

SINUMERIK

SINUMERIK MC 进给轴和主轴

功能手册

适用于

控制系统
SINUMERIK MC

CNC 软件 版本 1.21

01/2023
A5E47437747F AF

前言	1
基本安全说明	2
有关功能性的信息	3
回参考点	4
加速度和急动度	5
速度、设定值/实际值系统、闭环控制	6
进给率	7
端面轴	8
定位轴	9
回转轴	10
分度轴	11
龙门轴	12
主轴	13
同步主轴	14
轴耦合	15
手动运行	16

转下页

SINUMERIK MC
进给轴和主轴

功能手册

续

运行到固定挡块	17
设定值切换	18
切向控制	19
往复	20
软件挡块，行程开关信号	21
扩展的停止和退回	22
行程计算	23
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 Siemens AG 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。

目录

- 1 前言 23
 - 1.1 关于 SINUMERIK 23
 - 1.2 关于本手册 23
 - 1.3 网上文档 26
 - 1.3.1 MC 文档一览 26
 - 1.3.2 SINUMERIK 操作组件文档一览 26
 - 1.4 技术文档反馈 27
 - 1.5 mySupport 文档 27
 - 1.6 服务与支持 28
 - 1.7 OpenSSL 的使用 29
 - 1.8 遵守通用数据保护条例 30
- 2 基本安全说明 31
 - 2.1 一般安全说明 31
 - 2.2 应用示例的质保规定 31
 - 2.3 安全性信息 31
- 3 有关功能性的信息 33
- 4 回参考点 35
 - 4.1 简要说明 35
 - 4.2 轴专用回参考点 36
 - 4.3 通道专用回参考点 39
 - 4.4 在零件程序中写入回参考点（G74） 41
 - 4.5 使用增量测量系统回参考点 42
 - 4.5.1 硬件信号 42
 - 4.5.2 零脉冲选择 44
 - 4.5.3 时序 45
 - 4.5.4 阶段 1：运行到参考凸轮 47
 - 4.5.5 阶段 2：与零脉冲同步 50
 - 4.5.6 阶段 3：运行至参考点 55
 - 4.6 通过距离编码的参考脉冲回参考点 58
 - 4.6.1 概述 58
 - 4.6.2 基础参数设置 59

4.6.3	时序	61
4.6.4	阶段 1: 同步越过参考脉冲	62
4.6.5	阶段 2: 运行到目标点	63
4.7	通过实际值补偿回参考点	66
4.7.1	对回参考点测量系统进行实际值补偿	66
4.7.2	使用距离编码参考脉冲的测量系统中的实际值补偿	67
4.8	跟踪模式下回参考点	68
4.9	使用绝对值编码器回参考点	71
4.9.1	校准信息	71
4.9.2	通过输入参考点偏移校准	73
4.9.3	通过输入参考点值校准	74
4.9.4	通过测头自动校准	76
4.9.5	通过 BERO 校准	77
4.9.6	使用绝对值编码器回参考点	79
4.9.7	通过旋转绝对值编码器启用替代零脉冲回参考点	79
4.9.8	激活测量系统	82
4.9.9	不支持的回参考点方式	84
4.10	自动恢复机床参考	84
4.10.1	概述	84
4.10.2	自动回参考点	85
4.10.3	恢复实际位置	86
4.11	前提条件	89
4.11.1	较大运行范围	89
5	加速度和急动度	91
5.1	简要说明	91
5.2	详细说明	93
5.2.1	刚性加速模式 (BRISK/BRISKA) (通道/轴专用)	93
5.2.1.1	功能	93
5.2.1.2	参数设置	94
5.2.1.3	编程	95
5.2.2	恒速运行时间 (通道专用)	97
5.2.2.1	功能	97
5.2.2.2	参数设置	98
5.2.3	加速度自适应 (ACC) (轴专用)	98
5.2.3.1	功能	98
5.2.3.2	编程	99
5.2.4	加速裕量 (通道专用)	99
5.2.4.1	功能	99
5.2.4.2	参数设置	100
5.2.5	最大轨迹加速度 (通道专用)	100
5.2.5.1	功能	100

5.2.5.2	参数设置	100
5.2.5.3	编程	100
5.2.6	针对实时事件的轨迹加速度（通道专用）	101
5.2.6.1	功能	101
5.2.6.2	编程	102
5.2.7	快速移动条件下的单独加速度 (G00)（轴专用）	103
5.2.7.1	功能	103
5.2.7.2	参数设置	103
5.2.8	软加速模式 (SOFT/SOFTA)（轴专用）	104
5.2.8.1	功能	104
5.2.8.2	参数设置	104
5.2.9	在非相切的程序段过渡条件下上调加速度（轴专用）	104
5.2.9.1	功能	104
5.2.9.2	参数设置	105
5.2.10	为径向加速度保留的加速裕量（通道专用）	105
5.2.10.1	功能	105
5.2.10.2	参数设置	106
5.2.11	轨迹插补中的最大急动度 (SOFT)（通道专用）	107
5.2.11.1	功能	107
5.2.11.2	参数设置	109
5.2.11.3	编程	109
5.2.12	单轴插补中的最大急动度 (SOFTA)（轴专用）	110
5.2.12.1	参数设置	110
5.2.12.2	编程	111
5.2.13	最大轨迹急动度（通道专用）	112
5.2.13.1	功能	112
5.2.13.2	参数设置	112
5.2.13.3	编程	112
5.2.14	针对实时事件的轨迹急动度（通道专用）	113
5.2.14.1	功能	113
5.2.14.2	编程	114
5.2.15	快速移动条件下的急动度 (G00)（轴专用）	115
5.2.15.1	功能	115
5.2.15.2	参数设置	115
5.2.16	在非恒定曲率的程序段过渡处自动上调急动度（轴专用）	115
5.2.16.1	功能	115
5.2.16.2	参数设置	116
5.2.17	随速度提高急动度（轴专用）	116
5.2.18	急动度滤波器（位置设定值滤波器，轴专用）	118
5.2.18.1	功能	118
5.2.18.2	参数设置	121
5.2.19	折线式加速度特性曲线	122
5.2.19.1	功能：适应电机特性曲线	122
5.2.19.2	功能：对轨迹加速度的影响	123
5.2.19.3	功能：备用特性曲线	123

5.2.19.4	参数设置	126
5.2.19.5	编程：通道专用激活（DRIVE）	129
5.2.19.6	编程：轴专用激活（DRIVEA）	130
5.2.19.7	前提条件	130
5.2.20	JOG 运动中的加速度和急动度	131
5.2.20.1	参数设置	131
5.2.20.2	前提条件	133
5.3	示例	134
5.3.1	加速度	134
5.3.1.1	轨迹速度曲线	134
5.3.2	急动度	135
5.3.2.1	轨迹速度曲线	135
5.3.3	加速度和急动度	137
5.3.4	折线式加速度特性曲线	138
5.3.4.1	激活	138
6	速度、设定值/实际值系统、闭环控制	141
6.1	简要说明	141
6.2	速度、运行范围、精度	141
6.2.1	速度	141
6.2.2	运行范围	143
6.2.3	控制系统的定位精度	144
6.2.4	输入/显示精度，计算精度	145
6.2.5	机床数据和设定数据物理量的定标	147
6.3	英制/公制单位制	150
6.3.1	功能	150
6.3.1.1	参数设置和编写的单位制	150
6.3.1.2	扩展单位制功能	151
6.3.2	调试	151
6.3.3	编程	155
6.3.3.1	切换单位制 (G70/G71/G700/G710)	155
6.4	设定值/实际值系统	159
6.4.1	简介	159
6.4.2	设定值和编码器分配	162
6.4.3	调整电机/负载齿轮比	167
6.4.4	转速设定值输出	169
6.4.5	实际值系统的机床数据	171
6.4.6	实际值精度	172
6.4.6.1	实际值分辨率的机床数据	172
6.4.6.2	示例：带光栅尺的线性轴	175
6.4.6.3	示例：电机上带旋转编码器的线性轴	176
6.4.6.4	示例：机床上带旋转编码器的线性轴	178
6.4.6.5	示例：电机上带旋转编码器的回转轴	178

6.4.6.6	示例：机床上带旋转编码器的回转轴.....	180
6.4.6.7	示例：刀具上带编码器的中间齿轮比	180
6.5	闭环控制	181
6.5.1	进给轴/主轴的位置闭环控制	181
6.5.2	可切换的位置设定值—滤波器链条	184
6.5.3	位置设定值滤波器：限制滤波器长度	188
6.5.4	位置控制器的参数组	189
6.6	控制优化	190
6.6.1	位置控制器，位置设定值滤波器：对称滤波器	190
6.6.2	位置控制器，位置设定值滤波器：相位滤波器	194
6.6.3	位置控制器：位置差接通	196
6.6.4	带比例积分控制器的位置闭环控制	197
7	进给率	201
7.1	简要说明	201
7.2	轨迹进给率 F	202
7.2.1	进给类型 G93、G94、G95	204
7.2.2	进给类型 G96、G961、G962、G97、G971	208
7.2.3	螺纹切削（G33, G34, G35, G335, G336）中的进给率	212
7.2.3.1	G33 中的进给率	212
7.2.3.2	G34 和 G35 中线性递增式/递减式的螺距变化	213
7.2.3.3	G33、G34 和 G35 中轴的加速度特性	215
7.2.3.4	G33、G34、G35 中可编程的导入和导出行程（DITS、DITE）	217
7.2.3.5	螺纹切削期间快速返回	219
7.2.3.6	球螺纹（G335, G336）	223
7.2.4	刚性攻丝（G331、G332）中的进给率	228
7.2.5	带补偿夹具攻丝（G63）中的进给率	231
7.2.6	FGROUP 和 FGREF	231
7.3	定位轴的进给率（FA）	234
7.4	进给控制	236
7.4.1	进给禁止和进给/主轴停止	236
7.4.2	通过机床控制面板设置进给倍率	238
7.4.3	可编程的进给倍率	243
7.4.4	可编程的最大轨迹速度	244
7.4.5	空运行进给率	246
7.4.6	一个程序段中的多个进给值	248
7.4.7	固定进给值	253
7.4.8	可编程的进给特性曲线	255
7.4.9	倒角/倒圆进给率 FRC、FRCM	256
7.4.10	逐段进给率 FB	258
7.4.11	单轴动态响应的控制	259
7.5	边界条件	265

8	端面轴	267
8.1	功能	267
8.2	参数设置	270
8.3	编程	274
8.4	前提条件	275
8.5	示例	275
9	定位轴	277
9.1	简要说明	277
9.2	独立通道、定位轴或并行定位轴	279
9.2.1	独立通道	279
9.2.2	定位轴	281
9.2.3	并行定位轴	283
9.3	运动特性和插补过程	284
9.3.1	轨迹插补器和轴插补器	284
9.3.2	自控单轴运动	285
9.3.3	包含 NC 控制 ESR 的自控单轴运动	290
9.4	定位轴动态响应	292
9.5	编程	294
9.5.1	简介	294
9.5.2	外部编程时的旋转进给率	297
9.6	程序段切换	297
9.6.1	可设置的程序段切换时间点	299
9.6.2	程序段搜索时的运动结束条件	304
9.7	通过 PLC 控制	305
9.7.1	由 PLC 启动并行定位轴	308
9.7.2	由 PLC 控制的轴	308
9.7.3	PLC 轴的控制特性	311
9.8	特殊功能下的特性	313
9.8.1	空运行进给率 (DRY RUN)	313
9.8.2	单程序段	313
9.9	示例	313
9.9.1	运动特性和插补过程	313
10	回转轴	315
10.1	简要说明	315
10.2	360°模数	317
10.3	回转轴的编程	322

10.3.1	简介	322
10.3.2	激活模数功能的回转轴（无限旋转的回转轴）	323
10.3.3	没有激活模数功能的回转轴	326
10.3.4	回转轴的其他编程特殊性	328
10.4	调试回转轴	329
10.5	回转轴的特殊性	330
10.6	示例	331
11	分度轴	333
11.1	简要说明	333
11.2	详细说明	333
11.2.1	在 AUTO 模式下运行分度轴	333
11.2.2	在 JOG 模式下运行分度轴	333
11.2.3	PLC 运行分度轴	335
11.3	调试	336
11.3.1	分度轴的机床数据	336
11.3.2	等距分度的机床数据	342
11.3.2.1	属性	342
11.3.2.2	端面齿轴	343
11.3.3	轴专用系统变量	344
11.3.4	NC/PLC 接口信号	345
11.4	编程	345
11.5	边界条件	348
11.6	示例	351
11.6.1	示例 1：回转轴作为分度轴	351
11.6.2	示例 2：分度轴为线性轴	352
11.6.3	示例 3：模数回转轴为等距分度轴	353
11.6.4	示例 4：回转轴作为等距分度轴，运行区域受限	353
11.6.5	示例 5：线性轴作为等距分度轴	354
11.6.6	示例 6：“端面齿轴”	355
12	龙门轴	357
12.1	简要说明	357
12.2	“龙门轴”功能	358
12.2.1	定义一个龙门轴组	358
12.2.2	同步运行偏差监控	359
12.2.3	同步运行偏差扩展监控	360
12.2.4	龙门轴的回参考点和同步	361
12.2.5	控制动态响应	362
12.2.6	拆解龙门轴组	362
12.3	龙门轴的回参考点和同步	363

12.3.1	简介	363
12.3.2	自动同步	370
12.3.3	特殊性	372
12.4	调试龙门轴	374
12.5	参数设置：故障情况中的特性	379
12.6	龙门轴中的 PLC 接口信号	380
12.7	龙门轴的其它事项	384
12.8	示例	387
12.8.1	创建龙门轴组	387
12.8.2	NC-PLC 接口设置	388
12.8.3	开始调试	389
12.8.4	设置警告极限和误差限制	392
13	主轴	395
13.1	简要说明	395
13.2	运行方式	396
13.2.1	概述	396
13.2.2	运行方式切换	397
13.2.3	控制模式	398
13.2.4	往复模式	403
13.2.5	定位模式	403
13.2.5.1	一般功能	403
13.2.5.2	从旋转状态定位主轴	411
13.2.5.3	从静止状态定位主轴	415
13.2.5.4	换刀信号“主轴到达位置”	419
13.2.6	轴运行方式	421
13.2.6.1	一般功能	421
13.2.6.2	隐性过渡至进给轴模式	425
13.2.7	主轴初始设置	428
13.2.8	刚性攻丝	429
13.2.8.1	功能	429
13.2.8.2	编程	430
13.2.8.3	示例：使用 G331 / G332 攻丝	431
13.2.8.4	示例：给出当前齿轮级已编程的钻削转速	432
13.2.8.5	示例：使用第二齿轮级数据组	432
13.2.8.6	示例：未进行转速编程，齿轮级监控	432
13.2.8.7	示例：无法进行齿轮换档，齿轮级监控	433
13.2.8.8	示例：不带 SPOS 的编程	433
13.2.9	带弹性卡头的攻丝	434
13.2.9.1	功能	434
13.2.9.2	编程	434
13.3	回参考点/同步	436

13.4	可配置的齿轮档调整	441
13.4.1	主轴的齿轮档和齿轮档切换	441
13.4.2	主轴齿轮档 0	452
13.4.3	确定主轴齿轮档	455
13.4.4	齿轮档切换时的参数组选择	456
13.4.5	中间齿轮比	460
13.4.6	未应答的齿轮档切换	461
13.4.7	通过往复模式进行齿轮档切换	462
13.4.8	在固定位置切换齿轮档	470
13.4.9	执行 M70 时启用的齿轮档	479
13.4.10	在 DryRun、程序测试和 SERUPRO 下抑制齿轮档切换	481
13.5	其他主轴功能调整设置	484
13.6	可切换主轴	486
13.7	编程	489
13.7.1	在零件程序中编程	489
13.7.2	通过同步动作编程	494
13.7.3	通过 PLC 使用 SINU_RotateSpindle 编程主轴控制	495
13.7.4	外部编程 (PLC)	495
13.8	主轴监控	497
13.8.1	允许的转速范围	497
13.8.2	进给轴/主轴停止	498
13.8.3	主轴位于设定区域内	499
13.8.4	齿轮档的最小/最大转速	500
13.8.5	主轴转速上限	502
13.8.6	读取主轴参数	504
13.8.7	最大编码器极限频率	506
13.8.8	目标点监控	508
13.8.9	M40: 转速超出配置的开关阈值时的自动齿轮档选择	509
13.9	配备 SMI 24 的主轴 (Weiss 主轴)	511
13.9.1	简介	511
13.9.2	传感器数据	512
13.9.3	夹紧状态	514
13.9.4	其他驱动参数	516
13.10	边界条件	517
13.10.1	闭环控制参数的修改	517
13.11	示例	517
13.11.1	自动齿轮档选择 (M40)	517
14	同步主轴	519
14.1	简要说明	519
14.1.1	功能	519

14.1.2	主轴同步	521
14.1.3	主轴同步的前提条件	527
14.1.4	通过零件程序来激活主轴同步	528
14.1.5	通过零件程序来撤销主轴同步	530
14.1.6	通过 PLC 控制主轴同步	531
14.1.7	监控主轴同步	534
14.1.8	同步主轴的动态修调	536
14.2	编程	537
14.2.1	定义 (COUPDEF)	537
14.2.2	激活和撤销同步 (COUPON, COUPONC, COUPOF, COUPOFS)	541
14.2.3	主轴同步的轴系统变量	542
14.2.4	自动激活和撤销位置闭环控制	544
14.3	配置	545
14.3.1	NC 启动时同步主轴对的特性	546
14.3.2	复位时同步主轴对的特性	546
14.4	特殊性	547
14.4.1	主轴同步的一般特点	547
14.4.2	再次恢复副主轴和主主轴的同步	550
14.4.3	主轴同步和 NC/PLC 接口信号	552
14.4.4	主主轴和副主轴之间的差速	557
14.4.5	同步补偿中同步信号的特性	563
14.4.6	删除同步补偿及 NC 复位	563
14.4.7	主轴同步功能调试的特殊性	563
14.5	边界条件	569
14.6	示例	569
15	轴耦合	571
15.1	联动	571
15.1.1	简要说明	571
15.1.1.1	功能	571
15.1.1.2	前提条件	571
15.1.2	一般功能	572
15.1.3	编程	576
15.1.3.1	联动组的定义和启用 (TRAILON)	576
15.1.3.2	关闭 (TRAILOF)	577
15.1.4	PLC 接口信号的有效性	578
15.1.5	耦合状态	580
15.1.6	动态响应限制	581
15.1.7	边界条件	581
15.1.8	示例	582
15.2	曲线表	583
15.2.1	简要说明	583

15.2.1.1	功能	583
15.2.1.2	前提条件	583
15.2.2	一般功能	584
15.2.3	存储器结构	585
15.2.4	调试	587
15.2.4.1	存储器配置	587
15.2.4.2	刀具半径补偿	587
15.2.4.3	指定存储器类型	588
15.2.5	编程	588
15.2.6	访问表位置和表段	596
15.2.7	激活/禁用	599
15.2.8	模数引导轴特殊情况	600
15.2.9	AUTO、MDA、JOG 运行方式下的特性	601
15.2.10	PLC 接口信号的有效性	601
15.2.11	诊断和优化资源使用	602
15.2.12	边界条件	606
15.2.13	示例	606
15.3	引导值耦合	608
15.3.1	简要说明	608
15.3.1.1	功能	608
15.3.1.2	前提条件	608
15.3.2	一般功能	609
15.3.3	编程	612
15.3.4	AUTO、MDA、JOG 运行方式下的特性	617
15.3.5	PLC 接口信号的有效性	619
15.3.6	轴引导值耦合功能的特点	620
15.3.7	边界条件	620
15.4	电子齿轮箱 (EG)	621
15.4.1	简要说明	621
15.4.1.1	功能	621
15.4.1.2	前提条件	622
15.4.2	电子齿轮箱 (EG)	622
15.4.3	定义一个 EG 轴组	630
15.4.4	启用一个 EG 轴组	631
15.4.5	关闭一个 EG 轴组	636
15.4.6	删除一个 EG 轴组	637
15.4.7	旋转进给 (G95) 与电子齿轮箱的共同作用	637
15.4.8	上电、复位、运行方式切换、搜索时的特性	637
15.4.9	电子齿轮箱的系统变量	638
15.4.10	示例	640
15.4.10.1	使用线性耦合的示例	640
15.4.10.2	包含非线性部分的扩展示例	644
15.5	通用耦合	651

15.5.1	简要说明	651
15.5.1.1	功能	651
15.5.1.2	前提条件	651
15.5.2	基本介绍	655
15.5.2.1	耦合模块	655
15.5.2.2	关键字和耦合属性	656
15.5.2.3	系统变量	659
15.5.3	创建/删除耦合模块	662
15.5.3.1	创建耦合模块 (CPDEF)	662
15.5.3.2	删除耦合模块 (CPDEL)	663
15.5.3.3	定义引导轴 (CPLDEF 或 CPDEF+CPLA)	664
15.5.3.4	删除引导轴 (CPLDEL 或 CPDEL+CPLA)	666
15.5.4	激活/关闭耦合	667
15.5.4.1	激活耦合模块 (CPON)	667
15.5.4.2	关闭耦合模块 (CPOF)	668
15.5.4.3	激活耦合模块中的引导轴 (CPLON)	669
15.5.4.4	关闭耦合模块中的引导轴 (CPLOF)	670
15.5.4.5	隐性创建和删除耦合模块	670
15.5.5	编程耦合属性	671
15.5.5.1	耦合规律 (CPLNUM、CPLDEN、CPLCTID)	671
15.5.5.2	耦合基准 (CPLSETVAL)	674
15.5.5.3	参考坐标 (CPFRS)	675
15.5.5.4	程序段转换特性 (CPBC)	676
15.5.5.5	激活时跟随轴的同步位置 (CPFPOS+CPON)	678
15.5.5.6	激活时引导轴的同步位置 (CPLPOS)	679
15.5.5.7	同步模式 (CPFMSON)	680
15.5.5.8	激活时跟随轴的特性 (CPFMON)	682
15.5.5.9	关闭时跟随轴的特性 (CPFMOF)	682
15.5.5.10	关闭时跟随轴的位置 (CPFPOS+CPOF)	683
15.5.5.11	复位时的状态 (CPMRESET)	684
15.5.5.12	零件程序开始时的状态 (CPMSTART)	685
15.5.5.13	通过程序测试进行搜索的情况下零件程序开始时的状态 (CPMPRT)	687
15.5.5.14	偏移/缩放 (CPLINTR、CPLINSC、CPLOUTTR、CPLOUTSC)	688
15.5.5.15	1 级同步运行监控 (CPSYNCOF、CPSYNFIP、CPSYNCOV、CPSYNFIV)	691
15.5.5.16	2 级同步运行监控 (CPSYNCOF2、CPSYNFIP2)	695
15.5.5.17	对停止信号和停止指令的响应 (CPMBRAKE)	699
15.5.5.18	对特定 NC/PLC 接口信号的响应 (CPMVDI)	701
15.5.5.19	报警抑制 (CPMALARM)	706
15.5.6	耦合级联	707
15.5.7	兼容性	708
15.5.7.1	匹配循环	708
15.5.7.2	耦合类型 (CPSETTYPE)	709
15.5.7.3	配置的耦合 (CPRES)	715
15.5.8	跨通道耦合、跨通道取轴	716
15.5.9	回转轴的特性	716

15.5.10	上电、复位等时的特性	718
15.5.11	CP 软件限制监控.....	718
15.5.11.1	功能	718
15.5.11.2	参数设置	721
15.5.11.3	编程	721
15.5.11.4	边界条件	721
15.5.11.5	示例	722
15.5.12	故障特性	723
15.5.12.1	快速停止	723
15.5.13	跟踪同步运行偏差	724
15.5.13.1	基本介绍	724
15.5.13.2	测量同步运行偏差	726
15.5.13.3	直接指定同步运行偏差	729
15.5.13.4	同步运行补偿	729
15.5.13.5	同步运行补偿的诊断	730
15.5.13.6	删除同步运行补偿	731
15.5.13.7	边界条件	733
15.5.14	示例	734
15.5.14.1	编程示例	734
15.5.14.2	调整匹配循环	736
15.6	跟随轴的动态响应特性	737
15.6.1	参数设置的动态响应限制	737
15.6.2	编程的动态响应限制	738
15.6.2.1	编程（VELOLIMA、ACCLIMA）	738
15.6.2.2	示例	740
15.6.2.3	系统变量	741
15.7	一般边界条件	742
16	手动运行	743
16.1	概述	743
16.2	通过 PLC 接口控制	746
16.3	常规参数设置	747
16.4	连续手动运行	753
16.4.1	功能	753
16.4.2	参数设置	756
16.4.3	边界条件	756
16.5	增量手动运行	757
16.5.1	功能	757
16.5.2	参数设置	762
16.5.3	边界条件	763
16.6	通过手轮进行手动运行	763
16.6.1	功能	763

16.6.2	参数设置	772
16.6.3	运行请求	779
16.7	手动运行主轴	781
16.8	手动运行几何轴/定向轴	783
16.9	手动运行至固定点	787
16.9.1	功能	787
16.9.2	参数设置	791
16.9.3	编程	793
16.9.4	边界条件	793
16.9.5	应用示例	794
16.10	手动运行至位置	795
16.10.1	功能	795
16.10.2	参数设置	799
16.10.3	边界条件	799
16.10.4	应用示例	800
16.11	圆弧手动运行	801
16.11.1	功能	801
16.11.2	参数设置	808
16.11.3	边界条件	810
16.11.4	应用示例	811
16.12	沿刀具中心轴的退刀（手动退刀）	812
16.12.1	概述	812
16.12.2	参数设置	813
16.12.2.1	上电后自动选择手动退刀	813
16.12.2.2	运行方向使能	814
16.12.2.3	测量系统的状态	814
16.12.3	选择	814
16.12.4	退刀	816
16.12.5	取消	818
16.12.6	重新选择	819
16.12.7	继续程序执行	819
16.12.8	状态图	821
16.12.9	系统数据	821
16.12.10	边界条件	822
16.13	在 AUTO 模式下使用手轮	823
16.13.1	自动方式下的手轮叠加	823
16.13.1.1	一般功能	823
16.13.1.2	手轮叠加的编程和激活	828
16.13.1.3	自动方式下的手轮叠加的特殊性	831
16.13.2	轮廓手轮/通过手轮设定行程（选件）	832
16.13.3	DRF 偏移	835
16.13.3.1	功能	835

16.13.3.2	参数设置	838
16.13.3.3	编程：针对轴取消叠加（CORROF）	839
16.13.4	手轮的重复使用	842
16.14	监控	843
16.15	调试：手轮	844
16.15.1	简介	844
16.15.2	通过 PROFINET IO 连接	845
16.15.3	TIA 博途中的配置	848
17	运行到固定挡块	851
17.1	简要说明	851
17.2	详细说明	852
17.2.1	编程	852
17.2.2	工作流程	854
17.2.2.1	选择	854
17.2.2.2	到达固定挡块	855
17.2.2.3	未到达固定挡块	858
17.2.2.4	取消	859
17.2.3	程序段搜索时的特性	862
17.2.4	复位和功能中断时的特性	866
17.2.5	与其他功能综合使用时的特性	867
17.2.6	设定数据	868
17.2.7	系统变量	869
17.2.8	报警	871
17.3	示例	872
18	设定值切换	875
18.1	简要说明	875
18.1.1	简要说明	875
18.2	调试	877
18.3	流程图	882
18.4	边界条件	883
19	切向控制	885
19.1	调试	886
19.2	编程	888
19.2.1	定义耦合 (TANG)	888
19.2.2	激活中间程序段生成 (TLIFT)	890
19.2.3	激活耦合 (TANGON)	891
19.2.4	取消耦合 (TANGOF)	892
19.2.5	删除耦合 (TANGDEL)	892

19.3	极限角度	894
19.4	边界条件	895
19.5	示例	895
20	往复	899
20.1	简要说明	899
20.2	异步往复运动	900
20.2.1	异步往复运动	900
20.2.2	异步往复运动的设置选项	901
20.2.3	由 PLC 控制的异步往复运动	908
20.2.4	异步往复运动中的特殊响应	908
20.3	通过同步动作控制的往复运动	912
20.3.1	在换向点 1 或 2 上作进给运动	915
20.3.2	在换向区内作进给运动	916
20.3.3	在两个换向点上作进给运动	918
20.3.4	在换向点上停止往复运动	918
20.3.5	继续往复运动	919
20.3.6	阻止下一个进给行程过早开始	920
20.3.7	OSCILL 指令：为往复轴指定进给轴	921
20.3.8	POSP 指令：指定进给运动方式	921
20.3.9	从外部触发往复轴换向	922
20.4	边界条件	924
20.5	示例	924
20.5.1	异步往复运动程序示例	924
20.5.2	示例 1：带同步动作的往复运动	925
20.5.3	示例 2：带同步动作的往复运动	928
20.5.4	示例：起点	930
20.5.4.1	通过语言指令来定义起点	930
20.5.4.2	通过设定数据启动往复运动	931
20.5.4.3	起点和换向点重合时的往复运动（起点 = 换向点 1）	931
20.5.5	示例：从外部触发往复轴换向	934
20.5.5.1	由外部触发往复轴换向时通过同步动作修改换向点	934
21	软件挡块，行程开关信号	935
21.1	简要说明	935
21.2	挡块信号和挡块位置	936
21.2.1	挡块信号单独输出	936
21.2.2	挡块信号关联输出	939
21.2.3	挡块位置	942
21.2.4	提前时间/延迟时间（动态挡块）	944
21.3	挡块信号输出	945

21.3.1	激活	945
21.3.2	挡块信号输出到 PLC	946
21.3.3	挡块信号以位置控制周期输出到 NCK I/O	947
21.3.4	挡块信号定时输出	949
21.3.5	挡块信号单独定时输出	950
21.4	“行程-时间”挡块	951
21.5	边界条件	952
22	扩展的停止和退回	953
22.1	简要说明	953
22.2	驱动自控 ESR.....	953
22.2.1	基本原理	953
22.2.2	在驱动中配置停止功能	954
22.2.3	在驱动中配置退回	956
22.2.4	在驱动中配置再生运行	957
22.2.5	通过系统变量使能 ESR.....	959
22.2.6	通过系统变量触发 ESR.....	960
22.2.7	ESR 状态的反馈	960
22.2.8	应答 ESR 反应	961
22.2.9	在零件程序中配置 ESR.....	962
22.2.9.1	停止 (ESRS)	963
22.2.9.2	退回 (ESRR)	963
22.2.9.3	边界条件	965
22.3	边界条件	966
22.4	示例	967
22.4.1	NC 引导下的反应	967
22.4.2	螺纹切削时退回	968
22.4.3	通过 ASUB 和快速输入快速退刀	969
22.4.4	绝对快速退刀和增量快速退刀	969
22.4.5	以轴的线性关系快速退刀	970
23	行程计算.....	971
23.1	简要说明	971
23.2	数据	972
23.3	参数设置	972
23.3.1	常规激活	972
23.3.2	数据组.....	973
23.4	示例	973
23.4.1	每个零件程序的移行行程	973
A	附录.....	975
A.1	缩略语列表.....	975

索引 983

前言

1.1 关于 SINUMERIK

无论是普及型数控机床，还是标准型机床，或者是模块化高端机床，SINUMERIK 数控系统都能为不同类型的机床提供最佳解决方案。无论是单件生产还是批量生产、简单工件还是复杂工件，对于从样品和工具制造、模具制造乃至大批量生产的所有制造领域而言，SINUMERIK 自始至终都是高生产率的自动化解决方案。

详细信息请访问网页 SINUMERIK (<https://www.siemens.com/sinumerik>)。

1.2 关于本手册

本手册属于 SINUMERIK 功能手册系列。

SINUMERIK 功能手册

SINUMERIK 功能手册系列会介绍 SINUMERIK 控制系统的各种数控功能。

功能手册系列的目标读者是：机床配置选型人员、工艺工程师、调试人员和编程人员。

每本功能手册都涵盖了特定主题，介绍这些主题下的所有功能。

下表列出了 SINUMERIK 上的所有功能手册以及每本手册涵盖的主题：

功能手册	主题
基本功能	数控系统上的基本功能
进给轴和主轴	进给轴和主轴功能、轴耦合
坐标转换	坐标转换功能
监控和补偿	轴的监控和补偿功能、碰撞监控
刀具	刀具的选择、补偿和监控功能
同步动作	同步动作的功能
PLC	PLC 基本程序结构和功能
刀具管理	刀具管理的功能、调试和编程
工艺	工艺扩展功能

1.2 关于本手册

目录

功能手册的封面是一张目录，列出了手册的章节标题（即涉及的功能）。

适用性

手册封面还对本手册的适用性进行了说明，即该版本的编程手册适用于哪些 SINUMERIK 控制系统和哪些软件版本。

系统数据

在功能说明中，通常不会详细介绍功能涉及的系统数据（机床数据、设定数据、系统变量、接口信号和报警），除非是为了便于读者理解功能。这些数据的详细介绍可以查看对应的参数手册、报警查看诊断手册。

Basic Program Plus 和 Basic Program

在 TIA Portal 中为 PLC 功能的编程提供了两种不同的基本程序类型：

- Basis Program Plus 只能使用符号寻址进行编程。功能通过数据类型、PLC 程序块、PLC 函数和 PLC 指令来实现。Basic Program Plus 只可用于 SINUMERIK ONE。
- Basic Program 既可使用符号寻址，也可使用绝对寻址来编程。因此，大部分程序可与已有的 SIMATIC Manager 程序兼容。Basic Program 可用于 Steuerungen ONE 和 MC。

无法同时使用 Basic Program Plus 和 Basic Program。

PLC 程序块和数据类型

在功能说明中常会涉及 PLC 信号和 PLC 函数。在手册中主要是指 Basic Program Plus 的程序块和数据类型，这些只适用于 SINUMERIK ONE。

详细信息请参见各控制系统的 PLC 功能手册。

接口信号

进行接口信号寻址时，主要采用 Basic Program Plus 的符号寻址。对应的 Basic Program 信号，请参见各章节末尾的针对不同信号方向的 PLC 信号表，如以下示例所示：

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.enc1Synchronized	LBP_Axis*.E_RefSyn1	DB31, ...DBX60.4
...		

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.refPointApproachCam	LBP_Axis*.A_DelayRef	DB31, ...DBX12.7
...		

说明

数量结构

在 Basic Program Plus 中，通过相应的对象实例来映射轴、通道等。

在 Basic Program 中进行绝对寻址时，以下列组件的最大绝对数量为基础：运行方式组 (DB11)、通道 (DB21, ...)、轴/主轴 (DB31, ...)

标准功能范畴

本文档描述了标准功能范畴。该描述可能和交付的系统的功能有所不同。交付的系统的功能仅以订购资料为准。

在系统中也可能会运行本文档中未说明的功能，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。

为使文档简明清晰，本文档并不包含所有产品类型的所有详细信息，也无法对安装、运行和维护中可能出现的各种情况逐一进行说明。

机床制造商在产品上增添或者更改的功能，由机床制造商进行说明。

第三方网页

本文档可能包含第三方网页链接。西门子对此类网页的内容不承担任何责任，也不会声明或认可此类网页或其内容为西门子所有。西门子并不能控制此类网页上的信息，也不对上述网页的内容和信息负责。使用上述网页的风险由用户承担。

1.3 网上文档

1.3.1 MC 文档一览

有关 SINUMERIK MC 功能的详细文档，自版本 1.12 起，请参见 SINUMERIK MC 文档一览 (<https://support.industry.siemens.com/cs/ww/en/view/109771043>)。



您可以直接打开文档或者下载 PDF 和 HTML5 格式。

文档分为以下几个类别：

- 用户：操作
- 用户：编程
- 制造商/服务：调试
- 制造商/服务：功能
- 制造商/服务：硬件
- 制造商/服务：Safety Integrated
- 介绍和培训

1.3.2 SINUMERIK 操作组件文档一览

有关 SINUMERIK 操作组件的全部文档，请参见 SINUMERIK 操作组件文档一览 (<https://support.industry.siemens.com/cs/document/109783841/technische-dokumentation-zu-sinumerik-bedienkomponenten?dti=0&lc=en-WW>)。

您可以直接打开文档或者下载 PDF 和 HTML5 格式。

文档分为以下几个类别：

- 操作面板
- 机床控制面板
- 机床按钮面板
- 手持单元/微型手持单元
- 其他操作组件

有关“SINUMERIK”的重要文档、文章和链接，请参见 SINUMERIK 专题页 (<https://support.industry.siemens.com/cs/document/109766201/sinumerik-an-overview-of-the-most-important-documents-and-links?dti=0&lc=en-WW>)。

1.4 技术文档反馈

对于西门子工业在线支持上发布的任何技术文档，如有疑问、建议或改进意见，请点击文章末尾的链接“发送反馈”。

1.5 mySupport 文档

使用网页版“mySupport 文档”可以自由组合西门子文档内容，创建自己的文档。

在 SiePortal 页面“mySupport 链接和工具” (<https://support.industry.siemens.com/cs/cn/zh/my>)上点击“我的文档”，便可启动应用：

mySupport 链接和工具



配置的手册可以 RTF、PDF 或 XML 格式导出。

说明

在链接“配置”下可以查看网页版“mySupport 文档”支持的西门子文档内容。

1.6 服务与支持

产品支持

有关产品的详细信息请访问网址：

产品支持 (<https://support.industry.siemens.com/cs/cn/zh/>)

在该网址下可以提供：

- 最新产品信息（产品公告）
- FAQ（常见问题与解答）
- 手册
- 下载链接
- 持续提供产品最新信息的新闻。
- “技术论坛”，供全球用户和专家交流经验、分享信息
- “联系人”，提供全球联系人信息，方便查找本地联系人
- “售后服务”，提供现场服务、维修、备件等信息

技术支持

访问网址 (<https://support.industry.siemens.com/cs/cn/zh/sc/4868>)下的“联系方式”，便可以获取各个国家技术支持的电话号码。

如需咨询技术疑问，请使用“支持请求”一栏下的在线表格。

培训

访问网址 (<https://www.siemens.com/sitrain>)，可获取有关 SITRAIN 的相关信息。

SITRAIN 为西门子的驱动和自动化产品、系统和解决方案提供培训。

无论在何处都能得到最佳的支持



使用荣获大奖的“西门子工业在线支持”App，您可以随时随地查看超过 30 万份的西门子工业领域的产品文件。该应用也可帮助您：

- 解决项目实施中出现的问题
- 排除故障
- 进行设备扩展或重新规划

此外，您还可以登录技术论坛，查看我们的专家为您撰写的其他文章：

- 常见问题与解答
- 应用实例
- 手册
- 证书
- 产品公告等

“西门子工业在线支持”App 提供 Apple iOS 版和安卓版。

铭牌上的二维码

铭牌上的二维码包含了各设备的数据。使用任一智能手机通过“西门子工业在线支持”App 扫描该二维码，便可获取相应设备的技术信息。

1.7 OpenSSL 的使用

本产品可包含以下软件：

- 由 OpenSSL 项目开发并应用在 OpenSSL Toolkit 中的软件
- 由 Eric Young 开发的加密软件
- 由 Eric Young 开发的软件

1.8 遵守通用数据保护条例

详细信息请访问网址：

- OpenSSL (<https://www.openssl.org>)
- Cryptsoft (<https://www.cryptsoft.com>)

1.8 遵守通用数据保护条例

西门子遵守通用数据保护条例，特别是隐私保护设计（privacy by design）规定。

对于本产品意味着：

产品不会处理或保存个人相关数据，只会处理或保存技术功能数据（例如：时间戳）。用户如果将此类技术功能数据与其他数据（例如：排班表）关联或者将个人相关数据存储在同一介质（例如：硬盘）上而产生个人相关性，则应由用户自行确保遵循数据保护法规。

基本安全说明

2.1 一般安全说明



警告

未遵循安全说明和遗留风险可引发生命危险

忽视随附硬件文档中的安全说明和遗留风险会导致重伤或死亡。

- 遵守硬件文档中的安全说明。
- 进行风险评估时应考虑到遗留风险。



警告

因参数设置错误或修改参数设置引起机器故障

参数设置错误可导致机器出现故障，从而导致人员重伤或死亡。

- 采取保护措施，防止未经授权的参数设置。
- 采取适当措施（如驻停或急停）处理可能出现的故障。

2.2 应用示例的质保规定

应用示例在组态和配置以及各种突发事件方面对设备没有强制约束力，无需一一遵循。应用示例不会提供客户专用的解决方案，仅在典型任务设置中提供保护。

用户自行负责上述产品的规范运行事宜。应用示例并没有解除您在应用、安装、运行和维护时确保安全环境的责任。

2.3 安全性信息

Siemens 为其产品及解决方案提供了工业信息安全功能，以支持工厂、系统、机器和网络的安全运行。

为了防止工厂、系统、机器和网络受到网络攻击，需要实施并持续维护先进且全面的工业信息安全保护机制。Siemens 的产品和解决方案构成此类概念的其中一个要素。

客户负责防止其工厂、系统、机器和网络受到未经授权的访问。只有在有必要连接时并仅在采取适当安全措施（例如，防火墙和/或网络分段）的情况下，才能将该等系统、机器和组件连接到企业网络或 Internet。

2.3 安全性信息

关于可采取的工业信息安全措施的更多信息，请访问 <https://www.siemens.com/industrialsecurity>

Siemens 不断对产品和解决方案进行开发和完善以提高安全性。Siemens 强烈建议您及时更新产品并始终使用最新产品版本。如果使用的产品版本不再受支持，或者未能应用最新的更新程序，客户遭受网络攻击的风险会增加。

要及时了解有关产品更新的信息，请订阅 Siemens 工业信息安全 RSS 源，网址为 <https://www.siemens.com/cert>

其他信息请上网查找：

工业安全功能选型手册 (<https://support.industry.siemens.com/cs/cn/zh/view/108862708/en>)

**警告**

篡改软件会引起不安全的驱动状态

篡改软件（如：病毒、木马、蠕虫等）可使设备处于不安全的运行状态，从而可能导致死亡、重伤和财产损失。

- 总是使用最新版本的软件。
- 将自动化和驱动组件集成到设备或机器上的整套先进工业信息安全方案中。
- 全面考虑整套工业信息安全方案中使用的所有产品。
- 采取相应的保护措施（如：使用杀毒软件）防止移动存储设备中的文件受到恶意软件的破坏。
- 在调试结束后，检查所有和安全相关的设置。

有关功能性的信息

有关外围设备连接可能性和功能性的信息，参见控制器调试手册。

回参考点

4.1 简要说明

功能

在机床轴回参考点时，机床轴坐标系与机床坐标系同步。为此会将机床轴运行至机床零点，之后将机床轴的实际位置设为零。

若受结构限制而无法直接驶向机床零点，则须使用一个在机床轴运行范围内定义的参考点来实现机床轴的同步。其相对于机床零点的位置必须已知。回参考点时机床轴的实际位置会设为该值。

测量系统和回参考点方式

回参考点时可采用以下测量系统和回参考点方式使机床轴同步：

- 测量系统
 - 至少有一个零脉冲的增量旋转测量系统
 - 增量线性测量系统（长度测量系统）
 - 使用距离编码参考脉冲的旋转测量系统（Heidenhain 公司）
 - 使用距离编码参考脉冲的线性测量系统（Heidenhain 公司）
 - 绝对式旋转测量系统
 - 绝对式线性测量系统（长度测量系统）
- 回参考点方式
 - 使用增量测量系统通过无接触式接近开关和单/双脉冲沿识别回参考点
 - 基于增量测量系统使用参考凸轮通过无接触式接近开关回参考点
 - 使用增量测量系统在主轴应用中通过无接触式接近开关以配置的逼近速度回参考点
 - 基于使用距离编码参考脉冲的测量系统通过越过 2 个或 4 个零脉冲回参考点
 - 通过测量系统补偿执行被动测量系统回参考点
 - 跟踪模式下回参考点
 - 通过驱动上的凸轮开关回参考点

4.2 轴专用回参考点

启动

机床轴的回参考点运行可手动启动，或通过零件程序启动：

- 手动：JOG 和 MDI 运行方式，机床功能 REF
- 零件程序：指令 G74

4.2 轴专用回参考点

进行轴专用回参考点时，必须为每根需要回参考点的机床轴单独触发回参考点运行。

选择运行方式和机床功能

在进行机床轴回参考点前，首先切换至 JOG 或 MDI 运行方式：

<ModeGroup>.basic.out.jogMode (JOG 运行方式)

<ModeGroup>.basic.out.mdaMode (MDI 运行方式)

接下来必须选择机床功能 REF（回参考点）：

<ModeGroup>.basic.out.refPoint (机床功能 REF)

启动回参考点

使用轴专用回参考点时，必须为每根轴分别启动回参考点。

回参考点运行通过轴专用运行键启动：

<Axis>.basic.out.traversKeyMinus (负向运行键)

<Axis>.basic.out.traversKeyPlus (正向运行键)

方向使能

为了避免误操作，必须对方向使能进行参数设置。通过方向使能确定通过哪个运行键启动回参考点：

MD34010 \$MA_REFP_CAM_DIR_IS_MINUS = <值>

<值>	含义
0	沿正方向回参考点
1	沿负方向回参考点

长按模式

通过以下机床数据设定，是在按下一次方向键后完全执行回参考点，还是必须由机床操作人员长按方向键（长按模式）来执行（出于安全原因）：

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

机床操作人员在长按模式下松开方向键后，机床轴会制动至静止状态。回参考点运行不会终止。再次按下方向键便可继续执行回参考点。

回参考点状态

回参考点启动时，系统会反馈机床轴的回参考点状态：

<Axis>.basic.in.enc1Synchronized（已回参考点，已同步 1）

<Axis>.basic.in.enc2Synchronized（已回参考点，已同步 2）

<Chan>.basic.in.allAxesReferenced（所有应回参考点的进给轴已回参考点）

距离编码测量系统

使用距离编码测量系统时，可通过任意运行键启动回参考点。

顺序：

进行机床轴回参考点时，机床操作人员或机床制造商（通过 PLC 用户程序）须确保遵循特定的顺序。

- 机床操作人员
机床操作人员必须按照所要求的顺序启动机床轴。
- 机床制造商
机床制造商的 PLC 用户程序更能够确保机床轴按照相应的顺序启动。

多根机床轴同时回参考点

根据控制系统，可同时为多根机床轴执行回参考点：

回参考点完成

若机床轴回参考点成功完成，系统会设置回参考点状态来进行反馈：

<Axis>.basic.in.enc1Synchronized（已回参考点，已同步 1）

<Axis>.basic.in.enc2Synchronized（已回参考点，已同步 2）

4.2 轴专用回参考点

回参考点终止

使用轴专用回参考点时，机床轴在设定为机床轴主通道的通道中运行：

MD30550 \$MA_AXCONF_ASSIGN_MASTER_CHAN

因此，为了终止回参考点运行，必须为机床轴的主通道触发运行方式组复位或通道复位：

<ModeGroup>.basic.out.reset（运行方式组复位）

<Chan>.basic.out.reset（通道复位）

运行终止时尚未完成回参考点的机床轴都会保持“未回参考点”状态：

<Axis>.basic.in.enc1Synchronized（已回参考点，已同步 1）

<Axis>.basic.in.enc2Synchronized（已回参考点，已同步 2）

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.enc1Synchronized	LBP_Axis*.E_RefSyn1	DB31, ...DBX60.4
<Axis>.basic.in.enc2Synchronized	LBP_Axis*.E_RefSyn2	DB31, ...DBX60.5
<Chan>.basic.in.allAxesReferenced	LBP_Chan*.E_AxesRef	DB21, ...DBX36.2

PLC → NC

Basic Program Plus	Basic Program	
<ModeGroup>.basic.out.mdaMode	LBP_ModeGroup.A_MDA	DB11.DBX0.1
<ModeGroup>.basic.out.jogMode	LBP_ModeGroup.A_JOG	DB11.DBX0.2
<ModeGroup>.basic.out.refPoint	LBP_ModeGroup.A_REF	DB11.DBX1.2
<Axis>.basic.out.traversKeyMinus	LBP_Axis*.A_Minus	DB31, ...DBX4.6
<Axis>.basic.out.traversKeyPlus	LBP_Axis*.A_Plus	DB31, ...DBX4.7
<ModeGroup>.basic.out.reset	LBP_ModeGroup.A_MGReset	DB11, ...DBX0.7
<Chan>.basic.out.reset	LBP_Chan*.A_Reset	DB21, ...DBX7.7

4.3 通道专用回参考点

进行通道专用回参考点时，触发回参考点运行后，系统将按照参数设置的顺序为通道的所有机床轴执行回参考点。

选择运行方式和机床功能

在进行机床轴回参考点之前，首先应将相应的 BAG 切换至 JOG 或 MDI 运行方式：

<ModeGroup>.basic.out.jogMode（JOG 运行方式）

<ModeGroup>.basic.out.mdaMode（MDI 运行方式）

接下来必须选择机床功能 REF（回参考点）：

<ModeGroup>.basic.out.refPoint（机床功能 REF）

参数设置轴顺序

通道中的机床轴执行回参考点的顺序通过以下机床数据设置：

MD34110 \$MA_REFP_CYCLE_NR = <编号>

<编号>	含义
-1	对于通道中的 NC 启动，机床轴不必为已回参考点状态。
0	机床轴不参加通道专用回参考点。
1 - 15	通道专用回参考点的顺序编号

机床轴会按照编号的升序执行回参考点。

相同编号的机床轴会同时执行回参考点

多根机床轴同时回参考点

根据控制系统，可同时为多根机床轴执行回参考点：

启动回参考点

通道专用回参考点通过以下方式启动：

<Chan>.basic.out.refPoint（激活回参考点）

通道通过以下信号提示通道专用回参考点的状态：

<Chan>.basic.in.refActive（回参考点生效）

4.3 通道专用回参考点

回参考点状态

回参考点启动时，系统会反馈机床轴的回参考点状态：

<Axis>.basic.in.enc1Synchronized（已回参考点，已同步 1）

<Axis>.basic.in.enc2Synchronized（已回参考点，已同步 2）

回参考点完成

参与通道专用回参考点的所有机床轴均成功回参考点后，系统将进行相应反馈：

<Chan>.basic.in.allAxesReferenced（所有应回参考点的进给轴已回参考点）

回参考点终止

使用通道专用回参考点时，机床轴在其作为通道轴实际指定到的通道中运行。

因此，为了终止回参考点运行，必须为对应通道触发运行方式组复位或通道复位：

<ModeGroup>.basic.out.reset（运行方式组复位）

<Chan>.basic.out.reset（通道复位）

运行终止时尚未完成回参考点的机床轴都会保持“未回参考点”状态：

<Axis>.basic.in.enc1Synchronized（已回参考点，已同步 1）

<Axis>.basic.in.enc2Synchronized（已回参考点，已同步 2）

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.enc1Synchronized	LBP_Axis*.E_RefSyn1	DB31, ...DBX60.4
<Axis>.basic.in.enc2Synchronized	LBP_Axis*.E_RefSyn2	DB31, ...DBX60.5
<Chan>.basic.in.allAxesReferenced	LBP_Chan*.E_AxesRef	DB21, ...DBX36.2
<Chan>.basic.in.refActive	LBP_Chan*.E_Ref	DB21, ...DBX33.0

PLC → NC

Basic Program Plus	Basic Program	
<ModeGroup>.basic.out.mdaMode	LBP_ModeGroup.A_MDA	DB11.DBX0.1
<ModeGroup>.basic.out.jogMode	LBP_ModeGroup.A_JOG	DB11.DBX0.2

Basic Program Plus	Basic Program	
<ModeGroup>.basic.out.reset	LBP_ModeGroup.A_MGReset	DB11.DBX0.7
<ModeGroup>.basic.out.refPoint	LBP_ModeGroup.A_REF	DB11.DBX1.2
<Chan>.basic.out.refPointApproach	LBP_Chan*.A_Ref	DB21, ...DBX1.0
<Chan>.basic.out.reset	LBP_Chan*.A_Reset	DB21, ...DBX7.7

4.4 在零件程序中写入回参考点（G74）

功能

借助指令 G74 可通过零件程序或同步动作为机床轴执行首次回参考点，或重新回参考点。

例如在以下情形下便需要重新回参考点：

- 实际值偏移 PRESETON
- 超出位置测量系统的编码器极限频率

编程

句法

G74 <机床轴> {<机床轴>}

含义

G74:	用于机床轴回参考点的指令	
	在单独程序段中编程:	是
	有效性:	逐段生效
<机床轴>:	必须指定机床轴的名称（MD10000 \$MN_AXCONF_MACHAX_NAME_TAB）。 该机床轴必须为执行相应零件程序或同步动作的通道通道轴。	

4.5 使用增量测量系统回参考点

复位特性

运行方式组复位或通道复位会使编写的所有机床轴回参考点运行终止：

- <ModeGroup>.basic.out.reset（运行方式组复位）
- <Chan>.basic.out.reset（通道复位）

复位时尚未完成回参考点的机床轴都会保持“未回参考点”状态：

- <Axis>.basic.in.enc1Synchronized（已回参考点，已同步 1）== 0
- <Axis>.basic.in.enc2Synchronized（已回参考点，已同步 2）== 0

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.enc1Synchronized	LBP_Axis*.E_RefSyn1	DB31, ...DBX60.4
<Axis>.basic.in.enc2Synchronized	LBP_Axis*.E_RefSyn2	DB31, ...DBX60.5

PLC → NC

Basic Program Plus	Basic Program	
<ModeGroup>.basic.out.reset	LBP_ModeGroup.A_MGReset	DB11.DBX0.7
<Chan>.basic.out.reset	LBP_Chan*.A_Reset	DB21, ...DBX7.7

4.5 使用增量测量系统回参考点

4.5.1 硬件信号

根据机床结构和所使用的增量测量系统的属性，必须连接不同的硬件信号。

参考凸轮

- 连接
参考凸轮信号可连接至外部 PLC I/O 模块的数字量输入，或 NCU 接口 X142 的快速输入。
- NC/PLC 接口信号
参考凸轮信号必须由 PLC 用户程序传输至轴专用 NC/PLC 接口：
<Axis>.basic.out.refPointApproachCam（回参考点减速）

零脉冲选择

若在轴或主轴回参考点期间识别出多个测量系统零脉冲（例如电机和编码器间的测量齿轮箱），必须通过一个附加的无接触式接近开关信号选择指定的零脉冲。

- 连接
无接触式接近开关必须连接至 NCU 接口 X122 或 X132 的一个快速数字量输入。
- 激活
为了分析无接触式接近开关信号，必须在驱动参数 p0493 中为轴/主轴选择连接无接触式接近开关的数字量输入。

替代零脉冲

若使用的测量系统不能提供零脉冲信号，可通过无接触式接近开关信号生成一个替代零脉冲。

- 连接
无接触式接近开关必须连接至 NCU 接口 X122 或 X132 的一个快速数字量输入。
- 激活
为了分析无接触式接近开关信号，必须在驱动参数 p0494 或 p0495 中为轴/主轴选择连接无接触式接近开关的数字量输入。

概述

信号	接口：数字输入端，通过	设置
参考凸轮	外部 PLC 外设模块 或 NCU：X142	PLC 用户程序： <Axis>.basic.out.refPointApproachCam
零脉冲选择	NCU：X122 或 X132	驱动参数：p0493
外部零脉冲或替代零脉冲	NCU：X122 或 X132	驱动参数：p0494 或 p0495

PLC 信号**PLC → NC**

Basic Program Plus	Basic Program	
<Axis>.basic.out.refPointApproachCam	LBP_Axis*.A_DelayRef	DB31, ...DBX12.7

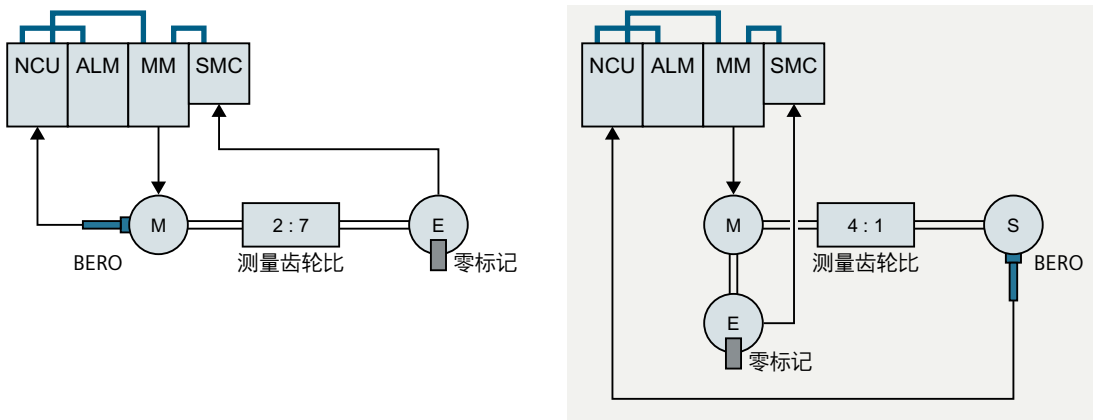
其它信息

- NCU 接口：产品手册
 连接 > 数字量输入/输出
- 驱动参数：SINAMICS S120/S150 参数手册

4.5.2 零脉冲选择

功能

增量测量系统的回参考点基于编码器零脉冲相对机床轴整个运行范围的唯一位置。若在特定的机床条件下能够在机床轴的运行范围内识别出多个编码器零脉冲（示例见下图），则须在机床上启用无接触式接近开关来确定唯一的零脉冲。此时参考点的位置由无接触式接近开关信号和编码器零脉冲的信号组合得出。



BER 无接触式接近开关
O

图 4-1 电机和编码器间的测量齿轮箱，或电机和主轴间的降档齿轮箱

参数设置

NC：回参考点模式

回参考点模式必须设置为“增量、旋转或线性测量系统的回参考点：编码器信号上的零脉冲”：

MD34200 \$MA_ENC_REFP_MODE[<轴>] = 1

驱动：零脉冲选择

在参数 p0493 中必须设置连接无接触式接近开关的 NCU 接口数字量输入。

说明

零脉冲选择

无接触式接近开关信号只在驱动中处理。连接和参数设置参见“硬件信号 (页 42)”章节。

4.5.3 时序

使用增量测量系统回参考点的时序分为三个阶段：

- 阶段 1：“阶段 1：运行到参考凸轮 (页 47)”
- 阶段 2：“阶段 2：与零脉冲同步 (页 50)”
- 阶段 3：“阶段 3：运行至参考点 (页 55)”

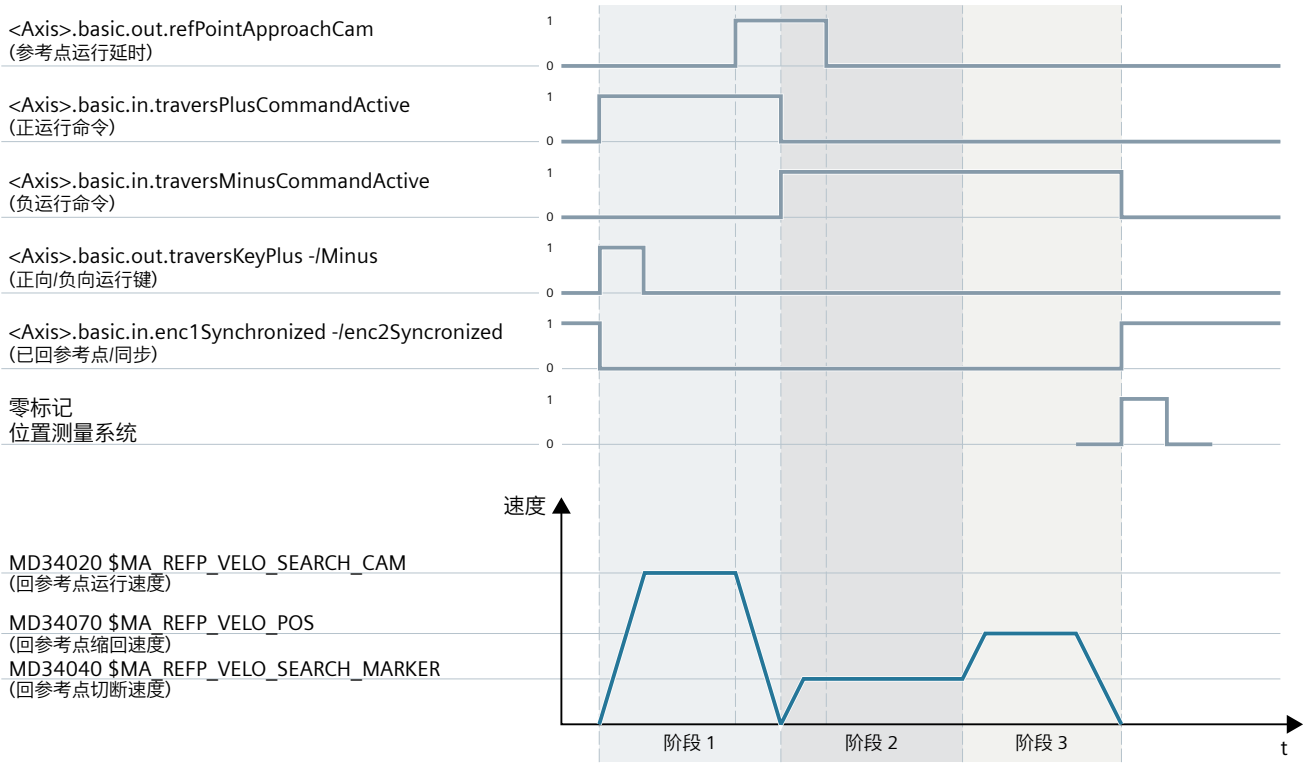


图 4-2 使用增量测量系统回参考点时的时序（示例）

4.5 使用增量测量系统回参考点

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.enc1Synchronized	LBP_Axis*.E_RefSyn1	DB31, ...DBX60.4
<Axis>.basic.in.enc2Synchronized	LBP_Axis*.E_RefSyn2	DB31, ...DBX60.5
<Axis>.basic.in.traversMinusCommandActive	LBP_Axis*.E_TCMinus	DB31, ...DBX64.6
<Axis>.basic.in.traversPlusCommandActive	LBP_Axis*.E_TCPlus	DB31, ...DBX64.7

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.refPointApproachCam	LBP_Axis*.A_DelayRef	DB31, ...DBX12.7
<Axis>.basic.out.traversKeyMinus	LBP_Axis*.A_Minus	DB31, ...DBX4.6
<Axis>.basic.out.traversKeyPlus	LBP_Axis*.A_Plus	DB31, ...DBX4.7

4.5.4 阶段 1：运行到参考凸轮

阶段 1：图形显示

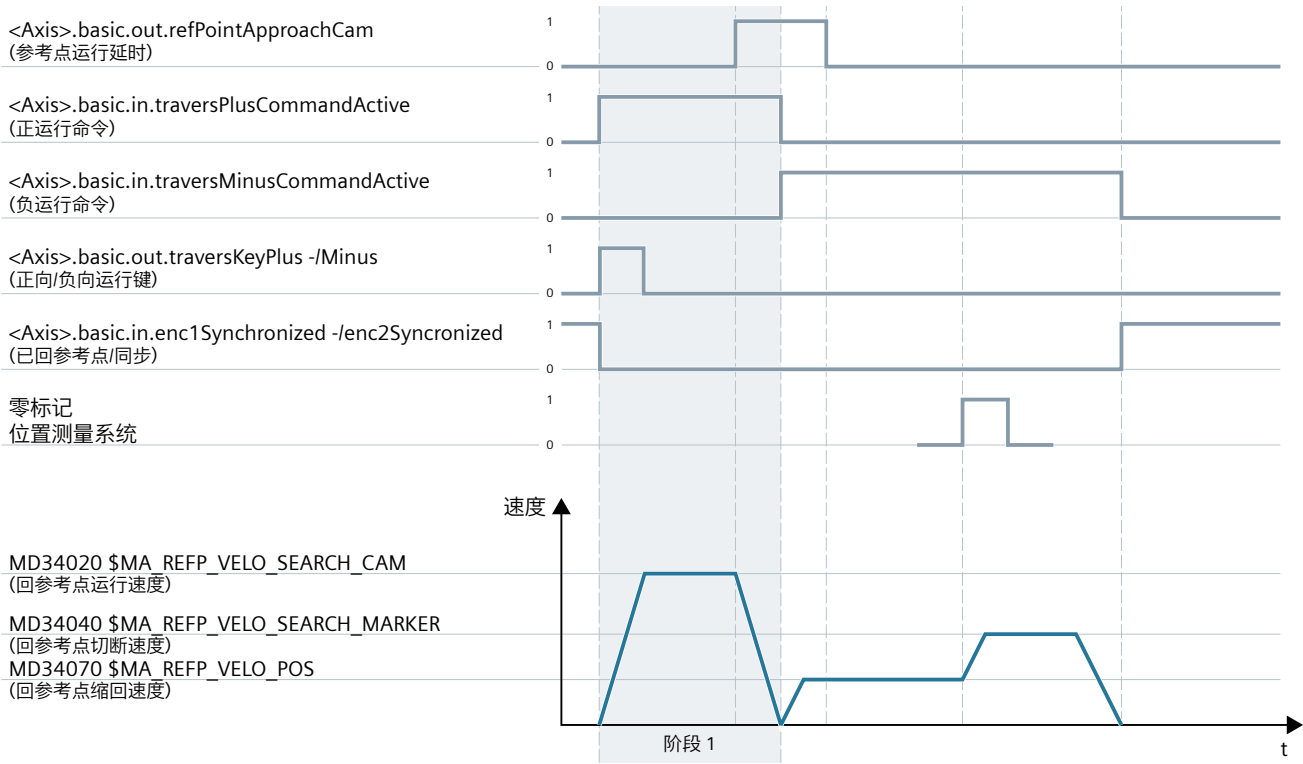


图 4-3 阶段 1：运行到参考凸轮

阶段 1：启动

回参考点运行的启动参见“轴专用回参考点 (页 36)”和“通道专用回参考点 (页 39)”章节。

阶段 1：过程

在阶段 1 中，根据机床轴相对参考凸轮的位置分为三种情形：

- 1. 机床轴位于参考凸轮前
- 2. 机床轴位于参考凸轮上
- 3. 机床轴无参考凸轮

4.5 使用增量测量系统回参考点

情形 1：机床轴位于参考凸轮前

启动回参考点运行后，机床轴沿设定的方向加速至参数设置的回参考点速度：

- MD34010 \$MA_REFP_CAM_DIR_IS_MINUS（沿负方向逼近参考点）
- MD34020 \$MA_REFP_VELO_SEARCH_CAM（回参考点速度）

必须在 PLC 用户程序中查询数字量输入来识别“到达参考凸轮”，并通过以下接口信号将此信息提供给 NC：

<Axis>.basic.out.refPointApproachCam（回参考点减速）

识别出 NC/PLC 接口信号时，机床轴将制动直至到达静止状态。此时至少还会运行 s_{min} 的行程。这样可确保在阶段 2 中机床轴能够以参数设置的参考点停止速度离开参考凸轮。

$$s_{min} = \frac{(MD34040 \$MA\_REFP\_VELO\_SEARCH\_MARKER)^2}{2 * MD32300 \$MA\_MAX\_AX\_ACCEL}$$

这样一来阶段 1 结束。之后将继续执行回参考点阶段 2。

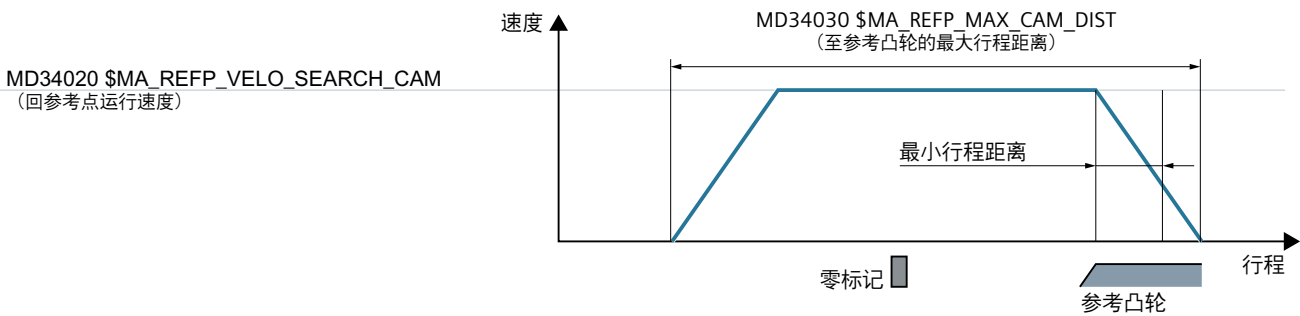


图 4-4 减速的最小行程

情形 2：机床轴位于参考凸轮上

机床轴停留在初始位置。
这样一来阶段 1 结束。之后将继续执行回参考点阶段 2。

情形 3：机床轴无参考凸轮

无参考凸轮的机床轴将停留在初始位置。

其中例如包括：

- 整个运行范围内只有一个零脉冲的机床轴
- 每转只有一个零脉冲的回转轴

对于无参考凸轮的机床轴，以下机床数据必须设置为零：

MD34000 \$MA_REFP_CAM_IS_ACTIVE = 0（带参考凸轮的轴）

这样一来阶段 1 结束。之后将继续执行回参考点阶段 2。

阶段 1：属性

- 进给倍率生效。
- 进给停止（通道专用和轴专用）生效。
- NC 停止和 NC 启动均生效。
- 在参数设置的最大行程内无法到达参考凸轮时，机床轴将停止：
MD34030 \$MA_REFP_MAX_CAM_DIST（至参考凸轮的最大行程距离）

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.enc1Synchronized	LBP_Axis*.E_RefSyn1	DB31, ...DBX60.4
<Axis>.basic.in.enc2Synchronized	LBP_Axis*.E_RefSyn2	DB31, ...DBX60.5
<Axis>.basic.in.traversMinusCommandActive	LBP_Axis*.E_TCMinus	DB31, ...DBX64.6
<Axis>.basic.in.traversPlusCommandActive	LBP_Axis*.E_TCPlus	DB31, ...DBX64.7

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.refPointApproachCam	LBP_Axis*.A_DelayRef	DB31, ...DBX12.7
<Axis>.basic.out.traversKeyMinus	LBP_Axis*.A_Minus	DB31, ...DBX4.6
<Axis>.basic.out.traversKeyPlus	LBP_Axis*.A_Plus	DB31, ...DBX4.7

4.5 使用增量测量系统回参考点

4.5.5 阶段 2：与零脉冲同步

阶段 2：图形显示

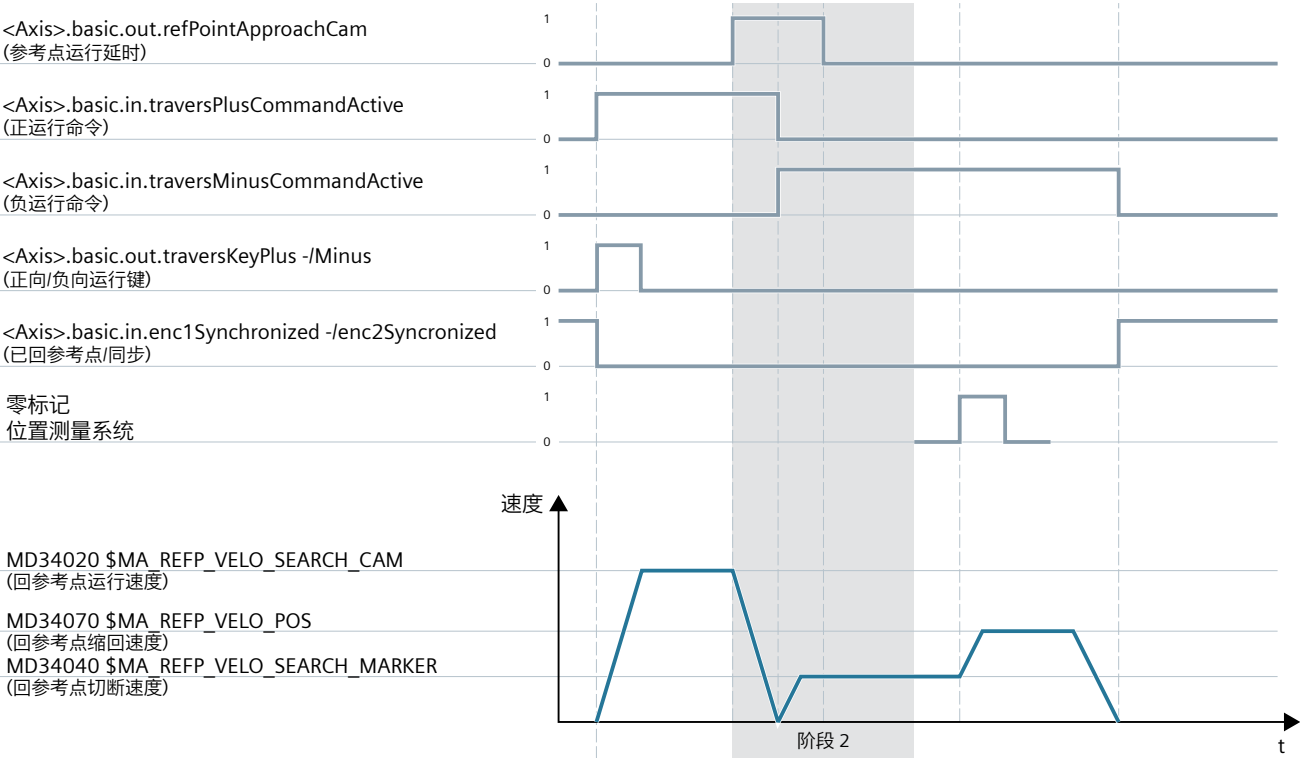


图 4-5 阶段 2：与零脉冲同步

阶段 2：启动

若第 1 阶段未发生报警且成功完成，则自动启动第 2 阶段。

初始情况：

机床轴位于参考凸轮上。

零脉冲搜索方向：

零脉冲搜索方向取决于以下机床数据中的设置：

- MD34010 \$MA_REFP_CAM_DIR_IS_MINUS（沿负方向逼近参考点）
- MD34050 \$MA_REFP_SEARCH_MARKER_REVERSE（参考凸轮上换向）

阶段 2：过程

阶段 2 中的同步可通过参考凸轮的下降沿或上升沿实现。这通过以下机床数据设置：

MD34050 \$MA_REFP_SEARCH_MARKER_REVERSE[<轴>] = <值>

值	含义
0	通过参考凸轮下降沿同步
1	通过参考凸轮上升沿同步

说明

若机床轴逼近参考凸轮时的实际速度尚未达到阶段 2 设定速度（参数设置的公差限值内），则会重新启动阶段 1。例如，启动回参考点时机床轴位于参考凸轮上时便会出现此情形。

MD35150 \$MA_SPIND_DES_VELO_TOL（主轴转速公差）

情形 1：通过参考凸轮下降沿同步

通过参考凸轮下降沿同步时，机床轴会逆向于设置的回参考点方向（阶段 1 的运行方向）加速至参数设置的参考点停止速度。

离开参考凸轮后会等待下一个编码器零脉冲：

`<Axis>.basic.out.refPointApproachCam == 0`

识别出编码器零脉冲时，阶段 2 结束。机床轴以恒定速度运行，并继续执行回参考点阶段 3。

- MD34040 \$MA_REFP_VELO_SEARCH_MARKER（参考点停止速度）

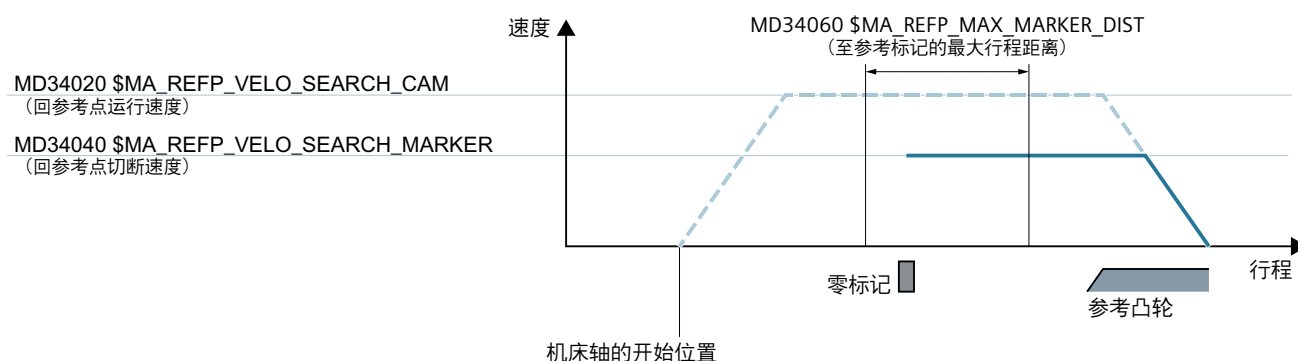


图 4-6 通过参考凸轮下降沿同步

情形 2：通过参考凸轮上升沿同步

通过参考凸轮上升沿同步时，机床轴会逆向于设置的回参考点方向（阶段 1 的运行方向）加速至参数设置的回参考点速度。

- MD34020 \$MA_REFP_VELO_SEARCH_CAM（回参考点速度）
- MD34010 \$MA_REFP_CAM_DIR_IS_MINUS（沿负方向逼近参考点）

4.5 使用增量测量系统回参考点

离开参考凸轮后机床轴会向静止状态制动：

```
<Axis>.basic.out.refPointApproachCam == 0
```

之后机床轴通过参数设置的参考点停止速度返回参考凸轮：

MD34040 \$MA_REFP_VELO_SEARCH_MARKER（参考点停止速度）

到达参考凸轮 (<Axis>.basic.out.refPointApproachCam = 1) 后，等待下一个编码器零脉冲。

识别出编码器零脉冲时，阶段 2 结束。机床轴以恒定速度运行，并继续执行回参考点阶段 3。

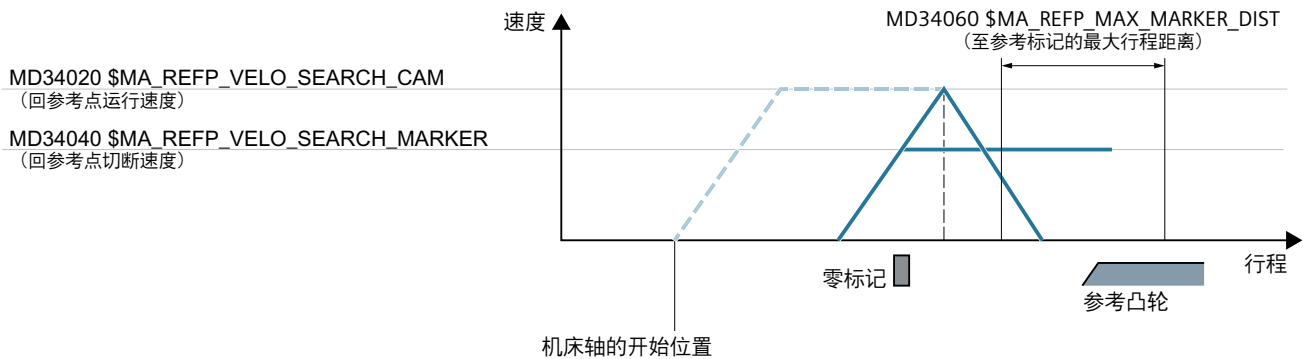


图 4-7 通过参考凸轮上升沿同步

电子参考凸轮偏移

电子参考凸轮偏移用于补偿温度造成的参考凸轮长度延展，从而确保始终与相同的编码器零脉冲同步。

MD34092 \$MA_REFP_CAM_SHIFT（提供等距零脉冲的增量测量系统的电子参考凸轮偏移）

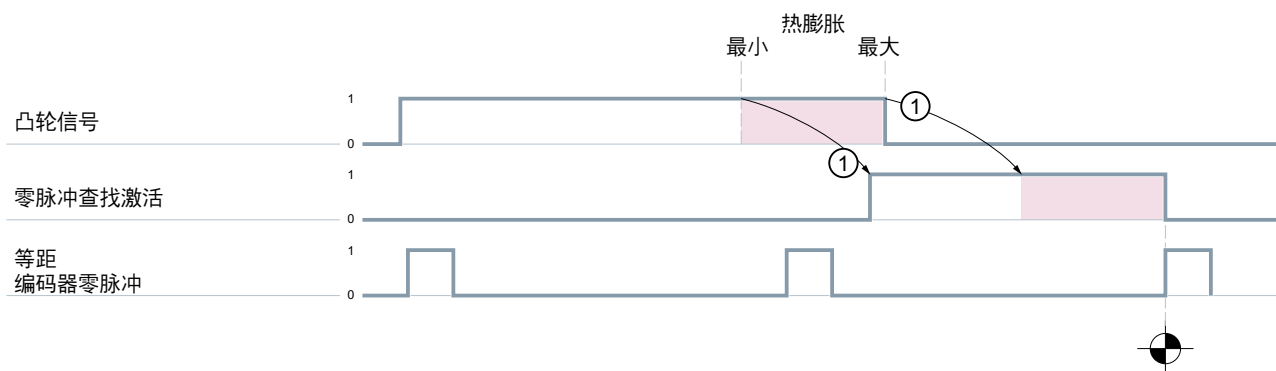
启用电子参考凸轮偏移时，识别出参考凸轮脉冲沿后不会立即与下一个编码器零脉冲同步，而是在经过设置的偏移后再进行同步。

生效的偏移 s_{shift} 基于自识别出参考凸轮脉冲沿起插补周期中经过的位移得出：

$$s_{\text{shift\_min}} = \text{MD34092 \$MA\_REFP\_CAM\_SHIFT}$$

$$s_{\text{shift\_max}} = \text{MD34092 \$MA\_REFP\_CAM\_SHIFT} + \text{MD34040 \$MA\_REFP\_VELO\_SEARCH\_MARKER} * \text{插补周期}$$

电子参考凸轮偏移始终在零脉冲搜索方向上生效。



① 参考凸轮偏移

图 4-8 电子参考凸轮偏移

前提条件

电子参考凸轮偏移仅适用于带参考凸轮的机床轴：

MD34000 \$MA_REFP_CAM_IS_ACTIVE == 1

参考凸轮校准

提供等距零脉冲的编码器

对于等距提供零脉冲的编码器，必须精确校准参考凸轮，确保回参考点运行中始终能识别正确的零脉冲。

时间特性

从到达机床上的参考凸轮，直到识别出从 PLC 用户程序传输至 NC 的参考凸轮信号，这期间的时间特性受以下因素影响：

- 参考凸轮开关的精度
- 参考凸轮开关的延时（常闭触点）
- PLC 输入上的延时
- PLC 循环时间
- NC/PLC 接口更新所需的循环时间
- 插补周期
- 位置控制周期

4.5 使用增量测量系统回参考点

设置说明

- 参考凸轮
实际应用中，在两个零脉冲中央校准参考凸轮的脉冲沿是可靠方案。
- 电子参考凸轮偏移

 **警告**

碰撞危险
参考凸轮的校准错误或不精确时，可能会分析错误的零脉冲。此时控制系统会因此确定错误的机床零点。这样一来机床轴会运行至错误的位置。软件限位开关、保护区和工作区域限制也将对错误的位置生效，这样便无法保护机床。位移差相当于机床轴两个零脉冲间经过的行程的 +/- 值。

以下只读机床数据用作设置电子参考凸轮偏移时的参考信息：
MD34093 \$MA_REFP_CAM_MARKER_DIST（参考凸轮/参考标记距离）
该机床数据显示的值对应从离开参考凸轮到出现参考标记时之间的距离。若该值过小，则存在温度影响或凸轮信号运行时间浮动造成参考点不确定的危险。

阶段 2：属性

- 进给倍率**不**生效。
内部按进给倍率 = 100% 运行。
若进给倍率设为 0%，则终止运行。
- 进给停止（通道专用和轴专用）生效。
- NC 停止和 NC 启动均**不**生效。
- 若阶段 2 中机床轴未在参数设置的行程内到达参考脉冲（编码器零脉冲），那么机床轴将停止：
MD34060 \$MA_REFP_MAX_MARKER_DIST（至参考脉冲的最大行程）

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.enc1Synchronized	LBP_Axis*.E_RefSyn1	DB31, ...DBX60.4
<Axis>.basic.in.enc2Synchronized	LBP_Axis*.E_RefSyn2	DB31, ...DBX60.5
<Axis>.basic.in.traversMinusCommandActive	LBP_Axis*.E_TCMinus	DB31, ...DBX64.6
<Axis>.basic.in.traversPlusCommandActive	LBP_Axis*.E_TCPlus	DB31, ...DBX64.7

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.refPointApproachCam	LBP_Axis*.A_DelayRef	DB31, ...DBX12.7
<Axis>.basic.out.traversKeyMinus	LBP_Axis*.A_Minus	DB31, ...DBX4.6
<Axis>.basic.out.traversKeyPlus	LBP_Axis*.A_Plus	DB31, ...DBX4.7

4.5.6 阶段 3：运行至参考点

阶段 3：图形显示

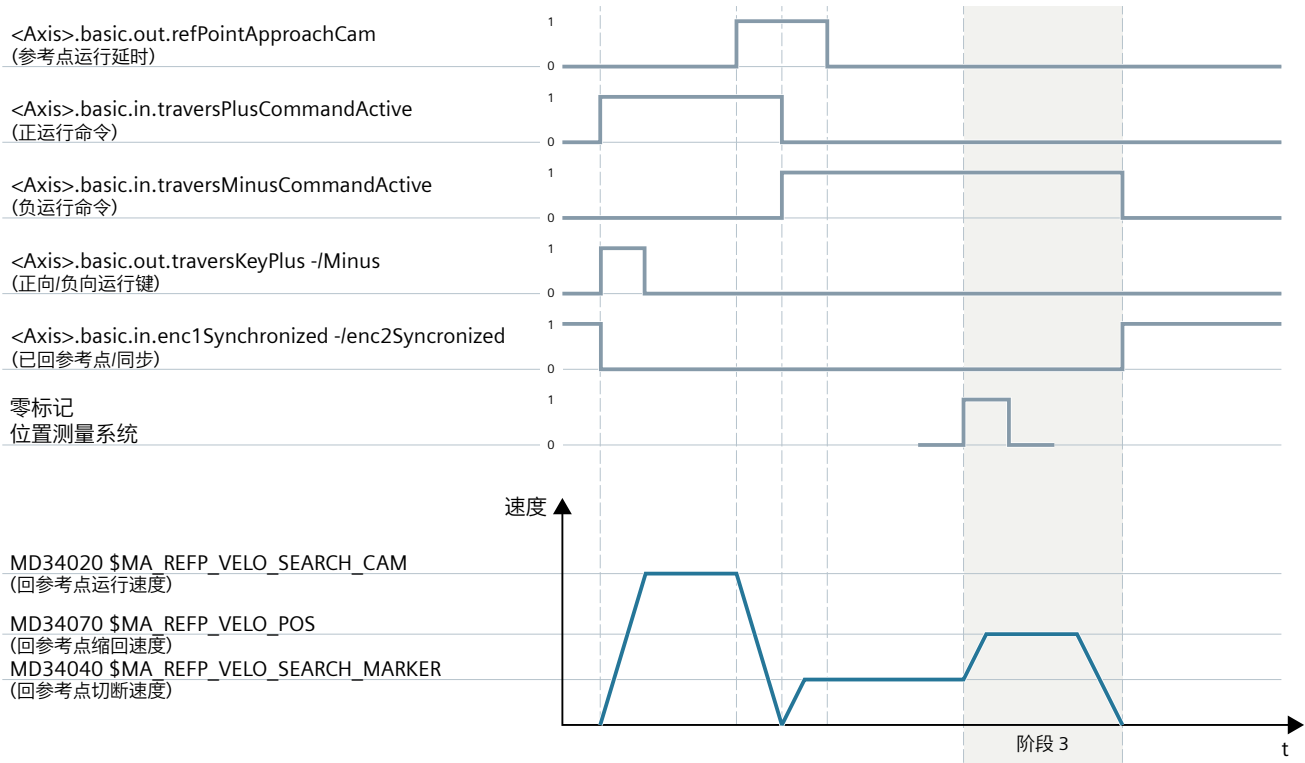


图 4-9 阶段 3：运行至参考点

阶段 3：启动

在阶段 2 的末尾，机床轴以参考点停止速度运行。因此，一旦阶段 2 在未触发报警的情形下顺利结束，便会立即切换至阶段 3。

4.5 使用增量测量系统回参考点

初始状态

已识别编码器零脉冲。

阶段 3：过程

机床轴通过参数设置的参考点定位速度：

MD34070 \$MA_REFP_VELO_POS（参考点定位速度）

从阶段 2 中识别的编码器零脉冲运行至参考点。

此时待运行的行程 s_{ref} 为参考点距离和参考点偏移之和：

MD34080 \$MA_REFP_MOVE_DIST（参考点距离）

MD34090 \$MA_REFP_MOVE_DIST_CORR（参考点偏移）

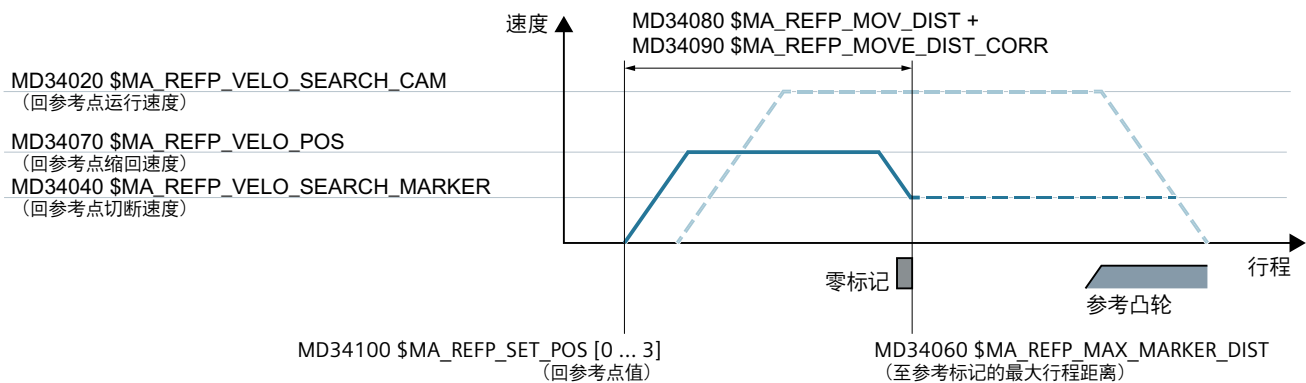


图 4-10 参考点位置

到达参考点后，机床轴将停止，实际值系统会与通过 NC/PLC 接口设定的参考点值 n 进行同步：

MD34100 \$MA_REFP_SET_POS[<n>]（参考点值）

参考点通过以下 NC/PLC 接口选择：

<Axis>.basic.out.refPointSelection（参考点值 1 ... 4）

实际值系统会与阶段 1 中到达参考凸轮 (<Axis>.basic.out.refPointApproachCam == 1) 时选择的参考点值进行同步。

就此机床轴已完成回参考点。根据生效的测量系统，系统会设置以下接口信号作为对 PLC 用户程序的反馈信息：

<Axis>.basic.in.enc1Synchronized / <Axis>.basic.in.enc2Synchronized
(已回参考点/已同步 1 / 2) = 1

阶段 3 的属性

- 进给倍率生效。
- 进给停止（通道专用和轴专用）生效。
- NC 停止和 NC 启动均生效。

阶段 3 的特殊性

在下列情形下，机床轴在识别出零脉冲后先会停止，并于随后返回参考点：

- 参考点距离和参考点偏移之和小于所需的制动行程（基于参考点定位速度）：
 $MD34080 + MD34090 < \text{所需制动行程（基于 MD34070）}$
- 在与当前运行方向相反的方向上，参考点位于参考凸轮“后”。

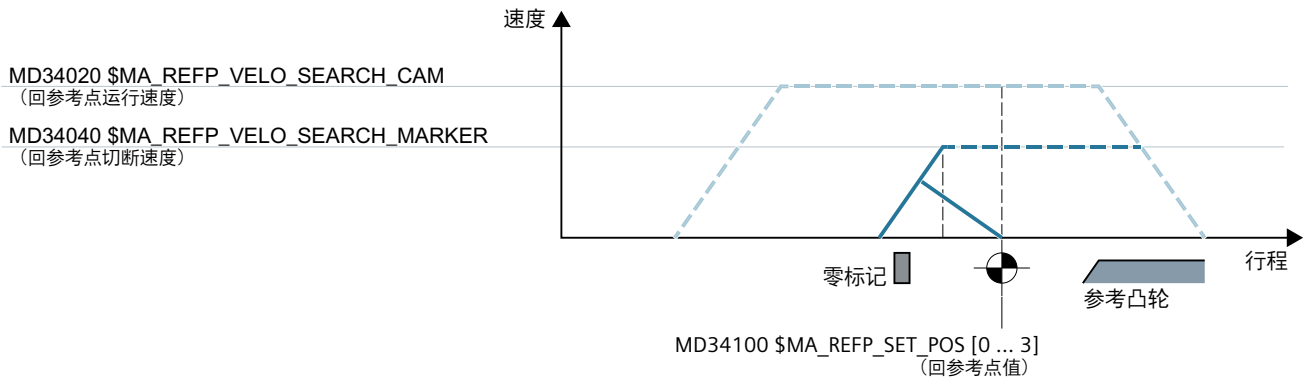


图 4-11 参考点距离和参考点偏移之和小于制动行程

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.enc1Synchronized	LBP_Axis*.E_RefSyn1	DB31, ...DBX60.4
<Axis>.basic.in.enc2Synchronized	LBP_Axis*.E_RefSyn2	DB31, ...DBX60.5
<Axis>.basic.in.traversMinusCommandActive	LBP_Axis*.E_TCMinus	DB31, ...DBX64.6
<Axis>.basic.in.traversPlusCommandActive	LBP_Axis*.E_TCPlus	DB31, ...DBX64.7

4.6 通过距离编码的参考脉冲回参考点

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.refPointApproachCam	LBP_Axis*.A_DelayRef	DB31, ...DBX12.7
<Axis>.basic.out.traversKeyMinus	LBP_Axis*.A_Minus	DB31, ...DBX4.6
<Axis>.basic.out.traversKeyPlus	LBP_Axis*.A_Plus	DB31, ...DBX4.7
<Axis>.basic.out.refPointSelection	LBP_Axis*.A_RefVal1..4	DB31, ...DBX2.4..7

4.6 通过距离编码的参考脉冲回参考点

4.6.1 概述

距离编码的参考脉冲

采用距离编码参考脉冲的测量系统由两个并行的标度信号构成：

- 增量光栅
- 参考脉冲信号

每两个相邻参考脉冲之间的距离已定义。当运行超过两个相邻参考脉冲后，可由此确定机床轴的绝对位置。例如：假设参考脉冲间距大约为 10 mm，则机床轴回参考点时仅需大约 20 mm 的运行路径即可。

可从任意的轴位置出发沿正方向或负方向回参考点（特例：运行区域末端）。

4.6.2 基础参数设置

线性测量系统

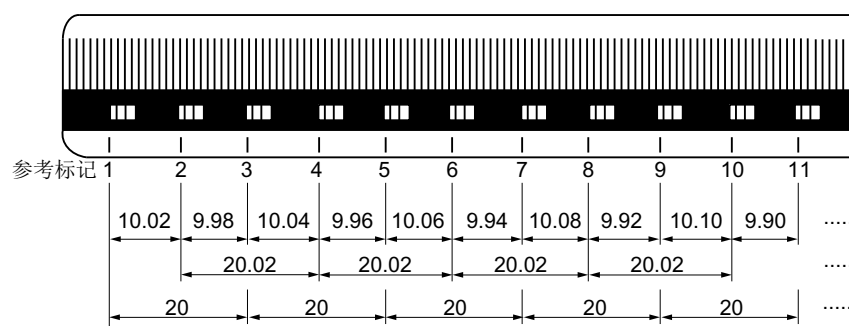


图 4-12 采用距离编码参考脉冲的玻璃量具，分度区段：20 mm

通过以下数据参数设置线性测量系统：

- 机床零点和长度测量系统第一个参考脉冲位置之间的绝对偏移量：
MD34090 \$MA_REFP_MOVE_DIST_CORR（参考点偏移/绝对偏移）
参见：确定绝对偏移
- 长度测量系统相对于机床坐标系的方向设置（相同或相反）：
MD34320 \$MA_ENC_INVERS（长度测量系统与机床坐标系方向相反）

旋转测量系统

旋转测量系统的规则与线性测量系统相同（参见上文）。

确定绝对偏移

确定机床零点和机床轴第一个参考脉冲位置之间的绝对偏移量时，建议如下操作：

1. 将绝对偏移量数值设置为零：
MD34090 \$MA_REFP_MOVE_DIST_CORR = 0
2. 执行回参考点操作。
提示：回参考点的执行位置应根据以下标准选择：例如能使用激光干涉仪轻松测得机床轴相对于机床零点的准确位置。
3. 通过操作界面（例如 SINUMERIK Integrate）读取显示在机床坐标系 MCS 中的机床轴实际位置。

4.6 通过距离编码的参考脉冲回参考点

- 4. 以机床零点为基准，测量机床轴的当前位置。
- 5. 计算出绝对偏移量并输入 MD34090 中。
根据测量系统相对机床坐标系的定向，绝对偏移量如下计算：

测量系统的方向	绝对偏移量
相同	(所测得的位置) + (显示的实际位置)
相反	(所测得的位置) - (显示的实际位置)

 **警告**

参考点偏差
得到绝对偏移量并填入 MD34090 后，必须重新启动机床轴的回参考点运行。

回参考点模式

通过距离编码的参考脉冲回参考点时，可采用以下两种模式：

- 分析计算相邻**两个**参考脉冲：
MD34200 \$MA_ENC_REFP_MODE = 3
优点：
 - 运行路径较短
- 分析计算相邻**四个**参考脉冲：
MD34200 \$MA_ENC_REFP_MODE = 8
优点：
 - 可通过 NC 进行合理性检查
 - 提高回参考点结果的安全性

4.6.3 时序

时序

通过距离编码的参考脉冲回参考点时，过程分为 2 个阶段：

- 阶段 1：同步越过参考脉冲
- 阶段 2：运行至固定目标点

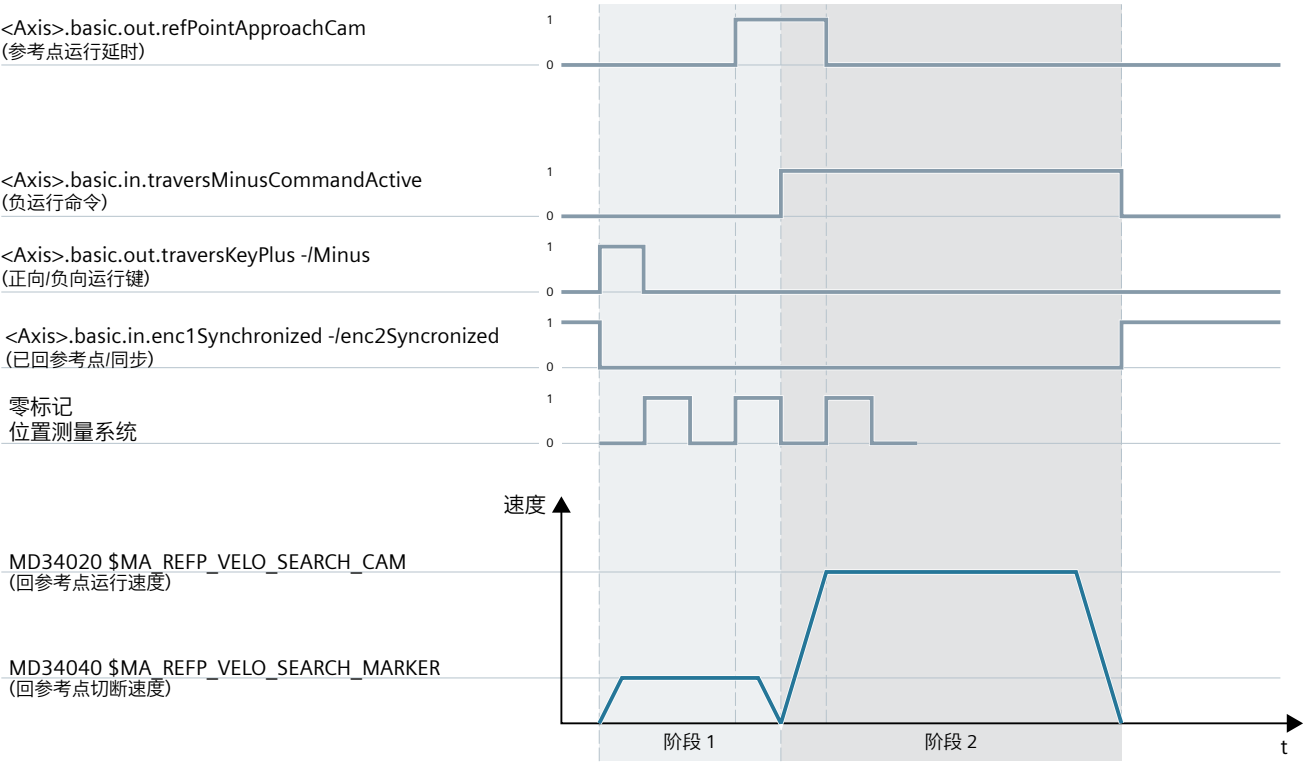


图 4-13 距离编码的参考脉冲

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.enc1Synchronized	LBP_Axis*.E_RefSyn1	DB31, ...DBX60.4
<Axis>.basic.in.enc2Synchronized	LBP_Axis*.E_RefSyn2	DB31, ...DBX60.5
<Axis>.basic.in.traversMinusCommandActive	LBP_Axis*.E_TCMinus	DB31, ...DBX64.6
<Axis>.basic.in.traversPlusCommandActive	LBP_Axis*.E_TCPlus	DB31, ...DBX64.7

4.6 通过距离编码的参考脉冲回参考点

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.refPointApproachCam	LBP_Axis*.A_DelayRef	DB31, ...DBX12.7
<Axis>.basic.out.traversKeyMinus	LBP_Axis*.A_Minus	DB31, ...DBX4.6

4.6.4 阶段 1：同步越过参考脉冲

阶段 1：启动

回参考点运行的启动参见“轴专用回参考点 (页 36)”和“通道专用回参考点 (页 39)”章节。

参考凸轮

对于使用距离编码的参考脉冲的测量系统，回参考点本来无需任何参考凸轮。出于功能性考虑，在通道专用回参考点运行和使用零件程序回参考点运行中（G74）须在机床轴的运行范围末端安装参考凸轮。

阶段 1：过程

未触碰到参考凸轮的过程

启动回参考点运行后，机床轴加速至设置的参考点停止速度：

MD34040 \$MA_REFP_VELO_SEARCH_MARKER（参考点停止速度）

越过设置数量的参考脉冲后，机床轴再次停止，其实际值系统同步至 NC 获得的绝对位置。

在参考凸轮上启动的过程

若启动回参考点运行时机床轴位于参考凸轮上，则其将加速至设置的参考点停止速度，运行方向与设置的回参考点运行方向相反：

MD34040 \$MA_REFP_VELO_SEARCH_MARKER（参考点停止速度）

MD34010 \$MA_CAM_DIR_IS_MINUS（沿负方向回参考点）

如此可保证机床轴在未越过设定数量的参考脉冲前不会撞上运行范围限制点。

越过设置数量的参考脉冲后，机床轴再次停止，其实际值系统同步至 NC 获得的绝对位置。

回参考点途中触碰到参考凸轮的过程

启动回参考点运行后，机床轴加速至设置的参考点停止速度：

MD34040 \$MA_REFP_VELO_SEARCH_MARKER（参考点停止速度）

机床轴在越过设定数量的参考脉冲前即触碰到了参考凸轮。此种情况下，机床轴会反向运行并沿相反的方向重新启动参考脉冲搜索。

越过设定数量的参考脉冲后，机床轴再次停止，其实际值系统同步至 NC 获得的绝对位置。

参考脉冲间距合理性检查

若 NC 在回参考点运行过程中探测到两个相邻参考脉冲的间距大于编写的参考脉冲间距的两倍，则表示存在错误：

MD34300 \$MA_ENC_REFP_MARKER_DIST（参考脉冲间距）

此种情况下，机床轴会以设置的参考点停止速度（MD34040）的一半沿反方向运行并重新启动参考脉冲搜索。

若又再次发现参考脉冲间距有误，则机床轴会停止并终止回参考点运行（报警 20003“测量系统故障”）。

终止标准

若在编写的行程中未识别出设定数量的参考脉冲，机床轴会停止并终止回参考点运行：

MD34060 \$MA_REFP_MAX_MARKER_DIST（至参考脉冲的最大行程）

阶段 1 的属性

阶段 1 成功结束后，机床轴实际值系统即完成同步。

4.6.5 阶段 2：运行到目标点

阶段 2：启动

若阶段 1 未发生报警且成功完成，则自动启动阶段 2。

初始情况：

- 机床轴直接停在设定数量的参考脉冲中的最后一个上。
- 机床轴的实际值系统已同步。

4.6 通过距离编码的参考脉冲回参考点

阶段 2：过程

为完成回参考点运行，机床轴将在阶段 2 中运行至定义的目标位置（参考点）。若需要缩短回参考点的运行路径，也可抑制此过程进行：

```
MD34330 $MA_STOP_AT_ABS_MARKER = <值>
```

值	含义
0	运行至目标位置
1	不运行至目标位置

运行至目标位置（通常情况）

机床轴加速至设置的参考点定位速度并运行至编写的目标点（参考点）：

```
MD34070 $MA_REFP_VELO_POS（回参考点速度）
```

```
MD34100 $MA_REFP_SET_POS（参考点值）
```

就此机床轴已完成回参考点。根据生效的测量系统，NC 会设置对应的接口信号作为提示：

```
<Axis>.basic.in.enc1Synchronized / <Axis>.basic.in.enc2Synchronized  
（已回参考点/已同步 1/2）= 1
```

不运行至目标位置

就此机床轴已完成回参考点。根据生效的测量系统，NC 会设置对应的接口信号作为提示：

```
<Axis>.basic.in.enc1Synchronized / <Axis>.basic.in.enc2Synchronized  
（已回参考点/已同步 1/2）= 1
```

阶段 2 的属性

根据是否为机床轴编写了参考凸轮，阶段 2 将具有不同特性。

无参考凸轮的机床轴

```
MD34000 $MA_REFP_CAM_IS_ACTIVE（带参考凸轮的轴）= 0
```

属性：

- 进给倍率生效。
- 进给停止（通道专用和轴专用）生效。
- NC 停止和 NC 启动均生效。

带参考凸轮的机床轴

```
MD34000 $MA_REFP_CAM_IS_ACTIVE（带参考凸轮的轴）= 1
```


属性:

- 进给倍率**不**生效。
内部按进给倍率 = 100% 运行。
若进给倍率设为 0%，则终止运行。
- 进给停止（通道专用和轴专用）生效。
- NC 停止和 NC 启动均**不**生效。
- 离开参考凸轮后，若未在编写的行程中识别出设定数量的参考脉冲，那么机床轴将停止：
MD34060 \$MA_REFP_MAX_MARKER_DIST（至参考脉冲的最大行程）

旋转测量系统的特点

在使用距离编码的旋转测量系统中，只能在旋转范围内确定唯一的绝对位置。由于编码器机械结构的差别，硬件方面的绝对位置溢出并不总是与回转轴的运行范围吻合。

模数回转轴上的特殊性

对于模数回转轴，参考点位置映射至编写的模数区域：

MD30330 \$MA_MODULO_RANGE（模数区域的大小）

MD30340 \$MA_MODULO_RANGE_START（模数区域起点）

说明

也可通过轴功能“在配置的模数区域内确定使用距离编码的旋转编码器的参考点位置”将参考点位置映射至编写的（虚拟）模数区域：

MD30455 \$MA_MISC_FUNCTION_MASK（轴功能），位 1 = 1

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.enc1Synchronized	LBP_Axis*.E_RefSyn1	DB31, ...DBX60.4
<Axis>.basic.in.enc2Synchronized	LBP_Axis*.E_RefSyn2	DB31, ...DBX60.5

4.7 通过实际值补偿回参考点

4.7.1 对回参考点测量系统进行实际值补偿

功能

在对回参考点测量系统进行实际值补偿中，根据成功执行的机床轴测量系统回参考点运行，系统会将得到的绝对位置实际值直接传输至该机床轴的其他测量系统，并将其作为已回参考点：

```
<Axis>.basic.in.enc1Synchronized / <Axis>.basic.in.enc2Synchronized  
(已回参考点/已同步 1/2) = 1
```

优点

在从实际回参考点的测量系统转换为通过实际值补偿回参考点的测量系统时，机床轴可继续保持闭环控制下（伺服使能生效），因为补偿的实际值不会产生实际值跃变。

说明

为了获取测量系统专用的编码器信息来提升定位精度，建议在测量系统切换后对通过实际值补偿回参考点的测量系统重新执行回参考点。

激活

对回参考点测量系统的实际值补偿针对机床激活：

```
MD34102 $MA_REFP_SYNC_ENCS = 1
```

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.enc1Synchronized	LBP_Axis*.E_RefSyn1	DB31, ...DBX60.4
<Axis>.basic.in.enc2Synchronized	LBP_Axis*.E_RefSyn2	DB31, ...DBX60.5

4.7.2 使用距离编码参考脉冲的测量系统中的实际值补偿

功能

为了获取测量系统专用的编码器信息来提升定位精度，建议在测量系统切换后对通过实际值补偿回参考点的测量系统重新执行回参考点。

若将使用距离编码参考脉冲的编码器用于被动测量系统，那么在满足以下前提条件时可不执行回参考点：

1. 主动测量系统：间接测量系统（电机测量系统），例如使用绝对值编码器
2. 被动测量系统：使用距离编码参考脉冲的直接测量系统
3. 切换测量系统前，对配备已回参考点的间接测量系统的机床轴执行运行，使其越过回参考点所需数量的零脉冲。此时被动直接测量系统自动成为已回参考点状态。

参数设置

除了针对各测量系统的回参考点专用机床数据外，还须设置以下机床数据：

- 使能实际值补偿：
MD34102 \$MA_REFP_SYNC_ENCS = 1
- 使用距离编码参考标记的直接测量系统：
 - MD34200 \$MA_ENC_REFP_MODE[测量系统] = 3
距离编码的参考脉冲
 - MD30242 \$MA_ENC_IS_INDEPENDENT[测量系统] = 2
实际值补偿中，系统将为被动直接测量系统补偿生效的间接测量系统的位置实际值，但是不会将其标记为已回参考点。在越过设定数量的参考脉冲后，该被动直接测量系统自动成为已回参考点状态。回参考点可在任意运行方式下进行。

过程

1. 初始情况：两个测量系统均未回参考点：
 <Axis>.basic.in.enc1Synchronized = 0（已回参考点/已同步 1）
 <Axis>.basic.in.enc2Synchronized = 0（已回参考点/已同步 2）
2. 根据测量系统类型对间接测量系统执行回参考点：
 <Axis>.basic.in.enc1Synchronized = 1（已回参考点/已同步 1）
 <Axis>.basic.in.enc2Synchronized = 0（已回参考点/已同步 2）
3. 运行机床轴使其越过设定数量的参考脉冲。
 借此直接测量系统自动回参考点：
 <Axis>.basic.in.enc1Synchronized = 1（已回参考点/已同步 1）
 <Axis>.basic.in.enc2Synchronized = 1（已回参考点/已同步 2）

4.8 跟踪模式下回参考点

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.enc1Synchronized	LBP_Axis*.E_RefSyn1	DB31, ...DBX60.4
<Axis>.basic.in.enc2Synchronized	LBP_Axis*.E_RefSyn2	DB31, ...DBX60.5

4.8 跟踪模式下回参考点

功能

增量测量系统和使用距离编码参考脉冲的测量系统也可在机床轴的跟踪模式下进行回参考点。前提条件是根据使用的测量系统对回参考点运行进行了正确地设置（参见“使用增量测量系统回参考点 (页 42)”和“通过距离编码的参考脉冲回参考点 (页 58)”章节）。

在跟踪模式下的回参考点中，机床轴不是通过 NC，而是借助穿过编码器零脉冲或设定数量的距离编码参考脉冲的外部运行来移动。识别出编码器零脉冲或设定数量的距离编码参考脉冲时，测量系统即完成回参考点。

说明

回参考点结果的可重复性

在 NC 控制的回参考点运行中，在运行期间遵循参数设置的运行速度即可保证回参考点结果的可重复性。在跟踪模式下回参考点时，机床制造商/用户必须采取合适的措施保证回参考点结果的可重复性。

零脉冲的唯一性

增量测量系统的回参考点基于编码器零脉冲相对机床轴整个运行范围的唯一位置。

在跟踪模式下回参考点时不会由 NC 分析参考凸轮信号，因此在未采取其他措施时，只能在以下情形下确定唯一的参考点：

- 机床轴运行范围内只有一个编码器零脉冲。
- 使用绝对编码参考脉冲的线性测量系统。
- 模数回转轴（一次旋转内的绝对位置）

存在多个零脉冲信号时的零脉冲选择

若在特定的机床条件下（例如编码器和负载间的降档齿轮箱）能够在机床轴的运行范围内识别出多个编码器零脉冲，则须将一个无接触式接近开关信号连接至 NCU 接口的数字量输入，从而通过该信号来确定唯一的零脉冲。

说明

无接触式接近开关信号：零脉冲选择

无接触式接近开关信号只在驱动中处理。连接和参数设置参见“硬件信号 (页 42)”章节。

使能

“跟踪模式下回参考点”功能通过以下机床数据使能：

MD34104 \$MA_REFP_PERMITTED_IN_FOLLOWUP = TRUE

启动回参考点运行

启动回参考点运行时，若机床轴处于跟踪模式下 (`<Axis>.basic.in.followUpModeActive == TRUE`)，那么测量系统会在跟踪模式下进行回参考点。

若机床轴在启动回参考点运行时不处于跟踪模式下，则会执行“一般的”由 NC 控制的回参考点运行。

跟踪模式下的回参考点可在以下运行方式中启动：

- JOG-REF：运行键
- AUTOMATIC：零件程序指令 G74

回参考点运行的过程（JOG-REF 运行方式）

1. 激活机床轴跟踪模式：
`<Axis>.basic.out.followUpMode（跟踪模式）= 1`
`<Axis>.basic.out.enableCtrl（伺服使能）= 0`
2. 等待激活跟踪运行：
`<Axis>.basic.in.followUpModeActive（跟踪生效）== 1`
3. 切换至 JOG 运行方式，机床功能 REF。

4.8 跟踪模式下回参考点

4. 外部运行机床轴，使其越过编码器零脉冲或设定数量的距离编码参考脉冲。机床轴移动后，立即在 NC 内部启动回参考点：系统会复位以下 NC/PLC 接口信号作为反馈信息：
`<Axis>.basic.in.isStationary`（轴/主轴停止）== 0
5. 成功识别出编码器零脉冲或设定数量的距离编码参考脉冲时，测量系统即完成回参考点。系统会置位以下 NC/PLC 接口信号作为反馈信息：
`<Axis>.basic.in.enc1Synchronized/<Axis>.basic.in.enc2Synchronized`
（已回参考点/已同步 1/2）== 1

终止回参考点运行

以下情形下，生效的回参考点运行将终止：

- 取消跟踪模式
- NC 复位

已回参考点的测量系统的特性

已回参考点的测量系统只能在 AUTOMATIC 运行方式下通过零件程序指令 G74 重新执行回参考点。

回参考点运行的过程（AUTOMATIC 运行方式）

1. 切换至 AUTOMATIC 运行方式。
2. 启动零件程序。
3. 激活机床轴跟踪模式：
`<Axis>.basic.out.followUpMode`（跟踪模式）= 1
`<Axis>.basic.out.enableCtrl`（伺服使能）= 0
4. 等待激活跟踪运行：
`<Axis>.basic.in.followUpModeActive`（跟踪生效）== 1
5. 执行指令 G74 后，立即在 NC 内部启动回参考点：
6. 外部运行机床轴，使其越过编码器零脉冲或设定数量的距离编码参考脉冲。
7. 成功识别出编码器零脉冲或设定数量的距离编码参考脉冲时，测量系统即完成回参考点。系统会置位以下 NC/PLC 接口信号作为反馈信息：
`<Axis>.basic.in.enc1Synchronized/<Axis>.basic.in.enc2Synchronized`
（已回参考点/已同步 1/2）== 1
8. 回参考点运行完成后执行程序段转换。

终止回参考点运行

以下情形下，生效的回参考点运行将终止：

- 取消跟踪模式
- NC 复位

已回参考点的测量系统的特性

已回参考点的测量系统可重新执行回参考点。

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.enc1Synchronized	LBP_Axis*.E_RefSyn1	DB31, ...DBX60.4
<Axis>.basic.in.enc2Synchronized	LBP_Axis*.E_RefSyn2	DB31, ...DBX60.5
<Axis>.basic.in.followUpModeActive	LBP_Axis*.E_Follow_upMode	DB31, ...DBX61.3
<Axis>.basic.in.isStationary	LBP_Axis*.E_Stat	DB31, ...DBX61.4

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.followUpMode	LBP_ModeGroup.A_Follow_upMode	DB31, ...DBX1.4
<Axis>.basic.out.enableCtrl	LBP_Chan*.A_ContrEnable	DB31, ...DBX2.1

4.9 使用绝对值编码器回参考点

4.9.1 校准信息

使用绝对值编码器的机床轴

使用绝对值编码器的机床轴的优势在于：在经过一次校准后，机床轴的实际值系统将立即与确定的绝对位置同步，无需像增量测量系统一样执行回参考点运行（例如控制系统启动、取消机床轴“驻留”等）。

校准

校准绝对值编码器时，编码器的实际值会通过机床零点一次性补偿，之后设置生效。

机床轴连接的绝对值编码器的当前校准状态会在以下轴专用机床数据中显示：

4.9 使用绝对值编码器回参考点

MD34210 \$MA_ENC_REFP_STATE（绝对值编码器的状态）

值	含义
0	编码器未校准
1	编码器校准已使能
2	编码器已校准

校准方式

支持以下校准方式：

- 通过输入参考点偏移校准
- 通过输入参考点值校准
- 通过测头自动校准
- 通过无接触式接近开关校准

重新校准

以下情形下必须重新校准绝对值编码器：

- 负载和绝对值编码器之间的齿轮档切换
- 拆卸/安装绝对值编码器
- 拆卸/安装带绝对值编码器的电机
- 静态 NC 存储器中数据丢失

- 电池掉电
- 实际值设置（PRESETON）

 **警告**

数据备份

备份机床 A 的机床数据时，机床轴的编码器状态（MD34210）也会一同被备份。

在将该数据组载入至相同类型的机床 B 时（例如批量调试或维修后），NC 会将相关机床轴自动视为以校准/已回参考点。在此类型情形下，用户/机床制造商须自行负责执行重新校准。

另见机床数据说明：

MD30250 \$MA_ACT_POS_ABS（关闭时间点的绝对编码器位置）

说明

控制系统只会在以下情形下识别出绝对值编码器需要重新校准：

- 齿轮箱切换，传动比切换
- 零脉冲监控响应
- 更换绝对值编码器后出现新的编码器序列号

在这些情形下控制系统会将绝对值编码器的状态设置为“0”。

MD34210 \$MA_ENC_REFP_STATE = 0（编码器未校准）

此时会显示以下报警：

报警 25022“轴 <轴名称> 编码器 <编号> 警告 0”

零脉冲监控响应时还会额外显示以下报警：

报警 25020“轴 <轴名称> 零脉冲监控生效编码器”

在所有其他情形下（例如 PRESETON），用户须自行通过将编码器状态**手动**设置为“0”来取消校准，之后重新执行校准。

4.9.2 通过输入参考点偏移校准

功能

在通过输入参考凸轮偏移校准时，先会确定操作界面中显示的位置实际值和机床中的实际值的差值，并将其作为参考点偏移提供给 NC。

4.9 使用绝对值编码器回参考点

步骤

1. 测定仪机床零点为基准的机床轴位置，例如通过：
 - 测量位置（例如激光干涉仪）
 - 将机床轴运行至一个已知位置（例如固定停止点）
2. 在操作界面上读取显示的机床轴位置实际值。
3. 计算参考点偏移（第 1 步和第 2 步中获取的位置实际值的差值）并输入机床数据：
MD34090 \$MA_REFP_MOVE_DIST_CORR（参考点偏移）
4. 此时该绝对值编码器被标记为已校准：
MD34210 \$MA_ENC_REFP_STATE = 2

说明

编码器校准要在下一次激活编码器（例如控制系统启动）时才生效。

5. 上电，触发复位。
6. 在操作界面上检查显示的机床轴位置。

说明

间隙补偿

若为使用绝对值编码器的测量系统设置了间隙补偿，则须注意：
将机床轴运行至校准机床位置时不得存在间隙！

持续激活参考点偏移

上电复位后，输入的参考点偏移（MD34090）才会持续生效。校准完绝对值编码器后，若在未上电复位的情况下运行机床轴，那么机床数据中输入的参考点偏移可能会在内部溢出补偿中被改写。

检查位置实际值

在校准绝对值编码器后，建议在下次控制系统启动（上电）后检验机床轴的位置实际值。

4.9.3 通过输入参考点值校准

功能

在通过输入参考点值校准时，会以机床零点为基准测定机床轴的绝对位置，例如通过：

- 测量位置（例如激光干涉仪）
- 将机床轴运行至一个已知位置（例如固定停止点）

测得的值作为参考点值提供给 NC。之后 NC 会通过编码器绝对值和参考点值的差值计算出对应的参考点偏移。

步骤

1. 检查回参考点模式是否被设为“接收参考点值”：
MD34200 \$MA_ENC_REFP_MODE == 0
如果没有，则在机床数据中输入值 0 并触发上电复位。
2. 在 JOG 运行方式下将机床轴运行至待测位置（例如：机床干涉仪）或已知位置（例如：固定挡块）。

说明

可通过运行键沿“为回参考点运行使能的方向”运行机床轴：

MD34010 \$MA_REFP_CAM_DIR_IS_MINUS（沿负方向逼近参考点）

为了避免驱动组中的间隙引起的位置错误，必须以较低的速度逼近已知位置。

3. 将以机床零点为基准的机床轴位置作为参考点值输入到以下机床数据中：
MD34100 \$MA_REFP_SET_POS = 位置
4. 使能编码器校准：
MD34210 \$MA_ENC_REFP_STATE = 1
5. 触发上电复位以接收输入的机床数据值。
6. 切换至 JOG-REF 运行方式。
7. 按下步骤 2 中使用过的、使能用于回参考点的运行键。
按下运行键时机床轴不会移动！
NC 通过输入的参考点值和绝对值编码器发送的值计算出参考点偏移。计算结果记录至以下机床数据：
MD34090 \$MA_REFP_MOVE_DIST_CORR（参考点偏移）
该绝对值编码器的状态会被设置为“编码器已校准”：
MD34210 \$MA_ENC_REFP_STATE = 2
机床轴的实际值系统会进行同步。
就此机床轴已完成回参考点。根据生效的测量系统，NC 会置位对应的结构信号作为反馈：
<Axis>.basic.in.enc1Synchronized/<Axis>.basic.in.enc2Synchronized
(已回参考点/已同步 1/2) = 1
8. 上电，触发复位。

说明

持续激活参考点偏移

上电复位后，输入的参考点偏移（MD34090）才会持续生效。

校准完绝对值编码器后，若在未上电复位的情况下运行机床轴，那么机床数据中输入的参考点偏移可能会在内部溢出补偿中被改写。

检查位置实际值

在校准绝对值编码器后，建议在下一控制系统启动（上电）后检验机床轴的位置实际值。

4.9 使用绝对值编码器回参考点

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.enc1Synchronized	LBP_Axis*.E_RefSyn1	DB31, ...DBX60.4
<Axis>.basic.in.enc2Synchronized	LBP_Axis*.E_RefSyn2	DB31, ...DBX60.5

4.9.4 通过测头自动校准

功能

使用测头自动校准时，机床轴会通过零件程序逼近机床中的一个已知位置。该位置值作为参考点值保存在 NC 中。一旦测头切换，即表示到达该位置，此时 NC 会通过编码器值和参考点值的差值计算出对应的参考点偏移。

说明

用于自动校准的零件程序

机床制造商/用户必须根据机床规格自行创建用于通过测头自动校准的零件程序。

无碰撞

由于没有为待回参考点的机床轴启用实际值监控，因此机床轴运行期间操作人员尤其要确保机床中不会发生碰撞。

零件程序

用于通过测头自动校准绝对值编码器的零件程序必须能为每根待回参考点的轴按顺序执行以下操作：

- 1. 逼近机床轴的校准位置，该位置通过测头响应进行识别。
逼近应从相同的方向以逐级降低的速度分多次执行，从而获取尽可能高的测量精度。测量值保存在系统变量 \$AA_IM 中。
- 2. 计算和写入参考点偏移：

MD34090 \$MA_REFP_MOVE_DIST_CORR = MD34100 \$MA_REFP_SET_POS – \$AA_IM

- 3. 将绝对值编码器状态设置为“编码器已校准”。
MD34210 \$MA_ENC_REFP_STATE = 2

过程

通过测头自动校准时，执行以下步骤：

1. 使能零件程序启动，包含未回参考点的机床轴：
MD20700 \$MC_REFP_NC_START_LOCK = 0
2. 为所有相关机床轴输入测头切换时机床轴相对机床零点的位置，作为参考点值：
MD34100 \$MA_REFP_SET_POS = 参考点值
3. 触发 NC 复位以接收输入的机床数据值。
4. 启动零件程序。
5. 零件程序运行完毕后，在未回参考点的机床轴上重新禁用零件程序启动：
MD20700 \$MC_REFP_NC_START_LOCK = 1
6. 触发上电复位，从而使通过零件程序写入的参考点偏移持续生效：
MD34090 \$MA_REFP_MOVE_DIST_CORR (参考点偏移)

说明

持续激活参考点偏移

上电复位后，输入的参考点偏移（MD34090）才会持续生效。

校准完绝对值编码器后，若在未上电复位的情况下运行机床轴，那么机床数据中输入的参考点偏移可能会在内部溢出补偿中被改写。

检查位置实际值

在校准绝对值编码器后，建议在下一控制系统启动（上电）后检验机床轴的位置实际值。

4.9.5 通过 BERO 校准

功能

在通过无接触式接近开关校准时，和使用增量测量系统时一样需要向定义的机床位置执行回参考点运行。此时无接触式接近开关用于代替使用绝对值编码器时不存在的编码器零脉冲。回参考点运行完成后，NC 会通过编码器绝对值和参数设置的参考点值的差值自动计算出参考点偏移。

参数设置

NC：回参考点模式

回参考点模式必须设置为“增量、旋转或线性测量系统的回参考点：编码器信号上的零脉冲”：

MD34200 \$MA_ENC_REFP_MODE[<轴>] = 1

4.9 使用绝对值编码器回参考点

NC：参考点值

参考点值通过以下机床数据设置：

```
MD34100 $MA_REFP_SET_POS[<轴>] = <参考点值>
```

驱动：替代零脉冲

在参数 p0494 或 p0495 中必须设置连接无接触式接近开关的 NCU 接口数字量输入。

执行

回参考点运行可在 JOG-REF 运行方式下手动启动，或在 AUTOMATIC 或 MDI 运行方式下通过零件程序（G74）启动。

回参考点运行完成后，绝对值编码器变为已校准状态，机床轴的实际值系统也完成同步。

根据生效的测量系统，NC 会为机床轴置位以下 NC/PLC 接口信号，作为对 PLC 用户程序的反馈：

```
<Axis>.basic.in.enc1Synchronized/<Axis>.basic.in.enc2Synchronized  
(已回参考点/已同步 1 / 2) = 1
```

说明

若在绝对值编码器校准后移除无接触式接近开关，则须将回参考点模式修改为“使用绝对值编码器回参考点”：

```
MD34200 $MA_ENC_REFP_MODE[<轴>] = 0
```

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.enc1Synchronized	LBP_Axis*.E_RefSyn1	DB31, ...DBX60.4
<Axis>.basic.in.enc2Synchronized	LBP_Axis*.E_RefSyn2	DB31, ...DBX60.5

4.9.6 使用绝对值编码器回参考点

设置

运行使能

机床轴将校准过的绝对值编码器作为生效测量系统时，若为其触发回参考点运行（在 JOG-REF 运行方式下手动触发，或通过零件程序指令 G74 自动触发），那么该轴的运动特性取决于设置的运行使能：

MD34330 \$MA_REFP_STOP_AT_ABS_MARKER = <值>

值	含义
0	运行已使能。 触发回参考点后，机床轴运行至回参考点位置。到达参考点后，回参考点运行结束。
1	运行未使能。 触发回参考点后，机床轴不运行，回参考点运行立即结束。

4.9.7 通过旋转绝对值编码器启用替代零脉冲回参考点

功能

为了在使用绝对值编码器时也能采用零脉冲回参考点（参见章节“使用增量测量系统回参考点 (页 42)”），则在使用绝对值编码器时不要根据硬件对现有零脉冲进行模拟。这样，在编码器的每转中控制系统都会在同一位置上生成一个替代零脉冲信号。

与通过增量编码器回参考点的区别

启用替代零脉冲的绝对值编码器不能视作完全等同于增量编码器。绝对值编码器的所有属性将保留。下表列出了增量编码器和绝对值编码器的不同属性：

表格 4-1 增量编码器和绝对值编码器的属性

属性	增量编码器	绝对值编码器
编码器类型	MD30240 \$MA_ENC_TYPE =	
	1	4
内部编码器位置	MD30250 \$MA_ACT_POS_ABS =	
	值仅在 MD34210 ≥ 1 时更新	值仅在 MD30270 = 0 时更新

4.9 使用绝对值编码器回参考点

属性	增量编码器	绝对值编码器
运行范围扩展	MD30270 \$MA_ENC_ABS_BUFFERING =	
	无作用	= 0（默认）：生效
参考点偏移	MD34090 \$MA_REFP_MOVE_DIST_CORR =	
	允许输入数值	只由控制系统更新数值
不支持的回参考点类型	MD34200 \$MA_ENC_REFP_MODE =	
	1, 3, 4, 8	0, 1
校准状态	MD34210 \$MA_ENC_REFP_STATE = 0, 1, 2	
	运行中关闭时，编码器校准自动失效	包含位置跃变的参数值切换或序列号变更时，编码器校准自动失效
模数区域绝对位置	MD34220 \$MA_ENC_ABS_TURNS_MODULO =	
	0	1 - 4096
编码器序列号	MD34230 \$MA_ENC_SERIAL_NUMBER =	
	0	每次编码器切换时必须由 PLC 更新数值，否则校准和报警会丢失。
接收批量调试文件	无限制	由于编码器属性（MD30250、MD30270、MD34090、MD34210、MD34220、MD34230）必须加以限制。
激活时间	0 秒	数秒
零脉冲	编码器每转 1 个	无
零脉冲监控	硬件	软件
上电后无实际值缓存时的位置	0.0	MD34220 内的最后位置
	MD34210 = 0	MD30270 = 1
上电后有实际值缓存时的位置	关闭前最后的静态位置	断电期间最后位置，含微小运行
	MD34210 = 1	MD30270 = 0
上电后的回参考点状态	取决于校准状态	

前提条件

此功能只适用于旋转绝对值编码器：

- MD31000 \$MA_ENC_IS_LINEAR == 0
- MD30240 \$MA_ENC_TYPE == 4

参数设置

通过替代零脉冲回参考点：

MD34200 \$MA_ENC_REFP_MODE = 1

功能的基本条件

- 参考点偏移（MD34090 \$MA_REFP_MOVE_DIST_CORR）不得进行参数设置：
在使用绝对值编码器时，此参数描述的是机床零点和绝对值编码器零点之间的偏移，此处含义则发生变化。
- 负载侧零脉冲搜索速度 MD34040 \$MA_REFP_VELO_SEARCH_MARKER 不可超出编码器绝对信号极限频率 MD36302 \$MA_ENC_FREQ_LIMIT_LOW。
速度过大时会无法读取绝对信息，也就无法生成替代零脉冲。
- 如果在所设置的路径上没有找到零脉冲（MD34060 \$MA_REFP_MAX_MARKER_DIST），则会显示报警：
- 除了最后位置外，若还需在绝对值编码器上掉电保持回参考点状态，必须设置以下机床数据：

MD34210 \$MA_ENC_REFP_STATE = 2

数据备份与批量调试

绝对值编码器的某些属性会限制调试存档向其他机床的传输。在控制系统上运行了调试存档后，必须检查以下机床数据并进行必要的修改：

- MD30250 \$MA_ACT_POS_ABS（内部编码器位置）
- MD30270 \$MA_ENC_ABS_BUFFERING（运行范围扩展）
- MD34090 \$MA_REFP_MOVE_DIST_CORR（绝对偏移）
- MD34210 \$MA_ENC_REFP_STATE（校准状态）
- MD34220 \$MA_ENC_ABS_TURNS_MODULO（模数区域）
- MD34230 \$MA_ENC_SERIAL_NUMBER（编码器序列号）

4.9 使用绝对值编码器回参考点

4.9.8 激活测量系统

在以下情形下，机床轴的测量系统会被激活：

- 控制系统启动（上电）
- 通过接口信号激活测量系统（取消“驻留”）：
 <Axis>.basic.out.enc1Activation /
 <Axis>.basic.out.enc2Activation（位置测量系统 1/2）
 <Axis>.basic.out.enableCtrl（伺服使能）
- 低于参数设置的编码器极限频率（主轴）：
 MD36300 \$MA_ENC_FREQ_LIMIT

测量系统激活时，NC 会将机床轴的实际值系统同步为当前绝对值。同步期间轴运行会被禁止，主轴则不会。

参数设置编码器极限频率（主轴）

Heidenhain 公司的绝对值编码器 EQN 1325 具有一个增量信号和一个绝对信号。

若使用超出增量信号编码器极限频率的转速运行主轴，必须将相比要小得多的绝对信号极限频率设置为编码器极限频率：

MD36300 \$MA_ENC_FREQ_LIMIT

否则，测量系统激活时低于设置的编码器极限频率将导致读取的绝对位置错误。这会引起机床轴实际值系统中的位置偏移。

确定编码器极限频率

待设置的编码器极限频率通过以下极限转速中较小的一个得出，或通过两个转速得出：

- 编码器
极限转速或编码器极限频率从编码器数据页获取（例如极限转速 2000 [rpm]）
- NC
取决于 NC 内部的分析过程，在确保 NC 能正确测算绝对值的情形下，极限转速为每个插补周期 4 编码器旋转。
例如，在插补周期为 12 ms 时：极限转速 = $4 / 12 \text{ ms} = 20000 \text{ rpm}$
极限转速对应的极限频率如下计算：

$$\text{MD36300} = \frac{4 * \text{MD31020}}{\text{MD10050} * \text{MD10070}}$$

MD31020 \$MA_ENC_RESOL（每转的编码器线数）

MD10050 \$MN_SYSCLOCK_CYCLE_TIME（系统基本周期）

MD10070 \$MN_IPO_SYSCLOCK_TIME_RATIO（插补周期系数）

说明

主轴相关的位置闭环控制激活转速必须根据主轴的绝对值编码器的编码器极限频率设置：

MD35300 \$MA_SPIND_POSCTRL_VELO（位置闭环控制激活转速）

MD36300 \$MA_ENC_FREQ_LIMIT（编码器极限频率）

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.enc1Activation	LBP_Axis*.A_PosMeas1	DB31, ...DBX1.5
<Axis>.basic.out.enc2Activation	LBP_Axis*.A_PosMeas2	DB31, ...DBX1.6
<Axis>.basic.out.enableCtrl	LBP_Axis*.A_ContrEnable	DB31, ...DBX2.1

4.10 自动恢复机床参考

4.9.9 不支持的回参考点方式

下列回参考点方式不适用于绝对值编码器：

- 通过编码器零脉冲回参考点/校准
- 距离编码的参考脉冲
- 包含两个脉冲沿分析的无接触式接近开关

4.10 自动恢复机床参考

4.10.1 概述

若未定义机床参考，那么在运行机床轴时将无法执行位置相关功能，例如转换或刀具框架。在很多情形下，编码器激活时就必须启用这些功能用于轴的运行，例如控制系统激活后或退出机床轴驻留（参见“监控与补偿”功能手册）后。但同时又不适合（或者无法）运行机床轴来重新回参考点。

绝对值编码器

对于使用校准过的绝对值编码器的测量系统，无需采取额外措施便可通过读取编码器值立即恢复机床参考。

增量编码器

对于增量测量系统，在不运行轴的情况下可通过“自动回参考点”或“恢复实际位置”恢复机床参考。

前提条件

 **警告**

因机床轴实际位置偏差造成的位置测量系统同步故障

机床轴测量系统关闭期间，不允许对其进行机械移动。否则会使缓存中保存的最后实际位置和机床轴的实际位置之间产生偏移。这可能会造成测量系统同步错误，并对人员和机械造成危险。

机床制造商须在机床上提供相应措施保证实际位置不变，用户须遵循这些措施。相关责任由机床制造商/用户自行承担。

若机床的构造决定了关闭状态下轴的移动无法避免，则必须使用绝对值编码器，或在上电后重新执行回参考点。

说明**编码器模块 SMCxx**

自动回参考点以及重启后将实际位置恢复为缓存中的最后位置这两个功能都必须有编码器模块 SMCxx（Externaly Mounted，外部安装）作为支持。使用编码器模块 SMCxx（Cabinet，机柜安装）或 SMIxx（集成）时，在控制系统重启（上电）后**无法**恢复实际位置。机床轴的测量系统必须重新执行回参考点。

4.10.2 自动回参考点

功能

启用自动回参考点功能时，控制系统接通后，生效的机床轴测量系统的实际位置会被设置为缓存中最后保存的位置，且编码器状态会被设置为“已回参考点”。这样一来，在 AUTOMATIC 和 MDI 运行方式下便可在控制系统启动后直接启动程序。

前提条件

- 控制系统接通后生效的测量系统必须已在系统关闭前执行了首次回参考点。
- 控制系统关闭时，机床轴必须处于“精准停”(<Axis>.basic.in.posFineReached == 1) 状态。

说明

若系统关闭时机床轴**不**处于“精准停”状态，那么上电后实际位置将以“0”值初始化。编码器状态则会显示“未回参考点”。

参数设置

自动回参考点通过将编码器状态设置为“自动回参考点已使能，但编码器尚未回参考点”来使能：

MD34210 \$MA_ENC_REFP_STATE[<编码器>] = 1

测量系统回参考点后，编码器状态中会显示“在下一次激活编码器时执行自动回参考点”：

MD34210 \$MA_ENC_REFP_STATE[<编码器>] == 2

NC/PLC 接口信号

自动回参考点后，生效测量系统的编码器状态会显示为“已回参考点”：

4.10 自动恢复机床参考

```
<Axis>.basic.in.enc1Synchronized/<Axis>.basic.in.enc2Synchronized
== 1（已回参考点/已同步 1 / 2）
```

边界条件

MD34210 \$MA_ENC_REFP_STATE[<编码器>] == 1 时的编码器激活

激活编码器时，编码器状态等于“1”表示“自动回参考点”已使能。但测量系统尚未回参考点，或在机床轴静止时未在“精准停”状态下关闭。此时对机床轴或生效的测量系统会进行以下设置：

- 实际位置 = 0
- 生效的系统，编码器状态 = “未回参考点”：
 <Axis>.basic.in.enc1Synchronized /
 <Axis>.basic.in.enc2Synchronized = 0（已回参考点/已同步 1 / 2）

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.enc1Synchronized	LBP_Axis*.E_RefSyn1	DB31, ...DBX60.4
<Axis>.basic.in.enc2Synchronized	LBP_Axis*.E_RefSyn2	DB31, ...DBX60.5
<Axis>.basic.in.posFineReached	LBP_Axis*.E_ExactFine	DB31, ...DBX60.7

更多信息

使用增量测量系统回参考点 (页 42)

4.10.3 恢复实际位置

功能

将实际位置恢复为缓存中最后保存的位置时，生效测量系统的编码器状态会被设置为“已恢复”。轴只可手动运行。

AUTO 运行方式

为了在 AUTOMATIC 运行方式下使能 NC 启动来实现程序的自动执行，必须对机床轴的测量系统重新执行回参考点。

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

前提条件

控制系统接通后生效的测量系统必须已在系统关闭前执行了首次回参考点。

参数设置**使能：恢复实际位置**

恢复实际位置通过将编码器状态设置为“恢复关闭前缓存中最后保存的位置，不自动回参考点”来使能：

MD34210 \$MA_ENC_REFP_STATE[<编码器>] = 3

使能：NC 启动，用于“MDI”和“溢出转存”运行方式

NC 启动通过以下机床数据使能，用于在“MDI”和“溢出转存”运行方式下执行零件程序或零件程序段，此时测量系统状态为“位置已恢复”：

MD34110 \$MA_REFP_CYCLE_NR ≠ -1（通道专用回参考点运行时的回参考点）

MD20700 \$MC_REFP_NC_START_LOCK = 2（不带参考点的 NC 启动禁用）

NC/PLC 接口信号

恢复的实际位置不被视作与回参考点后的实际位置有同等质量。因此机床轴测量系统的状态会显示“位置已恢复”，而不是“已回参考点/已同步”。

实际位置已恢复：

- <Axis>.basic.in.enc1Synchronized /
<Axis>.basic.in.enc2Synchronized = 0（已回参考点/已同步 1 / 2）
- <Axis>.basic.in.enc1Restored / <Axis>.basic.in.enc2Restored = 1
（位置已恢复，编码器 1 / 2）

4.10 自动恢复机床参考

测量系统已回参考点:

- <Axis>.basic.in.enc1Synchronized / <Axis>.basic.in.enc2Synchronized = 0 → 1 (已回参考点/已同步 1 / 2)
- <Axis>.basic.in.enc1Restored / <Axis>.basic.in.enc2Restored = 1 → 0 (位置已恢复, 编码器 1 / 2)

说明

在“位置已恢复”状态下, 运行范围限制监控(软件限位开关、工作区域限制等)已生效。

边界条件

主轴

超出编码器极限频率时, 主轴会被复位为“未回参考点/同步”状态:

- <Axis>.basic.in.enc1Synchronized / <Axis>.basic.in.enc2Synchronized = 1 → 0 (已回参考点/已同步 1 / 2)
- <Axis>.basic.in.enc1Restored / <Axis>.basic.in.enc2Restored = 1 → 0 (位置已恢复, 编码器 1 / 2)

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.enc1Synchronized	LBP_Axis*.E_RefSyn1	DB31, ...DBX60.4
<Axis>.basic.in.enc2Synchronized	LBP_Axis*.E_RefSyn2	DB31, ...DBX60.5
<Axis>.basic.in.enc1Restored	LBP_Axis*.E_POS_RESTORED1	DB31, ...DBX71.4
<Axis>.basic.in.enc2Restored	LBP_Axis*.E_POS_RESTORED2	DB31, ...DBX71.5

更多信息

使用增量测量系统回参考点 (页 42)

4.11 前提条件

4.11.1 较大运行范围

直线轴，运行范围 > 4096 编码器转数，配备旋转绝对值编码器 EQN 1325，设置的绝对值编码器区域 MD34220 \$MA_ENC_ABS_TURNS_MODULO = 4096:

最大运行范围等同于增量编码器。

使用绝对值编码器的连续旋转回转轴：

- 支持任意的整数传动比。
- 建议将使用绝对值编码器的连续旋转回转轴参数设置为模数轴（运行范围 0...360 度）：
MD30310 \$MA_ROT_IS_MODULO = 1
否则测量系统激活后该回转轴可能需要非常大的运行行程，用于运行至绝对零。

使用绝对值编码器的机床轴：

为了确保测量系统重新激活后控制系统能够正确测算位置实际值，测量系统关闭时机床轴只允许移动小于绝对值编码器区域的一半的位移。

MD34220 \$MA_ENC_ABS_TURNS_MODULO

对编码器位置唯一性的提示

说明

线性绝对值编码器

线性编码器（例如 Heidenhain LC181）的绝对值在可提供的标度长度内始终为唯一值。

旋转绝对值编码器

旋转绝对值编码器的绝对值只在最大编码器转数范围内具有唯一性。

例如，Heidenhain 公司的旋转绝对值编码器 EQN 1325 能够在 0 至 4096 的转数范围内提供唯一绝对值。

根据安装条件将得出：

- 负载上带编码器的回转轴：4096 负载转数
- 电机上带编码器的回转轴：4096 电机转数
- 电机上带编码器的直线轴：4096 电机转数

示例：

旋转绝对值编码器 EQN 1325 安装到了一根直线轴的电机上。有效主轴螺距为 10 mm 时，运行范围内的唯一绝对值为 -20.48 m 至 +20.48 m。

4.11 前提条件

加速度和急动度

5.1 简要说明

加速度

通过以下途径生效的加速度能够更好地适应机床和各种工况：通过各设定数据设置轴和通道的最大值；通过零件程序和同步动作设置加速模式、动态自适应功能和限制功能。

轴专用功能：

- 最大加速度可设置
- 加速模式可通过零件程序指令选择：
刚性加速模式 (BRISKA)
- 通过零件程序指令设置一个最大加速度 (ACC)
- 快速移动条件下的单独最大加速度 (G00)。
- 软加速模式下的单独最大加速度
- 在非相切的程序段过渡条件下上调加速度

通道专用功能：

- 加速模式可通过零件程序指令选择：
刚性加速模式 (BRISK)
- 恒速运行时间可设置，以避免极端加速跃变
- 为叠加运动保留的加速裕量可设置
- 最大加速度可设置
- 特定实时事件下的加速度可设置
- 为径向加速度保留的加速裕量可设置

急动度

与加速度类似，有效急动度也可以根据机床和相应的工况进行优化调整。

5.1 简要说明

轴专用功能:

- 加速模式可通过零件程序指令选择:
软加速模式 (SOFTA)
- 单轴插补中的最大急动度可设置
- 轨迹插补中的最大急动度可设置

通道专用功能:

- 加速模式可通过零件程序指令选择:
软加速模式 (SOFT)
- 最大急动度可设置
- 特定实时事件下的轨迹急动度可设置
- 快速移动条件下的单独最大急动度 (G00)
- 在非恒定曲率的程序段过渡处自动上调急动度

折线式加速度特性曲线

折线式加速度特性曲线表示: 在转矩紧跟转速变速的电机 (尤其是步进电机) 所在的机床轴上, 系统可自动下调加速度, 使电机在达到最佳负载率的同时, 不会出现过载。

折线式加速度特性曲线可由以下参数设置:

- 最大速度 $v_{\text{最大}}$
- 最大加速度 $a_{\text{最大}}$
- 转矩开始回落时的速度 $v_{\text{回落}}$
- 转矩开始回落时的加速度 $a_{\text{回落}}$
- 加速度回落的曲线 (恒定、抛物线、线性)

5.2 详细说明

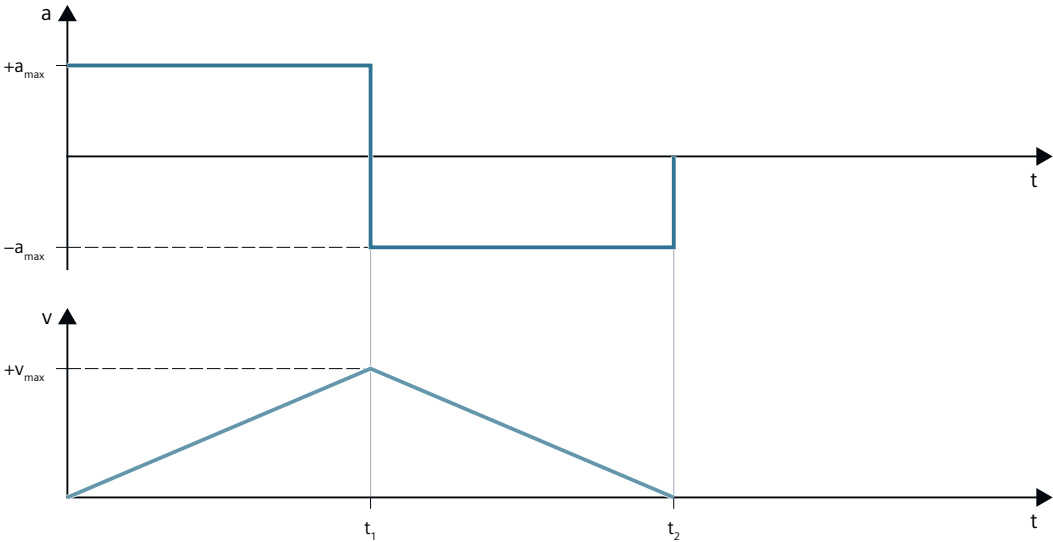
5.2.1 刚性加速模式 (BRISK/BRISKA) (通道/轴专用)

5.2.1.1 功能

在刚性加速模式中，轴会立即以最大加速度加速。和软加速模式相比，有以下不同：

- 优点
在同等的最大速度和加速度条件下，工时更短。
- 缺点
机械装置承受的负载更大，难以控制高频机械振动。

加速模式



$a_{\text{最大}}$: 最大加速度

$v_{\text{最大}}$: 最大速度

t: 时间

图 5-1 刚性加速模式下的速度和加速度原理图

5.2 详细说明

从上图中可发现加速模式的以下特性：

- 时间点： t_0
加速度从 0 跳到 $+a_{\text{最大}}$
- 间隔： $t_0 - t_1$
加速度保持在 $+a_{\text{最大}}$ ；此时速度线性增加
- 时间点： t_1
直接从加速切换至制动时回落 2 倍的 $a_{\text{最大}}$

说明

通常可通过指定恒速运行时间（参见章节“恒速运行时间（通道专用）（页 97）”）来避免加速度跃变。

- 间隔： $t_1 - t_2$
加速度保持在 $-a_{\text{最大}}$ ；此时速度线性降低

5.2.1.2 参数设置

轨迹运动中的最大轴加速度

轨迹运动中的每根机床轴的最大加速度可根据具体的工艺通过以下机床数据进行设置：

MD32300 \$MA_MAX_AX_ACCEL[<参数组下标>]

其中，参数组下标 = 0, 1, 2 ... （最大参数组编号 - 1）

有关工艺专用参数组的**更多信息**请参见功能手册之基本功能。

在设计轨迹时，系统会依据上述设置的最大值（该值不要超过参与轨迹运行的机床轴的最大值）来计算轨迹参数。

说明

在特定条件下允许超过最大值（参见章节“加速度自适应 (ACC)（轴专用）（页 98）”和“针对实时事件的轨迹加速度（通道专用）（页 101）”）。

定位轴运动中的最大轴加速度

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	生效的最大轴加速度：MD32300 \$MA_MAX_AX_ACCEL[0]
1	生效的最大轴加速度：MD32300 \$MA_MAX_AX_ACCEL[1]

JOG 最大轴加速度

参见“JOG 运动中的加速度和急动度 (页 131)”章节。

5.2.1.3 编程

轨迹刚性加速模式 (BRISK)

句法

BRISK

功能

通过零件程序指令 BRISK 可选择“刚性”加速模式。

G 功能组：21

有效性：模态有效

复位特性

进行复位操作后，以下通道专用的初始设置生效：

MD20150 \$MC_GCODE_RESET_VALUES[20]

5.2 详细说明

前提条件

如果中途在零件程序中切换了加速模式 (BRISK / SOFT)，系统会在程序段结束时执行一个准停。

单轴刚性加速模式 (BRISKA)

句法

BRISKA (~~轴~~, 轴)

功能

通过零件程序指令 BRISKA 可为单轴运动 (JOG、JOG/INC、定位轴、摆动轴等) 选择“刚性”加速模式。

G 功能组: -

有效性: 模态有效

轴:

- 取值范围: 通道轴名称

轴专用的初始设置

刚性加速模式可设为单轴运动中的初始设置:

MD32420 \$MA_JOG_AND_POS_JERK_ENABLE = FALSE

复位特性

进行复位操作后，以下轴专用的初始设置生效:

MD32420 \$MA_JOG_AND_POS_ENABLE

5.2.2 恒速运行时间（通道专用）

5.2.2.1 功能

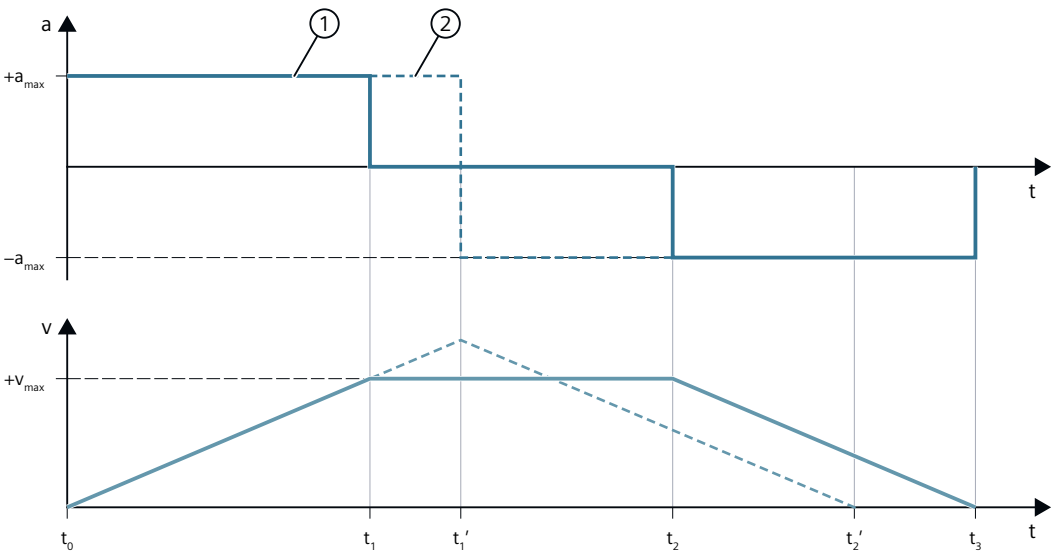
在刚性加速模式中，从加速切换至制动时，加速度会回落 2 倍的 $a_{\text{最大}}$ 。可设置一个通道专用的恒速运行时间以避免该加速度跃变。该时间就是指加速阶段和制动阶段之间的恒速运行时间：

MD20500 \$MC_CONST_VELO_MIN_TIME（恒速最小时间）

说明

恒速运行时间在以下情况中不起作用：

- 生效的功能：预读
- 运行程序段的运行时间小于或等于差补周期。



- 1: 带恒速运行时间的运行
- 2: 不带恒速运行时间的运行

$a_{\text{最大}}$ 最大加速度

大:

$v_{\text{最大}}$ 最大速度

大:

t: 时间

图 5-2 刚性加速模式下的原理图

5.2 详细说明

从上图中可发现恒速运行时间的作用：

- 时间点： t_1
加速阶段结束，加速度跳跃 1 倍的 $a_{\text{最大}}$
- 间隔： $t_1 - t_2$
加速度 0；通过设置恒速运行时间达到恒定速度
- 时间点： t_2
制动阶段开始，加速度跳跃 1 倍的 $a_{\text{最大}}$

时间点 t_0 、 t_1' 和 t_2' 表示的是没有恒速运行时间时的运行。

5.2.2.2 参数设置

恒速运行时间是通过以下机床数据设置的：

MD20500 \$MC_CONST_VELO_MIN_TIME
(恒速最小时间)

5.2.3 加速度自适应 (ACC) (轴专用)

5.2.3.1 功能

通过指令 ACC 可针对具体的轴将在加速度专用机床数据中设置的当前生效的轴最大加速度降低。降低通过设置百分比系数实现，该系数在进行指令编程时一同给定。

例如，可在加工程序段中对最大可能的轴加速度进行降低，以避免因巨大的加工力度而造成的机械振动。

有效性

使用 ACC 的加速度降低在 AUTO、MDA 以及空运行进给模式下的所有插补方式都**起效**。

在下面情况中加速度降低**无效**：

- JOG 运行方式下
- 执行 REF (回参考点) 机床功能期间。
- 由于检测到故障，轴通过快速停止 (设定值 = 0) 而静止时。

5.2.3.2 编程

句法

```
ACC [<轴>] =<降低系数>

ACC [SPI (<主轴编号>)] =<降低系数>

ACC (S<主轴编号>) =<降低系数>
```

含义

<ACC>:	针对具体轴的从机床数据中导出的可降低当前最大可能加速度的指令	
	有效性:	模态
<轴>:	轨迹轴的通道轴名称	
	数据类型:	AXIS
	值范围:	通道轴名称
SPI (<主轴编号>):	函数 SPI (...) 可将主轴编号转换为相应的通道轴名称。	
S<主轴编号>:	通道中的主轴名称	

其他信息

系统变量

通道中当前生效的，使用 ACC 设置的，加速度降低可针对具体轴通过以下指令读取：

```
$AA_ACC [<轴>]
```

复位特性

使用 ACC 设置的加速度降低不会因通道复位或程序结束而变化。这通过以下机床数据设置：

```
MD32320 $MA_DYN_LIMIT_RESET_MASK
```

5.2.4 加速裕量（通道专用）

5.2.4.1 功能

通常，轨迹加速度的预处理会 100% 地利用设置的机床轴的最大值。但也可以预留一定的轨迹加速度以便在“快速离开工件轮廓”功能范畴内为叠加运动保留一定的加速裕量。例如：当系数设为 0.2 时，预处理最多只能利用 80% 的轨迹加速度。剩余 20% 的加速裕量会保留给叠加运动。

5.2 详细说明

5.2.4.2 参数设置

加速裕量的设置是通过以下机床数据进行的：
MD20610 \$MC_ADD_MOVE_ACCEL_RESERVE
(为叠加运动保留的加速裕量)

5.2.5 最大轨迹加速度（通道专用）

5.2.5.1 功能

为了灵活应对各种工况，可通过设定数据来限制由预处理计算出的加速度：
SD42500 \$SC_SD_MAX_PATH_ACCEL（最大轨迹加速度）
只有当在设定数据中设置的值小于由预处理计算出的轨迹加速度时，系统才会考虑该值。
该限制必须由以下设定数据来激活：
SD42502 \$SC_IS_SD_MAX_PATH_ACCEL = TRUE

5.2.5.2 参数设置

设置是通过以下设定数据进行的：
SD42500 \$SC_SD_MAX_PATH_ACCEL（最大轨迹加速度）
SD42502 \$SC_IS_SD_MAX_PATH_ACCEL（激活最大轨迹加速度）

5.2.5.3 编程

最大值

句法

`$SC_SD_MAX_PATH_ACCEL = 最大值`

功能

最大轨迹加速度可通过编程设定数据来调整。

最大值:

- 值范围: ≥ 0
- 单位: m/s^2

可用于:

- 零件程序
- 静态同步动作

激活/关闭

句法

`$SC_IS_SD_MAX_PATH_ACCEL = 值`

功能

最大轨迹加速度可通过编程设定数据来启用/关闭。

参数: 值

- 值范围: TRUE、FALSE

可用于:

- 零件程序
- 静态同步动作

5.2.6 针对实时事件的轨迹加速度（通道专用）

5.2.6.1 功能

由于无需兼顾加速度在加工质量方面和时间方面的最优性，在以下实时事件中：

- NC 启停
- 进给倍率变化
- “Safety Integrated” 功能范畴内的速度设定值发生变化

可通过通道专用的系统变量来设定针对上述事件的轨迹加速度：

`$AC_PATHACC = 轨迹加速度`

5.2 详细说明

针对实时事件的轨迹加速度只会在因上述事件导致速度变化的这段时间内生效。

限制

一旦指定的轨迹加速度超出了参与到轨迹运行的机床轴的加速能力，控制系统就会对轨迹加速度进行限制，使得生成的轴加速度 ($a_{\text{回落}}$) 最高只能是设置的最大值 ($a_{\text{最大}}$) 的 2 倍。

$$a_{\text{回落}} = 2 \text{ 倍的 } a_{\text{最大}}, a_{\text{最大}} = \text{MD32300 \$MA\_MAX\_AX\_ACCEL}$$

说明

针对实时事件的轨迹加速度无需考虑径向加速度。

有效性

起作用	针对实时事件的轨迹加速度只有涉及到以下实时事件时才能在 AUTO 和 MDA 模式下起作用： - NC 启停 - 倍率变化 “Safety Integrated” 功能范畴内的速度设定值发生变化
不起作用	针对实时事件的轨迹加速度对由于通道中的预处理功能重新设计轨迹（如轮廓弯曲、拐角、运动转化限制等）导致的轨迹速度变化不起作用。 当编程的值小于由预处理为相关的轨迹段计算出的轨迹加速度时，针对实时事件的轨迹加速度不起作用。

编程

有关在零件程序或同步动作中设置系统变量的详细信息请参见章节“编程 (页 102)”。

5.2.6.2 编程

句法

$$\text{\$AC\_PATHACC} = \text{轨迹加速度}$$

功能

针对实时事件的轨迹加速度是通过通道专用的系统变量来设定的。

参数： 轨迹加速度

- 值范围：轨迹加速度 ≥ 0
- 单位：m/s²

关闭 \$AC_PATHACC = 0

可用于：

- 零件程序
- 静态同步动作

复位特性

复位时系统会关闭针对实时事件的轨迹加速度。

前提条件

在零件程序中设置 \$AC_PATHACC 时，系统会通过 Reorg 指令自动触发一个预处理停止 (STOPRE)。

5.2.7 快速移动条件下的单独加速度 (G00) (轴专用)

5.2.7.1 功能

通常来说，因加工条件的不同，参与加工的机床轴的加速度应设得比实际的加速能力低。

为了使快速移动条件下的加速时间最优化（零件程序指令 G00），可为机床轴设置一个单独最大加速度。

JOG 手动运行模式

在 JOG 手动运行模式中进行快速叠加时的加速度不受功能的影响。

5.2.7.2 参数设置

快速移动条件下的单独加速度 (G00) 是由以下轴专用机床数据设置的：

MD32434 \$MA_G00_ACCEL_FACTOR
(G00 G00 最大加速度比例)

5.2 详细说明

预处理中重新设计轨迹时所考虑的快速移动 (G00) 条件下的最大单独加速度是按如下方式计算的:

$$\text{加速度[轴]} = \text{MD32300 \$MA\_MAX\_AX\_ACCEL} * \text{MD32434 \$MA\_G00\_ACCEL\_FACTOR}$$

5.2.8 软加速模式 (SOFT/SOFTA) (轴专用)

5.2.8.1 功能

在最大加速度相同的条件下, 相比刚性加速模式, 软加速模式中的加工时间更长。可为软加速模式 (SOFT/SOFTA) 设置一个单独的轴加速度以缩短该时间。

该软加速模式中的最大加速度是通过相对于轴最大加速度的系数来设置的。因此, 预处理中重新设计轨迹时所考虑的最大加速度是按如下方式计算的:

$$\text{加速度[<轴>]} = \text{MD32300 \$MA\_MAX\_AX\_ACCEL} * \text{MD32433 \$MA\_SOFT\_ACCEL\_FACTOR}$$

5.2.8.2 参数设置

软加速模式 (SOFT/SOFTA) 中的最大加速度是由以下轴专用机床数据设置的:

$$\text{MD32434 \$MA\_SOFT\_ACCEL\_FACTOR}$$

(SOFT 指令时的最大加速度比例)

5.2.9 在非相切的程序段过渡条件下上调加速度 (轴专用)

5.2.9.1 功能

在非相切的程序段过渡 (拐角) 条件下, 控制系统有时必须对几何轴进行突然制动, 以保持设置的轴动态响应。为避免这种制动, 可以上调加速度。

加速度上调是通过相对于轴最大加速度的系数来设置的。因此, 预处理中重新设计轨迹时所考虑的非相切的程序段过渡条件下的最大加速度是按如下方式计算的:

$$\text{加速度[轴]} = \text{MD32300 \$MA\_MAX\_AX\_ACCEL} * \text{MD32310 \$MA\_MAX\_ACCEL\_OVL\_FACTOR}$$

5.2.9.2 参数设置

在非相切的程序段过渡条件中上调加速度是由以下轴专用机床数据设置的：

MD32310 \$MA_MAX_ACCEL_OVL_FACTOR

（速度跃变的过载系数）

5.2.10 为径向加速度保留的加速裕量（通道专用）

5.2.10.1 功能

弯曲的轮廓上除了产生轨迹加速度（切向加速度）以外，也会产生径向加速度。如果在设置轨迹参数时未考虑到这一点，那么在加速或制动期间，弯曲轮廓上的轴加速度短时间内会达到最大加速度的 2 倍。

生效的轴加速度 =

轨迹加速度分量 + 径向加速度分量 =

2 倍的 (MD32300 \$MA_MAX_AX_ACCEL)

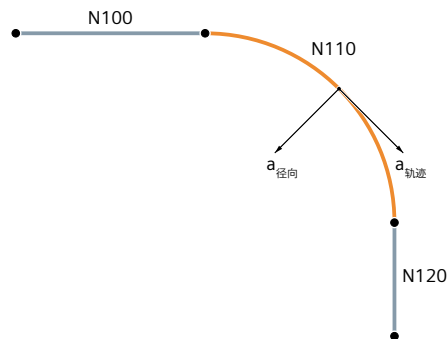


图 5-3 弯曲轮廓上的径向加速度和轨迹加速度

通过以下通道专用机床数据：

MD20602 \$MC_CURV_EFFECT_ON_PATH_ACCEL

（轨迹曲率对轨迹动态响应的影响）

可以设置轴加速度的分量，该分量应作为保留给径向加速度的加速裕量。

以值为 0.75 为例，75% 的轴专用加速度分配给了径向加速度，剩余 25% 分配给了轨迹加速度。

5.2 详细说明

通常，最大加速度是按以下方式计算的：

径向加速度 =

MD20602 \$MC_CURV_EFFECT_ON_PATH_ACCEL * MD32300 \$MA_MAX_AX_ACCEL

轨迹加速度 =

(1 - MD20602 \$MC_CURV_EFFECT_ON_PATH_ACCEL) * MD32300 \$MA_MAX_AX_ACCEL

示例

设定了以下下机床参数：

- MD32300 \$MA_MAX_AX_ACCEL 用于所有几何轴：3 m/s
- 基于机床的机械条件，轨迹半径为 10 mm 时的最大轨迹速度为：5 m/min

径向加速度是按以下方式计算的：

$$a_{\text{径向}} = \frac{v_{\text{轨迹}}^2 [\text{m/min}]}{r [\text{mm}] * 3.6 [\text{m/s}^2]} = \frac{5^2}{10 * 3.6} = 0.694 \text{ m/s}^2$$

加速裕量是按如下方式设置的：

$$\text{MD20602 \$MC\_CURV\_EFFECT\_ON\_PATH\_ACCEL} = \frac{a_{\text{径向}} [\text{m/s}^2]}{\text{MD32300 \$MA\_MAX\_AX\_ACCEL} [\text{m/s}^2]} = \frac{0.694}{3} \approx 0.23$$

线性程序段

在不带运动转换的线性程序段（直线插补）上，线性加速裕量不起作用。

5.2.10.2 参数设置

最大轴加速度的分量（该分量应作为弯曲轮廓上为径向加速度保留的加速裕量）是通过以下通道专用机床数据来设置的：

MD20602 \$MC_CURV_EFFECT_ON_PATH_ACCEL

（轨迹曲率对轨迹动态响应的影响）

5.2.11 轨迹插补中的最大急动度 (SOFT) (通道专用)

5.2.11.1 功能

该功能手册中描述的其他功能始终是针对加速度恒定和软加速模式（急动度 = 无穷值）。在软加速模式中，加速度会在 0 到最大值之间进行线性插补。

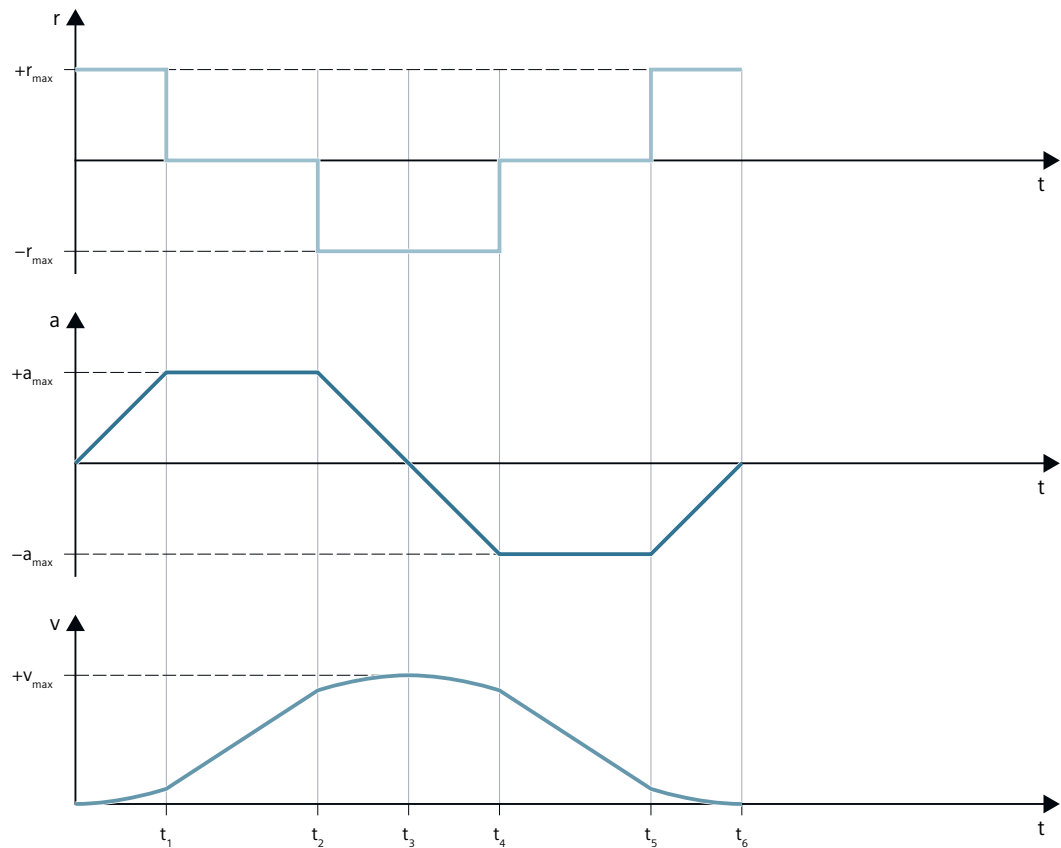
优点

机械装置承受的负载更小，不存在难以控制的高频机械振动。

缺点

在同等的最大速度和加速度条件下，工时更长。

加速模式



$r_{\text{最大}}$: 最大急动度

$a_{\text{最大}}$: 最大加速度

$v_{\text{最大}}$: 最大速度

t : 时间

图 5-4 软加速模式下的急动度、加速度和速度原理图

从上图可发现加速模式的以下特性:

- 间隔: $t_0 - t_1$
急动度保持在 $+r_{\text{最大}}$; 此时加速度线性增加; 此时速度平方增加
- 间隔: $t_1 - t_2$
加速度保持在 $+a_{\text{最大}}$; 此时速度线性增加
- 间隔: $t_2 - t_3$
急动度保持在 $-r_{\text{最大}}$; 此时加速度线性降低; 此时速度提升至最大值平方降低 $+v_{\text{最大}}$
- 间隔: $t_3 - t_4$
急动度保持在 $-r_{\text{最大}}$; 此时制动加速度线性增加; 此时速度平方降低

- 间隔: $t_4 - t_5$
制动加速度保持在 $-a_{\text{最大}}$: 此时速度线性降低
- 间隔: $t_5 - t_6$
急动度保持在 $+r_{\text{最大}}$: 此时制动加速度线性降低; 此时速度平方降低至静止状态 $v = 0$

5.2.11.2 参数设置

轨迹运动的最大急动度（轴专用）

轨迹运动中的每根机床轴的最大急动度可根据具体的工艺通过以下机床数据进行设置:

MD32431 \$MA_MAX_AX_JERK[<参数组下标>]

其中, 参数组下标 = 0, 1, 2 ... (最大参数组编号 - 1)

有关工艺专用参数组的**更多信息**请参见功能手册之基本功能。

在设计轨迹时, 系统会依据上述设置的最大值 (该值不要超过参与轨迹运行的机床轴的最大值) 来计算轨迹参数。

说明

在特定条件下允许超过最大值 (参见章节 “针对实时事件的轨迹急动度 (通道专用) (页 113)”)。

轨迹运动的最大急动度（通道专用）

除了能够设置轴专用的最大急动度, 也可以通过以下机床数据将最大急动度预设为通道专用的轨迹参数:

MD20600 \$MC_MAX_PATH_JERK (与轨迹相关的最大急动度)

为了避免轴专用和通道专用的最大急动度之间相互影响, 通道专用的最大急动度必须大于轴专用的最大急动度。

5.2.11.3 编程

句法

SOFT

5.2 详细说明

功能

通过零件程序指令 `SOFT` 可为通道中的几何轴运行选择软加速模式。

G 功能组：21

有效性：模态有效

复位特性

进行复位操作后，以下通道专用的初始设置生效：

`MD20150 $MC_GCODE_RESET_VALUES[20]`

前提条件

如果加工时在一个零件程序中变换加速模式（`BRISK` ↔ `SOFT`），则在连续路径运行时也会在程序段结束的过渡处使用准停来更换程序段。

5.2.12 单轴插补中的最大急动度 (SOFTA)（轴专用）

5.2.12.1 参数设置

最大轴急动度的初始设置

带急动度限制的加速（软加速模式）可设为轴的初始设置：

`MD32420 $MA_JOG_AND_POS_JERK_ENABLE== TRUE`

定位运行中的最大轴急动度

在软加速模式中运行定位轴时，以下机床数据中的其中一个值将作为最大轴急动度生效：

- `MD32430 $MA_JOG_AND_POS_MAX_JERK`（最大轴急动度）
- `MD32431 $MA_MAX_AX_JERK [0]`（动态模式 `DYNNORM` 下的轨迹运动中的最大轴急动度）
- `MD32431 $MA_MAX_AX_JERK [1]`（动态模式 `DYNNORM` 下的轨迹运动中的最大轴急动度）

所设的定位轴动态模式决定了使用哪个机床数据：

MD18960 \$MN_POS_DYN_MODE = <模式>

<模式>	含义
	作为定位轴运动中的最大轴急动度生效：
0	MD32430 \$MA_JOG_AND_POS_MAX_JERK G75 生效时（运行到固定点）： MD32431 \$MA_MAX_AX_JERK[0]
1	MD32431 \$MA_MAX_AX_JERK[1]

JOG 运动中的最大轴急动度

参见“JOG 运动中的加速度和急动度 (页 131)”章节。

5.2.12.2 编程

句法

SOFTA (轴{轴})

功能

通过零件程序指令 SOFTA 可为单轴运动（定位轴、摆动轴等）选择软加速模式。

G 功能组-

有效性： 模态有效

轴:

- 取值范围：通道轴名称

轴专用的初始设置

软加速模式可设为单轴运动中的初始设置：

MD32420 \$MA_JOG_AND_POS_JERK_ENABLE = TRUE

复位特性

进行复位操作后，以下轴专用的初始设置生效：

MD32420 \$MA_JOG_AND_POS_ENABLE

5.2 详细说明

5.2.13 最大轨迹急动度（通道专用）

5.2.13.1 功能

为了灵活应对各种工况，可通过设定数据来限制由预处理计算出的急动度：

SD42510 \$SC_SD_MAX_PATH_JERK（最大轨迹急动度）

只有当在设定数据中设置的值小于由预处理计算出的轨迹急动度时，系统才会考虑该值。

该限制必须由以下设定数据来激活：

SD42512 \$SC_IS_SD_MAX_PATH_JERK = TRUE

5.2.13.2 参数设置

设置是通过以下设定数据进行的：

SD42510 \$SC_SD_MAX_PATH_JERK（最大轨迹急动度）

SD42512 \$SC_IS_SD_MAX_PATH_JERK
（激活最大轨迹急动度）

5.2.13.3 编程

最大轨迹急动度

句法

\$SC_SD_MAX_PATH_JERK = *急动度值*

功能

最大轨迹急动度可通过编程设定数据来调整。

急动度值：

- 值范围：≥ 0
- 单位：m/s³

可用于：

- 零件程序
- 静态同步动作

激活/关闭

句法

`$SC_IS_SD_MAX_PATH_JERK = 值`

功能

最大轨迹急动度可通过编程设定数据来启用/关闭。

参数：值

- 值范围：TRUE、FALSE

可用于：

- 零件程序
- 静态同步动作

5.2.14 针对实时事件的轨迹急动度（通道专用）

5.2.14.1 功能

由于无需兼顾加速度在加工质量方面和时间方面的最优性，在以下实时事件中：

- NC 启停
- 进给倍率变化
- “Safety Integrated” 功能范畴内的速度设定值发生变化

可通过通道专用的系统变量来设定针对上述事件的轨迹急动度：

`$AC_PATHJERK = 轨迹急动度`

针对实时事件的轨迹急动度只会在因上述事件导致速度变化的这段时间内生效。

限制

由于急动度不是相关驱动的物理量，因此不限制预设的急动度。

5.2 详细说明

有效性

起作用	针对实时事件的轨迹急动度只有涉及到以下实时事件时才能在 AUTO 和 MDA 模式下起作用： - NC 启停 - 倍率变化 “Safety Integrated” 功能范畴内的速度设定值发生变化
不起作用	针对实时事件的轨迹急动度对由于通道中的预处理功能重新设计轨迹（如轮廓弯曲、拐角、运动转化限制等）导致的轨迹速度变化不起作用。 当编程的值小于由预处理为相关的轨迹段计算出的轨迹急动度时，针对实时事件的轨迹急动度不起作用。

编程

为了根据加速度来设置实时事件的急动度，可按以下步骤来设置系统变量：

$$\$AC\_PATHJERK = \$AC\_PATHACC / \text{平滑时间}$$

- \$AC_PATHACC：轨迹加速度 [m/s²]
平滑时间：可随意选择，如 0.02 s

有关在零件程序或同步动作中设置系统变量的详细信息请参见章节“编程 (页 114)”。

5.2.14.2 编程

句法

$$\$AC\_PATHJERK = \text{急动度值}$$

功能

针对实时事件的轨迹急动度是通过通道专用的系统变量来设定的。

急动度值：

- 取值范围：轨迹急动度 ≥ 0
- 单位： m/s³

可用于：

- 零件程序
- 静态同步动作

复位特性

复位时系统会关闭该功能。

前提条件

在零件程序中设置 \$AC_PATHJERK 时，系统会通过 Reorg 指令自动触发一个预处理停止 (STOPRE)。

5.2.15 快速移动条件下的急动度 (G00) (轴专用)

5.2.15.1 功能

通常来说，因加工条件的不同，参与加工的机床轴的最大急动度应设得比实际的能力低。

为了使快速移动条件下加速的时间最优化（零件程序指令 G00），可为机床轴设置一个最大轴急动度。

JOG 手动运行模式

在 JOG 运行模式中进行快速叠加时的急动度不受功能的影响。

5.2.15.2 参数设置

快速移动条件下的急动度 (G00) 是由以下轴专用机床数据设置的：

MD32434 \$MA_G00_ACCEL_FACTOR

（G00 指令时的最大加速度比例）

预处理中重新设计轨迹时所考虑的快速移动 (G00) 条件下的最大急动度是按如下方式计算的：

急动度[轴] =

MD32431 \$MA_MAX_AX_JERK * MD32435 \$MA_G00_JERK_FACTOR

5.2.16 在非恒定曲率的程序段过渡处自动上调急动度 (轴专用)

5.2.16.1 功能

在非相切的程序段过渡（拐角）条件下，控制系统有时必须对几何轴进行突然制动，以保持设置的轴动态响应。为避免这种制动，可以上调加速度。

上调急动度是通过一个单独的最大急动度来设置的。

5.2 详细说明

5.2.16.2 参数设置

在非恒定曲率的程序段过渡处自动上调急动度是通过以下轴专用机床数据设置的：

MD32432 \$MA_PATH_TRANS_JERK_LIM

（在非恒定曲率的程序段过渡处自动上调急动度）

5.2.17 随速度提高急动度（轴专用）

功能

轨迹动态响应是由参与到轨迹运动中恒定的轴的最大速度、最大加速度和最大急动度计算得出的：

- MD32000 \$MA_MAX_AX_VELO（最大轴速度）
- MD32300 \$MA_MAX_AX_ACCEL（最大轴加速度）
- MD32431 \$MA_MAX_AX_JERK（轨迹运动中的最大轴急动度）

在加工非恒定曲率的轮廓时，如加工任意形状的平面时，轴急动度是导致轨迹速度波动的主要因素，尤其是在高速区。该波动会导致工件的表面质量下降。

随速度相应地提高轴急动度（随速度调整急动度功能）可避免该波动。在有些条件下设置正确时可完全避免该波动。

该功能对允许的最大轨迹加速度和轨迹急动度没有任何的影响。这两个值即使是该功能激活时也可以从机床数据中设置的最大值算出。

由于直线运动中的曲率和挠度都为零，因此该功能对直线运动没有任何影响。

可用性

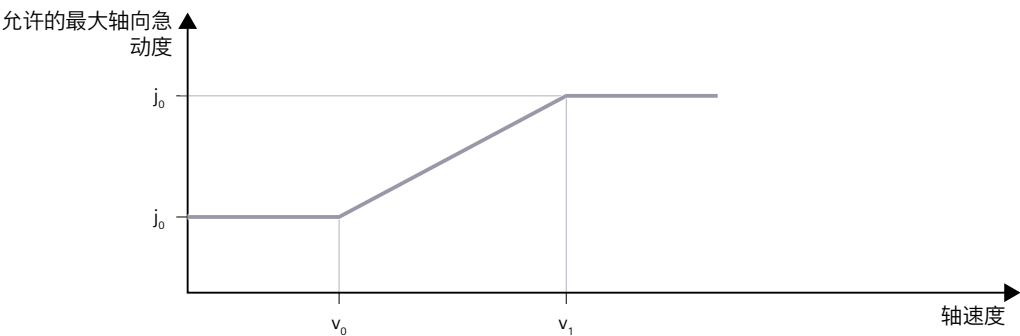
“随速度提高急动度”功能可独立于“自由形状表面模式：基本功能”功能存在。

关于“自由形状表面模式：基本功能”功能的更多信息参见功能手册之基本功能。

参数设置

“随速度提高急动度”功能是通过以下机床数据设置的：

- MD32437 \$MA_AX_JERK_VELO[<n>] = <阈值_{下限}>
急动度的速度阈值下限。从该速度起，“随速度提高急动度”功能生效。
可通过下标 n 为每个动态响应模式单独设置速度阈值下限。
- MD32438 \$MA_AX_JERK_VEL1[<n>] = <阈值_{上限}>
急动度的速度阈值上限。随速度提高的急动度在该速度时达到了它通过 MD32439 \$MA_MAX_AX_JERK_FACTOR 所设的最大值 $j_{最大}$ 。
可通过下标 n 为每个动态响应模式单独设置速度阈值上限。
- MD32439 \$MA_MAX_AX_JERK_FACTOR = <系数>
用于在达到速度阈值上限 MD32438 \$MA_AX_JERK_VEL1[<n>] 时设置随速度提高的最大急动度 $j_{最大}$ 的系数：
 $j_{最大} = (MD32431 \$MA\_MAX\_AX\_JERK) * (MD32439 \$MA\_MAX\_AX\_JERK\_FACTOR)$
值 > 1.0 时，随速度提高急动度有效。
值 = 1.0 时，随速度提高急动度无效。



v_0 MD32437 \$MA_AX_JERK_VELO
:
 v_1 MD32438 \$MA_AX_JERK_VEL1
:
 j_0 MD32431 \$MA_MAX_AX_JERK
:
 j_1 MD32439 \$MA_MAX_AX_JERK_FACTOR * MD32431 \$MA_MAX_AX_JERK
:

图 5-5 “轴速度-急动度”函数图

5.2 详细说明

说明

随速度提高急动度只在以下条件时有效：
MD32439 \$MA_MAX_AX_JERK_FACTOR > 1.0

示例

设置示例：

- MD32437 \$MA_AX_JERK_VELO = 3000 mm/min
- MD32438 \$MA_AX_JERK_VEL1 = 6000 mm/min
- MD32439 \$MA_MAX_AX_JERK_FACTOR[AX1] = 2.0
- MD32439 \$MA_MAX_AX_JERK_FACTOR[AX2] = 3.0
- MD32439 \$MA_MAX_AX_JERK_FACTOR[AX3] = 1.0

影响

- 随速度提高急动度对第 1 和第 2 根轴有效，而该函数对第 3 根轴无效。
- 当轴速度位于 0 到 3000 mm/min 范围内时，所设的急动度生效。
- 当轴速度位于 3000 mm/min 到 6000 mm/min 范围内时，最大急动度会直线提高。
- 当速度超过 6000 mm/min 时，第 1 根轴上允许的最大急动度是原先的 2 倍，第 2 根轴上允许的最大急动度是原先的 3 倍。
- 所设的值在每个动态模式中都有效。

5.2.18 急动度滤波器（位置设定值滤波器，轴专用）

5.2.18.1 功能

在某些应用情况中，如对任意形状的平面进行铣削时，对机床轴的位置设定值变化进行平滑可能会非常有用。平滑可减轻机床上的机械振动，从而达到更高的表面质量。

为进行该平滑，可激活位于位置控制级的急动度滤波器，该滤波器不受无需通道专用的或轴专用的最大急动度影响，该值在插补级中考虑。例如，可以通过以下机床数据设置急动限制：

- MD20600 \$MA_MAX_PATH_JERK（最大轨迹急动度）
- MD32431 \$MA_MAX_AX_JERK（轨迹运动时的最大轴急动度）

所有种类的位置设定值滤波器都有低通特性，并且倒圆出一个一直趋向曲线中点的轮廓（向内）。更大的时间常数或更低的滤波器频率会更强烈地倒圆出轮廓，不过也会使得机械共振更少摆动。总是需要在足够的阻尼效应和轮廓精确度之间找到一个平衡。

阻尼效应可以通过定位特性观察到，如通过一个圆弧形状测试来确定轮廓精确度。

为了使急动度滤波器能够更好地适应不同的机床条件，有以下几种滤波器类型可供选择：

- 二阶滤波器 (PT2)
- 滑动求平均值
- 带阻滤波器
- 双倍滑动求平均值

滤波器类型：二阶滤波器

二阶滤波器是一个简单的低通滤波器，只有在相对较小的滤波器时间常数（大约 10 ms）的条件下，才能满足上述要求。当时间常数较大时，系统会迅速产生过大的轮廓偏差。滤波效果相对较差。

基于历史原因，这种滤波器模式仍是缺省设置。

滤波器类型：滑动求平均值

在轮廓偏差较小时，通过滑动求平均值这种滤波器模式可将滤波器恒定时间设置在 20 - 40 ms 范围内。平滑效果也是非常对称的。

操作界面中显示的计算出的控制回路增益系数（ K_v 系数）低于根据滤波效果测出的值。轮廓精度高于显示的 K_v 系数所对应的水准。

从“二阶滤波器”模式切换至“滑动求平均值”模式时，在滤波器时间常量相同的条件下，尽管轮廓精度提高了，但所显示的 K_v 系数会有所降低。

滤波器类型：带阻滤波器

不推荐使用，只有出于兼容性原因时才（仍）可使用。

带阻滤波器是一个二阶滤波器，以分子和分母的形式表示：

5.2 详细说明

$$H(s) = \frac{\frac{s^2}{(2 * \pi * f_z)^2} + \frac{2 * s * D_z}{2 * \pi * f_z}}{\frac{s^2}{(2 * \pi * f_N)^2} + \frac{2 * s * D_N}{2 * \pi * f_N}}$$

f_z : 分子固有频率

f_N : 分母固有频率

D_z : 分子阻尼系数

D_N : 分母阻尼系数

为避免易振动的滤波器设置也能提供可用的结果，和低通“二阶滤波器”(PT2)模式一样，该模式也不提供 D_N 的设置选项。 D_N 固定为 1。

有 2 种设置带阻滤波器的方式：

- 真正的带阻滤波器
- 高频条件下幅值响应相应放大或衰减的带阻滤波器

真正的带阻滤波器

当分子和分母固有频率相同时，就可以获得真正的带阻滤波器：

$$f_z = f_N = f_{\text{截止}} \quad (\text{截止频率})$$

当分子阻尼系数为 0 时，就可以获得最大程度的衰减。3dB 阻带宽度是由以下方式得出的：

$$f_{\text{3dB 带宽}} = 2 * f_{\text{截止}}$$

如果不需要进行一个完整的衰减，只是希望按系数 k 衰减，则必须选择与分子阻尼系数相应的 k 。因此，上述 3dB 的阻带宽度就不再有效。

高频条件下幅值响应相应放大或衰减的带阻滤波器

在这种情况下，分子和分母固有频率的设置是不同的。此时，分子固有频率确定了截止频率。

选择一个比分子固有频率更高或更低的分母固有频率，就可以在高频条件下相应地衰减或放大幅值响应。在大多数情况下，高频条件下幅值响放大会有用，因为被控对象本身具有低通特性，也就是说：高频下幅值会衰减。

边界条件

如果分子固有频率过大，滤波器会自动关闭。此时，极限频率 $f_{z_{\text{最大}}}$ 取决于位置控制器周期。

$$f_{Z\text{最大}} = \frac{1}{2 * \pi * T_{Z\text{最小}}} = \frac{1}{2 * \pi * T_{\text{位置控制器周期}}}$$

(Shannon 温度计)

滤波器类型：双倍滑动求平均值

双倍滑动平均值相比于单倍滑动平均值，有更强的滤波效应，通过对第二个滑动平均值进行额外的参数调节，可平衡两个互为内插的轴之间的动态响应区别。

5.2.18.2 参数设置

激活

急动度滤波器是通过以下机床数据激活的：

MD32400 \$MA_AX_JERK_ENABLE（最大轴急动度）

急动度滤波器在每个运行方式和插补方式中都有效。

滤波器类型

滤波器类型是通过以下机床数据选择的：

MD32402 \$MA_AX_JERK_MODE（用于轴向最大急动度的滤波器类型）

值	滤波器类型
1	二阶滤波器
2	滑动求平均值
3	带阻滤波器
4	双倍滑动求平均值

轴急动度滤波器（滤波器类型 1、2 和 3）

轴急动度滤波器通过以下机床数据进行设置：

- 对于滤波器类型 1 和 2，使用 MD32410 \$MA_AX_JERK_TIME。
- 对于滤波器类型 3，使用机床数据 MD32412 AX_JERK_FREQ 和 MD 32414 AX_JERK_DAMP。

只有当时间常数大于位置控制周期时，急动度滤波器才会生效。

5.2 详细说明

双倍滑动求平均值（滤波器类型 4）

双倍滑动求平均值需要以下机床数据的设定：

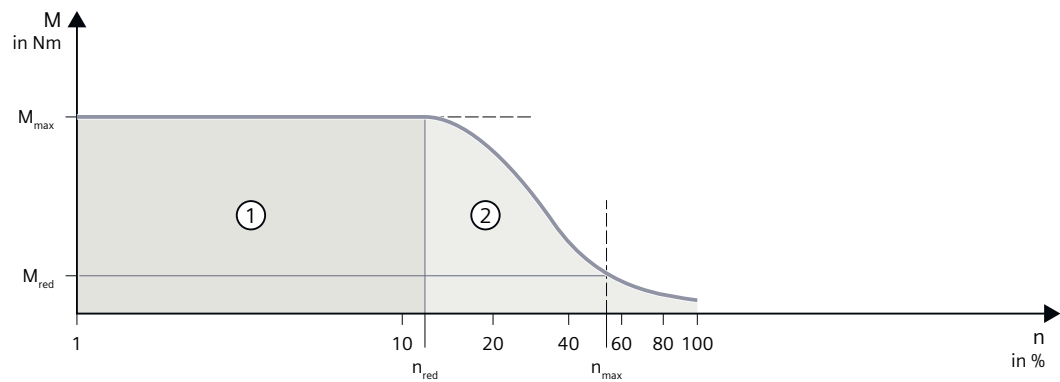
- MD32410 \$MA_AX_JERK_TIME
- MD32411 \$MA_AX_JERK_TIME_ADD

若两组机床数据被相同地参数化了，则会实现一个带有强劲阻尼和但也有很强的轮廓失真的三角滤波器。

5.2.19 折线式加速度特性曲线

5.2.19.1 功能：适应电机特性曲线

各种电机类型，尤其是步进电机，转矩会紧跟着转速变化，在高速区内转矩会急剧下降。为充分利用这种电机特性曲线，需要从某个转速点起降低相关 NC 轴的加速度。



① 普通区域

② 降低区域

$n_{\text{回落}}$ 从降低的转矩起开始运行的转速

$n_{\text{最大}}$ 最大转速

$M_{\text{最大}}$ 最大转矩

大

$M_{\text{回}}$ $n_{\text{最大}}$ 时的转矩（与转矩开始回落时的加速度一致）

落

特性曲线类型

可以为降低区域选择以下特性曲线类型（参见章节“参数设置 (页 126)”）作为控制器内部电机转矩特性曲线模拟：

- 恒定转矩曲线
- 抛物线转矩曲线
- 直线转矩曲线

5.2.19.2 功能：对轨迹加速度的影响

轨迹加速度的特性曲线是由参与到轨迹运动中的轴的特性曲线得出的。如果带有不同特性曲线的轴进行了共同插补，那么轨迹加速模式会由最具限制性的回落类型决定。

优先级的顺序如下，其中 1 表示最高优先级：

1. 加速度回落：0 = 恒定曲线
2. 加速度回落：1 = 抛物线
3. 加速度回落：2 = 直线
4. 无加速度回落生效
无生效的加速度回落可由以下方式得出，如：
MD35220 \$MA_ACCEL_REDUCTION_SPEED_POINT = 1
和/或
MD35230 \$MA_ACCEL_REDUCTION_FACTOR = 0

说明

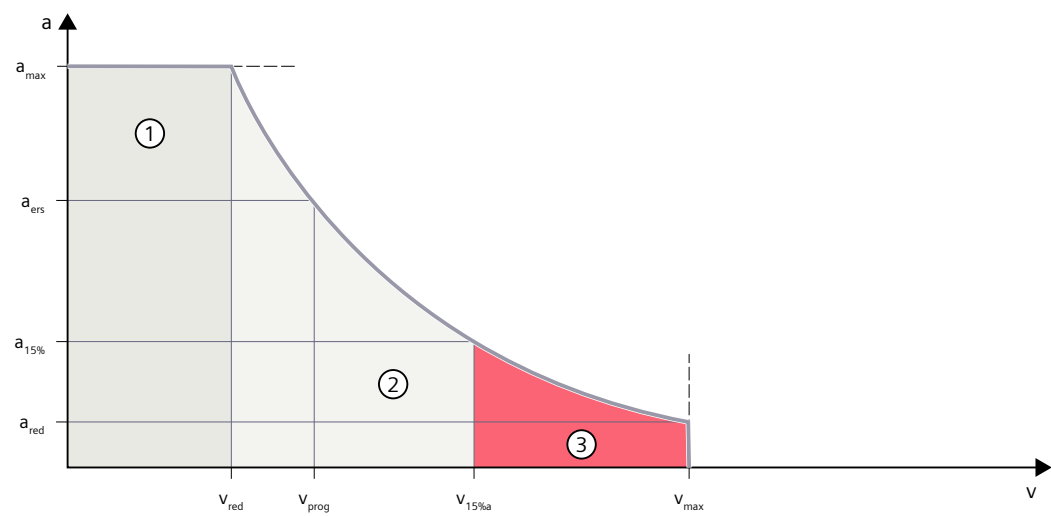
带有步进电机和直流电机的机床轴可共同插补。

5.2.19.3 功能：备用特性曲线

若通过相关轴的机床数据中所设置的加速度特性曲线无法运行编写的轮廓，例如在运动转换激活时以及在加速度方式 BRISK 或 SOFT 下，系统会降低动态限值来生成备用特性曲线。在确定降低的动态限值时，系统会使备用特性曲线在最大速度和恒定加速度之间达到最佳平衡。

直线轨迹段上的备用特性曲线

编程的轨迹速度会被限制在加速能力仍剩余 15% 的某个速度下 ($v_{15\%a}$)。这样，在任何一个加工条件下都至少有 15% 的最大电机加速能力或电机转矩可用。



① 普通区域

② 降低区域

③ 禁用区域

$a_{\text{备用}}$ 备用特性曲线的恒定加速度

$a_{15\%}$ 最小恒定加速度

$$a_{15\%} = 0.15 * (a_{\text{最大}} - a_{\text{回落}}) + a_{\text{回落}}$$

$v_{\text{备用}}$ 备用特性曲线速度

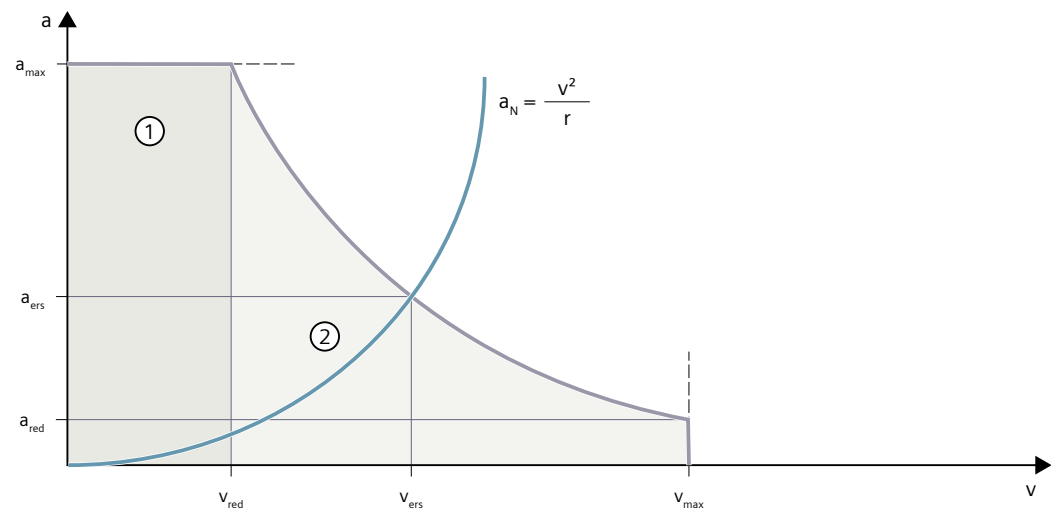
$v_{\text{编程}}$ 编程的速度

$v_{15\%a}$ $a_{15\%}$ 时的速度

图 5-6 轨迹备用特性曲线：直线轨迹

弯曲轨迹段上的备用特性曲线

在弯曲的轨迹段上，系统会同时考虑法线加速度和切向加速度。轨迹速度此时会降低至最多只为常规加速度留有 25% 的加速能力。剩余 75% 的加速能力会留给切向加速度，即预留给在轨迹上制动或加速。



- ① 普通区域
- ② 降低区域
- a_N 常规加速度
- $a_{\text{备用}}$ 备用特性曲线的恒定加速度
- $v_{\text{备用}}$ 备用特性曲线速度
- r 轨迹半径

图 5-7 轨迹备用特性曲线：弯曲轨迹

连续路径运行中的程序段过渡

如果连续路径运行是激活的，在以设置的轨迹速度运行时在非相切的程序段过渡处可能会出现轴速度跃变。

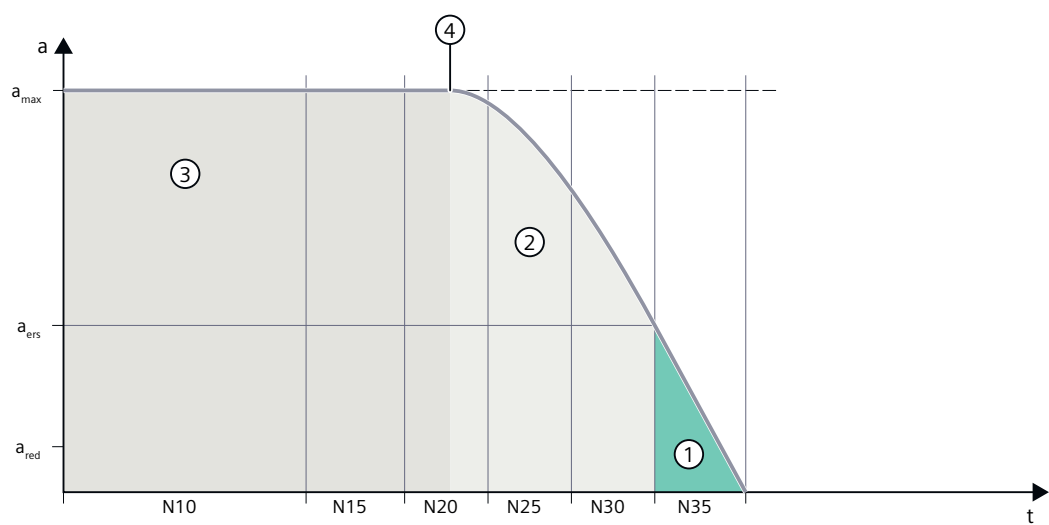
之后，系统会对轨迹速度加以控制，以避免在程序段过渡处出现大于转矩开始回落时的速度 $v_{\text{回落}}$ 的分量。

连续路径运行和预读功能下的制动斜坡

在多个连续短行程程序段中，加速过程或制动过程可能持续多个程序段。

此时“预读”功能也会考虑所设的随速度变化的加速度曲线。

5.2 详细说明



- ① 普通区域 $\Rightarrow a = a_{\max}$
 - ② 降低区域 $\Rightarrow a < a_{\max}$
 - ③ 恒速运行区域 $\Rightarrow a = 0 \text{ m/s}^2$
 - ④ 制动起点
- $v_{\text{回落}}$ 转矩开始回落时的速度
- $v_{\text{最大}}$ 最大速度
- Nx 程序段编号为 Nx 的运行程序段

图 5-8 带预读功能的制动过程

5.2.19.4 参数设置

轨迹轴

参数设置

通过机床数据设置轴专用的速度限值，超过该限值时降低的运行加速度作为轨迹轴生效：

MD35220 \$MA_ACCEL_REDUCTION_SPEED_POINT = <速度限值>

单轴

激活

通过机床数据激活轴专用的速度限值，超过该限值时降低的运行加速度作为单轴生效：

MD35240 \$MA_ACCEL_TYPE_DRIVE = TRUE

参数设置

需要以下机床数据来设置超出配置的速度限值时轴专用的加速度特性曲线：

- MD32000 \$MA_MAX_AX_VELO（最大轴速度）
- MD35220 \$MA_ACCEL_REDUCTION_SPEED_POINT = <速度限值>
从设置的转速起，降低的加速度生效
- MD35230 \$MA_ACCEL_REDUCTION_FACTOR = <降低系数>
从降低转速起，在最大加速度和最小加速度之间进行插补，直至达到最大转速。此时按以下方式计算最小加速度：
最小加速度 = 最大加速度 * (1 - 降低系数)
- MD35242 \$MA_ACCEL_REDUCTION_TYPE（参见以下章节：“选择转矩特性曲线”）
- MD32300 \$MA_MAX_AX_ACCEL（最大轴加速度）

选择转矩特性曲线

可以通过机床数据为降低区域选择以下特性曲线类型作为控制器内部电机转矩特性曲线模拟：

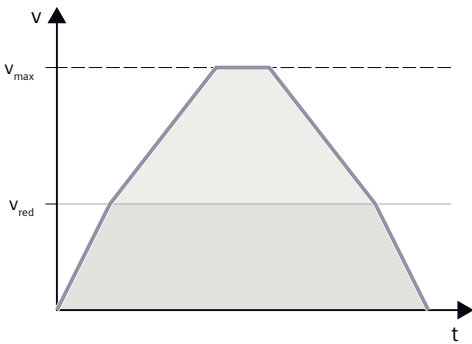
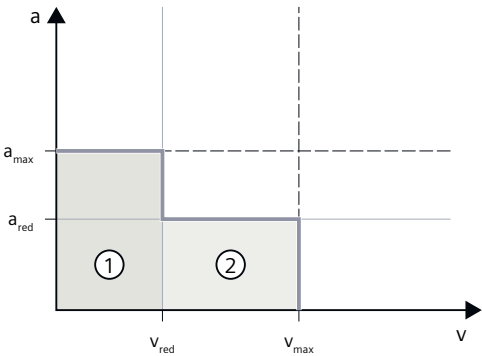
MD35242 \$MA_ACCEL_REDUCTION_TYPE = <值>

值	含义
0	恒定转矩曲线
1	抛物线转矩曲线
2	直线转矩曲线

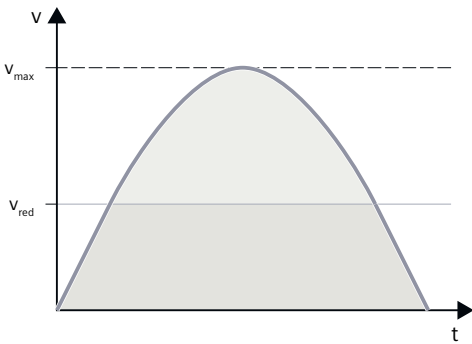
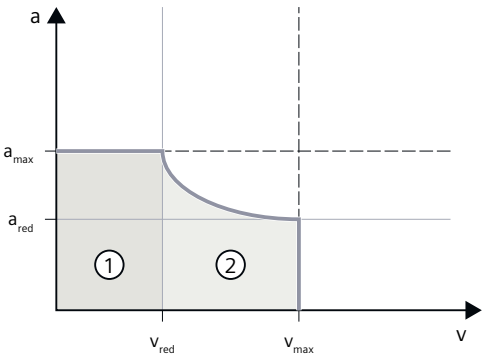
5.2 详细说明

下图展示了各个特性曲线类型的基本速度和加速度特性曲线：

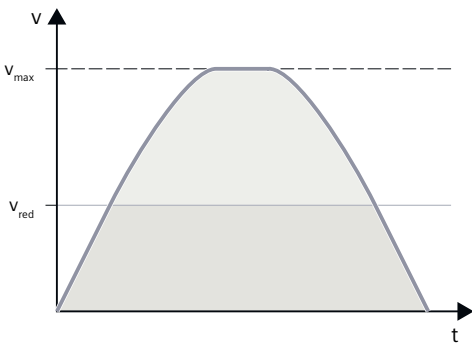
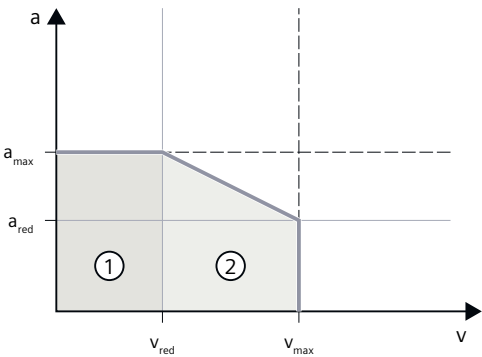
- 恒定转矩曲线（值 = 0）



- 抛物线转矩曲线（值 = 1）



- 直线转矩曲线（值 = 2）



特性曲线参数

特性曲线参数由以下机床数据得出：

① 普通区域

② 降低区域

$v_{\text{最大}}$ \$MA_MAX_AX_VELO

v_{回落}

$\$MA\_ACCEL\_REDUCTION\_SPEED\_POINT * \$MA\_MAX\_AX\_VELO$

a_{最大}

$\$MA\_MAX\_AX\_ACCEL$

a_{回落}

$(1 - \$MA\_ACCEL\_REDUCTION\_FACTOR) * \$MA\_MAX\_AX\_ACCEL$

5.2.19.5 编程：通道专用激活（DRIVE）

功能

通过指令 DRIVE 可以激活轴专用的速度限值，超过该限值时降低的运行加速度作为轨迹轴生效。

句法

DRIVE

含义

DRIVE:	超出所设速度限值时激活通道专用的加速度降低	
	G 功能组	21
	初始设置:	MD20150 \$MC_GCODE_RESET_VALUES[20]
	有效性:	模态

边界条件

特性曲线切换

如果通过编程 DRIVE 激活了加速度降低且通过 SOFT 或 BRISK 切换了加速度协议，则接着需要以降低的动态限值按照备用特性曲线运行。

重新编程 DRIVE 后可再次激活从所设速度限值起的加速度降低。

参见

参数设置 (页 126)

5.2 详细说明

5.2.19.6 编程：轴专用激活（DRIVEA）

功能

通过预定义的程序 DRIVEA () 可以激活已编程轴的轴专用的速度限值，超过该限值时降低的运行加速度作为单轴生效。

句法

```
DRIVEA (<轴_1>, <轴_2>, ...)
```

含义

DRIVEA:	激活已设置的轴专用折线式加速度特性曲线。	
	生效方式:	模态
<轴_x>:	已设置的折线式加速度特性曲线应激活的轴。	
	数据类型:	AXIS
	值域:	通道轴名称

边界条件

如果为轴设置了折线式加速度特性曲线，那么缺省条件下该轴会以该加速模式运行。

如果通过零件程序指令 SOFTA 或 BRISKA 切换了某个轨迹段的加速模式，那么系统会采用带有回落的动态限值的备用特性曲线来代替折线式加速度特性曲线。

通过编程 DRIVEA 可再次切换回折线式加速度特性曲线。

5.2.19.7 前提条件

单轴插补

在单轴插补（定位轴、摆动轴、手动运行等）中激活折线式加速度特性曲线后，就只能在 DRIVEA 模式下运行机床轴。

无法通过以下零件程序指令切换加速模式：

- 刚性加速模式 (BRISKA)
- 软加速模式 (SOFTA)

轨迹插补

如果在零件程序指令 DRIVE 未生效的情况下为参与到编程的轨迹的机床轴设置了折线式加速度特性曲线，系统将会为该轨迹确定一个带有回落的动态限值的备用特性曲线。

运动转换

运动转换生效时，系统不会考虑折线式加速度特性曲线。控制系统内部会切换至刚性加速模式 (BRISK)，同时备用特性曲线对轨迹加速度生效。

5.2.20 JOG 运动中的加速度和急动度

在 JOG 运行中，轴特有的加速度和急动度限制也生效。

此外，可以针对具体通道对用于几何轴及定向轴手动运行的加速度和急动度加以限制。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）。

5.2.20.1 参数设置

最大轴加速度和轴急动度

最大轴加速度

手动运行时的最大加速度在机床数据：

MD32300 \$MA_MAX_AX_ACCEL [0]（动态模式 DYNNORM 中轨迹运动中的最大轴加速度）

说明

对于 JOG 模式，始终只有动态模式 DYNNORM 有效。

另见章节“刚性加速模式 (BRISK/BRISKA)（通道/轴专用）（页 93）”。

最大轴急动度的初始设置

带急动度限制的加速（软加速模式）可设为 JOG 模式下轴的初始设置：

MD32420 \$MA_JOG_AND_POS_JERK_ENABLE== TRUE

最大轴急动度

在带急动度限制的 JOG 运动时，以下机床数据的值作为最大轴急动度生效：

MD32430 \$MA_JOG_AND_POS_MAX_JERK（最大轴急动度）

另见章节“单轴插补中的最大急动度 (SOFTA)（轴专用）（页 110）”。

通道专用的最大加速度和急动度

几何轴手动运行时的加速度上限

手动运行几何轴时，可通过以下机床数据来预设最大加速度：

MD21166 \$MC_JOG_ACCEL_GEO [<几何轴>]

其中 <几何轴> = 0, 1, 2

说明

MD21166 \$MC_JOG_ACCEL_GEO [<几何轴>] 对 MD32300 \$MA_MAX_AX_ACCEL 没有直接的限制。

说明

当转换激活时，MD32300 \$MA_MAX_AX_ACCEL 决定了最大轴加速度。

几何轴手动运行时的急动度上限

在加速模式 SOFT（软加速模式）中手动运行几何轴时，可通过以下机床数据设置最大加速度：

MD21168 \$MC_JOG_JERK_GEO [<几何轴>]

其中 <几何轴> = 0, 1, 2

说明

只有在为 JOG 模式中的机床轴激活了最大轴急动度时，MD21168 \$MC_JOG_JERK_GEO 才会生效：

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

当 MD21168 = 0 时，MD32430 \$MA_JOG_AND_POS_MAX_JERK 中的轴专用限值会代替通道专用的最大急动度生效。

定向轴手动运行时的急动度上限

手动运行定向轴时，可通过以下机床数据设置最大急动度：

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手动运行定向轴激活通道专用的最大急动度：

MD21159 \$MC_JOG_JERK_ORI_ENABLE == TRUE

5.2.20.2 前提条件

旋转指令生效时手动运行几何轴的特性

在加速模式 SOFT（软加速模式）中手动运行几何轴时，激活的旋转或激活的可定向刀架中也会使用 MD32430 \$MA_JOG_AND_POS_MAX_JERK 中的值。

零件程序指令 SOFTA / BRISKA / DRIVEA

零件程序指令 SOFTA (<轴 1>, <轴 2>, ...) 也可在 JOG 模式中生效，即在 JOG 模式下运行时，MD32430 \$MA_JOG_AND_POS_MAX_JERK 中的最大轴急动度对指定轴有效（和设置 MD32420 \$MA_JOG_AND_POS_JERK_ENABLE [<轴>] == TRUE 时一样）。

说明

零件程序指令 SOFT 对 JOG 模式无效。

和 SOFTA 一样，零件程序指令 BRISKA 和 DRIVEA 在 JOG 模式中也有效，也就是说，即使将 MD32420 \$MA_JOG_AND_POS_JERK_ENABLE 设为了“TRUE”，相关机床轴的刚性加速模式依然有效。

说明

手动运行定向轴不受 BRISKA/SOFTA/DRIVEA 的影响。

达到软件极限时，转换/插补中轴的特性

达到软件极限时，轴会非常迅速地制动。转换/插补链的所有其他轴都使用设置的制动斜坡进行制动。这导致轴的制动特性不同。

如果连续多次远离又接近软件极限，则由于轴的不同制动特性，会产生急动度。

示例：TRAANG（斜轴）的 Y 轴移动到软件极限，对于 X 轴，这会导致工件坐标系出现急动度。

参见

刚性加速模式 (BRISK/BRISKA)（通道/轴专用）(页 93)

单轴插补中的最大急动度 (SOFTA)（轴专用）(页 110)

5.3 示例

5.3 示例

5.3.1 加速度

5.3.1.1 轨迹速度曲线

在下面的示例中将借助编程设置的零件程序段的运动和动作来展示轨迹速度的变化曲线。

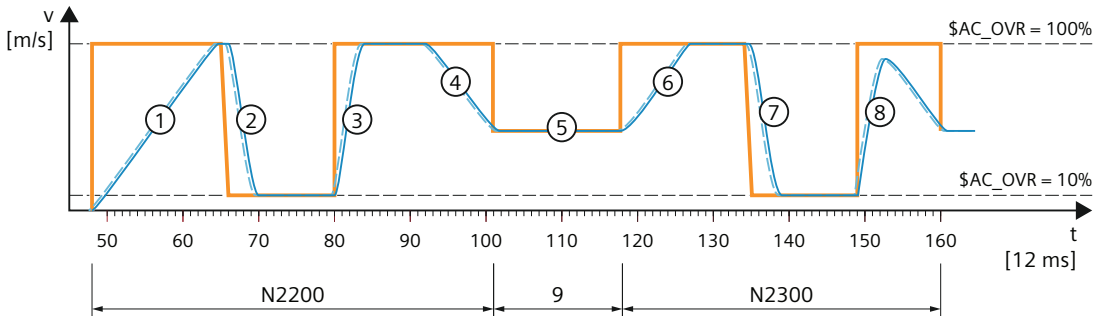
零件程序段

程序代码

```

; 同步动作： 取决于快速运行 1 ($A_IN[1]) 的加速度切换：
N53 ID=1 WHEN $A_IN[1] == 1 DO $AC_PATHACC = 2.*$MA_MAX_AX_ACCEL[X]
; 同步动作： 测试倍率曲线（模拟外部作用）：
N54 ID=2 WHENEVER ($AC_TIMEC > 16) DO $AC_OVR=10
N55 ID=3 WHENEVER ($AC_TIMEC > 30) DO $AC_OVR=100
; 逼近
N1000 G0 X0 Y0 BRISK
N1100 TRANS Y=-50
N1200 AROT Z=30 G642
; 轮廓
N2100 X0 Y0
N2200 X = 70 G1 F10000 RNDM=10 ACC[X]=30 ACC[Y]=30
N2300 Y = 70
N2400 X0
N2500 Y0
    
```

轨迹速度的变化曲线



加速曲线：BRISK

- 1: 按照加速度设定加速到轨迹速度 (F10000) 的 100%: ACC (N2200...)
- 2: 在倍率变化 (\$AC_OVR) 的基础上按照实时加速度 \$AC_PATHACC (N53/N54...) 制动至轨迹速度的 10%。
- 3: 在倍率变化 (\$AC_OVR) 的基础上按照实时加速度 \$AC_PATHACC (N53/N55...) 加速至轨迹速度的 100%。
- 4: 按照加速度设定: ACC (N2200...) 制动至倒圆中间程序段结束时的速度
- 5: 基于倒圆的最大速度 (见 9)
- 6: 按照加速度设定加速到轨迹速度 (\$AC_OVR) 的 100%: ACC (N2300...)
- 7: 在倍率变化的基础上按照实时加速度 \$AC_PATHACC (N53/N54...) 制动
- 8: 在倍率变化 (\$AC_OVR) 的基础上按照实时加速度 \$AC_PATHACC (N53/N55...) 加速至轨迹速度的 100%。
- 9: 在所编程基圆 (RNDM) (N2200...) 的基础上插入到控制系统内部的中间程序段

图 5-9 在在预处理中确定的轨迹加速度和实时加速度之间进行切换

5.3.2 急动度

5.3.2.1 轨迹速度曲线

在下面的示例中将借助编程设置的零件程序段的运动和动作来展示轨迹速度和急动度的变化曲线。

零件程序段

程序代码

；在有外部作用时设置轨迹加速度和轨迹急动度：

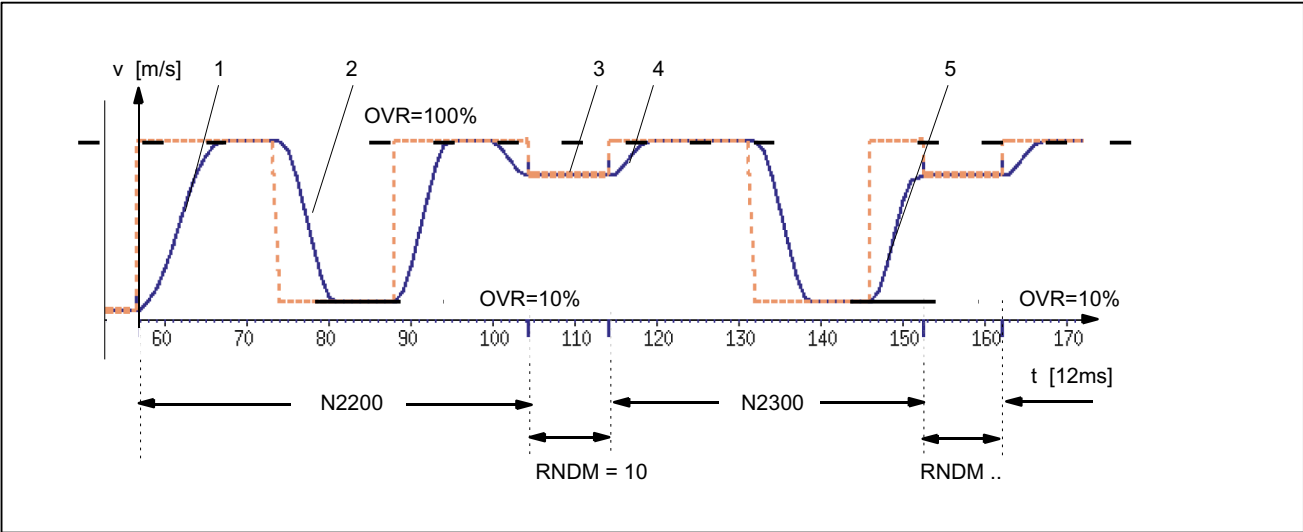
```
N0100 $AC_PATHACC = 0.0
N0200 $AC_PATHJERK = 4.0 * ($MA_MAX_AX_JERK[X] + $MA_MAX_AX_JERK[Y]) / 2.0
```

5.3 示例

程序代码

```
；同步动作： 倍率变化（模拟外部作用）
N0530 ID=1 WHENEVER ($AC_TIMEC > 16) DO $AC_OVR=10
N0540 ID=2 WHENEVER ($AC_TIMEC > 30) DO $AC_OVR=100
；逼近
N1000 G0 X0 Y0 SOFT
N1100 TRANS Y=-50
N1200 AROT Z=30 G642
；轮廓
N2100 X0 Y0
N2200 X = 70 G1 F10000 RNDM=10
N2300 Y = 70
N2400 X0
N2500 Y0
```

轨迹速度和急动度的变化曲线



加速曲线：SOFT

- 1: 急动度，符合 \$MA_MAX_AX_JERK[..]
- 2: 急动度，符合 \$AC_PATHJERK
- 3: 急动度，符合 \$MA_MAX_AX_JERK[..]（逼近程序段结束时的速度）
- 4: 因圆弧产生的速度限制
- 5: 急动度，符合 \$AC_PATHJERK

图 5-10 在在预处理中确定的轨迹急动度和 \$AC_PATHJERK 之间进行切换

5.3.3 加速度和急动度

以下示例中借助一个零件程序段展示了 X 轴的速度和加速度曲线以及哪些与速度和加速度相关的机床数据决定了哪些轮廓段。

零件程序段

程序代码	注释
N90 F5000 SOFT G64	；连续路径运行，软加速模式
N100 G0 X0 Y0 Z0	；快速移动
N110 G1 X10	；直线
N120 G3 CR=5 X15 Y5	；圆弧，半径 5 mm，程序段过渡：切向
N130 G3 CR=10 X5 Y15	；圆弧，半径 10 mm，程序段过渡：切向
N140 G1 X-5 Y17.679	；直线，15°弯曲

轮廓

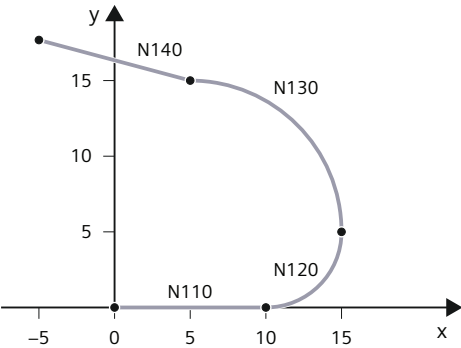


图 5-11 零件程序段的轮廓

速度和加速度变化曲线

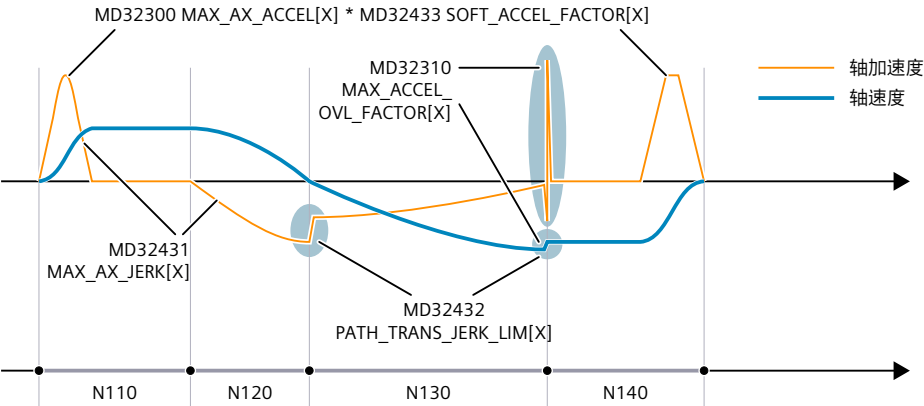


图 5-12 速度和加速度变化曲线：X 轴

5.3 示例

5.3.4 折线式加速度特性曲线

5.3.4.1 激活

该示例展示了如何借助机床数据和零件程序段激活折线式加速度特性曲线。

机床数据

机床数据	值
MD35220 \$MA_ACCEL_REDUCTION_SPEED_POINT[X]	= 0.4
MD35230 \$MA_ACCEL_REDUCTION_FACTOR[X]	= 0.85
MD35242 \$MA_ACCEL_REDUCTION_TYPE[X]	= 2
MD35240 \$MA_ACCEL_TYPE_DRIVE[X]	= TRUE
MD35220 \$MA_ACCEL_REDUCTION_SPEED_POINT[Y]	= 0.0
MD35230 \$MA_ACCEL_REDUCTION_FACTOR[Y]	= 0.6
MD35242 \$MA_ACCEL_REDUCTION_TYPE[Y]	= 1
MD35240 \$MA_ACCEL_TYPE_DRIVE[Y]	= TRUE
MD35220 \$MA_ACCEL_REDUCTION_SPEED_POINT[Z]	= 0.6
MD35230 \$MA_ACCEL_REDUCTION_FACTOR[Z]	= 0.4
MD35242 \$MA_ACCEL_REDUCTION_TYPE[Z]	= 0
MD35240 \$MA_ACCEL_TYPE_DRIVE[Z]	= FALSE

通过设定为通道专用的初始设置激活：

MC_GCODE_RESET_VALUE[20] = 3 (DRIVE)

零件程序段

程序代码	注释
N10 G1 X100 Y50 Z50 F700	； 带 DRIVE 的轨迹运动 (X,Y,Z)
N15 Z20	； 带 DRIVE 的轨迹运动 (Z)
N20 BRISK	； 切换至 BRISK
N25 G1 X120 Y70	； 带备用特性曲线的轨迹运动 (Y,Z)
N30 Z100	； 带 BRISK 的轨迹运动 (Z)
N35 POS[X] = 200 FA[X] = 500	； 带 DRIVEA 的定位运动 (X)
N40 BRISKA(Z)	； 为 Z 激活 BRISKA
N40 POS[Z] = 50 FA[Z] = 200	； 带 BRISKA 的定位运动 (Z)
N45 DRIVEA(Z)	； 为 Z 激活 DRIVEA
N50 POS[Z] = 100	； 带 DRIVE 的定位运动 (Z)

程序代码	注释
N55 BRISKA (X)	; 导致故障报警

5.3 示例

速度、设定值/实际值系统、闭环控制

6.1 简要说明

该功能手册描述的是如何设置机床轴参数，其中包括：

- 实际值系统或测量系统
- 设定值系统
- 操作精度
- 运行范围
- 进给轴速度
- 控制参数

6.2 速度、运行范围、精度

6.2.1 速度

最大轨迹速度、进给轴速度和主轴转速

最大轨迹速度、进给轴速度和主轴转速受机床结构、驱动动态响应配置和实际值采集装置的极限频率（编码器极限频率）影响。

最大进给轴速度是在以下机床数据：

MD32000 \$MA_MAX_AX_VELO（最大轴速度）

中定义的。

允许的最大主轴转速是通过以下机床数据：

MD35100 \$MA_SPIND_VELO_LIMIT（最大主轴转速）

设定的。

说明参见章节“主轴 (页 395)”。

进给率较大时（编程的进给率较大或受进给倍率的影响），速度会被限制为 $V_{\text{最大}}$ 。

由 CAD 系统生成的程序，其程序段很短，因而自动限制最大进给率可能导致速度降低。

6.2 速度、运行范围、精度

示例:

插补周期 = 12 毫秒

N10 G0 X0 Y0; [mm]

N20 G0 X100 Y100; [mm]

⇒ 程序段中编程的路径长度 = 141.42 毫米

⇒ $V_{\text{最大}} = (141.42 \text{ 毫米} / 12 \text{ 毫秒}) \cdot 0.9 = 10606.6 \text{ 毫米} / \text{秒} = 636.39 \text{ 米} / \text{分}$

最小轨迹速度、进给轴速度

最小轨迹速度或进给轴速度有以下限制:

$$V_{\min} \geq \frac{10^{-3}}{\text{计算精度} \cdot \left[\frac{\text{增量}}{\text{毫米或度}} \right] \cdot \text{插补周期} [\text{s}]}$$

计算精度是借助机床数据:

MD10200 \$MN_INT_INCR_PER_MM (线性位置计算精度)

或

MD10210 \$MN_INT_INCR_PER_DEG (角位置计算精度)

定义的:

值低于 $V_{\text{最小}}$ 时, 无法运行。

示例:

MD10200 \$MN_INT_INCR_PER_MM = 1000 [增量/毫米];

插补周期 = 12 毫秒;

⇒ $V_{\text{最小}} = 10^{-3} / (1000 \text{ 增量} / \text{毫米} \times 12 \text{ 毫秒}) = 0.005 \text{ 毫米} / \text{分};$

进给率的取值范围取决于所选的计算精度。

机床数据为缺省设置时:

MD10200 \$MN_INT_INCR_PER_MM

(线性位置计算精度) (1000 增量/毫米)

或者

MD10210 \$MN_INT_INCR_PER_DEG

(角位置计算精度) (1000 增量/度)

可通过指定的精度编程以下取值范围：

几何轴上的轨迹进给率 F 的取值范围：	
公制系统：	英制系统：
$0.001 \leq F \leq 9999999.999$ [毫米/分、毫米/转、度/分、度/转]	$0.001 \leq F \leq 3999999.999$ [英寸/分、英寸/转]

定位轴上的进给率的取值范围：	
公制系统：	英制系统：
$0.001 \leq FA \leq 9999999.999$ [毫米/分、毫米/转、度/分、度/转]	$0.001 \leq FA \leq 3999999.999$ [英寸/分、英寸/转]

主轴转速 S 的取值范围：	$0.001 \leq S \leq 9999999.999$ [转/分]

如果提高/降低了计算精度，取值范围也会发生相应的变化。

6.2.2 运行范围

运行范围的取值

运行范围的取值取决于所选的计算精度。

机床数据为缺省设置时：

MD10200 \$MN_INT_INCR_PER_MM

（线性位置计算精度）（1000 增量/毫米）

或者

MD10210 \$MN_INT_INCR_PER_DEG

（角位置计算精度）（1000 增量/度）

6.2 速度、运行范围、精度

可通过指定的精度编程以下取值范围：

表格 6-1 轴运行范围

	G71 [毫米、度]	G70 [英寸、度]
	区域	区域
线性轴 X、Y、Z、...	± 999999.999	± 399999.999
回转轴 A、B、C、...	± 999999.999	± 999999.999
插补参数 I、J、K	± 999999.999	± 399999.999

回转轴的尺寸单位始终为度。

如果计算精度提高/降低了 10 倍，取值范围也会发生相应的变化。

运行范围可以由软件限位开关和工作区加以限制（参见*监控与补偿功能手册*，*轴监控*章节）。

线性轴和回转轴的运行范围较大时的特殊性参见章节“回参考点 (页 35)”。

回转的运行范围可由机床数据加以限制。

更多信息：

回转轴 (页 315)

6.2.3 控制系统的定位精度

实际值精度和计算精度

控制系统的定位精度取决于实际值精度（即编码器增量/毫米或度）和计算精度（即内部增量/毫米或度）。

两个精度中较低的那个决定了控制系统的定位精度。

输入精度、插补周期和位置控制周期的选择对该精度没有任何影响。

除了可以用 MD32000 限制最大轨迹速度外，某些情况下控制系统也可以根据下列公式来限制最大轨迹速度：

内部增量/mm

编码器增量/mm

=

1

ENC_RESOL [n] * ENC_PULSE_MULT[n]

*

DRIVE_ENC_RATIO_NUMERA [n]

DRIVE_ENC_RATIO_DENOM [n]

*

DRIVE_AX_RATIO_DENOM [n]

DRIVE_AX_RATIO_NUMERA [n]

*

LEADSCREW_PITCH

*

INT_INCR_PER_MM

6.2.4 输入/显示精度，计算精度

精度：区别

线性位置和角位置、速度、加速度和急动度的精度可分为以下三种：

- 输入精度
通过操作面板或零件程序输入数据。
- 显示精度
通过操作面板显示数据。
- 计算精度
内部换算通过操作面板或零件程序输入的数据。

输入和显示精度是在操作面板上设定的，可通过以下机床数据：

MD9004 \$MM_DISPLAY_RESOLUTION（显示精度）

修改位置值的显示精度。

通过机床数据：

MD9011 \$MM_DISPLAY_RESOLUTION_INCH（英制单位制显示精度）

可将位置值的显示精度设为英制。

设为英制时，最多可以显示小数点后的六位。

零件程序中的编程应采用在 NC 编程手册中介绍的输入精度。

所需的计算精度是由机床数据：

MD10200 \$MN_INT_INCR_PER_MM（线性位置计算精度）

6.2 速度、运行范围、精度

与

MD10210 \$MN_INT_INCR_PER_DEG（角位置计算精度）

定义的。

这两个值与输入/显示精度无关，但它们至少应具有相同的精度。

利用计算精度可以确定零件程序中的位置值、速度等的小数点后的最大有效位，也可以确定刀具补偿值、零点偏移值等的小数点后的有效位（这样可以达到最高的精度）。

编程值的乘积通过计算精度取整，从而使输入的角度位置和线性位置的精度被限制在计算精度内。

为了确保所执行的取整易于跟踪，建议使用 10 的幂来表示计算精度。

取整示例：

计算精度：1000 增量/毫米

编程的行程：97.3786 mm

有效值 = 97.379 毫米

¹/₁₀-µm 范围内的编程示例：

所有线性轴都应在 0.1 - 1000 µm 范围中编程及运行。

⇒ 为精确定位到 0.1µm，必须将计算精度设为 ≥ 10⁴ 增量/毫米。

⇒ MD10200 \$MN_INT_INCR_PER_MM = 10000 [增量/毫米]：

⇒ 相关的零件程序示例：

程序代码	注释
N20 G0 X1.0000 Y1.0000	； 轴运行到位置 X=1.0000 mm, Y=1.0000 mm
N25 G0 X5.0002 Y2.0003	； 轴运行到位置 X=5.0002 mm, Y=2.0003 mm

6.2.5 机床数据和设定数据物理量的定标

输入/输出单位

根据所使用的基本单位制（公制/英制），有物理单位的机床数据和设定数据是以下列输入/输出单位来说明的：

物理单位	标准基本单位制的输入/输出单位	
	公制	英制
线性位置	1 毫米	1 英寸
角度位置	1 度	1 度
线性速度	1 毫米/分钟	1 英寸/分钟
角速度	1 转/分	1 转/分
线性加速度	1 米/秒 ²	1 英寸/秒 ²
角加速度	1 转/秒 ²	1 转/秒 ²
线性急动度	1 米/秒 ³	1 英寸/秒 ³
角急动度	1 转/秒 ³	1 转/秒 ³
时间	1 秒	1 秒
位置控制回路增益	1/秒	1/秒
旋转进给率	1 毫米/转	英寸/转
线性位置补偿值	1 毫米	1 英寸
角位置补偿值	1 度	1 度

控制系统内部使用以下单位，与所选择的基本单位制无关：

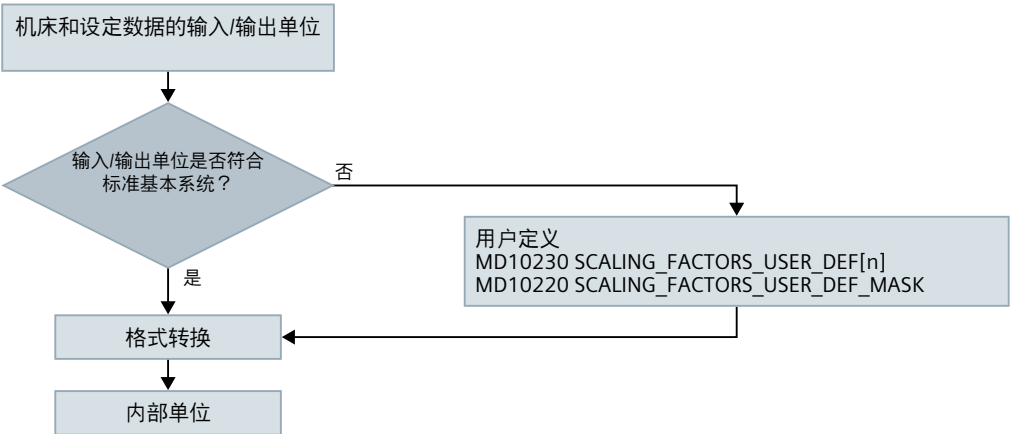
物理单位	单位
线性位置	1 毫米
角度位置	1 度
线性速度	1 毫米/秒
角速度	1 度/秒
线性加速度	1 毫米/秒 ²
角加速度	1 度/秒 ²
线性急动度	1 毫米/秒 ³
角急动度	1 度/秒 ³
时间	1 秒

6.2 速度、运行范围、精度

物理单位	单位
位置控制回路增益	1/秒
旋转进给率	1 毫米/度
线性位置补偿值	1 毫米
角位置补偿值	1 度

用户可以为机床数据和设定数据定义其他的输入/输出单位。为此必须通过以下机床数据实现新选择的输入/输出单位和内部单位之间的匹配：

- MD10220 \$MN_SCALING_USER_DEF_MASK
- MD10230 \$MN_SCALING_FACTORS_USER_DEF[n]



其中：

所选的输入/输出单位 = (MD10230 \$MN_SCALING_FACTORS_USER_DEF[n]) * 内部单位
因此在机床数据 MD10230 \$MN_SCALING_FACTORS_USER_DEF[n] 中须相应输入所选择的输入/输出单位，以内部单位 1 毫米、1 度和 1 秒表示。

示例 1：

线性速度的机床数据输入/输出应该以米/分替代毫米/分（初始设置）。
（内部单位为毫米/秒）

- ⇒ 线性速度的定标系数应不同于初始设置，为此，必须在机床数据：
MD10220 \$MN_SCALING_USER_DEF_MASK
中将位号设为 2。
⇒ MD10220 \$MN_SCALING_USER_DEF_MASK = 'H4'; （位号 2 作为十六进制值）

⇒ 线性速度的定标系数应不同于初始设置，为此，必须在机床数据：
MD10220 \$MN_SCALING_USER_DEF_MASK
中将位号设为 2。
⇒ MD10220 \$MN_SCALING_USER_DEF_MASK = 'H4'; （位号 2 作为十六进制值）

⇒ 定标系数按下列公式计算：

MD10230 SCALING_FACTORS_USER_DEF[n] =

所选的输入/输出单位

内部单位

MD10230 SCALING_FACTORS_USER_DEF[n] =

$\frac{1 \frac{\text{m}}{\text{min}}}{1 \frac{\text{mm}}{\text{s}}}$

=

$\frac{\frac{1000 \text{ mm}}{60 \text{ s}}}{1 \frac{\text{mm}}{\text{s}}}$

=

$\frac{1000}{60}$

=

16.667;

⇒ MD10230 SCALING_FACTORS_USER_DEF[n] =

16.667

下标 n 定义了“物理量定标系数”列表中的“线性速度”。

示例 2:

另外针对示例 1 的改变，机床数据输入/输出以（基本设置）英尺/秒²代替米/秒²。
（内部单位为毫米/秒²）

⇒ MD10220 SCALING_USER_DEF_MASK =

H14; （示例 1 的位编号 4 和 2 作为十六进制值）

⇒ MD10220 SCALING_FACTORS_USER_DEF[4] =

$\frac{1 \frac{\text{ft}}{\text{s}^2}}{1 \frac{\text{mm}}{\text{s}^2}}$

=

$\frac{12 \cdot 25,4 \frac{\text{mm}}{\text{s}^2}}{1 \frac{\text{mm}}{\text{s}^2}}$

=

304.8;

⇒ MD10220 SCALING_FACTORS_USER_DEF[4] =

304.8

下标 4 定义了“物理量定标系数”列表中的“线性加速度”。

进给轴和主轴
功能手册, 01/2023, A5E47437747F AF

149

6.3 英制/公制单位制

6.3 英制/公制单位制

6.3.1 功能

6.3.1.1 参数设置和编写的单位制

SINUMERIK 控制系统既可采用公制单位制，也可采用英制单位制工作。

参数设置的单位制（基本单位制）

单位制的基本设置（基本单位制）在机床数据 MD10240 \$MN_SCALING_SYSTEM_IS_METRIC 中设置（参见“调试(页 151)”）。

根据基本单位制，所有与长度相关的数据均以公制或英制单位制表示：

公制单位制：	mm, mm/min, m/s2, m/s3, mm/rev
英制单位制：	inch, inch/min, inch/s2, nch/s3, inch/rev

单位制也决定了为线性轴编写的 F 值的含义：

进给类型	公制单位制	英制单位制
G94	mm/min	mm/min
G95	mm/rev	inch/rev

编写的单位制

通过 G 功能组 13 的指令（英制/公制）可在零件程序内在公制与英制之间进行切换（参见“编程(页 155)”）。

在任何时候，编写的单位制与基本单位制可相同或不同。借助在零件程序段中切换单位制，例如可以在采用公制基本单位制的情况下在工件上加工英制螺纹。

G 功能组的初始设置在机床数据 MD20150 \$MC_GCODE_RESET_VALUES 中定义（参见“调试(页 151)”）。

NC/PLC 接口

对于包含尺寸相关信息的 NC/PLC 接口信号，如轨迹轴和定位轴的进给率，以基本单位制与 PLC 进行数据交换。

读入外部零件程序

在从外部读入零件程序时，若其包含的数据组（零点偏移、刀具补偿等）的编程所采用的单位制不同于基本单位制，则须预先修改基本单位制。

6.3.1.2 扩展单位制功能

自 SW 5 起提供扩展单位制功能，其使得单位制切换大幅简化。

新功能为：

- 新的 G 指令 G700/G710（参见“编程 (页 155)”）
- 借助单位制标识 INCH/METRIC 进行数据备份（参见“调试 (页 151)”）
- 在单位制切换时自动换算数据，例如针对：
 - 零点偏移
 - 补偿数据（EEC，QEC）
 - 刀具补偿
 - ...

出于兼容性原因，必须通过机床数据激活扩展单位制功能（参见“调试 (页 151)”）。

6.3.2 调试

NC 专用机床数据

基本单位制

通过以下机床数据定义在数据输入/输出时，控制系统为缩放长度相关物理量而使用的基本单位制：

MD10240 \$MN_SCALING_SYSTEM_IS_METRIC

值	含义
0 (= FALSE)	长度相关数据作为 英制 尺寸数据。
1 (= TRUE)	长度相关数据作为 公制 尺寸数据。

修改此机床数据后需要启动，否则相应具有物理单位的机床数据会被错误地标准化。

6.3 英制/公制单位制

遵守以下步骤：

- 通过手动输入修改 MD：
 - 1. 执行启动。
 - 2. 具有物理单位的机床数据：输入值。
- 通过机床数据文件修改 MD：
 - 1. 执行启动。
 - 2. 再次载入机床数据文件，从而将新的物理范围考虑在内。

在修改 MD10240 时，系统会发出报警 4070“标准化机床数据被修改”。

说明

在控制系统上进行单位制切换后，必须将所有长度相关数据一致且完整地换算至另一单位制。

换算系数（NC 专用）

从公制到英制的换算系数在以下机床数据中设置：

MD10250 \$MN_SCALING_VALUE_INCH（针对英制的换算系数）

缺省值：25.4

在选择非公制基本单位制（MD10240 \$MN_SCALING_SYSTEM_IS_METRIC = 0）时，此换算系数生效。

系统会将下列数据与换算系数相乘：

- 输入/输出时的长度相关数据（例如在上载机床数据、零点偏移时）
- 为线性轴编写的 F 值
- 在用 G70/G71 编写的单位制不同于基本单位制（MD10240 \$MN_SCALING_SYSTEM_IS_METRIC）的情况下，编写的轴几何数据（位置、多项式系数、圆弧编程中的半径等）

说明

通过修改换算系统，能够将控制系统调整成用户专用的单位制。

扩展单位制功能

扩展单位制功能可通过以下兼容性机床数据激活：

MD10260 \$MN_CONVERT_SCALING_SYSTEM

值	含义
0 (= FALSE)	扩展单位制功能不生效（缺省设置；与先前的软件版本兼容）
1 (= TRUE)	扩展单位制功能生效

在修改 MD10240 时，系统会发出报警 4070“标准化机床数据被修改”。

数据备份

在读取过程中，从控制系统读取并具有单位制相关数据的各数据组会得到一个标识，其对应当前的单位制设置：

- MD10260 \$MN_CONVERT_SCALING_SYSTEM
- MD10240 \$MN_SCALING_SYSTEM_IS_METRIC

通过此标识可确定以哪种单位制读取数据。这样可避免将具有与当前设置不同的单位制的数据组读入控制系统。

在零件程序中也会分析该标识，故能以上述方式防止误操作。这样可以防止例如只包含公制数据的零件程序在英制系统中运行。

采用以下设置时，存档和机床数据组向下兼容：

MD11220 \$MN_INI_FILE_MODE = 2

说明

仅在以下兼容性机床数据置位时，才会生成 INCH/METRIC 标识：

MD10260 \$MN_CONVERT_SCALING_SYSTEM = TRUE

说明**机床数据取整**

为了避免取整问题，所有与长度相关的机床数据在以英制单位制 (MD10240 \$MN_SCALING_SYSTEM_IS_METRIC=0 和 MD10260 \$MN_CONVERT_SCALING_SYSTEM=1) 写入时都被取整为 1 μm。

这样一来，以英制单位制读取备份数据时因转换为 ASCII 而产生的精度损失会在数据读入系统中时重新得到修正。

输入和计算精度

输入精度和计算精度以每毫米的内部增量数定义。

MD10200 \$MN_INT_INCR_PER_MM

6.3 英制/公制单位制

线性位置的输入精度被限制为计算精度。此时编写的位置值同计算精度的乘积会取整为最接近的整数。为了使取整易于理解，建议使用 10 的幂来表示计算精度。

示例：

$$1 \text{ Inch} = 25.4 \text{ mm} \Rightarrow 0.0001 \text{ Inch} = 0.00254 \text{ mm} = 2.54 \text{ }\mu\text{m} = 2540 \text{ nm}$$

为了编写和显示最后的 40 nm，必须将输入精度和计算精度的值设为 100000。

两种单位制的给定值必须相同，这样才能保证单位制的切换不会损失精度。如果已做过此设定，不需要每次转换系统时，对 MD10200 进行修改。

位置表的单位制

分度轴表格中的位置数据及软件凸轮切换点的单位制是通过以下机床数据进行配置的：

MD10270 \$MN_POS_TAB_SCALING_SYSTEM

更多信息：

功能手册之进给轴和主轴；软件凸轮；行程开关信号和分度轴

用户刀具数据

对于用户自定义的刀具数据和刀沿数据，可通过以下机床数据为其设置物理单位：

- MD10290 \$MN_CC_TDA_PARAM_UNIT
- MD10292 \$MN_CC_TOA_PARAM_UNIT

说明

切换单位制时，所有长度相关刀具数据均会换算至新的单位制。

通道专用机床数据

初始设置

在以下机床数据中为每个通道设置 G 功能组 13（英制/公制单位制）的初始设置：

MD20150 \$MC_GCODE_RESET_VALUES[12]（G 功能组 13 的初始设置）

轴/主轴专用机床数据

换算系数（轴专用）

为使定位轴独立于 G70/G71，可为每个轴单独设置从公制到英制单位制的换算系数：

MD31200 \$MA_SCALING_FACTOR_G70_G71（G70/G71 指令激活时的换算系数）

缺省值：25.4

说明

所有三个几何轴的换算系数应相同。

JOG 和手轮权重

以下机床数据包含两个值，分别包含针对两个单位制中的一个的轴专用增量加权：

MD31090 \$MA_JOG_INCR_WEIGHT

根据 MD10240 SCALING_SYSTEM_IS_METRIC 的当前设置，控制系统自动选择匹配值。

用户在调试阶段就已经确定了两个增量加权。

示例：增量加权，用于第 1 轴

- 公制：

MD31090 \$MA_JOG_INCR_WEIGHT[0 ; AX1] = 0.001 mm

- 英制：

MD31090 \$MA_JOG_INCR_WEIGHT[1 ; AX1] = 0.00254 mm $\triangleq$ 0.0001 inch

这样就无需在每次切换英制/公制时写入 MD31090。

由于所有内部位置始终以 mm 为基准，切换单位制时 JOG 增量运行不会累加任何余程。

垂度补偿的单位制

垂度补偿的单位制是通过以下机床数据进行配置的：

MD32711 \$MA_CEC_SCALING_SYSTEM_METRIC

更多信息：

功能手册之监控与补偿：补偿

6.3.3 编程

6.3.3.1 切换单位制 (G70/G71/G700/G710)

通过 G 指令组 13 的指令（英制/公制）可在零件程序范围内在公制与英制之间进行切换。

激活

为了能使用指令 G700 和 G710，必须激活扩展单位制功能（MD10260 \$MN_CONVERT_SCALING_SYSTEM = 1）。

6.3 英制/公制单位制

句法

G70
G71
G700
G710

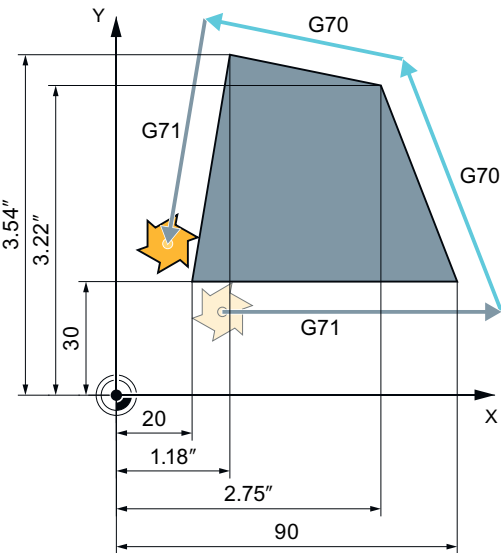
含义

G70:	激活英制单位制	
	在 英制单位制 中读取和写入 和长度相关的几何数据 。	
	在 设置的基本单位制 中读取和写入 和长度相关的的工艺数据 （如进给率、刀具补偿、可设定零点偏移、机床数据和系统变量）。	
	G 功能组:	13
	初始设置:	可通过 MD20150 \$MC_GCODE_RESET_VALUES 设置
G71:	生效方式:	模态
	激活公制单位制	
	在 公制单位制 中读取和写入 和长度相关的的几何数据 。	
	在 设置的基本单位制 中读取和写入 和长度相关的的工艺数据 （如进给率、刀具补偿、可设定零点偏移、机床数据和系统变量）。	
	G 功能组:	13
G700:	初始设置:	可通过 MD20150 \$MC_GCODE_RESET_VALUES 设置
	生效方式:	模态
	激活英制单位制	
	在 英制单位制 中读取和写入 所有和长度相关的几何数据和工艺数据 。	
	G 功能组:	13
G710:	初始设置:	可通过 MD20150 \$MC_GCODE_RESET_VALUES 设置
	生效方式:	模态
	激活公制单位制	
	在 公制单位制 中读取和写入 所有和长度相关的几何数据和工艺数据 。	
	G 功能组:	13

注意
回转轴的轴专用数据
总是在设置的基本单位制中读取和写入回转轴的轴专用数据。

示例

基本单位制为公制 (MD10240 \$MN_SCALING_SYSTEM_IS_METRIC = 1)。但工件图纸中的尺寸数据为英制。因此会在零件程序中切换为英制。根据英制尺寸数据，会再次切换为公制。



程序代码	注释
N10 G0 G90 X20 Y30 Z2 S2000 M3 T1	; X=20 毫米, Y=30 毫米, Z=2 毫米, F= 快速运行 毫米/分钟
N20 G1 Z-5 F500	; Z=-5 毫米, F=500 毫米/分钟
N30 X90	; X=90 毫米
N40 G70 X2.75 Y3.22	; 编写的单位制: 英制 X=2.75 英寸, Y=3.22 英寸, F=500 毫米/分钟
N50 X1.18 Y3.54	; X=1.18 英寸, Y=3.54 英寸, F=500 毫米/分钟
N60 G71 X20 Y30	; 编写的单位制: 公制 ; X=20 毫米, Y=30 毫米, F= 500 毫米/分钟
N70 G0 Z2	; Z=2 毫米, F=快速运行 毫米/分钟
N80 M30	; 程序结束

6.3 英制/公制单位制

更多信息

在 G70/G71 和 G700/G710 指令下读写数据

数据区	G70 / G71		G700 / G710	
	读取	写入	读取	写入
显示逗号后的位数 (WCS)	P	P	P	P
显示逗号后的位数 (MCS)	G	G	G	G
进给率	G	G	P	P
位置数据 X、Y、Z	P	P	P	P
插补参数 I、J、K	P	P	P	P
圆半径 (CR)	P	P	P	P
极半径 (RP)	P	P	P	P
螺距	P	P	P	P
可编程的框架	P	P	P	P
可设定框架	G	G	P	P
基本框架	G	G	P	P
外部零点偏移	G	G	P	P
轴预设偏移	G	G	P	P
工作区域限制 (G25/G26)	G	G	P	P
保护区	P	P	P	P
刀具补偿	G	G	P	P
与长度相关的机床数据	G	G	P	P
与长度相关的设定数据	G	G	P	P
与长度相关的系统变量	G	G	P	P
GUD	G	G	G	G
LUD	G	G	G	G
PUD	G	G	G	G
R 参数	G	G	G	G
西门子循环	P	P	P	P
增量加权 点动/手轮	G	G	G	G
P: 以编写的单位制读写				
G: 以参数设置的基本单位制读写				

同步动作

说明

读取同步动作中的位置数据

如果在同步动作（条件部分和/或动作部分或工艺功能）中未显式编程程序单位制，系统将始终以基本单位制读取同步动作中的与长度相关的位置数据。

更多信息：功能手册之同步动作

6.4 设定值/实际值系统

6.4.1 简介

控制环

可为闭环控制的进给轴/主轴配置具有以下结构的控制环：

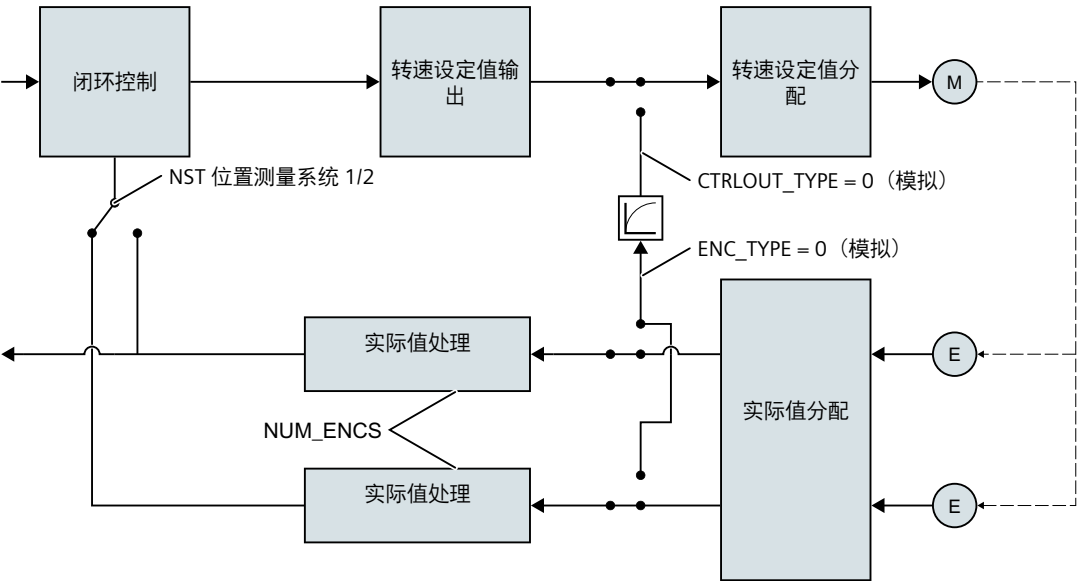


图 6-1 控制环原理图

设定值输出

可为每根进给轴/主轴输出一个设定值报文。在控制系统上将设定值输出至调节器。

6.4 设定值/实际值系统

实际值采集

每根进给轴/主轴最多可连接两个测量系统，例如：一个直接测量系统用于对精度有较高要求的加工过程，一个间接测量系统用于快速定位任务。

所使用的编码器数量输入在以下机床数据中：

MD30200 \$MA_NUM_ENCS（编码器数量）

可在现有的两个实际值分支上进行实际值采集。

位置闭环控制、绝对值计算和显示始终会采用激活的测量系统。如果通过 PLC 接口同时激活了这两个测量系统，控制系统内部会选择位置测量系统 1。

回参考点运行由所选的测量系统执行。

每个位置测量系统必须分开回参考点。

有关编码器监控的详细说明请参见 *监控与补偿功能手册*，*轴监控* 章节。

有关实际值采集补偿功能的详细说明请参见

更多信息：

功能手册之监控与补偿；补偿

切换测量系统

通过以下 NC/PLC 接口信号可在两个测量系统之间进行切换：

<Axis>.basic.out.enc1Activation（位置测量系统 1）

<Axis>.basic.out.enc2Activation（位置测量系统 2）

更多信息

参见功能手册之基本功能；不同的 NC/PLC 接口信号与功能。

可随时在两个测量系统之间进行切换，与轴的静止状态无关，但只有在未超过允许的两个测量系统实际值之间的偏差时才能实现切换。

相应的公差输入在以下机床数据中：

MD36500 \$MA_ENC_CHANGE_TOL（位置实际值切换时的最大公差）

切换时，位置测量系统 1 和 2 之间的当前偏差会立即运行。

监控

允许的两个测量系统实际值之间的偏差必须输入在以下机床数据中：

MD36510 \$MA_ENC_DIFF_TOL

在循环比较所使用的两个测量系统时，不能超过这个差值，否则会生成报警 25105 “测量系统差值过大”。

如果轴中的 2 个测量系统无一生效或是轴未回参考点（至少是在当前的闭环测量系统中），在 MD36510 = 0 时，相应的监控也不会生效。

实际值采集方式

所使用的编码器类型必须通过以下机床数据确定：

MD30240 \$MA_ENC_TYPE（实际值采集方式（位置实际值））

模拟轴

出于测试的目的，可对轴的转速控制回路进行模拟。

与真实轴相同，模拟轴运行时同样会出现跟随误差。

模拟轴是通过将以下两个机床数据置“0”来定义的：

MD30130 \$MA_CTRLOUT_TYPE[n]（设定值输出方式）

MD30240 \$MA_ENC_TYPE[n]（实际值采集方式）

载入缺省机床数据后，这些轴就成了模拟轴。

通过回参考点运行可将设定值/实际值设为回参考点值。

通过机床数据：

MD30350 \$MA_SIMU_AX_VDI_OUTPUT（模拟轴上的轴信号输出）

可以确定模拟期间轴专用的接口信号是否输出到 PLC 上。

实际值补偿

如果希望 NC 对所选位置控制编码器进行的实际值补偿不影响该轴上定义的其他编码器的实际值，则需要通过以下机床数据将该编码器定义为“独立”编码器：

MD30242 \$MA_ENC_IS_INDEPENDENT

实际值补偿包括：

- 模数处理
- 回参考点运行
- 测量系统调整
- PRESET（预设）

6.4 设定值/实际值系统

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.enc1Activation	LBP_Axis*.A_PosMeas1	DB31, ...DBX1.5
<Axis>.basic.out.enc2Activation	LBP_Axis*.A_PosMeas2	DB31, ...DBX1.6

6.4.2 设定值和编码器分配

设定值分配

机床轴的设定值分配与以下机床数据有关。

MD30100		\$MA_CTRLOUT_SEGMENT_NR
设定值分配：总线段		
系统	值	含义
	5	PROFINET IO

MD30110		\$MA_CTRLOUT_MODULE_NR
设定值分配：驱动编号/模块编号		
系统	值	含义
	x	通过驱动号从 MD13050 \$MN_DRIVE_LOGIC_ADDRESS[n] 中分配驱动的逻辑 I/O 地址。 驱动号 (x) 是由 MD13050 的索引 (n) 得出的： $x = n + 1$ 提示： 如果是模拟驱动 (MD30130 \$MA_CTRLOUT_TYPE = 0)，该机床数据没有任何意义。

MD30120		\$MA_CTRLOUT_NR
设定值分配：驱动模块/模块上的设定值输出		
系统	值	含义
	1	带有 PROFIdrive 协议的 PROFINET 上的模块化驱动

MD30130		\$MA_CTRLOUT_TYPE
设定值输出方式		
系统	值	含义
	0	模拟（无驱动运行）
	1	设定值输出有效

编码器分配

在 PROFIdrive 报文中传送的编码器信息和对应的机床轴编码器输入通过以下机床数据设定：

MD30210		\$MA_ENC_SEGMENT_NR[n]
实际值分配总线段		
系统	值	含义
	5	PROFINET

MD30220		\$MA_ENC_MODULE_NR[n]
实际值分配：驱动号/测量回路号		
系统	值	含义
	x	通过驱动号从 MD13050 \$MN_DRIVE_LOGIC_ADDRESS[n] 中分配驱动的逻辑 I/O 地址。 驱动号 (x) 是由 MD13050 的索引 (n) 得出的： $x = n + 1$

6.4 设定值/实际值系统

MD30230		\$MA_ENC_INPUT_NR[n]
实际值分配：驱动模块上的输入端/测量回路		
系统	值	含义
	x	PROFIdrive 报文内部的编码器接口编号 示例 PROFIdrive 报文 103 x = 1 → 第 1 编码器接口 (G1_ZSW、G1_XIST1、G1_XIST2) x = 2 → 第 2 编码器接口 (G2_ZSW、G2_XIST1、G2_XIST2) PROFIdrive 报文 118 x = 1 → 第 1 编码器接口 (G2_ZSW、G2_XIST1、G2_XIST2) x = 2 → 第 2 编码器接口 (G3_ZSW、G3_XIST1、G3_XIST2) 提示： 就 SINAMICS S120 而言： - 编码器 1 (G1_...)：电机编码器 - 编码器 2 (G2_...)：直接测量系统 - 编码器 3 (G3_...)：附加增量系统

MD30240		\$MA_ENC_TYPE[n]
检测实际值的编码器类型（位置实际值）		
系统	值	含义
	0	模拟（无编码器运行）
	1	增量编码器
	4	绝对值编码器
提示： 与 PROFIdrive 参数 p979 相符		

MD30242		\$MA_ENC_IS_INDEPENDENT[n, 轴]
编码器是独立的		
系统	值	含义
	0	编码器不是独立的。
	1	编码器是独立的。 如果对所选位置控制编码器进行的实际值补偿不影响该轴上定义的第二编码器的实际值，则需要通过以下机床数据将该编码器定义为“独立”编码器： 实际值补偿包括： • - 模数处理 • - 回参考点运行 • - 测量系统调整 • - PRESET（预设） 示例：一根轴、2 个编码器、第 2 个编码器是独立的 MD30200 \$MA_NUM_ENCS[AX1] = 2 MD30242 \$MA_ENC_IS_INDEPENDENT[0, AX1] = 0 MD30242 \$MA_ENC_IS_INDEPENDENT[1, AX1] = 1 选择位置测量系统 1 / 2: <Axis>.basic.out.enc1Activation / <Axis>.basic.out.enc2Activation 如果为位置闭环控制选择了编码器 1，由于编码器 2 是独立的，那么只在该编码器上执行实际值补偿。 如果为位置闭环控制选择了编码器 2，由于编码器 1 不是独立的，那么两个编码器上都会执行实际值补偿。 即：机床数据只对机床轴上的被动编码器起作用。
	2	被动编码器不是独立的。 编码器实际值是通过主动编码器修改的。与 MD35102 \$MA_REFP_SYNC_ENCS = 1 组合后，在进行回参考点运行时，被动编码器会自动根据主动编码器调整，但不回参考点。 在回参考点模式 MD34200 \$MA_ENC_REFP_MODE = 3（距离编码的参考标记）中，在驶过零脉冲距离后，被动编码器会自动在下一个运行动作时回参考点。这与当前的运行方式设置无关。
	3	编码器是独立的。

6.4 设定值/实际值系统

MD30242		\$MA_ENC_IS_INDEPENDENT[n, 轴]
编码器是独立的		
系统	值	含义
		在模数回转轴上，模数实际值补偿也是在被动编码器中执行的。

说明

机床数据下标 [n]

用于编码器分配的机床数据下标 [n] 有以下含义：

- n = 0:分配给机床轴的第一个编码器
- n = 1:分配给机床轴的第二个编码器

该分配是通过以下机床数据进行的：

- MD30220 \$MA_ENC_MODULE_NR[n]
- MD30230 \$MA_ENC_INPUT_NR[n]

更多信息

CNC 调试手册：NC、PLC、驱动；NC 和驱动之间的通讯 > 驱动：轴分配

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.enc1Activation	LBP_Axis*.A_PosMeas1	DB31, ...DBX1.5
<Axis>.basic.out.enc2Activation	LBP_Axis*.A_PosMeas2	DB31, ...DBX1.6

齿轮箱类型

齿轮箱类型	激活	调整	安装位置
“电机-负载” 齿轮箱	参数组	固定配置	齿轮箱盒
“编码器-测量” 齿轮箱	上电	随编码器变化	编码器侧的
“负载-中间” 齿轮箱	热启动	随负载变化	刀具侧的

齿轮箱/编码器的局部位置

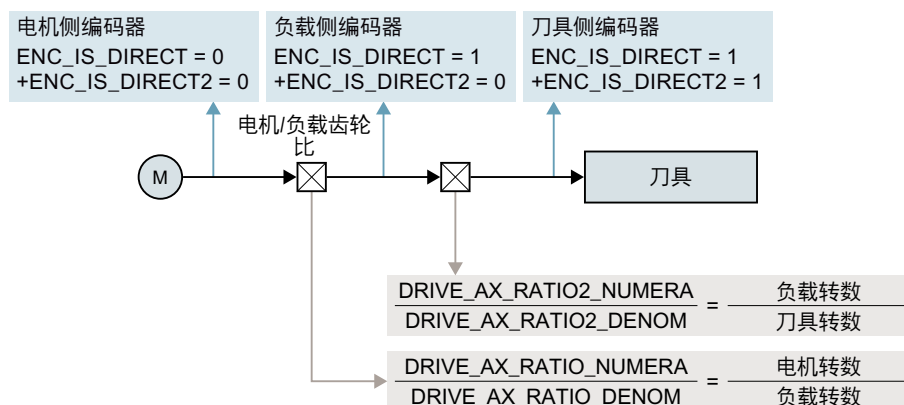


图 6-2 齿轮箱类型和编码器安装位置

“电机-负载”齿轮箱

SINUMERIK 支持的“电机-负载”齿轮箱是通过以下机床数据配置的:

- MD31060 \$MA_DRIVE_AX_RATIO_NUMERA (负载齿轮比分子)
- MD31050 \$MA_DRIVE_AX_RATIO_DENOM (负载齿轮比分母)

齿轮比是由两个机床数据的分子与分母之比得出的。控制系统会通过相应的参数组自动将位置控制器与各个齿轮比同步。

6.4 设定值/实际值系统

由于齿轮箱不必始终自动进行换档且有多种方式进行换档，因此，位置控制并不总是通过参数组整合在一起。

说明

有关齿轮箱换档时参数组的更多信息请参考章节“主轴 (页 395)”。

中间齿轮比

控制系统还支持可设定的负载中间齿轮比：

- MD31066 \$MA_DRIVE_AX_RATIO2_NUMERA（中间齿轮比分子）
- MD31064 \$MA_DRIVE_AX_RATIO2_DENOM（中间齿轮比分母）

动力刀具通常都有它们“自己的”中间齿轮比。可通过增加一个和“电机-负载”齿轮箱相乘的中间齿轮比来设置此类变速装置。

 小心

切换时的不同传动比

与“电机-负载”齿轮箱不同的是，中间齿轮比中没有任何参数组，因此，也就无法控制到零件程序或 PLC（NC/PLC 接口）的同步切换。它可以避免换档时的零件加工。如何协调和激活修改后的机床数据与相应的换档之间的同步，是用户的职责。在轴运行期间进行换档时，由于定标系数会发生跃变，因此，**无法**避免执行补偿过程。NC 不会监控这些补偿过程是否符合最大加速度的要求。

直接连接在刀具上的编码器

中间齿轮比还可以采用另一种用于“刀具侧编码器”的连接方式，这可通过配置以下机床数据实现：

MD31044 \$MA_ENC_IS_DIRECT2

非直接连接在刀具上的编码器

在对处于位置闭环控制模式中的中间齿轮比进行换档时，须注意以下前提条件：此种情况下，NC 会将待切换的齿轮比一同考虑到编码器信息的重新定标中。

此时，处于定位模式中的进给轴/主轴可以：

- 只有在静止状态下齿轮箱换档才不会引起跃变。
由于齿轮箱换档期间机械位置几乎或完全不会发生变化，为此，刀具侧的位置在换档前和换档后的变化与齿轮比发生变化时的情况是相同的。
建议：
为避免报警 21612“运行期间复位了伺服使能”，应只在“静止状态”下进行换档，且仍允许并建议在换档前或换档期间将进给轴或主轴切换至转速控制模式或跟踪运行模式。

前提条件

如果用于位置控制的编码器直接连接在了刀具上，那么电机-负载齿轮箱或中间齿轮比的换档只会影响 NC 和驱动器之间的转速接口上的物理量。此时，系统内部的参数组不会自动切换。

参考点和机床基准

 小心

机床基准丢失
控制系统无法识别出所有可能导致机床位置参考基准的状态，因此，出现此类情况时，调试人员或用户有责任主动地进行回参考点或执行零脉冲同步。

齿轮箱换档会对编码器定标产生影响，因此失去了参考点基准或机床位置基准。在此类情况下，控制系统有时会取消状态“已回参考点/已同步”。
如果机床基准丢失，必须通过校准或回参考点过程恢复已经丢失的机床基准。

参见

回参考点 (页 35)

6.4.4 转速设定值输出

进给轴的调节方向和运行方向

开始加工前须确定进给轴的运行方向。
调节方向
调试位置控制前，须对驱动的转速控制器和电流控制器进行调试和优化。

6.4 设定值/实际值系统

运行方向

通过机床数据：

MD32100 \$MA_AX_MOTION_DIR（运行方向）

可以切换轴的运行方向，
且不会影响位置控制的调节方向。

转速设定值调整

在转速设定值调整中，NC 会得知哪个转速设定值对应驱动中的哪个电机转速，以便设置轴的闭环控制和监控。此调整自动进行。

PROFINET IO 驱动上还可以选择手动调整转速设定值。

- **手动调整**

在机床数据：

MD32250 \$MA_RATED_OUTVAL

输入一个不为零的值。

说明

速度调整和最大转速设定值

由于本身已经具备了自动转速设定值调整功能，无需再进行速度调整！

最大转速设定值

最大转速设定值以百分比定义。100% 表示最大转速设定值或 PROFIdrive 驱动器上的最大转速（驱动中制造商专用的设置参数，如 SINAMICS 中的 p1082）。

主轴转速是在 NC 中输出的。

在控制系统中包含了 5 个齿轮档的数据。

齿轮档通过该档位的最小转速和最大转速以及用于自动换档的最小转速和最大转速定义。只有当新编程的转速设定值超出当前齿轮档时才会输出新的目标齿轮档。

通过机床数据：

MD36210 \$MA_CTRLOUT_LIMIT[n]（最大转速设定值）

可按百分比形式限制转速设定值。

最高可达 200%。

超出限值时，系统会输出报警。

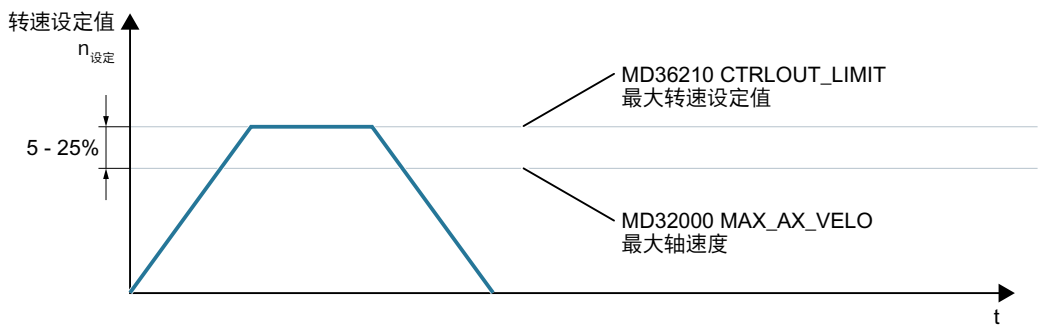


图 6-3 最大转速设定值

但是为改善控制，轴在 80% 到 95% 的转速设定值时就应达到最大速度 (MD32000 \$MA_MAX_AX_VELO)，而不是在 100% 的转速设定值时才达到。

如果轴在大约 80% 的转速设定值时达到了最大转速，则机床数据（缺省值 80%）：
MD32000 \$MA_MAX_AX_VELO（最大轴速度）
生效。

6.4.5 实际值系统的机床数据

轴专用机床数据

为了参数设置实际值系统，须设置以下轴专用机床数据：

与编码器和参数组无关的机床数据：\$MA_	含义
MD30200 NUM_ENCS	编码器数量
MD30300 IS_ROT_AX	回转轴/主轴
MD30310 ROT_IS_MODULO	回转轴/主轴的模数转换
MD30320 DISPLAY_IS_MODULO	回转轴和主轴的 360 度模数显示
MD30330 MODULO_RANGE	模数区域的大小
MD30340 MODULO_RANGE_START	模数区域的起始位置
MD31030 \$MA_LEADSCREW_PITCH	丝杠螺距
MD31064 \$MA_DRIVE_AX_RATIO2_DENOM	中间齿轮比分母
MD31066 \$MA_DRIVE_AX_RATIO2_NUMERA	中间齿轮比分子
MD32000 \$MA_MAX_AX_VELO	最大轴速度

6.4 设定值/实际值系统

编码器相关的机床数据：\$MA_	含义
MD30210 ENC_SEGMENT_NR[n]	实际值分配：总线段的编号
MD30220 ENC_MODULE_NR[n]	实际值分配：驱动编号/测量回路编号
MD30230 ENC_INPUT_NR[n]	实际值分配：驱动模块/测量回路上的输入
MD30240 ENC_TYPE[n]	检测实际值的编码器类型（位置实际值）
MD30242 ENC_IS_INDEPENDENT[n]	编码器是独立的
MD30244 ENC_MEAS_TYPE[n]	编码器测量类型
MD30250 ACT_POS_ABS[n]	内部编码器位置
MD30260 ABS_INC_RATIO[n]	绝对值编码器：绝对分辨率与增量分辨率的比例
MD30270 ENC_ABS_BUFFERING[n]	绝对值编码器：运行范围扩展
MD34090 \$MA_REFP_MOVE_DIST_CORR[n]	参考点偏移
MD34320 \$MA_ENC_INVERS[n]	长度测量系统方向相反
n:编码器下标，其中 n = 0、1、...（第 1 编码器、第 2 编码器、...）	

说明
可以在零件程序中通过 NEWCONF 指令，或者借助操作界面通过软键激活“激活机床数据”。

6.4.6 实际值精度

6.4.6.1 实际值分辨率的机床数据

根据测量系统（编码器）所安装于的位置，须考虑以下测量系统类型：

- 负载侧编码器：直接测量系统（DM）
- 电机侧编码器：间接测量系统（IM）

根据轴类型（线性轴/回转轴）参数设置实际值分辨率

基于以下机床数据计算控制系统的实际值分辨率。

用于计算实际值分辨率的机床数据	线性轴	线性轴		回转轴	
	光栅尺/ 直接测量系统	间接测量 系统	直接测量 系统：机 床/刀具	间接测 量系统	直接测量 系统：机 床/刀具
MD30300 \$MA_IS_ROT_AX	0	0	0	1	1
MD31000 \$MA_ENC_IS_LINEAR[n]	1	0	0	0	0
MD31010 \$MA_ENC_GRID_POINT_DIST[n]	分度	-	-	-	-
MD34320 \$MA_ENC_INVERS[n]	1)	-	-	-	-
MD31020 \$MA_ENC_RESOL[n]	-	增量/ 转	增量/ 转	增量/ 转	增量/ 转
MD31025 \$MA_ENC_PULSE_MULT[n]	编码器倍频				
MD31030 \$MA_LEADSCREW_PITCH	-	毫米/转	毫米/转	-	-
MD31040 \$MA_ENC_IS_DIRECT[n]	- / 1	0	1	0	1
MD31044 \$MA_ENC_IS_DIRECT2[n]	- / 1	0	1	0	1
MD31050 \$MA_DRIVE_AX_RATIO_DENOM[n]	-	负载 转数	- / 1	负载 转数	2)
MD31060 \$MA_DRIVE_AX_RATIO_NUMERA[n]	-	电机 转数，如 果进给 齿轮箱存 在	- / 1	电机 转数	2)
MD31070 \$MA_DRIVE_ENC_RATIO_DENOM[n]	-	编码器 转数	编码器 转数	编码器 转数	编码器 转数
MD31080 \$MA_DRIVE_ENC_RATIO_NUMERA[n]	-	电机 侧编码 器 3)	电机 转数	电机 转数	负载 转数

- 不相关

1) 在有间距编码的测量系统中

6.4 设定值/实际值系统

- 2) 尽管在匹配编码器时并不需要这些机床数据。但在计算设定值时要求这些机床数据必须正确输入！否则不会得到正确的 K_v 系数。在机床数据 MD31050 \$MA_DRIVE_AX_RATIO_DENOM 中输入负载转数，在机床数据 MD31060 \$MA_DRIVE_AX_RATIO_NUMERA 中输入电机转数。
- 3) 电机侧编码器是一个内置编码器，因此，**没有**测量齿轮箱。
齿轮比始终为 1:1。

实际值分辨率的机床数据

实际值精度是由机床的结构、是否有齿轮箱及其齿轮比、线性轴上的丝杠螺距和所用编码器的分辨率得出的。为此，须在控制系统中设置以下机床数据：

与编码器和参数组无关的机床数据：\$MA_	含义
MD30300 IS_ROT_AX	轴是回转轴/主轴
MD31010 ENC_GRID_POINT_DIST	光栅尺的栅距
MD31030 LEADSCREW_PITCH	丝杠螺距
MD31064 DRIVE_AX_RATIO2_DENOM	中间齿轮比分母
MD31066 DRIVE_AX_RATIO2_NUMERA	中间齿轮比分子

取决于编码器的机床数据：\$MA_	含义
MD31000 ENC_IS_LINEAR[n]	测量系统是一个光栅尺
MD31020 ENC_RESOL[n]	每转的编码器线数
MD31025 ENC_PULSE_MULT[n]	编码器倍频（高分辨率）
MD31040 ENC_IS_DIRECT[n]	直接或间接测量系统
MD31044 ENC_IS_DIRECT2[n]	中间齿轮比上的编码器
MD31070 DRIVE_ENC_RATIO_DENOM[n]	测量齿轮比分母
MD31080 DRIVE_ENC_RATIO_NUMERA[n]	测量齿轮比分子
n: 编码器下标, n = 0、1、...（第 1 个编码器、第 2 个编码器、...）	

取决于参数组的机床数据：\$MA_	含义
MD31050 \$MA_DRIVE_AX_RATIO_DENOM[m]	负载齿轮比分母
MD31060 \$MA_DRIVE_AX_RATIO_NUMERA[m]	负载齿轮比分子
m: 参数组下标, m = 0、1、...（第 1 个参数组、第 2 个参数组、...）	

参数设置计算精度

控制系统内部的计算精度与实际值精度之比是衡量由控制系统计算出的值转化精度的标准。

计算精度：线性轴

计算精度

实际值分辨率

=

内部增量/ (mm)

编码器增量/ (mm)

计算精度：回转轴

计算精度

实际值分辨率

=

内部增量/ (度)

编码器增量/ (度)

机床数据

通用机床数据：\$MN_	含义
MD10200 INT_INCR_PER_MM	线性位置计算精度
MD10210 INT_INCR_PER_DEG	角位置计算精度

设置建议

在选择上述影响实际值精度的分量和设置时，应确保实际值精度大于设置的计算精度。

计算精度

实际值分辨率

≤ 1

6.4.6.2 示例：带光栅尺的线性轴

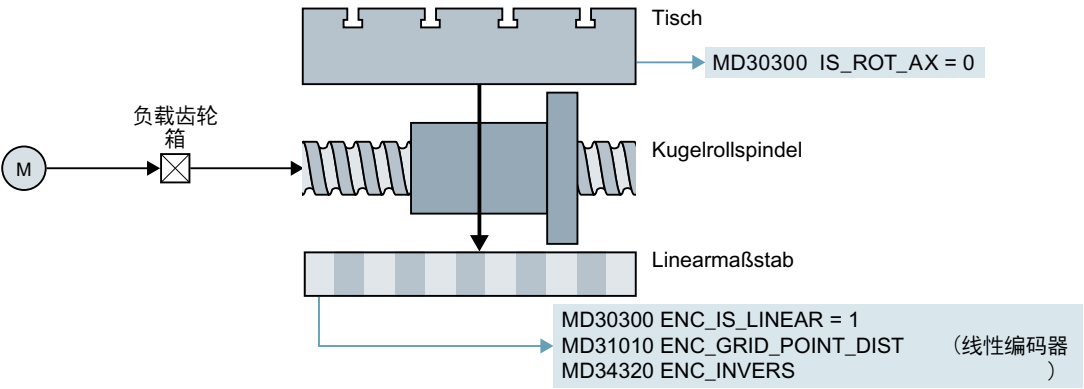


图 6-4 带光栅尺的线性轴

6.4 设定值/实际值系统

内部增量与编码器增量（每毫米）之比是按如下公式得出的：

内部增量/mm

编码器增量/mm

=

ENC_GRID_POINT_DIST [n] * INT_INCR_PER_MM

ENC_PULSE_MULT[n]

6.4.6.3 示例：电机上带旋转编码器的线性轴

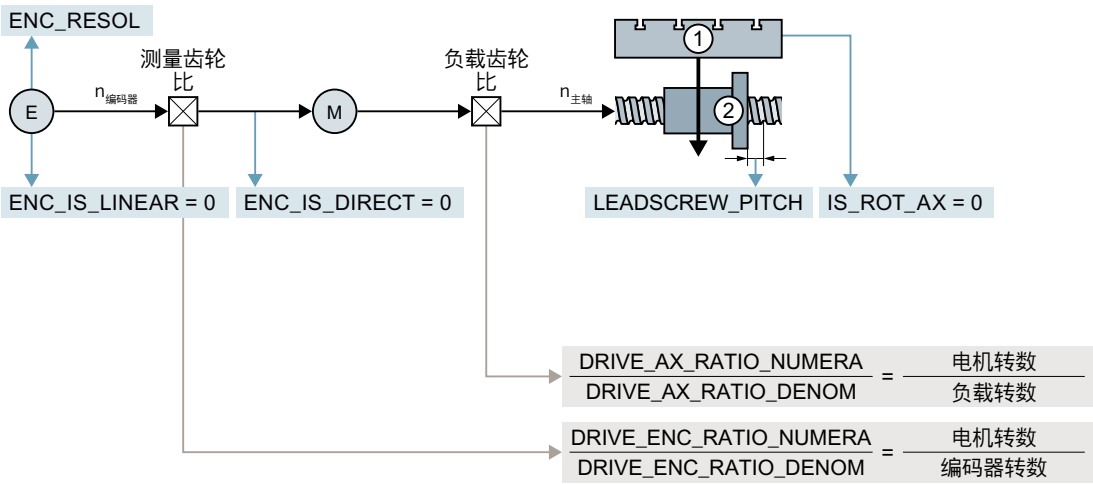


图 6-5 电机上带旋转编码器的线性轴

内部增量与编码器增量（每毫米）之比是按如下公式得出的：

内部增量/mm

编码器增量/mm

=

1

ENC_RESOL [n] * ENC_PULSE_MULT[n]

*

DRIVE_ENC_RATIO_NUMERA [n]

DRIVE_ENC_RATIO_DENOM [n]

*

DRIVE_AX_RATIO_DENOM [n]

DRIVE_AX_RATIO_NUMERA [n]

*

LEADSCREW_PITCH

*

INT_INCR_PER_MM

示例

假设：

- 电机上的旋转编码器：2048 脉冲/转
- 内部脉冲倍频：2048
- 电机/丝杠齿轮比：5:1
- 丝杠螺距：10 毫米/转
- 计算精度：10000 增量/毫米

机床数据	值
MD30300 \$MA_IS_ROT_AX	0
MD31000 \$MA_ENC_IS_LINEAR[0]	0
MD31040 \$MA_ENC_IS_DIRECT[0]	0
MD31020 \$MA_ENC_RESOL[0]	2048
MD31025 \$MA_ENC_PULSE_MULT	2048
MD31030 \$MA_LEADSCREW_PITCH	10
MD31080 \$MA_DRIVE_ENC_RATIO_NUMERA[0]	1
MD31070 \$MA_DRIVE_ENC_RATIO_DENOM[0]	1
MD31060 \$MA_DRIVE_AX_RATIO_NUMERA[0]	5
MD31050 \$MA_DRIVE_AX_RATIO_DENOM[0]	1
MD10210 \$MN_INT_INCR_PER_DEG	10000

➡

内部增量/mm

编码器增量/mm

=

1

2048 * 2048

*

1

1

*

1

5

*

10 mm

*10000 增量/mm

= 0,004768

一个编码器增量等于 0.004768 个内部增量；换句话说：209.731543 个编码器增量等于一个内部增量。

6.4 设定值/实际值系统

6.4.6.4 示例：机床上带旋转编码器的线性轴

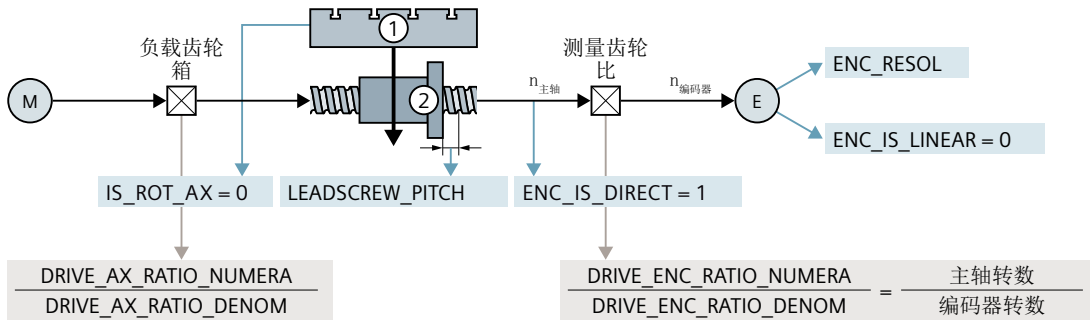


图 6-6 机床上带旋转编码器的线性轴

内部增量与编码器增量（每毫米）之比是按如下公式得出的：

$$\frac{\text{内部增量/mm}}{\text{编码器增量/mm}} = \frac{1}{\text{ENC\_RESOL [n]} * \text{ENC\_PULSE\_MULT[n]}} * \frac{\text{DRIVE\_ENC\_RATIO\_NUMERA [n]}}{\text{DRIVE\_ENC\_RATIO\_DENOM [n]}} * \text{LEADSCREW\_PITCH} * \text{INT\_INCR\_PER\_MM}$$

6.4.6.5 示例：电机上带旋转编码器的回转轴

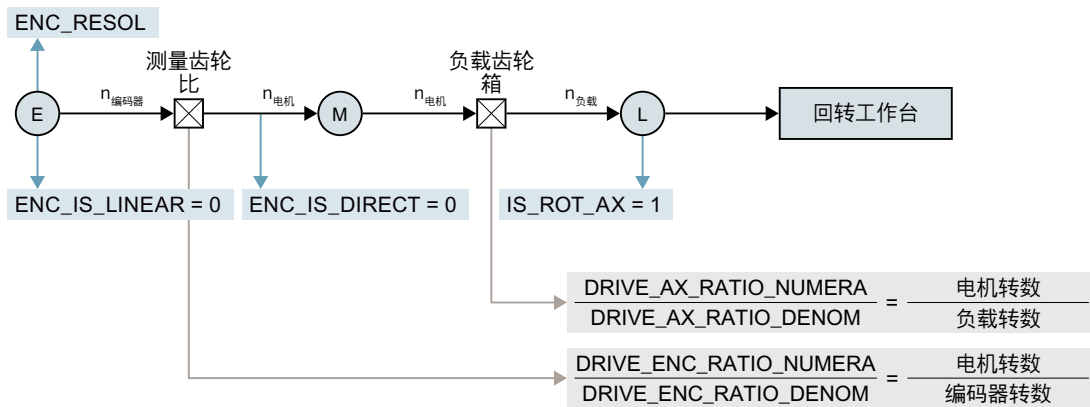


图 6-7 电机上带旋转编码器的回转轴

内部增量与编码器增量（每度）之比是按如下公式得出的：

内部增量/度

编码器增量/度

=

360 度

ENC_RESOL [n] * ENC_PULSE_MULT[n]

*

DRIVE_ENC_RATIO_NUMERA [n]

DRIVE_ENC_RATIO_DENOM [n]

*

DRIVE_AX_RATIO_DENOM [n]

DRIVE_AX_RATIO_NUMERA [n]

* INT_INCR_PER_DEG

示例

- 假设：
- 电机上的旋转编码器：2048 脉冲/转
 - 内部脉冲倍频：2048
 - 电机/回转轴齿轮比：5:1
 - 计算精度： 1000 增量/度

机床数据	值
MD30300 \$MA_IS_ROT_AX	1
MD31000 \$MA_ENC_IS_LINEAR[0]	0
MD31040 \$MA_ENC_IS_DIRECT[0]	0
MD31020 \$MA_ENC_RESOL[0]	2048
MD31025 \$MA_ENC_PULSE_MULT	2048
MD31080 \$MA_DRIVE_ENC_RATIO_NUMERA[0]	1
MD31070 \$MA_DRIVE_ENC_RATIO_DENOM[0]	1
MD31060 \$MA_DRIVE_AX_RATIO_NUMERA[0]	5
MD31050 \$MA_DRIVE_AX_RATIO_DENOM[0]	1
MD10210 \$MN_INT_INCR_PER_DEG	1000

➔

内部增量/度

编码器增量/度

=

360 度

2048 * 2048

*

1

1

*

1

5

*1000 增量/度

= 0,017166

6.4 设定值/实际值系统

一个编码器增量等于 0.017166 个内部增量；换句话说：58.254689 个编码器增量等于一个内部增量。

6.4.6.6 示例：机床上带旋转编码器的回转轴

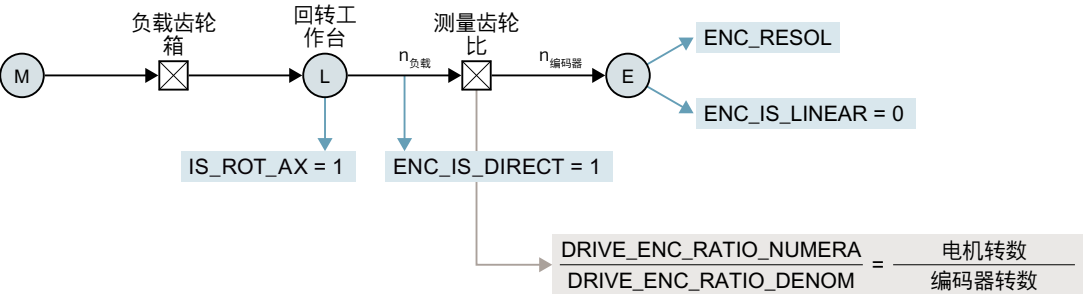


图 6-8 机床上带旋转编码器的回转轴

内部增量与编码器增量（每度）之比是按如下公式得出的：

$$\frac{\text{内部增量/度}}{\text{编码器增量/度}} = \frac{360 \text{ 度}}{ENC\_RESOL[n] * ENC\_PULSE\_MULT[n]}$$
$$* \frac{DRIVE\_ENC\_RATIO\_NUMERA[n]}{DRIVE\_ENC\_RATIO\_DENOM[n]}$$
$$* INT\_INCR\_PER\_DEG$$

6.4.6.7 示例：刀具上带编码器的中间齿轮比

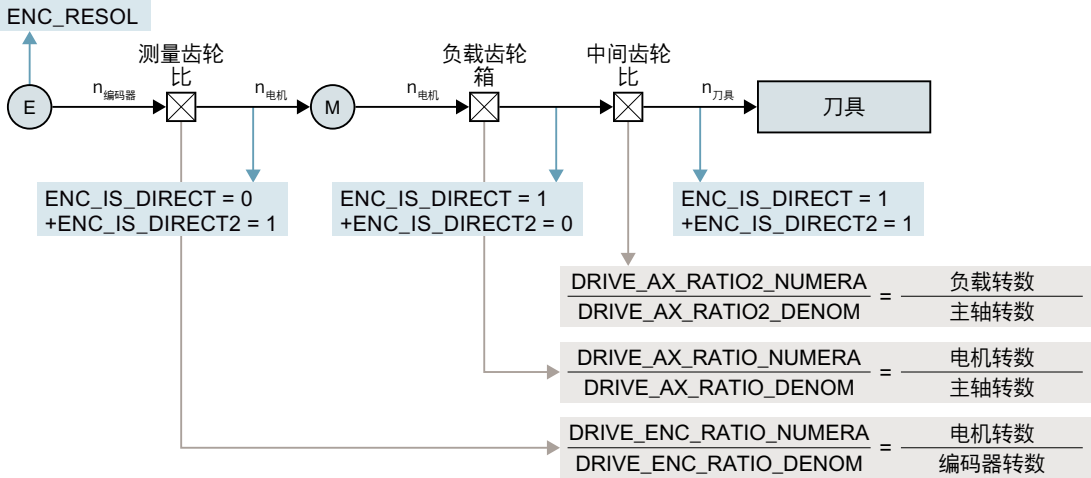


图 6-9 直接连接在动力刀具上的带编码器的中间齿轮比

内部增量与编码器增量（每度）之比是按如下公式得出的：

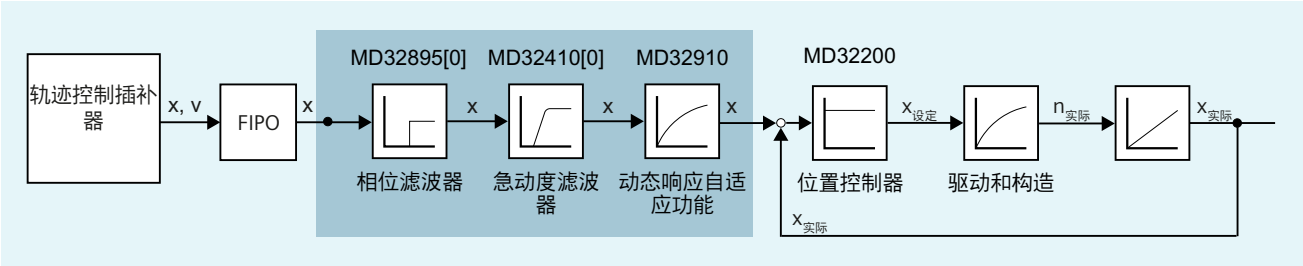
$$\frac{\text{内部增量/度}}{\text{编码器增量/度}} = \frac{360 \text{ 度}}{\text{ENC\_RESOL [n]} * \text{ENC\_PULSE\_MULT[n]}}$$
$$* \frac{\text{DRIVE\_ENC\_RATIO\_NUMERA [n]}}{\text{DRIVE\_ENC\_RATIO\_DENOM [n]}}$$
$$* \text{INT\_INCR\_PER\_DEG}$$

6.5 闭环控制

6.5.1 进给轴/主轴的位置闭环控制

轴的闭环控制包括驱动的电流和转速控制回路以及 NC 中的更高级位置控制回路。

进给轴/主轴的位置闭环控制原则上采用如下结构：



MD32410	\$MA_AX_JERK_TIME
MD32910	\$MA_DYN_MATCH_TIME
MD32895	\$MA_DESVAL_DELAY_TIME
MD32200	\$MA_POSCTRL_GAIN
x	位置设定值
v	速度设定值
x _{实际}	位置实际值
n _{实际}	转速实际值
n _{设定}	转速设定值

6.5 闭环控制

精插补

使用精插补 (FIPO) 时，可通过转速设定值的降低阶梯效应来进一步提高轮廓精度。可设置 3 种类型的精插补：

MD33000 \$MA_FIPO_TYPE = <FIPO 模式>

<FIPO 模式>	含义
1	微分精插补，包含持续一个插补周期的平均值计算（平滑）
2	三次精插补
3	三次精插补，含前馈，可获得最高轮廓精度

K_v 系数

为确保连续路径运行中不会出现较大的轮廓偏差，需要一个高 K_v 系数：

MD32200 \$MA_POSCTRL_GAIN[n]

K_v 系数太大会导致不稳、超调并可能使机床超出许可范围的负载。

允许的最大 K_v 系数取决于：

- 驱动的配置和动态性能
（上升时间、加速和制动能力）
- 机床质量
（弹性、振动阻尼）
- DSC 激活时的位置控制周期或转速控制周期

K_v 系数定义为：

$$K_v = \frac{\text{速度}}{\text{跟随误差}} \quad ; \quad \frac{[\text{m/min}]}{[\text{mm}]}$$

急动限制

有关最大轴急动度的信息请参见章节“加速度和急动度 (页 91)”。

动态响应自适应功能

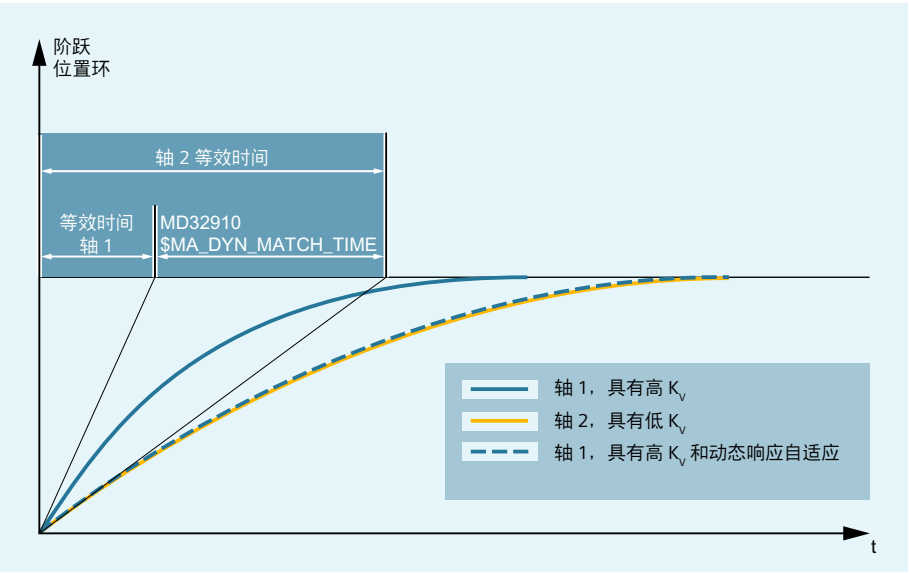
通过动态响应自适应功能能为可相互插补，但具有不同 K_v 系数的轴设置相同的跟随误差。将 K_v 系数降低至动态响应最弱的那根轴，就可以在不损失控制质量的同时进一步优化轮廓精度。

该功能是通过以下机床数据激活的：

MD32900 \$MA_DYN_MATCH_ENABLE = 1（动态响应自适应功能）

动态响应自适应功能是通过输入一个新的等效时间常数来设定的，是由动态性最弱轴与待匹配轴的等效时间常数之差计算得出的：

MD32910 \$MA_DYN_MATCH_TIME[n] = <等效时间常数的差值>



无转速前馈控制的 3 轴动态响应自适应功能示例

位置控制回路的等效时间常数为：	
轴 1：	30 ms
轴 2：	20 ms
轴 3：	24 ms

等效时间常数为 30 ms 的轴 1 是动态响应最弱的那根轴。

从而得出下列新的等效时间常数：

- 轴 1：MD32910 \$MA_DYN_MATCH_TIME = 0 毫秒
- 轴 2：MD32910 \$MA_DYN_MATCH_TIME = 30 毫秒 - 20 毫秒 = 10 毫秒
- 轴 3：MD32910 \$MA_DYN_MATCH_TIME = 30 毫秒 - 24 毫秒 = 6 毫秒

6.5 闭环控制

轴位置控制回路的等效时间常数的近似公式

根据前馈控制类型，轴位置控制回路的等效时间常数 T_{equiv} 的计算方式分为以下三种：

- 不带前馈控制：

$$T_{等效} \approx \frac{1}{MD32200 POSCTRL\_GAIN [1/s]}$$

- 带转速前馈控制：

$$T_{等效} \approx MD32810 EQUIV\_SPEEDCTRL\_TIME$$

- 转矩/转速前馈控制组合时：

$$T_{等效} \approx MD32800 EQUIV\_CURRCTRL\_TIME$$

说明

如果已经激活了一根几何轴的动态响应自适应功能，那么其他所有几何轴的动态响应都必须相同。

6.5.2 可切换的位置设定值—滤波器链条

优点

两个可切换的位置设定值滤波链使得滤波器阻尼效应适应加工方式，并在加工期间按需转换。如此便可在运动时在工件外或粗加工时引入高阻尼滤波器 (Blackman) 和很高的轴专用急动度 (MD32431 \$MA_MAX_AX_JERK, MD32432 \$MA_PATH_TRANS_JERK_LIM)，以此显著缩短加工时间。相对较高的轮廓偏离，结合高效的滤波器，在这些加工方式中不会产生很大的不良影响。

有效的是，将滤波器切换与动态响应 G 代码组 DYNNORM、DYNROUGH、DYNFINISH 和 DYNPREC 进行耦合，并将其连入一个循环（如 CUST_832）。滤波器切换的状态和数值可以通过以下系统变量进行查询。

- \$VA_DESVAL_FILTERS_ACTIVE；设定值滤波器链条的状态
- \$VA_DESVAL_FILTERS_DELAY_1/2；第 1/2 条设定值滤波器链条的有效延迟时间
- \$VA_DESVAL_FILTERS_DELTA_POS；设定值滤波器链条之间的位置差异。

选件

为了能够在**轴运动时**使用该功能，需要以下选件的一个许可：

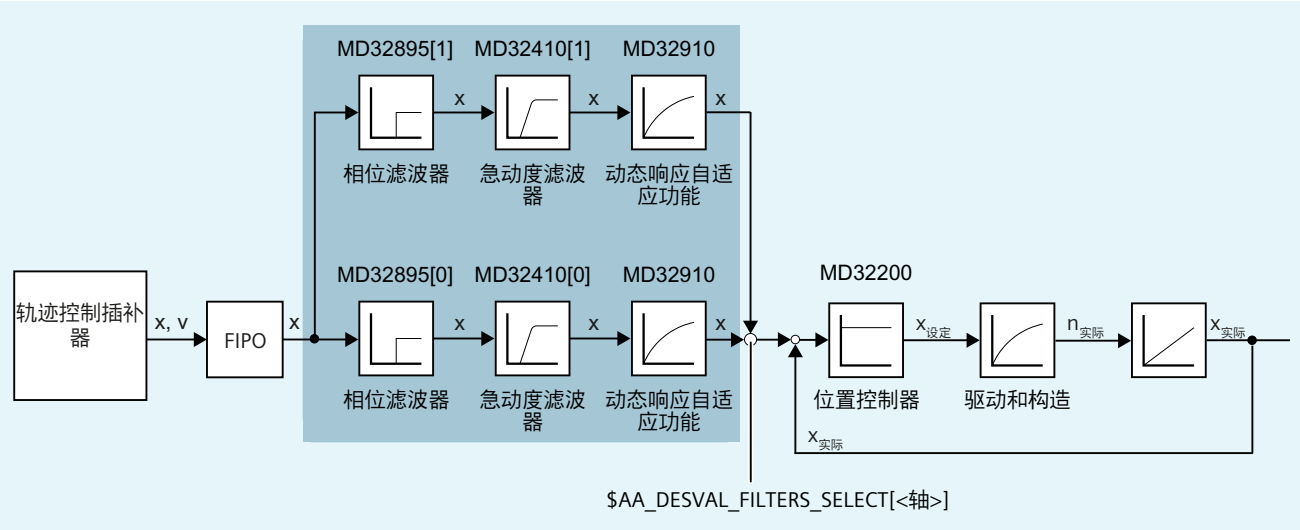
“急动度调整” (6FC5800-0xS22-0Yx0)

在**轴静止时**，没有选件也可以切换位置设定值滤波链。

6.5 闭环控制

功能

通过可切换的滤波器链条实现的进给轴/主轴的位置闭环控制



MD32410	\$MA_AX_JERK_TIME
MD32910	\$MA_DYN_MATCH_TIME
MD32895	\$MA_DESVAL_DELAY_TIME
MD32200	\$MA_POSCTRL_GAIN
x	位置设定值
v	速度设定值
x _{实际}	位置实际值
n _{实际}	转速实际值
n _{设定}	转速设定值

每个滤波器链条的滤波器类型选择和分配

可切换的位置设定值滤波器链条的激活可通过输入机床数据 MD32402 \$MA_AX_JERK_MODE（用于轴向最大急动度的滤波器类型）中的两位数字来实现。如此便可选择两个替代的急动度滤波器。

参数设置

为了能够在零件程序之外在两个滤波器链条间切换，必须选择两个滤波器类型：

- 第一个滤波器链条的急动度滤波器的滤波器类型在机床数据 MD32402 \$MA_AX_JERK_MODE 中的“个位”中进行选择，相应的滤波器设置则在机床数据的下标 0 中进行滤波器参数化。
- 第二个滤波器链条的急动度滤波器的滤波器类型在机床数据 MD32402 \$MA_AX_JERK_MODE 中的“十位”中进行选择，相应的滤波器设置则在机床数据的下标 1 中进行滤波器参数化。
- 下列机床数据有两个标记：
 - MD32402 \$MA_AX_JERK_FIR_FREQ
 - MD32408 \$MA_AX_JERK_FIR_ORDER
 - MD32409 \$MA_AX_JERK_FIR_WINDOW
 - MD32410 \$MA_AX_JERK_TIME
 - MD32411 \$MA_AX_JERK_TIME_ADD
 - MD32412 \$MA_AX_JERK_FREQ
 - MD32414 \$MA_AX_JERK_DAMP

边界条件

滤波器链条只有在满足下列条件时才可调整参数：

- MD33000 \$MA_FIPO_TYPE = 1 oder 3
- MD32220 \$MA_POSCTRL_INTEGR_ENABLE = 0
- MD32930 \$MA_POSCTRL_OUT_FILTER_ENABLE = 0
- MD32610 \$MA_VELO_FFW_WEIGHT = 1.0
- MD32620 \$MA_FFW_MODE = 0, 3 oder 4

调试

为预先分配滤波器链条的滤波器类型，选择两个滤波器类型。系统变量 \$AA_DESVAL_FILTERS_SELECT[<轴>] 用于切换已参数设置的滤波器链条。

6.5 闭环控制

可使用以下系统变量来询问滤波器链条状态：

系统变量	含义
\$VA_DESVAL_FILTERS_ACTIVE	当前生效滤波器链条
\$VA_DESVAL_FILTERS_DELTA_POS	滤波器链条之间的当前位置差异 用于确定合适的切换时间。合适的切换时间是具有恒定速度（加速度 = 0）的运动过程或位置差异很小的轴停滞。若在加速过程中切换，可能会造成一个加速跳跃，有时会引发轮廓监控。请求发送后立刻切换。
\$VA_DESVAL_FILTERS_DELAY_1	第一条滤波器链条的有效延迟时间
\$VA_DESVAL_FILTERS_DELAY_2	第二条滤波器链条的有效延迟时间

示例

通过 MD32402 \$MA_AX_JERK_MODE = 25，两个可切换的位置设定值滤波器链条可使用下列特性参数化：

- 若将值设为 5，则在第一个位置设定值滤波器链条中参数化了一个位置设定值滤波器（滤波器类型 5）=FIR 低通。
- 若将值设为 20，则在第二个位置设定值滤波器链条中参数化了一个位置设定值滤波器（滤波器类型 2）=滑动求平均值。
- 为了统一两个滤波器链条的总延迟时间，要使用 MD32895 \$MA_DESVAL_DELAY_TIME。

参见

急动度滤波器（位置设定值滤波器，轴专用）(页 118)

6.5.3 位置设定值滤波器：限制滤波器长度

系统内部将位置设定值滤波器的时间常数限制为以下最大值：

时间常数	最大值（位置控制器最大周期数）
MD32408 \$MA_AX_JERK_FIR_ORDER	200
MD32410 \$MA_AX_JERK_TIME	128
MD32411 \$MA_AX_JERK_TIME_ADD	128
MD32895 \$MA_DESVAL_DELAY_TIME	128
MD32910 \$MA_DYN_MATCH_TIME	16

如果在设置位置设定值滤波器时超过最大值，会发布报警 26033 并显示内部极限值相关信息。

然后，机床制造商可以修正时间常数值，达到优化滤波器设置的目的。

6.5.4 位置控制器的参数组

每根机床轴有六个参数组可用，以便根据运行期间机床条件的变化（比如：主轴换档）迅速调整位置控制，或以便根据攻丝时另一根轴的条件调整动态响应。

机床数据

一个参数组包括下列机床数据：

编号	名称 \$MA_	含义
31050	DRIVE_AX_RATIO_DENOM	负载齿轮比分母
31060	DRIVE_AX_RATIO_NUMERA	负载齿轮比分子
32200	POSCTRL_GAIN	K _V 系数
32452	BACKLASH_FACTOR	反向间隙补偿
32610	VELO_FFW_WEIGHT	前馈控制系数
36012	STOP_LIMIT_FACTOR	粗准停/精准停和停机系数
32800	EQUIV_CURRCTRL_TIME	等效时间常数前馈控制的电流控制回路
32810	EQUIV_SPEEDCTRL_TIME	等效时间常数前馈控制的转速控制回路
32910	DYN_MATCH_TIME	动态响应自适应的时间常数
36200	AX_VELO_LIMIT	速度监控阈值

攻丝/螺纹切削

下列轴参数组适用于攻丝/螺纹切削：

- 对于未参与攻丝或螺纹切削的机床轴，始终是参数组 1（下标为 0）有效。其他参数组不必考虑。
- 对于参与攻丝或螺纹切削的机床轴，参数组号等于当前主轴齿轮档的参数组有效。
所有等于主轴齿轮档的参数组必须进行参数设置。

操作界面上会显示当前生效的参数组：

操作区“诊断”>“服务轴”

6.6 控制优化

切换齿轮档时的参数组

给主轴的每个齿轮档分配一个合适的参数组。齿轮档是通过以下 NC/PLC 接口信号选择的：

```
<Axis>.spindle.out.gearStageSelection = <实际齿轮档>
```

实际齿轮档	<Axis>.spindle.out.gearStageSelection	参数组	
第 1 齿轮档	000	2	(下标 1)
第 1 齿轮档	001	2	(下标 1)
第 2 齿轮档	010	3	(下标 2)
第 3 齿轮档	011	4	(下标 3)
第 4 齿轮档	100	5	(下标 4)
第 5 齿轮档	101 110 111	6	(下标 5)

有关主轴齿轮档的更多信息请参见章节“主轴 (页 395)”。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.spindle.out.gearStageSelection	LBP_ModeGroup.A_ActGearA..C	DB31, ...DBX16.0.. 2

6.6 控制优化

6.6.1 位置控制器，位置设定值滤波器：对称滤波器

功能

前馈控制激活时，系统会在位置设定值到达真正的控制器前通过一个所谓的对称滤波器将其传送出去。这样就可以将转速设定值预先控制在 100 %，且在定位时不会出现振荡。

激活

通过修改以下轴专用机床数据来激活滤波器和选择前馈控制方式：

MD32620 \$MA_FFW_MODE（前馈控制方式）

值	含义
3	转速前馈控制
4	组合式转矩/转速前馈控制

通过零件程序激活/取消

可通过零件程序中的指令 FFWON 和 FFWOF 来激活或取消所有轴的前馈控制。

如果无法通过 FFWON/FFWOF 指令激活或取消单个轴的前馈控制，则必须对以下机床数据中的设置进行调整：

MD32630 \$MA_FFW_ACTIVATION_MODE（从程序激活前馈控制）

参数设置

首次调试时的推荐设置

在首次调试时或已事先载入了缺省值（调试开关上的开关位置 1 和 POWER ON 指令），则采用以下机床数据设置：

- MD32620 \$MA_FFW_MODE = 3
- MD32610 \$MA_VELO_FFW_WEIGHT（转速前馈控制系数）= 1

然后还须在以下机床数据中调整转速前馈控制的对称时间：

MD32810 \$MA_EQUIV_SPEEDCTRL_TIME（用于前馈控制的转速环等效时间常数）

转速控制回路的等效时间常数设置 (MD32810)

建议在 AUTO 模式下来回移动轴，使轴到达目标位置，即通过伺服跟踪功能观察当前生效的测量系统的位置实际值。

设置的初始值是转速控制回路的时间常数，可从转速控制回路的参考频率响应中读取。在常见的带转速设定值平滑的 PI 控制器中，也可从驱动机床数据 p1414、p1415、p1416 和 p1421 中读取近似等效时间。

现在输入该初始值（如 1.5 ms）：

MD32810 \$MA_EQUIV_SPEEDCTRL_TIME = 0.0015

然后来回移动轴并观察放大后朝目标位置运行时的位置实际值曲线。

6.6 控制优化

手动精调时须遵循以下规定：

监控	措施	影响
超调	提高 MD32810	位置控制器响应缓慢。 ⇒ 超调的倾斜度降低。 问题：曲率上的轮廓误差（与所编程的轮廓偏差）增大。
逼近过慢	降低 MD32810	位置控制器响应过快。 ⇒ 曲率上的轮廓误差减小。 问题：超调的倾斜度增加。

说明

影响与改变位置控制器增益（MD32200 \$MA_POSCTRL_GAIN）的效果类似。可在操作界面上的操作区域“诊断”>“轴信息”中基于计算出的 K_v 值进行观察。

精调 MD32810

因此，MD32810 的值应尽可能小，这样超调在定位时就可以设置限值。为此，在精调时通常只需对初始值进行微调，典型做法是向上或向下调整 0.25 毫秒。

例如：如果初始值为 1.5 毫秒，手动调整的最佳范围通常在 1.25 毫秒到 1.75 毫秒之间。

在装配了直接测量系统（负载编码器）和具备充足弹性的轴上，微米级的超调是可接受的。可借助用于动态响应自适应功能 (MD32910 \$MA_DYN_MATCH_TIME) 和急动度 (MD32410 \$MA_AX_JERK_TIME) 的位置设定值滤波器来降低超调，这会再次减缓轴的速度。

插补组中的相同轴数据

插补组中的所有轴在以下数据中的**设置应是相同的**：

- MD32200 \$MA_POSCTRL_GAIN（通过 MD32910 调整）
- MD32620 \$MA_FFW_MODE
- MD32610 \$MA_VELO_FFW_WEIGHT
- MD32810 \$MA_EQUIV_SPEEDCTRL_TIME 或 MD32800 \$MA_EQUIV_CURRCTRL_TIME（取决于机械系统和驱动器）
- MD32400 \$MA_AX_JERK_ENABLE
- MD32402 \$MA_AX_JERK_MODE
- MD32410 \$MA_AX_JERK_TIME

可通过操作界面上操作区域“诊断”>“轴信息”所显示的 K_v 值进行检查。

插补组中的不同轴数据

如果上述轴数据中包含了不同的值，则可以借助以下机床数据进行调整：

MD32910 \$MA_DYN_MATCH_TIME（动态响应自适应的时间常数）

这样，界面上就会显示相同的 K_v 值。

通常，不同的 K_v 显示值通常表示：

- 一根或多根轴中的齿轮级系数不匹配。
- 前馈控制的设定数据不一致。

电流控制回路的等效时间常数设置 (MD32800)

用于转矩前馈控制的滤波器是通过以下机床数据激活的：

MD32620 \$MA_FFW_MODE = 4

和转速前馈控制中的设置一样，电流控制回路的时间常数的设置 MD32800 \$MA_EQUIV_CURRCTRL_TIME 也须遵循此类规定和建议。

刚性机床上的限制

经验表明，只要在刚性极高的机床上才需要进行此设置，而且要求相应的设置经验。通常来说，接入转矩会提高机床弹性，虽然保持了轮廓精度，但也产生了振动。

在该情况下，建议使用动态刚性控制 Dynamic Stiffness Control (DSC) 进行验证：

MD32640 \$MA_STIFFNESS_CONTROL_ENABLE = 1

说明

使用 DSC 时需要满足以下条件：

- 不可以在 NC 中设置实际值取反 (MD32110 \$MA_ENC_FEEDBACK_POL = -1)。允许在驱动中设置实际值取反 (SINAMICS 参数 p0410)。
- 必须激活所有插补相关的（轨迹运动或耦合）轴的该项功能 (MD32640 = 1)。
- 如果轴编程为模拟轴 (MD30130 \$MA_CTRLOUT_TYPE = 0, MD30240 \$MA_ENC_TYPE = 0)，则无法使用 DSC。在此情况下，必须取消激活所有插补相关的轴的该项功能 (MD32640 = 0)。

 小心
特殊故障状态 当为不存在的测量系统设置 p1192 并且 DSC 激活时，就会出现特殊故障状态！ 维修时如要取消附加测量系统并且 DSC 激活，则应确保对 p1192 也进行相应的调整。

6.6 控制优化

POWER ON、RESET、REPOS 等指令时的控制特性

在执行 POWER ON 和 RESET 以及“设置机床数据生效”指令时，系统会重新读入前馈控制的设定数据（参见各个机床数据的相关说明）。

运行方式切换、程序段搜索和 REPOS 对前馈控制无效。

6.6.2 位置控制器，位置设定值滤波器：相位滤波器

功能

轴设定值相位滤波器（时滞/延时）可实现一个移相器的功能，借助其可调整设定值相位特性曲线。配合轴设定值急动滤波器 MD32402_\$MA_AX_JERK_MODE[<轴>] = 2 (页 118)，可相互独立地调整振幅响应和相位响应，以适应共同走出一个编程的轨迹的多根轴中动态特性最弱的那根。

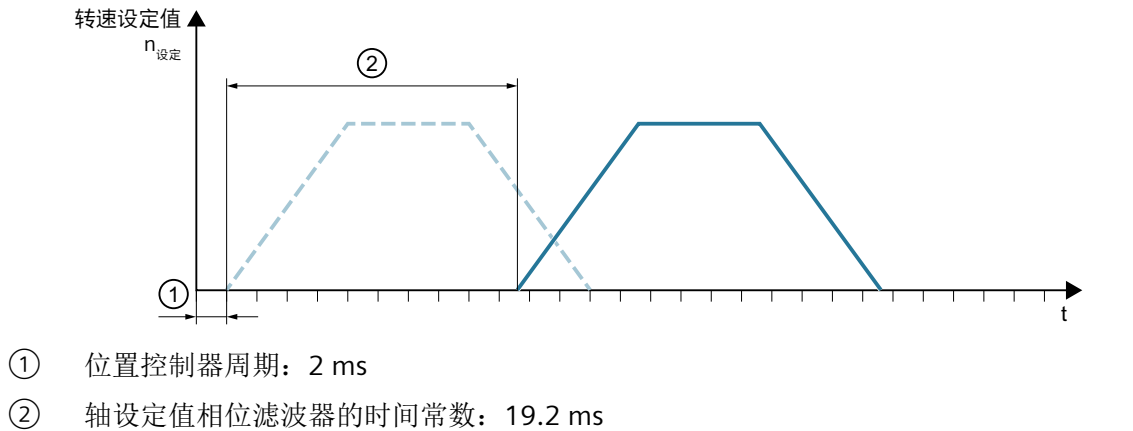


图 6-10 轴设定值相位滤波器的作用

参数设置：时间常数

轴设定值相位滤波器的时间常数的设置范围如下：

(0 至 128) * 位置控制器周期 (MD10061 \$MN_POSCTRL_CYCLE_TIME)

必须以秒 [s] 为单位将时间常数值输入以下机床数据：

MD32895 \$MA_DESVAL_DELAY_TIME[<轴>]

示例：位置控制器周期：2 ms $\Rightarrow$ 可设置的时间常数：0.0 至 0.256 s

说明

设定值相位滤波器的时间常数会延缓轴的响应特性，例如在攻丝、回退或准停/程序段转换时。因此建议将时间常数设置地尽可能小。

限制为最大值

当所设置的时间常数大于所允许的最大值（128 * 位置控制器周期）时，系统内部会将其限制为最大值。

$\rightarrow$ 更多信息：“位置设定值滤波器：限制滤波器长度 (页 188)”章节

参数设置：激活

轴设定值相位滤波器功能必须通过以下机床数据激活：

MD32890 \$MA_DESVAL_DELAY_ENABLE[<轴>] = "TRUE"

示例

假设：位置控制器周期 = 2 ms

1. MD32890 \$MA_DESVAL_DELAY_ENABLE[<轴>] = "FALSE"
MD32895 \$MA_DESVAL_DELAY_TIME[<轴>] = <时间常数>
 - 设定值相位滤波器：未生效
 - 时间常数：没有意义
2. MD32890 \$MA_DESVAL_DELAY_ENABLE[<轴>] = TRUE
MD32895 \$MA_DESVAL_DELAY_TIME[<轴>] = 0.002
 - 设定值相位滤波器：生效
 - 时间常数：2 ms $\Rightarrow$ 设定值输出延后一个位置控制器周期。
3. MD32890 \$MA_DESVAL_DELAY_ENABLE[<轴>] = TRUE
MD32895 \$MA_DESVAL_DELAY_TIME[<轴>] = 0.300
 - 设定值相位滤波器：生效
 - 设置的时间常数：300 ms;
可能的最大时间常数：128 * 2 ms = 256 ms
 $\Rightarrow$ 内部生效的时间常数：256 ms

6.6 控制优化

边界条件

SINUMERIK Safety Integrated

设定值相位滤波器会延后轴设定值的输出，例如在回退运行中（STOP E/SS2ESR 时 ESR）。但设定值相位滤波器不会影响取消进程，例如 SOS 的激活时间点。

6.6.3 位置控制器：位置差接通

前提条件

- 该功能只在带有两个编码器的轴上可用：
MD30200 \$MA_NUM_ENC = 2
其中一个编码器必须配置为间接测量系统，另一个则须配置为直接测量系统：
 - 直接测量系统：
MD31040 \$MA_ENC_IS_DIRECT[1]=1
用于实际值测量的编码器直接安装在机床上（负载编码器）。
 - 间接测量系统：
MD31040 \$MA_ENC_IS_DIRECT[0]=0
用于实际值测量的编码器安装在电机上（电机编码器）。
- 作为 PROFIdrive 的标准报文类型，必须在驱动或 NC (MD13060 \$MN_DRIVE_TELEGRAM_TYPE) 中配置为报文类型 136 或 138。

功能

位置差接通功能激活时，系统会测定一根轴的直接测量系统与间接测量系统之间的差值位置并根据加权系数的设置接通该差值位置，将其作为位置控制器周期中前馈控制的附加电流设定值。由此产生的振动阻尼有助于提升轴的稳定性和定位精度。

应用

该功能用于带有较高振动倾向的轴。

有效性

该功能仅在带有低固有频率（最大约 20 Hz）的轴上生效。

激活/设置

通过给定加权系数激活该功能：

MD32950 \$MA_POSCTRL_DAMPING（转速控制回路衰减）= <值>

取值范围：-100% ... +100%

输入值“100%”表示：如果测出的两个测量系统之间的位置差达到以下值，系统则会根据 SINAMICS 参数 p2003 接通附加转矩：

- 直线电机：1 毫米
- 带旋转电机的线性轴：MD31030 \$MA_LEADSCREW_PITCH（丝杠螺距）
- 回转轴/主轴：360 度

缺省设置为 0。此时，位置差接通功能无效。

说明

比如：可借助阶跃函数响应设置加权系数 MD32950 \$MA_POSCTRL_DAMPING。
如果闭环控制达到稳定限值（振动倾向增强），则表明参数值设置的过大或该值符号错误。

6.6.4 带比例积分控制器的位置闭环控制

功能

标准情况下，位置控制器的核心是比例控制器。针对某些特殊应用（如电子齿轮箱），可以激活一个积分分量。在设置适宜时，比例积分控制器会根据相应的机床数据在有限、可设置的时间范围内将设定位置 and 实际位置之间的误差修正为零。

 小心

比例积分控制器激活时，实际位置会出现超调

此时，用户必须确定此类影响是否是允许或可接受的。只有具备了控制方面的专业知识以及了解如何使用伺服跟踪功能进行测量才可以使用该功能。如果相关机床数据设置错误，机床可能会因不稳定而损坏。

6.6 控制优化

步骤

1. 按照之前章节中描述的方法先将位置控制回路作为比例控制器进行优化。
2. 在测量期间提高以下机床数据的公差，以便确定比例积分控制器的效果：
 - MD36020 \$MA_POSITIONING_TIME
 - MD36030 \$MA_STANDSTILL_POS_TOL
 - MD36040 \$MA_STANDSTILL_DELAY_TIME
 - MD36400 \$MA_CONTOUR_TOL
3. 通过设置以下机床数据将位置控制回路作为比例积分控制器激活：
 - MD32220 \$MA_POSCTRL_INTEGR_ENABLE；值设为 1
 - MD32210 \$MA_POSCTRL_INTEGR_TIME；积分时间 [s]
 积分时间的有效性：
 - $T_n \rightarrow 0$ ：
控制误差会被快速修正，但控制回路可能会变得不稳定。
 - $T_n \rightarrow \infty$ ：
积分部件的有效性不为 0。控制器的特性等同于纯比例控制器。
4. 应在这两个极端情况之间寻求一个可达到平衡效果的 T_n 。

说明

不可以选择一个非常接近稳定限值的 T_n ，否则，机床会因不稳定而损坏。
因此建议将 T_n 设置为不小于 1 秒的值。

通过一个自动执行的程序来回移动轴，使轴到达目标位置，用伺服跟踪功能记录该过程。

5. 伺服跟踪功能中会显示：
 - 跟随误差
 - 实际速度
 - 实际位置
 - 设定位置
6. 一旦查找出 T_n 的最佳值，则应将以下机床数据中的公差值再次复位为所需值。
 - MD36020 \$MA_POSITIONING_TIME
 - MD36030 \$MA_STANDSTILL_POS_TOL
 - MD36040 \$MA_STANDSTILL_DELAY_TIME
 - MD36400 \$MA_CONTOUR_TOL

前提条件

DSC

使用积分器功能时，必须取消 DSC（Dynamic Stiffness Control，动态刚度控制）。

示例

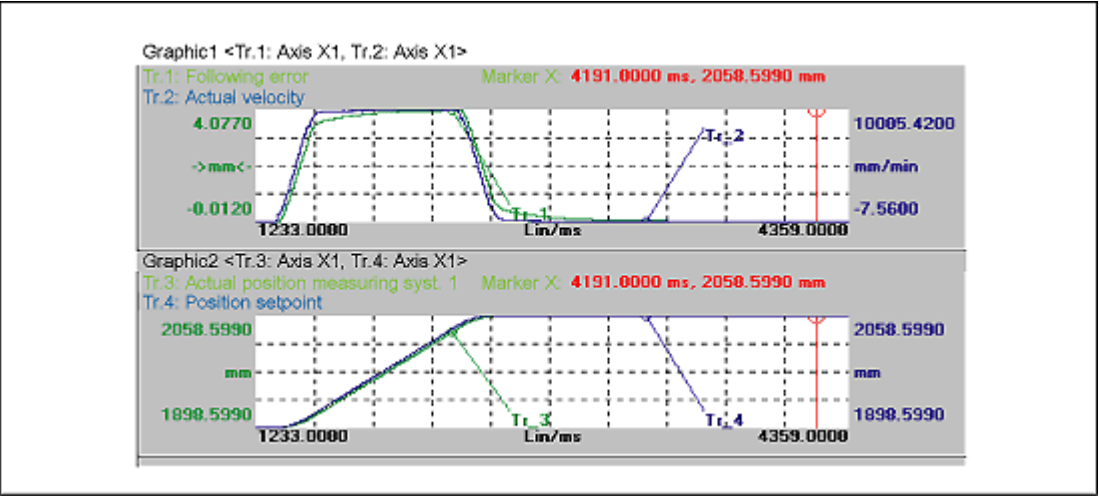
多次重复过程后的 K_R 和 T_n 的设置结果

机床数据设置：

- MD32220 \$MA_POSCTRL_INTEGR_ENABLE = 1
- MD32210 \$MA_POSCTRL_INTEGR_TIME = 0.003
- MD32200 \$MA_POSCTRL_GAIN[1] = 5.0

选择参数组 0

伺服功能会分别记录跟随误差、实际速度、位置实际值和位置设定值。在 JOG 模式中运行时，还会记录下图中显示的几个数据的曲线。



6.6 控制优化

进给率

7.1 简要说明

进给方式

进给率用于确定加工速度（轴进给率或轨迹进给率），每一种插补方式下的轮廓或刀具中心点轨迹都会遵循特定的进给率（取决于 G 指令），在启用刀具补偿时也是如此。

为了实现对各种工艺应用（车削、铣削、钻孔等）的最优匹配，可在下列进给类型中进行选择：

- 快进进给率（G0）
- 反比时间进给率（G93）
- 线性进给率（G94）
- 旋转进给率（G95）
- 恒定切削速度（G96、G961）
- 恒定转速（G97、G971）
- 螺纹切削（G33、G34、G35）时的进给率
- 刚性攻丝（G331、G332）中的进给率
- 带补偿夹具攻丝（G63）中的进给率
- 倒角/倒圆进给率 FRC、FRCM
- 逐段进给率 FB

进给率对轴的指定

为了适应各种技术需求，可以多变的形式将进给率应用于轴。

可供采用的方案例如包括：

- 对工作平面和进给轴分别启用不同的进给率
- 轨迹进给中轴的灵活指定
- 定位轴进给

进给控制

为了适应加工期间变化的技术条件以及进行测试，可通过以下方式修改编写的进给率：

- 通过机床控制面板
- 通过操作面板前端
- 通过 PLC
- 通过程序指令

7.2 轨迹进给率 F

轨迹进给率 F

轨迹进给率表示的是相关轴速度分量的几何总和。其通过相互插补的轴的单次运动得到。

缺省设置下使用的是所编写的几何轴的轴向速度。通过 FGROUP 指令也可将其他几何轴和/或同步轴纳入轨迹进给率的计算。

轨迹进给率 F 决定了加工速度，在每种插补方式都须遵循，在启用刀具补偿时也是如此。地址 F 下编写的值在一个程序中一直保留，直至编写新的 F 值或新的进给类型。

进给率 F 的取值范围

参见功能说明“速度，设定值/实际值系统，闭环控制”，章节：“速度 (页 141)”。

PLC 接口上的 F 值

当前轨迹进给率的 F 值会记录至用于辅助功能的通道专用 PLC 接口

(`<Chan>.auxFFuncs[1].in.extendedAddress`
和 `<Chan>.auxFFuncs[1].in.value`)。

对应接口信号（修改信号，F 值）的详细说明参见功能手册之基本功能，“辅助功能向 PLC 的输出”功能说明。

过渡圆弧处的进给率

参见 NC 编程手册。

向内和向外弯曲的轨迹段的进给率

对于圆弧程序段、同向弯曲的样条程序段，以及激活刀具半径补偿（G41/G42）时，编写的进给率可在中心点轨迹或轮廓上生效（取决于向内弯曲和向外弯曲的轨迹段）。

为此，有一个 G 指令组可供使用：

- CFTCP
编写的进给率在中心点轨迹上生效。
- CFC
编写的进给率在轮廓上生效。
- CFCIN
编写的进给率在轮廓上向内弯曲部分生效。

更多信息： NC 编程手册

最大轨迹速度

最大轨迹速度由参与进给的直线轴或回转轴的最大速度（MD32000 \$MA_MAX_AX_VELO）得出，也就是说将几个最大值中最低的一个用作最大轨迹速度。不可超出该速度。

若编写了 G0，则会采用受到 MD32000 \$MA_MAX_AX_VELO 限制的轨迹速度运行。

轨迹轴的极限速度

此外对于轨迹轴（几何轴和同步轴），还可通过 FL[<轴>] 指令为相应轴编写一个极限速度。

这样便可分别为工作平面和进给轴编写进给率。这表示同时为轨迹相关插补和进给轴设定一个进给速度。进给轴指的是垂直于所选加工平面生效的轴。可编写进给轴专用进给率来限制轴速度以及轨迹速度。此时切勿启用通过框架进行的坐标旋转。也就是说进给轴必须是基本坐标系中的轴。此功能例如用于应对以下情形：铣刀在端面上的切削效率小于通过整个铣刀圆周进行切削时的效率。

编程示例：

程序代码	注释
...G94 ...	选择进给类型 (mm/min)
X30 Y20 F200	; 轨迹进给率 = 200 mm/s
FL[Z]=50 Z-30	; Z 轴的最大进给率: 50 mm/s

低分辨率编码器

使用低分辨率的编码器时，可通过平滑实际值来获取更稳定的轨迹运动或轴运动。所设置的时间常量越大，达到的实际值平滑效果也就越佳，但滞后行程也会越大。

7.2 轨迹进给率F

MD34990 \$MA_ENC_ACTVAL_SMOOTH_TIME[<轴>]（实际值的平滑时间常量）

平滑实际值应用于：

- 螺纹切削（G33、G34、G35）
- 旋转进给率（G95、G96、G97、FPRAON）
- 显示转速、实际位置 and 实际速度

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Chan>.auxFFuncs[1].in.extendedAddress	LBP_Chan*.ExtF1 - ExtF6	DB21, ... DBW158, 164, 170, 176, 182, 188
<Chan>.auxFFuncs[1].in.value	LBP_Chan*.F1 - F6	DB21, ... DBD160, 166, 172, 178, 184, 190

7.2.1 进给类型 G93、G94、G95

有效性

进给类型 G93、G94、G95 在 AUTO 运行方式下对 G 指令组 1（除 G0）生效。

在 JOG 模式下可通过 G94 或 G95 运行。

更多信息

手动运行和通过手轮进行手动运行 (页 743)

反比时间进给率（G93）

编写程序段退出时，若编写持续时间比编写进给率更简单，则可使用反比时间进给率。

反比时间进给率通过以下公式计算：

$$F = v / s$$

其中 F: 反比时间进给率, 单位 rpm
 v: 所需的轨迹速度, 单位 mm/min 或 inch/min
 s: 轨迹长度, 单位 mm 或 inch

编程示例

程序代码	注释
N10 G1 G93 X100 Y200 F2	编程的轨迹行程在 0.5 分钟内运行完毕。
...	

说明

G41/G42 生效时不允许使用 G93。如果各程序段的长度差别很大, 那么在启用 G93 时应在每个程序段中编写一个新的 F 值。

线性进给率 (G94)

线性进给率采用以下单位基于直线轴或回转轴编写:

- 采用公制系统时: [mm/min, degrees/min]
- 采用英制系统时: [inch/min, degrees/min]

旋转进给率 (G95)

旋转进给率采用以下单位基于主轴编写:

- 采用公制系统时: [mm/rev]
- 采用英制系统时: [inch/rev]
- 对于回转轴: [degrees/rev]

轨迹速度由主轴的实际转速按以下公式得出:

$$V = n * F$$

其中 v: 轨迹速度, 单位 mm/min 或 inch/min
 n: 主轴转速, 单位 rev/min
 F: 编写的旋转进给率, 单位 mm/rev 或 inch/rev

7.2 轨迹进给率F

说明

若在进给类型 G93、G94、G95 之间切换，那么编写的 F 值将被删除。

每齿进给量

可编写实际应用中更为常用的齿进给值 FZ（每齿进给量）来替代旋转进给值 F，特别是对于铣削加工。

通过生效刀具补偿数据组的刀具参数 \$TC_DPNT（每转齿数），控制系统根据每个运行程序段中编写的每齿进给量计算生效的旋转进给率：

$$F = FZ * \$TC\_DPNT$$

其中 F: 旋转进给率，单位 mm/rev 或 inch/rev

FZ: 每齿进给量，单位 mm/tooth 或 inch/tooth

\$TC_D 刀具参数：齿数/转

PNT:

示例：5 齿铣刀 (\$TC_DPNT = 5)

程序代码	注释
N10 G0 X100 Y50	
N20 G1 G95 FZ=0.02	; 每齿进给率 0.02 毫米/齿
N30 T3 D1	; 切换刀具，并激活刀具补偿数据组。
M40 M3 S200	; 主轴转速 200 转/分钟
N50 X20	; 以 FZ = 0.02 毫米/齿铣削
	; 生效的旋转进给率：
	; F = 0.02 毫米/齿 * 5 齿/转 = 0.1 毫米/转
	; 或者
	; F = 0.1 毫米/转 * 200 转/分钟 = 20 毫米/分钟
...	

设定数据

在 JOG 运行方式下启用旋转进给率

JOG 运行方式下，关系到通道（轴目前分配于该通道）主主轴的、与旋转进给率相关的轴的特性取决于 NC 特定设定数据的设置：

SD41100 \$SN_JOG_REV_IS_ACTIVE, 位 <x>

位	值	含义
0	0	该特性取决于其他设定数据，取决于轴的功能： - 进给轴 / 主轴:SD43300 \$SA_ASSIGN_FEED_PER_REV_SOURCE - 带旋转框架的几何轴: SD42600 \$SC_JOG_FEED_PER_REV_SOURCE - 定向轴: SD42600 \$SC_JOG_FEED_PER_REV_SOURCE
	1	以旋转进给率运行轴。 边界条件 如果满足以下条件，则不以旋转进给率运行： - 进给轴 / 主轴:主主轴静止 且 SD43300 \$SA_ASSIGN_FEED_PER_REV_SOURCE == -3 - 带旋转框架的几何轴: 主主轴静止 且 SD42600 \$SC_JOG_FEED_PER_REV_SOURCE == -3 - 定向轴: 主主轴静止 且 SD42600 \$SC_JOG_FEED_PER_REV_SOURCE == -3
1	0	在 急速运行模式 中，以旋转进给率运行。
	1	在 急速运行模式 中，不以旋转进给率运行。
2 ¹⁾	0	在 JOG 方式下手轮运行 时，以旋转进给率运行。
	1	在 JOG 方式下手轮运行 时，不以旋转进给率运行。
3 ¹⁾	0	在 DRF 手轮运行 时，以旋转进给率运行。
	1	在 DRF 手轮运行 时，不以旋转进给率运行。
1) 该功能可额外用于设置位 1。		

NC/PLC 接口信号

- 旋转进给有效（**通道专用的**）

通过接口信号显示通道中的轴是以旋转进给率运行的：

- 运行方式 AUTOMATIC: 轨迹轴或同步轴
- 运行方式 JOG: 几何轴

<Chan>.basic.in.revolutionalFeedrateActive == 1（旋转进给有效）

- 旋转进给有效（**轴专用的**）

通过接口信号显示轴是以旋转进给率运行的：

<Axis>.basic.in.revolutionalFeedrateActive == 1（旋转进给有效）

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Chan>.basic.in.revolutionalFeedrateActive	LBP_Chan*.E_RevFD	DB21, ...DBX33.2
<Axis>.basic.in.revolutionalFeedrateActive	LBP_Axis*.E_RevFD	DB31, ...DBX62.2

7.2.2 进给类型 G96、G961、G962、G97、G971

恒定切削速度（G96、G961）

恒定切削速度在车床上使用，目的是独立于工件的工作半径保持恒定的切削条件。这样可保证刀具始终在最佳的切削效率范围运行，从而延长其寿命。

使用 G96、G961

编写 G96、G961 时，对应的 S 值会作为切削速度在平面轴中被解译为 m/min 或 ft/min。加工期间平面轴中的工作半径减小时，转速将一直提升，直至到达恒定的切削速度。

在零件程序中首次选择 G96、G961 时必须以 m/min 或 ft/min 输入一个恒定切削速度，重新选择时则可选择输入方法。

启用 G96 时，控制系统会自动切换至旋转进给率（和 G95 下相同），也就是说，编写的进给率 F 会被解译为 mm/rev 或 inch/rev。

在编写 G961 时会自动选择线性进给率（和 G94 下相同）。编写的进给率 F 会被解译为 mm/min 或 inch/min。

确定主轴转速

基于编写的切削速度（S_{G96} 或 S_{G961}）以及当前 TCP 上平面轴（半径）的坐标位置，控制系统按照以下公式确定主轴转速：

$$n = \frac{S_{Speed}}{2 * \pi * r}$$

- n: 主轴转速
- S_{Speed}: 编写的切削速度
- π 圆弧常数
- r: 半径（旋转中心与 TCP 的距离）

为了确定半径，此时会假定下列条件：

- WCS 中的平面轴位置 0 为旋转中心。
- 位置补偿（例如在线刀具补偿、外部零点偏移、\$AA_OFF、DRF 偏移）和耦合产生的位置分量（例如 TRAIL 下的跟随轴）在确定半径时不会被考虑。

框架（例如可编程框架 SCALE、TRANS 或 ROT）会在确定主轴转速时被考虑在内，且会引起转速变化，前提条件是 TCP 上的生效直径变化。

为通道中的多根平面轴编写直径和参考轴：

可同时或分别激活多根平面轴。

- 编程和显示在操作界面中通过直径进行。
- 设定的参考轴通过 SCC [<Achse>] 指定给恒定切削速度 G96、G961、G962

更多信息参见功能说明“端面轴 (页 267)”。

示例

$$S_{G96} = 230 \text{ m/min}$$

- 若 $r = 0.2 \text{ m} \rightarrow n = 183.12 \text{ rev/min}$
- 若 $r = 0.1 \text{ m} \rightarrow n = 366.24 \text{ rev/min}$

⇒ 工件直径越小，转速就越高。

必须为 G96、G961 或 G962 将一根几何轴定义为平面轴。

平面轴的位置会影响主轴转速，其通过以下通道专用机床数据定义：

MD20100 \$MC_DIAMETER_AX_DEF（带平面轴功能的几何轴）

G96、G961 或 G962 功能的前提条件是：机床零点或平面轴的工件零点位于主轴的旋转中心。

恒定转速（G97、G971）

启用 G97、G971 功能时，“恒定切削速度”（G96、G961）功能会被取消，最后计算的主轴转速会被保存。G97 下进给率会被解译为旋转进给率（同 G95）。在编写 G971 时会选择线性进给率（和 G94 下相同）。进给率 F 会被解译为 mm/min 或 inch/min。

G97、G971 生效时，可通过重新编写一个 S 值设定一个新的主轴转速。G96、G961 下编写的切削速度将不受其影响。

7.2 轨迹进给率 F

通过 G97、G971 可在无加工的情况下避免平面轴中运动时的转速变化。

说明

在工件上进行加工时（G1、G2、G3、样条插补等进给率 F 生效的过程），G96、G961 才生效。

G96、G961 以及 G0 程序段生效时主轴转速的特性可在以下通道专用机床数据中设置：

MD20750 ALLOW_G0_IN_G96（G96、G961 下的 G0 逻辑）

采用恒定切削速度 G96、G961 时不会发生齿轮档切换。

主轴倍率开关对确定的主轴转速生效。

平面轴中的 DRF 偏移不会影响主轴转速设定值计算。

再加工开始时（G0 后）时和 NC 停止、G60、G09 后，系统会等待轨迹启动“转速实际值 = 转速设定值”。

接口信号“转速实际值 = 转速设定值”和“设定转速已限制”不会因为内部转速设定而被修改。

在低于最低转速，或识别出“进给轴/主轴停止”信号时，“转速实际值 = 转速设定值”会被复位。

已开始的轨迹加工（G64，精磨）不会被中断。

G96、G961 下的转速限制

可为“恒定切削速度”功能设定一个最大主轴转速。

- 设定数据中：
SD43230 \$SA_SPIND_MAX_VELO_LIMS（启用 G96/G961 时的主轴转速限制）
- 在零件程序中（为主主轴）通过编程指令 LIMS

最后修改的值（LIMS 或 SD）将生效。

LIMS 适用于 G96、G961、G97，且可在一个零件程序的一个程序段中设定最多四个转速限制。为此可在零件程序指令 LIMS[<Sn>] 中通过主轴编号 <Sn> = 1、2、3 或 4 设定相应的主主轴。

当程序段进入主处理时，编写的所有值都会被接收至设定数据 SD43230 \$SA_SPIND_MAX_VELO_LIMS。

根据以下机床数据：

MD10710 \$MN_PROG_SD_RESET_SAVE_TAB[n]（待更新设定数据）

写入的包含 LIMS 的转速限制在控制系统断电后继续保存。

重新激活 G96、G961、G97 时该转速限制重新生效。

不允许超出通过 G26 或通过以下设定数据设定的主轴转速限值：

SD43220 \$SA_SPIND_MAX_VELO_G26（最大主轴转速）

编程错误导致超出转速限值（G26 或 SD43220 \$SA_SPIND_MAX_VELO_G26）时，以下 NC/PLC 接口信号将置位：

<Axis>.spindle.in.speedSetpointAboveMax（编程的转速过高）

为了确保在零件直径较大时能完成回转，转速不得低于定义的最小主轴转速。

该转速可通过以下设定数据设置：

SD43210 \$SA_SPIND_MIN_VELO_G25（最小主轴转速）

以及针对齿轮档通过以下机床数据设置：

MD35140 \$MA_GEAR_STEP_MIN_VELO_LIMIT（齿轮档的最小转速）

最小主轴转速可以在零件程序中用 G25 进行更改。编程低于转速限值（G25 或 SD43210 \$SA_SPIND_MIN_VELO_G25）时，以下 NC/PLC 接口信号将置位：

<Axis>.spindle.in.speedSetpointBelowMin（设定转速过低）

主轴转速限制的其它信息参见功能描述 S1：“主轴”，章节：“主轴 (页 395)”。

说明

零件程序中通过 G25/G26/LIMS 修改的转速限值会纳入设定数据并在程序结束后仍然保留。如果不愿在程序结束之后采用通过 G25/G26/LIMS 修改的转速限值，则必须在机床制造商 GUD 模块中增加以下定义：

```
REDEF $SA_SPIND_MIN_VELO_G25 PRLOC
REDEF $SA_SPIND_MAX_VELO_G26 PRLOC
REDEF $SA_SPIND_MAX_VELO_LIMS PRLOC
```

G96、G961 下的主轴切换

在 G96、G961 生效的情形下切换主轴时，原主轴的转速将保留。这相当于从 G96 过渡至 G97。通过 SETMS 重新定义的主轴将执行如此生成的“恒定切削速度”功能。

报警

恒定切削速度 G96、G961、G962

- 若未编写 F 值，则会输出报警 10860“未编写进给率”。G0 程序段不会生成该报警。
- 编写了负向的轨迹速度时，会触发报警 14800“编写的轨迹速度小于等于零”。

7.2 轨迹进给率F

- 若 G96、G961、G962 生效时未通过以下机床数据定义平面轴：
MD20100 \$MC_DIAMETER_AX_DEF（带平面轴功能的几何轴）
则会触发报警 10870“未定义平面轴”。
- G96、G961 生效时，若通过编程指令 LIMS 为最大主轴转速编写了负值，则会输出报警 14820“为 G96、G961 编写的最大主轴转速为负值”。
- 若首次选择 G96、G961 时未编写恒定切削速度，则会输出报警 10900“没有为恒定切削速度编写 S 值”。

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.spindle.in.speedSetpointAboveMax	LBP_Axis*.E_SetSpeedLimit	DB31, ...DBX83.1
<Axis>.spindle.in.speedSetpointBelowMin	LBP_Axis*.E_SetSpeedIncreased	DB31, ...DBX83.2

7.2.3 螺纹切削（G33, G34, G35, G335, G336）中的进给率

7.2.3.1 G33 中的进给率

G33

使用 G33 功能可以加工螺距恒定的螺纹。

转速 S、进给率 F、螺距

G33 螺纹加工中启用一个旋转进给率 [mm/rev]，其通过螺距 [mm/rev] 编程来设定。

螺纹长度所对应的轴速度通过编写的主轴转速和螺距计算：

进给率 F [mm/min] = 转速 S [rev/min] * 螺距 [mm/rev]

在加速斜坡末端，系统将建立主轴位置实际值（启用 SPCON 时主主轴的设定位置）与轴位置设定值的位置耦合。从此时轴相对主轴零脉冲（零脉冲偏移已计算在内）的位置看来，就像轴在程序段开始处越过螺纹起始位置（零脉冲 + SF）时进行了急剧加速一样。此时轴的跟随误差被补偿。

最小主轴转速

为了确保能在较低的转速下完成回转，不可低于最小主轴转速。

该限值可通过

- 设定数据设置：
SD43210 \$SA_SPIND_MIN_VELO_G25（最小主轴转速）
- 或针对齿轮档通过以下机床数据设置：
MD35140 \$MA_GEAR_STEP_MIN_VELO_LIMIT（齿轮档切换的最小转速）

最小主轴转速可以在零件程序中用 G25 进行更改。

NC 停止、单程序段

NC 停止和单程序段模式直到螺纹链末尾才生效。所有相连的 G33 程序段及后面跟随的第一个非 G33 程序段会像一个程序段一样被连续执行。

提前无损终止

在螺纹切削加工到达终点前，可实现不损坏螺纹的中断。这可通过激活回退运动来实现。

含 ROT 框架的螺纹切削

在 G33、G34、G35 螺纹切削中启用 ROT 框架时，若该框架会导致螺纹长度乃至螺距变化，则会输出报警 10607“无法执行含框架螺纹”。允许启用围绕螺纹轴的旋转。

若在应用中一定要使用 ROT 指令，可在机床数据 MD11410 \$MN_SUPPRESS_ALARM_MASK 中设置位 12 来抑制报警 10607“无法执行含框架螺纹”。

所有其他框架均会被 NC 接受，不会触发报警。系统会对 SCALE 引起螺距变化的特性进行提示。

7.2.3.2 G34 和 G35 中线性递增式/递减式的螺距变化

功能

螺距增加量（G34）描述的是螺距增大的数值。螺距越大，工件上螺纹线之间的距离越大。在启用恒定主轴转速的情形下，螺纹轴的速度将提升。

螺距降低量（G35）则是相反的情形。

7.2 轨迹进给率 F

针对螺距变化引入下面两个概念：

- G34：螺距递增变化
- G35：螺距递减变化

G34 和 G35 都会隐性激活 G33，并且提供在 F 下编写螺距变化的功能。如果已知一个螺纹的起始螺距和最终螺距，那么就可以根据下面的等式计算出需要编写的螺距变化：

$$F = \frac{|k_e^2 - k_a^2|}{2 * l_g}$$

含义：

F： 待编写的螺纹螺距变化 [mm/rev²]

k_e ： 螺纹轴轴目标点坐标的螺距 [mm/rev]

k_a ： 螺纹起始螺距 (在 I、J、K 下编程) [mm/rev]

l_g ： 螺纹长度 [mm]

根据需要增加还是降低螺距，相应选择 G34 或 G35 来编写 F 值。

螺纹长度 l_g 、螺纹变化 F、起始螺距 k_a 为已知时，可通过等式换算如下推导出程序段末尾的螺纹螺距 k_e ：

- 启用 G34（螺距增加）时：

$$k_e = \sqrt{k_a^2 + F * 2 * l_g}$$

- 启用 G35（螺距降低）时：

$$k_e = \sqrt{k_a^2 - F * 2 * l_g}$$

说明

若得出的结果是一个负数根值，则无法加工该螺纹！

在此情形下 NC 会发出报警 10605 或报警 22275。

应用

G34 和 G35 功能可用于实现自切螺纹。

示例

螺纹切削 G33，启用递减螺距 G35

程序代码	注释
N1608 M3 S10	； 主轴转速
N1609 G0 G64 Z40 X216	； 逼近起始点
N1610 G33 Z0 K100 SF=R14	； 恒定螺距（100 mm/rev） 螺纹
N1611 G35 Z-220 K100 F17.045455	； 螺距递减 17.045455 mm/rev2
	； 程序段结束处螺距： 50 mm/rev
N1612 G33 Z-240 K50	； 平稳运行螺纹程序段
N1613 G0 X218	
N1614 G0 Z40	
N1616 M17	

程序段预处理期间监控

在程序段预处理中可提前识别出引起螺纹轴过载的 G34 螺距变化和引起轴停止的 G35 螺距变化，并输出相应报警：报警 10604“螺距增加量过高”或报警 10605“螺距降低量过高”。

在特定的实际应用中，可能需要在螺纹切削中进行主轴转速的补偿。在此情形下，操作人员须确保螺纹轴的速度在允许范围内。

为此可采用以下设置抑制报警 10604 和 10605 的输出：

MD11410 \$MN_SUPPRESS_ALARM_MASK 位 10 = 1

之后程序段预处理继续正常进行。

执行期间监控

执行（插补）螺纹程序段期间，对以下情形进行循环监控：

- 超出螺纹轴的最大速度
- G35 中螺纹轴达到停止状态

系统会输出以下报警作为响应：

- 报警 22270“达到螺纹轴的最大速度”，或
- 报警 22275“螺纹轴达到速度零”

7.2.3.3 G33、G34 和 G35 中轴的加速度特性

在通过 G33、G34 或 G35 进行螺纹切削时，进给轴的加速度特性可通过以下通道专用设定数据设置：

7.2 轨迹进给率F

SD42010 \$SC_THREAD_RAMP_DISP[<n>] = <值>

<n>	含义	<值>	含义
0	螺纹导入时的加速度特性	< 0 (缺省设置: -1)	在螺纹导入时, 螺纹轴根据当前的 BRISK / SOFT 编程加速。
		= 0	在螺纹导入时, 螺纹轴跃变式加速 (≡ BRISK)。
		> 0	设定最大导入距离。 也可在零件程序中通过地址 DITS 设定最大导入距离 (参见 “G33、G34、G35 中可编程的导入和导出行程 (DITS、DITE) (页 217)”)。在主处理中切换至该程序段时, 系统将编写的导入距离接收至设定数据。 提示 距离过短可能会导致轴的加速度过载。
1	螺纹导出时的加速度特性	< 0 (缺省设置: -1)	在螺纹导出时, 螺纹轴根据当前的 BRISK / SOFT 编程制动。
		= 0	在螺纹导出时, 螺纹轴跃变式制动 (≡ BRISK)。
		> 0	设定最大导出距离。 也可在零件程序中通过地址 DITE 设定最大导出距离 (参见 “G33、G34、G35 中可编程的导入和导出行程 (DITS、DITE) (页 217)”)。在主处理中切换至该程序段时, 系统将编写的导出距离接收至设定数据。 提示 距离过短可能会导致轴的加速度过载。

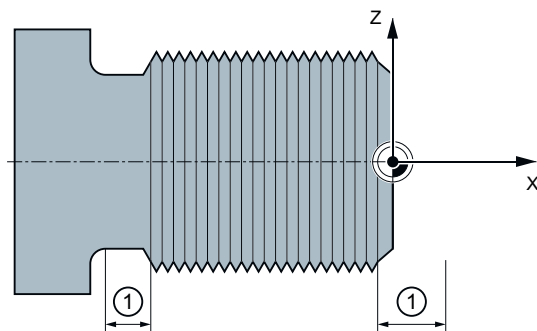
<n>	含义	<值>	含义
2	在螺纹链内的螺纹过渡处的加速度特性	= -1 (缺省设置)	在两个螺纹程序段之间的角部处进行平滑，以遵循动态特性极限。 在机床的动态特性允许硬过渡的情况下不进行平滑，例如基于轴速度跃变的过载系数的有效性 (MD32310 \$MA_MAX_ACCEL_OVL_FACTOR)。
		= 0	非平滑地经过两个螺纹程序段之间的角部，轴按照控制回路的设定跟随。
		> 0	预留

7.2.3.4

G33、G34、G35 中可编程的导入和导出行程 (DITS、DITE)

可在零件程序中通过地址 DITS 和 DITE 设定螺纹的导入和导出行程。

螺纹轴会在设定的行程内进行加速或制动。



① 导入或导出行程，根据加工方向

较短的导入行程

螺纹导入处有套环，给刀具启动斜坡留下的空间较小。

因此必须通过 DITS 缩短斜坡的长度。

较短的导出行程

由于连接至螺纹导出段，刀具减速斜坡的空间不够，可能会导致工件和刀沿之间的碰撞危险。可通过 DITE 设定较短的刀具减速斜坡。不过机械惯性的存在使得仍有发生碰撞的危险。

解决方法：编程更短的螺纹，减少主轴转速。

说明

DITE 在螺纹末端处作为精磨间距生效。从而使轴的运行平稳改变。

7.2 轨迹进给率F

影响

编写的导入和导出行程只会提升轨迹上的加速度。若两个行程的其中一个大于螺纹轴启用生效加速度时所需的行程，那么螺纹轴会以最大加速度加速或制动。

句法

DITS=<值> DITE=<值>

含义

DITS:	确定螺纹导入行程
DITE:	确定螺纹导出行程
<值>:	DITS 和 DITE 下只编写行程，不编写位置！ 编写的导入/导出行程会依据当前生效的单位制（英制或公制）接受处理。

示例

程序代码	注释
... N40 G90 G0 Z100 X10 SOFT M3 S500 N50 G33 Z50 K5 SF=180 DITS=1 DITE=3 N60 G0 X20	；在 z=53 时开始精磨。

更多信息

SD42010 \$SC_THREAD_RAMP_DISP

在主处理中切换至包含 DITS 和/或 DITE 的程序段时，编写的导入或导出行程会被接收至设定数据 SD42010 \$SC_THREAD_RAMP_DISP：

- SD42010 \$SC_THREAD_RAMP_DISP[0] = 编写的 DITS 值
- SD42010 \$SC_THREAD_RAMP_DISP[1] = 编写的 DITE 值

若在第一个螺纹程序段前未编写导入或导出行程，则会启动设定数据中的当前值。

通道/BAG/程序结束复位后的特性

通道/BAG/程序结束复位后，被 DITS 和/或 DITE 覆写的 SD42010 值仍然保持生效。

热启动后的特性

热启动时，设定数据恢复为被 DITS 和/或 DITE 覆写之前生效的值（标准特性）。

热启动后，如果也要将以 DITS 和 DITE 编程的值设为生效，则必须在机床数据 MD10710 \$MN_PROG_SD_RESET_SAVE_TAB 中列出设定数据 SD42010 \$SC_THREAD_RAMP_DISP:

MD10710 \$MN_PROG_SD_RESET_SAVE_TAB[<n>] = 42010

导入和/或导出行程非常短时的特性

如果导入和/或导出行程非常短，则螺纹轴的加速度要大于配置值。这会导致轴因加速而过载。

在导入螺纹时将会发出报警 22280 “编程的导入行程过短”（在机床数据 MD11411 \$MN_ENABLE_ALARM_MASK 中进行相应的设置）。报警只是一种信息，它对于零件程序的执行没有影响。

参见

G33、G34 和 G35 中轴的加速度特性（页 215）

7.2.3.5 螺纹切削期间快速返回

功能

“螺纹切削时的快速返回”功能可在以下情况下中断螺纹切削，并且不会损坏设备：

- NC 停止（NC/PLC 接口信号）
- 隐性触发 NC 停止的报警
- 切换快速输入

更多信息参见 NC 编程手册。

返回运行的编程可通过：

- 返回位移和返回方向（相对）
- 返回位置（绝对）

说明

攻丝

不能在攻丝（G331/G332）时使用快速返回功能。

编程

句法

使能快速返回，通过返回位移和返回方向编写返回运动：

G33 ... LFON DILF=<值> LFTXT/LFWP ALF=<值>

7.2 轨迹进给率F

使能快速返回，通过返回位置编写返回运动：

```
POLF[<轴名称>]=<值> LFPOS
POLFMASK/POLFMLIN(<轴名称 1>,<轴名称 2>,...)
G33 ... LFON
```

禁用螺纹切削快速返回：

```
LFOF
```

含义

LFON： 使能螺纹切削快速返回（G33）

LFOF： 禁用螺纹切削快速返回（G33）

DILF=： 确定返回行程的长度
可在零件程序中编程 DILF 来修改机床数据（MD21200 \$MC_LIFTFAST_DIST）中预设的值。

提示：
NC 复位后，设置的机床数据值总是生效。

LFTXT 使用 G 指令 LFTXT 和 LFWP 与 ALF 一起对返回方向进行控制。

LFWP： LFTXT： 执行返回运行的平面通过轨迹切线和刀具方向计算（缺省设置）。
LFWP： 执行返回运行的平面是有效的工作平面。

ALF=： 在返回平面中，使用 ALF 以不连续的角度编程返回方向。
使用 LFTXT 时，通过 ALF=1 确定返回方向为刀具方向。
使用 LFWP 时，工作平面中的方向被分配如下：

- G17（X/Y 平面）
ALF=1；以 X 方向返回
ALF=3；以 Y 方向返回
- G18（Z/X 平面）
ALF=1；以 Z 方向返回
ALF=3；以 X 方向返回
- G19（Y/Z 平面）
ALF=1；以 Y 方向返回
ALF=3；以 Z 方向返回

有关使用 ALF 编程的**更多信息**，请参见 NC 编程手册。

LFPOS： 将使用 POLFMASK 或 POLFMLIN 指定的轴退回到使用 POLF 编程的绝对轴位置

POLFMASK： 使能轴（<轴名称 1>,<轴名称 1>,...），使它独立退回至绝对位置

POLFMLIN:	使能轴，使它退回至线性关联的绝对位置。
提示:	受所有参与轴的动态特性的影响，到达退刀位置时不是总能建立起线性关联。
POLF[]:	确定目录中给定的几何轴或加工轴的绝对返回位置
有效性:	模态有效
=<值>:	在几何轴上，赋值被视为工件坐标系中的位置（WCS），在加工轴上，赋值被视为机床坐标系中的位置（MCS）。 赋值也可编程为增量值： =IC<值>
<轴名称>:	几何轴或加工轴的名称

说明

LFON 或 LFOF 总是可以编程，但只在螺纹切削（G33）时运用。

说明

POLF 和 POLFMASK/POLFMLIN 不仅仅限于在螺纹切削时使用。

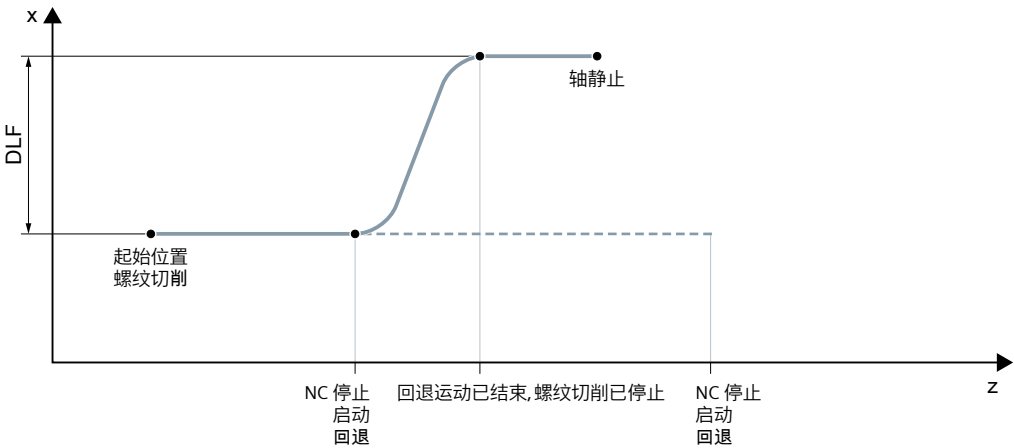


图 7-1 通过返回运动中断 G33

返回运动的动态特性

返回运动会以最大动态特性执行：

- MD32000 \$MA_MAX_AX_VELO[<轴>]（速度）
- MD32300 \$MA_MAX_AX_ACCEL[<轴>]（加速度）
- MD32431 \$MA_MAX_AX_JERK[<轴>]（急动）

7.2 轨迹进给率F

示例

程序代码	注释
N55 M3 S500 G90 G18	; 设置生效的加工平面
...	
N65 MSG ("螺纹切削")	
MM_THREAD:	
N67 \$AC_LIFTFAST=0	; 在螺纹开始前复位。
N68 G0 Z5	
N69 X10	
N70 G33 Z30 K5 LFON DILF=10 LFWP ALF=7	; 使能螺纹切削快速返回。
	; 返回行程=10 mm
	; 返回平面: Z/X (G18)
	; 返回方向 -X (ALF=3: 返回方向 +X)
N71 G33 Z55 X15	
N72 G1	; 取消螺纹切削。
N69 IF \$AC_LIFTFAST GOTOB MM_THREAD	; 螺纹切削中断时。
N90 MSG ("")	
...	
N70 M30	
N55 M3 S500 G90 G0 X0 Z0	
...	
N87 MSG ("攻丝")	
N88 LFOF	; 在攻丝前取消快速返回
N89 CYCLE...	; G33 攻丝循环。
N90 MSG ("")	
...	
N99 M30	

重新上电/复位后的特性

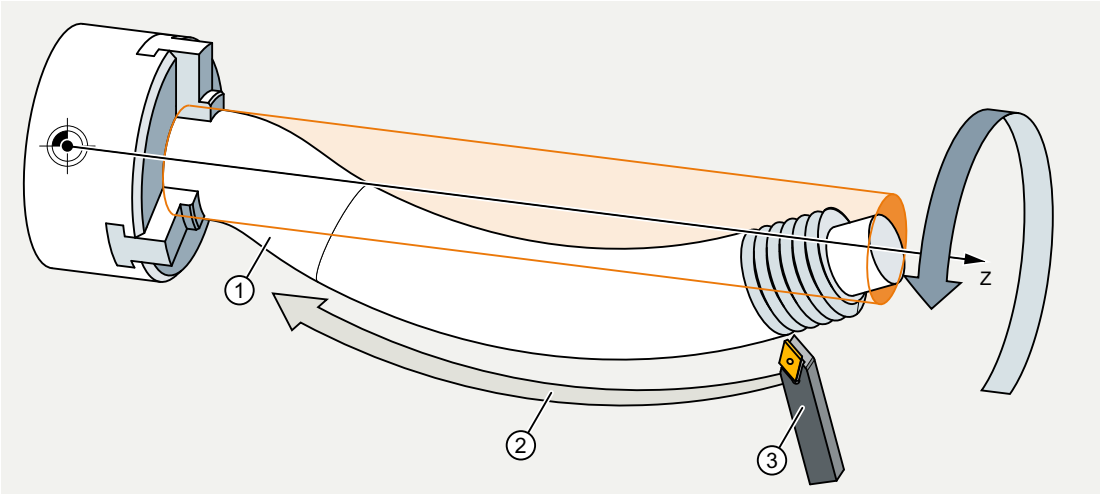
上电和复位后以下设置生效:

- 返回运动 (LFON / LFOF) 和返回方向 (LFTXT / LFWP) 的缺省设置: MD20150
\$MC_GCODE_RESET_VALUES
- 返回行程: MD21200 \$MC_LIFTFAST_DIST

7.2.3.6 球螺纹（G335，G336）

功能

借助 G 指令（G335 和 G336）可以车削球螺纹（即非柱形螺纹）。该功能应用在机床上因自重而下垂的超大型部件的加工。如果使用平行于轴的螺纹，会导致部件中心的螺纹牙过小。使用球螺纹可弥补该缺陷。



- ① 长的下垂工件
- ② 圆形刀具轨迹
- ③ 螺纹车刀

图 7-2 车削球螺纹

编程

车削球螺纹可通过 G335 或 G336 进行编程：

G335:	以 顺时针 圆弧刀具轨迹车削球螺纹
G336:	以 逆时针 圆弧刀具轨迹车削球螺纹

首先如同编写一个直螺纹一样，通过参数 I、J 和 K 指定轴终点和螺距。

7.2 轨迹进给率F

此外还要指定一段圆弧。该圆弧可如同 G2/G3 一样，通过圆心、半径、张角或中间点坐标进行编程。使用圆心编程球螺纹时要注意以下几点：因为在进行螺纹切削需要指定 I、J 和 K 以计算出螺距，因此在使用圆心编程时必须用 IR=...、JR=...和 KR=... 指定圆弧参数。

IR=...:	X 方向上的圆心直角坐标
JR=...:	Y 方向上的圆心直角坐标
KR=...:	Z 方向上的圆心直角坐标

说明

IR、JR 和 KR 是球螺纹插补参数的缺省名称，可通过机床数据 MD10651 \$MN_IPO_PARAM_THREAD_NAME_TAB 设置。

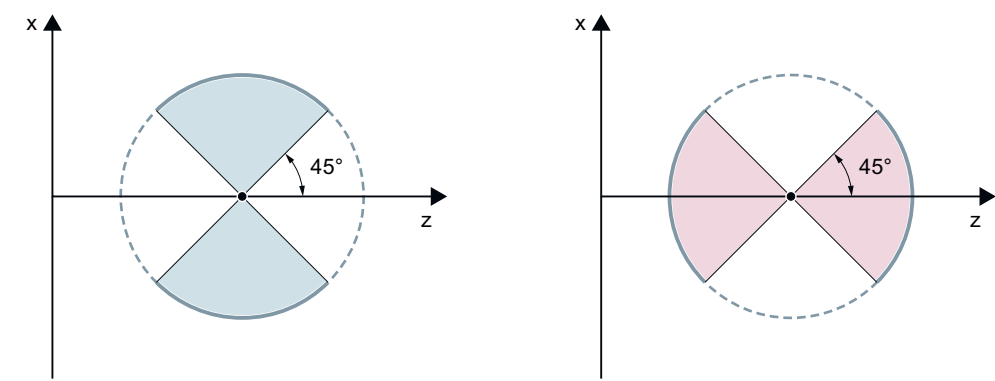
或者也可以指定起始点偏移 SF。

句法

编程球螺纹的句法具有以下常规格式：
G335/G336 <轴目标点坐标> <螺距> <圆弧> [<起始点偏移>]

允许的圆弧范围

G335/G336 中编程的圆弧必须在规定的范围内，即整个圆弧段都必须包含在指定的螺纹轴区间内（I、J 或 K）。



Z 轴允许的区域（通过 K 编程螺距）

X 轴允许的区域（通过 I 编程螺距）

下图所示的圆弧段是不允许的：

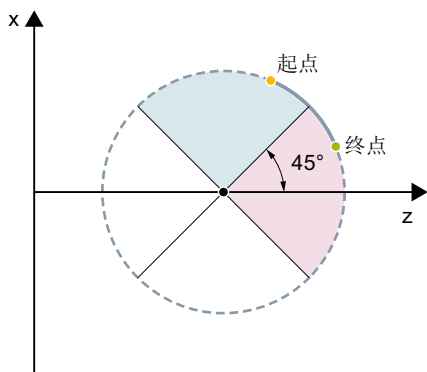


图 7-3 球螺纹：不允许的范围

前提条件

框架

在框架激活时也可使用 G335 和 G336。然而还是得注意遵守基本坐标系（BCS）中允许的圆弧范围。

特性

以下情形下 G335/G336 的特性：

- 上电/断电
- 运行方式切换
- NCK / BAG / 通道 / 零件程序结束复位
- 程序段搜索 / REPOS / ASUB
- 报警 / 急停 / 故障情形

对应 G33/G34/G35 的特性。

无特定限制。

示例

示例 1：使用终点和圆心编程顺时针球螺纹

程序代码	注释
N5 G0 G18 X50 Z50	; 逼近起始点。
N10 G335 Z100 K=3.5 KR=25 IR=-20 SF=90	; 车削顺时针球螺纹。

7.2 轨迹进给率F

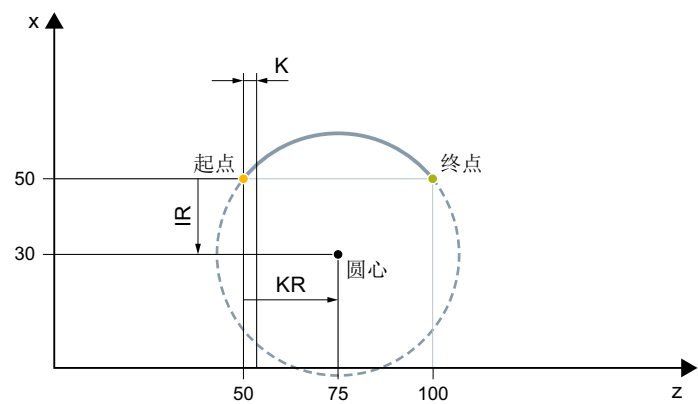


图 7-4 使用终点和圆心编程顺时针球螺纹

示例 2：使用终点和圆心编程逆时针球螺纹

程序代码	注释
N5 G0 G18 X50 Z50	; 逼近起始点。
N10 G336 Z100 K=3.5 KR=25 IR=20 SF=90	; 车削逆时针球螺纹。

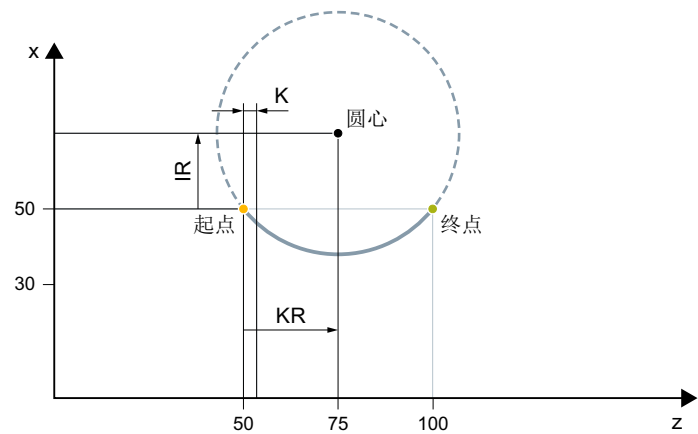


图 7-5 使用终点和圆心编程逆时针球螺纹

示例 3：使用终点和半径编程顺时针球螺纹

程序代码	注释
N5 G0 G18 X50 Z50	
N10 G335 Z100 K=3.5 CR=32 SF=90	

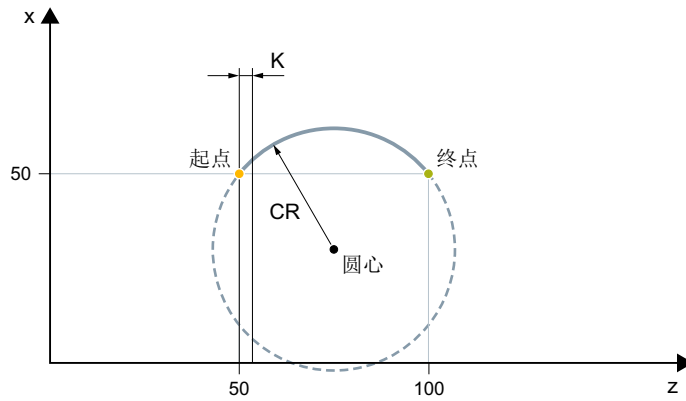


图 7-6 使用终点和半径编程顺时针球螺纹

示例 4：使用终点和张角编程顺时针球螺纹

程序代码

```
N5 G0 G18 X50 Z50
N10 G335 Z100 K=3.5 AR=102.75 SF=90
```

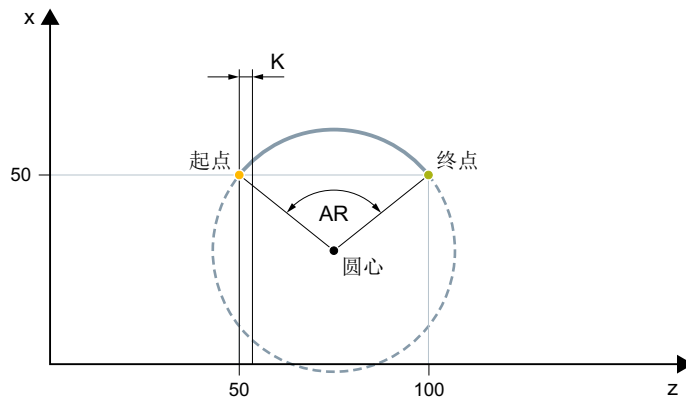


图 7-7 使用终点和张角编程顺时针球螺纹

示例 5：使用圆心和张角编程顺时针球螺纹

程序代码

```
N5 G0 G18 X50 Z50
N10 G335 K=3.5 KR=25 IR=-20 AR=102.75 SF=90
```

7.2 轨迹进给率F

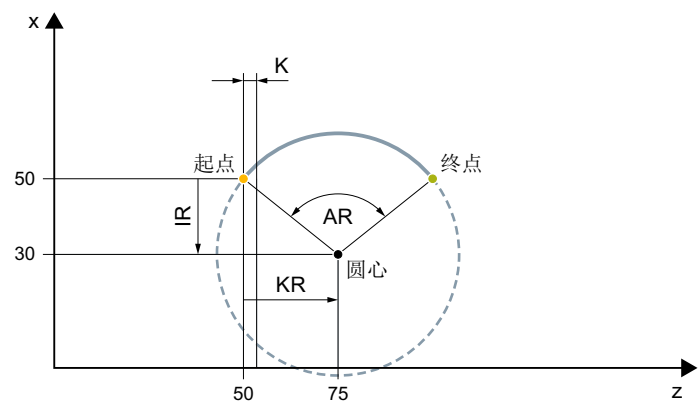


图 7-8 使用圆心和张角编程顺时针球螺纹

示例 6：使用终点和中间点编程顺时针球螺纹

```
程序代码
N5 G0 G18 X50 Z50
N10 G335 Z100 K=3.5 I1=60 K1=64
```

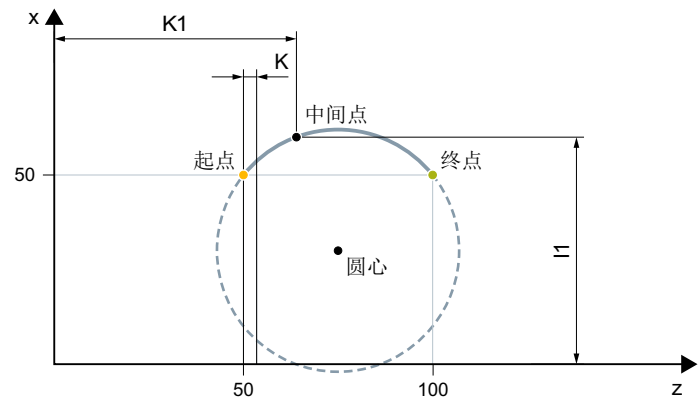


图 7-9 使用终点和中间点编程顺时针球螺纹

7.2.4 刚性攻丝（G331、G332）中的进给率

功能

通过 G331（攻丝）和 G332（攻丝回退）可进行刚性攻丝。

轨迹进给率

在使用 G331/G332 时，用于攻丝相关的轴的轨迹进给率 F 由编程的主轴转速 S 和编程的螺距大小决定：

$$F \text{ [mm/min]} = S \text{ [rpm]} * \text{螺距 [mm/rev]}$$

前提条件

刚性攻丝的前提是，主轴带位移测量系统并处于**位置闭环控制**中。

机床数据

- 防止停止事件
通过机床数据确定 G331 / G332 生效时的停止特性：
MD11550 \$MN_STOP_MODE_MASK, 位 <x> = <值>

位	值	含义
0	0	G331 / G332 生效时，轨迹运动或停留期间（G4） 不会 停止。
	1	G331 / G332 生效时，在轨迹控制运行中断（例如：G60 或 G4）时停止。

如果要在任何情况下都能防止 G331 / G332 生效时运动停止，则必须通过指令 DELAYFSTON 和 DELAYFSTOF 确定停止延迟区。

更多信息

参见功能手册之基本功能：程序运行，通过停止延迟区控制停止事件

说明

单程序段

如果在停止延迟区中激活单程序段，NC 会在停止延迟区外的第一个程序段处停止。如果在停止延迟区前已经选择了单程序段，NC 会在每个程序段交界处停止，在停止延迟区中也是如此！停止延迟区已取消。

倍率变化

如果在停止延迟区**前**调整倍率，倍率会在停止延迟区**中**生效。

如果在停止延迟区**中**调整倍率，倍率会在停止延迟区**后**生效。

- 倍率
可通过以下设置刚性攻丝时哪个倍率生效：
MD12090 \$MN_OVR_FUNCTION_MASK, 位 <x> = <值>

位	值	含义
0	0	在 G331/G332 中， 主轴倍率 生效 根据以下机床数据中的设置，倍率可基于编写的主轴转速，或者基于配置的主轴转速限制： MD12080 \$MN_OVR_REFERENCE_IS_PROG_FEED
	1	在 G331/G332 中， 轨迹倍率 生效 根据以下机床数据中的设置，倍率可基于编写的轨迹进给率，也可基于配置的轨迹进给限制： MD12082 \$MN_OVR_REFERENCE_IS_MIN_FEED 提示 当限值低于编程的轨迹进给率时才生效。

边界条件

无效补偿

以下补偿在刚性攻丝时无效：

- 可编程的轨迹进给倍率 OVR
- 快速移动倍率

7.2.5 带补偿夹具攻丝（G63）中的进给率

功能

G63 是攻丝的子功能，采用带补偿夹具的螺纹丝锥进行。该功能中不需要使用编码器（位移测量系统）。

转速 S、进给率 F、螺距

执行 G63 时，必须为主轴编写转速 S，为进给轴（加工螺纹长度的轴）编写进给率 F。

进给率须由用户通过转速 S 和螺距来计算：

进给率 $F [\text{mm/min}] = \text{转速 } S [\text{rev/min}] * \text{螺距 } [\text{mm/rev}]$

更多信息

有关 G63 的更多信息参见 NC 编程手册。

7.2.6 FGROUP 和 FGREF

编程

在刀具、工件或两者都被回转轴移动的加工过程中（例如：激光加工旋转的中空管），需要按照通用的方式在 F 值下作为轨迹进给率对生效的加工进给率进行编程。

为此，必须为参与运行的每根回转轴给定一个有效的半径（参考半径）。这可通过编程模式有效的 NC 地址实现：

`FGREF [<回转轴>] = <参考半径>`

参考半径的单位取决于 G70/G71/G700/G710 的设置。

7.2 轨迹进给率F

为了参与计算轨迹进给率，和以前一样，所有一同生效的轴都必须在 FGROUP 指令中进行定义。

为了在不进行 FGREF 编程的情况下与特性保持兼容，系统启动后及复位时，“1 度 = 1 毫米”的换算生效。

也就是说参考半径：

$$FGREF = 360 \text{ 毫米} / (2\pi) = 57.296 \text{ 毫米}$$

该缺省设置与生效的基本单位制 (MD10240 \$MN_SCALING_SYSTEM_IS_METRIC) 和当前生效的 G70/G71/G700/G710 设置无关。

FGROUP 中回转轴进给率值的特性：

程序代码
N100 FGROUP (X,Y,Z,A)
N110 G1 G91 A10 F100
N120 G1 G91 A10 X0.0001 F100

在上述程序中，程序段 N110 中编程的 F 值是回转轴进给率，单位为度/分，而在程序段 N120 中编程的进给率值取决于当前生效的英寸/公制设置，要么是 100 英寸/分，要么是 100 毫米/分。

注意
不同的单位制 如果只在程序段中编程了回转轴，FGREF 值也会生效。此时，常用的以度/分为单位的 F 值插补只适用于参考半径符合 FGREF 缺省设置的情况。 - 使用 G71/G710 时：FGREF[A]=57.296 - 使用 G70/G700 时：FGREF[A]=57.296/25.4

示例

下面的例子说明了 FGROUP 对轨迹行程和轨迹进给率的作用。变量 \$AC_TIME 包含从程序段开始的以秒为单位的时间。它只能在同步动作中使用。

程序代码	注释
N100 G0 X0 A0	
N110 FGROUP (X,A)	
N120 G91 G1 G710 F100	； 进给率 = 100 毫米/分或 100 度/分
N130 DO \$R1=\$AC_TIME	
N140 X10	； 进给率 = 100 毫米/分，轨迹行程 = 10 毫米，R1 = 约 6 秒
N150 DO \$R2=\$AC_TIME	
N160 X10 A10	； 进给率 = 100 毫米/分，轨迹行程 = 14.14 毫米，R2 = 约 8 秒
N170 DO \$R3=\$AC_TIME	

程序代码	注释
N180 A10	；进给率 = 100 度/分，轨迹行程 = 10 度，R3 = 约 6 秒
N190 DO \$R4=\$AC_TIME	
N200 X0.001 A10	；进给率 = 100 毫米/分，轨迹行程 = 10 毫米，R4 = 约 6 秒
N210 G700 F100	；进给率 = 2540 毫米/分或 100 度/分
N220 DO \$R5=\$AC_TIME	
N230 X10	；进给率 = 2540 毫米/分，轨迹行程 = 254 毫米，R5 = 约 6 秒
N240 DO \$R6=\$AC_TIME	
N250 X10 A10	；进给率 = 2540 毫米/分，轨迹行程 = 254.2 毫米，R6 = 约 6 秒
N260 DO \$R7=\$AC_TIME	
N270 A10	；进给率 = 100 度/分，轨迹行程 = 10 度，R7 = 约 6 秒
N280 DO \$R8=\$AC_TIME	
N290 X0.001 A10	；进给率 = 2540 毫米/分，轨迹行程 = 10 毫米，R8 = 约 0.288 秒
N300 FGREF[A]=360/(2*\$PI)	；1 度 = 1 英寸，通过有效的半径进行设置。
N310 DO \$R9=\$AC_TIME	
N320 X0.001 A10	；进给率 = 2540 毫米/分，轨迹行程 = 254 毫米，R9 = 约 6 秒
N330 M30	

诊断

读取参考半径

回转轴参考半径的值可由以下系统变量读取：

- 在同步动作或零件程序（会触发预处理停止）中写入以下系统变量，状态显示在操作界面中：

\$AA_FGREF[<轴>] 当前主运行值

- 在不带预处理停止的零件程序中，通过系统变量：

\$PA_FGREF[<轴>] 编程值

如果未编程值，则在两个回转轴变量中读取缺省值 $360 \text{ 毫米}/(2\pi) = 57.296 \text{ 毫米}$ （1 度对应 1 毫米）。

在线性轴上，两个变量中的值始终为 1 毫米。

读取影响速度的轨迹轴

参与到轨迹插补的轴可由以下系统变量读取：

- 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：

7.3 定位轴的进给率 (FA)

<code>\$AA_FGROUP[<轴>]</code>	当指定的轴通过初始设置或 FGROUP 编程而影响当前主运行程序段中的轨迹速度时，变量提供值“1”。无影响时，变量提供值“0”。
<code>\$AC_FGROUP_MASK</code>	提供一个使用 FGROUP 编程、会影响轨迹速度的通道轴的位码。

- 在不带预处理停止的零件程序中，通过系统变量：

<code>\$PA_FGROUP[<轴>]</code>	当指定的轴通过初始设置或 FGROUP 编程而影响轨迹速度时，变量提供值“1”。无影响时，变量提供值“0”。
<code>\$P_FGROUP_MASK</code>	提供一个使用 FGROUP 编程、会影响轨迹速度的通道轴的位码。

7.3 定位轴的进给率 (FA)

功能

轴专用进给率 FA 用于编写定位轴的速度。

FA 为模态有效。

其进给类型始终为 G94。

说明

不会超出最大轴速度 (MD32000 \$MA_MAX_AX_VELO)。

编程

每个零件程序段最多可编写 5 个轴专用进给率。

语法:

FA [<定位轴>]=<进给轴>

<定位轴>:	通道轴的名称 (MD20080 \$MC_AXCONF_CHANAX_NAME_TAB)
<进给值>:	进给速度
值域:	0.001...999 999.999 mm/min, degrees/min 或者 0.001...39 999.9999 inch/min

缺省设置

若未编写轴专用进给率 FA，以下机床数据中的缺省设置将生效:

MD32060 \$MA_POS_AX_VELO (定位轴速度的初始设置)

输出到 PLC

进给轴可输出到 PLC:

- 输入至通道专用 NC/PLC 接口:
`<Chan>.auxFFuncs[1].in.value`
- 输入至轴专用 NC/PLC 接口:
`<Axis>.basic.in.posAxisFeedrateValue`

输出时间可通过以下机床数据定义:

MD22240 \$MC_AUXFU_F_SYNC_TYP (F 功能的输出时间)

缺省设置下输出被抑制 (MD22240 = 3)，因为在连续路径运行中将 F 功能输出至 NC/PLC 接口可能会造成速度跃变。

更多信息: 功能手册之基本功能，“辅助功能向 PLC 的输出”功能说明

复位特性

程序结束或 NC 复位后的特性通过以下机床数据定义:

7.4 进给控制

MD22410 \$MC_F_VALUES_ACTIVE_AFTER_RESET（F 功能复位后保持生效）

值	含义
0	NC 复位后缺省值生效。
1	NC 复位后，最后一次编写的 FA 值生效。

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.posAxisFeedrateValue	LBP_Axis*.E_FPosAxis	DB31, ...DBD78
<Chan>.auxFFuncs[1].in.value	LBP_Chan*.F1 - F6	DB21, ...DBD160, 166, 172, 178, 184, 190

7.4 进给控制

7.4.1 进给禁止和进给/主轴停止

功能

在“进给禁止”或者“进给/主轴停止”状态下，轴将遵循制动特性曲线进行减速，直至达到停止状态。除螺纹加工 G33 外，遵循编写的轮廓。

通道专用进给禁止

借助以下通道专用 NC/PLC 接口信号，在所有运行模式下使一个通道的所有轴（几何轴和辅助轴）进入静止状态：

<Chan>.basic.out.disableFeed（进给禁用）

JOG 模式下针对几何轴的通道专用“进给停止”

借助以下通道专用 NC/PLC 接口信号，以针对具体轴的方式，将通道的相应几何轴的运行停止：

- `<Chan>.geoAxis1.out.disableFeed`（进给停止，几何轴）

这些接口信号仅在 **JOG** 运行模式下生效。

针对机床轴的轴专用“进给停止/主轴停止”

借助以下轴专用 NC/PLC 接口信号将机床轴的运行停止：

`<Axis>.basic.out.feedStopSpindleStop`（进给停止/主轴停止）

AUTO 和 MDA 运行模式

在 AUTO 和 MDA 运行模式下：

- 若使一个**轨迹轴**“进给停止”，则所有在当前程序段中运行的轴以及参与轨迹组的轴均进入静止状态。
- 若使一个**定位轴**“进给停止”，则仅该轴进入静止状态。

螺纹加工

螺纹加工	有效性
G33、G34、G35	生效 注意 轮廓偏差
G331、G332	生效
G63	轴：生效 主轴： 不 生效

针对机床轴的轴专用进给轴/主轴禁止

借助以下轴专用 NC/PLC 接口信号将机床轴的运行停止：

`<Axis>.basic.out.disable`（进给轴/主轴禁用）

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Chan>.basic.out.disableFeed	LBP_Chan*.A_FDdisable	DB21, ...DBX6.0
<Chan>.geoAxis1.out.disableFeed	LBP_Chan*.FDStop	DB21, ...DBX12.3, 16.3, 20.3
<Axis>.basic.out.feedStopSpindleStop	LBP_Axis*.A_FDSpStop	DB31, ...DBX4.3
<Axis>.basic.out.disable	LBP_Axis*.A_ASpDisable	DB31, ...DBX1.3

7.4.2 通过机床控制面板设置进给倍率

功能

“通过机床控制面板设置进给倍率”时，操作人员可在设备现场使用半分比值提升或降低轨迹进给率，调整将在机床上立即生效。此时调整的方法是将编写的进给率与 NC/PLC 上的对应倍率值相乘。

定位轴的进给率可针对轴修改。

主轴转速和切削速度（G96、G961）可通过“主轴倍率”修改。

进给率被修改时，系统仍将遵循加速度和速度限值。轨迹上不会出现轮廓错误。

可为轨迹轴和定位轴分别修改进给倍率。

倍率会对编写的值或限值（例如主轴转速限值 G26、LIMS）生效。

通道专用进给倍率和快速移动倍率

针对进给倍率和快速移动倍率，NC/PLC 接口中提供专属的使能信号和倍率系数：

<Chan>.basic.out.pathFeedrateOvr（进给倍率生效）

<Chan>.basic.out.pathFeedrateOvrFactor（进给倍率）

<Chan>.basic.out.pathRapidTraversOvr（快速移动倍率有效）

<Axis>.basic.out.enableOvr（快速移动倍率）

倍率系数可通过 PLC 以二进制编码或格雷码设定。启用的格式通过以下机床数据告知 NC：

MD12020 \$MN_OVR_FEED_IS_GRAY_CODE（轨迹进给倍率开关，格雷编码）

MD12040 \$MN_OVR_RAPID_IS_GRAY_CODE (快速移动倍率开关, 格雷编码)

下列固定指定关系适用于二进制码:

二进制码	十进制	倍率系数
00000000	0	0.00 $\triangleq$ 0%
00000001	1	0.01 $\triangleq$ 1%
00000010	2	0.02 $\triangleq$ 2%
00000011	3	0.03 $\triangleq$ 3%
00000100	4	0.04 $\triangleq$ 4%
...	...	...
01100100	100	1.00 $\triangleq$ 100%
...	...	...
11001000	200	2.00 $\triangleq$ 200%

采用格雷码时, 必须将开关位置所对应的倍率系数输入以下机床数据:

MD12030 \$MN_OVR_FACTOR_FEEDRATE [<n>] (轨迹进给倍率开关的权重系数)

MD12050 \$MN_OVR_FACTOR_RAPID_TRA [<n>] (快速移动倍率开关的权重系数)

激活的进给倍率对当前指定给通道的所有轨迹轴生效。激活的快速移动倍率对当前指定给通道的、且启用快速移动运行的所有轴生效。

无快速移动倍率开关存在 - 适用于 Basic Program

若无独立的快速移动倍率开关, 可在快速移动倍率和进给倍率之间进行切换。此时可通过 PLC 或操作面板前端选择启用哪一种倍率。快速移动倍率生效时, 进给倍率值被限制在 100% 的范围内。

- 通过操作面板前端激活快速移动倍率时, PLC 基本程序
 - 将为快速移动选择的进给倍率传输至快速移动倍率的激活信号:
DB21, ... DBX6.6 = DB21, ... DBX25.3
 - 将进给倍率值传输至快速移动倍率值:
DB21, ... DBB5 = DB21, ... DBB4
- 通过 PLC 选择快速移动倍率时, PLC 用户程序
 - 将快速移动倍率的激活信号置位:
DB21, ... DBX6.6 = 1
 - 将进给倍率值传输至快速移动倍率值:
DB21, ... DBB5 = DB21, ... DBB4

通道专用进给倍率和快速移动倍率的生效范围:	
• 对于生效的 G33、G34、G35:	不生效
• 对于生效的 G63:	不生效
• 对于生效的 G331、G332:	不生效

无快速移动倍率开关存在 - 适用于 Basic Program Plus

如果要在快速移动倍率和进给倍率之间切换，使用 Basic program Plus 时，必须在用户程序中对切换进行编程。

轨迹进给倍率的基准速度

“通过机床控制面板设置轨迹进给倍率”的基准速度可通过以下机床数据设置为和标准值（= 编写的进给率）不同的值：

MD12082 \$MN_OVR_REFERENCE_IS_MIN_FEED

轴专用进给倍率

对于每根定位轴，在 NC/PLC 接口中都有一个对应的使能信号，以及一个对应的用于进给倍率系数的字节：

<Axis>.basic.out.enableOvr（倍率生效）

<Axis>.basic.out.feedrateOvrFactor（进给倍率）

倍率系数可通过 PLC 以二进制编码或格雷码设定。启用的格式通过以下机床数据告知 NC：

MD12000 \$MN_OVR_AX_IS_GRAY_CODE（轴进给倍率开关，格雷编码）

下列固定指定关系适用于二进制码：

二进制码	十进制	倍率系数
00000000	0	0.00 $\triangleq$ 0%
00000001	1	0.01 $\triangleq$ 1%
00000010	2	0.02 $\triangleq$ 2%
00000011	3	0.03 $\triangleq$ 3%
00000100	4	0.04 $\triangleq$ 4%
...	...	...
01100100	100	1.00 $\triangleq$ 100%

二进制码	十进制	倍率系数
...	...	...
11001000	200	2.00 $\triangleq$ 200%

采用格雷码时，必须将开关位置所对应的倍率系数输入以下机床数据：

MD12010 \$MN_OVR_FACTOR_AX_SPEED [<n>]（轴进给倍率开关的权重系数）

轴专用进给倍率的生效范围：	
• 对于生效的 G33、G34、G35：	不生效
• 对于生效的 G63：	不生效 (倍率在 NC 中固定设为 100%)
• 对于生效的 G331、G332：	不生效 (倍率在 NC 中固定设为 100%)

主轴倍率

对于每根主轴，在 NC/PLC 接口中都有一个对应的使能信号，以及一个对应的用于主轴倍率系数的字节：

<Axis>.basic.out.enableOvr（倍率生效）

<Axis>.spindle.out.ovrFactor（主轴倍率）

倍率系数可通过 PLC 以二进制编码或格雷码设定。启用的格式通过以下机床数据告知 NC：

MD12060 \$MN_OVR_SPIND_IS_GRAY_CODE（主轴倍率开关，格雷编码）

下列固定指定关系适用于二进制码：

二进制码	十进制	倍率系数
00000000	0	0.00 $\triangleq$ 0%
00000001	1	0.01 $\triangleq$ 1%
00000010	2	0.02 $\triangleq$ 2%
00000011	3	0.03 $\triangleq$ 3%
00000100	4	0.04 $\triangleq$ 4%
...	...	...
01100100	100	1.00 $\triangleq$ 100%
...	...	...
11001000	200	2.00 $\triangleq$ 200%

7.4 进给控制

采用格雷码时，必须将开关位置所对应的倍率系数输入以下机床数据：
MD12070 \$MN_OVR_FACTOR_SPIND_SPEED [<n>]（主轴倍率开关的权重系数）

“主轴倍率” 的生效范围：	
• 对于生效的 G33、G34、G35：	生效
• 对于生效的 G63：	不生效
• 对于生效的 G331、G332：	生效

主轴倍率的基准

主轴倍率可以通过机床数据或设定数据限制的转速为基准，也可以编写的转速为基准。这通过以下机床数据设置：
MD12080 \$MN_OVR_REFERENCE_IS_PROG_FEED（倍率基准速度）

对倍率系数的限制

启用二进制编码倍率系数时，可对用于轨迹进给、轴进给和主轴转速的最大倍率进行限制：
MD12100 \$MN_OVR_FACTOR_LIMIT_BIN（二进制编码的倍率开关的限制）

倍率生效

倍率已使能时，通过机床控制面板设定的倍率值在所有运行方式下和所有机床功能中立即生效。

倍率无效

未激活倍率时，系统内部启用 100% 的倍率系数。NC/PLC 接口上的倍率系数不会被分析。
特例情形是二进制接口的零位置和格雷码接口的开关位置 1。输入到 NC/PLC 接口的倍率值会在这两个接口中接受分析。对于二进制接口，倍率系数始终为 0%。对于格雷码接口，机床数据中为开关位置 1 输入值将作为倍率值输出。其赋值应为“0”。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Chan>.basic.out.pathFeedrateOvr	LBP_Chan*.A_FD_ORA	DB21, ...DBX6.7
<Chan>.basic.out.pathRapidTraversOvr	LBP_Chan*.A_RT_ORA	DB21, ...DBX6.6
<Chan>.basic.out.pathFeedrateOvrFactor	LBP_Chan*.A_FD_OR	DB21, ...DBB4
<Chan>.basic.out.pathRapidTraversOvrFactor	LBP_Chan*.A_RT_OR	DB21, ...DBB5
<Axis>.basic.out.enableOvr	LBP_Axis*.A_ORActive	DB31, ...DBX1.7
<Axis>.spindle.out.ovrFactor	LBP_Axis*.A_SpOR	DB31, ...DBB19
<Axis>.basic.out.feedrateOvrFactor	LBP_Axis*.A_FD_OR	DB31, ...DBB0

OP → PLC

Basic Program Plus	Basic Program	
<HmiChan>.basic.in.rapidTraversOvrSelected	LBP_HMI.E_MMC_FD_OR4RT_OR	DB21, ...DBX25.3

7.4.3 可编程的进给倍率

功能

借助“可编程的进给倍率”功能可通过零件程序修改轨迹轴和定位轴的速度。

编程

句法	含义
OVR=<值>	修改轨迹进给率 F 的倍率
OVR A [<轴>=<值>	修改定位进给 FA 的倍率

可编程范围为 0 - 200%

默认设置：100%

有效性

NC/PLC 接口信号 <Chan>.basic.out.pathRapidTraversOvrFactor（快速移动或进给倍率生效）和 <Axis>.basic.out.enableOvr（轴专用倍率生效）不以可编程的进给倍率为基准。这些信号取消时，可编程的进给倍率继续生效。

生效的倍率为“可编程的进给倍率”与“通过机床控制面板设置进给倍率 (页 238)”的乘积。

“可编程的进给倍率”的缺省设置为 100%。

缺省设置在以下情形下生效：

- 未编写进给倍率，或
- 复位后，当以下机床数据未置位时：
MD22410 \$MC_F_VALUES_ACTIVE_AFTER_RESET（F 功能复位后继续生效）

说明

OVR 对 G33、G34、G35 无效。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Chan>.basic.out.pathRapidTraversOvrFactor	LBP_Chan*.A_RT_OR	DB21, ...DBB5
<Axis>.basic.out.enableOvr	LBP_Axis*.A_ORactive	DB31, ...DBX1.7

7.4.4 可编程的最大轨迹速度

利用“最大轨迹速度自适应”功能，系统会在一些关键程序段中自动限制或下调从各轴速度限值计算出的轨迹速度。通过 FLIM 的地址可以编程最大轨迹速度的限值。编程值在下一次 NC 复位或零件程序结束后才生效。

倍率始终对 FLIM 限制的进给率生效。因此，使用倍率开关可适当超出 FLIM。

生效方式

此功能在以下情况下生效：

- 在 AUTO 运行方式下
- 仅在轨迹轴上生效
- 仅在 G94 时
- 非快速移动

说明

“最大轨迹速度自适应”功能必须和直线插补 G94 组合使用。和其他插补类型（G93、G931、G95、G96、G97、G971、G972）组合时，该功能便无法使用。

句法

```
...  
FLIM=<值>  
...  
FLIM=-1  
...
```

含义

FLIM	最大轨迹速度自适用功能的指令地址	
<值>:	最大轨迹速度的限值	
	数据类型:	REAL
	取值范围:	$1.0 * 10^{-6} \dots 1.0 * 10^{38}$
	单位:	mm/min 或 inch/min（取决于当前单位制）
FLIM=-1	再次取消通过“FLIM=<值>”编程的限值。	

示例

程序代码	注释
... N1000 G0 X0 Y0 F10000 G64 G710 N1100 G1 X20 RNDM=5 N1200 G1 Y20 FLIM=5000	；滑块以最快 5 m/min 的速度在 Y 轴方向移动。

7.4 进给控制

程序代码	注释
N1300 G1 X0 Y40	； 滑块在 X 轴方向和 Y 轴方向的轨迹速度为最快 5 m/min。
N1400 G1 Y0 FLIM=-1	； 滑块以 10 m/min 的速度在 Y 轴方向移动。
M30	

7.4.5 空运行进给率

功能

在零件程序试车时，可使用空运行进给率功能来提升轨迹进给率，从而加快程序或部分程序的执行速度。此功能例如可应用于不加工工件的试运行。

激活

空运行进给率可在 AUTO 运行方式下选择，并通过 PLC 或操作面板前段激活。

通过操作面板前端激活时，以下接口信号会置位：

<HmiChan>.basic.in.dryRunSelected（空运行进给率已选择）

并通过 PLC 基本程序传输至接口信号：

<Chan>.basic.out.dryRunFeedrate（激活空运行进给率）

。

使用 Basic Program Plus 时，PLC 用户程序必须接管信号

<HmiChan>.basic.in.dryRunSelected（空运行进给率已选择）到接口信号

<Chan>.basic.out.dryRunFeedrate（激活空运行进给率）的传输。

有效性

接口信号“激活空运行进给率”置位的情形下，通过 SD42100 \$SC_DRY_RUN_FEED 设置的进给值将替代零件程序中编写的进给值生效，生效方式取决于设定数据 SD42101 \$SC_DRY_RUN_FEED_MODE 中的定义（参见参数设置）。

空运行进给率始终作为线性进给（G94）。

参数设置

激活试运行进给

该功能的激活时间点取决于以下机床数据中的设置：

MD10704 \$MN_DRYRUN_MASK（激活空运行进给率）

值	含义
0	空运行进给率只允许在程序段末尾激活和取消（缺省设置）
1	空运行进给率也可在程序执行期间（在零件程序段中）激活。 注意： 在程序执行期间激活会引起控制系统内部的重整操作，此时轴会短暂停止。这可能会对加工工件的表面质量产生影响。
2	空运行进给率可在不会触发轴停止的任意阶段激活和取消。但是其会随程序运行中一个“较晚”的程序段生效。

修改空运行进给率

空运行的进给率在以下设定数据中输入：

SD42100 \$SC_DRY_RUN_FEED（空运行进给率）

该设定数据可通过操作面板前端在“参数”操作区修改。

选择被 NC 接受时，以下 NC/PLC 接口信号将置位：

<Chan>.basic.in.dryRunFeedrateActive（空运行进给率有效）

满足以下条件时，生效的空运行进给率会在操作面板前段的状态栏以“DRY”显示：

- 选择在程序停止时在程序段末尾进行，或
- 机床数据 MD10704 \$MN_DRYRUN_MASK 在程序执行期间置“1”。

空运行进给率的生效方式

SD42100 中输入的空运行进给率的生效方式可通过以下设定数据设置：

SD42101 \$SC_DRY_RUN_FEED_MODE

值	含义
0	将编写的进给率与 SD42100 中的空运行进给率相比较，之后启用其中较大的一个运行（缺省设置）。
1	将编写的进给率与 SD42100 中的空运行进给率相比较，之后启用其中较小的一个运行。
2	SD42100 中输入的空运行进给率直接生效，不考虑编写的速度。
3 -9	预留

7.4 进给控制

值	含义
10	除螺纹切削（G33、G34 和 G35）和攻丝（G331、G332 和 G63）外，与配置 0 相同。执行上述功能采用编写的进给率。
11	除螺纹切削（G33、G34 和 G35）和攻丝（G331、G332 和 G63）外，与配置 1 相同。执行上述功能采用编写的进给率。
12	除螺纹切削（G33、G34 和 G35）和攻丝（G331、G332 和 G63）外，与配置 2 相同。执行上述功能采用编写的进给率。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Chan>.basic.out.dryRunFeedrate	LBP_Chan*.A_DRY	DB21, ... DBX0.6

NC → PLC

Basic Program Plus	Basic Program	
<Chan>.basic.in.dryRunFeedrateActive	LBP_Chan*.E_DRY	DB21, ... DBX318.6

OP → PLC

Basic Program Plus	Basic Program	
<HmiChan>.basic.in.dryRunSelected	LBP_HMI.E_MMC_DRY	DB21, ... DBX24.6

7.4.6 一个程序段中的多个进给值

功能

通过“一个程序段中的多个进给值”功能，可根据外部数字量和/或模拟量输入运行同步地激活一个 NC 程序段中的 6 个进给轴、暂停时间以及返回。

如果暂停时间或返回行程的输入被激活，那么轨迹轴或相关单个轴的剩余行程将被删除，并启动暂停时间或返回。

插补周期内会引导返回。

信号

此功能的输入信号汇总在一个输入字节中。该字节内的功能为固定指定。

表格 7-1 “一个程序段中多个进给值”功能的输入字节

	位							
	7	6	5	4	3	2	1	0
输入编号	E7	E6	E5	E4	E3	E2	E1	E0
进给地址	F7	F6	F5	F4	F3	F2	ST	SR

E7 至 E2：激活进给 F7 至 F2

E1：激活暂停时间 ST/STA（单位：s）

E0：激活返回运动 SR/SRA

信号的优先级

信号的询问顺序从 E0 开始升序排列。因此返回运行（SR）的优先级最高，进给率 F7 最低。

SR 和 ST 可以终止使用 F2 到 F7 激活的进给运行。

SR 也会终止 ST，即完整功能。

最高优先级信号决定当前的进给率。

最高优先级输入（F2 - F7）丢失时的特性可通过以下机床数据定义：

MD21230 \$MC_MULTFEED_STORE_MASK（“一个程序段中多个进给值”功能的存储特性）

位	值	含义
2 ... 7	0	最高优先级输入丢失时，对应进给率不保留（缺省设置）
	1	位 2 ... 7 置位时，最高优先级输入信号选择的对应进给率（F2 至 F7）将保留，即使该输入信号丢失，且一个较低优先级的输入存在。

以下情形下满足程序段结束标准：

- 达到编写的终点位置
- 返回运动（SR）结束
- 暂停时间（ST）结束

硬件指令

可向“一个程序段中多个进给值”功能的输入字节指定最多两个 NC I/O 的数字量输入字节或比较器字节：

MD21220 \$MC_MULTFEED_ASSIGN_FASTIN（为“一个程序段中多个进给值”功能指定 NC I/O 的输入字节），位 0 ... 15

此外还可取反输入位：

MD21220 \$MC_MULTFEED_ASSIGN_FASTIN，位 16 ... 31

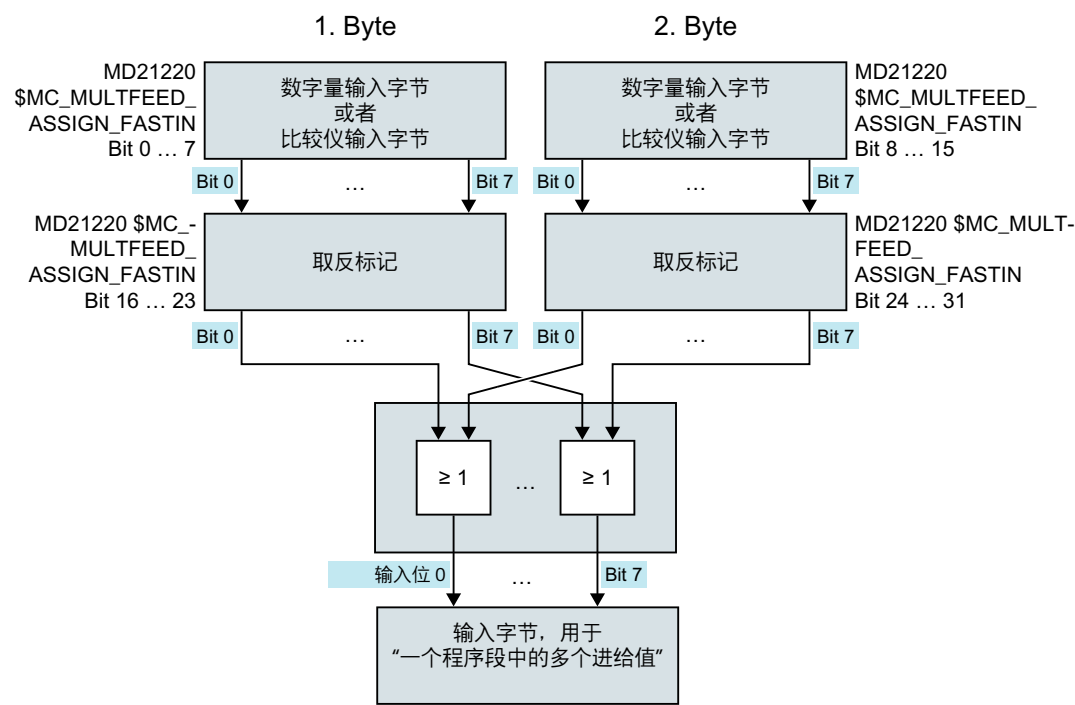


图 7-10 为“一个程序段中多个进给值”功能安排信号

数字量输入字节的安排以及比较器参数设置等相关信息参见：

更多信息

功能手册之基本功能：数字量和模拟量 NC I/O

编程

轨迹运行

在地址 F 中编程轨迹进给率，未出现输入信号时该值一直有效。其为模态有效。

除了轨迹进给率外，还可通过 F2=... 至 F7=... 在程序段中编写最多 6 个进给率。数字扩展给出了输入的位编号，改变它可以激活进给率：

示例：

F7=1000 ; 7 与输入位 7 相对应

编写的值逐段生效。后续程序段中 F 下编写的轨迹进给率生效。

餐厅时间和返回行程在程序段中的附加地址下编写：

ST=... 暂停时间（磨削时为修光时间）

SR=... 返回行程

这些地址逐段生效。

轴运行

在地址 FA 中编程轴向进给率，未出现输入信号时该值一直有效。这些进给率模态有效。

也可以用 FMA[2,<轴>]=... 至 FMA[7,<轴>]=... 在程序段内给每个轴编程最多 6 个进给率。

第一个尖括号中给出了输入的位编号，改变它可以激活进给率：第二个注明了进给率适用于哪根轴。

示例：

FMA[3,Y]=1000 ; Y 轴的轴进给，对应输入位 3

FMA 下编写的值逐段生效。后续程序段中 FA 下编写的进给率生效。

可为各轴设定额外设定暂停时间和返回行程：

STA[<轴>]=... 轴暂停时间（修光时间）

SRA[<轴>]=... 轴向返回行程

尖括号中的内容用于指定修光时间或返回行程适用于的轴。

示例：

STA[X]=2.5 ; 轴 X 的修光时间为 2.5 s。

SRA[X]=3.5 ; 轴 X 的返回行程为 3.5（单位例如为 mm）。

这些地址逐段生效。

说明

返回行程

返回行程的单位基于当前生效的单位制（mm 或 inch）
返回行程的方向始终与当前运行方向相反。使用 SR/SRA 编写返回行程的绝对值。不需要编写正负号。

说明

POS 替代 POSA

如果基于一个外部输入为一个轴编程了进给率、暂停时间（修光时间）或返回行程，那么在该程序段中不能将该轴编程为 POSA 轴（超过程序段限制的定位轴）。

说明

状态询问

也可以为不同轴的同步指令询问输入状态。

说明

预读

程序段预读功能对一个程序段内的多个进给率有效。如此就可以使用程序段预读功能来限制当前进给率。

应用

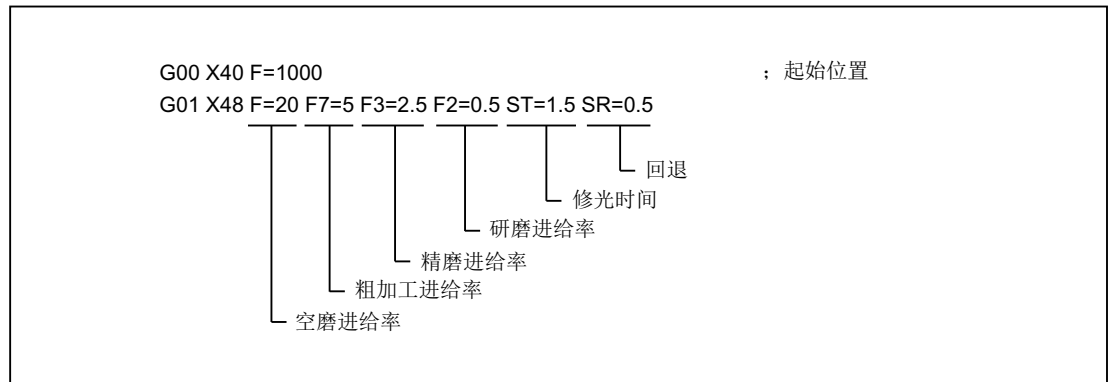
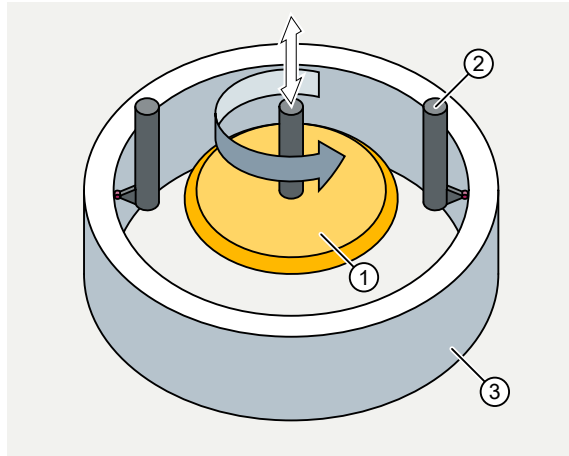
“一个程序段中多个进给值”功能主要应用于磨削工艺。但同时也不仅限于此功能。

其典型应用例如包括：

- 模拟量或数字量测量钳
基于外部模拟量和数字量输入，可激活不同的进给值、一个暂停时间以及一个返回行程，同时阈值通过设定数据设定。
- 通过接近开关从深进给切换至工作进给。

示例

滚珠轴承环的内部磨削，其中实际直径通过测量钳采集，并根据阈值为粗加工、精加工和修光激活必要的进给值。测量钳位置也提供终点位置。因此程序段结束标准不仅取决于进给轴的轴位置，也由测量钳决定。



7.4.7 固定进给值

功能

借助“固定进给值”功能，可通过机床数据激活已定义的固定进给值（最多 4 个）来取代编写的进给值或配置的 JOG 速度。

此功能可在 AUTO 和 JOG 运行方式下使用。

AUTO 运行方式下的特性

运行轮廓时启用激活的固定进给值代替编写的进给率。

JOG 运行方式下的特性

启动激活的固定进给值代替配置的 JOG 速度/JOG 快速移动速度来运行轴。运行方向通过接口信号设定。

参数设置

固定进给值如下设置：

- 针对直线轴通过机床数据：
MD12202 \$MN_PERMANENT_FEED[<n>]
- 针对回转轴通过机床数据：
MD12204 \$MN_PERMANENT_ROT_AX_FEED[<n>]

其中 <n> = 0、1、2、3（对应固定进给值 1、2、3、4）

说明

固定进给值始终为线性进给值。旋转进给率将在系统内部切换为线性。

激活

固定进给值通过 NC/PLC 接口激活：

- AUTO 运行方式下针对轨迹轴/几何轴，通过通道专用接口信号：
<Chan>.basic.out.pathFixedFeedrateSelection（激活固定进给 1）
- JOG 运行方式下针对机床轴，通过轴专用接口信号：
<Axis>.basic.out.fixedFeedrateSelection

边界条件**有效性**

以下情形下，“固定进给值”功能**不**生效：

- 主轴上
- 定位轴上
- 攻丝

倍率 = 0

倍率 = 0 时的运行特性取决于以下机床数据中的设置：

MD12200 \$MN_RUN_OVERRIDE_0

DRF 偏移
选择固定进给值时无法激活 DRF 偏移。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Chan>.basic.out.pathFixedFeedrateSelection	LBP_Chan*.A_FixedFD1	DB21, ...DBX29.0..3
<Axis>.basic.out.fixedFeedrateSelection	LBP_Axis*.A_FixedFD1..4	DB31, ...DBX3.2..5

7.4.8 可编程的进给特性曲线

功能

为了实现进给特性曲线的灵活设定，根据 DIN 66025 对进给特性曲线编程增加了线性曲线和三次曲线。
三次曲线可以直接编程或作为插补样条编程。

编程

可编写以下进给特性曲线：

- **FNORM**
特性符合 DIN 66025（缺省设置）
程序段中编写的 F 值在程序段的整个轨迹恒定设定，之后作为固定模态值。
- **FLIN**
程序段中编写的 F 值从程序段开始处的当前值，直至程序段末尾沿轨迹线性运行，之后作为固定模态值。
- **FCUB**
逐段编写的 F 值基于程序段终点通过样条连接。样条开始时与前一进给设定呈切向，结束时与后一进给设定呈切向。若一个程序段中缺少 F 地址，则为其使用最后编写的 F 值。
- **FPO**
F 地址 [句法：F=FPO (. . . , . . . , . . .)] 通过一个多项式从当前值到编写其的程序段的末尾。结束值作为模态值。

前提条件

FLIN/FCUB

通过 FLIN 或 FCUB 编写的轨迹速度曲线不和 G95 旋转进给率一起生效，也不和恒定切削速度 G96/G961、G97/G971 一起生效。

更多信息

可编程的进给特性曲线的更多信息参见 NC 编程手册。

7.4.9 倒角/倒圆进给率 FRC、FRCM

在从平面过渡至倒角/倒圆时，切屑条件可能会本质变化。因此，为了达到所需的表面质量，需要为轮廓单元倒角/倒圆启用专用的、经过优化的进给值。

功能

倒角/倒圆的进给率可通过 NC 地址编写。

编程

句法:
... FRC/FRCM=<值>

含义:

- FRC: 逐段生效的倒角/倒圆进给率
- FRCM: 模态生效的倒角/倒圆进给率
- <值>: 根据生效的进给类型，进给值如下解译:
- G94、G961、G971: 进给率，单位 mm/min, inch/min 或 %/min
 - G95、G96、G97: 旋转进给率，单位 mm/rev 或 inch/rev

说明

只有在程序段中编程了倒圆/倒角，或者激活了 RNDM 时，FRC 才生效。
FRC 会覆盖当前程序段中的 F 值或 FRCM 值。
FRC 中编程的进给率必须大于零。
通过 FRCM=0 激活 F 中编程的进给用于倒角/倒圆。

参数设置

将倒角/倒圆工艺指定给前一/后一程序段

在 F 和 FRC/FRCM 程序段中，进给类型（G94、G95、G96、G961 ...）及其转换为的内部格式必须统一。为此可启用以下机床数据：

MD20201 \$MC_CHFRND_MODE_MASK（倒角/倒圆特性）

位	值	含义
0	0	倒角/倒圆工艺（进给率、进给类型、M 指令）由后一程序段决定（缺省设置）。
	1	倒角/倒圆工艺由前一程序段决定。

最大空程序段数量

倒角/倒圆生效时，两个包含运行信息的程序段间的空程序段（补偿级中无运行信息的程序段）的数量受限。最大数量通过以下机床数据定义：

MD20200 \$MC_CHFRND_MAXNUM_DUMMY_BLOCKS（倒角/倒圆中的空程序段）

前提条件

FLIN/FCUB

进给插补 FLIN 和 FCUB 不适用于倒角和倒圆。

G0

通过 G0 运行倒角时，FRC/FRCM 无效。可根据 F 值编程，不会触发故障。

切换 G94 ↔ G95

如果编程了 FRCM，在 G94 ↔ G95 切换后必须对 F 和 FRCM 的值都进行重新编程。如果只重新编程了 F 值，且在进给类型转换前 FRCM > 0，则输出故障信息。

示例

示例 1：MD20201 位 0 = 0；从后一程序段接收进给值（缺省设置）

程序代码	注释
N10 G0 X0 Y0 G17 F100 G94	
N20 G1 X10 CHF=2	; 倒角 N20-N30, F=100 mm/min
N30 Y10 CHF=4	; 倒角 N30-N40, FRC=200 mm/min
N40 X20 CHF=3 FRC=200	; 倒角 N40-N60, FRCM=50 mm/min

7.4 进给控制

程序代码	注释
N50 RNDM=2 FRCM=50	
N60 Y20	; 模态倒圆 N60-N70, FRCM=50 mm/min
N70 X30	; 模态倒圆 N70-N80, FRCM=50 mm/min
N80 Y30 CHF=3 FRC=100	; 倒角 N80-N90, FRC=100 mm/min
N90 X40	; 模态倒圆 N90-N100, F=100 mm/min (取消 FRCM)
N100 Y40 FRCM=0	; 模态倒圆 N100-N120, G95 FRC=1 mm/rev
N110 S1000 M3	
N120 X50 G95 F3 FRC=1	
...	
M02	

示例 2：MD20201 位 0 = 1；从前一程序段接收进给值（推荐设置！）

程序代码	注释
N10 G0 X0 Y0 G17 F100 G94	
N20 G1 X10 CHF=2	; 倒角 N20-N30, F=100 mm/min
N30 Y10 CHF=4 FRC=120	; 倒角 N30-N40, FRC=120 mm/min
N40 X20 CHF=3 FRC=200	; 倒角 N40-N60, FRC=200 mm/min
N50 RNDM=2 FRCM=50	
N60 Y20	; 模态倒圆 N60-N70, FRCM=50 mm/min
N70 X30	; 模态倒圆 N70-N80, FRCM=50 mm/min
N80 Y30 CHF=3 FRC=100	; 倒角 N80-N90, FRC=100 mm/min
N90 X40	; 模态倒圆 N90-N100, FRCM=50 mm/min
N100 Y40 FRCM=0	; 模态倒圆 N100-N120, F=100 mm/min
N110 S1000 M3	
N120 X50 CHF=4 G95 F3 FRC=1	; 倒角 N120-N130 使用 G95 FRC=1 mm/rev
N130 Y50	; 模态倒圆 N130-N140, F=3 mm/rev
N140 X60	
...	
M02	

7.4.10 逐段进给率 FB

功能

使用“逐段进给率”功能可为一个零件程序段设定一个专用的进给率。在此程序段之后，之前模态生效的轨迹进给率将重新生效。

编程

句法:

...FB=<值>

含义:

FB: 当前程序段的专用进给率

<值>: 根据生效的进给类型，进给值如下解译:

- G94、G961、G971: 进给率，单位 mm/min, inch/min 或 °/min
- G95、G96、G97: 旋转进给率，单位 mm/rev 或 inch/rev

说明

FB 中编程的进给率必须大于零。

若该程序段中未编写运行（例如：计算程序段），那么 FB 将不生效。

如果没有给倒角/倒圆编程明确的进给率，那么 FB 的值还适用于该程序段中的一个倒角/倒圆轮廓元素。

不能同时编程 FB 和 FD（启用进给倍率的通过手轮进行的手动运行）或者 F（模态轨迹进给）。

7.4.11 单轴动态响应的控制

单轴

可以在零件程序、同步动作和 PLC 对单轴进行编程:

- 零件程序:
 - POS [<轴>]=...
 - POSA [<轴>]=...
 - SPOS [<轴>]=...
 - SPOSA [<轴>]=...
 - OS [<轴>]=...
 - OSCILL [<轴>]=...

7.4 进给控制

- 同步动作:
 - EVERY ... DO
 - POS [<轴>]=...
 - SPOS [<主轴>]=...
 - MOV [<轴>]=...
- PLC:
 - SINU_TraversePosAxis (Basic Program - FC18)

动态响应特性

通过下列方式控制轴的动态响应特性:

- MD32060 \$MA_POS_AX_VELO (定位轴速度)
 - 可以修改有效的定位轴速度:
 - 零件程序/同步动作: 轴进给率 FA 或者百分比的进给量修正 OVRA
 - PLC: 预先设定进给比率 FRate 或者覆盖轴倍率
- MD32300 \$MA_MAX_AX_ACCEL (最大轴加速度)
 - 可以修改有效的最大轴加速度:
 - 零件程序间接修改: 写入机床数据, 随后“激活机床数据”
 - 零件程序直接修改: 百分比的加速度倍率 ACC
 - 同步动作间接修改: 写入机床数据, 并触发用于激活“激活机床数据”功能的异步子程序
 - 同步动作直接修改: 百分比的加速度倍率 ACC (不能由 PLC 设定)。
 - 通过 PLC 同样可以进行同步动作中的操作。
- 零件程序指令: BRISKA, SOFTA, DRIVEA, JERKA
 - 不能在同步动作中编程 (只能间接通过 ASUP)。
 - 不能由 PLC 设定 (只能间接通过异步子程序设定)。
- 有效的伺服参数组
 - 可以修改有效的伺服参数组:
 - 零件程序/同步动作: SCPARA
 - PLC: <Axis>.basic.out.posCtrlParameterSetSelection (控制器参数组)
 - 有关伺服参数组的详细资料参见“位置控制器的参数组 (页 189)”。

说明**动态响应修改**

在零件程序中进行的动态响应修改不会影响指令轴或 PLC 轴的运行。在同步动作中进行的动态响应修改不会影响零件程序中已编程的程序运行。

前馈控制

前馈控制的类型以及哪些轨迹轴应当采用前馈控制，可以在零件程序中直接使用 FFWON/FFWOF 进行编程。在同步动作以及 PLC 控制中只能通过异步子程序间接进行编程。

百分比的加速度倍率（ACC）

在零件程序或同步动作中可以使用指令 ACC 对机床数据中：

MD32300 \$MA_MAX_AX_ACCEL（最大轴加速度）

设定的加速度在 0% – 200% 的范围内进行修改。

语法：

ACC [<轴>] = <值>

含义：

ACC: 百分比加速度倍率的编程关键字

<轴>: 通道轴或主轴的名称

<值>: 加速度变化率，相对于 MD32300 相关

值域: 0 ... 200

当前轴的加速度值可以通过系统变量 \$AA_ACC 进行读取。该值取决于：

$$\$AA\_ACC[<轴>] = (MD32300 \$MA\_MAX\_AX\_ACCEL[<轴>]) * ACC[<轴>] / 100$$

通过 MD32320 \$MA_DYN_LIMIT_RESET_MASK 可以确定通道复位时或者零件程序结束 M30 处 ACC 编程值的初始设置。

说明

用 ACC 编程的加速度倍率可以通过系统变量 \$AA_ACC 进行读取。然而在零件程序中读取 \$AA_ACC 要在其他时间进行，不同于在同步动作中读取。

系统变量 \$AA_ACC 只包含在零件程序中使用 ACC 编程的数值，如果在此期间没有在同步动作中编程 ACC 来修改加速度倍率。反之亦然。

百分比的加速度倍率与主运行轴

根据系统变量 \$AA_ACC 是在零件程序中或是在同步动作中被读取，分别输出用于 NC 轴或主运行轴（指令轴、PLC 轴、异步摆动轴等）的使用 ACC 编程的加速度倍率值。

为了保证结果正确，必须始终从原先 ACC 编程所在的位置（零件程序或同步动作）读取系统变量 \$AA_AC。

示例：

在零件程序中由 ACC 写入：

N80 G01 POS[X]=100 FA[X]=1000 ACC[X]=90 IPOENDA[X]

在同步动作中由 ACC 写入：

N100 EVERY \$A_IN[1] DO POS[X]=50 FA[X]=2000 ACC[X]=140 IPOENDA[X]

在零件程序中由 ACC 写入并读取 \$AA_ACC :

ACC[X]=50 ; 写
RO=\$AA_ACC[X] ; 读

IF (RO <> \$MA_MAX_AX_ACCEL[X] * 0.5) ; 检查
 SETAL(61000)
ENDIF

在同步动作中由 ACC 写入并读取 \$AA_ACC :

WHEN TRUE DO ACC[X]=25 R0=\$AA_ACC[X] ; 写入并读取
G4 F1

IF (RO <> \$MA_MAX_AX_ACCEL[X] * 0.25) ; 检查
 SETAL(61001)
ENDIF

单轴的运动结束条件

类似于轨迹插补时的程序段转换标准（G601, G602, G603），也可以在零件程序/同步动作中对单轴运动结束条件进行编程：

程序指令	运动结束条件
FINEA [<轴>]	“精准停”
COARSEA [<轴>]	“粗准停”
IPOENDA [<轴>]	“插补器停止” (IPO-Stopp)

在程序结束或 NC 复位后会保留最后编程的数值。

可通过轴专用的系统变量 \$AA_MOTEND 读取当前生效的运动结束条件。

说明

根据系统变量 \$AA_MOTEND 是在零件程序中或是在同步动作中被读取，其中分别包含有用于 NC 轴或主运行轴的数值。

示例：

零件程序：

```
N80 G01 POS[X]=100 FA[X]=1000 ACC[X]=90 COARSEA[X]
```

同步动作：

```
N100 EVERY $A_IN[1] DO POS[X]=50 FA[X]=2000 ACC[X]=140  
IPOENDA[X]
```

更多信息

更多关于程序段转换以及运动结束条件（FINEA、COARSEA 和 IPOENDA）的信息请参见：程序段切换 (页 297)

可编程的伺服参数组 (SCPARA)

在零件程序/同步动作中可以使用 SCPARA 设定伺服参数组。

句法

SCPARA [<轴>] = <参数组编号>

含义

SCPARA:	激活预设伺服参数组的关键字
<轴>:	通道轴的名称
<参数组编号>:	待激活伺服参数组的编号

说明

可以从 PLC 用户程序禁止激活由 SCPARA 设定的参数组：
`<Axis>.basic.out.disableParameterSetSwitchover = 1`（禁止通过 SCPARA 设定参数组）
此时 不 显示信息。

可通过系统变量 \$AA_SCPAR 读取当前有效参数组的编号。

边界条件

不同的运动结束条件

当运动结束条件不同时零件程序段会以不同速度结束。这可能会在工艺循环和 PLC 用户程序中导致边际效应。

参数组切换

如果要在某个零件程序中、某个同步动作或 PLC 中切换伺服参数组，就必须对 PLC 用户程序进行扩展。

上电

上电之后设定下列初始设置值：

- 用于所有单轴插补的百分比加速度倍率：100%
- 用于所有单轴插补的运动结束条件：FINEA
- 伺服参数组：1

运行方式切换

在运行方式从 AUTO 切换为 JOG 时已编程的动态响应修改会保持有效。

复位

在复位时会保留用于零件程序设定的最后一个编程值。用于主运行插补的设定值不变。

程序段搜索

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。程序段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已编程的运动结束条件。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.posCtrlParameterSetSelection	LBP_Axis*.A_Para_A..C	DB31, ...DBX9.0..2
<Axis>.basic.out.disableParameterSetSwitchover	LBP_Axis*.A_LockParSetDef	DB31, ...DBX9.3

7.5 边界条件

尺寸单位

进给率的有效尺寸单位取决于所设定的单位制以及所输入的轴类型：

MD10240 \$MN_SCALING_SYSTEM_IS_METRIC （控制器的初始单位制：英制或公制）

MD30300 \$MA_IS_ROT_AX （回转轴或线性轴）

进给率类型的初始设置

进给率类型的初始设置定义在机床数据中：

MD20150 \$MC_GCODE_RESET_VALUES （G 功能组的初始设置）

缺省设置为 G94。

只有当启动一个零件程序时才会显示进给率类型的初始设置。

复位后的有效性

最后所编程的 F、FA、OVR、OVRA 的值在复位之后是否有效，取决于机床数据中的设置：

MD22410 \$MC_F_VALUES_ACTIVE_AFTER_RESET （F 功能复位后保持生效）

7.5 边界条件

主轴定位

当 G95、G96、G961、G97、G971、G33、G34、G35 有效时不应进行主轴定位，因为进行主轴定位之后推导出的轨迹进给率为 0。如果此时尚未到达所编程的轴位置，则程序段会无法结束。

端面轴

8.1 功能

端面轴

在“车削”工艺中，端面轴指的是垂直于主轴的对称轴或纵轴 Z 运行的机床轴。

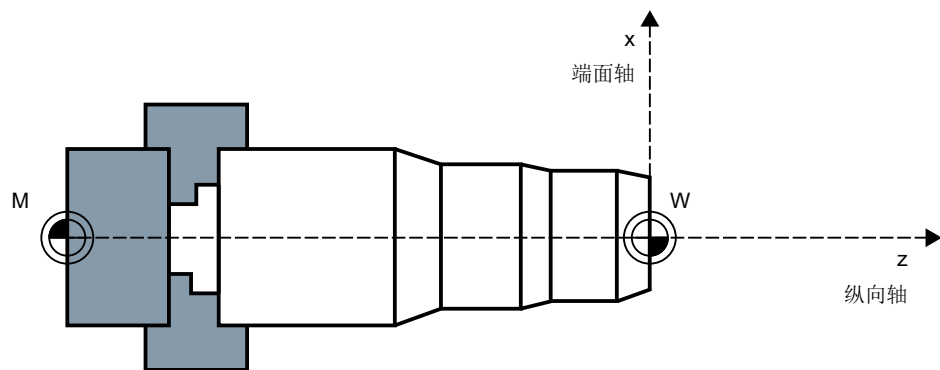


图 8-1 机床坐标系中端面轴的位置

几何轴作为端面轴

通道的每根几何轴均可定义为端面轴。

为了将几何轴作为端面轴，可同时或单独使能及激活下列功能：

- 以直径编程和显示
- 恒定切削速度 G96/G961/G962 的参考轴

通道中的多个端面轴

在通道中引入多个端面轴时，直径编程和 G96/G961/G962 的参考轴功能会解耦。直径编程和 G96/G961/G962 的参考轴功能可对不同的端面轴生效（见下表）。

	以直径 编程和显示			G96 / G961 / G962 的参考轴	
允许的轴类型：	几何轴		直线通道轴	几何轴	
在通道中选择：	一	m 中选 3	m 中选 n	一	3 中选一

8.1 功能

	以直径编程和显示		G96 / G961 / G962 的参考轴	
特殊作用：	通道	轴	通道	
机床数据：	MD20100	MD30460	MD20100	
编程：	DIAM* 针对通道模态生效 G 功能组 29		SCC [<轴>] 针对通道模态生效 G96/G961/G962 的参考轴	
在跨通道取轴时接收：	DIAM*A [<轴>] 针对轴模态生效			
针对轴非模态生效的直径/半径编程	DAC、DIC； RAC、RIC 针对轴逐段生效，仅编程			
DIAM*： DIAMOF, DIAMON, DIAM90, DIAMCYCOF DIAM*A[<轴>]:DIAMOFA[<轴>]、DIAMONA[<轴>]、DIAM90A[<轴>]、DIACYCOFA[<轴>]、DIAMCHANA[<轴>] <轴>： 几何轴/通道轴或机床轴的轴名称				

说明

回转轴不允许作为端面轴。

基于直径的数据

DIAMON/DIAMONA [<轴>]

通过 DIAMON/DIAMONA [<轴>] 激活直径编程后（参见“编程 (页 274)”），以下数据基于直径设定：

- 工件坐标系中端面轴的显示数据：
 - 设定位置 and 实际位置
 - 剩余行程
 - REPOS 偏移
- “JOG”运行方式：
 - 通过手轮进行增量手动运行时的步进尺寸（INC）（取决于生效的 MD）

- 零件程序编程：
 - 终点位置，与参考模式（G90 / G91）无关
 - 圆弧编程（G2 / G3）的插补参数，若其采用零件程序指令 AC 绝对编写。
- 基于工件坐标系（WCS）读取的实际值：
 - \$AA_MW[<端面轴>]
测量功能 MEAS（删除剩余行程的测量）和 MEAW（不删除剩余行程的测量）的系统变量
 - \$P_EP[<端面轴>]
 - \$AA_IW[<端面轴>]

DIAM90/DIAM90A[<轴>]

通过 DIAM90/DIAM90A[<轴>] 激活取决于参考模式的直径编程后，以下数据始终基于直径显示，与参考模式（G90 / G91）无关：

- 实际值
- 基于工件坐标系（WCS）读取的实际值：
 - \$AA_MW[<端面轴>]
测量功能 MEAS（删除剩余行程的测量）和 MEAW（不删除剩余行程的测量）的系统变量
 - \$P_EP[<端面轴>]
 - \$AA_IW[<端面轴>]

DIAMCYCOF/DIACYCOFA[<轴>]

编写 DIAMCYCOF 或 DIACYCOFA[<轴>] 时控制系统内部会切换至半径编程。

恒定基于半径的数据

对于端面轴，下列数据**始终**基于半径输入、编写和显示：

- 零件程序编程
 - CIP 圆弧编程的插补参数
- 偏移
 - 刀具补偿
 - 可编程框架和可设定框架
 - 外部零点偏移
 - DRF 偏移和预设偏移
 - 等
- 工作区域限制

8.2 参数设置

- 软件限位开关
- 进给率
- 基于机床坐标系的显示数据
- 轴、VSA 和 HSA 的维护图的显示数据

恒定基于半径数据的功能扩展：

对于 PLC 轴（通过 SINU_TraversePosAxis (Basic Program - FC18)）或只受 PLC 控制的轴：

- 对 PLC 轴的半径尺寸设定同样适用于带直径功能的多端面轴，且独立于通道专用或轴专用直径编程。
- 在 JOG (Inc) 运行方式下，PLC 轴归属于通道状态。若直径编程生效且 MD20624 \$MC_HANDWH_CHAN_STOP_COND 位 15 = 0，那么将只运行设定增量的一半行程。

8.2 参数设置

将几何轴定义为端面轴

通过以下机床数据将一根几何轴定义为通道中的端面轴：

MD20100 \$MC_DIAMETER_AX_DEF（带端面轴功能的几何轴）

示例

MD20100 \$MC_DIAMETER_AX_DEF="X"；几何轴 X 是通道中的端面轴。

对于此轴，即可采用直径编程，也可通过 G96/G961/G962 指定恒定切削速度。

通道中的多个端面轴

通过以下机床数据定义更多端面轴，这些轴可启用轴专用直径编程功能：

MD30460 BASE_FUNCTION_MASK，位 2 = <值>

位	值	含义
2	0	不允许轴专用的直径编程。
	1	允许轴专用的直径编程。

说明

MD30460 位 2 = 1 这一设置只适用于直线轴。

边界条件

在一个通道中只允许同时对一根端面轴进行通道专用（MD20100）和轴专用（MD30460，位 2）定义。通道专用的机床数据具有更高的优先级。

通过 MD20100 可为启动中的端面轴分配以下通道专用的功能

NC 启动时，端面轴可以分配以下轴专用的初始设置：“接收直径编程的通道状态”

DIAMCHANA [<轴>]

轴专用的直径编程使能时（MD30460，位 2 = 1），可在用户侧使用以下轴专用的指令：

- DIAMONA（轴专用的模态直径编程：激活）
- DIAMOFA（轴专用的模态直径编程：取消）
- DIAM90A（G90 和 AC：轴专用的模态的直径编程；G91 和 IC：半径编程）
- DIACYCOFA（轴专用的模态直径编程：循环中取消）
- DIAMCHANA（接收直径编程的通道状态）

刀具参数的尺寸设定

通过以下通道专用的机床数据可为通道中定义的所有端面轴将以下刀具参数激活作为直径值：

MD20360 \$MC_TOOL_PARAMETER_DEF_MASK, <位> = <值>

MD20360 \$MC_TOOL_PARAMETER_DEF_MASK		
位	值	含义
1	1	端面轴刀具长度作为直径
2	1	磨损或刀具长度作为直径时报警，以及平面切换
3 ¹⁾	0	零点偏移 \$P_EXTFRAME 和框架 对于端面轴，框架中的零点偏移总是作为半径值纳入计算。
	1	端面轴中的框架中零点偏移（NV）作为直径 框架内部将零点偏移作为半径值保存。在直径编程和半径编程间切换时不会进行换算。
4	1	预设值作为直径

8.2 参数设置

MD20360 \$MC_TOOL_PARAMETER_DEF_MASK		
位	值	含义
5 ¹⁾	0	外部零点偏移（轴叠加） 对于端面轴，外部零点偏移总是作为半径值纳入计算。
	1	端面轴中的外部零点偏移（NV）作为直径 在直径编程和半径编程间切换时不会进行换算。
6	1	端面轴实际值作为直径
7	1	端面轴实际值作为实际值显示
8 ¹⁾	1	WCS 中总是以半径显示剩余行程
9 ¹⁾	针对所有由手轮指定行程（MD11346 \$MN_HANDWH_TRUE_DISTANCE == 1）的端面轴有以下规定：	
	0	若通道专用或轴专用直径编程对该轴生效，那么运行的行程将为设定手轮增量的一半。
	1	运行行程始终为设定手轮增量的一半。
10	1	无刀具生效时，生效的可定向刀架的刀具分量
11	1	\$TC_DP6 的分析作为直径
12	1	\$TC_DP15 的分析作为刀具直径磨损
13 ¹⁾	1	在沿圆弧点动时，圆心坐标始终为半径值，参见 SD42690 \$SC_JOG_CIRCLE_CENTRE
14 ¹⁾	1	对于循环掩码，端面轴的绝对值总是作为半径。
15	1	循环掩码中端面轴的增量值作为直径
1) 该功能取决于 MD20100 \$MC_DIAMETER_AX_DEF 和 MD30460 \$MA_BASE_FUNCTION_MASK，位 2 中的设置		

通过手轮进行手动运行时的特性

直径编程生效时，端面轴在通过手轮进行手动运行（MD11346 \$MN_HANDWH_TRUE_DISTANCE == 1 或 3）时的特性通过以下机床数据调整：

MD20624 \$MC_HANDWH_CHAN_STOP_COND，位 15 = <值>

位	值	含义
15	0	只运行设定增量的一半。
	1	完全运行设定的增量。

以直径显示位置值

设置了

MD27100 \$MC_ABSBLOCK_FUNCTION_MASK 位 0 = 1 时，端面轴的位置值总是作为直径值显示。

启动、复位后的通道专用初始设置

启动、通道复位或零件程序结束后，G 功能组 29（DIAMON、DIAM90、DIAMOF、DIAMCYCOF）的通道专用初始设置通过以下机床数据 MD20110 \$MC_RESET_MODE_MASK 定义。

MD20110 \$MC_RESET_MODE_MASK，位 0 = 0:

- 启动：符合 MD20150 \$MC_GCODE_RESET_VALUE 的初始设置
- 通道复位或零件程序结束：符合 MD20150 \$MC_GCODE_RESET_VALUES 的初始设置

MD20110 \$MC_RESET_MODE_MASK，位 0 = 1:

- 启动后：符合 MD20150 \$MC_GCODE_RESET_VALUE 的初始设置
- 通道复位或零件程序结束后：符合 MD20152 \$MC_GCODE_RESET_MODE 的初始设置

用户可通过事件控制的程序调用（ProgEvent）设置所需的状态。

若 G96/G961/G962 为启动后的初始设置，则必须通过 MD20100 \$MC_DIAMETER_AX_DEF 定义一根端面轴，否则会触发报警 10870。

如要在复位、零件程序结束或零件程序开始时保留 G96 / G961 / G962 的参考轴，则需进行如下设置：

- 通道复位或零件程序结束：MD20110 \$MC_RESET_MODE_MASK，位 18 = 1
- 零件程序开始：MD20112 \$MC_START_MODE_MASK，位 18 = 1

也可不设置端面轴，在 MD20100 中通过 SCC[AX] 指定 G96 / G961 / G962 的参考轴。在此情形下不允许通过 G96 激活恒定切削速度。更多信息参见：

更多信息：编程手册之 NC 编程

8.3 编程

端面轴既可基于直径，也可基于半径编写。通常情况下采用直径编程，即使用双倍的行程值编写，从而可在零件程序中直接取用技术图纸中的对应尺寸值。

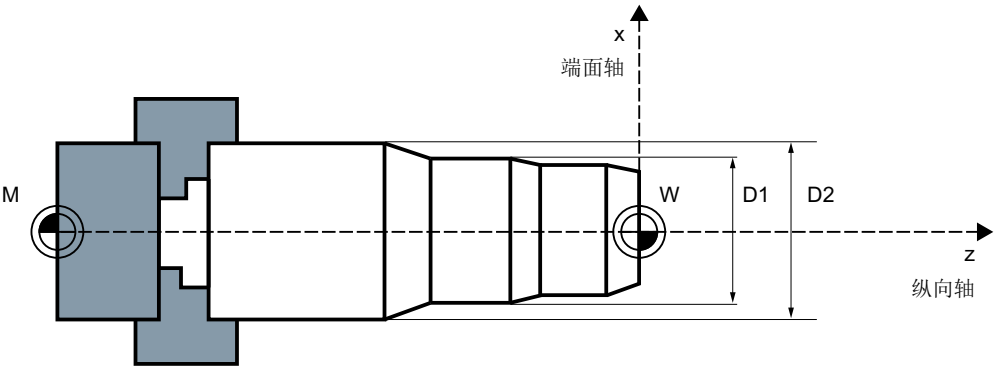


图 8-2 端面轴采用直径设定（D1、D2）

激活/取消直径编程

通道专用的直径编程

直径编程通过 G 功能组 29 中的模态生效的零件程序指令激活或取消：

- DIAMON：直径编程 ON
- DIAMOF：直径编程 OFF 或半径编程 ON
- DIAM90：直径编程或半径编程取决于参考模式：
 - 直径编程 ON，配合绝对尺寸设定 G90
 - 半径编程 ON，配合增量尺寸设定 G91
- DIAMCYCOF：G90 和 G91 的半径编程 ON

只以通道的端面轴为基准。

针对通道内多个端面轴的轴专用直径编程

说明

额外指定的轴必须通过 MD30460 \$MA_BASE_FUNCTION_MASK 位 2 = 1 激活。
指定的轴必须是通道内已知的轴。允许采用几何轴、通道轴或机床轴。
不允许在同步动作中编程。

下列轴专用模态生效指令可在一个零件程序段中多次编写：

- DIAMONA[<轴>]：G90、G91、AC 和 IC 的直径编程 ON
- DIAMOFA[<轴>]：直径编程 OFF 或半径编程 ON
- DIAM90A[<轴>]：直径编程或半径编程取决于参考模式：
 - 直径编程 ON，配合绝对尺寸设定 G90 和 AC
 - 半径编程 ON，配合增量尺寸设定 G91 和 IC
- DIACYCOFA[轴]：G90 和 G91 的半径编程 ON
- DIAMCHANA[轴]：接收直径编程通道状态
- DIAMCHAN：所有轴通过 MD30460 位 2 = 1 接收直径编程通道状态

相比通道设置，轴专用模态生效指令具有更高优先级。

8.4 前提条件

跨通道取轴

基于来自零件程序的 GET 请求，在通过 RELEASE[轴] 进行跨通道取轴时，系统会在新通道中为附加端面轴接收直径编程状态。

在同步动作中取轴

在同步动作中跨通道取轴时，若为下列情形，那么端面轴的轴专用直径编程状态将带入新通道：

- 通过 MD30460 位 2 = 1 为端面轴使能了轴专用直径编程。
- 在让渡轴的通道中，未将端面轴划归至通道专用直径编程下。

通过系统变量 \$AA_DIAM_STAT[<轴>] 可查询生效的尺寸设定。

8.5 示例

示例 1

X 是通过 MD20100 \$MC_DIAMETER_AX_DEF 定义的端面轴。

Y 为几何轴，U 为辅助轴。这两根轴同时又在 MD30460 \$MA_BASE_FUNCTION_MASK 中通过位 2 = 1 定义为基于直径设定的端面轴。

DIAMON 在启动后不生效。

程序代码	注释
N10 G0 G90 X100 Y50	; 无直径编程生效
N20 DIAMON	; 通道专用直径编程, 对 X 生效
N30 Y200 X200	; 尺寸设定: X 以直径, Y 以半径
N40 DIAMONA[Y]	; 轴专用模态生效直径编程, ; 对 Y 生效
N50 Y250 X300	; 尺寸设定: X 和 Y 以直径
N60 DIAM90	; 尺寸设定: X G90/AC 以直径, G91/IC 以半径
N70 Y200	; Y: 继续轴专用模态生效直径编程
N75 G91 Y20 U=DIC(40)	; 尺寸设定: Y 以直径, U 以直径逐段 IC
N80 X50 Y100	; 尺寸设定: X 以半径 (G91), Y 以直径
N85 G90 X100 U200	; 尺寸设定: X 以直径, U 以半径
N90 DIAMCHANA[Y]	; Y 接收通道状态 DIAM90
N95 G91 X100 Y100	; 尺寸设定: X 和 Y 以半径 (G91)
N100 G90 X200 Y200	; 尺寸设定: X 和 Y 以直径

示例 2

和上面的示例一样，端面轴设定基于直径。

X 和 Y 处于通道 1 中，且在通道 2 中已知，即支持跨通道取轴。

程序代码	注释
通道 1	
N10 G0 G90 X100 Y50	; 无直径编程生效
N20 DIAMON	; 通道专用直径编程, 对 X 生效
N30 Y200 X200	; 尺寸设定: X 以直径, Y 以半径
N40 DIAMONA[Y]	; Y 启用轴专用模态生效直径编程
N50 Y250 X300	; 尺寸设定: X 和 Y 以直径
N60 SETM(1)	; 同步标记 1
N70 WAIT(1,2)	; 等待通道 2 中的同步标记 1
通道 2	
...	
N50 DIAMOF	; 通道 2 无直径编程生效
...	
N100 WAIT(1,1)	; 等待通道 1 中的同步标记 1
N110 GETD(Y)	; 直接取 Y 轴
N120 Y100	; Y 划归于通道 2 中的通道专用直径编程, ; 即尺寸设定基于半径

定位轴

9.1 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。

9.1 简要说明

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” 选件

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” 选件提供辅助轴功能，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。
更多信息
功能手册之基本功能：连续轨迹控制运行、准停、预读
- 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 NC 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。

9.2 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 (页 279)”章节）。
- 与加工在同一个零件程序中编写（参见“定位轴 (页 281)”章节）。
- 只由 PLC 在加工期间触发（参见“并行定位轴 (页 283)”章节）。

9.2.1 独立通道

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：
 - <Chan>.basic.out.ncStart（NC 启动）
 - <Chan>.basic.out.ncStop（NC 停止）
 - <Chan>.basic.out.reset（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 指令
 - 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

更多信息

通道功能的更多说明参见：
功能手册之基本功能；运行方式组、通道、程序运行、复位特性

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Chan>.basic.out.ncStart	LBP_Chan*.A_NCStart	DB21, ... DBX7.1
<Chan>.basic.out.ncStop	LBP_Chan*.A_NCStop	DB21, ... DBX7.3
<Chan>.basic.out.reset	LBP_Chan*.A_Reset	DB21, ... DBX7.7

9.2.2 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 284)”章节）。

定位轴类型和程序段转换

程序段转换时间点取决于所编写的定位轴类型（另见“程序段切换 (页 297)”章节）：

类型	说明
1	当所有轨迹轴和所有定位轴到达编写的终点时，执行程序段转换。
2	当所有轨迹轴到达编写的终点时，执行程序段转换。 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转换条件时，可在单轴插补的制动斜坡内执行程序段转换。

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转换的运动结束条件

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转换。

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 轴的程序段转换。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
- 每根定位轴一个单独的进给倍率
- 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：
 - <Chan>.basic.out.ncStart (NC 启动)
 - <Chan>.basic.out.ncStop (NC 停止)
 - <Chan>.basic.out.reset (复位)
 - <Chan>.basic.out.disableReadIn (读入禁止)
- 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 指令组 1（模态运行指令，包含 G0、G1、G2...）不适用于定位轴。

更多信息

编程手册之 NC 编程

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转换，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。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Chan>.basic.out.ncStart	LBP_Chan*.A_NCStart	DB21, ...DBX7.1
<Chan>.basic.out.ncStop	LBP_Chan*.A_NCStop	DB21, ...DBX7.3
<Chan>.basic.out.reset	LBP_Chan*.A_Reset	DB21, ...DBX7.7
<Chan>.basic.out.disableReadIn	LBP_Chan*.A_Rldisable	DB21, ...DBX6.1

9.2.3 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。

9.3 运动特性和插补过程

- 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

由 PLC 通过 SINU_TraversePosAxis (Basic Program - FC18) 触发并行定位轴。

- 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

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

9.3 运动特性和插补过程

9.3.1 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。

9.3.2 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 NC 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 → NC: 请求接收轴的控制权
`<Axis>.spindle.out.plcCtrl = 1`（PLC 控制轴）
2. NC: 检查轴是否是主处理轴或中立轴。

9.3 运动特性和插补过程

3. NC: 检查另一根轴是否允许由 PLC 控制。
4. NC 确认控制权转交:
 - `<Axis>.plcCtrl.in.active = 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。

- 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

过程说明:

1. PLC → NC: PLC 将轴的控制权交还给 NC
`<Axis>.spindle.out.plcCtrl = 0` (PLC 控制轴)
2. NC: 检查是否有未处理的轴报警。
3. NC: 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“停止进给轴/主轴”用轴停止来停止运动。
4. NC: 按照使用案例 4“复位进给轴/主轴”执行轴复位。
5. NC 确认控制权转交:
 - `<Axis>.plcCtrl.in.resetDone = 0` (复位已执行)
 - `<Axis>.plcCtrl.in.active = 0` (PLC 控制轴)
 - `<Axis>.plcCtrl.in.stopped = 0` (轴停止生效)
 - 系统变量 `$AA_SINGLAX_STAT = 0`

结果: NC 获得进给轴/主轴的控制权。

可能出现的情形

以下情形下，NC 确认控制权转交，但系统内部会针对进给轴/主轴将通道状态设置为“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，NC 才会确认获得轴/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：

- `<Chan>.basic.out.deleteDistanceToGo`（删除剩余行程）
- `<Axis>.spindle.out.plcCtrl`（复位）
- `<Axis>.spindle.out.resume`（继续）
- `<Axis>.spindle.out.stop`（沿制动斜坡停止）

自控单轴运动中 NC/PLC 接口上 NC 和 PLC 间的信号流参见“通过 PLC 控制 (页 305)”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。

`<Chan>.basic.out.deleteDistanceToGo = 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

9.3 运动特性和插补过程

- 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。

过程说明：

- PLC → NC：请求停止进给轴/主轴
`<Axis>.spindle.out.stop = 1`（沿制动斜坡停止）
- NC：沿斜坡制动轴。
- NC 确认执行：
 - `<Axis>.basic.in.posCoarseReached = 1`（粗准停）
 - `<Axis>.basic.in.posFineReached = 1`（精准停）
 - `<Axis>.plcCtrl.in.stopped = 0`（轴停止生效）
 - `<Axis>.basic.in.traversMinusCommandActive /`
`<Axis>.basic.in.traversPlusCommandActive = 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。

更多信息

扩展的停止和退回 (页 953)

使用案例 3：继续进给轴/主轴

需要继续执行被使用案例 2“停止轴”中断的运行。

过程说明：

- PLC → NC：继续轴
`<Axis>.spindle.out.resume = 1`（继续）
- NC：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。

- NC: 检查轴是否能够继续运行? 如果是, 轴将转换至“生效”状态。
- NC 确认执行:
 - <Axis>.basic.in.posCoarseReached = 0 (粗准停)
 - <Axis>.basic.in.posFineReached = 0 (精准停)
 - <Axis>.plcCtrl.in.stopped = 0 (轴停止生效)
 - <Axis>.basic.in.traversMinusCommandActive /
 <Axis>.basic.in.traversPlusCommandActive = 0 (运行指令 -/+)
 - 轴状态: 系统变量 \$AA_SINGLAX_STAT == 4 (生效)。

结果: 进给轴/主轴继续运行。

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)

过程说明:

- PLC → NC: 请求复位进给轴/主轴
 <Axis>.spindle.out.plcCtrl = 1 (复位)
- NC: 将进给轴/主轴转换至“停止”状态。
- NC: 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。
- NC: 复位进给轴/主轴的内部状态。
- NC: 复位生效的轴专用机床数据生效。

说明

采用通道复位时, 对于由 PLC 控制的轴而言, 轴专用机床数据不生效。

- NC 确认执行:
 - <Axis>.plcCtrl.in.resetDone = 1 (复位已执行)
 - <Axis>.plcCtrl.in.stopped = 0 (轴停止生效)
 - 系统变量 \$AA_SINGLAX_STAT = 1
- NC 结束此进程。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Chan>.basic.out.deleteDistanceToGo	LBP_Chan*.A_DeleteDTG	DB21, ...DBX6.2
<Axis>.spindle.out.plcCtrl	LBP_Axis*.A_PLCAxis	DB31, ...DBX28.1
<Axis>.spindle.out.resume	LBP_Axis*.A_AxResume	DB31, ...DBX28.2
<Axis>.spindle.out.stop	LBP_Axis*.A_Stop	DB31, ...DBX30.0
<Axis>.plcCtrl.out.requestPlcCtrl	LBP_Axis*.A_PLCAxis	DB31, ...DBX28.7

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.posCoarseReached	LBP_Axis*.E_ExactCoarse	DB31, ...DBX60.6
<Axis>.basic.in.posFineReached	LBP_Axis*.E_ExactFine	DB31, ...DBX60.7
<Axis>.plcCtrl.in.resetDone	LBP_Axis*.E_AxReset	DB31, ...DBX63.0
<Axis>.plcCtrl.in.active	LBP_Axis*.E_PLCCtrlAx	DB31, ...DBX63.1
<Axis>.plcCtrl.in.stopped	LBP_Axis*.E_AxStop	DB31, ...DBX63.2
<Axis>.basic.in.traversMinusCommandActive	LBP_Axis*.E_TCMinus	DB31, ...DBX64.6
<Axis>.basic.in.traversPlusCommandActive	LBP_Axis*.E_TCPlus	DB31, ...DBX64.7

9.3.3 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。

更多信息

轴耦合 (页 571)

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

...

9.4 定位轴动态响应

```
$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

9.4 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 284)”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相互独立的进给倍率。每根定位轴均可通过专用的轴进给倍率进行调整。

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 DYN NORM 中轨迹运动中的最大单轴加速度）
- MD32300 \$MA_MAX_AX_ACCEL [1]（动态模式 DYN POS 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	wirksame maximale axiale Beschleunigung:MD32300 \$MA_MAX_AX_ACCEL[0]
1	wirksame maximale axiale Beschleunigung: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 DYNNORM 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 Bei aktivem G75 (Festpunkt anfahren):MD32431 \$MA_MAX_AX_JERK[0]
1	MD32431 \$MA_MAX_AX_JERK[1]

9.5 编程

9.5.1 简介

说明

定位轴的编程请参见以下文档：

更多信息

参见进给率 (页 201)和主轴 (页 395)

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。

更多信息

功能手册之同步动作

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>]

9.5 编程

```
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	

程序代码	注释
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。

9.5.2 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。
-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。

9.6 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>]

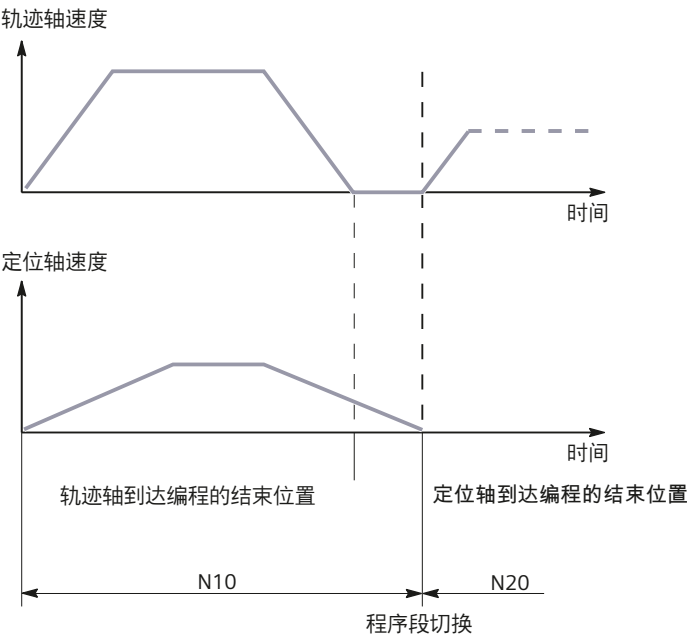


图 9-1 采用轨迹轴和定位轴类型 1 时的程序段切换

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。

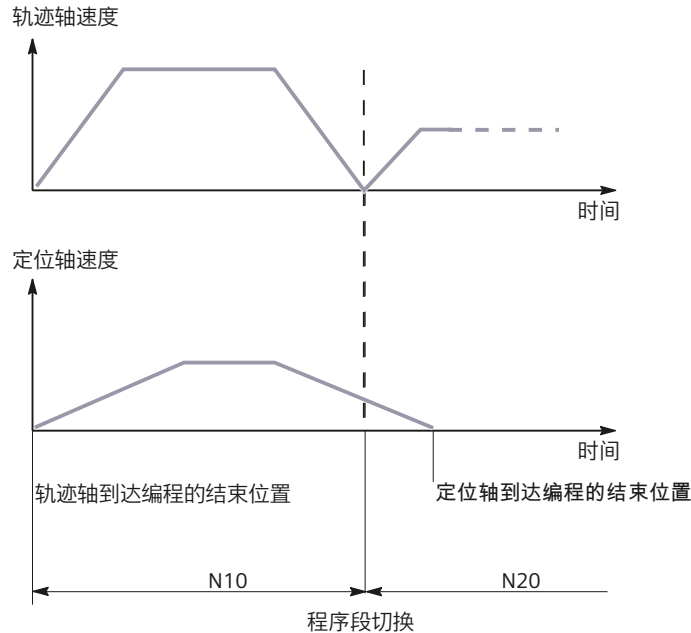


图 9-2 采用轨迹轴和定位轴类型 2 时的程序段切换

9.6.1 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>])；自身程序段
 - N(x+1) POS[<轴>]

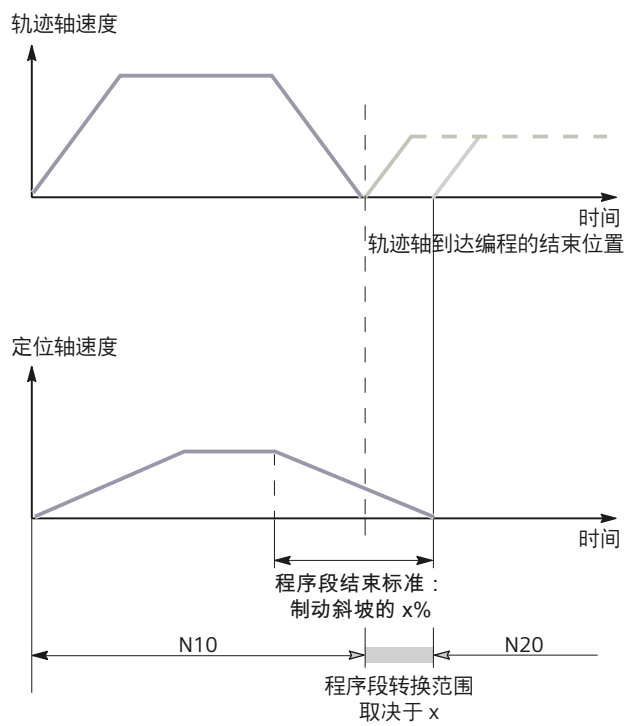


图 9-3 采用轨迹轴和定位轴类型 3 时的程序段切换

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。

附加程序段转换条件：“公差窗口” (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。

更多信息：参数手册 2

说明

可编程运动结束条件 FINEA、COARESA、IPOENDA 的更多相关信息参见章节：

- 运行方式 (页 396)
- 单轴动态响应的控制 (页 259)

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

有关定位轴编程的更多信息参见编程手册之 NC 编程。

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
N10 POS[X]=100	; 缺省设定生效。 ; X 轴定位到位置 100。 ; 程序段切换条件：“精准停”

程序代码	注释
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，就立即转换程序段。
N40 POS[X]=250	;X 轴并未停止在 200 位置，而是继续运行到位置 250。一旦轴开始制动，就立即转换程序段。
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”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 轴并未停止在 200 位置，而是继续运行到位置 250。一旦 X 轴开始制动，就立即进行工艺循环中的程序段转换。
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
	;缺省设定生效。
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 = 额定位置，公差 = 0.5
N30 POS[X]=200	;一旦 X 轴开始制动，就立即转换程序段。
N40 POS[X]=250	;X 轴并未停止在 200 位置，而是继续运行到位置 250。一旦轴开始制动，就立即转换程序段。
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	
N70 M30	

9.6 程序段切换

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”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 轴并未停止在 200 位置，而是继续运行到位置 250。一旦 X 轴开始制动以及 X 轴的位置 >= 249.7，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	

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

9.6.2 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

程序代码	注释
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。

9.7 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。

更多信息
 PLC 功能手册

并行定位轴

由 PLC 使用 SINU_TraversePosAxis (Basic Program - FC18) 启动并行定位轴。

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：

- 接口信号 <Chan>.basic.out.pathFeedrateOvrFactor（“进给倍率”）
- 接口信号 <Chan>.basic.out.deleteDistanceToGo（“删除剩余行程”）

轴专用信号

定位轴还有以下信号：

- 接口信号 `<Axis>.basic.in.isPosAxis`（“定位轴”）
- 定位轴/主轴的进给率（FA）
- 轴专用接口信号 `<Axis>.basic.out.feedrateOvrFactor`（“进给倍率”）
- 轴专用接口信号 `<Axis>.basic.out.resetMovement`（“剩余行程/主轴复位”）删除剩余行程

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：
`<Chan>.basic.out.disableFeed`（“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：
`<Chan>.basic.in.allAxesStationary`（“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 `<Chan>.basic.out.disableFeed`（“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 `<Chan>.basic.in.allAxesStationary`（“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 NC→PLC <Axis>.plcCtrl.in 将待取轴的类型信息提供给 PLC（另见功能手册“基本功能”，章节“K5：跨通道程序协调和逐通道试运行”）：

- 接口信号 <Axis>.plcCtrl.in.assignedChanNumber（“通道 NC 进给轴/主轴”）通道 1 至 10
- 接口信号 <Axis>.plcCtrl.in.newTypeRequested（“PLC 请求新类型”）
- 接口信号 <Axis>.plcCtrl.in.changeoverPossible（“可跨通道取轴”）
- 接口信号 <Axis>.plcCtrl.in.neutralModeActive（“中性轴/主轴”）
- 接口信号 <Axis>.plcCtrl.in.assignedToPlc（“PLC 轴/主轴”）

若接口信号 <Axis>.plcCtrl.in.changeoverPossible（“可跨通道取轴”）= 1，则可通过 PLC 发出取轴请求。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.feedrateOvrFactor	LBP_Axis*.A_FD_OR	DB31, ...DBB0
<Axis>.basic.out.resetMovement	LBP_Axis*.A_DeIDTGSReset	DB31, ...DBX2.2
<Chan>.basic.out.pathFeedrateOvrFactor	LBP_Chan*.A_FD_OR	DB21, ...DBB4
<Chan>.basic.out.disableFeed	LBP_Chan*.A_FDdisable	DB21, ...DBX6.0
<Chan>.basic.out.deleteDistanceToGo	LBP_Chan*.A_DeleteDTG	DB21, ...DBX6.2

NC → PLC

Basic Program Plus	Basic Program	
<Chan>.basic.in.allAxesStationary	LBP_Chan*.E_AxesStop	DB21, ...DBX36.3
<Axis>.basic.in.posCoarseReached	LBP_Axis*.E_ExactCoarse	DB31, ...DBX60.6
<Axis>.basic.in.posFineReached	LBP_Axis*.E_ExactFine	DB31, ...DBX60.7
<Axis>.plcCtrl.in.assignedChanNumber	LBP_Axis*.E_NCASpChanA..D	DB31, ...DBX68.0.. 3
<Axis>.plcCtrl.in.newTypeRequested	LBP_Axis*.E_PLCType	DB31, ...DBX68.4
<Axis>.plcCtrl.in.changeoverPossible	LBP_Axis*.E_ChPoss	DB31, ...DBX68.5
<Axis>.plcCtrl.in.neutralModeActive	LBP_Axis*.E_NeutralASp	DB31, ...DBX68.6

Basic Program Plus	Basic Program	
<Axis>.plcCtrl.in.assignedToPlc	LBP_Axis*.E_PLCAp	DB31, ...DBX68.7
<Axis>.basic.in.isPosAxis	LBP_Axis*.E_PosAxis	DB31, ...DBX76.5

9.7.1 由 PLC 启动并行定位轴

使用 SINU_TraversePosAxis (Basic Program - FC18) 从 PLC 用户程序出发移动 NC 中的一个轴作为并行定位轴：

下列功能此时采用固定设定：

- 线性插补（G01）
- 进给率，单位：毫米/分钟或度/分钟（G94）
- 准停（G09）
- 生效的可设定零点偏移被计算在内

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。

更多信息

PLC 功能手册；块说明 SINU_TraversePosAxis

9.7.2 由 PLC 控制的轴

PLC 动作

针对机床轴，对比以下 PLC 动作与 NC 响应：

- 通过 SINU_TraversePosAxis (Basic Program - FC18) 启动作为 PLC 轴的机床轴
- 触发 NC 启动或 NC 停止
- 触发轴专用停止、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

NC 响应示例

下表对比 PLC 动作与 NC 响应。

PLC 动作	NC 响应
机床轴 AX1 为通道 1 中的通道轴，作为 PLC 轴通过 SINU_TraversePosAxis (Basic Program - FC18) 启动	
触发进给轴和主轴的 NC 停止 <Chan>.basic.out.ncStopAxesAndSpindles = 1 (进给轴和主轴 NC 停止)	AX1 停止。
触发 <Chan>.basic.out.ncStart = 1 (NC 启动)	AX1 继续运行。
PLC 将控制 AX1， <Axis>.spindle.out.plcCtrl = 1 (PLC 控制轴)	将 AX1 的控制权转交给 PLC。<Axis>.plcCtrl.in.active = 1 (PLC 控制轴)
触发进给轴和主轴的 NC 停止 <Chan>.basic.out.ncStopAxesAndSpindles = 1 (“进给轴和主轴 NC 停止”)	AX1 不停止。
触发轴停止 <Axis>.spindle.out.stop = 1 (沿制动斜坡停止)	AX1 停止 <Axis>.plcCtrl.in.stopped == 1 (轴停止生效)
触发轴继续 <Axis>.spindle.out.resume = 1 (继续)	AX1 继续运行 <Axis>.plcCtrl.in.stopped == 0 (轴停止生效)
触发 NC 复位 触发 <Chan>.basic.out.reset = 1 (复位)	对 AX1 无影响
触发轴复位 <Axis>.spindle.out.plcCtrl = 1 (复位)	AX1 停止，运行终止： • <Axis>.plcCtrl.in.stopped = 0 (轴停止生效) • 读取轴专用机床数据 • <Axis>.plcCtrl.in.resetDone = 1 (复位已执行) • <Axis>.plcCtrl.in.stopped = 0 (轴停止生效)

9.7 通过 PLC 控制

PLC 动作	NC 响应
通过 SINU_TraversePosAxis (Basic Program - FC18) 启动作为 PLC 轴的机床轴 AX1	<Axis>.plcCtrl.in.resetDone = 0 (复位已执行)
取消 AX1 的伺服使能: <Axis>.basic.out.enableCtrl = 0 (控制器使能)	显示报警 21612“轴 %1 测量系统切换”
触发轴继续 <Axis>.spindle.out.resume = 1 (继续)	- 清除报警 21612“轴 %1 测量系统切换” <Chan>.geoAxis1.in.traversPlusCommand = 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 <Axis>.basic.out.enableCtrl = 1 (伺服使能)	AX1 运行至编写的终点。
触发轴复位 <Axis>.spindle.out.plcCtrl = 1 (复位)	- AX1 停止 - 读取轴专用机床数据 <Axis>.plcCtrl.in.resetDone = 0 (复位)
从 <Axis>.spindle.out.plcCtrl = 0 (PLC 控制轴) 起 PLC 将 AX1 的控制权转交给 NC	- NC 接收机床轴的控制权 <Axis>.plcCtrl.in.active = 0 (PLC 控制轴) <Axis>.plcCtrl.in.resetDone = 0 (复位)

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.enableCtrl	LBP_Chan*.A_ContrEnable	DB31, ...DBX2.1
<Chan>.basic.out.deleteDistanceToGo	LBP_Chan*.A_DeleteDTG	DB21, ...DBX6.2
<Chan>.basic.out.ncStart	LBP_Chan*.A_NCStart	DB21, ...DBX7.1
<Chan>.basic.out.ncStopAxesAndSpindles	LBP_Chan*.A_NCStopASp	DB21, ...DBX7.4
<Chan>.basic.out.reset	LBP_Chan*.A_Reset	DB21, ...DBX7.7
<Axis>.spindle.out.plcCtrl	LBP_Axis*.A_PLCAxis	DB31, ...DBX28.1
<Axis>.spindle.out.resume	LBP_Axis*.A_AxResume	DB31, ...DBX28.2

Basic Program Plus	Basic Program	
<Axis>.spindle.out.stop	LBP_Axis*.A_Stop	DB31, ...DBX30.0
<Axis>.plcCtrl.out.requestPlcCtrl	LBP_Axis*.A_PLCAxis	DB31, ...DBX28.7

NC → PLC

Basic Program Plus	Basic Program	
<Chan>.geoAxis1.in.traversPlusCommand	LBP_Chan*.TCPlus	DB21, ...DBX40.7
<Axis>.basic.in.posCoarseReached	LBP_Axis*.E_ExactCoarse	DB31, ...DBX60.6
<Axis>.plcCtrl.in.active	LBP_Axis*.E_PLCCtrlAx	DB31, ...DBX63.1
<Axis>.plcCtrl.in.stopped	LBP_Axis*.E_AxStop	DB31, ...DBX63.2

9.7.3 PLC 轴的控制特性

通道复位时的特性、激活机床数据、程序段搜索和 MD30460

表格 9-1

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。
NEWCONF	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。

9.7 通过 PLC 控制

以下情形下的控制特性	由 PLC 控制的轴
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 <Chan>.basic.out.disableFeed （“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 <Chan>.basic.in.allAxesStationary （“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。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Chan>.basic.out.disableFeed	LBP_Axis*.A_FDdisable	DB21, ...DBX6.0

NC → PLC

Basic Program Plus	Basic Program	
<Chan>.basic.in.allAxesStationary	LBP_Axis*.E_AxesStop	DB21, ...DBX36.3

9.8 特殊功能下的特性

9.8.1 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

更多信息

进给率 (页 201)

9.8.2 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。

9.9 示例

9.9.1 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

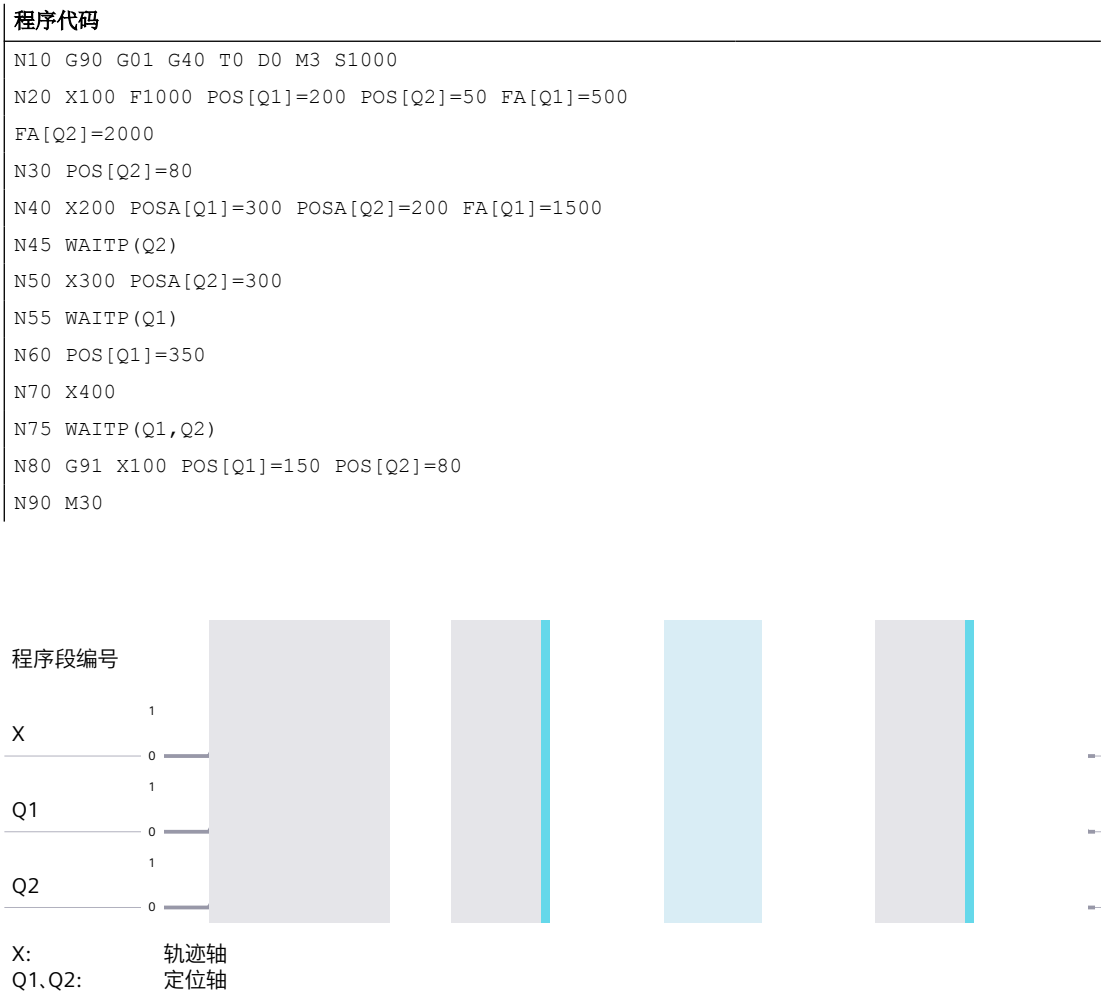


图 9-4 轨迹轴和定位轴的时序图

回转轴

10.1 简要说明

回转轴的典型应用有：

- 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

在数控机床上，直线轴、回转轴的名称及运动方向依据 DIN 66025 如下定义：

- 直线轴：X、Y、Z
- 回转轴：A、B、C

10.1 简要说明

- 分配：A 围绕 X 旋转，B 围绕 Y 旋转，C 围绕 Z 旋转
- 旋转方向：顺着直线轴的正方向看去，顺时针旋转方向为回转轴的正旋转方向。

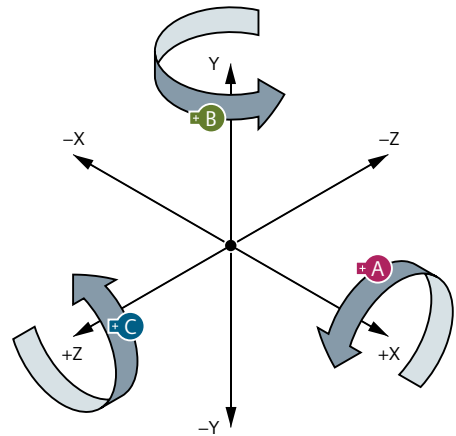


图 10-1 回转轴名称和运动方向

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	1) 转每分
角加速度	1) 转每秒 ²
角加加速度限制	1) 转每秒 ³
1) 轴为回转轴时，轴专用机床数据的单位。	

更多信息

速度、设定值/实际值系统、闭环控制 (页 141)

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[度每分]。

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 [比如毫米每分、英寸每分]。

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（度每分）和切线速度（毫米每分）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[毫米每分]

F_w = 角速度[度每分]

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

JOG 运行方式：旋转进给率

在 JOG 运行方式下，回转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)

10.2 360°模数

360° 模数回转轴是指其位置映射至 0° 至 359.999° 范围的回转轴。因此，360° 模数回转轴的运行范围原则上是没有界限的。360° 模数回转轴的运行范围限制（软件限位开关和工作区域限制）不生效。

下图用一条螺旋线来展示回转轴在正向的绝对位置。指示器在螺旋线上标记出了回转轴的绝对位置 (420°)。带模数圆弧的指示器的交点标记的是相应的 360° 模数位置 (60°)。

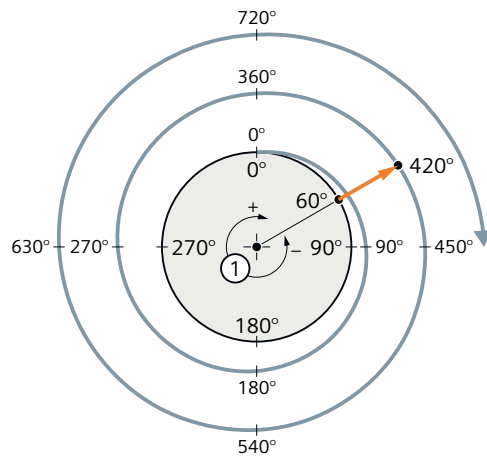


图 10-2 取模

参数设置

模数功能

回转轴的模数功能通过以下轴专用机床数据激活：

```
MD30310 $MA_ROT_IS_MODULO = 1
```

说明

建议在 360° 模数回转轴上同样设置模数 360° 位置显示。

模数范围

360° 模数回转轴的模数范围**必须**通过轴专用机床数据设为 360°：

```
MD30330 $MA_MODULO_RANGE = 360.0
```

360° 模数范围的起始位置

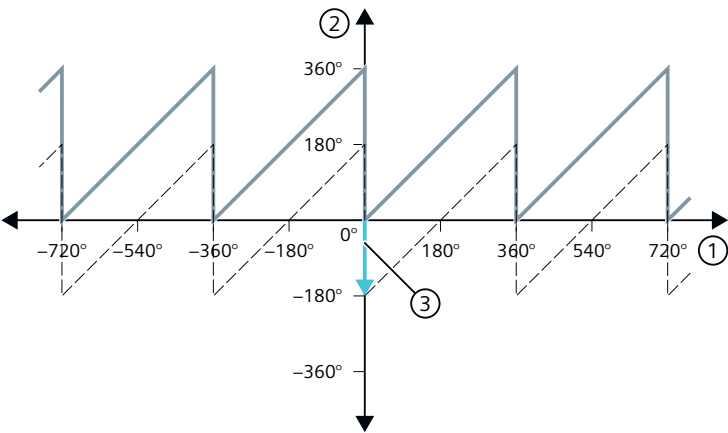
360° 模数范围的起始位置可通过以下轴专用机床数据设置：

```
MD30340 $MA_MODULO_RANGE_START = <起始位置>
```

设置的起始位置显示了模数功能在垂直方向上的移动。模数范围仍为 360°。

示例

- 标准起始位置：0° ⇒ 模数范围：0° - 360°
- 移动的起始位置：-180° ⇒ 模数范围：-180° 到 +180°



- ① 绝对位置
- ② 模数位置
- ③ 移动的起始位置 (-180°)

图 10-3 标准和移动的起始位置（红色）

说明**模数分度轴**

在 360° 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 360° 模数范围一样：

- MD30503 \$MA_INDEX_AX_OFFSET
- MD30340 \$MA_MODULO_RANGE_START

参见章节“分度轴”、“示例 3：模数回转轴为等距分度轴 (页 353)”和以下章节中的示例。

360° 模数回转轴：G90 下的定位特性（模态绝对位置编程）

通过机床数据设置 G90 下的缺省特性：

MD30455 \$MA_MISC_FUNCTION_MASK，位 2 = <值>

<值>	含义
0	采用缺省设置时，G90 编程下 360° 模数回转轴根据 AC 定位： • 目标位置 > 实际位置：沿正方向运行 • 目标位置 < 实际位置：沿负方向运行
1	采用缺省设置时，G90 编程下 360° 模数回转轴根据 DC 定位，即以最短行程。

位置显示设置为模数 360° 显示

通过以下轴专用机床数据将操作界面上对回转轴的位置显示设置为模数 360° 显示：

MD30320 \$MA_DISPLAY_IS_MODULO = 1

NC/PLC 接口信号**激活运行范围限制**

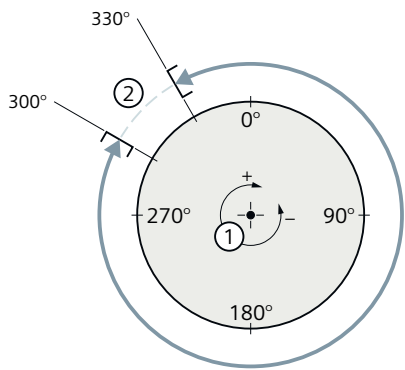
360° 模数回转轴的运行范围原则上是没有界限的。因此，缺省设置下，轴专用运行范围限制（如软件限位开关和工作区域限制）不生效。通过下列轴专用 NC/PLC 接口信号，必要时通过 NC 程序中的 M/H 功能触发，可激活 360° 模数回转轴的运行范围限制：

<Axis>.basic.out.enableModuloLimitSwitch = 1（360° 模数回转轴：激活运行范围限制）

前提条件:

- 360° 模数回转轴必须回参考点。
- 软件限位开关
需要监控的软件限位开关 1 或 2 必须处于激活状态:
 - <Axis>.basic.out.swLimitSwitch2Minus / <Axis>.basic.out.swLimitSwitch2Plus (激活)
 - MD36100 \$MA_POS_LIMIT_MINUS / MD36110 \$MA_POS_LIMIT_PLUS (软件限位开关 1)
 - MD36120 \$MA_POS_LIMIT_MINUS2 / MD36130 \$MA_POS_LIMIT_PLUS2 (软件限位开关 2)
- 工作区域限制
两个限制必须通过 G26/G25 或设定数据激活:
 - SD43400 \$SA_WORKAREA_PLUS_ENABLE / SD43410 \$SA_WORKAREA_MINUS_ENABLE (激活)
 - SD43420 \$SA_WORKAREA_LIMIT_PLUS / SD43430 \$SA_WORKAREA_LIMIT_MINUS (限制)
- 通过 STOPRE 同步
通过处于加工状态的 NC 程序中的 NC/PLC 接口信号激活运行范围限制，例如要将 M/H 指令发送至 PLC，则必须在下一个程序段中编程预处理停止 (STOPRE)。只有这样才能确保激活以下运行程序段中的运行范围限制。

注意
用户：激活和生效方式 如果在主运行程序段中，如通过 M/H 功能，通过 NC/PLC 接口信号 (<Axis>.basic.out.enableModuloLimitSwitch) 触发了激活 360°模数回转轴的运行范围限制，则可对以下运行程序段进行预加工且无需考虑运行范围限制。因此，用户有责任采取适当的措施，如在激活运行范围限制后在后续程序段中编程 STOPRE，使得运行范围限制从 NC 程序中的激活时间点起也可在所有以下程序段中生效。



- ① 旋转方向 +/-
- ② 软件限位开关 +/-

图 10-4 带软件限位开关的 360° 模数回转轴

激活运行范围限制的反馈信息

360° 模数回转轴的运行范围限制在 NC 中生效通过轴专用 NC/PLC 接口信号反馈：

<Axis>.basic.in.moduloLimitSwitchActive == 1（模数回转轴：运行范围限制有效）

示例：切换运行范围限制

在一根 360° 模数回转轴上，需要处理夹装了多个工件的托盘 1。之后通过旋转轴将托盘 1 更换为托盘 2。现在须对运行范围进行监控。

参数设置：

- MD30300 \$MA_IS_ROT_AX[AX5] = 1
- MD30310 \$MA_ROT_IS_MODULO[AX5] = 1
- MD36110 \$MA_POS_LIMIT_PLUS[AX5] = 340
- MD36100 \$MA_POS_LIMIT_MINUS[AX5] = 350

零件程序节选

程序代码	注释
M123	; 安装多夹具托盘 1
	; PLC: ;
	; 取消 B 轴的软件限位开关 => 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

10.3 回转轴的编程

程序代码	注释
...	
G0 Z540 B0	
M124	; 通过旋转轴安装托盘 2
	; PLC: ;
	; 激活 B 轴的软件限位开关 => DB35, DBX12.4=1
STOPRE	; 触发预处理停止
B270	

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.enableModuloLimitSwitch	LBP_Axis*.A_ModuloLimitEn	DB31, ...DBX12.4
<Axis>.basic.out.swLimitSwitch2Minus	LBP_Axis*.A_SWLimit2Minus	DB31, ...DBX12.2
<Axis>.basic.out.swLimitSwitch2Plus	LBP_Axis*.A_SWLimit2Plus	DB31, ...DBX12.3

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.moduloLimitSwitchActive	LBP_Axis*.E_ModLimEnAct	DB31, ...DBX74.4

10.3 回转轴的编程

10.3.1 简介

说明
关于编程的一般说明参见编程手册之 NC 编程。

MD30310

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，可通过轴专用机床数据：
MD30310 \$MA_ROT_IS_MODULO（回转轴模数换算）确定
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）。

旋转轴和主轴模数换算

MD30320 \$MA_DISPLAY_IS_MODULO (回转轴或主轴的模数显示)

在 MD30310 和 MD30320 置“1”的情况下，一旦实际值显示在 0° 与 359.999° 之间切换时，或者在沿正旋转方向超出 359.999° 时，或者在沿负旋转方向低于 0° 时，模数显示立即改变。

对于内部设定值及实际值而言，回转轴和主轴的特性不同：

- 就主轴而言，在每次主轴编程时（M3、M4 和 M5）均发生模数换算。
- 就回转轴而言，模数换算只会随新的运行指令发生。

10.3.2 激活模数功能的回转轴（无限旋转的回转轴）

绝对位置编程(AC,ACP,CAN,G90)

示例：通过绝对位置编程将模数回转轴作为定位轴：

POS[<轴名称>] = ACP(<值>)

- <值>：模数回转轴的目标位置，范围 0° 至 359.999°
通过以下机床数据进行了区域平移时，可采用负值：
 - MD30340 \$MA_MODULO_RANGE_START
 - MD30330 \$MA_MODULO_RANGE
- ACP：正旋转方向（逆时针方向）目标位置的绝对位置编程。
- ACN：负旋转方向（顺时针方向）目标位置的绝对位置编程。
- AC / G90：逐段/模态生效的目标位置绝对位置编程。
旋转方向取决于模数回转轴的实际位置。
 - 目标位置 > 实际位置：沿正旋转方向运行
 - 目标位置 < 实际位置：沿负旋转方向运行

说明

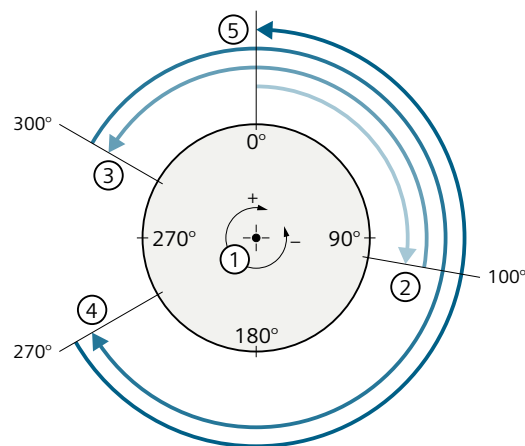
G90 定位特性可通过 MD30455 \$MA_MISC_FUNCTION_MASK，位 2 配置。参见章节“ 360° 模数 (页 317)”。

应用示例

ACP / ACN：在工件不对称的情形下定义旋转方向时，须确保旋转运动中不会发生碰撞。

编程示例

模数回转轴 C 的起始位置为 0° （见下图）。



- ① POS[C] = ACP(100); 沿正旋转方向运行到 100°位置
- ② POS[C] = ACN(300); 沿负旋转方向运行到 300°位置
- ③ POS[C] = ACP(240); 沿正旋转方向运行到 240°位置
- ④ POS[C] = AC(0); 沿负旋转方向运行到 0°位置

图 10-5 模数回转轴上的绝对位置编程

以最短行程定位的绝对位置编程(DC)

示例：通过绝对位置编程以最短行程将模数回转轴作为定位轴：

POS[<轴名称>] = ACP(<值>)

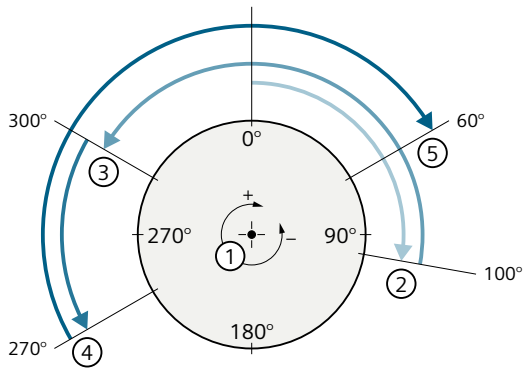
- <值>：模数回转轴的目标位置，范围 0°至 359.999°
通过以下机床数据进行了区域平移时，可采用负值：
 - MD30340 \$MA_MODULO_RANGE_START
 - MD30330 \$MA_MODULO_RANGE
- DC：目标位置的绝对位置编程，沿使行程最短的旋转方向逼近。

说明

若两个方向上的运行行程相同（180°），则沿正旋转方向运行。

编程示例

模数回转轴 C 的起始位置为 0°（见下图）



- ① POS[C] = DC(100); 以最短路径运行到 100°位置
 - ② POS[C] = DC(300); 以最短路径运行到 300°位置
 - ③ POS[C] = DC(240); 以最短路径运行到 240°位置
 - ④ POS[C] = DC(60); 以最短路径运行到 60°位置
- ; 沿正旋转方向

图 10-6 模数回转轴上采用最短运行的绝对位置编程

程序段搜索响应

在进行含计算的程序段搜索后，可通过系统变量 \$AC_RETPOINT[<轴>] 读取记录的经过模数转换的搜索位置。

含计算的程序段搜索结束后 ASUB 的边界条件

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。

增量位置编程(IC, G91)

定位轴举例： POS[轴名] = IC(+/-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

编程	结果
POS[C]=IC(720)	C 轴正转 720° (2 圈)
POS[C]=IC(-180)	C 轴反转 180°

10.3 回转轴的编程

程序示例：模数回转轴作为无限旋转的回转轴

```
程序代码
LOOP:
POS[C] = IC(720) ; 作为定位轴运行 270°
GOTOB LOOP      ; 跳回至 Label LOOP
```

10.3.3 没有激活模数功能的回转轴

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 $\geq \pm 360^\circ$ 。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！

10.3 回转轴的编程

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。但其运行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 $\geq +/-360^\circ$ 。
- 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。

10.3.4 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“分度轴 (页 333)”。

10.4 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	用于定位和编程的模数换算
MD30320 \$MA_DISPLAY_IS_MODULO	用于位置显示的模数换算
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，定位时不采用模数换算，即软件限位开关生效；位置显示为绝对位置
1	0	1	允许	轴是回转轴，定位时不采用模数换算，即软件限位开关生效；位置显示为模数位置； 应用：工作区域在 $\pm 1000^\circ$ 之间的轴。

10.5 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，定位时采用模数换算，即软件限位开关不生效，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）；可以采用/不采用针对轴的工作区域限制
1	1	0	允许	轴是回转轴，定位时采用模数换算，即软件限位开关不生效，工作区域不受限制；位置显示为绝对位置；可以采用/不采用针对轴的工作区域限制
0	0 或 1	0 或 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“手动运行 (页 743)”。

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）

10.5 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。

更多信息

功能手册之监控与补偿；回参考点运行 (R1)

功能手册之监控与补偿；回参考点运行 (R1) 回转轴镜像

指令 MIRROR (C) 或 AMIRROR (C) 可对回转轴进行镜像。

回参考点

更多信息

功能手册之监控与补偿；回参考点运行 (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：

更多信息

主轴 (页 395)

10.6 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。

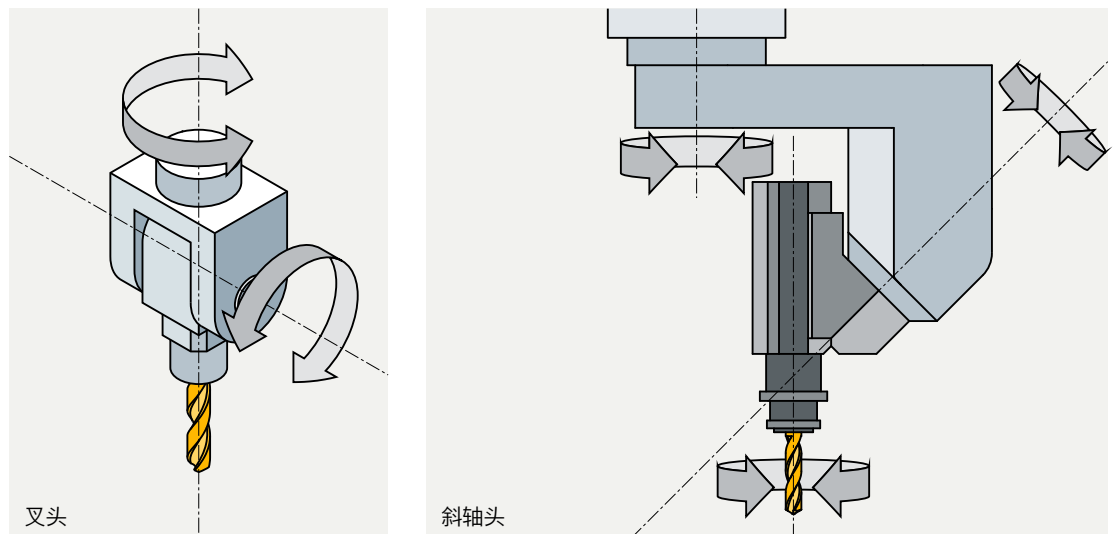


图 10-7 叉头、倾斜回转头

分度轴

11.1 简要说明

如果正常情况下机床轴只在一定数量的固定位置点之间运行，那么这些位置会被作为所谓的分度位置来编程。在 NC 程序中，这些被称为分度轴的机床轴可使用以分度位置为基准的专门指令来控制运行。

分度轴的典型应用场合例如有刀库（刀具转塔、链式刀库或盒式刀库）。此时的刀位作为分度位置来编程。

分度轴可在以下运行方式下运行：

- 自动方式
- MDA
- JOG
- JOG / INC1, INC10, ... INCvar

11.2 详细说明

11.2.1 在 AUTO 模式下运行分度轴

运行到任意位置

使用通用运行指令(G90, G91, AC, IC, ...)可将分度轴运行到机床轴运行范围内的任意位置。

运行至分度位置

使用分度轴的专用运行指令可确保只驶向所设置的分度位置。参见“编程 (页 345)”章节。

11.2.2 在 JOG 模式下运行分度轴

在 JOG 运行方式下，分度位置在分度轴的激活测量系统进行了回参考点/同步后才会被考虑。

更多信息

回参考点 (页 35)

连续运行(JOG)

点按模式

在点按模式(SD41050 \$SN_JOG_CONT_MODE_LEVELTRIGGRD = 1)下，按下运行键后分度轴会按照选择的方向运行。松开运行键后，分度轴会在最近的分度位置处停下。所停下的分度位置取决于：

- 至分度位置的距离
- 运行速度
- 轴动态响应

长按模式

在长按模式(SD41050 \$SN_JOG_CONT_MODE_LEVELTRIGGRD = 0)下，按下运行键（首个上升沿）后分度轴按通常方式运行。重新按下运行键（第二个上升沿）后，分度轴会在最近的分度位置处停下。所停下的分度位置取决于：

- 至分度位置的距离
- 运行速度
- 轴动态响应

分度轴一般在点按模式下运行。

更换方向

如果在到达分度位置前操作人员按下相反方向的运行键来更改方向，分度轴会继续运行到最近的可以停下的分度位置。接着需要重新按下运行键来启动反方向的运行。

更多信息

有关 JOG 运行方式下的轴运行的详细信息请见：

手动运行 (页 743)

断续运行 (INC)

与激活的工艺功能(INC1, INC10, ... , INCvar)无关，在按下运行键后分度轴会在所选择的方向上一直运行到下一个分度位置。

点按模式

在点按模式(SD41050 \$SN_JOG_CONT_MODE_LEVELTRIGGRD = 1)下，松开运行键后分度轴会立即停止运行。分度轴因此有时并不位于分度位置上。重新按下运行键后，分度轴会继续运行到分度位置。

长按模式

在长按模式(SD41050 \$SN_JOG_CONT_MODE_LEVELTRIGGRD=0)下，按下运行键（首个上升沿）后分度轴按通常方式运行。重新按下运行键（第二个上升沿）后，分度轴会立即停止运行。分度轴因此有时并不位于分度位置上。重新按下运行键后，分度轴会继续运行到分度位置。

旋转进给率

分度轴在 JOG 运行方式下的旋转进给率取决于：

SD41100 \$SN_JOG_REV_IS_ACTIVE, 位 0 = <值>

<值>	含义
0	取决于运行模式，旋转进给率由以下设定数据确定： - 分度轴作为定位轴运行： SD43300 \$SA_ASSIGN_FEED_PER_REV_SOURCE（定位轴的旋转进给率） - 分度轴作为几何轴运行，该几何轴上有一个带旋转的框架生效： SD42600 \$SC_JOG_FEED_PER_REV_SOURCE（JOG 下的旋转进给率）
1	分度轴以从主轴上导出的旋转进给率运行： 相关机床数据： - MD32050 \$MA_JOG_REV_VELO（JOG 旋转进给率） - MD32040 \$MA_JOG_REV_VELO_RAPID（JOG 旋转进给率叠加快速移动速度）

更多信息

- 进给率 (页 201)
- 手动运行 (页 743)
- 编程手册之 NC 编程

11.2.3 PLC 运行分度轴

分度轴也可以由 PLC 用户程序控制运行，作为：

- 并行定位轴，参见章节“定位轴 (页 277)”
- 异步子程序 (ASUB)

更多信息

功能手册之基本功能；运行方式组、通道、程序运行、复位特性

参见

回参考点 (页 35)

11.3 调试

11.3.1 分度轴的机床数据

分度轴

作为分度轴的轴定义通过进行分度位置表与该轴的分配完成，使用下面的轴专用机床数据：

MD30500 \$MA_INDEX_AX_ASSIGN_POS_TAB = <值>

<值>	含义
1	该轴分配给分度位置表 1。
2	该轴分配给分度位置表 2。
3	等距分度 (页 342)

等距分度中的第一个位置

对于等距分度，第一个分度位置的绝对位置通过以下轴专用机床数据设置：

MD30503 \$MA_INDEX_AX_OFFSET

分度位置表

在分度位置表中分度位置以毫米、英寸或度为单位进行保存。目前两个分度位置表可供使用。

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)

其中 <n> = 0 ... (最大数量 - 1)

分度位置表多次分配

一个分度位置表可分配给多个轴。

前提条件：这些轴都是同一类型的(线性轴、回转轴、360°模数轴)

一般边界条件

- 分度位置表中的分度位置必须以升序连续排列。
- 前后相连排列的分度位置不得相同。
- 分度轴应在基本坐标系中给定。

模数回转轴上的附加前提条件

- 允许范围: $0^\circ \leq \text{分度位置} < 360^\circ$
- 如果分度轴位于分度位置表的最后一个分度位置，那么在以正旋转方向向下一个分度位置继续运行时，分度轴会运行到第一个分度位置。
- 如果分度轴位于分度位置表的第一个分度位置，那么在以负旋转方向向下一个分度位置继续运行时，分度轴会运行到最后一个分度位置。

分度位置数

一个分度位置表中最多可以记录 60 个分度位置。实际使用或可编程的分度位置数量由以下机床数据设置:

MD10900 \$MN_INDEX_AX_LENGTH_POS_TAB_1 (分度位置表 1 的分度位置数量)

MD10920 \$MN_INDEX_AX_LENGTH_POS_TAB_2 (分度位置表 2 的分度位置数量)

说明

分度位置表中超出所设定数量的位置记录无效。

分度位置的单位制

分度位置的单位制由 NC 专门设定，使用：

MD10270 \$MN_POS_TAB_SCALING_SYSTEM = <值>

<值>	单位制
0	公制
1	英制

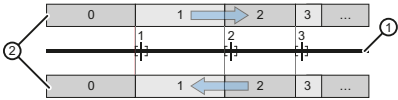
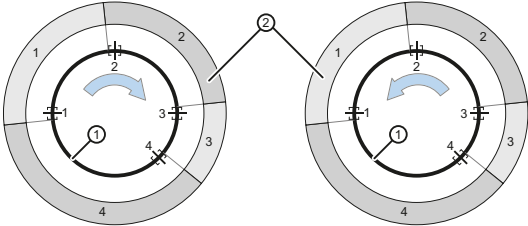
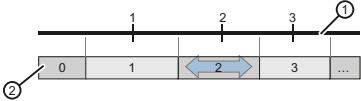
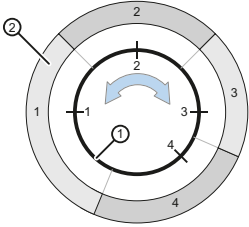
说明

此机床数据在适用以下条件时才被检测：MD10260 \$MN_CONVERT_SCALING_SYSTEM == 1

显示当前分度位置

使用以下机床数据可以确定，在分度轴运行时当运行到哪个实际位置时系统变量
\$AA_ACT_INDEX_AX_POS_NO 中当前分度位置的显示会发生变化：

MD10940 \$MN_INDEX_AX_MODE, 位 0 = <值>

位	<值>	含义
0	0	如果沿升序分度位置方向运行，当到达/超过下一个分度位置的“精准停”窗口 (MD36010 \$MA_STOP_LIMIT_FINE) 时，系统变量 \$AA_ACT_INDEX_AX_POS_NO 中显示的当前分度位置会发生变化。 相反，如过沿降序分度位置方向运行，则在退出“精准停”窗口时，分度位置显示就会发生变化。
		
		线性轴 
		模数回转轴
1		① 带分度位置和“精准停”窗口的分度轴运行范围
		② 系统变量 \$AA_ACT_INDEX_AX_POS_NO 中的分度位置显示
		当到达/超过运行方向上到下一个分度位置的行程段的一半时，系统变量 \$AA_ACT_INDEX_AX_POS_NO 中显示的当前分度位置会发生变化。
		
		线性轴 
		模数回转轴
		① 带分度位置的分度轴运行范围
		② 系统变量 \$AA_ACT_INDEX_AX_POS_NO 中的分度位置显示

说明

系统变量 \$AA_ACT_INDEX_AX_POS_NO 不能用于确定是否已到达分度位置。除此之外还必须评估以下两个 NC/PLC 接口信号之一：

<Axis>.basic.in.indexAxisInPos（已到达分度位置）

<Axis>.basic.in.posFineReached（已到达精准停）

示例

假设：

- MD10940 \$MN_INDEX_AX_MODE，位 0 == 1
- 轴位于公差窗口“精准停”(<Axis>.basic.in.posFineReached == 0) 范围外的分度位置前

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 (<Axis>.basic.in.posFineReached == 1) 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。

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.posFineReached	LBP_Axis*.E_ExactFine	DB31, ... DBX60.7
<Axis>.basic.in.indexAxisInPos	LBP_Axis*.E_IndexAxisPos	DB31, ... DBX76.6

11.3.2 等距分度的机床数据

11.3.2.1 属性

等分间隔

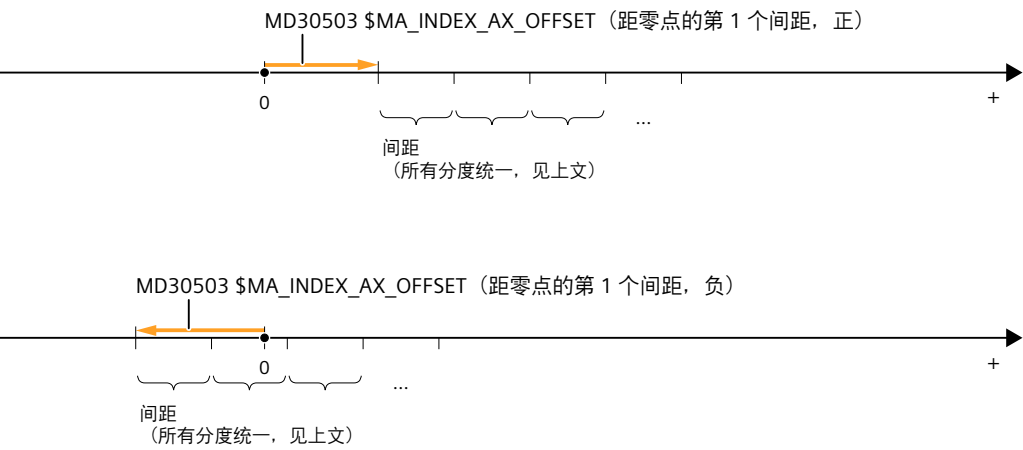
线性轴上的等分间隔是根据以下公式确定的：

间距 =

分子 (MD30501 \$MA_INDEX_AX_NUMERATOR)

分母 (MD30502 \$MA_INDEX_AX_DENOMINATOR)

线性轴



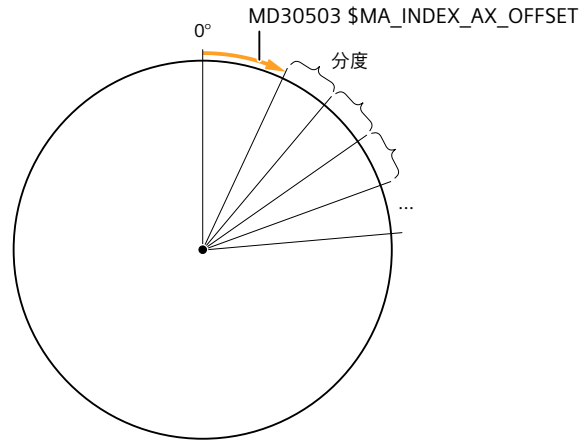
模数回转轴

分度 =

分子 (MD30330 \$MA_MODULO_RANGE)

分母 (MD30502 \$MA_INDEX_AX_DENOMINATOR)

$$\text{分度} = \frac{\text{模数区域的大小 (MD30330 \$MA\_MODULO\_RANGE)}}{\text{模数区域中的分度数量 (MD30502 \$MA\_INDEX\_AX\_DENOMINATOR)}}$$



11.3.2.2 端面齿轴

功能

对于“端面齿轴”，当到达一个分度位置时回转轴会通过一个特殊的齿轮（端面齿）被锁紧。此时，止动器或齿轮通过一根线性轴实现啮合。为了不损坏机床的机械机构，锁紧始终只能在分度位置上进行。对于端面齿轴，所有分度位置之间的距离必须相同（等距分度位置）。

前提条件

- “端面齿轴”应为回转轴。
- “端面齿轴”应为分度轴。参见章节“调试 (页 336)”
- 分度轴的激活测量系统必须已同步或已回参考点：


```
<Axis>.basic.in.enc1Synchronized /
<Axis>.basic.in.enc2Synchronized == 1 (已回参考点/已同步, 编码器 1/2)
```

更多信息

回参考点 (页 35)

调试

机床数据

- MD30300 \$MA_IS_ROT_AX = 1 (回转轴)
- MD30505 \$MA_HIRTH_IS_ACTIVE[<轴>] = 1 (带端面齿的分度轴)
- MD30500 \$MA_INDEX_AX_ASSIGN_POS_TAB[<轴>] = 3 (等距分度的分度轴)

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.enc1Synchronized	LBP_Axis*.E_RefSyn1	DB31, ...DBX60.4
<Axis>.basic.in.enc2Synchronized	LBP_Axis*.E_RefSyn2	DB31, ...DBX60.5

11.3.3 轴专用系统变量

\$AA_PROG_INDEX_AX_POS_NO

功能

系统变量含有为分度轴编程设置的分度位置的编号。

句法

\$AA_PROG_INDEX_AX_POS_NO[<轴>]

含义

\$AA_PROG_INDEX_AX_POS_NO	
0	此轴不是分度轴或分度轴当前未运行到一个分度位置
> 0	编程设置的分度位置的编号

\$AA_ACT_INDEX_AX_POS_NO

功能

系统变量含有分度轴最后到达或超过的分度位置的编号。

句法

\$AA_ACT_INDEX_AX_POS_NO[<轴>]

含义

\$AA_ACT_INDEX_AX_POS_NO	
0	此轴不是分度轴。
> 0	最后到达或越过的分度位置的编号

11.3.4 NC/PLC 接口信号

轴专用接口信号

当分度轴在一个分度位置 ± 精准停窗口(MD36010 \$MA_STOP_LIMIT_FINE)处停止时，轴专用接口信号会被置位：

<Axis>.basic.in.indexAxisInPos = 1（分度轴就位）

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.indexAxisInPos	LBP_Axis*.E_IndexAxisPos	DB31, ...DBX76.6

11.4 编程

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

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)

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。
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。
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。
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。
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。
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。

- 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。

系统变量

分度位置的信息可通过以下系统变量读取：

- `$AA_PROG_INDEX_AX_POS_NO`
最后编程的分度位置的编号
- `$AA_ACT_INDEX_AX_POS_NO`
最后越过的分度位置的编号
该显示取决于机床数据 MD10940 `$MN_INDEX_AX_MODE` 中的设置。
- `$AA_ACT_INDEX_AX_POS_NO`
最后到达或越过的分度位置的编号

分度位置			
轴类型	数值	说明	
模数回转轴	1, ... n	n = 分度位置数量 (页 336)	
线性轴	0, 1, 2, 3, ... 59, 60, 61	0	当前实际位置位于最小的分度位置以下
		61	当前实际位置位于最大的分度位置以上

等距分度位置			
轴类型	数值	说明	
模数回转轴	1 ... m	m = 分母(分子)	
线性轴	... -3, -2, -1, 0, 1, 2, 3, ...		

11.5 边界条件

分度轴

不同的通道专用和轴专用 NC/PLC 接口信号

如果在分度轴运行时至少出现了以下其中一个信号，则会立即停止运行。分度轴此时不予考虑。

- 通道专用信号
 - `<Chan>.basic.out.pathFeedrateOvrFactor == 0` (进给倍率)
 - `<Chan>.basic.out.deleteDistanceToGo == 1` (删除剩余行程)
 - `<Chan>.basic.out.ncStop == 1` (NC 停止)
 - `<Chan>.basic.out.ncStopAxesAndSpindles == 1` (进给轴和主轴 NC 停止)
 - `<Chan>.basic.out.reset == 1` (复位)
- 轴专用信号
 - `<Axis>.basic.out.feedStopSpindleStop == 1` (进给停止)

同步动作

- 停止 (MOV=0)
如果分度轴通过设置 MOV=0 的同步动作而停止，那么在运行方向上的下一个分度位置处也会停止。
- 删除剩余行程 (DELDTG)
如果为一个分度轴通过 DELDTG 执行的同步动作删除了剩余行程，分度轴会立即停止，不继续向分度位置运行。

更多信息：功能手册之同步动作

框架和零点偏移

分度轴通常不需要使用框架。因此建议在 NC 程序中取消分度轴的框架和零点偏移。

软件限位开关

在 JOG 或 JOG/INC 运行方式下手动运行分度轴时，分度轴最多可运行到软件限位开关之前的最后一个分度位置处。

通过手轮进行手动运行 (DRF)

通过手轮运行分度轴时，不考虑分度位置。也就是说分度轴在手轮脉冲结束时立即停止运行，即使正位于两个分度位置之间。

“端面齿轴”

对于“端面齿轴”，除了上述的所谓边界条件，还需注意以下的边界条件。

不同的通道专用和轴专用 NC/PLC 接口信号

如果在“端面齿轴”运行时至少出现了以下其中一个信号，则会在运行方向上的下一个分度位置处停止运行：

- 通道专用信号
 - `<Chan>.basic.out.pathFeedrateOvrFactor == 0`（进给倍率）
 - `<Chan>.basic.out.deleteDistanceToGo == 1`（删除剩余行程）
 - `<Chan>.basic.out.ncStop == 1`（NC 停止）
 - `<Chan>.basic.out.ncStopAxesAndSpindles == 1`（进给轴和主轴 NC 停止）
 - `<Chan>.basic.out.reset == 1`（复位）
- 轴专用信号
 - `<Axis>.basic.out.feedStopSpindleStop == 1`（进给停止）

急停

如果在“端面齿轴”运行时触发了急停，则轴会立即停止：

`<Nc>.basic.out.emergencyStop== 1`（急停）

更多信息：功能手册之基本功能，急停

注意

逼近分度位置

急停后重新激活机床上的端面齿锁紧之前，应先在 JOG 运行方式下由 PLC 用户程序或由操作人员手动将“端面齿轴”运行到分度位置。

同步动作

- 删除剩余行程（DELDTG）

如果为一个“端面齿轴”通过 DELDTG 执行的同步动作删除了剩余行程，会在运行方向上的下一个分度位置处停止运行。

更多信息：功能手册之同步动作，“详细说明”>“同步动作中的动作”>“删除剩余行程 (DELDTG)”

11.5 边界条件

不同的轴专用功能

“端面齿轴”不可与下列功能一起使用：

- 运动转换
更多信息：功能手册之转换分册，运动转换
- 借助实际值设置的零点偏移：PRESETON
更多信息：编程手册之 NC 编程；章节“坐标转换（框架）”
- 旋转进给率（G95, FPRAON）
- 通过手轮手动运行，以及借助手轮的行程或速度叠加（DFR）
更多信息：手动运行 (页 743)
- 框架
更多信息：功能手册之基本功能，轴、坐标系、框架 > 框架
- 插补补偿，如外部零点偏移，DRF 偏移
- 同步
“端面齿轴”不允许为以下耦合类型的从动轴：
 - 主值同步中的从动轴
更多信息：引导值耦合 (页 608)
 - 联动轴
更多信息：联动 (页 571)
 - 龙门轴组中的从动轴
更多信息：龙门轴 (页 357)

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Chan>.basic.out.pathFeedrateOvrFactor	LBP_Chan*.A_FD_OR	DB21, ...DBB4
<Chan>.basic.out.deleteDistanceToGo	LBP_Chan*.A_DeleteDTG	DB21, ...DBX6.2
<Chan>.basic.out.ncStop	LBP_Chan*.A_NCStop	DB21, ...DBX7.3
<Chan>.basic.out.ncStopAxesAndSpindles	LBP_Chan*.A_NCStopASp	DB21, ...DBX7.4
<Chan>.basic.out.reset	LBP_Chan*.A_Reset	DB21, ...DBX7.7
<Axis>.basic.out.feedStopSpindleStop	LBP_Axis*.A_FDSpStop	DB31, ...DBX4.3
<Nc>.basic.out.emergencyStop	LBP_NC.A_EMERGENCY	DB10.DBX56.1

11.6 示例

11.6.1 示例 1：回转轴作为分度轴

机床轴 AX5 为带有 8 个转塔刀位的刀具转塔（连续旋转的回转轴）。转塔刀位之间的距离是恒定的（等距分度位置）。

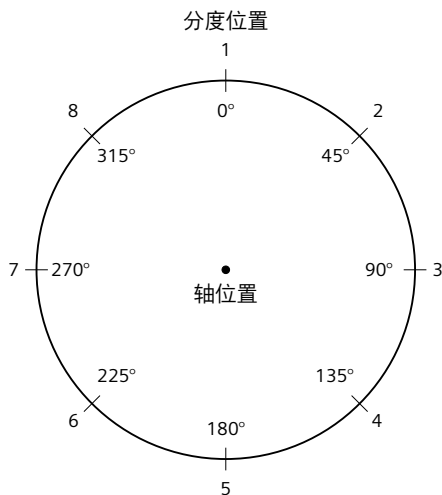


图 11-1 刀塔，包含 8 个刀位

分度位置表 1

MD10910 \$MN_INDEX_AX_POS_TAB_1[0] = 0	第 1 个分度位置 = 0°
MD10910 \$MN_INDEX_AX_POS_TAB_1[1] = 45	第 2 个分度位置 = 45°
MD10910 \$MN_INDEX_AX_POS_TAB_1[2] = 90	第 3 个分度位置 = 90°
MD10910 \$MN_INDEX_AX_POS_TAB_1[3] = 135	第 4 个分度位置 = 135°
MD10910 \$MN_INDEX_AX_POS_TAB_1[4] = 180	第 5 个分度位置 = 180°
MD10910 \$MN_INDEX_AX_POS_TAB_1[5] = 225	第 6 个分度位置 = 225°
MD10910 \$MN_INDEX_AX_POS_TAB_1[6] = 270	第 7 个分度位置 = 270°
MD10910 \$MN_INDEX_AX_POS_TAB_1[7] = 315	第 8 个分度位置 = 315°

分度位置表 1 的长度

MD10900 \$MN_INDEX_AX_LENGTH_POS_TAB_1 = 8	表 1 中的 8 个分度位置

机床轴 AX5 定义为模数回转轴和分度轴

MD30300 \$MA_IS_ROT_AX [AX5] = 1	回转轴
MD30310 \$MA_ROT_IS_MODULO [AX5] = 1	模数回转轴
MD30500 \$MA_INDEX_AX_ASSIGN_POS_TAB [AX5] = 1	分度轴，分度位置表 1

11.6.2 示例 2：分度轴为线性轴

机床轴 AX6 为带有 10 个工件位的工件托盘。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。刀具轴为 AX6。

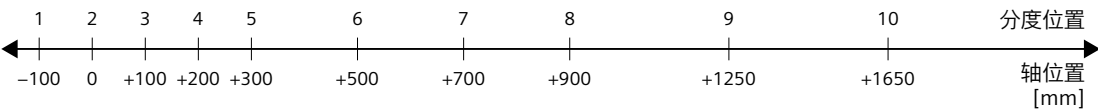


图 11-2 工件托盘的分度位置

分度位置表 2

MD10930 \$MN_INDEX_AX_POS_TAB_2[0] = -100	第 1 个分度位置 = -100 mm
MD10930 \$MN_INDEX_AX_POS_TAB_2[1] = 0	第 2 个分度位置 = 0 mm
MD10930 \$MN_INDEX_AX_POS_TAB_2[2] = 100	第 3 个分度位置 = 100 mm
MD10930 \$MN_INDEX_AX_POS_TAB_2[3] = 200	第 4 个分度位置 = 200 mm
MD10930 \$MN_INDEX_AX_POS_TAB_2[4] = 300	第 5 个分度位置 = 300 mm
MD10930 \$MN_INDEX_AX_POS_TAB_2[5] = 500	第 6 个分度位置 = 500 mm
MD10930 \$MN_INDEX_AX_POS_TAB_2[6] = 700	第 7 个分度位置 = 700 mm
MD10930 \$MN_INDEX_AX_POS_TAB_2[7] = 900	第 8 个分度位置 = 900 mm
MD10930 \$MN_INDEX_AX_POS_TAB_2[8] = 1250	第 9 个分度位置 = 1250 mm
MD10930 \$MN_INDEX_AX_POS_TAB_2[9] = 1650	第 10 个分度位置 = 1650 mm

分度位置表 1 的长度

MD10920 \$MN_INDEX_AX_LENGTH_POS_TAB_2 = 10	表 2 中的 10 个分度位置

机床轴 AX5 定义为分度轴

MD30500 \$MA_INDEX_AX_ASSIGN_POS_TAB [AX6] = 2	分度轴，分度位置表 2

11.6.3 示例 3：模数回转轴为等距分度轴

机床轴 AX4 为等距分度的模数回转轴。分度从 5°开始。分度位置差为 20°。

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、245 [度]

分度数据

MD30501 \$MA_INDEX_AX_NUMERATOR[AX4] ⇒ MD30330 \$MA_MODULO_RANGE[AX4] = 360.0	对于模数回转轴，该值会被忽略并使用 MD30330 \$MA_MODULO_RANGE（缺省值：360°）。
MD30502 \$MA_INDEX_AX_DENOMINATOR[AX4] = 18	分度位置数量 = $360^\circ / 20^\circ = 18$
MD30503 \$MA_INDEX_AX_OFFSET[AX4] = 5.0	首个分度位置 = 5.0°
MD30505 \$MA_HIRTH_IS_ACTIVE[AX4] = 0	无端面齿

机床轴 AX4 定义为模数回转轴和等距分度轴

MD30300 \$MA_IS_ROT_AX[AX4] = 1	回转轴
MD30310 \$MA_ROT_IS_MODULO[AX4] = 1	模数功能生效
MD30500 \$MA_INDEX_AX_ASSIGN_POS_TAB[AX4] = 3	等距分度轴

11.6.4 示例 4：回转轴作为等距分度轴，运行区域受限

机床轴 AX4 为等距分度的回转轴。分度位置每 20°等距分布，从 100°开始。100°以内和 260°以外的位置应无法逼近。

由此得到以下分度位置：

100、120、140、160、180、200、220、240、260 [度]

分度数据

MD30501 \$MA_INDEX_AX_NUMERATOR[AX4] = 360.0	分子 = 360°
MD30502 \$MA_INDEX_AX_DENOMINATOR[AX4] = 18	分度位置数量 = 360° / 20° = 18
MD30503 \$MA_INDEX_AX_OFFSET[AX4] = 100.0	首个分度位置 = 100,0°
MD30505 \$MA_HIRTH_IS_ACTIVE[AX4] = 0	无端面齿

机床轴 AX4 定义为模数回转轴和等距分度轴

MD30300 \$MA_IS_ROT_AX[AX4] = 1	回转轴
MD30310 \$MA_ROT_IS_MODULO[AX4] = 1	模数功能生效
MD30500 \$MA_INDEX_AX_ASSIGN_POS_TAB[AX4] = 3	等距分度轴
MD36100 \$MA_POS_LIMIT_MINUS[AX4] = 100.0	第 1 负向软件限位开关 = 100.0°
MD36110 \$MA_POS_LIMIT_PLUS[AX4] = 260.0	第 1 正向软件限位开关 = 260.0°

11.6.5 示例 5：线性轴作为等距分度轴

机床轴 AX4 为等距分度的线性轴。分度位置每 10 mm 等距分布，从 -200 mm 开始。

由此得到以下分度位置：

-200、-190、-180、... 180、190、200 [mm]

分度数据

MD30501 \$MA_INDEX_AX_NUMERATOR[AX4] = 10.0	分子 = 10
MD30502 \$MA_INDEX_AX_DENOMINATOR[AX4] = 1	分母 = 1
MD30503 \$MA_INDEX_AX_OFFSET[AX4] = -200.0	首个分度位置 = -200,0°
MD30505 \$MA_HIRTH_IS_ACTIVE[AX4] = 0	无端面齿

机床轴 AX4 定义为线性轴和等距分度轴

MD30300 \$MA_IS_ROT_AX[AX4] = 0	线性轴
MD30310 \$MA_ROT_IS_MODULO[AX4] = 0	模数功能未生效

MD30500 \$MA_INDEX_AX_ASSIGN_POS_TAB[AX4] = 3	等距分度轴
MD36100 \$MA_POS_LIMIT_MINUS[AX4] = -200.0	第 1 负向软件限位开关 = -200.0 mm
MD36110 \$MA_POS_LIMIT_PLUS[AX4] = 200.0	第 1 正向软件限位开关 = 200.0 mm

11.6.6 示例 6：“端面齿轴”

机床轴 AX4 为“端面齿轴”（等距分度并带端面齿的模数回转轴）。分度位置每 1°等距分布，从 0°开始。

分度数据

MD30501 \$MA_INDEX_AX_NUMERATOR[AX4] ⇒ MD30330 \$MA_MODULO_RANGE[AX4] = 360.0	对于模数回转轴，该值会被忽略并使用 MD30330 \$MA_MODULO_RANGE（缺省值：360°）。
MD30502 \$MA_INDEX_AX_DENOMINATOR[AX4] = 360	分度位置数量 = $360^\circ / 1^\circ = 360$
MD30503 \$MA_INDEX_AX_OFFSET[AX4] = 0.0	首个分度位置 = 0,0°
MD30505 \$MA_HIRTH_IS_ACTIVE[AX4] = 1	端面齿

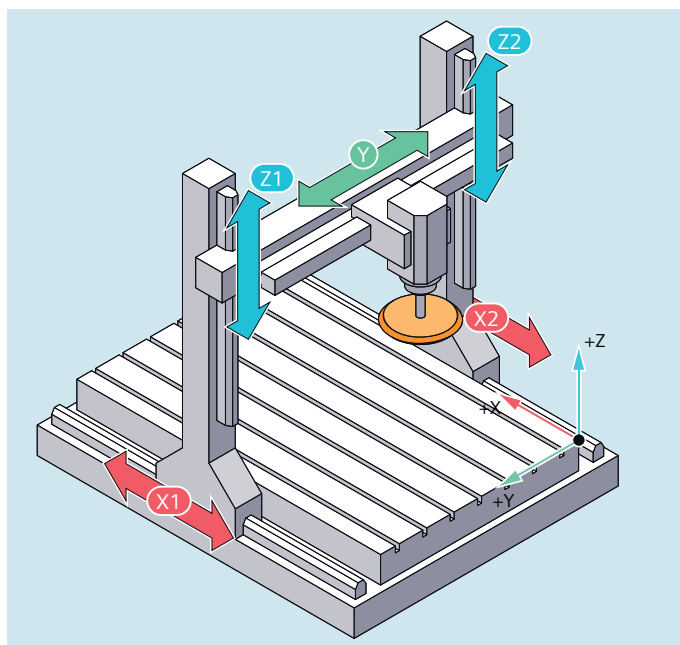
机床轴 AX4 定义为线性轴和等距分度轴

MD30300 \$MA_IS_ROT_AX[AX4] = 1	回转轴
MD30310 \$MA_ROT_IS_MODULO[AX4] = 1	模数功能生效
MD30500 \$MA_INDEX_AX_ASSIGN_POS_TAB[AX4] = 3	等距分度轴
MD36100 \$MA_POS_LIMIT_MINUS[AX4] = -200.0	第 1 负向软件限位开关 = -200.0 mm
MD36110 \$MA_POS_LIMIT_PLUS[AX4] = 200.0	第 1 正向软件限位开关 = 200.0 mm

龙门轴

12.1 简要说明

对于龙门机床，不同的机床元件，例如龙门和横梁，分别由多个并行工作的轴移动。一起移动机床部件的轴被称为龙门轴，或龙门轴组。由于机械结构，龙门轴彼此刚性连接，因此必须始终由控制系统同步运行。



X1/X2 龙门轴组：龙门

Z1/Z2 龙门轴组：横梁

图 12-1 示例：带有龙门和横梁的龙门铣床

主动轴

龙门轴组的主动轴是代表龙门轴组的轴。龙门轴组的运动只能通过该轴进行编程。

同步轴

龙门轴组的同步轴是指由于与主动轴连接而由控制系统自动共同运行的轴。一个主动轴可以分配有任意数量的同步轴。

同步运行偏差

同步运行偏差是指同步轴的轴向实际值与其理想位置相对于主动轴实际值的偏差。同步运行偏差由控制系统持续监控。一旦超出警告极限，则显示一条消息。一旦超出报警极限，则整个龙门轴组停止。极限值可通过机床数据设置。

12.2 “龙门轴” 功能

12.2.1 定义一个龙门轴组

定义

龙门轴组的轴通过以下轴向机床数据定义：

MD37100 \$MA_GANTRY_AXIS_TYPE[AX1] = xy	
x	小数点后 10 位：龙门轴类型（主动轴或同步轴）
y	小数点后 1 位：龙门轴组的 ID

最多可以定义 8 个龙门轴组（龙门轴组 ID：1 - 8）。根据分配的轴，龙门轴组 ID 在所有通道或 NCU 中必须是唯一的。

原则上，一个龙门轴组可以分配有任意数量的同步轴。

示例

定义一个 ID=1、主动轴 AX1 和同步轴 AX2 的龙门轴组

- MD37100 \$MA_GANTRY_AXIS_TYPE[AX1] = 01（主动轴）
- MD37100 \$MA_GANTRY_AXIS_TYPE[AX2] = 11（同步轴）

边界条件

龙门轴组需要满足下列边界条件：

- 龙门轴组不得包含主轴。
- 同步轴不能为并行 POS 轴。
- 同步轴不得属于转换。
- 同步轴不能为另一轴耦合的跟随轴。

- 同步轴不能为另一轴耦合的主动轴。
- 龙门轴组的所有轴必须为相同的轴类型（线性轴或回转轴）：MD30300 \$MA_IS_ROT_AX（回转轴/主轴）

说明

驱动优化

在一台 SINAMICS S120 驱动设备上，最多可以同时优化或测量 3 台驱动（转速控制器优化/函数发生器）。因此，对于具有 3 个以上同时耦合的驱动的耦合，强烈建议将其分配给多台驱动设备。

12.2.2 同步运行偏差监控

监控的极限值

关于同步运行偏差，可以指定 2 个极限值。

龙门警告极限

龙门警告极限通过以下机床数据设置：

MD37110 \$MA_GANTRY_POS_TOL_WARNING（龙门警告极限）

如果同步运行偏差超过龙门警告极限，则会显示消息“超出警告极限”。此外 NC/PLC 接口信号置位：

`<Axis>.coupling.in.gantryWarningLimitExceeded = 1`（超出龙门警告极限）

低于警告极限后，该消息和接口信号自动重新复位。

说明

龙门警告极限

如果未显示“超出警告极限”消息，则必须在 MD37110 中输入值 0。

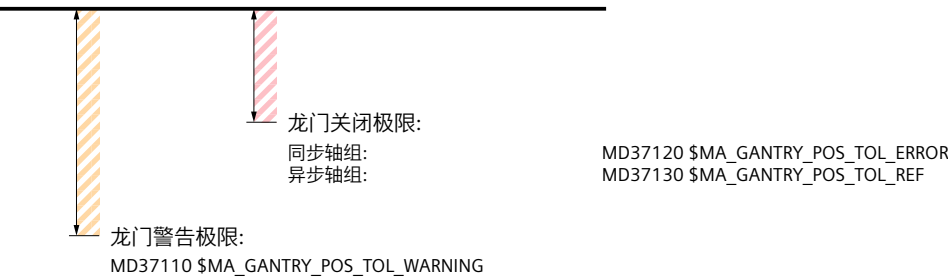
龙门关闭极限

龙门关闭极限通过以下机床数据设置：

- 对于同步龙门轴组：
MD37120 \$MA_GANTRY_POS_TOL_ERROR
- 对于异步龙门轴组：
MD37130 \$MA_GANTRY_POS_TOL_REF

如果同步运行偏差超过龙门关闭极限，则会显示报警 10653“超出误差限制”。此外 NC/PLC 接口信号置位：

```
<Axis>.coupling.in.gantryCutOffLimitExceeded = 1（超出龙门关闭极限）
```



在龙门轴组夹紧时，也会显示该报警（无伺服使能，龙门轴组处于“停止”状态）。

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.coupling.in.gantryCutOffLimitExceeded	LBP_Axis*.E_GantCOLimit	DB31, ...DBX101.2
<Axis>.coupling.in.gantryWarningLimitExceeded	LBP_Axis*.E_GantWarn	DB31, ...DBX101.3

12.2.3 同步运行偏差扩展监控

激活扩展监控

同步运行偏差扩展监控可以通过以下机床数据激活：

```
MD37150 $MA_GANTRY_FUNCTION_MASK, 位 0 = 1
```


扩展监控时还考虑到在跟随期间或龙门轴组松开时出现的主动轴和同步轴之间的同步运行偏差。

在第一次回参考点（增量编码器）或同步（绝对值编码器）后启动 NC 后，扩展监控生效。

超出龙门关闭极限

如果扩展监控激活时，超过同步运行偏差的关闭极限，则会显示报警 10653“超出误差限制”。

为了能够复位该报警，必须如下操作：

- 禁用扩展监控：
MD37150 \$MA_GANTRY_FUNCTION_MASK, 位 0 = 0
- 删除机床数据中显示的同步运行偏差：
MD37135 \$GANTRY_ACT_POS_TOL_ERROR = 0
- 删除报警
- 将龙门轴组的轴重新回参考点或同步
- 重新激活扩展监控：
MD37150 \$MA_GANTRY_FUNCTION_MASK, 位 0 = 1

12.2.4 龙门轴的回参考点和同步

应用场合

如果主动轴或同步轴使用增量测量系统，必须在 NC 启动后执行测量系统回参考点，同时保持轴耦合。

当龙门轴组的所有轴都运行至参考点后，必须对龙门轴可能出现的倾斜进行平衡和调整（龙门同步运行）。一旦完成，NC/PLC 接口信号置位：

<Axis>.coupling.in.gantrySyncOk = 1（龙门轴组已同步）

有关龙门轴回参考点或同步的步骤（参见“龙门轴的回参考点和同步 (页 363)”章节）。

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.coupling.in.gantrySyncOk	LBP_Axis*.E_GantGroupSyn	DB31, ...DBX101.5

12.2.5 控制动态响应

应用场合

从用户角度来看，龙门轴组仅通过主动轴运行。同步轴的设定值由 NC 直接从主动轴的设定值同步生成并输出给它。为了将同步运行偏差最小化，必须将龙门轴组中所有轴的控制动态响应设置为相同（参见“调试龙门轴 (页 374)”章节）。

说明

必须为龙门轴组的所有轴设置相同的控制动态响应。

故障特性

如果发生导致龙门轴组中的一个轴停止的故障，则始终停止整个龙门轴组。

12.2.6 拆解龙门轴组

说明

龙门轴组内的轴耦合可以通过以下机床数据取消：

MD37140 \$MA_GANTRY_BREAK_UP = 1（松开龙门轴组）

 小心

无同步运行
若龙门轴在此种运行状态下仍通过机械方式联结在一起，则运行主动轴或同步轴时会损坏机床！

设置生效后，龙门轴组的轴可以在 JOG、AUTOMATIC 和 MDA 运行方式下单独运行。

同步运行偏差监控或警告和关闭极限未生效。

NC/PLC 接口信号“龙门轴组已同步”将复位：

<Axis>.coupling.in.gantrySyncOk = 0

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.coupling.in.gantrySyncOk	LBP_Axis*.E_GantGroupSyn	DB31, ...DBX101.5

12.3 龙门轴的回参考点和同步

12.3.1 简介

启动时的倾斜位置

机床启动时，主动轴和同步轴可能会偏离其理想位置（比如龙门的位置倾斜）。通常情况下，此种偏离相对较小，因此各龙门轴仍可以回参考点。

特殊情况下（比如龙门轴由于故障、断电或急停停止时）必须在运行前检查偏移量是否在容许的范围之内，必要时需进行补偿性移动。

此种情况下，必须使用以下机床数据拆分龙门轴组：

MD37140 \$MA_GANTRY_BREAK_UP（松开龙门轴组）

龙门同步运行

控制系统打开后，所有的龙门轴必须首先回参考点，然后同步。在龙门同步运行中，所有龙门轴在**没有耦合的情况下运行至龙门轴组的参考点**。在龙门轴回参考点时，龙门轴组的参考点就是**主动轴的参考点**：

MD34100 REFP_SET_POS（使用距离编码系统时的参考点值/目标点）

否则，参考点即是**主动轴的当前实际位置**。

龙门轴的回参考点和同步程序根据特殊的过程表自动进行。

回参考点过程

装有增量测量系统的龙门轴同步过程按下列步骤完成：

部分 1：

主动轴回参考点

12.3 龙门轴的回参考点和同步

在激活 REF 机床功能的情况下，由 PLC 用户程序发出的主动轴接口信号启动龙门轴特有的轴方式回参考点：

```
<Axis>.basic.out.traversKeyPlus /  
<Axis>.basic.out.traversKeyMinus (运行键 +/-)。
```

主动轴运行至参考点（此过程与回参考点运行相对应）。

更多信息：

回参考点 (页 35)

所属同步轴同步运行。主动轴显示接口信号“已回参考点/同步”时，表明已到达参考点。

部分 2：

同步轴回参考点

一旦主动轴运行至其参考点，则同步轴**自动**回参考点（与回参考点运行相对应）。

更多信息：

回参考点 (页 35)

在此期间，由控制系统内部调节主动轴和同步轴之间的关联性，保证同步轴与主动轴的同步。同步轴显示接口信号“已回参考点/同步”时，表明已到达参考点。此时，之前的龙门轴间的相关性仍然有效。如果在龙门轴组中定义了另一个同步轴，则以同样的方式将其回参考点。

部分 3：

龙门同步运行

当龙门轴组的所有轴均已回参考点时，必须将其同步至已确定的参考位置。首先会对各同步轴的实际位置与已确定的主动轴的参考位置进行比较。

根据所得的主动轴与同步轴之间的实际值差值的不同，将其分为如下几种情况：

- 差值小于龙门警告极限：

MD37110 \$MA_GANTRY_POS_TOL_WARNING（龙门警告极限）

龙门同步运行将**自动**启动。此时，会显示消息“龙门轴组 X 同步中”。

消息“龙门轴组 X 同步中”可通过以下设置抑制：

MD37150 \$MA_GANTRY_FUNCTION_MASK 位 2 = 1

在**未耦合**的情况下，所有龙门轴均以机床数据中设置的速度运行至特定的位置数值。

MD34040 \$MA_REFP_VELO_SEARCH_MARKER（关闭速度）

在主动轴上确定位置数值：

MD34100 REFP_SET_POS（使用距离编码系统时的参考点/目标点）

当主动轴上有绝对编码器和间距编程编码器时，将选择性地运行至主动轴的当前实际位置或参考点数值，可通过以下机床数据进行设置：

MD34330 \$MA_REFP_STOP_AT_ABS_MARKER（无目标点的距离编码长度测量系统）

此时所使用的速度与回参考点驶入速度相对应。

MD34070 \$MA_REFP_VELO_POS（回参考点速度）

一旦所有的龙门轴达到其目标位置（理想位置），接口信号“龙门轴组已同步”将设置为 1 信号且龙门轴耦合再次激活。此时，龙门轴组所有轴的位置实际值必须相同。至此，龙门同步运行结束。

- 至少一个同步轴的偏差大于龙门警告极限时：

接口信号“龙门同步运行启动准备”将设置为 1 信号并给出消息“等待龙门轴组 X 启动同步”。此种情况下，龙门同步运行不会自动启动，而需由操作者或 PLC 用户程序启动。

主动轴通过接口信号“启动龙门同步运行”启动同步过程。此信号由主动轴发出。过程与上述过程类似。

后面的过程图中有回参考点过程和同步过程的图形显示。

12.3 龙门轴的回参考点和同步

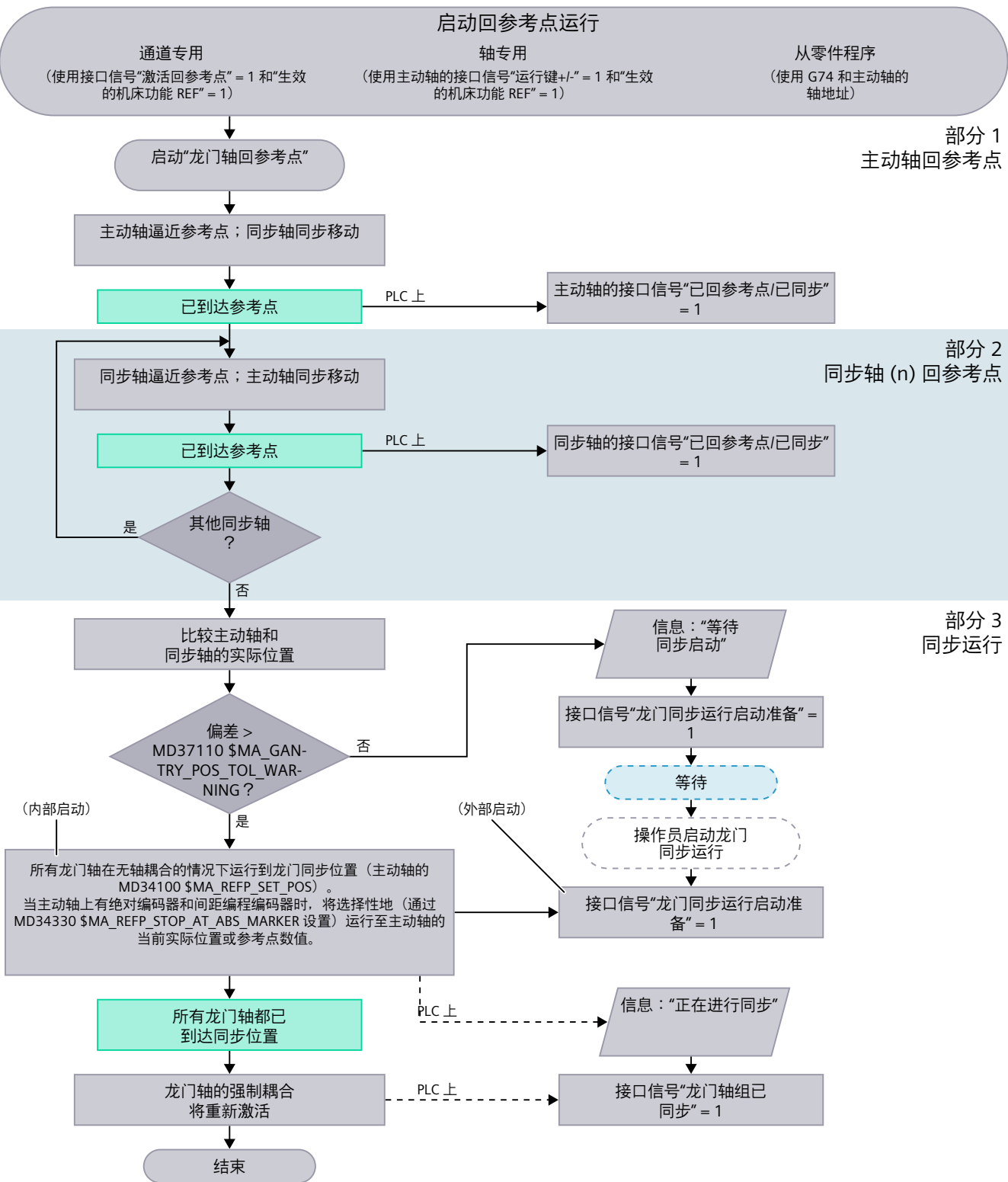


图 12-2 龙门轴回参考点过程和同步过程的过程图

同步运行

以下情形下必须进行同步：

- 龙门轴组所属各轴已运行至参考点，
- 同步丢失（如下）。

过程中断

若上述回参考点过程由于故障或复位中断，则应如下操作：

- **在步骤 1 或 2 中中断：**
重新将主动轴运行至参考点（参见第 1 步）
- **第 3 步中断：**
只要龙门轴仍然处于回参考点状态（接口信号“已回参考点/同步”= 1），则可通过接口信号“同步龙门轴组”重新启动龙门同步运行。

重新启动龙门同步运行

只能在下列条件下通过接口信号“启动龙门同步运行”启动龙门轴同步过程：

- 必须激活 JOG/REF 机床功能。必须设置好下列接口信号：
`<HmiModeGroup>.basic.in.refRequested = 1`（有效机床功能 REF）
- `<Axis>.coupling.in.gantrySyncOk = 0`（龙门轴组已同步）
- 轴组的各轴须位于公差窗口内：
`<Axis>.coupling.in.gantrySyncRunReady = 1`（龙门同步运行启动准备）
- NC 通道中没有任何轴回参考点
`<Chan>.basic.in.refPointApproachActive = 0`（回参考点激活）

若不是通过接口信号“启动龙门同步运行”从回参考点过程中启动龙门同步运行，则不会为同步轴指定参考位置作为目标位置：

MD34100 REFP_SET_POS（使用距离编码系统时的参考点值/目标点）

而是指定**主动轴的当前实际位置**作为目标位置，且在未进行轴耦合的情况下运行至此位置。

说明

可通过以下 NC/PLC 接口信号锁定主动轴的自动同步：

`<Chan>.toolMgmt.out.deactivateWorkpieceCounter = 1`（无自动同步运行）

在轴还没有获得使能的情况下，此操作总是有效的。此时也应针对性地启动同步运行，使用 NC/PLC 接口信号：

`<Chan>.basic.out.ptpTravers = 1`（启动龙门同步运行）

同步丢失

出现以下情况时，龙门组同步丢失：

- “跟随” 龙门轴
- 龙门轴的参考位置丢失，比如由于“驻留”（未激活任何测量系统）
- 龙门轴重新回参考点
- 拆解龙门轴组，通过：
`MD37140 $MA_GANTRY_BREAK_UP = 0`（松开龙门轴组）

相应的 NC/PLC 接口信号将复位：

- `<Axis>.basic.in.enc1Synchronized`
或 `<Axis>.basic.in.enc2Synchronized == 0`（已回参考点/已同步 1 或 2）
- `<Axis>.coupling.in.gantrySyncOk == 0`（龙门轴组已同步）

若在运行过程中由于故障原因导致龙门轴组同步丢失，可用以下 NC/PLC 接口信号重新启动同步运行：

`<Chan>.basic.out.ptpTravers == 1`（启动龙门同步运行）

前提是以下内容适用于龙门轴组的所有轴：

`<Axis>.basic.in.enc1Synchronized` 或
`<Axis>.basic.in.enc2Synchronized = 1`（已回参考点/已同步 1 或 2）

此时同步轴在未进行轴耦合的情况下运行至主动轴的当前实际位置。

如果在龙门轴组运行时设置“急停”信号(`<Nc>.basic.out.emergencyStop`)并再次复位，并且龙门轴的漂移量小于同步轴的静止公差，它们将自动重新同步。使用主动轴的 NC/PLC 接口信号可以禁止自动同步：

<Chan>.toolMgmt.out.deactivateWorkpieceCounter = 1（无自动同步运行）

参考点选择

为了保证龙门轴回参考点时的运行路径最短，以下机床数据中主动轴和同步轴的参考点数值应该相同：

MD34100 REFP_SET_POS（使用距离编码系统时的参考点值/目标点）

应通过机床数据考虑各轴的零脉冲与参考点之间的距离偏差：

MD34080 \$MA_REFP_MOVE_DIST（参考点距离）

MD34090 \$MA_REFP_MOVE_DIST_CORR（参考点偏移/绝对偏移）

选择回参考点方向

可通过以下机床数据确定同步轴的零脉冲搜索方向：

MD37150 \$MA_GANTRY_FUNCTION_MASK，位 1

位	值	含义
1	0	同步轴的零脉冲搜索方向类似机床数据： MD34010 \$MA_REFP_CAM_DIR_IS_MINUS
	1	同步轴的零脉冲搜索方向与主动轴相同

在回参考点过程的同步补偿运行中，主动轴的参考点数值将作为龙门轴组各轴的目标位置，并在未进行轴耦合的情况下运行至此位置。当主动轴上有绝对编码器和间距编程编码器时，将选择性地运行至主动轴的当前实际位置或参考点数值，可通过以下机床数据进行设置：

MD34330 \$MA_REFP_STOP_AT_ABS_MARKER（无目标点的距离编码长度测量系统）

若主动轴和同步轴只使用了一个回参考点挡块，则必须在 PLC 用户程序中考虑到此点。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.traversKeyMinus	LBP_Axis*.A_Minus	DB31, ...DBX4.6
<Axis>.basic.out.traversKeyPlus	LBP_Axis*.A_Plus	DB31, ...DBX4.7
<HmiModeGroup>.basic.in.refRequested	LBP_HMI.E_MMC_REF	DB11.DBX5.2
<Chan>.basic.out.ptpTravers	LBP_Axis*.A_PTP_Travel	DB31, ...DBX29.4

12.3 龙门轴的回参考点和同步

Basic Program Plus	Basic Program	
<Chan>.toolMgmt.out.deactivateWorkpieceCounter	LBP_Axis*.A_DisablWP_Counter	DB31, ...DBX29.5
<Nc>.basic.out.emergencyStop	LBP_NC.A_EMERGENCY	DB10.DBX56.1

NC → PLC

Basic Program Plus	Basic Program	
<Chan>.basic.in.refPointApproachActive	LBP_Chan*.E_Ref	DB21, ...DBX33.0
<Axis>.basic.in.enc1Synchronized	LBP_Axis*.E_RefSyn1	DB31, ...DBX60.4
<Axis>.basic.in.enc2Synchronized	LBP_Axis*.E_RefSyn2	DB31, ...DBX60.5
<Axis>.basic.in.moduloLimitSwitchActive	LBP_Axis*.E_ModLimEnAct	DB31, ...DBX74.4
<Axis>.coupling.in.gantrySyncRunReady	LBP_Axis*.E_SynRunStart	DB31, ...DBX 101.4
<Axis>.coupling.in.gantrySyncOk	LBP_Axis*.E_GantGroupSyn	DB31, ...DBX 101.5

OP → PLC

Basic Program Plus	Basic Program	
<HmiModeGroup>.basic.in.refRequested	LBP_HMI.E_MMC_REF	DB11.DBX5.2

12.3.2 自动同步

可在以下模式中运行自动同步：

- 回参考点模式（参见“简介 (页 363)”章节）
- 其他模式，如下所述：

将龙门轴组切换至跟随模式时，对主动轴和同步轴之间实际值的监控处于未激活状态。因此轴组不再同步。无论轴的位置如何，以下接口信号都设置为 0（由主动轴）

<Axis>.coupling.in.gantrySyncOk（龙门轴组已同步）

若将龙门轴组从跟随状态切换至位置控制状态，只要主动轴和同步轴位置之间实际值监控所得差值小于以下机床数据中确定的数值，则自动恢复同步过程：

MD36030 \$MA_STANDSTILL_POS_TOL（静止公差）

此种情况下，将不会进行插补，而为同步轴确定一个新的设定值。之前确定的差值将通过位置控制器进行修正。此过程中，只有同步轴在移动。

同步轴的此种移动过程与其从“停止”状态中切换至位置控制模式后的运行类似。此种情况下，只要停止监控没有在其间触发报警 25040（报警后会进行跟随操作），则位置控制器会对停止前所给出的位置进行重新设置。

这种自动同步模式下会使用与停止监控相同的公差窗口：

MD36030 \$MA_STANDSTILL_POS_TOL（静止公差）

根据参数组设置，使用机床数据：

MD36012 \$MA_STOP_LIMIT_FACTOR（粗准停/精准停和停机系数）

说明

除回参考点模式外，以下接口信号可在各种运行方式中锁定自动同步：

<Axis>.coupling.out.gantryDisableSync（无自动同步运行）

如果要在此处激活自动同步，则必须指定以下接口信号：

<Axis>.coupling.out.gantryDisableSync = 0（无自动同步运行）

随之应将龙门轴组的一个轴从跟随模式切换至位置控制运行模式。可通过接口信号实现此操作：

<Axis>.basic.out.followUpMode = 1（跟随模式）

<Axis>.basic.out.enableCtrl = 1（伺服使能）

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.followUpMode	LBP_Axis*.A_Follow_upMode	DB31, ...DBX1.4
<Axis>.basic.out.enableCtrl	LBP_Axis*.A_ContrEnable	DB31, ...DBX2.1
<Axis>.coupling.out.gantryDisableSync	LBP_Axis*.A_NoAutoSync	DB31, ...DBX29.5

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.coupling.in.gantrySyncOk	LBP_Axis*.E_GantGroupSyn	DB31, ...DBX 101.5

12.3.3 特殊性

每个龙门轴的第 2 个位置测量系统

一个轴组的龙门轴上可以连接不同的位置测量系统。此外，每个龙门轴还可以处理两个位置测量系统，可以随时在其间切换：

`<Axis>.basic.out.enc1Activation`（位置测量系统 1）

`<Axis>.basic.out.enc2Activation`（位置测量系统 2）

位置实际值切换时的最大公差应选择小于龙门警告极限：

`MD36500 $MA_ENC_CHANGE_TOL`（位置实际值切换时的公差）

但是，这两个位置测量系统必须事先回参考点。在回参考点过程之前，必须选择相应的测量系统。过程与上述过程类似。

通道专用回参考点

龙门轴也可用以下接口信号使用通道特有方式回参考点：

`<Chan>.basic.out.refPointApproach`（激活回参考点功能）

对于通过通道特有方式回参考点的轴顺序将使用主动轴轴向机床数据的数值：

`MD34110 $MA_REFP_CYCLE_NR`（通道专用回参考点运行时的轴顺序）

如上所述，在到达主动轴的参考点后，同步轴将首先回参考点。

利用指令 G74 通过零件程序回参考点

也可使用指令 G74 通过零件程序启动龙门轴的回参考点和同步过程。此方法中只对主动轴的名称进行编程。其过程与通过轴特有方式回参考点类似。

带间距编码参考标记的位置测量系统

为了在回参考点运行中节省路径，可以在龙门轴上使用带间距编码参考标记的位置测量系统作为唯一或第 2 个测量系统。如此可保证测量系统根据较短的运行路径（比如 20 mm）回参考点。龙门轴回参考点的方式与普通增量测量系统中的方式相同。

更多信息：

回参考点 (页 35)

绝对值编码器

在同步补偿运行中，龙门轴组的各轴在未进行轴耦合的情况下运行至主动轴的参考点数值，可通过以下机床数据确定：

MD34100 REFP_SET_POS（使用距离编码系统时的参考点值/目标点）

当主动轴上有绝对编码器和间距编程编码器时，将选择性地运行至主动轴的当前实际位置或参考点数值，可通过以下机床数据进行设置：

MD34330 \$MA_REFP_STOP_AT_ABS_MARKER（无目标点的距离编码长度测量系统）

激活轴补偿

主动轴和同步轴上均可激活补偿功能。龙门轴各轴的补偿值分开独立计算。因此，在调试时须确定主动轴和同步轴的补偿值并输入。

只有在轴已回参考点或龙门轴组已同步时，控制系统内部的补偿功能才有效。适用原则：

补偿方式	有效时间	PLC 接口信号
反向间隙补偿	轴已回参考点	“已回参考点/已同步”
SSFK	轴已回参考点	“已回参考点/已同步”
垂度补偿	龙门轴组已回参考点	“龙门轴组已同步”
温度补偿	龙门轴组已回参考点	“龙门轴组已同步”

若同步轴的运行是由于激活了补偿功能，则其会显示一个运行指令，此指令与主动轴无关。

监控生效

龙门轴与普通 NC 轴一样，只有在到达参考点（接口信号“已回参考点/已同步”）后下列监控才生效：

- 工作区域限制
- 软件限位开关
- 保护区

同步轴的轴向机床数据值也可作为监控极限值使用。

多通道式程序段搜索

在程序测试模式下通过跨通道程序段搜索（SERUPRO“Search-Run by Program test”）可以模拟运行龙门轴组。

说明

有关多通道式程序段搜索 SERUPRO 的更多信息请参见：

更多信息

功能手册之基本功能：运行方式组、通道、程序运行、程序测试

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.enc1Activation	LBP_Axis*.A_PosMeas1	DB31, ...DBX1.5
<Axis>.basic.out.enc2Activation	LBP_Axis*.A_PosMeas2	DB31, ...DBX1.6
<Chan>.basic.out.refPointApproach	LBP_Chan*.A_Ref	DB21, ...DBX1.0

12.4 调试龙门轴

概述

由于通常情况下龙门轴的主动轴和同步轴之间存在强制耦合，故在调试龙门轴组时必须像仅操作一个轴单位一样进行。因此，必须共同确定并输入主动轴和同步轴的轴向机床数据。

如果同步轴由于动态响应比主动轴弱而过载，则会显示报警 10656。

以下将叙述龙门轴调试中需特别注意的地方。

运行方向

进行龙门轴调试时，必须监控电机的转动方向是否与所需的轴运行方向一致。必须通过如下轴向机床数据进行补偿：

MD32100 \$MA_AX_MOTION_DIR（运行方向）

输入龙门关闭极限

为了监控同步轴与主动轴实际位置之间的位置实际值偏差，必须根据机床生产商的相关说明输入主动轴和同步轴的关闭极限值：

- MD37120 \$MA_GANTRY_POS_TOL_ERROR（龙门关闭极限）
- MD37130 \$MA_GANTRY_POS_TOL_REF（回参考点时的龙门关闭极限）

说明

然后关闭控制系统再重新启动，因为龙门轴定义和龙门关闭极限值必须在重启 (POWER ON) 后才生效。

控制和故障特性

由于数字驱动具有良好的控制和故障特性，因此无需在龙门轴之间进行补偿控制。但龙门轴完全同步运行的前提是：主动轴和同步轴控制回路的参数必须设置为**动态一致**。

为了达到最好的同步效果，主动轴和同步轴的动态响应必须相同，以确保良好的**控制特性**。必须分别**优化**各轴向控制回路（位置、转速和电流控制器），以尽快且有效地排除故障。额定值回路中的**动态响应自适应**功能可在不减损调节质量的前提下匹配不同类型的轴动力。

轴向优化

对于对称结构的龙门轴，建议为一个轴组的所有轴设置相同的 K_v 系数和前馈控制设置。

必须轴向优化主动轴和同步轴的下列调节参数设置：

- MD32200 \$MA_POSCTRL_GAIN（KV 系数）
- MD32620 FFW_MODE（前馈控制参数）
- MD32610 \$MA_VELO_FFW_WEIGHT（速度/转速前馈控制系数）
- MD32650 \$MA_AX_INERTIA（转矩前馈控制的转动惯量）
- MD32800 \$MA_EQUIV_CURRCTRL_TIME（前馈控制的电流控制回路等效时间常数）
- MD32810 \$MA_EQUIV_SPEEDCTRL_TIME（前馈控制的转速控制回路等效时间常数）

更多信息

功能手册之监控与补偿：补偿

相同设置

主动轴和同步轴的下列调节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)
- MD32420 \$MA_JOG_AND_POS_JERK_ENABLE (轴向最大急动度的初始设置)
- MD32430 \$MA_JOG_AND_POS_MAX_JERK (轴向急动)

更多信息

功能手册之基本功能：速度/设定值/实际值系统，控制

动态响应自适应功能

为了实现控制特性，主动轴和已耦合同步轴的动态响应必须相同。动态响应相同的意思是：相同转速下其滞后量必须相同。

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各轴（控制回路）之间的控制特性。动态响应自适应的时间常量指的是输入与其它相应轴相比动力“最弱”轴的额外时间常量的差值。

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。

主动轴

MD32810 \$MA_EQUIV_SPEEDCTRL_TIME [n] = 5ms (前馈控制的转速控制回路等效时间常数)

同步轴

MD32810 \$MA_EQUIV_SPEEDCTRL_TIME [n] = 3ms

- 同步轴动态响应自适应功能的时间常数：
MD32910 \$MA_DYN_MATCH_TIME [n] = 5ms - 3ms = 2ms (动态响应自适应功能的时间常数)

必须使用以下机床数据轴向激活动态响应自适应功能：

MD32900 \$MA_DYN_MATCH_ENABLE (动态响应自适应功能)

动态响应自适应功能监控：

相同速度下主动轴和同步轴的滞后量必须相同！

若需进行微调，可稍微调整 K_v 系数或前馈参数，以达到最佳的效果。

龙门轴回参考点

首先，主动轴和同步轴的参考点位置设置应大致相同。

为了使龙门轴同步补偿运行不会自动启动，首次调试时须在其回到参考点前将龙门警告极限设置为值 0：

MD37100 \$MA_GANTRY_POS_TOL_WARNING（龙门轴定义）

如此设置后，警告消息在运行过程中无效。

若主动轴和同步轴的瞬时倾斜位置对驱动有过大的附加力矩影响，则必须在运行前较准龙门轴组。然后将龙门轴回参考点。参见：

- 章节“龙门轴的回参考点和同步 (页 363)”
- 更多信息：回参考点 (页 35)

主动轴和同步轴均回到参考点后，必须通过比较位置实际值得出两者之间的差值且需考虑参考点偏移：

- MD34080 \$MA_REFP_MOVE_DIST（参考点距离）
- MD34090 \$MA_REFP_MOVE_DIST_CORR（参考点偏移/绝对偏移）

另外应为每个龙门轴确定零脉冲与参考点之间的距离偏差。必须使用以下机床数据对其进行调整，以使主动轴和同步轴的位置实际值在补偿运动后相同：

- MD34080 \$MA_REFP_MOVE_DIST（参考点距离）
- MD34090 \$MA_REFP_MOVE_DIST_CORR（参考点偏移/绝对偏移）

龙门轴同步

龙门同步运行必须通过 NC/PLC 接口信号激活（参见“龙门轴的回参考点和同步 (页 363)”章节）：

`<Axis>.coupling.out.gantryStartSyncRun = 1`（龙门同步）

同步过程的结束通过以下 NC/PLC 接口信号显示：

`<Axis>.coupling.in.gantrySyncOk == 1`（龙门轴组已同步）

同步过程结束后，必须检查主动轴和同步轴之间的偏移量是否都是 0。必要时，使用上述机床数据进行修正。

输入龙门警告极限

将主动轴和同步轴的参考点数值进行优化设置后，为了保证龙门轴同步后相互间无倾斜，必须在以下机床数据中为主动轴和同步轴输入警告极限值：

MD37110 \$MA_GANTRY_POS_TOL_WARNING（龙门警告极限）

此时，应逐步提高其数值，保证刚好无消息（已超过极限值）出现。当然，首先必须监控加速阶段。

还可通过此极限值确定在达到何种程度的位置偏差后内部控制系统自动启动龙门同步运行。

得出补偿并激活

若龙门轴需要补偿（反向间隙、垂度、温度或丝杠螺距误差补偿），则必须确定主动轴和同步轴的修正数值并将其输入相应的参数或表格中。

更多信息

功能手册之监控与补偿

功能发生器/测量功能

在同步轴上激活功能发生器和测量功能时，会出现错误消息而中止。若需要在同步轴上激活此功能（比如为了测量机床），则必须事先交换主动轴和同步轴。

特殊情况

若需要激活**单个**轴的此功能，则必须事先拆分龙门轴组。由于拆分后其它的轴将不再同步运行，所以必须保证已激活轴不再作为位置公差运行。

也就是说，在拆分龙门轴组后：

- 始终激活运行范围极限值并将其设置为较小的值（位置公差）
- 尽可能地同步龙门轴组，然后执行通电复位操作，**避免**重新回参考点。如此操作后，运行范围极限值总是指的相同位置（通电时的位置）
- 避免阶跃函数。只允许小于公差的位置阶跃存在。
- 与普通轴相反，对于功能发生器和测量功能中的补偿值我们推荐始终设置为 0。
- 功能发生器和测量功能的振幅应较小，保证已激活轴的运行距离少于位置公差。为了实施监控，必须一直激活运行范围极限值（参见上面所述）。

更多信息

SINAMICS S120 驱动功能功能手册；转速控制回路 (DD2)

龙门轴组的调试帮助

功能发生器和测量这两个调试功能通过PI服务设置参数。在JOG运行模式下，通过“启动NC”运行已设置参数的所有轴。

在操作面板“龙门轴组的测量功能和功能发生器”中会显示一幅图象，可在其中输入两个振幅数值、一个补偿值和一个带宽值。第一个振幅数值适用于测量轴，另一个数值适用于其它的耦合轴。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.coupling.out.gantryStartSyncRun	LBP_Axis*.A_StartGantSynRun	DB31, ...DBX29.4

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.coupling.in.gantrySyncOk	LBP_Axis*.E_GantGroupSyn	DB31, ...DBX 101.5

12.5 参数设置：故障情况中的特性

脉冲封锁

龙门轴组在触发脉冲封锁的错误方面的特性可以通过以下轴专用机床数据设置：

12.6 龙门轴中的 PLC 接口信号

MD30455 \$MA_MISC_FUNCTION_MASK, 位 9 = <值>

<值>	含义
0	如果发生触发脉冲封锁的错误（例如测量电路错误），那么龙门轴组的所有其他轴上的脉冲也会被封锁。 结果：龙门轴组的所有轴惯性滑行。
1	如果发生触发脉冲封锁的错误（例如测量电路错误），那么仅封锁该轴的脉冲。龙门轴组的其他轴将停止。这些轴的脉冲随后也会被封锁。为确保不会过早封锁这些轴的脉冲，必须考虑以下驱动参数，特别是对于要保持在静止位置的轴（垂直轴）： • p1135[0...n] OFF3 斜坡下降时间 • p1217 电机抱闸的闭合时间 • p1227 静止状态检测监控时间 • p1228 脉冲封锁延迟时间 有关驱动参数的详细说明请见： 更多信息 SINAMICS S120/S150 参数手册

12.6 龙门轴中的 PLC 接口信号

龙门轴的特有接口信号

已耦合龙门轴的特有 NC/PLC 接口信号通过主动轴或同步轴的轴向 NC/PLC 接口管理。下表中为全部的龙门特有 NC/PLC 接口信号以及其是否会被主动轴或同步轴使用的标识。

NC/PLC 接口信号	传输方向	<Axis>...	主动轴	同步轴
启动龙门同步运行	PLC → NC	<Axis>.coupling.out.gantryStartSyncRun	X	
无自动同步运行	PLC → NC	<Axis>.coupling.out.gantryDisableSync	X	
龙门轴	NC → PLC	<Axis>.coupling.in.gantryActive	1	1
龙门主动轴	NC → PLC	<Axis>.coupling.in.gantryLeadingActive	1	0

NC/PLC 接口信号	传输方向	<Axis>...	主动轴	同步轴
龙门轴组已同步	NC → PLC	<Axis>.coupling.in.gantrySyncOk	X	
龙门同步运行启动准备	NC → PLC	<Axis>.coupling.in.gantrySyncRunReady	X	
超出龙门警告极限	NC → PLC	<Axis>.coupling.in.gantryWarningLimitExceeded		X
超出龙门关闭极限	NC → PLC	<Axis>.coupling.in.gantryCutOffLimitExceeded		X

龙门轴轴向接口信号的作用

a) 从 PLC 至轴的轴向接口信号 (PLC → NC)

一般说来，从 PLC 至轴的轴向接口信号均涉及龙门轴组的所有轴。此时，龙门各轴（主动轴和同步轴）处于平等位置。

比如：若将主动轴的如下接口信号设置为“0”，则可同时停止龙门轴组的所有轴：

<Axis>.basic.out.enableCtrl（伺服使能）

下表中为各个接口信号（从 PLC 至轴）对龙门轴的作用：

NC/PLC 接口信号	<Axis>...	作用于	
		主动轴	同步轴
进给轴/主轴禁止	<Axis>.basic.out.disable	龙门轴组的所有轴	无效
位置测量系统 1/2	<Axis>.basic.out.enc1Activation <Axis>.basic.out.enc1Activation	轴向 ¹⁾	轴向 ¹⁾
伺服使能	<Axis>.basic.out.enableCtrl	龙门轴组的所有轴 ²⁾	
删除剩余行程（轴向）	<Axis>.basic.out.resetMovement	轴向	无效
正在进行夹紧	<Axis>.basic.out.clampingInProgress	轴向	轴向
参考点值 1 - 4	<Axis>.basic.out.refPointSelection	轴向	轴向
进给停止	<Axis>.basic.out.feedStopSpindleStop	龙门轴组的所有轴	
正/负硬件限位开关	<Axis>.basic.out.hwLimitSwitchMinus <Axis>.basic.out.hwLimitSwitchPluses	轴向报警；对龙门轴组的所有轴发出制动请求	

12.6 龙门轴中的 PLC 接口信号

NC/PLC 接口信号	<Axis>...	作用于	
		主动轴	同步轴
第 2 个正/负软件限位开关	<Axis>.basic.out.swLimitSwitch2Minus <Axis>.basic.out.swLimitSwitch2Plus	轴向	轴向
斜坡功能发生器快速停止 (HLGSS)	<Axis>.drive.out.rapidStop	龙门轴组的所有轴	
选择驱动数据组	<Axis>.drive.out.driveDataSetSelection	轴向	轴向
脉冲使能	<Axis>.drive.out.enablePulses	轴向	轴向
¹⁾ <Axis>.basic.out.enc1Activation 和 <Axis>.basic.out.enc2Activation (位置测量系统 1/2) 位置测量系统 1 和 2 之间的切换要针对每个龙门轴单独考虑。但对于所有龙门轴，要共同考虑两个位置测量系统的无效切换 (所谓的驻留位置)。 ²⁾ <Axis>.basic.out.enableCtrl (伺服使能) 取消一个龙门轴的伺服使能时，龙门轴组的所有轴同时停止。所有龙门轴的停止方式 (例如通过快速停止) 都是相同的。			

在龙门轴中，要么其“跟随”状态 (某个龙门轴的接口信号= 1)，要么其“停止”状态 (所有龙门轴的接口信号= 0) 的激活取决于如下接口信号：

<Axis>.basic.out.followUpMode (跟随模式)

b) 从轴至 PLC 的轴向接口信号 (NC → PLC)

一般说来，从轴至 PLC 的轴向接口信号由主动轴和同步轴分别设置并发送至 PLC。

示例：

<Axis>.basic.in.enc1Synchronized 或
<Axis>.basic.in.enc2Synchronized (已回参考点/已同步 1/2)。

例外：

主动轴运行时，也可在同步轴上设置接口信号：

<Axis>.basic.in.traversMinusCommandActive 和
<Axis>.basic.in.traversPlusCommandActive (运行指令 + 或 -)

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.disable	LBP_Axis*.A_ASpDisable	DB31, ...DBX1.3
<Axis>.basic.out.followUpMode	LBP_Axis*.A_Follow_upMode	DB31, ...DBX1.4
<Axis>.basic.out.enc1Activation	LBP_Axis*.A_PosMeas1	DB31, ...DBX1.5
<Axis>.basic.out.enc2Activation	LBP_Axis*.A_PosMeas2	DB31, ...DBX1.6
<Axis>.basic.out.enableCtrl	LBP_Axis*.A_ContrEnable	DB31, ...DBX2.1
<Axis>.basic.out.resetMovement	LBP_Axis*.A_DeIDTGSpReset	DB31, ...DBX2.2
<Axis>.basic.out.clampingInProgress	LBP_Axis*.A_Clamp	DB31, ...DBX2.3
<Axis>.basic.out.refPointSelection	LBP_Axis*.A_RefVal1..4	DB31, ...DBX2.4 - 2.7
<Axis>.basic.out.feedStopSpindleStop	LBP_Axis*.A_FDSpStop	DB31, ...DBX4.3
<Axis>.basic.out.traversKeyMinus	LBP_Axis*.A_Minus	DB31, ...DBX4.6
<Axis>.basic.out.traversKeyPlus	LBP_Axis*.A_Plus	DB31, ...DBX4.7
<Axis>.basic.out.hwlLimitSwitchMinus	LBP_Axis*.A_HWLlimitMinus	DB31, ...DBX12.0
<Axis>.basic.out.hwlLimitSwitchPlus	LBP_Axis*.A_HWLlimitPlus	DB31, ...DBX12.1
<Axis>.basic.out.swLimitSwitch2Minus	LBP_Axis*.A_SWLimit2Minus	DB31, ...DBX12.2
<Axis>.basic.out.swLimitSwitch2Plus	LBP_Axis*.A_SWLimit2Plus	DB31, ...DBX12.3
<Axis>.drive.out.rapidStop	-	DB31, ...DBX20.1
<Axis>.drive.out.driveDataSetSelection	LBP_Axis*.A_ParA..C; A_MotA..B	DB31, ...DBX21.0.. 2
<Axis>.drive.out.enablePulses	LBP_Axis*.A_PulseEnable	DB31, ...DBX21.7
<Axis>.coupling.out.gantryStartSyncRun	LBP_Axis*.A_StartGantSynRun	DB31, ...DBX29.4
<Axis>.coupling.out.gantryDisableSync	LBP_Axis*.A_NoAutoSync	DB31, ...DBX29.5

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.enc1Synchronized	LBP_Axis*.E_RefSyn1	DB31, ...DBX60.4
<Axis>.basic.in.enc2Synchronized	LBP_Axis*.E_RefSyn2	DB31, ...DBX60.5
<Axis>.basic.in.traversMinusCommandActive	LBP_Axis*.E_TCMinus	DB31, ...DBX64.6
<Axis>.basic.in.traversPlusCommandActive	LBP_Axis*.E_TCPlus	DB31, ...DBX64.7

12.7 龙门轴的其它事项

Basic Program Plus	Basic Program	
<Axis>.basic.in.moduloLimitSwitchActive	LBP_Axis*.E_ModLimEnAct	DB31, ...DBX74.4
<Axis>.coupling.in.gantryCutOffLimitExceeded	LBP_Axis*.E_GantCOLimit	DB31, ...DBX 101.2
<Axis>.coupling.in.gantryWarningLimitExceeded	LBP_Axis*.E_GantWarn	DB31, ...DBX 101.3
<Axis>.coupling.in.gantrySyncRunReady	LBP_Axis*.E_SynRunStart	DB31, ...DBX 101.4
<Axis>.coupling.in.gantrySyncOk	LBP_Axis*.E_GantGroupSyn	DB31, ...DBX 101.5
<Axis>.coupling.in.gantryLeadingActive	LBP_Axis*.E_GantLeadAxis	DB31, ...DBX 101.6
<Axis>.coupling.in.gantryActive	LBP_Axis*.E_GantAxis	DB31, ...DBX 101.7

12.7 龙门轴的其它事项

手动运行

在 JOG 运行模式下，不能直接手动运行同步轴。按同步轴的运行键时，会被内部控制系统忽略。同样，旋转同步轴的手动轮时也无效。

手轮叠加

各轴已耦合时，只能在主动轴上通过手轮实现倍率运行。此时，同步轴随同主动轴一起运行。

DRF 偏移

只能在主动轴上使用 DRF 偏移。此时，同步轴随同主动轴一起运行。

零件程序中的编程

在零件程序中，只能对龙门轴组的主动轴进行编程。对同步轴进行编程时，会出现报警，即使龙门轴组已拆分时仍如此 (MD37140 \$MA_GANTRY_BREAK_UP = 1)。

PLC 或指令轴

只有龙门轴组的主动轴可以由 PLC 使用 FC 18 运行，或者作为指令轴通过同步动作运行。

更多信息

- PLC 功能手册
- 功能手册之同步动作分册

PRESET（预设）

只能在主动轴上使用 PRESET（预设）功能。使用 PRESET（预设）功能后，内部控制系统会对龙门轴组的所有轴进行重新评估。因此，龙门轴会丢失其参考和同步：

`<Axis>.coupling.in.gantrySyncOk`（龙门轴组已同步）= 0

龙门轴的通道分配

需要注意的是，对于主动轴在多个通道中已知的龙门轴组，其同步轴在这些通道中也是已知的。如果不是这样，就会发出报警 10651，投诉原因为 60XX（XX 是被投诉的龙门轴组）。

跨通道取轴

使用 RELEASE（主动轴）会自动释放龙门轴组的所有轴。

只有在接收通道中已知一个封闭式龙门轴组所有轴时，才能对该轴组的主动轴进行跨通道取轴，否则会报告报警 10658。

当试图重新关闭以 MD37140 \$MA_GANTRY_BREAK_UP = 1 松开的龙门轴组时，不进行自动跨通道取轴，也不自动调整龙门轴的轴状态。用户自行对此负责。但是，在中断后会检查轴状态，必要时会发出相应的报警 10658。

说明

如果要再次关闭一个龙门轴组，用户必须确保该轴组的所有具有相应轴状态的轴都在一个通道中。

复位时的缺省设置

龙门轴组激活时，会忽略同步轴的下列机床数据参数设置：

MD30450 \$MA_IS_CONCURRENT_POS_AX = 1（复位时的缺省设置：中性轴/通道轴）

默认为主动轴的状态。用户可通过显示报警 4300 获悉无意义的设计。

显示数据

位置实际值显示中会显示主动轴和同步轴的实际值。操作区域“诊断”中的服务显示数值也会进行实际值显示。

软件限位开关

软件限位开关监控仅适用于主动轴。超过时，龙门轴组的所有轴均会停止。

与“联动”功能的区别

- “龙门轴”和“联动”功能之间存在以下本质区别：
- 龙门轴的轴耦合必须始终存在。因此无法通过零件程序将龙门轴的轴耦合分开。相反，联动组可以通过零件程序拆分，轴可以单独运行。
 - 借助“龙门轴”功能，可以持续监控主动轴和同步轴的位置实际值之间的差异，并在出现不允许的偏差时停止运行。使用“联动”功能时不进行监控。
 - 回参考点时，龙门轴也必须保持耦合。因此，对于龙门轴的回参考点运行存在特殊过程。而联动轴则单独回参考点。
 - 为了使龙门轴在运行时不出现机械偏移，同步轴的控制动态响应设置必须与主动轴相同。而使用“联动”功能时，轴也可以采用不同的控制动态响应。

更多信息：
联动 (页 571)

耦合生效时的程序段搜索

说明
我们建议在耦合生效时仅采用搜索类型 5 “通过程序测试进行程序段搜索”(SERUPRO) 来搜索程序段。

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.coupling.in.gantrySyncOk	LBP_Axis*.E_GantGroupSyn	DB31, ... DBX 101.5

12.8 示例

12.8.1 创建龙门轴组

简介

设置龙门轴组、将其轴回参考点、调整可能的偏移并最终同步相关轴是一个更复杂的过程。下例中描述了所需的单个步骤。

位置分布

机床轴 1 = 龙门主动轴，增量测量系统

机床轴 3 = 龙门同步轴，增量测量系统

机床数据

下列机床数据所述为程序开始时的初始值。以后可根据下列设置修正或补充单个数值。

龙门机床数据

轴 1

MD37100 \$MA_GANTRY_AXIS_TYPE = 1（龙门轴定义）

MD37110 \$MA_GANTRY_POS_TOL_WARNING = 0（龙门警告极限）

MD37120 \$MA_GANTRY_POS_TOL_ERROR = 如 1 mm（龙门关闭极限）

MD37130 \$MA_GANTRY_POS_TOL_REF = 如 100 mm（最大倾斜位置）（回参考点时的龙门关闭极限）

MD37140 \$MA_GANTRY_BREAK_UP = 0（松开龙门轴组）

轴 3

MD37100 \$MA_GANTRY_AXIS_TYPE = 11

MD37110 \$MA_GANTRY_POS_TOL_WARNING = 0

MD37120 \$MA_GANTRY_POS_TOL_ERROR = 如 1mm

MD37130 \$MA_GANTRY_POS_TOL_REF = 如 100 mm（最大倾斜位置）

MD37140 \$MA_GANTRY_BREAK_UP = 0

参考点机床数据（用于第一个编码器）

轴 1

MD34000 \$MA_REFP_CAM_IS_ACTIVE = TRUE

MD34010 \$MA_REFP_CAM_DIR_IS_MINUS = 如 FALSE

MD34020 \$MA_REFP_VELO_SEARCH_CAM =

MD34030 \$MA_REFP_MAX_CAM_DIST = 与最大运行距离相对应

MD34040 \$MA_REFP_VELO_SEARCH_MARKER =

MD34050 \$MA_REFP_SEARCH_MARKER_REVERSE = 如 FALSE

MD34060 \$MA_REFP_MAX_MARKER_DIST = 凸轮工作面和零标记之间的差值

MD34070 \$MA_REFP_VELO_POS =

MD34080 \$MA_REFP_MOVE_DIST = 0

MD34090 \$MA_REFP_MOVE_DIST_CORR = 0

MD34092 \$MA_REFP_CAM_SHIFT = 0

MD34100 \$MA_REFP_SET_POS = 0

MD34200 \$MA_ENC_REFP_MODE = 1

必须相应地指定轴 3 的参考点机床数据（用于第一个编码器）。

12.8.2 NC-PLC 接口设置

引言

在轴回参考点的过程中必须首先阻止自动同步运行，以避免损坏未较准的轴组。

防止自动同步

PLC 用户程序为轴 1 的轴数据块设置以下内容：

<Axis>.coupling.out.gantryStartSyncRun = 0（不启动龙门同步）

<Axis>.coupling.out.gantryDisableSync = 1（无自动同步运行）

NC 在轴 1 的轴块中设置以下内容并确认：

<Axis>.coupling.in.gantrySyncRunReady = 0（同步运行未做好启动准备）

<Axis>.coupling.in.gantryLeadingActive = 1（主动轴 LA）

<Axis>.coupling.in.gantryActive = 1（龙门轴）

PLC 用户程序为轴 3 的轴数据块设置以下内容：

<Axis>.coupling.out.gantryStartSyncRun = 0（不启动龙门同步）

NC 在轴 3 的轴块中设置以下内容并确认：

<Axis>.coupling.in.gantrySyncRunReady = 0（同步运行未做好启动准备）

<Axis>.coupling.in.gantryLeadingActive = 0（同步轴 GA）

<Axis>.coupling.in.gantryActive = 1（龙门轴）

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.coupling.out.gantryStartSyncRun	LBP_Axis*.A_StartGantSynRun	DB31, ...DBX29.4
<Axis>.coupling.out.gantryDisableSync	LBP_Axis*.A_NoAutoSync	DB31, ...DBX29.5

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.coupling.in.gantrySyncRunReady	LBP_Axis*.E_SynRunStart	DB31, ...DBX 101.4
<Axis>.coupling.in.gantryLeadingActive	LBP_Axis*.E_GantLeadAxis	DB31, ...DBX 101.6
<Axis>.coupling.in.gantryActive	LBP_Axis*.E_GantAxis	DB31, ...DBX 101.7

12.8.3 开始调试

回参考点

应执行下列步骤：

- 选择“REF”运行方式
- 启动轴 1（主动轴）回参考点
- 一直等到出现消息“10654 通道 1 等待启动同步过程”。

与此同时，NC 已做好轴 1 的同步准备并报告给接口信号：

```
<Axis>.coupling.in.gantrySyncRunReady = 1（同步运行启动准备）
```

```
<Axis>.coupling.in.gantryLeadingActive = 1（主动轴 LA）
```

```
<Axis>.coupling.in.gantryActive = 1（龙门轴）
```

除此之外，还应执行如下操作：

- 复位
- 读出机床坐标系中的数值：
例如
X = 0.941
Y = 0.000
XF = 0.000
- 以相反的符号将主动轴 1 的 X 值填入跟随轴 3 的机床数据中：
MD34090 \$MA_REFP_MOVE_DIST_CORR = - 0.941（参考点偏移/绝对偏移）

说明

此机床数据上电后生效。为了避免需事先通电，也可将此数值输入如下机床数据中：

```
MD34080 $MA_REFP_MOVE_DIST（参考点距离）
```

此机床数据在复位后即生效。

- 轴 1（主动轴）根据已修正的机床数据重新返回参考点
- 一直等到出现消息“10654 通道 1 等待启动同步过程”
- 与此同时，NC 已做好轴 1 的同步准备并报告给接口信号：
 <Axis>.coupling.in.gantrySyncRunReady = 1（同步运行启动准备）
 <Axis>.coupling.in.gantryLeadingActive = 1（主动轴 LA）
 <Axis>.coupling.in.gantryActive = 1（龙门轴）

- 注意机床的实际位置。此时，可能出现 A 或 B 两种情况：

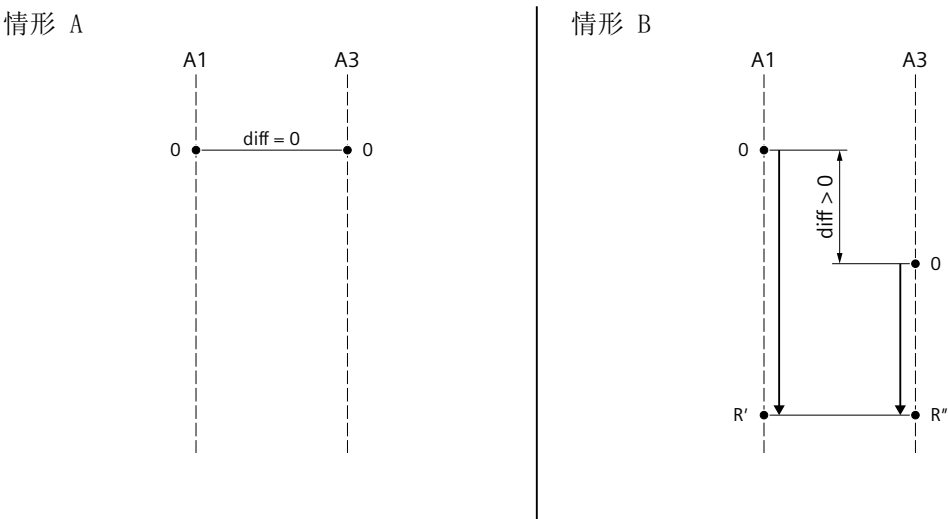


图 12-3 轴 1 回参考点后可能出现的结果（主动轴）

若为情况 A，则可立即启动同步运行（参见“启动同步”步骤）。若为情况 B，则必须得出偏移“差值”并对其予以考虑：

- 测量差值
- 通过机床底座（图片右侧）两个互为直角的参考点 R' 和 R'' 可得知 JOG 运行模式中的位置差。然后可读出偏移量，作为位置显示的差值。必须将此偏移差值填入轴 3（同步轴）的如下机床数据中：
MD34100 \$MA_REFP_SET_POS
继续第 1 个步骤（见上文）。
- 启动同步运行。PLC 设置：
`<Axis>.coupling.out.gantryStartSyncRun = 1`（龙门同步）

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<code><Axis>.coupling.out.gantryStartSyncRun</code>	<code>LBP_Axis*.A_StartGantSynRun</code>	<code>DB31, ...DBX29.4</code>

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.coupling.in.gantrySyncRunReady	LBP_Axis*.E_SynRunStart	DB31, ...DBX 101.4
<Axis>.coupling.in.gantryLeadingActive	LBP_Axis*.E_GantLeadAxis	DB31, ...DBX 101.6
<Axis>.coupling.in.gantryActive	LBP_Axis*.E_GantAxis	DB31, ...DBX 101.7

12.8.4 设置警告极限和误差限制

如果设置好龙门轴组且已同步，则还必须设置下列机床数据：

MD37110 \$MA_GANTRY_POS_TOL_WARNING（龙门警告极限）

MD37120 \$MA_GANTRY_POS_TOL_ERROR（龙门关闭极限）

步骤

- 先将各轴的如下机床数据设置为较大数值：
MD37120 \$MA_GANTRY_POS_TOL_ERROR（龙门关闭极限）
- 在如下机床数据中填入一个较小的数值：
MD37110 \$MA_GANTRY_POS_TOL_WARNING（龙门警告极限）
若此时各轴动态响应负载较大，则会不断发出报警“10652 通道 %1 轴 %2 超过龙门警告极限”，随之又自动消失。
- 现在增大以下机床数据的数值：
MD37110 \$MA_GANTRY_POS_TOL_WARNING（龙门警告极限）
继续重复以上操作，直到不出现报警消息为止。接口会显示状态：
<Axis>.coupling.in.gantryCutOffLimitExceeded = 0（未超出龙门关闭极限）
<Axis>.coupling.in.gantryWarningLimitExceeded = 0（未超出龙门警告极限）
<Axis>.coupling.in.gantrySyncRunReady = 0（同步运行未做好启动准备）
<Axis>.coupling.in.gantrySyncOk = 1（龙门轴组已同步）
<Axis>.coupling.in.gantryLeadingActive = 1（龙门主动轴）
<Axis>.coupling.in.gantryActive = 1（龙门轴）
如果监控只是偶尔进行，则可以在 PLC 用户程序中编程边缘标记。
- 在以下机床数据中填入为警告极限所得的数值和一个较小的安全撞击数值：
MD37120 \$MA_GANTRY_POS_TOL_ERROR（龙门关闭极限）

误差限制值

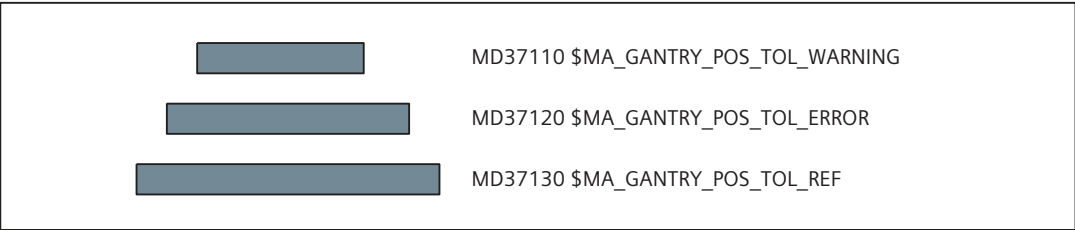
下列机床数据中已填有数值：

MD37110 \$MA_GANTRY_POS_TOL_WARNING（龙门警告极限）

MD37120 \$MA_GANTRY_POS_TOL_ERROR（龙门关闭极限）

MD37130 \$MA_GANTRY_POS_TOL_REF（回参考点时的龙门关闭极限）

匹配过程结束后，这些数值应符合如下大小比例关系：



说明

在调试各轴由**直线电机**和所属的测量系统驱动的龙门轴组时，应根据其要求运行。

若接口信号“龙门已同步”未处于排除等候处理状态，则机床数据 MD37110 和 MD37120 中输入的误差限制可相互作为引导轴和跟随轴之间实际值差值的额外公差（比如报警消除后未重新同步龙门轴组）。

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.coupling.in.gantryCutOffLimitExceeded	LBP_Axis*.E_GantCOLimit	DB31, ...DBX 101.2
<Axis>.coupling.in.gantryWarningLimitExceeded	LBP_Axis*.E_GantWarn	DB31, ...DBX 101.3
<Axis>.coupling.in.gantrySyncRunReady	LBP_Axis*.E_SynRunStart	DB31, ...DBX 101.4
<Axis>.coupling.in.gantrySyncOk	LBP_Axis*.E_GantGroupSyn	DB31, ...DBX 101.5
<Axis>.coupling.in.gantryLeadingActive	LBP_Axis*.E_GantLeadAxis	DB31, ...DBX 101.6
<Axis>.coupling.in.gantryActive	LBP_Axis*.E_GantAxis	DB31, ...DBX 101.7

主轴

13.1 简要说明

主轴主要用于使刀具或工件进入旋转运动，从而为切削加工创造前提条件。

为此，根据机床的类型主轴须支持以下功能：

- 设定主轴旋转方向（M3、M4）
- 设定主轴转速（S、SVC）
- 无定向主轴停止（M5）
- 定向主轴停止 / 主轴定位（SPOS、M19 和 SPOSA）
- 齿轮档切换（M40 至 M45）
- 主轴模式/轴模式（主轴作为回转轴，或反之）
- 螺纹切削（G33、G34、G35）
- 刚性攻丝（G331、G332）
- 带补偿夹具的攻丝（G63）
- 旋转进给率（G95）
- 恒定切削速度（G96、G961、G97、G971）
- 可编程的主轴转速限制（G25、G26、LIMS）
- 主轴或主轴电机上可安装位置测量编码器
- 主轴最小/最大转速以及最大编码器极限频率监控和主轴目标点监控
- 激活/取消位置闭环控制（SPCON、SPCOF、M70）
- 编写主轴功能：
 - 通过零件程序
 - 通过同步动作
 - 通过 PLC 使用 SINU_PositionSpindle 或 SINU_RotateSpindle (Basic Program - FC18)；或者通过用于简单主轴控制的特殊主轴接口

13.2 运行方式

13.2.1 概述

主轴为无限旋转的回转轴，其能够在各运行模式中在位置闭环控制或转速闭环控制下运行。

运行模式

- 控制模式（转速闭环控制）
- 往复模式（转速闭环控制）
- 定位模式（位置闭环控制）
- 同步模式
更多信息：同步主轴 (页 519)
- 刚性攻丝（G331 / G332）
更多信息：编程手册之 NC 编程
- 轴模式（位置闭环控制）

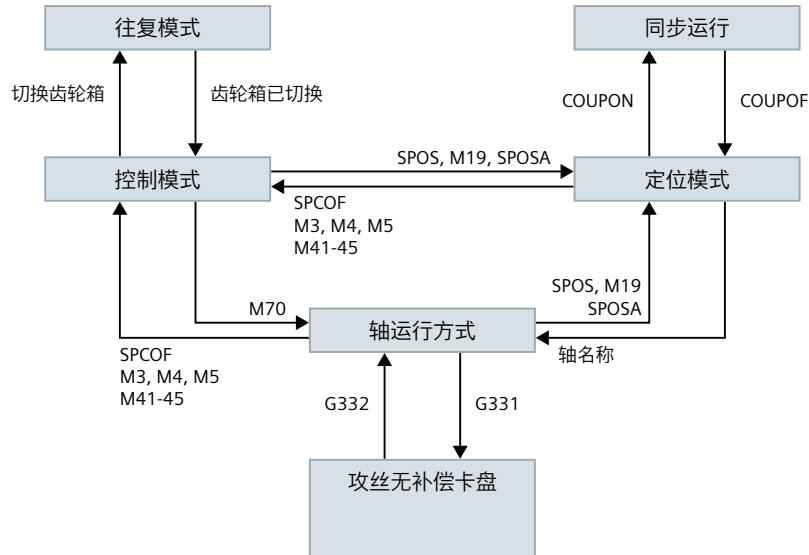
说明

轴/主轴模式切换

在轴模式和主轴模式共用一个电机的情况下，主轴可在轴模式与主轴模式之间切换。

13.2.2 运行方式切换

在主轴运行方式和轴运行方式之间可以如下切换：



- 控制模式 → 往复模式
通过自动齿轮档切换（M40）和新的 S 值，或通过 M41 至 M45 设定了新的齿轮档时，主轴切换至往复模式。只有当新的齿轮档不等于当前实际齿轮档时，主轴才会切换至往复模式。
- 往复模式 → 控制模式
如果创建了一个新的齿轮档，则接口信号：
`<Axis>.spindle.in.oscillModeActive`（往复模式）
会被复位，并使用接口信号：
`<Axis>.spindle.out.gearChangeoverDone`（齿轮档已切换）
切换至控制模式。
最后编程的主轴转速（S 值）再次生效。
- 控制模式 → 定位模式

如果要使主轴由旋转状态（M3 或 M4）定位停止，或者从静止状态（M5）重新定向，通过 SPOS、M19 或 SPOSA 切换至定位模式。
- 定位模式 → 控制模式
若需退出主轴定向，则用 M3、M4 或 M5 转换到控制模式。最后编程的主轴转速（S 值）再次生效。
- 定位模式 → 往复模式
若需退出主轴定向，可通过 M41 至 M45 切换至往复模式。齿轮档切换结束后，最后编程的主轴转速（S 值）和 M5（控制模式）再次生效。

13.2 运行方式

- 定位模式 → 轴模式
如果主轴已通过定向停止，则通过编写指定的轴名称切换到轴模式。齿轮档仍然保留。
- 控制模式 → 轴模式
如果要从控制模式切换到轴模式，则也可以通过编程 M70 进行。此时旋转的主轴如 M5 一样制动，位置闭环控制激活，且选择参数组零。
- 轴模式 → 控制模式
若需结束轴模式，则可以通过 M3、M4 或 M5 切换到运行模式。最后编程的主轴转速（S 值）再次生效。
- 轴模式 → 往复模式
若需结束轴模式，则可以通过 M41 至 M45 切换到往复模式（仅在编程的齿轮档不等于实际齿轮档时）。齿轮档切换结束后，最后编程的主轴转速（S 值）和 M5（控制模式）再次生效。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.spindle.out.gearChangeoverDone	LBP_Axis*.A_GearChangeOv	DB31, ...DBX16.3

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.spindle.in.oscillModeActive	LBP_Axis*.E_OscillMode	DB31, ...DBX84.6

13.2.3 控制模式

何时为控制模式？

执行下列功能时，主轴处于控制模式下：

- 恒定主轴转速：
 - S...S... M3/M4/M5 和 G93, G94, G95, G97, G971
 - S...S... M3/M4/M5 和 G33, G34, G35
 - S...S... M3/M4/M5 和 G63
- 恒定切削速度：
 - G96/G961 S...M3/M4/M5

主轴必须未同步。

前提条件

将 M3/M4/M5 和下列功能配合使用时，必须使用主轴位置实际值编码器：

- 旋转进给率（G95）
- 恒定切削速度（G96、G961、G97、G971）
- 螺纹切削（G33、G34、G35）
- 刚性攻丝（G331、G332）
- 激活位置闭环控制（SPCON、M70）

将 M3/M4/M5 和下列功能配合使用时，不需要使用主轴位置实际值编码器：

- 反比时间进给率编码（G93）
- 进给速度，单位 mm/min 或 inch/min（G94）
- 带补偿夹具的攻丝（G63）

转速控制模式

需要恒定的主轴转速，而主轴位置不重要时，尤其适合使用转速控制运行（例如恒定的铣刀转速用于生成均匀的工件表面）

- 转速控制运行在零件程序中通过 M3、M4、M5 或 SPCOF 激活。
- 以下 NC/PLC 接口信号置位：
`<Axis>.spindle.in.ctrlModeActive`（控制运行）
- 不启用位置闭环控制作业时，NC/PLC 接口信号：
`<Axis>.basic.in.posCtrlActive`（位置控制器生效）
复位。
- 转速控制运行中的加速度根据齿轮档在以下机床数据中定义：
`MD35200 $MA_GEAR_STEP_SPEEDCTRL_ACCEL`
数值应尽可能对应物理条件。

位置闭环控制运行

主轴位置需要在较长时间保持为已知，或需要激活同步主轴设定值耦合时，尤其适合使用位置闭环控制运行。

- 位置闭环控制运行在零件程序中通过以下指令激活：SPCON(<主轴编号>)
- 以下 NC/PLC 接口信号置位：
 <Axis>.basic.in.posCtrlActive（位置控制器生效）
- 位置闭环控制运行中的加速度根据齿轮档在以下机床数据中定义：
 MD35210 \$MA_GEAR_STEP_POSCTRL_ACCEL

独立主轴复位

复位或程序结束（M2、M30）后的株洲特性通过以下机床数据设置：

MD35040 \$MA_SPIND_ACTIVE_AFTER_RESET（独立主轴复位）

值	含义
0	复位或程序结束时，主轴立即启用有效加速度制动至静止状态。最后编写的主轴转速和主轴旋转方向会被删除。
1	复位或程序结束时，保留最后编写的主轴转速（S 值）和主轴旋转方向（M3、M4、M5）。主轴不会被制动。

如果在复位或程序结束前恒定切削速度 G96、G961 生效，则当前的主轴转速（基于 100% 主轴倍率）会在系统内部作为最后编写的主轴转速接收。

只能通过 NC/PLC 接口信号使主轴静止：

<Axis>.basic.out.resetMovement（删除剩余行程/主轴复位）

出现为主轴生成快速停止的报警时，旋转方向会被删除。最后编程的主轴转速（S 值）保留。消除报警原因后必须重新启动主轴。

G96/G961 下的实际转速显示和主轴特性

<Axis>.basic.in.isStationary（进给轴/主轴停止）

主轴被作为“停止”时的转速通过以下机床数据设置：

MD36060 \$MA_STANDSTILL_VELO_TOL

指定该值时，须确保静止状态下以下 NC/PLC 接口信号保持：

<Axis>.basic.in.isStationary（进给轴/主轴停止）

报告 <Axis>.basic.in.isStationary（进给轴/主轴停止），且没有为主轴启用位置闭环控制时，操作界面上实际转速会显示零值，且通过系统变量 \$AA_S[<n>] 将读取零值。

恒定切削速度 G96/G961 下的主轴特性

- 加工开始时（从 G0 过渡至 Gx），以及 NC 停止、G60（模态准停）和 G9（逐段准停）后，系统会等待轨迹启动，直至转速实际值到达转速设定值的公差范围：
<Axis>.spindle.in.speedSetpointReached（n 实际 = n 设定）
- 即使设定了大幅度的转速变化（平面轴向位置 0 运行），NC/PLC 接口信号：
<Axis>.spindle.in.speedSetpointReached（n 实际 = n 设定）
和
<Axis>.spindle.in.speedSetpointAboveMax（设定转速已限制）
也会设置为定义的值。
- 在低于最低转速
或识别出 NC/PLC 接口信号：
<Axis>.basic.in.isStationary（进给轴/主轴停止）
时，NC/PLC 接口信号：
<Axis>.spindle.in.speedSetpointReached（n 实际 = n 设定）
会被复位（例如用于机床紧急策略）。
- 已开始的轨迹加工（G64，精磨）不会被中断。

此外主轴特性还受以下机床数据影响：

MD35500 \$MA_SPIND_ON_SPEED_AT_IPO_START（主轴在设定区域时的进给使能）

齿轮档切换结束时的主轴特性

- NC/PLC 接口信号：
 <Axis>.spindle.out.gearChangeoverDone（齿轮档已切换）
 会提示 NC，新的齿轮档
 （NC/PLC 接口信号<Axis>.spindle.out.gearStageSelection（实际齿轮档 A 至 C））
 生效，且往复模式结束。
 此时以下 NC/PLC 接口信号设置与否则不重要：
 <Axis>.spindle.out.oscillEnable（往复转速）
 。
 实际齿轮档应与设定齿轮档相一致。
 选择参数组时，信号发送的实际齿轮档为相关信息。
- 由 PLC (<Axis>.spindle.out.gearChangeoverDone) 应答齿轮档切换 (GSW) 后，
 主轴处于转速控制运行 (<Axis>.spindle.in.ctrlModeActive = 1) 中。
 如果在 GSW 前编写了旋转方向（M3、M4、M5 或 SINU_RotateSpindle (Basic
 Program - FC18)：“启动主轴旋转”）或主轴转速（S 值），那么最后编写的的旋转方
 向和转速将在 GSW 后重新生效。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.resetMovement	LBP_Axis*.A_DelDTGSpReset	DB31, ...DBX2.2
<Axis>.spindle.out.gearStageSelection	LBP_Axis*.A_ActGearA..C	DB31, ...DBX16.0.. 2
<Axis>.spindle.out.gearChangeoverDone	LBP_Axis*.A_GearChangeOv	DB31, ...DBX16.3
<Axis>.spindle.out.oscillEnable	LBP_Axis*.A_OscilSpeed	DB31, ...DBX18.5

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.isStationary	LBP_Axis*.E_Stat	DB31, ...DBX61.4
<Axis>.basic.in.posCtrlActive	LBP_Axis*.E_PositContr	DB31, ...DBX61.5
<Axis>.spindle.in.speedSetpointAboveMax	LBP_Axis*.E_SetSpeedLimit	DB31, ...DBX83.1
<Axis>.spindle.in.speedSetpointReached	LBP_Axis*.E_SetRange	DB31, ...DBX83.5

Basic Program Plus	Basic Program	
<Axis>.spindle.in.oscillModeActive	LBP_Axis*.E_OscillMode	DB31, ...DBX84.6
<Axis>.spindle.in.ctrlModeActive	LBP_Axis*.E_ContrMode	DB31, ...DBX84.7

13.2.4 往复模式

对于主轴，往复模式在齿轮档切换期间激活。

对其原理的详细说明请见“通过往复模式进行齿轮档切换 (页 462)”。

13.2.5 定位模式

13.2.5.1 一般功能

何时为定位模式？

在定位模式下，主轴将在定义的位置处停止。此时位置闭环控制会被激活，并保持生效直至被取消。

执行下列功能时，主轴处于定位模式下：

- SPOS[<n>]=...
- SPOS[<n>]=ACP(...)
- SPOS[<n>]=ACN(...)
- SPOS[<n>]=AC(...)
- SPOS[<n>]=IC(...)
- SPOS[<n>]=DC(...)
- SPOSA[<n>]=ACP(...)
- SPOSA[<n>]=ACN(...)
- SPOSA[<n>]=AC(...)
- SPOSA[<n>]=IC(...)
- SPOSA[<n>]=DC(...) 与 SPOSA[<n>]=... 相同
- M19 或者 M[<n>]=19

其中 <n> = 主轴编号，对于主主轴可省去地址扩展 [<n>]。

SPOS[<n>]=AC(...)

将主轴定位至一个绝对位置（0 至 359.999 度）定位方向由当前的主轴旋转方向（主轴旋转）决定，或由剩余行程决定。

SPOS[<n>]=IC(...)

以编写的最后一个位置为基准将主轴定位至一个相对位置（+/- 999999.99 度）。定位方向通过待运行行程的符号确定。

SPOS[<n>]=DC(...)

将主轴沿最短行程定位至一个绝对位置（0 至 359.999 度）
定位方向由当前的主轴旋转方向决定（主轴旋转），或由控制系统自动确定（主轴停止）。

SPOS[<n>]=...

功能流程与 SPOS[<n>]=DC(...) 相同。

SPOS[<n>]=ACP(...)

从正方向逼近位置。

在从负方向定位时，主轴会制动直至转速为零，并沿相反方向加速，从而执行正逼近方向。

SPOS[<n>]=ACN(...)

从负方向逼近位置。

在从正方向定位时，主轴会制动直至转速为零，并沿相反方向加速，从而执行逼近负方向。

M19 (DIN 66025)

可通过 M19 定位主轴。此时位置和位置逼近模式通过下列设定数据读取：

SD43240 \$SA_M19_SPOS[<n>]（M19 主轴定位中的主轴位置）

SD43250 \$SA_M19_SPOSMODE[<n>]（M19 主轴定位中的主轴位置逼近模式）

M19 的定位作用与以下指令相同：

SPOS = <逼近模式> <位置/行程>

M19 会作为辅助功能输出至 NC/PLC 接口，和 M3、M4、M5 和 M70 一样。定位期间，M19 程序段在插补器中保持生效（同 SPOS）。

将 M19 作为宏（例如 `DEFINE M19 AS SPOS=0`）或子程序使用的零件程序将保持可执行状态。出于与之前版本控制系统的兼容性原因，可按照下面的示例取消 M19（NC 定位主轴）的内部处理：

```
MD22000 $MC_AUXFU_ASSIGN_GROUP[0] = 4           ; 辅助功能组：4
MD22010 $MC_AUXFU_ASSIGN_TYPE[0] = "M"           ; 辅助功能类型："M"
MD22020 $MC_AUXFU_ASSIGN_EXTENSION[0] = 0         ; 辅助功能扩展：0
MD22030 $MC_AUXFU_ASSIGN_VALUE[0] = 19           ; 辅助功能数值：19
```

隐性生成辅助功能 M19

为了使 M19 和 SPOS/SPOSA 在 NC/PLC 接口上的特性方面保持一致，可在使用 SPOS 和 SPOSA 时将辅助功能 M19 输出至 NC/PLC 接口。

若需激活该功能，可采用以下两种方案：

- 通过以下机床数据针对通道为通道中所有主轴激活：
MD20850 \$MC_SPOS_TO_VDI（启用 SPOS/SPOSA 时将 M19 输出至 PLC）

位	值	含义
0	0	若 MD35035 \$MA_SPIND_FUNCTION_MASK 位 19 也被设置为“0”，那么在启用 SPOS 和 SPOSA 时不会生成辅助功能 M19。这样一来也可省去对辅助功能的应答时间。
	1	在零件程序中编写 SPOS 和 SPOSA 时会在零件程序中生成辅助功能 M19，并将其输出至 PLC。地址扩展对应主轴编号。

- 通过以下机床数据针对主轴和跨通道激活：
MD35035 \$MA_SPIND_FUNCTION_MASK（主轴功能）

位	值	含义
19	0	若 MD20850 \$MC_SPOS_TO_VDI 位 0 也被设置为“0”，那么在启用 SPOS 和 SPOSA 时不会生成辅助功能 M19。这样一来也可省去对辅助功能的应答时间。
	1	在零件程序中编写 SPOS 和 SPOSA 时会隐性生成辅助功能 M19，并将其输出至 PLC。地址扩展对应主轴编号。

说明

在将一根主轴应用于多个通道的情形下（跨通道取轴/主轴），建议优先通过 MD35035 激活。

这两个机床数据中有一个的配置 = 1 时，将隐形生成辅助功能 M19。

此功能激活时，SPOS/SPOSA 程序段的最小执行时间会由 PLC 提升至辅助功能辅助功能的输出和应答时间。

隐形生成的辅助功能 M19 具有“Quick（快速）”和“运行中输出”属性。这些属性为固定设置，且与辅助功能专用机床数据（MD..._SM..._AUXFU_...）中 M19 的配置无关。

通过 FC 18 发出主轴定位指令时将不会生成辅助功能 M19。

定位结束

定位结束可如下编写：

FINEA[S<n>]:	在到达“精准停”时运行结束 (<Axis>.basic.in.posFineReached)
COARSEA[S<n>]:	在到达“粗准停”时运行结束 (<Axis>.basic.in.posCoarseReached)
IPOENDA[S<n>]:	在到达“插补停止”时运行结束

此外可通过 IPOBRKA 为单轴插补在制动斜坡(100-0%)中设置程序段转换的运行结束标准。

更多信息

定位轴 (页 277)

程序段转换

如果已经达到所有在程序段中编程的主轴或轴的运行结束标准，并且也达到了轨迹插补的程序段转换标准，那么将执行程序段转换。这既适用于零件程序程序段，也适用于工艺循环程序段。

SPOS、M19 和 SPOSA 功能相同，区别在于程序段转换特性：

- SPOS 和 M19
程序段中编写的所有功能达到其程序段结束标准（例如，所有辅助功能由 PLC 应答，所有轴到达终点），且主轴已到达定位终点时，执行程序段转换。
- SPOSA
程序段中编写的所有功能（主轴除外）达到其程序段结束标准后，执行程序段转换。
若 SPOSA 单独编写在程序段中，则会立即执行程序段转换。主轴定位可能会跨越过个程序段（参见 WAITS）。

协调

运动过程中的协调可通过以下方式实现：

- PLC
- 机床数据配置
- 零件程序中的编程

PLC

当 NC/PLC 接口信号：

`<Axis>.spindle.in.speedSetpointReached`（主轴处于设定区域）

不存在时，可设置通道专用 NC/PLC 接口信号：

`<Chan>.basic.out.disableReadIn`（读取禁止）

用于等待特定的主轴位置。

机床数据配置

通过以下机床数据设置：

`MD35500 $MA_SPIND_ON_SPEED_AT_IPO_START = 1`

并启用公差：

`MD35150 $MA_SPIND_DES_VELO_TOL`

实现：主轴加速到预设的转速后再进行轨迹插补。

采用以下设置时：

`MD35500 $MA_SPIND_ON_SPEED_AT_IPO_START = 2`

运行的轨迹轴会在加工开始前在最后一个 G0 处停止。

满足以下条件时继续执行加工：

- 收到下一个运行指令。
- 到达设定的主轴转速。
- 当 `MD35510 $MA_SPIND_STOPPED_AT_IPO_START = 1`
（主轴停止时，使能轨迹进给）

零件程序中的编程

零件程序中的协调措施有以下优点：

- 零件程序的创建者可决定在哪个程序位置完成主轴加速，例如用于开始加工工件。
- 避免不必要的等待时间。

零件程序中的协调通过编写 WAITS 指令实现：

WAITS:	用于主主轴
WAITS [<n>]:	用于编号为 <n> 的主轴
WAITS [<n>, <m>, ...]:	用于多根主轴，可支持最大主轴数的主轴

 **小心**

协调故障

零件程序的创建者必须为 WAITS 输入下列等待条件中的一个：

- 到达位置
- 主轴停止
- 主轴达到编写的转速

在跨通道主轴应用中，零件程序的创建者须确保：当其他通道的主轴已开始加速或制动，以实现设定的新转速或新旋转方向时，WAITS 必须在该阶段中最先启动。

执行后续程序段时将等待，直至：

- 到达使用 SPOSA 编写的位置
- 到达 M5 主轴停止状态：
 <Axis>.basic.in.isStationary（主轴停止）
 此时将启用公差：
 MD36060 \$MA_STANDSTILL_VELO_TOL
 识别出第一个信号的出现时，WAITS 将结束并切换至下一个程序段。
- 在 M3/M4（速度控制运行）中转速处于设定区域内：
 <Axis>.spindle.in.speedSetpointReached（主轴位于设定范围内）
 此时将启用公差：
 MD35150 \$MA_SPIND_DES_VELO_TOL
 识别出第一个信号的出现时，WAITS 将结束并切换至下一个程序段。
 此 WAITS 功能在编写的通道中生效。
 通过 WAITS 可等待该通道已知的所有主轴，包括已在其他通道中启动的主轴。

特殊情况

- **主轴速度的公差：**
采用以下机床数据设置时：
MD35150 \$MA_SPIND_DES_VELO_TOL = 0
NC/PLC 接口信号：
<Axis>.spindle.in.speedSetpointReached（主轴处于设定区域）
总是设置为 1。
一旦主轴在转速或旋转方向变化（M3/M4）中达到设定值，WAITS 立即结束。
- **缺少使能信号：**
若 WAITS 在转速控制运行中等待“主轴处于设定区域”，但另一方面主轴由于缺少使能信号（轴进给使能、控制器使能、脉冲使能等）停止或不旋转，那么要等到使能信号恢复，并存在“主轴处于设定区域”信号时，该程序段才会结束。
- **NC 停止和 BAG 停止时的特性：**
若在 WAITS 期间触发了一个 NC 停止或 BAG 停止程序段，那么 NC 启动后在上述边界条件下将继续等待。

说明

对于跨通道主轴应用，在编写程序时尤其需要注意：当其他通道中的主轴尚以“旧”转速旋转时，不可在通道中启动 WAITS（过早）。
在此情形下存在“主轴设定区域”信号，因此 WAITS 会过早结束。
为了避免此情形，建议一定要在 WAITS 前设置一个 WAITM。

进给率

定位速度在以下机床数据中配置：

MD35300 \$MA_SPIND_POSCTRL_VELO（位置闭环控制激活转速）

配置的定位速度可通过编程或同步动作进行修改。

FA[S<n>]=<值>

其中： <n>： 主轴编号

<值>： 定位速度，单位 度/分钟

FA[S<n>]=0 时配置的速度生效。

加速度

加速度在以下机床数据中配置：

MD35210 \$MA_GEAR_STEP_POSCTRL_ACCEL（位置闭环控制运行下的加速度）

13.2 运行方式

MD35200 \$MA_GEAR_STEP_SPEEDCTRL_ACCEL（转速控制运行下的加速度）

配置的定位加速度可通过编程或同步动作进行修改。

ACC[S<n>]=<值>

其中： <n>： 主轴编号
 <值>： 加速度，以基于配置的加速度的百分比设定

ACC[S<n>]=0 时配置的加速度生效。

定位终止

定位通过以下方式终止：

- 通过 NC/PLC 接口信号：
 <Axis>.basic.out.resetMovement（删除剩余行程/主轴复位）。
- 每次复位时（例如操作面板前端复位）
- 通过 NC 停止。

终止特性与以下机床数据无关：

MD35040 \$MA_SPIND_ACTIVE_AFTER_RESET（独立主轴复位）

特点

主轴倍率开关有效。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.resetMovement	LBP_Axis*.A_DeIDTGSpReset	DB31, ...DBX2.2
<Chan>.basic.out.disableReadIn	LBP_Chan*.A_Rldisable	DB21, ...DBX 6.1

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.posCoarseReached	LBP_Axis*.E_ExactCoarse	DB31, ...DBX60.6
<Axis>.basic.in.posFineReached	LBP_Axis*.E_ExactFine	DB31, ...DBX60.7

Basic Program Plus	Basic Program	
<Axis>.basic.in.isStationary	LBP_Axis*.E_Stat	DB31, ...DBX61.4
<Axis>.spindle.in.speedSetpointReached	LBP_Axis*.E_SetRange	DB31, ...DBX83.5

13.2.5.2 从旋转状态定位主轴

初始状态

启动定位（程序中的 SPOS、M19 或 SPOSA 指令）时，主轴可能处于转速控制运行或位置闭环控制运行下。

此时须区分下列情形：

情形 1：	主轴处于转速控制运行下，超出编码器极限频率
情形 2：	主轴处于转速控制运行下，未超出编码器极限频率
情形 3：	主轴处于位置闭环控制运行下
情形 4：	主轴转速 < 位置闭环控制激活转速

过程

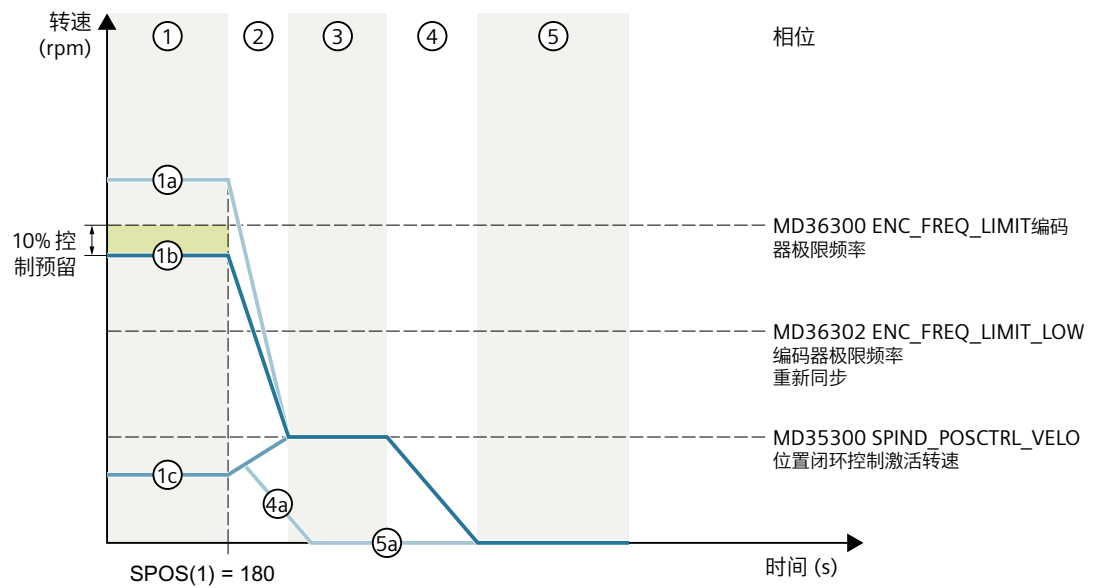


图 13-1 从旋转状态定位主轴

说明

为编码器重新同步配置的编码器极限频率（MD36302 \$MA_ENC_FREQ_LOW）所对应的转速必须大于位置闭环控制激活转速（MD35300 \$MA_SPIND_POSCTRL_VELO）。

阶段 ①

从阶段 1a 定位：

主轴以大于编码器极限频率的转速旋转。主轴未同步。

从阶段 1b 定位：

主轴以小于编码器极限频率的转速旋转。主轴已同步。

说明

若激活了位置闭环控制，则转速只可为主轴最大转速或编码器极限频率的 90%（需要 10% 的控制余量）。

从阶段 1c 定位：

主轴以编写的主轴转速旋转，转速小于配置的位置闭环控制激活转速：

MD35300 \$MA_SPIND_POSCTRL_VELO

主轴已同步。

阶段 ②

主轴转速 > 位置闭环控制激活转速

随着 SPOS、M19 或 SPOSA 指令生效，主轴通过配置的加速度开始制动，直至达到位置闭环控制激活转速。

MD35200 \$MA_GEAR_STEP_SPEEDCTL_ACCEL

低于编码器极限频率时主轴会进行同步。

主轴转速 < 位置闭环控制激活转速

SPOS、M19 或 SPOSA 指令使主轴切换至位置闭环控制运行（当其尚不处于位置闭环控制下时）。

配置的位置闭环控制运行中的加速度生效：

MD35210 \$MA_GEAR_STEP_POSCTRL_ACCEL

系统会计算与目标位置之间的行程。

主轴以时间最优的方式运行到编写的目标位置。也就是说采用最高速度逼近目标点（最大速度 MD35300 \$MA_SPIND_POSCTRL_VELO）。根据相应的边界条件，按 2 - 3 - 4 - 5，或者 2 - 4a - 5a 的阶段顺序运行。

阶段 ③

主轴转速 > 位置闭环控制激活转速

达到配置的位置闭环控制接通转速（MD35300 \$MA_SPIND_POSCTRL_VELO）时：

- 位置闭环控制激活（未处于位置闭环控制下时）
- 计算（与目标位置的）剩余行程
- 在位置闭环控制运行下切换至配置的加速度（或保持该加速度）：
MD35210 \$MA_GEAR_STEP_POSCTRL_ACCEL

主轴转速 < 位置闭环控制激活转速

为了到达目标点，主轴会先加速至配置的位置闭环控制激活转速（MD35300 \$MA_SPIND_POSCTRL_VELO）。此时不会超出该转速。

“制动启用点计算”能够确定，通过配置的位置闭环控制中的加速度（MD35210 \$MA_GEAR_STEP_POSCTRL_ACCEL）何时能够精确到达编写的主轴位置。

阶段 ④

主轴转速 > 位置闭环控制激活转速

主轴从计算得出的“制动点”开始采用机床数据：

MD35210 \$MA_GEAR_STEP_POSCTRL_ACCEL

中设定的加速度制动，直至到达目标位置。

主轴转速 < 位置闭环控制激活转速

在阶段 3 中“制动启用点计算”所确定的时间点，主轴以配置的位置闭环控制中的加速度（MD35210 \$MA_GEAR_STEP_POSCTRL_ACCEL）制动，直至到达停止状态。

阶段 4a:

在 SPOS 指令生效时，目标点过近以至于主轴无法加速到配置的位置闭环控制激活转速（MD35300 \$MA_SPIND_POSCTRL_VELO）。

此时主轴会以配置的位置闭环控制中的加速度（MD35210 \$MA_GEAR_STEP_POSCTRL_ACCEL）制动，直至到达停止状态。

阶段⑤

主轴转速 > 位置闭环控制激活转速

位置闭环控制保持生效，主轴停止在编写的位置。

说明

控制系统会对主轴位置实际值编码器的最大编码器极限频率进行监控（可能超出）；在位置闭环控制运行下，转速设定值会被降低到测量系统极限转速的 90%。

此时以下 NC/PLC 接口信号置位：

<Axis>.spindle.in.speedSetpointAboveMax（编程的转速过高）

若在降低转速设定值后仍检测到“超出测量系统极限频率”，则会触发报警。

主轴转速 < 位置闭环控制激活转速（阶段 5、5a）

主轴到达位置并停止。位置闭环控制生效，主轴停止在编写的位置上。

若主轴实际位置和编写的位置（主轴设定位置）之间的距离小于配置的精/粗准停限值，那么系统会置位以下 NC/PLC 接口信号：

<Axis>.basic.in.posCoarseReached（采用粗准停到达位置）

<Axis>.basic.in.posFineReached（采用精准停到达位置）

准停限值通过以下机床数据定义：

MD36010 \$MA_STOP_LIMIT_FINE（精准停）

MD36000 \$MA_STOP_LIMIT_COARSE（粗准停）

说明

达到定位结束标准并以信号提示后，定位结束。

前提条件是“精准停”。通过零件程序中的 SPOS、M19 或 SPOSA 指令、同步动作、以及通过 PLC 使用 FC 18 进行的主轴定位可实现精准停。

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.posCoarseReached	LBP_Axis*.E_ExactCoarse	DB31, ...DBX60.6
<Axis>.basic.in.posFineReached	LBP_Axis*.E_ExactFine	DB31, ...DBX60.7
<Axis>.spindle.in.speedSetpointAboveMax	LBP_Axis*.E_SetSpeedLimit	DB31, ...DBX83.1

13.2.5.3 从静止状态定位主轴

过程

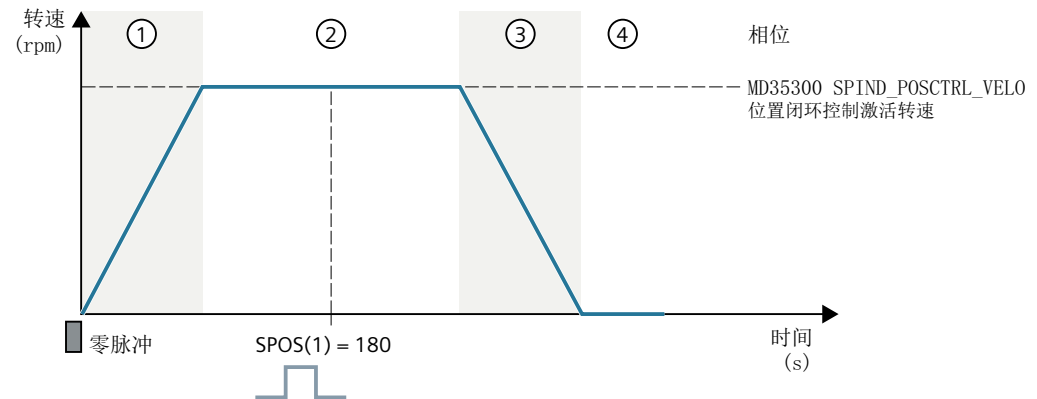
需要对静止状态下的主轴执行定位时，须区分两种情形：

- 情形 1：主轴未同步
在控制系统和驱动接通后，或者齿轮档切换后需要定位主轴时（例如用于换刀）便是这种情形。

MD31040 \$MA_ENC_IS_DIRECT = 0

- 情形 2：主轴已同步
控制系统和驱动接通后，在执行首次定位前通过 M3 或 M4 进行了至少一次主轴旋转，之后通过 M5 停止主轴（与零脉冲同步）时，便是这种情形。

情形 1：主轴未同步



情形 2：主轴已同步

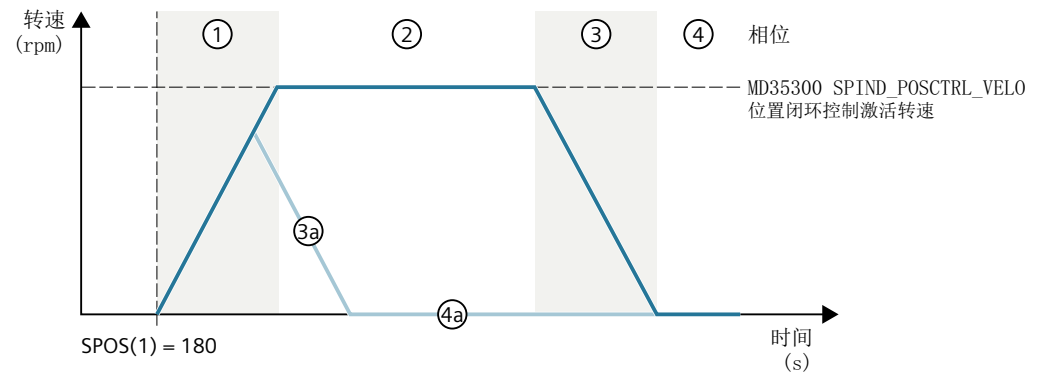


图 13-2 对停止主轴的定位

阶段 ①

情形 1：主轴未同步

主轴通过编写的 SPOS、M19 或 SPOSA 指令采用以下机床数据中设定的加速度进行加速：

MD35200 \$MA_GEAR_STEP_SPEEDCTRL_ACCEL（转速控制运行下的加速度）

旋转方向通过以下机床数据定义：

MD35350 \$MA_SPIND_POSITIONING_DIR（从静止状态定位时的旋转方向）

特例：

若通过 ACN、ACP、IC 定位，那么编写的运行方向会被激活。

主轴与主轴位置实际值编码器的下一个零脉冲进行同步，并进入位置闭环控制运行。

此时系统会监控，在设定的行程内能否找到零脉冲（IC 除外）：

MD34060 \$MA_REFP_MAX_MARKER_DIST（至参考脉冲的最大行程）

在主轴未同步的情况下达到以下机床数据中输入的转速时：

MD35300 \$MA_SPIND_POSCTRL_VELO（定位转速）

主轴将以位置闭环控制激活转速继续旋转（不再进行加速）。

情形 2：主轴已同步

主轴通过 SPOS、M19 或 SPOSA 指令切换至位置闭环控制运行。

此时启用以下机床数据中设定的加速度：

MD35210 \$MA_GEAR_STEP_POSCTRL_ACCEL（位置闭环控制运行下的加速度）

旋转方向通过编写的运动（ACP、ACN、IC、DC）或剩余行程确定。

此时不会超出以下机床数据中输入的转速：

MD35300 \$MA_SPIND_POSCTRL_VELO（位置闭环控制激活转速）

系统会计算与目标位置之间的行程。

主轴以时间最优的方式运行到编写的目标位置。也就是说采用最高速度逼近目标点（最大速度 MD35300 \$MA_SPIND_POSCTRL_VELO）。根据相应的边界条件，按 1 - 2 - 3 - 4，或者 1 - 3a - 4a 的阶段顺序运行。

阶段 ②

情形 1：主轴未同步

若主轴同步，那么位置闭环控制会被激活。

主轴以以下机床数据中设定的转速一直运行：

MD35300 \$MA_SPIND_POSCTRL_VELO

直到“制动启用点计算”识别出通过定义的加速度何时能够精确到达编写的主轴位置。

情形 2：主轴已同步

为了到达目标点，主轴会先加速至以下机床数据中输入的转速：

MD35300 \$MA_SPIND_POSCTRL_VELO

此时不会超出该转速。

“制动启用点计算”能够确定，通过以下机床数据中定义的加速度何时能够精确到达编写的主轴位置：

MD35210 \$MA_GEAR_STEP_POSCTRL_ACCEL

在阶段 1 中“制动启用点计算”所确定的时间点，主轴启用以下机床数据中定义的加速度制动，直至到达停止状态。

MD35210 \$MA_GEAR_STEP_POSCTRL_ACCEL

阶段 ③

在阶段 2 中“制动启用点计算”所确定的时间点，主轴启用以下机床数据中定义的加速度制动，直至到达停止状态。

MD35210 \$MA_GEAR_STEP_POSCTRL_ACCEL

阶段 3a：

在 SPOS 指令生效时，目标点过近以至于主轴无法加速到以下机床数据中设定的速度：

MD35300 \$MA_SPIND_POSCTRL_VELO

。

主轴通过以下机床数据中定义的加速度制动，直至到达停止状态：

MD35210 \$MA_GEAR_STEP_POSCTRL_ACCEL

阶段 ④，④a

主轴到达位置并停止。位置闭环控制生效，主轴停止在编写的位置上。

若主轴实际位置和编程的位置（主轴设定位置）之间的距离小于精/粗准停限值，那么系统会置位以下 NC/PLC 接口信号：

<Axis>.basic.in.posCoarseReached（采用粗准停到达位置）

和

13.2 运行方式

<Axis>.basic.in.posFineReached（采用精准停到达位置）
。

准停限值在以下机床数据中定义：

MD36010 \$MA_STOP_LIMIT_FINE
MD36000 \$MA_STOP_LIMIT_COARSE

阶段 3：

在阶段 2 中“制动启用点计算”所确定的时间点，主轴启用以下机床数据中定义的加速度制动，直至到达停止状态。

MD35210 \$MA_GEAR_STEP_POSCTRL_ACCEL

阶段 4：

主轴到达位置并停止。位置闭环控制生效，主轴停止在编写的位置上。

若主轴实际位置和编程的位置（主轴设定位置）之间的距离小于精/粗准停限值，那么系统会置位以下 NC/PLC 接口信号：

<Axis>.basic.in.posCoarseReached（采用粗准停到达位置）
和
<Axis>.basic.in.posFineReached（采用精准停到达位置）
。

准停限值在以下机床数据中定义：

MD36010 \$MA_STOP_LIMIT_FINE
MD36000 \$MA_STOP_LIMIT_COARSE

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.posCoarseReached	LBP_Axis*.E_ExactCoarse	DB31, ...DBX60.6
<Axis>.basic.in.posFineReached	LBP_Axis*.E_ExactFine	DB31, ...DBX60.7

13.2.5.4 换刀信号“主轴到达位置”

功能

换刀时的运动过程（特别是对于铣床）主要包括主轴定位，以及随后（通过优化时也可实现时间同步）通过轨迹轴进行的换刀位置逼近运行。因此执行换刀必须满足的前提条件是：在逼近换刀位置前到达主轴位置。

若换刀循环由于人为操作（例如通过 NC 停止、轴和主轴的 NC 停止、运行方式组停止等）中断，则须确保主轴不会以错误的位置进入机械手。

为此，在主轴定位中通过“精准停”到达最后编写的主轴位置时，系统会输出以下 NC/PLC 接口信号用于位置检查：

`<Axis>.spindle.in.inPos`（主轴就位）

说明

此信号只会在“主轴定位”功能中输出。

其中包括：

- 零件程序中编写的 SPOS、SPOSA 和 M19
 - 同步动作中的 SPOS 和 M19
 - 使用 SINU_PositionSpindle (Basic Program - FC18) 进行的主轴定位
 - 通过 PLC 接口 (`<Axis>.spindle.out.startPosOperation`) 进行的主轴定位
-

信号置位

输出信号 `<Axis>.spindle.in.inPos`（主轴就位）的前提条件是：

- 主轴的回参考点状态为：

`<Axis>.basic.in.enc1Synchronized/`

`<Axis>.basic.in.enc2Synchronized`（已回参考点/已同步 1/2）= 1

说明

由于主轴定位中会自动搜索零脉冲，因此定位顺利完成时始终会在结束处提供回参考点信号。

- 实现“精准停”：

`<Axis>.basic.in.posFineReached`（精准停）= 1

此外设定值方面必须已到达编写的主轴位置。

删除信号

移除信号 <Axis>.basic.in.posFineReached（精准停）时，信号 <Axis>.spindle.in.inPos（主轴就位）也会随之复位。

其他属性

- 若定位后主轴已位于编程的位置上，那么 NC/PLC 接口信号 <Axis>.spindle.in.inPos（主轴就位）将保持置位。
- 若定位后（信号“主轴就位”已输出）例如在 JOG 运行方式下运行主轴，那么 NC/PLC 接口信号 <Axis>.spindle.in.inPos（主轴就位）会被删除。之后在该运行方式下将主轴运行回到原始位置时，NC/PLC 接口信号 <Axis>.spindle.in.inPos（主轴就位）又将重新置位。最后编写的位置设定保留。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.spindle.out.startPosOperation	LBP_Axis*.A_PosSpindle	DB31, ...DBX30.4

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.enc1Synchronized	LBP_Axis*.E_RefSyn1	DB31, ...DBX60.4
<Axis>.basic.in.enc2Synchronized	LBP_Axis*.E_RefSyn2	DB31, ...DBX60.5
<Axis>.basic.in.posFineReached	LBP_Axis*.E_ExactFine	DB31, ...DBX60.7
<Axis>.spindle.in.inPos	LBP_Axis*.E_SplnPosition	DB31, ...DBX85.5

13.2.6 轴运行方式

13.2.6.1 一般功能

功能

对于特定的加工任务，例如在车床上加工端面时，若转速控制下通过 M3、M4、M5 执行的主轴运行和通过 SPOS、M19 或 SPOSA 进行的主轴定位不能满足需求，则可将主轴切换至位置控制下的轴运行方式，并将其用作回转轴。

回转轴功能示例：

- 通过轴名称编程
- 零点偏移（G54，G55，TRANS，...）
- G90、G91、IC、AC、DC、ACP、ACN
- 动态转换（例如 TRANSMIT）
- 轨迹插补
- 作为定位轴运行

更多信息

回转轴 (页 315)

前提条件

- 对于主轴运行方式和进给轴运行方式，主轴电机是同一个。
- 位置测量系统对于主轴运行方式和进给轴运行方式可以是同一个，或者也可以使用独立的位置测量系统。
- 对于进给轴运行方式，务必需要一个位置实际值编码器。
- 为了采用轴运行方式，主轴必须进行回参考点，例如通过 G74 回参考点。
示例：

程序代码	注释
M70	; 主轴切换到轴运行方式
G74 C1=0 Z100	; 轴回参考点
G0 C180 X50	; 在位置控制下运行轴

可设计的 M 功能

对于可以将主轴接通到轴运行方式的 M 功能，可通过下列机床数据针对通道进行设计：

MD20094 \$MC_SPIND_RIGID_TAPPING_M_NR

说明

控制系统会根据编程顺序自动识别到轴运行方式的过渡（参见“隐性过渡至进给轴模式（页 425）”）。因此在零件程序中不必明确编程 M 功能将主轴切换至轴运行方式。也可编程 M 功能，以提高零件程序的可读性。

特点

- 进给倍率开关有效。
- NC/PLC 接口信号正常情况下不会结束轴运行：
`<Chan>.basic.out.reset`（复位）。
- 当：
`<Axis>.basic.in.isSpindle == 0`（主轴/回转轴）时，NC/PLC 接口信号，见信号表（`<Axis>.spindle.out.xxx`，`<Axis>.spindle.in.xxx`）
 无意义
- 可以在任何传动级激活进给轴方式。
 如果已在电机上安装位置实际值编码器（间接测量系统），可在不同传动级之间得出不同的定位和轮廓精度。
- 如果进给轴方式有效，则不能更换传动级。如需切换齿轮档，首先必须通过 M41 ... M45 或 M5、SPCOF 将主轴切换到受控方式中。
- 在进给轴方式中，第一个参数组生效（机床数据系下标为零）。

更多信息

速度、设定值/实际值系统、闭环控制（页 141）

动力

在轴运行方式中，轴的动态限值生效。例如：

- MD32000 \$MA_MAX_AX_VELO[<轴>]（最大轴速度）
- MD32300 \$MA_MAX_AX_ACCEL[<轴>]（最大轴加速度）
- MD32431 \$MA_MAX_AX_JERK[<轴>]（轨迹运动时的最大轴向急动）

前馈控制

保持对轴生效的前馈模式。

“动态前馈控制”功能的详细说明请见：

更多信息

功能手册之监控与补偿；补偿 > 动态前馈控制（跟随误差补偿）

示例：采用模拟控制器时的分辨率切换

切换至轴运行方式

编程	注释
SPOS=...	
M5	； 伺服使能关闭（由 PLC） → 输出至 PLC
M70	； 控制器切换（由 PLC，基于 M70） 伺服使能打开（由 PLC）
C=...	； NC 以进给轴参数组运行

切换至主轴运行方式

编程	注释
C=...	
M71	； → 输出至 PLC 伺服使能关闭（由 PLC） 控制器切换（由 PLC） 在 NC 内部切换到主轴参数组（1-5）上，伺服使能开（由 PLC）
M3/4/5 或 SPOS=...	； NC 以主轴参数组运行

在主轴运行方式下切换

根据生效的传动级选择参数组 1...5。

原则上除了带补偿衬套的攻丝时外都需要接通前馈控制：

MD32620 \$MA_FFW_MODE（前馈控制方式） ≠ 0

参数组	进给轴模式	主轴模式
1	生效	-
2	-	生效
3	-	生效
4	-	生效

13.2 运行方式

参数组	进给轴模式	主轴模式
5	-	生效
6	-	生效
主轴方式：不同齿轮档下的参数组		

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Chan>.basic.out.reset	LBP_Ch*.*.A_Reset	DB21, ... DBX7.7
<Axis>.spindle.out.ovrFactor	LBP_Axis*.*.A_SpOR	DB31, ...DBX019.0
<Axis>.spindle.out.gearStageSelection	LBP_Axis*.*.A_ActGearA..C	DB31, ...DBX016.0
<Axis>.spindle.out.gearChangeoverDone	LBP_Axis*.*.A_GearChangeOv	..2
<Axis>.spindle.out.resyncEnc1	LBP_Axis*.*.A_Sp1Syn	DB31, ...DBX016.3
<Axis>.spindle.out.resyncEnc2	LBP_Axis*.*.A_Sp2Syn	DB31, ...DBX016.4
<Axis>.spindle.out.resyncEnc1ForPos	LBP_Axis*.*.A_Sp1SynP	DB31, ...DBX016.5
<Axis>.spindle.out.resyncEnc2ForPos	LBP_Axis*.*.A_Sp2SynP	DB31, ...DBX017.4
<Axis>.spindle.out.disableSpeedMonitor	LBP_Axis*.*.A_NoMonitorGear	DB31, ...DBX017.5
<Axis>.spindle.out.deleteSValue	LBP_Axis*.*.A_DeleteS	DB31, ...DBX016.6
<Axis>.spindle.out.useFeedrateOvr	LBP_Axis*.*.A_FD_ORSp	DB31, ...DBX016.7
<Axis>.spindle.out.invertM3M4	LBP_Axis*.*.A_M3M4Inv	DB31, ...DBX017.0
<Axis>.spindle.out.stop	LBP_Axis*.*.A_StopSpindle	DB31, ...DBX017.6
<Axis>.spindle.out.startCw	LBP_Axis*.*.A_StartCW	DB31, ...DBX030.0
<Axis>.spindle.out.startCcw	LBP_Axis*.*.A_StartCCW	DB31, ...DBX030.1
<Axis>.spindle.out.autoGearStageChange	LBP_Axis*.*.A_AutoGearStep	DB31, ...DBX030.2
<Axis>.spindle.out.startPosOperation	LBP_Axis*.*.A_PosSpindle	DB31, ...DBX030.3
<Axis>.spindle.out.oscillEnable	LBP_Axis*.*.A_OscilSpeed	DB31, ...DBX030.4
<Axis>.spindle.out.oscillRequest	LBP_Axis*.*.A_OscilPLC	DB31, ...DBX018.5
<Axis>.spindle.out.oscillSetDirCcw	LBP_Axis*.*.A_SetRotDirectLeft	DB31, ...DBX018.4
<Axis>.spindle.out.oscillSetDirCw	LBP_Axis*.*.A_SetRotDirectRight	DB31, ...DBX018.7
		DB31, ...DBX018.6

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.spindleModeActive	LBP_Axis*.E_SpNA	DB31, ... DBX60.0
<Axis>.spindle.in.mFuncNumber	LBP_Axis*.E_MFunct	DB31, ... DBB086
<Axis>.spindle.in.sFuncValue	LBP_Axis*.E_SFunct	DB31, ...DBD088
<Axis>.spindle.in.requestedGearStageNumber	LBP_Axis*.E_SetpGearA..C	DB31, ...DBX082.0
<Axis>.spindle.in.gearChangeoverRequested	LBP_Axis*.E_GearChange	..2
<Axis>.spindle.in.speedLimitationExceeded	LBP_Axis*.E_SpeedLimit	DB31, ...DBX082.3
<Axis>.spindle.in.speedSetpointBelowMin	LBP_Axis*.E_SetSpeedIncreased	DB31, ...DBX083.0
<Axis>.spindle.in.speedSetpointAboveMax	LBP_Axis*.E_SetSpeedLimit	DB31, ...DBX083.2
<Axis>.spindle.in.speedSetpointReached	LBP_Axis*.E_SetRange	DB31, ...DBX083.1
<Axis>.spindle.in.cwRotActive	LBP_Axis*.E_ActRotRight	DB31, ...DBX083.5
<Axis>.spindle.in.constantCuttingSpeedActive	LBP_Axis*.E_ConstCuttSpeed	DB31, ...DBX083.7
<Axis>.spindle.in.rigidTappingActive	LBP_Axis*.E_Tapping	DB31, ...DBX084.0
<Axis>.spindle.in.syncModeActive	LBP_Axis*.E_SyncMode	DB31, ...DBX084.3
<Axis>.spindle.in.posModeActive	LBP_Axis*.E_PosMode	DB31, ...DBX084.4
<Axis>.spindle.in.oscillModeActive	LBP_Axis*.E_OscillMode	DB31, ...DBX084.5
<Axis>.spindle.in.ctrlModeActive	LBP_Axis*.E_ContrMode	DB31, ...DBX084.6
<Axis>.spindle.in.toolWithDynLimitationActive	LBP_Axis*.E_ToolDynLimit	DB31, ...DBX084.7
<Axis>.spindle.in.inPos	LBP_Axis*.E_SplnPosition	DB31, ...DBX085.0
		DB31, ...DBX085.5

13.2.6.2 隐性过渡至进给轴模式

功能

控制系统能够基于编程顺序自动识别出向进给轴模式的过渡，并于系统内部生成必要的 M70 过程。执行的步骤取决于具体情形，其中主要包括：

1. 停止主轴
2. 激活位置闭环控制，处理前馈控制和参数组切换
3. 程序段预处理的位置同步（必要时内部预处理停止）

该功能始终生效。因此基本不需要在零件程序中编写 M70 指令。

过程

隐性过渡至进给轴模式的步骤（零件程序中未编写 M70）：

- 从转速控制运行（M3、M4、M5、SPCOF、...）过渡至进给轴模式：
控制系统内部识别出过渡，并在请求进给轴模式的程序段前插入一个中间程序段。生成的程序段包含 M70 功能。该程序段的执行时间与程序中编写的 M70 程序段的执行时间基本相同。放弃隐性生成辅助功能 M70 以及将其输出至 PLC 时（参见 MD35035），停止主轴上的短暂切换可能会有区别（无制动时间）。
- 从定位模式（M19、SPOS、SPOSA）过渡至进给轴模式：
过渡将立即执行，不会生成中间程序段。在进行了相应配置时（参见 MD35035），隐性生成的辅助功能 M70 会在切换至主轴处于进给轴模式的程序段时输出至 PLC。

辅助功能向 PLC 的输出

向进给轴模式的隐性过渡可以辅助功能输出的形式发送至 PLC。

激活/取消激活

此功能通过以下机床数据激活/取消：

MD35035 \$MA_SPIND_FUNCTION_MASK（主轴功能）

位	值	含义
20	0	控制系统内部生成 M70 时，不向 PLC 输出辅助功能
	1	控制系统内部生成 M70 时，隐性生成辅助功能 M70 并输出至 PLC。地址扩展对应主轴编号。

说明

零件程序中编写的辅助功能 M70 始终会输出至 PLC。

属性

隐性生成的辅助功能 M70 具有“Quick（快速）”和“运行中输出”属性。这些属性为固定设置，且与辅助功能专用机床数据（MD..._M..._AUXFU_...）中 M70 的配置无关。

过渡至进给轴模式时只会生成一次 M70。在之后的将主轴作为轴运行的程序段中，系统将不再生成和输出 M70 辅助功能。直到通过 POS、M3、M4、M5、SPCOF 等指定退出进给轴模式，并于之后重新过渡至进给轴模式时，系统才会重新生成隐性 M70 并输出。

前提条件

同步动作

在同步动作中将主轴编写为一般轴时，必须确保满足过渡至进给轴模式的前提条件。

当主轴处于转速控制运行中时，在将其定义为轴前必须编写 M70 或 SPOS 指令。否则轴编程可能会触发报警。

FC 18

和同步动作一样，在使用 FC 18 时也必须确保过渡至进给轴模式的应用条件，例如通过过渡预备定位指令。否则 PLC 会以 FC 18 状态字中的故障位对 FC 18 调用进行应答。

使用 FC 18 编程过渡至进给轴模式时，**不会**隐性生成辅助功能 M70。

示例

示例 1:

零件程序：将旋转主轴过渡至进给轴模式

配置：MD35035 \$MA_SPIND_FUNCTION_MASK，位 20 = 1

程序代码	注释
N05 M3 S1000	
N10 ...	
N15 POS[C]=77	; 切换至 N15 前生成一个 M70 中间程序段，通过该程序段停止主轴，并将 M70 输出至 PLC。
...	

示例 2:

零件程序：从定位模式过渡至进给轴模式

配置：MD35035 \$MA_SPIND_FUNCTION_MASK，位 20 = 1

程序代码	注释
N05 SPOS=0	
N10 ...	
N15 C77	; 将隐性 M70 输出至 PLC，无中间程序段
...	

示例 3:

同步动作：从主轴定位模式过渡至进给轴模式

13.2 运行方式

配置：MD35035 \$MA_SPIND_FUNCTION_MASK，位 20 = 1

程序代码	注释
WHEN COND1==TRUE DO SPOS=180	
WHEN COND2==TRUE DO POS[C]=270	; 将隐性 M70 输出至 PLC

示例 4：

同步动作：通过 M70 从转速控制运行过渡至进给轴模式

配置：MD35035 \$MA_SPIND_FUNCTION_MASK，位 20 = 1

程序代码	注释
WHEN COND11==TRUE DO M3 S1000	
WHEN COND12==TRUE DO M70	; 将 M70 输出至 PLC
WHEN COND13==TRUE DO POS[C]=270	; 不生成隐性 M70，因为已启用进给轴模式

示例 5：

同步动作：不允许的从转速控制运行到进给轴模式的过渡

配置：MD35035 \$MA_SPIND_FUNCTION_MASK，位 20 = 1

程序代码	注释
WHEN COND21==TRUE DO M3 S1000	
WHEN COND22==TRUE DO POS[C]=270	; 报警 20141！

13.2.7 主轴初始设置

主轴初始设置

可以使用下列机床数据将主轴运行方式确定为初始设置：

MD35020 \$MA_SPIND_DEFAULT_MODE

值	主轴初始设置
0	转速控制运行，取消位置调节
1	转速控制运行，打开位置调节
2	定位运行
3	轴运行

主轴初始设置的生效时间

在下列机床数据中可以设置主轴初始设置的生效时间：

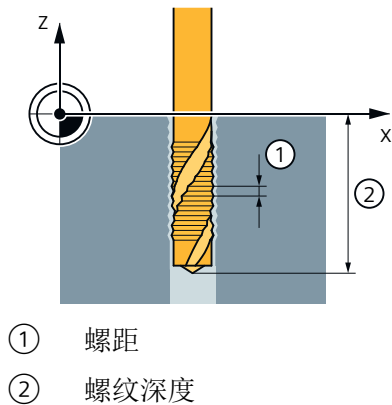
MD35030 \$MA_SPIND_DEFAULT_ACT_MASK

值	生效时间
0	上电
1	上电并启动程序
2	上电并复位（M2 / M30）

13.2.8 刚性攻丝

13.2.8.1 功能

在刚性攻丝中，系统在位置闭环控制下对线性轴和主轴的运行进行插补。前提条件是**主轴带行程测量系统**并受位置闭环控制。



螺纹旋转方向

螺纹旋转方向（右旋螺纹或者左旋螺纹）通过螺距的符号确定：

- 正螺距 → 右旋螺纹（旋转方向对应 M3）
- 负螺距 → 左旋螺纹（旋转方向对应 M4）

定义的主轴起始位置

在刚性攻丝中，在下列情形下必须编写定义的主轴起始位置（SPOS）：

- 在多重加工中制造的螺纹
- 需要既定的螺纹起始位置的螺纹

13.2 运行方式

```
SPOS = <起始位置>
G331 ...
```

13.2.8.2 编程

使用指令 G331 和 G332 编程不带补偿夹具的攻丝时，会执行以下运行：

- G331:在钻削方向上攻丝，直至螺纹终点
- G332:通过主轴旋转方向自动反向，回退至成套丝锥 G331

句法

```
G331 <轴> <螺距>
G331 <轴> <螺距> S...
G332 <轴> <螺距>
```

含义

G331:	攻丝 通过轴运行（钻深）和螺距描述攻丝。	
	生效方式:	模态
G332:	攻丝的回退 回退应以与攻丝时相同的螺距执行 (G331)。主轴自动换向。	
	生效方式:	模态
<轴>:	几何轴（X、Y 或 Z）至螺纹终点的运行行程/位置，例如 Z50	
<螺距>:	螺距 I (X)、J (Y) 或 K (Z): • 正螺距：右螺纹，例如 K1,25 • 负螺距：左螺纹，例如 K-1,25	
	值域:	±0.001 至 ±2000.00 mm/rev
S...:	主轴转速 如未给定主轴转速，则以上次生效的主轴转速运行。	

说明

第二齿轮级数据组

为了在攻丝时有效地调节主轴转速和电机转矩并提高加速度，可以在轴专用的机床数据中配置第二齿轮级数据组，以及另外两个可定义的开关阈值（最大转速和最小转速），该数据组不同于第一齿轮级数据组，且与其转速开关阈值无关。请务必注意机床制造商的规定。

更多的信息参见功能手册“轴和主轴”。

示例

- 示例：使用 G331 / G332 攻丝 (页 431)
- 示例：给出当前齿轮级已编程的钻削转速 (页 432)
- 示例：使用第二齿轮级数据组 (页 432)
- 示例：未进行转速编程，齿轮级监控 (页 432)
- 示例：无法进行齿轮换挡，齿轮级监控 (页 433)
- 示例：不带 SPOS 的编程 (页 433)

参见

可配置的齿轮档调整 (页 441)

13.2.8.3 示例：使用 G331 / G332 攻丝

程序代码	注释
N10 SPOS[n]=0	； 主轴：位置闭环控制运行
	； 起始位置 0 度
N20 G0 X0 Y0 Z2	； 轴：运行至起始位置
N30 G331 Z-50 K-4 S200	； 在 Z 轴上攻丝，
	； 螺距 K-4 负 =>
	； 主轴旋转方向：逆时针，
	； 主轴转速 200 转/分钟
N40 G332 Z3 K-4	； 沿 Z 轴回退，
	； 螺距 K-4 负（逆时针），
	； 自动反向旋转 =>
	； 主轴旋转方向顺时针
N50 G1 F1000 X100 Y100 Z100 S300 M3	； 主轴处于主轴运行中。

13.2.8.4 示例：给出当前齿轮级已编程的钻削转速

程序代码	注释
N05 M40 S500	； 编程设置的主轴转速：500 转/分钟 => ； 齿轮级 1 (20 至 1028 转/分钟)
...	
N55 SPOS=0	； 定位主轴
N60 G331 Z-10 K5 S800	； 攻丝 ； 主轴转速 800 转/分钟 => 齿轮级 1

使用 M40 时通过第一齿轮级数据组测定与编程的主轴转速 S500 匹配的齿轮级。在当前齿轮级中输出编程的钻削转速 S800，并且在必要时将其限制为齿轮级的最大转速。进行 SPOS 后不能进行自动齿轮换档。自动齿轮换档的前提条件是主轴的转速控制运行。

说明

如果主轴转速为 800 rpm，选择了齿轮级 2，此时必须在相应的第二齿轮级数据组的机床数据中设计最大转速和最小转速的开关阈值（参见以下例子）。

13.2.8.5 示例：使用第二齿轮级数据组

第二齿轮级数据组中最大转速和最小转速的开关阈值将在编程 G331/G332 时，以及编程有效主主轴 S 值时计算。自动齿轮换档 M40 必须有效。由此测定的齿轮级会和生效的齿轮级相比较。如果两齿轮级有差别，则进行齿轮换档。

程序代码	注释
N05 M40 S500	； 编程设置的主轴转速：500 转/分钟
...	
N50 G331 S800	； 主主轴：选择齿轮级 2
N55 SPOS=0	； 定位主轴
N60 G331 Z-10 K5	； 攻丝 ； 从第 2 齿轮级数据组加速主轴

13.2.8.6 示例：未进行转速编程，齿轮级监控

如果使用第二齿轮级数据组时，G331 指令中没有编程转速，则会以上次编程的转速加工螺纹。齿轮换档不生效。在此情况下会监控，上次编程的转速是否在当前齿轮级规定的转速范围（最大转速和最小转速开关阈值）内。若不在范围内，则显示报警 16748。

程序代码	注释
N05 M40 S800	； 编程设置的主轴转速：800 转/分钟
...	

程序代码	注释
N55 SPOS=0	; 定位主轴
N60 G331 Z-10 K5	; 攻丝
	; 监控主轴转速 800 转/分钟
	; 齿轮级 1 生效
	; 齿轮级 2 必须生效 => 报警 16748

13.2.8.7 示例：无法进行齿轮换档，齿轮级监控

如果使用第二齿轮级数据组时，在 G331 程序段中除了几何数据之外，还编程了主轴转速，而该转速不在当前齿轮级的规定转速范围（最大转速和最小转速的开关阈值）内，那么将不会执行齿轮换档，因为此时无法保持主轴和进给轴的路径运行。

同上述例子，在 G331 程序段中也会对转速和齿轮级进行监控，并相应地显示报警 16748。

程序代码	注释
N05 M40 S500	; 编程设置的主轴转速：500 转/分钟 =>
	; 齿轮级 1
...	
N55 SPOS=0	; 定位主轴
N60 G331 Z-10 K5 S800	; 攻丝
	; 无法进行齿轮换档，
	; 监控主轴转速 800 转/分钟
	; 带第 1 组齿轮级数据：齿轮级 2
	; 必须生效 => 报警 16748

13.2.8.8 示例：不带 SPOS 的编程

程序代码	注释
N05 M40 S500	; 编程设置的主轴转速：500 转/分钟 =>
	; 齿轮级 1 (20 至 1028 转/分钟)
...	
N50 G331 S800	; 主主轴：选择齿轮级 2
N60 G331 Z-10 K5	; 攻丝
	; 从第 2 齿轮级数据组加速主轴

主轴从当前位置开始执行螺纹插补，该位置由之前执行的零件程序部分决定，比如在执行齿轮换档时。因此可能无法再对螺纹进行再加工。

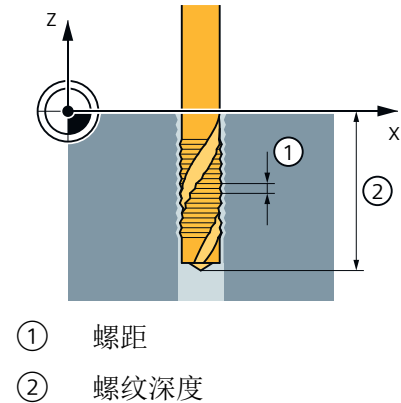
说明

必须注意的是，在使用多主轴加工时钻削主轴必须为主主轴。通过编程 SETMS (<主轴编号>) 可将钻削主轴设置为主主轴。

13.2.9 带弹性卡头的攻丝

13.2.9.1 功能

在带弹性卡头的攻丝（G63）中，不随线性轴对主轴进行插补，而是在转速闭环控制下运行。必须计算并明确编写取决于主轴转速及螺距的线性轴进给率。由于不同对主轴和线性轴进行插补，例如为对主轴的转速波动进行补偿，需要位于弹性卡头中的丝锥。



回退

回退同样通过 G63 编写，但采用相反的主轴旋转方向。

轴及主轴倍率值

在选择了“带弹性卡头的攻丝”功能的情况下，生效的轴及主轴倍率值为 100 %。

进给率

待编写的线性轴进给率 F 为主轴转速与丝锥螺距的乘积：

$$F [\text{mm/min}] = \text{主轴转速 } S [\text{rpm}] * \text{螺距 } [\text{mm/rev}]$$

13.2.9.2 编程

使用指令 G63 编程带补偿夹具的攻丝时，会执行以下运行：

- G63：在钻削方向上攻丝，直至螺纹终点
- G63：设置了主轴旋转方向反向的回退

说明

在 G63 程序段之后，上次生效的插补方式 G0、G1、G2 会再次生效。

句法

G63 <轴> <旋转方向> <转速> <进给率>

含义

G63:	带补偿夹具的攻丝	
	生效方式:	非模态
<轴>:	几何轴（X、Y 或 Z）至螺纹终点的运行行程/位置，例如 Z50	
<旋转方向>:	主轴旋转方向： • M3：顺时针旋转，右螺纹 • M4：逆时针方向，左螺纹	
<转速>:	攻丝时允许的最大主轴转速，例如 S100	
<进给率>:	钻削轴的进给率 F，其中 F = 主轴转速 * 螺距	

示例

钻削 M5 螺纹：

- 螺距符合标准：0.8 毫米/转
- 主轴转速 S：200 转/分钟
- 进给率 F = 200 转/分钟 * 0.8 毫米/转 = 160 毫米/分钟。

程序代码	注释
N10 G1 X0 Y0 Z2 F1000 S200 M3	； 逼近起始点 ； 主轴以顺时针方向旋转，200 转/分钟
N20 G63 Z-50 F160	； 带补偿夹具的攻丝 ； 钻削深度：绝对 Z=50mm ； 进给率：160 mm/min
N30 G63 Z3 M4	； 回退：绝对 Z=3mm ； 旋转方向反向 ； 主轴以逆时针方向旋转，200 转/分钟

13.3 回参考点/同步

为何进行同步?

为了确保控制系统在上电后能够精确识别主轴位置，必须将控制系统与主轴的位置测量系统同步。

仅在经过同步后，主轴才能支持下列功能：

- 螺纹切削
- 刚性攻丝
- 轴编程

对主轴同步的更多说明请见“回参考点 (页 35)”章节。

为何要回参考点?

为了确保控制系统在上电后能够精确识别机床零点，必须将控制系统与回转轴的位置测量系统同步。该操作称为回参考点。回参考点的过程被称为回参考点运行。

仅在执行回参考点后轴才能精确逼近编程的机床上的位置。

对回转轴回参考点的更多说明请见“回参考点 (页 35)”章节。

位置测量系统的安装位置

位置测量系统可如下安装：

- 直接安装在电机上，且主轴上设有充当零脉冲编码器的无接触式接近开关
- 通过测量齿轮箱安装在电机上，且主轴上设有充当零脉冲编码器的无接触式接近开关
- 直接安装在主轴上
- 通过测量齿轮箱安装在主轴上，且主轴上设有充当零脉冲编码器的无接触式接近开关（仅在传动比不等于 1:1 时）

若存在两个位置测量系统，可将其安装在相同的位置，或者分别安装在不同位置。

同步过程

控制系统上电后，主轴按如下过程进行同步：

- 主轴以设定的主轴转速（S 值）和主轴旋转方向（M3 或 M4）启动，并与下一个位置测量系统零脉冲或下一个无接触式接近开关的信号同步。
- 通过 SPOS、M19 或 SPOSA 对静止状态的主轴进行定位。主轴与下一个位置测量系统零脉冲或下一个无接触式接近开关的信号同步。之后定位至编写的位置。
- 可通过 SPOS、M19 或 SPOSA 对运动中（M3 或 M4）的主轴进行同步。
此时具有以下特性：
 - 使用 SPOS=<位置>、SPOS=DC (<位置>) 和 SPOS=AC (<位置>) 时，保持运动方向逼近位置。
 - 使用 SPOS=ACN (<位置>) 或 SPOS=ACP (<位置>) 时将始终采用正方向或负方向逼近。必要时需在定位前反转运动方向。
- 在 JOG 运行方式下通过方向键在转速控制运行中越过零脉冲。

说明

通过零件程序、FC 18 或同步动作触发的同步具有相同的效果。

说明

在主轴同步中，所有四个参考点值根据所选测量系统生效。测量系统偏移也以同样的方式生效。

请注意下列机床数据：

- MD34080 \$MA_REFP_MOVE_DIST
(使用距离编码系统时的参考点距离/目标点)
- MD34090 \$MA_REFP_MOVE_DIST_CORR
(距离编码的参考点偏移/绝对偏移)
- MD34100 \$MA_REFP_SET_POS
(参考点值，对距离编码系统无意义)

若通过 SPOS=IC(...) 和 <360 度的行程定位未回参考点的主轴，可能无法越过零脉冲，主轴位置仍未和零脉冲同步。这可能出现于以下情形：

- 上电后
- 通过设置轴专用 NC/PLC 接口信号：
 - <Axis>.spindle.out.resyncEnc2ForPos (定位中重新同步主轴 2)
 - <Axis>.spindle.out.resyncEnc1ForPos (定位中重新同步主轴 1)

通过无接触式接近开关同步时的特殊性

无接触式接近开关的信号延时所引起的位置错误可于 NC 内部通过输入信号运行时间补偿进行修正。

13.3 回参考点/同步

信号运行时间补偿通过以下机床数据设置：

- MD31122 \$MA_BERO_DELAY_TIME_PLUS
（无接触式接近开关的延时，正运动方向）
- MD31123 \$MA_BERO_DELAY_TIME_MINUS
（无接触式接近开关的延时，负运动方向）

作用取决于设置的回参考点模式：

MD34200 \$MA_ENC_REFP_MODE = <值>

<值>	含义
2	在不设定特定定位速度/转速的情况下执行位置同步。
7	将以下机床数据中的速度用作位置同步过程中的定位速度。 MD34040 \$MA_REFP_VELO_SEARCH_MARKER（关闭速度） 零脉冲搜索不会自动执行，而是需要通过以下 NC/PLC 信号的 0/1 脉冲沿显性请求： <Axis>.spindle.out.resyncEnc1 / <Axis>.spindle.out.resyncEnc2（重新同步主轴，测量系统 1/2）

说明

供货状态下已对信号运行时间补偿进行了相应预设，通常不需要修改。

回参考点过程

控制系统启动后，若需直接在进给轴模式下编写主轴，必须确保该轴已回参考点。

控制系统上电后，可对主轴执行回参考点（前提条件时每转一个零脉冲）。

回参考点的步骤参见“回参考点 (页 35)”章节。

同步主轴（参见“同步过程”部分）时，若主轴的位置测量系统也用于回转轴，那么回转轴也会同时进行回参考点。

主轴位置测量系统

为主轴模式和进给轴模式使用的是一个电机时，主轴可从主轴模式切换至进给轴模式（回转轴）。

主轴（主轴模式和进给轴模式）可配备一到两个位置测量系统。启用两个位置测量系统时，可将其中的一个指定给主轴，将另一个指定给回转轴，或者将两个位置测量系统都指定给主

轴。若存在两个位置测量系统，则其均由控制系统更新，不过只能有一个位置测量系统生效。

通过以下 NC/PLC 接口信号选择生效的位置测量系统：

`<Axis>.basic.out.enc1Activation / <Axis>.basic.out.enc2Activation`
(位置测量系统 1/2)

生效的位置测量系统应用于以下功能：

- 主轴位置闭环控制 (SPCON)
- 主轴定位 (SPOS、M19 和 SPOSA)
- 螺纹切削 (G33、G34、G35)
- 刚性攻丝 (G331、G332)
- 旋转进给率 (G95)
- 恒定切削速度 (G96、G961、G97、G971)
- 主轴转速实际值显示
- 进给轴模式
- 同步主轴设定值耦合

重新同步主轴的位置测量系统

在以下情形下，必须重新对主轴的位置测量系统执行同步：

- 位置测量编码器安装在电机上，无接触式接近开关安装在主轴上，并进行了齿轮档切换。主轴以新的齿轮档旋转时，系统内部触发同步（参见同步过程部分）。
- 机床可在垂直主轴和水平主轴之间进行切换。此时使用两个不同的位置测量编码器（一个用于水平主轴，一个用于垂直主轴），但只使用一个控制系统上的实际值输入。若在水平主轴和垂直主轴之间进行了切换，必须重新执行同步。

同步通过以下 NC/PLC 接口信号触发：

`<Axis>.spindle.out.resyncEnc1` (重新同步主轴 1)

或

`<Axis>.spindle.out.resyncEnc2` (重新同步主轴 2)

为此主轴必须处于控制模式下。

在热启动后恢复位置

对于使用增量位置测量系统的主轴，可在断电后将实际值继续保存在缓存中，并在上电后恢复为断电前最后保存的位置，这样一来例如可对转换等位置相关功能执行恢复（参见“自动恢复机床参考(页 84)”章节）。其应用例如有：使用刀具定向加工时在热启动后退刀（参见功能手册之基本功能：“上电后用定向坐标转换退刀”章节）。

位置恢复后，位置测量系统的状态通过以下 NC/PLC 接口信号显示：

<Axis>.basic.in.enc1Restored（“已恢复 1”），对应位置测量系统 1

<Axis>.basic.in.enc2Restored（“已恢复 2”），对应位置测量系统 2

在 JOG 运行方式下退刀后，必须对恢复位置的轴执行回参考点。这样一来信号

<Axis>.basic.in.enc1Restored/<Axis>.basic.in.enc2Restored（“已恢复 1/2”）会被删除，信号 <Axis>.basic.in.enc1Synchronized/

<Axis>.basic.in.enc2Synchronized（“已回参考点/已同步 1/2”）会被置位。

说明

若机床数据 MD20700 \$MC_REFP_NC_START_LOCK 设置为“2”，则可使用“恢复”的轴位置进行 NC 启动（MDI 运行方式下或溢出转存时）。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.enc1Activation	LBP_Axis*.A_PosMeas1	DB31, ...DBX1.5
<Axis>.basic.out.enc2Activation	LBP_Axis*.A_PosMeas2	DB31, ...DBX1.6
<Axis>.spindle.out.resyncEnc1	LBP_Axis*.A_Sp1Syn	DB31, ...DBX16.4
<Axis>.spindle.out.resyncEnc2	LBP_Axis*.A_Sp2Syn	DB31, ...DBX16.5
<Axis>.spindle.out.resyncEnc1ForPos	LBP_Axis*.A_Sp1SynP	DB31, ...DBX17.4
<Axis>.spindle.out.resyncEnc2ForPos	LBP_Axis*.A_Sp2SynP	DB31, ...DBX17.5

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.enc1Synchronized	LBP_Axis*.E_RefSyn1	DB31, ...DBX60.4
<Axis>.basic.in.enc2Synchronized	LBP_Axis*.E_RefSyn2	DB31, ...DBX60.5

Basic Program Plus	Basic Program	
<Axis>.basic.in.enc1Restored	LBP_Axis*.E_POS_RESTORED1	DB31, ...DBX71.4
<Axis>.basic.in.enc2Restored	LBP_Axis*.E_POS_RESTORED2	DB31, ...DBX71.5

13.4 可配置的齿轮档调整

13.4.1 主轴的齿轮档和齿轮档切换

为什么要启用齿轮档？

主轴的齿轮档用于实现电机转速降档，从而在较低的主轴转速下生成较高的转矩，或用于升档，从而获取较高的转速。

齿轮档数量

可为每根主轴配置 5 个齿轮档。

使用的齿轮档的数量通过以下机床数据定义：

MD35090 \$MA_NUM_GEAR_STEPS

齿轮档的参数设置

齿轮档 1 至 5 可通过下列机床数据进行设置：

机床数据	含义
MD35012 \$MA_GEAR_STEP_CHANGE_POSITION[<n>]	齿轮档切换位置
MD35110 \$MA_GEAR_STEP_MAX_VELO[<n>]	自动齿轮档切换的最大转速
MD35120 \$MA_GEAR_STEP_MIN_VELO[<n>]	自动齿轮档切换的最小转速
MD35130 \$MA_GEAR_STEP_MAX_VELO_LIMIT[<n>]	齿轮档的最大转速
MD35135 \$MA_GEAR_STEP_PC_MAX_VELO_LIMIT[<n>]	位置闭环控制下齿轮档的最大转速
MD35140 \$MA_GEAR_STEP_MIN_VELO_LIMIT[<n>]	齿轮档的最小转速
MD35200 \$MA_GEAR_STEP_SPEEDCTRL_ACCEL[<n>]	转速控制运行下的加速度
MD35210 \$MA_GEAR_STEP_POSCTRL_ACCEL[<n>]	位置闭环控制运行下的加速度

13.4 可配置的齿轮档调整

机床数据	含义
MD35300 \$MA_SPIND_POSCTRL_VELO[<n>]	位置闭环控制激活转速
MD35310 \$MA_SPIND_POSIT_DELAY_TIME[<n>]	定位延时
MD35550 \$MA_DRILL_VELO_LIMIT[<n>]	刚性攻丝的最大转速

齿轮档切换类型

齿轮档切换的类型通过以下机床数据设置：

MD35010 \$MA_GEAR_STEP_CHANGE_ENABLE

位	值	含义
0	0	主轴电机直接（1:1）或以不可变的传动比安装至主轴（缺省设置）。 第一齿轮档的机床数据生效。
	1	主轴电机具有最多 5 个齿轮档。 齿轮档切换如下进行： • 往复模式下 • 在不确定的切换位置上
1	0	含义同“位 0 = 0”。
	1	含义同“位 0 = 1”，但是齿轮档切换在配置的主轴位置上进行。 切换位置在以下机床数据中设置： MD35012 \$MA_GEAR_STEP_CHANGE_POSITION 齿轮档切换前会以当前齿轮档逼近该位置。 位 1 置位时，则可忽略位 0！
3	1	对 NC 和 PLC 间的齿轮档切换信号对话进行仿真。
5	1	在 G331/G332 攻丝中使用第二齿轮档数据组（参见下文中的“第二齿轮档数据组”部分）。必须为攻丝时使用的主主轴设置该位。

齿轮档切换的前提条件

原则上，仅在请求的齿轮档不同于生效齿轮档时，才会执行齿轮档切换。

齿轮档切换时的参数组选择

启用以下设置时，伺服参数组也会随齿轮档切换：

MD35590 \$MA_PARAMSET_CHANGE_ENABLE = 0 或 1

更多信息参见章节“齿轮档切换时的参数组选择 (页 456)”。

请求齿轮档切换

可通过以下方法请求齿轮档切换：

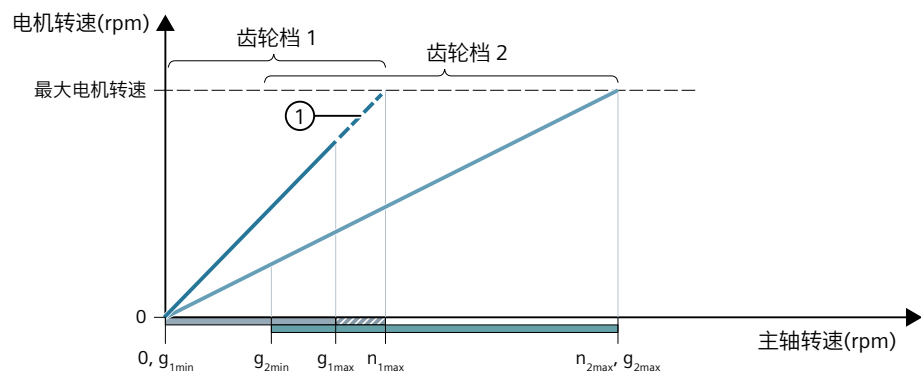
- 在零件程序中通过：
 - M40 S...
为编写的转速 S... 自动选择齿轮档
 - M41 ... M45
直接选择齿轮档 1 ... 5
 - M70
基于 MD35014 \$MA_GEAR_STEP_USED_IN_AXISMODE = 1 ... 5
(参见“执行 M70 时启用的齿轮档 (页 479)”)
 - G331 S...
基于 MD35010 \$MA_GEAR_STEP_CHANGE_ENABLE, 位 5 = 1
- 在同步动作中通过：
 - DO M40 S...
为编写的转速 S... 自动选择齿轮档
 - DO M41...M45
直接选择齿轮档 1 ... 5
 - DO M70
基于 MD35014 \$MA_GEAR_STEP_USED_IN_AXISMODE = 1 ... 5
- 通过 PLC 使用 SINU_RotateSpindle (Basic Program - FC18)
- 在复位状态下通过描述 NC/PLC 接口：
<Axis>.spindle.out.gearStageSelection (实际齿轮档 A 至 C)
特别是在上电后，可借此将机械生效的齿轮档信息提供给 NC。

说明

主轴电机直接 (1:1) 或以不可变的传动比安装至主轴时 (MD35010 = 0)，辅助功能 M40 和 M41 ... M45 对该主轴无意义。

齿轮档选择

下面的示例中显示的主轴转速设定为最大值时两个齿轮档之间的选择：



可通过 MD35110 GEAR_STEP_MAX_VELO 指定：

- n_{1max} ... 第 1 齿轮档的最大主轴转速
- g_{1min} ... 第 1 齿轮档的最小主轴转速，用于自动齿轮档选择
- g_{1max} ... 第 1 齿轮档的最大主轴转速，用于自动齿轮档选择
- n_{2gmax} ... 第 2 齿轮档的最大主轴转速
- g_{2min} ... 第 2 齿轮档的最小主轴转速，用于自动齿轮档选择
- g_{2max} ... 第 2 齿轮档的最大主轴转速，用于自动齿轮档选择

图 13-3 齿轮档切换，在两个齿轮档之间选择

齿轮档切换过程

预选了新的齿轮档时，切换过程如下：

1. 切换

以下两个 NC/PLC 接口信号置位：

<Axis>.spindle.in.requestedGearStageNumber (设定齿轮档 A 至 C)

<Axis>.spindle.in.gearChangeoverRequested (齿轮箱换档)

根据以下 NC/PLC 接口信号置位的时间：

<Axis>.spindle.out.oscillEnable (往复转速)

主轴启用往复加速度或转速控制运行/位置闭环控制运行加速度制动，直至到达停止状态。

最迟在主轴达到静止状态后：

<Axis>.basic.in.isStationary (进给轴/主轴停止)

可通过以下 NC/PLC 接口信号激活往复运行：

<Axis>.spindle.out.oscillEnable (往复转速)

。

原则上不使用往复运行也能切换至新的齿轮档。

切换至新的齿轮档后，PLC 程序会将下面两个 NC/PLC 接口信号置位：

<Axis>.spindle.out.gearStageSelection (实际齿轮档 A 至 C)

<Axis>.spindle.out.gearChangeoverDone (齿轮箱已换档)

2. 齿轮档切换结束

以下 NC/PLC 接口信号置位时，齿轮档切换结束（主轴运行方式“往复模式”取消）：

<Axis>.spindle.out.gearChangeoverDone (齿轮箱已换档)

在电机静止状态下，参数组切换为新的实际齿轮档的伺服参数组和插补参数组。

通过 NC/PLC 接口信号：

<Axis>.spindle.out.gearChangeoverDone (齿轮箱已换档)

NC 得到反馈信息：新齿轮档生效，可结束往复运行。

NC/PLC 接口信号：

<Axis>.spindle.in.gearChangeoverRequested (齿轮箱换档)

由 NC 复位，

并触发 PLC 程序对以下 NC/PLC 接口信号的复位：

<Axis>.spindle.out.gearChangeoverDone (齿轮箱已换档)

。

此时以下 NC/PLC 接口信号设置与否则不重要：

<Axis>.spindle.out.oscillEnable (往复转速)

。

选择参数组时，信号提示的实际齿轮档为相关信息，

其应与设定齿轮档相同。

若非此情形，则会在启用以下机床数据设置时：

MD11410 \$MN_SUPPRESS_ALARM_MASK, 位 3 = 0

触发报警 22010。

PLC

通过以下 NC/PLC 接口信号应答齿轮档切换后：

<Axis>.spindle.out.gearChangeoverDone (齿轮箱已换档)

主轴处于转速控制运行中 (<Axis>.spindle.in.ctrlModeActive)。

PLC 和 NC 间信号交换的**更多信息**参见功能手册之基本功能。

第二齿轮档数据组

自动齿轮档切换 M40 可通过另一个可配备的齿轮档数据组扩展。

13.4 可配置的齿轮档调整

第二齿轮档数据组只和刚性攻丝（G331、G332）功能配合使用，用于实现对主轴转速和电机转矩的高效调整。

为主主轴设置以下位来激活此功能：

MD35010 \$MA_GEAR_STEP_CHANGE_ENABLE，位 5 = 1

第二齿轮档数据组使用的齿轮档的数量通过以下机床数据定义：

MD35092 \$MA_NUM_GEAR_STEPS2

启用以下设置时，第二齿轮档数据组将被取消激活：

MD35092 \$MA_NUM_GEAR_STEPS2 = 0（缺省设置）

此时若 M40 生效，那么齿轮档选择将基于第一齿轮数据组进行。

说明

第二数据组中的齿轮档数量可与第一个数据组中不同。启用 M40 时，若未找到与编写的转速相匹配的齿轮档，则不执行齿轮档切换（特例参见“M40：转速超出配置的开关阈值时的自动齿轮档选择 (页 509)”）。

刚性攻丝 G331/G332 的典型程序流程的更多信息请参见编程手册之 NC 编程。

第二齿轮档数据组的齿轮档 1 至 5 可通过下列机床数据进行设置：

机床数据	含义
MD35112 \$MA_GEAR_STEP_MAX_VELO2[n]	自动齿轮档切换的最大转速
MD35122 \$MA_GEAR_STEP_MIN_VELO2[n]	自动齿轮档切换的最小转速
MD35212 \$MA_GEAR_STEP_POSCTRL_ACCEL2[n]	位置闭环控制运行下的加速度

说明

机械条件相关的伺服参数数组的数量保持不变。还可为主轴配置五个机械齿轮档，为进给轴模式配置一个。

转速限制针对每个齿轮档通过以下机床数据配置，与各种开关阈值无关。

机床数据	含义
MD35130 \$MA_GEAR_STEP_MAX_VELO_LIMIT[n]	齿轮档的最大转速
MD35140 \$MA_GEAR_STEP_MIN_VELO_LIMIT[n]	齿轮档的最小转速

对于刚性攻丝（G331、G332），还可将转速限制在电机的线性加速度范围内。为此，在以下机床数据中针对齿轮档设定线性电机特性曲线范围的转速上限：

```
MD35550 $MA_DRILL_VELO_LIMIT[n]
```

在零件程序中设定齿轮档

M40 生效时自动选择齿轮档

齿轮档由控制系统自动选择。此时系统会检查哪一个齿轮档适用于编写的主轴转速（S...）。若能确定一个与当前实际齿轮档不同的齿轮档，则会置位以下 NC/PLC 接口信号：

```
<Axis>.spindle.in.gearChangeoverRequested (齿轮箱换档)
<Axis>.spindle.in.requestedGearStageNumber (设定齿轮档 A 至 C)
```

确定匹配的齿轮档后，仅当新转速不在当前齿轮档所允许的范围内时，系统才会请求齿轮档切换。

必要时，系统会将转速限制为当前齿轮档的最大转速，或提升至最小转速，此时以下 NC/PLC 接口信号将置位：

```
<Axis>.spindle.in.speedSetpointAboveMax (设定转速已限制)
<Axis>.spindle.in.speedSetpointBelowMin (设定转速已提升)
```

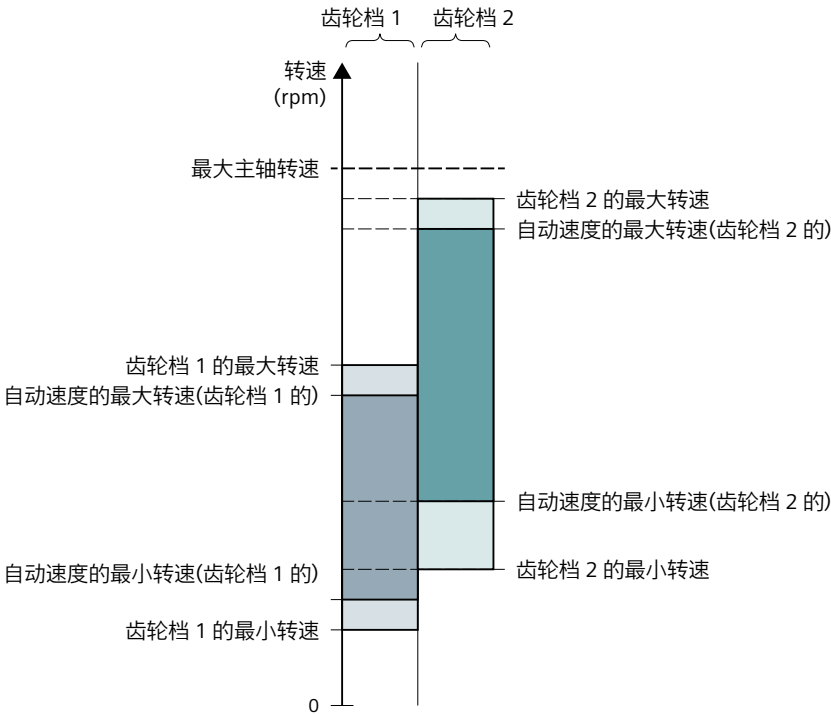


图 13-4 示例：自动齿轮档选择（M40）中两个转速范围相叠的齿轮档

说明

使用 M40 进行自动齿轮档切换时，主轴必须处于控制模式下并启用一个 S 字。否则齿轮档切换会被拒绝，并发出以下报警：

Alarm 22000“无法切换齿轮档”

说明

选择自动齿轮档切换时，生效的附加中间齿轮比不会被考虑在内。

通过 M41 至 M45 固定设定齿轮档

齿轮档可在零件程序中通过 M41 至 M45 固定设定。

若通过 M41 至 M45 设定了一个与当前实际齿轮档不同的齿轮档，那么以下 NC/PLC 接口信号将置位：

<Axis>.spindle.in.gearChangeoverRequested（齿轮箱换挡）

<Axis>.spindle.in.requestedGearStageNumber（设定齿轮档 A 至 C）

此时，编写的主轴转速（S...）将基于该固定设定的齿轮档：

- 若编写了一个超出固定设定齿轮档转速上限
（MD35130 \$MA_GEAR_STEP_MAX_VELO_LIMIT）的主轴转速，系统会将其限制为上限值，并置位以下 NC/PLC 接口信号：
<Axis>.spindle.in.speedSetpointAboveMax（设定转速已限制）
- 若编写了一个低于固定设定齿轮档转速下限
（MD35140 \$MA_GEAR_STEP_MIN_VELO_LIMIT）的主轴转速，系统会将其提升为下限值，并置位以下 NC/PLC 接口信号：
<Axis>.spindle.in.speedSetpointBelowMin（设定转速已提升）

程序段转换

在零件程序段中编写齿轮档切换时，齿轮档切换程序段一直保持生效，直至 PLC 结束齿轮档切换。

此效果相当于以下 NC/PLC 接口信号置位：

<Chan>.basic.out.disableReadIn（读入禁止）

由 PLC 使用 SINU_RotateSpindle (Basic Program - FC18) 设定齿轮档

可使用功能块 FC18 在复位状态下以及所有运行方式下设定齿轮档切换。

使用 FC18 设定转速和旋转方向时，可向 NC 请求选择与转速相匹配的齿轮档。这相当于 M40 下的自动齿轮档切换。

以下情形下齿轮档不会切换：

- 主轴通过 SINU_PositionSpindel (Basic Program - FC18) 定位。
- 主轴在进给轴模式下运行。

更多信息

功能块 SINU_PositionSpindel (Basic Program - FC18) 的更多详细信息参见 PLC 功能手册。

在同步动作中设定齿轮档

在同步动作中可通过以下方式请求齿轮档切换：

- DO M40 S...
为编写的转速 s... 自动选择齿轮档
- DO M41...M45
直接选择齿轮档 1 ... 5
- DO M70
基于 MD35014 \$MA_GEAR_STEP_USED_IN_AXISMODE = 1 ... 5
(参见“执行 M70 时启用的齿轮档 (页 479)”)

以下情形下齿轮档**不会**切换：

- 主轴通过同步动作定位。
- 主轴在进给轴模式下运行。

说明

更多信息请见“在零件程序中设定齿轮档”部分。

特例：

在同步动作中设定齿轮档不会影响程序段转换。

手动设定齿轮档

不借助零件程序，不通过 NC 请求也可在机床上切换齿轮档。这例如可通过直接手动切换齿轮档来实现。

为了选择匹配的参数组，必须将当前齿轮档的信息提供给 NC。为此，控制系统或零件程序必须处于复位状态。

边界条件

13.4 可配置的齿轮档调整

修改

以下 NC/PLC 接口将齿轮档接收至 NC:

`<Axis>.spindle.out.gearStageSelection` (实际齿轮档 A 至 C)

这三个位必须在运行期间保持稳定。

接收完成时会通过以下 NC/PLC 接口信号对 PLC 进行应答:

`<Axis>.spindle.in.requestedGearStageNumber` (设定齿轮档 A 至 C)

。

以下 NC/PLC 接口信号不可置位:

`<Axis>.spindle.out.gearChangeoverDone` (齿轮箱已换档)

。

若通过 PLC 使用 `<Axis>.spindle.out.gearStageSelection` 设定新齿轮档时位置闭环控制生效, 则须在切换期间将其取消。

齿轮档切换期间的 NC 停止

以下情形下将无法通过 NC/PLC 接口信号:

`<Chan>.basic.out.ncStopAxesAndSpindles` (NC 停止)

实现主轴停止:

- 主轴尚不处于用于齿轮档切换的往复模式下。
- 以下 NC/PLC 接口信号不存在:
`<Axis>.spindle.out.gearChangeoverDone` (齿轮箱已换档)
 。

说明

终止方法:

`<Axis>.basic.out.resetMovement` (删除剩余行程/主轴复位)

或

`<Axis>.spindle.out.gearChangeoverDone` (齿轮箱已换档)

与对应的实际齿轮档反馈信息:

`<Axis>.spindle.out.gearStageSelection` (实际齿轮档)

齿轮档切换后的主轴特定

齿轮档切换完成后，主轴的特性取决于以下初始条件：

- 如果在齿轮档切换前主轴处于停止状态（M5、SINU_RotateSpindle (Basic Program - FC18): "stop"）、处于定位模式或进给轴模式下，则 M5（主轴停止）在齿轮档切换后生效。
- 如果在齿轮档切换前编写了旋转方向（M3、M4、SINU_RotateSpindle (Basic Program - FC18): "start"），则最后编写的旋转方向和转速在齿轮档切换后重新生效。主轴会以新的齿轮档加速至最后编写的主轴转速（S...）。
- 若齿轮档切换前位置闭环控制生效（SPCON），那么其会在齿轮档切换后被重新激活。可开始执行零件程序中的下一个程序段。

特点

对于齿轮档切换须注意以下几点：

- 选择以下 NC/PLC 接口信号时：
`<Axis>.drive.out.rapidStop`（V/f 运行启动切换）
齿轮档切换不会结束。

此时会输出设定值 0。
齿轮档切换的应答通过以下 NC/PLC 接口信号进行：
`<Axis>.spindle.out.gearChangeoverDone`（齿轮箱已换档）
- 通过 PLC 结束齿轮档切换前，必须将信号“斜坡函数发生器快速停止”复位。
- NC 复位时，齿轮档切换过程结束，不输出报警。
NC/PLC 接口信号：
`<Axis>.spindle.out.gearStageSelection`（实际齿轮档 A 至 C）
发送的齿轮档会由 NC 接收。

使用 SINU_CtrlStarDelta (Basic Program FC17) 进行星/三角切换

对于数字化主主轴驱动，在主轴运转时也可通过 SINU_CtrlStarDelta 沿两个方向从星形切换为三角形连接。这项自动切换操作由 SINU_CtrlStarDelta 中定义的切换逻辑控制，切换时间可由用户针对具体主轴进行设置。

更多信息

功能块 SINU_CtrlStarDelta 的更多详细信息参见 PLC 功能手册。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.resetMovement	LBP_Axis*.A_DeIDTGSReset	DB31, ...DBX2.2
<Chan>.basic.out.disableReadIn	LBP_Chan*.A_Rldisable	DB21, ...DBX6.1
<Chan>.basic.out.ncStopAxesAndSpindles	LBP_Chan*.A_NCStopASp	DB21, ...DBX7.4
<Axis>.spindle.out.gearStageSelection	LBP_Axis*.A_ActGearA..C	DB31, ...DBX16.0.. 16.2
<Axis>.spindle.out.gearChangeoverDone	LBP_Axis*.A_GearChangeOv	DB31, ...DBX16.3
<Axis>.spindle.out.oscillEnable	LBP_Axis*.A_OscilSpeed	DB31, ...DBX18.5
<Axis>.drive.out.rapidStop	-	DB31, ...DBX20.1

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.isStationary	LBP_Axis*.E_Stat	DB31, ...DBX61.4
<Axis>.basic.in.posCtrlActive	LBP_Axis*.E_PositContr	DB31, ...DBX61.5
<Axis>.spindle.in.requestedGearStageNumber	LBP_Axis*.E_SetpGearA..C	DB31, ...DBX82.0.. 2
<Axis>.spindle.in.gearChangeoverRequested	LBP_Axis*.E_GearChange	DB31, ...DBX82.3
<Axis>.spindle.in.speedSetpointAboveMax	LBP_Axis*.E_SetSpeedLimit	DB31, ...DBX83.1
<Axis>.spindle.in.speedSetpointBelowMin	LBP_Axis*.E_SetSpeedIncreased	DB31, ...DBX83.2
<Axis>.spindle.in.speedSetpointReached	LBP_Axis*.E_SetRange	DB31, ...DBX83.5
<Axis>.spindle.in.oscillModeActive	LBP_Axis*.E_OscillMode	DB31, ...DBX84.6
<Axis>.spindle.in.ctrlModeActive	LBP_Axis*.E_ContrMode	DB31, ...DBX84.7

13.4.2 主轴齿轮档 0

技术背景

对于使用可切换主轴负载齿轮箱的机床，有时可能会出现电机和负载（工件/刀具）间的传动链断开的情形。例如在执行齿轮档切换期间按下复位或急停键，或者安装期间首次调试时都可能出现此状态。控制系统必须能够识别出传动链断开的状态，并执行下一次齿轮档切换。

功能

对于脱接的齿轮箱，PLC 会通过接口信号

位 `<Axis>.spindle.out.gearStageSelection`（实际齿轮档 A 至 C）将二进制编码值“0”（≙ 齿轮档 0）传输至 NC：

```
<Axis>.spindle.out.gearStageSelection = 0
```

该值用于向控制系统提示传动链断开的状态。

对齿轮档切换的影响

零件程序中的齿轮档切换

PLC 发出的实际齿轮档信号在零件程序开始时由 NC 读取。若此时读取到的实际齿轮档为“0”，那么 NC 将执行下一次齿轮档切换，并执行与 PLC 的齿轮档切换信号对话。若读取的值大于“0”，那么编程时即完成了请求的齿轮档和生效齿轮档之间的比较。若两个齿轮档相同，则不执行齿轮档切换；若编写有轨迹运行，则其也不会中断。

同步动作、SINU_RotateSpindle (Basic Program) 和 DBB30 中的齿轮档切换

进行齿轮档切换时，PLC 发出的实际齿轮档信号总是会由 NC 进行分析。读取值为“0”时，NC 总是会执行齿轮档切换。若读取值大于“0”，则会比较请求的齿轮档和生效齿轮档。仅当两个值不同时，才会通过 PLC 执行齿轮档切换，并输出 NC/PLC 接口信号

`<Axis>.spindle.in.gearChangeoverRequested`（齿轮箱换档）。

边界条件

- **输出 `<Axis>.spindle.out.gearStageSelection = 0`**

对于脱接的齿轮箱，必须通过 PLC 将齿轮档 0 输入至 NC/PLC 接

口 `<Axis>.spindle.out.gearStageSelection`（实际齿轮档 A 至 C）。

- **使能齿轮档切换**

读取到齿轮档 0 后，齿轮档切换的前提条件是齿轮档切换的总体使能：

MD35010 \$MA_GEAR_STEP_CHANGE_ENABLE（参数设置齿轮档切换）

MD35090 \$MA_NUM_GEAR_STEPS（建立的齿轮档的数量）

MD35092 \$MA_NUM_GEAR_STEPS2（第 2 齿轮档数据组：建立的齿轮档的数量），基于 MD35010 \$MA_GEAR_STEP_CHANGE_ENABLE，位 5 = 1（刚性攻丝）

- **PLC 用户程序/上电 ASUB**

通过 PLC 用户程序或上电 ASUB，能够在齿轮档脱接（齿轮档 0）的情形下确保在主轴运动前编写一个齿轮档切换请求。这例如可通过在 ASUB 中编写 M41 实现。主轴运动（例如 JOG 运行方式下或进给轴模式下）本身不会生成齿轮档切换。

示例

示例：上电后切换至第一齿轮档

- 1. 上电。
- 2. PLC 用户程序在机械环境中确定“齿轮档已脱接”状态。
- 3. PLC 通过置位以下信号将“齿轮档已脱接”状态传输至 NC：
 <Axis>.spindle.out.gearStageSelection = 0
- 4. 零件程序启动或上电 ASUB。
- 5. 执行 N05（零件程序见下文）：
 执行向齿轮档 1 的切换。
 NC：
 - 置位以下 NC/PLC 接口信号：
 <Axis>.spindle.in.gearChangeoverRequested（齿轮箱换档）
 - 将设定齿轮档 1 的信息发送给 PLC：
 <Axis>.spindle.in.requestedGearStageNumber = 1
- 6. 齿轮档的机械切换，应答
 若切换了齿轮档，则由 PLC：
 - 置位以下 NC/PLC 接口信号：
 <Axis>.spindle.out.gearChangeoverDone（齿轮箱已换档）
 - 将实际齿轮档 1 的信息发送给 NC：
 <Axis>.spindle.out.gearStageSelection = 1
- 7. 执行 N80：
 基于优化齿轮档切换频率的原则，不执行齿轮档切换。

零件程序：

程序代码	注释
N05 M41	; 选择齿轮档 1
...	
N80 M41	; 若切换至齿轮档 1，则不执行齿轮档切换。

主轴 1（AX5）的配置数据：

MD35010 \$MA_GEAR_STEP_CHANGE_ENABLE[AX5] = 1（使能齿轮档切换）

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.spindle.out.gearStageSelection	LBP_Axis*.A_ActGearA..C	DB31, ...DBX16.0..2
<Axis>.spindle.out.gearChangeoverDone	LBP_Axis*.A_GearChangeOv	DB31, ...DBX16.3

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.spindle.in.requestedGearStageNumber	LBP_Axis*.E_SetpGearA..C	DB31, ...DBX82.0..2
<Axis>.spindle.in.gearChangeoverRequested	LBP_Axis*.E_GearChange	DB31, ...DBX82.3

13.4.3 确定主轴齿轮档

主轴的当前齿轮档可通过系统变量读取：

- 在同步动作或零件程序（会触发预处理停止）中写入以下系统变量，状态显示在操作界面中：

\$VC_SGEAR[<n>]

当前换入的主轴齿轮档
\$VC_SGEAR 读取的是 PLC 发送的实际齿轮档。
取值范围：0 ... 5

\$AC_SGEAR[<n>]

生效的主轴齿轮档
\$AC_SGEAR 读取的是主处理中的设定齿轮档。
取值范围：1 ... 5
根据此齿轮档为主轴激活数据组。

13.4 可配置的齿轮档调整

说明
程序段搜索时，实际齿轮档（\$VC_SGEAR[<n>]）可能有别于设定齿轮档（\$AC_SGEAR[<n>]），这是因为程序段搜索期间未进行齿轮档切换。因此借助 \$VC_SGEAR[<n>] 和 \$AC_SGEAR[<n>] 也可查询程序段搜索后是否需要执行齿轮档切换。

- 在不带预处理停止的零件程序中，通过系统变量：

\$P_SGEAR[<n>]	设定齿轮档 \$P_SGEAR 读取的是零件程序中编写的（M41 ... M45）、M40 下选择的或 M70 下配置的齿轮档。
\$P_SEARCH_SGEAR[<n>]	程序段搜索:齿轮档 M 功能 \$P_SEARCH_SGEAR 读取的是程序段搜索中收集的、最后编写的齿轮档 M 功能。

13.4.4 齿轮档切换时的参数组选择

伺服参数组

伺服参数组 1 至 6 用于在主轴齿轮档切换时根据机床的属性变化调整位置控制器。

齿轮档切换时的参数组选择

齿轮档切换时，齿轮档参数组（插补参数组）也会随之切换。此外根据以下机床数据中的设置，伺服数据组也可能切换：

MD35590 \$MA_PARAMSET_CHANGE_ENABLE（参数组切换使能）

值	含义
0	系统内部的参数组选择 伺服参数组采用固定分配。 适用原则： - 对于进给轴模式下的轴和主轴，一般情况下第一个参数组生效。 特例： 使用 G33、G34、G35、G331 和 G332 时，为相关轴激活下列编号的数据组： 主轴齿轮档 + 1（对应参数组编号 2 ... 6） - 对于主进给轴模式下的主轴，系统会根据齿轮档设置匹配的参数组。
1	除了系统内部的参数组选择外，也可进行“外部”参数组选择： - 通过 PLC (<Axis>.basic.out.posCtrlParameterSetSelection) - 通过在零件程序或同步动作中写入指令 SCPARA 不过系统内部参数组选择具有更高优先级。 提示：值 1 只适用于轴。
2	伺服参数组只通过 PLC (<Axis>.basic.out.posCtrlParameterSetSelection) 或在零件程序或同步动作中编程 SCPARA 来设定（用于轴和主轴）。 上电后选择参数组 1。

主轴模式

MD35590 \$MA_PARAMSET_CHANGE_ENABLE = 0 或 1

参数组按照“齿轮档 + 1”选择。

生效的齿轮档通过以下接口获取：

<Axis>.spindle.out.gearStageSelection（实际齿轮档 A 至 C）

生效的参数组在以下接口输出：

<Axis>.drive.in.posCtrlParameterSetNumber（控制器参数组 A 至 C）

13.4 可配置的齿轮档调整

5 个齿轮档每个都有一个参数组，由 NC 如下指定：

数据组	NC/PLC 接口 <Axis>.drive.in.posCtrlParameterSetNumber	参数组 编号	参数组 索引 [n]
进给轴模式	最后生效的齿轮档	1	0
齿轮档 1	001	2	1
齿轮档 2	010	3	2
齿轮档 3	011	4	3
齿轮档 4	100	5	4
齿轮档 5	101 110 111	6	5

进给轴模式下的主轴

若主轴处于进给轴模式下，那么伺服中会选择参数组索引“0”（注意 MD35590 \$MA_PARAMSET_CHANGE_ENABLE！）

齿轮档切换特性取决于以下机床数据中的设置：

MD35014 \$MA_GEAR_STEP_USED_IN_AXISMODE（M70 下进给轴模式的齿轮档）

若没有为进给轴模式配置齿轮档（MD35014 = 0），M70 下将不会自动执行齿轮档切换（缺省设置）。最后的齿轮档会保存在系统内部，并在下一次主轴编程时与对应参数组一起重新生效。

相反的情形下，若为进给轴模式配置了齿轮档（MD35014 = 1 ... 5），那么在执行 M70 期间会切换至齿轮档 1 ... 5。在从进给轴模式切换至主轴模式时，通过 M70 换入的齿轮档保持生效。主轴模式下在 M70 前生效的齿轮档不会被自动重新换入。

另见“执行 M70 时启用的齿轮档 (页 479)”。

负载齿轮箱系数

可在进给轴模式下为每个齿轮档配置一个正的或负的**负载齿轮箱系数**。

此系数分为分子和分母通过以下机床数据设置：

MD31050 \$MA_DRIVE_AX_RATIO_DENOM[n]（负载齿轮比分母）

MD31060 \$MA_DRIVE_AX_RATIO_NUMERA[n]（负载齿轮比分子）

正负系数的设置范围一样大。

不可输入“0”值。

说明

若配置了一个间接编码器，且负载齿轮比系数变更，那么参考将丢失且 NC/PLC 接口信号：

<Axis>.basic.in.enc1Synchronized /

<Axis>.basic.in.enc2Synchronized（已回参考点/已同步 1 或 2）

针对相关测量系统复位。

更多信息

闭环控制和伺服参数组的更多信息参见：

- 速度、设定值/实际值系统、闭环控制 (页 141)
- NC 编程手册

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.posCtrlParameterSetSelection	LBP_Axis*.A_Para_A..C	DB31, ...DBX9.0..2
<Axis>.spindle.out.gearStageSelection	LBP_Axis*.A_ActGearA..C	DB31, ...DBX16.0..2
<Axis>.spindle.out.gearChangeoverDone	LBP_Axis*.A_GearChangeOv	DB31, ...DBX16.3

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.enc1Synchronized	LBP_Axis*.E_RefSyn1	DB31, ...DBX60.4
<Axis>.basic.in.enc2Synchronized	LBP_Axis*.E_RefSyn2	DB31, ...DBX60.5
<Axis>.drive.in.posCtrlParameterSetNumber	LBP_Axis*.E_ParS_A	DB31, ...DBX69.0..2

13.4.5 中间齿轮比

应用和功能

通过可配置的中间齿轮比可调整各种动力刀具。刀具侧的中间齿轮比以乘法作用于电机齿轮比/负载齿轮比。

中间齿轮比通过以下机床数据设置：

MD31066 \$MA_DRIVE_AX_RATIO2_NUMERA（中间齿轮比分子）

MD31064 \$MA_DRIVE_AX_RATIO2_DENOM（中间齿轮比分母）

- 用于中间齿轮比的刀具侧编码器通过以下机床数据配置：
MD31044 \$MA_ENC_IS_DIRECT2（中间齿轮比上的编码器）。

该机床数据被修改后，可通过调试软件 SINUMERIK Commissioning Tool 使用“激活机床数据”功能来激活经修改的参数设置。现有的电机齿轮比/负载齿轮箱则是上电后生效。

换刀

若需在换刀时一并切换中间齿轮比，那么用户必须通过中间齿轮比的机床数据重新配置其分母与分子的传动比。

示例：

 小心

配置故障
用户须负责在适宜的时间段停止、立即执行机床数据修改并通过“激活机床数据”功能激活。

为换入的传动比为 2:1 的刀具配置一个匹配的中间齿轮比，并在零件程序中通过 NEWCONF 指令立即激活。

程序代码
N05 \$MA_DRIVE_AX_RATIO2-NUMERA[AX5] = 2
M10 \$MA_DRIVE_AX_RATIO2-DENOM[AX5] = 1
N15 NEWCONF

切换

借助“激活机床数据”功能立即切换至新的传动比。技术上而言，对应的机械切换要持续一些时间，因为要通过动力刀具换入另一个辅助刀头。

说明

在静止状态下切换无急动。因此用户须采取相应的预防措施。

对于运动中需要进行切换，以及需要平滑或柔和转速过渡的应用，可通过驱动端现有的转速设定值滤波器应对。

对控制技术关联的更多说明请见“速度、设定值/实际值系统、闭环控制 (页 141)”。

13.4.6 未应答的齿轮档切换

运行方式切换

未应答的齿轮档切换无法通过运行方式切换（例如切换至 JOG）中断。

切换最多可延迟以下机床数据中输入的时间值：

MD10192 \$MN_GEAR_CHANGE_WAIT_TIME

若未在此时间内应答齿轮档切换，那么 NC 将触发报警。

其他事件

触发重整的事件同样会等待齿轮档切换结束。

机床数据：

MD10192 \$MN_GEAR_CHANGE_WAIT_TIME

用于设置对齿轮档切换的等待时间。

此时间结束后，若齿轮档切换仍未结束，那么 NC 将响应并发出报警。

以下事件具有类似的特性：

- 用户 ASUB
- 模式切换
- 删除剩余行程
- 跨通道取轴
- 激活 PI 用户数据

13.4 可配置的齿轮档调整

- 激活 PI 服务机床数据
- 切换跳转，切换 DryRun
- 在运行方式下编辑
- 补偿组报警
- 溢出转存
- G33、G34、G35 下的快速退刀
- 子程序级终止，子程序终止

上电后特性

上电后，在 NC 复位状态下可通过 PLC 控制系统设定机床上生效的齿轮档。

NC 会选择对应的参数组
并通过 NC/PLC 接口信号：
<Axis>.spindle.in.requestedGearStageNumber（设定齿轮档 A 至 C）
向 PLC 反馈。

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.spindle.in.requestedGearStageNumber	LBP_Axis*.E_SetpGearA..C	DB31, ...DBX82.0.. 2

13.4.7 通过往复模式进行齿轮档切换

在通过 M40（自动选择齿轮档）或 M41 至 M45 设定了新的齿轮档时，主轴处于往复模式中。

在往复运行中，主轴电机的旋转方向以较小的时间间隔交替反转。借助电机的往复运动，例如能够更顺利地切换至新的齿轮档。原则上不使用往复运行也能切换至新的齿轮档。

可采用下列往复类型：

- 通过 NC 往复
- 通过 PLC 往复
- 使用 FC 18 往复

更多信息

PLC 功能手册

调试：机床数据

- MD35010 \$MA_GEAR_STEP_CHANGE_ENABLE（参数设置齿轮档切换）
- MD35410 \$MA_SPIND_OSCILL_ACCEL（往复时的加速度）
- MD35430 \$MA_SPIND_OSCILL_START_DIR（往复时的起始方向）
- MD35400 \$MA_SPIND_OSCILL_DES_VELO（往复转速）
- MD35450 \$MA_SPIND_OSCILL_TIME_CCW（M4 方向的往复时间）
- MD35440 \$MA_SPIND_OSCILL_TIME_CW（M3 方向的往复时间）

调试：NC/PLC 接口信号

- <Axis>.spindle.out.gearStageSelection（实际齿轮档）
- <Axis>.spindle.out.gearChangeoverDone（齿轮箱已换档）
- <Axis>.spindle.out.oscillRequest（通过 PLC 往复）
- <Axis>.spindle.out.oscillEnable（往复使能）
- <Axis>.spindle.out.oscillSetDirCw（设定顺时针旋转方向）
- <Axis>.spindle.out.oscillSetDirCcw（设定逆时针旋转方向）
- <Axis>.basic.in.isStationary（主轴停止）
- <Axis>.spindle.in.requestedGearStageNumber（设定齿轮档）
- <Axis>.spindle.in.gearChangeoverRequested（齿轮箱换档）
- <Axis>.spindle.in.speedSetpointReached（主轴位于设定范围内）
- <Axis>.spindle.in.oscillModeActive（有效主轴运行方式：往复模式）

过程说明：齿轮档切换

参数设置：MD35010 \$MA_GEAR_STEP_CHANGE_ENABLE，位 0 = 1

13.4 可配置的齿轮档调整

过程:

- 主轴制动。
此时制动过程对应 M5 运动。
- NC/PLC 接口信号 NC → PLC:
 - <Axis>.spindle.in.oscillModeActive == 1 (往复模式)
 - <Axis>.spindle.in.gearChangeoverRequested == 1 (齿轮箱换档)
 - <Axis>.spindle.in.requestedGearStageNumber == <设定齿轮档>
 - <Axis>.basic.in.posCtrlActive == 0 (位置调节器生效)
- 现在可“脱接”负载齿轮箱。
- 现在可通过 NC 或 PLC 执行往复运行 (见下文)。
- 在 PLC 向 NC 反馈“齿轮档切换”和“新齿轮档生效”后, 继续执行程序:
- <Axis>.spindle.out.gearChangeoverDone = 1 (齿轮箱已换档)
- <Axis>.spindle.out.gearStageSelection (实际齿轮档)
- 对于采用间接编码器 (电机编码器) 的测量系统, 回参考点状态会被删除:


```
<Axis>.basic.in.enc1Synchronized /
<Axis>.basic.in.enc2Synchronized = 0
```

通过 NC 往复

请求

- <Axis>.spindle.out.oscillEnable = 1 (往复使能)

状态反馈信息

- <Axis>.spindle.in.oscillModeActive == 1 (往复模式)
- <Axis>.spindle.in.gearChangeoverRequested == 1 (齿轮箱换档)
- <Axis>.spindle.in.requestedGearStageNumber == <设定齿轮档>
- <Axis>.basic.in.posCtrlActive == 0 (位置调节器生效)

现在可“脱接”负载齿轮箱。

启动往复

现在可通过 PLC 用户程序将往复使能置位:

- <Axis>.spindle.out.oscillEnable = 1 (往复使能)

在取决于旋转方向的往复时间内，电机以起始旋转方向以往复加速度加速至往复转速：

- MD35410 \$MA_SPIND_OSCILL_ACCEL（往复时的加速度）
- MD35430 \$MA_SPIND_OSCILL_START_DIR（往复时的起始方向）
- MD35400 \$MA_SPIND_OSCILL_DES_VELO（往复转速）
- MD35450 \$MA_SPIND_OSCILL_TIME_CCW（M4 方向的往复时间）
- MD35440 \$MA_SPIND_OSCILL_TIME_CW（M3 方向的往复时间）

切换旋转方向

在取决于旋转方向的往复时间届满后，电机以往复加速度制动，并在另一往复时间内以相反的旋转方向加速至往复转速。

通过 PLC 往复

请求

- <Axis>.spindle.out.oscillRequest = 1（通过 PLC 往复）
- <Axis>.spindle.out.oscillEnable = 1（往复使能）

状态反馈信息

- <Axis>.spindle.in.oscillModeActive == 1（往复模式）
- <Axis>.spindle.in.gearChangeoverRequested == 1（齿轮箱换档）
- <Axis>.spindle.in.requestedGearStageNumber == <设定齿轮档>
- <Axis>.basic.in.posCtrlActive == 0（位置调节器生效）

现在可“脱接”负载齿轮箱。

启动往复

现在可通过 PLC 用户程序将“往复使能”置位：

- <Axis>.spindle.out.oscillSetDirCcw = 1（设定逆时针旋转方向）
或<Axis>.spindle.out.oscillSetDirCw = 1（设定顺时针旋转方向）

电机以设定旋转方向以往复加速度加速至往复转速：

- MD35410 \$MA_SPIND_OSCILL_ACCEL（往复时的加速度）
- MD35400 \$MA_SPIND_OSCILL_DES_VELO（往复转速）

切换旋转方向

往复时间以及旋转方向切换均需要在 PLC 用户程序中实现。

往复模式结束

往复模式结束后，主轴重新处于控制模式，并自动进入通过 SPCON 或 SPCOF 定义的模式。
所有针对齿轮档的限值（齿轮档的转速上限/下限等）对应设定的实际齿轮档的值。

程序段转换

若将主轴切换至往复模式（<Axis>.spindle.in.gearChangeoverRequested = 1（齿轮箱换档）），则 NC 程序的执行将会停止。
在齿轮档切换完成后（<Axis>.spindle.out.gearChangeoverDone == 1（齿轮箱已换档）），系统会继续执行该 NC 程序。

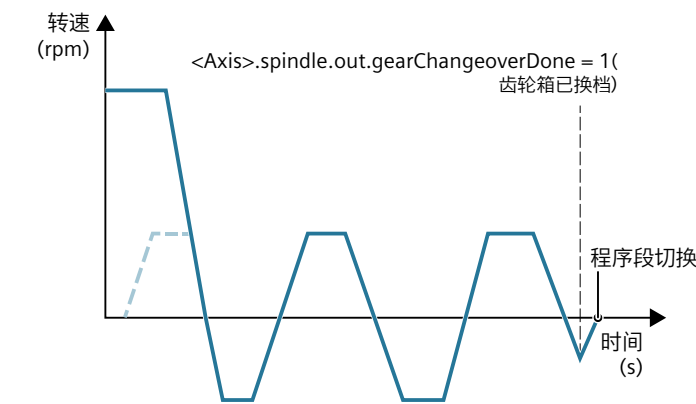
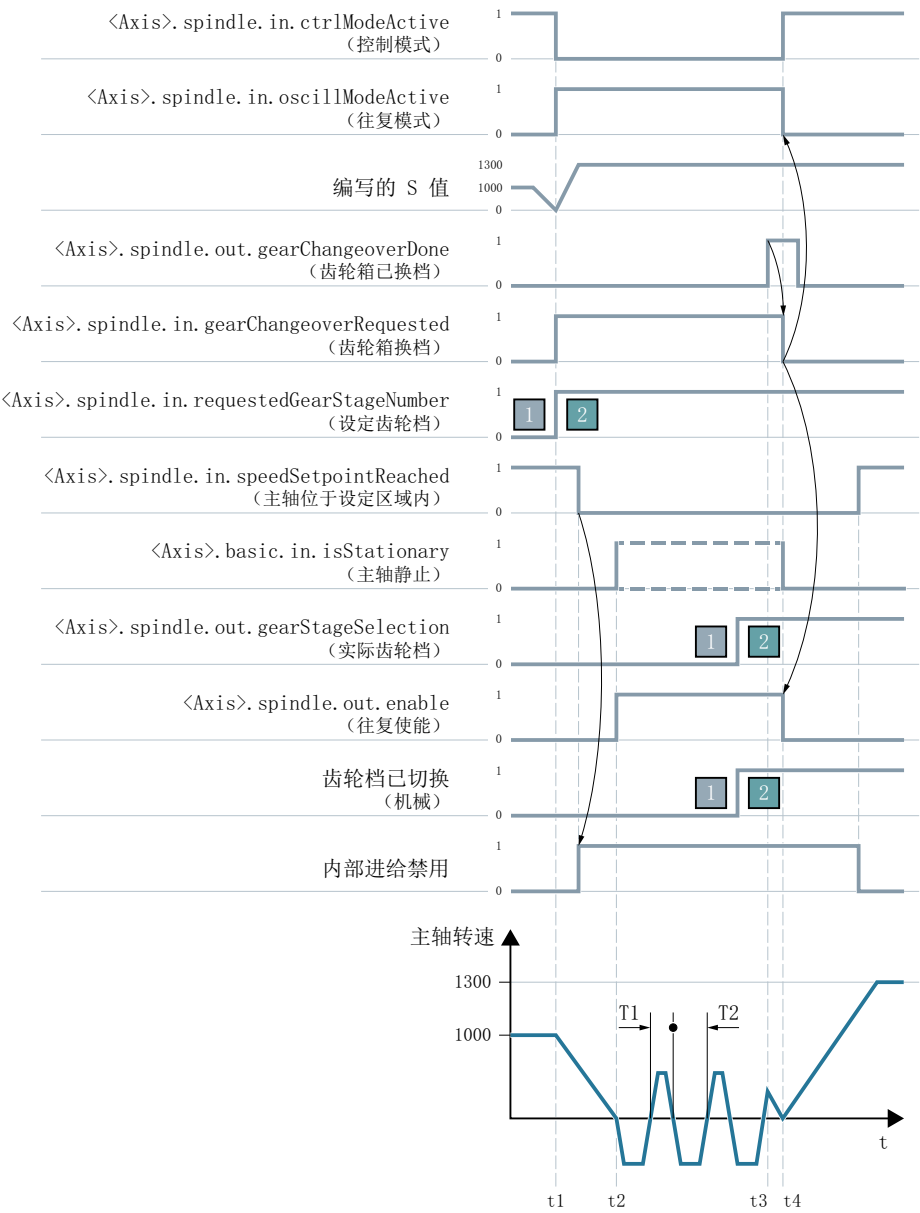


图 13-5 往复模式后的程序段切换

示例性信号图



t_1 : 由于 NC 程序中新编程的 S 值 S1300，NC 检测到需要切换齿轮档（第 1 → 第 2 齿轮档）并请求 PLC 使能：

- NC → PLC: `<Axis>.spindle.in.gearChangeoverRequested = 1`（齿轮箱换档）

NC 将程序执行禁止（内部进给禁止*）

13.4 可配置的齿轮档调整

t_2 : NC 向 PLC 发送**主轴转速 == 0** 信息:

- `<Axis>.basic.in.isStationary = 1` (主轴停止)

PLC 用户程序给予往复使能。

- `PLC → NC: <Axis>.spindle.out.oscillEnable = 1` (往复使能)

提示

往复使能最迟须在时间点 t_2 置位。

NC 开始往复。

t_3 : 在机床上执行齿轮档切换。

PLC 用户程序将切换信息反馈给 NC，并删除往复使能:

- `PLC → NC: <Axis>.spindle.out.gearStageSelection = 2` (实际齿轮档)
- `PLC → NC: <Axis>.spindle.out.gearChangeoverDone = 1` (齿轮箱已换档)
- `PLC → NC: <Axis>.spindle.out.oscillEnable = 0` (往复使能)

t_4 : NC 通过将对 PLC 的请求复位结束齿轮档切换，并通过将主轴运行模式从往复模式切换至控制模式来结束往复:

- `NC → PLC: <Axis>.spindle.in.gearChangeoverRequested = 0` (齿轮箱换档)
- `<Axis>.spindle.in.oscillModeActive = 0` (有效主轴运行方式: 往复模式)
- `<Axis>.spindle.in.ctrlModeActive = 1` (有效主轴运行方式: 控制模式)

NC 使能下一程序段用于加工，并将主轴加速至新的 S 值 S1300。

*: 以下情形下，内部进给禁止:

- 因在 NC 程序中编写 S 值而发生齿轮档切换**并且**
- **加工程序段生效** (非快速移动 G0)

提示

通过同步动作进行齿轮档切换，或通过 PLC 使用 FC 18 设定时，内部进给禁止**不会**置位。

图 13-6 静止主轴上的齿轮档切换

边界条件

- 为进行主轴制动，不必将以下 NC/PLC 接口信号置位:
`<Axis>.basic.out.feedStopSpindleStop` (主轴停止)
请求齿轮档切换时，控制系统内部使得主轴静止。
- 总是通过以下 NC/PLC 接口信号的置位来结束齿轮档切换:
`<Axis>.spindle.out.gearChangeoverDone = 1` (齿轮箱已换档)

- 通过以下 C/PLC 接口信号的复位来停止 NC 中的往复运动：
`<Axis>.spindle.out.oscillEnable`（往复使能）
 但主轴处于“往复模式”中。
- 在采用齿轮档切换和间接测量系统（MD31050 \$MA_ENC_IS_DIRECT = 0 电机编码器）时，主轴同步会丢失。
 齿轮箱换档 ⇒ `<Axis>.basic.in.enc1Synchronized` /
`<Axis>.basic.in.enc2Synchronized = 0`（已回参考点/已同步）
 切换齿轮档后，在下次越过零脉冲时重新对主轴进行同步。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<code><Axis>.basic.out.feedStopSpindleStop</code>	<code>LBP_Axis*.A_FDSpStop</code>	DB31, ...DBX4.3
<code><Chan>.basic.out.disableReadIn</code>	<code>LBP_Chan*.A_Rldisable</code>	DB21, ...DBX6.1
<code><Chan>.basic.out.ncStopAxesAndSpindles</code>	<code>LBP_Chan*.A_NCStopASp</code>	DB21, ...DBX7.4
<code><Axis>.spindle.out.gearStageSelection</code>	<code>LBP_Axis*.A_ActGearA..C</code>	DB31, ...DBX16.0.. 2
<code><Axis>.spindle.out.gearChangeoverDone</code>	<code>LBP_Axis*.A_GearChangeOv</code>	DB31, ...DBX16.3
<code><Axis>.spindle.out.oscillRequest</code>	<code>LBP_Axis*.A_OscilPLC</code>	DB31, ...DBX18.4
<code><Axis>.spindle.out.oscillEnable</code>	<code>LBP_Axis*.A_OscilSpeed</code>	DB31, ...DBX18.5
<code><Axis>.spindle.out.oscillSetDirCw</code>	<code>LBP_Axis*.A_SetRotDirectRight</code>	DB31, ...DBX18.6
<code><Axis>.spindle.out.oscillSetDirCcw</code>	<code>LBP_Axis*.A_SetRotDirectLeft</code>	DB31, ...DBX18.7

NC → PLC

Basic Program Plus	Basic Program	
<code><Axis>.basic.in.enc1Synchronized</code>	<code>LBP_Axis*.E_RefSyn1</code>	DB31, ...DBX60.4
<code><Axis>.basic.in.enc2Synchronized</code>	<code>LBP_Axis*.E_RefSyn2</code>	DB31, ...DBX60.5
<code><Axis>.basic.in.isStationary</code>	<code>LBP_Axis*.E_Stat</code>	DB31, ...DBX61.4
<code><Axis>.basic.in.posCtrlActive</code>	<code>LBP_Axis*.E_PositContr</code>	DB31, ...DBX61.5
<code><Axis>.spindle.in.requestedGearStageNumber</code>	<code>LBP_Axis*.E_SetpGearA..C</code>	DB31, ...DBX82.0.. 2
<code><Axis>.spindle.in.gearChangeoverRequested</code>	<code>LBP_Axis*.E_GearChange</code>	DB31, ...DBX82.3
<code><Axis>.spindle.in.speedSetpointAboveMax</code>	<code>LBP_Axis*.E_SetSpeedLimit</code>	DB31, ...DBX83.1

13.4 可配置的齿轮档调整

Basic Program Plus	Basic Program	
<Axis>.spindle.in.speedSetpointBelowMin	LBP_Axis*.E_SetSpeedIncreased	DB31, ...DBX83.2
<Axis>.spindle.in.speedSetpointReached	LBP_Axis*.E_SetRange	DB31, ...DBX83.5
<Axis>.spindle.in.oscillModeActive	LBP_Axis*.E_OscillMode	DB31, ...DBX84.6
<Axis>.spindle.in.ctrlModeActive	LBP_Axis*.E_ContrMode	DB31, ...DBX84.7

13.4.8 在固定位置切换齿轮档

应用和优点

标准化主轴齿轮越来越多地应用于机床，一方面是为了减少齿轮档切换时的技术时滞，另一方面则是为了享受使用标准化机械组件所带来的成本优势。

固定位置上的齿轮档切换功能能够支持对负载齿轮箱的“定向齿轮档切换”，并会通过 NC 请求另一种控制方案。此时齿轮档切换只能在一个定义的主轴位置上执行。传统负载齿轮箱所启用的往复运动将不再需要。

在固定位置切换齿轮档的过程

在固定位置切换齿轮档

机床数据配置：

MD35010 \$MA_GEAR_STEP_CHANGE_ENABLE = 2

下的后续过程：

- 主轴从静止状态或运动状态定位至以下机床数据中配置的位置：

MD35012 \$MA_GEAR_STEP_CHANGE_POSITION

若从运动状态开始执行齿轮档切换，则保持当前的旋转方向。定位期间主轴处于定位模式下。

以下 NC/PLC 接口信号会输出：

<Axis>.spindle.in.posModeActive（定位模式）

。

若未回参考点：

<Axis>.basic.in.enc1Synchronized/

<Axis>.basic.in.enc2Synchronized = 0

或以下 NC/PLC 接口信号置位：

<Axis>.spindle.out.resyncEnc1ForPos/

<Axis>.spindle.out.resyncEnc2ForPos（定位时测量系统 1/2 重新同步）

则定位将延长搜索零脉冲的过程。

- 到达以下机床数据中配置的换刀位置后：

MD35012 \$MA_GEAR_STEP_CHANGE_POSITION

在切换至往复模式前，等待以下机床数据中设定的时间：

MD35310 \$MA_SPIND_POSIT_DELAY_TIME

并开始齿轮档切换对话。

- 输出 NC/PLC 接口信号：

<Axis>.spindle.in.posModeActive（往复模式）

<Axis>.spindle.in.gearChangeoverRequested（齿轮箱换档）

<Axis>.spindle.in.requestedGearStageNumber（设定齿轮档 A 至 C）

- 使用间接编码器（电机编码器）的测量系统生效时，位置闭环控制不取消：

MD31040 \$MA_ENC_IS_DIRECT = 0

使用直接编码器（负载编码器）的测量系统生效时，取消位置闭环控制：

<Axis>.basic.in.posCtrlActive = 0，

这是因为至负载的力流中断，无法通过其实现位置闭环控制。

- 若无法启用位置闭环控制，可通过

复位“伺服使能”将其取消：

<Axis>.basic.out.enableCtrl = 0

。

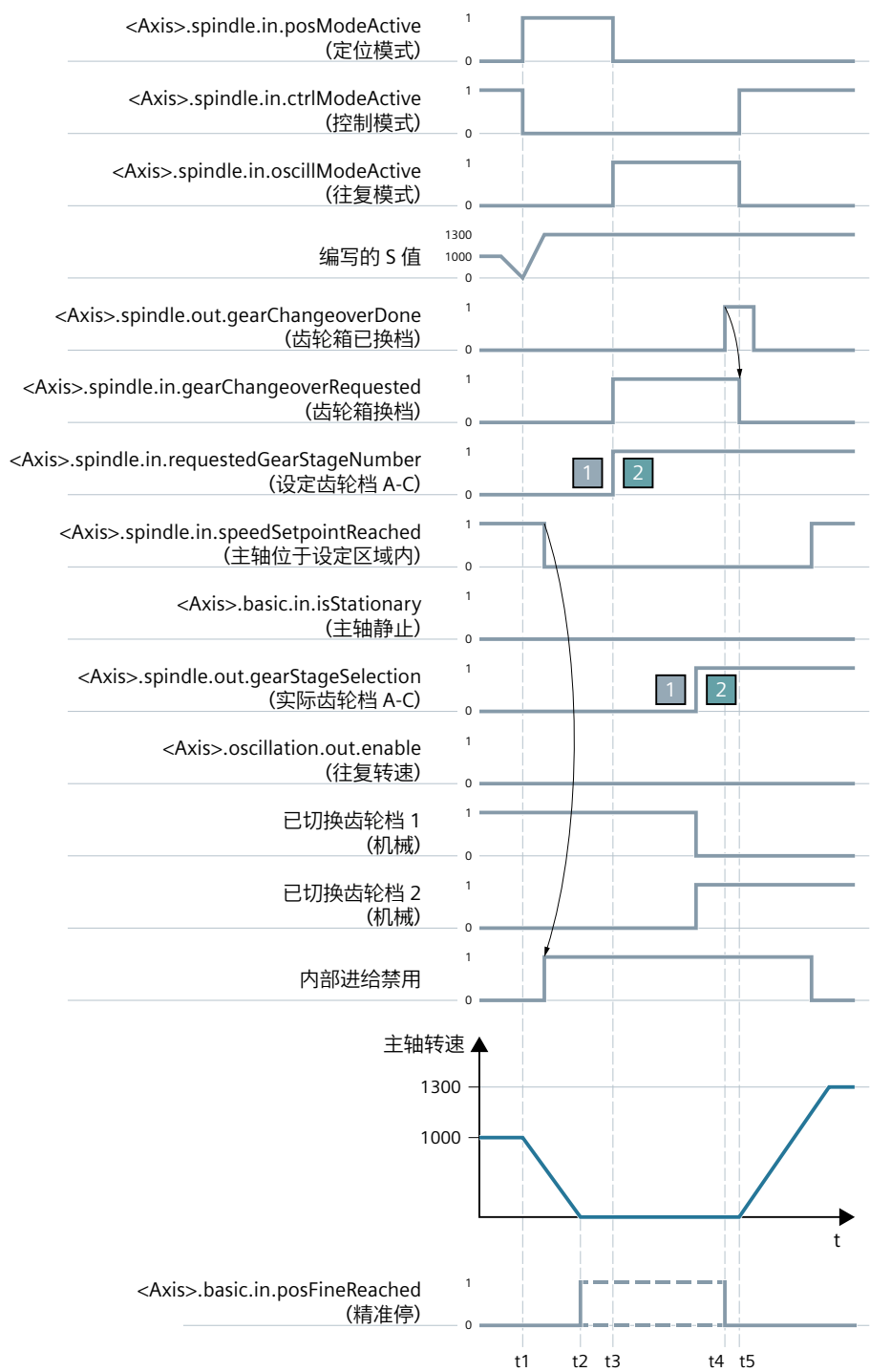
13.4 可配置的齿轮档调整

- 机床上齿轮档的机械切换
为此不需要电机的往复运行。
NC/PLC 接口信号：
`<Axis>.spindle.out.oscillEnable`（往复使能）
和
`<Axis>.spindle.out.oscillRequest`（通过 PLC 往复）
不用置位。
原则上现在还是可以启用往复运行。
- 通过 PLC 写入 NC/PLC 接口信号：
`<Axis>.spindle.out.gearStageSelection`（实际齿轮档 A 至 C）
。
- 发送以下信号后：
`<Axis>.spindle.out.gearChangeoverDone`（齿轮箱已换档）
若存在最后生效的运动，则会继续执行。
间接编码器（电机编码器）的回参考点状态会被删除：
`<Axis>.basic.in.enc1Synchronized/`
`<Axis>.basic.in.enc2Synchronized = 0。`
主轴处于转速控制运行下，且输出 NC/PLC 接口信号：
`<Axis>.spindle.in.ctrlModeActive`（控制模式）
。

齿轮档切换规定位置

在固定位置切换齿轮档的典型时序过程：

13.4 可配置的齿轮档调整



- t_1 : 通过编程 S1300, NC 识别一个新的齿轮档 (齿轮档 2), 置位接口信号 `<Axis>.spindle.in.posModeActive` (定位模式) 并禁止下一个零件程序段的执行 (= 内部进给禁止*)。
- t_2 : 主轴停止, 并发出准停提示信号。

- t_3 : 齿轮档切换等待时间
- t_4 : 新齿轮档接合。PLC 用户将新的（实际）齿轮档传输至 NC，并置位接口信号 `<Axis>.spindle.out.gearChangeoverDone`（齿轮箱已换档）。
- t_5 : NC 复位接口信号 `<Axis>.spindle.in.gearChangeoverRequested`（齿轮箱换档），使能下一个零件程序段用于加工，并将主轴加速至新的 S 值 (S1300)。
- *: 以下情形下，内部进给禁止：
- 主轴齿轮档切换通过零件程序编写并且
 - 一个加工程序段生效（即 G0 不生效）
- 通过同步动作进行齿轮档切换，或通过 PLC 使用 FC 18 设定时，内部进给禁止不会置位。

图 13-7 静止主轴上的齿轮档切换

齿轮档切换位置 MD35012

齿轮档切换位置在以下机床数据中针对每个齿轮档配置：

MD35012 \$MA_GEAR_STEP_CHANGE_POSITION

齿轮档切换等待时间 MD35310

定位后，等待以下机床数据中设定的时间：

MD35310 \$MA_SPIND_POSIT_DELAY_TIME

直至接收到齿轮箱换档请求：

`<Axis>.spindle.in.oscillModeActive`（往复模式）

`<Axis>.spindle.in.gearChangeoverRequested`（齿轮箱换档）

和

`<Axis>.spindle.in.requestedGearStageNumber`（设定齿轮档 A 至 C）

。

定位特性值/位置

主轴总是沿最短路径（等同于 DC）逼近齿轮档位置。

若未执行回参考点，且主轴处于静止状态（例如上电后），

那么逼近方向取决于以下机床数据：

MD35350 \$MA_SPIND_POSITIONING_DIR

若需启用可变的齿轮档切换位置，可通过写入机床数据并于随后通过“激活机床数据”激活修改来实现。

机床数据的修改可通过零件程序进行。

13.4 可配置的齿轮档调整

若未到达配置的位置，则会触发报警 22020，且不会发生 NC 和 PLC 间的齿轮档切换信号对话，防止齿轮箱损坏。此报警具有很高权重，因此无法继续执行零件程序，必须消除故障原因。根据经验，定位终止的原因可能是机床数据设置错误，或 PLC 信号不兼容。

速度

定位速度在以下机床数据中针对齿轮档配置：

MD35300 \$MA_SPIND_POSCTRL_VELO

NC/PLC 接口信号“主轴倍率/进给倍率”：

<Axis>.spindle.out.useFeedrateOvr = 0:

<Axis>.spindle.out.ovrFactor

以及：

<Axis>.spindle.out.useFeedrateOvr = 1:

<Axis>.basic.out.feedrateOvrFactor

在定位中照常生效。

定位速度可通过程序指令 OVRA[Sn] 按百分比修改。

说明

OVRA[Sn] 为模态生效。在齿轮档切换结束后，应将其设置为适宜用于加工的值。

零件程序指令 FA[Sn] 不会影响齿轮档切换中的定位速度。

加速度

加速度值针对齿轮档在以下机床数据中定义：

MD35200 \$MA_GEAR_STEP_SPEEDCTRL_ACCEL

与

MD35210 \$MA_GEAR_STEP_POSCTRL_ACCEL

加速度可通过程序指令 ACC[Sn] 按比例修改。

说明

ACC[Sn] 为模态生效。在齿轮档切换结束后，应将其设置为适宜用于加工的值。

取决于转速的加速度

“折线式加速度特性曲线”与使用 SPOS 或 SINU_PositionSpindle (Basic Program - FC18) 定位时一样是生效的。

急动度

目前尚无法限制加速度变化。

定位结束

定位结束 (<Axis>.spindle.in.posModeActive)

和往复模式开始 (<Axis>.spindle.in.oscillModeActive) 之间的过渡取决于到达“精准停” (<Axis>.basic.in.posFineReached) 以及以下机床数据中输入的时间：

MD3510 \$MA_SPIND_POSIT_DELAY_TIME

。

过渡条件的定义一方面会影响齿轮档切换时间，另一方面也会影响到达齿轮档切换位置时的精度。

程序段转换

程序段转换将停止，直至 PLC 发出齿轮档切换结束信号 (<Axis>.spindle.out.gearChangeoverDone)。

齿轮档切换结束

齿轮档切换结束后，主轴重新回到控制模式，并自动进入通过 SPCON 或 SPCOF 定义的闭环控制模式。

所有针对齿轮档的限值（齿轮档的转速上限/下限等）对应反馈的实际齿轮档。

边界条件

- 主轴至少须配备一个测量系统。
- 必须能启用位置闭环控制运行，并对其进行过调试。
- 原则上，通过零件程序、同步动作或 SINU_PositionSpindle (Basic Program - FC18): "execute" 调用的 SPOS 必须能无故障执行。

若无法满足所有前提条件，此功能可能无法成功应用。

激活

“在固定位置切换齿轮档”功能通过以下机床数据设置激活：

MD35010 \$MA_GEAR_STEP_CHANGE_ENABLE = 2

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.feedrateOvrFactor	LBP_Axis*.A_FD_OR	DB31, ...DBB0
<Axis>.basic.out.enableCtrl	LBP_Axis*.A_ContrEnable	DB31, ...DBX2.1
<Chan>.basic.out.ncStopAxesAndSpindles	LBP_Chan*.A_NCStopASp	DB21, ...DBX7.4
<Axis>.spindle.out.gearStageSelection	LBP_Axis*.A_ActGearA..C	DB31, ...DBX16.0.. 2
<Axis>.spindle.out.gearChangeoverDone	LBP_Axis*.A_GearChangeOv	DB31, ...DBX16.3
<Axis>.spindle.out.useFeedrateOvr	LBP_Axis*.A_FD_ORSp	DB31, ...DBX17.0
<Axis>.spindle.out.resyncEnc1ForPos	LBP_Axis*.A_Sp1SynP	DB31, ...DBX17.4
<Axis>.spindle.out.resyncEnc2ForPos	LBP_Axis*.A_Sp2SynP	DB31, ...DBX17.5
<Axis>.spindle.out.oscillRequest	LBP_Axis*.A_OscilPLC	DB31, ...DBX18.4
<Axis>.spindle.out.oscillEnable	LBP_Axis*.A_OscilSpeed	DB31, ...DBX18.5
<Axis>.spindle.out.oscillSetDirCw	LBP_Axis*.A_SetRotDirectRight	DB31, ...DBX18.6
<Axis>.spindle.out.oscillSetDirCcw	LBP_Axis*.A_SetRotDirectLeft	DB31, ...DBX18.7
<Axis>.spindle.out.ovrFactor	LBP_Axis*.A_SpOR	DB31, ...DBB19

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.enc1Synchronized	LBP_Axis*.E_RefSyn1	DB31, ...DBX60.4
<Axis>.basic.in.enc2Synchronized	LBP_Axis*.E_RefSyn2	DB31, ...DBX60.5
<Axis>.basic.in.posFineReached	LBP_Axis*.E_ExactFine	DB31, ...DBX60.7
<Axis>.basic.in.isStationary	LBP_Axis*.E_Stat	DB31, ...DBX61.4
<Axis>.basic.in.posCtrlActive	LBP_Axis*.E_PositContr	DB31, ...DBX61.5

Basic Program Plus	Basic Program	
<Axis>.spindle.in.requestedGearStageNumber	LBP_Axis*.E_SetpGearA..C	DB31, ...DBX82.0-82.2
<Axis>.spindle.in.gearChangeoverRequested	LBP_Axis*.E_GearChange	DB31, ...DBX82.3
<Axis>.spindle.in.posModeActive	LBP_Axis*.E_PosMode	DB31, ...DBX84.5
<Axis>.spindle.in.oscillModeActive	LBP_Axis*.E_OscillMode	DB31, ...DBX84.6
<Axis>.spindle.in.ctrlModeActive	LBP_Axis*.E_ContrMode	DB31, ...DBX84.7

13.4.9 执行 M70 时启用的齿轮档

技术背景

在某些机床上，可能需要在轴模式期间启用特定齿轮档。

可能的原因包括：

- 在轴模式的伺服参数组中（索引 0），对应一个齿轮档只匹配了一项优化（ K_v 、前馈控制、滤波器）。同时这些参数组的机床数据不能修改。
- 只有一个机械传动比，和其他齿轮档相比只有较小或没有反向间隙。只有启用该齿轮档时，主轴才能同其他轴一起实现轨迹运动或转换（例如 TRANSMIT）。

功能

此功能激活时，过渡至轴模式时将自动切换为预设的齿轮档。

齿轮档切换集成在 M70 进程中，并在主轴制动后和换入伺服参数组前通过索引 0 的参数组进行（注意 MD35590 \$MA_PARAMSET_CHANGE_ENABLE 的设置！）

启用此功能时，齿轮档切换期间 NC 和 PLC 的信号对话的执行方式同编写齿轮档切换（M41 ... M45）时相同。

前提条件

过渡至轴模式时切换齿轮档的前提条件是对齿轮档切换的总体使能：

MD35010 \$MA_GEAR_STEP_CHANGE_ENABLE（参数设置齿轮档切换）

MD35090 \$MA_NUM_GEAR_STEPS（建立的齿轮档的数量）

13.4 可配置的齿轮档调整

激活/取消激活

此功能通过以下机床数据激活/取消：
MD35014 \$MA_GEAR_STEP_USED_IN_AXISMODE（M70 下轴模式的齿轮档）

值	含义
0	执行 M70 时不进行隐性齿轮档切换。保留当前齿轮档（缺省设置！）
1 ... 5	执行 M70 时切换至齿轮档 1 ... 5。

前提条件

在固定位置切换齿轮档（MD35010 \$MA_GEAR_STEP_CHANGE_ENABLE = 2）
此功能支持“在固定位置切换齿轮档”。此时，执行 M70 时将增加主轴定位的过程。逼近齿轮档切换位置时采用当前齿轮档。
在未编写 M70 的情形下过渡至轴模式
控制系统能够基于编程顺序自动识别出向轴模式的过渡（参见“隐性过渡至进给轴模式 (页 425)”），并于系统内部生成必要的包含齿轮档切换的 M70 过程。

通过 FC 18 过渡至轴模式
通过 FC 18 过渡至轴模式（“启动轴”）时，不支持隐性齿轮档切换。此时必须通过 PLC 应用程序在切换至轴模式前换入匹配的齿轮档。也可通过 FC 18 进行齿轮档切换（“启动齿轮档切换”）。

从轴模式切换至主轴模式
在从轴模式切换至主轴模式时，通过 M70 换入的齿轮档保持生效。主轴模式下在 M70 前生效的齿轮档不会被自动重新换入。伺服参数组会从参数组 1（索引 0）切换至与齿轮档相匹配的参数组 2 ... 6（索引 1 ... 5）（MD35590 \$MA_PARAMSET_CHANGE_ENABLE < 2 时）。

示例

主轴过渡至轴模式时需要换入齿轮档 4。
配置：MD35014 \$MA_GEAR_STEP_USED_IN_AXISMODE[<主轴名称>] = 4

程序代码	注释
N05 M3 S1000	
N10 G1 X100 F1000	
N15 M70	； 换入齿轮档 4。

程序代码	注释
N20 POS[C]=77	
N25 ...	

说明

MD35014 可通过“激活机床数据”功能改变。这样一来，在过渡至轴模式前可根据需要在零件程序修改要换入的齿轮档。

13.4.10 在 DryRun、程序测试和 SERUPRO 下抑制齿轮档切换

功能

在空运行进给（DryRun）、程序测试和 SERUPRO 模式下通常不需要进行齿轮档切换。因此可为这些模式抑制该功能。此配置通过以下机床数据的位 0 ... 2 进行：

MD35035 \$MA_SPIND_FUNCTION_MASK

空运行进给（DryRun）	
位 0 = 0	DryRun 功能激活时，为含 M40、M41 至 M45 的零件程序段、SINU_RotateSpindle (Basic Program FC18) 编程和同步动作启用齿轮档切换。
位 0 = 1	DryRun 功能激活时，为含 M40、M41 至 M45 的零件程序段、SINU_RotateSpindle (Basic Program FC18) 编程和同步动作抑制齿轮档切换。
程序测试和 SERUPRO	
位 1 = 0	程序测试/SERUPRO 功能激活时，为含 M40、M41 至 M45 的零件程序段、SINU_RotateSpindle (Basic Program FC18) 编程和同步动作启用齿轮档切换。
位 1 = 1	程序测试/SERUPRO 功能激活时，为含 M40、M41 至 M45 的零件程序段、SINU_RotateSpindle (Basic Program FC18) 编程和同步动作抑制齿轮档切换。
DryRun、程序测试和 SERUPRO	

13.4 可配置的齿轮档调整

位 2 = 0	在 DryRun、程序测试和 SERUPRO 功能取消后，在 REPOS 中不针对编写齿轮档追加执行齿轮档切换。
位 2 = 1	在 DryRun、程序测试和 SERUPRO 功能取消后，在条件允许时针对编写齿轮档追加执行齿轮档切换。

过程

齿轮档切换被抑制时，编写的主轴转速同样会由插补器限制在生效的齿轮档所允许的范围内。

此类限制中生成的 NC/PLC 接口信号

<Axis>.spindle.in.gearChangeoverRequested（设定转速已提升）和

<Axis>.spindle.in.speedSetpointAboveMax（设定转速已限制）将被抑制。

DryRun 和空运行进给中不必通过 PLC 程序进行监控。

抑制齿轮档切换时，新的设定齿轮档

(<Axis>.spindle.in.requestedGearStageNumber) 不会输出至 PLC。

齿轮档切换请求 <Axis>.spindle.in.gearChangeoverRequested（齿轮箱换挡）也会被抑制。

这样可防止 PLC 程序处理齿轮档切换信息。

确定最后生效的齿轮档

系统变量 \$P_GEAR 提供的是通过零件程序编写的齿轮档（可能未输出至 PLC）。

系统变量 \$AC_SGEAR 提供的是零件程序、同步动作中最后生效的齿轮档，并可在操作界面上读取。

取消时的特性

DryRun 功能可在程序段运行期间取消。该功能取消后，必须确定零件程序所请求的齿轮档并将其换入。

只有在启用正确的齿轮档后才能确保剩余零件程序的顺利执行。主轴处于转速控制运行时，追加的齿轮档切换将在 DryRun 取消后启动的系统 REPOS 中执行。此时会与 PLC 进行完整的齿轮档切换信号对话，并换入最后编写的齿轮档。

在 REPOS 期间，若零件程序中编写的齿轮档与 NC/PLC 接口提供的当前齿轮档一致，则不会执行齿轮档切换。

上述情形同样适用于 SERUPRO 功能。

程序段搜索 SERUPRO 的更多相关信息请参见：
更多信息
 速度、设定值/实际值系统、闭环控制 (页 141)

边界条件

齿轮档切换被抑制时，输出的主轴转速被限制在为当前齿轮档设定的转速范围内。
 对 REPOS 期间追加执行齿轮档切换的限制：

- 主轴在取消程序段或目标程序段中作为命令主轴（同步动作）或 PLC 主轴（SINU_RotateSpindle (Basic Program FC18)）时，不追加执行齿轮档切换。
- 由于主轴处于定位模式或轴模式，或由于耦合生效导致无法换入齿轮档时，将触发报警 22011“通道%1 程序段%3 主轴 2% 无法切换至编写的齿轮档”。

示例

空运行进给（DryRun）下的齿轮档切换

```

; 第 1 个用于初始状态
N00 M3 S1000 M41                ; 换入齿轮档 1
M0                                ; 零件程序停止

; PI 服务：激活空运行进给（DryRun）
                                   ; （配置）
N10 M42                          ; 请求齿轮档 2，齿轮档切换不执行
N11 G0 X0 Y0 Z0                  ; 轴定位
N12 M0                            ; 零件程序停止

; PI 服务：取消空运行进给（DryRun）
                                   ; 执行 REORG 和 REPOS
                                   ; 现在执行齿轮档切换，换入齿轮档 2

N20 G1 Z100 F1000
...
N99 M30                          ; 零件程序结束
    
```

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.spindle.in.requestedGearStageNumber	LBP_Axis*.E_SetpGearA..C	DB31, ...DBX82.0..2
<Axis>.spindle.in.gearChangeoverRequested	LBP_Axis*.E_GearChange	DB31, ...DBX82.3
<Axis>.spindle.in.speedSetpointAboveMax	LBP_Axis*.E_SetSpeedLimit	DB31, ...DBX83.1
<Axis>.spindle.in.speedSetpointBelowMin	LBP_Axis*.E_SetSpeedIncreased	DB31, ...DBX83.2

13.5 其他主轴功能调整设置

通过机床数据可激活下列主轴功能：

MD35035 \$MA_SPIND_FUNCTION_MASK, <位> = <值>

<位>	<值>	含义
0 ... 2	1	激活：空运行进给（DryRun）、程序测试和 SERUPRO 模式下的齿轮档切换特性 参见“在 DryRun、程序测试和 SERUPRO 下抑制齿轮档切换 (页 481)”。
4	1	将 NC 程序中编写的主轴转速 s... 接收至 SD43200 \$SA_SPIND_S。
5	0	在 JOG 运行模式下运行主轴时，采用当前有效的转速设定。 参见 SD41200 \$SN_JOGSPIND_SET_VELO
	1	在 JOG 运行模式下运行主轴时，SD43200 \$SA_SPIND_S 中设置的转速生效
8	1	将 NC 程序中编写的切削速度 s... 接收至 SD43202 \$SA_SPIND_CONSTCUT_S。
10	0	SD 43206 \$SA_SPIND_SPEED_TYPE 不因 NC 程序或通道设置而改变。
	1	若主轴是主主轴，则进给类型（G 组 15 的有效值）根据 SD43206 \$SA_SPIND_SPEED_TYPE 得到应用。 若主轴不是主主轴，则 SD43206 \$SA_SPIND_SPEED_TYPE 保持不变。
12	零脉冲搜索时主轴倍率的有效性，M19、SPOS、SPOSA = 0:	
	0	主轴倍率不生效。
	1	主轴倍率生效。

<位>	<值>	含义
19	与隐性辅助功能输出 M19 相关的特性:	
	0	并且 MD20850 \$MC_SPOS_TO_VDI == 0: 在 SPOS 和 SPOSA 中不将辅助功能 M19 输出至 PLC。
	1	在 SPOS 和 SPOSA 中将辅助功能 M19 输出至 PLC。 地址扩展对应主轴编号。 参见“定位模式 (页 403)”
20	与隐性辅助功能输出 M70 相关的特性:	
	0	不将隐性辅助功能 M70 输出至 PLC。 提示: 编写的辅助功能 M70 始终会输出至 PLC。
	1	在从主轴模式过渡至轴模式时, 将辅助功能 M70 输出至 PLC。 地址扩展对应主轴编号。 参见“隐性过渡至进给轴模式 (页 425)”。
22	在刚性攻丝 (G331、G332) 中用于将主轴旋转方向反转的轴专用 NC/PLC 接口信号 <Axis>.spindle.out.invertM3M4 (M3/M4 反向) 的关闭。	
	0	接口信号 <Axis>.spindle.out.invertM3M4 生效 。 提示 • 默认情形: 在 NC 程序启动时确定该接口信号的值。该接口信号随后的变化在 NC 程序内无影响。 • 特殊情形: 若在 NC 程序执行期间也需要将该接口信号的变化考虑在内, 则须以针对具体用户的方式实现。
	1	接口信号 <Axis>.spindle.out.invertM3M4 不生效 。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.spindle.out.invertM3M4	LBP_Axis*.A_M3M4Inv	DB31, ...DBX17.6

13.6 可切换主轴

功能

借助“可切换的主轴”功能，能够就通道主轴写入可应用在不同通道中的通用 NC 程序。其中通过主轴编号转换器将 NC 程序中使用的逻辑主轴编号转换成物理主轴编号。因此，在 NC 程序中编写的主轴编号相同的情况下，在不同通道中运行不同的物理主轴（机床轴）。

这样一来，通过“跨通道取轴”功能换入或交还的物理主轴便不必在零件程序中显性指定。

机床数据

物理主轴编号

每个主轴必须借助明确的可配置的编号（物理主轴编号）被指定给机床轴：

```
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[<机床轴>] = <物理主轴编号>
```

激活主轴编号转换器

主轴编号转换器通过以下机床数据针对具体通道激活：

```
MD20092 $MC_SPIND_ASSIGN_TAB_ENABLE = TRUE
```

设定数据

主轴编号转换器

为设置通道专用的主轴编号转换器，为每个逻辑主轴编号或编写的地址扩展（借此在 NC 程序中响应主轴（通道主轴））指定一个物理主轴编号（机床轴）。

```
SD42800 $SC_SPIND_ASSIGN_TAB[<逻辑主轴编号>] = <物理主轴编号>
```

更多信息：

- 可自 SD42800 \$SC_SPIND_ASSIGN_TAB[0] 读取当前主主轴的逻辑主轴编号。该数据不允许写入。在编写 SETMS 时，该数据会更新。
对于未使用的主轴必须在 SD42800 中输入 0 值。
- 在辅助功能输出中，系统将物理主轴编号作为地址扩展输出。
- 主轴编号转换器适用于 NC 程序中的主轴编程以及同步动作。
- 主轴编号转换器不适用于通过 NC/PLC 接口使用功能块 SINU_PositionSpindle (Basic Program FC18) 进行的主轴设定，因为在此情况下直接对物理主轴进行定址。
- 主轴转换涉及的系统变量有：\$P_S、\$P_SDIR、\$P_SMODE、\$P_GWPS、\$AC_SDIR、\$AC_SMODE、\$AC_MSNUM、\$AA_S

边界条件

- 可切换的通道主轴**不能**替代“跨通道取轴”功能。
- 可通过 NC 程序、PLC 用户程序或 PI 服务写入设定数据（SD42800 \$SC_SPIND_ASSIGN_TAB）来修改主轴转换。该修改立即生效。
- 在主轴编号转换器中只可使用指定给该通道的物理主轴（MD20070 \$MC_AXCONF_MACHAX_USED）。
若在主轴编号转换器中指定在另一通道中生效的物理主轴，则根据 MD30552 \$MA_AUTO_GET_TYPE 中的设置，或是为该通道请求物理主轴，或是显示报警 16105“指定的主轴不存在”。
- 若通过 PLC 或 HMI 写入 SD42800 \$SC_SPIND_ASSIGN_TAB[<n>]，那么对应通道必须处于复位状态，或者必须确保待切换主轴未在运行中的零件程序中使用。
可以通过预处理停止 STOPRE 来实现同步特性。
- 可以将一个逻辑主轴转换成多个物理主轴。但是在操作界面上显示逻辑主轴时会依据转换表产生多重含义。

示例

指定主轴编号和机床轴：

```
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX [AX4] = 1
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[ AX5 ] = 2
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX [AX6] = 3
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX [AX7] = 5
```

在通道中接收机床轴：

```
MD20070 $MC_AXCONF_MACHAX_USED[0] = 4
MD20070 $MC_AXCONF_MACHAX_USED[1] = 5
MD20070 $MC_AXCONF_MACHAX_USED[2] = 6
MD20070 $MC_AXCONF_MACHAX_USED[3] = 7
```

定义主主轴：

```
MD20090 $MC_SPIND_DEF_MASTER_SPIND = 1
```

主轴编号转换器

```
MD20092 ; 激活主轴编号转换器
$MC_SPIND_ASSIGN_TAB_ENABLE=1
```

13.6 可切换主轴

SD42800	; 主主轴如配置
\$SC_SPIND_ASSIGN_TAB[0]=1	
SD42800	; 表的初始设置
\$SC_SPIND_ASSIGN_TAB[1]=1	
SD42800	
\$SC_SPIND_ASSIGN_TAB[2]=2	
SD42800	
\$SC_SPIND_ASSIGN_TAB[3]=3	
SD42800	; 逻辑主轴未指定
\$SC_SPIND_ASSIGN_TAB[4]=0	
; 地址扩展为 1 输出: M1=3 S1=1000	
; 主轴 1 (物理主主轴) 旋转	
M3 S1000	
...	
; 逻辑主轴 1 被指定给物理主轴 5	
SD42800	
\$SC_SPIND_ASSIGN_TAB[1]=5	
; 逻辑主轴 2 被指定给物理主轴 3	
SD42800	
\$SC_SPIND_ASSIGN_TAB[2]=3	
; 注意: 物理主轴 3 现在被设定了两次。	
; 编写逻辑主轴 2 和 3 时	
; 始终会对物理主轴 3 进行响应。	
; 在机床初始画面中两根主轴均旋转。	
SETMS(2)	; SD42800 \$SC_SPIND_ASSIGN_TAB[0] = 2
; 主主轴 = 地址扩展 = 2, 输出未转换的主轴编号 M3 = 5	
; 配置编号“3”的物理主轴停止。	
M5	
GET(S4)	; 报警 16105: 逻辑主轴 4
; 无法转换	
RELEASE(S1)	; 通道主轴 1 = 物理主轴 5
; 被使能。	

13.7 编程

13.7.1 在零件程序中编程

编程指令

指令	说明														
SETMS:	主主轴为以下机床数据中设定的主轴: MD20090 \$MC_SPIND_DEF_MASTER_SPIND (通道中主主轴的缺省设置)														
SETMS (<n>):	编号 <n> 的主轴为主主轴 (可与缺省设置不同: MD20090 \$MC_SPIND_DEF_MASTER_SPIND) 必须为以下功能定义主主轴: <table> <tr> <td>• G95:</td><td>旋转进给率</td></tr> <tr> <td>• G96 S.../G961 S...:</td><td>恒定切削速度, 单位 m/min 或 ft/min</td></tr> <tr> <td>• G97/G971:</td><td>取消 G96/G961, 冻结最后的主轴转速</td></tr> <tr> <td>• G63:</td><td>带弹性卡头的攻丝</td></tr> <tr> <td>• G33/G34/G35:</td><td>螺纹切削</td></tr> <tr> <td>• G331/G332:</td><td>刚性攻丝</td></tr> <tr> <td>• G4 S...:</td><td>暂停时间, 单位: 主轴转数</td></tr> </table> • 未指定主轴编号的 M3、M4、M5、S、SVC、SPOS、M19、SPOSA、M40、M41 至 M45 和 WAITS 编程 可跨越复位/零件程序结束和零件程序开始保留当前的主主轴设置。这通过以下机床数据设置: - MD20110 \$MC_RESET_MODE_MASK - MD20112 \$MC_START_MODE_MASK	• G95:	旋转进给率	• G96 S.../G961 S...:	恒定切削速度, 单位 m/min 或 ft/min	• G97/G971:	取消 G96/G961, 冻结最后的主轴转速	• G63:	带弹性卡头的攻丝	• G33/G34/G35:	螺纹切削	• G331/G332:	刚性攻丝	• G4 S...:	暂停时间, 单位: 主轴转数
• G95:	旋转进给率														
• G96 S.../G961 S...:	恒定切削速度, 单位 m/min 或 ft/min														
• G97/G971:	取消 G96/G961, 冻结最后的主轴转速														
• G63:	带弹性卡头的攻丝														
• G33/G34/G35:	螺纹切削														
• G331/G332:	刚性攻丝														
• G4 S...:	暂停时间, 单位: 主轴转数														
M3:	主主轴顺时针方向旋转														
M<n>=3:	编号为 <n> 的主轴顺时针方向旋转														
M4:	主主轴逆时针方向旋转														
M<n>=4:	编号为 <n> 的主轴逆时针方向旋转														
M5:	主主轴无定向停止														
M<n>=5:	编号为 <n> 的主轴无定向停止														

指令	说明
S...:	主主轴的转速，单位 rpm
S<n>=...:	编号为 <n> 的主轴的转速，单位 rpm
SVC=...:	主主轴切削速度，单位 m/min 或 ft/min
SVC [<n>]=...:	编号为 <n> 的主轴的切削速度，单位 m/min 或 ft/min
SPOS=...:	主主轴定位
SPOS [<n>]=...:	编号为 <n> 的主轴定位 主轴到达位置以后才执行程序段转换。
SPOSA=...:	主主轴定位
SPOSA [<n>]=...:	编号为 <n> 的主轴定位 立即执行程序段转换。主轴独立于之后的零件程序执行进行定位，直至到达位置。
SPOS=DC (...): SPOS [<n>]=DC (...): SPOSA=DC (...): SPOSA [<n>]=DC (...):	从运动状态开始定位时，运动方向保持不变，并逼近位置。从静止状态开始定位时，沿最短路径逼近位置。
SPOS=ACN (...): SPOS [<n>]=ACN (...): SPOSA=ACN (...): SPOSA [<n>]=ACN (...):	始终采用负方向逼近位置。必要时在定位前反转运动方向。
SPOS=ACP (...): SPOS [<n>]=ACP (...): SPOSA=ACP (...): SPOSA [<n>]=ACP (...):	始终采用正方向逼近位置。 必要时在定位前反转运动方向。
SPOS=IC (...): SPOS [<n>]=IC (...): SPOSA=IC (...): SPOSA [<n>]=IC (...):	设定运行行程。运行方向取决于行程的符号。如果主轴已在运行中，必要时会反转运行方向，从而能够以编写的方向运行。 在运行期间越过零脉冲时，若尚未执行回参考点，或通过接口信号请求回参考点，那么主轴将自动与零脉冲同步。
M19:	将主主轴定位至 SD43240 中的位置。
M [<n>]=19:	将编号为 <n> 的主轴定位至 SD43240 中的位置。 主轴到达位置以后才执行程序段转换。

指令	说明
M70: M<n>=70:	停止主轴并激活位置闭环控制，选择参数组零，激活轴模式 针对主主轴 针对编号为 <n> 的主轴
SPCON: SPCON (<n>): SPCON (<n>, <m>):	主轴位置闭环控制 ON 针对主主轴 针对编号为 <n> 的主轴 针对编号为 <n> 和 <m> 的主轴
PCOF: SPCOF (<n>): SPCOF (<n>, <m>):	主轴位置闭环控制 OFF，激活转速控制运行 针对主主轴 针对编号为 <n> 的主轴 针对编号为 <n> 和 <m> 的主轴
FPRAON (S<n>):	主轴 <n> 旋转进给率 ON，从主主轴推导
FPRAON (S<n>, S<m>):	主轴 <n> 旋转进给率 ON，从主轴 <m> 推导 旋转进给值必须通过 FA[S<m>] 设定。
FPRAOF (S<n>):	主轴 <n> 旋转进给率 OFF
C30 G90 G1 F3600	回转轴 C（轴模式下的主轴）以 3600 degrees/min = 10 rev/min 的速度运行至 30 度位置
G25 S...: G25 S<n>:	可编程的主轴转速下限 针对主主轴 针对编号为 <n> 的主轴
G26 S...: G26 S<n>:	可编程的主轴转速上限 针对主主轴 针对编号为 <n> 的主轴
LIMS=...: LIMS [<n>]=...:	G96、G961、G97 下可编程的主轴转速限值 针对主主轴 针对编号为 <n> 的主轴

指令	说明
VELOLIM[<Spindel>]=. ...	可编程的、针对配置的齿轮档的转速上限 通过机床数据（MD30455 \$MA_MISC_FUNCTION_MASK, 位 6）可为零件程序中的编程设置，VELOLIM 是同时对轴模式和主轴模式生效（位 6 = 1），还是需要为两种模式分别编写（位 6 = 0）。若分别编写，在编程中通过名称选择模式： • 主轴名称 S<n> 用于主轴模式 • 轴名称，例如 “C”，用于轴模式 补偿值的基准： • 轴模式下的主轴（MD30455 位 6 = 0）： 以配置的最大轴速度为基准（MD32000 \$MA_MAX_AX_VELO）。 • 主轴模式或轴模式下的主轴（MD30455 位 6 = 1）： 以生效的齿轮档的最大转速为基准 （MD35130 \$MA_GEAR_STEP_MAX_VELO_LIMIT[<n>]）
WAITS:	对主轴的同步指令 执行以下程序段时系统将会等待，直至之前的 NC 程序段中通过 SPOSA 编写的主轴到达其位置（精准停）。 M5 后 WAITS: 等待，直至主轴停止。 M3/M4 后 WAITS: 等待，直至主轴达到设定转速。
WAITS(<n>):	对编号为 <n> 的主轴的同步指令
WAITS(<n>, <m>):	对编号为 <n> 和 <m> 的主轴的同步指令
FA[S<n>]:	为主轴 <n> 以 [degrees/min] 为单位编写定位速度（轴进给） FA[S<n>]=0 时，为以下机床数据配置的值重新生效： MD35300 \$MA_SPIND_POSCTRL_VELO
OVRA[S<n>]:	为主轴 <n> 以 [%] 编写轴倍率值
ACC[S<n>]:	为主轴 <n> 以 [%] 编写轴加速度

指令	说明
SPI (<n>) :	通过 SPI (<n>) 根据以下机床数据： MD35000 \$MA_SPIND_ASSIGN_TO_MACHAX[] 将一个主轴编号转换为 AXIS 数据类型。 SPI 用于通过主轴编写轴功能。 以下指令可与 SPI 配合使用： - 框架指令： - CTRANS () - CFINE () - CMIRROR () - CSCALE () - 副主轴的速度和加速度值： - FA[SPI (<n>)] - ACC[SPI (<n>)] - OVRA[SPI (<n>)] - 系统变量： \$P_PFRAME[SPI (<n>), TR]=<值> \$P_PFRAME= CTRANS (X, <轴值>, Y, <轴值>, SPI (<n>), <轴值>) \$P_PFRAME= CSCALE (X, <标度>, Y, <标度>, SPI (<n>), <标度>) \$P_PFRAME=CMIRROR (S<n>, Y, Z) \$P_UBFR=CTRANS (A, 10) :CFINE (19, 0.1)
M40:	为主主轴自动选择齿轮档
M<n>=40:	为编号为 <n> 的主轴自动选择齿轮档
M41 至 M45:	为主主轴选择齿轮档 1 至 5
M<n>=41 至 M<n>=45:	为编号为 <n> 的主轴选择齿轮档 1 至 5

说明

通道中配置了主轴时，M 功能 M3、M4、M5 和 M70 不在 <Chan>.auxMDynFuncs.in.mxDecoded 中输出。这些 M 功能作为扩展 M 功能在 <Chan>.auxMFuncs[1].in.extendedAddress ff. 和相应的 <Axis>.spindle.in.mFuncNumber 中提供。

更多信息

关于主轴编程的详细信息参见编程手册之 NC 编程。

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Chan>.auxMFuncs[1].in.extendedAddress	LBP_Chan*.ExtM1 - ExtM5	DB21, ...DBW68
<Axis>.spindle.in.mFuncNumber	LBP_Axis*.E_MFunct	DB31, ...DBB86
<Chan>.auxMDynFuncs.in.mxxDecoded	LBP_Chan*.MDyn[x]	DB21, ...DBB194
<Chan>.auxMDynFuncs.in.m64Decoded	LBP_Chan*.MDyn[64]	DB21, ...DBB202

13.7.2 通过同步动作编程

说明

M 功能 M40 至 M45 也可在同步动作中编程。

此时须注意：

- 零件程序段中编写的 M40 ... M45 不会对同步动作的自动齿轮档切换的当前状态产生影响，反之亦然。
- 使用 S 值和 M40 编写的自动齿轮档切换分别对同步动作和零件程序生效。
- 上电后 M40 取消。
通过同步动作设定 S 值时不会进行齿轮档调整。
- 通过同步动作编写的 M40 对同步动作始终生效（模态生效），复位后仍保留。
- 依据零件程序中的编程，M41 ... M45 选择的是第一个到第五个齿轮档。
为执行功能需进行跨通道取轴。
执行完齿轮档切换后，主轴处于中性状态（与 M3、M4、M5 编程的特性相似）。

更多信息

通过同步动作编程主轴及主轴运动的更多详细信息参见：

- 编程手册之 NC 编程
- 功能手册之同步动作分册

13.7.3 通过 PLC 使用 SINU_RotateSpindle 编程主轴控制

通过 PLC 使用 SINU_RotateSpindle (Basic Program FC18) 设定旋转方向和转速时，NC 能确定与转速相匹配的齿轮档并将其换入。这相当于零件程序编程中的 M40 功能。

为了激活齿轮档选择，必须在 PLC 用户程序中调用 SINU_RotateSpindle (Basic Program FC18) 时设置对应的启动代码。

更多信息

通过 PLC 使用 SINU_RotateSpindle (Basic Program FC18) 编程主轴控制的更多详细信息参见：

- PLC 功能手册

13.7.4 外部编程 (PLC)

SD43300 和 SD42600

借助轴专用设定数据：

SD43300 \$SA_ASSIGN_FEED_PER_REV_SOURCE（主轴的旋转进给率）

可在 JOG 运行方式下通过以下通道专用设定数据：

SD42600 \$SC_JOG_FEED_PER_REV_SOURCE（JOG 下旋转进给控制）

从外部选择采用旋转进给的运行。

通过该设定数据可进行以下设置：

>0:	用于推导旋转进给率的回转轴/主轴的机床轴编号。
-1:	旋转进给率从轴/主轴所在通道的主主轴推导。
0:	功能取消。

FPRAON (S2)

主轴 S2 启用旋转进给率（ON），从主主轴推导

FPRAON (S2, A)

主轴 S2 启用旋转进给率（ON），从轴 A 推导
旋转进给值必须以 FA[Sn] 设定。

FPRAOF (S2)

主轴 S2 不启用旋转进给率 (OFF)。

SPI (n)

也可编写 SPI (n) 来代替 SPI (Sn)。

13.8 主轴监控

13.8.1 允许的转速范围

决定主轴转速范围的因素有：参数设置或编写的转速限值，以及生效的轴功能（G94、G95、G96、G961、G97、G971、G33、G34、G35、G331、G332 等）。

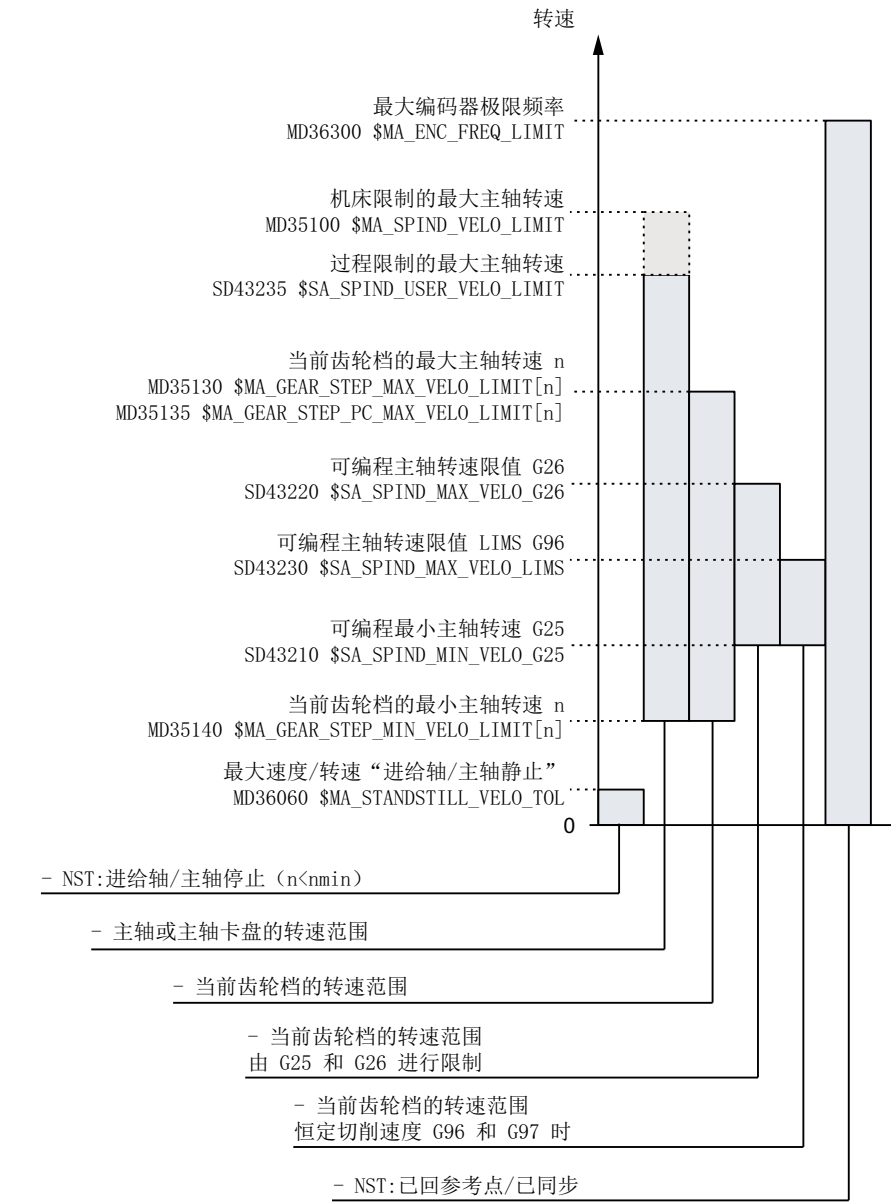


图 13-8 主轴监控范围/转速范围

13.8.2 进给轴/主轴停止

仅当主轴停止时，才能使用换刀、打开机床门、使能轨迹进给等功能。

功能

当不再生成设定值，且主轴转速实际值低于为“进给轴/主轴停止”配置的阈值时，便达到“进给轴/主轴停止”状态。

MD36060 \$MA_STANDSTILL_VELO_TOL（“进给轴/主轴停止” 最大速度/转速）

主轴进入停止状态时，以下 NC/PLC 接口信号置位：

<Axis>.basic.in.isStationary（进给轴/主轴停止）

有效性

主轴静止状态监控在所有主轴运行方式以及轴模式下生效。

禁止轨迹进给

主轴在控制模式下停止时（M5），若以下机床数据置位，那么轨迹进给会被禁止：

MD35510 \$MA_SPIND_STOPPED_AT_IPO_START（主轴停止时进给率使能）

主轴进入停止状态时，轨迹进给重新使能。

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.isStationary	LBP_Axis*.E_Stat	DB31, ...DBX61.4

13.8.3 主轴位于设定区域内

功能

主轴监控“主轴处于设定区域”用于检查：

- 是否达到编写的主轴转速
- 主轴是否停止：
`<Axis>.basic.in.isStationary`（进给轴/主轴停止）= 1
- 主轴是否仍处于加速或制动阶段

在控制模式下，系统会将设定转速和实际转速进行比较。若实际转速的偏离值大于通过机床数据设定的主轴转速公差（见下文），则：

- 将以下 NC/PLC 接口信号置“0”：
`<Axis>.spindle.in.speedSetpointReached`（主轴处于设定区域）= 0
- 不使能下一个加工程序段（取决于机床数据
`MD35500 $MA_SPIND_ON_SPEED_AT_IPO_START` 中的设置，参见“进给轴/主轴停止（页 498）”）。

主轴设定转速

主轴设定转速由编写的转速接合主轴倍率和生效的限制得出。

编写的转速被限制为上限值或提升至下限值的情形会通过

`<Axis>.spindle.in.speedSetpointAboveMax`（设定转速已限制）或

`<Axis>.spindle.in.speedSetpointBelowMin`（设定转速已提升）显示（另见“齿轮档的最小/最大转速（页 500）”）。设定转速到达公差范围的情形则不会被阻止。

设定转速的公差范围

设定转速的公差范围通过主轴转速公差系数定义：

`MD35150 $MA_SPIND_DES_VELO_TOL`

示例：

`MD35150 $MA_SPIND_DES_VELO_TOL = 0.1`

⇒ 允许主轴实际转速和设定转速有 $\pm 10\%$ 的偏差。

若主轴实际转速处于公差范围内，则以下 NC/PLC 接口信号会置“1”：

`<Axis>.spindle.in.speedSetpointReached`（主轴处于设定区域）= 1

特殊情形：

若主轴转速公差被设为“0”，那么 <Axis>.spindle.in.speedSetpointReached（主轴处于设定区域）将永久置“1”，此时将不会进行轨迹控制。

转速变化

系统仅会在运行程序段开始时以及编写了转速变化时进行轨迹控制。例如，过载导致超出转速公差范围时，轨迹运动不会自动停止。

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.isStationary	LBP_Axis*.E_Stat	DB31, ...DBX61.4
<Axis>.spindle.in.speedSetpointAboveMax	LBP_Axis*.E_SetSpeedLimit	DB31, ...DBX83.1
<Axis>.spindle.in.speedSetpointBelowMin	LBP_Axis*.E_SetSpeedIncreased	DB31, ...DBX83.2
<Axis>.spindle.in.speedSetpointReached	LBP_Axis*.E_SetRange	DB31, ...DBX83.5

13.8.4 齿轮档的最小/最大转速

最小转速

主轴齿轮档的最小转速在以下机床数据中配置：

MD35140 \$MA_GEAR_STEP_MIN_VELO_LIMIT[<n>]

基于倍率生成的转速设定值不会低于该最小转速。

若编写了小于最小转速的 S 值，那么该设定转速会被提升至最小转速，且以下 NC/PLC 接口信号置位：

<Axis>.spindle.in.speedSetpointBelowMin（设定转速已提升）

齿轮档的最小转速仅在转速运行中生效，且只有以下情形下转速才会低于该值：

- 主轴倍率 0%
- M5
- S0

- <Axis>.basic.out.feedStopSpindleStop（主轴停止）
- <Axis>.basic.out.enableCtrl（伺服使能）取消
- <Chan>.basic.out.reset（复位）
- <Axis>.basic.out.resetMovement（删除剩余行程/主轴复位）
- <Axis>.spindle.out.oscillEnable（往复转速）
- <Chan>.basic.out.ncStopAxesAndSpindles（进给轴和主轴 NC 停止）
- <Axis>.basic.out.disable（进给轴/主轴禁用）
- <Axis>.spindle.out.deleteSValue（删除 S 值）

最大转速

主轴齿轮档的最大转速在以下机床数据中配置：

MD35130 \$MA_GEAR_STEP_MAX_VELO_LIMIT[<n>]

基于倍率生成的转速设定值会被限制在该转速以下。

设定转速被限制为最大转速时，以下 NC/PLC 接口信号置位：

<Axis>.spindle.in.speedSetpointAboveMax（设定转速已限制）

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.disable	LBP_Axis*.A_ASpDisable	DB31, ...DBX1.3
<Axis>.basic.out.enableCtrl	LBP_Axis*.A_ContrEnable	DB31, ...DBX2.1
<Axis>.basic.out.resetMovement	LBP_Axis*.A_DeIDTGSReset	DB31, ...DBX2.2
<Axis>.basic.out.feedStopSpindleStop	LBP_Axis*.A_FDSpStop	DB31, ...DBX4.3
<Chan>.basic.out.ncStopAxesAndSpindles	LBP_Chan*.A_NCStopASp	DB21, ...DBX7.4
<Chan>.basic.out.reset	LBP_Chan*.A_Reset	DB21, ...DBX7.7
<Axis>.spindle.out.deleteSValue	LBP_Axis*.A_DeleteS	DB31, ...DBX16.7
<Axis>.spindle.out.oscillEnable	LBP_Axis*.A_OscilSpeed	DB31, ...DBX18.5

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.spindle.in.speedSetpointAboveMax	LBP_Axis*.E_SetSpeedLimit	DB31, ...DBX83.1
<Axis>.spindle.in.speedSetpointBelowMin	LBP_Axis*.E_SetSpeedIncreased	DB31, ...DBX83.2

13.8.5 主轴转速上限

最大主轴转速：受机床条件制约的、可参数设置的限值

受机床条件制约的最大主轴转速，例如用于保护主轴卡盘或刀具，通过以下机床数据设置：

MD35100 \$MA_SPIND_VELO_LIMIT（最大主轴转速）

NC 会对主轴转速进行实际值监控，即基于当前生效的齿轮档进行监控。

说明

机床制造商

如需修改机床数据 MD35100 中的最大主轴转速，建议只在主轴停止的情形下进行。对 NC 复位后保持生效的主轴尤其需要遵循这一点（另见 MD35040 SPIND_ACTIVE_AFTER_RESET）。否则可能会触发报警 22100。

更多信息

- 制造商文档：参数手册“详细机床数据说明”
- 用户文档：诊断手册

超出限值时的响应

主轴转速实际值超出所设置的主轴最大转速的幅度高于公差值时，

“主轴转速实际值” > MD35100 \$MA_SPIND_VELO_LIMIT（最大主轴转速）
+ MD35150 \$MA_SPIND_DES_VELO_TOL（主轴转速公差值）

NC 会进行以下响应：

- <Axis>.spindle.in.speedLimitationExceeded = 1（超出转速限值）
- 显示报警 22100“超出卡盘转速”
- 对通道中的所有轴和主轴进行制动，使其停止

说明

回转轴/主轴

若主轴还短暂地作为回转轴运行，那么以下情形下会显示报警 22100“超出卡盘转速”：

“回转轴转速实际值” > MD35100 \$MA_SPIND_VELO_LIMIT（最大主轴转速）
+ MD35150 \$MA_SPIND_DES_VELO_TOL（主轴转速公差值）

解决方法：主轴作为回转轴运行期间，将最大主轴转速调整为最大回转轴转速：

MD35100 \$MA_SPIND_VELO_LIMIT = MD32000 \$MA_MAX_AX_VELO（最大回转轴转速）

必须触发复位才能使此修改生效。

最大主轴转速：受过程制约的、可设置的限值

受过程制约的最大主轴转速通过以下设定数据设置：

SD43235 \$SA_SPIND_USER_VELO_LIMIT（最大主轴转速）

可通过以下方式写入设定数据来修改该值：

- 在 NC 程序中编程
- 由机床操作人员通过操作界面手动修改

修改将立即生效。

对设定转速的限制

必要时，控制系统会将编写的主轴转速设定值限制在设定数据中定义的限值范围内。之后此限制会通过以下系统变量显示：

\$AC_SMAXVELO_INFO[<主轴编号>] == 21

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.spindle.in.speedLimitationExceeded	LBP_Axis*.E_SpeedLimit	DB31, ...DBX83.0

13.8.6 读取主轴参数

系统变量

生效/受限的主轴参数可通过以下系统变量读取。这些系统变量只提供转速控制运行和主轴定位运行的相关数值，使用时将主轴的编号作为索引输入。

系统变量	含义
\$AC_SMAXVELO[<n>]	可能的最大转速 [rev/min]
\$AC_SMAXVELO_INFO[<n>]	日期标识（转速降低）
\$AC_SMINVELO[<n>]	可能的最小转速 [rev/min]
\$AC_SMINVELO_INFO[<n>]	日期标识（转速提升）
\$AC_SMAXACC[<n>]	可能的最大主轴加速度 [rev/s ²]
\$AC_SMAXACC_INFO[<n>]	日期标识（加速度减小）
\$AC_SPIND_STATE[<n>]	转速控制运行中的主轴状态
<n>: 0, 1, 2, ... 最大主轴编号（0: 当前通道主主轴）	

更多信息

有关系统变量的详细说明请见：

参数手册之系统变量

NC/PLC 接口信号

系统会通过以下 NC/PLC 接口信号显示主轴转速受到限制或提升。

- <Axis>.spindle.in.speedSetpointAboveMax（设定转速已限制）
- <Axis>.spindle.in.speedSetpointBelowMin（设定转速已提升）

边界条件

主轴运行方式

系统变量提供的值取决于主轴运行方式：

- 转速控制运行：
所有系统变量均提供当前值。
- 定位模式：
系统变量 \$AC_SMAXVELO、\$AC_SMAXACC 和 \$AC_SPIND_STATE 提供有效值。系统变量 \$AC_SMINVELO 和 \$AC_SMINVELO_INFO 提供切换至转速控制运行时生效的数据。
- 轴模式（例如主轴参与 TRANSMIT、TRACYL 等转换功能，或作为轨迹运动的辅助轴）：
系统变量 \$AC_SPIND_STATE 也可在轴模式下使用。此外轴模式下还可使用动态数据专用的系统变量：
\$AA_VMAXM、\$AA_VMAXB 和 \$AA_VLFCT。
还读取轴运行方式的加速度和速度限制。因此，除了系统变量 \$AC_SMAXVELO 和 \$AC_MAXACC 之外，关联的 \$AC_SMAXVELO_INFO 和 \$AC_MAXACC_INFO 也收到不同的返回值。

程序段搜索 SERUPRO

在程序段搜索 SERUPRO 下可通过这些系统变量获取以下控制特性：

- 系统变量 \$AC_SMAXVELO / \$AC_SMAXACC 提供可显示的最大速度/加速度。
- \$AC_SMAXVELO_INFO 和 \$AC_SMAXACC_INFO 提供值“0”（无限制生效）。
- \$AC_SMINVELO 和 \$AC_SMINVELO_INFO 提供的数据和一般程序执行时相同。
- \$AC_SPIND_STATE 提供的是对应 SERUPRO 功能的状态。

示例

生效/受限的主轴 1 的主轴参数被循环写入 R 参数中。

程序代码

```
N05 IDS=1 WHENEVER TRUE DO $R10=$AC_SMAXVELO[1]
N10 IDS=2 WHENEVER TRUE DO $R11=$AC_SMAXVELO_INFO[1]
N15 IDS=3 WHENEVER TRUE DO $R12=$AC_SMINVELO[1]
N20 IDS=4 WHENEVER TRUE DO $R13=$AC_SMINVELO_INFO[1]
N25 IDS=5 WHENEVER TRUE DO $R14=$AC_SPIND_STATE[1]
```

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.spindle.in.speedSetpointAboveMax	LBP_Axis*.E_SetSpeedLimit	DB31, ...DBX83.1
<Axis>.spindle.in.speedSetpointBelowMin	LBP_Axis*.E_SetSpeedIncreased	DB31, ...DBX83.2

13.8.7 最大编码器极限频率

 小心

超限

控制系统会对主轴位置实际值编码器的最大编码器极限频率进行监控（可能会超出）。机床制造商在设计主轴电机、齿轮箱、测量齿轮箱、编码器，以及对应的机床数据时，须确保不会超出主轴位置实际值编码器的最大转速（机械极限转速）。

超出最大编码器频率

若主轴在控制模式下达到超出最大编码器极限频率的转速（编写了较大的 S 值；此时不得超出编码器的最大转速），那么同步将丢失。主轴会继续旋转，但功能降低。

在下列功能中主轴转速会被降低，直至生效的测量系统重新在编码器极限频率下工作：

- 螺纹切削（G33、G34、G35）
- 刚性攻丝（G331、G332）
- 旋转进给率（G95）
- 恒定切削速度（G96、G961、G97、G971）
- SPCON（主轴的位置闭环控制运行）

超出编码器极限频率时，系统会针对相关测量系统复位以下

NC/PLC 接口信号：

<Axis>.basic.in.enclSynchronized（已回参考点/已同步 1）

或

<Axis>.basic.in.enc2Synchronized（已回参考点/已同步 2）

并置位以下 NC/PLC 接口信号：

<Axis>.basic.in.enclFreqLimitExceeded（超出编码器极限频率 1）

或

<Axis>.basic.in.enc2FreqLimitExceeded（超出编码器极限频率 2）
。

若主轴处于轴模式下，则不允许超出最大编码器极限频率。
最大速度（MD32000 \$MA_MAX_AX_VELO）必须低于最大编码器极限频率，否则会触发报警 21610 并停止轴运行。

低于最大编码器极限频率

在超出了最大编码器极限频率之后，若之后又重新达到低于最大编码器极限频率的转速（编写较低的 S 值，修改主轴倍率开关等），那么主轴将自动与下一个零脉冲或下一个无接触式接近开关信号进行同步。对于之前同步丢失，且当前最大编码器极限频率高于主轴转速的位置测量编码器总是会进行新同步。

特点

若下列功能生效，则不允许超出最大编码器极限频率：

- 主轴定位模式、轴模式
- 螺纹切削（G33、G34、G35）
- 刚性攻丝（G331、G332，不适用于 G63）
- 旋转进给率（G95）
- 恒定切削速度（G96、G961、G97、G971）
- SPCON

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.enc1FreqLimitExceeded	LBP_Axis*.E_EncodFreq1	DB31, ...DBX60.2
<Axis>.basic.in.enc2FreqLimitExceeded	LBP_Axis*.E_EncodFreq2	DB31, ...DBX60.3
<Axis>.basic.in.enc1Synchronized	LBP_Axis*.E_RefSyn1	DB31, ...DBX60.4
<Axis>.basic.in.enc2Synchronized	LBP_Axis*.E_RefSyn2	DB31, ...DBX60.5

13.8.8 目标点监控

目标点监控

在主轴进行定位时（定位模式下），控制系统会监控主轴（其实际位置）和编写的主轴设定位置（目标点）之间的距离。

为此可在以下机床数据中设定两个限值，作为以主轴目标位置为起点的增量行程：

MD36000 \$MA_STOP_LIMIT_COARSE（粗准停限值）

和

MD36010 \$MA_STOP_LIMIT_FINE（精准停限值）

在不启用这两个限值的情况下，主轴定位也能有很好的精度，这取决于连接的主轴测量编码器、间隙、齿轮比等因素。

基于参数组的准停窗口

可基于参数组配置不同的准停窗口。

这样便可在轴模式和主轴定位中启用不同的精度进行工作。针对主轴定位，可为每个齿轮档分别配置准停窗口。

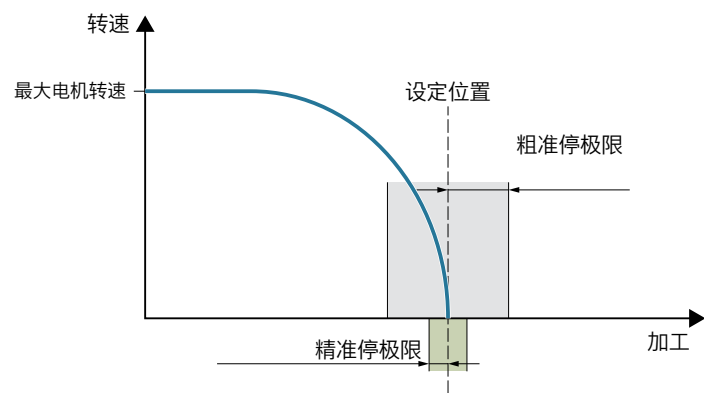


图 13-9 主轴的准停区域

<Axis>.basic.in.posCoarseReached 和 <Axis>.basic.in.posFineReached（采用粗/精准停到达位置）

通过机床数据：
MD36000 \$MA_STOP_LIMIT_COARSE（粗准停限值）
和
MD36010 \$MA_STOP_LIMIT_FINE（精准停限值）
定义的两个限值通过 NC/PLC 接口信号：
<Axis>.basic.in.posCoarseReached（采用粗准停到达位置）
和
<Axis>.basic.in.posFineReached（采用精准停到达位置）
输出至 PLC。

SPOS 和 M19 下的程序段转换

在通过 SPOS 或 M19 定位主轴时，程序段转换
根据目标点监控通过以下 NC/PLC 接口信号进行：
<Axis>.basic.in.posCoarseReached（采用精准停到达位置）。

此时程序段中编写的所有其他功能必须都已达到程序段结束标准（例如所有辅助功能均由 PLC 应答）。

使用 SPOSA 定位时，程序段转换独立于目标点监控进行。

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.posCoarseReached	LBP_Axis*.E_ExactCoarse	DB31, ...DBX60.6
<Axis>.basic.in.posFineReached	LBP_Axis*.E_ExactFine	DB31, ...DBX60.7

13.8.9 M40：转速超出配置的开关阈值时的自动齿轮档选择

功能

M40 生效时，即使编写的主轴转速 s... 超出配置的开关阈值，系统仍会自动选择齿轮档。

此时分为下列情形：

- **编写的转速过高**

编写的主轴转速大于为第 n 齿轮档配置的转速上限：

$S... > MD35110 \$MA\_GEAR\_STEP\_MAX\_VELO[<n>]$

在此情形下会切换至**最高齿轮档**（根据 MD35090 $\$MA\_NUM\_GEAR\_STEPS$ ）。

- **编写的转速过低**

编写的主轴转速小于为第一齿轮档配置的最小转速：

$S... < MD35120 \$MA\_GEAR\_STEP\_MIN\_VELO[1]$

在此情形下会切换至**第一齿轮档**。

- **编写的转速 = 0**

编写转速 0 时（s0），特性取决于第一齿轮档最小转速的配置 MD35120

$\$MA\_GEAR\_STEP\_MIN\_VELO[1]$ ：

- 若配置了 MD35120 $\$MA\_GEAR\_STEP\_MIN\_VELO[1] = 0$ ，那么在编写 s0 时将切换至**第一齿轮档**。
- 若配置了 MD35120 $\$MA\_GEAR\_STEP\_MIN\_VELO[1] > 0$ ，那么在编写 s0 时将**不切换齿轮档**，最后的齿轮档将保持生效。这样便可在不触发齿轮档切换的情况下通过 s0 停止主轴（替代 M5）。

有效性

通过零件程序、同步动作或 PLC - SINU_RotateSpindle (Basic Program FC18) 编程主轴转速 s... 时，自动齿轮档选择 (M40) 下向最高齿轮档或第一齿轮档的切换生效。

在 G331 攻丝的零件程序中进行转速编程时，也可通过此特性实现向第二齿轮组的切换（前提条件：MD35010 $\$MA\_GEAR\_STEP\_CHANGE\_ENABLE$ ，位 5 = 1）。

边界条件

使能齿轮档切换

此功能的前提条件是：通过下列机床数据对齿轮档切换进行了总体使能：

MD35010 $\$MA\_GEAR\_STEP\_CHANGE\_ENABLE$ （参数设置齿轮档切换）

MD35090 $\$MA\_NUM\_GEAR\_STEPS$ （建立的齿轮档的数量）

示例

自动齿轮档选择 M40 是 NC 复位后的缺省设置。

零件程序:

程序代码	注释
...	
N15 S3500 M3	; S3500 大于齿轮档 2 的 MD35110。切换至齿轮档 2。切换至齿轮档 2。
...	
N50 S0 M3	; 主轴停止, S0 不请求齿轮档切换 (S0 的特殊处理)。
...	
N95 S5 M3	; S5 小于齿轮档 1 的 MD35120。切换至齿轮档 1。
...	

主轴 1 (AX5) 的配置数据:

MD35010 \$MA_GEAR_STEP_CHANGE_ENABLE[AX5] = 1	; 使能齿轮档切换
MD35090 \$MA_NUM_GEAR_STEPS[AX5] = 2	; 现有齿轮档数量
MD35110 \$MA_GEAR_STEP_MAX_VELO[1,AX5] = 500	; 齿轮档 1 的开关阈值上限
MD35120 \$MA_GEAR_STEP_MIN_VELO[1,AX5] = 10	; 齿轮档 1 的开关阈值下限
MD35110 \$MA_GEAR_STEP_MAX_VELO[2,AX5] = 2000	; 齿轮档 2 的开关阈值上限
MD35120 \$MA_GEAR_STEP_MIN_VELO[2,AX5] = 500	; 齿轮档 2 的开关阈值下限

13.9 配备 SMI 24 的主轴 (Weiss 主轴)

13.9.1 简介

为了在控制系统中处理主轴的传感器数据, 到目前为止都需要将传感器连接至 I/O 模块, 或通过现场总线 (PROFINET IO) 与 PLC 通讯。

使用配备 SMI 24 的主轴 (Weiss 主轴) 则可将传感器数据通过 DRIVE-CLiQ 传输至驱动, 并作为驱动参数提供。使用循环驱动报文 139 又可将传感器数据从驱动传输至控制系统。这些数据在控制系统中作为以下系统数据提供:

- 系统变量
- OPI 变量
- NC/PLC 接口信号

前提条件

- 主轴使用编码器模块 SMI 24 通过 DRIVE-CLiQ 连接至驱动。
- 为主轴配置了驱动报文 139。

说明

驱动报文 139

配备编码器模块 SMI 24 的主轴一般也可通过另一个驱动报文运行。但是传感器数据只通过驱动报文 139 传输至控制系统。

13.9.2 传感器数据

主轴电机的传感器

主轴的传感器提供夹紧装置和电机轴角度位置的信息：

- 模拟量传感器 S1：夹紧状态
电压值在 0 - 10 V 范围内，取决于导杆位置。
- 数字量传感器 S4：活塞末端位置
 - 0 = 活塞未到达位置
 - 1 = 活塞到达位置，即活塞未占用
- 数字量传感器 S5：电机轴的角度位置
 - 0 = 电机轴未对准
 - 1 = 电机轴位置对准（前提条件：主轴停止）

传输传感器数据

传感器数据由编码器模块 SMI 24 在循环驱动报文 139 中作为过程数据 11 - 14 传输至控制系统。驱动报文 139 基于驱动报文 136，不过其传输的是传感器数据，而不是第 2 编码器的数据。对驱动报文 139 的详细说明参见：

更多信息

SINAMICS S120/S150 参数手册：功能图 - PROFIdrive

系统数据：传感器数据

传感器数据可在控制系统中通过以下系统数据读取：

含义	系统变量 \$VA_	NC/PLC 接口	OPI 变量	驱动参数
传感器配置	MOT_SENSOR_CONF[<轴>]	<Axis>.spindle.in.smi24.sensors... <Axis>.spindle.in.smi24.speedLimited	vaMotSensorConf	r5000
夹紧状态 ¹⁾	MOT_CLAMPING_STATE[<轴>]	<Axis>.spindle.in.smi24.sensorS1StateNumber	vaMotClampingState	r5001
模拟传感器 S1 测量值 ^{1) 2)}	MOT_SENSOR_ANA[<轴>]	<Axis>.spindle.in.smi24.sensorS1Value	vaMotSensorAna	r5002
数字量传感器状态	MOT_SENSOR_DIGI[<轴>]	<Axis>.spindle.in.smi24.sensorS4...	vaMotSensorDigi	r5003
<轴>：机床轴名称：AX1 ... AXn 或者主轴名称：S1 ... Sm				

¹⁾ 参见章节“夹紧状态 (页 514)”

²⁾ 传感器 S1：0 - 10 V

模拟量实际值：10000 增量，分辨率 1 mV

示例：

SIMATIC S7 输入模块：27648 增量，分辨率 0.36 mV

转换至配备 SMI 24 主轴时的匹配系数： $(10000 \text{ 增量} * 1 \text{ mV}) / (27648 \text{ 增量} * 0.36 \text{ mV})$
= 1.00469393

详细系统数据说明

NC/PLC 接口信号：

PLC 功能手册

系统变量：

参数手册“系统变量”

OPI 变量：

参数手册

驱动参数：

SINAMICS S120/S150 参数手册

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.spindle.in.smi24.sensorsPresent	LBP_Axis*.E_SensorAv	DB31, ...DBX132.0
<Axis>.spindle.in.smi24.sensorS1Present	LBP_Axis*.E_S1aMeasARod	DB31, ...DBX132.1
<Axis>.spindle.in.smi24.sensorS4Present	LBP_Axis*.E_S4dPistonStopP	DB31, ...DBX132.4
<Axis>.spindle.in.smi24.speedLimited	-	DB31, ...DBX133.2
<Axis>.spindle.in.smi24.sensorS1StateNumber	LBP_Axis*.E_ClampingState	DB31, ...DBX134.0 ..135.7
<Axis>.spindle.in.smi24.sensorS1Value	LBP_Axis*.E_AnalogSensor	DB31, ...DBX136.0 ..137.7
<Axis>.spindle.in.smi24.sensorS4Actuated	LBP_Axis*.E_S4PistonStopP	DB31, ...DBX138.4
<Axis>.spindle.in.smi24.sensorS5Actuated	LBP_Axis*.E_S5AngleShaft	DB31, ...DBX138.5

13.9.3 夹紧状态

传感器 S1 根据夹紧装置的位置提供一个 0 V - 10 V 的模拟电压值。该电压值通过系统数据提供给用户，供其分析夹紧状态。

说明

仅当驱动参数 r5000 中显示以下状态值时，下文说明的“分析传感器 S1 来生成夹紧状态的状态值”和“限制主轴转速”才会进行。

- r5000.0 == 1: 配有传感器
- r5000.1 == 1: 传感器 S1（夹紧状态）存在
- r5000.10 == 1: 生成状态值，转速限制 p5043 生效

另见章节“传感器数据 (页 512)”，段落：“系统数据：传感器数据”

状态值

为了简化分析，夹紧状态在系统数据中也作为状态字 0 - 11 提供。

夹紧范围对应特定的夹紧状态。电压范围可通过驱动参数 p5041[0...5] 设置。

针对电压范围，还可通过驱动参数 p5040 额外设置一个电压公差。

说明

“电压范围 ± 电压公差” 不允许重叠。

转速限值

对于状态值为 3 - 10 的夹紧状态，可通过驱动参数 p5043[0...6] 设定转速限值。在其他夹紧状态下（状态值 1、2 和 11），转速限值固定设置为 0 [rpm]。

控制系统会在各种夹紧状态下将主轴转速限制在相应限值内。

说明

转速限值的修改

满足以下条件时，在驱动参数 p5043[0...6] 中进行的转速限值修改才会在控制系统中生效（将主轴转速限制在新的转速限值范围内）：

- 上电复位，或控制系统的关闭/接通
- 为主轴取消“驻留”状态（参见功能手册“监控与补偿”，“机床轴驻留”章节）

关联：状态值、电压范围和转速限值之间

状态值 ¹⁾	夹紧状态	电压范围 ²⁾		转速限值
		上限	下限	
0	传感器 S1 不存在，或状态值无效	---	---	---
1	状态初始化中	---	---	³⁾
2	已松开，发出消息（故障状态）	---	p5041[0] + p5040	³⁾
3	已松开	p5041[0]	p5041[1]	p5043[0]
4	夹紧，带刀具	---	---	p5043[1]
5	松开，带刀具	---	---	p5043[2]
6	松开，无刀具	---	---	p5043[3]
7	已夹紧，带刀具 AND S4 == 0	p5041[2]	p5041[3]	p5043[4]
8	已夹紧，带刀具 AND S4 == 1			p5043[4]
9	夹紧，无刀具	---	---	p5043[5]

13.9 配备 SMI 24 的主轴 (Weiss 主轴)

状态值 ¹⁾	夹紧状态	电压范围 ²⁾		转速限值
		上限	下限	
10	已夹紧，无刀具	p5041[4]	p5041[5]	p5043[6]
11	已夹紧，发出消息（故障状态）	p5041[5] - p5040	---	³⁾

- 1) 状态值可在控制系统中通过以下系统数据读取：
 - 系统变量：

\$VA_MOT_CLAMPING_STATE[<轴>]
 - NC/PLC 接口：

<Axis>.spindle.in.smi24.sensorS1StateNumber
 - OPI 变量：

vaMotClampingState
 - 驱动参数：

r5001
- 2) p5041[0...5]：电压阈值，p5040：电压阈值公差
- 3) 转速限值固定设定：0 [rpm]

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.spindle.in.smi24.sensorS1StateNumber	LBP_Axis*.E_ClampingState	DB31, ...DBX134.0 ..135.7

13.9.4 其他驱动参数

p5042：过渡时间

在驱动参数 p5042 中可为夹紧状态识别设置以下时间：

- p5042[0]：“已夹紧，带刀具”的稳定时间
夹紧状态“已夹紧，带刀具”必须在主轴电机中保持至少为设置值的稳定时间，之后才能将状态发送至控制系统。
- p5042[1]：最大夹紧时间
从“已松开”状态过渡到“已夹紧，带道具”或“已夹紧，无道具”状态最多可持续的时间。

r5044: 夹紧循环中的转速限制

驱动参数 r5044 中显示在“已夹紧，无刀具”状态下生效的转速限值（p5043[6]）。
值 65535 表示转速限值未生效。

13.10 边界条件

13.10.1 闭环控制参数的修改

主轴不处于位置闭环控制下时，通过 NEWCONF 指令进行的机床数据修改在主轴未停止时同样生效。

修改闭环控制参数时，新值生效后可能会引起转速设定值跃变。闭环控制参数例如包括：

- MD32200 \$MA_POSCTRL_GAIN（KV 系数）
- MD32210 \$MA_POSCTRL_INTEGR_TIME（位置控制器积分时间）
- MD32410 \$MA_AX_JERK_TIME（轴向急动滤波器的时间常量）

13.11 示例

13.11.1 自动齿轮档选择 (M40)

示例

显示新程序段搜索变量的内容：
启用自动齿轮档选择（M40）：

S0...500	1. 齿轮档
S501..1000	2. 齿轮档
S1001..2000	3. 齿轮档

系统变量的内容：
\$P_SEARCH_S ; 收集的 S 值

13.11 示例

\$P_SEARCH_DIR ; 收集的旋转方向
\$P_SEARCH_GEAR ; 收集的齿轮档

收集的	S 值: ; 0/最后的转速	旋转方向:	齿轮档:
		-5	40/最后的副主 轴
N05 G94 M40 M3 S1000	; 1000	3	40
N10 G96 S222	; 222	3	40
N20 G97	; f (PlanAxPosPCS)*	3	40
N30 S1500	; 1500	3	40
N40 SPOS=0**	; 1500	-19	40
N50 M19**	; 1500	-19	40
N60 G94 G331 Z10 S300	; 300	-19	40
N70 M42	; 300	-19	42
N80 M4	; 300	4	42
N90 M70	; 300	70	42
N100 M3 M40	; 300	3	40
N999 M30			

* f (PlanAxPosPCS): 转速取决于工件坐标系中平面轴的当前位置。

** (\$P_SEARCH_SPOS 和 \$P_SEARCH_SPOSMODE 会被写入)

同步主轴

14.1 简要说明

14.1.1 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：
$$n_{\text{副主轴}} = k_{\text{传动比}} * n_{\text{主}} \quad \text{其中 } k_{\text{传动比}} = 1、2、3、\dots$$

主轴

位置同步性：
$$\varphi_{\text{副主轴}} = \varphi_{\text{主主轴}} + \Delta\varphi \quad \text{其中 } 0^\circ \leq \Delta\varphi < 360^\circ$$

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。

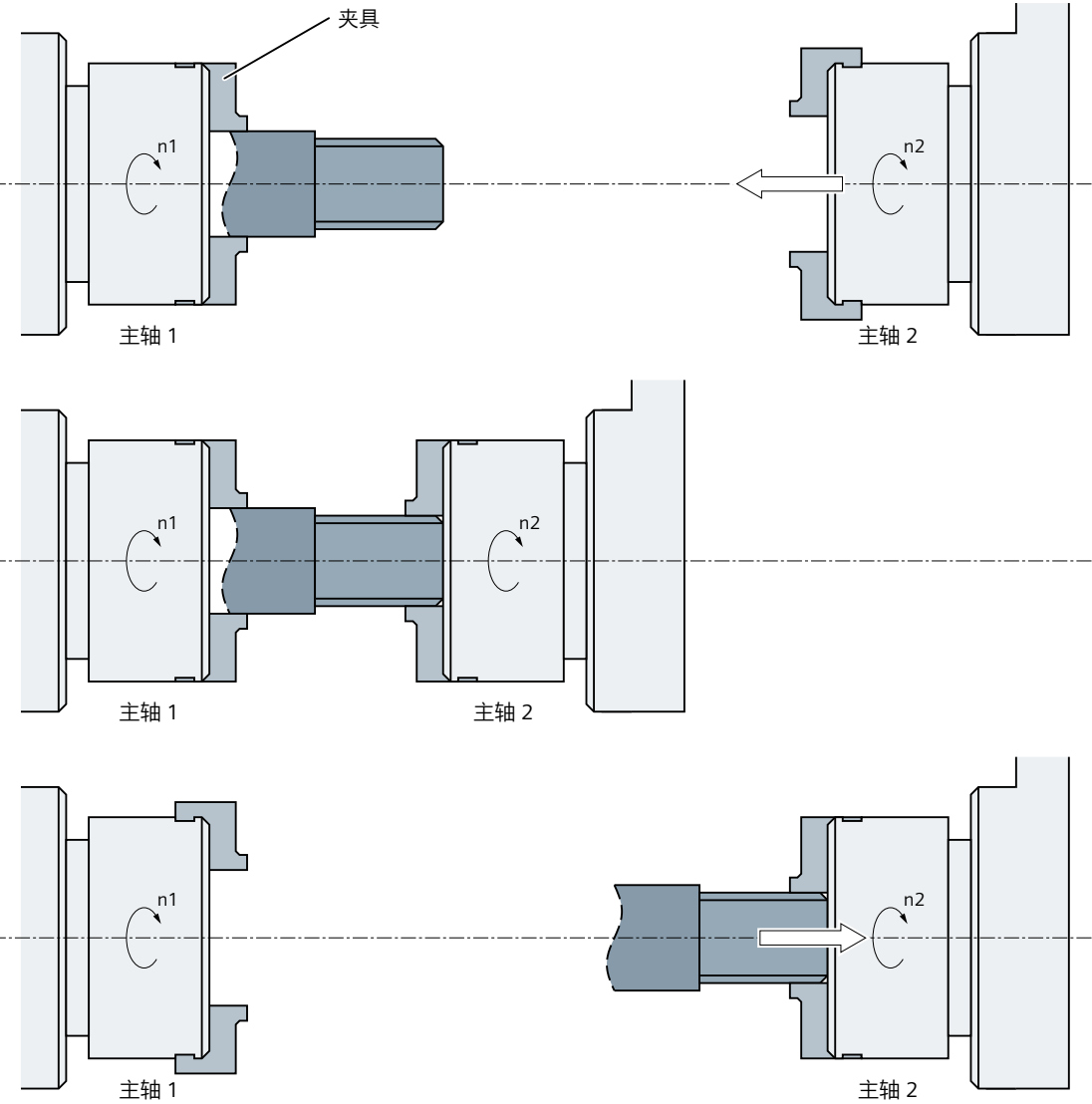


图 14-1 主轴同步：工件从主轴 1 快速传送至主轴 2

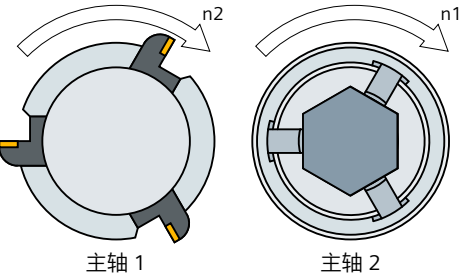


图 14-2 主轴同步：多面体车削

14.1.2 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, ..., 对应 MD35000 \$MA_SPIND_ASSIGN_TO_MACHAX 中定义的主轴号
(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：

接口信号“同步运行”(<Axis>.spindle.in.syncModeActive) = 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
($n_{\text{副主轴}} = n_{\text{主主轴}} ; k_{\text{传动比}} = 1$)
- 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“通过 PLC 控制主轴同步 (页 531)”章节）。

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)。
采用主主轴和副主轴时，在每个 COUP... 语言指令中均需要编写主主轴和副主轴的主轴名称 (Sn, SPI(n))。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 会通过对机床轴的接口信号报告 PLC，这些机床轴是否作为副主轴 (<Axis>.coupling.in.syncSpindleSlaveActive) 和主主轴 (<Axis>.coupling.in.syncSpindleMasterActive) 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。

同步主轴耦合的更多编程语句参见章节“编程(页 537)”。

同步特性

可为每个主轴同步对设置以下特性：

- **程序段转换特性**

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。

- 立即进行程序段转换
- 在满足“精同步”条件时进行程序段转换
- 在满足“粗同步”条件时进行程序段转换
- 在满足 IPOSTOP 条件时进行程序段转换(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“配置(页 545)”章节），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“定义(COUPDEF)(页 537)”章节）。

此外,在固定定义的同步中，同步方式和程序段转换特性还可通过语言指令 COUPDEF 修改。

更多信息：NC 编程手册

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转换特性）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“通过零件程序来激活主轴同步 (页 528)”章节）。
- 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”

(<Axis>.coupling.in.syncSpindleMotionOverlaid) 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。在用 SPOS 修改偏移时，可以用 FA[Sn] 来指定定位速度，并借助倍率（可通过接口信号“主轴进给倍率生效”

<Axis>.spindle.out.useFeedrateOvr 选择）对其进行调整。

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：

更多信息

定位模式 (页 403)

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°

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.spindle.out.useFeedrateOvr	LBP_Axis*.A_FD_ORSp	DB31, ...DBX17.0

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.spindle.in.syncModeActive	LBP_Axis*.E_SyncMode	DB31, ...DBX84.4
<Axis>.coupling.in.MotionOverlaid	LBP_Axis*.E_OverlayMotion	DB31, ...DBX98.4
<Axis>.coupling.in.leadingSpindleActive	LBP_Axis*.E_MasterSp	DB31, ...DBX99.0
<Axis>.coupling.in.followingSpindleActive	LBP_Axis*.E_SlaveSp	DB31, ...DBX99.1

14.1.3 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。该限制会通过接口信号“设定转速已限制”(`<Axis>.spindle.in.speedSetpointAboveMax`) 进行报告。

更多信息

主轴 (页 395)

- 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。

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.spindle.in.speedSetpointAboveMax	LBP_Axis*.E_SetSpeedLimit	DB31, ...DBX83.1

参见

回参考点 (页 35)

14.1.4 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“主轴同步的前提条件 (页 527)”章节）。

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}）

程序段转换特性

激活主轴同步之前，应确定使用哪种程序段转换条件（参见“定义 (COUPDEF) (页 537)”章节）。

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“主轴同步的轴系统变量(页 542)”章节）。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”(<Axis>.coupling.in.syncSpindleMotionOverlaid) 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非接触式接近开关信号得出，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。

_{副主轴}位置的范围：0 ... 359.999°。

更多信息

回参考点 (页 35)

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轴的当前位置偏移。其中可分为：

- 副主轴与主轴的设定值位置偏移
\$AA_COUP_OFFS [<副主轴名称>]
- 副主轴与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 521)”)。

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（接口信号“已回参考点/已同步 1 或 2” <Axis>.basic.in.enc1Synchronized 或 <Axis>.basic.in.enc2Synchronized=0）时激活同步运行。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。

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.enc1Synchronized	LBP_Axis*.E_RefSyn1	DB31, ...DBX60.4
<Axis>.basic.in.enc2Synchronized	LBP_Axis*.E_RefSyn2	DB31, ...DBX60.5
<Axis>.coupling.in.MotionOverlaid	LBP_Axis*.E_OverlayMotion	DB31, ...DBX98.4

14.1.5 通过零件程序来撤销主轴同步

使用零件程序指令 COUPOF 撤销指定主轴之间的同步。

- COUPOF - 在不停止副主轴运行的情况下撤销主轴同步

COUPOF (<FS>, <LS>)

如果使用 COUPOF 撤销指定主轴之间的同步，则不论该同步是固定定义还是用户定义的都不重要。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。

如果使用 COUPOF 在运行过程中撤销主轴同步，则副主轴会以当前的转速继续旋转 (n_{副主轴})。使用系统变量 \$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。

使用 COUPOF 撤销主轴同步时，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，系统内部自动生成一个预处理停止指令，功能与 STOPRE 类似。

- COUPOFS - 在停止副主轴运行的情况下撤销主轴同步

COUPOFS (<FS>, <LS>)

参见

回参考点 (页 35)

14.1.6 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。

这些信号对主轴没有任何的作用。可以使用以下同步专用的 VDI 信号 (PLC → NC)：

接口信号 **“禁用同步”** (<Axis>.coupling.out.syncSpindleSyncDisable)

“禁用同步”

使用轴接口信号 **“禁用同步”** (<Axis>.coupling.out.syncSpindleSyncDisable) 可以抑制副主轴的同步动作。

当包含 COUPON (副主轴, 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:

接口信号 **“禁用同步”** (<Axis>.coupling.out.syncSpindleSyncDisable)。

- 当接口信号 **“禁用同步”** (<Axis>.coupling.out.syncSpindleSyncDisable) = 0 时, 位置偏移如同之前一样生效。
- 当接口信号 **“禁用同步”** (<Axis>.coupling.out.syncSpindleSyncDisable) = 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>, <主轴>) 类似。

特点

接口信号 **“禁用同步”** (<Axis>.coupling.out.syncSpindleSyncDisable) 不影响通过以下方式生成的副主轴偏移运动:

- SPOS、POS
- 同步动作
- SINU_RotateSpindle (Basic Program FC18)

这些功能可通过 VDI 接口信号 **“进给停止/主轴停止”** (<Axis>.basic.out.feedStopSpindleStop) 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：

接口信号 “粗同步”(<Axis>.coupling.in.syncSpindleCoarseReached) 和

接口信号 “精同步”(<Axis>.coupling.in.syncSpindleFineReached)

虽然同步被抑制，但 COUPON 这个指令后续的程序段转换不会被阻止。

示例

COUPON 指令后续程序段转换的特性

程序代码	注释
	; 接口信号 “禁用同步”
	;
	(<Axis>.coupling.out.syncSpindleSyncDisable)
	= 1, 针对 S2 置位
N51 SPOS=10 SPOS[2]=10	; 定位，对应
	; 0° 的偏移
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”
	;
	(<Axis>.coupling.in.syncSpindleCoarseReached)
	和
	; 接口信号 “精同步”
	; (<Axis>.coupling.in.syncSpindleFineReached)
	; 置位，且程序段转换
	; 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”(<Axis>.coupling.out.syncSpindleSyncDisable)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”(<Axis>.coupling.out.syncSpindleSyncDisable)=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”(<Axis>.coupling.out.syncSpindleResync)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。

说明	NC 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”(<Axis>.basic.out.feedStopSpindleStop) 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：

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：

更多信息

回转轴 (页 315)

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.feedStopSpindleStop	LBP_Axis*.A_FDSpStop	DB31, ...DBX4.3
<Axis>.coupling.out.syncSpindleResync	LBP_Axis*.A_Resynchronize	DB31, ...DBX31.4
<Axis>.coupling.out.syncSpindleSyncDisable	LBP_Axis*.A_DisSync	DB31, ...DBX31.5

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.coupling.in.syncFineReached	LBP_Axis*.E_SyncronFine	DB31, ...DBX98.0
<Axis>.coupling.in.syncCoarseReached	LBP_Axis*.E_SyncronCoarse	DB31, ...DBX98.1

14.1.7 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”(<Axis>.coupling.in.syncFineReached) 或“粗同步”(<Axis>.coupling.in.syncCoarseReached) 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

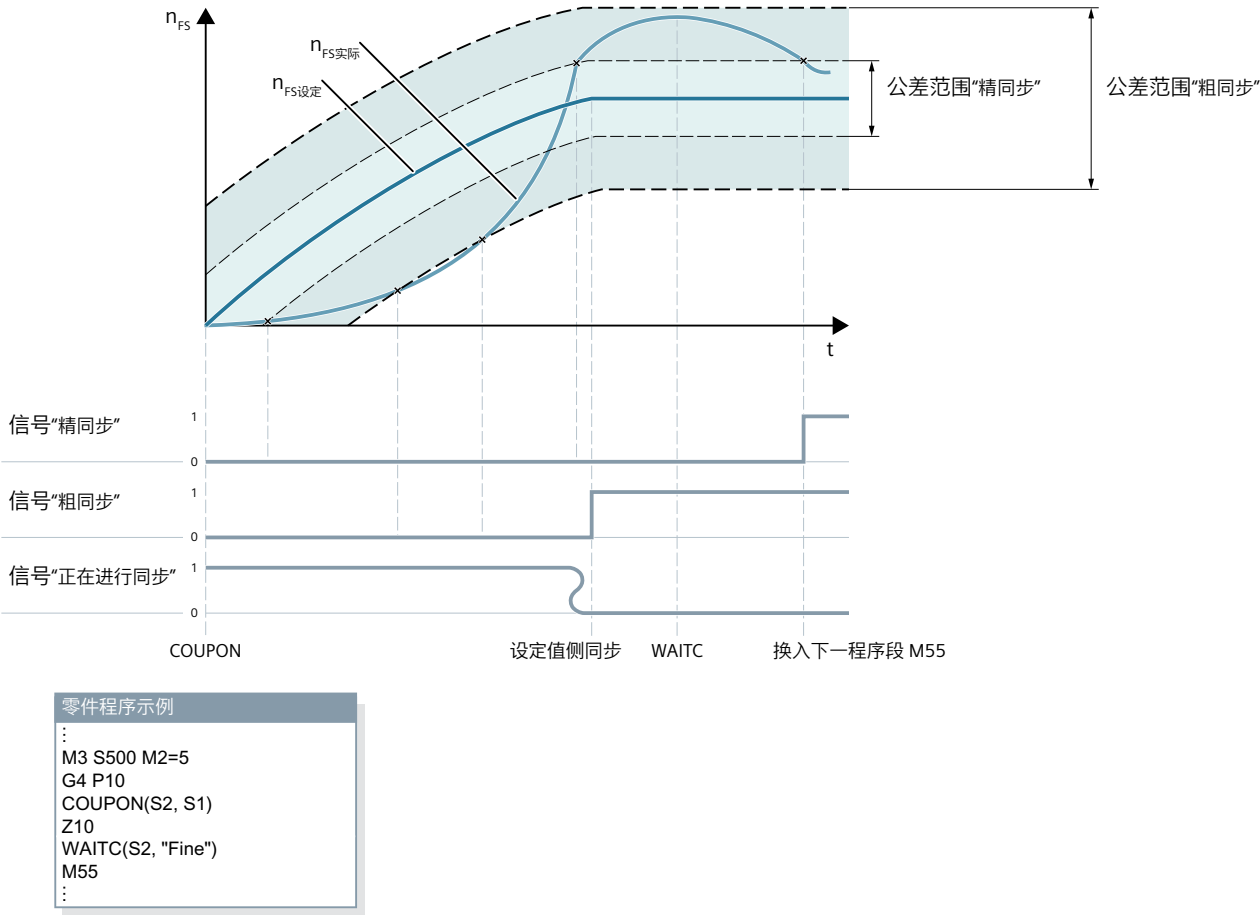


图 14-3 利用 COUPON 和 WAITC 等待标记实现主轴监控，副主轴要与正在旋转的主主轴同步

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，副主轴会跟随主轴旋转，且不会超出自身的限值。

比如主轴会自动减速，使得副主轴不会超出自身的最大转速，从而确保同步生效。

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.coupling.in.syncFineReached	LBP_Axis*.E_SyncronFine	DB31, ...DBX98.0
<Axis>.coupling.in.syncCoarseReached	LBP_Axis*.E_SyncronCoarse	DB31, ...DBX98.1

14.1.8 同步主轴的动态修调

加速度修调 ACCLIMA / ACC

副主轴适用 ACCLIMA（1% 至 200%）。

ACCLIMA 在主轴模式中对主轴的整个运动起效，即使在未耦合状态下。

速度修调 VELOLIMA

副主轴适用 VELOLIMA（1% 至 200%），在同步主轴上限制为 100%。

在主轴模式中，VELOLIMA 在 100% 时被切断，对主轴的整个运动起效，即使在未耦合状态下。

基本运动不需要 VELOLIMA 编程。这可以通过 VELOLIM 命令进行编程。

有效性（零件程序、同步动作...）

动态修调值 ACCLIMA、VELOLIMA、JERKLIMA 可以在零件程序或同步动作中编程。无论是在零件程序中还是同步动作中，最后的编程都是有效的。

14.2 编程

带主轴编程的同步指令

指令	功能
COUPDEF (<FS>, <LS>, ...)	定义同步或修改已定义的同步
COUPON (<FS>, <LS>, <POSFS>)	激活同步
COUPONC (<FS>, <LS>)	激活同步并接收当前生效的副主轴转速
COUPOF (<FS>, <LS>)	撤销同步
COUPOFS (<FS>, <LS>)	撤销同步并停止副主轴
COUPDEL (<FS>, <LS>)	删除同步
COUPRES (<FS>, <LS>)	重新激活定义的同步数据

不带主轴编程的同步指令

指令	功能
COUPOF (<FS>)	撤销同步
COUPOFS (<FS>)	撤销同步并停止副主轴
COUPDEL (<FS>)	删除同步
COUPRES (<FS>)	重新激活定义的同步数据

参见

定义 (COUPDEF) (页 537)

激活和撤销同步 (COUPON, COUPONC, COUPOF, COUPOFS) (页 541)

14.2.1 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。可用的轴数由总数减去一得到。与以前一样，也可以通过机床数据来定义同步主轴对。

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 可以创建新的同步主轴对（用户定义），并将现有的同步主轴对修改掉。

完整的同步参数的编程方式为：

```
COUPDEF (<FS>, <LS>, <TRNom>,  
[<TRDenom>], <BlockChange>, <CouplingType>)
```

使用 <FS> 和 <LS> 明确确定同步主轴对。

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：

- <FS>, <LS>：副主轴和主主轴的名称
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。
- <TRNom>, [<TRDenom>]：传动比参数（分子和分母）
传动比用分子和分母（传动参数）来规定（参见“主轴同步(页 521)”章节）。
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”。

- <BlockChange>

它可用于激活主轴同步时选择何时进行程序段转换：

- “NOC”：立刻使能程序段转换
- “FINE”：在满足“精同步”条件时进行程序段转换
- “COARSE”：在满足“粗同步”条件时进行程序段转换
- “IPOSTOP”：在满足 IPOSTOP 条件时进行程序段转换（即：在设定值同步后）

程序段转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。

如果未指定程序段转换特性，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。

- <CouplingType>

通过此参数设定同步方式：

- “DV” (Desired Values)：副主轴和主主轴之间的设定值同步
- “AV” (Actual Values)：副主轴和主主轴之间的实际值同步
- “VV” (Velocity Values)：副主轴和主主轴之间的速度同步

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：

- <TRNom>=1.0
- <TRDenom>=1.0
- <BlockChange> = "IPOSTOP"（在设定值达到同步后进行程序段转换）
- <CouplingType> = "DV"（设定值同步）

删除同步

使用语言指令 COUPDEL 删除用户自定义设置的同步。

```
COUPDEL (FS, LS)
```

说明

使用 COUPDEL 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。相当于 COUPOF(FS, LS) 的行为。

激活初始同步参数

使用语言指令 COUPRES 可以再次激活由机床数据定义的同步参数。

```
COUPRES (FS, LS)
```

此时，使用 COUPDEF 更改的参数（包括传动比）丢失。

使用 COUPRES 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（即用户自定义设置的同步）。

可编程的程序段转换

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“[监控主轴同步 \(页 534\)](#)”章节）。

```
WAITC (FS)
```

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")
```

停止及程序段转换

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。

使用程序指令 COUPON 和 WAITC 可以控制程序段转换。同时，使用 COUPDEF 或通过 MD21320 \$MC_COUPLE_BLOCK_CHANGE_CTRL_1 来确定程序段转换标准。

14.2.2 激活和撤销同步 (COUPON, COUPONC, COUPOF, COUPOFS)

激活主轴同步

使用语言指令 COUPON 激活主轴同步。

激活主轴同步有三种方式可选：

- COUPON (<FS>, <LS>)
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。
- COUPON (<FS>, <LS>, <POSFS>)
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 (POSFS)。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转换。
副主轴位置的范围：0 ... 359.999 度。
- COUPONC (<FS>, <LS>)
在使用 COUPONC 激活同步时，主轴会继续采用之前编程的 M3 S... 或 M4 S...。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。

更改角度偏移

在主轴同步已经激活时编程 COUPON(<FS>,<LS>,<POSFS>) 或 SPOS 可以更改副主轴与主主轴之间的角度偏移。

撤销主轴同步

撤销同步及主轴同步有两种方式可选：

- COUPOF (FS, LS)

立即使能程序段转换。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)
```

```
...
```

```
COUPDEL (S2, S1)
```

- COUPOFS (FS, LS)

使用 COUPOFS 附加停止副主轴运行。

14.2.3 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“主轴同步 (页 521)”章节）。

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]
```

如果使用 COUPON 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 <Axis>.coupling.out.syncSpindleResync（重新同步）可再次恢复最初写入的偏移。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.coupling.out.syncSpindleResync	LBP_Axis*.A_Resynchronize	DB31, ...DBX31.4

14.2.4 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）同步方式下，程序指令 COUPON、COUPONC 以及 COUPOF、COUPOFS 会在必要情况下自动激活或取消主主轴的位置闭环控制。如果一根主主轴有多根副主轴，**第一个 DV（设定值）同步激活主主轴的位置闭环控制，最后一个 DV（设定值）同步取消主主轴的位置闭环控制**，条件是没有编程 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	无动作
主主轴	开启位置闭环控制 ¹	无动作	无动作

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	位置闭环控制取消 ¹⁾	位置闭环控制取消 ¹⁾	无动作
主主轴	位置闭环控制取消 ²⁾	无动作	无动作

¹⁾ 激活副主轴转速开环模式。如果使用 SPCON 使副主轴进入位置闭环控制主轴运行模式，则位置闭环控制**不会自动取消**。

²⁾ 如果该主主轴没有其他的副主轴和它进行 DV（设定值）同步，则 COUPOF 自动**取消**位置闭环控制。如果主主轴位于定位模式或进给轴模式中**或是**使用 SPCON 使主主轴进入位置闭环控制主轴模式，则位置闭环控制**继续存在**。

14.3 配置

说明

每个通道都可以定义一个同步主轴对。

表格 14-1 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		

14.3 配置

表格 14-2 设定数据

编号	名称: \$SC_	功能
SD42300	COUPLE_RATIO_1[<n>]	转速传动比: 副主轴/主轴 = 分子/分母 ¹⁾ : • <n> = 0: 分子 (副主轴) • <n> = 1: 分母 (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		

14.3.1 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

14.3.2 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_MASK
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

14.4 特殊性

14.4.1 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“主轴同步功能调试的特殊性 (页 563)”章节）。

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。

前馈控制方式（转速还是转矩前馈控制）可以通过轴专用的 MD32620 \$MA_FFW_MODE 来确定（另见功能手册“监控与补偿”，章节“K3：补偿”）。

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。

更多信息

功能手册之基本功能，运行方式组、通道、程序运行、复位特性

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。

- `<Axis>.basic.out.enableCtrl = 0`（控制器使能）
- `<Axis>.basic.out.followUpMode = 1`（跟随模式）

（参见章节“主轴同步和 NC/PLC 接口信号 (页 552)”）

通过 NC/PLC 接口信号重新进行同步：

`<Axis>.coupling.out.syncSpindleResync = 0 → 1`（重新同步）

可再次恢复已写入的偏移（参见“再次恢复副主轴和主主轴的同步 (页 550)”章节）。

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)。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<code><Axis>.basic.out.followUpMode</code>	<code>LBP_Axis*.A_Follow_upMode</code>	<code>DB31, ...DBX1.4</code>
<code><Axis>.basic.out.enableCtrl</code>	<code>LBP_Axis*.A_ContrEnable</code>	<code>DB31, ...DBX2.1</code>
<code><Axis>.coupling.out.syncSpindleResync</code>	<code>LBP_Axis*.A_Resynchronize</code>	<code>DB31, ...DBX31.4</code>

参见

主轴 (页 395)

14.4.2 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):
`<Axis>.basic.in.followUpModeActive`（跟随模式生效）= 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:
`<Axis>.coupling.out.gantryDisableSync`（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：
`<Axis>.coupling.out.syncSpindleResync`（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 NC/PLC 接口信号置位：

`<Axis>.coupling.in.syncFineReached`（粗同步）

`<Axis>.coupling.in.syncCoarseReached`（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:
`<Axis>.coupling.out.syncSpindleSyncDisable`（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：

<Axis>.coupling.out.syncSpindleResync（重新同步）

NC 输出以下 NC/PLC 接口信号报告发现了上升沿：

<Axis>.coupling.in.syncSpindleProcessActive（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 <Axis>.coupling.out.syncSpindleResync（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。

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，见上）
	; 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，见上）

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 <Axis>.coupling.out.resync（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 <Axis>.coupling.out.syncSpindleResync（重新同步）才会生效。

14.4 特殊性

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。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.coupling.out.gantryDisableSync	LBP_Axis*.A_NoAutoSync	DB31, ...DBX29.5
<Axis>.coupling.out.syncSpindleResync	LBP_Axis*.A_Resynchronize	DB31, ...DBX31.4
<Axis>.coupling.out.syncSpindleSyncDisable	LBP_Axis*.A_DisSync	DB31, ...DBX31.5

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.followUpModeActive	LBP_Axis*.E_Follow_upMode	DB31, ...DBX61.3
<Axis>.coupling.in.syncFineReached	LBP_Axis*.E_SyncronFine	DB31, ...DBX98.0
<Axis>.coupling.in.syncCoarseReached	LBP_Axis*.E_SyncronCoarse	DB31, ...DBX98.1
<Axis>.coupling.in.ProcessActive	LBP_Axis*.E_Sync	DB31, ...DBX99.4

14.4.3 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 (<Axis>.spindle.out.ovrFactor)

在主轴同步中，只有主主轴的主轴倍率值生效。

主轴禁止 (<Axis>.basic.out.disable)

主主轴	副主轴	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

主主轴	副主轴	同步	特性
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 (<Axis>.basic.out.enableCtrl)

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 <Axis>.basic.out.followUpMode == <值>（跟随模式）：

<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 <Axis>.basic.out.enableCtrl（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 `<Axis>.basic.out.enableCtrl`（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：`<Axis>.coupling.out.syncSpindleResync = 1`（重新同步）。

跟随模式 (`<Axis>.basic.out.followUpMode`)

跟踪运行中会循环将设定位置设为当前实际位置：

`<Axis>.basic.out.followUpMode == 1`（跟随模式）和
`<Axis>.basic.out.enableCtrl == 0`（伺服使能）

⇒ 循环：设定位置 = 实际位置

说明

`<Axis>.basic.out.followUpMode`（跟随模式）只
与 `<Axis>.basic.out.enableCtrl == 0`（伺服使能）相关

位置测量系统 1/2（`<Axis>.basic.out.enc1Activation` 或 `<Axis>.basic.out.enc2Activation`）

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 (`<Axis>.basic.out.resetMovement`)

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）(<Axis>.basic.out.feedStopSpindleStop)

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: <Axis>.basic.out.feedStopSpindleStop = 1（主轴停止）
2. 等待停止: <Axis>.basic.in.isStationary == 1
3. 停止: <Axis>.basic.out.enableCtrl = 0（伺服使能）

删除 S 值 (<Axis>.spindle.out.deleteSValue)

主主轴：在同步运行时删除 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。

FS:在同步运行时删除 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 (<Axis>.spindle.out.resyncEnc1 和 <Axis>.spindle.out.resyncEnc2)

主主轴：在同步运行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 (<Axis>.coupling.out.syncSpindleResync)

FS: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: <Axis>.coupling.out.syncSpindleResync = 1 (重新同步)
- 应答: <Axis>.coupling.in.syncSpindleProcessActive == 1 (正在进行同步)

JOG 模式下的移动键 (<Axis>.basic.out.traversKeyMinus 和 <Axis>.basic.out.traversKeyPlus)

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 (<Chan>.basic.out.ncStopAxesAndSpindles)

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 (<Chan>.basic.out.ncStart)

(参见章节“NC 启动时同步主轴对的特性 (页 546)”)

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。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.disable	LBP_Axis*.A_ASpDisable	DB31, ...DBX1.3
<Axis>.basic.out.followUpMode	LBP_Axis*.A_Follow_upMode	DB31, ...DBX1.4
<Axis>.basic.out.enc1Activation	LBP_Axis*.A_PosMeas1	DB31, ...DBX1.5

Basic Program Plus	Basic Program	
<Axis>.basic.out.enc2Activation	LBP_Axis*.A_PosMeas2	DB31, ...DBX1.6
<Axis>.basic.out.enableCtrl	LBP_Axis*.A_ContrEnable	DB31, ...DBX2.1
<Axis>.basic.out.resetMovement	LBP_Axis*.A_DeIDTGSpReset	DB31, ...DBX2.2
<Axis>.basic.out.feedStopSpindleStop	LBP_Axis*.A_FDSpStop	DB31, ...DBX4.3
<Axis>.basic.out.traversKeyMinus	LBP_Axis*.A_Minus	DB31, ...DBX4.6
<Axis>.basic.out.traversKeyPlus	LBP_Axis*.A_Plus	DB31, ...DBX4.7
<Chan>.basic.out.ncStart	LBP_Chan*.A_NCStart	DB21, ...DBX7.1
<Chan>.basic.out.ncStopAxesAndSpindles	LBP_Chan*.A_NCStopASp	DB21, ...DBX7.4
<Axis>.spindle.out.resyncEnc1	LBP_Axis*.A_Sp1Syn	DB31, ...DBX16.4
<Axis>.spindle.out.resyncEnc2	LBP_Axis*.A_Sp2Syn	DB31, ...DBX16.5
<Axis>.spindle.out.deleteSValue	LBP_Axis*.A_DeleteS	DB31, ...DBX16.7
<Axis>.spindle.out.ovrFactor	LBP_Axis*.A_SpOR	DB31, ...DBB19.0
<Axis>.coupling.out.syncSpindleResync	LBP_Axis*.A_Resynchronize	DB31, ...DBX31.4
<Axis>.coupling.out.syncSpindleSyncDisable	LBP_Axis*.A_DisSync	DB31, ...DBX31.5

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.isStationary	LBP_Axis*.E_Stat	DB31, ...DBX61.4
<Axis>.coupling.in.syncFineReached	LBP_Axis*.E_SyncronFine	DB31, ...DBX98.0
<Axis>.coupling.in.syncCoarseReached	LBP_Axis*.E_SyncronCoarse	DB31, ...DBX98.1
<Axis>.coupling.in.ProcessActive	LBP_Axis*.E_Sync	DB31, ...DBX99.4

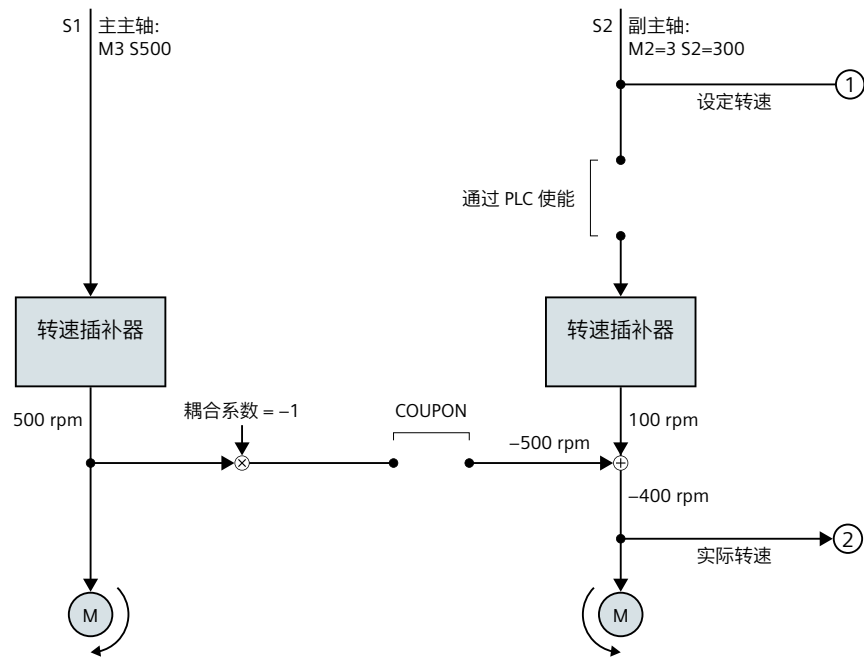
14.4.4 主轴和副主轴之间的差速

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。根据两个转速源之和（带符号），通过主轴的耦合系数可导出一个转速分量。此外还可为副主轴编程设置转速(S...)和旋转方向(M3/M4)。

通常耦合系数设为“-1”即可保持同步。对该系数取反则可为副主轴生成转速差，而无需进行额外的转速编程。

NC 的典型工作方式会在下图中详细说明。



示例

程序代码	注释
N01 M3 S500	; S1 以 500 转/分钟正转。
	; 主主轴是主轴 1。
N02 M2=3 S2=300	; S2 以 300 转/分钟正转。
N05 G4 F1	
N10 COUPDEF(S2,S1,-1)	; 耦合系数 = -1:1
N11 COUPON(S2,S1)	; 激活耦合。副主轴 S2 的转速由主主轴 S1 的转速和耦合系数得出。
N26 M2=3 S2=100	; 差速编程, S2 为副主轴。

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。

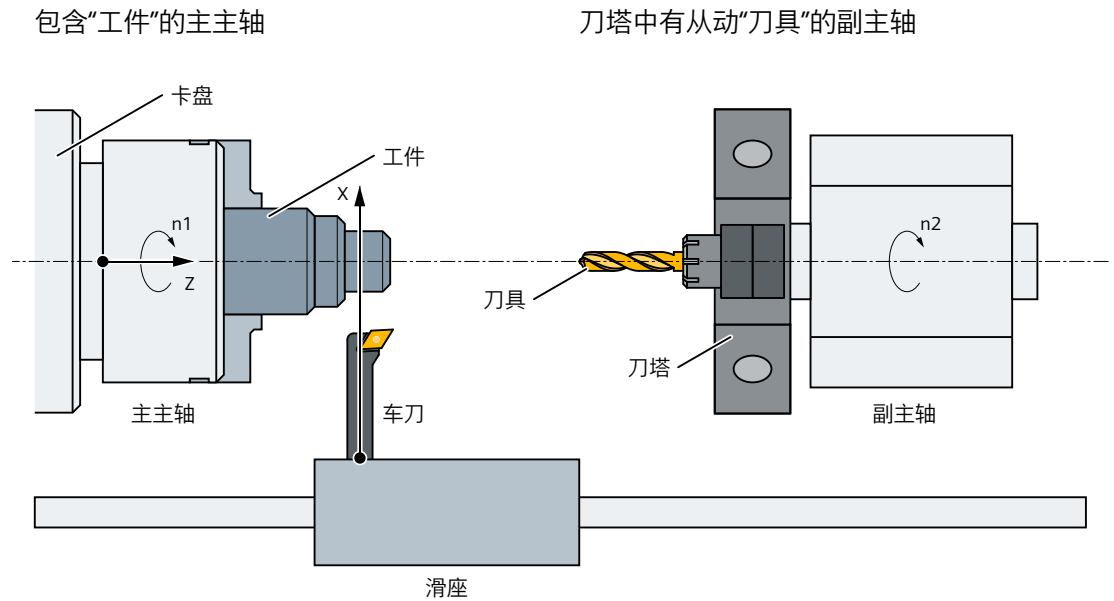


图 14-4 单滑块车床，刀塔围绕 Z 轴旋转

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” (`<Axis>.coupling.out.enableFollowOverlay`) 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” (`<Axis>.coupling.out.enableFollowOverlay`) 或结束耦合来清除。

说明

定位运动如 SPOS、SPOSA、M19 以及轴运动无需使能。

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。

关闭耦合

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 COUPONC 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 $500 \text{ rpm} * (-1) + 100 \text{ rpm} = -400 \text{ rpm}$ 。

接口信号：NC 至 PLC

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），NC 会置位副主轴的接口信号“主轴处于设定区域”(<Axis>.spindle.in.speedSetpointReached)。虽然写入了差速，但是若 PLC 未使能差速，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 NC → PLC “叠加运动” (`<Axis>.coupling.in.syncSpindleMotionOverlaid`) 置位。

实际旋转方向顺时针 (`<Axis>.spindle.in.cwRotActive`)

接口信号 “实际旋转方向顺时针” (`<Axis>.spindle.in.cwRotActive`) 指的是当前的电机旋转方向。

接口信号：PLC 至 NC

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 (`<Axis>.basic.out.resetMovement`)

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” (`<Axis>.basic.out.resetMovement`) 删除。置位接口信号 “删除 S 值” (`<Axis>.spindle.out.deleteSValue`) 可删除已写入的转速。

重新同步主轴 1/2 (`<Axis>.spindle.out.resyncEnc1` 和 `<Axis>.spindle.out.resyncEnc2`)

接口信号 “重新同步主轴 1/2” (`<Axis>.spindle.out.resyncEnc1/ <Axis>.spindle.out.resyncEnc2`) 未被禁用。但主轴同步不自动补偿产生的偏移。

M3/M4 反向 (`<Axis>.spindle.out.invertM3M4`)

接口信号 “M3/M4 反向” (`<Axis>.spindle.out.invertM3M4`) 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 (<Axis>.spindle.out.ovrFactor)

VDI 接口 “主轴倍率” (<Axis>.spindle.out.ovrFactor) 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到所有的轴输入端时，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。

耦合取消

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。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.resetMovement	LBP_Axis*.A_DeIDTGSReset	DB31, ...DBX2.2
<Axis>.spindle.out.resyncEnc1	LBP_Axis*.A_Sp1Syn	DB31, ...DBX16.4
<Axis>.spindle.out.resyncEnc2	LBP_Axis*.A_Sp2Syn	DB31, ...DBX16.5
<Axis>.spindle.out.deleteSValue	LBP_Axis*.A_DeleteS	DB31, ...DBX16.7
<Axis>.spindle.out.invertM3M4	LBP_Axis*.A_M3M4Inv	DB31, ...DBX17.6
<Axis>.spindle.out.ovrFactor	LBP_Axis*.A_SpOR	DB31, ...DBB19.0
<Axis>.coupling.out.enableFollowOverlay	LBP_Axis*.A_FollowAxOv	DB31, ...DBX26.4

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.isStationary	LBP_Axis*.E_Stat	DB31, ...DBX61.4
<Axis>.spindle.in.speedSetpointReached	LBP_Axis*.E_SetRange	DB31, ...DBX83.5
<Axis>.spindle.in.cwRotActive	LBP_Axis*.E_ActRotRight	DB31, ...DBX83.7
<Axis>.coupling.in.MotionOverlaid	LBP_Axis*.E_OverlayMotion	DB31, ...DBX98.4

14.4.5 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。

14.4.6 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。

14.4.7 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：

更多信息

CNC 调试手册：NC、PLC、驱动

主轴 (页 395)

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**(另见功能手册“监控与补偿”,章节“K3: 补偿”)。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[<主主轴>] = 5 ms
- 副主轴的等效时间常数：MD32810 \$MA_EQUIV_SPEEDCTRL_TIME[<副主轴>] = 3 ms
- 副主轴动态响应自适应功能的时间常数：MD32910 \$MA_DYN_MATCH_TIME[<副主轴>] = 5 ms - 3 ms = 2 ms
- 激活副主轴动态响应自适应功能：MD32900 \$MA_DYN_MATCH_ENABLE[<副主轴>] = 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”>“服务轴”

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）

FS: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：同步主轴耦合，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：同步主轴耦合：

控制参数的设置和在未激活主轴同步的状态下的设置一样。

更多信息

速度、设定值/实际值系统、闭环控制 (页 141)

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。

更多信息

加速度和急动度 (页 91)，折线式加速度特性曲线

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/或主主轴不需要位置闭环控制。不需要测量系统。

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

更多信息

回参考点 (页 35)

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）

更多信息

操作手册

14.5 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参见：

更多信息

轴耦合 (页 571)

14.6 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
	; 定义
N70 SPCON	; 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

14.6 示例

程序代码	注释
N360 G0 X0 Y0	
N365 WAITS (2)	; 等待主轴 2
N370 M5	; 停止副主轴
N375 M30	

轴耦合

15.1 联动

15.1.1 简要说明

15.1.1.1 功能

“联动”功能可以定义从引导轴到跟随轴的简单轴耦合，同时考虑耦合系数。

联动功能具备以下属性：

- NC 的每个轴都可以定义为引导轴。
- NC 的每个轴都可以定义为具有特定耦合系数的联动轴。
- 引导轴和联动轴构成联动组。
- 一个引导轴可以分配有任意数量的联动轴。
- 一个联动轴最多可分配有 2 个引导轴。
- 一个联动轴可以是另一个联动组的引导轴。
- 引导轴的运行由所有跟随轴同步执行，用耦合系数评估。
- 耦合生效时，联动轴也可以独立于引导轴执行运行（叠加运行）。
- 联动组引导轴和联动轴的定义以及耦合的启用/关闭是通过零件程序或同步动作中的零件程序指令进行的。
- 在 JOG、JOG REF、JOG INC 等手动操作模式下也可以进行联动。

15.1.1.2 前提条件

“联动”功能

“联动”功能是 NC 软件的固定组成部分。

通用耦合

联动功能也可用于通用耦合。

15.1 联动

但通用耦合的**基本类型**有以下限制：

- 联动组的最大数量限制为 4 个。
- 一个联动轴只能分配有 1 个引导轴。
- 无法级联。

说明

如果 NC 软件配备有通用耦合的相应可选类型，则这些限制不适用（参见通用耦合“简要说明”中的主题“前提条件”）。

15.1.2 一般功能

“联动”功能可以定义简单的轴耦合。耦合从一个引导轴到一个或多个跟随轴进行，即所谓的联动轴。可以为每个联动轴指定自己的耦合系数。

联动组

引导轴和分配给它的所有联动轴共同构成一个联动组。如果引导轴运行，则所有联动轴根据其耦合系数运行。

联动组可为线性轴和回转轴的任意组合。

引导轴

NC 的每个轴，包括模拟轴，都可以用作引导轴。

联动轴

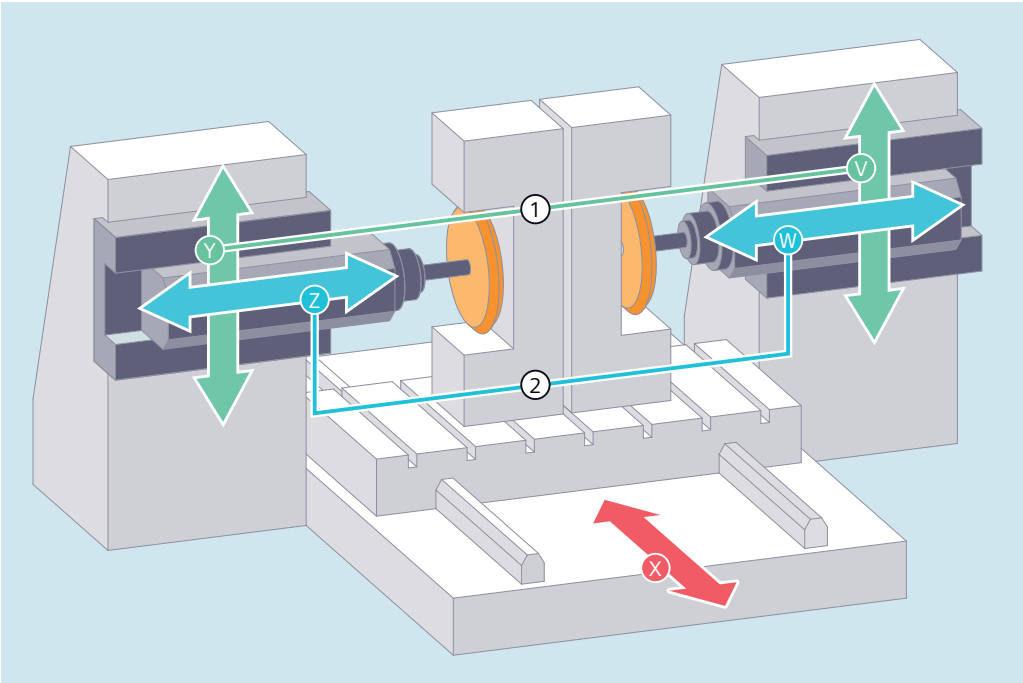
NC 的每个轴都可以用作联动轴。

耦合系数

联动轴相对于引导轴的运动比率通过耦合系数指定：

耦合系数 $K = \text{联动轴的运动} / \text{引导轴的运动}$

负耦合系数（反向运行联动轴）也是允许的。



	引导轴	联动轴	耦合系数
联动组 ①	Y 轴	V 轴	1
联动组 ②	Z 轴	W 轴	-1

图 15-1 联动：双面加工

复合同步

一个联动轴最多可分配有 2 个引导轴。联动轴的运行是引导轴的运行总和。

相关联动轴

如果一个联动轴由于引导轴运动而运行，则它是“相关联动轴”。

独立联动轴

如果一个联动轴根据直接运行指令运行，则它是“独立联动轴”。产生的联动轴的运行是“相关”和“独立”联动轴的运行总和。

联动轴作为引导轴

一个联动轴可以同时是另一个联动组的引导轴。

坐标系

联动始终在基准坐标系 (BCS) 中进行。

启用/关闭

联动同样可以通过零件程序和同步动作来启用和关闭。此时应注意，启用和关闭是通过相同的编程进行的：

- 接通：零件程序 → 关闭：零件程序
- 接通：同步动作 → 关闭：同步动作

急速同步

如果在引导轴运动时启用，则联动轴首先会加速到与耦合对应的速度。在引导轴和联动轴的速度同步的时间点，引导轴的位置是进一步联动的起始位置。

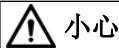
运行方式

联动在 AUTO、MDA 和 JOG 运行方式下生效。

回参考点运行

一个联动组的轴回参考点运行时：

- 引导轴
一个联动组的引导轴回参考点运行时，与所有联动轴的耦合保留。联动轴根据其耦合系数与引导轴同步运行。
- 联动轴：运行方式 JOG/REF
一个联动组的联动轴回参考点运行时，与引导轴的耦合将取消。耦合的取消通过以下报警显示：
报警 16772 “通道 通道编号程序段 程序段编号轴 轴编号为跟随轴，耦合将打开。”
只有在退出 JOG/REF 运行方式后，耦合才会再次激活。
可通过以下机床数据来抑制报警显示：
MD11410 \$MN_SUPPRESS_ALARM_MASK, 位 29 = 1（抑制专用报警输出的界面）



小心

无耦合

随着联动轴的回参考点运行，与引导轴的耦合将取消。如果现在立即对引导轴进行回参考点运行，即不切换 JOG/REF 运行方式，则联动轴不随引导轴运行。

- 联动轴：零件程序指令 G74
无法通过零件程序指令 G74 对联动组的联动轴进行回参考点运行。

剩余行程：联动轴

联动轴的剩余行程是指从相关和独立运行得出的要运行的总剩余行程。

删除剩余行程：联动轴

删除联动轴的剩余行程只导致独立于引导轴的运行中断。

NC 启动时的特性

NC 启动时联动组的特性取决于以下机床数据中的设置：

MD20112 \$MC_START_MODE_MASK（确定 NC 启动时控制系统的初始设置）

位	值	含义
8	0	联动组在 NC 启动时保留。
	1	联动组在 NC 启动时拆分。

15.1 联动

复位/零件程序结束时的特性

复位/零件程序结束时联动组的特性取决于以下机床数据中的设置：
MD20110 \$MC_RESET_MODE_MASK（确定复位/零件程序结束后控制系统的初始设置）

位	值	含义
8	0	联动组在复位/零件程序结束时拆分。
	1	联动组在复位/零件程序结束后仍然有效。

说明
如果在 NC 复位或零件程序结束时，在一个通道中引导轴未同时停止（跨通道耦合、指令轴、PLC 轴等），则无法完成请求的复位。
由于引导轴的运行，联动轴对于请求复位的通道仍然有效。通过适当的措施（引导轴通道中的 NC 复位、停止指令轴或 PLC 轴），引导轴也必须平行于联动轴停止。

15.1.3 编程

15.1.3.1 联动组的定义和启用 (TRAILON)

联动组的定义和启用是随零件程序指令 TRAILON 同时进行的。

编程

语法：TRAILON (<联动轴>、<引导轴>、[<耦合系数>])
生效方式：模态
参数：

联动轴:	类 AXIS 型:
	值域: 在通道中定义的所有轴和主轴名称
引导轴:	类 AXIS 型:
	值域: 在通道中定义的所有轴和主轴名称
耦合系数:	通过可选的耦合系数指定联动轴与引导轴的运行比率: 耦合系数 = 联动轴行程/引导轴行程 负耦合系数会导致引导轴和联动轴反向运行。
	类 REAL 型:
	值域: ± (2.2 * 10-308 ... 1.8 * 10+308)
	缺省值: +1.0

示例:

程序代码	注释
TRAILON(V,Y,2)	; 定义和启用联动轴 V 与引导轴 Y 的耦合。耦合系数为 2。

15.1.3.2 **关闭 (TRAILOF)**

通过零件程序指令 TRAILOF 关闭联动轴与引导轴的耦合。

编程

语法:	TRAILOF (<联动轴>、<引导轴>) TRAILOF (<联动轴>)
生效方式:	模态
参数:	
联动轴:	类 AXIS 型:
	值域: 在通道中定义的所有轴和主轴名称
引导轴:	类 AXIS 型:
	值域: 在通道中定义的所有轴和主轴名称

示例：

程序代码	注释
TRAILOF (V,Y)	； 关闭联动轴
	； V 与引导轴 Y 的耦合。

15.1.4 PLC 接口信号的有效性

独立联动轴

对于联动轴的独立运动，联动轴的所有相关通道和轴专用接口信号都有效，例如：

- <Chan>.basic.out.handwheelOffsetDrf（激活 DRF）
- <Axis>.basic.out.feedrateOvrFactor（进给倍率）
- <Axis>.basic.out.disable（进给轴禁用）
- <Axis>.basic.out.enableCtrl（伺服使能）
- <Axis>.basic.out.handwheelSelection（激活手轮）
- <Axis>.basic.out.feedStopSpindleStop（进给停止）
- ...

因此，对于联动轴的独立运动，可以通过进给倍率等改变速度，或者可以在 AUTO 和 MDA 运行方式下使用手轮指定 DRF 偏移。

相关联动轴

对于与引导轴相关的联动轴的运动，只有导致运动停止的联动轴的接口信号有效（例如轴专用进给停止、伺服使能等）。

引导轴

在联动组激活时，引导轴的接口信号通过轴耦合作用在相关的联动轴上，即：

- 引导轴的位置偏移或进给影响通过耦合系数引起联动轴的相应位置偏移或进给影响。
- 通过接口信号（例如轴专用进给停止、伺服使能等）停止引导轴会导致相关的联动轴停止。

位置测量系统 1/2 (<Axis>.basic.out.enc1Activation/<Axis>.basic.out.enc2Activation)

对于激活的联动组，引导轴和联动轴的位置测量系统的切换未锁定。此时耦合保留。

建议：在耦合关闭时进行切换。

跟随 (<Axis>.basic.out.followUpMode)

使用 PLC 用户程序通过设置以下 NC/PLC 接口信号激活轴的跟随模式：

<Axis>.basic.out.enableCtrl = 0 (控制器使能)

<Axis>.basic.out.followUpMode (跟随模式)

在联动组的跟随模式激活时，必须同时为联动组的所有轴（引导轴和跟随轴）设置指定的 NC/PLC 接口信号。

如果只为引导轴激活跟随模式，则在不再运行的耦合中会出现永久偏移。

可以通过以下 NC/PLC 接口信号和系统变量确定是否是以及哪个轴是引导轴或跟随轴：

<Axis>.coupling.in.syncCoarseReached (引导轴/主主轴激活)

<Axis>.coupling.in.ProcessActive (跟随轴/副主轴激活)

\$AA_COUP_ACT [轴名称] (参见：耦合的状态)

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Chan>.basic.out.handwheelOffsetDrf	LBP_Chan*.A_DRF	DB21, ...DBX0.3
<Axis>.basic.out.feedrateOvrFactor	LBP_Axis*.A_FD_OR	DB31, ...DBB0
<Axis>.basic.out.disable	LBP_Axis*.A_ASpDisable	DB31, ...DBX1.3
<Axis>.basic.out.followUpMode	LBP_Axis*.A_Follow_upMode	DB31, ...DBX1.4
<Axis>.basic.out.enc1Activation	LBP_Axis*.A_PosMeas1	DB31, ...DBX1.5
<Axis>.basic.out.enc2Activation	LBP_Axis*.A_PosMeas2	DB31, ...DBX1.6
<Axis>.basic.out.enableCtrl	LBP_Axis*.A_ContrEnable	DB31, ...DBX2.1
<Axis>.basic.out.resetMovement	LBP_Axis*.A_DeIDTGSpReset	DB31, ...DBX2.2
<Axis>.basic.out.handwheelSelection	LBP_Axis*.A_HW1..3	DB31, ...DBX4.0..2
<Axis>.basic.out.feedStopSpindleStop	LBP_Axis*.A_FDSpStop	DB31, ...DBX4.3

15.1 联动

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.coupling.in.syncCoarseReached	LBP_Axis*.E_SyncronCoarse	DB31, ...DBX98.1
<Axis>.coupling.in.ProcessActive	LBP_Axis*.E_Sync	DB31, ...DBX99.1

15.1.5 耦合状态

轴的耦合状态可以通过以下系统变量确定：

\$AA_COUP_ACT [轴名称]

值	含义
0	无耦合有效
1, 2, 3	切向跟随
4	同步主轴耦合
8	联动生效
16	引导值耦合
32	电子齿轮箱的跟随轴

说明

在同一时间点始终只能有一种耦合类型生效。

15.1.6 动态响应限制

动态响应限制取决于激活联动组的方式：

- 在零件程序中激活
如果在零件程序中激活，而所有的引导轴被用作激活通道中的编程轴，那么在引导轴运行时考虑所有联动轴的动态响应，避免出现过载。
如果在零件程序中激活，而其中的引导轴没有被用作激活通道中的编程轴 (\$AA_TYP ≠ 1)，那么在引导轴运行时不会考虑联动轴的动态响应。因此，如果某个联动轴的动态响应稍稍低于耦合要求的水平，会使该轴出现过载。
- 在同步动作中激活
如果在同步动作中激活，那么在引导轴运行时不会考虑联动轴的动态响应。因此，如果某个联动轴的动态响应稍稍低于耦合要求的水平，会使该轴出现过载。

 小心

轴过载
如果一个联动组

- 在同步动作中
- 或在零件程序中被激活，其中的引导轴不是联动轴通道中的编程轴，

那么用户/机床制造商应负责采取相应的措施，避免引导轴的运行导致联动轴出现过载。

15.1.7 边界条件

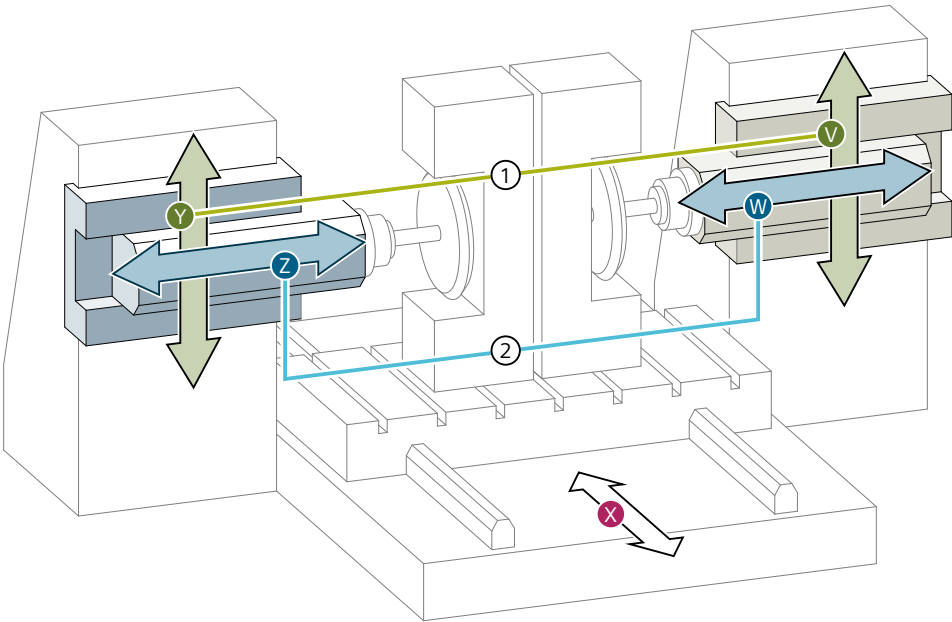
控制动态响应

建议在联动组内匹配引导轴和联动轴的位置调节器参数。

说明
引导轴和联动轴的位置调节器参数的调整可以通过参数组切换动态进行。

15.1.8 示例

应用示例：双面加工



示例 1

图中所示轴位置分布的 NC 零件程序示例：

程序代码	注释
TRAILON (V, Y, 1)	; 启用第 1 个联动组
TRAILON (W, Z, -1)	; 启用第 2 个联动组
G0 Z10	; Z 轴和 w 轴以相反的轴向进给
G0 Y20	; Y 轴和 v 轴以相同的轴向进给
G1 Y22 V25	; 叠加联动轴“v”的一个相关运动和独立运动
TRAILOF (V, Y)	; 关闭第 1 个联动组
TRAILOF (W, Z)	; 关闭第 2 个联动组

示例 2

联动时，联动轴的相关运动和独立运动相加。相关部分可以看作是相对于联动轴的坐标偏移。

程序代码	注释
N01 G90 G0 X100 U100	;
N02 TRAILON (U, X, 1)	; 启用联动组

程序代码	注释
N03 G1 F2000 X200	; U 的相关运动, Upos=200, UTrail=100
N04 U201	; 独立运动 Upos=U201+UTrail=301
N05 X250	; U 的相关运动, UTrail=UTrail(100)+50=150, Upos=351
N06 G91 U100	; 独立运动, Upos(351)+U100=451
N07 G90 X0	; U 的相关运动, Upos=Upos(451)-UTrail(250)=201
N10 TRAILOF(U,X)	

15.2 曲线表

15.2.1 简要说明

15.2.1.1 功能

借助“曲线表”功能，可以在曲线表中定义轴的复杂运动过程。

任何轴都可以定义为引导轴，并考虑到曲线表移动跟随轴。

这些运动过程的控制量是由控制系统生成或从外部变量（例如轴的模拟位置）推导出的抽象引导值。

曲线表是通过零件程序序列创建的。

静态 NC 存储器中的曲线表在零件程序结束和断电后保留。

曲线表可以存储在动态 NC 存储器中，以加快处理速度。这里需要注意，启动后必须重新读入表格。

带有曲线表的轴组必须在上电后重新激活，无论曲线表的存储位置如何。

线性曲线表段存储在自己的区域中，以节省存储空间。

15.2.1.2 前提条件

存储器配置

静态 NC 存储器

静态 NC 存储器中曲线表的存储空间用以下机床数据定义：

MD18400 \$MN_MM_NUM_CURVE_TABS（曲线表数量）

MD18402 \$MN_MM_NUM_CURVE_SEGMENTS（曲线段数量）

15.2 曲线表

MD18403 \$MN_MM_NUM_CURVE_SEG_LIN（线性曲线段数量）

MD18404 \$MN_MM_NUM_CURVE_POLYNOMS（曲线表多项式数量）

动态 NC 存储器

动态 NC 存储器中曲线表的存储空间用以下机床数据定义：

MD18406 \$MN_MM_NUM_CURVE_TABS_DRAM（曲线表数量）

MD18408 \$MN_MM_NUM_CURVE_SEGMENTS_DRAM（曲线段数量）

MD18409 \$MN_MM_NUM_CURVE_SEG_LIN_DRAM（线性曲线段数量）

MD18410 \$MN_MM_NUM_CURVE_POLYNOMS_DRAM（曲线表多项式数量）

15.2.2 一般功能

曲线表

曲线表中描述了控制量“引导值”与抽象跟随值之间的函数关系。

在引导值的一个定义好的数值范围内，可以明确给每个引导值分配跟随值。

曲线段

函数关系可以分为引导值轴的各个部分，即所谓的曲线段。

在一个曲线段内，引导值和跟随值之间的关系通常由最多 3 次的多项式来描述。也允许最多 5 次的多项式。

更多信息：编程手册之 NC 编程

在以下情况下需要曲线段：

- 编程了多项式或圆弧
- 样条已生效
- 压缩器已生效
- 多项式或圆弧在内部生成（倒角/倒圆，使用 G643 进行精磨、刀具半径补偿等）

刀具半径补偿

即使曲线表中出现不含轴运动或跟随轴跳跃的多项式程序段或程序段，也可以在表定义中指定刀具半径补偿（表定义中的 G41/G42/G40），可使用此类曲线表。

由多项式组成的曲线的等距曲线（刀具半径补偿的刀具-中点轨迹）无法再用多项式精确表示。因此，在这种情况下，必须使用多项式逐个逼近相关的曲线表。这就导致在这种情况下，曲线表的段数不再与编程的段数一致。曲线表所需的段数由曲线的曲率决定。编程曲线的曲率越大，曲线表所需的段数就越多。

由于曲线表中的刀具半径补偿，可能需要额外的存储空间。但通过选择存储器类型，这不会导致静态 NC 存储器中的存储空间不足。

选择存储器类型

定义曲线表时，可以指定曲线表是在静态还是在动态 NC 存储器中创建。

说明

静态 NC 存储器中的表定义在控制系统启动后仍然可用。动态 NC 存储器的曲线表必须在控制系统启动后重新定义。

15.2.3 存储器结构

存储器配置

静态和动态 NC 存储器中曲线表的可用存储空间在存储器配置时定义（参见章节“存储器配置 (页 587)”）。

存储器优化

对于具有线性段的曲线表，只有当以下两个机床数据 > 0 时，线性段才能更高效地存储在存储器中：

MD18403 \$MC_MM_NUM_CURVE_SEG_LIN（静态 NC 存储器中的线性曲线段数量）

MD18409 \$MC_MM_NUM_CURVE_SEG_LIN_DRAM（动态 NC 存储器中的线性曲线段数量）

如果没有使用这些机床数据创建存储区，则线性段将保存为多项式段，浪费空间。

内存不足报警

如果通过机床数据配置了线性段表和多项式段表的内存，并且在创建线性表时线性段的内存用尽，则使用多项式段的内存用于线性段（如果可用）。在这种情况下，内存被“浪费”了，因为多项式段需要不必要的大量内存来存储线性段。这种情况通过一个报警来传达，它还表明了不必要消耗的多项式段的数量。该报警只是一个警告，不会中断程序或曲线表的创建。

15.2 曲线表

如果一个曲线表由线性段和高次多项式组成，则需要一个线性段存储区和一个多项式段存储区来存储曲线表。相应存储区内存不足时会输出一个报警。通过报警参数可以识别哪些资源不足。

内存过小

如果由于内存不足而无法创建曲线表，则一旦出现报警，新创建的表就会被再次删除。

如果内存过小，则必须首先使用 CTABDEL 删除一个或多个不再需要的表，或者通过机床数据重新配置存储器。

临时曲线表

创建新曲线表时，首先在内存中创建一个临时曲线表，然后逐个程序段补充。最后 (CTABEND) 检查表的一致性。只有在曲线表有效时，临时表才会成为可用于耦合的表。

相同的表编号

要创建的曲线表可以采用现有表的编号。此时，新的曲线表将覆盖现有的具有相同编号的表。只有新曲线表不包含任何错误时才执行此操作。如果新曲线表有错误，则保留旧表。

如果用户想使用覆盖现有曲线表而不事先删除的方式，那么在确定表内存的大小时，用户必须始终预留出要覆盖的表的范围。

覆盖曲线表

可以覆盖在引导值耦合上不生效并用 CTABLOCK() 锁定的曲线表。

删除曲线表

可以删除在引导值耦合上不生效并用 CTABLOCK() 锁定的曲线表。

15.2.4 调试

15.2.4.1 存储器配置

在静态和动态 NC 存储器中为曲线表提供定义的存储空间，该存储空间由以下机床数据确定：

静态 NC 存储器	
MD18400 \$MN_MM_NUM_CURVE_TABS	指定可以在静态 NC 存储器中创建的 曲线表数量 。
MD18402 \$MN_MM_NUM_CURVE_SEGMENTS	指定可以在静态 NC 存储器中创建的 曲线表段数量 。
MD18403 \$MN_MM_NUM_CURVE_SEG_LIN	指定静态 NC 存储器中 线性段的最大数量 。
MD18404 \$MN_MM_NUM_CURVE_POLYNOMS	指定可以在静态 NC 存储器创建的 曲线表多项式数量 。

动态 NC 存储器	
MD18406 \$MN_MM_NUM_CURVE_TABS_DRAM	指定可以在动态 NC 存储器中创建的 曲线表数量 。
MD18408 \$MN_MM_NUM_CURVE_SEGNENTS_DRAM	指定可以在动态 NC 存储器中创建的 曲线表段数量 。
MD18409 \$MN_MM_NUM_CURVE_SEG_LIN_DRAM	指定动态 NC 存储器中 线性段的最大数量 。
MD18410 \$MN_MM_NUM_CURVE_POLYNOMS_DRAM	指定可以在动态 NC 存储器中创建的 曲线表多项式数量 。

说明

只有在以下情况下，具有线性段的曲线表才能更高效地存储在存储器中：

MD18403 > 0 或 MD18409 > 0

如果没有使用这些机床数据创建存储区，则线性段将保存为多项式段，浪费空间。

15.2.4.2 刀具半径补偿

MD20900

刀具半径补偿时，可能会出现跟随轴或引导轴不运动的段。跟随轴不运动通常没有问题。而如果引导轴不运动，则需要说明如何处理这种不连续性，即在这些情况下是否应创建曲线表。通过以下机床数据中的设置进行说明：

15.2 曲线表

MD20900 \$MC_CTAB_ENABLE_NO_LEADMOTION（包含跟随轴跳跃的曲线表）

值	含义
0	不创建包含跟随轴跳跃的曲线表。输出报警 10949，程序处理终止。
1	可以创建包含跟随轴跳跃的曲线表。如果在一个段中发生跟随轴跳跃，则输出报警 10955，但程序处理继续。
2	可以创建包含跟随轴跳跃的曲线表，不输出报警。

说明

对于包含无引导轴运动的段的曲线表（这意味着跟随轴在该位置进行跳跃），跟随轴只能在其动态响应限制（最大速度和最大加速度）内完成跳跃。这表明在这种情况下，始终会出现与编程曲线的偏差。

15.2.4.3 指定存储器类型

MD20905

如果在定义或删除曲线表时未指定存储器，则存储器类型由以下机床数据指定：

MD20905 \$MC_CTAB_DEFAULT_MEMORY_TYPE（曲线表的默认存储器类型）

值	含义
0	默认情况下，曲线表在静态 NC 存储器中创建。
1	默认情况下，曲线表在动态 NC 存储器中创建。

15.2.5 编程

定义

以下模态生效的语言指令适用于曲线表：（参数说明见功能列表末端。）

- 开始曲线表的定义：
CTABDEF(跟随轴, 引导轴, n, applim, memType)
- 结束曲线表的定义：
CTABEND()

- 删除曲线表：
CTABDEL(n)
; 曲线表 n
CTABDEL(n, m)
; [n < m], 编号范围中有多个
; 将在静态“SRAM”和动态“DRAM” NC 存储器中删除。
- CTABDEL(n, m, memType)
; 根据指定的存储器类型删除：
在指定的存储器类型中，从存储区中删除编号在上述存储器类型中的曲线表。其他曲线表保持不变。
删除某个特定存储器类型中的所有表：
CTABDEL(, "DRAM")
CTABDEL(, "SRAM")
CTABDEL()
; DRAM 中的全部或
; SRAM 中的全部：
; 全部，无关存储器类型
- 读取相对于引导值的跟随值
CTAB(引导值, n, grad, [跟随轴, 引导轴])
- 读取相对于跟随值的引导值
CTABINV(跟随值, 近似引导值, n, grad, [跟随轴, 引导轴])

访问曲线表段

- 读取表段的起始值（跟随轴值）
CTABSSV(引导值, n, grad, [跟随轴, 引导轴])
- 读取表段的终值（跟随轴值）
CTABSEV(引导值, n, grad, [跟随轴, 引导轴])

说明

如果 CTAB()、CTABINV()、CTABSSV() 等曲线表功能用于**同步动作**，则只有**主运行变量**，例如 \$AC_PARAM[...] 或 \$R[...]，才允许用于功能的返回值和“grad”自变量。

示例：

ID=1 WHEN TRUE DO \$R1 = CTABSSV(10, 1, \$R2)

或

ID=1 WHEN TRUE DO \$AC_PARAM[1] = CTABSSV(10, 1, \$AC_PARAM[2])

设置/取消锁定

以下功能允许零件程序设置或取消锁定以防止删除和覆盖。

- **设置**防止删除和覆盖的**锁定**功能。

一般形式: CTABLOCK(n, m, memType)

- **取消**防止删除和覆盖的**锁定**功能。

CTABUNLOCK 会再次激活用 CTABLOCK 锁定的表。在激活的耦合中有效的表继续保持锁定, 即仍然无法删除。但用 CTABLOCK 进行的锁定就已取消, 也就是说, 一旦通过禁用耦合取消了激活耦合的锁定, 就可以删除该表。不必再次调用 CTABUNLOCK。

一般形式: CTABUNLOCK(n, m, memType)

应用于以下形式:

编号为 n 的曲线表

CTABLOCK(n)

编号范围从 n 到 m 的曲线表

CTABLOCK(n, m)

所有曲线表, 无论存储器类型如何

CTABLOCK()

指定存储器类型中的所有曲线表

CTABLOCK(, memType)

编号为 n 的曲线表

CTABUNLOCK(n)

编号范围从 n 到 m 的曲线表

CTABUNLOCK(n, m)

所有曲线表, 无论存储器类型如何

CTABUNLOCK()

指定存储器类型中的所有曲线表

CTABUNLOCK(, memType)

在诊断和优化资源使用时用于确定和区分应用曲线表的其他指令：

- **定义的表的数量，无论存储器类型如何**
CTABNO()
- **在静态或动态 NC 存储器中定义的表的数量**
CTABNOMEM(memType)
- **存储器 memType 中尚可使用的曲线表数量。**
CTABFNO(memType)
- **第 n 个曲线表的表编号。**
一般形式：CTABID(n, memType)
在存储器类型 memType 中提供第 n 个曲线表的表编号。使用 CTABID(1, memType) 读取指定存储器类型的最高曲线编号 (105)。
CTABID(n)
在由以下机床数据指定的存储器中，提供第 n 个曲线表的表编号：
MD20905 \$MC_CTAB_DEFAULT_MEMORY_TYPE（曲线表的默认存储器类型）
CTABID(p)
提供作为第 p 个曲线表输入到存储器中的曲线表的 ID（表编号）。

说明

例如，如果在连续调用 CTABID() 之间更改了顺序，则可以使用 CTABID(n, ...) 来提供与存储器内更改前不同的曲线表。

- **返回编号为 n 的曲线表的锁定状态。**
CTABISLOCK(n)
- **检查编号为 n 的曲线表。**
CTABEXISTS(n)
- **返回在其中创建了编号为 n 的曲线表的存储器。**
CTABMEMTYP(n)
- **返回表周期性。**
CTABPERIOD(n)
- **存储器 memType 中已经使用的曲线段的数量。**
CTABSEG(memType, segType)
- **编号为 n 的曲线表所使用的曲线段数量**
CTABSEGID(n, segType)
- **存储器 memType 中尚可使用的曲线段数量。**
CTABFSEG(memType, segType)

- 存储器 memType 中**最大可用的曲线段**数量。
CTABMSEG(memType, segType)
- 存储器 memType 中**已经使用的多项式**数量。
CTABPOL(memType)
- 编号为 n 的曲线表所使用的**曲线多项式**数量。
CTABPOLID(n)
- 存储器 memType 中**尚可使用的多项式**数量。
CTABFPOL(n)
- 存储器 memType 中**最大可用的多项式**数量。
CTABMPOL(n)

曲线表的边界值

曲线表边缘的引导轴或跟随轴的特性：

- 由跟随轴读取曲线表**开头**的数值。
CTABTSV(n, grad, FAchse), 曲线表开头的跟随值
- 由跟随轴读取曲线表**结尾**的数值。
CTABTEV(n, grad, FAchse), 曲线表结尾的跟随值
- 由引导轴读取曲线表**开头**的数值。
CTABTSP(n, grad, FAchse), 曲线表开头的引导值
- 由引导轴读取曲线表**结尾**的数值。
CTABTEP(n, grad, FAchse), 曲线表结尾的引导值
- 确定跟随值的取值范围。
CTABTMIN(n, FAchse), 曲线表的最小跟随值
CTABTMAX(n, FAchse), 曲线表的最大跟随值

一览

- 跟随轴：
在定义中用于编程跟随轴的轴的名称。
- 引导轴：
用于编程引导值的轴的名称。

- **n, m**
曲线表的编号。
曲线表的编号可以任意分配。它们仅用于唯一识别曲线表。
要使用 CTABDEL(n, m) 指令删除一定范围的曲线表，m 必须大于 n。
- **p**
输入位置（在 memType 存储区）
- **applim:**
曲线表边缘上的特性。
 - 0 不具有周期性（即使是回转轴也只处理一次表。）
 - 1 周期性，模数（模数值取决于引导轴的表值。）
 - 2 周期性，模数（引导轴和跟随轴具有周期性。）
- **引导值**
要确定跟随值的位置值。
- **跟随值**
要计算引导值的位置值。
- **近似引导值**
在曲线表的反转功能不唯一的情况下，可以用于确定唯一引导值的位置值。
- **grad**
在其中返回表函数梯度的参数。
- **memType**
用于指定要在其中创建曲线表的存储器类型的可选参数。
允许的值：
“SRAM”曲线表在静态 NC 存储器中创建。
“DRAM”曲线表在动态 NC 存储器中创建。
如果输入了无效的 memType，则返回值 -2。
如果忽略该参数，则通过以下机床数据设置的存储器类型生效：
MD20905 \$MC_CTAB_DEFAULT_MEMORY_TYPE（曲线表的默认存储器类型）
- **segType**
用于指定段类型的可选参数
允许的值：
segType “L”线性段
segType “P”多项式段

更多信息：编程手册之 NC 编程

限制

- 对编程有以下限制：
- NC 程序段不得生成预处理程序停止。
 - 引导轴的运动中不得出现跳跃。
 - 每个包含跟随轴运动指令的程序段也必须具有引导轴运动。
 - 引导轴的运动方向不得在运动规律的任何位置反转，即引导轴的位置在运动过程中必须始终是明确的。
编程的轮廓不得垂直于引导轴运动。
 - 龙门轴组中的轴名称不能用于表定义（仅引导轴适用）。
 - 根据以下机床数据可以在缺少引导轴运动时允许跟随轴跳跃。
MD20900 \$MC_CTAB_ENABLE_NO_LEADMOTION（包含跟随轴跳跃的曲线表）
上述其他限制仍然适用。

轴分配

仅在激活与曲线表的耦合时进行。

说明

对曲线表运动指令的动态极限值的检查只在激活或插补过程中进行。

起始值

曲线表定义中的第一个运动指令确定了引导值和跟随值的起始值。
必须隐藏导致预处理程序停止的所有指令。

示例 1

无刀具半径补偿，无存储器类型

程序代码	注释
N100 CTABDEF (AX2, AX1, 3.0)	; 开始定义不具有 ; 周期性的 3 号曲线表
N110 AX1=0 AX2=0	; 第 1 个运动指令确定起始值 ; 引导值：0，跟随值：0
N110 AX1=20 AX2=0	; 第 1 个曲线段：引导值：0...20， ; 跟随值：起始值 ...0
N120 AX1=100 AX2=6	; 第 2 个曲线段：引导值：20...100，

程序代码	注释
	; 跟随值: 0...6
N130 AX1=150 AX2=6	; 第 3 个曲线段: 引导值: 100...150,
	; 跟随值 6
N130 AX1=180 AX2=0	; 第 4 个曲线段: 引导值: 150...180,
	; 跟随值: 6...0
N200 CTABEND	; 结束定义, 曲线表
	; 在其内部示意图中生成。
	; 预处理重组为
	; N100 开始时的状态

示例 2

具有激活刀具半径补偿的曲线表示例:

在用 CTABDEF() 定义曲线表之前, 刀具半径补偿不得激活, 否则会输出报警 10942。也就是说, 刀具半径补偿必须在曲线表的定义内激活。同样, 在曲线表定义结束前, 必须用 CTABEND 将其再次禁用。

程序代码	注释
N10 CTABDEF(Y, X, 1, 0)	; 开始定义不具有
	; 周期性的 1 号曲线表
N20 X0 Y0	
N30 G41 X10 Y0	; 刀具半径补偿开启
N40 X20 Y20	
N50 X40 Y0	
N60 X60 Y20	
N70 X80 Y0	
N80 G40 X90 Y0	; 刀具半径补偿关闭
N90 CTABEND	

在程序段 N30 中, 刀具半径补偿将激活, 因此在该程序段中进行半径补偿的逼近运动。同样, 在程序段 N80 中, 禁用半径补偿时进行退回运动。

说明

对于在 CTABDEF 中指定为引导轴名称和跟随轴名称的轴名称, 必须在 CTABDEF 和 CTABEND 之间指定值对。在故障情况下, 可能会生成报警或不需要的轮廓。

15.2.6 访问表位置和表段

读取表位置

使用程序指令 CTAB 和 CTABINV 可以从零件程序和同步动作中直接读取相对于引导值的跟随值 (CTAB) 或曲线表的反转，即相对于跟随值的引导值。用户可以使用梯度值来计算跟随轴或引导轴在所需表位置的速度。

读取段位置

可以通过调用 CTABSSV 和 CTABSEV 读取关于跟随轴值的曲线表段位置。

语言指令 CTABSSV 和 CTABSEV 一般提供跟随轴曲线表内部段的起始值或终值。这些值只有在编程段可以 1:1 转换为曲线表内部段时才与曲线表的编程值一致。如果在定义曲线表时仅使用 G1 程序段或轴多项式，而没有激活其他功能，则始终是这种情况。

在下列条件下，编程段可能**不会**原封不动地转换为曲线表的内部段：

- 编程了圆弧或者渐开线
- 倒角或倒圆 (CHF, RND) 生效
- 使用 G643 进行精磨的功能生效
- 压缩器功能 (COMP...) 生效
- 多项式插补时的刀具半径补偿生效

在这些情况下，语言指令 CTABSSV 和 CTABSEV 不能用于查询编程段的起点或终点。

CTABINV

使用曲线表的反转功能 CTABINV 时必须注意，从跟随值到引导值的映射可能不是唯一的。

在一个曲线表中，跟随值对于（任意）数量的引导值位置可以具有相同的值。为了能够解决这种不唯一性，程序指令 CTABINV 除了跟随值之外还需要另一个参数，利用该参数可以选择“正确的”可用引导值。CTABINV 始终返回最接近此辅助参数的引导值作为结果。例如，该辅助值可以是来自前一个插补节拍的引导值。

说明

虽然曲线表的反转功能可以利用辅助参数计算出唯一的结果，但必须指出的是，可以生成轮廓，由于数值不准确，反转功能不能给出无限精确的计算结果。

可选参数

CTAB、CTABINV、CTABSSV 和 CTABSEV 功能具有可选的引导轴或跟随轴参数。如果对这些参数之一进行了编程，则要用相关轴的比例缩放系数修改引导值和跟随值。

如果轴是以不同的长度单位（英制/公制）配置的，这一点尤其重要。如果未对可选参数进行编程，则在从外部转换为内部表示时，将引导值和跟随值视为轨迹位置。这表示，根据配置的解法（小数位）将值相乘，并省略剩余的小数位。

确定属于引导值 X 的段

使用 CTABSSV 和 CTABSEV 读取段起始值和段终值以确定属于引导值 X = 30 的曲线段的示例：

程序代码	注释
N10 DEF REAL STARTPOS	; 开始定义曲线表的起始位置和初始位置
N20 DEF REAL ENDPOS	
N30 DEF REAL GRADIENT	
N100 CTABDEF(Y,X,1,0)	; 开始表定义
N110 X0 Y0	; 起始位置，第 1 个表段
N120 X20 Y10	; 终点位置，第 1 个表段 =
	; 起始位置，第 2 个表段
N130 X40 Y40	
N140 X60 Y10	; 第 3 个曲线段
N150 X80 Y0	; 第 4 个曲线段
N160 CTABEND	; 结束表定义
N200 STARTPOS=CTABSSV(30.0,1,GRADIENT)	; 段 2 中的起始位置 Y = 10。
N210 ENDPOS=	; 段 2 中的终点位置 Y = 40
CTABSEV(30.0,1,GRADIENT)	; 属于引导值 x = 30.0 的段 2

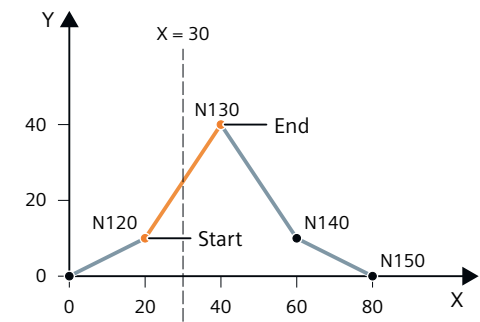


图 15-2 确定属于引导值 X=30 的曲线段

读取开头和结尾的值

调用以下内容可读取曲线表开头和结尾跟随轴和引导轴的值：

15.2 曲线表

R10 =CTABTSV(n. grad, 跟随轴, 表开头的跟随值)

R10 =CTABTEV(n. grad, 跟随轴, 表结尾的跟随值)

R10 =CTABTSP(n. grad, 引导轴, 表开头的引导值)

R10 =CTABTEP(n. grad, 引导轴, 表结尾的引导值)

跟随值的取值范围

以下示例显示使用 CTABTMIN 和 CTABTMAX 确定表的最小值和最大值：

程序代码	注释
N10 DEF REAL STARTVAL	; 开始定义曲线表的起始值和初始值
N20 DEF REAL ENDVAL	
N30 DEF REAL STARTPARA	
N40 DEF REAL ENDPARA	
N50 DEF REAL U_MINVAL	
N60 DEF REAL U_MAXVAL	
N70 DEF REAL GRADIENT	
...	
N100 CTABDEF(Y,X,1,0)	; 开始表定义
N110 X0 Y10	; 起始值，第 1 个表段
N120 X30 Y40	; 终点位置，第 1 个表段 =
	; 起始位置，第 2 个表段
N130 X60 Y5	; 终点位置，第 2 个表段 ...
N140 X70 Y30	; 终点位置，第 3 个曲线段 ...
N150 X80 Y20	; 终点位置，第 4 个曲线段 ...
N160 CTABEND	; 结束表定义
...	
N200 STARTPOS=CTABTSV(1,GRADIENT)	; STARTPOS = 10
	; 表的起始位置以及
N210 ENDPOS=CTABTEV(1,GRADIENT)	; ENDPOS = 20
	; 表的终点位置
N220 STARTPARA=CTABTSP(1,GRADIENT)	; STARTPARA = 0
	; 曲线表开头的引导值
N230 ENDPARA=CTABTEP(1,GRADIENT)	; ENDPARA = 80
	; 读取跟随轴取值范围曲线表结尾的引导值
N240 U_MINVAL=CTABTMIN(1)	; 当 Y = 5 时的最小值和
N250 U_MAXVAL=CTABTMAX(1)	; 当 Y = 40 时的最大值

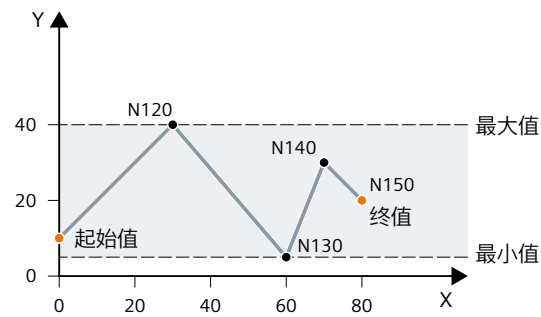


图 15-3 确定表的最小值和最大值

15.2.7 激活/禁用

激活

真实轴到曲线表的耦合通过以下指令激活：

LEADON (<跟随轴>, <引导轴>, <n>)

其中 <n> = 曲线表的编号

激活可以在：

- 零件程序中进行
- 定义同步动作时进行

示例：

...	
N1000 LEADON (A,X,3)	; A 轴根据 3 号曲线表中定义的运动规律跟随引导值 X
...	

禁用

通过以下指令关闭到曲线表的耦合：

LEADOF (<跟随轴>, <引导轴>)

15.2 曲线表

禁用可以在：

- 零件程序中进行
- 在同步动作中

说明

编程 LEADOF 时，也可以使用缩写形式，不指定引导轴。

示例：

```
...  
N1010 LEADOF (A,X)          ; A 轴到其引导值的耦合将取消  
...
```

多次使用

一个曲线表可以在一个零件程序中多次使用，并用于耦合不同的通道轴。

15.2.8 模数引导轴特殊情况

位置起绝对作用

在轴的引导值耦合时，跟随轴的位置通过曲线表唯一（即绝对地）分配到引导轴。
这表明，即使在模数回转轴作为引导轴的情况下，引导轴的位置也是绝对的。也就是说，曲线表中包含的模数回转轴的位置不是模数减少的，它起绝对作用。

示例

采用 LEADON 时模数回转轴的位置为 210°。210° 位置用作曲线表的输入值。模数回转轴旋转一周后，轴位置再次显示 210°。但曲线表的输入值采用绝对位置 570°：
 $210^{\circ} + 1 \text{ 周 } (360^{\circ}) = 570^{\circ}$

15.2.9 AUTO、MDA、JOG 运行方式下的特性

有效性

激活的曲线表在 AUTO、MDA 和 JOG 运行方式下生效。

启动后的初始位置

启动后没有曲线表生效。

15.2.10 PLC 接口信号的有效性

相关跟随轴

对于与引导轴相关的跟随轴的运动，只有导致运动停止的跟随轴的接口信号有效（例如轴专用进给停止、进给轴禁用、伺服使能等）。

引导轴

在轴组激活时，引导轴的接口信号通过轴耦合作用在相关的跟随轴上，即：

- 引导轴的进给影响会导致跟随轴的相应进给影响。
- 通过接口信号（例如轴专用进给停止、进给轴禁用、伺服使能等）停止引导轴会导致相关的跟随轴停止。

通过以下机床数据设置可以防止引导轴的轴禁用对跟随轴的影响：

MD37160 \$MA_LEAD_FUNCTION_MASK, 位 1 = 1

位置测量系统 1/2 (<Axis>.basic.out.enc1Activation/<Axis>.basic.out.enc2Activation)

对于激活的轴组，引导轴和跟随轴的位置测量系统的切换未锁定。此时耦合保留。

建议：在耦合关闭时进行切换。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.enc1Activation	LBP_Axis*.A_PosMeas1	DB31, ...DBX1.5
<Axis>.basic.out.enc2Activation	LBP_Axis*.A_PosMeas2	DB31, ...DBX1.6

15.2.11 诊断和优化资源使用

以下功能允许**零件程序**了解目前被曲线表、表段和多项式占用的资源情况。

作为诊断功能的结果，仍然可用的资源可以**动态**使用，必要时灵活使用下方功能，而不会增加绝对内存需求。“曲线表编程”章节中的参数说明也适用于以下功能。

a) 曲线表

- 确定定义的表的总数。
通过所有存储器类型进行确定（另见 CTABNOMEM）
CTABNO ()
- 确定在静态或动态 NC 存储器中定义的表的数量。
CTABNOMEM (memType)
如果未指定 memType，则适用以下机床数据中设置的存储器类型：
MD20905 \$MC_CTAB_DEFAULT_MEMORY_TYPE（曲线表的默认存储器类型）
结果：
>= 0: 定义的表的数量
-2: 存储器类型无效

- 确定存储器类型中尚可使用的曲线表数量。

CTABFNO(memType)

如果未指定 memType，则适用以下机床数据中设置的存储器类型：

MD220905 \$MC_CTAB_DEFAULT_MEMORY_TYPE

结果：

>= 0: 尚可使用的表的数量

-2: 存储器类型无效

- 确定可选指定存储器类型中第 p 个表的表编号

CTABID(p, memType)

如果未指定 memType，则适用以下机床数据中设置的存储器类型：

MD20905 \$MC_CTAB_DEFAULT_MEMORY_TYPE

结果：

表编号，或者

在 p 或 memType 无效时报警

在使用功能 CTABID(p, memType) 时，不应假定曲线表在存储器中的顺序。功能 CTABID(p, ...) 提供作为第 p 个曲线表输入到存储器中的曲线表的 ID（表编号）。

如果在连续调用 CTABID() 之间更改了存储器中曲线表的顺序，例如通过使用 CTABDEL() 删除曲线表，则功能 CTABID(p, ...) 可以提供与以前不同的具有相同编号的曲线表。

如果要防止这种情况，可以使用语言指令 CTABLOCK(...) 锁定相关曲线表。在这种情况下要注意，使用 CTABUNLOCK() 再次使能相关曲线表。

- 查询**锁定状态**

表 n

CTABISLOCK(n)

结果：

> 0: 表已锁定

锁定原因：

1: 由 CTABLOCK()

2: 由生效的耦合

3: 由 CTABLOCK() 和生效的耦合

= 0: 表未锁定

- 1: 表不存在

- 检查曲线表是否**存在**

CTABEXISTS(n)

结果：

1: 表存在

0: 表不存在

- 确定曲线表的**存储器类型**

CTABMEMTYP (n)

结果:

0: 表在静态 NC 存储器中“SRAM”

1: 表在动态 NC 存储器中“DRAM”

-1: 表不存在

- 确定表是否被定义为**周期性**

CTABPERIOD (n)

结果:

0: 表不具有周期性

1: 引导轴上表具有周期性

2: 引导轴和跟随轴上表具有周期性

-1: 表不存在

b) 曲线表段

- 确定 memType 存储区中 segType 类型**所使用的**曲线段的数量。

- CTABSEG (memType, segType)

- 如果未指定 memType, 则适用以下机床数据中设置的存储器类型:

MD20905 \$MC_CTAB_DEFAULT_MEMORY_TYPE

结果:

>= 0: 曲线段数量

-2: 存储器类型无效

如果未指定 segType, 则通过存储器类型中的线性段和多项式段输出总和。

-2: segType 不等于“L”或“P”

- 确定特定曲线表的 segType 类型所使用的曲线段的数量

CTABSEGID (n, segType)

结果:

>= 0: 曲线段数量

-1: 编号为 n 的曲线表不存在

-2: segType 不等于“L”或“P”

- 确定该存储器类型的 **segType** 类型的**未占用**曲线段的数量
`CTABFSEG (memType, segType)`
 如果未指定 **memType**，则适用以下机床数据中设置的存储器类型：
`MD20905 $MC_CTAB_DEFAULT_MEMORY_TYPE`
 结果：
 ≥ 0 ：尚未占用的曲线段的数量
 -2 ：存储器类型无效，**segType** 不等于“L”或“P”
 - 确定该存储器类型的 **segType** 类型的**最大可用**曲线段数量
`CTABMSEG (memType, segType)`
 如果未指定 **memType**，则适用以下机床数据中设置的存储器类型：
`MD20905 $MC_CTAB_DEFAULT_MEMORY_TYPE`
 结果：
 ≥ 0 ：最大可用曲线段数量
 -2 ：存储器类型无效，**segType** 不等于“L”或“P”
- c) 多项式**
- 确定该存储器类型**所使用的**多项式数量
`CTABPOL (memType)`
 如果未指定 **memType**，则适用以下机床数据中设置的存储器类型：
`MD20905 $MC_CTAB_DEFAULT_MEMORY_TYPE`
 结果：
 ≥ 0 ：该存储器类型中所使用的多项式数量
 -2 ：存储器类型无效
 - 确定一个曲线表所使用的曲线多项式数量
`CTABPOLID (n)`
 结果：
 ≥ 0 ：所使用的曲线多项式数量
 -1 ：编号为 **n** 的曲线表不存在

15.2 曲线表

- 确定该存储器类型尚未占用的多项式数量
CTABFPOL (memType)
如果未指定 memType，则适用以下机床数据中设置的存储器类型：
MD20905 \$MC_CTAB_DEFAULT_MEMORY_TYPE
结果：
≥ 0：尚未占用的曲线多项式数量
-2：存储器类型无效
- 确定该存储器类型的最大可用多项式数量
CTABMPOL (memType)
如果未指定 memType，则适用以下机床数据中设置的存储器类型：
MD20905 \$MC_CTAB_DEFAULT_MEMORY_TYPE
结果：
≥ 0：最大可用曲线多项式数量
-2：存储器类型无效

15.2.12 边界条件

转换

曲线表中不允许转换。TRAANG 除外。

TRAANG

如果编程了 TRAANG，则在基准坐标系中编程的运动规律将转换为相关的机床坐标系。通过这种方式，可以对带有倾斜线性轴的机床的曲线表进行笛卡尔编程。

然后必须在机床坐标系中满足条件“引导轴的运动方向不得在运动规律的任何位置反转”。请注意，此条件在基准坐标系中的含义与在机床坐标系中的含义不同，因为轮廓切线会因转换而改变。

15.2.13 示例

用线性程序段定义一个曲线表

```
%_N_TAB_1_NOTPERI_MPF  
; $PATH=/_N_WKS_DIR/_N_KURVENTABELLEN_WPD
```

% _N_TAB_1_NOTPERI_MPF	
;Def.TAB1 0-100mm Kue1/1 notperio.	
N10 CTABDEF(YGEO,XGEO,1,0)	; FA=Y LA=X 曲线编号=1 不具有周期性
N1000 XGEO=0 YGEO=0	; 起始值
N1010 XGEO=100 YGEO=100	
CTABEND	
M30	

用多项式程序段定义一个曲线表

% _N_TAB_1_NOTPERI_MPF	
; \$PATH=/_N_WKS_DIR/_N_KURVENTABELLEN_WPD	
;Def.TAB1 0-100mm Kue1/1 notperio.	
N10 CTABDEF(Y,X,1,0)	; FA=Y LA=X 曲线编号=1 不具有周期性
N16 G1 X0.000 Y0.000	
N17 POLY PO[X]=(31.734,0.352,-0.412) PO[Y]=(3.200,2.383,0.401)	
N18 PO[X]=(49.711,-0.297,0.169) PO[Y]=(7.457,1.202,-0.643)	
N19 PO[X]=(105.941,1.961,-0.938) PO[Y]=(11.708,-6.820,-1.718)	
N20 PO[X]=(132.644,-0.196,-0.053) PO[Y]=(6.815,-2.743,0.724)	
N21 PO[X]=(147.754,-0.116,0.103) PO[Y]=(3.359,-0.188,0.277)	
N22 PO[X]=(174.441,0.578,-0.206) PO[Y]=(0.123,1.925,0.188)	
N23 PO[X]=(185.598,-0.007,0.005) PO[Y]=(-0.123,0.430,-0.287)	
N24 PO[X]=(212.285,0.040,-0.206) PO[Y]=(-3.362,-2.491,0.190)	
N25 PO[X]=(227.395,-0.193,0.103) PO[Y]=(-6.818,-0.641,0.276)	
N26 PO[X]=(254.098,0.355,-0.053) PO[Y]=(-11.710,0.573,0.723)	
N26 PO[X]=(254.098,0.355,-0.053) PO[Y]=(-11.710,0.573,0.723)	
N27 PO[X]=(310.324,0.852,-0.937) PO[Y]=(-7.454,11.975,-1.720)	
N28 PO[X]=(328.299,-0.209,0.169) PO[Y]=(-3.197,0.726,-0.643)	
N29 PO[X]=(360.031,0.885,-0.413) PO[Y]=(0.000,-3.588,0.403)	
CTABEND	
N30 M30	

定义一个周期性曲线表

表编号： 2

引导值范围： 0 - 360

跟随轴从 N70 到 N90 执行从 0 到 45 再回到 0 的运动。

N10 DEF REAL DEPPOS	
N20 DEF REAL GRADIENT	
N30 CTABDEF(Y,X,2,1)	

15.3 引导值耦合

N10 DEF REAL DEPPOS	
N40 G1 X=0 Y=0	
N50 POLY	
N60 PO[X]=(45.0)	
N70 PO[X]=(90.0) PO[Y]=(45.0,135.0,-90)	
N80 PO[X]=(270.0)	
N90 PO[X]=(315.0) PO[Y]=(0.0,-135.0,90)	
N100 PO[X]=(360.0)	
N110 CTABEND	
N130 G1 F1000 X0	; 通过耦合 Y 到 X 对曲线进行测试
N140 LEADON(Y,X,2)	
N150 X360	
N160 X0	
N170 LEADOF(Y,X)	
N180 DEPPOS = CTAB(75.0,2,GRADIENT)	; 从编号为 2 的曲线表中读取引导值 75.0 时的表位置
N190 G0 X75 Y=DEPPOS	; 定位引导轴和跟随轴
N200 LEADON(Y,X,2)	; 启用耦合之后，无需对跟随轴进行同步
N210 G1 X110 F1000	
N220 LEADOF(Y,X)	
N230 M30	

15.3 引导值耦合

15.3.1 简要说明

15.3.1.1 功能

利用“轴的引导值耦合”功能，可以对短程序进行循环处理，使各轴之间和与引导值之间紧密耦合，引导值可以内部生成，也可以外部指定。

例如，引导值可以由传送带或中心轴得出。

轴的引导值耦合可以在 NC 零件程序中或通过同步动作启用和关闭。

15.3.1.2 前提条件

要使用该功能，需要“引导值耦合和曲线表插补”选项或通用耦合的相应可选类型（参见通用耦合“简要说明”中的主题“前提条件”）。

15.3.2 一般功能

曲线表

在轴的引导值耦合时同步运行一个引导轴和一个跟随轴。同时，跟随轴的相应位置通过一个曲线表或者通过一个从该表算得的多项式已明确分配给引导轴的一个（有可能是模拟的）位置。

引导值对象

引导值对象是曲线表的输入变量。

以下可以定义为引导值对象的位置：

- 轴实际位置（编码器测量的实际值）
或
- 设定位置（由插补器计算）（默认设置）

如果引导轴由同一个 NCU 插补，则设定值耦合提供比实际值耦合更好的跟随特性（在相同的插补节拍中）。

虚拟引导轴/模拟引导值

如果引导轴不由同一个 NCU 插补，则 NCU 中存在的用于该引导轴的插补器可用于引导值模拟。为此必须进行以下机床数据设置：

MD30132 \$MA_IS_VIRTUAL_AX[n] = 1（轴为虚拟轴）

MD30130 \$MA_CTRLOUT_TYPE[n] = 0（模拟作为设定值输出方式）

引导值模拟的属性：

- 分离插补器与伺服。
- 记录轴的实际值。
- 设定值由插补器生成，但不传输到伺服。
- 切换到模拟引导值时，可以根据最后读取的实际值对模拟进行编程，而实际值的变化过程通常不受 NCU 的控制。
- 如果为了引导值模拟将引导值对象从实际值耦合切换到设定值耦合，并在同一插补节拍中发出引导轴的运行指令，则 NC 初始化该轴的插补器，使一阶导数中产生引导值的连续变化过程。

说明

分配给真实驱动器的虚拟轴必须保持未锁定状态。

偏移和缩放

跟随轴的设定值可以偏移和缩放。为此使用以下设置数据：

SD43102 \$SA_LEAD_OFFSET_IN_POS（耦合到该轴时引导值的偏移）

SD43104 \$SA_LEAD_SCALE_IN_POS（耦合到该轴时引导值的缩放）

SD43106 \$SA_LEAD_OFFSET_OUT_POS（曲线表函数值的偏移）

SD43108 \$SA_LEAD_SCALE_OUT_POS（曲线表函数值的缩放）

如果 (x) 是周期性曲线表且被视为振动，则偏移和缩放也可以这样解释：

SD43102 \$SA_LEAD_OFFSET_IN_POS[Y] 移动振动的相位。

SD43106 \$SA_LEAD_SCALE_OUT_POS[Y] 作用于振幅。

SD43108 \$SA_LEAD_OFFSET_OUT_POS[Y] 移动振动的中点

如果耦合已启用且同步，写入该设置数据会导致立即实施新的设定位置。

SD43104 \$SA_LEAD_SCALE_IN_POS[FA]
输入值相乘。

SD43108 \$SA_LEAD_SCALE_OUT_POS[FA]
输出值相乘。设置数据在表定义之外生效，即使在负方向。

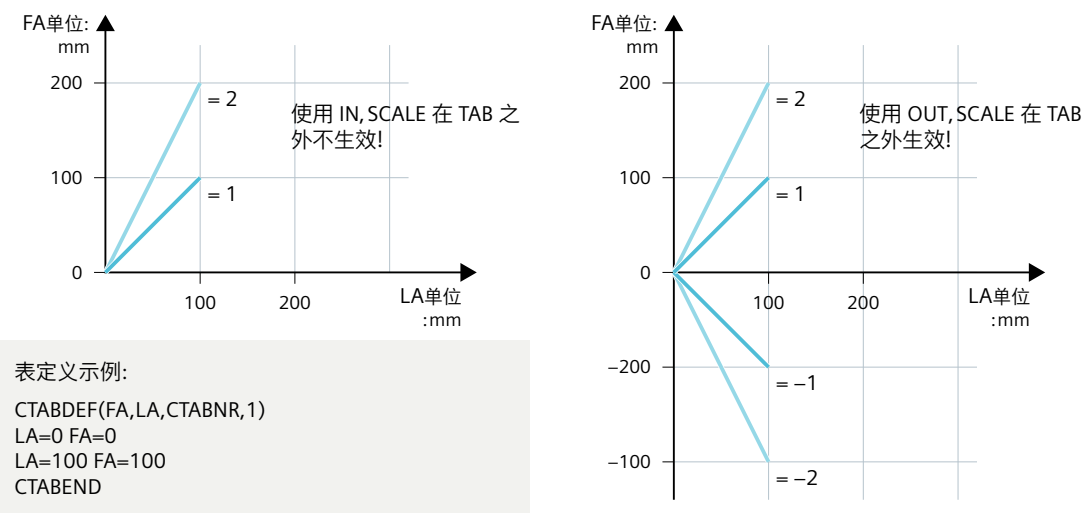
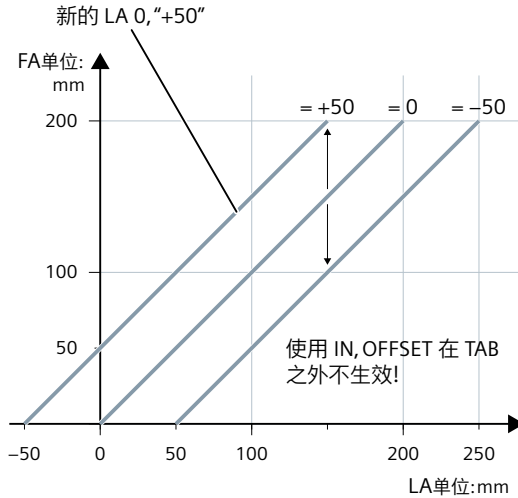


图 15-4 移动和缩放引导值耦合（相乘）

SD43102 \$SA_LEAD_OFFSET_IN_POS[FA]
引导轴的计算位置以指定的增量移动。



SD43106 \$SA_LEAD_OFFSET_OUT_POS[FA]
跟随轴的计算位置以指定的增量移动。如果 TAB 移到 FA LEADON 位置之外,则无法启用 KOP。设置数据在表定义之外生效,即使在负方向。

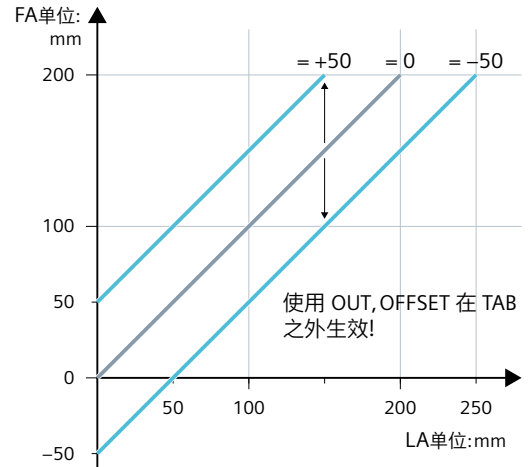


图 15-5 移动和缩放引导值耦合（用增量移动）

对停止的反应

所有引导值耦合的跟随轴对通道停止和 BAG 停止作出反应。

在因程序结束(M30, M02)而停止时,未通过静态同步动作(IDS=...)启用的引导值耦合的跟随轴作出反应。在这种情况下,必须注意以下机床数据:

MD20110 \$MC_RESET_MODE_MASK (确定复位/零件程序结束后控制系统的初始设置)

MD20112 \$MC_START_MODE_MASK (确定 NC 启动时控制系统的初始设置)

引导轴和跟随轴必须始终在同一通道内插补。无法耦合另一个通道中的跟随轴(跨通道取轴)。

通过启动和切换运行方式,停止的引导值耦合的跟随轴再次使能。

通过复位,停止的引导值耦合的跟随轴也再次使能。如果此使能是非期望的或危险的(例如由于跟随轴耦合到不受 NC 控制的外部引导值),则 MD20110 应编程为在复位时关闭引导值耦合(=2001H,即将位 13 设置为 1)。

轴的功能

实际值耦合时,引导轴和跟随轴之间会产生位置偏移。原因是位置调节器中与插补节拍相关的时滞,它在引导轴和跟随轴的实际值之间。

15.3 引导值耦合

默认情况下，引导值的线性外推为位置偏移和滞后量补偿该时滞，也就是说，引导值耦合的时滞补偿生效。要关闭时滞补偿，可设置以下机床数据：

```
MD37160 $MA_LEAD_FUNCTION_MASK 位 0 = 0
```

用于跨通道取轴的接口

引导值耦合的跟随轴从曲线表得到其设定值。**无法在零件程序中对该轴进行叠加编程**。因此，引导值耦合的跟随轴从通道退出，方式与跨通道取轴相同。当零件程序中的耦合启用时，这会自动进行。

如果要用同步动作启用耦合，则必须事先使用 RELEASE 来为此做好准备。

在关闭引导值耦合后，可以在零件程序中重新编程之前的跟随轴。

引导值耦合中的主轴

只有事先将主轴切换到轴运行方式，才能将其用作引导值耦合的跟随轴。此时适用轴运行方式的机床数据参数组。

示例：从同步动作中启用

程序代码	注释
SPOS=0	
B=IC(0)	; 主轴切换到轴运行方式。
RELEASE(Y)	; 为同步动作释放。
ID=1 WHEN (\$AA_IM[X]<-50) DO LEADON(B,X,2)	; Y 通过 2 号曲线表耦合到 X。

15.3.3 编程

定义和启用

轴引导值耦合的定义和启用与模态生效的语言指令 LEADON 同时发生。

语法：

```
LEADON(<FA>,<LA>,<CTABn>)
```

含义:

- <FA>

跟随轴作为几何、通道或机床轴名称 (X,Y,Z,...)
- <LA>

引导轴作为几何、通道或机床轴名称 (X,Y,Z,...)
- 软件轴也可以:
- MD30130 \$MA_CTRLOUT_TYPE=0 (设定值输出方式)
- <CTABn>

曲线表的编号
- 值域:

1 至 999

示例:

程序代码	注释
LEADON (Y, X, 1)	; 定义和启用引导轴 X 与跟随轴 Y 之间的引导值耦合。要计算跟随值，应使用 1 号曲线表。

边界条件:

- 启用耦合时不需要参考点。
- 一个定义的跟随轴无法在 JOG 运行方式下运行（即使尚未出现接口信号“同步运行”或“粗同步”也无法运行）。
- 已启用的耦合必须先用 LEADOF 再次关闭，然后才能用 LEADON 重新启用。在这方面，必须遵守以下机床数据中的设置：

MD20112 MC_START_MODE_MASK（确定 NC 启动时控制系统的初始设置）

MD20110 \$MC_RESET_MODE_MASK（确定复位/零件程序结束后控制系统的初始设置）

KOP 启用并与 FA 位置同步

只有当 LA 进入 FA 的表范围时,KOP 才会启用。

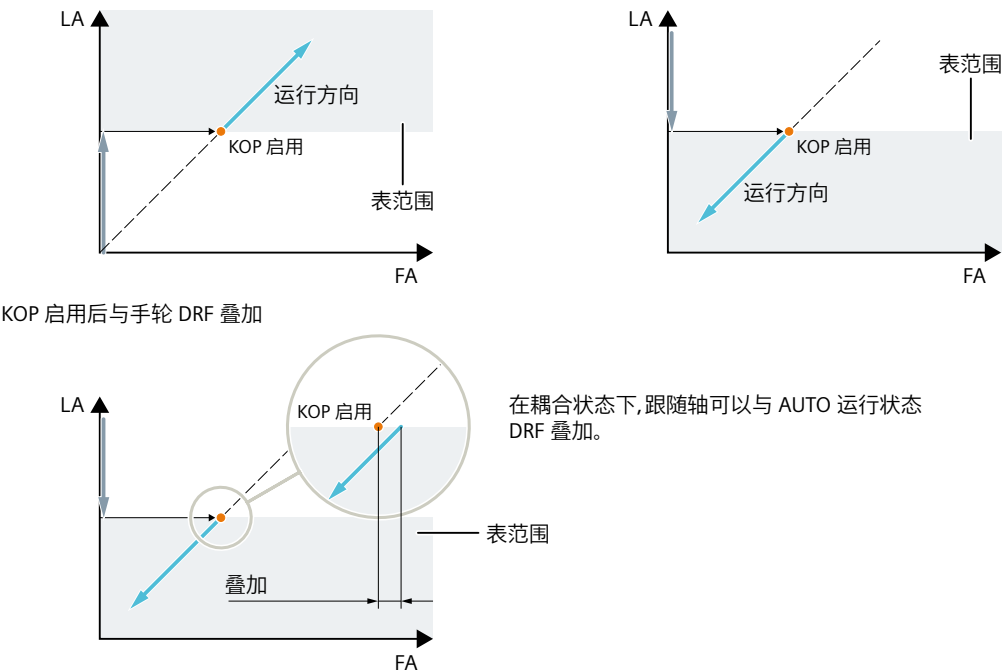


图 15-6 激活引导值耦合

关闭

轴引导值耦合的关闭通过语言指令 LEADOF 进行。

关闭轴引导值耦合时，跟随轴变为指令轴，会为跟随轴隐式生成停止指令。停止指令可以通过同步动作被其他指令覆盖。

语法:

LEADON (<FA> , <LA>)

含义:

- <FA> 跟随轴作为几何、通道或机床轴名称 (X,Y,Z,...)
- <LA> 引导轴作为几何、通道或机床轴名称 (X,Y,Z,...)
- 软件轴也可以:
- MD30130 \$MA_CTRLOUT_TYPE=0 (设定值输出方式)

示例:

程序代码	注释
LEADOF (Y,X,1)	; 关闭引导轴 X 与跟随轴 Y 之间的引导值耦合。

说明

在零件程序和同步动作中都允许用 LEADON / LEADOF 启用/关闭轴引导值耦合。

更多信息

功能手册之同步动作分册

耦合方式

耦合方式可通过以下轴专用设置数据定义:

SD43100 \$SA_LEAD_TYPE[<LA>] (引导值类型)

<LA>: 引导轴作为几何轴名称、通道轴名称或机床轴名称 (X,Y,Z,...)

值	含义
0	实际值耦合 (只有这种耦合方式可用于外部引导轴)
1	设定值耦合 (默认设置)
2	模拟引导值 (注意虚拟轴, 未针对跟随轴评估)

可以随时在实际值耦合和设定值耦合之间切换 (最好在空闲阶段)。

引导值的系统变量

以下引导值系统变量只能从零件程序和同步动作中读取：

系统变量	含义
\$AA_LEAD_V[ax]	引导轴的速度
\$AA_LEAD_P[ax]	引导轴的位置
\$AA_LEAD_P_TURN	引导值位置 在模数反应中减去的部分。 实际（非模数减少的）引导值位置为： $\$AA\_LEAD\_P\_TURN + \$AA\_LEAD\_P$

模拟引导值的速度和位置（\$SA_LEAD_TYPE[ax]=2 时）可以在零件程序和同步动作中读取和写入。

系统变量	含义
\$AA_LEAD_SV[ax]	每个插补节拍的模拟引导值速度
\$AA_LEAD_SP[ax]	机床坐标系中的模拟位置

跟随轴的系统变量

可以在零件程序和同步动作中读取跟随轴的以下系统变量：

系统变量	含义	
\$AA_SYNC[ax]	跟随轴和引导轴之间的耦合状态	
	值	含义
	0	未同步
	1	“粗同步” ¹⁾
	2	“精同步” ²⁾
	3	“粗同步” ¹⁾ 和“精同步” ²⁾
	关联数据：	
	1)	- MD37200 \$MA_COUPLE_POS_TOL_COARSE <Axis>.coupling.in.syncCoarseReached（粗同步）
	2)	- MD37210 \$MA_COUPLE_POS_TOL_FINE <Axis>.coupling.in.syncFineReached（精同步）

系统变量	含义	
\$AA_IN_SYNC[ax]	跟随轴和引导轴之间的同步状态	
	值	含义
	0	同步未启动或已结束。
	1	正在进行同步，即跟随轴正在进行同步。
	\$AA_IN_SYNC[ax] 对应 NC/PLC 接口信号： <Axis>.coupling.in.ProcessActive（正在进行同步）	

说明

如果跟随轴没有运行使能，它会停止并且不再同步。

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.coupling.in.syncFineReached	LBP_Axis*.E_SyncronFine	DB31, ...DBX98.0
<Axis>.coupling.in.syncCoarseReached	LBP_Axis*.E_SyncronCoarse	DB31, ...DBX98.1
<Axis>.coupling.in.ProcessActive	LBP_Axis*.E_Sync	DB31, ...DBX99.4

15.3.4 AUTO、MDA、JOG 运行方式下的特性

有效性

引导值耦合根据零件程序和以下机床数据中的设置生效：

MD20110 \$MC_RESET_MODE_MASK（确定复位/零件程序结束后控制系统的初始设置）

MD20112 \$MC_START_MODE_MASK（确定 NC 启动时控制系统的初始设置）

手动运行

在轴引导值耦合激活时，引导轴的运行（例如通过快速运行或增量 INC1 ... INC10000）会导致跟随轴运行，同时考虑曲线表的定义。

回参考点

在启用耦合之前，应使引导值耦合的跟随轴回参考点。耦合启用时，跟随轴无法回参考点。

剩余行程删除

删除一个引导轴的剩余行程会导致相关激活的引导值耦合的所有轴停止。

上电后的初始位置

启动后没有引导值耦合生效。（可通过异步子程序）。

NC 启动/复位时的特性

根据机床数据中的设置，会产生以下特性：

MD20110 \$MC_RESET_MODE_MASK（位 13）（确定复位/零件程序结束后控制系统的初始设置）

MD20112 \$MC_START_MODE_MASK（位 13）（确定 NC 启动时控制系统的初始设置）

- MD20110 \$MC_RESET_MODE_MASK=2001H
 &&
 MD20112 \$MC_START_MODE_MASK=0H
 → 引导值耦合在复位和启动时保留
- MD20110 \$MC_RESET_MODE_MASK=2001H
 &&
 MD20112 \$MC_START_MODE_MASK=2000H
 → 引导值耦合在复位时保留，在启动时拆分。但通过 IDS=... 启用的引导值耦合保留。
- MD20110 \$MC_RESET_MODE_MASK=1H
 → 引导值耦合在复位时拆分，无关以下机床数据：
 MD20112 \$MC_START_MODE_MASK
 通过 IDS=... 启用的引导值耦合仅在操作面板前复位时关闭，在程序结束/复位 (M30, M02) 时保留。
- MD20110 \$MC_RESET_MODE_MASK=0H
 → 引导值耦合在复位时保留，在启动时拆分，无关以下机床数据：
 MD20112 \$MC_START_MODE_MASK
 但通过 IDS=... 启用的引导值耦合保留。

更多信息

功能手册之基本功能：坐标系、轴类型、轴配置等

启用、关闭

通过静态同步动作 (IDS=...) 启用的引导值耦合:

- 在程序启动时不关闭, 无关以下机床数据值:
MD20110 \$MC_RESET_MODE_MASK (确定复位/零件程序结束后控制系统的初始设置)
和
MD20112 \$MC_START_MODE_MASK (确定 NC 启动时控制系统的初始设置)
- 在程序结束复位 (M30, M02) 时不关闭, 无关以下机床数据值:
MD20110 \$MC_RESET_MODE_MASK

15.3.5 PLC 接口信号的有效性

引导轴

引导值耦合激活时, 引导轴的接口信号通过轴耦合作用在相关的跟随轴上。即:

- 引导轴的进给影响会通过引导值耦合导致跟随轴的相应进给影响。
- 通过接口信号 (例如轴专用进给停止、进给轴禁用、伺服使能等) 停止引导轴会导致相关的跟随轴停止。

位置测量系统 1/2 (<Axis>.basic.out.enc1Activation/<Axis>.basic.out.enc2Activation)

对于激活的轴组, 引导轴和跟随轴的位置测量系统的切换未锁定。此时耦合保留。

建议: 在耦合关闭时进行切换。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.enc1Activation	LBP_Axis*.A_PosMeas1	DB31, ...DBX1.5
<Axis>.basic.out.enc2Activation	LBP_Axis*.A_PosMeas2	DB31, ...DBX1.6

15.3.6 轴引导值耦合功能的特点

控制动态响应

根据应用，为轴组调整引导轴和跟随轴的位置调节器参数（例如 K_v 系数）是有意义的。必要时应激活跟随轴的其他参数组。跟随轴应具有与引导轴相同或更好的动态响应。

耦合的状态

参见章节“耦合状态 (页 580)”

实际值显示

引导值耦合的轴组中所有轴（仅限真实轴）的实际位置显示都会更新。

插补

对曲线表中定义的运动进行插补时，会针对引导值及其速度计算轴位置和轴速度。

归档

通过定义运动过程创建的曲线表存储在缓冲存储器中。

曲线表在控制系统关闭后保留。

一方面，这些功能对循环机床不起作用，因为其运行不需要操作员干预；另一方面，对于外部引导值，由 NC 进行自动（重新）定位是没有意义的。

15.3.7 边界条件

外部引导值轴

使用零件程序指令 REPOS 或 REPOSA 时，关于外部引导值轴要注意，事先用零件程序指令 RELEASE 将其从通道中使能或设置为“中性状态”。

如果不使能该轴，在试图重新定位时会显示消息“等待：进给停止生效”，零件程序的处理不会继续。

15.4 电子齿轮箱 (EG)

15.4.1 简要说明

15.4.1.1 功能

概述

借助“电子齿轮箱”功能，能够根据最多五个引导轴对跟随轴的运动进行控制。引导轴和跟随轴之间的关联按照每个引导轴通过耦合系数进行定义。从各个引导轴运动分量计算的跟随轴运动分量是相加的。

耦合可能涉及：

- 引导轴的实际值
- 引导轴的设定值

齿轮组的以下功能可以通过零件程序指令编程：

- 定义
- 启用
- 关闭
- 删除

曲线表

非线性关系也可以通过曲线表在引导轴和跟随轴之间实现。

级联

电子齿轮箱可以级联，即一个电子齿轮箱的跟随轴可以是另一个电子齿轮箱的引导轴。

同步位置

通过一个附加功能，可以在同步跟随轴时选择一个同步位置：

- 以时间最优的方式逼近下一分度（齿隙）
- 以行程最优的方式逼近下一分度（齿隙）
- 沿轴的正旋转方向绝对逼近

15.4 电子齿轮箱 (EG)

- 沿轴的负旋转方向绝对逼近
- 以时间最优的方式运行到编程的同步位置
- 以行程最优的方式运行到编程的同步位置

应用示例:

- 齿轮生产机床
- 生产型机械的传动链

15.4.1.2 前提条件

要使用该功能，需要“电子齿轮箱”选项或通用耦合的相应可选类型（参见通用耦合“简要说明”中的主题“前提条件”）。

15.4.2 电子齿轮箱 (EG)

功能

借助“电子齿轮箱”功能，可以根据最多五个**引导轴 LA**对**跟随轴 FA**的运动进行插补。引导轴和跟随轴之间的关联按照每个引导轴通过耦合系数进行定义。以此从各个引导轴运动分量计算的跟随轴运动分量是相加的。

$$FA_{\text{设定}} = \text{SynPosFA} + (LA_1.\text{SynPosLA}_1) * KF_1 + \dots + (LA_5.\text{SynPosLA}_5) * KF_5$$

其中:

SynPosFA、SynPosLA_i: 来自调用 EGONSYN（见下）

FA_{设定}: 跟随轴的分量设定值

LA_i: 第 i 个引导轴的设定值或实际值（取决于耦合类型（见下）

KF_i: 第 i 个引导轴的耦合系数（见下）

所有行程以基准坐标系 **BCS** 为基准。

激活一个 EG 轴组时，可以使引导轴和跟随轴在一个定义的初始位置上同步。

一个齿轮组可以由零件程序:

- 定义,
- 启用,

- 关闭，
- 删除。

扩展

作为传动比（ $KF=\text{分子}/\text{分母}$ ）的替代方案，最多 5 个引导轴中的每一个的影响可以通过**曲线表**来指定。

因此对于每条曲线（与直线特殊情况不同），引导轴都有可能以**非线性**方式影响跟随轴。该功能只能与 EGONSYN 一起使用。

EG 功能可以通过包含 EGON 的曲线表来激活。

可以使用 EGONSYNE 功能来逼近跟随轴的同步位置，并指定逼近模式。

对于特殊应用，将位置控制器配置为 **PI 控制器**可能有意义。



小心

应用错误

使用此功能时，控制技术专业知识和使用伺服跟踪的测量是必不可少的。

更多信息

- CNC 调试手册：NC、PLC、驱动
- 速度、设定值/实际值系统、闭环控制 (页 141)

耦合类型

跟随轴运动可以选择性地由以下内容导出：

- 引导轴的设定值
- 引导轴的实际值

通过 EG 轴组的定义调用进行确定：

EGDEF

（参见“定义一个 EG 轴组 (页 630)”章节）

耦合系数

耦合组的每个引导轴都需要耦合系数。它由分子/分母定义。

耦合系数值的分子和分母在以下激活调用中为每个引导轴指定：

15.4 电子齿轮箱 (EG)

EGON

EGONSYN

EGONSYNE

(参见“定义一个 EG 轴组 (页 630)”章节)

EG 轴组数量

在同一时间点可以定义多个 EG 轴组。EG 轴组的最大数量由以下机床数据定义：

MD11660 \$MN_NUM_EG

最多允许 31 个 EG 轴组。

说明

必须启用该选项。

EG 级联

一个 EG 的跟随轴可以是另一个 EG 的引导轴。相关详细示例参见“示例”章节。

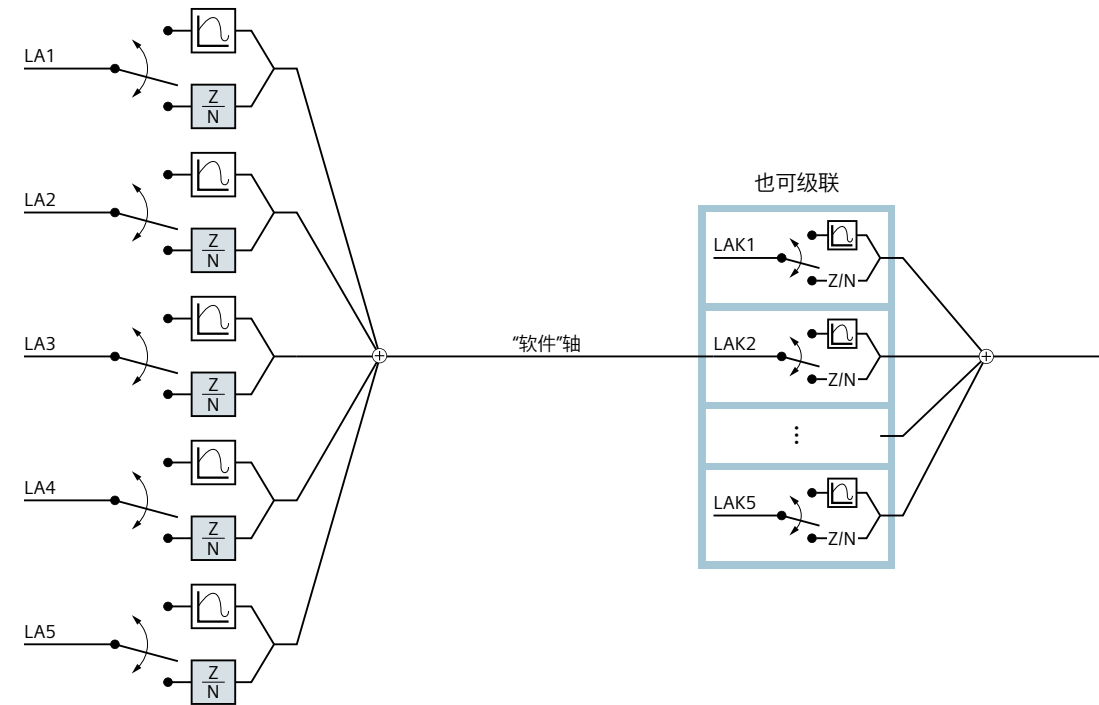


图 15-7 电子齿轮箱的方框图

同步位置

为了启动 EG 轴组，可以首先要求跟随轴逼近定义的位置。

同步位置用以下内容确定：

EGONSYN（详细信息见下文）

EGONSYNE（扩展调用 EGONSYN）

同步

如果使用 EGON() 或 EGONSYN() 或 EGONSYNE() 启用齿轮箱（见下文），则零件程序开发人员必须确保跟随轴在该时间点的实际位置与由齿轮箱的运动规律根据引导轴位置指定的设定位置完全一致。此时，控制系统通过跟随轴的运动，确保在引导轴继续运动时，尽快实现跟随轴设定位置 and 实际位置的一致。这个过程称为同步。跟随轴同步后，就可以说是一个同步齿轮箱。

启用方式

可通过两种方式激活电子齿轮箱：

1. 根据迄今为止在加工过程中已到达的轴位置，发出激活 EG 轴组的指令，而不指定各个轴的同步位置。
EGON（参见“启用一个 EG 轴组 (页 631)”章节）
2. 通过激活 EG 轴组的指令，为每个轴指定同步位置。从到达这些位置的时间点开始，EG 应同步。
EGONSYN（参见“启用一个 EG 轴组 (页 631)”章节）
3. 通过激活 EG 轴组的指令，为每个轴指定同步位置和逼近模式。从到达这些位置的时间点开始，EG 应同步。
EGONSYNE（参见“启用一个 EG 轴组 (页 631)”章节）

用 EGON 同步

使用 EGON() 时，不指定跟随轴应该同步的位置。控制器启用 EG 并报告“已达到同步位置”。

用 EGONSYN 同步

1. 使用 EGONSYN() 时，引导轴的位置和跟随轴的同步位置由指令指定。

15.4 电子齿轮箱 (EG)

- 然后控制系统以足够的加速度和速度将跟随轴运行到指定的同步位置，从而使跟随轴与引导轴处于同步位置。
- 跟随轴**静止**时：如果为跟随轴设置了接口信号“进给停止/主轴停止”`<Axis>.basic.out.feedStopSpindleStop`，那么跟随轴**不会**使用 EGON 或 EGOSYN 进行运动。为跟随轴发出一个运行指令，并阻止程序段转换，直到使能轴专用进给。EGOSYN 被复位终止并转换为 EGON。编程的同步位置随即丢失。
- 跟随轴**非静止**时：接口信号“进给停止/主轴停止”`<Axis>.basic.out.feedStopSpindleStop` 对电子齿轮箱没有直接影响。和以前一样，它间接作用于引导轴，前提是位于同一通道中。
- 对于**通道专用**进给使能和对于倍率都未进行任何实施。倍率仍然对电子齿轮箱没有直接影响。会根据当前倍率位置设置**轴专用**进给使能。

用 EGOSYNE 同步

使用 EGOSYNE() 时，引导轴的位置和跟随轴的同步位置由指令指定。

控制系统根据编程的逼近模式将跟随轴运行到同步位置。

终止用 EGOSYN 和 EGOSYNE 进行的同步

1. EGOSYN/EGOSYNE 指令在以下条件下终止并转换为 EGON 指令：

- 复位
- 轴开始跟随

不再考虑定义的同步位置。同步运行监控仍然会考虑到同步位置。

终止位置同步会导致报警 16774。

可以使用以下机床数据抑制报警本身：

MD11410 \$MN_SUPPRESS_ALARM_MASK 位 31 = 1

同步运行监控

在每个插补节拍中，使用跟随轴和引导轴的实际值来监控齿轮箱的同步性。为此，使用实际值重新计算耦合的运动规律。**同步运行偏差**是指跟随轴的实际值与依照耦合的运动规律根据引导轴的实际值得出的值之间的差值。可以从零件程序中查询同步运行偏差（见下文）。

更改同步运行偏差

由于加速时轴系统的转动惯量，同步运行偏差中可能出现动态波动。不断检查同步运行偏差，并通过机床数据中的公差值产生接口信号。

将同步运行偏差与以下机床数据进行比较：

MD37200 \$MA_COUPLE_POS_TOL_COARSE

MD37210 \$MA_COUPLE_POS_TOL_FINE

根据比较结果设置以下信号：

接口信号“精同步” <Axis>.coupling.in.syncSpindleFineReached

接口信号“粗同步” <Axis>.coupling.in.syncSpindleCoarseReached

差值 > ..TOL_COARSE

只要同步运行偏差大于以下机床数据，齿轮箱就不是同步的，接口中既不会出现接口信号“粗同步” <Axis>.coupling.in.syncSpindleCoarseReached，也不会出现接口信号“精同步” <Axis>.coupling.in.syncSpindleFineReached：

MD37200 \$MN_COUPLE_POS_TOL_COARSE

而是显示以下接口信号：

接口信号“正在进行同步” <Axis>.coupling.in.syncSpindleProcessActive

差值 < ..TOL_COARSE

如果同步运行偏差小于以下机床数据，则接口中出现接口信号“粗同步” <Axis>.coupling.in.syncSpindleCoarseReached，而接口信号“正在进行同步” <Axis>.coupling.in.syncSpindleProcessActive 将被删除：

MD37200 \$MN_COUPLE_POS_TOL_COARSE

差值 > ..TOL_FINE

如果同步运行偏差小于以下机床数据，则接口中出现接口信号“精同步” <Axis>.coupling.in.syncSpindleFineReached：

MD37210 \$MA_COUPLE_POS_TOL_FINE

EG 级联的同步运行偏差

EG 级联的同步运行偏差是跟随轴实际位置与根据运动规律得出的设定位置的偏差，它取决于相关的真实轴。

示例：

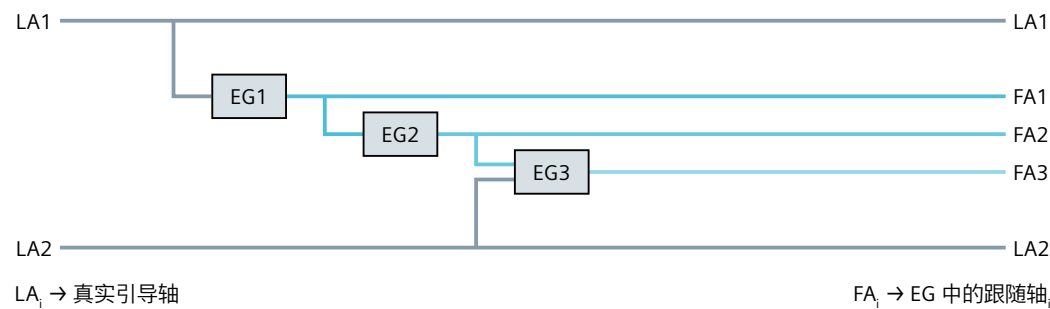


图 15-8 三级 EG 级联

以上示例中，跟随轴 FA3 的同步运行偏差是根据上面给出的定义由跟随轴实际值 FA3_{实际} 以及引导轴实际值 FA2_{实际} 和 LA2_{实际} 决定的，而不是由 LA1_{实际} 和 FA1_{实际}。

如果 FA2 不是真实轴，则实际值 FA2_{实际} 不可用。在这种情况下，必须使用轴的**设定值**而非实际值，此时该值只能从引导轴实际值 FA1_{实际} 推导出来。

其他信号

当主运行中出现 EGON()、EGONSYN() 或 EGONSYNE() 程序段时，为跟随轴设置“耦合生效”信号。如果仅叠加跟随轴，则设置“耦合生效”和“轴叠加”信号。如果 EGON()、EGONSYN() 或 EGONSYNE() 生效，也叠加跟随轴，则同样设置“耦合生效”和“轴叠加”信号。

接口信号“副主轴生效” <Axis>.coupling.in.syncSpindleSlaveActive: 耦合生效，

接口信号“叠加运动” <Axis>.coupling.in.syncSpindleMotionOverlaid: 叠加轴，

接口信号“跟随轴叠加使能” <Axis>.grinding.out.enableFollowOverlay

使用 EGONSYN()、EGONSYNE() 指令时，必须存在“跟随轴叠加使能”信号，以便可以根据指定的跟随轴同步位置同步齿轮箱。如果不存在，则出现报警 16771“叠加运动未使能”。如果存在该信号，则跟随轴根据其计算的加速度和速度逼近同步位置，必要时根据逼近模式。

其他监控信号

机床数据 MD37550 \$MA_EG_VEL_WARNING 可用于在机床数据 MD32000 \$MA_MAX_AX_VELO 和 MD32300 \$MA_MAX_AX_ACCEL 中指定速度和加速度相对于跟随轴的百分比，其中生成以下接口信号：

接口信号 “速度警告极限” <Axis>.coupling.in.egSpeedThresholdReached

接口信号 “加速度警告极限” <Axis>.coupling.in.egAccelThresholdReached

在机床数据 MD37560 \$MA_EG_ACC_TOL 中，可以基于跟随轴的机床数据 MD32300 \$MA_MAX_AX_ACCEL 定义一个百分比，当超过该百分比时，将生成接口信号 “轴加速” <Axis>.coupling.in.egAccelReached。

查询同步运行偏差

1. 同步运行偏差的计算结果可以通过系统变量 \$VA_EG_SYNCDIFF 读取为零件程序中的值。系统变量 \$VA_EG_SYNCDIFF_S 中提供具有符号的相应值。含义是：

- 负值（对于正向运动的引导轴和跟随轴）：跟随轴滞后于其计算的设定位置。
- 正值（对于正向运动的引导轴和跟随轴）：跟随轴超前于其计算的设定位置（过冲）。

带符号的同步运行偏差值对应于 \$VA_EG_SYNCDIFF 中不带符号的系统变量。

$$\$VA\_EG\_SYNCDIFF[ax] = ABS(\$VA\_EG\_SYNCDIFF\_S[ax])$$

程序段转换模式

1. 启用 EG 耦合组时，可以指定应在哪些条件下触发零件程序执行过程中的程序段转换：
2. 通过字符串参数进行指定，含义如下：
3. “NOC”：立即进行程序段转换
4. “FINE”：存在“精同步”时进行程序段转换
5. “COARSE”：存在“粗同步”时进行程序段转换
6. “IPOSTOP”：存在“设定值侧同步”时进行程序段转换

说明

在激活调用 EGON、EGONSYN、EGONSYNE 中编程时，可以将指定的字符串缩短为前两个字符。

如果尚未为 EG 轴组定义程序段转换模式，且当前没有指定，则“FINE”适用。

15.4 电子齿轮箱 (EG)

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.feedStopSpindleStop	LBP_Axis*.A_FDSpStop	DB31, ...DBX4.3
<Axis>.grinding.out.enableFollowOverlay	LBP_Axis*.A_FollowAxOv	DB31, ...DBX26.4

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.coupling.in.syncSpindleFineReached	LBP_Axis*.E_SyncronFine	DB31, ...DBX98.0
<Axis>.coupling.in.syncSpindleCoarseReached	LBP_Axis*.E_SyncronCoarse	DB31, ...DBX98.1
<Axis>.coupling.in.syncSpindleMotionOverlaid	LBP_Axis*.E_OverlayMotion	DB31, ...DBX98.4
<Axis>.coupling.in.egSpeedThresholdReached	LBP_Axis*.E_SWL	DB31, ...DBX98.5
<Axis>.coupling.in.egAccelThresholdReached	LBP_Axis*.E_AWL	DB31, ...DBX98.6
<Axis>.coupling.in.syncSpindleSlaveActive	LBP_Axis*.E_SlaveSp	DB31, ...DBX99.1
<Axis>.coupling.in.egAccelReached	LBP_Axis*.E_AxisAccel	DB31, ...DBX99.3
<Axis>.coupling.in.syncSpindleProcessActive	LBP_Axis*.E_Sync	DB31, ...DBX99.4

15.4.3 定义一个 EG 轴组

说明

电子齿轮箱的以下定义指令和换挡指令在零件程序中必须**单独在一个程序段中**。

电子齿轮箱的所有指令都会触发**预处理程序停止**，激活指令除外：

- EGON
- EGONSYN
- EGONSYNE

定义和激活

下面描述的定义和激活是独立的过程。如果没有事先定义则无法激活。

定义一个 EG 轴组

EG 轴组通过指定跟随轴和至少一个，最多五个具有相应耦合类型的引导轴来定义：

EGDEF(跟随轴, 引导轴 1, 耦合类型 1, 引导轴 2, 耦合类型 2,...)

耦合类型不必对所有引导轴都相同，因此必须为每个引导轴单独指定。

耦合类型：

分析引导轴的实际值：0

分析引导轴的设定值：1

对 EG 耦合组进行定义时，必须预设耦合系数为零。这意味着在激活之前对跟随轴没有影响。（参见 EGON、EGONSYN、EGONSYNE）。

EG 轴组定义的前提：

对于跟随轴还不允许定义轴耦合。（必要时必须提前用 EGDEL 删除现有轴耦合。）

EGDEF 触发预处理程序停止，并发出错误消息。

关于 EG 齿轮箱在滚齿加工中的应用示例请参见“示例”章节，“用于滚齿加工的电子齿轮箱”。

EGDEF

即使有一个或多个引导轴通过曲线表影响跟随轴，用 EGDEF 进行的齿轮定义在使用时也必须保持不变。

通过曲线表扩展了非线性耦合的类型参见“使用非线性分量的扩展示例”一章中的扩展示例。

15.4.4 启用一个 EG 轴组

无同步

EG 耦合组在**未选择同步**的情况下启用，使用：

EGON(FA, 程序段转换模式, LA1, Z1, N1, LA2 , Z2, N2,...LA5, Z5, N5.)

耦合将立即激活。

其中：

FA：跟随轴

根据程序段转换模式继续切换下一个程序段：

“NOC”：立即进行程序段转换

“FINE”：在“精同步”时进行程序段转换

15.4 电子齿轮箱 (EG)

“COARSE”：在“粗同步”时进行程序段转换

“IPOSTOP”：在设定值侧同步运行时进行程序段转换

LA_i ：引导轴 i 的轴名称

Z_i ：引导轴 i 耦合系数的分子

N_i ：引导轴 i 耦合系数的分母

只允许对先前用 EGDEF 进行详细说明的引导轴编程。必须至少编程一个引导轴。

将引导轴以及跟随轴启动时刻的位置作为“同步位置”保存。可以使用系统变量 \$AA_EG_SYN 读入“同步位置”。

有同步

EG 耦合组在已选择同步的情况下启用，使用：

1. EGONSYN

EGONSYN(FA, 程序段转换模式, SynPosFA, LA_i , SynPos LA_i , Z_{LA_i} , N_{LA_i})

其中：

FA：跟随轴

程序段转换模式：

“NOC”：立即进行程序段转换

“FINE”：在“精同步”时进行程序段转换

“COARSE”：在“粗同步”时进行程序段转换

“IPOSTOP”：在设定值侧同步运行时进行程序段转换

SynPosFA：跟随轴的同步位置

LA_i ：引导轴 i 的轴名称

SynPos LA_i ：引导轴 i 的同步位置

Z_i ：引导轴 i 耦合系数的分子

N_i ：引导轴 i 耦合系数的分母

说明

必须为至少一个，最多五个引导轴指定以 i 为索引的参数。

只允许对先前用 EGDEF 进行详细说明了的引导轴编程。

通过为跟随轴 (SynPosFA) 和引导轴 (SynPosLA) 编程的“同步位置”，定义耦合组被视为*同步*的位置。一旦启用时电子齿轮箱不处于同步状态，**跟随轴**就运行到定义的同步位置。

同步位置的位置指定采用设置的基本单位制，与可编程尺寸说明 (G70/G71) 无关。

如果耦合组中有模数轴，则其位置值会模数减少。这样可以可靠地逼近最快的同步位置，例如，由齿间距 ($360 \text{ 度} * Z_i / N_i$) 和相关同步位置得出的下一个齿。
(所谓的*相对同步*。)

逼近最快的同步位置。(所谓的*相对同步*，例如“定中心”后的下一个齿隙。)

如果没有为跟随轴发出以下接口信号，则不会运行到同步位置：

`<Axis>.coupling.out.enableFollowOverlay` (跟随轴叠加使能)

与此相反，程序在 EGONSYN 程序段处停止，报告自删除报警 16771，直到设置了上述信号。

2. EGONSYNE

EGONSYNE(FA, 程序段转换模式, SynPosFA, Anfahrmodus, LA_i, SynPosLA_i, Z_LA_i, N_LA_i)

其中：

“FA”：跟随轴

程序段转换模式：

“NOC”：立即进行程序段转换

“FINE”：在“精同步”时进行程序段转换

“COARSE”：在“粗同步”时进行程序段转换

“IPOSTOP”：在设定值侧同步运行时进行程序段转换

SynPosFA：跟随轴的同步位置

逼近模式：

“NTGT”：NextToothGapTime optimized，以时间最优的方式逼近下一个齿隙（如果未指定，预设值有效。）

“NTGP”：NextToothGapPath optimized，以行程最优的方式逼近下一个齿隙

“ACN”：AbsolutCoordinateNegativ，绝对尺寸指定，回转轴沿轴的负旋转方向运行

“ACP”：AbsoluteCoordinatePositiv，绝对尺寸指定，回转轴沿轴的正旋转方向运行

“DCT”：DirectCoordinateTime optimized，绝对尺寸指定，回转轴以时间最优的方式运行到编程的同步位置

15.4 电子齿轮箱 (EG)

“DCP”: DirectCoordinatePath optimized, 绝对尺寸指定, 回转轴以行程最优的方式运行到编程的同步位置

LA_i: 引导轴 i 的轴名称

SynPosLA_i: 引导轴 i 的同步位置

Z_i: 引导轴 i 耦合系数的分子

N_i: 引导轴 i 耦合系数的分母

说明

必须为至少一个, 最多五个引导轴指定以 i 为索引的参数。

该功能仅影响耦合到模数引导轴的模数跟随轴。

齿间距

齿间距由 $360 \text{ 度} * Z_i / N_i$ 得出

示例:

EGONSYNE(A, "FINE", FASynPos, "运行模式", B, LASynPos, 2, 10)

齿间距: $360 * 2 / 10 = 72 \text{ (度)}$

跟随轴静止时的逼近特性

在这种情况下, 时间最优和行程最优的运行模式是相同的。

下表显示了目标位置和带有方向标记 (括号中) 的运行路径, 具体取决于逼近模式:

编程的同步位置 FaSysPos	EGONSYNE 前 的跟随轴位置	逼近模式 NTGT/NTGP	逼近模式 DCT/DCP	逼近模式 ACP	逼近模式 ACN
110	150	182 (+32)	110 (-40)	110 (+320)	110 (-40)
110	350	326 (-24)	110 (+120)	110 (+120)	110 (-240)
130	0	346 (-14)	130 (+130)	130 (+130)	130 (-230)
130	30	58 (+28)	130 (+100)	130 (+100)	130 (-260)
130	190	202 (+12)	130 (-60)	130 (+300)	130 (-60)
190	0	334 (-26)	190 (-170)	190 (+190)	190 (-170)
230	0	14 (+14)	230 (-130)	230 (+230)	230 (-130)

跟随轴运行时的逼近特性

通过 EGONSUNE 激活耦合时，跟随轴大约以其最大轴速度正向运动。假设跟随轴的编程同步位置为 110，当前位置为 150。这就产生了两个备选的同步位置 110 和 182（见上表）。

采用运行模式 NTGP（行程最优）时，无论当前速度如何，都会选择同步位置 182。它与跟随轴当前位置的距离最短。采用运行模式 NTGT（时间最优）时，考虑跟随轴的当前速度，并基于轴的最大速度造成的限制实现制动，以在尽可能短的时间内实现同步运行（见图）。

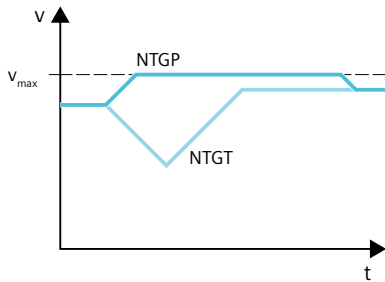


图 15-9 以行程最优（上）与时间最优（下）的方式到达跟随轴的下一个齿隙

示例符号

EGONSUNE(A, "FINE", 110, "NTGT", B, 0, 2, 10)

将 A 耦合到 B，同步位置 A = 110，B = 0，耦合系数 2/10，逼近模式 = NTGT

EGONSUNE(A, "FINE", 110, "DCT", B, 0, 2, 10)

将 A 耦合到 B，同步位置 A = 110，B = 0，耦合系数 2/10，逼近模式 = DCT

EGONSUNE(A, "FINE", 110, "NTGT", B, 0, 2, 10, Y, 15, 1, 3)

将 A 耦合到 B 和 Y，同步位置 A = 110，B = 0，Y = 15，

与 B 的耦合系数 = 2/10，与 Y 的耦合系数 = 1/3，

逼近模式 = NTGT

有同步

上面给出的语法有以下含义差异：

如果将一个**曲线表**用于引导轴之一，则必须将线性耦合的耦合系数的分母 (N_i) 设置为 0（分母 0 原不允许用于线性耦合）。

分母零对于控制系统而言表示，耦合系数的分子 (Z_i) 应解释为要使用的曲线表的编号。带有指定编号的曲线表在启用时刻必须已经定义（根据“曲线表”章节）。

引导轴 (LA_i) 的指定与通过耦合系数耦合时的引导轴指定一样（线性耦合）。

15.4 电子齿轮箱 (EG)

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.coupling.out.enableFollowOverlay	LBP_Axis*.A_FollowAxOv	DB31, ...DBX26.4

15.4.5 关闭一个 EG 轴组

方法 1

对于关闭一个有效的 EG 轴组，有不同方法。

EGOFS（跟随轴）

关闭电子齿轮箱。跟随轴制动到停止。此调用触发预处理停止。

方法 2

指令的以下参数设定允许有**选择性地**禁止各个引导轴对跟随轴的运动的影响。

EGOFS(跟随轴, 引导轴 1, ... 引导轴 5)

说明

必须至少指定一个引导轴。

有针对性地关闭指定引导轴对跟随轴的影响。此调用触发预处理停止。

如果尚有引导轴保持激活状态，则跟随轴将在其影响下继续运行。如果所有的引导轴影响都以这种方式关闭，则跟随轴制动到停止。

如果有选择性地关闭 EGONSYN 指令，则不执行轴运动。

方法 3

EGOFC（副主轴）

关闭电子齿轮箱。副主轴以关闭时刻有效的转速/速度继续运行。此调用触发预处理停止。

说明

调用仅适用于副**主轴**。使用 EGOFC 时必须编程主轴名称。

15.4.6 删除一个 EG 轴组

必须先根据“关闭一个 EG 轴组”章节关闭 EG 轴组，然后才能删除其定义。

EGDEL (跟随轴)

轴组的耦合定义将被删除。可以使用 EGDEF 重新定义更多轴组，直到达到同时激活的轴组的最大数量。

此调用触发预处理停止。

15.4.7 旋转进给 (G95) 与电子齿轮箱的共同作用

使用零件程序指令 FPR() 也可以将一个电子齿轮箱的跟随轴指定为旋转进给的进给率决定轴。对于这种情况，以下特性适用：

- 进给取决于电子齿轮箱跟随轴的设定速度。
- 设定速度可以由主轴和模数引导轴（不是轨迹轴）的速度和相应的耦合系数计算出来。
- 其他引导轴的速度比例和跟随轴的叠加运动不予考虑。

更多信息

进给率 (页 201)

15.4.8 上电、复位、运行方式切换、搜索时的特性

功能	特性，关于电子齿轮箱	
	耦合状态	配置
运行方式切换	保留	保留
零件程序结束	保留	保留
复位	保留	保留
上电 ¹⁾	不保留	不保留
1) 在上电之后没有耦合生效。		

程序段搜索

在以下边界条件下，可以对激活的耦合 (EG) 进行程序段搜索：

- 仅通过设定值耦合进行模拟。
- 所有跨通道引导轴不得锁定。
- 轴运动的所有真实位置都为 NC 所知。

如果在具有生效耦合的零件程序部分中，允许目标程序段用于采用计算或 SERUPRO 的程序段搜索在技术上没有意义或不可行，则可以阻止该部分重新启动。

编程

- IPTRLOCK()
- IPTRUNLOCK()

更多信息

功能手册之基本功能：运行方式组、通道、程序运行、复位特性，程序段搜索类型 5
SERUPRO > 阻止程序段重新启动

15.4.9 电子齿轮箱的系统变量

应用

利用以下系统变量，零件程序可以检测 EG 轴组的当前状态，必要时加以响应：

表格 15-1 系统变量，R 表示：允许读访问

名称	类型	访问		预处理停止		含义，值	含义，下标
		零件程序	同步动作	零件程序	同步动作		
\$AA_EG_TYPE[a,b]	INT	R		R		耦合方式： 0：实际值耦合 1：设定值耦合	轴名称 a：跟随轴 b：引导轴
\$AA_EG_NUMERA[a,b]	REAL	R		R		耦合系数 KF 的分子 KF = 分子/分母 默认设置：0 \$AA_EG_DENOM[a,b] 为 0 时曲线表的编号。	轴名称 a：跟随轴 b：引导轴

名称	类型	访问		预处理停止		含义，值	含义，下标
		零件程序	同步动作	零件程序	同步动作		
\$AA_EG_DENOM[a,b]	REAL	R		R		耦合系数 KF 的分母 KF = 分子/分母 默认设置：1 分母必须为正。 如果指定了曲线表的编号而不是分子 \$AA_EG_NUMERA[a,b]，则分母为 0。	轴名称 a: 跟随轴 b: 引导轴
\$AA_EG_SYN[a,b]	REAL	R		R		指定引导轴的同步位置 默认设置：0	轴名称 a: 跟随轴 b: 引导轴
\$AA_EG_SYNFA[a]	REAL	R		R		指定跟随轴的同步位置 默认设置：0	轴名称 a: 跟随轴
\$P_EG_BC[a]	STRING	R		R		EG 激活调用的程序段转换标准：EGON、EGONSYN： “NOC”：立即 “FINE”：精同步 “COARSE”：粗同步 “IPOSTOP”：设定值侧同步	轴名称 a: 跟随轴
\$AA_EG_NUM_LA[a]	INT	R		R		用 EGDEF 定义的引导轴数量。如果没有使用 EGDEF 将轴指定为跟随轴，则为 0。	轴名称 a: 跟随轴
\$AA_EG_AX[n,a]	AXIS	R		R		指定了下标 n 的引导轴的轴名称。	轴名称 n: EG 组中引导轴的下标 0 ... 4 a: 跟随轴

15.4 电子齿轮箱 (EG)

名称	类型	访问		预处理停止		含义，值	含义，下标
		零件程序	同步动作	零件程序	同步动作		
\$AA_EG_ACTIVE[a,b]	BOOL	R		R		确定一个引导轴的启用状态： 0：关闭 1：启用	轴名称 a：跟随轴 b：引导轴
\$VA_EG_SYNCDIFF[a]	REAL	R	R	R		同步运行偏差的实际值。与机床数据 MD37200 \$MA_COUPLE_POS_TOL_COARSE 和 MD37210 \$MA_COUPLE_POS_TOL_FINE 的比较提供接口信号。	轴名称 a：跟随轴

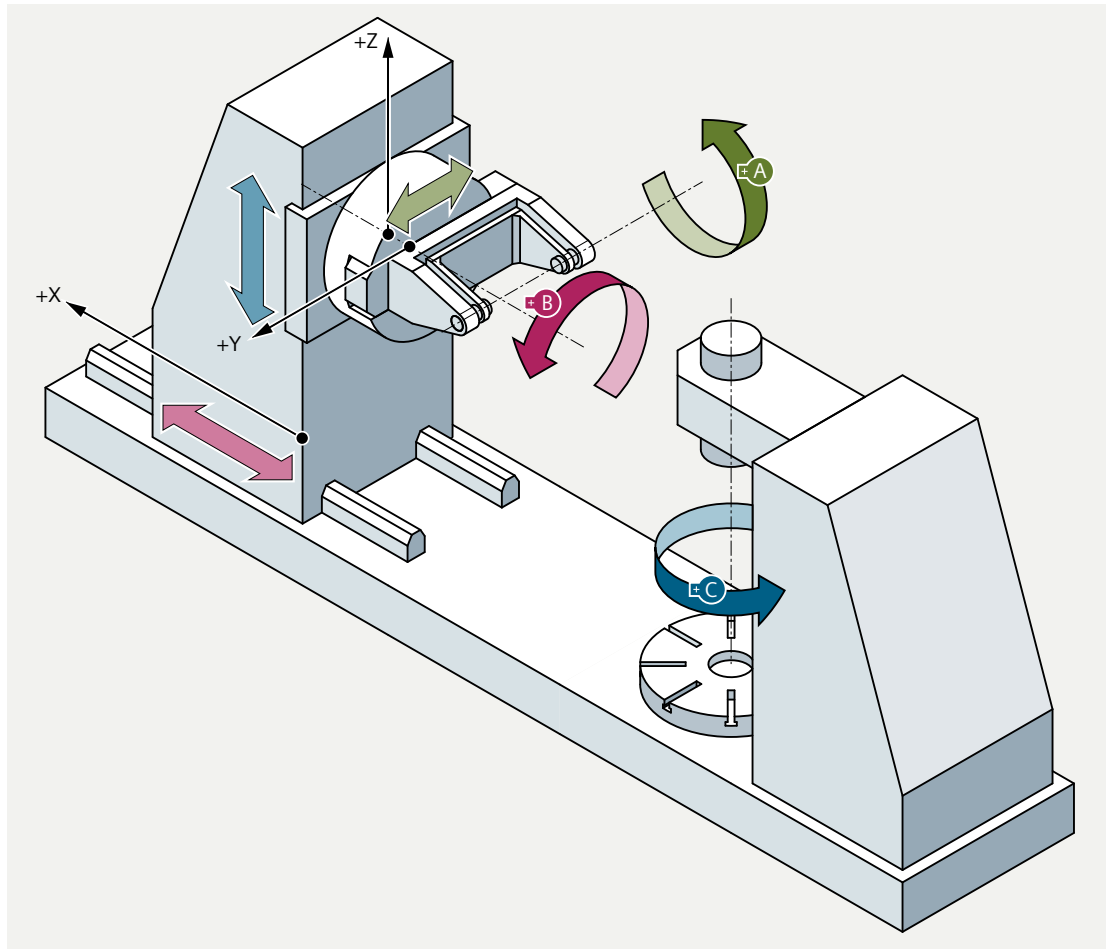
15.4.10 示例

15.4.10.1 使用线性耦合的示例

轴的使用

下图显示了典型滚齿机的配置。该机器包括五个数控轴和一个受控主主轴。分别为：

- 工件台的旋转运动 (C) 和滚铣刀的旋转运动 (B)
- 用于在整个工件宽度上产生进给运动的轴向轴 (Z)
- 用于将滚铣刀沿其轴线移动的切向轴 (Y)
- 用于将铣刀进给到齿深的径向轴 (X)
- 用于根据铣刀螺旋角和齿倾斜角调整工件滚铣刀的铣刀摆动轴 (A)

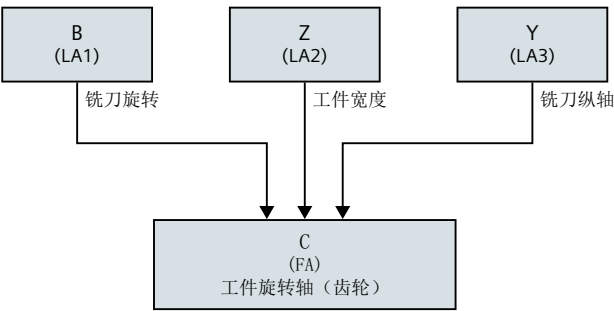


- X 径向轴
- Y 切向轴（主驱动 3）
- Z 轴向轴（主驱动 2）
- A 铣刀摆动轴
- C 工件旋转轴（从动装置）
- B 铣刀旋转轴，主主轴（主驱动 1）

图 15-10 滚齿机的轴定义（示例）

对于滚齿机，产生以下功能关系：

15.4 电子齿轮箱 (EG)



工件台轴 (C) 是跟随轴，在此示例中受三个主驱动的影响。

跟随轴的设定值通过以下关系等式循环确定：

$$n_c = n_b * (z_0 / z_2) + v_z * (u_{dz} / z_2) + v_y * (u_{dy} / z_2)$$

- 其中：
- n_c = 工件轴 (C) 的转速
 - n_b = 铣刀主轴 (B) 的转速
 - z_0 = 滚铣刀的螺纹数
 - z_2 = 工件的齿数
 - v_z = 轴向轴 (Z) 的进给速度
 - v_y = 切向轴 (Y) 的进给速度
 - u_{dz} = 轴向差动常数
 - u_{dy} = 切向差动常数

对工件轴 C 设定值的影响

上述等式的第一个加数决定了工件台和铣刀的转速比，从而决定了工件的齿数。

第二个加数在斜齿的情况下会导致 C 轴根据铣刀的轴向进给运动进行必要的额外旋转，以产生斜齿的斜度。

通过第三部分，还考虑了 C 轴的附加旋转，它补偿了铣刀对工件的切向运动。这样可以实现刀具在整个长度上的均匀应力。

工件/刀具参数

z_0 、 z_2 、 u_{dz} 和 u_{dy} 值取决于工件或刀具，由 NC 操作员或在零件程序中指定。

差动常数

差动常数 u_{dz} 和 u_{dy} 考虑了工件的斜齿斜度和铣刀的几何形状。差动常数可以在用户特定的周期内确定。

$$u_{dz} = (\sin \beta^\circ / (m_n * \pi)) * 360$$
$$u_{dy} = (\cos \gamma^\circ / (m_n * \pi)) * 360$$

[度 / mm]

[度 / mm]

其中： m_n = 法向模数（单位 mm）
 β° = 齿轮的倾斜角
 γ° = 滚铣刀的螺旋角

零件程序段

程序代码	注释
EGDEF(C,B,1,Z,1,Y,1)	; 定义 EG 轴组，设定值耦合 (1) 从 B、Z、Y 到 C（跟随轴）。
EGON(C,"FINE",B,z0,z2,Z,udz,z2,Y,udy,z2)	; 启用耦合。
...	

15.4.10.2 包含非线性部分的扩展示例

简介

以下示例为示例（参见“图 15-10 滚齿机的轴定义（示例）（页 641）”）扩展了以下内容：

- 非线性取决于 Z 轴的机器误差补偿，以及
- 齿几何中的 Z 轴相关分量。

例如，以此可以将齿轮中心的齿面设计成略微隆起的形状，以便在操作过程中，齿的中心比齿的边缘承受更大的负载。

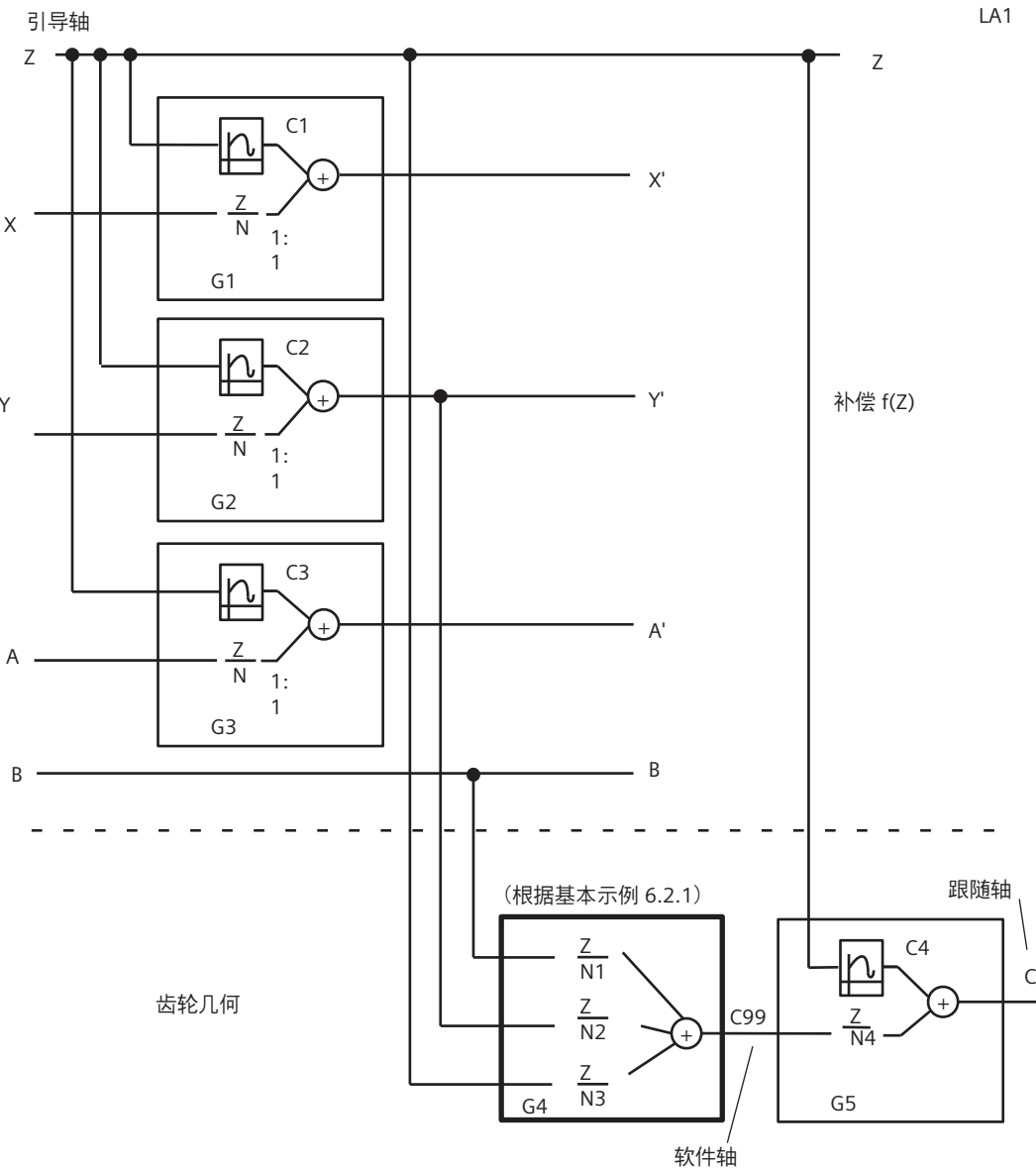


图 15-11 包含非线性机器误差补偿和齿几何上非线性分量的扩展示例

以下零件程序段应理解为一般概念，仍需对曲线表和齿轮/机床参数进行具体补充。要补充的部分用 <...> 注明。具体注明的参数在必要时也必须更改，例如耦合系数。

```

N100 CTABDEF(X, Z, 1, 0) ; 非周期性曲线表 C1 的声明和技术数据
N110 < ...                ; 曲线表的规定：曲线点或多项式程序段
      >
N190 CTABEND
N200 CTABDEF(Y, Z, 2,      ; 非周期性曲线表 C2 的声明和技术数据
0)
N210 < ...                ; 曲线表的规定：曲线点或多项式程序段
      >
N290 CTABEND
N300 CTABDEF(A, Z, 3, 0) ; 非周期性曲线表 C3 的声明和技术数据
N310 < ...                ; 曲线表的规定：曲线点或多项式程序段
      >
N390 TABEND
N400 CTABDEF(C, Z, 4, 0) ; 非周期性曲线表 C4 的声明和技术数据
N410 < ...                ; 曲线表的规定：曲线点或多项式程序段
      >
N490 CTABEND
N500 EGDEF(X, Z, 1)      ; 通过 C1 声明路径，设定值耦合
N510 G1 F1000 X10        ; 声明 X 的指令部分
N520 EGONSYN(X, "NOC",   ; 通过 C1 启用路径
      <SynPosX>, Z,
      <SynPosX_Z>, 1, 0)
N600 EGDEF(Y, Z, 1)      ; 通过 C2 声明路径，设定值耦合
N610 G1 F1000 Y10        ; 声明 Y 的指令部分
N620 EGONSYN(Y,          ; 通过 C2 启用路径
      "COARSE",
      <SynPosY>, Z,
      <SynPosY_Z>, 2, 0)
N700 EGDEF(A, Z, 1)      ; 通过 C3 声明路径，设定值耦合
N710 G1 F1000 A10        ; 声明 A 的指令部分
N720 EGONSYN(A, "FINE",  ; 通过 C3 启用路径
      <SynPosA>, Z,
      <SynPosA_Z>, 3, 0)
; 第 1 个齿轮档，C99 是两个电子齿轮箱之间的软件轴
N800 EGDEF(C99, Y, 1, Z,
      1, B, 1 )
N810 EGONSYN(C99, "NOC", ; 启用引导轴 B
      <SynPosC99>, B,
      <SynPosC99_B>,
18.2, &
      Y,
      ; 启用引导轴 Y
      <SynPosC99_Y>, R1 * π, 1, &

```

15.4 电子齿轮箱(EG)

```

      Z,                ; 启用引导轴 Z
<SynPosC99_Z>,10, 1)
                        ; 符号 & 表示: 指令在后续行中继续, 程序中不允许有 LF 和注释
; 第 2 个齿轮档
N900 EGDEF(C, C99, 1,   ; 将第 1 档的跟随轴 C99 声明为第 2 档的引导轴,
Z, 1)                  ; 设定值耦合
N910                   ; 通过 C4 声明路径, 设定值耦合
N900                   ; 启用软件轴 C99
N910                   ; 和引导轴 Z, 通过 C4
N920 EGONSYN(C, "NOC",
      <SynPosC>, C99,
      <SynPosC_C99>, 1,
1, &
      Z, <SynPosC_Z>, 4,
0)
N999 M30
```

机床数据

仅指定了超出几何/通道和机床轴参数的必要配置的一部分。

\$MN_NUM_EG = 5	; 齿轮箱的最大数量
\$MN_MM_NUM_CURVE_TABS = 5	; 曲线表的最大数量
\$MN_MM_NUM_CURVE_SEGMENTS = 50	; 曲线段的最大数量
\$MN_MM_NUM_CURVE_POLYNOMS = 100	; 曲线多项式的最大数量

设置数据

如果使用“电子齿轮箱(EG)(页 621)”章节中描述的缩放，则以下机床数据中的函数值会根据偏移而变化：

MD43108 \$SD_LEAD_SCALE_OUT_POS[4] = 1.2 ; 表 C4 的缩放

系统变量

根据上面给出的定义，控制系统在分配的系统变量中输入以下值。

更多信息

参数手册之系统变量

下面列出的系统变量只能理解为**注释**！

```
; ***** 齿轮箱 X (G1)
$AA_EG_TYPE[X, Z] = 1           ; 设定值耦合
$AA_EG_NUMERA[X, Z] = 1         ; 曲线表编号 = 1
$AA_EG_DENOM[X, Z] = 0          ; 分母 = 0 → 曲线表适用
$P_EG_BC[X] = "NOC"             ; 程序段转换标准
$AA_EG_NUM_LA[X] = 1            ; 引导轴数量
$AA_EG_AX[0, X] = Z             ; 引导轴名称
$AA_EG_SYN[X, Z] = <SynPosX_Z>  ; 引导轴 Z 的同步位置
$AA_EG_SYNFA[X] = <SynPosX>     ; 跟随轴的同步位置
; ***** 齿轮箱 Y (G2)
$AA_EG_TYPE[Y, Z] = 1           ; 设定值耦合
$AA_EG_NUMERA[Y, Z] = 2         ; 曲线表编号 = 2
$AA_EG_DENOM[Y, Z] = 0          ; 分母 = 0 → 曲线表适用
$P_EG_BC[Y10] = "COARSE"        ; 程序段转换标准
$AA_EG_NUM_LA[Y] = 1            ; 引导轴数量
$AA_EG_AX[0, Y] = Z             ; 引导轴名称
$AA_EG_SYN[Y, Z] = <SynPosY_Z>  ; 引导轴 Z 的同步位置
$AA_EG_SYNFA[Y] = <SynPosY>     ; 跟随轴的同步位置
; ***** 齿轮箱 A (G3)
$AA_EG_TYPE[A, Z] = 1           ; 设定值耦合
$AA_EG_NUMERA[A, Z] = 3         ; 曲线表编号 = 3
$AA_EG_DENOM[A, Z] = 0          ; 分母 = 0 → 曲线表适用
$P_EG_BC[A10] = "FINE"          ; 程序段转换标准
$AA_EG_NUM_LA[A] = 1            ; 引导轴数量
$AA_EG_AX[0, A] = Z             ; 引导轴名称
$AA_EG_SYN[A, Z] = <SynPosA_Z>  ; 引导轴 Z 的同步位置
$AA_EG_SYNFA[A] = <SynPosA>     ; 跟随轴的同步位置
; ***** 齿轮箱 C99 (G4)
$AA_EG_TYPE[C99, Y] = 1         ; 设定值耦合
$AA_EG_NUMERA[C99, Y] = 18      ; 耦合系数y 的分子
$AA_EG_DENOM[C99, Y] = 2        ; 耦合系数y 的分母
```

15.4 电子齿轮箱 (EG)

\$AA_EG_TYPE[C99, Z] = 1	; 设定值耦合
\$AA_EG_NUMERA[C99, Z] = R1 * π	; 耦合系数 z 的分子
\$AA_EG_DENOM[C99, Z] = 1	; 耦合系数 z 的分母
\$AA_EG_TYPE[C99, B] = 1	; 设定值耦合
\$AA_EG_NUMERA[C99, B] = 10	; 耦合系数 b 的分子
\$AA_EG_DENOM[C99, B] = 1	; 耦合系数 b 的分母
\$P_EG_BC[C99] = "NOC"	; 程序段转换标准
\$AA_EG_NUM_LA[C99] = 3	; 引导轴数量
\$AA_EG_AX[0, C99] = Y	; 引导轴 Y 的名称
\$AA_EG_AX[1, C99] = Z	; 引导轴 Z 的名称
\$AA_EG_AX[2, C99] = B	; 引导轴 B 的名称
\$AA_EG_SYN[C99, Y] = <SynPosC99_Y>	; 引导轴 Y 的同步位置
\$AA_EG_SYN[C99, Z] = <SynPosC99_Z>	; 引导轴 Z 的同步位置
\$AA_EG_SYN[C99, B] = <SynPosC99_B>	; 引导轴 B 的同步位置
\$AA_EG_SYNFA[C99] = <SynPosC99>	; 跟随轴的同步位置
; ***** 齿轮箱 C (G5)	
\$AA_EG_TYPE[C, Z] = 1	; 设定值耦合
\$AA_EG_NUMERA[C, Z] = 4	; 曲线表编号 = 4
\$AA_EG_DENOM[C, Z] = 0	; 分母 = 0 → 曲线表适用
\$AA_EG_TYPE[C, C99] = 1	; 设定值耦合
\$AA_EG_NUMERA[C, C99] = 1	; 耦合系数 c_{99} 的分子
\$AA_EG_DENOM[C, C99] = 1	; 耦合系数 c_{99} 的分母
\$P_EG_BC[C] = "NOC"	; 程序段转换标准
\$AA_EG_NUM_LA[C] = 2	; 引导轴数量
\$AA_EG_AX[0, C] = Z	; 引导轴 Z 的名称
\$AA_EG_AX[1, C] = C99	; 引导轴 C99 的名称
\$AA_EG_SYN[C, Z] = <SynPosC_Z>	; 引导轴 Z 的同步位置
\$AA_EG_SYN[C, C99] = <SynPosC_C99>	; 引导轴 C99 的同步位置
\$AA_EG_SYNFA[C] = <SynPosC>	; 引导轴 C 的同步位置

机床数据

机床数据选段:

; ***** 通道 1


```

CHANDATA(1)
; ***** 轴 1, "X"
$MC_AXCONF_GEOAX_NAME_TAB[0] = "X"
$MC_AXCONF_CHANAX_NAME_TAB[0] = "X"
$MC_AXCONF_MACHAX_USED[0] = 1
$MN_AXCONF_MACHAX_NAME_TAB[0] = "X1"
$MA_SPIND_ASSIGN_TO_MACHAX[AX1] = 0
$MA_IS_ROT_AX[AX1] = FALSE
; ***** 轴 2, "Y"
$MC_AXCONF_GEOAX_NAME_TAB[1] = "Y"
$MC_AXCONF_CHANAX_NAME_TAB[1] = "Y"
$MC_AXCONF_MACHAX_USED[1] = 2
$MN_AXCONF_MACHAX_NAME_TAB[1] = "Y1"
$MA_SPIND_ASSIGN_TO_MACHAX[AX2] = 0
$MA_IS_ROT_AX[AX2] = FALSE
; ***** 轴 3, "Z"
$MC_AXCONF_GEOAX_NAME_TAB[2] = "Z"
$MC_AXCONF_CHANAX_NAME_TAB[2] = "Z"
$MC_AXCONF_MACHAX_USED[2] = 3
$MN_AXCONF_MACHAX_NAME_TAB[2] = "Z1"
$MA_SPIND_ASSIGN_TO_MACHAX[AX3] = 0
$MA_IS_ROT_AX[AX3] = FALSE
; ***** 轴 4, "A"
$MC_AXCONF_CHANAX_NAME_TAB[3] = "A"
$MC_AXCONF_MACHAX_USED[3] = 4
$MN_AXCONF_MACHAX_NAME_TAB[3] = "A1"
$MA_SPIND_ASSIGN_TO_MACHAX[AX4] = 0
$MA_IS_ROT_AX[AX4] = TRUE
$MA_ROT_IS_MODULO[AX4] = TRUE

```

15.4 电子齿轮箱 (EG)

```
; ***** 轴 5, "B"

$MC_AXCONF_CHANAX_NAME_TAB[4] = "B"
$MC_AXCONF_MACHAX_USED[4] = 5
$MC_SPIND_DEF_MASTER_SPIND = 1
$MN_AXCONF_MACHAX_NAME_TAB[4] = "B1"
$MA_SPIND_ASSIGN_TO_MACHAX[AX5] = 1
$MA_IS_ROT_AX[AX5] = TRUE
$MA_ROT_IS_MODULO[AX5] = TRUE

; ***** 轴 6, "C"

$MC_AXCONF_CHANAX_NAME_TAB[5] = "C"
$MC_AXCONF_MACHAX_USED[5] = 6
$MN_AXCONF_MACHAX_NAME_TAB[5] = "C1"
$MA_SPIND_ASSIGN_TO_MACHAX[AX6] = 0
$MA_IS_ROT_AX[AX6] = TRUE
$MA_ROT_IS_MODULO[AX6] = TRUE

; ***** 轴 10, "C99"

$MC_AXCONF_CHANAX_NAME_TAB[9] = "C99"
$MC_AXCONF_MACHAX_USED[9] = 10
$MA_SPIND_ASSIGN_TO_MACHAX[AX10] = 0
$MA_IS_ROT_AX[AX10] = TRUE
$MA_ROT_IS_MODULO[AX10] = TRUE
```

15.5 通用耦合

15.5.1 简要说明

15.5.1.1 功能

功能

“通用耦合”为通用的耦合功能，已有耦合类型（联动、引导值耦合、电子齿轮箱和同步主轴）的所有耦合属性都涵盖其中。

该功能可实现灵活的编程方式：

- 用户可根据自身应用自由选择必要的耦合属性（模块原理）。
- 每个耦合属性都可独立编程。
- 已定义耦合的属性（例如耦合系数）可以修改。
- 日后仍可添加其他耦合属性。
- 跟随轴的参考坐标系（基准坐标系或机床坐标系）也可编程。
- 某些耦合属性也可以在同步动作中编程。

更多信息

功能手册之同步动作

匹配循环

到目前为止，联动 (TRAIL*)、引导值耦合 (LEAD*)、电子齿轮箱 (EG*) 和同步主轴 (COUP*) 的耦合调用还继续由匹配循环支持（参见章节“匹配循环 (页 708)”）。

15.5.1.2 前提条件

CP 规格

通用耦合有两种规格：

- CP-BASIC
- CP-COMFORT

选择规格的决定性因素是同时需要的耦合（跟随轴/副主轴）数量及其属性。

15.5 通用耦合

示例：

如果连续需要 1 x 同步主轴对将零件从主轴转移到副主轴，然后进行 1 x 多边形车削，则 CP-BASIC 规格就足够适用。如果不排除这两个过程在时间上也有重叠（零件转移开始时，多边形车削仍在运行），则需要 CP-COMFORT 规格。

各个规格彼此独立。它们可以相互组合，也可以同时激活。

表格 15-2 根据规格不同，可以同时激活的耦合模块的组态范围

类型	CP 规格允许同时有一个或多个不同的 CPSETTYPE 耦合对象：	CP-BASIC	CP-COMFORT
A	联动	4	4
B	包含 1 :1 耦合的同步主轴	-	-
C	o./a.同步主轴/多边形车削 o./a.引导值耦合/曲线表插补 o./a.机床坐标系耦合	1	4
D	o./a.电子齿轮箱“单一” o./a.自由通用耦合“单一”	-	1
E	o./a.电子齿轮箱 o./a.自由通用耦合	-	-

o./a. 代表或/和

表格 15-3 耦合属性可用性中的缩放

类型 A	类型 B	类型 C	类型 D	类型 E	
20	1	13	9	5	CPSETTYPE 相关功能的最大数量（每种类型）
TRAIL - 联动					
20	-	13	9	5	具有以下属性的联动组最大数量： → 参见 CPSETTYPE="TRAIL" ¹⁾
1		2	2	2	最大引导值数量
-		-	+	+	主轴和轴之间的耦合
+		+	+	+	回转轴和线性轴之间的耦合
-		+	+	+	从零件程序和同步动作
+		+	+	+	允许叠加/差速

类型 A	类型 B	类型 C	类型 D	类型 E	
-		-	-	+	允许级联
BCS		BCS / MCS	BCS / MCS	BCS / MCS	参考坐标 (标准: CPFRS="BCS")
包含 1 :1 耦合的同步主轴					
-	1	-	-	-	具有以下属性的同步主轴/多边形车削最大数量: → 参见 CPSETTYPE="COUP" ¹⁾
	1				最大引导值数量
	-				从零件程序和同步动作
	-				允许叠加/差速
	-				允许级联
	MCS				固定参考坐标 (CPFRS="MCS")
COUP - 同步主轴/多边形车削					
-	-	13	9	5	具有以下属性的同步主轴/多边形车削最大数量: → 参见 CPSETTYPE="COUP" ¹⁾
		1	1	1	最大引导值数量
		-	-	-	从零件程序和同步动作
		+	+	+	允许叠加/差速
		-	-	-	允许级联
		MCS	MCS	MCS	固定参考坐标 (CPFRS="MCS")
LEAD - 引导值耦合/曲线表插补					
-	-	13	9	5	具有以下属性的引导值耦合/曲线表插补最大数量: → 参见 CPSETTYPE="LEAD" ¹⁾
		1	1	1	最大引导值数量
		+	+	+	从零件程序和同步动作
		+	+	+	允许叠加/差速
		-	-	+	允许级联
		BCS / MCS	BCS / MCS	BCS / MCS	参考坐标 (标准: CPFRS="BCS")
EG - 电子齿轮箱					
-	-	-	9	5	具有以下属性的 电子齿轮箱最大数量: → 参见 CPSETTYPE="EG" ¹⁾

15.5 通用耦合

类型 A	类型 B	类型 C	类型 D	类型 E	
			3	5	最大引导值数量
			-	-	从零件程序和同步动作
			+	+	允许叠加/差速
			+	+	允许级联
			BCS / MCS	BCS / MCS	参考坐标（标准：CPFRS="BCS"）
			-	+	允许非线性耦合规律 (CPLCTID)
CP - 自由通用耦合					
-	-	-	9	5	具有以下属性的 自由通用耦合最大数量： 标准（相当于 CPSETTYPE="CP" ¹⁾ ）
			3	5	最大引导值数量
			+	+	从零件程序和同步动作
			+	+	允许叠加/差速
			+	+	允许级联
			BCS / MCS	BCS / MCS	参考坐标（标准：CPFRS="BCS"）
			-	+	允许非线性耦合规律 (CPLCTID)
¹⁾ 参见 " 耦合类型 (CPSETTYPE) (页 709) "。					

说明

通用耦合不考虑现有耦合选项（引导值耦合、电子齿轮箱和同步主轴）。只有当耦合涉及不同的轴/主轴时，才能同时使用现有耦合选项和通用耦合。

存储器配置

在动态 NC 存储器中要为通用耦合预留的存储空间由以下机床数据定义：

MD18450 \$MN_MM_NUM_CP_MODULES（CP 耦合模块的最大允许数量）

MD18452 \$MN_MM_NUM_CP_MODUL_LEAD（CP 引导值的最大允许数量）

说明

建议：在调试时就设置此机床在最大配置中同时需要的最大预期值。

15.5.2 基本介绍

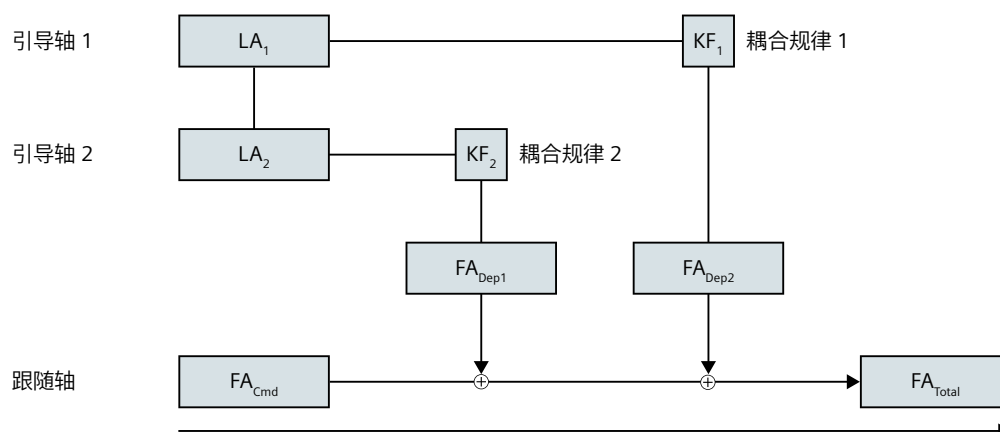
15.5.2.1 耦合模块

利用耦合模块，一个轴（→ 跟随轴）的运动可以根据其他轴（→ 引导轴）进行插补。

耦合规律

引导轴/引导值和跟随轴之间的关联按照每个引导轴/引导值通过耦合规律（耦合系数或曲线表）进行定义。各个引导轴/引导值的各运动分量是相加的。

通过以下示例（跟随轴与两个引导轴）说明这种关联：



FA_{Total}	跟随轴的总设定值
FA_{Cmd}	在零件程序中设置的设定值 = 跟随轴的独立运动分量
FA_{Dep1}	引导轴 1 的相关运动分量
FA_{Dep2}	引导轴 2 的相关运动分量
LA_1	第 1 个引导轴的设定值或实际值
LA_2	第 2 个引导轴的设定值或实际值
$SynPosLA_1$	第 1 个引导轴的同步位置
$SynPosLA_2$	第 2 个引导轴的同步位置
KF_1	第 1 个引导轴的耦合系数
KF_2	第 2 个引导轴的耦合系数

跟随轴位置由跟随轴的相关运动分量（ FA_{Dep1} 和 FA_{Dep2} ）和独立运动分量（ FA_{Cmd} ）叠加（求和）而得，前者由与引导轴的单独耦合关系产生。

15.5 通用耦合

$$FA_{Total} = FA_{Cmd} + FA_{DEP1} + FA_{DEP2}$$

跟随轴的运动分量如下得出：

引导轴 1 的相关运动分量：	$FA_{DEP1} = (LA_1 - SynPosLA_1) * KF_1$
引导轴 2 的相关运动分量：	$FA_{DEP2} = (LA_2 - SynPosLA_2) * KF_2$
跟随轴的独立运动分量：	FA_{Cmd}

跟随轴叠加

跟随轴的相关和独立运动分量的叠加称为跟随轴叠加。
可以使用所有可用的运动指令对跟随轴的独立运动分量进行编程。

15.5.2.2 关键字和耦合属性

关键字

现有的轴耦合通过语言指令进行编程，例如使用 TRAILON (X,Y,2) 编程联动。使用通用耦合时，用关键字替代语言指令。

这有以下优点：

- 耦合属性可以单独编程（参见下例）。
- 多个耦合的编程可以在一个程序段中进行（因为关键字不需要自己的程序段）。
 优点：缩短执行时间

示例：

使用现有耦合调用 TRAILON (X,Y,2) 设置的属性（跟随轴、引导轴和耦合系数）在通用耦合中由以下关键字定义：

$$CPON=(X1) \quad CPLA[X1]=(X2) \quad CPLNUM[X1,X2]=2$$

$CPON=(X1)$	启用与跟随轴 X1 的耦合。
$CPLA[X1]=(X2)$	将轴 X2 定义为引导轴。
$CPLNUM[X1,X2]=2$	将耦合系数的分子设置为 2。

表示方法

为明确分配，关键字标有前缀“CP”（代表 **Coupling**）。根据含义和使用位置，使用第三个字母：

关键字前缀	含义	示例
CP*	描述整个耦合的属性。	CPON ¹⁾
CPF*	描述跟随轴 (Following axis) 的属性。	CPFPOS ¹⁾
CPL*	描述与引导轴 (Leading axis) 或耦合规律相关的属性。	CPLON ¹⁾ 、 CPLNUM ¹⁾
CPM*	描述整个耦合在特殊状态下的属性。	CPMRESET ¹⁾
¹⁾ 关键字含义参见下表“所有关键字和耦合属性一览”。		

所有关键字和耦合属性一览

下表中列出了所有通用耦合的关键字和可编程的耦合属性一览：

关键字	耦合属性/含义	默认设置 (CPSETTYPE="CP")
CPDEF	创建耦合模块	
CPDEL	删除耦合模块	
CPLDEF	定义引导轴并创建耦合模块	
CPLDEL	删除耦合模块中的引导轴	
CPON	激活耦合模块	
CPOF	关闭耦合模块	
CPLON	激活耦合模块中的引导轴	
CPLOF	关闭耦合模块中的引导轴	
CPLNUM	耦合系数分子	1.0
CPLDEN	耦合系数分母	1.0
CPLCTID	曲线表的编号	未设置
CPLSETVAL	耦合基准	CMDPOS
CPFRS	参考坐标系	BCS
CPBC	程序段转换标准	NOC

15.5 通用耦合

关键字	耦合属性/含义	默认设置 (CPSETTYPE="CP")
CPFPOS + CPON	激活时跟随轴的同步位置	未设置
CPLPOS + CPON	激活时引导轴的同步位置	未设置
CPFMSON	同步模式	CFAST
CPFMON	激活时跟随轴的特性	STOP
CPFMOF	完全关闭时跟随轴的特性	STOP
CPFPOS + CPOF	关闭时跟随轴的关闭位置	未设置
CPMRESET	复位时的耦合特性	NONE
CPMSTART	零件程序启动时的耦合特性	NONE
CPMPRT	通过程序测试进行程序段搜索的情况下零件程序启动时的耦合特性	NONE
CPLINTR	引导轴输入值的偏移值	0.0
CPLINSC	引导轴输入值的缩放系数	1.0
CPLOUTTR	耦合输出值的偏移值	0.0
CPLOUTSC	耦合输出值的缩放系数	1.0
CPSYNCOF	“粗”位置同步运行的阈值	MD37200
CPSYNFIP	“精”位置同步运行的阈值	MD37210
CPSYNCOF2	“粗”位置同步运行的第二个阈值	MD37202
CPSYNFIP2	“精”位置同步运行的第二个阈值	MD37212
CPSYNCOV	“粗”速度同步运行的阈值	MD37220
CPSYNFIV	“精”速度同步运行的阈值	MD37230
CPMBRAKE	特定停止信号和指令下跟随轴的特性	1
CPMVDI	对特定 NC/PLC 接口信号的跟随轴特性	0 (用于位 0 ... 3, 5) 1 (用于位 4, 6)
CPMALARM	抑制耦合专用报警输出	MD11410 MD11415

关键字	耦合属性/含义	默认设置 (CPSETTYPE="CP")
CPSETTYPE	耦合类型	CP

说明

未明确编程（在零件程序或同步动作中）的耦合属性将使用默认设置（参见表中的右列）。根据关键字 CPSETTYPE 的设置，也可以取代默认设置 (CPSETTYPE="CP") 而使用预设的耦合属性（参见章节“耦合类型(页 709)”）。

15.5.2.3 系统变量

通过关键字设置的耦合属性的当前状态可以通过相应的系统变量读取和描述。

说明

写入零件程序时，会产生预处理停止。

表示方法

系统变量的名称通常为相应关键字的名称，前置相应的前缀。

前缀的第 1 个字母描述读取时的访问位置：

系统变量前缀	读取时的访问位置	特征
\$PA_CP	在程序段预处理 (Preparation) 中读取通道相关的轴专用耦合属性	不可用于同步动作。 不生成隐性预处理停止。
\$AA_CP	读取耦合模块的当前状态（跨通道）。	可用于零件程序和同步动作。 在零件程序中使用生成隐性预处理停止。

说明

系统变量 \$PA_CP... 的预处理值仅在零件程序执行激活时与相应系统变量 \$AA_CP... 的值不同

在程序结束或终止时，预处理与主运行状态同步。

与通用耦合相关的所有系统变量列表

标识符	含义
\$AA_ACCLIMA	用 ACCLIMA 设置的加速度修调 (HL)
\$AA_COUP_ACT	跟随轴/副主轴的耦合类型
\$AA_COUP_CORR	副主轴 - 跟踪同步运行偏差补偿值
\$AA_COUP_CORR_DIST	\$AA_COUP_CORR 的剩余行程
\$AA_COUP_OFFS	设定值侧位置偏移
\$AA_CPACTFA	当前有效跟随轴的轴名称
\$AA_CPACTFA	当前有效引导轴的轴名称
\$AA_CPBC	程序段转换标准
\$AA_CPDEFLA	定义引导轴的轴名称
\$AA_CPFACT	跟随轴/副主轴的同步方式
\$AA_CPFCDPT	所有耦合分量上的轴设定位置
\$AA_CPFCDVT	所有耦合分量上的轴设定速度
\$AA_CPFMOF	关闭时跟随轴的特性
\$AA_CPFMON	激活时跟随轴的特性
\$AA_CPFMSON	同步模式
\$AA_CPFERS	耦合的参照系
\$AA_CPLCMDP	引导轴的轴设定位置分量
\$AA_CPLCMDV	引导轴的轴设定速度分量
\$AA_CPLCTID	当前有效曲线表的表编号
\$AA_CPLDEN	耦合系数分母
\$AA_CPLNUM	耦合系数分子
\$AA_CPLSETVAL	引导轴的耦合基准
\$AA_CPLSTATE	耦合状态
\$AA_CPSYNCOF	“粗”位置同步运行的阈值 (HL)
\$AA_CPSYNCOV	“粗”速度同步运行的阈值 (HL)
\$AA_CPSYNFIP	“精”位置同步运行的阈值 (HL)
\$AA_CPSYNFIV	“精”速度同步运行的阈值 (HL)
\$AA_CPLINSC	引导轴输入值的缩放系数 (HL)
\$AA_CPLINTR	引导轴输入值的偏移值 (HL)
\$AA_CPLOUTSC	耦合输出值的缩放系数 (HL)

标识符	含义
\$AA_CPLOUTTR	耦合输出值的偏移值 (HL)
\$AA_CPLTYPE	同步方式
\$AA_CPMRESET	复位后的耦合状态
\$AA_CPMSTART	程序启动后的耦合状态
\$AA_CPNACTFA	当前有效的跟随轴数量
\$AA_CPNACTLA	当前有效的引导轴数量
\$AA_CPNDEFLA	定义的引导轴数量
\$AA_CPSETTYPE	预设置的耦合类型
\$AA_EG_ACTIVE	引导轴 b 的耦合生效
\$AA_EG_AX	第 n 个引导轴的名称
\$AA_EG_BC	程序段转换标准
\$AA_EG_DENOM	耦合系数分母
\$AA_EG_NUMERA	耦合系数分子
\$AA_EG_NUM_LA	用 EGDEF 定义的引导轴数量
\$AA_EG_SYN	引导轴的同步位置
\$AA_EG_SYNFA	跟随轴的同步位置
\$AA_EG_TYPE	同步方式
\$AA_IN_SYNC[FA]	跟随轴的同步状态
\$AA_JERKLIMA	用 JERKLIMA 设置的急动度修调 (HL)
\$AA_LEAD_SP	模拟的引导值 - LEAD 位置
\$AA_LEAD_SV	模拟的引导值 - LEAD 速度
\$AA_LEAD_P_TURN	当前引导值 - 由于模数减少而损失的位置分量
\$AA_LEAD_P	当前引导值 - 位置 (模数减少)
\$AA_LEAD_V	当前引导值 - 速度
\$AA_SYNC	跟随轴的耦合状态
\$AA_SYNCDIFF[FA]	设定值侧同步运行偏差
\$AA_SYNCDIFF_STAT[FA]	设定值侧同步运行偏差状态
\$AA_TYP/TYPE	轴类型
\$AA_VELOLIMA	用 VELOLIMA 设置的速度修调 (HL)
\$PA_ACCLIMA	用 ACCLIMA 设置的加速度修调 (VL)
\$PA_CPFAC	跟随轴/副主轴的同步方式

15.5 通用耦合

标识符	含义
\$PA_CPFOSSTAT	同步位置和停止位置的有效性
\$PA_CPSYNCOV	“粗”位置同步运行的阈值 (VL)
\$PA_CPSYNCOV	“粗”速度同步运行的阈值 (VL)
\$PA_CPSYNFIP	“精”位置同步运行的阈值 (VL)
\$PA_CPSYNFIV	“精”速度同步运行的阈值 (VL)
\$PA_CPLINSC	引导轴输入值的缩放系数 (VL)
\$PA_CPLINTR	引导轴输入值的偏移值 (VL)
\$PA_CPLOUTSC	耦合输出值的缩放系数 (VL)
\$PA_CPLOUTTR	耦合输出值的偏移值 (VL)
\$PA_CPSETTYPE	预设置的耦合类型
\$PA_JERKLIMA	用 JERKLIMA 设置的急动度修调 (VL)
\$PA_VELOLIMA	用 VELOLIMA 设置的速度修调 (VL)
\$VA_COUP_OFFS[S2]	实际值侧同步主轴位置偏移
\$VA_EG_SYNCDIFF	同步运行偏差
\$VA_EG_SYNCDIFF_S	带正负号的同步运行偏差
\$VA_SYNCDIFF[FA]	实际值侧同步运行偏差
\$VA_SYNCDIFF_STAT[FA]	同步运行偏差状态
\$P_COUP_OFFS[S2]	程序中编写的同步主轴位置偏移
\$P_EG_BC	程序段转换标准

更多信息

关于系统变量的详细说明请参见 → 系统变量参数手册。

15.5.3 创建/删除耦合模块**15.5.3.1 创建耦合模块 (CPDEF)**

通过定义跟随轴来创建轴的耦合模块。

编程

语法：CPDEF= (<跟随轴/副主轴>)

名称：Coupling Definition

功能：定义一个耦合模块。耦合不会激活。

跟随轴/
副主轴：类型： AXIS

值域：在通道中定义的所有轴和主轴名称

示例：

编程	注释
CPDEF= (X2)	; 以轴 X2 作为跟随轴创建一个耦合模块。

边界条件

- 耦合模块的最大数量受限（参见章节“前提条件”）。
- 可以对已创建的耦合模块应用 CPDEF，不会用报警确认。

15.5.3.2 删除耦合模块 (CPDEL)

使用 CPDEF 创建的耦合模块可以再次使用 CPDEL 删除。

编程

语法：CPDEL= (<跟随轴/副主轴>)

名称：Coupling Delete

功能：删除一个耦合模块。所有引导轴模块随耦合模块一起删除，预留的内存被再次释放。

15.5 通用耦合

跟随轴/
副主轴：

类型： **AXIS**

值域：

在通道中定义的所有轴和主轴名称

示例：

编程	注释
CPDEL=(X2)	; 删除包含跟随轴 X2 的耦合模块。

边界条件

- 当耦合生效时，开关指令 CPDEL 触发预处理停止。
例外：在 CPSETTYPE="COUP" 时不会发生预处理停止。
- 将 CPDEL 应用于程序段预处理中一个生效的耦合模块会导致隐性禁用该耦合。
- 将 CPDEL 应用于未定义的耦合模块无效。

15.5.3.3 定义引导轴（CPLDEF 或 CPDEF+CPLA）

定义的耦合模块的引导轴/主轴可以使用关键字 CPLDEF 或关键字 CPLA 与 CPDEF 编程/创建。

用 CPLDEF 编程

语法：

CPLDEF [FAx] = (<引导轴/主轴>)

名称：

Coupling Lead Axis Definition

功能：

为跟随轴/副主轴 FAx 定义引导轴/主轴。在耦合模块中创建一个引导轴/主轴模块。如果尚未创建跟随轴/副主轴 FAx 的耦合模块，则隐性创建耦合模块。

引导轴/
主轴：

类型： **AXIS**

值域：在通道中定义的所有轴和主轴名称

示例：

编程	注释
CPLDEF[X2]=(X1)	; 为跟随轴 X2 定义引导轴 X1。

用 CPLA 和 CPDEF 编程

语法：CPLA[FAx] = (<引导轴/主轴>)

名称：Coupling Lead Axis

功能：为跟随轴/副主轴 FAx 定义引导轴/主轴。

引导轴/主轴：类型： AXIS

值域：在通道中定义的所有轴和主轴名称

示例：

编程	注释
CPDEF=(X2) CPLA[X2]=(X1)	; 为跟随轴 X2 定义引导轴 X1。

边界条件

- CPLDEF 只允许在没有 CPDEF/CPON/CPOF/CPDEL 的程序段中使用。
(此限制仅适用于关键字涉及相同耦合模块的情况。)
- 每个耦合模块的引导轴模块最大数量受限（参见章节“前提条件”）。
- 可以为已定义或生效的耦合模块定义引导轴。新定义的引导轴及其属性（例如耦合系数）不会立即生效。激活时需要相应的激活指令（CPON 或 CPLON）。

15.5.3.4 删除引导轴（CPLDEL 或 CPDEL+CPLA）

定义的引导轴可以使用 CPLDEL 或使用 CPLA 与 CPDEL 再次删除，即从耦合模块中删除。

用 CPLDEL 编程

语法：	CPLDEL [FAx] = (<引导轴/主主轴>)
名称：	Coupling Lead Axis Delete
功能：	删除耦合到跟随轴/副主轴 FAx 的引导轴/主主轴。引导轴/主主轴模块将删除，相应的内存被释放。如果耦合模块不再有引导轴/主主轴，耦合模块也会删除，内存被释放。
引导轴/ 主主轴：	类型： AXIS
	值域： 在通道中定义的所有轴和主轴名称
示例：	

编程	注释
CPLDEL[X2] = (X1)	； 删除耦合到跟随轴 X2 的引导轴 X1。

用 CPLA 和 CPDEL 编程

语法：	CPLA [FAx] = (<引导轴/主主轴>)
名称：	Coupling Lead Axis
功能：	删除一个引导轴/主主轴。引导轴/主主轴模块将删除，相应的内存被释放。如果耦合模块不再有引导轴/主主轴，耦合模块也会删除，内存被释放。
引导轴/主主轴：	类型： AXIS

值域:

在通道中定义的所有轴和主轴名称

示例:

编程	注释
CPDEL= (X2) CPLA[X2]= (X1)	; 删除耦合到跟随轴 X2 的引导轴 X1。

边界条件

- CPLDEL 只允许在没有 CPDEF/CPON/CPOF/CPDEL 的程序段中使用。
(此限制仅适用于关键字涉及相同耦合模块的情况。)
- 如果要删除一个生效的引导轴，则与该引导轴的耦合将隐性禁用。
- 删除最后一个引导轴时，会删除整个耦合模块。

15.5.4

激活/关闭耦合

15.5.4.1

激活耦合模块 (CPON)

使用开关指令 CPON 激活定义的耦合模块。

与激活指令一起，耦合属性，如耦合基准和耦合规律，可以通过自己的关键字进行编程（参见章节“编程耦合属性(页 671)”）。

无需编程，联动组或同步主轴对基于设定值耦合（CPLSETVAL 的默认设置）以耦合规律 1:1（CPLNUM/CPLDEN 的默认设置）生效。

编程

语法:

CPON= (<跟随轴/副主轴>)

名称:

Coupling On

功能:

激活跟随轴与所有定义的引导轴的耦合。

15.5 通用耦合

跟随轴/
副主轴：

类型： **AXIS**

值域：

在通道中定义的所有轴和主轴名称

示例：

编程	注释
CPON= (X2)	； 跟随轴 X2 的耦合将激活。

边界条件

将 CPON 应用于已激活的耦合会导致重新同步。因此更改后的耦合属性必要时也会生效。可能丢失的同步（例如跟随轴正在跟随）将会被恢复。

15.5.4.2 关闭耦合模块 (CPOF)

激活的耦合可以通过开关指令 CPOF 来禁用。根据设置的关闭属性来禁用，即取消与引导轴的耦合（参见 CPFMOF）。

编程

语法：

CPOF= (<跟随轴/副主轴>)

名称：

Coupling Off

功能：

禁用跟随轴与所有定义的引导轴的耦合。

跟随轴/
副主轴：

类型： **AXIS**

值域：

在通道中定义的所有轴和主轴名称

示例:

编程	注释
CPOF=(X2)	; 跟随轴 X2 的耦合将禁用。

边界条件

- 当耦合生效时，开关指令 CPOF 触发预处理停止。
例外：在 CPSETTYPE="COUP" 时不会发生预处理停止：
- CPOF 开关指令对已禁用或已删除的耦合模块无效。
- CPOF 也可以在同步动作中编程。

15.5.4.3 激活耦合模块中的引导轴 (CPLON)

CPLON 激活一个引导轴与一个跟随轴的耦合。如果为一个耦合模块定义了多个引导轴，则使用 CPLON 单独激活引导轴。

编程

语法: CPLON [FAx] = <引导轴/主轴>

名称: **Coupling Lead Axis On**

功能: 激活一个引导轴/主轴与跟随轴/副主轴 FAx 的耦合。

引导轴/
主轴: 类型: AXIS

值域: 在通道中定义的所有轴和主轴名称

示例:

编程	注释
CPLON[X2]=(X1)	; 引导轴 X1 与跟随轴 X2 的耦合将激活。

边界条件

CPLON 也可以在同步动作中编程。

15.5.4.4 关闭耦合模块中的引导轴 (CPLOF)

CPLOF 禁用一个引导轴与一个跟随轴的耦合。如果为一个耦合模块定义了多个引导轴，则使用 CPLOF 单独禁用引导轴。

编程

语法:	CPLOF [FAx] = <引导轴/主主轴>
名称:	Coupling Lead Axis Off
功能:	禁用一个引导轴/主主轴与跟随轴/副主轴 FAx 的耦合。
引导轴/ 主主轴:	类型: AXIS
	值域: 通道的轴

示例:

编程	注释
CPLOF [X2] = (X1)	; 引导轴 X1 与跟随轴 X2 的耦合将禁用。

边界条件

CPLOF 也可以在同步动作中编程。

15.5.4.5 隐性创建和删除耦合模块

激活指令也可用于创建耦合模块（无需事先用 CPDEF 定义）。

示例

编程	注释
CPON= (X2) CPLA[X2]= (X1)	; 为跟随轴 X2 与引导轴 X1 创建一个耦合模块并激活耦合模块。
或	
CPLON[X2]= (X1)	
...	
CPOF= (X2)	; 隐性创建的耦合模块在禁用后会再次删除。

边界条件

- 隐性（通过激活指令）创建的耦合模块也会在完全禁用 (CPOF) 的同时删除。
优点：无需使用 CPDEL/CPLDEL 删除。
缺点（可能）：使用 CPOF 会丢失所有设置的耦合属性。
- 隐性创建的耦合模块可以通过后续的 CPDEF/CPLDEF 指令转换为显性耦合模块。这样，使用 CPOF 时不会删除耦合模块，数据得到保留。

15.5.5 编程耦合属性

15.5.5.1 耦合规律（CPLNUM、CPLDEN、CPLCTID）

引导值和跟随值之间的函数关系由每个引导轴/主主轴的耦合规律定义。这种函数关系可以通过一个耦合系数线性定义，也可以通过一个曲线表非线性定义。以此从各个引导值计算的跟随值分量是相加的。

编程：耦合系数

对耦合系数进行编程会禁用先前激活的非线性耦合比（曲线表）。

耦合系数由分子和分母编程。

在标准状态下，即在新建一个耦合模块后没有显性编程时，分子和分母都预设为 1。

如果只编程了分子，则它就是系数，因为分母是 1。

更精确的线性比例可以通过编程分子和分母来定义。

耦合系数分子

语法: CPLNUM [FAx, LAx] = <值>

名称: Coupling Lead Numerator

功能: 定义跟随轴/副主轴 FAx 与引导轴/主轴 LAx 的耦合规律的耦合系数分子。

值: 类型: REAL

值域: -2^{31} 至 $+2^{31}$

缺省值: +1.0

示例:

编程	注释
CPLNUM[X2,X1]=1.3	; 跟随轴 X2 与引导轴 X1 耦合的耦合系数分子应为 1.3。

耦合系数分母

语法: CPLDEN [FAx, LAx] = <值>

名称: Coupling Lead Denominator

功能: 定义跟随轴/副主轴 FAx 与引导轴/主轴 LAx 的耦合规律的耦合系数分母。

值: 类型: REAL

值域: -2^{31} 至 $+2^{31}$

缺省值: +1.0

示例:

编程	注释
CPLDEN[X2,X1]=2	; 跟随轴 X2 与引导轴 X1 耦合的耦合系数分母应为 2。

编程：曲线表

对表编号进行编程会禁用先前激活的线性耦合比（耦合系数）。

对于引导轴/主轴的引导值，引导轴/主轴专用的耦合分量通过指定的曲线表进行计算。

语法: CPLCTID[FAx, LAx] = <值>

名称: Coupling Lead Curve Table Id

功能: 指定用于计算引导轴/主轴 LAx 如何影响跟随轴/副主轴 FAx 的曲线表编号。

值: 类型: INT

值域: -2^{15} 至 $+2^{15}$

示例:

编程	注释
CPLCTID[X2,X1]=5	; 跟随轴 X2 与引导轴 X1 耦合的引导轴专用耦合分量通过 5 号曲线表计算。

边界条件

- 耦合系数为零 (CPLNUM=0) 是一个允许的值。在这种情况下，引导轴/主轴不为跟随轴/副主轴提供任何路径分量，但仍然是耦合的一部分。与关闭状态不同，引导轴/主轴仍然对跟随轴/副主轴有影响。例如，这会影响对故障、限位开关和 NC/PLC 接口信号的反应。
- CPLDEN=0 不是有效值，会被拒绝并发出报警。
- CPLNUM、CPLDEN 和 CPLCTID 也可以在同步动作中编程。
- 非线性耦合比 (CPLCTID) 的可用性取决于选项（参见章节“前提条件”）。

15.5.5.2 耦合基准 (CPLSETVAL)

跟随值可以选择性地由以下内容导出：

- 引导轴的设定位置
- 引导轴的设定速度/转速
- 引导轴的实际位置

据此可以编程以下耦合：

- 设定值耦合
- 速度耦合
- 实际值耦合

编程

语法：CPLSETVAL [FAx, LAx] = "<耦合基准>"

名称：Coupling Lead Set Value

功能：确定引导轴/主轴 LAx 的分接点和跟随轴/副主轴 FAx 上的作用点。

耦合基准：类型： STRING

值域：

"CMDPOS"	Commanded Position	设定值耦合
"CMDVEL"	Commanded Velocity	速度耦合
"ACTPOS"	Actual Value	实际值耦合

缺省值："CMDPOS"

示例：

编程	注释
CPLSETVAL [X2, X1] = "CMDPOS"	; 跟随轴 X2 与引导轴 X1 的耦合由设定值导出。

边界条件

对于一个耦合模块，速度耦合不能与另一个引导轴的设定值或实际值耦合同时激活。

15.5.5.3 参考坐标 (CPFRS)

跟随轴/副主轴的参考坐标规定了由耦合产生的耦合分量在哪个参考坐标系中生效：在基准坐标系还是机床坐标系。

还规定了引导轴/主轴的引导值必须具有哪个参考坐标。在转换激活时，如果指定了机床坐标系作为参考坐标 (CPFRS="MCS")，则使用转换输出端的值作为引导值。

编程

语法：	CPFRS [FAx] = "<参考坐标>"		
名称：	Coupling Following Relation System		
功能：	指定跟随轴/副主轴 FAx 的耦合模块的参考坐标系。		
参考坐标：	类型：	STRING	
	值域：		
	"BCS"	Basis Coordinate System	基准坐标系
	"MCS"	Machine Coordinate System	机床坐标系
	缺省值：	"BCS"	

示例：

编程	注释
CPFRS [X2] = "BCS"	; 包含跟随轴 X2 的耦合模块的参考坐标是基准坐标系。

边界条件

- 参考坐标必须在创建耦合模块时指定，否则使用缺省值。无法后续更改。
- 不支持通过复位同时进行主动转换和耦合。
解决方案：在复位时使用 CPMRESET="OF" 关闭耦合，并在零件程序启动时使用 CPMSTART="ON" 再次激活。
- 不支持同时使用之前的“机床坐标系中轴的耦合（MCS 耦合）”功能和通用耦合。
- CPFRRS 在主运行中不可用。

15.5.5.4 程序段转换特性 (CPBC)

利用程序段转换标准，可以指定应在哪些条件下允许零件程序执行过程中耦合激活时的程序段转换。因此，耦合状态会影响程序段转换特性。如果不满足指定条件，会阻止程序段转换。只有耦合激活时才分析程序段转换标准。

可以使用关键字 CPBC 或编程指令 WAITC 定义程序段转换标准。最后编程的指令有效。

用 CPBC 编程

语法： CPBC [FAx] = "<程序段转换标准>"

名称： Coupling Block Change Criterium

功能： 指定耦合激活时的程序段转换标准。

程序段转换标准： 类型： STRING

值域：	
"NOC"	程序段转换与耦合状态无关。
"IPOSTOP"	当设定值同步运行时进行程序段转换。
"COARSE"	当实际值侧“粗”同步运行时进行程序段转换。
"FINE"	当实际值侧“精”同步运行时进行程序段转换。
缺省值：	"NOC"

示例：

编程	注释
CPBC[X2]="IPOSTOP"	； 零件程序执行过程中的程序段转换在设定值同步运行时进行（与跟随轴 X2 的耦合激活时）。

用 WAITC 编程

语法：	WAITC (FAx1,BC)		
名称：	Wait for Coupling Condition		
功能：	指定耦合激活时的程序段转换标准。		
参数：	Fax：	指定跟随轴，从而指定耦合模块。	
	BC：	确定所需的程序段转换标准。	
FAx：	类型：	STRING	
	值域：	通道的轴	
BC：	类型：	STRING	
	值域：		
	"NOC"	程序段转换与耦合状态无关。	
	"IPOSTOP"	当设定值同步运行时进行程序段转换。	
	"COARSE"	当实际值侧“粗”同步运行时进行程序段转换。	
	"FINE"	当实际值侧“精”同步运行时进行程序段转换。	
	缺省值：	"NOC"	

15.5 通用耦合

示例:

编程	注释
WAITC (X2,"IPOSTOP")	; 零件程序执行过程中的程序段转换在设定值同步运行时进行（与跟随轴 X2 的耦合激活时）。

边界条件

WAITC 与关键字 CPBC 不同，只能在程序段中单独存在。

15.5.5.5 激活时跟随轴的同步位置 (CPFPOS+CPON)

激活耦合 (CPON) 时，可以为跟随轴编程逼近定义的同步位置。
激活时的同步位置立即生效。根据设置的同步模式 (CPFMSON) 逼近由同步位置和耦合规律得出的整体位置，同时考虑到动态响应限制。

编程

语法:	CPON=FAx CPFPOS [FAx] = <值>
名称:	Coupling Following Position
功能:	指定激活时跟随轴的同步位置。可用 AC、IC 和 GP 指定位置。
值:	类型: REAL
	值域: 运行范围限制内的所有位置

示例:

编程	注释
CPON=X2 CPFPOS[X2]=100	; 与跟随轴 X2 的耦合将激活。用 100 作为跟随轴 X2 的同步位置。

边界条件

- CPFPOS 仅在激活指令 CPON/CPLON 下作为同步位置生效。
在关闭指令 CPOF 下，CPFPOS 会被评估为关闭位置（参见章节“关闭时跟随轴的位置 (页 683)”）。
- 在没有激活指令的情况下使用 CPFPOS 会发出报警。
- 如果未设置激活时跟随轴的同步位置，则跟随轴的当前位置作为同步位置生效。
利用程序指令 IC，仍然可以移动当前位置。
- 无论可编程的尺寸说明 (G70/G71) 如何，位置指定都采用设置的基本单位制。

零件程序段（示例）

编程	注释
CPON=(X2) CPFPOS[X2]=100	; 与跟随轴 X2 的耦合将激活。用 100 作为跟随轴的同步位置。
...	
G00 X2=123	; 跟随轴 X2 运行到位置 123。
CPON=(X2)	; 用当前位置 (=123) 作为跟随轴的同步位置。因此，先前生效的同步位置 100 变为无效。

15.5.5.6 激活时引导轴的同步位置 (CPLPOS)

作为引导值使用的当前引导轴位置可以移动。因此，引导轴的同步位置定义了输入变量的零点。

编程

语法:	CPLPOS [FAx, LAx] = <值>
名称:	Coupling Lead Position
功能:	指定激活时引导轴的同步位置。仅可用 AC 指定位置。
值:	类型: REAL
	值域: 运行范围限制内的所有位置

15.5 通用耦合

示例:

编程	注释
CPLPOS[X2,X1]=200	; 用 200 作为引导轴 X1 在与跟随轴 X2 的耦合中的同步位置。

边界条件

- CPLPOS 只能在激活指令 CPON / CPLON 下设置。在没有激活指令的情况下使用 CPLPOS 会发出报警。
- 如果未设置激活时 (CPON) 引导轴的同步位置，则引导轴的当前位置作为同步位置生效，从而作为输入变量的零点生效。
- 无论可编程的尺寸说明 (G70 / G71) 如何，位置指定都采用设置的基本单位制。

零件程序段（示例）

编程	注释
CPON=(X2) CPFPOS[X2]=100 CPLPOS[X2,X1]=200	; 与跟随轴 X2 的耦合将激活。用 100 作为跟随轴的同步位置， 用 200 作为引导轴 X1 的同步位置。
...	
N20 X1=280 F1000	; 引导轴 X1 运行到位置 280。
CPON=(X2)	; 用当前位置 X1=280 作为引导轴的同步位置。因此，先生效的 引导轴同步位置 (200) 变为无效。

15.5.5.7 同步模式 (CPFMSON)

同步模式确定激活耦合时的同步特性。

编程

语法:	CPFMSON [FAx] = "<同步模式>"
名称:	Coupling Following Mode Strategy On
功能:	指定耦合时的同步模式。
同步模式:	类型: STRING

值域:

"CFAST"	Closed Coupling Fast	在最短的时间内完成耦合。
"CCOARSE"	Closed If Gab Coarse	当耦合规律所要求的跟随轴位置在当前跟随轴位置的范围内时，耦合才会被激活。
"NTGT"	Next Tooth Gap Time Optimized	在最短的时间内逼近下一个齿隙。
"NTGP"	Next Tooth Gap Path Optimized	以最短的行程逼近下一个齿隙。
"NRGT"	Next Ratio Gap Time Optimized	根据螺纹数与齿数之比，在最短的时间内逼近下个段。
"NRGP"	Next Ratio Gap Path Optimized	根据螺纹数与齿数之比，以最短的行程逼近下个段。
"ACN"	Absolute Coordinate Negative	仅针对回转轴！ 回转轴以 负 轴向逼近同步位置。同步立即执行。
"ACP"	Absolute Coordinate Positive	仅针对回转轴！ 回转轴以 正 轴向逼近同步位置。同步立即执行。
"DCT"	Direct Coordinate Time Optimized	仅针对回转轴！ 回转轴在 最短的时间内 逼近所编程的同步位置。同步立即执行。
"DCP"	Direct Coordinate Path Optimized	仅针对回转轴！ 回转轴以 最短的行程 逼近所编程的同步位置。同步立即执行。

缺省值: "CFAST"

示例:

编程	注释
CPFMSON[X2]="CFAST"	; 用 CFAST 作为与跟随轴 X2 耦合的同步模式。

15.5.5.8 激活时跟随轴的特性 (CPFMON)

激活耦合时跟随轴/副主轴的特性可以用关键字 CPMON 编程。

编程

语法: CPMON [FAx] = "<激活特性>"

名称: Coupling Following Mode On

功能: 指定激活耦合时跟随轴/副主轴的特性。

激活特性: 类型: STRING

值域:		
"STOP"	Stop	仅针对主轴！ 激活前副主轴的当前运动将停止。
"CONT"	Continue	仅针对主轴和主运行轴！ 跟随轴/副主轴的当前运动将在耦合时被作为起始运动。
"ADD"	Additional	仅针对主轴！ 耦合运动分量会额外叠加在当前的运动上，即跟随轴/副主轴的当前运动会保留为叠加运动。

缺省值: "STOP"

示例:

编程	注释
CPMON[X2]="CONT"	; 跟随轴 X2 的当前运动将被作为起始运动。

15.5.5.9 关闭时跟随轴的特性 (CPFMOF)

完全关闭激活的耦合时，跟随轴/副主轴的其他特性可以用关键字 CPMOF 编程。

编程

语法:	CPFMOF [FAx] = "<关闭特性>"		
名称:	Coupling Following Mode Off		
功能:	指定完全关闭耦合时跟随轴/副主轴的特性。		
关闭特性:	类型:	STRING	
	值域:		
	"STOP"	Stop	跟随轴/副主轴停止。 激活的叠加运动将制动到停止。之后将断开耦合（禁用）。
	"CONT"	Continue	仅针对主轴和主运行轴！ 副主轴以关闭时刻有效的转速/速度继续运行。
	缺省值:	"STOP"	

示例:

编程	注释
CPFMOF [S2] = "CONT"	； 副主轴 S2 以关闭时刻有效的转速/速度继续运行。

15.5.5.10 关闭时跟随轴的位置 (CPFPOS+CPOF)

关闭耦合 (CPOF) 时可以要求跟随轴逼近特定位置。

编程

语法:	CPOF= (FAx) CPFPOS [FAx] = <值>
功能:	指定跟随轴 FAx 的关闭位置。

15.5 通用耦合

值:

类型: REAL

值域:

运行范围限制内的所有位置

示例:

编程	注释
CPOF= (X2) CPFPOS[X2]=100	; 与跟随轴 X2 的耦合将禁用。逼近设置的跟随轴关闭位置 (100)。

边界条件

- CPFPOS 仅在关闭指令 CPOF 下作为关闭位置生效。
在开关指令 CPON 下，CPFPOS 会被评估为激活位置（参见章节“激活时跟随轴的同步位置 (页 678)”）。
- 只有在关闭模式下才允许设置关闭位置：
CPFMOF=STOP
- 以最大动态响应逼近关闭位置。
- 程序段转换特性取决于关键字 CPBC 的参数设置。

15.5.5.11 复位时的状态 (CPMRESET)

在复位时可以激活、禁用耦合或保留当前状态。可以为每个耦合模块单独设置特性。

编程

语法:

CPMRESET [FAx] = "<复位特性>"

名称:

Coupling Mode RESET

功能:

指定复位时的耦合特性。

复位特性:

类型: STRING

值域:

"NONE" 耦合的当前状态保留。

"ON"	如果创建了相应的耦合模块，则会激活耦合。定义的全部引导轴关系都会激活。当全部这些或部分引导轴关系都有效时仍会如此，也就是说即使是完全激活的耦合也会重新进行同步。
"OF"	激活的叠加运动将制动到停止。之后将关闭耦合。如果创建相应的耦合模块时未进行明确的定义 (CPDEF)，则该耦合模块将被删除。否则将会保留，即可继续进行使用。
"OFC"	仅针对主轴！ 副主轴以关闭时刻有效的转速/速度继续运行。耦合会被关闭。如果创建相应的耦合模块时未进行明确的定义 (CPDEF)，则该耦合模块将被删除。否则将会保留，即可继续进行使用。
"DEL"	激活的叠加运动将制动到停止。此后耦合将被禁用，然后删除。
"DELC"	仅针对主轴！ 副主轴以关闭时刻有效的转速/速度继续运行。耦合将被禁用，然后删除。
缺省值:	"NONE"

示例:

编程	注释
CPMRESET[X2]="DEL"	; 复位时与跟随轴 X2 的耦合将被禁用，然后删除。

边界条件

- 使用 CPMRESET 设置的耦合属性将一直保留，直到耦合模块删除 (CPDEL)。
- 如果设置了耦合类型 (CPSETTYPE="TRAIL"、"LEAD"、"EG" 或 "COUP")，则复位时的特性由以下机床数据确定：
MD20110 \$MN_RESET_MODE_MASK (确定复位/零件程序结束后控制系统的初始设置)
→ 参见“耦合类型 (CPSETTYPE) (页 709)”中的“默认设置”部分。

15.5.5.12 零件程序开始时的状态 (CPMSTART)

在零件程序开始时可以激活、禁用耦合或保留当前状态。可以为每个耦合模块单独设置特性。

编程

语法:	CPMSTART [FAx] = <值>
名称:	Coupling Mode Start
功能:	指定零件程序开始时的耦合特性。
值:	类型: STRING
	值域:
	"NONE" 耦合的当前状态保留。
	"ON" 如果创建了相应的耦合模块, 则会激活耦合。定义的全部引导轴关系都会激活。当全部这些或部分引导轴关系都有效时仍会如此, 也就是说即使是完全激活的耦合也会重新进行同步。
	"OF" 耦合会被关闭。如果创建相应的耦合模块时未进行明确的定义 (CPDEF), 则该耦合模块将被删除。否则将会保留, 即可继续进行使用。
	"DEL" 耦合将被禁用, 然后删除。
	缺省值: "NONE"

示例:

编程	注释
CPMSTART [X2] = "ON"	; 在零件程序开始时, 与跟随轴 X2 的耦合将激活。

边界条件

- 使用 CPMSTART 设置的耦合属性将一直保留, 直到耦合模块删除 (CPDEL)。
- 如果设置了耦合类型 (CPSETTYPE="TRAIL"、"LEAD"、"EG" 或 "COUP"), 则零件程序开始时的特性由以下机床数据确定:
MD20112 \$MC_START_MODE_MASK (确定 NC 启动时控制系统的初始设置)
→ 参见 “耦合类型 (CPSETTYPE) (页 709)”中的 “默认设置” 部分。

15.5.5.13 通过程序测试进行搜索的情况下零件程序开始时的状态 (CPMPRT)

通过程序测试进行程序段搜索 (SERUPRO) 的情况下，在零件程序开始时可以激活、禁用耦合或保留当前状态。可以为每个耦合模块单独设置特性。

编程

语法:	CPMPRT [FAx] = <值>
名称:	Coupling Mode Program Test
功能:	指定通过程序测试进行程序段搜索的情况下零件程序开始时的耦合特性。
值:	类型: STRING
值域:	
"NONE"	耦合的当前状态保留。
"ON"	如果创建了相应的耦合模块，则会激活耦合。定义的全部引导轴关系都会激活。当全部这些或部分引导轴关系都有效时仍会如此，也就是说即使是完全激活的耦合也会重新进行同步。
"OF"	耦合会被关闭。如果创建相应的耦合模块时未进行明确的定义 (CPDEF)，则该耦合模块将被删除。否则将会保留，即可继续进行使用。
"DEL"	耦合将被禁用，然后删除。
缺省值:	"NONE"

示例:

编程	注释
CPMPRT [X2] = "ON"	; 通过程序测试进行程序段搜索的情况下，在零件程序开始时，与跟随轴 X2 的耦合将激活。

边界条件

- 使用 CPMPT 设置的耦合属性将一直保留，直到耦合模块删除 (CPDEL)。
- 如果设置了 CPMPT="NONE"，则通过程序测试进行程序段搜索 (SERUPRO) 的情况下，该特性由 CPMSTART 确定。
- 如果设置了耦合类型 (CPSETTYPE="TRAIL"、"LEAD"、"EG" 或 "COUP")，则通过程序测试进行程序段搜索的情况下，零件程序开始时的特性由以下机床数据确定：
MD22620 \$MC_START_MODE_MASK_PRT (确定特殊启动时控制系统的初始设置)
MD22621 \$MC_ENABLE_START_MODE_MASK_PRT (激活 MD22620)
MD20112 \$MC_START_MODE_MASK (确定 NC 启动时控制系统的初始设置)
→ 参见“耦合类型 (CPSETTYPE) (页 709)”中的“默认设置”部分。

15.5.5.14 偏移/缩放 (CPLINTR、CPLINSC、CPOUTTR、CPOUTSC)

跟随轴和引导轴之间现有的耦合关系可以缩放和偏移。

这些功能对跟随轴总设定值的作用可以从以下公式中看出：

$$FA_{Total} = FA_{cmd} + \{ transOut_1 + scaleOut_1 * [(LA_1 - SynPosLA_1) * scaleIn_1 + transIn_1] * KF_1 \} + \{ transOut_2 + scaleOut_2 * [(LA_2 - SynPosLA_2) * scaleIn_2 + transIn_2] * KF_2 \}$$

FA _{Total}	跟随轴的总设定值
FA _{Cmd}	在零件程序中设置的设定值
LA _{1/2}	第 1 个或第 2 个引导轴/引导值的设定值或实际值
SynPosLA _{1/2}	第 1 个或第 2 个引导轴/引导值的同步位置
scaleIn _{1/2}	第 1 个或第 2 个引导值的缩放系数
transIn _{1/2}	第 1 个或第 2 个引导值的偏移
KF _{1/2}	第 1 个或第 2 个引导轴/引导值的耦合系数
scaleOut _{1/2}	第 1 个或第 2 个输出值的缩放系数
transOut _{1/2}	第 1 个或第 2 个输出值的偏移

说明

可以为每个引导轴定义缩放值和偏移值。

编程

输入值偏移

语法: `CPLINTR [FAx, LAx] = <值>`

名称: **Coupling Lead In Translation Displacement**

功能: 定义引导轴 LAx 输入值的偏移值。

值: 类型: REAL

缺省值: 0

示例:

编程	注释
<code>CPLINTR[X2,X1]=-50</code>	; 引导轴 X1 的输入值在负轴方向上偏移值 50。

输入值缩放

语法: `CPLINSC [FAx, LAx] = <值>`

名称: **Coupling Lead In Scale Factor**

功能: 定义引导轴 LAx 输入值的缩放系数。

值: 类型: REAL

缺省值: 1

示例:

编程	注释
<code>CPLINSC[X2,X1]=0.5</code>	; 引导轴 X1 的输入值乘以系数 0.5。

15.5 通用耦合

输出值偏移

语法: `CPLOUTTR [FAx, LAx] = <值>`

名称: **Coupling Lead Out Translation Displacement**

功能: 定义跟随轴 FAx 与引导轴 LAx 耦合的输出值的偏移值。

值: 类型: REAL

缺省值: 0

示例:

编程	注释
<code>CPLOUTTR[X2,X1]=100</code>	; 跟随轴 X2 与引导轴 X1 耦合的输出值在正轴方向上偏移值 100。

输出值缩放

语法: `CPLOUTSC [FAx, LAx] = <值>`

名称: **Coupling Lead Out Scale Factor**

功能: 定义跟随轴 FAx 与引导轴 LAx 耦合的输出值的缩放系数。

值: 类型: REAL

缺省值: 1

示例:

编程	注释
<code>CPLOUTSC[X2,X1]=3</code>	; 跟随轴 X2 与引导轴 X1 耦合的输出值乘以系数 3。

说明

现有耦合类型“引导值耦合”中使用的以下设置数据在通用耦合中得到考虑，无论设置的耦合类型 (CPSETTYPE) 如何：

SD43102 \$SA_LEAD_OFFSET_IN_POS[FAx]（引导值的偏移）

SD43104 \$SA_LEAD_SCALE_IN_POS[FAx]（引导值的缩放）

SD43106 \$SA_LEAD_OFFSET_OUT_POS[FAx]（曲线表函数值的偏移）

SD43108 \$SA_LEAD_SCALE_OUT_POS[FAx]（曲线表函数值的缩放）

这些设置数据：

- 作用于通过曲线表耦合到跟随轴的所有引导轴。与多个引导轴耦合时必须考虑这一点！
- 是对 CP 关键字 CPLINTR、CPLINSC、CPLOUTTR 和 CPLOUTSC 的补充。

15.5.5.15 1 级同步运行监控 (CPSYNCOV、CPSYNFIP、CPSYNCOV、CPSYNFIV)

1 级同步运行监控

在每个插补节拍中，在设定值和实际值侧监控耦合组的同步运行。一旦**同步运行偏差**（跟随轴的设定值或实际值与依照耦合规律根据引导轴的设定值或实际值得出的值之间的差值）达到以下可配置和可编程的阈值之一，同步运行监控就会作出反应：

- 设定值/实际值耦合时（参见“耦合基准 (CPLSETVAL) (页 674)”）：
 - “粗”位置同步运行的阈值
 - “精”位置同步运行的阈值
- 速度耦合时（参见“耦合基准 (CPLSETVAL) (页 674)”）：
 - “粗”速度同步运行的阈值
 - “精”速度同步运行的阈值

可以使用以下 CP 系统变量读取当前同步运行偏差：

系统变量	含义
\$AA_SYNCDIFF [FAx]	设定值侧同步运行偏差
\$VA_SYNCDIFF [FAx]	实际值侧同步运行偏差

说明

同步运行偏差是有符号的，因此可以确定跟随轴的超前或滞后。

同步运行时的耦合状态

状态	说明
未同步	只要同步运行偏差大于“粗”位置同步运行或“粗”速度同步运行的阈值，耦合组就被指定为不同步。
达到“粗”同步运行	同步运行偏差达到“粗”位置同步运行或“粗”速度同步运行的阈值。 如果实际值侧出现同步运行偏差，则置位以下 NC/PLC 接口信号： <Axis>.coupling.in.syncFineReached（粗同步）
达到“精”同步运行	同步运行偏差达到“精”位置同步运行或“精”速度同步运行的阈值。 如果实际值侧出现同步运行偏差，则置位以下 NC/PLC 接口信号： <Axis>.coupling.in.syncFineReached（精同步）

同步运行时的耦合状态可通过以下系统变量读取：

系统变量	含义	
\$AA_SYNC [FAx]	耦合状态	
	值	状态
	0	未同步
	1	达到“粗”同步运行
	2	达到“精”同步运行

信号响应

同步运行信号本身不会导致相关轴停止，但可以通过同步动作或 NC/PLC 接口信号触发（参见“扩展的停止和退回 (页 953)”章节）。

配置

设置一级同步运行监控的阈值：

- 对于设定值/实际值耦合，在以下机床数据中：
 - MD37200 \$MA_COUPLE_POS_TOL_COARSE（“粗同步”的阈值）
 - MD37210 \$MA_COUPLE_POS_TOL_FINE（“精同步”的阈值）
- 对于速度耦合，在以下机床数据中：
 - MD37220 \$MA_COUPLE_VELO_TOL_COARSE（“粗”速度公差）
 - MD37230 \$MA_COUPLE_VELO_TOL_FINE（“精”速度公差）

编程

一级同步运行监控的阈值也可以通过 CP 关键字进行编程：

“粗”位置同步运行的阈值

语法：CPSYNCOF [FAx] = <值>

名称：Coupling Synchronous Difference Coarse Position

功能：定义“粗”位置同步运行的阈值。

值：类型： REAL

缺省值符合以下机床数据中的设置：

MD37200 \$MA_COUPLE_POS_TOL_COARSE [FAx]

“精”位置同步运行的阈值

语法：CPSYNFIP [FAx] = <值>

名称：Coupling Synchronous Difference Fine Position

功能：定义“精”位置同步运行的阈值。

值：类型： REAL

缺省值符合以下机床数据中的设置：
MD37210 \$MA_COUPLE_POS_TOL_FINE [FAx]

“粗”速度同步运行的阈值

语法： CPSYNCOV [FAx] = <值>

名称： Coupling Synchronous Difference Coarse Velocity

功能： 定义“粗”速度同步运行的阈值。

值： 类型： REAL

缺省值符合以下机床数据中的设置：
MD37220 \$MA_COUPLE_VELO_TOL_COARSE [FAx]

“精”速度同步运行的阈值

语法： CPSYNFIV [FAx] = <值>

名称： Coupling Synchronous Difference Fine Velocity

功能： 定义“精”速度同步运行的阈值。

值： 类型： REAL

缺省值符合以下机床数据中的设置：
MD37230 \$MA_COUPLE_VELO_TOL_FINE [FAx]

示例

程序代码	注释
CPDEF=(S2) CPLA[S2]=(S1)	； 定义一个主轴耦合： 主主轴 S1 与副主轴 S2
CPON=(S2) CPSYNCOV[S2]=0.5 CPSYNFIV[S2]=0.25	； 激活与副主轴 S2 的耦合。位置同步运行的阈值设置为 0.5 (“粗”) 和 0.25 (“精”)。

程序代码	注释
...	

边界条件

- 在考虑同步运行偏差时，不考虑可能生效的耦合级联。这表示：如果在所考虑的耦合模块中，引导轴本身是另一个耦合模块中的跟随轴，则当前实际位置或设定位置用作计算同步运行偏差的输入变量。因此该同步运行偏差并不表示级联的总同步误差。
- 如果引导轴不是真实轴而是模拟轴，则实际值不能用于实际值侧的同步运行监控。在这种情况下，使用建模的实际值（根据机床数据设置）。

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.coupling.in.syncFineReached	LBP_Axis*.E_SynchronFine	DB31, ...DBX98.0
<Axis>.coupling.in.syncSpindleReached	LBP_Axis*.E_SynchronCoarse	DB31, ...DBX98.1

15.5.5.16 2 级同步运行监控（CPSYN COP2、CPSYN FIP2）

2 级同步运行监控

在 CP 位置耦合（设定值或实际值耦合；参见“耦合基准 (CPLSETVAL) (页 674)”）生效时，可以使用二级同步运行监控，在达到程序段转换标准“COARSE”/“FINE”（参见“程序段转换特性 (CPBC) (页 676)”）后，在实际值侧监控是否遵守与一级无关的同步运行公差的范围之一。

对于二级同步运行监控，必须配置或编程以下两个阈值：

- “粗”位置同步运行的阈值 2
- “精”位置同步运行的阈值 2

技术背景

作为“达到同步运行”和“退出同步窗口”监控的一部分，它对于正确评估在多边形车削（表面、废料等）时可能出现的问题非常有用 - 有时也适用于摆动的同步主轴 - 可以独立于一级同步运行监控的粗/精同步运行公差定义一个公差范围（类似于轴的准停和停止监控），以便能够接收可配置的错误消息或警告。

配置

二级同步运行监控的阈值在以下机床数据中设置：

MD37202 \$MA_COUPLE_POS_TOL_COARSE_2 （“粗同步”的第二个阈值）

MD37212 \$MA_COUPLE_POS_TOL_FINE_2 （“精同步”的第二个阈值）

说明

如果相应的阈值 = 0，则相关监控未生效。这也是缺省值，以便保留与旧软件版本的兼容性。

编程

二级同步运行监控的阈值也可以通过 CP 关键字进行编程：

“粗”位置同步运行的阈值 2

语法： CPSYNCO2 [FAx] = <值>

名称： Coupling Synchronous Difference Coarse Position 2

功能： 定义“粗”位置同步运行的第二个阈值。

值： 类型： REAL

缺省值符合以下机床数据中的设置：

MD37202 \$MA_COUPLE_POS_TOL_COARSE_2 [FAx]

“精”位置同步运行的阈值 2

语法： CPSYNFI2 [FAx] = <值>

名称： Coupling Synchronous Difference Fine Position 2

功能： 定义“精”位置同步运行的第二个阈值。

值： 类型： REAL

缺省值符合以下机床数据中的设置：

MD37212 \$MA_COUPLE_POS_TOL_FINE_2 [FAx]

与一般 CP 特性相似，编程在零件程序中只能随着下一个开关指令生效，而对于同步动作则立即生效。

过程

启动

只要满足以下条件，在耦合生效时，二级同步运行监控就会启动：

- 达到设定值侧同步运行：
`<Axis>.coupling.in.ProcessActive`（正在进行同步）= 0
- 达到实际值侧“粗”/“精”同步与一级同步运行监控的公差（参见“1 级同步运行监控（CPSYNCOV、CPSYNFIP、CPSYNCOV、CPSYNFIV）（页 691）”）：
`<Axis>.coupling.in.syncCoarseReached`（粗同步）= 1 / `<Axis>.coupling.in.syncFineReached`（精同步）= 1

监控

只要实际值侧的同步运行偏差不超过“精”位置同步运行阈值 2 和“粗”位置同步运行阈值 2，则设置以下 NC/PLC 接口信号：

`<Axis>.coupling.in.syncFine2Reached`（精同步 2）

`<Axis>.coupling.in.syncCoarse2Reached`（粗同步 2）

如果跟随轴/副主轴由于加工过程中的临时过载（例如多边形敲击时进给量过高）而不再遵循引导轴/主主轴的规定，并且偏差大于设定公差，则超过“精”/“粗”公差分别由可清除的显示报警显示：

报警 22026“通道 %1 程序段 %2 跟随轴/副主轴 %3 同步运行 (2)：超出粗公差”

报警 22025“通道 %1 程序段 %2 跟随轴/副主轴 %3 同步运行 (2)：超出精公差”

加工不会因此中断。

说明

两种报警也可以同时出现。

结束

二级同步运行监控在以下情况下结束：

- 关闭耦合（通过关闭指令：CPOF、CPDEL、CPLOF、CPLDEL）
- 新的同步请求：
 - CPON / CPLON / CPDO / CPLOF
 - <Axis>.coupling.out.syncSpindleResync（同步副主轴）= 1
 - 内部同步请求

如果在这些情况下再次满足启动条件，则重新启动监控。

- 同步动作中耦合规律变化时（CPLNUM、CPLDEN、CPLCTID）
- 由于缺少副主轴的使能（急停、报警响应）而复位设定值侧同步运行

结束监控时，信号 <Axis>.coupling.in.syncFine2Reached/
<Axis>.coupling.in.syncCoarse2Reached 复位。

边界条件**排除条件**

在以下情况下不进行监控：

- MD37202 或 MD37212 = 0。
- 速度耦合生效 (CPLSETVAL="CMDVEL")。
- <Axis>.coupling.out.syncSpindleSyncDisable（禁用副主轴同步）= 1
- 跟随轴/副主轴或生效的引导轴/主轴之一快速停止。
- SERUPRO 或程序段搜索生效。
- 逐通道试运行（如果跟随轴/副主轴或一个生效的引导轴/主轴没有实际运行）。
- <Axis>.basic.in.disabled（进给轴/主轴禁用）= 1 用于跟随轴/副主轴或生效的引导轴/主轴之一。

“跟踪同步运行偏差”生效

只要“跟踪同步运行偏差”功能生效（<Axis>.coupling.out.syncCorrection = 1；参见“跟踪同步运行偏差 (页 724)”），监控就暂停。

使用经典耦合类型时的 2 级同步运行监控

对于联动、引导值耦合、电子齿轮箱和同步主轴等经典耦合类型，“2 级同步运行监控”仅可通过 CP 匹配循环使用（参见“匹配循环 (页 708)”）。

示例

程序代码	注释
G0 Z-300 X50	
CPDEF=(S2) CPLA=(S1) CPLNUM=2 CPLDEN=1	; 传动比根据刀具。
CPBC="FINE"	
M3 S2000	
CPON=(S2) CPLA=(S1) CPSYNCOPI2[X]=1.6	; 达到“精同步”时进行程序段转换。
CPSYNFIP2[X]=0.8	; 同时 2 级同步运行监控激活
	; (公差: 粗 1.6, 精 0.8)。
X20	
G1 Z-200	; 加工程序段, 材料去除。
	同步运行决定质量。
CPOF=(S2)	; 取消耦合, 关闭监控。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.coupling.out.enableFollowOverlay	LBP_Axis*.A_FollowAxOv	DB31, ...DBX26.4
<Axis>.coupling.out.syncSpindleResync	LBP_Axis*.A_Resynchronize	DB31, ...DBX31.4
<Axis>.coupling.out.syncSpindleSyncDisable	LBP_Axis*.A_DisSync	DB31, ...DBX31.5
<Axis>.coupling.out.syncCorrection	LBP_Axis*.A_Tracksync	DB31, ...DBX31.6

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.disabled	LBP_Axis*.E_AxSpDisable	DB31, ...DBX63.3
<Axis>.coupling.in.syncFineReached	LBP_Axis*.E_SyncronFine	DB31, ...DBX98.0
<Axis>.coupling.in.syncCoarseReached	LBP_Axis*.E_SyncronCoarse	DB31, ...DBX98.1
<Axis>.coupling.in.syncSpindleSlaveActive	LBP_Axis*.E_SlaveSp	DB31, ...DBX99.1
<Axis>.coupling.in.ProcessActive	LBP_Axis*.E_Sync	DB31, ...DBX99.4
<Axis>.coupling.in.syncFine2Reached	LBP_Axis*.E_Syncron2Fine	DB31, ...DBX103.4
<Axis>.coupling.in.syncCoarse2Reached	LBP_Axis*.E_Syncron2Coarse	DB31, ...DBX103.5

15.5.5.17 对停止信号和停止指令的响应 (CPMBRAKE)

特定停止信号和指令下跟随轴的特性可以用 CP 关键字 CPMBRAKE 确定。

编程

语法:	CPMBRAKE [FAx] = <值>										
名称:	Coupling Mode Brake										
功能:	CPMBRAKE 是位编码的 CP 关键字，用于确定跟随轴 FAx 在以下事件中的制动特性： <table><tr><th>位</th><th>事件</th></tr><tr><td>0</td><td> - 接口信号 <Axis>.basic.out.feedStopSpindleStop（进给停止/主轴停止）等待处理 或 - 已设置 CP 软件限制停止（参见“CP 软件限制监控 (页 718)”） </td></tr></table> 提示： 位 0 仅对“可自由编程”耦合类型 (CPSETTYPE=CP) 有意义。 <table><tr><th>值</th><th>含义</th></tr><tr><td>0</td><td>制动不会传送到引导轴。</td></tr><tr><td>1</td><td>制动以上下文相关的方式传送到引导轴。</td></tr></table> 1 - 31 预留 对编程有以下规则： - CPMBRAKE 必须在包含 CPDEF 或 CPON 的程序段中编程（⇒ 只能为非生效耦合进行编程）。 - 如果没有明确的 CPMBRAKE 编程，定义耦合时适用以下预设： 对于 CPSETTYPE=CP 的预处理耦合，位 0 = 1（否则位 0 = 0）	位	事件	0	- 接口信号 <Axis>.basic.out.feedStopSpindleStop（进给停止/主轴停止）等待处理 或 - 已设置 CP 软件限制停止（参见“CP 软件限制监控 (页 718)”）	值	含义	0	制动不会传送到引导轴。	1	制动以上下文相关的方式传送到引导轴。
位	事件										
0	- 接口信号 <Axis>.basic.out.feedStopSpindleStop（进给停止/主轴停止）等待处理 或 - 已设置 CP 软件限制停止（参见“CP 软件限制监控 (页 718)”）										
值	含义										
0	制动不会传送到引导轴。										
1	制动以上下文相关的方式传送到引导轴。										

示例

示例 1:

编程	注释
CPDEF=(AX5) CPLA[AX5]=(AX4) CPMBRAKE[AX5]=0 ; ...	定义一个耦合（引导轴 AX4 与跟随轴 AX5）。接口信号“进给停止/主轴停止”或“CP 软件限制停止”不应制动耦合组。

示例 2:

编程	注释
CPDEF= (S2) CPLA[S2]=(S1) ;	定义一个主轴耦合： 主轴 S1 与副主轴 S2
CPON= (S2) CPMBRAKE[S2]=1 ;	激活与副主轴 S2 的耦合。接口信号“进给停止/主轴停止” 或“CP 软件限制停止”应制动耦合组。
...	

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.feedStopSpindleStop	LBP_Axis*.A_FDSpStop	DB31, ...DBX4.3

15.5.5.18 对特定 NC/PLC 接口信号的响应 (CPMVDI)

使用 CP 关键字 CPMVDI 可以确定耦合模块对特定 NC/PLC 接口信号的响应。

编程

语法:	CPMVDI [FAx] = <值>
名称:	Coupling Mode VDI Signal
功能:	CPMVDI 是位编码的 CP 关键字，用于确定跟随轴 FAx 的耦合模块对特定 NC/PLC 接口信号的响应。 位逻辑运算符 B_OR、B_AND、B_NOT 和 B_XOR 可用于设置各个位。

位	含义
0	预留。
1	预留。
2	预留。
3	NC/PLC 接口信号： <Axis>.basic.out.disable（进给轴/主轴禁用） 对跟随轴/副主轴的有效性可通过位 3 设置： 位 3 = 0 <Axis>.basic.out.disable 对跟随轴/副主轴无效。跟随轴/副主轴在进给轴/主轴禁用方面的状态仅由引导轴/主主轴的状态导出。 位 3 = 1 <Axis>.basic.out.disable 对跟随轴/副主轴有效。引导轴/主主轴在进给轴/主轴禁用方面的状态不影响跟随轴/副主轴。 提示： 当设置位 3 = 1 时，跟随轴/副主轴的进给轴/主轴禁用状态仅在程序测试或 SERUPRO 状态未激活时有效（另见位 5）。
4	利用位 4 指定当 NC/PLC 接口信号 <Axis>.basic.out.disable（进给轴/主轴禁用）对跟随轴/副主轴有效时，相关运动分量的使能： 位 4 = 0 无论相应引导轴/主主轴的进给轴/主轴禁用状态如何，引导轴/主主轴的相关运动分量都会生效。 位 4 = 1 仅当引导轴/主主轴的进给轴/主轴禁用状态与跟随轴/副主轴的进给轴/主轴禁用状态一致时，引导轴/主主轴的相关运动分量才会生效。否则会抑制运动分量。 提示： 位 4 仅在位 3 置位时有意义，即当 NC/PLC 接口信号 <Axis>.basic.out.disable（进给轴/主轴禁用）对跟随轴/副主轴有效时（参见位 3 和 5）。
5	NC/PLC 接口信号： <HmiChan>.basic.in.progTestSelected（程序测试已选择）或 <Chan>.basic.out.progTest（激活程序测试） 对跟随轴/副主轴的有效性可通过位 5 设置： 位 5 = 0 <HmiChan>.basic.in.progTestSelected 或 <Chan>.basic.out.progTest 对跟随轴/副主轴无效。跟随轴/副主轴在进给轴/主轴禁用方面的状态仅由引导轴/主主轴的状态导出。

位 5 = 1 如果耦合模块中一个轴的“程序测试”状态生效，则在进给轴/主轴禁用方面的固有状态对跟随轴/副主轴生效。引导轴/主轴在进给轴/主轴禁用方面的状态不影响跟随轴/副主轴。

提示：

位 5 置位时，即使引导轴/主轴处于不同状态，程序测试状态也对跟随轴/副主轴有效。

- 6 利用位 6 指定当 NC/PLC 接口信号 <HmiChan>.basic.in.progTestSelected（程序测试已选择）或 <Chan>.basic.out.progTest（激活程序测试）对跟随轴/副主轴有效时，相关运动分量的使能：
- 位 6 = 0 无论相应引导轴/主轴的进给轴/主轴禁用状态如何，引导轴/主轴的相关运动分量都会生效。
- 位 6 = 1 仅当引导轴/主轴的进给轴/主轴禁用状态与跟随轴/副主轴的进给轴/主轴禁用状态一致时，引导轴/主轴的相关运动分量才会生效。否则会抑制运动分量。

提示：

位 6 仅在位 5 置位时有意义，即当程序测试状态对跟随轴/副主轴有效时（参见位 3 和 5）。

- 7 预留。
- 8 预留。

说明

通过 NC/PLC 接口信号 <Axis>.basic.out.disable 为跟随轴/副主轴设置的进给轴/主轴禁用，可以像对任何其他轴/主轴一样，由程序测试状态（<HmiChan>.basic.in.progTestSelected 或 <Chan>.basic.out.progTest）和 SERUPRO 覆盖。

位 3/5 和 4/6 的作用

根据相关的进给轴/主轴禁用，跟随轴/副主轴的不同运动分量的有效性如下表所示：

A/S 禁用 FA	A/S 禁用 LA ₁	A/S 禁用 LA ₂	CPMVDI 位 3/5	CPMVDI 位 4/6	FA _{Total}	含义 对于 FA
0	0	0	0	0	FA _{Cmd} + FA _{DEP1} + FA _{DEP2}	真实运行。
0	0	0	1	0	FA _{Cmd} + FA _{DEP1} + FA _{DEP2}	真实运行。
0	1	1	0	0	FA _{Cmd} + FA _{DEP1} + FA _{DEP2}	模拟运行。
0	0	1	0	0	-	报警 16773 在 A/S 禁用方 面不同的引导 轴状态。
1	0	0	0	0	FA _{Cmd} + FA _{DEP1} + FA _{DEP2}	真实运行，FA 的主轴禁用不 生效。
1	0	1	1	0	FA _{Cmd} + FA _{DEP1} + FA _{DEP2}	模拟运行，FA 的主轴禁用生 效。
1	0	1	1	1	FA _{Cmd} + FA _{DEP2}	模拟运行，由 于位 4 已置 位，FA _{DEP1} 受 到抑制。
0	1	1	0	1		报警 16694 位 4/6 = 1 仅 在位 3/5 = 1 时得到支持。
1	0	0	0	1		报警 16694 位 4/6 = 1 仅 在位 3/5 = 1 时得到支持。

A/S 禁用:	进给轴/主轴禁用
	这是指产生的进给轴/主轴禁用的内部状态。通过 NC/PLC 接口信号 <Axis>.basic.out.disable（进给轴/主轴禁用）设置的主轴禁用，可以由程序测试（<HmiChan>.basic.in.progTestSelected 或 <Chan>.basic.out.progTest）等状态和 SERUPRO 覆盖，从而生成与 NC/PLC 接口信号不同的轴/主轴状态。
FA:	跟随轴
LA ₁ :	引导轴 1
LA ₂ :	引导轴 2
FA _{Total} :	跟随轴的总设定值
FA _{Cmd} :	跟随轴的独立运动分量
FA _{DEP1} :	引导轴 1 的相关运动分量
FA _{DEP2} :	引导轴 2 的相关运动分量
真实运行:	真实运行表示位置运动进入位置闭环控制。
模拟运行:	模拟运行表示没有位置运动进入位置闭环控制。真实机床轴保持停止。这对应于设置的进给轴/主轴禁用或程序测试的状态。

说明

引导轴 1 和 2 列中的状态也适用于在进给轴/主轴禁用方面具有相同状态的多个引导轴/主主轴。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.disable	LBP_Axis*.A_ASpDisable	DB31, ...DBX1.3
<Chan>.basic.out.progTest	LBP_Chan*.A_ProgTest	DB21, ...BX1.7

OP → PLC

Basic Program Plus	Basic Program	
<HmiChan>.basic.in.progTestSelected	LBP_HMI.E_MMC_ProgTest	DB21, ...DBX25.7

15.5.5.19 报警抑制 (CPMALARM)

使用 CP 关键字 CPMALARM 可以抑制与耦合相关的报警。

编程

语法: CPMALARM [FAx] = <值>

名称: Coupling Mode Alarm

功能: CPMALARM 是位编码的 CP 关键字，用于抑制耦合专用报警输出。
位逻辑运算符 B_OR、B_AND、B_NOT 和 B_XOR 可用于设置各个位。

位	值	含义
0	= 1	报警 16772 会被抑制。
1	= 1	报警 16773 会被抑制。
2	= 1	报警 16774 会被抑制。
3	= 1	报警 22012 会被抑制。
4	= 1	报警 22013 会被抑制。
5	= 1	报警 22014 会被抑制。
6	= 1	报警 22015 会被抑制。
7	= 1	报警 22016 会被抑制。
8	= 1	报警 22025 会被抑制。
9	= 1	报警 22026 会被抑制。
10	= 1	位置闭环控制已经激活时，在循环检查中为副主轴和主主轴抑制报警 22040。
11	= 1	报警 16771 会被抑制。
12 - 31		预留。

- 缺省值对应以下机床数据中的设置：
- MD11410 \$MN_SUPPRESS_ALARM_MASK（抑制专用报警输出的界面）
 - MD11415 \$MN_SUPPRESS_ALARM_MASK_2（屏蔽报警输出）

示例

程序代码	注释
CPMALARM[X2]='H300'	; 报警 22025 和 22026 会被抑制，用于耦合跟随轴 x2。

15.5.6 耦合级联

耦合级联

耦合模块可以串联。此时一个耦合模块的跟随轴/副主轴是另一个耦合模块的引导轴/主主轴。这就产生了耦合级联。

也可能有连续多个耦合级联。各个耦合模块的内部计算顺序使得在耦合关联中没有位置偏移。这同样适用于跨通道级联。

示例：

创建两个新的耦合模块。为包含跟随轴 X2 的耦合模块定义引导轴 X1。为包含跟随轴 A2 的耦合模块定义引导轴 X2 和 A1。

编程
CPDEF=(X2) CPLA[X2]=(X1) CPDEF=(A2) CPLA[A2]=(X2) CPLA[A2]=(A1)

边界条件

- 级联的可用性取决于选项（参见章节“前提条件”）。
- 现有耦合功能的耦合与通用耦合的耦合之间无法级联。
- 不允许环耦合。它会被拒绝，并发出报警 16778：
“不允许跟随轴 FAx 和引导轴 LAx 的环耦合”
当一个跟随轴又是其自身耦合模块的引导轴或前一个耦合模块中的引导轴时，会产生环耦合。

15.5.7 兼容性

15.5.7.1 匹配循环

匹配循环

通过将匹配循环作为 NC 软件的固定组成部分提供，可确保与现有耦合类型（联动、引导值耦合、电子齿轮箱和同步主轴）的耦合调用的语法和功能兼容性。这就是说，只要机器制造商/用户不需要任何新的耦合属性，就不必修改以前的耦合调用和与此相关的应用部分（例如在耦合信号的 PLC 评估中）。

分配给现有的耦合指令

匹配循环的数量对应现有耦合指令的数量。分配如下：

耦合指令	匹配循环
TRAILON	cycle700
TRAILOF	cycle701
LEADON	cycle702
LEADOF	cycle703
COUPDEF	cycle704
COUPON	cycle705
COUPONC	cycle706
COUPOF	cycle707
COUPOFS	cycle708
COUPDEL	cycle709
COUPRES	cycle710
EGDEF	cycle711
EGON	cycle712
EGONSYN	cycle713
EGONSYNE	cycle714
EGOFC	cycle715
EGOFS	cycle716
EGDEL	cycle717

存储位置

匹配循环位于“CST”目录中。

用户专用匹配循环

如有必要（功能补充），用户可以将一个匹配循环复制到“CMA”或“CUS”目录并在那里进行更改/扩展。在读入匹配循环时，考虑 CUS → CMA → CST 的顺序，并采用首先找到的循环类型，即优先选择由用户复制到 CMA / CUS 目录中的匹配循环。

说明

更新 NC 软件时，会在“CST”目录中存储一个日志文件（Changelog 文件），其中指出了对匹配循环的必要更改。

15.5.7.2 耦合类型 (CPSETTYPE)

耦合类型

如果需要根据现有耦合类型（联动、引导值耦合、电子齿轮箱和同步主轴）预设耦合属性，则在创建耦合模块（CPON/CPLON 或 CPDEF/CPLDEF）时还必须使用关键字 CPSETTYPE。

编程

语法: CPSETTYPE [FAx] = <值>

名称: Coupling Set Type

功能: 指定耦合属性的默认设置（耦合类型）。

值: 类型: STRING

值域:

- "CP" 可自由编程
- "TRAIL" 耦合类型“联动”
- "LEAD" 耦合类型“引导值耦合”
- "EG" 耦合类型“电子齿轮箱”

15.5 通用耦合

"COUP"耦合类型“同步主轴”

缺省值："CP"

示例：

程序代码	注释
CPLON[X2]=(X1) CPSETTYPE[X2]="LEAD"	； 为跟随轴 X2 与引导轴 X1 创建一个耦合模块并激活耦合模块。耦合属性被设置为与现有的耦合类型引导值耦合相对应。

默认设置

不同耦合类型的可编程耦合属性的默认设置见下表：

关键字	耦合类型				
	标准 (CP)	联动 (TRAIL)	引导值耦合 (LEAD)	电子齿轮箱 (EG)	同步主轴 (COUP)
CPDEF		-	-		
CPDEL		-	-		
CPLDEF		-	-		
CPLDEL		-	-		
CPON					
CPOF					
CPLON					
CPLOF					
CPLNUM	1.0	1.0	-	1.0	1.0
CPLDEN	1.0	1.0	-	1.0	1.0
CPLCTID	未设置	-	0	未设置	-
CPLSETVAL	CMDPOS	CMDPOS	CMDPOS	CMDPOS	CMDPOS
CPFRS	BCS	BCS	BCS	BCS	MCS
CPBC	NOC	NOC	NOC	FINE	IPOSTOP

关键字	耦合类型				
	标准 (CP)	联动 (TRAIL)	引导值耦合 (LEAD)	电子齿轮箱 (EG)	同步主轴 (COUP)
CPFPOS + CPON	未设置	-	-	未设置	未设置
CPFPOS + CPOF	未设置	-	-	-	未设置
CPFMSON	CFAST	CFAST	CCOARSE	NRGT	CFAST
CPFMON	STOP	VL ¹⁾ : STOP HL ²⁾ : CONT	VL ¹⁾ : CONT HL ²⁾ : CONT	STOP	CONT
CPFMOF	STOP	VL ¹⁾ : STOP HL ²⁾ : STOP	VL ¹⁾ : STOP HL ²⁾ : STOP	STOP	CONT
CPLPOS + CPON	未设置	-	-	未设置	-
CPMRESET	NONE	MD20110	MD20110	MD20110	MD20110
CPMSTART	NONE	MD20112	MD20112	MD20112	MD20112
CPMPRT	NONE	MD20112 / MD22620 ³⁾	MD20112 / MD22620 ³⁾	MD20112 / MD22620 ³⁾	MD20112 / MD22620 ³⁾
CPLINTR	0.0	0.0	0.0	0.0	0.0
CPLINSC	1.0	1.0	1.0	1.0	1.0
CPLOUTTR	0.0	0.0	0.0	0.0	0.0
CPLOUTSC	1.0	1.0	1.0	1.0	1.0
CPSYNCOV	MD37200	MD37200	MD37200	MD37200	MD37200
CPSYNFIP	MD37210	MD37210	MD37210	MD37210	MD37210
CPSYNCOV2	MD37202	MD37202	MD37202	MD37202	MD37202
CPSYNFIP2	MD37212	MD37212	MD37212	MD37212	MD37212
CPSYNCOV	MD37220	MD37220	MD37220	MD37220	MD37220
CPSYNFIV	MD37230	MD37230	MD37230	MD37230	MD37230
CPMBRAKE					
位 0	1	-	-	-	-

15.5 通用耦合

关键字		耦合类型				
		标准 (CP)	联动 (TRAIL)	引导值耦合 (LEAD)	电子齿轮箱 (EG)	同步主轴 (COUP)
CPMVDI						
	位 3	0	0	0	0	0
	位 4	1	1	1	1	1
	位 5	0	0	0	0	0
	位 6	1	1	1	1	1
CPMALARM						
		MD11410 MD11415	MD11410 MD11415	MD11410 MD11415	MD11410 MD11415	MD11410 MD11415
图例: 1) 预处理 2) 主运行 3) 还取决于 MD22621 - 不相关或不允许						

其他属性

下表总结了设置了耦合类型 (CPSETTYPE) 时其他属性的值域或可用性:

	标准 (CP)	联动 (TRAIL)	引导值耦合 LEAD)	电子齿轮箱 (EG)	同步主轴 (COUP)
引导轴数量	≤ 5	≤ 2	1	≤ 5	1
跟随轴的轴类型	轴/主轴	轴/主轴	轴/主轴	轴/主轴	主轴
定义/删除耦合模块	CPDEF/CPDEL 或 CPON/CPOF	CPON/CPOF	CPON/CPOF	CPDEF/CPDEL	CPDEF/CPDEL
定义/删除引导轴	CPLDEF/CPLDEL 或 CPLON/CPLOF	CPLON/CPLOF	CPLON/CPLOF	CPLDEF/CPLDEL	CPLDEF/CPLDEL
级联	+	+	+	+	-

	标准 (CP)	联动 (TRAIL)	引导值耦合 (LEAD)	电子齿轮箱 (EG)	同步主轴 (COUP)
主主轴的 动态响应考虑	-	-	-	-	+
隐性选择/取消 选择位置闭环 控制 ¹⁾	-	-	-	-	+
图例: ¹⁾ 另见章节“同步主轴 (页 519)” - 不相关或不允许					

指定属性的可用性取决于可用规格（参见章节“前提条件”）。

示例:

耦合类型联动 (CPSETTYPE="TRAIL") 最多允许两个引导轴和级联。但这在基本规格中不可用，需要 CP-EXPERT 选项。

边界条件

- CPSETTYPE 也可以在同步动作中编程。
- 如果设置了耦合类型 (CPSETTYPE)，则某些耦合属性已预设且无法再更改。使用关键字进行后续更改会导致错误，会被拒绝并发出报警：

CPSETTYPE=	TRAIL	LEAD	EG	COUP
CPDEF	报警 16686	报警 16686		
CPDEL	报警 16686	报警 16686		
CPLDEF				
CPLDEL				
CPON			报警 16686	报警 16686
CPLON				
CPOF			报警 16686	报警 16686
CPLOF				

15.5 通用耦合

CPSETTYPE=	TRAIL	LEAD	EG	COUP
CPRES	报警 16686	报警 16686	报警 16686	
CPLNUM	报警 16686	报警 16686		
CPLDEN	报警 16686	报警 16686		
CPLCTID	报警 16686			报警 16686
CPLSETVAL	CMDVEL 时 报警 16686	CMDVEL 时 报警 16686	CMDVEL 时 报警 16686	
CPFRS				BCS 时 报警 16686
CPBC	报警 16686	报警 16686		
CPFPOS + CPON	报警 16686	报警 16686		
CPFPOS + CPOF	报警 16686	报警 16686	报警 16686	
CPFMSON	报警 16686	报警 16686		报警 16686
CPFMON	报警 16686	ADD/STOP 时 报警 16686	报警 16686	STOP 时 报警 16686
CPFMOF	报警 16686	ADD/STOP 时 报警 16686	ADD 时 报警 16686	ADD 时 报警 16686
CPLPOS + CPON	报警 16686	报警 16686		报警 16686
CPMRESET	报警 16686	报警 16686	报警 16686	报警 16686
CPMSTART	报警 16686	报警 16686	报警 16686	报警 16686
CPMPRT	报警 16686	报警 16686	报警 16686	报警 16686
CPMBRAKE	报警 16686	报警 16686	报警 16686	报警 16686
引导轴数量 ($\sum$ LA)	$\sum$ LA > 2 时 报警 16672	$\sum$ LA > 1 时 报警 16672	$\sum$ LA > 5 时 报警 16672	$\sum$ LA > 1 时 报警 16672
跟随轴的轴类型				轴时 报警 14092

15.5.7.3 配置的耦合 (CPRES)

如果设置了耦合类型“同步主轴”（参见 CPSETTYPE），则可以激活存储在机床数据和设置数据中的耦合参数，而不是编程的耦合属性。

更多信息

同步主轴 (页 519)

编程

语法:	CPRES= (<副主轴>)
名称:	Coupling Restore
功能:	激活与副主轴 FAx 的同步主轴耦合的配置数据。
副主轴:	类型: AXIS
	值域: 所有在通道中定义的主轴名称

示例:

编程	注释
CPLON[S2]= (S1) CPSETTYPE[S2]="COUP"	; 为副主轴 S2 与主轴 S1 创建一个耦合模块并激活耦合模块。耦合属性被设置为与现有的耦合类型同步主轴相对应。
...	
CPRES= (S2)	; 激活与副主轴 S2 的同步主轴耦合的配置数据。

边界条件

- 只有设置了耦合类型“同步主轴”(CPSETTYPE="COUP") 时才允许 CPRES。
- 将 CPRES 应用于已激活的耦合会导致重新同步。
- 将 CPRES 应用于未定义的耦合模块无效。

15.5.8 跨通道耦合、跨通道取轴

跟随轴和引导轴必须是调用通道的已知轴。

跟随轴

在零件程序中编程 CP 关键字时，根据取轴配置 (MD30552) 请求将跟随轴交换到通道中，或使用语言指令 GETD 将其取到调用通道中。

激活耦合模块后，跟随轴的取轴仍然只允许在通道中进行。即仍然可以从通道交换到主运行，反之亦然，但不能跨通道边界取轴。适用取轴的边界条件和属性。禁用耦合模块后，再次使能通过通道轴进行取轴。

更多信息

功能手册之基本功能：运行方式组、通道、跨通道取轴

引导轴

无论耦合状态如何，都可以进行引导轴的取轴。

15.5.9 回转轴的特性

回转轴作为跟随轴或引导轴

可将回转轴耦合到线性轴，反之亦然。此时应注意，通过耦合规律直接将度数对应到 mm。

示例：

A = 回转轴，X = 线性轴

编程	注释
N10 G0 A0 X0	; 运行： X = 0 mm, A = 0 度
N20 CPON=(A) CPLA[A]=(X) CPLNUM[A,X]=2	; 为回转轴 A 创建并激活一个耦合模块，以线性轴 X 作为引导轴。耦合系数的值为 2。
N30 X100	; 运行： X = 100 mm, A = 200 度 (= 100*2)

模数减少的回转轴作为引导轴

使用模数减少的回转轴作为引导轴时，输入变量不会随着引导轴的减少而减少。仍然使用未减少的位置作为输入变量，即考虑已经过的行程。

示例：

A = 模数减少的回转轴，X = 线性轴

编程	注释
N10 G0 A0 X0	; 运行： A = 0 度，X = 0 mm
N20 CPON=(X) CPLA[X]=(A) CPLNUM[X,A]=0.5	; 为线性轴 X 创建并激活一个耦合模块，以回 转轴 A 作为引导轴。耦合系数的值为 0.5。
N30 A200	; 运行： A = 200 度，X = 100 mm (= 200*0.5)
N40 A=IC(200)	; A 沿正方向运行 200 度到 400 度，显示 A = 40。X 运行 100 mm 到 200。
N50 A=IC(100)	; A 从 40 度运行到 140 度，X 再运行 50 mm 到 250。
N60 A=ACP(80)	; A 沿正方向运行到 50 度，运行行程为正方 向 300 度。X 相应地沿正方向运行 150 mm。因此终点位置是 X = 400。

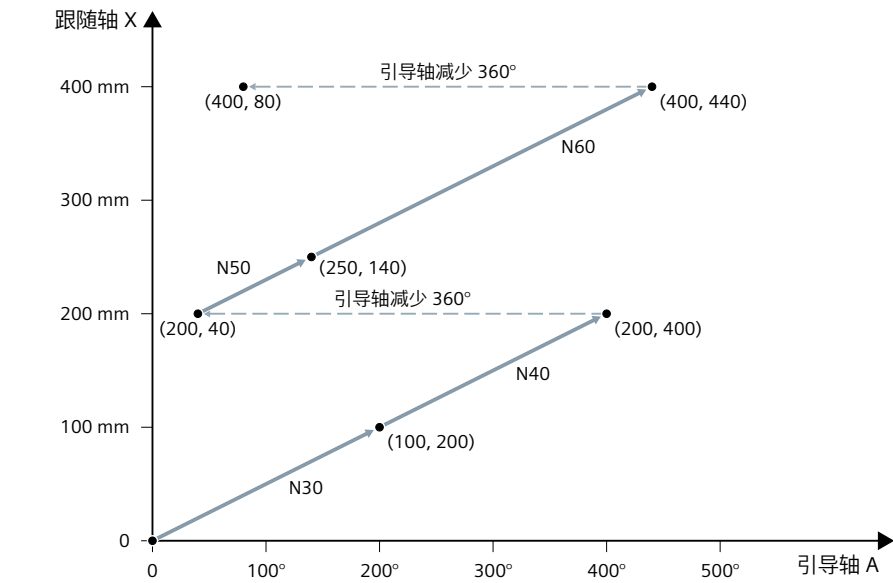


图 15-12 示例：模数减少的回转轴耦合到线性轴

(...) X、A 的位置显示

15.5.10 上电、复位等时的特性

上电

上电时没有耦合生效。不存在耦合模块。

复位

可以为每个耦合模块单独设置复位时的特性（参见 CPMRESET）。可以激活、禁用耦合或保留当前状态。

运行方式切换

在运行方式切换时耦合保持有效。只有在 JOG-REF 运行方式下，跟随轴回参考点时才会抑制耦合（不取消选择！）。

回参考点运行

跟随轴的 G74 在耦合生效时无法执行。系统输出一个报警。

如果选择 JOG-REF 运行方式并运行跟随轴，则耦合会被抑制。只有在退出 JOG-REF 运行方式后，才会再次执行耦合。

SERUPRO

SERUPRO 进程将模拟通用耦合并提供重启值。

轴耦合时，始终假定设定值耦合进行模拟，即如果采用实际值耦合，则在 SERUPRO 进程中切换到设定值耦合。这可能表示模拟不正确。

由于模拟速度的提高和轴动态响应限制的消除，可能会出现与真实过程的更大偏差。

15.5.11 CP 软件限制监控

15.5.11.1 功能

该功能有什么作用？

在接近软件限位开关时，“CP 软件限制监控”功能可改善跟随轴或副主轴在轴运行方式下的制动特性。

优点:

- 防止超过软件限位开关。
- 如果可能，保持耦合的同步运行。

可用性

“CP 软件限制监控”功能只能为以下跟随轴激活:

- “可自由编程”类型的通用耦合 (CPSETTYPE[FAx] = "CP")
- 最多包含一个有效引导轴/主轴的耦合 (CPSETTYPE[FAx] 不等于“CP”、联动、电子齿轮箱、引导值耦合或同步主轴的通用耦合)

在所有其他情况下 (以及“CP 软件限制监控”未激活时)，之前的软件限位开关监控功能有效。

监控和设置制动

在每个插补节拍中，“CP 软件限制监控”功能检查是否可以为下一个插补节拍使能跟随轴/副主轴的运动，因为它总是可以在软件限位开关之前及时停止。如果不是这种情况，则为跟随轴设置“CP 软件限制停止”，这意味着在轮廓上无条件地进行方向特定的制动。

将制动传送到轴

设置制动会导致跟随轴的与耦合无关的运动分量 (CMD 和 CORR 分量) 停止。而跟随轴的耦合相关运动分量 (DEP 分量) 只能通过制动引导轴来制动。

对于最多包含一个有效引导轴/主轴的耦合 (CPSETTYPE[FAx] 不等于“CP”、联动、电子齿轮箱、引导值耦合或同步主轴的通用耦合)，也对 (唯一) 有效引导轴 (根据耦合系数) 设置碰撞方向上的制动。

对于“可自由编程”类型的通用耦合 (CPSETTYPE[FAx] = "CP")，只有在编程了耦合属性 CPMBRAKE[FAx] 位 0 = 1 时，制动才会传送到所有生效的引导轴 (参见“对停止信号和停止指令的响应 (CPMBRAKE) (页 699)”)。

碰撞方向为引导轴专用，取决于耦合系数：如果耦合系数为负，则碰撞方向反转，如果耦合系数为“零”，则不进行传送。对于非线性耦合 (如曲线表)，耦合系数由确定为线性近似值的梯度导出。

说明

只要耦合生效，跟随轴的设置制动就只能影响其引导轴。

制动特性

在遵守耦合规律的同时，以轴的标准加速度斜坡在轮廓上沿软件限位开关的方向进行制动。

制动时，显示停止状态 75“制动请求”，并根据情况显示优先级更高的停止状态 22“等待：缺少主轴使能”、12“等待轴/主轴使能”或 71“等待变压器轴使能”。

如果跟随轴由于“CP 软件限制停止”而停止，并且不能再接近软件限位开关，则会显示以下报警：

10625“%?C{通道 %1: %}程序段 %3 跟随轴/副主轴 %2 使用 CP 软件限制停止 %4”

因此，在“最终”停止之前，仍然可以考虑通过沿相反方向运动跟随轴来化解情况。

说明

使用“CP 软件限制监控”时，遵守同步运行的耦合优先。因此，如果要遵守耦合规律，则不能保证轴精确停止。如果制动传送到引导轴并且插补器根据需要制动，则跟随轴将在软件限位开关之前停止。此时，到软件限位开关的最大距离由跟随轴的当前最大加速度乘以插补循环时间的平方得出。

退刀

退刀是用户的责任。

可用于诊断目的的有：

- 停止状态 75“制动请求”
(如果轴有运行指令，但因制动请求而停止或不允许运行)
- 和系统变量：
 - \$AA_BRAKE_STATE (当前制动状态)
 - \$AA_BRAKE_CONDB (基准坐标系中插补器停止的上下文相关条件)
 - \$AA_BRAKE_CONDM (机床坐标系中插补器停止的上下文相关条件)
 - 包括由此导出的 OPI 变量 aaBrakeState、aaBrakeCondB 和 aaBrakeCondM。

退刀示例：

1. 应答报警 10625。
2. 切换到 JOG 运行方式。
3. 用运行键运行引导轴，使跟随轴远离软件限位开关。

15.5.11.2 参数设置

激活

“CP 软件限制监控” 功能通过以下机床数据轴专用激活：

MD30455 \$MA_MISC_FUNCTION_MASK（轴功能）

位	值	含义
11	0	CP 软件限制监控未生效。
	1	CP 软件限制监控生效。

15.5.11.3 编程

将制动传送到引导轴

对于“可自由编程”类型的通用耦合 (CPSETTYPE[FAx] = "CP")，可通过编程耦合属性 CPMBRAKE（参见“对停止信号和停止指令的响应 (CPMBRAKE) (页 699)”）来设置由“CP 软件限制监控”功能触发的跟随轴制动是否也应传送到引导轴。

15.5.11.4 边界条件

可能的故障源

如果轴没有按需要制动，可能有以下原因：

至少一个引导轴不响应制动，因为它本身是一个跟随轴（级联），而制动未传送到它的引导轴。只有相关跟随轴适用以下内容时，制动才会（递归）传送到引导轴：CPSETTYPE[FAx] = "CP" 和 CPMBRAKE[FAx] 位 0 = 1

即使所有轴都按需要制动，如果这些值没有正确配置或编程，也会由于不同的加速度和/或耦合系数而发生轮廓故障。

即使满足了前面的所有前提条件，仍然有各种原因不能防止跟随轴超过软件限位开关：

- 引导轴的当前最大加速度发生变化。
- 跟随轴的当前最大加速度减小。
- 激活耦合或响应 CP 软件限制监控时，跟随轴的速度已经很高，不再能够以当前最大加速度及时提前停止。

如果不再能够及时停止，之前的软件限位开关监控仍会启动，使跟随轴在软件限位开关处停止。

转换时的制动性能

如果要制动的轴是一个转换的输出端，并且有一个在碰撞方向上的 (MCS) 运行指令，则制动在两个方向上传送到转换的所有输入轴，该转换发生轨迹停止。有以下特例：这不适用于转换的独立轴。对于 TRANSMIT 和 TRAANG，这是指在机床数据 \$MC_TRAFO_AXES_IN_*[2] 中指定的轴。在转换过程中，制动仅传送到其相关的输入轴。

DRF 偏移

设置 CP 软件限制停止会导致 DRF 运动中止，如下例所示：

基本运动是在正方向上以非常低的速度进行的轨迹运动。反方向上较大的 DRF 补偿运动对此产生影响。这就导致了一个负方向的运行指令。如果现在用 CP 软件限制停止来停止运动，就会在使用标准机床数据时导致 DRF 运动中止，并且在取消停止后仅继续轨迹运动。

15.5.11.5 示例

示例 1：“可自由编程”类型的通用耦合 (CPSETTYPE[FAx] = "CP")

配置：

MD26110 \$MA_POS_LIMIT_PLUS[AX2]=15 ;

跟随轴第 1 个软件限位开关的位置，
用于正方向的运行范围限制

MD30455 \$MA_MISC_FUNCTION_MASK[AX2] =
'H800' ;

位 11 = 1 (CP 软件限制监控已激活)

编程：

; X 和 Y 的起始位置

N100 G0 X0 Y0

; 激活耦合 FA=Y LA=X

; 用 CPMBRAKE 位 0 设置 (将制动传送到引导轴)

N120 CPON=Y CPSETTYPE="CP" CPLA=X CPLDEN=2 CPMBRAKE B_OR='H1'

N140 G4 F2

; 运行引导轴 X

N200 G1 X=1000 F1000

示例 2：最多包含一个有效引导轴/主主轴的耦合

如果设置了 MD30455 MA_MISC_FUNCTION_MASK[AX2] 位 11，则通过以下调用进行 CP 软件限制监控，包括制动到引导轴的传送：

TRAILON(Y,X,0.5) ; 定义和激活联动轴 Y 与引导轴 X 的耦合。

EGDEF(Y,X,1) ; 定义从 X 到 Y（跟随轴）的设定值耦合的 EG 轴组。
EGON(Y,"FINE",X,1,2) ; 激活耦合。

15.5.12 故障特性

15.5.12.1 快速停止

功能

快速停止导致轴/主轴在没有斜坡的情况下停止，即指定速度设定值为零。该预设值会导致在电流限值时进行制动。伺服使能保持有效。

在以下情况下设置快速停止：

- Stop A 和 Stop C（安全集成）
- 快速停止作为配置的制动特性的报警
- 达到硬件限位开关且快速停止作为配置的制动特性：
MD36600 \$MA_BRAKE_MODE_CHOICE = 1

切换到实际值耦合

一旦报告通用耦合的主主轴快速停止，该主主轴的实际值将用于计算设定值。

15.5 通用耦合

切换到实际值耦合时没有跳跃，并且一直有效，直到主主轴上再次出现伺服使能和脉冲使能，且不再进行位置补偿。只有满足这些条件时，才会重新进行用 CPLSETVAL 等编程的设定值计算。

说明

到达硬件限位开关触发的快速停止不会导致切换到实际值耦合。

副主轴的特性

如果在一个主主轴上检测到快速停止且副主轴本身不执行快速停止，则副主轴将尝试在其设置的动态响应范围内跟随主主轴的运动。由于建立了位置同步，跟随轴可能相对于要逼近的位置出现过冲。

可以跨 NCU 检测到引导轴/主主轴快速停止开始。

说明

对于同步主轴耦合类型 (CPSETTYPE="COUP")，出现伺服报警时，主主轴和副主轴同时快速停止。

15.5.13 跟踪同步运行偏差

15.5.13.1 基本介绍

同步运行偏差

要在正面和背面进行的工件加工，需要将工件转移到另一个工件夹持装置（例如副主轴卡盘）。

将工件从正面加工转移到背面加工时，由于工件夹具的关闭，可能会发生位置偏移。造成这种情况的原因可能是工件有棱角，或者在运动过程中快速关闭工件夹持装置（卡盘）时产生了旋转脉冲。根据工件的阻力，可以通过耦合中涉及的两个电机的电流消耗的增加和/或通过工件的扭转来识别张力。

根据设置的同步运行公差和偏移的大小，这可能导致以下 NC/PLC 接口信号复位：

<Axis>.coupling.in.syncSpindleCoarseReached（粗同步）和/或

<Axis>.coupling.in.syncSpindleFineReached（精同步）

在设定值耦合中，位置和速度设定值根据编程的耦合规律精确计算并输出到控制模块。假设有相同的驱动装置和刚性工件，则调整性偏差的一半将设置为主主轴和副主轴的设定值/实际值差异。

功能

“跟踪同步运行偏差”功能的任务是识别施加在实际值侧的位置偏移，并在副主轴的设定值计算中对其进行补偿。

前提条件

该功能的前提条件是通过零件/夹紧装置实现闭合耦合。

方式

确定同步运行偏差有两种不同的方法：

1. 同步运行偏差由 NC 确定（参见“测量同步运行偏差 (页 726)”）。
2. 偏差值已知，由用户直接指定（参见“直接指定同步运行偏差 (页 729)”）。

在这两种情况下，偏差值都会作为补偿值纳入副主轴的设定值计算中。

可用性

“跟踪同步运行偏差”功能是为 MCS 耦合 (CPFRS="MCS") 开发的。因此该功能也可用于“同步主轴”耦合类型 (CPSETTYPE="COUP")。

与其他叠加运动（例如差速）一样，该功能的可用性取决于选项（参见“前提条件 (页 651)”）。

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.coupling.in.syncFineReached	LBP_Axis*.E_SyncronFine	DB31, ...DBX98.0
<Axis>.coupling.in.syncCoarseReached	LBP_Axis*.E_SyncronCoarse	DB31, ...DBX98.1

15.5.13.2 测量同步运行偏差

副主轴同步运行时，控制系统测量设定值位置 and 实际值位置之间的差异。这会得出补偿值，保存在一个系统变量中。

前提条件

为了使控制系统能够确定补偿值，必须满足下列条件：

- 设置了“同步主轴”耦合类型 (CPSETTYPE="COUP") 时的前提条件：
 - 耦合有且只有一个主主轴（在 CPSETTYPE="COUP" 时满足）。
 - 耦合系数 (CPLNUM 和 CPLDEN 的商) 为 1 或 -1。
 - 跟随值由主主轴的设定位置 ("DV") 或实际位置 ("AV") 得出。
 - 必须达到设定值侧同步运行：
`<Axis>.coupling.in.ProcessActive`（正在进行同步）= 0
 - 设定值侧同步运行不得再次失效。
 - 不得有叠加运动
(`<Axis>.coupling.in.syncSpindleMotionOverlaid = 0`)。
 - 需要对主主轴进行动态响应限制，以防止副主轴过冲。
- 使用 CPFRS="MCS" 的自由通用耦合的前提条件：
 - 耦合已配置的主轴。
 - 耦合有且只有一个主主轴。
 - 耦合系数 (CPLNUM 和 CPLDEN 的商) 为 1 或 -1。
 - 跟随值由主主轴的设定位置 (CPLSETVAL="CMDPOS") 或实际位置 (CPLSETVAL="ACTPOS") 得出。
 - 必须达到设定值侧同步运行：
`<Axis>.coupling.in.syncSpindleProcessActive`（正在进行同步）= 0
 - 设定值侧同步运行不得再次失效。
 - 不得有叠加运动
(`<Axis>.coupling.in.syncSpindleMotionOverlaid = 0`)。
 - 需要对主主轴进行动态响应限制，以防止副主轴过冲。

说明

主主轴的动态响应限制

“主主轴的动态响应限制”属性在设置了“同步主轴”耦合类型 (CPSETTYPE="COUP") 时自动给出。对于其他耦合类型，用户/机床制造商有责任采取适当措施避免副主轴的动态过载。

激活

通过将以下 NC/PLC 接口信号设置为“1”来激活同步运行偏差的测量和跟踪：

`<Axis>.coupling.out.syncCorrection`（跟踪同步运行）

该信号只对副主轴有效。

说明

在以下情况下，将忽略信号 `<Axis>.coupling.out.syncCorrection`（跟踪同步运行）：

- 进给轴/主轴禁用生效 (`<Axis>.basic.out.disable = 1`)。
- 程序测试已选择。
- SERUPRO 生效。

如果在“跟踪同步运行偏差”功能已激活时出现其中一种情况，则“跟踪同步运行偏差”功能将被禁用。

测量的时间点

测量和计算补偿值的时间点取决于以下机床数据中位 7 的设置：

MD30455 \$MA_MISC_FUNCTION_MASK（轴功能）

位	值	含义
7	0	只要设置了 NC/PLC 接口信号 <code><Axis>.coupling.out.syncCorrection</code> （跟踪同步运行）并且设定值侧同步运行可用，就会不断计算补偿值（循环计算）。
	1	补偿值仅在将 NC/PLC 接口信号 <code><Axis>.coupling.out.syncCorrection</code> （跟踪同步运行）从 0 设置为 1 时计算（边沿分析）。

说明

如果在减小主主轴和副主轴之间的张力时预计会延长时间，则应设置位 7 = 0。此时接口信号为状态控制。

减小张力所需的时间取决于各种因素（例如位置闭环控制的 K_v 系数、电机的加速能力），应通过实验确定。

测量过程

只有达到设定值侧同步运行时，NC/PLC 接口信号 <Axis>.coupling.out.syncCorrection（跟踪同步运行）才能生效：

<Axis>.coupling.in.ProcessActive（正在进行同步）= 0

在信号 <Axis>.coupling.out.syncCorrection 激活期间，在插补节拍中读取副主轴的实际值并确定与设定位置的差异，也可在信号上升沿进行一次（取决于 MD30455 中位 7 的设置；参见“测量的时间点”段落）。

设定值侧和实际值侧同步运行位置之间的差异是补偿值。这存储在相关副主轴的以下系统变量中：

\$AA_COUP_CORR[S<n>]（副主轴：同步主轴耦合时的补偿值）

说明

应确保在测量期间引导轴和跟随轴的速度尽可能恒定，并且不会出现加速度跳跃。

示例

激活同步主轴 [S2] 的耦合时，还编程了 77 度的位置偏移：
CPON=S2 ... CPFPOS[S2]=AC(77)

关闭工件夹具会产生机械位置偏移。这导致实际值侧的位置偏移为 81 度。

激活“跟踪同步运行偏差”功能(<Axis>.coupling.out.syncCorrection = 1)且达到设定值侧同步运行(<Axis>.coupling.in.ProcessActive = 0)后，实际值侧位置偏移(\$VA_COUP_OFFS[S2] = 81)与设定值侧位置偏移(\$AA_COUP_OFFS[S2] = 77)比较。这会得出 4 度的补偿值，保存在系统变量 \$AA_COUP_CORR[S2] 中。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.disable	LBP_Axis*.A_ASpDisable	DB31, ...DBX1.3
<Axis>.coupling.out.syncCorrection	LBP_Axis*.A_Tracksync	DB31, ...DBX31.6

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.coupling.in.MotionOverlaid	LBP_Axis*.E_OverlayMotion	DB31, ...DBX98.4
<Axis>.coupling.in.ProcessActive	LBP_Axis*.E_Sync	DB31, ...DBX99.4

15.5.13.3 直接指定同步运行偏差

如果偏差值已知，也可以直接写入相关副主轴的系统变量 \$AA_COUP_CORR。这可以通过零件程序或同步动作来实现。

说明

必须确保仅在建立机械耦合后才写入系统变量。否则在关闭卡盘时可能再次发生偏移。

前提条件

为了能够从零件程序或同步动作中写入系统变量 \$AA_COUP_CORR，必须在最后一次控制系统启动后为相关副主轴至少激活一次通用 MCS 耦合。

15.5.13.4 同步运行补偿

如果补偿值 \$AA_COUP_CORR[S<n>] 不等于零，且副主轴 S<n> 的一个通用 MCS 耦合生效（通过 CPFRS="MCS" 或 CPSETTYPE="COUP"），则设置以下 NC/PLC 接口信号：

<Axis>.coupling.in.CorrectionActive（同步运行补偿纳入计算）

补偿值纳入耦合模块中副主轴的设定值计算中。通过调整耦合偏移的设定值，实现主主轴和副主轴之间的松弛。

通过比较实际值和经过补偿的设定值可以得出新的同步运行信号。补偿成功后同步运行信号应再次出现：

<Axis>.coupling.in.syncCoarseReached（粗同步）和/或

<Axis>.coupling.in.syncFineReached（精同步）

补偿值的实施可以与同步运行信号的实现一起进行，因为在张力情况下的“跟踪同步运行偏差”功能可以确保更好地实现同步运行。实施时的加速能力最多降低到最大加速度和最大速度的 10%。

如果完全实施 \$AA_COUP_CORR[S<n>]，则设置以下 NC/PLC 接口信号：

<Axis>.coupling.in.syncSpindleCorrectionDone（已实施同步运行补偿）

15.5 通用耦合

这也适用于 \$AA_COUP_CORR[S<n>] 为零且完全不实施的情况。

在同步运行补偿结束时，必须将 NC/PLC 接口信号
<Axis>.coupling.out.syncCorrection（跟踪同步运行）复位为“0”，以恢复耦合的刚性。

通过复位信号 <Axis>.coupling.out.syncCorrection 或关闭耦合（用 CPOF），补偿值不再改变。此时系统变量 \$AA_COUP_CORR[S<n>] 提供一个常数值。

只要不通过将系统变量 \$AA_COUP_CORR[S<n>] 设置为“0”来删除补偿值，补偿值就会纳入计算，该操作最迟必须在工件从主轴上移除后进行。

说明

由系统变量 \$AA_COUP_CORR[S<n>] 进行的设定值补偿会影响副主轴的所有后续编程，就像位置偏移一样，类似于 MCS 中的 DRF 偏移。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.coupling.out.syncCorrection	LBP_Axis*.A_Tracksync	DB31, ...DBX31.6

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.coupling.in.syncFineReached	LBP_Axis*.E_SyncronFine	DB31, ...DBX98.0
<Axis>.coupling.in.syncCoarseReached	LBP_Axis*.E_SyncronCoarse	DB31, ...DBX98.1
<Axis>.coupling.in.CorrectionDone	LBP_Axis*.E_SyncRunCorrDOut	DB31, ...DBX99.2
<Axis>.coupling.in.CorrectionActive	LBP_Axis*.E_SyncRunCorrIncl	DB31, ...DBX103.0

15.5.13.5 同步运行补偿的诊断

出于诊断目的，\$AA_COUP_CORR 的当前值（跟踪同步运行偏差的补偿值）显示在“服务轴/主轴”窗口中的“与引导轴/主主轴设定值的位置偏移”一行下。

利用系统变量 \$AA_COUP_CORR_DIST（\$AA_COUP_CORR 的剩余行程）可以确定，还要实施补偿值的多少行程。

15.5.13.6 删除同步运行补偿

方式

删除同步运行补偿有以下方法：

- 将值“0”写入变量 `$AA_COUP_CORR[S<n>]`
同步运行补偿通过一个加速能力降低的斜坡来减小（与实施补偿值时一样）。
- 通过 PLC 删除同步运行补偿
随着 NC/PLC 接口信号：
`<Axis>.coupling.out.syncResetCorrection`（删除同步运行补偿）
的上升沿，变量 `$AA_COUP_CORR[S<n>]` 置零，同步运行补偿如下删除：
 - 如果主轴处于转速控制模式，则补偿运动停止。然后在设定位置采用现有的同步运行补偿。
 - 在所有其他情况下，已实施的同步运行补偿以与变量 `$AA_COUP_CORR[S<n>]` 设置为零时完全相同的方式撤回。

前提条件

删除同步运行补偿的前提条件是当前没有计算补偿值（参见“测量同步运行偏差 (页 726)”章节）。

删除过程结束

删除过程结束时，置位以下 NC/PLC 接口信号：

`<Axis>.coupling.in.syncSpindleCorrectionDone`（已实施同步运行补偿）

此外，如果复位了 NC/PLC 接口信号

`<Axis>.coupling.in.syncSpindleCorrectionActive`（同步运行补偿纳入计算），则也可以复位 NC/PLC 接口信号

`<Axis>.coupling.out.syncResetCorrection`（删除同步运行补偿）。

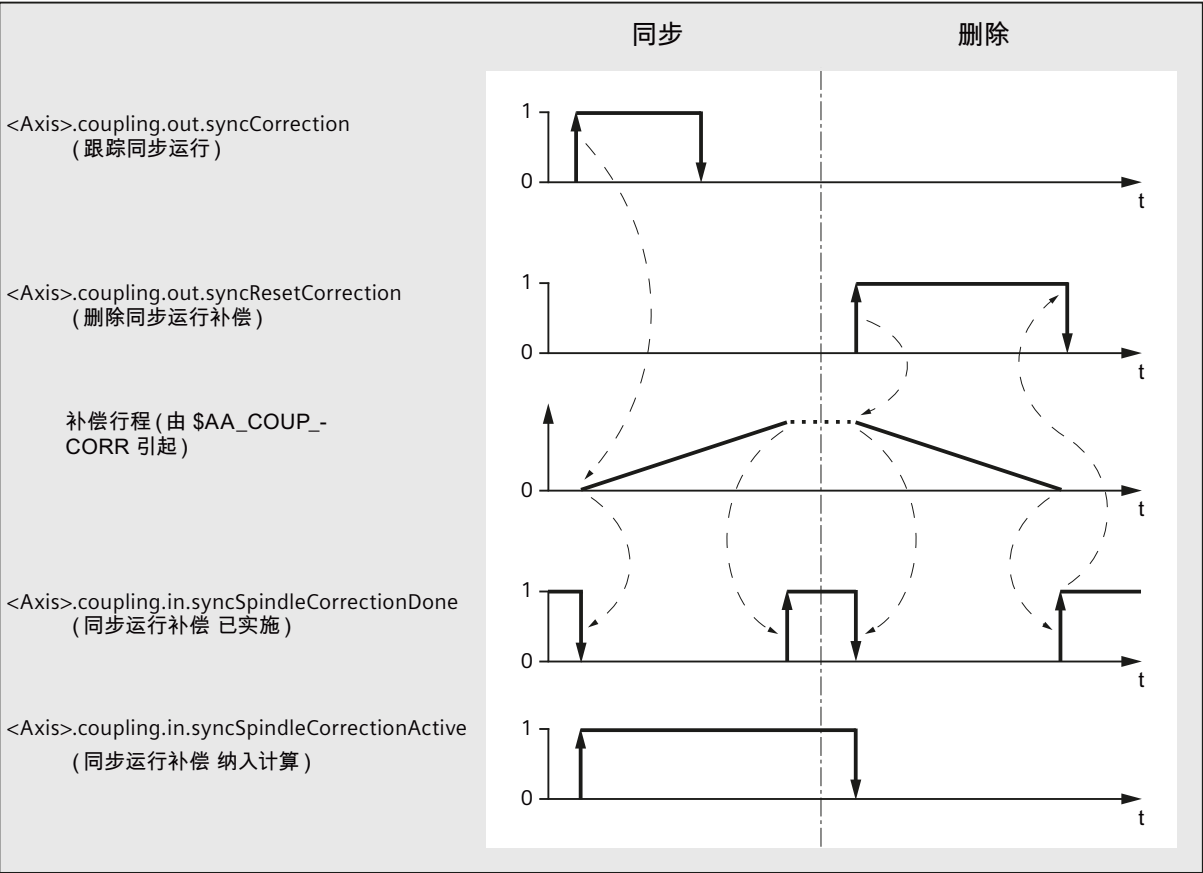


图 15-13 用于同步和删除同步运行补偿的时间图

说明

只要尚未完全实施补偿行程，且 NC/PLC 接口信号 `<Axis>.coupling.out.syncResetCorrection`（删除同步运行补偿）未复位，写入 `$AA_COUP_CORR[S<n>]` 就无效。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<code><Axis>.coupling.out.syncCorrection</code>	<code>LBP_Axis*.A_Tracksync</code>	<code>DB31, ...DBX31.6</code>
<code><Axis>.coupling.out.syncResetCorrection</code>	<code>LBP_Axis*.A_DelSyncRunCorr</code>	<code>DB31, ...DBX31.7</code>

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.coupling.in.CorrectionDone	LBP_Axis*.E_SyncRunCorrDOut	DB31, ...DBX99.2
<Axis>.coupling.in.CorrectionActive	LBP_Axis*.E_SyncRunCorrIncl	DB31, ...DBX103.0

15.5.13.7 边界条件

多个副主轴

如果一个主主轴有多个副主轴，则可以用轴的 NC/PLC 接口信号 <Axis>.coupling.out.syncCorrection（跟踪同步运行）分别处理每个副主轴。

写入变量 \$AA_COUP_CORR

只有与相关轴/主轴的通用 MCS 耦合已至少激活一次时，从零件程序或同步动作中写入系统变量 \$AA_COUP_CORR 才有效。

补偿值

如果补偿值 \$AA_COUP_CORR 通过零件程序/同步动作写入，并由于激活“跟踪同步运行偏差”功能 (<Axis>.coupling.out.syncCorrection = 1) 而确定，则始终是最后发生的事件生效。

删除同步运行补偿

在删除同步运行补偿的过程中，写入补偿值 \$AA_COUP_CORR（通过零件程序或同步动作或者在计算时）无效。

通道/运行方式组复位时的特性

在通道/运行方式组复位时，同步运行补偿不会被删除，而会保留。

回参考点运行和零脉冲同步时的特性

在主轴回参考点运行和零脉冲同步时，同步运行补偿会自动删除。

在回参考点运行/零脉冲同步的过程中，不得设置系统变量 \$AA_COUP_CORR，因此也不会测量同步运行偏差。

此外，必须在开始回参考点运行/零脉冲同步之前结束同步运行补偿的实施。

中断时的特性

在中断（如急停）时，同步运行补偿会自动删除，现有的同步运行补偿用于设定位置，NC/PLC 接口信号 <Axis>.coupling.in.syncSpindleCorrectionDone（已实施同步运行补偿）置位。

如果在取消中断后，通用 MCS 耦合仍然有效，并且 NC/PLC 接口信号 <Axis>.coupling.out.syncCorrection（跟踪同步运行）置位，则适用：

- 如果信号由电平触发（MD30455 \$MA_MISC_FUNCTION_MASK，位 7 = 0），则测量同步运行偏差并将其写入 \$AA_COUP_CORR，相应地补偿设定值。
- 如果信号由边沿触发（MD30455 \$MA_MISC_FUNCTION_MASK，位 7 = 1）并要测量同步运行偏差，则必须带来 NC/PLC 接口信号 <Axis>.coupling.out.syncCorrection（跟踪同步运行）的另一个上升沿。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.coupling.out.syncCorrection	LBP_Axis*.A_Tracksync	DB31, ...DBX31.6

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.coupling.in.CorrectionDone	LBP_Axis*.E_SyncRunCorrDOut	DB31, ...DBX99.2

15.5.14 示例

15.5.14.1 编程示例

包含一个引导轴的直接激活/关闭

创建一个包含跟随轴 X2 和引导轴 X1 的耦合模块并激活。耦合系数为 2。

```
CPON=(X2) CPLA[X2]=(X1) CPLNUM[X2,X1]=2
...
```

CPOF=(X2) ; 使用 CPOF 禁用耦合，并删除创建的耦合模块。

包含两个引导轴的直接激活/关闭

创建一个包含跟随轴 X2 和引导轴 X1 和 Z 的耦合模块并激活。引导轴 X1 的耦合系数为 2，引导轴 Z 的耦合系数为 3。

CPON=(X2) CPLA=(X1) CPLNUM[X2,X1]=2 CPLA=(Z) CPLNUM[X2,Z]=3
...
CPOF=(X2) ; 使用 CPOF 禁用耦合，并删除创建的耦合模块。

包含两个引导轴的选择性关闭

创建一个包含跟随轴 Y 和引导轴 X 和 Z 的耦合模块并激活。引导轴 X 的耦合系数为 2，引导轴 Z 的耦合系数为 1.2。

CPON=(X2) CPLA[X2]=(X1) CPLNUM[X2,X1]=2 CPLA[X2]=(Z) CPLNUM[X2,Z]=1.2
...
CPOF=(X2) CPLA[X1]=(Z) ; 使用 CPOF 禁用包含引导轴 Z 的耦合，包含引导轴 X1 的耦合保持有效。创建的耦合模块保持创建状态。

包含三个引导轴的选择性激活/关闭

创建一个包含跟随轴 X2 和引导轴 X1、Z 和 A 的耦合模块。

N10 CPDEF=(X2) CPLA[X2]=(X1) CPLA[X2]=(Z) CPLA[X2]=(A)
N20 CPON=(X2) ; 所有引导轴都生效，即全部根据耦合规律提供到轴 X2 的位置份额（耦合份额）。
N30 CPOF=(X2) ; 所有引导轴都禁用。
N40 CPLON[X2]=(X1) ; 引导轴 X1 将激活，只有它提供耦合份额。引导轴 Z 和 A 保持禁用。
N50 CPLON[X2]=(A) ; 引导轴 X1 保持生效，引导轴 A 将激活，X1 和 A 提供耦合份额（→ 选择性激活的作用是相加的，其他引导轴的状态保持不变）。
N60 CPLON[X2]=(Z) CPLOF[X2]=(A) ; 引导轴 Z 将激活，引导轴 A 将禁用。现在引导轴 X1 和 Z 生效。
N70 CPLOF[X2]=(X1) ; 引导轴 X1 将禁用。引导轴 Z 保持生效。

定义/删除耦合模块

用 CPDEF 创建一个包含跟随轴 X2 和引导轴 X1 和 Z 的耦合模块。耦合不会激活。跟随轴 X2 不遵循耦合规律！

```
CPDEF=(X2) CPLA[X2]=(X1) CPLNUM[X2,X1]=2 CPLA[X2]=(Z) CPLNUM[X2,Z]=3
...
```

可以用 CPON 激活，用 CPOF 禁用。

禁用与所有引导轴的耦合关系后，可以删除耦合对象。预留的内存将被释放：

```
CPDEL=(X2)
```

15.5.14.2 调整匹配循环

目标

机床坐标系中的联动应该可以使用现有的耦合指令 TRAILON 实现。为此，TRAILON 的匹配循环补充了耦合属性“参考坐标”(CPFRS)。

步骤

1. 将匹配循环 cycle700 从“CST”目录复制到“CMA”目录。
2. 为匹配循环 cycle700 补充以下条目：
CPFRS[_FA]="MCS"
3. 对循环中的更改进行注释（例如使用用户版本号和更改日期）。
4. 保存循环。


```
;*CHANGE :07.01.02.00 Mar 08, 2006
;SPATH=/_N_CST_DIR/_N_CYCLE700_SPF
;ANWENDER V1.1 Mar 22, 2006
;classic TRAILON(FA,LA,Factor)
PROC CYCLE700(AXIS _CPF=NO_AXIS, AXIS _CPL=NO_AXIS,REAL _CPLF=1) IPTRLOCK SBLOF DISPLOF ICYCOF
;*IF _CPLF==0 GOTOF _CPOF
CPLON[_CPF]=(_CPL) CPSETTYPE[_CPF]="TRAIL" CPLNUM[_CPF,_CPL]=(_CPLF) CPFRS[_CPF]="MCS"
RET
_CPOF:
IF ($P_TECCYCLE==TRUE)
  IF ($AA_CPSETTYPE[_CPF]<>"TRAIL") GOTOF _EXIT
ELSE
  IF ($PA_CPSETTYPE[_CPF]<>"TRAIL") GOTOF _EXIT
ENDIF
IF _CPL==NO_AXIS
  CPSETTYPE[_CPF]="TRAIL" CPOF=(_CPF)
ELSE
  CPSETTYPE[_CPF]="TRAIL" CPLOF[_CPF]=(_CPL)
ENDIF
_EXIT:RET
;* SysFehler 500918 因此注释掉 IF 程序段
```

图 15-14 调整后的 cycle700。更改用彩色条标记。

15.6 跟随轴的动态响应特性

15.6.1 参数设置的动态响应限制

跟随轴的动态响应通过以下机床数据值进行限制：

MD32000 \$MA_MAX_AX_VELO（最大轴速度）

MD32300 \$MA_MAX_AX_ACCEL（最大轴加速度）

15.6 跟随轴的动态响应特性

15.6.2 编程的动态响应限制

15.6.2.1 编程（VELOLIMA、ACCLIMA）

降低或提高动态响应限制

可以使用来自**零件程序**的语言指令来降低或提高通过 MD32000 和 MD32300 指定的跟随轴 (FA) 的动态响应限制：

指令	含义
VELOLIMA [FA]	降低或提高最大 轴速度
ACCLIMA [FA]	降低或提高最大 轴加速度

编程 VELOLIMA [FA] 和 ACCLIMA [FA] 时指定的值是**百分比值**。它定义应考虑的参数设置的动态响应限制的比例（MD32000 和 MD32300）：

值域	含义
$1 \leq \text{值} < 100$	导致动态响应限制 降低
$100 < \text{值} \leq 200$	导致动态响应限制 提高

跟随轴的速度和加速度的动态响应限制如下计算：

最大轴速度 = MD32000 \$MA_MAX_AX_VELO * VELOLIMA[FA]

最大轴加速度 = MD32300 \$MA_MAX_AX_ACCEL * ACCLIMA[FA]

说明

降低/提高影响轴的整个动态响应，即来自叠加和耦合的轴分量的总和。

在同步动作中编程

在**同步动作**中编程 VELOLIMA [FA] 和 ACCLIMA [FA] 的可能性取决于耦合类型：

耦合类型	零件程序	同步动作
切向跟随	x	
联动	x	x
引导值耦合	x	x
电子齿轮箱	x	

耦合类型	零件程序	同步动作
同步主轴	x	
通用耦合	x	x

跟随轴和引导轴之间的同步

由于设置的加速特性和设置的动态响应补偿，加速过程中跟随轴和引导轴之间的同步所需时间如下变化：

动态响应补偿	作用
动态响应降低	延长同步运行偏差。 从引导值到跟随值的监控可能在较长时间内超出允许范围。
动态响应提高	缩短同步运行偏差。 从引导值到跟随值的监控可能在较短时间内超出允许范围。

说明

用户必须重新建立加工和同步运行偏差之间的技术同步。

加速模式

只有 BRISKA（即跳跃式轴加速）可用于跟随轴。加速模式 SOFTA 和 DRIVEA 不适用于下述跟随轴。

此外，还可以将位置调节器配置为 PI 控制器。

 小心
应用错误
这种方法只能与伺服跟踪和控制技术专业知识和使用。

更多信息

- CNC 调试手册：NC、PLC、驱动
- 速度、设定值/实际值系统、闭环控制 (页 141)

上电

上电时，VELOLIMA 和 ACCLIMA 的值初始化为 100%。

15.6 跟随轴的动态响应特性

运行方式切换

AUTO → JOG 运行方式切换时，动态响应补偿仍然有效。

复位

复位后动态响应补偿（VELOLIMA 和 ACCLIMA）的有效性取决于通道专用机床数据中的设置：

MD22410 \$MC_F_VALUES_ACTIVE_AFTER_RESET（F 功能复位后保持生效）

值	含义
0	VELOLIMA[FA] 和 ACCLIMA[FA] 的值在复位后设置为 100%。
1	最后编程的 VELOLIMA[FA] 和 ACCLIMA[FA] 的值在复位后也有效。

此特性也适用于由静态同步动作设置的动态响应补偿。如果在 MD22410 = 0 时仍不是这种情况，则必须通过 IDS 同步动作再次或连续写入动态响应补偿。

更多信息
功能手册之同步动作

15.6.2.2 示例

电子齿轮箱

轴 4 通过 EG 耦合与 X 耦合。将跟随轴的加速能力限制为最大加速度的 70%。将允许的最大速度限制为最大速度的 50%。激活后，再将最大允许速度重新设为 100%。

```
...
N120 ACCLIMA[AX4]=70
N130 VELOLIMA[AX4]=50 ; 降低的速度
N150 EGON(AX4,"FINE",X,1,2)
N200 VELOLIMA[AX4]=100 ; 全速
...
```

引导值耦合

轴 4 通过引导值耦合与 X 耦合。将跟随轴的加速能力限制为最大加速度的 80%。

```
...
```

N120	ACCLIMA[AX4]=80	; 80%
N130	LEADON(AX4,X,2)	; 激活耦合
...		

使用同步动作的引导值耦合

轴 4 通过引导值耦合与 X 轴耦合。通过静态同步动作 2，从位置 100 起将加速特性限制为 80%。

...
N120 IDS=2 WHENEVER \$AA_IM[AX4] > 100 DO ACCLIMA[AX4]=80
N130 LEADON(AX4,X,2)
...

15.6.2.3 系统变量

对于几何轴、通道轴、机床轴和主轴等轴类型，在零件程序和同步动作中可使用以下可读系统变量：

标识符	数据类型	含义	单位
在预处理中			
\$PA_ACCLIMA[n]	REAL	用 ACCLIMA[Ax] 设置的加速度修调	%
\$PA_VELOLIMA[n]	REAL	用 VELOLIMA[Ax] 设置的速度修调	%
在主运行中			
\$AA_ACCLIMA[n]	REAL	用 ACCLIMA[Ax] 设置的加速度修调	%
\$AA_VELOLIMA[n]	REAL	用 VELOLIMA[Ax] 设置的速度修调	%

说明

读取主运行变量会隐性触发预处理停止。

15.7 一般边界条件

说明

驱动优化

在一台 SINAMICS S120 驱动设备上，最多可以同时优化或测量 3 台驱动（转速控制器优化/函数发生器）。因此，对于具有 3 个以上同时耦合的驱动的耦合，强烈建议将其分配给多台驱动设备。

说明

耦合生效时的程序段搜索

我们建议在耦合生效时仅采用搜索类型 5 “通过程序测试进行程序段搜索”(SERUPRO) 来搜索程序段。

手动运行

16.1 概述

应用

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：

示例：

- 调试机床
尤其是在调试新加工程序时，需手动运行机床轴。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。
- 程序中断后退刀
掉电或复位操作造成程序中断后，操作人员必须手动将刀具从当前加工位置退出来。此时须继续采用加工中生效的转换和坐标系。

运行方式 JOG

在 JOG 运行方式下手动运行轴。

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

`<ModeGroup>.basic.in.jogModeActive` (BAG1: JOG 运行方式)

更多信息

功能手册之基本功能；运行方式组、通道、程序运行、复位特性

显示

JOG 运行方式激活时，用户界面中会显示 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。

在 BCS 或 WCS 中手动运行

操作人员可在基本坐标系 (BCS) 或工件坐标系 (WCS) 中运行轴。

- 在 BCS 中手动运行
每根轴均可手动运行。
- 在 WCS 中手动运行
只有定义为几何轴的轴可在 WCS 中手动运行（通道专用）。此时选中的坐标平移或旋转在手动运行中保持生效。

手动运行多根轴

在 JOG 运行方式下可同时手动运行多根轴。不过这些轴之间不进行插补。

通过 PLC 接口控制

通过 PLC 接口控制手动运行。

参见章节“通过 PLC 接口控制 (页 746)”。

此时机床轴（轴专用）、几何轴（通道专用）和定向轴（通道专用）分别有专用的 PLC 接口。

常规参数设置

手动运行轴时有对所有 JOG 运动都有效（与运行方式和所选 JOG 功能无关）的可编程属性。
参见“常规参数设置 (页 747)”一章中的描述。

运行方式

在 JOG 模式下可采取以下手动运行方式：

- 连续运行（通过机床控制面板上的运行键），分点按模式和长按模式
参见章节“连续手动运行 (页 753)”。
- 增量运行（通过机床控制面板上的运行键），分点按模式和长按模式
参见章节“增量手动运行 (页 757)”。
- 通过电子手轮运行轴
参见章节“通过手轮进行手动运行 (页 763)”。

手动运行主轴

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。

手动运行主轴时须注意的特点请参考“手动运行主轴 (页 781)”一章。

手动运行几何轴

手动运行几何轴用于坐标转换和框架必须生效的运行。几何轴采用最后生效的坐标系运行。

手动运行几何轴（和定向轴）时须注意的特点请参考“手动运行几何轴/定向轴 (页 783)”一章。

其它功能

除不同的运行方式外，在 JOG 模式下还可选择以下功能：

- REF（回参考点运行）
- REPOS（重新定位）
- 逼近设置的固定点
参见章节“手动运行至固定点 (页 787)”。
- 逼近可设定位置
参见章节“手动运行至位置 (页 795)”。
- 沿着圆弧运行几何轴
参见章节“圆弧手动运行 (页 801)”。
- 手动退刀（掉电或复位操作造成程序中断后使刀具沿其中心轴方向退回）
参见章节“沿刀具中心轴的退刀（手动退刀） (页 812)”。

在 AUTO 模式下使用手轮

手轮也可在 AUTO 模式下使用：

- 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 (页 823)
- 轮廓手轮/通过手轮设定行程（选件） (页 832)
- 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） (页 823)

监控

手动运行时，控制系统内部监控的限制生效。

参见章节“监控 (页 843)”。

从 JOG 切换至 AUTO/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。

更多信息

功能手册之基本功能：运行方式组、通道、程序运行、复位特性

调试

有关电子手轮调试的详细信息请见“调试：手轮 (页 844)”一章。

PLC 信号

NC → PLC

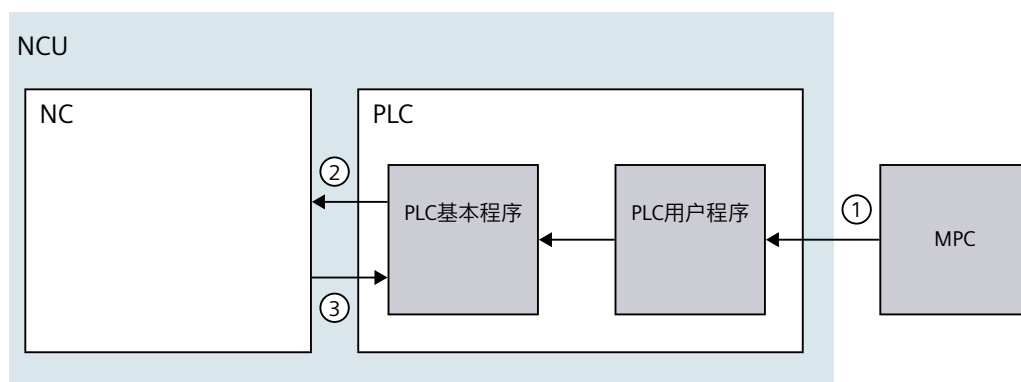
Basic Program Plus	Basic Program	
<ModeGroup>.basic.in.jogModeActive	LBP_ModeGroup.E_JOG	DB11.DBX6.2

16.2 通过 PLC 接口控制

在通过机床控制面板 (MCP) 激活 JOG 功能时，NC 是通过 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“连续运行”。
MCP 的输入信号通过 PLC 基本程序在循环中传送到 PLC 用户程序的 MCP 输入接口中。这可以是如数据块等。
- 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专用 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轴专用 NC/PLC 接口。

更多信息

关于如何配置机床控制面板以及如何将机床控制面板加入 PLC 用户程序的详细信息请见 PLC 功能手册。

参见

进给率 (页 201)

16.3 常规参数设置

进给特性

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）手动运行当前主主轴。
此时通过以下设定数据设置：

16.3 常规参数设置

SD41100 \$SN_JOG_REV_IS_ACTIVE (JOG: 旋转进给率/线性进给率)

位	值	含义
0	0	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) 若几何轴或定向轴上有含旋转的有效框架命令, 则该几何轴或定向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)
	1	在 JOG 运行中, 始终根据主轴以旋转进给率运行。
1	0	即使在急速运行模式中, 也以旋转进给率运行轴/主轴、几何轴或者定位轴 (选择参见位 0)。
	1	在急速运行模式中, 不以旋转进给率运行轴/主轴、几何轴或者定位轴。
2	0	即使在 JOG 运行模式中通过手轮进行手动运行时, 也以旋转进给率运行轴/主轴、几何轴或者定位轴 (选择参见位 0)。
	1	在 JOG 运行模式中通过手轮进行手动运行时, 不以旋转进给率运行轴/主轴、几何轴或者定位轴。
3	0	即使在 DRF 运行模式中通过手轮进行手动运行时, 也以旋转进给率运行轴/主轴 (选择参见位 0)。
	1	在 DR 运行模式中通过手轮进行手动运行时, 不以旋转进给率运行轴/主轴、几何轴或者定位轴。

说明

若未定义主主轴, 同时尝试在手动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。

速度

确定手动运行速度的系统数据取决于生效的进给特性。

直线进给率 (G94) 生效

- 机床轴

速度通过以下设定数据确定：

- 线性轴：

SD41110 \$SN_JOG_SET_VELO (JOG 轴速度)

- 回转轴：

SD41130 \$SN_JOG_ROT_AX_SET_VELO (回转轴的 JOG 速度)

若 SD41110/SD41130 中输入了值“0”，以下轴专用机床数据生效：

MD32020 \$MA_JOG_VELO (基准 JOG 轴速度)

进给速度的默认单位为 mm/min，而回转轴为 rev/min。

- 主轴

主轴的手动运行速度通过以下设定数据确定：

SD41200 \$SN_JOG_SPIND_SET_VELO (JOG 运行模式下主轴的转速)

若 SD41200 中输入了值“0”，以下轴专用机床数据生效：

MD32020 \$ _MA_JOG_VELO (基准 JOG 轴速度)

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。

更多信息

主轴 (页 395)

- 几何轴

几何轴的手动运行速度通过以下通道专用机床数据确定：

MD21165 \$MC_JOG_VELO_GEO (几何轴的基准 JOG 速度)

- 定向轴

定向轴的手动运行速度通过以下通道专用机床数据确定：

MD21155 \$MC_JOG_VELO_ORI (定向轴的基准 JOG 速度)

旋转进给率 (G95) 生效

进给率值 (单位：毫米/转) 通过以下设定数据确定：

SD41120 \$SN_JOG_REV_SET_VELO (JOG 旋转进给率)

若 SD41120 中输入了值“0”，以下轴专用机床数据生效：

MD32050 \$MD_JOG_REV_VELO (JOG 旋转进给率)

快速移动叠加

若在按下运行键/手轮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：

- 机床轴
 - MD32010 \$MA_JOG_VELO_RAPID（基准 JOG 快速移动速度）
 - MD32040 \$MA_JOG_REV_VELO_RAPID（JOG 旋转进给率叠加快速移动速度）
- 几何轴
 - MD21160 \$MC_JOG_VELO_RAPID_GEO（几何轴的 JOG 快速移动速度）
- 定向轴
 - MD21150 \$MC_JOG_VELO_RAPID_ORI（定向轴的 JOG 快速移动速度）

进给倍率

若下列 NC/PLC 接口信号置位，则可通过进给倍率开关对手动运行速度进行调整：

`<Axis>.basic.out.enableOvr`（轴专用进给倍率生效）

进给倍率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 `<Axis>.basic.out.enableOvr`（轴专用进给倍率生效）对开关位置 0 % 无作用。

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

更多信息

进给率 (页 201)

主轴倍率

主轴的手动运行速度可通过主轴倍率开关进行调整。

更多信息

主轴 (页 395)

加速度和加加速度

在 JOG 模式中，轴专用的加速度限值和急动度限值也有效。

此外，可以针对具体通道对用于几何轴及定向轴手动运行的加速度和急动度加以限制。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）。

JOG 运动中的最大轴加速度

轴手动运行时允许的最大加速度在机床数据

MD32300 \$MA_MAX_AX_ACCEL [0] (动态模式 DYNORM 中轨迹运动中轴专用的最大加速度) 中定义。

说明

对于 JOG 模式，始终只有动态模式 DYNORM 有效。

最大轴急动度的初始设置

带急动度限制的加速模式 (软加速模式) 可设为 JOG 模式中轴专用的初始设置:

MD32420 \$MA_JOG_AND_POS_JERK_ENABLE == TRUE

轴专有的最大急动度

在急动度限制激活的 JOG 运动中，下列机床数据的值作为轴专有的最大急动度生效:

MD32430 \$MA_JOG_AND_POS_MAX_JERK (轴专用最大急动度)

说明

只有在为 JOG 模式中的待运行机床轴激活了急动度限制时，MD32430 \$MA_JOG_AND_POS_MAX_JERK 才会生效:

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

这也可以通过在零件程序中编程 SOFTA (<Achse1>, <Achse2>, ...) 实现。

主轴手动运行时的最大加速度

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。

更多信息

主轴 (页 395)

几何轴手动运行时的最大加速度

手动运行几何轴时，可通过以下机床数据来预设最大加速度:

MD21166 \$MC_JOG_ACCEL_GEO [<几何轴>]

其中 <几何轴> = 0, 1, 2

说明

MD21166 \$MC_JOG_ACCEL_GEO [<几何轴>] 对 MD32300 \$MA_MAX_AX_ACCEL 没有直接的限制。

16.3 常规参数设置

说明

当转换激活时，MD32300 \$MA_MAX_AX_ACCEL 决定了最大轴加速度。

几何轴手动运行时的最大急动度

在加速模式 SOFT（软加速模式）中手动运行几何轴时，可通过以下机床数据设置最大加速度：

MD21168 \$MC_JOG_JERK_GEO [<几何轴>]

其中 <几何轴> = 0, 1, 2

说明

只有在为 JOG 模式中的机床轴激活了最大轴急动度时，MD21168 \$MC_JOG_JERK_GEO 才会生效：

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

当 MD21168 = 0 时，MD32430 \$MA_JOG_AND_POS_MAX_JERK 中的轴专用最大急动度会代替通道专用的最大急动度生效。

定向轴手动运行时的最大急动度

手动运行定向轴时，可通过以下机床数据设置最大急动度：

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手动运行定向轴激活通道专用的最大急动度：

MD21159 \$MC_JOG_JERK_ORI_ENABLE == TRUE

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.enableOvr	LBP_Axis.A_ORActive	DB31, ...DBX1.7

参见

手动运行几何轴/定向轴 (页 783)

手动运行主轴 (页 781)

16.4 连续手动运行

16.4.1 功能

连续手动运行时，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。

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。

16.4 连续手动运行

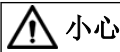
终止运行

操作人员可使用下列方法通过机床控制面板 (MCP) 的操作单元终止运行：

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

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

接口信号

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：

- 机床轴：
 - `<Axis>.basic.out.contTravers`（请求机床功能“连续手动运行”）
- 几何轴：
 - `<Chan>.geoAxis1.out.contManualTravers`（几何轴 1：请求机床功能“连续手动运行”）
- 定向轴：
 - `<Chan>.orientationAxis1.out.contManualTravers`（定向轴 1：请求机床功能“连续手动运行”）

反馈信息

连续运行生效后，向 PLC 发送反馈信息：

- 机床轴：
 - `<Axis>.basic.in.contTraversActive`（生效的机床功能“连续手动运行”）
- 几何轴：
 - `<Chan>.geoAxis1.in.contManualTraversActive`（几何轴 1：生效的机床功能“连续手动运行”）

- 定向轴：
 - <Chan>.orientationAxis1.in.contManualTraversActive（定向轴 1：生效的机床功能“连续手动运行”）

运行指令 +/-

一旦存在轴运行请求，系统便会根据运动方向将接口信号“运行指令-”或“运行指令+”输出至 PLC。

- 机床轴：
 - <Axis>.basic.in.traversMinusCommandActive（运行指令“-”）或
 - <Axis>.basic.in.traversPlusCommandActive（运行指令“+”）
- 几何轴 1：
 - <Chan>.geoAxis1.in.traversMinusCommand（几何轴 1：运行指令“-”）或
 - <Chan>.geoAxis1.in.traversPlusCommand（几何轴 1：运行指令“+”）
- 定向轴 1：
 - <Chan>.orientationAxis1.in.traversMinusCommand（定向轴 1：运行指令“-”）或
 - <Chan>.orientationAxis1.in.traversPlusCommand（定向轴 1：运行指令“+”）

说明

如果缺少运行轴所需的“伺服使能”和“脉冲使能”，系统会撤销在 JOG 模式下设置的运行指令。因此，使能轴后必须重新设置运行指令。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.contTravers	LBP_Axis*.A_ContManTravel	DB31, ...DBX5.6
<Chan>.geoAxis1.out.contManualTravers	LBP_Chan*.ContManTravel	DB21, ...DBX13.6
<Chan>.orientationAxis1.out.contManualTravers	-	DB21, ...DBX321.6

16.4 连续手动运行

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.contTraversActive	LBP_Axis*.E_ContManTravelAct	DB31, ...DBX65.6
<Chan>.geoAxis1.in.contManualTraversActive	-	DB21, ...DBX41.6
<Chan>.geoAxis1.in.traversMinusCommand	LBP_Chan*.TCMinus	DB21, ...DBX40.6
<Chan>.geoAxis1.in.traversPlusCommand	LBP_Chan*.TCPlus	DB21, ...DBX40.7
<Axis>.basic.in.traversMinusCommandActive	LBP_Axis*.E_TCMinus	DB31, ...DBX64.6
<Axis>.basic.in.traversPlusCommandActive	LBP_Axis*.E_TCPlus	DB31, ...DBX64.7
<Chan>.orientationAxis1.in.traversMinusCommand	LBP_Chan*.TCMinus	DB21, ...DBX332.6
<Chan>.orientationAxis1.in.traversPlusCommand	LBP_Chan*.TCPlus	DB21, ...DBX332.7
<Chan>.orientationAxis1.in.contManualTraversActive	-	DB21, ...DBX333.6

16.4.2 参数设置

点按模式或长按模式

连续运行中的点按/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）
长按模式为缺省设置。

16.4.3 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。

16.5 增量手动运行

16.5.1 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)。

断续运行中的点按模式

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。

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
- `<Axis>.basic.out.resetMovement`（删除剩余行程）

断续运行中的长按模式

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（参见章节“连续手动运行 (页 753)”）。

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
- `<Axis>.basic.out.resetMovement`（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。

接口信号

选择

在 JOG 运行方式下的“断续运行”子方式必须通过 PLC 接口激活：

- 机床轴：
 - <Axis>.basic.out.inc1（请求机床功能“INC1”）
 - <Axis>.basic.out.inc10（请求机床功能“INC10”）
 - <Axis>.basic.out.inc100（请求机床功能“INC100”）
 - <Axis>.basic.out.inc1000（请求机床功能“INC1000”）
 - <Axis>.basic.out.inc10000（请求机床功能“INC100000”）
 - <Axis>.basic.out.incVar（请求机床功能“INCvar”）
- 几何轴：
 - <Chan>.geoAxis1.out.inc1（请求机床功能“INC1”）
 - <Chan>.geoAxis1.out.inc10（请求机床功能“INC10”）
 - <Chan>.geoAxis1.out.inc100（请求机床功能“INC100”）
 - <Chan>.geoAxis1.out.inc1000（请求机床功能“INC1000”）
 - <Chan>.geoAxis1.out.inc10000（请求机床功能“INC10000”）
 - <Chan>.geoAxis1.out.incVar（请求机床功能“INCvar”）

- 定向轴：
 - <Chan>.orientationAxis1.out.inc1 (定向轴 1: 请求机床功能“INC1”)
 - <Chan>.orientationAxis1.out.inc10 (定向轴 1: 请求机床功能“INC10”)
 - <Chan>.orientationAxis1.out.inc100 (定向轴 1: 请求机床功能“INC100”)
 - <Chan>.orientationAxis1.out.inc1000 (定向轴 1: 请求机床功能“INC1000”)
 - <Chan>.orientationAxis1.out.inc10000 (定向轴 1: 请求机床功能“INC10000”)
 - <Chan>.orientationAxis1.out.incVar (定向轴 1: 请求机床功能“INCvar”)

反馈信息

断续运行生效后，向 PLC 发送反馈信息：

- 机床轴：
 - <Axis>.basic.in.inc1Active (生效的机床功能“INC1”)
 - <Axis>.basic.in.inc10Active (生效的机床功能“INC10”)
 - <Axis>.basic.in.inc100Active (生效的机床功能“INC100”)
 - <Axis>.basic.in.inc1000Active (生效的机床功能“INC1000”)
 - <Axis>.basic.in.inc10000Active (生效的机床功能“INC10000”)
 - <Axis>.basic.in.incVarActive (生效的机床功能“INCvar”)
- 几何轴：
 - <Chan>.geoAxis1.in.inc1Active (几何轴 1: 生效的机床功能“INC1”)
 - <Chan>.geoAxis1.in.inc10Active (几何轴 1: 生效的机床功能“INC10”)
 - <Chan>.geoAxis1.in.inc100Active (几何轴 1: 生效的机床功能“INC100”)
 - <Chan>.geoAxis1.in.inc1000Active (几何轴 1: 生效的机床功能“INC1000”)
 - <Chan>.geoAxis1.in.inc10000Active (几何轴 1: 生效的机床功能“INC10000”)
 - <Axis>.basic.out.incVar (几何轴 1: 生效的机床功能“INCvar”)

- 定向轴：
 - <Chan>.orientationAxis1.in.inc1Active (定向轴 1: 生效的机床功能“INC1”)
 - <Chan>.orientationAxis1.in.inc10Active (定向轴 1: 生效的机床功能“INC10”)
 - <Chan>.orientationAxis1.in.inc100Active (定向轴 1: 生效的机床功能“INC100”)
 - <Chan>.orientationAxis1.in.inc1000Active (定向轴 1: 生效的机床功能“INC1000”)
 - <Chan>.orientationAxis1.in.inc10000Active (定向轴 1: 生效的机床功能“INC10000”)
 - <Chan>.orientationAxis1.in.incVarActive (定向轴 1: 生效的机床功能“INCvar”)

运行指令 +/-

一旦存在轴运行请求，系统便会根据运动方向将接口信号“运行指令-”或“运行指令+”输出至 PLC。

- 机床轴：
 - <Axis>.basic.in.traversMinusCommandActive (运行指令“-”) 或
 - <Axis>.basic.in.traversPlusCommandActive (运行指令“+”)
- 几何轴 1：
 - <Chan>.geoAxis1.in.traversMinusCommand (几何轴 1: 运行指令“-”) 或
 - <Chan>.geoAxis1.in.traversPlusCommand (几何轴 1: 运行指令“+”)
- 定向轴 1：
 - <Chan>.orientationAxis1.in.traversMinusCommand (定向轴 1: 运行指令“-”) 或
 - <Chan>.orientationAxis1.in.traversPlusCommand (定向轴 1: 运行指令“+”)

说明

如果缺少运行轴所需的“伺服使能”和“脉冲使能”，系统会撤销在 JOG 模式下设置的运行指令。因此，使能轴后必须重新设置运行指令。

PLC 信号

PLC → NC

Basic Program Plus	Basic Proagram	
<Axis>.basic.out.resetMovement	LBP_Axis*.A_DeIDTGSReset	DB31, ...DBX2.2
<Axis>.basic.out.inc1	LBP_Axis*.A_INC1	DB31, ...DBX5.0
<Axis>.basic.out.inc10	LBP_Axis*.A_INC10	DB31, ...DBX5.1
<Axis>.basic.out.inc100	LBP_Axis*.A_INC100	DB31, ...DBX5.2
<Axis>.basic.out.inc1000	LBP_Axis*.A_INC1000	DB31, ...DBX5.3
<Axis>.basic.out.inc10000	LBP_Axis*.A_INC10000	DB31, ...DBX5.4
<Axis>.basic.out.incVar	LBP_Axis*.A_INCVar	DB31, ...DBX5.5
<Chan>.geoAxis1.out.inc1	LBP_Axis*.INC1	DB31, ...DBX5.0
<Chan>.geoAxis1.out.inc10	LBP_Axis*.INC10	DB31, ...DBX5.1
<Chan>.geoAxis1.out.inc100	LBP_Axis*.INC100	DB31, ...DBX5.2
<Chan>.geoAxis1.out.inc1000	LBP_Axis*.INC1000	DB31, ...DBX5.3
<Chan>.geoAxis1.out.inc10000	LBP_Axis*.INC10000	DB31, ...DBX5.4
<Chan>.geoAxis1.out.incVar	LBP_Axis*.INCVar	DB31, ...DBX5.5
<Chan>.geoAxis1.out.contManualTravers	LBP_Chan*.ContManTravel	DB21, ...DBX13.6
<Chan>.orientationAxis1.out.inc1	LBP_Chan*.INC1	DB21, ...DBX321.0
<Chan>.orientationAxis1.out.inc10	LBP_Chan*.INC10	DB21, ...DBX321.1
<Chan>.orientationAxis1.out.inc100	LBP_Chan*.INC100	DB21, ...DBX321.2
<Chan>.orientationAxis1.out.inc1000	LBP_Chan*.INC1000	DB21, ...DBX321.3
<Chan>.orientationAxis1.out.inc10000	LBP_Chan*.INC10000	DB21, ...DBX321.4
<Chan>.orientationAxis1.out.incVar	LBP_Chan*.INCVar	DB21, ...DBX321.5
<Chan>.orientationAxis1.out.contManualTrave rs	-	DB21, ...DBX321.6

NC → PLC

Basic Program Plus	Basic Proagram	
<Chan>.geoAxis1.in.inc1Active	LBP_Chan*.INC1	DB21, ...DBX41.0
<Chan>.geoAxis1.in.inc10Active	LBP_Chan*.INC10	DB21, ...DBX41.1
<Chan>.geoAxis1.in.inc100Active	LBP_Chan*.INC100	DB21, ...DBX41.2
<Chan>.geoAxis1.in.inc1000Active	LBP_Chan*.INC1000	DB21, ...DBX41.3

16.5 增量手动运行

Basic Program Plus	Basic Program	
<Chan>.geoAxis1.in.inc10000Active	LBP_Chan*.INC10000	DB21, ...DBX41.4
<Chan>.geoAxis1.in.incVarActive	LBP_Chan*.INCVar	DB21, ...DBX41.5
<Axis>.basic.in.inc1Active	LBP_Axis*.E_INC1	DB31, ...DBX65.0
<Axis>.basic.in.inc10Active	LBP_Axis*.E_INC10	DB31, ...DBX65.1
<Axis>.basic.in.inc100Active	LBP_Axis*.E_INC100	DB31, ...DBX65.2
<Axis>.basic.in.inc1000Active	LBP_Axis*.E_INC1000	DB31, ...DBX65.3
<Axis>.basic.in.inc10000Active	LBP_Axis*.E_INC10000	DB31, ...DBX65.4
<Axis>.basic.in.incVarActive	LBP_Axis*.E_INCVar	DB31, ...DBX65.5
<Axis>.basic.in.contTraversActive	LBP_Axis*.E_ContManTravelAct	DB31, ...DBX65.6
<Chan>.geoAxis1.in.contManualTraversActive	-	DB21, ...DBX41.6
<Chan>.geoAxis1.in.traversMinusCommand	LBP_Axis*.TCMinus	DB21, ...DBX40.6
<Chan>.geoAxis1.in.traversPlusCommand	LBP_Axis*.TCPlus	DB21, ...DBX40.7
<Axis>.basic.in.traversMinusCommandActive	LBP_Axis*.E_TCMinus	DB31, ...DBX64.6
<Axis>.basic.in.traversPlusCommandActive	LBP_Axis*.E_TCPlus	DB31, ...DBX64.7
<Chan>.orientationAxis1.in.traversMinusCommand	LBP_Chan*.TCMinus	DB21, ...DBX332.6
<Chan>.orientationAxis1.in.traversPlusCommand	LBP_Chan*.TCPlus	DB21, ...DBX332.7
<Chan>.orientationAxis1.in.inc1Active	LBP_Chan*.INC1	DB21, ...DBX333.0
<Chan>.orientationAxis1.in.inc10Active	LBP_Chan*.INC10	DB21, ...DBX333.1
<Chan>.orientationAxis1.in.inc100Active	LBP_Chan*.INC100	DB21, ...DBX333.2
<Chan>.orientationAxis1.in.inc1000Active	LBP_Chan*.INC1000	DB21, ...DBX333.3
<Chan>.orientationAxis1.in.inc10000Active	LBP_Chan*.INC10000	DB21, ...DBX333.4
<Chan>.orientationAxis1.in.incVarActive	LBP_Chan*.INCVar	DB21, ...DBX333.5
<Chan>.orientationAxis1.in.contManualTraversActive	-	DB21, ...DBX333.6

16.5.2 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。

点按模式或长按模式

断续运行中的点按/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 长按模式)

长按模式为缺省设置。

16.5.3 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。

16.6 通过手轮进行手动运行

16.6.1 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。

16.6 通过手轮进行手动运行

调试

有关电子手轮调试的详细信息请见“调试：手轮 (页 844)”一章。

选择

JOG 运行方式必须生效。操作人员还须设置在通过手轮进行手动运行时生效的步数 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。

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，则无法再用手轮进行手动运行。

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“参数设置 (页 772)”。
- 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“参数设置 (页 772)”。

手轮连接

最多可同时连接 2 个手轮。因此同时最多可运行 2 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。

参见章节“参数设置 (页 772)”。

手轮指定

可通过 PLC 用户接口或通过用户界面 (HMI) 指定通过旋转手轮运行的轴。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：
 - `<Axis>.basic.out.handwheelSelection` (激活手轮)
- 几何轴：
 - `<Chan>.geoAxis1.out.handwheelSelection` (几何轴 1: 激活手轮)
- 定向轴：
 - `<Chan>.orientationAxis1.out.handwheelSelection` (定向轴 1: 激活手轮)

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 和 2 的接口包括以下信息：

- 分配给手轮的**轴编号**（若选择手轮时选择了机床轴）：
 - DB10 DBX100.0 - 4 (手轮 1 轴编号)
 - DB10 DBX101.0 - 4 (手轮 2 轴编号)
- 分配给手轮的**通道编号**（若选择手轮时选择了几何轴）：
 - DB10 DBX97.0 - 3 (手轮 1 几何轴通道编号)
 - DB10 DBX98.0 - 3 (手轮 2 几何轴通道编号)
- 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)
- 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)

16.6 通过手轮进行手动运行**使能**

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：
 - `<Axis>.basic.out.handwheelSelection`（激活手轮）
- 几何轴：
 - `<Chan>.geoAxis1.out.handwheelSelection`（几何轴 1：激活手轮）

定向轴只能通过对应的 PLC 用户接口信号激活：

- `<Chan>.orientationAxis1.out.handwheelSelection`（定向轴 1：激活手轮）

说明

定向轴只能通过对应的 PLC 用户接口信号激活：

- `<Chan>.orientationAxis1.out.handwheelSelection`（定向轴 1：激活手轮）
-

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：
 - `<Axis>.basic.in.traversMinusRequestActive`（运行请求“-”）或
 - `<Axis>.basic.in.traversPlusRequestActive`（运行请求“+”）
- 几何轴 1：
 - `<Chan>.geoAxis1.in.traversMinusRequest`（几何轴 1：运行请求“-”）或
 - `<Chan>.geoAxis1.in.traversPlusRequest`（几何轴 1：运行请求“+”）
- 定向轴 1：
 - `<Chan>.orientationAxis1.in.traversMinusRequest`（定向轴 1：运行请求“-”）或
 - `<Chan>.orientationAxis1.in.traversPlusRequest`（定向轴 1：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 779)”一章。

运行指令

根据机床数据 MD17900 \$MN_VDI_FUNCTION_MASK 中的设置（参见章节“参数设置（页 772）”），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：
 - <Axis>.basic.in.traversMinusCommandActive（运行指令“-”）或
 - <Axis>.basic.in.traversPlusCommandActive（运行指令“+”）
- 几何轴 1：
 - <Chan>.geoAxis1.in.traversMinusCommand（几何轴 1：运行指令“-”）或
 - <Chan>.geoAxis1.in.traversPlusCommand（几何轴 1：运行指令“+”）
- 定向轴 1：
 - <Chan>.orientationAxis1.in.traversMinusCommand（定向轴 1：运行指令“-”）或
 - <Chan>.orientationAxis1.in.traversPlusCommand（定向轴 1：运行指令“+”）

说明

如果缺少运行轴所需的“伺服使能”和“脉冲使能”，系统会撤销在 JOG 模式下设置的运行指令。因此，使能轴后必须重新设置运行指令。

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：
 - <Chan>.basic.out.disableNcStart（取反手轮旋转方向）
- 几何轴：
 - <Chan>.geoAxis1.out.handwheelInvertDir（几何轴 1：取反手轮旋转方向）
- 定向轴：
 - <Chan>.orientationAxis1.out.handwheelInvertDir（定向轴 1：取反手轮旋转方向）
- 轮廓手轮：
 - <Chan>.basic.out.contourHandwheelInvertDir（轮廓手轮：取反手轮旋转方向）

16.6 通过手轮进行手动运行

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：

– <Axis>.basic.in.handwheelDirInverted（手轮旋转方向取反生效）

• 几何轴：

– <Chan>.geoAxis1.in.handwheelDirInverted（几何轴 1：手轮旋转方向取反生效）

• 定向轴：

– <Chan>.orientationAxis1.in.handwheelDirInverted（定向轴 1：手轮旋转方向取反生效）

• 轮廓手轮：

– <Chan>.basic.in.contourHandwheelDirInverted（轮廓手轮：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。

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	
<Chan>.basic.out.reset（NC 复位）	几何轴/机床轴	终止	
<Chan>.geoAxis1.out.handwheelSelection（几何轴 1：激活手轮）	几何轴	1 → 0	终止
<Chan>.basic.out.deleteDistanceToGo（删除剩余行程）	几何轴/机床轴	0 → 1	终止
<Axis>.basic.out.resetMovement（删除剩余行程/主轴复位）	几何轴/机床轴	0 → 1	终止
<Axis>.basic.out.handwheelSelection（激活手轮）	机床轴	1 → 0	终止

NC/PLC 接口信号	有效范围	结果	
<Axis>.basic.out.disable (进给轴/主轴禁用)	几何轴/机床轴	0 → 1	终止
<Axis>.basic.out.enc1Activation (位置测量系统 1)	几何轴/机床轴	0 → 1	终止
<Axis>.basic.out.enc2Activation (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）对通过手轮进行的手动运行的作用（终止或中断运行）可设置（参见章节“参数设置 (页 772)”）。

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。

参见

主轴 (页 395)

PLC 信号

PLC → NC

Basic Program Plus	Basic Proogram	
<Axis>.basic.out.disable	LBP_Axis*.A_ASpDisable	DB31, ...DBX1.3
<Axis>.basic.out.enc1Activation	LBP_Axis*.A_PosMeas1	DB31, ...DBX1.5
<Axis>.basic.out.enc2Activation	LBP_Axis*.A_PosMeas2	DB31, ...DBX1.6
<Axis>.basic.out.resetMovement	LBP_Axis*.A_DeIDTGSpReset	DB31, ...DBX2.2
<Axis>.basic.out.handwheelSelection	LBP_Axis*.A_HW1..3	DB31, ...DBX4.0..2
<Chan>.basic.out.deleteDistanceToGo	LBP_Chan*.A_DeleteDTG	DB21, ...DBX6.2
<Chan>.basic.out.disableNcStart	LBP_Axis*.A_NCStartDisabl	DB31, ...DBX7.0
<Chan>.basic.out.reset	LBP_Chan*.A_Reset	DB21, ...DBX7.7
<Chan>.geoAxis1.out.handwheelSelection	LBP_Chan*.HW1	DB21, ...DBX12.0..2, DB21, ...DBX16.0..2, DB21, ...DBX20.0..2.5

16.6 通过手轮进行手动运行

Basic Program Plus	Basic Program	
<Chan>.geoAxis1.out.handwheelInvertDir	LBP_Chan*.InvHWDirotOfRot	DB21, ...DBX15.0, DB21, ...DBX19.0 DB21, ...DBX23.0
<Chan>.basic.out.contourHandwheelInvertDir	LBP_Chan*.A_InvCHWDirotOfRot	DB21, ...DBX31.5
<Chan>.orientationAxis1.out.handwheelSelect ion	LBP_Chan*.HW1	DB21, ...DBX320.0..2, DB21, ...DBX324.0..2, DB21, ...DBX328.0..2
<Chan>.orientationAxis1.out.handwheelInvert Dir	LBP_Chan*.InvHWDirotOfRot	DB21, ...DBX323.0 ..2, DB21, ...DBX327.0..2, DB21, ...DBX331.0..2
<Chan>.orientationAxis1.in.handwheelDirInve rted	LBP_Chan*.InvHWDirotOfRot	DB21, ...DBX335.0 , DB21, ...DBX339.0 , DB21, ...DBX343.0

NC → PLC

Basic Program Plus	Basic Program	
<Chan>.basic.in.contourHandwheelDirInverte d	LBP_Chan*.E_InvCHWDirotOfRot	DB21, ...DBX39.5
<Axis>.basic.in.traversMinusRequestActive	LBP_Axis*.E_TReqMinus	DB31, ...DBX64.4
<Axis>.basic.in.traversPlusRequestActive	LBP_Axis*.E_TReqPlus	DB31, ...DBX64.5
<Axis>.basic.in.traversMinusCommandActive	LBP_Axis*.E_TCMinus	DB31, ...DBX64.6
<Axis>.basic.in.traversPlusCommandActive	LBP_Axis*.E_TCPlus	DB31, ...DBX64.7
<Chan>.geoAxis1.in.traversMinusRequest	LBP_Chan*.TReqMinus	DB21, ...DBX40.4, DB21, ...DBX46.4, DB21, ...DBX52.4

Basic Program Plus	Basic Program	
<Chan>.geoAxis1.in.traversPlusRequest	LBP_Chan*.TReqPlus	DB21, ...DBX40.5, DB21, ...DBX46.5, DB21, ...DBX52.5
<Chan>.geoAxis1.in.traversMinusCommand	LBP_Chan*.TCMinus	DB21, ...DBX40.6, DB21, ...DBX46.6, DB21, ...DBX52.6
<Chan>.geoAxis1.in.traversPlusCommand	LBP_Chan*.TCPlus	DB21, ...DBX40.7, DB21, ...DBX46.7, DB21, ...DBX52.7
<Chan>.geoAxis1.in.handwheelDirInverted	LBP_Chan*.InvHWDDirOfRot	DB21, ...DBX43.0, DB21, ...DBX49.0, DB21, ...DBX55.0
<Chan>.basic.in.refPointApproachActive	LBP_Chan*.E_Ref	DB21, ...DBX33.0
<Axis>.basic.in.handwheelDirInverted	LBP_Axis*.E_InvHWDDirOfRot	DB31, ...DBX67.0
<Chan>.orientationAxis1.in.handwheelAssignedNumber	LBP_Chan*.HW1	DB21, ...DBX332.0..2
<Chan>.orientationAxis1.in.traversMinusRequest	LBP_Chan*.TReqMinus	DB21, ...DBX332.4, DB21, ...DBX336.4, DB21, ...DBX340.4
<Chan>.orientationAxis1.in.traversPlusRequest	LBP_Chan*.TReqPlus	DB21, ...DBX332.5, DB21, ...DBX336.5, DB21, ...DBX340.5
<Chan>.orientationAxis1.in.traversMinusCommand	LBP_Chan*.TCMinus	DB21, ...DBX332.6, DB21, ...DBX336.6, DB21, ...DBX340.6

16.6 通过手轮进行手动运行

Basic Program Plus	Basic Program	
<Chan>.orientationAxis1.in.traversPlusCommand	LBP_Chان*.TCPlus	DB21, ...DBX332.7, DB21, ...DBX336.7, DB21, ...DBX340.7
<Chan>.orientationAxis1.in.handwheelDirInverted	LBP_Chان*.InvHWDIrOfRot	DB21, ...DBX335.0

16.6.2 参数设置

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>

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：

16.6 通过手轮进行手动运行

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	二进制编码显示（PLC Basic Program plus 初始设置） → 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

在通过手轮进行的手动运行中，也使用下列在 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）

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。

加速度

如手动运行部分（参见 “常规参数设置 (页 747)”）。

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。

软件限位/工作区极限处的特性

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：

16.6 通过手轮进行手动运行

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

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 旋转进给率调整)

NC/PLC 接口停止信号的作用

NC/PLC 接口停止信号对通过手轮手动运行的作用（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 (针对特定通道定义通过手轮进行的手动运行的特性)

MD32084 \$MA_HANDWH_STOP_COND (通过手轮进行的手动运行的特性)

NC/PLC 接口信号	有效范围	MD20624 \$MC_HANDWH_CHAN_STOP_COND	
		位 == 0	位 == 1
<ModeGroup>.basic.out.stopAxesOnly (运行方式组停止)	几何轴/机床轴	中断，至 NC 启动	终止
<ModeGroup>.basic.out.stopAll (运行方式组停止进给轴和主轴)	几何轴/机床轴	中断，至 NC 启动	终止
<Chan>.basic.out.ncStop (NC 停止)	几何轴/机床轴	中断，至 NC 启动	终止

NC/PLC 接口信号	有效范围	MD20624 \$MC_HANDWH_CHAN_STOP_COND	
		位 == 0	位 == 1
<Chan>.basic.out.ncStopAxesAndSpindles (NC 停止进给轴和主轴)	几何轴/机床轴	中断, 至 NC 启动	终止
<Chan>.basic.out.disableFeed (进给禁用)	几何轴/机床轴	中断	终止
<Chan>.basic.out.pathFeedrateOvrFactor (进给倍率) <Axis>.basic.out.enableRapidTravers (快速运行叠加)	几何轴	倍率 == 0: 中断, 如果快速运行叠加未生效	倍率 == 0: 终止, 如果快速运行叠加未生效
<Chan>.basic.out.pathRapidTraversOvrFactor (快速运行倍率) <Axis>.basic.out.enableRapidTravers (快速运行叠加)	几何轴	快速运行倍率 == 0: 中断, 如果快速运行叠加已生效	快速运行倍率 == 0: 终止, 如果快速运行叠加已生效
<Chan>.geoAxis1.out.disableFeed (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
<Axis>.basic.out.feedrateOvrFactor (轴进给倍率)	机床轴, 非主轴	倍率 == 0: 中断	倍率 == 0: 终止
<Axis>.spindle.out.ovrFactor (主轴倍率)	主轴	倍率 == 0: 中断	倍率 == 0: 终止
<Axis>.basic.out.feedStopSpindleStop (进给停止/主轴停止)	几何轴/机床轴	中断	终止
<Axis>.basic.out.clampingInProgress (正在夹紧)	几何轴/机床轴	无作用	终止

16.6 通过手轮进行手动运行

NC/PLC 接口信号	有效范围	MD32084 \$MC_HANDWH_CHAN_STOP_COND	
		位 == 0	位 == 1
<Axis>.basic.out.enableCtrl（伺服使能）	几何轴/机床轴	中断	终止
<Axis>.drive.out.enablePulses（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。

启用安全运行停止 (SOS) 时的特性

与启用 NC/PLC 接口的停止信号的情形类似，亦可对通过激活安全运行停止 (SOS) 触发停止时的特性进行设置。

停止条件	有效范围	MD20624 \$MC_HANDWH_CHAN_STOP_COND	
		位 12 == 0	位 12 == 1 (缺省设置)
速度设定值限制生效时启用安全运行停止	机床轴	中断	终止

缺省设置下位 12 置“1”，即在通过激活安全运行停止触发停止时终止进行中的运动，并忽略新的手轮脉冲。

插补停止的前提条件是速度设定值限制生效（即 MD36933 \$MA_SAFE_DES_VELO_LIMIT > 0）。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<ModeGroup>.basic.out.stopAxesOnly	LBP_ModeGroup.A_MGStop	DB11.DBX0.5
<ModeGroup>.basic.out.stopAll	LBP_ModeGroup.A_MGStopASp	DB11.DBX0.6
<Chan>.basic.out.ncStop	LBP_Chan*.A_NCStop	DB21, ...DBX7.3
<Chan>.basic.out.ncStopAxesAndSpindles	LBP_Chan*.A_NCStopASp	DB21, ...DBX7.4
<Chan>.basic.out.disableFeed	LBP_Chan*.A_FDdisable	DB21, ...DBX6.0
<Chan>.basic.out.pathFeedrateOvrFactor	LBP_Chan*.A_FD_OR	DB21, ...DBB4
<Axis>.basic.out.enableRapidTravers	LBP_Axis*.A_RapidTrOR	DB31, ...DBX4.5
<Chan>.basic.out.pathRapidTraversOvrFactor	LBP_Chan*.A_RT_OR	DB21, ...DBB5
<Chan>.geoAxis1.out.disableFeed	LBP_Chan*.FDStop	DB21, ...DBX12.3, DBX16.3, DBX20.3
<Axis>.basic.out.feedrateOvrFactor	LBP_Axis*.A_FD_OR	DB31, ...DBB0
<Axis>.spindle.out.ovrFactor	LBP_Axis*.A_SpOR	DB31, ...DBB19
<Axis>.basic.out.feedStopSpindleStop	LBP_Axis*.A_FDSpStop	DB31, ...DBX4.3
<Axis>.basic.out.clampingInProgress	LBP_Axis*.A_Clamp	DB31, ...DBX2.3
<Axis>.basic.out.enableCtrl	LBP_Axis*.A_ContrEnable	DB31, ...DBX2.1
<Axis>.drive.out.enablePulses	LBP_Axis*.A_PulseEnable	DB31, ...DBX21.7

16.6.3 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。

16.6 通过手轮进行手动运行

示例 1：设定行程的通过手轮进行的手动运行，停止条件不是终止标准

在设定行程的通过手轮进行的手动运行（MD11346 \$MN_HANDWH_TRUE_DISTANCE == 1 或 == 3）中，若停止条件不是终止标准（参见 MD32084 \$MA_HANDWH_STOP_COND 或 MD20624 \$MC_HANDWH_CHAN_STOP_COND），则 NC/PLC 接口信号“运行请求”和“运行指令”按下图所示特性输出。



图 16-1 “信号—时间”图：设定行程的通过手轮进行的手动运行，停止条件不是终止标准 MD17900 \$MN_VDI_FUNCTION_MASK 位 0 = 0

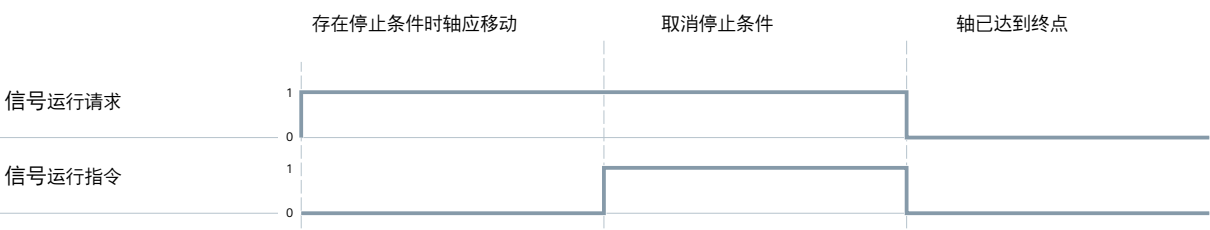


图 16-2 “信号—时间”图：设定行程的通过手轮进行的手动运行，停止条件不是终止标准 MD17900 \$MN_VDI_FUNCTION_MASK 位 0 = 1

示例 2：通过手轮进行手动运行，停止条件是终止标准

在通过手轮进行的手动运行中，在通过机床数据 MD32084 \$MA_HANDWH_STOP_COND 或 MD20624 \$MC_HANDWH_CHAN_STOP_COND 选择作为终止标准的停止条件下，**不输出运行指令，但输出相应的运行请求。**

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。

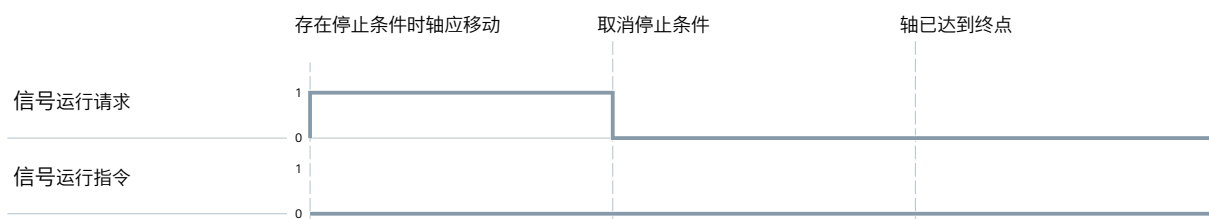


图 16-3 “信号—时间”图：通过手轮进行手动运行，停止条件是终止标准

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：设定速度的通过手轮进行的手动运行，停止条件是终止标准

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。

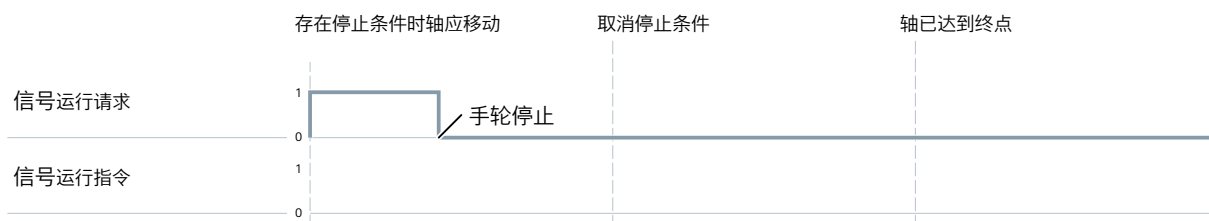


图 16-4 “信号—时间”图：设定速度的通过手轮进行的手动运行，停止条件是终止标准

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，则其特性和通过手轮进行的手动运行一样。

16.7 手动运行主轴

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。

速度

主轴的手动运行速度通常通过以下设定数据确定：

SD41200 \$SN_JOG_SPIND_SET_VELO（JOG 运行模式下主轴的转速）

若 SD41200 中输入了值“0”，以下轴专用机床数据生效：

MD32020 \$ _MA_JOG_VELO（基准 JOG 轴速度）

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。

更多信息

主轴 (页 395)

主轴倍率

主轴的手动运行速度可通过主轴倍率开关进行调整。

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。

更多信息

主轴 (页 395)

接口信号

手动运行主轴时，NC 与 PLC 之间的信号作用与机床轴相似。

仅在主轴处于位置闭环控制中时，接口信号 <Axis>.basic.in.posFineReached 或 <Axis>.basic.in.posCoarseReached（使用精准停或粗准停到达位置）才置位。

主轴专用接口信号

在 JOG 模式下，下列**发送至**主轴的接口信号不生效：

- <Axis>.spindle.out.invertM3M4（M3/M4 反向）
- <Axis>.spindle.out.oscillSetDirCw /
<Axis>.spindle.out.oscillSetDirCcw（往复旋转方向，顺时针/逆时针）
- <Axis>.spindle.out.oscillEnable（往复使能）
- <Axis>.spindle.out.deleteSValue（删除 S 值）

在 JOG 模式下，下列由主轴发出的接口信号不置位：

- <Axis>.spindle.in.cwRotActive（实际旋转方向顺时针）
- <Axis>.spindle.in.speedSetpointReached（主轴位于设定范围内）

PLC 信号

PLC → NC

Basic Program Plus	Basic Proagram	
<Axis>.spindle.out.invertM3M4	LBP_Axis*.A_M3M4Inv	DB31, ...DBX17.6
<Axis>.spindle.out.oscillSetDirCw	LBP_Axis*.A_SetRotDirectRight	DB31, ...DBX18.6
<Axis>.spindle.out.oscillSetDirCcw	LBP_Axis*.A_SetRotDirectLeft	DB31, ...DBX18.7
<Axis>.spindle.out.oscillEnable	LBP_Axis*.A_OscilSpeed	DB31, ...DBX18.5
<Axis>.spindle.out.deleteSValue	LBP_Axis*.A_DeleteS	DB31, ...DBX16.7

NC → PLC

Basic Program Plus	Basic Proagram	
<Axis>.basic.in.posCoarseReached	LBP_Axis*.E_ExactCoarse	DB31, ...DBX60.6
<Axis>.basic.in.posFineReached	LBP_Axis*.E_ExactFine	DB31, ...DBX60.7
<Axis>.spindle.in.cwRotActive	LBP_Axis*.E_ActRotRight	DB31, ...DBX83.7
<Axis>.spindle.in.speedSetpointReached	LBP_Axis*.E_SetRange	DB31, ...DBX83.5

16.8 手动运行几何轴/定向轴

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，则可通过手轮 1 至 3 同时运行 3 根几何轴。

端面轴

如果几何轴被定义为端面轴（参见 MD20100 \$MC_DIAMETER_AX_DEF）并选择了半径编程，则：

- 连续运行：
和其他轴的连续手动运行没有区别。
- 断续运行：
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，只运行一半手轮脉冲换算出的行程。

更多信息

端面轴 (页 267)

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：
信号列在 PLC 信号表中。
- 定向轴：
信号列在 PLC 信号表中。

速度

- 几何轴
几何轴的手动运行速度通过以下通道专用机床数据确定：
MD21165 \$MC_JOG_VELO_GEO（几何轴的基准 JOG 速度）
- 定向轴
定向轴的手动运行速度通过以下通道专用机床数据确定：
MD21155 \$MC_JOG_VELO_ORI（定向轴的基准 JOG 速度）

快速移动叠加

若在按下运行键/手轮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：

- 几何轴
 - MD21160 \$MC_JOG_VELO_RAPID_GEO（几何轴的 JOG 快速移动速度）
- 定向轴
 - MD21150 \$MC_JOG_VELO_RAPID_ORI（定向轴的 JOG 快速移动速度）

进给倍率

手动运行几何轴时，针对特定通道的进给倍率开关生效。

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来预设最大加速度：

MD21166 \$MC_JOG_ACCEL_GEO [<几何轴>]

其中 <几何轴> = 0, 1, 2

在加速模式 SOFT（软加速模式）中手动运行几何轴时，可通过以下机床数据设置最大加速度：

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设置最大急动度：

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手动运行定向轴激活通道专用的最大急动度：

MD21159 \$MC_JOG_JERK_ORI_ENABLE == TRUE

报警信息

报警 20060

若尝试在 JOG 方式下手动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”。

报警 20062

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。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Chan>.geoAxis1.out.handwheelSelection	LBP_Chan*.HW1	DB21, ...DBX12.0..2, DB21, ...DBX16.0..2, DB21, ...DBX20.0..2
<Chan>.geoAxis1.out.handwheelInvertDir	LBP_Chan*.InvHWDiOfRot	DB21, ...DBX15.0, DB21, ...DBX19.0, DB21, ...DBX23.0
<Chan>.geoAxis1.in.handwheelAssignedNumber	LBP_Chan*.HW1	DB21, ...DBX40.0..2
<Chan>.geoAxis1.in.traversMinusRequest	LBP_Chan*.TReqMinus	DB21, ...DBX40.4
<Chan>.geoAxis1.in.traversPlusRequest	LBP_Chan*.TReqPlus	DB21, ...DBX40.5
<Chan>.geoAxis1.in.traversMinusCommand	LBP_Chan*.TCMinus	DB21, ...DBX40.6
<Chan>.geoAxis1.in.traversPlusCommand	LBP_Chan*.TCPlus	DB21, ...DBX40.7
<Chan>.geoAxis1.in.inc1Active	LBP_Chan*.INC1	DB21, ...DBX41.0
<Chan>.geoAxis1.in.inc10Active	LBP_Chan*.INC10	DB21, ...DBX41.1
<Chan>.geoAxis1.in.inc100Active	LBP_Chan*.INC100	DB21, ...DBX41.2
<Chan>.geoAxis1.in.inc1000Active	LBP_Chan*.INC1000	DB21, ...DBX41.3
<Chan>.geoAxis1.in.inc10000Active	LBP_Chan*.INC10000	DB21, ...DBX41.4
<Chan>.geoAxis1.in.incVarActive	LBP_Chan*.INCVar	DB21, ...DBX41.5
<Chan>.geoAxis1.in.contManualTraversActive	-	DB21, ...DBX41.6
<Chan>.geoAxis1.in.handwheelDirInverted	LBP_Chan*.InvHWDiOfRot	DB21, ...DBX43.0
<Chan>.orientationAxis1.out.handwheelSelection	LBP_Chan*.HW1	DB21, ...DBX320.0..2, DB21, ...DBX324.0..2, DB21, ...DBX328.0..2

Basic Program Plus	Basic Program	
<Chan>.orientationAxis1.out.handwheelInvertDir	LBP_Chan*.InvHWDirot	DB21, ...DBX323.0 ..2, DB21, ...DBX327.0..2, DB21, ...DBX331.0..2
<Chan>.orientationAxis1.in.handwheelDirInverted	LBP_Chan*.InvHWDirot	DB21, ...DBX335.0 , DB21, ...DBX339.0 , DB21, ...DBX343.0

16.9 手动运行至固定点

16.9.1 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。

说明

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：

更多信息

编程手册之 NC 编程；补充指令 > 运行至固定点 (G75)

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。
机床功能 JOG-REPOS 或 JOG-REF 生效时以及在 AUTO 模式下的 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。

步骤

“手动运行至固定点”的步骤：

1. 选择 JOG 运行方式
2. 激活“手动运行至固定点”功能
3. 通过移动键或手轮运行机床轴

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：

`<Axis>.basic.out.jogFixedPointSelection`（手动运行至固定点）

此功能一旦生效，NC 便会通过以下接口信号确认激活：

`<Axis>.basic.in.jogFixedPointSelectedNumber`（手动运行至固定点生效）

说明

下列情形下无法激活此功能：

- NC 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：

`<Axis>.basic.in.jogFixedPointSelectedNumber`（已手动运行至固定点）

当轴通过 NC 程序、SINU_TraversePosAxis (Basic Program FC18) 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”公差范围 (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：方向已锁定<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：固定点已改变”

反馈信号 <Axis>.basic.in.jogFixedPointSelectedNumber（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：

<Axis>.basic.out.jogFixedPointSelection = 0

反馈信号 <Axis>.basic.in.jogFixedPointSelectedNumber（手动运行至固定点生效）和 <Axis>.basic.in.jogFixedPointReachedNumber（已手动运行至固定点）在轴离开固定点时被删除。

16.9 手动运行至固定点

特殊情形：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：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。

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

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）。

原理

PLC → NC

Basic Program Plus	Basic Proogram	
<Axis>.basic.out.jogFixedPointSelection	LBP_Axis*.A_JogFixPPos0..2	DB31, ...DBX13.0..2

NC → PLC

Basic Program Plus	Basic Proagram	
<Axis>.basic.in.jogFixedPointSelectedNumber	LBP_Axis*.E_JogFixPPos0..2Act	DB31, ...DBX75.0..2
<Axis>.basic.in.jogFixedPointReachedNumber	LBP_Axis*.E_JogFixPPos0..2	DB31, ...DBX75.3..5

16.9.2 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。

轴动态响应

“手动运行至固定点”中, 轴加速度和轴急动度通过下列机床数据定义:

- 通过**移动键或手轮**运行时:
 - MD32300 \$MA_MAX_AX_ACCEL [0] (动态模式 DYNNORM 中轨迹运动中轴专用的最大加速度) 中定义。
 - MD32430 \$MA_JOG_AND_POS_MAX_JERK (轴专用最大急动度)

说明

只有在为待运行的机床轴使能了轴急动度限制时, MD32430

\$MA_JOG_AND_POS_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 下的轨迹运动中的最大轴急动度)

说明

只有在为待运行的机床轴使能了轴急动度限制时, MD32431 \$MA_MAX_AX_JERK 才生效:

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

更多信息

加速度和急动度 (页 91)

16.9.3 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

16.9.4 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 `<Axis>.basic.in.jogFixedPointReachedNumber`。

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。

原理

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.jogFixedPointReachedNumber	LBP_Axis*.E_JogFixPPos0..2	DB31, ...DBX75.3..5

16.9.5 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：

<Axis>.basic.in.jogFixedPointSelectedNumber = 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：

<Axis>.basic.out.jogFixedPointSelection = 1（位 0 - 2 = 2）

通过 NC/PLC 接口信号确认激活：

<Axis>.basic.in.jogFixedPointSelectedNumber = 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：

<Axis>.basic.in.jogFixedPointReachedNumber = 1 (位 3 - 5 = 2)

PLC 信号

PLC → NC

Basic Program Plus	Basic Proagram	
<Axis>.basic.out.jogFixedPointSelection	LBP_Axis*.A_JogFixPPos0..2	DB31, ...DBX13.0.. 2

NC → PLC

Basic Program Plus	Basic Proagram	
<Axis>.basic.in.jogFixedPointSelectedNumber	LBP_Axis*.E_JogFixPPos0..2Act	DB31, ...DBX75.0.. 2
<Axis>.basic.in.jogFixedPointReachedNumber	LBP_Axis*.E_JogFixPPos0..2	DB31, ...DBX75.3.. 5

16.10 手动运行至位置

16.10.1 功能

使用“手动运行至位置”功能，操作人员能够通过设定数据在机床坐标系中给定一个位置，在该位置上可通过移动键或手轮运行所选机床轴。

应用

此功能适用于只能手动运行的机床，比如运行至钻孔位置。

16.10 手动运行至位置

前提条件

- “手动运行至位置”功能只能在 JOG 模式下激活。
机床功能 JOG-REPOS 或 JOG-REF 生效时以及在 AUTO 模式下的 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。

步骤

“手动运行至位置”的步骤：

1. 选择 JOG 运行方式
2. 给定目标位置
3. 激活“手动运行至位置”功能
4. 通过移动键或手轮运行机床轴

给定目标位置

在以下设定数据中输入目标位置：

SD43320 \$SA_JOG_POSITION（手动位置）

激活功能

选择“手动运行至位置”后，PLC 会发出以下接口信号：

<Axis>.basic.out.jogToPos（手动运行至位置）

此功能一旦生效，NC 便会通过以下接口信号确认激活：

<Axis>.basic.in.jogToPosActive（手动运行至位置生效）

说明

下列情形下无法激活此功能：

- NC 复位期间
- 有未经应答的急停
- 执行异步子程序时
- “手动运行至固定点”功能已激活时

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

通过移动键或手轮启动运行至目标位置。

轴运行至位置，到达后自动停止。

通过”准停“到达位置后发出以下 NC/PLC 接口信号：

`<Axis>.basic.in.jogPosReached`（已手动到达位置）

运行至其他位置

如果在运行至某个位置期间修改了设定数据 SD43320 的内容，轴运行将终止并发出以下报警：

报警 17822“通道 %1 轴 %2 手动运行至位置：位置发生变化”

如需运行至新位置，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. 轴停止后在 SD43320 中输入新位置。
 3. 重新启动 JOG 运行。
-

离开位置/取消

如需离开该位置，必须取消“手动运行至位置”功能。这通过将激活信号复位为“0”来进行：

`<Axis>.basic.out.jogToPos = 0`

取消后也会复位反馈信号 `<Axis>.basic.in.jogToPosActive` 和

`<Axis>.basic.in.jogPosReached`。

反向离开

反向离开该位置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，才能执行反向离开。

若该位未置位，反向离开则被禁用；尝试使用移动键或手轮使轴反向离开时，系统会输出以下通道状态信息：

“JOG：方向已锁定<轴>”

16.10 手动运行至位置

特殊情形：轴已位于目标位置

如果启动该功能时轴已位于目标位置，轴将不再运行。此时会显示以下通道状态信息：

“JOG：到达位置<轴>”

如需离开该位置，必须取消“手动运行至位置”功能。

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目标位置，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

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

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）。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.jogToPos	LBP_Axis*.A_JogToPos	DB31, ...DBX13.3

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.jogToPosActive	LBP_Axis*.E_JogToPos	DB31, ...DBX75.6
<Axis>.basic.in.jogPosReached	LBP_Axis*.E_JogPos	DB31, ...DBX75.7

参见

参数设置 (页 799)

16.10.2 参数设置

考虑轴框架和刀具长度补偿

考虑轴框架和配置为几何轴的轴的刀具长度补偿取决于以下机床数据中位 1 的设置：

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

位	值	含义
1	0	不考虑轴框架和刀具长度补偿。
	1	考虑轴框架和刀具长度补偿。

反向离开

反向离开位置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）

位	值	含义
2	0	不能朝反向离开（缺省设置）。
	1	可以朝反向离开。

目标位置

在以下设定数据中输入目标位置：

SD43320 \$SA_JOG_POSITION（手动位置）

16.10.3 边界条件

轴是分度轴

当需要运行的轴是一个分度轴且目标位置与分度位置不一致时候，轴不会运行，并会发出报警。

16.10 手动运行至位置

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）。
如果手动运行至可变位置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），系统会发出报警。轴不会运行至机床坐标系中指定的目标位置，而是运行至假设未激活补偿运行时应达到的位置。系统不会发出 NC/PLC 接口信号 <Axis>.basic.in.jogPosReached。

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。

PLC 信号

NC → PLC

Basic Program Plus	Basic Proogram	
<Axis>.basic.in.jogPosReached	LBP_Axis*.E_JogPos	DB31, ...DBX75.7

16.10.4 应用示例

目标

应通过“手动运行至位置”功能将回转轴（机床轴 4 [AX4]）运行至目标位置（100 度）。

参数设置

通过用户界面按如下方式指定目标位置：

SD43320 \$SA_JOG_POSITION[AX4] = 100 ; 机床轴 4 的目标位置 = 100

初始状态

轴 4 [AX4] 已回参考点且位于 0 度位置上。

运行至该位置

控制系统切换至 JOG 模式并选中机床轴 4。

“手动运行至位置” 功能激活：

```
<Axis>.basic.out.jogToPos = 1
```

此功能生效后，NC 会确认激活：

```
<Axis>.basic.in.jogToPosActive = 1
```

通过连续运行和机床控制面板上的正移动键启动 “手动运行至位置”。

轴运行至 100 度位置上，到达后自动停止。

通过” 准停 “到达位置后置位以下 NC/PLC 接口信号：

```
<Axis>.basic.in.jogPosReached = 1
```

PLC 信号

PLC → NC

Basic Program Plus	Basic Proagram	
<Axis>.basic.out.jogToPos	LBP_Axis*.A_JogToPos	DB31, ...DBX13.3

NC → PLC

Basic Program Plus	Basic Proagram	
<Axis>.basic.in.jogToPosActive	LBP_Axis*.E_JogToPos	DB31, ...DBX75.6
<Axis>.basic.in.jogPosReached	LBP_Axis*.E_JogPos	DB31, ...DBX75.7

16.11 圆弧手动运行

16.11.1 功能

“圆弧手动运行” 功能可使机床操作员利用移动键或手轮沿着圆弧同时运行生效平面中的两根几何轴。

应用

此功能适用于只能手动运行的机床。

前提条件

- “圆弧手动运行” 功能只能在 JOG 模式下激活。
机床功能 JOG-REPOS 和 JOG-REF 生效时无法激活此功能。
- 参与运行的轴必须回参考点。
- 生效平面不可斜置。

步骤

“圆弧手动运行” 的步骤：

1. 选择平面
2. 选择 JOG 运行方式
3. 给定圆弧参数
4. 激活“圆弧手动运行” 功能
5. 通过移动键或手轮运行几何轴

选择平面

通过编程以下其中一个 G 指令选择生效平面中的几何轴：

G 指令	平面中的第 1 根几何轴	平面中的第 2 根几何轴
G17	X	Y
G18	Z	X
G19	Y	Z

给定圆弧参数

圆弧手动运行至少需要以下数据：

- 圆心的坐标
- 内部加工时的最大圆弧半径或外部加工时的最小圆弧半径
- 加工模式（内部或外部加工）

通过用户界面在通道专用设定数据中说明：

SD42690 \$SC_JOG_CIRCLE_CENTRE（圆心）

SD42691 \$SC_JOG_CIRCLE_RADIUS（圆弧半径）

SD42692 \$SC_JOG_CIRCLE_MODE（圆弧的点动模式）

详细信息参见章节“参数设置 (页 808)”。

激活功能

选择“圆弧手动运行”后，PLC 会置位以下接口信号：

`<Chan>.basic.out.circularJogTravers`（圆弧手动运行）

此功能一旦生效，NC 便会通过以下接口信号确认激活：

`<Chan>.basic.in.circularJogTraversActive`（圆弧手动运行生效）

说明

下列情形下无法激活此功能：

- NC 复位期间
- 有未经应答的急停
- 执行异步子程序时
- 至少有一根轴已运行时
- PTP 运行激活时
- 笛卡尔手动运行在转换中生效时

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

激活此功能后，刀具可借助连续或增量运行通过移动键或手轮按如下方式运行：

- 对生效平面中的第 1 根几何轴发出运行请求时，轴会从当前位置通过带与指定圆弧相同的圆点的同心圆弧运行。
- 对生效平面中的第 2 根几何轴发出运行请求时，轴会按照与指定圆弧垂直的方向运行。为此，圆弧半径会发生改变。
- 在生效平面中分别只能执行一个运行：在圆弧上或者垂直于圆弧。
- 与生效平面中的运行无关，第 3 根几何轴可按照垂直于生效平面的方向运行。

说明

半径最大只能设为零，即：缩小到圆心以下。如果半径之后再次扩大，则应反向运行，直到缩小到圆心以下。

说明

如果在激活此功能时或更改圆心后，当前位置即为圆心，则：

- 如果是整圆，则按第 2 根几何轴的方向运行。
- 如果是圆弧段，则按角平分线的方向运行。

内部或外部加工

根据设定数据 SD42692 \$SC_JOG_CIRCLE_MODE 中的设置启动内部或外部加工：

- 内部加工
 - 只可在指定圆弧内运行。
 - SD42691 \$SA_CIRCLE_RADIUS 中的圆弧半径为**最大**半径。
- 外部加工
 - 只可在指定圆弧外运行。
 - SD42691 \$SA_CIRCLE_RADIUS 中的圆弧半径为**最小**半径。

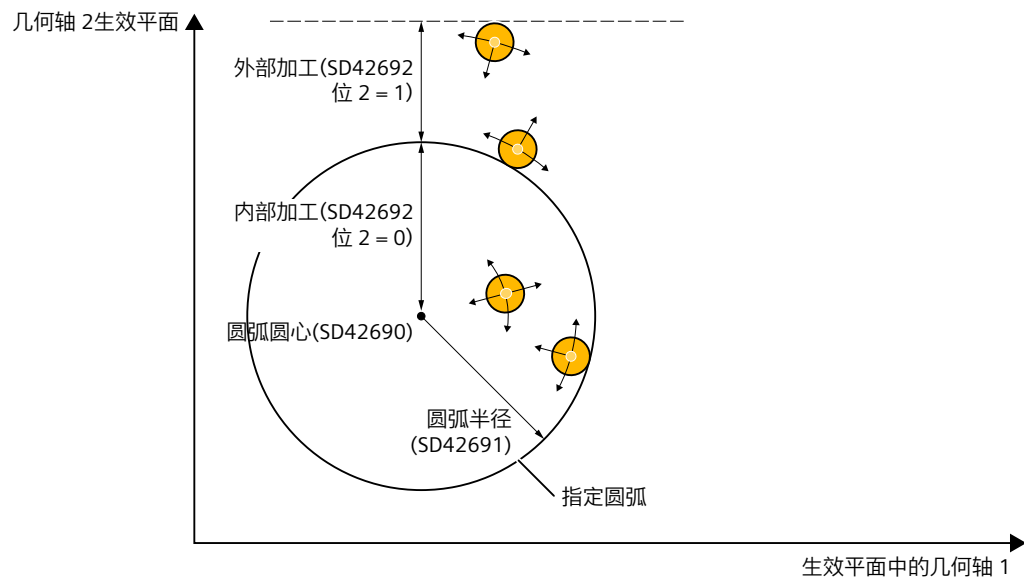


图 16-5 圆弧手动运行：内部/外部加工

圆弧段加工

通过给定起始角和结束角可将第 1 和第 2 根几何轴的工作区域限制在一个圆弧段上：

SD42693 \$SC_JOG_CIRCLE_START_ANGLE（圆弧起始角）

SD42694 \$SC_JOG_CIRCLE_END_ANGLE（圆弧结束角）

此外，为明确定义圆弧段，必须给定起始角到结束角的方向：

SD42692 \$SC_JOG_CIRCLE_MODE, 位 0

通过该位还可给定圆弧轨迹上的运行方向。

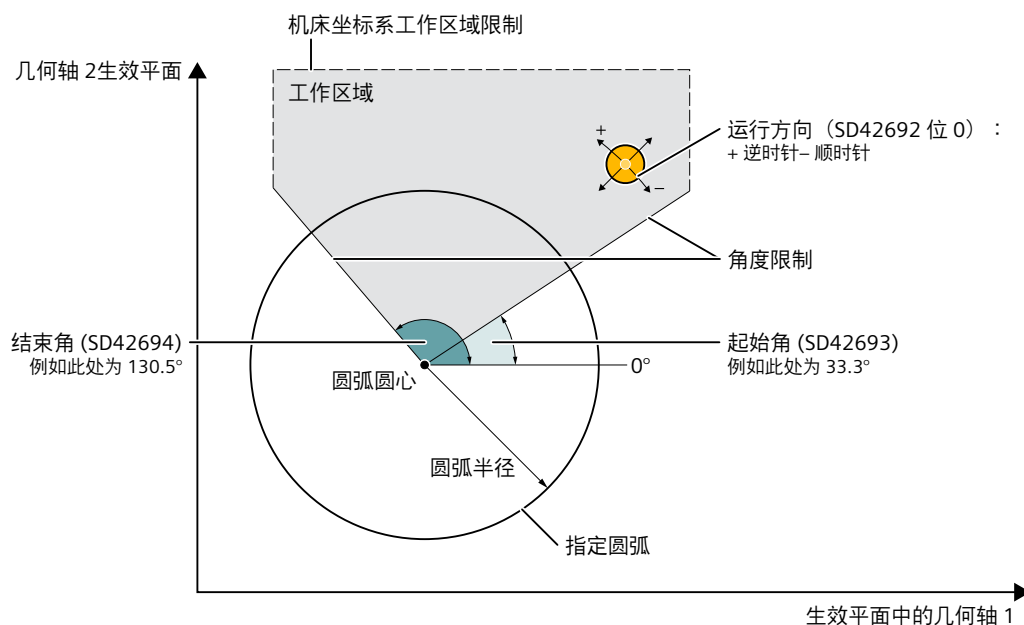


图 16-6 圆弧手动运行：圆弧段加工

考虑刀具半径

如果在设定数据 SD42692 \$SC_JOG_CIRCLE_MODE 中置位了位 1，之后在监控工作区域限制时（指定圆弧；圆弧段加工时还会有由起始角和结束角产生的限制）会考虑刀具半径：

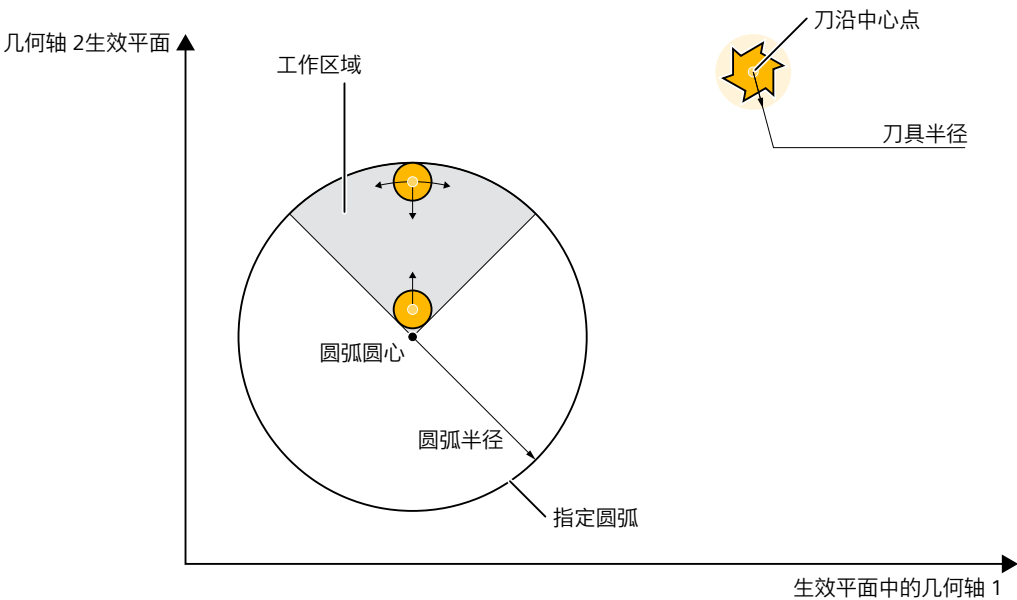


图 16-7 圆弧手动运行：带刀具半径补偿的圆弧段内部加工

说明

在考虑刀具半径的同时，还会限制内部加工时可能的最小半径。

轴监控

如果在激活此功能时当前轴位置位于工作区域限制外，轴不会运行，并会发出报警。

轴位置位于工作区域限制外：

- 内部加工时：位置在指定圆弧外
- 外部加工时：位置在指定圆弧内
- 圆弧段加工时：位置在由起始角和结束角产生的限制外（圆弧段限值）
- 刀具半径补偿激活时：位置接近限制的圆弧
- 刀具半径补偿激活和内部加工时：位置接近圆心
- 刀具半径补偿激活和外部加工时：位置接近圆弧段限值

设定数据中的更改

手动运行生效期间在设定数据 SD42690 至 SD42694 中所作的更改只有在通过触发新运行结束当前手动运动后才会生效。

说明

为了使更改快速生效，机床操作人员可进行如下操作：

1. 通过删除剩余行程终止当前运行。
2. 更改设定数据：
3. 重新启动 JOG 运行。

终止/取消

如果在手动运行期间取消“圆弧手动运行”功能 (<Chan>.basic.out.circularJogTravers = 0)，系统会终止手动运行并发出报警。

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. 轴停止后取消功能。

PLC 信号

PLC → NC

Basic Program Plus	Basic Proagram	
<Chan>.basic.out.circularJogTravers	LBP_Chan*.A_JOG_Circles	DB21, ...DBX30.6

NC → PLC

Basic Program Plus	Basic Proagram	
<Chan>.basic.in.circularJogTraversActive	LBP_Chan*.E_JOG_Circles	DB21, ...DBX377.6

16.11.2 参数设置

第二根几何轴的运行方向

可通过以下机床数据设置第二根几何轴的运行方向：

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

位	值	含义
6	0	正向运行生效平面中的第 2 根几何轴会导致半径扩大，负向运行会导致半径缩小。运行与内部或外部加工是否生效无关。
7	0	

位	值	含义
6	0	正向运行生效平面中的第 2 根几何轴始终在限制圆弧的方向上进行。即：内部加工时半径会扩大，外部加工时半径会缩小。
7	1	

位	值	含义
6	1	负向运行生效平面中的第 2 根几何轴始终在限制圆弧的方向上进行。即：内部加工时半径会扩大，外部加工时半径会缩小。
7	1	

圆心坐标作为半径值

初始设置中，轴的坐标数据在直径编程生效时始终是一个直径值。可通过以下机床数据进行修改，以给定圆弧手动运行时的圆心坐标：

MD20360 \$MC_TOOL_PARAMETER_DEF_MASK（定义刀具参数）

位	值	含义
13	0	圆弧手动运行时，圆心坐标是一个直径值（初始设置）。
	1	圆弧手动运行时，圆心坐标始终是一个半径值。

圆心

需要运行的圆弧的圆心在以下设定数据中设置：

SD42690 \$SC_JOG_CIRCLE_CENTRE（圆心）

坐标系数据以工件坐标系为基准。可用作直径值或半径值（取决于 MD20360 中位 13 的设置）。

分析应用时，仅使用活动平面上几何轴的中心坐标，而不使用与平面垂直的几何轴的坐标。

圆弧半径

内部加工时的最大圆弧半径或外部加工时的最小圆弧半径在以下设定数据中设置：

SD42691 \$SC_JOG_CIRCLE_RADIUS（圆弧半径）

半径数据以工件坐标系为基准。

内部或外部加工

加工模式（内部或外部加工）通过以下设定数据确定：

SD42692 \$SC_JOG_CIRCLE_MODE（圆弧的点动模式）

位	值	含义
2	0	进行 内部加工 。 SD42691 中的圆弧半径是可能的最大半径。
	1	进行 外部加工 。 SD42691 中的圆弧半径是可能的最小半径。

圆弧段加工

用于定义限制工作区域的圆弧段的起始角和结束角在以下设定数据中设置：

SD42693 \$SC_JOG_CIRCLE_START_ANGLE（圆弧起始角）

SD42694 \$SC_JOG_CIRCLE_END_ANGLE（圆弧结束角）

角度数据以生效平面的横坐标为基准。

起始角至结束角的方向在以下设定数据中给定：

SD42692 \$SC_JOG_CIRCLE_MODE（圆弧的点动模式）

位	值	含义
0	0	起始角至结束角的方向逆时针运行。
	1	起始角至结束角的方向顺时针运行。

说明

如果起始角和结束角等于零，可运行整圆。

考虑刀具半径

可通过以下设定数据选择考虑刀具半径：

SD42692 \$SC_JOG_CIRCLE_MODE（圆弧的点动模式）

位	值	含义
1	0	在监控工作区域限制时（指定圆弧；圆弧段加工时还会有由起始角和结束角产生的限制） 不考虑 刀具半径（初始设置）。
	1	在监控工作区域限制时 考虑 刀具半径。

16.11.3 边界条件

直径编程激活

生效平面中的几何轴的增量运行和手轮运行始终是在半径编程中进行的，即使两根几何轴中的其中一根的直径编程是激活的。

轴不是几何轴

生效平面中的两根轴必须定义为几何轴：

- 通过配置 MD20050 \$MC_AXCONF_GEOAX_ASSIGN_TAB
- 或通过在零件程序中编程指令 AXCTSWEC。

如果生效平面中的轴未定义为几何轴，轴不会运行，并发出报警。

轴是端面轴

如果参与圆弧轨迹的其中一根轴是端面轴，轴不会运行，并发出报警。

保护区

系统不会监控保护区。

16.11.4 应用示例

初始状态

- 按如下方式配置几何轴：

MD20050 \$MC_AXCONF_GEOAX_ASSIGN_TAB[0] = 1	第 1 几何轴 = 第 1 通道轴
MD20050 \$MC_AXCONF_GEOAX_ASSIGN_TAB[1] = 2	第 2 几何轴 = 第 2 通道轴
MD20050 \$MC_AXCONF_GEOAX_ASSIGN_TAB[2] = 3	第 3 几何轴 = 第 3 通道轴

- G17 生效（初始设置）：
 - X 轴 = 生效平面中的第 1 根几何轴
 - Y 轴 = 生效平面中的第 2 根几何轴
- 所选刀具的半径补偿值为 0.02 毫米。
- 当前轴位置是：
 - X 轴：10
 - Y 轴：10
- 轴已回参考点。

参数设置

SD42690 \$SC_JOG_CIRCLE_CENTRE[AX1] = 10	X 轴的圆心在 10 毫米位置
SD42690 \$SC_JOG_CIRCLE_CENTRE[AX2] = 20	Y 轴的圆心在 20 毫米位置
SD42691 \$SC_JOG_CIRCLE_RADIUS = 20	圆弧半径 20 毫米
SD42692 \$SC_JOG_CIRCLE_MODE = 2	使用 + 键运行： 逆时针运行 考虑刀具半径补偿 内部加工
SD42693 \$SC_JOG_CIRCLE_START_ANGLE = 0	无起始角生效
SD42694 \$SC_JOG_CIRCLE_END_ANGLE = 0	无结束角生效

结果

进行圆弧内部加工。

使用 + 键连续运行 X 轴时，可触发半径 10 毫米、圆心 (10, 20) 的逆时针 (G2) 运行。

16.12 沿刀具中心轴的退刀（手动退刀）

运行 Y 轴时，圆弧半径的变化范围在 0.02 到 19.98 之间。

16.12 沿刀具中心轴的退刀（手动退刀）

16.12.1 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，可通过机床功能“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

仅当下列功能之一由于通道复位或断电中断时，才能选择手动退刀：

- 使用 G33/G331/G332 攻丝
- 使用 2xx 类型的刀具（钻刀）执行加工

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

若程序终止后这些数据完整存在，在选择机床功能“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。

16.12.2 参数设置

16.12.2.1 上电后自动选择手动退刀

控制系统启动（上电）后，BAG 的通道处于参数设置的缺省运行方式下：

MD10720 \$MN_OPERATING_MODE_DEFAULT[<BAG>] = <缺省运行方式>

通过以下 NC 专用机床数据，可设置：在 BAG 的一个通道中存在退刀数据的情形下，在控制系统启动后选择 JOG 运行方式。

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	<Chan>.basic.in.jogRetractDataAvailabl == 1（退刀数据存在）时，选择 JOG 运行方式

PLC 信号

NC → PLC

Basic Program Plus	Basic Proagram	
<Chan>.basic.in.jogRetractDataAvailable	LBP_Ch* .E_RetractData	DB21, ...DBX377.5

16.12 沿刀具中心轴的退刀（手动退刀）

16.12.2.2 运行方向使能

通过以下 NC 专用机床数据，可将退刀方向限制为正运行方向，或同时使能两个运行方向：
MD10735 \$MN_JOG_MODE_MASK，位 8 = <值>

值	含义
0	使能：正运行方向
1	使能：正运行方向和负运行方向

16.12.2.3 测量系统的状态

以下轴专用机床数据显示测量系统的当前状态，或须设置为所需特性以进行选择：
MD34210 \$MA_ENC_REFP_STATE[<轴>] = <值>

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) 此处仅列出手动退刀涉及的值	

16.12.3 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。

- <Chan>.basic.in.jogRetractDataAvailable == 1（退刀数据存在）
- <ModeGroup>.basic.out.jogMode == 1（生效的运行方式：JOG）
- <Chan>.basic.in.stateReset == 1（通道状态：复位）

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。

更多信息：关于如何在控制系统重新接通（上电）后自动恢复实际位置，请参见章节“回参考点 (页 35)”。

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

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。手动退刀实际是通过 PLC 用户程序进行选择。

自动选择 JOG 运行方式

若设置了自动选择手动退刀（参见“上电后自动选择手动退刀 (页 813)”章节），且控制系统启动（上电）后存在一条通道的退刀数据，那么系统会为 BAG 选择 JOG 运行方式。

通过 PLC 用户程序选择手动退刀

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
`<Chan>.basic.in.jogRetractDataAvailable == 1`（退刀数据存在）
- 针对 BAG 选择 JOG 运行方式：
`<ModeGroup>.basic.out.jogMode == 1`（生效的运行方式：JOG）
- 通过 PI 服务“RETRAC”针对特定通道选择手动退刀。
更多信息：关于如何通过功能块 FB 4 激活 PI 服务“RETRAC”，请参见 PLC 功能手册。
- 在确认选择了手动退刀（见下）后，选择工件坐标系：
 - `<Hmi>.basic.out.displayWcsValues`（当前坐标系：1 ⇒ WCS，0 ⇒ MCS）
 - `<Hmi>.basic.in.mcsWcsSwitched`（请求：切换 MCS/WCS）

选择确认

成功选择手动退刀后，以下 NC/PLC 接口信号置位：

`<Chan>.basic.in.jogRetractActive = 1`（手动退刀生效）

16.12 沿刀具中心轴的退刀（手动退刀）

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。

PLC 信号

PLC → NC

Basic Program Plus	Basic Proogram	
<ModeGroup>.basic.out.jogMode	LBP_ModeGroup.A_JOG	DB11, ...DBX0.2

NC → PLC

Basic Program Plus	Basic Proogram	
<Chan>.basic.in.stateReset	LBP_Chan*.E_ChanReset	DB21, ...DBX35.7
<Chan>.basic.in.jogRetractActive	LBP_Chan*.E_JogRetract	DB21, ...DBX377.4
<Chan>.basic.in.jogRetractDataAvailable	LBP_Chan*.E_RetractData	DB21, ...DBX377.5

PLC → OP

Basic Program Plus	Basic Proogram	
<Hmi>.basic.out.displayWcsValues	LBP_HMI.A_ActWCS	DB19.DBX0.7

OP → PLC

Basic Program Plus	Basic Proogram	
<Hmi>.basic.in.mcsWcsSwitched	LBP_HMI.E_ActWCS	DB19.DBX20.7

16.12.4 退刀

退刀的常规特性

退刀是通过在工件坐标系（WCS）中手动运行选择“手动退刀”时定义的回退轴（几何轴）来实现的。退刀可通过机床控制面板（MCP）上的移动键或手轮来执行。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”/“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。

可切换坐标系（MCS $\Leftrightarrow$ WCS）。

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 = 1

取决于加工方式和刀具类型的退刀特性

根据加工方式和生效刀具的类型，可采用的退刀运动如下：

- 螺纹切削（G33）或攻丝（G331、G332）：
不论采用何种刀具类型，只能在回退轴（刀具轴）上执行退刀
- 刀具类型 2xx（钻孔刀具）：只能在回退轴（刀具轴）上执行退刀
- 刀具类型 4xx（磨削刀具）：无法退刀
- 刀具类型 5xx（车削刀具）：无法退刀

螺纹加工时的退刀

螺纹切削（G33）或攻丝（G331、G332）时，在运行回退轴或主轴时执行退刀。

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. 回退轴
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

手动退刀期间，下列功能将被禁用或不予执行：

- 通过 <Axis>.spindle.out.startCw / <Axis>.spindle.out.startCcw 启动主轴
- 通过功能块 SINU_PositionSpindle / SINU_TraversePosAxis (Basic Program FC18) 运行主轴或参与退刀的轴

16.12 沿刀具中心轴的退刀（手动退刀）

- 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轴设为主运行轴（指令轴、往复轴、SINU_PositionSpindle / SINU_TraversePosAxis (Basic Program FC18) / 并行轴）

PLC 信号

PLC → NC

Basic Program Plus	Basic Proogram	
<Axis>.spindle.out.startCw	LBP_Axis*.A_StartCW	DB31, ...DBX30.1
<Axis>.spindle.out.startCcw	LBP_Axis*.A_StartCCW	DB31, ...DBX30.2

16.12.5 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：<Chan>.basic.out.reset = 1（复位）
- 操作界面：软键“返回”（“<<”）

此时下列缺省设置生效：

- 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。

即刻起所有通道轴均可重新手动运行。退刀数据将保留，因此可重新选择手动退刀。

NC/PLC 接口信号

取消手动退刀后，以下 NC/PLC 接口信号复位：

<Chan>.basic.in.jogRetractActive = 0（手动退刀生效）

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Chan>.basic.out.reset	LBP_Chان*.A_Reset	DB21, ...DBX7.7
<Chan>.basic.in.jogRetractActive	LBP_Chان*.E_JogRetract	DB21, ...DBX377.4

16.12.6 重新选择

退刀数据存在时，始终可重新选择手动退刀：

<Chan>.basic.in.jogRetractDataAvailable == 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。

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，退刀将基于新的轴位置进行。

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Chan>.basic.in.jogRetractDataAvailable	LBP_Chان*.E_RetractData	DB21, ...DBX377.5

16.12.7 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。在机床功能“手动退刀”下退刀后，位置恢复后的轴必须执行回参考点。

16.12 沿刀具中心轴的退刀（手动退刀）

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
```

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：

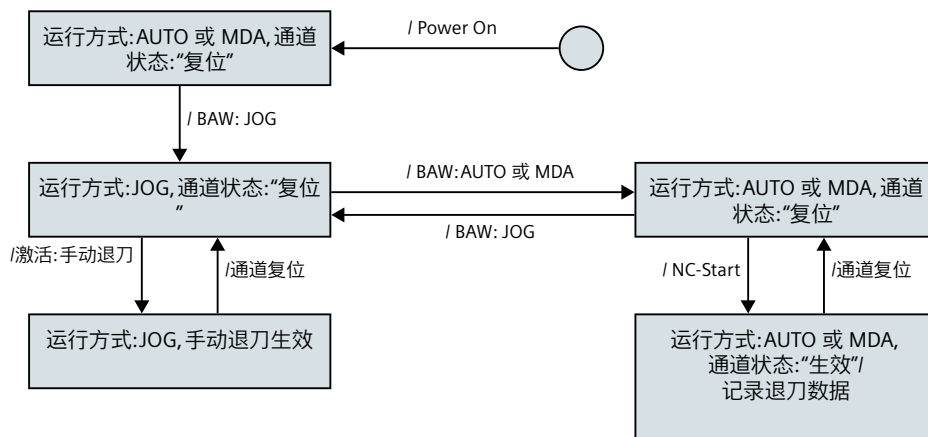
- 删除退刀数据。
- 复位 NC/PLC 接口信号：
 <Chan>.basic.in.jogRetractDataAvailable = 0（退刀数据存在）
- 备份当前的数据环境。

PLC 信号

NC → PLC

Basic Program Plus	Basic Proagram	
<Chan>.basic.in.jogRetractDataAvailable	LBP_Chan*.E_RetractData	DB21, ...DBX377.5

16.12.8 状态图



BA 运行方式

BAW 运行方式切换

图 16-8 状态图：手动退刀

16.12.9 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	-	<Chan>.basic.in.jogRetractDataAvailable	retractState, 位 0
手动退刀生效	-	<Chan>.basic.in.jogRetractActive	retractState, 位 1
回退轴	-	-	retractState, 位 2 - 3
恢复后的位置, 第 1 测量系统	\$AA_POSRES	<Chan>.basic.in.jogRetractActive	aaPosRes
恢复后的位置, 第 2 测量系统	\$AA_POSRES	<Chan>.basic.in.jogRetractDataAvailable	aaPosRes

更多信息

OPI 变量和系统变量

- 参数手册之系统变量
- PLC 功能手册

16.12 沿刀具中心轴的退刀（手动退刀）

PLC 信号

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.enc1Restored	LBP_Axis*.E_POS_RESTORED1	DB31, ...DBX71.4
<Axis>.basic.in.enc2Restored	LBP_Axis*.E_POS_RESTORED2	DB31, ...DBX71.5
<Chan>.basic.in.jogRetractActive	LBP_Chan*.E_JogRetract	DB21, ...DBX377.4
<Chan>.basic.in.jogRetractDataAvailable	LBP_Chan*.E_RetractData	DB21, ...DBX377.5

16.12.10 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：

更多信息

功能手册之进给轴和主轴

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）。

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。

16.13 在 AUTO 模式下使用手轮

16.13.1 自动方式下的手轮叠加

16.13.1.1 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

16.13 在 AUTO 模式下使用手轮

“自动方式下的手轮叠加”可调整的轴		
并行定位轴	参数“手轮叠加生效”= 1， 且 SINU_TraversePosAxis (Basic Program FC18) 的轴进给速度 > 0	参数“手轮叠加生效”= 1， 且 SINU_TraversePosAxis (Basic Program FC18) 的轴进给速度 = 0
轨迹轴	FD > 0； 对轨迹速度生效	不允许

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

- 通过轴接口信号：
`<Axis>.basic.out.resetMovement`（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，系统便进行程序段转换。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，便进行程序段转换。

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，并触发程序段转换（通过激活 `<Axis>.basic.out.resetMovement`（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“通过手轮进行手动运行 (页 763)”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：
 - `<Axis>.basic.out.handwheelSelection`（激活手轮）
- 几何轴：
 - `<Chan>.geoAxis1.out.handwheelSelection`（几何轴 1/2/3：激活手轮）

16.13 在 AUTO 模式下使用手轮

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“通过手轮进行手动运行 (页 763)”章节）：

- 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：
 <Axis>.basic.in.handwheelOvrActive（手轮叠加生效）
- 针对轨迹轴：
 <Chan>.basic.in.handwheelOvrActive（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：
 - <Axis>.basic.in.traversMinusCommandActive / <Axis>.basic.in.traversPlusCommandActive（运行指令“-”/“+”）
- 几何轴：
 - <Chan>.geoAxis1.in.traversMinusCommand / <Chan>.geoAxis1.in.traversPlusCommand（几何轴 1/2/3：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[通过接口信号 <Chan>.geoAxis1.out.handwheelSelection（几何轴 1/2/3：激活手轮）]，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。

16.13 在 AUTO 模式下使用手轮

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.resetMovement	LBP_Axis*.A_DeIDTGSReset	DB31, ...DBX2.2
<Axis>.basic.out.handwheelSelection	LBP_Axis*.A_HW1..3	DB31, ...DBX4.0..2
<Chan>.geoAxis1.out.handwheelSelection	LBP_Chan*.HW1	DB21, ...DBX12.0.. 2, .DBX16.0..2, .DB X20.0..2

NC → PLC

Basic Program Plus	Basic Program	
<Chan>.basic.in.handwheelOvrActive	LBP_Chan*.E_HWOverlay	DB21, ...DBX33.3
<Chan>.geoAxis1.in.traversMinusCommand	LBP_Chan*.TCMinus	DB21, ...DBX40.6, .DBX4 6.6, .DBX52.6
<Chan>.geoAxis1.in.traversPlusCommand	LBP_Chan*.TCPlus	DB21, ...DBX40.7, .DBX4 6.7, .DBX52.7
<Axis>.basic.in.handwheelOvrActive	LBP_Axis*.E_HWOverlay	DB31, ...DBX62.1
<Axis>.basic.in.traversMinusCommandActive	LBP_Axis*.E_TCMinus	DB31, ...DBX64.6
<Axis>.basic.in.traversPlusCommandActive	LBP_Axis*.E_TCPlus	DB31, ...DBX64.7

16.13.1.2 手轮叠加的编程和激活

概述

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转换。
- 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

手轮叠加语法: $FDA[AXi] = [\text{进给速度值}]$

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

手轮叠加语法: $FD = [\text{进给速度值}]$

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：

- 生效的组 1 的运动指令：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）

16.13 在 AUTO 模式下使用手轮

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

FD=1500

轨迹轴 X、Y 和 Z 的目标位置

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 SINU_TraversePosAxis (Basic Program FC18) 置位对应的接口信号来激活：

```
<Axis>.basic.in.handwheelOvrActive (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“定位轴 (页 277)”章节）：

```
MD32060 $MA_POS_AX_VELO (定位轴速度的初始设置)
```

更多信息

PLC 功能手册

PLC 信号

NC → PLC

Basic Program Plus	Basic Proagram	
<Axis>.basic.in.handwheelOvrActive	LBP_Axis*.E_HWOOverlay	DB31, ...DBX62.1

16.13.1.3 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 `<Chan>.basic.out.dryRunFeedrate`（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

（`<Chan>.basic.out.handwheelOffsetDrf = 1`）。

若该功能激活，那么手轮脉冲也将影响 DRF 偏移。因此应事先取消 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。

16.13 在 AUTO 模式下使用手轮

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Chan>.basic.out.handwheelOffsetDrf	LBP_Chan*.A_DRF	DB21, ...DBX0.3
<Chan>.basic.out.dryRunFeedrate	LBP_Chan*.A_DRY	DB21, ...DBX0.6

16.13.2 轮廓手轮/通过手轮设定行程（选件）

功能“轮廓手轮”

此功能激活时，可在 AUTO 和 MDA 运行方式下通过手轮运行轨迹轴和同步轴。

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。

可用性

“轮廓手轮”功能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**
控制器会将通过每个插补周期手轮脉冲的行程设定得出的运行速度限制在允许的最大速度。非立即开始运行的手轮脉冲不会累加且会在之后运行。这样可能会导致轴运行滞后于手轮。脉冲不会丢失。
- **速度设定**
通过每个插补周期手轮脉冲指定轴的运行速度。只要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。

设定方式通过以下机床数据设置：

MD11346 \$MN_HANDWH_TRUE_DISTANCE

轨迹速度

轨迹速度的单位为 mm/min，取决于：

- 每个插补周期的手轮脉冲
- 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
- 激活的增量 (INC1, 10, 100, ...)
- 可用第一几何轴的步长：
MD31090 \$MA_JOG_INCR_WEIGHT

轨迹速度与下列因素无关：

- 编写的进给速度单位 (mm/min, mm/rev)
- 编写的轨迹速度
- G0 程序段中的快速移动速度
- 倍率（除倍率 0% 外）

运行方向

运行方向取决于手轮的旋转方向：

- 顺时针方向：按编程的方向运行轴。
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：按编程方向的相反方向运行轴。
只能运行到当前程序段的开始处。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在 NC 程序中激活：

- 通过接口信号激活：
`<Chan>.basic.out.reposModeSelection`
- 通过 NC 程序激活
指令 `FD=0`。轮廓手轮逐段式接通。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。

说明

若轮廓手轮前的程序段中未编写进给速度，系统会显示报警。

功能“模拟轮廓手轮”

“轮廓手轮”功能激活时，通过“模拟轮廓手轮”功能可以模拟手轮或手轮脉冲。

激活

通过以下接口信号激活功能：

```
<Chan>.basic.out.contourHandwheelSim = 1
```

进给率之后不再通过手轮脉冲给定，而是通过 NC 程序中编写的进给率 F...。

给定旋转方向

模拟手轮的旋转方向通过以下接口信号给定：

```
<Chan>.basic.out.contourHandwheelSimInvertDir
```

倍率

“模拟轮廓手轮”功能激活时，倍率像执行 NC 程序时一样生效。

停止运行

如果取消了“模拟轮廓手轮”功能或切换了旋转方向，当前运动通过制动斜坡停止。

边界条件

- **速度限制**

- 通过接口信号 (<Chan>.basic.out.reposModeSelection) 激活“轮廓手轮”时，可通过手轮给定的轨迹速度被限制在：
最大轨迹速度 = (编程的轨迹速度 F...) * (MD12030 \$MN_OVR_FACTOR_FEEDRATE[30] + 动态响应裕量)
- 通过 NC 程序中的指令 FD=0 激活“轮廓手轮”功能时，通过机床数据中确定的最大轴速度 (MD32000 \$MA_MAX_AX_VELO) 限制可通过手轮给定的轨迹速度。

- **加速度和速度限值**

通过“轮廓手轮”功能运行轴时，应遵循机床数据中确定的轴专用加速度限值 (MD32300 \$MA_MAX_AX_ACCEL) 和速度限值 (MD32000 \$MA_MAX_AX_VELO)。

- **运行中断**

如果在运行期间通过“轮廓手轮”功能触发了 NC 停止 (<Chan>.basic.out.ncStop 或 <Chan>.basic.out.ncStopAxesAndSpindles)，该功能继续选中。不分析 NC 停止期间出现的手轮脉冲。前提条件：MD32084 \$MA_HANDWH_CHAN_STOP_COND，位 2 == 1

- **DRF**

激活的 DRF 功能 (<Chan>.basic.out.handwheelOffsetDrf) 叠加生效。

- 针对特定通道的剩余行程删除

通道专用的剩余行程删除 (<Chan>.basic.out.dryRunFeedrate) 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。

- 模具制造

通过轮廓手轮来执行模具制造程序，表面精度达不到要求或者加工速度无法保持恒定。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Chan>.basic.out.handwheelOffsetDrf	LBP_typeChanX.A_DRF	DB21, ...DBX0.3
<Chan>.basic.out.dryRunFeedrate	LBP_typeChanX.A_DRY	DB21, ...DBX0.6
<Chan>.basic.out.ncStop	LBP_typeChanX.A_NCStop	DB21, ...DBX7.3
<Chan>.basic.out.ncStopAxesAndSpindles	LBP_typeChanX.A_NCStopASp	DB21, ...DBX7.4
<Chan>.basic.out.reposModeSelection	LBP_typeChanX.A_REPOSPM_0	DB21, ...DBX30.0.. 2
<Chan>.basic.out.contourHandwheelSim	LBP_typeChanX.A_ContHWsim	DB21, ...DBX30.3
<Chan>.basic.out.contourHandwheelSimInvertDir	LBP_typeChanX.A_ContHWneg	DB21, ...DBX30.4

16.13.3 DRF 偏移

16.13.3.1 功能

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。

16.13 在 AUTO 模式下使用手轮

有效性

通过 DRF 偏移产生的零偏在所有运行方式中均生效，且在程序结束/通道复位后继续生效。

但也可程序中非模态式抑制该偏移。

通过暖启动会删除 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）

激活

为了能够在 AUTO 运行方式下通过手轮修改 DRF 偏移，DRF 必须生效。

作为发送至 NC 的激活此功能的请求，PLC 程序须将以下 NC/PLC 接口信号置位：

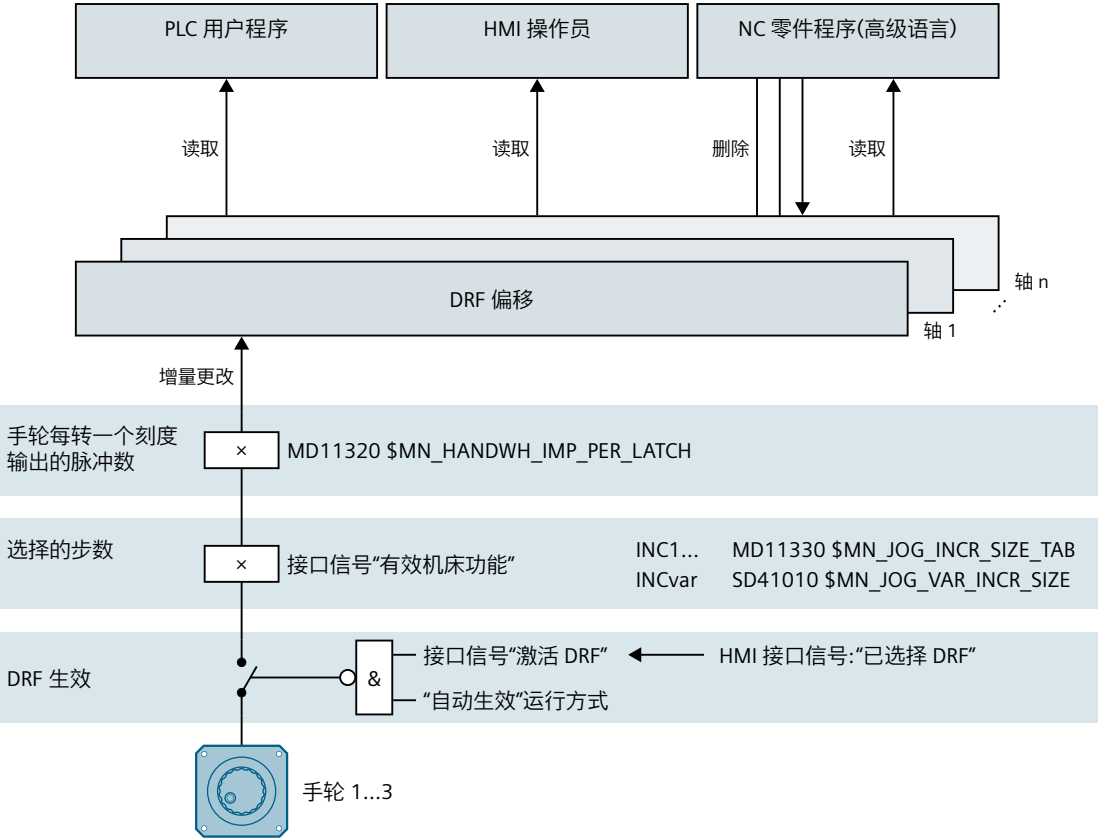
```
<Chan>.basic.out.handwheelOffsetDrf（激活手轮偏移）== 1
```

作为发送至 PLC 用户程序以表明功能在 NC 中生效的反馈消息，将以下 NC/PLC 接口信号置位：

```
<Chan>.basic.in.handwheelOvrActive（手轮叠加生效）== 1
```

对 DRF 偏移的控制和检查

可修改、删除或读取 DRF 偏移：



组件	操作
操作员	手轮运行
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）非模式式抑制 更多信息：NC 编程手册
PLC 用户程序	读取 DRF 偏移（针对特定轴） 更多信息：PLC 功能手册
操作界面	显示 DRF 偏移（针对特定轴）

16.13 在 AUTO 模式下使用手轮

说明
显示
需要在用户界面上显示轴的当前 DRF 偏移以进行检查。该显示的实现由机床制造商全权负责！
说明
删除 DRF 偏移
删除 DRF 偏移时，轴不会运行！

回参考点运行

在机床轴回参考点运行的阶段 1 中，系统会删除对应的几何轴或辅助轴的 DRF 偏移。

机床轴回参考点运行的过程中，不允许同时执行几何轴或辅助轴的 DRF 偏移。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Chan>.basic.out.handwheelOffsetDrf	LBP_Chan*.A_DRF	DB21, ...DBX0.3

NC → PLC

Basic Program Plus	Basic Program	
<Chan>.basic.in.handwheelOvrActive	LBP_Chan*.E_HWOverlay	DB21, ...DBX33.3

16.13.3.2 参数设置

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）

16.13.3.3 编程：针对轴取消叠加（CORROF）

通过程序 CORROF 将下列轴专用叠加删除：

- 通过手轮运行设置的累加零点偏移（DRF 偏移）
- 通过系统变量 \$AA_OFF 编写的位置偏移

删除叠加值会触发预处理停止，并将取消的叠加运动的位置分量接收到基准坐标系中的位置。在此情形下不运行轴。

可通过系统变量 \$AA_IM（轴的当前 MCS 设定值）读取的机床坐标系中的位置值不变。

可通过系统变量 \$AA_IW（轴的当前 WCS 设定值）读取的工件坐标系中的位置值改变，因为其包含叠加运动的被取消的分量。

说明

在 NC 程序中允许编写 CORROF。

在同步动作中不允许编写 CORROF。

句法

CORROF (<轴>,"<字符串>" [,<轴>,"<字符串>"])

含义

CORROF:	用于取消轴的下列偏移或叠加的程序： • DRF 偏移 • 位置偏移（\$AA_OFF）	
	生效方式:	模态
<轴>:	轴名称（通道轴，几何轴或者加工轴名称）	
	数据类型:	AXIS
<字符串>:	用于定义叠加类型的字符串	
	数据类型:	BOOL
	值	含义
	DRF	DRF 偏移
	AA_OFF	位置偏移（\$AA_OFF）

16.13 在 AUTO 模式下使用手轮

示例

示例 1：针对轴取消 DRF 偏移（1）

通过 DRF 手轮运行产生 X 轴上的 DRF 偏移。对于该通道中的所有其它轴，DRF 偏移不生效。

程序代码	注释
N10 CORROF (X,"DRF")	； 此处，CORROF 作用如同 DRFOF。
...	

示例 2：针对轴取消 DRF 偏移（2）

通过 DRF 手轮运行产生了 X 轴和 Y 轴上的 DRF 偏移。对于该通道中的所有其它轴，DRF 偏移不生效。

程序代码	注释
； 仅取消 X 轴的 DRF 偏移，保留 Y 轴的 DRF 偏移。	
； 如果是 DRFOF，则两个偏移均被取消。	
N10 CORROF (X,"DRF")	
...	

示例 3：针对轴取消 \$AA_OFF 位置偏移

程序代码	注释
； 为 X 轴插补一个位置偏移 == 10。	
N10 WHEN TRUE DO \$AA_OFF[X] = 10 G4 F5	
...	
； 取消 X 轴的位置偏移：\$AA_OFF[X]=0	
； 不运行 X 轴。	
； 位置偏移添加到 X 轴的当前位置。	
N80 CORROF (X,"AA_OFF")	
...	

示例 4：针对轴取消 DRF 偏移和 \$AA_OFF 位置偏移（1）

通过 DRF 手轮运行产生 X 轴上的 DRF 偏移。对于该通道中的所有其它轴，DRF 偏移不生效。

程序代码	注释
； 为 X 轴插补为 10 的位置偏移。	
N10 WHEN TRUE DO \$AA_OFF[X] = 10 G4 F5	
...	
； 仅取消 X 轴的 DRF 偏移和位置偏移。	
； 保留 Y 轴的 DRF 偏移。	
N70 CORROF (X,"DRF",X,"AA_OFF")	
...	

示例 5：针对轴取消 DRF 偏移和 \$AA_OFF 位置偏移（2）

通过 DRF 手轮运行产生 X 轴和 Y 轴上的 DRF 偏移。对于该通道中的所有其它轴，DRF 偏移不生效。

程序代码	注释
； 为 X 轴插补一个位置偏移 == 10。	
N10 WHEN TRUE DO \$AA_OFF[X] = 10 G4 F5	
...	
； 取消 Y 轴的 DRF 偏移和 X 轴的位置偏移。	
； 保留 X 轴的 DRF 偏移。	
N70 CORROF(Y,"DRF",X,"AA_OFF")	
...	

其它信息

\$AA_OFF_VAL

通过 \$AA_OFF 取消位置偏移后，相应轴的系统变量 \$AA_OFF_VAL（轴叠加的积分行程）也归零。

运行方式 JOG 下的 \$AA_OFF。

在运行方式 JOG 下，通过机床数据 MD36750 \$MA_AA_OFF_MODE 使能了该功能后，更改 \$AA_OFF 时位置偏移将作为叠加运行插补。

同步动作下的 \$AA_OFF

在通过 CORROF(<轴>,"AA_OFF") 取消位置偏移时，若立即重新设置 \$AA_OFF 的同步动作（DO \$AA_OFF[<轴>]=<值>）生效，则 取消且不重新设置 \$AA_OFF，并且显示报警 21660。如果同步动作在取消之后，比如在 CORROF 之后的程序段中才生效，则 \$AA_OFF 置位并插补位置偏移。

自动通道切换

如果一个 CORROF 编程的轴在另一个通道被激活，则在通道中可获得轴和轴交换（前提：MD30552 \$MA_AUTO_GET_TYPE > 0），然后位置偏移和/或 DRF 偏移被取消。

16.13.4 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；为第 1 几何轴选择了手轮 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）”。

若使用第 2 手轮对机床轴 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：定位轴速度叠加

假设：

通道 1：通道轴 A 对应机床轴 4，为该轴分配了手轮 1。

如果在主处理中执行程序段 `POS[A]=100 FDA[A]=0`，则无法使用 DRF 功能运行机床轴 4。也就是说，如果通道专用信号 DRF 生效，系统会发出报警 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 轴行程叠加 (SINUMERIK ONE)

假设：

通道 1：为机床轴 4 分配了手轮 2。

如果在主处理中执行由 SINU_TraversePosAxis (Basic Program FC18) 触发的、包含第 4 机床轴行程叠加的轴运动，则无法使用 DRF 功能运行机床轴 4。也就是说，如果通道专用信号 DRF 生效，系统会发出报警 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 SINU_TraversePosAxis (Basic Program FC18) 触发的、包含行程叠加的轴运动。此时会发出报警 14320“在通道 1 轴 A 上重复使用手轮 2 (4)”。

16.14 监控

限制

在手动运行和通过手轮进行手动运行时，下列限制生效：

- 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：

更多信息

功能手册之监控与补偿：轴监控、保护区

16.15 调试：手轮

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。

更多信息

加速度和急动度 (页 91)

速度、设定值/实际值系统、闭环控制 (页 141)

16.15 调试：手轮

16.15.1 简介

为了在 SINUMERIK 控制系统上使用手轮，必须通过 NC 机床数据对其进行参数设置。

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 PROFINET IO-MCP 或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 MC 控制系统上最多可设置 2 个手轮。

可连接的控制系统

手轮可通过下列组件连接：

- 通过 PROFINET IO 连接 (页 845) 模块

说明

在 SINUMERIK 控制系统上可同时运行多个通过不同模块接入的手轮。

16.15.2 通过 PROFINET IO 连接

参数设置

通过 PROFINET 模块（例如机床控制面板“MCP 483C IE”）连接的手轮在下列 NC 机床数据中进行参数设置：

- MD11350 \$MN_HANDWHEEL_SEGMENT[< x - 1 >] = 5
通过 PROFINET IO 模块连接手轮时，硬件段赋值必须为 5 (PROFINET)。
- MD11351 \$MN_HANDWHEEL_MODULE[< x - 1 >] = 1
通过 PROFINET IO 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, ... (PROFINET IO 总线上的手轮接口)
- MD11353 \$MN_HANDWHEEL_LOGIC_ADDRESS = <xx>
手轮插槽的逻辑基本地址，必须由 TIA 博途中的配置应用

其中 x = 1, 2, 3, ... (NC 中的手轮编号)

PROFINET IO 总线上的手轮接口

PROFINET IO 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
- 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：
 - 接口 X60：1.操作组件接口 MCP1 / MCP2 上连接手轮
 - 接口 X61：2.操作组件接口 MCP1 / MCP2 上连接手轮

操作组件接口 ->		MCP1		MCP2	
手轮接口 ¹⁾		1	2	1	2
FB1 参数 ²⁾		MCP1BusAdr		MCP2BusAdr	
手轮指定 ³⁾					
	MCP 483C IE	X60	X61	X60	X61

16.15 调试：手轮

操作组件接口 ->	MCP1		MCP2	
PROFINET IO 总线上的手轮接口 (y) ⁴⁾ ->	1	2	3	4
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 FC1 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) PROFINET IO 总线上手轮接口的编号 -> MD11352 \$MN_HANDWHEEL_INPUT[< x - 1 >] = y				

示例

设置 2 个手轮，通过下列操作组件连接：

操作组件接口 ->	MCP1		MCP2	
操作组件	MCP 483C		MCP 483C	
FC1 参数	MCP1BusAdr := 39		MCP2BusAdr := 192	
手轮接口	X60	-	-	X61
PROFINET IO 总线上的手轮接口 ->	1	2	3	4

表格 16-1 用于指定手轮的 NC 机床数据

机床数据	值	说明
		MCP 483C IE:NC 中手轮编号 = 1
MD11350 \$MN_HANDWHEEL_SEGMENT[0]	5	段：PROFINET IO
MD11351 \$MN_HANDWHEEL_MODULE[0]	1	模块：PROFINET IO
MD11352 \$MN_HANDWHEEL_INPUT[0]	1	PROFINET IO 总线上的手轮接口
MD11353 \$MN_HANDWHEEL_LOGIC_ADDRESS	xx	手轮的逻辑基本地址，从 TIA 博途中应用
		MCP 483C IE:NC 中手轮编号 = 2
MD11350 \$MN_HANDWHEEL_SEGMENT[1]	5	段：PROFINET IO
MD11351 \$MN_HANDWHEEL_MODULE [1]	1	模块：PROFINET IO

机床数据	值	说明
MD11352 \$MN_HANDWHEEL_INPUT [1]	4	PROFINET IO 总线上的手轮接口
MD11353 \$MN_HANDWHEEL_LOGIC_ADDRESS	xx	手轮的逻辑基本地址，从 TIA 博途中应用

表格 16-2 FC1 参数（选段）

参数	值	注释
MCPNum	:= 2	// 连接的 MCP 的数量 // MCP1 = MCP 483C IE
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 PROFINET IO

滤波时间

手轮脉冲在 PROFINET IO 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。

建议滤波时间：0.2 – 0.5 s

静态辨识

静态辨识通过连接手轮的 PROFINET IO 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。
DB10.DBX245.1	0	手轮 2 运行。
	1	手轮 2 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：

- `<Axis>.basic.out.resetMovement = 1`（轴的剩余行程删除）
- `<Chan>.basic.out.deleteDistanceToGo = 1`（通道专用的剩余行程删除）

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<code><Axis>.basic.out.resetMovement</code>	<code>LBP_typeAxisX.A_DeIDTGSpReset</code>	<code>DB31, ...DBX2.2</code>
<code><Chan>.basic.out.deleteDistanceToGo</code>	<code>LBP_typeChanX.A_DeleteDTG</code>	<code>DB21, ...DBX6.2</code>

16.15.3 TIA 博途中的配置

安装 GSD 文件

手轮通过机床控制面板连接，并在 TIA 博途中进行相应配置。如果尚未安装机床控制面板的设备描述文件，请执行以下操作：

1. 从 SIOS 门户（SIEMENS 在线支持）下载 GSD 文件。
 - 在 SIOS 搜索中输入文献 ID 62332440。
 - 下载相应的 PROFINET IO GSD 文件
2. 将 GSD 文件导入 TIA 博途
 - “其他>管理设备描述文件 (GSD)”菜单
3. 勾选要安装的文件复选框。

插入 MCP/MPP

您已经插入了一个 SINUMERIK MC 并创建了一个 PROFINET IO 子网。

1. 切换到网络视图。
2. 在硬件目录中打开以下文件夹：
 - 其他现场设备 > PROFINET IO > I/O > SIEMENS AG > SINUMERIK
 - 将列出可用的 MCP/MPP。
3. 将 MCP 拖到网络视图中。
4. 通过单击“未分配”并在快捷菜单中选择 PROFINET 接口 X150 来分配 IO 控制器。X150 是手轮操作的必选项。
5. 通过在拓扑视图中分配一个 X150 插槽来完成配置。
6. 在设备视图中继续配置 PROFINET 接口的设置。

定义模块（MCP 和手轮）

插入的模块代表一个占位符，必须进一步指定：

1. 在网络视图中双击显示的 MCP。
显示 MCP 的设备视图。在硬件目录中显示“头部模块”和“模块”条目。
2. 打开“模块”条目。
3. 将相关模块（如 MCP 483 PN）拖到设备概览中。一旦您单击目录中的模块，模块应拖到的目标行会在设备概览中选中。
4. 选择“手轮选项”条目并将模块拖到设备概览中。
读取输入地址并将其输入机床数据 MD11353 \$MN_HANDWHEEL_LOGIC_ADDRESS，以在 NC 中进行配置。

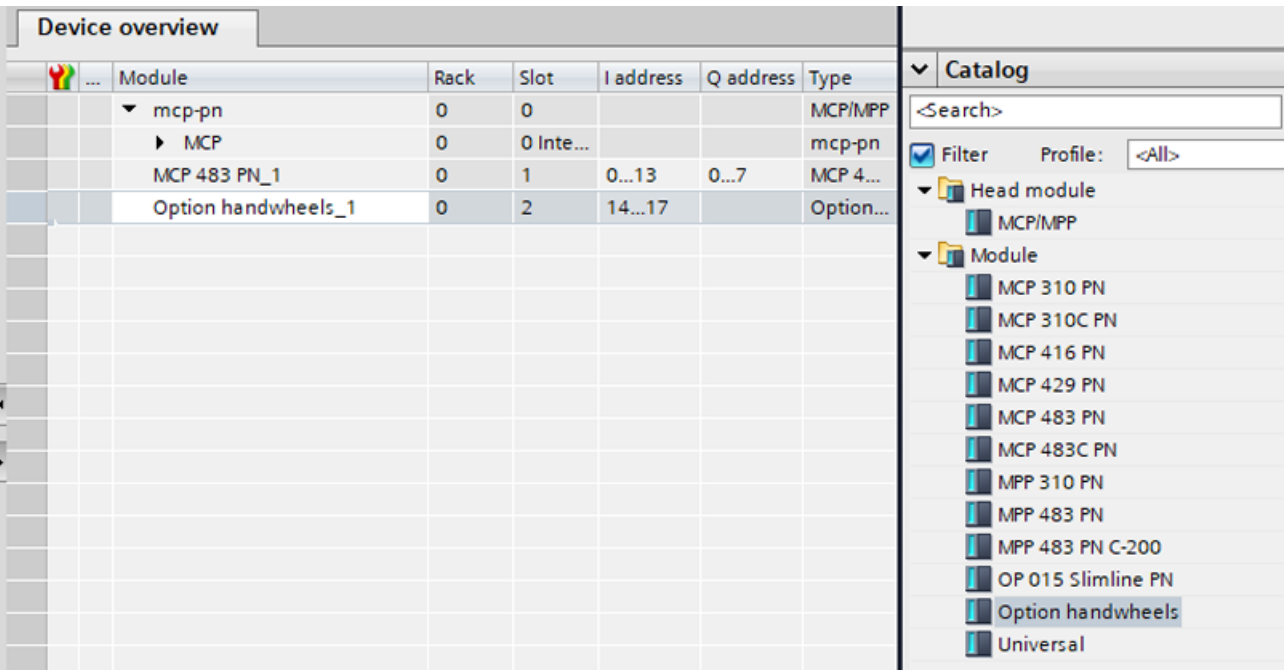


图 16-9 TIA 博途中的手轮

运行到固定挡块

17.1 简要说明

功能

借助“运行到固定挡块”功能，可运行某些运动机床部件（如尾架或套筒），从而通过这些部件在任意时间范围内对其他机床部件施加可定义的转矩或力。

特性

- 可通过零件程序或同步动作中的指令编程：
 - 激活和取消“运行到固定挡块”功能
 - 设置夹紧转矩
 - 设置监控窗口
- 通过设定数据手动调整：
 - 激活和取消“运行到固定挡块”功能
 - 设置夹紧转矩
 - 设置监控窗口
- 通过机床数据预设
 - 夹紧转矩
 - 监控窗口
- 通过 NC/PLC 接口信号识别功能状态
- 通过 NC/PLC 接口信号使能或应答
- 通过系统变量读取功能的设定状态和实际状态
- 可通过进给轴和主轴执行“运行到固定挡块”。
- 可同时为多个轴执行“运行到固定挡块”功能，并且可以同时运行其他进给轴。
- 可在多通道中对所需的所有附加数据进行带计算的程序段搜索（SERUPRO）。
- 可配合“运行到固定挡块”和“转矩降低”执行轴的仿真运行。
- “垂直”轴可在 FXS 报警时运行到固定挡块。

17.2 详细说明

17.2.1 编程

功能

运行到固定挡块

通过 FXS、FXST 和 FXSW 指令可对“运行到固定挡块”功能进行控制。

在不运行相关轴的情况下同样可以激活该功能。此时系统会立即对转矩进行限制。一旦轴运行，系统便会对该功能进行监控。

说明

同步动作

“运行到固定挡块”功能也可以由同步动作控制。

更多信息

功能手册之同步动作分册

语法

FXS [<轴>]=<请求>
FXST [<轴>]=<夹紧转矩>
FXSW [<轴>]=<窗口宽度>

含义

参数	含义
FXS:	“运行至固定挡块”功能，生效方式：模态有效
<请求>:	0 = 取消 1 = 激活
FXST:	设置夹紧转矩
<夹紧转矩>:	夹紧转矩以驱动最大转矩的 % 形式设定。 SINAMICS S120: p2003
FXSW:	设置监控窗口

参数	含义
<窗口宽度>:	以固定挡块为中心的监控窗口宽度 单位：毫米，英寸或度
<轴>:	通道轴的名称，类型：AXIS

修改转矩限制（FXST）

可在每个程序段中修改转矩限值。修改会在执行程序段中编写的运动前生效。转矩限制用作对加速度限制（ACC）的补充。

斜坡式变化

通过以下机床数据可设置一个时间，在此时间内线性修改转矩限值：

MD37012 \$MA_FIXED_STOP_TORQUE_RAMP_TIME（达到新转矩限值所经过的时间）

修改监控窗口（FXSW）

可在每个程序段中修改监控窗口。修改会在执行程序段中编写的运动前生效。

修改监控窗口时，不仅窗口宽度会改变，窗口相对轴当前实际位置的参考点也会改变。

连续路径运行（G64）激活时“运行到固定挡块”

通过以下机床数据可以设置：在连续路径运行（G64）生效期间取消功能（FXS）时，不会触发引起程序段转换（G60）的准停。

MD37060 \$MA_FIXED_STOP_ACKN_MASK PLC（是否等待 PLC 响应来执行“运行到固定挡块”功能）

位	值	含义
0	0	无需 PLC 响应即可启动运动
	1	需要 PLC 响应即可启动运动

边界条件

- 该运行可作为单个轨迹、单个程序段或跨程序段的定位轴运行进行编程。
- 多个机床轴可同时运行到固定挡块。
- 通过机床轴运行至固定挡块时，该轴上不得激活转换、耦合及框架功能。

- 在零件程序中，运行距离和该功能的激活指令必须在一个程序段中编程。
- 如果通过同步动作激活了“运行到固定挡块”功能，则也可以在单独的程序段中编程运行距离和该功能的激活指令。

17.2.2 工作流程

17.2.2.1 选择

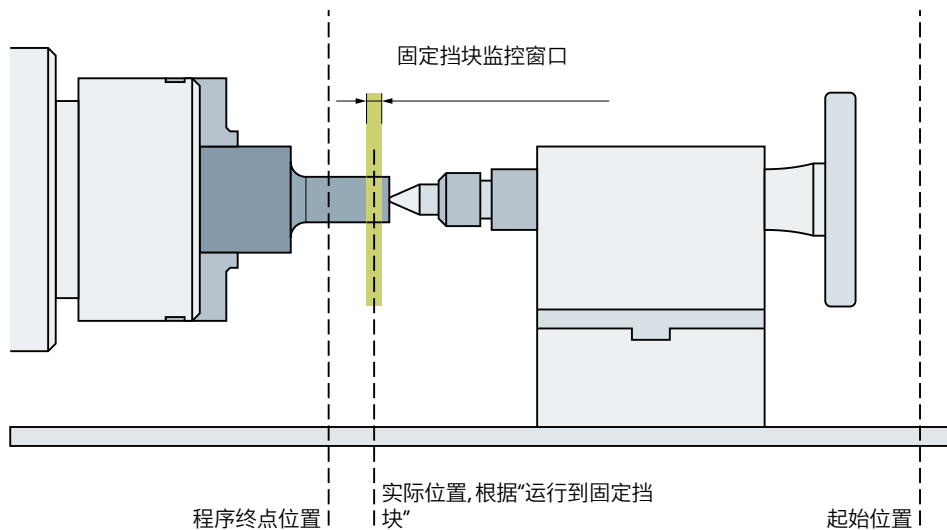


图 17-1 运行到固定挡块示例

过程

NC 检测到已通过指令 `FXS[x]=1` 选择了“运行到固定挡块”功能并通过接口信号 `<Axis>.basic.in.fixedStopRequest`（“激活运行到固定挡块”）通知 PLC，表示此功能已选择。

如果设定了相应的机床数据：

`MD37060 $MA_FIXED_STOP_ACKN_MASK` PLC（是否等待 PLC 响应来执行“运行到固定挡块”功能）

系统会等待 PLC 通过接口信号 `<Axis>.basic.out.fixedStopEnable`（“运行到固定挡块”）发出应答。

获得响应后，轴会以编程的速度从起始位置运行至编程的目标位置。固定挡块必须位于进给轴/主轴的起始位置和目标位置之间。编程的最大转矩（通过 `FXST[<轴>]` 设置的夹紧转矩）

自程序段开始时便生效，即：运动到固定挡块时也是采用下调后的转矩。转矩下调是通过 NC 中的自动加速度回落来实现。

如果在程序段中或是从程序开始时没有编程任何转矩，系统将会采用在轴专用机床数据：MD37010 \$MA_FIXED_STOP_TORQUE_DEF（固定挡块夹紧转矩的缺省设置）中输入的值。

PLC 信号

PLC → NC

Basic Program Plus	Basic Proogram	
<Axis>.basic.out.fixedStopEnable	LBP_Axis*.A_EnabTravFixedStop	DB31, ...DBX3.1

NC → PLC

Basic Program Plus	Basic Proogram	
<Axis>.basic.in.fixedStopRequest	LBP_Axis*.E_ActTravFStop	DB31, ...DBX62.4

17.2.2.2 到达固定挡块

识别固定挡块

可通过以下机床数据设置对固定挡块的识别，或对“机床轴到达固定挡块”状态的识别：
MD37040 \$MA_FIXED_STOP_BY_SENSOR = <值>（通过传感器识别是否到达固定挡块）

<值>	含义
0	机床轴的轮廓偏差超出机床数据中设置的值时，系统会认定到达固定挡块： MD37030 \$MA_FIXED_STOP_THRESHOLD（用于识别是否到达固定挡块的阈值）
1	到达固定挡块是由一个外部传感器来识别的并通过以下轴 NC/PLC 接口信号进行通知： <Axis>.basic.out.fixedStopSensor == 1（固定挡块传感器）
2	当满足 <值> == 0 或 <值> == 1 下的条件时，系统会认定到达固定挡块。

失效的 NC/PLC 接口信号

如果轴处于“到达固定挡块”状态，那么以下 NC/PLC 接口信号失效：

- `<Axis>.basic.out.disable`（进给轴/主轴禁用）
- `<Axis>.basic.out.enableCtrl`（伺服使能）

到达固定挡块时的动作

到达固定挡块时，系统会执行以下动作：

- 将驱动中的转矩提升至编写的夹紧转矩（FXST）
- 删除剩余行程
- 跟踪位置设定值
- 置位 NC/PLC 接口信号：`<Axis>.basic.in.fixedStopReached = 1`（“到达固定挡块”）
- 执行程序段转换：
 - 到达固定挡块时执行程序段转换：`MD37060 $MA_FIXED_STOP_ACKN_MASK`，位 1 = 0
到达固定挡块后立即执行程序段转换。
 - 通过 PLC 用户程序应答后再执行程序段转换：`MD37060 $MA_FIXED_STOP_ACKN_MASK`，位 1 = 1
到达固定挡块后进行等待，通过 PLC 用户程序应答后再执行程序段转换：
`<Axis>.basic.out.fixedStopAckReached == 1`（响应到达固定挡块）
为了在程序段转换后继续保持夹紧转矩，NC 会继续为机床轴输出设定值。
- 激活固定挡块监控或监控窗口

监控窗口

若在运行至固定挡块程序段中或自程序通过 FXSW 开始起未专门为监控窗口编写数值，那么将启用以下机床数据中设置的值：

`MD37020 $MA_FIXED_STOP_WINDOW_DEF`（固定挡块监控窗口的缺省设置）

如果进给轴离开了识别出固定挡块时的位置并超出了设定窗口，那么会触发报警 20093“固定挡块监控已响应”并取消“运行到固定挡块”功能。

选择监控窗口时，用户须确保只有在挡块取消时才会触发报警。

注意

"到达固定挡块"和挡块取消

一旦识别出“到达固定挡块”状态，系统便会在驱动上固定设定通过轴专用机床数据 K_V 系数（MD32200）和固定挡块识别阈值（MD37030）推导出的转速设定值。在此状态下取消挡块时，轴会加速直至达到监控窗口限值。此时达到的速度与上述机床数据中设置的值成比例。在值较大时，轴可能会加速至驱动的最大电机转速。

概述

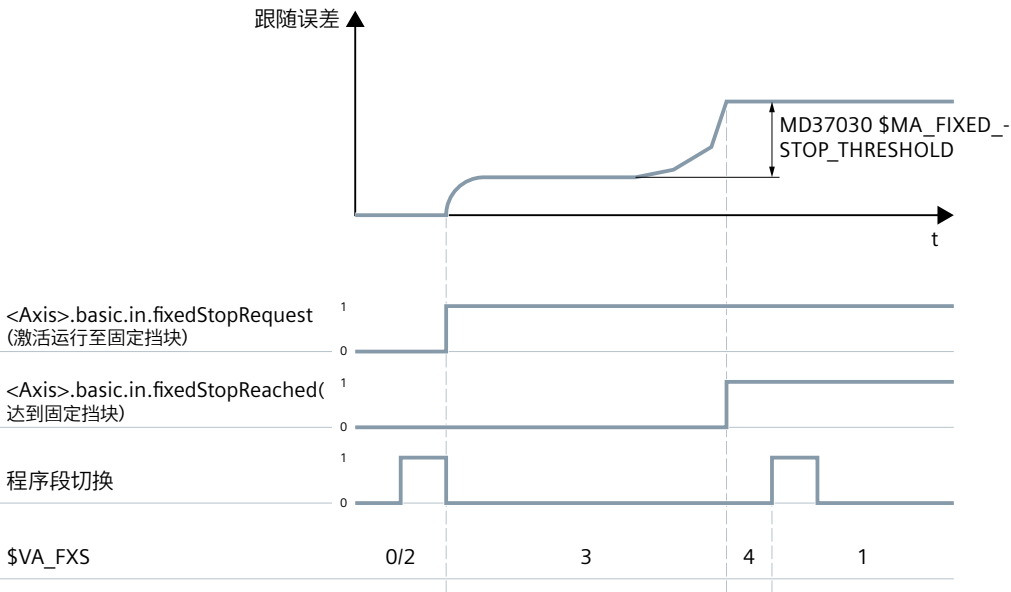


图 17-2 到达固定挡块

PLC 信号

PLC → NC

Basic Program Plus	Basic Proogram	
<Axis>.basic.out.fixedStopAckReached	LBP_Axis*.A_AckFixedStop	DB31, ...DBX1.1
<Axis>.basic.out.fixedStopSensor	LBP_Axis*.A_SensorFixedStop	DB31, ...DBX1.2
<Axis>.basic.out.disable	LBP_Axis*.A_ASpDisable	DB31, ...DBX1.3
<Axis>.basic.out.enableCtrl	LBP_Axis*.A_ContrEnable	DB31, ...DBX2.1

NC → PLC

Basic Program Plus	Basic Proagram	
<Axis>.basic.in.fixedStopReached	LBP_Axis*.E_FixedStop	DB31, ...DBX62.5

17.2.2.3 未到达固定挡块

报警抑制

可通过以下机床数据为各种终止原因抑制报警：

MD37050 \$MA_FIXED_STOP_ALARM_MASK = <值>

值	说明：被抑制的报警
0	报警 20091 “未到达固定挡块”
2	报警 20091 “未到达固定挡块” 报警 20094“固定挡块终止”
3	报警 20094“固定挡块终止”
8	报警 25042 “FOC：停止监控”
16	仅当制动测试力矩小于保持力矩时触发报警 20095。

故障或终止时的动作

出现故障或终止时，系统会执行以下动作：

- 置位 NC/PLC 接口信号：<Axis>.basic.in.fixedStopRequest = 0（激活运行到固定挡块）
- 根据以下机床数据中的设置等待应答：
 - MD37060 \$MA_FIXED_STOP_ACKN_MASK
 - <Axis>.basic.out.fixedStopEnable == 0（使能运行到固定挡块）
- 取消转矩限制
- 执行程序段转换

概述

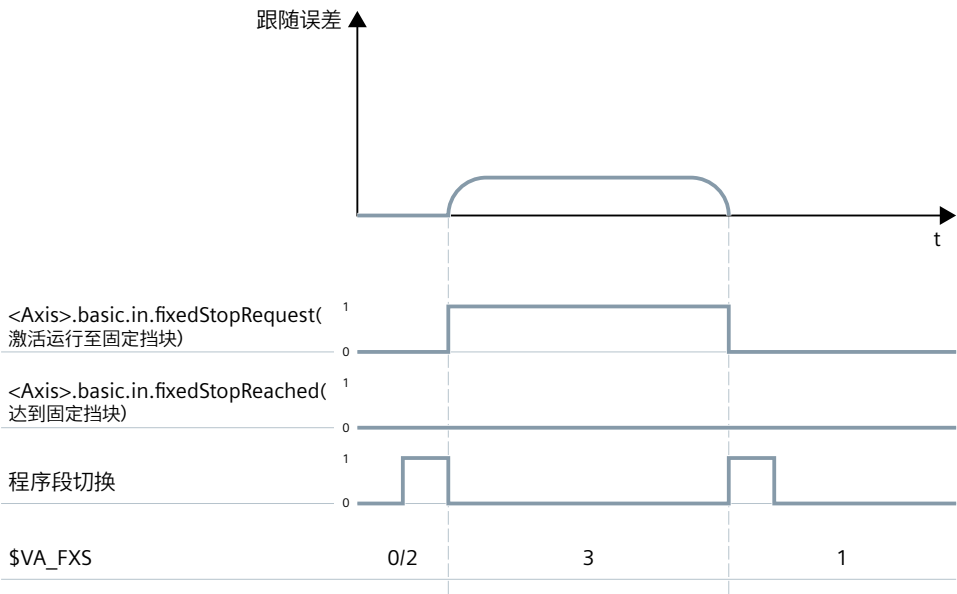


图 17-3 未到达固定挡块

PLC 信号

PLC → NC

Basic Program Plus	Basic Proagram	
<code><Axis>.basic.out.fixedStopEnable</code>	<code>LBP_Axis*.A_EnabTravFixedStop</code>	<code>DB31, ...DBX3.1</code>

NC → PLC

Basic Program Plus	Basic Proagram	
<code><Axis>.basic.in.fixedStopRequest</code>	<code>LBP_Axis*.E_ActTravFStop</code>	<code>DB31, ...DBX62.4</code>
<code><Axis>.basic.in.fixedStopReached</code>	<code>LBP_Axis*.E_FixedStop</code>	<code>DB31, ...DBX62.5</code>

17.2.2.4 取消

“运行到固定挡块”功能通过在 NC 程序的程序段中写入 `FXS[<轴>] = 0` 指令来取消。

取消功能时的动作

取消此功能时，系统会执行以下动作：

- 触发预处理停止（STOPRE）
- 复位 NC/PLC 接口信号
 - <Axis>.basic.in.fixedStopRequest = 0（激活运行到固定挡块）
 - <Axis>.basic.in.fixedStopReached = 0（到达固定挡块）
- 根据以下机床数据等待 PLC 用户程序应答：
 - MD37060 \$MA_FIXED_STOP_ACKN_MASK
 - <Axis>.basic.out.fixedStopEnable == 0（使能运行到固定挡块）
 - <Axis>.basic.out.fixedStopAckReached == 0（响应到达固定挡块）
- 结束跟踪运行
- 将轴重新接收至位置闭环控制

脉冲使能

可通过以下方式撤销脉冲使能或脉冲禁用：

- 驱动：通过端子 EP（使能脉冲）
- NC/PLC 接口信号：<Axis>.drive.out.enablePulses（“脉冲使能”）

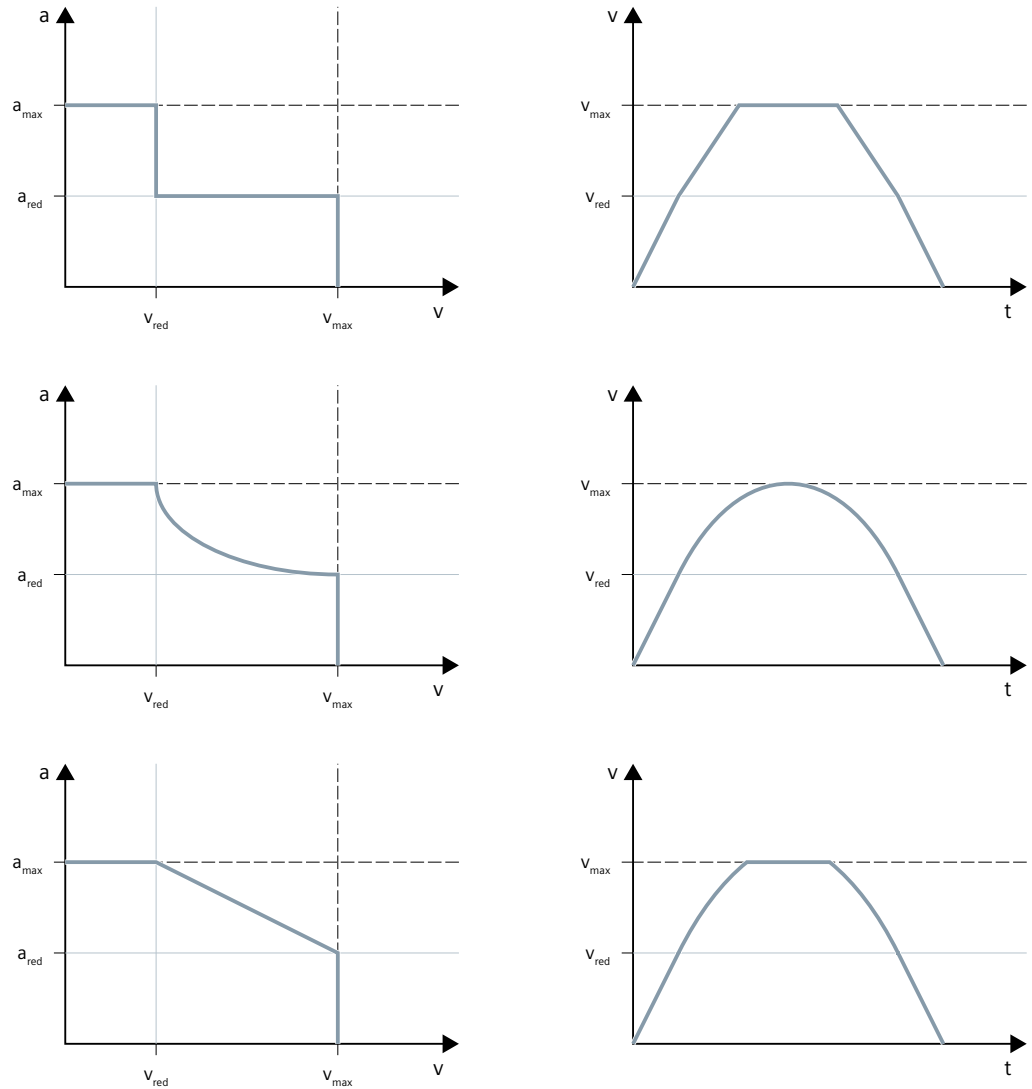
通过以下机床数据可以设置固定挡块处的特性：

MD37002 \$MA_FIXED_STOP_CONTROL，位 0 和位 1（对“运行到固定挡块”功能的顺序控制）

位	值	含义
0	挡块上脉冲 禁用 时的特性	
	0	“运行到固定挡块”功能终止
	1	“运行到固定挡块”功能中断，即驱动不再运转。
1	挡块上脉冲 使能 时的特性	
	0	转矩跳到最大转矩
	1	通过以下机床数据中设定的时间线性启用转矩： MD37012 \$MA_FIXED_STOP_TORQUE_RAMP_TIME

特殊情形：取消功能期间删除脉冲使能

若在取消功能期间在“等待 PLC 应答”状态下删除脉冲使能，那么转矩限制会被设置为 0。在此阶段中重新设置脉冲使能时，驱动中将不再生成转矩。一旦取消完成，便可重新正常运行轴。

概述

① 写入了取消指令 $\text{FXS}[\text{<轴>}]=0$ 的运行程序段

图 17-4 取消“运行到固定挡块”功能

PLC 信号

PLC → NC

Basic Program Plus	Basic Proogram	
<Axis>.basic.out.fixedStopAckReached	LBP_Axis*.A_AckFixedStop	DB31, ...DBX1.1
<Axis>.basic.out.fixedStopEnable	LBP_Axis*.A_EnabTravFixedStop	DB31, ...DBX3.1
<Axis>.drive.out.enablePulses	LBP_Axis*.A_PulseEnable	DB31, ...DBX21.7

NC → PLC

Basic Program Plus	Basic Proogram	
<Axis>.basic.in.fixedStopRequest	LBP_Axis*.E_ActTravFStop	DB31, ...DBX62.4
<Axis>.basic.in.fixedStopReached	LBP_Axis*.E_FixedStop	DB31, ...DBX62.5

17.2.3 程序段搜索时的特性

功能

带计算的程序段搜索

- 如果目标程序段位于一个含 FXS 指令的程序段中且轴尚未到达固定挡块，NC 会使轴逼近固定挡块。
- 如果目标程序段位于一个不含 FXS 指令的程序段中且轴还处于那个位置，NC 会使轴离开固定挡块。
- 若轴处于“到达固定挡块”状态，则会显示信息 10208“通过 NC 启动继续执行程序”。通过 NC 启动可继续执行程序。
- 启动目标程序段时，夹紧力矩 FXST 和监控窗口 FXSW 都为正常程序加工时的值。

不带计算的程序段搜索

程序段搜索期间，系统会忽略 FXS、FXST 和 FXSW 指令。

带 FXS 或 FOC 的程序段搜索

用户在目标程序段的程序区域内标出 FXS 或 FOC 功能，以获知该加工过程中所有最后有效的状态和功能。NC 在程序测试模式下自行启动所选择的程序。查找到目标程序段后，NC 会

停在目标程序段开头处，在内部再次取消程序测试，接着在程序段显示窗口中显示停止条件“搜索目标已找到”。

 小心

SERUPRO 实际上并没有考虑 FXS 指令

NC 只会在没有转矩限制的条件下仿真逼近编程的 FXS 程序段的终点位置。

如果 FXS 指令位于程序开头和搜索目标中间，NC 不会真正地执行该指令，它只会仿真到编程终点的运动。

用户可以在零件程序中同时记录 FXS 的激活和取消。如果有需要，用户可以启动异步子程序，从而在该 SERUPRO 异步子程序中激活或取消 FXS。

系统变量

可通过以下系统变量读取“运行到固定挡块”功能的设定和实际状态：

- \$AA_FXS（设定状态）
- \$VA_FXS（实际状态）

SERUPRO: \$AA_FXS（设定状态）

在执行 SERUPRO 期间，\$AA_FXS 会根据“运行到固定挡块”功能的激活状态提供以下值：

“运行到固定挡块”功能的激活状态	系统变量 \$AA_FXS ==
“取消”	0（轴不在固定挡块处）
“激活”	3（“运行到固定挡块”选择生效）

说明

在执行 SERUPRO 期间，变量 \$AA_FXS 只提供值 0 和 3。因此，如果有基于 \$AA_FXS 的程序分支，SERUPRO 指令下的程序运行会和正常的程序加工有所区别。

SERUPRO: \$VA_FXS（实际状态）

在执行 SERUPRO 期间，变量 \$VA_FXS 始终提供轴的真正状态。

17.2 详细说明

示例

在 SERUPRO 异步子程序中，可通过系统变量 \$AA_FXS 和 \$VA_FXS 得出“运行到固定挡块”功能的当前状态并对此作出响应：

程序代码：FXS_SERUPRO_ASUP.MPF	注释
N100 WHEN (\$AA_FXS[X]==3) AND (\$VA_FXS[X]==0) DO FXS[X]=1	; 设定状态==“选择未生效”，且 ; 实际状态==“轴不在固定挡块处” ; => “激活”
N200 WHEN (\$AA_FXS[X]==0) AND (\$VA_FXS[X]==1) DO FXS[X]=0	; 设定状态==“轴不在固定挡块处”，且 ; 实际状态==“成功逼近固定挡块” ; => “激活”
N1020 REPOSA	; 所有轴再次逼近轮廓

显示 REPOS 偏移

找到搜索目标后，以下 NC/PLC 接口信号会显示每根轴当前的“运行到固定挡块”相关状态：

- <Axis>.basic.in.fixedStopRequest（激活运行到固定挡块）
- <Axis>.basic.in.fixedStopReached（到达固定挡块）

示例：

如果轴停止在固定挡块处且目标程序段在取消 FXS 后依然存在，系统会通过 <Axis>.basic.in.fixedStopReached（到达固定挡块）显示作为 REPOS 偏移的新的目标位置。

REPOS 和 FXS

通过 REPOS 指令，FXS 的功能可以自行重复，在下文以 FXS-REPOS 代指该功能。FXS-REPOS 指令的顺序与 FXS_SERUPRO_ASUP.MPF 程序类似。执行该程序时，NC 会考虑每根轴的情况，并使用搜索目标前最后编程的转矩。

用户可在 SERUPRO 异步子程序中对 FXS 作特殊处理。

这适用于：

在 SERUPRO 异步子程序中执行的每个 FXS 动作都会自动负责

\$AA_FXS[<轴>] = \$VA_FXS[<轴>]。

这样，FXS-REPOS 对轴 X 无效。

取消 FXS-REPOS

FXS-REPOS 是通过以下方式取消的：

- FXS 同步动作与 REPOSA 指令有关
- SERUPRO 异步子程序中的 \$AA_FXS[X] = \$VA_FXS[X]

说明
不带 FXS 指令的 SERUPRO 异步子程序或没有 SERUPRO 异步子程序都会自动导致 FXS-REPOS。

 **小心**

FXS-REPOS 时轴高速运行
FXS-REPOS 会使所有的轨迹轴同时向目标点运行。这样，带或不带 FXS 指令的轴都会同时以在目标程序段中有效的 G 指令和进给率运行。这会使轴快速 (G0) 或高速逼近固定挡块。

FOC-REPOS

FOC-REPOS 的作用与 FXS-REPOS 类似。
通过 FOC-REPOS 指令无法实现在程序预处理期间出现不断变化的转矩曲线。

示例
轴 X 从位置 0 运行到位置 100。此时 NC 每隔 20 毫米就激活一次 FOC，一次长达 10 毫米。转矩曲线是由非模态 FOC 生成的，因此无法通过 FOC-REPOS 进行跟踪。FOC-REPOS 可使带或不带 FOC 的轴 X 按照在目标程序段前最后编程的位置运行。
FXS “运行到固定挡块” 功能的编程示例，参见功能手册之基本功能。

PLC 信号

NC → PLC

Basic Program Plus	Basic Proagram	
<Axis>.basic.in.fixedStopRequest	LBP_Axis*.E_ActTravFStop	DB31, ...DBX62.4
<Axis>.basic.in.fixedStopReached	LBP_Axis*.E_FixedStop	DB31, ...DBX62.5

17.2.4 复位和功能中断时的特性

NC 复位

只要“运行到固定挡块”功能还未处于“轴成功到达固定挡块”状态，就可以通过 NC 复位中断该功能。

即使已经到达了固定挡块，但还未完全到达指定的固定挡块转矩，就仍可以通过 NC 复位中断该功能。此时，轴的设定位置会与当前实际位置同步。

一旦该功能处于“轴成功到达固定挡块”状态，同样是通过 NC 复位就可以继续保持有效。

功能中断

功能中断是通过以下事件触发的：

- 急停

 小心

运行到固定挡块时可导致机床故障

必须要确保在“运行到固定挡块”功能激活期间触发和复位“急停”不会导致机床故障。

例如：设置和撤销脉冲禁用时的特性：

MD37002 \$MA_FIXED_STOP_CONTROL, 位 0（固定挡块处脉冲禁用时的特性）

- 位 0 = 0：“运行到固定挡块”功能被中断
- 位 0 = 1：“运行到固定挡块”功能被中断，即驱动不再运转。
一旦再次取消脉冲禁用，驱动会再次以固定挡块转矩运转。

说明

NC 和驱动在“急停”状态下是断开的，也就是说，PLC 必须作出响应。

- 功能状态：“未到达固定挡块”
- 功能状态：“固定挡块终止”
- 通过 PLC 用户程序终止：
`<Axis>.basic.in.fixedStopRequest = 0`（“激活运行到固定挡块”）
- 撤销脉冲使能和机床参数设置：
MD37002 \$MA_FIXED_STOP_CONTROL, 位 0 = 0（见上）

PLC 信号

NC → PLC

Basic Program Plus	Basic Proagram	
<Axis>.basic.in.fixedStopRequest	LBP_Axis*.E_ActTravFStop	DB31, ...DBX62.4

17.2.5 与其他功能综合使用时的特性

带剩余行程删除的测量

无法在一个程序段中同时编程“运行到固定挡块”功能 (FXS) 和“带剩余行程删除的测量”功能 (MEAS)。一个功能作用于轨迹轴，另一个作用于定位轴，或者两个功能都作用于定位轴。

轮廓监控

在“运行到固定挡块”功能激活期间无法进行轮廓监控。

定位轴

在定位轴 POSA 上使用“运行到固定挡块”功能时，即使定位轴在该时间点上仍未到达固定挡块，NC 依然会执行程序段转换。

垂直轴

即使 NC 发出报警信息，仍可以在垂直轴上使用“运行到固定挡块”功能。

如果在运行到固定挡块期间，垂直轴上出现了功能专用的报警，NC 就不会复位 NC/PLC 接口信号 <ModeGroup>.basic.in.ready（运行方式组准备就绪）。这样，相应的驱动也不会断电。

该功能与电子配重对垂直轴造成的影响一样，可通过以下机床数据进行配置：

MD37052 \$MA_FIXED_STOP_ALARM_REACTION

更多信息

有关垂直轴的更多信息请参考：

- SINAMICS S120 驱动功能功能手册
- 功能手册之监控与补偿；补偿、电子配重

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<ModeGroup>.basic.in.ready	LBP_ModeGroup.E_MGOK	DB11.DBX6.3

17.2.6 设定数据

通过功能专用指令 FXS、FXST 和 FXSW 编程的值会和程序段同步写入以下立即生效的轴专用设定数据中：

激活/取消该功能

SD43500 \$SA_FIXED_STOP_SWITCH（激活/取消“运行到固定挡块”功能）

夹紧转矩

SD43510 \$SA_FIXED_STOP_TORQUE（夹紧转矩）

说明

夹紧转矩超过最大电机转矩的 100%

SD43510 中的夹紧转矩只允许短时间内超过最大电机转矩 100%。除此以外，驱动会对最大电机转矩进行限制。比如：可通过以下驱动参数对最大电机转矩进行限制：

- p1520/p1521 转矩/力上限值/转矩/力下限值
- p1522/p1523 转矩/力上限值/转矩/力下限值
- p1530/p1531 电机功率限值/再生功率限值
- p0640 电流限值
- p0326 电机翻转力修调系数

有关驱动参数和功能的详细信息请参考：

更多信息

- SINAMICS S120/S150 参数手册
- SINAMICS S120 驱动功能功能手册

监控窗口

SD43520 \$SA_FIXED_STOP_WINDOW（监控窗口）

缺省设置

可通过以下机床数据设置设定数据的缺省设置：

- 夹紧转矩：
MD37010 \$MA_FIXED_STOP_TORQUE_DEF（夹紧转矩的缺省设置）
- 监控窗口：
MD37020 \$MA_FIXED_STOP_WINDOW_DEF（监控窗口的缺省设置）

有效性

夹紧转矩和监控窗口的设定数据是立即生效的。因此，用户或 PLC 用户程序可随时调整夹紧状态以适应不同的加工条件。

更多信息

有关机床数据和设定数据的详细信息请参考：
参数手册之详细的机床数据描述

17.2.7 系统变量

设定状态/实际状态

可通过以下系统变量读取“运行到固定挡块”功能的设定和实际状态：

- \$AA_FXS = <值>（“运行到固定挡块”功能的设定状态）
- \$VA_FXS = <值>（“运行到固定挡块”功能的实际状态）

<值>	说明
0	轴不在固定挡块处
1	轴成功到达固定挡块
2	轴未到达固定挡块
3	“运行到固定挡块”选择生效
4	已识别出固定挡块
5	“运行到固定挡块”功能被取消

附加信息

若在运行至固定挡块时出现错误（\$VA_FXS == 2），则会在以下系统变量中显示附加信息：

- \$VA_FXS_INFO = <值>（“运行到固定挡块”时的附加信息）

<值>	说明
0	无附加信息
1	未编写逼近运动
2	到达编写的终点位置，运动结束
3	通过复位终止，<Chan>.basic.out.reset == 1
4	已离开监控窗口
5	转矩降低已被驱动拒绝
6	PLC 已撤销使能，<Axis>.basic.out.fixedStopEnable == 0（运行到固定挡块已使能）

变量 \$AA_FXS 的应用示例

为了实现程序段转换，故障情形下不可触发报警。故障原因可于随后通过系统变量 \$VA_FXS_INFO 获取并相应应对。

前提条件：MD37050 \$MA_FIXED_STOP_ALARM_MASK = 0（运行到固定挡块时不触发任何报警）

程序代码	注释
X300 Y500 F200 FXS[X1]=1 FXST[X1]=25 FXSW[X1]=5	; 运行到固定挡块
IF \$AA_FXS[X1]==2 GOTOF FXS_ERROR	; IF 固定挡块 == 到达
R100=\$AA_IM[X1]	; THEN（正常情形）
IF R100 ...	; 分析当前
...	; 实际位置
GOTOF PROG_END	
FXS_ERROR:	; ELSE（故障情形）
CASE(\$VA_FXS_INFO[X1]) OF 0 GOTOF LABEL_0 OF 1 GOTOF LABEL_1 ...	; 故障处理
LABEL_0: ...	
GOTOF CASE_END	
LABEL_1: ...	
GOTOF CASE_END	
...	
CASE_END:	
...	
PROG_END:M30	; ENDIF 程序结束

PLC 信号

PLC → NC

Basic Program Plus	Basic Proagram	
<Chan>.basic.out.reset	LBP_Chان*.A_Reset	DB21, ...DBX7.7
<Axis>.basic.out.fixedStopEnable	LBP_Axis*.A_EnabTravFixedStop	DB31, ...DBX3.1

17.2.8 报警

报警 20091 “未到达固定挡块”

如果在执行“运行到固定挡块”功能时未到达目标位置，NC 会显示报警 20091 “未到达固定挡块”并执行程序段转换。

报警 20092 “运行到固定挡块功能仍有效”

如果轴到达固定挡块后，NC 向轴发出了新的运行请求或是功能选择，NC 会显示报警 20092 “运行到固定挡块功能仍有效”。

报警 20093 “固定挡块处的静态监控已触发”

如果轴到达固定挡块后，轴运动的行程超过了设定数据

SD43520 FIXED_STOP_WINDOW（固定挡块监控窗口）

中给定的值，NC 会显示报警 20093 “固定挡块处的静态监控已触发”，取消“运行到固定挡块”功能并设置以下系统变量：

\$AA_FXS[x] = 2

报警 20094 “功能已被中断”

如果因撤销脉冲使能而导致无法再输出夹紧转矩或 NC/PLC 接口上的响应信号已经复位，NC 会中断“运行到固定挡块”功能：

- 所需的应答信号：MD37060 \$MA_FIXED_STOP_ACKN_MASK，位 0 = 1
- 应答信号：<Axis>.basic.out.fixedStopEnable == 0（使能运行到固定挡块）

激活固定挡块报警

通过以下机床数据

- 报警 20091 “未到达固定挡块”
- 报警 20094“固定挡块终止”

17.3 示例

可以设置是否显示固定挡块报警：

MD37050 \$MA_FIXED_STOP_ALARM_MASK（激活固定挡块报警）

固定挡块报警时的功能特性可设置

通过以下机床数据可以设置：在出现功能报警时，该功能不会被中断：

- 报警 20090 无法运行到固定挡块
- 报警 20091 未到达固定挡块
- 报警 20092 运行到固定挡块功能仍有效
- 报警 20093 固定挡块处的静态监控已触发
- 报警 20094 运行到固定挡块功能被中断

MD37052 \$MA_FIXED_STOP_ALARM_REACTION（固定挡块报警时的响应）

重新编程后的报警抑制

“运行到固定挡块”功能可用于简单的测量过程。

例如：可以使刀具顶住一个指定的障碍物来测量刀具长度，从而判断刀具是否折断。在该情况下需抑制固定挡块报警。在之后“正常”使用该功能来夹紧工件时，需要通过零件程序指令重新激活该报警。

PLC 信号

PLC → NC

Basic Program Plus	Basic Proogram	
<Axis>.basic.out.fixedStopEnable	LBP_Axis*.A_EnabTravFixedStop	DB31, ...DBX3.1

17.3 示例

示例 1：通过静态同步动作运行到固定挡块

在一个静态动态动作中通过 R 参数（\$R1）请求触发“运行到固定挡块”（FXS）。

程序代码	注释
N10 IDS=1 WHENEVER	；静态同步动作 1：

程序代码	注释
((R1==1) AND (\$AA_FXS[Y]==0)) DO	; R1==1 (请求 Y 的 FXS) 以及 ; 避免重复选择 ; \$AA_FXS[Y]==0 (轴不在固定挡块处): =>
\$R1=0 FXS[Y]=1	; 复位 \$R1, 激活 Y 的 FXS
FXST[Y]=10	; 固定挡块转矩: 10%
FA[Y]=200	; 轴向进给率 Y: 200
POS[Y]=150	; 定位运动 Y
N11 IDS=2 WHENEVER (\$AA_FXS[Y]==4) DO	; 静态同步动作 2: ; \$AA_FXS[Y]==4 (识别出固定挡块): =>
FXST[Y]=30	; 固定挡块转矩: 30%
N12 IDS=3 WHENEVER (\$AA_FXS[Y]==1) DO	; 静态同步动作 3: ; \$AA_FXS[Y]==1 (轴成功到达固定挡块): =>
FXST[Y]=\$R0	; 固定挡块转矩: R 参数 \$R0 中的值
N13 IDS=4 WHENEVER ((R3==1) AND (\$AA_FXS[Y]==1)) DO	; 静态同步动作 4: ; R3==1: 请求取消 Y 的 FXS ; \$AA_FXS[Y]==1 (轴成功到达固定挡块): =>
FXS[Y]=0	; 取消 FXS
FA[Y]=1000 POS[Y]=0	; 定位运动 Y
N20 FXS[Y]=0 G0 G90 X0 Y0	; 设置初始设置: FXS 已取消, ; X 和 Y 位于起始位置
N30 RELEASE(Y)	; 使能同步动作中的运动 Y
...	
N60 GET(Y)	将轴 Y 重新纳入轨迹组合中
...	

说明

避免重复选择 FXS

为避免重复选择, 建议在激活 FXS 前询问系统变量 \$AA_FXS==0 或一个用户专用标志位。见上: 程序示例 N10

示例 2: 通过逐段同步动作运行到固定挡块

从后续程序段运行的某个位置开始激活“运行到固定挡块”功能

程序代码	注释
N10 G0 G90 X0	; 起始位置
N20 WHEN \$AA_IW[X] > 17 DO FXS[X]=1	; 同步动作: 轴的实际位置 X > 17 ; => 为 X 轴激活 FXS

17.3 示例

程序代码	注释
N30 G1 F200 X100	; X 轴的运行

说明

逐段同步动作会在后续主处理程序段中执行。

设定值切换

18.1 简要说明

18.1.1 简要说明

功能

如果仅使用一台电机驱动多个轴/主轴，则需要“设定值切换”功能。例如，对于铣头来说，主轴电机既用来驱动刀具又用来确定铣头的方向。

前提条件

为了执行设定值切换，必须满足下列条件：

- 参与设定值切换的所有轴都处于静止状态
- PROFINET 上没有生命符号故障和干扰等待处理
- 以下功能在具有驱动控制的轴中均无效：
 - 回参考点运行
 - 测量
 - 运行到固定挡块
 - 功能发生器
 - 星形/三角形切换
 - 参数组切换

位置控制回路

在设定值切换期间，驱动链和位置控制回路都断开。为避免不稳定，仅在静止状态和已删除控制器使能的情况下进行切换。

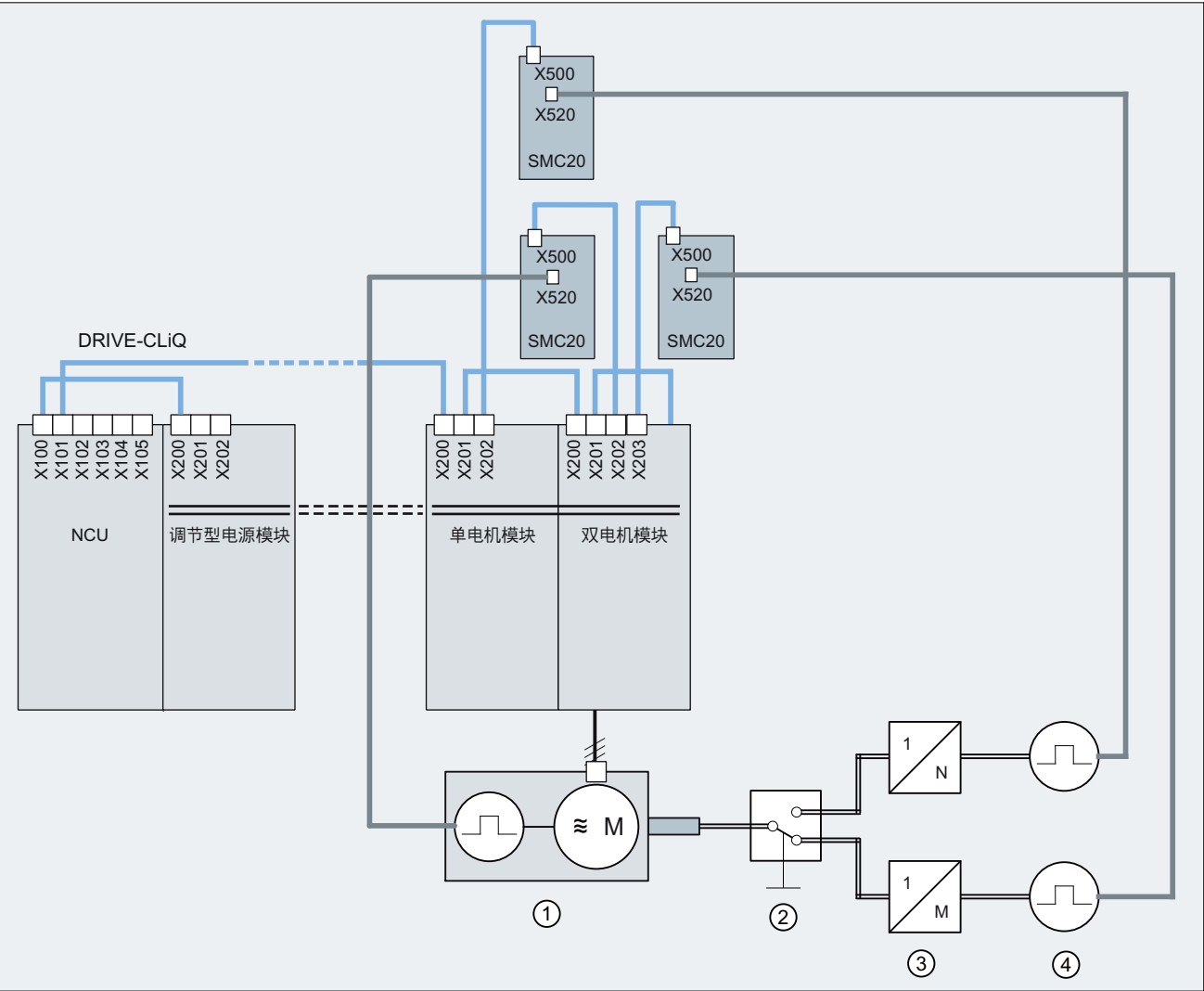
由于使用单驱动，在同一时间点只能闭合一个控制回路。没有驱动控制的轴利用开放式位置控制器和跟踪位置来运行。

说明

垂直轴

在切换期间，控制系统自动取消所有参与的轴的控制使能。没有现有驱动控制的轴不在控制范围内。因此，应为垂直轴实现制动控制。

示例



- ① 带编码器的电机
- ② 机械切换装置
- ③ 齿轮箱 1
- ④ 编码器 1（如用于铣头）
- ⑤ 齿轮箱 2
- ⑥ 编码器 2（如用于主轴）

图 18-1 带 2 个轴的设定值切换

18.2 调试

机床数据

设定值分配

为确定参与设定值切换的轴，要重复分配驱动的设定值通道。为此，必须使用该驱动的逻辑编号为每个轴预设以下机床数据：

MD30110 \$MA_CTRLOUT_MODULE_NR（设定值分配：模块编号）

说明

如果缺少该选项，会发出报警 26018。

编码器分配

编码器在机床数据中按轴分配：

MD30230 \$MA_ENC_INPUT_NR（实际值分配：驱动模块/测量电路板上的输入）

NC/PLC 接口信号

说明

访问权限

尽管将一个驱动分配给多个机床轴，但轴特定的 NC/PLC 接口信号的使用保持不变。因此，在访问参与设定值切换的各机床轴的 NC/PLC 接口信号方面，就需要在 PLC 用户程序中做明确的协调。

请求驱动控制

通过以下方式请求接收驱动控制：

<Axis>.drive.out.requestDriveCtrl（设定值切换：请求驱动控制）

驱动控制状态

驱动控制的当前状态通过以下方式显示：

<Axis>.drive.in.driveCtrlActive（设定值切换：驱动控制生效）

状态信号

在首次接收参与设定值切换的一个机床轴的驱动控制后（（设定值切换：切换）），所有参与的机床轴的以下状态信号始终以相同的方式显示：

```

<Axis>.drive.in.rapidStopActive;
<Axis>.drive.in.autonomousMotionActive;
<Axis>.drive.in.holdingBrakeReleased; <Axis>.drive.in.enabled;
<Axis>.drive.in.driveDataSetNumber;
<Axis>.drive.in.driveDataSetNumber;
<Axis>.drive.in.speedCtrlIntegratorDisabled;
<Axis>.drive.in.speedSetpointReached;
<Axis>.drive.in.speedAtSetpoint; <Axis>.drive.in.speedBelowMin;
<Axis>.drive.in.torqueBelowThreshold;
<Axis>.drive.in.speedBelowMin>;
<Axis>.drive.in.varSignalingFunc1Active>;
<Axis>.drive.in.varSignalingFunc2Active;
<Axis>.drive.in.varSignalingFunc3Active;
<Axis>.drive.in.esrDcVoltageAlarmActive;
<Axis>.drive.in.esrSpeedBelowMin

```

控制信号

以下控制信号只作用于当前具有驱动控制的机床轴

(<Axis>.drive.in.driveCtrlActive == 1 (设定值切换生效)):

```

<Axis>.drive.out.rapidStop;
<Axis>.drive.out.releaseHoldingBrake;
<Axis>.drive.out.driveDataSetSelection;
<Axis>.drive.out.motorDataSetSelection;
<Axis>.drive.out.motorChangeoverDone;
<Axis>.drive.out.disableSpeedCtrlIntegrator;
<Axis>.drive.out.enablePulses

```

控制器使能

伺服使能 <Axis>.basic.out.enableCtrl 只在当前具有驱动控制的机床轴中有效

(<Axis>.drive.in.driveCtrlActive == 1 (设定值切换状态))

跟踪方式

在参与设定值切换的机床轴中，**无**驱动控制的机床轴始终在控制系统内部被切换到跟踪方式 (`<Axis>.basic.in.followUpModeActive == 0`)。

说明**唯一性**

在任何时间点都始终应该有且只有一个轴具有驱动控制
`<Axis>.drive.out.requestDriveCtrl == 1`
 和 `<Axis>.drive.in.driveCtrlActive == 1`。

特殊情况**同时有多个接收请求**

如果同时有多个接收请求，则不会进行切换。对驱动的控制仍由最后使用的轴负责。缺少接收请求会导致同样的结果。

引导启动中缺少接收请求

如果控制系统的引导启动中缺少接收请求，则在**控制系统内部**将驱动控制分配给找到的第一个具有相同逻辑驱动编号的机床轴。

```
MD30110 $MA_CTRLOUT_MODULE_NR[0,AX1] = 1
```

```
MD30110 $MA_CTRLOUT_MODULE_NR[0,AX2] = 2
```

```
MD30110 $MA_CTRLOUT_MODULE_NR[0,AX3] = 3
```

```
MD30110 $MA_CTRLOUT_MODULE_NR[0,AX4] = 4 ; 设定值切换: 第 1 轴
```

```
MD30110 $MA_CTRLOUT_MODULE_NR[0,AX5] = 4 ; 设定值切换: 第 2 轴
```

为了能够使驱动随轴移行，必须事先使请求驱动控制

(`<Axis>.drive.out.requestDriveCtrl`) 的 NC/PLC 接口信号置位。

PLC 信号**PLC → NC**

Basic Program Plus	Basic Program	
<code><Axis>.basic.out.enableCtrl</code>	<code>LBP_Axis*.A_ContrEnable</code>	DB31, ...DBX2.1
<code><Axis>.drive.out.requestDriveCtrl</code>	<code>LBP_Axis*.A_ChangSetpOutAss0</code>	DB31, ...DBX24.5
<code><Axis>.drive.out.rapidStop</code>	-	DB31, ...DBB20.1

18.2 调试

Basic Program Plus	Basic Program	
<Axis>.drive.out.releaseHoldingBrake	LBP_Axis*.A_HoldBrakeToOpen	DB31, ...DBB20.5
<Axis>.drive.out.driveDataSetSelection	LBP_Axis*.A_ParA..C; A_MotA..B	DB31, ...DBB21.0.. 2
<Axis>.drive.out.motorDataSetSelection	-	DB31, ...DBB21.3.. 4
<Axis>.drive.out.motorChangeoverDone	LBP_Axis*.A_MotOK	DB31, ...DBB21.5
<Axis>.drive.out.disableSpeedCtrlIntegrator	LBP_Axis*.A_IntegratDisable	DB31, ...DBB21.6
<Axis>.drive.out.enablePulses	LBP_Axis*.A_PulseEnable	DB31, ...DBB21.7

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.followUpModeActive	LBP_Axis*.E_Follow_upMode	DB31, ...DBX61.3
<Axis>.drive.in.rapidStopActive	LBP_Axis*.E_RUEncDisabl	DB31, ...DBB92.1
<Axis>.drive.in.autonomousMotionActive	LBP_Axis*.E_RLIact	DB31, ...DBB92.4
<Axis>.drive.in.holdingBrakeReleased	LBP_Axis*.E_HoldBrakeOpened	DB31, ...DBB92.5
<Axis>.drive.in.enabled	LBP_Axis*.E_DriveRunEn	DB31, ...DBB92.7
<Axis>.drive.in.driveDataSetNumber	LBP_Axis*.E_ParA..C	DB31, ...DBB93.0.. 2
<Axis>.drive.in.motorDataSetNumber	LBP_Axis*.E_MotA..B	DB31, ...DBB93.3.. 4
<Axis>.drive.in.speedCtrlIntegratorDisabled	LBP_Axis*.E_IntegratDisable	DB31, ...DBB93.6
<Axis>.drive.in.speedSetpointReached	LBP_Axis*.E_RUComplete	DB31, ...DBB94.2
<Axis>.drive.in.speedAtSetpoint	LBP_Axis*.E_NactNset	DB31, ...DBB94.6
<Axis>.drive.in.speedBelowMin	LBP_Axis*.E_NactNmin	DB31, ...DBB94.4
<Axis>.drive.in.torqueBelowThreshold	LBP_Axis*.E_MdMdx	DB31, ...DBB94.3
<Axis>.drive.in.speedBelowMin	LBP_Axis*.E_NactNmin	DB31, ...DBB94.4
<Axis>.drive.in.varSignalingFunc1Active	LBP_Axis*.E_Message	DB31, ...DBB94.7
<Axis>.drive.in.varSignalingFunc2Active	LBP_Axis*.E_Message2	DB31, ...DBB95.4
<Axis>.drive.in.varSignalingFunc3Active	LBP_Axis*.E_Message3	DB31, ...DBB95.5
<Axis>.drive.in.esrDcVoltageAlarmActive	LBP_Axis*.E_UDClessAlThreshold	DB31, ...DBB95.1
<Axis>.drive.in.esrSpeedBelowMin	-	DB31, ...DBB95.3

Basic Program Plus	Basic Proagram	
<Axis>.drive.in.driveCtrlActive	LBP_Chan*.E_BitValCtrlOutChang	DB31, ...DBX96.5
<Axis>.basic.in.fixedStopReached	LBP_Chan*.E_FixedStop	DB31, ...DBX62.5

18.3 流程图

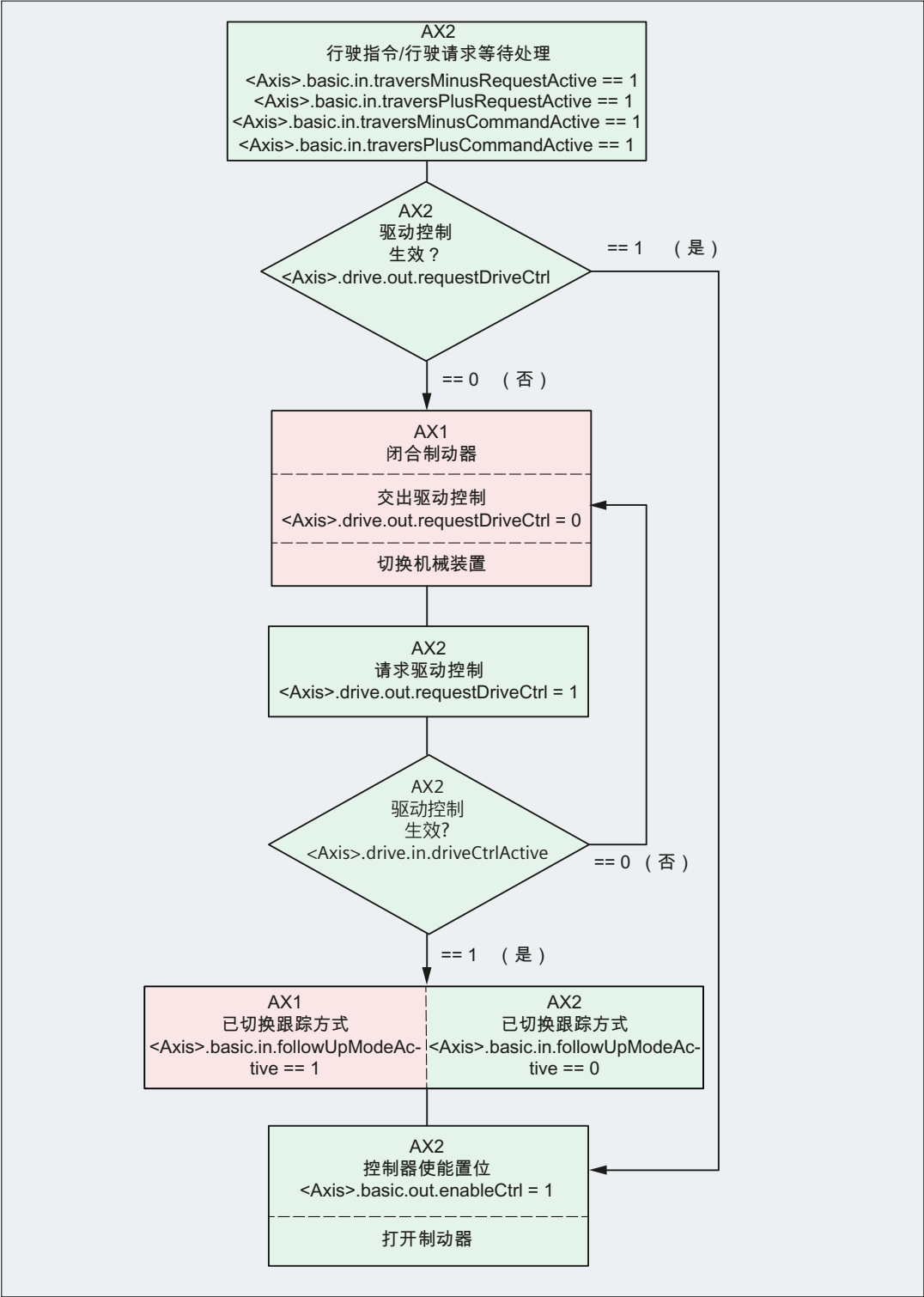


图 18-2 从机床轴 AX1 到 AX2 的设定值切换流程

PLC 信号

PLC → NC

Basic Program Plus	Basic Proogram	
<Axis>.basic.out.enableCtrl	LBP_Axis*.A_ContrEnable	DB31, ...DBX2.1
<Axis>.drive.out.requestDriveCtrl	LBP_Axis*.A_ChangSetpOutAss0	DB31, ...DBX24.5

NC → PLC

Basic Program Plus	Basic Proogram	
<Axis>.basic.in.followUpModeActive	LBP_Axis*.E_Follow_upMode	DB31, ...DBX61.3
<Axis>.basic.in.fixedStopReached	LBP_Axis*.E_FixedStop	DB31, ...DBX62.5
<Axis>.basic.in.traversMinusRequestActive	LBP_Axis*.E_TReqMinus	DB31, ...DBX64.4
<Axis>.basic.in.traversPlusRequestActive	LBP_Axis*.E_TReqPlus	DB31, ...DBX64.5
<Axis>.basic.in.traversMinusCommandActive	LBP_Axis*.E_TCMinus	DB31, ...DBX64.6
<Axis>.basic.in.traversPlusCommandActive	LBP_Axis*.E_TCPlus	DB31, ...DBX64.7

18.4 边界条件

报警

仅驱动控制生效的轴才会显示驱动报警。

位置控制回路

在设定值切换期间，驱动链和位置控制回路都断开。为避免不稳定，仅在静止状态和已删除控制器使能的情况下进行切换。

由于使用单驱动，在同一时间点只能闭合一个控制回路。没有驱动控制的轴利用开放式位置控制器和跟踪位置来运行。

参考点

使用负载侧编码器时，轴特定的参考点不受设定值切换的影响。

相反，如果负载侧位置是从电机编码器得出的，则在设定值切换后，负载的机械参考可能会丢失。这样的轴必须在每次设定值切换后重新寻参。

18.4 边界条件

参数组切换

如果存在以下状态之一，则不执行设定值转换：

- 两个机床轴之一的参数组切换未完成。
- 两个机床轴的参数组不同。

说明

如果由于边界条件而未执行设定值切换，则**不会**显示信息。

“驻车”状态

如果一个轴处于“驻车”状态，则只有在该轴具有驱动控制时才能再次取消该状态。

驱动服务显示

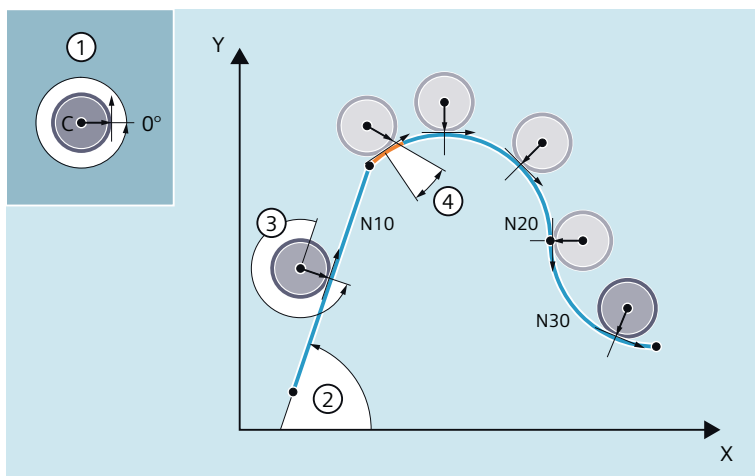
HMI 诊断画面“驱动服务显示”**不**考虑机床轴和驱动之间变化的分配。

通过 SinuCom NC 调试

使用“SinuCom NC”调试工具，设定值切换只能通过专家列表进行调试。

切向控制

“切向控制”功能将作为跟随轴的回转轴与作为引导轴的两个几何轴耦合，使得跟随轴的方向是引导轴的轨迹切线的函数。



- ① 跟随轴/回转轴 C 的初始位置和正旋转方向
- ② 加工平面 X/Y 中轨迹切线的角度
- ③ 偏移角 = 270° 或 -90°
- ④ 平滑距离和瞬时角度误差

在图中，确定轨迹的引导轴是几何轴 X 和 Y，回转轴 C 是切向耦合的跟随轴。回转轴是以与轨迹切线成 270° 的偏移角来跟踪的，该偏移角可编程。在从 N10 到 N20 的程序段过渡处，轮廓包含一个不稳定的过渡或拐角。由于动态响应过低，跟随轴无法在一个轨迹范围（蓝色）内跟随轨迹切线。然而，跟随轴的动力学响应足以在圆程序段 N20 和 N30 的进一步走向中精确地跟随轨迹切线。

属性

切向耦合具有以下特点：

- 切向耦合的引导轴必须为两个几何轴。
- 切向耦合的跟随轴必须为一个回转轴。
- 切向耦合可以在 NC 程序中定义、开启、关闭和删除。
- 切向耦合是针对基准坐标系和工件坐标系定义的。
- 轨迹切线的正方向等于轨迹的移行方向。

- 回转轴与轨迹切线之间的偏移角为 0° 时，表示 0° 方向。
- 切向耦合的轴必须是同一通道的通道轴。
- 跟随轴的位置可以作为进一步坐标转换的输入值。
- 切向耦合仅在自动和 MDI 运行方式中生效。

应用实例

- 步冲加工中对可旋转刀具的切向调节。
- 使用带锯时对工件定向进行跟踪。
- 按照砂轮调整修整刀具。
- 玻璃或纸张加工用的小切削轮的调整。
- 在 5 轴焊接中切向送入焊丝。

拐角特性

如果引导轴描述的轮廓具有不连续的程序段过渡或拐角，则可以在以下拐角特性之间进行选择：

- 回转轴的动态响应不会对引导轴产生影响。
- 在计划引导轴的轨迹时会连同可编程参数“平滑距离”和“角度公差”一起考虑回转轴的动态响应。
- 引导轴在拐角前停止，跟随轴在自动生成的中间程序段中重新取向。
拐角识别取决于机床数据 MD37400 \$MA_EPS_TLIFT_TANG_STEP（用于识别拐角的切线角度）的值。

19.1 调试

机床数据

拐角识别 TLIFT()

与拐角识别功能 TLIFT() (页 890) 结合使用时，若在编程的引导轴轨迹中的某个中断位置处，跟随轴切线角度的变化大于机床数据中指定的以下值，则将该中断识别为拐角：

MD37400 \$MA_EPS_TLIFT_TANG_STEP[<跟随轴>] = <切线变化角度>

可参数设置的偏移角

通过机床数据为跟随轴指定可参数设置的偏移角：

机床数据 MD37402 \$MA_TANG_OFFSET[<跟随轴>] = <可参数设置的偏移角>

跟随轴的偏移角是指，切线耦合激活时跟随轴的零位与编程的引导轴轨迹的轨迹切线之间的角度。

有效偏移角是机床数据中参数设置的偏移角与切向耦合开启时使用 TANGON() (页 891) 编程的偏移角之和：

程序段搜索设置

该机床数据用于选择在程序段搜索期间或之后切向耦合开启时切向轴运行到其位置的速度：

- 快速移动速度 (G0)
- 编程的速度 (G1 F...)
- 设定数据 SD42121 \$SC_AX_ADJUST_FEED ≠ 0

MD11450 \$MN_SEARCH_RUN_MODE, 位 7

另见：边界条件 (页 895)

系统变量

程序段类型 \$AC_BLOCKTYPE

通过系统变量 \$AC_BLOCKTYPE 可确定当前程序段是否是 TLIFT() 创建的中间程序段。如果符合以下情况，则当前程序段是 TLIFT() 创建的中间程序段：

\$AC_BLOCKTYPE == 6

19.2 编程

19.2.1 定义耦合 (TANG)

通过预定义程序 TANG(...) 定义一根回转轴（跟随轴）与两根几何轴（引导轴）之间的切向耦合。此时，跟随轴连续对准引导轴的轨迹切线。

说明

耦合系数

不必精确编程耦合系数 1。
通过将耦合系数 -1 来旋转切向轴的方向。

句法

TANG(<FAx>, <LAx1>, <LAx2>, <CoupFac>, <CoordSys>, <OptMode>)

含义

TANG (...)	定义切向耦合		
<FAx>	跟随轴（回转轴）的轴名称		
	数据类型：	AXIS	
	值域：	通道轴名称	
<LAx1>	引导轴（几何轴）的轴名称 ¹⁾		
<LAx2>	数据类型：	AXIS	
	值域：	通道的几何轴	
<CoupFac>	用于修改引导轴轨迹切线的跟随轴角度变化的系数 n： 角度变化 _{跟随轴} = 角度变化 _{轨迹切线} * n		
	数据类型：	REAL	
	缺省值：	1.0	
<CoordSys>	生效坐标系 ²⁾		
	数据类型：	CHAR	
	值：	"B"	基本坐标系（缺省值）
		"W"	工件坐标系（不可用）

<OptMode>	优化模式		
	数据类型:	CHAR	
	值:	"S"	标准（缺省值） 回转轴的动态响应不会对引导轴产生影响。如果回转轴的动态响应高于跟踪所需的动态响应，该运行则足够精确。如果回转轴的动态响应还不足以修改轨迹切线，回转轴沿着非指定的平滑距离的校准则会与设定校准产生偏差。
		"P"	在计划引导轴的轨迹时会考虑回转轴的动态响应。 为此，在通过 TANGON() 激活切向耦合时，必须给定两个附加参数： • 平滑距离 • 角度公差 参见章节“激活耦合 (TANGON) (页 891)” 提示 结合运动转换，建议使用优化方式“P”。
提示 缺省值不必精确编程。			
1) 提示 作为切向耦合的引导轴，必须以机床的初始位置为基准、在机床坐标系 (MCS) 中按编程的轨迹运行几何轴。如果在带旋转头的铣床上使用旋转循环 CYCLE800，则根据循环配置在工件坐标系中通过几何轴 X 和 Y 进行插补。但必须通过在机床坐标系 (MCS) 中按编程的轨迹运行的几何轴将切向耦合定义为引导轴。为此，作为引导轴，几何轴必须在机床处于非旋转状态下使用。			
2) 提示 基准坐标系 (BCS) 不可相对于 MCS 旋转。如果通过指令 ROT 或通过旋转循环 CYCLE800 旋转 BCS，切向控制则无法再准确运行。			

19.2.2 激活中间程序段生成 (TLIFT)

如果跟随轴在编程的引导轴轨迹的某个位置上的切线变化超出了机床数据 MD37400 \$MA_EPS_TLIFT_TANG_STEP 中设置的限值，其他轨迹设计则须依据设置的拐角特性进行。如果不使用预定义程序 TLIFT(...)，则根据通过 TANG(...) (页 888) 和 TANGON(...) (页 891) 编程的平滑特性运行。

激活中间程序段生成

通过编程 TLIFT(...) 和 TANG(...)，从预处理开始，在该轨迹位置上发现拐角时系统会添加一个由控制系统自动生成的中间程序段。

执行程序时，引导轴在到达中间程序段时停止。在中间程序段中，具有最大轴动态响应的跟随轴按下一程序段的轨迹切线的方向旋转。之后，引导轴再次在编程的轨迹上运行。

取消中间程序段生成

为取消中间程序段生成，必须借助 TANG(...) 重新定义切向耦合，但不包括下一次借助 TLIFT(...) 激活中间程序段生成。

句法

TLIFT (<FAx>)

含义

TLIFT (. . .)	激活带中间程序段计算的拐角识别	
<FAx>	跟随轴（回转轴）的轴名称	
	数据类型:	AXIS
	值域:	通道轴名称

跟随轴的转速

轨迹轴

如果跟随轴在激活切向耦合前已经作为轨迹轴运行，则作为轨迹轴在中间程序段中旋转。

通过 FGREF[<Ax>]=0.001 设定参考半径，以参数设置的最大轴速度进行旋转运动：

MD32000 \$MA_MAX_AX_VELO[<跟随轴>]

定位轴

如果跟随轴在激活切向耦合前尚未作为轨迹轴运行，则作为定位轴在中间程序段中旋转。

此时，以设置的定位轴速度旋转：

```
MD32060 $MA_POS_AX_VELO[<跟随轴>]
```

19.2.3 激活耦合 (TANGON)

通过预定义程序 TANGON(...) 激活之前通过 TANG(...) (页 888) 定义的切向耦合。之后，在下次运行引导轴时，跟随轴会连续对准引导轴的轨迹切线。

跟随轴的角度

跟随轴同轨迹切线所形成的角度取决于 TANG(...) 中给定的传动比、机床数据 MD37402 \$MA_TANG_OFFSET 中设置的偏移角和 TANGON(...) 时给定的额外生效的偏移角。

优化 “P”

如果在定义切向耦合 (TANG(...)) 时给定值 “P” 作为优化参数，则须在激活耦合时给定参数 “平滑距离” 和参数 “角度公差” （可选）。

如果给定值 0 作为角度公差，则只有参数 “平滑距离” 生效。

如果给定一个大于 0 的值作为角度公差，有效平滑距离由设置的最小平滑距离和以设置的角度公差为基准的平滑距离得出。

如果跟随轴的动力学响应不满足设定条件，则相应地减小跟随轴的轨迹速度。

句法

```
TANGON(<FAx>, <OffsetAngle>, <MaxRoundingPath>, <MaxAngTol>)
```

含义

TANGON (...)	激活切向控制	
<FAx>	跟随轴（回转轴）的轴名称	
	数据类型:	AXIS
	值域:	通道轴名称
<OffsetAngle>	跟随轴与轨迹切线之间的偏移角 参考点为回转轴零点。	
	数据类型:	REAL
<MaxRoundingPath>	允许的最大平滑距离 如果平滑距离就动态响应条件来说较大，则减小引导轴的轨迹速度。	
	数据类型:	REAL

<MaxAngTol>	跟随轴零位与轨迹切线之间允许的最大角度公差	
	数据类型:	REAL

19.2.4 取消耦合 (TANGOF)

通过预定义程序 TANGOF(...) 取消通过 TANG(...) (页 888) 定义和通过 TANGON(...) (页 891) 激活的切向耦合。之后，跟随轴不再对准引导轴的轨迹切线。取消后，跟随轴在引导轴上的耦合仍然保留，以避免触发以下功能：

- 平面切换
- 几何轴切换
- 为跟随轴定义新的切向耦合

只有在通过 TANGDEL(...) (页 892) 删除耦合后，才能完全取消跟随轴在引导轴上的耦合。

编程

TANGOF (<FAx>)

含义

TANGOF (. . .)	取消切向耦合	
<FAx>	跟随轴（回转轴）的轴名称	
	数据类型:	AXIS
	值域:	通道轴名称

19.2.5 删除耦合 (TANGDEL)

在通过 TANGOF(...) (页 892) 取消切向耦合后，通过 TANG(...) (页 888) 定义的切向耦合仍然保留。现有切向耦合之后仍会避免触发以下功能：

- 平面切换
- 几何轴切换
- 为跟随轴定义新的切向耦合

在通过 TANGOF(...) 取消切向耦合后，通过预定义程序 TANGDEL(...) 删除现有切向耦合。

句法

TANGDEL (<FAx>)

含义

TANGDEL (. . .)	删除通过 TANG() 定义的切向耦合	
	生效方式:	逐段式
<FAx>	应删除切向耦合的跟随轴的轴名称	
	数据类型:	AXIS
	值域:	通道轴名称

示例

引导轴切换

在通过另一根引导轴为跟随轴定义新的切向耦合前，必须先删除现有切向耦合。

程序代码	注释
N10 TANG (A, X, Y, 1)	; 定义跟随轴 A 的切向耦合: A 到 X 和 Y
N20 TANGON (A)	; 激活跟随轴 A 的切向耦合
N30 X10 Y20	
...	
N80 TANGOF (A)	; 取消跟随轴 A 的切向耦合
N90 TANGDEL (A)	; 删除跟随轴 A 的切向耦合
...	
N120 TANG (A, X, Z)	; 为跟随轴 A 定义新的切向耦合
N130 TANGON (A)	; 为跟随轴 A 激活新的切向耦合
...	

几何轴切换

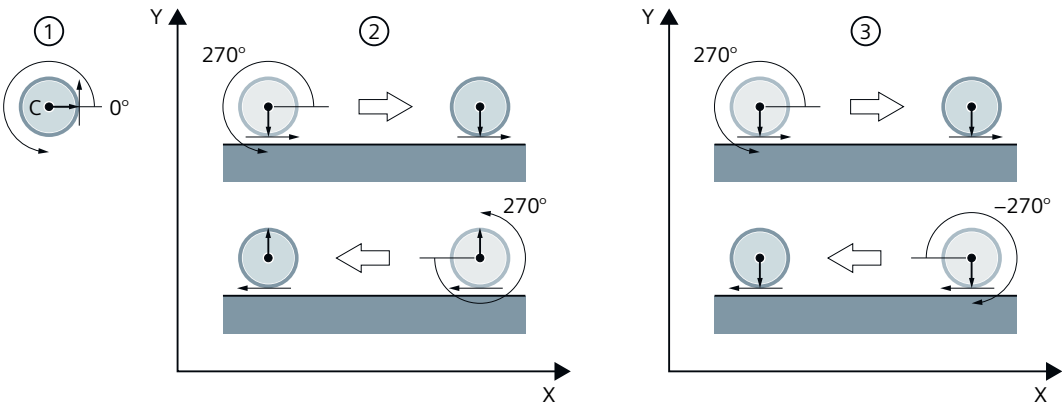
在对现有耦合进行几何轴切换前，必须先删除耦合。

程序代码	注释
N10 GEOAX (2, Y1)	; 第 2 几何轴 = 机床轴 Y1
N20 TANG (A, X, Y)	; 定义跟随轴 A 的切向耦合
N30 TANGON (A, 90)	; 激活跟随轴 A 的切向耦合
N40 G2 F8000 X0 Y0 I0 J50	; 运动程序段
N50 TANGOF (A)	; 取消跟随轴 A 的切向耦合
N60 TANGDEL (A)	; 删除跟随轴 A 的切向耦合
N70 GEOAX (2, Y2)	; 第 2 几何轴 = 机床轴 Y2
N80 TANG (A, X, Y)	; 为跟随轴 A 定义新的切向耦合
N90 TANGON (A, 90)	; 为跟随轴 A 激活新的切向耦合

程序代码	注释
...	

19.3 极限角度

交替移行运动时，轨迹切线的方向在换向点跳跃 180°。跟随轴的方向也因此跳跃 180°。该特性通常没有意义（见图，②）。相反，在返回运动中，跟随轴应以与去时运动相同的方向移行（见图，③）。



- ① 跟随轴/回转轴 C 的初始位置和正旋转方向
- ② 270° 度偏移角相同的往复运动
- ③ 270° 和 -270° 度偏移角交替的往复运动

避免换向跳跃

必须通过加工区域限位 G25（下限）和 G26（上限）基于基准坐标系为跟随轴指定位置值的最小值和最大值。然后必须根据加工情况使用 WALIMON / WALIMOF 指令开启/关闭跟随轴的加工区域限位。在轨迹换向的时间点，加工区域限制必须生效。

影响

如果在以当前偏移角移行跟随轴时，轴位置移动到生效的工作范围限位之外，则控制系统会检查：在使用负偏移角时，跟随轴是否会再次在允许的工作范围内移动。在这种情况下，将以负偏移角移行跟随轴。

更多信息

编程手册之基本原理：补充指令 > 加工区域限制 > BCS 中的加工区域限制 (G25/G26, WALIMON, WALIMOF)

19.4 边界条件

切向耦合生效时的程序段搜索

我们建议，在切向耦合生效时，仅使用在 TANG() (页 888) 下可编程的优化方式“P”和搜索类型 2（“在轮廓上进行计算的程序段搜索”）或 5（“在程序测试模式 (SERUPRO) 下进行计算的程序段搜索”）进行程序段搜索。

跟随轴的明确定位

如果明确地移行生效的切向耦合的跟随轴，则在此期间移行的行程对编程的偏移角产生附加的影响。

19.5 示例

示例 1：定义并开启切向耦合

程序代码	注释
； 定义切向耦合：	
； - 跟随轴 C，引导轴 X 和 Y	
； - 耦合系数：1.0	
； - 坐标系：BCS	
； - 优化方式："P"	
N10 TANG(C,X,Y,1,"B","P")	； 简化：TANG(C,X,Y,,, "P")
； 开启切向耦合：	
； - 偏移角：90°	
N20 TANGON(C,90)	
...	

示例 2：平面切换

程序代码	注释
； 定义切向耦合：	
； - 跟随轴 A，引导轴 X 和 Y	
； - 耦合系数：1.0	
； - 坐标系：BCS	
； - 优化方式："S"	
N10 TANG(A,X,Y)	
； 开启切向耦合：	
； - 偏移角：0°	
N20 TANGON(A)	

19.5 示例

程序代码	注释
N30 X10 Y20	; 引导轴移行
...	
N80 TANGOF (A)	; 关闭切向耦合
N90 TANGDEL (A)	; 删除切向耦合
...	
; 定义新切向耦合:	
; - 跟随轴 A, 引导轴 X 和 Z	
; - 耦合系数: 1.0	
; - 坐标系: BCS	
; - 优化方式: "S"	
TANG (A, X, Z)	
TANGON (A)	
...	

示例 3：几何轴切换

程序代码	注释
N10 GEOAX (2, Y1)	; 第 2 几何轴: Y1
; 定义切向耦合:	
; - 跟随轴 A, 引导轴 X 和 Y	
; - 耦合系数: 1.0	
; - 坐标系: BCS	
; - 优化方式: "S"	
N20 TANG (A, X, Y)	
; 开启切向耦合:	
; - 偏移角: 90°	
N30 TANGON (A, 90)	
N40 G2 F8000 X0 Y0 I0 J50	; 圆移行
N50 TANGOF (A)	; 关闭切向耦合
N60 TANGDEL (A)	; 删除切向耦合
N70 GEOAX (2, Y2)	; 第 2 几何轴: Y2
; 定义新切向耦合:	
; - 跟随轴 A, 引导轴 X 和 Y	
; - 耦合系数: 1.0	
; - 坐标系: BCS	
; - 优化方式: "S"	
N80 TANG (A, X, Y)	
; 开启切向耦合:	
; - 偏移角: 90°	
N90 TANGON (A, 90)	
...	

示例 4：优化方式“P”

程序代码	注释
...	
N100 G0 C0	
N110 G1 X1000 Y500 F50000	
N120 TRAORI	; 开启坐标转换
N130 G642	; 开启精磨
N171 TRANS X50 Y50	; 工件坐标系零点偏移
; 定义切向耦合:	
; - 跟随轴 C, 引导轴 X 和 Y	
; - 耦合系数: 1.0	
; - 坐标系: BCS	
; - 优化方式: "P"	
N180 TANG(C,X,Y,1,, "P")	
; 开启切向耦合:	
; - 偏移角: 0°	
; - 平滑距离 5.0 mm	
; - 角度公差 2.0°	
N190 TANGON(C,0,5.0,2.0)	
N210 G1 X1310 Y500	
N215 G1 X1420 Y500	
N220 G3 X1500 Y580 I=AC(1420) J=AC(580)	
N230 G1 X1500 Y760	
N240 G3 X1360 Y900 I=AC(1360) J=AC(760)	
N250 G1 X1000 Y900	
N280 TANGOF(C)	; 关闭切向耦合
N290 TRAFOOF	; 关闭坐标转换
...	

示例 5：空间内的拐角

程序代码	; 注释
; 定义切向耦合:	
; - 跟随轴 A, 引导轴 X 和 Y	
; - 耦合系数: 1.0	
; - 坐标系: BCS	
; - 优化方式: "S" (标准)	
TANG(A,X,Y,1.0, "B")	
TLIFT(A)	; 开启中间程序段生成
G1 G641 X0 Y0 Z0 A0	; 逼近初始位置
; 开启切向耦合:	
; - 偏移角: 0°	
TANGON(A,0)	

19.5 示例

程序代码	; 注释
N4 X10	; 引导轴 x 移行
N5 Z10	; 几何轴 z 移行
N6 Y10	; 引导轴 y 移行
M30	

在程序段 N4 和 N6 中移行几何轴会导致不连续的走向，即在切向耦合的引导轴的轮廓中出现拐角。通过编程 TLIFT()，在 N6 之前插入一个中间程序段，在该程序段中，将跟随轴 A 沿 Y 轴的切线方向旋转 90°。然后才在 N6 中移行 Y 轴。

往复

20.1 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。

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。该进给率可和主轴、回转轴或主轴相关（旋转进给率）。

更多信息

进给率 (页 201)

- 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。

20.2 异步往复运动

20.2.1 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，与程序中当前生效的 G 指令无关。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。
- 同时可以有多个往复轴，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。
- 往复轴可为坐标转换中的输入轴（例如倾斜轴，参见“坐标转换”功能手册，“运动坐标转换”章节）。
- 往复轴可以设为龙门轴和主-从轴上的主动轴。

更多信息

龙门轴 (页 357)

- 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”）。

20.2.2 异步往复运动的设置选项

设定数据

往复需运动所需的设定数据可通过 NC 零件程序中的专用语言指令、通过 HMI 和/或通过 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“定位轴 (页 277)”章节）。
- 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。

更多信息

进给率 (页 201)

- 速度/位置可由手轮加以控制，见章节“手动运行 (页 743)”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。

- 在往复运动期间，换向点的位置还可通过手动运行（手轮、JOG 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 NC/PLC 接口信号 <Chan>.basic.out.handwheelOffsetDrf（激活 DRF）生效且同时要修改换向点，则手轮信号不仅会用于 DRF 偏移，还会用于换向点偏移，换句话说：换向点偏移了两倍的距离。

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 20-1 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 20-2 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。

更多信息

编程手册

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 G04 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 20-3 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“起点和换向点重合时的往复运动（起点 = 换向点 1）(页 931)”章节。

编程示例

在章节“异步往复运动程序示例 (页 924)”举例说明了异步往复运动中重要的一些程序指令。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Chan>.basic.out.handwheelOffsetDrf	LBP_Chan*.A_DRF	DB21, ...DBX0.3

20.2.3 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：

- <Axis>.spindle.out.stopAtNextReversalPoint（在下一个换向点上停止）
- <Axis>.spindle.out.stop（沿制动斜坡停止）

PLC 信号

PLC → NC

Basic Program Plus	Basic Proagram	
<Axis>.spindle.out.stopAtNextReversalPoint	LBP_Axis*.A_StopRP	DB31, ...DBX28.5
<Axis>.spindle.out.stop	LBP_Axis*.A_Stop	DB31, ...DBX30.0

20.2.4 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→NC) <Axis>.spindle.out.plcCtrl (PLC 控制轴) 置 1 后，PLC 从 NC 接管对轴/主轴的控制。

关于 PLC 控制轴的相关信息参见“定位轴 (页 277)”章节。

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。适用规则：首先发出的移动指令优先。

表格 20-4 支持往复运动的运行方式

运行方式	往复运动
AUTO	是
MDI	是
MDA REPOS	是
MDA TEACHIN	是
MDA REF	否
JOG	是

运行方式	往复运动
JOG REF	否
JOG REPOS	是

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。

ASUB

执行异步子程序(ASUP)期间会继续往复运动。

PLC 信号

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.spindle.out.plcCtrl	LBP_Axis*.A_PLCAxis	DB31, ...DBX28.7

20.3 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。
下文仅介绍一些和往复运动相关的运动同步动作。

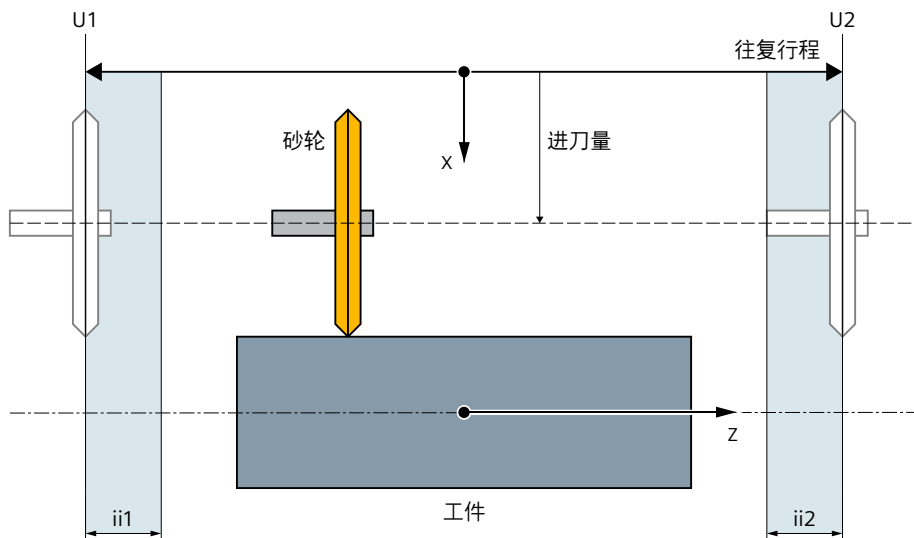
更多信息

功能手册之同步动作分册

功能

- 使用下列详细描述指令可以实现下列功能：
- 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 915)”。
 - 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 916)”。
 - 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 918)”。
 - 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 918)”。

5. 继续往复运动，见章节“继续往复运动 (页 919)”。
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 920)”。



U1 换向点 1

U2 换向点 2

ii1 换向区 1

ii2 换向区 2

图 20-1 往复轴、进给轴的布局

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 921)”）、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 901)”。

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。

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。

- 同步动作的条件（WHEN / WHENEVER 的频繁程度）
- 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

说明

通过将运动同步动作部分 (DO \$AA_OVR[<Axis>]=0) 中的轴向倍率置零可在满足对应的条件（WHEN / WHENEVER 的频繁程度）时停止往复轴和进给轴。
如果不再满足条件，倍率 100% 自动生效。

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。
同步动作的完整说明请见：

更多信息

功能手册之同步动作分册

示例

示例 1：换向点不变

针对运动同步，系统会使用预处理平面上的换向点 \$SA_OSCILL_REVERSE_POS。对应的设定数据被修改时，经过修改的值在程序中不生效。

程序代码	注释
...	
\$SA_OSCILL_REVERSE_POS1[Z]=-10	； 预处理变量：换向点 1
\$SA_OSCILL_REVERSE_POS2[Z]=10	； 预处理变量：换向点 2
G0 X0 Z0	
WAITP(Z)	
； 同步动作 1：低于换向点 1 ⇒	
； 通过倍率 0% 停止进给轴	
； 预处理变量使用换向点 1	
ID=1 WHENEVER \$AA_IM[Z] < \$SA_OSCILL_REVERSE_POS1[Z] DO \$AA_OVR[X]=0	
； 同步动作 2：超出换向点 2 ⇒	
； 通过倍率 0% 停止进给轴	
； 预处理变量使用换向点 2	
ID=2 WHENEVER \$AA_IM[Z] > \$SA_OSCILL_REVERSE_POS2[Z] DO \$AA_OVR[X]=0	
OS[Z]=1 FA[X]=1000 POS[X]=40	； 激活往复

程序代码	注释
OS[Z]=0	; 取消往复
M30	

示例 2：换向点改变

针对运动同步动作，系统会使用主运行变量 **\$\$AA_OSCILL_REVERSE_POSx**。对应的设定数据被修改时，经过修改的值在程序中**生效**。

程序代码	注释
...	
\$\$SA_OSCILL_REVERSE_POS1[Z]=-10	; 预处理变量：换向点 1
\$\$SA_OSCILL_REVERSE_POS2[Z]=10	; 预处理变量：换向点 2
G0 X0 Z0	
WAITP(Z)	
; 同步动作 1：低于换向点 1 ⇒	
; 通过倍率 0% 停止进给轴	
; 主运行变量 使用换向点 1	
ID=1 WHENEVER \$AA_IM[Z] < \$\$SA_OSCILL_REVERSE_POS1[Z] DO \$AA_OVR[X]=0	
; 同步动作 2：超出换向点 2 ⇒	
; 通过倍率 0% 停止进给轴	
; 主运行变量 使用换向点 2	
ID=2 WHENEVER \$AA_IM[Z] > \$\$SA_OSCILL_REVERSE_POS2[Z] DO \$AA_OVR[X]=0	
OS[Z]=1 FA[X]=1000 POS[X]=40	; 激活往复
OS[Z]=0	; 取消往复
M30	

20.3.1 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

```
换向点 1
WHENEVER $AA_IM[Z] <> $SA_OSCILL_REVERSE_POS1[Z]
DO $AA_OVR[X] = 0 $AA_OVR[Z] = 100
```

20.3 通过同步动作控制的往复运动

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

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 921)”。

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 921)”。

20.3.2 在换向区内作进给运动

换向区 1

功能

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

在往复轴进入换向区 1 后，再开始进给运动，见章节“图 20-1 往复轴、进给轴的布局 (页 913)”。

编程

```
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：换向区的大小（用户变量）

换向区 2

功能

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

在往复轴进入换向区 2 后，再开始进给运动，见章节“图 20-1 往复轴、进给轴的布局 (页 913)”。

编程

```
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 的大小（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 921)”。

对应关系

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 921)”。

参见

通过同步动作控制的往复运动 (页 912)

20.3.3 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

- 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 915)”和“在换向区内作进给运动 (页 916)”)

20.3.4 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。

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
```

系统变量的含义

- \$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

20.3.5 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。

20.3 通过同步动作控制的往复运动

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

20.3.6 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。

编程

```
WHENEVER $AA_DTEPW[进给轴]==0 DO $AC_MARKER[下标]=1
```

换向点 1

```
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

20.3.7 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。

20.3.8 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。

20.3.9 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 `<Axis>.spindle.out.externalReversal`（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 `<Axis>.spindle.in.externalReversalActive`（往复轴换向生效）报告给 PLC。

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 `<Axis>.spindle.out.activateReversalPoint`（修改换向点）。

PLC 输入信号 `<Axis>.spindle.out.selectReversalPoint`（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（参见“示例 (页 924)”章节）。

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“由外部触发往复轴换向时通过同步动作修改换向点 (页 934)”章节）。

PLC 信号

PLC → NC

Basic Program Plus	Basic Proogram	
<Axis>.spindle.out.externalReversal	LBP_Axis*.A_OscillAxExtRev	DB31, ...DBX28.0
<Axis>.spindle.out.selectReversalPoint	LBP_Axis*.A_SetRP	DB31, ...DBX28.3
<Axis>.spindle.out.activateReversalPoint	LBP_Axis*.A_ChangeRP	DB31, ...DBX28.4

NC → PLC

Basic Program Plus	Basic Proogram	
<Axis>.spindle.in.externalReversalActive	LBP_Axis*.E_OscillAxExtRev	DB31, ...DBX100.2

20.4 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。

20.5 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。

20.5.1 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。

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

程序代码	注释
	; 进给率为 500，目标位置为 15

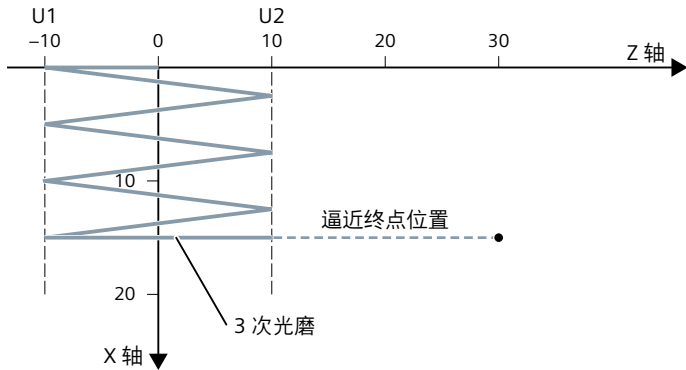


图 20-2 示例 1 中的往复运动和进给运动

20.5.2 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 912)”）。

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

程序代码	注释
	;
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
;	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)

程序代码	注释
;	
;	
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	
;	
; -----	
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

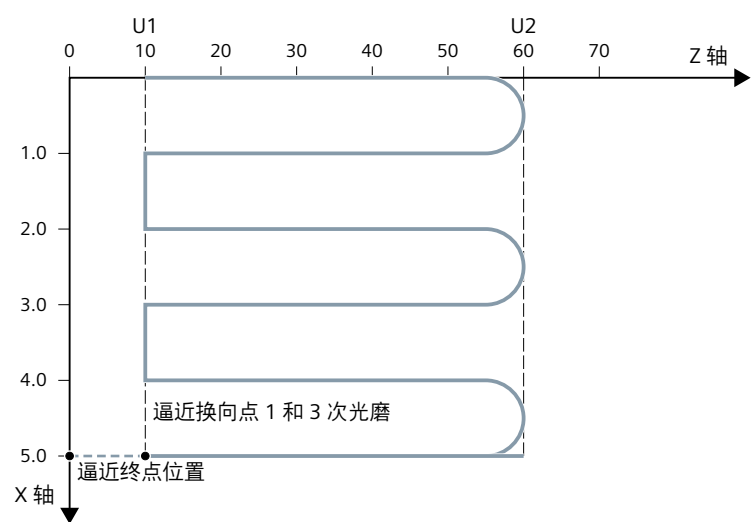


图 20-3 示例 1 中的往复运动和进给运动

20.5.3 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
;	
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

程序代码	注释
;	
; 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)
;	
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	

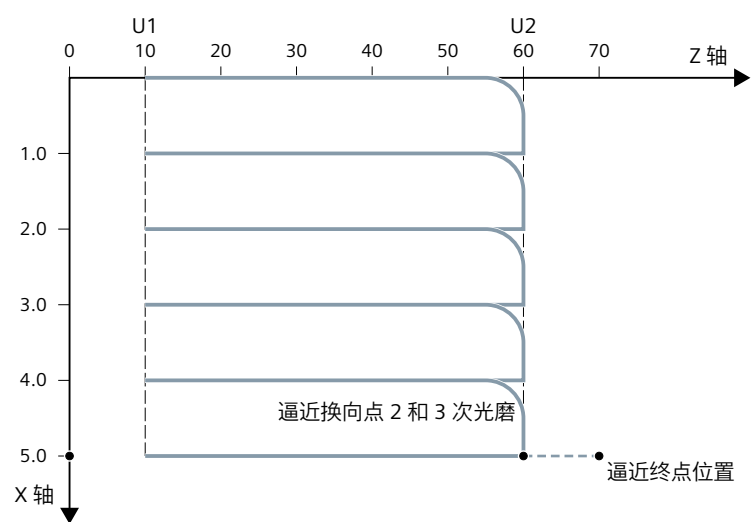


图 20-4 示例 2 中的往复运动和进给运动

20.5.4 示例：起点

20.5.4.1 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
	; 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。

20.5.4.2 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。

20.5.4.3 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
;	

程序代码	注释
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
;	
; 就	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	
;	
; 任何时候只要;	往复轴的当前坐标
;	等于换向点 1 的坐标
; 就	将往复轴的倍率设为
;	0, 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	
;	
; 任何时候只要	单次进给的剩余行程等于 0

程序代码	注释
;	
; 就	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	
;	
; 任何时候只要	下标为 0 的标志为 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！
WHENEVER \$AC_MARKER[0]==1 DO \$AA_OVR[X] = 0	
;	
; 任何时候只要	下标为 1 的标志为 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！ 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	
;	
; 任何时候只要	往复轴的当前坐标
;	等于换向点 1 的坐标
; 就	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)	
; 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。

20.5.5 示例：从外部触发往复轴换向

20.5.5.1 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	

软件挡块，行程开关信号

21.1 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 NC 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。

21.2 挡块信号和挡块位置

正负挡块和挡块范围的对应关系如下：

- 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

21.2 挡块信号和挡块位置

21.2.1 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：

- 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。

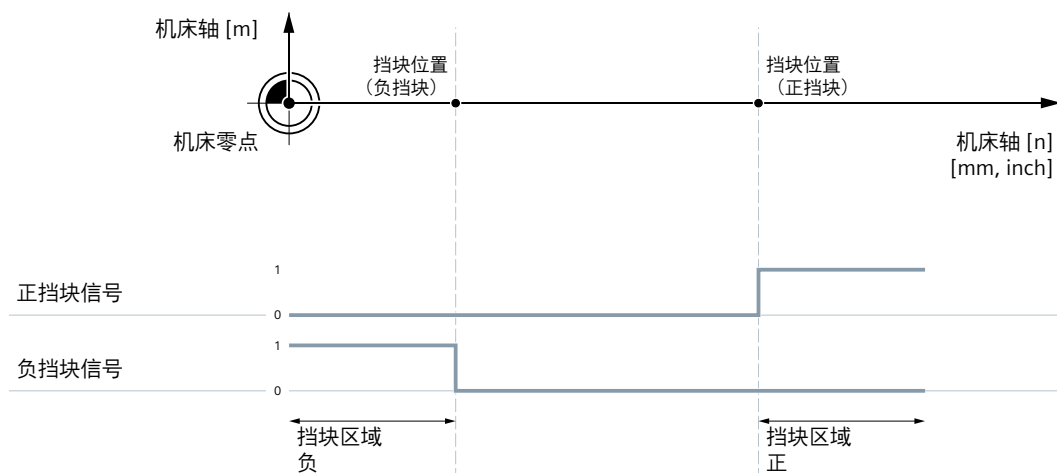


图 21-1 线性轴的软件挡块（负挡块 < 正挡块）

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。

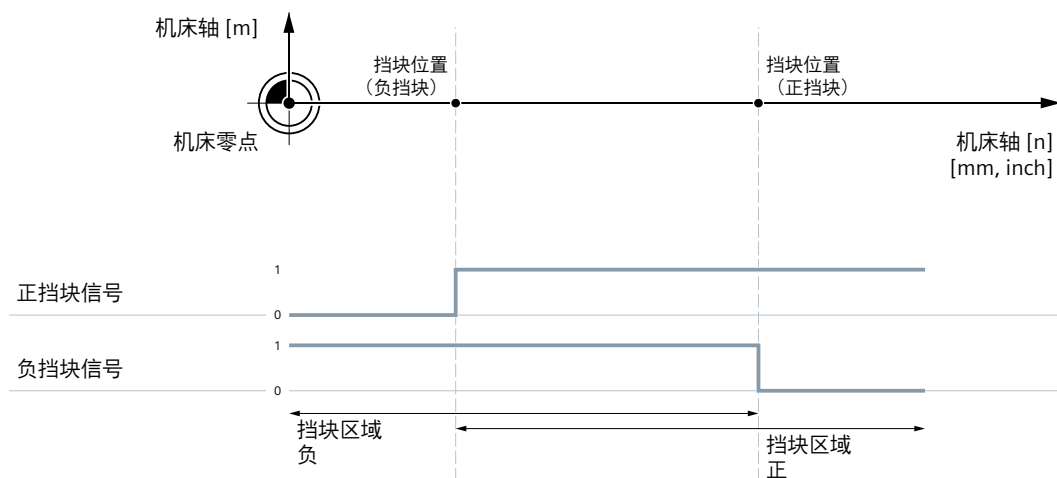


图 21-2 线性轴的软件挡块（正挡块 < 负挡块）

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

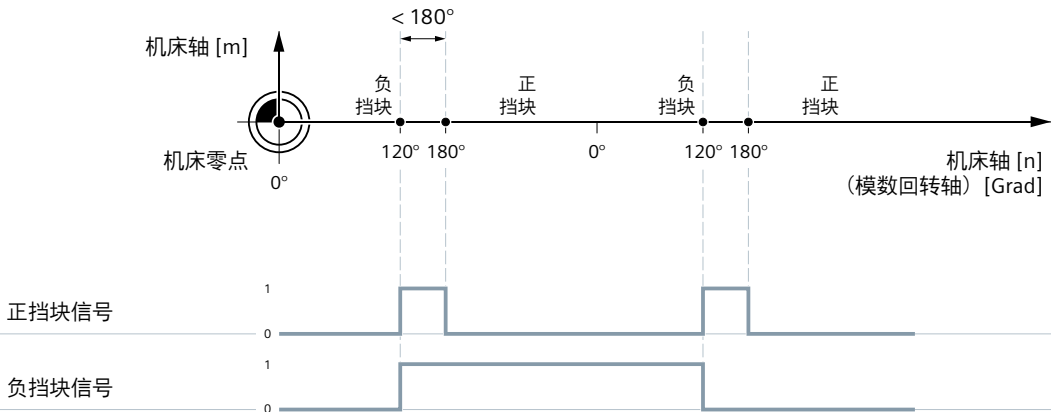


图 21-3 模数回转轴的软件挡块（正挡块 - 负挡块 < 180 度）

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 NC 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。

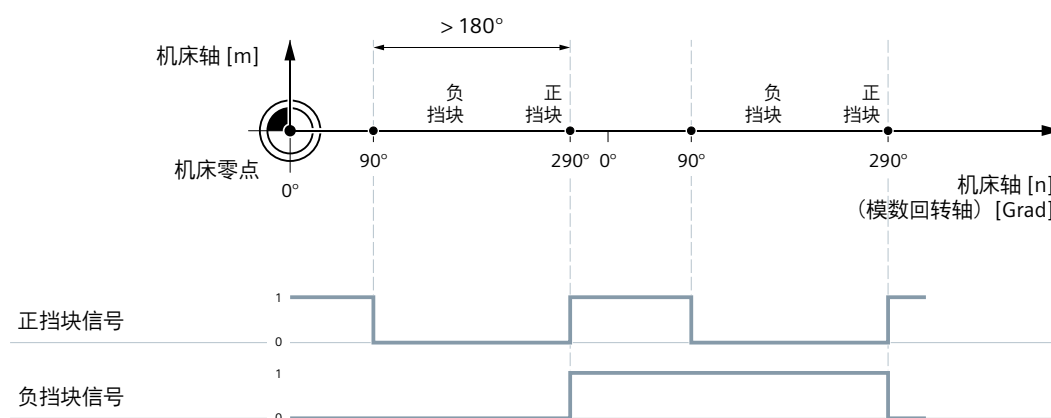


图 21-4 模数回转轴的软件挡块（正挡块 - 负挡块 > 180 度）

21.2.2 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
- Ausgabe auf die NC-Peripherie, falls das 2.Byte in den folgenden Maschinendaten nicht spezifiziert wurde (= "0"):
MD10470 SW_CAM_ASSIGN_FASTOUT_2
...
MD10473 SW_CAM_ASSIGN_FASTOUT_4

21.2 挡块信号和挡块位置

线性轴

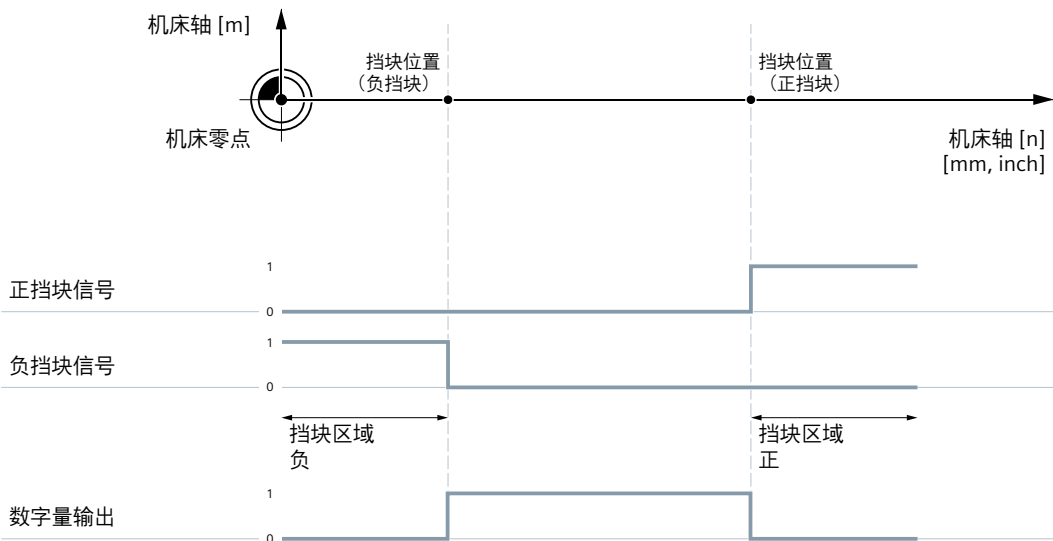


图 21-5 线性轴的行程开关信号（负挡块 < 正挡块）

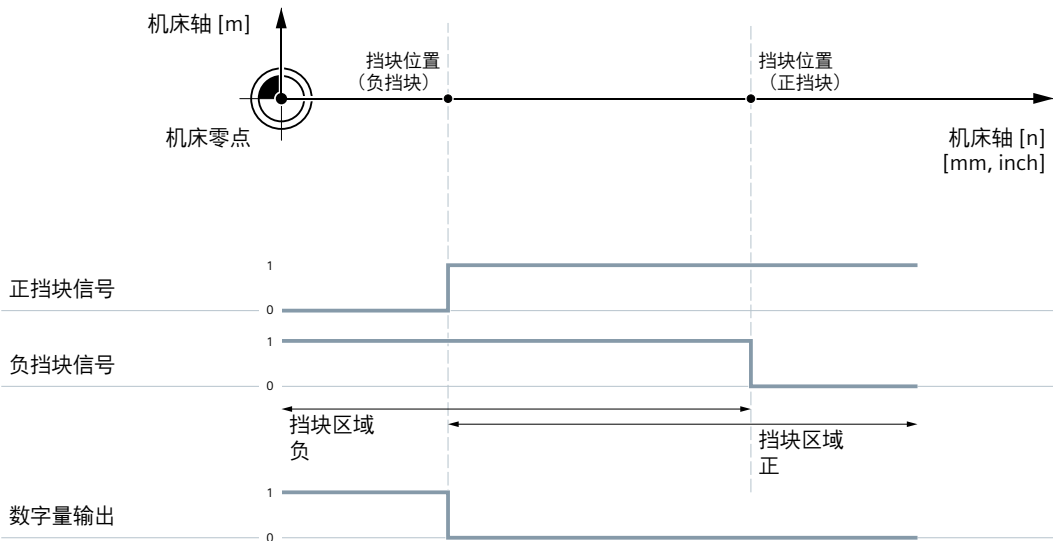


图 21-6 线性轴的行程开关信号（正挡块 < 负挡块）

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：

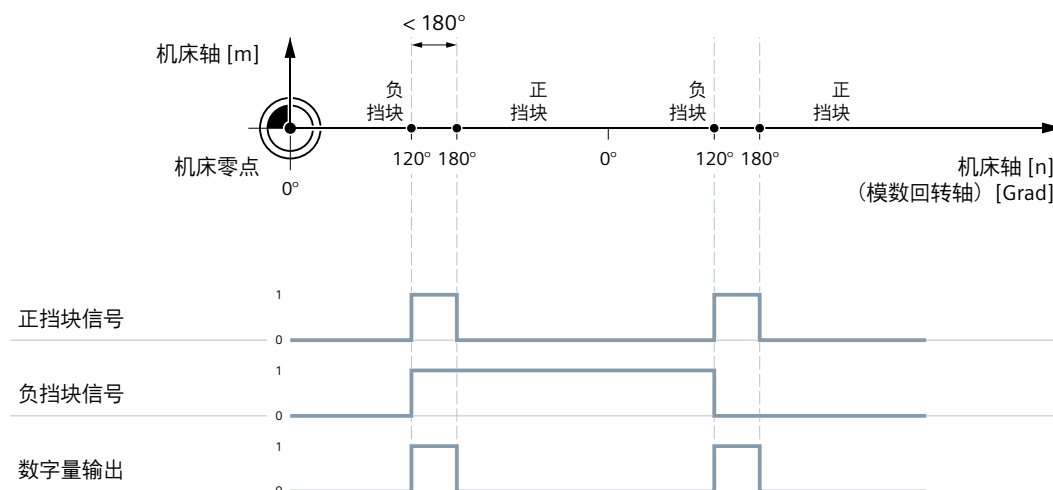


图 21-7 模数回转轴的软件挡块（正挡块 - 负挡块 < 180 度）

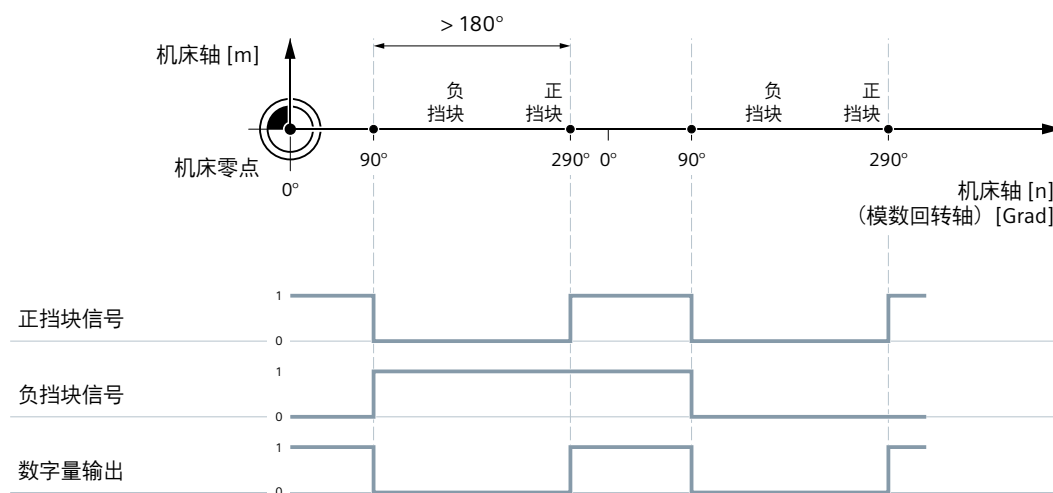


图 21-8 模数回转轴的软件挡块（正挡块 - 负挡块 > 180 度）

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

21.2 挡块信号和挡块位置

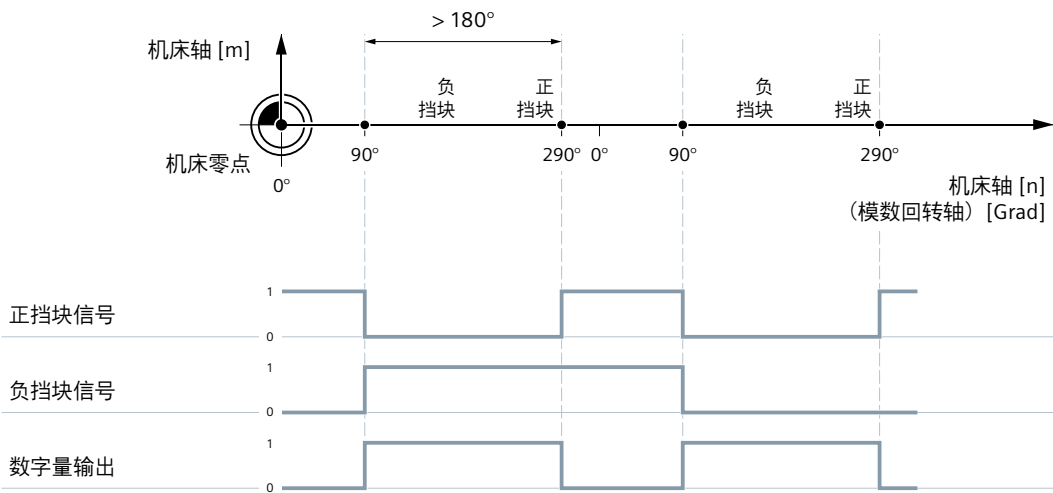


图 21-9 模数回转轴的软件挡块（正挡块 - 负挡块 > 180 度）；正挡块信号特性的反转被抑制

21.2.3 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

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 NC 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。

21.2.4 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

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。

21.3 挡块信号输出

21.3.1 激活

挡块的状态（挡块信号）可输出至 PLC，也可另外输出至 NC 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：

```
<Axis>.basic.out.swCams（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：

```
<Axis>.basic.in.revolutionalFeedrateActive（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）。

21.3 挡块信号输出

PLC 信号

PLC → NC

Basic Program Plus	Basic Proogram	
<Axis>.basic.out.swCams	LBP_Axis*.A_SWCam	DB31, ...DBX2.0

NC → PLC

Basic Program Plus	Basic Proogram	
<Axis>.basic.in.revolutionalFeedrateActive	LBP_Axis*.E_RevFD	DB31, ...DBX62.2

21.3.2 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：
<Nc>.basic.in.swCams[0].minus（负挡块信号 1 至 32）

正挡块信号

正挡块信号的状态由以下 NC/PLC 接口信号输出：
E_SWCamPlus[0]（正挡块信号 1 至 32）

说明 若未选择测量系统或 NC/PLC 接口信号 <Axis>.basic.out.swCams（激活挡块）置 0，那么下列 NC/PLC 接口信号同样会置 0： • <Nc>.basic.in.swCams[0].minus（负挡块信号 1-32） • <Nc>.basic.in.swCams[0].minus（正挡块信号 1-32） • <Axis>.basic.in.swCamsActive（挡块生效）

PLC 信号

PLC → NC

Basic Program Plus	Basic Proogram	
<Axis>.basic.out.swCams	LBP_Axis*.A_SWCam	DB31, ...DBX2.0

NC → PLC

Basic Program Plus	Basic Proogram	
<Axis>.basic.in.swCamsActive	LBP_Axis*.E_SWCam	DB31, ...DBX62.0
<Nc>.basic.in.swCams	LBP_NC.E_SWCamMinus	DB10.DBX110.0
<Nc>.basic.in.swCams[0].minus	LBP_NC.E_SWCamMinus[0]	DB10.DBX110.0..D BX113.7
<Nc>.basic.in.swCams[0].plus	LBP_NC.E_SWCamPlus[0]	DB10.DBX114.0..D BX117.7

21.3.3 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。

NC 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 NC 输出。

更多信息

参见功能手册之基本功能：NC 数字量和模拟量 I/O。

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 NC 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 NC I/O

21.3 挡块信号输出

- MD10472 SW_CAM_ASSIGN_FASTOUT_3 硬件指定，用于将挡块 17 - 24 的信号输出至 NC 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 NC 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 NC I/O 或板载输出。

挡块信号的切换精度受位置控制周期的时间跨度限制。

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：

$V_{\text{当前}} = 20 \text{ m/min}$ ，位置控制周期 = 4 ms $\Delta \text{位置} = 1.33 \text{ mm}$
 $V_{\text{当前}} = 2000 \text{ rpm}$ ，位置控制周期 = 2 ms $\Delta \text{位置} = 24 \text{ 度}$

21.3.4 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）

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 接口上可查看板载输出的 NC 映射以及正负挡块的状态。

21.3 挡块信号输出

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

21.3.5 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。

21.4 “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。

21.5 边界条件

- 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。

21.5 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。

扩展的停止和退回

22.1 简要说明

通过“扩展停止和退回”功能（亦称为 ESR 功能）可在故障状态下根据进程进行灵活反应：

- **扩展停止**
在所允许的特定故障状况下，对针对“扩展停止”而使能的轴执行有序停止。
- **退回**
所涉及刀具会以最快的速度从工件退回。
- **再生运行（SINAMICS 驱动功能“Vdc 控制”）**
若未达到可参数设置的直流母线电压值（例如由于掉电），将会通过预设驱动回馈的制动能提供退回所需的电能（再生运行）。

ESR 和生效的耦合

在停止和退回过程中，生效的耦合保持一个可参数设置的时间。

在轨迹上退回

除了纯轴向退回之外，还可以将直线编程为退回轨迹。

22.2 驱动自控 ESR

22.2.1 基本原理

功能

驱动自控的扩展停止和退回功能 (ESR) 能够独立于上级控制系统 (NC) 快速分离工件和刀具。

在驱动中可配置以下轴特定的功能：

- 再生运行
- 扩展停止
- 退回

22.2 驱动自控 ESR

在出错时，会自动触发驱动自控的反应。驱动自控的反应也可以按用户自己的要求通过零件程序或同步动作由上级控制系统触发。

由于驱动自控 ESR 的停止和退回运动纯粹是轴特定的，因此不考虑耦合，这与控制系统引导下的 ESR 相反。

驱动自控 ESR 也可以与控制系统引导下的 ESR 结合使用。这意味着，即使控制系统没有指定（例如由于通信故障），也可以在驱动中进行停止和退回运动。

说明

DRIVE-CLiQ

为了在 DRIVE-CLiQ 出错的情况下尽量减少对驱动自控 ESR 功能的影响，建议在 ESR 框架下，所涉及的电机模块不通过线性拓扑结构，而是直接与控制单元 (CU) 连接。

前提条件

必须满足下列前提条件：

- SINAMICS 和 SINUMERIK 软件版本：≥ V4.4
- 伺服驱动对象，PROFIdrive 报文：102 - 199
- 伺服驱动对象，“ESR”功能模块生效
- 伺服驱动对象、再生运行（Vdc 控制）在驱动中生效
- 控制单元驱动对象，PROFIdrive 报文：390、391
- 通过 CSM 或 UPS 支持模块的 24V 电源

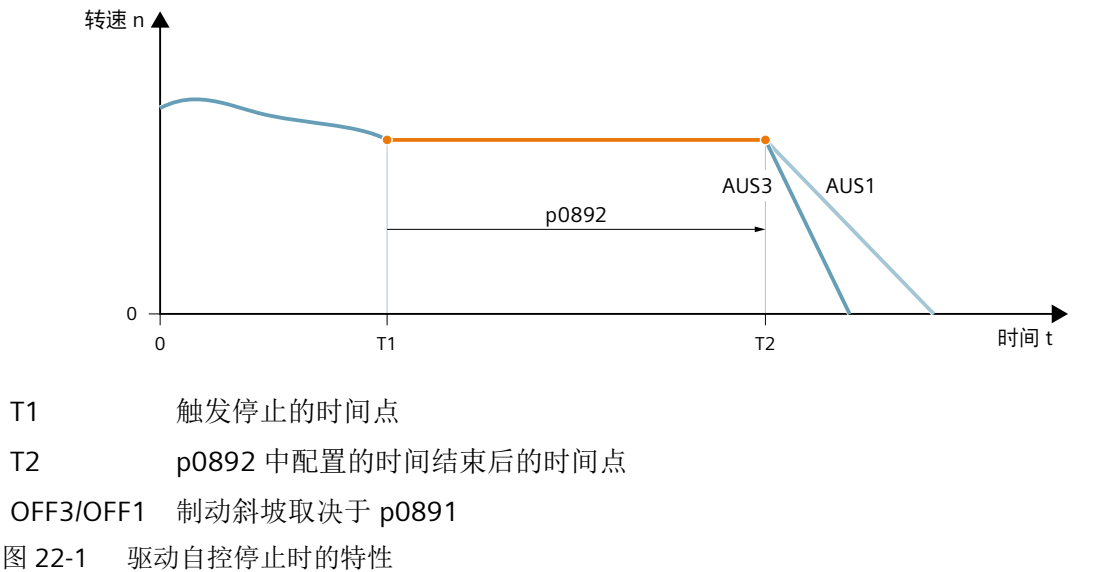
22.2.2 在驱动中配置停止功能

通过以下驱动参数配置驱动自控停止功能：

参数	说明	
p0888	ESR: 组态	
	值	含义
	1	扩展停止（驱动自控）

参数	说明	
p0891	ESR: 减速斜坡	
	值	含义
	0	OFF3（默认） 立即预设设定转速 (n_{set}) = 0 后，驱动沿着 OFF3 减速斜坡（p1135: OFF3 斜坡下降时间）减速制动。
	1	OFF1 立即预设设定转速 (n_{set}) = 0 后，驱动沿着斜坡函数发生器的减速斜坡（p1121: 斜坡功能发生器斜坡下降时间）减速制动。

参数	说明
p0892	ESR: 延时
	驱动以故障发生时的转速设定值继续运行一段配置的时间。



反馈信息

停止的状态会反馈给控制系统（参见章节“ESR 状态的反馈 (页 960)”）。

22.2 驱动自控 ESR

更多信息

驱动参数的详细说明请见：
SINAMICS S120/S150 参数手册

22.2.3 在驱动中配置退回

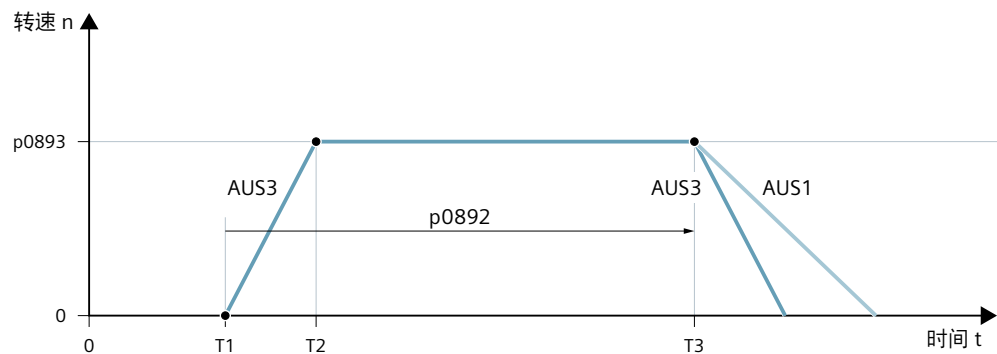
通过以下驱动参数配置驱动自控退回：

参数	说明	
p0888	ESR: 组态	
	值	含义
	2	扩展退回（驱动自控）

参数	说明	
p0891	ESR: 减速斜坡	
	值	含义
	0	OFF3（默认） 立即预设设定转速 (n_set) = 0 后，驱动沿着 OFF3 减速斜坡（p1135: OFF3 斜坡下降时间）减速制动。
	1	OFF1 立即预设设定转速 (n_set) = 0 后，驱动沿着斜坡函数发生器的减速斜坡（p1121: 斜坡功能发生器斜坡下降时间）减速制动。

参数	说明
p0892	ESR: 延时
	该参数规定了按 p0893 中规定的转速起动和随后的恒定运行总共所经过的时间。
	然后，根据 p0891 中的参数设置，沿 OFF1 或 OFF3 斜坡运行。

参数	说明
p0893	ESR: 转速
	该参数规定了触发退回时沿着 OFF3 斜坡起动的转速（退回转速）。



- T1 触发退回时的时间点
- T2 达到 p0893 中配置的退回转速时的时间点
- T3 p0892 中配置的时间结束后的时间点

图 22-2 驱动自控退回时的特性

反馈信息

退回的状态会反馈给控制系统（参见章节“ESR 状态的反馈 (页 960)”）。

更多信息

驱动参数的详细说明请见：
SINAMICS S120/S150 参数手册

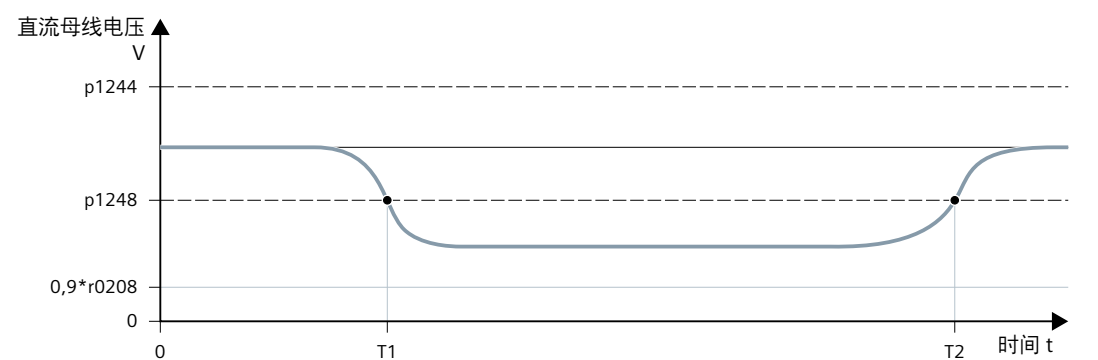
22.2.4 在驱动中配置再生运行

通过以下驱动参数配置驱动自控 ESR 的再生运行：

参数	说明	
p0888	ESR: 组态	
	值	含义
	3	再生运行（Vdc 控制器）

参数	说明	
p1240	Vdc 控制器或 Vdc 监控的组态	
	值	含义
	2	当达到直流母线电压阈值下限 (p1248) 时，电机在 ESR 时制动，以利用其动能来为直流母线提供缓冲。

参数	说明
p1248	直流母线电压阈值下限
	设置直流母线电压阈值下限。当 p1240=2 时，该阈值被用作 Vdc_min 控制器的限制设定值。



- p124 直流母线电压阈值上限
4
- p124 直流母线电压阈值下限
8
- r020 功率单元的标称电源电压。
8 如果直流母线电压低于 $0.9 \cdot r0208$ ，就会出现脉冲删除。
- T1 触发再生运行时的时间点
- T2 结束再生运行时的时间点

图 22-3 驱动自控_再生运行_配置

发电机最小转速

发电机轴的电机转速下限通过驱动参数 p2161 来配置：

参数	说明
p2161	转速阈值 3 触发报警的转速阈值： n_ist < 转速阈值 3

反馈信息

再生运行的状态会反馈给控制系统（参见章节“ESR 状态的反馈 (页 960)”）。

更多信息

有关驱动参数和所发生的 Vdc 控制的详细说明请见：

- SINAMICS S120/S150 参数手册
- SINAMICS S120 驱动功能的功能手册；伺服控制 > Vdc 控制

22.2.5 通过系统变量使能 ESR

轴的通过驱动参数配置的 ESR 反应的使能必须通过以下轴特定的系统变量在零件程序/同步动作中按用户自己的要求来编程：

\$AA_ESR_ENABLE[<轴>]	
值	含义
1	在驱动中已使能 ESR 反应
0	在驱动中已禁止 ESR 反应

22.2.6 通过系统变量触发 ESR

所有通过驱动参数进行配置的并通过 \$AA_ESR_ENABLE[<轴>] 使能的对驱动自控 ESR 反应的触发必须通过以下 NC 特定的系统变量在零件程序/同步动作中按用户自己的要求来编程：

\$AN_ESR_TRIGGER	
值	含义
1	触发已使能的驱动自控 ESR 反应
0	不触发已使能的驱动自控 ESR 反应

说明
触发信号在零件程序/同步动作中必须至少存在 2 个 IPO 周期，才能通过 PROFIdrive 报文安全地传送到驱动中。

驱动设备中的映射

系统变量 \$AN_ESR_TRIGGER 被映射到驱动参数 p0890.0 上。通过驱动设备对 ESR 反应进行调试时，驱动参数 p0890.0 与控制单元上的报文 390 或 391 中的相应信息自动互联。

更多信息

驱动参数的详细说明请见：
SINAMICS S120/S150 参数手册

22.2.7 ESR 状态的反馈

当前 ESR 状态的反馈由驱动通过循环 PROFIdrive 报文的信号字 (MELDW) 提供给控制系统。在控制系统中，可以通过系统变量和 NC/PLC 接口信号来读取状态。

系统变量
系统变量可以在零件程序/同步动作中进行评估，例如触发驱动自控 ESR（参见“通过系统变量触发 ESR (页 960)”章节）或应答驱动中触发的 ESR 反应（参见“应答 ESR 反应 (页 961)”章节）。

信号关系

MELDW	系统变量	NC/PLC 接口信号	含义
MELDW.Bit 9 (r0887.12)	\$AA_ESR_STAT.Bit 2	<Axis>.drive.in.esrResponseInitiated	ESR: 反应已触发或再生运行生效 (r0887.12)
MELDW.Bit 4 (r0056.15)	\$AA_ESR_STAT.Bit 3	<Axis>.drive.in.esrDcVoltageAlarmActive	ESR: 直流母线欠电压 (p1248)
MELDW.Bit 2 r2199.0	\$AA_ESR_STAT.Bit 4	<Axis>.drive.in.esrSpeedBelowMin	ESR、再生运行: 低于最小转速 (p2161)

PLC 信号

NC → PLC

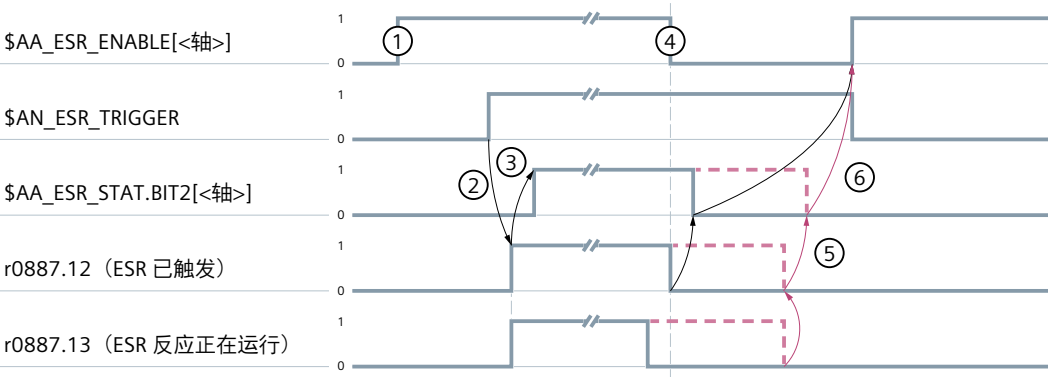
Basic Program Plus	Basic Program	
<Axis>.drive.in.esrDcVoltageAlarmActive	LBP_Axis*.E_UDClessAlThreshold	DB31, ...DBX95.1
<Axis>.drive.in.esrResponseInitiated	LBP_Axis*.E_I2TLimit	DB31, ...DBX95.2
<Axis>.drive.in.esrSpeedBelowMin	-	DB31, ...DBX95.3

22.2.8 应答 ESR 反应

对轴特定的驱动自控 ESR 反应的应答必须在零件程序/同步动作中按用户自己的要求来编程。驱动中的 ESR 反应完成后，必须在系统变量 \$AA_ESR_ENABLE 中产生信号沿变化：1→0→1。系统变量 \$AN_ESR_TRIGGER 也必须重置为值 0，以便可以再次触发驱动自控 ESR 反应。

系统变量和驱动参数

下图显示了触发和应答 ESR 反应时系统变量和驱动参数之间的关系。



- ① NC: 通过 $\$AA\_ESR\_ENABLE = 1$ (轴特定) 使能 ESR 反应
- ② NC: 通过 $\$AN\_ESR\_TRIGGER = 1$ 触发 ESR 反应
- ③ 驱动: 已识别到对 ESR 反应的触发 ($CU\_STW.2 == 1$) $\Rightarrow$
 $r0887.12 = 1$ 和 $r0887.13 = 1$
通过 $\$AA\_ESR\_STAT.BIT2 = 1$ (轴特定) 反馈给 NC
- ④ NC: 通过 $\$AA\_ESR\_ENABLE = 0$ (轴特定) 应答 ESR 反应
- 黑色: ESR 反应不再在时间点 ④ 时运行 ($r0887.13 == 0$) $\Rightarrow$
- ⑤ 驱动: “ESR 已触发” 复位: $r0887.12 = 0$
通过 $\$AA\_ESR\_STAT.BIT2 = 0$ (轴特定) 反馈给 NC
- ⑥ NC: 可以再次使能 ESR 反应 $\Rightarrow$
 $\$AA\_ESR\_ENABLE = 1$ 和 $\$AN\_ESR\_TRIGGER = 0$
- 红色: ESR 反应在时间点 ④ 时仍在运行 ($r0887.13 == 1$) $\Rightarrow$
- ⑤ 驱动: “ESR 已触发” 仅在驱动中的 ESR 反应结束时复位 ($r0887.12 = 0$): $r0887.13 == 0$
通过 $\$AA\_ESR\_STAT.BIT2 = 0$ (轴特定) 反馈给 NC
- ⑥ 见上文 ⑥

图 22-4 过程: 触发和应答驱动自控 ESR 反应

22.2.9 在零件程序中配置 ESR

可以通过下面介绍的指令在零件程序中更改驱动自控 ESR 功能“停止”和“退回”的配置。

22.2.9.1 停止 (ESRS)

句法

ESRS (<轴_1>,<停止时间_1>[, ..., <轴_n>,<停止时间_n>])

含义

ESRS:	该功能可用于更改与驱动自控 ESR 功能“停止”相关的驱动参数。 特点： • 必须位于单独的程序段中 • 会触发预处理程序停止 • 不可以在同步动作中使用 • 每次功能调用最多可编程 5 个轴；n=5	
<轴_n>:	为指定轴写入驱动参数 p0888（组态）：p0888 = 1	
	数据类型:	AXIS
	取值范围:	通道轴名称
<停止时间_n>:	为 <轴_n> 下指定的轴写入驱动参数 p0892（延时）：p0892 = <停止时间_n>	
	单位:	s
	数据类型:	REAL
	取值范围:	0.00 - 20.00

22.2.9.2 退回 (ESRR)

句法

ESRR (<轴_1>,<退回行程_1>,<退回速度_1>[, ..., <轴_n>,<退回行程_n>,<退回速度_n>])

含义

ESRR:	该功能可用于更改与驱动自控 ESR 功能“退回”相关的驱动参数。 特点： • 必须位于单独的程序段中 • 会触发预处理程序停止 • 不可以在同步动作中使用 • 每次功能调用最多可编程 5 个轴；n=5	
<轴_n>:	为指定轴写入驱动参数 p0888（组态）：p0888 = 2	
	数据类型:	AXIS
	取值范围:	通道轴名称
<退回速度_n>:	为 <轴_n> 下指定的轴写入驱动参数 p0893（退回转速）： p0893 = (<退回速度> 根据当前有效总传动比换算成对应的退回转速)	
	单位:	mm/min、inch/min、Grad/min（取决于轴的单位）
	数据类型:	REAL
	取值范围:	0.00 - MAX
<退回行程_n>:	为 <轴_n> 下指定的轴写入驱动参数 p0892（退回持续时间）： p0892 = <退回行程_n> / <退回速度_n>	
	单位:	mm、inch、Grad（取决于轴的单位）
	数据类型:	REAL
	取值范围:	MIN - MAX

说明

退回转速

必须确保驱动可以在退回持续时间（参见参数 <退回行程_n>）内达到编程的退回转速（参见参数 <退回速度_n>）。为此，请检查以下驱动参数中的设置：

- p1082[0...n]（最大转速）
- p1135[0...n]（OFF3 斜坡下降时间）

退回行程

退回行程换算成驱动参数 p0892 的退回持续时间，前提是在轴静止时触发退回运动（参见“图 22-3 驱动自控_再生运行_配置(页 958)”）。

如果在轴的移行运动期间触发退回运动，由于退回运动与移行运动的叠加，实际运行的退回行程会偏离编程的退回行程。

相关性

退回行程和退回速度的编程值涉及负载侧。在写入驱动参数之前，将它们换算为电机侧适用的值。换算涉及到 NC 中功能执行时间点的有效传动比。如果针对一个轴使能了驱动自控 ESR (\$AA_ESR_ENABLE[<轴>] == 1)，则不允许更改轴向传动比。

说明

更改传动比

如果针对一个轴未使能驱动自控 ESR (\$AA_ESR_ENABLE[<轴>] == 0)，则可以更改轴向传动比。用户/机床制造商全权负责在传动比改变后相应地调整退回行程和退回速度。

参见

在驱动中配置再生运行 (页 957)

22.2.9.3 边界条件

说明

一致的参数更改

为了使驱动参数始终有效，必须在零件程序中取消驱动自控 ESR 使能，再使用功能 ESRS (...) 和 ESRR (...).

示例：

程序代码	注释
N100 \$AA_ESR_ENABLE[X1]=0	； 取消 ESR 使能
N110 \$AA_ESR_ENABLE[Z1]=0	
N120 \$AA_ESR_ENABLE[Y1]=0	
N130 ESRR(X1,20.0,10000.0,Z1,-15.0,20000.0)	； 更改 ESR 参数
N140 ESRS(Y1, 0.5, B1, 0.5)	
N150 \$AA_ESR_ENABLE[X1]=1	； 置位 ESR 使能
N160 \$AA_ESR_ENABLE[Z1]=1	
N170 \$AA_ESR_ENABLE[Y1]=1	

执行功能 ESRS (...) 和 ESRR (...) 时，零件程序的进一步处理将暂停，直到驱动参数的写入得到应答。在此期间，将显示以下信息作为对用户的反馈：“等待来自 <Funktion> 的驱动参数”。

22.3 边界条件

带计算的程序段搜索

在带计算的程序段搜索过程中，将汇集 ESRS (...) 和 ESRR (...) 功能并在操作程序段中执行。

在“程序测试”模式 (SERUPRO) 下进行计算的程序段搜索

SERUPRO 期间将立即执行 ESRS (...) 和 ESRR (...) 功能。

复位特性

通过功能 ESRS (...) 和 ESRR (...) 写入的参数值在关闭/开启驱动后的驱动引导启动时或在驱动热启动时被驱动中所存储的参数值覆盖。

驱动参数可通过 HMI 用户界面保存或在驱动中保存。通过 HMI 用户界面：

- 操作区域 “调试”>“机床数据”>“驱动 MD”>“保存/复位”>“保存”>“驱动对象”或“驱动设备”或“驱动系统”

控制系统引导启动 (POWERON)

如果要在控制系统引导启动 (POWER ON) 后驱动自控 ESR 便已生效，则必须在由 PLC 启动的 ASUB 内或通过 PROGEVENT 机制进行编程和使能。

更多信息

有关事件控制式程序调用的详细信息，见：

功能手册之基本功能；运行方式组、通道、程序运行、复位特性；事件控制式程序调用

22.3 边界条件

坐标转换

如果已为参与了生效的坐标转换的一个轴激活了 NC 引导下的扩展停止和退回，则还必须为参与了坐标转换的所有其他轴激活扩展停止和退回。否则，在激活扩展停止和退回时，会导致坐标转换组的解体。

驱动组件

参与的驱动组件必须功能齐全，以便可以执行配置的停止或退回反应。轴特定的伺服或驱动报警（用于描述这些组件中的一个出现失灵情况或功能故障）将隐式报告所涉及轴的配置的停止和退回反应不再可用或仅部分可用。

运动同步动作

运动同步动作以插补周期进行处理。插补周期的增加（例如由于大量生效的运动同步动作）因此导致用于评估触发条件和用于触发停止和退回反应的时间网格变得更加不准确。

22.4 示例

22.4.1 NC 引导下的反应

任务

轴 A 应作为发电机使用，轴 X 在故障情况下应以最大速度退回 10mm。Y 轴和 Z 轴将延迟 100 ms 停止，以便退回轴 X 有足够的时间取消机械耦合。

该示例展示了 NC 引导下的反应。

仅显示了整体功能的基本部分。

前提条件

以下功能必须可用：

- 扩展停止和退回 (ESR)
- 静态同步动作
- 异步子程序 (ASUB)

设置

机床数据

轴特定的 ESR 运行方式

- MD37500 \$MA_ESR_REACTION[X] = 21（NC 引导下的退回轴）
- MD37500 \$MA_ESR_REACTION[Y] = 22（NC 引导下的停止轴）
- MD37500 \$MA_ESR_REACTION[Z] = 22（NC 引导下的停止轴）

组态：在 NC 引导下扩展停止

- MD21380 \$MC_ESR_DELAY_TIME1 = 0.1（以秒为单位继续进行轨迹插补的时长）
- MD21381 \$MC_ESR_DELAY_TIME2 = 0.04（制动时间，单位：秒）

22.4 示例

编程

程序代码	注释
LFPOS	轴特定退刀返回到一个位置
POLF[X]=IC(10)	退回位置，增量
POLFMASK(X)	退回使能
; 使能“扩展停止和退回”	
\$AA_ESR_ENABLE[X] = 1	; X 轴
\$AA_ESR_ENABLE[Y] = 1	; Y 轴
\$AA_ESR_ENABLE[Z] = 1	; Z 轴

触发条件和静态同步动作

示例 1

触发条件为，出现了会触发跟踪方式的报警：

程序代码

IDS=02 WHENEVER (\$AC_ALARM_STAT B_AND 'H2000') > 0 DO \$AC_ESR_TRIGGER=1

示例 2

触发条件为，例如在将 Y 定义为电子齿轮箱跟随轴且允许的最大同步运行偏差为 100 mm 时，对电子齿轮箱同步运行监控做出了响应：

程序代码

IDS=03 WHENEVER \$VA_EG_SYNCDIFF[Y] > 0.1 DO \$AC_ESR_TRIGGER=1

22.4.2 螺纹切削时退回

在螺纹切削期间，应对故障/中断做出这样的反应：将 X 轴退回到 POLF 下指定的位置。Z 轴继续正常移行，直到停止。

程序代码	注释
N10 G0 G90 X200 Z0 S200 M3	;
N20 G0 G90 X170	;
N30 POLF[X]=210	; X 轴退回位置，绝对
N40 LFPOS	; 通过 POLF/POLFMASK 退回开启
N50 POLFMASK(X)	; X 轴退回使能
N60 LFON	; 螺纹切削快速退刀开启
	; 螺纹切削
N70 G33 X100 I10	; 退回反应，轴向：X（绝对）
N80 X130 Z-45 K10	; dtO.

程序代码	注释
N90 X155 Z-128 K10	; dto.
N100 X145 Z-168 K10	; dto.
N110 X120 I10	; dto.
N120 G0 Z0 LFOF	; 螺纹切削快速退刀关闭
N130 POLFMASK()	; 所有轴退回禁止

22.4.3 通过 ASUB 和快速输入快速退刀

通过快速输入 1 由程序指令 LIFTFAST（快速退刀）激活 ASUB:

程序代码	注释
N10 SETINT (1) PRIO=1 ABHEB_Y LIFTFAST	; 激活 ASUB, 快速输入 1
N20 LFPOS	; 通过 POLF/POLFMASK 退回开启
N30 POLF[X]=19.5 POLF[Y]=33.3	; X、Y 轴退回位置, 绝对
N40 POLF[Z]=100	; Z 轴退回位置, 绝对
N50 X0 Y0 G0	;
N60 POLFMASK(X, Y)	; X、Y 轴退回使能
N70 Z100 G1 F1000	; 退回反应, 轴向: X、Y (绝对)
N80 POLFMASK(Z)	; X、Y 轴退回禁止和
	; Z 轴退回使能
N90 Y10	; 退回反应, 轴向: Z (绝对)
N100 POLFMASK()	; 所有轴退回禁止

22.4.4 绝对快速退刀和增量快速退刀

退回到绝对位置并按增量行程进行退回:

程序代码	注释
N10 \$AA_ESR_ENABLE[X]=1	; 退回使能, X 轴
N20 \$AA_ESR_ENABLE[Z]=1	; 退回使能, Z 轴
N30 \$AA_ESR_ENABLE[Y]=1	; 退回使能, Y 轴
N40 LFPOS	; 通过 POLF/POLFMASK 退回开启
N50 POLF[X]=IC(3.0) POLF[Y]=-4.0	; X 轴退回位置, 增量
	; Y 轴退回位置, 绝对
N60 POLF[Z]=100	; Z 轴退回位置, 绝对
N70 G0 X0 Y0 Z0	;
N80 POLFMASK(X, Y)	; X、Y 轴退回使能
N90 Z100 G1 F1000	; 退回反应, 轴向: X (增量)、Y (绝对)
N100 POLF[X]=10	; X 轴退回位置, 绝对
N110 Y0 G1 F1000	; 退回反应, 轴向: X、Y (绝对)

22.4 示例

程序代码	注释
N120 POLFMASK(Z)	; X、Y 轴退回禁止和 ; Z 轴退回使能
N130 Y10	; 退回反应，轴向：Z（绝对）
N140 POLFMASK()	; 所有轴退回禁止

22.4.5 以轴的线性关系快速退刀

以线性关系退回，绝对和增量

程序代码	注释
N10 \$AA_ESR_ENABLE[X]=1	; 退回使能，X 轴
N20 \$AA_ESR_ENABLE[Y]=1	; 退回使能，Y 轴
N30 \$AA_ESR_ENABLE[Z]=1	; 退回使能，Z 轴
N40 LFPOS	; 通过 POLF/POLFMASK 退回开启
N50 POLF[X]=IC(3.0) POLF[Z]=-4.0	; X 轴退回位置，增量 ; Z 轴退回位置，绝对
N60 POLF[Y]=100	; Z 轴退回位置，绝对
N70 X0 Y0 Z0 G0	; ---
N80 POLFMLIN(X, Y)	; X、Y 轴退回使能，线性 ; 关系
N85 POLFMASK(Z)	; Z 轴退回使能
N90 Z100 G1 F1000	; 退回反应： ; - 线性关系：X（增量）、Y（绝对） ; - 轴向：Z（绝对）
N95 POLF[X]=10	; X 轴退回位置，绝对
N100 Y0 G1 F1000	; 退回反应： ; - 线性关系：X（绝对），Y（绝对） ; - 轴向：Z（绝对）
N110 POLFMLIN()	; X、Y 轴退回禁止
N120 Y10	; 退回反应，轴向：Z（绝对）
N130 POLFMASK()	; 所有轴退回禁止

行程计算

23.1 简要说明

功能

通过“行程计算”功能，提供机床轴的 NC 特定的数据作为系统变量和 OPI 变量，借助这些数据可以估算出机床轴的负担，从而能够对机床的维护状态得出结论。

所记录的数据

记录以下数据：

- 总移行行程
- 总移行时间
- 移行过程总数
- 高轴速下的总移行行程
- 高轴速下的总移行时间
- 高轴速下的移行过程数
- 急动的总和
- 急动时轴的移行时间
- 急动时的移行过程数

评估

当该功能激活时，所选数据由控制系统自动带入，并通过系统变量和 OPI 变量提供用于在零件程序或同步动作中进行评估或通过用户特定的 HMI 功能进行评估。

含义

数据始终存储在控制系统中，因此即使在断电/上电后也可以继续进行使用。因此，一个数据的当前值表示自该功能激活以来所有测量值之和。

23.2 数据

可供使用的数据如下：

系统变量 ¹⁾	OPI 变量	含义
\$AA_TRAVEL_DIST	aaTravelDist	总移行行程： 机床坐标系 ²⁾ 中所有设定位置变化之和，单位 [mm] 或 [Grad]。
\$AA_TRAVEL_TIME	aaTravelTime	总移行时间： 机床坐标系 ²⁾ 中的设定位置变化的插补周期之和，单位 [s]（分辨率：1 个插补周期）
\$AA_TRAVEL_COUNT	aaTravelCount	移行过程总数： 机床轴的一个移行过程通过相对设定位置的下列状态序列标示：静止 > 移行 > 静止
\$AA_TRAVEL_DIST_HS	aaTravelDistHS	更高轴速 ³⁾ 下的总移行行程
\$AA_TRAVEL_TIME_HS	aaTravelTimeHS	更高轴速 ³⁾ 下的总移行时间
\$AA_TRAVEL_COUNT_HS	aaTravelCountHS	更高轴速 ³⁾ 下的移行过程总数
\$AA_JERK_TOT	aaJerkTotal	轴的急动的总和： 所有急动设定值之和，单位 [m/s ³] 或 [Grad/s ³]
\$AA_JERK_TIME	aaJerkTime	急动时的总移行时间： 急动的设定值变化的插补周期之和，单位 [s]（分辨率：1 个插补周期）
\$AA_JERK_COUNT	aaJerkCount	急动时的移行过程总数
¹⁾ 可以从零件程序和同步动作中读取系统变量 ²⁾ 机床坐标系：机床坐标系 ³⁾ 机床轴的实际速度 ≥ 参数设置的最大轴速度 (MD32000 MAX_AX_VELO) 的 80%		

23.3 参数设置

23.3.1 常规激活

该功能通常通过 NC 特定的机床数据激活：

MD18860 \$MN_MM_MAINTENANCE_MON（维护数据记录激活）

23.3.2 数据组

数据被汇总在数据组中。

数据组通过轴特定的机床数据激活：

MD33060 \$MA_MAINTENANCE_DATA（维护数据记录组态）

位	值	激活以下数据：系统变量/OPI 变量
0	1	- 总移行行程：\$AA_TRAVEL_DIST / aaTravelDist - 总移行时间：\$AA_TRAVEL_TIME/aaTravelTime - 移行过程总数：\$AA_TRAVEL_COUNT / aaTravelCount
1	1	- 高速下的总移行行程： \$AA_TRAVEL_DIST_HS / aaTravelDistHS - 高速下的总移行时间 \$AA_TRAVEL_TIME_HS / aaTravelTimeHS - 高速下的移行过程总数 \$AA_TRAVEL_COUNT_HS / aaTravelCountHS
2	1	- 急动的总和：\$AA_JERK_TOT / aaJerkTotal - 急动时的移行时间：\$AA_JERK_TIME / aaJerkTime - 急动时的移行过程总数： \$AA_JERK_COUNT / aaJerkCount

23.4 示例

23.4.1 每个零件程序的移行行程

一台机床上有三个几何轴 AX1、AX2 和 AX3。针对几何轴 AX1，应确定在零件程序中运行的总移行行程、总移行时间和移行过程数。

参数设置

激活整体功能：

MD18860 \$MN_MM_MAINTENANCE_MON = TRUE

激活组：几何轴 AX1 的“总移行行程、总移行时间和移行过程数”

MD33060 \$MA_MAINTENANCE_DATA[AX1] = 1

编程

要确定与零件程序相关的值，必须将系统变量的当前值保存在零件程序开头的计算变量中。然后在零件程序结束时计算差值。

零件程序（节选）：

程序代码	注释
...	
R10=\$AA_TRAVEL_DIST[AX1]	; 当前值:
R11=\$AA_TRAVEL_TIME[AX1]	; AX1 总移行行程
R12=\$AA_TRAVEL_COUNT[AX1]	; AX1 总移行时间
	; AX1 移行过程数
...	
	; 轴的移行运动
	; 计算差值:
R10=R10-\$AA_TRAVEL_DIST[AX1]	; 零件程序中的 AX1 移行行程
R11=R11-\$AA_TRAVEL_TIME[AX1]	; 零件程序中的 AX1 移行时间
R12=R12-\$AA_TRAVEL_COUNT[AX1]	; 零件程序中的 AX1 移行过程数

附录

A

A.1 缩略语列表

A	
A	输出
AFIS	Automatic Filter Switch: 自动切换滤波器
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
AUTO	AUTOMATIC, 一种运行方式
AUXFU	Auxiliary Function: 辅助功能
AWL	指令列表

B	
BAG	运行方式组
BCD	Binary Coded Decimals: 用二进制代码编码的十进制数
BICO	Binector Connector
BIN	Binary Files: 二进制文件
BKS	基准坐标系
BM	运行信息
BO	Binector Output
BTSS	操作面板接口

C	
CLC	间隙控制
CNC	Computerized Numerical Control: Computerunterstützte numerische Steuerung
COM	Communication
CP	Communication Processor

A.1 缩略语列表

C	
CPU	Central Processing Unit:中央处理器
CST	Configured Stop:配置停止

D	
DDS	Drive Data Set:驱动数据组
DIR	Directory:目录
DO	Drive Object
DRF	Differential Resolver Function:微分旋转变压器功能（手轮）
DRY	Dry Run:空运行进给
DWORD	双字（当前 32 位）

E	
E	输入
EES	Execution from External Storage
E/A	输入/输出
ESR	扩展的停止和退回
ETC	ETC 键">", 用于在同一层菜单中扩展软键栏

F	
FDD	Feed Disable:进给禁止
FdStop	Feed Stop:进给停止
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
FM	故障信息
FUP	功能图（一种 PLC 编程方法）
FW	Firmware

G	
GEO	几何属性，例如几何值
GP	基本程序 (PLC)
GUD	Global User Data:全局用户数据

H	
HEX	十六进制数代号
HiFu	辅助功能
HMI	Human Machine Interface: SINUMERIK 操作介面
HSA	主轴驱动
HT	Handheld Terminal
HW	硬件

I	
IBN	调试
INC	Increment: 增量尺寸
INI	Initializing Data: 初始化数据
IPO	Interpolator

J	
JOG	Jogging: 点动模式

K	
KOP	梯形图（一种 PLC 编程方法）

L	
LED	Light Emitting Diode: 发光二极管
LMS	位置测量系统
LR	位置控制器

M	
Main	Main program: 主程序 (OB1, PLC)
MCP	Machine Control Panel: 机床控制面板
MD	机床数据

A.1 缩略语列表

M	
MDA	Manual Data Automatic:手动数据输入
MDS	Motor Data Set:电机数据组
MELDW	信息字
MKS	机床坐标系
MM	Motor Module
MPF	Main Program File:主程序（NC）
MPI	Multi Point Interface
MSTT	机床控制面板

N	
NC	Numerical Control:带有程序段处理、运行范围等等的数控系统
NCU	Numerical Control Unit:NC 硬件单元
NCK	Numerical Control Kernel
NCSD	NC Start Disable
NST	接口信号
NV	零点偏移
NX	Numerical Extension:轴扩展模块

O	
OB	PLC 中组织块
OEM	Original Equipment Manufacturer
OP	Operation Panel:操作面板

P	
PCU	PC Unit:PC 主机（计算机元件）
PG	编程器
PI	程序实例
PLC	Programmable Logic Control:可编程逻辑控制器
PN	PROFINET
PO	Power On

P	
POS	位置/定位
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
PTP	Point to Point: 点到点
PZD	过程数据: PPO 的过程数据部分

R	
REF	返回参考点功能
REPOS	再定位功能
RESU	Retrace support
RID	Read In Disable
RP	R 参数, 计算参数, 预定义用户变量

S	
SA	同步动作
SBL	Single Block: 单程序段
SBT	Safe Brake Test
SCC	Safety Control Channel
SCL	Structured Control Language
SD	Settingdatum: 设定数据
SDI	Safe Direction
SERUPRO	Search-Run by Program Test: 通过程序测试的程序段查找
SIC	Safety Info Channel
SKP	Skip: 跳至零件程序段末尾
SLP	Safe Limited Position
SLS	Safely Limited Speed
SMI	Sensor Module Integrated

A.1 缩略语列表

S	
SOS	Safe Operating Stop
SPF	Sub Program File:子程序（NC）
SS1	Safe Stop 1
SS2	Safe Stop 2
STO	Safe Torque Off
STW	控制字
SUG	砂轮外缘速度
SW	软件

T	
TCU	Thin Client Unit
TIA	Totally Integrated Automation
TM	Terminal Module (SINAMICS)
TO	Tool Offset:刀具补偿
TOA	Tool Offset Active:刀具补偿标识（文件类型）
TOFF	在线—刀具长度补偿
TRANSMIT	Transform Milling Into Turning:车床上铣削加工的坐标转换

U	
UP	子程序
USB	Universal Serial Bus

V	
VDI	NC 和 PLC 间的内部通讯接口

W	
WKS	工件坐标系
WPD	Work Piece Directory:工件目录
WZ	刀具
WZV	刀具管理

Z	
ZSW	(驱动) 状态字

索引

\$

\$A_OUT[n], 948
\$AA_ACC, 261
\$AA_ACT_INDEX_AX_POS_NO, 347
\$AA_BRAKE_CONDB, 720
\$AA_BRAKE_CONDM, 720
\$AA_BRAKE_STATE, 720
\$AA_COUP_ACT, 542, 579, 580
\$AA_COUP_CORR, 728
\$AA_COUP_CORR_DIST, 730
\$AA_COUP_OFFS, 543
\$AA_ESR_ENABLE, 959
\$AA_FGREF, 233
\$AA_FGROUP, 234
\$AA_FIX_POINT_ACT, 793
\$AA_FIX_POINT_SELECTED, 793
\$AA_IN_SYNC, 617
\$AA_JERK_COUNT, 972
\$AA_JERK_TIME, 972
\$AA_JERK_TOT, 972
\$AA_LEAD_P, 616
\$AA_LEAD_P_TURN, 616
\$AA_LEAD_SP, 616
\$AA_LEAD_SV, 616
\$AA_LEAD_V, 616
\$AA_MOTEND, 263, 301
\$AA_PROG_INDEX_AX_POS_NO, 347
\$AA_S, 401
\$AA_SCPAR, 264
\$AA_SYNC, 616, 692
\$AA_SYNCDIFF, 691
\$AA_TRAVEL_COUNT, 972
\$AA_TRAVEL_COUNT_HS, 972
\$AA_TRAVEL_DIST, 972
\$AA_TRAVEL_DIST_HS, 972
\$AA_TRAVEL_TIME, 972
\$AA_TRAVEL_TIME_HS, 972
\$AA_VLFCT, 505
\$AA_VMAXB, 505
\$AA_VMAXM, 505
\$AC_FGROUP_MASK, 234
\$AC_PATHACC, 102, 114
\$AC_PATHJERK, 113, 114, 115
\$AC_RETPOINT, 325
\$AC_SGEAR, 455, 482
\$AC_SMAXACC, 504
\$AC_SMAXACC_INFO, 504

\$AC_SMAXVELO, 504
\$AC_SMAXVELO_INFO, 504
\$AC_SMINVELO, 504
\$AC_SMINVELO_INFO, 504
\$AC_SPIND_STATE, 504
\$AN_ESR_TRIGGER, 960
\$P_COUP_OFFS, 543
\$P_FGROUP_MASK, 234
\$P_GEAR, 482
\$P_SEARCH_SGEAR, 456
\$P_SGEAR, 456
\$PA_FGREF, 233
\$PA_FGROUP, 234
\$SC_IS_SD_MAX_PATH_ACCEL, 101
\$SC_IS_SD_MAX_PATH_JERK, 113
\$SC_SD_MAX_PATH_ACCEL, 100
\$SC_SD_MAX_PATH_JERK, 112
\$TC_DPNT, 206
\$VA_COUP_OFFS, 543
\$VA_DESVAL_FILTERS_ACTIVE, 188
\$VA_DESVAL_FILTERS_DELAY_1, 188
\$VA_DESVAL_FILTERS_DELAY_2, 188
\$VA_DESVAL_FILTERS_DELTA_POS, 188
\$VA_SYNCDIFF, 691
\$VC_SGEAR, 455

“

“电机-负载” 齿轮箱, 167
“行程-时间” 挡块, 951

3

360°模数, 317

A

aaJerkCount, 972
aaJerkTime, 972
aaJerkTotal, 972
aaTravelCount, 972
aaTravelCountHS, 972
aaTravelDist, 972
aaTravelDistHS, 972
aaTravelTime, 972
aaTravelTimeHS, 972
ACC, 261, 492
ACCLIMA[FA], 738

ACN, 490
ACP, 490
ALF, 219
ASCALE, 328
ATRANS, 328

B

BRISK, 95
BRISKA, 96

C

CFC, 203
CFCIN, 203
CFTCP, 203
COARSEA, 295
CORROF, 839
CP-BASIC, 651
CPBC, 676
CP-COMFORT, 651
CPDEF, 663
CPDEF+CPLA, 665
CPDEL, 663
CPDEL+CPLA, 666
CPFMOF, 683
CPFMON, 682
CPFMSON, 680
CPFPOS, 678
CPFRS, 675
CPLDEF, 664
CPLDEL, 666
CPLDEN, 672
CPLINSC, 689
CPLINTR, 689
CPLNUM, 672
CPLOF, 670
CPLON, 669
CPLOUTSC, 690
CPLOUTTR, 690
CPLPOS, 679
CPLSETVAL, 674
CPMALARM, 706
CPMBRAKE, 699
CPMPRT, 687
CPMRESET, 684
CPMSTART, 686
CPMVDI, 701
CPOF, 668
CPOF+CPFPOS, 683
CPON, 667

CPRES, 715
CPSETTYPE, 709
CPSYNCOF, 693
CPSYNCOF2, 696
CPSYNCOV, 694
CPSYNFIP, 693
CPSYNFIP2, 696
CPSYNFIV, 694

D

DB10
DBX100.0 - 4, 765
DBX100.6, 765
DBX100.7, 765
DBX101.0 - 4, 765
DBX101.6, 765
DBX101.7, 765
DBX97.0 - 3, 765
DBX98.0 - 3, 765
DC, 490
DIACYCOFA, 275
DIAM90, 274
DIAM90/DIAM90A[AX], 269
DIAM90A, 275
DIAMCHAN, 275
DIAMCHANA, 275
DIAMCYCOF, 274
DIAMCYCOF/DIACYCOFA[AX], 269
DIAMOF, 274
DIAMOF, 275
DIAMON, 274
DIAMON/DIAMONA[AX], 268
DIAMONA, 275
DILF, 219
DITE, 217
DITS, 217
DRF, 835
DRIVE, 129
DRIVEA, 130

E

EG 轴组, 624
定义, 630
关闭, 636
启用, 631
删除, 637

F

FA, 234, 492
FB, 258
FCUB, 255
FINEA, 295
FL, 203
FLIM, 244
FLIN, 255
FNORM, 255
FPO, 255
FPRAOF, 491
FPRAON, 491
FRC, 256
FRCM, 256
FXS, 852
FXS-REPOS, 865
FXST, 852
FXSW, 852
FZ, 206

G

G25, 491
G26, 491
G33, 212
G331, 228
G332, 228
G335, 223
G336, 223
G63, 231, 434
G70, 155
G700, 155
G71, 155
G710, 155
G74, 41
G75, 787
G93, 204
G94, 205
G95, 205
G96, 208
G96、G961 下的主从切换, 211
G96、G961 下的转速限制, 210
G961, 208
G97, 209
G971, 209

I

IC, 490

IPOBRKA, 295
IPOENDA, 295
IR, 223

J

JOG
运行至固定点, 787
JR, 223

K

KR, 223
Kv 系数, 182

L

LFOF, 219
LFON, 219
LFPOS, 219
LFTXT, 219
LFWP, 219
LIMS, 491

M

M19, 404, 490
M3, 489
M4, 489
M40, 493, 509
M41, 493
M42, 493
M43, 493
M44, 493
M45, 493
M5, 489
M70, 491
MD10050, 83
MD10070, 83
MD10192, 461
MD10200, 142, 143, 144, 146, 153
MD10210, 142, 143, 144, 146, 177, 179, 329
MD10220, 148
MD10230, 148
MD10240, 150, 151, 157, 265
MD10250, 152
MD10260, 153, 155, 943
MD10270, 154, 337, 943
MD10290, 154
MD10292, 154

MD10450, 943
MD10460, 944, 952
MD10461, 944, 952
MD10470, 947
MD10471, 947
MD10472, 948
MD10473, 948
MD1048, 950
MD10480, 949, 951
MD10485, 941, 949, 950
MD10651, 224
MD10704, 247
MD10710, 210, 219
MD10720, 813
MD10721, 813
MD10722, 307
MD10735, 791, 797, 799, 808, 814, 817
MD10900, 337
MD10910, 336
MD10920, 337
MD10930, 336
MD10940, 339
MD1103, 868
MD1104, 868
MD1105, 868
MD11220, 153
MD11300, 37, 763
MD11310, 775
MD11320, 764, 772
MD11322, 833
MD11324, 774
MD11330, 763, 773, 826
MD11346, 772, 790, 798, 832
MD11410, 213, 445, 575, 706
MD11410 \$MN_SUPPRESS_ALARM_MASK 位 31, 626
MD11550, 230
MD11660 \$MN_NUM_EG, 624
MD12000, 240
MD12010, 241
MD12020, 238
MD12030, 239, 834
MD12040, 239
MD12050, 239
MD12060, 241
MD12070, 242
MD12080, 242
MD12082, 240
MD12090, 230
MD12100, 242
MD12200, 254
MD12202, 254
MD12204, 254
MD1230/1231, 868
MD17900, 774
MD18400, 583, 587
MD18402, 583, 587
MD18403, 584, 587
MD18404, 584, 587
MD18406, 584, 587
MD18408, 584, 587
MD18409, 584, 587
MD18410, 584, 587
MD18450, 654
MD18452, 654
MD18860, 972
MD18960, 95, 111, 293
MD20070, 527
MD20090, 489
MD20092, 486
MD20100, 209, 212, 270
MD20110, 273, 576, 611, 614, 617, 618, 619, 818
MD20112, 273, 575, 611, 614, 617, 618, 619
MD20150, 95, 110, 129, 154, 265, 273, 818
MD20151, 818
MD20200, 257
MD20201, 257
MD20360, 271, 808
MD20500, 97, 98
MD20600, 109
MD20602, 105, 106
MD20610, 100
MD20620, 773
MD20621, 773
MD20624, 272, 776, 781
MD20700, 77, 820
MD20750, 210
MD20850, 405
MD20900, 588, 594
MD20905, 588, 591, 593, 602
MD21150, 330
MD21155, 749, 784
MD21158, 132, 752, 785
MD21159, 132, 752, 785
MD21160, 750, 784
MD21165, 749, 784
MD21166, 132, 751, 785
MD21168, 132, 752, 785
MD21220, 250
MD21230, 249
MD21300, 523, 564
MD21310, 564
MD21320, 541, 564
MD21330, 564
MD21340, 525, 564

- MD21380, 967
MD21381, 967
MD22000, 405
MD22010, 405
MD22020, 405
MD22030, 405
MD22240, 235
MD22410, 236, 244, 265, 740
MD27100, 273
MD30100, 162
MD30110, 162, 877
MD30120, 162
MD30130, 161, 163, 609, 613, 615
MD30132, 609
MD30200, 160, 171
MD30210, 163, 172
MD30220, 163, 172
MD30230, 164, 172, 877
MD30240, 161, 164, 172
MD30242, 67, 161, 165, 166, 172
MD30244, 172
MD30250, 73, 172
MD30260, 172
MD30270, 172
MD30300, 171, 173, 177, 179, 265, 321, 329
MD30310, 171, 318, 321, 322, 329
MD30320, 171, 319, 323, 329
MD30330, 65, 171, 323, 324
MD30340, 65, 171, 318, 319, 323, 324
MD30350, 161
MD30450, 385
MD30455, 65, 492, 533, 547, 566, 721, 727
MD30460, 270, 274, 306, 311, 312
MD30500, 336, 344
MD30503, 319
MD30505, 344
MD30550, 38, 527
MD30552, 284
MD30600, 791
MD31000, 173, 177, 179
MD31010, 173
MD31020, 83, 173, 177, 179
MD31025, 177
MD31030, 171, 173, 177
MD31040, 173, 177, 179, 196, 415, 471
MD31044, 168, 173, 460
MD31050, 167, 173, 174, 177, 179, 458, 469
MD31060, 167, 173, 174, 177, 179, 458
MD31064, 168, 171, 460
MD31066, 168, 171, 460
MD31070, 173, 177, 179
MD31080, 173, 177, 179
MD31090, 155, 763, 764, 773, 826, 833
MD31122, 438
MD31123, 438
MD31200, 154
MD32000, 141, 171, 203, 507, 737, 827, 834
MD32000 \$MA_MAX_AX_VELO, 629
MD32010, 750
MD32020, 749, 774, 782
MD32040, 293, 335, 750, 776
MD32050, 293, 335, 749, 776
MD32060, 260, 284, 292, 830
MD32080, 773
MD32084, 776, 834
MD32090, 838
MD32100, 170, 374
MD32200, 182, 375, 564
MD32210, 198
MD32220, 198
MD32250, 170
MD32300, 94, 104, 105, 106, 131, 260, 261, 293, 737, 751, 834, 843
\$MA_MAX_AX_ACCEL, 629
MD32300 \$MA_MAX_AX_ACCEL, 629
MD32310, 104, 105
MD32320, 261
MD32400, 121, 375, 566
MD32402, 121, 186, 187, 376, 566
MD32408, 187
MD32409, 187
MD32410, 121, 187, 376, 566
MD32411, 122, 187
MD32412, 121, 187
MD32414, 187
MD32420, 96, 110, 111, 131, 376, 566, 751
MD32430, 110, 131, 294, 376, 566, 751
MD32431, 109, 115, 294
MD32432, 116
MD32433, 104
MD32434, 103, 104, 115
MD32435, 115
MD32439, 117
MD32610, 375, 564
MD32620, 191, 375, 547, 564
MD32630, 191
MD32640, 193
MD32650, 375, 564
MD32711, 155
MD32800, 375, 564
MD32810, 191, 375, 376, 564
MD32890, 194, 195
MD32900, 183, 376
MD32910, 183, 193, 376, 565

- MD32950, 197
MD33000, 182, 375
MD33060, 973
MD34000, 53, 64, 65
MD34010, 36, 48, 50, 51, 62, 75
MD34020, 48, 51
MD34040, 50, 52, 62, 63, 365, 438
MD34050, 50
MD34060, 54, 63, 65, 416
MD34070, 56, 64, 365
MD34080, 56, 369, 377, 390, 529, 568
MD34090, 56, 59, 60, 74, 75, 76, 77, 172, 369, 377, 390, 529, 568
MD34092, 52
MD34093, 54
MD34100, 56, 64, 75, 77, 363, 365, 367, 369, 373, 391, 529, 568
MD34102, 66, 67
MD34104, 69
MD34110, 39, 372
MD34200, 60, 67, 69, 75, 78, 438
MD34210, 72, 73, 74, 75, 76, 814
MD34300, 63
MD34320, 59, 172, 173
MD34330, 79, 365, 369, 373
MD34990, 204
MD35000, 486, 493, 523, 527
MD35010, 442, 446, 463, 471, 478
MD35012, 441, 471, 475
MD35014, 480
MD35020, 428
MD35030, 429
MD35035, 405, 481, 484
MD35040, 400
MD35090, 441
MD35092, 446
MD35100, 141, 502
MD35110, 441, 510
MD35112, 446
MD35120, 441, 510
MD35122, 446
MD35130, 441, 446, 448, 501
MD35135, 441
MD35140, 211, 213, 441, 446, 448, 500
MD35150, 51, 407, 409, 499, 502
MD35200, 399, 410, 412, 416, 441, 476
MD35210, 400, 409, 412, 413, 416, 417, 441, 476
MD35212, 446
MD35220, 123, 127, 567
MD35230, 123, 127, 567
MD35240, 126
MD35242, 127, 567
MD35300, 83, 409, 412, 416, 417, 442, 476, 492
MD35310, 442, 471, 475
MD35350, 416, 475
MD35400, 463, 465
MD35410, 463, 465
MD35430, 463, 465
MD35500, 401, 407
MD35510, 407, 498
MD35550, 442, 447
MD35590, 442, 457
MD36000, 414, 418, 508, 509
MD36010, 414, 418, 508, 509
MD36012, 371
MD36030, 370
MD36060, 400, 408, 498
MD36100, 320, 321
MD36110, 320, 321
MD36120, 320
MD36130, 320
MD36210, 170
MD36300, 82, 83
MD36302, 412
MD36500, 160, 372
MD36510, 160
MD36600, 723
MD36933, 778
MD37002, 860
MD37010, 855
MD37012, 860
MD37020, 856, 869
MD37030, 855
MD37040, 855
MD37060, 854, 858, 860
MD37100, 377, 387
MD37110, 359, 365, 378, 387, 392, 393
MD37120, 360, 375, 387, 392, 393
MD37130, 360, 375, 387, 393
MD37135, 361
MD37140, 363, 368, 384, 385
MD37150, 360, 361, 369
MD37160, 601, 612
MD37200, 535, 565, 693
MD37200 \$MA_COUPLE_POS_TOL_COARSE, 627, 640
MD37200 \$MN_COUPLE_POS_TOL_COARSE, 627
MD37202, 696
MD37210, 535, 565, 693
MD37210 \$MA_COUPLE_POS_TOL_FINE, 627
MD37210 \$MA_COUPLE_POS_TOL_FINE, 627, 640
MD37212, 696
MD37220, 535, 565, 693
MD37230, 535, 568, 693

MD37400, 886
MD37500, 291, 967
MD37510, 290, 291
MD37511, 290, 291
MD37550 \$MA_EG_VEL_WARNING, 629
MD37560 \$MA_EG_ACC_TOL, 629
MD43108, 646
MD7200, 568
MD9004, 145
MD9011, 145
MELDW
 位 9, 961

O

OS, 904
OSB, 907
OSCILL, 914
OSCTRL, 905, 906
OSE, 907
OSNSC, 906
OSP, 905
OST, 905
OVR, 243
OVRA, 243, 492

P

p0888, 954, 956, 957
p0891, 955, 956
p0892, 955, 956
p0893, 956
p1240, 958
p1248, 958
p2161, 959
PCOF, 491
PI 控制器, 739
PLC 轴, 305
 固定指定给 PLC 的轴, 306
 通过 SINU_TraversePosAxis 启动, 309
 只由 PLC 控制的轴, 306
POLF, 219
POLFMASK, 219
POLFMLIN, 219
POSP, 914
PRESETON, 41

R

REPOS 偏移, 864

S

S..., 490
SCPARA, 263
SD41010, 763, 773, 826
SD41050, 334, 756
SD41100, 207, 293, 317, 335, 748, 776
SD41110, 749, 774
SD41120, 749
SD41130, 330, 749, 774
SD41200, 749, 782
SD41500, 942
SD41501, 942
SD41502, 942
SD41503, 942
SD41504, 942
SD41505, 942
SD41506, 942
SD41507, 942
SD41520, 944, 952
SD41521, 944, 952
SD41522, 944
SD41523, 944
SD41524, 945
SD41525, 945
SD41526, 945, 952
SD41527, 945, 952
SD42010, 216, 218
SD42100, 247, 313, 831, 902
SD42101, 247, 313, 902
SD42300, 524, 564
SD42500, 100
SD42502, 100
SD42510, 112
SD42512, 112
SD42600, 207, 293, 297, 335, 495, 776
SD42690, 802, 808
SD42691, 802, 809
SD42692, 802, 805, 806, 809, 810
SD42693, 804, 809
SD42694, 804, 809
SD42800, 486
SD43100, 615
SD43102, 610
SD43104, 610
SD43106, 610
SD43108, 610
SD43210, 211, 213
SD43220, 211
SD43230, 210
SD43235, 503

SD43240, 404, 490
SD43250, 404
SD43300, 207, 293, 297, 335, 495, 776
SD43320, 796, 799
SD43400, 320
SD43410, 320
SD43420, 320
SD43430, 320
SD43500, 868
SD43510, 868, 869
SD43520, 869
SD43600, 304
SD43770, 907
SD43790, 907
SERUPRO ASUB, 864
SETMS, 489
SOFT, 109
SOFTA, 111
SPCOF, 491
SPCON, 491
SPI, 493
SPOS, 403, 490
SPOSA, 490
SVC, 490

T

TANG, 888
TANGDEL, 892
TANGOF, 892
TANGON, 891
TLIFT, 890
TRAANG, 606
TRAILOF, 577
TRAILON, 576
TRANS, 328

V

VELOLIM, 492
VELOLIMA[FA], 738

W

WAITC, 677
WAITP, 296
 往复轴, 913
WAITS, 492

报

报警 10653, 361

闭

闭环控制, 181

编

编程方式, 528

并

并行定位轴
 由 PLC 启动, 308

参

参考轴
 用于 G96 / G961 / G962, 267
参考坐标, 675
参数组
 在轴上, 189

测

测量系统, 35, 160

插

插补, 620
插补过程, 313
插补器
 轨迹, 284
 轴, 285

差

差速, 560

程

程序段切换
 定位轴类型 1, 298
 定位轴类型 2, 299
程序段转换标准, 676

齿

齿轮档, 441
 M70, 479
 手动设定, 449
 通过 PLC 设定, 448
齿轮档选择
 自动, 509

垂

垂直轴, 867

存

存储器配置
 曲线表, 583
 通用耦合, 654

达

达到同步, 532

单

单程序段
 定位轴类型 1, 313
 定位轴类型 2, 313
 定位轴类型 3, 313
单位制, 150, 155
单轴
 NC 控制的扩展回退, 291
 NC 控制的扩展停止, 290
 应用, 287
 由 PLC 控制轴, 285
单轴动态响应, 259

挡

挡块
 对, 935
 范围, 935
 位置, 942
 信号, 935
挡块信号
 单独输出, 936
 挡块信号定时输出, 949
 负, 946

关联输出, 939
硬件指定, 947
正, 946

点

点按模式, 753, 757

电

电子齿轮箱, 621, 622
 系统变量, 638

定

定位精度, 144
定位轴, 143, 278
 并行, 283, 305
 刀具补偿, 297
 空运行进给率, 313
 类型, 281
 通道专用信号, 305
 轴类型, 281
 轴专用信号, 306
 最大数量, 294
定位轴动态响应, 293
定义
 EG 轴组, 630

动

动态响应限制, 737
动态响应自适应功能, 182

独

独立的副主轴插补器, 523
独立联动轴, 573

读

读取偏移, 533

端

端面齿, 343
端面轴, 267
 尺寸设定, 274

断

断续手动运行, 757
断续运行, 762

对

对停止的反应, 611

多

多个端面轴
跨通道取轴, 275

反

反比时间进给率 (G93), 204

分

分度位置
数量, 337
分度位置表, 336
分度轴
编程, 345
编码位置, 345
单位制, 337

负

负
挡块, 935

副

副主轴
重新同步, 551
副主轴插补器, 523

跟

跟随轴叠加, 656

工

工作区域限制
圆弧手动运行时, 806

公

公差
主轴转速, 499

攻

攻丝
带弹性卡头, 434

固

固定点位置, 791
固定定义的同步, 522
固定进给值, 253

关

关键字, 656

归

归档, 620

轨

轨迹
插补器, 284
进给率 F, 202
最大速度, 203
轨迹进给率, 143
轨迹速度
最大值, 244

行

行程开关信号, 935
提前时间/延迟时间, 944

恒

恒定切削速度, 208
恒定转速, 209

换

换向点, 899

回

回参考点
 使用增量测量系统, 45
 旋转绝对值编码器, 79
回参考点方式, 35
回参考点运行, 838
回退
 方向, 螺纹切削, 220
回转轴
 360°模数, 317
 尺寸单位, 316
 进给率, 316
 镜像, 331
 绝对位置编程, 323, 326
 模态转换, 326
 软件限位开关, 330
 调试, 329
 增量位置编程, 325, 328
 轴地址, 315

机

机床零点, 35

基

基本单位制, 150
基于半径的数据, 269

激

激活同步, 528

极

极限
 轨迹轴速度, 203

急

急动度
 上调, 随速度提高的, 116

计

计算精度, 145

加

加速模式, 739

监

监控
 同步运行, 691

角

角度偏移“位置副主轴”, 529

接

接口信号“进给停止/主轴停止”, 533

进

进给, 899
进给倍率, 304
 可编程, 243
进给禁止
 通道专用, 236
进给率
 (G93、G94、G95), 204
 -补偿, 238
 倒角/倒圆, 256
 反比时间进给率(G93), 204
 轨迹进给率 F, 202
 进给/主轴停止, 236
 进给禁止, 236
 空运行进给率, 246
 -类型, 201
 每齿, 206
 线性(G94), 205
 旋转(G95), 205
进给轴禁止, 237

精

精插补, 182

空

空运行进给率, 246

控

控制动态响应, 620
控制性能, 564

跨

跨通道取轴, 716

快

快速停止, 723
快速移动
-补偿, 238

扩

扩展监控
龙门轴, 360

连

连续手动运行, 753

联

联动, 571
编程, 576
动态响应限制, 581
接口信号, 578
控制动态响应, 581
启用/关闭, 574
剩余行程, 575
联动轴, 572
联动轴作为引导轴, 573
联动组, 572
定义和启用, 576
关闭, 577

临

临时曲线表, 586

龙

龙门轴
与联动的区别, 386

轮

轮廓手轮, 832

螺

螺纹
切削, 219
螺纹切削 G33, 212

滤

滤波器类型
滑动求平均值, 121
滤波器链条
可切换的, 185
切换, 187

每

每齿进给量, 206

模

模拟引导值, 609
模拟轴, 161
模数引导轴, 600

耦

耦合
关闭, 668
耦合的状态, 580, 620
耦合方式, 615
耦合规律, 655, 671
耦合基准, 674
耦合类型, 709
耦合模块, 655
创建, 662
删除, 663
耦合系数, 671
分母, 672
分子, 672

偏

偏移
引导值, 688

起

起始值, 594

切

切换精度

挡块信号, 948

切向控制, 885

求

求平均值

双倍, 121

球

球螺纹, 223

曲

曲线表, 583, 673

编程, 588

存储器类型, 585

存储器优化, 585

刀具半径补偿, 584

多次使用, 600

覆盖, 586

激活, 599

接口信号, 601

禁用, 599

默认存储器类型, 588

内存不足, 585

删除, 586

在运行方式下的特性, 601

转换, 606

曲线段, 584

软

软件

挡块, 935

设

设定齿轮档, 443

设定值输出, 159

设定值系统, 159

进给轴和主轴

功能手册, 01/2023, A5E47437747F AF

实

实际值补偿, 161

实际值采集, 160

实际值处理, 169

实际值显示, 620

使

使用 SINU_CtrlStarDelta 进行星/三角切换, 451

手

手动退刀, 812

手轮

JOG 方式下的运行, 763

分配, 765

行程设定, 772, 832

速度设定, 772, 832

通过 HMI 选择, 765

手轮连接

PROFINET, 845

输

输入精度, 145

速

速度, 141

缩

缩放

引导值, 688

替

替代零脉冲, 79

调

调节方向, 169

调整电机/负载齿轮比, 167

停

停止, 953

通

通道, 279
通过 PLC 取轴, 307
通过手轮设定行程, 832

同

同步, 739
 定义新的, 538
 跟随轴, 621
 固定定义, 538
同步模式, 680
同步位置
 跟随轴, 678
 引导轴, 679
同步运行
 -补偿, 729
 监控, 691
 偏差, 691, 724
同步运行监控
 2 级, 695
同步运行偏差, 626
 查询, 629

退

退回, 953

往

往复运动, 899
 带同步运动, 925
 连续进给, 899
 异步, 899
往复轴, 899

未

未应答的齿轮档切换, 461

位

位置差接通, 196

位置控制回路, 181
位置偏移
 在主轴同步中, 550

无

无火花光磨, 899

物

物理量, 147

系

系统变量, 659

显

显示精度, 145

限

限制, 594

线

线性进给率 (G94) , 205

相

相关联动轴, 573
相位滤波器, 194

虚

虚拟引导轴, 609

旋

旋转进给率 (G95) , 205

引

引导值
 对象, 609
 耦合, 608

- 偏移, 688
- 缩放, 688
- 引导值耦合
 - 关闭, 614
 - 接口信号, 619
 - 接通, 612
 - 在运行方式下的特性, 617
- 引导轴
 - 定义, 664
 - 关闭, 670
 - 启用, 669
 - 删除, 666
- 引导轴/主轴模块, 664

用

- 用户定义的同步: , 522

由

- 由 PLC 控制的轴
 - MD30460 位 6 和 位 7 对应的控制特性, 312
 - 控制特性, 311

圆

- 圆弧半径
 - 圆弧手动运行时, 809
- 圆弧段加工, 804
- 圆弧运行
 - 在 JOG 方式下, 801
- 圆心
 - 圆弧手动运行时, 808

运

- 运动范围限制, 319
- 运动结束条件
 - 程序段搜索时, 304
 - 单轴时, 263
- 运动特性, 313
- 运行到固定挡块, 851
 - 程序段搜索, 862
 - 到达固定挡块, 855
 - 定位轴, 867
 - 功能中断, 866
 - 监控窗口, 856
 - 轮廓监控, 867
 - 取消, 860
 - 选择, 854

- 运行范围, 143
- 运行键, 753
- 运行请求, 766
- 运行使能, 79
- 运行指令, 767
- 运行至固定点
 - 使用 G75, 787
 - 在 JOG 方式下, 787

在

- 在主轴/进给轴模式下进行动态响应编程, 566

长

- 长按模式, 753, 757

诊

- 诊断和优化资源使用, 602

正

- 正
 - 挡块, 935

直

- 直接连接在刀具上的编码器, 168
- 直径编程
 - 通道专用, 274
 - 轴专用, 275

中

- 中间齿轮比, 168, 460

重

- 重新同步, 550

轴

- 轴
 - 插补器, 285
 - 用于辅助运动, 277
- 轴分配, 594

轴类型

- 针对定位轴, 281

主

主运行

- 轴, 262

主轴

- 补偿, 238, 241
- 齿轮档 0, 453
- 设定转速, 499
- 引导值耦合中, 612
- 转速限值, 504
- 最大转速, 501
- 最小转速, 500

主轴禁止, 237

主轴同步, 521

- 位置偏移, 550
- 折线式加速度特性曲线, 567

主轴同步方式, 522

主轴转速, 143

转

转速

- 前馈控制, 191

转速控制回路, 181

转速设定值输出, 169

转速设定值调整, 170

自

自动方式下的手轮叠加

- 行程设定, 824
- 速度叠加, 825

自控单轴运动

- NCK 响应, 309
- PLC 动作, 308

最

最大轴加速度, 738

最大轴速度, 738

坐

坐标系, 574