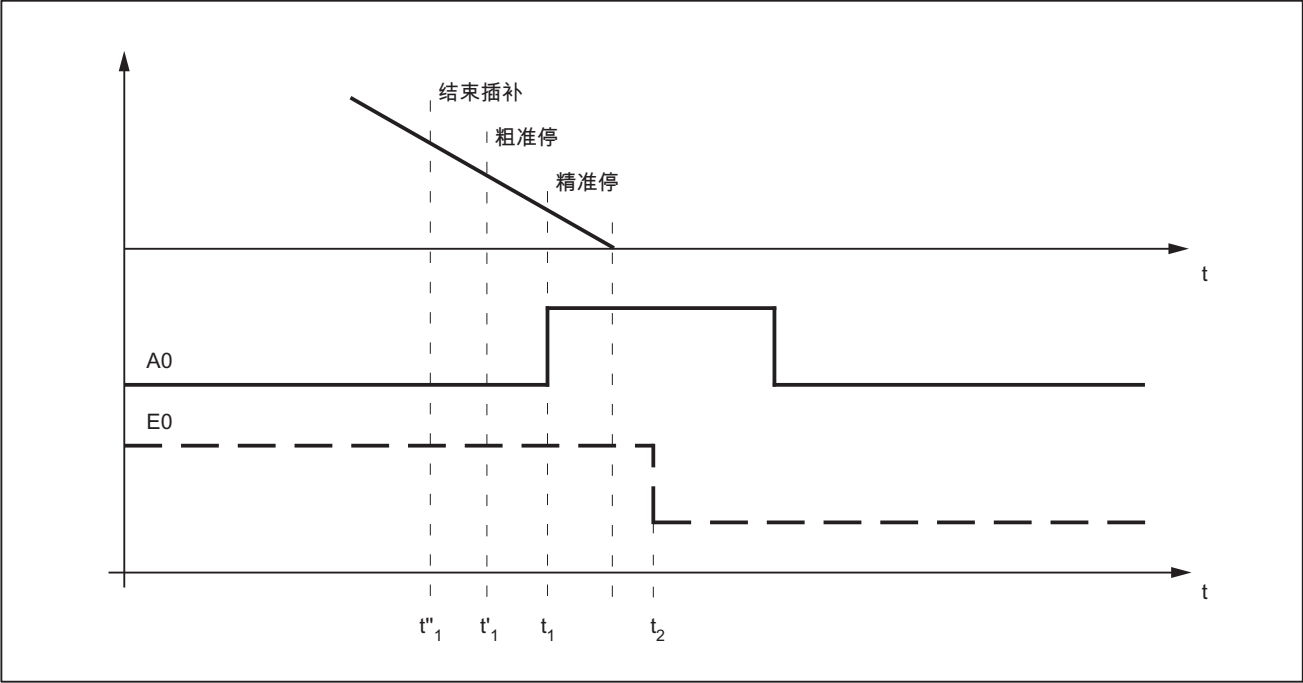


图 9-2 信号图：冲程释放条件

t_1 和 t_2 之间的时间段是冲裁单元对 A_0 信号置位的响应时间。该时间不能改变，但可用作预处理时间，以降低时滞。

图中展示了缺省设置：达到“精准停窗口”后输出置位（G601; G 功能组 12 的缺省设置）。冲程释放的时间 t_1'' 和 t_1' 分别针对使用 G602 和 G603 的情况（参见下表）。

编程	结果	描述
G603	停止插补	插补达到程序段末尾。在这种情况下轴继续运行，直到惯性停转。也就是说，在轴停止前会明确输出信号（参见 t_1'' ）。
G602	达到粗定位窗口	当轴进入粗定位窗口时，输出信号。如果选择该条件，则可以通过定位窗口的大小改变冲程释放时间点（参见 t_1' ）。
G601	达到精定位窗口	选择该条件时，如果轴数据的设置合适就可以确保板材在冲裁时已静止。但也会导致最大时滞（参见 t_1 ）。

说明

包含 G601、G602 和 G603 的 G 组（G 组 12）的缺省设置通过以下机床数据定义：

MD20150 \$MC_GCODE_RESET_VALUES[11]

缺省设置是 G601。

G603

取决于不同的速度和机床动态响应，插补结束后经过大约 3 到 5 个插补周期，轴才停止。

MD26018 \$MC_NIBBLE_PRE_START_TIME

通过以上机床数据可以延长并优化插补结束与高速输出“冲程释放”置位之间的时间。

除 MD26018 外还有以下设定数据可供使用：

SD42402 \$SC_NIBPUNCH_PRE_START_TIME

可以从零件程序修改 SD42402，因此它可以根据零件程序处理的进程和冲裁过程相匹配。

以下条件适用于延迟时间：

MD26018 = 0 → SD42402 生效

MD26018 ≠ 0 → MD26018 生效

如果功能“带暂停时间 PDELAYON 的冲裁”生效，则功能中设置的暂停时间生效。而 MD26018 和 SD42402 失效。

9.2 冲程控制

9.2.4 冲裁后的轴启动

输入信号“冲程开”

通过输入信号“冲程开”控制轴运行的开始。

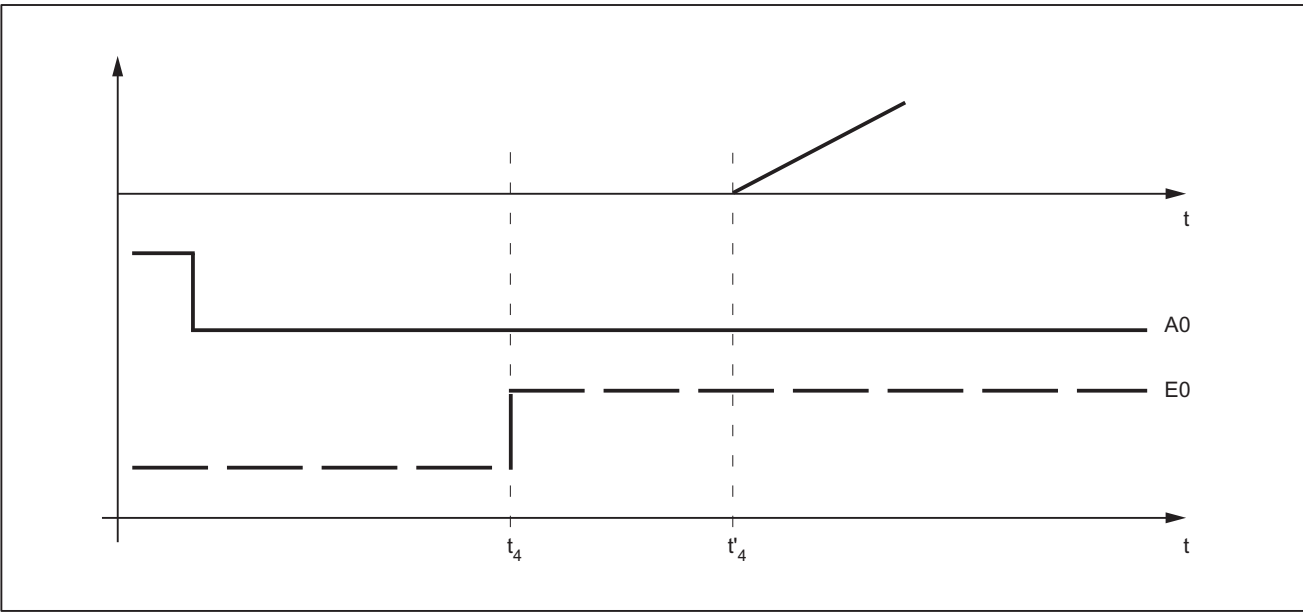


图 9-3 信号变化：冲裁后的轴启动

从而产生 t_4 和 t'_4 之间的时间段，它作为由开关时间影响的反应时间。该时间由插补扫描时间和编程的冲裁/步冲类型决定。

PON/SON

由 PON/SON 控制冲裁单元时，计算出的延迟时间最大为：

$$|t'_4 - t_4| = 3 \times \text{插补脉冲}$$

PONS/SONS

由 PONS/SONS 控制冲头时，由以下时间决定延迟时间：

$$|t'_4 - t_4| \leq 3 \times \text{位置控制器脉冲}$$

前提条件：冲程持续时间($t_4 - t_2$) > 4 个插补脉冲

9.2.5 仲裁和步冲专用的 PLC 信号

功能

除了直接控制冲程的信号外，还存在通道专用的 PLC 接口信号。这些信号一方面用于调节仲裁过程，一方面用于显示状态。

信号

信号	结果
DB21, ... DBX3.0 (无冲程使能)	禁止 NC 释放任何冲程。 NC 等待冲程使能，之后才继续零件程序。
DB21, ... DBX3.2 (封锁冲程)	不释放冲程，但零件程序可以试运行。 在行程划分生效时，轴以“Stop and Go”方式运行。
DB21, ... DBX3.4 (延时冲程)	激活冲程延时，正如通过 PDELAYON 实现。
DB21, ... DBX3.1 (手动冲程释放)	该功能使操作员可以释放冲程（通过 PLC 控制），即使无任何零件程序。 手动冲程释放通过以下信号应答： DB21, .. DBX38.1 (手动冲程释放应答)

9.2 冲程控制

9.2.6 冲裁和步冲对标准 PLC 信号的特定反应

DB21, ... DBX12.3（进给停止）

在接口信号：
DB21, ... DBX12.3（进给停止）置位时
NC 冲程控制作出如下响应：

在时间 t_1 前发现该信号：	抑制冲程释放。 只有在下一次启动或者取消“进给停止”后才释放 下一次冲程。 随后加工继续，如同没有中断一样。
在时间 t_1 后发现该信号：	仍执行完当前冲程。 接着，NC 处于 t_4 中的状态。 为支持该特性，信号“冲程有效”和“冲程释放” 的时间监控失效。

9.2.7 信号监控

振荡信号

由于冲裁气动系统的老化，在冲程结束后“冲程有效”信号可能因冲模超调而发生振荡。
此时会根据机床数据：
MD26020 \$MC_NIBBLE_SIGNAL_CHECK
生成报警（报警 22054“冲裁信号杂乱”）。

复位特性

NCK 复位时信号接口：
DB21, ... DBX38.0（冲程有效）
立即被清除，不等待高速输入的应答。
不能抑制当前生效的冲程。

9.3 激活和撤销

9.3.1 语言指令

冲裁和步冲功能的激活和取消通过可设定的语言指令来执行。在以前系统中经常使用的特殊 M 功能被取消。

文档:

编程手册之工作准备分册

组

语言指令划分为以下各组:

组 35	
通过以下的语言指令可以激活或取消冲裁和步冲专用功能:	
PON	= 冲裁开
SON	= 步冲开
PONS	= 冲裁开, 在位置控制器周期内释放冲程
SONS	= 步冲开, 在位置控制器周期内释放冲程
SPOF	= 冲裁/步冲关

组 36	
该组中包含具有准备作用和可确定具体冲裁特性的指令:	
PDELAYON	= 冲裁延时开
PDELAYOF	= 冲裁延时关
通常 PLC 需要对此类准备功能进行预处理, 所以必须在激活指令前编程。	

组 38	
该组中包含了用于切换到第二个冲裁接口的指令。其可用于例如第二个冲裁单元或用于剪切。通过机床数据定义第二个用于冲裁功能的 I/O 组。	
SPIF1	= 首个接口激活
SPIF2	= 第二个接口激活

9.3 激活和撤销

说明

在一个 G 代码组中只能激活一个功能（与不同的插补方式 G0, G1, G2, G3 等相类似，它们之间相互排斥）。

SPOF

取消冲裁和步冲

SPOF 功能结束所有的冲裁和步冲功能。在该状态下 NCK 既不对“冲程有效”信号作出响应，也不对冲裁/步冲专用 PLC 信号作出响应。

如果 SPOF 与一个位移指令一同编程在一个程序段中（以及所有后面的程序段，如果不用 SON 或 PON 激活冲裁/步冲），则机床不带冲裁释放运行编程位置。SPOF 取消选择 SON, SONS, PON 和 PONS，相当于复位状态。

编程示例：

程序代码	注释
:	
:	
N20 G90 X100 SON	; 激活步冲
N25 X50 SPOF	; 取消步冲
	; 定位，不释放冲程
:	
:	

SON

激活步冲

SON 打开步冲功能并取消 G 功能组 35 的其它功能（例如 PON）。

与冲裁相反，首个冲程在程序段的起点处就开始执行，即在机床开始运动前执行。

SON 模态有效，即一直有效，直到编程了 SPOF 或者 PON，或者程序结束。

当程序段中不包含标为冲裁轴或步冲轴的运行信息（通常为当前平面），冲程释放受到抑制。如果仍要释放冲程，则必须明确地为冲裁轴/步冲轴编程 0 行程。如果带有 SON 的首个程序段不带运行信息，则在该程序段中只执行唯一一个冲程，因为起点和终点重合。

编程示例：

程序代码	注释
:	
:	
N70 X50 SPOF	; 定位，不释放冲裁
N80 X100 SON	; 激活步冲，
	; 在运动（X=50）前和编写的
	; 运动（X=100）结束处释放冲程
:	
:	

SONS

激活步冲(在位置调节器周期中)

SONS 与 SON 性能相同。冲程在一个位置调节器周期中释放。以便可以达到冲程释放的最优时间并提高每分钟的冲裁率。

PON

激活冲裁

PON 激活冲裁功能并取消 SON。

PON 与 SON 一样模态有效。

与 SON 不同，在程序段终点才执行冲程，或自动划分行程时的每段行程终点才执行冲程。在不带运行信息的程序段中，生效方式与 SON 的一致。

编程示例：

程序代码	注释
:	
:	
N100 Y30 SPOF	; 定位，不释放冲裁
N110 Y100 PON	; 激活冲裁，在定位过程结束时
	; （Y=100）释放冲裁
:	
:	

PONS

激活冲裁(在位置调节器脉冲中)

PONS 与 PON 性能相同。说明参见 SONS。

9.3 激活和撤销

PDELAYON

冲裁延时

PDELAYON 是一种准备功能。这表示，通常 PDELAYON 要在 PON 之前编程。到达编程终点位置后，冲程延迟释放。

延迟时间在以下设定数据中设置，单位秒：

SD42400 \$SC_PUNCH_DWELLTIME

如果输入值不是插补周期的整数倍值，在系统会自动将它取为下一个整数倍值。

该功能模态有效。

PDELAYOF

取消冲裁延时

PDELAYOF 关闭冲裁延时，即继续正常冲裁模式。PDELAYON 和 PDELAYOF 构成了一个 G 代码组。

编程示例：

SPIF2 激活第二个冲裁接口，即，通过第二对高速 I/O 控制冲程（参见“通道专用机床数据 (页 635)”章节，机床数据 MD26004 和 MD26006）。

程序代码	注释
:	
:	
N170 PDELAYON X100 SPOF	; 定位，不释放冲裁， ; 激活冲裁延时
:	
:	
N180 X800 PON	; 激活冲裁。 到达编程终点位置后 ; 冲程延迟释放
:	
:	
N190 PDELAYOF X700	; 取消冲裁延时， ; 在编写的运动结束处 ; 释放一般冲裁
:	
:	
:	

SPIF1

激活第一个仲裁接口

SPIF1 激活第一个仲裁接口，即，通过第一对高速 I/O 控制冲程（参见“通道专用机床数据 (页 635)”章节，机床数据 MD26004 和 MD26006）。

复位或控制系统启动后，首个仲裁接口始终有效。如果只使用一个接口，则无须进行编程。

SPIF2

激活第二个仲裁接口

SPIF2 激活第二个仲裁接口，即，通过第二对高速 I/O 控制冲程（参见“通道专用机床数据 (页 635)”章节，机床数据 MD26004 和 MD26006）。

编程示例：

程序代码	注释
:	
:	
N170 SPIF1 X100 PON	; 程序段结束后，在第一个高速输出上 ; 释放冲程。 信号“冲程有效” ; 在第一个输入上被监控。
:	
:	
:	
:	
N180 X800 SPIF2	; 在第二个高速输出上 ; 进行第二次冲程释放。 信号“冲程有效” ; 在第二个输入上被监控。
:	
:	
:	
N190 SPIF1 X700	; 对所有后续冲程的控制 ; 通过第一个接口进行。
:	

9.3.2 功能扩展

可转换的接口

如果需要机床交替式使用第二个仲裁单元或者类似的装置，可以切换到第二个 I/O 对。

9.3 激活和撤销

可以通过以下机床数据定义第二个 I/O 对：

MD26004 \$MC_NIBBLE_PUNCH_OUTMASK

MD26006 \$MC_NIBBLE_PUNCH_INMASK

通过指令 SPIF1 或者 SPIF2 进行转换。

这两个接口可使用所有的仲裁/步冲功能。

示例：

冲程控制上的硬件分配

定义 CPU 上的高速字节（高速仲裁接口）：

MD26000 \$MC_PUNCHNIB_ASSIGN_FASTIN = 'H00030001' → 字节 1

MD26002 \$MC_PUNCHNIB_ASSIGN_FASTOUT = 'H00000001'

注释：

第一位和第二位经过取反。

高速输入位和高速输出位：

MD26004 \$MC_NIBBLE_PUNCH_OUTMASK[0]	= 1	第一个接口输出位 → 位 1 SPIF1
MD26004 \$MC_NIBBLE_PUNCH_OUTMASK[1]	= 2	第二个接口输出位 → 位 2 SPIF2
MD26006 \$MC_NIBBLE_PUNCH_INMASK[0]	= 1	第一个接口输入位 → 位 1 SPIF1
MD26006 \$MC_NIBBLE_PUNCH_INMASK[1]	= 2	第二个接口输入位 → 位 2 SPIF2

可自动激活的预释放时间

若冲程在到达轴插补窗口前就已经释放，则可以尽量缩短仲裁单元响应时间导致的时滞。参考点为插补终点。冲程释放可自动由 G603 激活并在到达插补终点后延迟一段时间。

冲程释放的延迟时间可以通过以下机床数据设置：

MD26018 \$MC_NIBBLE_PRE_START_TIME

示例：

当插补周期为 5 毫秒时，需要在插补结束后的 2 个周期内释放冲程：

⇒ MD26018 \$MC_NIBBLE_PRE_START_TIME = 0.01 [s]

也可以在设定数据中设置预释放时间：

SD42402 \$SC_NIBPUNCH_PRE_START_TIME

只有当设置了 MD26018 = 0，该值才生效。

输入信号监控

如果“冲程激活”信号例如由于冲程间的冲模超调而抖动，则除插补停止之外还会输出信息“杂乱仲裁信号”。

信息的输出取决于机床数据中的设置：

MD26020 \$MC_NIBBLE_SIGNAL_CHECK

MD26020 = 0	没有报警
MD26020 = 1	报警

两个冲程间的最短时间

通过下列设定数据可以设定两个连续冲程间的最短时间：

SD42404 \$SC_MINTIME_BETWEEN_STROKES

示例：

和空间距离无关，两个冲程释放间应至少间隔 1.3 秒。

⇒ SD42404 \$SC_MINTIME_BETWEEN_STROKES = 1.3 [s]

如果此外还编程了仲裁暂停时间 (PDELAYON)，则两个时间相加的结果为间隔时间。

只有在 SD42404 设定时间届满后才到达插补终点时，用 G603 设置的预释放时间才生效。

编程的时间立即有效。依据程序段缓冲器的大小，之前编程的的冲程可以该最短间隔执行。该功能会受以下编程（举例）的阻碍：

程序代码
N...
N100 STOPRE
N110 \$SC_MINTIME_BETWEEN_STROKES = 1.3

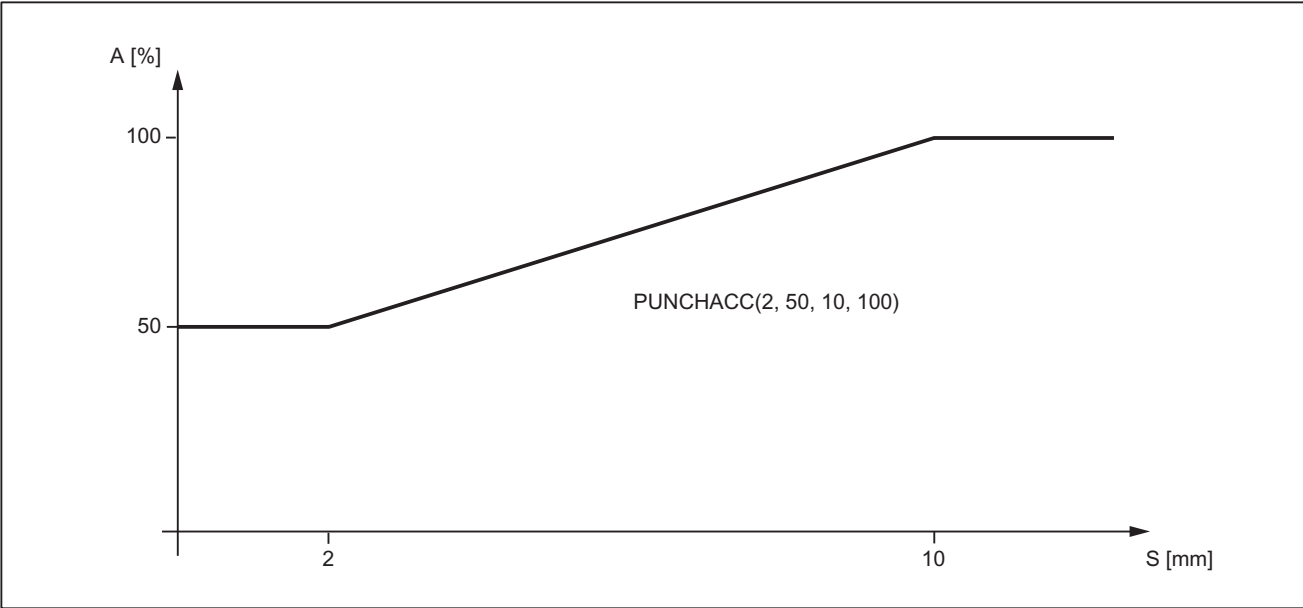
SD42404 = 0 时该功能无效。

9.3 激活和撤销

位移决定的加速度

通过语言指令 PUNCHACC (Smin, Amin, Smax, Amax) 可以定义加速度特征曲线。在此可以依据孔距定义不同的加速度。

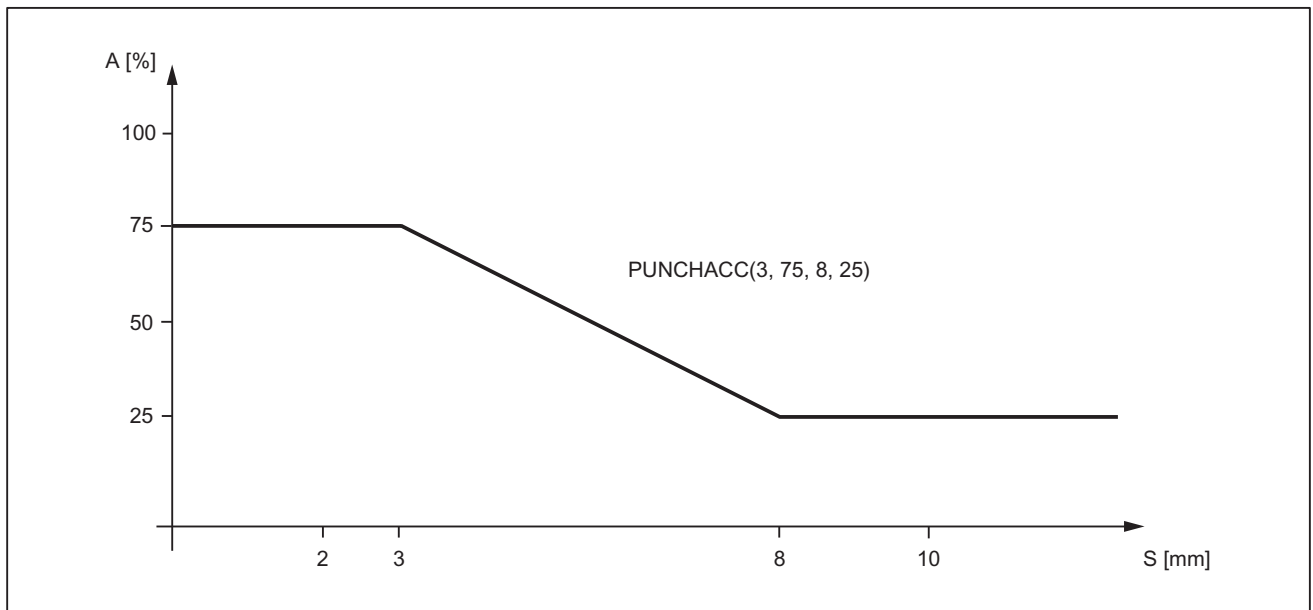
示例 1:



特性曲线确定以下加速度:

孔间距	加速度
< 2 mm	以最大加速度的 50 % 运行。
2 - 10 mm	该加速度与距离成正比提高到 100 %。
> 10 mm	以 100% 的加速度运行。

示例 2:



特性曲线确定以下加速度:

孔间距	加速度
< 3 mm	以最大加速度的 75 % 运行。
3 - 8 mm	加速度与间距成正比降低到 25%。
> 10 mm	以 25 % 的加速度运行。

如果之前通过 ACC 编程了递减的加速度, 则通过 PUNCHACC 定义的极限加速度以该递减加速度为参考。

取消功能通过:

$$S_{\min} = S_{\max} = 0$$

之前 ACC 编程的加速度继续有效。

程序段搜索

在搜索带步冲功能的程序段时可以设置, 是否执行或抑制程序段开头的冲程。

在以下机床数据中进行设置:

MD11450 \$MN_SEARCH_RUN_MODE

9.3 激活和撤销

位	值	含义
5	0	抑制程序段开头的冲程。
	1	执行程序段开头的冲程。

文档:

功能手册之基本功能分册; “BAG、通道、程序运行、复位特性 (K1)”, 章节 “程序段搜索”

9.3.3 与老系统的兼容性

M 功能的应用

利用宏指令技术, 仍可以和以前一样用特殊的 M 功能来编程, 而无需使用语言指令 (兼容性)。

其中, M 功能和语言指令的对应关系如下:

M20, M23	△	SPOF
M22	△	SON
M25	△	PON
M26	△	PDELAYON

说明

M 功能可以通过机床数据设定。
在为语言指令设定 M 功能时, 注意 M 功能的分组。

示例

```
DEFINE M20 AS SPOF           仲裁/步冲关闭
或者
DEFINE M20 AS SPOF M=20      带辅助功能输出的仲裁

DEFINE M20 AS SPOF PDELAYOF  仲裁/步冲关闭, 仲裁延时关闭
```

```
DEFINE M22 AS SON          激活步冲
或者
DEFINE M22 AS SON M=22     带辅助功能输出的步冲激活

DEFINE M25 AS PON          激活仲裁
或者
DEFINE M25 AS PON M=25     带辅助功能输出的仲裁激活

DEFINE M26 AS PDELAYON     仲裁延时
或者
DEFINE M26 AS PDELAYON M=26 仲裁和辅助功能输出
```

编程示例:

程序代码	注释
:	
:	
N100 X100 M20	; 定位, 不释放仲裁
N110 X120 M22	; 激活步冲, 在运行前后释放冲程
	; 冲程释放
:	
N120 X150 Y150 M25	; 激活仲裁, 在运动结束时
	; 释放冲程
:	
:	

9.4 自动划分行程

9.4.1 简介

功能

在自动划分编程的行程时，有以下几种方式：

- 按照语言指令 SPP 写入的最大子行程进行划分
- 按照语言指令 SPN 写入的子行程数量进行划分

两种功能都生成独立的程序段。

在老系统中对应的功能为：

- E<数量>对应 SPP<数量>
- H<数量>对应 SPN<数量 1>

由于地址 E 和 H 现在用于表示辅助功能，因此为避免冲突使用语言指令 SPP 和 SPN。因此不和以前的系统兼容。两个语言指令(SPP 和 SPN)都可通过机床数据加以定义。

说明

根据单位制的设置，SPP 编程的单位为毫米或英寸（和进给轴的一致）。

在直线插补和圆弧插补中，“自动划分行程”功能采用等行程划分法。

自动划分行程(SPP/SPN)生效而程序中断时，只能在程序段开头处回到轮廓。第一个冲程在子程序段结束时执行。

只有定义了几何轴时，SPP 和 SPN 才可以被激活。

SPP

自动划分行程 SPP 按照预设的子行程即以相同的行程划分编程的总行程。

需要满足下列条件：

- 只有当激活了 SON 或者 PON 时，行程划分才生效。
(例外： MD26014 \$MC_PUNCH_PATH_SPLITTING = 1)
- SPP 模态有效，即：编程的子行程持续有效，直至重新编程；SPN 可以按照非模态方式抑制该功能。
- 必要时控制系统会圆整子行程，从而可以使编程的总行程能够化分为整数个行程。

- 子行程单位为毫米/冲程或者英寸/冲程（根据轴设定）。
- 如果编程的 SPP 值大于总行程，则轴会直接移动到编程的终点位置，而不划分行程。
- SPP = 0、复位或者程序结束将删除编程的 SPP 值。取消冲裁/补充不会删除编程的 SPP 值。

SPN

自动划分行程 SPN 按照编程的子行程数划分总行程。

需要满足下列条件：

- 只有当激活了 SON 或者 PON 时，行程划分才生效。
(例外: MD26014 \$MC_PUNCH_PATH_SPLITTING = 1)
- SPN 非模态有效。
- 之前编程的 SPP 值对于该程序段失效，但对后续的程序段再次生效。

边界条件

- 行程划分在直线插补和圆弧插补中生效。
插补方式保持不变，即：圆弧插补时沿圆弧离开。
- 如果在一个程序段中既编程了 SPN（冲程数量）又编程了 SPP（冲程行程），则在当前程序段中冲程数量生效，而在后续的程序段中冲程行程生效。
- 只有当激活了冲裁或者步冲时，行程划分才生效。
(特例: MD26014 \$MC_PUNCH_PATH_SPLITTING = 1)
- 在程序段开始或者结束时才输出编程的辅助功能。
- 在不带运行信息的程序段中，SPP/SPN 遵循和 SON 和 PON 相同的编程规则。即：如果编程了一个进给轴运动，则只释放一段冲程。

9.4.2 针对轨迹轴的特性

MD26010

所有通过机床数据

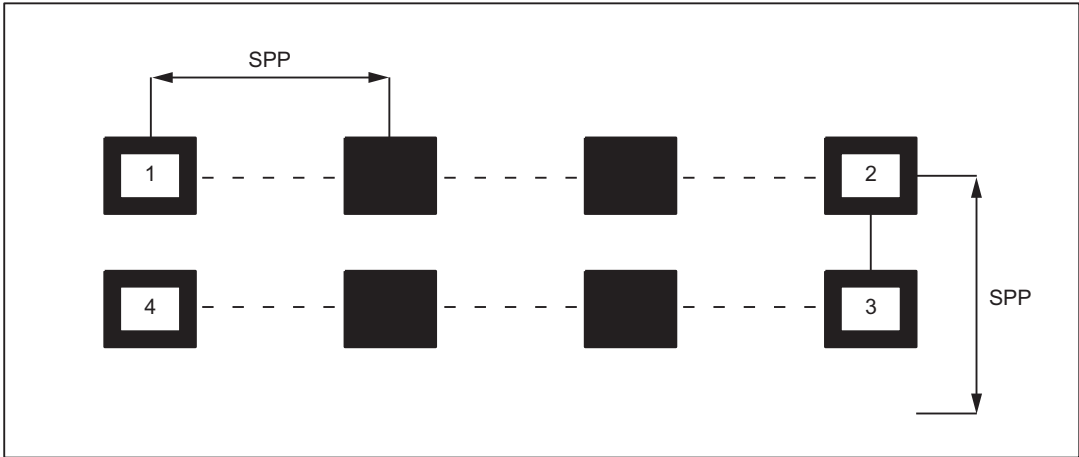
MD26010 \$MC_PUNCHNIB_AXIS_MASK

定义的轴和程序轴都可以用 SPP 和 SPN 等分行程。可旋转的刀具轴同样如此。每个轴的运行方式可以单独设置。

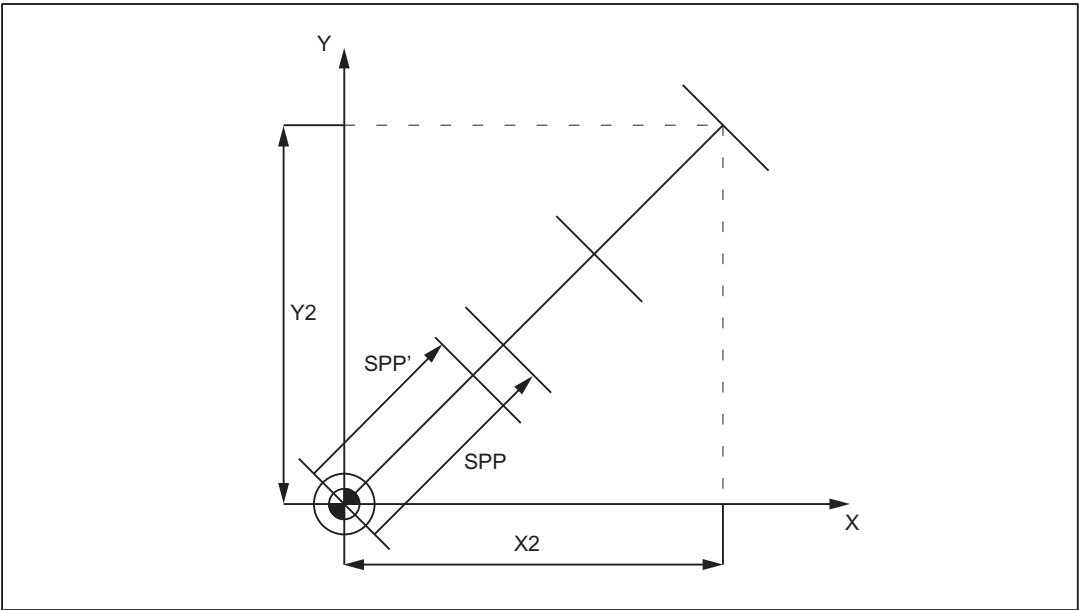
9.4 自动划分行程

示例 SPP

程序代码	注释
N1 G01 X0 Y0 SPOF	; 定位，不释放仲裁
N2 X75 SPP=25 SON	; 子行程为 25 mm 的步冲；
	; 在第一次运动及每段子行程后
	; 释放仲裁。
:	
:	
N3 Y10	; 以下调的 SPP 值定位，因为
:	; 总行程 < SPP 值，且在运动后
	; 释放仲裁。
:	
:	
N4 X0	; 每段子行程后
	; 回退并释放仲裁。
:	



如果编程的子行程无法被总行程整除，则系统会自动降低子行程：



X2/Y2: 编程的总行程
SPP: 编程的 SPP 值
SPP': 自动取整的子行程

图 9-4 行程划分

示例 SPN

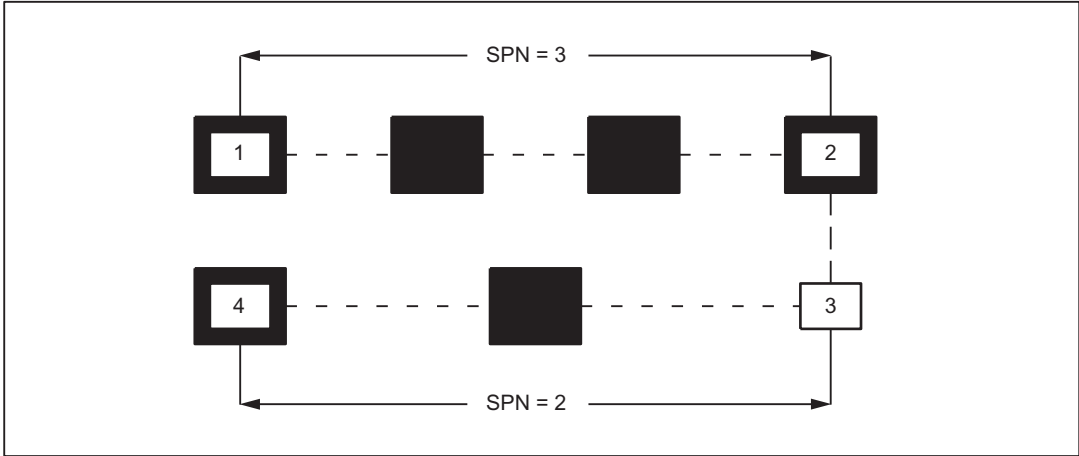
通过 SPN 可以编程每个程序段的子行程数量。

SPN 值在冲裁和步冲时都以非模态方式生效。两种运行方式唯一的区别在于第一次冲程。通常在步冲中，第一个子行程开始时即释放冲程，而在冲裁中，在第一个子行程结束时才释放冲程。也就是说，如果编程了 n 个子行程，则在冲裁中释放 n 个冲程，而在步冲中则释放 $n+1$ 个冲程。此外，如果没有运行信息，即使编程了多个子行程，也只执行一个冲程。如果需要在某个位置上进行多次冲程，则必须编程相应数量的多个无运行信息的程序段。

程序代码	注释
N1 G01 X0 Y0 SPOF	; 定位，不释放冲裁
N2 X75 SPN=3 SON	; 激活步冲。 总行程划分为
	; 3 段子行程。 第一次运动前
	; 和每段子行程后
	; 释放冲程。
:	

9.4 自动划分行程

程序代码	注释
:	
:	
:	
N3 Y10 SPOF	; 定位，不释放仲裁
N4 X0 SPN=2 PON	; 激活仲裁。 总行程划分为 2
	; 段子行程。 由于仲裁已激活，
	; 在第一段子行程末尾
	; 触发第一个冲程。
:	
:	
:	
:	



9.4 自动划分行程

程序代码	注释
N170 G02 G91 X-62.5 Y62.5 I0 J62.5 SON	; 通过插补参数进行增量圆弧插补,
	; 步冲
	; 激活
N180 G00 G90 Y300 SPOF	; 定位

9.4.3 针对单根轴的特性

MD26016

除了轨迹轴之外的单根轴的行程自动均分为若干中间程序段。

在以下章节中，附加回转轴 C 轴定位为“同步轴”。

如果该轴还被定义作为“冲裁/步冲轴”：

MD26010 \$MC_PUNCHNIB_AXIS_MASK = 1,

根据机床数据：

MD26016 \$MC_PUNCH_PARTITION_TYPE

，会得出不同的同步轴特性。

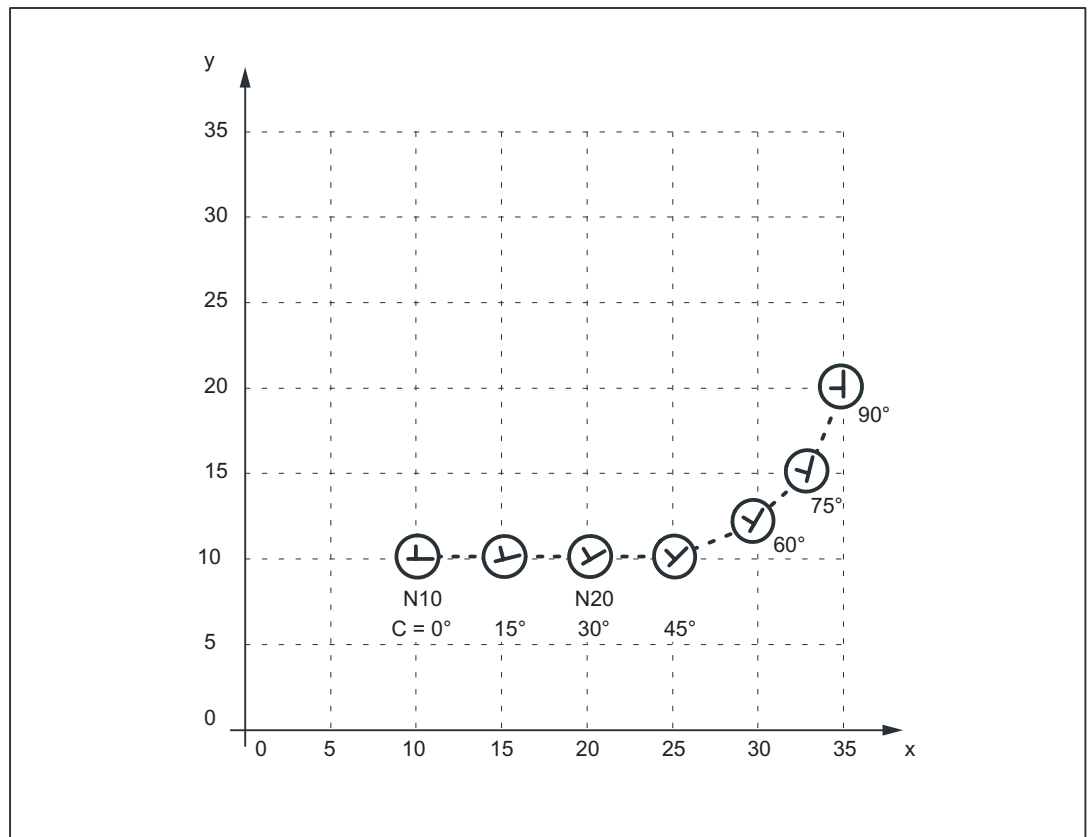
编程示例：

程序代码
N10 G90 G1 PON X10 Y10 C0 F10000
N20 SPP=5 X25 C45
N30 G3 SPN=3 X35 Y20 I0 J10 C90

MD26016 \$MC_PUNCH_PARTITION_TYPE=0 （缺省设置）

在该缺省设置中，在所有的插补方式中，编程的辅助轴运动按写入的行程划分指令被分为多个中间程序段。

程序段 N20 划分为三个中间程序段，每个中间程序段 C 轴旋转 15°。程序段 N30 中轴的属性与圆弧插补相同（三个子程序段，每个各旋转 15°）。

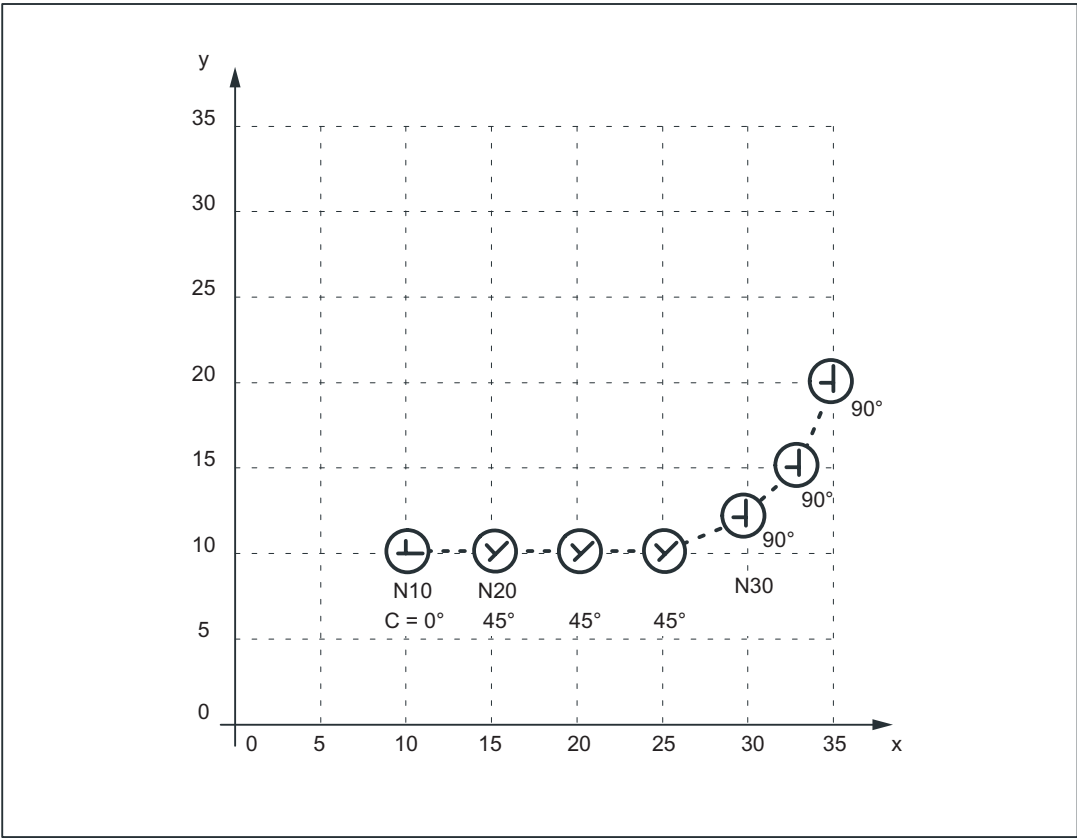


MD26016 \$MC_PUNCH_PARTITION_TYPE=1

和上面描述的属性相反，在该设置中，同步轴编程的总旋转量位于写入行程划分指令的第一个程序段中。

在示例中，C 轴在 X 轴位置 X=15 就已经到达程序段中编程的终点位置 C=45。在后面的圆弧插补程序段中，轴属性相同。

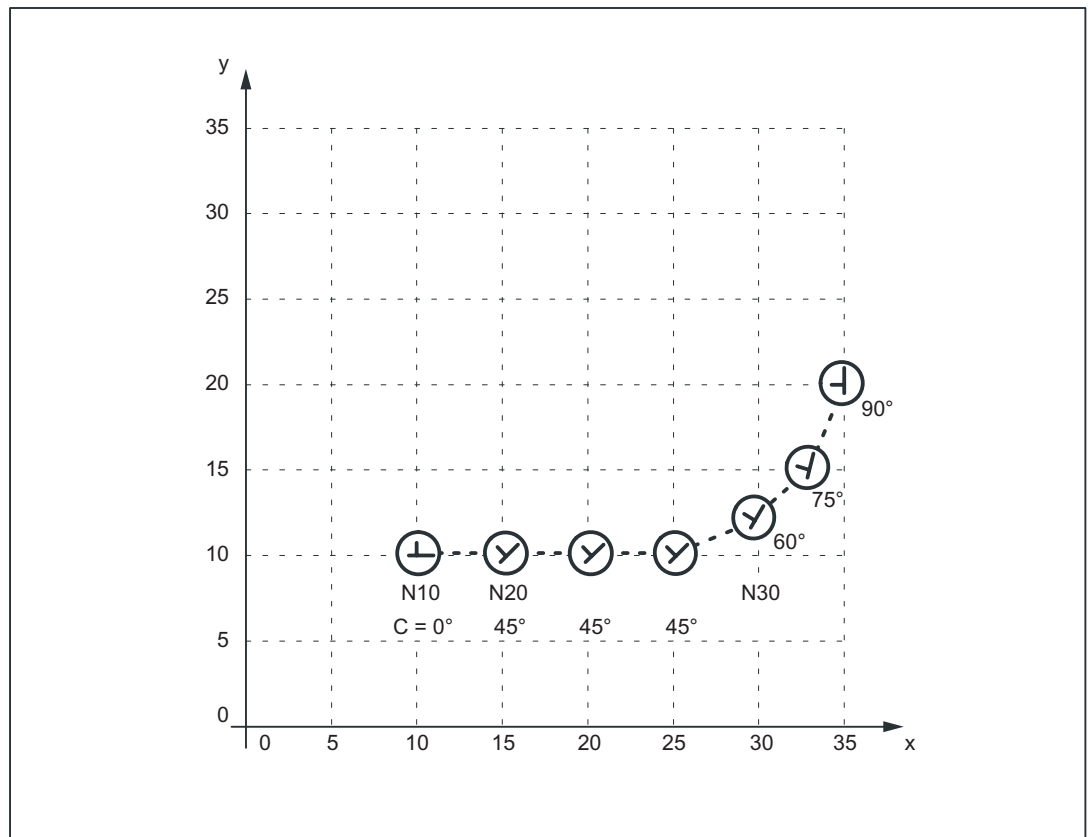
9.4 自动划分行程



MD26016 \$MC_PUNCH_PARTITION_TYPE=2

如果只需要在直线插补时轴的属性同上，而在圆弧插补时为缺省属性（见第 1 种情况），则可设置 MD26016=2。

在示例中：程序段 N20 的第一个子程序段中，C 轴旋转到 C=45°。在后面的圆弧插补程序段中，在每个子程序段中，C 轴旋转 15°。



上述说明的属性特别适用于可旋转的刀具轴，因为它可以在定义的方向上使刀具靠近轮廓，而不需要使用切向控制。但它不能替代切向控制功能，因为仍必须编程旋转轴起点和终点位置。

说明

辅助轴（此处为回转轴 C）的额外补偿运动通过零点偏移实现。

9.4 自动划分行程

边界条件

- 如果没有将 C 轴定位为“冲裁/步冲轴”，则在上述示例的程序段 N30 中不划分 C 轴行程，并且不会在程序段结束时释放冲程。
- 如果需要在非步冲工艺、但需要定位辅助轴的应用中使用该功能，则可以通过下列 PLC 信号抑制冲程的释放：

DB 21, 22 DBX3.2（冲程抑制）

（应用：例如：焊接时定位电子射线）

也可以通过下列机床数据的设置实现类似的属性：

MD26014 \$MC_PUNCH_PATH_SPLITTING=1

无需激活冲裁/步冲功能即可实现行程划分。

9.5 可旋转刀具

9.5.1 概述

功能概述

在带可旋转冲头和子刀具（模具）的冲裁/步冲机床上具备下列两种功能：

- 跟随功能
主刀具和子刀具相同的旋转
- 切向控制
用于冲头旋转轴定位，进行加工

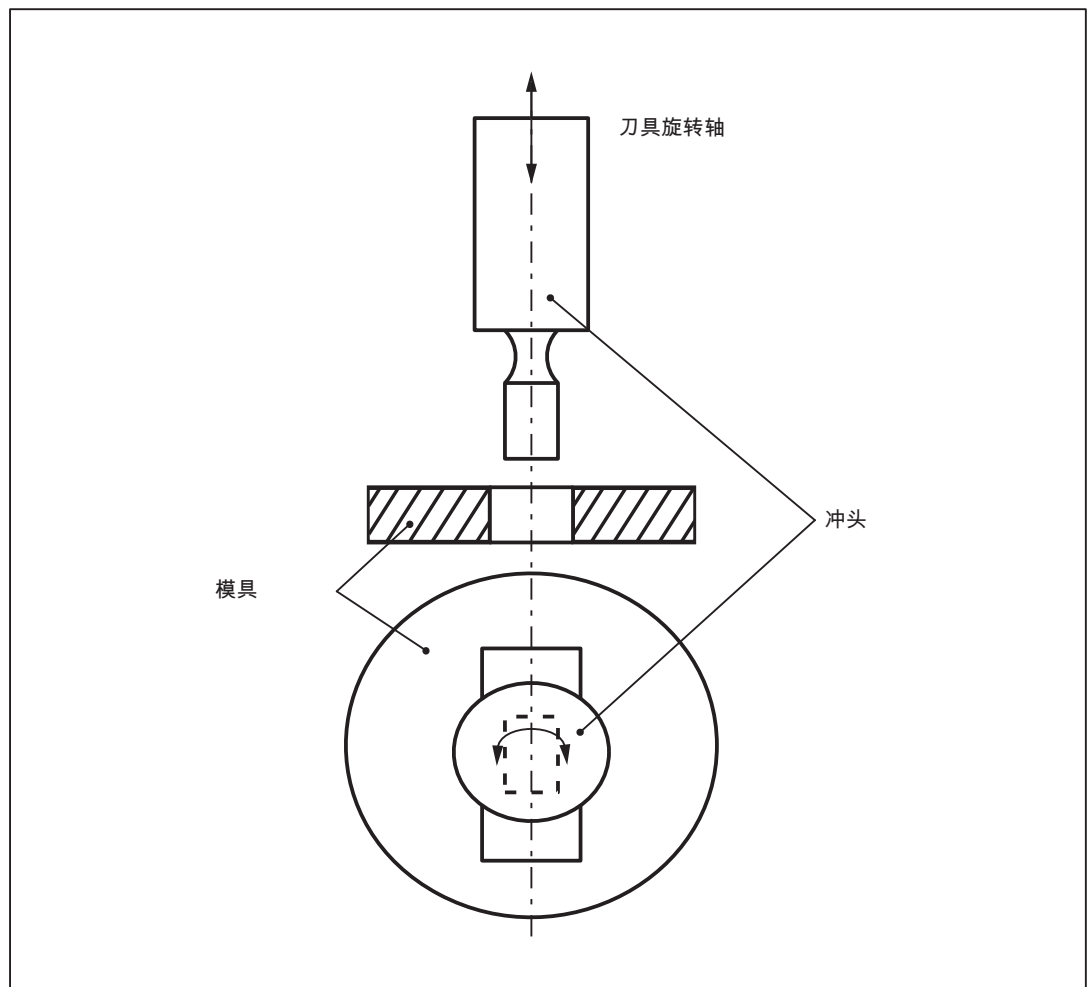


图 9-6 显示可旋转的刀具轴

9.5 可旋转刀具

9.5.2 主刀具和子刀具的“耦合”

功能

使用标准功能“耦合”可以将子刀具轴作为从动轴分配给可旋转的主刀具轴。

激活

语言指令 TRAILON 或者 TRAILOF 可以激活或取消激活耦合功能。

文档：
功能手册之特殊功能分册；章节“轴耦合和 ESR(M3)”

示例

示例：典型的冲床，C 轴为可旋转的主刀具轴，C1 子刀具轴：

程序代码	注释
:	
:	
TRAILON (C1, C, 1)	; 激活跟随组
G01 X100 Y100 C0 PON	; 释放冲程, C 轴/ ; C1 轴位置 C=0=C1
X150 C45	; 释放冲程, C 轴/ ; C1 轴位置 C=45=C1
:	
:	
M30	

初始设置

系统启动后耦合组失效。一般在两个刀具轴回参考点后不再释放耦合组。

过程如下：

- 单次激活耦合组（见上例）
- 设置机床数据：
MD20110 \$MC_RESET_MODE_MASK, 位 8=1

这样一来，耦合组在复位/零件程序启动后便可继续保持生效。

9.5.3 切向控制

功能

使用功能“切向控制”可以使冲床上的可旋转刀具与编程轨迹相切定位。

激活

语言指令 TANGON 或者 TANGOF 可以激活或取消激活“切向控制”功能。

文档：
编程手册之工作准备分册

工作原理

相切轴和主动轴的插补关联在一起。因此相切轴不能实现和速度无关、与轨迹相切地到冲裁位置的定位。和主动轴相比，旋转轴的动态性能较差时，可能导致加工速度的降低。附加的偏移角可直接通过语言指令 TANGON 可以激活。

说明
如果需要通过 2 个单独的驱动定位刀具（冲模和凹模），则可以通过“切向控制”和“耦合”功能实现。
注意：必须首先激活“切向控制”，然后激活“耦合”功能。

切向控制功能将自动定位冲模，使它垂直于编程的轨迹方向矢量。在第一次释放冲程行程前首先定位相切刀具。相切角总是以 X 轴正方向为基准。编程的附加角和计算出的角度相加。
切向控制在直线插补和圆弧插补中生效。

示例：线性插补

冲裁/步冲机床具备可旋转、单独驱动的主刀具和子刀具。
编程示例：

程序代码	注释
:	
:	
N2 TANG (C, X, Y, 1, "B")	； 定义主动轴和从动轴， ； C 轴是 X 轴和 Y 轴的从动轴， ； 基本坐标系

9.5 可旋转刀具

程序代码	注释
N5 G0 X10 Y5	; 起点
N8 TRAILON (C1, C, 1)	; 激活可旋转 ; 刀具轴 C/C1 之间的耦合
N10 Y10 C225 PON F60	; C/C1 轴旋转到 225° → 释放冲程
N15 X20 Y20 C45	; C/C1 轴旋转到 45° → 释放冲程
N20 X50 Y20 C90 SPOF	; C/C1 轴旋转到 90°, 无 ; 冲程释放
N25 X80 Y20 SPP=10 SON	; 总行程划分为 4 个冲程 ; 刀具旋转到 90°
N30 X60 Y40 SPOF	; 定位
N32 TANGON (C, 180)	; 激活切向控制, ; 可旋转刀具轴的偏移角 180°
N35 X30 Y70 SPN=3 PON	; 总行程划分为 3 个冲程 ; 激活切向控制和 ; 180°的偏移角
N40 G91 C45 X-10 Y-10	; C/C1 旋转至 225° (180° + 45° INC) ; 切向控制无效, 因为没有 ; 行程划分 → 冲程
N42 TANGON (C, 0)	; 无偏移角的切向控制
N45 G90 Y30 SPN=3 SON	; 总行程划分为 3 个冲程 ; 无偏移角的切向控制
N50 SPOF TANGOF	; 取消冲程释放 + ; 切向控制
N55 TRAILOF (C1, C)	; 取消可旋转 ; 刀具轴 C/C1 之间的耦合
N60 M2	

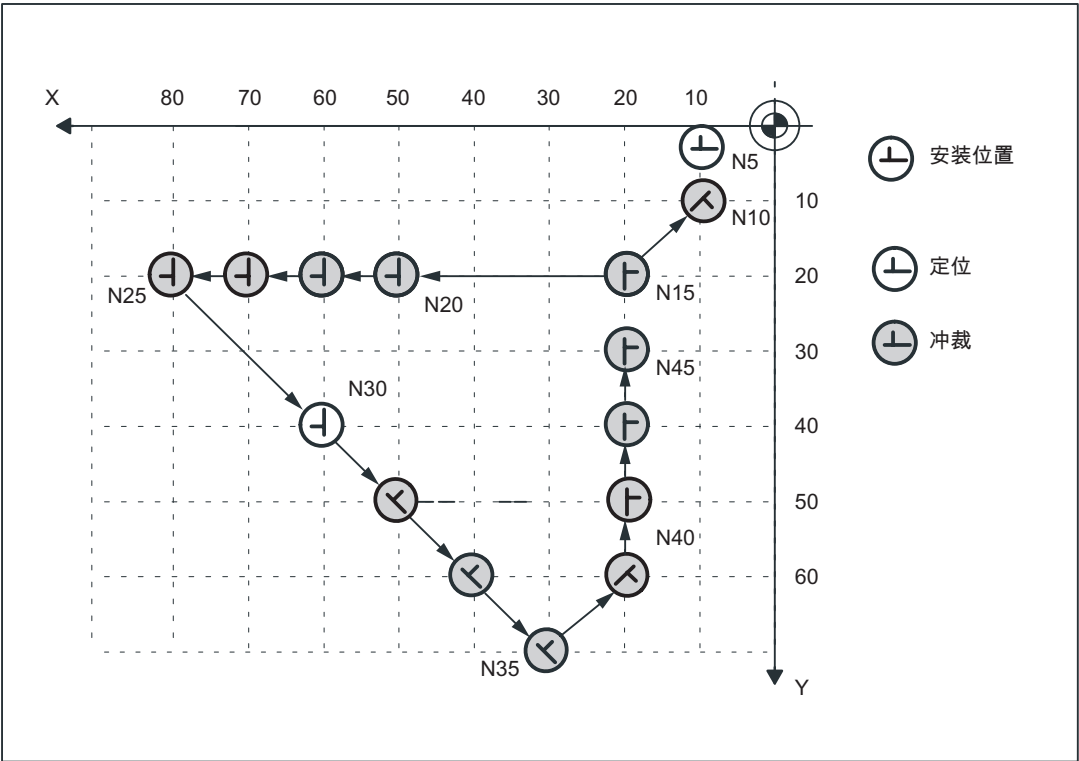


图 9-7 XY 平面中的编程示例

示例：圆弧插补

在圆弧插补以及行程划分生效时，在每个子程序段中刀具轴都旋转到和编程的轨迹轴相切的位置。

编程示例：

程序代码	注释
:	
:	
N2 TANG (C, X, Y, 1, "B")	; 定义主动轴和从动轴， ; C 轴是 X 轴和 Y 轴的从动轴， ; 基本坐标系
N5 G0 F60 X10 Y10	; 起始位置
N8 TRAILON (C1, C, 1)	; 激活可旋转 ; 刀具轴 C/C1 之间的耦合 ; ，一个是主刀具轴，一个是子刀具轴
N9 TANGON (C, -90)	; 激活切向控制 ; 偏移角为 270°
N10 G02 X30 Y30 I20 J0 SPN=2 PON	; 圆弧插补，划分为 ; 2 个冲程，启用 270°

9.5 可旋转刀具

程序代码	注释
	; 偏移角和与圆弧轨迹的
	; 相切定位
N15 G0 X70 Y10 SPOF	; 定位
N17 TANGON (C, 90)	; 激活切向控制
	; 偏移角为 90°
N20 G03 X35,86 Y24,14 CR=20 SPP=16 SON	; 圆弧插补, 划分为 4 个
	; 冲程, 启用 90°
	; 偏移角和与圆弧轨迹的
	; 相切定位
N25 G0 X74,14 Y35,86 C0 PON	; 刀具轴旋转至
	; 0°, 冲程
N27 TANGON (C, 0)	; 激活切向控制
	; 偏移角为 0°
N30 G03 X40 Y50 I-14,14 J14,14 SPN=5 SON	; 圆弧插补, 划分为
	; 5 个冲程, 偏移角为 0°
	; 并启用与圆弧轨迹的
	; 相切定位
N35 G0 X30 Y65 C90 SPOF	; 定位, 不激活
	; 切向控制
N40 G91 X-10 Y-25 C180	; 定位, C 轴旋转到 270°
N43 TANGOF	; 关闭切向控制
N45 G90 G02 Y60 I0 J10 SPP=2 PON	; 圆弧插补, 划分为
	; 2 个冲程, 不激活切向控制
	; C=270°
N50 SPOF	; 关闭冲裁
N55 TRAILOF (C1, C)	; 取消可旋转
	; 刀具轴 C/C1 之间的耦合
N60 M2	

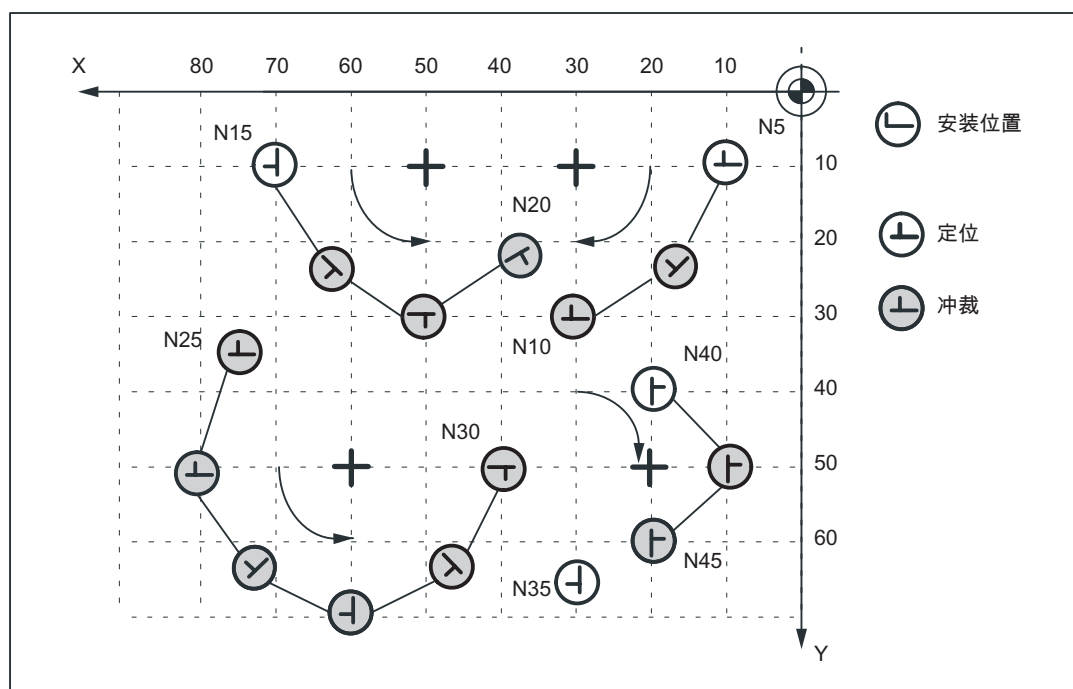


图 9-8 XY 平面中的编程示例

9.6 保护区

夹钳死区

“夹钳死区”功能是“保护区”功能范畴内的子功能。其只用于监控夹钳和刀具是否可能碰撞。

说明

如果超出夹钳死区，则**不能**采取绕行方案。

文档：

功能手册之基本功能分册；章节“轴监控，保护区 (A3)”

9.8 示例

9.7 边界条件

“仲裁和步冲”功能的可用性

该功能是一个软件选件，授权后方可和硬件组合使用。

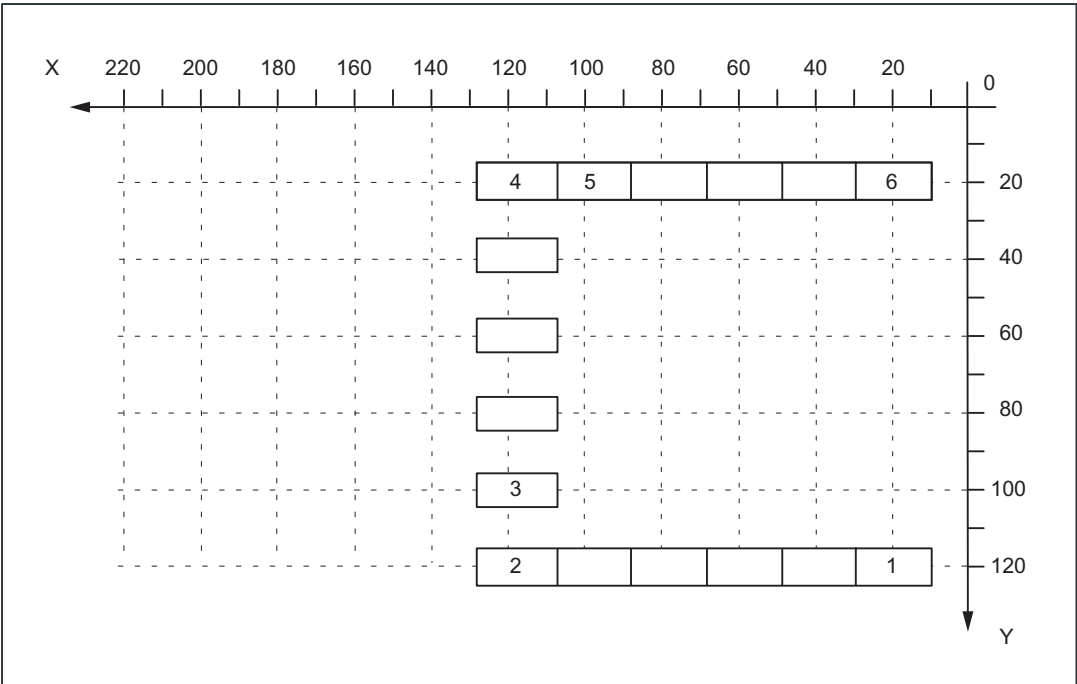
9.8 示例

9.8.1 定义步冲开始的示例

示例 1

定义步冲开始的示例。

程序代码	注释
:	
:	
N10 G0 X20 Y120 SPP= 20	; 移动到位置 1
N20 X120 SON	; 定义的步冲开始，首次冲程到
	; “1”，最后冲程到“2”
N30 Y20	; 定义的步冲开始，首次冲程到
	; “3”，最后冲程到“4”
N40 X20	; 定义的步冲开始，首次冲程到
	; “5”，最后冲程到“6”
N50 SPOF	
N60 M2	

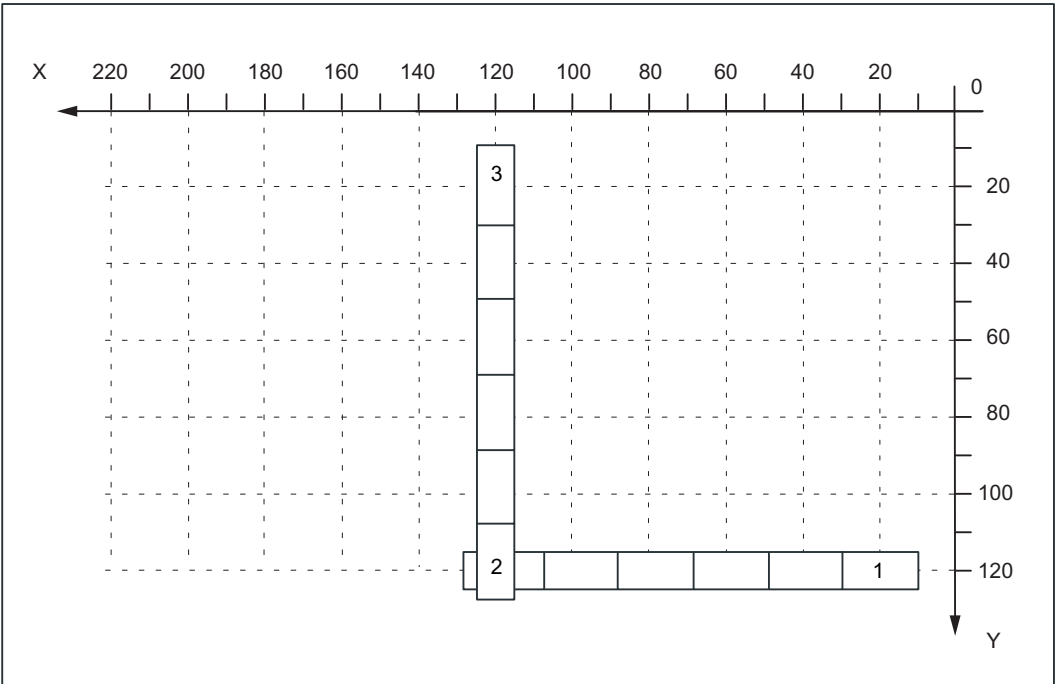


示例 2

该示例使用功能“切向控制”。相切轴为 Z 轴。

程序代码	注释
:	
:	
N5 TANG (Z, X, Y, 1, "B")	; 相切轴定义
N8 TANGON (Z, 0)	; 选择切向控制
N10 G0 X20 Y120	; 移动到位置 1
N20 X120 SPP=20 SON	; 定义的步冲开始, ; 激活切向控制, ; 首次冲程到"1", 最后冲程到"2"
N30 SPOF TANGOF	; 取消步冲并取消 ; 切向控制
N38 TANGON (Z, 90)	; 选择切向控制
N40 Y20 SON	; 定义的步冲开始, ; 激活切向控制, ; 首次冲程到"2", 旋转 90 度至 ; 程序段 N20, 最后冲程至"3"
N50 SPOF TANGOF	; 取消步冲并取消 ; 切向控制
N60 M2	

9.8 示例



定义步冲开始的示例 3 和 4

示例 3： SPP 编程

程序代码	注释
:	
:	
N5 G0 X10 Y10	; 定位
N10 X90 SPP=20 SON	; 定义的步冲开始,
	; 5 次冲程释放
N20 X10 Y30 SPP=0	; 在行程终点释放冲程
N30 X90 SPP=20	; 4 次冲程释放, 间隔 20 mm
N40 SPOF	
N50 M2	

示例 4： SPN 编程

程序代码	注释
:	
:	

程序代码	注释
N5 G0 X10 Y10	; 定位
N10 X90 SPN=4 SON	; 定义的步冲开始， ; 5 次冲程释放
N20 X10 Y30 PON	; 在行程终点释放冲程
N30 X90 SPN=4	; 4 次冲程释放
N40 SPOF	
N50 M2	

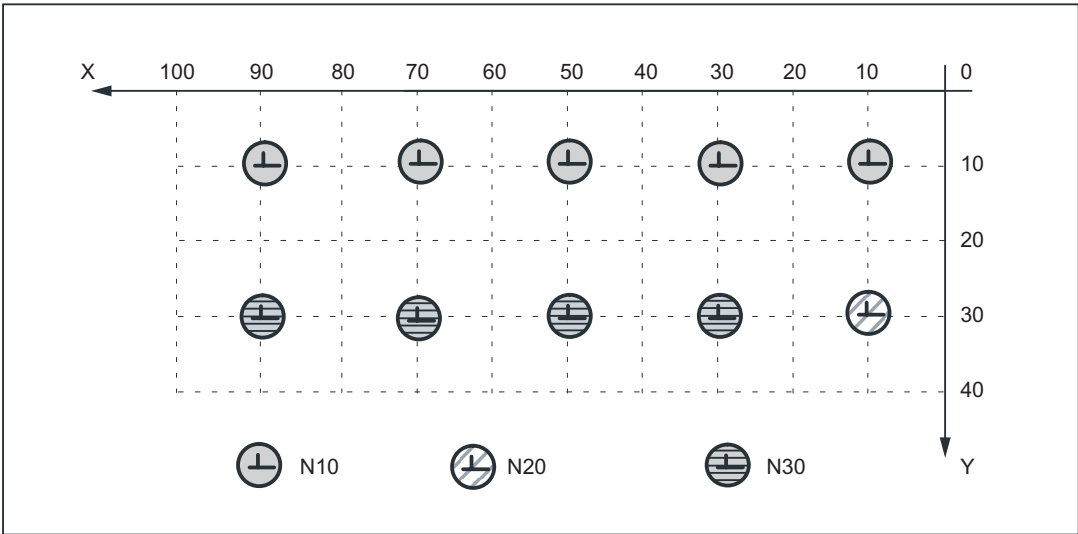


图 9-9 定义步冲开始的示例 3 和 4

没有定义步冲开始的示例 5 和 6

示例 5: SPP 编程

程序代码	注释
:	
:	
N5 G0 X10 Y30	; 定位
N10 X90 SPP=20 PON	; 未定义冲程开始， ; 4 次冲程释放
N15 Y10	; 在行程终点释放冲程
N20 X10 SPP=20	; 4 次冲程释放，间隔 20 mm
N25 SPOF	

9.8 示例

程序代码	注释
N30 M2	

示例 6：SPN 编程

程序代码	注释
:	
:	
N5 G0 X10 Y30	; 定位
N10 X90 SPN=4 PON	; 未定义冲程开始， ; 4 次冲程释放
N15 Y10	; 在行程终点释放冲程
N20 X10 SPN=4	; 4 次冲程释放
N25 SPOF	
N30 M2	

示例 7：SPP 编程的应用示例

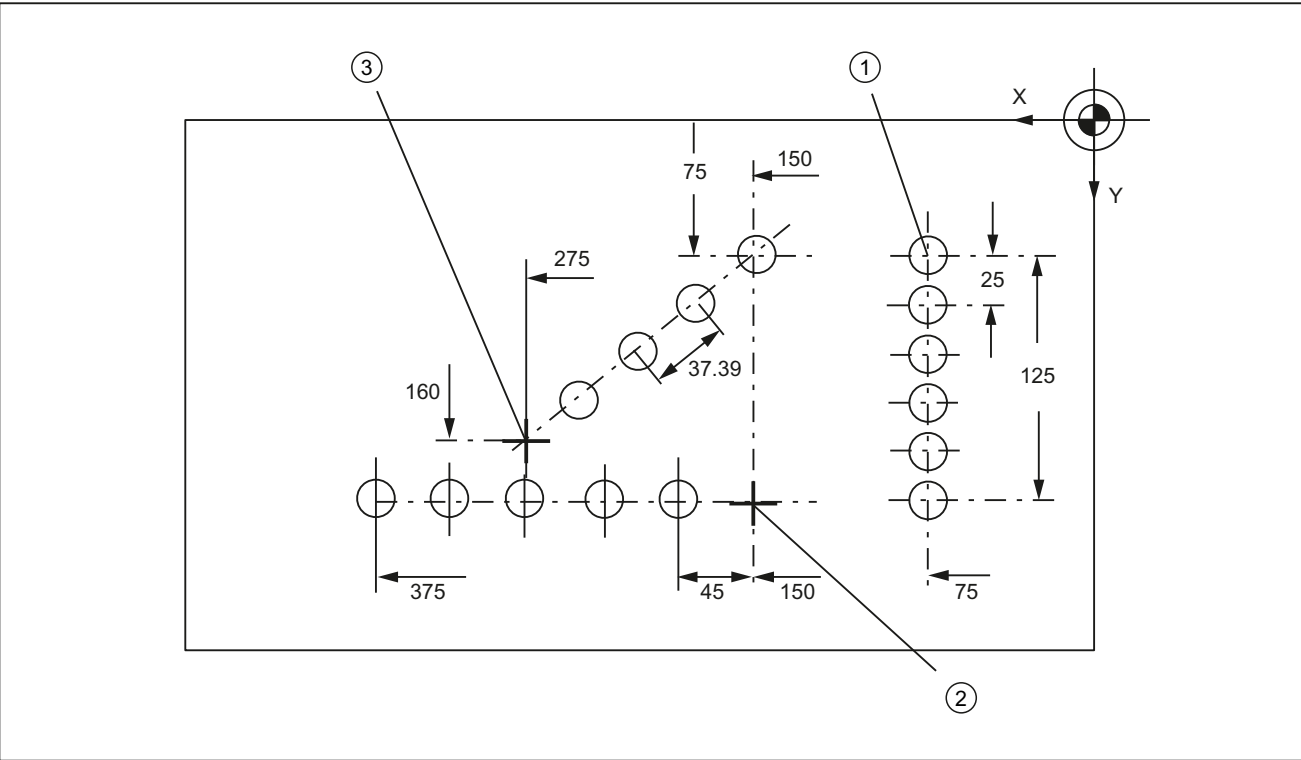


图 9-10 工件

程序段:

程序代码	注释
N100 G90 X75 Y75 F60 PON	; 定位到垂直成排孔的起点 (1) 处
	; 冲裁单孔
N110 G91 Y125 SPP=25 PON	; 终点坐标 (增量),
	; 子行程: 25 mm, 激活冲裁
N120 G90 X150 SPOF	; 绝对坐标, 定位到
	; 水平成排孔的起点 (2) 处
N130 X375 SPP=45 PON	; 终点坐标, 子行程: 45 mm
N140 X275 Y160 SPOF	; 定位至斜向成排孔的起点 (3) 处
	;
N150 X150 Y75 SPP=40 PON	; 终点坐标, 编写的
	; 子行程: 40 mm, 计算出的
	; 子行程: 37.39 mm
N160 G00 Y300 SPOF	; 定位

9.9 数据表

9.9.1 机床数据

9.9.1.1 通用机床数据

编号	名称: \$MN_	描述
11450	SEARCH_RUN_MODE	编程搜索模式

9.9.1.2 通道专用机床数据

编号	名称: \$MC_	描述
20150	GCODE_RESET_VALUES[n]	G 功能组的初始设置
26000	PUNCHNIB_ASSIGN_FASTIN	冲程控制时输入字节的硬件指定
26002	PUNCHNIB_ASSIGN_FASTOUT	冲程控制时输出字节的硬件指定
26004	NIBBLE_PUNCH_OUTMASK[n]	快速输出位的掩码
26006	NIBBLE_PUNCH_INMASK[n]	快速输入位的掩码
26008	NIBBLE_PUNCH_CODE[n]	定义 M 功能

9.9 数据表

编号	名称: \$MC_	描述
26010	PUNCHNIB_AXIS_MASK	定义仲裁轴和步冲轴
26012	PUNCHNIB_ACTIVATION	激活仲裁和步冲功能
26014	PUNCH_PATH_SPLITTING	激活自动划分行程
26016	PUNCH_PARTITION_TYPE	自动划分行程时的单轴特性
26018	NIBBLE_PRE_START_TIME	可自动激活的预释放时间
26020	NIBBLE_SIGNAL_CHECK	输入信号监控

9.9.2 设定数据

9.9.2.1 通道专用设定数据

编号	名称: \$SC_	描述
42400	PUNCH_DWELL_TIME	暂停时间
42402	NIBPUNCH_PRE_START_TIME	预释放时间
42404	MINTIME_BETWEEN_STROKES	两个连续冲程之间的最短时间

9.9.3 信号

9.9.3.1 发送至通道的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
无冲程释放	DB21, ...DBX3.0	-
手动冲程释放	DB21, ...DBX3.1	-
封锁冲程	DB21, ...DBX3.2	-
冲程未运行	DB21, ...DBX3.3	-
冲程延时	DB21, ...DBX3.4	-
手动冲程释放	DB21, ...DBX3.5	-

9.9.3.2 来自通道的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
冲程释放有效	DB21, ...DBX38.0	-
手动冲程释放应答	DB21, ...DBX38.1	-

9.9.4 语言指令

G 功能组	语言指令	含义	
35	SPOF	Stroke / Punch Off	取消冲裁和步冲
35	SON	Stroke On	激活步冲
35	SONS	Stroke On	激活步冲（位置控制器）
35	PON	Punch On	激活冲裁
35	PONS	Punch On	激活冲裁（位置控制器）
36	PDELAYON	Punch with Delay On	激活冲裁延时
36	PDELAYOF	Punch with Delay Off	取消冲裁延时
行程划分			
	SPP		每个冲程的行程，模态生效
	SPN		每个程序段的冲程数量，非模态生效

P2: 定位轴

10.1 简要说明

用于辅助运动的轴

现代机床上除了需要配备一些轴执行工件加工外，还需配备一些轴执行辅助运动，例如：

- 用于旋转刀库的轴
- 用于旋转刀塔的轴
- 用于传送工件的轴
- 用于传送托盘的轴
- 用于移动装刀器的轴（也可为多轴）
- 用于移动机械手的轴
- 用于移动套筒/顶尖的轴

执行工件加工的轴被称为轨迹轴。这些轴在通道内由插补器控制，从而同时启动、加速、到达终点和停止。

执行辅助运动的轴则以单独的进给率独立于轨迹轴运行。这类轴过去很多都采用液压驱动，并由零件程序通过辅助功能启动。借助 NC 中的轴闭环控制，可在零件程序中通过名称访问这些轴，以及在屏幕上显示其实际位置。

说明

“定位轴/辅助主轴”选项

用于辅助运动的轴必须不是插补轴（“真正的 NC 轴”）。“定位轴/辅助主轴”选项提供辅助轴功能，用于执行辅助运动。

功能限制

被选择作为定位轴/辅助主轴时，轴的功能受到一定限制。此类轴不支持以下功能：

- 用作轨迹轴、几何轴或轨迹辅助轴
- 加入几何轴组（GEOAX）
- 螺纹切削和刚性攻丝

调试

缺省设置下，轴被定义作为插补轴：

MD30460 \$MA_BASE_FUNCTION_MASK, 位 8 = 0

若需将一根轴定义为功能受限的定位轴/辅助主轴运行，必须将位 8 的值设置为“1”：

MD30460 \$MA_BASE_FUNCTION_MASK, 位 8 = 1

功能

通过“定位轴”功能可将执行辅助运动的轴更方便地集成至控制系统：

- 编程时：
定位轴和工件加工轴编写在同一零件程序中，这样便不会浪费宝贵的加工时间。
有一些针对定位轴的特殊运行指令（POS、POSA）。
- 程序测试/程序试运行：
程序测试和程序试运行针对所有轴有效。
- 操作时：
加工启动以及加工过程监控针对所有轴有效。
- PLC 配置/调试时：
在 PLC 或外部计算机（PC）中不需要考虑加工轴和辅助轴之间的同步机制。
- 系统配置时：
不需要另一个通道。

运动和插补

每个通道都有一个轨迹插补器以及至少一个轴插补器，其具有以下插补功能：

- 轨迹插补器：
直线插补（G1）、圆弧插补（G2 / G3）、样条插补等。
- 轴插补器：
编写了一根定位轴时，控制系统中将启动一个轴插补器（直线插补 G1）。
- 运动结束条件：
满足运动结束条件 FINEA、COARSA、IPOENDA 时，即表明到达编写的定位轴终点位置。
- 快速移动中的轨迹轴：
快速移动（G0）中可选择通过线性插补或非线性插补运行轨迹轴。
- 自控单轴运动：
通过静态同步动作启动的单个 PLC 轴、指令轴或异步往复轴可独立于 NCK 进行插补。此时，通过主处理插补的进给轴/主轴响应不受 NC 程序影响。由程序运行触发的通道特性失效，以便将特定轴/通道的控制权转交给 PLC。
- 通过 PLC 控制：
通常情况下，所有通道专用信号对轨迹轴和定位轴具有同等作用。
定位轴还可通过附加的专用信号进行控制。
PLC 轴由 PLC 通过基本程序中特殊的功能块运行，可以与所有其他的轴异步运行。其运行不受轨迹运动和同步动作的影响。

10.2 独立通道、定位轴或并行定位轴

若机床上配备了执行辅助运动的轴，则须基于要求的属性确定该轴是否：

- 在一个独立的零件程序中编写（参见“独立通道 - 仅用于 840D sl (页 641)”章节）。
- 与加工轴在同一个零件程序中编写（参见“定位轴 (页 642)”章节）。
- 只由 PLC 在加工期间触发（参见“并行定位轴 (页 645)”章节）。

10.2.1 独立通道 - 仅用于 840D sl

一个通道代表一个数控系统，它可通过零件程序对进给轴和主轴的运动以及机床功能进行控制，而不受其他通道的影响。

通道间的独立性

以下因素确保了通道间的独立性：

- 每个通道一个生效的零件程序
- 通道专用接口信号，如：
 - DB21, ... DBX7.1 (NC 启动)
 - DB21, ... DBX7.3 (NC 停止)
 - DB21, ... DBX7.7 (复位)
- 每个通道一个进给倍率
- 每个通道一个快速移动倍率
- 单通道分析和显示报警
- 单通道显示以下数据：
 - 轴位置
 - 生效的 G 功能
 - 生效的辅助功能
 - 当前程序段
- 单通道测试和程序控制：
 - 单程序段
 - 试运行进给率 (DRY RUN)
 - 程序段搜索
 - 程序测试

文档

通道功能的更多信息请见：

功能手册之基本功能分册；章节“BAG、通道、程序运行、复位特性(K1)”

10.2.2 定位轴

定位轴和轨迹轴（即执行工件加工的轴）在同一个零件程序中编写。

其中可将定位轴指令和轨迹轴指令编写在同一个程序段中。但即便处于同一程序段中，轨迹轴和定位轴也不会相互插补，并且不会同时到达程序段终点（无时间耦合，另见“运动特性和插补过程 (页 646)” 章节）。

定位轴类型和程序段切换

程序段切换时间点取决于所编写的定位轴类型（另见“程序段切换 (页 661)” 章节）：

类型	描述
1	当所有轨迹轴和所有定位轴到达编写的终点时，执行程序段切换。
2	当所有轨迹轴到达编写的终点时，执行程序段切换。 采用类型 2 的定位轴时，可在经过多个程序段后逼近编写的终点。
3	满足轨迹插补的运动结束条件或程序段切换条件时，可在单轴插补的制动斜坡内执行程序段切换。

运动同步

通过定位轴可从同一加工程序中触发辅助运动，并在程序段边界处（类型 1）或通过 WAITP 指令明确设定的位置（类型 2）实现运动同步。

在制动斜坡内进行程序段切换的运动结束条件

针对单轴插补可设置一个运动结束条件，以便在制动斜坡内便可进行程序段切换。

G0 下将轨迹轴作为定位轴运行

快速移动（G0）中每根轨迹轴都可作为定位轴运行。此时所有轴都相互独立地运行至终点。

因此在 G0 下依次编写 X 轴 和 Z 轴，可将这两根轴作为定位轴运行。可基于设置的制动斜坡时间点（100-0%）由 X 轴触发向 Z 轴的程序段切换。X 轴运行期间 Z 轴即已启动。两根轴相互独立地运行至终点。

轴类型

定位轴既可为线性轴也可为回转轴。

定位轴还可以是分度轴。

定位轴和轨迹轴之间的独立性

以下因素确保了定位轴和轨迹轴之间的独立性：

- 不相互插补
- 每根定位轴一个单独的插补器

- 每根定位轴单独的进给倍率
- 单独的可编程进给率
- 单独的接口信号“删除单轴剩余行程”。

关联性

定位轴和轨迹轴之间有以下关联性：

- 共用一个零件程序
- 只能在零件程序中的程序段边界处启动定位轴
- 在快速移动 G0 中将轨迹轴作为定位轴运行时，可选择采用两种不同的模式。
- 无快速移动倍率
- 以下接口信号对整个通道生效，因此也对定位轴生效：
 - DB21, ... DBX7.1 (NC 启动)
 - DB21, ... DBX7.3 (NC 停止)
 - DB21, ... DBX7.7 (复位)
 - DB21, ... DBX6.1 (读取禁止)
- 程序专用和通道专用的报警也会使定位轴停止。
- 程序控制（试运行进给率、程序测试、DRF 等）也对定位轴生效。
- 程序段搜索和单程序段方式也对定位轴生效。
- 搜索中处理的最后一个含运行结束条件的程序段用于保存所有轴的设置。
- G 功能组 1（模态运行指令，包含 G0、G1、G2...）不适用于定位轴。

文档：

编程手册之基本原理分册

应用

定位轴的典型应用有：

- 单轴装刀器
- 无插补的多轴装刀器（PTP → 点对点运行）
- 送料器和传料器

其他应用还有：

- 在 G0 下送料器和传料器相互独立地运行至终点。
- 在执行多个连续工序的机床上：在单轴插补的制动斜坡内便可进行程序段切换，显著减少加工步骤。

说明

定位轴不适用于需要轴间插补（轨迹插补器）的多轴装刀器。

10.2.3 并行定位轴

并行定位轴是具有以下属性的定位轴：

- 不必在程序段界限处由 PLC 触发，而是能在任意运行方式下的任意时间点触发（在已有零件程序运行的通道中也可以触发）。
- 为了在上电后通过零件程序实现并行定位轴的立即运动，需要使用 WAITP 指令。
- 即使并行定位轴尚未到达 PLC 设定的位置，零件程序也不受影响地继续运行。
- 根据机床数据 MD30552 \$MA_AUTO_GET_TYPE 中的设置可能会自动取轴。
- 通过程序指令：
 - GET(<轴>) 或 WAITP(<轴>) 可将并行定位轴恢复为通道轴。
 - RELEASE(<轴>) 或 WAITP(<轴>) 可将通道轴设为并行定位轴，由 PLC 控制。

由 PLC 触发

在 SINUMERIK 840D sl 中，并行定位轴由 PLC 通过 FC18 触发。

- 进给率
进给率为 0 时将采用以下机床数据中的进给率：
MD32060 \$MA_POS_AX_VELO（定位轴速度的初始设置）
- 绝对尺寸（G90），增量尺寸（G91）
回转轴以最短行程绝对定位（<回转轴名称>=DC(<值>））

下列功能采用固定设定：

- 线性插补（G1）
- 进给率，单位：毫米/分钟或度/分钟（G94）
- 准停（G9）
- 当前选择的可设定零点偏移生效

10.3 运动特性和插补过程

应用

并行定位轴的典型应用有：

- 加工期间需要手动装刀和卸刀的刀库
- 加工期间需要准备刀具的刀库

10.3 运动特性和插补过程

10.3.1 轨迹插补器和轴插补器

轨迹插补器

每个通道都有一个轨迹插补器用于执行各种类型的插补，例如直线插补（G1）、圆弧插补（G2/G3）、样条插补等。

轴插补器

除轨迹插补器外，每个通道还配备有轴插补器。轴插补器的最大数量取决于通道轴的最大数量。

编写了一根定位轴时，控制系统中将启动一个轴插补器用于直线插补 G1。该轴插补器一直独立于轨迹插补器运行，直至到达编写的定位轴终点位置。

轨迹插补器和轴插补器间以及轴插补器相互之间不存在时间上的关联。

定位轴无法采用连续路径运行（G64）。

满足运动结束条件 FINEA、COARSA 或 IPOENDA 时，即表明到达编写的定位轴终点位置。

10.3.2 G0 下轨迹轴的插补特性

快速移动（G0）中可选择通过线性插补或非线性插补运行轨迹轴。

线性插补

特性：

- 轨迹轴一同进行插补。
- 使用 G0 编写的刀具运动会以最快的运行速度执行（快速移动）。

- 快速移动速度在以下机床数据中针对各轴分别定义：
MD32000 \$MA_MAX_AX_VELO
 - 如果同时在多根轴上执行快速移动，那么快速移动速度由所需时间最长的轴来决定。
- 以下情形中总是执行线性插补：
- 将定位轴运动不支持的 G 代码与 G0 组合使用时，例如：
G40、G41、G42、G96、G961 和 MD20750 \$MC_ALLOW_G0_IN_G96 == FALSE
 - 将 G0 和 G64 组合使用时
 - 压缩器生效，或转换生效时
 - 点对点（PTP）运行中
 - 选择了轮廓手轮时（FD=0）
 - 包含几何轴旋转的框架生效时
 - 几何轴步冲生效时

非线性插补

特性：

- 每根轨迹轴作为单轴（定位轴）独立于其他轴通过以下机床数据中设定的快进速度插补：
MD32000 \$MA_MAX_AX_VELO
- 通过 PLC 和同步动作对所有编写作为轨迹轴的定位轴启用通道特定的剩余行程删除。

在非线性插补中，轴急动方面采用：

- 针对相应轴的设置（BRISKA、SOFTA、DRIVEA）

或者

10.3 运动特性和插补过程

- 以下机床数据中的设置：
MD32420 \$MA_JOG_AND_POS_JERK_ENABLE
与
MD32430 \$MA_JOG_AND_POS_MAX_JERK

不支持已存在的、基于轨迹剩余行程的系统变量（\$AC_PATH、\$AC_PLTBB 和 \$AC_PLTEB）。

 小心

碰撞危险
由于在非线性插补模式下可运行另一个轮廓，因此基于原始轨迹坐标的同步有时可能无效。

选择插补类型

G0 下生效的插补类型通过以下机床数据预设：
MD20730 \$MC_G0_LINEAR_MODE（G0 下的插补特性）

值	含义
0	快速移动（G0）中 非线性 插补生效。 轨迹轴作为定位轴运行。
1	快速移动（G0）中 线性 插补生效。 轨迹轴一同进行插补。

此外还可通过下面两个零件程序指令编写所需的 G0 下的插补特性，且不考虑预设：

- RTLIOF

取消线性插补。
⇒ 快速移动（G0）中**非线性**插补生效。
- RTLION

启用线性插补。
⇒ 快速移动（G0）中**线性**插补生效。

当前设置的 G0 下的轨迹轴插补特性可通过系统变量 \$AA_G0MODE 查询。

说明
在两种插补方式下，快进补偿均作为倍率针对特定通道生效。

10.3.3 自控单轴运动

功能

通过静态同步动作启动的单个 PLC 轴、指令轴或异步往复轴可独立于 NCK 进行插补。此时，通过主处理插补的进给轴/主轴的以下响应不受 NC 程序影响。

- NC 停止
- 报警处理
- 程序控制
- 程序结束
- 复位

边界条件

当前通过 NC 程序运行的进给轴/主轴无法由 PLC 控制。

针对由 PLC 控制的进给轴/主轴，**无法**通过非模态和模态同步动作启动指令轴运动。系统会发出报警 20143。

由 PLC 接收轴控制

顺序说明

1. PLC → NCK: 请求接收轴的控制权
DB31, ... DBX28.7 = 1 (PLC 控制轴)
2. NCK: 检查轴是否是主处理轴或中立轴。
3. NCK: 检查另一根轴是否允许由 PLC 控制。
4. NCK 确认控制权转交:
 - DB31, ... DBX63.1 = 1 (PLC 控制轴)
 - 系统变量 \$AA_SINGLAX_STAT = 1

结果: PLC 获得进给轴/主轴的控制权。

可能出现的情形

初始状态: 轴由 PLC 控制。通道停止指令使通道处于“中断”状态。

10.3 运动特性和插补过程

- 轴状态“未生效” ⇒
 - 取消停止状态。
 - 轴启动指令将直接触发轴运动。
- 轴状态“生效” ⇒
 - 不取消停止状态。
 - 按照**使用案例 2“停止轴”**生成轴状态。
 - 按照**使用案例 3“继续运行轴”**继续执行轴运动。
- 通道中执行复位 ⇒

此进程与“由 PLC 接收控制”不同步进行。上述两种情形都有可能出现，或者轴仍指定给通道并被复位。

边界条件

通过 NC 程序运行的进给轴/主轴不能转由 PLC 控制。只有通过静态同步动作运行或者作为往复轴、中立轴、并行定位轴或指令轴运行的进给轴/主轴的控制权才能被转交。

由 PLC 交还轴控制权

顺序说明:

1. PLC → NCK: PLC 将轴的控制权交还给 NCK
DB31, ... DBX28.7 = 0 (PLC 控制轴)
2. NCK: 检查是否有未处理的轴报警。
3. NCK: 检查是否有激活的运动尚未完成。如果是，则按照 **使用案例 2“停止轴”** 停止进给轴/主轴运动。
4. NCK: 按照 **使用案例 4“复位进给轴/主轴”** 执行轴复位。
5. NCK 确认控制权转交:
DB31, ... DBX63.0 = 0 (复位已执行)
DB31, ... DBX63.1 = 0 (PLC 控制轴)
DB31, ... DBX63.2 = 0 (轴停止生效)
系统变量 \$AA_SINGLAX_STAT = 0

结果: NCK 获得进给轴/主轴的控制权。

可能出现的情形

以下情形下，NCK 确认控制权转交，但是会在系统内部将进给轴/主轴的通道状态设置为“停止”：

- 通道处于“中断”状态。
- 存在针对通道的停止报警
- 存在针对 BAG 的停止报警

边界条件

进给轴/主轴必须处于 PLC 控制下。

仅当不存在轴报警时，NCK 才会确认获得进给轴/主轴的控制权。

基于使用案例的顺序说明**前提条件**

进给轴/主轴处于 PLC 控制下。

相关 NC/PLC 接口信号

可通过以下 NC/PLC 接口信号对 PLC 控制的进给轴/主轴进行控制，而不受 NC 程序的影响：

- DB21, ... DBX6.2（删除剩余行程）
- DB31, ... DBX28.1（复位）
- DB31, ... DBX28.2（继续）
- DB31, ... DBX28.6（沿制动斜坡停止）

自控单轴运动中 NC/PLC 接口上 NCK 和 PLC 间的信号流参见“通过 PLC 控制 (页 669)”章节。

使用案例 1：终止进给轴/主轴

终止轴时的特性与使用“删除剩余行程”功能相似。

DB21, ... DBX6.2 = 1（删除剩余行程）

使用案例 2：停止进给轴/主轴

停止下列由主处理控制的轴运动：

- PLC 轴
- 异步往复轴
- 指令轴（通过静态同步动作）
- 叠加运动：\$AA_OFF、DRF 手轮运行、在线刀具补偿和外部零点偏移

进给轴/主轴的从动轴运动不会被停止。

顺序说明：

10.3 运动特性和插补过程

- PLC → NCK: 请求停止进给轴/主轴
DB31, ... DBX28.6 = 1 (沿制动斜坡停止)
- NCK: 沿斜坡制动轴。
- NCK 确认执行:
 - DB31, ... DBX60.6 = 1 (粗准停)
 - DB31, ... DBX60.7 = 1 (精准停)
 - DB31, ... DBX63.2 = 0 (轴停止生效)
 - DB31, ... DBX64.6 / 7 = 0 (运行指令 -/+)
 - 轴状态: 系统变量 \$AA_SINGLAX_STAT == 3 (中断)。

结果: 进给轴/主轴停止。

说明

从动轴运动

从动轴的运动只能通过停止主动轴来中断。

回退运动

通过“扩展停止和退回”功能触发的回退运动无法停止。

文档

功能手册之特殊功能手册, 扩展停止和退回 (R3)

使用案例 3: 继续运行进给轴/主轴

需要继续执行被使用案例 2“停止轴”中断的运行。

顺序说明:

- PLC → NCK: 继续运行轴
DB31, ... DBX28.2 = 1 (继续)
- NCK: 检查是否存在带“CANCELCLEAR”或“NCSTARTCLEAR”清除条件的轴报警。
如存在, 则清除此报警。
- NCK: 检查轴是否能够继续运行? 如何是, 轴将转换至“生效”状态。
- NCK 确认执行:
 - DB31, ... DBX60.6 = 0 (粗准停)
 - DB31, ... DBX60.7 = 0 (精准停)
 - DB31, ... DBX63.2 = 0 (轴停止生效)
 - DB31, ... DBX64.6 / 7 = 1 (运行指令 -/+)
 - 轴状态: 系统变量 \$AA_SINGLAX_STAT == 4 (生效)。

结果：进给轴/主轴运行继续。

边界条件

以下情形下，继续运行请求会被忽略：

- 进给轴/主轴不再由 PLC 控制。
- 进给轴/主轴不再处于停止状态。
- 存在针对进给轴/主轴的报警。

使用案例 4：复位进给轴/主轴（复位）

顺序说明：

- PLC → NCK：请求复位进给轴/主轴
DB31, ... DBX28.1 = 1（复位）
- NCK：将进给轴/主轴转换至“停止”状态。
- NCK：终止停止的顺序运行，向 PLC 发送终止信息，与“删除剩余行程”类似。
- NCK：复位进给轴/主轴的内部状态。
- NCK：复位生效的轴专用机床数据生效。

说明

与 DB31, ... DBX28.1 = 1 触发的复位不同，采用通道复位时不会为 PLC 控制的轴激活机床数据。

- NCK 确认执行：
 - DB31, ... DBX63.0 = 1（复位已执行）
 - DB31, ... DBX63.2 = 0（轴停止生效）
 - 系统变量 \$AA_SINGLAX_STAT = 1
- NCK 结束此进程。

10.3.4 包含 NC 控制 ESR 的自控单轴运动

NC 控制的扩展停止

NC 控制的扩展停止和回退（ESR）功能也适用于单轴，且可通过轴专用机床数据配置：

单轴 ESR 的延时

MD37510 \$MA_AX_ESR_DELAY_TIME1

单轴插补制动的 ESR 时间

MD37511 \$MA_AX_ESR_DELAY_TIME2

不过仅当进给轴/主轴为单轴时，这些轴专用机床数据才生效。

NC 控制的扩展停止和回退通过轴触发器 \$AA_ESR_TRIGGER[轴] 触发。其功能与 \$AC_ESR_TRIGGER 相似，且只对单轴生效。

文档:

功能手册之特殊功能分册：章节“轴耦合和 ESR（M3）”

NC 控制的扩展回退

进行单轴回退时，必须通过 POLFA(轴, 类型, 值)赋值，同时须满足以下条件：

- 目标轴在触发时间点必须为单轴。
- \$AA_ESR_ENABLE[轴]=1
- POLFA(轴, 类型, 值) 仅用于类型 = 1 或类型 = 2 的情形
POLFA(轴, , 值, 轴, 类型, , 轴, 类型)

说明

NC 控制的单轴扩展停止：

仅当目标轴在触发时间点为单轴时触发才生效，否则触发会被忽略，并且不会为该轴执行停止指令。

NC 控制的单轴扩展回退：

NC 控制的通道扩展回退对单轴无效。在通道回退中，所有在触发时间点 \$AC_ESR_TRIGGER 为单轴的轴都会被忽略。

即便设置了所有回退参数（示例如下）也无法改变此情形。

MD37500 \$MA_ESR_REACTION

轴的 \$AA_ESR_ENABLE 等

示例

单轴的扩展停止：

MD37500 \$MA_ESR_REACTION[AX1]=22

MD37510 \$MA_AX_ESR_DELAY_TIME1[AX1]=0.3

MD37511 \$MA_AX_ESR_DELAY_TIME2[AX1]=0.06

```
...  
$AA_ESR_ENABLE[AX1] = 1  
$AA_ESR_TRIGGER[AX1]=1; 停止从此处开始生效  
单轴的扩展回退:  
MD37500 $MA_ESR_REACTION[AX1]=21  
...  
$AA_ESR_ENABLE[AX1] = 1  
POLFA(AX1, 1, 20.0); 为 AX1 指定绝对回退位置 20.0  
$AA_ESR_TRIGGER[AX1]=1; AX1 从此处开始回退  
POLFA(轴, 类型) ; 简单编程方式  
POLFA(轴, 0/1/2) ; 快速取消/激活
```

 **警告**

无预处理限制

若采用简单编程方式且只修改类型，应用中须确保回退位置或回退行程为有效值！
仅在特例情形下才应采用简单编程方式。
特别是：
在上电后必须重新设置回退行程或回退位置时
POLFA（轴，1，\$AA_POLFA[轴]）；会触发预处理停止
POLFA（轴，1）；不会触发预处理停止

10.4 定位轴动态响应

速度

定位轴采用为其定义的专用进给率运行。如“运动特性和插补过程 (页 646)”中所示，此进给率不受轨迹轴影响。

此进给率作为专用速度以毫米/分钟、英寸/分钟或度/分钟为单位编写。

此进给率通过轴名称固定指定给一根定位轴。

若编写定位轴时未设定进给率，那么控制系统会自动从以下轴专用机床数据取值：

MD32060 \$MA_POS_AX_VELO（定位轴速度的初始设置）

编写的进给率在程序结束后失效。

进给倍率

轨迹轴和定位轴使用单独的进给倍率。每根定位轴均可通过其单独的进给倍率进行调整。

快速移动倍率

快速移动倍率仅对轨迹轴生效。定位轴无快速移动插补（只有轴线性插补 G01），因此也无快速移动倍率。

旋转进给率

在 JOG 运行方式下，进给轴/主轴特性也取决于 SD41100 \$SN_JOG_REV_IS_ACTIVE（JOG 模式下旋转进给率生效）的设置：

- 该设定数据生效时，进给轴/主轴总是视主主轴的情况而定采用以下旋转进给率：
MD32050 \$MA_JOG_REV_VELO（JOG 旋转进给率）或 MD32040
\$MA_JOG_REV_VELO_RAPID（JOG 旋转进给率叠加快速移动速度）。
- 若该设定数据未生效，那么进给轴/主轴的进给率采用 SD43300
\$SA_ASSIG_FEED_PER_REV_SOURCE（定位轴/定位主轴的旋转进给率）。
- 若该设定数据未生效，而几何轴上有含旋转的有效框架命令，则该几何轴的特性取决于通道专用设定数据 SD42600 \$SC_JOG_FEED_PER_REV_SOURCE。（JOG 运行方式下，有含旋转的有效框架命令的几何轴的旋转进给率）。

最大单轴加速度

在定位轴运动中，根据设定的定位轴动态模式，下列最大值中的一个生效：

- MD32300 \$MA_MAX_AX_ACCEL [0]（动态模式 DYNNORM 中轨迹运动中的最大单轴加速度）
- MD32300 \$MA_MAX_AX_ACCEL [1]（动态模式 DYNPOS 中轨迹运动中的最大单轴加速度）

定位轴动态模式在以下 NC 专用机床数据中设置：

MD18960 \$MN_POS_DYN_MODE = <模式>

<模式>	含义
0	生效的最大单轴加速度: MD32300 \$MA_MAX_AX_ACCEL[0]
1	生效的最大单轴加速度: MD32300 \$MA_MAX_AX_ACCEL[1]

最大单轴加加速度

在加加速度限制生效的情况下运行定位轴时，下列机床数据中的一个作为最大加加速度生效：

- MD32430 \$MA_JOG_AND_POS_MAX_JERK（定位轴运动时的最大单轴加加速度）
- MD32431 \$MA_MAX_AX_JERK [0]（动态模式 DYNORM 中轨迹运动中的最大单轴加加速度）
- MD32431 \$MA_MAX_AX_JERK [1]（动态模式 DYNPOS 中轨迹运动中的最大单轴加加速度）

启用哪个机床数据取决于所设置的定位轴动态模式：

MD18960 \$MN_POS_DYN_MODE = <模式>

<模式>	含义
	定位轴运动时作为最大单轴加加速度生效：
0	MD32430 \$MA_JOG_AND_POS_MAX_JERK G75（运行至固定点）生效时：MD32431 \$MA_MAX_AX_JERK[0]
1	MD32431 \$MA_MAX_AX_JERK[1]

10.5 编程

10.5.1 简介

说明

定位轴的编程请参见以下文档：

文档：

编程手册之基本原理分册；章节“进给控制”和“主轴运动”

说明

一个程序段中可编写的定位轴的最大数量取决于存在的通道轴的最大数量。

定义

- 定位轴通过以下设定确定：
- 轴类型：定位轴类型 1、类型 2 或类型 3
 - 终点坐标（绝对坐标或增量坐标）
 - 进给率，直线轴以[毫米/分钟]为单位，回转轴以[度/分钟]为单位

示例：定位轴类型 1

程序代码	注释
POS[Q1]=200 FA[Q1]=1000	； 轴 Q1 以 1000 毫米/分钟的进给率运行至位置 200。

示例：定位轴类型 2

程序代码	注释
POSA[Q2]=300 FA[Q2]=1500	； 轴 Q2 以 1500 毫米/分钟的进给率运行至位置 300。

说明

原则上来讲，零件程序内的每根轴都可以是轨迹轴或定位轴。但是在一条运行程序段中必须为每根轴明确指定一种类型。

在同步动作中编程

也可通过同步动作完全异步于零件程序来进行轴的定位。

示例：

程序代码	注释
ID=1 WHENEVER \$R==1 DO POS[Q4]=10 FA[Q3]=990	； 轴进给率采用固定设定。

文档：

功能手册之同步动作分册

程序段切换

采用定位轴类型 1 和类型 2 时，可通过以下方式设置程序段切换：

FINEA=<轴名称> 或
FINEA [<轴名称>]

COARSEA=<轴名称> 或
COARSEA [<轴名称>]

IPOENDA=<轴名称> 或
IPOENDA [<轴名称>]

采用定位轴类型 3 时，还可通过以下指令设置在单插补制动斜坡内进行程序段切换：

IPOBRKA=<轴名称> 或
IPOBRKA (<轴名称> [, <时间点*>])

* 程序段切换时间点，是制动斜坡的 %

绝对坐标 / 增量坐标

终点可以绝对坐标（G90）或增量坐标（G91）编写。

示例	含义
	终点坐标编程：
G90 POS [Q1]=200	以绝对坐标
G91 POS [Q1]=AC (200)	以绝对坐标
G91 POS [Q1]=200	以增量坐标
G90 POS [Q1]=IC (200)	以增量坐标

重新编写类型 2 的定位轴

对于类型 2（越过程序段边界运动）的定位轴，必须能在零件程序中确定定位轴是否已到达终点位置。只有这样才能重新编程此定位轴（否则会触发报警）。

若先编写了 POSA，之后重新编写了 POSA 和 IPOBRKA（在制动斜坡中进行程序段切换），则不会触发报警。更多提示请参见“可设置的程序段切换时间点”章节中的语言指令 IPOBKA 部分。

协调（WAITP）

使用协调指令 WAITP 可以在 NC 程序中标记位置，并在此位置上等待，直到在前一 NC 程序段中用 POSA 编程的轴到达终点。

WAITP 需要单独占用一个程序段。

需要等待的轴必须明确设定。

示例：

程序代码	注释
N10 G01 G90 X200 F1000 POSA[Q1]=200 FA[Q1]=500	
N15 X400	
N20 WAITP(Q1)	； 程序处理自动停止，直至 Q1 到达位置。
N25 X600 POS[Q1]=300	； Q1 为定位轴类型 1（采用程序段 N10 中的进给率 FA[Q1]）。
N30 X800 Q1=500	； Q1 为轨迹轴（采用程序段 N10 中的轨迹进给率 F1000）。

刀具补偿

定位轴可通过零点偏移进行刀具长度补偿。这例如会使装刀器的定位行程发生变化。因此可将零点偏移代替刀具长度补偿用于以下情形：将装有不同尺寸刀具的装刀器需要越过一个障碍。

程序结束

程序结束（程序状态已选择）会一直延迟，直至所有轴（轨迹轴 + 定位轴）到达编写的终点。

10.5.2 外部编程时的旋转进给率

通过下面两个设定数据，可设定从另一根回转轴/主轴推导出定位轴的旋转进给率：

SD43300 ASSIGN_FEED_PER_REV_SOURCE（定位轴/主轴的旋转进给率）

SD42600 JOG_FEED_PER_REV_SOURCE（JOG 旋转进给率调整）

可以进行下列设置：

值	含义
0	不选择旋转进给率。
>0	旋转进给率从此处设定的机床轴下标对应的回转轴/主轴推导。
-1	旋转进给率从进给轴/主轴所在通道的主主轴推导。

值	含义
-2	旋转进给率从机床轴下标为 0 的回转轴/主轴推导。
-3	旋转进给率从进给轴/主轴所在通道的主主轴推导。主主轴停止时不选择旋转进给率。

10.6 程序段切换

轨迹轴和定位轴相互独立地进行插补，因此它们会在不同的时间点到达编写的终点位置。若将轨迹轴和定位轴编写在同一程序段中，那么程序段切换特性取决于可编写的定位轴类型。

类型 1：程序段相关定位轴

特性：

- 所有轨迹轴和定位轴符合编写的运动结束条件后，立即进行程序段切换：
 - 轨迹轴：G601、G602、G603
 - 定位轴：FINEA、COARSA、IPOENDA
- 定位轴的编程：POS[<轴>]

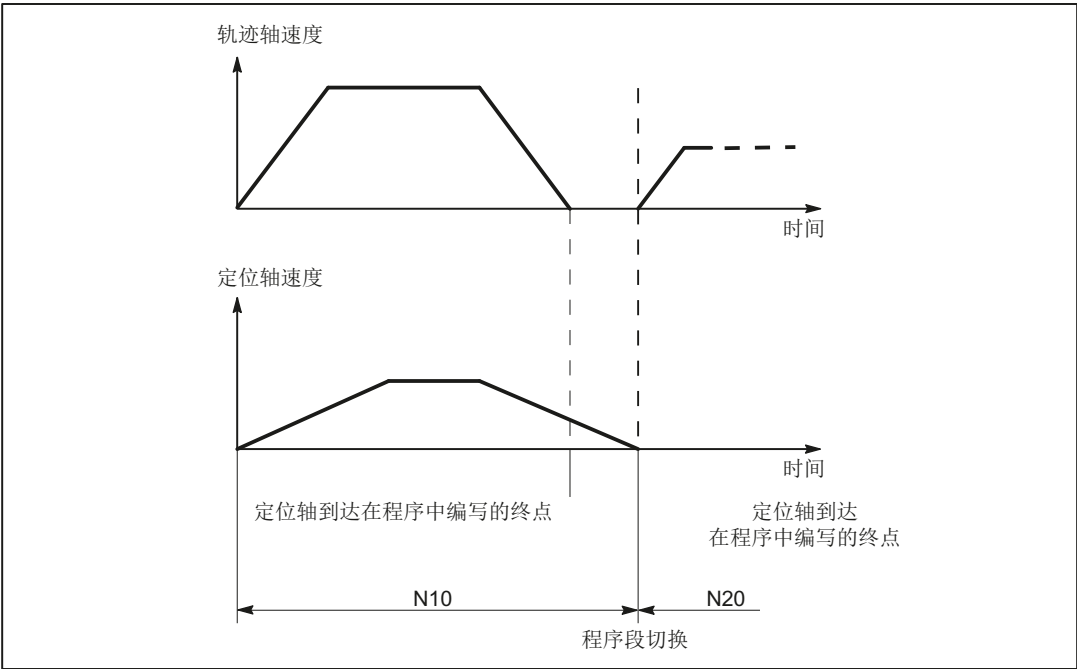


图 10-1 采用轨迹轴和定位轴类型 1 时的程序段切换

说明

连续路径运行

仅当定位轴先于轨迹轴满足运动结束条件时（上图不符合此条件），才能进行跨程序段的连续路径运行（G64）。

类型 2：跨程序段定位轴

特性：

- 所有轨迹轴符合编写的运动结束条件（G601、G602、G603）后，立即进行程序段切换。
- 定位轴的编程：POSA[<轴>]
- 定位轴跨程序段运行至编写的终点位置。满足运动结束条件前不允许重新编程定位轴。

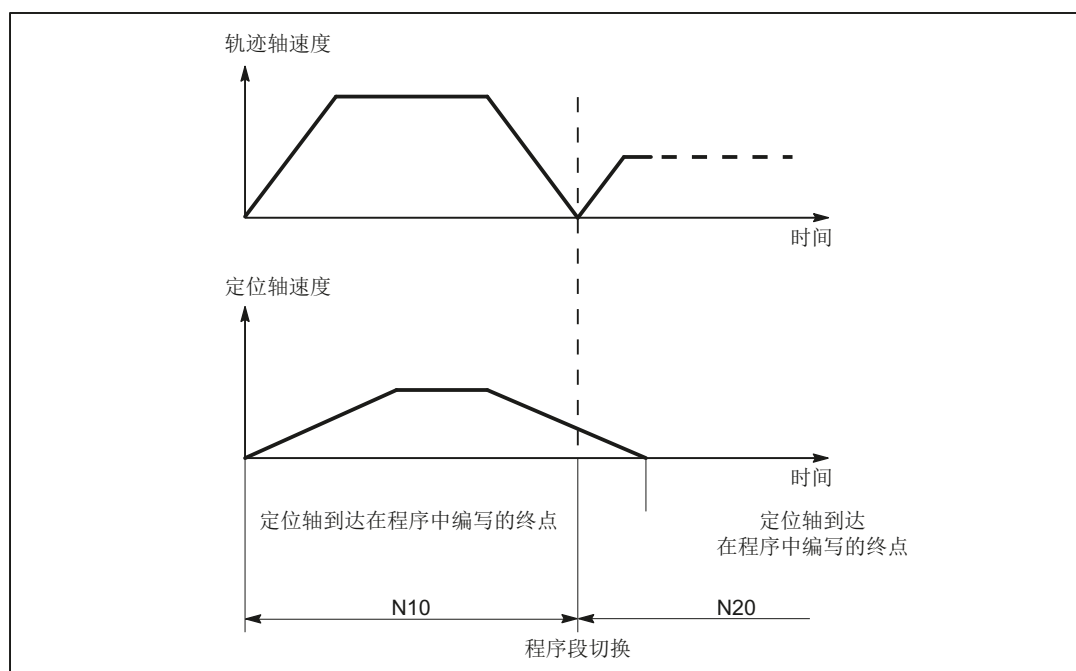


图 10-2 采用轨迹轴和定位轴类型 2 时的程序段切换

10.6.1 可设置的程序段切换时间点

类型 3: 与程序段有条件关联的定位轴

特性:

- 所有轨迹轴和定位轴符合编写的运动结束条件后，立即进行程序段切换：
 - 轨迹轴：G601、G602、G603
 - 定位轴：IPOBRKA
- 定位轴的编程：
 - N(x) IPOBRK(<轴>[, <时间点>]) ; 独立程序段
 - N(x+1) POS [<轴>]

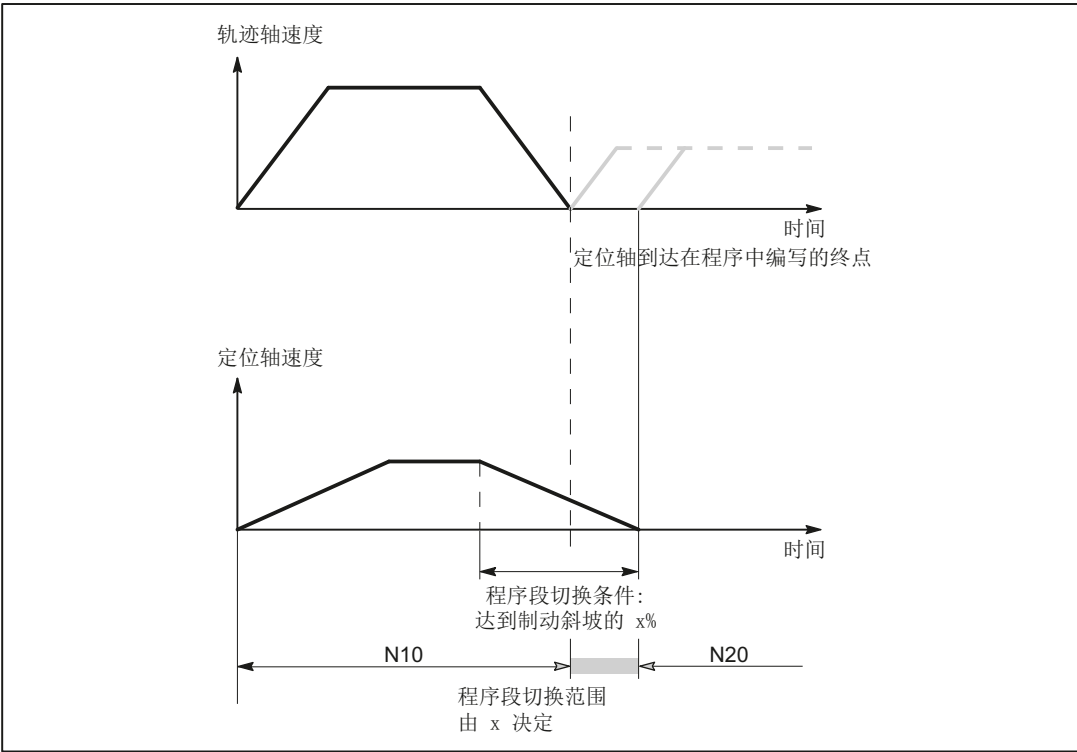


图 10-3 采用轨迹轴和定位轴类型 3 时的程序段切换

程序段切换条件：“制动斜坡”（IPOBRKA）

如果激活了程序段切换条件“制动斜坡”，并对可选参数 <时间点> 进行了赋值，那么该值会在下一次定位运动中生效，并和主处理同步地写入设定数据中。没有指定该时间时，设定数据中的当前值生效。

SD43600 \$SA_IPOBRAKE_BLOCK_EXCHANGE

可进行程序段切换的时间点以制动斜坡百分比的形式设定：

- 100% = 制动斜坡开始
- 0% = 制动斜坡结束，与程序段切换条件 IPOENDA 含义相同

编程

IPOBRKA (<轴> [, <时间点>])

IPOBRKA: 程序段切换条件：制动斜坡
生效方式: 模态有效
<轴>: 通道轴名称 (X, Y, ...)

<时间点>: 程序段转换时间, 是制动斜坡的 % 值:

- 100% = 制动斜坡开始
- 0% = 制动斜坡结束, 和 IPOENDA 含义相同

类型: REAL

重新编写轴运动结束条件 (FINEA、COARSEA、IPOENDA) 时, 对应轴的 IPOBRKA 会被取消。

附加程序段切换条件: “公差带” (ADISPOSA)

可将一个以程序段终点 (可选择实际位置或设定位置) 为中心的公差窗口设为附加程序段切换条件。这样一来程序段切换须满足两个条件:

- 程序段切换条件: “制动斜坡”
- 程序段切换条件: “公差窗口”

编程

ADISPOSA (<轴>[, <模式>, <窗口大小>])

ADISPOSA: 用作运动结束条件的公差窗口

生效方式: 模态有效

<轴>: 通道轴名称 (X, Y, ...)

<模式>: 公差窗口的中心

类型: INT

取值范围:

0	公差窗口不生效
1	公差窗口以设定位置为中心
2	公差窗口以实际位置为中心

<窗口大小>: 公差窗口的大小

类型: REAL

运动结束条件的系统变量

可通过系统变量 \$AA_MOTEND 读取当前生效的运动结束条件。

文档: 参数手册 “手册 2”

说明

可编程运动结束条件 FINEA、COARESA、IPOENDA 的更多相关信息请见:

文档: 功能手册之基本功能分册

- 主轴 (S1), “主轴运行方式”
 - 进给率 (V1), “可编程的单轴动态响应”
-

边界条件

提前程序段切换

在以下情形下无法提前进行程序段切换:

- 往复轴
在带单次进给的往复运动中, 往复运动必须保持生效, 直至轴通过单次进给到达终点位置。
- 手轮
在手轮运行中, 最近一次设置的运动结束条件生效。

轴状态切换

如果轴在制动斜坡内进行了程序段切换, 那么下一个程序段中只允许在相同的轴状态下重新编写该轴。

轴状态发生切换时, 例如从 POS 切换至 SPOS 时, 最近一次编写的运动结束条件 FINEA、COARSEA、IPOENDA 将生效。这也适用于以下情形:

- 定位轴切换为轨迹轴
- 等待定位运动结束时: WAITP、M30、工艺循环技术、预处理停止
- 激活或取消速度叠加时

说明

定位轴的更多相关信息请见:

文档:

编程手册之基本原理分册, “进给控制和主轴运动” 章节

编程手册之工作准备分册, “特殊行程指令” 章节

示例

零件程序中采用程序段切换条件“制动斜坡”

程序代码	注释
	; 缺省设置生效。
N10 POS[X]=100	; X 轴定位到位置 100。
	; 程序段切换条件: “精准停”
N20 IPOBRKA(X,100)	; 程序段切换条件: “制动斜坡”, 100% = 制动斜坡开始处
N30 POS[X]=200	; 一旦 X 轴开始制动, 就立即切换程序段。
N40 POS[X]=250	; X 轴不再制动至位置 200, 而是继续运行至 250。一旦 X 轴开始制动就立即切换程序段。
N50 POS[X]=0	; X 轴制动并返回到位置 0, 在位置 0 和“精准停”条件下切换程序段。
N60 X10 F100	; X 轴作为轨迹轴运动到位置 10。
N70 M30	

同步动作中采用程序段切换条件“制动斜坡”

在工艺循环中:

程序代码	注释
FINEA	; 运动结束条件: “精准停”
N10 POS[X]=100	; 当 X 轴到达位置 100 和精准停窗口时, 进行工艺循环程序段切换。
N20 IPOBRKA(X,100)	; 激活程序段切换条件“制动斜坡”, 100% = 制动斜坡开始处
N30 POS[X]=200	; 一旦 X 轴开始制动, 就立即进行工艺循环程序段转换。
N40 POS[X]=250	; X 轴不再制动至位置 200, 而是继续运行至 250。一旦 X 轴开始制动就立即进行工艺循环中的程序段切换。
N50 POS[X]=0	; X 轴制动并返回到位置 0, 在位置 0 和“精准停”条件下切换程序段。
N60 M17	

零件程序中采用程序段切换条件“制动斜坡”和“公差窗口”

程序代码	注释
	; 缺省设置生效。
N10 POS[X]=100	; X 轴定位到位置 100。
	; 程序段切换条件: “精准停”
N20 IPOBRKA(X,100)	; 程序段切换条件: “制动斜坡”, 100% = 制动斜坡开始处
N21 ADISPOSA(X,1,0.5)	; 公差窗口: 1 = 设定位置, 公差 = 0.5
N30 POS[X]=200	; 一旦 X 轴开始制动, 就立即切换程序段。
N40 POS[X]=250	; X 轴不再制动至位置 200, 而是继续运行至 250。一旦 X 轴开始制动就立即切换程序段。

10.6 程序段切换

程序代码	注释
N50 POS[X]=0	; X 轴制动并返回到位置 0， 在位置 0 和“精准停”条件下切换程序段。
N60 X10 F100	
N70 M30	

同步动作中采用程序段切换条件“制动斜坡”和“公差窗口”

在工艺循环中：

程序代码	注释
FINEA	; 运动结束条件：“精准停”
N10 POS[X]=100	; 当 X 轴到达位置 100 和精准停窗口时，进行工艺循环程序段切换。
N20 IPOBRKA(X,100)	; 激活程序段切换条件“制动斜坡”，100% = 制动斜坡开始处
N21 ADISPOSA(X,2,0.3)	; 公差窗口：2 = 实际位置，公差 = 0.3
N30 POS[X]=200	; 一旦 X 轴开始制动以及 X 轴的实际位置 >= 199.7，立即进行工艺循环程序段切换。
N40 POS[X]=250	; X 轴不再制动至位置 200，而是继续运行至 250。一旦 X 轴开始制动以及 X 轴的 >= 249.7，立即进行工艺循环程序段切换。
N50 POS[X]=0	; X 轴制动并返回到位置 0， 在位置 0 和“精准停”条件下切换程序段。
N60 M17	

若以下设定数据中已输入了相应值，那么在程序段 N20 中可不设定时间写入 IPOBRKA (X)。

```
SD43600 $SA_IPOBRAKE_BLOCK_EXCHANGE[X] == 100
```

参见

通过 PLC 控制 (页 669)

10.6.2 程序段搜索时的运动结束条件

最后一个程序段用作容器

最后一次编写的轴运动结束条件会被记录，并在一个动作程序段中输出。搜索中处理的最后一个包含运动结束条件的程序段用作容器，在其中可设置针对所有通道编写的所有运行结束条件。

示例

两个动作程序段，包含针对三根轴的运动结束条件：

程序代码	注释
N01 G01 POS[X]=20 POS[Y]=30	
IPOENDA[X]	; 最后一次编写的运动结束条件 IPOENDA
N02 IPOBRKA(Y, 50)	; 第二个动作程序段，用于 Y 轴 IPOENDA
N03 POS[Z]=55 FINEA[Z]	; 第二个动作程序段，用于 Z 轴 FINEA
N04 \$A_OUT[1]=1	; 第一个动作程序段，用于作为数字量输出
N05 POS[X]=100	;
N06 IPOBRKA(X, 100)	; 第二个动作程序段，用于 X 轴 IPOBRKA
...	; 并作为编写的所有运动结束条件的容器
TARGET:	; 程序段搜索目标

在第一个动作程序段中输出数字量输出：

\$A_OUT[1]=1。

在第二个动作程序段中设置运动结束条件：

X 轴：IPOBRKA / \$SA_IPOBRAKE_BLOCK_EXCHANGE[AX1]=100

Y 轴：IPOBRKA / \$SA_IPOBRAKE_BLOCK_EXCHANGE[AX2]=50

Z 轴：FINEA。此外为 X 轴标记最后一次编写的运动结束条件 IPOENDA。

10.7 通过 PLC 控制

PLC 轴

PLC 轴由 PLC 运行，可与所有其他轴异步运动。其运行不受轨迹运动和同步动作的影响。

文档：

功能手册之基本功能分册；SINUMERIK 840D sl 的 PLC 基本程序（P3）或 SINUMERIK 828D 的 PLC（P4）

并行定位轴

在 SINUMERIK 840D sl 上，并行定位轴通过 PLC 的 FC18 启动。

通道专用信号

所有通道专用信号对轨迹轴和定位轴都有效。

但以下信号除外：

- IS DB21, ... DBB4 (“进给倍率”)
- IS DB21, ... DBX6.2 (“删除剩余行程”)

轴专用信号

定位轴还有以下信号：

- IS DB31, ... DBX76.5 (“定位轴”)
- 定位轴/主轴的进给率 (FA)
- IS DB31, ... DBB0 (“进给倍率”)，轴专用
- IS DB31, ... DBX2.2 (“剩余行程/主轴复位”)，删除单轴剩余行程

PLC 控制的轴的单轴功能

单个 PLC 轴的特性可通过以下机床数据
MD30460 \$MA_BASE_FUNCTION_MASK
如下修改：

- 位 4 = 1
该轴只由 PLC 控制。
- 位 5 = 1
该轴是固定指定给 PLC 的轴。
- 位 6 = 1
通道专用 NC/PLC 接口信号：
DB21, ... DBX6.0 (“进给禁用”)
不对由 PLC 控制的轴生效。
- 位 7 = 1
通道专用 NC/PLC 接口信号：
DB21, ... DBX36.3 (“所有轴停止”)
置位时**不考虑**由 PLC 控制的轴。

对于由 PLC 控制的轴：

- 机床数据 MD30460 \$MA_BASE_FUNCTION_MASK 位 6 = 0 时，通道专用 NC/PLC 接口信号 DB21, ... DBX6.0 (“进给禁用”)生效。
- 仅当机床数据 MD30460 \$MA_BASE_FUNCTION_MASK 位 7 = 0 时，通道专用 NC/PLC 接口信号 DB21, ... DBX36.3 (“所有轴停止”)才置位。

若尝试将只由 PLC 控制的轴指定给 NC 程序或为 NC 程序请求该轴，那么操作会被拒绝并触发报警 26075。与之类似，对固定指定给 PLC 的轴进行这些操作会触发报警 26076。

固定指定给 PLC 的轴启动时转换为“中立轴”。通过 NC/PLC 接口发出运行请求时，若此前未取轴，那么并行定位轴将自动转换成 PLC 轴。

通过 PLC 取轴

通过轴接口信号 NCK→PLC DB31, ... DBB68 将待取轴的类型信息提供给 PLC（另见“K5: 运行方式组、通道、跨通道取轴 (页 323)”章节）：

- 接口信号 DB31, ... DBX68.0-68.3 (“通道 NC 进给轴/主轴”)通道 1 至 10
- 接口信号 DB31, ... DBX68.4 (“PLC 请求的新类型”)
- 接口信号 DB31, ... DBX68.5 (“可跨通道取轴”)
- 接口信号 DB31, ... DBX68.6 (“中立轴/主轴”)
- 接口信号 DB31, ... DBX68.7 (“PLC 进给轴/主轴”)

若接口信号 DB31, ... DBX68.5 (“可跨通道取轴”) = 1，则可通过 PLC 发出取轴请求。

10.7.1 由 PLC 启动并行定位轴

由 PLC 触发

若需由 PLC 触发并行定位轴，须调用 FC 18 并提供以下参数：

- 轴名称或轴编号
- 终点位置
- 进给率（进给率=0 时将采用机床数据 MD32060 \$MA_POS_AX_VELO 中的进给率）。
- 绝对坐标(G90)、
相对坐标(G91)、
回转轴以最短行程绝对定位：回转轴名 = DC(值)

10.7 通过 PLC 控制

下列功能采用固定设定：

- 线性插补（G01）
- 进给率，单位：毫米/分钟或度/分钟（G94）
- 准停（G09）
- 当前选择的可设定零点偏移生效

每根轴只能被指定给一个通道，因此控制系统可通过轴名称和轴编号选择正确的通道，并在此通道中启动并行定位轴。

10.7.2 由 PLC 控制的轴

PLC 动作

下面以机床轴 1 为例说明 PLC 动作和 NCK 响应：

- 通过 FC 18 将机床轴作为 PLC 轴启动
- 触发 NC 启动或 NC 停止
- 触发轴停止、继续或复位
- 触发 NC 复位
- 为机床轴取消或设置伺服使能
- 将机床轴的控制权交换给 NC

NCK 响应示例

下表列出了 PLC 动作和与之对应的 NCK 响应。

PLC 动作	NCK 响应
通过 FC 18 启动设为 PLC 轴的机床轴 AX1 (处于第 1 通道)	
触发进给轴和主轴的 NC 停止 DB21, ... DBX7.4 = 1 (进给轴和主轴 NC 停止)	AX1 停止。
触发 DB21, ... DBX7.1 = 1 (NC 启动)	AX1 继续运行。
PLC 请求控制 AX1, DB31, ... DBX28.7 = 1 (PLC 控制轴)	将 AX1 的控制权转交给 PLC。DB31, ... DBX63.1 = 1 (PLC 控制轴)

PLC 动作	NCK 响应
触发进给轴和主轴的 NC 停止 DB21, ... DBX7.4 = 1 (进给轴和主轴 NC 停止)	AX1 不停止。
触发轴停止 DB31, ... DBX28.6 = 1 (沿制动斜坡停止)	AX1 停止 DB31, ... DBX63.2 == 1 (轴停止生效)
触发轴继续运行 DB31, ... DBX28.2 = 1 (继续)	AX1 继续运行 DB31, ... DBX63.2 == 0 (轴停止生效)
触发 NC 复位 触发 DB21, ... DBX7.7 = 1 (复位)	对 AX1 无影响
触发轴复位 DB31, ... DBX28.1 = 1 (复位)	AX1 停止, 运行终止: • DB31, ... DBX63.2 = 0 (轴停止生效) • 读取轴专用机床数据 • DB31, ... DBX63.0 = 1 (复位已执行) • DB31, ... DBX63.2 = 0 (轴停止生效)
通过 FC 18 启动设为 PLC 轴的机床轴 AX1	DB31, ... DBX63.0 = 0 (复位已执行)
取消 AX1 的伺服使能: DB31, ... DBX2.1 = 0 (伺服使能)	显示报警 21612“轴 %1 测量系统切换”
触发轴继续运行 DB31, ... DBX28.2 = 1 (继续)	• 清除报警 21612“轴 %1 测量系统切换” • DB21, ... DBX40.7 = 1 (运行指令 +) • 由于缺少伺服使能, AX1 不运行。
设置 AX1 的伺服使能: DB31, ... DBX2.1 = 1 (伺服使能)	AX1 运行至编写的终点。
触发轴复位 DB31, ... DBX28.1 = 1 (复位)	• AX1 停止 • 读取轴专用机床数据 • DB31, ... DBX63.0 = 0 (复位)
从 DB31, ... DBX28.7 = 0 (PLC 控制轴) 起 PLC 将 AX1 的控制权转交给 NCK	• NCK 接收机床轴的控制权 • DB31, ... DBX63.1 = 0 (PLC 控制轴) • DB31, ... DBX63.0 = 0 (复位)

10.7.3 PLC 轴的控制特性

通道复位、NEWCONFIG、程序段搜索和 MD30460 时的特性

表格 10-1

以下情形下的控制特性	由 PLC 控制的轴
运行方式切换和 NC 程序控制	此时不受影响。
通道复位	轴专用机床数据不生效，且运行不会终止。
NEWCONFIG	轴专用机床数据不生效。
程序段搜索 SERUPRO，类型 5	在 SERUPRO 期间模仿正常流程处理，例如 PLC 接收或转交 PLC 轴的控制权。
所有程序段搜索方案 类型 1、2、4 和 5	PLC 在逼近程序段前接收轴的控制权，并负责该轴的定位。
通过 \$AC_ESR_TRIGGER 激活的 NC 控制的回退	无作用，且只对指定的 PLC 轴生效。
机床数据： MD30460 \$MA_BASE_FUNKTION_MASK	不局限于 PLC 轴
位 4 = 0	通过取轴 GET(轴) 或 AXTOCHAN(轴, 通道)指令无法将该轴直接切换为由 NC 程序控制的轴，参见* 取轴提示。
位 4 = 1	无法为 NC 程序请求该轴。通过 NC 程序或同步动作执行的 GET 或 AXTOCHAN 以及在 NC 程序中编写该轴均会被系统拒绝，并发出报警 26075。
MD30460 \$MA_BASE_FUNKTION_MASK	对于由 PLC 控制的轴
位 6 = 1	通道专用接口信号 IS DB 21, ... DBX6.0 (“进给禁用”) 不生效。激活进给禁用时该轴不停止，而是继续运行。
位 7 = 1	生成接口信号 IS DB 21, ... DBX36.3 (“所有轴停止”) 时不考虑该轴。通道的所有其他轴停止，只有由 PLC 控制的轴生效时，此信号仍输出 1。

* 取轴提示

仅当 PLC 按照使用案例“交还控制权”实际交还了控制权时，才能实现从“中立轴”到“NC 程序轴”转换。此时取轴等待会通过操作面板前端 HMI 显示。

10.8 特殊功能下的特性

10.8.1 空运行进给率（DRY RUN）

编写的进给率不大于空运行进给率的情况下，空运行进给率也对定位轴生效。

SD42100 \$SA_DRY_RUN_FEED 中输入的空运行进给率的有效性可通过 SD42101 \$SA_DRY_RUN_FEED_MODE 控制。参见

文档：

功能手册之基本功能分册；章节“进给率(V1)”

10.8.2 单程序段

定位轴类型 1

采用定位轴类型 1 时，单程序段方式同样生效。

定位轴类型 2

定位轴类型 2 在单程序段方式下仍会越过程序段边界。

定位轴类型 3

定位轴类型 3 在单程序段方式下仍会越过程序段边界。

10.9 示例

10.9.1 运动特性和插补过程

在此示例中，定位轴 Q1 和 Q2 分别代表两个独立的运动单元。两者不进行插补。此示例中定位轴将作为类型 1（例如 N20 中）和类型 2（例如 N40 中）编程。

程序示例

程序代码

```
N10 G90 G01 G40 T0 D0 M3 S1000
N20 X100 F1000 POS[Q1]=200 POS[Q2]=50 FA[Q1]=500
FA[Q2]=2000
N30 POS[Q2]=80
N40 X200 POSA[Q1] = 300 POSA[Q2]=200] FA[Q1]=1500
N45 WAITP[Q2]
N50 X300 POSA[Q2]=300
N55 WAITP[Q1]
N60 POS[Q1]=350
N70 X400
N75 WAITP[Q1, Q2]
N80 G91 X100 POS[Q1]=150 POS[Q2]=80
N90 M30
```

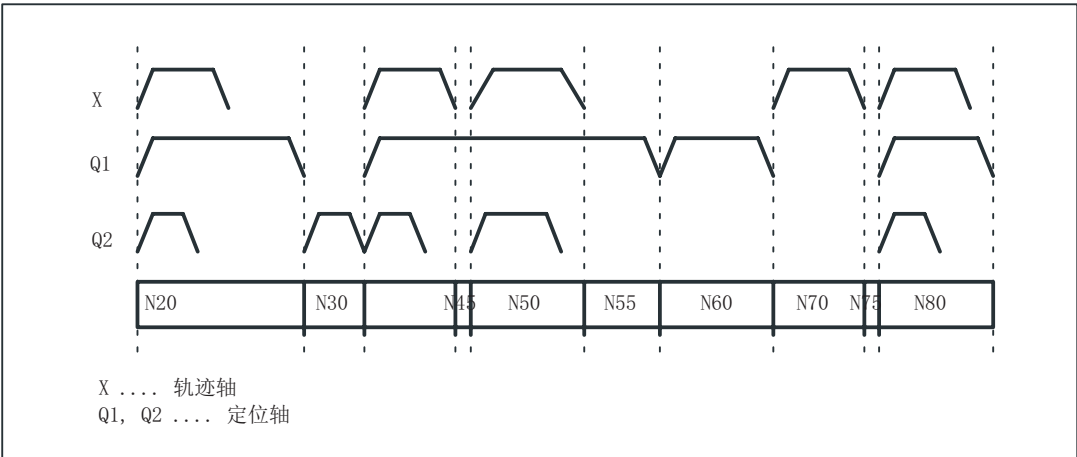


图 10-4 轨迹轴和定位轴的时序图

10.9.1.1 G0 下轨迹轴无插补运行

G0 下针对定位轴的示例

快速移动（G0）中轨迹轴作为无插补的定位轴运行：

程序代码	注释
	; 激活非线性
	; 插补
	; MD20730 \$MC_GO_LINEAR_MODE == FALSE
	; 已设置
G0 X0 Y10	; 无插补运行
G0 G43 X20 Y20	; 在轨迹模式中（插补）
	; 运行
G0 G64 X30 Y30	; 在轨迹模式中（插补）
	; 运行
G0 G95 X100 Z100 m3 s100	; 无插补运行
	; 无旋转进给率生效

10.10 数据表

10.10.1 机床数据

10.10.1.1 NC 专用机床数据

编号	名称: \$MN_	描述
18960	POS_DYN_MODE	定位轴动态模式

10.10.1.2 通道专用机床数据

编号	名称: \$MC_	描述
20730	G0_LINEAR_MODE	G0 的插补特性
20732	EXTERN_G0_LINEAR_MODE	G00 的插补特性
22240	AUXFU_F_SYNC_TYPE	F 功能的输出时间

10.10 数据表

10.10.1.3 进给轴/主轴专用机床数据

编号	名称: \$MA_	描述
30450	IS_CONCURRENT_POS_AX	并行定位轴
30460	BASE_FUNCTION_MASK	轴功能
32060	POS_AX_VELO	定位轴的进给率
32300	MAX_AX_ACCEL	最大轴加速度
32430	JOG_AND_POS_MAX_JERK	定位轴运动中的单轴加加速度上限
32431	MAX_AX_JERK	轨迹运动中的单轴加加速度上限
37510	AX_ESR_DELAY_TIME1	ESR 单轴延时
37511	AX_ESR_DELAY_TIME2	单轴插补制动的 ESR 时间

10.10.2 设定数据

10.10.2.1 进给轴/主轴专用设定数据

编号	名称: \$SA_	描述
43600	IPOBRAKE_BLOCK_EXCHANGE	程序段切换条件“制动斜坡”
43610	ADISPOSA_VALUE	制动斜坡公差窗口

10.10.3 信号

10.10.3.1 发送至通道的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
进给禁用	DB21, ...DBX6.0	DB3200.DBX6.0
NC 启动	DB21, ...DBX7.1	DB3200.DBX7.1
NC 停止进给轴和主轴	DB21, ...DBX7.4	DB3200.DBX7.4
复位	DB21, ...DBX7.7	-

10.10.3.2 来自通道的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
所有轴停止	DB21, ...DBX36.3	DB3300.DBX4.3
运行指令 -	DB21, ...DBX40.6	DB3300.DBX1000.6
运行指令 +	DB21, ...DBX40.7	DB3300.DBX1000.7

10.10.3.3 到进给轴/主轴的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
单轴进给倍率	DB31, ...DBB0	DB380x.DBB0
伺服使能	DB31, ...DBX2.1	DB380x.DBX2.1
删除单通道剩余行程/主轴复位	DB31, ...DBX2.2	DB380x.DBX2.2
复位	DB31, ...DBX28.1	-
继续	DB31, ...DBX28.2	-
沿制动斜坡停止	DB31, ...DBX28.6	-
PLC 控制轴	DB31, ...DBX28.7	-

10.10.3.4 来自进给轴/主轴的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
粗准停	DB31, ...DBX60.6	DB390x.DBX0.6
精准停	DB31, ...DBX60.7	DB390x.DBX0.7
轴报警	DB31, ...DBX61.1	DB390x.DBX1.1
轴运行就绪 (AX_IS_READY)	DB31, ...DBX61.2	DB390x.DBX1.2
轴容器旋转生效	DB31, ...DBX62.7	-
AXRESET DONE	DB31, ...DBX63.0	-
PLC 控制轴	DB31, ...DBX63.1	-
进给轴停止有效	DB31, ...DBX63.2	DB390x.DBX3.2
运行指令 -	DB31, ...DBX64.6	DB390x.DBX4.6
运行指令 +	DB31, ...DBX64.7	DB390x.DBX4.7
定位轴	DB31, ...DBX76.5	DB390x.DBX1002.5

10.10 数据表

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
定位轴的 F 功能（进给率）	DB31, ...DBB78-81	-
紧急回退生效	DB31, ...DBX98.7	DB390x.DBX5002.7

P5: 往复运动 - 仅限 840D sl

11.1 简要说明

定义

在“往复运动”功能中，一根轴以程序写入的进给率或者计算出的进给率（旋转进给率）在两点之间来回移动，该轴称为“往复轴”，这两点称为“换向点”。可以同时有多根轴一起作往复运动。

往复运动的类型

往复运动可按换向条件和进给运动加以区分：

- 模态式异步往复运动
在往复运动期间，其他轴可以任意插补。往复轴可以作为动态坐标转换的输入轴或龙门轴/主-从轴上的主动轴。往复运动不自动进入“自动”方式。
- 伴随有连续进给的往复运动
可以同时有多根轴一起作进给运动。但是进给运动和往复运动之间不存在插补上的关联。
- 往复运动可伴随两个换向点上或某个换向点上的进给运动。进给运动也可从和换向点相距程序指定间距的位置点上提前开始。
- 在往复运动后进行无火花光磨（简称光磨）。
- 往复运动可在指定位置上开始和结束。

换向点上的运动方式

换向的触发条件有：

- 未进入准停公差带（精准停或粗准停）
- 到达了程序写入的位置或者
- 到达了程序写入的位置且暂停时间结束。
- 由 PLC 给出的外部信号触发。

11.2 异步往复运动

影响

有多种方式可控制往复运动：

- 通过删除剩余行程中断往复运动和/或进给运动。
- 通过数控程序、PLC、HMI、手轮或移动键修改换向点。
- 通过数控程序、PLC、HMI 或倍率修改往复轴的进给率。该进给率可和主主轴、回转轴或主轴相关（旋转进给率）。

文档：

功能手册之基本功能分册；章节“进给率(V1)”

- 往复运动可完全由 PLC 控制。

往复运动的控制方式

往复运动可分为：

1. 异步往复运动：
模态生效，也可由 PLC/HMI 启动。
2. 通过同步动作实现同步：
异步往复运动和进给运动可通过一个同步动作实现同步，因此在往复运动中可伴随换向点上的进给运动，该进给运动非模态生效。

11.2 异步往复运动

属性

异步往复运动有以下几个特点：

- 往复轴以指定的进给率在换向点 1 和 2 之间来回移动，直到取消了往复运动，或者某项监控作出响应。如果一开始往复轴不位于换向点 1 上，该轴首先移动到该点。
- 对于往复轴而言，线性插补 G01 始终有效，不管程序中当前写入的 G 代码是什么。此外，也可以激活旋转进给率 G95。
- 异步往复运动是特定轴的运动，为模态生效。
- 同时可以有多根往复轴，数量是最大定位轴数量。
- 在往复运动期间，其他轴可以任意插补。通过一个轨迹运动或者用一个定位轴可以达到一个连续进给。但是进给运动和往复运动之间不存在插补上的关联。

- PLC 不对轴进行控制时，在异步往复运动中，NC 便如同处理普通的定位轴一样处理往复轴。PLC 进行控制时，PLC 程序可通过 NC/PLC 接口信号中的对应停止位使往复轴按所需方式动作。此类 NC/PLC 接口信号中也包含了：程序结束信号、运行方式切换信号和单步信号。
- 往复轴可以设为坐标转换中的输入轴（比如：“倾斜轴”功能，见章节“M1: 坐标转换 (页 367)”）。
- 往复轴可以设为龙门轴和主-从轴上的主动轴。
文档：
功能手册之特殊功能分册，章节“龙门轴 (G1)”
- 如同定位轴一样，可为往复轴激活急动限制(SOFT)和/或折线式加速曲线。
- 通过零件程序可以确保往复运动和程序段同步启动。
- 往复运动也可以由 PLC/HMI 启动、控制和停止。
- 不能实现插补式往复运动（比如：“倾斜轴”）。

11.2.1 异步往复运动的设置选项

设定数据

往复需运动需要使用的设定数据可以由 NCK 中专门的语言指令、HMI 或 PLC 设置。

进给速度

往复轴进给率的选择或指定方式有：

- 使用为定位轴指定的进给率。该速度可通过指令 FA[轴名] 写入程序，为模态生效。如果在程序中没有写入该速度，系统会使用机床数据 POS_AX_VELO 的赋值（另见章节“P2: 定位轴 (页 639)”）。
- 在往复运动进行期间通过设定数据加以修改。此时可通过零件程序或设定数据确定，修改是立即生效还是从下一个换向点起才开始生效。
- 通过倍率（轴专用 NC/PLC 接口信号或程序写入的倍率值）加以修改。
- 试运行期间采用试运行进给率，如果该值大于当前程序写入的进给率。
SD42100 \$SC_DRY_RUN_FEED 中输入的试运行进给率的生效方式可由 SD42101 \$SC_DRY_RUN_FEED_MODE 设置。

文档：

功能手册之基本功能分册；章节“进给率(V1)”

11.2 异步往复运动

- 速度/位置可由手轮加以控制，见章节“H1: 手动运行和手轮运行 (页 155)”中的表格。
- 往复轴也可以采用旋转进给率。

旋转进给率

旋转进给率也可以用于往复轴。

换向点

换向点的坐标既可以在往复运动激活前通过设定数据加以修改，也可以在往复运动激活后加以修改。

- 在往复运动进行期间，换向点的坐标还可通过手轮和移动键加以修改。该修改和往复运动是否曾中断无关。

修改换向点坐标时注意：如果修改发生在往复运动激活后，新坐标只有在下一次朝该换向点移动时才生效。如果此时已经离开了该换向点，那修改在下一次往复中生效。

说明

如果在 NC/PLC 接口信号 DB21, ... DBX0.3 （激活 DRF ）置位的同时修改了换向点坐标，则手轮信号不仅会用于 DRF 偏移，还会用于换向点移动，换句话说：换向点移动了两倍的距离。

暂停时间

可通过设定数据设置每个换向点上的一段暂停时间。

在后续程序段中可修改该设置。该修改程序段同步，从下一个换向点起生效。

通过设定数据可以异步修改暂停时间。该修改随后在离开对应换向点后生效。

下表列出了在不同的暂停时间设定下轴在准停区内或换向点上的运动方式。

表格 11-1 不同暂停时间设定的作用

暂停时间设定	特性
-2	继续插补，不等待轴达到准停状态
-1	在换向点上等待轴达到粗准停状态
0	在换向点上等待轴达到精准停状态
>0	在换向点上等待轴达到精准停状态，接着等待暂停时间结束。

取消往复运动

在取消往复运动时，往复运动的结束有以下几种方式供选择：

- 在下一个换向点上结束
- 始终在换向点 1 上结束
- 始终在换向点 2 上结束

接着往复轴可能会执行光磨或移动到终点。

在从异步往复运动切换到光磨时和光磨期间，往复轴在换向点上的准停方式和“暂停时间设定”中的准停方式一样。所谓的一次光磨指轴从一个换向点移动到下一个换向点、然后回到出发换向点的过程（见下表）。

说明

带有运动同步动作和暂停时间的往复运动“OST1/OST2”

在设定的暂停时间结束后，系统内部会自动切换程序段，这一点可从轴的新余程上看出。在切换程序段后系统会检查往复运动是否被取消。此时会根据设置的运动控制方式“OSCTRL”确定运动结束方式。

可通过进给倍率来调节这种时间特性。

在开始光磨或者朝终点移动前，轴有时还会执行一次往复运动。

这会使人觉得往复运动的结束方式发生了变化，但实际上并不是如此。

表格 11-2 往复运动结束方式

功能	设定	说明
在某个换向点上结束	光磨次数等于 0 没有设定终点	往复运动在对应的换向点上结束。
进行指定次数的光磨后结束	光磨次数 不等于 0 没有设定终点	从对应的换向点起继续完成指令中指定次数的光磨。
进行指定次数的光磨和移动到指定终点后结束	光磨次数 不等于 0 设定了终点	从对应的换向点起完成指令中指定次数的光磨，然后移动到指定的终点。
不进行指定次数的光磨，但移动到指定终点后结束	光磨次数 等于 0 设定了终点	从对应的换向点起，轴移动到指定终点。

NC 语言:

可借助数控编程语言编写零件程序来控制异步往复运动。通过以下指令可以从零件程序中激活并控制异步往复运动，使它和数控程序的执行一致。

说明

如果在零件程序中直接写入设定数据，则修改在预处理时就已经生效，这对于程序执行来说太早。可以通过预处理停止指令 `STOPRE` 来实现同步。

文档:

编程说明

1) 启用/禁用往复运动:

- `OS[往复轴] = 1`; 为该轴启用往复运动
- `OS[往复轴] = 0`; 为该轴禁用往复运动

说明

每个轴都可以作为往复轴使用。

2) 往复运动结束:

- `WAITP`(往复轴)

定位轴指令，会暂停程序段处理，直到往复轴达到精准停状态，并可使预处理和主处理同步。往复轴再次设为定位轴，接着可以和平常一样使用。

如果希望一根轴作往复运动，应事先编写 `WAITP`（轴名）激活该轴。

在由 `PLC/HMI` 启动往复运动时也是如此。如果事先在数控程序中编写了该轴，也需要编写 `WAITP`（轴名）指令。从软件版本 **3.2** 起可通过机床数据确定

`$MA_AUTO_GET_TYPE` 是要继续手动编写 `WAITP()` 还是由系统自动执行该指令。

说明

编写了 `WAITP` 后，程序会等待往复运动结束。结束往复运动的方式有：由数控程序、`PLC` 或 `HMI` 取消往复运动或删除余程。

3) 设置换向点:

- `OSP1[往复轴] = 换向点 1 的坐标`
- `OSP2[往复轴] = 换向点 2 的坐标`

该坐标会在主处理中与程序段同步地进入对应的设定数据，在下一次修改该设定数据前一直保持生效。

如果当前采用的是增量坐标，则该坐标是相对于上一个在数控程序中写入的换向点的坐标。

4) 换向点上的暂停时间:

- OST1[往复轴] = 换向点 1 上的暂停时间，单位秒
- OST2[往复轴] = 换向点 2 上的暂停时间，单位秒

暂停时间数据会在主处理中与程序段同步地进入对应的设定数据，在下一次修改该设定数据前一直保持生效。

暂停时间的单位与通过 G4 写入的暂停时间的单位相同。

5) 设置进给率:

- FA[轴]= 进给率
定位轴的进给率。

进给率会在主处理中与程序段同步地传送给对应的设定数据。如果往复轴以旋转进给率旋转，则需要如功能说明 V1 所述指定关联性。

6) 设置运动控制选项:

- OSCTRL[往复轴] = (设置选项，复位选项)
设置选项的定义如下（复位选项可以取消该设置）：

表格 11-3 设置/复位选项

选项值	含义
0	取消往复运动后，轴移动到下一个换向点后停止（缺省）。只有复位选项值 1 和 2 才可进行该设置。
1	取消往复运动后，轴在换向点 1 上停止
2	取消往复运动后，轴在换向点 2 上停止
3	如果没有写入光磨次数，取消往复运动后，轴不移动到换向点
4	完成光磨后移动到终点
8	删除剩余行程时，往复轴会紧接着进行光磨，有时也移动到终点
16	删除剩余行程时，往复轴会移动到对应的换向点，如同取消往复运动一样
32	修改后的进给率从下一个换向点起才生效
64	若进给率为 0，则激活行程叠加，否则激活速度叠加
128	回转轴以最短行程定位
256	单程光磨
512	首先移动到起点

说明

选项值 0 到 3 指定了取消往复运动时轴在换向点上的运动方式。可以选择 0 到 3 中的任意一个选项值。可以根据需要结合选中的该值进行其他设置。其他选项值用加号“+”隔开。

示例：在取消 Z 轴的往复运动后，Z 轴应停止在换向点 1 上，然后移动到终点。若进给率有所变化，新速度应立即生效。在出现删除余程信号后，轴应立即停止。

OSCTRL[Z] = (1+4, 16+32+64)

设置选项和复位选项会在主处理中与程序段同步地进入对应的设定数据，在下一次修改该设定数据前一直保持生效。

说明

系统会首先分析复位选项，然后分析设置选项。

7) 光磨次数:

- OSNSC[往复轴] = 光磨次数

光磨次数会在主处理中与程序段同步地进入对应的设定数据，在下一次修改该设定数据前一直保持生效。

8) 结束往复运动后轴的目标终点:

- OSE[往复轴] = 往复轴的终点

该终点会在主处理中与程序段同步地进入对应的设定数据，在下一次修改该设定数据前一直保持生效。系统会自动隐性设置选项值 4，使轴移动到设置的终点。

9) 开始往复运动前轴的目标起点:

- OSB [往复轴] = 往复轴的起点

该起点会在主处理中与程序段同步地进入对应的设定数据 SD43790

\$SA_OSCILL_START_POS，在下一次修改该设定数据前一直保持生效。必须在设定数据 SD43770 \$SA_OSCILL_CTRL_MASK 中置位 9，往复轴才能移动到起点。往复轴**首先移动到起点**，然后移动到换向点 1。如果起点和换向点 1 重合，则轴接着移动到换向点 2。

除了程序指令 OSB 外，起点也可以直接在设定数据 SD43790 \$SA_OSCILL_START_POS 中设置。

所有设定数据和系统数据中的位置都是基本坐标系(BCS)中的坐标。OSB, OSE 中的位置是工件坐标系(WCS)中的坐标。

在到达起点后系统不会等待暂停时间结束，而是等待轴达到精准停，即使起点和换向点 1 重合。因此可确保满足设置的准停条件。

在起点和换向点 1 重合时，如果不希望在起点上有进给运动，可以用另一个同步动作进行设定，另见章节“起点和换向点重合时的往复运动（起点 = 换向点 1）(页 713)”。

编程举例

在章节“异步往复运动程序示例 (页 705)”举例说明了异步往复运动中重要的一些程序指令。

11.2.2 由 PLC 控制的异步往复运动

激活

PLC 可通过以下设定数据在所有运行方式（除 MDA REF 和 JOG REF 外）中激活往复运动：

SD43780 OSCILL_IS_ACTIVE（激活往复运动）

设置

PLC 可通过设定数据激活/禁用往复运动并修改以下数据：换向点位置、换向点上的暂停时间、进给率、运动控制选项、光磨次数和往复轴终点。但这些值也可以在 HMI 上直接作为设定数据输入，或者由数控程序设置。这些值在重新上电后仍保持生效，PLC 因此可直接通过设定数据 OSCILL_IS_ACTIVE（即通过变量服务）重启之前设置的往复运动。

边界条件

如果希望主轴作为进给轴执行由 PLC 启动的往复运动，则该主轴必须同时满足定位轴的相关前提条件，比如：主轴必须事先进入位置闭环控制方式 (SPOS)。

不管轴是否由 PLC 控制，都会对以下两个“停止”位作出响应：

- DB31, ... DBX28.5（在下一个换向点上停止）
- DB31, ... DBX28.6（沿制动斜坡停止）

11.2.3 异步往复运动中的特殊响应

由 PLC 控制

PLC 程序可通过 NC/PLC 接口信号接管对往复轴的控制。此类 NC/PLC 接口信号中也包含了程序结束信号、运行方式切换信号和单步信号。

正在由主处理进行插补的异步往复轴通过 NCU 系统软件对该信号作出响应，而不考虑数控程序的影响：比如“NC STOP”、报警处理、程序结束、程序控制和复位。

NC/PLC 接口信号(PLC→NCK) DB31, ... DBX28.7 (PLC 控制轴) 置 1 后，PLC 从 NC 接管对进给轴/主轴的控制。

关于 PLC 控制轴的相关信息参见章节“P2: 定位轴 (页 639)”。

不由 PLC 控制

PLC 不对轴进行控制时，在异步往复运动中，NC 便如同处理普通的定位轴 (POSA) 一样处理往复轴。

剩余行程删除

通道的删除剩余行程指令被忽略。

删除轴向剩余行程：

- 不由 PLC 控制：通过制动斜坡停止
- 由 PLC 控制：不停止（必须由 PLC 触发）

在上述两种情形中：往复轴停止后，系统可能会使往复轴继续移动到指定的换向点，然后删除剩余行程。接着往复轴执行指定次数的光磨，然后移动到终点，条件是 OSCILL_CTRL_MASK 中进行了对应的设置。

往复运动因此结束。

说明

在磨削时，测量钳可能因删除轴的剩余行程而夹在一起。

急停

使用急停功能结束往复运动且必须重启系统。

复位

往复轴沿制动斜坡停止，往复运动结束且被取消。选中的后续运动不被执行（光磨、移动到终点）。

工作区域限制和限位开关

如果在预处理时系统就已确定往复轴会超出当前激活的限位，系统会显示报警并且不启动往复运动。

在往复运动激活时，如果往复轴超出了暂时激活的一个限位（比如：第 2 个软件限位开关），则会沿制动斜坡停止，系统随后发出报警。

 小心

保护区
进行往复运动时无保护区生效。

跟踪方式

往复轴响应方式和定位轴一样。

程序结束

往复轴不由 PLC 控制时，只有在结束往复运动后才结束程序（和 POSA“模态式定位”一样）。

往复轴由 PLC 控制时，在程序结束后继续在两点之间来回移动。

运行方式切换

下表列出了哪些运行方式支持往复运动。在这些运行方式之间来回转换时，往复运动不受影响。转换到不支持往复运动的其他运行方式时，系统会发出报警，拒绝执行转换。同时由数控程序自动移动往复轴和手动移动往复轴时，系统会发出报警，拒绝执行轴运动。原则是首先发出的移动指令优先。

表格 11-4 支持往复运动的运行方式

运行方式	往复运动
AUTO	支持
MDA	支持

11.2 异步往复运动

运行方式	往复运动
MDA REPOS	支持
MDA TEACHIN	支持
MDA REF	不支持
JOG	支持
JOG REF	不支持
JOG REPOS	支持

单步方式（SINGLE BLOCK）

往复轴不由 PLC 控制时，在单步方式中和定位轴(POSA)的运行方式一样，即：往复轴会继续移动。

倍率

倍率的设定途径有：

NC/PLC 接口

轴倍率对往复轴生效。

编程

如同对定位轴一样，倍率对往复轴生效。

程序段搜索

在搜索程序段时，系统会记录最后一次有效的往复运动。在结束搜索后，系统会根据机床数据 OSCILL_MODE_MASK 的设置，在不同的时间点上激活往复运动。

OSCILL_MODE_MASK 位 0:

0: 到达起点后开始往复运动（缺省值）。

1: NC 启动后立即开始往复运动。

REORG

往复轴始终首先移动到换向点 1，然后继续移动。

ASUP

执行异步子程序(ASUP)期间会继续往复运动。

11.3 通过同步动作控制的往复运动

原理

异步往复运动可通过同步动作和进给运动关联在一起。

文档:

功能手册之同步动作分册

下文仅介绍一些和往复运动相关的运动同步动作。

功能

使用语言指令可以实现下列综合功能，语言指令在下文详细说明：

1. 在一个换向点上作进给运动，见章节“在换向点 1 或 2 上作进给运动 (页 696)”。
2. 在换向区内作进给运动，见章节“在换向区内作进给运动 (页 697)”。
3. 在两个换向点上作进给运动，见章节“在两个换向点上作进给运动 (页 699)”。
4. 在换向点上停止往复运动，直到进给运动结束，见章节“在换向点上停止往复运动 (页 699)”。
5. 继续往复运动，见章节“继续往复运动 (页 701)”。
6. 阻止下一个进给行程过早开始，见章节“阻止下一个进给行程过早开始 (页 701)”。

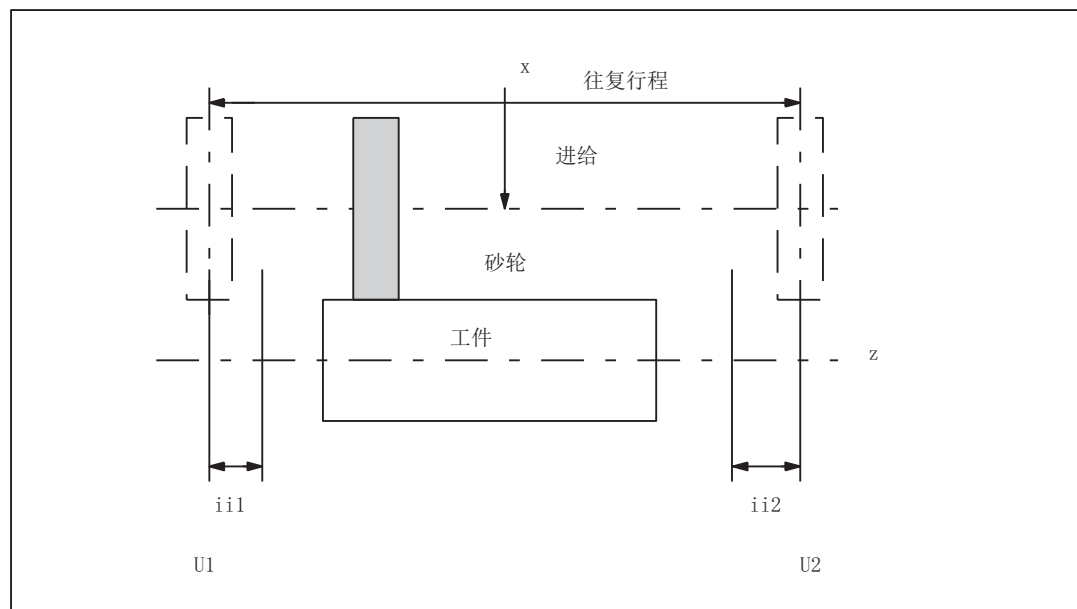


图 11-1 往复轴、进给轴的分布和定义

图例:

11.3 通过同步动作控制的往复运动

U1: 换向点 1

U2: 换向点 2

ii1: 换向区 1

ii2: 换向区 2

编程

在编写包含了进给轴指定、往复轴指定（见章节“OSCILL 指令：为往复轴指定进给轴（页 702）”）、进给方式 POSP 和运动同步动作的程序段前，首先要确定往复运动的相关参数，参见章节“异步往复运动的设置选项（页 683）”。

编写 WAITP [往复轴]（见 MD30552 \$MA_AUTO_GET_TYP），可使能该轴用于往复运动，往复运动参数因此可同时输入到系统中，更确切地说是设定数据中。在编写运动同步动作时可使用符号名，例如：\$SA_REVERSE_POS1 中使用 SA43700。

说明

在含 \$SA_REVERSE_POS 的运动同步动作中，比较结果始终在插补时间点上输出。因此此后的设定数据修改无效。

在含 \$AA_REVERSE_POS 的运动同步动作中，比较结果始终在插补期间输出。因此换向点的修改可立即生效。

- 运动同步条件 WHEN 和 WHENEVER

- 由运动程序段激活
 - OSCILL 指定进给轴和往复轴
 - POSP 指定进给方式

下文会详细介绍上述内容。

“示例”一章列出了几个例子。

说明

如果用运动同步条件 WHEN 和 WHENEVER 定义的条件不再有效，则该条件的结果从原先的“倍率设为 0%”自动改为“倍率设为 100%”。

主处理计算

同步条件可按插补周期在主处理中与当前实际值进行比较（比较式中右边的\$\$变量）。在正常的系统变量对比中，表达式是在预处理中计算的。关于同步动作的详细说明参见

文档:

功能手册之同步动作分册

示例 1

通过设定数据固定设置往复运行和换向点位置:

程序代码	注释
\$SA_OSCILL_REVERSE_POS1[Z]=-10	
\$SA_OSCILL_REVERSE_POS2[Z]=10	
G0 X0 Z0	
WAITP(Z)	
ID=1 WHENEVER \$AA_IM[Z] <	
\$SA_OSCILL_REVERSE_POS1[Z] DO	
\$AA_OVR[X]=0	
ID=2 WHENEVER \$AA_IM[Z] >	; ; 如果往复轴的实际位置
\$SA_OSCILL_REVERSE_POS2[Z] DO	
\$AA_OVR[X]=0	
	; 已经超过换向点,
	; ; 则进给轴停止。
	; 激活往复运动
OS[Z]=1 FA[X]=1000 POS[X]=40	; 取消往复运动
OS[Z]=0	
M30	

示例 2

往复运动进行过程中需要修改换向点, 即通过操作界面修改换向点 1 的位置, 且该修改要立即生效:

程序代码	注释
\$SA_OSCILL_REVERSE_POS1[Z]=-10	
\$SA_OSCILL_REVERSE_POS2[Z]=10	
G0 X0 Z0	
WAITP(Z)	
ID=1 WHENEVER \$AA_IM[Z] < \$	
\$SA_OSCILL_REVERSE_POS1[Z] DO	
\$AA_OVR[X]=0	
ID=2 WHENEVER \$AA_IM[Z] > \$	
\$SA_OSCILL_REVERSE_POS2[Z] DO	
\$AA_OVR[X]=0	
	; ; 如果往复轴的实际位置
	; 已经超过换向点,
	; 则进给轴停止。

11.3 通过同步动作控制的往复运动

程序代码	注释
OS[Z]=1 FA[X]=1000 POS[X]=40	; 激活往复运动
OS[Z]=0	; 取消往复运动
M30	

11.3.1 在换向点 1 或 2 上作进给运动

功能

只要往复轴没有到达换向点，进给轴就不作进给运动。

应用

在换向点上直接作进给运动

编程

```
针对换向点 1:
WHENEVER $AA_IM[Z] <> $SA_OSCILL_REVERSE_POS1[Z]
DO $AA_OVR[X] = 0 $AA_OVR[Z] = 100

针对换向点 2:
WHENEVER $AA_IM[Z] <> $SA_OSCILL_REVERSE_POS2[Z]
DO $AA_OVR[X] = 0 $AA_OVR[Z] = 100

系统变量的含义:
$AA_IM[Z]: 机床坐标系中往复轴 Z 轴的实际坐标
$SA_OSCILL_REVERSE_POS1[Z]: 往复轴换向点 1 的坐标
$AA_OVR[X]: 进给轴的倍率
$AA_OVR[Z]: 往复轴的倍率

指令的含义:
WHENEVER ... DO ...只要 xxx，就 xxx
```

进给

进给量由指令 POSP 定义，参见章节“POSP 指令：指定进给运动方式 (页 703)”。

分配

往复轴和进给轴之间的关联由指 OSCILL 定义，参见章节“OSCILL 指令：为往复轴指定进给轴 (页 702)”。

11.3.2 在换向区内作进给运动

功能

换向区 1:

只要往复轴没有进入换向区（从换向点 1 起，长度为变量 ii1），进给轴就不开始进给运动。前提是换向点 1 的坐标小于换向点 2 的坐标。如果不满足条件，则需要修改换向点坐标。

应用

换向区 1:

在往复轴进入换向区 1 后，再开始进给运动，见章节“图 11-1 往复轴、进给轴的分布和定义 (页 693)”。

编程

换向区 1:

```
WHENEVER $AA_IM[Z] > $SA_OSCILL_REVERSE_POS1[Z] + ii1
```

```
DO $AA_OVR[X] = 0
```

系统变量的含义：

\$AA_IM[Z]: 往复轴 Z 轴的实际坐标

\$SA_OSCILL_REVERSE_POS1[Z]: 往复轴换向点 1 的坐标

\$AA_OVR[X]: 进给轴的倍率

ii1: 换向区长度（用户变量）

指令的含义：

```
WHENEVER ... DO ... 只要 xxx，就 xxx
```

功能

换向区 2:

11.3 通过同步动作控制的往复运动

只有在往复轴进入换向区 2 后（往复轴坐标大于等于“换向点 2 坐标和 ii2 之差”），进给轴才开始进给运动，前提是换向点 2 的坐标大于换向点 1 的坐标。如果不满足条件，则需要修改换向点坐标。

应用

换向区 2:

在往复轴进入换向区 2 后，再开始进给运动，见章节“图 11-1 往复轴、进给轴的分布和定义 (页 693)”。

编程

换向区 2:

```
WHENEVER $AA_IM[Z] < $SA_OSCILL_REVERSE_POS2[Z] - ii2
```

```
DO $AA_OVR[X] = 0
```

说明:

\$AA_IM[Z]: 往复轴 Z 轴的实际坐标

\$SA_OSCILL_REVERSE_POS2[Z]: 往复轴换向点 2 的坐标

\$AA_OVR[X]: 进给轴的倍率

ii2: 换向区 2 长度（用户变量）

进给

进给量由指令 POSP 定义，参见章节“POSP 指令：指定进给运动方式 (页 703)”。

分配

往复轴和进给轴之间的关联由指 OSCILL 定义，参见章节“OSCILL 指令：为往复轴指定进给轴 (页 702)”。

参见

通过同步动作控制的往复运动 (页 693)

11.3.3 在两个换向点上作进给运动

原理

上文介绍的两种进给运动方式可以随意组合在一起使用。

组合

双侧进给运动:

换向点 1 + 换向点 2

换向点 1 + 换向区 2

换向区 1 + 换向点 2

换向区 1 + 换向区 2

单侧进给运动

换向点 1

换向点 2

换向区 1

换向区 2

(参见章节“在换向点 1 或 2 上作进给运动 (页 696)”和“在换向区内作进给运动 (页 697)”。

11.3.4 在换向点上停止往复运动

功能

换向点 1:

只要往复轴到达了换向点 1，就将其速度倍率设为零来停止往复轴，然后进给轴开始进给运动。

11.3 通过同步动作控制的往复运动

应用

该同步动作用于停止往复轴，直到进给轴完成了一个进给行程。如果不希望往复轴在换向点 1 上等待该进给行程的结束，就无需编写该同步动作。该同步动作同时可用于重启被前一个有效的同步动作暂停的进给运动。

编程

```
WHENEVER $AA_IM[往复轴] == $SA_OSCILL_REVERSE_POS1[往复轴]
```

```
DO $AA_OVR[往复轴] = 0 $AA_OVR[进给轴] = 100
```

系统变量的含义：

\$AA_IM[往复轴]: 往复轴的当前坐标

\$SA_OSCILL_REVERSE_POS1[往复轴]: 往复轴换向点 1 的坐标

\$AA_OVR[往复轴]: 往复轴的倍率

\$AA_OVR[进给轴]: 进给轴的倍率

功能

换向点 2:

只要往复轴到达了换向点 2，就将其速度倍率设为 0 来停止往复轴，然后进给轴开始进给运动。

应用

该同步动作用于停止往复轴，直到进给轴结束了一次进给行程。如果不希望往复轴在换向点 2 上等待该进给行程的结束，就无需编写该同步动作。该同步动作同时可用于重启被前一个有效的同步动作暂停的进给运动。

编程

```
WHENEVER $AA_IM[往复轴] == $SA_OSCILL_REVERSE_POS2[往复轴]
```

```
DO $AA_OVR[往复轴] = 0 $AA_OVR[进给轴] = 100
```

说明:

\$AA_IM[往复轴]: 往复轴的当前坐标

\$SA_OSCILL_REVERSE_POS2[往复轴]: 往复轴换向点 2 的坐标

\$AA_OVR[往复轴]: 往复轴的倍率

\$AA_OVR[进给轴]: 进给轴的倍率

11.3.5 继续往复运动

功能

只要进给轴的当前剩余行程变为 0，即一个进给行程结束，往复轴就开始下一个往复行程。

应用

该同步动作用于在结束一个进给行程后继续往复轴的往复运动。 如果不希望往复轴等待单个进给行程的结束，可以不用编写指定往复轴在换向点上停止的同步动作。

编程

```
WHENEVER $AA_DTEPW[进给轴] == 0
```

```
DO $AA_OVR[往复轴] = 100
```

系统变量的含义:

\$AA_DTEPW[进给轴]: 工件坐标系中进给轴的剩余行程 进给轴目标行程

\$AA_OVR[往复轴]: 往复轴倍率

指令的含义:

WHENEVER ... DO ... 只要 xxx，就 xxx

11.3.6 阻止下一个进给行程过早开始

功能

上文说明的功能可阻止在往复轴没有到达换向点或换向区时进给轴作进给运动，但无法在一个进给行程完成后阻止进给轴过早开始下一个进给行程。

应用

为此可引入一个通道专用的标记位，该标记位会在一个进给行程完成后置位（剩余行程为 0），在往复轴离开换向区后复位。然后用同步动作阻止下一个进给行程。

11.3 通过同步动作控制的往复运动

编程

```
WHENEVER $AA_DTEPW[进给轴] == 0
```

```
DO $AC_MARKER[下标] = 1
```

以换向点 1 为例:

```
WHENEVER $AA_IM[Z] <> $SA_OSCILL_REVERSE_POS1[Z]
```

```
DO $AC_MARKER[下标] = 0
```

```
WHENEVER $AC_MARKER[下标] == 1
```

```
DO $AA_OVR[进给轴] = 0
```

系统变量的含义:

\$AA_DTEPW[进给轴]: 工件坐标系中进给轴的剩余行程 进给轴目标行程

\$AC_MARKER[下标]: 通道专用的标记, 带下标

\$AA_IM[往复轴]: 往复轴的当前坐标

\$SA_OSCILL_REVERSE_POS1[往复轴]: 往复轴换向点 1 的坐标

\$AA_OVR[进给轴]: 进给轴倍率

指令的含义:

WHENEVER ... DO ... 只要 xxx, 就 xxx

11.3.7 OSCILL 指令: 为往复轴指定进给轴

功能

OSCILL 指令用于为往复轴指定一根或多根进给轴。该指令同时会启动往复运动。

此处指定的进给轴会通过 NC/PLC 接口通知给 PLC。如果往复轴由 PLC 控制, PLC 会一同监控这些进给轴。在分析进给轴的信号后, PLC 会通过接口信号中的 2 个停止位来相应地控制往复轴。

应用

为往复轴指定对应的进给轴, 进给轴的工作方式已事先通过同步条件定义完毕。往复运动开始。

编程

OSCILL[往复轴] = (进给轴 1, 进给轴 2, 进给轴 3)

不需要使用括弧中的进给轴 2 和进给轴 3 时，可直接省略这两根轴和逗号。

11.3.8 POSP 指令：指定进给运动方式

功能

必须为进给轴设定以下行程数据：

- 总进给行程
- 换向点/换向区的单次进给行程
- 终点上的进给行程

应用

该指令必须在指令 OSCILL 后写入，使系统得知进给轴在换向点/换向区需要移动的进给行程。

编程

POSP[进给轴] = (终点, 单次行程, 方式)

终点：所有进给行程结束后进给轴的终点。

单次行程：换向点/换向区的单次进给行程

方式 0：进给轴到终点的总进给行程不均匀划分，最后两次的进给行程比较长，且相同（缺省值）。

方式 1：进给轴到终点的总进给行程均匀划分，即所有单次进给行程都一样。

11.3.9 从外部触发往复轴换向

通过 PLC 设计按键之类的措施可以由 PLC 切换往复区或使往复轴立即换向。

PLC 可通过边沿触发式 PLC 输入信号 DB31, ... DBX28.0（往复轴换向）使往复轴制动，然后使往复轴反向移动。制动状态通过 PLC 输出信号 DB31, ... DBX100.2（往复轴换向生效）报告给 PLC。

11.3 通过同步动作控制的往复运动

如果希望将往复轴的制动点设为**新的换向点**，可设置 PLC 信号 DB31, ... DBX28.4（修改换向点）。

PLC 输入信号 DB31, ... DBX28.3（选择换向点）失效，最后一次由外部触发的换向生效。

不允许通过手轮或移动键来修改换向点。若执行该操作，系统会发出报警 20081“制动点不能用作换向点 - 手轮生效”。在排除错误后，该报警自行消失。

暂停时间

由外部触发的换向无暂停时间。程序会等待轴达到精准停状态。因此可确保满足设置的准停条件。

进给运动

在起点和换向点重合的往复运动中，如果外部触发了换向，则进给轴不作进给运动，因为此时往复轴还没有到达换向点，不满足对应的同步条件。

系统变量

在移动到换向点 1 的过程中往复轴的制动点可通过系统变量 \$AA_OSCILL_BREAK_POS1 读取；

在移动到换向点 2 的过程中往复轴的制动点可通过系统变量 \$AA_OSCILL_BREAK_POS2 读取。

在重新移动到对应的换向点过程中，变量 \$AA_OSCILL_BREAK_POS1 和 \$AA_OSCILL_BREAK_POS2 指出的是对应的换向点。

也就是说，在由外部触发换向后，\$AA_OSCILL_BREAK_POS1 和 \$AA_OSCILL_REVERSE_POS1 的值不同；或者 \$AA_OSCILL_BREAK_POS2 和 \$AA_OSCILL_REVERSE_POS2 的值不同。

因此可通过一个同步动作来**检测**由外部触发的换向，见章节“示例 (页 705)”。

特殊情况

如果 PLC 输入信号“往复轴换向”在往复轴移动到起点的过程中置位，往复轴会终止到起点的移动，从制动点 1 起换向。

如果 PLC 输入信号“往复轴换向”在某段暂停时间内置位，该暂停时间会终止。如果此时往复轴还没有到达精准停状态，程序会一直等待，之后使轴换向。

如果 PLC 输入信号“往复轴换向”在往复轴移动到终点的过程中置位，往复轴会终止到终点的移动，往复运动结束。

由外部触发的往复轴换向的示例参见章节“由外部触发往复轴换向时通过同步动作修改换向点 (页 715)”。

11.4 边界条件

“往复运动”功能的可用性

该功能是一个软件选件，授权后方可和硬件组合使用。

11.5 示例

前提条件

下文列举的示例需要使用段落

- 异步往复运动
- 与
- 通过同步动作控制的往复运动
- 介绍的程序指令。

11.5.1 异步往复运动程序示例

任务

往复轴 Z 轴要在 -10 和 10 之来回移动。往复轴在换向点 1 上要达到粗准停状态；在换向点 2 上无需达到任何准停状态。往复轴以进给率 5000 移动。往复结束后往复轴要执行 3 次光磨，终点是 30。进给轴是 X 轴，进给率是 1000，终点是 15。

11.5 示例

程序段落

程序代码	注释
OSP1[Z]=-10	; 换向点 1
OSP2[Z]=10	; 换向点 2
OST1[Z]=-1	; 换向点 1 上的暂停时间: 粗准停
OST2[Z]=-2	; 换向点 2 上的暂停时间: 无准停
FA[Z]=5000	; 往复轴的进给率
OSNSC[Z]=3	; 三次光磨
OSE[Z]=-3	; 终点位置
OS1 F500 X15	; 启动往复运动, x 轴为进给轴
	; 进给率为 500, 目标位置为 15

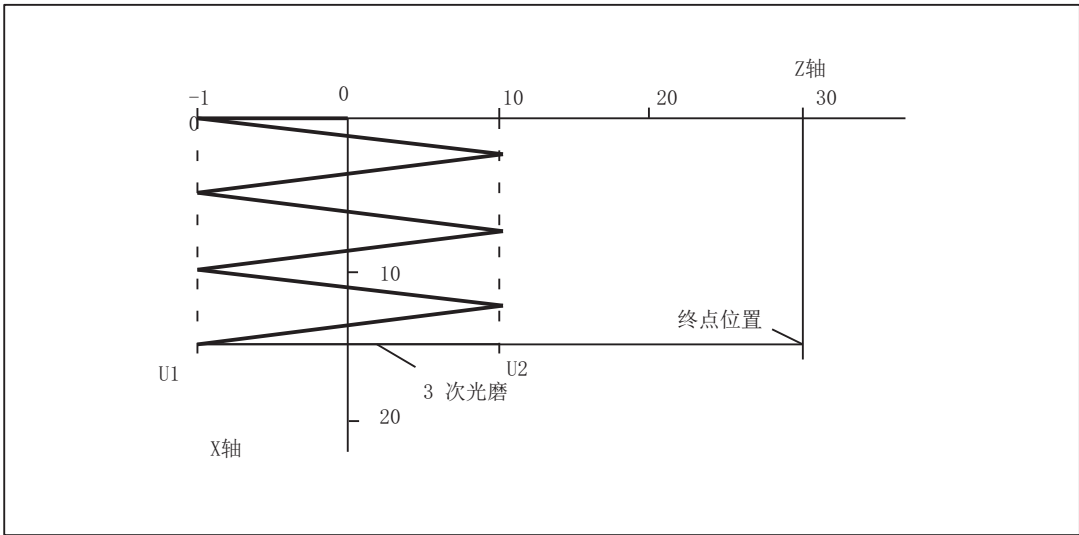


图 11-2 示例 1 中的往复运动和进给运动

11.5.2 示例 1：带同步动作的往复运动

任务

在换向点 1 一侧，一旦往复轴到达该点，进给轴就开始作进给运动，之后往复轴才继续移动。而在换向点 2 一侧，在往复轴距离换向点 2 “-6 ” 的位置，进给轴就开始作进给运动，往

往复轴到达后换向点继续移动，不管进给轴是否完成进给。往复轴为 Z 轴，进给轴为 X 轴（参见章节“通过同步动作控制的往复运动 (页 693)”）。

说明

设定数据 OSCILL_REVERSE_POS_1/2 是机床坐标系中的坐标值，只能和 \$AA_IM[n] 比较。

程序段落

程序代码	注释
； 示例 1： 含同步动作的往复运动	
OSP1[Z]=10 OSP2[Z]=60	； 定义换向点 1 和 2
OST1[Z]=-2 OST2[Z]=0	； 换向点 1： 无准停
	； 换向点 2： 精准停
FA[Z]=5000 FA[X]=250	； 往复轴进给率，进给轴进给率。
OSCTRL[Z]=(1+8+16,0)	； 在换向点 1 处结束往复运动
	； 在删除剩余行程后往复轴进行光磨，移动到终点
	； 在删除剩余行程后往复轴移动到对应的换向点
	；
OSNSC[Z]=3	； 3 次光磨
OSE[Z]=0	； 终点 = 0
WAITP(Z)	； 允许 z 轴往复
；	
； 运动同步动作：	
；	
； 任何时候只要	往复轴在机床坐标系
；	中的当前坐标
； 不等于	换向点 1 的坐标
； 就	将下标为 1 的标志置 0
；	(标志 1 复位)
WHENEVER \$AA_IM[Z]<>\$SA_OSCILL_REVERSE_POS1[Z] DO \$AC_MARKER[1]=0	
；	
； 任何时候只要	往复轴在机床坐标系
；	中的当前坐标
； 小于	换向区 2 的起点坐标
；	(本例中为： 换向点 2 -6)，
； 就	将进给轴的倍率设为 0%
； 并且	将下标为 2 的标志置 0
；	(标志 2 复位)。
WHENEVER \$AA_IM[Z]<\$SA_OSCILL_REVERSE_POS2[Z]-6 DO \$AA_OVR[X]=0 \$AC_MARKER[2]=0	
；	
； 任何时候只要	往复轴在机床坐标系

11.5 示例

程序代码	注释
;	中的当前坐标
; 等于	换向点 1 的坐标,
; 就	将往复轴的倍率设为 0%
;	
; 并且	将进给轴的倍率设为
;	100% (此时会取消之前的
;	同步动作!)
WHENEVER \$AA_IM[Z]==\$SA_OSCILL_REVERSE_POS1[Z] DO \$AA_OVR[Z]=0 \$AA_OVR[X]=100	
;	
; 任何时候只要	单次进给的剩余行程
; 等于	0,
; 就	将下标为 2 的标志置 1
; 并且	将下标为 1 的标志置 1
WHENEVER \$AA_DTEPW[X]==0 DO \$AC_MARKER[2]=1 \$AC_MARKER[1]=1	
;	
; 任何时候只要	下标为 2 的标志
; 等于	1,
; 就	将进给轴的倍率设为
;	0%, 以阻止下一次进给行程过早开始
;	(往复轴还没有离开换向点 1)
;	
WHENEVER \$AC_MARKER[2]==1 DO \$AA_OVR[X] = 0	
;	
; 任何时候只要	下标为 1 的标志
; 等于	1,
; 就	将进给轴的倍率设为
;	0%, 以阻止下一次进给行程过早开始
;	(往复轴还没有离开换向点 2)
;	
; 并且	将往复轴的倍率设为
;	100% (开始往复运动)
WHENEVER \$AC_MARKER[1]==1 DO \$AA_OVR[X]=0 \$AA_OVR[Z]=100	
;	
; 如果往复轴在机床坐标系中的当前坐标	
; 等于	换向点 1 的坐标,
; 就	将往复轴的倍率设为
;	100%
; 并且	将进给轴的倍率设为
;	0% (此时会取消第二个
;	同步动作!)
WHEN \$AA_IM[Z]==\$SA_OSCILL_REVERSE_POS1[Z] DO \$AA_OVR[Z]=100 \$AA_OVR[X]=0	
;	
;	-----

程序代码	注释
OSCILL[Z]=(X)	; 往复轴为 Z 轴，进给轴为 X 轴
POSP[X]=(5,1,1)	
	; 进给轴要移动到
	; 终点 5，单次进给行程为
	; 1，总进给行程
	; 均匀划分
M30	; 程序结束

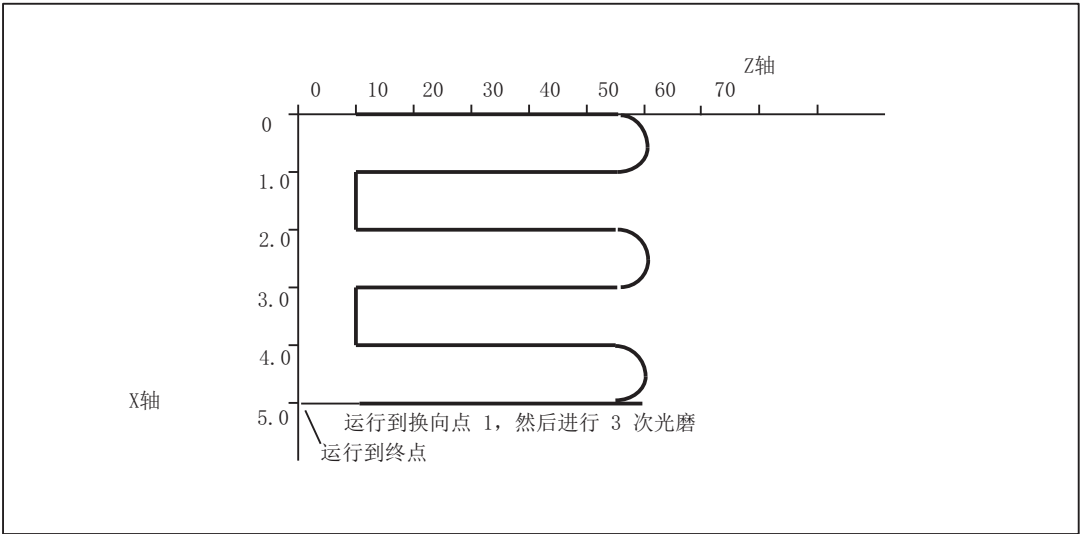


图 11-3 示例 1 中的往复运动和进给运动

11.5.3 示例 2：带同步动作的往复运动

任务

在换向点 1 上无进给运动。在换向点 2 一侧，在往复轴距离该点 ii2 的位置上，进给轴就开始作进给运动，往复轴到达换向点等待进给轴完成进给。往复轴为 Z 轴，进给轴为 X 轴。

程序段落

示例 2：含同步动作的往复运动

程序代码	注释
DEF INT ii2	; 定义变量，用于换向区 2

11.5 示例

程序代码	注释
;	
OSP1[Z]=10 OSP2[Z]=60	; 定义换向点 1 和 2
OST1[Z]=0 OST2[Z]=0	; 换向点 1: 精准停
;	换向点 2: 精准停
FA[Z]=5000 FA[X]=100	; 往复轴进给率, 进给轴进给率
OSCTRL[Z]=(2+8+16,1)	; 在换向点 2 处结束往复运动
;	在删除剩余行程后往复轴进行光磨, 然后移动到终点
;	在删除剩余行程后往复轴移动到
;	对应的换向点
OSNSC[Z]=3	; 3 次光磨
OSE[Z]=70	; 终点 = 70
ii2=2	; 设定换向区
WAITP(Z)	; 允许 z 轴往复
;	
; 运动同步动作:	
; 任何时候只要	往复轴在机床坐标系
;	中的当前坐标
; 小于	换向区 2 的起点坐标,
; 就	将进给轴的倍率设为
;	0%
; 并且	将下标为 0 的标志置 0
WHENEVER \$AA_IM[Z]<\$SA_OSCILL_REVERSE_POS2[Z]-ii2 DO \$AA_OVR[X]=0 \$AC_MARKER[0]=0	
;	
; 任何时候只要	往复轴在机床坐标系
;	中的当前坐标
; 大于等于	换向点 2 的坐标
; 就	将往复轴的倍率设为
;	0%
WHENEVER \$AA_IM[Z]>=\$SA_OSCILL_REVERSE_POS2[Z] DO \$AA_OVR[Z]=0	
;	
; 任何时候只要	单次进给的剩余行程
; 等于	0,
; 就	将下标为 0 的标志置 1
WHENEVER \$AA_DTEPW[X]==0 DO \$AC_MARKER[0]=1	
;	
; 任何时候只要	下标为 0 的标志
; 等于	1,
; 就	将进给轴的倍率设为 0%, 以阻止
;	下一个进给行程过早开始
;	(往复轴还没有离开换向点 2,
;	进给轴就已经准备
;	好下一个进给行程了)
; 并且	将往复轴的倍率设为 100% (以便取消
;	第 2 个同步动作)

程序代码	注释
;	
WHENEVER \$AC_MARKER[0]==1 DO \$AA_OVR[X]=0 \$AA_OVR[Z]=100	
;	
OSCILL[Z]=(X)	; 启动往复轴
POSP[X]=(5,1,1)	
;	Z 轴，进给轴
;	为 X 轴
;	X 轴的终点为 5
;	单次进给行程为 1
;	
M30	

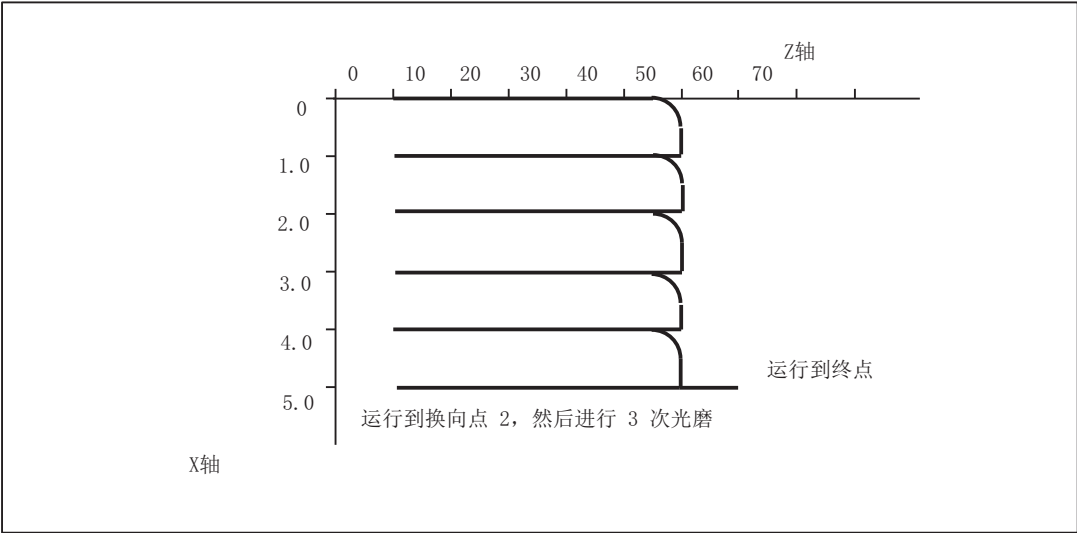


图 11-4 示例 2 中的往复运动和进给运动

11.5.4 示例：起点

11.5.4.1 通过语言指令来定义起点

程序代码	注释
WAITP(Z)	; 允许 Z 轴往复
OSP1[Z]=10 OSP2[Z]=60	; 定义换向点 1 和 2
OST1[Z]==-2 OST2[Z]=0	; 换向点 1: 无准停
	; 换向点 2: 精准停
FA[Z]=5000 FA[X]=2000	; 往复轴的进给率

11.5 示例

程序代码	注释
	; 进给轴的进给率
OSCTRL[Z]=(1+8+16,0)	; 在换向点 1 结束往复运动
	; 在删除剩余行程后往复轴进行光磨，移动到终点
	; 在删除剩余行程后往复轴移动到对应的换向点
	;
OSNSC[Z]=3	; 3 次光磨
OSE[Z]=0	; 终点= 0
OSB[Z]=0	; 起点 = 0
OS[Z]=1 X15 F500	; 启动往复运动，伴随有连续进给运动
OS[Z]=0	; 取消往复运动
WAITP(Z)	; 等待往复运动结束
M30	

说明

往复轴 Z 轴首先移动到起点，本例中为 0，然后再开始在换向点 10 到 60 之间来回移动。
在 X 轴移动到终点 15 后，往复轴进行 3 次光磨，然后移动到终点 0。

11.5.4.2 通过设定数据启动往复运动

程序代码	注释
WAITP(Z)	
STOPRE	
\$SA_OSCILL_REVERSE_POS1[Z]=-10	; 换向点 1 = -10
\$SA_OSCILL_REVERSE_POS2[Z]=30	; 换向点 2 = 30
\$SA_OSCILL_START_POS[Z] = -50	; 起点 = -50
\$SA_OSCILL_CTRL_MASK[Z] = 512	; 移动到起点，
	; 在取消往复运动时
	; 在下一个换向点停止
	; 不移动到终点
	; 在删除剩余行程时不进行光磨
\$SA_OSCILL_VELO[Z] = 5000	; 往复轴的进给率
\$SA_OSCILL_IS_ACTIVE[Z] = 1	; 启动
\$SA_OSCILL_DWELL_TIME1[Z] = -2	; 不等待达到准停
\$SA_OSCILL_DWELL_TIME2[Z] = 0	; 等待达到精准停
STOPRE	
X30 F100	
\$SA_OSCILL_IS_ACTIVE[Z] = 0	; 停止
WAITP(Z)	
M30	

说明

往复轴 Z 轴首先移动到起点，本例中为 -50，然后再开始在换向点 -10 到 30 之间来回移动。
在 X 轴移动到终点 30 后，往复轴移动到下一个换向点，结束往复运动。

11.5.4.3 起点和换向点重合时的往复运动（起点 = 换向点 1）

含同步动作的往复运动

程序代码	注释
N701	; 含同步动作的往复运动，
	; 起点 = 换向点 1
;	
N702 OSP1[Z]=10 OSP2[Z]=60	; 定义换向点 1 和 2
N703 OST1[Z]=0 OST2[Z]=0	; 换向点 1: 粗准停
	; 换向点 2: 精准停
N704 FA[Z]=5000 FA[X]=2000	; 往复轴的进给率
	; 进给轴的进给率
N705 OSCTRL[Z]=(1+8+16,0)	; 在换向点 1 上取消往复运动
	; 在删除剩余行程后进行光磨
	; 然后移动到终点
	; 在删除剩余行程后移动到对应的
	; 换向点
	;
N706 OSNSC[Z]=3	; 3 次光磨
N707 OSE[Z]=0	; 终点 = 0
N708 OSB[Z]=10	; 起点 = 10
N709 WAITP(Z)	; 允许 Z 轴往复
;	
; 运动同步动作:	
; 将下标为 2 的标志设为 1 (初始化)	
WHEN TRUE DO \$AC_MARKER[2]=1	
;	
; 任何时候只要	下标为 2 的标志为 0,
;	且往复轴的当前坐标
;	不等于换向点 1 的坐标
; 就	将下标为 1 的标志设为 0
;	
WHENEVER (\$AC_MARKER[2] == 0) AND \$AA_IW[Z]>\$SA_OSCILL_REVERSE_POS1[Z]	
DO \$AC_MARKER[1]=0	
; 任何时候只要	往复轴的当前坐标
;	小于换向区 2 的起点坐标
;	

11.5 示例

程序代码	注释
;	将进给轴的倍率设为
;	0, 且将下标为 0 的标志设为
;	0
WHENEVER \$AA_IW[Z]<\$SA_OSCILL_REVERSE_POS2[Z]-6 DO \$AA_OVR[X]=0 \$AC_MARKER[0]=0	
;	
;	往复轴的当前坐标
;	等于换向点 1 的坐标
;	将往复轴的倍率设为
;	0, 且将进给轴的倍率
;	设为 100%, 此时会取消
;	之前的同步动作!
WHENEVER \$AA_IW[Z]==\$SA_OSCILL_REVERSE_POS1[Z] DO \$AA_OVR[Z]=0 \$AA_OVR[X]=100	
;	
;	单次进给的剩余行程等于 0
;	
;	将下标为 0 的标志设为 1, 且将
;	将下标为 1 的标志设为 1
WHENEVER \$AA_DTEPW[X]==0 DO \$AC_MARKER[0]=1 \$AC_MARKER[1]=1	
;	
;	下标为 0 的标志为 1
;	将进给轴的倍率设为
;	0%, 以阻止下一次
;	进给行程过早开始!
WHENEVER \$AC_MARKER[0]==1 DO \$AA_OVR[X] = 0	
;	
;	下标为 1 的标志为 1
;	将进给轴的倍率设为
;	0%, 以阻止下一次
;	进给行程过早开始! 并且
;	将往复轴的倍率设为 100%
;	此时会取消之前的
;	同步动作!
WHENEVER \$AC_MARKER[1]==1 DO \$AA_OVR[X]=0 \$AA_OVR[Z]=100	
;	
;	往复轴的当前坐标
;	等于换向点 1 的坐标
;	复位下标为 2 的标志
;	使能第 1 个同步动作 (在移动到起点
;	后没有进给运动)
;	== 换向点 1)
WHEN \$AA_IW[Z]==\$SA_OSCILL_REVERSE_POS1[Z] DO \$AC_MARKER[2]=0	
;	
;	
N750 OSCILL[Z]=(X) POSP[X]=(5,1,1)	

程序代码	注释
； 往复轴为 Z 轴，进给轴为 X 轴	
； X 轴终点为 5	
； 单次进给行程为 1	
； 总行程均匀划分	
；	
N780 WAITP(Z)	； Z 轴再次使能
；	
N790 X0 Z0	
N799 M30	； 程序结束

说明

起点和换向点 1 重合。通过同步动作 **WHEN**可阻止到达起点后的进给运动。

11.5.5 示例：从外部触发往复轴换向

11.5.5.1 由外部触发往复轴换向时通过同步动作修改换向点

程序代码	注释
DEFINE BREAKPZ AS \$AA_OSCILL_BREAK_POS1[Z]	
DEFINE REVPZ AS \$SA_OSCILL_REVERSE_POS1[Z]	
WAITP(Z)	； 允许 Z 轴往复
OSP1[Z]=10 OSP2[Z]=60	； 定义换向点 1 和 2
OSE[Z]=0	； 终点 = 0
OSB[Z]=0	； 起点 = 0
	； 在从外部触发往复轴换向时，修改
	； 换向点 1 的坐标
WHENEVER BREAKPZ <> REVPZ DO \$\$SA_OSCILL_REVERSE_POS1 = BREAKPZ	
OS[Z]=1 X150 F500	； 启动往复运动，伴随有连续进给运动
OS[Z]=0	； 取消往复运动
WAITP(Z)	； 等待往复运动结束
M30	

11.6 数据表

11.6 数据表

11.6.1 机床数据

11.6.1.1 通用机床数据

编号	名称: \$MN_	描述
10710	PROG_SD_RESET_SAVE_TAB	需要从设定数据中备份的往复运动设置
11460	OSCILL_MODE_MASK	异步往复运动的控制位

11.6.2 设定数据

11.6.2.1 进给轴/主轴专用设定数据

编号	名称: \$SA_	描述
43700	OSCILL_REVERSE_POS1	换向点 1 坐标
43710	OSCILL_REVERSE_POS2	换向点 2 坐标
43720	OSCILL_DWELL_TIME1	换向点 1 上的暂停时间
43730	OSCILL_DWELL_TIME2	换向点 2 上的暂停时间
43740	OSCILL_VELO	往复轴的进给速度
43750	OSCILL_NUM_SPARK_CYCLES	修光次数
43760	OSCILL_END_POS	在结束光磨后往复轴的终点
43770	OSCILL_CTRL_MASK	往复运动的控制位
43780	OSCILL_IS_ACTIVE	激活/取消往复运动
43790	OSCILL_START_POS	往复轴在到达换向点 1 前的起点, 在 SD43770 中激活。

11.6.3 信号

11.6.3.1 到进给轴/主轴的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
从外部触发往复轴换向	DB31, ...DBX28.0	-
设置换向点	DB31, ...DBX28.3	-
修改换向点	DB31, ...DBX28.4	-
在下一个换向点上停止	DB31, ...DBX28.5	-
沿制动斜坡停止	DB31, ...DBX28.6	-
PLC 控制轴	DB31, ...DBX28.7	-

11.6.3.2 来自进给轴/主轴的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
往复轴换向生效	DB31, ...DBX100.2	-
往复运动无法启动	DB31, ...DBX100.3	-
往复运动出错	DB31, ...DBX100.4	-
正进行光磨	DB31, ...DBX100.5	-
正进行往复运动	DB31, ...DBX100.6	-
往复运动激活	DB31, ...DBX100.7	-

11.6.4 系统变量

11.6.4.1 用于同步动作的主处理变量

读主处理变量

以下变量可用于读主处理变量:

\$A_IN[<算数 表达式>]	数字量输入端 (布尔数)
\$A_OUT[<算数 表达式>]	数字量输出端 (布尔数)

11.6 数据表

\$A_INA[<算数 表达式>]	模拟量输入端（布尔数）
\$A_OUTA[<算数 表达式>]	模拟量输出端（布尔数）
\$A_INCO[<算数 表达式>]	比较器输入端（布尔数）
\$AA_IW[<轴表达式>]	轴实际位置 PCS（实数）
\$AA_IB[<轴表达式>]	轴实际位置 BCS (Real)
\$AA_IM[<轴表达式>]	轴实际位置 MCS, IPO 设定值（实数） \$AA_IM[S1] 可读取测定主轴的实际值。取决于机床数据 \$MA_ROT_IS_MODULO 和 \$MA_DISPLAY_IS_MODULO 的设置，主轴和回转轴会进行相应的模数计算。
\$AA_OSCILL_BREAK_POS1	往复轴向换向点 1 移动时外部触发了换向而导致的制动位置
\$AA_OSCILL_BREAK_POS2	往复轴向换向点 2 移动时外部触发了换向而导致的制动位置
\$AC_TIME	从程序段起点起经过的时间，包含了内部生成过渡程序段的时间，单位：秒（实数）
\$AC_TIMES	从程序段起点起经过的时间，不包含内部生成过渡程序段的时间，单位：秒（实数）
\$AC_TIMEC	从程序段起点起经过的时间，包含了内部生成过渡程序段的周期数，单位：插补周期（实数）
\$AC_TIMESC	从程序段起点起经过的时间，不包含内部生成过渡程序段的周期数，单位：插补周期（实数）
\$AC_DTBB	从 BCS 程序段起点起经过的行程 Distance to begin, baseCoor（实数）
\$AC_DTBW	从 PCS 程序段起点起经过的行程 Distance to begin, workpieceCoor（实数）
\$AA_DTBB[<轴表达式>]	从 BCS 程序段起点起经过的单轴行程 Distance to begin, baseCoor（实数）
\$AA_DTBW[<轴表达式>]	从 PCS 程序段起点起经过的单轴行程 Distance to begin, baseCoor（实数）
\$AC_DTEB	距离 BCS 程序段终点的行程 Distance to end, baseCoor（实数）

\$AC_DTEW	距离 PCS 程序段终点的行程 Distance to end, workpieceCoor (实数)
\$AA_DTEB[<轴表达式>]	距离 BCS 运动终点的单轴行程 Distance to end, baseCoor (实数)
\$AA_DTEW[<轴表达式>]	距离 PCS 运动终点的单轴行程 Distance to end, workpieceCoor (实数)
\$AC_PLTBB	从 BCS 程序段起点起经过的轨迹行程 Path Length from begin, baseCoor (实数)
\$AC_PLTEB	距离 BCS 程序段终点的轨迹行程 Distance Path Path Length, baseCoor (实数)
\$AC_VACTB	BCS 轨迹速度 (Velocity actual, baseCoor) (实数)
\$AC_VACTW	PCS 轨迹速度 (Velocity actual, workPieceCoor) (实数)
\$AA_VACTB[<轴表达式>]	BCS 单轴速度 (Velocity actual, baseCoor) (实数)
\$AA_VACTW[<轴表达式>]	PCS 单轴速度 (Velocity actual, workPieceCoor) (实数)
\$AA_DTEPB[<轴表达式>]	BCS 往复运动/进给运动的单轴剩余行程 Distance to end, pendulum, baseCoor (实数)
\$AA_DTEPW[<轴表达式>]	PCS 往复运动/进给运动的单轴剩余行程 Distance to end, pendulum, workpieceCoor (实数)
\$AC_DTEPB	BCS 往复运动/进给运动的轨迹剩余行程 (不是 P2) Distance to end, pendulum, baseCoor (实数)
\$AC_DTEPW	PCS 往复运动/进给运动的轨迹剩余行程 (不是 P2) Distance to end, pendulum, workpieceCoor (实数)
\$AC_PATHN	标定后的轨迹参数: Path-Parameter normalized (实数) 0 (程序段开头) 到 1 (程序段末尾)

11.6 数据表

\$AA_LOAD[<轴表达式>]	驱动负载率
\$AA_POWER[<轴表达式>]	驱动有功功率，单位：W
\$AA_TORQUE[<轴表达式>]	驱动转矩设定值，单位：Nm
\$AA_CURR[<轴表达式>]	驱动电流实际值
\$AC_MARKER[<算数表达式>]（初始化）	标志变量：可应用在同步动作中构成复杂条件： 一用有 8 个标志（下标 0 到 7）可使用。 复位时，这些标志会置 0。 示例： WHEN DO \$AC_MARKER[0]=2 WHEN DO \$AC_MARKER[0]=3 WHEN \$AC_MARKER[0]==3 DO \$AC_OVR=50 也可以独立于同步动作在零件程序中单独读/写： IF \$AC_MARKER == 4 GOTOF SPRUNG
\$AC_PARAM[<算数表达式>]（实数）	用于同步动作的浮点参数。用于缓存和计算同步动作。 一共有 50 个参数（下标 0 到 49）可使用。
\$AA_OSCILL_REVERSE_POS1 [<轴表达式>]（实数）	
\$AA_OSCILL_REVERSE_POS2 [<轴表达式>]（实数）	往复运动的当前换向点 1 和 2： 分别从设定数据 \$SA_OSCILL_REVERSE_POS1 或 \$SA_OSCILL_REVERSE_POS2 读取数值。 因此在往复运动进行期间，也可以修改设定数据中的换向点，即：修改会立即进入同步条件。

条件

运动同步条件的句法为：

主处理变量 比较符 表达式

详细参见：

文档：

功能手册之同步动作分册

R2:回转轴

12.1 简要说明

机床上的回转轴

目前在很多机床上都使用回转轴。回转轴可用于刀具或刀具的旋转定位、完成辅助运动或满足工艺或运动方面的其他要求。

回转轴的典型应用是 5 轴铣床。在该机床上只有借助回转轴，刀尖才能对准工件的任意一个位置。

机床型号不同，对回转轴的要求也各有不同。为使数控系统满足不同机床的要求，可借助机床数据或特殊编程来激活各个回转轴功能。

回转轴通常用度数来编程。它的主要特点是旋转一圈后（360°模数）仍然回到同一个位置。根据不同应用，回转轴的工作区域可限制在 360°内，比如：刀架的回转轴；也不受限制地回转，比如：刀具或工件的旋转轴。

在很多方面，回转轴的工作特性和线性轴基本相同。本功能手册仅说明回转轴相对于线性轴的不同特性。

回转轴的定义

通过以下轴专用的机床数据可以将轴定义为“回转轴”：

MD30300 \$MA_IS_ROT_AX

几何轴是线性轴。如果将几何轴定义为回转轴，系统会输出报警 4200（几何轴不允许定义为回转轴），拒绝该定义。

只有在将一根轴定义为回转轴后，才可以执行并使用下文说明的功能，比如：无限工作区域、模数显示法。

可以同时多根轴定义为回转轴。

回转轴的类型

取决于应用的需求，回转轴的工作区域可以毫无限制，即：双向全范围旋转(MD30310 \$MA_ROT_IS_MODULO = 1)，或者通过软件限位开关限定在特定范围内，比如：0°到60°，或者限定主轴旋转圈数，比如：1000°。

下文列举了几个典型的重转轴应用。

典型应用

- 5 轴加工（工作区域不受限制或受限）
- 非圆加工（工作区域不受限制）
- 外圆/内圆磨削和成形磨削（工作区域不受限制）
- TRANSMIT 中的 C 轴（工作区域不受限制）
- 卷曲机（工作区域不受限制）
- 滚齿机上的工件回转轴 C 轴（工作区域不受限制）
- 圆形刀库和刀塔（工作区域不受限制）
- 柱面转换（工作区域不受限制）
- 机械手（工作区域 360°）
- 平面回转轴（工作区域 < 360°，比如：60° 以内）
- 滚齿机上的铣刀回转轴 A 轴（工作区域：90° 以内）

轴地址

DIN 标准规定了数控机床上坐标轴的名称和各运动方向。

在 DIN 66025 中，回转轴的名称如下定义：

- A 轴、B 轴 和 C 轴、中心轴为 X 轴、Y 轴和 Z 轴
分别为：A 轴绕 X 轴旋转、B 轴绕 Y 轴旋转、C 轴绕 Z 轴旋转（见下图）。
- 回转轴的正转如下定义：顺着中心轴正向观察，回转轴顺时针旋转（见下图）。

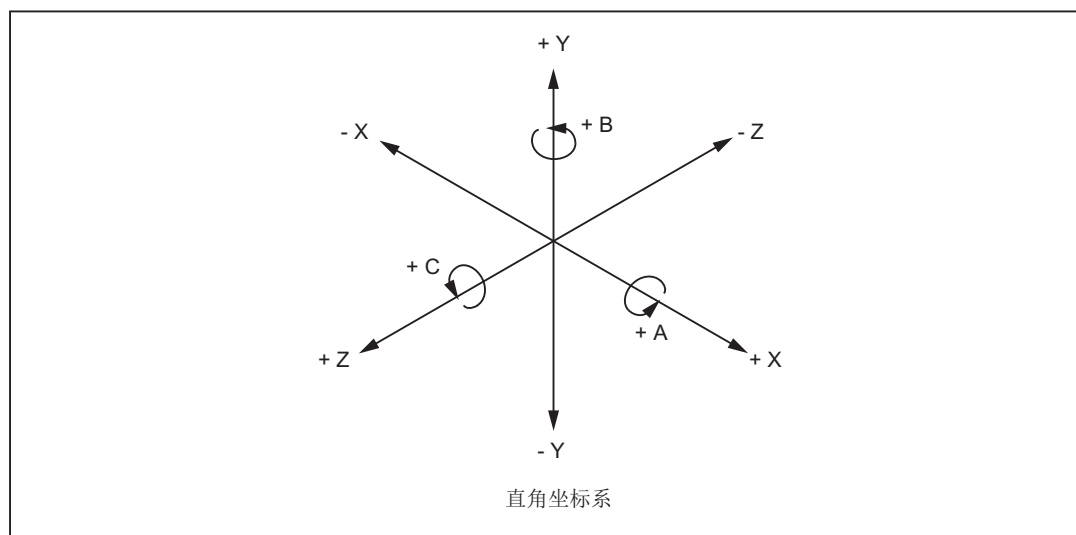


图 12-1 回转轴名称和运动方向

如需使用更多回转轴，可使用扩展轴名（比如：C2=）或者自定义轴名。

说明

机床数据 MD20050 \$MC_AXCONF_GEOAX_ASSIGN_TAB（为几何轴指定通道轴）必须为回转轴应用经过适当的调整。

尺寸单位

回转轴的缺省输入/输出单位如下：

回转轴的单位	
物理单位	单位
角度位置	度
程序写入的角速度	度每分
角速度机床数据	1) 转每分
角加速度机床数据	1) 转每秒 ²
角加加速度机床数据	1) 转每秒 ³

- 1) 在将轴定义为回转轴后，控制系统就会将轴专用机床数据自动换算为上述单位。用户可以通过机床数据确定其他的输入/输出单位制。

文档:

功能手册之基本功能分册；章节“速度，设定值/实际值系统，闭环控制（G2）”

工作区域

工作区域可以通过轴专用机床数据/设定数据（软件限位开关和工作区域限制）定义。一旦激活了回转轴的模数换算功能（即 MD30310 \$MA_ROT_IS_MODULO = 1），相当于回转轴可以无限回转，软件限位开关和工作区域限制因此失效。

PLC 也可以通过以下接口信号为模数回转轴动态激活软件限位开关和工作区域限制（激活操作可由零件程序通过 M/H 功能触发）：

DB31, ... DBX12.4（激活模数限制）

NC 通过以下接口信号反馈：

DB31, ... DBX74.4（模数限制生效）

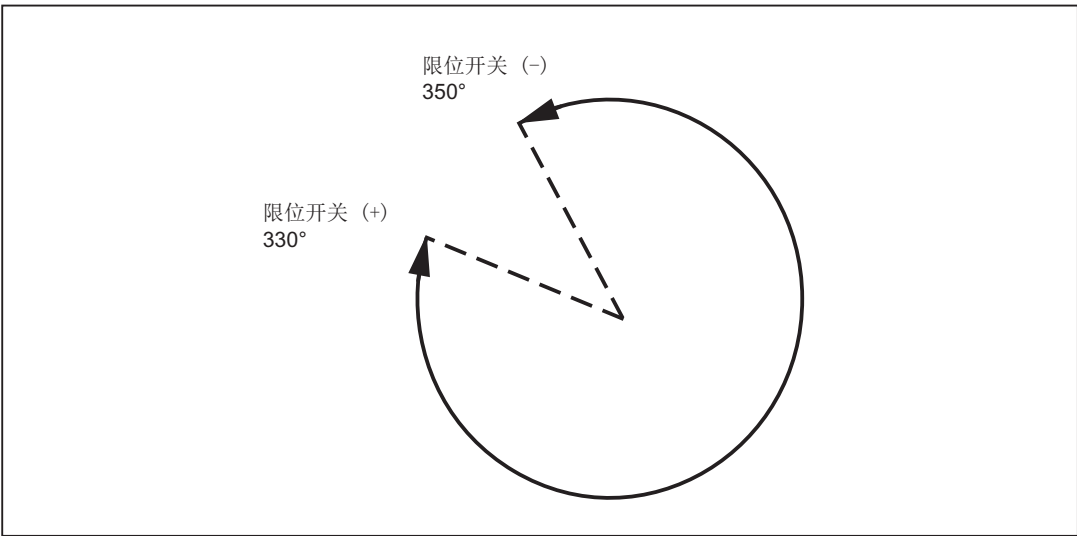


图 12-2 模数回转轴的受限工作区域

位置显示

回转轴的位置通常以 360° 模数显示：

MD30320 \$MA_DISPLAY_IS_MODULO = 1

进给率

对于回转轴而言，程序中写入的进给率 F 相当于角速度[度/分]。

如果在 G94 或 G95 中回转轴和线性轴要在同一条轨迹上运动，则必须用线性轴单位来编写回转轴进给率：比如：[毫米/分、英寸/分]。

回转轴的切线速度指的是在直径 D_E （ $D_E=360/\pi$ ）上的速度。如果直径等于单位直径（ $D=D_E$ ），则程序中写入的角速度（度/分）和切线速度（毫米/分）在大小上是相同的。

切线速度的计算公式为：

$$F = F_w * D / D_E$$

F = 切线速度[毫米/分]
 F_w = 角速度[度/分]
 D = F 有效时的直径[毫米]

其中 $D_E = 360 / \pi$ D_E = 单位直径[毫米]
 π = 圆周率 Pi

旋转进给率

在 JOG 运行方式下，进给轴/主轴特性也取决于下列设定数据的设置：

SD41100 \$SN_JOG_REV_IS_ACTIVE（JOG 运行方式中的旋转进给率有效）

SD41100 \$SN_JOG_REV_IS_ACTIVE	
生效	进给轴/主轴总是视主轴的情况而定采用以下旋转进给率： MD32050 \$MA_JOG_REV_VELO（JOG 旋转进给率） 或 MD32040 \$MA_JOG_REV_VELO_RAPID （JOG 旋转进给率叠加快速移动速度）。
未生效	进给轴/ 主轴的特性取决于设定数据： SD43300 \$SA_ASSIGN_FEED_PER_REV_SOURCE（定位轴/主轴的旋转进给率）
	若几何轴上有含旋转的有效框架命令，则该几何轴的特性取决于以下通道专用设定数据： SD42600 \$SC_JOG_FEED_PER_REV_SOURCE（JOG 运行方式下，有含旋转的有效框架命令的几何轴的旋转进给率）

12.2 360°模数

术语：360°模数

回转轴的编程通常需要以 360° 单位。使用这种模数特性的前提是将该轴定义为了回转轴。

回转轴上“模数”指的是系统自动将回转轴位置换算为 0° 到 359.999° 之间的某个位置。当目标位置超过 360° 时（比如：增量尺寸编程 G91），系统会自动将该位置换算到 0° 到 360° 之间。无论是 JOG 还是 AUTO 运行方式都支持该换算。例外：信息显示。

下图用一条螺旋线来展示回转轴在正向的绝对位置。螺旋线上的箭头标出了真实的绝对位置，比如 $C' = 420^\circ$ 。沿着圆弧追溯指针可以发现（条件是螺旋线和圆弧的 0° 位置相同），任意一个绝对位置都对应了一个 360° 的模数位置。在本例中， $C' = 420^\circ$ 经过取模后对应 $C = 60^\circ$ 。

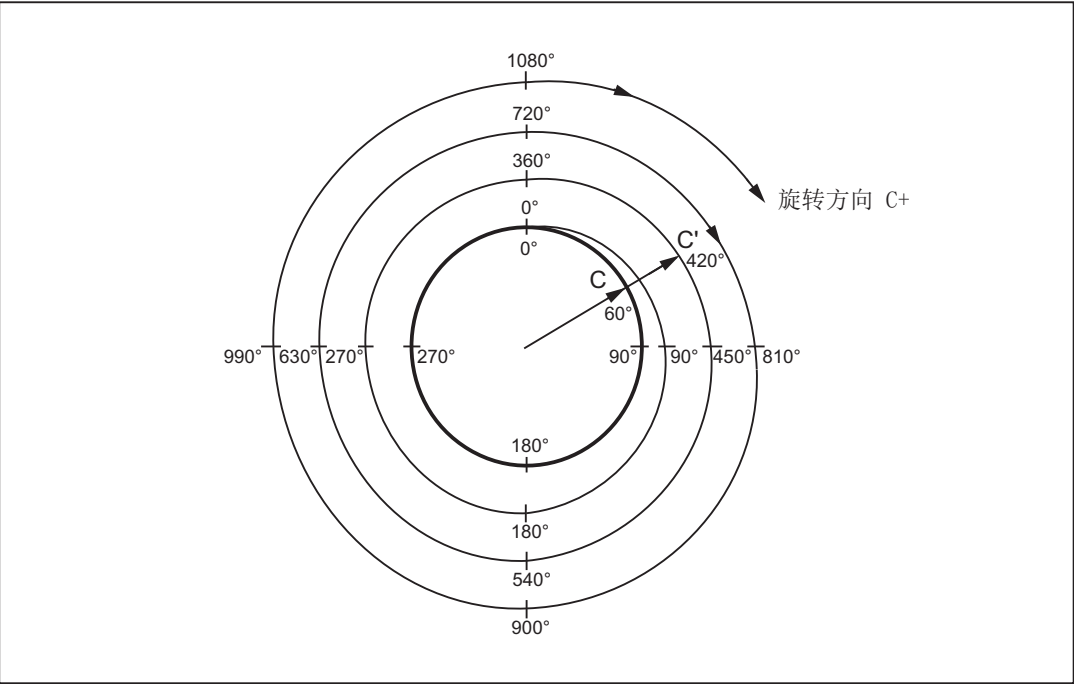


图 12-3 360°取模

机床数据设定

根据不同机床条件，可通过机床数据 MD30310 \$MA_ROT_IS_MODULO 为每根回转轴确定其编程、定位是否采用 360° 模数，通过机床数据 MD30320 \$MA_DISPLAY_IS_MODULO 确定其位置显示是否采用该模数。

模数轴

MD30310 \$MA_ROT_IS_MODULO = 1:

该机床数据生效后，回转轴具有一些特殊属性。该数据同时也定义了程序指令(G90, AC, ACP, ACN 或 DC)中回转轴的定位方式。在定位时，系统会考虑当前零点偏移的条件下进行 360° 模数换算。回转轴因此可在一圈内转动到**目标位置**。
软件限位开关和工作区域限制此时无效，因此回转轴可以**无限旋转**。

位置模数显示法

MD30320 \$MA_DISPLAY_IS_MODULO	
= 1	回转轴的位置通常要以一圈 360° 为单位加以显示，即：在回转轴正转时，一旦越过 359.999° 位置，系统内部就会周期性将位置恢复为 0.000°加以显示；在回转轴反转时，系统同样会将位置换算到 0° 到 359.999°内加以显示。
= 0	和模数 360°显示法相反，在绝对位置显示方式下，回转轴正转 1 圈后会显示 +360°，正转 2 圈后会显示 +720°，以此类推。此时，系统会根据线性轴设定来限制显示范围。

MD30320 \$MA_DISPLAY_IS_MODULO = 1

说明
如为回转轴激活了模数功能（(MD30310 \$MA_ROT_IS_MODULO = 1)，最好也选择“模数 360°显示法”。

模数回转轴的起点

可以为模数范围定义一个不为 0 的起点：
MD30340 \$MA_MODULO_RANGE_START（模数范围起点）
比如：MD30340 中输入 -180 后，模数范围便是 -180° 到 +180°。
默认值为 0，表明模数范围为 0° 到 360°。

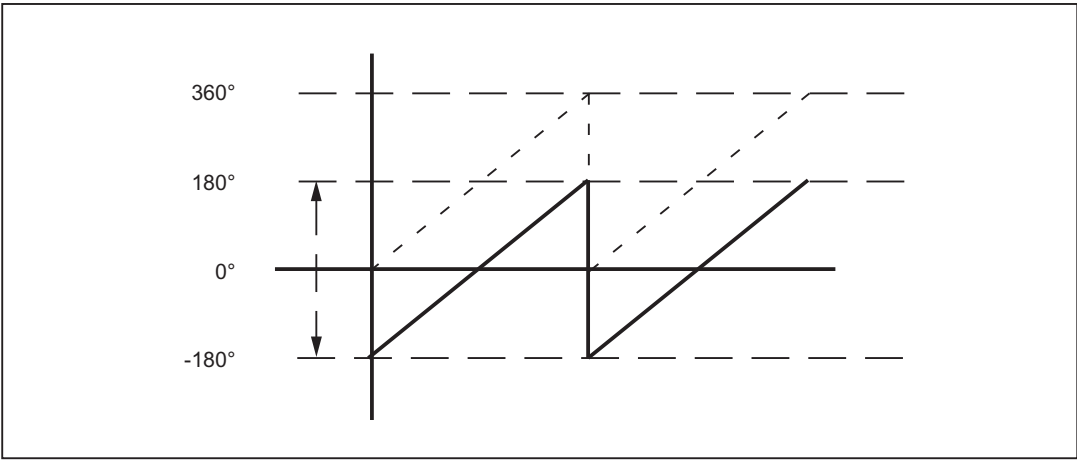


图 12-4 起点 -180° 将模数范围移至 -180° 到 + 180°。

应用

在模数分度轴上也可以用上述两个机床数据来确定分度位置，方式和确定模数范围一样，见章节“T1：分度轴 (页 811)”。

MD30503 \$MA_INDEX_AX_OFFSET

MD30340 \$MA_MODULO_RANGE_START

12.3 回转轴的编程

12.3.1 简介

说明

关于编程的一般说明参见

文档：

编程手册之基本原理分册

MD30310

通过轴专用机床数据:

MD30310 ROT_IS_MODULO (回转轴模数换算)

可确定在编程和定位方面回转轴是和定位轴完全一样还是有特殊方式。

下文将介绍回转轴的特殊编程方式以及两者区别 (主要是在绝对坐标编程方面)。

12.3.2 激活模数功能的回转轴 (不受限制的回转轴)

激活模数功能

→ 设置 MD30310 \$MA_ROT_IS_MODULO = 1

说明

建议将模数回转轴的位置显示也一同设为“360° 模数显示法”，即设置 MD30320 \$MA_DISPLAY_IS_MODULO = 1。

绝对位置编程(AC,ACP,CAN,G90)

定位轴举例: POS[轴名]=ACP(值)

- 值表示回转轴在 0 到 359.999°范围内的目标位置。
其中也允许输入负值,条件是实现在以下机床数据中修改了模数范围:
MD30340 \$MA_MODULO_RANGE_START
MD30330 MA_MODULO_RANGE
- 指定 ACP 和 ACN 指定了回转轴的运动方向:前者指定正转,后者指定反转,该方向和实际位置无关。
- 而指令 AC 或 G90 不指定运动方向,此时回转轴运动方向由实际位置决定。如果目标位置大于实际位置,则回转轴正转,否则反转。
定位方式可通过以下机床数据定义:
MD30455 \$MA_MISC_FUNCTION_MASK 位 2
位 2 = 0: G90 中模数轴默认以 AC 定位
位 2 = 1: G90 中模数轴以 DC “最短行程” 定位
- ACP 和 ACN 的应用:在不对称工件的加工中,由于指定了明确的运动方向,回转轴运动时可避免碰撞。

12.3 回转轴的编程

示例:

C 轴起点为 0°（见下图）。

①	POS[C] = ACP(100)	回转轴 C 轴正转到位置 100°
②	POS[C] = ACN(300)	C 轴反转到位置 300°
③	POS[C] = ACP(240)	C 轴正转到位置 240°
④	POS[C] = AC (0)	C 轴反转到位置 0°

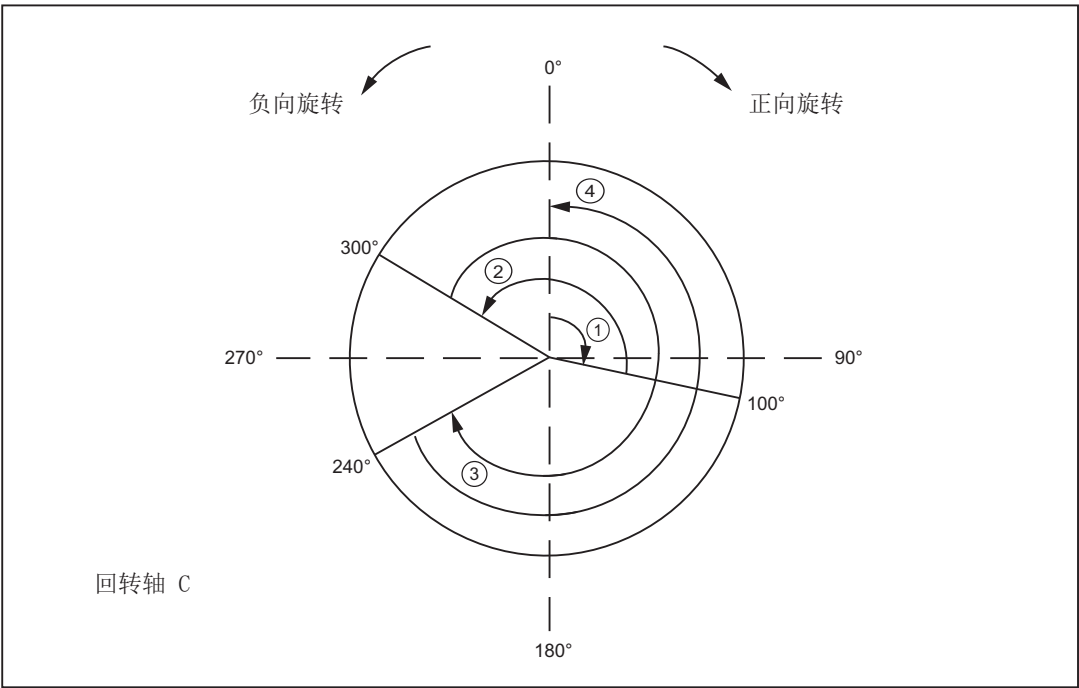


图 12-5 模数轴的绝对位置编程示例

以最短行程定位的绝对位置编程(DC)

POS[轴名] = DC(值)

- 值表示回转轴在 0 到 359.999°范围内的目标位置。如果其中写入负位置或 $\geq 360^\circ$ 的位置，则系统会输出报警 16830“编程了错误的模数位置”。
- 指令 DC (Direct Control)指定回转轴以**最短行程**在一圈内转动到编程的绝对位置（运行范围最大为 $\mp 180^\circ$ ）。

- 根据当前的实际位置，控制系统计算出旋转方向和剩余行程。如果剩余行程在两个方向都相同(180°)，则首先选择正转。
- DC 的应用举例：回转工作台要在最短时间内（即最短行程）转动到切换位置上。
- 如果为线性轴编程了 DC，则系统会输出报警 16800“不能使用运动指令 DC”。

示例：

C 轴起点为 0°（见下图）。

①	POS [C] = DC (100)	C 轴以最短行程转到位置 100°
②	POS [C] = DC (300)	C 轴以最短行程转到位置 300°
③	POS [C] = DC (240)	C 轴以最短行程转到位置 240°
④	POS [C] = DC (60)	C 轴以最短行程转到位置 60°。 如果剩余行程在两个方向都相同(180°)，则首先选择正转。

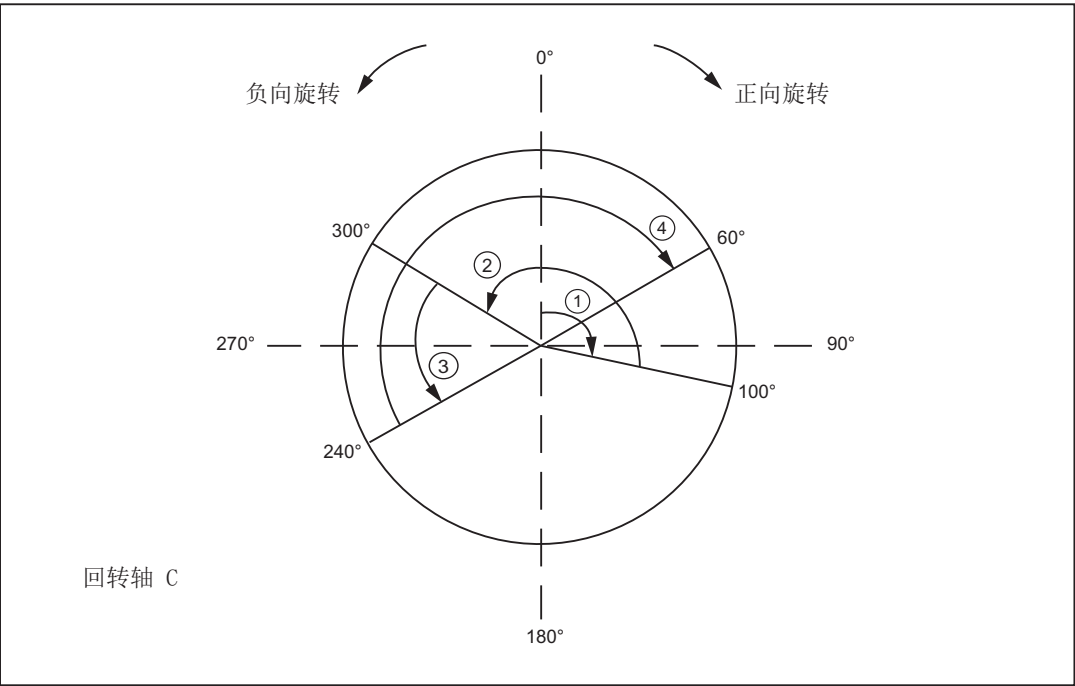


图 12-6 DC 编程示例

程序段搜索响应

在结束了一个含计算的程序段搜索后，可通过系统变量 `$AC_RETPOINT` 查看记录下的模数断点位置。

该系统变量提供的是模数位置。

含计算的程序段搜索结束后异步子程序要满足的前提条件：

在这种情形以及跨通道的程序段搜索 `SERUPRO` 中，必须在零件程序中执行在搜索中模拟的模数转换。

带/不带工作区域限制的模数回转轴

PLC 可置位以下接口信号，来动态激活/取消模数回转轴的软件限位开关/工作区域限制，激活/取消方式和对普通回转轴的一样：

`DB31, ... DBX12.4`

NC 通过以下接口信号反馈限位的当前状态：

`DB31, ... DBX74.4`

PLC 置位了接口信号 `DB31, ... DBX12.4` 后，监控功能激活。

触发 PLC 置位接口信号的 M 指令或 H 指令后必然跟随一个 `STOPRE` 指令，以便通过同步确保只监控切换后的程序段。

边界条件：

PLC 接口对软件限位开关监控的控制只针对模数轴。

模数轴上的运行范围监控始终要求轴已回参考点和一对生效的限位开关。

软件限位开关通常只能成对激活或取消。为正确实现对加工区域的监控，必须同时激活一对**软件限位开关**，途径有：`G26/G25` 或设定数据：

`SD43400 $SA_WORKAREA_PLUS_ENABLE`

与

`SD43410 $SA_WORKAREA_MINUS_ENABLE。`

示例：切换工作区域限制

在一个模数回转轴上，需要依次加工夹装了多个工件的托盘。之后必须用旋转轴更换托盘，该轴的工作区域必须加以限制，以防止损坏电缆。

配置：

`MD30300 $MA_IS_ROT_AX[AX4] = 1`

MD30310 \$MA_ROT_IS_MODULO[AX4] = 1

MD36110 \$MA_POS_LIMIT_PLUS[AX4] = 340

MD36100 \$MA_POS_LIMIT_MINUS[AX4] = 350

零件程序节选:

程序代码	注释
M123	; 机械安装多夹具托盘
	从 PLC 为 B 轴激活软件限位开关
	DB35, DBX12.4=0
STOPRE	; 触发预处理停止
S1000 M3	
G4 F2	
G1 X0 Y300 Z500 B0 F5000	
CYCLE84(500,400,0,350,0,1,4,10,,0,500,1000)	; 钻削循环
Z500	
B90	
CYCLE84(500,400,0,350,0,1,4,10,,0,500,1000)	; 钻削循环
Z500	
B180	
CYCLE84(500,400,0,350,0,1,4,10,,0,500,1000)	; 钻削循环
Z500	
B270	
CYCLE84(500,400,0,350,0,1,4,10,,0,500,1000)	; 钻削循环
Z500	
G0 Z540 B0	
M124	; 机械安装多夹具托盘
	从 PLC 为 B 轴激活软件限位开关
	DB35, DBX12.4=1
	;
STOPRE	; 触发预处理停止
B270	

增量位置编程(IC, G91)

定位轴举例: $POS[\text{轴名}] = IC(+/-\text{值})$

- 该值定义了回转轴的剩余行程。它可以为负值或者 $\geq +/-360^\circ$ 。
- 值的符号强制定义了回转轴的旋转方向。
- 应用示例: 旋转几圈后铣出一个螺旋槽

示例:

12.3 回转轴的编程

编程	结果
POS [C] = IC (720)	C 轴正转 720° (2 圈)
POS [C] = IC (-180)	C 轴反转 180°

不受限的工作范围

一旦模数功能有效，工作范围将不受限制(软件限位开关无效)。通过适当编程回转轴可以不停地回转。

示例：

程序代码
LOOP: POS [C] = IC (720) GOTOB LOOP

12.3.3 没有激活模数功能的重转轴

取消模数功能

→ 设置 MD30310 \$MA_ROT_IS_MODULO = 0

绝对位置编程(AC, G90)

定位轴举例： POS[轴名] = AC (+/-值)

- 该值和符号明确定义了回转轴的目标位置。它可以≥ +/-360°。位置值受软件限位开关的限制。
- 回转轴旋转方向由系统根据回转轴的当前实际位置确定。
- 如果编程了 ACP 或 ACN，系统会输出报警 16810“不能使用运动指令 ACP” 或 16820“不能使用运动指令 ACN”。
- 应用示例：
回转轴和线性运动同步（凸轮齿轮）。因此不能越过指定终点。

示例：

编程	结果
POS[C] = AC (-100)	回转轴 C 轴转到位置 -100°; 旋转方向取决于出发位置
POS[C] = AC (1500)	回转轴 C 轴转到位置 1500°

以最短行程定位的绝对位置编程(DC)

POS[轴名] = DC(值)

即使回转轴未定义成模数轴，该轴可以使用 DC (Direct Control) 来定位。此时，其定位方式相当于模数轴。

- 值表示回转轴在 **0 到 359.999°范围（模数 360°）** 内的目标位置。如果输入负值或 $\geq 360^\circ$ 的值，则系统会输出报警 16830“编程了错误的模数位置”。
- 指令 DC(Direct Control)指定回转轴以**最短行程**在一圈内转动到编程的绝对位置（工作范围最大为 $\mp 180^\circ$ ）。
- 根据当前的实际位置（经过 360° 模数计算），控制系统计算出旋转方向和剩余行程。如果剩余行程在两个方向都相同(180°)，则首先选择正转。
- DC 的应用举例：回转工作台要在最短时间内（即最短行程）转动到切换位置上。
- 如果为线性轴编程了 DC，则系统会输出报警 16800“不能使用运动指令 DC”。

示例：

编程	结果
POS[C] = AC (7200)	回转轴 C 轴转到位置 7200°; 旋转方向取决于出发位置
POS[C] = DC (300)	回转轴 C 轴以最短行程转到 模数位置 300°。 此时 C 轴会反转到位置 60°，相当于位于绝对位置 7140° 上。
POS[C] = AC (7000)	回转轴 C 轴转到绝对位置 7000°; 此时 C 轴会反转到位置 140°

说明

在本例中最好一同激活 360° 模数位置显示法，即设置 MD30320
\$MA_DISPLAY_IS_MODULO = 1!

增量位置编程(IC, G91)

定位轴举例： **POS[轴名] = IC(+/-值)**

在编程增量位置时，回转轴的行程和模数轴一样。但其工作范围受到软件限位开关的限制。

- 该值定义了回转轴的剩余行程。
它可以为负值或者 $\geq \pm 360^\circ$ 。
- 值的符号强制定义了回转轴的旋转方向。

工作范围受限

工作范围受限，和线性轴一样。工作范围由一对软件限位开关定义。

12.3.4 回转轴的其他编程特殊性

偏移

指令 TRANS（绝对平移）和 ATRANS（附加平移）可用于回转轴。

比例

指令 SCALE 和 ASCALE 对回转轴无用，因为系统会自动按照 360° 计算位置。

实际值设定

指令 PRESETON 可用。

分度轴

参见章节“T1：分度轴 (页 811)”。

12.4 调试回转轴

步骤

回转轴的调试方式基本上和线性轴一样，除了几个特例。此处需要注意的是，一旦将轴定义为回转轴，即设置 MD30300 \$MA_IS_ROT_AX = 1，系统便会自动将轴专用机床数据换算为以下单位：

位置	单位：度
速度	单位：转每分
加速度	单位：转每平方秒
加加速度限制	单位：转每立方秒

特殊机床数据

另外还需要根据应用的需求，设置回转轴机床数据：

MD30310 \$MA_ROT_IS_MODULO	定位和编程采用模数
MD30320 \$MA_DISPLAY_IS_MODULO	位置显示采用模数
MD10210 \$MN_INT_INCR_PER_DEG	角度位置计算精度

下表列出了各回转轴机床数据的组合方式：

回转轴机床数据的组合方式				
MD30300	MD30310	MD30320	允许使用	注释
0	0	0	允许	轴是线性轴（缺省值）
1	0	0	允许	轴是回转轴，定位时不采用模数，即：软件限位开关有效，位置显示为绝对位置
1	0	1	允许	轴是机床轴，定位时不采用模数，即：软件限位开关有效，位置显示为模数位置； 应用：工作区域在 $\pm 1000^\circ$ 之间的轴。

12.5 回转轴的特殊性

回转轴机床数据的组合方式				
1	1	1	允许	轴是回转轴，定位时采用模数，即：软件限位开关无效，工作区域不受限制；位置显示为模数位置（回转轴的最常见设置）；工作区域限制可由 PLC 动态激活。
1	1	0	允许	轴是回转轴，定位时采用模数，即：软件限位开关无效，工作区域不受限制；位置显示为绝对位置；工作区域限制可由 PLC 动态激活。
0	0 到 1	0 到 1	不推荐	轴不是回转轴；系统因此忽略其他机床数据

回转轴的 JOG 速度

SD41130 \$SN_JOG_ROT_AX_SET_VELO（回转轴的 JOG 速度）

通过上述设定数据可为所有回转轴统一设定有效的 JOG 速度，另见章节“H1：手动运行和手轮运行 (页 155)”。

如果该数据为 0，则轴专用机床数据就用作回转轴的 JOG 速度：

MD21150 \$MC_JOG_VELO（基准 JOG 轴速度）

12.5 回转轴的特殊性

软件限位开关

通常软件限位开关和工作区域限制都要生效，用于限制轴的工作区域。但是在旋转不受限的回转轴上，即 MD30310 \$MA_ROT_IS_MODULO = 1，软件限位开关和工作区域限制变为无效。

不管模数轴有或没有工作区域限制，都可投入使用。

文档：

功能手册之基本功能分册；章节“轴监控，保护区 (A3)”

回转轴的镜像

指令 MIRROR (C) 或 AMIRROR (C) 可对回转轴进行镜像。

回参考点

文档：

功能手册之基本功能分册；章节“回参考点（R1）”

主轴设为回转轴

关于如何将主轴设为回转轴（即所谓的 C 轴）的说明参见：

文档：

功能手册之基本功能分册；章节“主轴（S1）”

12.6 示例

叉头、倾斜回转头

在 5 轴铣床上需要频繁使用回转轴会回转刀具或者工件。该机床可以使刀尖对准工件的任意一个点，可将刀具中心轴调整到任意一个位置。应用不同，需要使用的铣头也不同。下图展示了回转轴的典型应用：叉头和倾斜回转头。

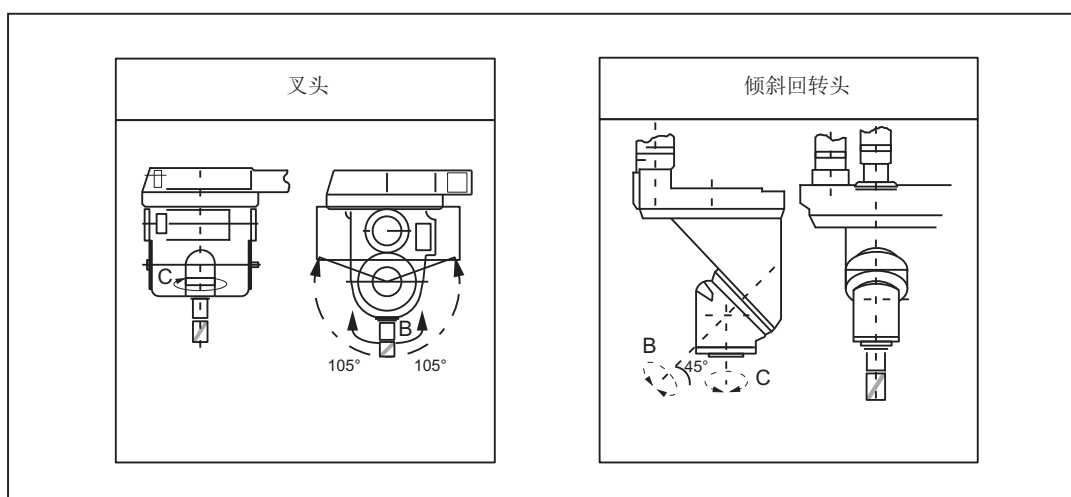


图 12-7 叉头、倾斜回转头

12.7 数据表

12.7.1 机床数据

12.7.1.1 通用机床数据

编号	名称: \$MN_	描述
10210	INT_INCR_PER_DEG	角度位置计算精度

12.7.1.2 进给轴/主轴专用机床数据

编号	名称: \$MA_	描述
30300	IS_ROT_AX	轴是回转轴
30310	ROT_IS_MODULO	模数回转轴
30320	DISPLAY_IS_MODULO	实际值以模数显示
30330	MODULO_RANGE	模数范围
30340	MODULO_RANGE_START	模数范围起点
30455	MISC_FUNCTION_MASK	轴功能
36100	POS_LIMIT_MINUS	负向软件限位开关
36110	POS_LIMIT_PLUS	正向软件限位开关

12.7.2 设定数据

12.7.2.1 通用设定数据

编号	名称: \$SN_	描述
41130	JOG_ROT_AX_SET_VELO	回转轴的 JOG 速度

12.7.2.2 进给轴/主轴专用设定数据

编号	名称: \$SA_	描述
43420	WORKAREA_LIMIT_PLUS	工作区极限正
43430	WORKAREA_LIMIT_MINUS	工作区极限负

12.7.3 信号

12.7.3.1 到进给轴/主轴的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
模数轴的工作范围限制	DB31, ...DBX12.4	DB380x.DBX1000.4

12.7.3.2 来自进给轴/主轴的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
模数轴上软件限位开关监控的状态	DB31, ...DBX74.4	DB390x.DBX1000.4

S3:主轴同步

13.1 简要说明

13.1.1 功能

通过“主轴同步”功能可使 2 个主轴的位置同步或转速同步。其中一个主轴定义为主主轴 (LS)，另一个定义为副主轴(FS)。

转速同步性: $n_{\text{副主轴}} = k_{\text{传动比}} * n_{\text{主主轴}}$ 其中 $k_{\text{传动比}} = 1、2、3、...$

轴

位置同步性: $\varphi_{\text{副主轴}} = \varphi_{\text{主主轴}} + \Delta\varphi$ 其中 $0^\circ \leq \Delta\varphi < 360^\circ$

应用

背面加工

其中一种应用是在双主轴车床上进行背面加工，即：在主主轴和副主轴位置同步后，将工件从主主轴快速传送至副主轴（或相反），且主主轴和副主轴无须处于静止状态。

多面体加工（多面体车削）

“主轴同步”功能中，可设置主主轴和副主轴之间的整数传动比： $k_{\text{传动比}}$ ，以此来实现多面体加工。

副主轴数量

与主主轴同步运行的副主轴的数量只受所使用 NC 的性能限制。原则上，NC 通道中的任意多个副主轴可以同时和一个主主轴实现同步。

每个 NC 通道最多可以同时激活 2 个同步主轴对。

定义

既可以通过通道专用的机床数据来固定设置同步主轴对，也可以通过零件程序指令来灵活定义同步主轴对。

激活/撤销

主轴同步功能是通过零件程序指令来激活/撤销的。

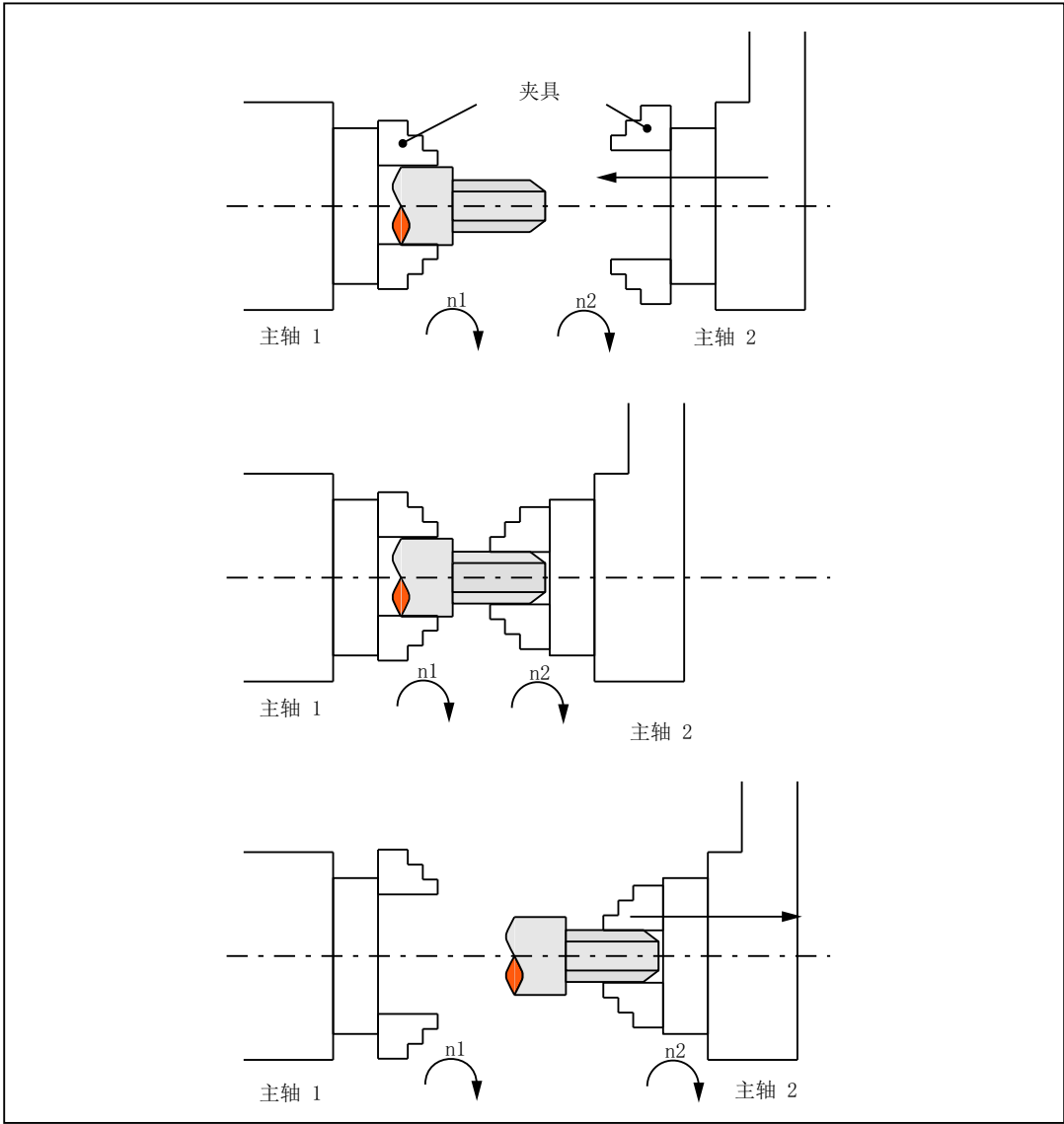


图 13-1 主轴同步：工件从主轴 1 快速传送至主轴 2

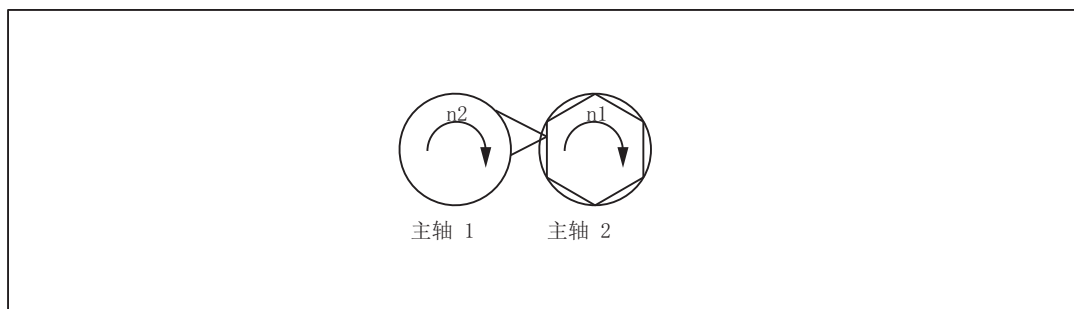


图 13-2 主轴同步：多面体车削

13.1.2 主轴同步

说明

<轴表达式>:	可以是： - 轴名称 - 主轴名称
<轴名称>:	C (主轴在进给轴方式中的名称为“C”。)
<主轴名称>:	Sn、SPI(n) 其中 n = 主轴号
<主轴号>:	1, 2, ... 为 MD35000 \$MA_SPIND_ASSIGN_TO_MACHAX 中确定的主轴号
(副主轴,主主轴,偏移):	LS = 主主轴、FS = 副主轴、偏移 = 系统变量读取的副主轴的可编程偏移
\$P_COUP_OFFS[Sn]	程序中编写的同步主轴位置偏移

同步主轴对

在主轴同步中有一个副主轴(FS)和一个主主轴(LS)，即**同步主轴对**。在激活主轴同步时，副主轴根据确定的函数关系跟随主主轴。

主轴同步

主轴同步(也称为同步主轴方式)是另一种主轴运行方式。激活主轴同步前，副主轴必须转换到位置闭环控制方式下。激活主轴同步后，副主轴和主主轴同步。撤销主轴同步后，副主轴进入开环控制方式。

一旦副主轴的主轴同步激活，NC 便会将以下接口信号报告给 PLC：

“主轴同步”(DB31, ... DBX84.4) = 1。

同步主轴数量

多个副主轴可以和一个主主轴同步。与该主主轴同步的副主轴的数量取决于相应的软件版本。

同一个 NCU 或另一个 NCU 通道中的任意多个副主轴可以和该主主轴同步。

对此需要注意的是：在所有主轴中，始终要有一根主主轴，因此同步主轴对的数量是主轴总数减一。

主轴同步方式

主轴同步具有以下功能：

- 副主轴和主主轴以相同的转速旋转
($n_{\text{副主轴}} = n_{\text{主主轴}}$ ；传动比 $k_{\text{传动比}} = 1$)
- 副主轴和主主轴可以以相同的方向或相反的方向旋转
($k_{\text{传动比}}$ 可确定为正或负)
- 副主轴和主主轴以不同的转速旋转
($n_{\text{副主轴}} = k_{\text{传动比}} \times n_{\text{主主轴}}$ ； $k_{\text{传动比}} \neq 1$)
应用：多面体车削
- 副主轴和主主轴之间的角度可定义($\varphi_{\text{副主轴}} = \varphi_{\text{主主轴}} + \Delta\varphi$)
主主轴和副主轴以转速同步的方式运行，同时存在指定的角度偏移(位置同步)。
应用：异型工件
- 既可以在运行状态，也可以在静止状态下激活副主轴和主主轴的同步运行。
- 主主轴可以使用开环控制方式和位置闭环控制的全部功能。
- 主轴同步未激活时，副主轴和主主轴可以在任意一种主轴运行方式中运行。
- 在主轴运动时激活主轴同步后也可以改变传动比。
- 可在主轴同步激活后更改副主轴和主主轴的偏移量(叠加运动)。

主轴同步方式

主轴同步的定义有：

- 通过通道专用的机床数据固定定义
(以下称为“固定定义的同步”)
- 通过零件程序中的语言指令(COUP...)灵活定义
(以下称为“用户定义的同步”)

这两种途径。两种途径有以下组合方式：

1. 一个同步主轴对由机床数据固定定义。第二个同步主轴对由零件程序灵活定义。
2. 没有通过机床数据定义的同步主轴对。对此，可通过零件程序对同步主轴对进行设置和用户定义。

独立的副主轴插补器

通过独立的**副主轴插补器**可以让来自不同通道或另一个 NCU 的多个副主轴按照用户自定义的方式与唯一的一个主主轴同步运行。副主轴插补器可通过

- COUPON 或 COUPONC 激活
- COUPOF 或 COUPOFS 撤销

副主轴插补器始终位于用 COUPON 和 COUPONC 指令编程副主轴的通道中。如果需要激活同步的副主轴已经在另一个通道中编程，COUPON/COUPONC 指令会生成一个跨通道取轴指令，从中取出该主轴，将它放入插补器所在的通道。

通过主轴同步专用的 VDI 接口信号，PLC 可以控制一些同步功能。但这些信号只对副主轴有效，对主主轴无效（参见章节“通过 PLC 控制主轴同步 (页 755)”）。

定义主轴同步

激活主轴同步之前必须确定需要同步的主轴(副主轴、主主轴)。

这可根据应用情况的不同按两种不同的方式进行：

1. 固定定义同步:
需要作为副主轴和主主轴生效的机床轴由通道专用的机床数据 MD21300 `$MC_COUPLE_AXIS_1[n]` 确定。
这种方式中确定的主主轴和副主轴无法由 NC 零件程序修改。
同步参数也可由 NC 零件程序修改。
2. 用户定义的同步:
使用语言指令 “COUPDEF(副主轴,主主轴,...)” 可以在 NC 零件程序中新建和修改同步。如果要定义一个新的同步关系, 必须在此之前删除现有的用户定义的同步, 即使用语言指令 COUPDEL(副主轴,主主轴)。
在每个 COUP... 语言指令中都应写入副主轴号和主主轴号。由此可以明确确定同步主轴对。有效的主轴号必须事先在以下轴专用机床数据中指定了对应的机床轴。
MD35000 `$MA_SPIND_ASSIGN_TO_MACHAX`。

NC 会通过对应机床轴的接口信号报告 PLC, 这些机床轴是否作为副主轴(DB31, ... DBX99.1) 和主主轴(DB31, ... DBX99.0)生效。

可选择通过零件程序、PLC 或同步动作对主主轴进行编程。

传动比

传动比用分子和分母（传动参数）来规定。这样便可以精确规定有理数形式的传动比。

通常情况下下列关系成立：

$$k_{\text{传动比}} = \text{分子} : \text{分母} = \text{传动比}_{\text{分子}} : \text{传动比}_{\text{分母}}$$

在控制系统内部, 传动参数的值范围(传动比_{分子}, 传动比_{分母})几乎不受限制。

对于通过机床数据定义的同步, 可以使用通道专用的 SD42300: COUPLE_RATIO_1[n] 来确定传动参数。此外, 还可通过语言指令 COUPDEF(副主轴,主主轴,传动比_{分子},传动比_{分母},...) 来修改传动比。设定数据中的值不可以改写(缺省值)。

对于通过 NC 零件程序定义的同步, 只可通过语言指令 COUPDEF (...)来设置传动比。

新的传动参数会在 COUPDEF 指令执行完毕后立即生效。

主轴同步功能的其他编程指令在“编程主轴同步”章节中有详细说明。

同步特性

可为每个主轴同步对设置以下特性：

- **程序段切换特性**

激活主轴同步时，或在主轴同步激活的条件下修改传动比或指定的角度偏移时，可确定何时切换到下个程序段。

- 立即进行程序段切换
- 在满足“精同步”条件时进行程序段切换
- 在满足“粗同步”条件时进行程序段切换
- 在满足 IPOSTOP 条件时进行程序段切换(即：在设定值同步后)
- 使用 WAITC 等待标记可在任意时间点检查同步条件。

- **副主轴和主主轴的同步方式**

既可使用主主轴的位置设定值作为副主轴的控制量，也可使用主主轴的位置实际值作为副主轴的控制量。因此可选择以下同步方式：

- **设定值同步(DV:设定值)**
在位置闭环模式中使用。此时，两根主轴的动态响应几乎一致。应优先使用这种同步方式。
- **实际值同步(AV:实际值)**
在没有主主轴位置闭环控制，或副主轴与主主轴的控制方式存在着巨大偏差时使用。副主轴的设定值从主主轴的实际值得出。当主主轴的转速不断变化时，实际值同步的效果比设定值同步的效果差。
- **速度同步(VV:速度值)**
速度同步位于设定值同步内部。它对副主轴和主主轴的要求较低。副主轴和主主轴无需位置闭环控制和测量系统。
副主轴和主主轴的位置偏移未经定义。

在**固定定义的同步**中是通过机床数据来定义同步特性的（参见章节“定义(页 768)”），而在**用户定义的同步**中是通过语言指令 COUPDEF 来定义同步特性的（参见章节“定义(COUPDEF)(页 761)”）。

此外,在固定定义的同步中，同步方式和程序段切换特性还可通过语言指令 COUPDEF 修改。

文档：

编程手册之工作准备分册(“主轴同步”)。

同步特性设有写保护

使用通道专用的 MD21340 \$MC_COUPLE_IS_WRITE_PROT_1 可以确定固定定义的同步的参数（传动比、同步方式以及程序段切换特性）是否受 NC 零件程序的影响：

0: 同步参数可由 NC 零件程序指令 COUPDEF 加以修改

1: 同步参数无法由 NC 零件程序修改，一旦尝试，系统会拒绝此次修改并发出报警信息。

叠加运动

主轴同步激活后，副主轴会按照指定的传动比跟踪主主轴。

同时，副主轴可以叠加运动，以修改定义的副主轴和主主轴之间的角度位置。

副主轴的叠加运动可以以不同的形式触发：

- **AUTO 和 MDA 模式下，副主轴的可编程位移：**
 - 使用语言指令 COUPON 和 SPOS 激活主轴同步后，可以修改副主轴和主主轴之间的相对位置（参见章节“通过零件程序来激活主轴同步 (页 752)”）。
- **副主轴的手动位移：**
 - 在 JOG 模式下（连续 JOG 或断续 JOG）
主轴同步激活时，使用手轮或正/负移动键给出副主轴的叠加运动量。
 - 在 AUTO 和 MDA 模式下
使用手轮通过 DRF 偏移给出副主轴的叠加运动量

一旦副主轴执行叠加运动，接口信号“叠加运动”(DB31, ... DBX98.4)会置为 1 信号。

在编写 COUPON 指令时，副主轴会以可能的最大速度进行叠加运动。用 SPOS 修改偏移后，还可以用 FA[Sn] 来指定定位速度以及用接口信号“主轴进给倍率生效”(DB31, ... DBX17.0)给出一个主轴倍率值。

说明

更多关于如何通过 FA[Sn] 来指定定位速度的信息请参考：

文档：

功能手册之基本功能分册：“主轴(S1)”章节之“主轴运行方式之定位方式”

设定值补偿

由系统变量 \$AA_COUP_CORR[Sn] 读出的设定值偏移如同一个位移值一样，会对副主轴后续的所有编程指令生效，相当于 MCS 中的 DRF 偏移。

补偿值测定实例

如果通过 COUPON(....,77) 指令写入了一个 7°的位移，而且由于工件夹具闭合而产生了 81°的机械位置偏移，系统会得出 4°的补偿值。

系统变量为副主轴提供以下值：

\$P_COUP_OFFS[S2]：程序中编写的位置偏移 = 77°

\$AA_COUP_OFFS[S2]; 设定位置偏移 = 77°

\$VA_COUP_OFFS[S2]; 实际位置偏移约 77°

\$AA_COUP_CORR[S2]; 补偿值 = 4°

13.1.3 主轴同步的前提条件

激活主轴同步的条件

只有具备以下条件时才可激活主轴同步，否则系统会发出报警信息。

- 必须事先确定好同步主轴对（要么通过机床数据进行固定定义，要么借助 COUPDEF 指令通过零件程序进行用户定义）。
- 激活主轴同步的 NC 通道必须知道哪些主轴需要进行同步。
通道专用 MD20070 \$MC_AXCONF_MACHAX_USED
轴专用 MD35000 \$MA_SPIND_ASSIGN_TO_MACHAX
- 激活主轴同步的 NC 通道中必须分到了副主轴。
轴专用的 MD30550 AXCONF_ASSIGN_MASTER_CHAN 的缺省设置
- 设定值同步和实际值同步(DV、AV)需要满足的特定条件：
主主轴和副主轴必须至少具备一个位置测量系统进行位置采集，且位置闭环控制必须事先调试完毕。

说明

位置闭环控制激活时，主主轴的最大设定转速会被限制在最大转速的 90%(控制裕量)以内。NC 会通过接口信号“设定转速受限”(DB31, ... DBX83.1)将这一状态报告给 PLC。

文档:

功能手册之基本功能分册；章节“主轴(S1)”

- 设定值同步需要满足的特定条件：
在激活主轴同步之前，应编写指令 SPCON 使主主轴进入位置闭环控制。
- 激活主轴同步之前必须选择合适的主主轴和副主轴齿轮档。主轴同步中无法切换主主轴和副主轴齿轮档，因此也无法进行往复运动。当你要求执行上述操作时，系统会发出报警。

跨通道同步

主主轴可位于任意一个通道中。

- 借助“跨通道取轴”功能，主主轴可以在通道之间来回切换。
- 当多个副主轴与一个主主轴同步时，整个同步组的动态响应取决于最慢的那个轴。主主轴的最大加速度和最大转速会适当降低，直到跟它同步的副主轴不会出现过载为止。
- 副主轴始终在用 COUPON 或 COUPONC 激活了主轴同步的通道中运行。

13.1.4 通过零件程序来激活主轴同步

使用 COUPON、COUPONC 激活同步

使用语言指令 COUPON 可在零件程序中激活指定主轴之间的同步，同步采用最后有效的参数。主轴同步可固定定义或用户定义。此时的主主轴和/或副主轴既可处于静止状态，也可处于运行状态。

必须具备以下条件才可以激活主轴同步（参见章节“主轴同步的前提条件 (页 751)”）。

语言指令 COUPONC 可使主轴继续采用零件程序中之前写入的方向和转速。但它不能写入角度偏移。

COUPON 编程方式

激活主轴同步有两种方式可选：

1. COUPON 指令可以尽快的激活主轴同步，但不设定主主轴和副主轴之间的 **角度偏移**。
COUPON(副主轴,主主轴)
2. 使用 COUPON 指令激活主轴同步，同时主主轴和副主轴之间存在**指定的角度偏移**：位置_{副主轴}。选择这种方式时还须写入角度偏移。
COUPON (副主轴、主主轴、位置_{副主轴})

程序段切换特性

激活主轴同步之前，应确定使用哪种程序段切换条件（参见章节“定义 (COUPDEF) (页 761)”）。

确定当前的同步状态

通过轴系统变量 \$AA_COUP_ACT[<轴表达式>]可以在零件程序中确定进给轴/主轴的当前同步状态（参见章节“主轴同步的轴系统变量 (页 765)”）。一旦副主轴的同步激活，位 2 就置“1”。

修改指定的角度偏移

使用语言指令 COUPON 和 SPOS 可在主轴同步激活时修改指定的角度偏移。作为叠加运动，副主轴定位到以“位置_{副主轴}”写入的角度偏移位置。在此期间接口信号“叠加运动”(DB31, ... DBX98.4)会置位。

角度偏移“位置_{副主轴}”

指定的角度偏移“位置_{副主轴}”是个绝对位置，是在正向上和主主轴零度位置相差的一个位置值。

位置闭环控制中主轴的“0°位置”可以通过以下方式得出：

- 由测量系统的零脉冲信号或开关信号得出以及
- 由轴专用机床数据定义的基准值得出：
MD34100 \$MA_REFP_SET_POS、参考点值、
在间距编码系统中无任何作用。
MD34080 \$MA_REFP_MOVE_DIST 参考点间距/目标点
在间距编码系统中，
MD34090 \$MA_REFP_MOVE_DIST_CORR 参考点偏移/间距编码的绝对偏移。

位置_{副主轴}的范围：0 ... 359.999°。

文档：

功能手册之基本功能分册；章节“回参考点(R1)”

读取当前的角度偏移

通过轴系统变量可以在 NC 零件系统中读取副主轴与主主轴的当前位置偏移。其中可分为：

- 副主主与主主轴的设定值位置偏移
\$AA_COUP_OFFS [<副主轴名称>]
- 副主主与主主轴的实际值位置偏移
\$VA_COUP_OFFS [<副主轴名称>]

(<轴名称>说明，参见章节“主轴同步 (页 745)”))

上电后激活

也可以在副主轴或主主轴还未回参考点/未同步时，即接口信号“已回参考点/已同步 1 或 2” (DB31, ... DBX60.4 或 DBX60.5 = 0) 复位时激活主轴同步。出现这种情况时系统会发出报警。

示例:

上电后，副主轴与主主轴已经通过一个工件固定在一起。

13.1.5 通过零件程序来撤销主轴同步**使用 COUPOF 或 COUPOFS 撤销同步**

使用语言指令 COUPOF 可以撤销指定主轴之间的同步。有三种方式可选。

使用指令 COUPOF 可以撤销指定主轴之间的同步，不论该同步是固定定义还是用户定义的。此时的主主轴和副主轴即可处于静止状态，也可处于运行状态。

使用 COUPOF 指令撤销同步时，副主轴会切换至 **开环控制方式**。之前编写的转速值（S 字）不再对副主轴有效，副主轴可以像一个普通的主轴一样运行。

使用 COUPOF 撤销同步时，系统会自动生成一个预处理停止指令，功能与 STOPRE 类似。

使用指令 COUPOFS 既可以无需达到指定位置就快速撤销同步，也可以到了指定位置时再撤销同步。

COUPOF 方式

使用 COUPOF 撤销同步有三种方式可选：

1. 快速撤销同步。
立即使能程序段切换。
COUPOF(副主轴,主主轴)
2. 等副主轴越过指定位置“位置_{副主轴}”时再撤销同步。
然后使能程序段切换。
COUPOF(副主轴,主主轴,位置_{副主轴})
3. 等副主轴和主主轴都越过指定位置“位置_{副主轴}”和“位置_{主主轴}”时再撤销同步。
然后使能程序段切换。
COUPOF(副主轴,主主轴,位置_{主主轴},位置_{主主轴})

位置_{副主轴},位置_{主主轴}

“位置_{副主轴}”或“位置_{主主轴}”是副主轴和主主轴相对于参考点的实际位置。

“位置_{副主轴}”或“位置_{主主轴}”:0 ... 359.999°。

文档:

功能手册之基本功能分册；章节“回参考点(R1)”

在运行过程中使用 COUPOF 指令

使用 COUPOF 指令可以在运行过程中撤销同步，而副主轴会以当前的转速继续旋转($n_{\text{副主轴}}$)。通过系统变量 \$AA_S 可以在零件程序中读取转速。

撤销同步后，从零件程序中使用 M05、SPOS、SPOSA 或相应的 PLC 接口信号可以使副主轴停止。

COUPOFS 自动停止副主轴

该指令增加了一个副主轴停止的特性：

- 无需达到指定位置即尽快撤销同步，并停止副主轴
然后使能程序段切换。

COUPOFS(副主轴,主主轴)

- 尽快的撤销同步，同时在指定的位置上停住副主轴。然后使能程序段切换。

边界条件：

主轴同步未激活时，COUPOFS(副主轴,主主轴)和 COUPOFS(副主轴,主主轴,位置_{副主轴})指令没有任何意义。

13.1.6 通过 PLC 控制主轴同步

副主轴由 PLC 控制

通过同步专用的轴 VDI 接口信号可以从 PLC 程序中控制副主轴的同步运动。其中 PLC 可以禁止、抑制或重新激活通过编写了偏移的副主轴同步动作。

这些信号对主主轴没有任何的作用。可以使用以下同步专用的 VDI 信号(PLC → NCK)：

“禁用同步”(DB31, ... DBX31.5)

“禁用同步动作”

使用轴接口信号 “禁用同步”(DB31, ... DBX31.5)可以抑制副主轴的同步动作。

当包含 COUPON (副主轴,主主轴,偏移)零件程序指令进入主处理时，NC 会分析副主轴的以下接口信号：

“禁用同步”(DB31, ... DBX31.5)。

13.1 简要说明

- 当“禁用同步”(DB31, ... DBX31.5) = 0 时，位置偏移如同之前一样生效。
- 当“禁用同步”(DB31, ... DBX31.5) = 1 时，同步只以恒速进行。副主轴不会有额外的运动。
随后的主轴同步与编写指令 COUPON (<副主轴>, <主主轴>) 类似。

特殊性

“禁用同步”(DB31, ... DBX31.5)不影响通过以下方式生成的偏移运动：

- SPOS、POS
- 同步动作
- FC18 (在 840D sl 中)
- JOG

这些功能可通过 VDI 接口信号“进给停止/主轴停止”(DB31, ... DBX4.3)来控制。

达到同步

一旦达到同步，以下两个 VDI 信号都会置位，不管同步是否禁止：

“粗同步”(DB31, ... DBX98.1)以及

“精同步”(DB31, ... DBX98.0)

虽然同步被抑制，但 COUPON 这个指令后续的程序段切换不会被阻止。

示例

COUPON 指令后续程序段切换的特性

程序代码	注释
N51 SPOS=10 SPOS[2]=10	; 接口信号“禁用同步动作” ; (DB31, ... DBX31.5) = 1, 用于 S2 ; 位置等于偏移量: ; 0°
N52 COUPDEF(S2,S1,1,1,"FINE","DV")	
N53 COUPON(S2,S1,77)	; 保留 0 度的实际偏移 ; 副主轴不运动 ; VDI 信号 ; 接口信号“粗同步” ; (DB31, ... DBX98.1) 以及 ; 接口信号“精同步” ; (DB31, ... DBX98.0)

程序代码	注释
	; 置位且程序段切换
	; 使能。
N54 M0	
N57 COUPOF(S2,S1)	
N99 M30	

复位及重新激活

复位接口信号“禁用同步”(DB31, ... DBX31.5)不会对副主轴偏移产生影响。如果之前通过该 VDI 接口信号抑制了副主轴的偏移运行，那么该信号复位后，偏移并不会自动生效。

可通过以下方式重新激活同步：

- 在接口信号“禁用同步”(DB31, ... DBX31.5)置 0 时，再次写入零件程序指令 COUPON (副主轴,主主轴,偏移)。
比如可以将 COUPON (副主轴,主主轴,偏移)写入 ASUB 中。
- 将接口信号“重新同步”(DB31, ... DBX31.4)置 1

读取偏移

通过以下系统变量可以从零件程序和同步动作中读取三个不同的副主轴位置偏移值。其中变量 \$P_COUP_OFFS[Sn] 只可以在零件程序中使用。

描述	NCK 变量
程序中编写的同步主轴位置偏移	\$P_COUP_OFFS[Sn]
同步主轴的设定值位置偏移	\$AA_COUP_OFFS[Sn]
同步主轴的实际值位置偏移	\$VA_COUP_OFFS[Sn]

“进给停止/主轴停止”

通过定义 MD30455 MISC_FUNCTION_MASK 中的位 4 可以确定副主轴接口信号“进给停止/主轴停止”(DB31, ... DBX4.3)的特性。

位 4 = 0 兼容方法：

通过撤销副主轴的进给使能使同步组制动。

位 4 = 1：

13.1 简要说明

进给使能只作用于插补分量(SPOS,...)，对主轴同步没有任何影响。

说明

更多通过 MD30455 \$MA_MISC_FUNCTION_MASK 来定义轴功能的方法请参考：

文档：

功能手册之基本功能分册；章节“回转轴(R2)”之回转轴的编程

13.1.7 监控主轴同步

精/粗同步

除了常见的主轴监控方式外，还可以在主轴同步中对副主轴与主主轴的同步过程进行监控。

在监控中，NC 可通过接口信号“精同步”(DB31, ... DBX98.0)或“粗同步”(DB31, ... DBX98.1)报告 PLC：副主轴的当前实际位置(AV、DV)或实际速度(VV)是否在指定公差窗口内。

主轴同步激活时，“精同步”和“粗同步”信号会在达到设定值同步状态后自动更新。

公差窗口的大小可由副主轴的机床数据来设置。达到同步运行的过程可由以下系数来控制：

- AV(实际值)、DV(设定值): 副主轴和主主轴的位置差
- VV(速度值): 副主轴和主主轴的速度差

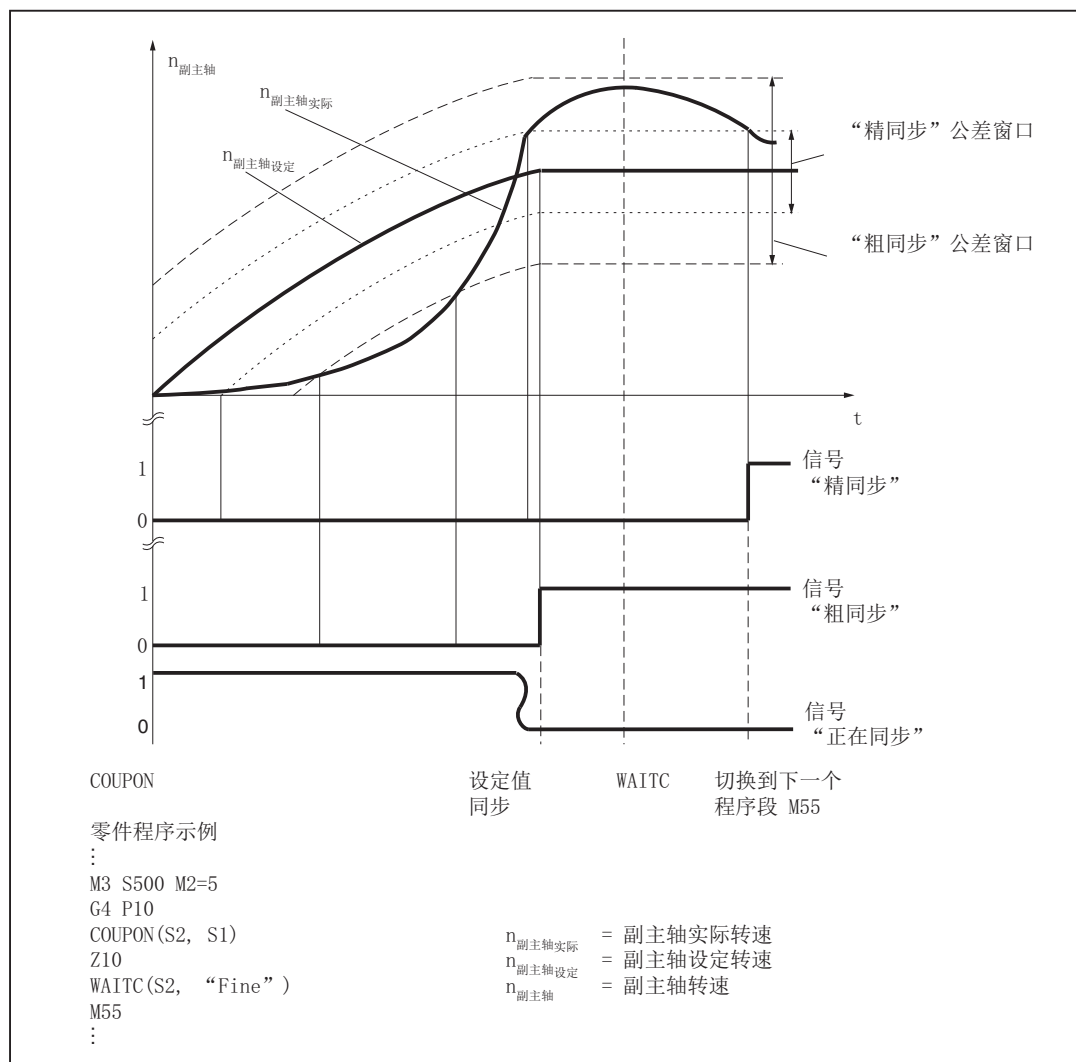


图 13-3 利用 COUPON 和 WAITC 等待标记实现主轴监控，副主轴要与正在旋转的主主轴同步

阈值

副主轴和主主轴的位置公差范围和速度公差范围的单位为度或 rpm。

- “粗同步” 的阈值
轴专用 MD37200: AV(实际值)、DV(设定值): COUPLE_POS_TOL_COARSE
MD37220: VV(速度值): COUPLE_VELO_TOL_COARSE
- “精同步” 的阈值
轴专用 MD37210: AV(实际值)、DV(设定值):COUPLE_POS_TOL_FINE
MD37230: VV(速度值): COUPLE_VELO_TOL_FINE

最大转速/加速度

主轴同步中，主主轴的最大转速和加速度会自动调整，使得在当前的齿轮级的传动比条件下，副主轴会跟随主主轴旋转，且不会超出自身的限值。

比如主主轴会自动减速，使得副主轴不会超出自身的最大转速，从而确保同步生效。

13.2 编程

带主主轴编程的同步指令

指令	功能
COUPDEF（副主轴、主主轴）	定义同步或修改已定义的同步
COUPON（副主轴、主主轴、位置副主轴）	激活同步
COUPONC（副主轴、主主轴）	激活同步并接收当前生效的副主轴转速
COUPOF（副主轴、主主轴、位置副主轴、位置主主轴）	撤销同步
COUPOFS（副主轴、主主轴、位置副主轴）	撤销同步并停止副主轴
COUPDEL（副主轴、主主轴）	删除同步
COUPRES（副主轴、主主轴）	重新激活定义的同步数据

不带主主轴编程的同步指令

指令	功能
COUPOF (副主轴、位置副主轴、位置主主轴)	撤销同步
COUPOFS (副主轴、位置副主轴)	撤销同步并停止副主轴
COUPDEL (副主轴)	删除同步
COUPRES (副主轴)	重新激活定义的同步数据

参见

定义 (COUPDEF) (页 761)

激活和撤销同步 (COUPON、COUPONC、COUPOF) (页 764)

13.2.1 定义 (COUPDEF)

可写入的同步主轴对

可写入的同步主轴对的数量是由可用的轴数决定的总数减去一得到的。和以前一样，也可以通过机床数据来定义主轴同步。

固定定义的同步

在固定定义的主轴同步中，通过 NC 零件程序可以修改同步特性(前提是未设置写保护)和传动比。副主轴和主主轴的机床数据不可以更改。

定义新的同步

使用语言指令“COUPDEF”可以创建新的同步(用户定义)，并将现有的同步修改掉。

完整的同步参数的编程方式为：

COUPDEF (副主轴,主主轴,传动比_{分子},传动比_{分母},程序段切换特性, 同步方式)

通过“副主轴”和“主主轴”可以明确确定采用的是哪个同步主轴对。

仅当需要改变其他同步参数时，才必须编写这些同步参数。其中未编写的参数采用上次的值。

下文具体介绍每个同步参数：

- **副主轴、主主轴：**副主轴名称和主主轴名称

例如： S1, SPI(1), S2, SPI(2)

有效的主轴号必须事先在以下轴专用的 MD35000 \$MA_SPIND_ASSIGN_TO_MACHAX 中指定了对应的机床轴。

- **传动比_{分子}, 传动比_{分母}：**传动比分子和分母

传动比用分子和分母（传动参数）来规定（参见章节“主轴同步 (页 745)”）。其中至少必须规定分子。如果没有规定分母，通常系统假设它为“1.0”。

- **程序段切换特性**

它可用于激活主轴同步时选择何时进行程序段切换：

NOC：立即使能程序段切换

FINE：在满足“精同步”条件时进行程序段切换

COARSE：在满足“粗同步”条件时进行程序段切换

IPOSTOP：在满足 IPOSTOP 条件时进行程序段切换（即：在设定值同步后）

程序段切换特性以字符串的形式指定(即：带引号)。

程序段转换特性用粗体字母指定即可。其余的字母虽然没有什么含义，却可以使零件程序更加清晰易懂。

如果未指定程序段切换特性，那么在这个时间点生效的特性将继续保持有效。

使用可编程的 **WAITC** 等待标记可将切换新的程序段延迟到达到指定的同步过程时再进行。

- **同步方式**

DV (Desired Values: 设定值)：副主轴和主主轴的设定值同步

DV (Actual Values: 实际值)：副主轴和主主轴的实际值同步

DV (Velocity Values: 速度值)：副主轴和主主轴的速度同步

如果未指定同步方式，那么在之前的同步方式继续保持有效。

说明

仅当主轴同步撤销时才能修改同步方式！

示例

COUPDEF (SPI(2), SPI(1), 1.0, 1.0, "FINE", "DV")

COUPDEF (S2, S1, 1.0, 4.0)

COUPDEF (S2, SPI(1), 1.0)

缺省设置

用户定义的主轴同步有以下缺省设置：

- 传动比_{分子}=1.0
- 传动比_{分母}=1.0
- 程序段切换特性 = IPOSTOP (在设定值达到同步后进行程序段切换)
- 同步方式 = DV (设定值同步)

删除同步

使用语言指令“COUPDEL”可以删除用户定义的同步。

COUPDEL(副主轴,主主轴)

说明

使用 COUPDEL 指令可撤销已激活的同步并删除同步数据。此时，报警 16797 无效。

副主轴继续采用以前的转速旋转。该特性与 COUPOF(副主轴,主主轴)指令的特性相一致。

激活初始同步参数

通过语言指令“COUPRES”可以再次激活由机床数据定义的同步参数。

COUPRES(副主轴, 主主轴)

此时用 COUPDEF 指令更改的参数(包括传动比)丢失。

通过指令“COUPRES”可以激活在机床数据和设定数据中设置的参数(即由机床数据定义的同步)以及缺省设置(即用户定义的同步)。

可编程的程序段切换

使用语言指令“WAITC”可以在 NC 程序中标记位置，系统在此位置上等待指定副主轴达到同步条件,并将切换程序段延迟到达到指定的同步状态后再进行（参见章节“图 13-3 利用 COUPON 和 WAITC 等待标记实现主轴监控，副主轴要与正在旋转的主主轴同步(页 759)”）。

WAITC (副主轴)

优点： 工艺上可能会要求主轴同步激活后经过一段时间再达到同步状态。

说明

原则上，等待标记 **WAITC** 可始终写入在程序中。如果指定的主轴不是作为副主轴生效的，那么该指令对该主轴无效。

在 **WAITC** 里没有指定同步条件，系统就会检查主轴同步功能中由程序定义的/机床数据定义的同步条件，起码会检查是否达到设定值同步条件。

示例：

```
WAITC(S2),
:
WAITC(S2, "Fine"),
:
WAITC(S2, S4, "Fine")
```

停止及程序段切换

如果在这个时间段内撤销了轴使能而导致主主轴和副主轴停止，当伺服驱动再次给出轴使能时，主主轴和副主轴会再次运行到**最后的**设定位置上。

可通过程序指令 **COUPON** 和 **WAITC** 来控制程序段切换。同时，可通过 **COUPDEF** 指令或 **MD21320 \$MC_COUPLE_BLOCK_CHANGE_CTRL_1** 来确定程序段切换标准。

13.2.2 激活和撤销同步 (COUPON、COUPONC、COUPOF)

激活主轴同步

使用语言指令 **COUPON** 可激活主轴同步。

激活主轴同步有两种方式可选：

1. **COUPON (副主轴、主主轴)**
使用 **COUPON** 指令可以尽快的激活主轴同步，但不设定主主轴和副主轴之间的角度偏移。
2. **COUPON (副主轴、主主轴、位置_{副主轴})**
激活主轴同步，同时主主轴和副主轴之间存在指定的角度偏移：位置_{副主轴}。该位置在正向上和主主轴零度位置相差的一个位置值。根据设置使能程序段切换。位置_{副主轴}的范围：0 ... 359.999°。
3. **COUPONC (副主轴、主主轴)**
在用 **COUPONC** 指令激活同步时，主轴会继续采用之前编写的转速和方向(M3 S... 或 M4)。此时主主轴和副主轴会以差速运行。该指令中不能写入偏移位置。

写入 COUPON(副主轴,主主轴,位置_{副主轴})或在主轴同步激活时写入 SPOS, 可以修改副主轴与主主轴之间的角度偏移。

撤销主轴同步

撤销主轴同步有三种方式可选:

1. **COUPOF(副主轴,主主轴)**
快速撤销主轴同步。立即使能程序段切换。
2. **COUPOF(副主轴, 主主轴, 位置_{副主轴})**
在副主轴越过指定位置“位置_{副主轴}”后撤销主轴同步。在越过指定位置后使能程序段切换。
3. **COUPOF(副主轴,主主轴,位置_{副主轴}, 位置_{主主轴})**
在副主轴越过“位置_{副主轴}”且主主轴越过“位置_{主主轴}”后撤销主轴同步。在越过两个指定位置后再使能程序段切换。
位置_{副主轴}、位置_{主主轴}的范围: 0 ... 359.999°。

如果写入了连续路径运行(G64), 控制系统内部会自动生成一个程序段停止。

示例:

```
COUPDEF (S2, S1, 1.0, 1.0, "FINE, "DV")
```

```
:
```

```
COUPON (S2, S1, 150)
```

```
:
```

```
COUPOF (S2, S1, 0)
```

```
:
```

```
COUPDEL (S2, S1)
```

1. **COUPOFS (副主轴、主主轴)**
撤销主轴同步, 并停止副主轴。立即进行程序段切换。
2. **COUPOFS(副主轴,主主轴,位置_{副主轴})**
副主轴越过指定位置(以机床坐标系为参照)后才使能程序段切换。
取值范围 0 ... 359.999°。

13.2.3 主轴同步的轴系统变量

确定当前的同步状态

副主轴的当前同步状态可在 NC 件程序中通过下列轴系统变量读取:

\$AA_COUP_ACT [<轴表达式>]

<轴表达式>说明(参见章节“主轴同步 (页 745)”)。

示例:

\$AA_COUP_ACT[S2]

读取的值对于副主轴有以下含义：

- 字节 = 0: 无主轴同步有效
- 位 2 = 1: 主轴同步有效
- 位 2 = 0: 主轴同步无效

读取当前的角度偏移

副主轴与主主轴的当前设定值位置偏移在 NC 件程序中可以用下列轴系统变量读取：

- 副主轴与主主轴的设定值位置偏移：
\$AA_COUP_OFFS[<轴表达式>]
- 副主轴与主主轴的实际值位置偏移：
\$VA_COUP_OFFS[<轴表达式>]

示例：

\$AA_COUP_OFFS[S2]

如果在 COUPON 中写入了一个角度偏移，在达到设定值同步后，读取的值和该角度偏移一致。

读取写入的角度偏移

程序最后写入的副主轴与主主轴的位置偏移在 NC 件程序中可以用下列轴系统变量读取：

\$P_COUP_OFFS[<轴表达式>]

说明

如果在主轴同步激活后撤销伺服使能，主轴会进入跟踪模式，此后再次给出伺服使能后，会产生一个新的位置偏移值，它和之前的程序写入值不同。

\$P_COUP_OFFS 只读取最初的程序写入值。当前值由 **\$AA_COUP_OFFS** 和 **\$VA_COUP_OFFS** 读取。使用接口信号 DB31, ... DBX31.4 (重新同步)可再次恢复最初写入的偏移。

13.2.4 自动激活和撤销位置闭环控制

转速开环模式下的特性

在 DV(设定值)同步方式中，程序指令 COUPON、COUPONC 以及 COUPOF、COUPOFS 会在必要情况下自动激活或撤销主主轴的位置闭环控制。如果在一根主主轴上有多根副主轴，**第一个 DV(设定值) 同步会激活主主轴的位置闭环控制，最后一个 DV(设定值) 同步会撤销主主轴的位置闭环控制**，条件是没有编写 SPCON 指令。

主主轴和副主轴不一定要在同一通道中。

在 COUPON 和 COUPONC 指令下自动激活

同步方式不同，COUPON 和 COUPONC 指令对位置闭环控制的影响也不同：

同步方式	DV(设定值)	AV(实际值)	VV(速度值)
副主轴	开启位置闭环控制	开启位置闭环控制	无影响
主主轴	开启位置闭环控制 ¹	无影响	无影响

¹ 在 DV(设定值)同步方式下，如果 **至少有一个** 副主轴与主主轴同步，那么位置闭环控制会在 COUPON 和 COUPONC 指令下自动激活。

在 COUPOF 和 COUPOFS 指令下自动撤销

同步方式不同，COUPOF 和 COUPOFS 指令对位置闭环控制的影响也不同：

同步方式	DV(设定值)	AV(实际值)	VV(速度值)
副主轴	关闭位置闭环控制 ²	关闭位置闭环控制 ²	无影响 ²
主主轴	关闭位置闭环控制 ³	无影响	无影响

²COUPOF 和 COUPOFS 指令，不指定位置

激活副主轴转速开环模式。当 COUPFS 指定停止位置就可以自动激活位置模式。在写入了 SPCON 指令使副主轴进入位置闭环控制主轴模式，或在 COUPFS 指令中指定了位置时，位置闭环控制**不会自动关闭**。

³ 当该主主轴没有其他的副主轴和它进行 DV(设定值)同步时，COUPOF 指令就会自动**关闭**位置闭环控制。当主主轴位于定位模式或进给轴模式中**或是**因 SPCON 指令进入位置闭环控制主轴模式时，位置闭环控制**都会继续存在**。

13.3 定义

说明
每个通道都可以定义一个同步主轴对。

表格 13-1 机床数据

编号	名称: \$MC_	功能
MD21300	COUPLE_AXIS_1[<n>]	同步主轴对的机床轴： • <n> = 0: 副主轴的机床轴编号 • <n> = 1: 主主轴的机床轴编号 机床轴编号对应: MD20070 \$MC_AXCONF_MACHAX_USED（通道中的机床轴） 机床轴编号 == 0: 未定义同步。以下系统数据也变无效。 注: 为同步主轴对定义的机床轴 无法 通过程序指令修改。
MD21320	COUPLE_BLOCK_CHANGE_CTRL_1	激活主轴同步后使能程序段切换 ¹⁾ ： • 立即生效 • 达到“精同步” • 达到“粗同步” • 达到设定值同步 注: 只要没有设置写保护 ¹⁾ ，就可以通过指令 COUPDEF 修改程序段切换使能。
MD21310	COUPLING_MODE_1	同步方式 ¹⁾ ： • 实际值同步 • 设定值同步 • 速度同步 注: 只要没有设置写保护 ¹⁾ ，就可以在同步 撤销 后通过指令 COUPDEF 修改同步方式。
MD21330	COUPLE_RESET_MODE_1	NC 启动、NC 停止和复位时同步主轴对的特性。
MD21340	COUPLE_IS_WRITE_PROT_1	同步写保护
1) 参见 MD21340 \$MC_COUPLE_IS_WRITE_PROT_1		

表格 13-2 设定数据

编号	名称: \$SC_	功能
SD42300	COUPLE_RATIO_1[<n>]	转速传动比: 副主轴/主主轴 = 分子/分母 ¹⁾ : • <n> = 0: 分子 (副主轴) • <n> = 1: 分母 (主主轴) 注: 只要没有设置写保护 ¹⁾ , 就可以通过指令 COUPDEF 修改传动比。
1) 参见 MD21340 \$MC_COUPLE_IS_WRITE_PROT_1		

13.3.1 NC 启动时同步主轴对的特性

NC 启动时同步主轴对的特性取决于以下机床数据中的设置:

定义的同步主轴对

特性	MD21330 \$MC_COUPLE_RESET_MODE_1
保留同步	位 0 = 0
撤销同步	位 0 = 1
激活定义的数据	位 5 = 1
激活同步	位 9 = 1

编程的同步主轴对

特性	MD20112 \$MC_START_MODE_MASK
保留同步	位 10 = 0
撤销同步	位 10 = 1

13.3.2 复位时同步主轴对的特性

复位和程序结束时主轴同步的特性取决于以下机床数据中的设置:

定义的同步主轴对

特性	MD21330 \$MC_COUPLE_RESET_ MODE_1	MD20110 \$MC_RESET_MODE_MAS K
保留同步	位 1 = 0	位 0 = 1
撤销同步	位 1 = 1	位 0 = 1
激活定义的数据	位 6 = 1	位 0 = 1

编程的同步主轴对

	MD20110 \$MC_RESET_MODE_MASK
保留同步	位 0 = 1, 位 10 = 1
撤销同步	位 0 = 1, 位 10 = 0

13.4 特点

13.4.1 主轴同步的一般特点

动态响应

使用设定值同步时，副主轴和主主轴的位置控制器参数应该一致(例如：K_v 系数)。必要时还要为转速闭环控制和主轴同步激活不同的参数组。与位置闭环控制、前馈控制和参数组不同的是，副主轴的控制器参数可以通过 MD30455 \$MA_MISC_FUNCTION_MASK 来设置，如同未经同步时(参见章节“主轴同步功能调试的特殊性 (页 784)”)。

前馈控制

为确保主轴同步中的主主轴和副主轴具有更好的动态响应性能，前馈控制**始终是激活的**。

通过轴专用的 MD32620 \$MA_FFW_MODE 可以撤销副主轴和主主轴的前馈控制。当 MD32620 \$MA_FFW_MODE 置零时，功能会受限。也就无法通过 SPCON 指令在运动过程中激活位置闭环控制。SPOS、M19 或 SPOSA 指令因此失效。也无法通过在 NC 零件程序中编写 FFWOF 指令来撤销副主轴和主主轴的前馈控制。

可以通过轴专用的 MD32620 \$MA_FFW_MODE 来确定前馈控制方式(转速还是转矩前馈控制)，参见章节“K3: 补偿 (页 241)”。

最大转速/加速度

同步主轴对里的最大转速和最大加速度是由其中“最慢的那根主轴”来决定的。此时系统会考虑当前的齿轮档、程序写入的加速度和当前的位置闭环状态(激活/撤销)。

在控制系统内部，主主轴的最大转速和加速度是由副主轴的传动比和最大转速/加速度来决定的。

复合同步

如果在激活主轴同步时发现了其中有一根副主轴或主主轴当前有生效的同步关系，系统会忽略此激活指令并发出一个报警信息。

复合同步示例:

- 主轴作为副主轴用于多个主主轴
- 同步级联(一个副主轴是下一个同步对的主主轴)

每个通道中可定义的主轴数量

通道中的每根轴都可定义为主轴。每个通道中的轴数量取决于控制性能。

跨通道设定值同步

- 系统支持跨通道的主同步，不会对 DV(设定值)、AV(实际值)和 VV(速度值)同步产生额外的限制。
- 一个主主轴可以和 NCU 任意一个通道内的副主轴同步，副主轴的最大数量是主轴总数减一。

由 ASUB 启动主轴同步

通过 PLC 启动 ASUB 后，可在 AUTO 或 MDA 模式下激活/撤销或结束同步。

文档:

功能手册之基本功能分册；章节“BAG、通道、程序运行、复位特性 (K1)”

报警时的属性

如果在主轴同步期间出现了报警，而且该报警会在控制系统内部导致“撤销伺服使能”和“激活跟踪方式”，则控制系统的后续响应和由以下 NC/PLC 接口信号导致的响应一致。

- DB31, ... DBX2.1 = 0 (伺服使能)
- DB31, ... DBX1.4 = 1 (跟踪方式)

(参见章节“主轴同步和 NC/PLC 接口信号 (页 774)”)。

通过 NC/PLC 接口信号重新进行同步：

DB31, ... DBX31.4 = 0 → 1 (重新同步)

可再次恢复已写入的偏移 (参见章节“再次恢复副主轴和主主轴的同步 (页 772)”)。

在激活的主轴同步中进行程序段搜索

说明

我们建议在主轴同步激活时采用搜索类型 5 “通过程序测试进行程序段搜索”(SERUPRO)。

13.4.2 再次恢复副主轴和主主轴的同步

位置偏移的原因

撤销伺服使能后，在跟踪方式下重新激活同步时，主主轴和副主轴之间可能会出现位置偏移。位置偏移的产生原因可能为：

- 一个零件被夹住或是两根主轴被手动旋转(加工区被打开、驱动器未通电)。
- 撤销主轴使能后，两根主轴会因为惯性停转速度不同产生位置偏移。
- 已出现驱动报警(内部的跟踪方式):
DB31, ... DBX61.3 (跟踪方式生效) = 1
删除报警时不允许从 NC 中触发任何的同步动作。
- 由于副主轴同步禁用，不可以执行同步动作:
DB31, ... DBX29.5 (禁用同步)

基本流程

通过以下步骤可再次恢复副主轴和主主轴之间已经丢失的同步或还未执行的同步：

1. 设置轴使能，取消同步禁用，如果之前已设置同步禁用。
2. 通过 NC/PLC 接口信号启动副主轴重新同步：
DB31, ... DBX31.4 (重新同步)
只有在执行重新同步后才可以再次完整恢复设定值同步。
3. 等待主轴完成同步。

使能重新同步

设置重新同步使能后，副主轴和主主轴的实际位置进行重新同步。以下两个接口信号会置位：

DB31, ... DBX98.1 (粗同步)

DB31, ... DBX98.0 (精同步)

重新同步须满足以下条件：

- 副主轴获得使能。
- PLC 没有为副主轴设置同步禁用：
DB31, ... DBX31.5 (禁用同步)

重新同步副主轴

一旦 NC 识别到以下信号出现一个上升沿，它就会为对应的副主轴启动重新同步：

DB31, ... DBX31.4 (重新同步)

NC 输出以下 NC/PLC 接口信号报告发现了上升沿：

DB31, ... DBX99.4 (正在进行同步)

当出现以下情况时，接口信号“正在进行同步”会复位：

- 副主轴和主主轴的设定值达到同步。
- 接口信号 DB31, ... DBX31.4 (重新同步)已复位。

副主轴进行附加运动时，同步信号的特性

会计算出用于测定同步信号的叠加分量。

13.4 特点

示例

程序代码	注释
N51 SPOS=0 SPOS[2]=90	
N52 OUPDEF(S2,S1,1,1,"FINE","DV")	
N53 COUPON(S2,S1,77)	
N54 M0	; 偏移=77°, 出现同步信号“粗”/“精”。
N55 SPOS[2]=0 FA[S2]=3600	; 修改偏移, 报告同步信号“粗”/“精”
N56 M0	; (注意公差)
	; 偏移=0°, 出现同步信号“粗”/“精”。
N60 M2=3 S2=500	; 差速, 报告同步信号“粗”/“精”。
	; 偏移未确定, 报告同步信号“粗”/“精”。
N65 M0	; (注意公差)

说明

通过撤销轴使能可以中断副主轴的叠加运行(例如:SPOS)。该运动分量不受 NC/PLC 接口信号 DB31, ... DBX31.4(重新同步)控制, 而是通过 REPOS 过程修复。

边界条件

只有当副主轴和主主轴存在**定义的偏移**时, 接口信号 DB31, ... DBX31.4 (重新同步)才会生效。

偏移的生成方式有: 包含位置偏移的 COUPON 指令 (COUPON(...,77)); SPOS、SPOSA、M19 指令生成的偏移。

13.4.3 主轴同步和 NC/PLC 接口信号

说明

在主轴同步中, 始终都要考虑到主主轴或副主轴的各个接口信号对同步的影响。

主轴倍率(DB31, ... DBB19)

在主轴同步中, 只有主主轴的倍率值有效。

主轴禁用 (DB31, ... DBX1.3)

LS	FS	同步	特性
0	0	关	输出设定值
0	1	关	不向副主轴输出设定值
1	0	关	不向主主轴输出设定值
1	1	关	不向主主轴和副主轴输出设定值
0	0	开	输出设定值
0	1	开	不禁用副主轴
1	0	开	禁用副主轴，不输出设定值
1	1	开	不向主主轴和副主轴输出设定值

伺服使能 (DB31, ... DBX2.1)

主主轴：在主轴同步激活时复位“伺服使能”

在主轴同步中，如果在**设定值**同步激活时复位主主轴的伺服使能，控制系统内部则会自动切换至**实际值**同步。如果在主主轴运行时复位伺服使能，那么主主轴会停止并显示一个报警。主轴同步依然保留。

主主轴和副主轴：选择不带伺服使能的主轴同步

如果在选择主轴同步时未设置主主轴和副主轴的伺服使能，系统则会激活主轴同步。但副主轴和主主轴会一直处于静止状态，直至主主轴和副主轴都获得了伺服使能。

主主轴和副主轴：设置“伺服使能”

设置“伺服使能”后，主轴接收哪个位置取决于 DB31, ... DBX1.4 == <值>（跟踪运行）：

<值>	接收的主轴位置
0	撤销伺服使能时的位置
1	当前位置

说明

建议在程序段搜索期间写入始终共用于副主轴和主主轴的接口信号 DB31, ... DBX2.1（伺服使能）。否则，由于缺少主主轴的伺服使能，程序段搜索会在副主轴加工后停止。

说明**同步误差**

主轴停止后，在没有事先撤销主轴同步的情况下撤销了接口信号 DB31, ... DBX2.1（伺服使能），那么在重新激活伺服使能时，就不会对由于外部影响而产生的同步误差进行补偿。这可使副主轴和主主轴之间编程的相对角度丢失。

通过重新同步可再次建立相对角度：DB31, ... DBX31.4 = 1（重新同步）

跟踪方式 (DB31, ... DBX1.4)

跟踪运行中会循环将设定位置设为当前实际位置：

DB31, ... DBX1.4 == 1（跟踪运行）和 DB31, ... DBX2.1 == 0（伺服使能）

⇒ 循环：设定位置 = 实际位置

说明

DB31, ... DBX1.4（跟踪运行）只与 DB31, ... DBX2.1 == 0（伺服使能）相关

位置测量系统 1/2 (DB31, ... DBX1.5 或 1.6)

主轴同步期间系统不会禁止位置测量系统的切换。切换后主轴同步仍保持生效。

说明

建议在撤销主轴同步后再进行切换。

删除剩余行程/主轴复位(DB31, ... DBX2.2)

主主轴：在主轴同步激活时设置主轴复位

设置主主轴复位后，主主轴会以设定好的速度减速至静止状态。主轴同步依然保留。

叠加运动（除了在 COUPON / COUPONC 中）也会尽快结束。

主轴停止(进给停止)(DB31, ... DBX4.3)

主主轴和副主轴：在主轴同步激活时设置主轴停止

设置副主轴或主主轴的“主轴停止”后，两根主轴会同步减速至静止状态。主轴同步依然保留。

复位主轴停止

复位了两根主轴的“主轴停止”后，两根主轴会再次加速至最后生效的转速设定值。

应用示例

在主轴同步激活的状态下打开防护门时会使副主轴和主主轴停止。

主主轴和副主轴的信号曲线：

1. 停止：DB31, ... DBX4.3 = 1（主轴停止）
2. 等待停止：DB31, ... DBX61.4 == 1
3. 停止：DB31, ... DBX2.1 = 0（伺服使能）

删除 S 值(DB31, ... DBX16.7)

主主轴：在主轴同步激活时删除 S 值

如果设置了“删除 S 值”，主主轴会通过一个斜坡减速至静止状态。主轴同步依然保留。

副主轴：在主轴同步激活时删除 S 值

接口信号不作用于主轴同步中的副主轴。

重新同步主轴 1/2 (DB31, ... DBX16.4 和 16.5)

主主轴：在主轴同步激活时重新同步位置测量系统

说明

建议在撤销主轴同步后再重新同步位置测量系统。

重新同步 (DB31, ... DBX31.4)

副主轴：恢复编程的角度偏移

在一个已经丢失或未执行副主轴和主主轴同步的系统中，可再次恢复编程的角度偏移。

- 要求：DB31, ... DBX31.4 = 1（重新同步）
- 应答：DB31, ... DBX99.4 == 1（正在进行同步）

13.4 特点

JOG 模式下的移动键(DB31, ... DBX4.6 和 4.7)

在主轴同步中，系统内部不会自动禁用 JOG 模式下副主轴的“正/负移动键”，因此，在按下移动键以后会产生副主轴的一个叠加运动。

说明

如果需要阻止叠加运动，那么必须由 PLC 用户程序来执行。

NC 停止进给轴和主轴(DB21, ... DBX7.4)

在主轴同步中，通过“NC 停止进给轴和主轴”信号可使已同步的主轴在设定的动态响应下制动。此时，主轴同步依然保留。

NC 启动(DB21, ... DBX7.1)

(参见章节“NC 启动时同步主轴对的特性 (页 769)”)

说明

在给出 NC 停止信号后给出再启动 NC 不会撤销主轴同步。

13.4.4 主主轴和副主轴之间的差速

什么时候会出现差速？

在车床应用中，两根主轴相对放置时会产生差速。副主轴的转速是两个带符号的转速分量之和，其中，一个转速分量是主主轴转速和传动比的乘积，另一个转速分量以及旋转方向通过以下指令写入：

- S...指定转速
- M3、M4 或 M5 指定旋转方向

通常传动比设为 -1 即可达到同步。负数传动比可反转方向，主主轴转速与之相乘后的乘积再加上另一个转速分量即可得出副主轴的转速。NC 的典型工作方式会在下图中详细说明。

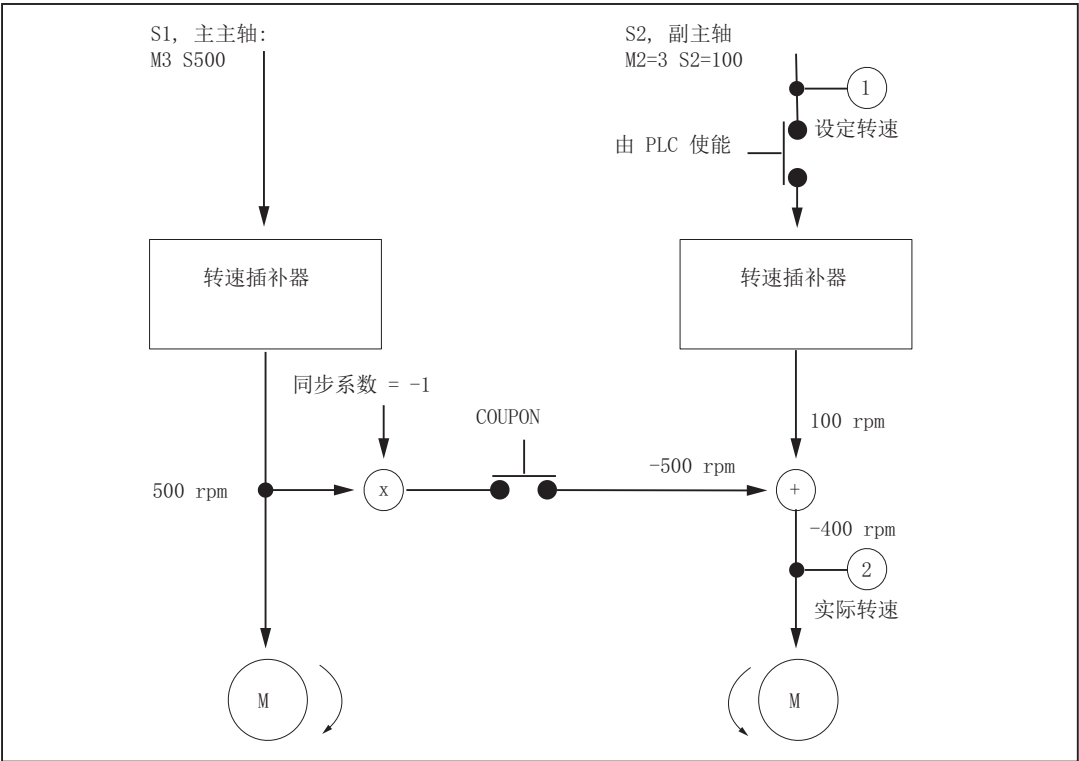


图 13-4 差速生成示意图

示例

程序代码	注释
N01 M3 S500	; 主轴 1 以 500 转/分正转
	; 主主轴是主轴 1
N02 M2=3 S2=300	, 主轴 2 以 300 转/分正转
N05 G4 F1;	
N10 COUPDEF(S2,S1,-1)	; 传动比 -1:1
N11 COUPON(S2,S1)	; 激活同步, 副主轴 2 的转速
	; 由主主轴 1 的转速及
	; 传动比得出
N26 M2=3 S2=100	; 编程差速,
	; 主轴 2 是副主轴

应用

在定位主主轴上用动力刀具进行加工时要求副主轴能够精确的同步。围绕副主轴旋转的刀塔提供有不同的刀具进行加工。下图展示的是一个应用示例，示例中的刀具和主主轴是平行的。

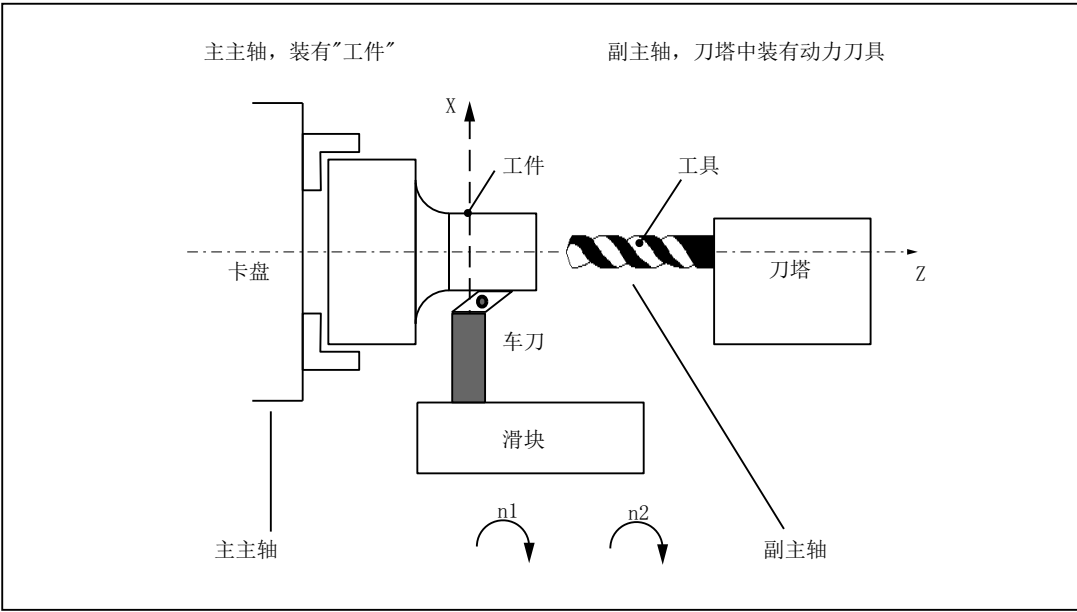


图 13-5 单滑块车床应用，刀塔绕 Z 轴旋转。

前提条件

差速编程的基本条件:

- 具备功能“主轴同步”。
- 副主轴的动态响能力至少要和主主轴的相同。否则会在进行不带补偿夹具的攻丝 G331/G332 时出现质量问题。
- 差速必须在副主轴位于的通道内编程。主主轴可在另一个通道内进行编程。
- 差速必须由 PLC 通过设置副主轴接口信号“使能叠加”(DB31, ... DBX26.4)来使能。如果该使能不存在，系统会发出报警 16771“通道%从动轴%叠加运动未使能”。该报警可通过置位接口信号“使能叠加运动”(DB31, ... DBX26.4)或撤销主轴同步来清除。

说明

差速对主轴同步的激活没有影响。

在同步激活时不可以进行主主轴和副主轴的齿轮换挡。

通过 COUPONC 指令激活同步

主轴同步激活时，副主轴和以前一样，按传动比加速到主主轴转速。如果在激活主轴同步时副主轴已经在某方向旋转(M3, M4)，那么系统会继续采用副主轴的该转速。

撤销同步

撤销主轴同步后，副主轴仍以两个转速分量之和继续旋转。就好像主轴还是以之前编程的转速和旋转方向旋转。也就是说撤销主轴同步后主轴的工作方式保持不变。

差速

差速的形成途径有:在主轴同步期间在转速开环模式中重新编程副主轴的旋转方向和转速，比如:本示例中为 S2=... 或 M2=3, M2=4；另一种方法是通过编写 COUONC 指令来激活主轴同步，该指令会时使副主轴继续采用之前编程的转速。

边界条件:

指定旋转方向 M3 或 M4 时，转速 S... 必须要一同重新指定。否则系统会发出报警 16111 “通道%程序段%主轴%未编写转速”。

读取副主轴的偏移

在写入差速后，当前的偏移会一直发生变化。可通过 \$AA_COUP_OFFS[Sn] 读取设定值的偏移，通过 \$VA_COUP_OFFS[Sn] 读取实际值的偏移。

可通过变量 \$P_COUP_OFFS[Sn] 读取程序写入的最后偏移。

显示差速

在写入了差速后(本例中是 100 rpm)，差速作为转速设定值显示。

转速实际值指的是电机转速。在本例中实际转速为 $500 \text{ rpm} * (-1) + 100 \text{ rpm} = -400 \text{ rpm}$ 。

从 NCK 到 PLC 的接口信号

转速开环模式中的副主轴

当达到已写入的差速时(参见先前的示例 N26 M2=3 S2=100)，NCK 会置位副主轴的接口信号“主轴位于设定值区域内”(DB31, ... DBX83.5)。虽然写入了差速，但是若 PLC 未使用差速，该信号也不会置位。

编写了差速后，如果现有的主轴同步有需要，副主轴可仍然保持在位置闭环控制模式下。

说明

当写入的差速生成了除同步设定值外其他的设定值时，轴 VDI 接口信号 NCK → PLC “叠加运动”(DB31, ... DBX98.4)置位。

实际旋转方向为正转(DB31, ... DBX83.7)

接口信号“实际旋转方向为正转”(DB31, ... DBX83.7)指的是当前的电机旋转方向。

从 PLC 到 NCK 的接口信号

通过 PLC 接口控制副主轴

下文说明了各个轴 VDI 接口信号对转速开环模式中副主轴差速的影响。

删除剩余行程/主轴复位(DB31, ... DBX2.2)

已写入的差速和旋转方向可由接口信号“删除剩余行程/主轴复位”(DB31, ... DBX2.2)删除。
置位接口信号“删除 S 值”(DB31, ... DBX16.7)可删除已写入的转速。

重新同步主轴 1/2 (DB31, ... DBX16.4 和 16.5)

接口信号“重新同步主轴 1/2”(DB31, ... DBX16.4/16.5)未被禁用。但主轴同步不自动补偿产生的偏移。

M3/M4 取反(DB31, ... DBX17.6)

接口信号“M3/M4 取反”(DB31, ... DBX17.6)只能使额外写入的副主轴转速分量取反。
该信号不会影响通过主轴同步生成的转速分量。

主轴倍率(DB31, ... DBB19)

VDI 接口信号“主轴倍率”(DB31, ... DBB19)只会影响为副主轴额外写入的转速分量。当主轴倍率开关传送至所有的轴输入端时，副主轴的倍率值会发生两次变化：

- 一次因主轴转速变化而间接变化
- 另一次是副主轴本身转速分量的直接变化。

在 PLC 程序中可对倍率值进行相应的调整。

撤销同步

撤销同步后，副主轴会以两个转速分量之和继续旋转。即在撤销同步后主轴转速是恒定的。

在使用 COUPOF 撤销同步时，主轴的转速和旋转方向仍保持不变，仿佛在程序中写入了这两个指令。本例中主轴旋转方向和转速为 M4 S400。

在 COUPOFS 指令下，副主轴会从当前的转速运行至静止状态。

激活附加功能

副主轴也可作为主主轴。这种情况下可使用附加功能。

- G95、G96、G97 指令下的旋转进给率。通过指令 G96 S2=... 可以激活副主轴的“恒定切削速度”功能。
取决于 X 轴位置的转速是主轴 2 转速插补器的设定转速，计入主轴 2 的总转速计算中。
- 不带补偿夹具的攻丝 G331、G332。

13.4.5 同步补偿中同步信号的特性

同步补偿的作用

通过比较实际值和经过补偿的设定值可以得出新的同步信号。补偿成功后同步信号应再次出现。

13.4.6 删除同步补偿及 NC 复位

无论在哪种情况下删除了同步补偿值，变量 \$AA_COUP_CORR[Sn] 提供的值都是零。

- 使用 COUPON(..)/COUPONC(..) 指令激活相关副主轴的主轴同步时，当前的同步补偿会附加到设定位置上。
- 但在 NC 复位时，系统却不会在零件程序结束时将当前的同步补偿附加到设定位置上。同步信号不受此影响。
- 当前的同步补偿不受 M30 的影响。
- 用户也可以提前把变量 \$AA_COUP_CORR 设为零来删除这个补偿值。同步补偿会立即结束或是在较大值时通过一个斜坡减速至结束。

13.4.7 主轴同步功能调试的特殊性

主轴调试

首先，主主轴和副主轴要像一个普通的主轴一样进行一次调试。相应的调试步骤参见：
文档:

调试手册 CNC: NCK、PLC、驱动

功能手册之基本功能分册；章节“主轴(S1)”

定义

接着可由以下数据对同步主轴对进行定义:

- 主主轴和副主轴的机床轴号
(在使用通道专用机床数据 MD21300 MC_COUPLE_AXIS_1[n] 固定定义的同步中)
- 所需的同步方式(设定值同步、实际值同步或速度同步)
(在使用通道专用机床数据 MD21310 \$MC_COUPLING_MODE_1[n] 固定定义的同步中)
- 确定副主轴和主主轴的齿轮档
- 在固定定义的主轴同步中还需要确定以下同步特性:
 - 主轴同步中的程序段切换特性:
MD21320 \$MC_COUPLE_BLOCK_CHANGE_CTRL_1
 - 同步撤销特性:
MD21330 \$MC_COUPLE_RESET_MODE_1
 - 同步参数设有写保护:
MD21340 \$MC_COUPLE_IS_WRITE_PROT_1
 - 主轴同步的传动比:
SD42300 \$SC_COUPLE_RATIO_1[n]

设定值同步中副主轴和主主轴的控制性能

为了在使用**设定值同步**时达到最好的同步效果，副主轴和主主轴的动态响应必须相同，以确保良好的控制性能。最好分别**优化**各轴控制回路(位置、转速和电流控制器)，以尽快且有效地调节误差。

为了使各轴不同的动态响应相同且不降低控制性能，系统提供了设定值回路的**动态响应自适应功能**（参见章节“K3: 补偿 (页 241)”）。必须优化主主轴和副主轴的下列控制参数设置:

- K_v 系数(MD32200 \$MA_POSCTRL_GAIN)
- 前馈控制参数
 - MD32620 \$MA_FFW_MODE
 - MD32610 \$MA_VELO_FFW_WEIGHT
 - MD32650 \$MA_AX_INERTIA
 - MD32800 \$MA_EQUIV_CURRCTRL_TIME
 - MD32810 \$MA_EQUIV_SPEEDCTRL_TIME

丢失同步时的特性:

- 轴专用的 MD32620 \$MA_FFW_MODE

我们建议设置 MD32620 = 3，将副主轴的**前馈方式**设为采用 Tt 对称的转速前馈。

可通过更改轴专用的机床数据对这种前馈方式的对称过程进行进一步优化:

机床数据	含义
MD32810 EQUIV_SPEEDCTRL_TIME	转速环前馈控制的等效时间常数
MD37200 COUPLE_POS_TOL_COURSE	‘粗同步’的阈值
MD37210 COUPLE_POS_TOL_FINE	‘精同步’的阈值
MD37220 COUPLE_VELO_TOL_COURSE	速度公差 ‘粗’
MD37220 COUPLE_VELO_TOL_FINE	速度公差 ‘精’

其中，同步信号的阈值越高，位置公差和速度公差越大，结果也会越稳定。

动态响应自适应功能

副主轴和主主轴的动态响应必须相同，以达到最好的同步效果。相同转速下其滞后量相同。设定值回路中的动态响应自适应功能可有效平衡动态响应不同的主轴之间的控制性能。动态响应自适应功能的时间常数应是动态响应“最慢”的轴和其他轴之间等效时间常数的差值。

示例

在转速前馈控制已激活的前提下，动态响应基本上由“最慢”转速控制回路的等效时间常数来确定。

- 主主轴的等效时间常数: MD32810 \$MA_EQUIV_SPEEDCTRL_TIME[<LS>] = 5 ms
- 副主轴的等效时间常数: MD32810 \$MA_EQUIV_SPEEDCTRL_TIME[<FS>] = 3 ms

- 副主轴动态响应自适应功能的时间常数：MD32910 \$MA_DYN_MATCH_TIME[<FS>] = 5 ms - 3 ms = 2 ms
- 激活副主轴的动态响应自适应功能：MD32900 \$MA_DYN_MATCH_ENABLE[<FS>] = 1

如果动态响应自适应功能设置正确，副主轴和主主轴的滞后量则相同：操作区“诊断”>“轴信息”

必要时可稍微调整 K_v 系数或前馈参数，以达到最佳的效果

闭环控制参数组

主轴上，每个齿轮级对应一个闭环控制参数组。通过这些参数组可对主轴同步中的主主轴和副主轴的动态响应进行调整。为此，须为转速模式和定位模式及主轴同步各指定另一个齿轮级。激活各个运行方式前须切换至对应的齿轮级。

控制参数

主主轴和副主轴的下列闭环控制参数的设置必须相同：

- MD33000 \$MA_FIPO_TYPE（精插补器类型）
- MD32400 \$MA_AX_JERK_ENABLE（最大轴急动度）
- MD32402 \$MA_AX_JERK_MODE（用于轴向最大急动度的滤波器类型）
- MD32410 \$MA_AX_JERK_TIME（轴向急动滤波器的时间常量）
- MD32412 \$MA_AX_JERK_FREQ（轴向急动滤波器的禁用频率）
- MD32414 \$MA_AX_JERK_DAMP（轴向急动滤波器的衰减）
- MD32420 \$MA_JOG_AND_POS_JERK_ENABLE（使能最大急动度）
- MD32430 \$MA_JOG_AND_POS_MAX_JERK（轴向急动）

副主轴：自动设置闭环控制参数

可通过机床数据按照以下方式设置副主轴的闭环控制参数：

MD30455 \$MA_MISC_FUNCTION_MASK

位 5=0: 主轴同步，副主轴：

位置闭环控制、前馈控制和参数组的设置和副主轴上的设置一样。

位 5=1: 主轴同步：

控制参数的设置和在没有激活主轴同步的状态下的设置一样。

文档:

功能手册之基本功能分册；章节“速度，设定值/实际值系统，闭环控制 (G2)”

主轴同步中自动接收主主轴参数

在主轴同步中，可由主轴接收闭环控制、前馈控制和副主轴参数组的参数值，以自动执行副主轴和主主轴的动态响应自适应功能。

MD30455 \$MA_MISC_FUNCTION_MASK, 位 5

主轴模式/进给轴模式下不同的动态响应

可通过以下机床数据设置，使得动态响应指令 FA、OVRA、ACC 和 VELOLIMA 对主轴或进给轴单独生效：

MD30455 \$MA_MISK_FUNCTION_MASK 位 6=0

随后在上述指令中写入进给轴名或主轴名。例如：VELOLIMA[S1]=50 指令只能使处于主轴模式的主轴的最大转速降到 50%，而 VELOLIMA[C]=50 指令只能使处于进给轴模式中的主轴的最大转速降到 50%。

如果希望 VELOLIMA[S1]=50 和 VELOLIMA[C]=50 指令具有相同的作用，就如下设置该机床数据，使得指令 FA、OVRA、ACC 和 VELOLIM 对轴统一生效，如同之前的指令编程方式。

MD30455 \$MA_MISK_FUNCTION_MASK 位 6=1

折线式加速度特性曲线

主主轴的折线式加速度特性曲线对副主轴的影响可通过以下轴专用的机床数据来确定：

- MD35220 \$MA_ACCEL_REDUCTION_SPEED_POINT (加速度下调起始转速)及
- MD35230 \$MA_ACCEL_REDUCTION_FACTOR (加速度下调系数)。

只要有数据 MD35242 \$MA_ACCEL_REDUCTION_TYPE，就可以对加速度下调方式进行定义。否则加速度会呈双曲线的形式下降。

如果在计算了传动比的条件下，副主轴的动态响应小于主主轴的动态响应，系统会自动将主主轴的动态响应下调至一个合理值。

副主轴的加速度在整个转速范围内应保持**恒定**。如果通过上述机床数据也为副主轴定义了一条折线式加速度特性曲线，那系统只会在激活同步的过程中考虑该加速度特性曲线。副主轴的设定值按定义的折线式加速度特性曲线变化。

13.4 特点

文档:

功能手册之基本功能分册；章节“加速度(B2)”

章节:折线式加速度特征曲线

实际值同步

使用实际值同步(AV)时，副主轴的动态响应大大超过主主轴的动态响应。系统也会根据各轴动态响应来最佳地设置各个驱动。

实际值同步只应在特殊情况下使用。

速度同步

在系统内部是一个设定值同步，但速度同步(VV)对副主轴和主主轴的动态响应要比设定值同步(DV)低一些。副主轴和/或主主轴并不强制要求位置闭环控制和测量系统。

粗/精同步的阈值

在优化了控制器以及设置了前馈控制后，必须为副主轴输入粗/精同步的阈值。

- “粗同步”的阈值

轴专用 MD7200: AV(实际值), DV(设定值): COUPLE_POS_TOL_COARSE

MD37220: VV(速度值): COUPLE_VELO_TOL_COARSE

- “精同步”的阈值

轴专用 MD37210: AV(实际值)、DV(设定值): COUPLE_POS_TOL_FINE

MD37230: VV(速度值): COUPLE_VELO_TOL_FINE

副主轴的上述值应根据机床制造商要求的加工精度来确定，并由 PLC 接口通过信息显示窗口进行检查。

主主轴/副主轴的角度偏差

例如：如果在激活主轴同步时需要两者之间存在指定的角度偏差，那就必须使副主轴和主主轴的“零度位置”一致。该操作可通过以下机床数据完成：

- MD34100 \$MA_REFP_SET_POS
- MD34080 \$MA_REFP_MOVE_DIST
- MD34090 \$MA_REFP_MOVE_DIST_CORR

文档:

功能手册之基本功能分册；章节“回参考点(R1)”

显示副主轴信息

操作区域“诊断”中会显示副主轴的以下值，以便于调试：

- 副主轴与主主轴的设定值偏差
显示值：和主主轴之间的位置偏移（设定值）
(该值与通过轴变量 \$AA_COUP_OFFS 在零件程序中可读的角度偏差一致)
- 副主轴与主主轴的实际值偏差
显示值：和主主轴之间的位置偏移（实际值）

文档：
操作手册

13.5 边界条件

“主轴同步”功能的可用性

该功能是一个选件(“主轴同步/多面体车削”或“通用耦合”中的某个对应的选件)，它必须获得授权后方可和硬件配用。

说明
有关通用耦合的不同类型的信息请参考：
文档：
功能手册之特殊功能分册；章节“轴耦合(M3)”

13.6 示例

编程举例

程序代码	注释
	; 主主轴 = 主轴 1
	; 副主轴 = 主轴 2
N05 M3 S3000 M2=4 S2=500	; 主主轴以 3000 转/分转动
	; 副主轴:500 转/分。
N10 COUPDEF (S2, S1, 1, 1, "No", "Dv")	; 同步也可以由机床数据

13.7 数据表

程序代码	注释
N70 SPCON	; 定义 ; 主主轴进入位置闭环控制 ; (设定值同步)。
N75 SPCON(2)	; 副主轴进入位置闭环控制
N80 COUPON (S2, S1, 45)	; 实时同步，两者之间的角度偏移为 45 度。
N200 FA[S2]=100	; 定位速度 = 100 度/分
N205 SPOS[2] = IC(-90)	; 负方向上叠加转动 90 度
N210 WAITC(S2, "Fine")	; 等待“精同步”
N212 G1 X.., Y..F...	; 加工
N215 SPOS[2]=IC(180)	; 正方向上叠加转动 180 度
N220 G4 S50	; 暂停时间 = 50 转
	; 主主轴
N225 FA[S2]=0	; 激活由机床数据定义的速度
N230 SPOS[2]=IC(-7200)	; 20 转，以机床数据定义的速度 ; 在负方向上
N350 COUPOF (S2, S1)	; 快速撤销同步，S = S2 = 3000
N355 SPOSA[2]=0	; 副主轴停止在零度位置上
N360 G0 X0 Y0	
N365 WAITS(2)	; 等待主轴 2
N370 M5	; 停止副主轴
N375 M30	

13.7 数据表

13.7.1 机床数据

13.7.1.1 NC 专用机床数据

编号	名称: \$MN_	描述
10000	AXCONF_MACHAX_NAME_TAB	机床轴名称

13.7.1.2 通道专用机床数据

编号	名称: \$MC_	描述
20070	AXCONF_MACHAX_USED	通道内有效的机床轴编号
21300	COUPLE_AXIS_1	定义同步主轴对

编号	名称: \$MC_	描述
21310	COUPLING_MODE_1	主轴同步方式
21320	COUPLE_BLOCK_CHANGE_CTRL_1	主轴同步中的程序段切换特性
21330	COUPLE_RESET_MODE_1	同步撤销特性
21340	COUPLE_IS_WRITE_PROT_1	无法修改同步参数

13.7.1.3 进给轴/主轴专用机床数据

编号	名称: \$MA_	描述
30455	MISK_FUNKTION_MASK	轴功能
30550	AXCONF_ASSIGN_MASTER_CHAN	用于跨通道取轴的通道缺省设置
32200	POSCTRL_GAIN	KV 系数
32400	AX_JERK_ENABLE	单轴加加速度上限
32410	AX_JERK_TIME	用于单轴加加速度滤波器的时间常数
32420	JOG_AND_POS_JERK_ENABLE	单轴加加速度上限的缺省设置
32430	JOG_AND_POS_MAX_JERK	单轴加加速度
32610	VELO_FFW_WEIGHT	转速前馈控制系数
32620	FFW_MODE	前馈控制类型
32650	AX_INERTIA	转矩前馈控制的转动惯量
32800	EQUIV_CURRCTRL_TIME	电流环前馈控制的等效时间常数
32810	EQUIV_SPEEDCTRL_TIME	转速环前馈控制的等效时间常数
34080	REFP_MOVE_DIST	参考点间距
34090	REFP_MOVE_DIST_CORR	参考点偏移
34100	REFP_SET_POS	参考点值
35000	SPIND_ASSIGN_TO_MACHAX	分配主轴至机床主轴
37200	COUPLE_POS_TOL_COARSE	“粗同步”的阈值
37210	COUPLE_POS_TOL_FINE	“精同步”的阈值
37220	COUPLE_VELO_TOL_COARSE	主轴和副主轴的“粗”速度公差
37230	COUPLE_VELO_TOL_FINE	主轴和副主轴的“精”速度公差

13.7.2 设定数据

13.7.2.1 通道专用设定数据

编号	名称: \$SC_	描述
42300	COUPLE_RATIO_1	主轴同步的传动比

13.7.3 信号

13.7.3.1 发送至通道的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
NC 启动	DB21, ...DBX7.1	DB3200.DBX7.1
NC 停止进给轴和主轴	DB21, ...DBX7.4	DB3200.DBX7.4

13.7.3.2 来自通道的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
空运行进给已选择	DB21, ...DBX24.6	DB1700.DBX0.6
快速移动进给率修调已选择	DB21, ...DBX25.3	DB1700.DBX1.3

13.7.3.3 到进给轴/主轴的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
进给轴/主轴禁用	DB31, ...DBX1.3	DB380x.DBX1.3
跟踪方式	DB31, ...DBX1.4	DB380x.DBX1.4
位置测量系统 1, 位置测量系统 2	DB31, ...DBX1.5/6	DB380x.DBX1.5/6
伺服使能	DB31, ...DBX2.1	DB380x.DBX2.1
剩余行程/主轴复位	DB31, ...DBX2.2	DB380x.DBX2.2
主轴停止/进给停止	DB31, ...DBX4.3	DB380x.DBX4.3
JOG 模式下的移动键	DB31, ...DBX4.6/7	DB380x.DBX4.6/7
重新同步主轴 1, 重新同步主轴 2	DB31, ...DBX16.4/5	DB380x.DBX2000.4/5

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
删除 S 值	DB31, ...DBX16.7	DB380x.DBX2000.7
进给倍率有效	DB31, ...DBX17.0	DB380x.DBX2001.0
M3/M4 取反	DB31, ...DBX17.6	DB380x.DBX2001.6
主轴倍率	DB31, ...DBB19	DB380x.DBB2003
重新同步	DB31, ...DBX31.4	-
禁用同步	DB31, ...DBX31.5	-

13.7.3.4 来自进给轴/主轴的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
已回参考点/已同步 1/ 2	DB31, ...DBX60.4/5	DB390x.DBX0.4/5
主轴同步	DB31, ...DBX84.4	DB390x.DBX2002.4
精同步	DB31, ...DBX98.0	-
粗同步	DB31, ...DBX98.1	-
实际值同步	DB31, ...DBX98.2	-
叠加运动	DB31, ...DBX98.4	DB390x.DBX5002.4
主主轴/主动轴激活	DB31, ...DBX99.0	-
副主轴/从动轴激活	DB31, ...DBX99.1	-

13.7.4 系统变量

系统变量	描述
\$P_COUP_OFFS[副主轴]	程序中编写的副主轴和主主轴之间的偏移
\$AA_COUP_OFFS[副主轴]	副主轴和主主轴之间设定位置的偏移
\$VA_COUP_OFFS[副主轴]	副主轴和主主轴之间实际位置的偏移

对系统变量的详细说明请见：

文档：

参数手册“系统变量”

S7:存储器配置

14.1 简要说明

存储器类型

NC 需要一个静态和一个动态存储器用于进行数据维护和管理:

- **静态 NC 存储器**

在静态 NC 存储器中, 程序数据(零件程序、循环等)以及当前的系统数据和用户数据(刀具管理、全局用户数据等)都是**永久**保存的。

- **动态 NC 存储器**

动态 NC 存储器中的数据是**非永久**保存的。因此它通常用于保存由 NC 生成的、只用于某个特定的时间段内的数据(例如宏指令、局部用户数据、插补缓冲器等)。

存储器结构

在静态和动态 NC 存储器中, 每个数据组的存储区域都有指定的容量, 该存储区域的容量是在配置存储器时固定设置的。

这种存储器结构形式也就保证了控制系统具有**确定性**: 在执行零件程序的任意一个时间点上都有预留的存储区域。

存储器配置

第一次启动系统时, 用于配置存储器的机床数据为缺省值。通常情况下这种缺省配置足够使用。 机床制造商/用户可随时对缺省配置进行修改(**重新配置**)。

14.2 存储器结构

14.2.1 主动和被动文件系统

主动和被动文件系统位于静态 NC 存储器中。

主动文件系统

主动文件系统包含了用于设置 NCK 的系统数据:

- 机床数据
- 设定数据
- 选件数据
- 全局用户数据(GUD)
- 刀具补偿数据/刀库数据
- 保护区
- R 参数
- 零点偏移/FRAME
- 垂度补偿
- 象限误差补偿
- 丝杠螺距误差补偿

这些数据表示的是 NCK 的当前工作数据。

从用户的角度来看，主动文件系统是以数据为单位构建的。

被动文件系统

被动文件系统包含了加载在 NCK 中的所有数据:

- 主程序
- 子程序
- 全局用户数据和宏指令的定义文件(*.DEF)
- 标准循环
- 用户循环
- 工件
- 注释

从用户的角度来看，被动文件系统是以文件为单位构建的。

14.2.2 重新配置

重新配置

注意
数据丢失 重新配置静态 NC 存储器会导致主动和被动文件系统中的数据丢失。 因此，在激活新的存储器配置之前必须创建一份批量调试文件来备份旧的存储器配置。

执行以下某项操作后需要重新配置静态和/或动态 **NC** 存储器:

- 修改了用于配置存储器的机床数据:
MD... \$...**MM**...
- 修改了通道数量

14.3 配置静态用户存储器

14.3.1 静态 NC 存储器的结构

下图展示的是 SINUMERIK 840D sl 中静态 NC 存储器的结构:

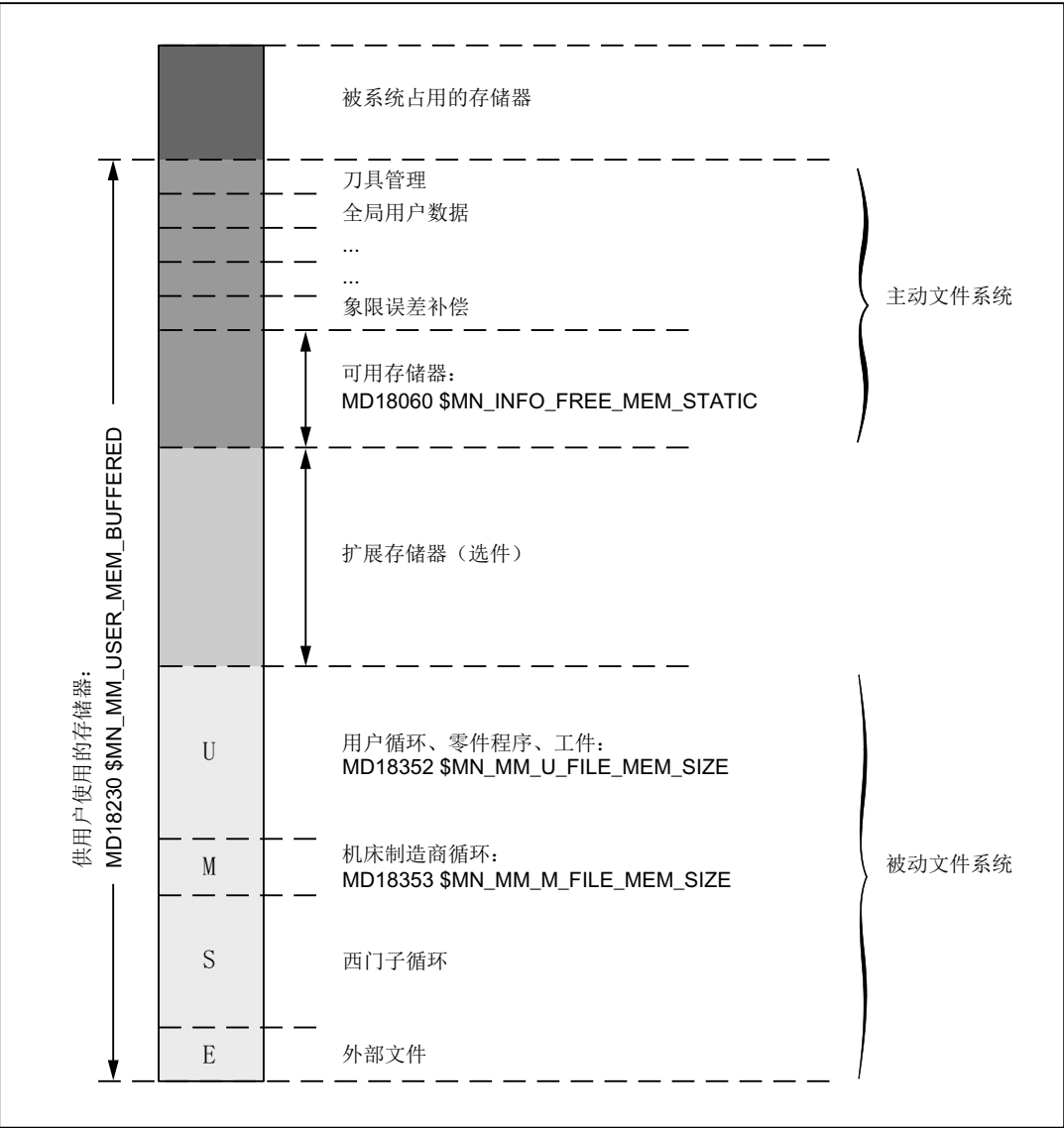


图 14-1 SINUMERIK 840D sl 中的静态 NC 存储器

静态用户存储器

静态 NC 存储器可供系统和用户一同使用。

供用户使用的存储区域称为“静态用户存储器”。它包含了主动和被动文件系统的数据库。

静态用户存储器容量

静态用户存储器的容量可在机床数据中进行设置：

MD18230 \$MN_MM_USER_MEM_BUFFERED

用于被动文件系统的存储器容量

被动文件系统的存储器有指定容量，有以下分区：

分区	内容
S (Siemens = 系统制造商)	_N_CST_DIR(西门子循环)目录下的文件
M (Manufacturer = 机床制造商)	_N_CMA_DIR(机床制造商循环)目录下的文件
U (User = 最终用户)	_N_CUS_DIR(用户循环、零件程序、工件)目录下的文件
E (EXT = 外部)	外部文件(例如:USB 闪存驱动器 或 CD 光盘中的零件程序)。 提示: 为了避免与内部零件程序重名，从外部数据载体载入的零件程序应保存在一个单独的目录下 (_N_EXT_DIR)。

分区优点

通过分区可确保在 NCK 软件升级或循环包升级后（该过程会导致西门子循环存储区扩大），批量开机调试文件能够再次加载到 NCK 中。

分区大小

分区 **S** 和 **E** 的大小是预先设置好的，不可更改。

剩下的存储器可任意划分为两个分区：**U** 和 **M**。可通过以下机床数据进行设置：

MD18352 \$MN_MM_U_FILE_MEM_SIZE

（用于零件程序/循环/文件的最终用户存储器容量）

14.3 配置静态用户存储器

MD18353 \$MN_MM_M_FILE_MEM_SIZE

(用于机床制造商循环/文件的存储器容量)

可设置的最大值取决于:

- 系统以及可用存储器容量 (包括扩展存储器选件)
- 指定的最大值, 参见:

文档:

详细机床数据说明

用于主动文件系统的存储器容量

用于主动文件系统的存储器被划分为不同的数据区: 刀具管理、全局用户数据等, 每个数据区容量都可通过机床数据来设置。

剩余的可用存储器容量显示在机床数据中:

MD18060 \$MN_INFO_FREE_MEM_STATIC

(未使用的静态存储器)

可通过相应的机床数据适当扩大主动文件系统各个数据区的容量, 直到占满该存储器。

说明

在操作界面上的“调试”操作区中会显示修改数据区需要占用的存储器。借助该数据调试人员就可以了解计划的修改实际需要占用多少存储器。

扩展存储器 (选件)

附加的静态用户存储器作为选件供机床制造商选购。

该附加存储器可根据扩大分区 U 和 M 或扩大主动文件系统存储器的需求来投入使用。

14.3.2 调试

步骤

1. 载入标准机床数据。
2. 设置机床数据
MD18230 \$MN_MM_USER_MEM_BUFFERED
，值要大于标配存储器容量和附加存储器容量之和。

3. 执行 NCK 复位。
系统会发出报警 6030 “最大用户存储器被修改”，并将供用户使用的最大存储器输入到 MD18230 中(包括附加存储器)。所有其他存储器配置涉及的机床数据都变为缺省值。
4. 在以下机床数据中设置分区 U 和 M 的容量:
MD18352 \$MN_MM_U_FILE_MEM_SIZE
(用于零件程序/循环/文件的最终用户存储器容量)
MD18353 \$MN_MM_M_FILE_MEM_SIZE
(用于机床制造商循环/文件的存储器容量)
5. 激活所需通道和轴的数量。
6. 可通过扩大/缩小主动文件系统的各个数据区(刀具管理、全局用户数据等)的容量来自定义该存储器的分区。
 - 可在以下机床数据中读取可用静态用户存储器:
MD18060 \$MN_INFO_FREE_MEM_STATIC
(未使用的静态存储器)
 - 设置机床数据(参见章节“数据表 (页 804)”)文档:
详细机床数据说明
7. 执行 NCK 复位。
存储器现在成功重新配置。

14.4 配置动态用户存储器

14.4.1 动态 NC 存储器的结构

下图展示的是动态 NC 存储器的结构:

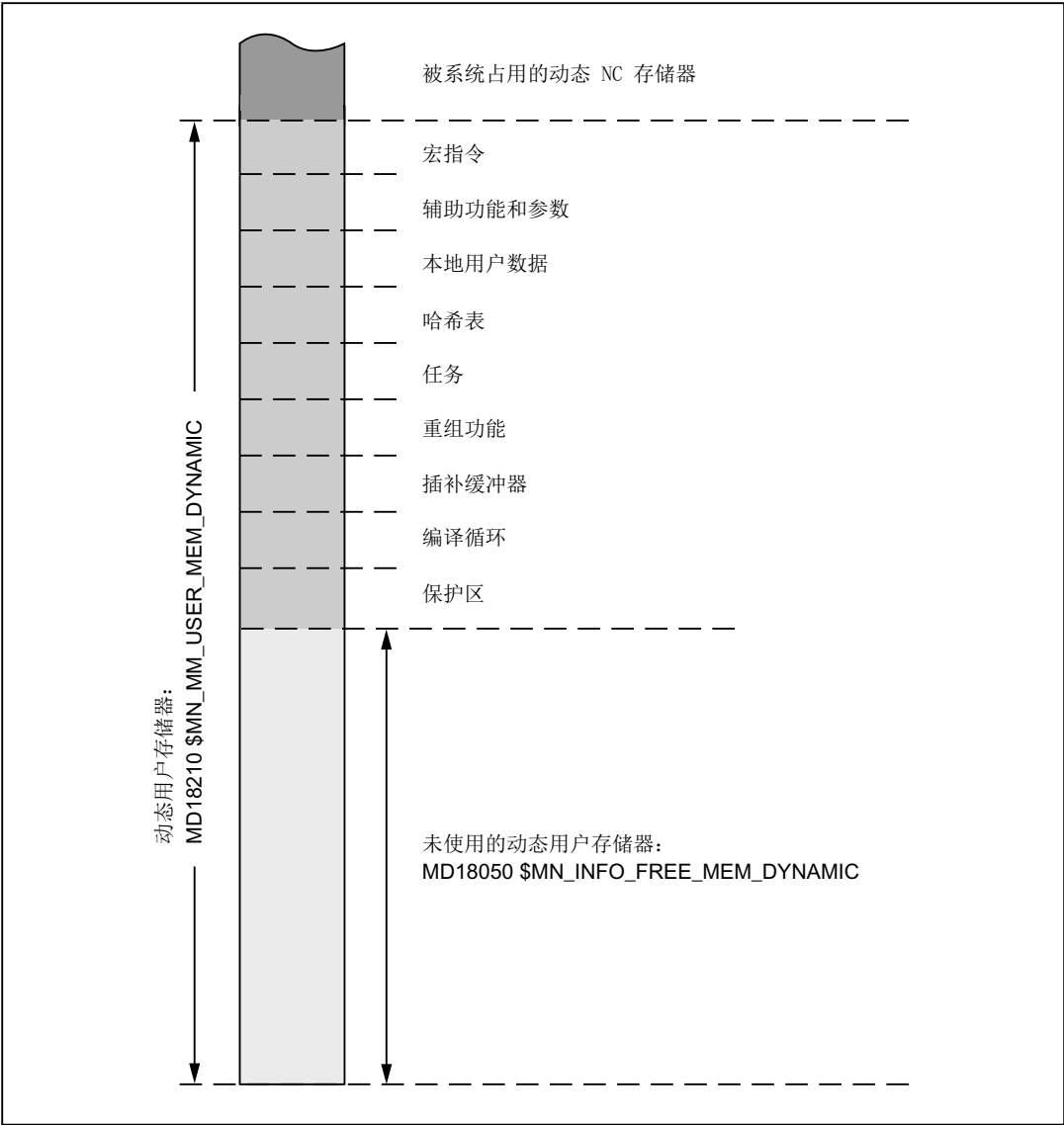


图 14-2 动态 NC 存储器

动态用户存储器

动态 NC 存储器可供系统和用户一同使用。供用户使用的存储区域称为“动态用户存储器”。

动态用户存储器的容量

可通过以下机床数据来设置动态用户存储器的容量:

MD18210 \$MN_MM_USER_MEM_DYNAMIC

由于重新配置时系统会自动设置一个合理值,所以通常情况下无需进行修改。

未使用的动态用户存储器

未使用的动态用户存储器会显示在以下机床数据中:

MD18050 \$MN_INFO_FREE_MEM_DYNAMIC

(未使用的动态存储器)

该机床数据指出了每条通道还剩余多大存储器可用于扩大用户数据区(局部用户数据、插补缓冲器等)。

14.4.2 调试

可通过扩大/缩小各个存储区来修改默认的存储器分区。

步骤

1. 可在以下机床数据中读取未使用的动态用户存储器:
MD18050 \$MN_INFO_FREE_MEM_DYNAMIC
(未使用的动态存储器)
2. 设置机床数据(参见章节“数据表 (页 804)”) **文档:**
详细机床数据说明
3. 执行 NCK 复位。
存储器现在成功重新配置。

14.5 数据表

14.5.1 机床数据

14.5.1.1 通用机床数据

编号	名称: \$MN_	描述
10134	MM_NUM_MMC_UNITS	可同时支持的 HMI 通讯节点的数量
10850	MM_EXTERN_MAXNUM_OEM_GCODES	OEM G 代码的最大数量
10880	MM_EXTERN_CNC_SYSTEM	定义需要调整的控制系统
10881	MM_EXTERN_GCODE_SYSTEM	ISO_3 模式: G 代码系统
18050	INFO_FREE_MEM_DYNAMIC	未使用的动态存储器
18060	INFO_FREE_MEM_STATIC	未使用的静态存储器
18070	INFO_FREE_MEM_DPR	DUAL-PORT-RAM 中 未使用的存储器
18072	INFO_FREE_MEM_CC_MD	CC MD 存储器中未使用的存储器
18078	MM_MAX_NUM_OF_HIERARCHIES	刀库位置类型可定义的最大等级数
18079	MM_MAX_HIERARCHY_ENTRIES	一个刀库位置类型级中允许的最大条目数
18080	MM_TOOL_MANAGEMENT_MASK	预留给刀具管理使用的存储器
18082	MM_NUM_TOOL	由 NCK 管理的刀具的数量
18084	MM_NUM_MAGAZINE	由 NCK 管理的刀库的数量
18086	MM_NUM_MAGAZINE_LOCATION	刀库位置数
18088	MM_NUM_TOOL_CARRIER	最大可定义刀架数量
18090	MM_NUM_CC_MAGAZINE_PARAM	编译循环刀具管理: 刀库数据数
18092	MM_NUM_CC_MAGLOC_PARAM	编译循环刀具管理: 刀库位置数据数
18094	MM_NUM_CC_TDA_PARAM	编译循环刀具管理: TDA 数据数
18096	MM_NUM_CC_TOA_PARAM	编译循环刀具管理: TOA 数据数
18098	MM_NUM_CC_MON_PARAM	编译循环刀具管理: 监视器数据数
18100	MM_NUM_CUTTING_EDGES_IN_TOA	NCK 中的刀具补偿数

编号	名称: \$MN_	描述
18102	MM_TYPE_OF_CUTTING_EDGE	D 号编程方式
18104	MM_NUM_TOOL_ADAPTER	TO 区域中的刀具适配器
18105	MM_MAX_CUTTING_EDGE_NO	D 号最大值
18106	MM_MAX_CUTTING_EDGE_PERTOOL	每把刀具的最大 D 号数
18108	MM_NUM_SUMCORR	TO 区域中的总补偿
18110	MM_MAX_SUMCORR_PER_CUTTEDGE	每个刀沿的最大总补偿数
18112	MM_KIND_OF_SUMCORR	TO 区域中的总补偿特性
18114	MM_ENABLE_TOOL_ORIENT	指定刀沿定向
18116	MM_NUM_TOOL_ENV	TO 区域中的刀具环境数
18118	MM_NUM_GUD_MODULES	全局用户数据模块数
18120	MM_NUM_GUD_NAMES_NCK	全局用户变量数
18130	MM_NUM_GUD_NAMES_CHAN	通道专用的用户变量数
18140	MM_NUM_GUD_NAMES_AXIS	轴专用的用户变量数
18150	MM_GUD_VALUES_MEM	全局用户变量的存储器容量
18160	MM_NUM_USER_MACROS	宏指令数
18170	MM_NUM_MAX_FUNC_NAMES	附加功能数
18180	MM_NUM_MAX_FUNC_PARAM	附加参数数
18190	MM_NUM_PROTECT_AREA_NCK	NCK 中保护区的数量
18200	MM_NUM_CCS_MAGAZINE_PARAM	西门子/OEM 刀库数据数
18201	MM_TYPE_CCS_MAGAZINE_PARAM	西门子/OEM 刀库数据类型
18202	MM_NUM_CCS_MAGLOC_PARAM	西门子/OEM 刀库位置数据数
18203	MM_TYPE_CCS_MAGLOC_PARAM	西门子/OEM 刀库位置数据类型
18204	MM_NUM_CCS_TDA_PARAM	西门子/OEM 刀具数据数
18205	MM_TYPE_CCS_TDA_PARAM	西门子/OEM 刀具数据类型
18206	MM_NUM_CCS_TOA_PARAM	每个刀沿的西门子/OEM 数据数
18207	MM_TYPE_CCS_TOA_PARAM	每个刀沿 西门子/OEM 数据类型
18208	MM_NUM_CCS_MON_PARAM	西门子/OEM 监视器数据数
18209	MM_TYPE_CCS_MON_PARAM	西门子/OEM 监视器数据类型
18210	MM_USER_MEM_DYNAMIC	DRAM 中的用户存储器
18220	MM_USER_MEM_DPR	双端口 RAM 中的用户存储器

14.5 数据表

编号	名称: \$MN_	描述
18230	MM_USER_MEM_BUFFERED	SRAM 中的用户存储器
18231	MM_USER_MEM_BUFFERED_TYPEOF	数据缓冲工艺
18232	MM_ACTFILESYS_LOG_FILE_MEM	系统: 日志文件大小
18238	MM_CC_MD_MEM_SIZE	SRAM 中的编译循环机床数据
18240	MM_LUD_HASH_TABLE_SIZE	用户变量的哈希表大小
18242	MM_MAX_SIZE_OF_LUD_VALUE	LUD 变量的最大数组大小
18250	MM_CHAN_HASH_TABLE_SIZE	通道专用数据的哈希表大小
18260	MM_NCK_HASH_TABLE_SIZE	全局数据的哈希表大小
18270	MM_NUM_SUBDIR_PER_DIR	子目录数
18280	MM_NUM_FILES_PER_DIR	每个目录的文件数
18290	MM_FILE_HASH_TABLE_SIZE	目录文件的哈希表大小
18300	MM_DIR_HASH_TABLE_SIZE	子目录的哈希表大小
18310	MM_NUM_DIR_IN_FILESYSTEM	被动文件系统中的目录数
18320	MM_NUM_FILES_IN_FILESYSTEM	被动文件系统中的文件数
18332	MM_FLASH_FILE_SYSTEM_SIZE	PCNC 上闪存文件系统的大小
18342	MM_CEC_MAX_POINTS	垂度补偿的最大哈希表
18350	MM_USER_FILE_MEM_MINIMUM	最小的零件程序存储器
18352	MM_U_FILE_MEM_SIZE	用于零件程序/循环/文件的最终用户存储器容量
18353	MM_M_FILE_MEM_SIZE	用于机床制造商循环/文件的存储器容量
18354	MM_S_FILE_MEM_SIZE	用于系统制造商循环/文件的存储器容量
18355	MM_T_FILE_MEM_SIZE	用于临时文件的存储器容量
18356	MM_E_FILE_MEM_SIZE	用于外部文件的存储器容量
18360	MM_EXT_PROG_BUFFER_SIZE	用于执行外部程序的 FIFO 缓冲器大小
18362	MM_EXT_PROG_NUM	可同时执行的外部程序的数量
18370	MM_PROTOD_NUM_FILES	最大日志文件数
18371	MM_PROTOD_NUM_ETPD_STD_LIST	标准数据列表 ETPD 数
18372	MM_PROTOD_NUM_ETPD_OEM_LIST	OEM 数据列表 ETPD 数
18373	MM_PROTOD_NUM_SERVO_DATA	用于记录的伺服数据数
18374	MM_PROTOD_FILE_BUFFER_SIZE	日志文件的缓冲器大小
18375	MM_PROTOD_SESS_ENAB_USER	允许参与会话的用户

编号	名称: \$MN_	描述
18390	MM_COM_COMPRESS_METHOD	支持的压缩方法
18400	MM_NUM_CURVE_TABS	曲线表数量(SRAM)
18402	MM_NUM_CURVE_SEGMENTS	曲线段数量(SRAM)
18403	MM_NUM_CURVE_SEG_LIN	线性曲线段数量(SRAM)
18404	MM_NUM_CURVE_POLYNOMS	曲线表多项式数量(SRAM)
18406	MM_NUM_CURVE_TABS_DRAM	曲线表数量(DRAM)
18408	MM_NUM_CURVE_SEGMENTS_DRAM	曲线段数量(DRAM)
18409	MM_NUM_CURVE_SEG_LIN_DRAM	线性曲线段数量(DRAM)
18410	MM_NUM_CURVE_POLYNOMS_DRAM	曲线表多项式数量(DRAM)
18450	MM_NUM_CP_MODULES	CP 模块的最大数量
18452	MM_NUM_CP_MODUL_LEAD	每个 CP 同步模块的最大主值数
18500	MM_EXTCOM_TASK_STACK_SIZE	外部通讯任务的栈大小(DRAM)
18502	MM_COM_TASK_STACK_SIZE	通讯任务的栈大小, 单位为 kByte(DRAM)
18510	MM_SERVO_TASK_STACK_SIZE	伺服任务的栈大小(DRAM)
18512	MM_IPO_TASK_STACK_SIZE	插补任务的栈大小(DRAM)
18520	MM_DRIVE_TASK_STACK_SIZE	驱动任务的栈大小(DRAM)
18540	MM_PLC_TASK_STACK_SIZE	PLC 任务的栈大小(DRAM)
18600	MM_FRAME_FINE_TRANS	FRAME 时的精位移(SRAM)
18601	MM_NUM_GLOBAL_USER_FRAMES	全局预定义用户框架的数量(SRAM)
18602	MM_NUM_GLOBAL_BASE_FRAMES	全局基本框架的数量(SRAM)
18660	MM_NUM_SYNACT_GUD_REAL	可定义的 REAL 型全局用户数据变量的数量
18661	MM_NUM_SYNACT_GUD_INT	可定义的 INT 型全局用户数据变量的数量
18662	MM_NUM_SYNACT_GUD_BOOL	可定义的 BOOL 型全局用户数据变量的数量
18663	MM_NUM_SYNACT_GUD_AXIS	可定义的 AXIS 型全局用户数据变量的数量
18664	MM_NUM_SYNACT_GUD_CHAR	可定义的 CHAR 型全局用户数据变量
18665	MM_NUM_SYNACT_GUD_STRING	可定义的 STRING 型全局用户数据变量
18700	MM_SIZEOF_LINKVAR_DATA	NCU-Link 变量存储器的容量
18710	MM_NUM_AN_TIMER	用于同步动作的全局时间变量的数量
18720	MM_SERVO_FIFO_SIZE	插补器和位置闭环控制之间的目标缓冲器大小
18780	MM_NCU_LINK_MASK	激活 NCU-Link 通讯

14.5 数据表

编号	名称: \$MN_	描述
18781	NCU_LINK_CONNECTIONS	内部 Link 数
18782	MM_LINK_NUM_OF_MODULES	NCU-Link 模块的数量
18790	MM_MAX_TRACE_LINK_POINTS	用于 NCU-Link 的跟踪数据缓冲器的大小
18792	MM_TRACE_LINK_DATA_FUNCTION	指定 NCU-Link 文件的内容
18794	MM_TRACE_VDI_SIGNAL	VDI 信号的跟踪特性
18800	MM_EXTERN_LANGUAGE	激活外部 NC 语言
18860	MM_MAINTENANCE_MON	激活维护数据的记录
18870	MM_MAXNUM_KIN_CHAINS	最大运动链数
18880	MM_MAXNUM_KIN_CHAIN_ELEM	最大运动链单元数
18890	MM_MAXNUM_3D_PROT_AREAS	3D 保护区中的最大单元数
18892	MM_MAXNUM_3D_PROT_AREA_ELEM	最大保护区单元数
18894	MM_MAXNUM_3D_PROT_GROUPS	最大保护区组数
18896	MM_MAXNUM_3D_COLLISION	用于碰撞检查的最大临时存储器容量

14.5.1.2 通道专用机床数据

编号	名称: \$MC_	描述
20096	T_M_ADDRESS_EXIT_SPINO	主轴号作为地址扩展名
27900	REORG_LOG_LIMIT	用于使能日志文件的插补缓冲器大小, %
28000	MM_REORG_LOG_FILE_MEM	用于重组的存储器容量
28010	MM_NUM_REORG_LUD_MODULES	重组时局部用户变量的模块数量
28020	MM_NUM_LUD_NAMES_TOTAL	局部用户变量的数量
28040	MM_LUD_VALUES_MEM	局部用户变量的存储器容量
28050	MM_NUM_R_PARAM	通道专用 R 参数的数量
28060	MM_IPO_BUFFER_SIZE	插补缓冲器中的 NC 程序段数量
28070	MM_NUM_BLOCKS_IN_PREP	预处理的程序段数量
28080	MM_NUM_USER_FRAMES	可设置的框架数
28081	MM_NUM_BASE_FRAMES	基本框架数(SRAM)
28082	MM_SYSTEM_FRAME_MASK	系统框架(SRAM)
28083	MM_SYSTEM_DATAFRAME_MASK	系统框架(SRAM)
28085	MM_LINK_TOA_UNIT	TO 单元至通道的分配(SRAM)

编号	名称: \$MC_	描述
28090	MM_NUM_CC_BLOCK_ELEMENTS	用于编译循环的程序段单元数
28100	MM_NUM_CC_BLOCK_USER_MEM	用于编译循环的程序段存储器的容量
28105	MM_NUM_CC_HEAP_MEM	用于编译循环的堆存储器容量(DRAM), 单位: kByte
28150	MM_NUM_VDIVAR_ELEMENTS	用于写入 PLC 变量的单元数
28160	MM_NUM_LINKVAR_ELEMENTS	用于写入 NCU Link 变量的单元数
28180	MM_MAX_TRACE_DATAPOINTS	跟踪数据缓冲器的大小
28200	MM_NUM_PROTECT_AREA_CHAN	用于通道专用保护区的模块数
28210	MM_NUM_PROTECT_AREA_ACTIVE	同时激活的保护区数
28212	MM_NUM_PROTECT_AREA_CONTOUR	用于有效保护区的单元(DRAM)
28250	MM_NUM_SYNC_ELEMENTS	用于同步动作中表达式的单元数
28252	MM_NUM_FCTDEF_ELEMENTS	FCTDEF 单元数
28254	MM_NUM_AC_PARAM	\$AC_PARAM。
28255	MM_BUFFERED_AC_PARAM	\$AC_PARAM[] 保存在 SRAM 中。
28256	MM_NUM_AC_MARKER	\$AC_MARKER
28257	MM_BUFFERED_AC_MARKER	\$AC_MARKER[] 保存在 SRAM 中。
28258	MM_NUM_AC_TIMER	时间变量 \$AC_TIMER 的数量(DRAM)
28274	MM_NUM_AC_SYSTEM_PARAM	用于运动同步动作的 \$AC_SYSTEM_PARAM 的数量
28276	MM_NUM_AC_SYSTEM_MARKER	用于运动同步动作的 \$AC_SYSTEM_MARKER 的数量
28290	MM_SHAPED_TOOLS_ENABLE	为轮廓刀具使能刀具半径补偿
28300	MM_PROTOD_USER_ACTIVE	为用户激活记录
28301	MM_PROTOD_NUM_ETP_OEM_TYP	OEM 事件类型 ETP 的数量
28302	MM_PROTOD_NUM_ETP_STD_TYP	标准事件类型 ETP 的数量
28400	MM_ABSBLOCK	激活绝对值程序段显示
28402	MM_ABSBLOCK_BUFFER_CONF	确定上传缓冲器的大小
28450	MM_TOOL_DATA_CHG_BUFF_SIZE	刀具数据修改的缓冲器(DRAM)
28500	MM_PREP_TASK_STACK_SIZE	预处理任务的栈的大小
28520	MM_MAX_AXISPOLY_PER_BLOCK	每个程序段轴多项式的最大数量
28530	MM_PATH_VELO_SEGMENTS	用于限制轨迹加速度的存储空间单元数量

14.5 数据表

编号	名称: \$MC_	描述
28535	MM_FEED_PROFILE_SEGMENTS	用于进给曲线的存储器单元数
28540	MM_ARCLENGTH_SEGMENTS	用于显示弧长函数的存储器单元数
28560	MM_SEARCH_RUN_RESTORE_MODE	模拟后恢复数据
28580	MM_ORIPATH_CONFIG	轨迹定向 ORIPATH 的设置

14.5.1.3 进给轴/主轴专用机床数据

编号	名称: \$MA_	描述
38000	MM_ENC_COMP_MAX_POINTS	插补补偿中的控制点数量
38010	MM_QEC_MAX_POINTS	象限误差补偿值的数量

T1: 分度轴

15.1 简要说明

机床上的分度轴

在某些应用情况下要求轴只靠近一些确定的分度点（例如：某个刀位）。这种运行方式无论是在“自动”还是在“调节”方式中都需要使用。

这种轴被称为“分度轴”。分度轴确定的位置被称为“编码位置”或者“分度位置”。

应用

分度轴最常用于刀塔、链式刀库或者盒式刀库上。其中，一个编码位置代表一个刀位。在换刀时，刀库定位到需要新刀的刀位上。

显示分度

通过系统变量可以查询以下数据：

- 当前分度位置的编号：
 - 进入分度位置的“精准停”窗口
 - 经过到下一分度位置的一半路程
- 编程的分度位置

15.2 运行分度轴

分度轴的运行途径有：

- 在“调节”运行方式 JOG 和 INC 下手动运行
- 使用零件程序中用于“编码位置”的特殊指令
- 由 PLC

在达到分度位置时 NC 向 PLC 输出下列接口信号：

DB31, ... DBX76.6 (分度轴就位)

15.2.1 在 JOG 模式下运行分度轴

回参考点

分度轴的回参考点方式和其他轴类似。参考点不必须和某个分度位置重合。

在 JOG 连续运行和断续运行中只有在分度轴回参考点后，
即 DB31, ... DBX60.4 或 5 (已回参考点/已同步 1 或 2) = 1，
分度轴才运行至分度位置。

例外：在使用手轮时分度轴不运行至分度位置。

如果分度轴未回参考点(DB31, ... DBX60.4 或 5 = 0)，在 JOG 下使用手轮运行分度轴时分度位置失效！

说明

在回参考点前不能在 JOG 下运行设有端面齿功能的分度轴。

JOG 连续运行

- 长按模式激活:

SD41050 \$SN_JOG_CONT_MODE_LEVELTRIGGRD = 1

按下正向/负向运行键后，分度轴开始运行，如同在 JOG 下一样。松开运行键后，分度轴运行至下一个可到达的分度位置，然后停止。

- 点按模式激活:

SD41040 \$SN_JOG_CONT_MODE_LEVELTRIGGRD = 0

只要按下运行键一次(发出首个上升沿)，分度轴就会朝对应方向运行。松开运行键后，轴仍将继续运行。再次按下运行键(发出第二个上升沿)，分度轴继续运行至下一个可到达的分度位置，然后停止。

分度轴通常会默认在长按模式下运行。点按模式在这儿只是第二选择。

如果在到达分度位置后操作人员改变了运行方向，分度轴会首先运行至原方向上的下一分度位置。接着开始相反方向的运行。

关于“JOG 连续运行的点按模式和长按模式”的更多信息(参见章节“H1: 手动运行和手轮运行 (页 155)”)。

JOG 断续运行(INC)

不管当前设置的步数是多少(INC1, ... ,INCvar)，按下正向/负向运行键后，每次分度轴始终只朝所选方向**前进 1 个分度位置**。

在长按模式中松开运行键后，轴运行中断。再次按下运行键后，轴继续运行到下一个分度位置。

而在点按模式中，再次按下运行键后，轴运行会终止。分度轴因此不位于分度位置上。

两个分度位置之间

如果分度轴位于 2 个分度位置间，则在 JOG 方式下按下正向运行键后，轴前进到下一个分度位置。与此类似，按下负向运行键后，轴返回到上一个分度位置。

使用手轮运行

在 JOG 方式下使用手轮运行分度轴时，**分度位置**失效。手轮旋转时，分度轴根据设置的单位（毫米、英寸或者度）运行到任意位置。

如果需要锁定使用手轮运行分度轴这一功能，则可以通过 PLC 用户程序实现锁定。

PLC 信息 “分度轴就位”

分度轴在 JOG 模式下到达分度位置这一状态可由以下 NC/PLC 接口信号指明：

DB31, ... DBX76.6 (分度轴就位)

前提条件：分度轴已回参考点(DB31, ... DBX60.4 或 5 = 1)

JOG 中的报警

如果 JOG 中分度轴的运行超出了分度位置表定义的有效运行区域，系统会发出报警 20054“JOG 中分度轴的索引错误”。

旋转进给率

在 JOG 运行方式下，进给轴/主轴特性也取决于下列设定数据的设置：

SD41100 \$SN_JOG_REV_IS_ACTIVE (JOG 运行方式中的旋转进给率有效)

SD41100	含义
= 1 (有效)	视主轴的情况而定，进给轴/主轴总是采用以下旋转进给率： MD32050 \$MA_JOG_REV_VELO (JOG 旋转进给率) 或者 MD32040 \$MA_JOG_REV_VELO_RAPID (JOG 旋转进给率叠加快速移动速度)
= 0 (无效)	进给轴/ 主轴的特性取决于设定数据： SD43300 \$SA_ASSIGN_FEED_PER_REV_SOURCE (定位轴/主轴的旋转进给率) 若几何轴上有含旋转的有效框架命令，则该几何轴特性取决于设定数据： SD42600 \$SC_JOG_FEED_PER_REV_SOURCE

15.2.2 在 AUTO 模式下运行分度轴

运行到任意位置

在 AUTO 模式下，定义为分度轴的轴可以借助 NC 零件程序运行到**任意位置**。轴也可以运行到定义的分度位置之间的某个位置。

该位置的编程也采用其他轴通用的单位（毫米、英寸或度）。通用的编程指令(G90, G91, AC, IC, ...)在编程手册中详细说明。

运行到“编码位置”

此外，分度轴也可以使用零件程序中的特殊指令运行到“编码位置”：

指令	结果
CAC	编码位置是绝对位置，轴运行到该位置
CACP	编码位置是绝对位置，轴朝正向运行到该位置
CACN	编码位置是绝对位置，轴朝负向运行到该位置
CIC	编码位置是增量位置，轴运行到该位置
CDC	以最短行程运行到编码位置

无论是以绝对方式还是以增量方式定位，都需要编程正向/负向上的待运行分度数量。

如果轴为回转轴，轴可以按最短行程(CDC)或者指定的旋转方向(CACP, CACN)直接运行到分度位置。

到达分度位置

在进入“精准停”窗口并且分度轴运行到某个分度位置后，以下 NC/PLC 接口信号会置位，不管分度轴是如何就位的：

DB31, ... DBX76.6 (分度轴就位)

15.2.3 PLC 运行分度轴

分度轴也可以由 PLC 用户程序运行。

有以下方式：

- 并行定位轴
在此可以由 PLC 指定目标分度位置(参见章节“P2: 定位轴 (页 639)”)。

- 异步子程序 (ASUB)

文档：

功能手册之基本功能分册；章节“BAG、通道、程序运行、复位特性(K1)”

15.3 设置分度轴

分度轴的定义

轴（线性轴或回转轴）可由轴机床数据定义为分度轴：

MD30500 \$MA_INDEX_AX_ASSIGN_POS_TAB

值	含义
0	该轴不是分度轴。
1	该轴是分度轴。对应的分度位置保存在分度位置表 1 中。
2	该轴是分度轴。对应的分度位置保存在分度位置表 2 中。

分度位置表

每根分度轴的分度位置（单位：毫米或度）都必须通过机床数据以位置表的形式加以定义。

15.3 设置分度轴

可定义 2 个分度位置表:

MD10910 \$MN_INDEX_AX_POS_TAB_1 [n] (分度位置表 1)

MD10930 \$MN_INDEX_AX_POS_TAB_2 [n] (分度位置表 2)

说明

一个分度位置表可包含多个轴。前提是这些分度轴都是同一类型的轴(线性轴、回转轴、360°模数轴)。

分度位置数

在每个分度位置表中最多可以输入 60 个位置:

[n = 0 ... 59]

实际使用的位置数可由以下机床数据设置:

MD10900 \$MN_INDEX_AX_LENGTH_POS_TAB_1 (分度位置表 1 的位置数)

MD10920 \$MN_INDEX_AX_LENGTH_POS_TAB_2 (分度位置表 2 的位置数)

说明

分度位置表中超出上述数量(MD10900 或 MD10920)的位置无效。

有效单位制

MD10900 和 MD10920 定义的分度位置采用的是为位置表定义的单位制:

MD10270 \$MN_POS_TAB_SCALING_SYSTEM

值	单位制
0	公制
1	英制

说明

MD10270 也会影响以下设定数据:

SD41500 \$SN_SW_CAM_MINUS_POS_TAB_1 (挡块下降沿 1-8 的切换点)

...

SD41507 \$SN_SW_CAM_PLUS_POS_TAB_4 (挡块上升沿 25-32 的切换点)

指定分度位置

需要遵循以下规则:

- 在表格中以递升顺序(从负向运行区开始到正向运行区)、零间距地输入分度位置。
- 两个连续的位置值不能为相同值。
- 轴位置是基本坐标系中的位置。

模数回转轴定义为分度轴

将分度轴定义为 360°模数回转轴:

MD30300 \$MA_IS_ROT_AX = 1

与

MD30310 \$MA_ROT_IS_MODULO = 1

在这种情况下, 指定分度位置时还应注意以下几点:

- 允许范围: $0^\circ \leq \text{分度位置} < 360^\circ$
- 由于分度轴被定义为连续旋转的回转轴, 在轴到达表中最高号的分度位置, 然后按下正向运行键后, 轴会朝正向继续运行到分度位置 1。与此类似, 按下负向运行键后, 轴会从分度位置 1 朝负向返回到最高号的分度位置。

15.4 编程分度轴

编码位置

借助 NC 零件程序定位分度轴时可以使用两个特殊的指令，指令不采用毫米/度编写，而是采用分度位置号编写(例如：刀位号)。这些特殊指令的可用性取决于轴的类型(即线性轴或回转轴)：

指令	结果	可用性
CAC (i)	编码位置是绝对位置，轴运行到该位置	线性轴、回转轴
CACP (i)	编码位置是绝对位置，轴朝正向运行到该位置	回转轴
CACN (i)	编码位置是绝对位置，轴朝负向运行到该位置	回转轴
CDC (i)	以 最短行程 运行到 编码位置	回转轴
CIC (i)	编码位置是增量位置，轴运行到该位置	线性轴、回转轴

i: 编码位置(分度位置)
i 的取值范围: 0 ... 59；整数(在 CIC 指令中正负值都可)

示例

程序代码	注释
POS[B]=CAC(20)	；分度轴 B 轴运行到 绝对 分度位置 20。运行方向取决于当前实际位置。
程序代码	注释
POS[B]=CACP(10)	；分度轴 B 轴 朝正向 运行到绝对分度位置 10，该情况仅适用于回转轴。
程序代码	注释
POS[B]=CACN(10)	；分度轴 B 轴 朝负向 运行到绝对分度位置 10，该情况仅适用于回转轴。
程序代码	注释
POS[B]=CDC(50)	；分度轴 B 轴 沿着最短行程 运行到分度位置 50，该功能仅适用于回转轴。
程序代码	注释
POS[B]=CIC(-4)	；分度轴 B 轴从当前分度位置 朝负向增量 运行 4 个分度位置。

程序代码	注释
POS[B]=CIC(35)	; 分度轴 B 轴从当前分度位置朝正向增量运行 35 个分度位置。

特殊性

- 模数回转轴定义为分度轴
如果分度轴是模数回转轴，轴会直接运行到下一个分度位置，分度位置除以 360°，不计余数。
- 分度轴位于两个分度位置之间
上述说明的定位指令在 AUTO 运行模式中具有以下作用：

POS[B]=CIC(1) 运行到下一个更高号的分度位置。
 POS[B]=CIC(-1) 运行到下一个更低号的分度位置。
 POS[B]=CIC(0) 不运行分度轴。

显示分度位置

最后编程的分度位置号可由以下系统变量读取：

\$AA_PROG_INDEX_AX_POS_NO

最后越过的分度位置号可由以下系统变量显示：

\$AA_ACT_INDEX_AX_POS_NO

该显示取决于机床数据中的设置：

MD10940 \$MN_INDEX_AX_MODE (分度位置设置)

位	值	含义
0	0	一旦轴到达一个分度位置（进入“精准停”窗口），分度位置的显示便立即变化。也就是说：在轴从一个分度位置运行到下一个分度位置期间，显示不变化。
	1	一旦轴到达一半的分度位置，显示便立即变化。 因此，在满足以下前提时位置显示是一个以分度位置为中心的对称范围：线性轴等分、模数轴的分度范围是模数的整数倍(MD30330 \$MA_MODULO_RANGE)，否则分度位置不是位置显示的中心，而是有一定偏移。 分度轴是模数回转轴、位于最后一个分度位置和第一个分度位置间时，显示按第一个和最后一个显示范围的长度比划分。

下图清晰地展示了位 0 = 0 和位 0 = 1 时的区别:

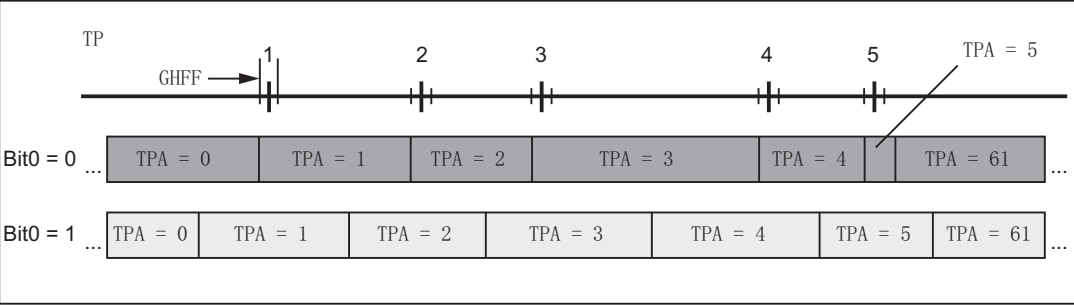
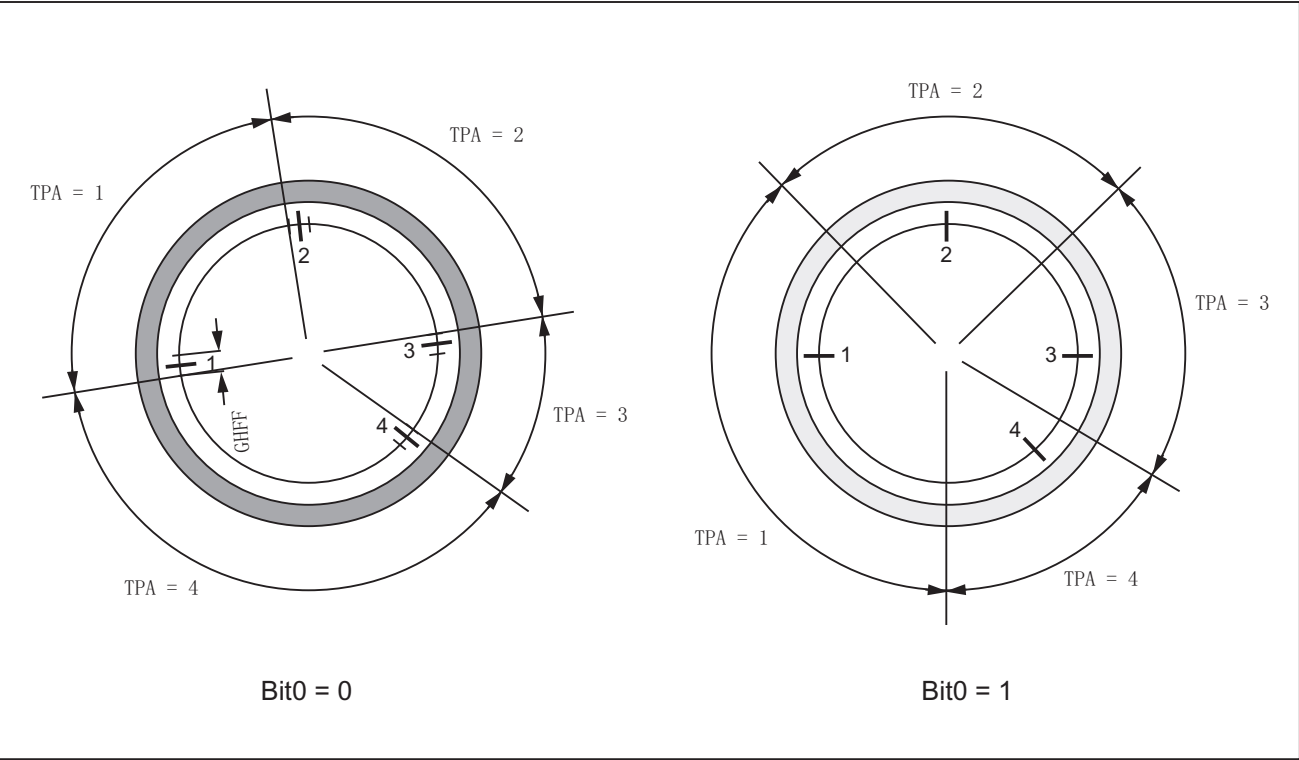


图 15-1 分度位置显示: 线性轴



TP 编程的分度位置

TPA 显示的分度位置

GHFF “精准停” 窗口

图 15-2 分度位置显示: 模数回转轴

\$AA_ACT_INDEX_AX_POS_NO 的取值范围

由系统变量 \$AA_ACT_INDEX_AX_POS_NO 读出的值的范围:

表中的分度位置			
模数回转轴	1 ... n	没有 0 n 最大为 60	
线性轴	0*, 1, 2, 3, ... 59, 60, 61*	0*:	在总分度范围以下
		61* :	在总分度范围以上

等距分度位置		
模数回转轴	1 ... m	没有 0 m = 分母(分子)
线性轴	... -3, -2, -1, 0, 1, 2, 3, ...	

运行至下一个分度位置

“运行至下一个分度位置”指令的特性取决于机床数据中的设置:

MD10940 \$MN_INDEX_AX_MODE (分度位置设置)

位	值	含义
0	0	分度轴运行至下一个分度位置。
	1	分度轴始终运行至运动方向上的下一个分度位置。

以下例说明:

位 0 = 1, 且轴位于分度位置下方(但在“精准停”窗口之外)。

虽然系统变量 \$AA_ACT_INDEX_AX_POS_NO 显示的是分度位置 2, 但指令“运行至下一位置”不会使轴运行至分度位置 3, 而只是使轴精确地运行到分度位置 2。只有当轴精确地运行到该位置(进入“精准停”窗口)或越过该位置后, 该指令才能使轴运行至下一个分度位置(本例中为分度位置 3)。

分度轴总是先运行至运行方向上的下一个分度位置! 有时指令“运行至下一位置”必须给出两次, 这样分度轴才可以从当前显示的分度位置运行至下一个分度位置号(例如:从分度位置 2 运行至分度位置 3)。

框架

和程序中编写的位置一样，系统会将分度位置表中的位置换算为毫米、英寸或度，因此在分度轴上也可以使用框架指令。

但分度轴上通常不需要使用框架指令，具体情况要视应用而定。因此应在零件程序中禁用分度轴的框架指令和零点偏移。

15.5 等距分度

15.5.1 属性

- 具体有：
- 任意数量的等距分度位置
 - 分度轴机床数据修改的生效方式

等距分度可用于：

- 线性轴
- 模数回转轴
- 回转轴

等分间隔

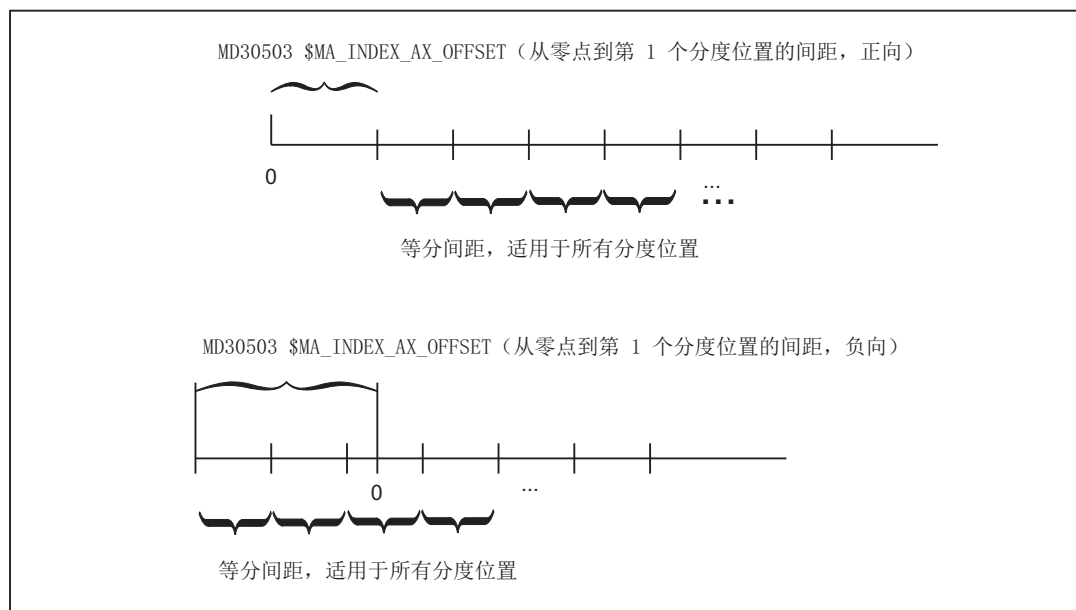
线性轴上的等分间隔是根据以下公式确定的：

距离 =

分子 (MD30501 \$MA_INDEX_AX_NUMERATOR)

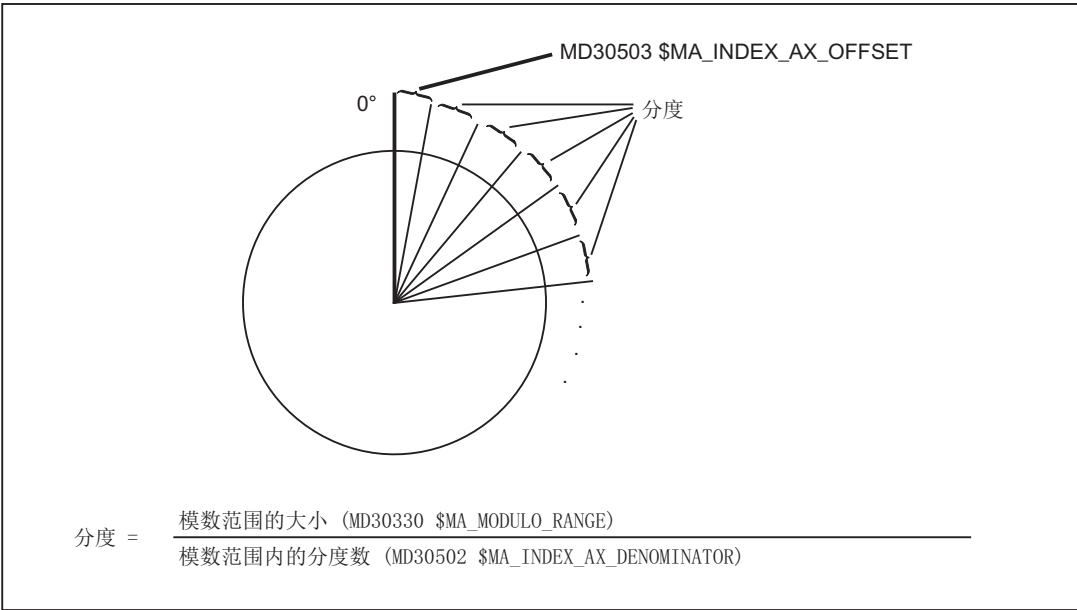
分母 (MD30502 \$MA_INDEX_AX_DENOMINATOR)

线性轴



模数回转轴

$$\text{分度} = \frac{\text{分子 (MD30330 \$MA_MODULO_RANGE)}}{\text{分母 (MD30502 \$MA_INDEX_AX_DENOMINATOR)}}$$



激活

可通过以下设置激活轴（线性轴、模数回转轴或回转轴）的等距分度:

```
MD30500 $MA_INDEX_AX_ASSIGN_POS_TAB[轴] = 3
```

15.5.2 端面齿

功能

通常情况下，使用“端面齿”功能可以通过一个线性轴在回转轴的一些分度位置卡入一个止动销或另一个齿轮来锁定这些位置。只有在精确到达分度位置后，才可以锁定该位置。分度位置在整个范围内等分（即等距）。

前提条件

该回转轴必须是分度轴。该轴必须已回参考点。

文档:

功能手册之基本功能分册；章节“回参考点（R1）”

激活

机床数据:

MD30505 \$MA_HIRTH_IS_ACTIVE(轴是带端面齿的分度轴)

必须置 1。

机床数据:

MD30500 \$MA_INDEX_AX_ASSIGN_POS_TAB (轴是分度轴)

必须置 3(等距分度)。

结果

- 回转轴在所有的运行方式和运行状态下都只能运行到分度位置。
- 在 JOG 下，轴可以连续或断续运行。
前提条件：轴已回参考点。
- 无法采用手轮运行(参见章节“H1: 手动运行和手轮运行 (页 155)”)。
- 在 AUTO、MDA 模式下或是使用 ASUB 时，轴只能运行到分度位置。
- PLC 只能使轴运行到分度位置。

15.5.3 特殊情况下端面齿轴的特性

停止/复位

如果在轴运行期间触发了 NC 停止和复位，轴仍会运行至下一个分度位置。

急停

急停后在纵轴再次下降前，PLC 或操作人员必须在 JOG 下将分度轴再次调整到某个分度位置上。

倍率为 0 或信号“轴停止”

如果当这些事件发生时轴已经离开了一个分度位置，控制系统仍会使轴运行至下一个分度位置。

剩余行程删除

在运行至下一个分度位置后，轴终止运行。

指令轴

如果为已经运动的指令轴输出指令 MOV=0，该轴仍会运行至下一个分度位置。

文档:
功能手册之同步动作分册

指令 MOV

MOV=1 分度轴上带有/没有端面齿。
MOV=0 两种情况下的结果相同，轴都会运行至下一个分度位置。

指令 DELDTG

分度轴上没有端面齿: 轴立即停止。
分度轴上带有端面齿: 轴运行至下一个位置。

15.5.4 限制

转换

定义了端面齿的轴不支持运动转换。

PRESET（预设）

定义了端面齿的轴无法通过 PRESET 设新值。

旋转进给率

定义了端面齿的轴不支持旋转进给率。

行程叠加/速度叠加

定义了端面齿的轴不支持行程叠加/速度叠加。

框架、外部零偏、DRF

定义了端面齿的轴不支持框架指令和插补补偿(如外部零点偏移、DRF 等)。

同步

定义了端面齿的轴不可以为以下类型的轴：

- 主值同步中的从动轴
- 联动轴
- 龙门轴组中的从动轴

文档：

功能手册之特殊功能分册；章节“轴耦合(M3)”

15.5.5 机床数据修改的生效方式

RESET

为以下机床数据设置新值后需要执行复位操作：

MD10900 \$MN_INDEX_AX_LENGTH_POS_TAB_1

MD10920 \$MN_INDEX_AX_LENGTH_POS_TAB_2

MD10910 \$MN_INDEX_AX_POS_TAB_1

MD10930 \$MN_INDEX_AX_POS_TAB_2

MD30500 \$MA_INDEX_AX_ASSIGN_POS_TAB

15.6 调试分度轴

步骤

分度轴的调试方式和普通 NC 轴（线性轴或回转轴）一样。

回转轴

如果回转轴是 360°模数回转轴((MD30300 \$MA_IS_ROT_AX = "1", MD30310 \$MA_ROT_IS_MODULO = "1"), 则也可以运行到分度位置：模数 360°。在分度位置表中只允许输入 0°到 359.999°范围内的位置。否则在启动时将生成报警 4080 “MD [名称] 中的分度轴配置错误”。

可按如下方式设置 360°模数位置显示：

15.6 调试分度轴

MD30320 \$MA_DISPLAY_IS_MODULO = 1

特殊机床数据

此外，还必须设置以下机床数据：

通用机床数据	
MD10900 \$MN_INDEX_AX_LENGTH_POS_TAB_1	分度位置表 1 的位置数量
MD10920 \$MN_INDEX_AX_LENGTH_POS_TAB_2	分度位置表 2 的位置数量
MD10910 \$MN_INDEX_AX_POS_TAB_1 [n]	分度位置表 1
MD10930 \$MN_INDEX_AX_POS_TAB_2 [n]	分度位置表 2

轴机床数据	
MD30500 \$MA_INDEX_AX_ASSIGN_POS_TAB	轴是分度轴(分度位置表 1 或 2 或 3, 等距分度)
MD30505 \$MA_HIRTH_IS_ACTIVE	轴具有“端面齿”的特性
MD30501 INDEX_AX_NUMERATOR	等距分度的分子
MD30502 INDEX_AX_DENOMINATOR	等距分度的分母
MD30503 INDEX_AX_OFFSET	与第 1 个分度位置的间距为零

示例

下文依据两个示例对上述机床数据进行说明。

示例 1：分度轴为回转轴

刀塔，包含 8 个刀位。刀塔被定义为连续旋转的回转轴。8 个刀位间的间距是恒定的。第一个刀位在位置 0°：

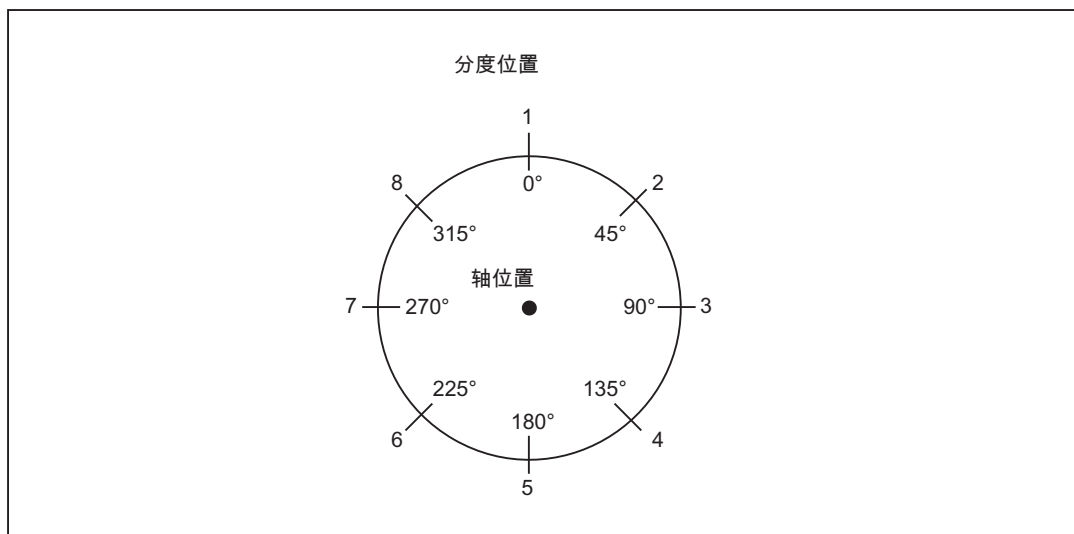


图 15-3 示例：刀塔，包含 8 个刀位

刀塔的分度位置输入到分度位置表 1 中：

MD10910 \$MN_INDEX_AX_POS_TAB_1[0] = 0	；第 1 个分度位置在 0°处
MD10910 \$MN_INDEX_AX_POS_TAB_1[1] = 45	；第 2 个分度位置在 45°处
MD10910 \$MN_INDEX_AX_POS_TAB_1[2] = 90	；第 3 个分度位置在 90°处
MD10910 \$MN_INDEX_AX_POS_TAB_1[3] = 135	；第 4 个分度位置在 135°处
MD10910 \$MN_INDEX_AX_POS_TAB_1[4] = 180	；第 5 个分度位置在 180°处
MD10910 \$MN_INDEX_AX_POS_TAB_1[5] = 225	；第 6 个分度位置在 225°处
MD10910 \$MN_INDEX_AX_POS_TAB_1[6] = 270	；第 7 个分度位置在 270°处
MD10910 \$MN_INDEX_AX_POS_TAB_1[7] = 315	；第 8 个分度位置在 315°处

其他机床数据：

MD10900 \$MN_INDEX_AX_LENGTH_POS_TAB_1 = 8	；表 1 中的 8 个分度位置
MD30500 \$MA_INDEX_AX_ASSIGN_POS_TAB [AX5] = 1	；轴 5 为分度轴
	分度位置位于表 1 中
MD30300 \$MA_IS_ROT_AX [AX5] = 1	；轴 5 是回转轴
MD30310 \$MA_ROT_IS_MODULO [AX5] = 1	；激活了模数转换

示例 2：分度轴为线性轴

工件托盘，包含 10 个位置。

10 个工件位间的距离是恒定的。第一个托盘位在位置 -100 mm 处。

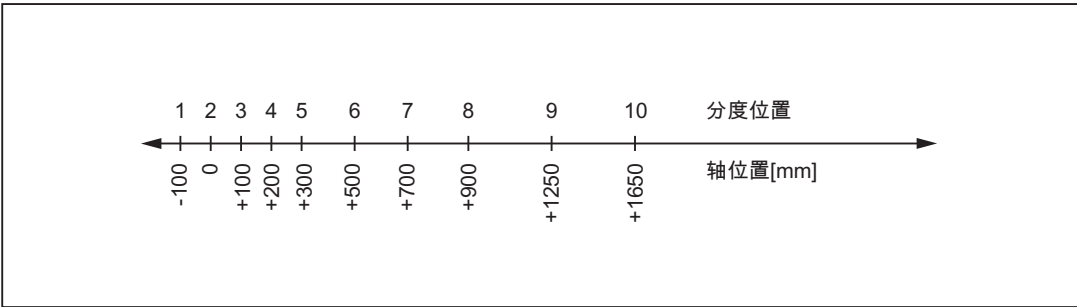


图 15-4 示例：工件托盘是分度轴

工件托盘的分度位置输入到分度表格 2 中。

MD10930 \$MN_INDEX_AX_POS_TAB_2[0] = -100	；第 1 个分度位置在 -100 处
MD10930 \$MN_INDEX_AX_POS_TAB_2[1] = 0	；第 2 个分度位置在 0 处
MD10930 \$MN_INDEX_AX_POS_TAB_2[2] = 100	；第 3 个分度位置在 100 处
MD10930 \$MN_INDEX_AX_POS_TAB_2[3] = 200	；第 4 个分度位置在 200 处
MD10930 \$MN_INDEX_AX_POS_TAB_2[4] = 300	；第 5 个分度位置在 300 处
MD10930 \$MN_INDEX_AX_POS_TAB_2[5] = 500	；第 6 个分度位置在 500 处
MD10930 \$MN_INDEX_AX_POS_TAB_2[6] = 700	；第 7 个分度位置在 700 处
MD10930 \$MN_INDEX_AX_POS_TAB_2[7] = 900	；第 8 个分度位置在 900 处
MD10930 \$MN_INDEX_AX_POS_TAB_2[8] = 1250	；第 9 个分度位置在 1250 处
MD10930 \$MN_INDEX_AX_POS_TAB_2[9] = 1650	；第 10 个分度位置在 1650 处

其他机床数据：

MD10920 \$MN_INDEX_AX_LENGTH_POS_TAB_2=10;	表 2 中的 10 个分度位置
MD30500 \$MA_INDEX_AX_ASSIGN_POS_TAB [AX6] = 2	；轴 6 为分度轴，分度位置位于表 2 中

15.7 分度轴的特点

DRF

分度轴同样可通过 DRF 功能在 AUTO 模式下利用手轮生成一个附加的增量零点偏移。

软件限位开关

分度轴回参考点后的运行期间，软件限位开关也生效。

在 JOG 连续或断续手动运行中，分度轴会停在软件限位开关前面的分度位置上。

回参考点

只有在分度轴回参考点后，

即 DB31, ... DBX60.4 或 60.5 (已回参考点/已同步 1 或 2) = 1,

在 JOG 连续运行和断续运行中分度轴才运行至分度位置。

如果轴未回参考点，

即 DB31, ... DBX60.4 或 60.5 (已回参考点/已同步 1 或 2) = 0,

在手动运行中分度位置失效！

由于分度位置表中保存的轴位置只在轴回参考点后才和实际机床位置一致，因此只要分度轴没有回参考点，就必须锁定 NC 启动。

位置显示

分度轴位置显示的单位为轴通用单位(毫米、英寸或度)。

用 RESET 键终止运行

按下 RESET 键可以使分度轴停止，终止运行。此时，分段轴不再停在某个分度位置上。

15.8 示例

15.8.1 等距分度示例

模数回转轴

MD30502 \$MA_INDEX_AX_DENOMINATOR[AX4] = 18

MD30503 \$MA_INDEX_AX_OFFSET[AX4] = 5

MD30500 \$MA_INDEX_AX_ASSIGN_POS_TAB[AX4] = 3

MD30300 \$MA_IS_ROT_AX[AX4] = TRUE

MD30310 \$MA_ROT_IS_MODULO[AX4] = TRUE

通过上述机床数据可将轴 4 定义为一个模数回转轴兼分度轴，从 5°开始每隔 20°一个位置。
因此可以得出以下分度位置：

5°、25°、45°、65°、85°、105°、125°、145°、165°、185°、205°、225°、245°、265°、
285°、305°、325°和 245°。

说明

输入：

MD30502 \$MA_INDEX_AX_DENOMINATOR[AX4] = 18

会得出一个 20°的间隔，因为机床数据 MD30330 \$MA_MODULO_RANGE
的缺省值是 360°。

回转轴

MD30501 \$MA_INDEX_AX_NUMERATOR[AX4] = 360

MD30502 \$MA_INDEX_AX_DENOMINATOR[AX4] = 18

MD30503 \$MA_INDEX_AX_OFFSET[AX4] = 100

MD30500 \$MA_INDEX_AX_ASSIGN_POS_TAB[AX4] = 3

MD30300 \$MA_IS_ROT_AX[AX4] = TRUE

MD36100 \$MA_POS_LIMIT_MINUS[AX1] = 100

MD36110 \$MA_POS_LIMIT_PLUS[AX1] = 260

通过上述机床数据可将轴 4 定义为一个回转轴兼分度轴，从 100°开始每隔 20°一个位置。
由此便得到以下位置：

100°、120°、140°，后续位置以此类推。

100°以内的位置不是分度位置。

建议将该位置设为软件限位开关（下限）。分度位置从该位置开始，在软件限位开关（上限）处结束，本例为 260°。因此，回转轴只能在 100°和 260°之间运行。

线性轴

MD30501 \$MA_INDEX_AX_NUMERATOR[AX1] = 10

MD30502 \$MA_INDEX_AX_DENOMINATOR[AX1] = 1

MD30503 \$MA_INDEX_AX_OFFSET[AX1] = -200

MD30500 \$MA_INDEX_AX_ASSIGN_POS_TAB[AX1] = 3

MD30300 \$MA_IS_ROT_AX[AX1] = FALSE

MD36100 \$MA_POS_LIMIT_MINUS[AX1] = -200

MD36110 \$MA_POS_LIMIT_PLUS[AX1] = 200

通过上述机床数据可将轴 4 定义为一个线性轴兼分度轴，从 -200 mm 开始每隔 10 mm 一个位置。

因此便得到以下位置：

-200 mm、-190 mm、-180 mm，后续位置以此类推。

分度位置在到达软件限位开关（上限）后结束，本例为 200 mm。

端面齿

MD30502 \$MA_INDEX_AX_DENOMINATOR[AX4] = 360

MD30503 \$MA_INDEX_AX_OFFSET[AX4] = 0

MD30500 \$MA_INDEX_AX_ASSIGN_POS_TAB[AX4] = 3

MD30300 \$MA_IS_ROT_AX[AX4] = TRUE

MD30310 \$MA_ROT_IS_MODULO[AX5] = TRUE

MD30505 \$MA_HIRTH_IS_ACTIVE[AX4] = TRUE

通过上述机床数据可将轴 4 定义为带有端面齿特性的模数回转轴兼分度轴，从 0°开始每隔 1°一个位置。

15.9 数据表

15.9.1 机床数据

15.9.1.1 通用机床数据

编号	名称: \$MN_	描述
10260	CONVERT_SCALING_SYSTEM	基本单位制切换有效
10270	POS_TAB_SCALING_SYSTEM	位置表的单位制
10900	INDEX_AX_LENGTH_POS_TAB_1	分度位置表 1 的位置数量
10910	INDEX_AX_POS_TAB_1[n]	分度位置表 1
10920	INDEX_AX_LENGTH_POS_TAB_2	分度位置表 2 的位置数量
10930	INDEX_AX_POS_TAB_2[n]	分度位置表 2
10940	INDEX_AX_MODE	分度位置选项

15.9.1.2 进给轴/主轴专用机床数据

编号	名称: \$MA_	描述
30300	IS_ROT_AX	回转轴
30310	ROT_IS_MODULO	回转轴是模数轴
30320	DISPLAY_IS_MODULO	360°模数位置显示
30500	INDEX_AX_ASSIGN_POS_TAB	轴是分度轴
30501	INDEX_AX_NUMERATOR	等距分度分子
30502	INDEX_AX_DENOMINATOR	等距分度分母
30503	INDEX_AX_OFFSET	等距分度中的第一个位置
30505	HIRTH_IS_ACTIVE	激活端面齿

15.9.2 设定数据

15.9.2.1 通用设定数据

编号	名称: \$SN_	描述
41050	JOG_CONT_MODE_LEVELTRIGGRD	连续手动运行采取长按模式

15.9.3 信号

15.9.3.1 来自进给轴/主轴的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
已回参考点/已同步 1/ 2	DB31, ...DBX60.4/5	DB390x.DBX0.4/5
分度轴就位	DB31, ...DBX76.6	DB390x.DBX1002.6

15.9.4 系统变量

名称	描述
\$AA_ACT_INDEX_AX_POS_NO[轴]	最后达到或越过的分度位置号
\$AA_PROG_INDEX_AX_POS_NO[轴]	编程的分度位置号

W3:换刀

16.1 简要说明

换刀

为进行工件的完整加工，数控机床通常配有刀库和自动换刀功能。

过程

换刀过程分为三步：

1. 将刀架从加工位置移动到换刀位置
2. 换刀
3. 将刀架从换刀位置移动到新的加工位置。

要求

换刀要满足以下几点要求：

- 换刀时间短
- 快速找刀，在加工期间就要准备好取刀和放刀
- 换刀循环编程简单
- 轴和机械手所需的运动自动完成
- 故障顺利清除

16.2 刀库和换刀装置

刀库和换刀装置是由机床类型决定的:

机床类型	刀库	换刀装置
车床	刀塔 (盘式/方式/倾斜式)	没有单独的换刀装置。 通过旋转刀塔实现换刀。
铣床	刀库 (链式/盘式/圆形/盒式)	有换刀装置：单机械手/双机械手。

因为换刀会中断加工，因此要尽可能地缩短换刀时间。

16.3 换刀时间

换刀时间主要取决于机床的机械结构。

典型的换刀时间	
0.1 至 0.2 s	刀塔旋转时间
0.3 至 2 s	机械手抓刀插刀的时间

16.4 换刀导致的加工中断时间

换刀导致的加工中断时间指为执行换刀，正在旋转的主轴离开工件到夹紧新刀具返回工件的时间。

换刀导致的典型加工中断时间有：

换刀导致的典型加工中断时间	
0.3 至 1 s	带刀塔的车床
0.5 至 5 s	带有机械手的铣床

16.5 换刀过程的控制

方式

可通过以下方式控制换刀过程：

- T 功能
- M 指令（最适宜采用 M06）

参数设置

生效的控制方式可由以下机床数据设置：

MD22550 \$MC_TOOL_CHANGE_MODE

值	含义	典型应用
0	T 功能立即装入新刀。	带有刀塔的车床
1	T 功能用于刀库选刀，在加工的同时将新刀具转入换刀位。 通过 M 指令进行换刀时，主轴松开旧刀，然后装入新刀。	带刀库的铣床

用于换刀的 M 指令可在以下机床数据中进行设置：

MD22560 \$MC_TOOL_CHANGE_M_CODE

缺省设置是 6(根据标准 DIN 66025)。

说明

当刀具补偿号由 PLC 或 HMI 刀具管理功能指定时，必须在合适的位置添加一个预处理停止指令 STOPRE。但是在刀具半径补偿(G41 / G42)激活时或样条插补中，由于事先需要占用多个程序段来计算轨迹，所以此时应避免使用预处理停止指令 STOPRE。

文档

更多关于 M 功能（包含换刀指令 M06）的信息，例如：扩展地址、到 PLC 的输出时间、辅助功能组、程序段搜索特性、溢出转存特性的信息请参见：

功能手册之同步动作分册

16.6 换刀点

换刀点

换刀点的选择对换刀导致的加工中断时间 (页 838)有着巨大的影响。换刀点要依据机床结构和当前的加工任务而定。

运行至固定点

“运行至固定点”功能可以使轴运行到机床数据中定义的固定点。因此该功能可用于定义和控制一个或多个换刀点。

运行至固定点有两种方式可选:

- 手动运行至固定点
机床操作人员可在 JOG 模式下通过运行键或手轮使轴 “运行至固定点”(参见章节 “手动运行至固定点 (页 197)”)。
- 使用 G75 运行至固定点
也可通过指令 G75 从零件程序中调用该功能。

文档:
编程手册之基本原理分册; 章节 “补充指令 > 运行至固定点 (G75)”

16.7 边界条件

换刀的前提是: 借助合理设计的刀具管理功能新刀具可提前位于换刀点上。

16.8 示例

铣床

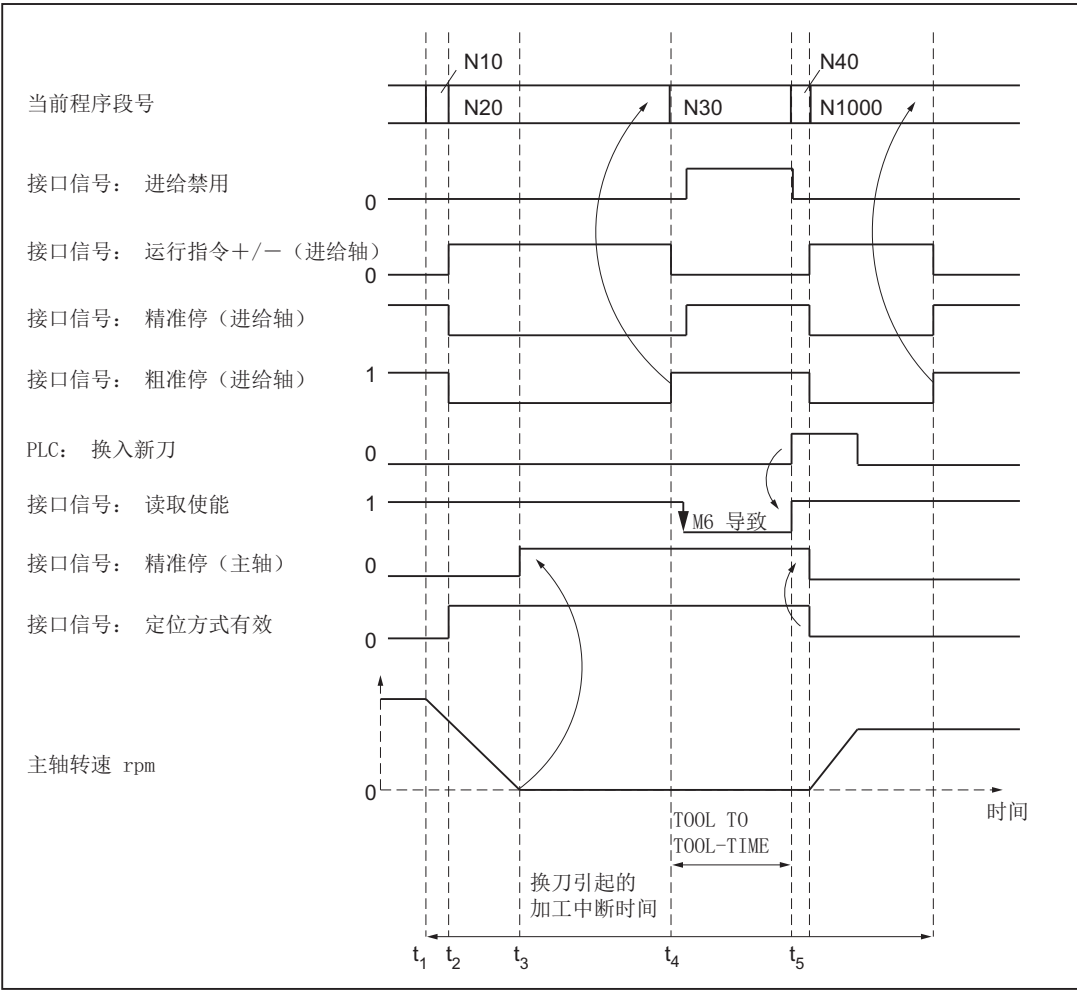
下面以一台带一个机械手、有固定绝对换刀点的铣床为例, 展示典型的换刀动作。
加工程序:

程序代码	注释
N970 G0 X=... Y=... Z=... LF	; 主轴离开工件

程序代码	注释
N980 T1 LF	; 刀具预选
N990 W_WECHSEL LF	; 子程序调用, 无参数
N1000 G90 G0 X=... Y=... Z=... M3 S1000 LF	; 继续加工

换刀子程序:

程序代码	注释
PROC W_WECHSEL LF	
N10 SPOSA=... S0 LF	; 主轴定位
N20 G75 FP=2 X1=0 Y1=0 Z1=0	; 逼近换刀点
N30 M06 LF	; 更换刀具
N40 M17 LF	



t_1 : 进给轴停止。

主轴旋转。

在 N10 中启动换刀循环

t_2 : 在 N20 中，指令 G75 使轴逼近换刀点。

t_3 : 轴从程序段 N10 起到达编程位置

t_4 : 轴从 N20 起进入粗准停窗口；然后开始 N30:

M06 指令使主轴松开旧刀，装入并夹紧新刀。

t_5 : 机械手回到初始位置。

图 16-1 换刀时序

因此可在主程序的 N1000 中:

- 选择新的刀具补偿。
- 使轴再次返回轮廓。
- 使主轴加速。

16.9 数据表

16.9.1 机床数据

16.9.1.1 通用机床数据

编号	名称: \$MN_	描述
18082	MM_NUM_TOOL	刀具数

16.9.1.2 通道专用机床数据

编号	名称: \$MC_	描述
22200	AUXFU_M_SYNC_TYPE	M 功能的输出时间
22220	AUXFU_T_SYNC_TYPE	T 功能的输出时间
22550	TOOL_CHANGE_MODE	使用 M 功能时的新刀具补偿
22560	TOOL_CHANGE_M_CODE	用于换刀的 M 功能

16.9.1.3 进给轴/主轴专用机床数据

编号	名称: \$MA_	描述
30600	FIX_POINT_POS[n]	G75 指令中机床轴的固定位置

16.9 数据表

16.9.2 信号

16.9.2.1 来自通道的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
M 功能 M06	DB21, ...DBX194.6	DB2500.DBB1000.6

W4: 磨削专用的刀具补偿和监控

内容

本章介绍以下内容：

- 磨削专用的刀具补偿
- 在线刀具补偿(Continuous Dressing)
- 磨削专用的刀具监控
- 恒定砂轮圆周速度(GWPS)

文档

基本原理参见：

- 功能手册之基本功能分册，章节“刀具补偿(W1)”

编程方式、生效方式和操作方式请参见：

- 编程手册之基本原理分册

17.1 磨削专用的刀具补偿

17.1.1 刀具数据的结构

磨削刀具

磨削刀具的类型号从 400 到 499。

磨削刀具的补偿

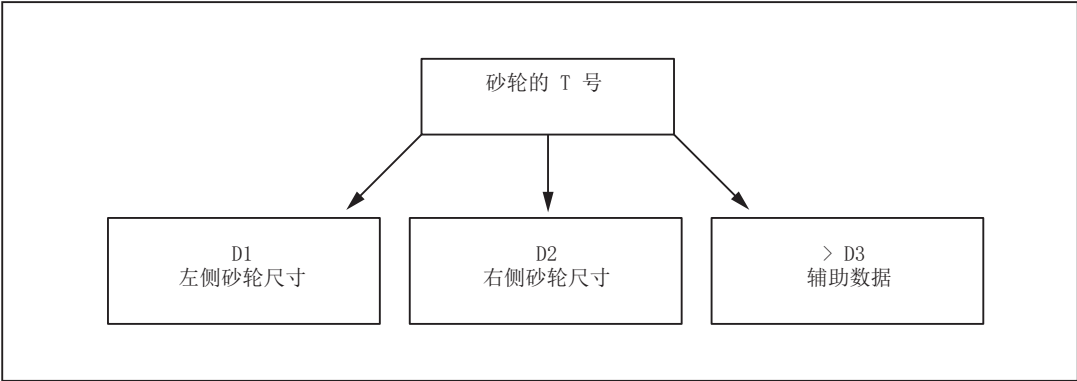
通常情况下，磨削刀具不仅包含刀沿数据，也包含刀具数据和修整器数据。

对于砂轮而言，“刀沿数据”即砂轮左侧和右侧的几何尺寸，分别保存一个 T 号下的 D1 或 D2 中。

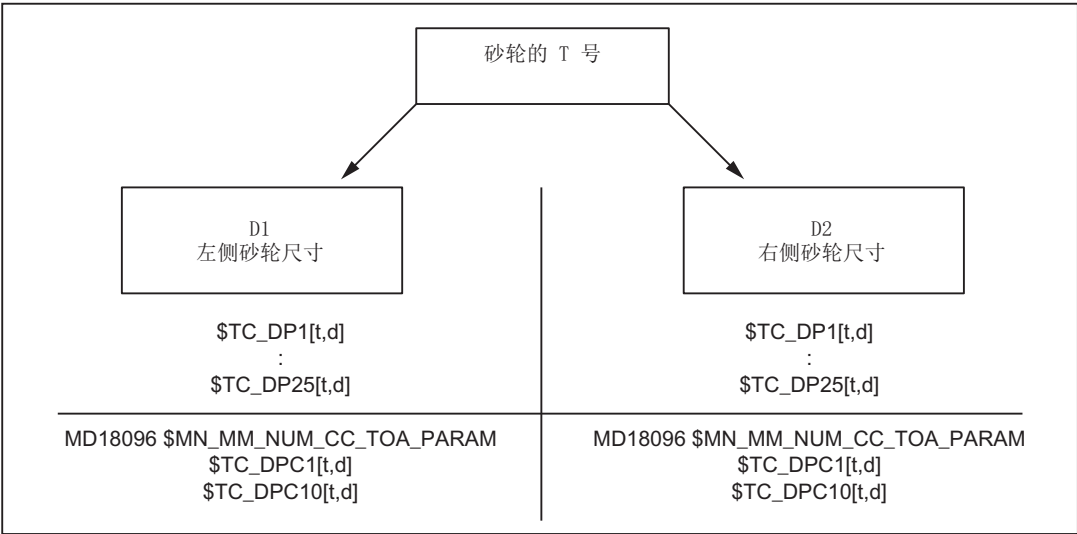
17.1 磨削专用的刀具补偿

所需的修整器几何尺寸可保存在该 T 号从 D3 起的 D 号中或附加的刀沿数据中 (MD18096 \$MN_MM_NUM_CC_TOA_PARAM)。

示例 1:



示例 2:



在一个 T 号下汇集了所有和砂轮、修整器相关的补偿数据，其中的 D1、D2 指出砂轮左侧和右侧的几何尺寸，D3, D4 指出修整器几何尺寸。

- D1: 砂轮左侧几何尺寸
- D2: 砂轮右侧几何尺寸
- D3: 修整器左侧几何尺寸
- D4: 修整器右侧几何尺寸

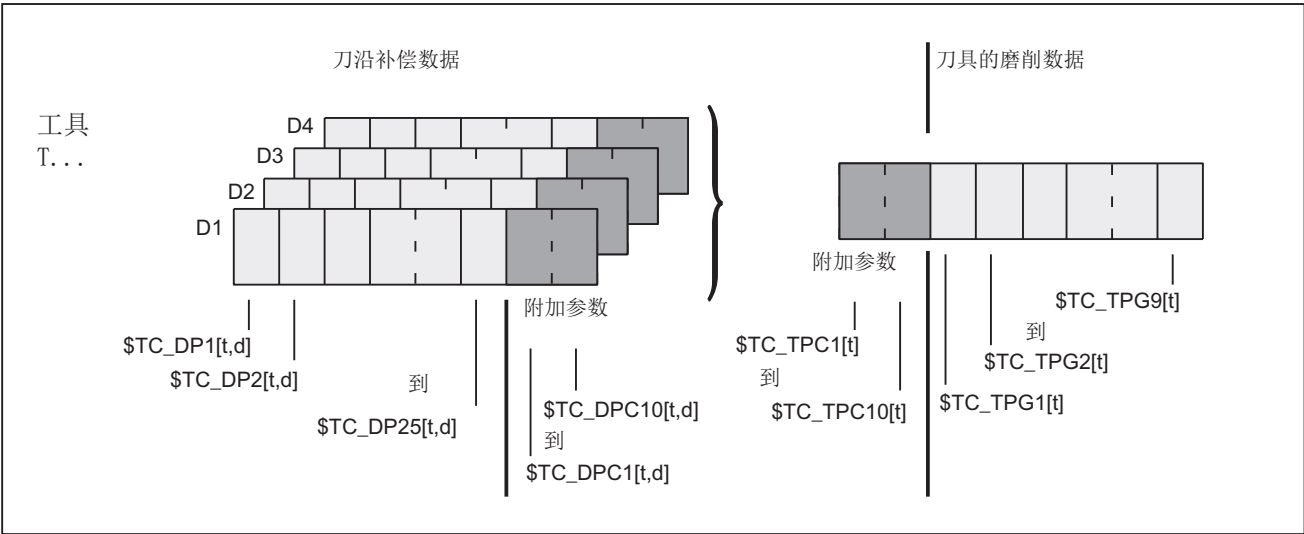


图 17-1 磨削刀具补偿数据的结构

17.1.2 刀沿补偿数据

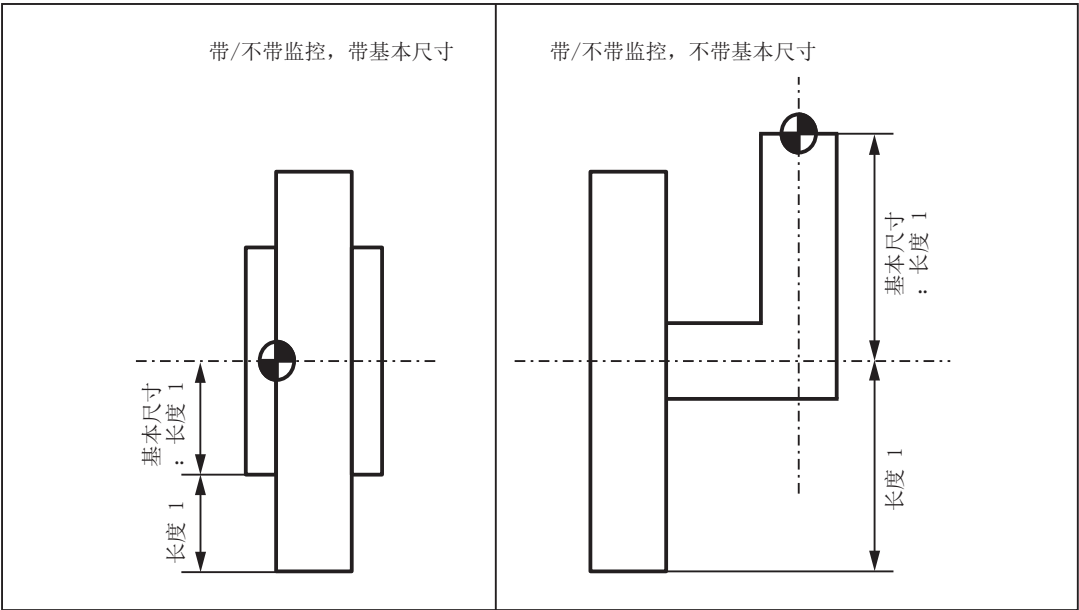
刀具参数

磨削刀具的参数和车刀、铣刀具有相同的含义。

刀具参数	含义	注释	
1	刀具类型		
2	刀沿位置	只适用于车刀和磨削刀具	
几何尺寸 - 刀具长度补偿			
3	长度 1		
4	长度 2		
5	长度 3		
几何尺寸 - 刀具半径补偿			
6	半径 1		
7			预留
8			预留
9			预留
10			预留

17.1 磨削专用的刀具补偿

刀具参数	含义	注释	
11			预留
磨损尺寸 - 刀具长度补偿			
12	长度 1		
13	长度 2		
14	长度 3		
磨损尺寸 - 刀具刀具补偿			
15	半径 1		
16			预留
17			预留
18			预留
19			预留
20			预留
基本尺寸/适配器尺寸 - 刀具长度补偿			
21	基本长度 1		
22	基本长度 2		
23	基本长度 3		
工艺			
24	后角	仅用于车刀	
25			预留



说明

选中磨削刀具的“刀沿数据” D1 和 D2 相互链接，也就是说：如果 D1 或 D2 中的一个参数被修改了，那么另一个刀沿（D1 或 D2）的同一个参数也会自动改写，参见刀具数据 \$TC_TPG2。

定义附加参数 \$TC_DPC1...10

用户自定义的刀沿数据可通过附加参数 \$TC_DPC1 到 10 由以下通用机床数据单独设置，而不受刀具类型的影响：

MD18096 \$MN_MM_NUM_CC_TOA_PARAM

 小心

数据丢失

上电后，机床数据 MD18096 的修改生效，存储器被格式化。
事先须进行数据备份。

17.1 磨削专用的刀具补偿

 小心

无砂轮切换

进行轮廓磨削时，砂轮左/右侧补偿**不**自动切换。
必须手动编程切换。

磨削刀具类型

磨削刀具类型号为以下结构：

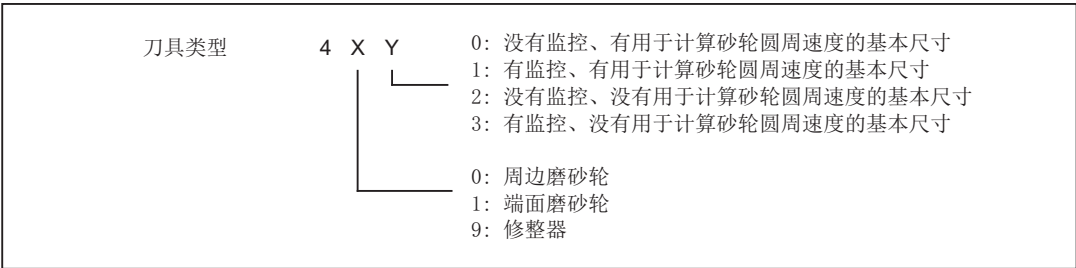


图 17-2 磨削刀具类型号的结构

说明

MD20350 \$MC_TOOL_GRIND_AUTO_TMON

通过该通道专用的机床数据可以确定：在选择刀具时是否就激活设有监控的磨削刀具(即：类型为奇数的磨削刀具)的**监控功能**。

依据上图会产生以下磨削刀具类型号：

类型	描述
400	周边磨砂轮
401	周边磨砂轮、有监控、有用于计算砂轮圆周速度的基本尺寸
402	周边磨砂轮、没有监控、没有用于计算砂轮圆周速度的基本尺寸
403	周边磨砂轮、有监控、没有用于计算砂轮圆周速度的基本尺寸
410	端面磨砂轮
411	端面磨砂轮、有监控、有用于计算砂轮圆周速度的基本尺寸

类型	描述
412	端面磨砂轮、没有监控、没有用于计算砂轮圆周速度的基本尺寸
413	端面磨砂轮、有监控、没有用于计算砂轮圆周速度的基本尺寸
490	修整器

17.1.3 刀具的磨削数据

刀具的磨削数据

刀具的磨削数据只能供每个 T 号(400 - 499)使用一次。每创建一个新的磨削刀具(400 - 499)就会自动创建一个刀具磨削数据。

说明

刀具磨削数据与刀沿相类似。

在设定刀沿数量时需要考虑到该特性:

MD18100 \$MN_MM_NUM_CUTTING_EDGES_IN_TOA

删除刀具的所有刀沿后，现有的刀具磨削数据也会自动删除。

参数

参数如下设置:

参数	含义	数据类型
\$TC_TPG1	主轴号	整数
\$TC_TPG2	链接规则	整数
\$TC_TPG3	最小的砂轮半径	实数
\$TC_TPG4	最小的砂轮宽度	实数
\$TC_TPG5	实际的砂轮宽度	实数
\$TC_TPG6	最大转速	实数
\$TC_TPG7	最大圆周速度	实数
\$TC_TPG8	砂轮倾斜角度	实数
\$TC_TPG9	用于半径计算的参数号	整数

17.1 磨削专用的刀具补偿

参数	含义	数据类型
附加参数(用户自定义刀沿数据)		
\$TC_TPC1 到 \$TC_TPC10		实数

定义附加参数 \$TC_DPC1...10

用户自定义的刀沿数据可通过附加参数 \$TC_DPC1 到 10 单独设置，而不受刀具类型的影响。该设置由以下通用机床数据完成：

MD18096 \$MN_MM_NUM_CC_TDA_PARAM

 小心
数据丢失 上电后，机床数据 MD18096 的修改生效，存储器被格式化。 事先须进行数据备份。

主轴号 \$TC_TPG1

指定需要监控的几何尺寸(例如:砂轮半径与砂轮宽度)，指定主轴转速(例如:砂轮圆周速度)。

链接规则 \$TC_TPG2

指定刀沿 2(D2)和刀沿 1(D1)的哪些刀具参数应相互链接。构成链接后，一旦修改其中某个参数的数值，则另一个相连刀沿的参数也会自动修改。

刀具参数	含义	\$TC_TPG2 中的位	十六进制	十进制
\$TC_DP1	刀具类型	0	0001	1
\$TC_DP2	刀沿位置	1	0002	2
几何尺寸 - 刀具长度补偿				
\$TC_DP3	长度 1	2	0004	8
\$TC_DP4	长度 2	3	0008	16
\$TC_DP5	长度 3	4	0010	32
\$TC_DP6	半径	5	0020	64
\$TC_DP7	预留	6	0040	128

刀具参数	含义	\$TC_TPG2 中的位	十六进制	十进制
\$TC_DP8		7	0080	256
\$TC_DP9		8	0100	512
\$TC_DP10		9	0200	1024
\$TC_DP11	预留	10	0400	2048
磨损尺寸 - 刀具长度补偿				
\$TC_DP12	长度 1	11	0800	4096
\$TC_DP13	长度 2	12	1000	8192
\$TC_DP14	长度 3	13	2000	16384
\$TC_DP15	半径	14	4000	32768
\$TC_DP16	预留	15	8000	65536
\$TC_DP17		16	10000	131072
\$TC_DP18		17	20000	262144
\$TC_DP19		18	40000	524288
\$TC_DP20	预留	19	80000	1048576
基本尺寸/适配器尺寸 - 刀具长度补偿				
\$TC_DP21	基本长度 1	20	100000	2097152
\$TC_DP22	基本长度 2	21	200000	4194304
\$TC_DP23	基本长度 3	22	400000	8388608
工艺				
\$TC_DP24	预留	23	800000	16777216
\$TC_DP25	预留	24	1000000	33554432

链接示例:

磨削刀具(以 T1 为例)的以下数据要链接在一起: 几何尺寸: 长度 1、2 和 3、磨损尺寸: 长度 1、2 和 3 和基本尺寸/适配器尺寸。

此外, 刀沿 1 和 2 要具有相同的刀具类型号。

刀具类型	\$TC_DP1	位 0
长度 1	\$TC_DP3	位 2
长度 2	\$TC_DP4	位 3
长度 3	\$TC_DP5	位 4

17.1 磨削专用的刀具补偿

磨损		
长度 1	\$TC_DP12	位 11
长度 2	\$TC_DP13	位 12
长度 3	\$TC_DP14	位 13
基本尺寸/适配器尺寸		
长度 1	\$TC_DP21	位 20
长度 2	\$TC_DP22	位 21
长度 3	\$TC_DP23	位 22

因此，参数 \$TC_TPG2 必须按如下方式赋值：

二进制：	\$TC_TPG2[1]= 'B111 0000 0011 1000 0001 1101' (位 22 ... 位 0)
十六进制：	\$TC_TPG2[1]= 'H70381D'
十进制：	\$TC_TPG2[1]='D7354397'

说明
如果事后修改了链接规则，两个刀沿的参数不会立即自动匹配，而是要在其中一个刀沿的参数发生改变后才自动匹配。

最小砂轮半径和砂轮宽度 \$TC_TPG3 \$TC_TPG4

在这两个参数中输入磨削砂轮半径限制和宽度限制。 这些参数值可用于监控磨削砂轮的几何尺寸。

说明
使用斜砂轮时应注意，此处要输入笛卡尔坐标系中的最小砂轮半径。 如果砂轮宽度或砂轮半径低于限值，PLC 接口会输出相应信号。 用户可使用这些信号来设计故障报告方案。

当前砂轮宽度 \$TC_TPG5

经过修整后的砂轮宽度会进入该参数。

最大转速和最大圆周速度 \$TC_TPG6 \$TC_TPG7

在这两个参数中输入砂轮的最大转速和最大圆周速度。

前提条件:已定义了一根主轴。

砂轮倾斜角度 \$TC_TPG8

该参数指定了砂轮在当前平面上的倾斜度。该角度用于砂轮圆周速度。

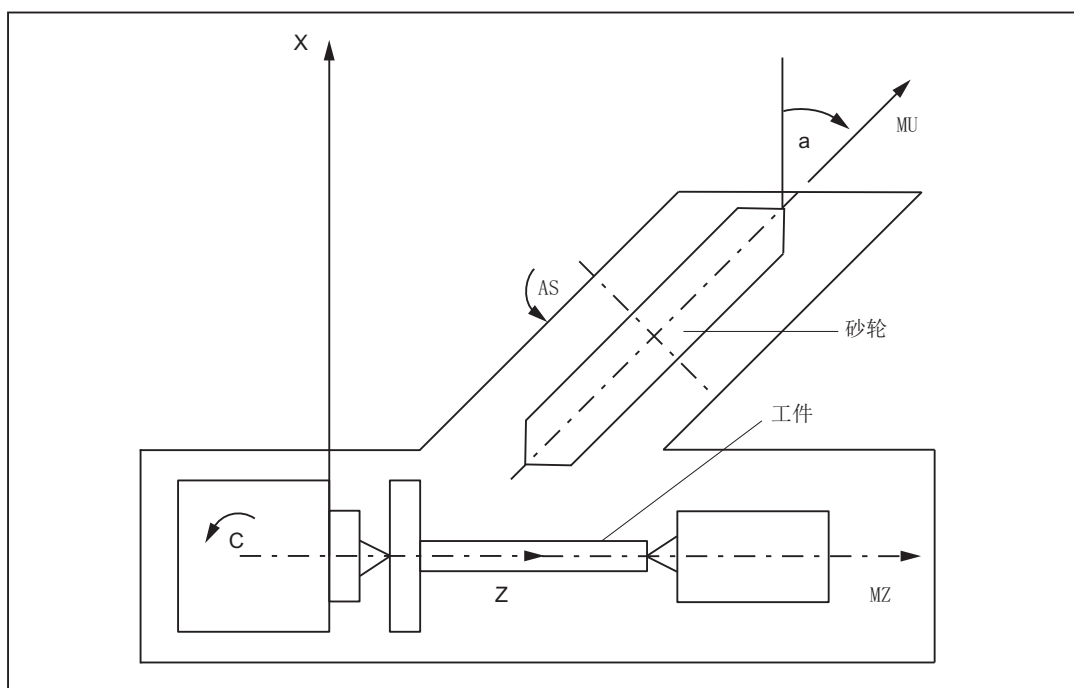


图 17-3 进给轴倾斜一定角度的机床

说明

在角度变化时不自动补偿砂轮长度。

设置规定: $-90^\circ \leq \$TC\_TPG8 < +90^\circ$ 。

使用斜轴机床时,斜轴和斜砂轮必须为相同的角度。

参与半径计算的参数号 \$TC_TPG9

该参数可以确定应使用哪些补偿值来计算砂轮圆周速度和监控最小砂轮半径(\$TC_TPG3)。

\$TC_TPG9 = 3	长度 1(几何尺寸 + 磨损尺寸 + 基本尺寸，取决于刀具类型)
\$TC_TPG9 = 4	长度 2(几何尺寸 + 磨损尺寸 + 基本尺寸，取决于刀具类型)
\$TC_TPG9 = 5	长度 3(几何尺寸 + 磨损尺寸 + 基本尺寸，取决于刀具类型)
\$TC_TPG9 = 6	半径

从零件程序中访问

可从零件程序中读写参数。

示例	编程
读取刀具 2 的当前宽度并保存到 R10 中	R10 = \$TC_TPG5 [2]
将刀具 3 的最大转速值写为 2000	\$TC_TPG6 [3] = 2000

当前刀具的 \$P_ATPG[m]

通过该系统变量可以访问**当前刀具**的磨削数据。

m: 参数号（数据类型：实数）

示例：

参数 3 (\$TPG3[<T 号>])

\$P_ATPG[3]=R10

说明

监控数据同时作用于砂轮左侧和右侧。

在编程 GWPSON (选择恒定的砂轮圆周速度)和 TMON (选择刀具监控)时，刀具专用的磨削数据生效。需要激活数据修改时，必须重新编程 GWPSON 或 TMON。

无论在何种情况下，长度补偿都是指笛卡尔坐标系中刀架基准点和刀尖之间的距离，在倾斜砂轮上也要注意这一点。

17.1.4 磨削刀具示例

长度补偿设置

几何轴的长度补偿和半径补偿设置只针对当前的平面生效。

平面

系统中包含以下平面和轴(横轴、纵轴和竖轴分别对应几何轴 1、2、3):

指令	平面 (横轴/纵轴)	垂直于平面的轴 (竖轴)
G17	X/Y	Z
G18	Z/X	Y
G19	Y/Z	X

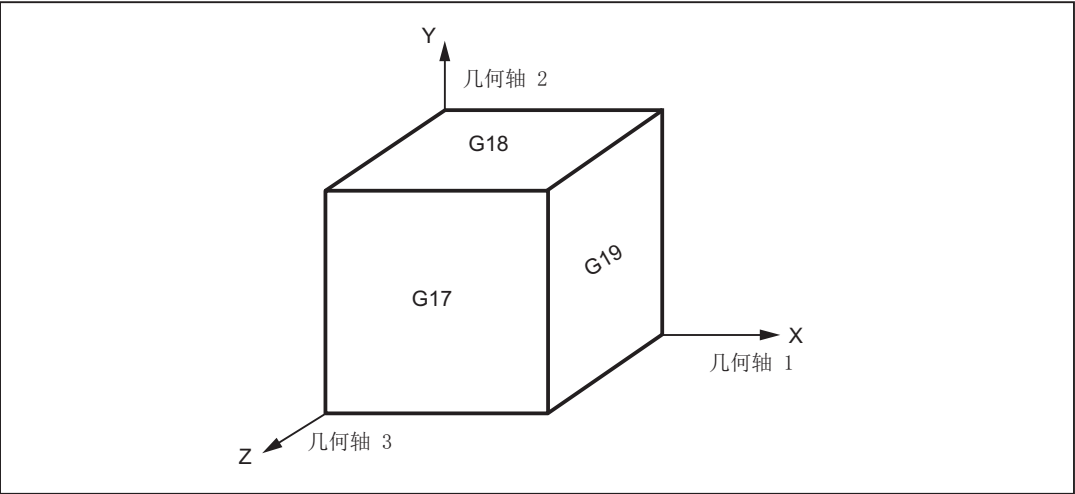


图 17-4 平面和轴

17.1 磨削专用的刀具补偿

周边磨砂轮

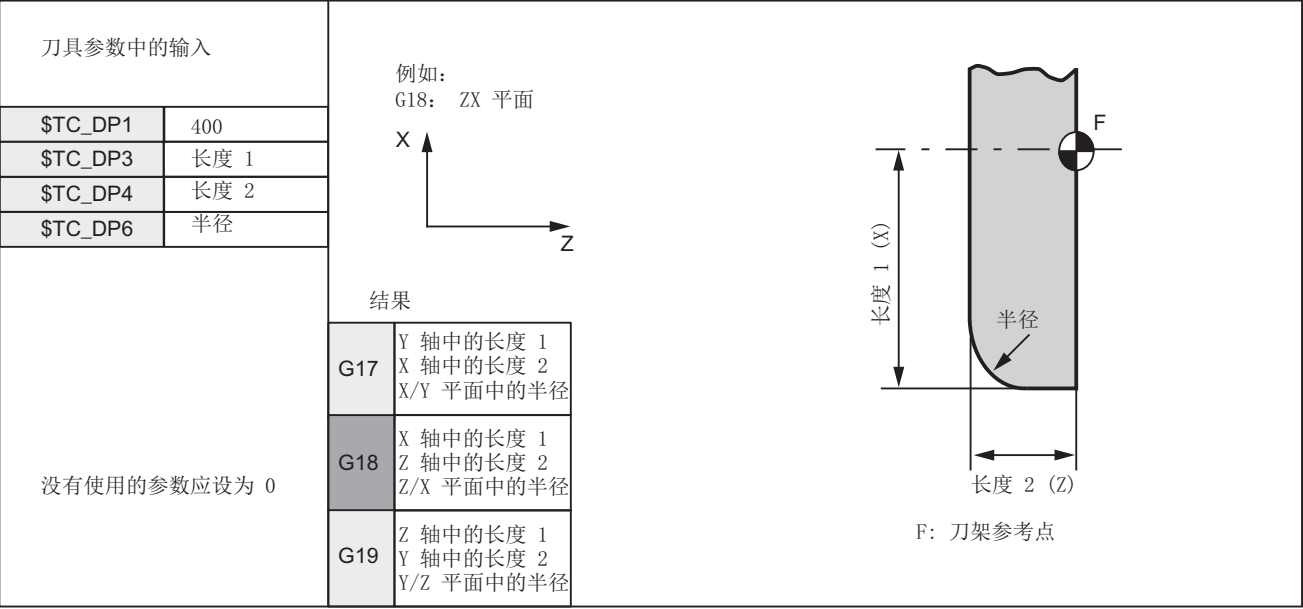


图 17-5 周边磨砂轮需要的补偿值

斜砂轮

无基本尺寸用于计算砂轮圆周速度 GWPS

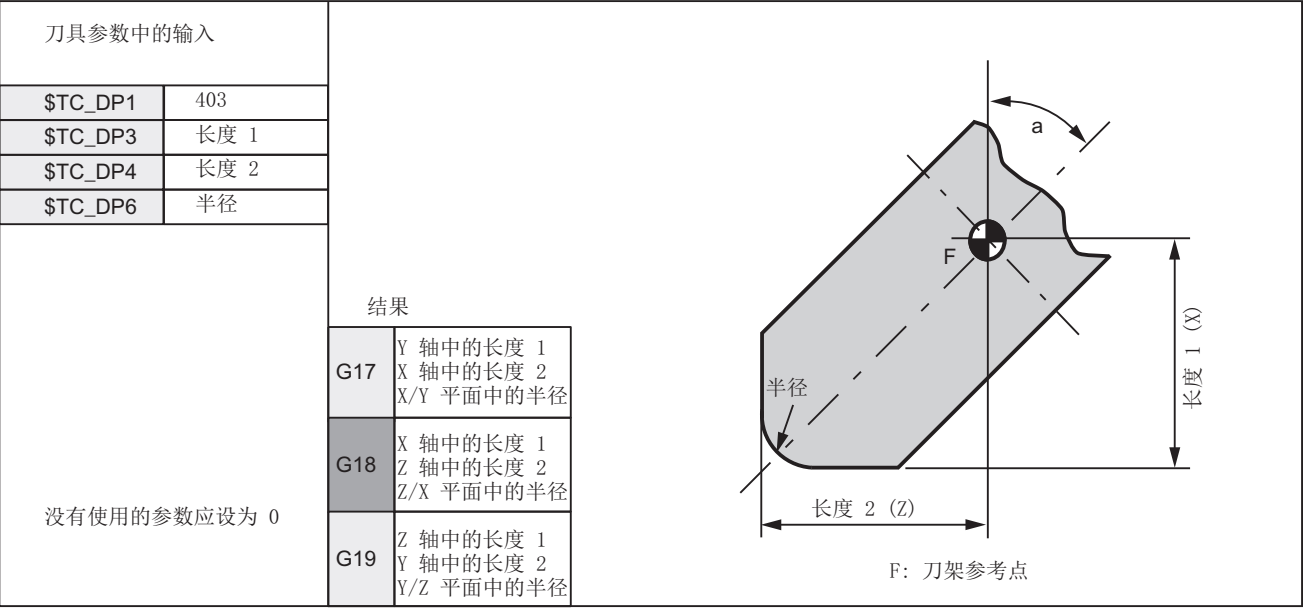


图 17-6 斜砂轮要求的补偿值，带监控自动选择

斜砂轮

带基本尺寸用于计算砂轮圆周速度 GWPS

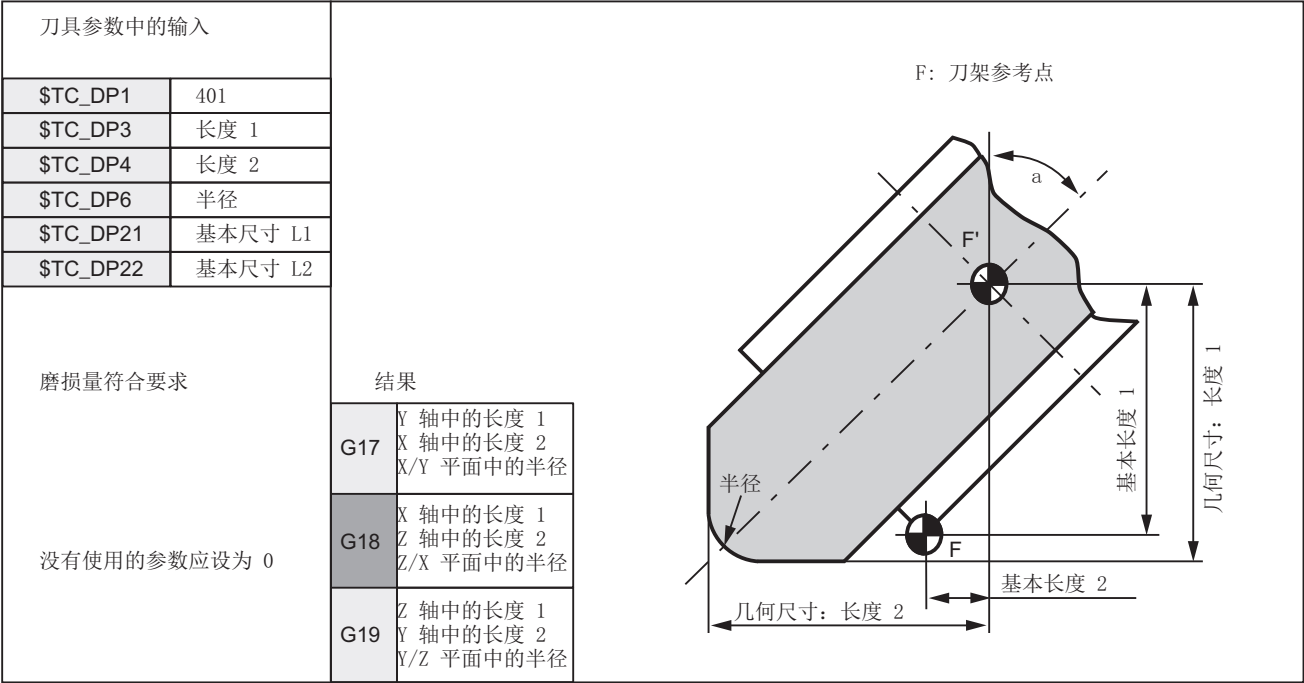


图 17-7 斜砂轮要求的补偿值，带监控自动选择，带基本尺寸用于计算砂轮圆周速度 GWPS

17.1 磨削专用的刀具补偿

周边磨砂轮

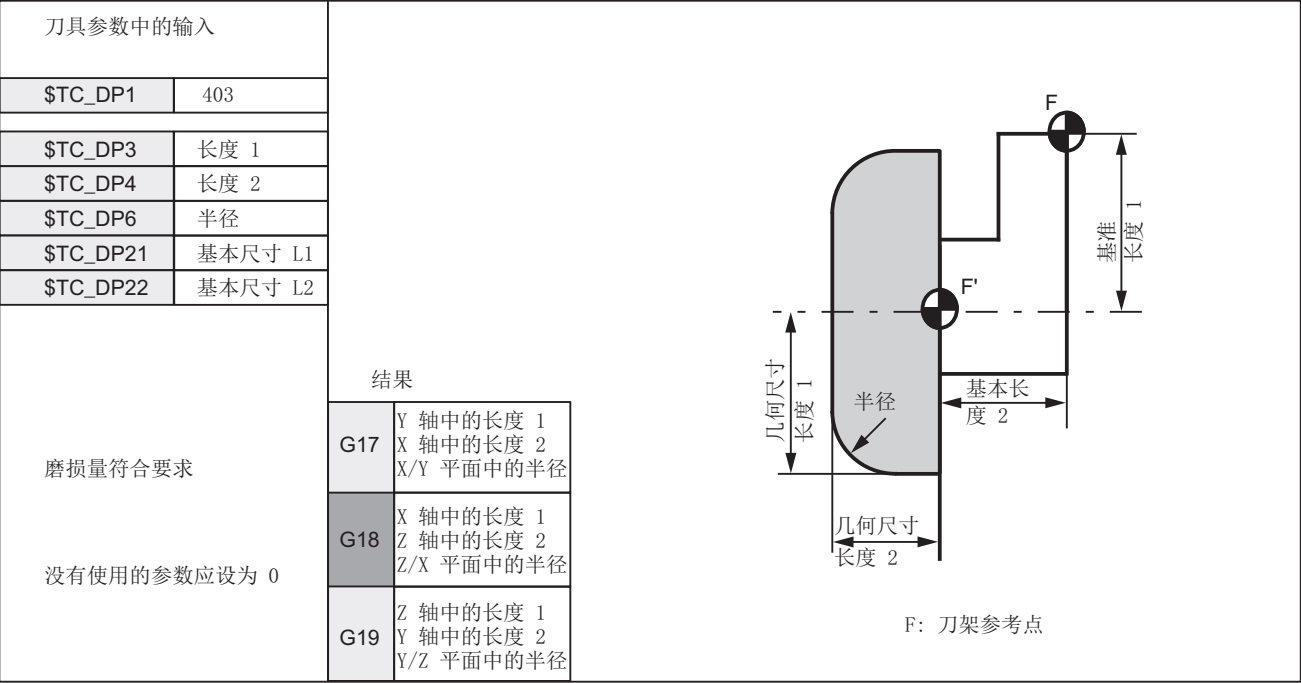


图 17-8 周边磨砂轮需要的补偿值，无基本尺寸用于计算砂轮圆周速度 GWPS

端面磨砂轮

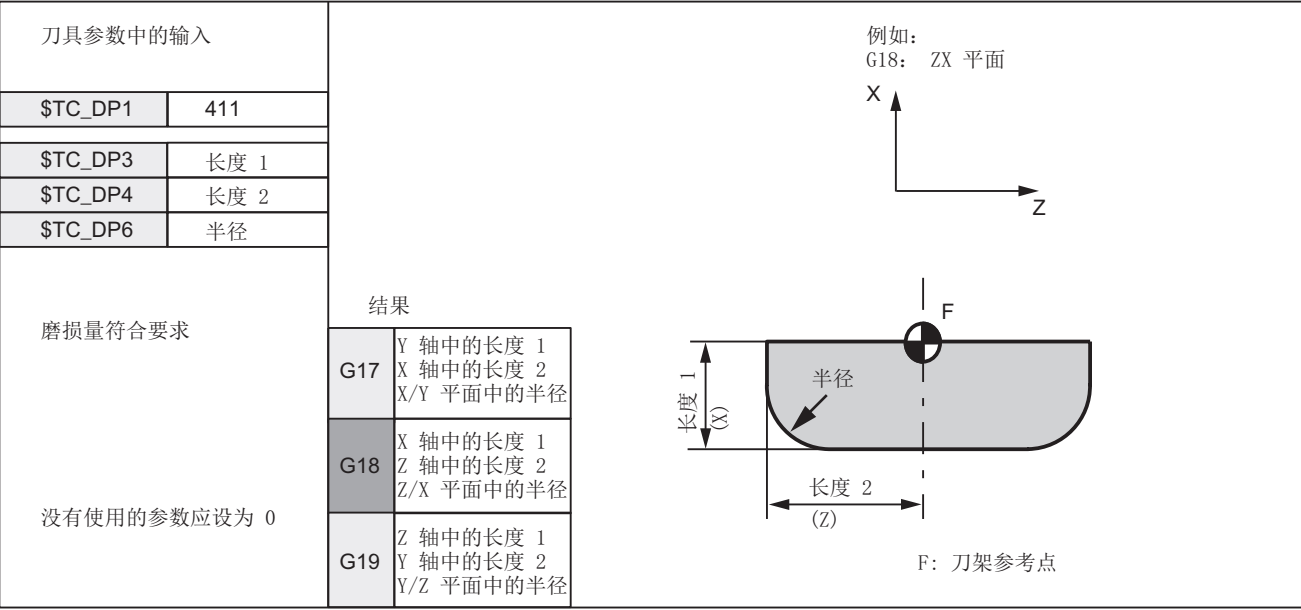


图 17-9 端面磨砂轮需要的补偿值，带有监控参数

17.2 在线刀具补偿

17.2.1 简介

应用

磨削工艺既包含了用砂轮磨削一个工件的过程，也包含了用修整器修整砂轮的过程。这两个过程可以在同一个通道中进行，也可以在不同的通道中进行。

要想在砂轮磨削工件的同时对砂轮进行修整，机床就必须具备补偿功能，来补偿因修整而被去除的砂轮尺寸。该补偿可通过“在线刀具补偿”(Continuous Dressing)功能实现。

在磨削期间同步修整砂轮

为了在磨削期间同步修整砂轮，因修整而被去掉的砂轮尺寸必须在加工通道中作为“刀具补偿量”传送给当前砂轮，并且该补偿必须立即生效。

该补偿可通过“Continuous Dressing、并行修整、在线刀具补偿”功能实现。

说明

在线刀具补偿只能应用于磨削刀具。

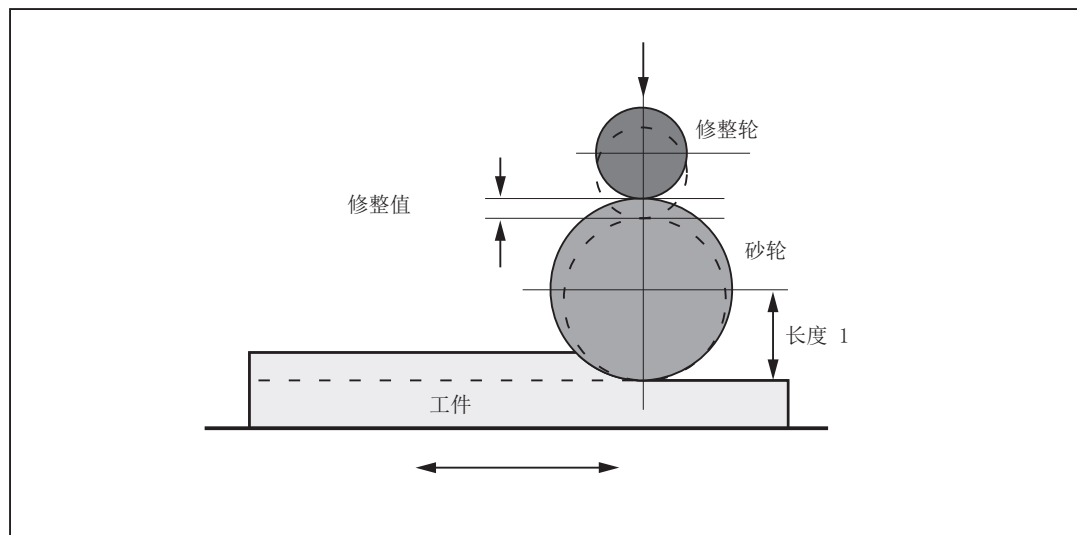


图 17-10 加工期间使用修整轮进行修整

17.2 在线刀具补偿

简介

可为每个通道中的每个磨削刀具激活在线刀具补偿功能。

在线刀具补偿量通常附加在长度磨损尺寸上。与几何尺寸和磨损尺寸类似，长度对应的几何轴是由当前平面、刀具类型决定的。

在线刀具补偿生效时磨削主轴的监控功能也同样生效。

说明

补偿对象始终是所选长度的磨损尺寸参数。如果多个刀沿的长度补偿都相同，就必须设置对应的链接规则来确保刀沿 2 的参数值也自动调整。

如果加工通道中的在线补偿已激活，就不得从加工程序或者通过操作来修改该通道中当前刀具的磨损尺寸。

只有在重新选择刀具后系统才考虑新的半径磨损尺寸(P15)。

在线补偿也会影响恒定砂轮圆周速度(GWPS)，即：主轴转速会按对应值调整。

指令

在线刀具补偿有以下指令：

指令	含义
FCTDEF <多项式编号>, <下限值>, <上限值>, <系数 0>, <系数 1>, <系数 2>, <系数 3>)	设置函数 (最多为 3 次多项式) (Fine Tool Offset Definition)
PUTFTOCF (<多项式编号>, <参考值>, <长度 1_2_3>, <通道号>, <主轴号>)	写入在线刀具补偿 连续 (Put Fine Tool Offset Compensation)
PUTFTOC (<值>, <长度 1_2_3>, <通道号>, <主轴号>)	写入在线刀具补偿 不连续 (Put Fine Tool Offset Compensation)
FTOCON	激活在线刀具补偿 (Fine Tool Offset Compensation On)
FTOCOF	撤销在线刀具补偿 (Fine Tool Offset Compensation Off)

说明
只有在重新编程 T 号或 D 号后，TOA 存储器中补偿值的修改才会生效。
文档：
编程手册之工作准备分册

17.2.2 连续写入在线刀具补偿

FCTDEF

在一些修整方案中，比如：使用修整轮时，砂轮半径会随着修整轮的进给连续线性减小。为此需要使用由修整轮进给量和对应长度方向上的磨损量构成的线性函数。

FCTDEF 可用于定义 3 个独立的函数，句法如下。

设置函数

函数设置必须在一个独立的程序段中完成，其中包括以下句法：

FCTDEF<多项式编号>,<下限值>,<上限值>,<系数 a0>,<系数 a1>,<系数 a2>,<系数 a3>)

FCTDEF	函数定义
多项式编号:	函数号(例如:1、2 或 3)
下限/上限值:	确定函数的值域; (输入精度的限值)
系数 a ₀ , a ₁ , a ₂ :	多项式系数

一个 3 次多项式通常为：

$$y = a_0 + a_1 * x + a_2 * x^2 + a_3 * x^3$$

17.2 在线刀具补偿

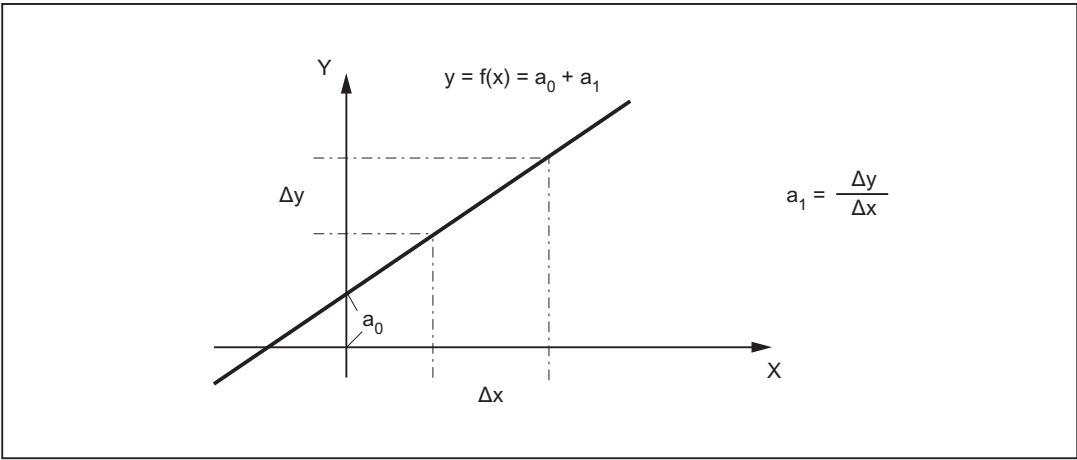


图 17-11 直线

说明

FCTDEF 必须单独在一个 NC 程序段中编程。

举例:

已知: 斜率: $a_1 = +1$
 $a_2 = 0$
 $a_3 = 0$

在定义前，函数值 y 应为 0，并且由机床轴 **XA** 轴（比如：修整器轴）导出。

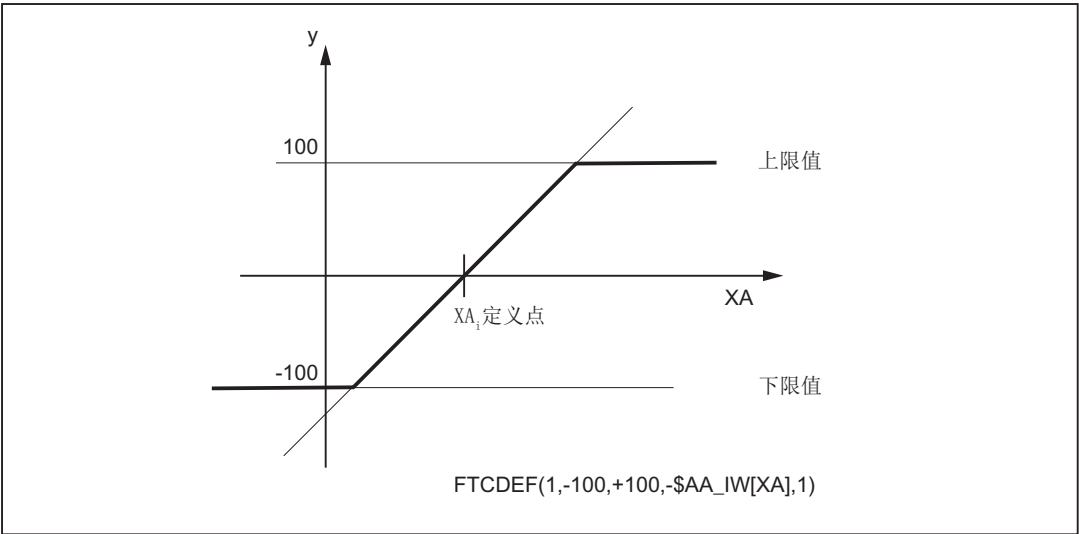


图 17-12 斜率为 1 的直线

连续写入在线刀具补偿

PUTFTOCF(<多项式编号>, <参考值>, <长度 1_2_3>, <通道号>, <主轴号>)

PUFTOCF

- 多项式编号: 函数号(1、2、3)
- 参考值: 函数参考值
- 长度 1_2_3: 计入补偿值的磨损尺寸参数
- 通道号: 补偿值生效的通道
- 主轴号: 补偿值生效的主轴

在线补偿在修整器轴的运动程序段之前激活。

举例:

程序代码	注释
FTCDEF(1,-100,100,-\$AA_IW[X],1)	; 定义函数
PUTFTOCF(1,\$AA_IW[X],1,2,1)	; 连续写入在线刀具补偿

17.2 在线刀具补偿

通道 2 中主轴 1 的刀具长度 1 是随 X 轴的运动而变化的。

说明

指定主轴号也可以为当前未生效（指几何尺寸未生效）的磨削刀具激活在线刀具补偿。
如果省略了通道号，在线补偿会在自身通道中激活。
如果省略了主轴号，在线补偿会对当前刀具生效。

在线刀具补偿也可作为同步动作调用。

文档：

功能手册之同步动作分册

17.2.3 激活/撤销在线刀具补偿

激活/撤销在线刀具补偿

以下指令可在加工通道中激活/撤销在线刀具补偿(磨削通道、目标通道):

FTOCON 激活在线刀具补偿

只有当在线刀具补偿(FTOCON)激活时，加工通道才能处理在线刀具补偿 (PUTFTOC)。 否则系统会输出报警 20204 “PUTFTOC 指令非法”。

FTOCOF 撤销在线刀具补偿

使用指令 FTOCOF 即可撤销在线刀具补偿。 已写入的值依然保留在各个长度磨损尺寸中。

在线补偿在基本坐标系中进行，也就是说，即使是工件坐标系经过旋转后，长度补偿始终平行于原先工件坐标系的各坐标轴。

不论需要补偿的轴是否在当前的程序段中，系统都会使轴走完补偿量。

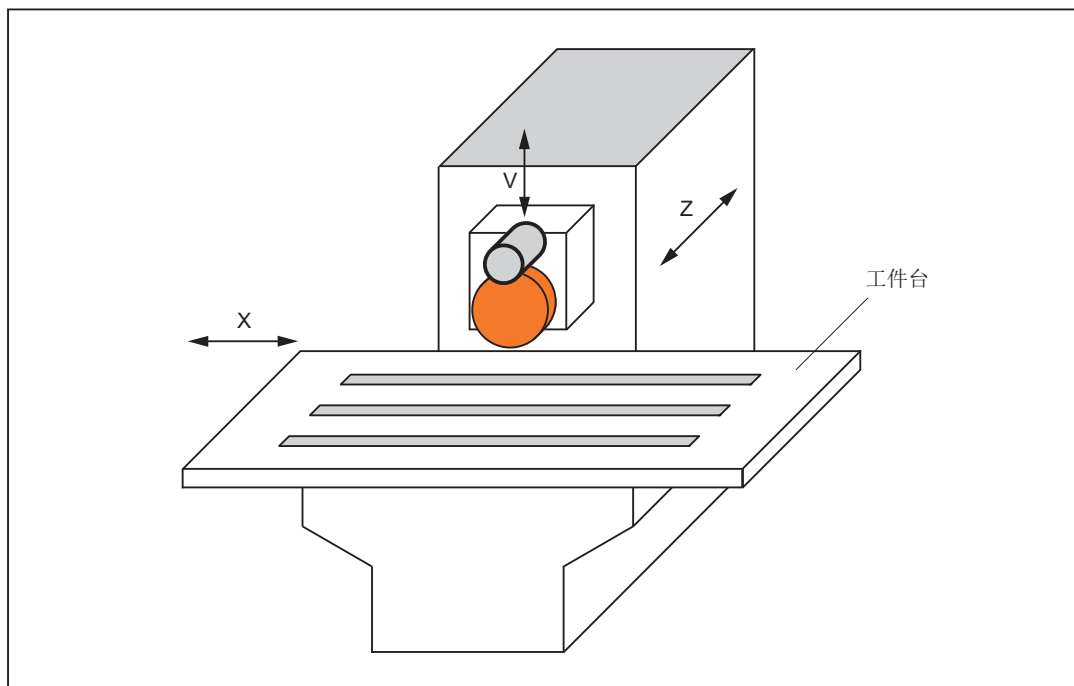
说明

指令 FTOCON 必须在补偿所在通道中写入(磨削用加工通道)。

FTOCOF 始终保持复位。只有在零件程序和 FTOCON 指令激活时，PUTFTOC 指令才会生效。

17.2.4 连续写入在线刀具补偿示例

平面磨床



Y: 砂轮进给轴

V: 修整轮进给轴

X: 左右往复轴

刀具补偿平面:G19(Y/Z 平面)

长度 1 位于 Z 轴上, 长度 2 位于 Y 轴上, 刀具类型 = 401

加工: 通道 1, 使用 Y 轴、X 轴

修整: 通道 2, 使用 V 轴

任务

在 Y100 位置处开始磨削后, 砂轮应被修掉 0.05(在 V 方向上)。该修整量应该用在线刀具补偿连续加以补偿。

17.2 在线刀具补偿

通道 1 中的加工主程序

程序代码	注释
...	
G1 G19 F10 G90	; 初始设置
T1 D1	; 选择当前刀具
S100 M3 Y100	; 启动主轴，运行到
	; 起始位置
FTOCON	; 激活在线补偿
INIT (2, "/_N_MPF_DIR/_N_ABRICHT_MPF", "S")	; 选择程序
	; 在通道 2 中
START(2)	; 启动程序
	; 在通道 2 中
Y200	; 运行至目标位置
...	
M30	

通道 2_N_ABRICHT_MPF 中的修整程序

程序代码	注释
...	
FCTDEF(1, -1000, 1000, -\$AA_IW[V], 1)	; 定义函数
PUTFTOCF (1, \$AA_IW[V], 2, 1)	; 写入在线刀具长度补偿
	; 连续
U-0.05 G1 F0.01 G91	; 用于修整的进给运动
...	
M30	

说明
进行修整时，V 轴平行与 Y 轴运行，也就是说，长度 2 位于 Y 轴上，因此须加以补偿。

17.2.5 不连续写入在线刀具补偿

PUTFTOC

每写入一个该指令，便写入一个补偿值。

PUTFTOC(<值>, <长度 1_2_3>, <通道号>, <主轴号>)

Put Fine Tool-Offset-Compensation

长度 1、2 或 3 的磨损尺寸按写入值在线变化。

说明

指定主轴号也可以为当前未生效（指几何尺寸未生效）的磨削刀具激活在线刀具补偿。

如果省略了通道号，在线补偿会在自身通道中激活。

如果省略了主轴号，在线补偿会对当前刀具生效。

17.2.6 关于在线补偿的提示

换刀时的特性

- 如果从最后一个换刀或换刀沿开始 FTOCON 指令就已经激活了，那么在换刀时，控制系统内部会输出一个带重新同步的预处理停止指令。
- 没有预处理停止指令，也可执行换刀沿。

说明

T 号可同时用于换刀和在线刀具补偿。

指令 M6 不可同时用于换刀和在线刀具补偿。

加工平面和坐标转换

- 指令 FTOCON 只能用于坐标转换“斜轴”。
- 指令 FTOCON 指令不支持行平面切换(例如: 从 G17 切换至 G18)和坐标转换切换，除了在 FTOCOF 指令状态下。

复位和运行方式切换

- 当在线补偿激活时，NC STOP 和 M2/M30 程序结束指令会延迟一段时间，直到系统完成了补偿。
- NC RESET 会立即撤销在线刀具补偿。
- 在线刀具补偿作用于 AUTO 模式下的当前程序。

17.3 在线刀具半径补偿

边界条件

- 在线刀具补偿会遵循当前设定的限值（比如：速度限值）附加在编程的轴运动量上。如果 DRF 偏移和在线补偿同时供轴使用，则系统首先考虑 DRF 偏移。
- 在不超出最大加速度的前提下，系统使轴以 JOG 速度走完在线补偿。
FTOCON 指令会考虑以下通道专用的机床数据：
MD20610 \$MC_ADD_MOVE_ACCEL_RESERVE
通过该数据可为补偿运动预留一定的加速度裕量，轴因此可立即启动补偿运动。
- 使用 G74 回参考点时，现有的在线补偿会被删除。
- 使用 M6 进行换刀时，精补偿不会被撤销。

17.3 在线刀具半径补偿

简介

当刀具纵轴与轮廓相互垂直时，补偿值可作为长度补偿值作用于三个几何轴中的其中一个(即在线刀具长度补偿)。

如果不满足此条件，那么补偿值可作为实际半径补偿值输入到系统中(即在线刀具半径补偿)。

功能使能

在线刀具半径补偿可由以下机床数据激活:

MD20254 \$MC_ONLINE_CUTCOM_ENABLE (使能在线刀具半径补偿)

激活/撤销

和在线刀具长度补偿一样，在线刀具半径补偿可由 FTOCON 和 FTOCOF 指令来激活/撤销。

参数设置

在线刀具补偿的设置可由 PUTFTOCF 和 PUTFTOC 指令来完成。必须按如下方式设置在线刀具半径补偿的参数“长度 1_2_3”：

参数 <长度 1_2_3> = 4

计入补偿值的磨损尺寸参数。

边界条件

- 只有当选中刀具的半径不等于零时，才能激活刀具半径补偿和在线刀具半径补偿。因此单纯靠在线刀具半径补偿是无法实现加工操作的。
- 在线补偿值应小于初始半径，这样在进行补偿运动时就不会超过允许的动态响应裕量。
- 在磨削刀具和车刀(类型号 400 - 599)中使用在线刀具半径补偿时，补偿值会考虑刀沿位置，也就是说，当刀具半径补偿激活时，它作为半径补偿值作用于刀沿位置确定的轴；当刀具半径补偿关闭时，它作为长度补偿值作用于刀沿位置确定的轴。
在所有其他的刀具类型中，只有在使用 G41 或 G42 来激活刀具半径补偿时，系统才会考虑补偿值。使用 G40 撤销刀具半径补偿时，系统使轴走完补偿值。

17.4 磨削专用的刀具监控

17.4.1 简介

激活

刀具监控由几何尺寸监控和转速监控组成，且对每个磨削刀具(刀具类型号：400 至 499)都有效。

选择

选择方式有：

- 在零件程序中编写 TMON
或者
- 选择刀具类型号为奇数的磨削刀具的长度补偿来自动选择

说明

监控的自动选择必须通过以下通道专用的机床数据设置：

MD20350 \$MC_TOOL_GRIND_AUTO_TMON。

17.4 磨削专用的刀具监控

监控生效

对磨削刀具的监控一直保持生效，直到通过零件程序指令 TMOF 将其撤销。

说明

当两把刀具位于不同的主轴时，对其中一把刀具的监控不会因为选择了监控另一把刀具而撤销。

对于一个主轴而言，在任何时候只能有一把刀具和一个监控生效。

激活的监控在复位后继续保持生效。

17.4.2 几何尺寸监控

功能

监控对象有：

- 当前的砂轮半径
与
- 当前的砂轮宽度

系统会比较当前的砂轮半径会与参数 \$TC_TPG3 的值。

比较半径时，系统会考虑第一个刀沿 (D1) 的参数 \$TC_TPG9 中确定的补偿值。

当前的砂轮宽度通常是由修整循环计算得出的，且可以输入到磨削刀具的参数 \$TC_TPG5 中。在监控激活时，系统会比较该值与参数 \$TC_TPG4 的值。

何时监控？

也可在在线刀具激活时对砂轮半径进行监控：

- 在激活监控时
- 在当前半径(在线刀具补偿、磨损参数)或当前宽度(\$TC_TPG5)变化时

监控响应

如果当前砂轮半径小于参数 \$TC_TPG3 的值，或当前砂轮宽度(\$TC_TPG5)小于参数 \$TC_TPG4 的值，PLC 接口上进给轴/主轴专用的信号 DB31 的位 DBX83.3 会置 1。否则置 0。

DB31, ... DBX83.3 = 1 ⇒ 几何尺寸监控已响应

DB31, ... DBX83.3 = 0 ⇒ 几何尺寸监控未响应

说明

系统内部无任何障响应。

17.4.3 转速监控

功能

监控对象是最大砂轮圆周速度(参数 \$TC_TPG7)或最大主轴转速(参数 \$TC_TPG6)。

单位是:

- 砂轮圆周速度 $\text{m} \cdot \text{s}^{-1}$
- 主轴转速 rpm

监控循环进行, 始终以第一个达到的限值为上限对转速进行监控。

何时监控?

对转速设定值是否超出最大转速的监控是循环进行的, 并考虑了主轴倍率。

何时重新计算转速限值?

重新计算转速限值的情况有:

- 在选择监控时
- 在修改在线补偿值时(磨损参数)

监控响应

转速监控发出响应后:

- 转速被限制在转速限值内
与
- 输出接口信号:
DB31, ... DBX83.6 (转速监控)
。

DB31, ... DBX83.6 = 1 ⇒ 达到转速监控限值

17.4 磨削专用的刀具监控

DB31, ... DBX83.6 = 0 ⇒ 未达到转速监控限值

说明
系统内部无任何障响应。

17.4.4 激活/撤销刀具监控

零件程序指令

可使用以下零件程序指令来激活或撤销磨削刀具专用的监控，不管该刀具是否当前有效。

指令	含义
TMON tool monitoring on	在通道中激活当前刀具的监控。
TMOF tool monitoring off	在通道中撤销当前刀具的监控。
TMON (T 号) tool monitoring on (T 号)	激活由 T 号确定的刀具（非当前刀具）的监控。
TMOF (T 号) tool monitoring off (T 号)	取消由 T 号确定的刀具（非当前刀具）的监控。
TMOF (0) tool monitoring off (0)	撤销所有刀具的监控。

17.5 恒定砂轮圆周速度(GWPS)

17.5.1 简介

什么是 GWPS?

通常对于砂轮而言不编程主轴转速，而是砂轮圆周速度（英语简称 GWPS）。该变量是由工艺流程决定的(例如:砂轮特性参数、材料兼容性等)。系统通过写入程序的 GWPS 和当前砂轮半径计算出主轴转速。

说明

可为磨削刀具(刀具类型号 400 - 499)选择选择 GWPS。

转速计算

转速的计算公式为:

$$n[\text{min}^{-1}] = \frac{\text{GWPS} [\text{m} \cdot \text{s}^{-1}] * 60}{2\pi * R[\text{m}]}$$

说明

可为磨削刀具(400 - 499)选择和编写 GWPS。

计算半径(参数 \$TC_TPG9)时系统会考虑到磨损尺寸。

该功能也适用于斜砂轮/斜轴。

通过 \$TC_TPG9 指定的补偿量会附加到对应的磨损尺寸和基本尺寸（视刀具类型而定）上。

当参数 \$TC_TPG8(砂轮倾斜角度)为正值时，总和除以“cos”(\$TC_TPG8); 参数为负值时，总和除以“sin”(\$TC_TPG8)。

何时重新计算转速?

出现以下事件时系统会重新计算转速:

- 程序写入了 GWPS
- 修改了在线补偿值(磨损尺寸参数)

17.5 恒定砂轮圆周速度(GWPS)

17.5.2 激活/撤销以及编程 GWPS，系统变量

零件程序指令

GWPS 可由以下零件程序指令来激活/撤销:

指令	含义
GWPSON grinding wheel peripheral speed on	在通道中激活当前刀具的 GWPS。
GWPSOF grinding wheel peripheral speed off	在通道中撤销当前刀具的 GWPS。
GWPSON(T 号) grinding wheel peripheral speed on (T 号)	激活由 T 号确定的刀具（非当前刀具）的 GWPS。
GWPSOF(T 号) grinding wheel peripheral speed off (T 号)	撤销由 T 号确定的刀具（非当前刀具）的 GWPS。
S[主轴号] = 值	编程恒定砂轮圆周速度。 值得单位由基本单位制决定(m/s 或 ft/s)。

文档：
编程手册之基本原理分册

说明

通过参数 \$TC_TPG1 可为刀具指定一个主轴。GWPS(见上)激活后，该主轴的每个后续 S 值都是砂轮圆周速度。

如果要为 GWPS 已激活的一个主轴上的一把新刀具选择 GWPS，则首先必须用 GWPSOF 来撤销现有的 GWPS，否则会发出报警。

可以为同一个通道内装有不同磨削刀具的多个主轴激活 GWPS。

用 GWPSON 激活 GWPS 不会自动激活刀具长度补偿、几何尺寸监控和转速监控。撤销 GWPS 时， 最后一个计算出的转速作为设定值保留。

\$P_GWPS[主轴号]

用该系统变量可以从零件程序中确定 GWPS 是否对一个特定的主轴生效。

TRUE: 主轴的 GWPS 编程已激活
FALSE: 主轴的 GWPS 编程未激活

文档：
编程手册之基本原理分册

17.5.3 GWPS 在所有运行方式中生效

简介

通过该功能可以在上电后立即激活主轴的恒定砂轮圆周速度(GWPS)，确保在运行方式切换、复位或零件程序结束后，GWPS 继续保持生效。

该功能可由以下机床数据激活：

MD35032 \$MA_SPIND_FUNC_RESET_MODE (GWPS 功能设置)

上电后激活 GWPS

磨削专用的刀具可由以下机床数据定义：

MD20110 \$MC_RESET_MODE_MASK

MD20120 \$MC_TOOL_RESET_VALUE

MD20130 \$MC_CUTTING_EDGE_RESET_VALUE

说明

MD35032 \$MA_SPIND_FUNC_RESET_MODE

在设置了上述机床数据、设置了磨削专用的刀具（刀具类型号 400 - 499、MD20110、MD20120、MD20130）并指定了主轴（参数 \$TC_TPG1）后，该主轴的 GWPS 激活。该通道中所有其他主轴的 GWPS 都被撤销。

\$TC_DP12[1,1] = 0	; 磨损长度 1
\$TC_DP13[1,1] = 0	; 磨损长度 2
\$TC_DP21[1,1] = 300	; 基本长度 1
\$TC_DP22[1,1] = 400	; 基本长度 2
\$TC_TPG1[1] = 1	; 主轴号
\$TC_TPG8[1] = 0	; 砂轮倾斜角度
\$TC_TPG9[1] = 3	; 用于半径计算的参数号

刀具 T5(斜砂轮)的数据

\$TC_DP1[5,1] = 401	; 刀具类型
\$TC_DP3[5,1] = 120	; 长度 1
\$TC_DP4[5,1] = 30	; 长度 2
\$TC_DP12[5,1] = 0	; 磨损长度 1
\$TC_DP13[5,1] = 0	; 磨损长度 2
\$TC_DP21[5,1] = 100	; 基本长度 1
\$TC_DP22[5,1] = 150	; 基本长度 2
\$TC_TPG1[5] = 2	; 主轴号
\$TC_TPG8[5] = 45	; 砂轮倾斜角度
\$TC_TPG9[5] = 3	; 用于半径计算的参数号

编程

程序代码	注释
N20 T1 D1	; 选择 T1 和 D1
N25 S1=1000 M1=3	; 主轴 1 转速 1000 rpm
N30 S2=1500 M2=3	; 主轴 2 转速 1500 rpm
...	
N40 GWPSON	; 为当前刀具 T1 激活 GWPS
N45 S[\$P_AGT[1]]=60	; 将当前刀具的 GWPS 设置为 60 米/秒 n=1909.85 rpm
...	
N50 GWPSON(5)	; 为刀具 5(第 2 主轴)激活 GWPS
N55 S[\$TC_TPG1[5]]=40	; 将主轴 2 的 GWPS 设置为 40 米/秒 n=1909.85 rpm
...	
N60 GWPSOF	; 撤销当前刀具的 GWPS

17.7 数据表

程序代码	注释
N65 GWPSOF (5)	； 撤销刀具 5 (主轴 2) 的 GWPS
...	

更多信息(参见章节 “P5: 往复运动 - 仅限 840D sl (页 681)”)。

补充文档

- 功能手册之基本功能分册； 章节 “进给率(V1)”
- 功能手册之同步动作分册

17.6 边界条件

17.6.1 换刀和在线刀具补偿组合使用

换刀

M6 换刀指令和在线刀具补偿不能组合使用。

17.7 数据表

17.7.1 机床数据

17.7.1.1 通用机床数据

编号	名称: \$MN_	描述
18094	MM_NUM_CC_TDA_PARAM	TDA 的数量
18096	MM_NUM_CC_TOA_PARAM	TOA 的数量
18100	MM_NUM_CUTTING_EDGES_IN_TOA	每个 TOA 的刀具补偿数

17.7.1.2 通道专用机床数据

编号	名称: \$MC_	描述
20254	ONLINE_CUTCOM_ENABLE	激活在线刀具半径补偿
20350	TOOL_GRIND_AUTO_TMON	激活自动刀具监控
20610	ADD_MOVE_ACCEL_RESERVE	为叠加运动保留的加速裕量

17.7.1.3 进给轴/主轴专用机床数据

编号	名称: \$MA_	描述
32020	JOG_VELO	JOG 轴速度
35032	SPIND_FUNC_RESET_MODE	砂轮圆周速度的设置

17.7.2 信号

17.7.2.1 来自进给轴/主轴的信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
几何尺寸监控	DB31, ...DBX83.3	DB390x.DBX2001.3
转速监控	DB31, ...DBX83.6	DB390x.DBX2001.6
砂轮圆周速度生效	DB31, ...DBX84.1	DB390x.DBX2002.1

Z2: NC/PLC 接口信号

18.1 NCK 数字量/模拟量输入/输出(A4)

18.1.1 发送至 NC 信号(DB10)

PLC 向 NC 发送的信号一览

DB10	PLC → NC 的接口信号							
DBB	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
0	禁用 NCK 数字量输入							
	非硬件数字量输入*)				板载输入 **)			
	输入 8	输入 7	输入 6	输入 5	输入 4	输入 3	输入 2	输入 1
1	通过 PLC 置位 NCK 数字量输入							
	非硬件数字量输入*)				板载输入 **)			
	输入 8	输入 7	输入 6	输入 5	输入 4	输入 3	输入 2	输入 1
4	禁用 NCK 数字量输出							
	非硬件数字量输出*)				板载输出 **)			
	输出 8	输出 7	输出 6	输出 5	输出 4	输出 3	输出 2	输出 1
5	NCK 数字量输出的改值位							
	非硬件数字量输出*)				板载输出 **)			
	输出 8	输出 7	输出 6	输出 5	输出 4	输出 3	输出 2	输出 1
6	PLC 给出的 NCK 数字量输出的设置值							
	非硬件数字量输出*)				板载输出 **)			
	输出 8	输出 7	输出 6	输出 5	输出 4	输出 3	输出 2	输出 1
7	NCK 数字量输出的写值位							
	非硬件数字量输出*)				板载输出 **)			
	输出 8	输出 7	输出 6	输出 5	输出 4	输出 3	输出 2	输出 1

18.1 NCK 数字量/模拟量输入/输出(A4)

注:

*) NCK 数字量输入/输出 (位 4 到位 7) 虽然并非硬件输入/输出, 但也可由 PLC 控制。这些位因此还可用于 NCK 和 PLC 之间的通讯。

**) NCK 数字量输入/输出 1 到 4 是板载硬件输入/输出。

DB10	PLC → NC 的接口信号							
DBB	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
122	禁用 NCK 数字量输入							
	输入 16	输入 15	输入 14	输入 13	输入 12	输入 11	输入 10	输入 9
123	通过 PLC 置位 NCK 数字量输入							
	输入 16	输入 15	输入 14	输入 13	输入 12	输入 11	输入 10	输入 9
124	禁用 NCK 数字量输入							
	输入 24	输入 23	输入 22	输入 21	输入 20	输入 19	输入 18	输入 17
125	通过 PLC 置位 NCK 数字量输入							
	输入 24	输入 23	输入 22	输入 21	输入 20	输入 19	输入 18	输入 17
126	禁用 NCK 数字量输入							
	输入 32	输入 31	输入 30	输入 29	输入 28	输入 27	输入 26	输入 25
127	通过 PLC 置位 NCK 数字量输入							
	输入 32	输入 31	输入 30	输入 29	输入 28	输入 27	输入 26	输入 25
128	禁用 NCK 数字量输入							
	输入 40	输入 39	输入 38	输入 37	输入 36	输入 35	输入 34	输入 33
129	通过 PLC 置位 NCK 数字量输入							
	输入 40	输入 39	输入 38	输入 37	输入 36	输入 35	输入 34	输入 33
130	禁用 NCK 数字量输出							
	输出 16	输出 15	输出 14	输出 13	输出 12	输出 11	输出 10	输出 9
131	NCK 数字量输出的改值位							
	输出 16	输出 15	输出 14	输出 13	输出 12	输出 11	输出 10	输出 9
132	PLC 给出的 NCK 数字量输出的设置值							
	输出 16	输出 15	输出 14	输出 13	输出 12	输出 11	输出 10	输出 9
133	NCK 数字量输出的写值位							
	输出 16	输出 15	输出 14	输出 13	输出 12	输出 11	输出 10	输出 9
134	禁用 NCK 数字量输出							
	输出 24	输出 23	输出 22	输出 21	输出 20	输出 19	输出 18	输出 17
135	NCK 数字量输出的改值位							

18.1 NCK 数字量/模拟量输入/输出(A4)

	输出 24	输出 23	输出 22	输出 21	输出 20	输出 19	输出 18	输出 17
136	PLC 给出的 NCK 数字量输出的设置值							
	输出 24	输出 23	输出 22	输出 21	输出 20	输出 19	输出 18	输出 17
137	NCK 数字量输出的写值位							
	输出 24	输出 23	输出 22	输出 21	输出 20	输出 19	输出 18	输出 17
138	禁用 NCK 数字量输出							
	输出 32	输出 31	输出 30	输出 29	输出 28	输出 27	输出 26	输出 25
139	NCK 数字量输出的改值位							
	输出 32	输出 31	输出 30	输出 29	输出 28	输出 27	输出 26	输出 25
140	PLC 给出的 NCK 数字量输出的设置值							
	输出 32	输出 31	输出 30	输出 29	输出 28	输出 27	输出 26	输出 25
141	NCK 数字量输出的写值位							
	输出 32	输出 31	输出 30	输出 29	输出 28	输出 27	输出 26	输出 25
142	禁用 NCK 数字量输出							
	输出 40	输出 39	输出 38	输出 37	输出 36	输出 35	输出 34	输出 33
143	NCK 数字量输出的改值位							
	输出 40	输出 39	输出 38	输出 37	输出 36	输出 35	输出 34	输出 33
144	PLC 给出的 NCK 数字量输出的设置值							
	输出 40	输出 39	输出 38	输出 37	输出 36	输出 35	输出 34	输出 33
145	NCK 数字量输出的写值位							
	输出 40	输出 39	输出 38	输出 37	输出 36	输出 35	输出 34	输出 33
146	禁用 NCK 模拟量输入							
	输入 8	输入 7	输入 6	输入 5	输入 4	输入 3	输入 2	输入 1
147	NCK 模拟量输入的写值位							
	输入 8	输入 7	输入 6	输入 5	输入 4	输入 3	输入 2	输入 1
148, 149	PLC 给出的 NCK 模拟量输入 1 的设置值							
150, 151	PLC 给出的 NCK 模拟量输入 2 的设置值							
152, 153	PLC 给出的 NCK 模拟量输入 3 的设置值							
154, 155	PLC 给出的 NCK 模拟量输入 4 的设置值							
156, 157	PLC 给出的 NCK 模拟量输入 5 的设置值							
158, 159	PLC 给出的 NCK 模拟量输入 6 的设置值							
160, 161	PLC 给出的 NCK 模拟量输入 7 的设置值							

18.1 NCK 数字量/模拟量输入/输出(A4)

162, 163	PLC 给出的 NCK 模拟量输入 8 的设置值							
166	NCK 模拟量输出的改值位							
	输出 8	输出 7	输出 6	输出 5	输出 4	输出 3	输出 2	输出 1
167	NCK 模拟量输出的写值位							
	输出 8	输出 7	输出 6	输出 5	输出 4	输出 3	输出 2	输出 1
168	禁用 NCK 模拟量输出							
	输出 8	输出 7	输出 6	输出 5	输出 4	输出 3	输出 2	输出 1
170, 171	PLC 给出的 NCK 模拟量输出 1 的设置值							
172, 173	PLC 给出的 NCK 模拟量输出 2 的设置值							
174, 175	PLC 给出的 NCK 模拟量输出 3 的设置值							
176, 177	PLC 给出的 NCK 模拟量输出 4 的设置值							
178, 179	PLC 给出的 NCK 模拟量输出 5 的设置值							
180, 181	PLC 给出的 NCK 模拟量输出 6 的设置值							
182, 183	PLC 给出的 NCK 模拟量输出 7 的设置值							
184, 185	PLC 给出的 NCK 模拟量输出 8 的设置值							

PLC 向 NC 发送的信号说明

DB10 DBB0, 122, 124, 126, 128	禁用 NCK 数字量输入	
脉冲沿计算：不支持	信号更新速率：周期	
信号稳态为 1 或输出上升沿 0 → 1	PLC 禁用 NCK 数字量输入。该输入由控制系统自动置 0。	
信号稳态为 0 或输出下降沿 1 → 0	NCK 数字量输入激活。输入上的信号状态可从零件程序直接读出。	
关联信号	DB10 DBB1（通过 PLC 置位 NCK 数字量输入） DB10 DBB60（NCK 数字量输入的实际值） MD10350 \$MN_FASTIO_DIG_NUM_INPUTS	

18.1 NCK 数字量/模拟量输入/输出(A4)

DB10 DBB1, 123, 125, 127, 129	通过 PLC 置位 NCK 数字量输入	
脉冲沿计算：不支持	信号更新速率：周期	
信号稳态为 1 或输出上升沿 0 → 1	PLC 将 NCK 数字量输入置 1。硬输入的信号状态及接口信号“禁用 NCK 数字量输入”导致的输入禁用因此失效。	
信号稳态为 0 或输出下降沿 1 → 0	允许从零件程序读取 NCK 输入的信号状态。前提条件是 PLC 没有禁用该输入，即接口信号“禁用 NCK 数字量输入”为 0。	
关联信号	DB10 DBB0（禁用 NCK 数字量输入） DB10 DBB60（NCK 数字量输入的实际值） MD10350 \$MN_FASTIO_DIG_NUM_INPUTS	

DB10 DBB4, 130, 134, 138, 142	禁用 NCK 数字量输出	
脉冲沿计算：不支持	信号更新速率：周期	
信号稳态为 1 或输出上升沿 0 → 1	NCK 数字量输出禁用。硬输出上因此输出“0 V”。	
信号稳态为 0 或输出下降沿 1 → 0	NCK 数字量输出激活。硬输出上因此可输出由零件程序或 PLC 设置的值。	
关联信号	DB10 DBB5（NCK 数字量输出的改值位） DB10 DBB7（NCK 数字量输出的写值位） DB10 DBB6（通过 PLC 置位 NCK 数字量输出） MD10360 \$MN_FASTIO_DIG_NUM_OUTPUTS	

DB10 DBB5, 131, 135, 139, 143	NCK 数字量输出的改值位	
脉冲沿计算：不支持	信号更新速率：周期	
信号稳态为 1 或输出上升沿 0 → 1	在该信号输出上升沿 0 → 1 时，当前的 NCK 值被 PLC 改写，新值为接口信号“PLC 给出的 NCK 数字量输出的设置值”。通过零件程序直接设置的当前的 NCK 值因此丢失。 由设置值指定的信号状态构成新的 NCK 值。	
信号稳态为 0 或输出下降沿 1 → 0	NCK 只会在该信号输出上升沿 0 → 1 时计算该信号，在下一个 PLC 周期中，PLC 用户程序会再次将该信号复位为 0。	

18.1 NCK 数字量/模拟量输入/输出(A4)

DB10 DBB5, 131, 135, 139, 143	NCK 数字量输出的改值位
特殊情况， 出错，...	注： 改值位（上升沿 0 → 1）和写值位（信号稳态为 1）使用的是同一个 PLC 设置值(DB10 DBB6)。这两个位可通过 PLC 用户程序避免同时激活。
关联信号	DB10 DBB4（禁用 NCK 数字量输出） DB10 DBB7（NCK 数字量输出的写值位） DB10 DBB6（PLC 给出的 NCK 数字量输出的设置值） MD10360 \$MN_FASTIO_DIG_NUM_OUTPUTS

DB10 DBB6, 132, 136, 140, 144	PLC 给出的 NCK 数字量输出的设置值
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	通过设置值 PLC 可修改硬数字量输出上的信号状态。可以采取两种方式： - 通过“改值位”： 当“改值位”信号输出上升沿 0 → 1 时，PLC 用“设置值”改写当前的“NCK 值”。该值成为新的“NCK 值”。 - 通过“写值位”： 当“写值位”稳态为 1 时，当前的“NCK 值”切换为“PLC 值”。PLC 值由“设置值”指定。 在设置值为 1 时，硬输出上输出 1 电平。在设置值为 0 时，硬输出上输出 0 电平。对应的电压值参见： 文档： NCU 7x0.3 PN 手册
信号稳态为 0 或输出下降沿 1 → 0	NCK 只会在该信号输出上升沿 0 → 1 时计算该信号，在下一个 PLC 周期中，PLC 用户程序会再次将该信号复位为 0。
特殊情况， 出错，...	注： 改值位（上升沿 0 → 1）和写值位（信号稳态为 1）使用的是同一个 PLC 设置值(DB10 DBB6)。这两个位可通过 PLC 用户程序避免同时激活。
关联信号	DB10 DBB4（禁用 NCK 数字量输出） DB10 DBB5（NCK 数字量输出的改值位） DB10 DBB7（NCK 数字量输出的写值位） MD10360 \$MN_FASTIO_DIG_NUM_OUTPUTS

18.1 NCK 数字量/模拟量输入/输出(A4)

DB10 DBB7, 133, 137, 141, 145	NCK 数字量输出的写值位	
脉冲沿计算：不支持	信号更新速率：周期	
信号稳态为 1 或输出上升沿 0 → 1	硬数字量输出上输出 PLC 值，而不是 NCK 值。该 PLC 值需要事先由接口信号“PLC 给出的 NCK 数字量输出的设置值”确定。 当前 NCK 值不会丢失。	
信号稳态为 0 或输出下降沿 1 → 0	硬数字量输出上输出 NCK 值。	
特殊情况， 出错，...	注： 改值位（上升沿 0 → 1）和写值位（信号稳态为 1）使用的是同一个 PLC 设置值(DB10 DBB6)。这两个位可通过 PLC 用户程序避免同时激活。	
关联信号	DB10 DBB4（禁用 NCK 数字量输出） DB10 DBB5（NCK 数字量输出的改值位） DB10 DBB6（PLC 给出的 NCK 数字量输出的设置值） MD10360 \$MN_FASTIO_DIG_NUM_OUTPUTS	

DB10 DBB146	禁用 NCK 模拟量输入	
脉冲沿计算：不支持	信号更新速率：周期	
信号稳态为 1 或输出上升沿 0 → 1	PLC 禁用 NCK 模拟量输入。该输入由控制系统自动置 0。	
信号稳态为 0 或输出下降沿 1 → 0	NCK 模拟量输入激活。只有 PLC 没有将该 NCK 输入对应的写值位设为 0，就可以直接在零件程序中直接读取该输入上的模拟量值。	
关联信号	DB10 DBB147（NCK 模拟量输入的写值位） DB10 DBB148（PLC 给出的 NCK 模拟量输入的设置值） DB10 DBB199（NCK 模拟量输入的实际值） MD10300 \$MN_FASTIO_ANA_NUM_INPUTS	

DB10 DBB147	NCK 模拟量输入的写值位	
脉冲沿计算：不支持	信号更新速率：周期	
信号稳态为 1 或输出上升沿 0 → 1	PLC 给出的设置值作为模拟量值生效 (设置值来自接口信号“PLC 给出的 NCK 模拟量输入的设置值”)。	

18.1 NCK 数字量/模拟量输入/输出(A4)

DB10 DBB147	NCK 模拟量输入的写值位
信号稳态为 0 或输出下降沿 1 → 0	允许从零件程序读取 NCK 输入上的模拟量值。前提条件是 PLC 没有禁用该输入，即接口信号“禁用 NCK 模拟量输入”为 0。
关联信号	DB10 DBB146（禁用 NCK 模拟量输入） DB10 DBB148-163（PLC 给出的 NCK 模拟量输入的设置值） DB10 DBB199-209（NCK 模拟量输入的实际值） MD10300 \$MN_FASTIO_ANA_NUM_INPUTS

DB10 DBB148 - 163	PLC 给出的 NCK 模拟量输入的设置值
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	通过设置值 PLC 可给定某个模拟量值。是用硬输入上的模拟量值还是用 PLC 通过设置值给出的模拟量值还取决于 PLC 对接口信号“NCK 模拟量输入的写值位”的控制。 如果该信号置 1，则 PLC 给出的设置值生效。 PLC 给出的设置值作为定点数（16 位，含符号）以补码表示。
关联信号	DB10 DBB146（禁用 NCK 模拟量输入） DB10 DBB147（NCK 模拟量输入的写值位） DB10 DBB199-209（NCK 模拟量输入的实际值） MD10300 \$MN_FASTIO_ANA_NUM_INPUTS

DB10 DBB166	NCK 模拟量输出的改值位
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	在该信号输出上升沿 0 → 1 时，当前的 NCK 值被 PLC 改写，新值为接口信号“PLC 给出的 NCK 模拟量输出的设置值”。通过零件程序直接设置的当前的 NCK 值因此丢失。 PLC 通过设置值给出的模拟量值构成新的 NCK 值。
信号稳态为 0 或输出下降沿 1 → 0	NCK 只会在该信号输出上升沿 0 → 1 时计算该信号，在下一个 PLC 周期中，PLC 用户程序会再次将该信号复位为 0。

18.1 NCK 数字量/模拟量输入/输出(A4)

DB10 DBB166	NCK 模拟量输出的改值位
特殊情况， 出错，…	注： 改值位（上升沿 0 → 1）和写值位（信号稳态为 1）使用的是同一个 PLC 设置值。这两个位可通过 PLC 用户程序避免同时激活。
关联信号	DB10 DBB168（禁用 NCK 模拟量输出） DB10 DBB167（NCK 模拟量输出的写值位） DB10 DBB170-185（PLC 给出的 NCK 模拟量输出的设置值） MD10310 \$MN_FASTIO_ANA_NUM_OUTPUTS

DB10 DBB167	NCK 模拟量输出的写值位
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	硬模拟量输出上输出 PLC 值，而不是 NCK 值。该 PLC 值需要事先由接口信号“PLC 给出的 NCK 模拟量输出的设置值”确定。 当前 NCK 值不会丢失。
信号稳态为 0 或输出下降沿 1 → 0	硬模拟量输出上输出 NCK 值。
特殊情况， 出错，…	注： 改值位（上升沿 0 → 1）和写值位（信号稳态为 1）使用的是同一个 PLC 设置值。这两个位可通过 PLC 用户程序避免同时激活。
关联信号	DB10 DBB168（禁用 NCK 模拟量输出） DB10 DBB166（NCK 模拟量输出的改值位） DB10 DBB170-185（PLC 给出的 NCK 模拟量输出的设置值） MD10310 \$MN_FASTIO_ANA_NUM_OUTPUTS

DB10 DBB168	禁用 NCK 模拟量输出
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	NCK 模拟量输出禁用。硬输出上因此输出“0 V”。

18.1 NCK 数字量/模拟量输入/输出(A4)

DB10 DBB168	禁用 NCK 模拟量输出
信号稳态为 0 或输出下降沿 1 → 0	NCK 模拟量输出激活。硬输出上因此可输出由零件程序或 PLC 设置的值。
关联信号	DB10 DBB166 (NCK 模拟量输出的改值位) DB10 DBB167 (NCK 模拟量输出的写值位) DB10 DBB170-185 (PLC 给出的 NCK 模拟量输出的设置值) MD10310 \$MN_FASTIO_ANA_NUM_OUTPUTS

DB10 DBB170 - 185	PLC 给出的 NCK 模拟量输出的设置值
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	通过设置值 PLC 可修改硬模拟量输出上的值。可以采取两种方式： - 通过“改值位”： 当“改值位”信号输出上升沿 0 → 1 时，PLC 用“设置值”改写当前的“NCK 值”。该值成为新的“NCK 值”。 - 通过“写值位”： 当“写值位”稳态为 1 时，当前的“NCK 值”切换为“PLC 值”。PLC 值由“设置值”指定。 PLC 给出的设置值作为定点数（16 位，含符号）以补码表示。
信号稳态为 0 或输出下降沿 1 → 0	NCK 只会在该信号输出上升沿 0 → 1 时计算该信号，在下一个 PLC 周期中，PLC 用户程序会再次将该信号复位为 0。
特殊情况， 出错，...	注： 改值位（上升沿 0 → 1）和写值位（信号稳态为 1）使用的是同一个 PLC 设置值。这两个位可通过 PLC 用户程序避免同时激活。
关联信号	DB10 DBB168 (禁用 NCK 模拟量输出) DB10 DBB166 (NCK 模拟量输出的改值位) DB10 DBB167 (NCK 模拟量输出的写值位) MD10310 \$MN_FASTIO_ANA_NUM_OUTPUTS

18.1.2 从 NC 发出的信号(DB10)

NC 向 PLC 发送的信号一览

DB10	NC → PLC 的接口信号							
DBB	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
60	NCK 数字量输入的实际值							
					板载输入 **)			
					输入 4	输入 3	输入 2	输入 1
64	NCK 数字量输出的设定值							
	非硬件数字量输入*)				板载输出 **)			
	输出 8	输出 7	输出 6	输出 5	输出 4	输出 3	输出 2	输出 1
186	NCK 数字量输入的实际值							
	输入 16	输入 15	输入 14	输入 13	输入 12	输入 11	输入 10	输入 9
187	NCK 数字量输入的实际值							
	输入 24	输入 23	输入 22	输入 21	输入 20	输入 19	输入 18	输入 17
188	NCK 数字量输入的实际值							
	输入 32	输入 31	输入 30	输入 29	输入 28	输入 27	输入 26	输入 25
189	NCK 数字量输入的实际值							
	输入 40	输入 39	输入 38	输入 37	输入 36	输入 35	输入 34	输入 33
190	NCK 数字量输出的设定值							
	输出 16	输出 15	输出 14	输出 13	输出 12	输出 11	输出 10	输出 9
191	NCK 数字量输出的设定值							
	输出 24	输出 23	输出 22	输出 21	输出 20	输出 19	输出 18	输出 17
192	NCK 数字量输出的设定值							
	输出 32	输出 31	输出 30	输出 29	输出 28	输出 27	输出 26	输出 25
193	NCK 数字量输出的设定值							
	输出 40	输出 39	输出 38	输出 37	输出 36	输出 35	输出 34	输出 33
注:								
*) NCK 数字量输入/输出 (位 4 到位 7) 虽然并非硬件输入/输出, 但也可由 PLC 控制。这些位因此还可用于 NCK 和 PLC 之间的通讯。								
**) NCK 数字量输入/输出 1 到 4 是板载硬件输入/输出。								
DB10	NC → PLC 的接口信号							

18.1 NCK 数字量/模拟量输入/输出(A4)

DBB	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
194, 195	NCK 模拟量输入 1 的实际值							
196, 197	NCK 模拟量输入 2 的实际值							
198, 199	NCK 模拟量输入 3 的实际值							
200, 201	NCK 模拟量输入 4 的实际值							
202, 203	NCK 模拟量输入 5 的实际值							
204, 205	NCK 模拟量输入 6 的实际值							
206, 207	NCK 模拟量输入 7 的实际值							
208, 209	NCK 模拟量输入 8 的实际值							
210, 211	NCK 模拟量输出 1 的设定值							
212, 213	NCK 模拟量输出 2 的设定值							
214, 215	NCK 模拟量输出 3 的设定值							
216, 217	NCK 模拟量输出 4 的设定值							
218, 219	NCK 模拟量输出 5 的设定值							
220, 221	NCK 模拟量输出 6 的设定值							
222, 223	NCK 模拟量输出 7 的设定值							
224, 225	NCK 模拟量输出 8 的设定值							

NC 向 PLC 发送的信号说明

DB10 DBB60, 186 - 189	NCK 数字量输入的实际值	
脉冲沿计算：不支持	信号更新速率：周期	
信号稳态为 1 或输出上升沿 0 → 1	NCK 硬数字量输入上为 1 电平。	
信号稳态为 0 或输出下降沿 1 → 0	NCK 硬数字量输入上为 0 电平。	
特殊情况，出错，…	该实际值不受接口信号： DB10 DBB0（禁用 NCK 数字量输入）的影响。	
关联信号	DB10 DBB0（禁用 NCK 数字量输入） MD10350 \$MN_FASTIO_DIG_NUM_INPUTS	

DB10 DBB64, 190 - 193	NCK 数字量输出的设定值	
脉冲沿计算：不支持	信号更新速率：周期	
信号稳态为 1 或输出上升沿 0 → 1	数字量输出的当前 NCK 值（设定值）是 1。	
信号稳态为 0 或输出下降沿 1 → 0	数字量输出的当前 NCK 值（设定值）是 0。	
信号失效条件	该设定值只有在以下条件下才输出给硬输出： - 该输出没有被接口信号“禁用 NCK 数字量输出”禁用 - PLC 没有通过接口信号“NCK 数字量输出的写值位”改写 NCK 值 只要满足上述条件，数字量输出的“设定值”就和“实际值”一致。	
关联信号	DB10 DBB4（禁用 NCK 数字量输出） DB10 DBB5（NCK 数字量输出的改值位） DB10 DBB6（PLC 给出的 NCK 数字量输出的设置值） DB10 DBB7（NCK 数字量输出的写值位） MD10310 \$MN_FASTIO_DIG_NUM_OUTPUTS	

18.1 NCK 数字量/模拟量输入/输出(A4)

DB10 DBB194 - 209	NCK 模拟量输入的实际值
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	将 NCK 模拟量输入上的模拟量值报告给 PLC。 NCK 实际值作为定点数（16 位，含符号）以补码表示。
信号稳态为 0 或输出下降沿 1 → 0	实际值（模拟量值）不受 PLC 的控制，比如：受接口信号“禁用 NCK 模拟量输入”的影响。
关联信号	DB10 DBB146（禁用 NCK 模拟量输入） DB10 DBB147（NCK 模拟量输入的写值位） DB10 DBB148-163（PLC 给出的 NCK 模拟量输入的设置值） MD10300 \$MN_FASTIO_ANA_NUM_INPUTS

DB10 DBB210 - 225	NCK 模拟量输出的设定值
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	将数字量输出的当前 NCK 值（设定值）报告给 PLC。 NCK 设定值作为定点数（16 位，含符号）以补码表示。
信号稳态为 0 或输出下降沿 1 → 0	该设定值只有在以下条件下才输出给硬输出： - 该输出没有被接口信号“禁用 NCK 模拟量输出”禁用 - PLC 没有通过接口信号“NCK 模拟量输出的写值位”改写 NCK 值
关联信号	DB10 DBB168（禁用 NCK 模拟量输出） DB10 DBB166（NCK 模拟量输出的改值位） DB10 DBB170-185（PLC 给出的 NCK 模拟量输出的设置值） DB10 DBB167（NCK 模拟量输出的写值位） MD10310 \$MN_FASTIO_ANA_NUM_OUTPUTS

18.2 分布式系统(B3)

18.2.1 已定义的逻辑功能/ Defines

BUSTYP

名称	值	接口 DB19	含义
MPI	1	DBW100,102,104, 120, 130 位 8-15	操作单元连在 MPI 上, 187.5 Kbaud
OPI	2	"	操作单元连在 MPI 上, 1.5 Kbaud

STATUS

名称	值	接口 DB19	含义
OFFL_REQ_PLC	1	在线接口 1.: DBB124 2.: DBB134	PLC → 操作单元: PLC 发出离线请求, 要求操作单元离线。
OFFL_CONF_PLC	2	在线接口 1.: DBB124 2.: DBB134	操作单元 → PLC: 对 OFFL_REQ_PLC 的应答 该信号的含义取决于 Z_INFO DBB125 或者 DBB135
OFFL_REQ_OP	3	在线接口 1.: DBB124 2.: DBB134	操作单元 → PLC: 操作单元发出离线请求, 请求和该 NCU 断开
OFFL_CONF_OP	4	在线接口 1.: DBB124 2.: DBB134	PLC → 操作单元: 对 OFFL_REQ_OP 的应答 该信号的含义取决于 Z_INFO DBB125 或者 DBB135
ONL_PERM	5	呼叫等待(CW)接口 DBB108	PLC → 操作单元: PLC 告知操作单元它是否在线。 该信号的含义取决于 Z_INFO:DBB109
S_ACT	6	在线接口 1.: DBB124 2.: DBB134	操作单元 → PLC: 操作单元在线或者切换操作焦点。 该信号的含义取决于 Z_INFO DBB125 或者 DBB135

名称	值	接口 DB19	含义
OFFL_REQ_FOC	7	在线接口 1.: DBB124 2.: DBB134	操作单元 → PLC: 操作单元希望从该 NCU 处取回操作焦点
OFFL_CONF_FOC	8	在线接口 1.: DBB124 2.: DBB134	PLC → 操作单元: 对 OFFL_REQ_FOC 的应答 该信号的含义取决于 Z_INFO DBB125 或者 DBB135
ONL_REQ_FOC	9	在线接口 1.: DBB124 2.: DBB134	操作单元 → PLC: 操作单元希望将操作焦点交给该 NCU
ONL_PERM_FOC	10	在线接口 1.: DBB124 2.: DBB134	PLC → 操作单元: 对 ONL_REQ_FOC 的应答 该信号的含义取决于 Z_INFO DBB125 或者 DBB135

Z_INFO

名称	值	接口 DB19	含义
DISC_FOC	9	DBB125 DBB135	操作单元将操作焦点切到另一台 NCU 上。
OK	10	DBB109 位 0-3 DBB125 DBB135	积极应答
CONNECT	11	DBB125 DBB135	操作单元和该 NCU 在线连接。
MMC_LOCKED	13	DBB109 位 0-3 DBB125 DBB135	HMI 设置了切换锁。 在该操作单元上运行了不允许因切换操作而中断的进程。
PLC_LOCKED	14	DBB109 位 0-3 DBB125 DBB135	在 HMI-PLC 接口内设置了 HMI 切换锁。操作单元无法和该 NCU 在线连接, 也无法切换操作焦点。
PRIO_H	15	DBB109 位 0-3 DBB125 DBB135	该 NCU 和具有最高优先级的操作单元在线连接。“呼叫等待”的操作单元无法和该 NCU 在线连接。

STATUS 和 Z_INFO 可以组合使用

名称: STATUS	Z_INFO	含义
OFFL_REQ_PLC	OK	PLC 发出离线请求, 要求在线的操作单元离线。
OFFL_CONF_PLC	OK	操作单元对 PLC 的离线请求作出积极应答。 操作单元接着离线。
OFFL_CONF_PLC	MMC_LOCKED	操作单元对离线请求作出消极应答。 操作单元不离线, 因为上面有不可中断的进程正在运行。
OFFL_REQ_OP	OK	操作单元发出离线请求, 请求和该 NCU 断开。
OFFL_CONF_OP	OK	PLC 对离线请求作出积极应答。 操作单元接着和该 NCU 断开。
OFFL_CONF_OP	PLC_LOCKED	PLC 对操作单元的离线请求作出消极应答。 用户设置了 HMI 切换锁, 操作单元无法离线, MMCx_SHIFT_LOCK = TRUE, x=1 或 2, 表明第 1 或第 2 HMI-PLC 接口。
ONL_PERM	HMI-PLC 在线接口的 编号, OK	PLC 为呼叫等待的操作单元给出在线使能。该操作单元随后 和该 NCU 在线连接。Z_INFO 的内容: 位 0...3:OK 位 4..7: HMI-PLC 在线接口的编号, 操作单元要和该接口连 接: 1 表示第一个 HMI-PLC 在线接口 2 表示第二个 HMI-PLC 在线接口
ONL_PERM	MMC_LOCKED	“呼叫等待”的操作单元无法在线。 该 NCU 上有两个操作单元在线, 上面运行了不可中断的进 程。PLC 无法使这两个操作单元离线
ONL_PERM	PLC_LOCKED	“呼叫等待”的操作单元无法在线。 用户设置了 HMI 切换锁, MMCx_SHIFT_LOCK = TRUE, x=1 或 2, 代表第 1 个或第 2 个 HMI 在线接口。
ONL_PERM	PRIO_H	“呼叫等待”的操作单元无法在线。 该 NCU 上有两个操作单元在线, 其优先级高于“呼叫等待” 的操作单元。 PLC 无法使这两个操作单元离线。
S_ACT	CONNECT	“呼叫等待”的操作单元已在线。 PLC 现在激活 HMI 生命符号监控。

18.2 分布式系统(B3)

名称: STATUS	Z_INFO	含义
S_ACT	DISC_FOCUS	服务器 HMI 删除了该 NCU 的操作焦点。
OFFL_REQ_FOC	OK	服务器 HMI 发出操作焦点的离线请求, 希望删除该 NCU 的操作焦点。
OFFL_CONF_FOC	OK	PLC 对操作焦点的离线请求作出积极应答。 服务器 HMI 现在可删除操作焦点。
OFFL_CONF_FOC	PLC_LOCKED	PLC 对操作焦点的离线请求作出消极应答。 用户设置了 HMI 切换锁, 服务器 HMI 无法删除操作焦点, MMCx_SHIFT_LOCK = TRUE, x=1 或 2, 代表第 1 个或第 2 个 HMI-PLC 接口。
ONL_REQ_FOC	OK	服务器 HMI 发出操作焦点的在线请求, 希望将操作焦点交给该 NCU。
ONL_PERM_FOC	OK	PLC 对操作焦点的在线请求作出积极应答。 服务器 HMI 接着将操作焦点交给该 NCU。
ONL_PERM_FOC	PLC_LOCKED	PLC 对操作焦点的在线请求作出消极应答。用户设置了 HMI 切换锁, 服务器 HMI 无法设置操作焦点, MMCx_SHIFT_LOCK = TRUE, x=1 或 2, 代表第 1 个或第 2 个 HMI-PLC 接口。

18.2.2 多操作面板配多 NC: DB19 中的接口

DB19 中的 HMI/PLC 接口分为 3 个区域。

呼叫等待(CW)接口

如果一个操作单元希望在线, 需要在该接口上执行呼叫等待。

HMI 将其客户端 ID 写入到 ONL_REQUEST 中, 然后等待 PLC 在 ONL_CONFIRM 中返回其客户端 ID。

在 PLC 给出积极应答后, 操作单元传递其参数, 并等待在线许可 (包含在 PAR_STATUS 和 PAR_Z_INFO 中)。

HMI 参数传递:

客户端 ID -> PAR_CLIENT_IDENT

HMI 型号 -> PAR_MMC_TYP

OPI 地址 -> PAR_MSTT_ADR

在 PLC 给出积极应答后，操作单元从 PLC 处得知其 HMI-PLC 在线接口编号：DBB109.4-7。

接着操作单元在线连接，占用指定的在线接口。

在线接口

在一个 NCU 上可以同时有两个操作单元在线。

其中的每个在线操作单元都占据一个单独的在线接口。

在呼叫等待成功结束后，PLC 告知操作单元其在线接口的编号。

PLC 接着将 HMI 参数传递到对应的在线接口。

操作单元在线连接，占用指定的在线接口，HMI 和 PLC 之后便通过该接口进行数据交换。

HMI 数据接口

此处定义了来自/送至 HMI 的有效数据：

- DBB 0-49 接口操作单元 1
- DBB 50-99 接口操作单元 2

这些数据 and 信号主要用于操作单元的运行。

“操作面板切换”生命符号监控

该监控是另一种监控，不可与 HMI 生命符号监控混淆。详细说明参见各信号。

在某些运行状态下，激活了操作单元切换的操作单元（可通过 NETNAMES.INI 设置）必须能够借助一个 PLC 数据来判断是否需要执行呼叫等待方可在线。

示例：

激活了操作单元切换的操作单元无需执行呼叫等待，即可和 NCU 在线连接。

现在该操作单元必须在线，以便开展服务。

协调是在呼叫等待接口的数据 DBW110 中进行的：**M_TO_N_ALIVE**

“多操作面板配多 NCU”生命符号是一个循环计数器，读数由 PLC 循环增加，在上溢时置 1。

在一个操作单元呼叫前，必须根据该生命符号判断 PLC 中是否激活了操作单元切换。

操作步骤：

18.2 分布式系统(B3)

HMI 在 T_0 和 $T_0 + 1$ 这两个时间点上读取生命符号。

情形 1: PLC 对读任务作消极应答: DB19 不存在。操作单元不经过呼叫等待在线。

情形 2: $m\_to\_n\_alive = 0$: 操作单元切换未激活。操作单元不经过呼叫等待在线。

情形 3: $m\_to\_n\_alive(T_0) = m\_to\_n\_alive(T_0+1)$: 操作单元切换未激活。操作单元不经过呼叫等待在线。

情形 4: $m\_to\_n\_alive(T_0) \neq m\_to\_n\_alive(T_0+1)$: 操作单元切换激活。

情形 1 到 3 是特殊情形, 不是正常情形。

呼叫等待(CW)接口

DB19 DBW100	ONL_REQUEST
客户端 ID	操作单元希望在线, 希望占用呼叫等待接口。HMI 首先写入其客户端 ID, 作为在线请求。 位 8.. 15: 总线类型: MPI1 或 OPI2 位 0.. 7: HMI 总线地址
DB19 DBW102	ONL_CONFIRM
客户端 ID	如果呼叫等待接口没有被另一个操作单元占用, 则作为积极应答 PLC 返回客户端 ID。 位 8.. 15: 总线类型: MPI1 或 OPI2 位 0.. 7: HMI 总线地址
DB19 DBW104	PAR_CLIENT_IDENT 传递给 PLC 的 HMI 参数
客户端 ID	位 8.. 15: 总线类型: MPI1 或 OPI2 位 0.. 7: HMI 总线地址
DB19 DBB106	PAR_MMC_TYP 传递给 PLC 的 HMI 参数
HMI 型号, 来自 NETNAMES.INI	指在文件 NETNAMES.INI 中确定的操作单元的型号。它在 PLC 请求操作单元离线时被分析(服务器、主操作区/副操作区等), 详见文件 NETNAMES.INI 的说明。
DB19 DBB107	PAR_MSTT_ADR 传递给 PLC 的 HMI 参数
OPI 地址, 来自 NETNAMES.INI	要和操作单元一起切换或一起激活/禁用的 OPI 的地址。 来自 NETNAMES.INI 的参数
255	操作单元不含 OPI, 不会有 OPI 和它一起激活/禁用
DB19 DB108	PAR_STATUS PLC 向 HMI 给出积极/消极应答

ONL_PERM (5)	PLC 告知 HMI 操作单元能否在线。该信号的含义取决于 PAR_Z_INFO。
DB19 DBB109	PAR_Z_INFO PLC 向 HMI 给出积极/消极应答
HMI-PLC 在线接口的编号，OK(10)	PLC 为呼叫等待的操作单元给出在线使能。 该操作单元随后和该 NCU 在线连接。 位 0...3:OK 位 4.. 7: HMI-PLC 在线接口的编号，操作单元要和该接口连接： 1 表示第一个 HMI-PLC 在线接口 2 表示第二个 HMI-PLC 在线接口
MMC_LOCKED (13)	“呼叫等待”的操作单元无法在线。该 NCU 上有两个操作单元在线，上面运行了不可中断的进程。PLC 无法使这两个操作单元离线。
PLC_LOCKED (14)	在 HMI-PLC 接口内设置了操作单元切换锁。
PRIO_H (15)	“呼叫等待”的操作单元无法在线。该 NCU 上有两个操作单元在线，其优先级高于“呼叫等待”的操作单元。PLC 无法使这两个操作单元离线。

“操作面板切换”的生命符号

DB19 DBW110	M_TO_N_ALIVE
1 ... 65535	循环计数器，读数由 PLC 循环增加。便于 HMI 判断是否激活了操作面板切换。

第 1 个 HMI/PLC 在线接口

DB19 DBW120	MMC1_CLIENT_IDENT
	见“PAR_CLIENT_IDENT” 在 PLC 给出积极应答后，PLC 将 HMI 参数传递给在线接口 PAR_CLIENT_IDENT -> MMC1_CLIENT_IDENT
DB19 DBB122	MMC1_TYP
	见“PAR_MMC_TYP” 在 PLC 给出积极应答后，PLC 将 HMI 参数传递给在线接口 PAR_MMC_TYP -> MMC1_TYP
DB19 DBB123	MMC1_MSTT_ADR

	见“PAR_MSTT_ADR” 在 PLC 给出积极应答后, PLC 将 HMI 参数传递给在线接口 PAR_MSTT_ADR -> MMC1_MSTT_ADR
DB19 DBB124	MMC1_STATUS
	HMI 向 PLC 请求在线或相反。该信号的含义取决于 MMC1_Z_INFO。
OFFL_REQ_PLC (1)	PLC → HMI: PLC 发出离线请求, 要求操作单元离线。
OFFL_CONF_PLC (2)	HMI → PLC: 对 OFFL_REQ_PLC 的应答
OFFL_REQ_OP (3)	HMI → PLC: 操作单元发出离线请求, 请求和该 NCU 断开
OFFL_CONF_OP (4)	PLC → HMI: 对 OFFL_REQ_OP 的应答
S_ACT (6)	HMI → PLC: 操作单元在线或者切换操作焦点。
OFFL_REQ_FOC (7)	HMI → PLC: 操作单元希望从该 NCU 处取回操作焦点
OFFL_CONF_FOC (8)	PLC → HMI: 对 OFFL_REQ_FOC 的应答
ONL_REQ_FOC (9)	HMI → PLC: 操作单元希望将操作焦点交给该 NCU
ONL_PERM_FOC (10)	PLC → HMI: 对 ONL_REQ_FOC 的应答
DB19 DBB125	MMC1_Z_INFO
	HMI 向 PLC 请求在线或相反。该信号的含义取决于 MMC1_STATUS。
DISC_FOC (9)	操作单元将操作焦点切到另一台 NCU 上。
OK (10)	积极应答
CONNECT (11)	操作单元和该 NCU 在线连接。
PLC_LOCKED (14)	在 HMI-PLC 接口内设置了操作单元切换锁。操作单元无法和该 NCU 在线连接, 也无法切换操作焦点。
PRIO_H (15)	该 NCU 和具有最高优先级的操作单元在线连接。“呼叫等待”的操作单元无法和该 NCU 在线连接。

位信号

DB19 DBX 126.0 数据块	MMC1_SHIFT_LOCK 禁止/激活操作单元切换
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或 输出上升沿 0 → 1	操作单元切换或操作焦点切换被锁定。 当前的操作单元和 NCU 的在线连接不受影响。
信号稳态为 0 或 输出下降沿 1 → 0	操作单元切换或操作焦点切换激活。

DB19 DBX 126.1 数据块	MMC1_MSTT_SHIFT_LOCK 禁止/激活 OPI 切换
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或 输出上升沿 0 → 1	OPI 切换被锁定。 当前的 OPI 和 NCU 的组合不受影响。
信号稳态为 0 或 输出下降沿 1 → 0	OPI 切换激活。

DB19 DBX 126.2 数据块	MMC1_ACTIVE_REQ 操作单元 1 请求进入“主动”操作模式
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或 输出上升沿 0 → 1	HMI → PLC： “被动”操作单元 1 请求进入“主动”操作模式
信号稳态为 0 或 输出下降沿 1 → 0	PLC → HMI： 收到请求

18.2 分布式系统(B3)

DB19 DBX 126.3 数据块	MMC1_ACTIVE_PERM 主动/被动操作模式
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或 输出上升沿 0 → 1	PLC → HMI： 允许“被动”操作单元 1 进入“主动”操作模式
信号稳态为 0 或 输出下降沿 1 → 0	PLC → HMI： “主动”操作单元应进入“被动”操作模式

DB19 DBX 126.4 数据块	MMC1_ACTIVE_CHANGED HMI 的主动/被动操作模式
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或 输出上升沿 0 → 1	HMI → PLC： 操作单元已经从“被动”操作模式进入“主动”操作模式。
信号稳态为 0 或 输出下降沿 1 → 0	HMI → PLC： 操作单元已经从“主动”操作模式进入“被动”操作模式。

DB19 DBX126.5 数据块	MMC1_CHANGE_DENIED 拒绝操作模式切换
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或 输出上升沿 0 → 1	HMI → PLC 或 PLC → HMI，取决于接口的状态： 无法切换操作模式，因为在“主动”操作单元上有不可中断的进行正在运行
信号稳态为 0 或 输出下降沿 1 → 0	HMI → PLC 或 PLC → HMI，取决于接口的状态： 对 MMC1_CHANGE_DENIED 的应答(FALSE → TRUE)

第 2 个 HMI/PLC 在线接口

第 2 个 HMI/PLC 在线接口的各个信号的含义和第 1 个 HMI/PLC 在线接口类似，只是其中 MMC2_ 取代了 MMC1_...

HMI 生命符号监控

在操作单元和 NCU 在线连接后，接口内的 HMI 生命符号置位。(E_BTSSReady, E_MMCMPI_Ready, E_MMC2Ready)

一旦操作单元在线，HMI 就会自动置位这些信号，直到操作单元离线。

每个 HMI/PLC 接口都单独的这些信号，用于 PLC 对 HMI 生命符号的监控。

第 1 个 HMI/PLC 在线接口

此处要区分操作单元是通过 OPI(1.5 MBaud)还是通过 MPI(187.5 kBaud)在线。

一旦操作单元在线，总线类型对应的信号便置位。

DB10 DBX104.0	OPI1 就绪
FALSE	OPI1 尚未就绪
TRUE	OPI1 就绪
DB10 DBX104.1	OPI2 就绪
FALSE	OPI2 尚未就绪
TRUE	OPI2 就绪
DB10 DBX104.2	手动操作设备就绪
FALSE	手动操作设备尚未就绪
TRUE	手动操作设备就绪
DB10 DBX108.3	E_MMCBTSSReady
FALSE	OPI 上没有操作单元在线
TRUE	OPI 上有操作单元在线
DB10 DBX108.2	E_MMCMPIReady
FALSE	MPI 上没有操作单元在线
TRUE	MPI 上有操作单元在线

第 2 个 HMI/PLC 在线接口

该接口上两个总线类型采用一个共用信号。即不区分 OPI 或 MPI。

18.2 分布式系统(B3)

DB10 DBX108.1	E_MMC2Ready
FALSE	OPI 或 MPI 上没有操作单元在线
TRUE	OPI 或 MPI 上有操作单元在线

一旦接口上的一个操作单元在线连接，PLC 就激活生命符号监控，在操作单元离线后，监控关闭。

激活生命符号监控：

一旦操作单元或 HMI 通过 HMI/PLC 接口报告 S_ACT/CONNECT，该监控就激活。

关闭生命符号监控：

一旦操作单元离线，该监控就关闭。

1. HMI 切换，向 PLC 报告 OFFL_REQ_OP/OK
2. PLC 用 OFFL_CONF_OP/OK 应答 HMI
3. PLC 用 OFFL_REQ_PLC/OK 请求操作单元或 HMI 离线
HMI 用 OFFL_CONF_PLC/OK 应答 PLC

在这两种情形中 PLC 会检测操作单元是否离线，并等待其 HMI 生命符号由 TRUE 变为 FALSE。

此后 PLC 不再监控该生命符号。

18.2.3 从 NC 发出的信号(DB10)

DB10 DBX107.6	NCU-Link 生效
脉冲沿计算：	信号更新速率：
信号稳态为 1 或输出上升沿 0 → 1	NCU-Link 通讯生效。
信号稳态为 0 或输出下降沿 1 → 0	没有 NCU-Link 通讯生效。
信号失效条件	系统只有一个 NCU。
更多参考	NCU 7x0.3 PN 手册

18.2.4 从进给轴/主轴发出的信号(DB31, ...)

DB31, ... DBX60.1	NCU-Link 轴生效
脉冲沿计算:	信号更新速率:
信号稳态为 1 或输出上升沿 0 → 1	轴作为 NCU-Link 轴生效。
信号稳态为 0 或输出下降沿 1 → 0	轴用作本地轴。
信号失效条件	系统只有一个 NCU。
更多参考	NCU 7x0.3 PN 手册

DB31, ... DBX61.2	轴运行就绪
脉冲沿计算:	信号更新速率:
含义	该信号由 NCU 组中和轴物理相连的 NCU 控制。
信号稳态为 1 或输出上升沿 0 → 1	轴运行就绪。
信号稳态为 0 或输出下降沿 1 → 0	轴尚未运行就绪。 当通道、运行方式组或 NCK 发出报警“未就绪”时，轴变为该状态。

DB31, ... DBX62.7	轴容器旋转生效
脉冲沿计算:	信号更新速率:
信号稳态为 1 或输出上升沿 0 → 1	该轴所处的轴容器旋转生效。
信号稳态为 0 或输出下降沿 1 → 0	该轴所处的轴容器没有旋转生效。

18.3 手动运行和手轮运行(H1)

18.3.1 从 NC 发出的信号(DB10)

DB10 DBB97, 98, 99	由手轮 1、2、3 移动的几何轴所处通道的编号								
脉冲沿计算：不支持				信号更新速率：周期					
信号含义	用户可直接通过操作面板为轴指定手轮(1, 2, 3)。如果该轴是几何轴（接口信号“机床轴”置 0），NC 会将对应的通道号传送给 PLC。								
	PLC 基本程序随后根据操作员设置的状态（接口信号“手轮已选中”）置位选中几何轴的接口信号“激活手轮”。								
	通道号的编码方式为：								
	位							通道号	
	7	6	5	4	3	2	1		0
	0	0	0	0	0	0	0	0	-
	0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	0	2	
轴是机床轴时（接口信号“机床轴”置 1），接口信号“手轮 1、2、3 移动的几何轴所处通道的编号”没有含义。									
更多信息参见接口信号“由手轮 1、2、3 移动的轴的编号”。									
关联信号	DB10 DBB100 ff（由手轮 1、2、3 移动的轴的编号） DB10 DBX100.6 ff（手轮已选中） DB10 DBX100.7 ff（机床轴） DB21, ... DBX12.0 - 12.2 ff（激活手轮）								
应用举例	如果 DB10 DBB97 = 2，则表明手轮 1 对应通道 2。								

DB10 DBB100, 101, 102 位 0 - 4	由手轮 1、2、3 移动的轴的编号																																																																
脉冲沿计算：不支持		信号更新速率：周期																																																															
信号含义	用户可直接在操作面板上给每根轴指定一个手轮。为此，用户需要指定所需轴(如 X 轴)。该轴编号以及属性（是机床轴还是几何轴，接口信号“机床轴”）经由 PLC 基本程序作为 HMI 接口信号提供。PLC 基本程序随后置位指定轴的接口信号“激活手轮”。取决于 HMI 接口信号“机床轴”的状态，PLC 使用几何轴接口或机床轴接口。																																																																
	轴名和轴号的对应关系为：																																																																
	接口信号“机床轴”置 1，即该轴是机床轴：																																																																
	轴名由以下机床数据指定：																																																																
	MD10000 \$MN_AXCONF_MACHAX_NAME_TAB[n]（机床轴名称）。																																																																
	接口信号“机床轴”置 0，即该轴是几何轴：																																																																
	轴名由以下机床数据指定：																																																																
	MD20060 \$MC_AXCONF_GEOAX_NAME_TAB[n]（通道中几何轴名称）																																																																
	接口信号“由手轮移动的几何轴所处的通道”指出了手轮对应的通道编号。																																																																
	轴号的编码方式为：																																																																
		<table><tr><td colspan="5">位</td><td rowspan="2">轴编号</td></tr><tr><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>—</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>2</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>3</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>4</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>5</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>6</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>7</td></tr></table>					位					轴编号	4	3	2	1	0	0	0	0	0	0	—	0	0	0	0	1	1	0	0	0	1	0	2	0	0	0	1	1	3	0	0	1	0	0	4	0	0	1	0	1	5	0	0	1	1	0	6	0	0	1	1	1	7
	位					轴编号																																																											
4	3	2	1	0																																																													
0	0	0	0	0	—																																																												
0	0	0	0	1	1																																																												
0	0	0	1	0	2																																																												
0	0	0	1	1	3																																																												
0	0	1	0	0	4																																																												
0	0	1	0	1	5																																																												
0	0	1	1	0	6																																																												
0	0	1	1	1	7																																																												

18.3 手动运行和手轮运行(H1)

DB10 DBB100, 101, 102 位 0 - 4	由手轮 1、2、3 移动的轴的编号							
		0	1	0	0	0	8	
关联信号	DB10 DBX97 ff (由手轮移动的几何轴所处通道的编号) DB10 DBX100.6 ff (手轮已选中) DB10 DBX100.7 ff (机床轴) DB21, ... DBX12.0 到 DBX12.2 ff (激活手轮) DB31, ... DBX4.0 bis DBX4.2 (激活手轮) MD10000 \$MN_AXCONF_MACHAX_NAME_TAB[n] (机床轴名称) MD20060 \$MC_AXCONF_GEOAX_NAME_TAB[n] (通道中的几何轴名称)							

DB10 DBX100.6, 101.6, 102.6		手轮已选中（手轮 1、2、3）	
脉冲沿计算：不支持		信号更新速率：周期	
信号稳态为 1 或输出上升沿 0 → 1	操作人员操作面板上选中了指定轴的手轮，即激活了手轮。该信息经由 PLC 基本程序传送到 HMI 接口上。 PLC 基本程序随后将指定轴的接口信号 DB21, ... DBX12.0-12.2 ff（激活手轮）置 1。 轴属性和编号同样显示在 HMI 接口上： DB10 DBX100.7 ff（机床轴） 和 DB10 DBB100 ff（由手轮 1 移动的轴的编号）。 一旦手轮激活，用户便可在 JOG 方式下用手轮移动轴（DB21, ... DBX40.0-40.2 ff（手轮生效）。		

DB10 DBX100.6, 101.6, 102.6	手轮已选中（手轮 1、2、3）
信号稳态为 0 或输出下降沿 1 → 0	操作员在操作面板上将指定轴的手轮设为“无效”。该信息经由 PLC 基本程序传送到 HMI 接口上。 PLC 基本程序随后将指定轴的接口信号 DB21, ... DBX12.0-12.2 ff（激活手轮）置 0。
关联信号	DB10 DBB100 ff（轴号） DB10 DBX100.7 ff（机床轴） DB21, ... DBX12.0-12.2 ff（激活手轮） DB21, ... DBX40.0 - DBX40.2 ff（手轮生效） DB31, ... DBX4.0 - DBX4.2（激活手轮） DB10 DBB97 ff（由手轮 1、2、3 移动的几何轴所处通道的编号）

DB10 DBX100.7, 101.7, 102.7	机床轴（手轮 1、2、3）
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	用户已直接通过操作面板为某根轴指定了手轮(1, 2, 3)。该轴是机床轴。 更多信息参见接口信号“轴号”。
信号稳态为 0 或输出下降沿 1 → 0	用户已直接通过操作面板为某根轴指定了手轮(1, 2, 3)。该轴是几何轴。 更多信息参见接口信号“轴号”。
关联信号	DB10 DBB100 ff（轴号） DB10 DBX100.6 ff（手轮已选中） DB10 DBB97 ff（由手轮 1、2、3 移动的几何轴所处通道的编号）

18.3 手动运行和手轮运行(H1)

18.3.2 发送至通道的信号 (DB21, ...)

发送到通道内某 NCK 的信号的说明

DB21, ... DBX0.3	激活 DRF
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1	请求激活 DRF 功能。 通过 DRF 功能可以在 AUTO 和 MDI 方式下用手轮来修改 DRF 偏移。
信号稳态为 0	不请求 DRF 功能。
信号失效条件	JOG 运行方式
关联信号	DB21, ... DBX24.3 (DRF 已选中)

DB21, ... DBX12.0-2, DBX16.0-2, DBX20.0-2	几何轴的手轮(1, 2, 3)				
脉冲沿计算：不支持		信号更新速率：周期			
信号不为 0	请求激活几何轴的对应手轮。				
	该接口可以位编码或二进制编码方式表示。 表示方式由以下机床数据确定： MD11324 \$MN_HANDWH_VDI_REPRESENTATION				
	位编码: 最多 3 个手轮				
	注				
	一个时间点上一根轴只能有一个手轮。 如果同时有多个接口信号置位，则优先级从高到低依次如下： 手轮 1、手轮 2、手轮 3				
		手轮的编号	位 2	位 1	位 0
		1	0	0	1
2		0	1	0	
3		1	0	0	
二进制编码： 最多 6 个手轮					
	手轮的编号	位 2	位 1	位 0	
	1	0	0	1	
	2	0	1	0	
	3	0	1	1	

18.3 手动运行和手轮运行(H1)

DB21, ... DBX12.0-2, DBX16.0-2, DBX20.0-2	几何轴的手轮(1, 2, 3)				
		4	1	0	0
		5	1	0	1
		6	1	1	0
信号稳态为 0	不请求激活手轮。				
关联信号	DB21, ... DBX40.6-7 ff (几何轴的手轮生效)				

DB21, ... DBX12.4, DBX16.4, DBX20.4	几何轴(1, 2, 3)的移动键禁用	
脉冲沿计算：不支持		信号更新速率：周期
信号稳态为 1	不考虑通过几何轴的正向/负向移动键发出的运行请求。 如果在轴运行过程中激活了移动键锁定，则运行终止。	
信号稳态为 0	正向/负向移动键有效。	
应用举例	利用该信号，PLC 用户程序可以根据运行状态禁止在 JOG 方式中通过移动键移动几何轴。	
关联信号	DB21, ... DBX12.6-7 ff（几何轴的正向/负向移动键）	

18.3 手动运行和手轮运行(H1)

DB21, ... DBX12.5, DBX16.5, DBX20.5	几何轴的快速移动速度叠加(1, 2, 3)	
脉冲沿计算：不支持		信号更新速率：周期
信号稳态为 1	发出运行请求，移动轴。 ● DB21, ... DBX12.7 == 1（正向移动键） ● DB21, ... DBX12.6 == 1（负向移动键） 该接口信号置位后，后，几何轴以快速移动速度运行。 快速移动速度在以下机床数据中定义： MD32010 \$MA_JOG_VELO_RAPID（基准 JOG 快速移动速度） 在 JOG 的以下方式中快速移动速度叠加有效： ● 连续运行 ● 断续运行(INC1, INC10, ...) 快速移动速度可用快速倍率开关加以修调。	
信号稳态为 0	几何轴以指定的 JOG 速度移动： SD41110 \$SN_JOG_SET_VELO（JOG 轴速度） 或 MD32020 \$MA_JOG_VELO（基准 JOG 轴速度）。	
信号失效条件	● 运行方式 AUTO 和 MDI ● 回参考点(JOG 方式)	
关联信号	DB21, ... DBX12.6-7 ff（几何轴的正向/负向移动键）	
更多参考	功能手册之基本功能分册；章节“进给率（V1）”	

DB21, ... DBX12.6-7, DBX16.6-7, DBX20.6-7	几何轴(1, 2, 3)的正向/负向移动键	
脉冲沿计算：不支持		信号更新速率：周期
信号稳态为 1	选中的几何轴可以在 JOG 方式下通过正向/负向移动键朝两个方向移动。 取决于当前生效的机床功能及“点按或是长按”设置： SD41050 \$SN_JOG_CONT_MODE_LEVELTRIGGRD（JOG 连续运行中的点按/长按模式） 和 MD11300 \$MN_JOG_INC_MODE_LEVELTRIGGRD（JOG 断续运行和回参考点采用长按模式） 系统在信号状态切换时作出不同的响应。 - 连续运行采用长按模式 在该接口信号置 1 后，几何轴就一直朝对应方向移动，直到达某个有效限位。 - 连续运行中的点按模式 在该信号发出首个上升沿 0 → 1 后，几何轴开始朝对应方向移动。即使之后信号输出下降沿 1 → 0，轴也继续移动。在该信号再次给出上升沿 0 → 1 后（同一个移动方向！），轴终止移动。 - 断续运行采用长按模式 该信号置 1 后，几何轴移动指定的步数。如果在没有移动完指定步数前，该信号就置 0，则轴中断移动。信号再次置 1 后，轴继续移动。在没有移动完指定步数前，轴的移动可以如上文所述暂停和继续。 - 断续运行采用点按模式 在该信号输出第一个上升沿 0 → 1 后，几何轴开始移动指定的步数。如果在几何轴移动完指定的步数前，同一个方向信号再次输出上升沿 0 → 1，则几何轴终止移动。轴因此不再移动余下的行程。 提示 - 如果两个方向信号同时置位，几何轴会不移动或终止当前移动。 - 和机床轴不同的是，在一个时间点上只能通过移动键移动一根几何轴。 - DBX12.4 = 1 ff.（移动键禁用）可以禁用移动键。	
信号稳态为 0	见“情形 1 到 4”。	
信号失效条件	运行方式 AUTO 和 MDI	

18.3 手动运行和手轮运行(H1)

DB21, ... DBX12.6-7, DBX16.6-7, DBX20.6-7	几何轴(1, 2, 3)的正向/负向移动键
特殊情况, 出错, ...	几何轴不能在 JOG 方式下移动: ● 当前已经通过轴 PLC 接口信号作为机床轴移动。 ● 当前正有另一根几何轴被移动键移动。 此时系统会输出报警 20062“轴已激活”。
关联信号	DB31, ... DBX8.7 以及 DBX8.6 (机床轴的正向/负向移动键) DB21, ... DBX12.4 ff (几何轴的移动键禁用)

DB21, ... DBX13.0-5, DBX17.0-5, DBX21.0-5	请求几何轴(1, 2, 3)的“断续移动”机床功能
脉冲沿计算: 不支持	信号更新速率: 周期
信号稳态为 1	请求几何轴在 JOG 方式下的“断续移动”机床功能: ● 位 0 = INC1 ● 位 1 = INC10 ● 位 2 = INC100 ● 位 3 = INC1000 ● 位 4 = INC10000 ● 位 5 = INCvar “步”是按下一次正向/负向移动键或者手轮转动一个刻度轴移动的最小单位。确定步数的数据为 ● INC1 到 INC10000: MD11330 \$MN_JOG_INCR_SIZE_TAB (手动运行/手轮运行中的步数) ● INCvar: SD41010 \$SN_JOG_VAR_INCR_SIZE (JOG 方式下的可变步数) 注 同时发出多个功能请求时, 没有任何机床功能生效。
信号稳态为 0	未请求某个机床功能。 如果在几何轴当前正在执行某机床功能时取消该功能或者切换到另一个功能, 则几何轴终止移动。
关联信号	DB21, ... DBB41 ff (几何轴的生效机床功能: INC1, ...) DB21, ... DBX13 ff (几何轴的机床功能: 连续移动)

DB21, ... DBX13.6, DBX17.6, DBX21.6	请求几何轴(1, 2, 3)的“连续移动”机床功能		
脉冲沿计算：不支持		信号更新速率：周期	
信号稳态为 1	请求通过正向/负向移动键在 JOG 方式下“连续移动”的机床功能。		
信号稳态为 0	不请求“连续移动”机床功能。		
关联信号	DB21, ... DBB41.6 ff（连续移动生效）		

DB21, ... DBX15.0, DBX 19.0, DBX 23.0	取反几何轴(1, 2, 3)的手轮的旋转方向		
脉冲沿计算：不支持		信号更新速率：周期	
信号稳态为 1	请求取反手轮的旋转方向。 只允许在几何轴处于静止时修改该取反信号。		
信号稳态为 0	不取反几何轴 1、2、3 的手轮的旋转方向。		
应用举例	- 手轮旋转方向和预期的轴移动方向不一致。 - 一个手轮(HT2, HT8)指定给了多根轴。		
关联信号	DB21, ... DBX43.0, 49.0, 55.0（对几何轴 1、2、3 的手轮旋转方向的取反生效）		

DB21, ... DBX30.0-2		激活轮廓手轮			
脉冲沿计算：不支持		信号更新速率：周期			
信号稳态为 1		请求为功能“轮廓手轮/手轮设定行程”激活对应的手轮。			
		该接口可以位编码或二进制编码方式表示。表示方式用以下机床数据确定：			
		MD11324 \$MN_HANDWH_VDI_REPRESENTATION			
		位编码: 最多 3 个手轮			
		注			
		一个时间点上一根轴只能有一个手轮。如果同时有多个接口信号置位，则优先级从高到低依次如下：手轮 1、手轮 2、手轮 3			
	手轮的编号		位 2	位 1	位 0
	1		0	0	1
	2		0	1	0

18.3 手动运行和手轮运行(H1)

DB21, ... DBX30.0-2	激活轮廓手轮				
	3	1	0	0	
	二进制编码: 最多 6 个手轮				
	手轮的编号	位 2	位 1	位 0	
	1	0	0	1	
	2	0	1	0	
	3	0	1	1	
	4	1	0	0	
	5	1	0	1	
	6	1	1	0	
信号稳态为 0	不请求激活手轮。				
关联信号	DB21, ... DBX37.0-2 (轮廓手轮生效)				

DB21, ... DBX320.0-2, DBX324.0-2, DBX328.0-2	激活定向轴的手轮(1, 2, 3)				
脉冲沿计算：不支持		信号更新速率：周期			
信号稳态为 1	请求激活定向轴的对应手轮。				
	该接口可以位编码或二进制编码方式表示。 表示方式用以下机床数据确定： MD11324 \$MN_HANDWH_VDI_REPRESENTATION				
	位编码: 最多 3 个手轮				
	注				
	一个时间点上一根轴只能有一个手轮。 如果同时有多个接口信号置位，则优先级从高到低依次如下： 手轮 1、手轮 2、手轮 3				
		手轮的编号	位 2	位 1	位 0
		1	0	0	1
	2	0	1	0	
	3	1	0	0	
二进制编码： 最多 6 个手轮					
	手轮的编号	位 2	位 1	位 0	
	1	0	0	1	
	2	0	1	0	

18.3 手动运行和手轮运行(H1)

DB21, ... DBX320.0-2, DBX324.0-2, DBX328.0-2	激活定向轴的手轮(1, 2, 3)				
		3	0	1	1
		4	1	0	0
		5	1	0	1
		6	1	1	0
信号稳态为 0	不请求激活手轮。				
关联信号	DB21, ... DBX332.Bit 0-2 ff. (激活定向轴(1, 2, 3)的手轮)				

DB21, ... DBX323.0, DBX327.0, DBX331.0	取反定向轴(1, 2, 3)的手轮的旋转方向	
脉冲沿计算：不支持		信号更新速率：周期
信号稳态为 1	请求取反定向轴 1、2 或 3 的手轮的旋转方向。 注 只允许在轴静止状态下修改该取反信号。	
信号稳态为 0	不请求取反。	
应用举例	- 手轮的旋转方向应和轴的运动方向一致。 - 手轮指定给了多根定向轴。	
关联信号	DB21, ... DBX335.0 ff.（对定向轴(1, 2, 3)的手轮旋转方向的取反生效）	

18.3.3 从通道发出的信号 (DB21, ...)

从通道发送到 PLC 的信号说明

DB21, ... DBX24.3	DRF 已选择	
脉冲沿计算：不支持		信号更新速率：周期
信号稳态为 1	DRF 功能生效。	
信号稳态为 0	DRF 功能不生效。	

18.3 手动运行和手轮运行(H1)

DB21, ... DBX24.3	DRF 已选择
信号失效条件	JOG 运行方式
关联信号	DB21, ... DBX0.3 (激活 DRF)

DB21, ... DBX33.3	手轮叠加生效
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1	功能“自动方式中的手轮叠加”对程序写入的轨迹轴生效。几何轴 1 的手轮发出的脉冲作为“速度”叠加在程序写入的轨迹进给率上。 在下面的条件下该叠加失效： ● 轨迹轴到达程序写入的终点 ● 删除了剩余行程：DB21, ... DBX6.2 == 1 (删除剩余行程) ● 触发了复位
信号稳态为 0	功能“自动方式中的手轮叠加”不生效。

DB21, ... DBX37.0-2		轮廓手轮生效																		
脉冲沿计算：不支持		信号更新速率：周期																		
信号稳态为 1		报告哪个手轮激活了功能“轮廓手轮/手轮设定行程”。																		
		该接口可以位编码或二进制编码方式表示。表示方式用以下机床数据确定： MD11324 \$MN_HANDWH_VDI_REPRESENTATION																		
		位编码: 最多 3 个手轮																		
		注																		
		一个时间点上一根轴只能有一个手轮。如果同时有多个接口信号置位，则优先级从高到低依次如下：手轮 1、手轮 2、手轮 3																		
		<table><tr><td rowspan="4"></td><td>手轮的编号</td><td>位 2</td><td>位 1</td><td>位 0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>2</td><td>0</td><td>1</td><td>0</td></tr><tr><td>3</td><td>1</td><td>0</td><td>0</td></tr></table>					手轮的编号	位 2	位 1	位 0	1	0	0	1	2	0	1	0	3	1
	手轮的编号	位 2	位 1	位 0																
	1	0	0	1																
	2	0	1	0																
	3	1	0	0																
		二进制编码：最多 6 个手轮																		
		<table><tr><td rowspan="3"></td><td>手轮的编号</td><td>位 2</td><td>位 1</td><td>位 0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>2</td><td>0</td><td>1</td><td>0</td></tr></table>					手轮的编号	位 2	位 1	位 0	1	0	0	1	2	0	1	0		
			手轮的编号	位 2	位 1		位 0													
1	0		0	1																
2	0		1	0																

DB21, ... DBX37.0-2	轮廓手轮生效				
		3	0	1	1
		4	1	0	0
		5	1	0	1
		6	1	1	0
信号稳态为 0	功能“轮廓手轮/手轮设定行程”无指定手轮。				
关联信号	DB21, ... DBX30.0-2（轮廓手轮指定）				

DB21, ... DBX40.0-2, DBX46.0-2, DBX52.0-2	生效的几何轴手轮(1, 2, 3)				
脉冲沿计算：不支持		信号更新速率：周期			
信号稳态为 1	报告哪个几何轴手轮生效。				
	该接口可以位编码或二进制编码方式表示。表示方式用以下机床数据确定： MD11324 \$MN_HANDWH_VDI_REPRESENTATION				
	位编码: 最多 3 个手轮				
	注				
	一个时间点上一根轴只能有一个手轮。如果同时有多个接口信号置位，则优先级从高到低依次如下：手轮 1、手轮 2、手轮 3				
		手轮的编号	位 2	位 1	位 0
		1	0	0	1
		2	0	1	0
	3	1	0	0	
二进制编码: 最多 6 个手轮					
	手轮的编号	位 2	位 1	位 0	
	1	0	0	1	
	2	0	1	0	
	3	0	1	1	
	4	1	0	0	
	5	1	0	1	
	6	1	1	0	

18.3 手动运行和手轮运行(H1)

DB21, ... DBX40.0-2, DBX46.0-2, DBX52.0-2	生效的几何轴手轮(1, 2, 3)
信号稳态为 0	没有任何几何轴手轮生效。
关联信号	DB21, ... DBX12.0-2 ff (激活手轮)

DB21, ... DBX40.4-5, DBX46.4-5, DBX52.4-5	几何轴(1, 2, 3)的正向/负向运行请求
脉冲沿计算: 不支持 信号更新速率: 周期	
信号稳态为 1	几何轴当前有朝某个方向的运行请求。 - 位 4 = 负向运行请求 - 位 5 = 正向运行请求 运行方式: - 运行方式 JOG: 正向或负向移动键 - 运行方式 REF: 触发轴朝参考点运行的移动键 - 运行方式 AUTO 和 MDI: 执行一个包含几何轴运行指令的程序段。
信号稳态为 0	几何轴当前没有运行请求。
关联信号	DB21, ... DBX40.7 以及 DBX40.6 DB21, ... DBX46.7 以及 DBX46.6 DB21, ... DBX52.7 以及 DBX52.6 (正向/负向运行指令)

DB21, ... DBX40.7-6, DBX46.7-6, DBX52.7-6	几何轴(1, 2, 3)的正向/负向运行指令				
脉冲沿计算: 不支持 信号更新速率: 周期					
描述	运行指令的输出取决于: MD17900 \$MN_VDI_FUNCTION_MASK 位 0 <table> <tr> <td>位 0</td><td>含义</td></tr> <tr> <td>0</td><td>当出现一个运行请求时, 就输出运行指令。</td></tr> </table>	位 0	含义	0	当出现一个运行请求时, 就输出运行指令。
位 0	含义				
0	当出现一个运行请求时, 就输出运行指令。				

DB21, ... DBX40.7-6, DBX46.7-6, DBX52.7-6	几何轴(1, 2, 3)的正向/负向运行指令	
	1	只有在轴移动时才输出运行指令。
信号稳态为 1	请求朝对应方向移动几何轴。 - 位 6 = 负向运行指令 - 位 7 = 正向运行指令 系统对运行指令的响应取决于当前的运行方式： - 运行方式 JOG： 系统使轴朝正向或负向移动。 - 运行方式 REF： 系统使轴只朝参考点移动。 - 运行方式 AUTO 和 MDI： 执行一个包含轴位置的程序段。	
信号稳态为 0	几何轴当前没有运行请求。	
应用举例	一旦检测到运行指令就松开轴，使轴运行。 注 此类只有检测到运行指令才松开的轴不支持连续路径方式 (G64)。	
关联信号	DB21, ... DBX12.7 以及 DBX12.6 ff（几何轴的正向/负向移动键） DB21, ... DBX 40, 46, 52 位 5（正向/负向移动请求）	

DB21, ... DBX41.0-6, DBX47.0-6, DBX53.0-6	几何轴(1, 2, 3)的机床功能 INC1, INC10, INC100, INC1000, INC10000, INCvar, 连续移动	
脉冲沿计算：不支持		信号更新速率：周期
信号稳态为 1	某机床功能激活。 - 位 0 = 1 INC - 位 1 = 10 INC - 位 2 = 100 INC - 位 3 = 1000 INC - 位 4 = 10000 INC - 位 5 = 可变 INC - 位 6 = 连续移动 生效的移动步数不同，在按下移动轴或旋转手轮时，系统的响应各不相同。	

18.3 手动运行和手轮运行(H1)

DB21, ... DBX41.0-6, DBX47.0-6, DBX53.0-6	几何轴(1, 2, 3)的机床功能 INC1, INC10, INC100, INC1000, INC10000, INCvar, 连续移动
信号稳态为 0	相应的机床功能无效。
关联信号	DB21, ... DBB13 位 0-5 ff (几何轴的机床功能: INC1, ...) DB21, ... DBB13 位 6 ff (几何轴的机床功能: 连续移动)

DB21, ... DBX43.0, 49.0, 55.0	对几何轴(1, 2, 3)的手轮旋转方向的取反生效
脉冲沿计算: 不支持	信号更新速率: 周期
信号稳态为 1	对几何轴的手轮旋转方向的取反生效。
信号稳态为 0	对几何轴的手轮旋转方向的取反不生效。
关联信号	DB31, ... DBX15.0, DBX19.0, DBX23.0 (取反几何轴的手轮旋转方向)

DB21, ... DBX332.0-2, DBX336.0-2, DBX340.0-2	定向轴(1, 2, 3)的手轮生效			
脉冲沿计算：不支持		信号更新速率：周期		
信号稳态为 1	报告哪个几何轴的手轮生效。			
	该接口可以位编码或二进制编码方式表示。 表示方式用以下机床数据确定： MD11324 \$MN_HANDWH_VDI_REPRESENTATION			
	位编码: 最多 3 个手轮			
	注			
	一个时间点上一根轴只能有一个手轮。 如果同时有多个接口信号置位，则优先级从高到低依次如下： 手轮 1、手轮 2、手轮 3			
		手轮的编号	位 2	位 1
	1	0	0	1
	2	0	1	0
	3	1	0	0
二进制编码: 最多 6 个手轮				
	手轮的编号	位 2	位 1	位 0
	1	0	0	1

18.3 手动运行和手轮运行(H1)

DB21, ... DBX332.0-2, DBX336.0-2, DBX340.0-2	定向轴(1, 2, 3)的手轮生效				
		2	0	1	0
		3	0	1	1
		4	1	0	0
		5	1	0	1
		6	1	1	0
信号稳态为 0	没有任何定向轴的手轮生效。				
关联信号	DB21, ... DBX332.0-2 ff (激活手轮)				

DB21, ... DBX332.4-5, DBX336.4-5, DBX340.4-5	定向轴(1, 2, 3)的正向/负向运行请求	
脉冲沿计算：不支持		信号更新速率：周期
该信号和之前的“运行指令”信号相同。		
信号稳态为 1	- 运行方式 JOG： 按下正向/负向移动键。 - 运行方式 REF： 按下使轴朝参考点移动的移动键。 - 运行方式 AUTO 和 MDI： 执行一段包含轴坐标值的程序段。	
信号稳态为 0	在对应轴方向上没有任何运行请求，或者运行已结束。 - 运行方式 JOG： 取决于“点按或是长按”的设置，运行指令复位，另见 DBX12.7 或 DBX12.6 ff。 使用手轮运行。 - 运行方式 REF： 达到参考点。 - 运行方式 AUTO 和 MDI： 执行一条程序段，之后的程序段不包含对应轴的坐标值。 通过“复位”终止，等等 接口信号 DB21, ... DBX25.7（轴禁止）有待处理。	
关联信号	DB31, ... DBX332.7 以及 DBX332.6 DB31, ... DBX336.7 以及 DBX336.6 DB31, ... DBX340.7 以及 DBX340.6（正向/负向运行指令）	

18.3 手动运行和手轮运行(H1)

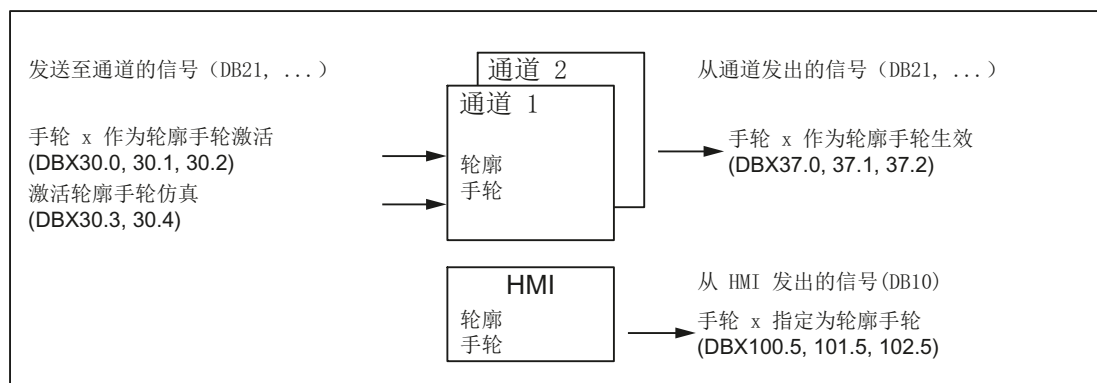
DB21, ... DBX332.6-7, DBX336.6-7, DBX340.6-7	定向轴(1, 2, 3)的正向/负向运行指令
脉冲沿计算：不支持	信号更新速率：周期
机床数据设置不同，该接口信号的作用有所不同： • MD17900 \$MN_VDI_FUNCTION_MASK, 位 0 == 0 作用如下文所述 • MD17900 \$MN_VDI_FUNCTION_MASK 位 0 == 1 信号稳态为 1：只有在几何轴真正移动时。 MD17900 位 0 = 0 时，输出的接口信号 DB21, ... DBX 332, 336, 340 位 5 和位 4（正向/负向移动请求）的作用 和信号“正向/负向移动指令”一样。	
信号稳态为 1	轴朝指定方向移动。运行方式不同，运行指令的触发方式也有所不同： • 运行方式 JOG： 按下正向/负向移动键。 • 运行方式 REF： 按下使轴朝参考点移动的移动键。 • 运行方式 AUTO 和 MDI： 执行一段包含轴坐标值的程序段。
信号稳态为 0	在对应轴方向上没有任何运行请求，或者运行已结束。 • 运行方式 JOG： 取决于“点按或是长按”的设置，运行指令复位，另见“DB21, ... DBX12.7 或 DBX12.6 ff”。 • 运行方式 REF： 达到参考点。 • 运行方式 AUTO 和 MDI： 执行一条程序段，之后的程序段不包含对应轴的坐标值。 通过“复位”终止，等等 接口信号 DB21, ... DBX25.7（轴禁止）有待处理。
应用举例	当轴被夹紧时松开轴夹紧装置，比如：回转工作台上。 注： 如果只有在发出运行指令时轴才被松开，则这种轴不支持轨迹运行！
关联信号	DB21, ... DBX12.7 以及 DBX12.6 ff （几何轴的正向/负向移动键） DB21, ... DBX 332, 336, 340 位 5 和 4（正向/负向移动请求）

DB21, ... DBX377.4	手动退刀生效	
脉冲沿计算：不支持	信号更新速率：周期	
信号稳态为 1	手动退刀被选中且生效。	
信号稳态为 0	手动退刀未被选中。	
关联信号	DB21, ... DBX377.5（手动退刀：退刀数据存在）	

DB21, ... DBX377.5	退刀数据存在	
脉冲沿计算：不支持	信号更新速率：周期	
信号稳态为 1	手动退刀：存在沿刀具中心轴方向退刀的数据。可以选择“手动退刀”（通过操作界面或者 PI 服务“RETRAC”）	
信号稳态为 0	手动退刀：不存在沿刀具中心轴方向退刀的数据。手动退刀无法被选中。	
关联信号	DB21, ... DBX377.4（手动退刀生效）	

18.3.4 轮廓手轮相关信号

轮廓手轮相关信号一览



18.3 手动运行和手轮运行(H1)

轮廓手轮相关信号的说明

DB10 DBX100.5 DBX101.5 DBX102.5	手轮 1 指定为轮廓手轮 手轮 2 指定为轮廓手轮 手轮 3 指定为轮廓手轮	
脉冲沿计算：不支持		信号更新速率：周期
描述	这些信号指出了通过 HMI 将哪个手轮指定为轮廓手轮：	
	1 信号	手轮 x 通过 HMI 指定为轮廓手轮
	0 信号	手轮 x 未通过 HMI 指定为轮廓手轮
	为使通过 HMI 指定为轮廓手轮的手轮生效，该信号必须和对应的接口信号 DB21, ... DBX30.0, 30.1, 30.2 （作为轮廓手轮激活手轮 x）互联在一起。	
特殊情况， 出错，...	取决于 PLC 基本程序中 FB1 参数 HWheelMMC 的设置，这些信号由基本程序提供或者必须由 PLC 用户程序提供。	
关联信号	DB21 ... DBX30.0, 30.1, 30.2（激活手轮 x 用作轮廓手轮） FB1 参数 HWheelMMC	

DB21, ... DBX30.0 DBX30.1 DBX30.2	激活手轮 1 用作轮廓手轮 激活手轮 2 用作轮廓手轮 激活手轮 3 用作轮廓手轮	
脉冲沿计算：不支持		信号更新速率：周期
描述	通过这些信号可以选定/撤销其中某个手轮用作轮廓手轮。	
	1 信号	手轮 x 选作轮廓手轮
	0 信号	手轮 x 不选作轮廓手轮
	轮廓手轮的激活/禁用在程序段中编写。 在激活轮廓手轮时，系统会首先制动轴，然后使轴根据轮廓手轮移动。 在禁用轮廓手轮时，系统会首先制动轴，然后直接继续执行 NC 程序。如果希望 NC 程序在重新给出“NC 启动”信号后才继续执行，则必须在 PLC 用户程序中将该信号和“NC 停止”信号互联在一起。	
特殊情况， 出错，...	NC 复位后，该信号仍保持不变。	
关联信号	DB21, ... DBX37.0, 37.1, 37.2（手轮 x 作为轮廓手轮激活）	

DB21, ... DBX30.3 DBX30.4	轮廓手轮仿真激活 轮廓手轮负向仿真		
脉冲沿计算： 不支持		信号更新速率： 周期	
描述	设置以下信号，以激活/取消轮廓手轮仿真和指定运行方向：		
	位 3	位 4	含义
	0	0	取消仿真
	0	1	取消仿真
	1	0	激活方针，方向如程序
	1	1	激活方针，方向和程序相反
	在仿真中，进给率不再由轮廓手轮给出，而是来自程序。 取消仿真功能后，轴会沿制动斜坡制动。 在切换运行方向时，轴会沿制动斜坡制动，然后在反方向上运行。		
特殊情况， 出错，...	仿真只在 AUTO 运行方式下有效，只能在轮廓手轮激活后激活。		

DB21, ... DBX31.5	取反轮廓手轮的旋转方向		
脉冲沿计算： 不支持		信号更新速率： 周期	
描述	置位该 PLC 接口信号可以取反一个轮廓手轮的旋转方向。		
应用举例	- 手轮旋转方向和预期的轴移动方向不一致。 - 一个手轮(HT2, HT8)指定给了多根轴。		
特殊情况， 出错， ...	只允许在静止状态下修改取反信号。		
关联信号	DB31, ... DBX39.5.（轮廓手轮旋转方向取反生效）		

DB21, ... DBX37.0 DBX37.1 DBX37.2	手轮 1 作为轮廓手轮生效 手轮 2 作为轮廓手轮生效 手轮 3 作为轮廓手轮生效	
脉冲沿计算：不支持		信号更新速率： 周期
描述	这些信号指出了哪个手轮被选定为轮廓手轮：	
	1 信号	手轮 x 选作轮廓手轮
	0 信号	手轮 x 不选作轮廓手轮

18.3 手动运行和手轮运行(H1)

DB21, ... DBX37.0 DBX37.1 DBX37.2	手轮 1 作为轮廓手轮生效 手轮 2 作为轮廓手轮生效 手轮 3 作为轮廓手轮生效
特殊情况, 出错, ...	NC 复位后, 该信号仍保持不变。
关联信号	DB21, ... DBX30.0, 30.1, 30.2 (作为轮廓手轮激活手轮 x)

DB21, ... DBX39.5	轮廓手轮旋转方向取反生效
脉冲沿计算: 不支持	信号更新速率: 周期
描述	该信号可指出轮廓手轮的旋转方向是否已取反:
1 信号	轮廓手轮的旋转方向已取反。
0 信号	轮廓手轮的旋转方向未取反。
关联信号	DB31, ... DBX31.5. (取反轮廓手轮旋转方向)

18.3.5 发送到进给轴/主轴的信号(DB31, ...)

到达进给轴/主轴的信号描述

DB31, ... DBB4 Bit 0-2	激活手轮 (1 至 3)
脉冲沿计算: 不支持	信号更新速率: 周期
信号稳态为 1 或输出上升沿 0 → 1	这些 PLC 接口信号确定是否为该机床轴指定手轮 1、2 或 3, 或者根本不指定手轮。一个时间点上一根轴只能有一个手轮。 如多个接口信号 DB31, ... DBX4.0, 4.1, 4.2 (激活手轮) 置位, 则优先级从高到低依次为: 手轮 1、手轮 2、手轮 3。 在手轮指定有效时, 在 JOG 方式下可用手轮移动机床轴; 在 AUTO 或 MDI 方式下可用手轮产生 DRF 偏移。
信号稳态为 0 或输出下降沿 1 → 0	该机床轴没有手轮 1、2 或 3。

DB31, ... DBB4 Bit 0-2	激活手轮（1 至 3）
应用举例	通过该接口信号 PLC 用户程序可以阻止手轮旋转对轴的影响。
关联信号	DB31, ... DBX64.0 到 DBX64.2（手轮生效）

DB31, ... DBX4.4	移动键禁用
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	正向和负向移动键对对应的机床轴不起作用。比如：在 JOG 方式下便无法通过操作面板上的移动键使轴移动。 如果在轴移动期间激活了“移动键禁用”，则机床轴停止。
信号稳态为 0 或输出下降沿 1 → 0	正向/负向移动键有效。
应用举例	利用该信号，PLC 用户程序可以根据运行状态禁止在 JOG 方式中通过移动键移动机床轴。
关联信号	DB31, ... DBX4.7 或 DBX4.6（正向/负向移动键）

DB31, ... DBX4.5	快速移动速度叠加
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	如果接口信号 DB31, ... DBX4.0, 4.1, 4.2（正向/负向移动键） 和 PLC 接口信号 DB31, ... DBX4.5（快速移动速度） 一起给出，则对应的机床轴以快速移动速度移动。 快速移动速度在机床数据 MD32010 \$MA_JOG_VELO_RAPID（基准 JOG 快速移动速度） 中定义。 快速移动速度叠加在以下 JOG 方式下有效： ● 连续运行 ● 断续运行 在快速移动速度叠加生效时，可通过进给倍率修调开关修改速度。

18.3 手动运行和手轮运行(H1)

DB31, ... DBX4.5	快速移动速度叠加
信号稳态为 0 或输出下降沿 1 → 0	机床轴以指定的 JOG 速度移动: SD41110 \$SN_JOG_SET_VELO (JOG 轴速度) 或 MD32020 \$MA_JOG_VELO (基准 JOG 轴速度)。
信号失效条件	运行方式 AUTO 和 MDI 回参考点(JOG 方式)
关联信号	DB31, ... DBX4.7 或 DBX4.6 (正向/负向移动键) DB31, ... DBB0 (进给倍率/主轴倍率)

DB31, ... DBB4 位 7 和位 6	正向/负向移动键
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1	选中的机床轴可以在 JOG 方式下通过正向/负向移动键朝两个方向移动。 取决于当前激活的机床功能以及设置： • JOG 连续运行 SD41050 \$SN_JOG_CONT_MODE_LEVELTRIGGRD（JOG 连续运行中的点按/长按模式） • JOG 断续运行（JOG 断续运行和回参考点采用长按模式） MD11300 \$MN_JOG_INC_MODE_LEVELTRIGGRD 系统在信号状态切换时作出不同的响应。 情形 1：连续运行采用长按模式 在该接口信号置 1 后，机床轴就一直朝对应方向移动，直到达某个有效限位。 情形 2：连续运行采用点按模式 在该信号发出首个上升沿 0 → 1 后，机床轴开始朝对应方向移动。即使之后信号输出下降沿 1 → 0，轴也继续移动。在该信号再次给出上升沿 0 → 1 后（同一个移动方向！），轴终止移动。 情形 3：断续运行采用长按模式 该信号置 1 后，机床轴移动指定的步数。如果在没有移动完指定步数前，该信号就置 0，则轴中断移动。信号再次置 1 后，轴继续移动。在没有移动完指定步数前，轴的移动可以如上文所述暂停和继续。 情形 4：断续运行采用点按模式 在该信号输出第一个上升沿 0 → 1 后，机床轴开始移动指定的步数。如果在轴移动完指定的步数前，同一个方向信号再次输出上升沿 0 → 1，则轴终止移动。轴因此不再移动余下的行程。 如果两个方向信号同时置位，则轴不会移动或终止当前移动。 通过 PLC 接口信号 DB31, ... DBX4.4（移动键禁用） 可以阻止移动键对某轴的影响。
信号稳态为 0	见“情形 1 到 4”。
信号失效条件	运行方式 AUTO 和 MDI。
应用举例	如果机床轴已经由某通道专用的 PLC 接口作为几何轴移动，则它不能在 JOG 方式下移动。 系统会发出报警 20062。

18.3 手动运行和手轮运行(H1)

DB31, ... DBB4 位 7 和位 6	正向/负向移动键
特殊情况, ...	分度轴
关联信号	DB21, ... DBX12.7, DBX12.6 ff (几何轴的正向/负向移动键) DB31, ... DBX4.4 (移动键禁用)

DB31, ... DBB5 Bit 0-5	机床功能 INC1, INC10, INC100, INC1000, INC10000, INCvar
脉冲沿计算: 不支持	信号更新速率: 周期
信号稳态为 1	请求激活某个“断续移动”机床功能: • 位 0:INC1 • 位 1:INC10 • 位 2:INC100 • 位 3:INC1000 • 位 4: INC10000 • 位 5:INCvar “步”是按下一次正向/负向移动键或者手轮转动一个刻度轴移动的最小单位。 步数在以下系统数据中定义: • INC1 到 INC10000: MD11330 \$MN_JOG_INCR_SIZE_TAB (手动运行/手轮运行中的步数) • INCvar: SD41010 \$SN_JOG_VAR_INCR_SIZE (JOG 方式下的可变步数) NC 通过以下接口信号反馈哪个机床功能激活: DB31, ... DBB65 (生效的机床功能: INC1, ...) 注意 多个位同时置位时, 没有任何机床功能生效。
信号稳态为 0	不请求对应的机床功能。 如果在轴当前正在执行某机床功能时取消该功能或者切换到另一个功能, 则轴终止移动。
关联信号	DB31, ... DBB65 (生效的机床功能: INC1, ...) DB31, ... DBX5.6 (机床功能: 连续运行)

18.3 手动运行和手轮运行(H1)

DB31, ... DBX5.6	机床功能：连续运行	
脉冲沿计算：不支持	信号更新速率：周期	
信号稳态为 1	机床轴可以在 JOG 方式下通过正向/负向移动键朝两个方向连续移动。	
信号稳态为 0	不请求“连续移动”机床功能。	
关联信号	DB31, ... DBB65（生效的机床功能：INC1, ..., 连续运行） DB31, ... DBB5（机床功能：INC1, ..., INC10000）	

DB31, ... DBX7.0	取反机床轴手轮的旋转方向	
脉冲沿计算：不支持	信号更新速率：周期	
信号稳态为 1	取反指定给机床轴的手轮的旋转方向。 只允许在静止状态下修改取反信号。	
信号稳态为 0	不取反手轮的旋转方向。	
应用举例	- 手轮旋转方向和预期的轴移动方向不一致。 - 手轮指定给了多根定向轴。	
关联信号	DB31, ... DBX67.0（对机床轴手轮旋转方向的取反生效）	

DB31, ... DBB13 位 0-2	手动运行至固定点	
脉冲沿计算：不支持	信号更新速率：周期	
信号稳态为 1	激活“手动运行至固定点”。 目标固定点的编号通过二进制数字 0-2 编码。 一旦该功能激活（见 DB31, ... DBX75.0-2），就可以用移动键或手轮将选中的机床轴移动到指定的固定点。 固定点通过以下机床数据设置： MD30600 \$MA_FIX_POINT_POS[n]	
信号稳态为 0	撤销“手动运行至固定点”。	
关联信号	DB31, ... DBX75.0-2（手动运行至固定点生效） DB31, ... DBX75.3-5（已手动运行至固定点） MD30600 \$MA_FIX_POINT_POS[n]（固定点位置）	

18.3 手动运行和手轮运行(H1)

18.3.6 从进给轴/主轴发出的信号(DB31, ...)

来自进给轴/主轴的信号描述

DB31, ... DBX62.1	手轮叠加生效	
脉冲沿计算：不支持		信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	功能“自动方式中的手轮叠加”对程序写入的轨迹轴 (FDA[AXi]) 生效。该轴手轮发出的脉冲因此可作为行程 (FDA=0) 或速度 (FDA > 0) 叠加在程序写入的进给率上。 在 840D sl 上，用 FC18 为并行定位轴激活了“自动方式下的手轮叠加”后，该接口信号也会置位。	
信号稳态为 0 或输出下降沿 1 → 0	功能“自动方式中的手轮叠加”对程序写入的轨迹轴或并行定位轴生效。 在以下条件下，当前生效的手轮叠加会变为无效： - 定位轴到达了编写的目标位置。 - 通过轴接口信号： DB31, ... DBX2.2（删除剩余行程） 删除了剩余行程。 - 按 RESET 键。	

DB31, ... DBB64 位 0-2	手轮生效 (1 至 3)	
脉冲沿计算：不支持		信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	这些 PLC 接口信号用于反馈为该机床轴指定了哪个手轮：手轮 1、2 或 3，或者根本没有指定手轮。 一个时间点上一根轴只能有一个手轮。 如多个接口信号 DB31, ... DBX4.0 到 DBX4.2（激活手轮）置位，则优先级从高到低依次为： 手轮 1、手轮 2、手轮 3。 在手轮指定有效时，在 JOG 方式下可用手轮移动机床轴；在 AUTO 或 MDI 方式下可用手轮产生 DRF 偏移。	
信号稳态为 0 或输出下降沿 1 → 0	该机床轴没有手轮 1、2 或 3。	
关联信号	DB31, ... DBX4.0 bis DBX4.2（激活手轮） DB10 DBB100.6 ff（手轮已选中）	

DB31, ... DBB64 位 5 和位 4	运行请求 +/-
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	轴朝指定方向移动。运行方式不同，运行指令的触发方式也有所不同： - 运行方式 JOG： 按下正向/负向移动键。 - 运行方式 REF： 按下使轴朝参考点移动的移动键。 - 运行方式 AUTO 和 MDI： 执行一段包含轴坐标值的程序段。
信号稳态为 0 或输出下降沿 1 → 0	在对应轴方向上没有任何运行请求，或者运行已结束。 - 运行方式 JOG： 取决于“点按或是长按”的设置，运行指令复位，另见“DB31, ... DBX4.7 或 DBX4.6”。 - 使用手轮运行。 - 运行方式 REF： 达到参考点。 - 运行方式 AUTO 和 MDI： 执行一条程序段，之后的程序段不包含对应轴的坐标值。 - 通过“复位”终止，等等 - DB21 ... DBX25.7（轴禁止）存在。
应用举例	当轴被夹紧时松开轴夹紧装置，比如：回转工作台上。 注： 如果只有在发出运行指令时轴才被松开，则这种轴不支持轨迹运行！
关联信号	DB31, ... DBX4.7 或 DBX4.6（正向/负向移动键） DB31, ... DBX64.7 或 DBX64.6（运行指令 +/-）

18.3 手动运行和手轮运行(H1)

DB31, ... DBB64 位 7 和位 6	运行指令 +/-
脉冲沿计算：不支持	信号更新速率：周期
该信号的作用和以下机床数据的位 0 设为 0 时一样： MD17900 \$MN_VDI_FUNCTION_MASK（VDI 信号设置）。 如果该机床数据的位 0 设为 1，则只有在轴真正移动时，该信号才显示 1。 当 MD17900 位 0 = 0 时，输出的接口信号 DB31, ... DBX64 位 5 和位 4（运行请求 +/-）的作用 和信号“运行指令 +/-”一样。	
信号稳态为 1 或输出上升沿 0 → 1	轴朝指定方向移动。 运行方式不同，运行指令的触发方式也有所不同： ● 运行方式 JOG： 按下正向/负向移动键。 ● 运行方式 REF： 按下使轴朝参考点移动的移动键。 ● 运行方式 AUTO 和 MDI： 执行一段包含轴坐标值的程序段。
信号稳态为 0 或输出下降沿 1 → 0	在对应轴方向上没有任何运行请求，或者运行已结束。 ● 运行方式 JOG： 取决于“点按或是长按”的设置，运行指令复位，另见“DB31, ... DBX4.7 或 DBX4.6”。 ● 运行方式 REF： 达到参考点。 ● 运行方式 AUTO 和 MDI： 执行一条程序段，之后的程序段不包含对应轴的坐标值。 通过“复位”终止，等等 DB21 ... DBX25.7（轴禁止）存在。
应用举例	当轴被夹紧时松开轴夹紧装置，比如：回转工作台上。 注： 如果只有在发出运行指令时轴才被松开，则这种轴不支持轨迹运行！
关联信号	DB31, ... DBX4.7 或 DBX4.6（正向/负向移动键） DB31, ... DBX64.5 或 DBX.4（运行指令 +/-）

DB31, ... DBB65 位 0-6	生效的机床功能: INC1, ..., 连续移动	
脉冲沿计算: 不支持	信号更新速率: 周期	
信号稳态为 1 或输出上升沿 0 → 1	该 PLC 接口信号用于反馈在 JOG 方式下机床轴的哪个机床功能生效。 生效的移动步数不同, 在按下移动轴或旋转手轮时, 系统的响应各不相同。	
信号稳态为 0 或输出下降沿 1 → 0	相应的机床功能无效。	
关联信号	DB31, ... DBB5 (机床功能: INC1, ..., 连续运行)	

DB31, ... DBX67.0	对机床轴手轮的旋转方向的取反生效	
脉冲沿计算: 不支持	信号更新速率: 周期	
描述	该信号可指出某个机床轴手轮的旋转方向是否已取反:	
	1 信号	手轮的旋转方向已取反。
	0 信号	手轮的旋转方向未取反。
关联信号	DB31, ... DBX7.0. (取反机床轴手轮的旋转方向)	

DB31, ... DBB75 位 0-2	手动运行至固定点生效	
脉冲沿计算: 不支持	信号更新速率: 周期	
信号稳态为 1 或输出上升沿 0 → 1	向 PLC 反馈功能“手动运行至固定点”生效。 现在可以用移动键或手轮将选中的机床轴移动到位 0 到 2 中以二进制码指定的固定点。	
信号稳态为 0 或输出下降沿 1 → 0	“手动运行至固定点”未生效。	
关联信号	DB31, ... DBX13.0-2 (手动运行至固定点) DB31, ... DBX75.3-5 (已手动运行至固定点)	

18.5 运行方式组、通道和跨通道取轴(K5)

DB31, ... DBB75 位 3-5	已手动运行至固定点
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	向 PLC 反馈选中轴已手动运行至固定点，并进入“精准停”公差窗口。 当轴通过 NC 程序、FC18（适用于 840D sl）或同步动作等其他方法到达机床坐标系中的目标固定点位置，且实际位置位于“精准停”公差窗口（MD36010 \$MA_STOP_LIMIT_FINE）内时，也会发出上述信号。
信号稳态为 0 或输出下降沿 1 → 0	轴还没有到达固定点。
关联信号	DB31, ... DBX13.0-2（手动运行至固定点） DB31, ... DBX75.0-2（手动运行至固定点生效）

18.4 补偿(K3)

无需信号说明。

18.5 运行方式组、通道和跨通道取轴(K5)

18.5.1 发送到进给轴/主轴的信号(DB31, ...)

DB31, ... DBB8	跨通道取进给轴/主轴												
脉冲沿计算：支持	信号更新速率：周期												
信号稳态为 1 或输出上升沿 0 → 1	必须指定当前轴类型和该轴的当前所处通道。 在由 PLC 触发跨通道取轴时，发送到进给轴/主轴的信号 DB31, ... DBB8 的含义为： <table border="1" data-bbox="363 1621 1439 1921"> <tr> <td>位 0:</td><td>A 为通道指定 NC 进给轴/主轴</td></tr> <tr> <td>位 1:</td><td>B</td></tr> <tr> <td>位 2:</td><td>C</td></tr> <tr> <td>位 3:</td><td>D 为通道指定 NC 进给轴/主轴</td></tr> <tr> <td>位 4:</td><td>由上升沿激活和指定</td></tr> <tr> <td>位 5:</td><td>-</td></tr> </table>	位 0:	A 为通道指定 NC 进给轴/主轴	位 1:	B	位 2:	C	位 3:	D 为通道指定 NC 进给轴/主轴	位 4:	由上升沿激活和指定	位 5:	-
位 0:	A 为通道指定 NC 进给轴/主轴												
位 1:	B												
位 2:	C												
位 3:	D 为通道指定 NC 进给轴/主轴												
位 4:	由上升沿激活和指定												
位 5:	-												

DB31, ... DBB8	跨通道取进给轴/主轴	
	位 6:	-
	位 7:	请求 PLC 进给轴/主轴
信号稳态为 0 或输出下降沿 1 → 0		
关联信号	DB31, ... DBB68 (跨通道取进给轴/主轴) MD20070 \$MC_AXCONF_ASSIGN_MASTER_USED (通道中有效的机床轴号) MD30550 \$MA_AXCONF_ASSIGN_MASTER_CHAN (跨通道取轴中通道的缺省设置)	
特殊情况, 出错, ...		

18.5.2 从进给轴/主轴发出的信号(DB31, ...)

DB31, ... DBB68	跨通道取进给轴/主轴	
脉冲沿计算: 支持	信号更新速率: 周期	
信号稳态为 1 或输出上升沿 0 → 1	指出当前轴类型和该轴的当前所处通道。 在由 PLC 触发跨通道取轴时, 从进给轴/主轴发出的信号 DB31, ... DBB68 的含义为:	
	位 0:	A 通道中的 NC 进给轴/主轴
	位 1:	B
	位 2:	C
	位 3:	D 通道中的 NC 进给轴/主轴
	位 4:	PLC 请求了新类型
	位 5:	允许跨通道取轴
	位 6:	中立进给轴/主轴以及指令轴/往复轴
	位 7:	PLC 进给轴/主轴
信号稳态为 0 或输出下降沿 1 → 0		

18.6 坐标转换 (M1)

DB31, ... DBB68	跨通道取进给轴/主轴
关联信号	DB31, ... DBB8（跨通道取进给轴/主轴） MD20070 \$MC_AXCONF_ASSIGN_MASTER_USED （通道中有效的机床轴号） MD30550 \$MA_AXCONF_ASSIGN_MASTER_CHAN （跨通道取轴中通道的缺省设置）
特殊情况，出错， ...	

18.6 坐标转换 (M1)

18.6.1 从通道发出的信号（DB21, ...）

DB21, ... DBX33.6	转换生效
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	在零件程序中编程了 NC 指令 TRANSMIT、TRACYL、TRAANG 或 TRAORI。NC 执行完相应的程序段后，转换功能开始生效。
信号稳态为 0 或输出下降沿 1 → 0	没有转换功能生效。
更多参考	编程说明之工作准备分册 功能手册之特殊功能分册：章节“3 轴转换到 5 轴转换（F2）”

18.7 测量(M5)

18.7.1 从 NC 发出的信号(DB10)

DB10 DBX107.0 和 DBX107.1	测头被操作	
脉冲沿计算：不支持	信号更新速率：周期	
信号稳态为 1 或输出上升沿 0 → 1	测头 1 或 2 被操作。	
信号稳态为 0 或输出下降沿 1 → 0	测头 1 或 2 未被操作。	
更多参考	NCU 设备手册	

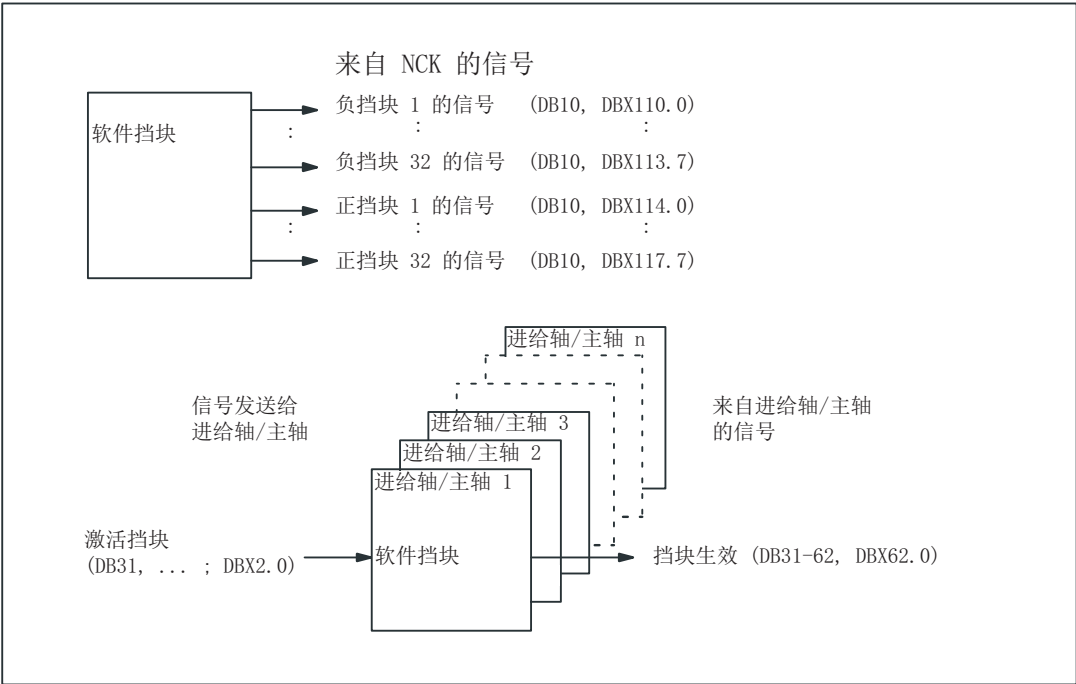
18.7.2 从进给轴/主轴发出的信号(DB31, ...)

DB31, ... DBX62.3	测量状态	
脉冲沿计算：不支持	信号更新速率：周期	
信号稳态为 1 或输出上升沿 0 → 1	功能” 测量“ 有效。 该信号用于测量，指出当前轴的测量状态。	
信号稳态为 0 或输出下降沿 1 → 0	功能” 测量 “无效。	

18.8 软件挡块和行程开关信号(N3)

18.8.1 信号一览

和“软件挡块和行程开关信号”相关的 PLC 接口信号



18.8.2 从 NC 发出的信号(DB10)

DB10 DBX110.0-113.7	负挡块信号 1-32	
脉冲沿计算：不支持		信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	负挡块信号 1 - 32 的脉冲沿和轴（回转轴）的运行方向相关，按插补周期发送给 PLC 接口。 线性轴： 当轴朝负向移动越过负挡块时，负挡块信号会从 0 变 1。 模数回转轴： 每次正挡块信号输出上升沿时，负挡块信号便切换电平。	
信号稳态为 0 或输出下降沿 1 → 0	线性轴： 当轴朝正向移动越过负挡块时，负挡块信号会从 1 变 0。 模数回转轴： 每次正挡块信号输出上升沿时，负挡块信号便切换电平。	

18.8 软件挡块和行程开关信号(N3)

DB10 DBX114.0-117.7	正挡块信号 1-32
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	正挡块信号 1 - 32 的脉冲沿和轴（回转轴）的运行方向相关，按插补周期发送给 PLC 接口。 线性轴： 当轴朝正向移动越过正挡块时，正挡块信号会从 0 变 1。 模数回转轴： 当轴正转越过负挡块时，正挡块信号会从 0 变 1。 上文说明的正挡块作用针对以下条件： 正挡块 – 负挡块 < 180 度 如果不满足此条件或者所选的负挡块大于正挡块，则正挡块信号的响应正好相反。负挡块信号的响应保持不变。
信号稳态为 0 或输出下降沿 1 → 0	线性轴： 当轴朝负向移动越过正挡块时，正挡块信号会从 1 变 0。 模数回转轴： 当轴正转越过正挡块时，正挡块信号会从 1 变 0。 上文说明的正挡块作用针对以下条件： 正挡块 – 负挡块 < 180 度 如果不满足此条件或者所选的负挡块大于正挡块，则正挡块信号的响应正好相反。负挡块信号的响应保持不变。

18.8.3 发送到进给轴/主轴的信号(DB31, ...)

DB31, ... DBX2.0	激活挡块
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	将轴的正向和负挡块信号输出到通用 PLC 接口上。 在处理后接口信号“激活挡块”后，挡块立即激活。
信号稳态为 0 或输出下降沿 1 → 0	不将轴的正向和负挡块信号输出到通用 PLC 接口上。
关联信号	DB10 DBX110.0 - 113.7（负挡块信号 1-32） DB10 DBX114.0 - 117.7（正挡块 1-32）

18.8.4 从进给轴/主轴发出的信号(DB31, ...)

DB31, ... DBX62.0	挡块生效	
脉冲沿计算：不支持	信号更新速率：周期	
信号稳态为 1 或输出上升沿 0 → 1	所有由 NC/PLC 接口信号 DB31, ... DBX2.0（激活挡块） 选择的轴的挡块都成功激活。	
信号稳态为 0 或输出下降沿 1 → 0	轴的挡块未激活。	
关联信号	DB31, ... DBX2.0（激活挡块） DB10 DBX110.0 - 113.7（负挡块信号 1-32） DB10 DBX114.0 - 117.7（正挡块 1-32）	

18.9 冲裁和步冲(N4)

18.9.1 信号一览

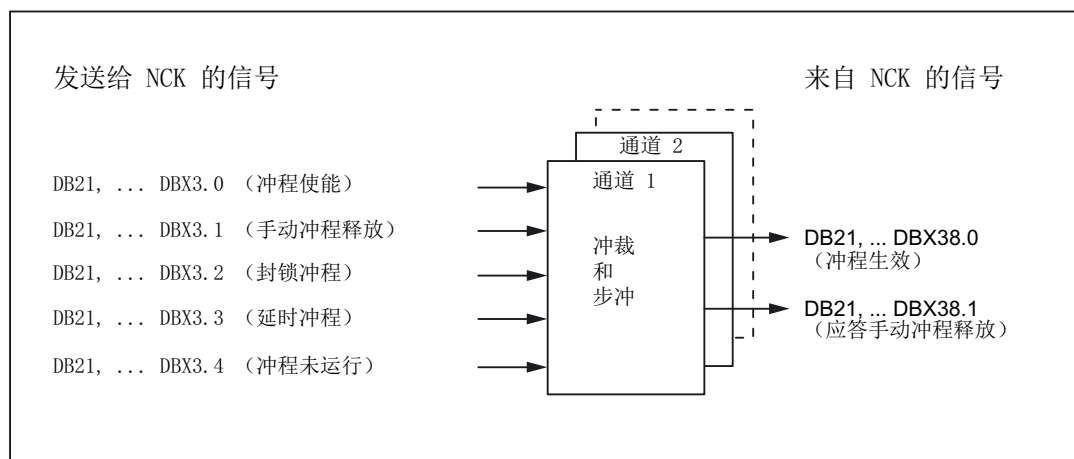


图 18-1 “冲裁和步冲”涉及的 PLC 接口信号

18.9 冲裁和步冲(N4)

18.9.2 发送至通道的信号 (DB21, ...)

DB21, ... DBX3.0		无冲程释放
脉冲沿计算:		信号更新速率:
信号稳态为 1 或输出上升沿 0 → 1	允许由 PLC 使能冲程。 1 信号: 封锁冲程, NC 不允许启动冲程。	
信号稳态为 0 或输出下降沿 1 → 0	0 信号: 具有冲程使能。只有使能未设置, NC 就可以启动冲程。	

DB21, ... DBX3.1		手动冲程释放
脉冲沿计算:		信号更新速率:
信号稳态为 1 或输出上升沿 0 → 1	允许在手动方式中启动一个冲程。 1 信号: 手动释放冲程。	
信号稳态为 0 或输出下降沿 1 → 0	0 信号: 无作用	

DB21, ... DBX3.2		封锁冲程
脉冲沿计算:		信号更新速率:
信号稳态为 1 或输出上升沿 0 → 1	仅封锁冲程。机床仍运行。 如果当前行程自动划分激活, 该功能仍保持激活。 它仅仅是封锁了信号“冲程释放”。 机床在“Stop and Go”方式下运行。 步长由行程划分定义。 1 信号: 封锁冲程。	
信号稳态为 0 或输出下降沿 1 → 0	0 信号: 不封锁冲程。	

DB21, ... DBX3.3		冲程延时
脉冲沿计算:		信号更新速率:
信号稳态为 1 或输出上升沿 0 → 1	激活“冲程延时”。该信号和程序指令 PDELAYON 的功能类似。另外，NCK 将不分析不符合标准的 PLC 信号。信号分析受限，除了手动释放冲程 PON。 1 信号：冲程延时生效。	
信号稳态为 0 或输出下降沿 1 → 0	0 信号：冲程延时无效。	

DB21, ... DBX3.4		冲程未运行
脉冲沿计算:		信号更新速率:
信号稳态为 1 或输出上升沿 0 → 1	NC 立即停止运动来响应该信号。如果该信号会中断某一运动或者其他动作，系统会报警。 物理上该信号与信号“冲程有效”相同，也就是说：这两个信号经过“与”逻辑运算连接到同一个 NC 输入端上。 1 信号：冲程未运行（相当于信号“冲程使能”）	
信号稳态为 0 或输出下降沿 1 → 0	0 信号：冲程未运行（相当于信号“冲程使能”）	

DB21, ... DBX3.5		手动释放冲程
脉冲沿计算:		信号更新速率:
信号稳态为 1 或输出上升沿 0 → 1	允许操作人员释放冲程，即使没有零件程序。此时，冲程由 PLC 控制。 冲程成功启动后，NC 通过 NC/PLC 接口信号 DB21, ... DBX38.1（手动释放冲程应答）将这一状态报告给 PLC。 1 信号：冲程手动释放有效。	
信号稳态为 0 或输出下降沿 1 → 0	0 信号：冲程手动释放无效。	

18.10 定位轴(P2)

18.9.3 从通道发出的信号（DB21, ...）

DB21, ... DBX38.0		冲程释放有效
脉冲沿计算:		信号更新速率:
信号稳态为 1 或输出上升沿 0 → 1	该信号表示冲程启动是否有效。 1 信号: 冲程释放有效。	
信号稳态为 0 或输出下降沿 1 → 0	0 信号: 冲程释放无效。	

DB21, ... DBX38.1		手动冲程释放应答
脉冲沿计算:		信号更新速率:
信号稳态为 1 或输出上升沿 0 → 1	表示冲程是否已手动释放。 1 信号: 冲程已手动释放。	
信号稳态为 0 或输出下降沿 1 → 0	0 信号: 冲程未手动释放。	

18.10 定位轴(P2)

NCK-HMI-PLC 接口上的以下信号或指令只用于定位轴:

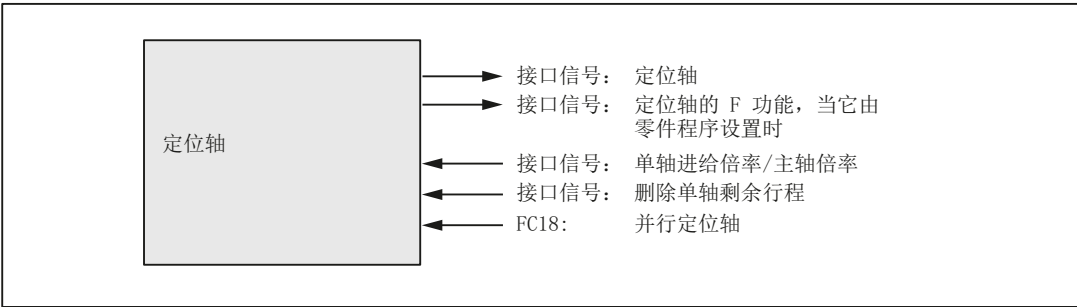


图 18-2 PLC 接口信号

18.10.1 发送到进给轴/主轴的信号(DB31, ...)

DB31, ... DBB0	单轴进给倍率/主轴倍率
脉冲沿计算：不支持	信号更新速率： 周期
信号稳态为 1 或输出上升沿 0 → 1	定位轴有单独的一个单轴进给倍率，其计算方式和单通道进给倍率相同。
信号失效条件	接口信号 DB31, ... DBX74.5 ("定位轴") 置零
更多参考	计算方式参见： DB21, ... DBB4（单通道进给倍率）

DB31, ... DBX2.2	删除单轴剩余行程
脉冲沿计算：支持	信号更新速率： 周期
信号稳态为 1 或输出上升沿 0 → 1	删除定位轴的单轴进给行程。定位轴按减速度制动，滞后量减小至零。系统认为轴到达了程序写入的终点。接口信号“删除剩余行程”对单轴生效，不影响轨迹轴。轨迹轴要使用接口信号“删除单通道剩余行程”。
特殊情况， 出错，...	如果接口信号“删除单通道剩余行程”置位，尽管在该程序段中编写了定位轴，NCK 不作任何响应。
关联信号	DB21, ... DBX6.2（删除单通道剩余行程，可作用于轨迹轴）

DB31, ... DBX28.1	复位
脉冲沿计算：支持	信号更新速率： 周期
信号稳态为 1 或输出上升沿 0 → 1	PLC 向 NCK 请求复位自己控制的进给轴/主轴 NCK 通过以下信号反馈给 PLC： DB31 ... DBX63.1 = 1（复位已执行） DB31 ... DBX63.2 = 1（轴停止生效）
特殊情况， 出错，...	边界条件： ● 当前进给轴/主轴必须由 PLC 控制。
关联信号	DB31 ... DBX63.1（复位已执行） DB31, ... DBX63.2（轴停止生效） 系统变量：\$AA_SINGLAX_STAT OPI 变量：aaSnglAxStat

18.10 定位轴(P2)

DB31, ... DBX28.2	继续
脉冲沿计算：支持	信号更新速率： 周期
信号稳态为 1 或输出上升沿 0 → 1	请求由 PLC 控制的进给轴/主轴继续运行。 该请求可通过接口信号 DB31 ... DBX63.2（“轴停止生效”）终止。
特殊情况， 出错，...	边界条件： ● 当前进给轴/主轴必须由 PLC 控制。 ● 出现以下错误时，系统会忽略该信号： – 进给轴/主轴不再由 PLC 控制。 – 进给轴/主轴不再处于停止状态。 – 进给轴/主轴由于出现报警而无法继续运行。
关联信号	DB31, ... DBX28.1（复位） DB31, ... DBX60.6（粗准停） DB31, ... DBX60.7（精准停） DB31 ... DBX63.2（轴停止生效） DB31, ... DBX64.6（运行指令 -） DB31, ... DBX64.7（运行指令 +） 系统变量： \$AA_SNGLAX_STAT OPI 变量： aaSnglAxStat

DB31, ... DBX61.1	轴报警
脉冲沿计算：不支持	信号更新速率： 周期
信号稳态为 1 或输出上升沿 0 → 1	结果： ● NCK 使进给轴/主轴沿着一个制动斜坡停止。 ● OPI 变量： aaSnglAxStat = 5（报警） ● \$AA_SNGLAX_STAT = 5（出现轴报警） ● DB31 ... DBX61.1 = 1（轴报警）

DB31, ... DBX63.0	复位已执行
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	复位结束后的状态： ● 重新载入进给轴/主轴的机床数据 ● DB31 ... DBX63.0 == 1（复位已执行） ● DB31 ... DBX63.2 == 0（轴停止生效） ● 系统变量 \$AA_SINGLAX_STAT == 1 ● OPI 变量：aaSnglAxStat == 1
关联信号	DB31, ... DBX28.1（复位）

DB31, ... DBX63.1	PLC 控制轴
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	NC 向 PLC 确认，轴现在已交由 PLC 控制。
关联信号	DB31 ... DBX28.7（PLC 控制轴） 系统变量：\$AA_SINGLAX_STAT

DB31, ... DBX63.2	轴停止生效
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	NC 向 PLC 反馈正在停止轴。
信号稳态为 0 或输出下降沿 1 → 0	NC 向 PLC 反馈轴已停止。 系统变量：\$AA_SINGLAX_STAT = 3（单轴运行中断）
关联信号	DB31, ... DBX60.6（粗准停） DB31, ... DBX60.7（精准停） DB31 ... DBX63.2（轴停止生效） DB31, ... DBX64.6（运行指令 -） DB31, ... DBX64.7（运行指令 +） 系统变量：\$AA_SINGLAX_STAT

18.10 定位轴(P2)

DB31, ... DBX76.5	定位轴
脉冲沿计算：不支持	信号更新速率： 周期
信号稳态为 1 或输出上升沿 0 → 1	NC 向 PLC 反馈轴已成为定位轴。

DB31, ... DBD78	定位轴的 F 功能（进给率）
脉冲沿计算： 不支持	信号更新速率： 立即
功能	为定位轴编写的单轴进给率。 在 840D sl 上不输出由 FC18 指定的值。
信号失效条件	DB31, ... DBX76.5 == 0（轴不是定位轴）
特殊情况， 出错，...	如果定位轴以机床数据中的进给率移动，则 NC 不会向 PLC 输出进给率： MD32060 \$MA_POS_AX_VELO（定位轴速度的初始设置）
关联信号	DB31, ... DBX76.5（定位轴） MD22240 \$MC_AUXFU_F_SYNC_TYP（F 功能的输出时间）

18.10.2 Function Call - 仅限 840D sl

FC18

在 SINUMERIK 840D sl 上，通过 PLC 的 FC18 (Function Call 18)可以由 PLC 启动并行定位轴。此时，FC 会获得以下参数：

- 轴名/轴号
- 终点位置
- 进给率
(在进给率为 0 时，采用来自 MD32060 \$MA_POS_AX_VELO 的进给率)
FC18 的 F 值不传送到轴专用接口信号 DB31, ...DBB78-81“定位轴的 F 功能：进给率”中。
- 绝对坐标(G90)、相对坐标(G91)、回转轴以最短行程绝对定位：回转轴名 = DC(值)

由于一根轴只能属于一个通道，因此系统可从轴名或轴号中选择其所在通道，然后启动该通道中的并行定位轴。

文档:

功能手册之基本功能分册: SINUMERIK 840D sl 的 PLC 基本程序 (P3)

18.11 往复运动(P5)

18.11.1 发送给进给轴/主轴信号(DB31, ...)

VDI 输入信号

通过以下信号 PLC 用户程序可以控制往复运动。

DB31, ... DBX28.0	从外部触发往复轴换向	
脉冲沿计算: 支持	信号更新速率: 周期	
信号稳态为 1 或输出上升沿 0 → 1	中断往复运动, 然后使往复轴在相反方向上继续运行。	
信号稳态为 0 或输出下降沿 1 → 0	不中断往复运动	

DB31, ... DBX28.3	设置换向点	
脉冲沿计算: 不支持	信号更新速率: 周期	
信号稳态为 1 或输出上升沿 0 → 1	换向点 2	
信号稳态为 0 或输出下降沿 1 → 0	换向点 1	

DB31, ... DBX28.4	修改换向点	
脉冲沿计算: 不支持	信号更新速率: 周期	
信号稳态为 1 或输出上升沿 0 → 1	允许通过手动运行修改选中的换向点。 和 DB31, ...DBX28.0 组合使用: 外部触发了往复轴换向导致的制动点应成为新的换向点。	

18.11 往复运动(P5)

DB31, ... DBX28.4	修改换向点
信号稳态为 0 或输出下降沿 1 → 0	不允许通过手动运行修改选中的换向点。 和 DB31, ...DBX28.0 组合使用： 不改变换向点
关联信号	DBX28.3

DB31, ... DBX28.5	在下一个换向点上停止
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	在下一个换向点上往复运动中断。
信号稳态为 0 或输出下降沿 1 → 0	在经过下一个换向点后往复运动继续。
关联信号	DBX28.6, DBX28.7

DB31, ... DBX28.6	沿制动斜坡停止
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	轴沿斜坡停止，往复运动中断。
信号稳态为 0 或输出下降沿 1 → 0	往复运动继续，不被中断。
关联信号	DBX28.5, DBX28.7

DB31, ... DBX28.7	PLC 控制轴
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	轴由 PLC 控制。 对接口信号的响应由 PLC 通过 2 个停止位控制，其他会导致制动的信号被忽略。
信号稳态为 0 或输出下降沿 1 → 0	轴不由 PLC 控制。
关联信号	DBX28.5, DBX28.6

18.11.2 从进给轴/主轴发出的信号(DB31, ...)

VDI 输出信号

NCK 向 PLC 用户程序返回以下信号。

DB31, ... DBX100.2	往复轴换向生效
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	当前正在进行外部触发的往复轴换向(DB31, ...DBX28.0)导致的制动
信号稳态为 0 或输出下降沿 1 → 0	没有由外部触发的往复轴换向导致的制动

DB31, ... DBX100.3	往复运动无法启动
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	无法启动往复轴，编程错误。该状态也可能出现在往复轴正在运行的过程中。
信号稳态为 0 或输出下降沿 1 → 0	可以启动往复运动。

DB31, ... DBX100.4	往复运动出错
脉冲沿计算：	信号更新速率：
信号稳态为 1 或输出上升沿 0 → 1	往复运动因出现异常而结束。
信号稳态为 0 或输出下降沿 1 → 0	往复运动正常进行。

DB31, ... DBX100.5	正进行光磨
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	轴正在进行光磨。

18.11 往复运动(P5)

DB31, ... DBX100.5	正进行光磨
信号稳态为 0 或输出下降沿 1 → 0	轴当前不正在进行光磨。
关联信号	DBX100.7

DB31, ... DBX100.6	正进行往复运动
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	轴正在 2 个换向点之间来回移动。
信号稳态为 0 或输出下降沿 1 → 0	轴当前不作往复运动。
信号失效条件	DBX100.7 = 0
关联信号	DBX100.7

DB31, ... DBX100.7	往复运动激活
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	轴目前作为往复轴运行。
信号稳态为 0 或输出下降沿 1 → 0	轴是定位轴
关联信号	DBX100.5, DBX100.6

DB31, ... DBX104.0 - 7	生效的进给轴
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	发出该信号的轴当前是往复轴，它用该信号反馈当前生效的进给轴（104.0 表明轴 1 是进给轴；104.1 表明轴 2 是进给轴，以此类推）。
信号稳态为 0 或输出下降沿 1 → 0	轴不是进给轴。
关联信号	DBX100.7

18.12 回转轴(R2)

18.12.1 发送到进给轴/主轴的信号(DB31, ...)

DB31, ... DBX12.4	模数回转轴上的工作区域限制	
脉冲沿计算：不支持	信号更新速率：周期	
信号稳态为 1 或输出上升沿 0 → 1	激活模数回转轴上的工作区域限制（软件限位开关、工作区域限制）。	
信号稳态为 0 或输出下降沿 1 → 0	不激活模数回转轴上的工作区域限制。	
信号失效条件	线性轴和没有激活模数功能的回转轴。	
应用举例	周边磨砂轮，带有监控	

18.12.2 从进给轴/主轴发出的信号(DB31, ...)

DB31, ... DBX74.4	模数回转轴的监控状态	
脉冲沿计算：不支持	信号更新速率：周期	
信号稳态为 1 或输出上升沿 0 → 1	模数回转轴上的工作区域限制生效（软件限位开关、工作区域限制）。	
信号稳态为 0 或输出下降沿 1 → 0	模数回转轴上的工作区域限制不生效。	
信号失效条件	线性轴和没有激活模数功能的回转轴。	
应用举例	周边磨砂轮，带有监控	

18.13 同步主轴(S3)

18.13.1 发送到进给轴/主轴的信号(DB31, ...)

DB31, ... DBX31.5	禁用同步
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	PLC 不禁止副主轴的同步运动。 位置偏移和之前一样仍生效。
信号稳态为 0 或输出下降沿 1 → 0	PLC 禁止副主轴的同步运动。 由写入偏移量产生的副主轴同步运动被封锁。副主轴不会有额外的运动。
关联信号	DB31, ... DBX98.1 (粗同步) DB31, ... DBX98.0 (精同步)

18.13.2 从进给轴/主轴发出的信号(DB31, ...)

DB31, ... DBX84.4	主轴同步
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	主轴处于“主轴同步”运行方式中。因此，副主轴以定义的传动比跟踪主主轴的运动。 在主轴同步中，系统执行“精同步”和“粗同步”监控。 注： 该信号仅针对设为副主轴的机床轴，即接口信号“副主轴有效”为 1。
信号稳态为 0 或输出下降沿 1 → 0	主轴不作为“主轴同步”中的副主轴运行。 在撤销主轴同步后，副主轴进入“开环控制方式”。
关联信号	DB31, ... DBX98.0 (精同步) DB31, ... DBX98.1 (粗同步) DB31, ... DBX99.1 (副主轴有效)

DB31, ... DBX98.0	精同步
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	副主轴和主主轴之间的位置差或者速度差进入“精同步”公差带。
信号稳态为 0 或输出下降沿 1 → 0	副主轴和主主轴之间的位置差或者速度差没有进入“精同步”公差带。 注： 该信号只用于主轴同步中的副主轴。
应用示例	在将工件从主主轴传送到副主轴时夹紧工件。只有在两个主轴达到充分同步时，PLC 用户程序才会触发工件的夹紧动作。
关联信号	DB31, ... DBX84.4（主轴同步） MD37210 \$MA_COUPLE_POS_TOL_FINE（“精同步”的阈值） MD37230 \$MA_COUPLE_VELO_TOL_FINE（精速度公差）

DB31, ... DBX98.1	粗同步
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	副主轴和主主轴之间的位置差或者速度差进入“粗同步”公差带。 注： 该信号只用于主轴同步中的副主轴。
信号稳态为 0 或输出下降沿 1 → 0	副主轴和主主轴之间的位置差或者速度差没有进入“粗同步”公差带。
应用示例	在将工件从主主轴传送到副主轴时夹紧工件。只有在两个主轴达到充分同步时，PLC 用户程序才会触发工件的夹紧动作。
关联信号	DB31, ... DBX84.4（主轴同步） MD37200 \$MA_COUPLE_POS_TOL_COARSE（“粗同步”的阈值） MD37220 \$MA_COUPLE_VELO_TOL_COARSE（粗速度公差）

DB31, ... DBX98.2	实际值同步
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	主主轴和副主轴之间的同步方式为实际值同步，参见 MD21310。 注： 该信号只用于主轴同步中的副主轴。

18.13 同步主轴(S3)

DB31, ... DBX98.2	实际值同步
信号稳态为 0 或输出下降沿 1 → 0	主主轴和副主轴之间的同步方式为设定值同步，参见 MD21310。
特殊情况，出错，...	如果在副主轴上出现的故障导致该轴的伺服使能被取消，在一定前提条件下，系统会自动将两者的同步方式切换到实际值同步。
关联信号	DB31, ... DBX84.4（主轴同步） MD21310 \$MC_COUPLING_MODE_1（主轴同步方式）

DB31, ... DBX98.4	叠加运动
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	除了由主主轴产生的运动分量外，副主轴还正在执行额外的一个运动分量。 副主轴的叠加运动示例： - 激活主轴同步，同时主主轴和副主轴之间存在指定的角度偏移 - 在主主轴正在旋转时激活了主轴同步 - 在主轴同步生效期间修改了传动比 - 在主轴同步生效期间指定了新的角度偏移 - 在主轴同步生效期间用正负移动键或手轮手动运行了副主轴 一旦副主轴执行了叠加运动，便可以清除接口信号“精同步”或“粗同步”（取决于阈值）。 注： 该信号只用于主轴同步中的副主轴。
信号稳态为 0 或输出下降沿 1 → 0	副主轴不正在执行叠加运动或该运动已经结束。
关联信号	DB31, ... DBX84.4（主轴同步）

DB31, ... DBX99.0	主主轴有效
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	机床轴当前作为主主轴工作。 注： 该信号只用于主轴同步。
信号稳态为 0 或输出下降沿 1 → 0	机床轴当前不作为主主轴工作。

DB31, ... DBX99.0	主主轴有效
特殊情况，出错，...	如果在副主轴上出现的故障导致该轴的伺服使能被取消，在一定前提条件下，系统会自动将互换两根轴，并将两者的同步方式切换到实际值同步。 此时原先的主主轴变为新的副主轴（接口信号“副主轴有效”）。
关联信号	DB31, ... DBX84.4（主轴同步） DB31, ... DBX99.1（副主轴有效）

DB31, ... DBX99.1	副主轴有效
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	机床轴当前作为副主轴工作。 副主轴以定义的传动比跟踪主主轴的运动。 注： 该信号只用于主轴同步。
信号稳态为 0 或输出下降沿 1 → 0	机床轴当前不作为副主轴工作。
特殊情况，出错，...	如果在副主轴上出现的故障导致该轴的伺服使能被取消，在一定前提条件下，系统会自动将互换两根轴，并将两者的同步方式切换到实际值同步。
关联信号	DB31, ... DBX84.4（主轴同步） DB31, ... DBX99.0（主主轴有效）

18.14 存储器配置(S7)

无需信号说明。

18.15 分度轴(T1)

18.15.1 从进给轴/主轴发出的信号(DB31, ...)

DB31, ... DBX76.6	分度轴就位
脉冲沿计算：不支持	信号更新速率：周期
信号稳态为 1 或输出上升沿 0 → 1	该信号取决于“精准停”状态。 一旦进入“精准停”窗口，该信号就置位。 一旦离开“精准停”窗口，该信号就复位。 - 分度轴位于一个分度位置上。 - 分度轴通过特殊的分度位置指令定位。 注： 一旦分度轴进入“精准停”窗口，位于某个分度位置，该信号就会置位，不管分度轴是如何就位的：
信号稳态为 0 或输出下降沿 1 → 0	- 该轴不是分度轴。 - 分度轴运行： DB31, ... DBX64.7/64.6（运行指令 -/+）置位。 - 分度轴位于某个非分度位置的位置上。 示例： - 在 JOG 方式中以 RESET 终止轴运动。 - 在 AUTO 方式中分度轴用 AC 或 DC 指令定位到任意一个位置上。 - 在 AUTO 方式中，分度轴不用分度位置指令 (CAC, CACP, CACN, CDC, CIC)定位。 - 分度轴的伺服使能被取消： DB31, ... DBX2.1（伺服使能）
信号失效条件	... 轴没有定位为分度轴： MD30500 \$MA_INDEX_AX_ASSIGN_POS_TAB = 0
应用举例	刀库：一旦分度轴就位，即：DB31, ... DBX76.6（分度轴就位） = 1，从刀库中取刀的机械手就激活。 该过程由 PLC 用户程序控制。

18.17 磨削专用的刀具补偿和监控(W4)

DB31, ... DBX76.6	分度轴就位
特殊情况，出错， ...	注： 也可以通过零点偏移（比如 DRF）来修改分度位置表中的各个位置。 如果分度轴的实际位置 等于分度位置表中位置减去补偿量之差， 则接口信号 DB31, ... DBX76.6（分度轴就位）置 1。 如果在 AUTO 方式中分度轴上设置了 DRF 偏移，则即使轴不再位于分度位置上，接口 信号（分度轴就位）也继续保持置位。
关联信号	MD30500 \$MA_INDEX_AX_ASSIGN_POS_TAB（轴是分度轴）

18.16 换刀(W3)

无需信号说明。

18.17 磨削专用的刀具补偿和监控(W4)

18.17.1 从进给轴/主轴发出的信号(DB31, ...)

DB31, ... DBX83.3	几何尺寸监控
脉冲沿计算：不支持	信号更新速率： -
信号稳态为 1 或输出上升沿 0 → 1	砂轮几何尺寸出错。 注： 该监控功能响应不会导致任何后续动作。 如需后续动作，需要在 PLC 用户程序中编程。
信号稳态为 0 或输出下降沿 1 → 0	砂轮几何尺寸正常。
应用举例	磨削专用的刀具监控。

18.17 磨削专用的刀具补偿和监控(W4)

DB31, ... DBX83.6	转速监控	
脉冲沿计算：不支持	信号更新速率： -	
信号稳态为 1 或输出上升沿 0 → 1	砂轮转速出错。 注： 该信号状态不会导致任何后续动作。 如需后续动作，需要在 PLC 用户程序中编程。	
信号稳态为 0 或输出下降沿 1 → 0	砂轮转速正常。	
应用举例	磨削专用的刀具监控。	

DB31, ... DBX84.1	砂轮圆周速度生效	
脉冲沿计算：不支持	信号更新速率： -	
信号稳态为 1 或输出上升沿 0 → 1	“恒定砂轮圆周速度（GWPS）” 功能生效。 GWPS 生效时，所有 PLC 给出的 S 值被视为砂轮圆周速度。	
信号稳态为 0 或输出下降沿 1 → 0	“恒定砂轮圆周速度（GWPS）” 功能未生效。	
应用举例	GWPS 在所有运行方式中生效。	

附录

A

A.1 缩略语列表

A	
A	输出
ADI4	Analog Drive Interface for 4 Axes: 4 轴的模拟驱动接口
AC	Adaptive Control: 自适应控制
ALM	Active Line Module: 调节型电源模块
ARM	异步旋转电机
AS	自动化系统
ASCII	American Standard Code for Information Interchange: 美国信息互换标准码
ASIC	Application Specific Integrated Circuit: 用户自行开发的专用集成电路
ASUP	异步子程序
AUXFU	Auxiliary Function: 辅助功能
AWL	指令列表
AWP	用户程序

B	
BA	运行方式
BAG	运行方式组
BCD	Binary Coded Decimals: 用二进制代码编码的十进制数
BERO	带有反馈振荡器的非接触式限位开关
BI	Binector Input: 二进制互联输入
BICO	Binector Connector
BIN	Binary Files: 二进制文件
BIOS	Basic Input Output System: 基本输入输出系统
BCS	基准坐标系

A.1 缩略语列表

B	
BO	Binector Output: 二进制互联输出
BTSS	操作面板接口

C	
CAD	Computer-Aided Design: 计算机辅助设计
CAM	Computer-Aided Manufacturing: 计算机辅助制造
CC	Compile Cycle: 编译循环
CI	Connector Input: 模拟量互联输入
CF 卡	Compact Flash-Card: CF 卡
CNC	Computerized Numerical Control: 计算机数字控制
CO	Connector Output: 模拟量互联输出
CoL	Certificate of License: 许可证书
COM	通讯
CPA	Compiler Projecting Data: 编译器投影数据
CRT	Cathode Ray Tube: 阴极射线管
CSB	Central Service Board: PLC 模块
CU	Control Unit: 控制单元
CP	Communication Processor: 通讯处理器
CPU	Central Processing Unit: 中央处理器
CR	Communication Processor (回车键)
CTS	Clear To Send: 串行接口发送就绪状态
CUTCOM	Cutter Radius Compensation: 刀具半径补偿

D	
DAU	数模转换器
DB	PLC 数据模块
DBB	数据块-字节 (PLC)
DBD	数据块-双字 (PLC)
DBW	数据块-字 (PLC)

D	
DBX	数据块-位（PLC）
DDE	Dynamic Data Exchange: 动态数据交换
DIN	德国工业标准
DIO	Data Input/Output: 数据传送显示
DIR	Directory: 目录
DLL	Dynamic Link Library: 动态链接库
DO	Drive Object: 驱动对象
DPM	Dual Port Memory: 双端口存储器
DPR	Dual Port RAM: 双端口存储器
DRAM	动态存储器（未缓冲）
DRF	Differential Resolver Function: 微分旋转变压器功能（手轮）
DRIVE-CLiQ	带 IQ 的驱动组件连接
DRY	Dry Run: 空运行进给
DSB	Decoding Single Block: 解码单程序段
DSC	Dynamic Servo Control / Dynamic Stiffness Control: 动态伺服控制
DW	数据字
双字	双字（当前 32 位）

E	
E	输入
I/O	输入/输出
ENC	Encoder: 实际值编码器
EFP	简易外设模块（PLC I/O 模块）
EGB	静电敏感元器件
EMC	电磁兼容性
EN	欧洲标准
EnDat	编码器接口
EPROM	Erasable Programmable Read Only Memory: 可删除、可编程的只读存储器
ePS 网络服务	以网络为基础的机床远程维护服务

A.1 缩略语列表

E	
EQN	2048 正弦信号/转绝对值编码器的类型名称
ES	Engineering System: 工程系统
ESR	扩展的停止和退回
ETC	ETC 键“>”；相同菜单中的扩展软键栏

F	
FB	功能块 (PLC)
FC	Function Call: 功能块 (PLC)
FEPROM	Flash-EPROM: 可读写存储器
FIFO	First In First Out: 存储器, 工作无需地址说明, 数据按存储的顺序读入
FIPO	精插补器
FPU	Floating Point Unit: 浮点单元
FRK	铣削半径补偿
FST	Feed Stop: 进给停止
FUP	功能图 (一种 PLC 编程方法)
FW	固件

G	
GC	全局控制 (PROFIBUS: 广播报文)
GEO	几何属性, 例如几何值
GIA	Gear Interpolation Data: 齿轮插补素具
GND	Signal Ground: 信号地
GP	基本程序 (PLC)
GS	齿轮级
GSD	设备主数据文件, 用于说明 PROFIBUS 从站。+
GSDML	Generic Station Description Markup Language: 基于 XML 的描述语言, 用于创建 GSD 文件
GUD	Global User Data: 全局用户数据

H	
HEX	十六进制数代号
HiFu	辅助功能
HLA	液压直线驱动
HMI	Human Machine Interface: SINUMERIK 操作介面
HSA	主轴驱动
HW	硬件

I	
IBN	调试
IKA	可插补补偿
IM	Interface-Modul: 接口模块
IMR	Interface-Modul Receive: 接收方接口模块
IMS	Interface-Modul Send: 发送方接口模块
INC	Increment: 增量尺寸
INI	Initializing Data: 初始化数据
IPO	插补器
ISA	国际标准体系
ISO	International Standard Organization: 国际标准组织

J	
JOG	Jogging: 点动模式

K	
K_v	控制环的增益系数
K_p	比例增益
$K_{\bar{u}}$	传动比
LAD	梯形图（一种 PLC 编程方法）

A.1 缩略语列表

L	
LAI	Logic Machine Axis Image: 逻辑加工轴映射
LAN	Local Area Network: 本地局域网
LCD	Liquid-Crystal Display: 液晶显示器
LED	Light Emitting Diode: 发光二极管
LF	Line Feed: 进线电源
LMS	位置测量系统
LR	位置控制器
LSB	Least Significant Bit: 最低位
LUD	Local User Data: 用户数据（局部）

M	
MAC	Media Access Control: 媒体访问控制
MAIN	Main program: 主程序 (OB1, PLC)
MB	兆字节
MCI	Motion Control Interface: 运动控制接口
MCIS	运动控制信息系统
MCP	Machine Control Panel: 机床控制面板
MD	机床数据
MDA	Manual Data Automatic: 手动数据输入
MELDW	信息字
MCS	机床坐标系
MLFB	产品订货号
MM	电机模块
MPF	Main Program File: 主程序（NC）
MSTT	机床控制面板

N	
NC	Numerical Control: 数字控制系统
NCK	Numerical Control Kernel: 带有程序段处理，运行范围等等的数字内核

N	
NCU	Numerical Control Unit: NCK 的硬件单元
NRK	NCK 操作系统名称
NST	接口信号
NURBS	非一致性数理 B 样条
NV	零点偏移
NX	Numerical Extension: 轴扩展模块

O	
OB	PLC 中组织块
OEM	原始设备制造商
OP	Operation Panel: 操作面板
OPI	Operation Panel Interface: 操作面板接口
OPT	Options: 选件
OLP	Optical Link Plug: 光导线总线插头
OSI	Open Systems Interconnection: 计算机通讯标准

P	
PAA	输出端过程图
PAE	输入端过程图
PC	个人计算机
PCIN	与控制系统进行数据更换的软件名称
PCMCIA	个人计算机存储卡国际协会: 存储器插卡标准
PCU	PC Unit: PC 主机 (计算机元件)
PG	编程器
PKE	参数识别: PKW 的一部分
PKW	参数识别: 值 (PPO 的参数部分)
PLC	Programmable Logic Control: 可编程逻辑控制器
PN	PROFINET

A.1 缩略语列表

P	
PNO	PROFIBUS 用户组织
PO	上电
POE	程序组织单元
POS	位置/定位
POSMO A	Positioning Motor Actuator: 定位电机
POSMO CA	Positioning Motor Compact AC: 集成了功率模块、控制模块、定位单元以及程序存储器的完整驱动单元; 使用交流电源
POSMO CD	Positioning Motor Compact DC: 同 CA, 但采用直流电源
POSMO SI	Positioning Motor Servo Integrated: 定位电机, 采用直流电源
PPO	Parameter Prozessdaten Objekt: 通过 PROFIBUS-DP 和“转速可变驱动”传输时的循环数据报文
PPU	Panel Processing Unit: 基于面板的 CNC 控制系统 (如 SINUMERIK 828D) 的核心硬件
PROFIBUS	Process Field Bus: 串行数据总线
PRT	程序测试
PSW	程序控制字
PTP	Point to Point: 点到点
PUD	Program Global User Data: 程序全局用户变量
PZD	过程数据: PPO 的过程数据部分

Q	
QFK	象限误差补偿

R	
RAM	Random Access Memory: 读写存储器
REF	返回参考点功能
REPOS	再定位功能
RISC	Reduced Instruction Set Computer: 精简指令集计算机: 带有小命令集和快速命令处理的处理器类型
ROV	Rapid Override: 快速倍率

R	
RP	R 参数，计算参数，预定义用户变量
RPA	R-Parameter Active: NCK 中 R 参数号的存储区
RPY	Roll Pitch Yaw: 一种坐标系旋转方式
RTL	Rapid Traverse Linear Interpolation: 快速运行时的直线插补
RTS	Request To Send: 开启发送方，控制信号自串行数据接口
RTCP	Real Time Control Protocol: 实时控制协议

S	
SA	同步动作
SBC	Safe Break Control: 安全制动控制
SBL	Single Block: 单程序段
SBR	Subroutine: 子程序（PLC）
SD	设定数据
SDB	系统数据块
SEA	Setting Data Active: 设定数据符号（文件类型）
SERUPRO	Search-Run by Program Test: 程序测试时搜索
SFB	系统功能块
SFC	系统功能调用
SGE	安全输入
SGA	安全输出
SH	安全停止
SIM	单直列模块
SK	软键
SKP	Skip: 跳至零件程序段末尾
SLM	同步直线电机
SM	步进电机
SMC	Sensor Module Cabinet Mounted: 机柜安装式编码器模块
SME	Sensor Module Externally Mounted: 外部安装的编码器模块
SMI	Sensor Module Integrated: 集成编码器模块
SPF	Sub Program File: 子程序（NC）

A.1 缩略语列表

S	
SPS	Speicherprogrammierbare Steuerung（可编程逻辑控制）= PLC
SRAM	静态存储器（缓存）
SRK	刀沿半径补偿
SRM	同步旋转电机
SSFK	主轴丝杆螺距误差补偿
SSI	Serial Synchron Interface: 串行同步接口
SSL	程序段搜索
STW	控制字
GWPS	砂轮外缘速度
SW	软件
SYF	System Files: 系统文件
SYNACT	Synchronized Action: 同步动作

T	
TB	Terminal Board: 端子板（SINAMICS）
TCP	Tool Center Point: 刀尖
TCP/IP	Transport Control Protocol / Internet Protocol: 传输控制协议/因特网互联协议
TCU	Thin Client Unit: 薄型客户单元
TEA	Testing Data Active: 机床数据标识
TIA	全集成自动化
TM	Terminal Module: 端子模块（SINAMICS）
TO	Tool Offset: 刀具补偿
TOA	Tool Offset Active: 刀具补偿标识（文件类型）
TRANSMIT	Transform Milling Into Turning: 车床上铣削加工的坐标转换
TTL	逻辑门电路（接口类型）
TZ	工艺循环

U	
UFR	User Frame: 零点偏移
UP	子程序
USB	Universal Serial Bus: 通用串行总线
USV	不间断电源

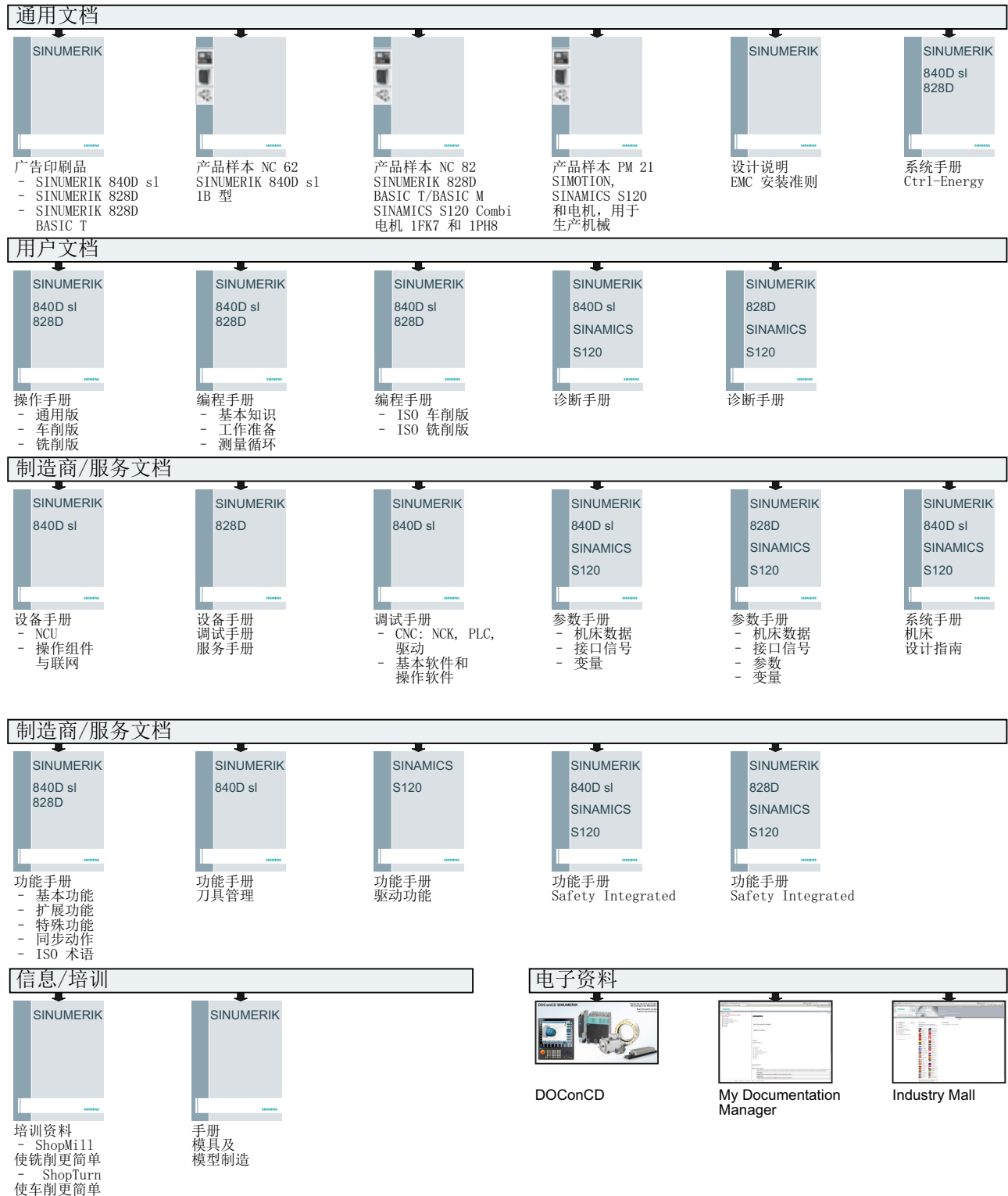
V	
VDI	NCK 和 PLC 间的内部通讯接口
VDI	德国工程师协会
VDE	德国电工技术人员联合会
VI	电压输入
VO	电压输出
VSA	进给驱动

W	
SAR	平滑逼近和退回功能
WCS	工件坐标系
WKZ	刀具
WLK	刀具长度补偿
WOP	现场编程
WPD	Work Piece Directory: 工件目录
WRK	刀具半径补偿
WZ	刀具
WZK	刀具补偿
WZV	刀具管理
WZW	换刀

X	
XML	Extensible Markup Language: 可扩展标记语言

Z	
ZOA	Zero Offset Active: 零点偏移标识
ZSW	(驱动) 状态字

A.2 手册一览



扩展功能

功能手册, 03/2013, 6FC5397-1BP40-3RA1

词汇表

“行程-时间”挡块

“行程-时间”挡块是一个软件挡块对，可在指定的轴输出持续指定时间的脉冲。

C 样条

C 样条是最为人所知也是一种最常用的样条。支点之间的过渡为相切过渡，保持恒定曲率。使用 3 级多项式。

C 轴

一根作受控的旋转运动以定位工件主轴的轴。

CNC

参见 → NC

CNC 标准语言

标准语言提供：→ 用户定义变量、→ 系统变量、→ 宏技术。

COM

控制系统部件，用于执行和和协调通讯。

CPU

中央处理单元，参见 → 可编程控制器

DRF

Differential Resolver Function: NC 功能，在自动方式下利用电子手轮产生增量式零点偏移。

HIGHSTEP

编程方法汇编，用于。→ 系统 AS300/AS400 中的 PLC。

HW-Config

SIMATIC S7 工具，用于配置和设置 S7 项目中的硬件。

JOG

控制系统的一种运行方式（调试方式）：在 JOG 运行方式下，可以调试机床。各个进给轴和主轴可以通过方向键断续运行。在 JOG 手动运行方式中还有其它的一些功能，如 → 回参考点运行，→ 重新定位以及预设（设定实际值）。

Kp

传动比

KV

回路增益系数，控制理论中控制环的一个参数

MDA

控制系统的一种运行方式：手动输入，自动运行 在 MDA 方式下，可以输入单个程序段或者几个程序段，按下 CYCLE START 键可以立即执行。

NC

Numerical Control: 数控系统/控制系统，包括所有机床控制组件：→ NCK, → PLC, HMI, → COM.

说明

对于 SINUMERIK 控制系统而言，更为准确的名称应该是计算机数控系统：Computerized Numerical Control。

NCK

Numerical Control Kernel: 数控系统部件，执行 → 零件程序，并控制机床的运动过程。

NRK

Numeric Robotic Kernel (NCK → 的操作系统)

NURBS

控制系统内部的运动控制和轨迹插补是基于 NURBS (**Non Uniform Rational B-Splines** 非均匀有理 B 样条) 进行的。因此, 在系统内部所有插补均采用相同的方法。

OEM

为机床制造商提供了各种不同的解决方案 (OEM 应用), 制造商可以自己设计操作界面或者在系统中开发专用的工艺功能。

PLC

Programmable Logic Control: → 可编程逻辑控制器。 -> 数控系统的组件: 用于调整机床控制逻辑。

PLC 编程

PLC 用软件 **STEP 7** 编程。编程软件 **STEP 7** 基于 **WINDOWS** 标准操作系统开发, 并包含创新的 **STEP 5** 编程功能。

PLC 可编程控制器

SINUMERIK 840D sl: 在 PLC 用户存储器中保存了 PLC 用户程序、用户数据与 PLC 基本程序。

R 参数

计算参数, → 零件程序编程人员可在程序中读写这些参数。

TOA 单元

每个 → **TOA** 区域可以包含多个 **TOA** 单元。**TOA** 单元的数量以最大有效的 → 通道数量为极限。一个 **TOA** 单元包括一个刀具数据模块和一个刀库数据模块。另外, 也可包含一个刀架数据模块 (选件)。

TOA 区域

TOA 区域包含所有刀具和刀库数据。默认设置下, **TOA** 区域即 → 通道。但通过机床数据可以确定, 多个通道组成一个 → **TOA** 单元, 以使这些通道也可使用通用的刀具管理数据。

V.24

用于数据输入/输出的串行接口。通过该接口可以装载和备份加工程序以及制造商和用户数据。

WinSCP

WinSCP 是一个可自由使用的开源程序，用于 Windows 的文件传输。

安全功能

系统中始终处于激活的监控功能，可以及早识别出 → CNC、→ PLC 和机床中的故障，从而排除一切对工件、刀具或者机床可能造成的损害。在故障发生时，加工过程会中断，驱动停止，故障原因被保存并作为报警显示。同时通知 PLC 有一报警。

保护区

→ 加工区内、不允许刀尖进入的一个三维空间。

报警

所有 → 提示信息和报警都包含纯文本说明、日期时间以及指出清除条件的符号，显示在机床操作面板上。报警和信息单独显示。

1. 零件程序中的报警和信息
报警和信息可以直接以纯文本的形式从零件程序中输出。
2. PLC 报警和信息
机床报警和信息可以直接以纯文本的形式从 PLC 程序中输出。而无需另外的功能块软件包。

备份电池

备份电池可实现 → CPU 中的 → 用户程序、特定数据区、标志位、定时器和计数器的掉电保持。

倍率

一种手动干预方法或编程方法，允许操作人员覆盖已编程的进给率或者转速，使加工速度与具体的工件和材料相匹配。

比例

是 → 框架的一个分量，可以改变某个轴的比例尺。

编程码

编程码是 → 零件程序编程语言中具有特定含义的字符和字符串。

编辑器

利用编辑器可以进行程序/文本/程序段的创建、修改、补充、合并和插入。

变量定义

变量定义包括确定数据类型和变量名。使用变量名可以调用该变量的值。

标准循环

标准循环适用于最常见的加工任务：

- 钻削/铣削
- 车削

在“程序”操作区“循环支持”菜单下列出了所有可使用的循环。选择了所需加工循环后，屏幕上会以纯文本的形式显示需要赋值的参数。

波特率

数据传输速度（位/秒）。

补偿表

控制点组成的一张表格。补偿表指出和参照轴的位置对应的补偿轴的补偿值。

补偿存储器

控制系统中的一个数据区，刀具补偿数据存储在其中。

补偿值

测量编码器实测出的位置与程序目标位置之间的差值/

补偿轴

设定值或者实际值可以通过补偿值进行修改的轴。

参考点

机床上的一点，→ 机床轴的测量系统以该点为基准。

参照轴

在计算补偿值需要考虑其设定值或实际值的轴。

操作界面

操作界面（BOF）是控制系统的显示屏幕。它带有水平软键和垂直软键。

插补补偿

插补补偿用于补偿加工工艺中必然存在的丝杠螺距误差和测量系统误差。

插补器

→ NCK 的逻辑单元，根据零件程序中目标位置确定轴的中间值。

成品轮廓

成品工件的轮廓。参见 → 毛坯件。

程序段搜索

在进行零件程序测试时或者在中断一个加工后，可以通过“程序段搜索”功能找到程序中的任意位置，在此位置加工可以启动或者继续。

程序级

在通道中启动的零件程序作为程序级 0（主程序级）上的 → 主程序运行。每个在主程序中调用的零件程序作为单独程序级 1 ... n 上的子程序运行。

程序块

程序块包含 → 零件程序的主程序和子程序。

从轮廓快速退刀

出现中断事件时，可以通过 **CNC** 加工程序触发一个运动，使刀具迅速离开当前正在加工的工件轮廓。此外还可以设定退刀角度和退刀行程。在快速退刀以后可以另外执行一个中断程序。

存档

读出文件和/或目录，存储到**外部**存储器设备中。

单位制：公制和英制

在加工程序中，位置值和螺距值可以用英制编程。不管程序中的尺寸指令(G70/G71)如何，系统始终采用基本单位制。

刀具

机床中进行加工的工具（诸如车刀、铣刀、钻头、激光…）

刀具半径补偿

为了可以直接编程所需的 → 工件轮廓，控制系统必须考虑所用刀具的半径，使刀具等距绕行编程的轮廓。（G41/G42）。

刀具补偿

计算轨迹时考虑到刀具尺寸。

刀沿半径补偿

在编程一个轮廓时，是从刀尖计算轮廓的。但是，这在实际加工过程中无法实现，因为使用的刀具会有一个弯曲半径，控制系统必须考虑该半径。此时，系统会使弯曲中心点偏移弯曲半径，从而等距绕行轮廓。

地址

地址用于标出特定操作数或操作数区域，例如：输入、输出等。

定位轴

在机床中执行辅助运动的轴。（例如刀库，托盘运输）。定位轴不与 → 轨迹轴进行插补。

定向刀具退回

RETTOOL: 当加工中断时（比如刀具折断），刀具可以根据编程指令按照指定方向后撤一段距离。

定向主轴准停

使工件主轴停止在一指定角度位置上，以便在某一确定位置进行辅助加工。

动态前馈控制

和加速度相关的动态前馈控制几乎可完全消除跟随误差导致的 → 轮廓误差。因此可获得高加工精度，即使是在 → 轨迹速度很高的情况下。前馈控制可以通过 → 零件程序逐轴选择或者撤销。

多项式插补

用多项式插补功能可以产生不同的曲线，如**线性函数、抛物线函数和幂函数**（SINUMERIK 840D sl）。

反比时间进给率

也可以编程一个程序段的轨迹行程所需要的时间（G93），而不用编程轴的进给速度。

分度轴

分度轴使工件或者刀具旋转到一个和分度头对应的角度位置。到达分度位置后，回转轴“就位”。

辅助程序段

由“N”开头的程序段，包含一个加工步骤的信息，比如一个位置数据。

辅助功能

在 → 零件程序中，使用辅助功能可以把机床制造商定义的 → 参数传送到 → PLC 中，并释放其所定义功能。

刚性攻丝

用此功能可以不带补偿衬套攻螺纹。通过插补，主轴作为回转轴和钻削轴将螺纹精确钻至最终钻深，比如盲孔螺纹（前提条件：主轴作为进给轴运行）。

高速数字量输入/输出

通过该数字量输入可以启动高速程序（中断程序）。通过该数字量输出端可以触发程序控制的高速开关功能。

工件

需由机床制造/加工的零件。

工件零点

工件零点构成了→工件坐标系的原点。它由与→机床零点的距离定义。

工件轮廓

待加工→工件的目标轮廓。

工件坐标系

以→工件零点为原点的坐标系。在工件坐标系中编程时，坐标和方向以工件坐标系为基准。

工作存储器

→ CPU 中的 RAM 存储器，在程序执行期间，处理器访问其上的用户程序。

工作区域限制

除限位开关之外，还可以使用工作区域限制功能对轴的行程进行限制。每根轴都可以使用一对值设定保护区。

公制单位

单位均为公制：比如：长度采用 mm（毫米），m（米）。

关键字

在 → 零件程序编程语言中具有一定含义、具有固定句法的指令字。

轨迹进给率

轨迹进给率作用于 → 轨迹轴。它是参与轨迹运动的 → 几何轴的合成进给率。

轨迹速度

最大可编程轨迹速度取决于输入精度。比如精度为 0.1 毫米，则可编程的最大轨迹速度为 1000 米/分钟。

轨迹轴

轨迹轴指 → 通道的所有加工轴，由 → 插补器控制，它们可以同时启动、加速、停止并同时达终点。

宏指令技术

包含有一系列指令，但名称只有一个。在程序中，一个名称就代表了这一系列指令。

回转轴

回转轴使工件或者刀具旋转到一个指定的角度位置。

机床固定点

由机床明确定义的点，比如：机床参考点。

机床控制面板

机床的控制面板有各个操作按键、旋钮开关等，以及各个显示单元如 LED。它们直接通过 PLC 对机床进行控制。

机床零点

机床固定点，所有测量系统均以此点为原点。

机床轴

在机床上实际存在的轴。

机床坐标系

以机床轴为基准的坐标系。

基本坐标系

一个直角坐标系，可转换为机床坐标系。

在 → 零件程序中使用基本坐标系中的轴名称。如果没有 → 有效的坐标转换，则它平行于 → 机床坐标系。不同点在于 → 轴名称。

极限转速

最大/最小（主轴）转速：在机床数据、→ PLC 数据或者 → 设定数据中可以限制主轴的最大转速。

极坐标

极坐标系指在一个平面中通过点到零点的距离、半径和一个固定轴之间的夹角来确定点位置的坐标系。

几何尺寸

→ 工件 → 在工件坐标系中的尺寸。

几何轴

几何轴构成一个 2 维或 3 维 → 工件坐标系，零件程序在该坐标系中编写工件的几何尺寸。

加工区

加工区是机床中刀尖可以进入的三维区域。参见 → 保护区

加工通道

采用通道结构可以进行并行运动，缩短非加工时间，比如在装料的同时可以进行加工。一条通道可以看作是一个独立的数控系统，可以译码、程序段预处理并进行插补。

加速度，带加加速度限制

为了在机床上获得最佳的加速方式，同时又要保护机械，在加工程序中可以在跃变式加速度和平缓式加速度之间进行转换。

间隙补偿

对机床上的机械间隙进行补偿，比如滚珠丝杠的反向间隙。每根轴可以分别输入间隙补偿。

接地

接地指设备上所有相互连接的不带电部件构成的整体，即使在出现故障时也不会出现危险的接触电压。

进给倍率

机床控制面板或者→ PLC 设定的、叠加在编程速度上的倍率值，以修改当前速度（0—200%） 另外，进给速度也可以在加工程序中通过一个百分比（1—200%）进行修改。

镜像

镜像使轮廓轴坐标值的符号相反。可以同时多个轴进行镜像。

绝对坐标

指出轴在某一个方向上相对于当前有效的坐标系零点的运动终点。参见 → 增量坐标。

可编程的工作区域限制

将刀具的运动空间限制在一个指定范围内。

可编程的框架

使用可编程的 → 框架可以在零件程序执行过程中动态地定义新的坐标系原点。框架可分为全新框架和附加在原有框架上的附加框架。

可编程逻辑控制器

可编程逻辑控制器(PLC)是电子控制器，其功能以程序的形式存储。因此，PLC 的结构和布线与控制系统的功能无关。PLC 具有计算机的结构，它由带存储器的 CPU（中央模块）、输入/输出模块和内部总线系统构成。输入/输出和编程语言以控制系统的需求为准。

快速移动

轴最快的运行速度。比如，可用于使刀具从静止状态运行到 → 工件轮廓或者从工件轮廓返回。快速移动速度可以根据不同机床在机床数据中设置。

框架

框架是一种运算规范，指出如何把一种直角坐标系转换到另一种直角坐标系。框架中包含几个部分 → 零点偏移、→ 旋转、→ 比例、→ 镜像。

连续路径运行

该方式可以在零件程序段分界处避免 → 轨迹轴急速停止，并尽可能保持相同的轨迹速度转到下一个程序段。

零点偏移

通过 → 框架相对于坐标系零点指定一个新的参考点。

1. 可设定
可为每个 CNC 轴设定不同数量的零点偏移。可通过 G 功能选择的偏移可以选择性地使用。
2. 外部
除了用于确定工件零点的所有偏移外，还可以通过手轮（DRF 偏移）或者由 PLC 叠加一个外部零点偏移。
3. 可编程
TRANS 指令可以为所有的轨迹轴和定位轴确定零点偏移。

零件程序

数控系统中由一系列指令构成、用于实现特定 → 工件的加工的程序。也就是说，在一个指定的 → 毛坯上进行指定的加工。

零件程序段

一行 → 零件程序，换行后结束。分为 → 主程序段和 → 辅助程序段。

零件程序管理

零件程序可以按照→工件管理。用户存储器的容量确定了需要管理的程序和数据的数量。每个文件（程序和数据）的名称可以最多为 24 个字符（字母和数字）。

轮廓

→ 工件的外边沿

轮廓监控

为确保轮廓精度，系统会监控跟随误差是否在定义的公差带内。比如，当驱动负载过高时，就可能导致跟随误差过大。在这种情况下会产生一个报警，轴停止运行。

螺旋线插补

螺旋线插补特别适用于利用成形铣刀简单地加工内螺纹和外螺纹以及铣削润滑槽。

螺旋线由两个运动组成：

- 平面中的回转运动
- 与此平面垂直的直线运动

毛坯

未经加工的工件。

名称

符合 DIN 66025 标准的“字”可补充变量标识/变量名（计算变量、系统变量和用户变量）、子程序名、关键字名和带多个地址字母的字。这些补充的名称和程序段“字”的含义一样。名称必须是唯一的。同一个名称不可以用于不同的对象。

模块

模块是指编程和程序执行时所需要的所有文件。

倾斜加工

通过倾斜加工可以方便地进行无法在机床坐标系中进行的钻削和铣削加工。

清零

在清零时，→ CPU 中以下的存储器将被清零：

- → 工作存储器
- → 装载存储器的读写区
- → 系统存储器
- → 备份存储器

曲率

轮廓的曲率 k 是轮廓点上该段圆弧半径 r 的倒数 ($k = 1/r$)。

驱动

属于 CNC 的组件，它根据 NC 设定执行转速控制和转矩控制。

软件限位开关

软件限位开关用于限制轴的移动范围，阻止滑块驶出硬件限位开关。每个轴可以给定 2 组数值，它们可以由 → PLC 分别激活。

软键

在屏幕上显示的按键，可以动态地根据当前的操作变化。这些功能键（软键）可以任意指定由软件规定的功能。

设定数据

设定数据确定机床的性能，按照系统软件定义的方法在系统中设定。

输入/输出模块

用于建立 CPU 和过程之间的联系。

输入/输出模块是：

- → 数字量输入/输出模块
- → 模拟量输入/输出模块
- → 模拟器模块

数据传输程序 PCIN

PCIN 是一种辅助程序，通过串行接口发送和接收 CNC 用户数据，如零件程序、刀具补偿等等。PCIN 程序可以在标准工业 PC 中的 MSDOS 下运行。

数据块

1. → PLC 的数据单元，可以访问 → HIGHSTEP 程序。
2. → NC 的数据单元：数据块包含全局用户数据的数据定义。数据可以在定义时直接初始化。

数据字

一个 → 数据块中两个字节大小的数据单位。

丝杠螺距误差补偿

控制系统根据保存的测量值补偿作进给运动的滚珠丝杠的机械误差。

速度控制

如果每个程序段轴的移动量非常小，可以使用预读功能(→ Look Ahead) 预读多个程序段来获得合理的移动速度。

通道

通道可以单独处理一个 → 零件程序，而不受其他通道的影响。一个通道仅控制指定给它的进给轴和主轴。不同通道的零件程序执行过程可以通过 → 同步功能进行协调。

同步

→ 零件程序中的指令，用于协调同一加工地点时不同 → 通道中的加工过程。

同步动作

1. 辅助功能输出
在工件加工期间，可以把工艺功能（→ 辅助功能）从 CNC 程序中输出到 PLC 中。通过这些辅助功能可以控制机床的附加设备，比如顶尖套筒、机械手、卡盘等等。
2. 快速辅助功能输出
如果开关功能对时间要求较高，可以缩短 → 辅助功能的应答时间，避免加工过程不必要的停顿。

同步轴

同步轴为 → 龙门轴，其设定位置始终由主动轴导出，因此两者同步运行。对于操作员和编程者而言，同步轴是“不存在的”。

同步轴

同步轴需要获知其行程。而几何轴需要获知其轨迹行程。

外部零点偏移

由 → PLC 给定的零点偏移。

网络

网络指通过 → 连接电缆相连的多个 S7-300 和其它终端设备，比如一台编程器。相连设备通过网络进行数据交换。

文本编辑器

参见 → 编辑器

系统变量

无需 → 零件程序程序员的工作，已经存在的变量。它由数据类型和变量名称定义，变量名称前有符号\$。参见 → 用户自定义变量。

系统存储器

系统存储器是 CPU 中的一个存储器，其中保存有：

- 操作系统所需要的数据
- 操作数：定时器、计数器和标志位

线性插补

刀具以直线运行到目标，同时进行工件的加工。

线性轴

与回转轴相反，线性轴指按直线运行的轴。

象限误差补偿

由导轨上不断变化的摩擦阻力导致的过象限轮廓误差可以最大程度地用象限误差补偿消除。象限误差补偿的参数可以通过圆弧测试确定。

信息

零件程序中编入的所有信息以及系统检测出的 → 报警均在操作面板上显示，带日期和时间，并有相应的清除条件符号。报警和信息单独显示。

旋转

→ 框架的一个分量，定义坐标系的旋转角度。

循环

受保护的子程序，用于执行 → 工件上反复出现的加工过程。

样条插补

通过样条插补，控制系统只需少数几个目标轮廓控制点便可生成一条光滑的曲线。

异步子程序

指可以通过一个中断信号（比如信号“高速 NC 输入”）启动的、与当前程序状态异步执行（即无关）的零件程序。

引导启动

上电后装载系统程序。

英制单位

单位制，距离采用“英寸”及其下级小数单位。

用户程序

可编程控制器 S7-300 中用 STEP 7 语言编写的用户程序。用户程序为模块化结构，由各个模块构成。

基本的模块类型有：

- 代码模块
该模块含有 STEP 7 指令。
- 数据模块
该模块包含有用于 STEP 7 程序的常量和变量。

用户存储器

包含了所有的程序和数据，比如零件程序、子程序、注释、刀具补偿、零点偏移、框架以及通道和程序用户数据。

用户自定义变量

用户可以在 → 零件程序或者数据块（全局用户数据）中自由使用自定义变量。一个定义通常含有数据类型和变量名称。参见 → 系统变量。

预读功能

利用功能**预读**可以预读一定数量的程序段而优化加工速度。

预符合

如果轨迹行程几乎快接近设定三角形的终端位置，则进行程序段转换。

圆弧插补

在轮廓上两个固定点之间，→ 刀具以给定的进给率沿圆弧运行，加工工件。

钥匙开关

→ 机床控制面板上的钥匙开关占据 4 个位置，它们由控制系统的操作系统指定相应的功能。钥匙开关有三把不同颜色的钥匙，钥匙可以在指定位置插拔。

运行到固定点

机床中有一些固定点，比如换刀点、装料点、托盘更换点等等。这些点的坐标在控制系统中定义。控制系统使对应轴移动到这些固定点，如果可能，→ 快速移动轴。

运行范围

线性轴最大允许的运行范围可以达到 ± 9 位。绝对值取决于所选择的输入精度、位置控制精度以及单位制（英制或者公制）。

运行方式

SINUMERIK 控制系统的运行控制方式。定义的运行方式有 → Jog, → MDA, → 自动。

运行方式组

加工工艺上关联的进给轴和主轴可以归为一个工作方式组（BAG）。一个 BAG 的进给轴和主轴可以由一个或多个 → 通道控制。同一个工作方式组中的通道均有相同的 → 工作方式。

增量坐标

也称为相对坐标：指出轴相当于起点需要移动的行程以及方向。参见 → 绝对坐标

增量坐标

以增量数（步数）指定行程。步数可以作为 → 设定数据保存或者通过相应的按键（10、100、1000 和 10000）选取。

诊断

1. 系统操作区
2. 控制系统不仅有自诊断程序，还可以为维修提供辅助测试。状态、报警和服务信息

中断程序

中断程序是一种特殊 → 子程序，由程序执行过程中的外部事件（外部信号）启动。正在执行的程序段终止，系统自动保存轴的中断位置。

中间程序段

带 → 刀具补偿（G41/G42）的运动可以由一定数量的中间程序段（位于补偿级的程序段，没有轴运动）中断，此时刀具补偿仍能进行正确计算。控制系统能预读的最大中间程序段数量可以通过系统参数设定。

轴

数控系统中的轴根据其功能可以分为：

- 轴： 进行插补的轨迹轴
- 辅助轴： 不进行插补的进给轴和定位轴，具有单轴进给率。辅助轴不参与加工，而是用于执行辅助运动，比如移动机械手、转动刀库等。

轴地址

参见 → 轴名称

轴名称

为了明确标识控制系统的所有通道轴和 → 机床轴，每根轴有一个通道范围内或系统范围内唯一的名称。→ 几何轴名为 X 轴、Y 轴和 Z 轴。围绕几何轴旋转的回转轴名为 A 轴、B 轴和 C 轴。

主程序

之前零件程序固定划分为主程序和 → 子程序，因此有“主程序”的概念。在如今的 SINUMERIK NC 语言中，这种固定划分已不再存在。原则上，每个零件程序都可以在通道中选择和启动。其在 → 程序级 0（主程序级）上运行。在主程序中，其他的零件程序或 → 循环都可以作为子程序来调用。

主程序段

冒号开头的程序段，包含了 → 零件程序中启动加工流程所需的所有数据。

主动轴

主动轴为 → 龙门轴，对于操作员和编程者而言，它可以象普通轴一样控制。

装载存储器

在 → PLC 的 CPU314 中，装载存储器就是 → 工作存储器。

准停

使轴准确地移动到指定位置，或者以非常慢的速度移动到指定位置。为缩短移动时间，可以为快速移动运动和进给运动定义 → 准停界限。

准停界限

如果所有的轨迹轴均到达准停界限，则控制系统会认为轴已经精确到达目标。随后 → 零件程序会切换程序段。

子程序

之前零件程序固定划分为主程序和子程序，因此有“子程序”的概念。在如今的 SINUMERIK NC 语言中，这种固定划分已不再存在。原则上，每个零件程序或 → 循环在另一个零件程序中都可作为子程序调用。其在下一个 → 程序级 (x+1) (子程序级 (x+1)) 上运行。

自动方式

控制系统的运行方式（程序段连续运行，符合 DIN 标准）：控制系统中的运行方式，这种方式下选择 → 零件程序并连续执行程序。

坐标系

参见 → 机床坐标系, → 工件坐标系

坐标转换

轴的附加或者绝对零点偏移。

索引

操作单元切换锁, 905
操作单元请求进入“主动”操作模式, 905
定位轴
 空运行进给率, 675
动态间隙, 251
接口信号
 定向轴的正向和负向运行请求, 927
 定向轴的正向和负向运行指令, 928
 几何轴(1, 2, 3)的生效机床功能, 925, 926
 几何轴的正向和负向运行指令, 924, 925
 轮廓手轮(1到3)生效, 922, 923
 生效的几何轴手轮(1到3), 923, 924, 926, 927
 手轮叠加生效, 922
 正向和负向几何轴运行请求, 923, 924, 926, 927

\$

\$A_DP_IN_CONF, 68
\$A_DP_IN_STATE, 68
\$A_DP_IN_VALID, 68
\$A_DP_OUT_CONF, 68
\$A_DP_OUT_STATE, 68
\$A_DP_OUT_VALID, 68
\$A_DPx_IN, 66
\$A_DPx_OUT, 66
\$A_IN, 33, 37
\$A_INA, 33
\$A_INCO, 53
\$A_OUT, 33, 39
\$A_OUT[n], 584
\$A_OUTA, 33, 47
\$A_PBx_IN, 57
\$A_PBx_OUT, 57
\$AA_ACT_INDEX_AX_POS_NO, 819
\$AA_COUP_ACT, 765
\$AA_COUP_OFFS, 766
\$AA_ENC_COMP, 257
\$AA_ENC_COMP_IS_MODULO, 259
\$AA_ENC_COMP_MAX, 259
\$AA_ENC_COMP_MIN, 258
\$AA_ENC_COMP_STEP, 257
\$AA_FIX_POINT_ACT, 203
\$AA_FIX_POINT_SELECTED, 203
\$AA_G0MODE, 648
\$AA_ISTEST, 335

\$AA_MOTEND, 665
\$AA_PROG_INDEX_AX_POS_NO, 819
\$AC_AXCTSWA, 120
\$AC_AXCTSWE, 120
\$AC_ISTEST, 335
\$AC_RETPOINT, 732
\$AN_AXCTAS, 120
\$AN_AXCTSWA, 120
\$AN_CEC, 266
\$AN_CEC_DIRECTION, 267
\$AN_CEC_INPUT_AXIS, 266
\$AN_CEC_IS_MODULO, 267
\$AN_CEC_MAX, 266
\$AN_CEC_MIN, 266
\$AN_CEC_MULT_BY_TABLE, 267
\$AN_CEC_OUTPUT_AXIS, 266
\$AN_CEC_STEP, 266
\$AN_LAI_AX_IS_AXCTAX, 120
\$AN_LAI_AX_IS_LEADLINKAX, 120
\$AN_LAI_AX_IS_LINKAX, 120
\$AN_LAI_AX_TO_IPO_NC_CHANAX, 120
\$AN_LAI_AX_TO_MACHAX, 120
\$AN_REBOOT_DELAY_TIME, 318
\$P_COUP_OFFS, 766
\$P_ISTEST, 335
\$VA_COUP_OFFS, 766

“

“行程-时间”挡块, 588

1

1 维
 设定值给定 (\$AC_MEAS_TYPE = 19), 523
103lp, 118

2

2 维
 设定值给定 (\$AC_MEAS_TYPE = 20), 525

3

3 维
 设定值给定 (\$AC_MEAS_TYPE = 21), 526
32402, 786

360°模数, 726

39lp, 118

3D 测头, 486

4

42lp, 118

A

ACN, 729

ACP, 729

ASCALE, 736

ATRANS, 736

AxAlarm, 954

AXCTSWE, 118

AXCTSWE, 118

AXCTSWED, 118

AxReset, 953

AxResume, 954

AxStop 生效, 955

AXTOCHAN, 357

C

CC-Binding, 69

CLEAR, 326

CLEAR, 327

COARSE, 659

Continuous Dressing, 861

CT, 118

D

DB10

DBB0, 38, 886

DBB1, 38, 39, 887

DBB100, 911, 912

DBB101, 911, 912

DBB102, 911, 912

DBB122, 886

DBB122 ..., 38

DBB123, 887

DBB123 ..., 38, 39

DBB124, 886

DBB125, 887

DBB126, 886

DBB127, 887

DBB128, 886

DBB129, 887

DBB130, 887

DBB130 ..., 41

DBB131, 887, 888

DBB131 ..., 40

DBB132, 888

DBB132 ..., 40

DBB133, 889

DBB133 ..., 41

DBB134, 887

DBB135, 887, 888

DBB136, 888

DBB137, 889

DBB138, 887

DBB139, 887, 888

DBB140, 888

DBB141, 889

DBB142, 887

DBB143, 887, 888

DBB144, 888

DBB145, 889

DBB146, 45, 889

DBB147, 46, 889, 890

DBB148-163, 890

DBB166, 49, 890, 891

DBB167, 49, 891

DBB168, 49, 891, 892

DBB170-185, 892

DBB186 ..., 41

DBB186-189, 895

DBB190-193, 895

DBB194-209, 896

DBB210-225, 49, 896

DBB4, 41, 887

DBB5, 40, 887, 888

DBB6, 40, 888

DBB60, 895

DBB64, 41, 895

DBB7, 41, 889

DBB97, 910

DBB98, 910

DBB99, 910

DBW148-162, 46

DBW170 ..., 49

DBW194-208, 45

DBX100.0-4, 168

DBX100.6, 168, 912, 913

DBX100.7, 168, 913

DBX101.0-4, 168

DBX101.6, 168, 912, 913

DBX101.7, 168, 913

DBX102.0-4, 168

DBX102.6, 168, 912, 913

- DBX102.7, 168, 913
- DBX107.0, 945
- DBX107.1, 945
- DBX107.6, 908
- DBX108.7, 317
- DBX110.0 - 113.7, 583
- DBX110.0-113.7, 947
- DBX114.0 - 117.7, 583
- DBX114.0-117.7, 948
- DBX97.0-3, 168
- DBX98.0-3, 168
- DBX99.0-3, 168
- DB10, ...
 - DBX107.0, 471
 - DBX107.1, 471
- DB11
 - DB6.2, 156
 - DBX 6.3, 317
 - DBX186.2, 156
 - DBX26.2, 156
- DB11, ...
 - DBX($n*20+6$).2, 207
- DB19
 - DBX0.7, 209
 - DBX20.7, 209
- DB21, ...
 - DBX0.3, 190, 194, 684, 914
 - DBX0.6, 190
 - DBX100.5, 930
 - DBX101.5, 930
 - DBX102.5, 930
 - DBX12.0-2, 914, 915, 919, 920
 - DBX12.0-5, 918
 - DBX12.3, 600
 - DBX12.4, 915
 - DBX12.5, 916
 - DBX12.6, 917, 918
 - DBX12.7, 917, 918
 - DBX13.6, 919
 - DBX15.0, 171
 - DBX15.0, 19.0, 23.0, 919
 - DBX16.0-2, 914, 915, 919, 920
 - DBX16.0-5, 918
 - DBX16.4, 915
 - DBX16.5, 916
 - DBX16.6, 917, 918
 - DBX16.7, 917, 918
 - DBX17.6, 919
 - DBX19.0, 171
 - DBX20.0-2, 914, 915, 919, 920
 - DBX20.0-5, 918
 - DBX20.4, 915
 - DBX20.5, 916
 - DBX20.6, 917, 918
 - DBX20.7, 917, 918
 - DBX21.6, 919
 - DBX23.0, 171
 - DBX24.3, 194, 921, 922
 - DBX3.0, 599, 950
 - DBX3.1, 599, 950
 - DBX3.2, 599, 950
 - DBX3.3, 951
 - DBX3.4, 599, 951
 - DBX3.5, 951
 - DBX30.0, 930
 - DBX30.0-2, 192
 - DBX30.1, 930
 - DBX30.2, 930
 - DBX30.3, 192, 931
 - DBX30.4, 193, 931
 - DBX31.5, 171, 931
 - DBX323.0, 171
 - DBX323.0, 327.0, 331.0, 920, 921
 - DBX327.0, 171
 - DBX33.3, 186, 922
 - DBX33.6, 944
 - DBX331.0, 171
 - DBX332.4, 927
 - DBX332.5, 927
 - DBX332.6, 928
 - DBX332.7, 928
 - DBX335.0, 171
 - DBX335.0, 339.0, 343.0, 929
 - DBX336.4, 927
 - DBX336.5, 927
 - DBX336.6, 928
 - DBX336.7, 928
 - DBX339.0, 171
 - DBX340.4, 927
 - DBX340.5, 927
 - DBX340.6, 928
 - DBX340.7, 928
 - DBX343.0, 171
 - DBX37.0, 931, 932
 - DBX37.0-2, 922, 923
 - DBX37.1, 931, 932
 - DBX37.2, 931, 932
 - DBX377.4, 209, 211
 - DBX377.5, 208, 211
 - DBX38.0, 600, 952
 - DBX38.1, 599, 952
 - DBX39.5, 171, 932
 - DBX40.5, 169
 - DBX40.6, 924, 925

- DBX40.7, 160, 170, 924, 925
DBX41.0-6, 925, 926
DBX43.0, 171
DBX43.0, 49.0, 55.0, 926
DBX46.5, 169
DBX46.6, 924, 925
DBX46.7, 170, 924, 925
DBX47.0-6, 925, 926
DBX49.0, 171
DBX52.5, 169
DBX52.6, 924, 925
DBX52.7, 170, 924, 925
DBX53.0-6, 925, 926
DBX55.0, 171
DBX67.0, 941
DBX7.1, 778
DBX7.4, 778
DB21, ...
 DBX 36.5, 317
 DBX12.0-2, 167, 185
 DBX13.6, 160
 DBX16.0-2, 167, 185
 DBX20.0-2, 167, 185
 DBX320.0-2, 167
 DBX324.0-2, 167
 DBX328.0-2, 167
 DBX332.4, 169
 DBX332.5, 169
 DBX332.6, 170
 DBX332.7, 170
 DBX336.4, 169
 DBX336.5, 169
 DBX336.6, 170
 DBX336.7, 170
 DBX340.4, 169
 DBX340.5, 169
 DBX340.6, 170
 DBX340.7, 170
 DBX35.7, 207
 DBX377.5, 207
 DBX40.4, 169
 DBX40.6, 160, 170
 DBX41.6, 160
 DBX46.4, 169
 DBX46.6, 170
 DBX52.4, 169
 DBX52.6, 170
DB31, ...
 DBB19, 774, 782
 DBX1.3, 775
 DBX1.4, 772, 776
 DBX1.5, 776
 DBX1.6, 776
 DBX1.7, 157
 DBX100.2, 703
 DBX102.0, 252
 DBX13.0-2, 198, 199
 DBX14.0, 334
 DBX16.4, 777, 782
 DBX16.5, 777, 782
 DBX16.7, 777, 782
 DBX17.6, 782
 DBX2.0, 582
 DBX2.1, 772, 775
 DBX2.2, 164, 184, 776, 782
 DBX25.0, 252
 DBX26.4, 780
 DBX28.0, 703
 DBX28.1, 651
 DBX28.2, 651
 DBX28.3, 704
 DBX28.4, 704
 DBX28.5, 689
 DBX28.6, 651, 689
 DBX28.7, 690
 DBX29.5, 772
 DBX31.4, 757, 766, 772, 773, 776
 DBX31.5, 755, 773
 DBX4.3, 756, 776
 DBX4.6, 778
 DBX4.7, 778
 DBX5.0 - 5.5, 163
 DBX6.2, 651
 DBX60.4, 753, 812, 813
 DBX60.4/5, 244
 DBX60.5, 753, 812, 813
 DBX60.6, 652
 DBX60.7, 652
 DBX60.7 或 DBX60.6, 228
 DBX61.3, 772
 DBX62.0, 583
 DBX62.1, 186, 189
 DBX62.3, 471
 DBX63.0, 650, 653
 DBX63.1, 650
 DBX63.2, 650, 652, 653
 DBX64.5, 169
 DBX64.7, 170, 652
 DBX65.0 - 65.5, 163
 DBX67.0, 171
 DBX7.0, 171
 DBX74.4, 724
 DBX75.0-2, 198
 DBX75.3-5, 198

- DBX76.6, 812, 813, 815
 - DBX83.1, 751
 - DBX83.5, 781
 - DBX83.7, 782
 - DBX84.4, 746
 - DBX98.0, 756, 758, 773
 - DBX98.1, 756, 758, 773
 - DBX98.4, 750, 753, 782
 - DBX99.0, 748
 - DBX99.1, 748
 - DBX99.4, 773
 - DB31, ...
 - DBX 61.2, 317
 - DBX4.0-2, 167, 185
 - DBX64.4, 169
 - DBX64.6, 170
 - DB31, ...
 - DBB0, 953
 - DBB68, 943, 944
 - DBB78-81, 956
 - DBB8, 942, 943
 - DBX100.2, 959
 - DBX100.3, 959
 - DBX100.4, 959
 - DBX100.5, 959, 960
 - DBX100.6, 960
 - DBX100.7, 960
 - DBX104.0 - 7, 960
 - DBX12.4, 724, 732, 961
 - DBX13.0-2, 937
 - DBX2.0, 948
 - DBX2.2, 953
 - DBX28.0, 957
 - DBX28.1, 953
 - DBX28.2, 954
 - DBX28.3, 957
 - DBX28.4, 957, 958
 - DBX28.5, 958
 - DBX28.6, 958
 - DBX28.7, 958
 - DBX31.5, 962
 - DBX4.0-2, 932, 933
 - DBX4.4, 933
 - DBX4.5, 933, 934
 - DBX4.6, 935, 936
 - DBX4.7, 935, 936
 - DBX5.0-5, 936
 - DBX5.6, 937
 - DBX60.1, 909
 - DBX61.1, 954
 - DBX61.2, 909
 - DBX62.0, 949
 - DBX62.1, 938
 - DBX62.3, 945
 - DBX62.7, 909
 - DBX63.0, 955
 - DBX63.1, 955
 - DBX63.2, 955
 - DBX64.0-2, 938
 - DBX64.4, 939
 - DBX64.5, 939
 - DBX64.6, 940
 - DBX64.7, 940
 - DBX65.0-6, 941
 - DBX7.0, 937
 - DBX74.4, 732, 961
 - DBX75.0-2, 941
 - DBX75.3-5, 942
 - DBX76.5, 956
 - DBX76.6, 966, 967
 - DBX83.3, 967
 - DBX83.6, 968
 - DBX84.1, 968
 - DBX84.4, 962
 - DBX98.0, 963
 - DBX98.1, 963
 - DBX98.2, 963, 964
 - DBX98.4, 964
 - DBX99.0, 964, 965
 - DBX99.1, 965
 - DRF, 193, 831
- F**
- FC18, 669
 - FIFO 变量, 474
 - FINEA, 659
- G**
- G75, 197
 - GET, 345
 - GETD, 346
 - GWPS, 875
 - 在所有运行方式中, 877
- I**
- I/O 区域, 63
 - INCH 或 METRISCH 单位, 492
 - INIT, 326
 - IPOBRKA, 659
 - IPOENDA, 659

J

JOG, 156, 230, 413
运行至固定点, 197
JOG 模式下的特点, 399

L

LEC, 256

M

MD10010, 325
MD10050, 93
MD10061, 93
MD10070, 93
MD10071, 93
MD10088, 318
MD10185, 94
MD10200, 314
MD10210, 314, 737
MD10260, 131, 259, 268, 580
MD10270, 580, 816
MD10300, 34
MD10310, 34
MD10320, 36, 46
MD10330, 36, 50
MD10350, 34, 42
MD10360, 34, 42
MD10361, 43
MD10362, 35
MD10364, 35
MD10366, 35
MD10368, 35
MD10394, 58
MD10395, 58
MD10396, 58
MD10397, 58
MD10398, 58
MD10399, 59
MD10450, 581
MD10460, 581, 589
MD10461, 581, 589
MD10470, 584
MD10471, 584
MD10472, 584
MD10473, 584
MD1048, 586
MD10480, 585, 587
MD10485, 578, 585, 587

MD10500, 64
MD10501, 64
MD10502, 64
MD10510, 64
MD10511, 64
MD10512, 65, 69
MD10530, 53
MD10531, 54
MD10540, 54
MD10541, 54
MD10720, 206
MD10721, 206
MD10722, 350, 352, 671
MD10735, 200, 207, 209
MD10900, 816, 827
MD10910, 816, 827
MD10920, 816, 827
MD10930, 816, 827
MD10940, 819, 821
MD11300, 165
MD11310, 175, 176
MD11320, 166, 173
MD11322, 191
MD11324, 174
MD11330, 163, 173, 186
MD11346, 173, 191, 200
MD11350, 215, 216, 218
MD11351, 215, 216, 218
MD11352, 215, 216, 218
MD11353, 218
MD11410, 317
MD11450, 609
MD12701, 115, 144
MD12702, 115, 144
MD12703, 115, 144
MD12704, 115, 144
MD12705, 115, 144
MD12706, 115, 144
MD12707, 115, 144
MD12708, 115, 144
MD12709, 115, 144
MD12710, 115, 144
MD12711, 115, 144
MD12712, 115, 144
MD12713, 115, 144
MD12714, 115, 144
MD12715, 115, 144
MD12716, 115, 144
MD12717, 115
MD12750, 111
MD13211, 475
MD17900, 174

- MD18000, 94
MD18050, 803
MD18060, 800, 801
MD18096, 846, 849, 852
MD18100, 851
MD18210, 803
MD18230, 799
MD18351, 534
MD18352, 799, 801
MD18353, 800, 801
MD18600, 482
MD18720, 84
MD18960, 656
MD20050, 723
MD20070, 339, 340, 751
MD20100, 231, 493
MD20110, 131, 210, 339, 624, 877
MD20112, 339
MD20120, 877
MD20130, 877
MD20150, 210, 493, 597
MD20151, 210
MD20254, 870
MD20350, 850, 871
MD20360, 493
MD20390, 246
MD20610, 870
MD20620, 174
MD20621, 174
MD20624, 176, 181
MD20700, 211
MD20730, 648
MD20750, 647
MD21106, 427
MD21150, 738
MD21158, 226
MD21159, 227
MD21166, 226
MD21168, 226
MD21220, 36, 55
MD21300, 748, 784
MD21310, 784
MD21320, 764, 784
MD21330, 784
MD21340, 749, 784
MD22550, 839
MD22560, 839
MD24120, 428
MD26000, 606
MD26002, 606
MD26004, 606
MD26006, 606
MD26010, 613, 618
MD26014, 612, 613, 622
MD26016, 618
MD26018, 597, 606
MD26020, 600, 607
MD28264, 474
MD30300, 721, 732, 737, 817
MD30310, 722, 724, 726, 727, 729, 733, 737, 817
MD30320, 724, 726, 729, 737, 828
MD30330, 729, 819
MD30340, 727, 729
MD30455, 757, 770, 786, 787
MD30460, 670, 674
MD30500, 815, 824, 825, 827
MD30503, 728
MD30505, 825
MD30550, 339, 341, 344, 346, 751
MD30552, 338, 339, 346, 645
MD30600, 200
MD31090, 163, 166, 174, 186, 191
MD32000, 187, 647
MD32010, 157
MD32020, 157, 175, 228
MD32040, 157, 176, 230, 656, 814
MD32050, 157, 176, 230, 656, 814
MD32060, 189, 645, 655, 671
MD32074, 356
MD32080, 174
MD32084, 177
MD32090, 194
MD32200, 785
MD32300, 229, 656
MD32301, 158
MD32400, 786
MD32410, 786
MD32420, 158, 648, 786
MD32430, 648, 657, 786
MD32431, 657
MD32436, 158
MD32450, 250
MD32452, 250
MD32456, 252
MD32457, 252
MD32490, 306, 313
MD32500, 306, 308, 313
MD32510, 306, 313
MD32520, 307, 308, 314
MD32530, 314
MD32540, 307, 308
MD32550, 314
MD32560, 314
MD32570, 314

MD32610, 299, 785
MD32620, 297, 770, 785
MD32630, 297
MD32650, 301, 785
MD32700, 254, 256
MD32710, 254, 263, 306
MD32711, 268
MD32720, 268
MD32730, 268
MD32750, 245, 247, 306
MD32760, 246
MD32800, 301, 785
MD32810, 299, 785
MD32900, 302
MD32910, 302, 786
MD34080, 753, 788
MD34090, 753, 788
MD34100, 753, 788
MD35000, 748, 751
MD35032, 877, 878
MD35040, 878
MD35220, 787
MD35230, 787
MD35242, 787
MD36100, 733
MD36110, 733
MD36500, 251
MD36610, 317
MD36620, 317
MD37200, 760, 785
MD37210, 760, 785
MD37220, 760, 785
MD37230, 760, 788
MD37500, 654, 655
MD37510, 653, 654
MD37511, 654
MD38000, 257
MD7200, 788
MEAS、MEAW 指令, 469
MEASA、MEAWA、MEAC 指令, 471
MSEC, 256

N

NCK 模拟量输出的改值位, 890, 891
NCK 模拟量输出的设定值, 896
NCK 模拟量输出的写值位, 891
NCK 模拟量输入的实际值, 896
NCK 模拟量输入的写值位, 889, 890
NCK 数字量输出的改值位, 887, 888
NCK 数字量输出的设定值, 895
NCK 数字量输出的写值位, 889

NCK 数字量输入的实际值, 895
NCK 作为定位轴来处理该轴, 956
NCU-Link, 285
NCU-Link 生效, 908
NCU-Link 轴生效, 909

O

OPI 切换锁, 905
OS, 686
OSB, 688
OSCILL, 694
OSCTRL, 687, 688
OSE, 688
OSNSC, 688
OSP, 686
OST, 687

P

p0680, 475
p0684, 475
p0922, 475
PLC 给出的 NCK 模拟量输出的设置值, 892
PLC 给出的 NCK 模拟量输入的设置值, 890
PLC 给出的 NCK 数字量输出的设置值, 888
PLC 控制轴, 955, 958
PLC 信息显示, 471, 476
PLC 轴, 338, 349, 669
 固定指定给 PLC 的轴, 670
 通过 FC18 启动, 672
 只由 PLC 控制的轴, 670
POSP, 694
PROFIBUS I/O, 62
PTP/CP 切换
 JOG 下的运行方式切换, 426
PUNCHACC, 608

R

RELEASE, 344
RTLIOF, 648
RTLION, 648

S

SD41010, 163, 173, 186
SD41040, 812
SD41050, 162, 812
SD41100, 157, 176, 230, 656, 725, 813

SD41110, 157, 175, 228
SD41120, 157, 230
SD41130, 157, 175, 738
SD41200, 227
SD41300, 263
SD41500, 579
SD41500 - 41507, 817
SD41501, 579
SD41502, 579
SD41503, 579
SD41504, 579
SD41505, 579
SD41506, 579
SD41507, 579
SD41520, 582, 589
SD41521, 582, 589
SD41522, 582
SD41523, 582
SD41524, 582
SD41525, 582
SD41526, 582, 589
SD41527, 582, 589
SD41600, 55
SD41601, 55
SD42100, 190, 675, 683
SD42101, 675, 683
SD42300, 748, 784
SD42400, 604
SD42402, 597, 607
SD42404, 607
SD42600, 176, 656, 660, 814
SD42650, 428
SD43300, 176, 656, 660, 814
SD43400, 732
SD43410, 732
SD43600, 668
SD43770, 688
SD43790, 688
SD43900, 245
SD43910, 245
SD43920, 245, 247
SETM, 327, 330
START, 326

T

TRAANG
 简要说明, 369
 限制, 406
TRAANG 坐标转换, 441
TRACYL, 368
 数量, 390

轴映射, 394

TRACYL 坐标转换, 440
TRACYL_BAE_TOOL_t, 396
TRACYL_ROT_AX_OFFSET_t, 395
TRACYL_Rot_Sign_IS_PLUS_t, 395
TRANS, 736
TRANSMIT, 367
TRANSMIT 坐标转换, 440
TRANSMIT_ROT_AX_OFFSET_t, 373

W

WAITE, 326
WAITM, 326
WAITMC, 327, 328
WAITP, 659
 往复轴, 694

X

x 棱边 (\$AC_MEAS_TYPE = 1), 494
x 棱边 (\$AC_MEAS_TYPE = 3), 498

Y

y 棱边 (\$AC_MEAS_TYPE = 2), 497

报

报文选择, 475

被

被动文件系统, 796

本

本地 NCU, 104

比

比较器输入, 33, 53

边

边界条件, 413, 421

编

编程方式, 752
编码位置, 814
编写关节位置
STAT 地址, 423

变

变量接口, 478
变形
温度影响造成的, 242

表

表
补偿, 253, 254

并

并行定位轴
由 PLC 启动, 671

补

补偿
丝杠螺距误差, 256
测量系统误差, 256
插补, 253
垂度, 261
跟随误差, 297
角度误差, 261

不

不连续写入在线刀具补偿, 868

操

操作单元的主动/被动操作模式, 906

槽

槽壁补偿, 368

测

测量

槽 (\$AC_MEAS_TYPE = 12), 507
测量过程中的运行速度, 556
测量精度, 556
测量信号延时, 556
隔断 (\$AC_MEAS_TYPE = 13), 510
平面中的角度 (\$AC_MEAS_TYPE = 17), 516
斜边 (\$AC_MEAS_TYPE = 16), 514
延时补偿, 556
轴 (\$AC_MEAS_TYPE = 9), 506
钻孔 (\$AC_MEAS_TYPE = 8), 503
最大运行速度, 556

测量刀具直径 (\$AC_MEAS_TYPE = 11), 540

测量方法

用于测定矩形 (\$AC_MEAS_TYPE = 25), 532
用于恢复数据管理框架的赋值 (\$AC_MEAS_TYPE = 27), 534
用于将数据管理框架以当前的赋值备份至文件 (\$AC_MEAS_TYPE = 26), 533
用于为生效平面或所选平面设定叠加旋转 (\$AC_MEAS_TYPE = 28), 535
用于位置的坐标转换 (\$AC_MEAS_TYPE = 24), 528

测量拐角 C1, 501

测量过程, 556

测量接口

输出值, 488
输入值, 479
诊断, 494

测量结果

MEAC, 476
MEAS、MEAW, 470
MEASA、MEAWA, 475

测量模式, 473

测量输入参数, 485

测量循环, 489

测量状态, 945

测头

类型, 468
测头被操作, 945
测头功能测试, 557
测头功能测试示例, 557

插

插补

G0 下, 646
非线性, 647

线性, 646
插补过程, 676
插补器
 轨迹, 646
 轴, 646

差

差速, 781

长

长按模式, 161, 163

车

车床 X 轴, 231

程

程序
 协调, 326
程序段切换
 定位轴类型 1, 662
 定位轴类型 2, 663
程序段搜索, 609

冲

冲程释放有效, 952
冲程未运行, 951
冲程延时, 951

触

触发事件, 473

垂

垂度补偿, 285

从

从外部触发往复轴换向, 957

粗

粗同步, 963

扩展功能
功能手册, 03/2013, 6FC5397-1BP40-3RA1

存

存储器结构, 795

达

达到同步, 756

带

带槽壁补偿的编程, 447

单

单程序段
 定位轴类型 1, 675
 定位轴类型 2, 675
 定位轴类型 3, 675
单个坐标转换, 452
单向测头, 469
单轴
 NC 控制的扩展回退, 654
 NC 控制的扩展停止, 653
 应用, 651
 由 PLC 控制轴, 649
单轴进给倍率/主轴倍率, 953

挡

挡块
 对, 569
 范围, 569
 位置, 579
 信号, 569
挡块生效, 949
挡块信号
 单独输出, 570
 挡块信号定时输出, 585
 负, 583
 关联输出, 575
 硬件指定, 584
 正, 583

刀

刀具测量, 536
 启用各自参考点的两把车刀, 544

刀具长度

(\$AC_MEAS_TYPE = 10), 538

通过标记位置或当前位置测量 (\$AC_MEAS_TYPE = 23), 543

通过放大镜功能测量 (\$AC_MEAS_TYPE = 22), 541

刀具零点位置, 396

点

点按模式, 161, 163

叠

叠加运动, 964

定

定位轴, 640

并行, 645, 669

刀具补偿, 660

类型, 643

通道专用信号, 670

轴类型, 643

轴专用信号, 670

最大数量, 658

定位轴动态响应, 656

定向转换, 439

定义几何轴, 437

动

动态 NC 存储器, 795

动态响应

自适应, 302

动态用户存储器, 803

读

读取偏移, 757

独

独立的副主轴插补器, 747

端

端面齿, 824

断

断续运行, 162

对

对 HMI 操作的影响, 412

对刀, 477

对停止状态的应答, 955

多

多向测头 (3D), 468

反

反向间隙, 250

非

非坐标转换专用, 442

分

分度位置

数量, 816

分度位置表, 815

分度轴

编程, 818

编码位置, 818

单位制, 816

调试, 827

回参考点, 831

框架, 822

手轮, 813

分度轴就位, 966, 967

封

封锁冲程, 950

负

负

挡块, 569

负挡块信号 1-32, 947

复

复位, 421

副

副主轴

重新同步, 773

副主轴插补器, 747

副主轴有效, 965

该

该轴出现停止报警, 954

跟

跟随误差, 296

更

更改对应关系, 438

工

工件测量, 478

工作区域限制, 229

固

固定点位置, 200

固定定义的同步, 747

拐

拐角 C1 - C4 (\$AC_MEAS_TYPE = 4, 5, 6, 7) , 499

轨

轨迹

插补器, 646

横

横向槽, 368

换

换刀

固定点, 840

过程, 837

换刀导致的加工中断时间, 838

换刀点, 840

换刀时间, 838

换算至另一坐标系, 484

换向点, 681

回

回参考点, 196

回转轴, 721

360°模数, 726

尺寸单位, 723

调试, 737

工作区域, 724

进给率, 725

镜像, 739

绝对位置编程, 729, 734

模态转换, 729, 734

软件限位开关, 738

位置显示, 724

增量位置编程, 733, 736

轴地址, 722

回转轴位置的不唯一性

示例, 426

机

机床数据修改的生效方式, 827

机床轴（手轮 1、2、3）, 913

机床轴名称, 286

机械间隙, 249

激

激活, 420

激活挡块, 948

激活取轴, 不触发预处理停止, 353

激活同步, 752

级

级联方向, 410

几

几何尺寸监控, 967
几何轴轴组, 356

计

计算出的框架, 483
计算方法, 488

加

加工
 端面, 367
加速度, 228
加速度特征曲线, 608

夹

夹钳死区, 629

间

间隙
 补偿, 动态, 251
 补偿, 机械, 250
 动态, 251
 机械, 249

监

监控, 229

将

将轴的控制权转交给 PLC, 955

角

角度偏移“位置副主轴”, 753

接

接口信号
 DRF 已选择, 921, 922
 对机床轴手轮的旋转方向的取反生效, 941
 对几何轴(1, 2, 3)的手轮旋转方向的取反生效, 926

机床功能 INC1, INC10, INC100, INC1000,
INC10000, INCvar, 936

机床功能: 连续运行, 937

激活 DRF, 914

激活几何轴(1, 2, 3)的手轮(1 到 3), 914, 915, 919,
920

激活轮廓手轮仿真, 931

激活手轮 1 用作轮廓手轮, 930

激活手轮 2 用作轮廓手轮, 930

激活手轮 3 用作轮廓手轮, 930

激活手轮 (1 至 3), 932, 933

几何轴(1, 2, 3)的机床功能, 925, 926

几何轴(1, 2, 3)的移动键禁用, 915

几何轴(1, 2, 3)的正向/负向移动键, 917, 918

几何轴的快速移动速度叠加(1, 2, 3), 916

快速移动速度叠加, 933, 934

轮廓手轮仿真负方向, 931

轮廓手轮旋转方向取反生效, 932

请求几何轴(1, 2, 3)的“连续移动”机床功能, 919

取反定向轴(1, 2, 3)的手轮的旋转方向, 920, 921

取反机床轴手轮的旋转方向, 937

取反几何轴(1, 2, 3)的手轮的旋转方向, 919

取反轮廓手轮的旋转方向, 931

生效的机床功能: INC1, ..., 连续移动, 941

手动运行至固定点 0/1/2, 937

手动运行至固定点生效, 941

手轮 1 指定为轮廓手轮, 930

手轮 1 作为轮廓手轮生效, 931, 932

手轮 2 指定为轮廓手轮, 930

手轮 2 作为轮廓手轮生效, 931, 932

手轮 3 指定为轮廓手轮, 930

手轮 3 作为轮廓手轮生效, 931, 932

手轮叠加生效, 938

手轮生效 (1 至 3), 938

移动键禁用, 933

已手动运行至固定点, 942

运行请求 +/-, 939

运行指令 +/-, 940

正向/负向移动键, 935, 936

接口信号“进给停止/主轴停止”, 757

进

进给, 681

进给倍率, 668, 669

进给修调, 157

进给修调/快速移动修调, 226

禁

禁用 NCK 模拟量输出, 891, 892

禁用 NCK 模拟量输入, 889
禁用 NCK 数字量输出, 887
禁用 NCK 数字量输入, 886
禁用同步, 962

精

精同步, 963

静

静态 NC 存储器, 795
静态用户存储器, 799

拒

拒绝操作模式切换, 906

可

可切换的几何轴, 374, 396
可转换的接口, 605
可自动激活的预释放时间, 606

控

控制点, 253
控制性能, 784

跨

跨通道取进给轴/主轴, 324, 338, 942, 943, 944
跨通道取轴, 338

- 不触发预处理停止, 352
- 经过旋转的 WCS 中的几何轴, 356
- 另一个通道中的轴, 339
- 前提条件, 339
- 使能轴容器旋转, 351
- 释放时间点, 350
- 通过同步动作取轴, 357
- 轴类型, 338
- 自动生成的 GET/GETD, 346

跨通道取轴时可能出现的轴状态, 343
跨通道取主轴, 338

快

快速移动

- 插补方式, 646

扩展功能

功能手册, 03/2013, 6FC5397-1BP40-3RA1

快速移动叠加, 157

框

框架, 413

扩

扩展, 406
扩展存储器, 800

连

连续运行, 160

链

链接规则, 852

两

两个连续冲程间的最短时间, 607

零

零件程序, 454

轮

轮廓手轮, 191

逻

逻辑起始地址, 63

模

模数回转轴

- 定义为分度轴, 817
- 工作区域限制, 732

模数回转轴的监控状态, 961
模数回转轴上的工作区域限制, 961

磨

磨削刀具, 845
磨削刀具的补偿, 845

磨削刀具类型, 850

批

批量调试文件, 797

平

平面设置, 482
平移, 428, 482

前

前馈控制
 转矩, 300
 转速, 298
前馈控制, 297

切

切换精度
 挡块信号, 585

区

区域偏移, 69
区域下标, 69

确

确定性, 795

软

软件
 挡块, 569
软件限位开关, 229, 831

砂

砂轮圆周速度生效, 968

删

删除单轴剩余行程, 953

上

上电, 420

设

设定值, 481
设置换向点, 957

生

生效的进给轴, 960

时

时间常数
 动态响应自适应, 302

实

实际值设定, 477
 针对辅助轴（\$AC MEAS TYPE = 15）, 513
 针对几何轴和辅助轴（\$AC MEAS TYPE = 14）, 511
实际值同步, 963, 964

使

使能主动轴, 344

示

示例, 414
 TRAANG, 450
 TRACYL, 445
 TRANSMIT, 443

试

试车
 单通道式, 333

手

手动冲程释放, 950
手动冲程释放应答, 952
手动释放冲程, 951

手动退刀, 205
手动运行, 155, 225
手动运行中的几何轴, 225
手轮
 JOG 方式下的运行, 165
 分配, 166
 连接, 166
 速度设定, 173, 191
 通过 HMI 选择, 168
 行程设定, 173, 191
手轮连接
 以太网, 221
手轮连接 (828D)
 PPU, 215
 PROFIBUS, 216
手轮已选中 (手轮 1、2、3), 912, 913

输

输出值, 478
输入信号监控, 607
输入值, 478
 测量类型, 479
 设定值, 481, 483

数

数量
 转换, 371, 389, 401

双

双向测头, 468

速

速度, 175, 227
速度和加速度, 229
速度控制, 406
速度控制的优化, 403

所

所有坐标转换, 439

特

特例情形, 406
特殊性, 422

扩展功能
功能手册, 03/2013, 6FC5397-1BP40-3RA1

通

通道, 323, 641
 同步, 326
通道轴, 338
通道轴名称, 286
通过 GETD 自动取轴, 346
通过 PLC 取轴, 671
通过 PLC 置位 NCK 数字量输入, 887
通过手轮设定行程, 191
通过在程序中写入定位轴名来指定进给率, 956

同

同步
 定义新的, 761
 固定定义, 761

网

网格结构, 271

往

往复运动, 681
 带同步运动, 707
 连续进给, 681
 异步, 681
往复运动出错, 959
往复运动激活, 960
往复运动无法启动, 959
往复轴, 681
往复轴换向生效, 959

位

位置的不唯一性
 示例, 425
位置偏移
 在主轴同步中, 772

温

温度
 补偿, 242
 影响, 242
温度补偿
 系数 $\tan\beta(T)$, 247

无

无槽壁补偿的编程, 448
无冲程释放, 950
无火花光磨, 681

误

误差

测量系统, 255
垂度, 261
角度, 261
曲线, 用于温度补偿, 242
丝杠螺距, 255

系

系统变量, 413, 470, 475

相

相乘

表格, 263

行

行程划分, 612
行程开关信号, 569
提前时间/延迟时间, 581

修

修改换向点, 957, 958

旋

旋转方向, 395
旋转角度, 395

选

选择, 421
选择刀具或刀沿, 485
选择和取消, 405, 411

延

延时, 556

沿

沿制动斜坡停止, 958

一

一般通道配置, 452

应

应用, 409

硬

硬件限位开关, 229

用

用户定义的同步: , 747

由

由 PLC 控制的轴
MD30460 位 6 和 位 7 对应的控制特性, 674
控制特性, 674
由手轮 1、2、3 移动的几何轴所处通道的编号, 910

语

语言指令, 601
SPN, 613, 615
SPP, 612, 614

圆

圆柱外表面, 368

运

运动结束条件
程序段搜索时, 668
运动特性, 676

运行方式, 230

JOG, 156

运行方式组, 323, 324

运行键, 160

运行至固定点, 840

使用 G75, 197

在 JOG 方式下, 197

在

在磨削期间同步修整砂轮, 861

在倾斜平面上重新定义 WCS' 坐标系

(\$AC_MEAS_TYPE = 18), 520

在下一个换向点上停止, 958

在主轴/进给轴模式下进行动态响应编程, 787

正

正

挡块, 569

正挡块信号 1-32, 948

正进行光磨, 959, 960

正进行往复运动, 960

中

中立轴, 338

重

重新配置, 795, 797

重新同步, 773

轴

轴

补偿, 253

参照, 253

插补器, 646

用于辅助运动, 639

轴复位已执行, 955

轴角度的叠加范围

TU 地址, 424

轴类型

针对定位轴, 643

轴配置, 409

轴容器, 287

名称, 111

轴容器旋转生效, 909

轴运行就绪, 909

扩展功能

功能手册, 03/2013, 6FC5397-1BP40-3RA1

主

主动/被动操作模式, 906

主动文件系统, 796

主轴的手动运行, 227

主轴号, 852

主轴同步, 745, 962

撤销, 765

位置偏移, 772

折线式加速度特性曲线, 787

主轴同步方式, 747

主主轴有效, 964, 965

属

属性, 436

转

转换生效, 944

转速监控, 873, 968

自

自动方式下的手轮叠加

编程和激活, 188

速度叠加, 184

行程设定, 184

自控单轴运动

NCK 响应, 672

PLC 动作, 672

纵

纵向槽, 368

坐

坐标 PTP 运动, 420

STAT 地址, 423

TU 地址, 424

坐标手动运动, 427

坐标转换

级联顺序, 410

坐标转换的数量, 409

坐标转换级联, 408, 454

固定生效的坐标转换, 411

激活, 410

取消, 411
示例, 411, 451
数量, 410
特殊性, 411
坐标转换级联, 设定位置, 417
坐标转换级联数量, 410