OMRON

Sysmac Library

User's Manual
for Vibration Suppression Library
SYSMAC-XR006

SYSMAC
always in control

W550-E1-04

NOTE

Trademarks

Copyrights

Introduction

Thank you for purchasing an NJ/NX-series CPU Unit or an NY-series Industrial PC.

This manual provides information required to use the function blocks in the Vibration Suppression Library. (“Function block” is sometimes abbreviated as “FB.”) Please read this manual and make sure you understand the functionality and performance of the NJ/NX-series CPU Unit before you attempt to use it in a control system.

This manual contains the specifications of the Function Block. It does not include restrictions on use of the Controller, Units, or components, or restrictions due to combinations. Make sure to read the user's manual for each product before use.

Keep this manual in a safe place where it will be available for reference during operation.

Features of the Library

The Vibration Suppression Library is used to suppress residual vibration caused by the operation of machines. You can use this library together with motion control instructions of the NJ/NX/NY-series Controller.

Refer to the motion control instructions reference manual for details on motion control instructions of the NJ/NX/NY-series Controller.

Intended Audience

This manual is intended for the following personnel, who must also have knowledge of electrical systems (an electrical engineer or the equivalent).

- Personnel in charge of introducing FA systems.
- Personnel in charge of designing FA systems.
- Personnel in charge of installing and maintaining FA systems.
- Personnel in charge of managing FA systems and facilities.
- Personnel with knowledge of control logic.

For programming, this manual is intended for personnel who understand the programming language specifications in international standard IEC 61131-3 or Japanese standard JIS B 3503.

Applicable Products

For the model numbers and versions of an NJ/NX-series CPU Unit, NY-series Industrial PC, and the Sysmac Studio that this library supports, refer to Sysmac Library Version Information in the *SYSMAC-XR□□□ Sysmac Library Catalog* (Cat. No. P102). This catalog can be downloaded from the OMRON website (http://www.ia.omron.com/products/family/3459/download/catalog.html).

Manual Structure

Special Information

Special information in this manual is classified as follows:

Precautions for Safe Use

Precautions on what to do and what not to do to ensure safe usage of the product.

Precautions for Correct Use

Precautions on what to do and what not to do to ensure proper operation and performance.

Additional Information

Additional information to read as required.
This information is provided to increase understanding or make operation easier.

Version Information

Information on differences in specifications and functionality for CPU Units and Industrial PCs with different unit versions and for different versions of the Sysmac Studio are given.

Note References are provided to more detailed or related information.

CONTENTS

Terms and Conditions Agreement

Warranty, Limitations of Liability

Warranties

- **Exclusive Warranty**

  Omron's exclusive warranty is that the Products will be free from defects in materials and workmanship for a period of twelve months from the date of sale by Omron (or such other period expressed in writing by Omron). Omron disclaims all other warranties, express or implied.

- **Limitations**

  OMRON MAKES NO WARRANTY OR REPRESENTATION, EXPRESS OR IMPLIED, ABOUT NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OF THE PRODUCTS. BUYER ACKNOWLEDGES THAT IT ALONE HAS DETERMINED THAT THE PRODUCTS WILL SUITABLY MEET THE REQUIREMENTS OF THEIR INTENDED USE.

  Omron further disclaims all warranties and responsibility of any type for claims or expenses based on infringement by the Products or otherwise of any intellectual property right.

- **Buyer Remedy**

  Omron's sole obligation hereunder shall be, at Omron's election, to (i) replace (in the form originally shipped with Buyer responsible for labor charges for removal or replacement thereof) the non-complying Product, (ii) repair the non-complying Product, or (iii) repay or credit Buyer an amount equal to the purchase price of the non-complying Product; provided that in no event shall Omron be responsible for warranty, repair, indemnity or any other claims or expenses regarding the Products unless Omron's analysis confirms that the Products were properly handled, stored, installed and maintained and not subject to contamination, abuse, misuse or inappropriate modification. Return of any Products by Buyer must be approved in writing by Omron before shipment. Omron Companies shall not be liable for the suitability or unsuitability or the results from the use of Products in combination with any electrical or electronic components, circuits, system assemblies or any other materials or substances or environments. Any advice, recommendations or information given orally or in writing, are not to be construed as an amendment or addition to the above warranty.

See http://www.omron.com/global/ or contact your Omron representative for published information.

Limitation on Liability; Etc

OMRON COMPANIES SHALL NOT BE LIABLE FOR SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, LOSS OF PROFITS OR PRODUCTION OR COMMERCIAL LOSS IN ANY WAY CONNECTED WITH THE PRODUCTS, WHETHER SUCH CLAIM IS BASED IN CONTRACT, WARRANTY, NEGLIGENCE OR STRICT LIABILITY.

Further, in no event shall liability of Omron Companies exceed the individual price of the Product on which liability is asserted.

Application Considerations

Suitability of Use

Omron Companies shall not be responsible for conformity with any standards, codes or regulations which apply to the combination of the Product in the Buyer's application or use of the Product. At Buyer's request, Omron will provide applicable third party certification documents identifying ratings and limitations of use which apply to the Product. This information by itself is not sufficient for a complete determination of the suitability of the Product in combination with the end product, machine, system, or other application or use. Buyer shall be solely responsible for determining appropriateness of the particular Product with respect to Buyer's application, product or system. Buyer shall take application responsibility in all cases.

NEVER USE THE PRODUCT FOR AN APPLICATION INVOLVING SERIOUS RISK TO LIFE OR PROPERTY OR IN LARGE QUANTITIES WITHOUT ENSURING THAT THE SYSTEM AS A WHOLE HAS BEEN DESIGNED TO ADDRESS THE RISKS, AND THAT THE OMRON PRODUCT(S) IS PROPERLY RATED AND INSTALLED FOR THE INTENDED USE WITHIN THE OVERALL EQUIPMENT OR SYSTEM.

Programmable Products

Omron Companies shall not be responsible for the user's programming of a programmable Product, or any consequence thereof.

Disclaimers

Performance Data

Data presented in Omron Company websites, catalogs and other materials is provided as a guide for the user in determining suitability and does not constitute a warranty. It may represent the result of Omron's test conditions, and the user must correlate it to actual application requirements. Actual performance is subject to the Omron's Warranty and Limitations of Liability.

Change in Specifications

Product specifications and accessories may be changed at any time based on improvements and other reasons. It is our practice to change part numbers when published ratings or features are changed, or when significant construction changes are made. However, some specifications of the Product may be changed without any notice. When in doubt, special part numbers may be assigned to fix or establish key specifications for your application. Please consult with your Omron's representative at any time to confirm actual specifications of purchased Product.

Errors and Omissions

Information presented by Omron Companies has been checked and is believed to be accurate; however, no responsibility is assumed for clerical, typographical or proofreading errors or omissions.

Safety Precautions

Definition of Precautionary Information

The following notation is used in this user's manual to provide precautions required to ensure safe usage of an NJ/NX-series Controller and an NY-series Industrial PC.

The safety precautions that are provided are extremely important to safety. Always read and heed the information provided in all safety precautions.

The following notation is used.

⚠WARNING	Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury. Additionally, there may be severe property damage.
⚠ Caution	Indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury, or property damage.

Symbols

	The circle and slash symbol indicates operations that you must not do. The specific operation is shown in the circle and explained in text. This example indicates prohibiting disassembly.
	The triangle symbol indicates precautions (including warnings). The specific operation is shown in the triangle and explained in text. This example indicates a precaution for electric shock.
	The triangle symbol indicates precautions (including warnings). The specific operation is shown in the triangle and explained in text. This example indicates a general precaution.
	The filled circle symbol indicates operations that you must do. The specific operation is shown in the circle and explained in text. This example shows a general precaution for something that you must do.

Cautions

⚠ Caution

Read all related manuals carefully before you use this library.	
Emergency stop circuits, interlock circuits, limit circuits, and similar safety measures must be provided in external control circuits.	
Check the user program, data, and parameter settings for proper execution before you use them for actual operation.	
The Sysmac Library and manuals are assumed to be used by personnel that is given in Intended Audience in this manual. Otherwise, do not use them.	
Perform the test run by holding an emergency stop switch in hand or otherwise prepare for rapid motor operation in an application to control the motor. Also perform the test run by using the parameters for which the motor does not rapidly accelerate or decelerate before you gradually adjust the parameters.	
In an application of heating or cooling, perform the test run by using the parameters for which rapid temperature changes will not occur before you gradually adjust the parameters.	
You must confirm that the user program and parameter values are appropriate to the specifications and operation methods of the devices.	
The sample programming shows only the portion of a program that uses the function or function block from the library.	
When using actual devices, also program safety circuits, device interlocks, I/O with other devices, and other control procedures.	
Understand the contents of sample programming before you use the sample programming and create the user program.	
If you input incorrect values to the input parameters for the functions or function blocks in the Vibration Suppression Library, the machine may possibly broken or may cause injury. Completely check the input parameters are appropriate before you perform the test run or actual operation.	
In the VSConstVelProfile1, VSConstTimeProfile1, and MultiVSFilter1 function blocks, when the Count Mode is set to Rotary Mode for an axis in the MC Function Mudule, make sure to properly set the direction in the MC_SyncMoveAbsolute (Cyclic Synchronous Absolute Positioning) instruction. The axis may rotate in the reverse direction from the intended direction.	
If you use the MultiVSFilter1 function block and the TimeToMoveParam1 function together, use the MC_MoveRelative (Relative Positioning) instruction to generate a command position that is input to the MultiVSFilter1 function block. If you do not use this instruction, unintended operation may occur.	

Precautions for Safe Use

Operation

- Input appropriate values to the input parameters for the functions or function blocks in the Vibration Suppression Library. If inappropriate values are input, vibration of the devices may be amplified. Completely check the input parameter values before you perform actual operation.
- If you specify the resonance frequency in the functions or function blocks in the Vibration Suppression Library, input appropriate values and check the validity in the test run. When your perform the test run, make sure to gradually increase the command velocity or the velocity limit value.

Precautions for Correct Use

Using the Library

- When you use the library, functions or function blocks that are not described in the library manual may be displayed on the Sysmac Studio. Do not use functions or function blocks that are not described in the manual.
- You cannot change the source code of the functions or function blocks that are provided in the Sysmac Library.
- The multi-execution (buffer mode) cannot be performed in the Sysmac Library.
- Confirm the specifications of the MC_SyncMoveAbsolute (Cyclic Synchronous Absolute Positioning) instruction when you use the Vibration Suppression Library. Refer to the motion control instructions reference manual for details on the MC_SyncMoveAbsolute (Cyclic Synchronous Absolute Positioning) instruction.

Using Sample Programming

- Create a user program that will produce the intended device operation.
- Check the user program for proper execution before you use it for actual operation.

Operation

- Specify the input parameter values within the valid range.
- In the function or function block with an Enabled output variable, if the value of Enabled is FALSE, do not use the processing result of the function or function block as a command value to the control target.
- In the function block with Execute, do not perform re-execution by the same instance. The output value of the function block will return to the default value.

Related Manuals

The following are the manuals related to this manual. Use these manuals for reference.

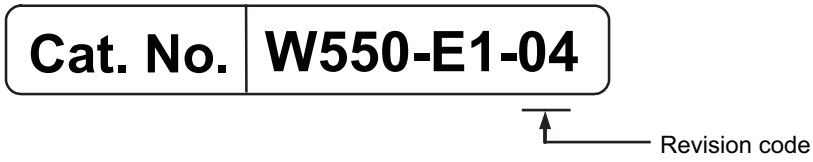
Manual name	Cat. No.	Model numbers	Application	Description
NX-series CPU Unit Hardware User's Manual	W535	NX701-□□□□	Learning the basic specifications of the NX-series NX701 CPU Units, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided	An introduction to the entire NX701 CPU Unit system is provided along with the following information on the CPU Unit. Features and system configuration Overview Part names and functions General specifications Installation and wiring Maintenance and inspection
NX-series NX102 CPU Unit Hardware User's Manual	W593	NX102-□□□□	Learning the basic specifications of the NX102 CPU Units, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided.	An introduction to the entire NX102 system is provided along with the following information on the CPU Unit. Features and system configuration Introduction Part names and functions General specifications Installation and wiring Maintenance and Inspection
NX-series NX1P2 CPU Unit Hardware User's Manual	W578	NX1P2-□□□□	Learning the basic specifications of the NX-series NX1P2 CPU Units, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided	An introduction to the entire NX1P2 CPU Unit system is provided along with the following information on the CPU Unit. Features and system configuration Overview Part names and functions General specifications Installation and wiring Maintenance and Inspection
NJ-series CPU Unit Hardware User's Manual	W500	NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning the basic specifications of the NJ-series CPU Units, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided	An introduction to the entire NJ-series system is provided along with the following information on the CPU Unit. Features and system configuration Overview Part names and functions General specifications Installation and wiring Maintenance and inspection
NY-series IPC Machine Controller Industrial Panel PC Hardware User's Manual	W557	NY532-□□□□	Learning the basic specifications of the NY-series Industrial Panel PCs, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided	An introduction to the entire NY-series system is provided along with the following information on the Industrial Panel PC. Features and system configuration Introduction Part names and functions General specifications Installation and wiring Maintenance and inspection

Manual name	Cat. No.	Model numbers	Application	Description
NY-series IPC Machine Controller Industrial Box PC Hardware User's Manual	W556	NY512-□□□□	Learning the basic specifications of the NY-series Industrial Box PCs, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided	An introduction to the entire NY-series system is provided along with the following information on the Industrial Box PC. Features and system configuration Introduction Part names and functions General specifications Installation and wiring Maintenance and inspection
NJ/NX-series CPU Unit Software User's Manual	W501	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning how to program and set up an NJ/NX-series CPU Unit. Mainly software information is provided	The following information is provided on a Controller built with an NJ/NX-series CPU Unit. CPU Unit operation CPU Unit features Initial settings Programming based on IEC 61131-3 language specifications
NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Software User's Manual	W558	NY532-□□□□ NY512-□□□□	Learning how to program and set up the Controller functions of an NY-series Industrial PC	The following information is provided on NY-series Machine Automation Control Software. Controller operation Controller features Controller settings Programming based on IEC 61131-3 language specifications
NJ/NX-series Instructions Reference Manual	W502	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning detailed specifications on the basic instructions of an NJ/NX-series CPU Unit	The instructions in the instruction set (IEC 61131-3 specifications) are described.
NY-series Instructions Reference Manual	W560	NY532-□□□□ NY512-□□□□	Learning detailed specifications on the basic instructions of an NY-series Industrial PC	The instructions in the instruction set (IEC 61131-3 specifications) are described.
NJ/NX-series CPU Unit Motion Control User's Manual	W507	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning about motion control settings and programming concepts of an NJ/NX-series CPU Unit.	The settings and operation of the CPU Unit and programming concepts for motion control are described.
NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Motion Control User's Manual	W559	NY532-□□□□ NY512-□□□□	Learning about motion control settings and programming concepts of an NY-series Industrial PC.	The settings and operation of the Controller and programming concepts for motion control are described.
NJ/NX-series Motion Control Instructions Reference Manual	W508	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning about the specifications of the motion control instructions of an NJ/NX-series CPU Unit.	The motion control instructions are described.
NY-series Motion Control Instructions Reference Manual	W561	NY532-□□□□ NY512-□□□□	Learning about the specifications of the motion control instructions of an NY-series Industrial PC.	The motion control instructions are described.
NJ/NY-series NC Integrated Controller User's Manual	O030	NJ501-5300 NY532-5400	Performing numerical control with NJ/NY-series Controllers.	Describes the functionality to perform the numerical control. Use this manual together with the *NJ/NY-series G code Instructions Reference Manual* (Cat. No. O031) when programming.

Manual name	Cat. No.	Model numbers	Application	Description
G code Instructions Reference Manual	O031	NJ501-5300 NY532-5400	Learning about the specifications of the G code/M code instructions.	The G code/M code instructions are described. Use this manual together with the *NJ/NY-series NC Integrated Controller User's Manual* (Cat. No. O030) when programming.
Sysmac Studio Version 1 Operation Manual	W504	SYSMAC -SE2□□□	Learning about the operating procedures and functions of the Sysmac Studio.	Describes the operating procedures of the Sysmac Studio.
CNC Operator Operation Manual	O032	SYSMAC -RTNC0□□□D	Learning an introduction of the CNC Operator and how to use it.	An introduction of the CNC Operator, installation procedures, basic operations, connection operations, and operating procedures for main functions are described.

Revision History

A manual revision code appears as a suffix to the catalog number on the front and back covers of the manual.



Revision code	Date	Revised content
01	December 2015	Original production
02	July 2016	Changed the manual name.
03	November 2016	Changed the manual name.
04	January 2019	Added compatible models.

Procedure to Use Sysmac Libraries

Procedure to Use Sysmac Libraries Installed Using the Installer

This section describes the procedure to use Sysmac Libraries that you installed using the installer.

There are two ways to use libraries.

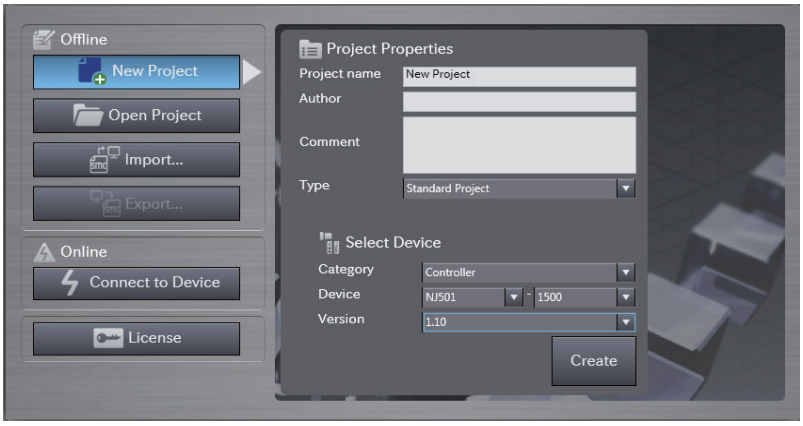
- Using newly installed Sysmac Libraries
- Using upgraded Sysmac Libraries

Version Information

To use Sysmac Libraries, you need the Sysmac Studio version 1.14 or higher.

Using Newly Installed Libraries

1 Start the Sysmac Studio and open or create a new project in which you want to use Sysmac Libraries.

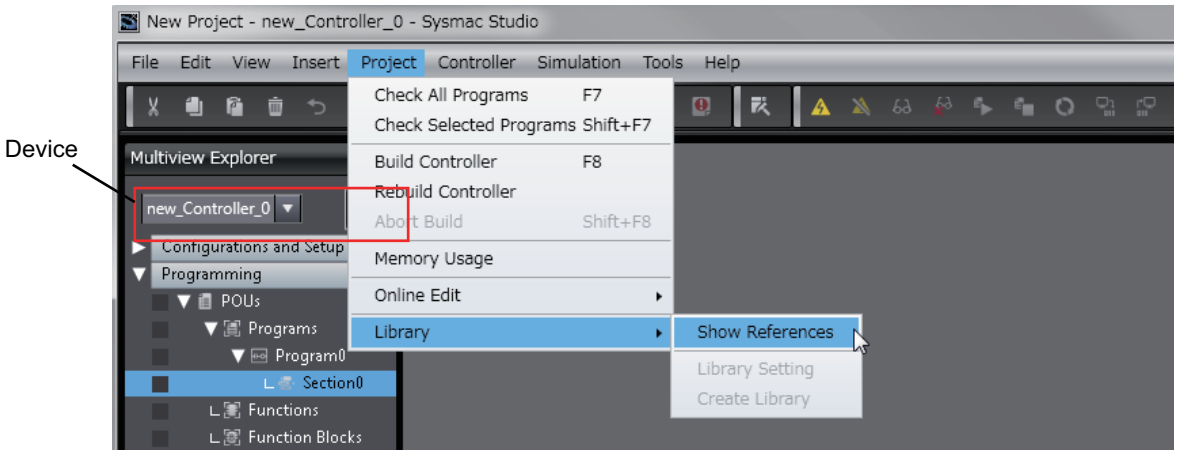


Precautions for Correct Use

If you create a new project, be sure to configure the settings as follows to enable the use of Sysmac Libraries. If you do not configure the following settings, you cannot proceed to the step 2 and later steps.

- Set the project type to Standard Project or Library Project.
- Set the device category to Controller.
- Set the device version to 1.01 or later.

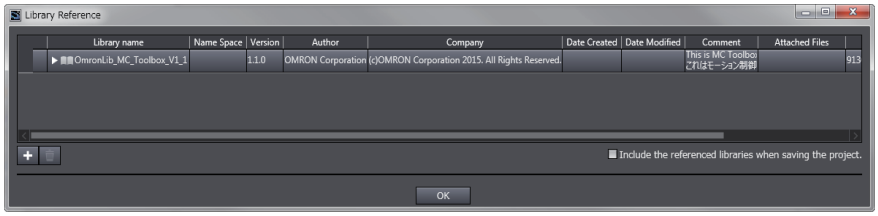
2 Select **Project** – **Library** – **Show References**.



Precautions for Correct Use

If you have more than one registered device in the project, make sure that the device selected currently is an NJ/NX-series CPU Unit or an NY-series Industrial PC. If you do not select an NJ/NX-series CPU Unit or an NY-series Industrial PC as the device, Library References does not appear in the above menu. When the device selected currently is an NJ/NX-series CPU Unit or an NY-series Industrial PC, the device icon is displayed in the Multiview Explorer.

3 Add the desired Sysmac Library to the list and click the **OK** Button.



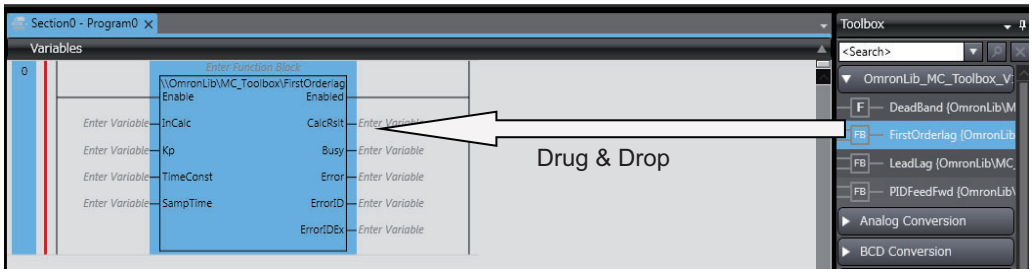
The Sysmac Library file is read into the project.

Now, when you select the Ladder Editor or ST Editor, the function blocks and functions included in a Sysmac Library appear in the Toolbox.

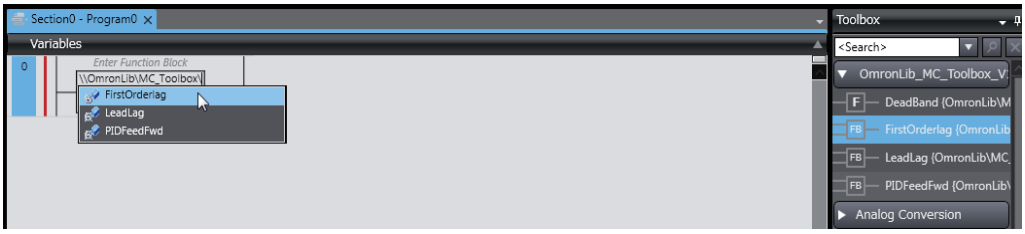
For the procedure for adding and setting libraries in the above screen, refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504).

4 Insert the Sysmac Library's function blocks and functions into the circuit using one of the following two methods.

- Select the desired function block or function in the Toolbox and drag and drop it onto the programming editor.



- Right-click the programming editor, select **Insert Function Block** in the menu, and enter the fully qualified name (\\name of namespace\name of function block).



Precautions for Correct Use

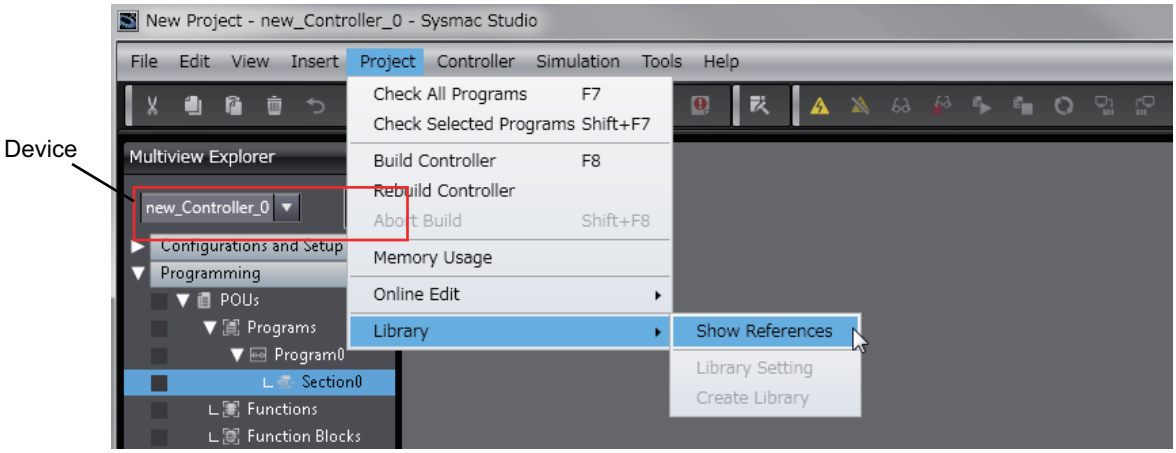
After you upgrade the Sysmac Studio, check all programs and make sure that there is no error of the program check results on the Build Tab Page.

Select **Project** – **Check All Programs** from the Main Menu.

Using Upgraded Libraries

1 Start the Sysmac Studio and open a project in which any old-version Sysmac Library is included.

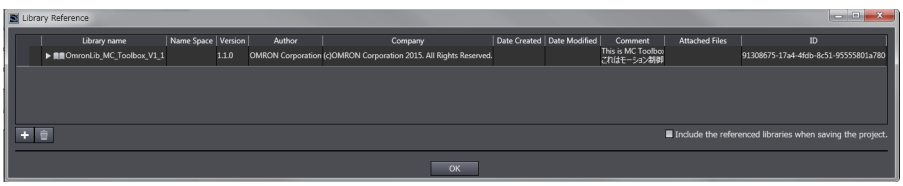
2 Select **Project** – **Library** – **Show References**.



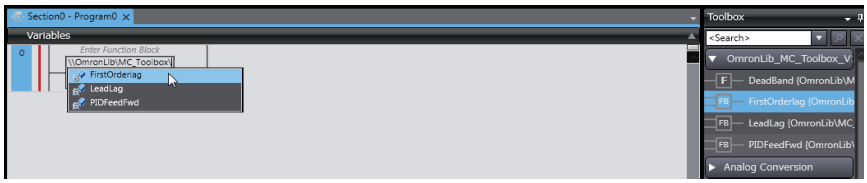
Precautions for Correct Use

If you have more than one registered device in the project, make sure that the device selected currently is an NJ/NX-series CPU Unit or an NY-series Industrial PC. Otherwise, Library References does not appear in the above menu. When the device selected currently is an NJ/NX-series CPU Unit or an NY-series Industrial PC, the device icon is displayed in the Multiview Explorer.

3 Select an old-version Sysmac Library and click the **Delete Reference** Button.



4 Add the desired Sysmac Library to the list and click the **OK** Button.



Procedure to Use Sysmac Libraries Uploaded from a CPU Unit or an Industrial PC

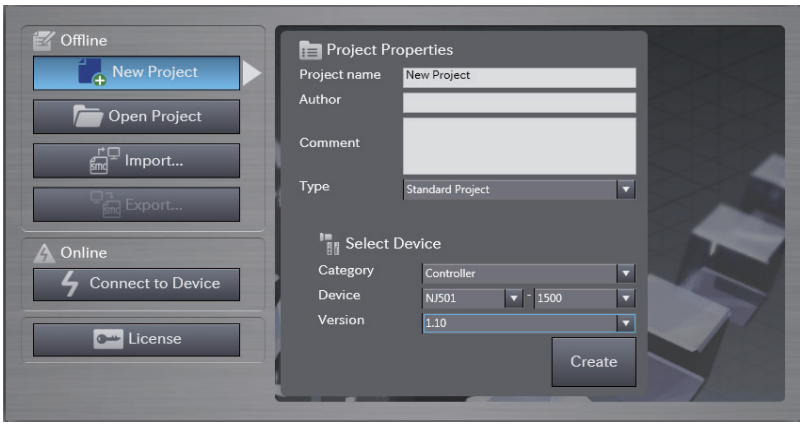
You can use Sysmac Libraries uploaded from a CPU Unit or an Industrial PC to your computer if they are not installed.

The procedure to use uploaded Sysmac Libraries from a CPU Unit or an Industrial PC is as follows.

Version Information

To use Sysmac Libraries, you need the Sysmac Studio version 1.14 or higher.

1 Start the Sysmac Studio and create a new project in which you want to use Sysmac Libraries.



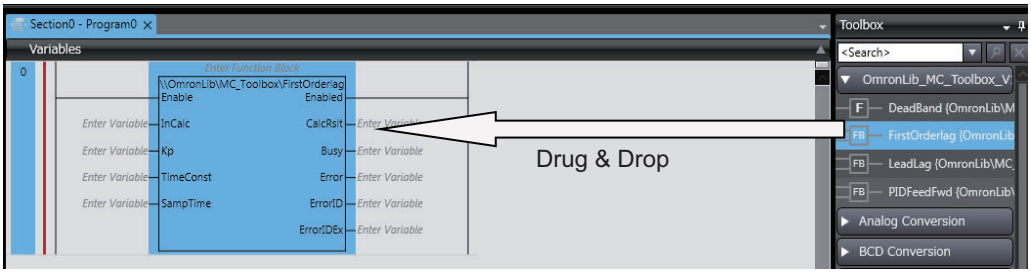
2 Connect the computer to the CPU Unit or the Industrial PC and place it online.

3 Upload POUs in which any Sysmac Library is used to the computer.

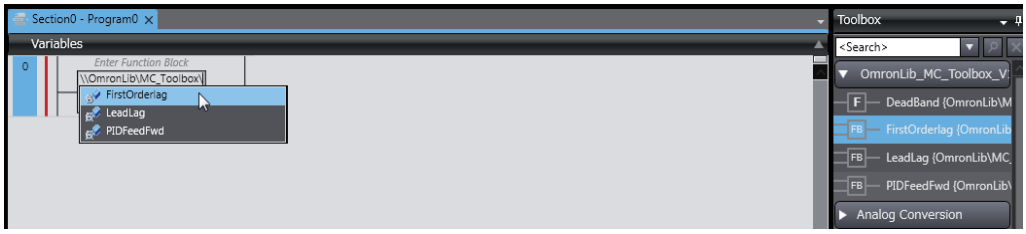
Now, when you select the Ladder Editor or ST Editor, the function blocks and functions included in the Sysmac Library used in the uploaded POUs appear in the Toolbox.

4 Insert the Sysmac Library's function blocks and functions into the circuit using one of the following two methods.

- Select the desired function block or function in the Toolbox and drag and drop it onto the Ladder Editor.



- Right-click the programming editor, select **Insert Function Block** in the menu, and enter the fully qualified name (\\name of namespace\name of function block).



Precautions for Correct Use

- The Sysmac Studio installs library files of the uploaded Sysmac Stutio to the specified folder on the computer if they are not present. However, the Sysmac Studio does not install library files to the specified folder on the computer if they are present.

  The specified folder here means the folder in which library files are installed by the installer.
- Note that uploading Sysmac Libraries from a CPU Unit or an Industrial PC does not install the manual and help files for the Sysmac Libraries, unlike the case where you install then using the installer. Please install the manual and help files using the installer if you need them.

Common Specifications of Function Blocks

Common Variables

This section describes the specifications of variables (*EN*, *Execute*, *Enable*, *Abort*, *ENO*, *Done*, *CalcRslt*, *Enabled*, *Busy*, *CommandAborted*, *Error*, *ErrorID*, and *ErrorIDEx*) that are used for more than one function or function block. The specifications are described separately for functions, for execute-type function blocks, and for enable-type function blocks.

Definition of Input Variables and Output Variables

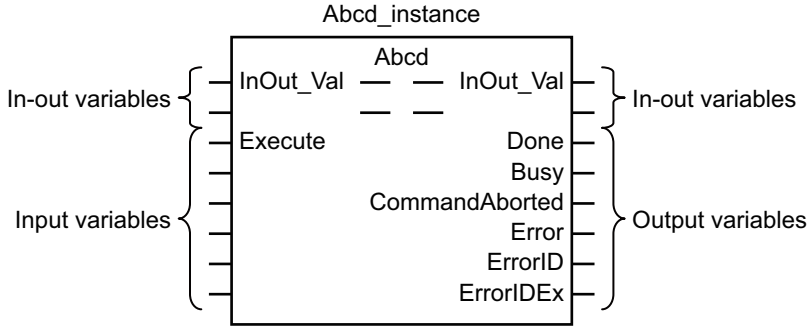
Common input variables and output variables used in functions and function blocks are as follows.

Variable	I/O	Data type	Function/function block type to use			Meaning	Definition
			Function block		Function		
			Execute-type	Enable-type			
EN	Input	BOOL			OK	Execute	The processing is executed while the variable is TRUE.
Execute			OK			Execute	The processing is executed when the variable changes to TRUE.
Enable				OK		Run	The processing is executed while the variable is TRUE.
Abort		BOOL	OK			Abort	The processing is aborted. You can select the aborting method.

Variable	I/O	Data type	Function/function block type to use			Meaning	Definition
			Function block		Function		
			Execute-type	Enable-type			
ENO	Output	BOOL			OK	Done	The variable changes to TRUE when the processing ends normally. It is FALSE when the processing ends in an error, the processing is in progress, or the execution condition is not met.
Done		BOOL	OK			Done	The variable changes to TRUE when the processing ends normally. It is FALSE when the processing ends in an error, the processing is in progress, or the execution condition is not met.
Busy		BOOL	OK	OK		Executing	The variable is TRUE when the processing is in progress. It is FALSE when the processing is not in progress.
CalcRslt		LREAL		OK		Calculation Result	The calculation result is output.
Enabled		BOOL		OK		Enabled	The variable is TRUE when the output is enabled. It is used to calculate the control amount for motion control, temperature control, etc.
Command Aborted		BOOL	OK			Command Aborted	The variable changes to TRUE when the processing is aborted. It changes to FALSE when the processing is re-executed the next time.
Error		BOOL	OK	OK		Error	This variable is TRUE while there is an error. It is FALSE when the processing ends normally, the processing is in progress, or the execution condition is not met.
ErrorID		WORD	OK	OK		Error Code	An error code is output.
ErrorIDEx		DWORD	OK	OK		Expansion Error Code	An expansion error code is output.

Execute-type Function Blocks

- Processing starts when *Execute* changes to TRUE.
- When *Execute* changes to TRUE, *Busy* also changes to TRUE. When processing is completed normally, *Busy* changes to FALSE and *Done* changes to TRUE.
- When continously executes the function blocks of the same instance, change the next *Execute* to TRUE for at least one task period after *Done* changes to FALSE in the previous execution.
- If the function block has a *CommandAborted* (Instruction Aborted) output variable and processing is aborted, *CommandAborted* changes to TRUE and *Busy* changes to FALSE.
- If an error occurs in the function block, *Error* changes to TRUE and *Busy* changes to FALSE.
- For function blocks that output the result of calculation for motion control and temperature control, you can use the BOOL input variable *Abort* to abort the processing of a function block. When *Abort* changes to TRUE, *CommandAborted* changes to TRUE and the execution of the function block is aborted.

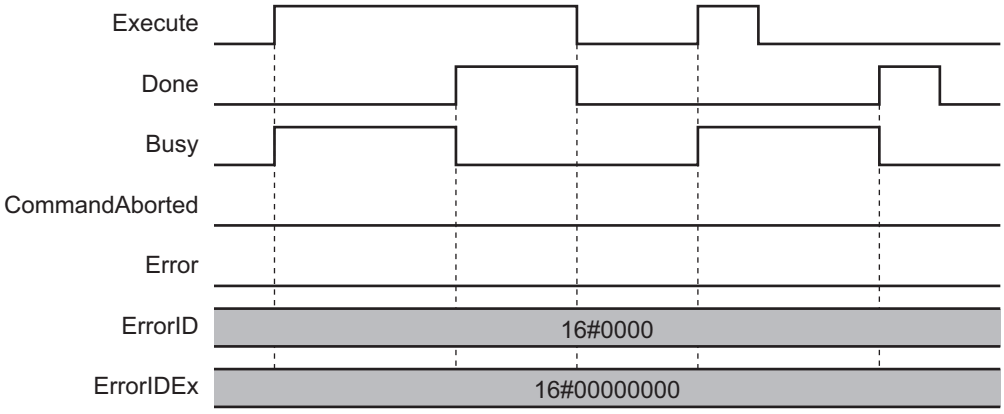


- If *Execute* is TRUE and *Done*, *CommandAborted*, or *Error* changes to TRUE, *Done*, *Command-Aborted*, and *Error* changes to FALSE when *Execute* is changed to FALSE.
- If *Execute* is FALSE and *Done*, *CommandAborted*, or *Error* changes to TRUE, *Done*,*Command-Aborted*, and *Error* changes to TRUE for only one task period.
- If an error occurs, the relevant error code and expansion error code are set in *ErrorID* (Error Code) and *ErrorIDEx* (Expansion Error Code). The error codes are retained even after *Error* changes to FALSE, but *ErrorID* is set to 16#0000 and *ErrorIDEx* is set to 16#0000 0000 when *Execute* changes to TRUE.

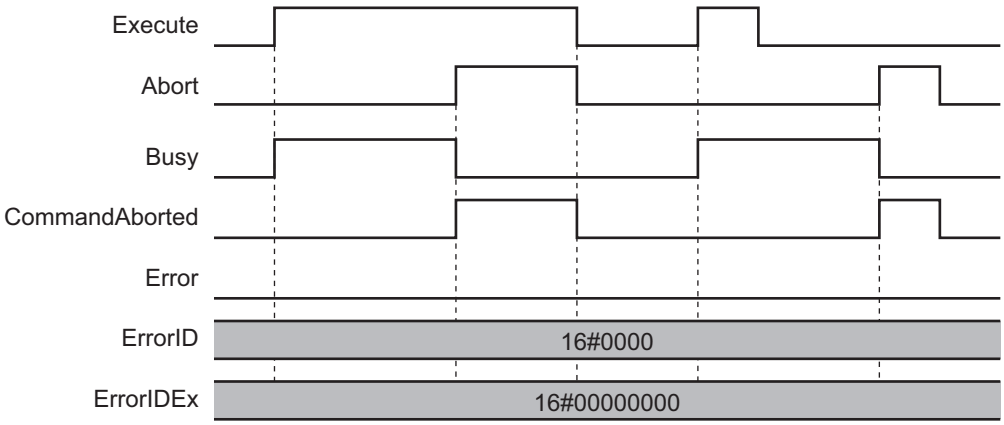
Timing Charts

This section provides timing charts for a normal end, aborted execution, and errors.

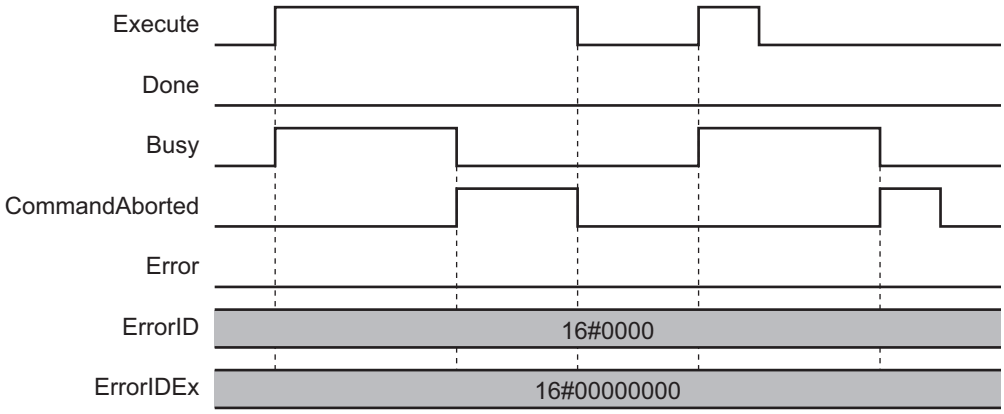
● Normal End



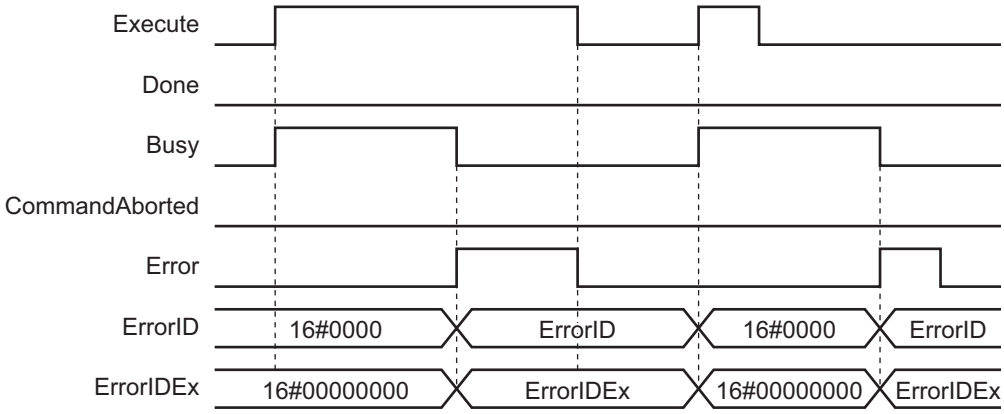
● Canceled Execution



Aborted Execution

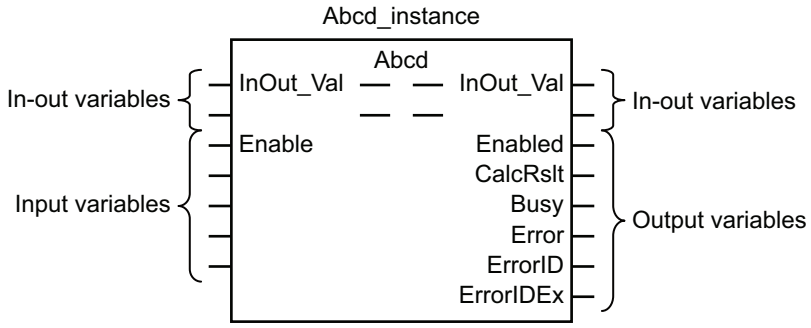


Errors



Enable-type Function Blocks

- Processing is executed while *Enable* is TRUE.
- When *Enable* changes to TRUE, *Busy* also changes to TRUE. *Enabled* is TRUE during calculation of the output value.
- If an error occurs in the function block, *Error* changes to TRUE and *Busy* and *Enabled* change to FALSE. When *Enable* changes to FALSE, *Enabled*, *Busy*, and *Error* change to FALSE.

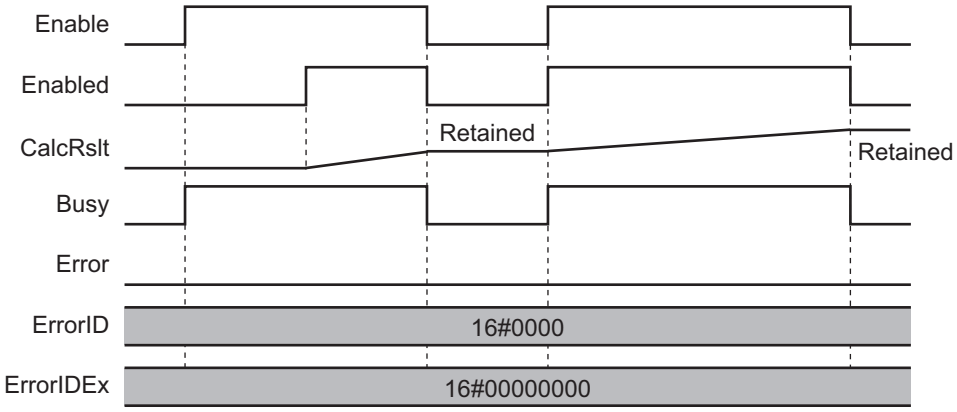


- If an error occurs, the relevant error code and expansion error code are set in *ErrorID* (Error Code) and *ErrorIDEx* (Expansion Error Code). The error codes are retained even after *Error* changes to FALSE, but *ErrorID* is set to 16#0000 and *ErrorIDEx* is set to 16#0000 0000 when *Enable* changes to TRUE.
- For function blocks that calculate the control amount for motion control, temperature control, etc., *Enabled* is FALSE when the value of *CalcRslt* (Calculation Result) is incorrect. In such a case, do not use *CalcRslt*. In addition, after the function block ends normally or after an error occurs, the value of *CalcRslt* is retained until *Enable* changes to TRUE. The control amount will be calculated based on the retained *CalcRslt* value, if it is the same instance of the function block that changed *Enable* to TRUE. If it is a different instance of the function block, the control amount will be calculated based on the initial value.

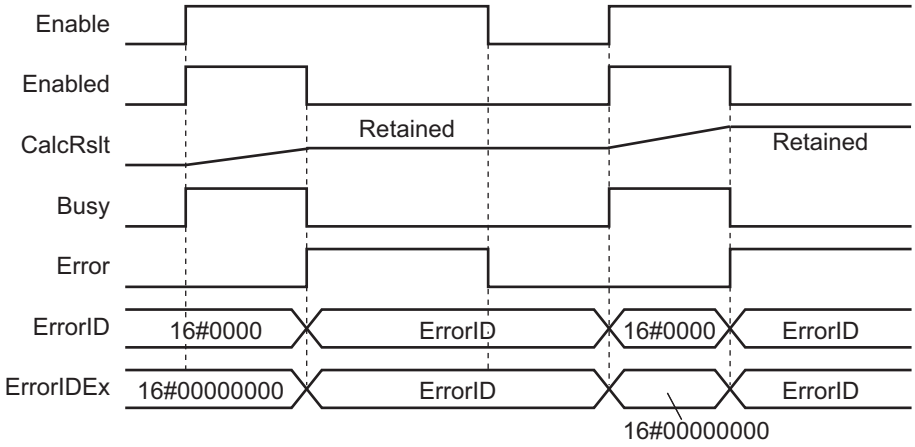
Timing Charts

This section provides timing charts for a normal end and errors.

● Normal End



● Errors


Enable
Enabled
CalcRslt
Busy
Error
ErrorID
ErrorIDEx
Retained
Retained
16#0000
ErrorID
16#0000
ErrorID
16#00000000
ErrorID
ErrorID
16#00000000

Precautions

This section provides precautions for the use of this function block.

Nesting

You can nest calls to this function block for up to four levels.

For details on nesting, refer to the software user's manual.

Instruction Options

You cannot use the upward differentiation option for this function block.

Re-execution of Function Blocks

Execute-type function blocks cannot be re-executed by the same instance.

If you do so, the output value will be the initial value.

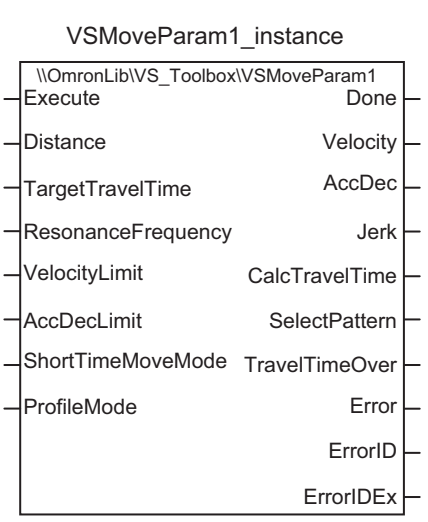
For details on re-execution, refer to the motion control user's manual.

Individual Specifications of Function Blocks

Function block name	Name	Page
VSMoveParam1	Resonance Frequency Specific Vibration Suppression Parameter Calculation 1	P.34
MultiVSFilter1	Multiple Frequency VS Filter 1	P.48
VSConstTimeProfile1	Time Specific Vibration Suppression Profile 1	P.65
VSConstVelProfile1	Constant Velocity Specific VS Profile 1	P.84
TimeToMoveParam1	Time Specific Positioning Parameter Calculation 1	P.99

VSMoveParam1

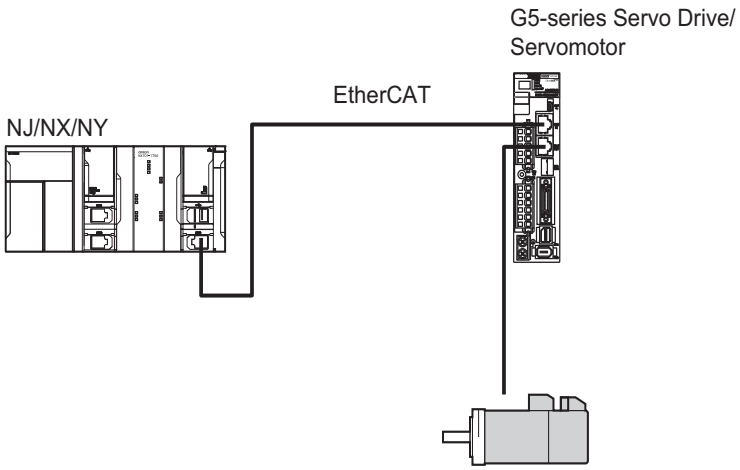
The VSMoveParam1 function block calculates the S-curve (i.e., velocity, acceleration, and jerk) parameters for suppressing the vibration that occurs during single axis positioning.

Function block name	Name	FB/ FUN	Graphic expression	ST expression
VSMoveParam1	Resonance Frequency Specific Vibration Suppression Parameter Calculation 1	FB		VSMoveParam1(Execute:=, Distance:=, TargetTravelTime:=, ResonanceFrequency:=, VelocityLimit:=, AccDecLimit:=, ShortTimeMoveMode:=, ProfileMode:=, Done=>, Velocity=>, AccDec=>, Jerk=>, CalcTravelTime=>, SelectPattern=>, TravelTimeOver=>, Error=>, ErrorID=>; ErrorIDEx=>,)

Function Block and Function Information

Item	Description
Library file name	OmronLib_VS_Toolbox_V1_0.slr
Namespace	OmronLib\VS_Toolbox
Function block and function number	00028
Source code published/not published	Not published
Function block and function version	1.00

Hardware Configuration Diagram



Variables

Input Variables

Name	Meaning	Data type	Initial value	Valid range	Description
Execute	Execute	BOOL	FALSE	TRUE or FALSE	Executes the function block when the value is changed to TRUE.
Distance	Travel Distance	LREAL	0	Depends on data type.	Sets the travel distance. The unit is [command unit]. Inputs the relative distance from the current position.
TargetTravel-Time	Target Travel Time	LREAL	0	0 ≤ Set value ≤100,000	Sets the target travel time. The unit is [ms]. Any values below 0.001 ms (1 us) are truncated.
Resonance Frequency	Resonance Frequency	LREAL	0	0.5 ≤ Set value ≤ 200	Set the resonance frequency. The unit is [Hz].*1
VelocityLimit	Velocity Limit	LREAL	0	Positive number	Sets the velocity limit. The unit is [command unit/s].*2
AccDecLimit	Acceleration/Deceleration Limit	LREAL	0	Positive number	Specifies the acceleration/deceleration limit. The unit is [command unit/s^2].*3
ShortTime MoveMode	Short Time Travel Mode	BOOL	FALSE	TRUE or FALSE	Enables or disables the short time travel mode.
ProfileMode	Travel Time Preference Mode	BOOL	FALSE	TRUE or FALSE	Enables or disables the travel time preference mode. If an S-curve cannot be created with *TargetTravelTime* when this mode is enabled, the damping control is disabled and an S-curve is created with the travel time of *TargetTravelTime*.

*1. Always set the resonance frequency. If you do not set it, the Resonance Frequency Setting Out of Range (*ErrorIDEx*=16#00000003) will occur.

*2. Always set the velocity limit. If you do not set it, the Velocity Limit Setting Out of Range (*ErrorIDEx*=16#00000004) will occur.

*3. Always set the acceleration/deceleration limit. If you do not set it, the Acceleration/Deceleration Limit Setting Out of Range (*ErrorIDEx*=16#00000005) will occur.

Output Variables

Name	Meaning	Data type	Valid range	Unit	Description
Done	Done	BOOL	TRUE or FALSE	---	TRUE when the function block execution is completed.
Velocity	Velocity	LREAL	Depends on data types.	Command unit/s	Gives the velocity which will be the input to the motion function block.
AccDec	Acceleration/Deceleration	LREAL	Depends on data types.	Command unit/s^2	Gives the acceleration which will be the input to the motion function block.
Jerk	Jerk	LREAL	Depends on data types.	Command unit/s^3	Gives the jerk which will be the input to the motion function block.
CalcTravelTime	Calculated Travel Time	LREAL	Depends on data types.	ms	Gives the travel time found from the created S-curve. Any values below 0.001 ms (1 us) are truncated.
SelectPattern	S-curve Pattern	UINT	0 ≤ Set value ≤ 10	---	Gives the selected S-curve pattern. 0: Not selected 1: Vibration suppression level 1 2: Vibration suppression level 2 3: Vibration suppression level 3 10: Travel time preference
TravelTimeOver	Travel Time Exceeded	BOOL	TRUE or FALSE	---	TRUE when the value of the *CalcTravelTime* output variable is greater than the value of the *TargetTravelTime* input variable.
Error	Error	BOOL	TRUE or FALSE	---	TRUE while there is an error.
ErrorID	Error Code	WORD	Depends on data types.	---	Contains the error code when an error occurs. The code 16#0000 indicates normal execution.
ErrorIDEx	Expansion Error Code	DWORD	Depends on data types.	---	Contains the expansion error code when an error occurs. The code 16#00000000 indicates normal execution.

Additional Information

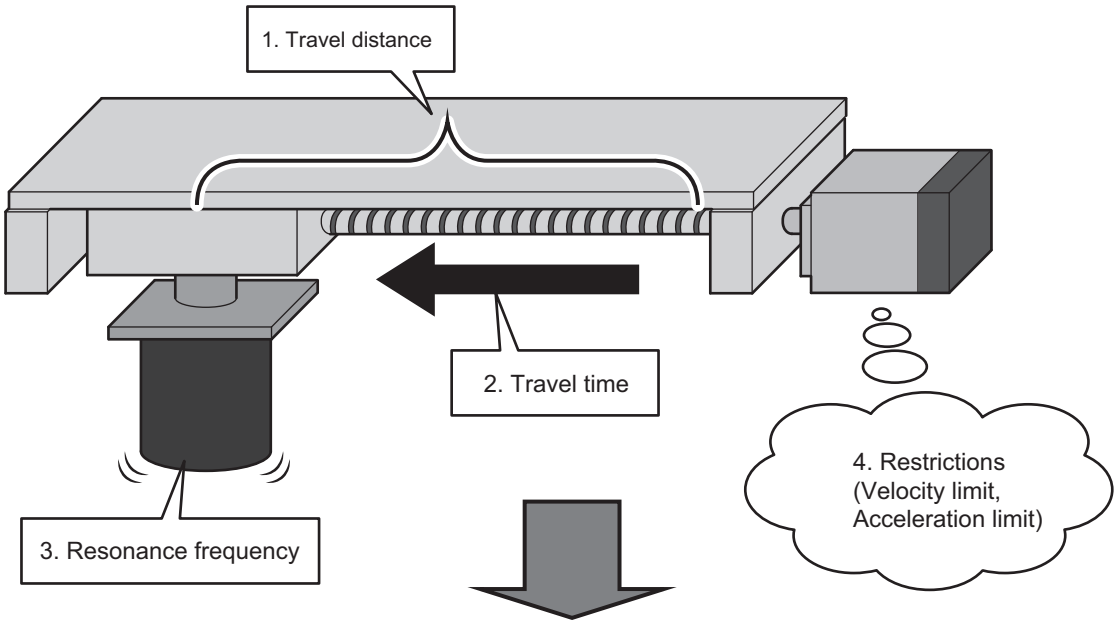
The command unit is indicated either in [mm], [µm], [nm], [degree], [inch], or [pulse].

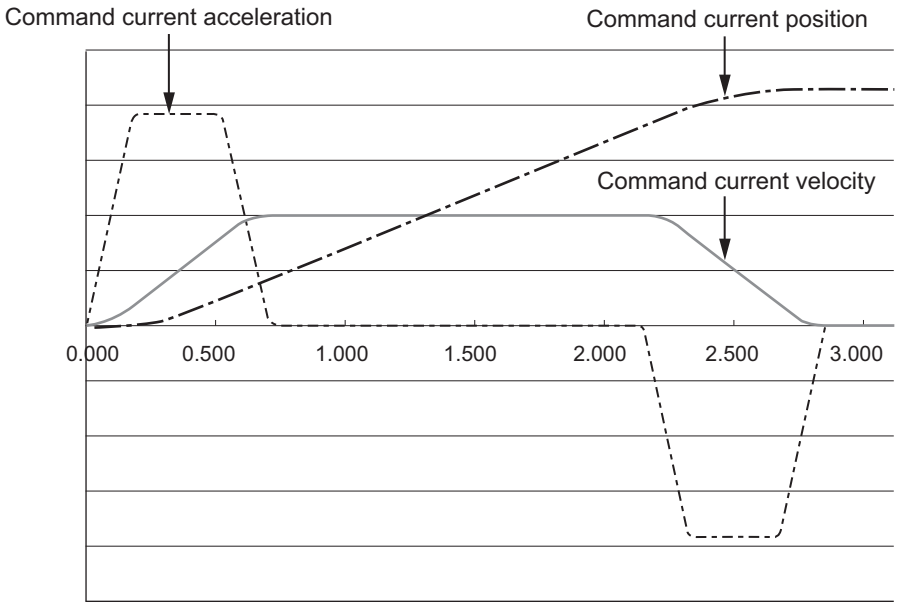
For details, refer to *Unit Conversion Settings* in the motion control user's manual.

Function

The VSMoveParam1 function block calculates the S-curve (i.e., velocity, acceleration, and jerk) parameters for suppressing the vibration that occurs during single axis positioning.

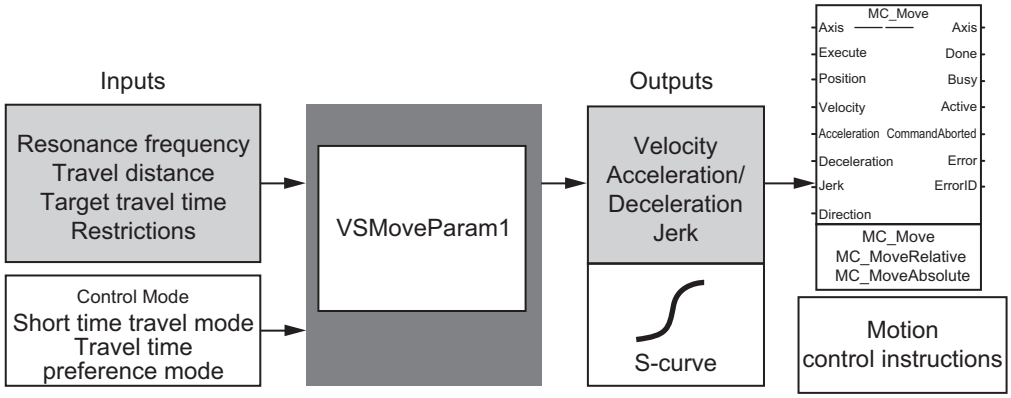
When restrictions such as velocity limit and acceleration/deceleration limit are input together with travel distance (angle), target travel time, and resonance frequency, an S-curve is created to suppress vibrations.





S-curve to suppress vibrations is created from the conditions 1 to 4.

This function block is used with motion control instructions, i.e., the MC_Move (Positioning) instruction, MC_MoveAbsolute (Absolute Positioning) instruction, and the MC_MoveRelative (Relative Positioning) instruction. The function block is executed only once before these instructions are expected. When the execution result of this function block is linked to the input variables (i.e., velocity, acceleration, deceleration, and jerk) to these instructions, command positions that lead to vibration suppression can be given.



Precautions for Correct Use

- When this function block is used for damping control, the *CalcTravelTime* (Calculated Travel Time) is restricted by the lower limit which enables the damping control (i.e., the shortest travel time). Refer to *Short Time Travel Mode* on page 42 for the shortest travel time.
- The following table specifies the recommended application rages for this function block. If an input value is out of the recommended application range, sufficient vibration suppression effect may not be ensured.

Item	Applicable range
Resonance Frequency	0.5 Hz to 50 Hz
Target Travel Time	50 ms to 1,000 ms

Precautions for Correct Use

This function block cannot produce sufficient vibration effect depending on the control target profile and operation conditions. If the given vibration suppression effect is not enough, it may be possible to improve the effect by reviewing the following settings.

- Measure the resonance frequency for the target, and adjust the value of *ResonanceFrequency* to be closest to the measured resonance frequency.
- Disable the short time travel mode (*ShortTimeMoveMode*=FALSE).However, this measure may extend the travel time.
- Lower the acceleration/deceleration limit. However, this measure may extend the travel time.
- Extend the target travel time.

Processing List

The table below lists the processing.

Processing	Description
Checking the input parameter ranges	When the *Execute* input variable changes to TRUE, the function block checks the input parameters for effective ranges. When they are in the effective ranges, the function block calculates S-curve. When they are out of the effective ranges, the processing stops with an error end and an error is output.
Creating the S-curve	When the *Execute* input variable changes to TRUE, the function block calculates S-curve. The function block automatically selects a VS level from the input parameters, VS Level 1 to 3, calculates S-curve that suppresses vibrations, outputs the calculation results to *Velocity, AccDec, Jerk,* and *CalcTravelTime* output variables. If *CalcTravelTime* (Calculated Travel Time) exceeds *TargetTravelTime* (Target Travel Time), *TravelTimeOver* changes to TRUE. For details, refer to *Creating S-curve* on page 41.
Short Time Travel Mode	When the *ShortTimeMoveMode* input variable changes to TRUE, the VS Level 3 which makes the travel time shortest is enabled. For details, refer to *Short Time Travel Mode* on page 42.
Travel Time Preference Mode	If *CalcTravelTIme* (Calculation Travel Time) exceeds *TargetTravelTime* (Target Travel Time), you can change the ProfileMode input variable to TRUE to ignore the vibration suppression effect and to calculate an S-curve that allows operation in the target travel time. Enable the Travel Time Preference Mode when you want to give priority to operations according to the target travel time over the vibration suppression effect. For details, refer to *Travel Time Preference Mode* on page 43.

Checking the Input Parameter Ranges

When the *Execute* input variable changes to TRUE, the function block checks the input parameters for effective ranges.

When they are in the effective ranges, the function block calculates S-curve.

When they are out of the effective ranges, an error occurs at execution start.

Creating S-curve

When the *Execute* input variable changes to TRUE, the function block calculates S-curve. Four patterns, i.e., VS level 1 to 3 and the travel time preference, are provided for S-curve.

The table below specifies the vibration suppression levels.

Item	Specification of the vibration suppression level 1 to 3
Vibration suppression effect	The lower VS level produces a greater effect.
Acceleration/Deceleration degree	The higher VS level produces a larger acceleration/deceleration.
Travel time found from the S-curve that is created by this function block	The higher VS level gives a shorter time.
Applicable condition	VS Level 1 and 2: Applied in all conditions VS Level 3: Applied only when the short time travel mode is enabled.

This function block preferentially selects the VS Level 1 which is higher in vibration suppression effect. However, if the travel time found by the created S-curve does not reach the target travel time, the function block uses the higher VS Level, in the order of 2 then 3, to have a travel time that can reach the target travel time. If none of the three VS levels can give a S-curve and travel time that reach the target travel time, the function block uses the VS level that gives a travel time that is closest to the target travel time.

When the travel time preference is selected, vibration suppression effect is disabled. S-curve calculation is performed in the following pattern selection order, which is specified by the combination of Short Time Travel Mode and Travel Time Preference Mode.

If the selected pattern cannot create an S-curve that gives the travel time reaching the target travel time, the function block use the next VS level in the order to calculate the parameter.

The function block continues the calculation until the travel time given by the created S-curve reaches the target travel time.

Short Time Travel Mode	Travel Time Preference Mode	Selection order among VS levels and Travel time preference
Enabled	Enabled	1. Vibration suppression level 1 2. Vibration suppression level 2 3. Vibration suppression level 3 4. Travel Time Preference Mode
Enabled	Disabled	1. Vibration suppression level 1 2. Vibration suppression level 2 3. Vibration suppression level 3
Disabled	Enabled	1. Vibration suppression level 1 2. Vibration suppression level 2 3. Travel Time Preference Mode
Disabled	Disabled	1. Vibration suppression level 1 2. Vibration suppression level 2

If the travel time given by the finally created S-curve is longer than the target travel time, *TravelTimeOver* changes to TRUE.

Precautions for Correct Use

If the calculated travel time is longer than the target travel time, this function block calculates S-curve again by automatically switching the vibration suppression level. This may extend the calculation time of this function block. Before you use this function block for actual operation, confirm that the calculation time of this function block does not exceed the task execution time.

Additional Information

The following guides apply to the VS levels that are selected for the target travel time. When you change the VS level, refer to the table below and adjust the parameters. However, when the calculated parameters are restricted by *VelocityLimit* (Velocity Limit) or *AccDecLimit* (Acceleration/Deceleration Limit), the guides in the table also vary.

Target travel time [ms]	Selected vibration suppression level
4,000/Resonance frequency [Hz] or over	VS level 1
4,000/Resonance frequency [Hz] to 20,000/9/Resonance frequency [Hz]	VS level 2
20,000/9/Resonance frequency [Hz] to 1,100/Resonance frequency [Hz]	VS level 3
1,100/Resonance frequency [Hz] or less	No VS level can creates S-curve that produces a travel time in the target travel time.*1

*1. When the Travel Time Preference Mode is enabled, the S-curve created by the travel time preference is selected.

Additional Information

The *TravelTimeOver* output variable is used for the adjustment of this function block. If the *TravelTimeOver* output variable is TRUE, the travel time given by the S-curve created based on the calculated parameters is longer than the target travel time. Take the following measures to match the target travel time with the travel time given by S-curve.

- Increase the value of *TargetTravelTime* (Target Travel Time).
- Increase the value of *VelocityLimit* (Velocity Limit) or *AccDecLimit* (Acceleration/Deceleration Limit).
- Enable the travel time preference mode. However, this may disable the vibration suppression effect.

Short Time Travel Mode

When the *ShortTimeMoveMode* input variable changes to TRUE, the short time travel mode is enabled. When the short time travel mode is enabled, the S-curve patten from the VS level 3, which allows operations in the shortest travel time, becomes selectable. For details on the relationship between the VS levels and the target travel time, refer to *Creating S-curve* on page 41.

Precautions for Correct Use

When the short time travel mode is enabled, the vibration suppression effect may not be ensured. In such a case, disable the short time travel mode.

Travel Time Preference Mode

When the *ProfileMode* input variable changes to TRUE, the travel time preference mode is enabled. When the following condition applies to the target travel time, an S-curve is created by the vibration suppression calculation regardless of whether the travel time preference mode is enable or disabled, so that the target travel time equals to the calculated travel time.

- The short time travel mode is enabled, and the target travel time is the same as or greater than the target travel time lower limit that is calculated by the VS level 3.
- The short time travel mode is disabled, and the target travel time is the same as or greater than the target travel time lower limit that is calculated by the VS level 2.

If neither of the above conditions applies, an S-curve is created by giving priority to the target travel time over the shortest travel time. The following table shows the processing content.

Control mode	Processing
ProfileMode (Travel Time Preference Mode) is enabled	Vibration suppression calculation is not performed. An S-curve that meets the target travel time is created. Target travel time = Calculated travel time
ProfileMode (Travel Time Preference Mode) is disabled	An S-curve is created by vibration suppression calculation. Target travel time < Calculated travel time

Additional Information

- For details on the target travel time lower limits that can be calculated by the VS level 2 or 3, refer to *Creating S-curve* on page 41.
- When the axis moves in the target travel time and a value that exceeds the velocity limit or deceleration limit is given, the limit is given priority. As a result, the target travel time may be greater than the calculated travel time even if the travel time preference mode is enabled.

Timing Charts

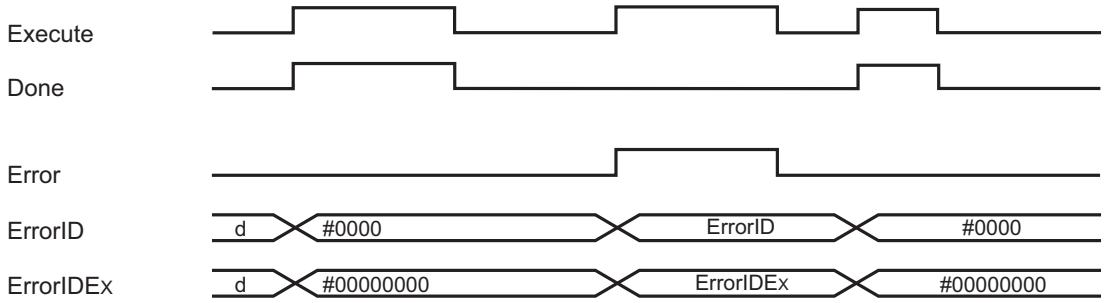
The function block executes the processing when it detects the *Execute* input variable changes to TRUE.

If *Execute* remains TRUE even after the processing is completed, the *Done* and *TravelTimeOver* output variables are retained. The *Velocity, AccDec, Jerk, CalcTravelTime, SelectPattern, ErrorID*, and *ErrorIDEx* output variables are retained even if the function block is executed when the *Execute* input variable is FALSE.

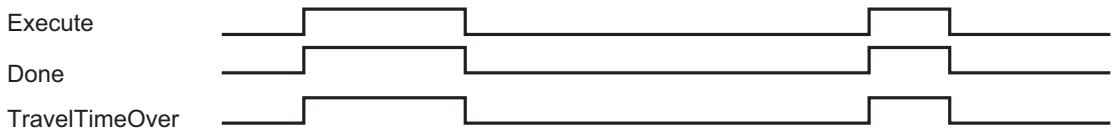
The *ErrorID* and *ErrorIDEx* output variables are cleared to zero when the *Execute* input variable changed to TRUE.

For details on *ErrorID* and *ErrorIDEx*, refer to *Troubleshooting* on page 45.

● Relationship between Execute, Done, Error, ErrorID, and ErrorIDEx



● Relationship between Execute, Done, and TravelTimeOver



Troubleshooting

Error Code	Expansion Error Code	Status	Description	Correction
16#0000	16#00000000	Normal End	---	---
16#3C17	16#00000001	Travel Distance Out of Range	A value set to the *Distance* input parameter is out of the valid range.	Check the valid range of the *Distance* input parameter, and set a value in the valid range.
	16#00000002	Target Travel Time Setting Out of Range	A value set to the *TargetTravelTime* input parameter is outside the valid range.	Check the valid range of the *TargetTravelTime* input parameter, and set a value within the valid range.
	16#00000003	Resonance Frequency Setting Out of Range	A value set to the *Resonance Frequency* input parameter is outside the valid range.	Check the valid range of the *Resonance Frequency* input parameter, and set a value within the valid range.
	16#00000004	Velocity Limit Setting Out of Range	A value set to the *VelocityLimit* input parameter is outside the valid range.	Check the valid range of the *VelocityLimit* input parameter, and set a value within the valid range.
	16#00000005	Acceleration/Deceleration Limit Setting Out of Range	A value set to the *AccDecLimit* input parameter is outside the valid range.	Check the valid range of the *AccDecLimit* input parameter, and set a value within the valid range.
	16#00000006	Parameter Calculation Failed	The S-curve parameters cannot be calculated correctly.	Check each input parameter to confirm if a value close to the maximum value or the minimum value of LREAL is set to it.

Sample Programming

Program Operation

The sample program uses the function block to suppress vibration of the control target and to achieve relative positioning. The sample program performs the following processing.

1 When the *Start* input variable is executed, the function block is executed and S-curve commands (*Velocity, AccDec*, and *Jerk* output variables) are calculated.

2 The S-curve commands are set to the *Velocity, AccDec*, and *Jerk* input variables to the motion control instruction.

Preconditions

The sample program assumes the following status.

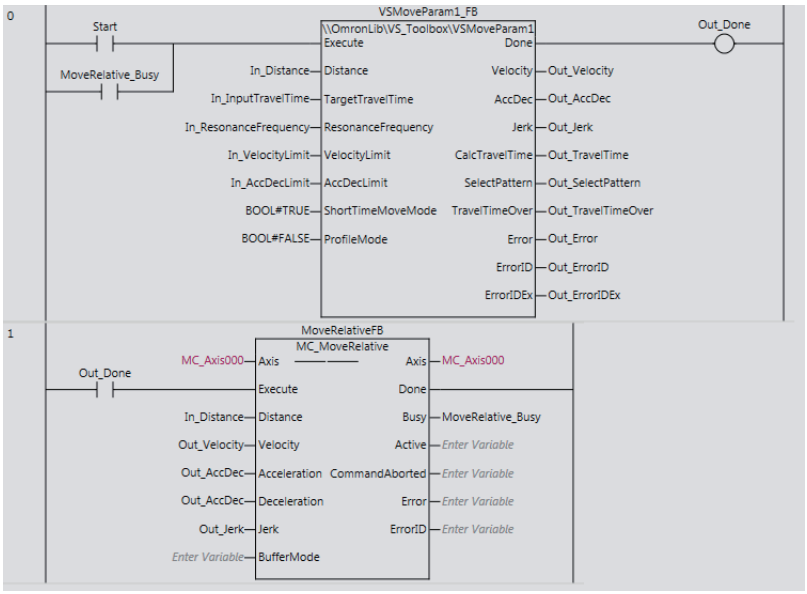
- The Axis 0 (axis variable *MC_Axis000*) is operable.

Main Variables

Variable	Data type	Initial value	Comment
Start	BOOL	FALSE	Starts relative positioning when the variable is TRUE.
MC_Axis000	_sAXIS_REF	---	Axis variable for Axis 0
In_Distance	LREAL	0.0	Travel distance for relative positioning
In_TargetTravelTime	UINT	0	Target travel time for relative positioning
In_ResonanceFrequency	LREAL	0.0	Specifies the resonance frequency of the vibration to be suppressed.
In_VelocityLimit	LREAL	0.0	Velocity limit
In_AccDecLimit	LREAL	0.0	Acceleration/Deceleration limit

Ladder Diagram

When *Start* is changed to TRUE, the relative travel starts for the portion specified by *In_Distance*.



Structured Text (ST)

```
IF Start OR MoveRelative_Busy THEN
   VSMoveParam1_FB_Execute:=TRUE;
ELSE
   VSMoveParam1_FB_Execute:=FALSE;
END_IF;

IF Out_Done THEN
   MoveRelativeFB_Execute:=TRUE;
ELSE
   MoveRelativeFB_Execute:=FALSE;
END_IF;

//VSMoveParam1
VSMoveParam1_FB
(Execute:=VSMoveParam1_FB_Execute,
Distance:=In_Distance,
TargetTravelTime:=In_InputTravelTime,
ResonanceFrequency:=In_ResonanceFrequency,
VelocityLimit:=In_VelocityLimit,
AccDecLimit:=In_AccDecLimit,
ShortTimeMoveMode:=BOOL#TRUE,
```

```
ProfileMode:=BOOL#FALSE,
Done=>Out_Done,
Velocity=>Out_Velocity,
AccDec=>Out_AccDec,
Jerk=>Out_Jerk,
CalcTravelTime=>Out_TravelTime,
TravelTimeOver=>Out_TravelTimeOver,
Error=>Out_Error,
ErrorID=>Out_ErrorID,
ErrorIDEx=>Out_ErrorIDEx
);


//MC_MoveRelative
MoveRelativeFB
(Axis:=MC_Axis000,
Execute:=MoveRelativeFB_Execute,
Distance:=In_Distance,
Velocity:=Out_Velocity,
Acceleration:=Out_AccDec,
Deceleration:=Out_AccDec,
Jerk:=Out_Jerk,
Busy=>MoveRelative_Busy);
```

Additional Information

For the programming procedure, refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504).

MultiVSFilter1

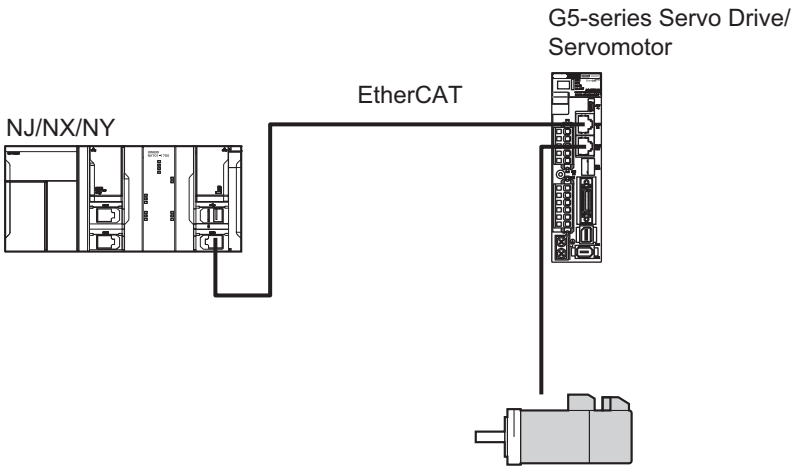
The function block suppresses vibrations of the control target with the position command created by the filter.

Function block name	Name	FB/ FUN	Graphic expression	ST expression
MultiVSFilter1	Multiple Frequency VS Filter 1	FB	MultiVSFilter1_Instance \\OmronLib\VS_Toolbox\MultiVSFilter1 Axis — Axis VSParam — VSParam Enable — Enabled CalcRslt Busy Error ErrorID ErrorIDEx	MultiVSFilter1_instance(Axis:=, VSParam:=, Enable:=, Enabled=>, CalcRslt=>, Busy=>, Error=>, ErrorID=>, ErrorIDEx=>);

Function Block and Function Information

Item	Description
Library file name	OmronLib_VS_Toolbox_V1_0.slr
Namespace	OmronLib\VS_Toolbox
Function block and function number	00029
Source code published/not published	Not published
Function block and function version	1.00

Hardware Configuration Diagram



Variables

Input Variables

Name	Meaning	Data type	Valid range	Initial value	Description
Enable	Enable	BOOL	TRUE or FALSE	FALSE	Enables the VS filter when the value changes to TRUE.

Precautions for Correct Use

When the axis setting for the function block is used in Rotary Mode, the processing makes an error end if the command current velocity of the axis variable exceeds the following maximum allowable velocity [command unit/s].

Maximum allowable velocity = (Modulo maximum position - Modulo minimum position) / 2 × (Maximum buffering time)

The condition for the above expression is as follows.

- Maximum buffering time = 0.5/ Resonance frequency + Δt
- Δt = 0.002 s (Task frequency of the task which executes this function < 0.002 s)
  = Task frequency (Task frequency of the task which executes this function ≥ 0.002 s)
- Minimum resonance frequency is the value in vibration suppression parameters when *VSType* (VS Type)≠ 0 and *ResonacceFrequency* is minimum.

Precautions for Correct Use

When the axis setting for the function block is used in Rotary Mode, be sure to select mcShortestWay (Shortest way) to the *Direction* input variable to the MC_SyncMoveAbsolute instruction.

Additional Information

- The command unit is indicated either in [mm], [µm], [nm], [degree], [inch], or [pulse].
  For details, refer to *Unit Conversion Settings* in the motion control user's manual.
- When you use a G5-series Servo Drive or Servomotor, the valid range for the modulo maximum and minimum position setting values is -2^{63} to $2^{63}-1$.

Output Variables

Name	Meaning	Data type	Valid range	Description
Enabled	Enabled	BOOL	TRUE or FALSE	Outputs TRUE in any period in which *CalcRslt* (Processing Result) is updated.
CalcRslt	Processing Result	LREAL	Depends on data types.	Gives the calculation result (the command position calculated with filters).
Busy	Executing	BOOL	TRUE or FALSE	TRUE when the function block is acknowledged.
Error	Error	BOOL	TRUE or FALSE	TRUE while there is an error.
ErrorID	Error Code	WORD	*1	Contains the error code when an error occurs. The code 16#0000 indicates normal execution.
ErrorIDEx	Expansion Error Code	DWORD	*1	Contains the expansion error code when an error occurs.

*1. Refer to *Troubleshooting* on page 58 for error codes.

In-Out Variables

Name	Meaning	Data type	Valid range	Description
Axis	Axis	_sAXIS_REF	---	Specifies the axis.
VSParam	Vibration Suppression Parameter	OmronLib\VS_Toolbox\sVSParam[0-4]	---	Specifies the vibration suppression parameters. Input values are read when *Enable* changes to TRUE.

● Members of the sVSParam Structure

Member	Meaning	Data type	Valid range	Description
VSType	VS Type	UINT	0,1, 2, or 3	Selects the type of vibration suppression. 0: No suppression 1: Type 1 2: Type 2 3: Type 3
Resonance Frequency	Resonance Frequency	LREAL	0.5 ≤ Resonance frequency ≤ 50	Specifies the resonance frequency of the vibration to be suppressed. The unit is [Hz].

Function

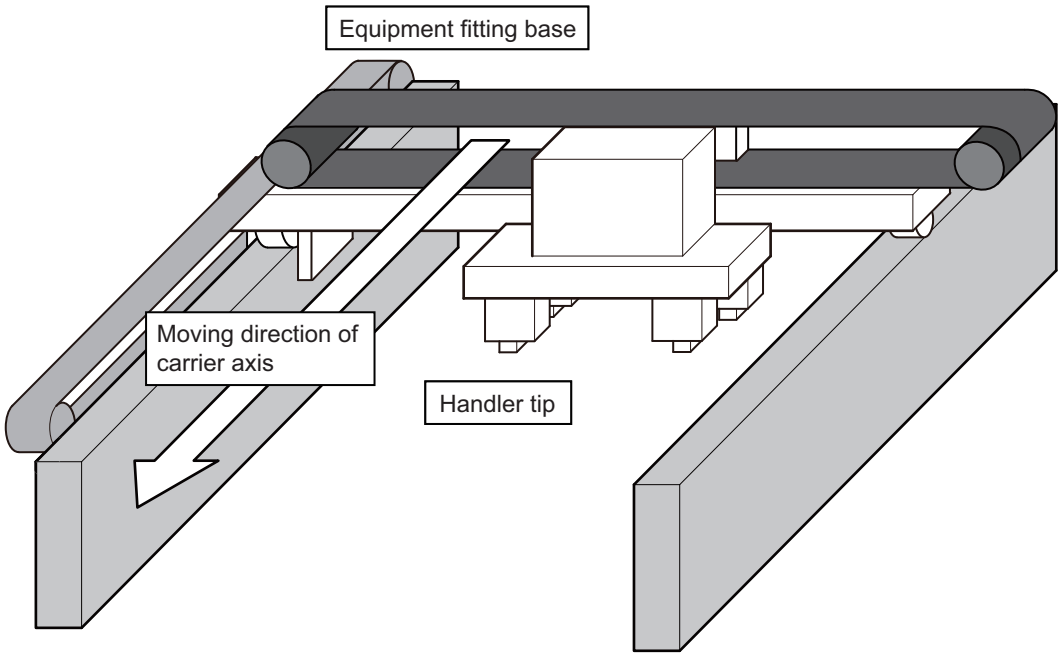
Outline of Function

This function block creates, from the command position of the axis, the command position that can suppress up to five vibrations that occur on the equipment. The function block can operate up to five vibration suppression filters of resonance frequency against the command position created by the Motion Control Function Module. Vibration suppression with this function block extends the travel time. For details, refer to *Function in Detail* on page 54.

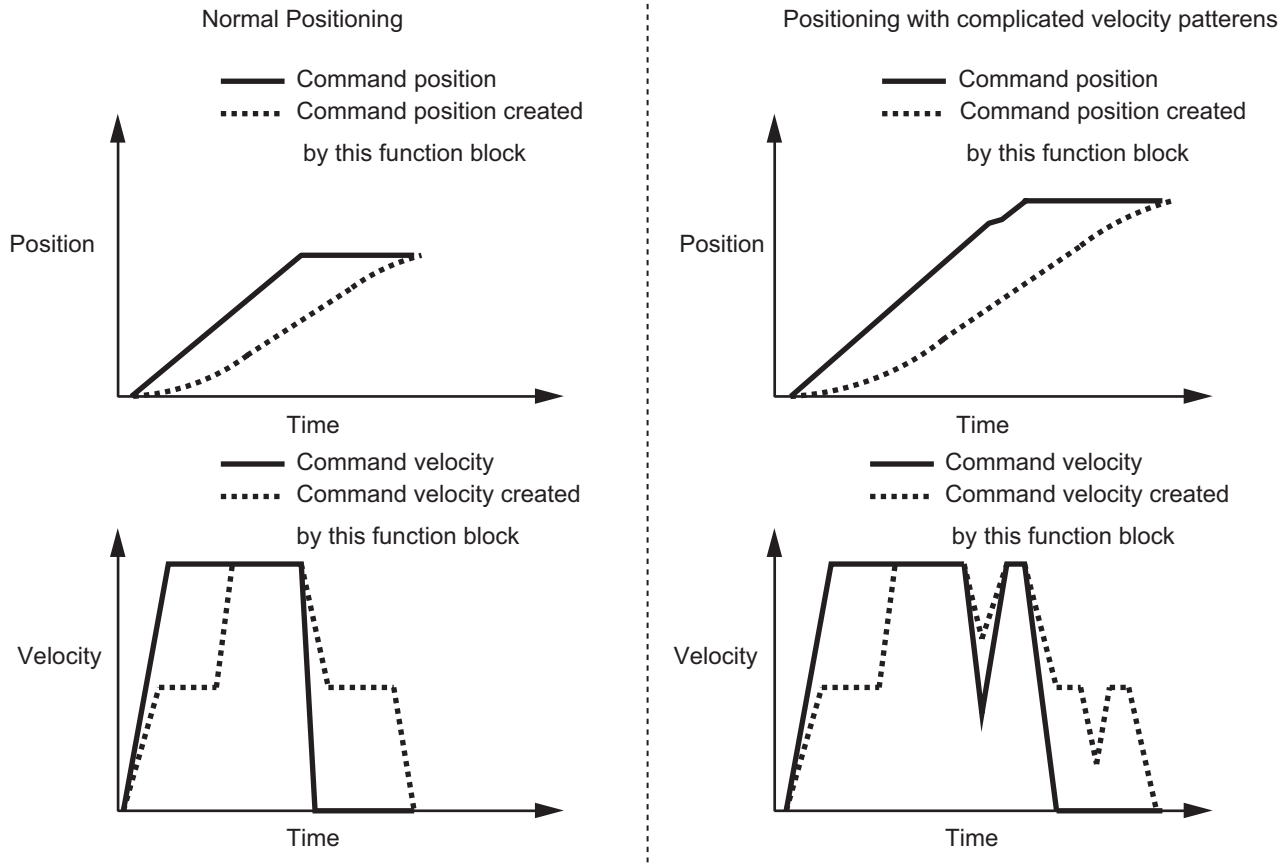
The following case examples are applicable to this function block.

- **When the function block is used for a handling equipment, it can suppress vibrations on the tip of the hand and the equipment fitting base that are caused by the motion of the carrier axis.**

<Example: Handling Equipment (Linear Mode)>



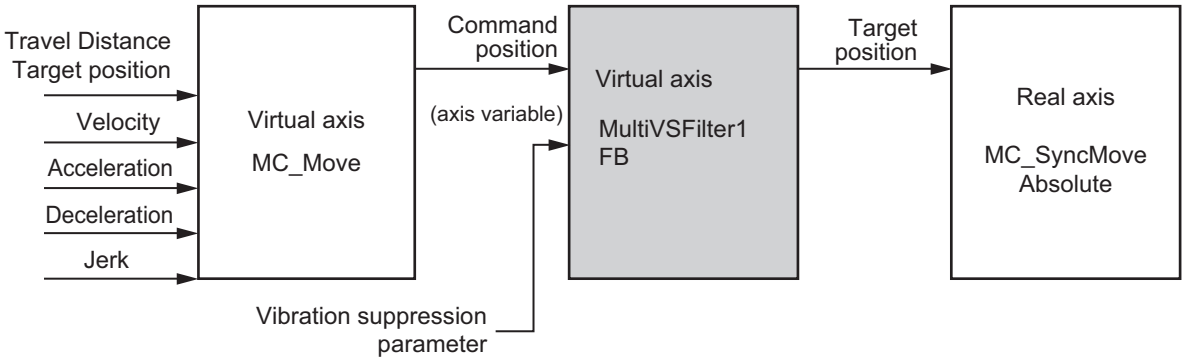
- **The function block suppresses vibrations on the equipment that performs positioning with complicated velocity patterns, such as a target position change during multi-execution, re-execution, or other motion, in addition to normal positioning.**



How to Use this Function Block

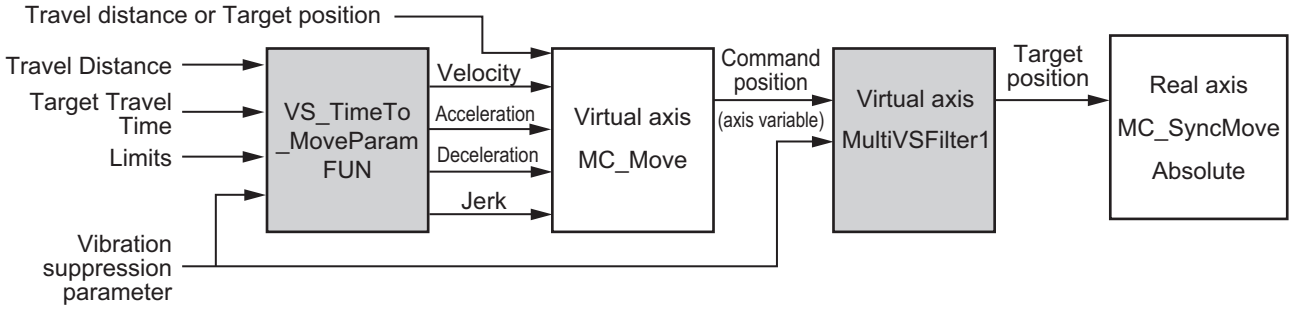
Usage of this function block can be mainly classified into the following three types.

● When Single Axis Positioning Is Used for Multi-execution or Re-execution



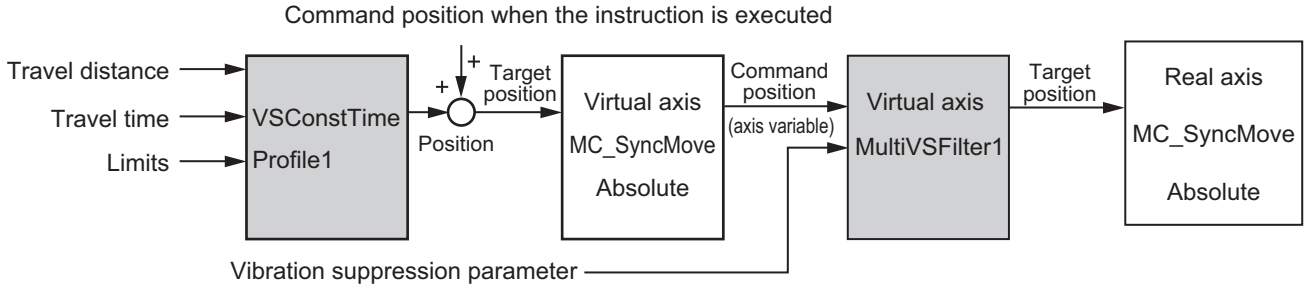
A command position for single axis positioning is created with a virtual servo axis, and is input to this function block via an axis variable to create a target position to suppress vibrations. Then the created target position is input to the MC_SyncMoveAbsolute instruction to a real servo axis.

● When Single Axis Positioning Is Used for Travel Time Specific Positioning



Travel time specific damping control is realized by adding TimeToMoveParam1 function to the combination of the MC_Move instruction and this function block. Refer to *TimeToMoveParam1* on page 99 for details on TimeToMoveParam1 function.

● When the Command Position Generated by VSConstTimeProfile1FB or VSConstVelProfile1FB Is Used in Combination with this Function Block



Highly efficient damping control is realized by the use of the vibration suppression profile that uses VSConstTimeProfile1FB or VSConstVelProfile1FB to generate the command position to be input to this function block. Refer to *VSConstTimeProfile1* on page 65 and *VSConstVelProfile1* on page 84for details on VSConstTimeProfile1FB and VSConstVelProfile1FB.

Function in Detail

This function block performs the following two functions.

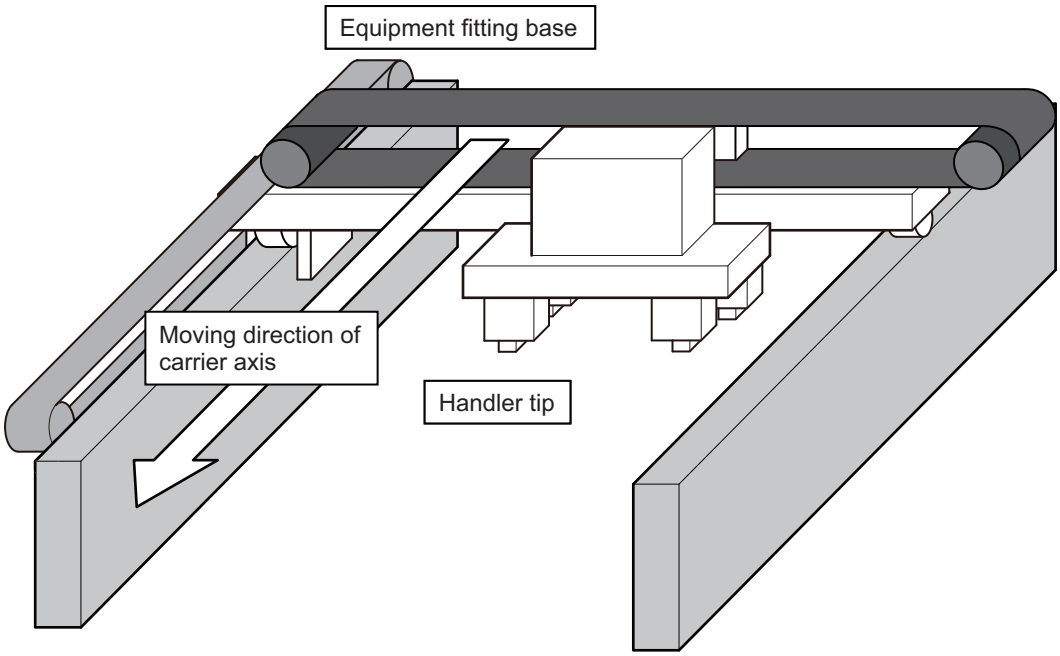
- Suppressing vibrations on equipment that has up to five resonance frequencies
- Adjusting robust against resonance frequency errors

The following section describes each function in detail.

● Suppressing Vibrations on Equipment That Has up to Five Resonance Frequencies

The function block can suppress vibrations on the tip of the handler and the equipment fitting base that are caused by the motion of the carrier axis. Resonance frequencies between 0.5 to 50 Hz can be suppressed.

<Example: Handling Equipment>



Vibration suppression with this function block cause a delay. A delay means a difference between the travel time in the single axis positioning instruction and the time taken from when the single axis positioning starts to when the position given by MultiVSFilter1 reaches the target position.

The following procedures can be used to calculate the delay cased by vibration suppression with this function block.

1 Use *VSParam[n].VSType*, and calculate the delay $_n$ [ms] caused by vibration suppression of n-th (n=0 to 4) resonance frequency.

VSType	VS type	Delay$_n$ [ms]*1
0	No suppression	0
1	Type 1	$\frac{1{,}000}{2 \bullet F}$ [ms]
2	Type 2	$\frac{2{,}000}{2 \bullet F}$ [ms]
3	Type 3	$\frac{3{,}000}{2 \bullet F}$ [ms]

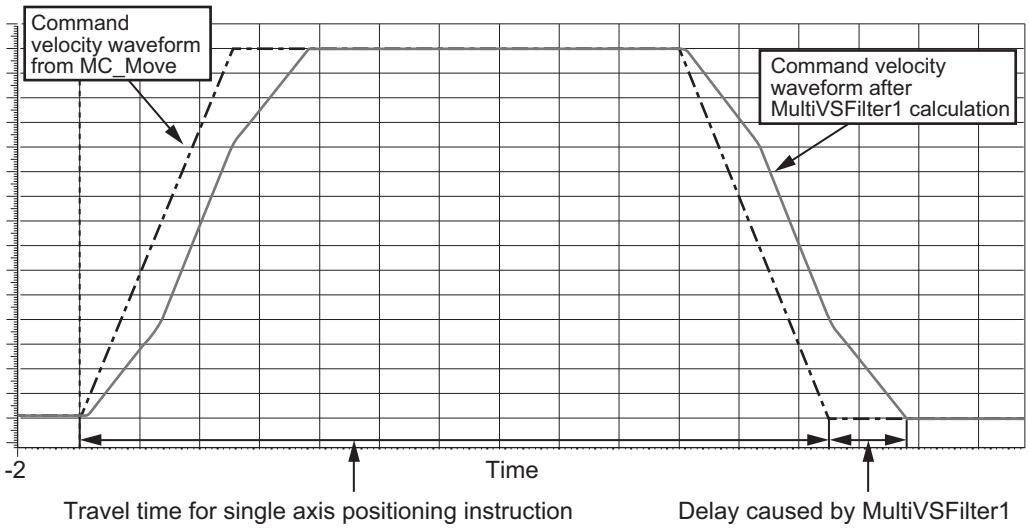
*1. The character “F” in the table indicates the *Resonance-Frequency* input variable.

2 Add up to the calculated Delay $_n$ [ms].

$$\text{Delay [ms]} = \text{Delay}_0 + \text{Delay}_1 + \ldots + \text{Delay}_4$$

$$\text{Delay}_n\ [\text{ms}] = \frac{1{,}000}{2 \bullet Fn} \bullet \text{VSType}$$

(F indicates resonance frequency.)



● Adjusting Robust against Resonance Frequency Errors

When a large value is set to *VSType*, robust improves so that vibration suppression effect can be maintained even if there is a resonance frequency error. The filter delay is also extended.

Parameter Setting Procedure

This section gives the procedure to set parameters for this function block.

1 *VSParam[i] .ResonanceFrequency* In-Out Variable

Measure the resonance frequency for the control target, and set it.

2 *VSParam[i] .VSType* In-Out Variable

Adjust the robust. Normally set *VSType* to 1.

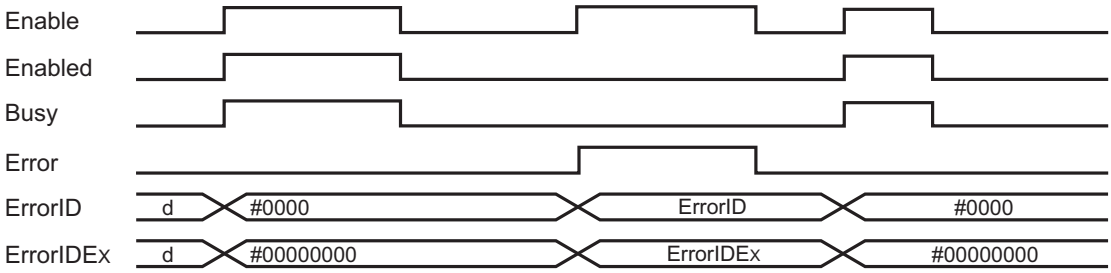
If sufficient vibration suppression effect cannot be given due to resonance frequency errors, set *VSType* as shown in the following table according to the variance of the resonance frequency.

VSType	VS type	Variance of Resonance Frequency	Delay*1
1	Type 1	5% or less	$\frac{1{,}000}{2 \cdot F}$ [ms]
2	Type 2	10% or less	$\frac{2{,}000}{2 \cdot F}$ [ms]
3	Type 3	15% or less	$\frac{3{,}000}{2 \cdot F}$ [ms]

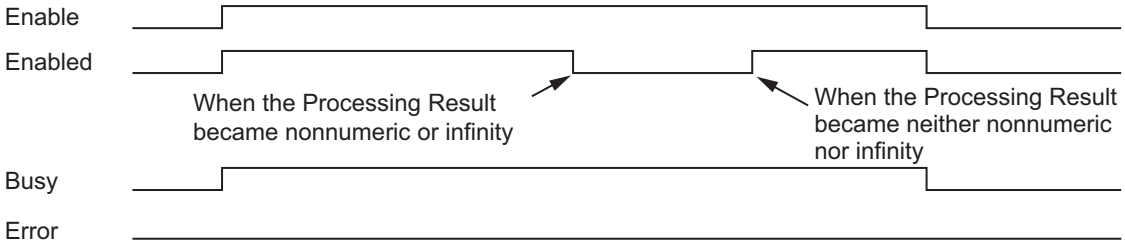
*1. F indicates resonance frequency.

Timing Charts

● Norma Execution, or Change from Error to Normal Execution

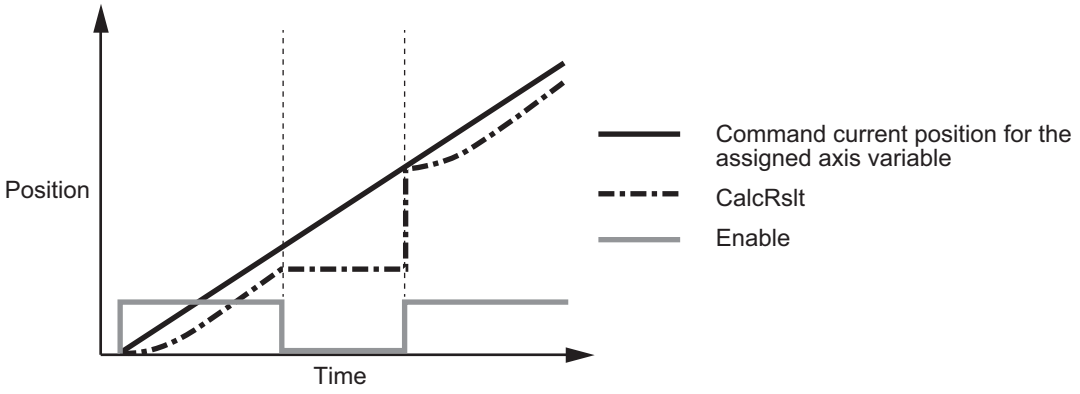


If the *CalcRslt* (Processing Result) of this function block is an infinity, *Enabled* will change to FALSE.
If the *CalcRslt* (Processing Result) is not an infinity, *Enabled* will change to TRUE.



● Update of Calculation Results

After this function block is executed, the command position calculated with the filter (the processing result when *Enabled*=TRUE) is given to the *CalcRslt* (Processing Result). For example, if Enable to this function block is changed between TRUE and FALSE, the most recent valid output value for TRUE is given to the *CalcRslt* (Processing Result) while Enable is FALSE. Even if this function block ends in an error during execution, the most recent valid command position (the latest processing result when *Enabled*=TRUE) is given to the *CalcRelt* (Processing Result).



Troubleshooting

Error Code	Expansion Error Code	Status	Description	Correction
16#0000	16#00000000	Normal End	---	---
16#3C18	16#00000001	VS Type Selection Out of Range	A value set to the *VSType* member of *VSParam* is out of the valid range.	Check the valid range of the *VSType* member of *VSParam*, and set a value within the valid range.
	16#00000002	Resonance Frequency Setting Our of Range	A value set to the *ResonanceFrequency* member of *VSParam* is out of the valid range.	Check the valid range of the *ResonanceFrequency* member of *VSParam*, and set a value within the valid range.
	16#00000003	Command Current Velocity Exceeded Maximum Allowable Velocity	The command current velocity of the axis variable exceeds the maximum allowable velocity.	Set a command position so that the command current velocity of the axis variable does not exceed the maximum allowable velocity. For details, refer to *Input Variables* on page 49.

Sample Programming

Program Operation

The sample program use the function block to operate a servo axis specified to operate in Rotary Mode, in relative positioning. The table below shows the processing order.

No.	Outline	Description
1.	Specifying the time specific positioning and the parameters	The program determines whether the TimeToMoveParam1 function is used or not, and calculate the values set to *Distance, Velocity, Acceleration, Deceleration,* and *Jerk* input parameters to the MC_MoveRelative instruction.*1
2.	Absolute positioning	The program executes the relative positioning for the virtual servo axis.
3.	Starting or stopping synchronization	• Starting synchronization The program changes the command current position of the virtual servo axis to the command current position of the servo axis, and starts synchronization for the servo axis. • Stopping synchronization The program stops synchronization for the servo axis when an error occurs or when an vibration suppression parameter is changed.
4.	Vibration suppression filtering	The program executes vibration suppression filtering for the command current position of the virtual servo axis.
5.	Cyclic synchronous absolute positioning	The servo axis is synchronized to the output value of the vibration suppression filter.
6.	Canceling synchronization	The synchronization to the servo axis is canceled (it decelerates to stop).

*1. When the function block is used with the TimeToMoveParam1 function against the servo axis in Rotary Mode, the relative positioning of the MC_Move instruction or the MC_MoveRelative instruction must be used.

Preconditions

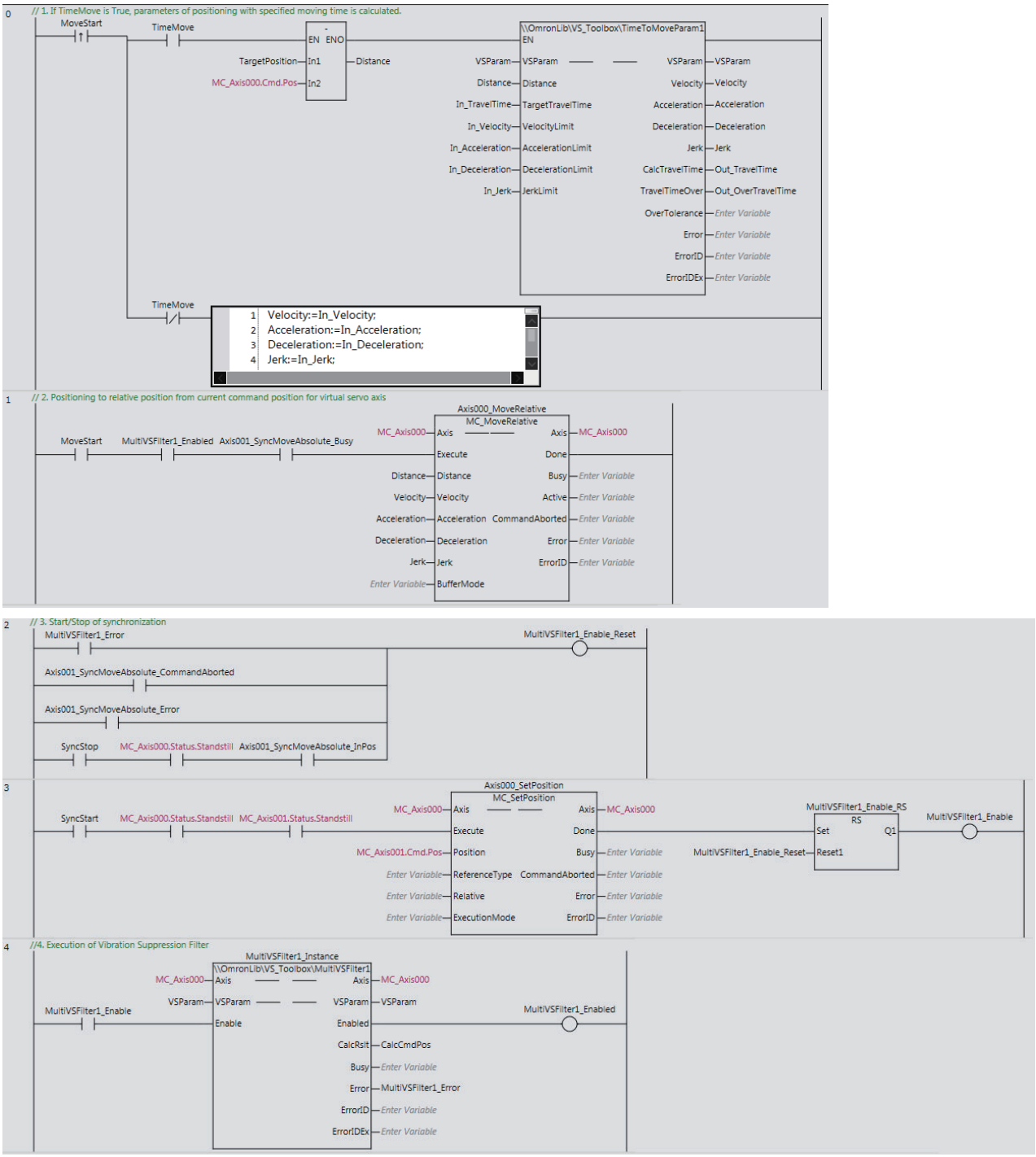
- The virtual servo axis settings are the same as the servo axis settings (except for homing).
- The virtual servo axis and servo axis are ready to operate.
- The home for the virtual servo axis and servo axis are defined.

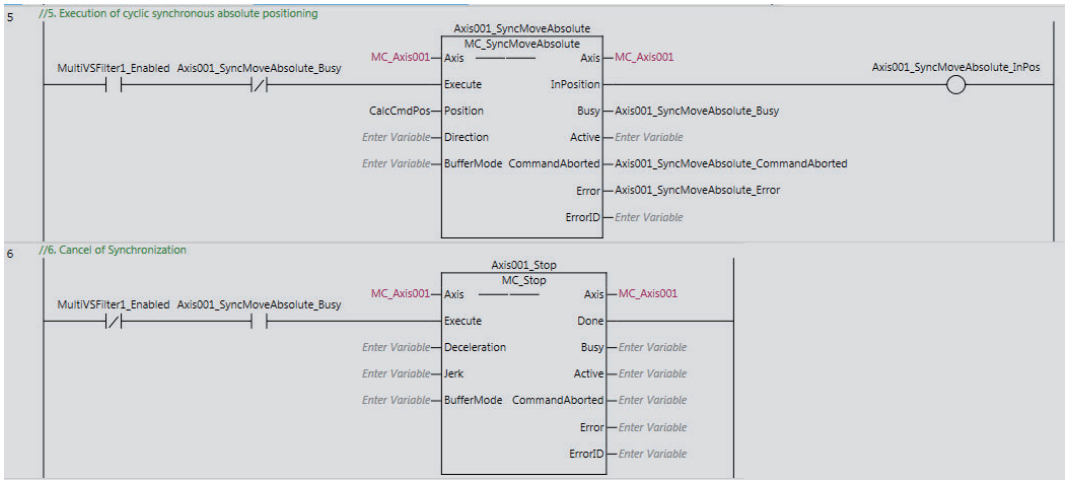
Ladder Diagram

Main Variables

Name	Data type	Initial value	Comment
MC_Axis000	_sAXIS_ REF	---	Axis variable for the virtual axis. The variable is assigned to the in-out variables to MultiVSFilter1.
TimeMove	BOOL	FALSE	When the flag changes to TRUE, the time specific positioning is enabled.
MoveStart	BOOL	FALSE	When the flag changes to TRUE, positioning the virtual servo axis is started.
SyncStart	BOOL	FALSE	When the flag changes to TRUE, synchronization for the servo axis is started.
SyncStop	BOOL	FALSE	When the flag changes to TRUE, synchronization for the servo axis is stopped.
VSParam	ARRAY[0..4] OF OmronLib\VS_Tool-box\sVSParam	---	Specifies the vibration suppression parameters.
TargetPosition	LREAL	0	Specifies the target position. The unit is [command unit].
Distance	LREAL	0	Specifies the relative distance from the current position to the target position.
In_Velocity	LREAL	0	Specifies the velocity limit. The unit is [command unit/s].
In_Acceleration	LREAL	0	Specifies the acceleration limit. The unit is [command unit/s^2].
In_Deceleration	LREAL	0	Specifies the deceleration limit. The unit is [command unit/s^2].
In_Jerk	LREAL	0	Specifies the jerk limit. The unit is [command unit/s^3].

Programming




5
//5. Execution of cyclic synchronous absolute positioning
Axis001_SyncMoveAbsolute
MC_SyncMoveAbsolute
MultiVSFilter1_Enabled
Axis001_SyncMoveAbsolute_Busy
MC_Axis001
Axis
MC_Axis001
Execute
InPosition
Axis001_SyncMoveAbsolute_InPos
CalcCmdPos
Position
Busy
Axis001_SyncMoveAbsolute_Busy
Enter Variable
Direction
Active
Enter Variable
Enter Variable
BufferMode
CommandAborted
Axis001_SyncMoveAbsolute_CommandAborted
Error
Axis001_SyncMoveAbsolute_Error
ErrorID
Enter Variable
6
//6. Cancel of Synchronization
Axis001_Stop
MC_Stop
MultiVSFilter1_Enabled
Axis001_SyncMoveAbsolute_Busy
MC_Axis001
Axis
MC_Axis001
Execute
Done
Enter Variable
Deceleration
Busy
Enter Variable
Enter Variable
Jerk
Active
Enter Variable
Enter Variable
BufferMode
CommandAborted
Enter Variable
Error
Enter Variable
ErrorID
Enter Variable

Structured Text (ST)

● Main Variables

Name	Data type	Initial value	Comment
MC_Axis000	_sAXIS_ REF	---	Axis variable for the virtual axis. The variable is assigned to the in-out variables to MultiVSFilter1.
TimeMove	BOOL	FALSE	When the flag changes to TRUE, the time specific positioning is enabled.
MoveStart	BOOL	FALSE	When the flag changes to TRUE, positioning the virtual servo axis is started.
WriteParameter	BOOL	FALSE	The variable changes to TRUE by the upward differentiation of *MoveStart*, and the parameters to the MC_MoveRelative are over-written.
SyncStart	BOOL	FALSE	When the flag changes to TRUE, synchronization for the servo axis is started.
SyncStop	BOOL	FALSE	When the flag changes to TRUE, synchronization for the servo axis is stopped.
VSParam	ARRAY[0..4] OF OmronLib\VS_Tool-box\sVSParam	---	Specifies the vibration suppression parameters.
TargetPosition	LREAL	0	Specifies the target position. The unit is [command unit].
Distance	LREAL	0	Specifies the relative distance from the current position to the target position.
In_Velocity	LREAL	0	Specifies the velocity limit. The unit is [command unit/s].
In_Acceleration	LREAL	0	Specifies the acceleration limit. The unit is [command unit/s^2].
In_Deceleration	LREAL	0	Specifies the deceleration limit. The unit is [command unit/s^2].
In_Jerk	LREAL	0	Specifies the jerk limit. The unit is [command unit/s^3].

● Programming

```
// 1. If TimeMove is True, parameters of positioning with specified moving time is
calculated.

R_TRIG_Instance
(Clk:=MoveStart,
Q=>WriteParameter);

IF WriteParameter THEN
   IF TimeMove THEN
      Distance:=TargetPosition-MC_Axis000.Cmd.Pos;
      \\OmronLib\VS_Toolbox\TimeToMoveParam1
      (VSParam:=VSParam,
      Distance:=Distance,
      TargetTravelTime:=In_TravelTime,
      VelocityLimit:=In_Velocity,
      AccelerationLimit:=In_Acceleration,
      DecelerationLimit:=In_Deceleration,
      JerkLimit:=In_Jerk,
      Velocity=>Velocity,
      Acceleration=>Acceleration,
      Deceleration=>Deceleration,
      Jerk=>Jerk,
      CalcTravelTime=>Out_TravelTime,
      TravelTimeOver=>Out_TravelTimeOver);
   ELSE
      Velocity:=In_Velocity;
      Acceleration:=In_Acceleration;
```

```
      Deceleration:=In_Deceleration;
      Jerk:=In_Jerk;
   END_IF;
END_IF;

// 2. Positioning to relative position from current command position for virtual
servo axis
IF MoveStart AND MultiVSFilter1_Enabled AND Axis001_SyncMoveAbsolute_Busy THEN
   Axis000_Move_Execute:=TRUE;
ELSE
   Axis000_Move_Execute:=FALSE;
END_IF;

// 3. Start/Stop of synchronization
IF MultiVSFilter1_Error
   OR Axis001_SyncMoveAbsolute_CommandAborted
   OR Axis001_SyncMoveAbsolute_Error
   OR (SyncStop AND MC_Axis000.Status.Standstill AND Axis001_SyncMoveAbsolute_InPos)
THEN
   MultiVSFilter1_Enable_Reset:=TRUE;
END_IF;
IF SyncStart AND MC_Axis000.Status.Standstill AND MC_Axis001.Status.Standstill THEN
   Axis000_SetPosition_Execute:=TRUE;
ELSE
   Axis000_SetPosition_Execute:=FALSE;
END_IF;


//4. Execution of Vibration Suppression Filter
IF MultiVSFilter1_Ready THEN
   MultiVSFilter1_Enable:=TRUE;
ELSE
   MultiVSFilter1_Enable:=FALSE;
END_IF;

//5. Execution of cyclic synchronous absolute positioning
IF MultiVSFilter1_Enabled AND NOT(Axis001_SyncMoveAbsolute_Busy) THEN
   Axis001_SyncMoveAbsolute_Execute:=TRUE;
ELSE
   Axis001_SyncMoveAbsolute_Execute:=FALSE;
END_IF;

//6. Cancel of Synchronization
IF NOT(MultiVSFilter1_Enabled) AND Axis001_SyncMoveAbsolute_Busy THEN
   Axis001_Stop_Execute:=TRUE;
ELSE
   Axis001_Stop_Execute:=FALSE;
END_IF;



//MC_MoveRelative
Axis000_MoveRelative
(Axis:=MC_Axis000,
Execute:=Axis000_Move_Execute,
Distance:=Distance,
Velocity:=Velocity,
Acceleration:=Acceleration,
Deceleration:=Deceleration,
Jerk:=Jerk);

// MC_SetPosition
Axis000_SetPosition
(Axis:=MC_Axis000,
Execute:=Axis000_SetPosition_Execute,
```

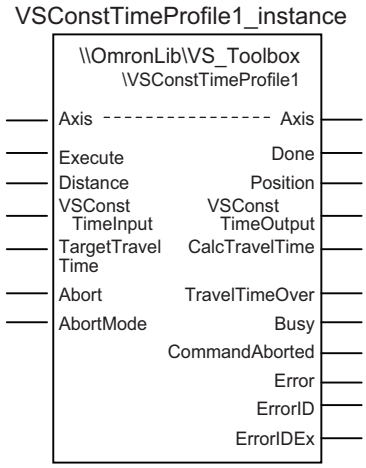
```
Done=>Axis000_SetPosition_Done);

// RS
MultiVSFilter1_Enable_RS
(Set:=Axis000_SetPosition_Done,
Reset1:=MultiVSFilter1_Enable_Reset,
Q1=>MultiVSFilter1_Ready);

// MultiVSFilter1
MultiVSFilter1_instance
(
Axis:=MC_Axis000,
VSParam:=VSParam,
Enable:=MultiVSFilter1_Enable,
Enabled=>MultiVSFilter1_Enabled,
CalcRslt=>CalcCmdPos,
Error=>MultiVSFilter1_Error
);


//MC_SyncMoveAbsolute
Axis001_SyncMoveAbsolute
(Axis:=MC_Axis001,
Execute:=Axis001_SyncMoveAbsolute_Execute,
Position:=CalcCmdPos,
Direction:=_eMC_DIRECTION#_mcShortestWay,
InPosition=>Axis001_SyncMoveAbsolute_InPos,
Busy=>Axis001_SyncMoveAbsolute_Busy,
CommandAborted=>Axis001_SyncMoveAbsolute_CommandAborted,
Error=>Axis001_SyncMoveAbsolute_Error
);

//MC_Stop
Axis001_Stop
(Axis:=MC_Axis001,
Execute:=Axis001_Stop_Execute);
```

Additional Information

Refer to *TimeToMoveParam1* on page 99 for the sample programming that uses this function block and operates the servo axis set to Linear Mode in the absolute positioning.

VSConstTimeProfile1

The function block calculates the position profile for the high level S-curve with a specified travel time.

Function block name	Name	FB/ FUN	Graphic expression	ST expression
VSConstTimeProfile1	Time Specific Vibration Suppression Profile 1	FB		VSConstTimeProfile1 (Axis:=, Execute:=, Distance:=, VSConstTimeInput:=, TargetTravelTime:=, Abort:=, AbortMode:=, Done=>, Position=>, VSConstTimeOutput=>, CalcTravelTime=>, TravelTimeOver=>, Busy=>, CommandAborted=>, Error=>, ErrorID=>, ErrorIDEx=>,)

Precautions for Correct Use

With some damping control libraries, you cannot set the Count Mode for the axis to Rotary Mode.

Function Block and Function Information

Item	Description
Library file name	OmronLib_VS_Toolbox_V1_0.slr
Namespace	OmronLib\VS_Toolbox
Function block and function number	00030
Source code published/not published	Not published
Function block and function version	1.00

Variables

Input Variables

Name	Meaning	Data type	Initial value	Valid range	Unit	Description
Execute	Execute	BOOL	FALSE	TRUE or FALSE	---	The function block is executed when the variable changed to TRUE.
Distance	Travel Distance	LREAL	0	Positive or negative number	Command unit	Specifies the travel distance from the current position.
VSConstTimeInput	Time Specific VS Input	OmronLib\VS_Toolbox\sVS_CONST_TIME_INPUT	---	---	---	Specifies the input value to the time specific vibration suppression profile.
TargetTravelTime*1	Target Travel Time	LREAL	0	0 ≤ Set value ≤100,000	ms	Specifies the travel time. When it is 0, the value determined from velocity, acceleration/deceleration, jerk, and jerk differential limits is applied.
Abort	Abort	BOOL	FALSE	TRUE or FALSE	---	Aborts the processing according to the stopping method specified by the *AbortMode* (Stopping Mode Selection) input variable.
AbortMode*2	Stopping Mode Selection	UINT	0	0: Immediate stop	---	Specifies the stopping method applied when *Abort* is enabled.

*1. The position profile is calculated by the time rounded up to an integer multiple of the task period of the task in which this function block is used.

*2. Only 0 : Immediate stop is selectable.

Precautions for Correct Use

The input parameter values must be within the valid range.

Output Variables

Name	Meaning	Data type	Valid range	Unit	Description
Done	Done	BOOL	TRUE or FALSE	---	TRUE when the function block execution is completed.
Position	Position	LREAL	Depends on data types.	---	Gives a relative value from the position when the function block is executed as the command current position.
VSConstTime-Output	Time Specific VS Output	OmronLib\VS_Toolbox\sVS_CONST_TIME_OUTPUT	---	---	This is the output value as the time specific vibration suppression profile.
CalcTravel-Time	Calculated Travel Time	LREAL	Positive number or 0	ms	Gives the travel time determined by velocity, acceleration/deceleration, jerk, and jerk differential limits after rounding it up to an integer multiple of the task period of the task in which the function block is used.
TravelTime-Over	Travel Time Exceeded	BOOL	TRUE or FALSE	---	TRUE when *TargeTravelTIme* (Target Travel Time) exceeds the time rounded up to an integer multiple of the task period of the task in which the function block is used.TRUE when a value for *TargetTravelTime* is specified to 0.
Busy	Executing	BOOL	TRUE or FALSE	---	TRUE when the function block is acknowledged.
Command-Aborted	Instruction Aborted	LREAL	TRUE or FALSE	---	TRUE when the function block is aborted.
Error	Error	BOOL	TRUE or FALSE	---	TRUE while there is an error.
ErrorID	Error Code	WORD	*1	---	Contains the error code when an error occurs. 16#0 indicates a normal execution.
ErrorIDEx	Expansion Error Code	DWORD	*1	---	Contains the error code when an error occurs. 16#0 indicates a normal execution.

*1. Refer to *Troubleshooting* on page 78 for error codes.

Additional Information

The command unit is indicated either in [mm], [µm], [nm], [degree], [inch], or [pulse].

For details, refer to *Unit Conversion Settings* in the motion control user's manual.

In-Out Variables

Name	Meaning	Data type	Valid range	Description
Axis	Axis	_sAXIS_REF	---	Specifies the axis.*1

*1. Use the axis variable name (MC_Axis*** by default) that you created on the Axis Basic Setting display of the Sysmac Studio as a user-defined variable or the axis variable name (_MC_AX[**]) as a system-defined variable.

Structures

The data type of the *VSConstTimeInput* input variable to this function block is OmronLib\VS_Toolbox\sVS_CONST_TIME_INPUT. The table below gives the specifications.

Variable	Meaning	Description	Data type	Valid range	Unit	Initial value
VSConstTimeInput	Time Specific VS Input		OmronLib\VS_Toolbox\sVS_CONST_TIME_INPUT	---	---	---
VelocityLimit	Velocity Limit	Specifies the velocity limit.	LREAL	Positive number	Command unit/s	0
AccDecLimit	Acceleration/Deceleration Limit	Specifies the acceleration/deceleration limit.	LREAL	Positive number	Command unit/s^2	0
JerkLimit	Jerk Limit	Specifies the jerk limit.	LREAL	Positive number	Command unit/s^3	0
DiffJerkLimit	Jerk Differential Limit	Specifies the jerk differential limit.	LREAL	Positive number	Command unit/s^4	0

Precautions for Correct Use

When the *VelocityLImit, AccDecLimit, JerkLimit*, and *DiffJerkLimit* are set, the actual travel time may become longer than *TargetTravelTime.*

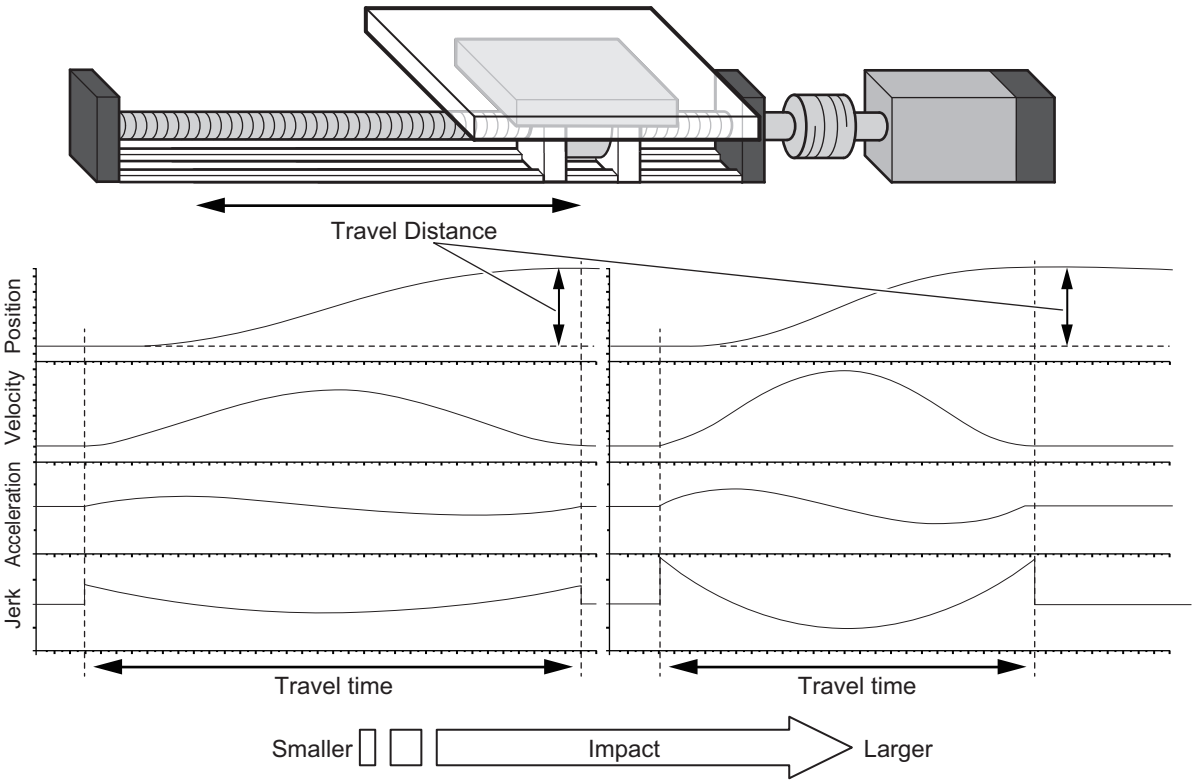
When this occurs, *TravelTimeOver* changes to TRUE, and the new travel time is given to *CalcTravelTime*.

The data type of the *VSConstTimeOutput* output variable to this function block is OmronLib\VS_Toolbox\sVS_CONST_TIME_OUTPUT. The table below gives the specifications.

Variables		Meaning	Description	Data type	Valid range	Unit
VSConstTime-Output		Time Specific VS Output		OmronLib\VS_Toolbox\sVS_CONST_TIME_OUTPUT	---	---
	MaxVelocity	Maximum Velocity	Gives the maximum velocity during operation.	LREAL	Positive number or 0	Command unit/s
	MaxAccDec	Maximum Acceleration/Deceleration	Gives the maximum acceleration/deceleration during operation.	LREAL	Positive number or 0	Command unit/s^2
	MaxJerk	Maximum Jerk	Gives the maximum jerk during operation.	LREAL	Positive number or 0	Command unit/s^3
	MaxDiffJerk	Maximum Jerk Differential Value	Gives the maximum jerk differential during operation.	LREAL	Positive number or 0	Command unit/s^4

Function

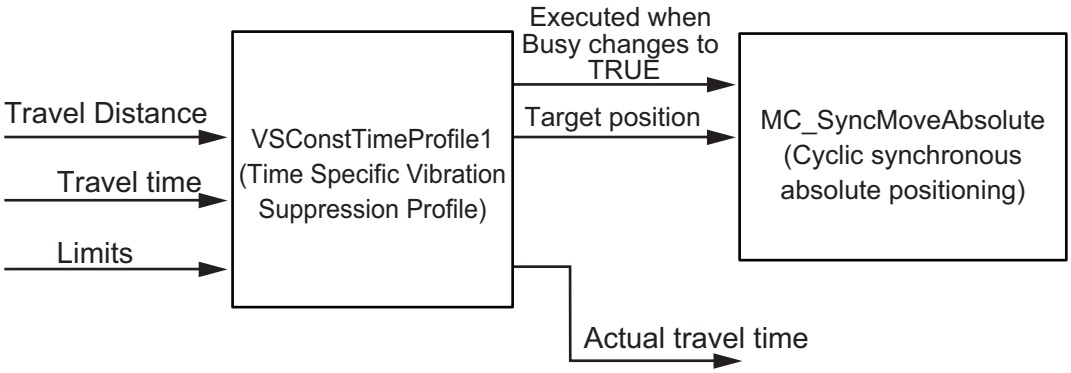
The function block calculates the position profile for the high level S-curve with a specified travel time. With this function block, positioning can be made in smooth velocity and acceleration/deceleration from the start to end points.



A position profile is given every task period during motion from the start point where the function block is executed to the end point (goal) that is specified by the target travel time.

If the axis cannot reach the end point (goal) within the specified target travel time because of the input values to this function block or of a restriction caused by the axis operation setting, the position profile that satisfies all restrictions specified with the input variables are given.

The position profile that is given during the function block execution is used as *Position* (Set Point) to the MC_SyncMoveAbsolute (Cyclic Synchronous Absolute Positioning) motion control instruction to operate the axis.

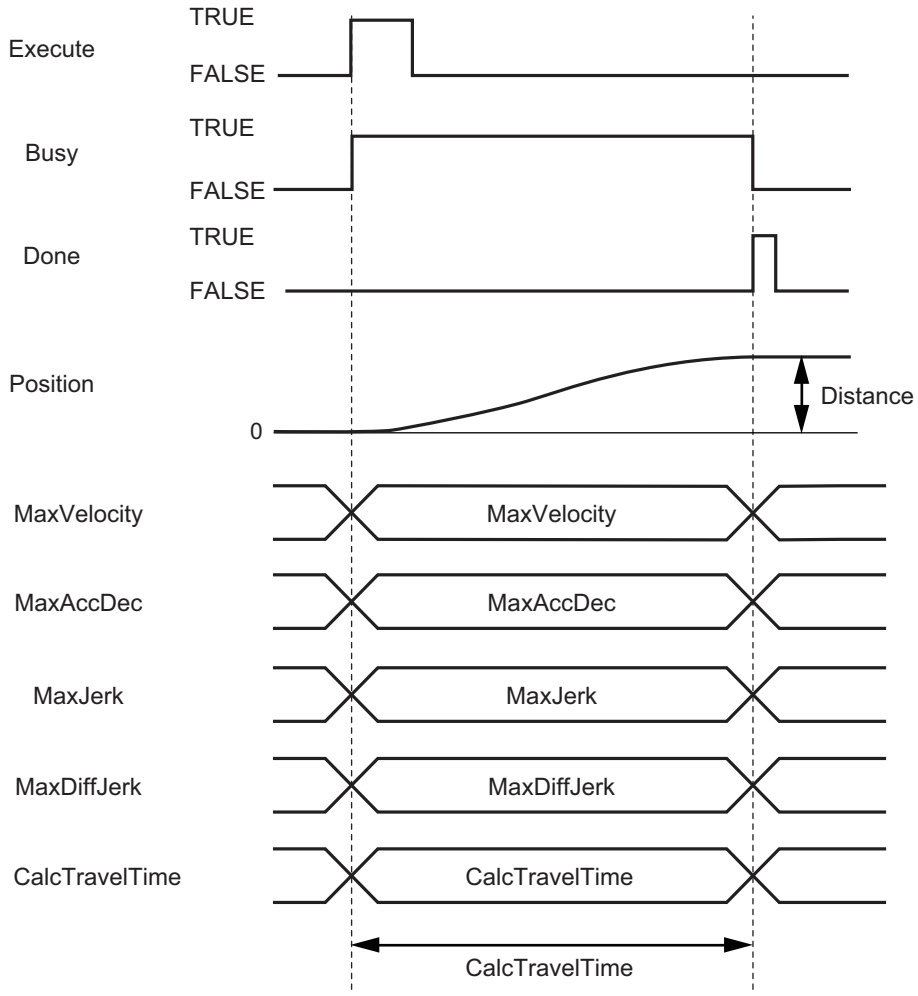


Function in Detail

This function block performs processing as follows.

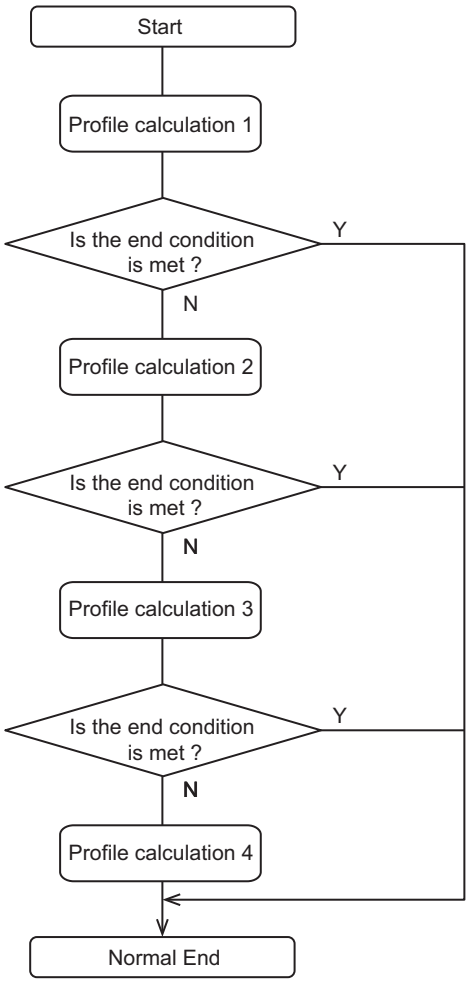
- Outputs the S-curve position command value that uses a high-level function to *Position*, with the value of *Distance* as the target position. The value of *Position* is a relative value from the position when this function block is executed.
- Starts operation when *Execute* changes to TRUE.
- Outputs the maximum velocity, maximum acceleration, maximum jerk, and maximum jerk differential during operation to *MaxVelocity*, *MaxAccDec*, *MaxJerk*, and *MaxDiffJerk*, respectively.
- Outputs the travel time to *CalcTravelTime* (Calculated Travel Time).

An operation example is provided below.



Position Profile Calculation

A position profile is calculated as follows.



The table below gives the descriptions of each profile calculation in the above flowchart.

Calculation	Description
Profile calculation 1	Calculates the profile for the high level S-curve (with no constant velocity zone) according to the setting of *TargetTravelTime* (Target Travel Time).
Profile calculation 2	Calculates the profile for the high level S-curve (with a constant velocity zone) according to the setting of *TargetTravelTime* (Target Travel Time). The maximum velocity is the value specified for *VelocityLimit* (Velocity Limit).
Profile calculation 3	Calculates the profile for the high level S-curve (with a constant velocity zone) with *CalcTravelTime* (Calculated Travel Time) as the minimum value.
Profile calculation 4*1	Calculates the profile for the high level S-curve (with no constant velocity zone) with *CalcTravelTime* (Calculated Travel Time) as the minimum value.

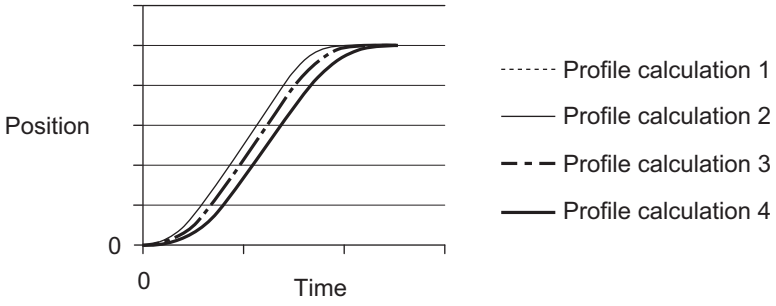
*1. This calculation will be performed if the velocity does not reach the *VelocityLimit* (Velocity Limit) in profile calculation 3 due to limits other than *VelocityLimit*.

Also note that, in the flowchart, the end condition for each profile calculation is met when the calculated velocity, acceleration/deceleration, jerk, or jerk differential limit value is equal to or less than *VelocityLimit* (Velocity Limit), *AccDecLimit* (Acceleration/Deceleration Limit), *JerkLimit* (Jerk Limit), or *DiffJerkLimit* (Jerk Differential Limit).

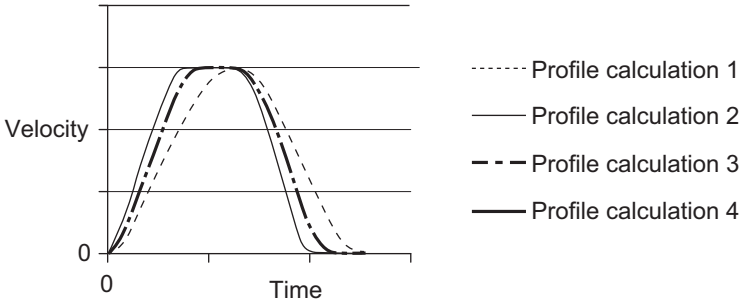
Comparison of Profile Calculation Results

The following graphs show the difference among the position, velocity, acceleration/deceleration, jerk, and jerk differential calculated in profile calculations 1 to 4.

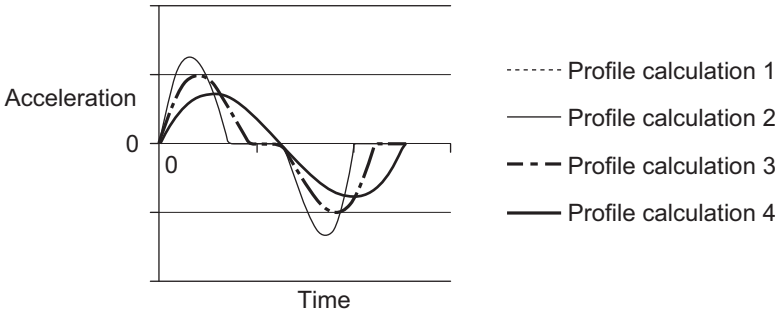
● Position



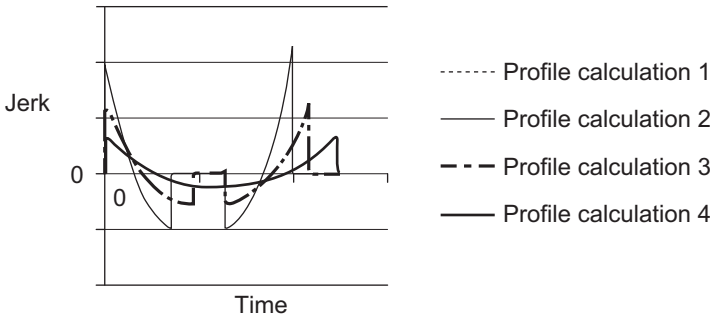
● Velocity



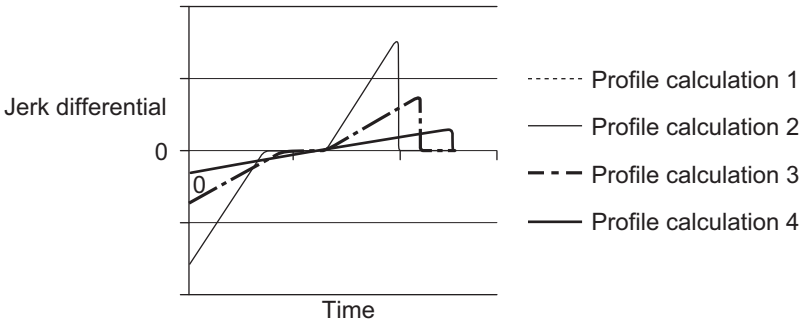
● Acceleration



● Jerk



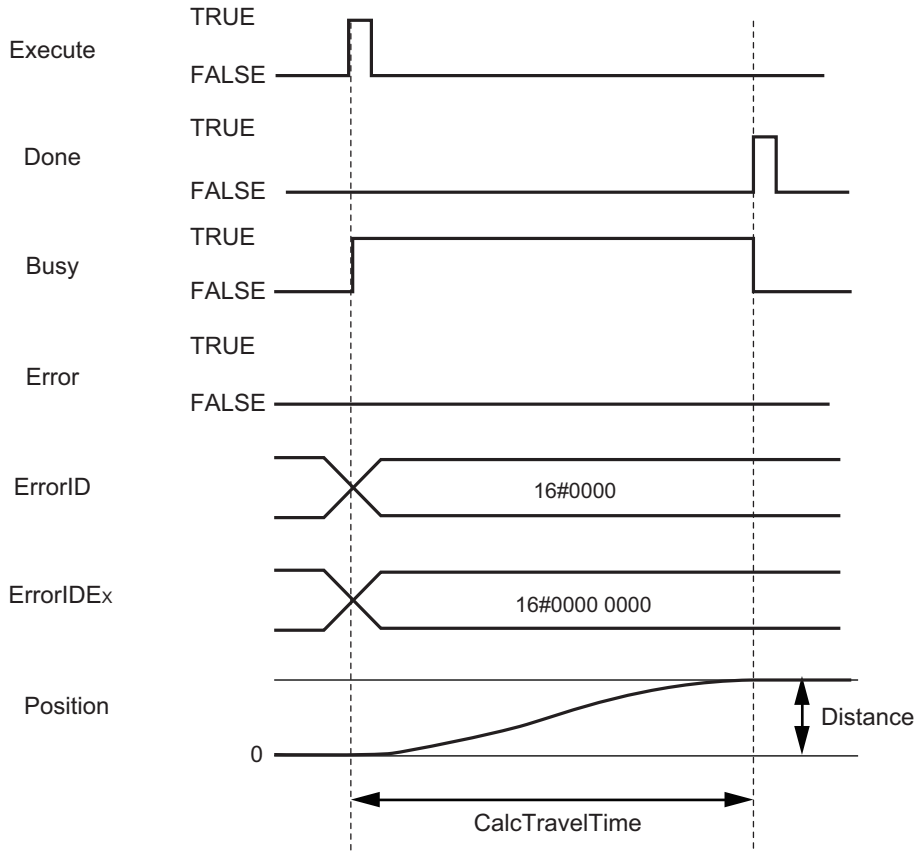
● Jerk differential



Timing Charts

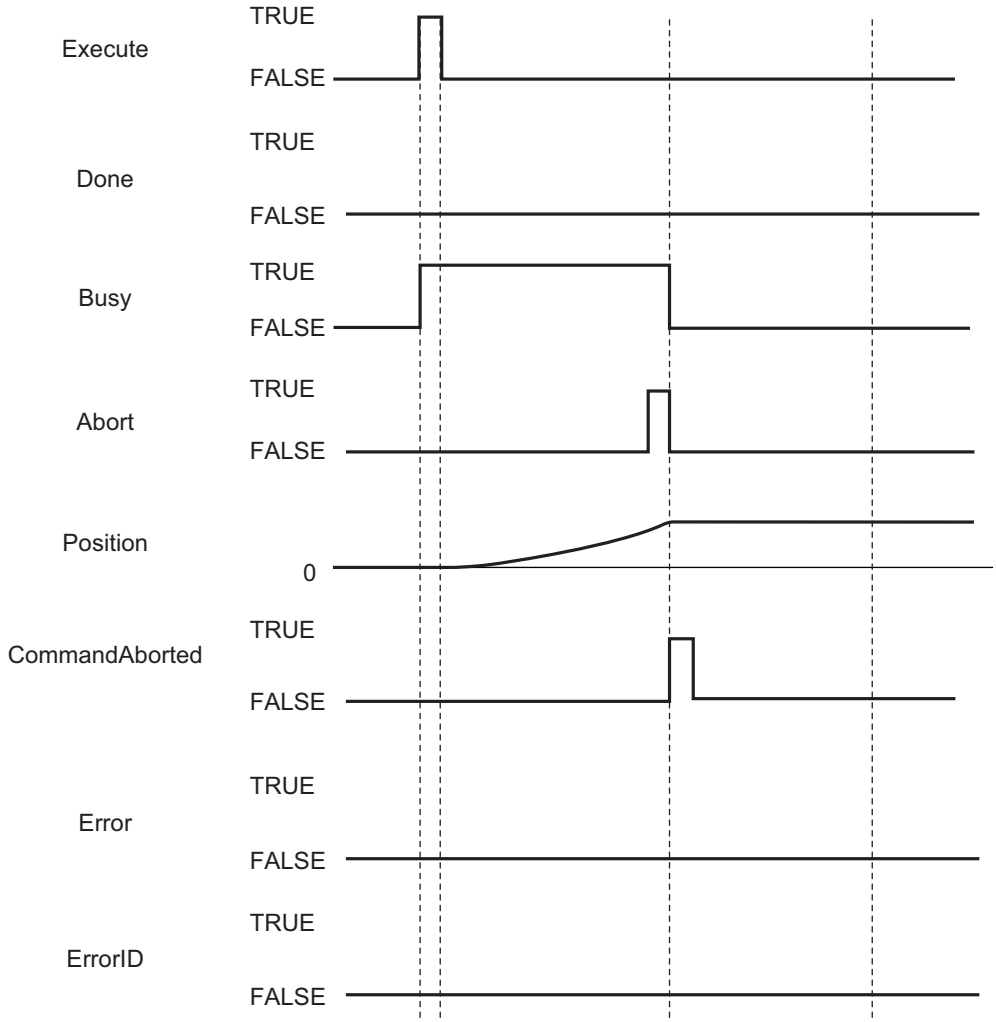
Normal End

- When *Execute* changes to TRUE, *Busy* (Executing) changes to TRUE and position profile calculation starts.
- *Done* changes to TRUE when the position profile calculation is completed for the distance specified by *Distance*.



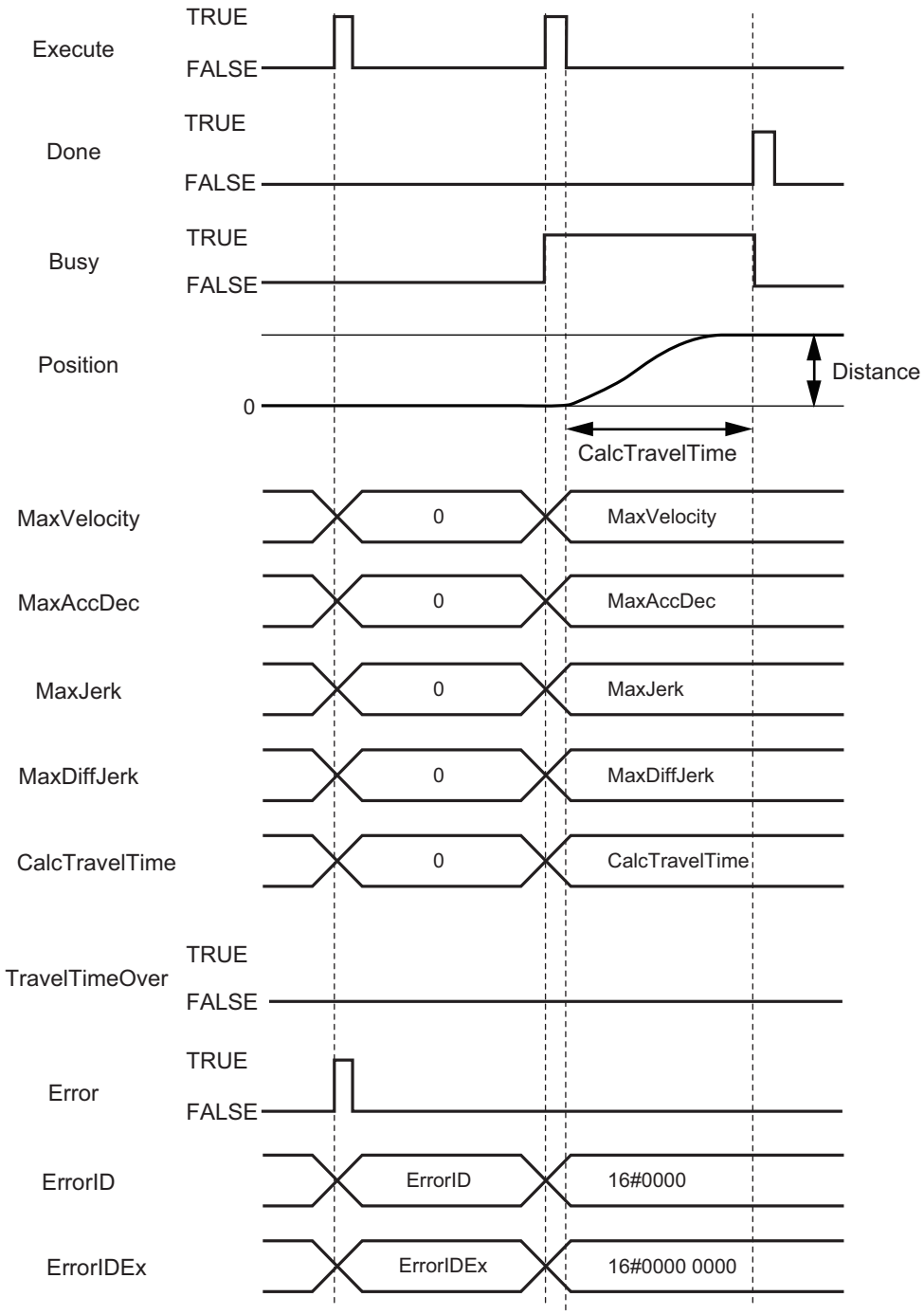
Normal End, and Stop by Abort

When *Abort* is changed to TRUE during the execution of this function block, the position profile calculation is stopped. In this case, *CommandAborted* (Instruction Aborted) is given. *Position* retains the value given when *Abort* changes to TRUE.



Changing from Error End to Normal End

If an error occurs during function block execution, *Error* will change to TRUE. You can find out the cause of the error by referring to the values output by *ErrorID* (Error Code) and *ErrorIDEx* (Expansion Error Code). When *Execute* to this function block changes to TRUE, *ErrorID* (Error Code) and *ErrorIDEx* (Expansion Error Code) are cleared.



Troubleshooting

Error Code	Expansion Error Code	Status	Description	Correction
16#0000	16#00000000	Normal End		
16#3C16	16#00000001	Input Parameter Out of Range	The input parameter is out of range. The value of *TargetTravelTime* or *Distance* is 0.	The input parameter values must be within the valid range.
	16#00000002	Calculated Travel Time Out of Range	The calculated travel time is out of the valid range for LREAL data. • Any of the values set to the following input parameters is too small. *VelocityLimit, AccDecLimit, JerkLimit, DiffJerkLimit* • The value set to the following input parameter is too large. *Distance*	• Set larger values to the following input parameters. *VelocityLimit, AccDecLimit, JerkLimit, DiffJerkLimit* • Set a smaller value to the following input parameter. *Distance*
	16#00000003	Calculated Command Position Out of Range	The calculated travel time is out of the valid range for LREAL data. • The value set to the following input parameter is too large. *Distance*	• Set a smaller value to the following input parameter. *Distance*
	16#00000005	Instruction Execution Error Caused by Axis Type	The axis type of the axis specified with *Axis* is set to an encoder axis or a virtual encoder axis.	Set the axis specified with *Axis* to a servo axis or a virtual servo axis.
	16#00000006	Instruction Execution Error Caused by Count Mode Setting	A Count Mode other than Linear Mode was specified for the axis specified by *Axis*.	Set the axis specified with *Axis* to Linear Mode.
	16#00000007	Incorrect Unit Version	The unit version of CPU Unit or Industrial PC is invalid.	Use the CPU Unit or Industrial PC with the unit version which supports this function block.
	16#00000008	Incorrect Task Setting	The task that executes this function block is invalid.	Execute this function block in periodic task.

Sample Programming

Program Operation

For the Axis 0, perform relative positioning in which you specify the travel distance from the current position and absolute positioning in which you specify the target position in absolute coordinates.

This sample program consists of the following three parts.

- Homing
- Relative positioning
- Absolute positioning

Preconditions

The sample program assumes the following status.

- The Axis 0 (axis variable *MC_Axis000)* is operable and the home is defined.

Precautions for Correct Use

Set the Count Mode to Linear Mode for an axis that is specified for Axis 0 (axis variable *MC_Axis000*). If you set Rotary Mode, an Instruction Execution Error Caused by Count Mode Setting (ErrorIDEx =16#0000 0006) will occur at execution.

Ladder Diagram(Homing)

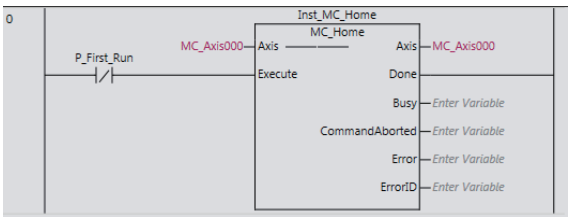
● Internal Variables

Name	Data type	Initial value	Comment
Inst_MC_Home	MC_Home		

● External Variables

Name	Data type	constant	Comment
MC_Axis000	_sAXIS_REF	✓	

● Programming



Ladder Diagram(Relative Positioning)

● Internal Variables

Name	Data type	Initial value	Comment
Inst_VSConstTimeProfile1	OmronLib\VS_Toolbox\VSConstTime-Profile1		Instance of VSConstTimeProfile1 FB
Start	BOOL		Start
StartPosition	LREAL		Start Position
Distance	LREAL		Distance
VSConstTimeInput	OmronLib\VS_Toolbox\sVS_CONST_-TIME_INPUT		Time Specific VS Input
TargetTravelTime	LREAL		Target Travel Time
Abort	BOOL		Abort
AbortMode	UINT		Stopping Mode Selection
Done	BOOL		Done
Position	LREAL		Position
VSConstTimeOutput	OmronLib\VS_Toolbox\sVS_CONST_-TIME_OUTPUT		Time Specific VS Output
CalcTravelTime	LREAL		Calculated Travel Time
TravelTimeOver	BOOL		Travel Time Exceeded
Busy	BOOL		Executing
CommandAborted	BOOL		Instruction Aborted
Error	BOOL		Error
ErrorID	WORD		Error Code
ErrorIDEx	DWORD		Expansion Error Code
AbsolutePosition	LREAL		Absolute Position
Inst_MC_SyncMoveAbsolute	MC_SyncMoveAbsolute		
Direction	_eMC_DIRECTION		
BufferMode	_eMC_BUFFER_MODE		
SyncAbs_Busy	BOOL		
SyncAbs_Active	BOOL		
SyncAbs_CommandAborted	BOOL		
SyncAbs_Error	BOOL		
SyncAbs_ErrorIDEx	WORD		

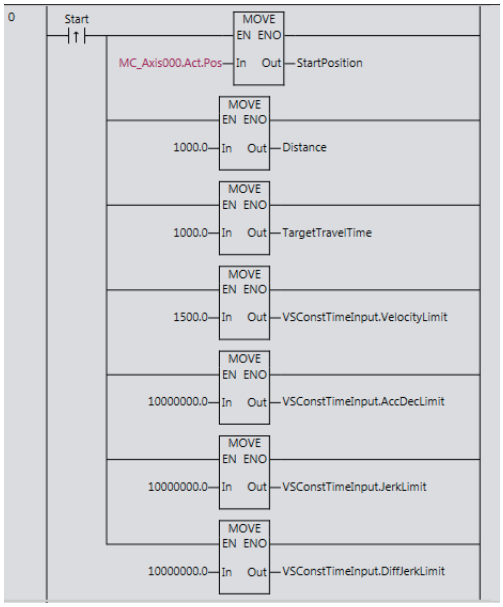
● External Variables

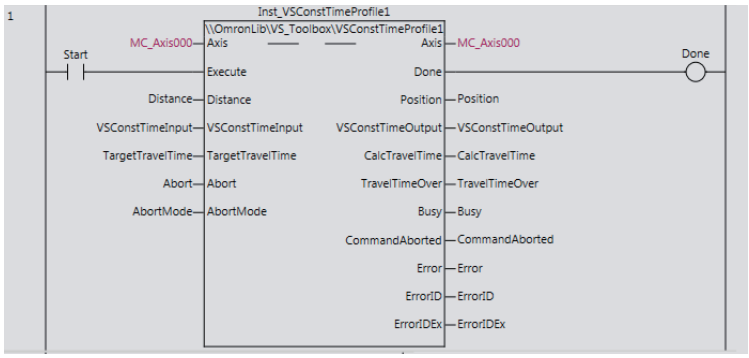
Name	Data type	constant	Comment
MC_Axis000	_sAXIS_REF	✓	

Programming

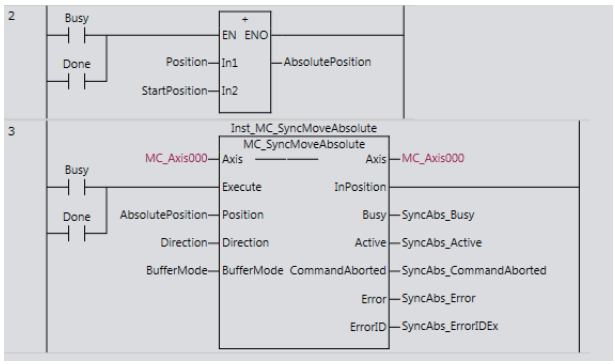
Since a relative travel is performed according to the command value by absolute coordinate, maintain the position at execution start.

Set the profile as follows.





Add the calculation result of this function block to the current position at execution start, and convert it to the absolute coordinate command position.



Ladder Diagram(Absolute Positioning)

● Internal Variables

Name	Data type	Initial value	Comment
Start	BOOL		Start
StartPosition	LREAL		Start Position
Distance	LREAL		Distance
TargetTravelTime	LREAL		Target Travel Time
VSConstTimeInput	OmronLib\VS_Toolbox\sVS_CONST_-TIME_INPUT		Time Specific VS Input
Abort	BOOL		Abort
AbortMode	UINT		Stopping Mode Selection
Position	LREAL		Position
VSConstTimeOutput	OmronLib\VS_Toolbox\sVS_CONST_-TIME_OUTPUT		Time Specific VS Output
CalcTravelTime	LREAL		Calculated Travel Time
TravelTimeOver	BOOL		Travel Time Exceeded
Busy	BOOL		Executing
CommandAborted	BOOL		Instruction Aborted
Error	BOOL		Error
ErrorID	WORD		Error Code
ErrorIDEx	DWORD		Expansion Error Code
Inst_VSConstTimeProfile1	OmronLib\VS_Toolbox\\VSConstTime-Profile1		Instance of VSConstTimeProfile1 FB
Done	BOOL		Done
AbsolutePosition	LREAL		Absolute Position
Direction	_eMC_DIRECTION		
BufferMode	_eMC_BUFFER_MODE		
SyncAbs_Busy	BOOL		
SyncAbs_Active	BOOL		
SyncAbs_CommandAborted	BOOL		
SyncAbs_Error	BOOL		
SyncAbs_ErrorIDEx	WORD		
Inst_MC_SyncMoveAbsolute	MC_SyncMoveAbsolute		
TargetPosition	LREAL		Target Position

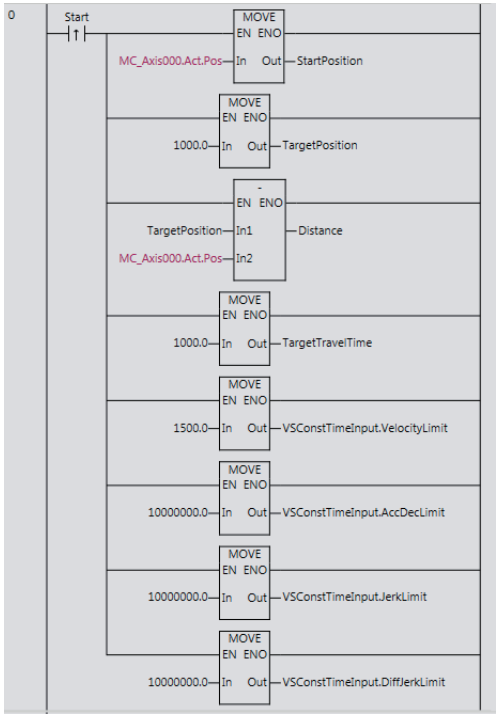
● External Variables

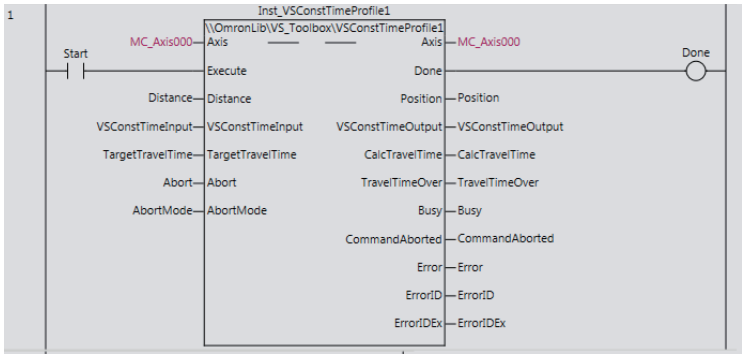
Name	Data type	constant	Comment
MC_Axis000	_sAXIS_REF	✓	

Programming

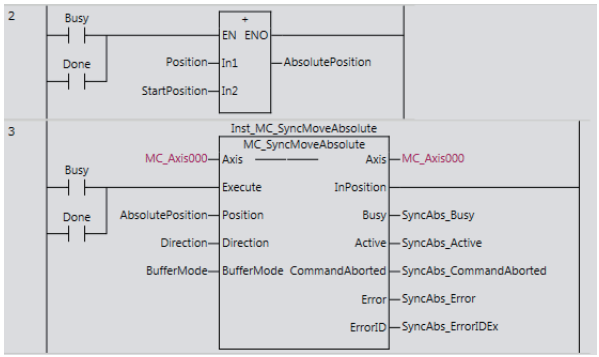
Convert a command value by absolute coordinate to the relative value.

Set the profile as follows.



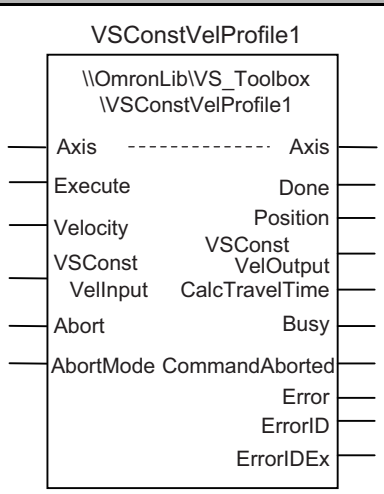


Add the calculation result of this function block to the current position at execution start, and convert it to the absolute coordinate command position.



VSConstVelProfile1

The function block calculates the position profile with the specified velocity in constant velocity zone.

Function block name	Name	FB/ FUN	Graphic expression	ST expression
VSConstVelProfile1	Constant Velocity Specific VS Profile 1	FB		VSConstVelProfile1_instance (Axis:=, Execute:=, VSConstVelInput:=, Abort:=, AbortMode:=, Done=>, Position=>, VSConstVelOutput=>, CalcTravelTime=>, Busy=>, CommandAborted=>, Error=>, ErrorID=>, ErrorIDEx=>)

Precautions for Correct Use

With some damping control libraries, you cannot set the Count Mode for the axis to Rotary Mode.

Function Block and Function Information

Item	Description
Library file name	OmronLib_VS_Toolbox_V1_0.slr
Namespace	OmronLib\VS_Toolbox
Function block and function number	00031
Source code published/not published	Not published
Function block and function version	1.00

Variables

Input Variables

Name	Meaning	Data type	Initial value	Valid range	Unit	Description
Execute	Execute	BOOL	FALSE	TRUE or FALSE	---	The function block is executed when the variable changed to TRUE.
Velocity	Velocity in Constant Velocity Zone	LREAL	0	Positive number	Command unit/s	Specifies the maximum velocity in the constant velocity zone.
VSConstVelInput	VS Constant Velocity Profile Input	OmronLib\VS_Toolbox\sVS_CONST_VEL_INPUT	---	---	---	Specifies the input value to the constant velocity zone specific profile.
Abort	Abort	BOOL	FALSE	TRUE or FALSE	---	Aborts the processing according to the stopping method specified by the *AbortMode* (Stopping Mode Selection) input variable.
AbortMode*1	Stopping Mode Selection	UINT	0	0: Immediate stop	---	Specifies the stopping method applied when *Abort* is enabled.

*1. Only 0 : Immediate stop is selectable.

Precautions for Correct Use

The input parameter values must be within the valid range.

Additional Information

The command unit is indicated either in [mm], [µm], [nm], [degree], [inch], or [pulse].

For details, refer to *Unit Conversion Settings* in the motion control user's manual.

Output Variables

Name	Meaning	Data type	Valid range	Unit	Description
Done	Done	BOOL	TRUE or FALSE	---	TRUE when the function block execution is completed.
Position	Position	LREAL	Depends on data types.	---	Gives a relative value from the position when the function block is executed as the command current position.
VSConstVelOutput	VS Constant Velocity Profile Output	OmronLib\VS_Toolbox\sVS_CONST_VEL_OUTPUT	---	---	This is the output value as the constant velocity zone specific profile.
CalcTravelTime	Calculated Travel Time	LREAL	Positive number or 0	---	Gives the travel time determined by the input values to this function block after rounding it up to an integer multiple of the task period of the task in which the function block is used. Any settings below 0.001 ms are truncated.
Busy	Executing	BOOL	TRUE or FALSE	---	TRUE when the function block is acknowledged.
CommandAborted	Instruction Aborted	BOOL	TRUE or FALSE	---	TRUE when the function block is aborted.
Error	Error	BOOL	TRUE or FALSE	---	TRUE while there is an error.
ErrorID	Error Code	WORD	*1	---	Contains the error code when an error occurs. 16#0 indicates a normal execution.
ErrorIDEx	Expansion Error Code	DWORD	*1	---	Contains the error code when an error occurs. 16#0 indicates a normal execution.

*1. Refer to *Troubleshooting* on page 93 for error codes.

In-Out Variables

Name	Meaning	Data type	Valid range	Description
Axis	Axis	_sAXIS_REF	---	Specifies the axis.*1

*1. Use the axis variable name (MC_Axis*** by default) that you created on the xxxx display of the Sysmac Studio as a user-defined variable or the axis variable name (_MC_AX[**]) as a system-defined variable.

Structures

The data type of the *VSConstVelInput* input variable to this function block is OmronLib\VS_Toolbox\sVS_CONST_TIME_INPUT. The table below gives the specifications.

Variables	Meaning	Description	Data type	Valid range	Unit	Initial value
VSConstVelInput	VS Constant Velocity Profile Input		OmronLib\VS_Toolbox\sVS_CONST_VEL_INPUT	---	---	---
AccDistance*1	Acceleration Distance	Specifies the acceleration distance.	LREAL	Positive or negative number	---	0
ConstVelDistance*1	Constant Velocity Distance	Specifies the constant velocity distance.	LREAL	Positive or negative number	---	0
DecDistance*1	Deceleration Distance	Specifies the deceleration distance.	LREAL	Positive or negative number	---	0

*1. Specify the same sign to the values of *AccDistance, ConstVelDistance,* and *DecDistance.* An error occurs if a different sign is specified.

Precautions for Correct Use

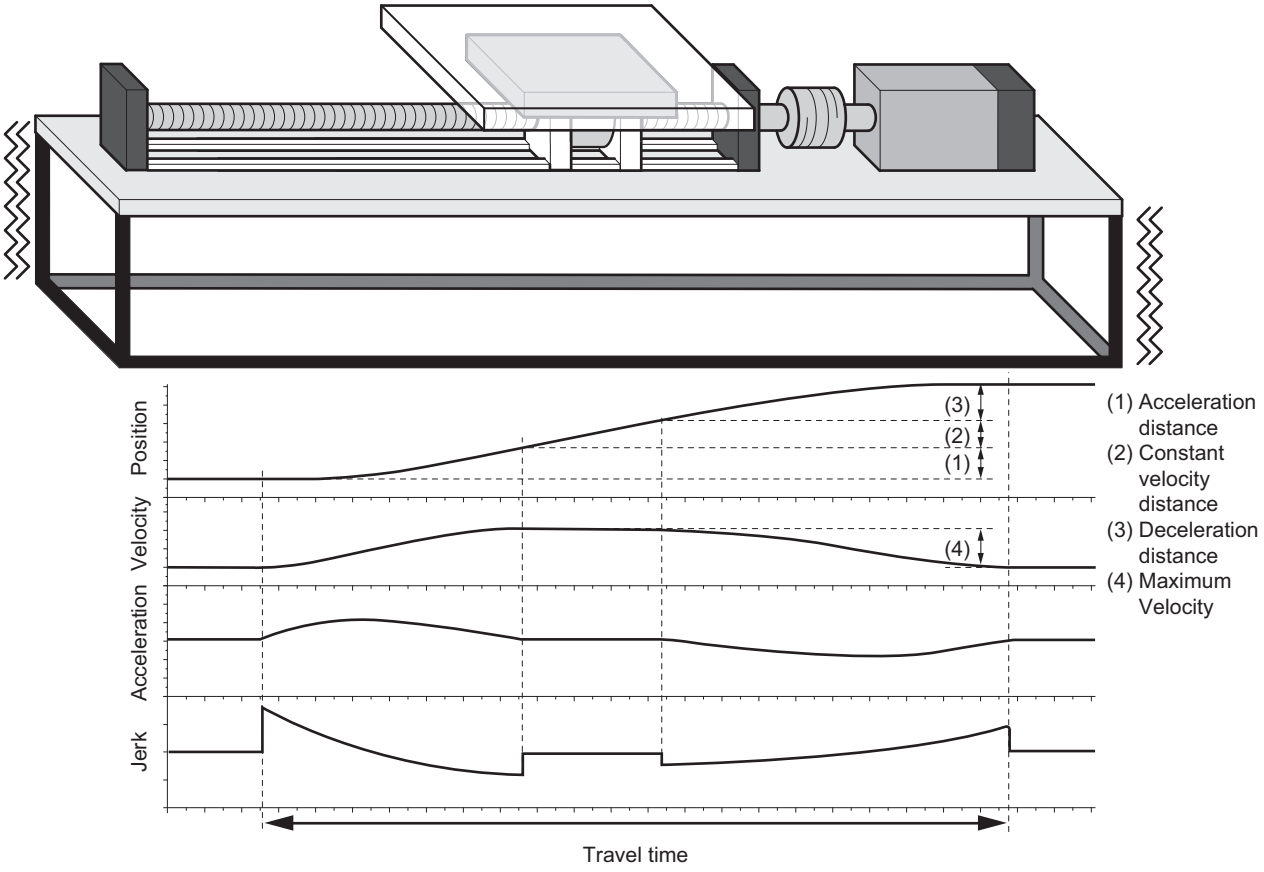
If the acceleration or deceleration distance is shorter in regard to the command velocity, greater vibration may be given to the equipment. When you set a command velocity, or an acceleration or deceleration distance, ensure safety by performing test runs after making small changes to these parameters.

The data type of the *VSConstVelOutput* output variable from this function block is OmronLib\VS_Toolbox\sVS_CONST_VEL_OUTPUT. The table below gives the specifications.

Variables	Meaning	Description	Data type	Valid range	Unit
VSConstVelOutput	VS Constant Velocity Profile Output		OmronLib\VS_Toolbox\sVS_CONST_VEL_OUTPUT	---	---
MaxAcc	Maximum Acceleration	Gives the maximum acceleration during operation.	LREAL	Positive number or 0	Command unit/s^2
MaxDec	Maximum Deceleration	Gives the maximum acceleration during operation.	LREAL	Positive number or 0	Command unit/s^2
MaxAccJerk	Maximum Deceleration Jerk	Gives the maximum jerk in the acceleration zone of operation.	LREAL	Positive number or 0	Command unit/s^3
MaxDecJerk	Maximum Deceleration Jerk	Gives the maximum jerk in the deceleration zone of operation.	LREAL	Positive number or 0	Command unit/s^3

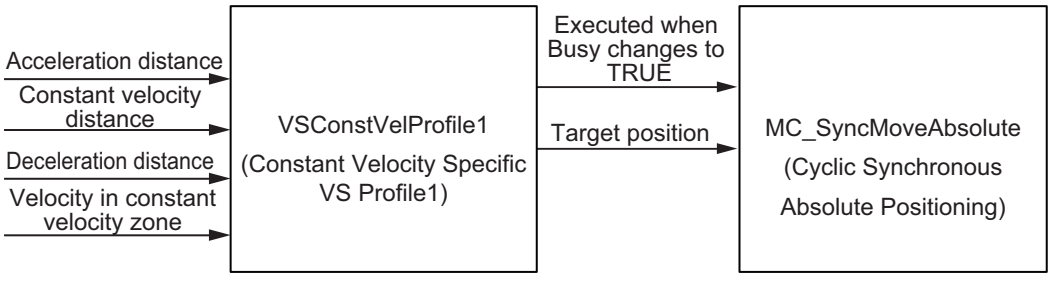
Function

The function block calculates a smooth position profile for each of the acceleration and deceleration distances, so that residual vibration at the stopping position can be reduced and the vibration during constant velocity motion can also be suppressed.



● Control Block Diagram

When *AccDistance* (Acceleration Distance), *ConstVelDistance* (Constant Velocity Distance), *DecDistance* (Deceleration Distance), and *Velocity* (Velocity in Constant Velocity Zone) are specified, a position profile is given every task period during motion from the start point where the function block is executed to the end point (goal).

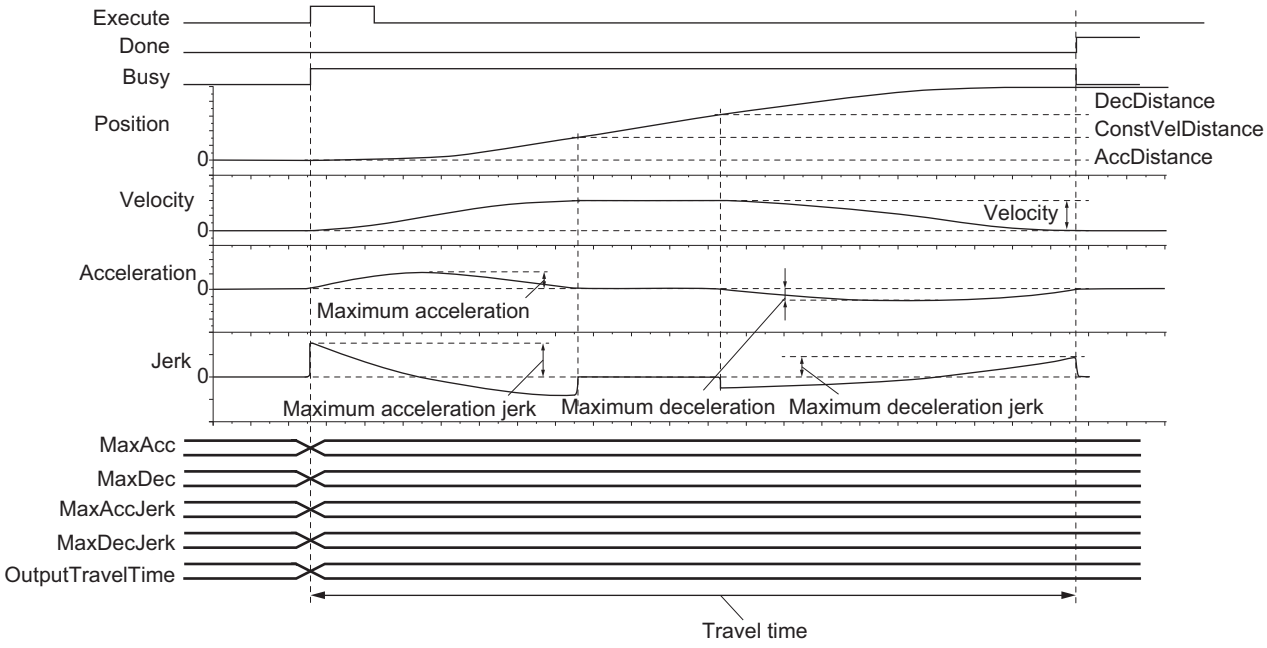


Function in Detail

This function block performs processing as follows.

- Outputs the S-curve position command value that uses a high-level function to *Position* in the operation pattern specified by *AccDistance* (Acceleration Distance), *ConstVelDistance* (Constant Velocity Distance), *DecDistance* (Deceleration Distance), and *Velocity* (Velocity in Constant Velocity Zone).
- Starts operation when *Execute* changes to TRUE.
- Outputs the maximum acceleration, maximum deceleration, maximum acceleration jerk, and maximum deceleration jerk during operation to *MaxAcc*, *MaxDec*, *MaxAccJerk*, and *MaxDecJerk*, respectively.
- Outputs the travel time to *CalcTravelTime* (Calculated Travel Time).

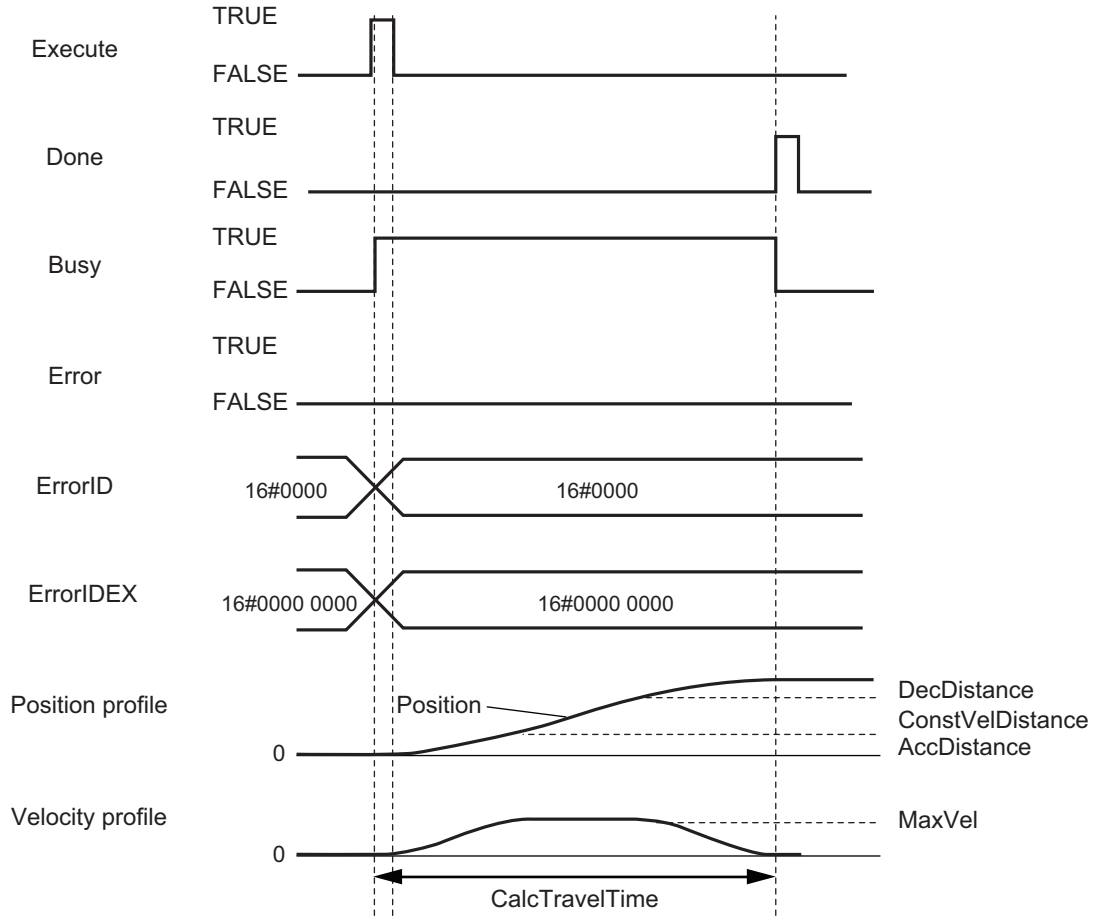
An operation example is provided below.



Timing Charts

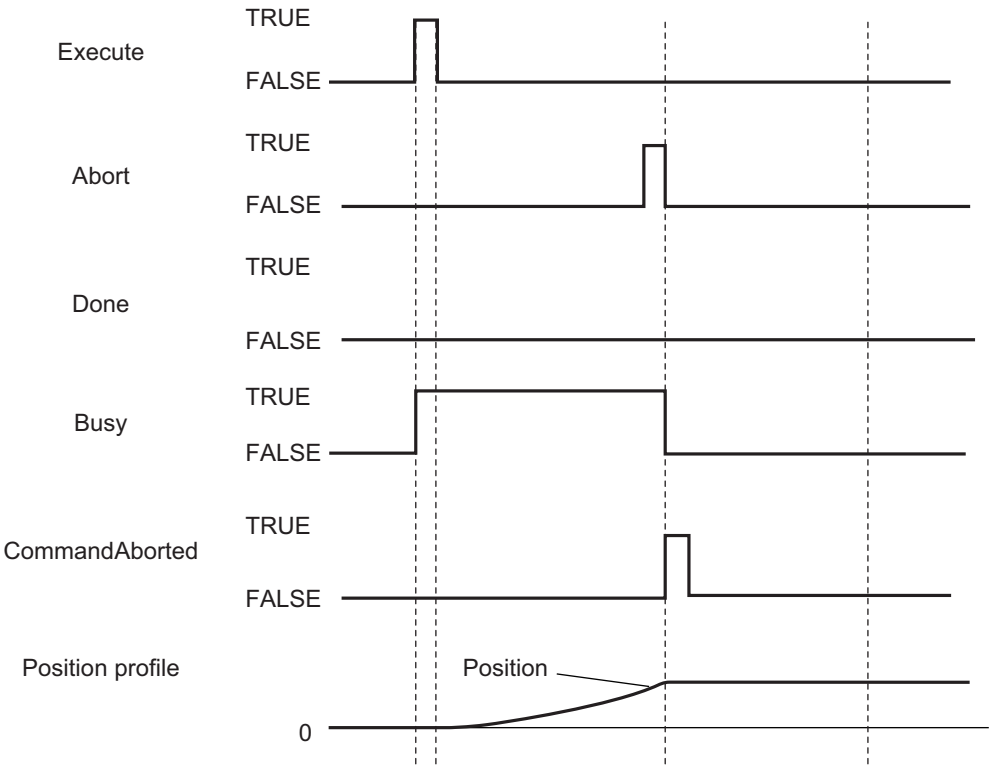
● Normal End

- When *Execute* changes to TRUE, *Busy* (Executing) changes to TRUE and position profile calculation starts.
- *Done* changes to TRUE when the position profile calculation is completed for the distance specified by *AccDistance (Acceleration Distance), ConstVelDistance* (Constant Velocity Distance), and *Dec-Distance* (Deceleration Distance).



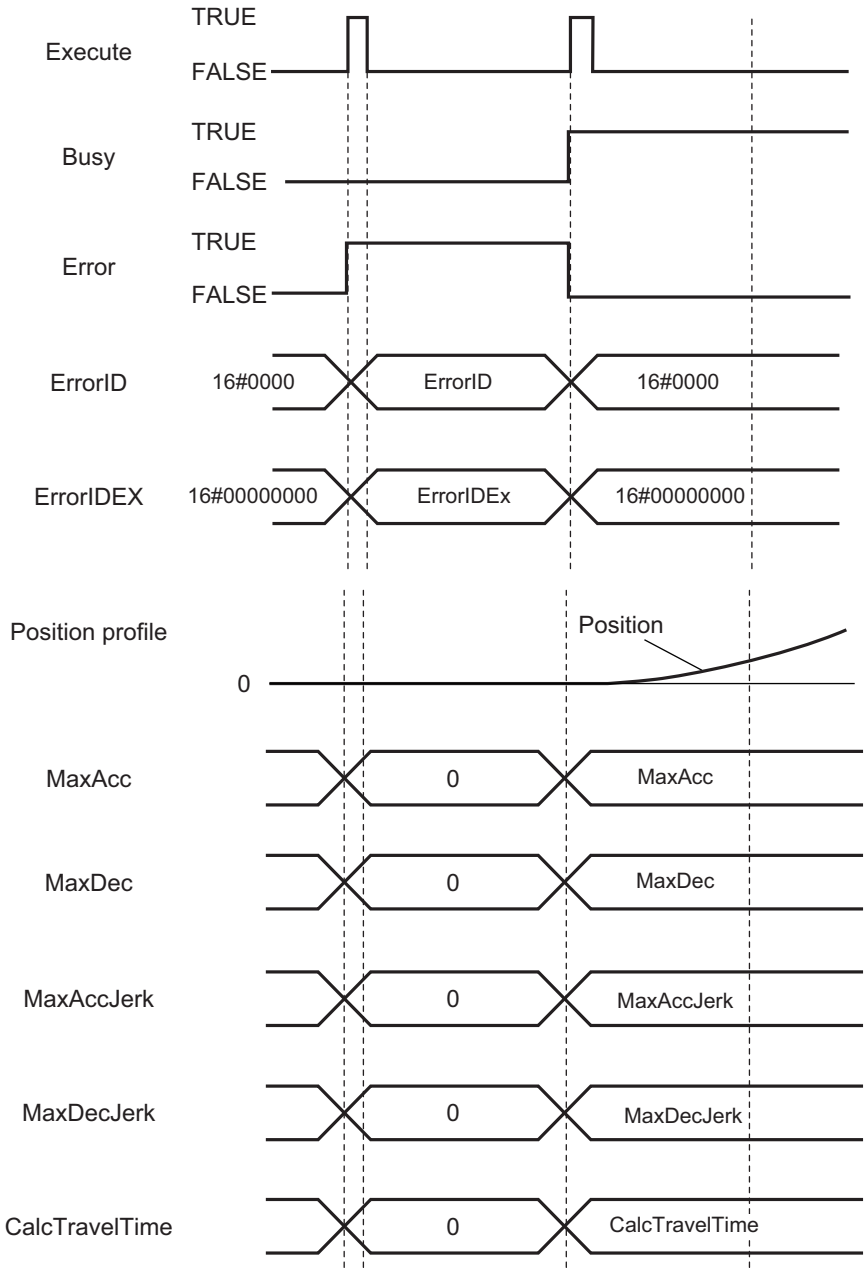
Normal End, and Stop by Abort

When *Abort* is changed to TRUE during the execution of the function block, the profile calculation is stopped. In this case, *CommandAborted* (Instruction Aborted) is given. *Position* retains the value given when *Abort* changes to TRUE.



Changing from Error End to Normal End

If an error occurs during function block execution, *Error* will change to TRUE. You can find out the cause of the error by referring to the values output by *ErrorID* (Error Code) and *ErrorIDEx* (Expansion Error Code). When *Execute* to this function block changes to TRUE, *ErrorID* (Error Code) and *ErrorIDEx* (Expansion Error Code) are cleared.



Troubleshooting

Error Code	Expansion Error Code	Status	Description	Correction
16#0000	16#00000000	Normal End		
16#3C17	16#00000001	Input Parameter Out of Range	The input parameter is out of range. All limits are set to 0, or the specified travel time is 0.	The input parameter values must be within the valid range.
	16#00000002	Incorrect VSConstVelInput Value	The values specified to *AccDistance, ConstVelDistance*, and *DecDistance* do not have the same sign.	Specify the same sign to the values of *AccDistance, ConstVelDistance,* and *DecDistance.*
	16#00000003	Calculated Travel Time Out of Range	The calculated travel time is out of the valid range for LREAL data. • Any of the values set to the following input parameters is too large. *AccDistance* *ConstVelDistance* *DecDistance* • The value set to the following input parameter is too small. *Velocity*	• Set smaller values to the following input parameters. *AccDistance* *ConstVelDistance* *DecDistance* • Set a larger value to the following input parameter. *Velocity*
	16#00000005	Instruction Execution Error Caused by Axis Type	The axis type of the axis specified with *Axis* is set to an encoder axis or a virtual encoder axis.	Set the axis specified with *Axis* to a servo axis or a virtual servo axis.
	16#00000006	Instruction Execution Error Caused by Count Mode Setting	A Count Mode other than Linear Mode was specified for the axis specified by *Axis*.	Set the axis specified with *Axis* to Linear Mode.
	16#00000007	Incorrect Unit Version	The unit version of CPU Unit or Industrial PC is invalid.	Use the CPU Unit or Industrial PC with the unit version which supports this function block.
	16#00000008	Incorrect Task Setting	The task that executes this function block is invalid.	Execute this function block in periodic task.

Sample Programming

Program Operation

The sample program uses the function block to perform relative positioning for the Axis 0 in which the travel distance from the current position is specified, and also absolute positioning which the target position in an absolute coordinate is specified.

Preconditions

The sample program assumes the following status.

- The Axis 0 (axis variable *MC_Axis000*) is operable and the home is defined.

Precautions for Correct Use

Set the Count Mode to Linear Mode for an axis that is specified for Axis 0 (axis variable *MC_Axis000*). If you set Rotary Mode, an Instruction Execution Error Caused by Count Mode Setting (ErrorIDEx =16#0000 0006) will occur at execution.

Ladder Diagram(Homing)

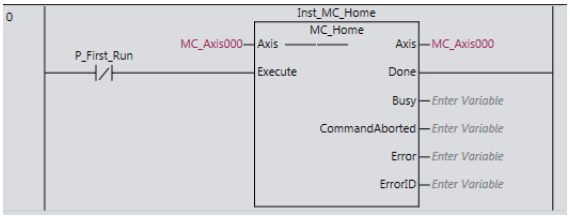
Internal Variables

Name	Data type	Initial value	Comment
Inst_MC_Home	MC_Home		

External Variables

Name	Data type	constant	Comment
MC_Axis000	_sAXIS_REF	✓	

Programming



Ladder Diagram(Relative Positioning)

Internal Variables

Name	Data type	Initial value	Comment
Start	BOOL		Start
StartPosition	LREAL		Start Position
Abort	BOOL		Abort
AbortMode	UINT		Stopping Mode Selection
Done	BOOL		Done
Position	LREAL		Position
CalcTravelTime	LREAL		Calculated Travel Time
Busy	BOOL		Executing
CommandAborted	BOOL		Instruction Aborted
Error	BOOL		Error
ErrorID	WORD		Error Code
ErrorIDEx	DWORD		Expansion Error Code
AbsolutePosition	LREAL		Absolute Position
Inst_MC_SyncMoveAbsolute	MC_SyncMoveAbsolute		
Direction	_eMC_DIRECTION		
BufferMode	_eMC_BUFFER_MODE		
SyncAbs_Busy	BOOL		
SyncAbs_Active	BOOL		
SyncAbs_CommandAborted	BOOL		
SyncAbs_Error	BOOL		
SyncAbs_ErrorIDEx	WORD		
Inst_VSConstVelProfile1	OmronLib\VS_Toolbox\VSConstVelProfile1		Instance of VSConstVelProfile1 FB
Velocity	LREAL		Velocity in Constant Velocity Zone
VSConstVelInput	OmronLib\VS_Toolbox\sVS_CONST_VEL_INPUT		VS Constant Velocity Profile Input
VSConstVelOutput	OmronLib\VS_Toolbox\sVS_CONST_VEL_OUTPUT		VS Constant Velocity Profile Output

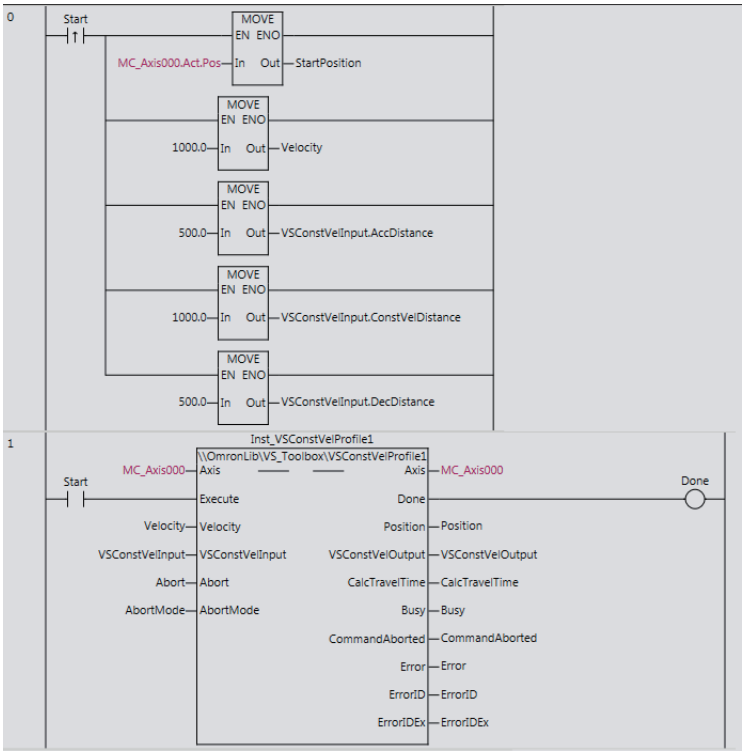
External Variables

Name	Data type	constant	Comment
MC_Axis000	_sAXIS_REF	✓	

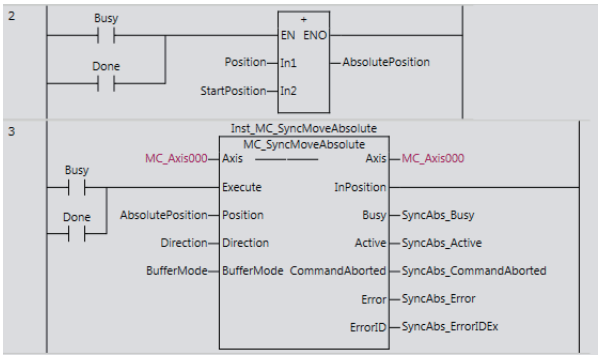
Programming

Since a relative travel is performed according to the command value by absolute coordinate, maintain the position at execution start.

Set the profile as follows.



Add the calculation result of this function block to the current position at execution start, and convert it to the absolute coordinate command position.



Ladder Diagram(Absolute Positioning)

● Internal Variables

Name	Data type	Initial value	Comment
Start	BOOL		Start
StartPosition	LREAL		Start Position
Abort	BOOL		Abort
AbortMode	UINT		Stopping Mode Selection
Done	BOOL		Done
Position	LREAL		Position
CalcTravelTime	LREAL		Calculated Travel Time
Busy	BOOL		Executing
CommandAborted	BOOL		Instruction Aborted
Error	BOOL		Error
ErrorID	WORD		Error Code
ErrorIDEx	DWORD		Expansion Error Code
AbsolutePosition	LREAL		Absolute Position
Inst_MC_SyncMoveAbsolute	MC_SyncMoveAbsolute		
Direction	_eMC_DIRECTION		
BufferMode	_eMC_BUFFER_MODE		
SyncAbs_Busy	BOOL		
SyncAbs_Active	BOOL		
SyncAbs_CommandAborted	BOOL		
SyncAbs_Error	BOOL		
SyncAbs_ErrorIDEx	WORD		
Inst_VSConstVelProfile1	OmronLib\VS_Toolbox\VSConstVelProfile1		Instance of VSConstVelProfile1 FB
Velocity	LREAL		Velocity in Constant Velocity Zone
VSConstVelInput	OmronLib\VS_Toolbox\sVS_CONST_VEL_INPUT		VS Constant Velocity Profile Input
VSConstVelOutput	OmronLib\VS_Toolbox\sVS_CONST_VEL_OUTPUT		VS Constant Velocity Profile Output
AbsAcc	LREAL		
AbsConst	LREAL		
AbsDec	LREAL		

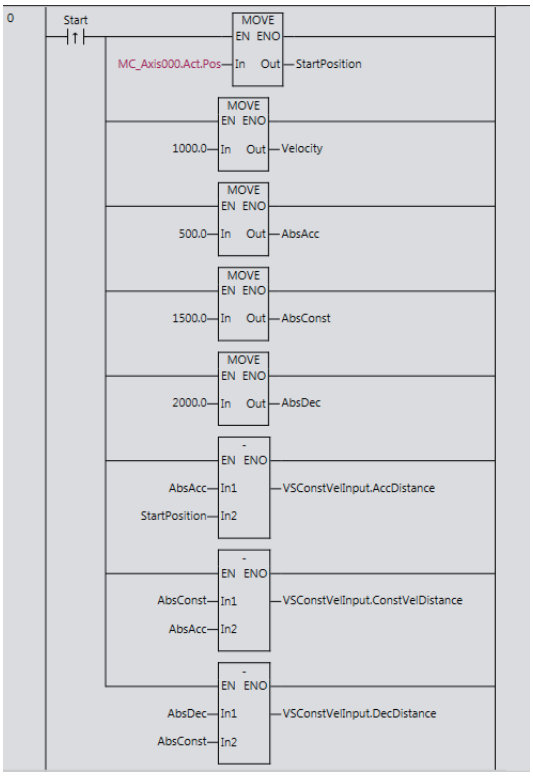
● External Variables

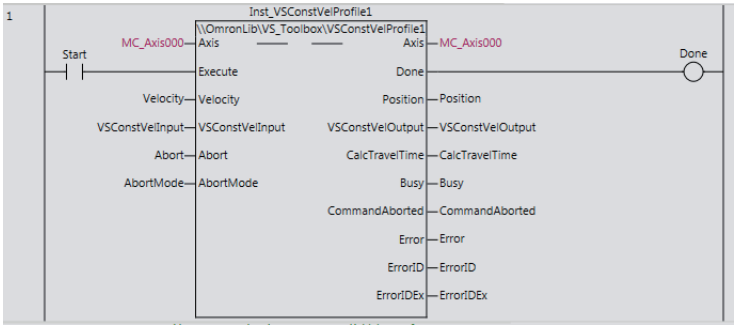
Name	Data type	constant	Comment
MC_Axis000	_sAXIS_REF	✓	

Programming

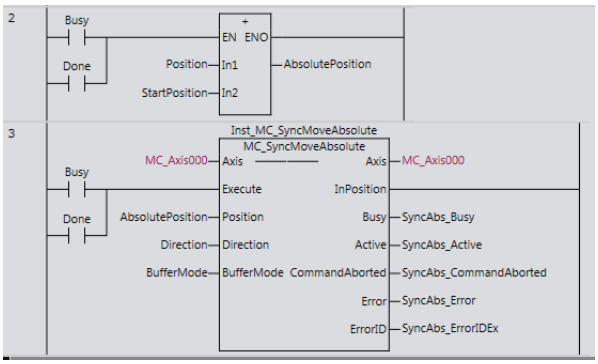
Convert a command value by absolute coordinate to the relative value.

Set the profile as follows.



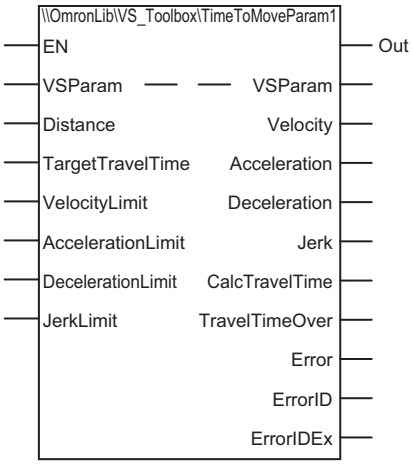


Add the calculation result of this function block to the current position at execution start, and convert it to the absolute coordinate command position.



TimeToMoveParam1

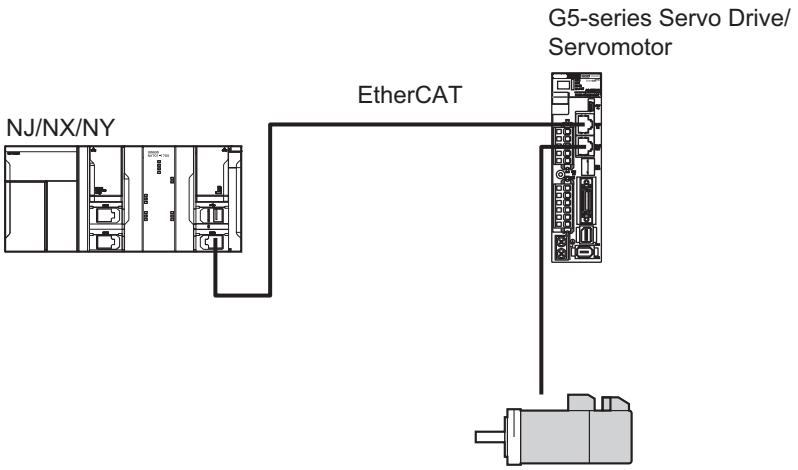
The function preforms positioning in a specified time.

Function name	Name	FB/ FUN	Graphic expression	ST expression
TimeToMovePa-ram1	Time Spe-cific Position-ing Parameter Calculation 1	FUN		Out:= TimeToMoveParam1(VSParam:=, Distance:=, TargetTravelTime:=, VelocityLimit:=, AccelerationLimit:=, DecelerationLimit:=, JerkLimit:=, Velocity=>, Acceleration=>, Deceleration=>, Jerk=>, CalcTravelTime=>, TravelTimeOver=>, Error=>, ErrorID=>, ErrorIDEx=>,) You can omit *Out*.

Function Block and Function Information

Item	Description
Library file name	OmronLib_VS_Toolbox_V1_0.slr
Namespace	OmronLib\VS_Toolbox
Function block and function number	00032
Source code published/not published	Not published
Function block and function version	1.00

Hardware Configuration Diagram



Variables

Input Variables

Name	Meaning	Data type	Valid range	Initial value	Description
Distance	Travel Distance	LREAL	Negative, 0, or Positive number	0	Specifies the travel distance from the command current position. The unit is [command unit].
TargetTravelTime	Target Travel Time	LREAL	Positive number from 0 to 100,000	0	Specified the travel time in which the command position should reach the target position. The travel time includes the damping control delay from MultiVSFilter1. The unit is [ms]. Any values below 0.001 ms are truncated.
VelocityLimit	Velocity Limit	LREAL	Positive number	0	Specifies the velocity limit. The unit is [command unit/s].
Acceleration Limit	Acceleration Limit	LREAL	Positive number	0	Specifies the acceleration limit. The unit is [command unit/s^2].
Deceleration Limit	Deceleration Limit	LREAL	Positive number	0	Specifies the deceleration limit. The unit is [command unit/s^2].
JerkLimit	Jerk Limit	LREAL	Positive number, or 0	0	Specifies the jerk limit. The unit is [command unit/s^3]. The jerk limit will be invalid when 0 is specified.

Additional Information

The command unit is indicated either in [mm], [μm], [nm], [degree], [inch], or [pulse].

For details, refer to *Unit Conversion Settings* in the motion control user's manual.

Output Variables

Name	Meaning	Data type	Valid range	Description
Out	Return Value	BOOL	TRUE only	Always TRUE
Velocity	Velocity	LREAL	Positive number, or 0	Gives the value set to the *Velocity* input variable to the MC_Move instruction.
Acceleration	Acceleration	LREAL	Positive number, or 0	Gives the value set to the *Acceleration* input variable to the MC_Move instruction.
Deceleration	Deceleration	LREAL	Positive number, or 0	Gives the value set to the *Deceleration* input variable to the MC_Move instruction.
Jerk	Jerk	LREAL	Positive number, or 0	Gives the value set to the *Jerk* input variable to the MC_Move instruction.
CalcTravelTime	Calculated Travel Time	LREAL	Positive number, or 0	Gives the calculated travel time. Any values below 0.001 ms are truncated.
TravelTimeOver	Travel Time Exceeded	BOOL	TRUE or FALSE	TRUE when the value of the *CalcTravelTime* output variable is greater than the value of the *TargetTravelTime* input variable.
Error	Error	BOOL	TRUE or FALSE	
ErrorID	Error Code*1	WORD	Depends on data types.	
ErrorIDEx	Expansion Error Code	DWORD	Depends on data types.	

*1. Refer to *Troubleshooting* on page 112 for error codes.

In-Out Variables

Name	Meaning	Data type	Valid range	Description
VSParam	Vibration Suppression Parameter	Omron-Lib\VS_Toolbox\sVSParam[0-4]	---	Specifies the vibration suppression parameters.

● Members of the sVSParam Structure

Member	Meaning	Data type	Valid range	Description
VSType	VS Type	UINT	0,1, 2, or 3	Selects the type of vibration suppression. 0: No vibration suppression 1: Type 1 2: Type 2 3: Type 3
Resonance Frequency	Resonance Frequency	LREAL	0.5 ≤ Resonance frequency ≤ 50	Specifies the resonance frequency of the vibration to be suppressed. The unit is [Hz].

Precautions for Correct Use

Set the same value to this function and to the *VSParam* in-out variable to MultiVSFilter1 function block.

Additional Information

For *VSType* and *ResonanceFrequency*, refer to *MultiVSFilter1* on page 48.

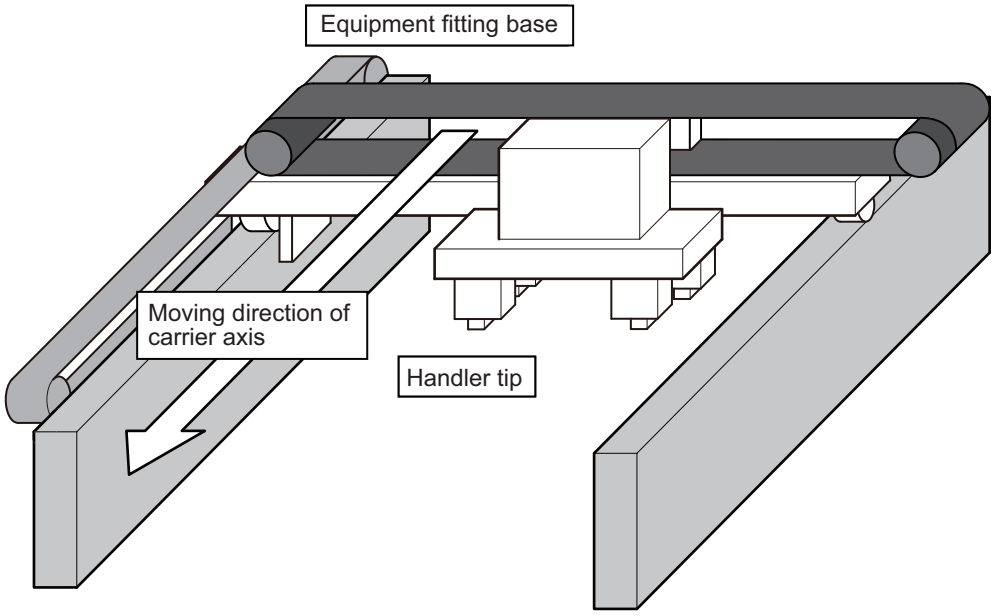
Function

When used together with MultiVSFilter1, the function can simultaneously suppress multiple vibrations that occur during axis motion, and it also calculates parameters needed for creating the command pattern that can reach the target position in the specified target travel time.

MultiVSFilter1 suppresses vibrations and can create a command position that has vibration suppression effect. Instead, the travel distance is extended. To avoid it, a velocity, acceleration, deceleration, and jerk that satisfy the target travel time is calculated from the resonance frequency, target travel time, velocity limit, acceleration limit, deceleration limit and jerk limit that are specified with this function. The calculated parameters and the single axis positioning instruction are used to create the command pattern. The pattern is input to MultiVSFilter1. Then it calculates the command position that can suppress vibrations.

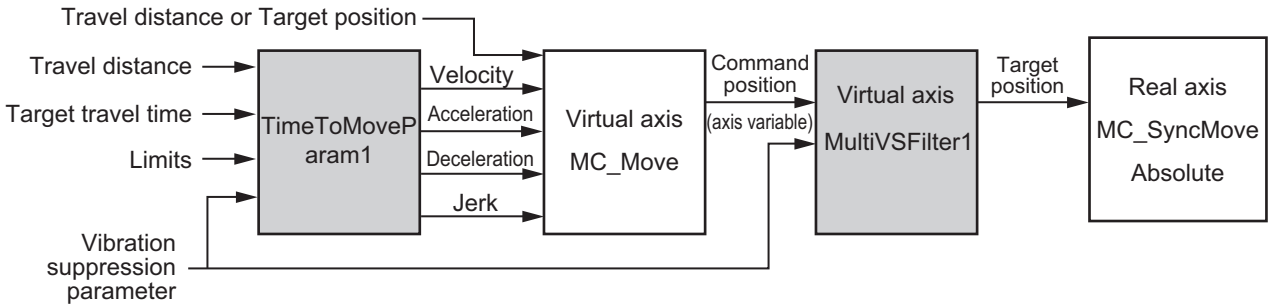
In the following equipment example, vibrations occur at the tip of the handler and on the equipment fitting base when the carrier axis moves.

<Example: Handling Equipment>



This function is used with a single axis positioning instruction such as MC_Move, MC_MoveAbsolute, and MC_MoveRelative motion control instructions, with a MultiVSFilter1 (Multiple Frequency VS Filter 1) function block, or with a MC_SyncMoveAbsolute (Cyclic Synchronous Absolute Positioning) motion control instruction. A virtual axis is assigned to the single axis positioning instruction. The function is executed only once before the single axis positioning for the virtual axis is expected. The output variable that contains the execution result of this function is linked to the input variable to the single axis positioning instruction for the virtual axis.

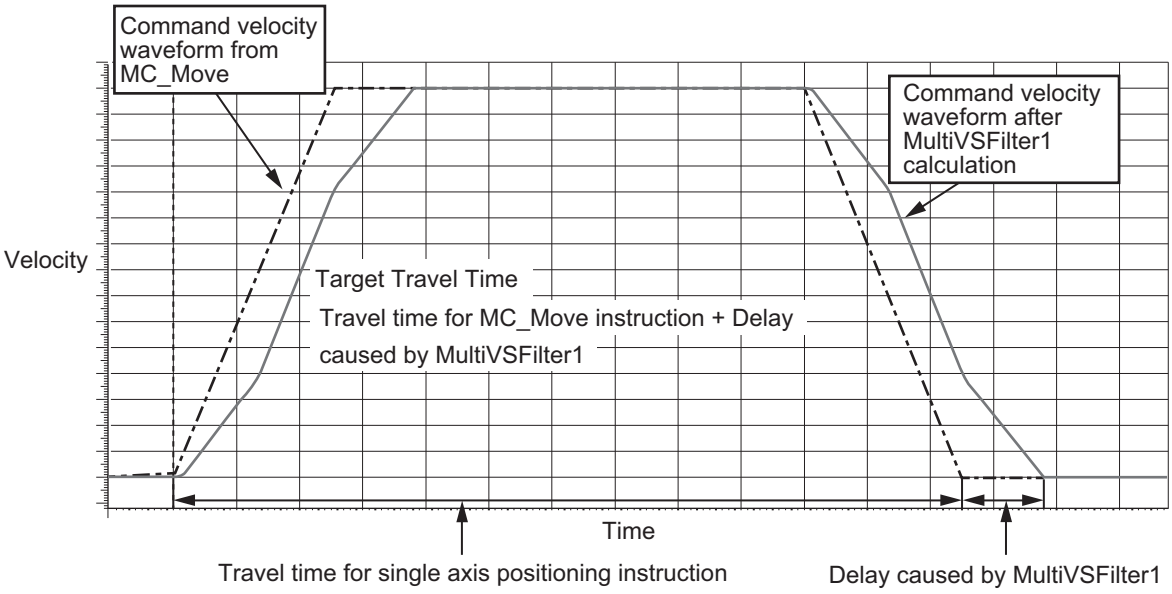
The function takes the travel time delay in MultiVSFilter1 into account to calculate the target velocity, acceleration, deceleration, and jerk for the MC_Move instruction, so that the real axis operates in the specified target time.



Additional Information

The MultiVSFilter1 function block suppresses vibrations of up to five different resonance frequencies. Refer to *MultiVSFilter1* on page 48 for details on the MultiVSFilter1 function block.

A target travel time means a duration from when the single axis positioning starts to when the position given by MultiVSFilter1 reaches the target position. As shown by the waveform of the command velocity (command position variance) given from the MC_Move instruction and the waveform of the command velocity given from the calculation by MultiVSFilter1 in the figure below, the travel time is extended for the delay caused by MultiVSFilter1. Therefore, the target travel time is produced by adding the travel time from the MC_Move instruction and the delay caused by MultiVSFilter1.



This function calculates the solution combined by velocity, acceleration, deceleration, and jerk, which satisfies the target travel time, and outputs the result. Depending on the calculation results, the following processing is performed.

1 If the shortest operable travel time is longer than the target travel time, a waring is output to indicate that the target travel time is exceeded.

2 Even if the shortest travel time is shorter than the target travel time, the processing ends with an error if there is no solution combined by velocity, acceleration, deceleration, and jerk, which satisfies the target travel time.

3 In some cases, there is a solution combined by velocity, acceleration, deceleration, and jerk, which satisfies the target travel time, but correct parameters cannot be calculated due to errors in numerical processing. The processing ends with an error if the travel time calculation error caused by the incorrect parameter is greater than 10% of the delay caused by MultiVSFilter1.If the delay is 0, and the measurement error is greater than 10% of the travel time, the processing ends with an error.

For details, refer to *Calculating the Parameters to MC_Move Instruction* on page 107.

Processing List

Processing			Description
Checking the input parameters	Limits	Velocity Limit	Checks to see if the value of *VelocityLImit* input variable is within the valid range. An Input Parameter Out of Range status is given if the value is out of the valid range.
		Acceleration Limit	Checks to see if the value of *AccelerationLImit* input variable is within the valid range. An Input Parameter Out of Range status is given if the value is out of the valid range.
		Deceleration Limit	Checks to see if the value of *DecelerationLImit* input variable is within the valid range. An Input Parameter Out of Range status is given if the value is out of the valid range.
		Jerk Limit	Checks to see if the value of *JerkLImit* input variable is within the valid range. An Input Parameter Out of Range status is given if the value is out of the valid range.
	Vibration Suppression Parameter*1	VS Type, Resonance Frequency	Checks to see if the values of *VSParam[i].VSType* and *VSParam[i].ResonanceFrequency* in-out variables are within the valid ranges. An Input Parameter Out of Range status is given if the value is out of the valid range.
	Travel Distance		Checks to see if the value of *Distance* input variable is within the valid range. An Input Parameter Out of Range status is given if the value is out of the valid range.
Calculating the delay	---	---	Calculates the delay caused by vibration suppression. For details, refer to *Calculating the Delay* on page 107.
Calculating the MC_Move instruction parameters	---	---	Calculates target velocity, acceleration, deceleration, and jerk that are specified to input parameters to the MC_Move instruction. For details, refer to *Calculating the Parameters to MC_Move Instruction* on page 107.
Calculating the travel time	---	---	Calculates the travel time. For details, refer to *Calculating the Travel Time* on page 111.

*1. Only when the value of the*VSParam[i].VSType* in-out variable is other than 0, perform the check to see if the values of vibration suppression parameters are within valid ranges.

Calculating the Delay

The function calculates the delay from the resonance frequency value specified to the input variable.

First, the function uses the value of *VSParam[n].VSType*, and calculates the delay $_n$ [ms] caused by vibration suppression of n-th (n=0 to 4) resonance frequency.

VSType	Vibration suppression type	Delay$_n$ [ms]*1
0	No vibration suppression	0
1	Type 1	$\frac{1{,}000}{2 \bullet F}$[ms]
2	Type 2	$\frac{2{,}000}{2 \bullet F}$[ms]
3	Type 3	$\frac{3{,}000}{2 \bullet F}$[ms]

*1. The character “F” in the table indicates the *ResonanceFrequency* input variable.

The function adds delays up to the calculated Delay $_n$ [ms].

Delay [ms] = Delay$_0$ + Delay$_1$ + … + Delay$_4$

Calculating the Parameters to MC_Move Instruction

In the following procedure, velocity, acceleration, deceleration, and jerk are calculated.

1 The value of *Distance* (Travel Distance) is checked if it is 0. If the value is 0, the following values are output to respective variables. CalcTravelTime=0, Velocity= VelocityLimit, Acceleration=AccelerationLimit, Deceleration=DecelerationLimit, Jerk=JerkLimit.
Then the processing makes a normal end.
If the value is not 0, the step 2 is performed.

2 The travel time for the MC_Move instruction (the time taken from when the instruction is executed to when the target position is reached).

Travel Time [ms] for the MC_Move instruction

= TargetTravelTime input variable - Delay caused by MultiVSFilter1

3 The MC_Move instruction parameters are calculated.

(1) The shortest operable travel time is calculated from the entered parameters. If *TargetTravelTime* is equals to or smaller than 0, or the shortest travel time is longer than the target travel time, *TravelTimeOver* changes to TRUE, the values of *Velocity, Acceleration, Deceleration,* and *Jerk* then are given, and the processing ends.

(2) If the shortest travel time is shorter than the target travel time, an analysis is made to check if there is a solution combined by velocity, acceleration, deceleration, and jerk, which satisfies the target travel time. If the analysis finds no such solution, the values of *Velocity, Acceleration, Deceleration,* and *Jerk* then are given, and the processing ends with an error.

(3) If there is such solution, velocity, acceleration, deceleration, and jerk that satisfy the target travel time are calculated. Due to numerical processing errors, the travel time to the MC_Move instruction, which is processed by the calculated velocity, acceleration, decelera-

tion, and jerk, may vary from the expected value found in step 2. If the error is greater than 10% of the delay caused by MultiVSFilter1, the previously-executed value is given from *Velocity, Acceleration, Deceleration,* and *Jerk*, and the processing ends with an error. If the delay is 0, and the error is greater than 10% of the travel time, the previously-executed value is given from *Velocity*, *Acceleration*, *Deceleration*, and *Jerk*, and the processing ends with an error.

(4) If the error is equal to or smaller than 10% of the delay caused by MultiVSFilter1, the value then are given from *Velocity, Acceleration, Deceleration,* and *Jerk*, and the processing makes a normal end. Also, if the delay is 0 and the error is equal to or smaller than 10% of the travel time, the value then are given from *Velocity, Acceleration, Deceleration,* and *Jerk*, and the processing makes a normal end.

The following section describes the corrections when the parameters to MC_Move cannot be calculated correctly. In some cases, you cannot solve a problem by one correction. In such cases, perform more than one correction together.

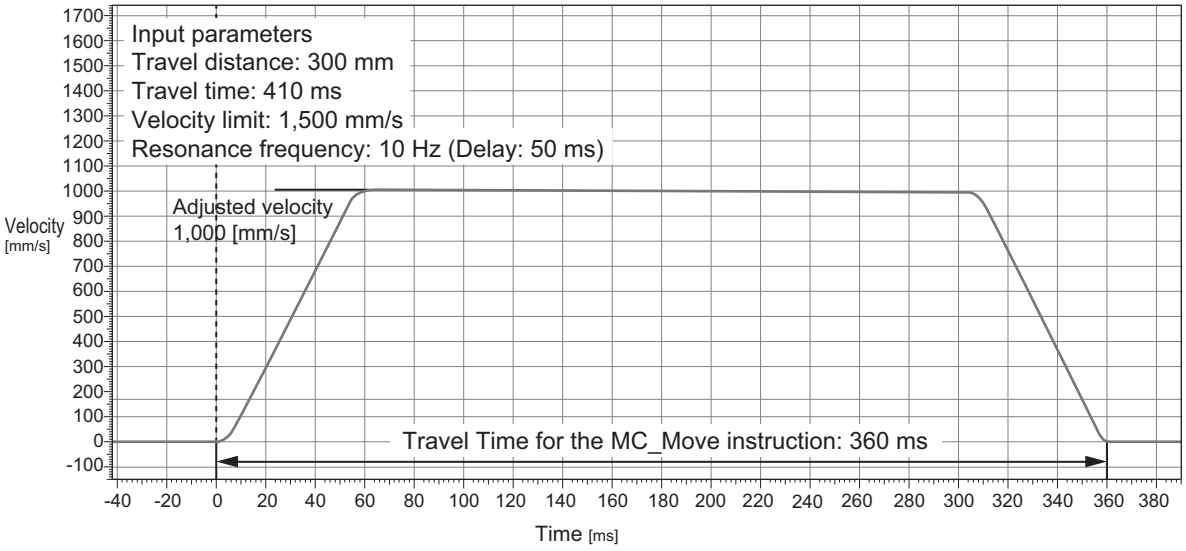
Description	Status	Correction
If the shortest travel time is longer than the target travel time (see Step 3-(1))	The TravelTimeOver output variable changes to TRUE.	• Expand the acceleration and deceleration limits. • Expand the jerk limit. • Change the value so the result of acceleration limit divided by jerk limit (or the result of deceleration limit divided by jerk limit) becomes a smaller number. • Extend the target travel time.
If there is no solution combined by velocity, acceleration, deceleration, and jerk, which satisfies the target travel time (see Step 3-(2))	ErrorID=WORD#16#3C19 ErrorID=DWORD#16#00000006 are output.	• Lower the acceleration and deceleration limits. • Shorten the target travel time.
If the travel time calculation error is greater than 10% of the delay caused by MultiVSFilter1 (see Step 3-(3))		

The following section describes examples in which the function calculates the parameters to MC_Move.

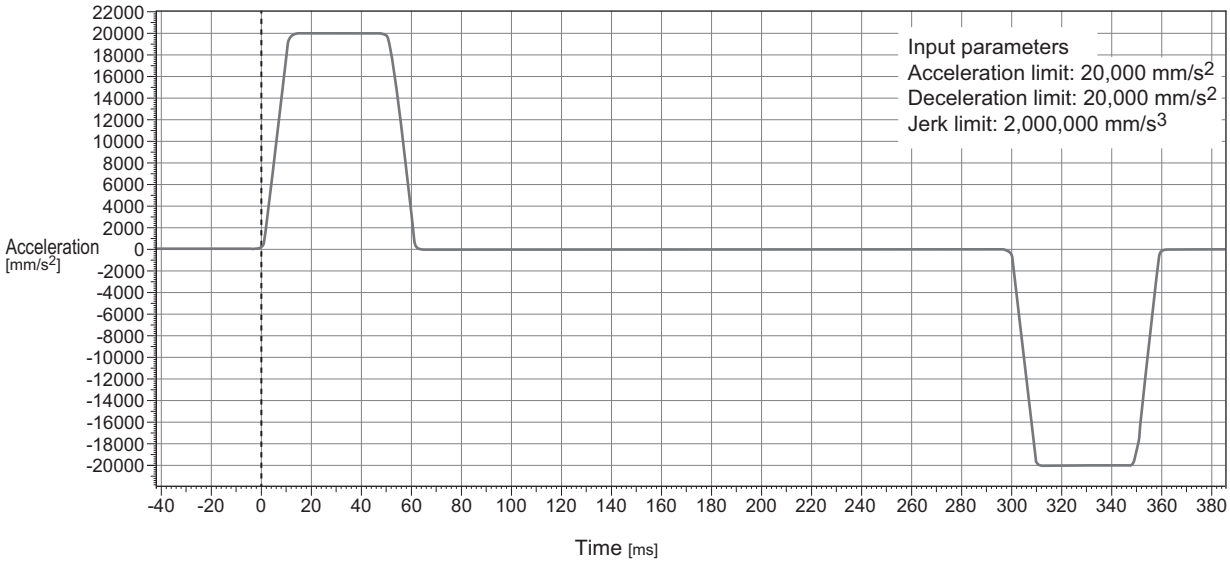
● Example 1: Adjusting the Velocity

The figures below are the velocity and acceleration waveforms from the MC_Move instruction. Only velocity among the parameters input to the instruction was adjusted according to the calculation processing in Step 3-(2).

<Example of Velocity Waveform from MC_Move Instruction>



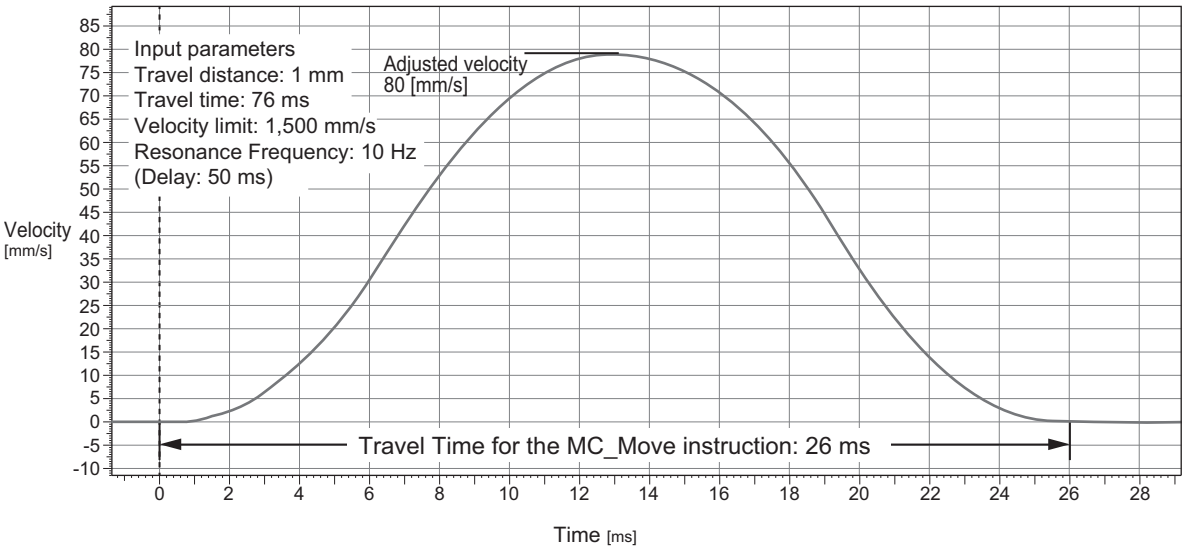
<Example of Acceleration Waveform from MC_Move Instruction>



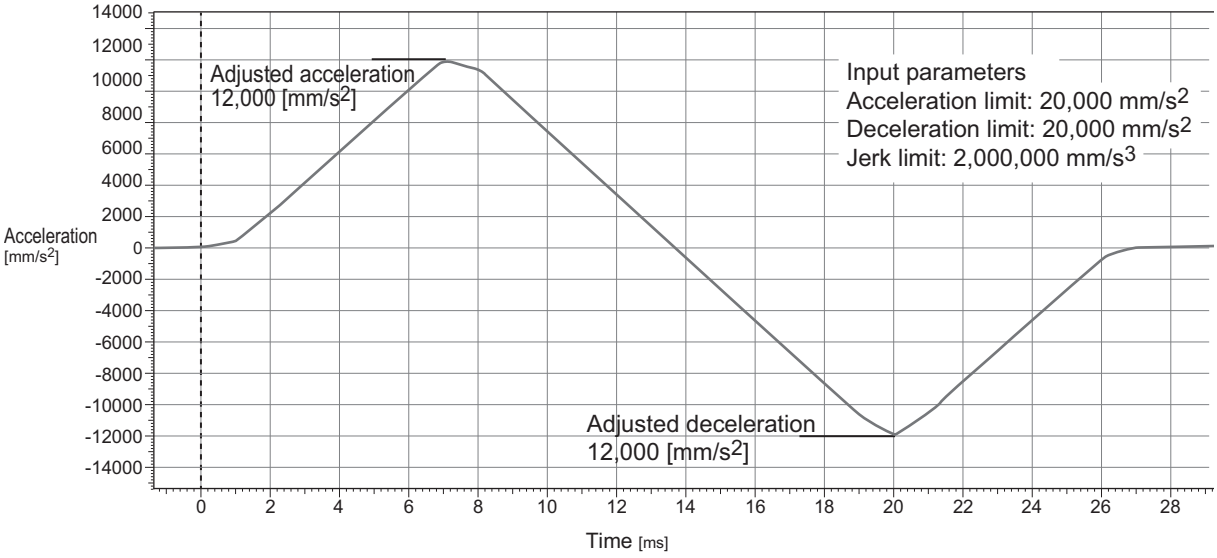
● Example 2: Adjusting the Velocity, Acceleration, and Deceleration

The figures below are the velocity, acceleration, and deceleration waveforms that were adjusted according to the calculation processing in Step 3-(2). While the travel distance is short, the target travel distance may be exceeded by an increased acceleration or deceleration before the acceleration/deceleration limit. In such a case, lower the acceleration or deceleration below the limits to ensure target travel distance and time.

<Example of Velocity Waveform from MC_Move Instruction>



<Example of Acceleration Waveform from MC_Move Instruction>



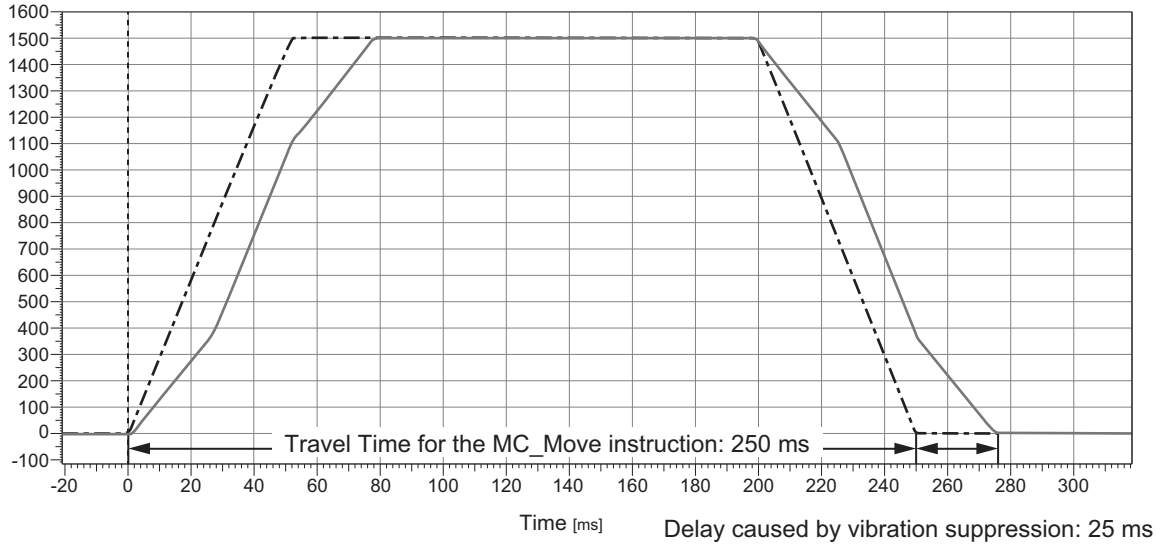
Calculating the Travel Time

The function calculates the travel time that includes the delay cased by vibration suppression.
Travel Time [ms] = Travel Time for MC_Move instruction + Delay

For details on delay, refer to *Calculating the Delay* on page 107.
If the processing ends with an error, the travel time = 0.

The example below shows the case where the travel time reaches 275 ms.

<Example of Velocity Waveform>



Troubleshooting

Error Code	Expansion Error Code	Status	Cause	Correction
16#0000	16#00000000	Normal End	---	---
16#3C19	16#00000001	Target Travel Time Setting Out of Range	A value set to the *TargetTravelTime* input parameter is outside the valid range.	Check the valid range of the *TargetTravelTime* input parameter, and set a value within the valid range.
	16#00000002	Velocity Limit Setting Out of Range	A value set to the *VelocityLimit* input parameter is outside the valid range.	Check the valid range of the *VelocityLimit* input parameter, and set a value within the valid range.
	16#00000003	Acceleration Limit Setting Out of Range	A value set to the *AccelerationLimit* input parameter is outside the valid range.	Check the valid range of the *AccelerationLimit* input parameter, and set a value within the valid range.
	16#00000004	Deceleration Limit Setting Out of Range	A value set to the *DecelerationLimit* input parameter is outside the valid range.	Check the valid range of the *DecelerationLimit* input parameter, and set a value within the valid range.
	16#00000005	Jerk Limit Setting Out of Range	A value set to the *JerkLimit* input parameter is outside the valid range.	Check the valid range of the *JerkLimit* input parameter, and set a value within the valid range.
	16#00000006	MC_Move Parameter Calculation Failed	The MC_Move instruction parameters cannot be calculated with the given input parameters.	For details, refer to *Calculating the Parameters to MC_Move Instruction* on page 107.
	16#00000007	VS Type Selection Out of Range	A value set to the *VSType* member of *VSParam* is outside the valid range.	Check the valid range of the *VSType* member of *VSParam*, and set a value within the valid range.
	16#00000008	Resonance Frequency Setting Out of Range	A value set to the *ResonanceFrequency* member of *VSParam* is outside the valid range.	Check the valid range of the *ResonanceFrequency* member of *VSParam*, and set a value within the valid range.
	16#00000009	Travel Distance Setting Out of Range	A value set to the *Distance* input parameter is outside the valid range.	Check the valid range of the *Distance* input parameter, and set a value within the valid range.

Sample Programming

Program Operation

The sample program uses the function block to operate a servo axis in absolute positioning. The table below shows the processing order.

No.	Outline	Description
1.	Calculating the time specific positioning parameters	The program uses TimeToMoveParam1 to calculate values to be set to Velocity, Acceleration, Deceleration, and Jerk input parameters to MC_Move, from travel distance, travel time, velocity limit, acceleration limit, deceleration limit, jerk limit, and VSParam vibration suppression parameters used for MultiVSFilter1.
2.	Absolute positioning	The program executes the absolute positioning for the virtual servo axis.
3.	Starting or stopping synchronization	• Synchronization Start The program changes the command current position of the virtual servo axis to the command current position of the servo axis, and starts synchronization for the servo axis. • Stopping synchronization The program stops synchronization for the servo axis when an error occurs or when an vibration suppression parameter is changed.
4.	Vibration suppression filtering	The program executes vibration suppression filtering for the command current position of the virtual servo axis.
5.	Cyclic synchronous absolute positioning	The servo axis is synchronized to the output value of the vibration suppression filter.
6.	Canceling synchronization	The synchronization to the servo axis is canceled (it decelerates to stop).

Preconditions

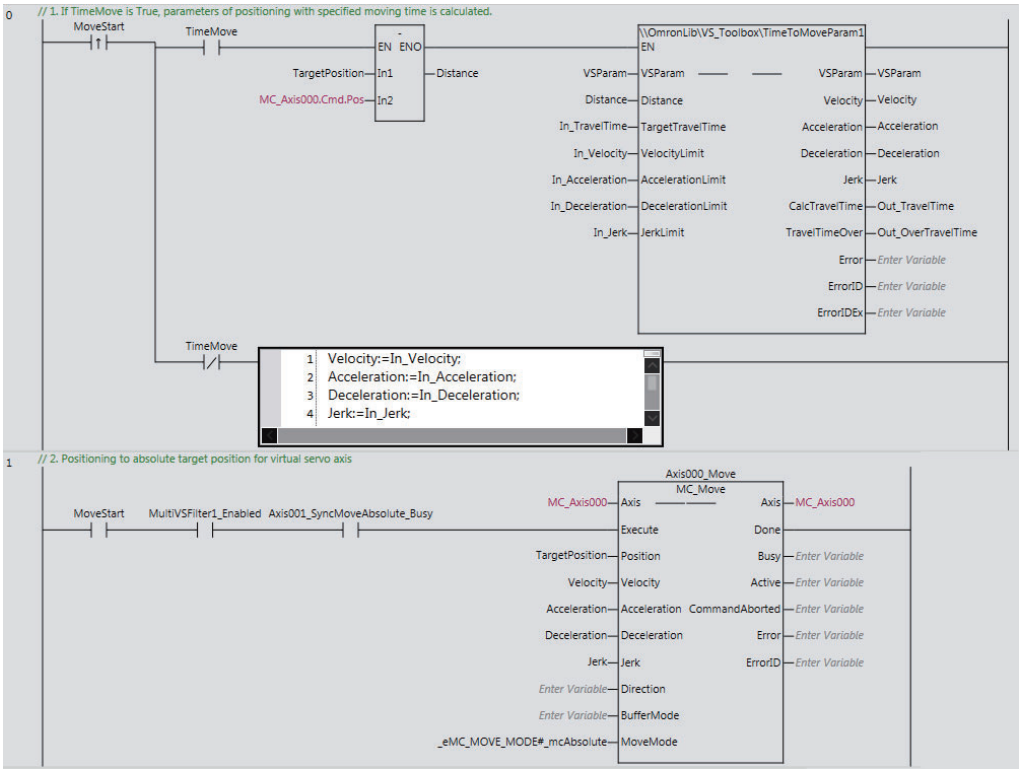
- The virtual servo axis settings are the same as the servo axis settings (except for homing).
- The virtual servo axis and servo axis are ready to operate.
- The home for the virtual servo axis and servo axis are defined.

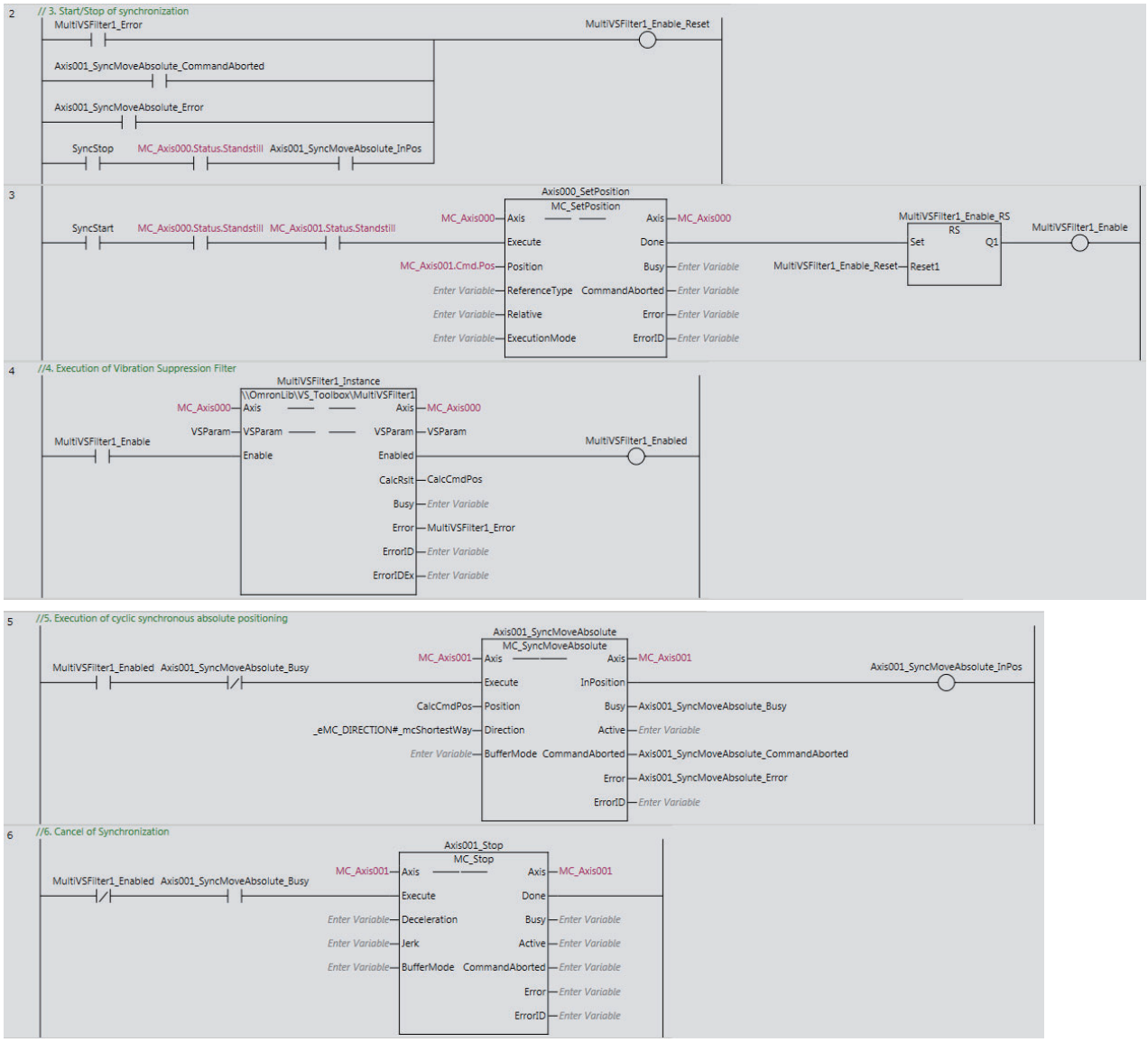
Ladder Diagram

● Main Variables

Name	Data type	Initial value	Comment
TimeMove	BOOL	FALSE	When the flag changes to TRUE, the time specific positioning is enabled.
MoveStart	BOOL	FALSE	When the flag changes to TRUE, positioning the virtual servo axis is started.
SyncStart	BOOL	FALSE	When the flag changes to TRUE, synchronization for the servo axis is started.
SyncStop	BOOL	FALSE	When the flag changes to TRUE, synchronization for the servo axis is stopped.
VSParam	ARRAY[0..4] OF OmronLib\VS_Toolbox\sVSParam	---	Specifies the vibration suppression parameters. Use the same value for the TimeToMoveParam1 function and the MultiVSFilter function block.
TargetPosition	LREAL	0	Specifies the target position. The unit is [command unit].
In_Velocity	LREAL	0	Specifies the velocity limit. The unit is [command unit/s].
In_Acceleration	LREAL	0	Specifies the acceleration limit. The unit is [command unit/s^2].
In_Deceleration	LREAL	0	Specifies the deceleration limit. The unit is [command unit/s^2].
In_Jerk	LREAL	0	Specifies the jerk limit. The unit is [command unit/s^3].

● Programming




2
// 3. Start/Stop of synchronization
MultiVSFilter1_Error
MultiVSFilter1_Enable_Reset
Axis001_SyncMoveAbsolute_CommandAborted
Axis001_SyncMoveAbsolute_Error
SyncStop
MC_Axis000.Status.Standstill
Axis001_SyncMoveAbsolute_InPos
3
Axis000_SetPosition
MC_SetPosition
SyncStart
MC_Axis000.Status.Standstill
MC_Axis001.Status.Standstill
MC_Axis000
Axis
Execute
Position
ReferenceType
Relative
ExecutionMode
Done
Busy
CommandAborted
Error
ErrorID
MC_Axis001.Cmd.Pos
Enter Variable
MultiVSFilter1_Enable_RS
RS
Set
Reset1
Q1
MultiVSFilter1_Enable_Reset
MultiVSFilter1_Enable
4
//4. Execution of Vibration Suppression Filter
MultiVSFilter1_Instance
\\OmronLib\VS_Toolbox\MultiVSFilter1
MC_Axis000
Axis
VSParam
Enable
Enabled
CalcRsit
CalcCmdPos
Busy
Error
ErrorID
ErrorIDEx
MultiVSFilter1_Enable
MultiVSFilter1_Enabled
MultiVSFilter1_Error

5
//5. Execution of cyclic synchronous absolute positioning
Axis001_SyncMoveAbsolute
MC_SyncMoveAbsolute
MC_Axis001
Axis
Execute
Position
Direction
BufferMode
InPosition
Busy
Active
CommandAborted
Error
ErrorID
MultiVSFilter1_Enabled
Axis001_SyncMoveAbsolute_Busy
CalcCmdPos
_eMC_DIRECTION#_mcShortestWay
Enter Variable
Axis001_SyncMoveAbsolute_InPos
Axis001_SyncMoveAbsolute_CommandAborted
Axis001_SyncMoveAbsolute_Error
6
//6. Cancel of Synchronization
Axis001_Stop
MC_Stop
Deceleration
Jerk
Done

Structured Text (ST)

● Main Variables

Name	Data type	Initial value	Comment
TimeMove	BOOL	FALSE	When the flag changes to TRUE, the time specific positioning is enabled.
MoveStart	BOOL	FALSE	The flag is changed to TRUE before positioning the virtual servo axis is started.
WriteParameter	BOOL	FALSE	The variable changes to TRUE by the upward differentiation of *MoveStart*, and the parameters to the MC_Move are overwritten.
SyncStart	BOOL	FALSE	When the flag changes to TRUE, synchronization for the servo axis is started.
SyncStop	BOOL	FALSE	When the flag changes to TRUE, synchronization for the servo axis is stopped.
VSParam	ARRAY[0..4] OF OmronLib\VS_Toolbox\sVSParam	---	Specifies the vibration suppression parameters. The MultiVSFilter1 function block suppresses vibrations of up to five different resonance frequencies (of 0.5 to 50 Hz).
TargetPosition	LREAL	0	Specifies the target position. The unit is [command unit].
In_Velocity	LREAL	0	Specifies the velocity limit. The unit is [command unit/s].
In_Acceleration	LREAL	0	Specifies the acceleration limit. The unit is [command unit/s^2].
In_Deceleration	LREAL	0	Specifies the deceleration limit. The unit is [command unit/s^2].
In_Jerk	LREAL	0	Specifies the jerk limit. The unit is [command unit/s^3].

● Programming

```
// 1. If TimeMove is True, parameters of positioning with specified moving time is
calculated.

R_TRIG_Instance
(Clk:=MoveStart,
Q=>WriteParameter);

IF WriteParameter THEN
   IF TimeMove THEN
      Distance:=TargetPosition-MC_Axis000.Cmd.Pos;
      \\OmronLib\VS_Toolbox\TimeToMoveParam1
      (VSParam:=VSParam,
      Distance:=Distance,
      TargetTravelTime:=In_TravelTime,
      VelocityLimit:=In_Velocity,
      AccelerationLimit:=In_Acceleration,
      DecelerationLimit:=In_Deceleration,
      JerkLimit:=In_Jerk,
      Velocity=>Velocity,
      Acceleration=>Acceleration,
      Deceleration=>Deceleration,
      Jerk=>Jerk,
      CalcTravelTime=>Out_TravelTime,
      TravelTimeOver=>Out_TravelTimeOver);
   ELSE
      Velocity:=In_Velocity;
      Acceleration:=In_Acceleration;
      Deceleration:=In_Deceleration;
      Jerk:=In_Jerk;
   END_IF;
END_IF;
```

```
// 2. Positioning to relative position from current position for virtual servo axis
IF MoveStart AND MultiVSFilter1_Enabled AND Axis001_SyncMoveAbsolute_Busy THEN
   Axis000_Move_Execute:=TRUE;
ELSE
   Axis000_Move_Execute:=FALSE;
END_IF;

// 3. Start/Stop of synchronization
IF MultiVSFilter1_Error
   OR Axis001_SyncMoveAbsolute_CommandAborted
   OR Axis001_SyncMoveAbsolute_Error
   OR (SyncStop AND MC_Axis000.Status.Standstill AND Axis001_SyncMoveAbsolute_InPos)
THEN
   MultiVSFilter1_Enable_Reset:=TRUE;
END_IF;
IF SyncStart AND MC_Axis000.Status.Standstill AND MC_Axis001.Status.Standstill THEN
   Axis000_SetPosition_Execute:=TRUE;
ELSE
   Axis000_SetPosition_Execute:=FALSE;
END_IF;


//4. Execution of Vibration Suppression Filter
IF MultiVSFilter1_Ready THEN
   MultiVSFilter1_Enable:=TRUE;
ELSE
   MultiVSFilter1_Enable:=FALSE;
END_IF;

//5. Execution of cyclic synchronous absolute positioning
IF MultiVSFilter1_Enabled AND NOT(Axis001_SyncMoveAbsolute_Busy) THEN
   Axis001_SyncMoveAbsolute_Execute:=TRUE;
ELSE
   Axis001_SyncMoveAbsolute_Execute:=FALSE;
END_IF;

//6. Cancel of Synchronization
IF NOT(MultiVSFilter1_Enabled) AND Axis001_SyncMoveAbsolute_Busy THEN
   Axis001_Stop_Execute:=TRUE;
ELSE
   Axis001_Stop_Execute:=FALSE;
END_IF;



//MC_Move
Axis000_Move
(Axis:=MC_Axis000,
Execute:=Axis000_Move_Execute,
Position:=TargetPosition,
Velocity:=Velocity,
Acceleration:=Acceleration,
Deceleration:=Deceleration,
Jerk:=Jerk,
MoveMode:=_eMC_MOVE_MODE#_mcAbsolute);

// MC_SetPosition
Axis000_SetPosition
(Axis:=MC_Axis000,
Execute:=Axis000_SetPosition_Execute,
Position:=MC_Axis001.Cmd.Pos,
Done=>Axis000_SetPosition_Done);

// RS
MultiVSFilter1_Enable_RS
```

```
(Set:=Axis000_SetPosition_Done,
Reset1:=MultiVSFilter1_Enable_Reset,
Q1=>MultiVSFilter1_Ready);

// MultiVSFilter1
MultiVSFilter1_instance
(Axis:=MC_Axis000,
VSParam:=VSParam,
Enable:=MultiVSFilter1_Enable,
Enabled=>MultiVSFilter1_Enabled,
CalcRslt=>CalcCmdPos,
Error=>MultiVSFilter1_Error
);


//MC_SyncMoveAbsolute
Axis001_SyncMoveAbsolute
(Axis:=MC_Axis001,
Execute:=Axis001_SyncMoveAbsolute_Execute,
Position:=CalcCmdPos,
Direction:=_eMC_DIRECTION#_mcShortestWay,
InPosition=>Axis001_SyncMoveAbsolute_InPos,
Busy=>Axis001_SyncMoveAbsolute_Busy,
CommandAborted=>Axis001_SyncMoveAbsolute_CommandAborted,
Error=>Axis001_SyncMoveAbsolute_Error
);

//MC_Stop
Axis001_Stop
(Axis:=MC_Axis001,
Execute:=Axis001_Stop_Execute);
```

Additional Information

Refer to *MultiVSFilter1* on page 48 for the sample programming that uses this function and operates the servo axis set to Rotary Mode in the relative positioning.

Appendix

Referring to Library Information

When you make an inquiry to OMRON about the library, you can refer to the library information to identify the library to ask about.

The library information is useful in identifying the target library among the libraries provided by OMRON or created by the user.

The library information consists of the attributes of the library and the attributes of function blocks and functions contained in the library.

- Attributes of libraries

  Information for identifying the library itself
- Attributes of function blocks and functions

  Information for identifying the function block and function contained in the library

Use the Sysmac Studio to access the library information.

Attributes of Libraries, Function Blocks and Functions

The following attributes of libraries, function blocks and functions are provided as the library information.

● Attributes of Libraries

No.*1	Attribute	Description
(1)	Library file name	The name of the library file
(2)	Library version	The version of the library
(3)	Author	The name of creator of the library
(4)	Comment	The description of the library*2

*1. These numbers correspond to the numbers shown on the screen images in the next section, *Referring to Attributes of Libraries, Function Blocks and Functions* on page 121.

*2. It is provided in English and Japanese.

● Attributes of Function Blocks and Functions

No.*1	Attribute	Description
(5)	FB/FUN name	The name of the function block or function
(6)	Name space	The name of name space for the function block or function
(7)	FB/FUN version	The version of the function block or function
(8)	Author	The name of creator of the function block or function
(9)	FB/FUN number	The function block number or function number
(10)	Comment	The description of the function block or function*2

*1. These numbers correspond to the numbers shown on the screen images in the next section, *Referring to Attributes of Libraries, Function Blocks and Functions* on page 121.

*2. It is provided in English and Japanese.

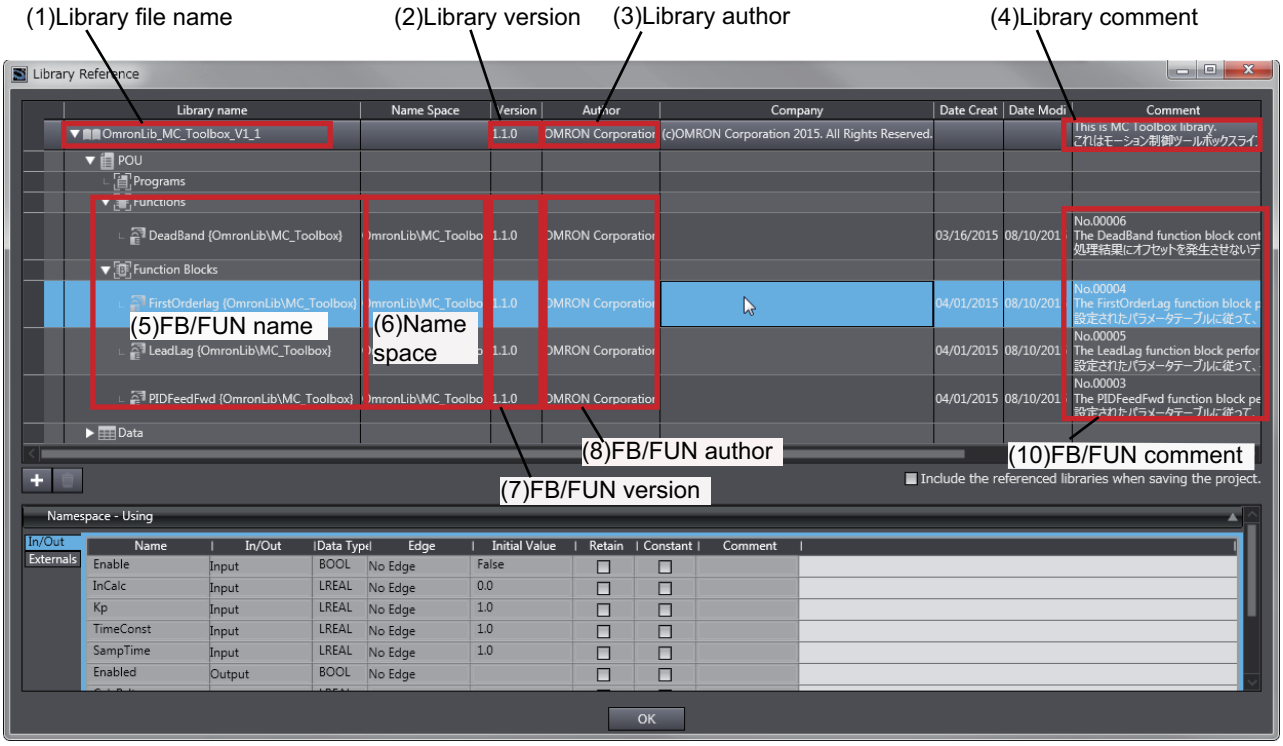
Referring to Attributes of Libraries, Function Blocks and Functions

You can refer to the attributes of libraries, function blocks and functions of the library information at the following locations on the Sysmac Studio.

- Library Reference Dialog Box
- Toolbox Pane
- Ladder Editor

(a) Library Reference Dialog Box

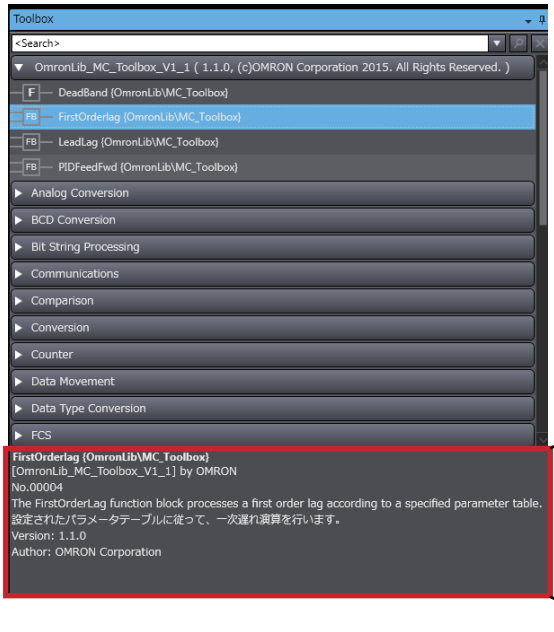
When you refer to the libraries, the library information is displayed at the locations shown below.



(b) Toolbox Pane

Select a function block and function to display its library information at the bottom of the Toolbox Pane.

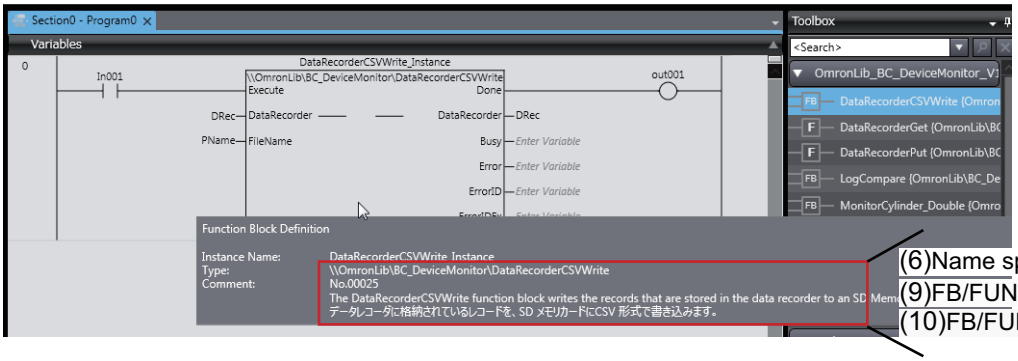
The text “by OMRON” which is shown on the right of the library name (1) indicates that this library was provided by OMRON.



(5)FB/FUN name (6)Name space
(1)Library file name
(9)FB/FUN number
(10)FB/FUN comment
(7)FB/FUN version
(8)FB/FUN author

(c) Ladder Editor

Place the mouse on a function block and function to display the library information in a tooltip.



(6)Name space (5)FB/FUN name
(9)FB/FUN number
(10)FB/FUN comment

Referring to Function Block and Function Source Codes

You can refer to the source codes of function blocks and functions provided by OMRON to customize them to suit the user's environment.

User function blocks and user functions can be created based on the copies of these source codes.

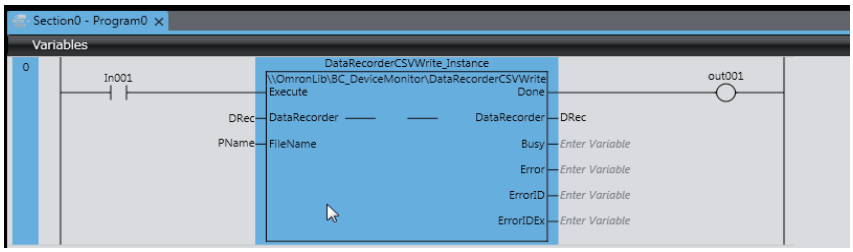
The following are the examples of items that you may need to customize.

- Customizing the size of arrays to suit the memory capacity of the user's Controller
- Customizing the data types to suit the user-defined data types

Note that you can access only function blocks and functions whose Source code published/not published is set to Published in the library information shown in their individual specifications.

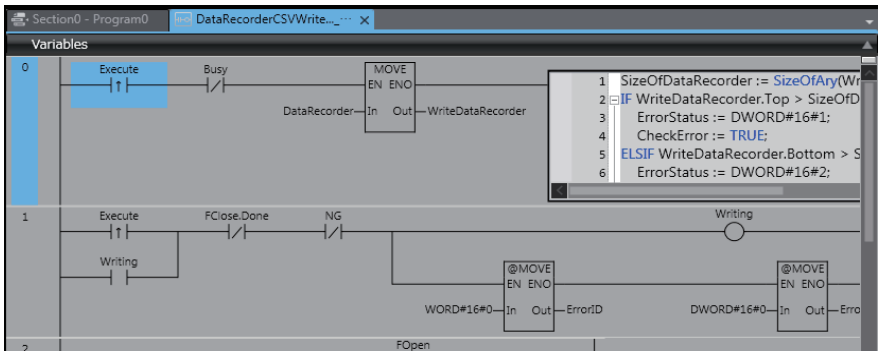
Use the following procedure to refer to the source codes of function blocks and functions.

1 Select a function block or function in the program.



2 Double-click or right-click and select **To Lower Layer** from the menu.

The source code is displayed.



Precautions for Correct Use

For function blocks and functions whose source codes are not published, the following dialog box is displayed in the above step 2. Click the **Cancel** button.

OMRON Corporation **Industrial Automation Company**
Kyoto, JAPAN
Contact: www.ia.omron.com

Regional Headquarters
OMRON EUROPE B.V.
Wegalaan 67-69, 2132 JD Hoofddorp
The Netherlands
Tel: (31)2356-81-300/Fax: (31)2356-81-388

OMRON ELECTRONICS LLC
2895 Greenspoint Parkway, Suite 200
Hoffman Estates, IL 60169 U.S.A.
Tel: (1) 847-843-7900/Fax: (1) 847-843-7787

OMRON ASIA PACIFIC PTE. LTD.
No. 438A Alexandra Road # 05-05/08 (Lobby 2),
Alexandra Technopark,
Singapore 119967
Tel: (65) 6835-3011/Fax: (65) 6835-2711

OMRON (CHINA) CO., LTD.
Room 2211, Bank of China Tower,
200 Yin Cheng Zhong Road,
PuDong New Area, Shanghai, 200120, China
Tel: (86) 21-5037-2222/Fax: (86) 21-5037-2200

Authorized Distributor:

Cat. No. W550-E1-04 0119