9

Motion Control Functions

This section describes the motion control functions that are used when connected to OMRON G5-series Servo Drives with built-in EtherCAT communications.

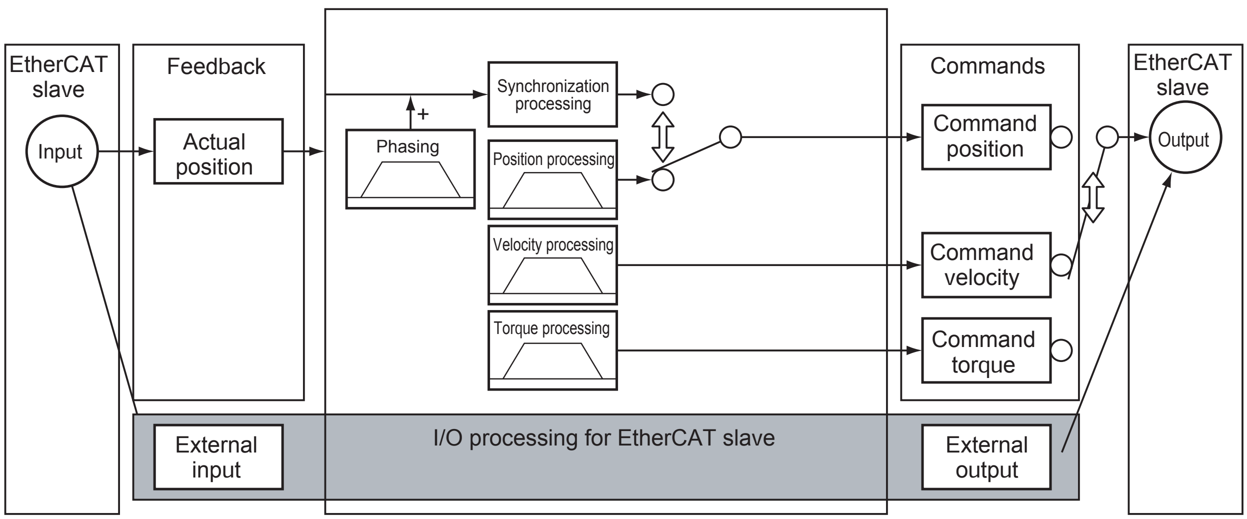
9-1 Single-axis Position Control

The MC Function Module can be connected to OMRON G5-series Servo Drives with built-in EtherCAT communications to implement position control, velocity control, and torque control. This section describes positioning operation for single axes.

Some of the functions of the MC Function Module are different when NX-series Pulse Output Units are used. Refer to the *NX-series Position Interface Units User's Manual* (Cat. No. W524) for details.

9-1-1 Outline of Operation

The single-axis control function of the MC Function Module consists of control for motion profile commands and synchronized control. There are three Control Modes for motion profile commands: position control, velocity control, and torque control. In synchronized control, the slave axis (i.e., the axis being controlled) operates in a synchronized relationship to the master axis, as expressed by a cam profile curve or a gear ratio. Manual operations such as jogging and homing are also supported.



Note You can use the command position or actual position as the input to the synchronization processing.

Resetting Axis Errors

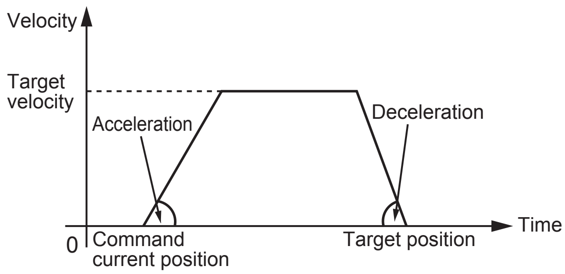
If an error occurs in an axis, you can use the MC_Reset instruction to remove the error once you have eliminated the cause.

For details on resetting axis errors, refer to the MC_Reset (Reset Axis Error) instruction in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

Refer to the *NX-series Position Interface Units User's Manual* (Cat. No. W524) for the differences when you use NX-series Pulse Output Units.

9-1-2 Absolute Positioning

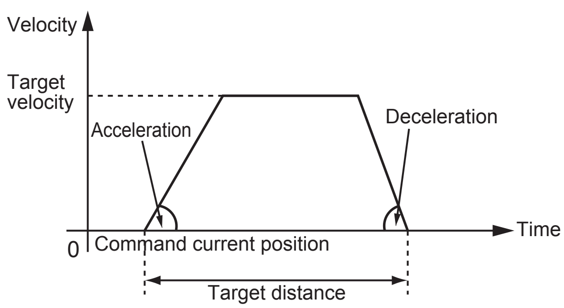
Absolute positioning specifies the absolute coordinates of the target position in relation to home. You can perform positioning, such as shortest way positioning on a rotary table, by setting the Count Mode to Rotary Mode and specifying the operation direction.



For details, refer to the MC_MoveAbsolute (Absolute Positioning) and MC_Move (Positioning) instructions in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

9-1-3 Relative Positioning

Relative positioning specifies the distance from the actual position. You can specify a travel distance that exceeds the ring counter range by setting the Count Mode to Rotary Mode.



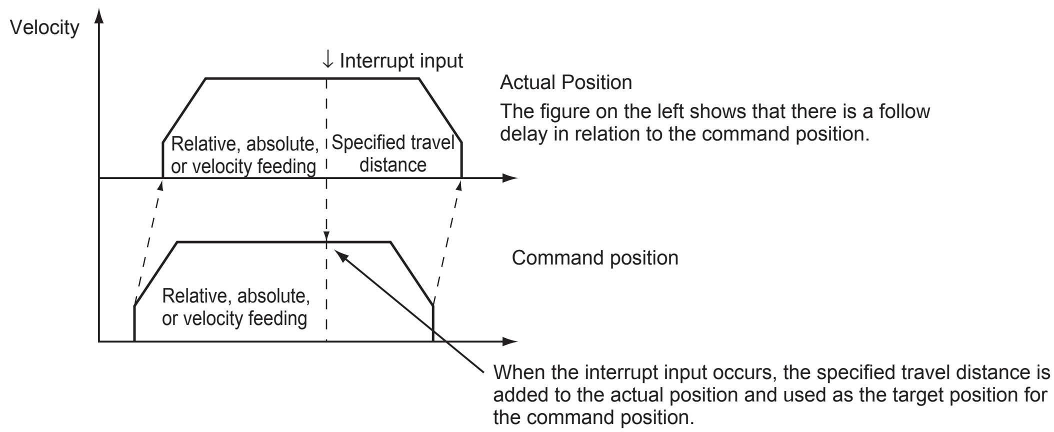
For details, refer to the MC_MoveRelative (Relative Positioning) and MC_Move (Positioning) instructions in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

9-1-4 Interrupt Feeding

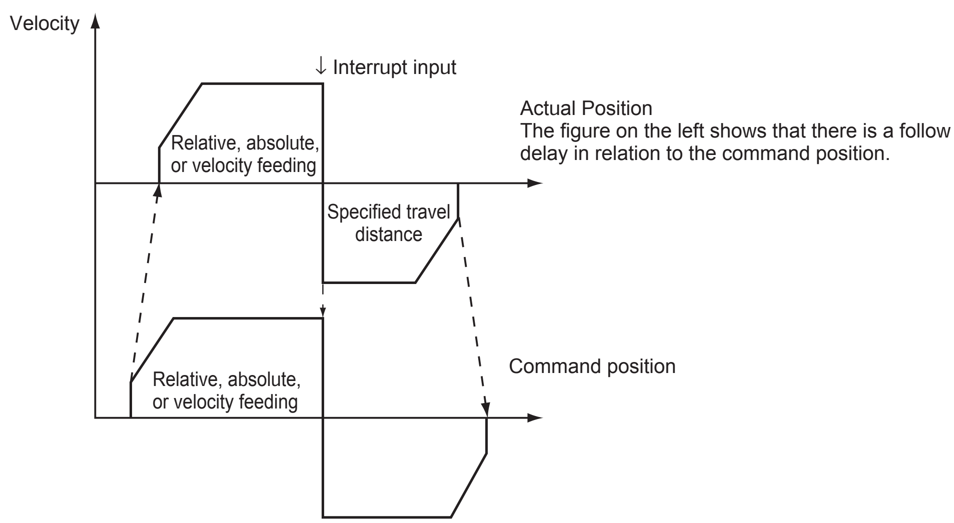
Interrupt feeding feeds the axis at the specified velocity and for the specified distance from the actual position when a trigger signal occurs. You can also select to output an error if the trigger signal does not occur within the specified travel distance when you specify either absolute or relative travel. Feeding is not affected by following error. This is achieved by using the latch function of the Servo Drive to determine the actual position when the trigger signal occurs. You can also use the window function to disable trigger signals that occur outside of a specified position range. For applications such as wrapping machines, this enables feeding only on trigger signals for printed marks on films and eliminates other influences.

● Motion Relative to the Actual Position

Feeding for a Specified Distance in the Moving Direction



Feeding for a Specified Distance in the Direction Opposite to the Moving Direction



If decelerating to a stop after a reverse turn is specified for the Operation Selection at Reversing axis parameter, an acceleration/deceleration curve is used when reversing.

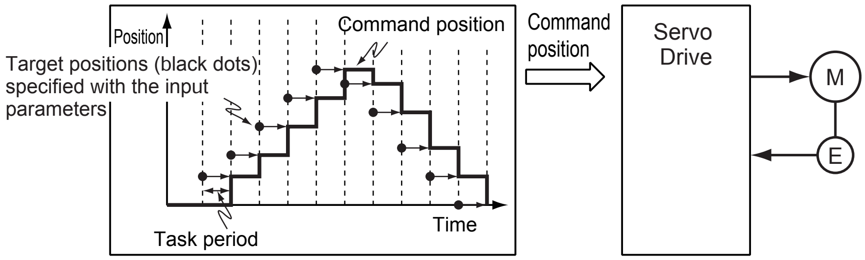
For details, refer to the MC_MoveFeed (Interrupt Feeding) instruction in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

Refer to the *NX-series Position Interface Units User's Manual* (Cat. No. W524) for the differences when you use NX-series Pulse Output Units.

9-1-5 Cyclic Synchronous Positioning

Cyclic synchronous positioning is used to output a target position to a specified axis each control period in the primary periodic task or a periodic task. The target position is specified as an absolute position.

You can use it to move in a specific path that you create.



For details, refer to the MC_SyncMoveAbsolute (Cyclic Synchronous Absolute Positioning) instruction in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

Version Information

A CPU Unit with unit version 1.03 or later and Sysmac Studio version 1.04 or higher are required to use cyclic synchronous positioning.

9-1-6 Stopping

Functions to stop axis operation include immediate stop input signal and limit input signals connected to the Servo Drive, stop functions of motion control instructions in the user program, and stopping due to errors.

Refer to the *NX-series Position Interface Units User's Manual* (Cat. No. W524) for the differences when you use NX-series Pulse Output Units.

Stopping for Servo Drive Input Signals

Axis motion is stopped for the immediate stop input signal or a limit input signal from the Servo Drive. You can select the stop method with the Sysmac Studio.

● Immediate Stop Input

Stop processing in the MC Function Module is executed according to the state of the Servo Drive input signals. You can select one of the following stopping methods for the MC Function Module.

- Immediate stop
- Immediate stop and error reset
- Immediate stop and Servo OFF

Precautions for Correct Use

The immediate stop input for the OMRON G5-series Servo Drive also causes an error and executes stop processes in the Servo Drive itself.

● Limit Inputs (Positive Limit Input or Negative Limit Input)

Stop processing in the MC Function Module is executed according to the state of the Servo Drive input signals. You can select one of the following stopping methods for the MC Function Module.

- Immediate stop
- Deceleration stop
- Immediate stop and error reset
- Immediate stop and Servo OFF

Precautions for Correct Use

- If a limit input signal turns ON, do not execute an instruction for axis command of the axis in the same direction as the limit input signal.
- If a limit input signal is ON for any axis in an axes group, do not execute an instruction for an axes group command for that axes group.
- If the signal to decelerate to a stop is input during execution of a synchronous movement instruction that has a *Deceleration* input variable, the axis decelerates to a stop at the deceleration rate given by *Deceleration*.
- If the signal to decelerate to a stop is input during execution of a synchronous movement instruction that does not have a *Deceleration* input variable, the axis decelerates to a stop at the maximum deceleration rate that is set in the axis parameters.

Additional Information

- You must set up the Servo Drive in order to use the input signals from the Servo Drive. An OMRON G5-series Servo Drive with built-in EtherCAT communications has an immediate stop input and limit input assigned in its default settings.
- Refer to *A-1 Connecting the Servo Drive* for setting examples for connection to an OMRON G5-series Servo Drive.

Stopping with Motion Control Instructions

Use the MC_Stop or MC_ImmediateStop instruction to stop single-axis operation.

● MC_Stop Instruction

You can specify the deceleration rate and jerk for single-axis control and synchronized control to decelerate to a stop. Specify a deceleration rate of 0 to send a command that immediately stops the Servo Drive. Other operation commands are not acknowledged while decelerating to a stop for this instruction and while the input variable *Execute* is TRUE.

● MC_ImmediateStop Instruction

You can perform an immediate stop for single-axis control or synchronized control functions. You can also execute this instruction on axes that are enabled in an axes group.

For details, refer to the MC_Stop and MC_ImmediateStop instructions in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

Additional Information

When the input variable *Enable* to the MC_Power (Servo ON) instruction changes to FALSE, the MC Function Module immediately stops the command value and turns OFF the Servo. When the Servo is turned OFF, the Servo Drive will operate according to the settings in the Servo Drive.

Stopping Due to Errors or Other Problems

● Stopping for Errors during Single-axis Operation

When an error occurs during single-axis operation, the axis will stop immediately or decelerate to a stop depending on the error. Refer to *11-2-2 Error Descriptions* for details on the stop method for each error.

● Stopping for a Software Limit

To stop for a software limit, set the Software Limits axis parameter. You can select from the following stop methods for the software limits.

- Enabled for command position. Decelerate to a stop.
- Enabled for command position. Immediate stop.
- Enabled for actual position. Decelerate to a stop.
- Enabled for actual position. Immediate stop.

Refer to *9-8-5 Software Limits* for details on software limits.

● Stopping Due to Excessively Long Motion Control Period

If motion control processing does not end within two periods, it is considered to be an excessive control period. Control will be stopped immediately.

Errors That Cause the Servo to Turn OFF

An immediate stop is performed if an error occurs that causes the Servo to turn OFF. When the Servo is turned OFF, the Servo Drive will operate according to the settings in the Servo Drive.

Stopping Due to Start of MC Test Run

All axes will decelerate to a stop at their maximum deceleration if a MC Test Run is started from the Sysmac Studio.

Stopping Due to End of MC Test Run

All axes will decelerate to a stop at their maximum deceleration if a MC Test Run is stopped from the Sysmac Studio.

- Click the **Stop MC Test Run** Button on the MC Test Run Tab Page of the Sysmac Studio.
- Close the MC Test Run Tab Page on the Sysmac Studio.
- Exit the Sysmac Studio.

Stopping Due to Change in CPU Unit Operating Mode

All axes will decelerate to a stop at their maximum deceleration if the CPU Unit operating mode changes.

Precautions for Correct Use

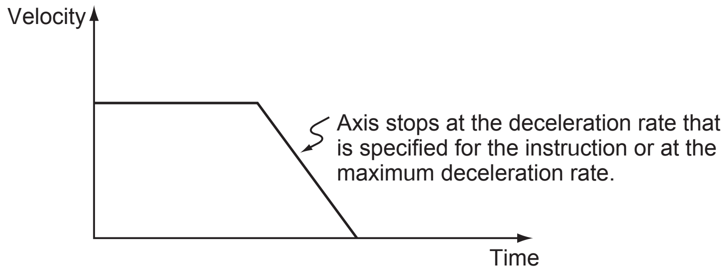
- If an error that results in deceleration to a stop occurs during execution of a synchronous movement instruction that has a *Deceleration* input variable, the axis decelerates to a stop at the deceleration rate given by *Deceleration*.
- If an error that results in deceleration to a stop occurs during execution of a synchronous movement instruction that does not have a *Deceleration* input variable, the axis decelerates to a stop at the maximum deceleration rate that is set in the axis parameters.

Additional Information

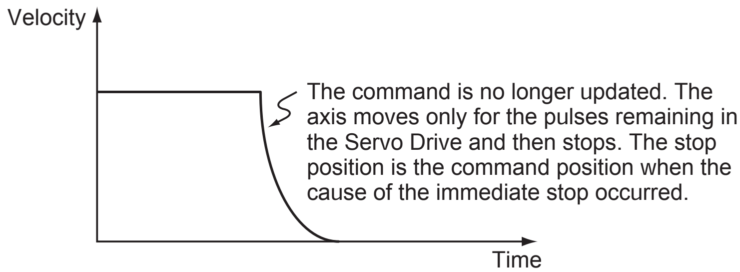
- When RUN mode changes to PROGRAM mode, any motion control instructions for current motions are aborted. The *CommandAborted* output variable from the instructions remains FALSE. The Servo remains ON even after changing to PROGRAM mode.
- If the operating mode returns to RUN mode while a deceleration stop is in progress after the operating mode changes from RUN to PROGRAM mode, the output variables from motion control instructions are cleared. The *CommandAborted* output variables from the motion control instructions therefore remain FALSE.
- The save process will continue during a save for the MC_SaveCamTable Instruction.
- The generation process will continue when generation of the cam table is in progress for the MC_GenerateCamTable (Generate Cam Table) instruction.

Stop Method

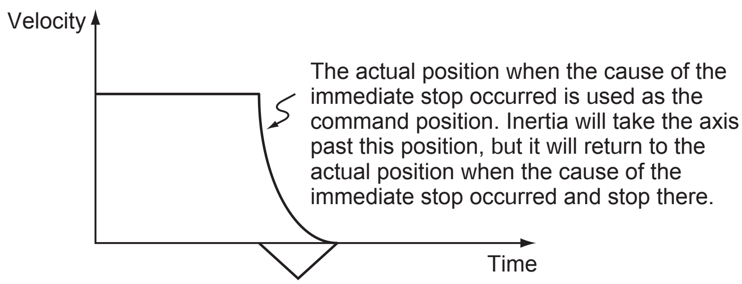
- Deceleration Stop



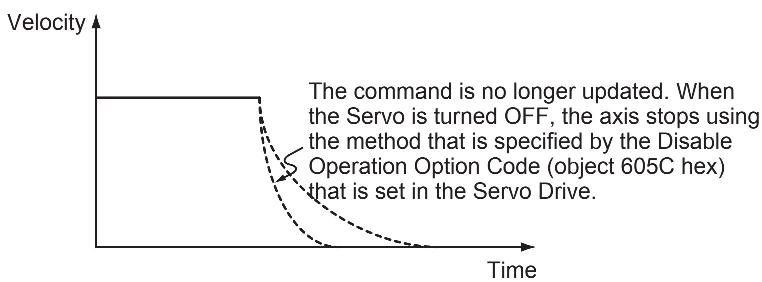
- Immediate Stop



- Immediate Stop and Error Reset



- Immediate Stop and Servo OFF



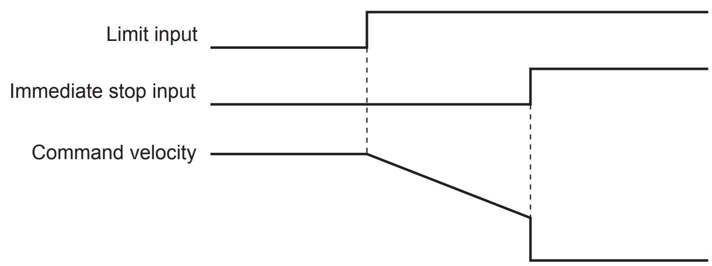
Stop Priorities

The priorities for each stop method are listed in the following table. If a stop with a higher priority stop method occurs while stopping, the stop method will switch to the higher priority method.

Stop method	Priority (higher numbers mean higher priority)
Immediate stop and Servo OFF	4
Immediate stop and error reset	3
Immediate stop	2
Deceleration stop	1

Example:

The following figure is an example of an immediate stop when the limit input signal is ON and the immediate stop input changes to ON during a deceleration to a stop.

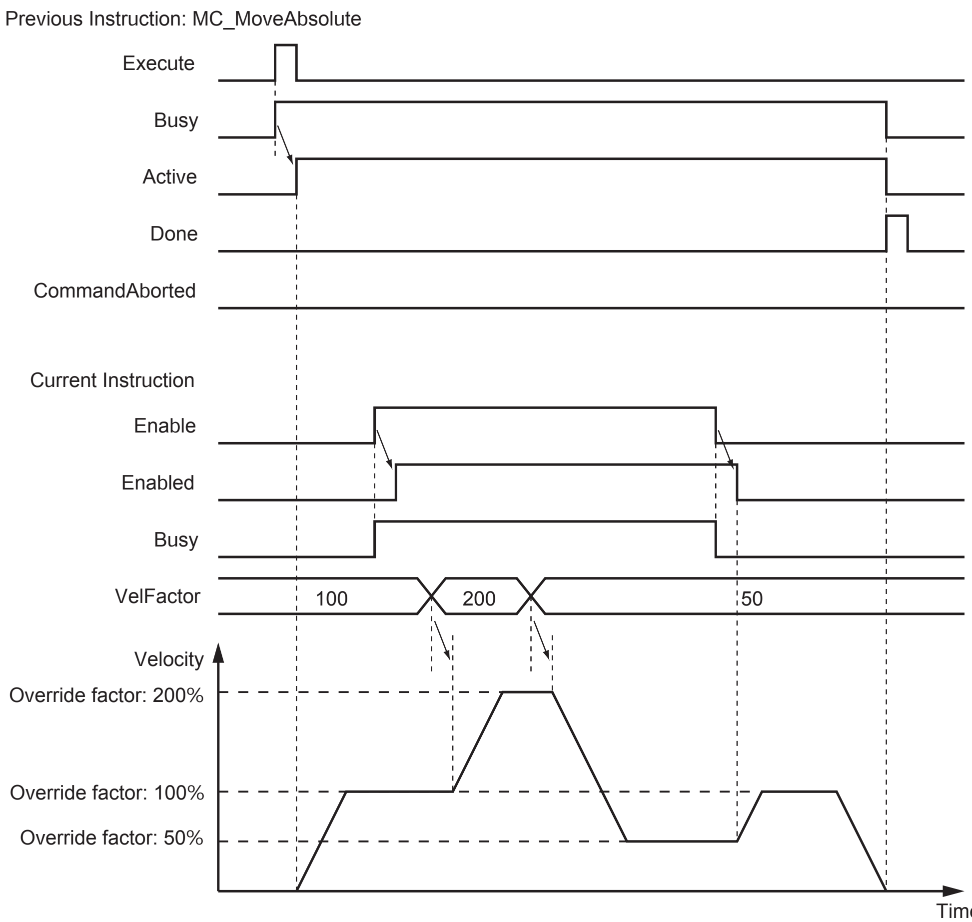


9-1-7 Override Factors

You can use the MC_SetOverride instruction to set override factors for the motion of the axes that are currently in motion. The velocity override factor is set as a percentage of the target velocity. It can be set between 0% and 500%. If an override factor of 0% is set for the target velocity, operating status will continue with the axis stopped as a velocity of 0. The set override factor is read as long as the overrides are enabled. If the overrides are disabled, the override factors return to 100%. If the maximum velocity is exceeded when an override factor is changed, the maximum velocity for the axis is used.

● Overriding the MC_MoveAbsolute Instruction

An example of a time chart for using the Set Override Factors instruction for the MC_MoveAbsolute (Absolute Positioning) instruction is given below.



For details, refer to the MC_SetOverride (Set Override Factors) instruction in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

9-2 Single-axis Synchronized Control

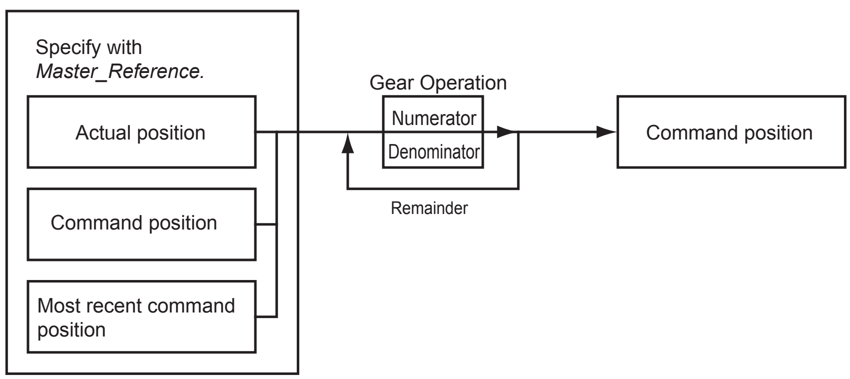
This section describes the operation of synchronized control for single axes.

9-2-1 Overview of Synchronized Control

Synchronous control synchronizes the position of a slave axis with the position of a master axis. The command position or actual position of any axis can be specified for the master axis. If the command velocity for the slave axis exceeds the maximum velocity that is set in the axis parameters, the command is performed at the maximum velocity of the axis. If this occurs, any insufficient travel distance is distributed and output in the following periods.

9-2-2 Gear Operation

This function specifies the gear ratio between the master axis and the slave axis and starts operation. Start gear operation with the MC_GearIn (Start Gear Operation) instruction. End synchronization with the MC_GearOut (End Gear Operation) instruction or the MC_Stop instruction.



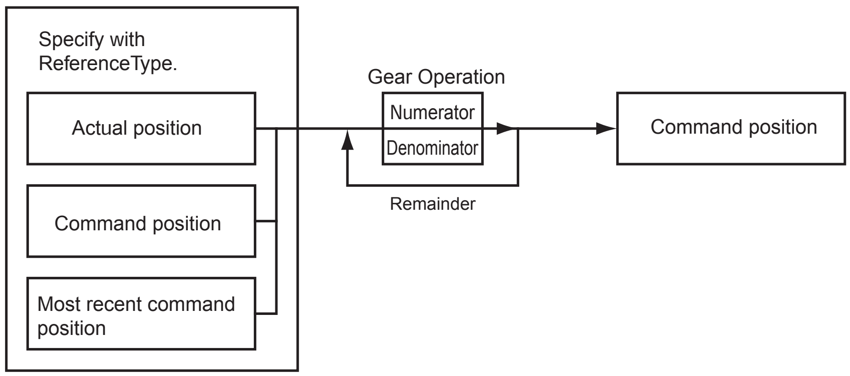
You can set the gear ratio numerator, gear ratio denominator, position type, acceleration rate, and deceleration rate for the slave axis to operate. For the master axis, you can specify the command position, actual position, or most recent command position.

After operation starts, the slave axis uses the velocity of the master axis times the gear ratio for its target velocity, and accelerates/decelerates accordingly. The catching phase exists until the target velocity is reached. The *InGear* phase exists after that. If the gear ratio is positive, the slave axis and master axis move in the same direction. If the gear ratio is negative, the slave axis and master axis move in the opposite directions.

For details on gear operation, refer to the MC_GearIn (Start Gear Operation), MC_GearOut (End Gear Operation), and MC_Stop instructions in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

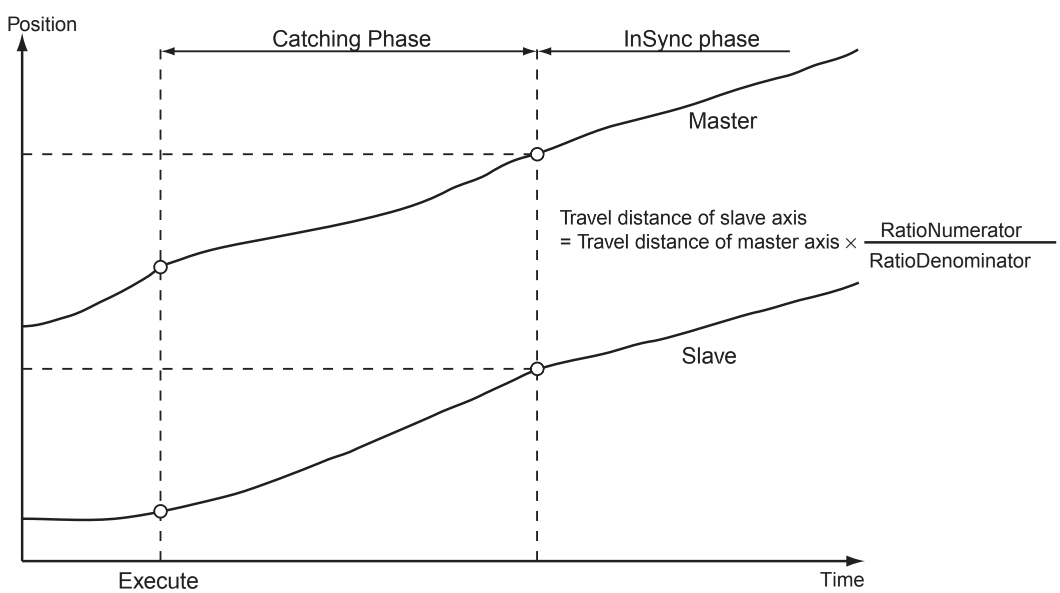
9-2-3 Positioning Gear Operation

This function specifies the gear ratio between the master axis and the slave axis and starts operation. Positioning gear operation allows you to set the positions of the master and slave axes at which to start synchronization. Start positioning gear operation with the MC_GearInPos instruction. End synchronization with the MC_GearOut instruction or the MC_Stop instruction.



You can set the gear ratio numerator, gear ratio denominator, position type, acceleration rate, and deceleration rate for the slave axis to operate. For the master axis, you can specify the command position, actual position, or most recent command position.

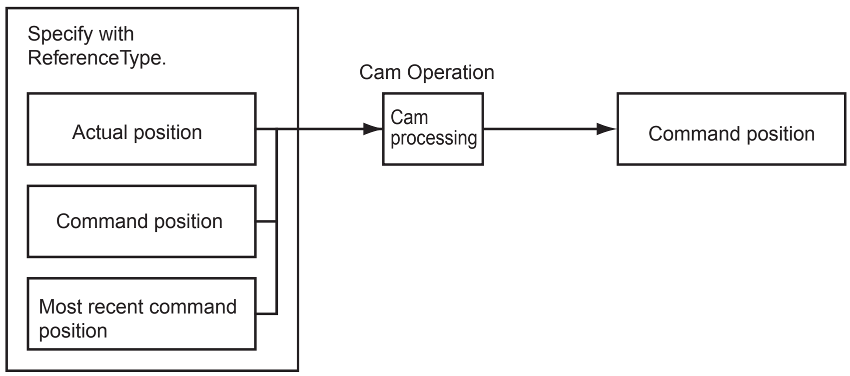
After operation starts, the slave axis uses the velocity of the master axis times the gear ratio for its target velocity, and accelerates/decelerates accordingly. The slave axis is in the catching phase until it reaches the slave axis sync start position. The slave axis enters the *InGear* phase after it reaches the slave sync start position. For either, the position of the slave axis is synchronized with the master axis. If the gear ratio is positive, the slave axis and master axis move in the same direction. If the gear ratio is negative, the slave axis and master axis move in the opposite directions. The following figure shows the operation when the gear ratio is positive.

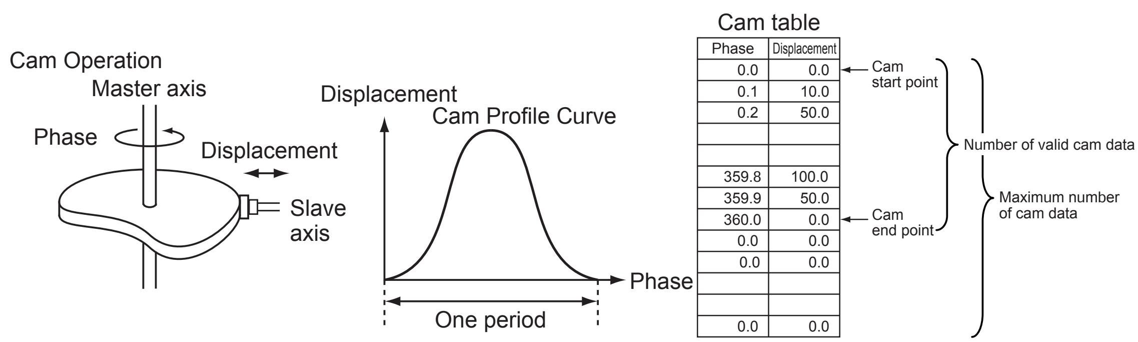


For details on positioning gear operation, refer to the MC_GearInPos (Positioning Gear Operation), the MC_GearOut (End Gear Operation), and the MC_Stop instructions in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

9-2-4 Cam Operation

Cam operation synchronizes the position of the slave axis with the master axis according to a cam table. Start cam operation with the MC_CamIn (Start Cam Operation) instruction. End cam operation with the MC_CamOut (End Cam Operation) instruction or the MC_Stop instruction. Create a cam table using the Cam Editor in the Sysmac Studio and download it to the CPU Unit. Use the Synchronization menu command of the Sysmac Studio to download the project to the CPU Unit.





In a combination of a CPU Unit with unit version 1.06 or later and Sysmac Studio version 1.07 or higher, the following operation is possible: if another MC_CamIn (Start Cam Operation) instruction is executed by using multi-execution with the Buffer Mode set for blending while the current MC_CamIn (Start Cam Operation) instruction is executed, the operation can continue using the switched cam table and the slave axis does not stop.

For details on cam operation, refer to the MC_CamIn (Start Cam Operation), MC_CamOut (End Cam Operation), and MC_Stop instructions in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

For details on the Cam Editor, refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504).

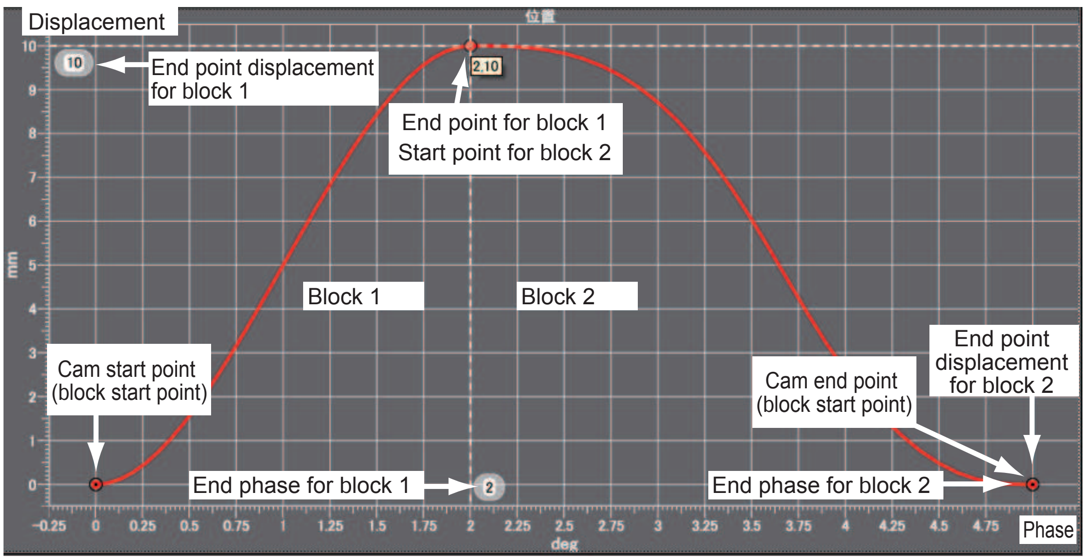
9-2-5 Cam Tables

This section describes the cam tables that are used for cam operation.

Cam Table Terminology

Term	Description
cam operation	An operation that takes one master axis and one slave axis and follows the cam profile curve to derive the displacement of the slave axis from the phase of the master axis.
cam profile curve	A curve that shows the relationship between phases and displacements in a cam operation. Cam profile curves are used in the Cam Editor. The cam profile curve is created on the Sysmac Studio. You can use the cam profile curve with a cam data variable after the cam profile curve is downloaded to the CPU Unit. Use the Synchronization menu command of the Sysmac Studio to download the project to the CPU Unit.
cam block	You can select a cam curve in this block. It represents the area between the end point of the previous cam block and the end point of the current cam block.
cam curve	A curve that represents the cam characteristics. You can select a cam curve for each cam block. The Sysmac Studio calculates the phase widths and displacement widths from the specified points and creates the actual cam profile curve. You can choose from different curves, such as straight line, parabolic, and trapecloid.
cam data	Data made up of phases (master axis) and displacements (slave axis) for cam operation.
cam data variable	A variable that represents the cam data as a structure array.
cam table	A data table that contains cam data. If phase data is not in ascending order the cam table is treated as an illegal cam table.
cam start point	The first point in the cam data.
cam end point	The last point of valid cam data in the cam data. If the cam end point is less than the number of cam data, all phases and displacements after the cam end point will be 0.
cam block start point	The start point for a cam block. It is the same as the cam start point at the start of the cam operation. If the cam profile curve continues, this will be the same as the cam block end point.
cam block end point	The end point for a cam block. It is the same as the cam end point at the end of the cam operation. If the cam profile curve continues, this will be the same as the cam block start point. The cam block end point is defined as (horizontal axis, vertical axis) = (phase end point, displacement end point).
original cam data	Cam data that is created by dividing up the cam profile curve in the Cam Editor.
program-modified cam data	The cam data changed by the user program while the CPU Unit is in operation.
master axis	The axis that serves as the input to the cam operation. You can specify either Linear Mode or Rotary Mode.
slave axis	The axis that serves as the output from the cam operation. You can specify either Linear Mode or Rotary Mode.
phase	The relative distance on the master axis from the start point of the cam table.
displacement	The relative distance on the slave axis from the master following distance.
valid cam data	The cam data other than the cam start point and other than data where the phase is 0.
invalid cam data	The cam data other than the cam start point where the phase is 0.
number of valid cam data	The number of sets of cam data.
maximum number of cam data	The maximum number of sets of cam data that the cam table can contain.
cam data index	The number of the cam data that is executed.

Term	Description
cam table start position	The absolute position of the master axis that corresponds to the cam start point (phase = 0).
master following distance	The master start distance where the slave axis starts cam operation represented as either an absolute position or relative position. The relative position is based on the cam start point position.
start mode	A specification of whether to represent the master following distance as an absolute position or relative position.
null cam data	Cam data that can be set after the end point where the phase and displacement are 0.
connecting velocity	The connecting velocity that is used to connect cam profile curves. The connecting velocity cannot be specified for some curves.
connecting acceleration	The acceleration rate that is used to connect cam profile curves. The connecting acceleration cannot be specified for some curves.
phase pitch	The width when dividing the cam profile curve by phases (horizontal axis). The points after dividing the curve into the phase pitch correspond to the cam data in the cam table. You must specify the phase pitch for each cam block.



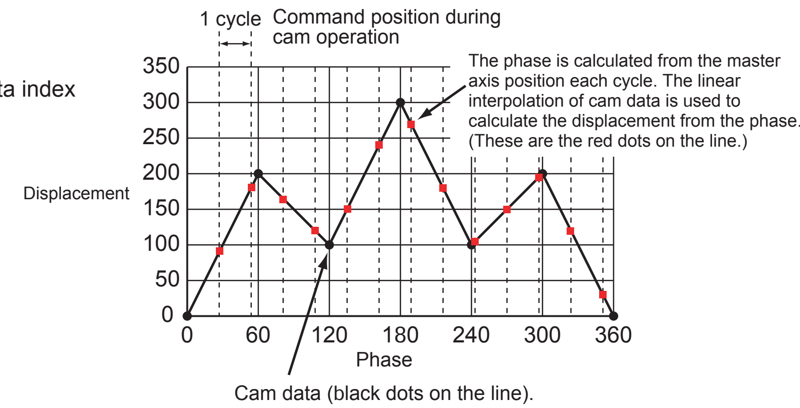
Cam Tables

The MC Function Module defines a single element of data consisting of the phase of the master axis and the displacement of the slave axis as one cam data. A cam table is defined as the combination of multiple sets of cam data. The cam table is created with the Cam Editor in the Sysmac Studio. You can modify cam data in the cam table from the user program.

The phases and displacements in the cam data that makes up the cam table are represented as relative distances from the start point 0.0. During cam operation, the command position sent to the slave axis is the displacement determined by interpolating linearly between the two cam data elements adjacent to the phase of the master axis. The more cam data there is in the cam table, the more accurate the trajectory and the smoother the cam profile curve will be.

Cam table

	Phase	Displacement	Cam data index
Cam start point →	0	0	0
Cam data →	60	200	1
	120	100	2
	180	300	3
	240	100	4
	300	200	5
Cam end point →	360	0	6



Precautions for Correct Use

- Make sure that the cam data is arranged in the cam table so that the phases are in ascending order. An instruction error occurs if a cam operation instruction is executed when the phases are not in ascending order.
- Cam data variables are global variables. You can therefore access or change the values of cam data variables from more than one task. If you change the values of cam data variables from more than one task, program the changes so that there is no competition in writing the value from more than one task.
- If you use exclusive control of global variables between tasks for a cam data variable, do not use the cam data variable for motion control instructions in a task that does not control the variable. An Incorrect Cam Table Specification error (error code: 5439 hex) will occur.

Cam Table Specifications

Item	Description
Maximum number of cam data per cam table	65,535 points
Maximum size of all cam data	1,048,560 points[*1]
Maximum number of cam tables	640 tables[*2]
Switching cam operation	You can switch to a different cam operation by executing a motion control instruction
Changing cam data	Cam data can be edited from the user program. Cam data can be overwritten with the Generate Cam Table instruction.[*3]
Saving cam data	Cam data can be saved to non-volatile memory by using the Save Cam Table instruction.
Information attached to the cam data	Information can be downloaded or uploaded for display in the Cam Editor[*4]
Timing to load cam data to main memory	• When the data is downloaded from the Sysmac Studio • When power is turned ON

*1 If 65,535 points are used for each cam table, there will be a maximum of 16 cams. A resolution of 0.1° allows for a maximum of 3,600 points per cam table for a maximum of 291 cams.

*2 The total size is 10 MB max.

*3 A CPU Unit with unit version 1.08 or later and Sysmac Studio version 1.09 or higher are required to use the Generate Cam Table instruction.

*4 Use the Synchronization menu command of the Sysmac Studio to upload and download the project.

Data Type of Cam Tables

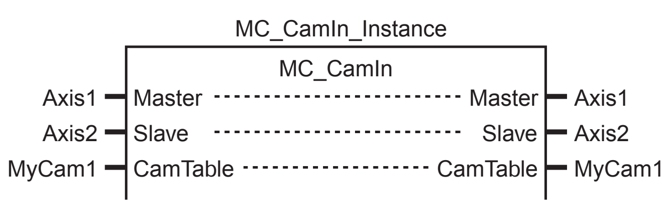
A cam table is declared as an array of cam data structures. The type declaration for the cam data structure is shown below.

```
TYPE
        (*Cam data structure*)
        _sMC_CAM_REF :
        STRUCT
                Phase       : REAL;         (*Phase*)
                Distance    : REAL;         (*Displacement*)
        END_STRUCT;
END_TYPE
```

You must create the cam data with the Cam Editor in the Sysmac Studio and then specify the name of the cam table and the number of cam data (i.e., the size of the array). For example, to make a cam table called *MyCam1* with 1,000 points use the following declaration.

```
VAR
        (*Cam table*)
        MyCam1          :       ARRAY [0..999] OF _sMC_CAM_REF;
END_VAR
```

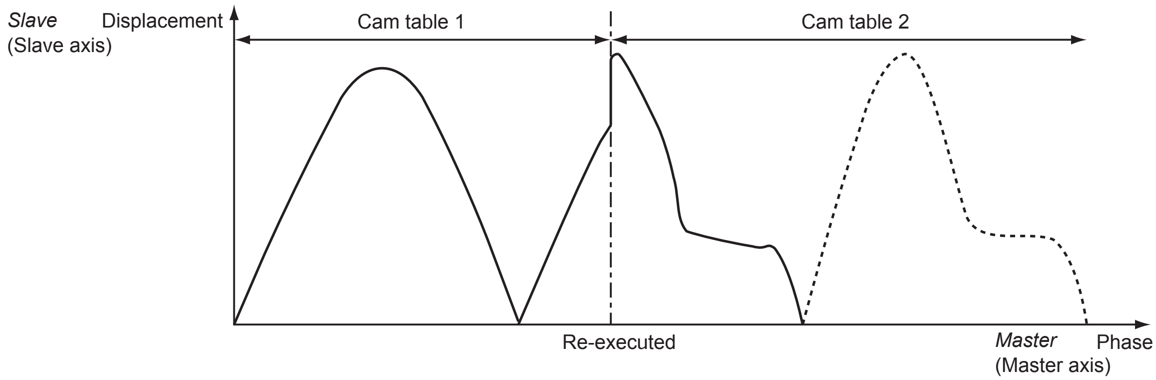
The following notation is used to specify *MyCam1* for a cam operation instruction. In this example, the master axis is *Axis1* and the slave axis is *Axis2*.



An error will occur if the specified cam table does not exist in the Controller. You can also specify the same cam table for more than one axis.

Switching Cam Tables

You can switch cam tables by re-executing the cam operation instruction during cam operation. After switching, cam operation will be performed with the cam table you specified for re-execution of the instruction. The *EndOfProfile* and *Index* output variables from the MC_CamIn instruction are output according to the new cam table.



Precautions for Correct Use

- The cam table you want to switch to must be saved to non-volatile memory before it can be used.
- Switching cam tables during cam operation will cause discontinuous velocities. Adjust the timing for switching the cam table to avoid excessive velocity discontinuity.

Loading/Saving Cam Data and Saving Cam Tables

Cam data can be loaded and saved from the user program just like any other variables. For example, you can use *MyCam1[0].Phase* to specify the phase and *MyCam1[0].Distance* to specify the displacement in the first array elements of a cam table named MyCam1. Cam data overwritten from the user program can be saved to the non-volatile memory in the CPU Unit as a cam table by executing the MC_SaveCamTable instruction.

Precautions for Correct Use

- Overwritten cam data will be lost if the CPU Unit is turned OFF or the cam data is downloaded from the Sysmac Studio before the Save Cam Table instruction is executed or if the instruction fails to save the data for any reason.
- Overwritten cam data will be lost if the CPU Unit is turned OFF before the Save Cam Table instruction is executed or if the instruction fails to save the data for any reason. Be careful not to lose the overwritten data when overwriting cam data from the user program in the CPU Unit.
- Cam data saved to non-volatile memory can be loaded by using the upload function of the Sysmac Studio.
- Use the Synchronization menu command of the Sysmac Studio to upload and download the project.

For details on arrays, refer to the *NJ-series CPU Unit Software User's Manual* (Cat. No. W501).

For details on the Save Cam Table instruction, refer to the MC_SaveCamTable instruction in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

Updating Cam Table Properties

The MC Function Module must identify the cam end point of the cam table. If an overwrite is performed from the user program during cam operation and the number of valid cam data changes, you must update the number of valid cam data to the latest value. Use the MC_SetCamTableProperty instruction for this.

The cam end point is the data located one cam data before the first cam data with a phase of 0 after the start point in the cam table. All cam data after phase 0 is detected will be invalid.

For example, refer to the following cam table. The *EndPointIndex* (End Point Index) output variable is 999 and the *MaxDataNumber* (Maximum Number of Cam Data) output variable is 5,000 from the MC_SetCamTableProperty instruction.

Cam data structure array	Phase	Displacement
MyCam1 [0]	0	0
.	.	.
.	.	.
.	.	.
MyCam1 [997]	359.8	2
MyCam1 [998]	359.9	1
MyCam1 [999]	360.0	0
MyCam1 [1000]	0	0
.	.	.
.	.	.
.	.	.
MyCam1 [4999]	0	0

Cam start point

Valid data

Cam end point

Invalid data

Maximum number of data: 5,000

Precautions for Correct Use

- You cannot change the maximum number of cam data from the user program.
- Execute this instruction after overwriting the cam data in any way that changes the number of valid cam data. If the number of valid cam data is not updated, the cam operation and the operation of the *EndOfProfile* (End of Cam Cycle) of the MC_CamIn instruction may not be as expected.

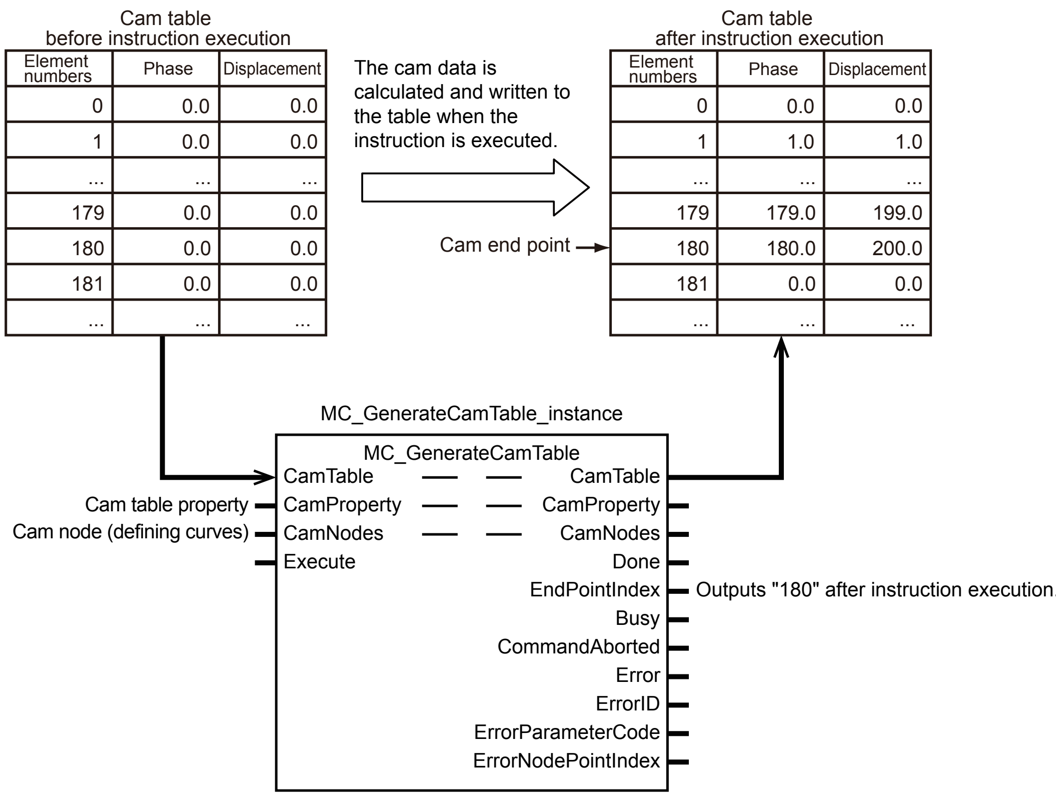
For details on the Set Cam Table Properties instruction, refer to the MC_SetCamTableProperty (Set Cam Table Properties) instruction in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

Generate Cam Table

With a CPU Unit with unit version of 1.08 or later and the Sysmac Studio version 1.09 or higher, you can generate the cam table by executing the MC_GenerateCamTable (Generate Cam Table) instruction.

The MC_GenerateCamTable instruction calculates the cam data using the values specified for *CamProperty* (Cam Properties) and *CamNodes* (Cam Nodes), and rewrites the cam data variable specified for the *CamTable* (Cam Table) in-out variable.

When rewriting is completed, the MC_GenerateCamTable instruction updates the end point index of the cam table and outputs the element number of the cam end point to *EndPointIndex* (End Point Index).
It is not necessary to execute the MC_SetCamTableProperty (Set Cam Table Properties) instruction after the MC_GenerateCamTable instruction is completed.



The cam data variable is an array variable with the data type of cam data structure _sMC_CAM_REF. You create the cam data variable on the Cam Editor of the Sysmac Studio.
For *CamProperty*, specify the cam property variable. The cam property variable is an array variable with the data type of cam property structure _sMC_CAM_PROPERTY. You create the cam property variable as a user-defined variable on the global variable table of the Sysmac Studio. Or, you create the variable using the cam data settings on the Sysmac Studio.
For *CamNode*s, specify the cam node variable. The cam node variable is an array variable with the data type of cam node structure _sMC_CAM_NODE. You create the cam node variable as a user-defined variable on the global variable table of the Sysmac Studio. Or, you create the variable using the cam data settings on the Sysmac Studio.
The cam property variable and the cam node variable are collectively called “cam definition variable”.

If the cam definition variable is created as a user-defined variable, the default of its Retain attribute is Non-retain. You must set the Retain attribute of variable to Retain, if you want to reuse the variable after changing its value and switching the operating mode to PROGRAM mode or cycling the power supply. If you set the variable each time of use from the PT, etc., the attribute can be left Non-retain.
If the cam definition variable is created with the cam data settings on the Sysmac Studio, the Retain attribute of variable will be fixed to Retain.

By using the PT, etc. to set the values for the MC_GenerateCamTable instruction, you can create the cam data variable and adjust the cam operation without using the Sysmac Studio.
The following is the procedure used to adjust the cam operation.

1. Create a user program, in advance, that includes the following processing.
   - Assigning the value of the cam definition variable that is set from the PT to the Generate Cam Table instruction.
   - Displaying the cam variable that is created by the Generate Cam Table instruction graphically on the PT.
   - Displaying the value of *EndPointIndex* (End Point Index) on the PT.
2. Set the value of the cam definition variable from the PT.
3. Execute the Generate Cam Table instruction.
4. Verify the curve shape of the generated cam table and the value of the end point index displayed on the PT.
5. If there is no problem with the curve shape of the cam table and the number of the cam data, then execute the cam operation.
6. Verify the result of the cam operation and consider changing the value of the cam definition variable.
7. Repeat steps 2 to 6.

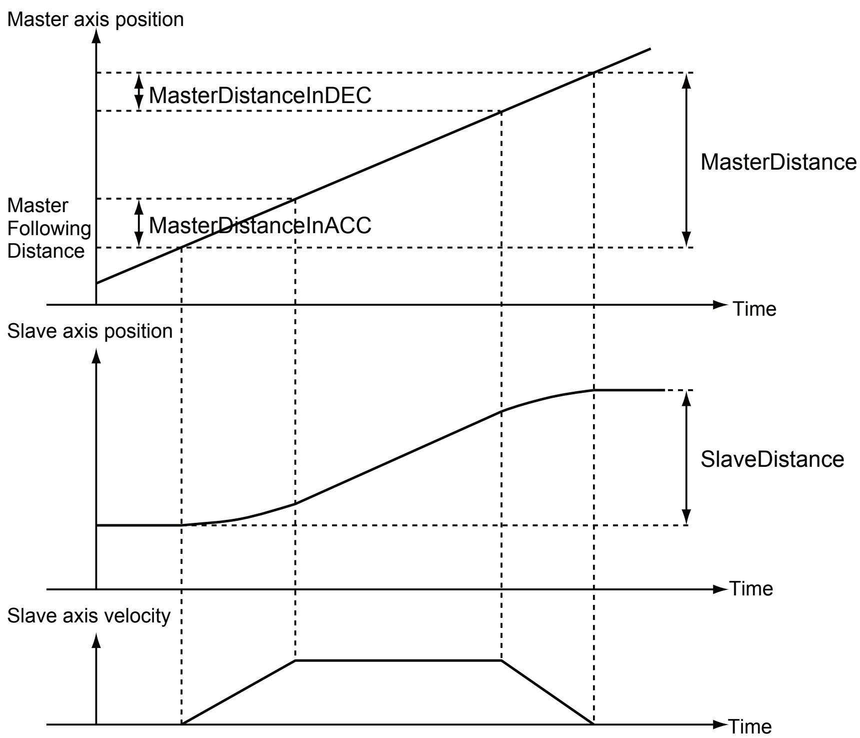
For details on the cam definition variable and the Generate Cam Table instruction, refer to the MC_GenerateCamTable instruction in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508-E1-08 or later).

Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504-E1-10 or higher) for information on creating and transferring the cam definition variables using the Sysmac Studio.

9-2-6 Synchronous Positioning

This function performs positioning using a trapezoidal curve while synchronizing the specified slave axis to the specified master axis. This is a type of electronic cam, but it does not use cam tables created in the Cam Editor. Operation starts when the MC_MoveLink (Synchronous Positioning) instruction is executed. Use the MC_Stop instruction to stop the axes in motion. Operation is performed for the *Slave* (Slave Axis) and the following are set: *Master* (Master Axis), *MasterDistance* (Master Axis Travel Distance), *MasterDistanceInACC* (Master Distance In Acceleration), *MasterDistanceInDEC* (Master Distance In Deceleration), *SlaveDistance* (Slave Axis Travel Distance), and *MasterStartDistance* (Master Following Distance). The command position or actual position can be specified for the master axis. You can specify one of the following as the start condition for synchronous operation: start of instruction, when trigger is detected, or when master axis reaches the master following distance.

The velocity and position of the slave axis are determined by the ratio of the travel distances of the master axis and the slave axis as shown in the following figure. The sync start position shown in the following figure represents the position where the sync start condition is met.

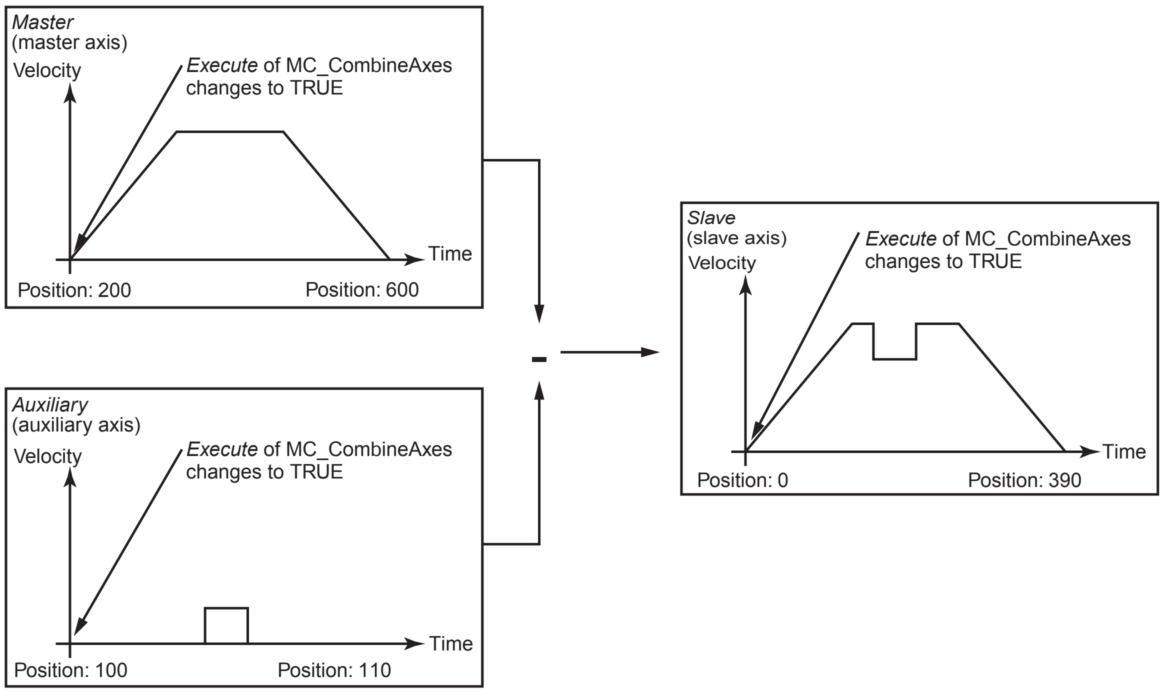


For details on synchronous positioning, refer to the MC_MoveLink (Synchronous Positioning) and MC_Stop instructions in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

9-2-7 Combining Axes

The sum or difference of two positions can be used as the command position for the slave axis. Operation starts when the MC_CombineAxes instruction is executed. Use the MC_Stop instruction to stop axes in motion.

The following figure is an example demonstrating operation when subtracting axes. *Slave* (Slave Axis) command current position = *Master* (Master Axis) command current position – *Auxiliary* (Auxiliary Axis) command current position)

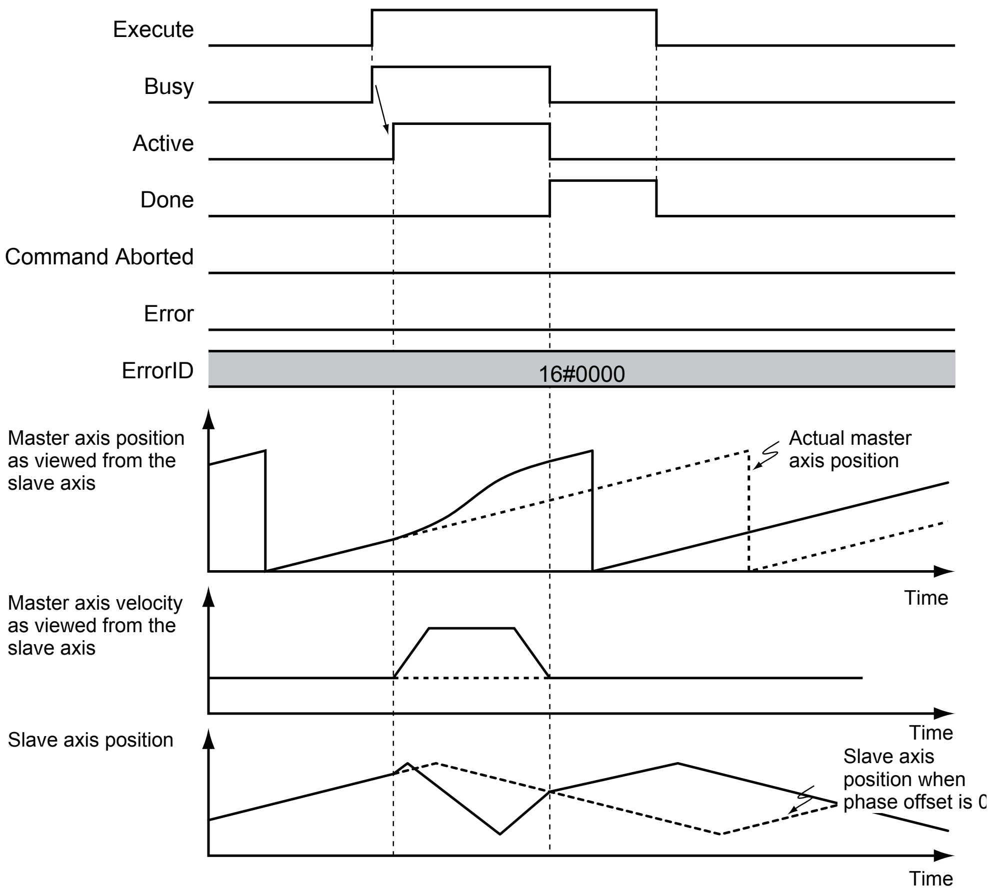


For details on combining axes, refer to the MC_CombineAxes and MC_Stop instructions in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

9-2-8 Master Axis Phase Shift

The phase of the master axis as viewed from the slave axis can be shifted for the current instruction. The shift amount as viewed from the slave axis is a relative amount. During synchronization, the slave axis will synchronize to the relative distance of the master axis. You can execute the MC_Phasing (Shift Master Axis Phase) instruction to shift the phase for a synchronized control instruction.

You can specify the phase shift amount, target velocity, acceleration rate, deceleration rate, and jerk for the MC_Phasing (Shift Master Axis Phase) instruction.



For details on the shift master axis phase function and the synchronized control instructions for which a master axis phase shift can be applied, refer to the MC_Phasing (Shift Master Axis Phase) instruction in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

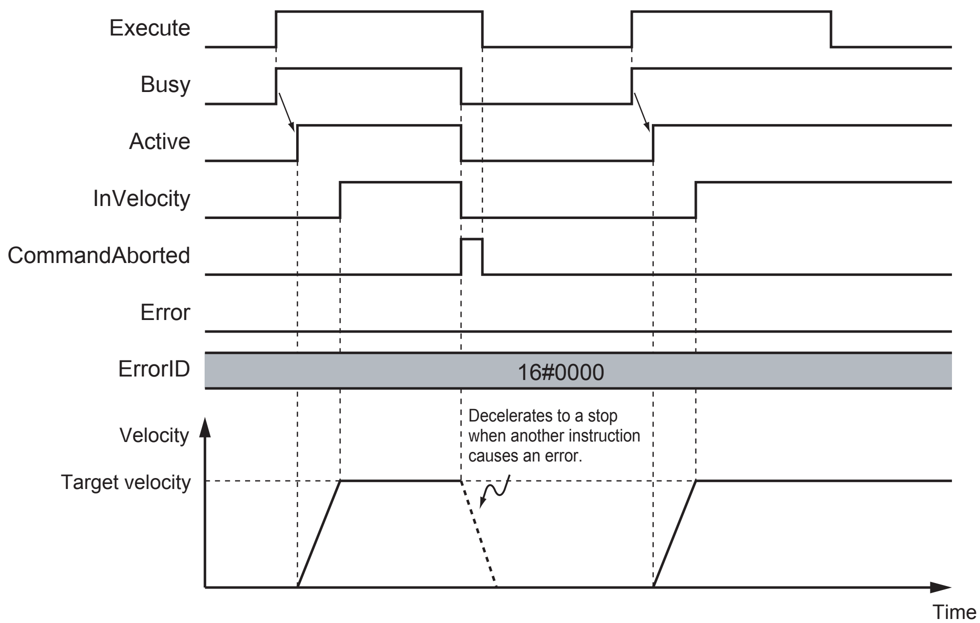
9-3 Single-axis Velocity Control

This section describes the operation of velocity control for single axes.

Refer to the *NX-series Position Interface Units User's Manual* (Cat. No. W524) for the differences when you use NX-series Pulse Output Units.

9-3-1 Velocity Control

Velocity control is used to constantly move an axis at the specified velocity. You can also specify the acceleration rate, deceleration rate, and jerk. To stop an axis, use the MC_Stop instruction or execute another motion instruction. If you specify a target velocity of 0, the axis will not move but the axis status will indicate that it is moving. If any other motion control instruction is executed with multi-execution of instructions during velocity control, the operation will switch only after reaching the target velocity.

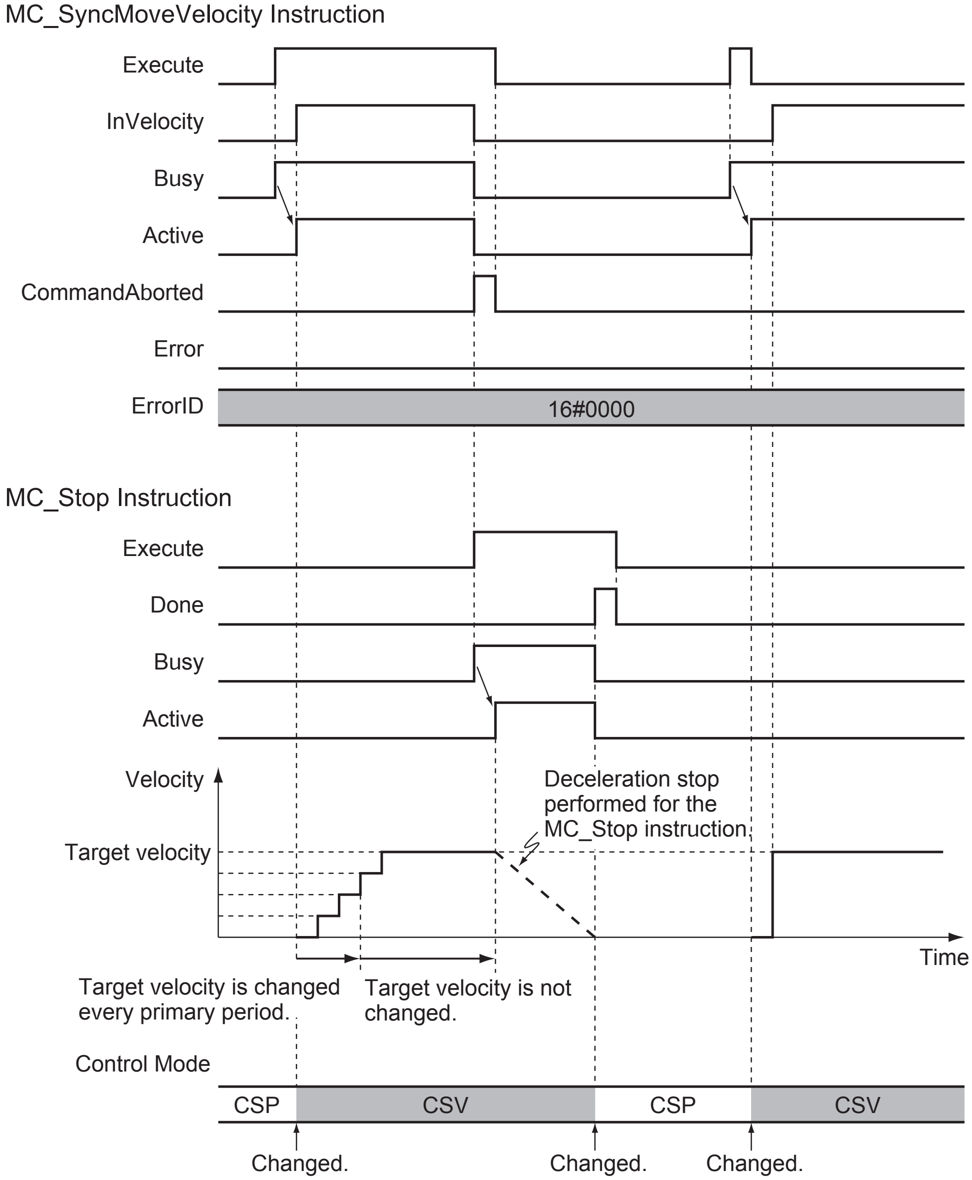


The MC Function Module uses Position Control Mode of the Servo Drive or other device and sends target position commands to achieve the specified target velocity.
The position control loop is enabled in the Servo Drive or other device. Therefore, as the command velocity slows down, e.g., due to disturbance, and the following error increases, the velocity will change to eliminate this following error.

For details, refer to the MC_MoveVelocity (Velocity Control) instruction in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

9-3-2 Cyclic Synchronous Velocity Control

The control mode of the Servo Drive is set to Velocity Control Mode and a command speed is output every control period. To stop an axis, use the MC_Stop instruction or execute another motion control instruction. If you specify a target velocity of 0, the axis will not move but the axis status will indicate that it is moving.



The Servo Drive will receive commands in the velocity control loop. Therefore, if any disturbance causes the velocity to decrease below the command velocity, no change in velocity will occur to remove the following error.

For details, refer to the MC_SyncMoveVelocity (Cyclic Synchronous Velocity Control) instruction in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

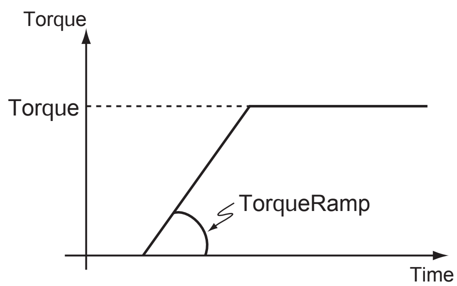
Precautions for Correct Use

You cannot use cyclic synchronous velocity control for an NX-series Pulse Output Unit.

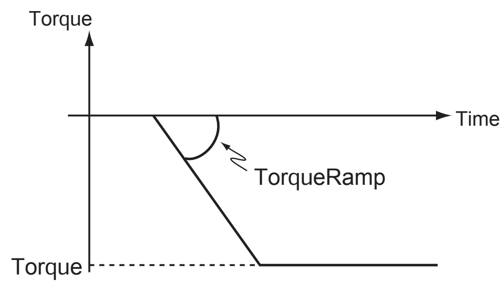
9-4 Single-axis Torque Control

Torque control continuously applies the specified amount of torque. You can use *TorqueRamp* to specify the rate of change of the torque until the *Torque* (Target Torque) is reached. To stop an axis, use the MC_Stop instruction or execute another motion instruction. If you specify a *Torque* (Target Torque) of 0, the axis will not move but the axis status will indicate that it is moving.

Example 1: Direction Designation = Positive Direction



Example 2: Direction Designation = Negative Direction



The MC Function Module uses the Torque Control Mode of the Servo Drive. The Servo Drive receives the torque command value from the MC Function Module in the torque control loop and to control the torque. You can specify the velocity limit value for the Servo Drive in the *Velocity* (Velocity Limit) input variable to the motion control instruction. You can use this to limit high-speed revolution of the motor when the load on the motor is low in Torque Control Mode.

Precautions for Correct Use

- To be safe, always set a velocity limit value for torque control.
- You cannot use single-axis torque control for an NX-series Pulse Output Unit.

For details, refer to the MC_TorqueControl instruction in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

9-5 Common Functions for Single-axis Control

This section describes the common functions used for single-axis control.

Refer to the *NX-series Position Interface Units User's Manual* (Cat. No. W524) for the differences when you use NX-series Pulse Output Units.

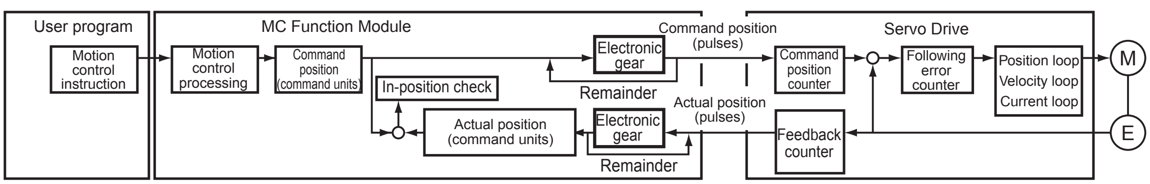
9-5-1 Positions

Types of Positions

The MC Function Module uses the following two types of positions.

Type of position	Definition
Command position	This is the position that the MC Function Module outputs to control an axis.
Actual position	The actual position as input from the Servo Drive or encoder input.

The following figure shows the relationship between the command position and the actual position for an EtherCAT slave Servo Drive.



The command position and actual position share the following items.

Item	Command position	Actual position
Count Mode	You can set Linear Mode or Rotary Mode.	The same Count Mode is used as for the command position.
Position increment	You can set one of the following: mm, μm, nm, inch, degree, or pulse.	The unit is the same as the unit of the command position.
Software limits	You can set the range of operation of the software.	The range is the same as the range for the command position.
Changing the current position	You can change the actual position to any desired position.	This value will be set to the same position as the command position.*
Defining home	Home is either defined or undefined.	The status of home is the same as the command position.

* If there is any following error before the change, the following error value is maintained in the actual position.

Additional Information

Refer to the *NX-series Position Interface Units User's Manual* (Cat. No. W524) for information on the NX-series Position Interface Units.

Axis Parameters That Are Related to Positions

Parameter name	Function	Setting range	Default
In-position Range	Set the in-position width. (Unit: command units)	Non-negative long reals	10
In-position Check Time	Set the in-position check time in milliseconds. Set 0 to check for the end of positioning only when you define the home position during homing and not check positioning at other times. (Unit: ms)	0 to 10,000	0
Software Limits	Select the software limit function. 0: Disabled. 1: Deceleration stop for command position 2: Immediate stop for command position 3: Deceleration stop for actual position 4: Immediate stop for actual position	0 to 4	0
Positive Software Limit	Set the software limit in the positive direction. (Unit: command units)	Long reals	2,147,483,647
Negative Software Limit	Set the software limit in the negative direction. (Unit: command units)	Long reals	–2,147,483,648
Following Error Over Value	Set the excessive following error check value. Set 0 to disable the excessive following error check. (Unit: command units)	Non-negative long reals	0
Following Error Warning Value	Set the following error warning check value. Set 0 to disable the following error warning check. (Unit: command units)	Non-negative long reals that are less than or equal to the Following Error Over Value	0

Specifying Target Positions for Axis Operations

The actual position or distance for a positioning motion is specified with the *Position* (Target Position) and *Distance* (Travel Distance) input variables to the motion control instruction.

Monitoring Positions

You can read Axis Variables in the user program to monitor positions.

Variable name	Data type	Meaning	Function
_MC_AX[0-63].Cmd.Pos	LREAL	Command Current Position	This is the current value of the command position. When the Servo is OFF and the mode is not the position control mode, the actual current position is output.
_MC_AX[0-63].Act.Pos	LREAL	Actual Current Position	This is the actual current position.

9-5-2 Velocity

Types of Velocities

The following two types of axis velocities are used in the MC Function Module.

Velocity type	Definition
Command velocity	This is the velocity that the MC Function Module outputs to control an axis.
Actual velocity	This is the velocity calculated in the MC Function Module based on the actual position input from the Servo Drive or encoder input.*

* This value is given if the Velocity actual value (606C hex) is mapped in the PDOs and assigned to the Actual Current Velocity.

Velocity Unit

A velocity is given in command units/s. The command unit is the value obtained from unit conversion of the position display unit and the electronic gear.

Axis Parameters That Are Related to Velocities

Parameter name	Function	Setting range	Default
Maximum Velocity	Specify the maximum velocity for the axis. If a target velocity that exceeds the maximum velocity is specified for an axis motion instruction, the axis will move at the maximum velocity.	Positive long reals	400,000,000
Start Velocity*1	Set the start velocity for each axis. Set a value that does not exceed the maximum velocity. (Unit: command units/s)	Positive long reals	0
Maximum Jog Velocity	Set the maximum jog velocity for each axis.*2 Set a value that does not exceed the maximum velocity. (Unit: command units/s)	Positive long reals	1,000,000
Velocity Warning Value	Set the percentage of the maximum velocity at which to output a velocity warning for the axis. No velocity warning is output if 0 is set. (Unit: %)	0 to 100	0
Actual Velocity Filter Time Constant	Set the time period to calculate the average travel of the actual velocity in milliseconds. The average travel is not calculated if 0 is set. (Unit: ms) Use this to reduce variations in the actual current velocity when axis velocity is slow.	0 to 100	0

*1 A CPU Unit with unit version 1.05 or later and Sysmac Studio version 1.06 or higher are required to use this parameter.

*2 The maximum jog velocity is used as the command velocity if you specify a velocity command value that is greater than the maximum jog velocity.

Specifying Target Velocities for Axis Operations

The velocity used in an actual positioning motion is specified by the *Velocity* (Target Velocity) input variable to the motion control instruction.

Monitoring Velocities

You can read Axis Variables in the user program to monitor velocities.

Variable name	Data type	Meaning	Function
_MC_AX[0-63].Cmd.Vel	LREAL	Command Current Velocity	This is the current value of the command velocity. A plus sign is added during travel in the positive direction, and a minus sign is added during travel in the negative direction.
_MC_AX[0-63].Act.Vel	LREAL	Actual Current Velocity	This is the actual current velocity. A plus sign is added during travel in the positive direction, and a minus sign is added during travel in the negative direction.

9-5-3 Acceleration and Deceleration

Unit of Acceleration and Deceleration Rates

Acceleration rates and deceleration rates are given in command units/s^2. The command unit is the value obtained from unit conversion of the position display unit and the electronic gear.

Axis Parameters That Are Related to Acceleration and Deceleration

Parameter name	Function	Setting range	Default
Maximum Acceleration	Set the maximum acceleration rate for an axis operation command. There will be no limit to the acceleration rate if 0 is set. (Unit: command units/s^2)	Non-negative long reals	0
Maximum Deceleration	Set the maximum deceleration rate for an axis operation command. There will be no limit to the deceleration rate if 0 is set. (Unit: command units/s^2)	Non-negative long reals	0
Acceleration/Deceleration Over	Set the operation for when the maximum acceleration/deceleration rate would be exceeded after excessive acceleration/deceleration during acceleration/deceleration control of the axis because stopping at the target position is given priority. 0: Use rapid acceleration/deceleration. (Blending is changed to Buffered.) 1: Use rapid acceleration/deceleration. 2: Minor fault stop	0 to 2	0
Acceleration Warning Value	Set the percentage of the maximum acceleration rate at which to output an acceleration warning for the axis. No acceleration warning is output if 0 is set. (Unit: %)	0 to 100	0

Parameter name	Function	Setting range	Default
Deceleration Warning Value	Set the percentage of the maximum deceleration rate at which to output a deceleration warning for the axis. No deceleration warning is output if 0 is set. (Unit: %)	0 to 100	0

Specifying Acceleration and Deceleration Rates for Axis Operation

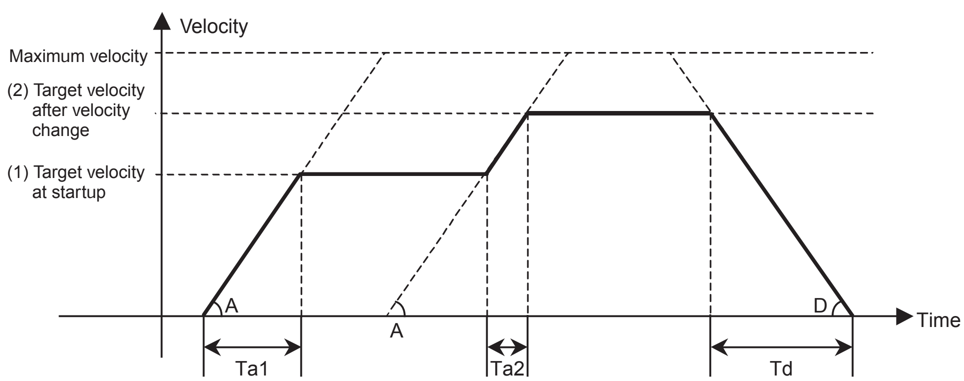
The acceleration and deceleration rates used in an actual positioning motions are specified by the *Acceleration* (Acceleration Rate) and *Deceleration* (Deceleration Rate) input variables to the motion control instruction.

Monitoring Acceleration and Deceleration Rates

You can read Axis Variables in the user program to monitor acceleration and deceleration rates.

Variable name	Data type	Meaning	Function
_MC_AX[0-63].Cmd.AccDec	LREAL	Command Current Acceleration/Deceleration	This is the current value of the command acceleration/deceleration rate. A plus sign is added for acceleration, and a minus sign is added for deceleration.

Example of Acceleration/Deceleration Operation



When Starting
Ta1: Actual acceleration time
A: Acceleration rate

For Velocity Changes
Ta2: Actual acceleration time
A: Acceleration rate

When Decelerating
Td: Actual deceleration time
D: Deceleration rate

If you specify a short travel distance or a low acceleration/deceleration rate, the target velocity may not be reached. If the target position is exceeded after re-execution of the motion control instruction with the newly updated acceleration or deceleration rate, positioning is performed at an acceleration or deceleration rate that will enable stopping at the target position.

9-5-4 Jerk

The jerk specifies the rate of change in the acceleration rate or deceleration rate. If the jerk is specified, the velocity waveform during acceleration will be an S-curve, which will reduce the shock and vibration on the machine.

Additional Information

Jerk is also called jolt, surge and lurch.

Jerk Unit

Jerk is given in command units/s^3. The command unit is the value obtained from unit conversion of the position display unit and the electronic gear.

Specifying Jerk for Axis Motion

The jerk used in an actual positioning motion is specified with the *Jerk* input variable to the motion control instruction. The same value is used for acceleration and deceleration.

Use the following formula to calculate the value to set for the jerk.
Jerk = Acceleration rate ÷ (Time of acceleration × Ratio of time to apply jerk during acceleration/2)

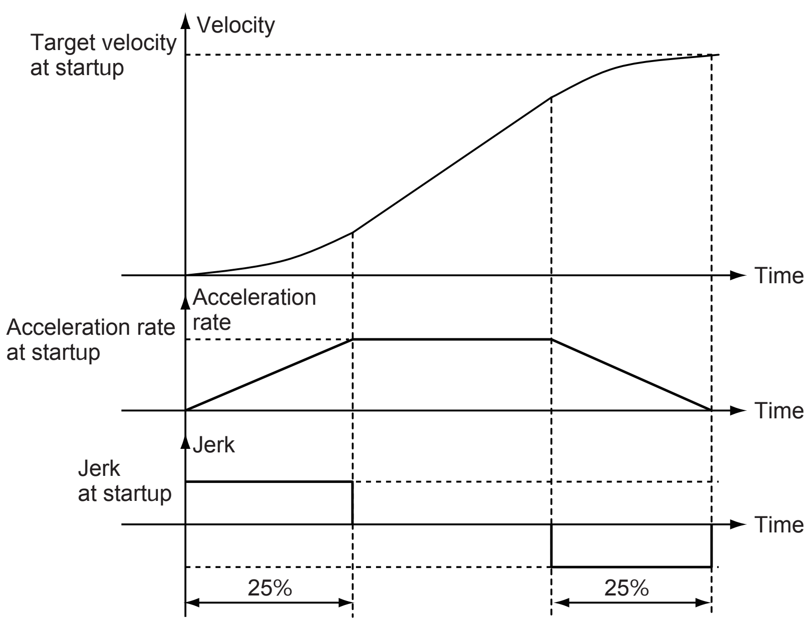
Jerk is applied in two sections: at the start of acceleration and at the end of acceleration. The time that jerk is applied is therefore divided by 2.

● Example of Velocity Control When Jerk Is Specified

The acceleration will change at a constant rate over the range where jerk is specified. The command velocity will form a smooth S curve. A fixed acceleration rate is used in areas where the jerk is set to 0. This command velocity will form a straight line.

Example: Acceleration of 25,000 mm/s^2, Acceleration Time of 0.1 s, and a Jerk Application Rate of 50%

Jerk = 25,000/(0.1 × 0.5/2) = 1,000,000 (mm/s^3)



Monitoring Jerk

You can read Axis Variables in the user program to monitor jerk.

Variable name	Data type	Meaning	Function
_MC_AX[0-63].Cmd.Jerk	LREAL	Command Current Jerk	This is the current value of the command jerk.

9-5-5 Specifying the Operation Direction

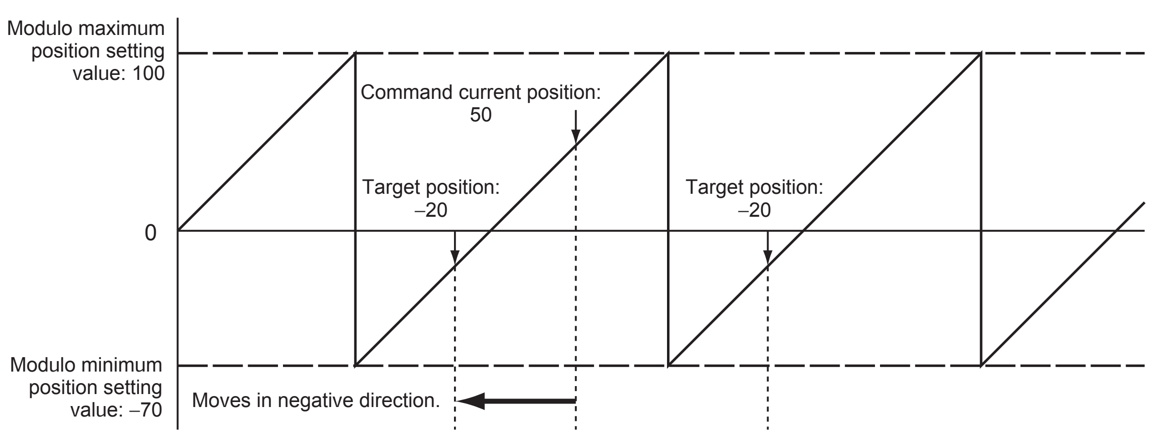
If you want to specify a rotation direction, such as shortest way, using an index table, set the Count Mode to Rotary Mode. Next, set the operation direction with the *Direction* input variable to the motion control instruction for an absolute position. If you set the direction to the shortest way, positive direction, negative direction, or current direction, you can specify a position that is greater than or equal to the modulo minimum position and less than the modulo maximum position within one turn of the ring counter. The *Direction* input variable will be ignored when the Count Mode is set to Linear Mode. Positioning will be performed to the target position.

The following table lists the different directions you can specify in the MC Function Module.

Direction	Operation
Shortest way	Motion starts in the direction where the command current position and the target position are closer to each other.
Positive direction	Motion starts in the positive direction.
Negative direction	Motion starts in the negative direction.
Current direction	Motion starts in the same direction as the previous operation.
No direction specified	Motion starts in the direction that does not pass through the upper and lower limits of the ring counter. With this direction specification, you can specify a target position that exceeds the upper or lower limits of the ring counter. If that occurs, relative positioning is performed using the difference between the target position and the command current position as the target distance. This enables you to perform multi-turn positioning on the ring counter.

Example for Shortest Way

The following example illustrates when positioning is performed towards a target position of –20 when the command current position is 50.

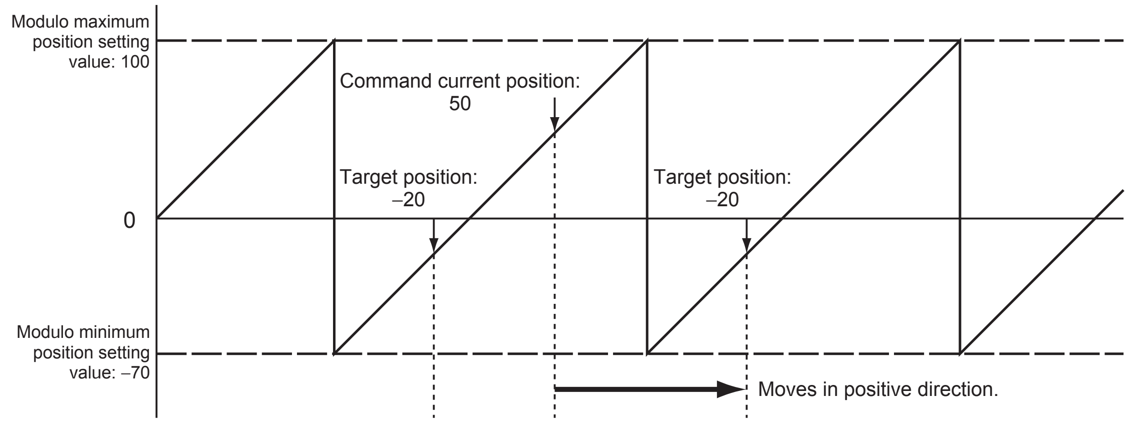


Additional Information

Moves in the same direction as the Current Direction specification if the travel distance is the same in the positive and negative directions.

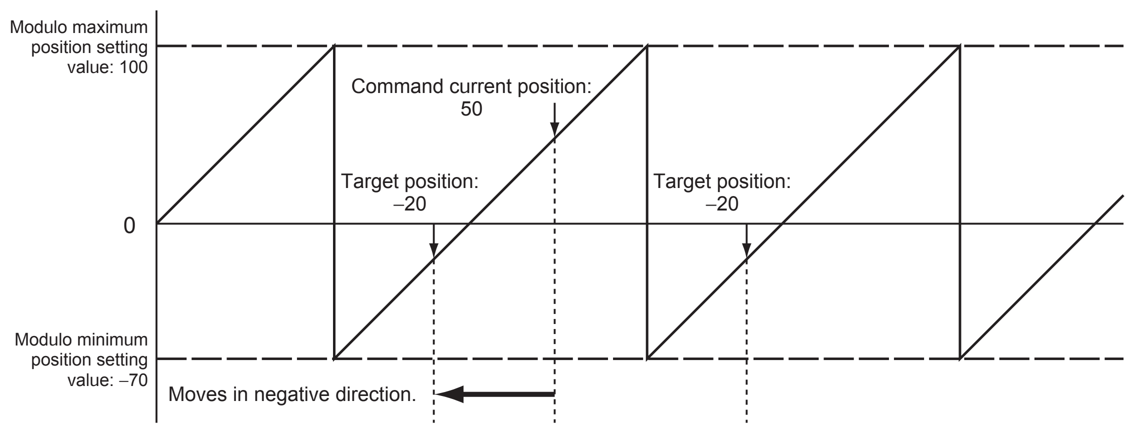
Example for Positive Direction

The following example illustrates when positioning is performed towards a target position of –20 when the command current position is 50.



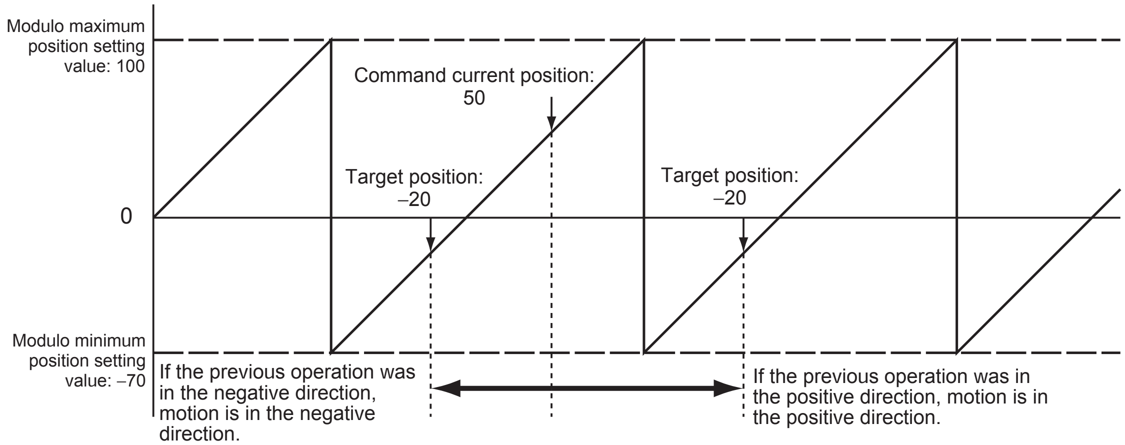
Example for Negative Direction

The following example illustrates when positioning is performed towards a target position of –20 when the command current position is 50.



Example for Current Direction

The following example illustrates when positioning is performed towards a target position of –20 when the command current position is 50.



The direction of the previous operation is given in the Command Direction in the Axis Variable.

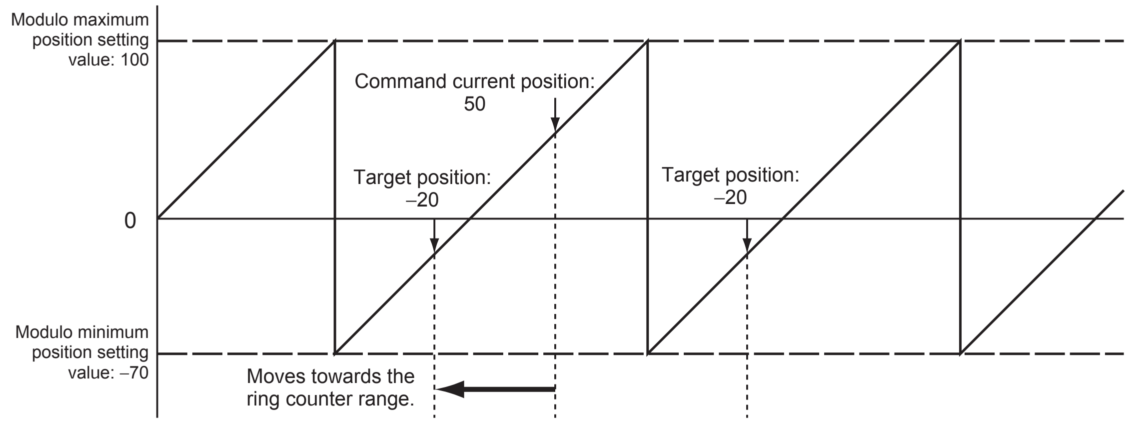
Precautions for Correct Use

Observe the following precautions on the operation direction of the previous operation.

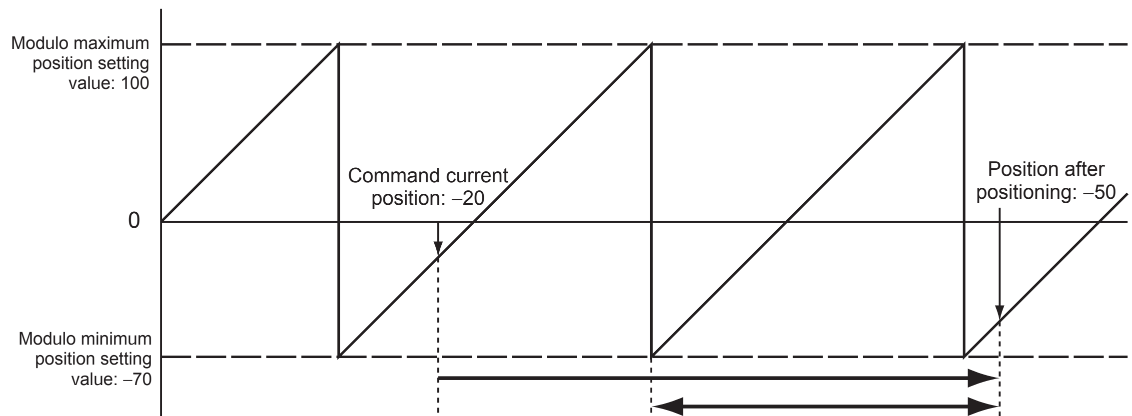
- If the MC_Home or MC_HomeWithParameter instruction exceeds the point where the home input was detected and reverses operation, the opposite direction of the home input detection direction is used.
- If a homing compensation value is set for the MC_Home or MC_HomeWithParameter instruction, the axis will move in the direction of the compensation value.
- If an immediate stop is specified for the MC_TouchProbe (Enable External Latch) instruction, the latch position may be exceeded and the direction may be reversed.
- The direction may be reversed for the MC_MoveFeed (Interrupt Feeding) instruction.
- When the MC_ResetFollowingError instruction is executed, the error is set to zero, so the command direction is used.
- If an immediate stop is specified for an external input signal or resetting the error counter is specified for stopping for a limit input, the operation may reverse direction toward the position where the external input signal was received.

Example for No Direction Specification

The following example illustrates when positioning is performed towards a target position of –20 when the command current position is 50.



Similarly, the following example illustrates when the ring counter upper limit is 100, the lower limit is –70, the command current position is –20, and positioning is performed towards a target position of 290.



9-5-6 Re-executing Motion Control Instructions

This section describes how to modify input variables of the same instance of a motion control instruction during operation of a single axis and re-execute that instruction. The input variables *Position* (Target Position), *Distance* (Travel Distance), *Velocity* (Target Velocity), *Acceleration* (Acceleration Rate), *Deceleration* (Deceleration Rate), and *Torque* (Target Torque) and sometimes other input variables can be changed by re-execution. An instruction error will occur if you change an input variable that cannot be changed and attempt to re-execute the instruction. If you re-execute an instruction that has been buffered due to multi-execution of instructions, the input variables for the instruction in the buffer will change.

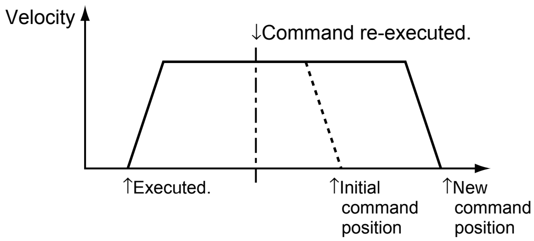
For details on input variables that can be changed, refer to the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

Changing the Target Position

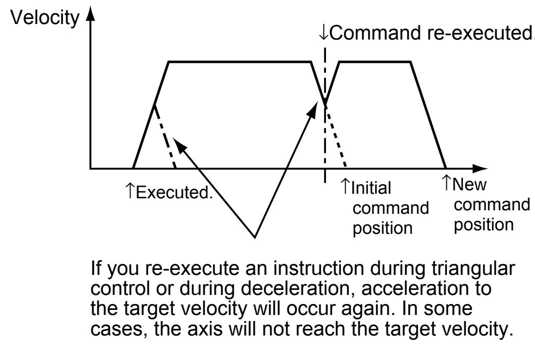
If you change the target position with re-execution, the operation may change depending on the timing of the change and the new target position. If the direction of motion reverses due to a change in the target position, you can choose to decelerate to a stop after a reverse turn or stop immediately after reversing with the Operation selection at Reversing axis parameter.

● When a Reverse Turn Does Not Occur for the New Command Value

Re-executing Instruction during Constant-velocity Motion

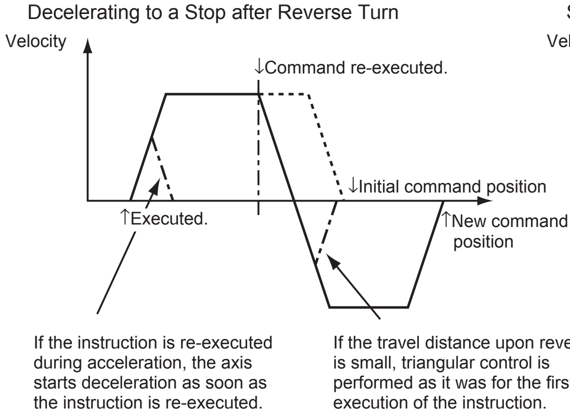


Re-executing Instruction during Acceleration/deceleration

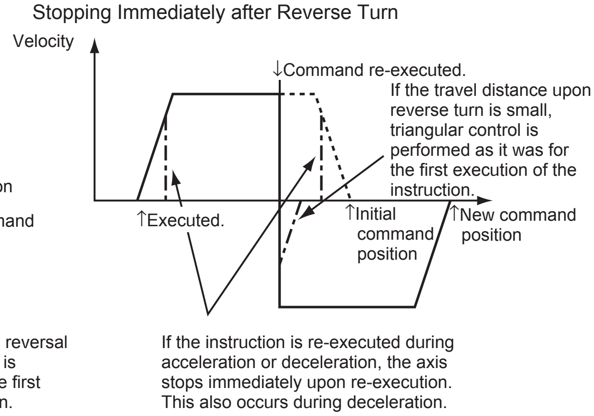


● When a Reverse Turn Occurs for the New Command Value

Decelerating to a Stop after Reverse Turn



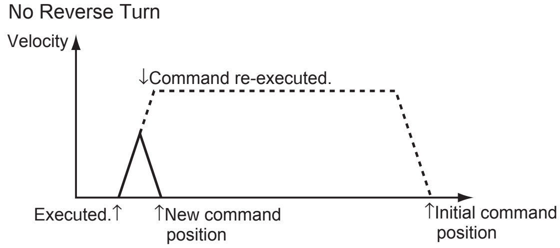
Stopping Immediately after Reverse Turn



● Triangular Control Patterns

The triangular control shown in the figure below may result if the travel distance is shortened due to a change in the target position.

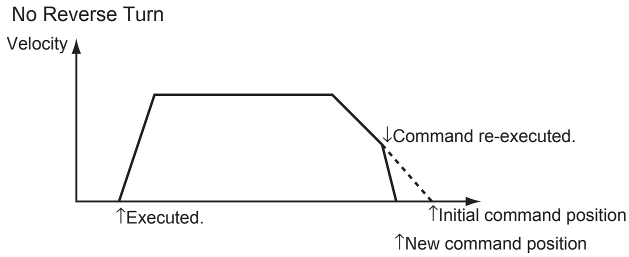
No Reverse Turn



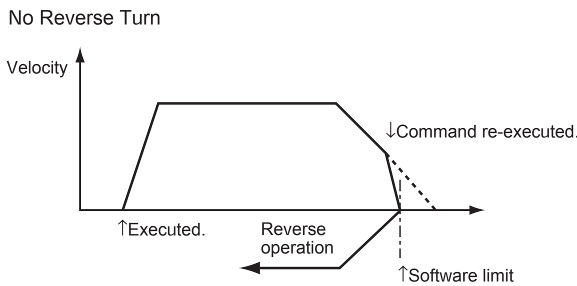
● Excessive Deceleration Patterns

In the following case, priority is given to stopping at the target position. Therefore, the deceleration rate will exceed the specified deceleration rate. If the deceleration rate exceeds the rate that is set in the Maximum Deceleration axis parameter, the operation set in the Acceleration/Deceleration Over axis parameter setting is performed.

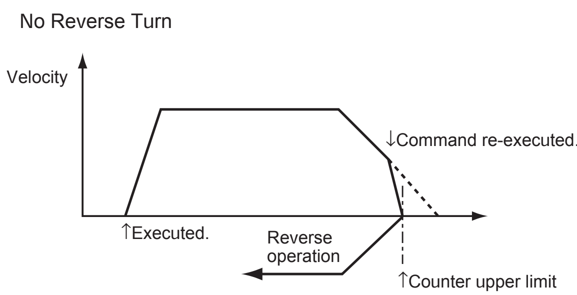
If There Is No Reverse Turn and the Target Position Would Be Exceeded at the Specified Deceleration Rate



If There Is A Reverse Turn and Decelerating to a Stop Would Exceed a Software Limit

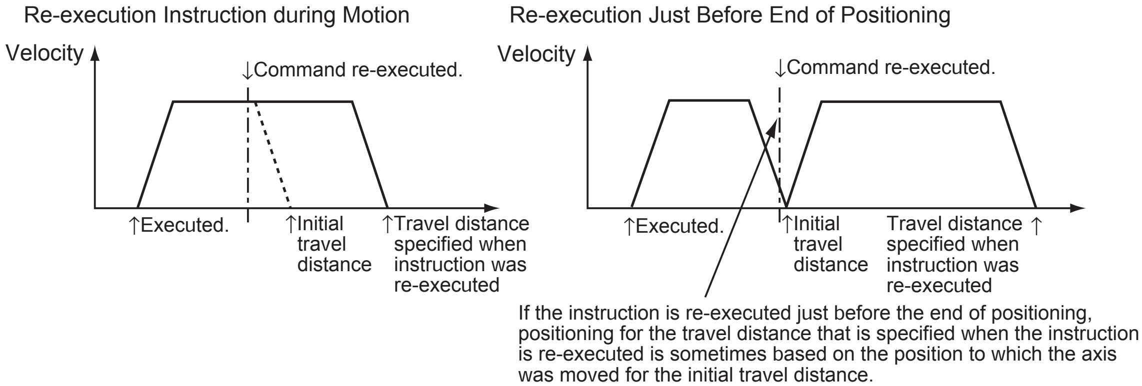


If There Is A Reverse Turn and Decelerating to a Stop Would Result in Command Current Position Overflow or Underflow



Changing the Travel Distance

Even if you change the travel distance and re-execute the MC_MoveRelative (Relative Positioning) instruction, positioning is performed for the new travel distance in reference to the position where the motion first started. However, if the instruction is executed again just before positioning is completed, it may be executed as a new instruction rather than as a re-execution of the same instruction.



Precautions for Correct Use

Do not change the travel distance and re-execute the instruction just before the end of positioning.

Changing the Target Velocity

The operation is changed only during acceleration (including acceleration for triangular control) and constant-velocity motion. Changes are also accepted when the axis is decelerating, but operation is not affected.

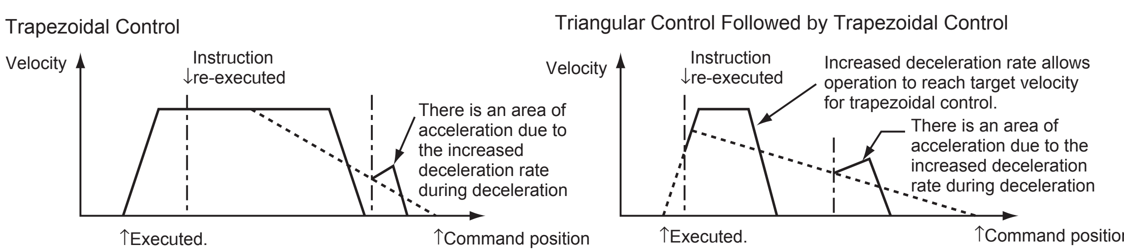
Changing the Acceleration Rate

The operation is changed only during acceleration and acceleration during triangular control. If it is changed when moving at a constant speed, the changed rate applies to acceleration for an override. Changes are also accepted when the axis is decelerating, but operation is not affected.

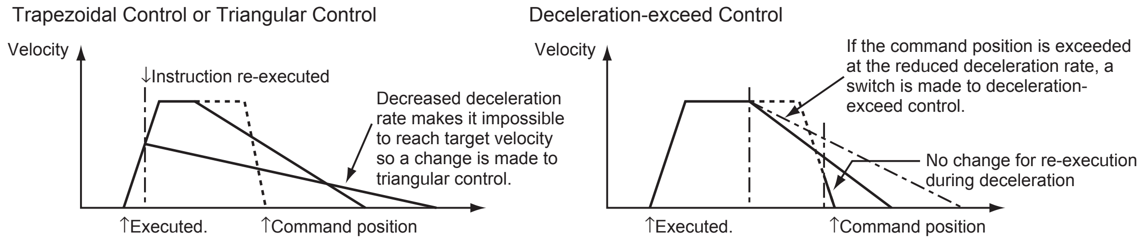
Changing the Deceleration Rate

The deceleration rate is changed only during acceleration, constant-velocity motion, deceleration, triangular control, or during deceleration-exceed control. If the new deceleration rate causes the axis to exceed the target position, stopping at the target position is given the highest priority. Therefore, in this case, the actual deceleration rate will exceed the specified deceleration rate.

● Patterns Where Deceleration Rate Increases



● Patterns Where Deceleration Rate Decreases

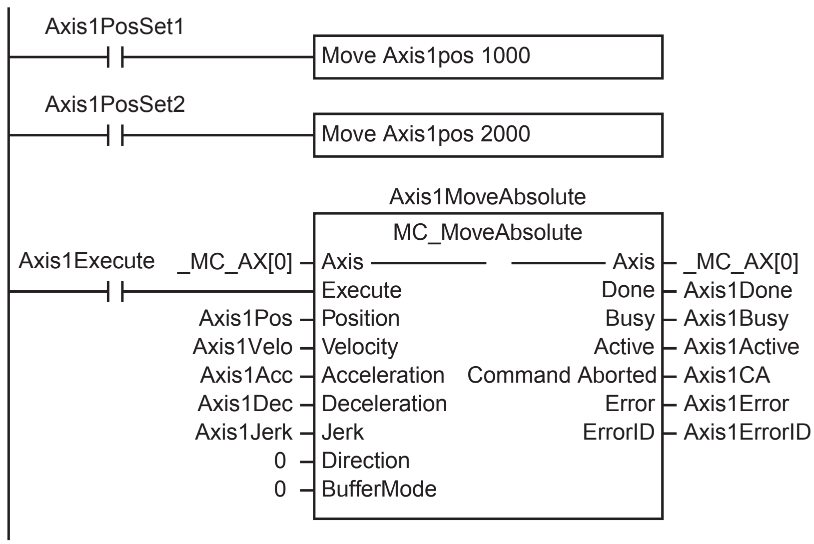


Changing the Torque Command

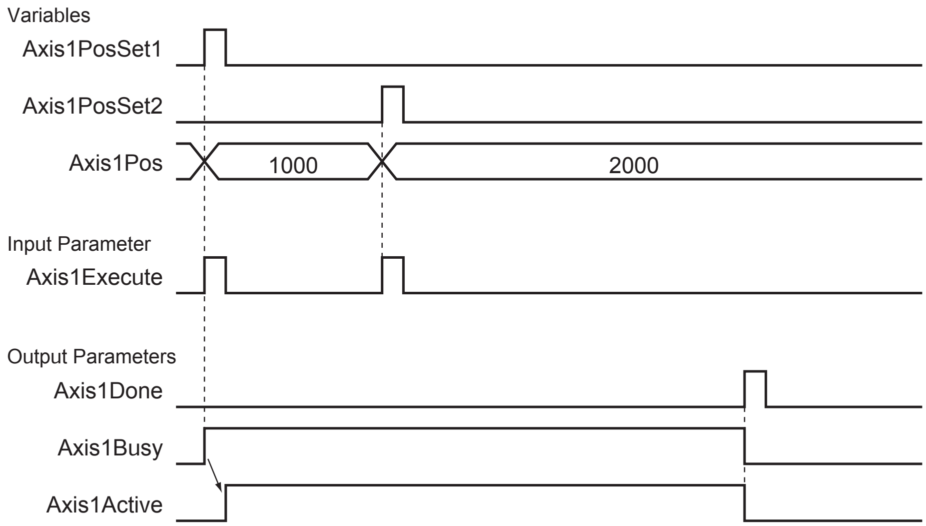
The torque command value will change based on the torque ramp specification when you re-execute a motion control instruction.

Programming Example for Re-execution

This example demonstrates changing the target position from 1000 to 2000 for absolute positioning. In this example, the variable *Axis1Pos* is used as the input parameter to the target position. Specify the target position to 1000 with the MOV instruction and change *Axis1Execute* to TRUE to begin positioning. Specify the target position to 2000 during operation and change *Axis1Execute* to TRUE again to switch to a positioning operation for the new target position of 2000.



- **Timing Charts**



Precautions for Correct Use

For input variables that are not changed, always use the same values as before re-execution of the instruction.

9-5-7 Multi-execution of Motion Control Instructions (Buffer Mode)

You can execute another motion control instruction while an axis is moving. In the PLCopen® technical specifications, this functionality is defined as Buffer Mode, but in the MC Function Module this is sometimes referred to as multi-execution of instructions. You can use multi-execution of instructions to execute multiple motion control instructions in sequence without stopping the overall motion.

The following terms are used in relation to multi-execution of instructions in the MC Function Module.

Term		Meaning
This manual	PLCopen®	
Current instruction	Previous function block	The motion control instruction that was in operation just before executing the multi-execution instruction.
Buffered instruction	Next function block	A motion control instruction that was executed during an axis motion and is waiting to be executed.
Transit velocity	Blending	When blending is specified, it specifies the command velocity to use by the current instruction to move to the specified target position.

You can set the *BufferMode* (Buffer Mode Selection) input variable to motion control instruction to select one of the following Buffer Modes. The main difference between these modes is the timing at which the buffered instructions are executed and the transit velocity.

Buffer Mode		Description of operation
Aborting		The current instruction is aborted and the multi-executed instruction is executed.
Buffered		The buffered instruction is executed after the operation for the current instruction is normally finished.
Blending		The buffered instruction is executed after the target position of the current instruction is reached. In this mode, no stop is performed between the current instruction and the buffered instruction. You can select from the following transit velocities for when the current instruction reaches the target position.
	Blending Low (low velocity)	The transit velocity is set to the target velocity of the current instruction or the buffered instruction, whichever is lowest.
	Blending Previous (previous velocity)	The target velocity of the current instruction is used as the transit velocity.
	Blending Next (next velocity)	The target velocity of the buffered instruction is used as the transit velocity.
	Blending High (high velocity)	The transit velocity is set to the target velocity of the current instruction or the buffered instruction, whichever is highest.

The multi-execution instruction is buffered in the MC Function Module and will be executed at the specified *BufferMode* timing and transit velocity for both buffered and blending modes. There is one buffer for each axis. If aborting is specified, the instruction that was executed last is executed immediately, so it is not buffered.

Precautions for Correct Use

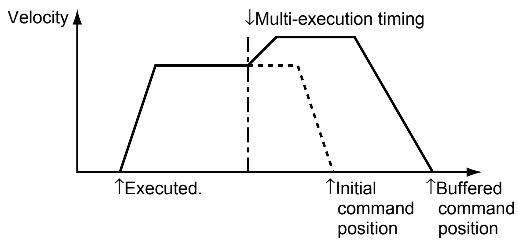
- Only one multi-execution instruction is buffered for each axis. If multi-execution is performed for two or more instructions, an instruction error will occur.
- Multi-execution of multi-axes coordinated control instructions (axes group instructions) is not possible for axes operating as a single axis. Similarly, multi-execution of single-axis control instructions is not possible for axes operating under multi-axes coordinated control (axes group instructions). An instruction error will occur if these rules are broken.

Aborting

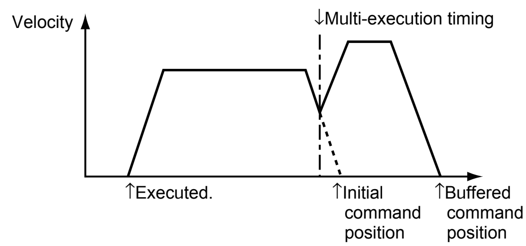
This is the default mode. No buffering is performed in this mode. The current command is aborted and the new instruction is executed. Aborting Mode can be used for multi-execution of instructions for motion control instructions for both single-axis control and synchronized control.

● When a Reverse Turn Does Not Occur for the Command Position of the Multi-execution Instruction

Executing More than One Instruction during Constant-velocity Motion



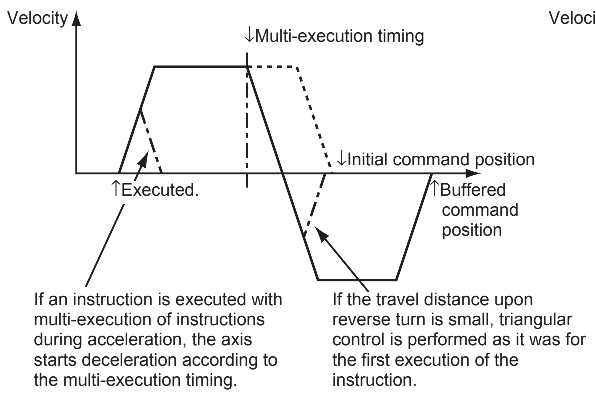
Multi-execution during Acceleration/Deceleration



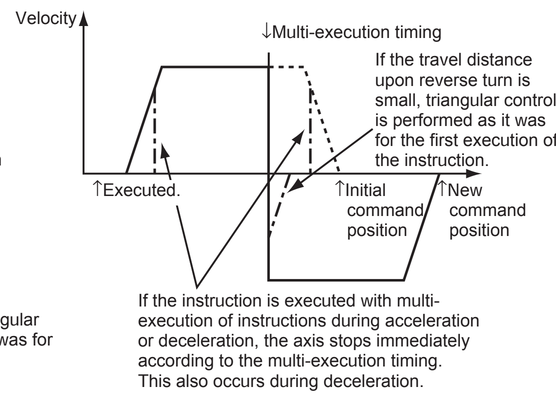
If you use multi-execution of an instruction during triangular control or during deceleration, the axis will accelerate to the target velocity of the buffered instruction. In some cases, the axis will not reach the target velocity.

● When a Reverse Turn Occurs for the Command Position of the Multi-execution Instruction

Decelerating to a Stop after Reversing

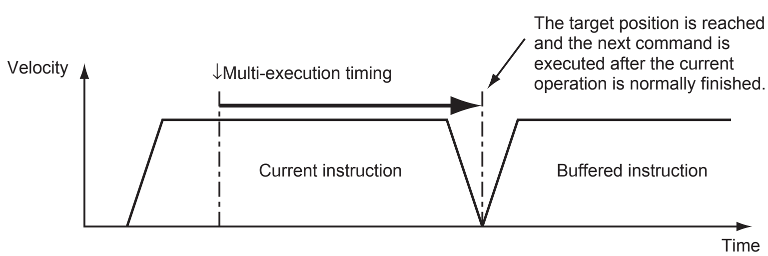


Stopping Immediately after Reversing



Buffered

The buffered instruction remains in the buffer until the operation of the current instruction is finished. The buffered instruction is executed after the operation for the current instruction is normally ended.



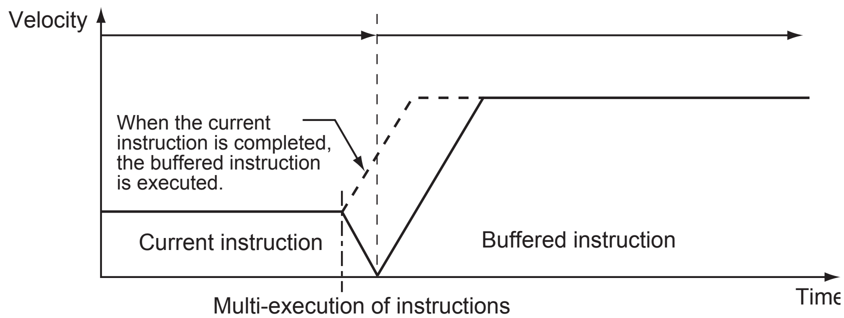
Blending

The buffered instruction remains in the buffer until the target position of the current instruction is reached. The buffered instruction is executed after the current instruction's target position is reached. However, motion does not stop at this time. Operation transitions to the next instruction at the velocity specified with the *BufferMode* (Buffer Mode Selection) input variable. For relative travel, the final position will be the total of the values for both instructions. For absolute travel, the final position will be the target position of the second multi-execution instruction. The Acceleration/Deceleration Over axis parameter is used to select one of the following operations for when the target position would be exceeded with the values that are set in the Maximum Acceleration and Maximum Deceleration axis parameters.

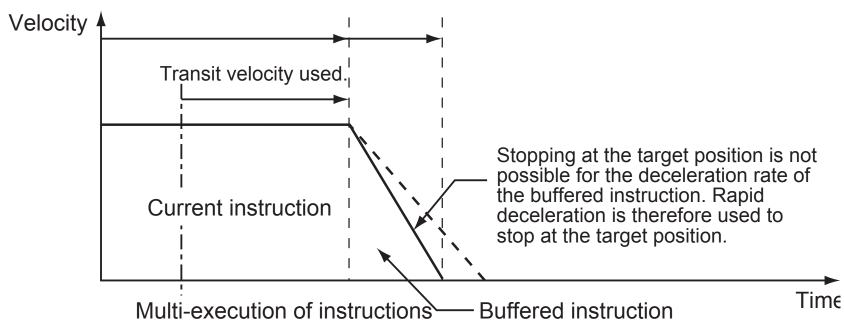
- Use rapid acceleration/deceleration. (Blending is changed to Buffered.)
- Use rapid acceleration/deceleration.
- Minor fault stop

An example for an Acceleration/Deceleration Over operation is given below.

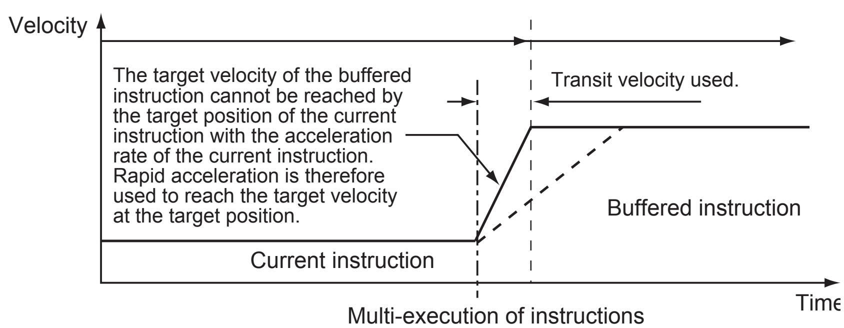
- Use Rapid Acceleration/Deceleration (Blending Is Changed to Buffered)



- Use Rapid Acceleration/Deceleration
(Here, BufferMode is set to blend with previous.)



- Use Rapid Acceleration/Deceleration
(Here, BufferMode is set to blend with next.)



In a blending mode you cannot combine single-axis and synchronized control.

- **Blending Low (Low Velocity)**

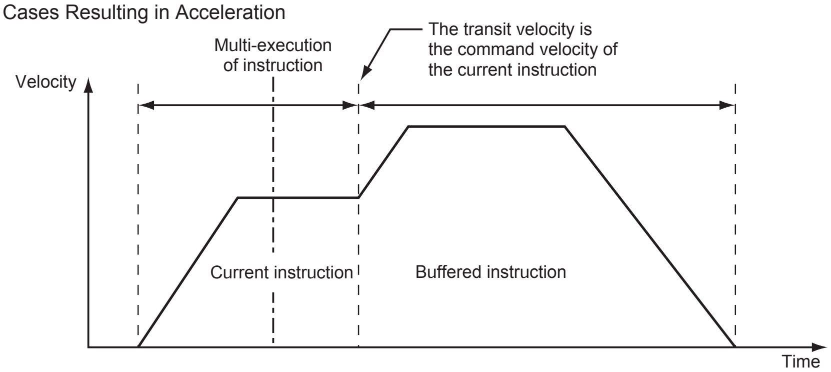
Operation is performed using the target position of the current instruction and the target velocity that is the slower of the target velocities for the current instruction and buffered instruction.

- **Blending Previous (Previous Velocity)**

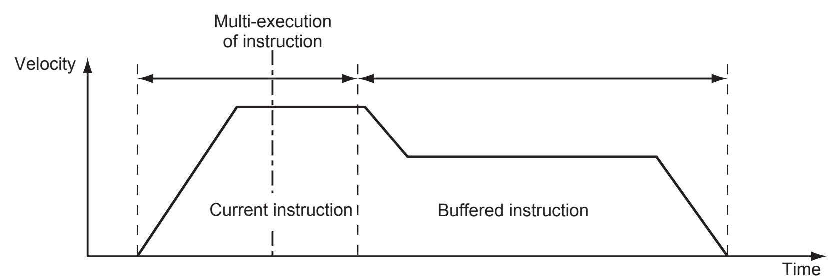
Operation is performed with the target velocity of the current instruction until the target position of the current instruction is reached. Operation is performed after acceleration/deceleration to the target velocity of the buffered instruction once the target position is reached.

When the Direction of Operation Does Not Change

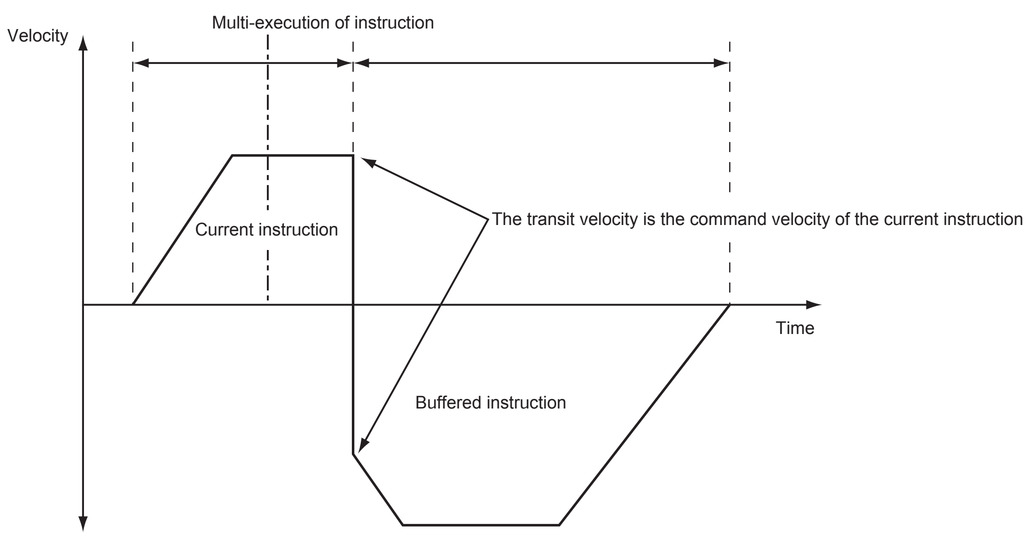
Cases Resulting in Acceleration



Cases Resulting in Deceleration

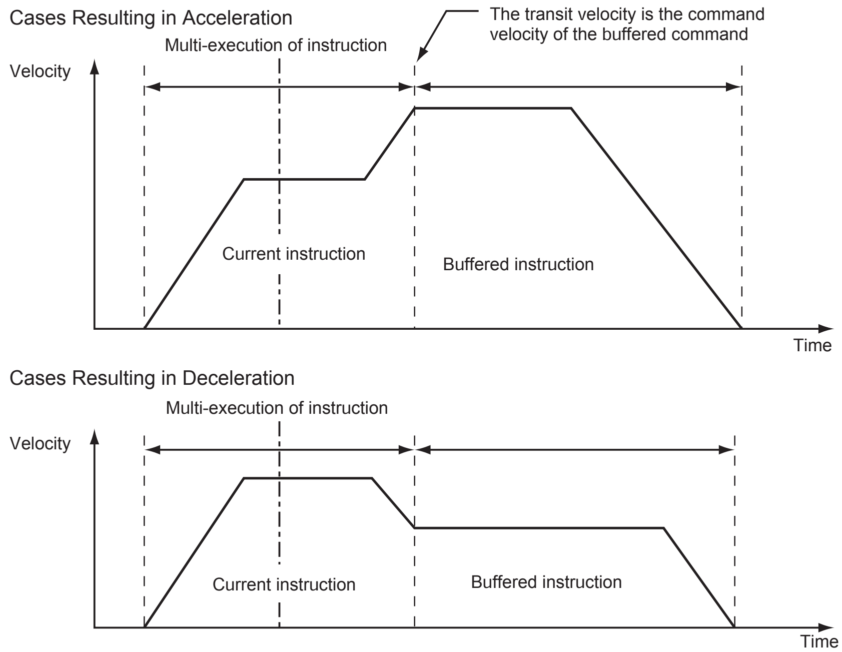


When the Direction of Operation Changes



Blending Next (Next Velocity)

Operation is performed using the target position of the current instruction and the target velocity of the buffered instruction.



Blending High (High Velocity)

Operation is performed using the target position of the current instruction and the target velocity that is the faster of the target velocities for the current instruction and buffered instruction.

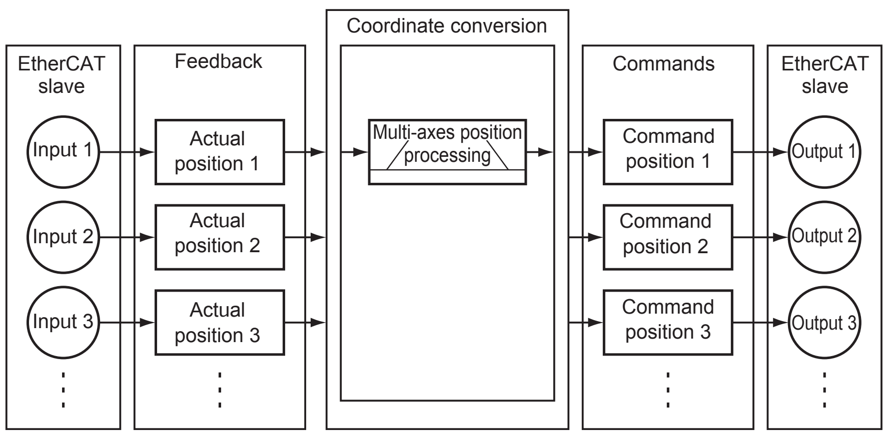
9-6 Multi-axes Coordinated Control

This section describes the operation of multi-axes coordinated control. With the MC Function Module, you can set an axes group in advance from the Sysmac Studio to perform interpolation control for multiple axes.

9-6-1 Outline of Operation

Multi-axes coordinated control performs a motion with multiple related axes together as a single group to control the path of the target control object. The MC Function Module treats all axes that perform coordinated operation as an axes group. Axes groups are set from the Sysmac Studio. In the user program, turn ON the Servo for each axis and then enable the axes group that is going to perform the multi-axes coordinated control. The purpose of multi-axes coordinated control is the coordinated operation of all axes belonging to the target axes group. Therefore, you cannot execute any single-axis operation motion control instructions on the axes in an enabled axes group. Furthermore, if any error occurs for any axis in an axes group, all axes in the axes group will stop according to the setting of the Axis Group Stop Method group axes parameter.

The MC Function Module can perform linear interpolation with two to four axes or circular interpolation with two axes.



Additional Information

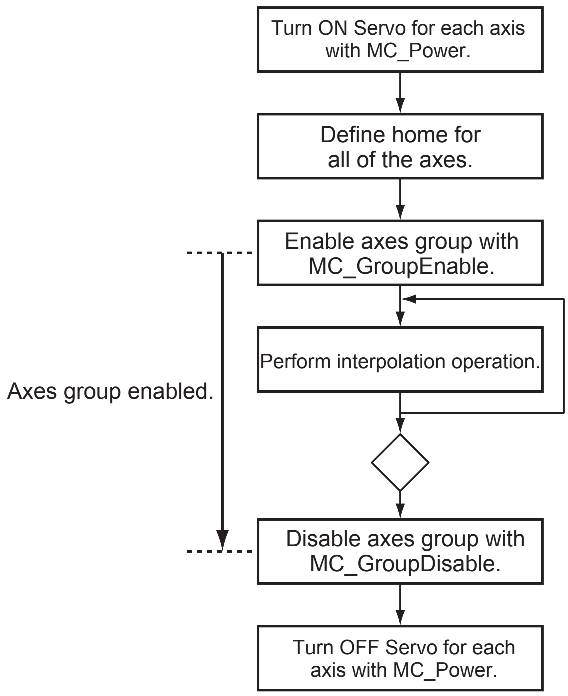
For devices that require you to modify the grouping of axes in motion to perform interpolation control, you must create multiple axes groups that include the axes to modify from the Sysmac Studio beforehand. After completing this step, you can execute by specifying the enabled axes groups from the user program during operation.

With a CPU Unit with a unit version of 1.01 or later and Sysmac Studio version 1.02 or higher, you can use the MC_ChangeAxesInGroup (Change Axes in Group) instruction to change the composition axes for an axes group that is disabled.

For details on axes groups, refer to *3-3 Axes Groups*.

Enabling and Disabling Axes Groups

To enable an axes group, specify the axes group for the MC_GroupEnable (Enable Axes Group) instruction. An instruction error will occur if you try to execute an axes group instruction when the axes group is still disabled. To disable an axes group, specify the axes group for the MC_GroupDisable (Disable Axes Group) instruction. When you disable an axes group that is in operation, all axes in that axes group will decelerate to a stop at the maximum deceleration rate that is specified in their axis parameter settings.



For details on enabling and disabling axes groups, refer to the MC_GroupEnable (Enable Axes Group) and MC_GroupDisable (Disable Axes Group) instructions in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

Changing the Axes in an Axes Group

You can use the MC_ChangeAxesInGroup (Change Axes in Group) instruction to temporarily change the composition axes for an axes group that is disabled. If the axes group is enabled, use the MC_GroupDisable (Disable Axes Group) instruction to disable the axes group before you change the composition axes. A CPU Unit with unit version 1.01 or later and Sysmac Studio version 1.02 or higher are required to use this instruction.

Precautions for Correct Use

Changes made using the MC_ChangeAxesInGroup (Change Axes in Group) instruction will not be saved to non-volatile memory in the CPU Unit. If you cycle the power supply or download the settings from the Sysmac Studio, the parameter settings in the non-volatile memory are restored.

For details on changing the composition axes of an axes group, refer to the MC_ChangeAxesInGroup (Change Axes in Group) instruction in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

Reading Axes Group Positions

You can use the MC_GroupReadPosition (Read Axes Group Position) instruction to read the command current positions and the actual current positions of an axes group. A CPU Unit with unit version 1.01 or later and Sysmac Studio version 1.02 or higher are required to use this instruction.

For details on reading the axis positions for an axes group, refer to the MC_GroupReadPosition (Read Axes Group Position) instruction in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

Resetting Axes Group Errors

If an error occurs in an axes group, you can use the MC_GroupReset instruction to remove the error once you have eliminated the cause.

For details on resetting axes group errors, refer to the MC_GroupReset (Group Reset) instruction in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

Refer to the *NX-series Position Interface Units User's Manual* (Cat. No. W524) for the differences when you use NX-series Pulse Output Units.

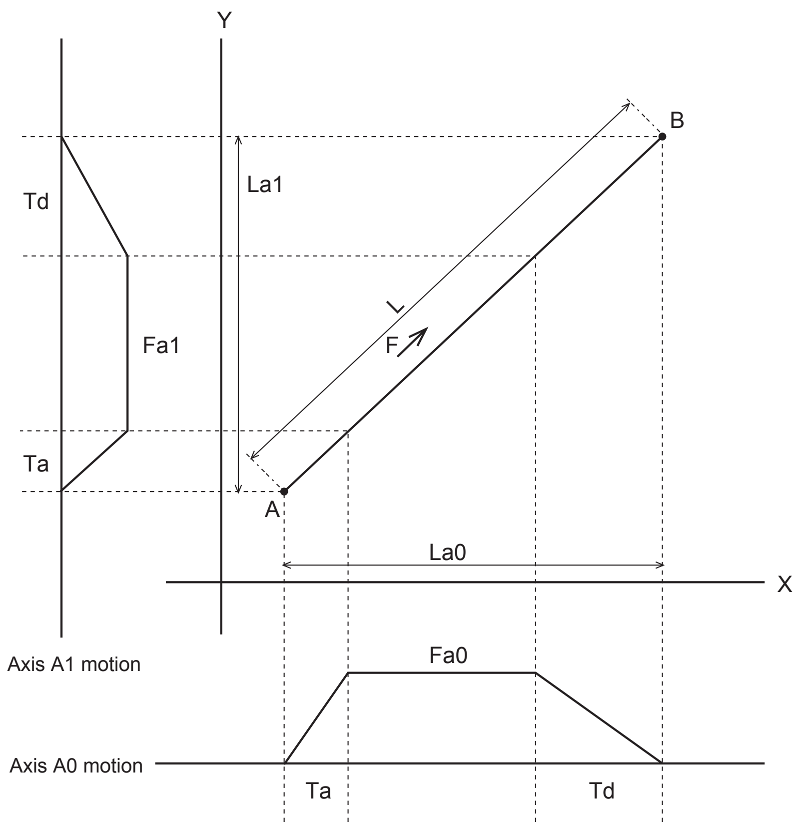
9-6-2 Linear Interpolation

Linear interpolation is used to move 2 to 4 of the logical axes A0 to A3 in a straight line between a start point and an end point. Either absolute or relative positioning is possible. You can specify the interpolation velocity, interpolation acceleration, interpolation deceleration, and jerk.

The MC Function Modules uses the following three kinds of linear interpolation instructions.

- MC_MoveLinear (Linear Interpolation)
  You can specify the *MoveMode* input variable to select between linear interpolation to an absolute value or linear interpolation to a relative value. This instruction is unique to the MC Function Module.
- MC_MoveLinearAbsolute (Absolute Linear Interpolation)
  This instruction performs linear interpolation to an absolute value. This instruction is defined in the PLCopen® technical specifications.
- MC_MoveLinearRelative (Relative Linear Interpolation)
  This instruction performs linear interpolation to a relative value. This instruction is defined in the PLCopen® technical specifications.

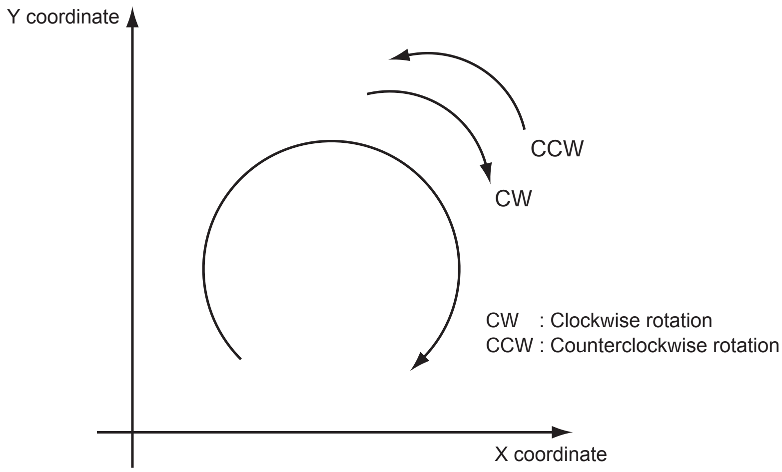
The following figure shows linear interpolation of 2 axes from point A to point B.



For details on linear interpolation, refer to the MC_MoveLinear (Linear Interpolation), MC_MoveLinearAbsolute (Absolute Linear Interpolation), and MC_MoveLinearRelative (Relative Linear Interpolation) instructions in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

9-6-3 Circular Interpolation

Circular interpolation is used to move two of the logical axes A0 to A3 in a circular motion on a 2D plane. Either absolute or relative positioning is possible. You can specify the circular interpolation mode, path direction, interpolation velocity, interpolation acceleration, interpolation deceleration, and combined jerk for the two axes.



With the MC Function Module, you can specify the following three kinds of circular interpolation methods with the input variable *CircMode* (Circular Interpolation Mode).

- Border point
- Center
- Radius

Precautions for Correct Use

Set the Count Mode to Linear Mode for the axis that you use for circular interpolation. If the instruction is executed with this axis in Rotary Mode, an instruction error will occur.

9-6-4 Axes Group Cyclic Synchronous Positioning

You can cyclically output specified target positions for the axes in an axes group. You can specify target positions that are calculated in the user program as absolute positions to move the axes in any desired path.

A CPU Unit with unit version 1.01 or later and Sysmac Studio version 1.02 or higher are required to use this instruction.

For details on axes group cyclic synchronous positioning for an axes group, refer to the MC_GroupSyncMoveAbsolute (Axes Group Cyclic Synchronous Absolute Positioning) instruction in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

9-6-5 Stopping Under Multi-axes Coordinated Control

Multi-axes coordinated control of axes groups will stop when you execute certain motion control instructions in the user program or when an error or some other problem occurs.

Stopping with Motion Control Instructions

Use the MC_GroupStop or MC_GroupImmediateStop instruction to stop axes group operation.

● MC_GroupStop Instruction

For linear interpolation or circular interpolation performed on an axes group, you can decelerate to a stop along the control path. You specify the deceleration rate and jerk. Specify a deceleration rate of 0 to send a command that immediately stops the Servo Drive or other device. Other operation commands are not acknowledged while decelerating to a stop for this instruction and while the input variable *Execute* is TRUE.

● MC_GroupImmediateStop Instruction

You can perform an immediate stop for all axes in the axes group. The immediate stopping method is determined by the setting of the Immediate Stop Input Stop Method axis parameter for each axis. The MC_GroupImmediateStop instruction can also be executed for an axes group that is decelerating to a stop for an MC_GroupStop instruction.

For details, refer to the MC_GroupStop and MC_GroupImmediateStop instructions in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

Stopping Due to Errors or Other Problems

● Stopping for Errors during Axes Group Motion

If an error that results in a deceleration stop occurs for any composition axis in the axes group during an axes group motion, all of the axes will decelerate to a stop on the interpolation path at the interpolation deceleration rate. The interpolation deceleration rate is determined by the deceleration rate that is specified for the controlling instruction. If an error that results in an immediate stop occurs for any composition axis in the axes group during an axes group motion, the other axes in the axes group will stop according to the setting of the Axes Group Stop Method parameter in the axes group parameters.

You can select one of the following stop methods for axes groups.

- Immediate stop
- Decelerate axes to a stop at maximum deceleration rate of the axes.
- Immediate stop and Servo OFF

● Stopping Due to Excessively Long Control Period

If motion control processing does not end within two periods, it is considered to be an excessive control period. Control will be stopped immediately.

● Stopping Due to Start of MC Test Run

All axes will decelerate to a stop at their maximum deceleration if a MC Test Run is started from the Sysmac Studio.

Stopping Due to Change in CPU Unit Operating Mode

All axes will decelerate to a stop at their maximum deceleration when the CPU Unit operating mode changes.

Additional Information

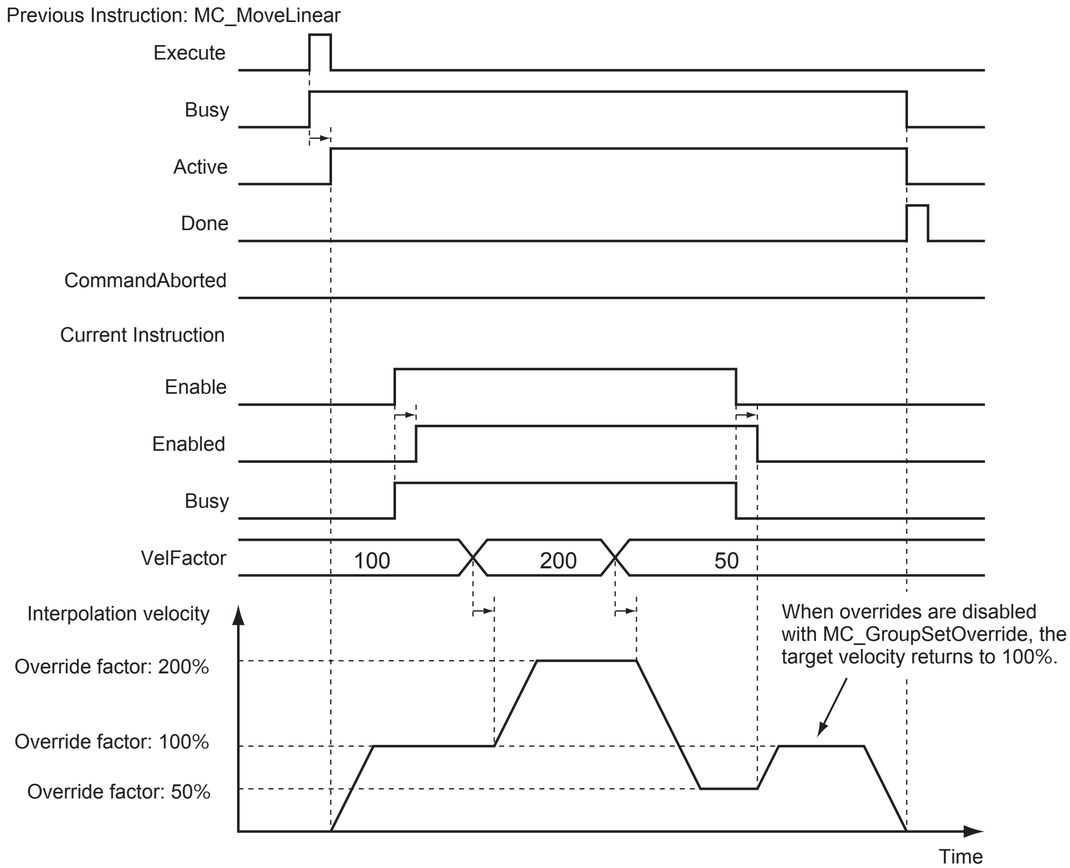
- If you execute the MC_GroupDisable (Disable Axes Group) instruction during axes group operation, the axes in the group will decelerate to a stop at their maximum deceleration rates.
- If you execute the MC_Stop instruction while an axes group is in operation, an error will occur for the axes and axes group and the axes group operation will decelerate to a stop with interpolation. The interpolation deceleration rate is determined by the deceleration rate that is specified for the controlling instruction.
- When the input variable *Enable* to the MC_Power (Servo ON) instruction changes to FALSE during axes group motion, the MC Function Module immediately stops the command value for that axis and turns OFF the Servo. When the Servo is turned OFF, the Servo Drive or other device will operate according to the settings in the Servo Drive or other device. Other axes in that axes group will stop with the stop method that is set in the Axes Group Stop Method axes group parameter. An error will occur for the axes group if this happens.
- When RUN mode changes to PROGRAM mode, any motion control instructions for current motions are aborted. The *CommandAborted* output variable from the instructions remain TRUE and the Servo remains ON.
- If the operating mode returns to RUN mode while a deceleration stop is in progress after the operating mode changes from RUN to PROGRAM mode, the output variable *CommandAborted* from the current motion control instructions change to TRUE.
- The save process will continue during a save for the MC_SaveCamTable Instruction.
- The generation process will continue when generation of the cam table is in progress for the MC_GenerateCamTable (Generate Cam Table) instruction.

9-6-6 Overrides for Multi-axes Coordinated Control

You can use the MC_GroupSetOverride (Set Group Overrides) instruction to set override factors for multi-axes coordinated control of the axes group in the current interpolation operation. The velocity override factor is set as a percentage of the target velocity for interpolation. It can be set between 0% and 500%. If an override factor of 0% is set for the interpolation target velocity, operating status will continue with the axis stopped at a velocity of 0. The set override factor is read as long as the overrides are enabled. If the overrides are disabled, the override factors return to 100%. If the maximum interpolation velocity is exceeded when an override factor is changed, the maximum interpolation velocity for the axes group is used.

Overrides for the MC_MoveLinear (Linear Interpolation) Instruction

An example of a time chart for using the Set Override Factors instruction for the MC_MoveLinear (Linear Interpolation) instruction is given below.



For details, refer to the MC_GroupSetOverride (Set Group Overrides) instruction in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

9-7 Common Functions for Multi-axes Coordinated Control

This section describes the common functions for multi-axes coordinated control.

9-7-1 Velocity Under Multi-axes Coordinated Control

To specify the velocity for multi-axes coordinated control, specify the interpolation velocity on the path. The unit is the same as for single axes, command units/s.

Types of Velocities

The following is the only type of interpolation velocity for axes groups supported by the MC Function Module.

Velocity type	Definition
Command interpolation velocity	This is the actual value of the command interpolation velocity output by the MC Function Module to control an axes group.

Axis Parameters That Are Related to Velocities

Parameter name	Function	Setting range	Default
Maximum Interpolation Velocity	Set the maximum interpolation velocity for the path. Set 0 for no interpolation velocity limit. If a target velocity that exceeds the maximum interpolation velocity is specified for an axes group operation instruction, the axis will move at the maximum interpolation velocity.	Non-negative long reals	800,000,000
Interpolation Velocity Warning Value	Set the percentage of the maximum interpolation velocity at which to output an interpolation velocity warning. No interpolation velocity warning is output if 0 is set. (Unit: %)	0 to 100	0

Specifying Target Velocities for Axis Operations

The interpolation velocity used in an actual positioning motion is specified by the *Velocity* (Target Velocity) input variable to the motion control instruction.

Monitoring Velocities

You can read Axes Group Variables from the user program to monitor the interpolation velocity.

Variable name	Data type	Meaning	Function
_MC_GRP[0-31].Cmd.Vel	LREAL	Command Interpolation Velocity	This is the current value of the command interpolation velocity. A plus sign is added during travel in the positive direction, and a minus sign is added during travel in the negative direction.

9-7-2 Acceleration and Deceleration Under Multi-axes Coordinated Control

Multi-axes coordinated control performs control on the path for the interpolation acceleration and interpolation deceleration rates. The unit is the same as for single axes, command units/s^2.

Axis Parameters That Are Related to Interpolation Acceleration and Interpolation Deceleration

Parameter name	Function	Setting range	Default
Maximum Interpolation Acceleration	Set the maximum interpolation acceleration for the path. Set 0 for no interpolation acceleration limit. (Unit: command units/s^2)	Non-negative long reals	0
Maximum Interpolation Deceleration	Set the maximum interpolation deceleration for the path. Set 0 for no interpolation deceleration limit. (Unit: command units/s^2)	Non-negative long reals	0
Interpolation Acceleration/Deceleration Over	Set the operation for when the maximum interpolation acceleration/deceleration rate would be exceeded after excessive acceleration/deceleration during acceleration/deceleration control of the axes group because stopping at the target position is given priority. 0: Use rapid acceleration/deceleration. (Blending is changed to Buffered.) 1: Use rapid acceleration/deceleration. 2: Minor fault error	0 to 2	0
Interpolation Acceleration Warning Value	Set the percentage of the maximum interpolation acceleration at which to output an interpolation acceleration warning. No interpolation acceleration warning is output if 0 is set. (Unit: %)	0 to 100	0
Interpolation Deceleration Warning Value	Set the percentage of the maximum interpolation deceleration rate at which to output an interpolation deceleration warning. No interpolation deceleration warning is output if 0 is set. (Unit: %)	0 to 100	0

Specifying an Interpolation Acceleration and Interpolation Deceleration for an Axes Group

The interpolation acceleration and interpolation deceleration rates used in an actual positioning motion are specified by the *Acceleration* (Acceleration Rate) and *Deceleration* (Deceleration Rate) input variables to the motion control instruction.

Monitoring Interpolation Acceleration and Interpolation Deceleration Rates

You can read Axes Group Variables in the user program to monitor interpolation acceleration and interpolation deceleration rates.

Variable name	Data type	Meaning	Function
_MC_GRP[0-31].Cmd.AccDec	LREAL	Command Interpolation Acceleration/Deceleration	This is the current value of the command interpolation acceleration/deceleration rate. A plus sign is added for acceleration, and a minus sign is added for deceleration.

9-7-3 Jerk for Multi-axes Coordinated Control

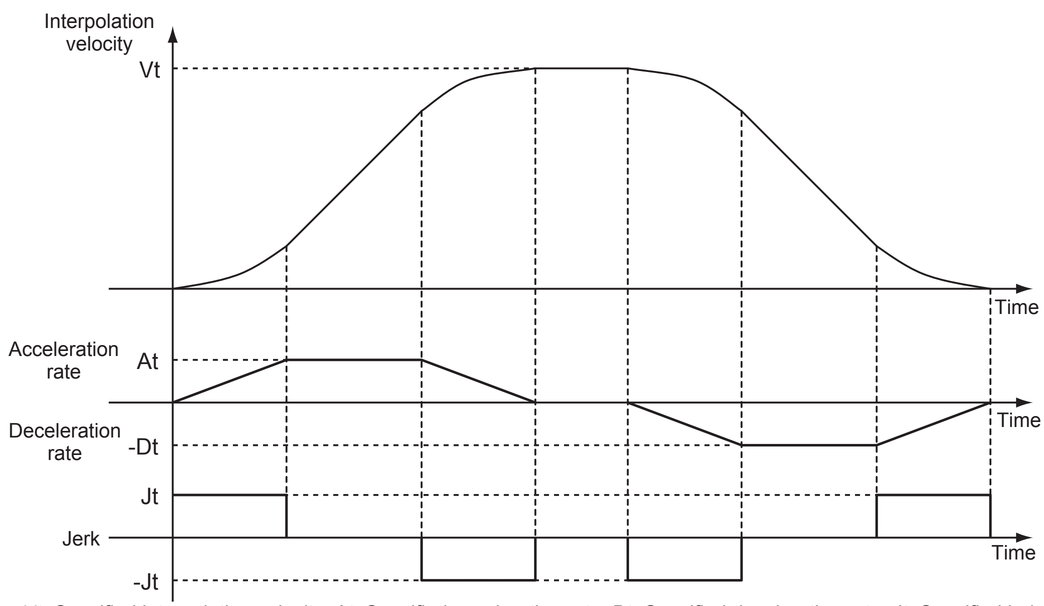
Jerk for multi-axes coordinated control is used to reduce shock and vibration on the machine by smoothing the interpolation acceleration/deceleration rate along the interpolation path into an S-curve. The unit is the same as for single axes, command units/s^3.

Specifying Jerk for Axes Group Motion

The jerk used in an actual interpolation is specified by the *Jerk* input variable to the motion control instruction.

Jerk Example (Setting Other than 0)

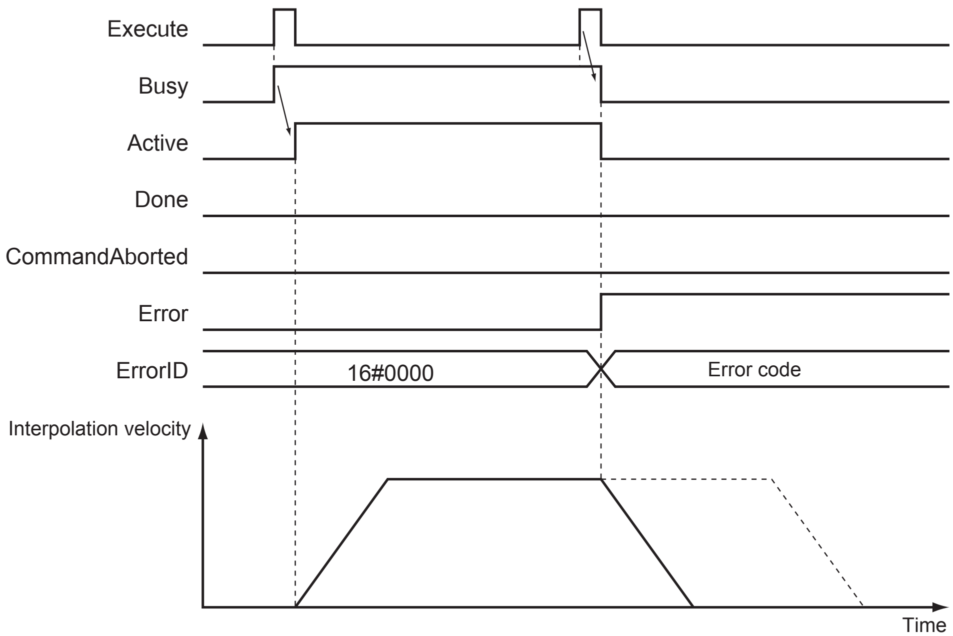
The acceleration/deceleration rate will change at a constant rate over the range where jerk is specified. The command interpolation velocity will form a smooth S-curve. A fixed interpolation acceleration rate is used in areas where the jerk is set to 0. This command interpolation velocity will form a straight line.



Vt: Specified interpolation velocity, At: Specified acceleration rate, Dt: Specified deceleration rate, Jt: Specified jerk

9-7-4 Re-executing Motion Control Instructions for Multi-axes Coordinated Control

If you re-execute a linear interpolation or circular interpolation instruction, an instruction error will occur.



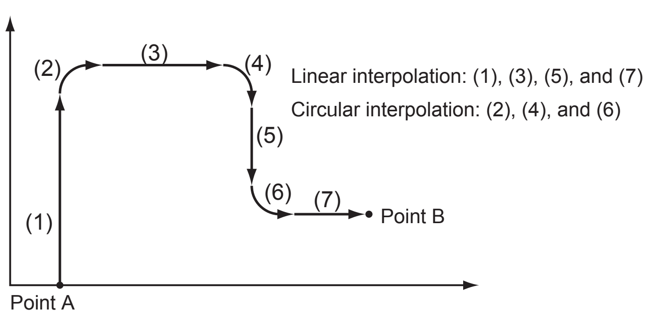
You can change the deceleration rate if you re-execute the MC_GroupStop instruction, but you cannot change the jerk in this way.

If you re-execute the MC_GroupReset instruction, the re-execution command will be ignored and error reset processing will continue.

For details on re-executing motion control instructions, refer to each instruction in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

9-7-5 Multi-execution (Buffer Mode) of Motion Control Instructions for Multi-axes Coordinated Control

You can perform multi-execution for multi-axes coordinated control in axes groups the same way as you can for axis operations. You can perform path control for multiple continuous lines and/or arcs if you use Buffer Mode under multi-axes coordinated control.



You can set the *BufferMode* input variable to motion control instruction to select one of the same Buffer Modes as are supported for single-axis operations. There are a total of eight instruction buffers for axes groups. Each axes group has one buffer for the instruction currently in operation and seven buffers for multi-execution instructions. Multi-execution of instruction cannot be used from an axis operation instruction to an axes group operation instruction and vice-versa.

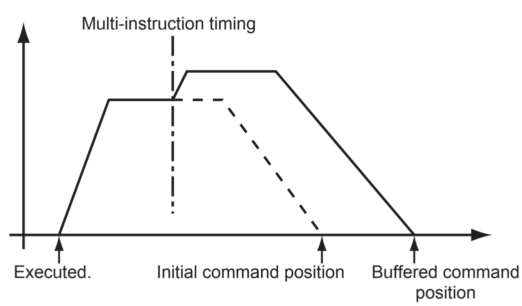
Precautions for Correct Use

- Up to seven instructions can be buffered at the same time for a single axes group. If multi-execution is performed for eight or more instructions, an instruction error will occur.
- Multi-execution of multi-axes coordinated control instructions (axes group instructions) is not possible for axes operating as a single axis. Similarly, multi-execution of single-axis control instructions is not possible for axes operating under multi-axes coordinated control (axes group instructions). An instruction error will occur if these rules are broken.

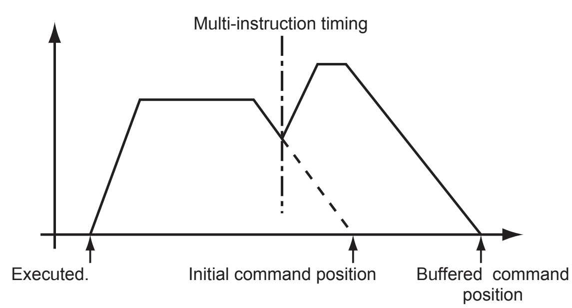
Aborting

This is the default mode. No buffering is performed in this mode. The current command is aborted and the new instruction is executed. Multi-execution of motion control instructions that have no *BufferMode* input variable will operate in Aborting Mode. Operation of the multi-execution instruction starts at the current interpolation velocity when the multi-execution instruction is executed. With Aborting Mode you cannot combine single-axis control, including synchronized single-axis control and axes group control. An instruction error will occur at the time of multi-execution if you execute an axes group operation on an axis currently in a single-axis motion. This will stop both the axes group and the single axis.

Multi-execution during Constant-velocity Motion

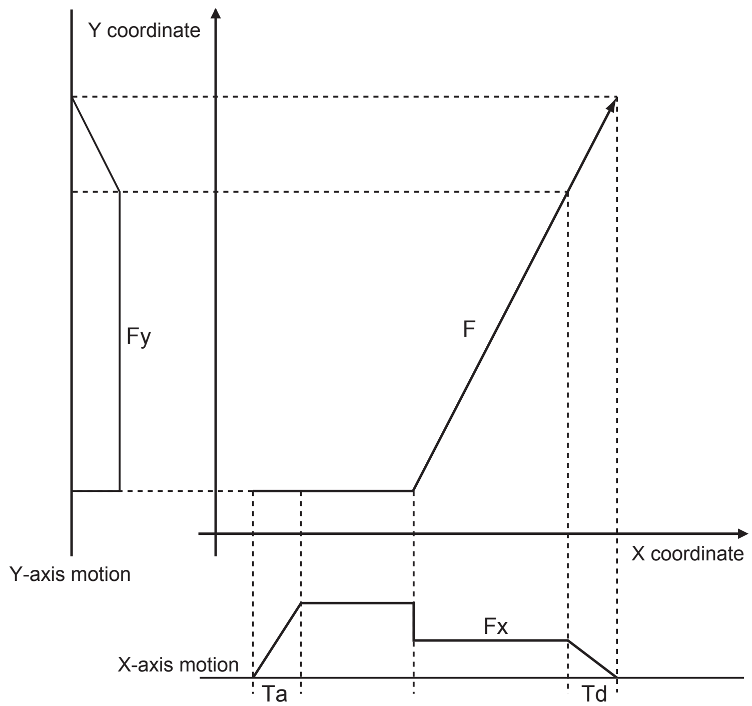


Multi-execution during Acceleration/Deceleration



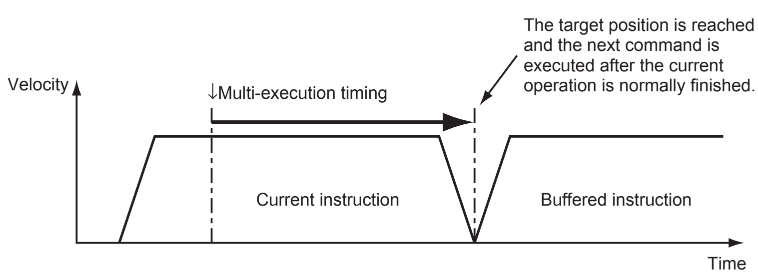
Multi-execution for axes groups is done so that the interpolation velocity remains continuous between instructions. If continuous operation is performed with an instruction with a travel distance of 0, the velocity changes for the axes will not be continuous.

Example: Interpolation Velocity and Velocities of Axes for Two-axis Cartesian Coordinates



Buffered

The multi-execution instruction remains in the buffer until the current operation is finished. The buffered instruction is executed after the operation for the current instruction is normally ended.



Blending

Blending for axes groups works in the same way as blending for single-axis operations. The buffered instruction remains in the buffer until the target position of the current instruction is reached. The buffered instruction is executed after the target position of the current instruction is reached. The axes do not stop at the target position. The two motions are blended together at the interpolation velocity specified with the *BufferMode* input variable.

The Interpolation Acceleration/Deceleration Over axes group parameter is used to select one of the following operations for when the acceleration/deceleration that is specified in the buffered instruction would exceed the target position.

- Use rapid acceleration/deceleration. (Blending is changed to Buffered.)
- Use rapid acceleration/deceleration.
- Minor fault stop (Treat blending operation as buffered operation.)

Precautions for Correct Use

For blending in multi-axes coordinated control, buffered operation is used if the results of profile processing shows that the execution time of the current instruction is less than four control periods. A Notice of Insufficient Travel Distance to Achieve Blending Transit Velocity observation will occur.

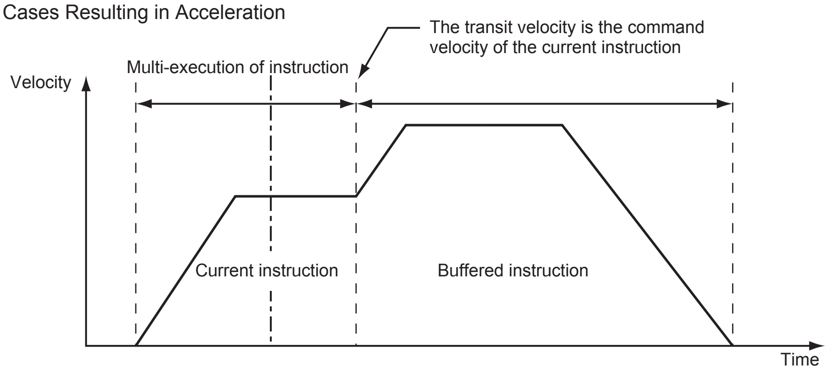
● Blending Low (Low Velocity)

Operation is performed using the target position of the current instruction and the target velocity that is the slower of the target velocities for the current instruction and buffered instruction.

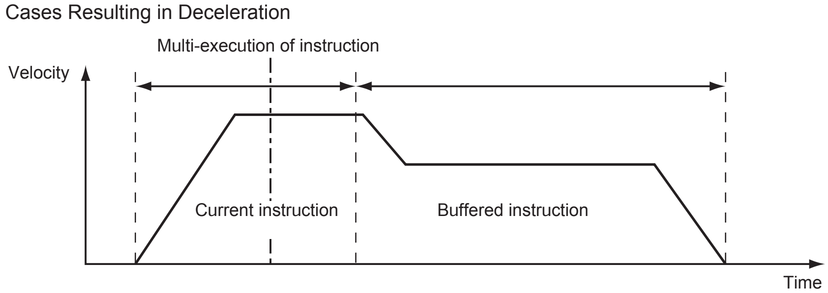
● Blending Previous (Previous Velocity)

Operation is performed with the target velocity of the current instruction until the target position of the current instruction is reached. Operation is performed after acceleration/deceleration to the target velocity of the buffered instruction once the target position is reached.

Cases Resulting in Acceleration



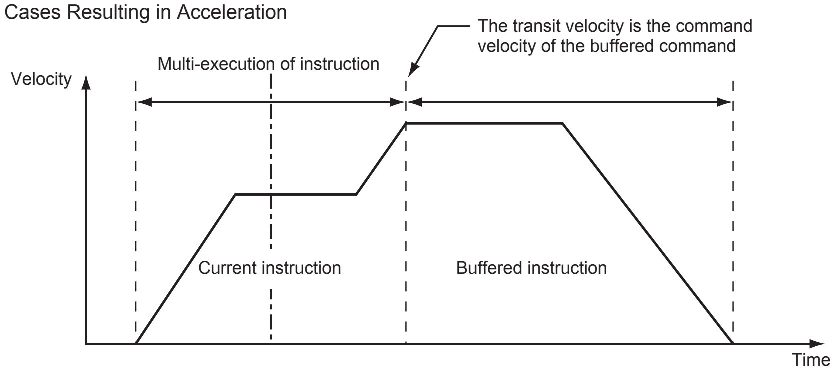
Cases Resulting in Deceleration



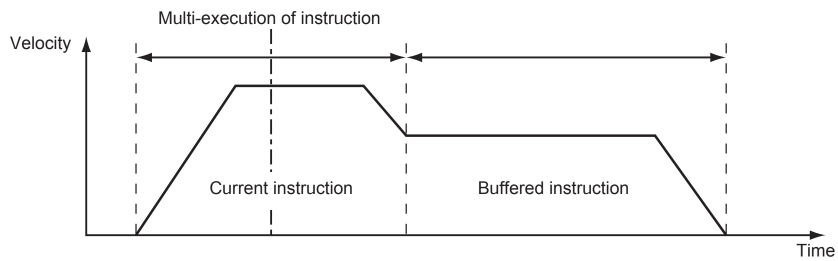
● Blending Next (Next Velocity)

Operation is performed using the target position of the current instruction and the target velocity of the buffered instruction.

Cases Resulting in Acceleration



Cases Resulting in Deceleration



● Blending High (High Velocity)

Operation is performed using the target position of the current instruction and the target velocity that is the faster of the target velocities for the current instruction and buffered instruction.

Transition Modes

Multi-execution of instructions for axes groups may create some shock on the device and/or workpiece due to changes in the direction of the interpolation path. You can specify the *TransitionMode* input variable to the motion control instruction to select a transition method to use between instructions in order to lessen this shock. You can choose from the following transition modes in the MC Function Module.

No.	Transition mode	Description
0	Transition Disabled (*_mcTMNone*)	Do not perform any processing for transitions (default). No attempt is made to lessen the shock, but this results in a shorter operation time.
10	Superimpose Corners (*_mcTMCornerSuperimposed*)	The deceleration of the current instruction is superimposed on the acceleration of the buffered instruction. You can keep the linear velocity of the interpolation path constant.

Additional Information

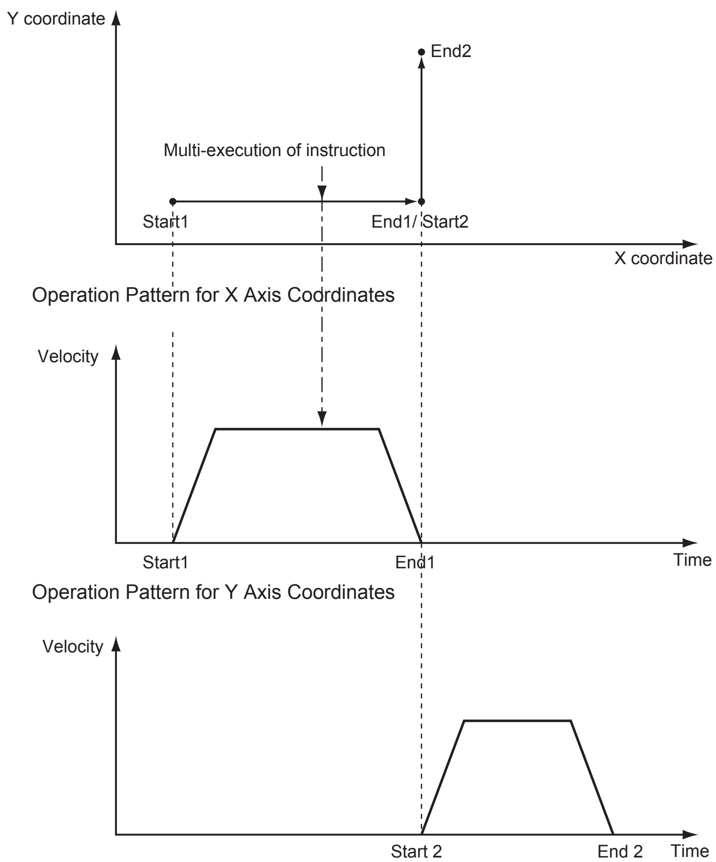
The PLCopen® technology specifications define numbers 0 through 9. Number 10 is unique to the MC Function Module.

● Transition Disabled (0: *_mcTMNone*)

No processing is performed to connect the two positions.

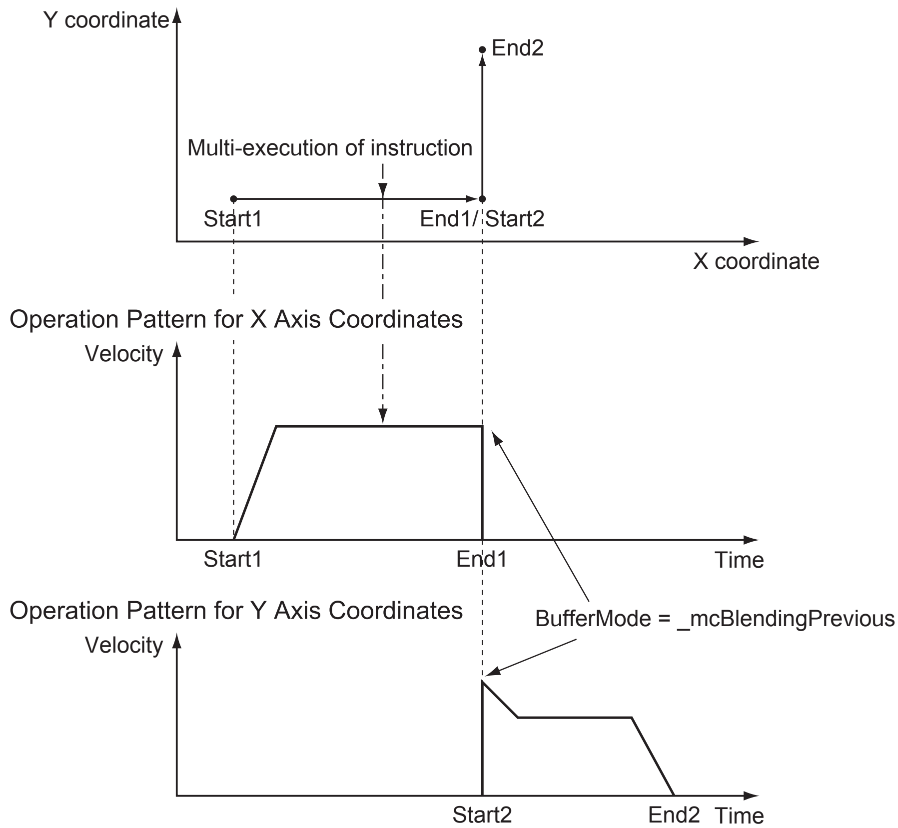
TransionMode = *_mcTMNone* and BufferMode = *_mcBuffered*

The axis moves to position End1, stops, and then moves to position End2.



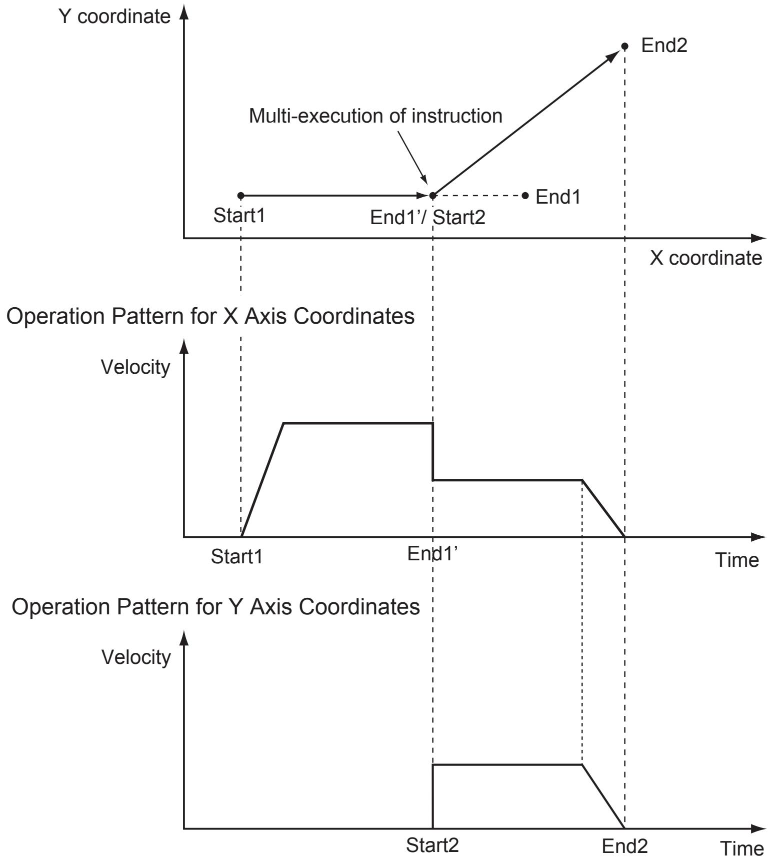
TransionMode = *_mcTMNone* and BufferMode = *_mcBlending*

The axis moves to position End1, and then moves to position End2.



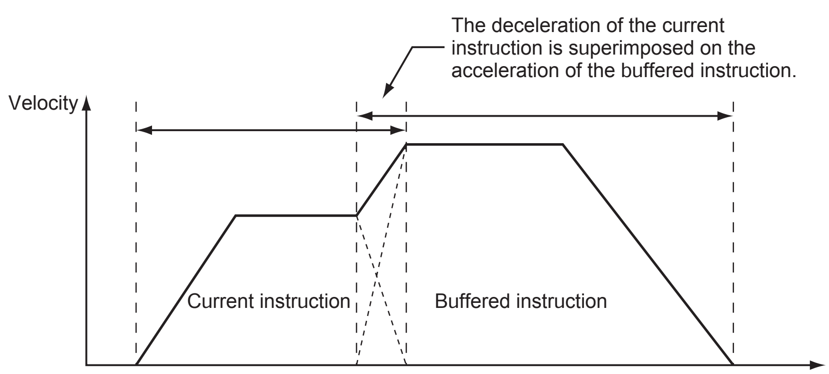
TransionMode = *_mcTMNone* and BufferMode = *_mcAborting*

The axis moves from End1’ (multi-execution of instruction) to End2.



● Superimpose Corners (10: *_mcTMCornerSuperimposed*)

The deceleration of the current instruction is superimposed on the acceleration of the buffered instruction. Operation is executed in the same amount of time as for the deceleration of the current instruction, no matter what is specified as the acceleration for the buffered instruction. The superimposed area will apply no jerk even if jerk is specified.



The output variable *Done*, which indicates the end of a motion control instruction, will change to TRUE for *_mcTMCornerSuperimposed* when the area of superimposition is completed.

Additional Information

The path linear velocity is constant if the following two conditions are met.

- The target velocities of the current instruction and the buffered instruction are the same.
- The deceleration rate of the current instruction and the acceleration rate of the buffered instruction are the same.

Combining Transition Modes and Multi-execution of Instructions

The following table shows the combinations of Transition Modes and Buffer Modes.

OK: Operation possible. ---: Generates an error and stops.

Buffer Mode / Transition Mode	Aborting	Buffered	Blending Low	Blending Previous	Blending Next	Blending High
Transition Disabled (*_mcTMNone*)	OK	OK	OK	OK	OK	OK
Superimpose Corners*1 (*_mcTMCornerSuperimposed*)	---	---	OK	OK	OK	OK

*1 For superimpose corners, the deceleration for the current instruction and the acceleration for the buffered instruction will be superimposed.

9-8 Other Functions

This section describes other functions of the MC Function Module.

Refer to the *NX-series Position Interface Units User's Manual* (Cat. No. W524) for the differences when you use NX-series Pulse Output Units.

9-8-1 Changing the Current Position

The command current position of a Servo axis can be changed to a specified value. The actual current position changes to a value that maintains the current following error with the command current position. For an encoder axis, you can change the actual current position. Use the MC_SetPosition instruction to specify the actual position you want to modify.

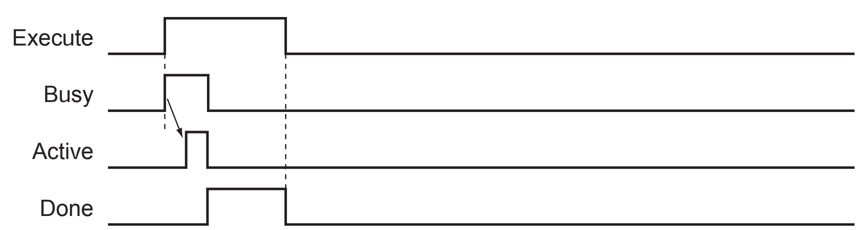
You can change the actual position even while an axis is in motion. If positioning to an absolute value is being executed, positioning will be performed to the target position using the new absolute coordinates. However, the travel distance will stay the same when you position to a relative value.

Precautions for Correct Use

- When the Count Mode is Rotary Mode, an instruction error will occur if you specify a position outside the ring counter range.
- After changing the current position the home will be undefined and you will not be able to use the following functions and instructions.

  Software limits
  High-speed homing
  Interpolation instructions (linear and circular interpolation)

● Timing Chart for Execution While Axis Is Stopped



Additional Information

You can change the actual position while home is defined by specifying a zero position preset for the MC_Home or MC_HomeWithParameter instruction.

For details on the MC_SetPosition instruction, refer to the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

9-8-2 Torque Limit

The output torque is limited by enabling and disabling the torque limit function of the Servo Drive and by setting the torque limit value.
Different limits can be specified for the positive torque limit and negative torque limit.

For details, refer to the MC_SetTorqueLimit instruction in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

Precautions for Correct Use

You cannot use the torque limit function for an NX-series Pulse Output Unit.

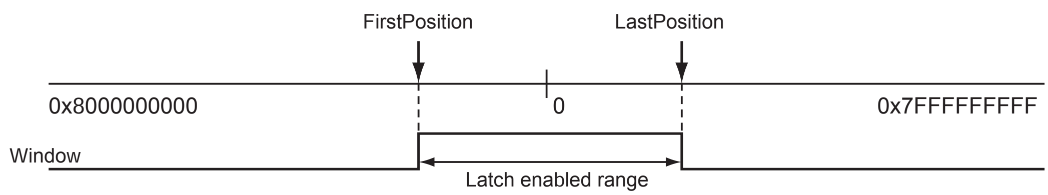
9-8-3 Latching

Latching is used to control positioning based on the position where a trigger signal occurs, such as a signal from a sensor input. The position of the axis is recorded (i.e., latched) when the trigger signal occurs. You can set up to two trigger signals for each axis. Use the MC_TouchProbe (Enable External Latch) instruction to specify the Trigger Input Condition variable, Window Only variable, and Stopping Mode Selection variable for the axis you want to latch. In addition to signals that connect to the Servo Drive, you can also specify variables in the user program to use as a trigger. Use the MC_AbortTrigger (Disable External Latch) instruction to abort latching. You can use latching only with a Servo Drive that support latching (touch probe), such as the G5-series Servo Drives, or a GX-EC0211/EC0241 Encoder Input Terminal.

Use *WindowOnly* to detect only trigger signals within a specific start point and end point. The following chart shows the ranges for different Count Modes.

● Linear Mode

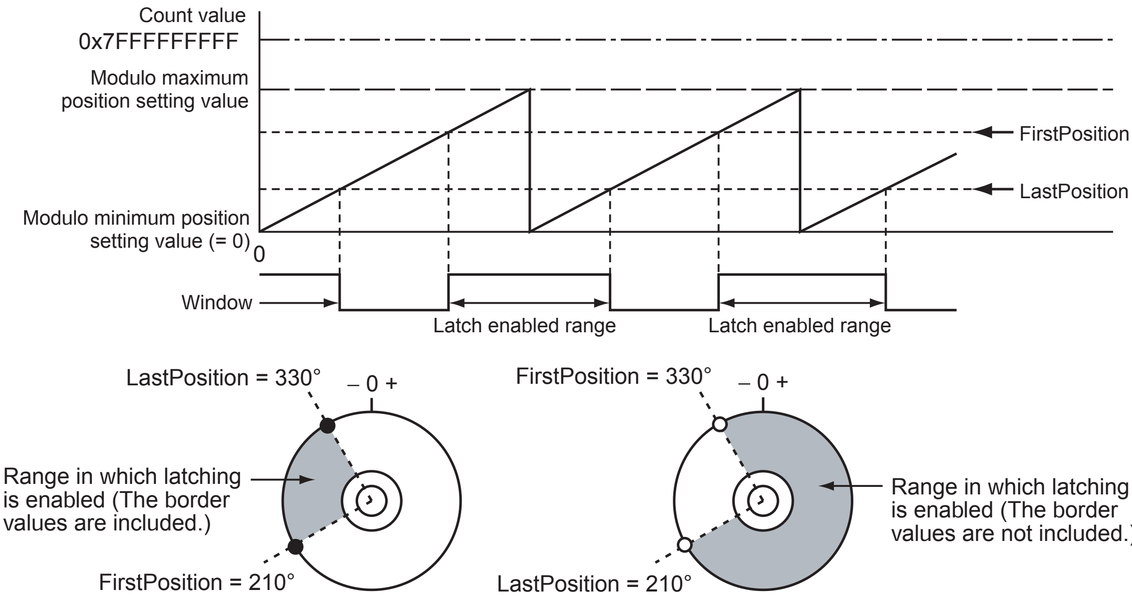
- The *FirstPosition* must be less than or equal to the *LastPosition*.
- An instruction error will occur if the *FirstPosition* is greater than the *LastPosition*.
- An instruction error will occur if a position beyond the position range of Linear Mode is specified.



● Rotary Mode

- The *FirstPosition* can be less than, equal to, or greater than the *LastPosition.* If the *FirstPosition* is greater than the *LastPosition*, the setting will straddle the modulo minimum position setting value.
- An instruction error will occur if a position beyond the upper and lower limits of the ring counter is specified.

	First Position ≤ Last Position	First Position > Last Position
Valid range	FirstPosition to LastPosition	LastPosition to FirstPosition



For details on latching, refer to the MC_TouchProbe (Enable External Latch) and MC_AbortTrigger (Disable External Latch) instructions in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

Refer to the *NX-series Position Interface Units User's Manual* (Cat. No. W524) for the differences when you use NX-series Pulse Output Units.

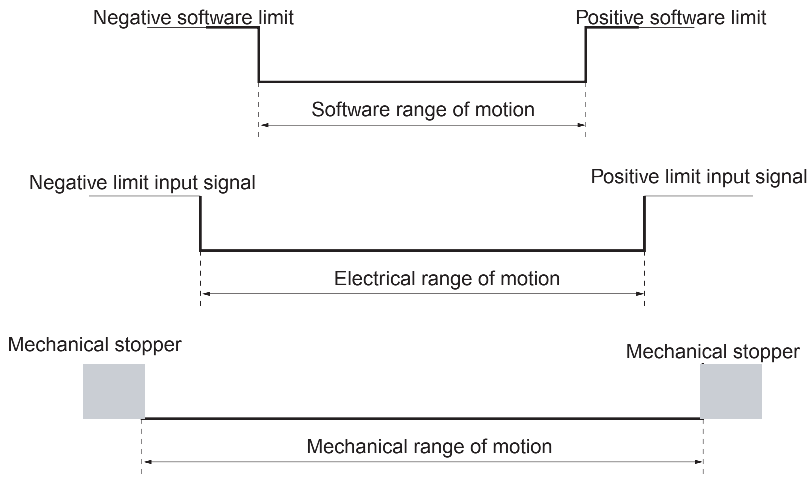
9-8-4 Zone Monitoring

This function detects whether the command position or actual position of an axis is in the specified range (zone). Use the MC_ZoneSwitch (Zone Monitor) instruction to specify the first position and last position of the zone to check. The *InZone* output variable for the Zone Monitor instruction will change to TRUE when the position of the axis enters the specified zone. You can also specify multiple zones for a single axis. Zones can overlap.

For details on zone monitoring, refer to the MC_ZoneSwitch (Zone Monitor) instruction in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

9-8-5 Software Limits

Actual positions can be monitored in the MC Function Module software. This function is separate from the hardware-based limit input signals. Set the range to monitor by setting the software limits in the Positive Software Limit and Negative Software Limit axis parameters. During normal positioning, motion is possible within the range of these software limits. Set software limits to prevent potential damage to machinery caused by mistakes in the user program or improper operation.



● Axis Parameters That Are Related to Software Limits

Parameter name	Function	Setting range	Default
Software Limits	Select the software limit function. 0: Disabled 1: Deceleration stop for command position*1 2: Immediate stop for command position 3: Deceleration stop for actual position*1 4: Immediate stop for actual position	0 to 4	0: Disabled
Positive Software Limit	Set the software limit in the positive direction. The unit is command units.	Long reals*2	2,147,483,647
Negative Software Limit	Set the software limit in the negative direction. The unit is command units.		−2,147,483,648

*1 If the actual position goes beyond a software limit during execution of a movement instruction that has a *Deceleration* input variable, the axis decelerates to a stop at the deceleration rate given by *Deceleration*. If the actual position goes beyond a software limit during execution of a movement instruction that does not have a *Deceleration* input variable, the axis decelerates to a stop at the maximum deceleration that is set in the axis parameters.

*2 Positions can be set within a 40-bit signed integer range when converted to pulses.

You can use the axis settings of the Sysmac Studio, the MC_Write (Write MC Setting) instruction, or the MC_WriteAxisParameter (Write Axis Parameters) instruction to set the above axis parameters.
If any setting values are changed for an axis or axes group in operation, those settings are enabled when the next operation begins.

Software limits function in the following two cases based on the axis operation state and the motion control instruction that is used.

● Executing Motion Instructions

- When the Actual Position Is within the Software Limits
  An instruction error will occur if the target position is outside the software limit range.

- When the Actual Position Is outside the Software Limits
  Motion is allowed only toward the software limit range. As long as the motion is toward the range, the target position does not need to be within the software limit range.

Precautions for Correct Use

Do not execute an instruction for an axis command for a target position that is outside of the software limit range.

● During Axis Motion

When the axis is in discrete motion, synchronized motion, continuous motion, or coordinated motion:

- An axis error will occur if the software limits are enabled for the command position and the command position leaves the range.
- An axis error will occur if the software limits are enabled for the actual position and the actual position leaves the range.

Additional Information

Software limits can be enabled when the Count Mode is set to Linear Mode and home is defined. Software limits are disabled in the following situations no matter what axis parameters have been set.

- When Count Mode is set to Rotary Mode.
- When home is not defined.
- During homing.

For details on the instruction to write the MC settings and the instruction to write the axis parameters, refer to the MC_Write instruction and MC_WriteAxisParameter instruction in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

9-8-6 Following Error Monitoring

Following error is the difference between the command position and the actual position of an axis. The MC Function Module monitors the following error every motion control period.
If the value of the following error exceeds the Following Error Over Value that is set in the axes parameters, Following Error Limit Exceeded minor fault level error occurs. If it exceeds the Following Error Warning Value, a Following Error Warning observation occurs. Monitoring the following error is disabled during execution of the holding operation for homing.

● Axis Parameters That Are Related to Monitoring the Following Error

You can set the check values for monitoring the following error by setting the appropriate axis parameters. Set the Following Error Warning Value so that it is less than the Following Error Over Value.
Set the axis parameters from the Sysmac Studio.

Parameter name	Function	Setting range	Default
Following Error Over Value	Set the excessive following error check value. Set 0 to disable the excessive following error check. (Unit: command units)	Non-negative long reals	0
Following Error Warning Value	Set the following error warning check value. Set 0 to disable the following error warning check. (Unit: command units)	Non-negative long reals that are less than or equal to the Following Error Over Value	0

Refer to the *NX-series Position Interface Units User's Manual* (Cat. No. W524) for the differences when you use NX-series Pulse Output Units.

9-8-7 Following Error Counter Reset

Resetting the following error counter resets the following error to 0.
Use the MC_ResetFollowingError instruction in the user program to reset the following error counter.
You can use the MC_ResetFollowingError instruction for each axis during positioning or during homing.
If you execute a following error counter reset while the axis is in motion, the current motion control instruction will be aborted and the command position will be set to the same value as the actual position.
The home will remain defined even after executing a following error counter reset.
For details on resetting the following error counter, refer to the MC_ResetFollowingError instruction in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

● Axis Parameters That Are Related to Resetting the Following Error Counter

You can choose to reset the following error counter on an immediate stop, on a limit input stop, or after homing is completed by setting the appropriate axis parameters. Set the axis parameters from the Sysmac Studio.

Parameter name	Function	Setting range	Default
Immediate Stop Input Stop Method	Set the stopping method in the MC Function Module when the immediate stop input is enabled. 0: Immediate stop 2: Immediate stop and error reset 3: Immediate stop and Servo OFF	0, 2, or 3	0
Limit Input Stop Method	Set the stopping method in the MC Function Module when the positive limit input or negative limit input is enabled. 0: Immediate stop 1: Deceleration stop 2: Immediate stop and error reset 3: Immediate stop and Servo OFF	0 to 3	0

Refer to the *NX-series Position Interface Units User's Manual* (Cat. No. W524) for the differences when you use NX-series Pulse Output Units.

9-8-8 Axis Following Error Monitoring

You can monitor the amount of following error for the command position or the actual position between two axes. Use the MC_AxesObserve (Monitor Axis Following Error) instruction to specify the permitted following error and the two axes to monitor. If the permitted following error is exceeded, the *Invalid* output variable for the Monitor Axis Following Error instruction will change to TRUE.

You can use this monitoring function to program the actions to take when the following error between axes grows too large for gantry control and other devices where both axes perform the same operation.

Precautions for Correct Use

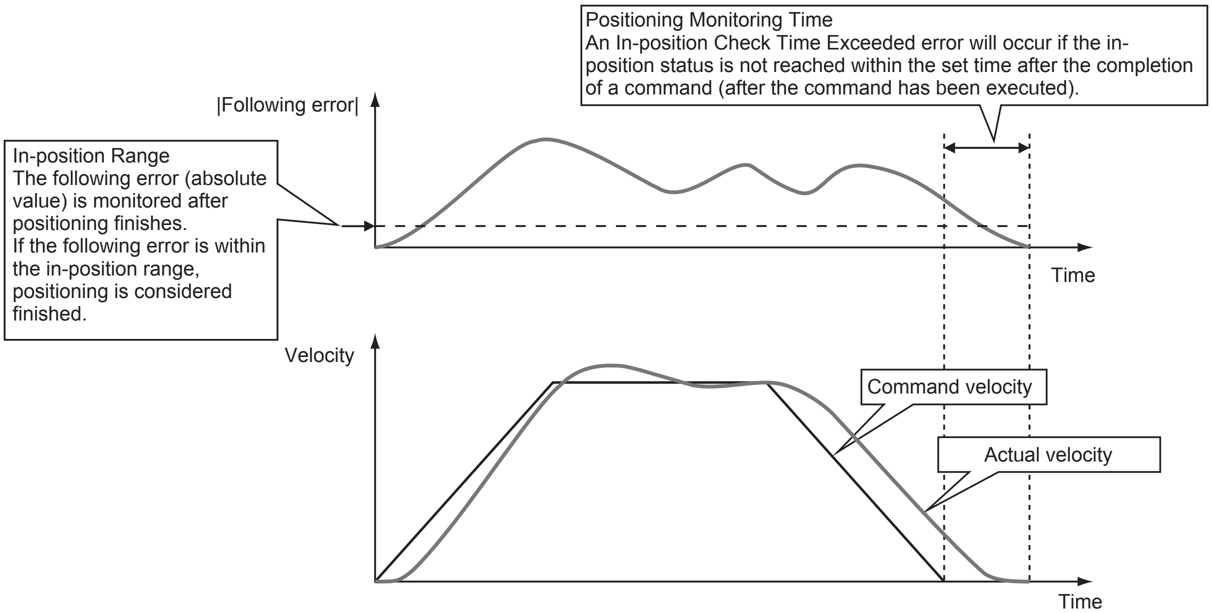
Even if the permitted following error between axes is exceeded, no error will occur in the MC Function Module. Check the *Invalid* output variable to stop axis operation or to take some other action as appropriate in the user program.

For details on axis following error monitoring, refer to the MC_AxesObserve (Monitor Axis Following Error) instruction in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

Refer to the *NX-series Position Interface Units User's Manual* (Cat. No. W524) for the differences when you use NX-series Pulse Output Units.

9-8-9 In-position Check

You can check to see if the actual current position has reached the specified range for the target position during positioning or homing. After command output of the target position is completed, positioning is considered to be finished when the difference between the target position and the actual current position is within the in-position range. An instruction error occurs if the position is not within the in-position within the in-position check time.



Axis Parameters That Are Related to In-position Checks

You can set the check conditions for the in-position check by setting the appropriate axis parameters. Set the in-position check time if you want to start any of the following operations only after confirming that axes are in position.

Parameter name	Function	Setting range	Default
In-position Range	Set the in-position width. (Unit: command units)	Non-negative long reals	10
In-position Check Time	Set the in-position check time in milliseconds. Set 0 to check for the end of positioning only when you define the home position during homing and not check positioning at other times. (Unit: ms)	0 to 10,000	0

Additional Information

- The in-position check is processed by the MC Function Module. The function in the Servo Drive is not used.
- Do not set an in-position check time if you want to start the next operation as quickly as possible without waiting for positioning to finish.

You can use the axis settings of the Sysmac Studio, the MC_Write (Write MC Setting) instruction, or the MC_WriteAxisParameter (Write Axis Parameters) instruction to set the above axis parameters.

Additional Information

The value set from the Sysmac Studio is restored if power to the CPU Unit is cycled or the user program is downloaded with the Synchronization menu command of the Sysmac Studio. Use the MC_Write (Write MC Setting) and MC_WriteAxisParameter (Write Axis Parameters) instructions only when you need to temporarily change the in-position check time.

● Monitor Information That Is Related to In-position Checks

You can read Axis Variables from the user program to monitor when positioning finishes.

Variable name	Data type	Meaning	Function
_MC_AX[0-63].Details.Idle	BOOL	Idle	TRUE when processing is not currently performed for the command value, except when waiting for in-position state.* *Idle* and *InPosWaiting* are mutually exclusive. They cannot both be TRUE at the same time.
_MC_AX[0-63].Details.InPosWaiting	BOOL	In-position Waiting	TRUE when waiting for in-position state. The in-position check is performed when positioning for the in-position check.

* This also includes states where processing is performed while in motion at velocity 0, during following error counter resets, during synchronized control, and during coordinated motion.

You can read Axes Group Variables from the user program to monitor when positioning finishes for the axes group.

Variable name	Data type	Meaning	Function
_MC_GRP[0-31].Details.Idle	BOOL	Idle	TRUE when processing is not currently performed for the command value, except when waiting for in-position state.*1 *Idle* and *InPosWaiting* are mutually exclusive. They cannot both be TRUE at the same time.
_MC_GRP[0-31].Details.InposWaiting	BOOL	In-position Waiting	TRUE when waiting for in-position state for any composition axis.*2 The in-position check is performed when positioning for the in-position check.

*1 This also includes states where processing is performed while in motion at a velocity of 0.

*2 This variable is FALSE when all composition axes in the axes group are within the in-position ranges set in the axis parameters.

For details on the instruction to write the MC settings and the instruction to write the axis parameters, refer to the MC_Write (Write MC Setting) and MC_WriteAxisParameter (Write Axis Parameters) instruction in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508).

Refer to the *NX-series Position Interface Units User's Manual* (Cat. No. W524) for the differences when you use NX-series Pulse Output Units.

9-8-10 Changing Axis Use

You can use the MC_ChangeAxisUse (Change Axis Use) instruction to temporarily change the setting of the Axis Use axis parameter. To change an axis in this way, it must be set as a *Used axis* or as an *Unused axis (changeable to used axis)* in the Axis Use axis parameter. If the Axis Use axis parameter is set to *Unused axis (changeable to used axis)* and the Axis Type parameter is set to a servo axis or virtual servo axis, you can set the axis in an axes group. A CPU Unit with unit version 1.04 or later and Sysmac Studio version 1.05 or higher are required.

Precautions for Correct Use

- Do not attempt to change an axis that is set to *Unused axis (unchangeable to used axis)* to a used axis.
- You cannot set an axis in an axes group if the Axis Use axis parameter is set to *Unused axis (unchangeable to used axis)*.

For details, refer to the MC_ChangeAxisUse instruction in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508-E1-05 or later).

For an application example of the MC_ChangeAxisUse instruction, refer to the *NJ-series CPU Unit Software Users Manual* (Cat. No. W501-E1-05 or later).

9-8-11 Enabling Digital Cam Switch

You can use the MC_DigitalCamSwitch (Enable Digital Cam Switch) instruction to turn the digital outputs ON or OFF according to the axis position.

The setting of the *ValueSource* input variable to the instruction also allows you to adjust for the acceleration or deceleration rate.

Always use this function together with the NX_AryDOutTimeStamp instruction and with a Digital Output Unit that supports time stamp refreshing. The NX_AryDOutTimeStamp instruction turns the specified digital outputs ON or OFF at specified timing of the time stamp.

A CPU Unit with unit version 1.06 or later and Sysmac Studio version 1.07 or higher are required to use this function.

Precautions for Correct Use

You can use this instruction for an axis that is assigned to an NX-series Position Interface Unit.

The NX Units that can be used are NX-EC0□□□ and NX-ECS□□□, also must be running the time stamping.

Refer to the MC_DigitalCamSwitch (Enable Digital Cam Switch) instruction in the *NJ-series Motion Control Instructions Reference Manual* (Cat. No. W508-E1-07 or later) for details on enabling digital cam switch.

Refer to the *NJ-series Instructions Reference Manual* (Cat. No. W502-E1-08 or later) for details on NX_AryDOutTimeStamp instruction.

Refer to the *NX-series Digital I/O Units User's Manual* (Cat. No. W521-E1-02 or later) for Digital Output Unit that supports time stamp refreshing.

Refer to the *NX-series Position Interface Units User's Manual* (Cat. No. W524-E1-02 or later) for time stamping and time stamps.

9-8-12 Displaying 3D Motion Monitor for User Coordinate System

In the case that coordinate systems (such as SCARA robot and vertical articulated robot) other than orthogonal coordinate system are implemented by user programs, this function can be used to display the path of robot hands, etc. in 3D with Sysmac Studio.

You can create an _sMC_POSITION_REF type user-defined variable and display in 3D Motion Monitor Display Mode.

A CPU Unit with unit version 1.06 or later and Sysmac Studio version 1.07 or higher are required to use this function.

● _sMC_POSITION_REF

The followings are the members of _sMC_POSITION_REF type data.

Member	Data type	Meaning
CommandPosition	ARRAY [0..5] OF LREAL	Command Current Position
ActualPosition	ARRAY [0..5] OF LREAL	Actual Current Position

The following list describes each member.

Member	Description
User-defined variable.CommandPosition[0]	This is an X-axis component for the command current position. This member is assigned a user-defined variable that indicates the X-axis position of the command current position generated by a user program.
User-defined variable.CommandPosition[1]	This is a Y-axis component for the command current position. This member is assigned a user-defined variable that indicates the Y-axis position of the command current position generated by a user program.
User-defined variable.CommandPosition[2]	This is a Z-axis component for the command current position. This member is assigned a user-defined variable that indicates the Z-axis position of the command current position generated by a user program.
User-defined variable.CommandPosition[3] to [5]	Not used.
User-defined variable.ActualPosition[0]	This is an X-axis component for the actual current position. This member is assigned a user-defined variable that indicates the X-axis position of the actual current position handled in a user program.
User-defined variable.ActualPosition[1]	This is a Y-axis component for the actual current position. This member is assigned a user-defined variable that indicates the Y-axis position of the actual current position handled in a user program.
User-defined variable.ActualPosition[2]	This is a Z-axis component for the actual current position. This member is assigned a user-defined variable that indicates the Z-axis position of the actual current position handled in a user program.
User-defined variable.ActualPosition[3] to [5]	Not used.

Each member is assigned a user-defined variable. The followings are the examples.

Name	Data type	Description
3D_position	_sMC_POSITION_REF	User-defined variable for 3D display
MCS_Cmd_TransX	LREAL	User-defined variable that indicates the X-axis position of the command current position generated by a user program
MCS_Cmd_TransY	LREAL	User-defined variable that indicates the Y-axis position of the command current position generated by a user program
MCS_Cmd_TransZ	LREAL	User-defined variable that indicates the Z-axis position of the command current position generated by a user program
MCS_Act_TransX	LREAL	User-defined variable that indicates the X-axis position of the actual current position handled in a user program
MCS_Act_TransY	LREAL	User-defined variable that indicates the Y-axis position of the actual current position handled in a user program
MCS_Act_TransZ	LREAL	User-defined variable that indicates the Z-axis position of the actual current position handled in a user program

```
3D_position.CommandPosition[0] := MCS_Cmd_TransX;
3D_position.CommandPosition[1] := MCS_Cmd_TransY;
3D_position.CommandPosition[2] := MCS_Cmd_TransZ;
3D_position.ActualPosition[0] := MCS_Act_TransX;
3D_position.ActualPosition[1] := MCS_Act_TransY;
3D_position.ActualPosition[2] := MCS_Act_TransZ;
```

● Overview of Operating Procedures

1. Create an _sMC_POSITION_REF type user-defined variable.
2. Create a program in which user-defined variables that indicate the command current position and actual current position for 3D display are assigned to each member of the created user-defined variable.
3. Select *Specified coordinate* in the *Type* Box in the 3D Machine Model List.

   The _sMC_POSITION_REF data type is displayed in the 3D Machine Model Parameter Settings section.
4. Set the created user-defined variable in the *Value* Column in the 3D Machine Model Parameter Settings section.
5. Execute the user program.
6. Start tracing the data with the data trace to sample the data.
7. Check the trace results on the Data Trace Tab Page.

Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for details on 3D Motion Monitor Display Mode.