

1 概述

自动化工程中，同步运行功能承担着越来越重要的作用，随着开环及闭环控制系统的发展，机械解决方案越来越频繁地被不同的电气解决方案所替代。SIMOTION 工艺对象的同步运行功能提供了使用“控制工程”替代“刚性机械连接的选项”，可提供更加柔性，友好维护的解决方案。

SIMOTION 的同步运行功能由同步对象(synchronous object)提供。主值(Mater)产生的量(含位置,速度和加速度)经过同步对象的处理后赋值给从轴(Slave)从而实现同步运行。

同步运行关系至少包含一个主值和一个从轴，主值可以是一个位置轴或者外部编码器，或者由 Fixed gear ,Addition object Formula object(不能用于分布式同步)等 TO 提供。从轴包含一个跟踪轴 (slave)，一个或两个同步对象(synchronous objects)以及一个或多个 cam。

通过电子齿轮功能(GEAR)可完成主值与从轴间线性传递函数功能，与机械中的齿轮功能相同。指定的齿轮比用于主值与从轴间的线性位置关系。

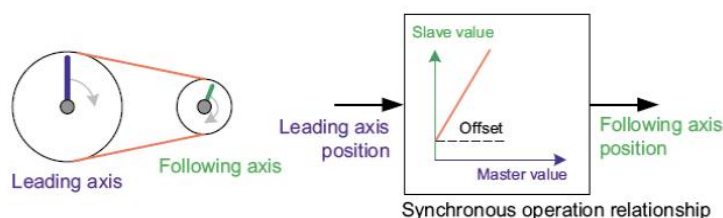


图1 GEAR 同步

通过速度同步(Velocity Gear)可实现主值与从轴间的线性速度关系。

通过电子凸轮功能(CAM)完成主值与从轴间的非线性传递函数功能。

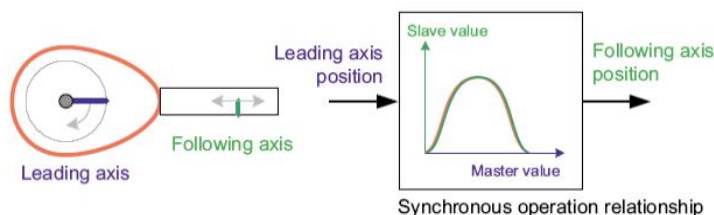


图2 CAMMING 同步

同步运行可分为三个过程

- 建立同步(Synchronization)
- 同步运行(Synchronized traversing)
- 解除同步(Desynchronization)

2 同步介绍

2.1 建立同步

建立同步的过程中主要受以下四个参数影响：

- 同步轮廓
- 同步模式
- 同步位置参考
- 同步方向

相关的参数如下表所示：

Parameters		Meaning	Value
同步轮廓	SyncProfile Reference	reference of the synchronization profile; 由同步长度或时间决定同步过程	RELATE_SYNC_PROFILE_TO_LEADING_VALUE 由位置长度决定同步长度 RELATE_SYNC_PROFILE_TO_TIME 由时间即动态响应决定同步长度
同步模式	Synchronizing Mode	synchronization criterion /synchronization position 同步模式	IMMEDIATELY 立即开始同步 IMMEDIATELY_AND_SLAVE_POSITION 立即开始同步从轴带偏差 ON_MASTER_POSITION , 同步点参考主轴位置 ON_MASTER_AND_SLAVE_POSITION 同步点参考主轴位置从轴带偏差 ON_SLAVE_POSITION 同步点参考从轴位置 AT_THE_END_OF_CAM_CYCLE 上一个 Cam 完整周期结束后开始同步
同步位置参考	SynPosition Reference	leading	BE_SYNCHRONOUS_AT_POSITION 由 synchronization criterion 决定同步终点, 同步开始点由同步长度或由动态响应参数决定
		trailing	SYNCHRONIZE_WHEN_POSITION_REACHED 由 synchronization criterion 决定同步起点 同步结束点由同步长度或动态响应参数决定(由 Profile 决定)
		symmetrical	SYNCHRONIZE_SYMMETRIC 开始点以及结束点由同步位置以及同步长度决定,两边对称, 此时同步轮廓参考不能选择由动态响应决定
同步方向	Synchronization direction	synchronization direction 同步方向	SYSTEM_DEFINED 最短路径, 但不改变方向 SAME_DIRECTION 保持方向 POSITIVE_DIRECTION 正向 NEGATIVE_DIRECTION 反向 SHORTEST_WAY 最短路径

表1 同步参数

2.1.1 同步轮廓参考 syncProfileReference

“同步轮廓参考”由参数 syncProfileReference 确定，该参数决定同步过程是由指定的位置长度还是由动态响应来决定同步过程。

- **RELATE_SYNC_PROFILE_TO_LEADING_VALUE** 由位置长度来决定同步过程

- **RELATE_SYNC_PROFILE_TO_TIME** 由动态响应来决定同步过程

由位置长度来决定同步过程就是用同步长度的方式来确定同步的整个过程，采用这种方式时，同步过程中的速度和加速度由系统计算得出，根据同步长度的不同，会得到不同的动态响应和位置曲线轮廓。由动态响应决定同步过程时，系统以指定的速度和加速度运行，此时位置曲线轮廓是基本固定的，但不同的动态响应数据将得到不同的同步长度。

下图为 syncProfileReference 选择 RELATE_SYNC_PROFILE_TO_LEADING_VALUE 时的实例曲线。同步位置参考主轴位置，同步长度为 500。

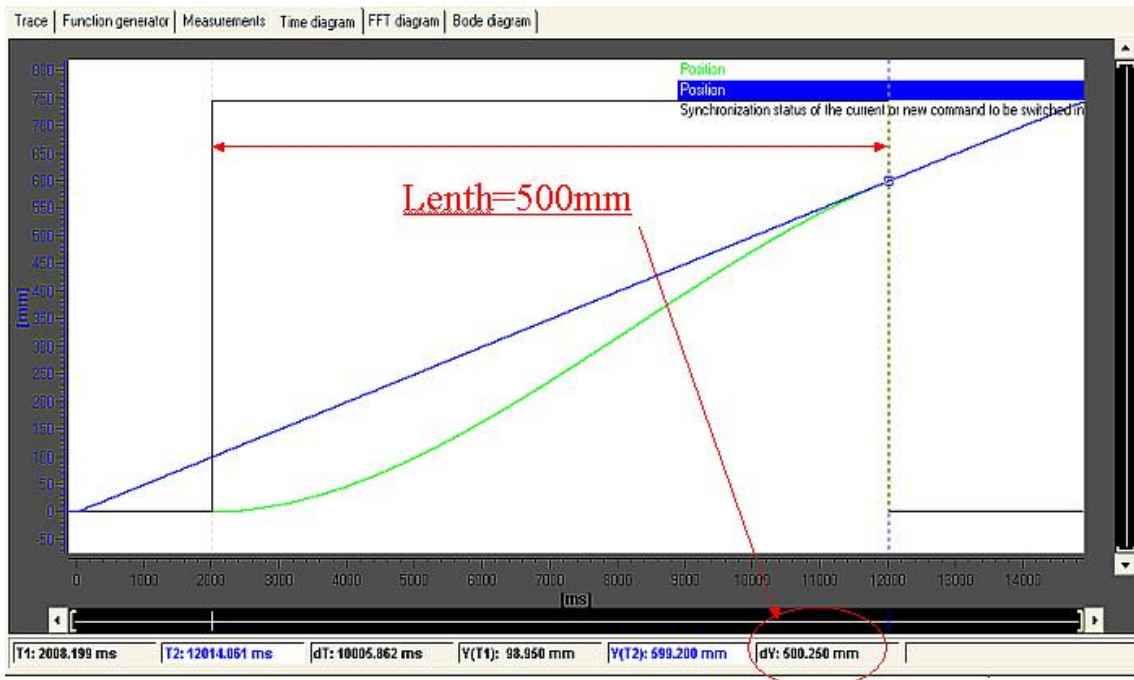


图3 由位置长度决定同步过程

图 4 为 syncProfileReference 选择 RELATE_SYNC_PROFILE_TO_TIME 时的实例曲线，此时系统以指定的加速度和速度进行同步。

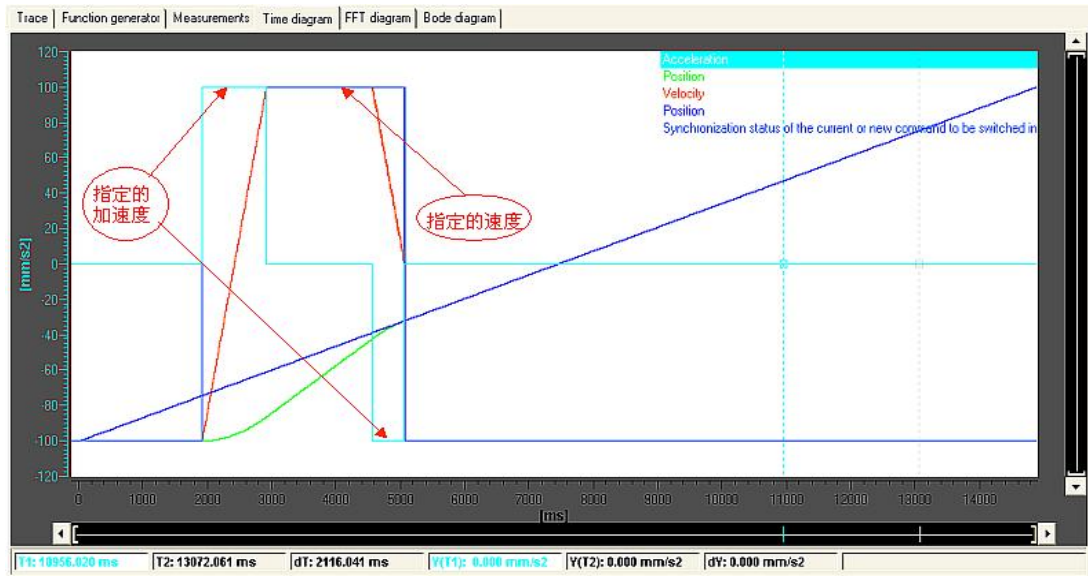


图4 由动态响应来决定同步过程

2.1.2 同步模式 synchronizingMode

“同步模式”决定了同步的开始点或者结束点，它用于指定一个参考位置，再由参数“同步位置参考（synPositionReference）”决定该参考位置是作为同步的终点、起点或者对称点。同步模式参数还决定了从轴与主轴在同步之后是否带偏差。

- **IMMEDIATELY** 立即开始同步
- **IMMEDIATELY_AND_SLAVE_POSITION** 立即开始同步从轴带偏差
- **ON_MASTER_POSITION** 同步位置参考主轴位置
- **ON_MASTER_AND_SLAVE_POSITION** 同步位置参考主轴位置从轴带偏差
- **ON_SLAVE_POSITION** 同步位置参考从轴位置
- **AT_THE_END_OF_CAM_CYCLE** 在前一个 CAM CYCLE 完成的之后开始同步

IMMEDIATELY 表示从同步指令执行时立即开始同步，此时同步位置参考参数不再起作用，即相当于 **synchronizingMode** 决定了同步的起点，同步的终点可通过位置长度或动态响应确定，如图 5 所示，开始同步的信号一发出，系统立即开始同步。

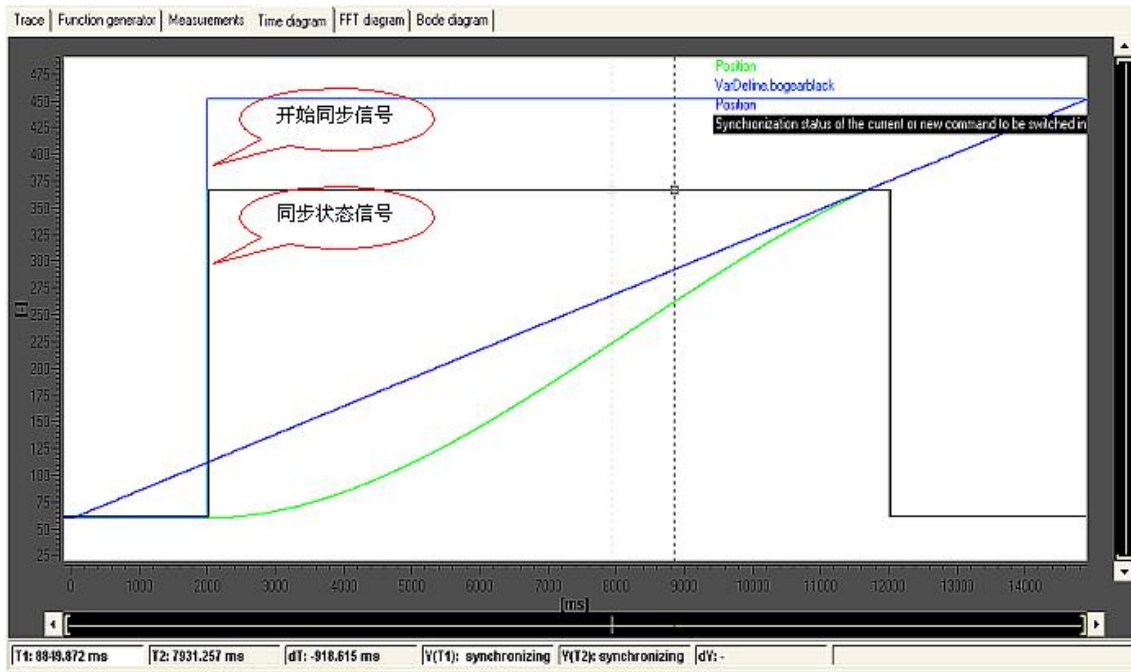


图5 SynchronizingMode=IMMEDIATELY

IMMEDIATELY_AND_SLAVE_POSITION 和 **IMMEDIATELY** 相同，但可以指定同步后从轴与主值之间的位置偏差，如下图，从轴和主值之间的偏差为 30mm

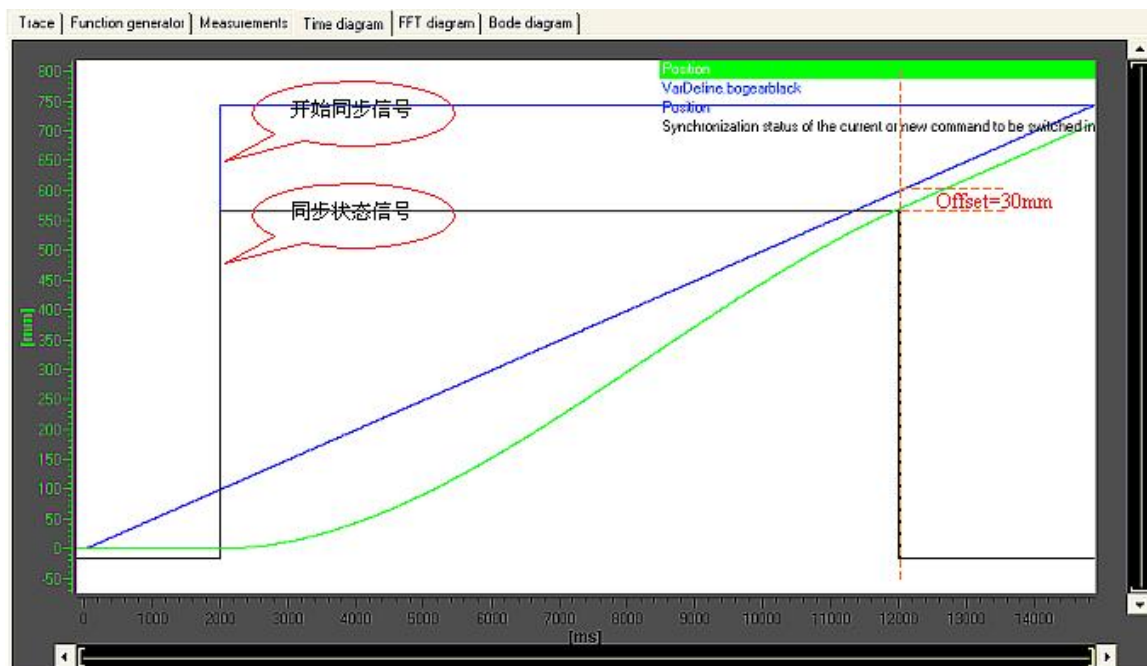


图6 SynchronizingMode= IMMEDIATELY_AND_SLAVE_POSITION

ON_MASTER_POSITION 表示同步的起点，对称点或终点是参考主轴位置确定的，下图的例子是在主轴位置为 200mm 时开始同步。

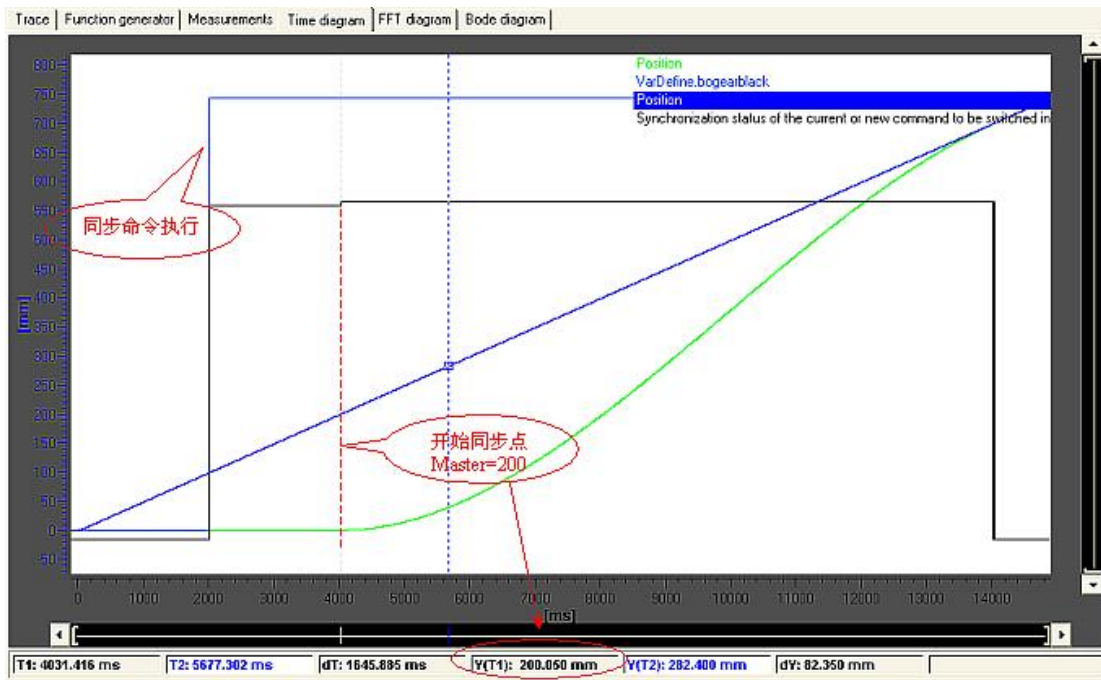


图7 SynchronizingMode= ON_MASTER_POSITION

ON_SLAVE_POSITION 表示同步的起点，对称点或终点是参考从轴位置确定的，此时从轴与主轴的偏差不可指定，下图的例子为从轴位置为 100mm 时开始同步。

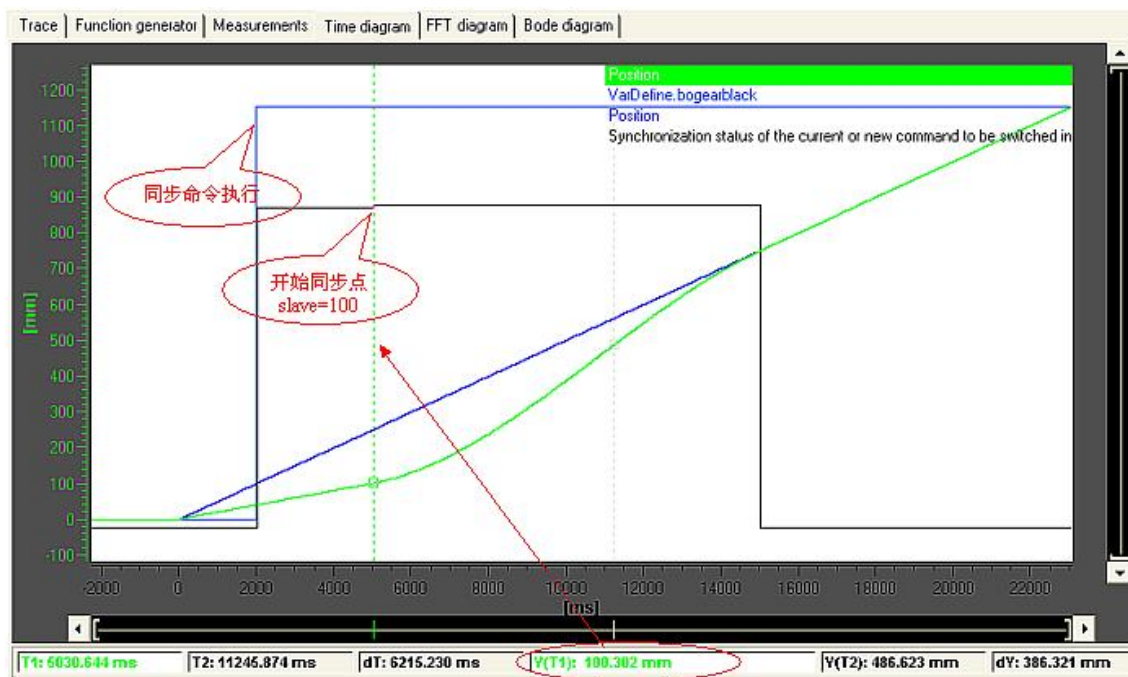


图8 SynchronizingMode=ON_SLAVE_POSITION

ON_MASTER_AND_SLAVE_POSITION 与 **ON_MASTER_POSITION** 相同，但可以指定同步后从轴与主值之间的位置偏差，下图的例子中为从主轴位置为 200mm 时开始同步，并且从轴的偏差为 30mm。

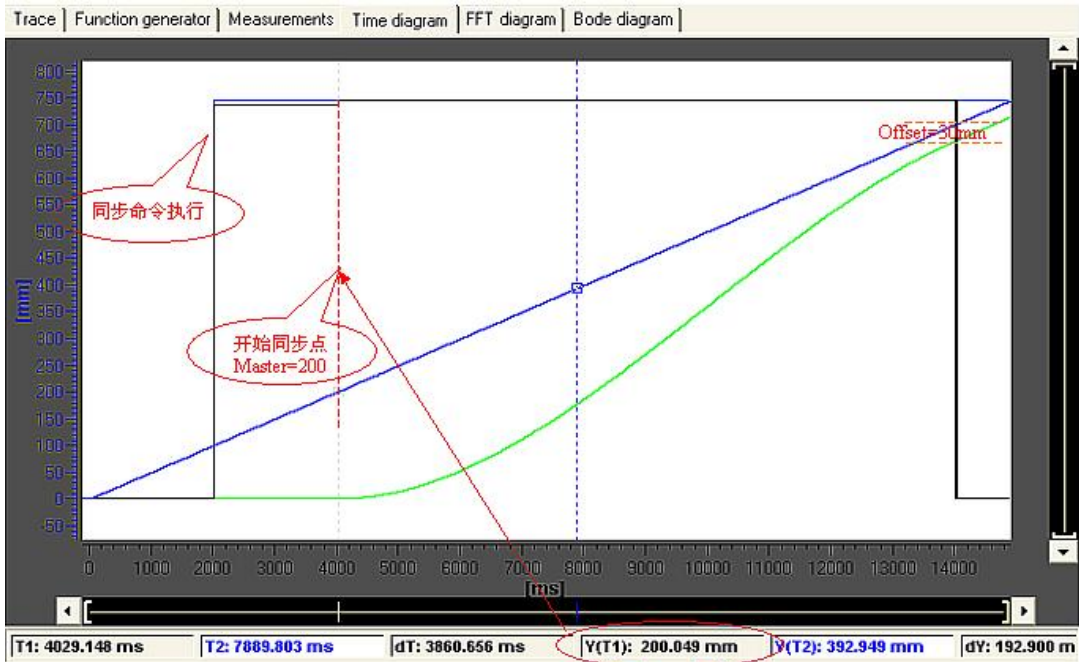


图9 SynchronizingMode= ON_MASTER_AND_SLAVE_POSITION

AT_THE_END_OF_CAM_CYCLE 是在 CYCLIC CAMMING 同步时两个 CAM 曲线切换时使用，即在上次运行着的周期性 cam 完成一个循环后运行本指令中指定的 cam，如下图：

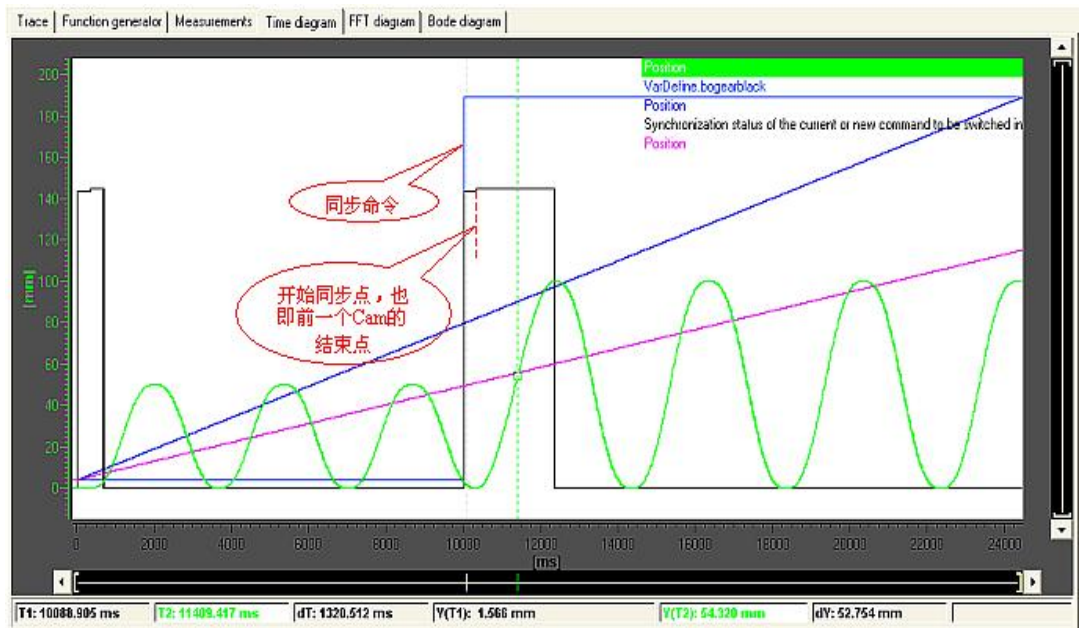


图10 SynchronizingMode= AT_THE_END_OF_CAM_CYCLE

2.1.3 同步位置参考 synPositionReference

“同步位置参考”决定了由“同步模式 synchronizingMode”指定的位置是作为同步过程的起点，终点还是对称点。

- **LEADING/ BE_SYNCHRONOUS_AT_POSITION**

在 synchronizingMode 指定的位置之前完成同步，即 synchronizingMode 指定的位置是终点。同步开始点由同步长度或由动态响应参数决定。

- **TRAILING/SYNCHRONIZE_WHEN_POSITION_REACHED**

在 synchronizingMode 指定的位置开始进行同步，即 synchronizingMode 指定的位置是起点。同步终点由同步长度或由动态响应参数决定。

- **SYNCHRONIZE_SYMMETRIC**

同步开始点以及结束点由 synchronizingMode 指定的位置以及同步长度决定，两边距离相同，即 synchronizingMode 指定的是对称点。此时“同步轮廓参考 synchronization reference”不能选择“RELATE_SYNC_PROFILE_TO_TIME”，否则会出错：30001: illegal parameter”。

下图的例子中为同步位置参考分别采用起点，终点还是对称点所做的曲线记录，其中黑线为 LEADING，绿线为 SYMMETRICAL，红线为 TRAILING，同步位置为 500mm。



图11 同步位置参考 synPositionReference

2.1.4 同步方向 synchronizingDirection

同步方向决定同步过程中从轴的运动方向：

- **SYSTEM_DEFINED:**
 - 按最短路径进行同步，
 - 从轴在同步运行过程中不反向
- **SAME_DIRECTION:** 保持从轴的方向，如果从轴静止则为正向同步
- **POSITIVE_DIRECTION:** 从轴的运动方向始终保持正向
- **NEGATIVE_DIRECTION:** 从轴的运动方向始终保持负向
- **SHORTEST_WAY:**
 - 按最短路径进行同步
 - 从轴在同步运行过程中会产生反向

注：选择同步方向时应确保该方向的运动能够实现同步，否则系统将会出错，如果不能确定方向可选择 SYSTEM_DEFINED 或 SHORTEST_WAY（方向自动判定）以确保同步完成。

下图的例子为同步方向分别为正向和反向时的同步曲线，当同步方向为 POSITIVE 时采用的是正向 GEARING，当同步方向为 NEGATIVE 时采用的是反向 GEAR。

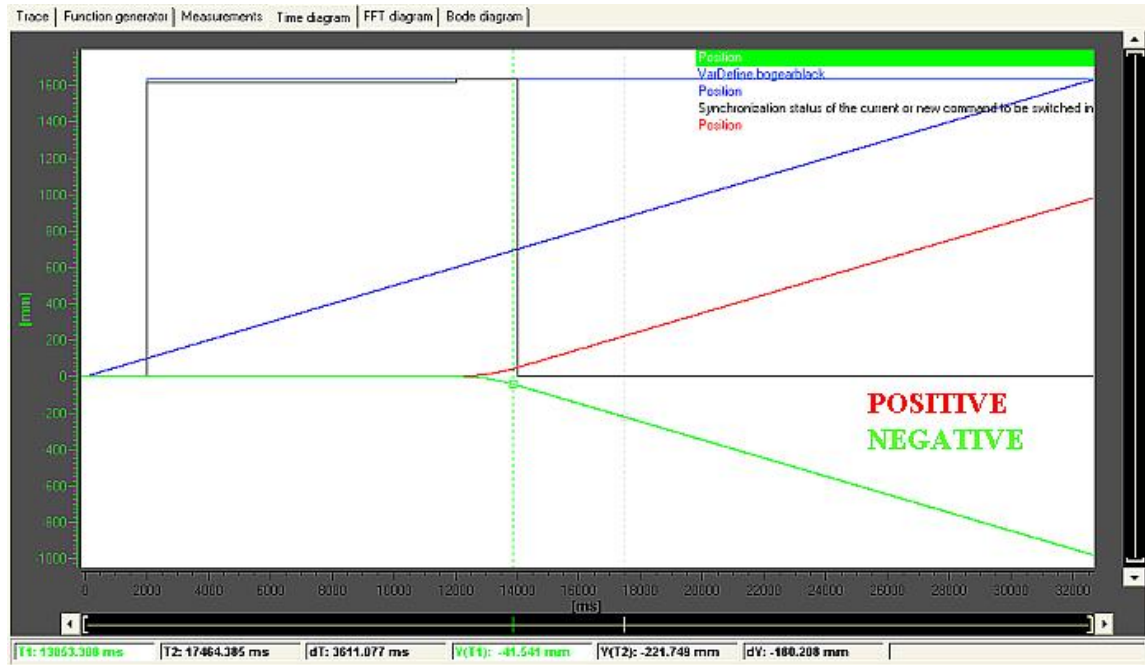


图12 synchronizingDirection= POSITIVE_DIRECTION/ NEGATIVE_DIRECTION
synchronizingDirection=SAME_DIRECTION 时，类似 POSITIVE_DIRECTION 或者
NEGATIVE_DIRECTION，不再介绍示例曲线。

图 13 中的例子为同步方向采用 SHORTEST_WAY 时，分别在不同的主轴位置启动同步，根据最短路径的方法来确定从轴的同步运动方向，图中可以看出绿线时从轴为正向同步，红线为先反向后正向，例中主轴和从轴均采用模态轴。

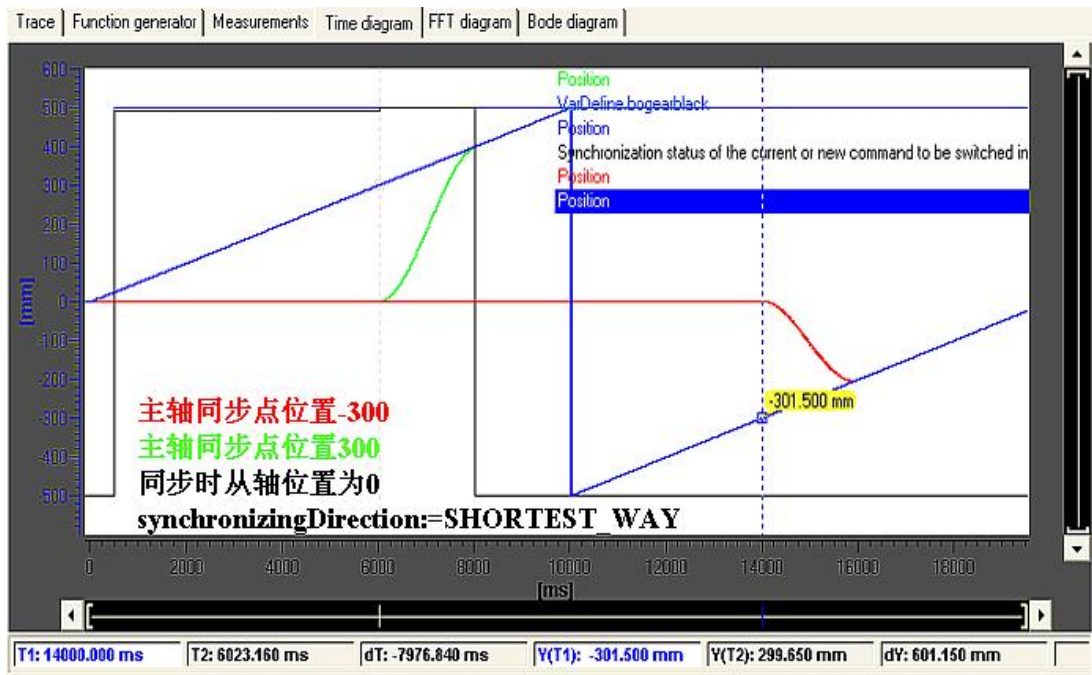


图13 synchronizingDirection= SHORTEST_WAY

下图中例子为同步方向分别采用 SHORTEST_WAY 和 SYSTEM_DEFINED 时的从轴同步曲线，从图中可以看出，SYSTEM_DEFINED 和 SHORTEST_WAY 的区别在于从轴是否会在同步过程中反向，图中绿色为 SHORTEST_WAY，可以看出绿线为先反转再正转，而红线则一直是正转，例中主轴和从轴均为模态轴。

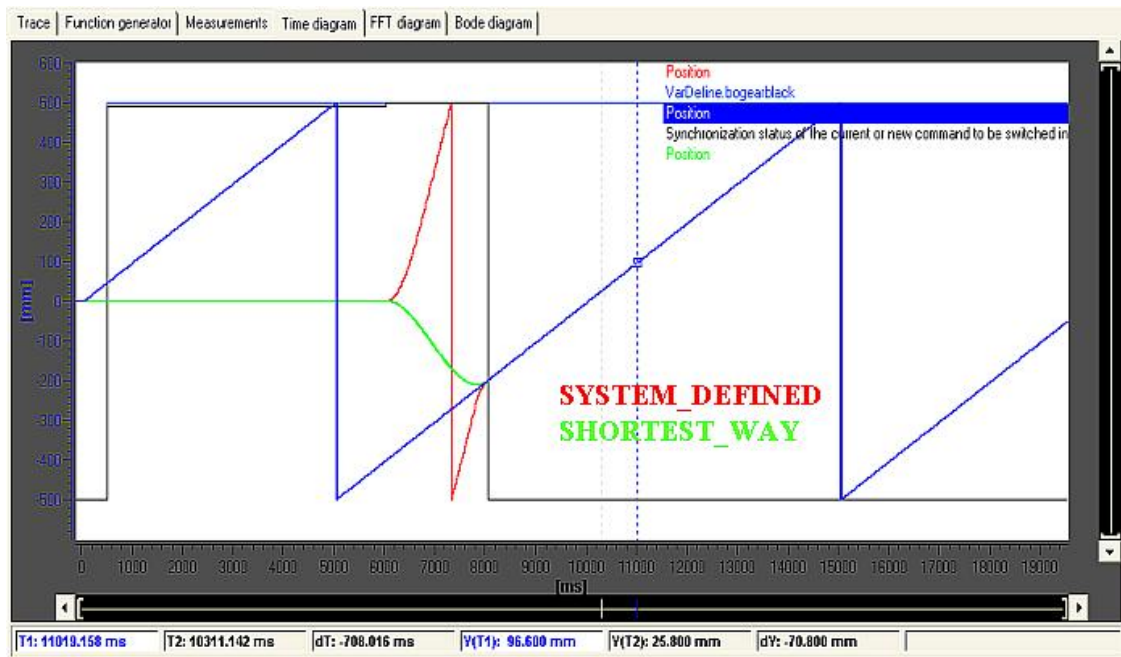


图14 synchronizing Direction=SYSTEM_DEFINED/SHORTEST_WAY

2.1.5 同步状态监控

在建立同步期间，可通过对以下参数的监控来确定当前的同步状态：

同步对象的同步状态	
state	CAMMING: 当前同步为 CAMMING 同步 GEARING: 当前同步为 GEARING 同步 VELOCITY_GEARING: 当前同步为速度同步 INACTIVE :没有同步操作
syncState	Yes: 已达到同步状态 No: 没有达到同步状态
synchronizingState	WAITING_FOR_SYNC_POSITION: 等待主值达到同步位置 WAITING_FOR_CHANGE_OF_MASTER_DIRECTION: 等待主值反向 SYNCHRONIZING_NOT_POSSIBLE:同步无法完成 SYNCHRONIZING: 正在同步 INACTIVE: 同步未激活 WAITING_FOR_MERGE: 同步命令已执行但还未激活
从轴状态	
syncMonitoring.syncState	Yes: 已达到同步状态 NO: 没有达到同步状态
syncMonitoring.followingMotionState	INACTIVE: 同步未激活 BASIC_MOTION_ACTIVE:基本同步 SUPERIMPOSED_MOTION_ACTIVE: 叠加同步 BASIC_AND_SUPERIMPOSED_MOTION: 基本同步和叠加同步

表2 同步状态监控参数

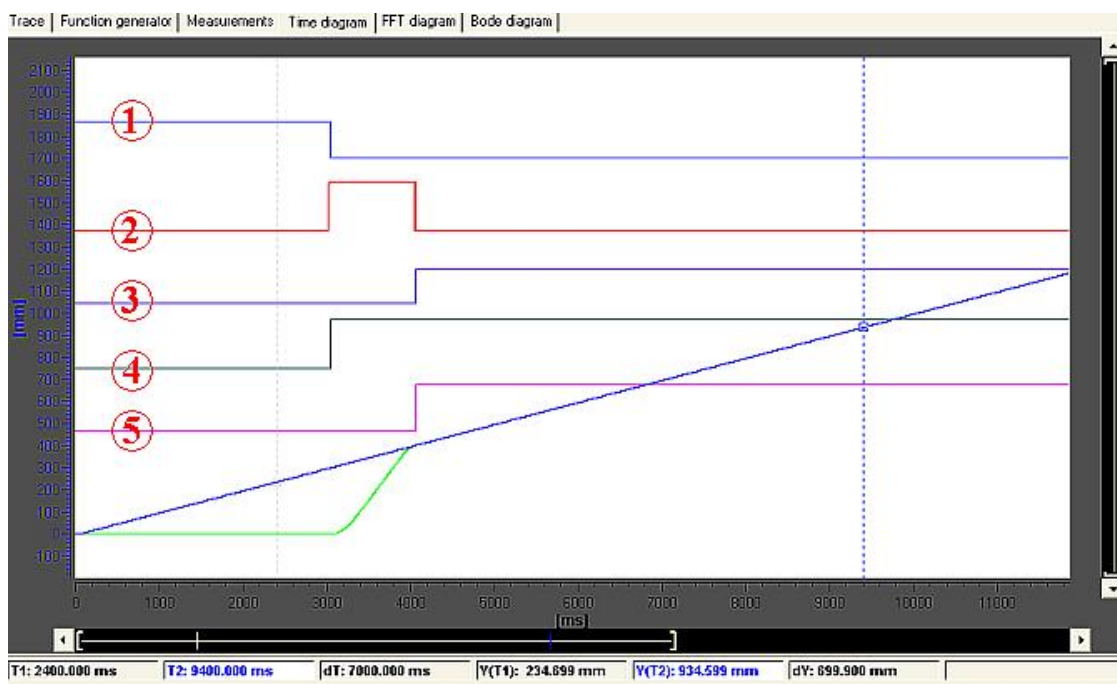


图15 同步状态监控

注：图 15 中 1-5 曲线分别为

- No.1: 同步对象.state
- No.2: 同步对象.synchronizingstate
- No.3: 同步对象.syncstate
- No.4: 从轴.syncmonitoring.followingstate

No.5: 从轴.syncstate

2.2 解除同步

解除同步过程的相关参数和建立同步类似，包含以下的参数

Parameters		Meaning	Value
同步轮廓参考	syncProfileReference	Desynchronization profile 由位置或时间决定解除同步长度	RELATE_SYNC_PROFILE_TO_LEADING_VALUE 由位置长度决定解除同步长度
			RELATE_SYNC_PROFILE_TO_TIME 由时间即动态响应决定解除同步长度
解除同步模式	syncOffMode	desynchronization criterion/desynchronization 解除同步开始的位置	IMMEDIATELY 立即开始解除同步 ON_MASTER_POSITION 解除同步点参考主轴位置 ON_SLAVE_POSITION 解除同步点参考从轴位置 AT_THE_END_OF_CAM_CYCLE 上一个 Cam 完整周期结束后开始同步
解除同步位置 作为起点或终点或中点	syncOffPositionReference:	类似 leading	AXIS_STOPPED_AT_POSITION 由 desynchronization criterion 决定解除同步终点, 解除同步开始点由解除同步长度或由动态响应参数决定 (由 Profile 决定)
		类似 trailing	BEGIN_TO_STOP_WHEN_POSITION_REACHED 由 synchronization criterion 决定解除同步起点 解除同步结束点由解除同步长度或动态响应参数决定(由 Profile 决定)
		类似 symmetrical	STOP_SYMMETRIC_WITH_POSITION 开始点以及结束点由解除同步位置以及解除同步长度决定, 两边对称,此时 Profile 不能选择由动态决定

表3 解除同步参数

2.2.1 解除同步轮廓参考 syncProfileReference

该参数与建立同步过程的 syncProfileReference 参数相同，有以下两种值，代表解除同步过程是由位置长度还是由动态响应决定。

- **syncProfileReference:=RELATE_SYNC_PROFILE_TO_LEADING_VALUE**

通过指定的主值距离解除同步，当主值运行通过解除同步长度时，从值运行至零速，解除同步长度在“syncOffLength”参数中设置

- **syncProfileReference:=RELATE_SYNC_PROFILE_TO_TIME**

通过指定的动态响应参数解除同步，遵照在解除同步命令中指定的动态响应值，从值运行至零速。

2.2.2 解除同步模式 syncOffMode

解除同步模式参数 syncOffMode 参数和同步模式 synchronizingMode 的功能相同，用于指定一个参考位置，再由参数“解除同步位置参考（syncOffPositionReference）”决定该参考位置是作为终点还是起点或者对称点。

- **IMMEDIATELY** 立即开始解除同步
- **ON_MASTER_POSITION** 解除同步点参考主轴位置

- **ON_SLAVE_POSITION** 解除同步点参考从轴位置
- **AT_THE_END_OF_CAM_CYCLE** 上一个 Cam 完整周期结束后开始同步

2.2.3 解除同步位置参考 syncOffPositionreference

- **syncOffPositionReference:=AXIS_STOPPED_AT_POSITION**

在指定位置完成解除同步

- **syncOffPositionreference:=BEGIN_TO_STOP_WHEN_POSITION_REACHED**

从指定的位置开始解除同步

- **syncOffPositionreference:=STOP_SYMMETRIC_WITH_POSITION**

从指定的位置对称解除同步（不能用于基于动态响应参数的解除同步轮廓）

下图为同步解除的实例，例中同步过程由位置长度决定，即 syncProfileReference 等于 RELATE_SYNC_PROFILE_TO_LEADING_VALUE，解除同步的位置由主轴位置决定，即 syncOffMode 为 ON_MASTER_POSITION。

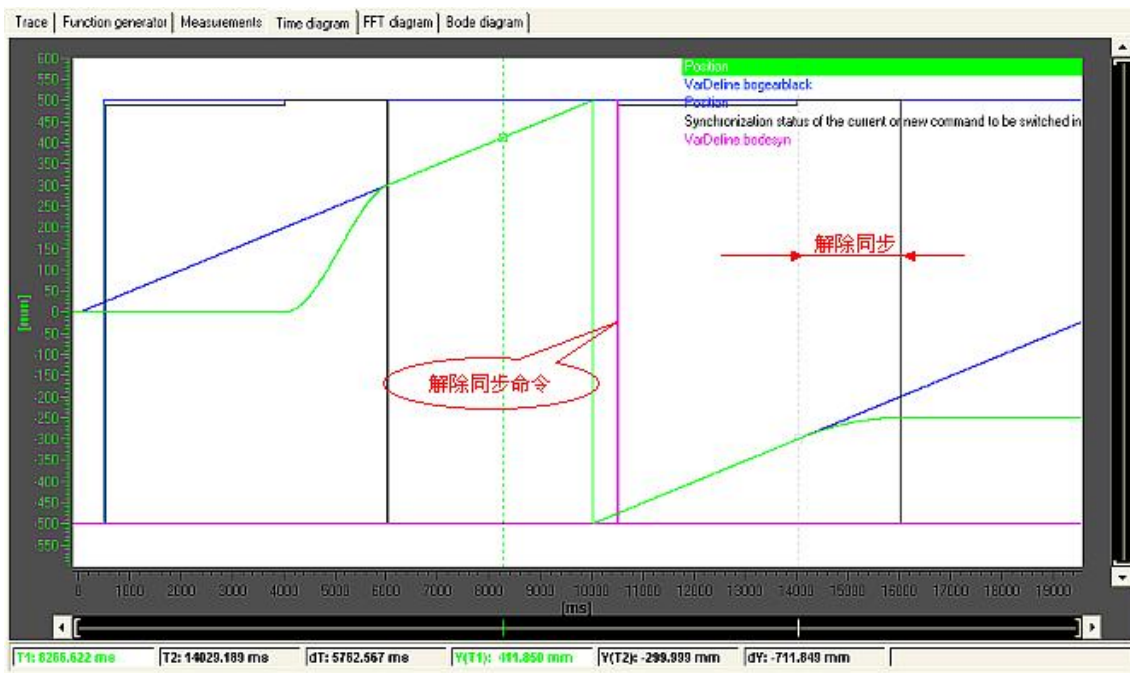


图16 解除同步过程

3 其它同步相关内容

3.1 Set_Master

从轴可通过同步对象与多个主值相关联。但在特定的时间只能有一个主值被激活，并可通过使用 “_setMaster()”命令切换主值。主值的切换并不是一次新的同步过程，同步状态参数始终保持 YES 状态。主值的切换独立于同步过程和解除同步过程，下图为主值切换实例：

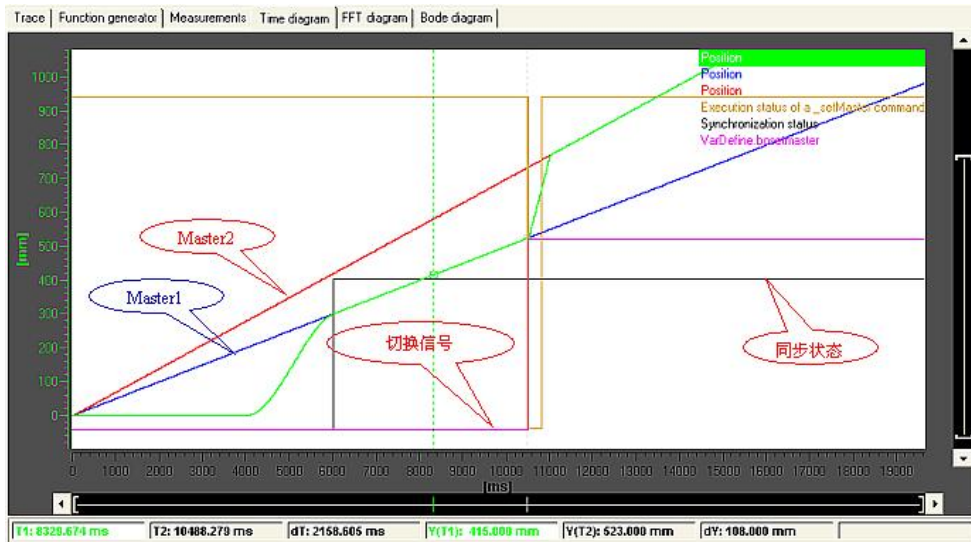


图17 主值切换

_setMaster()的参数transientBehavior决定了切换过程的运动。

- **DIRECT:** 无动态响应参数，根据从轴的动态响应限制值进行运动
- **WITH_DYNAMICS:** 带动态响应，根据动态响应参数进行切换过程的运动调整
- **WITH_NEXT_SYNCHRONIZING:** 下一次同步命令时主值切换再生效

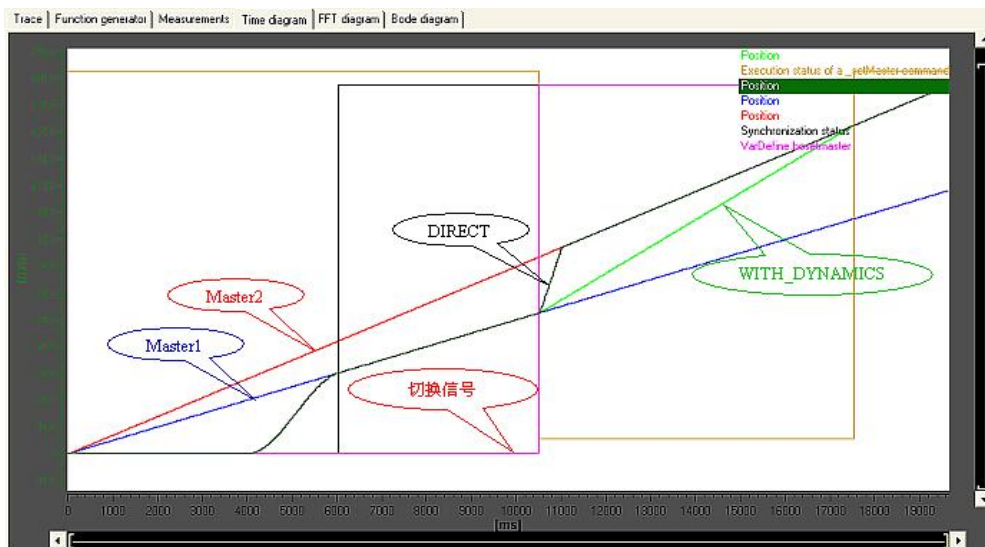


图18 主值切换中的 transientBehavior 参数

3.2 叠加同步

最多两个同步对象可以被连接至一个从轴，分别被称为基本同步对象和叠加同步对象。

添加叠加同步对象方法为 Expert > Insert Superimposed synchronous object，如下图：

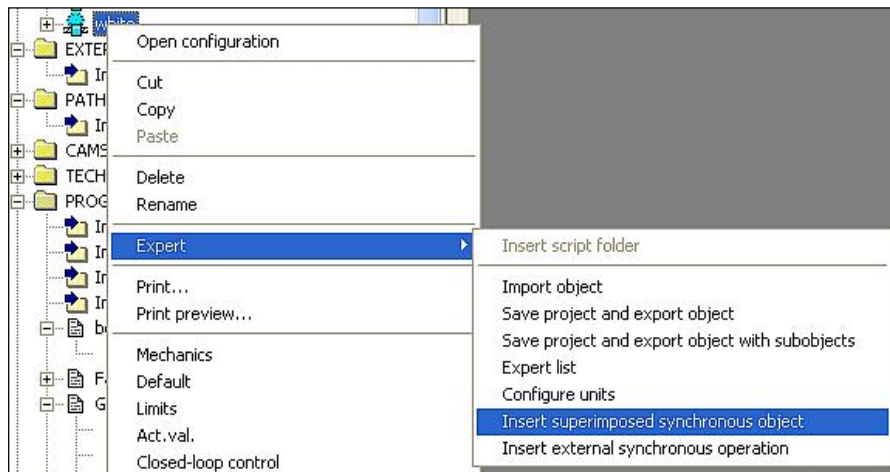


图19 添加叠加同步对象

叠加同步操作和基本同步操作相同，但从轴的参考坐标为叠加坐标系统，即叠加同步的从轴位置有自己独立的坐标系统，并根据自己的坐标进行操作。举个例子，如果叠加同步对象和基本同步对象同时和同一个主轴进行相同的 1:1 的绝对位置同步操作，那么最终的从轴位置将会是主轴的两倍，如下图：

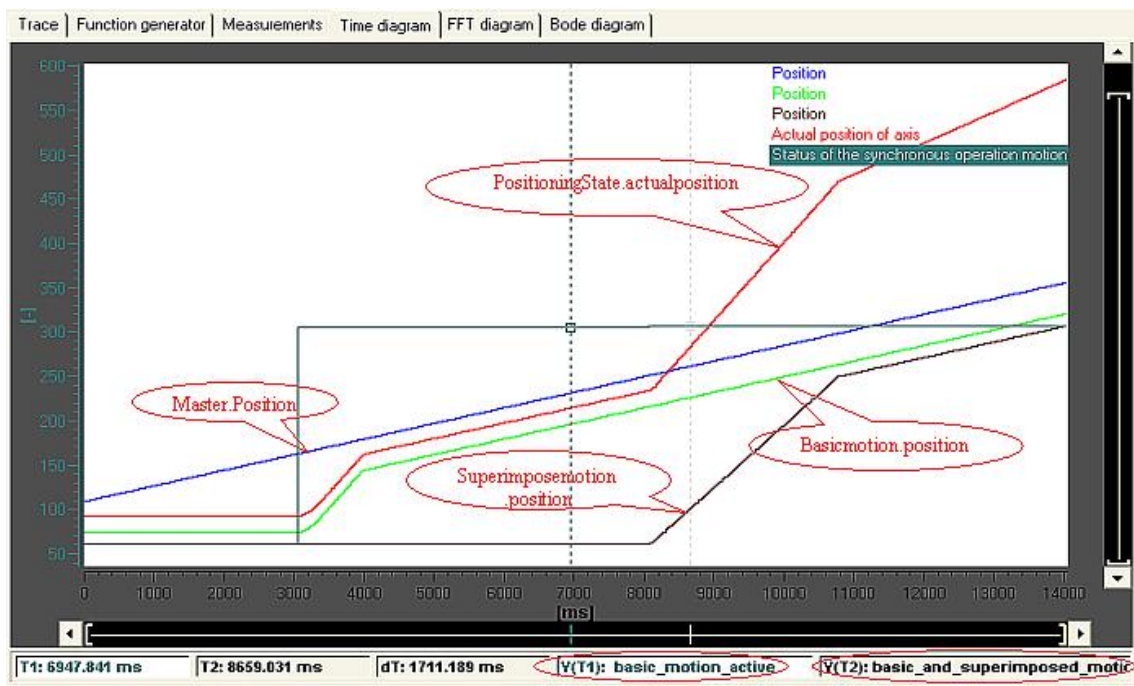


图20 叠加同步

注：图中的基本同步和叠加同步均采用绝对 Gearing 同步。

当叠加同步激活时，从轴的同步状态变量 `syncMonitoring.syncState` 是变化的，即当基本同步完成为 `yes`，当叠加同步激活时又变成 `NO`，当叠加同步也完成后又变成 `yes`，如下图：



图21 从轴同步状态

叠加同步和基本同步的操作还可通过以下变量进行监控：

基本同步数据	Slave.basicmotion.position
	Slave.basicmotion.velocity
	Slave.basicmotion.acceleration
叠加同步数据	Slave.superimposedmotion.position
	Slave.superimposedmotion.velocity
	Slave.superimposedmotion.acceleration
总的的数据	Slave.positioningstate.actualposition
	Slave.motionstatedata.actualvelocity
	Slave.motionstatedata.actualacceleration
同步状态	Slave.syncmonitoring.followingmotionstate

表4 叠加同步从轴系统变量监控

其中 followingMotionState 变量有如下状态：

- **INACTIVE:** 同步运动没有激活
- **BASIC_MOTION_ACTIVE:** 基本同步激活
- **SUPERIMPOSED_MOTION_ACTIVE:** 叠加同步激活
- **BASIC_AND_SUPERIMPOSED_MOTION_ACTIVE:** 基本同步和叠加同步均已激活

3.3 耦合规则

同步运行功能中主值和从轴间耦合有以下的规则:

- 同步对象和从轴必须在同一控制器下
- 支持主对象与跟踪轴采用不同的 IPO 循环时钟
- 主值和从轴可在不同控制器下(分布式同步)
- 同步对象及从轴在配置时被永久地分配
- 最多两个同步对象可以被连接至同一个从轴
- 主值对象可以被互连到多个同步对象
- 从轴可通过同步对象与多个主值相关联。但在特定的时间只能有一个主值被激活, 并可通过使用“_setMaster()”命令切换主值.
- 一个 Cam 曲线可连至多个同步对象
- 一个同步对象可连至多个 Cam, 并通过使用“_enableCamming()”命令进行切换
- 当轴做为主值时, 可使用设定值耦合(Setpoint coupling)或通过外推的实际值耦合(Actual value with extrapolation coupling).
- 当使用外部编码器作为主值时可使用实际值耦合(Actual value coupling)或外推实际值(Actual value with extrapolation coupling).
- 一个轴不能既做为另一个轴的从轴同时又做为这个轴的主值

3.4 仿真模式

同步运行可以切换至仿真模式, 在仿真模式下, 同步对象仍然会计算相应的从轴设定值, 但不是输出到从轴, 仿真模式可以在任何时候启用和恢复:

- **_enableFollowingObjectSimulation():** 激活仿真模式
- **_disableFollowingObjectSimulation():** 退出仿真模式

如果在退出仿真模式时, 当前的设定值与同步对象给定的设定值有偏差, 那么从轴有可能输出最大动态响应来达到所需的设定值。

参数 **disableSynchronousOperation** 将决定在仿真模式激活期间, 如果从轴的使能信号丢失, 是否取消同步对象的同步运行模式。

3.5 同步运行监控

在同步状态达到后，即保持同步运行过程中(syncState = YES)，同步运行监控激活：

- 设定值故障 Setpoint error
- 实际值故障 Actual value error

设定值故障是指同步对象计算出的从轴设定值与考虑了动态响应限制值之后所能得到设定值之间的偏差超过了一定范围之后产生的故障。从轴的系统变量 syncMonitoring.Difference CommandValue 显示该偏差的实际值。

动态响应限制值可在相应轴的 Limits 中进行设定：

There are two different limits:
- Limits which must never be exceeded because of mechanical reasons.
- Limits which can be modified by programs.

Max. velocity: mm/s

Pos. prog. velocity: mm/s

图22 轴的速度限制值

Position and velocity | Dynamic response | Fixed endstop

Hardware limits (limits that must not be exceeded for mechanical reasons):

Acceleration: mm/s² Jerk: mm/s³

图23 轴的加速度限制值

实际值故障是指同步对象计算出的从轴设定值与实际值之间的偏差超过设定范围后产生的故障。从轴的系统变量 syncMonitoring.differenceActualValue 显示该偏差的实际值。监控值的产生如下图 24 所示。

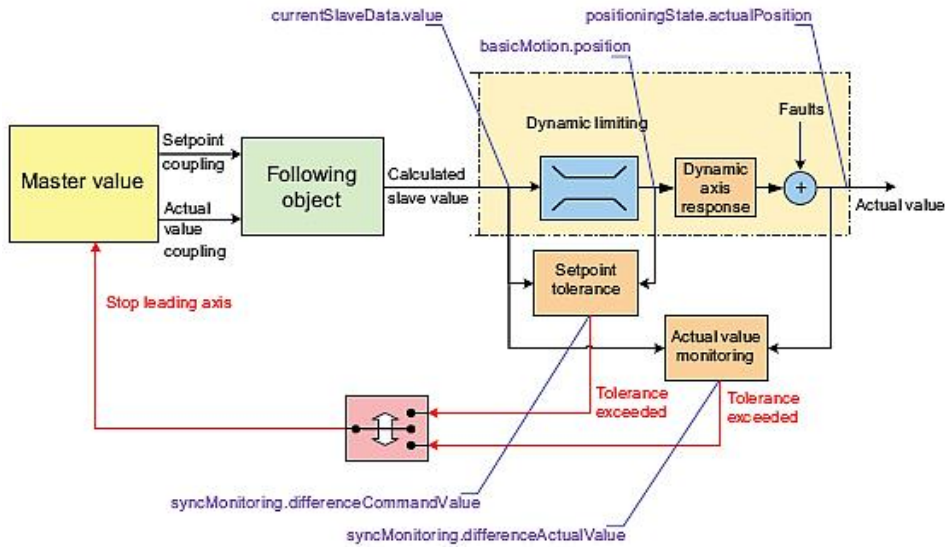


图24 同步运行监控

超出监控值时，从轴会产生"40201 Synchronous operation tolerance exceeded on the following axis"的故障，该故障也可通过设置参数 `TypeOfAxis.GearingPos Tolerance.enableErrorReporting` 传送到主值中，主值输出故障"40110 Error triggered on slave during synchronous operation (error number)"

监控的偏差设定值，可以通过从轴的Monitoring中的以下画面进行设置

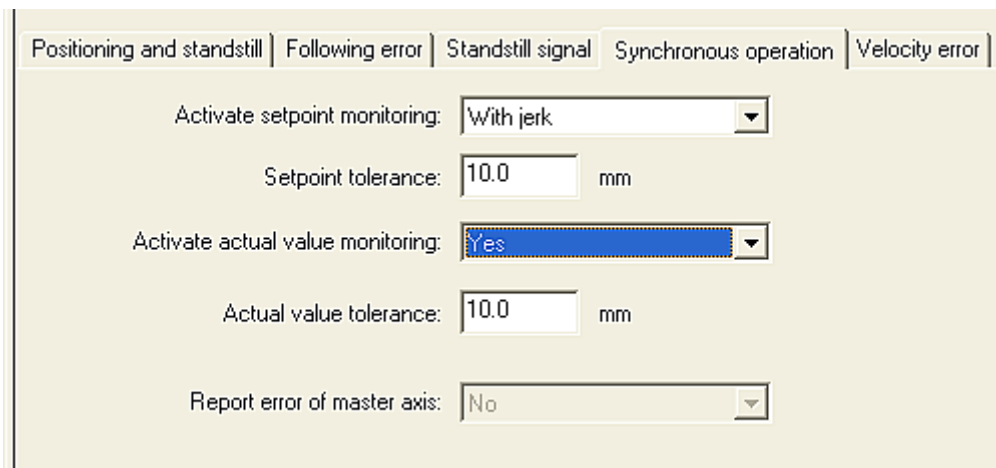


图25 同步运行监控参数设置

4 Gearing

4.1 Gearing 基本介绍

如前文所述，通过电子齿轮功能(GEARING)的功能是完成主值与从轴间线性传递函数功能，与机械中使用的齿轮功能相同，其主值和从轴按以下的公式进行计算：

$$\text{Slave value} = \text{Gear ratio} \times \text{Master value} + \text{Offset}$$

其相关的重要因素如下：

1) 齿轮比

齿轮比用于指定主值与从轴间的线性位置关系，齿轮比可以用分数（分子加分母）形式或浮点数形式进行指定。

Type of gear ratio Floating-point number	Type of gear ratio Fraction (nominator/denominator)
Gear ratio type Value entry	Gear ratio type Value entry
Gear ratio 1	Gear ratio numerator 1
	Gear ratio denominator 1
浮点数	分数形式

图26 齿轮比

2) 偏差

从轴和主值还可通过 Offset 保持相应的偏差。Offset 的实现是通过同步命令 `_enableGearing()` 中的同步模式参数 `synchronizingMode` 选择带 Offset 的值来实现的：

- **IMMEDIATELY_AND_SLAVE_POSITION** 立即开始同步从轴带偏差
- **ON_MASTER_AND_SLAVE_POSITION** 同步位置参考主轴位置从轴带偏差

3) Gearing Type

Gearing 类型分为绝对 Gearing 和相对 Gearing。绝对 Gearing 即主值和从轴的位置关系是绝对位置关系，即双方的位置关系是以零点作为参考的。相对 Gearing 则是主值和从轴都以当前位置值作为参考点，之后保持位置的线性关系。

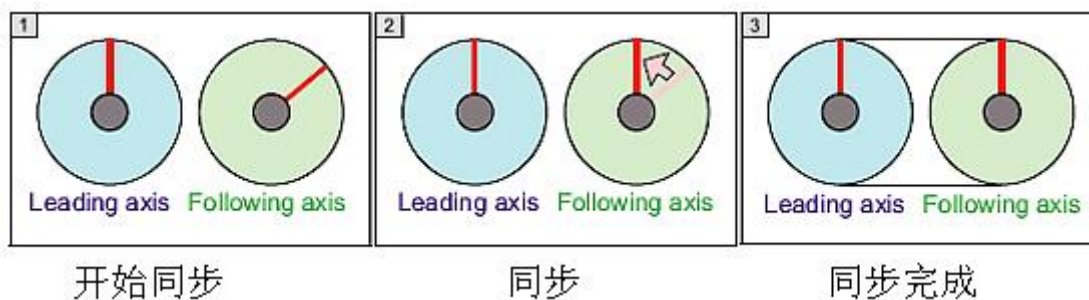


图27 绝对 Gearing 过程



图28 相对 Gearing 过程

4) Gearing Direction

- **POSITIVE:** 主值与从轴按相同方向运行
- **NEGATIVE:** 主值与从轴按相反方向运行
- **CURRENT:** 从轴的运动方向被保持，如果从轴静止，那么按 POSITIVE 运行
- **REVERSE:** 从轴的运动方向与主值相反

5) Look-ahead

- **STANDARD-LOOKAHEAD** 同步计算时考虑主轴的速度和位置
- **EXTENDED LOOK-AHEAD** 同步计算时考虑主轴的速度和位置以及加速度

MCC 中 GEARING ON 的参数和同步参数如下：

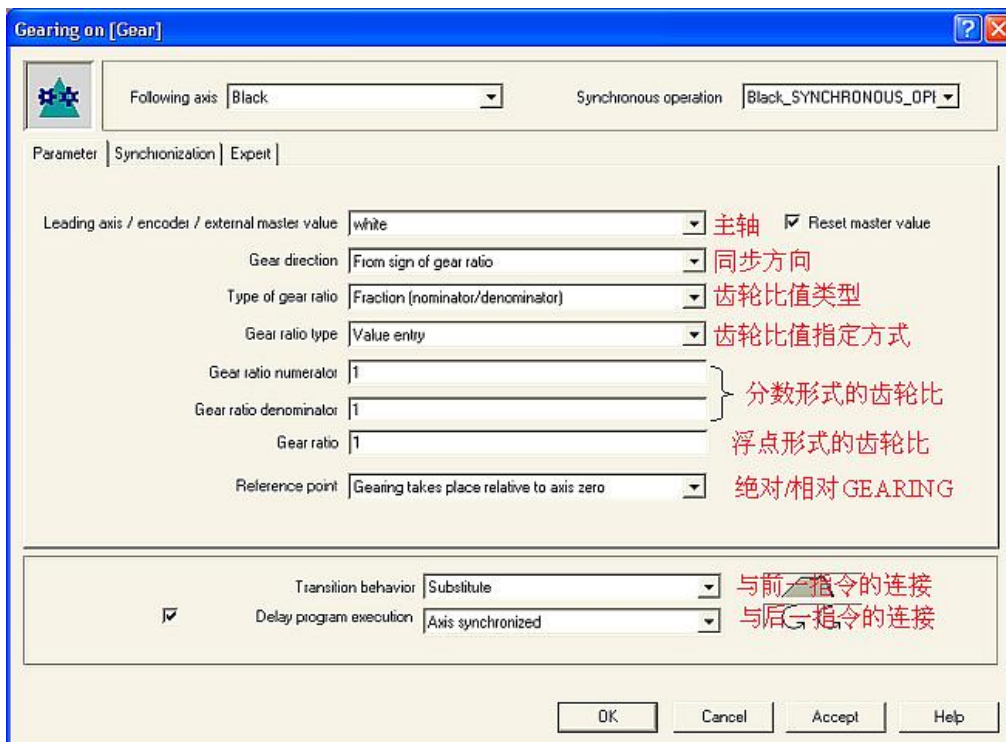


图29 MCC 中的 GEARING ON 参数

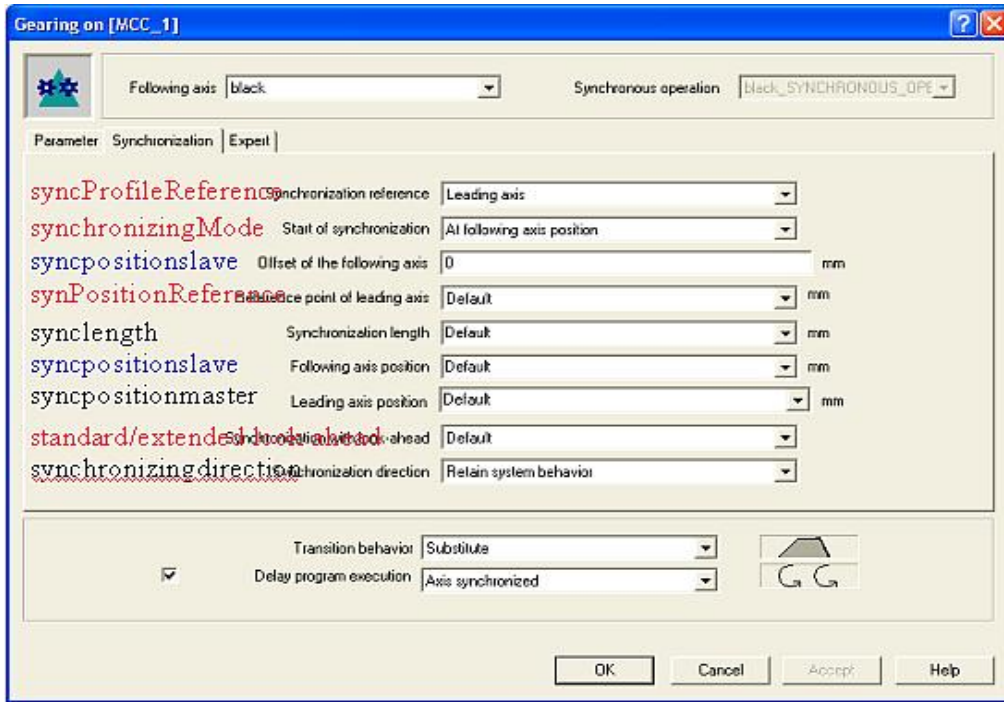


图30 MCC 中的 GEARING 同步参数

Gear 同步的参数含义请参考节 2.1 中的内容。

Gearing 同步的默认设置值

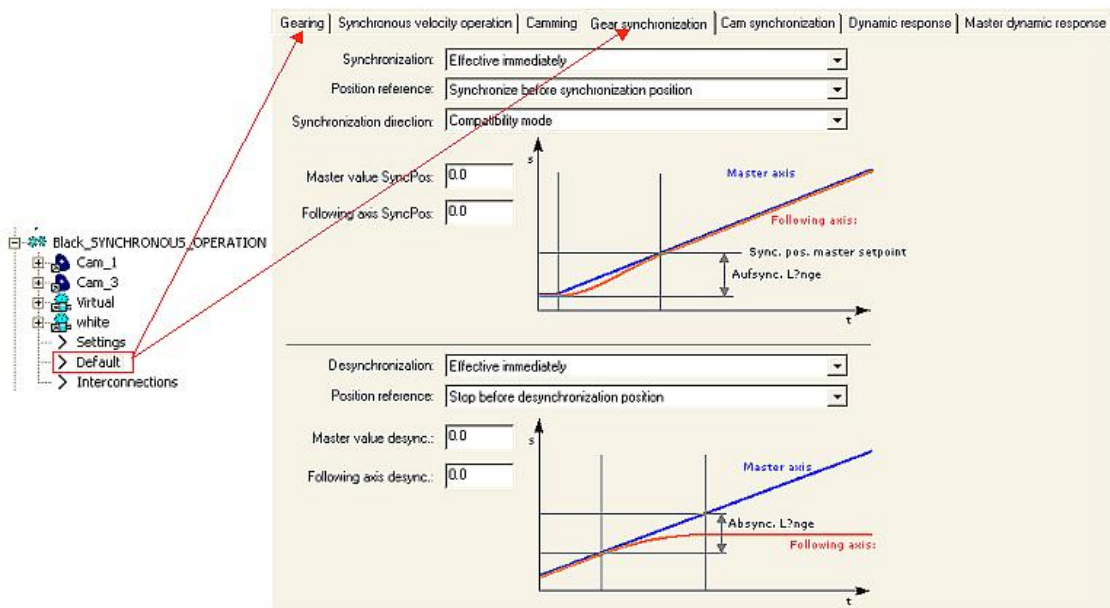


图31 GEAR 同步默认参数

4.2 基本配置过程

Simotion 的项目创建及驱动的配置请参看相关手册。下面以创建两个轴为例说明同步配置：

(1)离线插入主轴 Red 轴，即例中的主轴，选择其为位置轴，即在下图中选中 Speed control 和 Positioning.

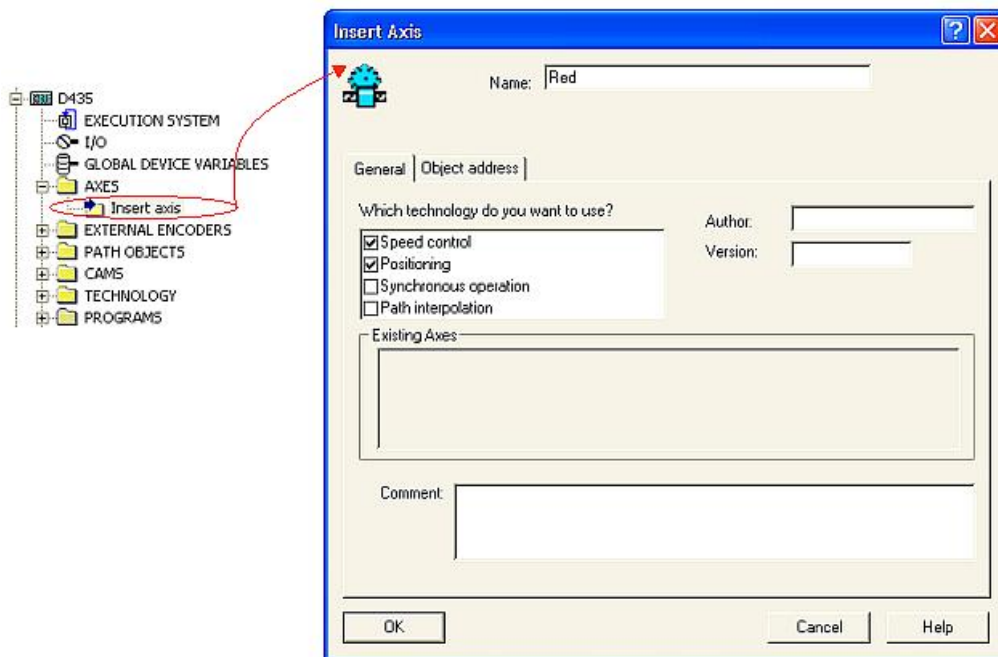


图32 插入主轴

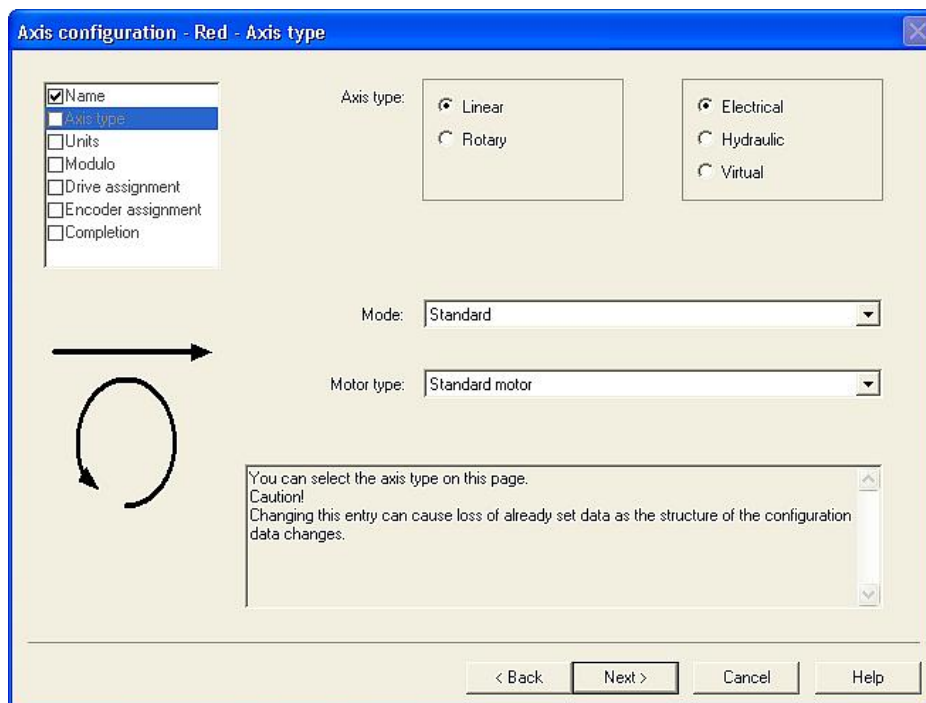


图33 选择线性轴

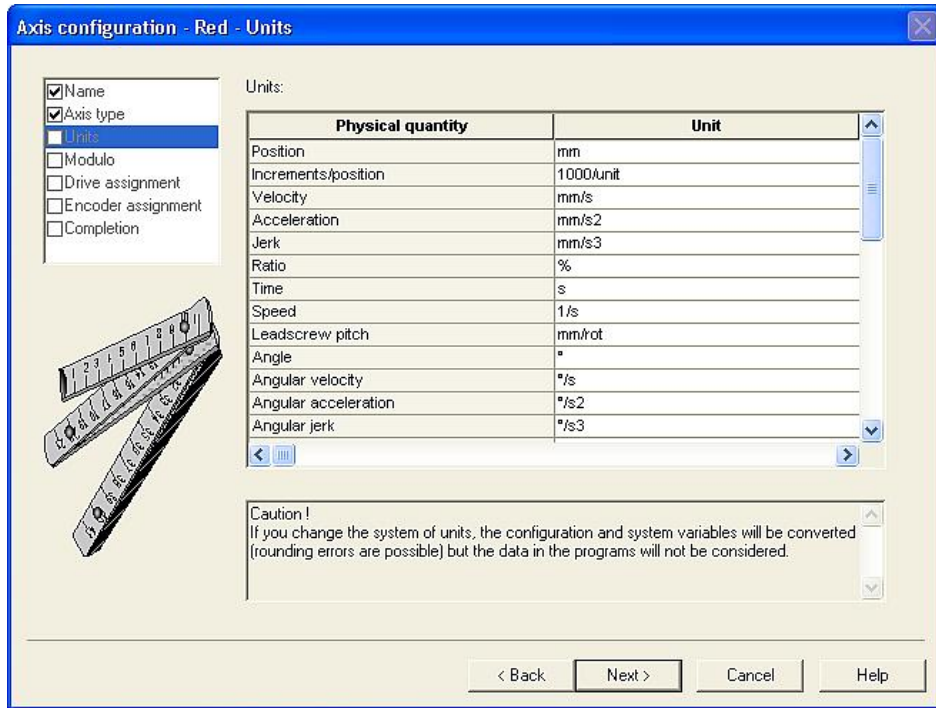


图34 单位保持默认

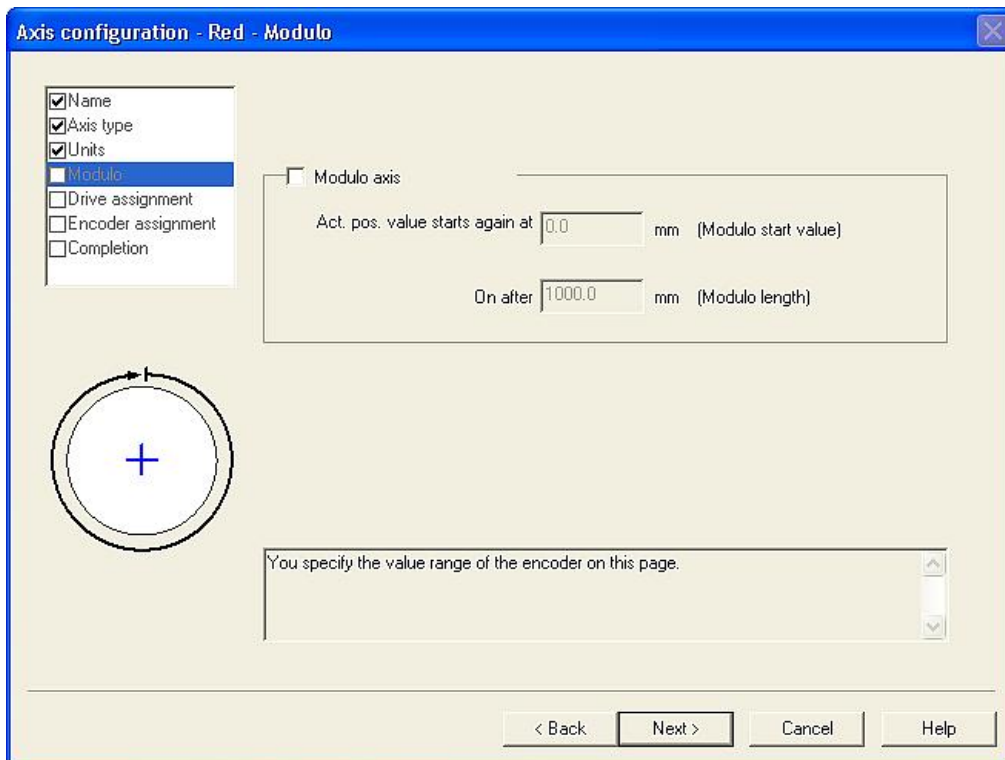


图35 选择非模态

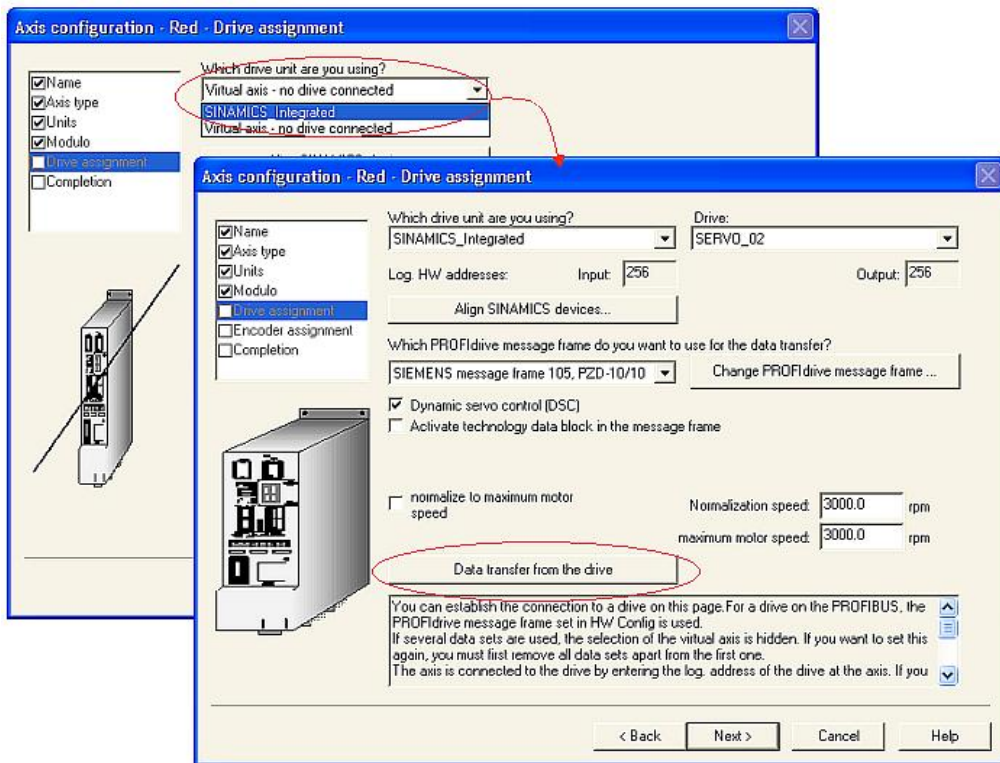


图36 选择连接集成的 SINAMICS 驱动

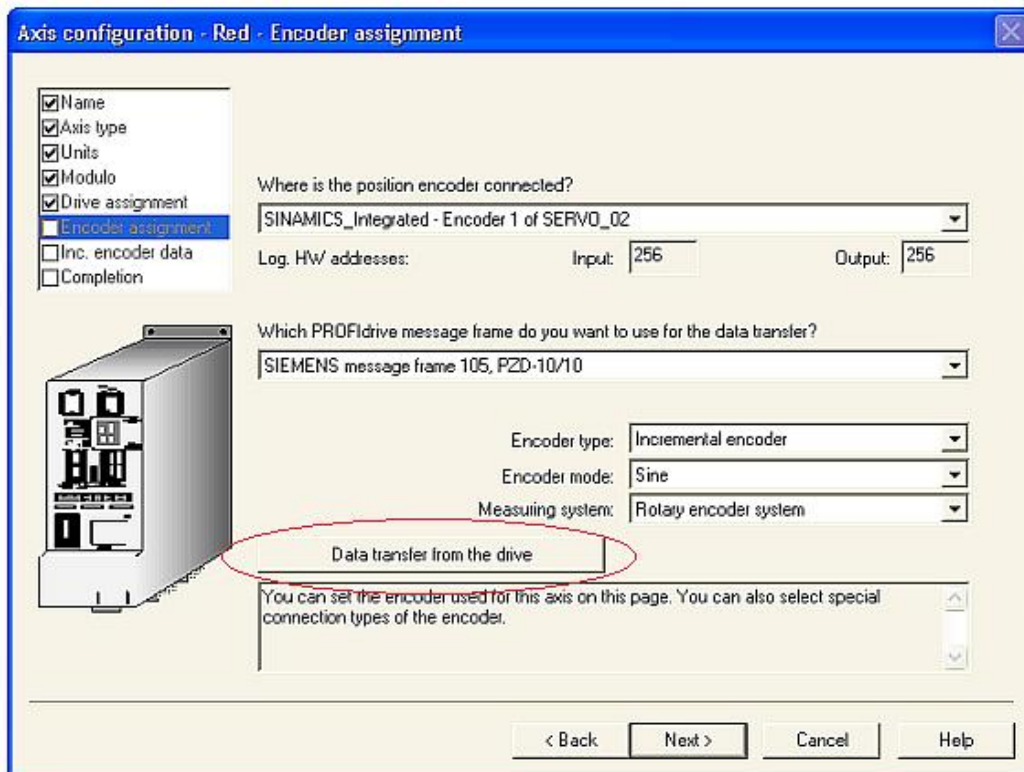


图37 选择位置环的编码器

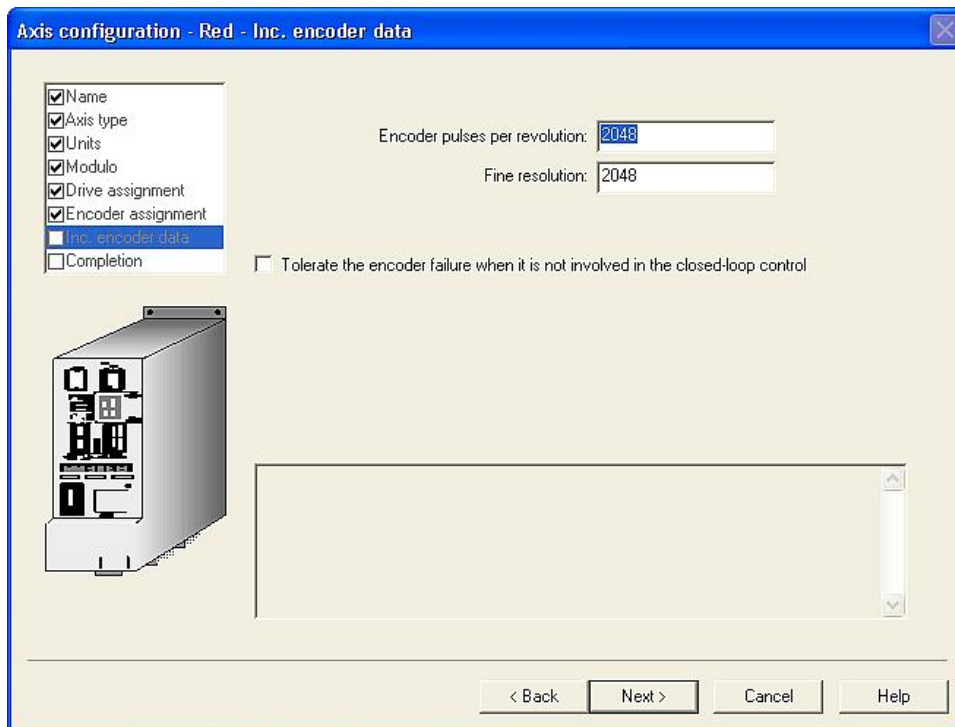


图38 设置增量编码器的脉冲数

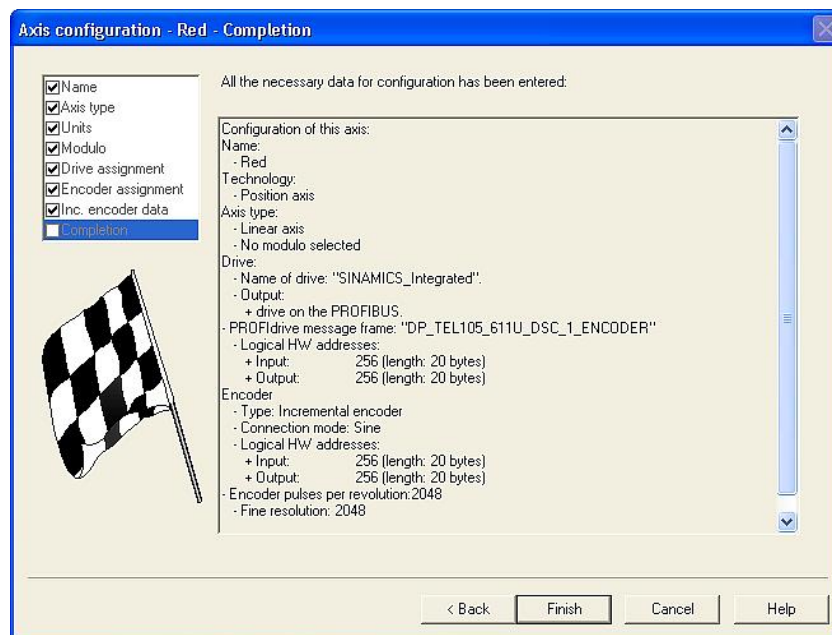


图39 完成轴的配置

(2)离线插入从轴 Blue 轴，并选择同步轴，即在图 32 中选中 Speed control 和 Positioning 以及 Synchronous Operation，并连接到驱动的 SERVOR_03 中，其它配置过程和配置 Red 轴相同。

(3)配置同步关系

按图 40 设置同步运行连接。

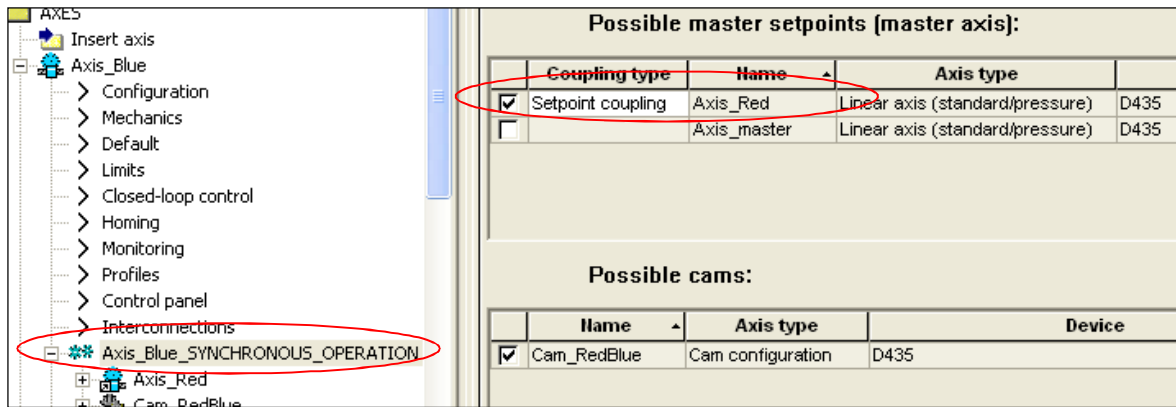


图40 设置同步运行连接

4.3 Gearing 编程

以 MCC 为例编程如下

首先插入 MCC 单元 Gearing，并建立如下的子程序块：

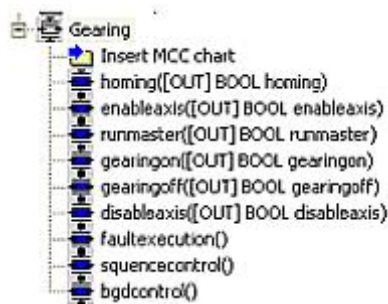


图41 程序块

Homing: 回零子程序，在 squencecnotrol 中调用

Enableaxis: 轴使能子程序，在 squencecnotrol 中调用

Runmaster: 运行主轴子程序，在 squencecnotrol 中调用

Gearingon: 建立同步子程序，在 squencecnotrol 中调用

Gearingoff: 解除同步子程序，在 squencecnotrol 中调用

Disableaxis: 停止主轴，并解除轴使能，在 squencecnotrol 中调用

Faultexecution: 故障处理子程序，为空程序，在故障处理中断中调用

Squencecontrol: 顺序控制子程序，在 MotionTask1 中调用

Bgdcontrol: 背景任务子程序，用于激活 MotionTask1

在 Gearing 单元中建立 Bool 变量 boGearonTestStart。

程序中的具体内容如下：

1) enable axis 子程序

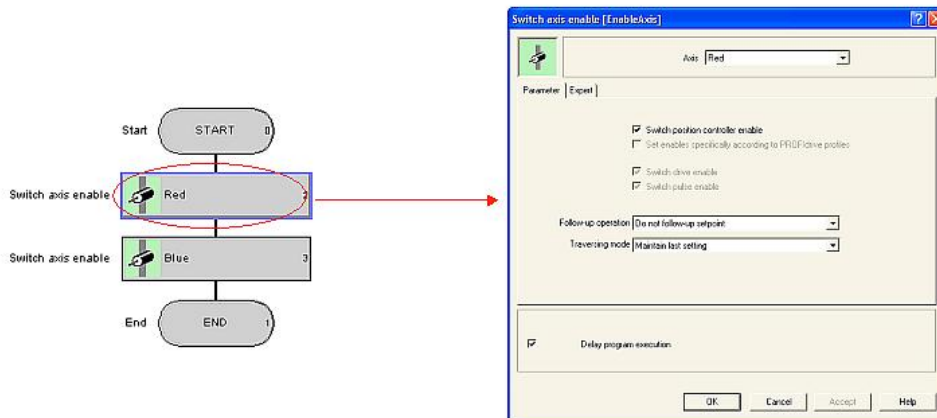


图42 enable axis 子程序

2) Homing 子程序

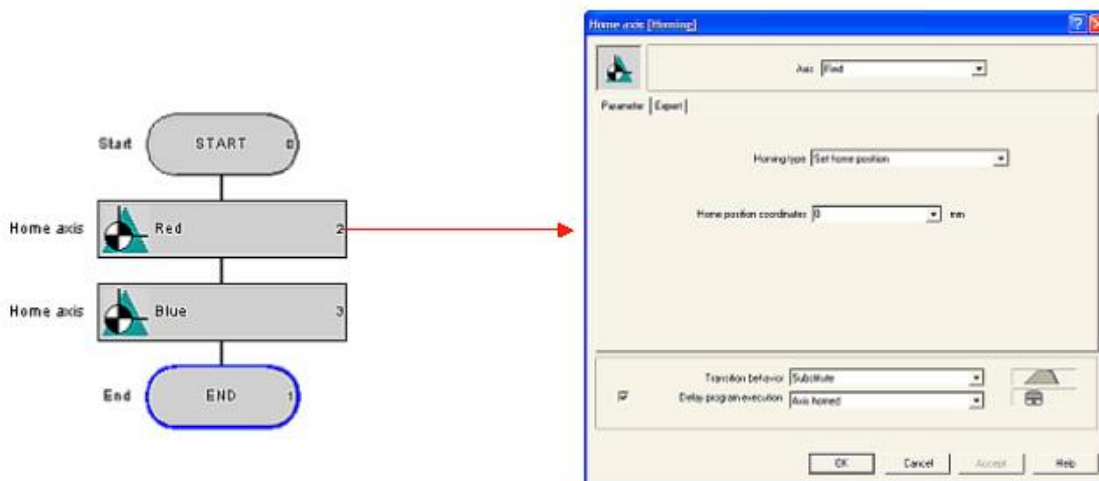


图43 Homing 子程序

3) 运行主轴子程序

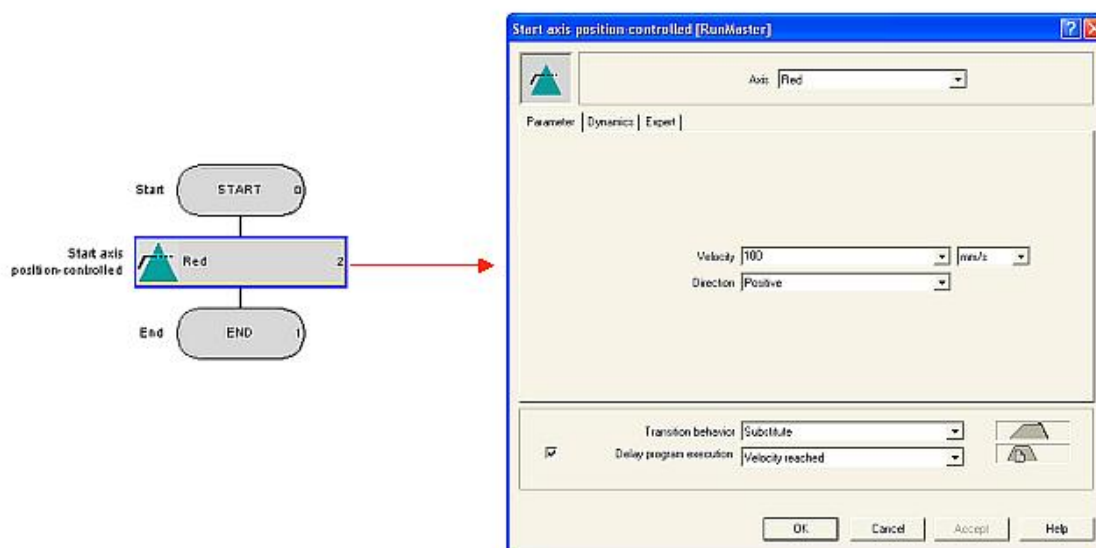


图44 主轴运行子程序

- 4) Gearing On 子程序，本例中采用绝对同步，用浮点数直接指定齿轮比，同步轮廓参考为基于位置长度的方式，同步模式为立即同步，同步方向为正向同步，同步长度为 100。

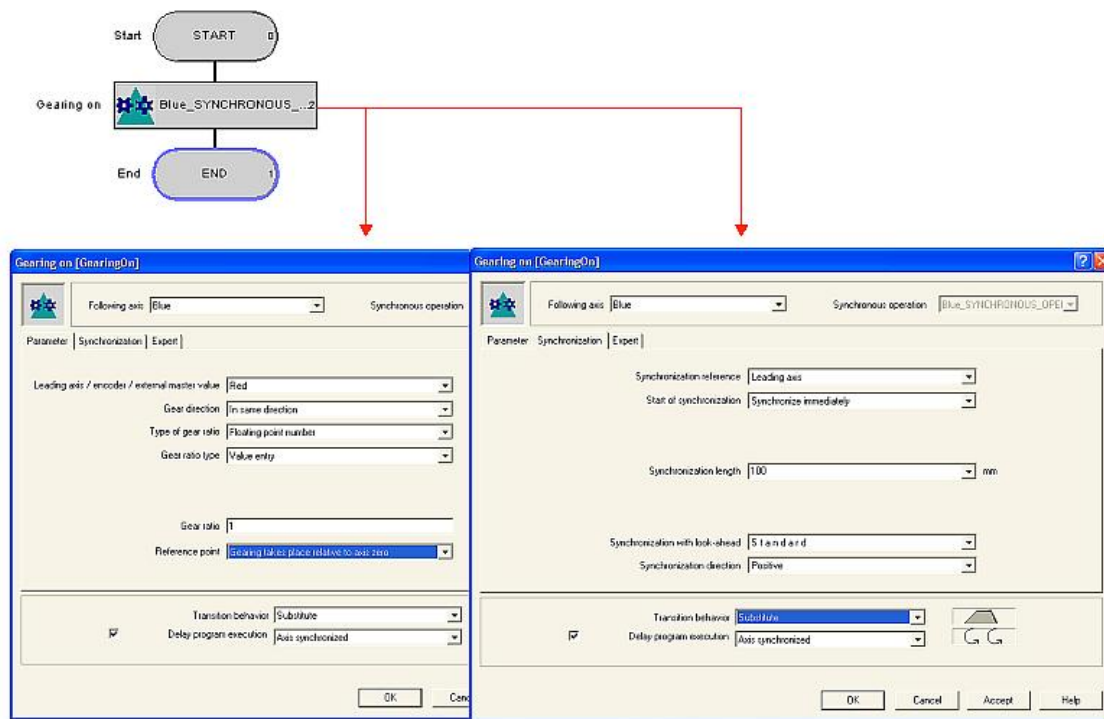


图45 Gearing On 子程序

- 5) Gearing Off 子程序

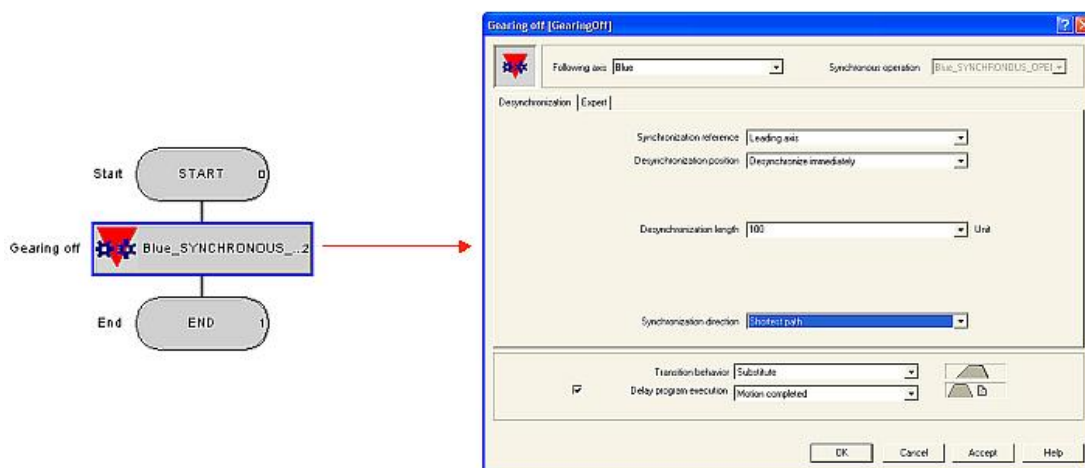


图46 Gearing Off 子程序

- 6) disableaxis 子程序：停止主轴，并解除轴使能

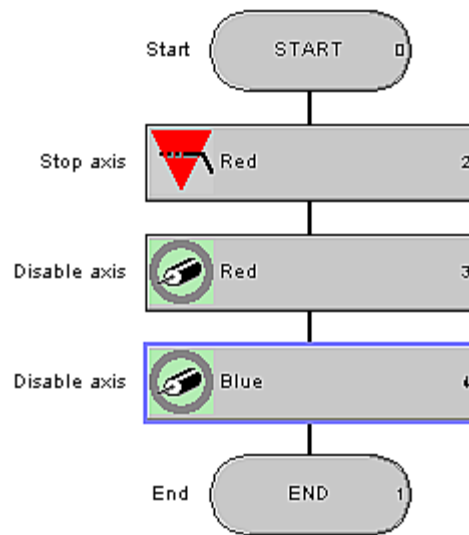


图47 disableaxis 子程序

- 7) Faultexecution 空程序，在 TechnologicalFaultTask 和 PeripheralFaultTask 中调用
- 8) squencecontrol 子程序：

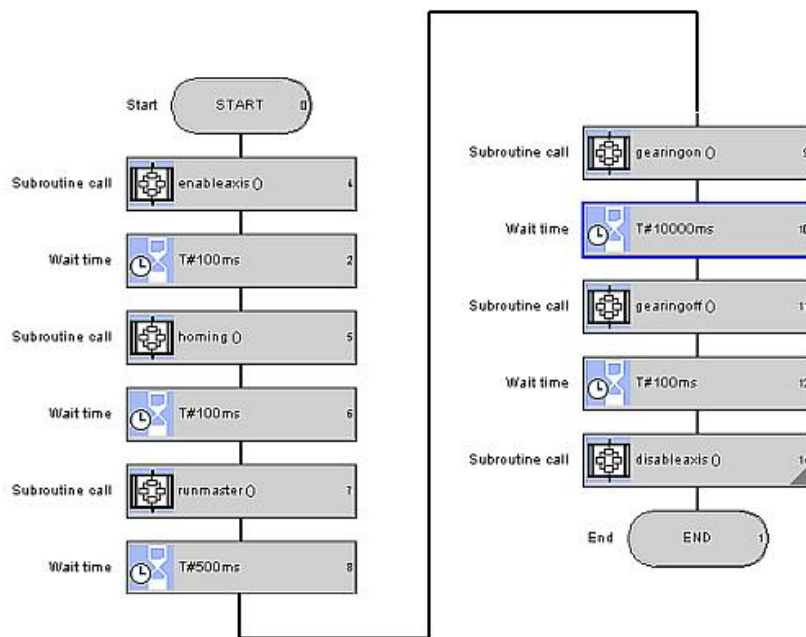


图48 squencecontrol 子程序

图49

- 9) Bgdcontrol 子程序，在 BackgroundTask 中调用，用于根据条件激活 MotionTask1

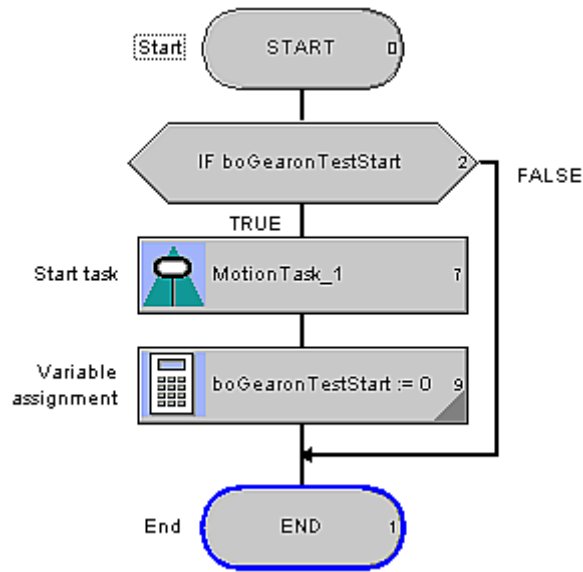


图50 Bgdcontrol 子程序

4.4 程序测试

在本例中，只需将 boGearingOnTest 设为 True,系统即按以下顺序运行：

- 1) 轴使能
- 2) 轴回零
- 3) 主轴运行
- 4) Gearing ON
- 5) Gearing OFF
- 6) 停止主轴
- 7) 轴去使能

建立状态表，将变量 boGearingOnTest 加入：

Watch_table_1:						
	Name	I/O address	Data type	Status value	Display format	
1	D435\Gearing.bogearonteststart		BOOL	FALSE	BOOL	<input checked="" type="checkbox"/> TRUE

图51 变量表

SIMOTION 运行后将其置为 TRUE 即开始运行。

用 Trace 记录主轴和从轴的位置以及同步状态信号，曲线如下：

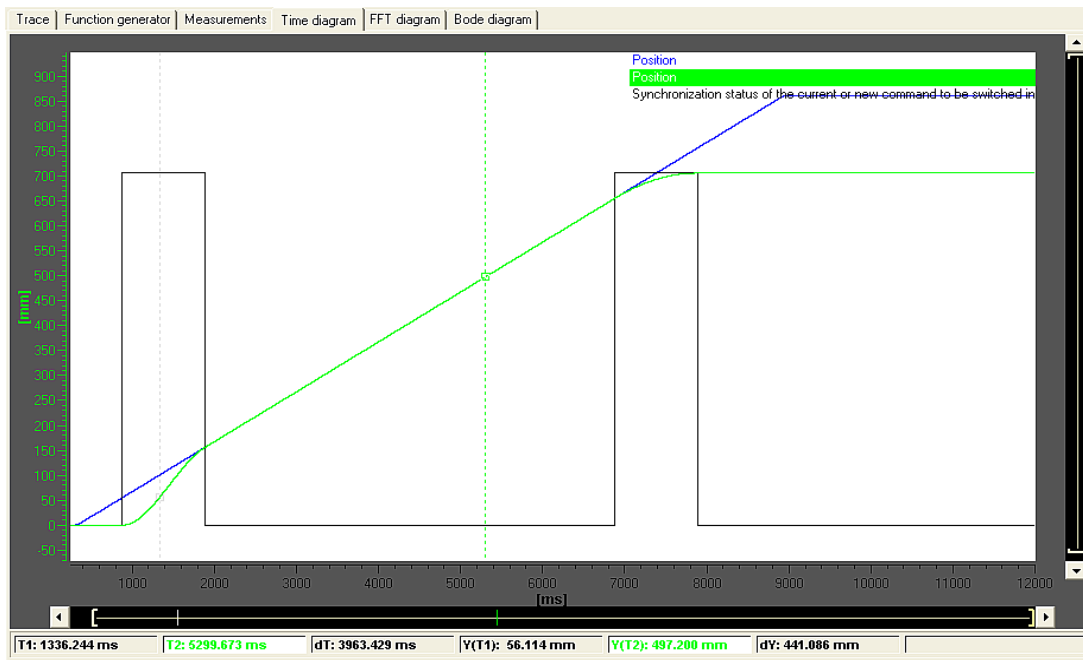


图52 Trace 曲线记录

程序例子请参见附件。

5 Velocity Gearing

5.1 Velocity Gearing 基本介绍

Gearing 和 Camming 都是实现位置间的函数关系，而 Velocity Gearing 则是实现主值和从轴间的速度函数关系。同步对象将计算出从轴的速度设定值，一旦同步启动，从轴即按照指定的加速度调整到同步速度。

其相关的重要因素如下：

1) 速度比

速度比可通过正的浮点数进行指定

2) 同步方向

- **POSITIVE:** 主值与从轴按相同方向运行
- **NEGATIVE:** 主值与从轴按相反方向运行

5.2 Velocity Gearing 程序实例

速度同步的程序和上面 Gearing 的例子程序绝大部分相同，只需修改其中的 GearingOn 子程序和 GearingOff 子程序即可：

1) Velocity Gearing On 子程序，本例中采用正向速度同步，速比为 2

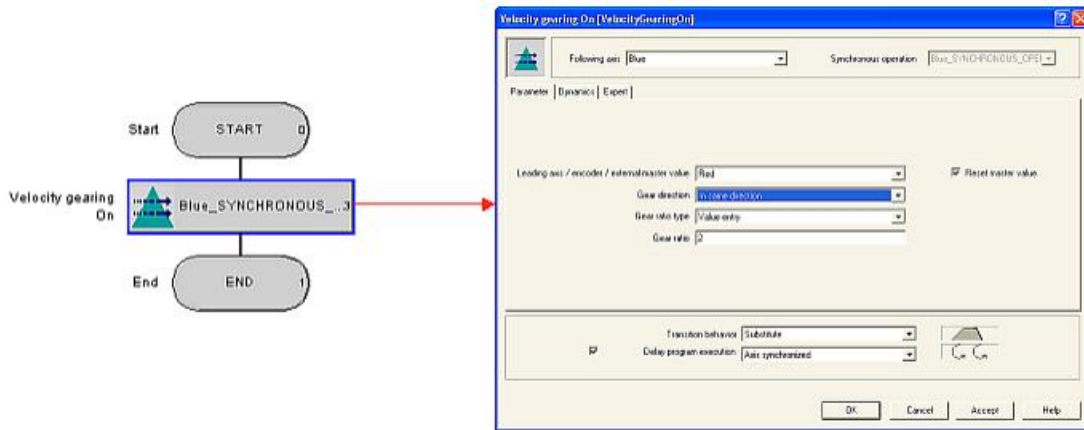


图53 Velocity Gearing On 子程序

2) Velocity Gearing Off 子程序

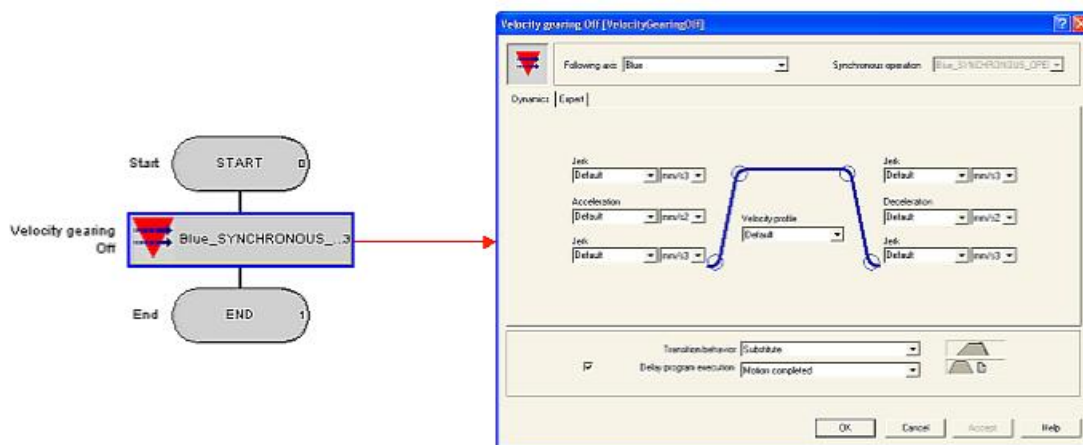


图54 Velocity Gearing Off 子程序

3) 用 Trace 记录主轴和从轴的速度以及同步状态信号



图55 Velocity Gearing 曲线记录

注: Velocity Gearing 在同步过程中的动态响应是在指令直接给定的, 本例中采用 Default:

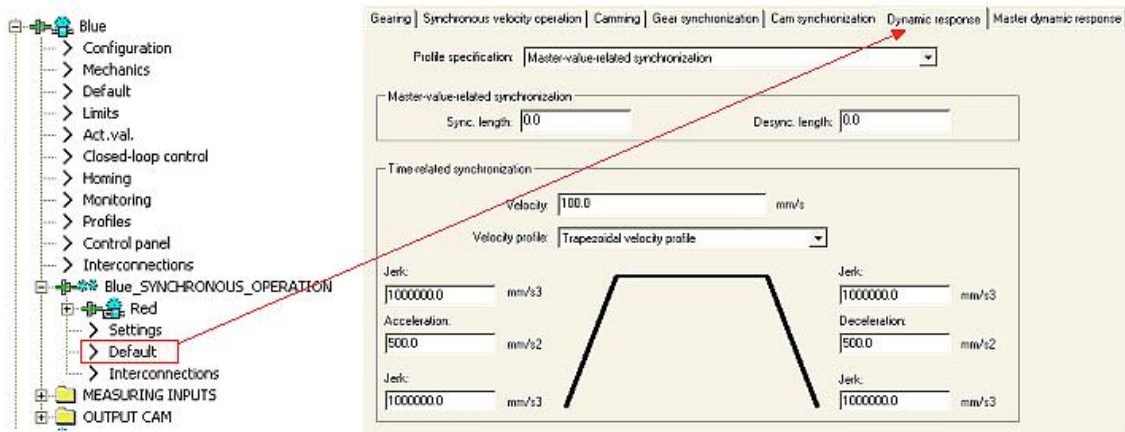


图56 动态响应的默认值

6 Camming

6.1 Camming 基本介绍

Camming 和 Gearing 类似，区别在于它完成的是主值与从轴间的非线性传递函数功能，而 Gearing 是线性函数关系。Camming 需要用 Cam 来确认非线性传递函数，按以下公式确认位置关系：

$$\text{Slave value} = \text{KS}(\text{Master value} + \text{Offset}_{\text{master value}}) + \text{Offset}_{\text{slave value}}$$

其中，KS 表示 Cam, 即传递函数

Camming 的相关重要因素如下：

1) Cam, 即传递函数

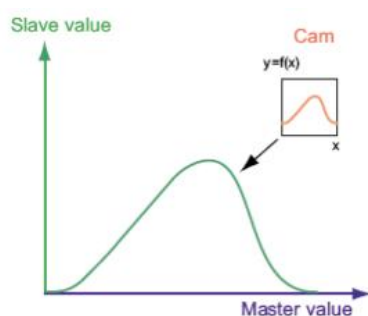


图57 Cam 曲线

在 6.2 中具体介绍了 Cam 生成的方法。

2) Master Mode / Slave Mode

Cam 同时支持主轴和从轴的绝对和相对模式，分别在 Master Mode 和 Slave Mode 中进行设定，以下图为例：

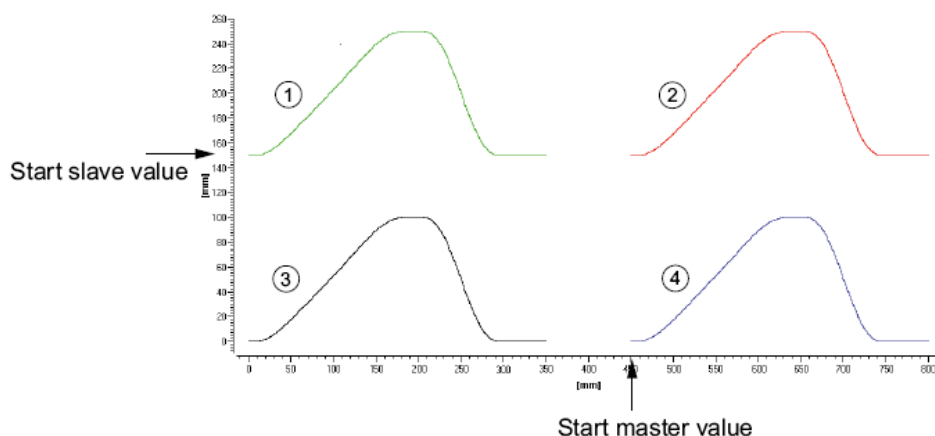


图58 绝对和相对模式示意图

图中的例子采用的 Cam 的主值位置范围为 {0-300}，从轴位置范围为 {0-100}，主轴为 0-1000 的模态轴。主轴和从轴的当前位置分别为 145mm 和 450mm，图中的 4 条曲线分别表示在主轴和从轴的相对和绝对模式下的运行轨迹。图中的 1#曲线表示主轴为绝对值模式，从为相对模式

的曲线，因为主轴从当前位置 450 切换到了 0 才进行同步，也就是等于 Cam 中主值的绝对值，因而是绝对模式，而从轴则保持为当前的 145 位置不变，而没有切换为 Cam 中从轴的绝对值 0，因而从轴为相对 Cam 模式。同理可看出，2#曲线表示主从均为相对模式的运行曲线，3#表示主从均为绝对模式的运行曲线，4#表示主为相对模式，而从为绝对模式的运行曲线。

注意，从轴为绝对模式时，当同步模式为 ON_MASTER_AND_SLAVE_POSITION 或 IMMEDIATELY_AND_SLAVE_POSITION 时仍可以通过同步参数 syncPositionSlave 产生偏差，从轴为相对模式时，也要考虑同步参数 syncPositionSlave 产生偏差，即当同步模式为 ON_MASTER_AND_SLAVE_POSITION 或 IMMEDIATELY_AND_SLAVE_POSITION 时偏差为当前值与 Cam 值之差加上 syncPositionSlave 产生的偏差才是最终的偏差值。

3) Camming Mode

Camming 模式有两种，即周期性 Cam(Cyclic Camming)和非周期性 Camming(Non-Cyclic Camming)。

非周期性 Camming 就是 Camming 同步只在 Cam 中定义的主值范围内进行一次，如果主值再次经过 Cam 自变量的范围或主值反向运行至该区段，Camming 同步也不会再发生。

周期性 Camming 是指主值的范围会周期性地镜像到整个主值范围。不管主管在哪个位置，都会保持 Camming 同步运行状态。如果 Cam 的因变量的开始点和结束点的位置相同，那么运动会平滑地周期性地运行。如果 Cam 的因变量的开始点和结束点的位置不同，那么从轴的位置设定值会不连续，但从轴的实际运动受限于动态响应，不能突变。

4) Cam direction

- **POSITIVE:** 即主值增加时，从轴值也增加，两者运动方向相同
- **NEGATIVE:** 即主值增加时，从轴值减小，两者运动方向相反

6.2 CAM 曲线的生成实例

下面是主轴与从轴操作的示例：

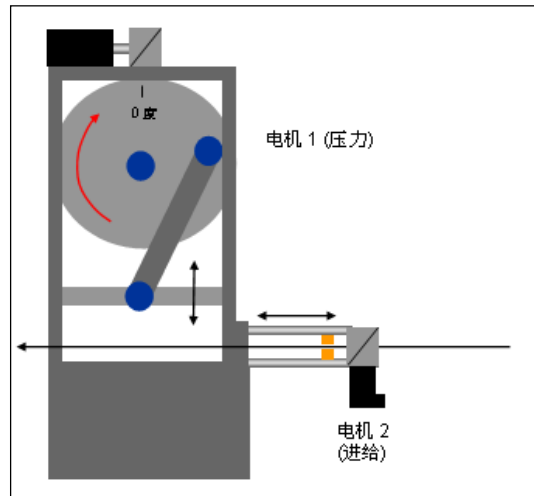


图59 CAM 简单应用

图中，压力电机作为一个旋转的主轴，进给电机作为直线运动的从轴，主轴反复旋转运动带动从轴反复直线运动。根据工艺参数先定义主轴从轴运行曲线，例如工艺参数如下：

抬升速度：在满足机械要求的情况下自定义

进给长度：0- 50mm

开始角度：1:250 度

结束角度：1: 0 度

开始角度：2: 50 度

结束角度：2: 150 度

根据要求绘制曲线图下图所示：

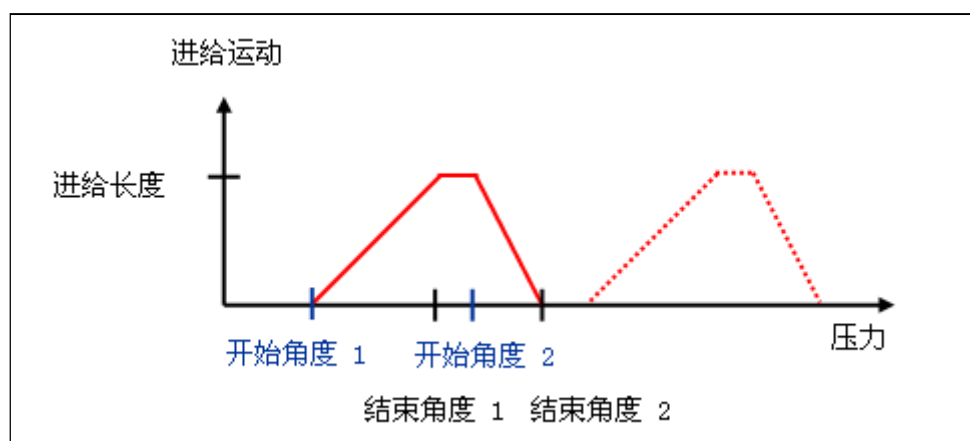


图60 绘制位置曲线

在这个例子中开始的角度为 250 度，为了便于在一个周期进行绘制，可以将角度左移 250 度，对应关系如下图所示：

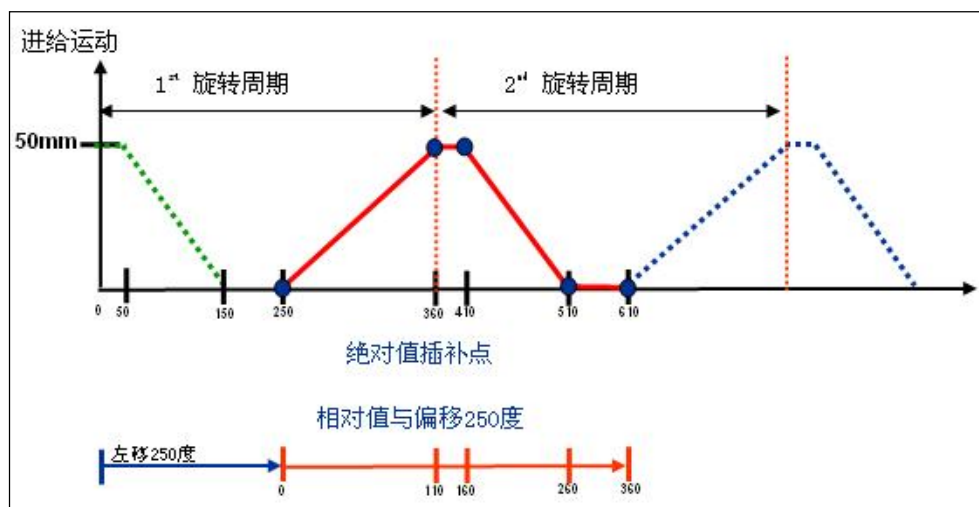


图61 曲线左移后的对应关系

主轴与从轴的同步操作位置曲线可以通过 CAM 的工具进行设置。CAM 曲线工具分为插补表和多项式两种。

6.2.1 通过插补表设定 CAM 曲线

点击树状菜单“ Cam ” ->“ Insert cam ” 插入一个 CAM 曲线，选择曲线类型为“ Interpolation point table ” 弹出插补点表如下图所示：

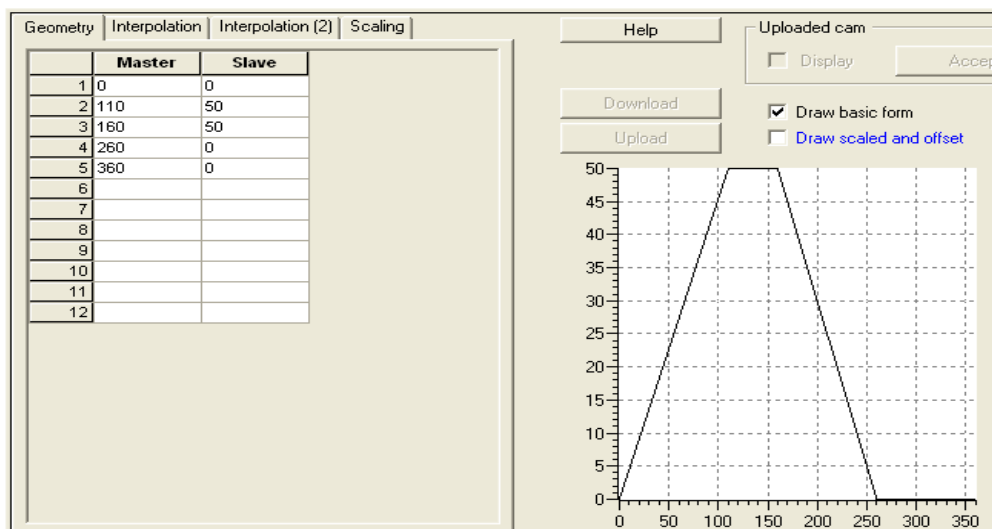


图62 C
AM 插补
点表

点击
“ Geometry ” 按
工艺要求
将插补点
填入表中，

在右边的视图中出现绘制的曲线。

点击“ Interpolation2 ” 如下图所示：

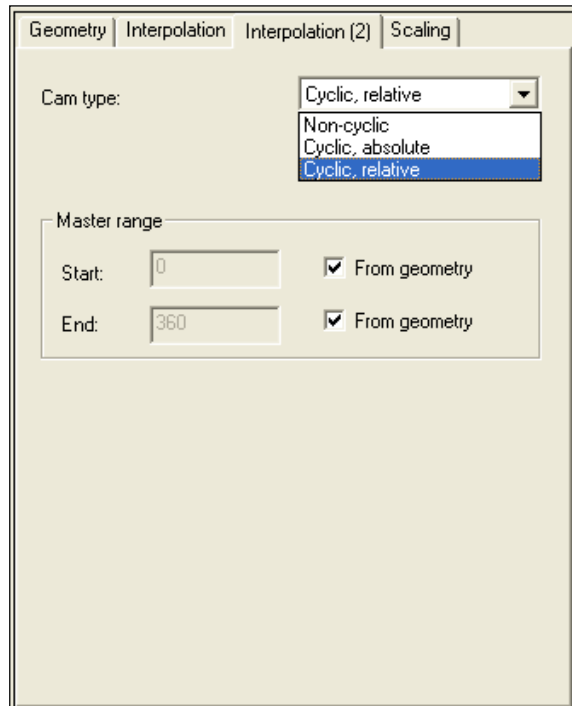


图63 CAM 插补点表—Interpolation2 属性界面

在界面中可以选择主轴的范围及 CAM 的类型。点击标签“ Scaling”可以对主轴从轴位置进行缩放如下图所示：

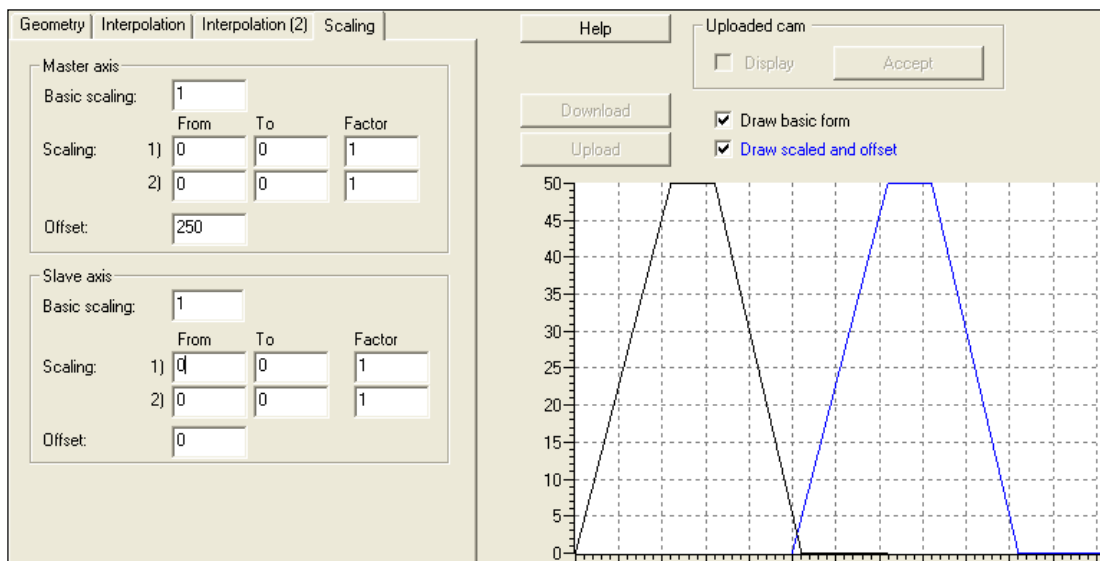


图64 CAM 插补点表—Scaling 属性界面

在图中选择偏移 250 度时选择“ Draw scaled and offset”选项后，可以显示经过缩放功能后的位置曲线，通过颜色与原位置曲线区分。这样就生成一个 CAM 曲线。

6.2.2 通过多项式生成 CAM 曲线

通过多项式生成 CAM 曲线可以使速度变化比较圆滑，点击树状菜单“ Cam” ->“ Insert cam” 插入一个 CAM 曲线，选择曲线类型为“ Polynomial”，点击 OK 键确认后弹出操作界面如下图所示：

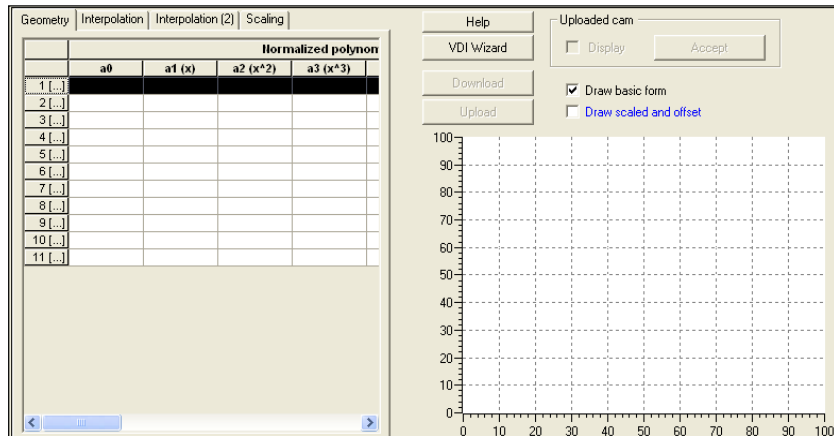


图65 多项式生成 CAM 曲线界面

点击“ Geometry”输入多项式的系数，在右边的图示中将自动生成曲线，一个曲线可以由多个多项式组成。各个多项式组成的曲线往往不能平稳连接（头尾）或有间隙，这样需要在“ Interpolation”栏中进行设置如下图所示，在界面中设置间隙、如果出现间隙后采样哪种样条曲线进行插补以及曲线出现交叉后处理的方法。

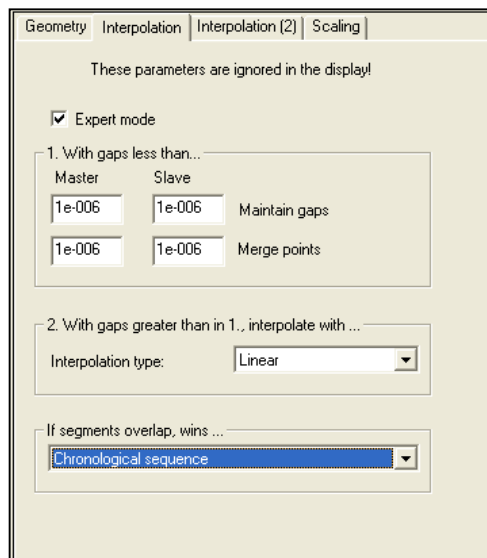


图66 CAM —Interpolation 属性界面

在右边的视图中也可以选择“ VDI Wizard”的方法生成多项式曲线，例如可以将使用插补点表生成的位置曲线进行速度优化，首先将原曲线分为4条曲线（0,0~110,50； 110,50~160,50； 160,50~260,0； 260,0~360,0），分别生成4个多项式，每个多项式表示一条曲线，4条曲线组成一条曲线，点击“ VDI Wizard”按钮，在样图中选择“ DWELL TO DWELL” ->在选择“ Symmetric”格式—>定义开始点和结束后生成第一条曲线，以相同的方法生成其它3条曲线。4条曲线中第二和第四条曲线为直线，在样图中应选择“ DWELL STATE”，这样曲线生成如下图所示，速度变化得到优化。

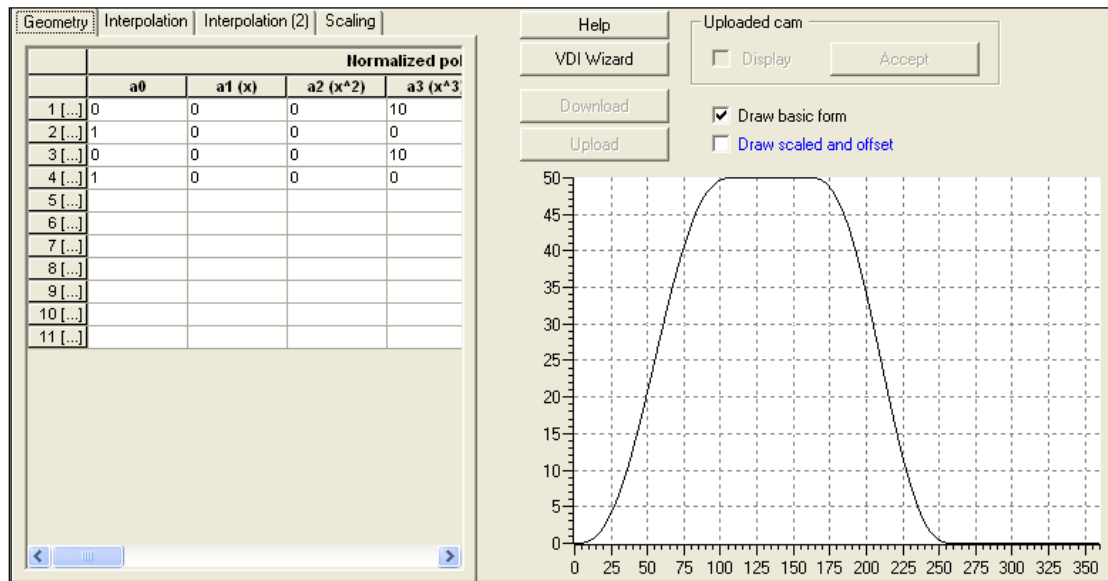


图67 CAM 多项式生成曲线

6.3 Camming 程序实例

首先轴的组态过程和 Gearing 完全相同。

Cam 曲线的生成采用 6.2 中的多项式的方法生成，如下图：

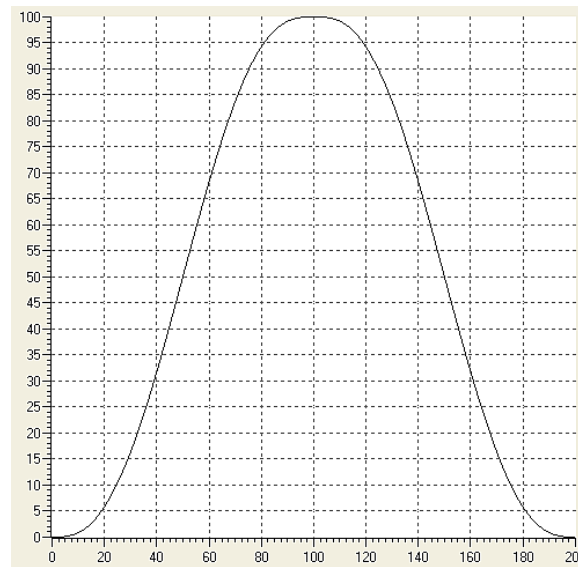


图68 CAM

程序部分和 GEARING 基本相同，除了将 GearOn 和 GearingOff 换成 CammingOn 和 CammingOff 即可，CammingOn 和 CammingOff 子程序如下：

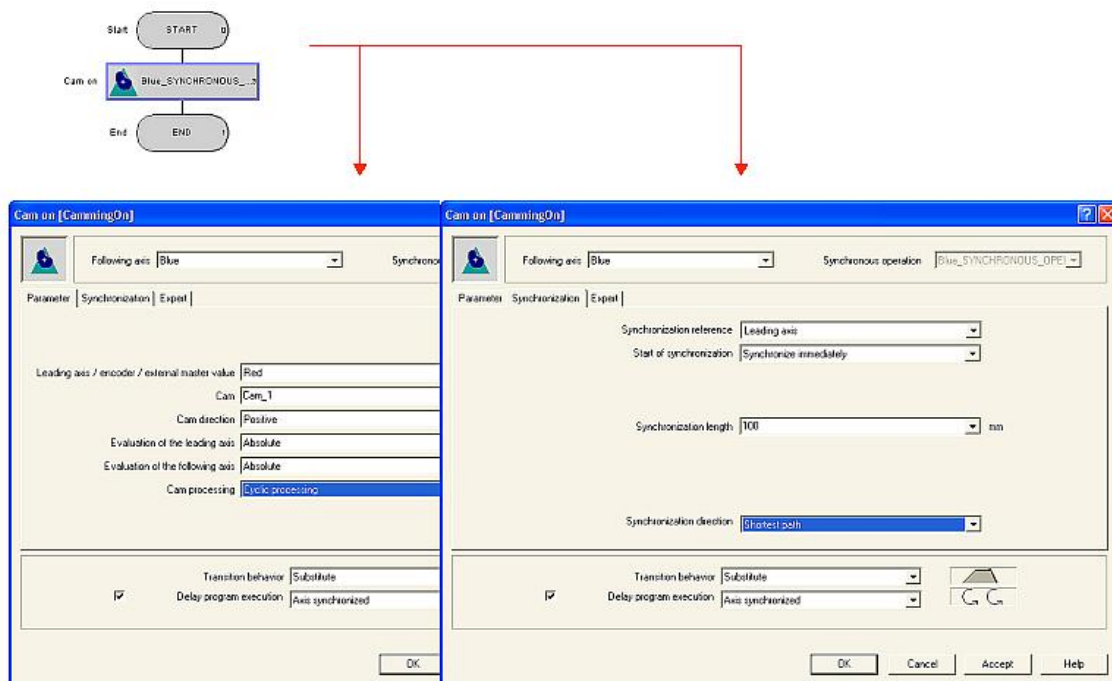


图69 CammingOn 子程序

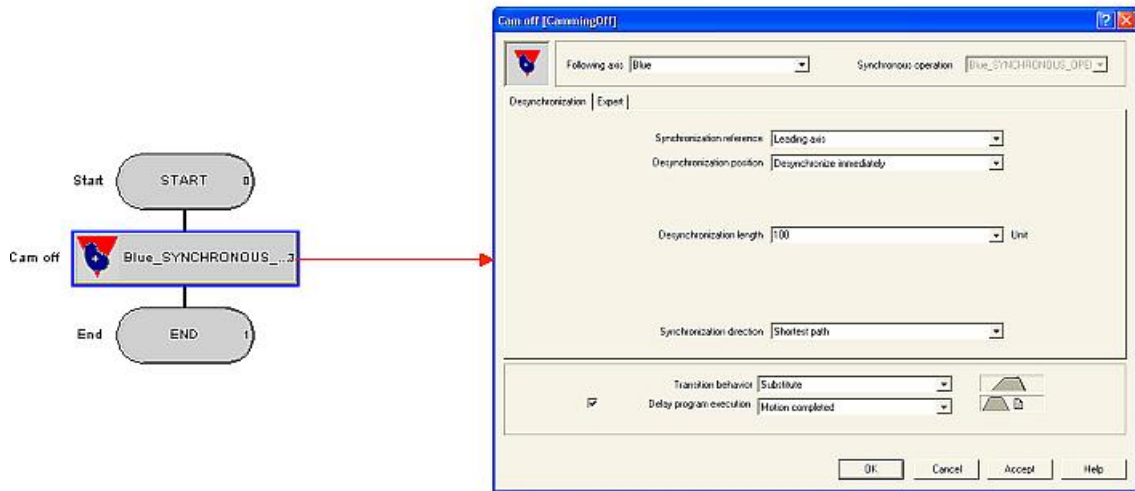


图70 CammingOff 子程序

在本例中，主轴和从轴均采用绝对同步，Camming 模式为 Cyclic 同步。

Camming 同步的 Trace 曲线记录：

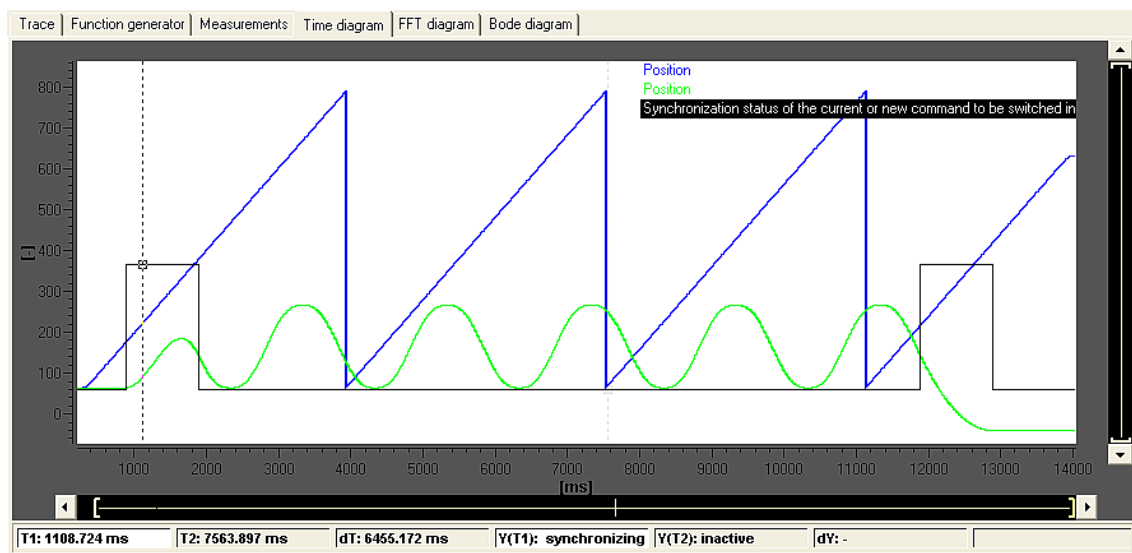


图71 Camming 曲线记录

参考文献

SIMOTION CAM 功能使用入门

<http://www.ad.siemens.com.cn/download/searchResult.aspx?searchText=SIMOTION+CAM>

SIMOTION Technology Objects Synchronous Operation, Cam Function Manual

<http://support.automation.siemens.com/CN/view/zh/32817445>