# NJ 直线插补与圆弧插补

一、　轴组基本知识

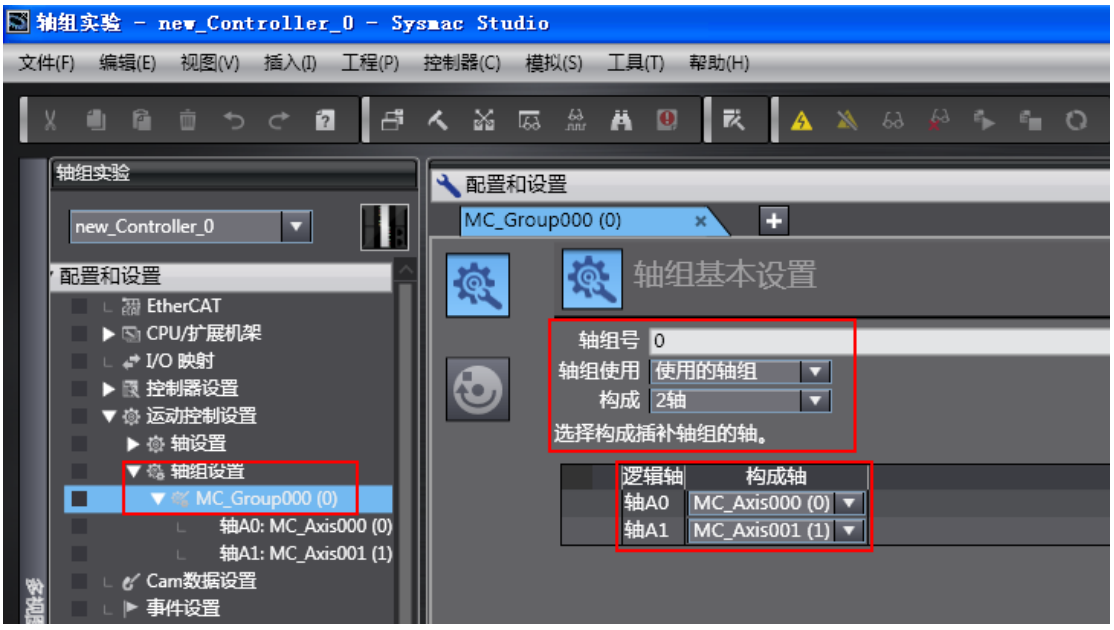
轴组主要用于多轴协调控制，例如直线插补、圆弧插补。轴组由 2~4 个轴构成，NJ 运动控制器最多可添加 32 组轴组。

1、登录轴组

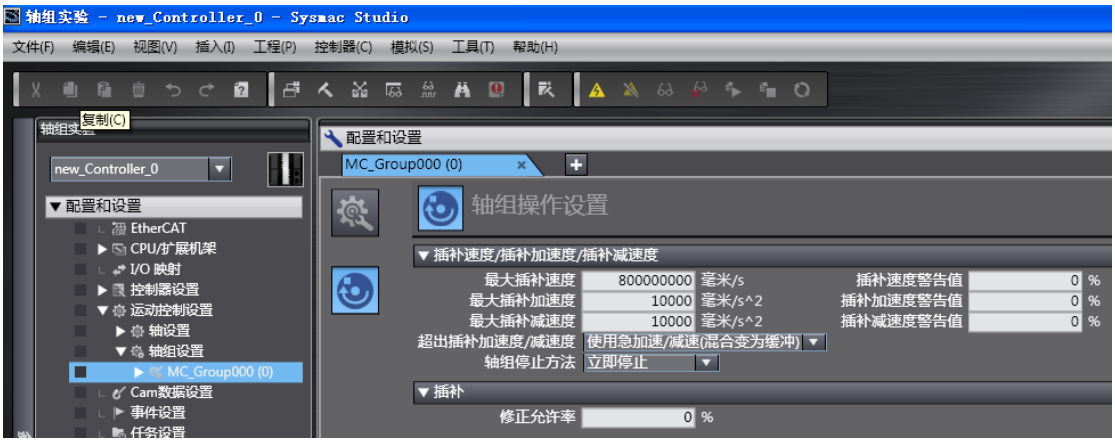
- 确定进行插补的轴组，轴组即 _MC_GRP[***]。
- 通过该轴组变量的轴构成指定轴的构成。可指定的轴数为 2～4 轴。
- 通过该轴组变量的轴选择指定进行插补的轴的组合。
- 轴不使用轴号（轴 0～轴 63），而是使用逻辑轴（轴 A0～轴 A3）。
- 通过轴选择以前移方式向逻辑轴 A0～A3 分别指定轴号 0～63。

| 逻辑轴 | 轴号 |
|---|---|
| 轴 A0 | 轴 0 ~ 轴 63 |
| 轴 A1 | 轴 0 ~ 轴 63 |
| 轴 A2 | 轴 0 ~ 轴 63 |
| 轴 A3 | 轴 0 ~ 轴 63 |

**轴组基本设置**

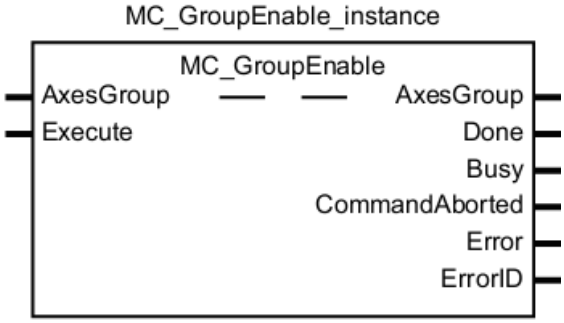



**轴组参数设置**




2、启动轴组

- 将轴组的各构成轴置为伺服 ON 状态，并使其处于原点确定状态。
- 执行 MC_GroupEnable( 启用轴组 ) 指令，使登录的轴组有效。




**注：AxesGroup：指定轴组；Execute：启动条件，上升沿有效**

基本功能：

• MC_GroupEnable(启用轴组)指令将通过 AxesGroup(轴组)指定的轴组变为 GroupEnable(启用轴组)状态。

• 变为 GroupEnable( 启用轴组 ) 状态后,轴组便可以执行所有多轴协调指令。

• 可指定到轴组中的轴的种类只能是“伺服轴”和“虚拟伺服轴”，若指定了其他轴种类，将出现异常。

• 属于轴组的所有轴必须处于停止状态。

轴变量 Status.Disabled( 轴无效 ) 或 Status.Standstill( 停止中 ) 为 TRUE 时，即表明轴处于停止状态。

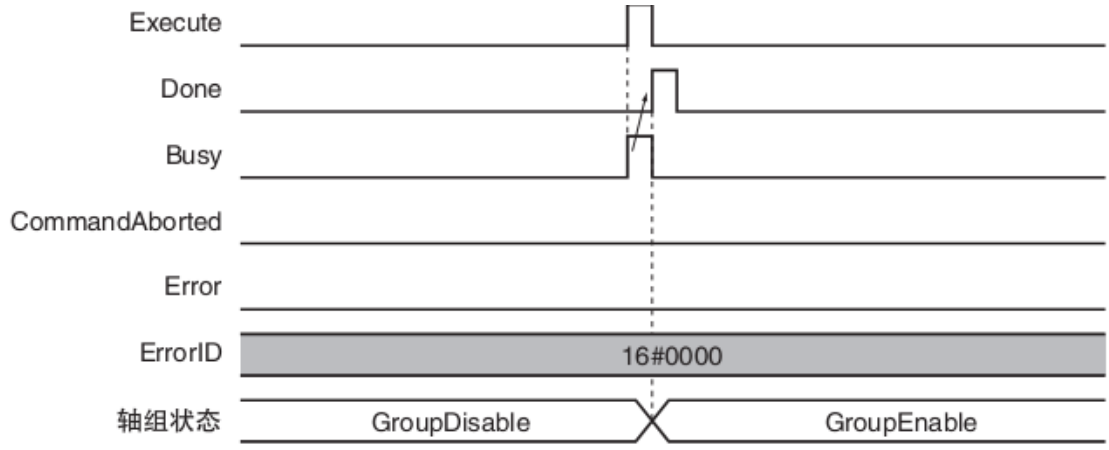
• 如果存在已经属于其他轴组、且该轴组已启用的轴，则不能执行 MC_GroupEnable( 启用轴组 ) 指令，会发生异常。

• 轴组启用后，属于轴组的轴的状态为 “ 多轴协调动作中 ”。

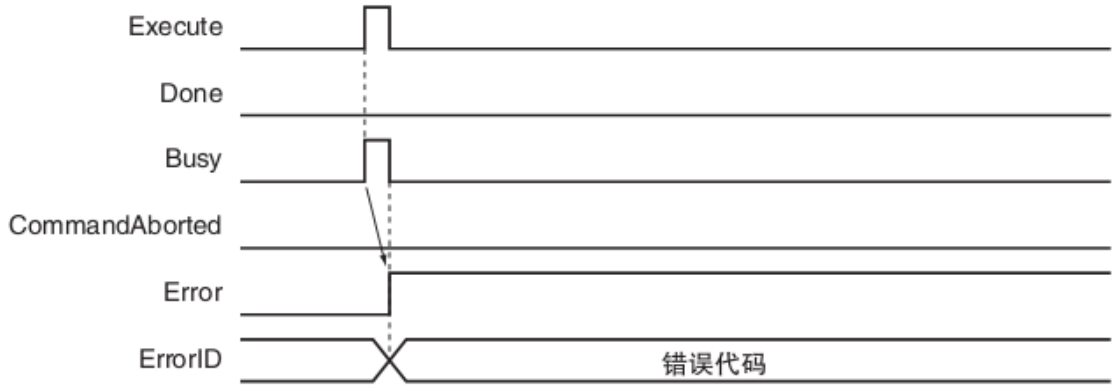
轴变量 Status.Coordinated( 多轴协调动作中 ) 变为 TRUE。

• 使轴组无效的条件有执行 MC_GroupDisable( 不启用轴组 ) 指令、切换到程序模式使运行停止、以及开始 MC 试运行。

执行时序图-正常结束时




执行时序图-异常结束时



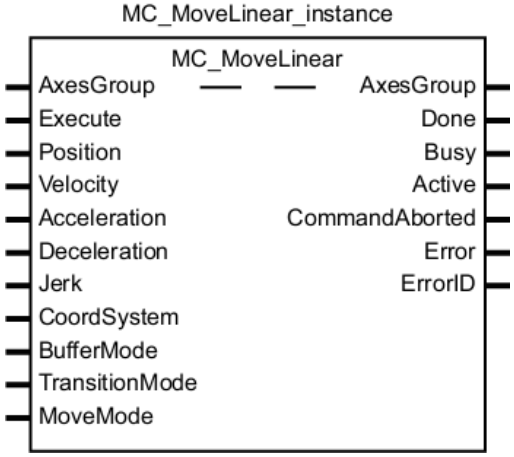

TIPS：轴组即使发生异常，轴组中包含的各轴并不会有异常。

二、 直线插补与圆弧插补

直线插补及圆弧插补属于多轴协调控制，执行多轴协调控制前必须先登录轴组，并启动轴组（使用 MC_GroupEnable）。

**注意：使用直线插补或圆弧插补操作，轴组中包含编码器轴或虚拟编码器轴时，将发生“不符合轴类型 （ 错误代码：543D Hex)”的错误，结束动作。请务必选择伺服轴或虚拟伺服轴。**

**直线插补**



使用注意事项：

- 轴组中的任意一个构成轴原点未确定时，将发生“原点未确定状态下的指令启动异常 （ 异常代码：5466 Hex)”的错误。
- 进行 2 轴圆弧插补时，属于该轴组的任意一个逻辑轴的极限输入“ON”时，将无法启动指令。

| 输入变量 | 名 称 | 数据类型 | 有效范围 | 初始值 | 内 容 |
|---|---|---|---|---|---|
| Execute | 启动 | BOOL | TRUE, FALSE | FALSE | 在上升沿开始指令。 |
| Position | 目标位置 | ARRAY [0..3] OF LREAL | 负数、正数、“0” | 0 | 指定直线插补的目标位置。<br>单位为 [ 指令单位 ]。*1 |
| Velocity*2 | 目标速度 | LREAL | 正数 | 0 | 指定目标速度。<br>单位为 [ 指令单位 /s] 。*1 |
| Acceleration | 加速度 | LREAL | 正数或 “0” | 0 | 指定加速度。<br>单位为 [ 指令单位 /$s^2$]。*1 |
| Deceleration | 减速度 | LREAL | 正数或 “0” | 0 | 指定减速度。<br>单位为 [ 指令单位 /$s^2$]。*1 |
| Jerk | 跃度 | LREAL | 正数或 “0” | 0 | 指定跃度。<br>单位为 [ 指令单位 /$s^3$]。*1 |
| CoordSystem | 坐标系 | _eMC_ COORD_ SYSTEM | 0 : _mcACS | 0 *3 | 指定坐标系。<br>0：轴坐标系 (ACS) |
| BufferMode | 缓存<br>模式选择 | _eMC_ BUFFER_ MODE | 0 : _mcAborting<br>1 : _mcBuffered<br>2 : _mcBlendingLow<br>3 : _mcBlendingPrevious<br>4 : _mcBlendingNext<br>5 : _mcBlendingHigh | 0 *3 | 指定多重启动运动指令时的动作。<br>0：中断<br>1：等待<br>2：以低速合并<br>3：以前一个速度合并<br>4：以后一个速度合并<br>5：以高速合并 |
| Transition Mode | 过渡模式<br>( 切换模式 ) | _eMC_ TRANSITION_ MODE | 0 : _mcTMNone<br>10 : _mcTMCornerSuperimposed | 0 *3 | 指定动作的路径。<br>0：过渡无效<br>10：附加角 |
| MoveMode | 移动方法选择 | _eMC_MOVE_ MODE | 0 : _mcAbsolute<br>1 : _mcRelative | 0 *3 | 选择移动方法。<br>0：绝对值定位<br>1：相对值定位 |

| 输入输出变量 | 名 称 | 数据类型 | 有效范围 | 内 容 |
|---|---|---|---|---|
| AxesGroup | 轴组 | _sGROUP_REF | - | 指定轴组。*1 |

重要参数说明：

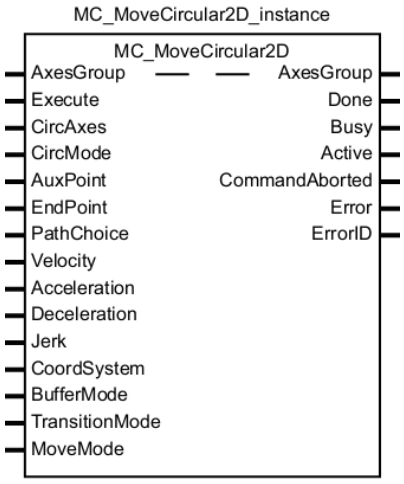
1、Position，所用轴组中的轴数少于 4 个轴时，也必须定位为 1×4 的数组；

2、Velocity，请务必设定目标速度。不作设定就执行动作，将发生异常；

3、BufferMode(缓存模式选择)，指定前一个插补动作和本次插补动作的连接方式。

| 缓存模式选择 | | 说 明 |
|---|---|---|
| 中断 | | 立即中止当前正在执行的指令，切换为本指令。<br>轴的动作方向因指令切换而反转时，根据轴参数中的 “ 反转时的动作 ” 进行反转。 |
| 等待 | | 当前正在执行的指令正常完成后，已缓存的本指令自动启动。 |
| 合并 | | 以当前正在执行的指令到达目标位置时的速度 ( 中继速度 ) 为启动速度，连续使已缓存的本指令动作。变更当前正在执行的指令的动作，确保以中继速度到达目标位置。中继速度的指定方法分为如下 4 种。<br>此外，还有过渡指定 ( 后述 ) 作为合并的选项。 |
| | 以低速合并 | 当前正在执行的目标速度与已缓存的目标速度中，以速度较低者为中继速度。 |
| | 以前一个速度合并 | 以当前正在执行的目标速度为中继速度。 |
| | 以后一个速度合并 | 以已缓存的本指令的目标速度为中继速度。 |
| | 以高速合并 | 当前正在执行的目标速度与已缓存的目标速度中，以速度较高者为中继速度。 |

## 圆弧插补




使用注意事项：

- 轴组中的任意一个构成轴原点未确定时，将发生“原点未确定状态下的指令启动异常 （ 异常代码：5466 Hex)”的错误。
- 进行 2 轴圆弧插补时，属于该轴组的任意一个逻辑轴的极限输入“ON”时，将无法启动指令。

| 输入变量 | 名 称 | 数据类型 | 有效范围 | 初始值 | 内 容 |
|---|---|---|---|---|---|
| Execute | 启动 | BOOL | TRUE, FALSE | FALSE | 在上升沿开始指令。 |
| CircAxes | 圆弧轴指定 | ARRAY [0,1] OF UINT | 0 ~ 3 | 0 | 指定进行圆弧插补的轴。<br>0：轴 A0<br>1：轴 A1<br>2：轴 A2<br>3：轴 A3 |
| CircMode | 圆弧插补模式 | _eMC_CIRC_MODE | 0: _mcBorder<br>1: _mcCenter<br>2: _mcRadius | 0 *1 | 指定圆弧插补的方法。<br>0：指定为通过点<br>1：指定为中心点<br>2：指定为半径 |
| AuxPoint | 辅助点 | ARRAY [0,1] OF LREAL | 负数、正数、“0” | 0 | 指定通过点位置 / 中心位置 / 半径。<br>单位为 [ 指令单位 ]。*2 |
| EndPoint | 终点 | ARRAY [0,1] OF LREAL | 负数、正数、“0” | 0 | 指定目标位置。<br>单位为 [ 指令单位 ]。*2 |
| PathChoice | 路径选择 | _eMC_CIRC_PATHCHOICE | 0 : _mcCW<br>1 : _mcCCW | 0 *1 | 指定路径方向。<br>0：CW<br>1：CCW |
| Velocity *3 | 目标速度 | LREAL | 正数 | 0 | 指定目标速度。<br>单位为 [ 指令单位 /s] 。*2 |
| Acceleration | 加速度 | LREAL | 正数或 “0” | 0 | 指定加速度。<br>单位为 [ 指令单位 /$s^2$]。*2 |
| Deceleration | 减速度 | LREAL | 正数或 “0” | 0 | 指定减速度。<br>单位为 [ 指令单位 /$s^2$]。*2 |
| Jerk | 跃度 | LREAL | 正数或 “0” | 0 | 指定跃度。<br>单位为 [ 指令单位 /$s^3$]。*2 |
| Coord System | 坐标系 | _eMC_COORD_SYSTEM | 0 : _mcACS | 0 *1 | 指定坐标系。<br>0：轴坐标系 (ACS) |
| BufferMode | 缓存模式选择 | _eMC_BUFFER_MODE | 0 : _mcAborting<br>1 : _mcBuffered<br>2 : _mcBlendingLow<br>3 : _mcBlendingPrevious<br>4 : _mcBlendingNext<br>5 : _mcBlendingHigh | 0 *1 | 指定多重启动运动指令时的动作。<br>0：中断<br>1：等待<br>2：以低速合并<br>3：以前一个速度合并<br>4：以后一个速度合并<br>5：以高速合并 |
| Transition Mode | 过渡模式 ( 切换模式 ) | _eMC_TRANSITION_MODE | 0 : _mcTMNone<br>10 : _mcTMCornerSuperimposed | 0 *1 | 指定动作的路径。<br>0：过渡无效<br>10：附加角 |
| MoveMode | 移动方法选择 | _eMC_MOVE_MODE | 0 : _mcAbsolute<br>1 : _mcRelative | 0 *1 | 选择移动方法。<br>0：绝对值定位<br>1：相对值定位 |

| 输入输出变量 | 名 称 | 数据类型 | 有效范围 | 内 容 |
|---|---|---|---|---|
| AxesGroup | 轴组 | _sGROUP_REF | – | 指定轴组。*1 |

| 输出变量 | 变为 TRUE 的时间 | 变为 FALSE 的时间 |
|---|---|---|
| Done | 定位完成后 | ・Execute 为 TRUE 时，与 Execute 的 FALSE 同时<br>・Execute 为 FALSE 时，1 个周期后 |
| Busy | Execute 的上升沿 | ・Done 变为 TRUE 时<br>・Error 变为 TRUE 时<br>・CommandAborted 变为 TRUE 时 |
| Active | 轴开始移动时 | ・Done 变为 TRUE 时<br>・Error 变为 TRUE 时<br>・CommandAborted 变为 TRUE 时 |
| CommandAborted | ・利用其它指令多重启动运动指令 ( 中断 ), 中止本指令时<br>・因发生异常，中止本指令时<br>・发生异常过程中，启动本指令时<br>・执行 MC_GroupStop 指令中，启动本指令时 | ・Execute 为 TRUE 时，与 Execute 的 FALSE 同时<br>・Execute 为 FALSE 时，1 个周期后 |
| Error | 本指令的启动条件或输入参数中含有异常因素时 | 异常已解除时 |

**圆弧插补的步骤**

1、登录进行插补的轴组

• 确定进行插补的轴组。

轴组即 _MC_GRP[***]。

• 通过该轴组变量的轴构成指定轴的构成。

• 通过该轴组变量的轴选择指定进行插补的轴的组合。

• 轴不使用轴号（轴 0～轴 63），而是使用逻辑轴（ 轴 A0～轴 A3）。

• 通过轴选择以前移方式向逻辑轴 A0～A3 分别指定轴号 0～63。

例：轴构成为 2 轴，将轴号 0～1 指定为轴 A0～A1 时，按以下方法指定。

| 逻辑轴 | 轴号 | 说 明 |
|---|---|---|
| 轴 A0 | 轴 0 | 从轴 A0 开始，按照轴号从小到大的顺序进行指定。 |
| 轴 A1 | 轴 1 | |

2、启用轴组

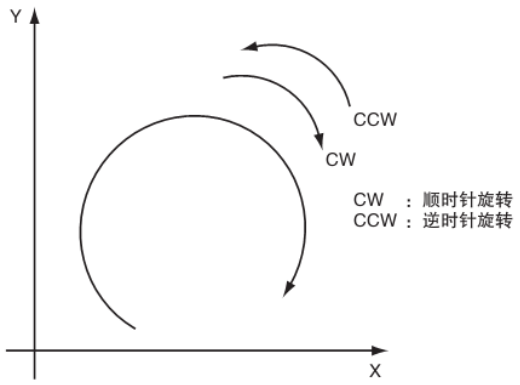
• 将轴组的各构成轴置为伺服 ON 状态，并使其处于原点确定状态。

• 执行 MC_GroupEnable( 启用轴组 ) 指令，使登录的轴组有效。

3、调用 MC_MoveCircular2D 指令，执行圆弧插补控制。

主要参数说明

① CircAxes(圆弧轴指定)，指定轴组中参与圆弧插补的轴




- 通过 CircAxes( 圆弧轴指定 ) 来指定向 X 轴、Y 轴分配的轴。
- 轴的指定不使用轴号 ( 轴 0～轴 63)，而是使用逻辑轴 (轴 A0～轴 A3)。

注意：

- 指定为 X 轴或 Y 轴的轴的计数模式请设定为 [ 线性模式 ]。
- 作为 [ 旋转模式 ] 启动时，将发生 “计数模式设定导致的圆弧插补指令启动异常( 错误代码：544A Hex)” 的错误。

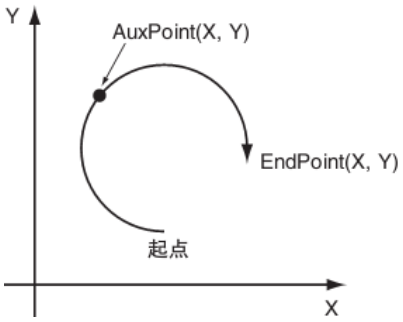
② CircMode(圆弧插补模式)

圆弧插补方式有 “通过**边缘点指定/圆心指定/半径指定**”三种，通过 CircMode(圆弧插补模式)指定。

以这些方法指定的位置有“绝对值定位”、“相对值定位”，通过 MoveMode(移动方法选择)指定。

**通过边缘点指定**

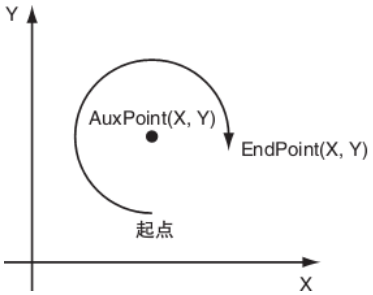
以当前位置为起点，执行连接通过点位置 AuxPoint(X,Y) 和终点 EndPoint(X, Y) 的圆弧插补。




- 当起点和边缘点为同一点或者终点和边缘点为同一点时，将进行直线插补；
- 起点、通过点位置、终点为同一点时，会发生异常；
- 起点和终点为同一点时，以起点和通过点为直径绘制正圆。这种情况下，通过 PathChoice( 路径选择 ) 指定圆弧的旋转方向。

**通过圆心指定**

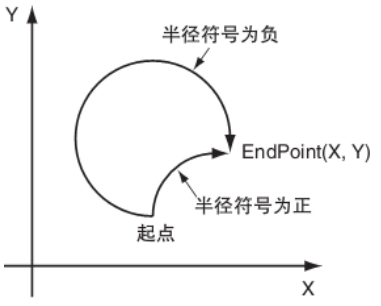
以当前位置为起点，执行以通过圆心位置 AuxPoint(X,Y) 指定的圆弧连接终点 EndPoint(X, Y) 的圆弧插补。
圆弧的旋转方向通过 PathChoice( 路径选择 ) 指定。




- 起点、终点为同一点时，绘制正圆。
- 从指定的中心位置到起点的半径与到终点的半径不同时，使用两个半径的平均值进行圆弧插补。这种情况下，按照与半径指定同样的方法计算中心位置，使用该半径和中心位置。

**通过半径指定**

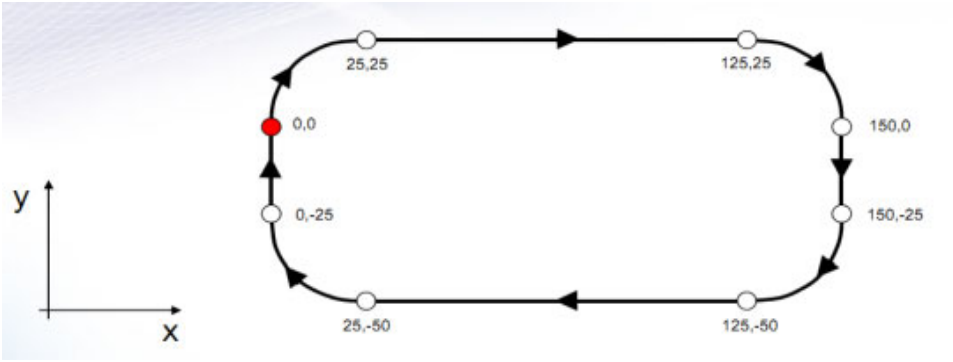
以当前位置为起点，执行以通过半径 AuxPoint(X, Y)指定的圆弧连接终点 EndPoint(X, Y) 的圆弧插补。
半径符号为负时，绘制出较长的圆弧；半径符号为正时，绘制出较短的圆弧。
圆弧的旋转方向通过 PathChoice( 路径选择 ) 指定。




- 半径通过 AuxPoint(X, Y) 的最初的元素来指定。不使用第二个元素。例如，半径为 100 时，AuxPoint(X, Y) 指定为 AuxPoint(100, 0)；
- 起点和终点为同一点时，会发生圆弧插补起点终点相同的异常，该组中所有轴停止动作；
- 指定的半径小于起点和终点间距离的 1/2 时，无法绘制圆弧，将发生异常。

④ BufferMode(缓存模式选择)，参考直线插补中描述。

三、使用直线插补和圆弧插补指令实现下图运动轨迹




以上图形由 4 段圆弧和 4 段直线构成，可使用 4 个圆弧插补和 4 个直线插补实现以上轨迹。程序中使用多重启动操作，实现不同插补操作间的连续性。

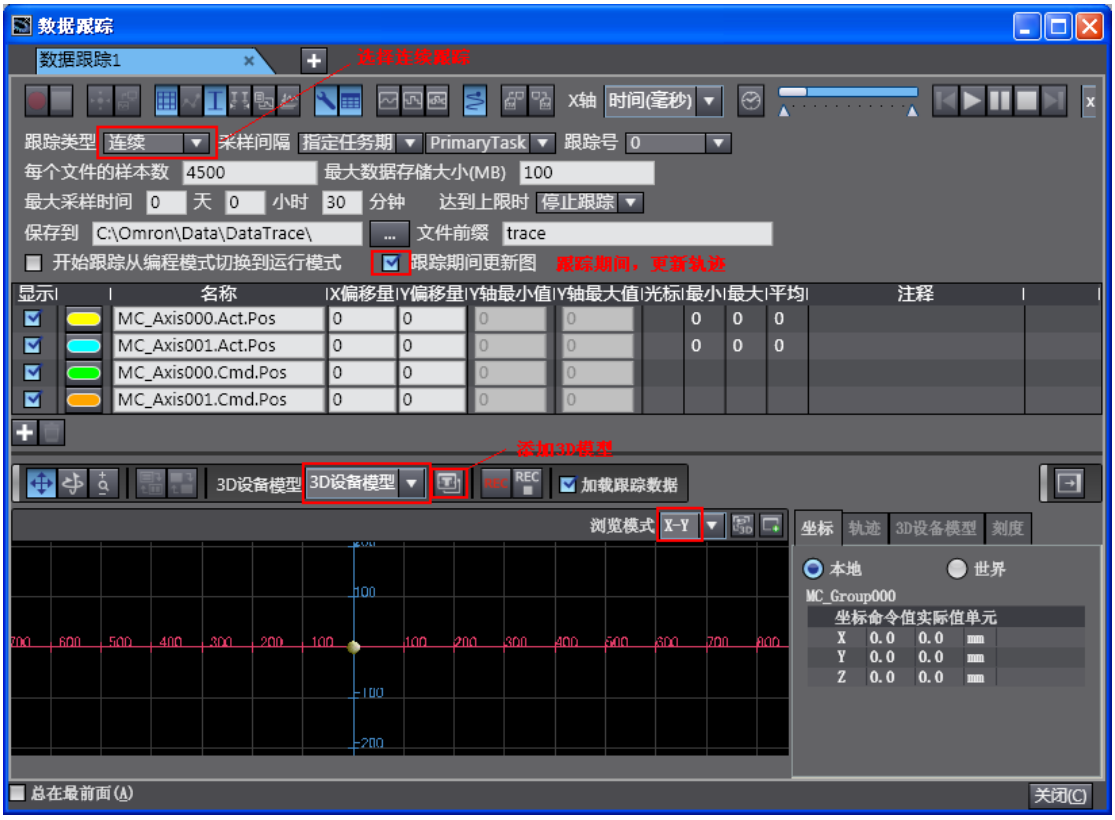
四、轨迹监视

通过 Sysmac Studio 数据跟踪功能，监视伺服动作过程。具体操作步骤如下：

1、添加新的数据跟踪

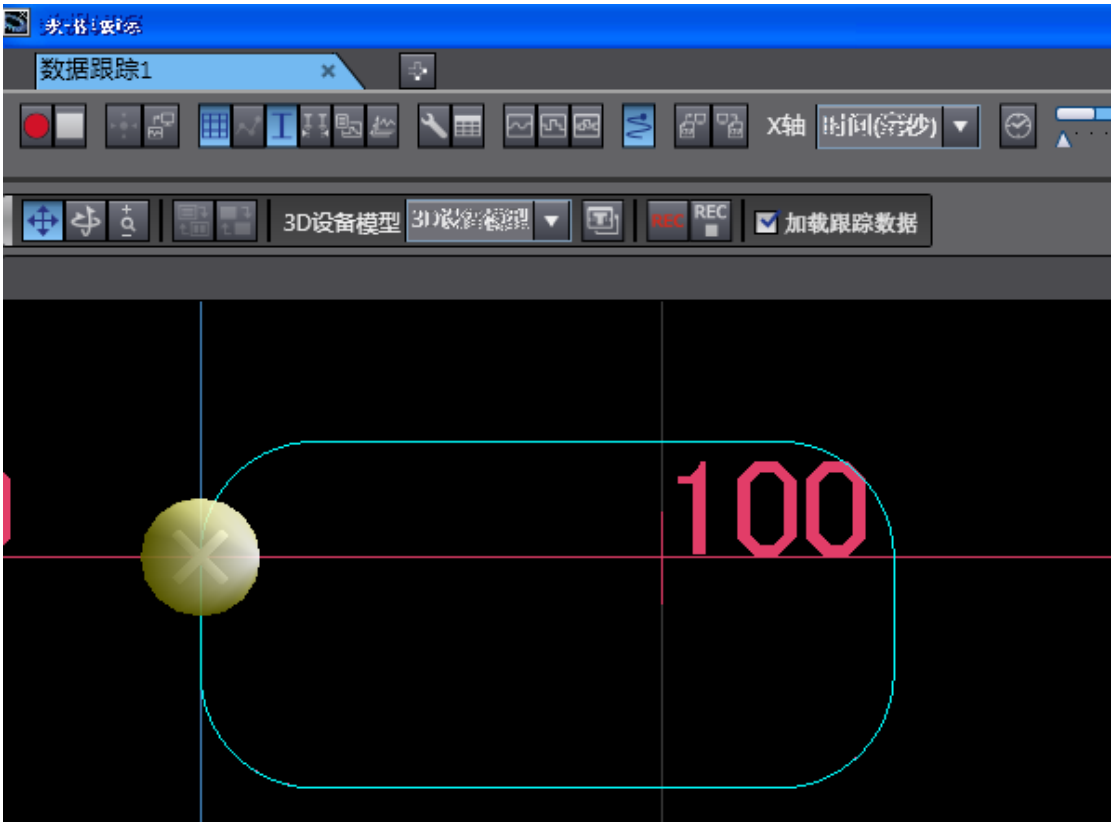
通过左侧目录树→“数据跟踪设置”，鼠标右键→“添加”→ “数据跟踪”；

2、对数据跟踪进行基本设置，如下图所示：




3、执行数据跟踪

执行数据跟踪获得如下轨迹图形。


数据跟踪1
X轴
3D设备模型
REC
加载跟踪数据
100