Motion Coordinate System

1756-HYD02, 1756-M02AE, 1756-M02AS, 1756-M03SE, 1756-M08SE, 1756-M16SE, 1768-M04SE

Rockwell Automation Publication MOTION-UM002I-EN-P - March 2022
Supersedes Publication MOTION-UM002H-EN-P - November 2021

User Manual

Original Instructions

Important User Information

Read this document and the documents listed in the additional resources section about installation, configuration, and operation of this equipment before you install, configure, operate, or maintain this product. Users are required to familiarize themselves with installation and wiring instructions in addition to requirements of all applicable codes, laws, and standards.

Activities including installation, adjustments, putting into service, use, assembly, disassembly, and maintenance are required to be carried out by suitably trained personnel in accordance with applicable code of practice.

If this equipment is used in a manner not specified by the manufacturer, the protection provided by the equipment may be impaired.

In no event will Rockwell Automation, Inc. be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment.

The examples and diagrams in this manual are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular installation, Rockwell Automation, Inc. cannot assume responsibility or liability for actual use based on the examples and diagrams.

No patent liability is assumed by Rockwell Automation, Inc. with respect to use of information, circuits, equipment, or software described in this manual.

Reproduction of the contents of this manual, in whole or in part, without written permission of Rockwell Automation, Inc., is prohibited.

Throughout this manual, when necessary, we use notes to make you aware of safety considerations.

WARNING: Identifies information about practices or circumstances that can cause an explosion in a hazardous environment, which may lead to personal injury or death, property damage, or economic loss.

ATTENTION: Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss. Attentions help you identify a hazard, avoid a hazard, and recognize the consequence.

IMPORTANT Identifies information that is critical for successful application and understanding of the product.

Labels may also be on or inside the equipment to provide specific precautions.

SHOCK HAZARD: Labels may be on or inside the equipment, for example, a drive or motor, to alert people that dangerous voltage may be present.

BURN HAZARD: Labels may be on or inside the equipment, for example, a drive or motor, to alert people that surfaces may reach dangerous temperatures.

ARC FLASH HAZARD: Labels may be on or inside the equipment, for example, a motor control center, to alert people to potential Arc Flash. Arc Flash will cause severe injury or death. Wear proper Personal Protective Equipment (PPE). Follow ALL Regulatory requirements for safe work practices and for Personal Protective Equipment (PPE).

Chapter 3

Geometries with no orientation support

Chapter 4

Geometries with orientation support

Chapter 5

Configure Camming

Index

This manual provides information on how to configure various coordinated motion applications. Use this table to choose a motion coordinated instruction. Information about the coordinate instructions can be found in the Logix5000™ Controllers Motion Instruction Reference Manual, publication MOTION-RM002.

If you want to	Use this instruction
Initiate a single or multi-dimensional linear coordinated move for the specified axes within a Cartesian coordinate system.	Motion Coordinated Linear Move (MCLM)
Initiate a two- or three-dimensional circular coordinated move for the specified axes within a Cartesian coordinate system.	Motion Coordinated Circular Move (MCCM)
Initiate a change in path dynamics for coordinate motion active on the specified coordinate system.	Motion Coordinated Change Dynamics (MCCD)
Stop the axes of a coordinate system or cancel a transform.	Motion Coordinated Stop (MCS)
Initiate a controlled shutdown of all of the axes of the specified coordinate system.	Motion Coordinated Shutdown (MCSD)
Start a transform that links two coordinate systems together.	Motion Coordinated Transform (MCT)(1)
Start a transform that links to coordinate systems together. The MCTO instruction incorporates translation and orientation in its position transformation.	Motion Coordinated Transform with Orientation (MCTO)(2)
Calculate the position of one coordinate system with respect to another coordinate system.	Motion Calculate Transform Position (MCTP)(1)
Calculate the position of a point in one coordinate system to the equivalent point in a second coordinate system.	Motion Coordinated Transform Position with Orientation (MCTPO)(2)
Initiate a reset of all of the axes of the specified coordinate system from the shutdown state to the axis ready state and clear the axis faults.	Motion Coordinated Shutdown Reset (MCSR)
Start a single or multi-dimensional linear coordinated path move (CP) for the specified axes within a Cartesian coordinate system.	Motion Coordinated Path Move (MCPM)(2)

(1) Instruction cannot be used with SoftLogix™ controllers.

(2) Instruction only available for Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, and GuardLogix 5580 controllers.

Studio 5000 environment

The Studio 5000 Automation Engineering & Design Environment® combines engineering and design elements into a common environment. The first element is the Studio 5000 Logix Designer® application. The Logix Designer application is the rebranding of RSLogix 5000® software and will continue to be the product to program Logix 5000™ controllers for discrete, process, batch, motion, safety, and drive-based solutions.

The Studio 5000® environment is the foundation for the future of Rockwell Automation® engineering design tools and capabilities. The Studio 5000 environment is the one place for design engineers to develop all elements of their control system.

This manual contains new and updated information. Use these reference tables to locate new or changed information.

Grammatical and editorial style changes are not included in this summary.

Global changes

This table identifies changes that apply to all information about a subject in the manual and the reason for the change. For example, the addition of new supported hardware, a software design change, or additional reference material would result in changes to all of the topics that deal with that subject.

Change	Topic
New Studio 5000 Logix Designer branding	Studio 5000 environment on page 12

New or enhanced features

Topic Name	Reason
Configure the SCARA Independent J1J2J3J6 Coordinate System on page 194	Added section to configure a SCARA Indepent J1J2J3J6 Coordinate System.
Configure an Articulated Dependent J1J2J3J6 robot on page 115	Added section to configure an Articulated Dependent J1J2J3J6 robot.
Configure an Articulated Independent J1J2J3J4J5J6 robot on page 74	Added section to configure an Articulated Independent J1J2J3J4J5J6 robot.
Update application data for managed applications on page 38	Added instructions for updating managed applications, such as robots, to newer versions of characterized data.

Before you begin

This manual is a redesigned manual from publication LOGIX-UM002. A companion manual is available called the SERCOS and Analog Motion Configuration and Start-Up User Manual, publication MOTION-UM001. For CIP motion configuration information, see the CIP Motion Configuration and Startup User Manual, publication MOTION-UM003. If you have any comments or suggestions, please see the back cover of this manual.

Sample projects

The Rockwell Automation sample project's default location is:

c:\Users\Public\Public Documents\Studio 5000\Sample\ENU\v<current_release>\Rockwell Automation

There is a PDF file name **Vendor Sample Projects** that explains how to work with the sample projects. Free sample code is available at http://samplecode.rockwellautomation.com/.

The **Vendor Sample Projects.pdf** default location is:

c:\Users\Public\Public Documents\Studio 5000\Sample\ENU\v<current_release>\Third Party Products

Tip: To access the **Vendor Sample Projects.pdf** file from Logix Designer application, click **Vendor Sample Projects** from the **Help** menu.

Additional resources

These documents contain additional information concerning related Rockwell Automation products. You can view or download publications at http://literature.rockwellautomation.com.

Resource	**Description**
Sercos and Analog Motion Configuration and Startup User Manual, publication MOTION -UM001	Describes how to configure a motion application and to start up your motion solution by using Logix5000 motion modules.
>l5k> Controllers Motion Instructions Reference Manual, publication MOTION-RM002	Provides a programmer with details about motion instructions for a Logix-based controller.
Integrated Motion on the Ethernet/IP Network: Configuration and Startup User Manual, publication MOTION-UM003	Describes how to configure an integrated motion application and to start up your motion solution by using Studio 5000 Logix Designer® application.
Logix5000 Controllers Common Procedures, publication 1756-PM001	Provides detailed and comprehensive information about how to program a Logix5000 controller.
Logix5000 Controllers General Instructions Reference Manual, publication 1756-RM003	Provides a programmer with details about general instructions for a Logix-based controller.
Logix5000 Controllers Process and Drives Instructions Reference Manual, publication 1756-RM006.	Provides a programmer with details about process and drives instructions for a Logix-based controller.
ControlLogix System User Manual, publication 1756-UM001	Describes the necessary tasks to install, configure, program, and operate a ControlLogix® system.
ControlLogix 5580 and GuardLogix 5580 Controllers User Manual, publication 1756-UM543	Provides complete information on how to install, configure, select I/O modules, manage communication, develop applications, and troubleshoot the ControlLogix 5580 and GuardLogix 5580 controllers.
CompactLogix 5370 Controllers User Manual, publication 1769-UM021	Describes the necessary tasks to install, configure, program, and operate a CompactLogix™ system.
GuardLogix Controllers User Manual, publication 1756-UM020	Describes the GuardLogix®-specific procedures you use to configure, operate, and troubleshoot the controller.
GuardLogix 5570 and Compact GuardLogix 5370 Controller Systems Safety Reference Manual, publication 1756-RM099	Contains detailed requirements for achieving and maintaining SIL 3/PLe with the GuardLogix 5570 or CompactLogix 5370 controller safety system, using the Studio 5000 Logix Designer application.
GuardLogix 5580 and Compact GuardLogix 5380 Controller Systems Safety Reference Manual, publication 1756-RM012	Provides information on safety application requirements for GuardLogix 5580 and Compact GuardLogix 5380 controllers in Studio 5000 Logix Designer® applications.
Industrial Automation Wiring and Grounding Guidelines, publication 1770-4.1	Provides general guidelines for installing a Rockwell Automation industrial system.
Product Certifications website, www.rockwellautomation.com/global/certification/overview.page	Provides declarations of conformity, certificates, and other certification details.

Create and configure a coordinate system

Create a Coordinate System

Use the Coordinate System tag to set the attribute values used by the Multi-Axis Coordinated Motion instructions in motion applications. Create the Coordinate System tag before executing any of the Multi-Axis Coordinated Motion instructions.

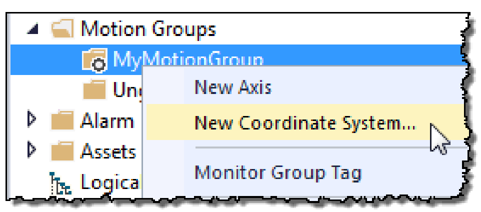
The Coordinate System tag:

- Defines the COORDINATE_SYSTEM data type
- Associates the Coordinate System to a Motion Group
- Associates the axes to the Coordinate System
- Sets the dimension
- Defines the values used by the operands of the Multi-Axis Motion Instructions

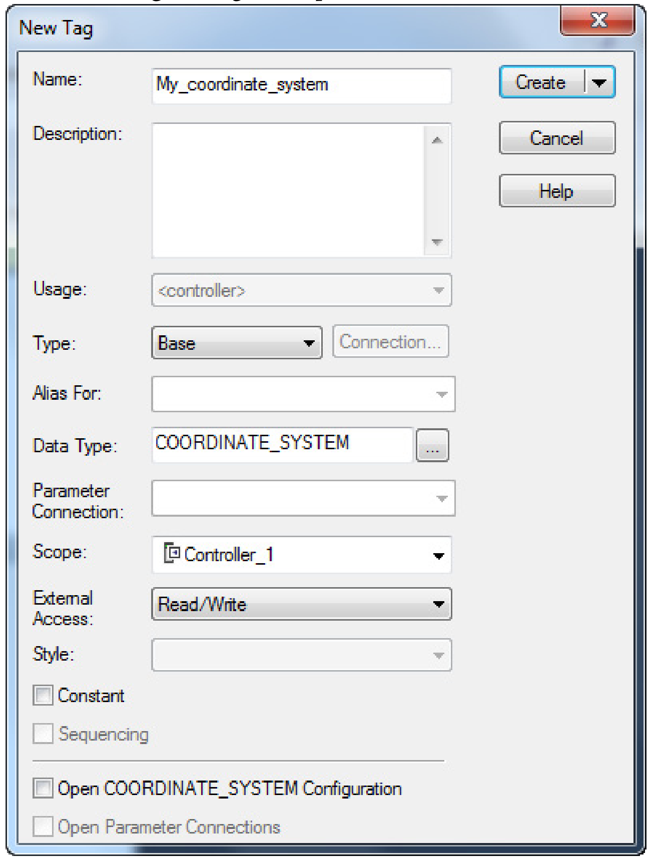
Configuring the Coordinate System tag defines the values for Coordination Units, Maximum Speed, Maximum Acceleration, Maximum Deceleration, Actual Position Tolerance, and Command Position Tolerance.

To create a coordinate system:

1. In the Controller Organizer, right-click the motion group and select **New Coordinate System**.



The **New Tag** dialog box opens.



2. In **Name**, enter the name of the coordinate system.
3. [optional] In **Description**, type a description of the coordinate system.
4. In **Type**, select the type of tag to create. For a coordinate system, the only valid choices are:
   - Base - Refers to a normal tag and is the default
   - Alias - Refers to a tag that references another tag with the same definition
5. In **Data Type**, select **COORDINATE_SYSTEM**.
6. In **External Access**, select whether the tag has None, Read/Write, or Read Only access from external applications such as HMIs.
7. Select **Constant** to prevent executing logic from writing values to the tag. Refer to the online help for more information about the **Constant** check box.

   Tip: Equipment Sequencing is not available when Redundancy is enabled.

8. Select **Open COORDINATE_SYSTEM Configuration** to open the Coordinate System Wizard after creating the tag.

   Once the tag is created, double-click the coordinate system to open the **Coordinate System Properties** dialog box to edit the coordinate system tag.

9. Select **Create** to create the tag.

See also

Coordinate System Properties dialog box on page 19

Edit Coordinate System properties

Use the **Coordinate System Properties** dialog box to modify an existing Coordinate System or configure the Coordinate System.

To edit the Coordinate System properties:

1. In the **Controller Organizer**, expand the **Motion Group** folder, and double-click the **Coordinate System**, or right-click the Coordinate System and select **Properties**.
2. Use the tabs in the **Coordinate System Properties** dialog box to make the appropriate changes. An asterisk appears on the tab to indicate that changes have been made but not implemented.
3. Select **Apply** to save the changes. To exit without saving any changes, select **Cancel**.

See also

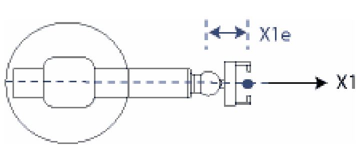
Coordinate System Properties dialog box on page 19

In the Logix Designer application, a coordinate system is a grouping of one or more primary or ancillary axes created to generate coordinated motion. The Logix Designer application supports these geometry types.

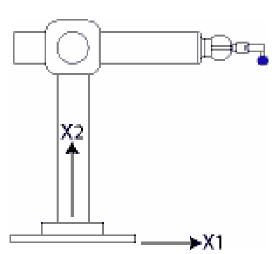
- Cartesian
- Articulated Dependant
- Articulated Independent
- Selective Compliant Assembly Robot Arm (SCARA) Independent
- Delta
- SCARA Delta

These are coordinate system examples.

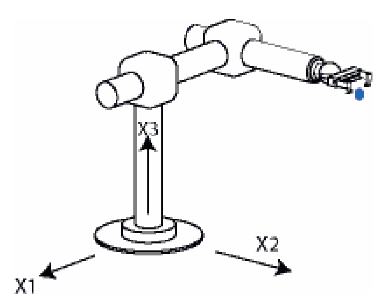
Coordinate system with orthogonal axes



Cartesian coordinate system

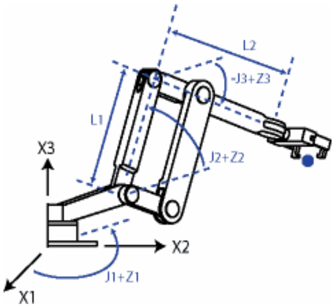


Two-dimensional Cartesian coordinate system

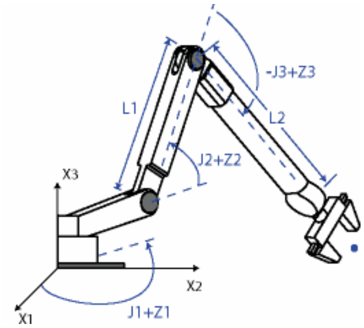


Three-dimensional Cartesian coordinate system

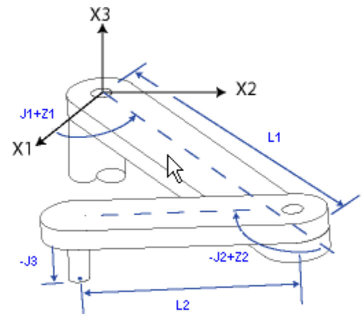
Coordinate systems with non-orthogonal axes



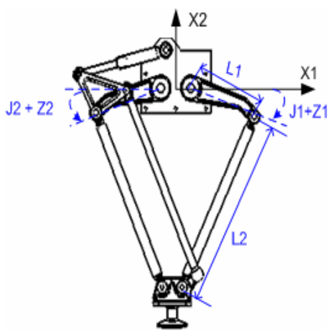
Articulated Dependent coordinate system



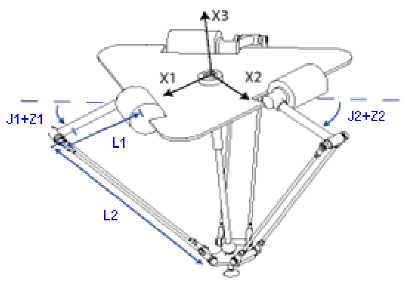
Articulated Independent coordinate system



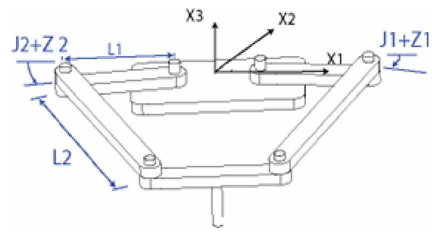
SCARA Independent coordinate system



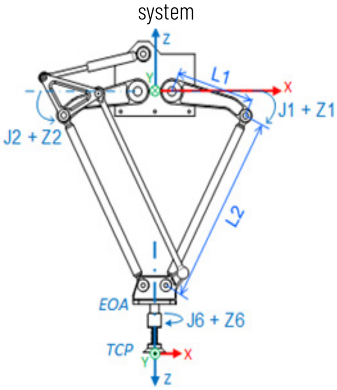
Delta Two-dimensional coordinate system



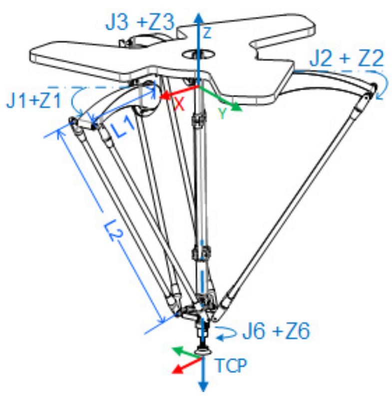
Delta Three-dimensional coordinate system



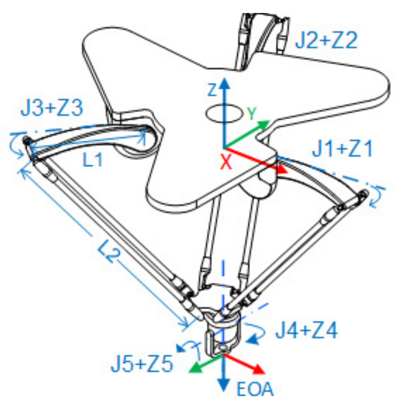
SCARA Delta coordinate system



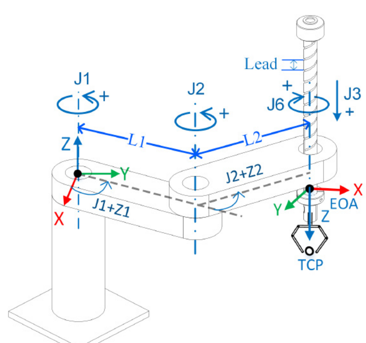
Delta J1J2J6 coordinate system



Delta J1J2J3J6 coordinate system



Delta J1J2J3J4J5 coordinate system



Scara Independent J1J2J3J6

See also

Create a coordinate system on page 15

Determining the coordinate system type on page 35

Coordinate System Properties dialog box

Use the **Coordinate System Wizard** or **Coordinate System Properties** dialog box to configure the Coordinate System tag. The dialog box contains tabs for configuring different facets of the Coordinate System.

Wizard/Coordinate System Properties tab	Description
General	The **General** tab is used to: • Associate the tag to a Motion Group. • Select the coordinate system type. • Select the coordinate definition for the geometry type. • If applicable, specify the number of dimensions and transform dimensions for the geometry type. • Enter the associated axis information. • Select whether to update Actual Position values of the coordinate system automatically during operation.
Geometry	The **Geometry** tab configures key attributes related to non-Cartesian geometry and shows the bitmap of the associated geometry.
Offset	The **Offset** tab configures the offsets for the base and end effector. This tab shows the bitmaps for the offsets related to the geometry.
Units	The **Units** tab defines the **Coordination Units** and the **Conversion Ratios**.
Dynamics	The **Dynamics** tab configures the Vector, Actual and Command Position Tolerance, and Orientation values for a Cartesian coordinate system.
Joints	The **Joints** tab defines the Joints Conversion ratios.
Motion Planner	The **Motion Planner** tab enables or disables Master Delay Compensation or Master Position Filter.
Tag	The **Tag** tab is used to rename the tag, edit the description, and review the **Tag Type**, **Data Type**, and **Scope** information.

Coordinate System Properties dialog box - General tab

How do I open the General tab?

1. In the **Controller Organizer**, expand the **Motion Group** folder, and double-click the coordinate system.
2. On the **Coordinate System Properties** dialog box, click the **General** tab.

Use the settings on **General** tab in the **Coordinate System Properties** dialog box to:

- Associate the coordinate system tag to a Motion Group.
- Select the type of coordinate system to configure.
- Select the coordinate definition based on the robot geometry structure.
- Select the dimension and transform dimension if the coordinate definition is <none>. Otherwise the dimension and transform dimension values are automatically set depending on the geometry type.

- View the catalog number of the robot to which this axis belongs.
- View the current version of characterized data for the application.
- View the unique identifier assigned by Application Code Manager to all objects in the system that comprise one application.
- Specify the number of axes to transform.
- Assign axes to the coordinate system tag.
- Enable or disable automatically updating the tag.

The Logix Designer application supports only one Motion Group tag per controller.

See also

Coordinate System Properties dialog box - General tab parameters on page 20

Coordinate System Properties dialog box - General tab parameters

The settings on the **General** tab in the **Coordinate System Properties** dialog box define the coordinate system. Use the settings to assign the coordinate system to a Motion Group, select the coordinate system type, and enter associated axis information.

Tip: The **Type** selection determines the tabs available in the **Coordinate System Properties** dialog box.

Parameter	Description
Motion Group	The Motion Group associated with the Coordinate System. A Coordinate System assigned to a Motion Group is displayed in the **Motion Groups** folder in the **Controller Organizer**, under the selected Motion Group sub-folder. Selecting <none> terminates the Motion Group association and moves the coordinate system to the **Ungrouped Axes** sub-folder in the **Motions Groups** folder.
...	Opens the **Motion Group Properties** dialog box for the selected Motion Group to edit the motion group properties. If no Motion Group is assigned to this coordinate system, this button appears dimmed.
New Group	Opens the **New Program Parameter or Tag** dialog box to create a new Motion Group tag. This button is available only if no Motion Group has been created.
Type	The robot geometry type associated with the Motion Group. Available choices are: • Cartesian • Articulated Dependent • Articulated Independent • Selective Compliant Assembly Robot Arm (SCARA) Independent • Delta • SCARA Delta

Parameter	Description
Coordinate Definition	Defines the number of coordinates in a coordinate system type. For geometries without orientation support, the coordinate definition defaults to <none>. For geometries with orientation support, the coordinate definition depends on the geometry **Type** selection. Available choices. • <none> • J1J2J6 • J1J2J3J6 • J1J2J3J4J5 • XYZRxRyRz
Dimension	The number of axes that this coordinated system supports. This parameter may be read only depending on the controller and the **Coordination Definition** selection.
Transform Dimension	The number of axes in the coordinate system that you want to transform. This parameter may be read only depending on the controller and the **Coordination Definition** selection. **Tip:** The number of axes to be transformed must be equal to or less than the specified coordinate system dimension. The transform function always begins at the first axis. For example, if the coordinate system has three axes but **Transform Dimension** is set to two axes, then axis one and axis two are transformed. You cannot specify that only axes two and three be transformed.
Application Catalog Number	The catalog number of the robot that this axis belongs to. **Tip:** When an axis is associated to a robot, it might be managed, meaning some axis parameters are not configurable depending on the robot type. Refer to the specific robot documentation for a list of configurable parameters.
Application Version	The current version of characterized data for the application.
Instance	The unique identifier assigned by Application Code Manager to all objects in the system that comprise one application. For example, for a robot, the application is the coordinate system and all joint axes. All of these objects receive the same instance number to indicate that they are part of a specific application.
Axis Grid	Assigns a motion axis to robot geometry joint for control. The five columns in the Axis Grid provide information about the axes in relation to the coordinate system. The number of rows in the grid depends on the robot geometry type and coordinate definition.
Brackets [ ]	Displays the indices in tag arrays used with the current coordinate system. The tag arrays used in multi-axis coordinated motion instructions map to axes using these indices.
Coordinate	Displays the cross-reference to the axes in the grid.

Parameter	Description
Axis Name	Associates an axis tag to the coordinate. The default is <none>. The list displays the Base Tag axes defined in the project. (Alias Tag axes do not display in the list.) The tags can be axes associated with the motion group, axes associated with other coordinated systems, or axes from the **Ungrouped Axes** folder. It is possible to assign fewer axes to the coordinate system than the maximum for the **Dimension** field. However, a warning displays when verifying the coordinate system, and, if left in that state, the instruction generates a run-time error. An axis can be assigned only once in a coordinate system. Ungrouped axes also generate a run-time error.
...	Opens the **Axis Properties** dialog box for the axis.
Coordination Mode	Displays the axes used in the velocity vector calculations. Possible modes: • Ancillary • Primary • Orientation The **Coordination Mode** depends on the **Coordinate Definition** selection.
Enable Coordinate System Auto Tag Update	Determines whether or not the Actual Position values of the current coordinated system are automatically updated during operation. Select the check box to enable this feature. This feature can ease the programming burden when adding GSV statements to the program. However, enabling this feature increases the Coarse Update rate which may impact performance. Whether to use the Coordinate System Auto Tag Update feature depends upon the trade-offs between ease in programming and increase in execution time. **Tip:** Lower the execution time by enabling this feature in initial system programming to formulate the process and then disable it and enter the GSV statements in the program.

See also

Coordinate System Properties dialog box - General tab on page 19

Determine the Coordinate System Type on page 35

Update application data for managed applications on page 38

Coordinate System Properties dialog box - Geometry tab

How do I open the Geometry tab?

1. In the **Controller Organizer**, expand the **Motion Group** folder, and double-click the coordinate system.
2. On the **Coordinate System Properties** dialog box, click the **Geometry** tab.

Use the settings on the **Geometry** tab in the **Coordinate Systems Properties** dialog box to:

- Specify the link lengths in an articulated robotic arm.
- Enter the rotational offset of the individual joint axes.

See also

Coordinate System Properties dialog box - Geometry tab parameters on page 23

Coordinate System Properties dialog box - Geometry tab parameters

The settings on the **Geometry** tab in the **Coordinate System Properties** dialog box define the dimensional characteristics for the robotic geometry type to configure key.

The graphic displayed on the tab shows a typical representation of the type of coordinate system selected on the **General** tab. Your robot typically looks similar to the one shown in the graphic, but can be different depending on the application.

The settings are unavailable for a Cartesian coordinate system.

Parameter	Description
Type	Read-only. The robot geometry type selected on the **General** tab.
Coordinate Definition	Read-only. The coordinate definition selected on the **General** tab.
Dimension	Read-only. The dimension entered on the **General** tab.
Transform Dimension	Read-only. The transform dimension entered on the **General** tab.
Link Lengths	The length of each link in an articulated robotic arm (coordinate system). The measurement units for the articulated coordinate system are defined by the measurement units configured for the affiliated Cartesian coordinate system. The two coordinate systems are linked or affiliated with each other by an MCT instruction. When specifying the link length values be sure that the values are calculated using the same measurement units as the linked Cartesian coordinate system. For example, if the manufacturer specifies the robot link lengths using millimeter units and you want to configure the robot using inches, then convert the millimeter link measurements to inches and enter the values in the appropriate link length fields. **Important:** Be sure that the link lengths specified for an articulated coordinate system are in the same measurement units as the affiliated Cartesian coordinate system. Your system will not work properly if the measurement units are different. The number link identifiers available for configuration is determined by the geometry type and coordinate definition entered on the **General** tab.
Zero Angle Orientation	The rotational offset of the individual joint axes. If applicable, enter the offset value in degrees for each joint axis. The number of angle identifiers available for configuration is determined by the geometry type and coordinate definition entered on the **General** tab.

See also

Coordinate System Properties dialog box - Geometry tab on page 22

Determine the Coordinate System Type on page 35

Coordinate System Properties dialog box - Units tab

How do I open the Units tab?

1. In the **Controller Organizer**, expand the **Motion Group** folder, and double-click the coordinate system.
2. On the **Coordinate System Properties** dialog box, click the **Units** tab.

Use the settings on the **Units** tab in the **Coordinate System Properties** dialog box to:

- Define the units used for measuring and calculating motion-related values such as position and velocity.
- Define the relationship of axis position units to coordination units for each axis.

See also

Coordinate System Properties dialog box - Units tab parameters on page 24

Coordinate System Properties dialog box - Units tab parameters

The settings on the **Units** tab in the **Coordinate System Properties** dialog box define the units of measure and conversion to be used for each coordinate.

Parameter	Description
Type	Read-only. The robot geometry type selected on the **General** tab.
Coordinate Definition	Read-only. The coordinate definition selected on the **General** tab.
Dimension	Read-only. The dimension entered on the **General** tab.
Transform Dimension	Read-only. The transform dimension entered on the **General** tab.
Coordination Units	Defines the units used for measuring and calculating motion-related values such as position and velocity. The coordination units do not need to be the same for each coordinate system. The units are relevant to your application and maximize ease of use. When the **Coordination Units** change, the second portion of the **Coordination Ratio Units** automatically changes to reflect the new units. Coordination Units is the default.
Axis Name	Displays the name of the axis assigned to the coordinate system.
Conversion Ratio	Defines the relationship of axis position units to coordination units for each axis. For example, if the position units for an axis is in millimeters and the axis is associated with a coordinate system whose units are in inches, then the conversion ratio for this axis/coordinate system association is 25.4/1 and can be specified in the appropriate row of the Axis Grid. **Tip:** The numerator can be entered as a float or an integer. The denominator must be entered as an integer only.
Conversion Ratio Units	Displays the axis position units to coordination units used. The coordination units are defined in the **Coordination Units** parameter on this tab. The Axis Position units are defined on the **Units** tab in the **Axis Properties** dialog box. These values are dynamically updated when changes are made to either axis position units or coordination units.

See also

Coordination System Properties dialog box - Units tab on page 23

Coordinate System Properties dialog box - Offsets tab

How do I open the Offsets tab?

1. In the **Controller Organizer**, expand the **Motion Group** folder, and double-click the coordinate system.
2. On the **Coordinate System Properties** dialog box, click the **Offsets** tab.

Use the settings on the **Offsets** tab in the **Coordinate System Properties** dialog box to define the end effector and base offset values for the robotic arm.

The **Offset** tab shows the views of a typical robotic arm based on the configuration of the robot geometry type on the **General** tab. The type of offsets and the number of available offsets is determined by the coordinate system and the number of axes associated with the coordinate system.

When specifying the end effector and base offset values, be sure that the values are calculated using the same measurement units as the linked Cartesian coordinate system. For example, if the manufacturer specifies the robot offset using millimeter units and you want to configure the robot using inches, then convert the millimeter link measurements to inches and enter the values in the appropriate offset fields.

See also

Coordinate System Properties dialog box - Offsets tab parameters on page 25

Coordinate System Properties dialog box - Offsets tab parameters

The settings on the **Offsets** tab in the **Controller System Properties** dialog box define the offsets associated with the coordinate system. The tab also shows the bitmaps for the offsets related to the geometry.

Parameter	Description
Type	Read-only. The robot geometry type selected on the **General** tab.
Coordinate Definition	Read-only. The coordinate definition selected on the **General** tab.
Dimension	Read-only. The dimension entered on the **General** tab.
Transform Dimension	Read-only. The transform dimension entered on the **General** tab.
End Effector Offsets	The length of the end effector. The correct end effector offsets are typically available from the manufacturer. The end effector offset indicators are X1e, X2e and X3e when the Coordination Definition is <none>.

Parameter	Description
Base Offsets	The Logix Designer application Kinematics internal equations define the robot origin relative to the first joint of the robotic arm. The robot manufacturer may specify the origin at a different location. The difference between these two locations is the base offset values. The correct base offset values are typically available from the robot manufacturer. The base offset indicators are X1b, X2b and X3b when the Coordination Definition is <none>.
Base and Effector Plate Dimensions	Rb indicates the Base plate radius and Re indicates the End Effector plate radius. This parameter is available only when the Geometry Type is Delta and the Coordinate Definition is J1J2J3J6 or J1J2J3J4J5.
Swing Arm Offsets	D3, A3, D4, A4, and D5 are offsets indicated in DH parameter style. This parameter is available only when the Geometry Type is Delta and the Coordinate Definition is J1J2J6, J1J2J3J6 or J1J2J3J4J5.
Coupling Direction	Indicates the direction of coupling between J4 and J5. There are 3 options available: • <none> - J4 rotation does not cause any J5 tilt motion • Same - J4 positive rotation causes the tilt motion in the same direction of the positive J5 motion • Opposite - J4 positive rotation causes tilt motion in the opposite direction of positive J5 motion. This parameter is available only when the Geometry Type is Delta and the Coordinate Definition is J1J2J3J4J5.
Coupling Ratio J4:J5	The ratio of the rotation axis to the tilt axis. This parameter is available only when Geometry Type is Delta and the Coordinate Definition is J1J2J3J4J5.

See also

Coordinate System Properties dialog box - Offsets tab on page 25

Determine the Coordinate System Type on page 35

Coordinate System Properties dialog box - Joints tab

How do I open the Joints tab?

1. In the **Controller Organizer**, expand the **Motion Group** folder, and double-click the coordinate system.
2. On the **Coordinate System Properties** dialog box, click the **Joints** tab.

Use the settings on the **Joints** tab in the **Coordinate System Properties** dialog box to define the Joints Conversion ratios. Joint axis units are specified in degrees.

The **Joints** tab is available only if you are configuring non-Cartesian coordinate systems.

See also

Coordinate System Properties dialog box - Joints tab parameters on page 27

Coordinate System Properties dialog box - Joints tab parameters

The settings on the **Joints** tab configure the Joints Conversion ratios. The tab includes the following parameters. Settings that do not pertain to the controller are hidden.

Parameter	Description
Type	Read-only. The robot geometry type selected on the **General** tab.
Coordinate Definition	Read-only. The coordinate definition selected on the **General** tab.
Dimension	Read-only. The dimension entered on the **General** tab.
Transform Dimension	Read-only. The transform dimension entered on the **General** tab.
Axis Name	The name of axis associated with the coordinate system. The names appear in the order that they were configured in the coordinate system.
Joint Ratio	Defines the relationship between the axis position units and degrees. The Joint Ratio is divided into two fields: • The left-half of the Joint Ratio column is used to specify the numerator value of Joint Position units per degree for each joint axis in the system. • The right-half of the Joint Ratio column is used to specify the denominator value of Joint Position units per degree for each joint axis in the system. For example, if axis units are defined in revolutions, then the ratio might be 1/360 revolution/degrees. The denominator is always specified in Degrees. The actual Joint axes units are what is configured for the individual Joint axes.
Joint Units	The configured axis position units to degrees relationship. The Axis Position units are defined on the **Units** tab in the **Axis Properties** dialog box. Joint units are always defined as degrees.

See also

Coordinate System Properties dialog box - Joints tab on page 26

Coordinate System Properties dialog box - Dynamics tab

How do I open the Dynamics tab?

1. In the **Controller Organizer**, expand the **Motion Group** folder, and double-click the coordinate system.
2. On the **Coordinate System Properties** dialog box, on the **General** tab, select **Cartesian** as the **Type**.
3. Click the **Dynamics** tab.

Use the settings on the **Dynamics** tab in the **Coordinate System Properties** dialog box to enter Vector, Actual and Command Position Tolerance, and Orientation values for a Cartesian coordinate system.

The **Dynamics** tab is only available when configuring a Cartesian coordinate system.

See also

Coordinate System Properties dialog box - Dynamics tab parameters on page 28

Coordinate System Properties dialog box - Dynamics tab parameters

The settings on the **Dynamics** tab in the **Coordinate System Properties** dialog box are used to enter vector, position and tolerance, and orientation values for a Cartesian coordinate system.

The **Vector** values are used by the Coordinated Motion instructions in calculations when the operands are expressed as percent of Maximum. The **Coordination Units** automatically change when the coordination units are redefined on the **Units** tab.

The **Orientation** values are used by the Motion Coordinate Path Move (MCPM) instruction. These values are always in units of degrees, and only available when **System Type** is Cartesian and **Coordinate Definition** is <none>.

Parameter	Description
Vector Maximum Speed	The value used by the Coordinated Motion instructions to calculate vector speed when speed is expressed as a percent of maximum.
Vector Maximum Acceleration	The value used by the Coordinated Motion instructions to determine the acceleration rate to apply to the coordinate system vector when acceleration is expressed as a percent of maximum.
Vector Maximum Deceleration	The value used by the Coordinated Motion instructions to determine the deceleration rate to apply to the coordinate system vector when deceleration is expressed as a percent of maximum. The Maximum Deceleration value must be a non-zero value to achieve any motion using the coordinate system.

Parameter	Description
Vector Maximum Acceleration Jerk	The maximum acceleration jerk rate of the axis. The jerk parameters only apply to S-curve profile moves using these instructions: • MCS • MCCD • MCCM • MCLM The Maximum Acceleration Jerk rate of the coordinate system, in Coordination Units/second^3, defaults to 100% of the maximum acceleration time. The speed and the acceleration rate for this calculation are defined as: MaxAccel^2/Speed = Maximum Acceleration Jerk This value is used when the motion instruction is set with Jerk Units=% of Maximum. When a Multi-axis Motion Instruction has Jerk Units=units per sec^3 then the maximum acceleration jerk value is derived from the motion instruction faceplate. The jerk units for the motion instruction also allow for Jerk Units=% of Time, with 100% of Time. This means that the entire S-curve move will have Jerk limiting. This is the default mode. An S-curve move with 0% of Time will result in a trapezoidal profile, and have 0% Jerk limiting. If set manually, enter the value in units=Coordination Units/second^3 units. Use **Calculate** to view this value in terms of units=% of Time.
Vector Maximum Deceleration Jerk	The maximum deceleration jerk rate of the axis. The jerk parameters only apply to S-curve profile moves using these instructions: • MCS • MCCD • MCCM • MCLM The Maximum Deceleration Jerk rate of the coordinate system, in Coordination Units/second^3, defaults to 100% of the maximum deceleration time. The speed and deceleration rate for the calculation are defined as: MaxDecel^2/Speed - Maximum Deceleration Jerk This value is used when the motion instruction is set with Jerk Units=% of Maximum. When a Multi-axis motion instruction has Jerk Units=units per sec^3 then the Max Deceleration Jerk value is derived from the Motion Instruction faceplate. The jerk units for the motion instruction also allow for Jerk Units=% of Time, with 100% of Time meaning the entire S-curve move will have Jerk limiting, which is the default mode. An S-curve move with 0% of Time will result in a trapezoidal profile, and have 0% Jerk limiting. If set manually, enter the value in units=Coordination Units/second^3 units. Use **Calculate** to view the value in terms of units=% of Time.
Calculate	Opens the **Calculate Maximum Acceleration/Deceleration Jerk** dialog box to view and set the Maximum Acceleration or Maximum Deceleration Jerk in terms of the Jerk Units=% of Time. **Calculate** is available only when the software is online with the controller.

Parameter	Description
Actual	The value in coordination units, for Actual Position to be used by Coordinated Motion instructions when they have a Termination Type of Actual Tolerance.
Command	The value in coordination units, for Command Position to be used by Coordinated Motion instructions when they have a Termination Type of Command Tolerance.
Orientation Maximum Speed	The maximum speed of the orientation axes of the coordinate system. This value is used by the Motion Coordinate Path Move (MCPM) instruction.
Orientation Maximum Acceleration	The maximum acceleration of the orientation axes of the coordinate system. This value is used by the Motion Coordinate Path Move (MCPM) instruction.
Orientation Maximum Deceleration	The Maximum deceleration of the orientation axes of the coordinate system. This value is used by the Motion Coordinate Path Move (MCPM) instruction.
Manual Adjust	Opens the **Manual Adjust Properties** dialog box to allow changes to the Vector, Position Tolerance, and Orientation values. **Manual Adjust** is available when online with the controller and there are no pending edits.

See also

Coordinate System Properties dialog box - Dynamics tab on page 27

Manual Adjust dialog box - Dynamics tab on page 30

Manual Adjust dialog box - Dynamics tab

How do I open the Manual Adjust dialog box?

1. In the **Controller Organizer**, expand the **Motion Group** folder, and double-click the coordinate system.
2. On the **Coordinate System Properties** dialog box, click the **Dynamics** tab, and then click **Manual Adjust**.

Use the settings on the **Dynamics** tab in the **Manual Adjust** dialog box to change Vector, Position Tolerance and Orientation values. Changes can be made either online or offline.

When a value changes, a blue arrow appears next to it. This means the values are immediately updated to the controller if online or to the project file if offline.

Parameter	Description
Vector Maximum Speed	The value used by the Coordinated Motion instructions to calculate vector speed when speed is expressed as a percent of maximum.
Vector Maximum Acceleration	The value used by the Coordinated Motion instructions to determine the acceleration rate to apply to the coordinate system vector when acceleration is expressed as a percent of maximum.

Parameter	Description
Vector Maximum Deceleration	The value used by the Coordinated Motion instructions to determine the deceleration rate to apply to the coordinate system vector when deceleration is expressed as a percent of maximum. The Maximum Deceleration value must be a non-zero value to achieve any motion using the coordinate system.
Vector Maximum Accel Jerk	The maximum acceleration jerk rate of the axis. The Maximum Acceleration Jerk rate of the coordinate system, in Coordination Units/second^3, defaults to 100% of the maximum acceleration time. The speed and the acceleration rate for this calculation are defined as: MaxAccel^2/Speed = Maximum Acceleration Jerk This value is used when the motion instruction is set with Jerk Units=% of Maximum.
Vector Maximum Decel Jerk	The maximum deceleration jerk rate of the axis. The Maximum Deceleration Jerk rate of the coordinate system, in Coordination Units/second^3, defaults to 100% of the maximum deceleration time. The speed and deceleration rate for the calculation are defined as: MaxDecel^2/Speed - Maximum Deceleration Jerk This value is used when the motion instruction is set with Jerk Units =% of Maximum.
Actual	The value in coordination units, for Actual Position to be used by Coordinated Motion instructions when they have a Termination Type of Actual Tolerance.
Command	The value in coordination units, for Command Position to be used by Coordinated Motion instructions when they have a Termination Type of Command Tolerance.
Orientation Maximum Speed	The maximum speed of the orientation axes of the coordinate system.
Orientation Maximum Acceleration	The maximum acceleration of the orientation axes of the coordinate system.
Orientation Maximum Deceleration	The Maximum deceleration of the orientation axes of the coordinate system.
Reset	Returns the values back to their initial values. The values are immediately reset when clicking **Reset**.

See also

Coordinate System Properties dialog box - Dynamics tab parameters on page 28

Coordinate System Properties dialog box - Motion Planner tab

How do I open the Motion Planner tab?

1. In the **Controller Organizer**, expand the **Motion Group** folder, and double-click the coordinate system.
2. On the **Coordinate System Properties** dialog box, click the **Motion Planner** tab.

Use the settings on the **Motion Planner** tab in the **Coordinate System Properties** dialog box to:

- Enable or disable Master Delay Compensation.

- Enable or disable Master Position Filter.
- Enter the bandwidth for the Master Position Filter.

The **Motion Planner** tab is available only when configuring a Cartesian coordinate system

See also

Coordinate System Properties dialog box - Motion Planner tab parameters on page 33

Coordinate System Properties dialog box - Motion Planner tab parameters

The settings on the **Motion Planner** tab specify whether to enable or disable Master Delay Compensation and Master Position Filter.

Parameter	Description
Master Delay Compensation	Determines whether to enable or disable Master Delay Compensation. The Master Delay Compensation is used to balance the delay time between reading the Master Axis command position and applying the associated slave command to the slave's servo loop. It ensures that the slave coordinate command position accurately tracks the actual position of the Master Axis (that is, zero tracking error when gearing or camming to the actual position of a Master Axis for Cartesian coordinate motion in Master Driven mode). Clear the check box to disable Master Delay Compensation. Tips: • If the axis is configured for Feedback only, disable Master Delay Compensation. • In some applications, there is no requirement for zero tracking error between the Master and the Slave axis. In these cases, it may be beneficial to disable Master Delay Compensation to eliminate the disturbances introduced to the Slave Axis. • Master Delay Compensation, even if it is enabled, is not applied in cases where a Slave Axis is gearing or camming to the Master Axis's command position because there is no need to compensate for master position delay.
Enable Master Position Filter	Determines whether to enable or disable Master Position Filter. The Master Position Filter filters the specified master axis position input to the slave axis's gearing or position camming operation. The filter smooths out the actual position signal from the Master Axis, and thus smooths out the corresponding motion of the Slave Axis. Select the check box to enable the Master Position Filter.
Master Position Filter Bandwidth	The bandwidth used for master position filter. This parameter is only available when Master Position Filter is enabled. **Tip:** Entering a zero also disables the Master Position Filter.

See also

Coordinate System Properties dialog box - Motion Planner tab on page 31

Coordinate System Properties dialog box - Tag tab

How do I open the Tag tab?

1. In the **Controller Organizer**, expand the **Motion Group** folder, and double-click the coordinate system.
2. On the **Coordinate System Properties** dialog box, click the **Tag** tab.

Use the settings on the **Tag** tab in the **Coordinate System Properties** dialog box to modify the name and description of the coordinate system. When the controller is online, the parameters are read-only.

Tip: Save your changes before going online. Otherwise, pending changes revert to their previously-saved state.

See also

Coordinate System Properties dialog box - Tag tab parameters on page 34

Coordinate System Properties dialog box - Tag tab parameters

The settings on the **Tag** tab in the **Coordinate System Properties** dialog box provide information about the Coordinate System tag. The tag name and description can be updated only when the application is offline.

Tip: Save the changes before going online. Otherwise, pending changes revert to their previously-saved state.

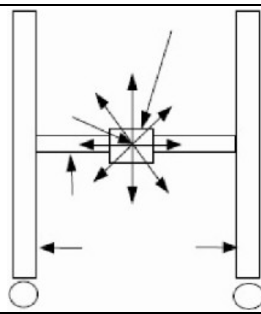
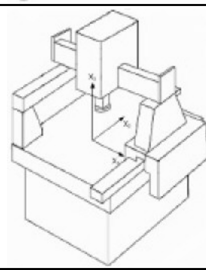
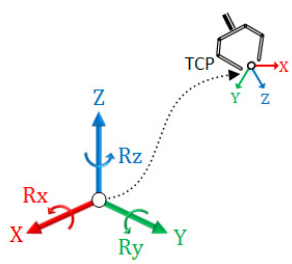
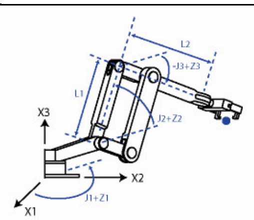
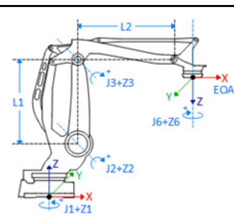
Parameter	Description
Name	The name of the tag. The name can be up to 40 characters and can include letters, numbers, and underscores (_).
Description	The description for the tag.
Type	The type of Coordinate System tag. Coordinate System tags can be either a base or an alias tag.
Data Type	The data type of the Coordinate System tag.
Scope	Displays the scope of the Coordinate System tag. Coordinate System tags can only be controller scope tags.
Class	Displays the class of the Coordinate System tag. Coordinate System tags can only be a Standard class.
External Access	Displays whether the Coordinate System tag has Read/Write, Read Only, or no access (NONE) from external applications such as HMIs.

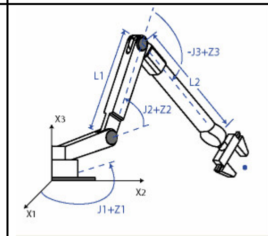
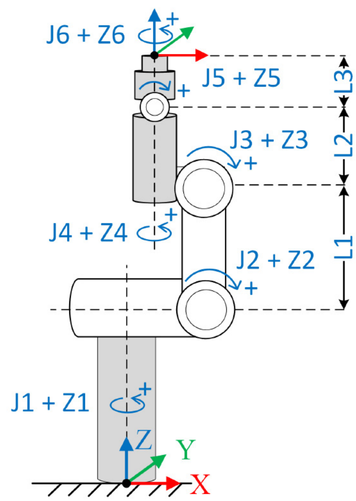
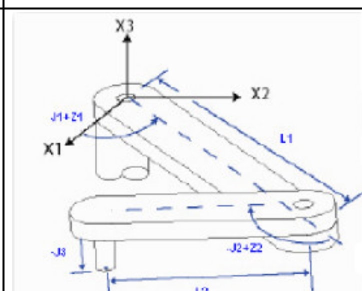
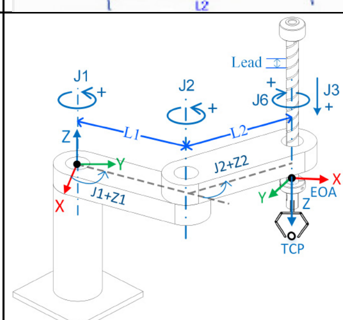
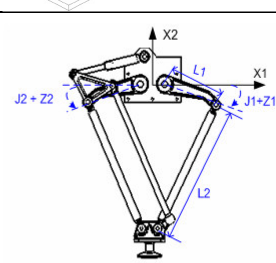
See also

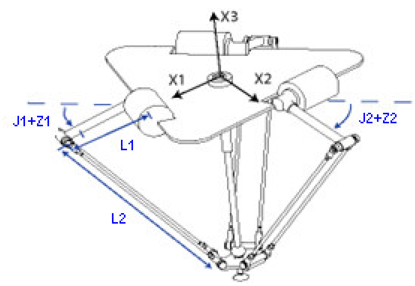
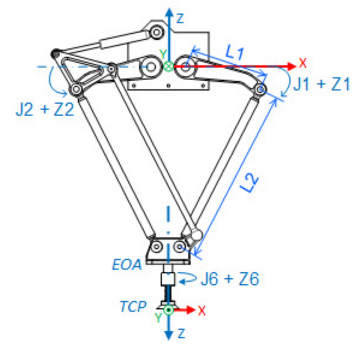
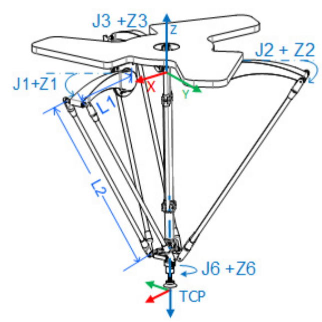
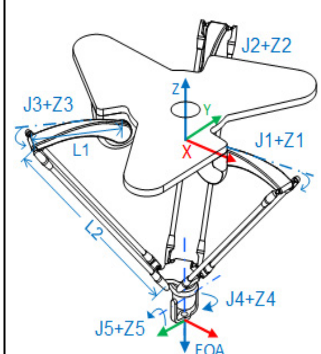
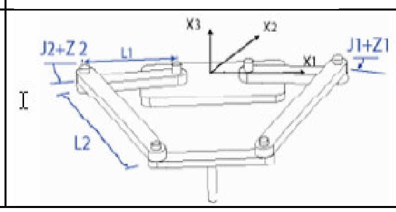
Coordinate System Properties dialog box - Tag tab on page 33

Determine the Coordinate System type

Use this table to help determine the type of Kinematics coordinate system you need.

Geometry Type	Coordinate Definition	Transform Dimension	The robot will look similar to:	See also
Cartesian	<none>	2		Configure a Cartesian H-bot on page 159
Cartesian	<none>	3		Configure a Cartesian Gantry robot on page 158
Cartesian	XYZRxRyRz	6		Configure a Cartesian XYZRxRyRz Coordinate System on page 39
Articulated Dependent	<none>	2 or 3		Configure an Articulated Dependent robot on page 108
Articulated Dependent	J1J2J3J6	4		Configure an Articulated Dependent Robot on page 108

Geometry Type	Coordinate Definition	Transform Dimension	The robot will look similar to:	See also
Articulated Independent	<none>	2 or 3		Configure an Articulated Independent robot on page 65
Articulated Independent	J1J2J3J4J5J6	6		Configure an Articulated Independent Robot on page 65
SCARA Independent	<none>	2		Configure a SCARA Independent Robot on page 155
SCARA Independent	J1J2J3J6	4		Configure a SCARA Independent on page 155
Delta	<none>	2		Configure a Delta Two-dimensional robot on page 145

Geometry Type	Coordinate Definition	Transform Dimension	The robot will look similar to:	See also
Delta	<none>	3		Configure a Delta Three-dimensional robot on page 136
Delta	J1J2J6	3		Configuring a Delta J1J2J6 robot on page 210
Delta	J1J2J3J6	4		Configuring a Delta J1J2J3J6 robot on page 223
Delta	J1J2J3J4J5	5		Configure a Delta J1J2J3J4J5 robot on page 236
SCARA Delta	<none>	2		Configuring a SCARA Delta robot on page 150

See also

Coordinate System Properties dialog boxes on page 19

Update application data for managed applications

Use the **Update Application Data** dialog to update managed applications, such as robots, to newer versions of characterized data. The characterized data determines the parameter settings for the application.

To update application data for managed applications

1. On the main menu, select **Tools > Motion > Update Application Data** to open the **Update Application Data** dialog.
2. The application table lists all the managed applications in the system. The table lists this information for each application:
   - **Name**. The name of the managed application. The name is the tag name of the object and appears in the tag editor. The Logix Designer application assigns the name when you create the object.
   - **Catalog Number**. The Rockwell Automation catalog number for the application.
   - **Instance**. The unique identifier assigned by Application Code Manager to all objects in the system that comprise one application. For example, for a robot, the application is the coordinate system and all joint axes. All of these objects receive the same instance number to indicate that they are part of a specific application.
   - **Version**. The current version of characterized data for the application.
3. In the **Version** column, select a version of characterized data for an application.

   When the version for a managed object is more recent than the version in the repository, a warning appears and states that the repository does not contain this version and should be updated. This version mismatch can happen when you upload a project from a controller to an instance of the Logix Designer application that has an out-of-date repository. You can update the repository or select an older version of characterized data for the managed application. You can find and download new repositories on the Product Compatibility and Download Center (PCDC).
   When you select a newer version of characterized data for an application, an asterisk appears in front of the name of the object to indicate there are un-applied edits, and the **Update** button is enabled.
4. Select **Update** to apply the changes to every application for which you updated the version.

   If an error occurs, a message indicates that there was a problem and the updates are canceled. The dialog stays open so you can make adjustments and try the updates again.

Cartesian coordinate system

Use this information to configure a Cartesian coordinate system.

See also

Program coordinate system with no orientation on page 42

Configure a Cartesian coordinate system

Use these guidelines to configure a Cartesian coordinate system in the **Coordinate System Properties** dialog box.

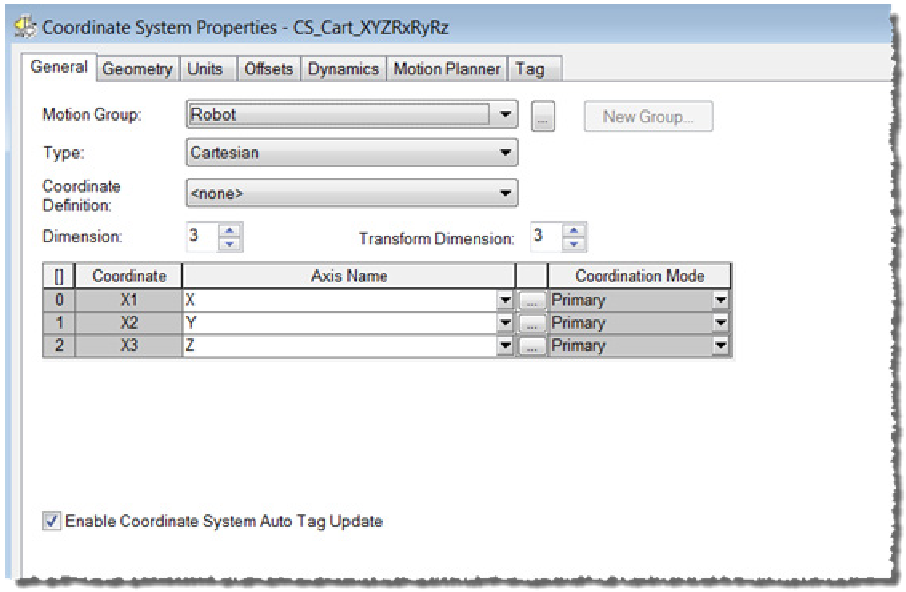
General tab

On the **General** tab, select **Cartesian** as the coordinate system type. There are two **Coordination Definitions** available for a Cartesian coordinate system:

- <none>
- XYZRxRyRz

Select **<none>** to configure the Cartesian coordinate system without orientation support and then select the **Dimension** and **Transform Dimension** for the coordinate system. The **Dimension** and **Transform Dimension** can range from 0 to 3.

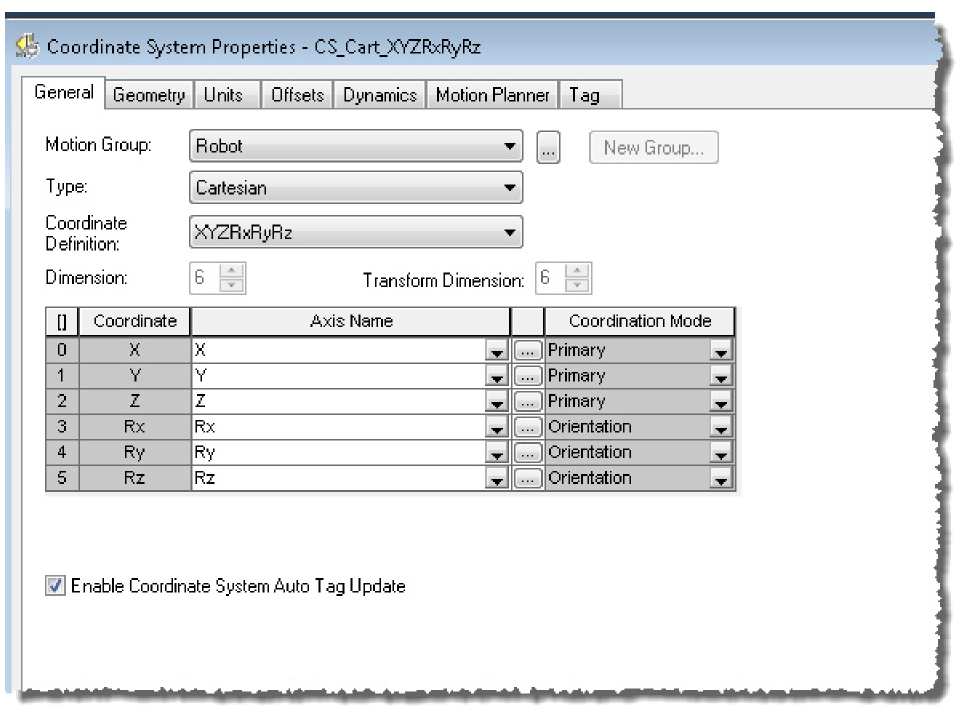
The **Coordinate** column displays X1, X2 or X3, depending on the Dimension and Transform Dimension. The **Coordination** mode is **Primary** for all the axes.



Select **XYZRxRyRz** to configure a Cartesian coordinate system with orientation support. The **Dimension** and **Transform Dimension** values are automatically set to 6 and are unavailable to modify.

The **Coordinate** column displays the World Cartesian Coordinate names X, Y, and Z for the Primary axes and Rx, Ry, and Rz for the Orientation axes. Rx is the rotation around the X axis, Ry is the rotation around the Y axis, and Rz is the rotation around the Z axis, with X-Y-Z fixed angle rotation.

In the **Axis Name** column, associate an axis tag to each coordinate.



Geometry tab

On the **Geometry** tab, the **Link Length** and **Zero Angle Orientation** parameters are unavailable. These parameters are not applicable for the Cartesian coordinate system.

Offsets tab

Set the **Coordinate Definition** to **<none>**, then click the **Offsets** tab to configure the **End Effector Offsets** and the **Base Offsets**.

The available parameters depend on the **Transform Dimension** value.

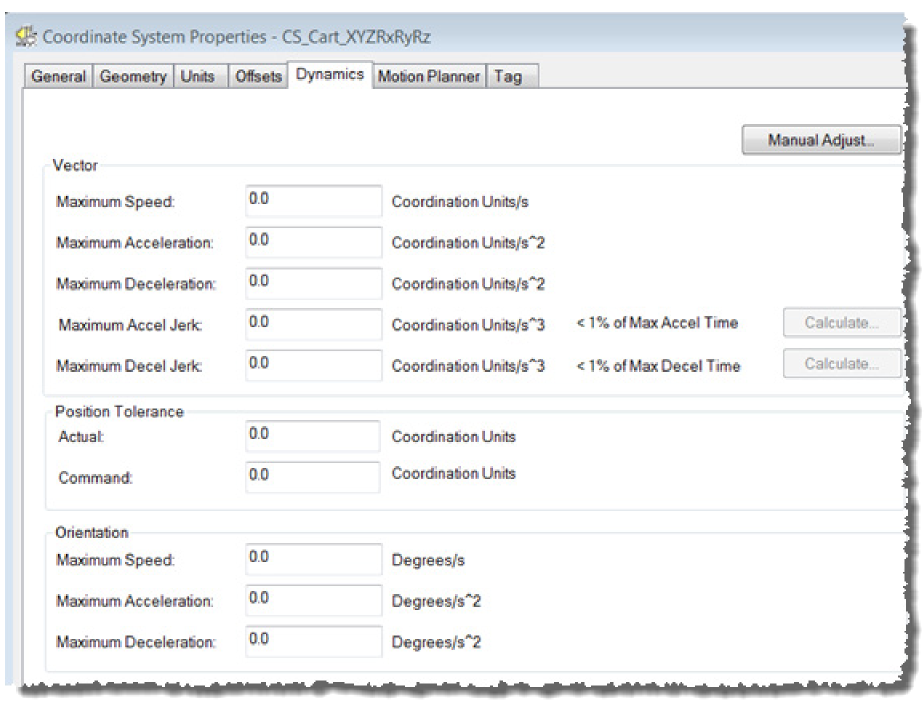
Tip: The **Base Offsets** and **End Effector Offsets** parameters are unavailable if the **Coordinate Definition** is XYZRxRyRz.

Dynamics tab

The **Dynamics** tab is only valid for a Cartesian coordinate system. Use the tab to configure the orientation values required for the Motion Coordinated Path Move (MCPM) instruction:

- Orientation Maximum Speed
- Orientation Maximum Acceleration
- Orientation Maximum Deceleration

The **Orientation** parameters are only available on the **Dynamics** tab when **Type** is Cartesian and **Coordinate Definition** is XYZRxRyRz. The orientation values are always in units of degrees.



Tip: The parameters on the **Dynamics** tab are unavailable when online. To update the parameters, click **Manual Adjust**.

See also

Coordinate System Properties dialog box on page 19

Program coordinate system with no orientation

Use these multi-axis coordinated motion instructions to perform linear and circular moves in single and multidimensional spaces. A Cartesian coordinate system with no orientation in the Logix Designer application can include one, two, or three axes.

Instruction	Description
Motion Coordinated Linear Move (MCLM)	Use the MCLM instruction to start a single or multi-dimensional linear coordinated move for the specified axes within a Cartesian coordinate system.
Motion Coordinated Circular Move (MCCM)	Use the MCCM instruction to initiate a two or three-dimensional circular coordinated move for the specified axes within a Cartesian coordinate system.
Motion Coordinated Transform (MCT)	Use the MCT instruction to start a transform that links two coordinate systems together.
Motion Calculate Transform Position (MCTP)	Use the MCTP instruction to calculate the position of a point in one coordinate system to the equivalent point in a second coordinate system.

See the Logix 5000 Motion Controllers Instructions Reference Manual, publication MOTION-RM002, for more information about the MCLM, MCCM, MCT, and MCTP instructions.

Blended moves and termination types with MCLM or MCCM

To blend two MCLM or MCCM instructions, start the first one and queue the second one. The tag for the coordinate system gives two bits for queuing instructions.

- MovePendingStatus
- MovePendingQueueFullStatus

For example, this ladder diagram uses coordinate system cs1 to blend Move1 into Move2.

See also

Example ladder diagram for blended instructions on page 42

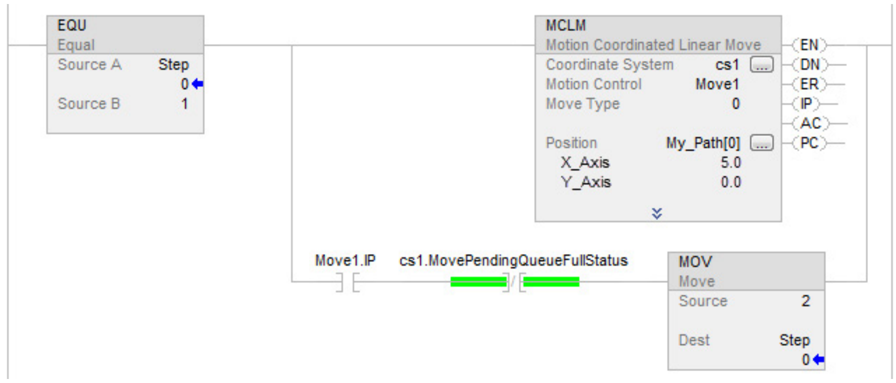
Example ladder diagram for blended instructions

If Step = 1, then:

Move1 starts and moves the axes to a position of 5, 0.

and once Move1 is in process, and there is room to queue another move, then:

Step = 2.



If Step = 2, then:

Move1 is already happening.

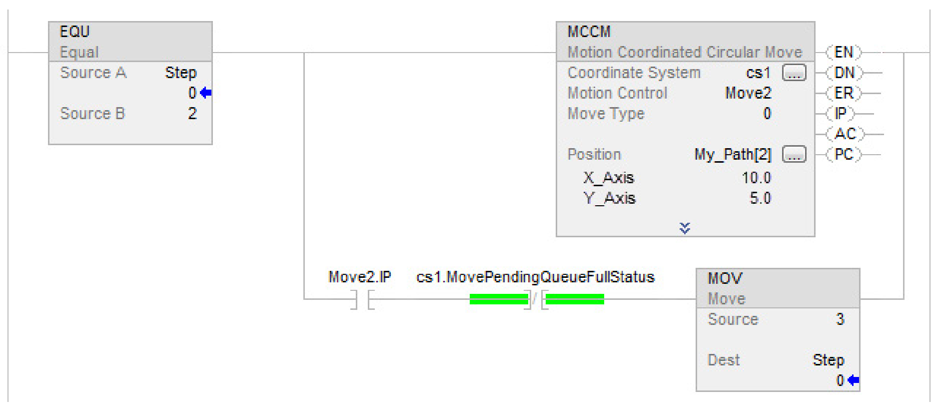
Move2 goes into the queue and waits for Move1 to complete.

When Move1 is complete:

Move2 moves the axes to a position of 10, 5.

And once Move2 is in process and there is room in the queue:

Step = 3.



When an instruction completes, it is removed from the queue and there is space for another instruction to enter the queue. Both bits always have the same value because you can queue only one pending instruction at a time. If the application requires several instructions to be executed in sequence, the bits are set by using these parameters.

When	Then
One instruction is active and a second instruction is pending in the queue	• MovePendingStatus bit = 1 • MovePendingQueueFullStatus bit = 1 • You cannot queue another instruction
An active instruction completes and leaves the queue	• MovePendingStatus bit = 0 • MovePendingQueueFullStatus bit = 0 • You can queue another instruction

The termination type operand for the MCLM or MCCM instruction specifies how the currently executing move gets terminated. These illustrations show the states of instruction bits and coordinate system bits that get affected at various transition points (TP).

The termination types are:

- 0 - Actual tolerance
- 1 - No Settle
- 2 - Command Tolerance
- 3 - No Decel
- 4 - Follow Contour Velocity Constrained
- 5 - Follow Contour Velocity Unconstrained
- 6 - Command Tolerance Programmed

See also

Termination types on page 42

Program coordinate system with orientation

Use these multi-axis coordinated motion instructions to program Cartesian moves on robots with orientation control.

Instruction	Description
Motion Coordinated Path Move (MCPM)	Use the MCPM instruction to start a multi-dimensional coordinated path move for the specified Primary axes (X, Y, Z) and orientation axes (Rx, Ry, Rz) of a Cartesian coordinate system.
Motion Coordinated Transform with Orientation (MCTO)	Use the MCTO instruction to establish a bidirectional transform that is set up between a Cartesian and a robot system with coordinates that are joint axes of a robot. The XYZ translation coordinates and the RxRyRz orientation coordinates in the fixed angle convention define the Cartesian coordinates.
Motion Calculate Transform Position with Orientation (MCTPO)	Use the MCTPO instruction to calculate the position of a point in one coordinate system to the equivalent point in a second coordinate system.

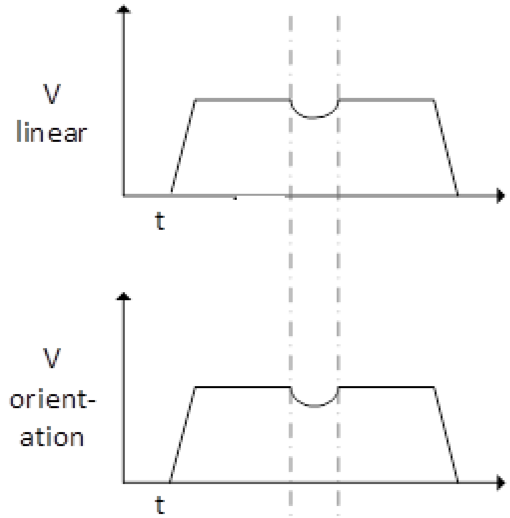
See the Logix 5000 Motion Controllers Instructions Reference Manual, publication MOTION-RM002, for more information about the MCPM, MCTO, and MCTPO instructions.

Blending Path Moves with MCPM

The MCPM instruction supports blending two or more moves together.

Tip: Be sure to review the command tolerance termination type blending for MCLM and MCCM to understand the fundamentals of blending.

- The linear and orientation vector components of the MCPM moves are blended simultaneously.



- The MCPM instruction supports blending through the Blending Termination Type 6. The other blending termination types (Termination Types 2 and 3) are not supported for the MCPM instruction.
- The Termination Type for MCPM is specified via the PATH_DATA member variable TerminationType.
  The Cartesian position where blending should start is specified in the PATH_DATA structure member CommandToleranceLinear.

path[0]	{...}
path[0].InterpolationType	1
path[0].Position	{...}
path[0].RobotConfiguration	0
path[0].TurnsCounters	{...}
path[0].MoveType	0
path[0].TerminationType	6
path[0].CommandToleranceLinear	50.0

- For orientation path blending, there is no equivalent programmable parameter to

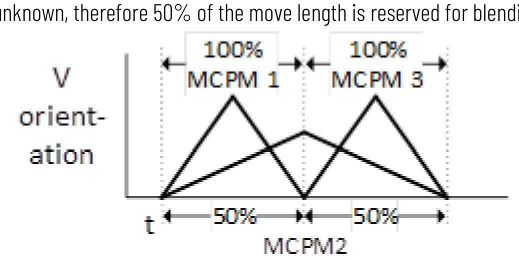
CommandToleranceLinear for specifying start orientation.

Instead, orientation blending is planned to coincide with

- The blended linear trajectory path dynamics, if such a component exists, or
- 100%/50% rules are used to blend the orientation move over the full length of the path move when a linear component does not exist.

In the second case where there is only an orientation component involved in the blend, the planner reserves 100% of the path length for the first and last moves in a series of blended moves. For the blended moves other than first and last, 50% of the path length is reserved for blending.

In the example shown, MCPM1 is a TT6 orientation-only move with a queued MCPM2 TT6 orientation-only move. The MCPM1 move is a starting move, but end move is unknown, therefore 50% of the move length is reserved for blending.

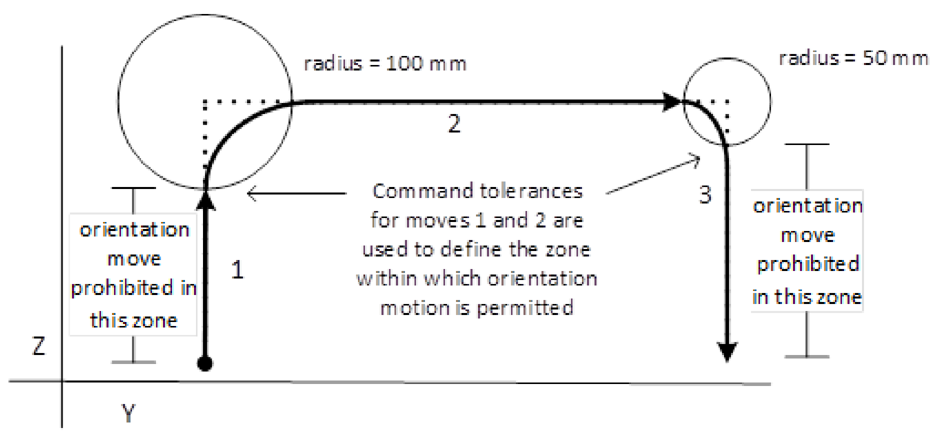


See also

Use MCPM blending with orientation to synchronize Cartesian path and orientation motion on page 46

Use MCPM blending with orientation to synchronize Cartesian path and orientation motion

This is an example for using MCPM blending with orientation to synchronize Cartesian path (CP) and orientation motion.



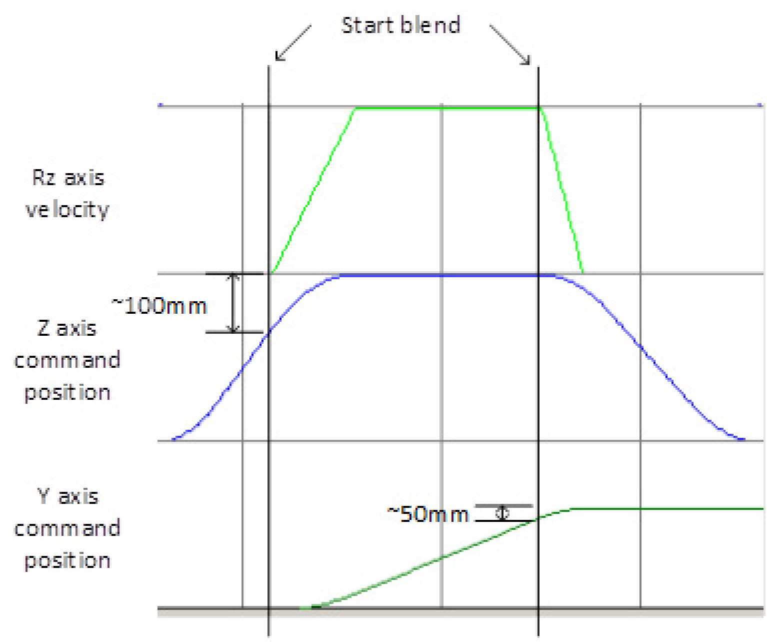
This example shows a robot system using three MCPM instructions to execute a picking trajectory in a pick and place application. The application has the following requirements:

- First move: vertical (Z) move to 300 millimeter height.
- Second move: horizontal (Y) move to the target position 600 millimeters.
- Third move: vertical move 300 millimeters down to the target position.
- The orientation of (Rz) must change by +50.0° by the end of the move trajectory.

- The orientation is prohibited from moving for the first 200 millimeters of move 1, and also prohibited from moving the final 250 millimeters.

Move 1 PATH_DATA	Move 2 PATH_DATA	Move 3 PATH_DATA
1].InterpolationType	,2].InterpolationType	,3].InterpolationType
1].Position	,2].Position	,3].Position
[0,1].Position[0]	ı[0,2].Position[0]	ı[0,3].Position[0]
[0,1].Position[1]	ı[0,2].Position[1]	ı[0,3].Position[1]
[0,1].Position[2]	ı[0,2].Position[2]	ı[0,3].Position[2]
[0,1].Position[3]	ı[0,2].Position[3]	ı[0,3].Position[3]
[0,1].Position[4]	ı[0,2].Position[4]	ı[0,3].Position[4]
[0,1].Position[5]	ı[0,2].Position[5]	ı[0,3].Position[5]
[0,1].Position[6]	ı[0,2].Position[6]	ı[0,3].Position[6]
[0,1].Position[7]	ı[0,2].Position[7]	ı[0,3].Position[7]
[0,1].Position[8]	ı[0,2].Position[8]	ı[0,3].Position[8]
1].RobotConfiguration	,2].RobotConfiguration	,3].RobotConfiguration
1].TurnsCounters	,2].TurnsCounters	,3].TurnsCounters
1].MoveType	,2].MoveType	,3].MoveType
1].TerminationType	,2].TerminationType	,3].TerminationType
1].CommandToleranceL	,2].CommandToleranceL	,3].CommandToleranceL
The vertical move is configured with termination type 6 and the desired command tolerance.	The horizontal move also is termination type 6 with command tolerance.	The final vertical move is blended with the previous when command tolerance is satisfied.

This trend shows the Rz orientation velocity profile and the Z and Y axis position profiles versus time, and illustrates how the linear command tolerance parameter is used with queued MCPM instructions to synchronize the orientation move with respect to the CP linear motion.



For more information about Motion Instructions, see Logix 5000 Controllers Motion Instructions Reference Manual, publication MOTION-RM002.

See also

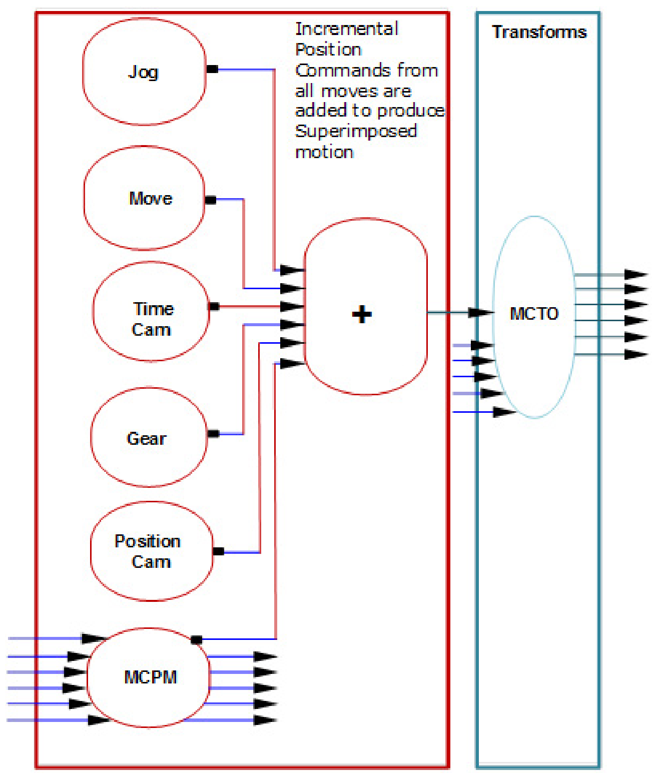
Blending Path Move with MCPM on page 44

Superimposed motion with MCPM

Use the superimposed move feature to superimpose multiple moves/instructions on a single axis. This feature synchronizes a robot's motion with other parts of the application (for example, conveyor tracking and vision systems).

As shown in the illustration, the inputs from various motion instructions are added to produce superimposed motion on a single axis of a coordinate system. The output can be seen on the Transforms side on all or one joint axes of a coordinate system.

As the robot moves with incremental moves, towards the end point, the superimposed move on the concerned axis results in a different axis position than the one programmed on the path point, resulting in joint values which reach the user desired position (thereby tracking the object).



Conveyor belt tracking example

The Kinematics ToolFrame sample project shows an example of conveyor tracking using a 4-axis delta robot. In this example, the conveyor axis is a Master axis which commands the slave axis: X.

The Conveyor axis is moved using a MAJ instruction. When the MCPM instruction is executed, the X position on the path point is added to the X axis position output from the MAG, which is an input into MCTO. MCTO outputs joint values for the robot, there by tracking the object on the conveyor belt.

The application code also superimposes pick cycle moves using absolute coordinated moves to pick the objects from a conveyor belt. Because of the addition of position, the object appears to be on a stationary conveyor. The net result of the superimposed moves, results in the object getting picked from the moving conveyor.

Tip: To use the Kinematic sample projects, on the **Help** menu, click **Vendor Sample Projects** and then click the **Motion** catego
The Rockwell Automation sample project's default location is:
c:\Users\Public\Public Documents\Studio 5000\Sample\ENU\v<current_release>\Rockwell Automation

Bit state diagrams for blended moves

The following diagrams show bit states at the transition points for various types of blended moves.

See also

Bit States at transition points of blended move by using actual tolerance or no settle on page 52

Bit States at transition points of blended move by using no decel on page 51

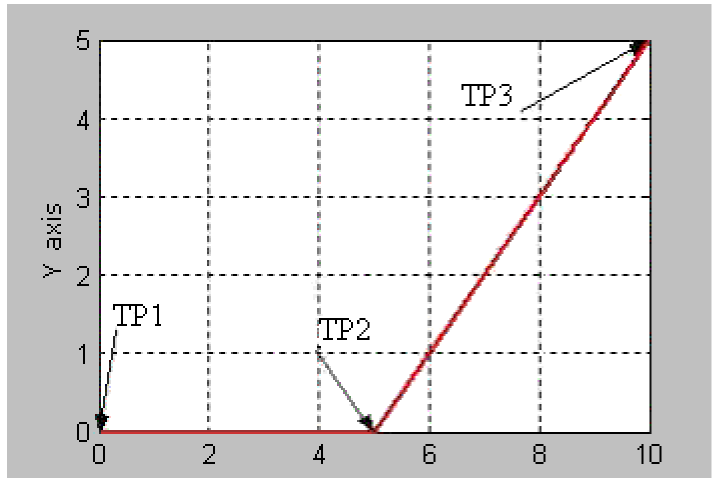
Bit states at transition points of blended move by using command tolerance on page 52

Bit states at transition points of blended move by using follow contour velocity constrained or unconstrained on page 53

Bit States at transition points of blended move by using actual tolerance or no settle

This topic lists the bit states at transition points of Blended Move by using Actual Tolerance or No Settle.

linear → linear move



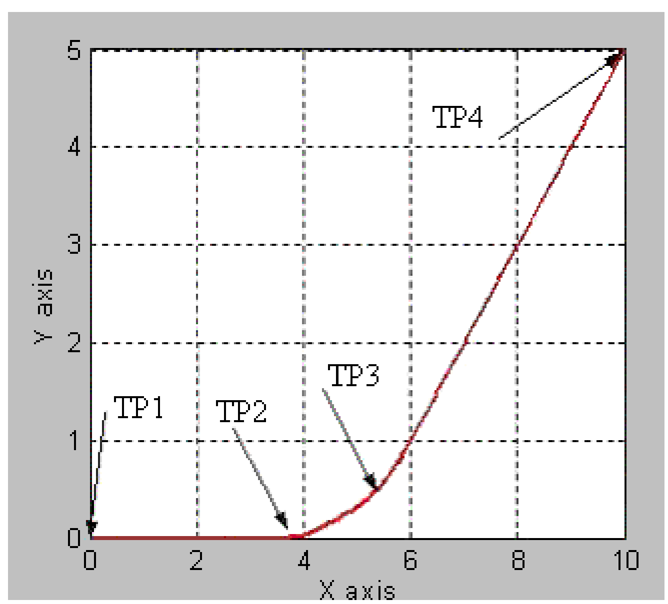
This table shows the bit status at the various transition points shown in the preceding graph with termination type of Actual Tolerance or No Settle.

Bit	TP1	TP2	TP3
Move1.DN	T	T	T
Move1.IP	T	F	F
Move1.AC	T	F	F
Move1.PC	F	T	T
Move2.DN	T	T	T
Move2.IP	T	T	F
Move2.AC	F	T	F
Move2.PC	F	F	T
cs1.MoveTransitionStatus	F	F	F
cs1.MovePendingStatus	T	F	F
cs1.MovePendingQueueFullStatus	T	F	F

Bit States at transition points of blended move by using no decel

This lists the bit states at transition points of blended move by using no decel.

linear → linear move



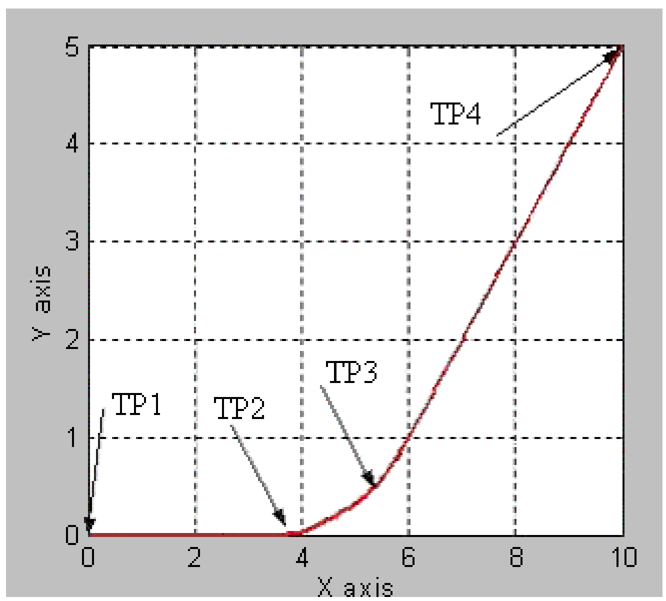
This table shows the bit status at the various transition points shown in the preceding graph with termination type of No Decel. For No Decel termination type distance-to-go for transition point TP2 is equal to deceleration distance for the Move1 instruction. If Move 1 and Move 2 are collinear, then Move1.PC will be true at TP3, which is the programmed end-point of first move.

Bit	TP1	TP2	TP3	TP4
Move1.DN	T	T	T	T
Move1.IP	T	F	F	F
Move1.AC	T	F	F	F
Move1.PC	F	T	T	T
Move2.DN	T	T	T	T
Move2.IP	T	T	T	F
Move2.AC	F	T	T	F
Move2.PC	F	F	F	T
cs1.MoveTransitionStatus	F	T	F	F
cs1.MovePendingStatus	T	F	F	F
cs1.MovePendingQueueFullStatus	T	F	F	F

Bit states at transition points of blended move by using command tolerance

linear → linear move

This lists the bit states at transition points of Blended Move by using Command Tolerance.



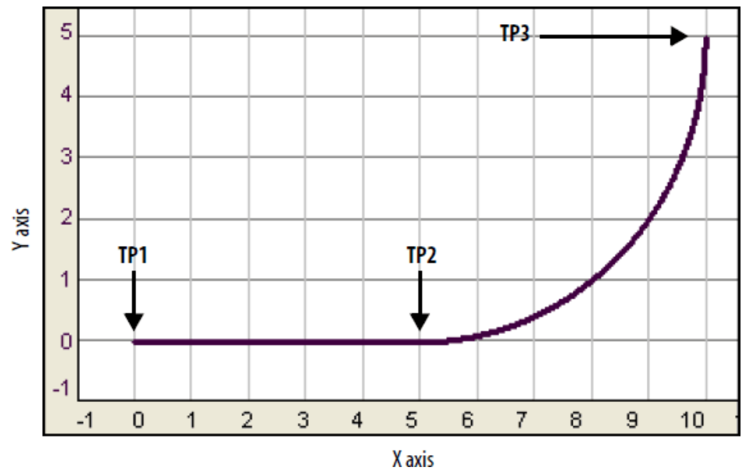
This table shows the bit status at the various transition points shown in the preceding graph with termination type of Command Tolerance. For Command Tolerance termination type distance-to-go for transition point TP2 is equal to Command Tolerance for the coordinate system cs1.

Bit	TP1	TP2	TP3	TP4
Move1.DN	T	T	T	T
Move1.IP	T	F	F	F
Move1.AC	T	F	F	F
Move1.PC	F	T	T	T

Bit	TP1	TP2	TP3	TP4
Move2.DN	T	T	T	T
Move2.IP	T	T	T	F
Move2.AC	F	T	T	F
Move2.PC	F	F	F	T
cs1.MoveTransitionStatus	F	T	F	F
cs1.MovePendingStatus	T	F	F	F
cs1.MovePendingQueueFullStatus	T	F	F	F

Bit states at transition points of blended move by linear → circular move using follow contour velocity constrained or unconstrained

This lists the bit states at transition points of blended move by using follow contour velocity constrained or unconstrained.



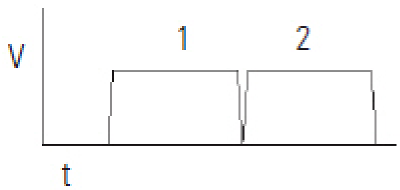
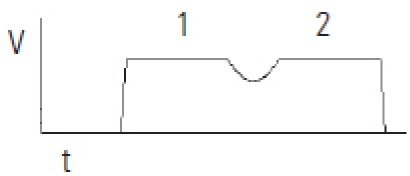
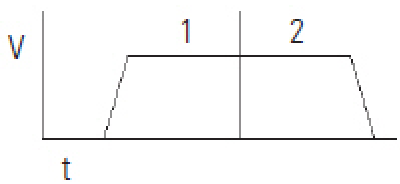
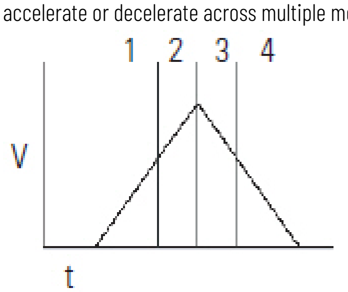
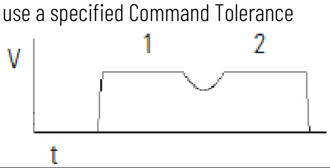
This table shows the bits status at the transition points.

Bit	TP1	TP2	TP3
Move1.DN	T	T	T
Move1.IP	T	F	F
Move1.AC	T	F	F
Move1.PC	F	T	T
Move2.DN	T	T	T
Move2.IP	T	T	F
Move2.AC	F	T	F
Move2.PC	F	F	T
cs1.MoveTransitionStatus	F	F	F
cs1.MovePendingStatus	T	F	F
cs1.MovePendingQueueFullStatus	T	F	F

Choose a Termination Type

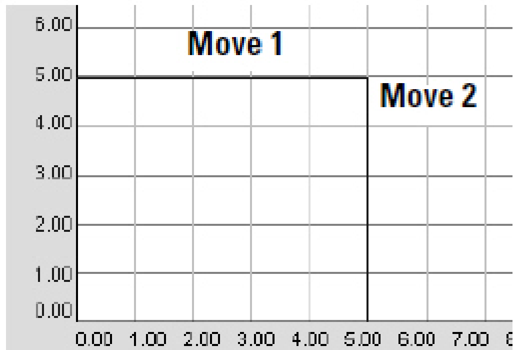
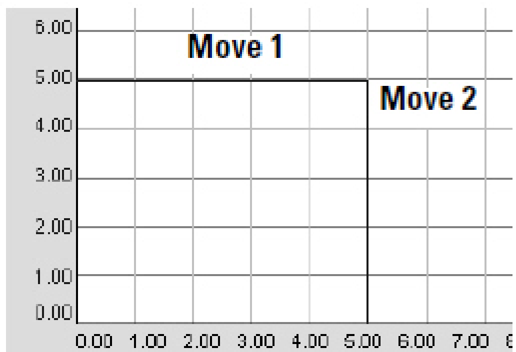
The termination type determines when the instruction is complete. It also determines how the instruction blends its path into the queued MCLM or MCCM instruction, if there is one.

To choose a termination type:

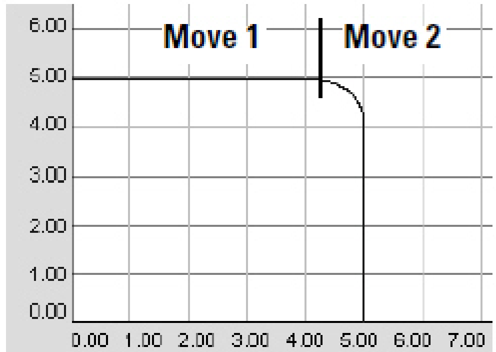
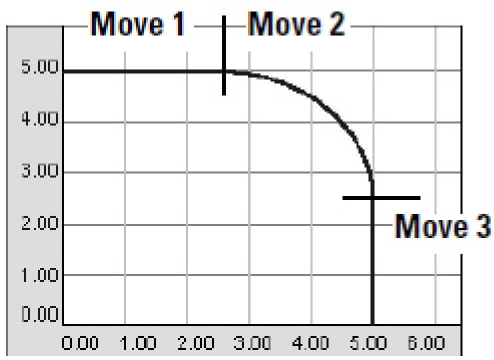
If you want the axes to (vector speeds)	And you want the instruction to complete when	Then use this Termination Type
stop between moves. 	The following occurs: • Command position equals target position. • The vector distance between the target and actual positions is less than or equal to the Actual Position Tolerance of the Coordinate System.	0 - Actual Tolerance
	The command position equals the target position.	1 - No Settle
keep the speed constant **except** between moves. 	The command position gets within the Command Position Tolerance of the coordinate system.	2 - Command Tolerance
	The axes get to the point at which they must decelerate at the deceleration rate.	3 - No Decel
transition into or out of a circle without stopping. 		4 - Follow Contour Velocity Constrained
accelerate or decelerate across multiple moves. 		5 - Follow Contour Velocity Unconstrained
use a specified Command Tolerance 	The command position gets within the Command Position Tolerance of the coordinate system.	6 - Command Tolerance Programmed

To make sure that this is the right choice for you:

- Review these tables.

Termination Type	Example Path	Description
0 - Actual Tolerance		The instruction stays active until both of these happen: • Command position equals target position. • The vector distance between the target and actual positions is less than or equal to the Actual Position Tolerance of the coordinate system. At that point, the instruction is complete and a queued MCLM or MCCM instruction can start. **Important:** Make sure that you set the Actual Tolerance to a value that your axes can reach. Otherwise the instruction stays in process.
1 - No Settle		The instruction stays active until the command position equals the target position. At that point, the instruction is complete and a queued MCLM or MCCM instruction can start.
2, 6 - Command Tolerance		The instruction stays active until the command position gets within the Command Tolerance of the Coordinate System. At that point, the instruction is complete and a queued MCLM or MCCM instruction can start. If you don't have a queued MCLM or MCCM instruction, the axes stop at the target position.

The Logix Designer application compares	To the	And uses the	For the
100% of the configured length of the first instruction using a Command Tolerance termination type	configured Command Tolerance for the Coordinate System	shorter of the two lengths	command Tolerance length used for the **first** instruction
100% of the configured length of the last move instruction using a Command Tolerance termination type	configured Command Tolerance for the Coordinate System	shorter of the two lengths	command Tolerance length used for the **next to last** instruction
50% of each of the lengths of all other move instructions	configured Command Tolerance for the Coordinate System	shorter of the two lengths	command Tolerance length used for **each individual** instruction

Termination Type	Example Path	Description
3 - No Decel		The instruction stays active until the axes get to the deceleration point. At that point, the instruction is complete and a queued MCLM or MCCM instruction can start. • The deceleration point depends on whether you use a trapezoidal or S-curve profile. • If you don't have a queued MCLM or MCCM instruction, the axes stop at the target position.
4 - Follow Contour Velocity Constrained		The instruction stays active until the axes get to the target position. At that point, the instruction is complete and a queued MCLM or MCCM instruction can start. • This termination type works best with tangential transitions. For example, use it to go from a line to a circle, a circle to a line, or a circle to a circle. • The axes follow the path. • The length of the move determines the maximum speed of the axes. If the moves are long enough, the axes will not decelerate between moves. If the moves are too short, the axes decelerate between moves.
5 - Follow Contour Velocity Unconstrained		This termination type is similar to the contour velocity constrained. It has these differences: • Use this termination type to get a triangular velocity profile across several moves. This reduces jerk. • To avoid position overshoot at the end of the last move, you must calculate the deceleration speed at each transition point during the deceleration-half of the profile. • You must also calculate the starting speed for each move in the deceleration half of the profile.

Important Considerations

If you stop a move (that is, using an MCS or by changing the speed to zero with an MCCD) during a blend and then resume the move (that is, by reprogramming the move or by using another MCCD), it will deviate from the path that you would have seen if the move had not been stopped and resumed. The same phenomenon can occur if the move is within the decel point of the start of the blend. In either case, the deviation will most likely be a slight deviation.

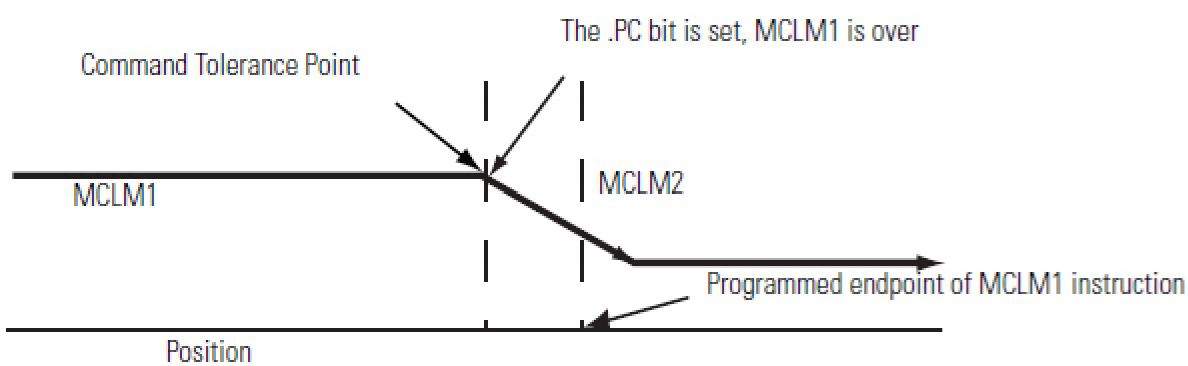
Velocity Profiles for Collinear Moves

Collinear moves are those that lie on the same line in space. Their direction can be the same or opposite. The velocity profiles for collinear moves can be complex. This section provides you with examples and illustrations to help you understand the velocity profiles for collinear moves programmed with MCLM instructions.

Velocity Profiles for Collinear Moves with Termination Type 2 or 6

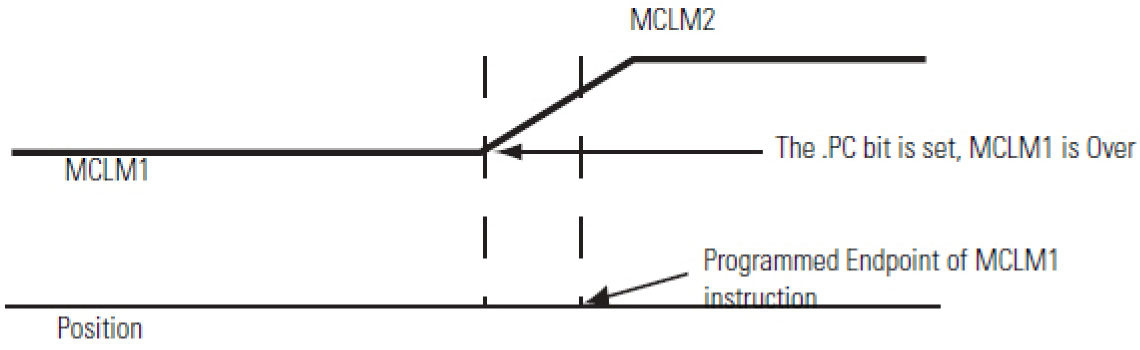
This illustration shows the velocity profile of two collinear moves using a Command Tolerance (2) termination type. The second MCLM instruction has a **lower** velocity than the first MCLM instruction. When the first MCLM instruction reaches its Command Tolerance point, the move is over and the .PC bit is set.

Velocity Profile of Two Collinear Moves When the Second Move has a Lower Velocity than the First Move and Termination Type 2 or 6 is Used



This illustration shows the velocity profile of two collinear moves using a Command Tolerance (2) termination type. The second MCLM instruction has a **higher** velocity than the first MCLM instruction. When the first MCLM instruction reaches its Command Tolerance point, the move is over and the .PC bit is set.

Velocity Profile of Two Collinear Moves When the Second Move has a Higher Velocity than the First Move and Termination Type 2 or 6 is Used



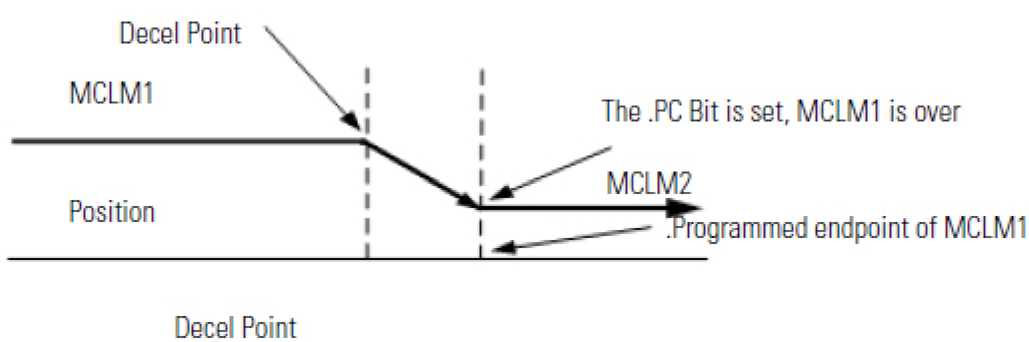
Velocity Profiles for Collinear Moves with Termination Types 3, 4, or 5

This illustration shows a velocity profile of two collinear moves. The second MCLM instruction has a **lower** velocity than the first MCLM instruction and one of these termination types are used:

- No Decel (3)
- Follow Contour Velocity Constrained (4)
- Follow Contour Velocity Unconstrained (5)

When the first MCLM instruction reaches the deceleration point, it decelerates to the programmed velocity of the second move. The first move is over and the .PC bit is set.

Velocity Profile of Two Collinear Moves When the Second Move has a Lower Velocity than the First Move and Termination Type 3, 4, or 5 is Used

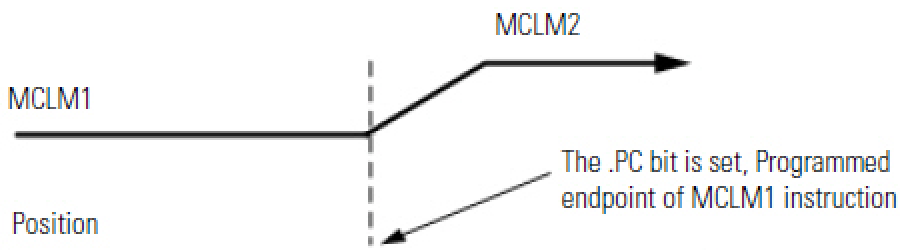


This illustration shows a velocity profile of two collinear moves. The second MCLM instruction has a **higher** velocity than the first MCLM instruction and one of these termination types are used:

- No Decel (3)
- Follow Contour Velocity Constrained (4)
- Follow Contour Velocity Unconstrained (5)

The .PC bit is set when the first move reaches its programmed endpoint.

Velocity Profile of Two Collinear Moves When the Second Move has a Higher Velocity than the First Move and Termination Type 3, 4, or 5 is Used



Symmetric Profiles

Profile paths are symmetric for all motion profiles.

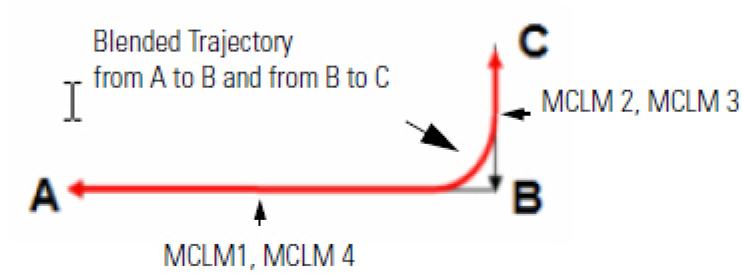
Programming the velocity, acceleration, and deceleration values symmetrically in the forward and reverse directions generates the same path from point A to point C in the forward direction, as from point C to point A in the reverse direction.

While this concept is most easily shown in a two-instruction sequence, it applies to instruction sequences of any length provided that they are programmed symmetrically.

Refer to this Example of a Symmetric Profile for more details.

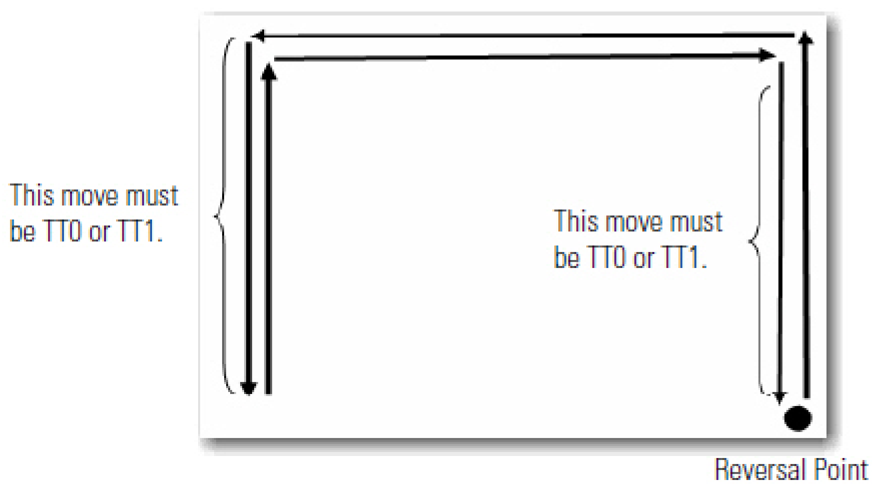
- MCLM 1 (point A to point B) is followed by MCLM 2 (point B to point C).
- MCLM 3 (point C to point B) is followed by MCLM 4 (point B to point A).
- The acceleration of MCLM 1 must be equal to the deceleration of MCLM 4.
- The deceleration of MCLM 1 must be equal to the acceleration a MCLM 4.
- The acceleration of MCLM 2 must be equal to the deceleration of MCLM 3.
- The deceleration of MCLM 2 must be equal to the acceleration of MCLM 3.

MCLM 1 (Pos = [2,0], Accel = 1, Decel = 2)
MCLM 2 (Pos = [2,1], Accel = 3, Decel = 4)
MCLM 3 (Pos = [2,0], Accel = 4, Decel = 3)
MCLM 4 (Pos = [0,0], Accel = 2, Decel = 1)



IMPORTANT We recommend that you terminate any sequence of moves by either Termination Type 0 or 1, that is, TT0 or TT1.

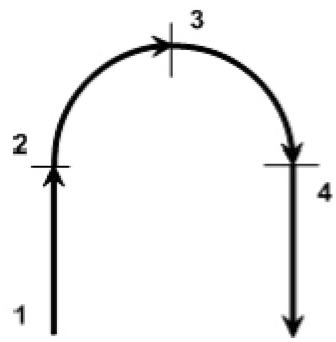
To guarantee that your trajectory is symmetric, you must terminate any sequence of moves by either Termination Types 0 or 1. You should also use a Termination Type of 0 or 1 at the Reversal Point of a profile that moves back on itself.



Using a TT2, TT3, TT4, TT5, ot TT6 as the last move in a profile (or the reversal point) is safe. However, the resulting trajectory from A to B may not always be the same as that from B to A. Explicit termination of the sequence of moves helps the controller to optimize the velocity profile, reduce the CPU load, and guarantee a symmetric profile.

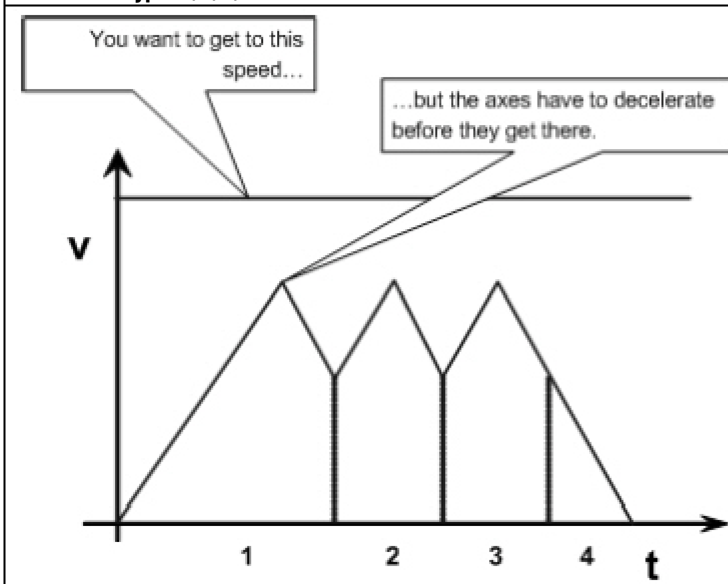
How To Get a Triangular Velocity Profile

If you want to program a pick and place action in four moves, minimize the Jerk rate, and use a triangular velocity profile.



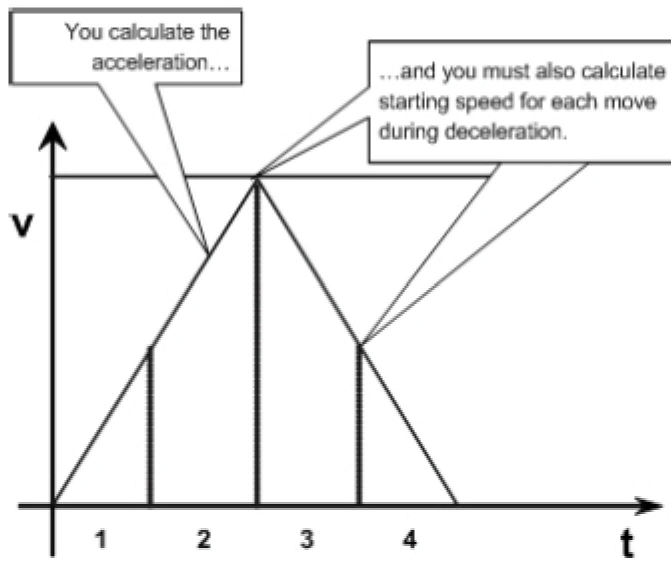
Then, use termination type 5. The other termination types may not let you get to the speed you want.

Termination Types 2, 3, 4, or 6



The length of each move determines its maximum speed. As a result, the axes will not reach a speed that causes them to overshoot the target position during deceleration.

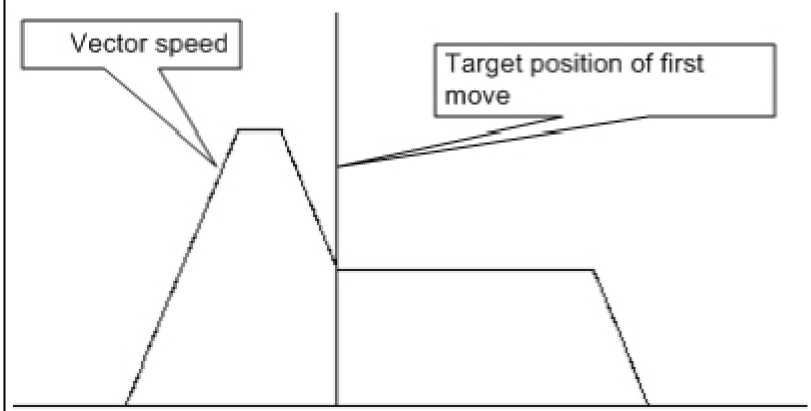
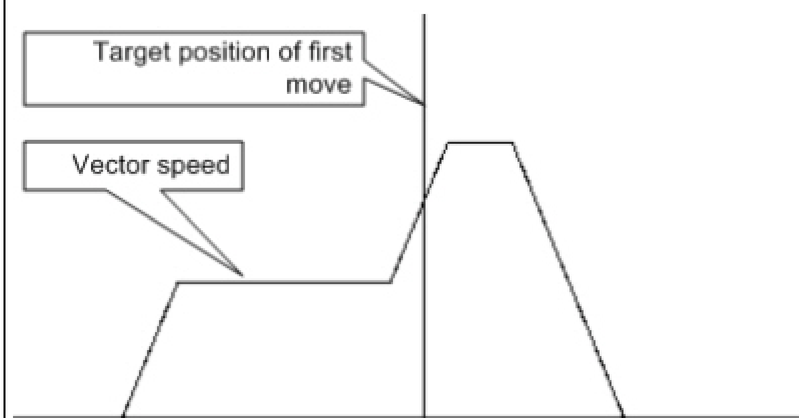
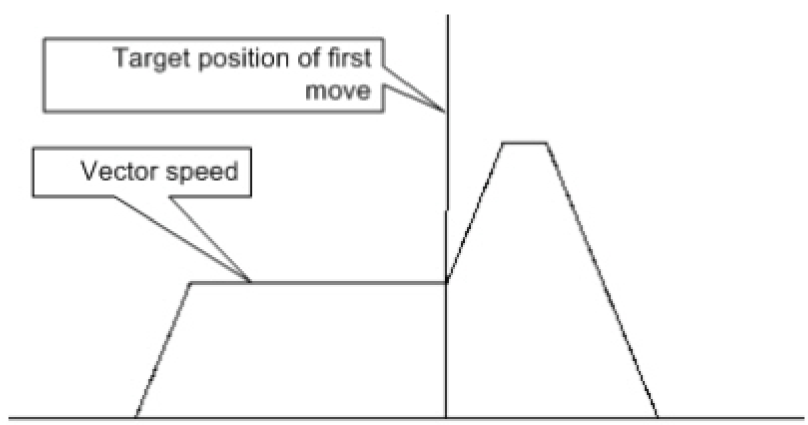
Termination Type 5



The axes accelerate to the speed that you want. You must calculate the starting speed for each move in the deceleration-half of the profile.

Blending Moves at Different Speeds

You can blend MCLM and MCCM instructions where the vector speed of the second instruction is different from the vector speed of the first instruction.

If the next move is	And the Termination Type of the first move is	Then
Slower	2 - Command Tolerance 3 - No Decel 4 - Contour Velocity Constrained 5 - Contour Velocity Unconstrained 6 - Command Tolerance Programmed	
Faster	2 - Command Tolerance 3 - No Decel 6 - Command Tolerance Programmed	
	4 - Contour Velocity Constrained 5 - Contour Velocity Unconstrained	

Geometries with no orientation support

Use these guidelines to configure the 3-axis robot geometries with no orientation support in Logix Designer application. These robot geometries include:

- Articulate Independent robot
- Articulate Dependent robot
- Delta Three-dimensional robot
- Delta Two-dimensional robot
- SCARA Delta robot
- SCARA Independent robot
- Cartesian Gantry robot
- Cartesian H-bot robot

The **Coordinate Definition** parameter in the **Coordinate System Properties** dialog box determines whether or not there is orientation support in the coordinate system.

See also

Configure a Cartesian Coordinate System on page 39

Configure Articulated Independent robots

Follow these guidelines for configuring articulated independent robots:

- Articulated independent J1J2J3 robots
- Articulated independent J1J2J3J4J5J6 robots

WARNING: Before turning ON the Transform and/or establishing the reference frame, be sure to do the following for the joints of the target coordinate system.

- Set and enable the soft travel limits.
- Enable the hard travel limits.

Failure to do this can allow the robot to move outside of the work envelope causing machine damage and/or serious injury or death to personnel.

See also

Configure an Articulated Independent J1J2J3 robot on page 66

Configure an Articulated Independent J1J2J3J4J5J6 robot on page 74

Configure an Articulated Independent J1J2J3 robot

This section describes the reference frame, work envelope, and configuration parameters for Articulated Independent J1J2J3 robots.

WARNING: Before turning ON the Transform and/or establishing the reference frame, be sure to do the following for the joints of the target coordinate system.

- Set and enable the soft travel limits.
- Enable the hard travel limits.

Failure to do this can allow the robot to move outside of the work envelope causing machine damage and/or serious injury or death to personnel.

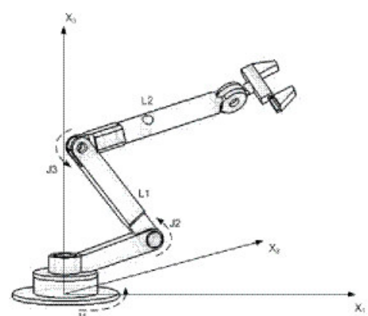
Establish the reference frame for articulated independent J1J2J3 robots

The reference frame is the Cartesian coordinate frame that defines the origin and the three primary axes (X1, X2, and X3). These axes measure the real Cartesian positions.

WARNING: Failure to properly establish the correct reference frame for your robot can cause the robotic arm to move to unexpected positions causing machine damage and/or injury or death to personnel.

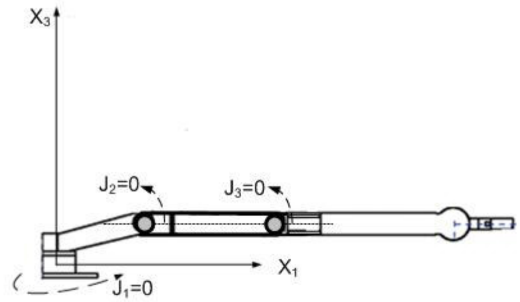
The reference frame for an Articulated Independent J1J2J3 robot is located at the base of the robot as shown in this figure.

Illustration 1



Before establishing the Joint-to-Cartesian reference frame relationship, it is important to know some information about the Kinematic mathematical equations used in the Logix controllers. The equations are written as if the robot joints are positioned as shown in the following illustration.

Illustration 2 - Side view

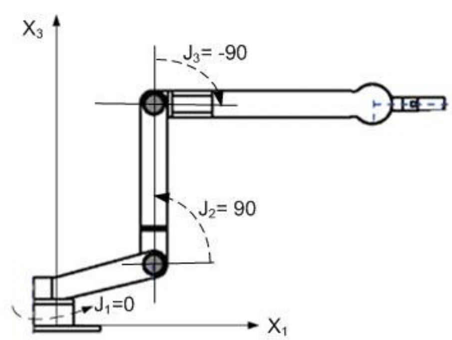


- +J1 is measured counterclockwise around the +X3 axis starting at an angle of J1=0 when L1 and L2 are both in the X1-X2 plane.
- +J2 is measured counterclockwise starting with J2=0 when L1 is parallel to X1-X2 plane.
- +J3 is measured counterclockwise with J3=0 when L2 is aligned with link L1.

When the robot is physically in this position, the Logix Designer application Actual Position tags for the axes must be:

- J1 = 0.
- J2 = 0.
- J3 = 0.

Illustration 3 - Side view



When the robot is physically in the above position, the Logix Designer application Actual Position tags for the axes must be:

- J1 = 0.
- J2 = 90.
- J3 = -90.

If the physical position and joint angle values of the robot cannot match those shown in the preceding illustrations, use one of the Alternate Methods for Establishing the Joint-to-Cartesian reference frame relationship.

See also

[Methods for establishing a reference frame for an articulated independent J1J2J4 robot] on [page 68]

Methods to establish a reference frame for an articulated independent J1J2J3 robot

Use these methods to establish a reference frame for the robot.

For each:	Use one of these methods to establish the reference frame:
Incremental axis	Each time the power for the robot is cycled.
Absolute axis	Only to establish absolute home.

- Method 1 - Establishes a Zero Angle Orientation and allows the configured travel limits and home position on the joint axes to remain operational. Use this method when operating the axes between the travel limits determined prior to programming a Motion Redefine Position (MRP) instruction and want these travel limits to stay operational.
- Method 2 - Uses an MRP instruction to redefine the axes position to align with the joint reference frame. This method may require the soft travel limits to be adjusted to the new reference frame.

See also

[Method 1 - Establish a reference frame] on [page 68]

[Method 2 for an absolute axis] on [page 69]

Method 1 - Establish a reference frame using zero angle orientation

Each axis for the robot has the mechanical hard stop in each of the positive and negative directions. Manually move or press each axes of the robot against its associated mechanical hard stop and redefine it to the hard limit actual position provided by the robot manufacturer. J1 is the axis at the base of the robot that rotates around X3.

When the robot is moved so that Link1 is parallel to the X3 axis and Link2 is parallel to X1 axis, the values for the Actual Position tags for the axes in the Logix Designer application should be:

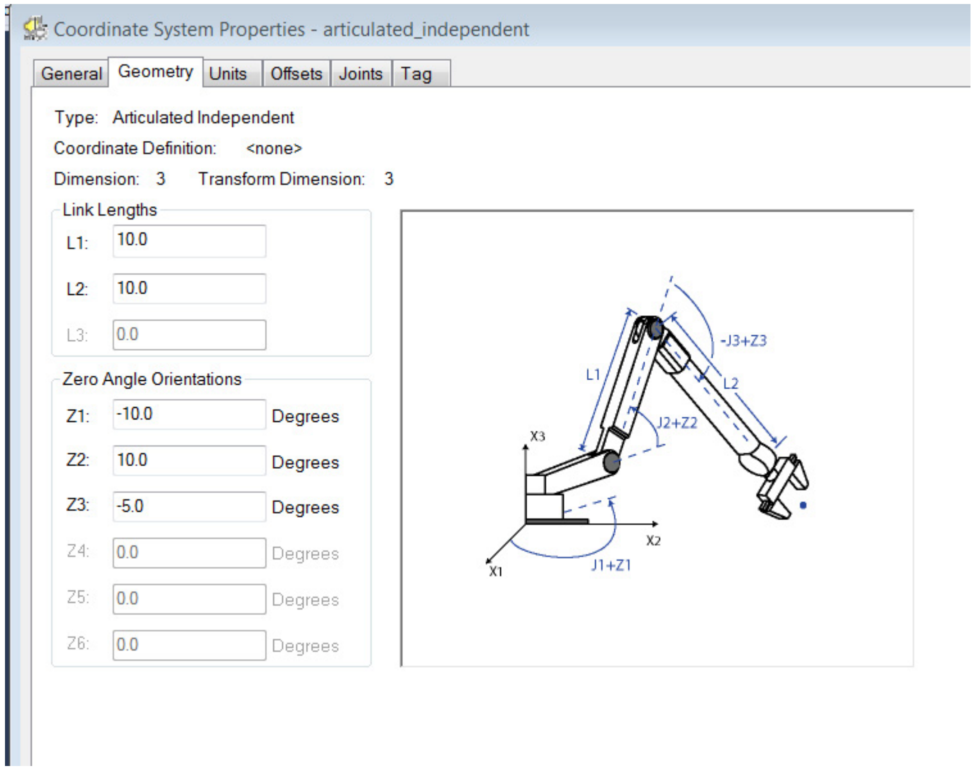
- J1 = 0
- J2 = 90°
- J3 = 0°

If the Actual Position tags do not show these values, configure the **Zero Angle Orientation** parameters in the **Coordinate System Properties** dialog box for the joint or joints that do not correspond.

If the Logix Designer application read-out values are:	Set the Zero Angle Orientations on the Coordinate System Properties dialog box to:

J1 = 10 J2 = 80 J3 = 5	Z1 = -10 Z2 = 10 Z3 = -5

The Joint-to-Cartesian reference frame relationship is automatically established by the Logix controller after the Joint coordinate system parameters (link lengths, base offsets, and end effector offsets) are configured and the MCT instruction is enabled.



See also

Methods to establish a reference frame on page 68

Method 2 - Establish a reference frame using a MRP instruction

Position the robot so that:

- L1 is parallel to the X3 axis.
- L2 is parallel to X1 axis.

Program a Motion Redefine Position (MRP) instruction for all three axes with the following values:

- J1 = 0
- J2 = 90°
- J3 = -90°

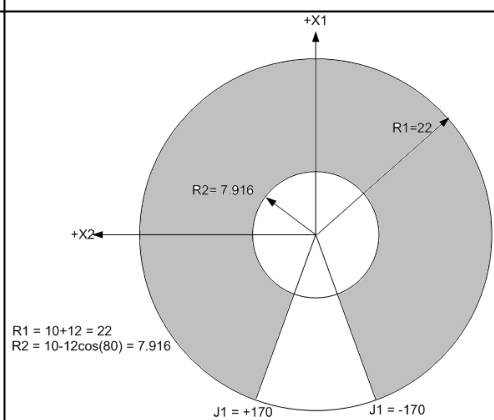
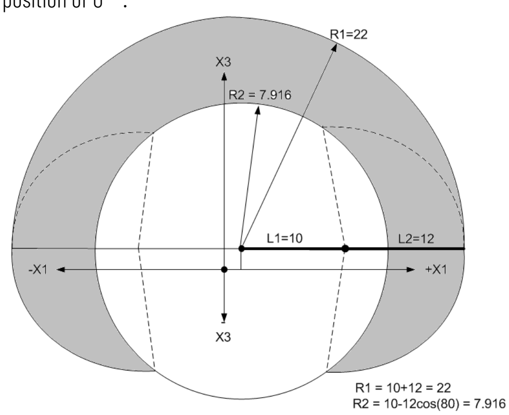
The Joint-to-Cartesian reference frame relationship is automatically established by the Logix controller after the Joint coordinate system parameters, which are the link lengths, base offsets, and end-effector offsets, are configured and the MCT instruction is enabled.

See also

[Method 1 - Establish a reference frame using zero angle orientation](#) on [page 68](#)

Work envelope for Articulated Independent J1J2J3 robots

The work envelope is the three-dimensional region of space that defines the reaching boundaries for the robot arm. The work envelope for an articulated robot is ideally a complete sphere with an inner radius equal to L1- L2 and outer radius equal to L1+L2. Due to the range of motion limitations on individual joints, the work envelope may not be a complete sphere.

If the range-of-motion values for the articulated robot are:	Typically, the work envelope is:
J1 = ± 170 J2 = 0 to 180 J3 = ± 60 L1 = 10 L2 = 12	 Top view - Depicts the envelope of the tool center point sweep in J1 and J3 while J2 remains at a fixed position of 0°.  Side view - Depicts the envelope of the tool center point sweep in J2 and J3 while J1 remains at a fixed position of 0°.

See also

[Configuration parameters for articulated independent robot](#) on [page 71](#)

[Configure an articulated independent robot](#) on [page 65](#)

Configuration parameters for Articulated Independent J1J2J3 robots

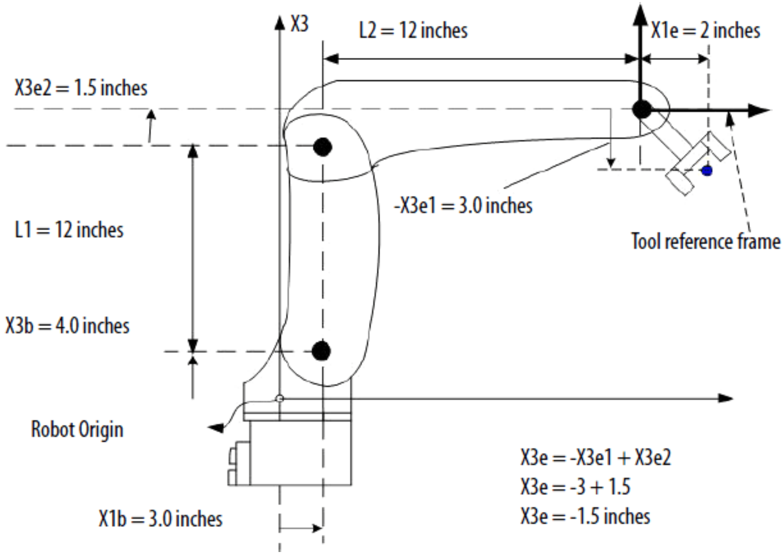
Configure the Logix Designer application to control robots with varying reach and payload capacities. The configuration parameter values for the robot include:

- Link lengths
- Base offset
- End effector offsets

The configuration parameter information is available from the robot manufacturer.

IMPORTANT	Verify that the values for the Link Lengths, Base Offsets, and End-Effector Offsets are entered in the Coordinate System Properties dialog box using the same measurement units.

This example illustrates the typical configuration parameters for an Articulated Independent J1J2J3 robot.



If the robot is two-dimensional, then X3b and X3e are X2b and X2e.

See also

Link lengths for Articulated Independent robots on page 71

Base offsets for Articulated Independent robots on page 72

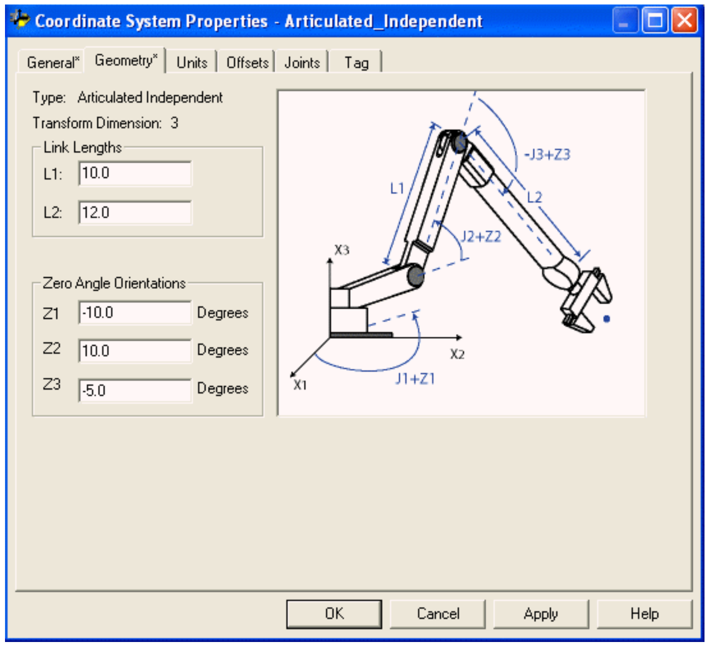
End effector offsets for Articulated Independent robots on page 73

Link lengths for Articulated Independent J1J2J3 robots

Link lengths are the rigid mechanical bodies attached at joints.

For an articulated independent robot with	The length of	Is equal to the value of the distance between
2 dimensions	L1 L2	J1 and J2 J2 and the end-effector
3 dimensions	L1 L2	J2 and J3 J3 and the end-effector

Enter the link lengths on the **Geometry** tab in the **Coordinate System Properties** dialog box.



See also

Base offset for Articulated Independent robots on page 72

End effector offsets for Articulated Independent robots on page 73

Configuration parameters for Articulated Independent robots on page 71

Base Offsets for Articulated Independent J1J2J3 robots

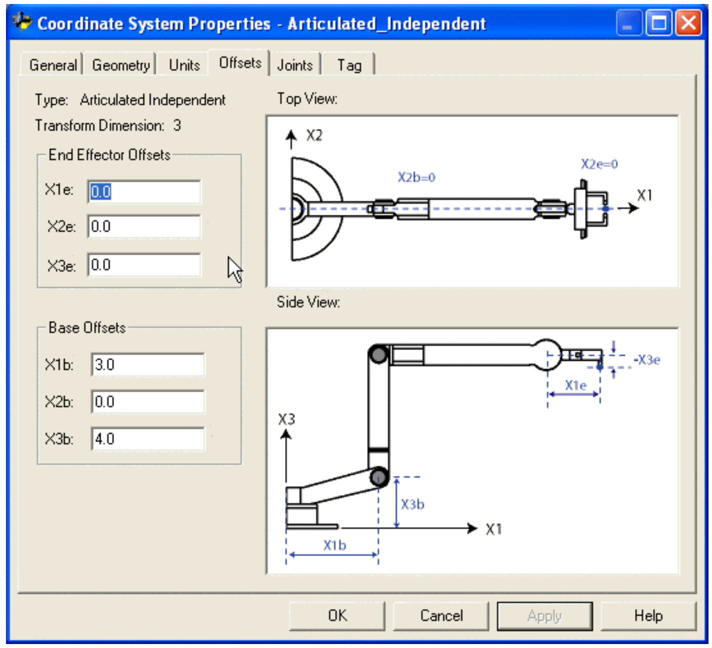
The Base Offset is a set of coordinate values that redefines the origin of the robot. The correct base offset values are typically available from the robot manufacturer. Type the values for the base offsets in the **X1b** and **X3b** boxes of the **Coordinate System Properties** dialog box.

This example shows the Offsets tab for an Articulated Independent J1J2J3 robot.

Type the Base Offset values.

For the robot shown in our example, the Base Offset values are:

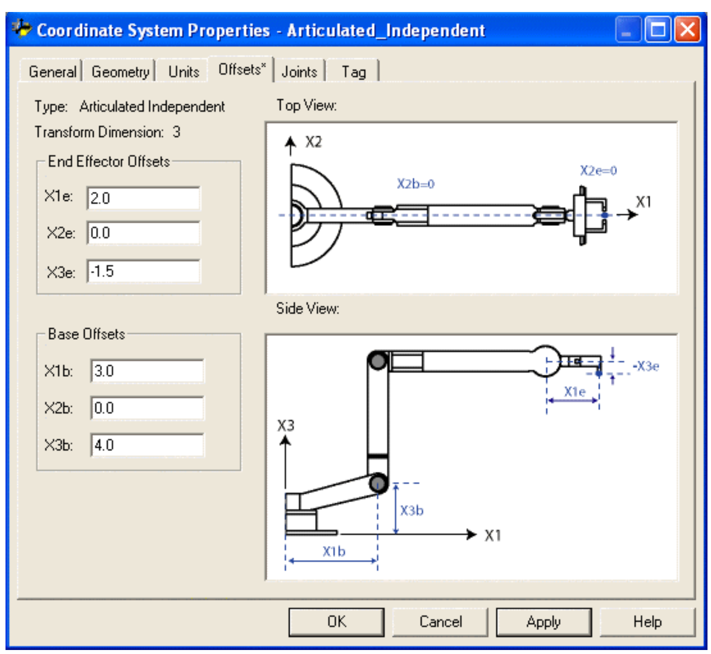
- X1b = 3.0
- X3b = 4.0



End-Effector Offsets for Articulated Independent J1J2J3 robots

The robot can have an end effector attached to the end of robot link L2. If there is an attached end effector, configure the **End-Effector Offset** value on the **Offsets** tab in the **Coordinate System Properties** dialog box. The **End-Effector Offsets** are defined with respect to the tool reference frame at the tool tip.

Some robots also have an offset defined for the J3 joint. Account for this value when computing the **X3e** end effector offset value. If the value for **X3e** offset is entered as the sum of **X3e1+X3e2 (-3+1.5 = -1.5)**, the configured value for **X3e** is **-1.5**.



See also

Configuration parameters for Articulated Independent robots on page 71

Link Lengths for Articulated Independent robots on page 71

Base Offsets for Articulated Independent robots on page 72

Error conditions

Kinematics error conditions are detected:

- Upon activation of a transformation by executing an MCT instruction.
- In some movement conditions.

  Errors can occur for certain movement conditions for either the source or target coordinate system after a transformation has been established. These types of errors are reported in the MCT instruction error codes. Singularity and other movement error conditions are also reported in the MCT error codes.
- Computing an invalid position via an MCTP instruction.

For a list and description of error codes, see Logix5000 Controllers Motion Instructions Reference Manual, publication MOTION-RM002.

Configure an Articulated Independent J1J2J3J4J5J6 robot

Articulated Independent J1J2J3J4J5J6 robots have six revolute joints that allow six degrees of freedom for the end position or end of arm movement.

WARNING: Before turning ON the Transform and/or establishing the reference frame, be sure to do the following for the joints of the target coordinate system.

- Set and enable the soft travel limits.
- Enable the hard travel limits.

Failure to do this can allow the robot to move outside of the work envelope causing machine damage and/or serious injury or death to personnel.

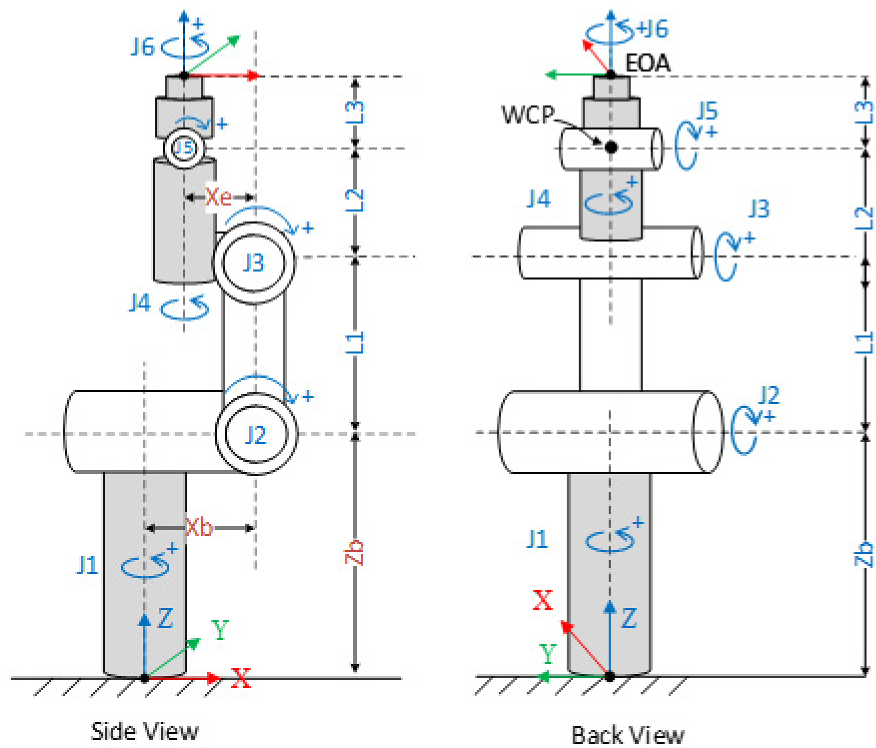
Articulated Independent J1J2J3J4J5J6 robot geometry

The Articulated Independent J1J2J3J4J5J6 robot geometry has six revolute joints that allow six degrees of freedom for the end position or end of arm movement.

Configure the robot geometry by using Link 1 (L1), Link 2 (L2), and Link 3 (L3) link lengths, and X-axis Base Offset (Xb), Z-axis Base Offset (Zb), and X-axis End Effector Offset (Xe). All offset directions and signs coincide with the robot-base frame direction.

Tip: Refer to Robot Joint Direction Sense for information on configuring joint direction senses other than the default settings.

This illustration shows an Articulated Independent J1J2J3J4J5J6 geometry.



Keep these guidelines in mind when configuring Articulated Independent J1J2J3J4J5J6 robots:

- In the Logix Designer application, the six degrees of freedom is configured as six joint axes (J1, J2, J3, J4, J5, J6) in the Articulated Independent J1J2J3J4J5J6 robot's coordinate system. The six joint axes are either:
  - Directly programmed in joint space and controlled by using Motion Axis Move (MAM) instructions.
  - Automatically controlled by using the Logix Designer application Kinematics instructions, programmed in a Cartesian coordinate system.
- In this geometry:
  - Joint J1 produces rotational motion around the Z-axis of the base frame.
  - Joint J2 produces motion to move the robot's lower arm (Link L1) in forward or backward direction. Joint J2 is known as shoulder of the robot.
  - Joint J3 produces motion to raise or lower the upper arm (Link L2) of the robot. Joint J3 is known as elbow of the robot.
  - Joint J4 produces motion to roll the upper arm (Link L2) of the robot.
  - Joint J5 produces motion to raise or lower the link L3. Joint J5 is known as the wrist of the robot.

- Joint J6 produces rotation motion at End Of Arm (EOA).
- Axis of rotations for the last three Joints J4, J5, and J6 intersect at a single reference point. This reference point is called the Wrist Center Point (WCP).
- End Of Arm (EOA) position is represented by the Cartesian coordinate system.

See also

Configuration types for Articulated Independent J1J2J3J4J5J6 robots on page 78

Configuration parameters for Articulated Independent J1J2J3J4J5J6 robots on page 85

Reference frame for Articulated Independent J1J2J3J4J5J6 robots

The base frame, also called the Robot reference XYZ frame, for the Articulated Independent geometry is at the base of the robot. Robot geometry target points refer to this base frame. Translating from the Cartesian base frame to the robot system at the end of arm (EOA) and vice versa creates transformations for this geometry. For the transformations to work correctly, establish the origins for the axes in the joint space with respect to the robot base Cartesian frame.

WARNING: Failure to properly establish the correct reference frame for the robot can cause the robotic arm to move to unexpected positions causing machine damage, or injury or death to personnel.

Base frame

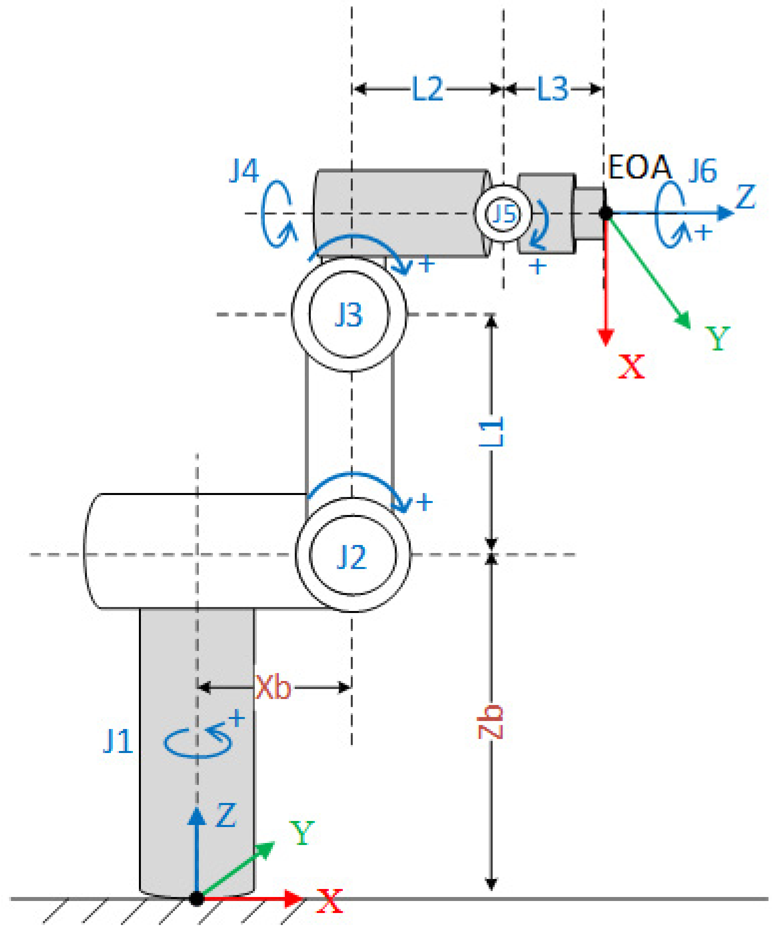
The reference XYZ frame, or base frame, for an Articulated Independent geometry is located at the center of the base plate that connects to Joint J1. When you configure an Articulated Independent coordinate system in the Logix Designer application:

- Joint J3 is homed at 90 degrees.
- All other joints are homed to 0.
- In the XZ plane of the Robot Base frame, the robot arm is aligned along the positive x-axis.

End of Arm frame

The End of Arm (EOA) frame is set at the end of the robot end effector. The EOA frame is aligned independent of the base frame. The orientation axes J4, J5, and J6 control the EOA frame. In its natural orientation, the EOA is aligned with the base frame. The XYZ for the EOA and the base XYZ frame have the same sense of direction. In the homed calibration position, the positive Z-axis of the end effector and the positive X of the base frame are aligned.

This side view of the robot shows the homing and arm alignment for the base frame and the EOA frame.



See also

[Work envelope for Articulated Independent J1J2J3J4J5J6 robots on page 94]

[Maximum joint limits for Articulated Dependent J1J2J3J4J5J6 robots on page 95]

Commission an Articulated Independent J1J2J3J4J5J6 robot

Follow these steps to commission an Articulated Independent J1J2J3J4J5J6 robot.

To commission an Articulated Dependent J1J2J3J4J5J6 robot

1. Get the angle values from the robot manufacturer for joints J1, J2, J3, J4, J5, and J6 at the calibration position. Use these values to establish the

zero, or reference, position. Refer to Reference frame for Articulated Independent J1J2J3J4J5J6 robots on page 76 for a description of the reference position.

2. Refer to the manufacturer's data sheet to determine if the associated sized motor contains an internal or external gearbox from the motor to actuation, at the links or joints, to move the robot.
3. Open the **Axis Properties** and select the **Scaling** tab.
   a. In **Controller Organizer**, expand the **Motion Groups** folder, right-click the axis and then select **Properties**.
   b. In **Axis Properties**, in **Categories**, select **Scaling**.
4. In **Transmission Ratio I/O**, set the gear ratio for each axis.
5. In **Scaling**, enter the scaling to apply to all axes so that one revolution equals 360°.
6. Move all joints to the zero position by jogging the robot under programmed control, or manually moving the robot when the joint axes are in an open-loop state.
7. Do one of these steps to set zero positions for the axes:
   - Use the Motion Redefine Position (MRP) instruction to set the positions of the joint axes to the calibration values obtained in step 1.
   - Set the configuration value for the joint axes home position to the calibration values obtained in step 1 and execute a Motion Axis Home (MAH) instruction for each joint axis.
8. Move each joint to an absolute position of 0.0. Verify that each joint position reads 0°.

   If joint position values do not read 0°, configure the values for the zero angle offsets to be equal to the values of the joints when in a horizontal position. See [insert link to zero angle offsets topic] for instructions for setting the offsets.

   Tip: The robot axes are absolute, so you probably will establish the zero positions only once. Re-establish the zero positions if you change the controller or lose them.

See also

Configuration types for Articulated Independent J1J2J3J4J5J6 robots on page 78

Maximum joint limits for Articulated Dependent J1J2J3J4J5J6 robots on page 95

Configuration types for Articulated Independent J1J2J3J4J5J6 robots

Articulated Independent J1J2J3J4J5J6 robots support three configuration types:

- Arm, or shoulder
- Elbow

- Wrist

Each configuration has two possible values and one Singularity Position condition.

IMPORTANT	Avoid passing through the singularity when performing moves in the Cartesian coordinate system. Moves that pass through the singularity can result in loss of control of kinematics.

See also

Arm configuration for Articulated Independent J1J2J3J4J5J6 robots on page 79

Elbow configuration for Articulated Independent J1J2J3J4J5J6 robots on page 80

Wrist configuration for Articulated Independent J1J2J3J4J5J6 robots on page 81

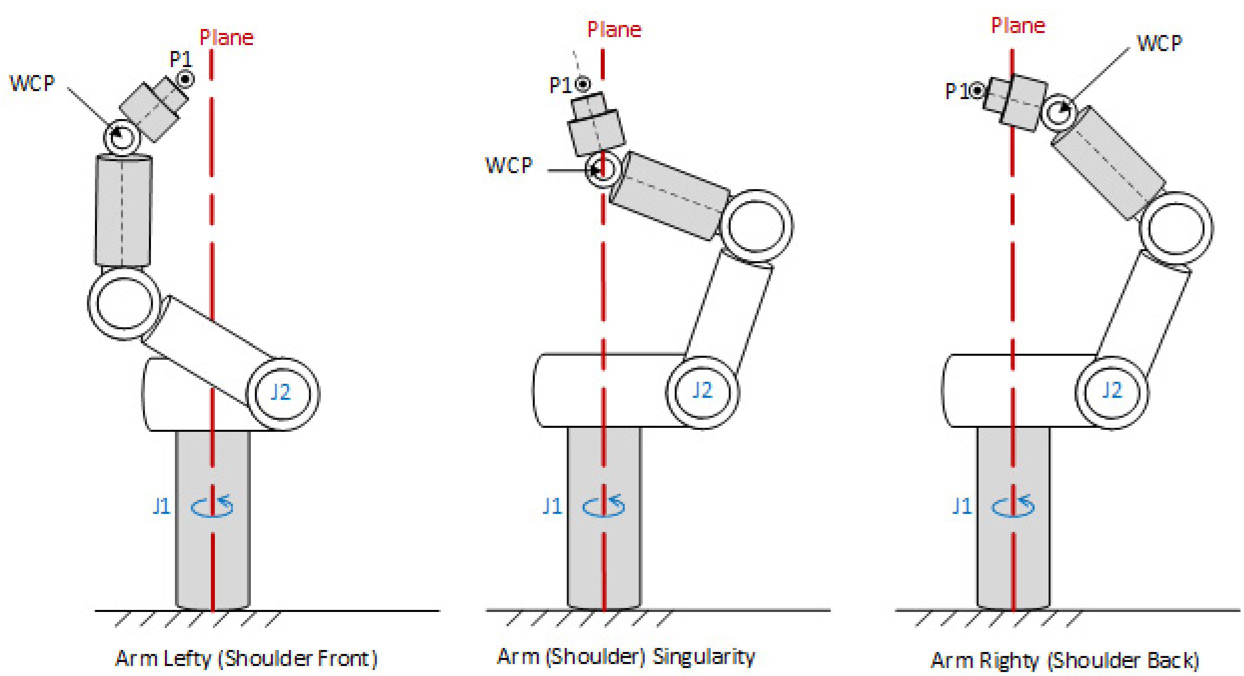
Configuration examples on page 82

Singularity error conditions on page 85

Arm configuration for Articulated Independent J1J2J3J4J5J6 robots

The arm, or shoulder, configuration is determined by the position of the Wrist Center Point (WCP) of the robot with reference to the plane passing through the axis of joint J1 and parallel to the axis of joint J2.

This illustration shows the arm configuration. The plane in this illustration is perpendicular to the line of sight and is represented as a dotted line. An End of Arm (EOA) is at the same Cartesian position P1, and is reached with lefty, righty, and singularity arm configurations.



- A WCP in front of the plane is a Lefty (Front) Arm configuration.
- A WCP behind the plane is a Righty (Rear) Arm configuration.
- A WCP lying in the plane is an Arm singularity condition.
- If the Logix Designer application calculates the forward transform on the joints when an arm is at singularity conditions, then the transform sets the default arm configuration as Lefty.

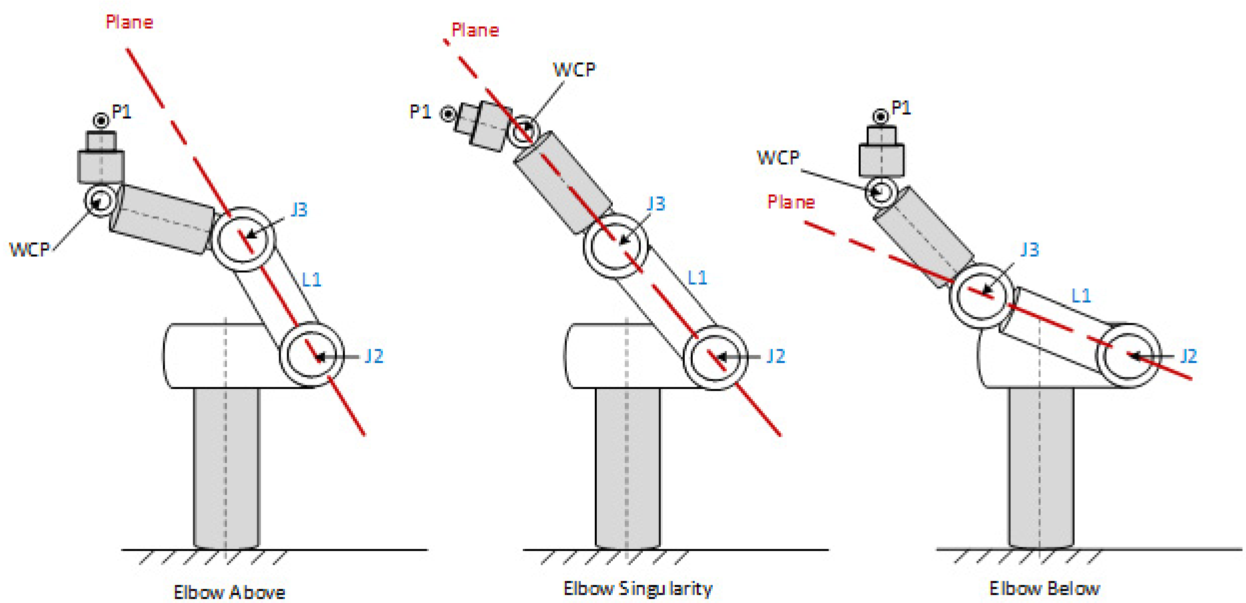
See also

Configuration examples on page 82

Elbow configuration for Articulated Independent J1J2J3J4J5J6 robots

The elbow configuration is determined by the position of the Wrist Center Point (WCP) of the robot with reference to the plane passing through the center line of Link L1 between joints J2 and J3.

This illustration shows the elbow configuration. The plane in this illustration is perpendicular to the line of sight and is represented as a dotted line. An End of Arm (EOA) is at the same Cartesian position P1 and is reached with above, below, and singularity elbow configurations.



- A WCP at the front of the plane is an Above Elbow configuration.
- A WCP at the back of the plane is a Below Elbow configuration.
- A WCP lying in the plane is an Elbow singularity condition.
- If the Logix Designer application calculates the forward transform on the joints when an elbow is at singularity conditions, then the transform sets the default elbow configuration as Above.

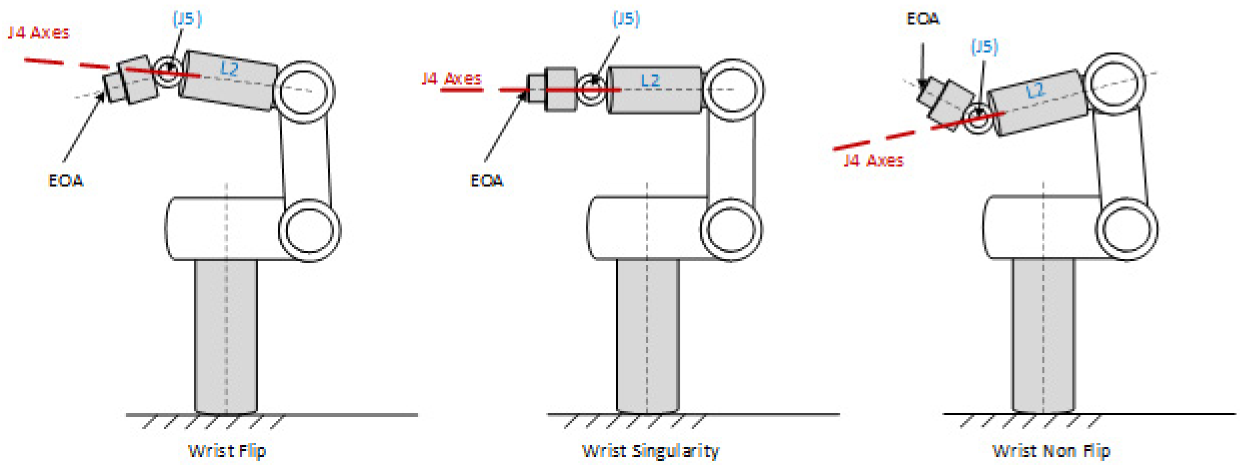
See also

Configuration examples on page 82

Wrist configuration for Articulated Independent J1J2J3J4J5J6 robots

The wrist configuration is determined by the position of the End Of Arm (EOA) of the robot with reference to the center line passing through link L2 (the J4 axes). The joint J5 is assumed to be the wrist joint, so the positive or negative sign of joint J5 determines the wrist configuration.

This illustration shows the wrist configuration. An EOA is at the same Cartesian position and is reached with flip, non-flip, and singularity wrist configurations.



- An EOA above the centerline of link L2 is considered a Flip configuration. In this case, J5 is negative.
- An EOA below the centerline of link L2 is considered a Non-Flip configuration. In this case, J5 is positive.
- Wrist Singularity occurs when the axes of joints J4 and J6 become coincident. At this position Joint J5 is 0°.
- If the forward transform is calculated on the joints when a wrist is at singularity conditions, then the transform sets the default wrist configuration as Non-Flip.

See also

Configuration examples on page 82

Robot configuration in an MCTPO instruction

In the Motion Calculate Transform Position with Orientation (MCTPO) instruction, a robot configuration is either an input or output parameter, depending on the transform direction.

Tip: Bit 0 of the robot configuration is ignored for the MCTPO instruction. Bits 4 through 31 are always 0.

- If the MCTPO Transform direction is set to Forward, the instruction computes the robot configuration and updates the tag data.
- If the MCTPO Transform direction is set to Inverse, the instruction requires the user to provide the robot configuration as an input tag.

The robot configuration is stored in a tag with a DINT data type. This table lists the definition of the tag.

Bit position	Description
31 - 4	0
3	Flip (1) / No Flip (0)
2	Above (1) / Below (0)
1	Lefty (1) / Righty (0)
0	Change (1) / Same (0)

Configuration examples

These examples illustrate how to use the robot configuration parameter in both the Forward Transform calculation and in the Inverse Transform calculation.

This table shows eight joint solutions for a specific Cartesian position.

Cartesian position:

X	Y	Z	Rx	Ry	Rz
49.14021	13.19838	430.4179	11.42836	-4.36741	152.4557

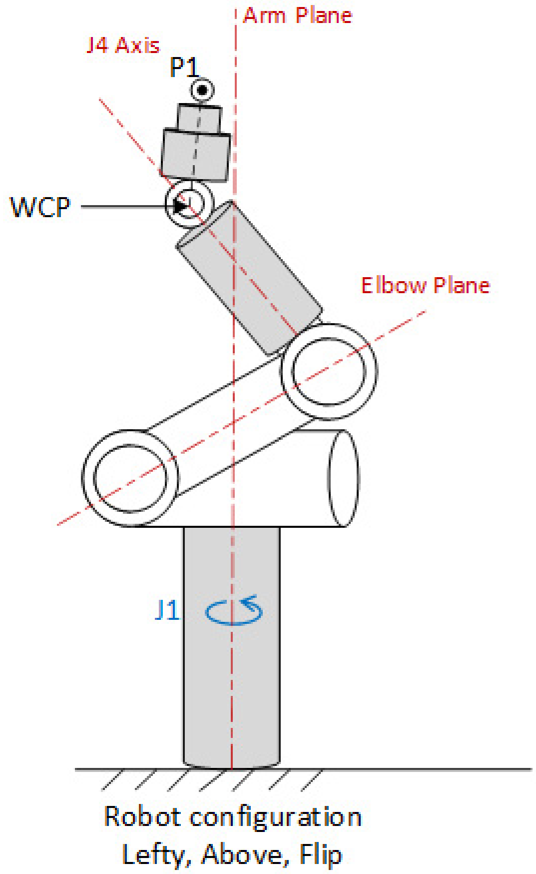
Joint and robot configuration:

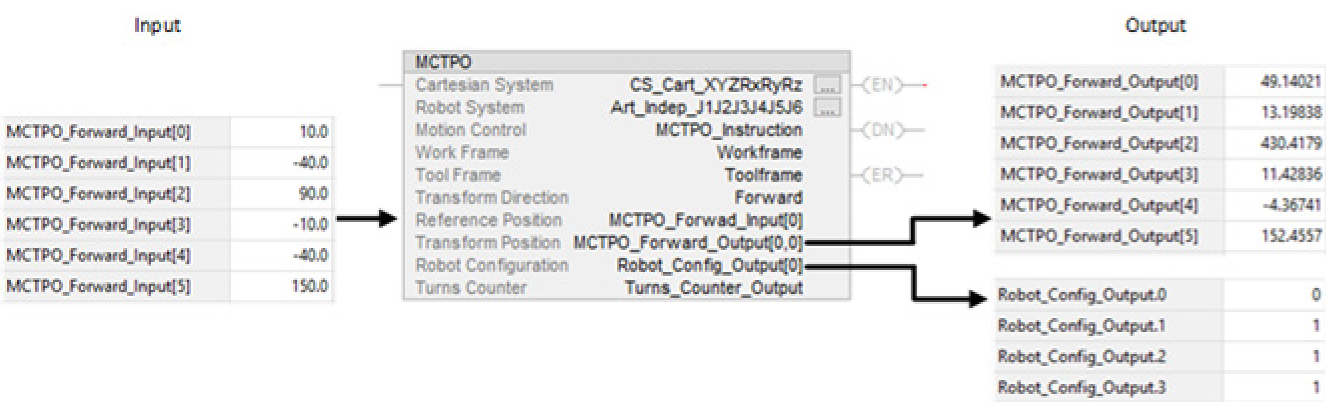
J1	J2	J3	J4	J5	J6	Wrist	Elbow	Arm	Attribute Value
-170	23.40698	-68.3597	-11.2105	35.03857	-28.4752	Non-Flip	Below	Righty	0
10	50	-74.8107	11.01504	35.74566	133.3298	Non-Flip	Below	Lefty	2
-170	-59.3211	83.54901	-168.828	35.17375	133.1365	Non-Flip	Above	Righty	4
10	-40	90	170	40	-30	Non-Flip	Above	Lefty	6
-170	23.40698	-68.3597	168.7895	-35.0386	151.5248	Flip	Below	Righty	8
10	50	-74.8107	-168.985	-35.7457	-46.6702	Flip	Below	Lefty	10
-170	-59.3211	83.54901	11.17249	-35.1738	-46.8635	Flip	Above	Righty	12
10	-40	90	-10	-40	150	Flip	Above	Lefty	14

Forward Transform example

This example illustrates a Motion Calculate Transform Position with Orientation (MCTPO) instruction with the transform direction as Forward. The configured target positions are guided into the reference position operand as input. The MCTPO instruction computes the corresponding Cartesian positions and robot configuration as the output.

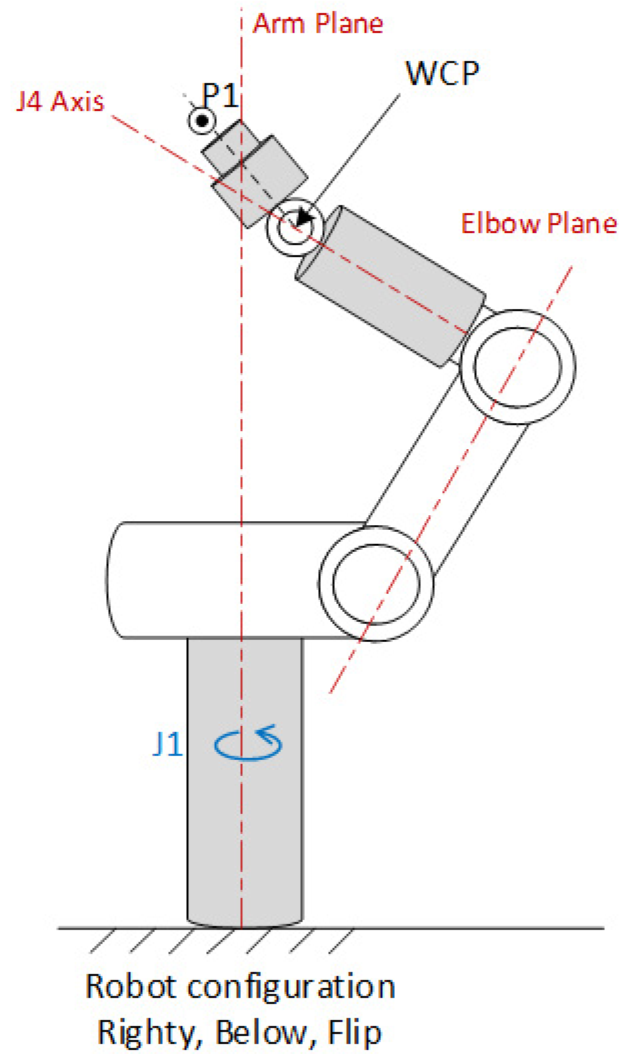
In this example, the target positions are evaluated as Lefty (1), Above (1), and Flip (1) configurations.

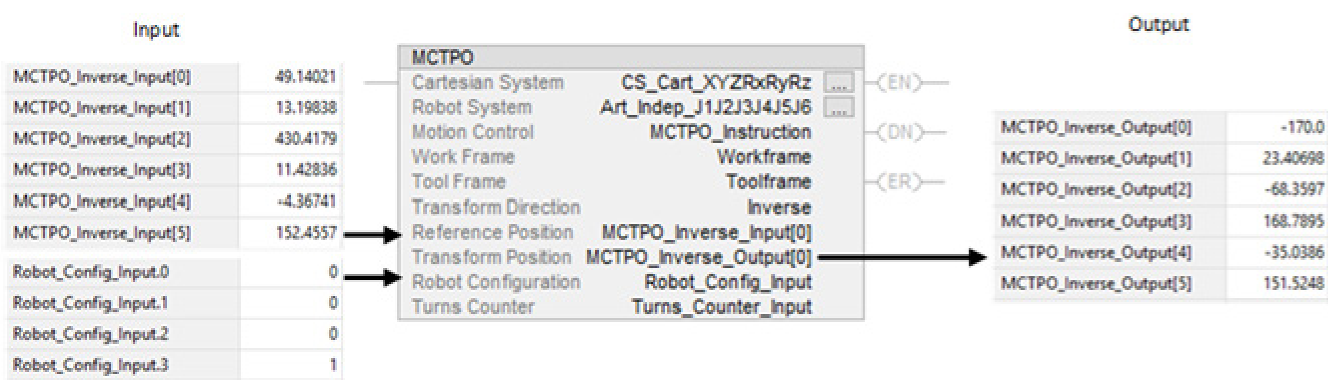




Inverse Transform example

This example illustrates an MCTPO instruction with the transform direction as Inverse, where the user provides the Cartesian position and robot configuration for Righty (0), Below (0), and Flip (1) configurations as an input. The instruction computes the corresponding target joint-angle positions for the robot configuration and writes them to the transform position parameter as the output.





Singularity error conditions

For an Articulated Independent J1J2J3J4J5J6 robot geometry, the motion instruction returns error code 156, SINGULARITY_CONDITION_ERROR, when the coordinate system is at singularity position.

- For the Arm Singularity, the extended error code is 1 (MOP_ARM_SINGULARITY)
- For the Elbow Singularity, the extended error code is 2 (MOP_ELBOW_SINGULARITY)
- For the Wrist Singularity, the extended error code is 3 (MOP_WRIST_SINGULARITY)

If any bit between 4 and 31 is set in the robot configuration while performing Inverse Transform, the motion instruction returns the error code 137 (INVALID_ROBOT_CONFIGURATION).

Configuration parameters for Articulated Independent J1J2J3J4J5J6 robots

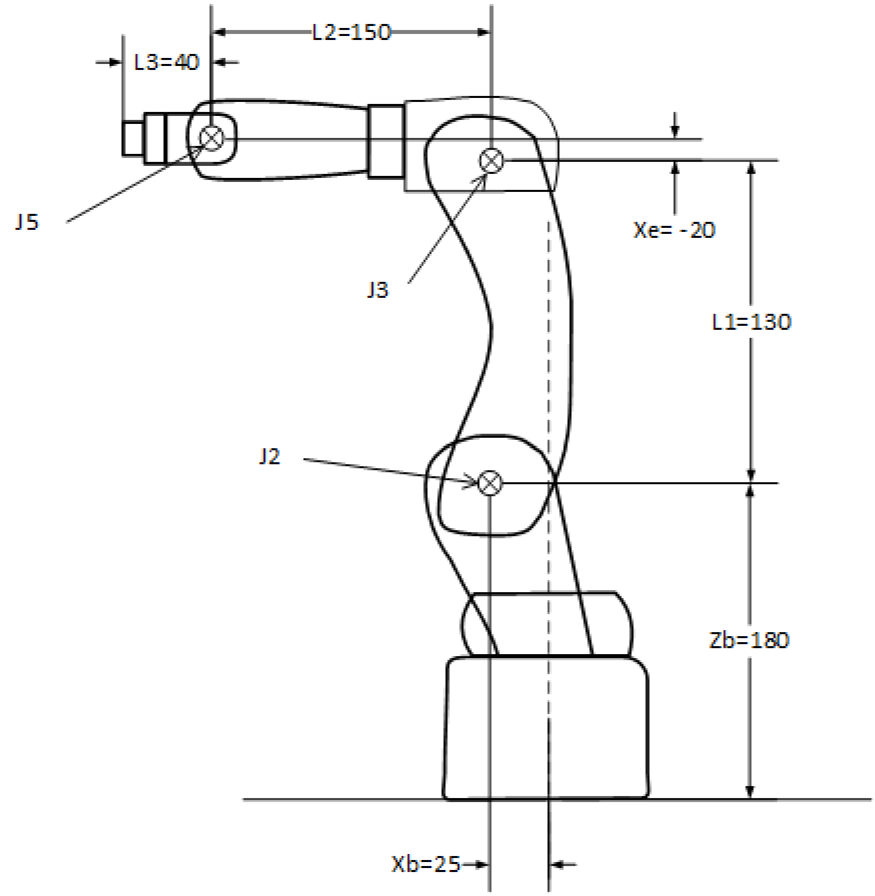
Configure these parameters for Articulated Independent J1J2J3J4J5J6 robots with varying reach and payload capacities:

- Link lengths
- Zero-angle orientation
- Base offsets
- End-effector offsets

The configuration parameter information is available from the robot manufacturer.

IMPORTANT	Be sure to use the same measurement units when you enter values for the link lengths, base offsets, and end-effector offsets.

This illustration shows the configuration parameters in a typical configuration for an Articulated Independent J1J2J3J4J5J6 robot.



See also

Link lengths for Articulated Independent J1J2J3J4J5J6 robots on page 86

Zero-angle orientations for Articulated Independent J1J2J3J4J5J6 robots on page 87

Base offsets for Articulated Independent J1J2J3J4J5J6 robots on page 89

End-effector offsets for Articulated Independent J1J2J3J4J5J6 robots on page 91

Offset error conditions on page 93

Link lengths for Articulated Independent J1J2J3J4J5J6 robots

Links L1, L2, and L3 are the rigid members of the robot joints.

Use the **Geometry** tab on the **Coordinate System Properties** dialog to configure link lengths L1, L2, and L3.

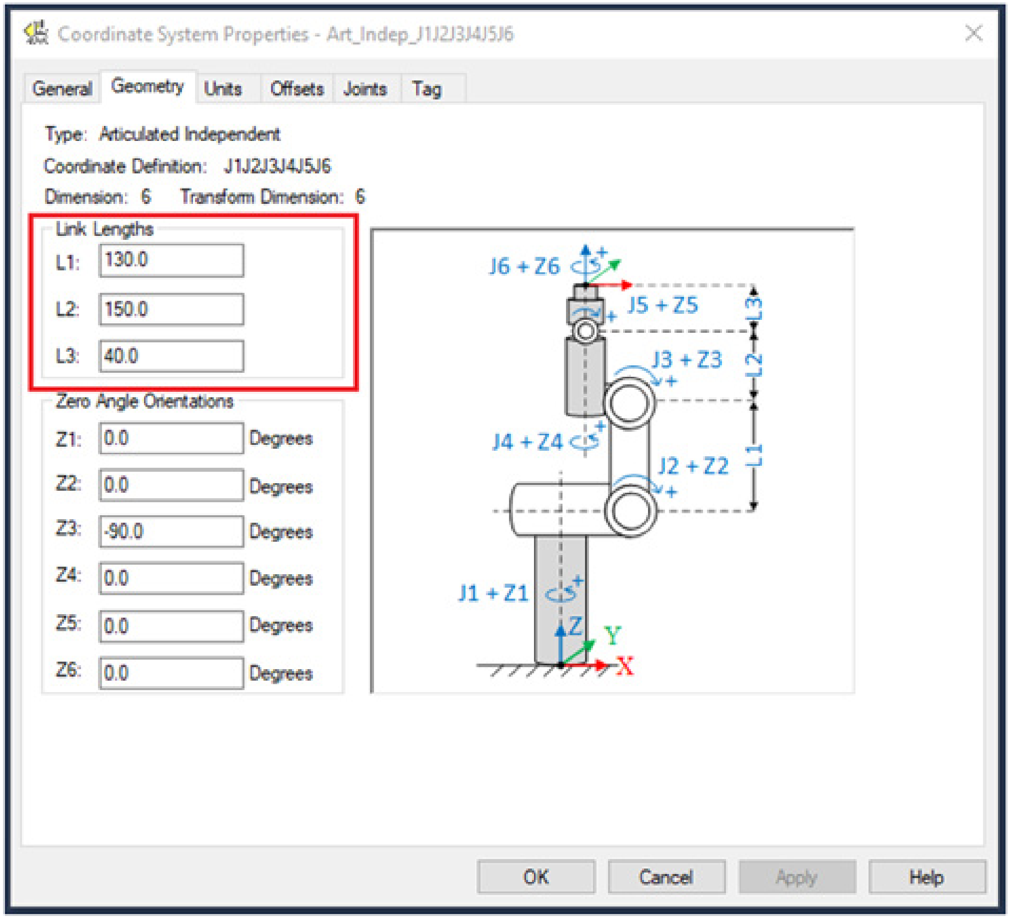
To open the **Coordinate System Properties** dialog, in the **Controller Organizer**, expand the **Motion Groups** folder, right-click the axis and then select **Properties**.

The link lengths are the distance between the axis of rotation of two joints:

- L1 is the link length between axes of rotation J2 and J3.
- L2 is the link length between axes of rotation J3 and J5.
- L3 is the link length between axes of rotation J5 and End of Arm (EOA).

This example shows link length values as:

- L1 = 130.0
- L2 = 150.0
- L3 = 40.0



Zero-angle orientations for Articulated Independent J1J2J3J4J5J6 robots

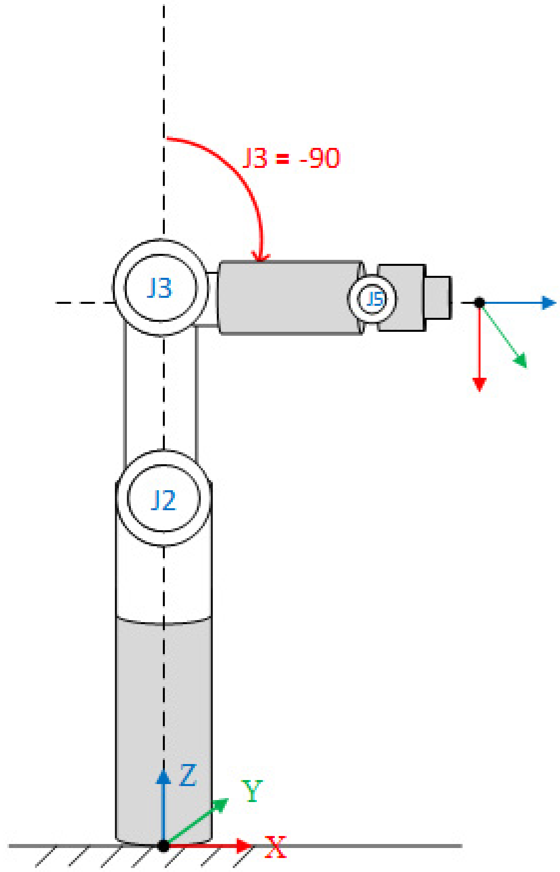
The zero-angle orientation is the rotational offset of the individual joint axes.

For Articulated Dependent J1J2J3J4J5J6 robot geometry, the internal transformation equations in the Logix Designer application assume that the initial positions for joints J1, J2, J3, J4, J5, and J6 are homed to 0°.

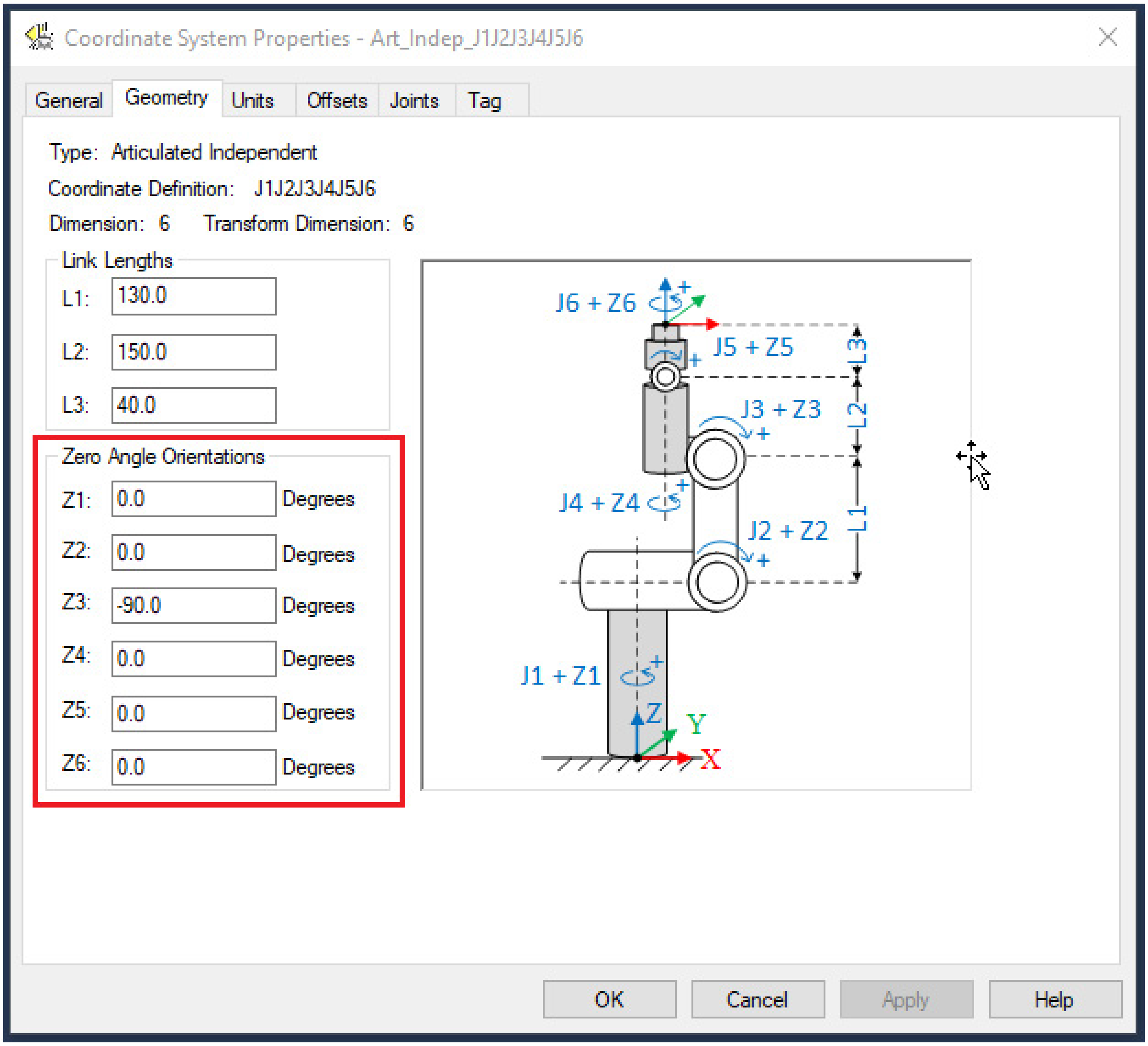
Zero-angle offsets establish reference frames other than the default home position. To set the angular positions for joints J1 through J6 to any value other than 0, configure the zero-angle orientation values on the **Geometry** tab in the **Coordinate System Properties** dialog to align the joint angle positions with the internal equations.

To open the **Coordinate System Properties** dialog, in the **Controller Organizer**, expand the **Motion Groups** folder, right-click the axis and then select **Properties**.

For example, to set the joint J3 axis position to a 0° home position at 90°, enter -90° for the **Z3** parameter. This illustration shows the J3 axis position set to 0°.



This screen capture shows the settings on the **Geometry** tab in the **Coordinate System Properties** dialog.



Base offsets for Articulated Independent J1J2J3J4J5J6 robots

Base offsets are a set of coordinate values that define the offset between the robot base and the origin of joint J2. The correct base offset values should be available from the robot manufacturer.

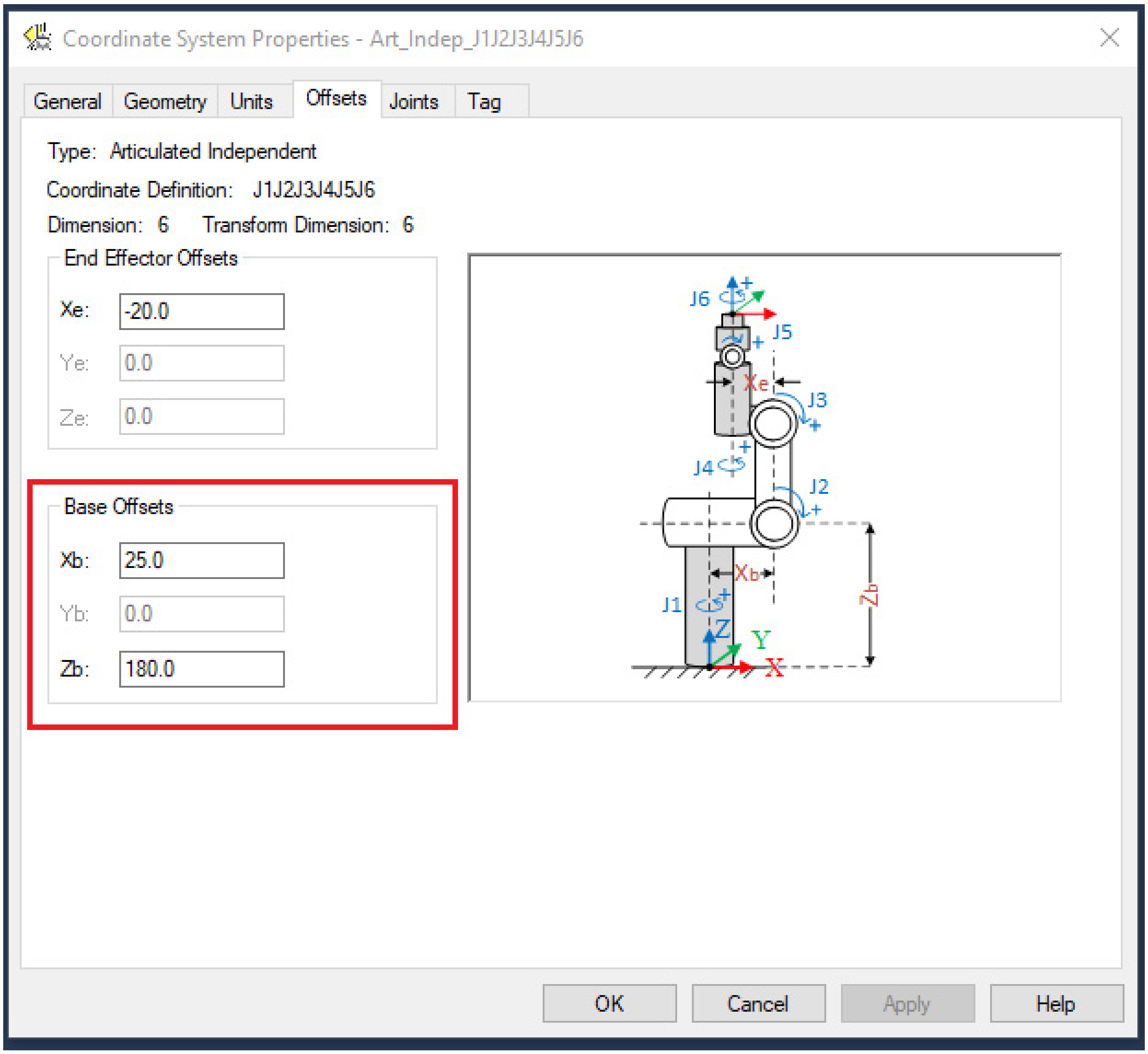
For an Articulated Independent J1J2J3J4J5J6 robot:

- **Xb** is the base offset between the robot base frame and the origin of joint J2 in the X-axis direction.
- **Zb** is the base offset between the robot base frame and the origin of joint J2 origin in the Z-axis direction.

Configure the values for the base offsets in the **Xb** and **Zb** boxes on the **Offsets** tab in the **Coordinate System Properties** dialog.

To open the **Coordinate System Properties** dialog, in the **Controller Organizer,** expand the **Motion Groups** folder, right-click the axis and then select **Properties**.

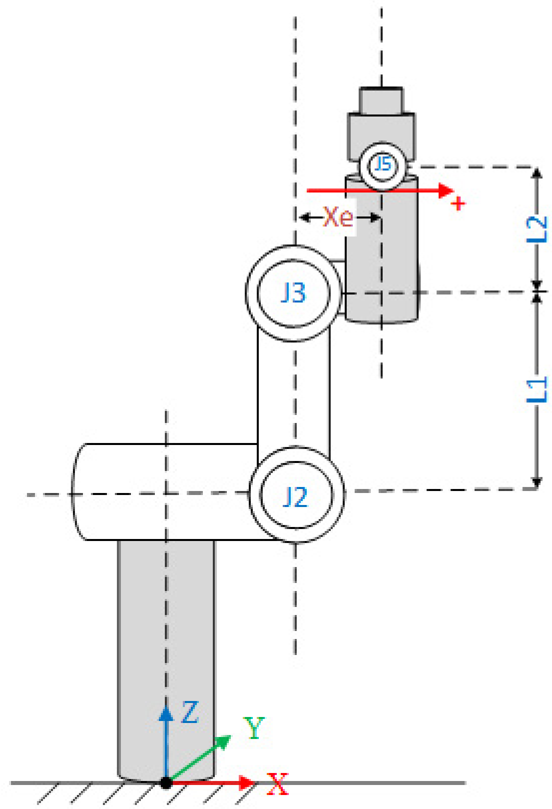
This illustration shows the base offsets on the **Offsets** tab. In this example, Xb = 25.0 and Zb = 180.0. The Logix Designer application does not support the Yb offset for an Articulated Independent 6-axis geometry.



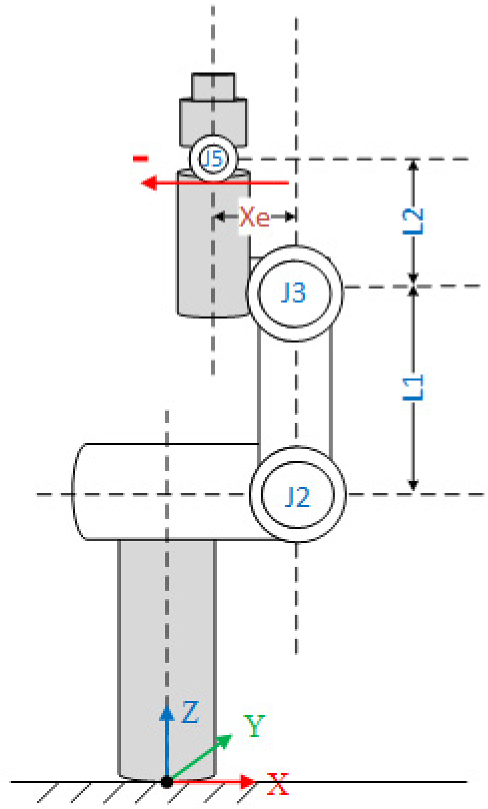
End-effector offsets for Articulated Independent J1J2J3J4J5J6 robots

For Articulated Independent J1J2J3J4J5J6 robots, the end-effector offset is **Xe**. Xe is a coordinate value that defines the offset between the end of link L1 and link L2 in the X-axis direction.

The sign of the Xe end-effector offset value is based on the plus (+) or minus (-) direction of the base frame X-axis. For example, the end-effector offset Xe is positive when the offset between link L1 and link L2 is on the right side of the J3 joint (in the same direction as the +X-axis). This illustration shows the positive Xe offset.



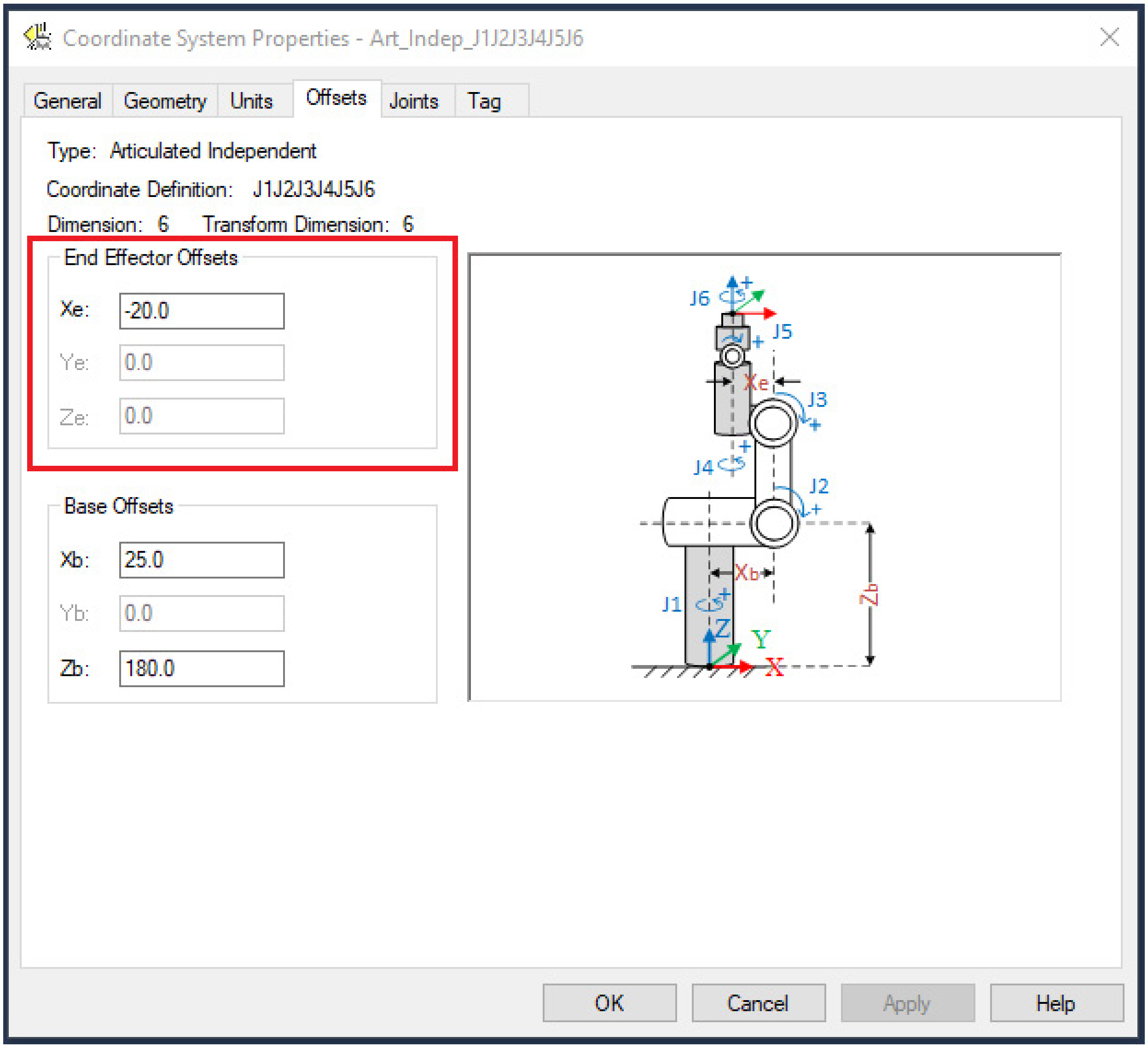
The end-effector offset Xe is negative when the offset between link L1 and link L2 is on the left side of the J3 joint (in the opposite direction of the +X-axis). This illustration shows the negative Xe offset.



Configure the values for the Xe end-effector offset in the **Xe** box on the **Offsets** tab in the **Coordinate System Properties** dialog.

To open the **Coordinate System Properties** dialog,in the **Controller Organizer**, expand the **Motion Groups** folder, right-click the axis and then select **Properties**.

This illustration shows the end-effector offsets on the **Offsets** tab. In this example, the Xe offset is -20.0. The Logix Designer application does not support the Ye and Ze offsets for an Articulated Independent 6-axis geometry.



Error conditions for Articulated Independent J1J2J3J4J5J6 robots

For an Articulated Independent 6-axis geometry, these conditions must be met:

- Base offset Yb must be equal to 0.0.
- End-effector offset Ye must be equal to 0.0.
- End-effector offset Ze must be equal to 0.0.

If these conditions are not met, the Logix Designer application generates error code 61 (CONNECTION_CONFLICT) and extended error 18 (TRANSFORM_INVALID_ARTICULATED_CONFIGURATION).

See also

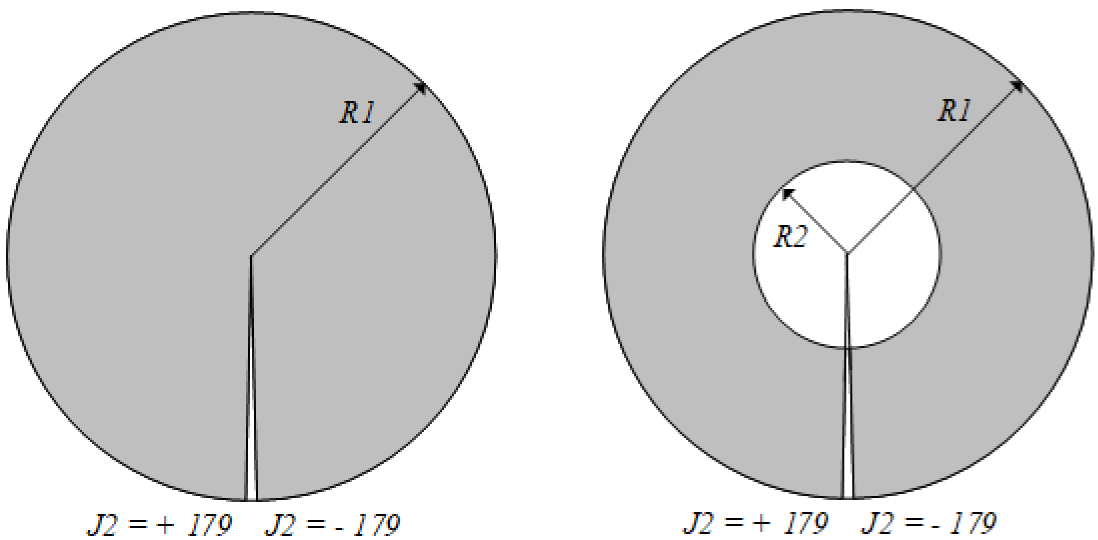
Configuration parameters for Articulated Independent J1J2J3J4J5J6 robots on page 85

Work envelope for Articulated Independent J1J2J3J4J5J6 robots

The work envelope is the three-dimensional region of space that defines the reaching boundaries for the Articulated Independent J1J2J3J4J5J6 robot geometry. The work envelope for the robot is a sphere with:

- An outer radius (R1) = L1 + L2 + L3
- An Inner radius (R2) = L1- (L2 + L3)

If the value of L2 + L3 is greater than the value of L1, then work envelope is a solid sphere excluding the mechanical limitation on J2. If the value of L1 is greater than the value of L2 + L3, then the work envelope is a hollow sphere.

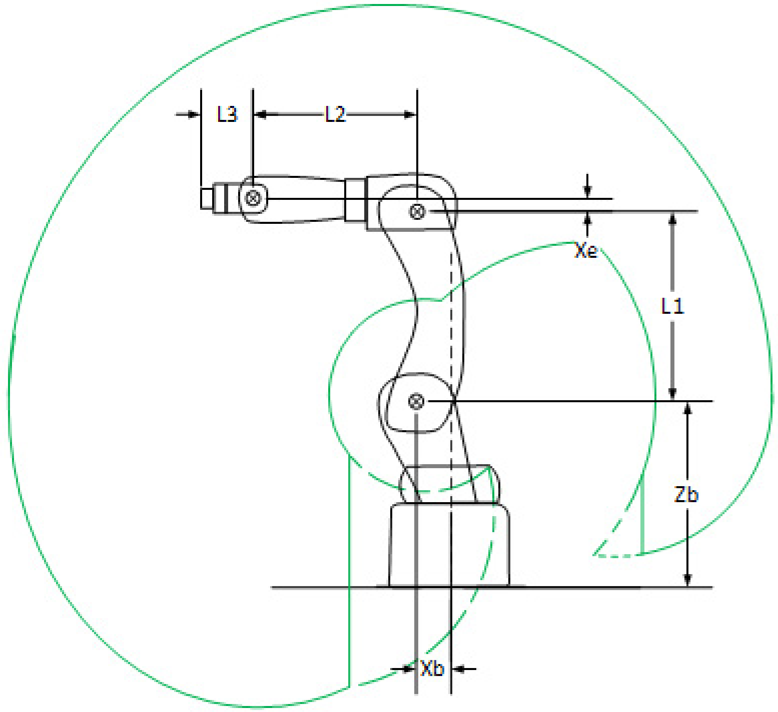


When L1 < L2 + L3   When L1 > L2 + L3

Keep these considerations in mind when determining the work envelope:

- Due to the limited range of motion on individual joints J2 and J3, the work envelope might not be a complete sphere.
- The work envelope for the Articulated Independent J1J2J3J4J5J6 robot varies if a tool is attached to the robot. The tool shape and dimensions might modify the work envelope.

This drawing shows the typical work envelope for Articulated Independent J1J2J3J4J5J6 robots, where R1 (Outer Radius - L1+L2+L3) is almost a complete sphere but the inner hollow section made by R2 is not an exact sphere.



See also

Configuration parameters for Articulated Independent J1J2J3J4J5J6 robots on page 85

Maximum joint limits for Articulated Dependent J1J2J3J4J5J6 robots on page 95

Maximum joint limits for Articulated Dependent J1J2J3J4J5J6 robots

Some robot joints have a movement range with multiple turns, but some do not. The ranges of robot joints are limited within -180.00° to 179.99°. To avoid any numerical calculation errors at +/-180°, joint calculations need to be restricted within +/-179.99° range. The turns-counter functionality supports joints that move beyond the +/-180° range.

- The maximum and minimum joint limits for joints J2, J3, and J5 are set to -180° to 179.99°. If the joint exceeds the limit, the Motion Coordinated Transform instruction generates error code 151 (JOINT_ANGLE_BEYOND_LIMIT) with the extended error code, specifying which joint exceeds the limit.
- Joints J1, J4, and J6 support multiple turns, so their limits are beyond the standard joint limits. The maximum and minimum joint limits for joints J1, J4, and J6 are set to -45899.99 to 45900.

See also

Configure joint limits on page 96

Configure joint limits

Use soft travel limits to configure joint limits for joint axes on Articulated Independent J1J2J3J4J5J6 robots.

To adjust soft travel limits

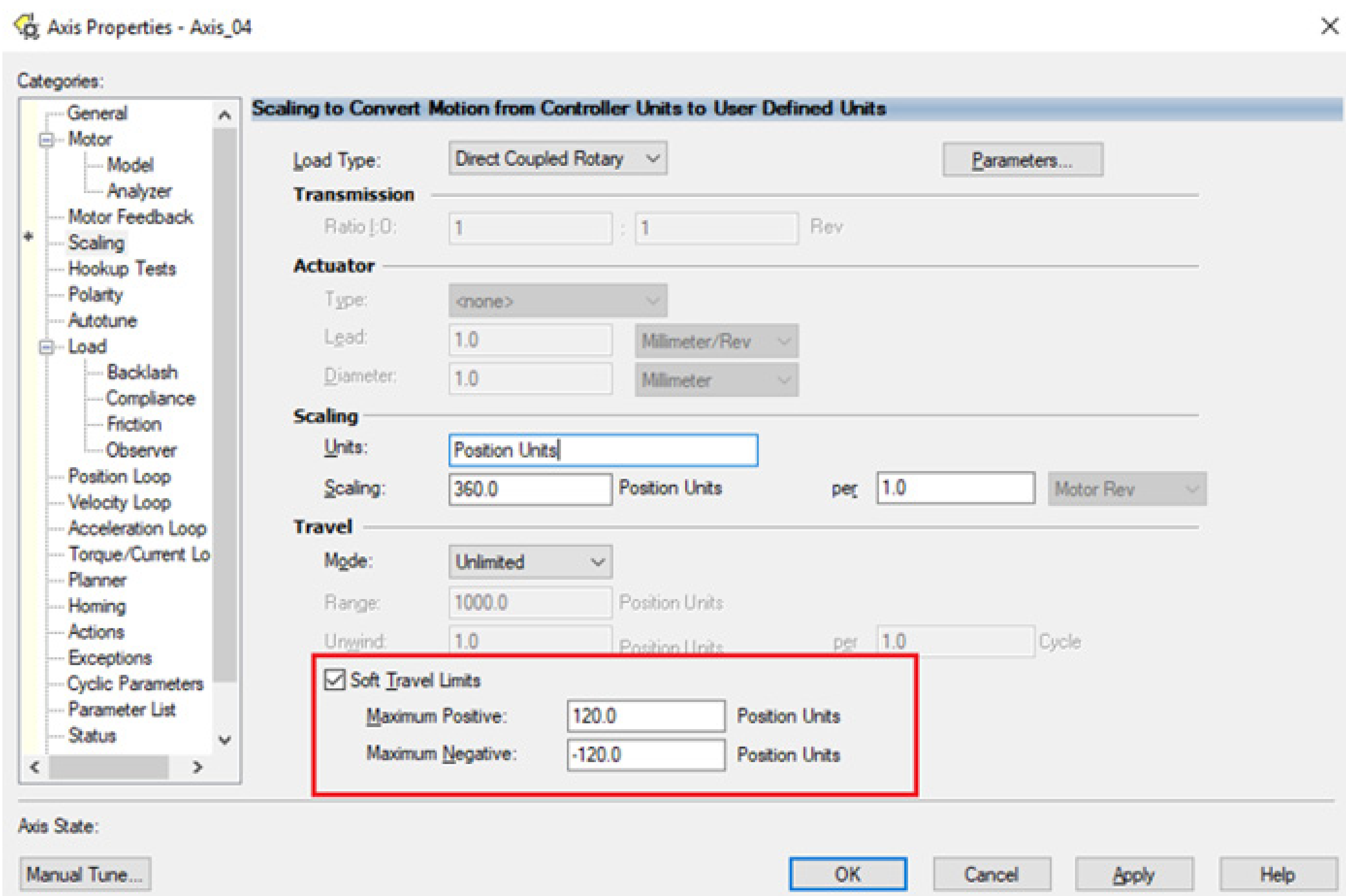
1. In **Axis Properties,** select the **Scaling** tab.

a. In the **Controller Organizer**, expand the **Motion Groups** folder, and then double-click the axis.

b. Select the **Scaling** tab.

2. Select **Soft Travel Limits**.
3. Enter the maximum positive and maximum negative limit values based on the mechanical limits of the joint axis. If the axis moves beyond the travel limits, the Software Positive/Negative Overtravel fault occurs.

This illustration shows the Soft Travel Limits settings.



See also

Configuration types for Articulated Independent J1J2J3J4J5J6 robots on page 78

Work envelope for Articulated Independent J1J2J3J4J5J6 robots on page 94

Work and tool frame offset limits

The work envelope for Articulated Independent J1J2J3J4J5J6 robots relies on the work and tool frame offset values defined in the Motion Coordinated Transform with Orientation (MCTO) and Motion Calculate Transform Position with Orientation (MCTPO) instructions. Work frame offsets are the offsets used to locate the user work frame of the robot relative to the origin of the robot base frame. These offsets consist of an XYZ and RxRyRz value.

Tool frame offsets locate the tool center relative to the center of the End of Arm (EOA). These offsets consist of XYZ and RxRyRz values.

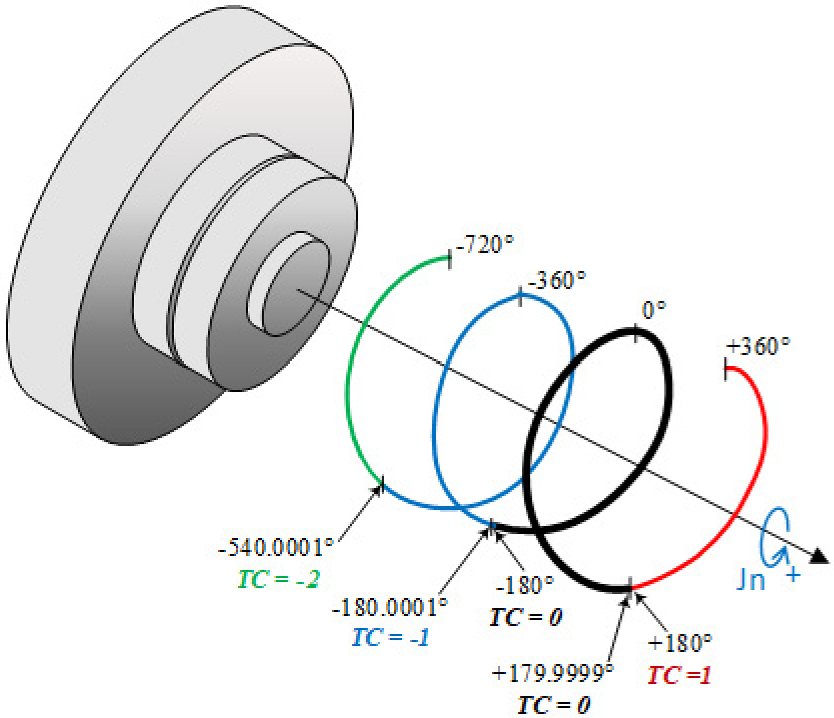
Any offset values on X, Y, Z, Rx, Ry, and Rz axis are allowed for the work and tool frame offsets.

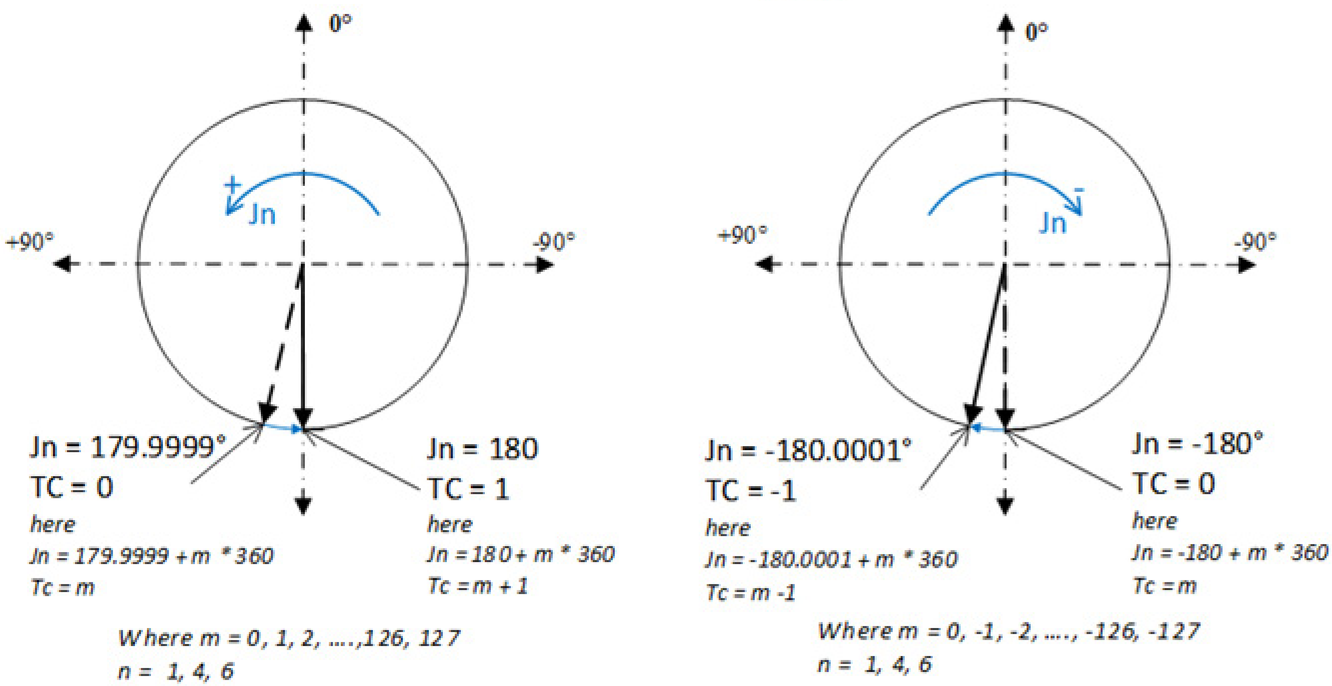
Turns counters for Articulated Independent J1J2J3J4J5J6 robots

The Articulated Independent J1J2J3J4J5J6 geometry support turns counters on multiple revolute joints. Three turns counters are supported for the geometry: J1, J4, and J6. The maximum range for joint axes is -180 degrees to 179.9999 degrees. For turns counter axes, when the joint position limit is exceeded, the relevant joint turns counter decrements or increments by one and the joint position changes sign, from 179.9999 degrees to -180 degrees.

Turns counters monitor how many revolutions the robot joints accumulate. You can use this number to monitor how close a robot is to the physical-joint limits, and to help keep joint values in the range of -180 degrees to 179.9999 degrees.

This example illustrates the relationship between the turns-counts and the joint-angle.





Tip: If a joint reaches the point -180 degrees but does not cross over, the joint does not flip and stays at -180 degrees. If the joint reaches the point 180 degrees, the value flips to -180 degrees and the turns counter value is updated. The turns counter and the joint angle behavior are relative to an absolute joint position moving for the robot.

This table lists absolute joint angles with relative turns counter values and representations of joint angles within the range 179.9999 to -180.0000.

Absolute Joint Angle	Effective Turns counter	Effective Joint angle value
179.9999	0	179.9999
180.0000	1	-180.0000
180.0001	1	-179.9999

181.0000	1	-179.0000
190.0000	1	-170.0000
360.0000	1	0.0000
-179.9999	0	-179.9999
-180.0000	0	-180.0000
-180.0001	-1	179.9999
-181.0000	-1	179.0000
-190.0000	-1	170.0000
-360.0000	-1	0.0000

See also

Turns counter limits on page 99

Turns counter example on page 99

Turns counter limits

Each turns counter for the Articulated Independent geometry has a maximum limit of ±127. Exceeding this limit for any joint axis generates a JOINT_ANGLE_BEYOND_LIMIT error with an extended error code for the relevant joint. For example, these errors could appear for joints J1, J4, and J6:

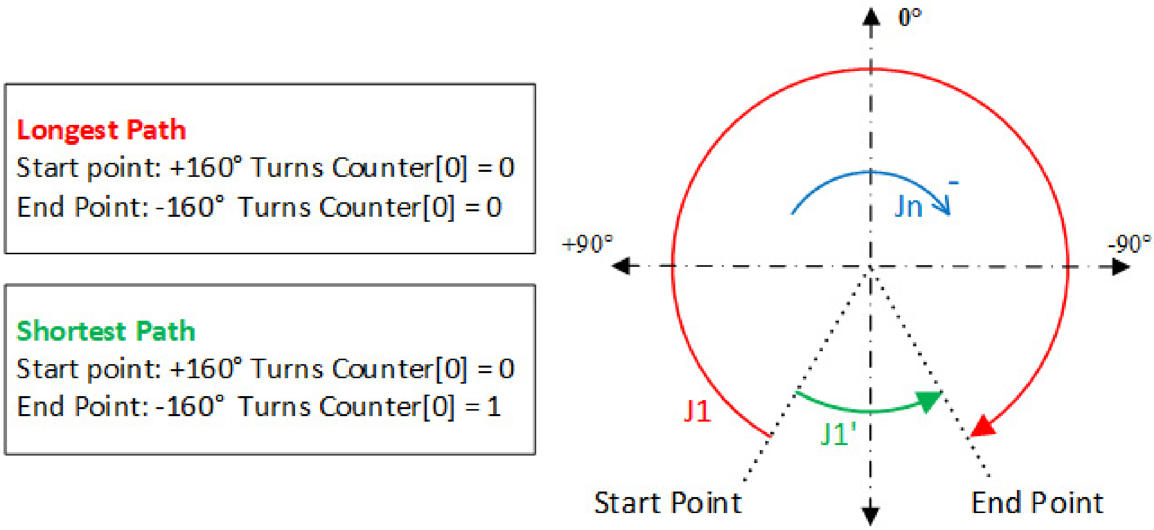
- JOINT_J1_BEYOND_LIMIT
- JOINT_J4_BEYOND_LIMIT
- JOINT_J6_BEYOND_LIMIT

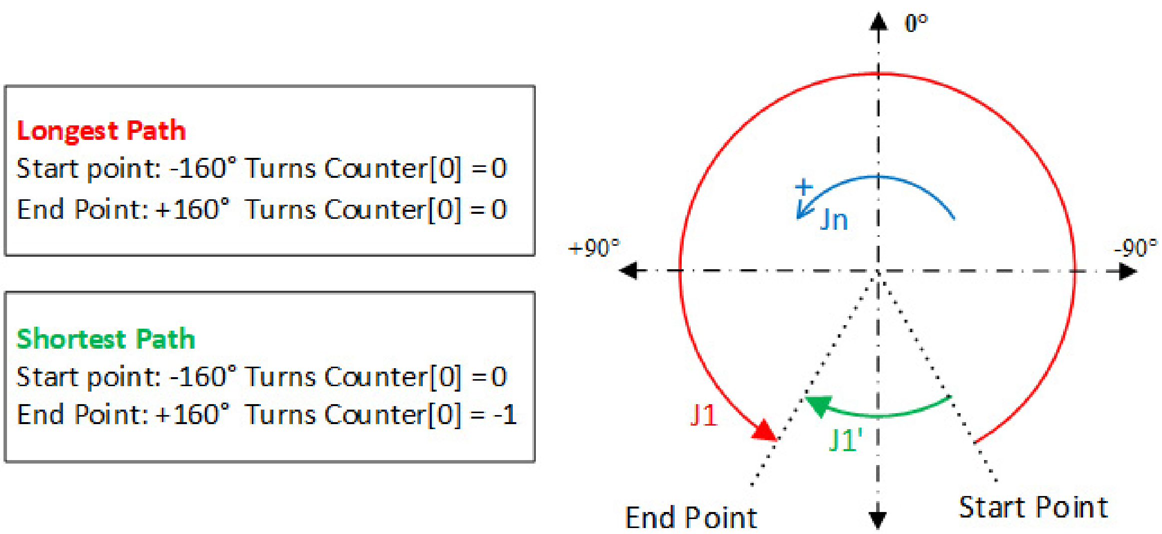
If any JOINT_ANGLE_BEYOND_LIMIT errors are generated, all motion of the Articulated Independent robot ceases. The robot cannot move until you clear the error.

Turns counter example

Use turns counters to monitor how many revolutions the robot joints accumulate. You can use the number of accumulated revolutions in program logic to monitor how close a robot is to the physical-joint limits, and to help keep joint values in the range of -180 degrees to 179.9999 degrees.

This illustration shows the shortest and longest paths for the joints to travel in the Cartesian space. The illustration shows the effect on the turns counter calculation. You can configure the same Cartesian position with a different combination of the turns counter as an input.

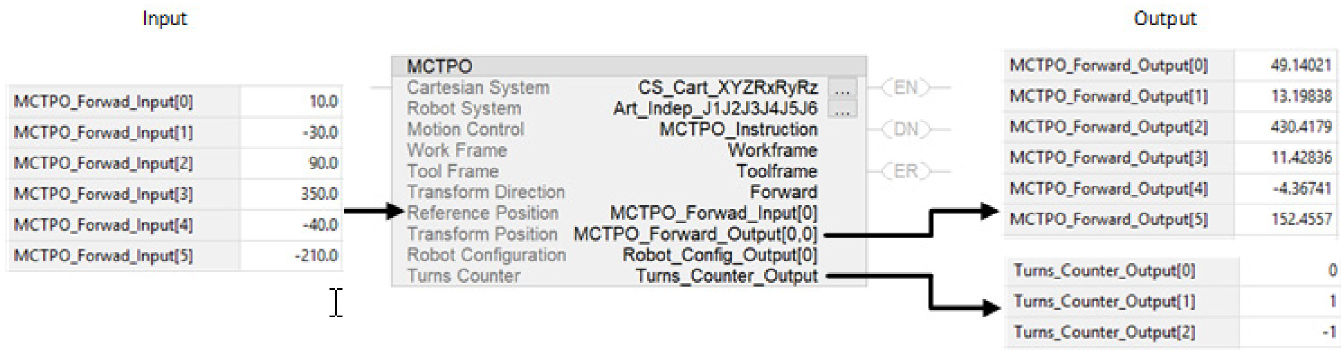




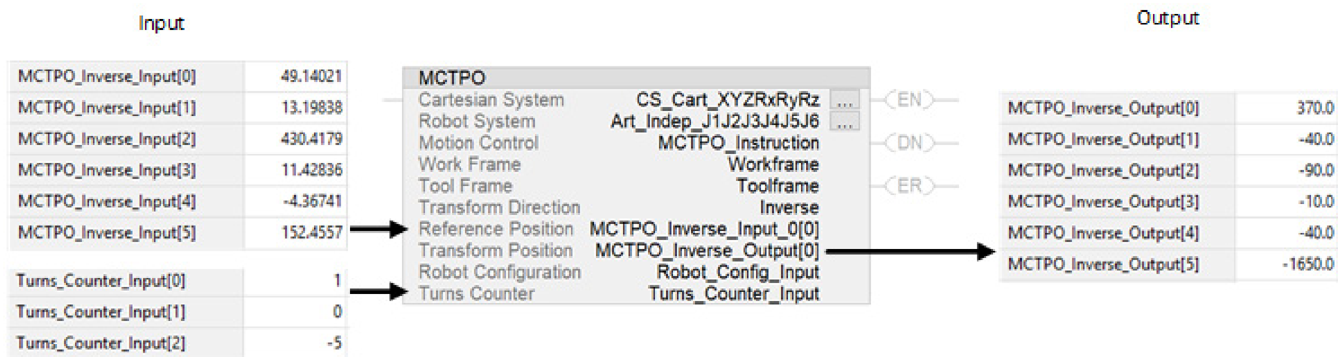
This table lists the Cartesian positions, turns counter values, and joint positions for this example. The turns counter is an input for the inverse transform calculation and an output for the forward transform calculation.

Cartesian positions						Turns counters			Joint positions					
X	Y	Z	Rx	Ry	Rz	J1	J4	J6	J1	J2	J3	J4	J5	J6
49.14021	13.19838	430.4179	11.42836	-4.3674	152.4557	0	0	0	10	-40	90	-10	-40	150
49.14021	13.19838	430.4179	11.42836	-4.3674	152.4557	0	1	-1	10	-40	90	350	-40	-210
49.14021	13.19838	430.4179	11.42836	-4.3674	152.4557	1	0	-5	370	-40	90	-10	-40	-1650

A Motion Calculate Transform Position with Orientation (MCTPO) instruction that uses a forward transform at a given position, with joints with values greater than ±180 degrees, produces these results:



A similar application that uses inverse transforms with the robot geometry accepts Cartesian positions and the turns counter as an input.



For this application, the MCTPO instruction calculates the appropriate joint positions for the Cartesian input but adds turns to the joint axes according to the user-specified turns counter input to the instruction.

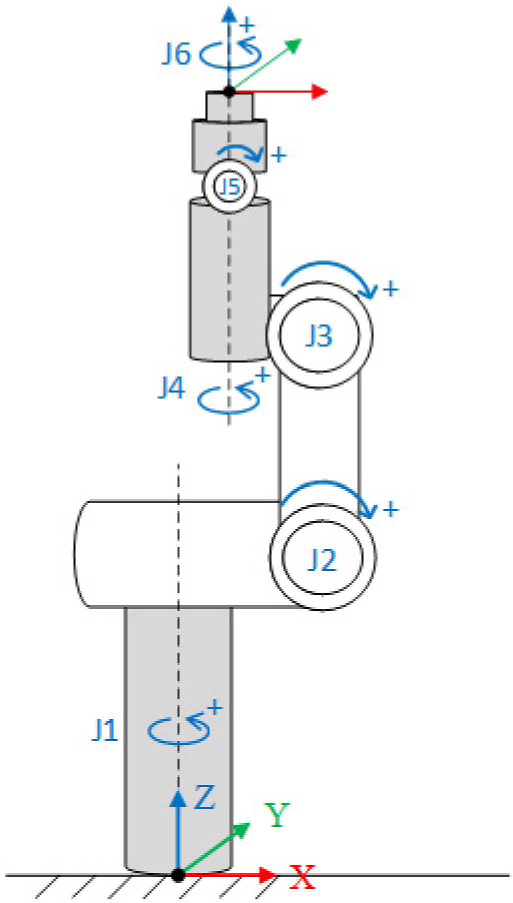
Robot joint direction sense bits

Use the joint direction sense functionality to change the convention of the default direction of the joint axes to match the robot setup.

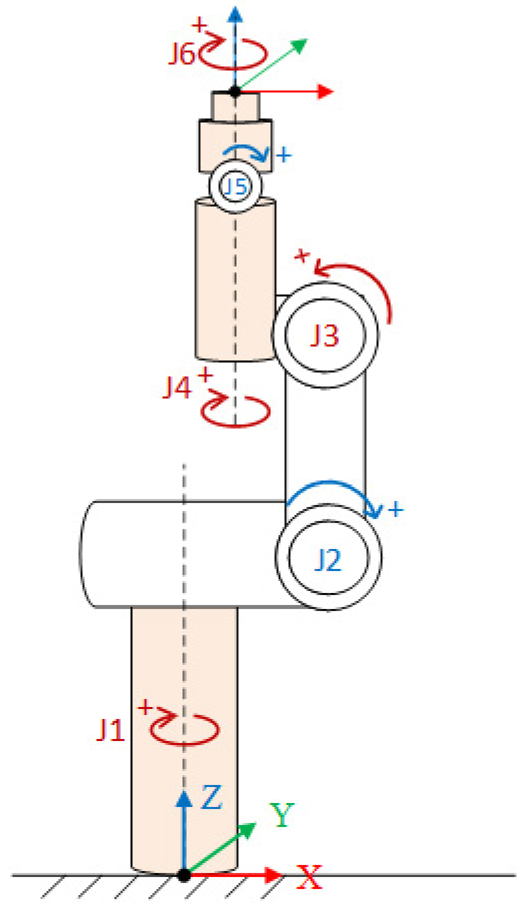
Some robots use joint directions differ from the default directions in the Logix Designer application for the Articulated Independent J1J2J3J4J5J6 robot geometry. For joints with inverted conventions that differ from Rockwell Automation conventions, program the coordinate system using the attribute Robot Joint Direction Sense.

IMPORTANT Changing the robot-joint-direction senses in the Logix Designer application does not affect the robot configuration of the geometry. For the user convention, the robot configuration remains the same as the Rockwell Automation convention.

This illustration shows the default Rockwell Automation convention compared to an example of a geometry with inverted direction senses for joints J1, J3, J4, and J6.



Rockwell Automation Convention



User Convention

See also

Program the robot joint direction senses on page 102

Transform, zero-angle offset, and turns counter calculations when using joint direction sense on page 104

Joint direction sense bit error conditions on page 108

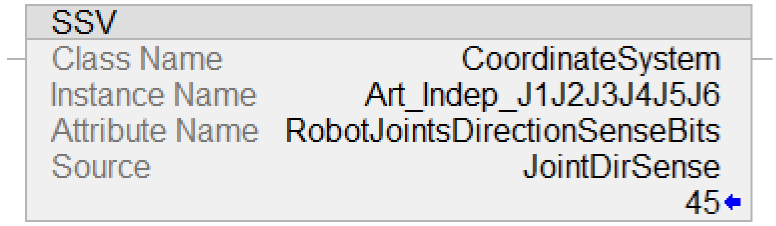
Program the robot joint direction senses

Use the Set System Value (SSV) instruction for the coordinate system to program the robot joint direction senses. By default, all joint direction sense bits are zero.

This example shows a bitmap for the joint direction sense attribute and a corresponding SSV instruction.

Bitmap	Joint	Joint direction sense
Bits 6 - 31	Not applicable	Not applicable
Bit 5	J6	1

Bitmap	Joint	Joint direction sense
Bit 4	J5	0
Bit 3	J4	1
Bit 2	J3	1
Bit 1	J2	0
Bit 0	J1	1



MCTO Behavior

The joint direction changes go into effect when the Motion Coordinated Transform with Orientation (MCTO) instruction is reinitiated.

If the new joint-direction-sense bits change while the MCTO is active, the transform ignores the new changes.

MCTPO Behavior

When a user updates the robot-joint-direction sense attribute, the consecutive Motion Calculate Transform Position with Orientation (MCTPO) instruction calculates the transform position using the new robot joint direction senses.

See also

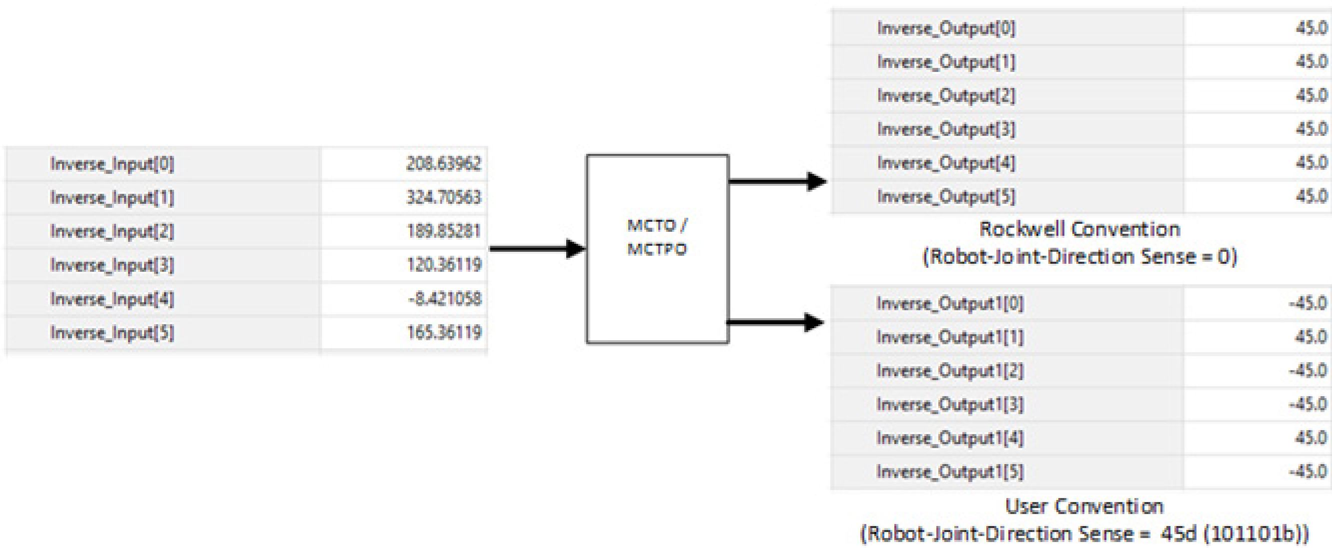
Joint direction sense bit error conditions on page 108

Transform, zero-angle offset, and turns counter calculations when using joint direction sense

These examples show the effect of the joint direction sense attribute on transform calculations.

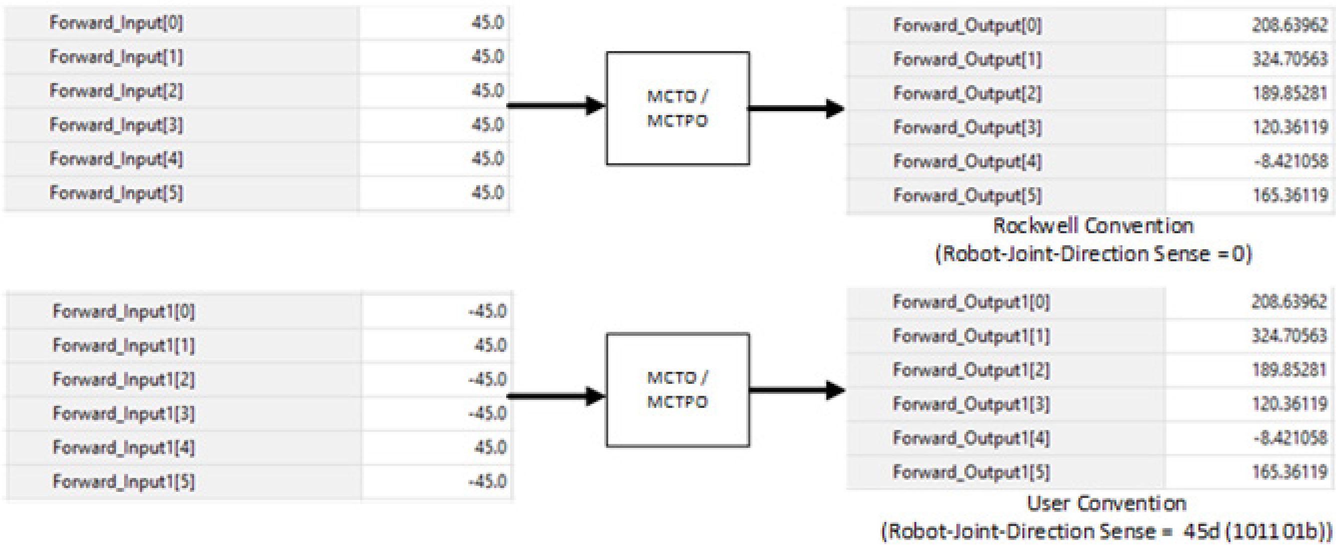
Inverse Transform

This illustration compares the inverse transform for the Rockwell Automation convention to a user-defined convention. In this example, the joint direction sense attribute is set for joints J1, J3, J4, and J6 in the user convention. For the given Cartesian position, robot configuration, and turns counter, the inverse transform calculates different joint positions. Notice the sign change on joints J1, J3, J4, and J6.



Forward Transform

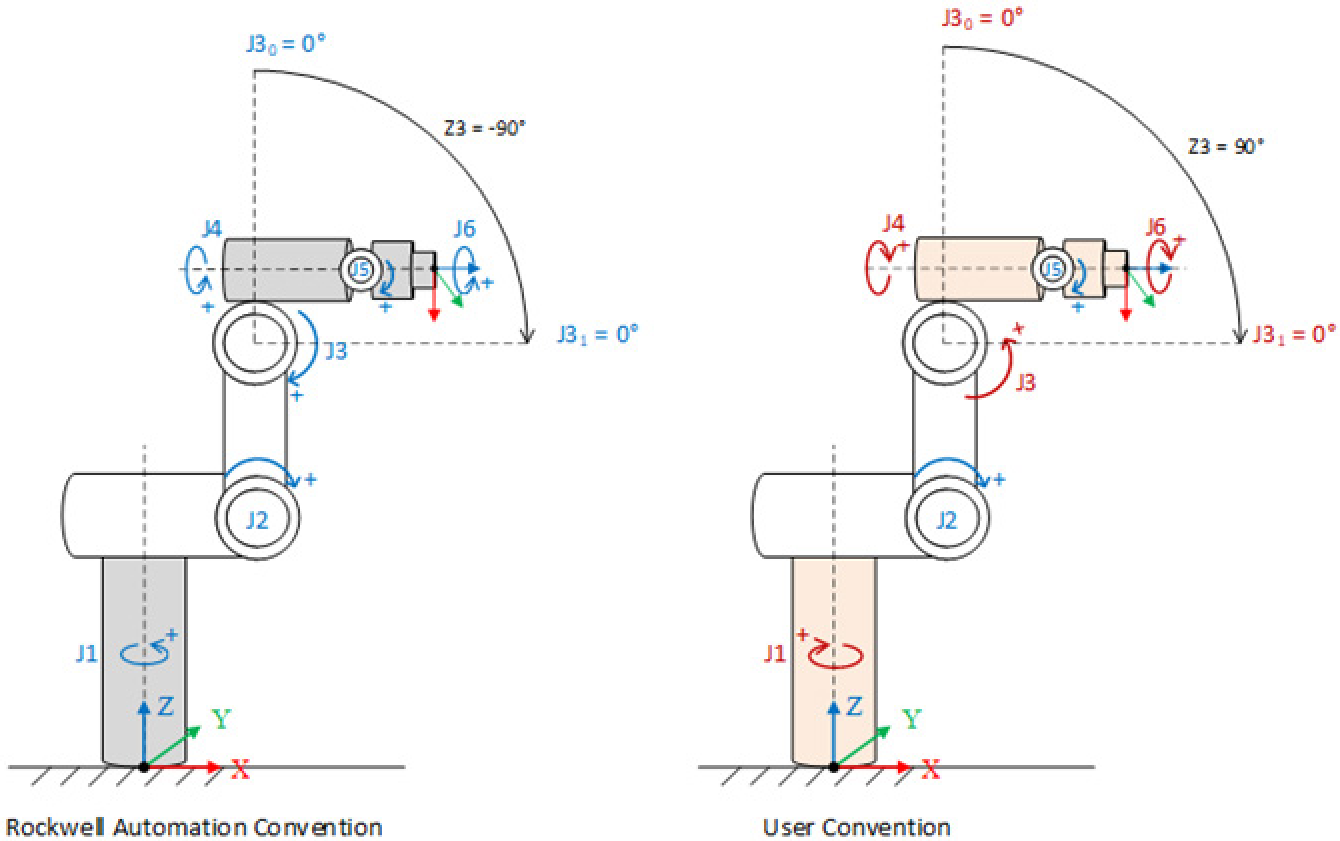
This illustration compares the forward transform calculation for the Rockwell Automation convention to a user-defined convention. In this example, the joint direction sense attribute is set for joints J1, J3, J4, and J6 in the user-defined convention. The transform calculates the same Cartesian output positions for joint positions for both instances.



Zero-angle offset calculation

A zero-angle offset defines the new zero-angle for the joint of the robot. The zero-angle offset is applicable to all six joints.

This illustration compares the zero-angle offset calculation in the Rockwell Automation convention and the user convention. In this example, the joint-direction sense attribute is set for joints J1, J3, J4, and J6. A 90° offset is added to joint J3. The calculated zero-angle offset is noted as Z3.



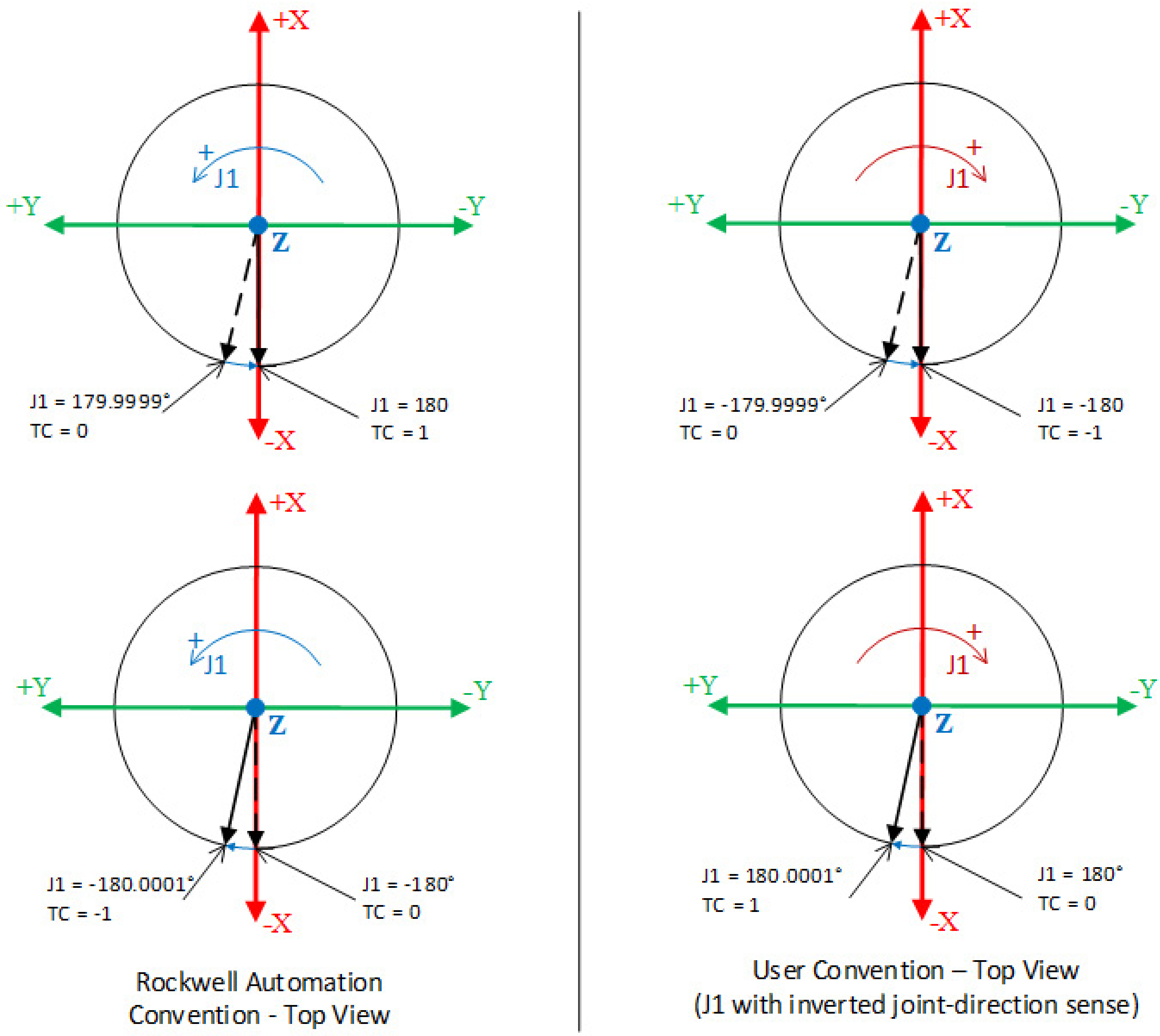
Position $J3_0$ is the default zero position for a joint J3, and position $J3_1$ is the new zero position after adding the zero-angle offset. This table lists the programmed offsets:

Orientation offset	Rockwell Automation convention	User example convention
Z3	-90	90

Turns-counter calculations

The turns-counter value is calculated based on the user convention. If the user-defined convention sets the joint direction sense bit, then the turns-count sign is the opposite of the Rockwell Automation convention. The threshold angle at which the turns counter increments changes when the user sets the joint-direction sense.

This illustration compares a turns-counter calculation for joint J1 in the Rockwell Automation convention to an example user convention. In the user convention, the joint direction sense is set for joint J1. Physical rotational direction for the joint axis is inverted when the joint-direction sense is inverted. The turns-counter calculation value depends on the joint values. This turns-counter calculation is applicable for joints J1, J4, and J6.



This table lists the results of the calculation.

Joint J1 (degrees)	Turns Counter J1 (Rockwell Automation Convention)	Turns Counter J1 (Inverted Joint Senses)
0 to 179.9999	0	0
179.9999	0	0
180	1	0
180.0001	1	1
0 to -179.9999	0	0
-179.9999	0	0
-180	0	-1
-180.0001	-1	-1

See also

Joint direction sense bit error conditions on page 108

Joint direction sense bit error conditions

The geometry configured with the joint-direction senses does not support coordinated moves that use the Motion Coordinated Path Move (MCPM) instruction. The MCPM instruction returns error code 157 (MCPM_JOINT_DIRECTION_SENSE_NOT_SUPPORTED) when the joint-direction senses are programmed.

Tip: To move an individual axis in a coordinate system, use an axis move instruction such as Motion Axis Jog (MAJ), Motion Coordinated Linear Move (MCLM), or Motion Axis Move (MAM).

Configure Articulated Dependent robots

Follow these guidelines for configuring articulated dependent robots:

- Articulated dependent J1J2J3 robots
- Articulated dependent J1J2J3J6 robots

WARNING: Before turning ON the Transform and/or establishing the reference frame, do the following for the joints of the target coordinate system:

- Set and enable the soft travel limits.
- Enable the hard travel limits.

Failure to perform these steps can cause robotic arm to move to unexpected positions causing machine damage and/or injury or death to personnel.

See also

Configure an Articulated Dependent J1J2J3 robot on page 108

Configure an Articulated Dependent J1J2J3J6 robot on page 115

Configure an Articulated Dependent J1J2J3 robot

Articulated dependent J1J2J3 robots contain motors for the elbow and the shoulder at the base of the robot. The dependent link controls J3 at the elbow.

WARNING: Before turning ON the Transform and/or establishing the reference frame, do the following for the joints of the target coordinate system:

- Set and enable the soft travel limits.
- Enable the hard travel limits.

Failure to perform these steps can cause robotic arm to move to unexpected positions causing machine damage and/or injury or death to personnel.

See also

Reference frame for Articulated Dependent J1J2J3 robots on page 109

Work envelope for Articulated Dependent J1J2J3 robots on page 111

Configuration parameters for Articulated Dependent J1J2J3 robots on page 112

Base offsets for Articulated Dependent J1J2J3 robots on page 114

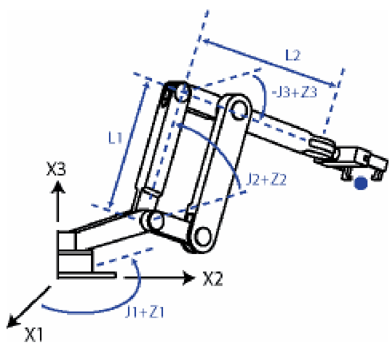
Reference frame for Articulated Dependent J1J2J3 robots

The reference frame is the Cartesian (typically the source) coordinate frame that defines the origin and the primary axes, X1, X2, and X3. These are used to measure the real Cartesian positions.

WARNING: Failure to properly establish the correct reference frame for the robot can cause the robotic arm to move to unexpected positions causing machine damage and/or injury or death to personnel.

Example 1: Articulated Dependent robot 1

This diagram illustrates the reference frame for an Articulated Dependent robot at the base of the robot.



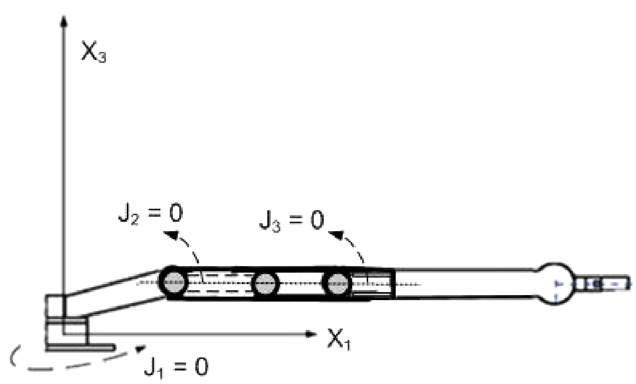
These equations represent the Articulated Dependent robot joint positioning shown in Articulated Dependent robot 1 diagram.

- +J1 is measured counterclockwise around the +X3 axis starting at an angle of J1=0 when L1 and L2 are both in the X1-X2 plane.
- +J2 is measured counterclockwise starting with J2=0 when L1 is parallel to X1-X2 plane.
- +J3 is measured counterclockwise with J3=0 when L2 is parallel to the X1-X2 plane.

When the robot is in this position, the Logix Designer application Actual Position tags for the axes must be:

- J1 = 0.
- J2 = 0.
- J3 = 0.

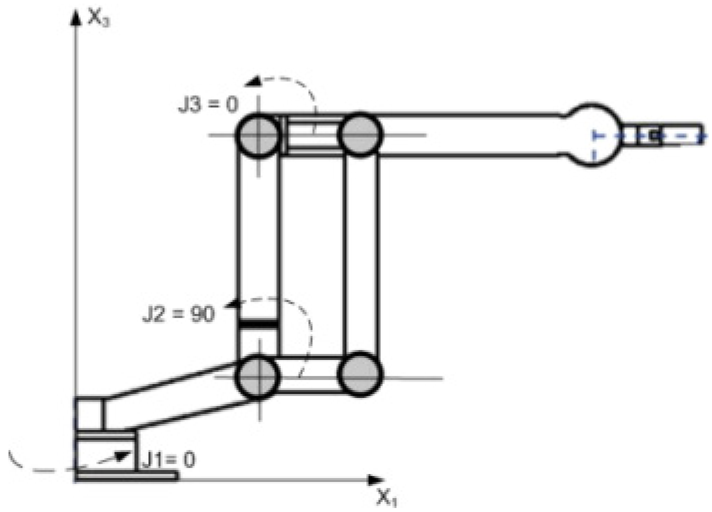
Example 2: Figure 79 - Articulated Dependent 2



When the robot is in this position, the Logix Designer application Actual Position tags for the axes must be:

- J1 = 0.
- J2 = 90.
- J3 = -90.

Example 3: Articulated Dependent 3



If the position and joint angle values of the robot are unable to match the Articulated Dependent 2 or in Articulated Dependent 3 examples, use a method outlined in the Method to Establish a Reference Frame for an articulated dependent robot topic to establish the Joint-to-Cartesian reference frame relationship.

Methods to establish a reference frame for an Articulated Dependent J1J2J3 robot

Use these methods to establish a reference frame for the robot.

For each:	Use one of these methods to establish the reference frame:
Incremental axis	Each time the power for the robot is cycled.
Absolute axis	Only to establish absolute home.

- Method 1 - Establishes a Zero Angle Orientation and allows the configured travel limits and home position on the joint axes to remain operational. Use this method when operating the axes between the travel limits determined prior to programming a Motion Redefine Position (MRP) instruction and want these travel limits to stay operational.
- Method 2 - Uses an MRP instruction to redefine the axes position to align with the joint reference frame. This method may require the soft travel limits to be adjusted to the new reference frame.

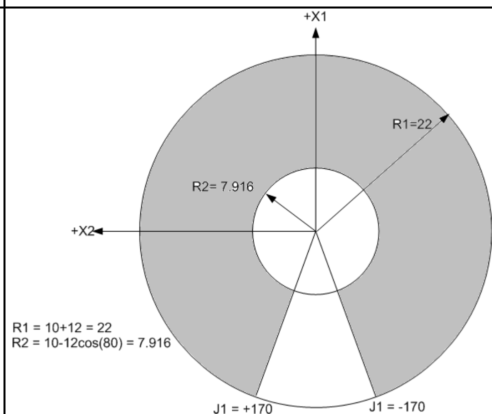
See also

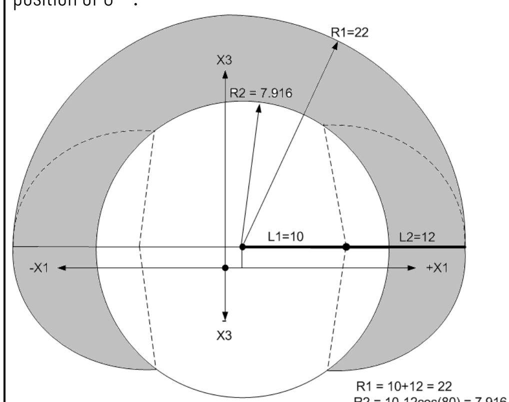
Method 1 - Establish a reference frame using zero angle orientation on page 68

Method 2 - Establish a reference frame using an MRP instruction on page 69

Work envelope for Articulated Dependent J1J2J3 robots

The work envelope is the three-dimensional region of space that defines the reaching boundaries for the robot arm. The work envelope for an articulated robot is ideally a complete sphere with an inner radius equal to L1- L2 and outer radius equal to L1+L2. Due to the range of motion limitations on individual joints, the work envelope may not be a complete sphere.

<table>
<tr><th>If the range-of-motion values for the articulated robot are:</th><th>Typically, the work envelope is:</th></tr>
<tr><td>J1 = ± 170
J2 = 0 to 180
J3 = ± 60
L1 = 10
L2 = 12</td><td>


Top view - Depicts the envelope of the tool center point sweep in J1 and J3 while J2 remains at a fixed position of 0°.


Side view - Depicts the envelope of the tool center point sweep in J2 and J3 while J1 remains at a fixed position of 0°.
</td></tr>
</table>

See also

Configuration parameters for Articulated Dependent robot on page 112

Articulated dependent robot on page 108

Configuration parameters for Articulated Dependent J1J2J3 robots

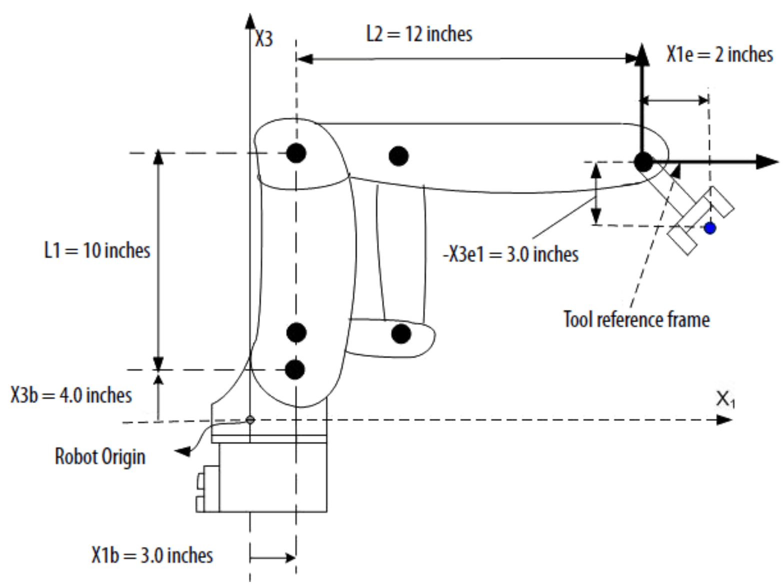
Configure the Logix Designer application to control robots with varying reach and payload capacities. Be sure to have these configuration parameter values for the robot:

- Link lengths
- Base offsets
- End-effector offsets

The configuration parameter information is available from the robot manufacturer.

IMPORTANT	Verify that the values for the link lengths, base offsets, and end-effector offsets are entered into the **Configuration Parameters** dialog box using the same measurement units.

This example illustrates the typical configuration parameters for an Articulated Dependent robot.



If the robot is two-dimensional, the X3b and X3e are X2b and X2e.

See also

Link lengths for Articulated Dependent robot on page 112

Link lengths for Articulated Dependent J1J2J3 robots

Link lengths are the rigid mechanical bodies attached at joints.

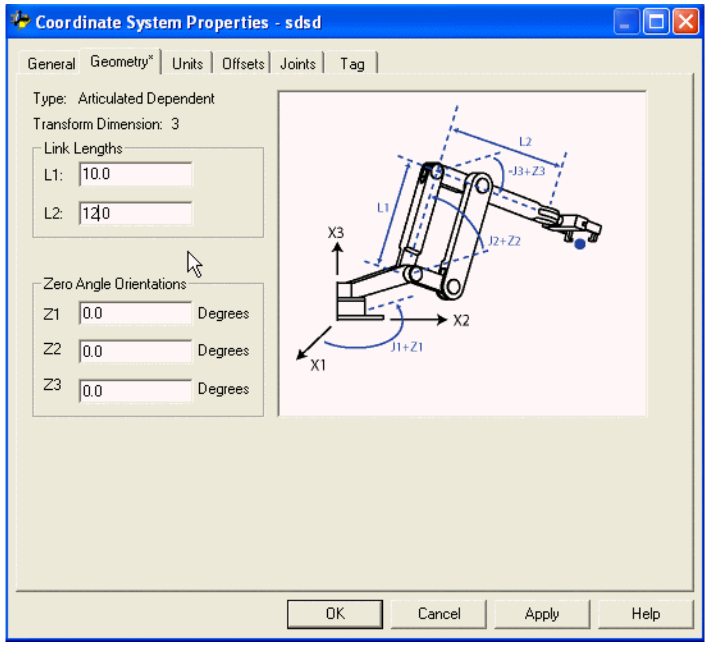
For an articulated dependent robot with	The length of	Is equal to the value of the distance between
2 dimensions	L1 L2	J1 and J2 J2 and the end-effector
3 dimensions	L1 L2	J2 and J3 J3 and the end-effector

Enter the link lengths on the **Geometry** tab in the **Coordinate System Properties** dialog box.

Type the **Link Length** values.

The **Link Length** values in this example are:

- L1 = 10.0
- L2 = 12.0

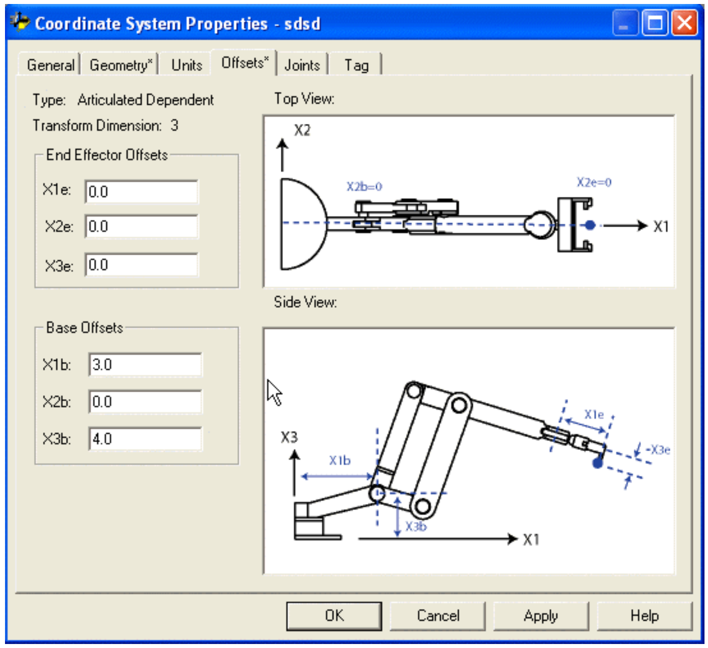


See also

Configuration parameters for Articulated Dependent robot on page 112

Base offsets for Articulated Dependent J1J2J3 robots

The Base Offsets are a set of coordinate values that redefine the origin of the robot. The correct base-offset values are typically available from the robot manufacturer. Type the values for the Base Offsets in the **X1b** and **X3b** boxes on the **Geometry** tab in the **Coordinate System Properties** dialog box.



See also

Configuration parameters for Articulated Dependent robot on page 112

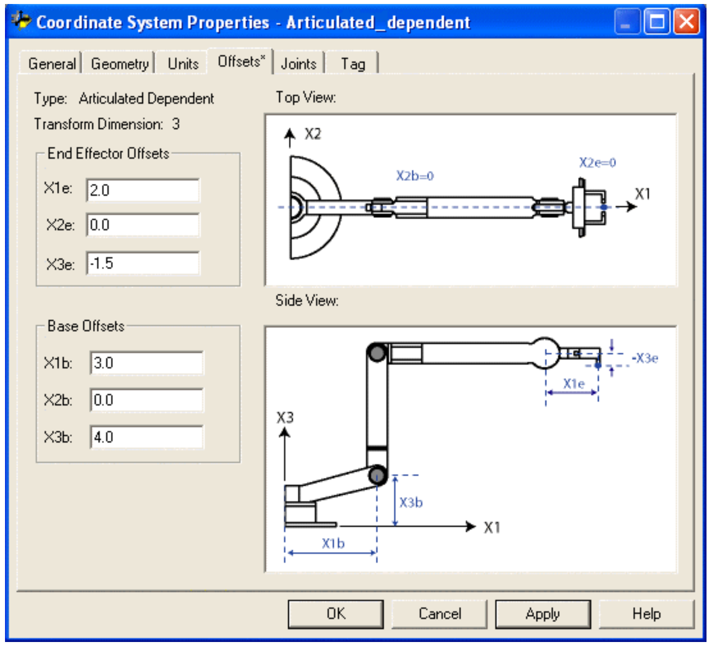
Link lengths for Articulated Dependent robot on page 112

End-Effector Offsets for Articulated Dependent robot on page 114

End-Effector Offsets for Articulated Dependent J1J2J3 robots

The robot can have an end effector attached to the end of robot link L2. If there is an attached end effector, configure the **End-Effector Offset** value on the **Offsets** tab in the **Coordinate System Properties** dialog box. The **End-Effector Offsets** are defined with respect to the tool reference frame at the tool tip.

Some robots also have an offset defined for the J3 joint. Account for this value when computing the **X3e** end effector offset value. If the value for **X3e** offset is entered as the sum of **X3e1+X3e2 (-3+1.5 = -1.5)**, the configured value for **X3e** is **-1.5**.



See also

[Configuration parameters for Articulated Dependent robot](#) on [page 112](#)

[Link lengths for Articulated Dependent robot](#) on [page 112](#)

[Base offsets for Articulated Dependent robot](#) on [page 114](#)

Configure an Articulated Dependent J1J2J3J6 robot

The typical Articulated Dependent J1J2J3J6 robot has four revolute joints: J1, J2, J3, and J6.

WARNING: Before turning on the transform or establishing the reference frame, or both, do the following actions for the joints of the target coordinate system:

- Set and enable the soft travel limits.
- Enable the hard travel limits.

Failure to perform these steps can cause the robotic arm to move to unexpected positions causing machine damage, or injury or death to personnel.

See also

[Soft and hard travel limit adjustments](#) on [page 124](#)

[Reference frame for Articulated Dependent J1J2J3J6 robots](#) on [page 116](#)

[Commission an Articulated Dependent J1J2J3J6 robot](#) on [page 117](#)

[Work envelope for Articulated Dependent J1J2J3J6 robots](#) on [page 121](#)

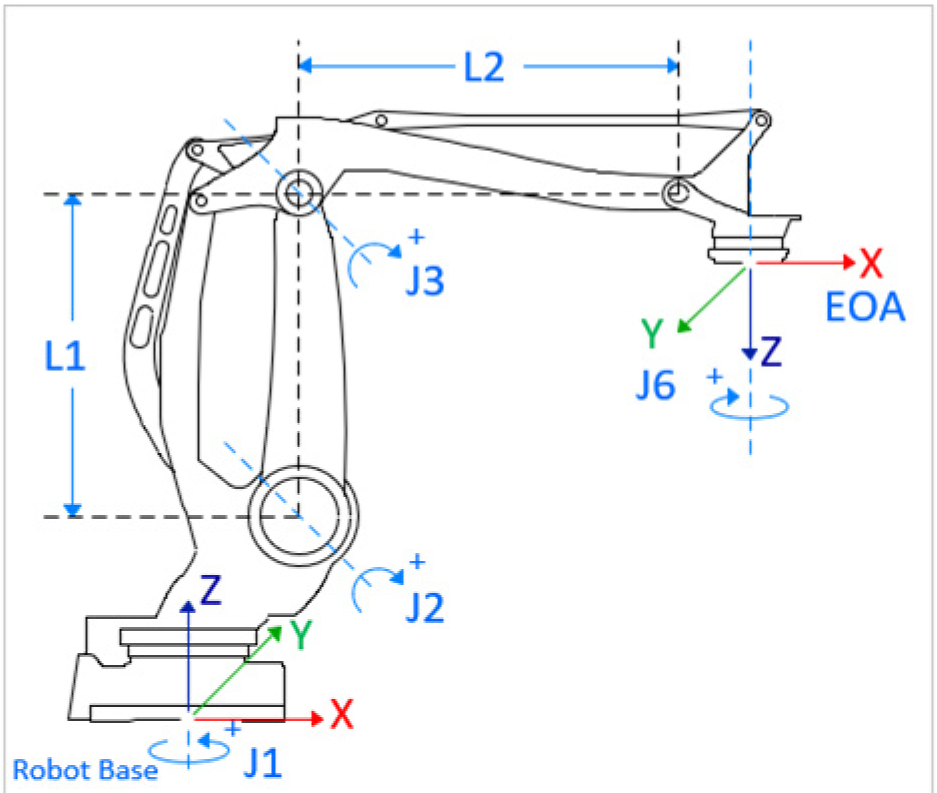
Reference frame for Articulated Dependent J1J2J3J6 robots

The robot reference frame is located at the base of the robot. The reference frame is the Cartesian, or source, coordinate frame that defines the origin and the primary axes X, Y, and Z. The primary axes make up the measurements for the Cartesian position of a target with reference to the robot base frame.

WARNING: Failure to properly establish the correct reference frame for the robot can cause the robotic arm to move to unexpected positions causing machine damage, or injury or death to personnel.

This side view of the robot shows:

- L1 and L2 are the rigid members of the robot, connecting all joints.
- J1 is connected to the base.
- J2 is perpendicular to joint J1 and connects to the shoulder.
- J3 is at the shoulder.
- J6 is at the wrist, which is at the end of link L2. The J6 position on the robot is the End of Arm (EOA).



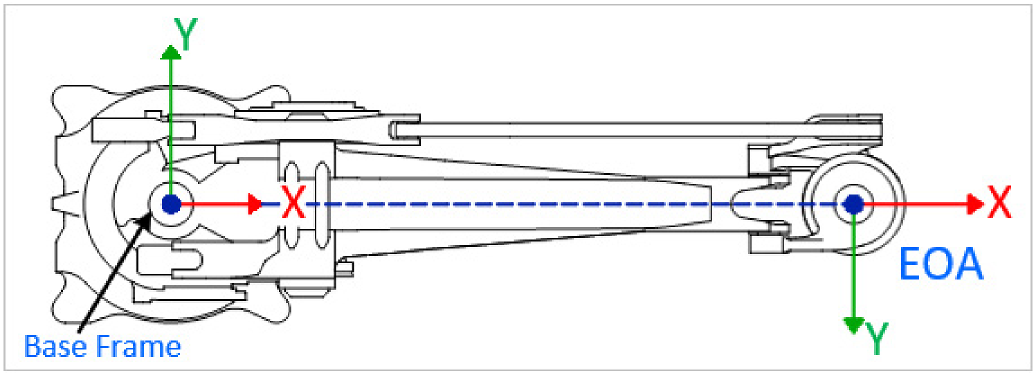
The joint angle measurements:

- +J1 is measured counterclockwise around +Z-axis of the base frame.
- +J2 is measured clockwise around +Y-axis of the base frame. When J2 = 0, link L1 is perpendicular to the XY plane.
- +J3 is measured clockwise around +Y-axis of the base frame. When J2=0 and J3 =0, link L2 is parallel to the XY plane.
- +J6 is measured clockwise around the +Z-axis at the EOA frame.

Base frame

The reference XYZ frame, or base frame, for an articulated geometry is located near the center of the base plate, which connects with joint J1. When you configure an Articulated Dependent J1J2J3J6 coordinate system in the Logix Designer application, with the joints homed as 0° in the XY plane of the robot base frame, the L2 link is aligned along the X positive axis.

This illustration shows the top view of the robot and the X and Y-axes for the base and EOA frames.



End of Arm frame

The EOA frame in the XYZ reference frame is set at the end of link L2. This frame rotates by Rx = 180° with reference to the base frame. As a result, the X-axis is in the same direction as the base frame X-axis, but the Z-axis direction points down, toward the direction of the Tool approach vector. The J6 axis of rotation aligns with the Z-axis of the EOA frame.

To set the home position for the J6 axis, move the J6 axis so that the X-axis of EOA is aligned with link L1, or the X-axis of the base frame.

See also

Work envelope for Articulated Dependent J1J2J3J6 robots on page 121

Maximum joint limits for Articulated Dependent J1J2J3J6 robots on page 123

Soft and hard travel limit adjustments on page 124

Commission an Articulated Dependent J1J2J3J6 robot

Follow these steps to commission an Articulated Dependent J1J2J3J6 robot.

To commission an Articulated Dependent J1J2J3J6 robot

1. Get the angle values from the robot manufacturer for joints J1, J2, J3, and J6 at the calibration position. Use these values to establish the zero position.
2. Refer to the manufacturer's data sheet to determine if the associated sized motor contains an internal or external gearbox from the motor to actuation, at the links or joints, to move the robot.
3. Open the **Axis Properties** and then select the **Scaling** tab.

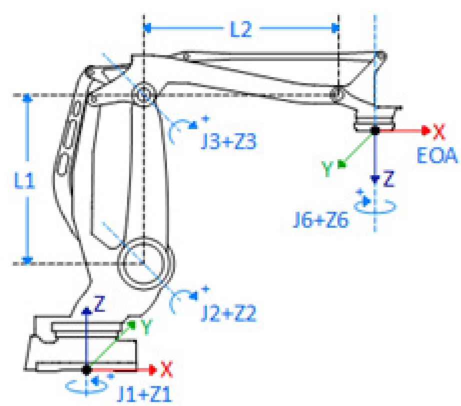
a. In **Controller Organizer**, expand the **Motion Groups** folder, right-click the axis and then select **Properties**.

b. In **Axis Properties**, in **Categories**, select **Scaling**.

4. In **Transmission Ratio I/O**, set the gear ratio for each axis.
5. In **Scaling**, enter the scaling to apply to axes J1, J2, and J3 so that one revolution equals 360°.

Tip: Axis J6 is a rotary axis, and the units are defined in motor revolutions. If necessary, use the manufacturer's data sheet to convert units into motor revolutions.

6. Move all joints to the zero position by jogging the robot under programmed control, or manually moving the robot when the joint axes are in an open-loop state.
7. Do one of these steps to set zero positions for the axes:
   - Use the Motion Redefine Position (MRP) instruction to set the positions of the joint axes to the calibration values obtained in step 1.
   - Set the configuration value for the joint axes home position to the calibration values obtained in step 1 and execute a Motion Axis Home (MAH) instruction for each joint axis.
8. Move each joint (J1, J2, J3, and J6) to an absolute position of 0.0. Verify that each joint position reads 0 and the respective L1 and L2 members are aligned. This step establishes the X-axis of the robot base frame for transformations.



Tip: If you prefer a reference position for the axes that is different from the transform position zero, you can use zero angle offsets to adjust the positions for axes J1, J2, J3, and J6.

9. Move J6 to an absolute position of 0.0. Verify that joint position reads 0 and that the J6 position is in the Z-axis direction of the End of Arm (EOA) Frame.

Tip: The robot axes are absolute, so you probably will establish the zero positions only once. Re-establish the zero positions if you change the controller or lose them.

See also

Configuration type for Articulated Dependent J1J2J3J6 robots on page 119

Configuration parameters for Articulated Dependent J1J2J3J6 robots on page 126

Configuration type for Articulated Dependent J1J2J3J6 robots

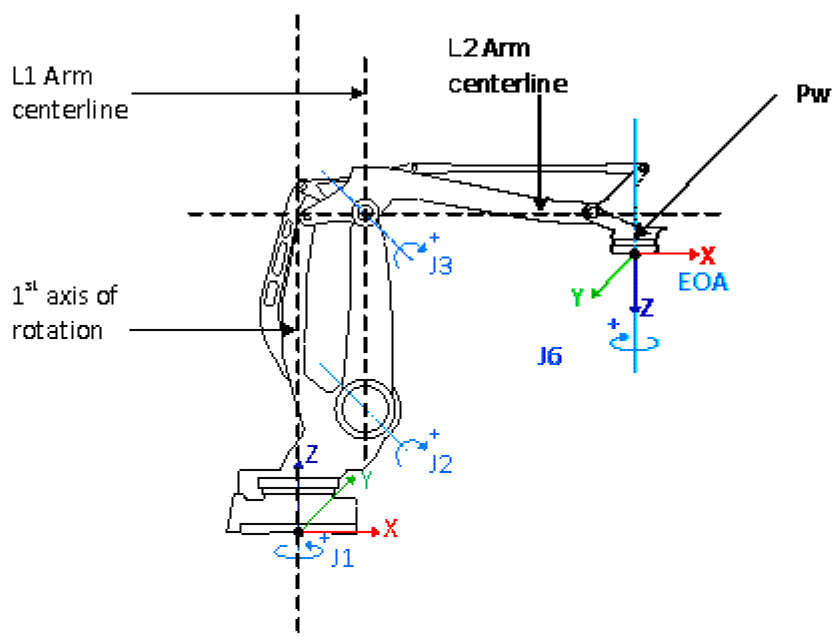
Articulated Dependent J1J2J3J6 robots support only the Lefty-Above-Non-Flip robot configuration.

For more information on robot configuration types, see Determine the coordinate system type on page 35.

In Articulated Dependent J1J2J3J6 robots:

- The wrist point of the robot (Pw) does not cross the first axis of rotation, so the robot is always in a Lefty configuration.
- The point Pw is always below the L1 arm centerline, so the robot is always in an Above configuration.
- The End of Arm (EOA) is always pointing down because of the linkage (toward the negative Z axis of the robot base). The point Pw does not cross the L2 arm centerline, so the robot is always in a Non-flip configuration.

This illustration shows the wrist point and the L1 and L2 arm centerlines.



Configuration type in MCPM instructions

The Motion Coordinated Path Move (MCPM) instruction accepts only the Lefty-Above and Non-Flip configuration type. Other configuration types cause error 136 (MCPM_ROBOT_CONFIGURATION_CONFLICT).

Configuration type in MCTPO instructions

In the Motion Calculate Transform Position with Orientation (MCTPO) instruction, the robot configuration is either an input or output parameter depending on the transformation direction.

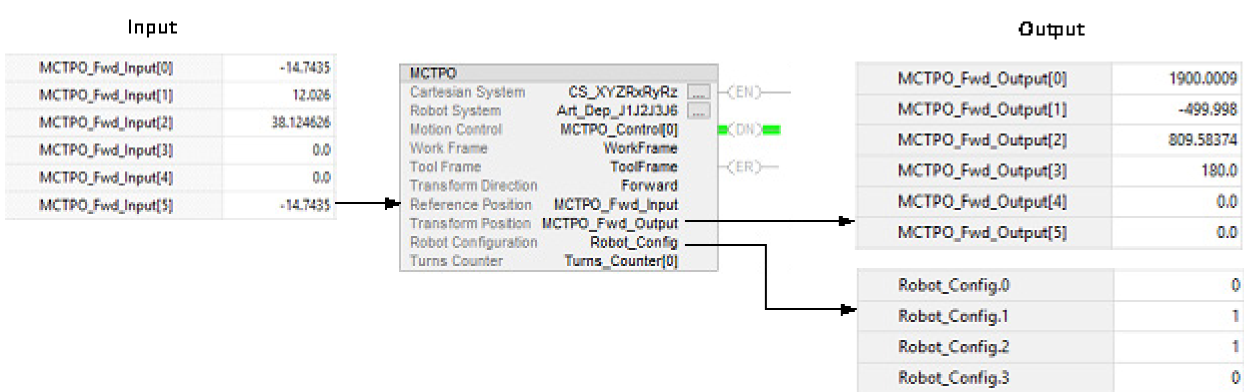
- If the Transform Direction is set to Forward, the instruction returns the robot configuration in the tag data.
- If the Transform Direction is set to Inverse, the instruction requires the user to provide the robot configuration as an input tag data.

This table lists the bit positions and corresponding robot configurations in the MCTPO instruction.

Bit position	3	2	1	0
Description	Flip (1)/ No Flip (0)	Above (1)/ Below (0)	Lefty (1)/ Righty (0)	Change (1)/ Same (0)
Robot configuration	0	1	1	(Ignored in MCTPO)

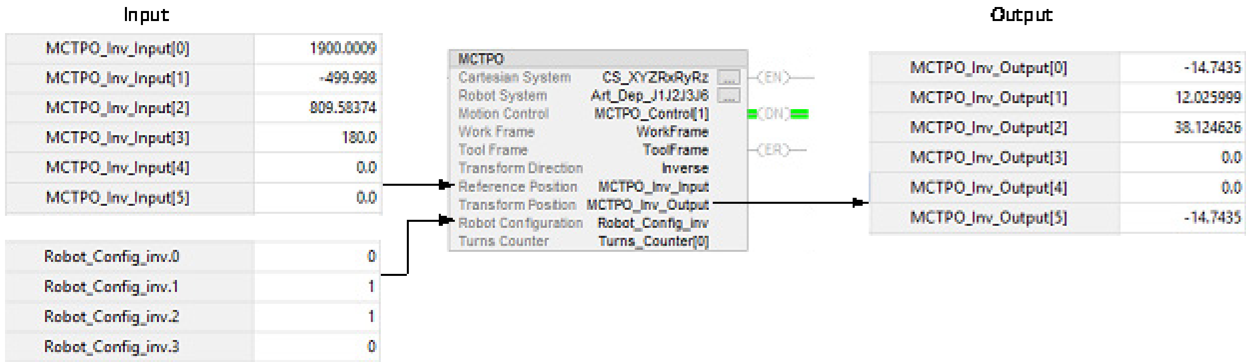
These ladder logic examples show the operation of the MCTPO instruction when Transform Direction is set to Forward and Inverse.

- This example illustrates the MCTPO instruction with Transform Direction set to Forward. The target positions configured are provided to Reference position operand as the input. The instruction computes the corresponding Cartesian positions and Robot Configuration as the output. In this example, the target positions are evaluated as Lefty-Above-Non-Flip configuration.



- This example illustrates the MCTPO instruction with Transform Direction set to Inverse, where the user provides the Cartesian Position and Robot Configuration for Lefty-Above-Non-Flip as input.

The instruction computes the corresponding target joint angle positions and is written to the Transform Position parameter as the output.



See also

Determine the Coordinate System type on page 35

Reference frame for Articulated Dependent J1J2J3J6 robots on page 116

Configuration parameters for Articulated Dependent J1J2J3J6 robots on page 126

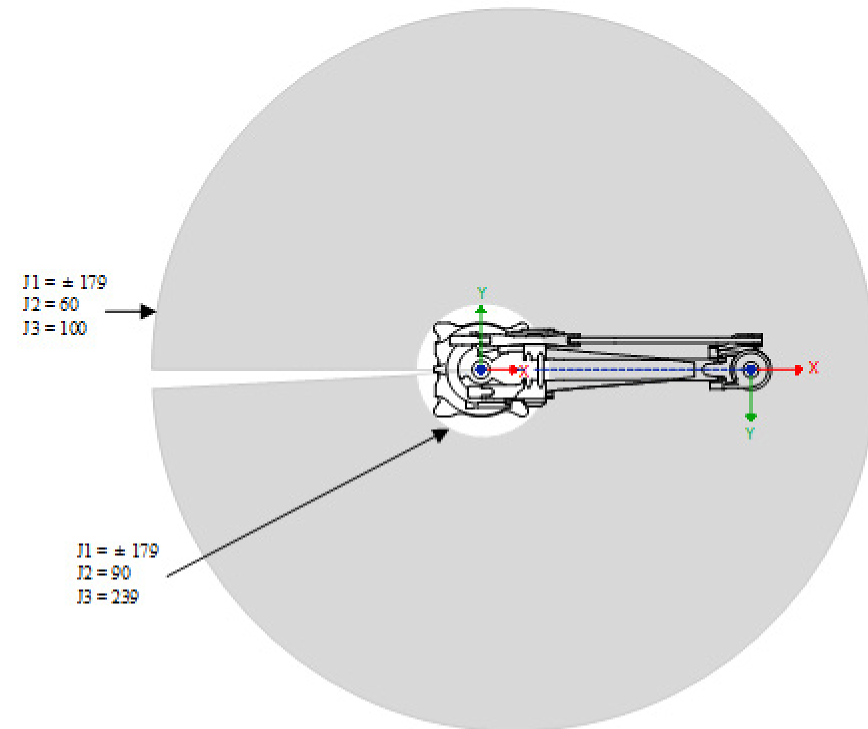
Work envelope for Articulated Dependent J1J2J3J6 robots

The work envelope is the three-dimensional region of space that defines the reaching boundaries of the robot arm.

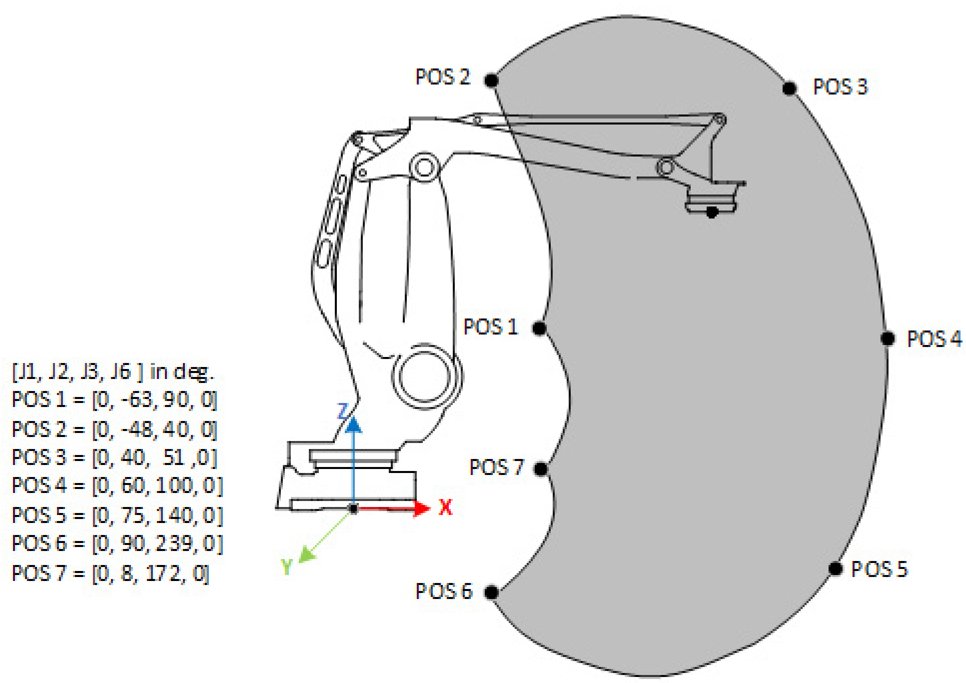
The work envelope for an Articulated Dependent J1J2J3J6 robot looks like a sphere, with a travel limit of axes J2 and J3 along the Z- axis. Due to the range of motion limitations on individual joints, the work envelope might not be a complete sphere.

IMPORTANT	The work envelope for the Articulated Dependent J1J2J3J6 robot varies when a tool is attached to the robot. The tool shape and dimensions can modify the work envelope.

This illustration shows a top view of the typical work envelope for an Articulated Dependent J1J2J3J6 robot with limited target points.



This illustration shows a side view of the typical work envelope for an Articulated Dependent J1J2J3J6 robot with limited target points.



See also

Maximum joint limits for Articulated Dependent J1J2J3J6 robots on page 123

Soft and hard travel limit adjustments on page 124

Reference frame for Articulated Dependent J1J2J3J6 robots on page 116

Maximum joint limits for Articulated Dependent J1J2J3J6 robots

This table lists the maximum and minimum joint limits for Articulated Dependent J1J2J3J6 robots, and the errors that are reported when the limits are exceeded.

Axis	Joint limit	Error reported
J1	+/-179°	The Motion Coordinated Transform (MCT) instruction reports error code 151 (Joint Angle Beyond its Limit) with extended error code 1 (Joint J1 Beyond Limit) when Joint J1 exceeds the limit.
J2	+/-179°	The MCT instruction reports error code 151 (Joint Angle Beyond its Limit) with the extended error code 2 (Joint J2 Beyond Limit) when Joint J2 exceeds the limit.
J3	0 to 359°	The MCT instruction reports error code 151 (Joint Angle Beyond its Limit) with extended error code 3 (Joint J3 Beyond Limit) when Joint J3 exceeds the limit.
J6	-45899.99 to 45900.00	The Joint 6 (J6) axis is the rotational axis that could have multiple turns. The value for maximum number of turns is +/-127.

The MCT instruction monitors target positions that keep the robot from becoming fully stretched or from folding back on itself at or near the origin of the coordinate system.

For positions close or near the origin, the MCT instruction reports error 67 (Invalid Transform Position) and error 69 (Max Joint Limit Velocity Exceeded) for singularity positions.

The valid robot configuration for an Articulated Dependent J1J2J3J6 robot is always Lefty, Above, and Non-flip. Any other configuration generates error code 137: *The robot configuration parameter for the Motion Coordinated Transform instruction is not valid for this Robot geometry*.

See also

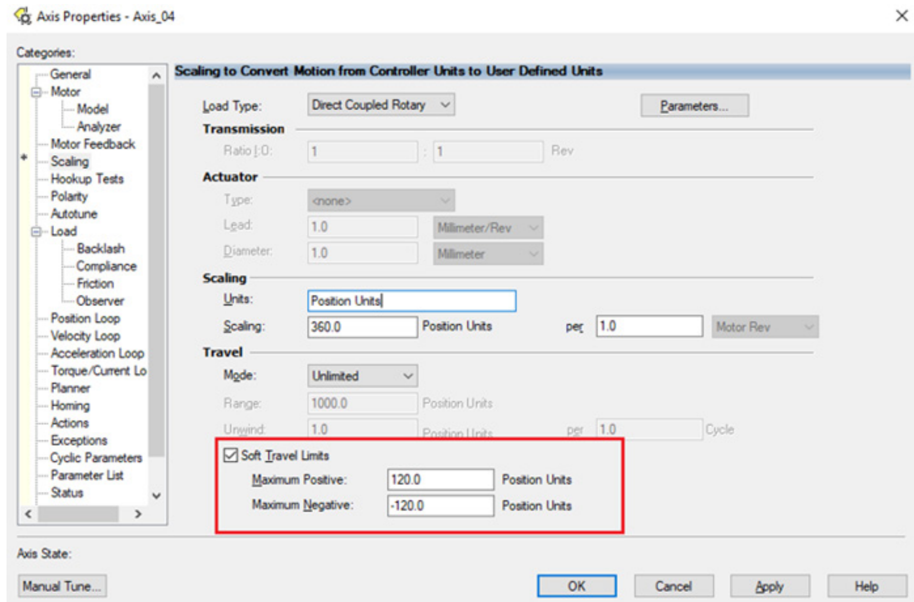
Soft and hard travel limit adjustments on page 124

Work envelope for Articulated Dependent J1J2J3J6 robots on page 121

Reference frame for Articulated Dependent J1J2J3J6 robots on page 116

Configuration type for Articulated Dependent J1J2J3J6 robots on page 119

Soft and hard travel limit adjustments



Use either soft travel limits or hard travel limits to configure joint limits for joint axes on Articulated Dependent J1J2J3J6 robots.

To adjust soft travel limits

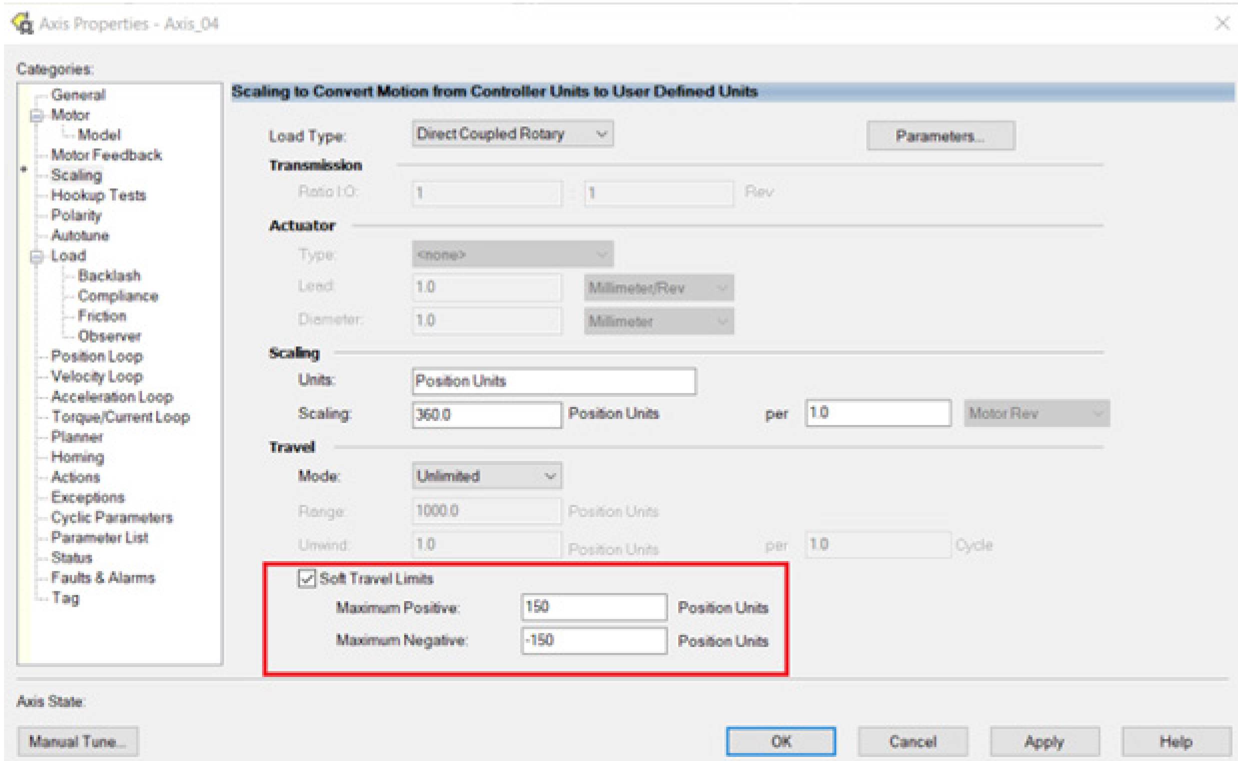
1. In **Axis Properties,** select the **Scaling** tab.

a. In the **Controller Organizer**, expand the **Motion Groups** folder, and then double-click the axis.

b. Select the **Scaling** tab.

2. Select **Soft Travel Limits**.
3. Enter the maximum positive and maximum negative limit values based on the mechanical limits of the joint axis. If the axis moves beyond the travel limits, the Software Positive/Negative Overtravel fault occurs.

This illustration shows the Soft Travel Limits settings.



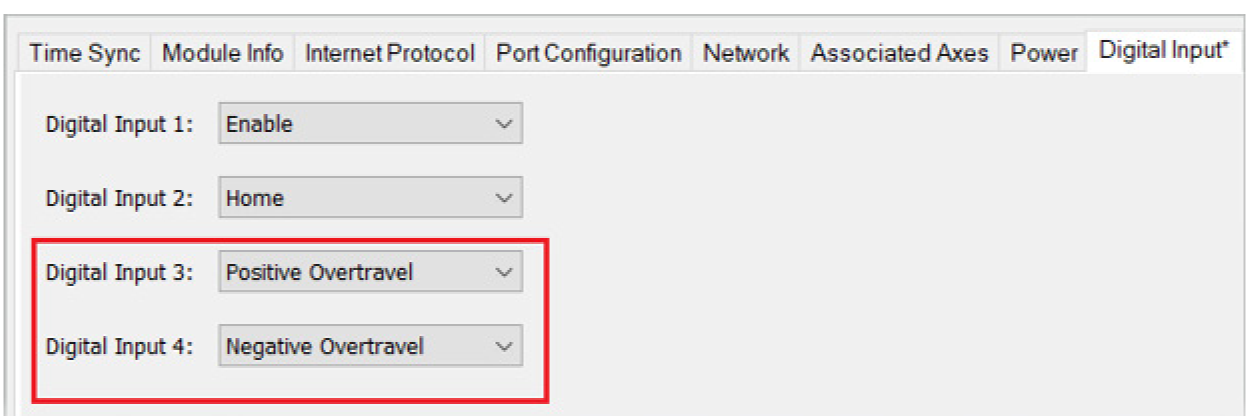
Using Hard Travel Limits

The hard travel limit uses limit-switch sensors to help prevent the axis from moving beyond the current position limits. Hardware overtravel limit switches mounted on the equipment establish the limits.

If the axis moves beyond the hard overtravel limit switch, the PosHardOvertravelFault/NegHardOvertravelFault occurs. The fault can only occur when the drive is in the enabled state and the Hard Overtravel Checking bit is set in the Fault Configuration Bits attribute.

To enable the Positive Overtravel and Negative Overtravel limits, use the **Digital Input** tab in the **Drive Properties** dialog.

This illustration shows the Overtravel settings.



See also

Maximum joint limits for Articulated Dependent J1J2J3J6 robots on page 123

Work envelope for Articulated Dependent J1J2J3J6 robots on page 121

Reference frame for Articulated Dependent J1J2J3J6 robots on page 116

Work and tool frame offset limits for Articulated Dependent J1J2J3J6 robots

Work Frame offsets locate the user work frame of the robot relative to the origin of the robot base frame. These offsets consist of an XYZ and RxRyRz value.

Tool frame offsets locate the tool center relative to the center of the End of Arm (EOA). These offsets consist of an XYZ and RxRyRz value.

The target end position range changes based on the work and tool frame offsets.

The following offset values are allowed for work and tool frames. The Motion Coordinated Transform (MCT) instruction generates error 148 for invalid offset values.

- Offset values on the X, Y, Z and Rz axes are allowed for the Work Frame offsets. Rx and Ry offsets are restricted and must be set to 0.
- Offset values on the X, Y, Z and Rz axes are allowed for the Tool Frame offsets. Rx and Ry offsets are restricted and must be set to 0.

See also

Work envelope for Articulated Dependent J1J2J3J6 robots on page 121

Maximum joint limits for Articulated Dependent J1J2J3J6 robots on page 123

Soft and hard travel limit adjustments on page 124

Configuration parameters for Articulated Dependent J1J2J3J6 robots

Configure these parameters for Articulated Dependent J1J2J3J6 robots with varying reach and payload capacities:

- Link lengths
- Zero-angle orientation
- Base offsets
- End-effector offsets

The configuration parameter information is available from the robot manufacturer.

Tip: For Articulated Dependent J1J2J3J6 robots, the **Dimension** and **Transform Dimension** values on the **Coordinate System Properties** dialog box - **General** tab are automatically set to 4 when you select **J1J2J3J6** as the **Coordinate Definition**. You cannot change the Dimension settings.

See also

Link lengths for Articulated Dependent J1J2J3J6 robots on page 127

Zero Angle Orientations for Articulated Dependent J1J2J3J6 robots on page 128

Base offsets for Articulated Dependent J1J2J3J6 robots on page 130

End-Effector Offsets for Articulated Dependent J1J2J3J6 robots on page 131

Work and tool frame offset limits for Articulated Dependent J1J2J3J6 robots on page 126

Link lengths for Articulated Dependent J1J2J3J6 robots

Links L1 and L2 are the rigid members of the robot joints.

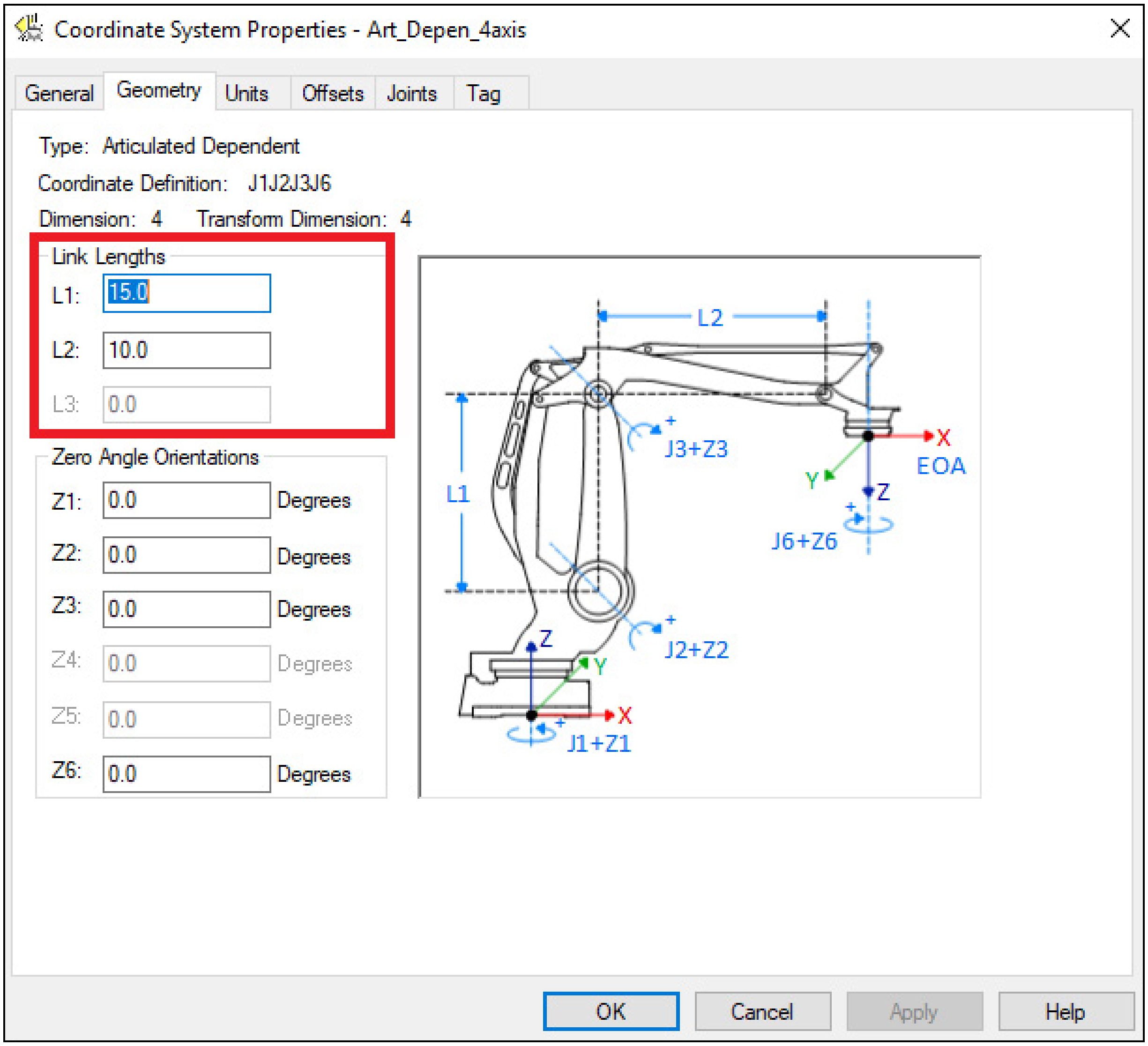
Use the **Geometry** tab on the **Coordinate System Properties** dialog to configure link lengths L1 and L2.

To open the **Coordinate System Properties** dialog, in the **Controller Organizer**, expand the **Motion Groups** folder, right-click the axis and then select **Properties**.

This example shows link length values as:

- L1 = 15.0

- L2 = 10.0



See also

[Zero Angle Orientations for Articulated Dependent J1J2J3J6 robots](#) on [page 128](#)

[Base offsets for Articulated Dependent J1J2J3J6 robots](#) on [page 130](#)

[End-Effector Offsets for Articulated Dependent J1J2J3J6 robots](#) on [page 131](#)

Zero Angle Orientations for Articulated Dependent J1J2J3J6 robots

The zero-angle orientation is the rotational offset of the individual joint axes.

For Articulated Dependent J1J2J3J6 robot geometry, the internal transformation equations in the Logix Designer application assume that:

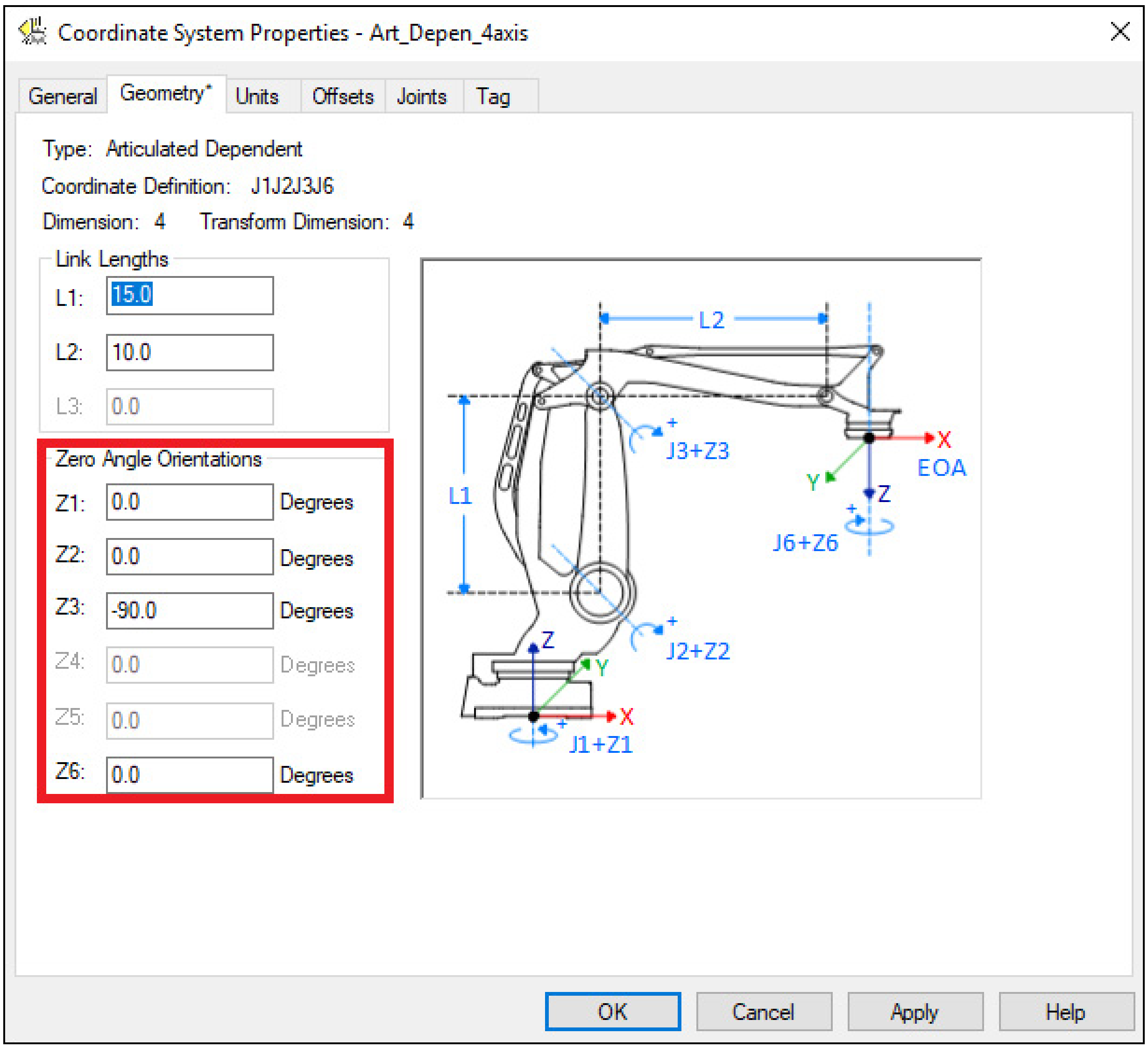
- Joints J1, J2, J3 and J6 are homed to 0°.

- The J6 axis of rotation is aligned with the Z-axis of the End of Arm (EOA) frame (Z-axis of EOA frame pointing down with respect to the base frame).

To set the angular positions for joints J1, J2, J3 and J6 to any value other than 0, configure the zero-angle orientation values on the **Geometry** tab in the **Coordinate System Properties** dialog box to align the joint angle positions with the internal equations.

To open the **Coordinate System Properties** dialog, in the **Controller Organizer**, expand the **Motion Groups** folder, right-click the axis and then select **Properties**.

For example, to set the joint J3 axis position to 0° instead of 90°, enter -90° for the **Z3** parameter. This illustration shows the J3 axis position set to 0°.



See also

Link lengths for Articulated Dependent J1J2J3J6 robots on page 127

Base offsets for Articulated Dependent J1J2J3J6 robots on page 130

End-Effector Offsets for Articulated Dependent J1J2J3J6 robots on page 131

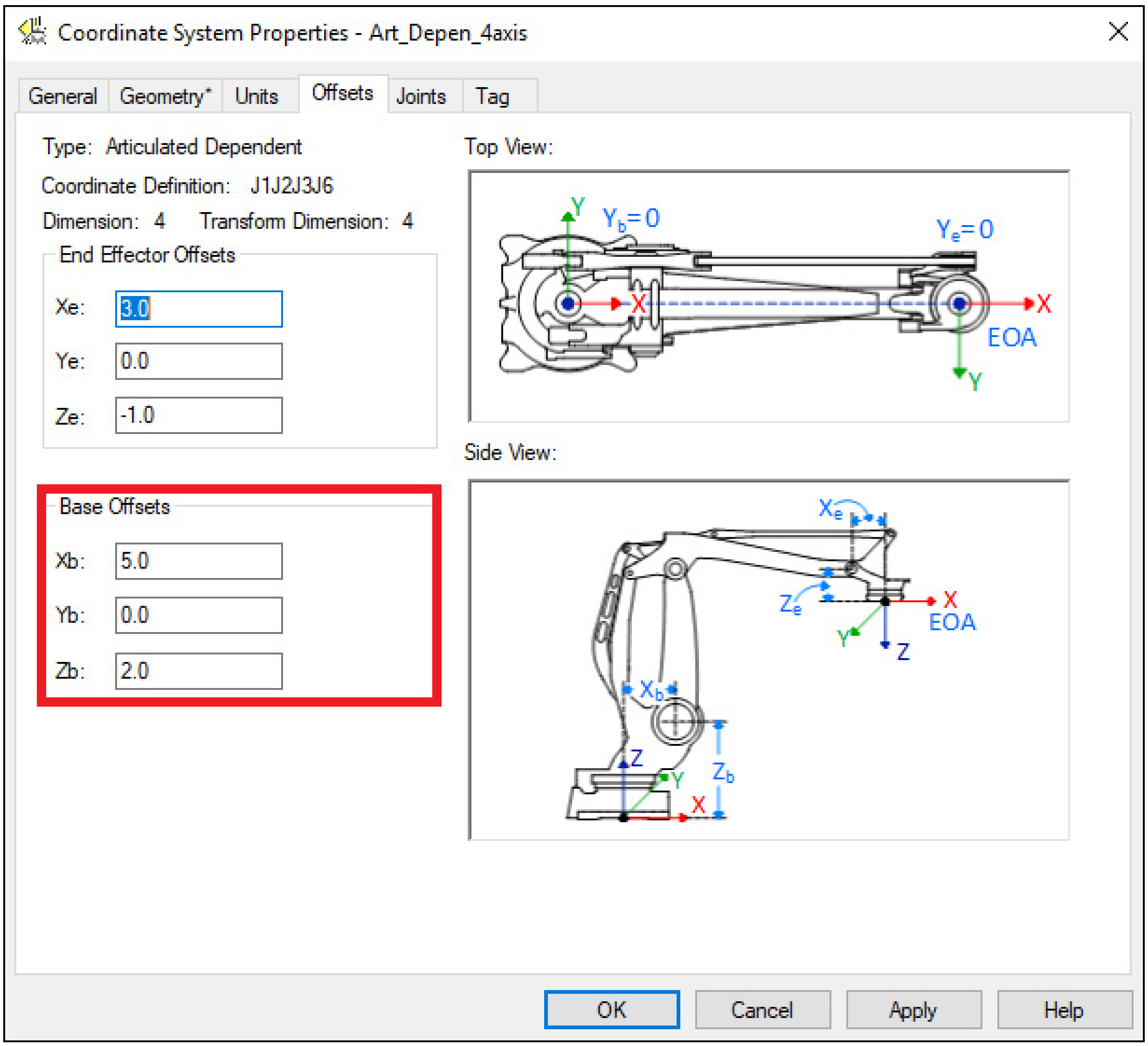
Base offsets for Articulated Dependent J1J2J3J6 robots

Base offsets are a set of coordinate values that define the offset between the robot base and joint J2. The correct base offset values should be available from the robot manufacturer.

Configure the values for the base offsets in the **Xb**, **Yb**, and **Zb** boxes on the **Offsets** tab in the **Coordinate System Properties** dialog.

To open the **Coordinate System Properties** dialog, in the **Controller Organizer,** expand the **Motion Groups** folder, right-click the axis and then select **Properties**.

This illustration shows the base offsets on the **Offsets** tab.



See also

Link lengths for Articulated Dependent J1J2J3J6 robots on page 127

Zero Angle Orientations for Articulated Dependent J1J2J3J6 robots on page 128

End-Effector Offsets for Articulated Dependent J1J2J3J6 robots on page 131

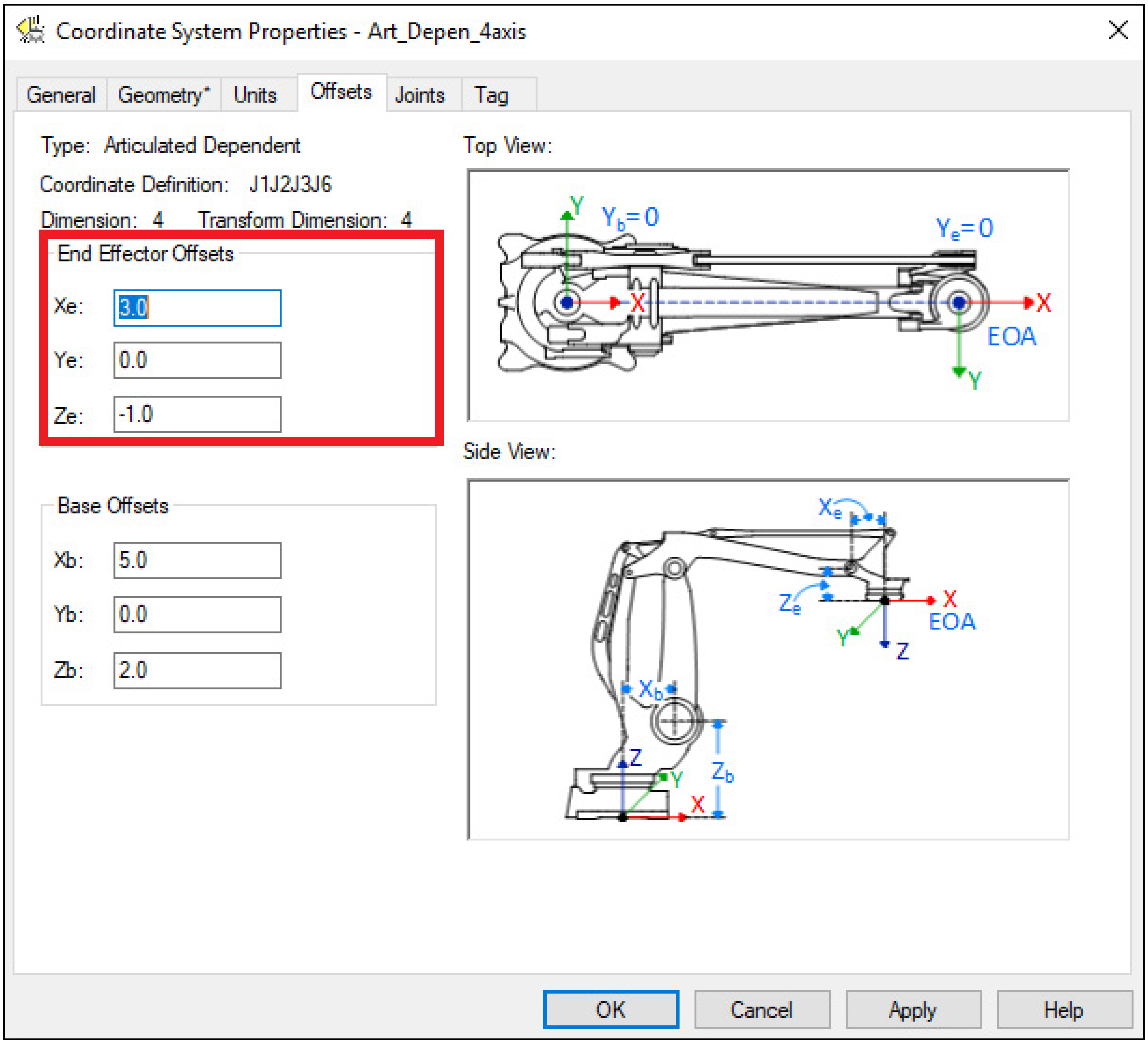
End-effector offsets for Articulated Dependent J1J2J3J6 robots

The end-effector offsets set coordinate values that define the offset between the end of link L2 and the End of Arm (EOA). Xe, Ye, and Ze are radial offsets and do not change because of an attached tool.

Configure the values for the end-effector offsets in the **Xe**, **Ye**, and **Ze** boxes on the **Offsets** tab in the **Coordinate System Properties** dialog.

To open the **Coordinate System Properties** dialog,in the **Controller Organizer**, expand the **Motion Groups** folder, right-click the axis and then select **Properties**.

This illustration shows the end-effector offsets on the **Offsets** tab.



See also

[Link lengths for Articulated Dependent J1J2J3J6 robots](#) on [page 127](#)

[Zero Angle Orientations for Articulated Dependent J1J2J3J6 robots](#) on [page 128](#)

[Base offsets for Articulated Dependent J1J2J3J6 robots](#) on [page 130](#)

Arm solutions

A kinematic arm solution is the position of all joints on the robot that correspond to a Cartesian position. When the Cartesian position is inside the workspace of the robot, then at least one solution always exists. Many of the geometries have multiple joint solutions for a single Cartesian position.

- Two axis robots - two joint solutions typically exist for a Cartesian position.

- Three axis robots - four joint solutions typically exist for a Cartesian position.

See also

Left-arm and right-arm solutions for two-axes robots on page 133

Solution mirroring for three-dimensional robots on page 133

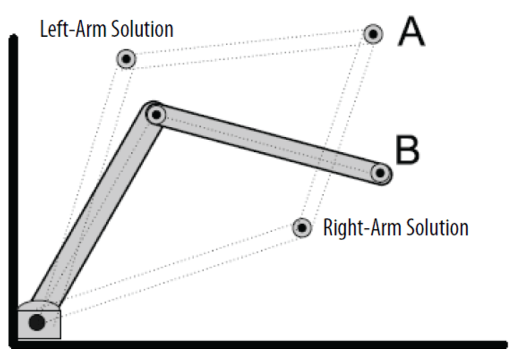
Change the robot arm solution on page 134

Plan for singularity on page 135

Encounter a no-solution position on page 136

Left-arm and right-arm solutions for two-axes robots

A robot having an arm configuration has two kinematics solutions when attempting to reach a given position. Point A is shown in the following illustration. One solution satisfies the equations for a right-armed robot, the other solution satisfies the equations for a left-armed robot.



See also

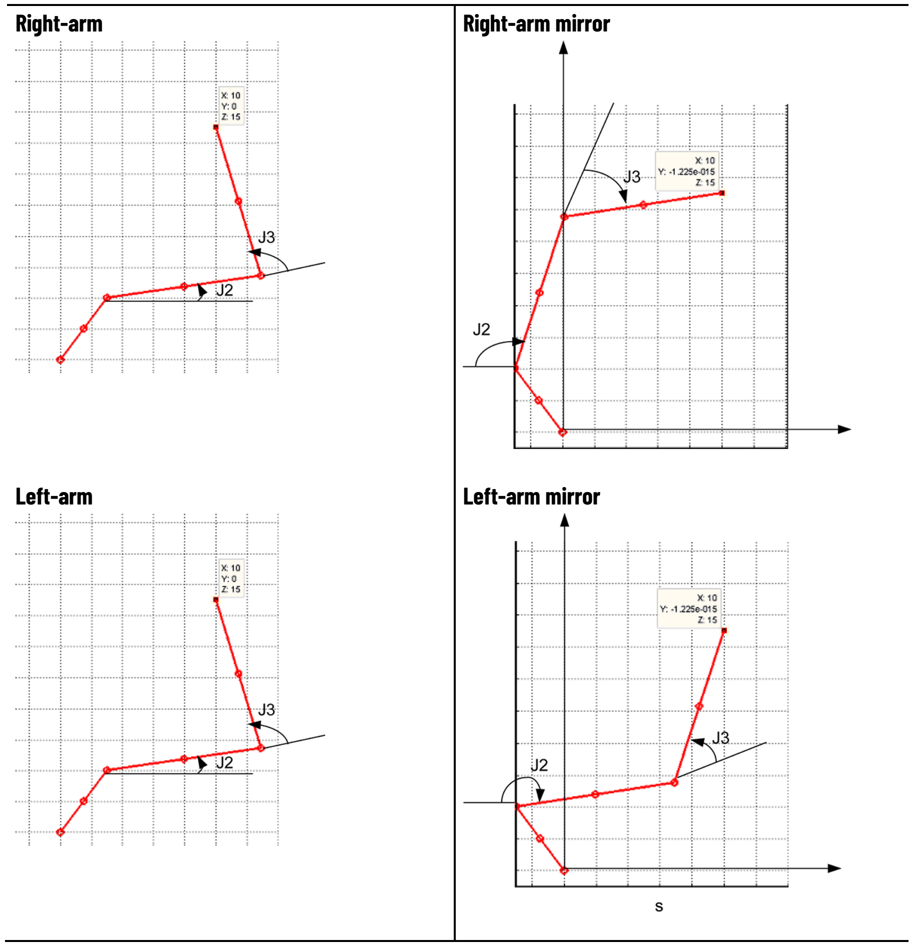
Arm solutions on page 132

Solution mirroring for three-dimensional robots

For a three-dimensional Articulated Independent robot, there are four solutions for the same point:

- Left-arm
- Right-arm
- Left-arm mirror
- Right-arm mirror

For example, consider the Cartesian point XYZ (10,0,15). The joint position corresponding to this point has four joint solutions. Two of the solutions are the same as the solutions for the two-dimensional case. The other solutions are mirror image solutions where J1 is rotated 180°.



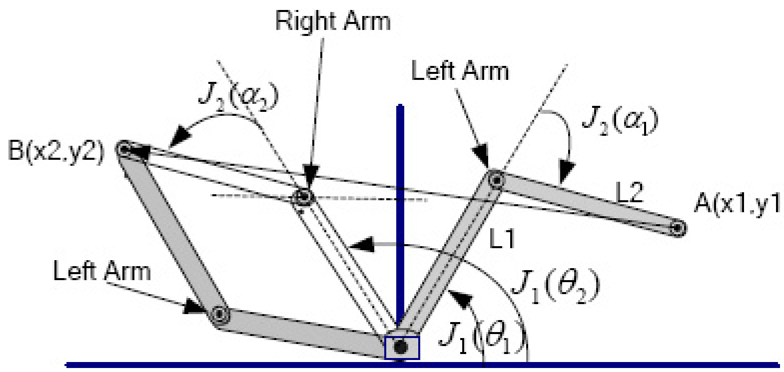
See also

Arm solutions on page 132

Change the robot arm solution

You can switch the robot from a left-arm solution to a right-arm solution or vice versa. This is done automatically when a joint move is programmed forcing a left/right change to occur. After the change is performed, the robot stays in the new arm solution when Cartesian moves are made. If required, the robot arm solution changes again when another joint move is made.

Example: Suppose, you want to move the robot from position A (x1,y1) to position B (X2,Y2) as shown in the following figure. At position A, the system is in a left arm solution. When programming a Cartesian move from A (X1,Y1) to B (X2,Y2), the system moves along the straight line from A to B while maintaining a left arm solution. If you want to be at position B in a right-arm solution, you must make a joint move in J1 from θ1 to θ2 and a joint move in J2 from α1 to α2.



See also

Arm solutions on page 132

Plan for singularity

A singularity occurs when an infinite number of joint positions (mathematical solutions) exist for a given Cartesian position. The Cartesian position of a singularity is dependent on the type of the robot geometry and the size of the link lengths for the robot. Not all robot geometries have singularity positions.

For example, singularities for an Articulated Independent robot occur when:

- The robot manipulator folds its arm back onto itself and the Cartesian position is at the origin.
- The robot is fully stretched at or very near the boundary of its workspace.

An error condition is generated when a singularity position is reached.

WARNING: Avoid programming the robot towards a singularity position when programming in Cartesian mode. The velocity of the robot increases rapidly as it approaches a singularity position and can result in injury or death to personnel.

See also

Arm solutions on page 132

Encounter a no-solution position

When a robot is programmed to move beyond its work envelope, there is no mathematical joint position for the programmed Cartesian position. The system forces an error condition.

For example, if an Articulated Independent robot has two 10-inch arms, the maximum reach is 20 inches. Programming to a Cartesian position beyond 20 inches produces a condition where no mathematical joint position exists.

WARNING: Avoid programming the robot towards a no-solution position when programming in Cartesian mode. The velocity of the robot increases rapidly as it approaches this position and can result in injury or death to personnel.

See also

Arm solutions on page 132

Configure Delta robot geometries

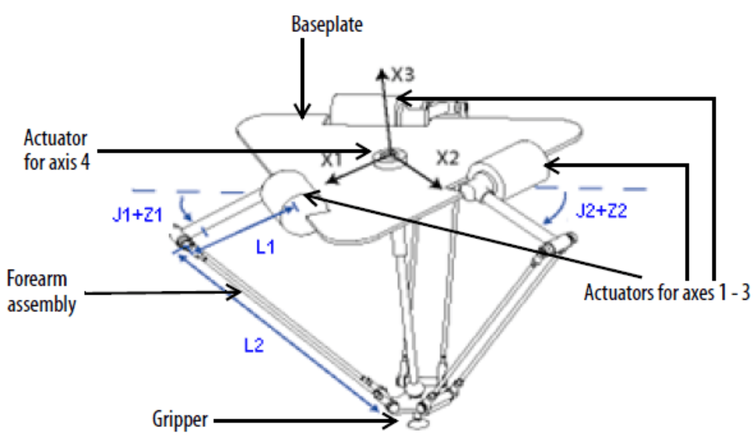
The Logix Designer application supports three types of geometries that are often called parallel manipulators.

- Three-dimensional Delta
- Two-dimensional Delta
- SCARA Delta

In these geometries, the number of joints is greater than the degrees of freedom, and not all the joints are actuated (motor driven). These un-actuated joints are typically spherical joints.

Configure a Delta Three-dimensional robot

This illustration shows a four axes Delta robot that moves in three-dimensional Cartesian (X1, X2, X3) space. This type of robot is often called a spider or umbrella robot.



The Delta robot in this illustration is a three-degree of freedom robot with an optional fourth degree of freedom used to rotate a part at the tool tip. In the Logix Designer application, the first three-degrees of freedom are configured as three joint axes (J1, J2, J3) in the robots coordinate system. The three joint axes are:

- Directly programmed in joint space.

- Automatically controlled by the embedded Kinematics software in the Logix Designer application from instructions programmed in a virtual Cartesian coordinate system.

This robot contains a fixed top plate and a moving bottom plate. The fixed top plate is attached to the moving bottom plate by three link-arm assemblies. All three of the link-arm assemblies have a single top link arm (L1) and a parallelogram two-bar link assembly (L2).

As each axis (J1, J2, J3) is rotated, the TCP of the gripper moves correspondingly in (X1, X2, X3) direction. The gripper remains vertical along the X3 axis while its position is translated to (X1, X2, X3) space by the mechanical action of the parallelograms in each of the forearm assemblies. The mechanical connections of the parallelograms via spherical joints ensures that the top and bottom plates remain parallel to each other.

Program the TCP to an (X1, X2, X3) coordinate, then the Logix Designer application computes the commands necessary for each of the joints (J1, J2 ,J3) to move the gripper linearly from the current (X1, X2, X3) position to the programmed (X1, X2, X3) position, at the programmed vector dynamics.

When each top link (L1) moves downward, its corresponding joint axis (J1, J2, or J3) is assumed to be rotating in the positive direction. The three joint axes of the robot are configured as linear axes.

To rotate the gripper, configure a fourth axis as a linear or rotary, independent axis.

See also

Establish the reference frame for a Delta Three-dimensional robot on page 138

Calibrate a Delta Three-dimensional robot on page 138

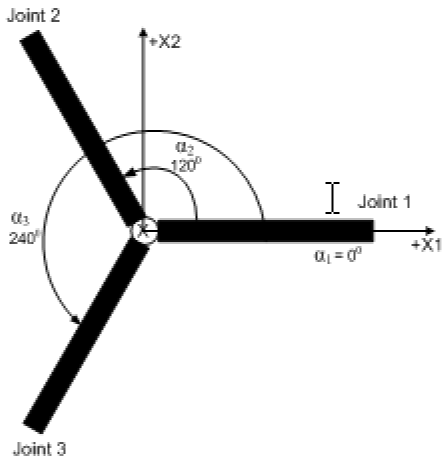
Configure Zero Angle Orientation for Delta Three-dimensional robot on page 139

Identify the Work Envelope for Delta Three-dimensional robot on page 140

Define Configuration Parameters for Delta Three-dimensional robot on page 142

Establish the reference frame for a Delta Three-dimensional robot

The reference frame for the Delta geometries is located at the center of the top fixed plate. Joint 1, Joint 2, and Joint 3 are actuated joints. If the Delta coordinate system in the Logix Designer application is configured with the joints homed at 0° in the horizontal position, then L1 of one of the link pairs will be aligned along the X1 positive axis as shown. Moving in the counter-clockwise direction from Joint 1 to Joint 2, the X2 axis will be orthogonal to the X1 axis. Based on the right hand rule, X3 positive will be the axis pointing up (out of the paper).



See also

Calibrate a Delta Three-dimensional robot on page 138

Calibrate a Delta Three-dimensional robot

Use these steps to calibrate the robot.

To calibrate a Delta Three-dimensional robot:

1. Obtain the angle values from the robot manufacturer for J1, J2, and J3 at the calibration position. Use these values to establish the reference position.
2. Move all joints to the calibration position by jogging the robot under programmed control or manually moving the robot when the joint axes are in an open loop state.
3. Do one of the following:

a. Use the Motion Redefine Position (MRP) instruction to set the positions of the joint axes to the calibration values obtained in step 1.

b. Set the configuration value for the joint axes home position to the calibration values obtained in step 1 and execute a Motion Axis Home (MAH) instruction for each joint axis.

4. Move each joint to an absolute position of 0.0. Verify that each joint position reads 0 degrees and the respective L1 is in a horizontal position.

   If L1 is not in a horizontal position, see the alternate method for calibrating a Delta three-dimensional robot.

See also

Alternate method for calibrating a Delta Three-dimensional robot on page 139

Alternate method for calibrating a Delta Three-dimensional robot

Rotate each joint to a position so that the respective link is at a horizontal position. Perform one of the following:

- Use an MRP instruction to set all the joint angles to 0° at this position.
- Configure the values for the Zero Angle Offsets on the **Geometry** tab in the **Coordinate System Properties** dialog box equal to the values of the joints in a horizontal position.

Configure Zero Angle Orientations for Delta Three-dimensional robot

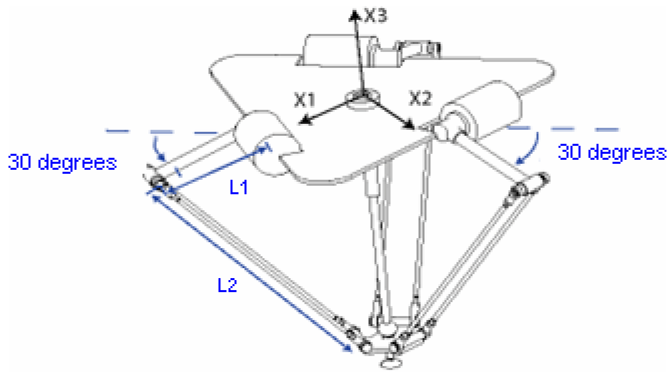
For Delta robot geometries, the internal transformation equations in the Logix Designer application are written assuming that:

- Joints are at 0° when link L1 is horizontal.
- As each top link (L1) moves downward, its corresponding joint axis (J1, J2, or J3) is rotating in the positive direction.

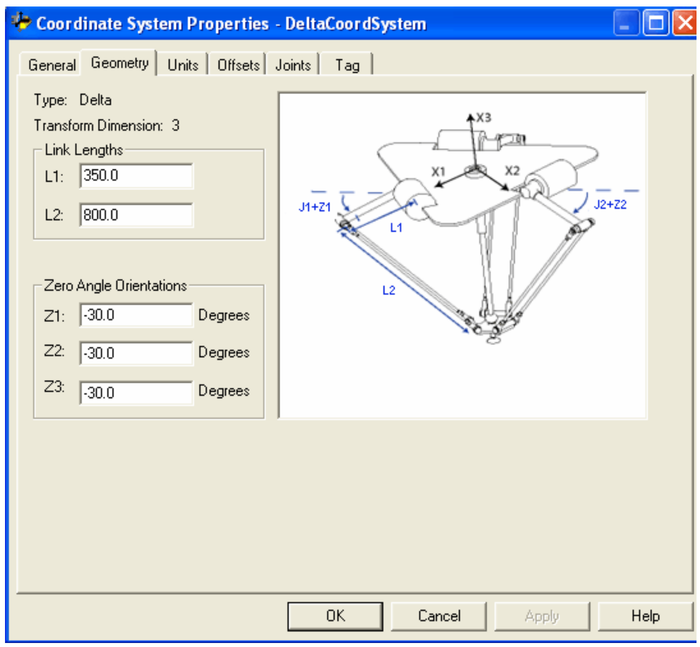
If you want the joint angular position when L1 is horizontal to be at any other value than 0°, then configure the zero angle orientation values on the **Geometry** tab on **Coordinate System Properties** dialog box to align the joint angle positions with the internal equations.

For example, if the Delta robot is mounted so that the joints attached at the top plate are homed at 30° in the positive direction below horizontal and you want the Logix Designer application readout values to be zero in this position, then configure the Zero Angle Orientation values to -30° on the **Geometry** tab on the **Coordinate System Properties** dialog box.

Delta Robot with Joints Homed at 30°



Configuring Delta robot Zero Angle orientation



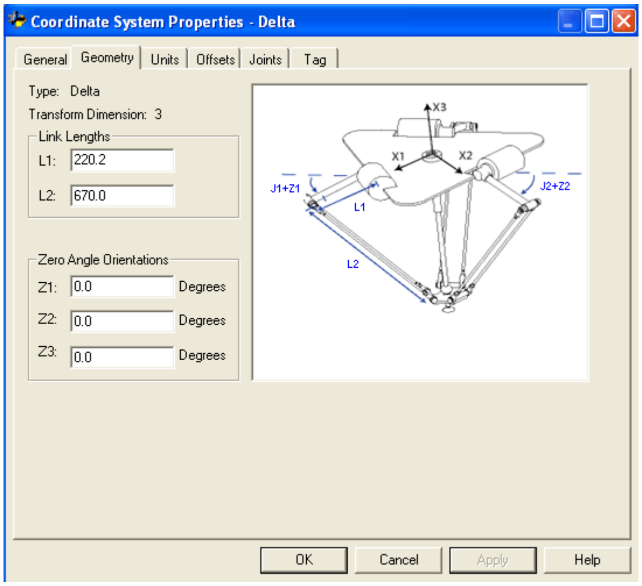
Identify the work envelope for a Delta Three-dimensional robot

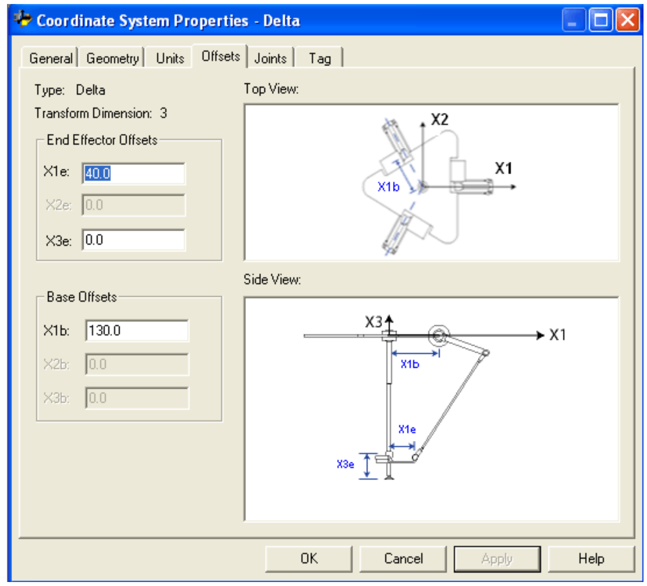
The work envelope is the three-dimensional region of space that defines the reaching boundaries for the robot arm. The typical work envelope for a Delta robot looks similar to plane in the upper region, with sides similar to a hexagonal prism and the lower portion similar to a sphere. For more information regarding the work envelope of Delta three-dimensional robots, see the documentation provided by the robot manufacturer.

Program the robot within a rectangular solid defined inside the robot's work zone. The rectangular solid is defined by the positive and negative dimensions of the X1, X2, X3 virtual source axes. Be sure that the robot position does not go outside the rectangular solid. Check the position in the event task.

To avoid issues with singularity positions, the MCT instruction internally calculates the joint limits for the Delta robot geometries. When an MCT instruction is invoked for the first time, the maximum positive and maximum negative joint limits are internally calculated based upon the link lengths and offset values entered on the **Geometry** and **Offsets** tabs in the **Coordinate System Properties** dialog box.

Delta three-dimensional Configuration Systems Properties dialog box - Geometry and Offsets tabs



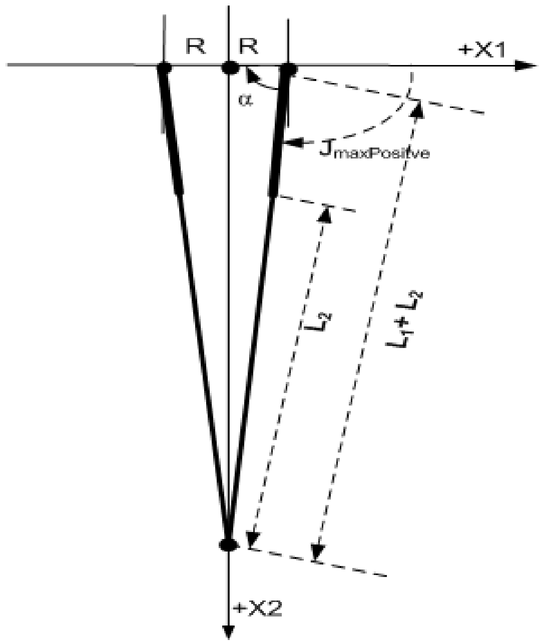


During each scan, the joint positions in the forward and inverse kinematics routines are checked to ensure that they are within the maximum and minimum negative joint limits.

Homing or moving a joint axis to a position beyond a computed joint limit and invoking a MCT instruction results in an error 67 (Invalid Transform position). For more information regarding error codes, see Logix 5000 Controllers Motion Instructions Reference Manual, publication MOTION-RM002.

Maximum positive joint limit condition

The derivations for the maximum positive joint applies to the condition when L1 and L2 are collinear.



Maximum positive joint limit position

R = absolute value of (X1b - X1e)

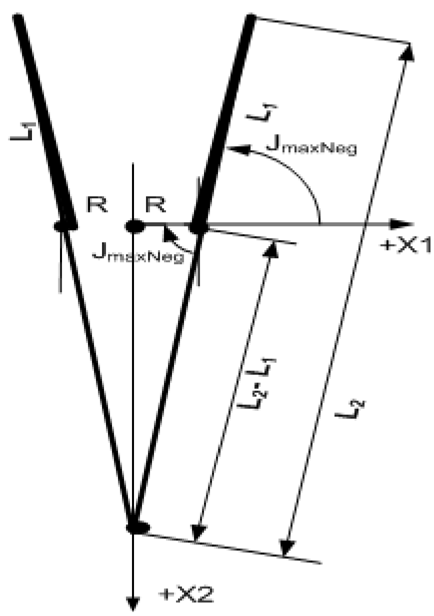
$$\alpha = \cos^{-1}\left(\frac{R}{L1 + L2}\right)$$

$$Jmax\ Positive = 180^\circ - \alpha$$

Maximum negative joint limit condition

The derivations for the maximum negative joint limit applies to the condition when L1 and L2 are folded back on top of each other.

R is computed by using the base and end-effector offsets values (X1b and X1e).



Maximum negative joint limit condition

R = absolute value of (X1b - X1e)

JMaxNeg = -cos-1

$$\left(\frac{R}{L2 - L1}\right)$$

Define configuration parameters for a Delta Three-dimensional robot

Configure the Logix Designer application to control robots with varying reach and payload capacities. The configuration parameter values for the robot include:

- Link lengths
- Base offsets

- End-effector offsets

The configuration parameter information is available from the robot manufacturer.

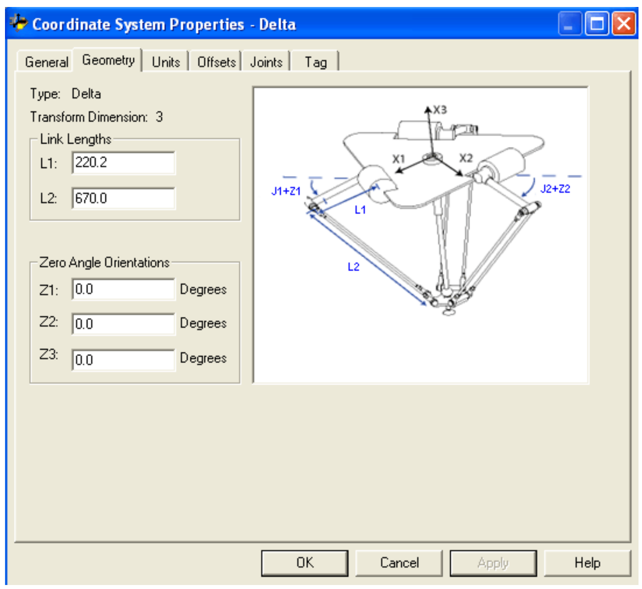
IMPORTANT Verify that the values for the Link Lengths, Base Offsets, and End-Effector Offsets are entered in the Coordinate System Properties dialog box using the same measurement units.

Link Lengths for Delta Three-dimensional robot

Link lengths are the rigid mechanical bodies attached at the rotational joints. The three-dimensional Delta robot geometry has three link pairs made up of L1 and L2. Each of the link pairs has the same dimensions.

- **L1** - is the link attached to each actuated joint (J1, J2, and J3).
- **L2** - is the parallel bar assembly attached to L1.

Enter the link lengths on the **Geometry** tab in the **Coordinate System Properties** dialog box.



See also

[Define configuration parameters for a Delta Three-dimensional robot](#) on [page 142](#)

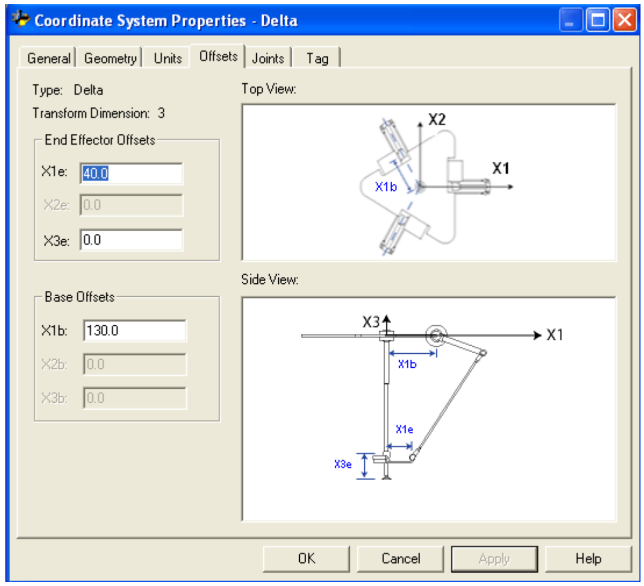
[Base Offset for Delta Three-dimension robot](#) on [page 143](#)

[End-Effector Offset for Delta Three-dimensional robot](#) on [page 144](#)

Base Offsets for Delta Three-dimensional robot

The **X1b** base offset value is available for the three-dimensional Delta robot geometry. Enter a value equal to the distance from the origin of the robot coordinate system to one of the actuator joints.

Enter the base offset value for the three-dimensional Delta robot on the **Offset** tab in the **Coordinate System Properties** dialog box.



See also

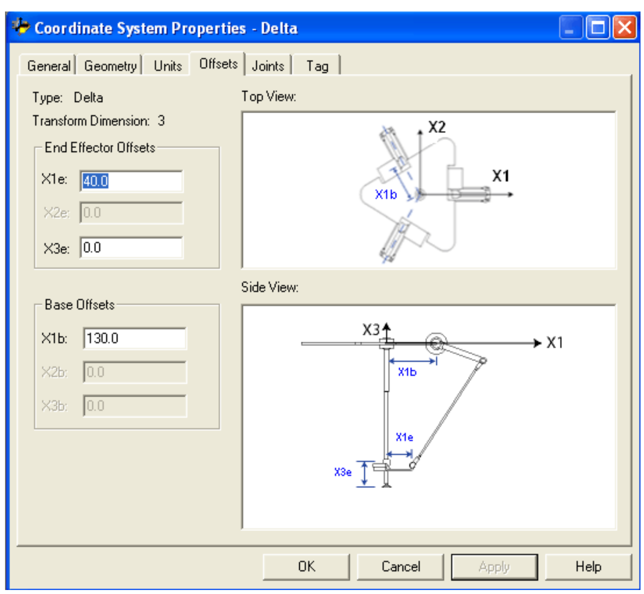
Define configuration parameters for a Delta Three-dimensional robot on page 142

End-Effector Offsets for Delta Three-dimensional robot

The two End Effector Offsets available for the three-dimensional Delta robot geometry are:

- **X1e** - This is the distance from the center of the moving plate to the lower spherical joints of the parallel arms.
- **X3e** - This is the distance from the base plate to the TCP of the gripper.

Offset values are always positive numbers. Enter the end effector offset values on the **Offsets** tab in the **Coordinate System Properties** dialog box.



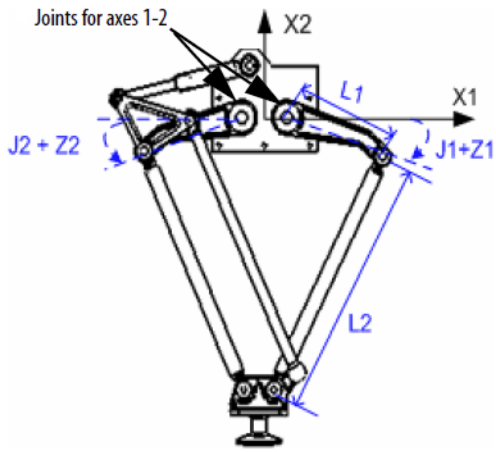
See also

Define configuration parameters for a Delta Three-dimensional robot on page 142

Base Offsets for Delta Three-dimensional robot on page 143

Configure a Delta Two-dimensional robot

This illustration shows a two-dimensional Delta robot that moves in two-dimensional Cartesian space.



This robot has two rotary joints that move the gripper in the (X1, X2) plane. Two forearm assemblies attach a fixed top plate to a movable bottom plate. A gripper is attached to the movable bottom plate. The bottom plate is always orthogonal to the X2 axis and its position is translated in Cartesian space (X1, X2) by mechanical parallelograms in each forearm assembly. The two joints, J1, and J2, are actuated joints. The joints between links L1 and L2 and between L2 and the base plate are unactuated joints.

Each joint is rotated independently to move the gripper to a programmed (X1, X2) position. As each joint axis (J1 or J2 or J1 and J2) is rotated, the TCP of the gripper moves correspondingly in the X1 or X2 direction or X1 and X2 direction. Program the TCP to a (X1, X2) coordinate, then the Logix Designer application uses internal vector dynamic calculations to compute the proper commands needed for each joint to move the gripper linearly from the current (X1, X2) position to the programmed (X1, X2) position.

The two joint axes (J1 and J2) of the robot are configured as linear axes.

To rotate the gripper, configure a third axis as a linear or rotary, independent axis.

See also

Establish the reference frame for a Delta Two-dimensional robot on page 146

Calibrate a Delta Two-dimensional robot on page 146

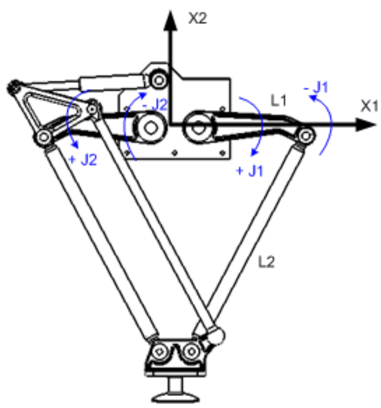
Identify the work envelope for a Delta Two-dimensional robot on page 146

Define configuration parameters for a Delta Two-dimensional robot on page 147

Establish the reference frame for a Delta Two-dimensional robot

The reference frame for the two-dimensional Delta geometry is located at the center of the fixed top plate. When the angles of joints J1 and J2 are both at 0° , each of the two L1 links is along the X1 axis. One L1 link is pointing in the positive X1 direction, the other in the negative X1 direction.

When the right-hand link L1 moves downward, joint J1 is assumed to be rotating in the positive direction and when L1 moves upward, the J1 is assumed to be moving in the negative direction. When the left-hand link L1 moves downward, joint J2 is assumed to be rotating in the positive direction and when left-hand L1 moves upward, the J2 is assumed to be moving in the negative direction.



See also

Calibrate a Delta Two-dimensional robot on page 146

Calibrate a Delta Two-dimensional robot

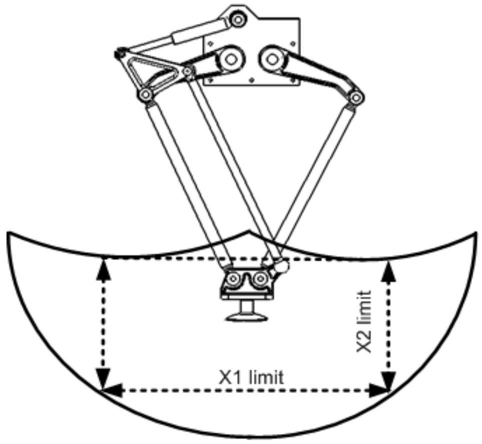
Calibrate a Delta two-dimensional robot using the same method for calibrating a Delta three-dimensional robot. Obtain the angle values from the robot manufacturer for J1 and J2 at the calibration position. Use these values to establish the reference position.

See also

Calibrate a Delta Three-dimensional robot on page 138

Identify the work envelope for a Delta Two-Dimensional robot

The work envelope is the two-dimensional region of space that defines the reaching boundaries for the robot arm. The typical working envelope for a two-dimensional Delta robot is a boundary composed of circular arcs.



Program the parameters for the two-dimensional Delta robot within a rectangle, dotted lines in the illustration, inside the robots work zone. Define the rectangle by the positive and negative dimensions of the X1, X2 virtual source axes. Be sure that the robot position does not go outside the rectangle. Check the position in the event task.

To avoid problems with singularity positions, the Logix Designer application internally calculates the joint limits for the Delta robot geometries. When an MCT instruction is invoked for the first time, the maximum positive and maximum negative joint limits are internally calculated based upon the link lengths and offset values entered on the **Geometry** and **Offsets** tabs of the **Coordinate System Properties** dialog box.

For more information about maximum positive and negative joint limits, see Maximum positive joint limit condition and Maximum negative joint limit condition.

Homing or moving a joint axis to a position beyond a computed joint limit and then invoking an MCT instruction, results in an error 67 (Invalid Transform position). For more information regarding error codes see the Logix 5000 Controllers Motion Instructions Reference Manual, publication MOTION-RM002.

Define configuration parameters for a Delta Two-dimensional robot

Configure the Logix Designer application to control robots with varying reach and payload capacities. The configuration parameter values for the robot include:

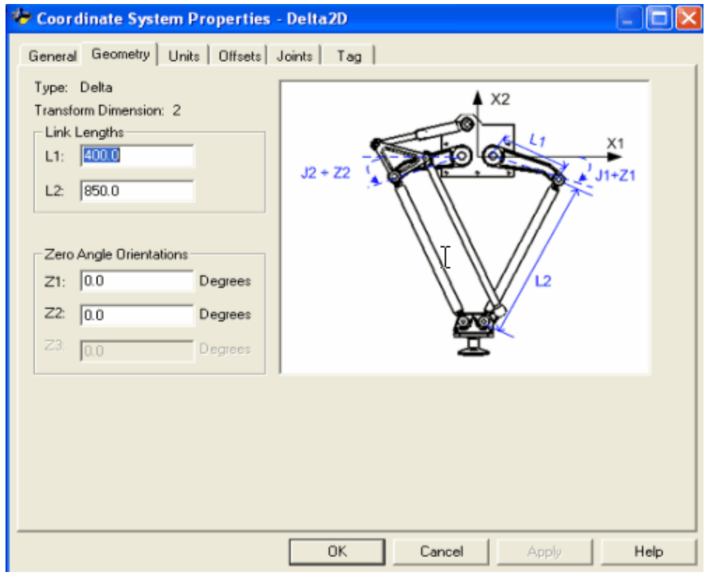
- Link lengths
- Base offsets
- End-effector offsets

The configuration parameter information is available from the robot manufacturer.

IMPORTANT	Verify that the values for the Link Lengths, Base Offsets, and End-Effector Offsets are entered in the Coordinate System Properties dialog box using the same measurement units.

Link Lengths for Delta Two-dimensional robot

Links are the rigid mechanical bodies attached at joints. The two-dimensional Delta geometry has two link pairs each with the same lengths. The link attached to each actuated joint (J1 and J2) is **L1**. The parallel bar assembly attached to link **L1** is link **L2**.



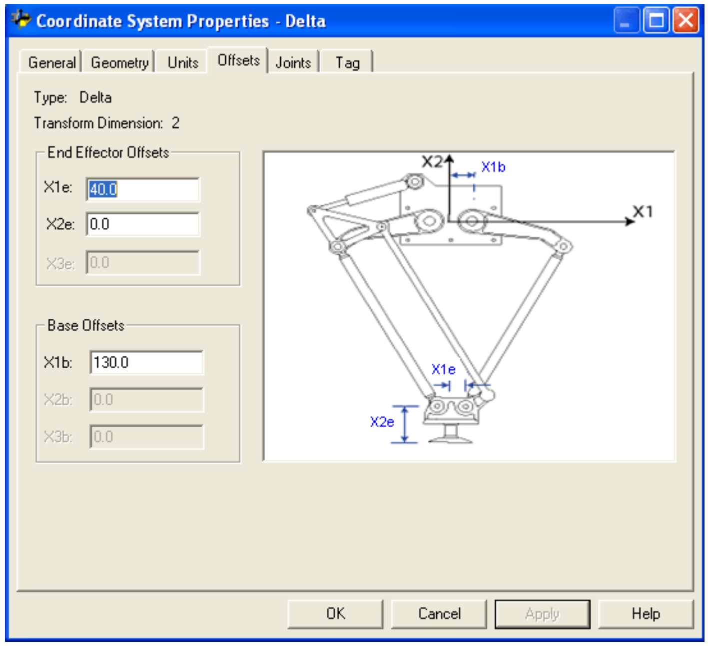
See also

Configuration parameters for a Delta Two-dimensional robot on page 147

Base Offsets for Delta Two-dimensional robot

The **X1b** base offset value is available for the two-dimensional Delta robot geometry. Enter a value equal to the distance from the origin of the robot coordinate system to one of the actuator joints.

Enter the base offset value for the two-dimensional Delta robot on the **Offset** tab in the **Coordinate System Properties** dialog box.



See also

[Define configuration parameters for a Delta Two-dimensional robot](#) on [page 147](#)

[Link lengths for Two-dimensional robot](#) on [page 143](#)

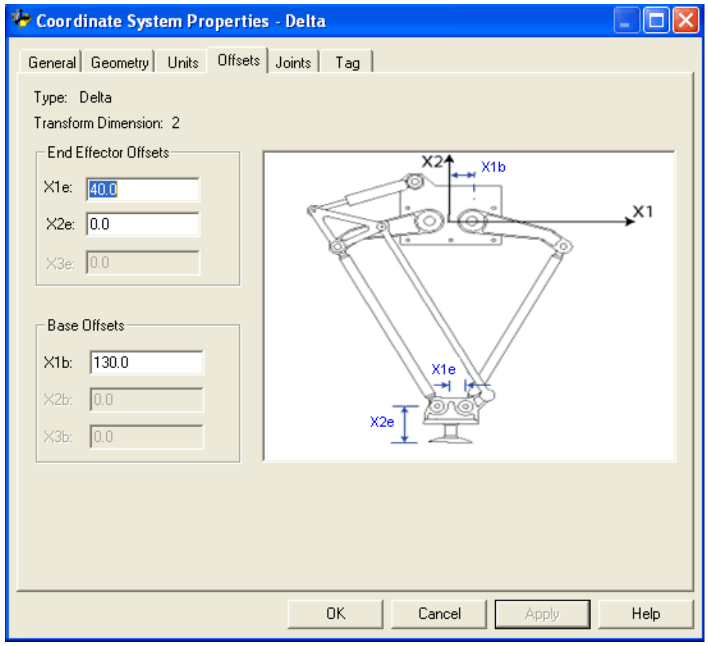
[End-Effector Offsets for Two-dimensional robot](#) on [page 144](#)

End-Effector Offsets for Delta Two-dimensional robot

There are two end effector offsets available for the two-dimensional Delta robot geometry.

- **X1e** - This is the offset distance from the center of the lower plate to the lower spherical joints of the parallel arms.
- **X2e** - This is the distance from the lower plate to the TCP of the gripper.

Enter the end effector offset values on the **Offsets** tab in the **Coordinate System Properties** dialog box.



See also

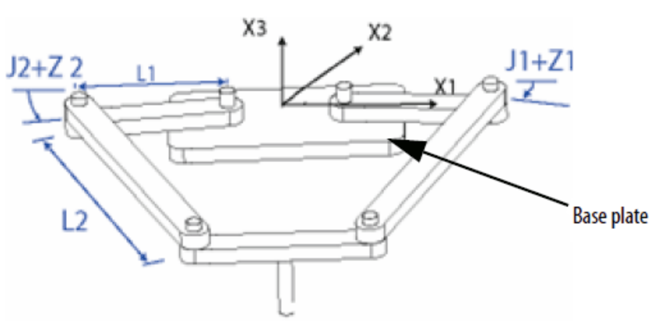
Define configuration parameters for a Delta Two-dimensional robot on page 147

Link lengths for Two-dimensional robot on page 143

Base Offsets for Two-dimensional robot on page 148

Configure a SCARA Delta robot

The SCARA Delta robot geometry is similar to a two-dimensional Delta robot geometry except that the X1-X2 plane is tilted horizontally with the third linear axis in the vertical direction (X3).



See also

Establish the reference frame for a SCARA Delta robot on page 151

Calibrate a SCARA Delta robot on page 152

Identify the work envelope for a SCARA Delta robot on page 152

Define configuration parameters for a SCARA Delta robot on page 152

Configure a Delta robot with a Negative X1b offset on page 154

Establish the reference frame for a SCARA Delta robot

The reference frame for the SCARA Delta robot is located at the center of the base plate.

When the angles of joints J1 and J2 are both at 0°, the two L1 links are along the X1 axis. One L1 link is pointing in the positive X1 direction, the other in the negative X1 direction.

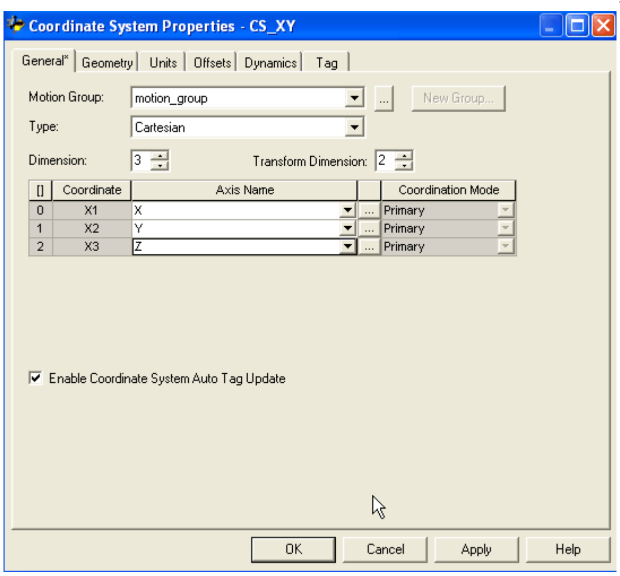
When the right-hand link L1 moves in the clockwise direction (looking down on the robot), joint J1 is assumed to be rotating in the positive direction. When the right-hand link L1 moves counterclockwise, joint J1 is assumed to be moving in the negative direction.

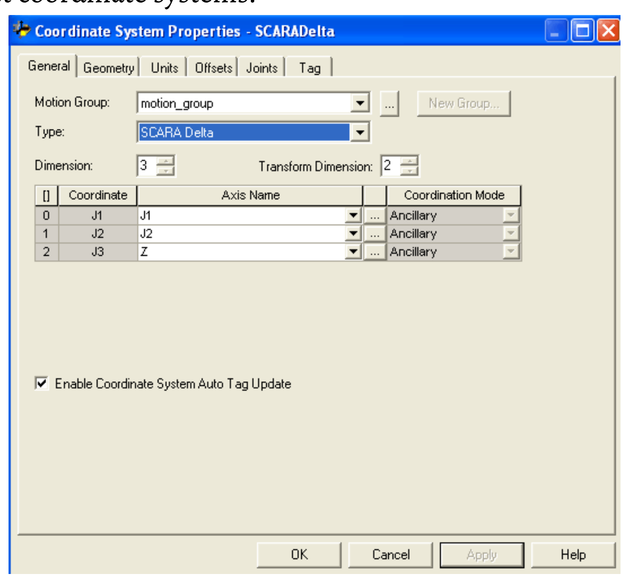
When left-hand link L1 moves in the clockwise direction, joint J2 is assumed to be moving in the negative direction. When the left-hand link L1 moves in the counterclockwise direction, joint J2 is assumed to be rotating in the positive direction.

Based on the right hand rule, X3 positive will be orthogonal to the X1-X2 plane pointing up. The linear axis will always move in the X3 direction.

When configuring a SCARA Delta robot in the Logix Designer application, observe these guidelines:

- Configure the source and the target coordinate system with a transform dimension of two.
- The linear axis configured as a third axis must be the same for both the source and target coordinate systems.





Calibrate a SCARA Delta robot

Calibrate a SCARA Delta robot using the same method for calibrating a Delta three-dimensional robot. For more information about calibration, see Calibrate a Delta Three-dimensional Robot.

See also

Calibrate a Delta Three-dimensional Robot on page 138

Identify the work envelope for a SCARA Delta robot

The work envelope for a SCARA Delta robot is similar to the two-dimensional Delta robot in the X1-X2 plane. The third linear axis extends the work region making it a solid region. The maximum positive and negative limits of the linear axis define the height of the solid region.

It is recommended to program the SCARA Delta robot within a rectangular solid defined inside the work zone of the robot. Define the rectangular solid by the positive and negative dimensions of the X1, X2, X3 virtual source axes. Be sure that the robot position does not go outside the rectangular solid. Check the position in the event task.

To avoid problems with singularity positions, the Logix Designer application internally calculates the joint limits for the Delta robot geometries. For more information about maximum positive and negative joint limits, see Maximum positive joint limit condition and Maximum negative joint limit condition.

Homing or moving a joint axis to a position beyond a computed joint limit, and invoking an MCT instruction, results in an **error 67 Invalid Transform position**. For more information regarding error codes, see Logix 5000 Controllers Motion Instructions Reference Manual, publication MOTION-RM002.

See also

Maximum positive joint limit condition on page 141

Maximum negative joint limit condition on page 142

Define configuration parameters for a SCARA Delta robot

The Logix Designer application can be configured for control of robots with varying reach and payload capacities. The configuration parameter values for the robot include:

- Link lengths
- Base offset
- End-effector offset

The configuration parameter information is available from the robot manufacturer.

IMPORTANT	Verify that the values for the Link Lengths, Base Offsets, and End-Effector Offsets are entered in the Coordinate System Properties dialog box using the same measurement units.

See also

Link length for SCARA Delta robot on page 153

Base Offset for SCARA Delta robot on page 153

End Effector Offset for SCARA Delta robot on page 154

Link lengths for SCARA Delta Robot

Links are the rigid mechanical bodies attached at joints. The SCARA Delta robot has two link pairs each with the same lengths. The link attached to each actuated joint (J1 and J2) is **L1**. The parallel bar assembly attached to link **L1** is link **L2**.

See also

Define configuration parameters for a SCARA Delta robot on page 152

Base Offset for SCARA Delta Robot

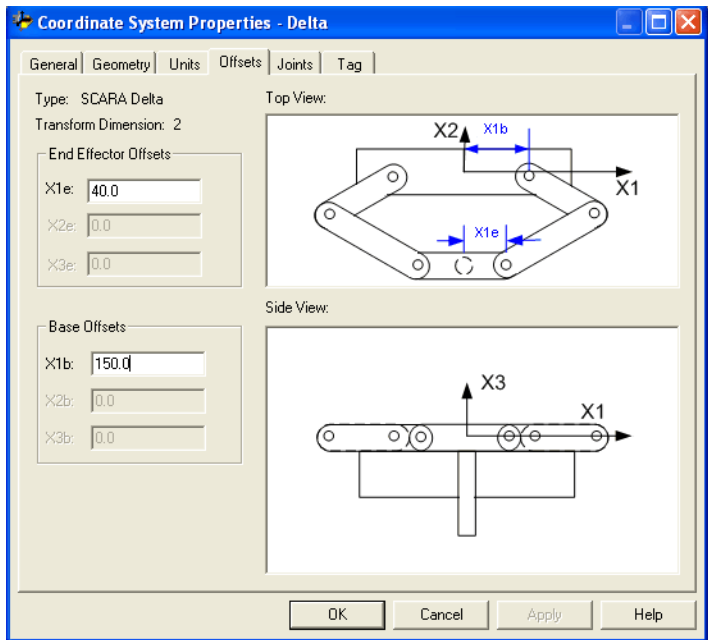
The **X1b** Base Offset is available for the SCARA Delta robot geometry. Type the value equal to the distance from the origin of the robot coordinate system to an actuator joint. The Base Offset value is always a positive number.

See also

Define configuration parameters for a SCARA Delta robot on page 152

End Effector Offset for SCARA Delta Robot

The **X1e End-Effector Offsets** is available for the SCARA Delta robot geometry on the **Offsets** tab in the **Coordinate System Properties** dialog box. Type the value for the distance from the center of the moving plate to one of the spherical joints of the parallel arms. The **End-Effector Offsets** value is always a positive number.

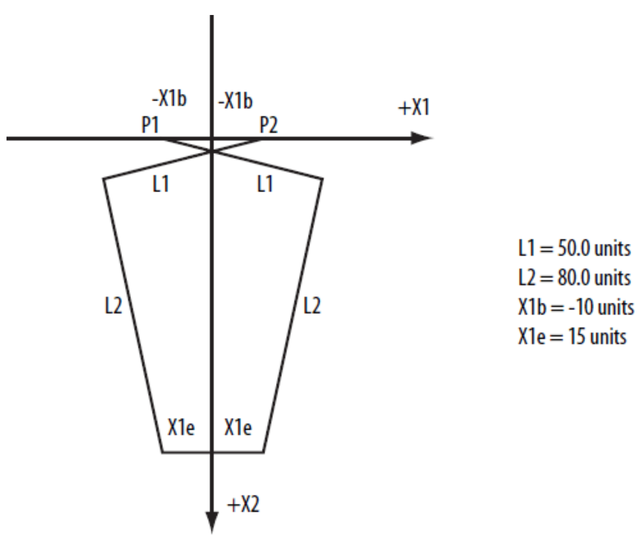


See also

Define configuration parameters for a SCARA Delta robot on page 152

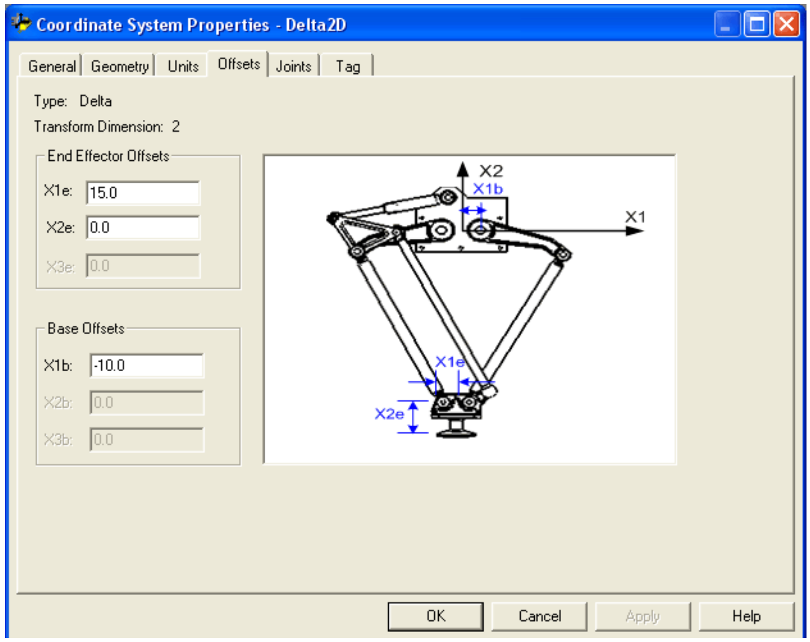
Configure a Delta robot with a Negative X1b offset

Beginning with version 17 of the application, you can use negative offsets for the X1b base offset on 2D and 3D delta geometries. For example, a mechanical 2D delta robot using a negative X1b offset has a mechanical configuration as shown in the diagram.



The base offset X1b is the value equal to the distance from the origin of the robot coordinate system to one of the actuator joints. In the previous figure, one of the actuator joints (P1), is on the negative side of X1. The base offset X1b is -10 units from the origin of the coordinate system (X1 - X2 intersection) to P1.

The Logix Designer application coordinate system configuration for the offset tab used with the preceding example is shown in the following example.



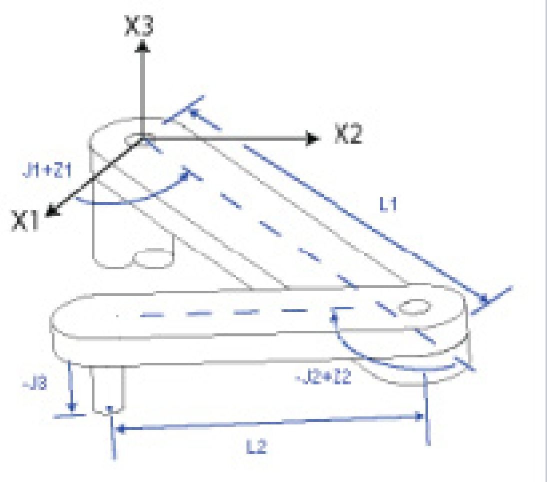
This negative offset description also applies for Delta 3D and SCARA-Delta configurations.

Configure a SCARA Independent Robot

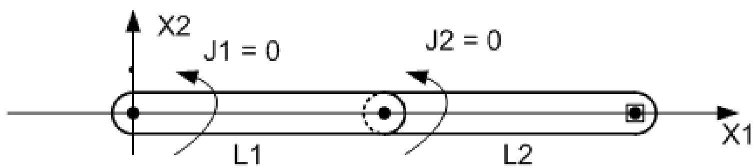
The typical SCARA Independent robot has two revolute joints and a single prismatic joint. This robot is identical to the Articulated Independent two dimensional robot except that the X1-X2 plane is tilted horizontally with a third linear axis in the vertical direction. Use these guidelines when configuring a SCARA Independent robot.

Establish the reference frame for a SCARA Independent robot

The reference frame for the SCARA Independent geometry is at the base of link L1.



The internal kinematic equations are written as if the start position for the SCARA Independent robot joints are as shown in this diagram.



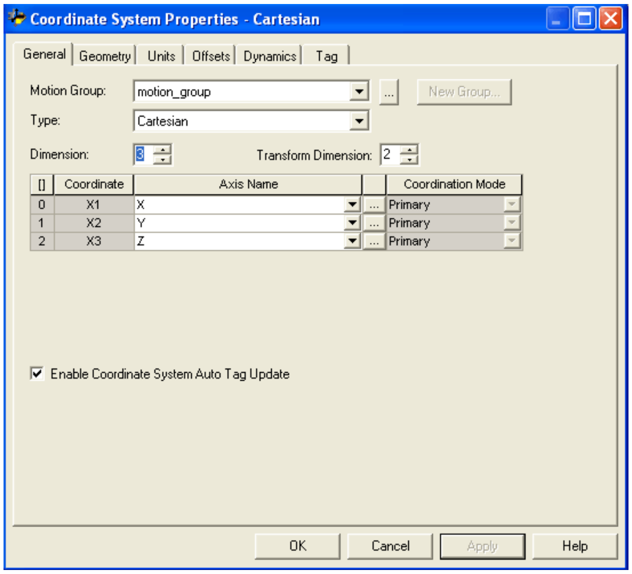
- +J1 is measured counterclockwise around +X3 axis starting at an angle of J1 =0.0 when L1 is along the X1 axis.
- +J2 is measured counterclockwise starting with J2 = 0 when Link L2 is aligned with link L1.
- +J3 is a prismatic axis that moves parallel to +X3 axis.

For information about alternate methods for establishing a reference frame, see Articulated Independent robot.

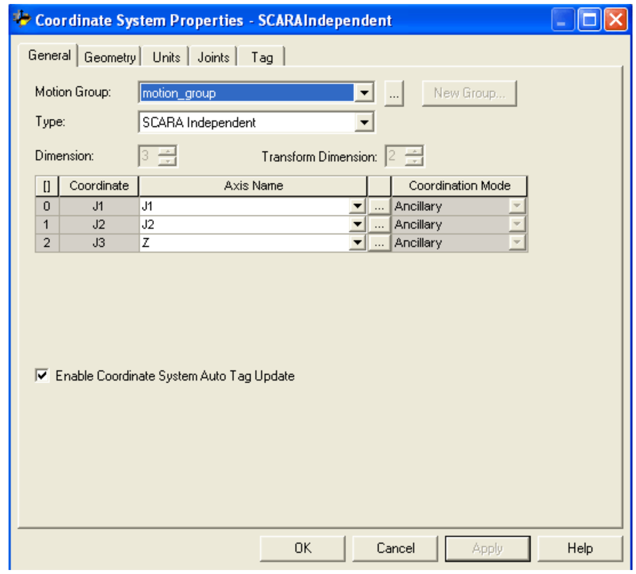
When configuring the parameters for the source coordinate system and the target coordinate system for a SCARA Independent robot, observe these guidelines:

- The transform dimension value should be set to two for both the source and target coordinate systems because only J1 and J2 are involved in the transformations.
- The Z axis is configured as a member of both the source and target coordinate systems.

For additional information about establishing a reference frame, see Articulated Independent robot.



Source coordinate system configuration



Target coordinate system configuration

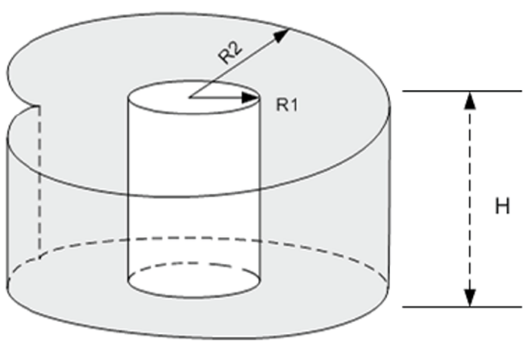
See also

Articulated Independent robot on page 65

Identify the work envelope for a SCARA Independent robot

The work envelope is the three-dimensional region of space that defines the reaching boundaries for the robot arm. The work envelope for the SCARA Independent robot is a hollow cylinder with:

- A height equal to the travel limit of the J3 axis.
- An inner radius (R1) equal to |L1-L2|.
- An outer radius (R2) equal to |L1+L2|.



Define configuration parameters for a SCARA Independent robot

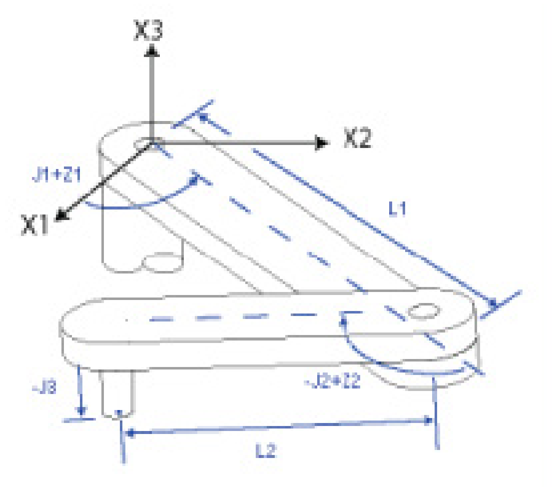
Configure the Logix Designer application to control robots with varying reach and payload capacities. The configuration parameter values for the robot include:

- Link lengths

The configuration parameter information is available from the robot manufacturer.

Tip: Base offsets and end-effector offsets do not apply to a SCARA Independent robot.

This example illustrates the typical configuration parameters for a SCARA Independent robot.



See also

Link Lengths for SCARA Independent robot on page 158

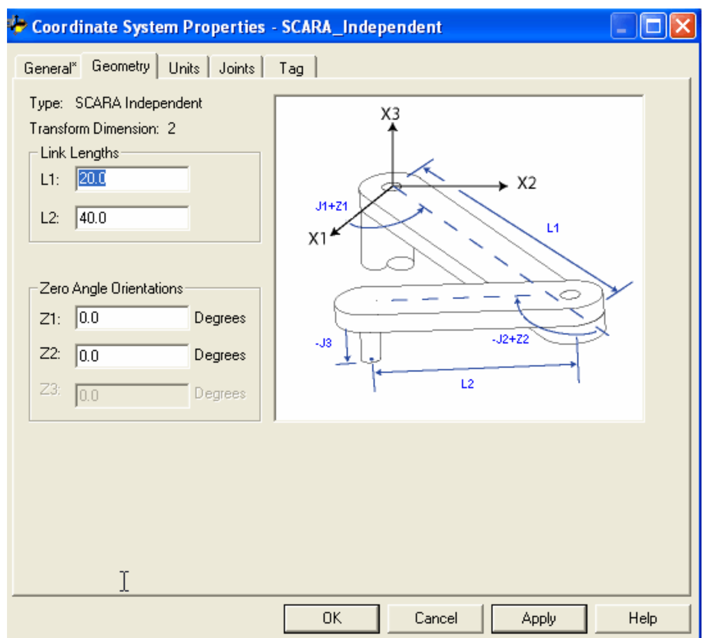
Link lengths for SCARA Independent robot

Link lengths are the rigid mechanical bodies attached at joints.

Type the **Link Lengths** values.

For the robot shown in SCARA Independent, the **Link Length** values are:

- L1 = 20
- L2 = 40



Base offsets and end-effector offsets do not apply to a SCARA Independent robot configuration.

Configure a Cartesian Gantry robot

Use these guidelines when configuring a Cartesian Gantry robot.

See also

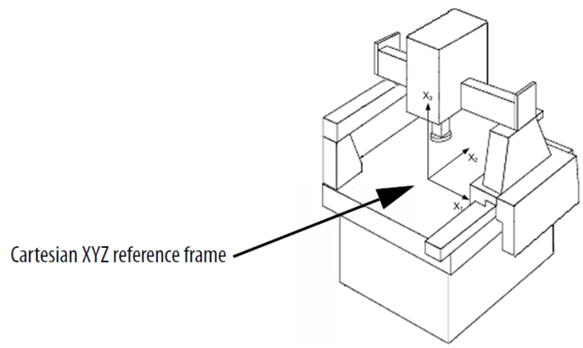
Establish the reference frame for a Cartesian Gantry robot on page 159

Identify the work envelope for a Cartesian Gantry robot on page 159

Define configuration parameters for a Cartesian Gantry robot on page 159

Establish the reference frame for a Cartesian Gantry robot

For a Cartesian Gantry robot, the reference frame is an orthogonal set of X1, X2, and X3 axes positioned anywhere on the Cartesian robot. All global coordinate measurements (points) are relative to this reference frame. Typically, the reference frame is aligned with the X1, X2, and X3 axes of the machine.



To establish a Local coordinate system with axes positions different from the reference frame, use the Motion Redefine Position (MRP) instruction to reset the position register. Also use the Offset Vector in the MCT transform instruction to establish an offset between the Local coordinate system and the reference frame.

For more information about Motion Instructions, see Logix 5000 Controllers Motion Instructions Reference Manual, publication MOTION-RM002.

Identify the work envelope for a Cartesian Gantry robot

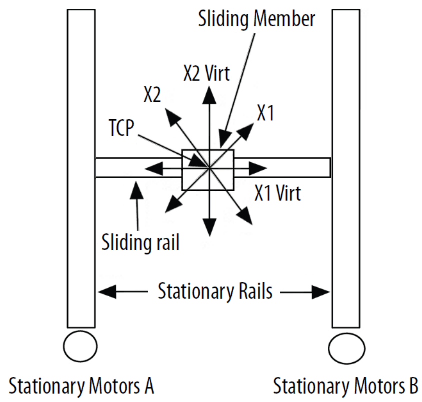
The work envelope for a Cartesian Gantry robot is typically a solid rectangle of length, width, and height that is equal to the axis travel limits.

Define configuration parameters for a Cartesian Gantry robot

Defining the link lengths, base offset, or end-effector offset configuration parameters is not required for a Cartesian Gantry robot.

Configure a Cartesian H-bot robot

The H-bot is a special type of Cartesian two-axis gantry robot. This type of machine has three rails positioned in the form of a letter H. Two motors are positioned at the end of each leg of the robot. Unlike a standard gantry robot, neither motor is riding on top of the moving rails. Use these guidelines when configuring a Cartesian H-bot.



In the Cartesian H-bot illustration, the X1 and X2 axes are the real axes on the robot. X1 Virt and X2 Virt are configured as the virtual axes.

The configuration of the H-bot mechanical linkages enable it to move at a 45° angle to the axes when motor A or motor B is rotated.

For example, when:

- Motor A (X1 axis) is rotated, the robot moves along a straight line at + 45° angle.
- Motor B (X2 axis) is rotated, the machine moves at an angle of -45°.
- Motors A and B are rotated clockwise at the same speed, then the machine moves along a horizontal line.
- Motors A and B are rotated counterclockwise at the same speed then, the machine moves along a vertical line.

Any X,Y position can be reached by properly programming the two motors.

For example, a move of (X1 = 10, X2 = 0) causes the X1X2 axes to move to a position of (X1=7.0711, X2=7.0711). A move to (X1=10, X2 =10) causes the robot to move to a position of (X1=0, X2=14.142).

Utilizing the Logix Designer application Kinematics function configured with two Cartesian coordinate systems and a -45° rotation performs the function.

To configure two Cartesian coordinate systems:

Coordinate System 1 (CS1) and Coordinate System 2 (CS2) each contain two linear axes.

1. Configure CS1 to contain the virtual X1 and X2 axes.
2. Configure CS2 to contain the real X1 and X2 axes.

3. Configure the Orientation vector of the MCT instruction as (0,0, -45), a negative degree rotation around the X3 axis.
4. Configure the Translation vector as (0, 0, 0).
5. Link the CS1 and CS2 by using a MCT instruction.
6. Home the H-bot and then program all moves in CS1.

The machine moves the tool center point (TCP) to the programmed coordinates in CS2. The -45° rotation introduced by the Kinematics, counteracts the 45° rotation introduced by the mechanics of the machine and the H-bot moves to the CS1 configured coordinates. As a result, a programmed move of X1virt=10, X2virt=5 moves to a real mechanical position of X1=10, X2=5.

See also

Establish the reference frame for a Cartesian H-bot robot on page 161

Identify the work envelope for a Cartesian H-bot robot on page 161

Define configuration parameters for a Cartesian H-bot robot on page 161

Establish the reference frame for a Cartesian H-bot

For a Cartesian H-bot, the Base coordinate system is an orthogonal set of X1, X2 axes postponed anywhere on the Cartesian H-bot. The angular rotation of the reference frame may not be rotated for this robot since the angular rotation vector is used to achieve the 45° rotation required for the mechanical operation.

Identify the work envelope for a Cartesian H-bot

The work envelope for a Cartesian H-bot is a rectangle of length and width equal to the axis soft travel limits.

Define configuration parameters for a Cartesian H-bot robot

Defining the link lengths, base offset, or end-effector offset configuration parameters is not required for a Cartesian H-bot robot.

Geometries with orientation support

Use these guidelines and information to configure the robot geometries with orientation support in Logix Designer application. These robot geometries include:

- Delta J1J2J6 robot
- Delta J1J2J3J6 robot
- Delta J1J2J3J4J5 robot

Also included is information about:

- Cartesian Coordinate System frame
- Defining frames for programming different robot applications
- Configuring and programming turns counters
- Using MCPM to program Ry axis position to exhibit mirror image orientation behavior

The **Coordinate Definition** parameter in the **Coordinate System Properties** dialog box determines whether or not there is orientation support in the coordinate system.

See also

Configure a Cartesian Coordinate System on page 39

Cartesian coordinate frame

This information provides information about the Cartesian coordinate frame. A Cartesian coordinate frame is a set of orthogonal lines that intersect at an origin, such as two lines in a plane or three in space. A Cartesian coordinate frame in a plane has two perpendicular lines (the x-axis and y-axis); in three-dimensional space, it has three (the x-axis, y-axis, and z-axis).

See also

Cartesian Point specification on page 164

Transform representation of point on page 167

Orientation specification on page 171

Point conversion on page 173

RxRyRz, flip, mirror flip condition on page 174

Translation and rotation example on page 179

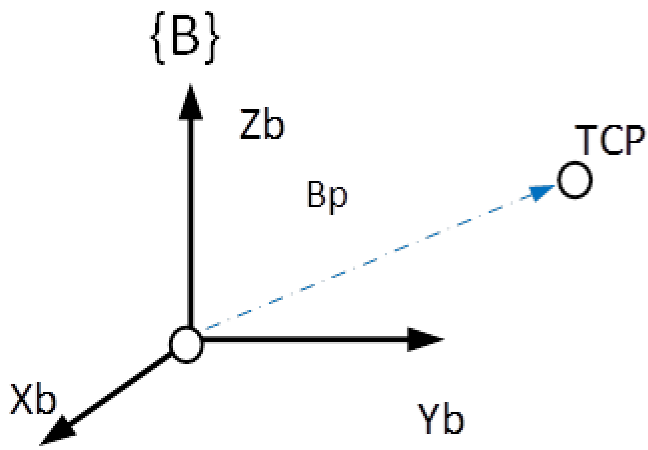
Cartesian Point specification

The Cartesian Point is composed of the following two components:

- Translation - describes the vector connecting two Cartesian points
- Orientation - the three ordered rotations around the X, Y, and Z Cartesian axes

Translation Specification

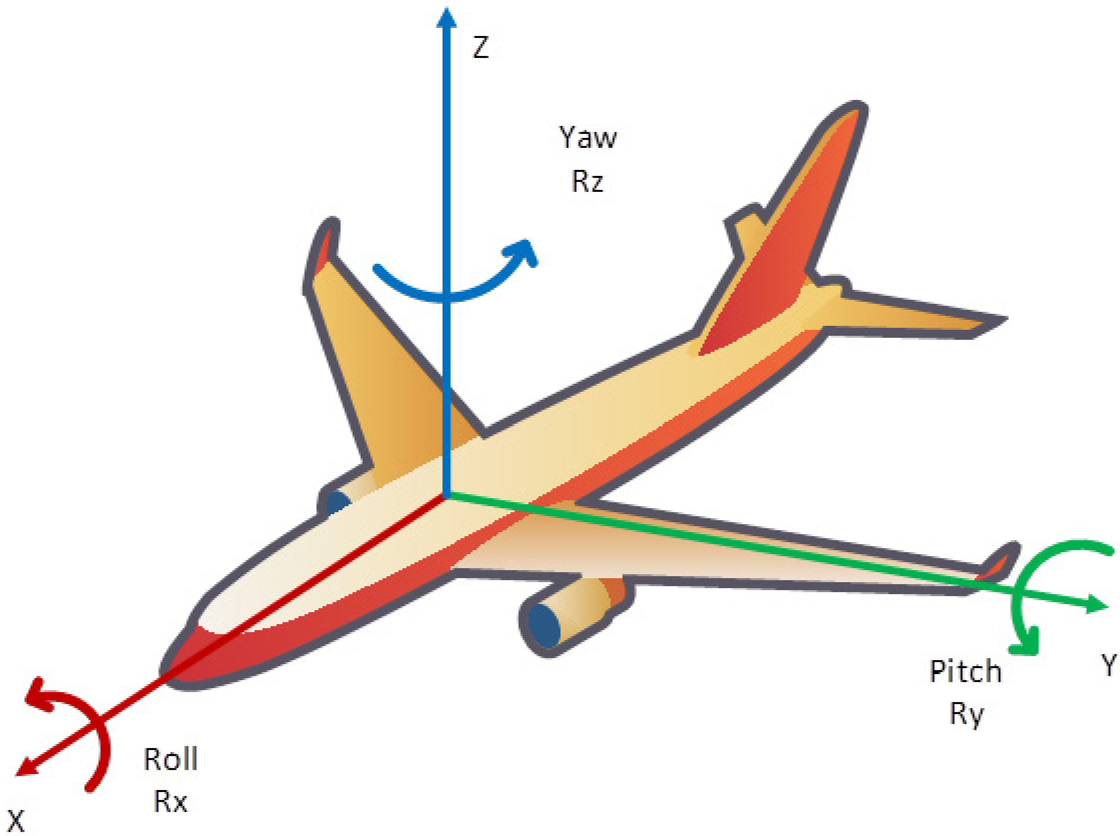
Typically, a point in space is specified by the three coordinates of the point with respect to the base coordinate system as shown in the following figure. The three coordinates of the point are X, Y, Z. This specification is also called 3 by 1 position vector with respect to the base coordinate system.



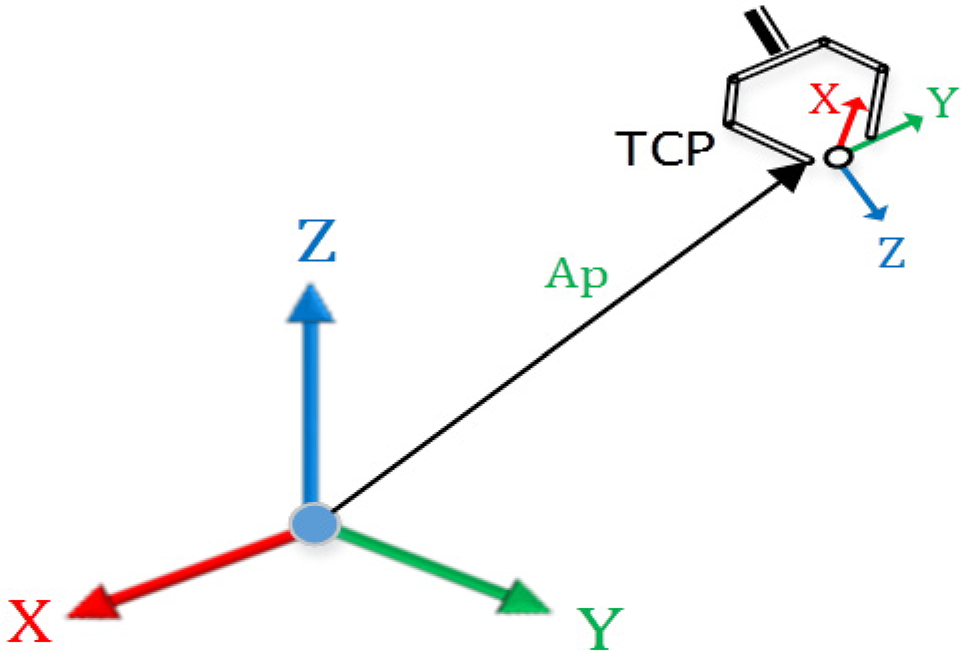
$$\mathbf{Bp} = \begin{vmatrix} X \\ Y \\ Z \end{vmatrix}$$

Orientation Specification

It is often necessary to represent a point in space, and describe the orientation of a body in space. See the orientation of the aircraft in the following diagram. Orientation specifies the roll, pitch and yaw (orientation) of a flying aircraft. Roll, pitch and yaw are standard navigation terms for airplanes and ships, and represent the rotations around X, Y, and Z axes of the base coordinate system.



Another example is the point directly between the fingertips of a manipulator shown in the following diagram. The orientation or pose specifies how the manipulator is oriented. For example, one of the orientation parameters is how the manipulator is approaching the object between the fingers.



The position and orientation explained above describe the point in space with respect to the base frame as shown in the preceding diagram.

See also

[Transform Representation of Point on page 167]

Orientation Specification on page 171

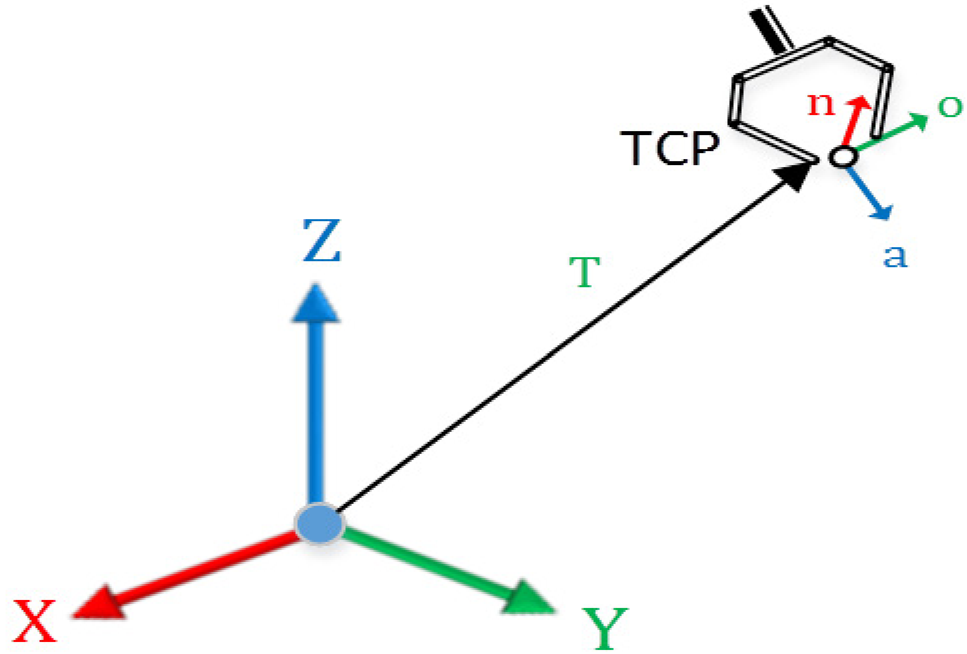
Point Conversion on page 173

RxRyRz, flip, mirror flip condition on page 174

Translation and Rotation example on page 179

Transform representation of point

The mathematical forms described above to specify the points can also be used to translate points and rotate vectors or do both. The figure above can be modified to show the position vector and orientation frame as shown below.



Translation Specification of Point

The translation specifies the position vector of the point as discussed above with three components X,Y,Z.

$$\mathbf{T} = \begin{vmatrix} Tx \\ Ty \\ Tz \end{vmatrix}$$

Rotation Specification of Point - *n,o,a*

The orientation specifies the orientation of the point specified by three vectors as shown in the figure above. The approach vector *a* specifies how the object is approached by the robot's end effector as shown in the figure above. The orientation vector *o* specifies orientation of the end effector fingertip to fingertip when approaching the object as shown in the figure above. The final vector, known as the normal vector *n* is a vector normal to the plane formed by approach and orientation vectors. The *n* vector is X in the robot wrist coordinate system, the *o* vector is Y, and the *a* vector is Z.

The three 3 by 1 vectors *n o a* form a 3 by 3 Rotation matrix which defines the rotated frame with respect to the base frame of the robot. The vectors *n o a* are unit vectors with respect to the base coordinate system. The columns of the rotation matrix *n o a* represent the direction cosines of the rotated orientation frame with respect to the base coordinate system.

$$\mathbf{R} = \begin{vmatrix} N_x & O_x & A_x \\ N_y & O_y & A_y \\ N_z & O_z & A_z \end{vmatrix}$$

Translation Specification of Point - *n,o,a,t*

The translation and rotation specifications are combined to form a 4 by 4 transform matrix with elements from translation and orientation specification as shown below which completely specify the position and orientation of a point.

$${}^{A}_{B}\mathrm{P} = \begin{bmatrix} [\mathrm{R}_{3x3}] & & [\mathrm{p}_{3x1}] \\ 0 \quad 0 & 0 & 1 \end{bmatrix} \text{ or}$$

$${}^{A}_{B}\mathrm{P} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{or}$$

$${}^{A}_{B}\mathrm{P} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Transform

It turns out that the transform specification for point can also represent transform that can be used to transform any point in the reference coordinate system to the target coordinate system. And so the transform T to transform points from reference frame {A} to target frame {B} is given by the following matrix equation.

$$ {}^{A}_{B}T = \begin{bmatrix} [R_{3x3}] & [p_{3x1}] \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix} \text{ or} $$

$$ {}^{A}_{B}T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{or} $$

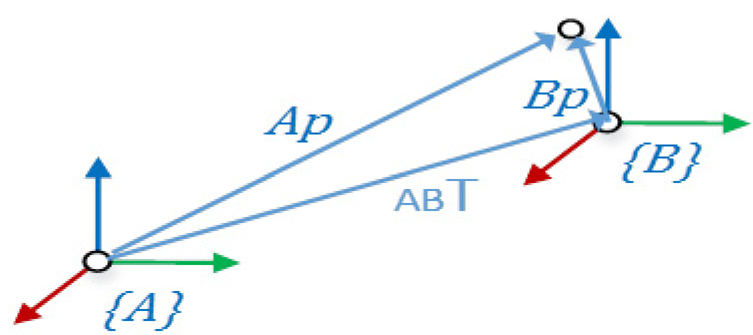
$$ {}^{A}_{B}T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

The transformation can be used to convert a point with respect to reference frame {A} to reference frame {B} using the following matrix equation.

$$ {}_{A}P = {}^{A}_{B}T \, {}_{B}P $$

Translation Transform

The translation transform is simpler and shown by the following figure as two dimensional coordinate transform example in the XZ plane. With 3D space the example would be a little more complex but can be worked using matrix multiplication mathematics.

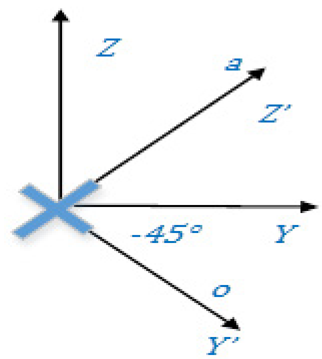


$$
{}_{A}P = {}^{A}_{B}T \times {}_{B}P = {}_{A}P
$$

$$
\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 5-1 \\ 0 & 0 & 1 & 3+2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Rotation Transform

Matrix R known as Rotation matrix transforms a base coordinate frame to the rotated coordinate frame as shown by the rotation around Y axis in the figure below.



The three rotation matrices which rotate base frame about the three base coordinate systems are important and rotate the base frame by angle Rx around X, angle Ry around Y or angle Rz around Z of the base axis as shown below. Notice that the columns represent the unit vectors of the rotated frame with respect to the base frame. The transforms align XYZ base frame to *n o a* with one to 3 successive rotations. The transforms below only represent one rotation.

$$
Rot_x(Rx) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(Rx) & -\sin(Rx) & 0 \\ 0 & \sin(Rx) & \cos(Rx) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
Rot_y(Ry) = \begin{bmatrix} \cos(Ry) & 0 & \sin(Ry) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(Ry) & 0 & \cos(Ry) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
Rot_z(Rz) = \begin{bmatrix} \cos(Rz) & -\sin(Rz) & 0 & 0 \\ \sin(Rz) & \cos(Rz) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Using this rotation matrix one can rotate ϴ to any value in the range of +/-180 ° to obtain the rotation matrix around desired base axis.

Translation + Rotation Transform

The translation plus rotation transform is more complex. With 3D space the example would be more complex but can be worked using matrix multiplication and trigonometric mathematics.

Orientation specification

The 4 by 4 matrix form of point specification is sometimes difficult to handle for user defined points but as shown in the calculations above easy to map points from one coordinate frame to another coordinate frame. E.g. End of Arm Frame to TCP frame.

When the points need to be taught it becomes difficult to teach approach and orientation vector to specify the orientation. A representation that requires only three numbers to completely specify the orientation is more desirable. It also facilitates jogging the robot around a robot base coordinate axis. E.g. Z axis.

There are several representations that require three numbers to specify the rotations. As these are rotations around an axes they are specified in degrees. The two common rotations are XYZ Fixed Angle convention and ZY'X" Euler angle conventions described below.

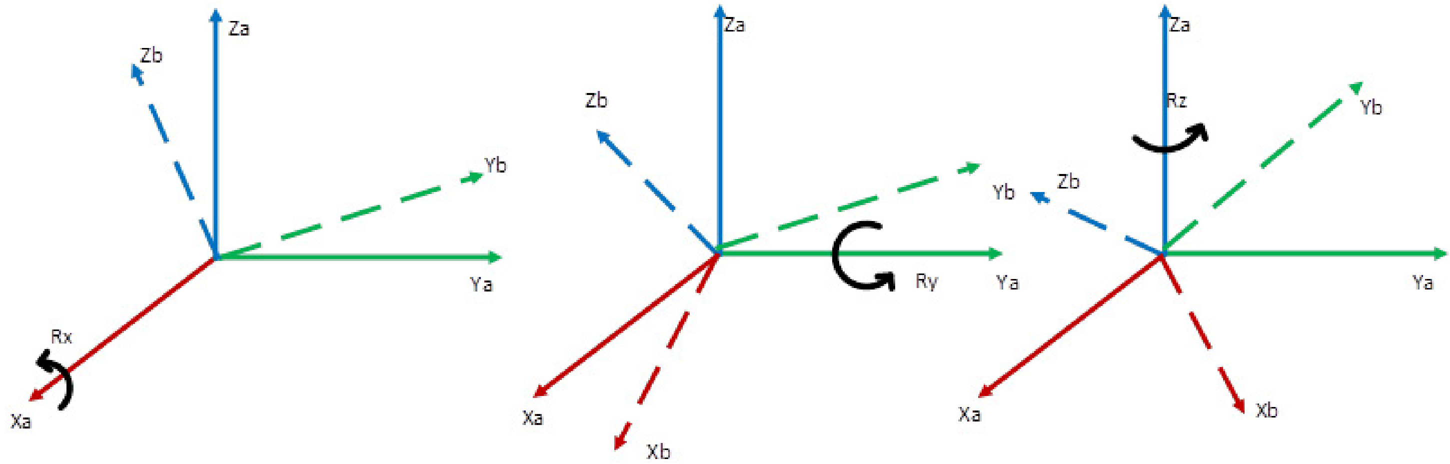
Fixed Angle - X-Y-Z

One method of describing the orientation of a frame {B} is as follows:

- Start with the frame coincident with a known reference frame {A}.
- Rotate {B} first about X_A by an angle Rx,
- then about Y_A by an angle Ry,
- and, finally, about Z_A by an angle Rz.

Each of the three rotations takes place about an axis in the fixed reference frame {A}. We call this convention for specifying the orientation X-Y-Z fixed angle. The word fixed refers to the fact that the rotations are specified about the fixed reference frame {A} as shown below.

Important:	The Logix firmware uses this convention for specifying the points. Any point in Cartesian space is specified by 6 numbers XYZRxRyRz where Rx, Ry and Rz are specified with fixed angle convention.



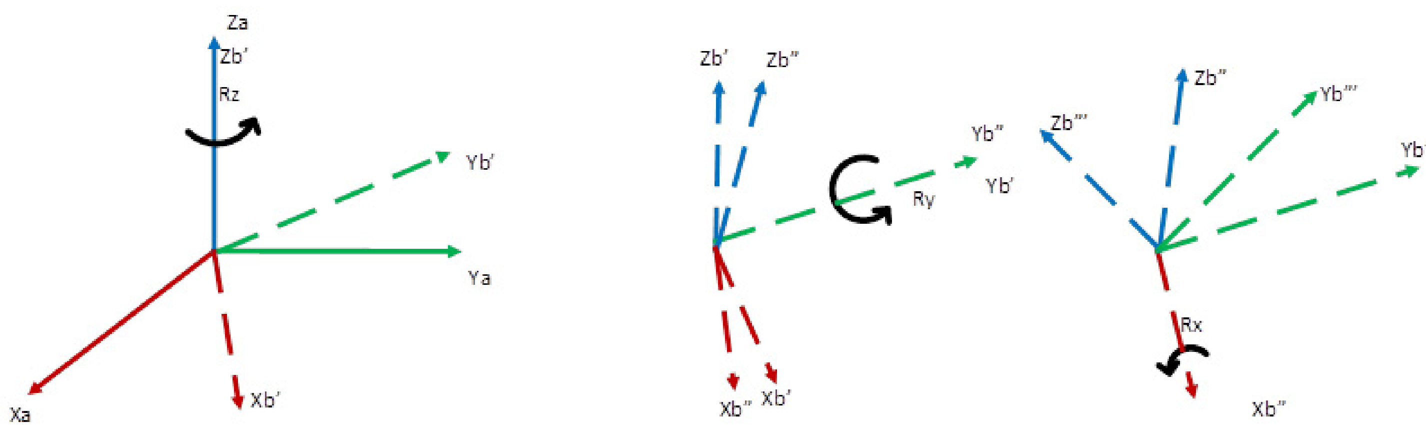
Start with a frame coincident with reference frame {A}. First rotate {B} about Xa by an angle γ, then rotate about Ya by an angle β and then rotate about Za by an angle α. It is also important to note that order of rotation is important which in this case is X-Y-Z. If this order is changed then orientation will get altered. This fact is shown in the equation below.

$${}^{A}_{B}R(\gamma,\beta,\alpha) = R_Z(\alpha)R_Y(\beta)\,R_X(\gamma)$$

Euler Angle - Z - Y' - X"

Another possible convention of a frame {B} is as follows

- Start with the frame coincident with a known reference frame {A}.
- Rotate {B} first about Z_B by an angle Rz,
- then about Y_B' by an angle Ry,
- and, finally, about X_B" by an angle Rx.



In this convention, each rotation is performed about an axis moving frame {B} rather than one of the fixed reference frame {A}. Such sets of three rotations are called Euler angles. Because the three rotations occur about the Z_B, Y_B' and X_B", we will call this representation Z-Y-X Euler angles. ZYX Euler angles is also referred in the literature as ZYX moving frame OR ZY'X".

Tip: XYZ in fixed frame convention is equivalent to ZY'X" moving frame convention.

The two conventions described above are commonly used conventions. There are other conventions like Z-Y-Z that user may be more familiar. In all there are 12 fixed angle and 12 moving frame conventions. It is possible to develop application code to convert from any of these conventions to fixed angle convention used by Logix embedded software using application code.

See also

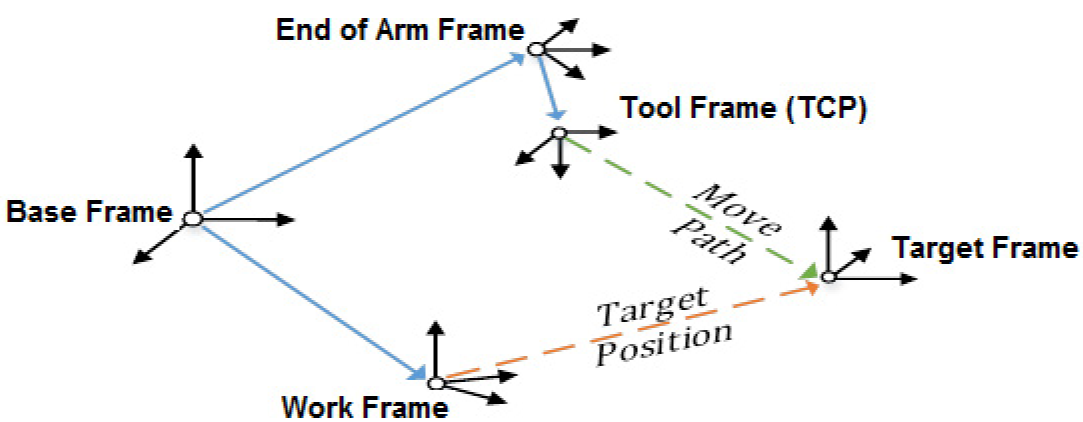
Configure a Cartesian Coordinate System on page 39

Point conversion

Conversion from XYZRxRyRz to Transform Point

A robot application sometimes needs to represent different frames for programming and moving a robot manipulator with various frames as shown in the figure below.

As a result, it is necessary to convert target point specified in XYZRxRyRz user format to its equivalent transform point represented by the 4 x 4 transform matrix. The transform point along with other transforms that map for instance tool tip with respect to the end of arm is used to set up motion of Robot manipulator through its work envelope in Cartesian or joint space to achieve the specified motion.



Conversion from Transform Point to XYZRxRyRz

It is also then necessary to transform the points in the 4 x 4 transform matrix format to the user XYXRxRyRz format for user reference, teaching and display purpose.

Transforming between the frames is complex and sometimes has limitations on computational solutions available. For the XYZ fixed format that get used by the Logix firmware, there are points with Ry rotation of 90° that has multiple solutions. This condition is described as gimbal lock condition which occurs at Ry equal to +/- 90°. The system has to handle this condition by picking a solution out of the multiple possible solutions.

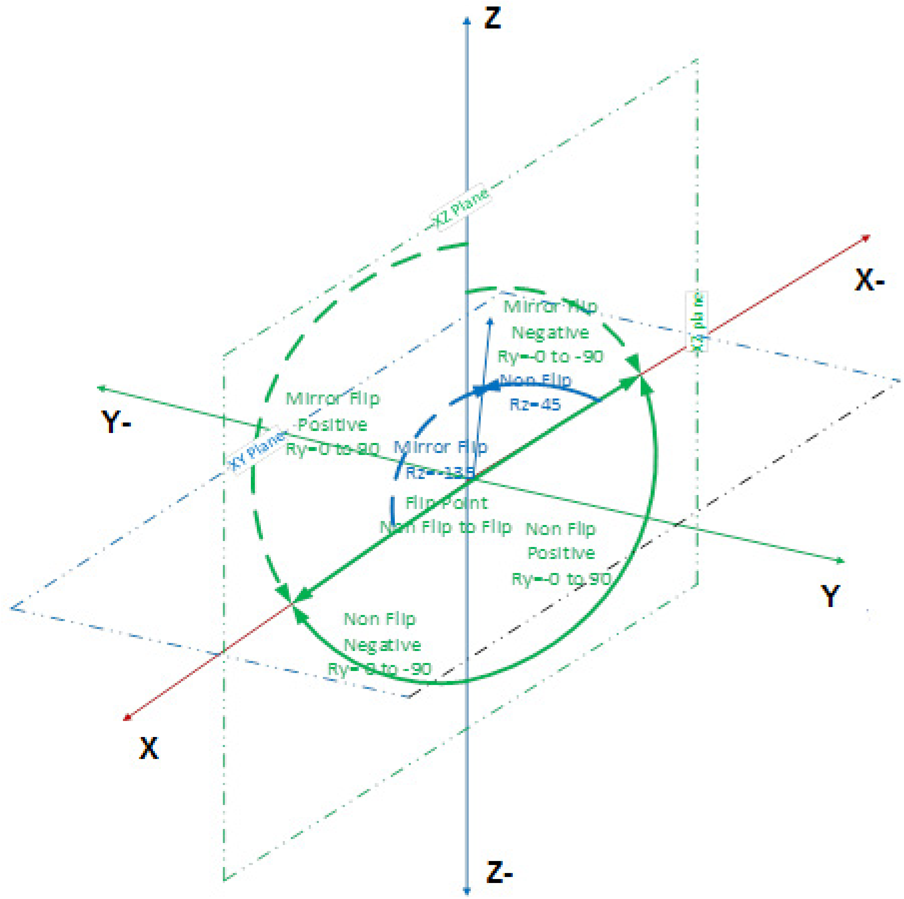
Also, solutions are not available when Ry rotates beyond 90°.

RxRyRz, flip, mirror flip condition

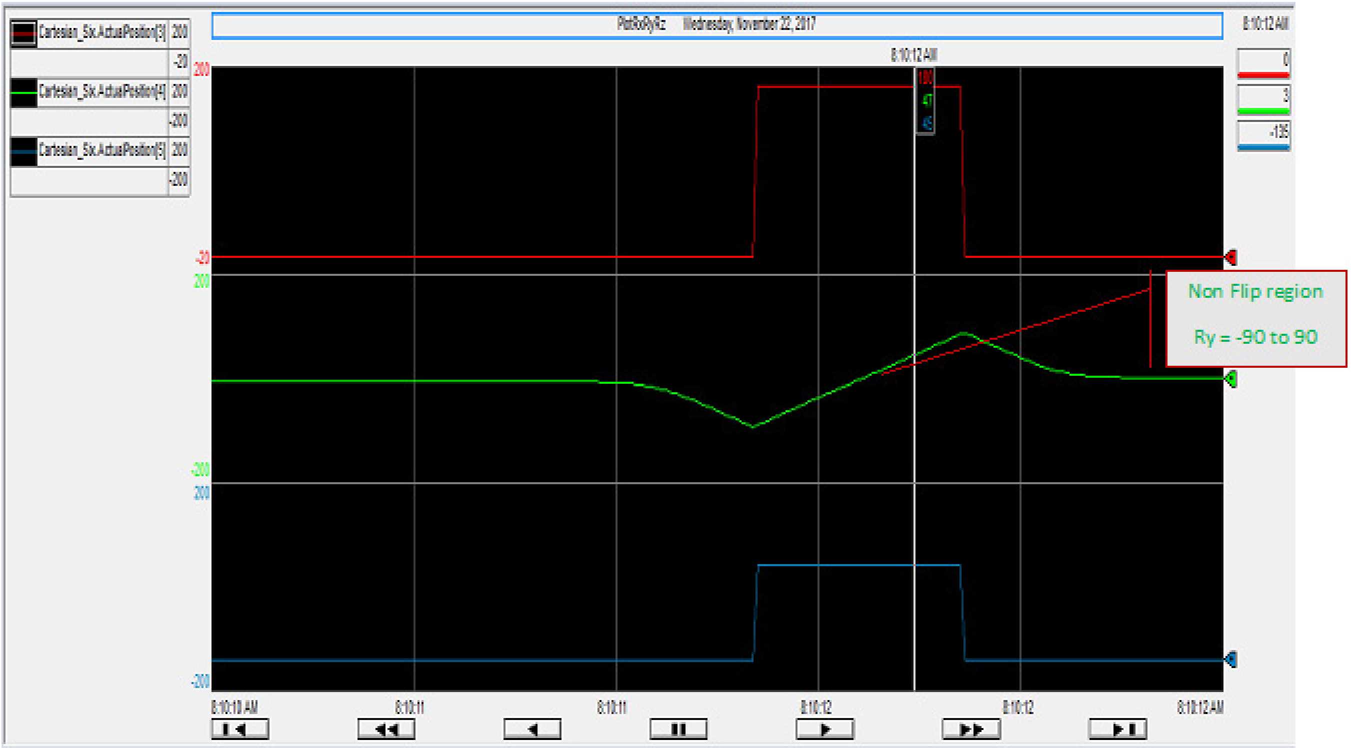
A rotation matrix can be used to rotate Rx, Ry or Rz to any value in the range of +/-180° and obtain the rotation matrix around the base axis. Trigonometric equations can rotate beyond 180° in either direction. They flip to the positive or negative side at the boundary condition of 180°. This behavior is followed in the Logix firmware for Rx and Rz rotations. The Ry rotation needs to follow a different behavior.

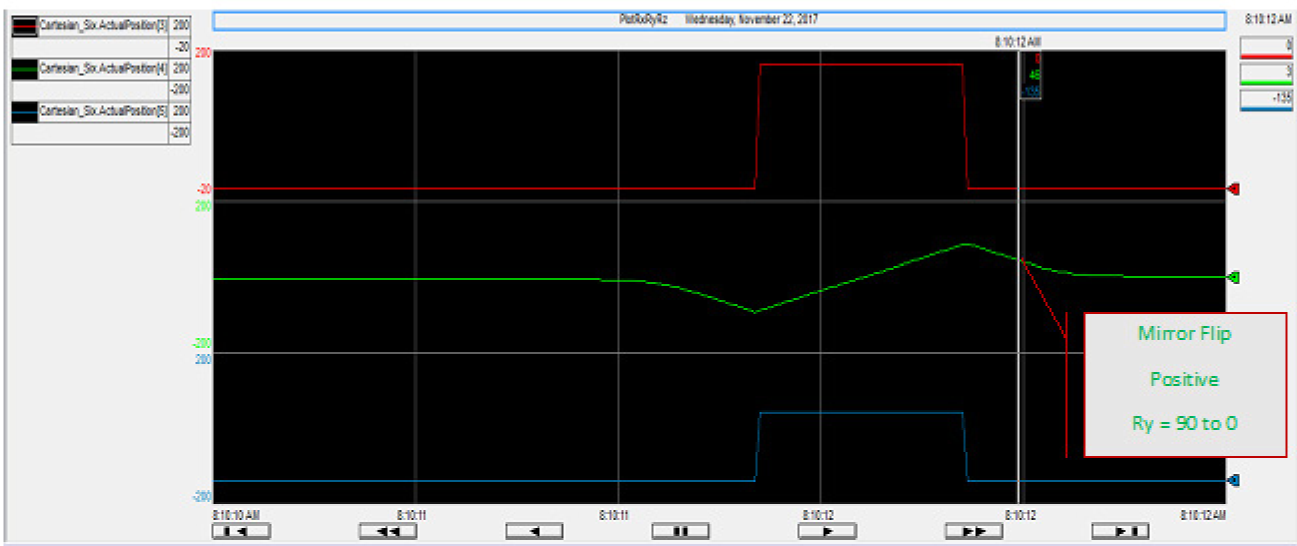
Transforming between the frames sometimes has limitations on computational solutions available. For the XYZ fixed format used by the Logix firmware, certain orientations, such as Ry rotation of 90° or -90°, can result in multiple solutions known as singularity. Also, solutions are not available when Ry rotates beyond 90°. As a result, Ry is restricted to +/-90° and has four regions as shown in the following diagram to handle full rotation of 360° around Y axis. At the 90° point of Ry, the Rx and Rz need to mirror flip as shown in the trends.

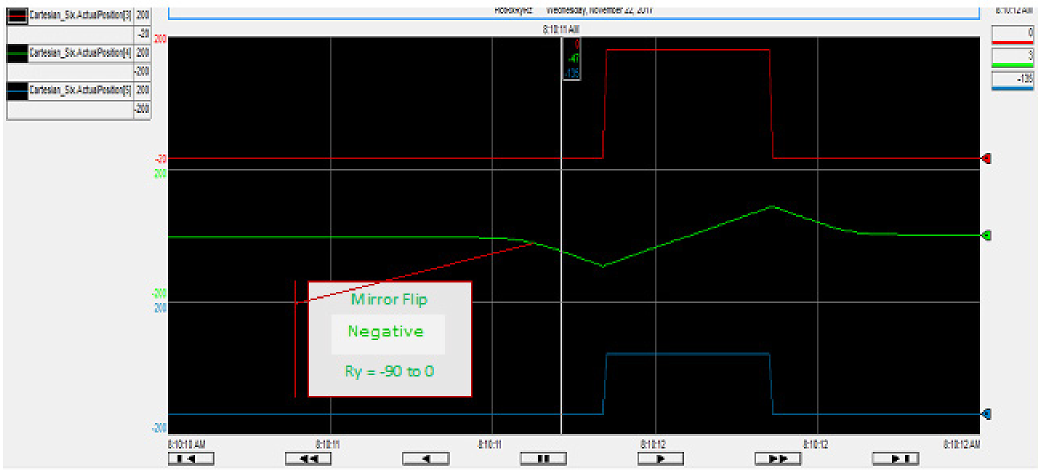
The following is a 3D diagram of a series of points with Ry which has four regions as shown in the diagram. This covers 360° range of rotation around Y axis while restricting Ry to +/-90° using mirror flip implementation. Rz rotation in XY plane flips from 45 to -135.



Tip: For non flip angle Ry is measured with Z- axis and for flip condition angle Ry is measured with Z axis.

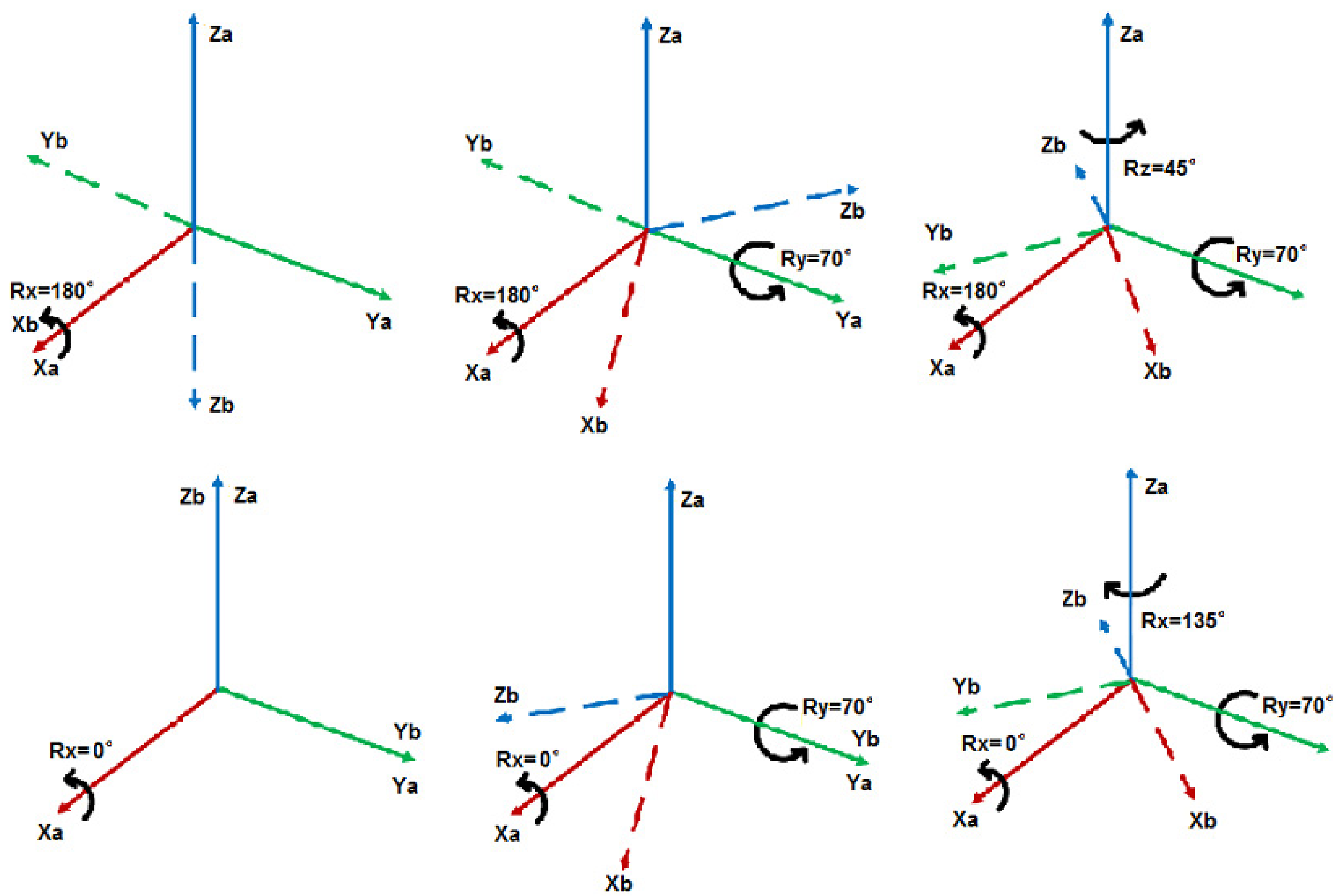

Non Flip region
Ry = -90 to 90


Mirror Flip
Positive
Ry = 90 to 0

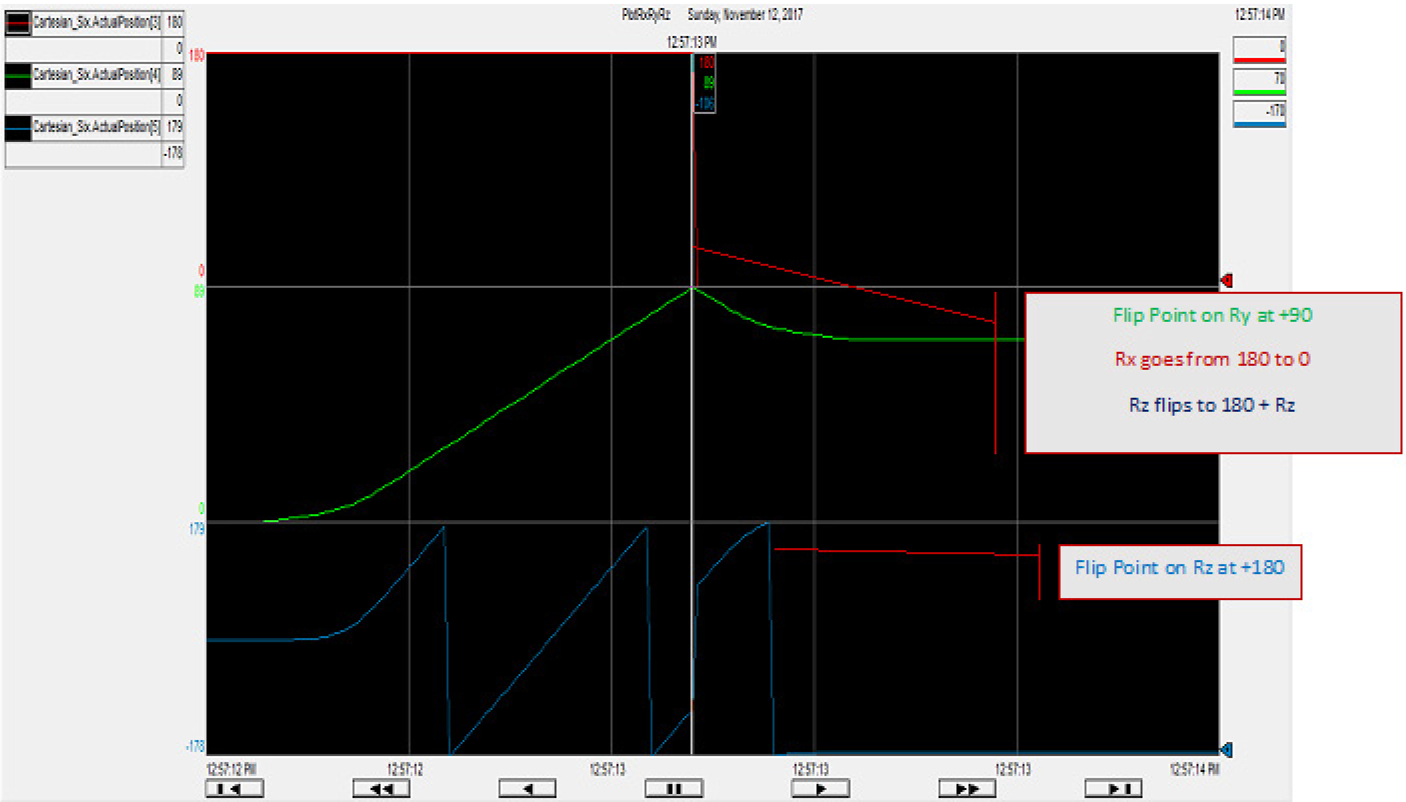


The trends above show the same Ry range in non flip and flip region and Rx (180 to 0) and Rz (45 to -135) transitions at flip points. Ry range goes from -90 to 0 (flip negative) to -90 to 90 (non flip) to 90 to 0 (flip positive) in this example. Ry only has a range of +/- 90° with flip points.

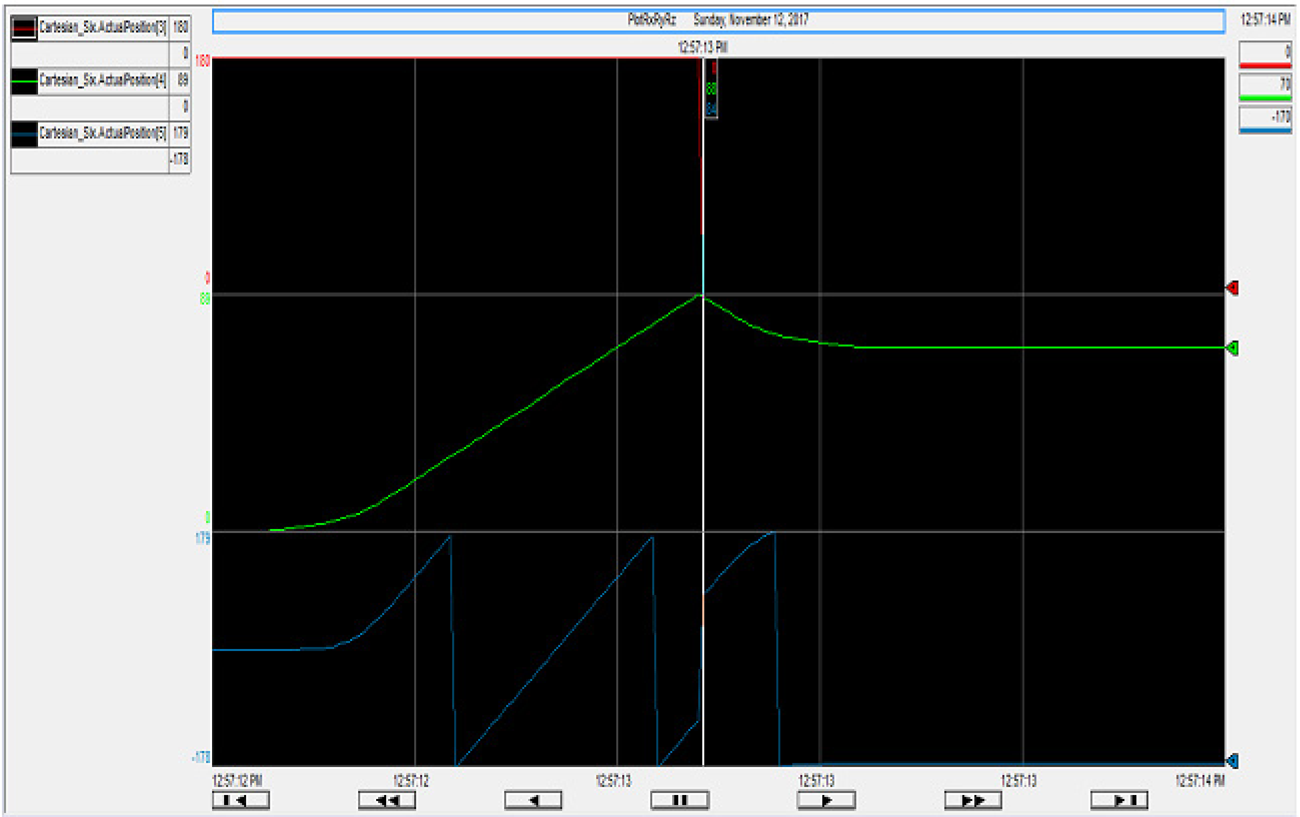
Important:	Even though the trends for Rx, Ry and Rz may look discontinuous, the transformations generate smooth trends for corresponding J4, J5 and J6 axes.



The Ry Mirror Image Point shown on 3D space with fixed angle rotations. [0,0,0,180,70,45] and mirror image [0,0,0,0,70,-135]. The points are the same from orientation point of view at final orientation point but the orientation is achieved by rotating with different sequence. The solid arrows show the fixed frame. Dotted arrows show the orientation frames after each fixed angle rotation.



The Rx Ry Rz Mirror Image Point shown from trends in Logix Designer. The point 180,89,-106 is mirror non-flip condition. Notice that Rz trend shows flip at 180 Rz = 180 and a mirror image flip at Ry = 90. In this example, the Rz moves through multiple turns and has Rz flip points in addition to mirror flip points.

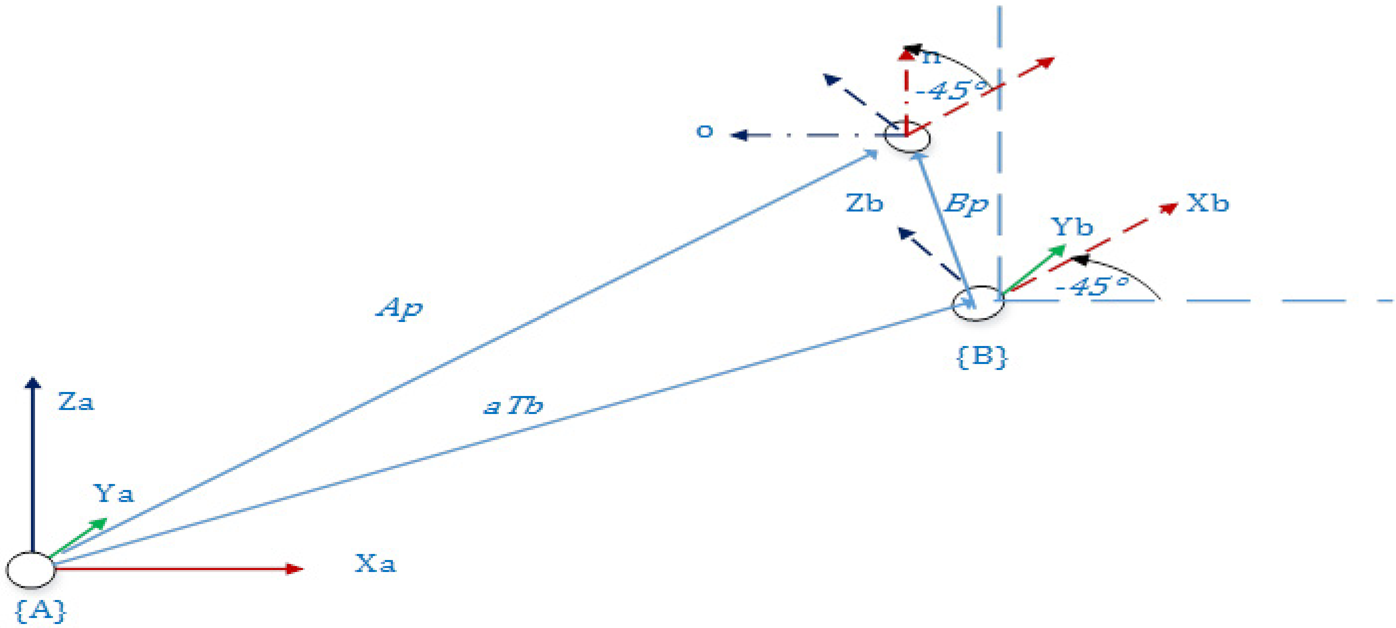


The Rx Ry Rz Mirror Image Point same trend shown from trends in Logix Designer. Rx trend in red, Ry in green and Rz in blue. The point 0,88,84 is mirror flip condition. In this example, the Rz moves through multiple turns and has Rz flip points in addition to mirror flip points.

Translation and rotation example

The following is an example of translation and rotation using user and transform formats.

This diagram uses the combined transform matrix of translation and rotation matrix around the Y axis.



The following diagram uses the combined transform matrix of the translation matrix used with the translation vector of $[5\ 0\ 3]^T$ and rotation matrix of -45° around Y axis.

The transform matrix $^{A}T_{B}$ is:

$$ {}^{A}_{B}T = \begin{bmatrix} \cos(-45) & 0 & \sin(-45) & X \\ 0 & 1 & 0 & 0 \\ -\sin(-45) & 0 & \cos(-45) & Z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.7071 & 0 & -0.7071 & 5 \\ 0 & 1 & 0 & 0 \\ 0.7071 & 0 & 0.7071 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

The translation matrix above can also be represented in user format with X = 5, Y = 0, Z = 3, Rx = 0, Ry = 0, Rz = -45.

The point ^{A}P is with respect to base coordinate frame {A} with the translation vector of $[4\ 0\ 5]^T$ and rotation matrix of 0° rotation or identity matrix.

$$ {}^{A}P = \begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

The point ^{A}P is also specified in user format with X = 4, Y = 0, Z = 5, Rx = 0, Ry = 0, Rz = 0.

The point ^{B}P is with respect to coordinate frame {B} with the translation vector of [-2.1171 0 .7071]^T and rotation matrix of -45° rotation.

$$^{B}P = \begin{bmatrix} \cos(-45) & 0 & \sin(-45) & Xb \\ 0 & 1 & 0 & 0 \\ -\sin(-45) & 0 & \cos(-45) & Zb \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.7071 & 0 & -0.7071 & -2.1171 \\ 0 & 1 & 0 & 0 \\ 0.7071 & 0 & 0.7071 & .7071 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The point ^{B}P is also specified in user format with X = -2.1171, Y = 0, Z = 0.7071, Rx = 0, Ry = 0, Rz = -45.

$$^{A}P \quad = \quad {}^{A}_{B}T \quad \times \quad {}^{B}P$$

$$\begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.7071 & 0 & 0.7071 & 5 \\ 0 & 1 & 0 & 0 \\ -0.7071 & 0 & 0.7071 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0.7071 & 0 & -0.7071 & -2.1171 \\ 0 & 1 & 0 & 0 \\ 0.7071 & 0 & 0.7071 & .7071 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^{A}P = \begin{bmatrix} 0.4999 + 0.4999 & 0 & 0 & -1.4999 + 0.4999 + 5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.4999 + 0.4999 & 1.4999 + 0.4999 + 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Use the matrix representation to convert points from one frame to another frame. It enables computation of the right translation and orientation or pose in the specified frame.

For further information on the methods to determine the point specifications in the example, see the work frame and tool frame topics.

See also

Work Frame example on page 186

Tool frame offsets on page 189

Cartesian Point Specification on page 164

Point Conversion on page 173

RxRyRz, flip, mirror flip condition on page 174

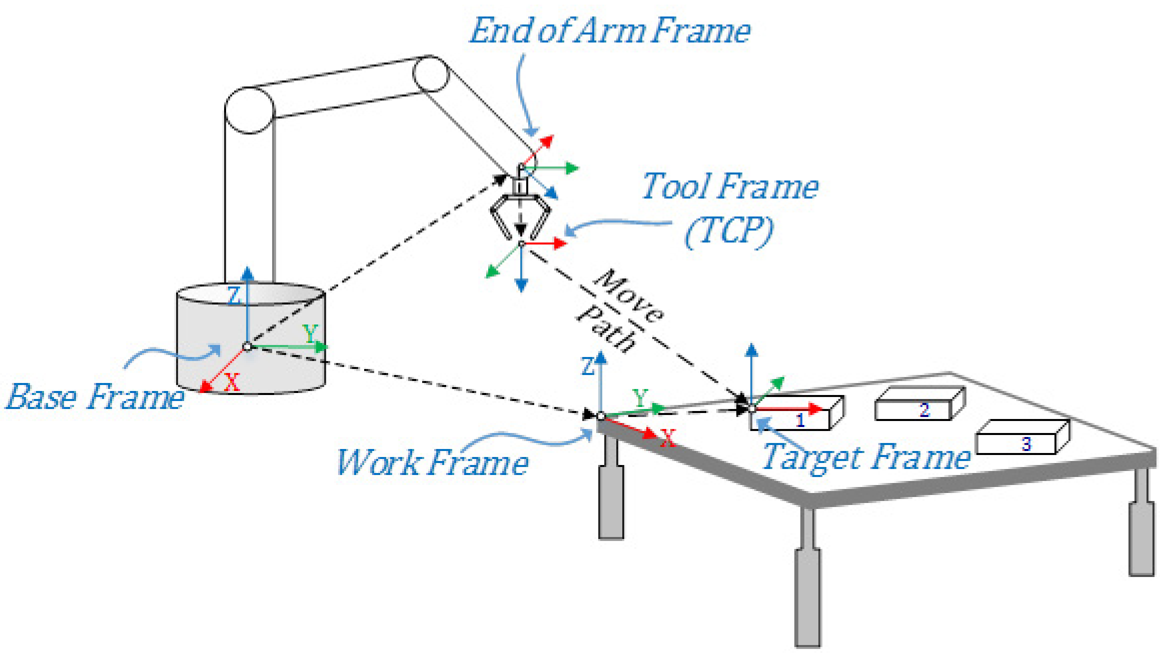
Define coordinate system frames

Studio 5000 kinematics supports these frames for programming different robot applications. Forward and Inverse transformation equations are established for a Cartesian point in space based on frames indicated by the program.

- **Base Frame** - Located at the base of the robot (origin of the robot). End of Arm (EOA) and work frames are measured from the robot's base frame. Refer to the robot geometry specific configuration manuals for establishing the base coordinate system frame.

- **End of Arm Frame** - Located at the last link of the robot and measured from the base frame. Refer to the robot geometry configuration manuals for establishing the end of arm coordinate system frame.
- **Work Frame** - Used when the target positions are measured with respect to a different coordinate frame other than the base coordinate frame of the robot, such as conveyor, vision camera system, and pallets. Define this new reference frame using the work frame offsets. All target positions are measured from the work frames.
- **Tool Frame** - Associated with tools attached at the end of arm of a robot. Define this new tool frame using the tool frame offsets. The tool center point (TCP) is the origin of the tool frame. The Z axis of the tool frame is pointing towards the tool approach vector. The end position of the robot and its movements are always measured related to the TCP.
- **Target Frame** - Represents the various target positions or any object positions programmed for the robot moves in Cartesian space. The target frame is always specified relative to the work frame.

This diagram illustrates simple robot application setup for picking an object from the table using a gripper tool. Reference frames are established from the base frame of the robot for the user program. Boxes are placed on a table at known positions with respect to the table corner, and the table is at a known vector distance or offset from the robot. Table is set as work frame for this application. A gripper is attached at the EOA and tool frame is established at the TCP.



In the diagram, the relationship between different frames are shown using arrow pointing from one origin to another origin of the frame. The arrow direction indicates which way the frames are defined. The end-of-arm frame and work frame are defined from the base frame of the robot. The Tool frame is defined from the end-of-arm frame. All target positions are measured from the work frame using target frames. The Kinematics planner computes the path for TCP from the current position to a target position.

See also

Work frame offsets on page 183

Tool frame offsets on page 189

Configure a Cartesian Coordinate System on page 39

Configure a Delta J1J2J6 Coordinate System on page 210

Configure a Delta J1J2J3J6 Coordinate System on page 223

Configure a Delta J1J2J3J4J5 Coordinate System on page 236

Work frame offsets

The work frame offset is a set of (XYZRxRyRz) coordinate values that redefines the origin of the robot from the new work frame. X, Y, Z represents distance of a work frame from the robot's base frame and Rx, Ry, and Rz represents rotations around those axes.

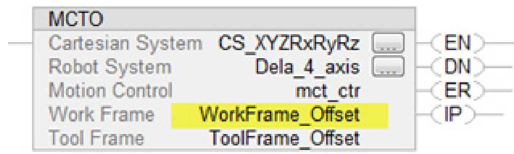
Configure Offset Parameters

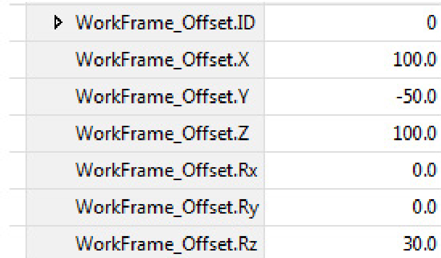
Configure the work frame offsets in the MCTO or MCTPO instruction in Logix Designer application. Measure the offset distance and rotation for the work frame with respect to the base frame. Enter the degrees of rotation offsets into the Rx, Ry, and Rz tag members in units of degrees, and enter the offset distances into the X, Y, and Z tag members in coordination units.

Default values of the work frame offsets are set as (0, 0, 0) for translation and (0, 0, 0) for rotation. These values set the robot's base frame as the default work frame.

Work frame ID helps define multiple work frames using the same tag variable with different ID numbers. Set the ID member to a value greater than or equal to zero.

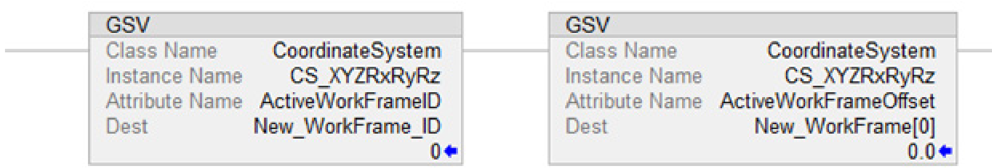
The following image shows the work frame offset configuration in the MCTO instruction and offset values defined for a work frame tag "WorkFrame_Offset".





Status Attributes (ActiveWorkFrameID and ActiveWorkFrameOffset)

- ActiveWorkFrameID and ActiveWorkFrameOffset attributes reflect the information specified in the work frame operand when the MCTO instruction is activated.
- When the MCTO instruction is executed, Work Frame ID and Work Frame Offset members of the Work Frame operand of the MCTO instruction are copied to the ActiveWorkID, ActiveWorkOffset members of the source coordinate system (specified in the MCTO instruction).
- ActiveWorkFrameID will be set to default value as -1 when no work frame is active. It will also be reset to this value when transform instruction terminates. The ActiveWorkFrameOffset values are cleared when the transform instruction terminates.
- These two attributes of the coordinate system are available via GSV instructions as shown in the image below.



For more information about Motion Instructions, see Logix 5000 Controllers Motion Instructions Reference Manual, publication MOTION-RM002.

Restrictions

In some robot geometries, for example Delta robots, due to mechanical constrains some of the work frame orientation offsets are restricted so that the robot cannot be programmed for unreachable positions through the work frame offsets.

The following table shows the current restrictions on the work frame offsets for different robot geometries supported by Logix Designer application.

Geometry Type	Coordinate Definition	Work Frame Offsets					
		X	Y	Z	Rx	Ry	RZ
Delta	J1J2J6	Allowed	Allowed	Allowed	Not Allowed	Not Allowed	Allowed
	J1J2J3J6	Allowed	Allowed	Allowed	Not Allowed	Not Allowed	Allowed
	J1J2J3J4J5	Allowed	Allowed	Allowed	Not Allowed	Not Allowed	Allowed

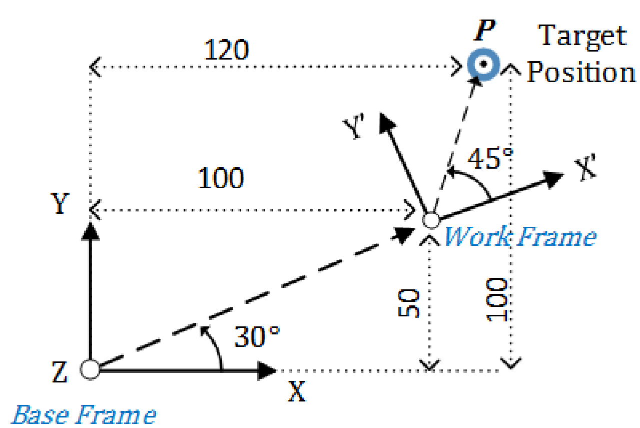
Tip: Offset values must be set to 0° for restricted orientation offset inputs. MCTO/MCTPO instructions generate error #148 for invalid orientation offsets.

Establish a work frame

Following illustration shows an example of establishing a new work frame (X'Y'Z') from the base frame (XYZ) and change in target position P with reference to a new work frame.

Work frame X'Y'Z' is located at 100 units on X axis, 50 units on y axis and rotated 30 degree on Z axis of the robot's base frame XYZ. Work frame offset values are set as (X = 100, Y = 50, Z = 0, Rx = 0, Ry = 0, Rz = 30°).

Assume that the target position (P) is measured as P1 (X = 120, Y = 100, Z = 0, Rx = 0, Ry = 0, Rz = 75°) from the robot's base frame. Now, with respect to a new work frame, target position (P) will change as P2 (X = 42.321, Y = 33.301, Z = 0, Rx = 0, Ry = 0, Rz = 45°).



Position from the Base Frame (P1): (X = 120, Y = 100, Z = 0, Rx = 0, Ry = 0, Rz = 75°)
Work Frame Offsets: (X = 100, Y = 50, Z = 0, Rx = 0, Ry = 0, Rz = 30°)
Position from the Work Frame (P2): (X = 42.321, Y = 33.301, Z = 0, Rx = 0, Ry = 0, Rz = 45°)

See also

Define coordinate system frames on page 181

Work frame examples on page 186

Tool frame offsets on page 189

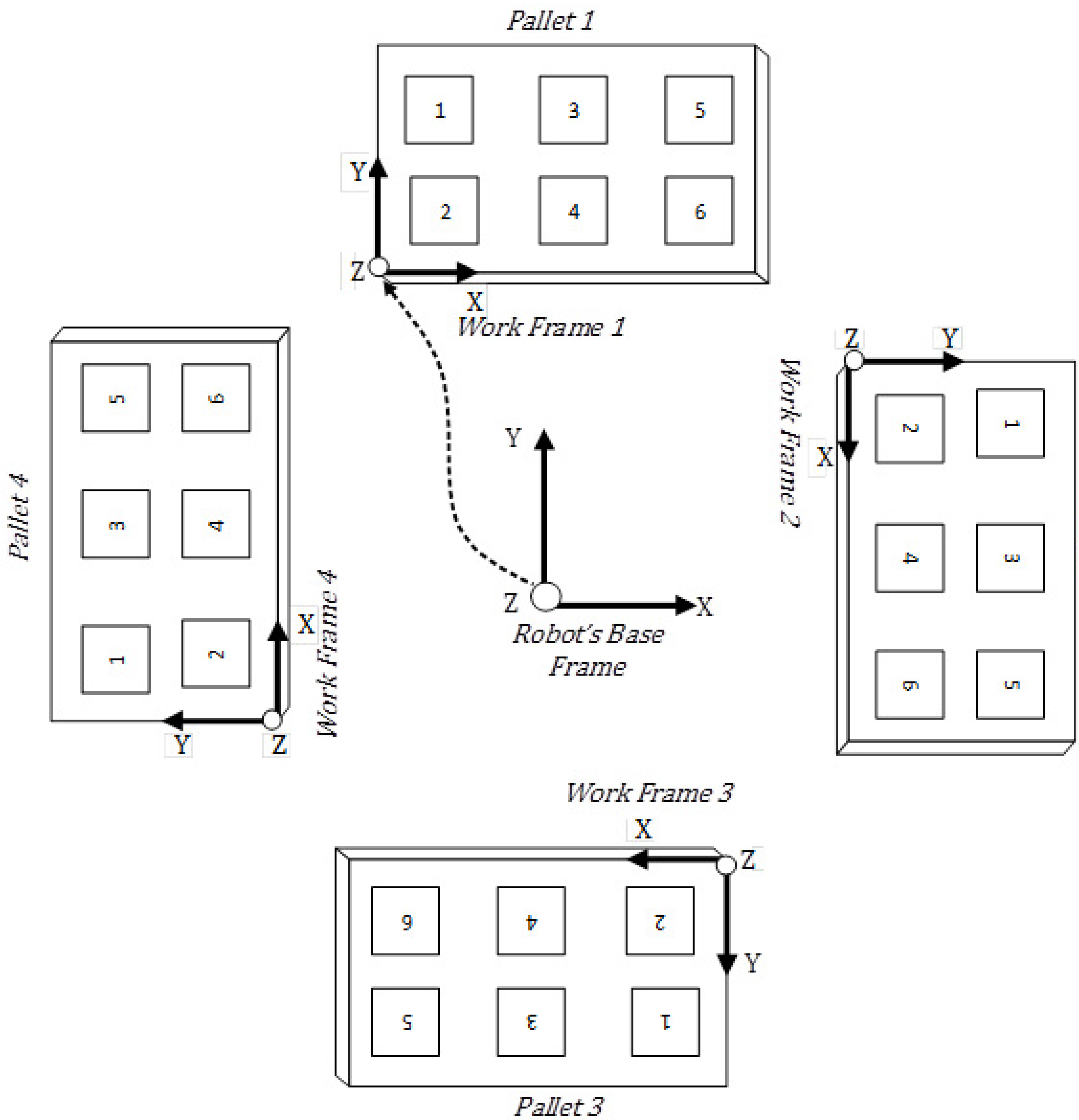
Work frame examples

These examples illustrate how to use work frames in different scenarios.

Multiple work frames with one robot base frame

Use work frames in scenarios where one robot works with multiple work frames or multiple robots work with the same work frames. In this example, the target positions and program remain the same, but the work frame's offsets change based on the different work frame positions.

This diagram illustrates multiple work frames for one robot base frame. The robot is picking six boxes from the Pallet 1 and the positions of all boxes are measured from the Pallet 1. The same pick and place program is used for the other pallets placed at different positions and orientations. Use the MCTO instruction with different work frame offset values and run the same program. The MCTO instruction re-computes the new target positions based on the different work frame offset inputs. For example, the Position of Box-1 is same for all four pallets, but the robot places at different positions and orientations from the robot base frame.



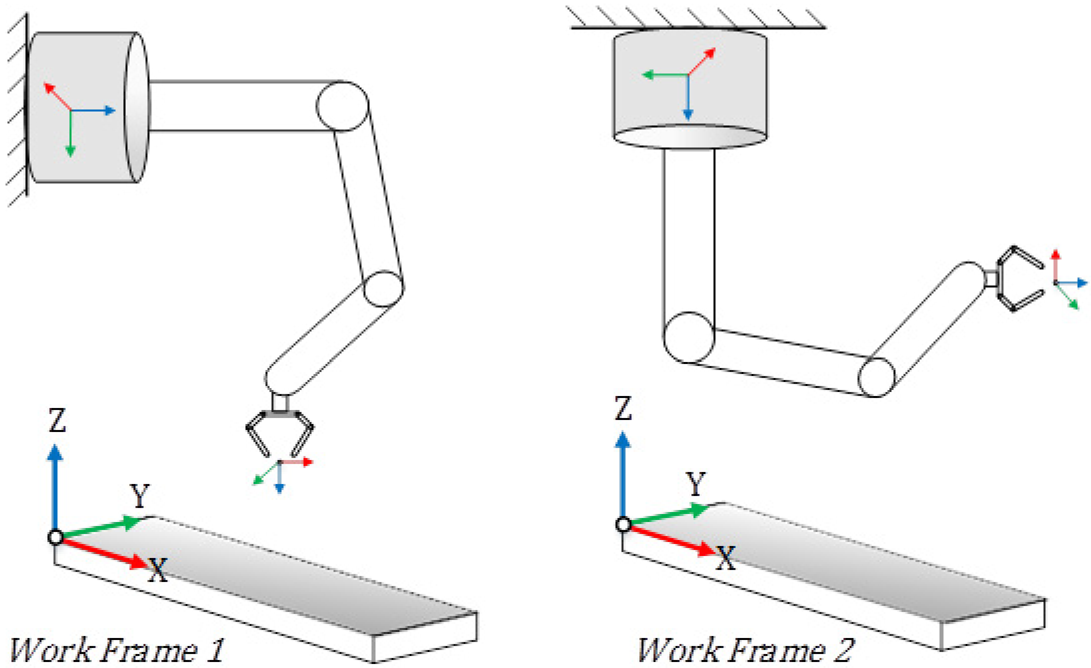
Work Frames	Work ID	Work Frame Offsets					
		X	Y	Z	Rx	Ry	Rz
Work Frame 1	0	-50	100	-800	0	0	0
Work Frame 2	1	100	50	-800	0	0	-90
Work Frame 3	2	50	-100	-800	0	0	180

Work Frames	Work ID	Work Frame Offsets					
Work Frame 4	3	-100	-50	-800	0	0	90

Work frames with different robot positions

It is acceptable to mount robots with different orientations, such as upside down and horizontal positions. Work frame offsets set the relationship between the work frame and the base frames so that programing the target position is convenient for the users.

This diagram illustrates robots mounted in horizontal and upside down positions. Work frame offsets 1 and 2 convert the target positions to conveyor coordinate system assuming it is placed on the ground.



Work Frames	Work ID	Work Frame Offsets					
		X	Y	Z	Rx	Ry	Rz
Work Frame 1	0	100	500	100	90	0	90
Work Frame 2	1	-100	100	500	180	0	90

Tip: To use these Kinematic sample projects, on the **Help** menu, click **Vendor Sample Projects** and then click the **Motion** category. The Rockwell Automation sample project's default location is:
c:\Users\Public\Public Documents\Studio 5000\Sample\ENU\v<current_release>\Rockwell Automation

See also

Define coordinate system frames on page 181

Work frame offsets on page 183

Tool frame offsets on page 189

Tool frame example on page 193

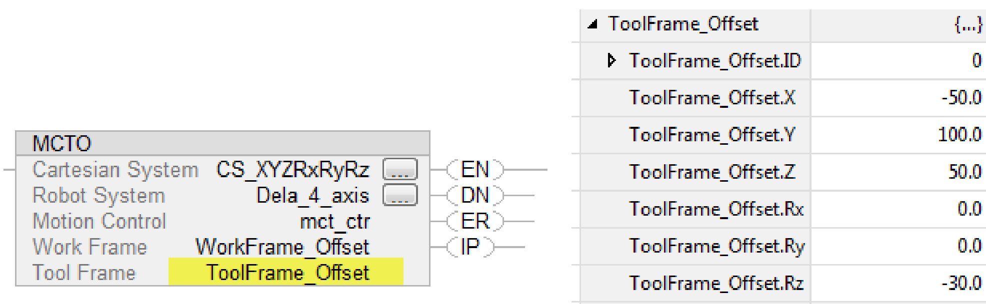
Tool frame offsets

The tool frame offset is a set of (XYZRxRyRz) coordinate values that defines the tool frame at tool center point (TCP) from the End of Arm (EOA) frame. The X,Y,Z represents the translation coordinates that define the TCP from the EOA frame and Rx, Ry, and Rz represents rotations around those axes.

Configure Offset parameters

Configure the tool frame offsets in the MCTO or MCTPO instructions in Logix Designer application. Measure the offset distance and rotation for the tool frame with respect to the robot's EOA frame axes. Enter the degree of rotation offsets into the Rx, Ry, and Rz tag members in units of degrees. Then enter the offset distances into the X, Y, and Z tag members in coordination units.

Default values of the tool frame offsets are set as (0, 0, 0) for translation and (0, 0, 0) for rotation. This sets the EOA frame of the robot as a default TCP point. The Tool Frame ID helps define multiple tools using the same tag variable with different ID numbers. Set the ID member to a value greater than or equal to zero. This image shows the Tool Frame offset configuration in the MCTO instruction and offset values defined for a tool frame tag **ToolFrame_Offset**.

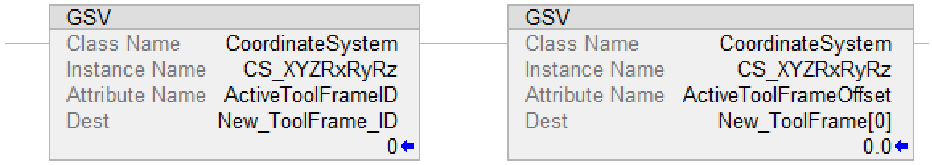


Status Attributes

ActiveToolFrameID and ActiveToolFrameOffset

- **ActiveToolFrameID** and **ActiveToolFrameOffset** attributes reflect the information specified in the tool frame operand when the MCTO instruction activates.
- When the MCTO instruction executes, **Tool Frame ID** and **Tool Frame Offset** members of the **Tool Frame** operand of the MCTO instruction are copied to the **ActiveToolID**, **ActiveToolOffset** members of the source coordinate system as specified in the MCTO instruction.

- **ActiveToolFrameID** is set to default value as -1 when no tool frame is active. It also resets to this value when transform instruction terminates. The **ActiveToolFrameOffset** values are cleared when the transform instruction terminates.
- These two attributes of the coordinate system are exposed to the user through the GSV instructions as shown in this image.



ToolChangeAllowedStatus

- **ToolChangeAllowedStatus** attribute allows the user to change the tool dynamically through the MCTO instruction while coordinated moves are finished or any source axis is in motion through the MAG or MAPC instruction as a slave axis.
- The **ToolChangeAllowed** bit is present in all coordinate systems, and it is set in the source and target coordinate system of an **active MCTO** instruction.
- The bit is set when the MCTO instruction goes IP. It is cleared when any motion is active on source axis or target axis. The bit remains set when output of MAG and MAPC generates motion on any axis associated with source coordinate system of **active MCTO** instruction.
- The ToolChangeAllowed bit is cleared when a MCTO instruction is terminated for any reason, such as MCS, MGS, MGSD, MGSDR, MASR, MASD, and MSF.

Restriction

In robot geometries, such as Delta robots, some of the tool frame orientation offsets are restricted. This prevents programming the robot with unreachable positions through the tool frame offsets.

This table shows the current restrictions on the tool frame offsets for different robot geometries supported by Logix Designer applications.

Geometry Type	Coordinate Definition	Tool Frame Offsets					
		X	Y	Z	Rx	Ry	Rz
Delta	J1J2J6	Allowed	Allowed	Allowed	Not Allowed	Not Allowed	Allowed
	J1J2J3J6	Allowed	Allowed	Allowed	Not Allowed	Not Allowed	Allowed
	J1J2J3J4J5	Allowed	Allowed	Allowed	Not Allowed	Allowed	Not Allowed

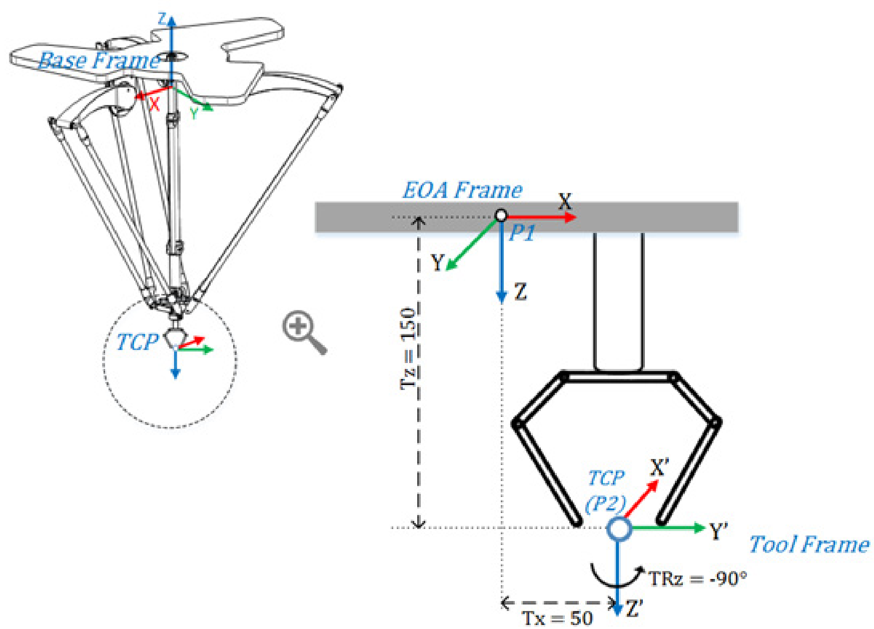
Tip: The offset values must be set to 0° for restricted orientation offset inputs. The MCTO/MCTPO instruction generates error #148 for invalid orientation offsets.

Establish a Tool frame

This diagram illustrates establishing a new Tool frame (X'Y'Z') from the EOA frame (XYZ) and change in the end position P of the robot with reference to a new Tool Frame.

The simple gripper tool is attached at the end plate of 4 axis delta robot. TCP point is measured from the EOA frame of the End plate. The Tool Frame X'Y'Z' is located at 50 units on X axis, 150 units on Z axis, and rotated at -90 degree on Z axis of the EOA frame XYZ. The Tool frame offset values are set as (X = 50, Y = 0, Z = 150, Rx = 0, Ry = 0, Rz = -90°)

Assume that the robot's end position (P) is measured as P1 (X = 0, Y = 0, Z = -800, Rx = 180°, Ry = 0, Rz = 0) from the base frame of the robot to the EOA frame. With respect to a new tool frame, the end position (P) changes as P2 (X = 50, Y = 0, Z = -950, Rx = 180, Ry = 0, Rz = 90°).



End position from the Base Frame (P1): (X = 0, Y = 0, Z = -800, Rx = 180°, Ry = 0, Rz = 0)
Tool Frame Offsets: (Tx =50, Ty = 0, Tz = 150, TRx = 0, TRy = 0, TRz = -90°)
End position with Tool Frame (P2): (X = 50, Y = 0, Z = -950, Rx = 180°, Ry = 0, Rz = 90°)

Refer to the manufacturer CAD drawings or datasheet to find relevant Tool Offset values for the tool.

See also

Define coordinate system frames on page 181

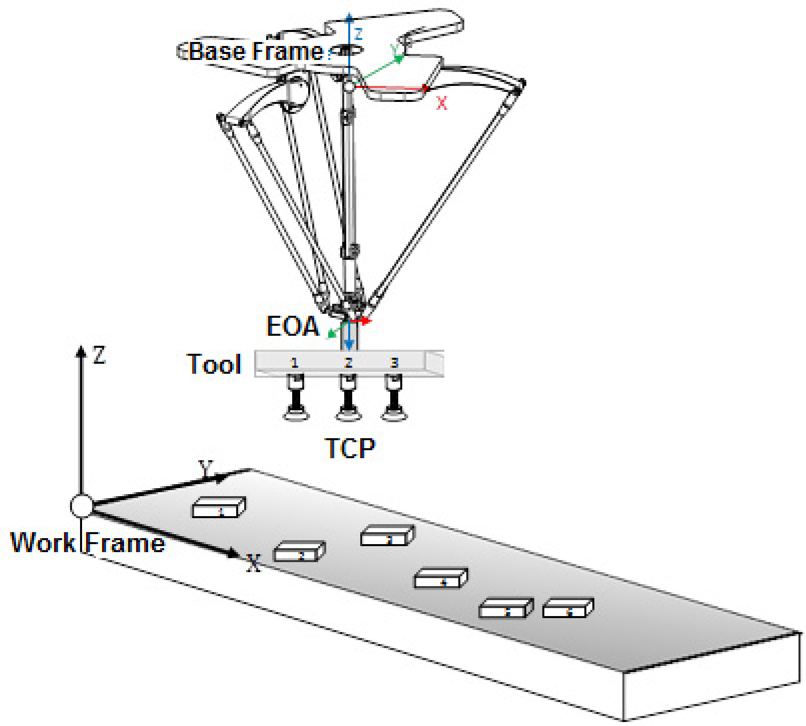
Tool frame example on page 193

Work frame examples on page 186

Work frame offsets on page 183

Tool frame example

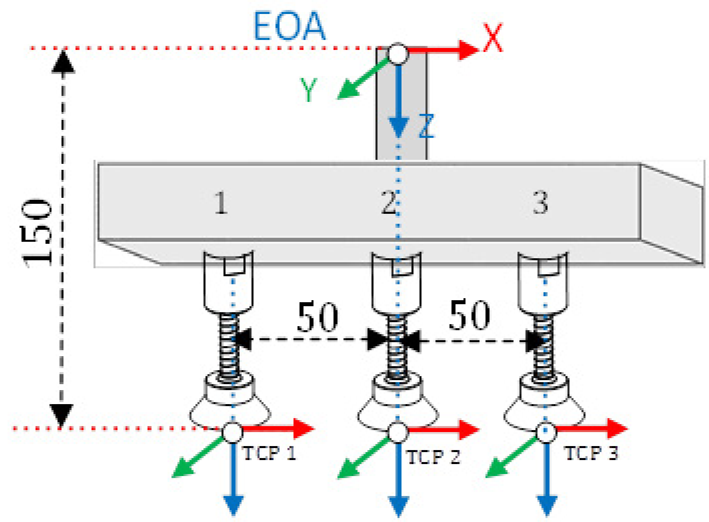
This illustration shows an example of using the Tool Frame in Pick & Place applications. The custom tooling with three grippers (1, 2 and 3) is attached at the end of 4-axis Delta robot. Each gripper is picking an object (1, 2, 3...6), placed at different orientations from the moving conveyor and then putting them in to a box with same orientations.



Each gripper is programmed as a separate tool and tool frames is associated with it. All three TCP positions are measured using the tool offset values shown in the image. Individual tool frames are established through the tool frame offsets shown in the table below.

In the application program, dynamically change the tool using the MCTO instruction, while tracking the conveyor positions using the MAG or MAPC instructions. Initiate the MCTO instruction with the first gripper's tool frame offset values. The robot picks the object using first gripper while the conveyor is moving. When first move is completed, initiate new MCTO instruction with the second gripper's tool frame offsets. The robot picks another object using second gripper.

Tip: Refer to ToolChangeAllowedStatus status bit for dynamically changing the tool frame offsets. If this bit is not set and new MCTO is initiated for tool change then new MCTO will generate #61 with extended error #10. First the MCTO instruction bit (IP) is cleared when the second MCTO is initiated successfully.



Tool Frames	Tool ID	Tool Frame Offsets					
		X	Y	Z	Rx	Ry	Rz
Tool 1	0	-50	0	150	0	0	0
Tool 2	1	0	0	150	0	0	0
Tool 3	2	50	0	150	0	0	0

Tip: To use this Kinematic sample projects, on the **Help** menu, click **Vendor Sample Projects** and then click the **Motion** category. The Rockwell Automation sample project's default location is: **c:\Users\Public\Public Documents\Studio 5000\Sample\ENU\v<current_release>\Rockwell Automation**

See also

Define coordinate system frames on page 181

Tool frame offsets on page 189

Work frame offsets on page 183

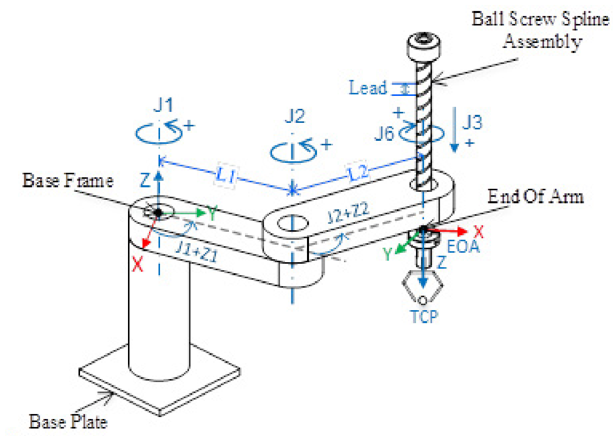
Work frame examples on page 186

Configure the SCARA Independent J1J2J3J6 Coordinate System

This illustration shows a SCARA Independent J1J2J3J6 coordinate system robot. The typical SCARA Independent J1J2J3J6 robot has three revolute joints and one prismatic joint. From base frame, Link 1 (L1) is rigid arm which connects Joint J1/J2 and Link 2 (L2) is also a rigid arm connecting J2/J3/J6. Two independent motors producing coordinated motion at Joint 1 (J1) and Joint 2 (J2) respectively to control the SCARA's X-Y motion. Joint 3 (J3) and Joint 6 (J6) produce Z-Rz motion at the end of arm.

Some of the SCARA geometries have ball screw spline assembly. This assembly can provide linear and rotary motion as well as combined spiral motion, where J3 controls the linear motion in the Z axis and J6 controls the rotational motion.

Use these guidelines when configuring a SCARA Independent J1J2J3J6 robot.



See also

[Configuration Parameters for the Robot on page 198]

[Robot Configuration for SCARA Independent J1J2J3J6 Robot on page 205]

[Maximum Joint Limits condition for SCARA Independent J1J2J3J6 robot on page 209]

[Sample Project for SCARA Independent J1J2J3J6 Robot on page 210]

Establish the reference frame for a SCARA Independent J1J2J3J6 robot

The reference frame is a Cartesian frame, which is the base frame for the robot and all the target points are specified with respect to this base frame. The robot transformations are set up from base frame to end of arm frame to transform any Cartesian target positions into joint space and vice versa. To ensure transformations work correctly, establish the origins for all the axes in the joint space with respect to the robot base Cartesian frame.

The reference frame for the SCARA Independent J1J2J3J6 robot is the base of link L1. The End of Arm (EOA) and the Base Frame are in the same XY plane.

Calibrate the Robot

Use these steps to calibrate a SCARA Independent J1J2J3J6 robot:

1. Obtain the angle values from the robot manufacturer for J1, J2, J3 and J6 at the calibration position. Use these values to establish the reference position.
2. Refer to manufacturer's datasheet to determine if the associated sized motor contains an internal or external gearbox from the motor to actuation at the links or Joints to move the robot.
3. From the **Axis Properties** dialog box - **Scaling** tab, set the gear ratio for each axis in **Transmission Ratio I/O**.
4. In **Scaling,** enter the scaling to apply to each axis (J1, J2, J6), such that one revolution around the Link1 (load rev) equals 360 degrees.

J3 is a linear axis and the units are defined in mm. Use manufacturer's datasheet to convert into motor revolutions.

5. Move all joints to the calibration position for the robot manufacturer by jogging the robot under programmed control or manually moving the robot when the joint axes are in an open loop state.
6. Either:
   - Use the Motion Redefine Position (MRP) instruction to set the positions of the joint axes to the calibration values obtained in Step 1.
   - Set the configuration value for the joint axes home position to the calibration values obtained in step 1 and execute a Motion Axis Home (MAH) instruction for each joint axis.
7. Move each Joint (J1, J2, J3 & J6) to an absolute position of 0.0. Verify that each joint position reads 0 and the respective L1 and L2 are aligned. This establishes the X axis of the base frame and the robot base frame for transformations.
8. If the desired reference position for J1, J2 and J6 axis is different from the transform position zero, then you can use zero angle offsets to adjust the positions for any of the revolute axes J1, J2 and J6.
9. Move J6 to an absolute position of 0.0. Verify that joint position reads 0 and the J6 position is in the Z axis direction of the End of Arm Frame.

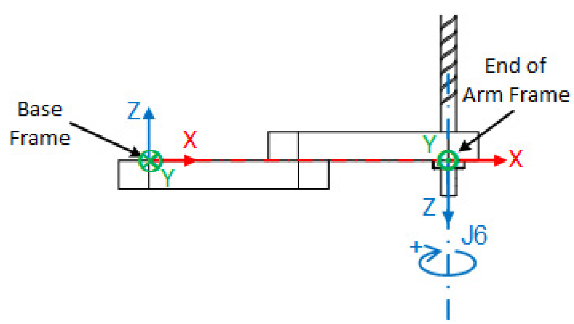
Tip: Since the robot axes are absolute, the reference positions may only need establishing once. If the reference positions are lost, for example, the controller changes, then reestablish the reference positions.

Establish the End of Arm Frame

End of Arm (EOA) in XYZ reference frame is set at the end of the link L2. This frame is rotated by Rx = 180 degrees with reference to the Base frame. As a result, the X axis is in the same direction as Base frame X axis but the Z axis direction is pointing down, towards the direction of the Tool approach vector.

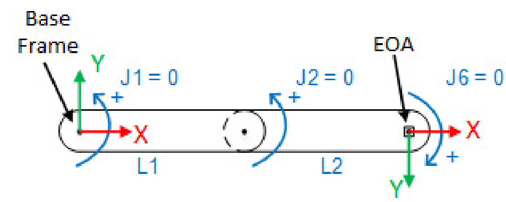
J6 axis of rotation is aligned with the Z axis of Base frame.

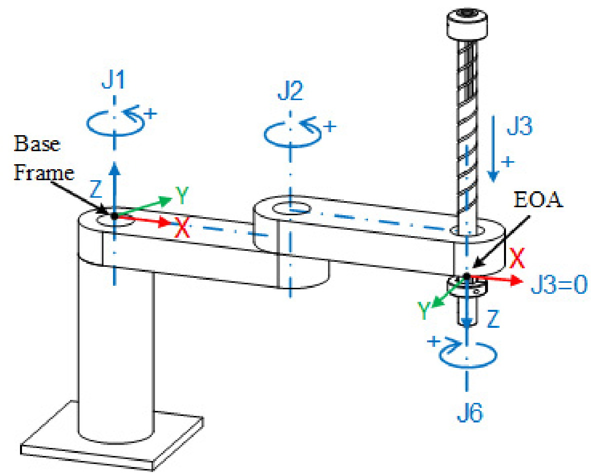
To set the home position for J6 axis, move the J6 axis so that the X axis of EOA is aligned with the link (L1) that is, X axis of Base frame.



Establish the Base Frame

The reference XYZ frame (Base frame) for a SCARA geometry is located near the center of the Joint 1 (J1) axis as shown in theses image. The first diagram shows the top view. The second diagram shows the side view.





+J1 is measured counterclockwise around +Z axis of Base Frame starting at an angle of J1 = 0 when L1 is along the Base Frame X axis.

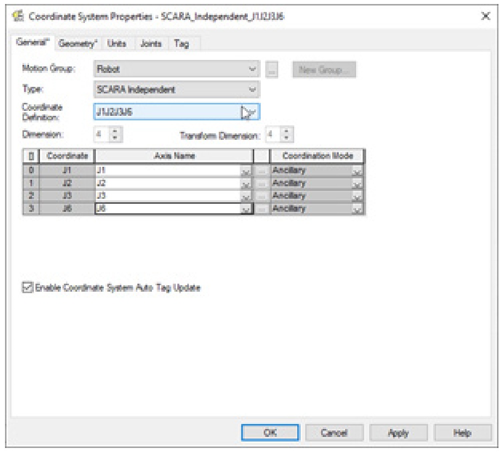
+J2 is measured counterclockwise around the +Z axis at the base frame starting with J2 = 0 when link L2 is aligned with link L1.

+J3 is a prismatic axis that moves in -Z direction of base frame axis. J3 has an absolute zero position at the end of link L2 at the EOA frame, and as it travels in a positive direction, it moves downwards along the Z axis of the EOA frame.

+J6 is measured clockwise around the +Z axis at the Base frame starting with J6 = 0.

When configuring the parameters for the Robot Coordinate System for a SCARA Independent J1J2J3J6 Robot, observe this guideline:

The Dimension and Transform Dimension values are automatically set to 4 and are unavailable to modify because all four axis J1, J2, J3 & J6 are involved in the transformations.



Configuration Parameters for the Robot

Configure the Logix Designer application, to control robots with varying reach and payload capacities. The configuration parameter values for the robot includes:

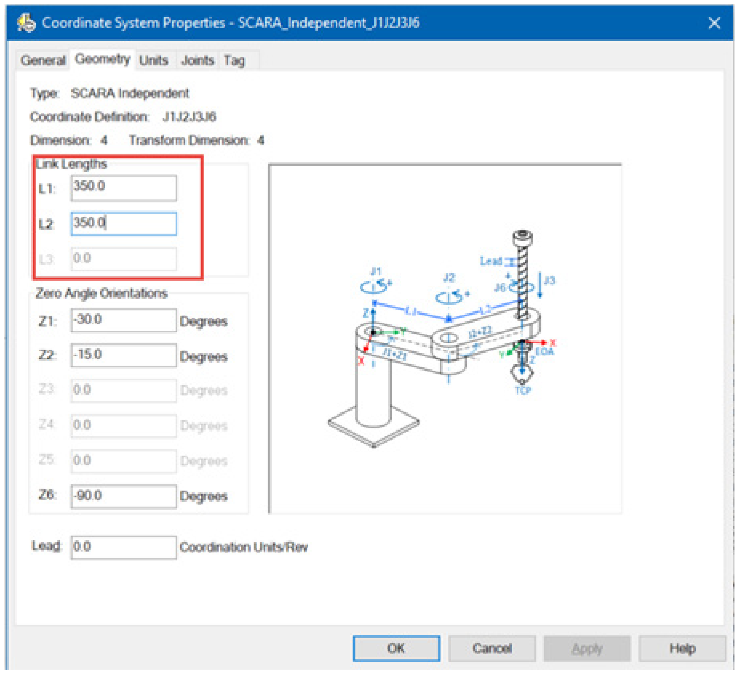
- Link Lengths
- Zero Angle Orientations
- Ball Screw Lead

The configuration parameter information is available from the robot manufacturer.

Tip: Base offsets and end-effector offsets do not apply to a SCARA Independent J1J2J3J6 robot.

Link Lengths for SCARA Independent J1J2J3J6 Robot

Link lengths are the rigid mechanical bodies attached to the joints. Configure **Link Lengths L1** and **L2** in the **Geometry** tab of **Coordinate System Properties** dialog box.



Zero Angle Orientations for SCARA Independent J1J2J3J6 Robot

For SCARA robot geometries, the internal transformation equations in the Logix Designer application assume:

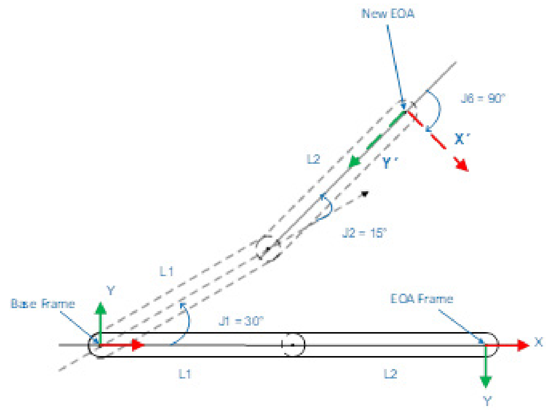
- J1 and J2 are at 0° when link L1 is aligned to L2 along with X axis of the base frame.
- J6 axis of rotation is aligned with Z axis of End of Arm frame (Z axis of End of Arm frame pointing down with respect to base frame) or in parallel with Z axis of base frame when J6 is at 0.

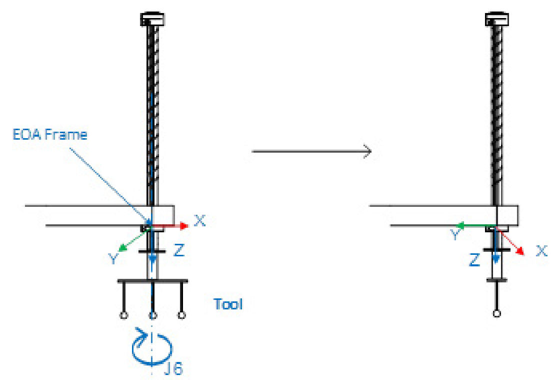
To have joints J1, J2, and J6 angular positions be any value other than 0, configure the **Zero Angle Orientation** values on the **Geometry** tab in the **Coordinate System Properties** dialog box to align the joint angle positions with the internal equations.

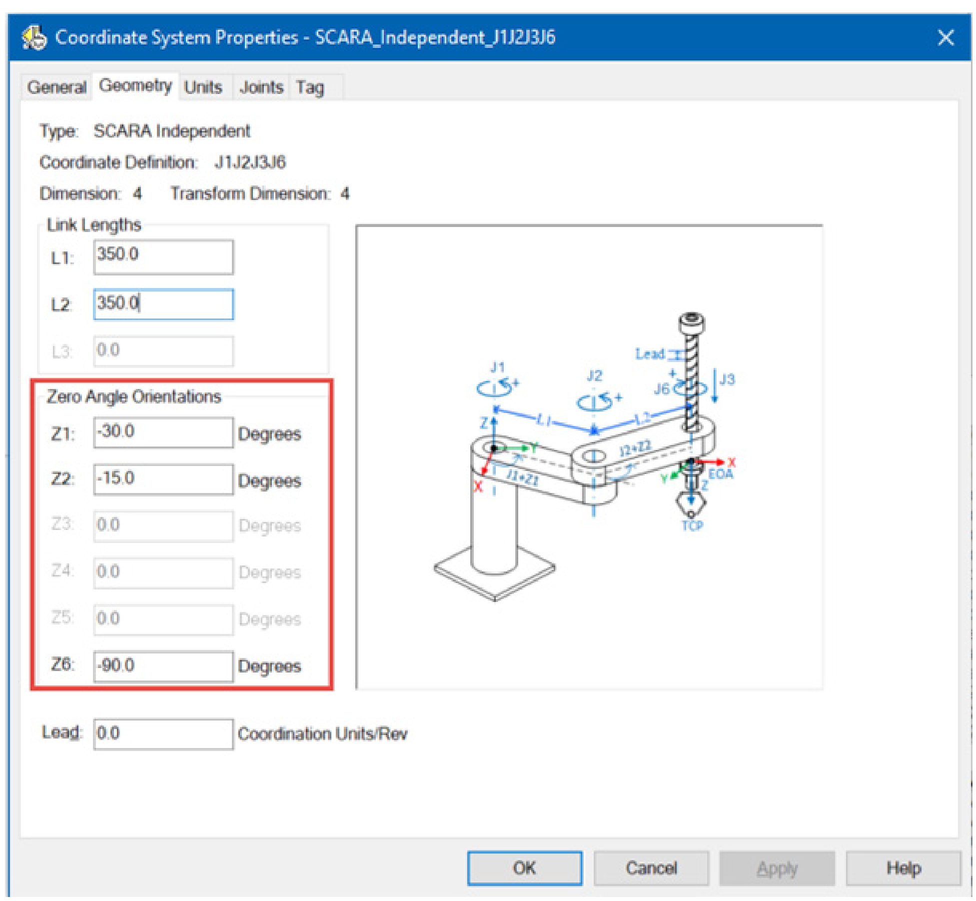
For example:

- Joint J1 is moved by 30° and J2 is moved by 15° from their default home positions and this is the new Home position for J1 and J2. If you need the readout values in the application to be zero in this new position, enter -30° in Z1 and -15° in Z2 parameter on the Geometry tab.
- The Z6 offset is used to set J6 axis home position other than the default 0 position. In this example, the Joint J6 is moved by -90° from its default home position. To get the new home position for J6, we need to set Z6 to -90°.

The first diagram shows the top view with Zero Angle orientation. The second diagram shows the side view of J6 with Zero Angle Offset before and after -90° rotation.

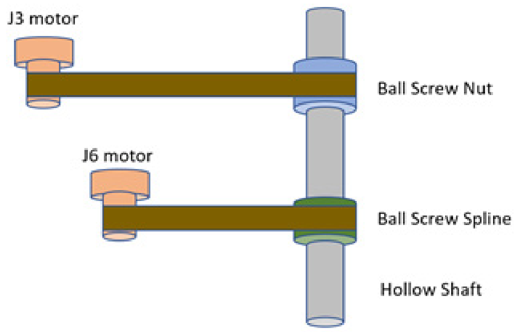






Ball Screw Coupling for SCARA Independent J1J2J3J6 Robot

In Some SCARA robots Ball screw and spline mechanism is used to get rotation and linear movement using a single shaft setup.



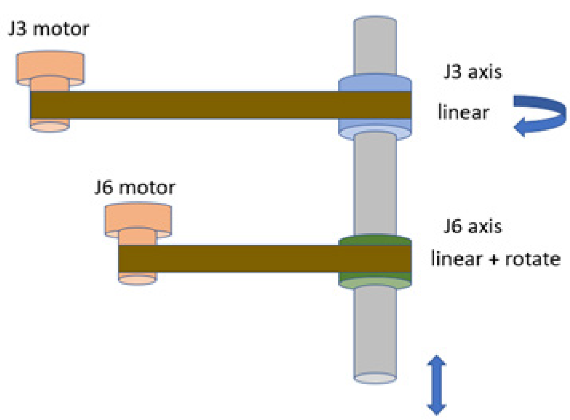
In general, as shown in this image, to control the position and orientation of the Shaft, the Ball Screw Nut and Ball Spline Nut need to work together.

The Ball Screw Nut only introduces linear motion of the shaft (up and down, no rotation), the direction of the movement depends on the thread types of the ball screws. The J3 motor is producing the linear motion by rotating the ball screw nut.

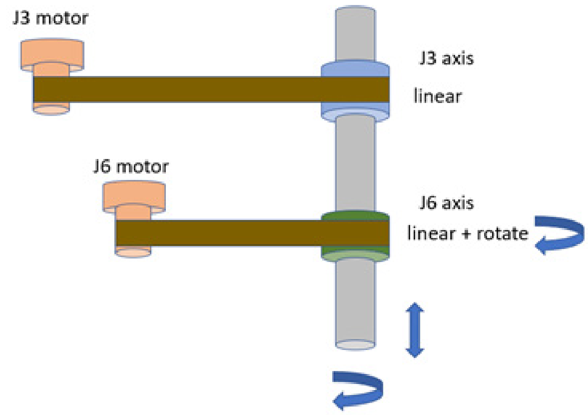
For the Ball Spline Nut, it introduces the rotation of the shaft, and the linear position of the shaft also changes The Ball Spline Nut is rotated by the J6 motor.

In many cases, you would use Ball Screw Nut and Ball Spline Nut together to compensate the linear movement for each other, to introduce the rotation only movement of the Shaft.

For the SCARA robot, in the Logix firmware, J3 is associated with the Ball Screw Nut; and J6 is associated with the Ball Spline Nut.



As shown in the image above, J3 performs linear movement to change the Cartesian Z position of Shaft. To change the linear position of the Shaft only, J3 is used.

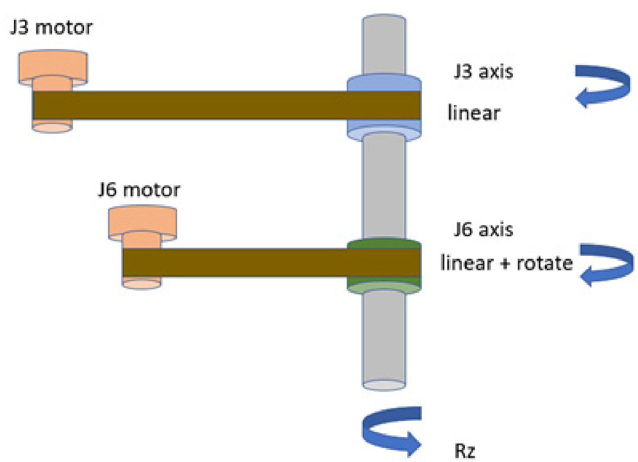


In the image above, J6 rotation introduces the rotation of the shaft, which also causes the linear movement.

The distance of the linear movement caused by the rotation of the shaft is calculated by the **Lead** parameter, the formula is

Lead = Linear Movement Distance / One Revolution of the rotation.

As shown in the image above, J3 can perform linear movement to change the Cartesian Z position of Shaft. To change the linear position of the shaft only, J3 is used.



As shown in the image above, to rotate the shaft only without the linear movement. Move both J3 and J6.

In Studio 5000 Logix Designer application, when only the Rz move is programmed and Z remains the same, the Kinematics transformations in the controller compensate the upward or downward motion caused by the mechanical coupling of the J6 axis by generating opposite movement for J3 axis. The net effect is that you observe only the rotational Rz movement.

These examples address the three scenarios shown in the images.

Assuming **Lead** is 36 mm/revolution, and J3, J6, Z and Rz are all set to 0.

Example 1: Moving J3 only:

If J3 moves up 3 mm, J3 = -3 mm

Z = -J3

= 3 mm

Example 2: Moving J6 only:

If J6 is rotated 30 degree in clockwise.

Rz = -J6 = -30

Z = -J6 * Lead

= -30 * 36/360

= -3 mm

Example 3: Moving Rz only:

If Rz is rotated by 30 deg in clockwise direction.

Rz = -30

then

J6 = -Rz = 30

Since J6 is moved by 30 degree it produces linear movement on Z axis. To compensate this linear move effect J3 needs to move in the opposite direction.

J3 = -J6 * Lead

= - 30 * 36/360

= -3 mm

so

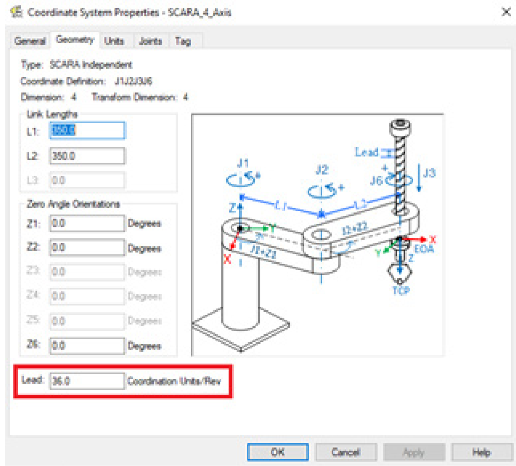
Z = 0

Means there is no linear movement.

These three examples are included in the table.

Joint Configuration (Lead=36 mm/rev)	X	Y	Z	Rx	Ry	Rz
Original Setting, J3 = 0, J6 =0	0	0	0	0	0	0
Example 1: J3 = -3	0	0	3	0	0	0
Example 2: J6 = 30	0	0	-3	0	0	-30
Example 3: J3 = -3, J6 = 30	0	0	0	0	0	-30

The three examples are shown in Studio 5000 Logix Designer.



A SCARA 4 Axis example is shown here.

First, the **Lead** parameter is set to 36.0 Coordination Unit per Revolution.

Rz.CommandPosition	Controller	0.0
J6.CommandPosition	Controller	0.0
J3.CommandPosition	Controller	0.0
Z.CommandPosition	Controller	0.0

And currently as shown in the figure above, in Joint space, J3 = 0 and J6 = 0.

And in cartesian space, Z = 0 and Rz = 0.

First we move J3 to -3 position.

Rz.CommandPosition	Controller	-30.0
J6.CommandPosition	Controller	30.0
J3.CommandPosition	Controller	0.0
Z.CommandPosition	Controller	-3.0

Now Z = -J3 =3, shown in the figure above.

Then, reset all the parameters to 0 and move J6 to 30.

Rz.CommandPosition	Controller	-30.0
J6.CommandPosition	Controller	30.0
J3.CommandPosition	Controller	0.0
Z.CommandPosition	Controller	-3.0

Now in the figure above, Rz = -30 and Z = -3, which is consistent with the results of Example 2.

Reset all the parameters again and move J3 to 3 and J6 to 30.

Rz.CommandPosition	Controller	-30.0
J6.CommandPosition	Controller	30.0
J3.CommandPosition	Controller	-3.0
Z.CommandPosition	Controller	0.0

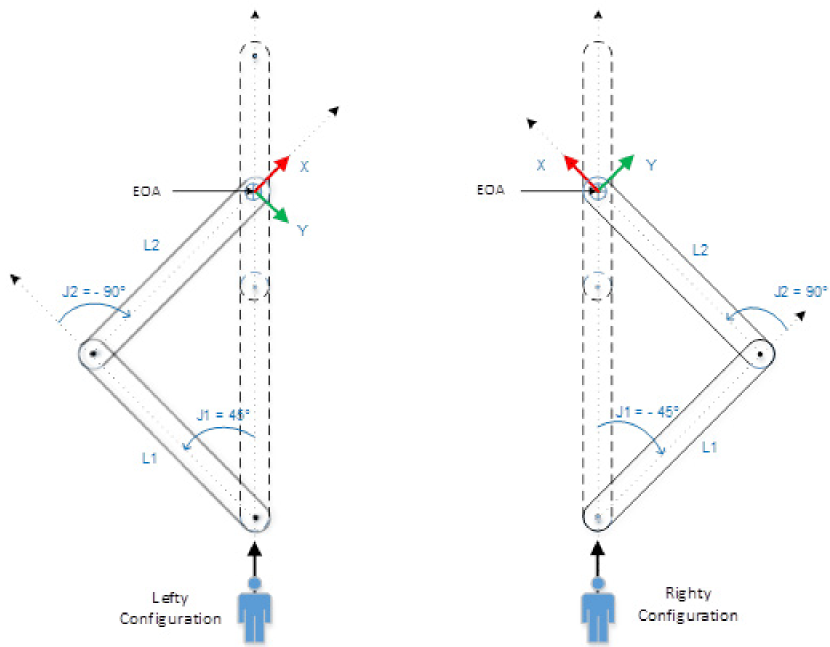
Now in the figure above, Z = 0 and Rz = -30, which is consistent to the Example 3.

Robot Configuration for SCARA Independent J1J2J3J6 Robot

The SCARA robot has two kinematics solutions when attempting to reach a given position.

While achieving a given target position, if J2 is moving in the negative direction with respect to the frame at the end of link L1 (J2 angle is negative), the configuration is considered Lefty Configuration. If J2 is moving in a positive direction with respect to the frame at the end of the link L1 (J2 angle is positive), the configuration is considered Righty Configuration.

The illustration below shows the same cartesian position achieved by Righty and Lefty configuration.



Robot Configuration in MCPM instruction

- When looking at the EOA:
  - If the elbow is to the right, the configuration is Righty.
  - If the elbow is to the left, the configuration is Lefty.
- When MCTO is initiated, it sets Robot Configuration based on current J2 position and while MCTO is active, it remains in the same configuration.
- If MCPM continuous path (CP) move is programmed with a robot configuration parameter that is different from the robot configuration set by the MCTO instruction, it gives error 136.

For Error codes and instruction details refer to the MCPM instruction section.

Robot Configuration in MCTPO instruction

In MCTPO, Bit 0 of the Robot Configuration is ignored. Robot Configuration parameter is input and output parameter for MCTPO instruction which depends on Transform Direction used.

- If MCTPO Transform direction is set to "Forward Transform", then the system computes the Robot Configuration for the user and updates to tag data.

- If MCTPO Transform direction is set to "Inverse Transform" then the user provides Robot Configuration as an input tag.

Robot Configuration is DINT datatype tag and its definition is shown in this table:

Bit Position	3	2	1	0
Description	Flip (1)/ No Flip (0)	Above (1)/ Below (0)	Left (1)/ Right (0)	Change (1)/ Same (0)
Robot configured as Right	N/A	N/A	0	x
Robot configured as Left	N/A	N/A	1	x

Notations:

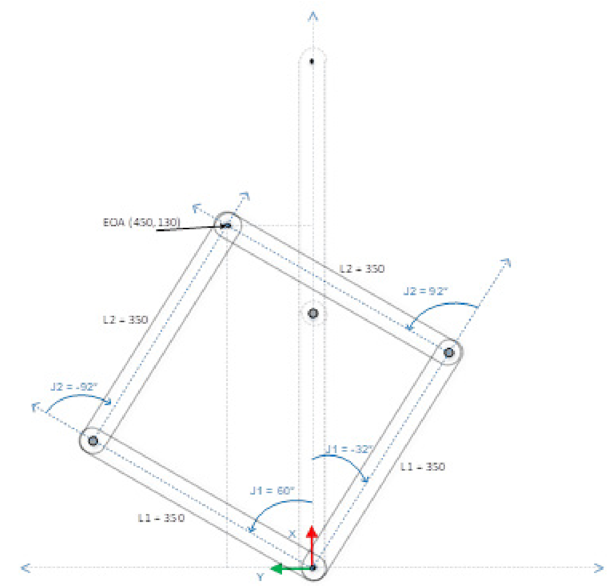
N/A: Not applicable for SCARA J1J2J3J6 Robot.

×: Value is ignored.

For more Error codes and instruction details refer to the MCTPO instruction section.

Robot Configuration Example

For an example, suppose we have L1 and L2 of length 350 units each. The SCARA robot needs to move to the EOA at cartesian coordinates x=450, y=130. The two solutions are shown in this image.

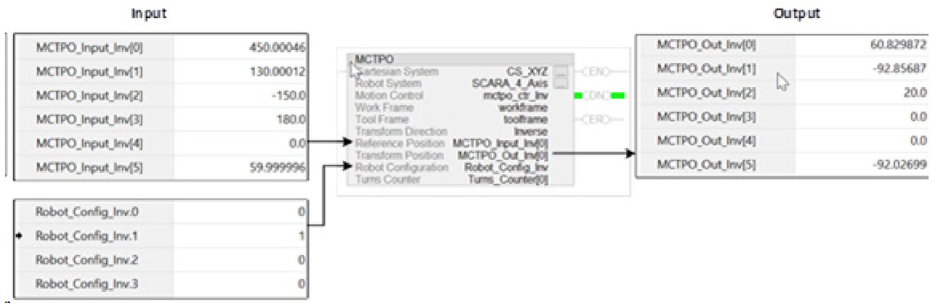


The Studio 5000 Logix Designer application detects a certain Cartesian Position and needs to know the Joint positions with respect to a certain Robot configuration.

This example illustrates an MCTPO instruction with Transform Direction as Inverse, where the user feeds **Cartesian Position** and **Robot Configuration** for Left Configuration as input. The instruction computes the corresponding target joint angle positions and writes the value to the **Transform Position** parameter as the output.

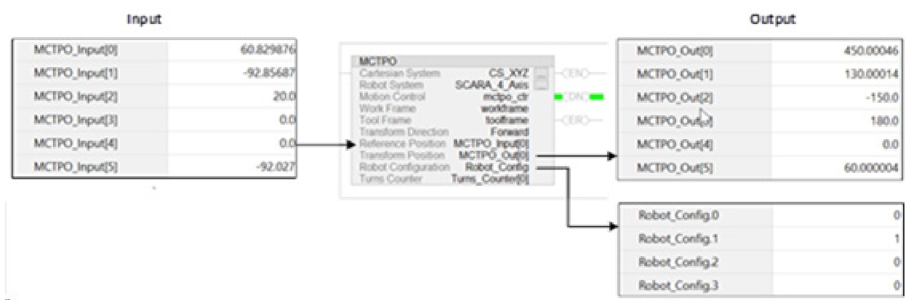
MCPTO1

This example illustrates an MCTPO instruction with Transform Direction as Inverse, where the user feeds **Cartesian Position** and **Robot Configuration** for Right Configuration as input. The instruction computes the corresponding target joint angle positions and writes the value to the **Transform Position** parameter as the output.



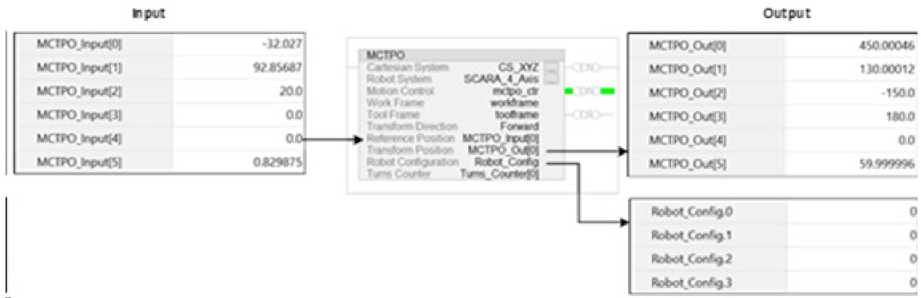
The application knows the Joint Positions and would like to know the Cartesian Position and Robot configuration associated to that Robot position.

This example illustrates the MCTPO instruction with Transform Direction as Forward. The target positions configured are guided into the Reference position operand as input. The instruction computes the corresponding Cartesian positions and Robot Configuration as the output. In this example target positions are evaluated as Left configuration. The target positions configured are guided into the Reference position operand as input. The instruction computes the corresponding Cartesian positions and Robot Configuration as the output. In this example target positions are evaluated as Right configuration.



The Studio 5000 Logix Designer application knows the Joint Positions and would like to know the Cartesian Position and Robot configuration associated to that Robot position.

This example illustrates the MCTPO instruction with Transform Direction as Forward. The target positions configured are guided into the Reference position operand as input. The instruction computes the corresponding Cartesian positions and Robot Configuration as the output. In given example target positions are evaluated as Left configuration.

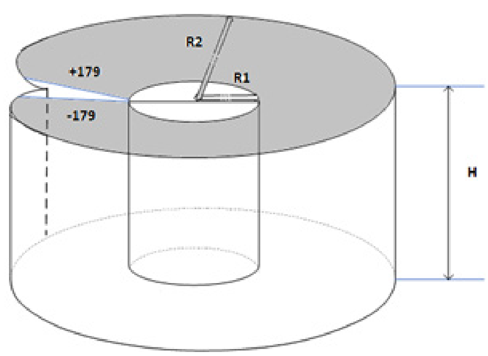


This example illustrates the MCTPO instruction with Transform Direction as Forward. The target positions configured are guided into the Reference position operand as input. The instruction computes the corresponding Cartesian positions and Robot Configuration as the output. In given example target positions are evaluated as Right configuration.

Identify the Work Envelope for the Robot

The work envelope is the three-dimensional region of space that defines the reaching boundaries for the SCARA Independent J1J2J3J6 robot arm. The work envelope for the SCARA Independent J1J2J3J6 Robot is a hollow cylinder with:

- A height (H) equal to the travel limit of the J3 axis.
- An inner radius (R1) equal to |L1-L2|.
- An outer radius (R2) equal to |L1+L2|.



Due to the limited range of motion on individual joints J1 and J2, the work envelope may not be a complete cylinder.

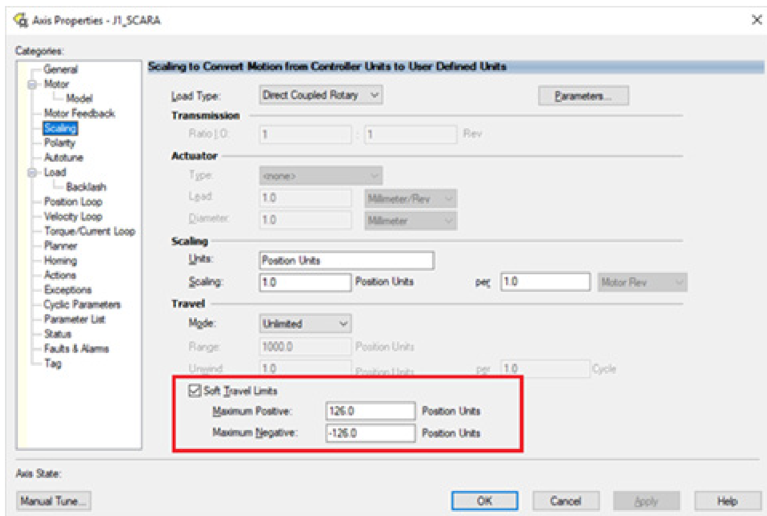
The work envelope for the SCARA Independent J1J2J3J6 robot varies if the tool is attached to the robot. The tool shape and dimensions may modify the work envelope.

Maximum Joint Limits condition for SCARA Independent J1J2J3J6 robot

- The maximum joint limits for configuring Joint 1(J1) and Joint 2(J2) axes is +/-179°. If the joint exceeds the limit, the Motion Coordinated Transform instruction generates an error with error code 151 (JOINT_ANGLE_BEYOND_LIMIT) with the extended error code, specifying which joint exceeds the limit.
- The Joint 3(J3) is a linear axis and does not have any kinematics limits. J3 range depends on the stroke length value provided by manufacturers.
- The Joint 6(J6) axis is the rotational axis that can have multiple turns. The maximum number of turns supported is +/-127. Maximum positive and negative range is checked based on number of turns supported on J6.

Configure the Joint Limits

Additional joint limits are set as a **Soft Travel Limit** on the **Scaling** tab in the **Axis Properties** dialog box.



Work and Tool Frame offset limits for SCARA Independent J1J2J3J6 robot

The Work and Tool Frame offset values defined in the MCTO and MCTPO instruction. SCARA Independent J1J2J3J6 Robot geometry has orientation limitations at the end of the arm, so Work and Tool frame offset values are limited up to reachable work envelope.

These offset values are allowed for Work and Tool frames. The MCTO and MCTPO instructions generates error 148 for invalid offset values.

- Offset values on X, Y, Z and Rz axis are allowed for the Work Frame offsets. Rx and Ry offsets are restricted and must be set to 0. Specify these offsets through the Work Frame parameter in the MCTO and MCTPO instructions.
- Offset values on X, Y, Z and Rz axis are allowed for the Tool Frame offsets. Rx and Ry offsets are restricted and must be set to 0. Specify these offsets through the Tool Frame parameter in the MCTO and MCTPO instructions

Sample Project for SCARA Independent J1J2J3J6 Robot

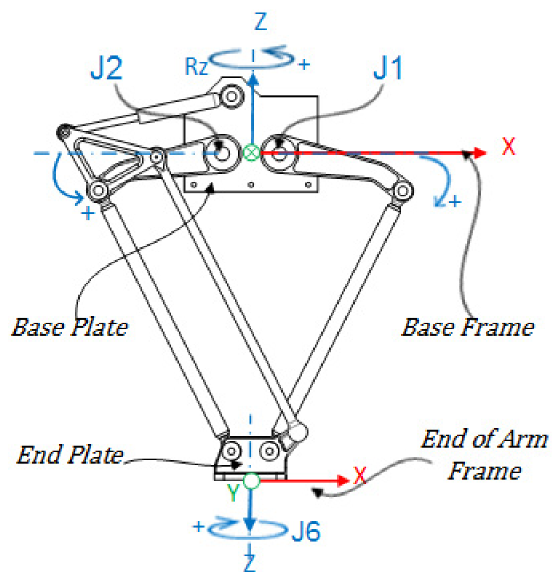
To use the Kinematic sample project on configuring a SCARA Independent J1J2J3J6 Robot, on the Help menu, select **Vendor Sample Projects** and then select the **Motion** category.

The Rockwell Automation sample project's default location is:

c:\Users\Public\Public Documents\Studio 5000\Sample\ENU\v<current_release>\Rockwell Automation

Configure a Delta J1J2J6 Coordinate System

This illustration shows a three-axis Delta robot that moves in three-dimensional Cartesian (X, Z, Rz) space.

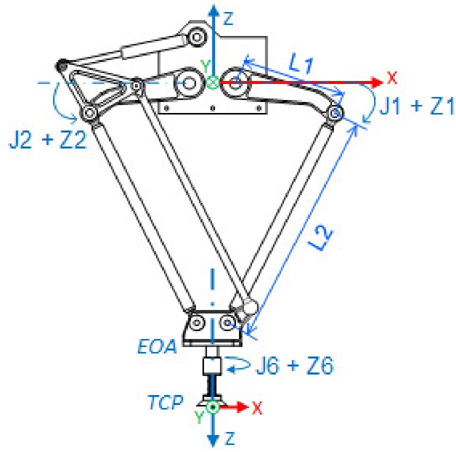


In Logix Designer application, the three-degrees of freedom for this robot are configured as Joint 1 (J1), Joint 2 (J2), and Joint 6 (J6) axes in the robot's coordinate system.

The three joint axes are either:

- Directly programmed in joint space.
- Automatically controlled by the kinematics calculations when instructions are executed in the application, programmed in a virtual Cartesian coordinate system.

This robot contains a fixed top plate (Base Plate) and a moving bottom plate (End Plate). The fixed top plate is attached to the moving bottom plate by two, two link-arm assemblies (L1 and L2) which are identical in mechanical arm lengths.



When joints (J1, J2) are rotated, the arms connected to these joints move in the (X, Z) plane, the mechanical connections of the end plate via spherical joints to the end of second link (L2) ensure that the base and end plates remain parallel to each other.

The J6 is connected at the end of the end plate and provides rotation at the end of the arm. Using the default work and tool frame settings, program the End of Arm (EOA) to a (X, Z, Rz) coordinate. When a tool is attached to the EOA or a different work frame (other than the default) is defined, program the Tool Center Point (TCP) to a full six axis Cartesian point (X, Y, Z, Rx, Ry, Rz). The application computes the joint values (J1, J2, J6) to move the TCP linearly from the current position to the programmed full Cartesian position, using the programmed vector dynamics.

Since there is no rotation on Rx and Ry Orientation axis, Rx orientation value can only be programed to a value of 180°, Ry is always 0°, and Rz orientation values is programed within fixed XYZ Euler Angle range of Rz, within +/-180°

.

See also

Establish a Reference frame on page 212

Configuration parameters on page 214

Identify the Work Envelope on page 218

Maximum Joint Limit condition on page 220

Work and tool frame offset limits on page 222

Invalid Cartesian positions on page 223

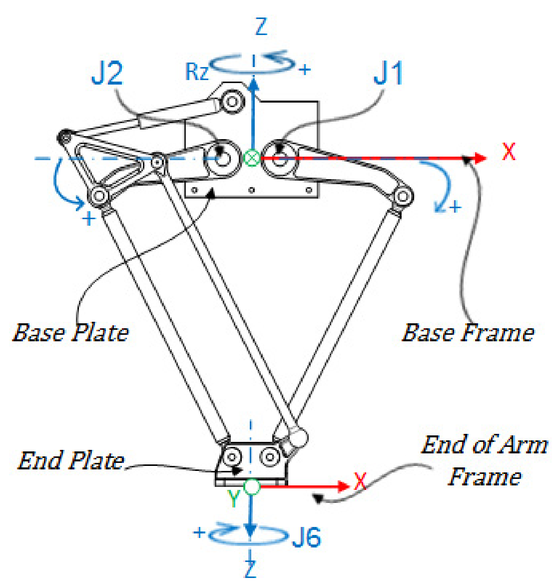
Establish the reference frame for a Delta J1J2J6 robot

The reference frame is a Cartesian frame which is the base frame for the robot and all the target points are specified with respect to this base frame. The robot transformations are set up from base frame to end of arm frame to transform any Cartesian target positions in to joint space and vice versa. In order for the transformations to work correctly, it is required to establish the origins for all the axes in the joint space with respect to the robot base Cartesian frame.

Establish the Base frame

The reference XYZ frame (Base frame) for the Delta geometry is located near the center of the base plate, between Joint 1 and Joint 2, placed 180° apart. Top link of one of the arm is aligned along the positive X axis and the other to negative X axis. Based on the right hand rule, Z axis positive is the axis pointing up (out of the paper in the top view), as shown in the illustration.

- +J1 rotation is measured clockwise around the -Y axis at the Base frame (+Y axis is pointing inside).
- Direction of Joint Axis (J1 and J2) in positive direction causes movement of the top link (associated with J1 or J2 axis) in the downward direction. The two joints are configured as linear axes.
- Directions of Rz orientations at the Base frame as shown in the illustration.



Establish the End of Arm frame

End of Arm (EOA) in XYZ reference frame is set at the end of the End Plate. This frame is rotated by Rx =180° with reference to the Base frame. As a result, the X axis is in the same direction as the Base frame X axis but the Z axis direction is pointing down, towards the direction of Tool approach vector.

J6 axis of rotation is aligned with the Z axis of Base frame.

- To set the home position for J6 axis, move the J6 axis so that the X axis of EOA is aligned with the top link of the arm, that is, the X axis of Base frame.
- +J6 is measured clock wise around the +Z axis at the Base frame.

See also

Calibrate the Delta J1J2J6 robot on page 213

Calibrate a Delta J1J2J6 robot

Use these steps to calibrate a Delta J1J2J6 robot.

To calibrate a Delta J1J2J6 robot:

1. Obtain the angle values from the robot manufacturer for J1, J2, and J6 at the calibration position. Use these values to establish the reference position.
2. Refer to manufacturer's datasheet to determine if the associated sized motor contains an internal or external gearbox from the motor to actuation at the links or Joints to move the robot.
3. On the **Scaling** tab in the **Axis Properties** dialog box, in the **Transmission Ration I/O** box, set the gear ratio for each axis.
4. In the **Scaling** box, enter the scaling to apply to each axis (J1, J2) such that one revolution around the Link1 (load rev) equals 360°.

   The same applies to the J6 axis. One revolution of the J6 axis equals 360°.
5. Move all joints to the calibration position by jogging the robot under programmed control or manually moving the robot when the joint axes are in an open loop state.
6. Do one of the following:

   a. Use the Motion Redefine Position (MRP) instruction to set the positions of the joint axes to the calibration values obtained in step 1.

   b. Set the configuration value for the joint axes home position to the calibration values obtained in step 1 and execute a Motion Axis Home (MAH) instruction for each joint axis.
7. Move each Joint (J1, J2) to an absolute position of 0.0. Verify that each joint position reads 0° and the respective L1 is in a horizontal position (XY Plane).
8. If the top link of arm (L1) is not in a horizontal position, configure the values for the **Zero Angle Offsets** on the **Geometry** tab in the

Coordinate System Properties dialog box to be equal to the values of the joints when in a horizontal position.

9. Move J6 to an absolute position of 0.0. Verify that the joint position reads 0°.

Tip: Since the robot axes are absolute, the reference positions may only need establishing once. If the reference positions are lost, for example, the controller changes, then reestablish the reference positions.

See also

Establish a Reference frame for a Delta J1J2J6 robot on page 212

Configuration parameters for Delta J1J2J6 robot

Configure the Logix Designer application to control robots with varying reach and payload capacities. The configuration parameter values for the robot include:

- Link lengths
- Base offsets
- Effector Plate offsets
- Swing Arm offsets
- Zero Orientation

The configuration parameter information is available from the robot manufacturer.

IMPORTANT Verify that the values for the Link Lengths, Base Offsets, and End-Effector Offsets are entered in the Coordinate System Properties dialog box using the same measurement units.

See also

Link lengths for Delta J1J2J6 robot on page 214

Base and Effector Plate dimensions for Delta J1J2J6 robot on page 215

Swing Arm Offsets for Delta J1J2J6 robot on page 216

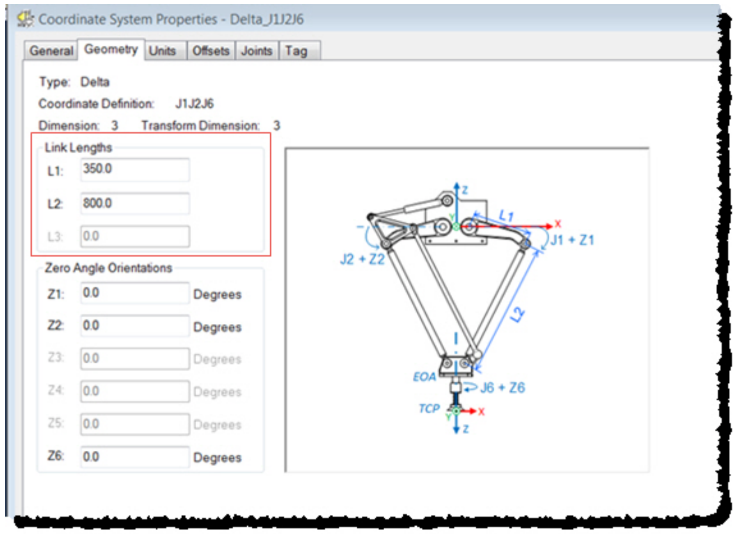
Configure Zero Angle Orientation for Delta J1J2J6 robot on page 217

Link Lengths for Delta J1J2J6 robot

Link lengths are the rigid mechanical bodies attached at the rotational joints. The three-dimensional Delta robot geometry has two link pairs (L1 and L2) that make up of Top link of the arm. Each link pair has the same dimensions.

- **L1** - link attached to each actuated J1 and J2
- **L2** - link attached to L1 on one end and the end plate at the other end

Enter the link lengths on the **Geometry** tab in the **Coordinate System Properties** dialog box.



See also

[Configuration parameters for Delta J1J2J6 robot](#) on [page 214](#)

[Base and Effector Plate dimensions for Delta J1J2J6 robot](#) on [page 215](#)

[Swing Arm Offsets for Delta J1J2J6 robot](#) on [page 216](#)

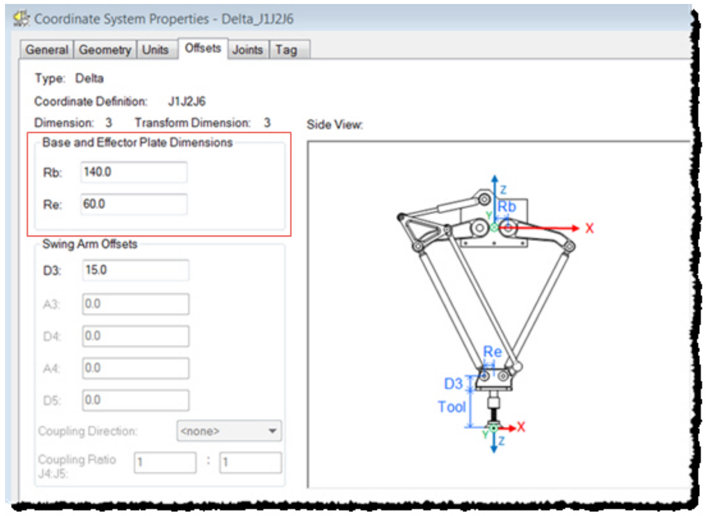
[Configure Zero Angle Orientation for Delta J1J2J6 robot](#) on [page 217](#)

Base and Effector Plate dimensions for Delta J1J2J6 robot

In a 3-axis Delta robot configuration, Base and End plate offsets are represented as **Rb** and **Re** offsets.

- **Rb** - This offset represents the Base plate offset value. Enter the value equal to the distance from the origin of the robot coordinate system to one of the actuator joints.
- **Re** - This offset represents the End plate offset value. Enter the value equal to the distance from the center of the moving end plate to the lower spherical joints of the parallel arms (L2).

On the **Offsets** tab in the **Coordinate System Properties** dialog box, enter the base offset and effector plate offset for the 3-axis Delta robot.



See also

[Configuration parameters for Delta J1J2J6 robot](#) on [page 214](#)

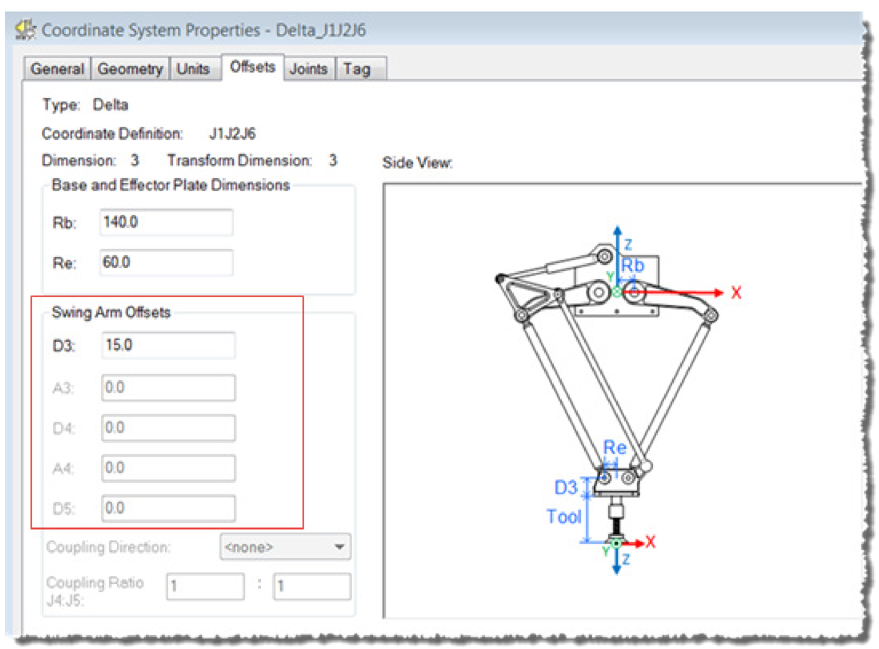
[Swing Arm Offsets for Delta J1J2J6 robot](#) on [page 216](#)

[Configure Zero Angle Orientation Delta J1J2J6 robot](#) on [page 217](#)

[Configuring offset variables in a GSV/SSV instruction](#) on [page 217](#)

Swing Arm Offsets for Delta J1J2J6 robot

Use the **Offsets** tab in the **Coordinate System Properties** dialog box to enter the D3 Swing Arm Offsets value. The **D3** value is the distance on Z axis from the center of end plate to the J6 axis of rotation.



Denavit - Hartenberg (DH) notation is used to configure the offset values. Use XYZ axis directions, shown in the image at end plate center point, as a reference frame to measure the offset values. As per DH convention, Offset values are positive or negative based on XYZ reference frames shown here.

Tip: For all Swing Arm offsets, positive Z direction is pointing down at the End plate center point.

Refer to the manufacturer's CAD drawings or datasheet to find relevant Swing Arm Offset values for the robot. Some offset values will be zero based on the mechanical setup.

See also

Configuration parameters for Delta J1J2J6 robot on page 214

Configure Zero Angle Orientations for Delta J1J2J6 robot on page 217

Configuring offset variables in a GSV/SSV instruction on page 217

Configuring offset variables in a GSV/SSV instruction

The **Offset** parameters in the **Coordinate System Properties** dialog box for the 3-axis Delta robot are not mapped to the attributes of the same name in the GSV/SSV instruction. Use the table to associate the parameters in the **Coordinate System Properties** dialog box to the attributes in the GSV/SSV instruction.

Parameter in Coordinate System dialog box	Class name	Attribute name	Data type	GSV	SSV
Base Plate dimension: Rb	CoordinateSystem	BaseOffset1	REAL	Yes	Yes
Base Plate dimension: Re	CoordinateSystem	EndEffectorOffset1	REAL	Yes	Yes
Swing Arm Offset: D3	CoordinateSystem	EndEffectorOffset3	REAL	Yes	Yes

See also

Base and Effector Plate dimensions for Delta J1J2J6 robot on page 215

Swing Arm Offsets for Delta J1J2J6 robot on page 216

Configure Zero Angle Orientations for Delta J1J2J6 robot

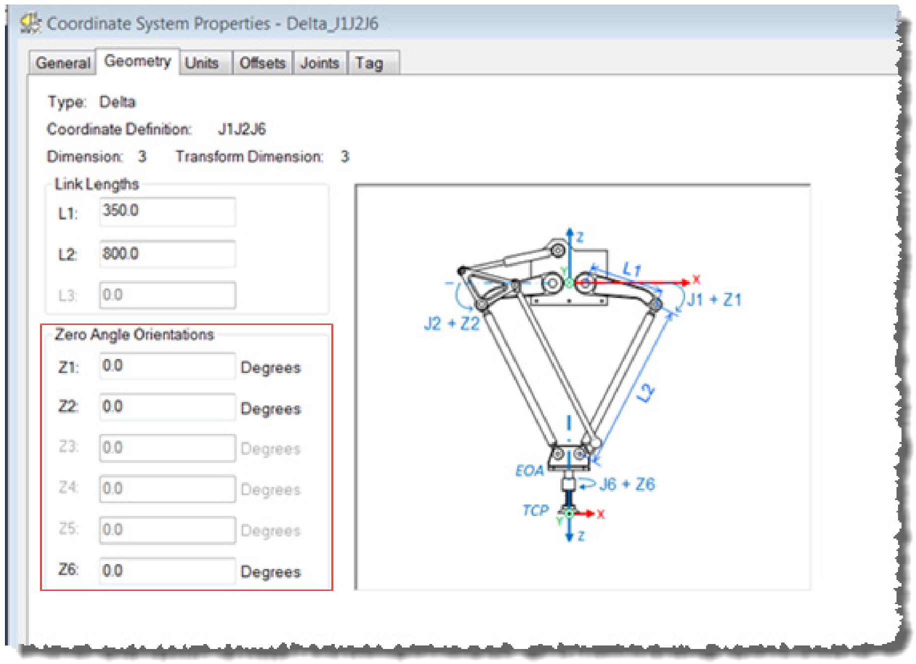
For Delta robot geometries, the internal transformation equations in the Logix Designer application assume:

- J1 and J2 are at 0° when link L1 is horizontal, parallel to XY plane.
- As each top link (L1) moves downward, its corresponding joint axis (J1 or J2) is rotating in the positive direction.
- Joint 6 axis of rotation is aligned with Z axis of base frame when J6 is at 0°.
- End of Arm (EOA) frame has Rx value of 180° with respect to base frame that results in Z axis pointing downward.

To have joints J1 and J2 angular positions to be any value other than 0° when L1 is horizontal, then configure the **Zero Angle Orientation** values on the

Geometry tab in the **Coordinate System Properties** dialog box to align the joint angle positions with the internal equations.

For example, if the Delta robot is mounted so that the joints attached at the top plate are homed at 30° in the positive direction below horizontal and you want the readout values in the application to be zero in this position, then enter -30° in the **Z1** and **Z2** parameters on the **Geometry** tab. The **Z6** offset is used to set J6 axis other than default 0° position.



See also

[Configuration parameters for Delta J1J2J6 robot](#) on page 214

Link lengths on page 214

Base and Effector Plate dimensions on page 215

Swing Arm Offsets on page 216

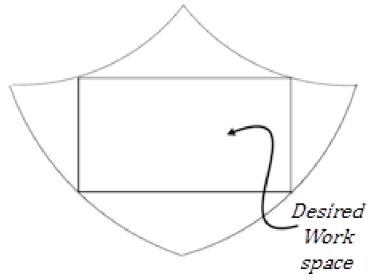
Identify the work envelope for Delta J1J2J6 robot

For Delta robot geometries, the internal transformation equations in the Logix Designer application assume:

- Joints (J1, J2) are at 0° when link L1 is horizontal, parallel to XY

The work envelope is the two-dimensional region of space that defines the reaching boundaries for the robot arm (using the default work and tool frame settings). The typical work envelope for a Delta robot looks similar to a two dimensional inverted umbrella, as shown in this example:

Example of a two-dimensional Delta robot workspace



For exact workspace region, refer to the documentation provided by the robot manufacturer.

Program the robot within a rectangle (desired workspace) defined inside the robot's work space. The rectangle is defined by the positive and negative dimensions of the X, Z virtual source axes.

To avoid issues with the singularity positions, the Motion Coordinated Transform with Orientation (MCTO) instruction internally calculates the joint limits for the Delta robot geometries. When an MCTO instruction is invoked for the first time, the maximum positive and maximum negative joint limits are internally calculated based upon the Link Lengths and Offset values entered on the **Geometry** and **Offsets** tabs of the **Coordinate System Properties** dialog box.

For more information about the maximum positive and maximum negative joint limits, refer to:

- Maximum Joint Limit Conditions
- Work and Tool Frame Offset Limits

During each scan, the joint positions are checked to ensure that they are within the maximum and minimum joint limits.

Homing or moving a joint axis to a position beyond a computed joint limit and then invoking an MCTO instruction results in an error 67 (Invalid Transform position). For more information regarding error codes, refer to the MCTO instruction in the online help or the Logix 5000 Controllers Motion Instructions Reference Manual, publication MOTION-RM002.

See also

Maximum Joint Limit condition for Delta J1J2J6 robot on page 220

Work and Tool Frame Offset limits for Delta J1J2J6 robot on page 222

Link length for Delta J1J2J6 robot on page 214

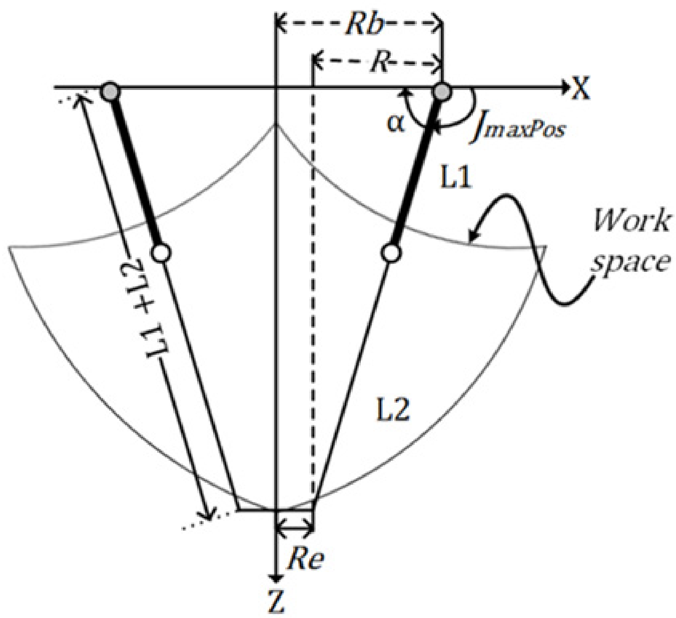
Base and Effector Plate dimension for Delta J1J2J6 robot on page 215

Maximum joint limit condition for Delta J1J2J6 robot

Use these guidelines to determine the maximum joint limit conditions for the four-dimensional robot.

Maximum J1, J2 Positive joint limit condition

The derivations for the maximum positive joint apply to the condition when L1 and L2 are collinear.



Maximum Positive Joint Limit Position

R = absolute value of (Rb - Re)

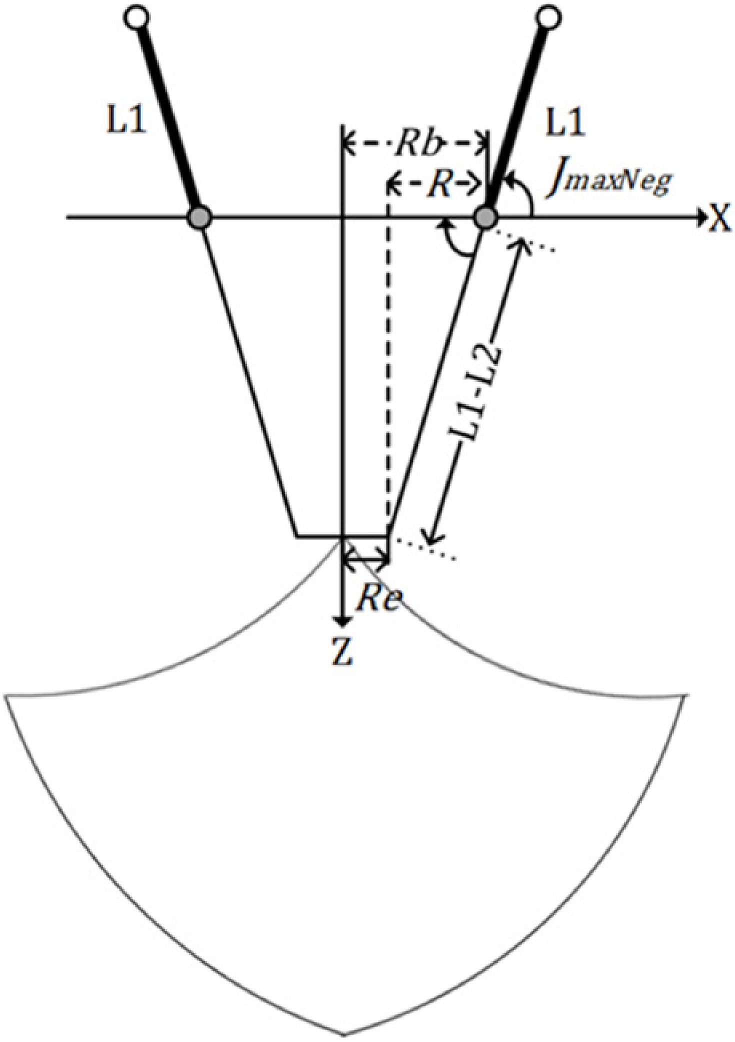
$$\alpha = \cos^{-1}\left(\frac{R}{L1 + L2}\right)$$

$$JmaxPos = 180 - \alpha$$

Maximum J1, J2 Negative joint limit condition

The derivations for the maximum negative joint limit apply to the condition when L1 and L2 are folded back on top of each other.

R is computed by using the base and end-effector offsets values (Rb and Re).



Maximum Negative Joint Limit Condition

R = absolute value of (Rb - Re)

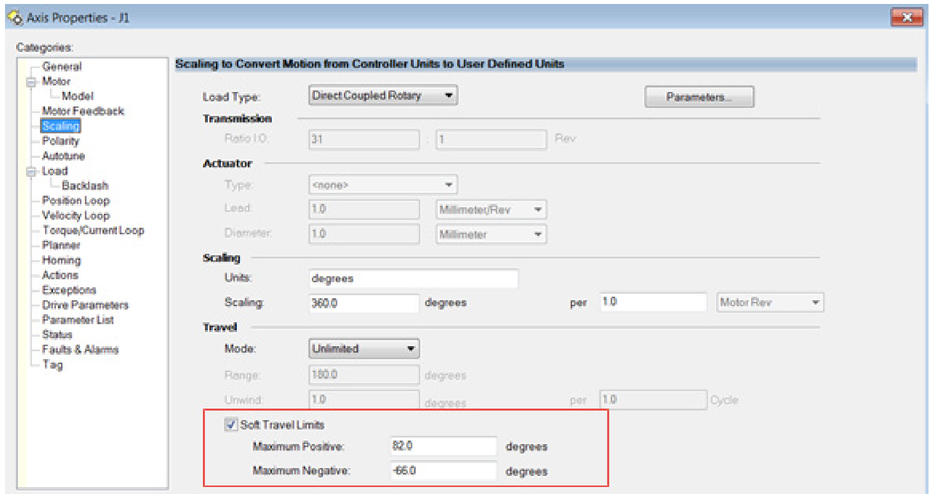
$$\text{JmaxNeg} = -\cos^{-1}\left(\frac{R}{L1 - L2}\right)$$

Maximum J6 joint limit condition

The J6 axis is the rotational axis that could have multiple turns. The maximum number of turns supported is +/-127. Maximum positive and negative range is checked based on number of turns supported on J6.

Configure the joint limits

Refer to robot manufacturer's data sheet to compute the range of J1, J2, and J6 axes. These limits are set as a **Soft Travel Limit** on the **Scaling** tab in the **Axis Properties** dialog box.



See also

Identify the Work Envelope for Delta J1J2J6 robot on page 218

Work and Tool Frame offset limits for Delta J1J2J6 robot

The work envelope for the 3-axis Delta robot relies on the Work and Tool Frame offset values defined in the MCTO instruction. The target end position range changes based on the Work and Tool Frame offsets.

In the Delta robot, the End plate is always parallel to the Base plate and the 3-axis Delta robot can reach only up to limited orientation positions. Work and Tool frame offset values are limited up to reachable work envelope. The following offset values are allowed for Work and Tool frames. The MCTO instruction generates error 148 for invalid offset values.

- Offset values on X, Y, Z and Rz axis are allowed for the Work Frame offsets. Rx and Ry offsets are restricted and must be set to 0°. Specify these offset values through the **WorkFrame** parameter in the MCTO instruction.
- Offset values on X, Y, Z and Rz axis are allowed for the Tool Frame offsets. Rx and Ry offsets are restricted and must be set to 0°. Specify these offset values through the **ToolFrame** parameter in the MCTO instruction.

See also

Identity the Work Envelope on page 218

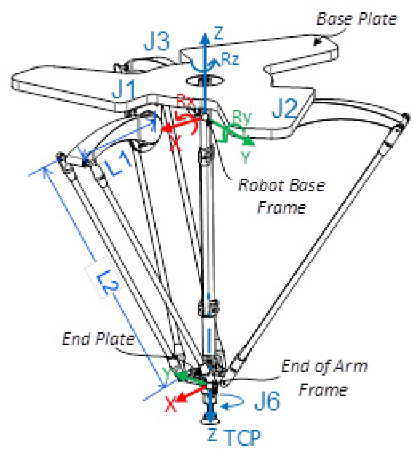
Invalid Cartesian positions

The End of Arm (EOA), using the default work and tool frame settings, can be programmed only in (X, Z, Rz). Note the following:

- If there is a Y component (Translation on Y is not equal to 0), MCTO and MCTPO instructions error with Error code: 153 and Extended Error code: 2.
- If there is any Rx component (Orientation on Rx is not equal to 180°), MCTO and MCTPO instructions error with Error code: 67 and Extended Error code: 1.
- If there is a Ry component (Orientation on Ry is not equal to 0), MCTO and MCTPO instructions error with Error code: 67 and Extended Error code: 2.

Configure a Delta J1J2J3J6 Coordinate System

A four-axis Delta robot that moves in six-dimensional Cartesian (X, Y, Z, Rx, Ry, Rz) space is often called a spider or umbrella robot. This illustration is an example of a four-dimensional Delta robot.



In Logix Designer application, the four-degrees of freedom are configured as four joint axes (J1, J2, J3, and J6) in the robots coordinate system. All joint axes are either:

- Directly programmed in joint space.
- Automatically controlled by the embedded Kinematics software in the application from instructions programmed in a virtual Cartesian coordinate system.

This robot contains a fixed top plate (Base Plate) and a moving bottom plate (End Plate). The fixed top plate is attached to the moving bottom plate by three link-arm assemblies. All three of the link-arm assemblies have a top link arm (L1) and bottom link arm (L2).

As each axis (J1, J2, J3) is rotated, the end plate always moves in XYZ plane parallel to the base plate. The mechanical connections of the Link L2 via spherical joints ensure that the base and end plates remain parallel to each other.

When each top link (L1) moves downward, its corresponding joint axis (J1, J2, or J3) is assumed to be rotating in the positive direction. The three joint axes of the robot are configured as linear axes.

The J6 is connected at the end of the end plate and provides rotation at the end of the arm.

Without a work and tool frame, the End of Arm (EOA) is programmed to a (X, Y, Z, Rz) coordinate. When a tool is attached to the EOA or a different work frame (other than the default) is defined, the Tool Center Point (TCP) can be programmed to a full six axis Cartesian point (X, Y, Z, Rx, Ry, and Rz). The MCTO instruction computes the joint values (J1, J2, J3, and J6) to move the TCP linearly from the current position to the programmed full Cartesian position, using the programmed vector dynamics.

In four-axis Delta robots, the End Plate always remains parallel to Base plate (in XY Plane). As a result, program the Rx, Ry and Rz orientation values with following valid range of values.

Orientation Axis	Valid Ranges
Rx	180°
Ry	0°
Rz	-179.9999° to 180°

See also

Establish the reference frame for Delta J1J2J3J6 Robot on page 224

Calibrate a Delta J1J2J3J6 robot on page 226

Configuration parameters for Delta J1J2J3J6 robot on page 227

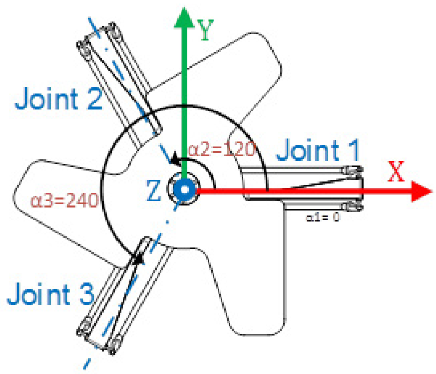
Identify the Work Envelop for Delta J1J2J3J6 robot on page 233

Maximum Joint Limit condition for Delta J1J2J3J6 robot on page 233

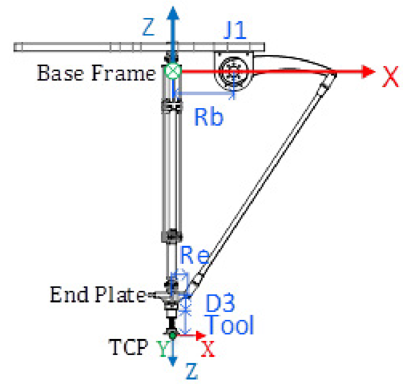
Work and Tool Frame offset limits for Delta J1J2J3J4 robot on page 235

Establish the reference frame for a Delta J1J2J3J6 robot

The reference frame is a Cartesian frame which is the base frame for the robot and all the target points are specified with respect to this base frame. The robot transformations are set up from base frame to end of arm frame to transform any Cartesian target positions in to joint space and vice versa. In order for the transformations to work correctly, it is required to establish the origins for all the axes in the joint space with respect to the robot base Cartesian frame.



Top View



Side View

Establish the Base frame

The reference XYZ frame (Base frame) for the Delta geometry is located near the center of the base plate. Joint 1(J1), Joint 2 (J2), and Joint 3 (J3) are actuated joints and placed 120° apart, shown as α1, α2, and α3.

When configuring a Delta J1J2J3J6 coordinate system in the Logix Designer application, with the joints homed as 0° in the XY plane, then the L1 link is aligned along the X positive axis as shown in the Top View figure. The Side View figure shows that the X axis will pass through the center of J1's motor to the center of Link L1 and L2 joint.

Moving in the counter clockwise direction from J1 to J2 and J3, the Y axis is orthogonal to the X axis. Based on the right hand rule, Z positive axis is the axis pointing up in side view (out of the paper in the top view).

- +J1 rotation is measured clockwise around the -Y axis at the Base frame (+Y axis is pointing inside in Side View).
- As each top link (L1) moves downward, its corresponding joint axis (J1, J2, or J3) is rotating in the positive direction.

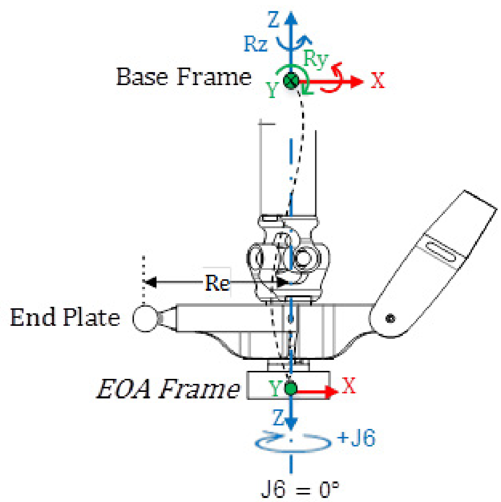
Establish the End of Arm frame

End of Arm (EOA) in XYZ reference frame is set at the end of the End Plate. This frame is rotated by Rx =180° with reference to the Base frame. As a result, the X axis is in the same direction as Base frame X axis but the Z axis direction is pointing down, towards the direction of the Tool approach vector.

J6 axis of rotation is aligned with the Z axis of Base frame.

- To set the home position for J6 axis, move the J6 axis so that the X axis of EOA is aligned with the top link (L1) of the arm (J1), that is, X axis of Base frame.
- +J6 is measured clock wise around the +Z axis at the Base frame.

The following illustration shows the rotation of the axis and its directions for J6 axis.



See also

[Calibrate a Delta J1J2J3J6 robot](#) on [page 226](#)

Calibrate a Delta J1J2J3J6 robot

Use these steps to calibrate a Delta J1J2J3J6 robot.

To calibrate a Delta J1J2J3J6 robot:

1. Obtain the angle values from the robot manufacturer for J1, J2, J3, and J6 at the calibration position. Use these values to establish the reference position.
2. Refer to manufacturer's datasheet to determine if the associated sized motor contains an internal or external gearbox from the motor to actuation at the links or Joints to move the robot.
3. On the **Scaling** tab in the **Axis Properties** dialog box, in the **Transmission Ration I/O** box, set the gear ratio for each axis.
4. In the **Scaling** box, enter the scaling to apply to each axis (J1, J2, and J3) such that one revolution around the Link1 (load rev) equals 360°.

   The same applies to the J6 axis. One revolution of the J6 axis should equal 360°.
5. Move all joints to the calibration position by jogging the robot under programmed control or manually moving the robot when the joint axes are in an open loop state.
6. Do one of the following:

a. Use the Motion Redefine Position (MRP) instruction to set the positions of the joint axes to the calibration values obtained in step 1.

b. Set the configuration value for the joint axes home position to the calibration values obtained in step 1 and execute a Motion Axis Home (MAH) instruction for each joint axis.

7. Move each J1, J2, J3 joint to an absolute position of 0.0. Verify that each joint position reads 0° and the respective L1 is in a horizontal position (XY Plane).
8. If the top link of arm (L1) is not in a horizontal position, configure the values for the **Zero Angle Offsets** on the **Geometry** tab in the **Coordinate System Properties** dialog box to be equal to the values of the joints when in a horizontal position.
9. Move J6 to an absolute position of 0.0. Verify that joint position reads 0 ° and the J6 position is in the Z axis direction of the Base Frame.

Tip: Since the robot axes are absolute, the reference positions may only need establishing once. If the reference positions are lost, for example, the controller changes, then reestablish the reference positions.

See also

Establish the reference frame for a Delta J1J2J3J6 robot on page 224

Configuration parameters for Delta J1J2J3J6 robot

Configure the Logix Designer application to control robots with varying reach and payload capacities. The configuration parameter values for the robot include:

- Link lengths
- Base offsets
- End-effector offsets
- Swing Arm offsets
- Configure Zero Angle Orientation

The configuration parameter information is available from the robot manufacturer.

IMPORTANT Verify that the values for the Link Lengths, Base Offsets, and End-Effector Offsets are entered in the Coordinate System Properties dialog box using the same measurement units.

See also

Link Lengths for Delta J1J2J3J6 robot on page 228

Base and Effector Plate dimensions for Delta J1J2J3J6 robot on page 228

Swing Arm offsets for Delta J1J2J3J6 robot on page 229

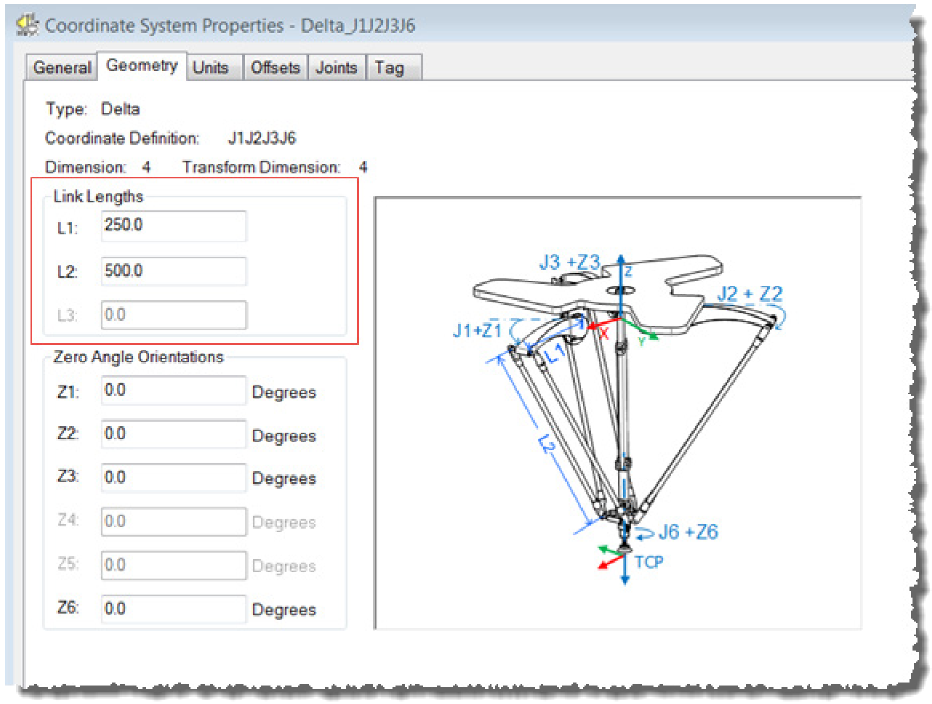
Configure Zero Angle Orientation for Delta J1J2J3J6 robot on page 231

Link Lengths for Delta J1J2J3J6 robot

Link lengths are the rigid mechanical bodies attached at the rotational joints. The four-dimensional Delta robot geometry has three link pairs made up of **L1** and **L2**. Each link pair has the same dimensions.

- **L1** - link attached to each actuated joint (J1, J2, and J3)
- **L2** - link attached to L1 on one end and the end plate at the other end

Enter the link lengths on the **Geometry** tab in the **Coordinate System Properties** dialog box.



See also

[Configuration parameters for Delta J1J2J3J6 robot](#) on [page 227](#)

[Base and Effector Plate dimensions for Delta J1J2J3J6 robot](#) on [page 228](#)

[Swing Arm offsets for Delta J1J2J3J6 robot](#) on [page 229](#)

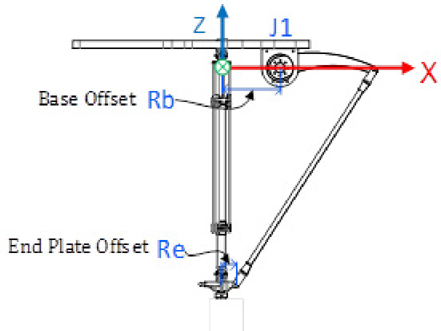
[Configure Zero Angle Orientation for Delta J1J2J3J6 robot](#) on [page 231](#)

Base and Effector Plate dimensions for Delta J1J2J3J6 robot

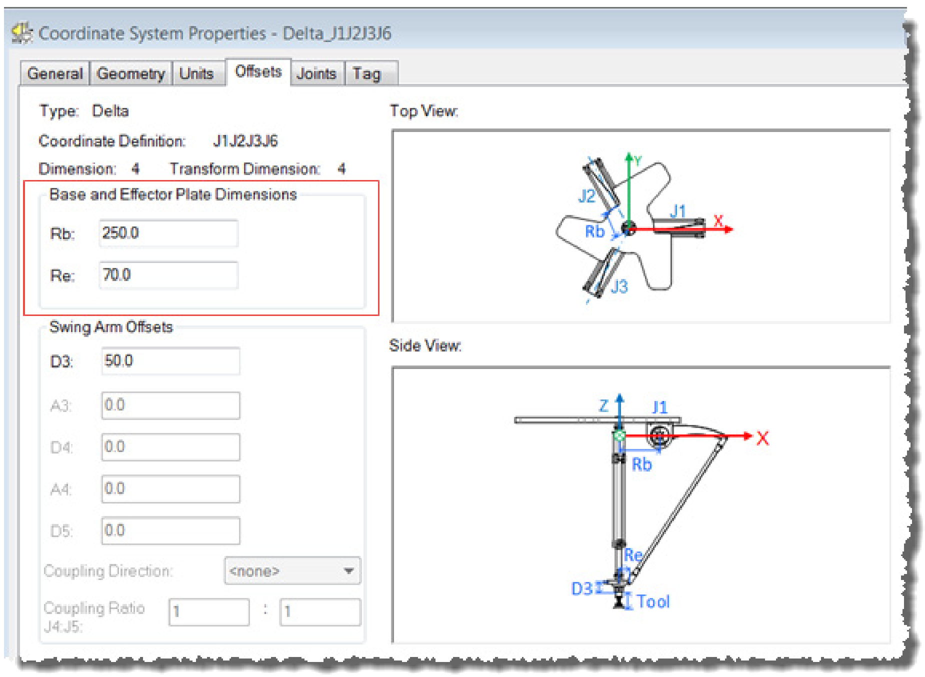
In a 4-axis Delta robot configuration, Base and End plate offsets are represented as **Rb** and **Re** offsets.

- **Rb** - This offset represents the Base plate offset value. Enter the value equal to the distance from the origin of the robot coordinate system to one of the actuator joints.

- **Re** - This offset represents the End plate offset value. Enter the value equal to the distance from the center of the moving end plate to the lower spherical joints of the parallel arms (L2).



In the **Offsets** tab in the **Coordinate System Properties** dialog box, enter the base offset and effector plate offset for the 4-axis Delta robot.



See also

[Configuration parameters for Delta J1J2J3J6 robot](#) on [page 227](#)

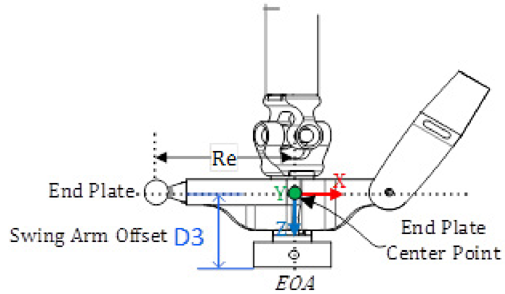
[Swing Arm offsets for Delta J1J2J3J6 robot](#) on [page 229](#)

[Configuring offset variables in a GSV/SSV instruction](#) on [page 231](#)

[Configure Zero Angle Orientations for Delta J1J2J3J6 robot](#) on [page 231](#)

Swing Arm Offsets for Delta J1J2J3J6 robot

In the 4-axis Delta robot configuration, only one Swing Arm Offset (**D3**) is allowed. The **D3** value is the distance on Z axis from the center of end plate to the J6 axis of rotation.

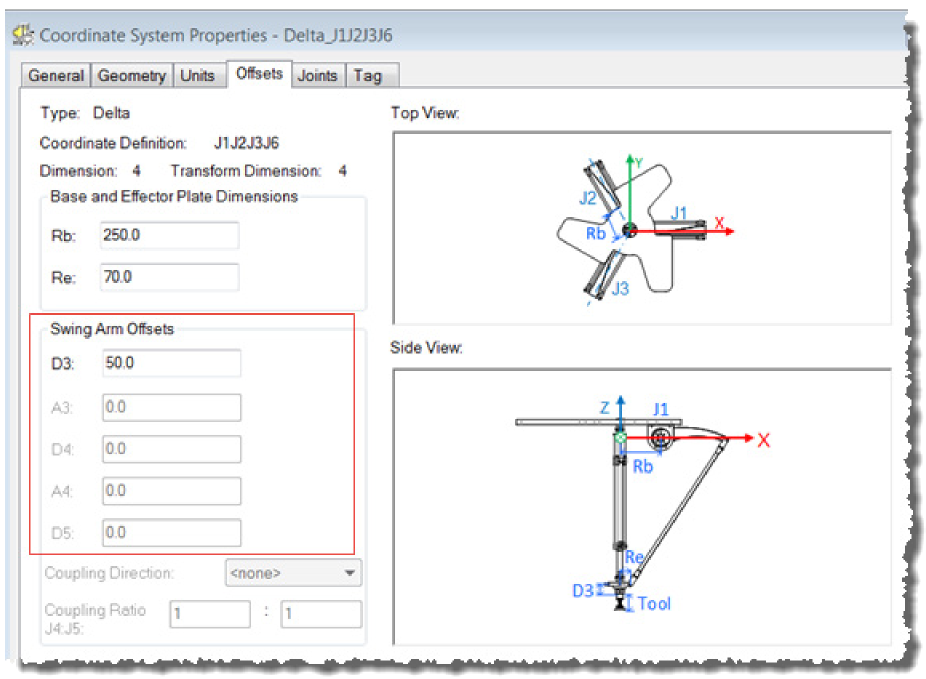


Joint 6 axis is configured using Swing Arm Offset **D3**. Denavit - Hartenberg (DH) notation is used to configure these offset values in which joint offsets in Z direction is shown as **D3**. Offset values can be positive or negative.

Tip: For Swing Arm Offsets, positive Z direction is pointing down at the End plate center point.

Refer to manufacturer's CAD drawings or datasheet to find relevant Swing Arm Offset values for the project.

Enter the Swing Arm Offset values on the **Offsets** tab in the **Coordinate System Properties** dialog box.



See also

Configuration parameters for Delta J1J2J3J6 robot on page 227

Configurable variable to attribute name mapping on page 231

Configure Zero Angle Orientation for Delta J1J2J3J6 robot on page 231

Configuring offset variables in a GSV/SSV instruction

The **Offset** parameters in the **Coordinate System Properties** dialog box for the 4-axis Delta robot are not mapped to the attributes of the same name in the GSV/SSV instruction. Use the table to associate the parameters in the **Coordinate System Properties** dialog box to the attributes in the GSV/SSV instruction.

Parameter in Coordinate System dialog box	Class name	Attribute name	Data type	GSV	SSV
Base Plate dimension: Rb	CoordinateSystem	BaseOffset1	REAL	Yes	Yes
Base Plate dimension: Re	CoordinateSystem	EndEffectorOffset1	REAL	Yes	Yes
Swing Arm Offset: D3	CoordinateSystem	EndEffectorOffset3	REAL	Yes	Yes

See also

Base and Effector Plate dimensions for Delta J1J2J3J6 robot on page 228

Swing Arm Offsets for Delta J1J2J3J6 robot on page 229

Configure Zero Angle Orientations for Delta J1J2J3J6 robot

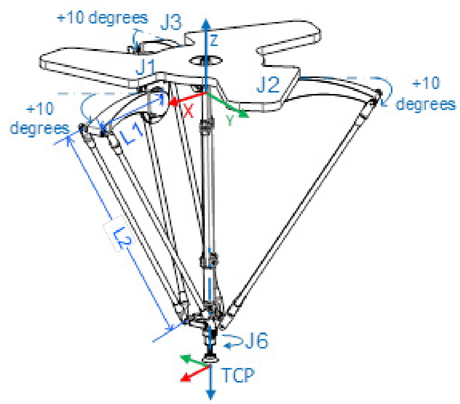
For Delta robot geometries, the internal transformation equations in the Logix Designer application assume:

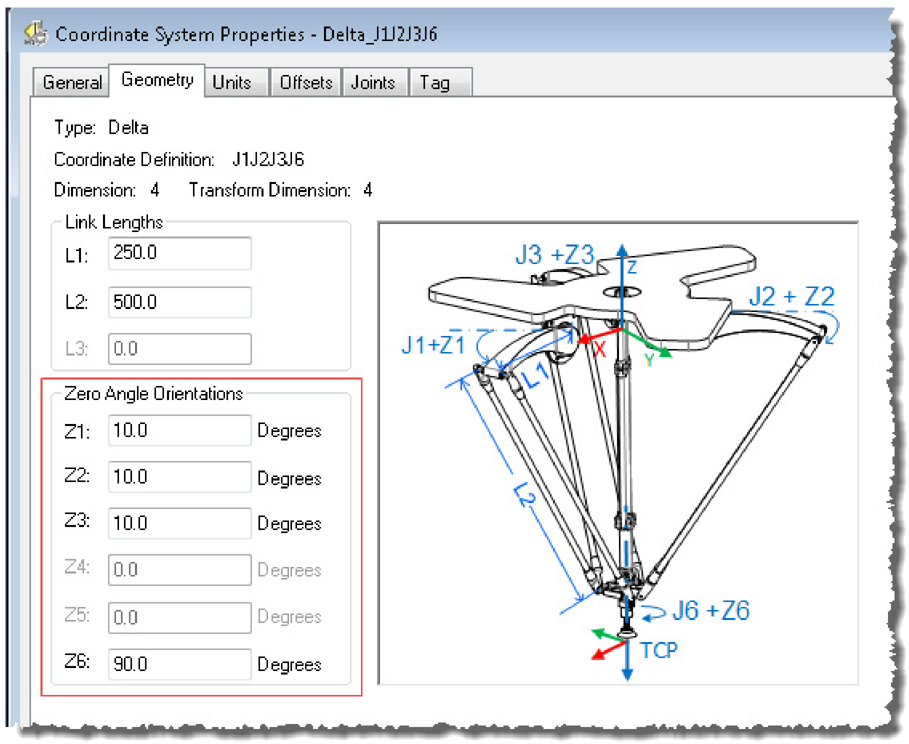
- Joints (J1, J2, and J3) are at 0° when link L1 is horizontal in the XY plane.
- As each top link (L1) moves downward, its corresponding joint axis (J1, J2, or J3) is rotating in the positive direction.
- Joint 6 axis of rotation is aligned with Z axis of the base frame. When J6 is at 0°, End of Arm (EOA) frame is rotated by 180° on Rx (Z axis pointing down) with respect to base frame.

To have joints J1, J2, and J3 angular positions to be any value other than 0° when L1 is horizontal, then configure the **Zero Angle Orientation** values on the **Geometry** tab in the **Coordinate System Properties** dialog box to align the joint angle positions with the internal equations.

For example, if the Delta robot is mounted so that the joints attached at the top plate are homed at 10° in the positive direction below horizontal and you want the robot's coordinate system actual position tag values to be zero in this position, then enter -10° in the **Z1**, **Z2**, and **Z3** parameters on **Geometry** tab. The **Z6** offset is used to set J6 axis other than default 0° position.

Example of Zero Angle Orientation set up in 4-axis Delta robot





See also

Configuration parameters for Delta J1J2J3J6 robot on page 227

Link Lengths for Delta J1J2J3J6 robot on page 228

Base and Effector Plate dimensions for Delta J1J2J3J6 robot on page 228

Swing Arm Offsets for Delta J1J2J3J6 robot on page 229

Identify the work envelope for Delta J1J2J3J6 robot

The work envelope is the three-dimensional region of space that defines the reaching boundaries for the robot arm. The typical work envelope for a Delta robot looks similar to a plane in the upper region, with sides similar to a hexagonal prism and the lower portion similar to a sphere. For more detailed information regarding the work envelope of the four-dimensional Delta robot, refer to the documentation provided by the robot manufacturer.

Program the robot within a rectangular solid defined inside the robot's work zone. The rectangular solid is defined by the positive and negative dimensions of the X, Y, Z virtual source axes. Be sure that the robot position does not go outside the rectangular solid. Check the position in the event task.

To avoid issues with the singularity positions, the Motion Coordinated Transform with Orientation (MCTO) instruction internally calculates the joint limits for the Delta robot geometries. When an MCTO instruction is invoked for the first time, the maximum positive and maximum negative joint limits are internally calculated based upon the Link Lengths and Offset values entered on the **Geometry** and **Offsets** tabs of the **Coordinate System Properties** dialog box.

For more information about the maximum positive and maximum negative joint limits, refer to:

- Maximum Joint Limit Conditions
- Work and Tool Frame Offset Limits

During each scan, the joint positions are checked to ensure that they are within the maximum and minimum joint limits.

Homing or moving a joint axis to a position beyond a computed joint limit and then invoking an MCTO instruction results in an error 67 (Invalid Transform position). For more information regarding error codes, refer to the MCTO instruction in the online help or the Logix 5000 Controllers Motion Instructions Reference Manual, publication MOTION-RM002.

See also

Maximum Joint Limit condition for Delta J1J2J3J6 robot on page 233

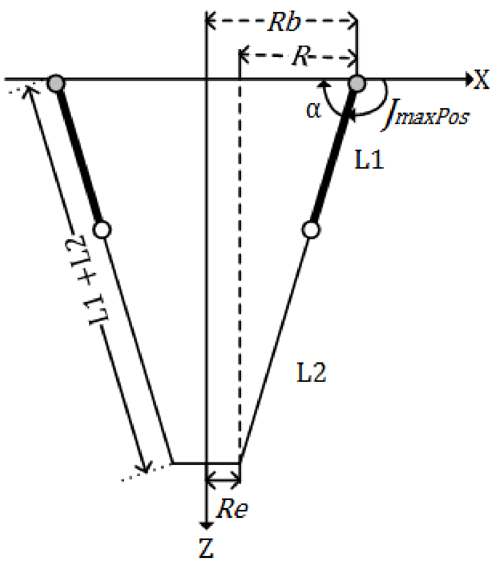
Work and Tool Frame offset limits for Delta J1J2J3J6 robot on page 235

Maximum joint limit condition for Delta J1J2J3J6 robot

Use these guidelines to determine the maximum joint limit conditions for the four-dimensional robot.

Maximum J1, J2, J3 positive joint limit condition

The derivations for the maximum positive joint apply to the condition when L1 and L2 are collinear.



Maximum Positive Joint Limit Position

R = absolute value of (Rb - Re)

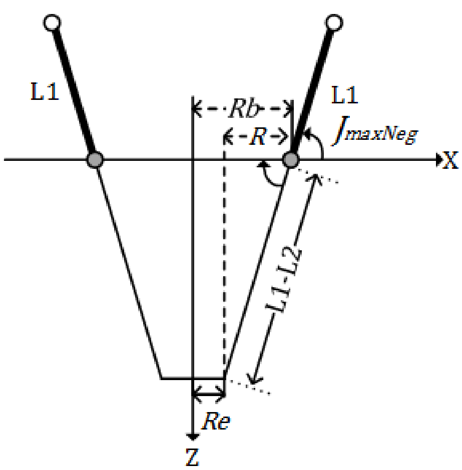
$$\alpha = \cos^{-1}\left(\frac{R}{L1 + L2}\right)$$

JmaxPos = 180 - α

Maximum J1, J2, J3 negative joint limit condition

The derivations for the maximum negative joint limit apply to the condition when L1 and L2 are folded back on top of each other.

R is computed by using the base and end-effector offsets values (Rb and Re).



Maximum Negative Joint Limit Condition

R = absolute value of (Rb - Re)

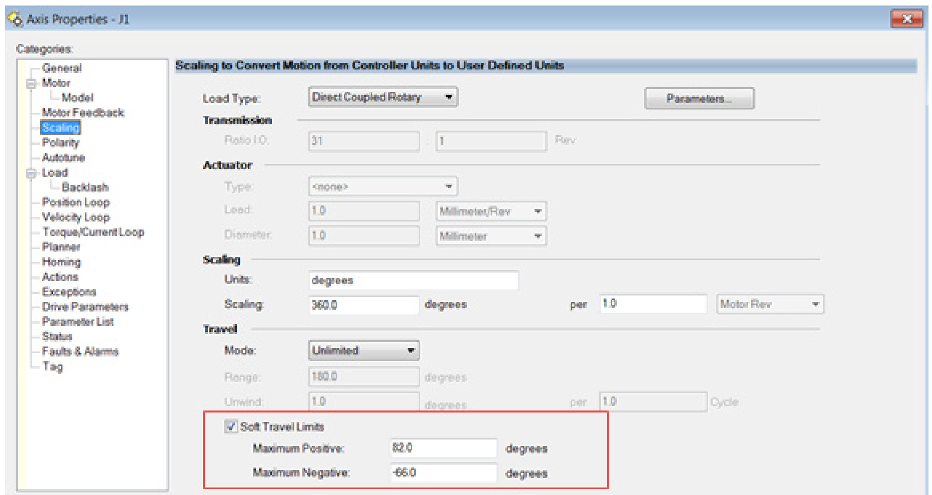
$$JmaxNeg = -\cos^{-1}\left(\frac{R}{L1 - L2}\right)$$

Maximum J6 joint limit condition

The J6 axis is the rotational axis that could have multiple turns. The maximum number of turns supported is +/-127. Maximum positive and negative range is checked based on number of turns supported on J6.

Configure the joint limits

Refer to robot manufacturer's data sheet to compute the range of J1, J2, J3, and J6 axes. These limits are set as a **Soft Travel Limit** on the **Scaling** tab in the **Axis Properties** dialog box.



See also

Identify the Work Envelope for Delta J1J2J3J6 robot on page 233

Work and Tool Frame offset limits for Delta J1J2J3J6 robot

The work envelope for the 4-axis Delta robot relies on the Work and Tool Frame offset values defined in the MCTO and MCTPO instruction. The target end position range changes based on the Work and Tool Frame offsets.

In the Delta robot, the End plate is always parallel to the Base plate and the 4-axis Delta robot can reach only up to limited orientation positions. Work and Tool frame offset values are limited up to reachable work envelope. The following offset values are allowed for Work and Tool frames. The MCTO and MCTPO instructions generates error 148 for invalid offset values.

- Offset values on X, Y, Z and Rz axis are allowed for the Work Frame offsets. Rx and Ry offsets are restricted and must be set to 0°. Specify these offsets through the **WorkFrame** parameter in the MCTO and MCTPO instructions.
- Offset values on X, Y, Z and Rz axis are allowed for the Tool Frame offsets. Rx and Ry offsets are restricted and must be set to 0°. Specify

these offsets through the **ToolFrame** parameter in the MCTO and MCTPO instructions.

See also

Identify the Work Envelope for Delta J1J2J3J6 robot on page 233

Sample project for Delta J1J2J3J6 robot

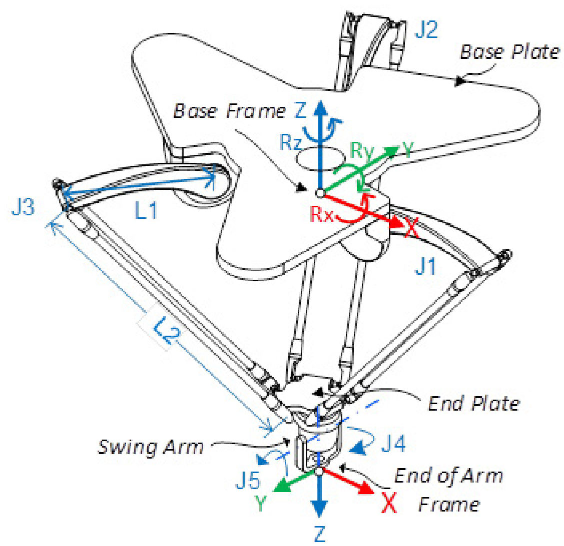
To use the Kinematic sample project on configuring a Delta J1J2J3J6 Delta robot, on the **Help** menu, click **Vendor Sample Projects** and then click the **Motion** category.

The Rockwell Automation sample project's default location is:

c:\Users\Public\Public Documents\Studio 5000\Sample\ENU\v<current_release>\Rockwell Automation

Configure a Delta J1J2J3J4J5 Coordinate System

This illustration shows a five-axis Delta robot that moves in six-dimensional Cartesian (X, Y, Z, Rx, Ry, Rz) space. It is often called a spider or umbrella robot.



In the Logix Designer application, the five-degrees of freedom are configured as five joint axes (J1, J2, J3, J4, J5) in the robots coordinate system. The five joint axes are either:

- Directly programmed in joint space.
- Automatically controlled by the embedded Kinematics software in the application from instructions programmed in a virtual Cartesian coordinate system.

This robot contains a fixed top plate (Base Plate) and a moving bottom plate (End Plate). The fixed top plate is attached to the moving bottom plate by three link-arm assemblies. All three of the link-arm assemblies are identical in that they each have a top link arm (L1) and bottom link arm (L2).

As each axis (J1, J2, J3) is rotated, the end plate moves correspondingly in the (X, Y, Z) direction. The mechanical connections of the parallelograms via spherical joints ensure that the base and end plates remain parallel to each other.

When each top link (L1) moves downward, its corresponding joint axis (J1, J2, or J3) is assumed to be rotating in the positive direction. The three joint axes of the robot are configured as linear axes.

The J4 and J5 axes that form the Swing Arm are connected at the end of the end plate which provides rotation and tilt for the product at the end of the arm.

Some five dimensional Delta robots have a mechanical coupling (gearing) between the Swing Arm rotation and the tilt movement. When the robot moves only the J4 axis, it rotates and tilts the swing arm due to internal gearing. To compensate this tilt effect, the robot needs to move the J5 axis. This relationship is set using **J4:J5 Coupling Ratio** and **Coupling Direction** on the **Offsets** tab in the **Coordinate System Properties** dialog box.

Program the Tool Center Point (TCP) to a (X, Y, Z, Rx, Ry, Rz) coordinate. Then, the application computes the commands necessary for each of the joints (J1,J2,J3,J4,J5) to move the TCP linearly from the current (X, Y, Z, Rx, Ry, Rz) position to the programmed (X, Y, Z, Rx, Ry, Rz) position at the programmed vector dynamics. Directions of Rx, Ry, Rz orientations at the Base frame are shown in above image.

In five-axis Delta robots, the End Plate always remains parallel to Base plate (in XY Plane). As a result, Rx orientation value can only be programed with 0° or 180° values. Ry and Rz orientation values are programed as fixed frame XYZ Euler Angles with their range of +/- 90° and +/-180° respectively.

See also

Establish a reference frame for a Delta J1J2J3J4J5 robot on page 237

Calibrate a Delta J1J2J3J4J5 robot on page 239

Configuration parameters for Delta J1J2J3J4J5 robot on page 240

Identify the Work Envelope for Delta J1J2J3J4J5 robot on page 250

Maximum joint limit condition for Delta J1J2J3J4J5 robot on page 250

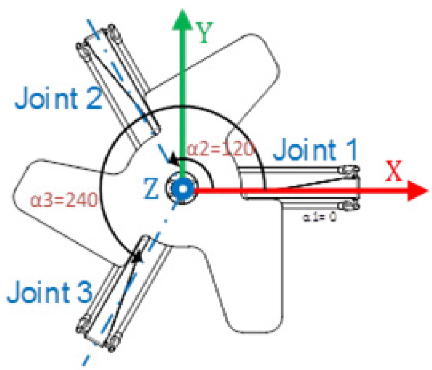
Work and Tool Frame offset for Delta J1J2J3J4J5 robot on page 253

Establish the reference frame for a Delta J1J2J3J4J5 robot

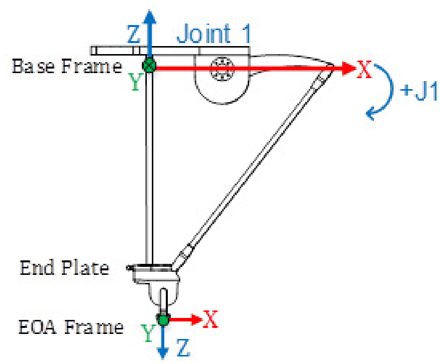
The reference frame is a Cartesian frame which is the base frame for the robot and all the target points are specified with respect to this base frame. The robot transformations are set up from base frame to end of arm frame to transform any Cartesian target positions in to joint space and vice versa. In order for the transformations to work correctly, it is required to establish the origins for all the axes in the joint space with respect to the robot base Cartesian frame.

Establish the Base frame

The reference XYZ frame (Base frame) for the Delta geometry is located near the center of the base plate. Joint 1, Joint 2, and Joint 3 are actuated joints and placed at 120° apart, shown as α1, α2, and α3.



Top view



Side view

Configuring a Delta J1J2J3J4J5 coordinate system in the Logix Designer application with the joints homed as 0° in the XY plane, then L1 of one of the link pairs is aligned along the X positive axis as shown in top view. The side view shows the X axis passing through the center of Joint 1's motor to the center of Link L1 and L2 joint.

Moving in the counter clockwise direction from Joint 1 to Joint 2 and Joint 3, the Y axis is orthogonal to the X axis. Based on the right hand rule, Z positive axis is the axis pointing up in side view (out of the paper in the top view).

- +J1 rotation is measured clockwise around the -Y axis at the Base frame (+Y axis is pointing inside in side view).
- As each top link (L1) moves downward, its corresponding joint axis (J1, J2, or J3) is rotating in the positive direction.

Establish the End of Arm frame

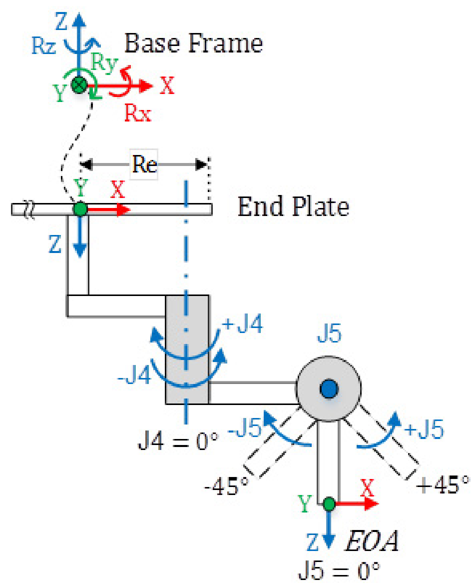
Joint 4 and Joint 5 are the swing arm axes used for rotation and tilt movement of the Swing Arm. End of Arm (EOA) XYZ reference frame is set at the end of the Swing Arm. The EOA frame direction is rotated by Rx =180° with Base frame. At the EOA, X axis is in the same direction as Base frame X axis and the Z axis direction is pointing down towards the direction of Tool approach vector.

Joint 4 axis of rotation is aligned with the Z axis of Base frame and Joint 5 axis of rotation is aligned with Y axis of Base Frame.

- To set the home position for J4 axis, move the J4 and J5 axis such a way that X axis of EOA is aligned with link L1 of the J1 axis (X axis of Base frame).
- Homing of J5 axis is set with reference to J4 position. When J4 axis is homed to 0° position, J5 rotation is aligned with the Y axis of Base frame. At J5 home position, swing arm link (D5) should be vertical aligned with X axis of Base frame.

  The following illustration show axis of rotations and their directions for J4 and J5.

  Tip: In case of coupling to prevent tilt motion caused by J4 homing, first home the J4 to 0° then home J5 to 0° with reference to the J4 home position.



- + J4 is measured clock wise around the +Z axis at the Base Frame.
- + J5 is measured counterclockwise around the -Y axis at the Base Frame (+Y axis is pointing inside) when J4 is homed at 0° position.

See also

Calibrate a Delta J1J2J3J4J5 robot on page 239

Calibrate a Delta J1J2J3J4J5 robot

Use these steps to calibrate a five-dimensional robot.

To calibrate a Delta J1J2J3J4J5 robot:

1. Obtain the angle values from the robot manufacturer for J1, J2, J3, J4, and J5 at the calibration position. These values are used to establish the reference position.

2. Refer to manufacturer's datasheet to determine if the associated sized motor contains an internal or external gearbox from the motor to actuation at the links or Joints to move the robot.
3. On the **Scaling** tab in the **Axis Properties** dialog box, in the **Transmission Ration I/O** box, set the gear ratio for each axis.
4. In the **Scaling** box, enter the scaling to apply to each axis (J1, J2, and J3) such that one revolution around the Link1 (load rev) equals 360 degrees.

   The same applies to the J4 and J5 axis. One revolution of the J4 or J5 axis should equal 360 degrees.

5. Move all joints to the calibration position by jogging the robot under programmed control or manually moving the robot when the joint axes are in an open loop state.
6. Do one of the following:

a. Use the Motion Redefine Position (MRP) instruction to set the positions of the joint axes to the calibration values obtained in step 1.

   b. Set the configuration value for the joint axes home position to the calibration values obtained in step 1 and execute a Motion Axis Home (MAH) instruction for each joint axis.

7. Move each J1, J2, J3 joint to an absolute position of 0.0. Verify that each joint position reads 0 degrees and the respective L1 is in a horizontal position (XY Plane).

   If L1 is not in a horizontal position, configure the values for the **Zero Angle Offsets** on the **Geometry** tab in the **Coordinate System Properties** dialog box to be equal to the values of the joints when in a horizontal position.

8. Move each J4, J5 joint to an absolute position of 0.0. Verify that each joint position reads 0 degrees and the respective J4 and J5 positions are in the Z axis and Y axis direction of the Base Frame.

Tip: Since the robot axes are absolute, the reference positions may only need establishing once. If the reference positions are lost, for example, the controller changes, then reestablish the reference positions.

See also

Establish the reference frame for Delta J1J2J3J4J5 robot on page 237

Configuration parameters for Delta J1J2J3J4J5 robot

Configure the Logix Designer application to control robots with varying reach and payload capacities. The configuration parameter values for the robot include:

- Link lengths
- Base offsets
- End-effector offsets

- Swing Arm offsets
- Coupling Ratio

The configuration parameter information is available from the robot manufacturer.

IMPORTANT	Verify that the values for the Link Lengths, Base Offsets, and End-Effector Offsets are entered in the Coordinate System Properties dialog box using the same measurement units.

See also

Link Lengths for Delta J1J2J3J4J5 robot on page 241

Base and Effector Plate dimensions for Delta J1J2J3J4J5 robot on page 242

Swing Arm Offsets for Delta J1J2J3J4J5 robot on page 243

Coupling between J4 and J5 axis on page 246

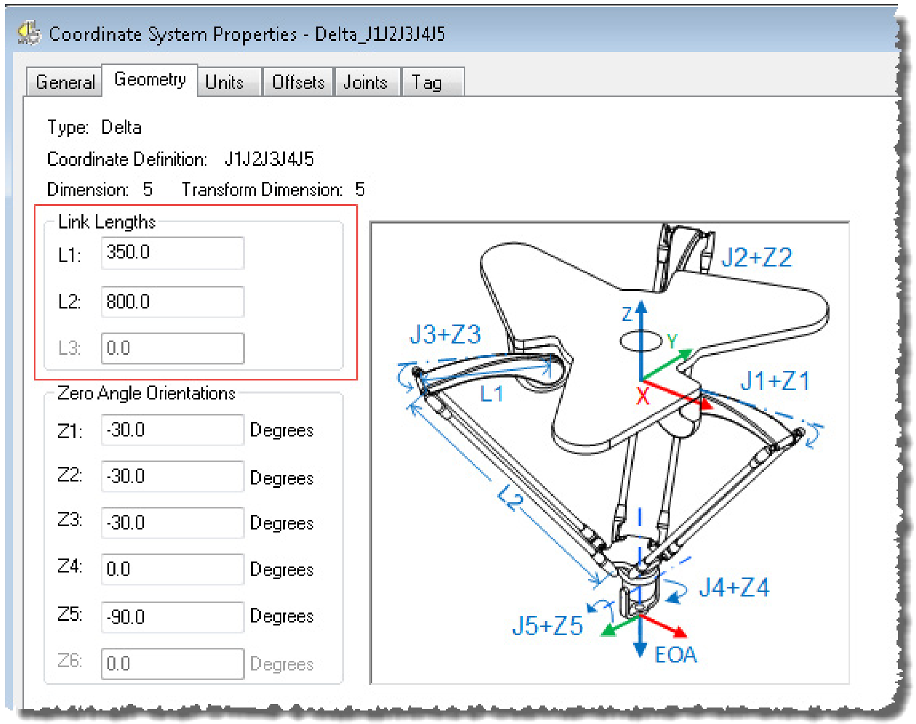
Configure Zero Angle Orientation for Delta J1J2J3J4J5 robot on page 248

Link Lengths for Delta J1J2J3J4J5 robot

Link lengths are the rigid mechanical bodies attached at the rotational joints. The five-dimensional Delta robot geometry has three link pairs made up of **L1** and **L2**. Each link pair has the same dimensions.

- **L1** - link attached to each actuated joint (J1, J2, and J3)
- **L2** - the parallel bar assembly attached to L1

Enter the link lengths on the **Geometry** tab in the **Coordinate System Properties** dialog box.



See also

[Configuration parameters for Delta J1J2J3J4J5 robot](#) on [page 240](#)

[Base and Effector Plate dimensions for Delta J1J2J3J4J5 robot](#) on [page 242](#)

[Swing Arm Offsets for Delta J1J2J3J4J5 robot](#) on [page 243](#)

[Coupling between J4 and J5 axis](#) on [page 246](#)

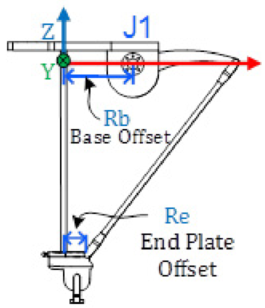
[Configure Zero Angle Orientations for Delta J1J2J3J4J5 robot](#) on [page 248](#)

Base and Effector Plate dimensions for Delta J1J2J3J4J5 robot

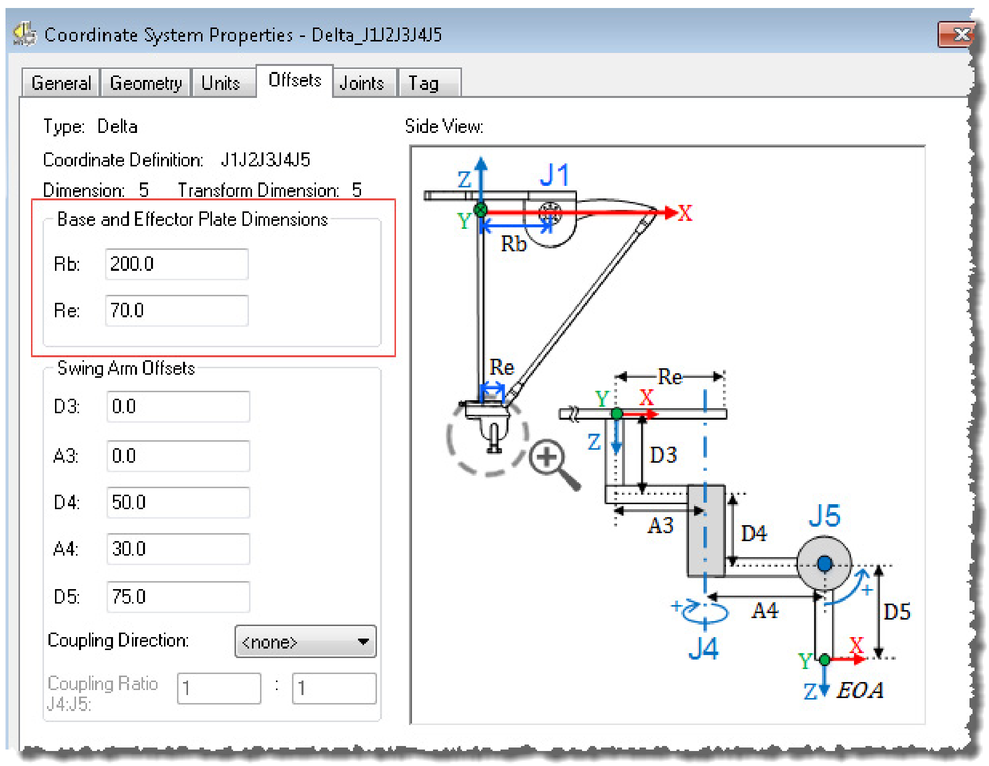
In a 5-axis Delta robot configuration, Base and End plate offsets are represented as Rb and Re offsets.

- **Rb** - This offset represents the Base plate offset value. Enter the value equal to the distance from the origin of the robot coordinate system to one of the actuator joints.

- **Re** - This offset represents the End plate offset value. Enter the value equal to the distance from the center of the moving end plate to the lower spherical joints of the parallel arms (L2).



In the **Offsets** tab in the **Coordinate System Properties** dialog box, enter the base offset and effector plate offset for the 5-axis Delta robot.



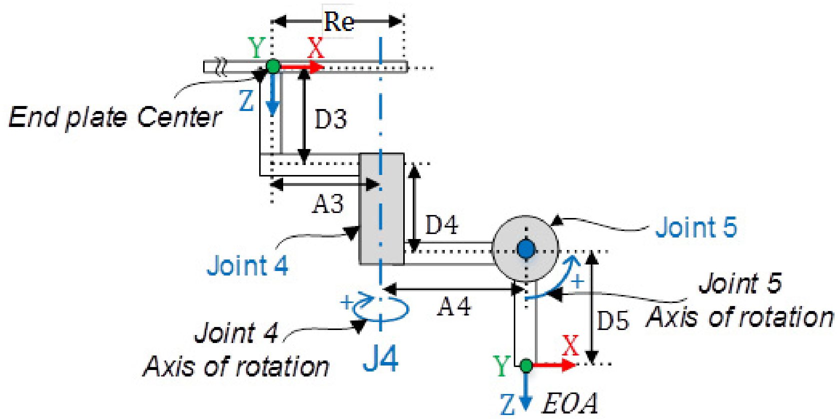
See also

[Configuration parameters for Delta J1J2J3J4J5 robot](#) on [page 240](#)

[Swing Arms offsets for Delta J1J2J3J4J5 robot](#) on [page 243](#)

Swing Arm Offsets for Delta J1J2J3J4J5 robot

In the 5-axis Delta robot configuration, the Joint 4 and Joint 5 axis are configured using Swing Arm offsets **A3**, **D3**, **A4**, **D4**, and **D5**. Denavit - Hartenberg (DH) notation is used to configure these offset values. Use XYZ axis directions, shown in the image at end plate center point, as a reference frame to measure the offset values. As per DH convention, Joint offsets in X direction are represented as **A3** and **A4**, and Joint offsets in Z direction are shown as **D3**, **D4**, and **D5**. Offset values are positive or negative based on XYZ reference frames shown in the illustration.



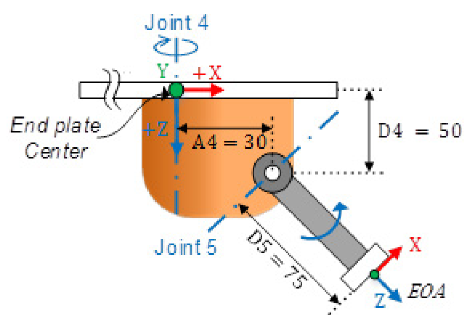
- **D3** - The distance on Z axis from the center of end plate to the J4 axis of rotation.
- **A3** - The distance on X axis from center of end plate to the J4 axis of rotation.
- **D4** - The distance on Z axis from the J4 axis of rotation to the J5 axis of rotation.
- **A4** - The distance on X axis from the J4 axis of rotation to the J5 axis of rotation.
- **D5** - The distance on Z axis from the J5 axis of rotation to the EOA frame.

Tip: For all Swing Arm offsets, positive Z direction is pointing down at the End plate center point.

Refer to the manufacturer's CAD drawings or datasheet to find relevant Swing Arm Offset values for the project. Some offset values will be zero based on the mechanical setup. These examples shows how to configure Swing Arm offsets with two different mechanical setups.

Example 1

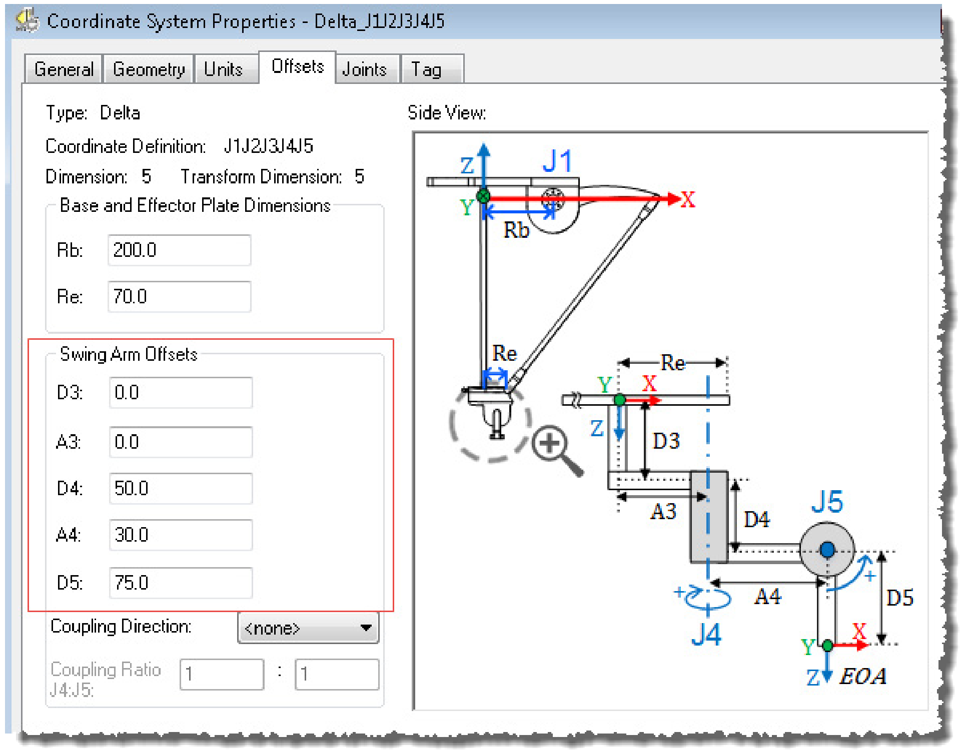
The image shows one of the typical setups for a Swing Arm mechanism. Here Joint 4 and Joint 5 axes are not intersecting each other. Joint 4 axis is passing through the End plate center point.



The table shows configuring offsets and Swing Arm Offset values:

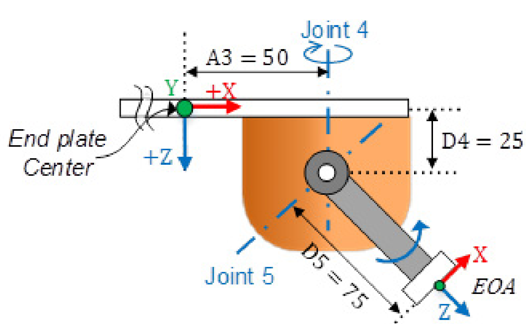
Configuring offsets	**Swing Arm offset value**
Joint 4 axis is starting right at the End plate center point so **A3** and **D3** offsets are zero.	D3 = 0 A3 = 0
Joint 5 is at a distance from Joint 4. Distance on the positive X axis is configured as **A4** = 30mm, distance on positive Z axis is measured as **D4** = 50mm.	D4 = 50 A4 = 30
From Joint 5 to EOA is measured as **D5** = 75 mm.	D5 = 75

Enter these offset values on the **Offsets** tab in the **Coordinate System Properties** dialog box.



Example 2

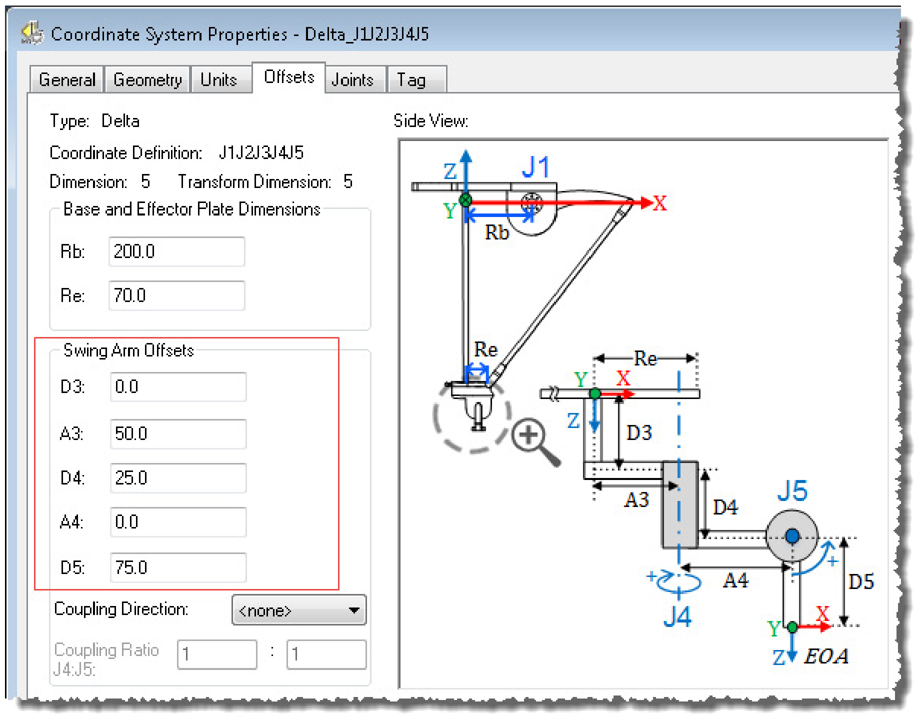
In this example, Joint 4 axis of rotation is at a distance from End plate center point. Joint 4 and Joint 5 axis are intersecting each other.



The table to shows configuring offsets and Swing Arm Offset values:

Configuring offsets	Swing Arm offset value
Joint 4 axis is at a distance from End plate center point. Offset distance in X positive direction is measured as **A3**= 50 mm and in Z positive direction is as measured as **D4** = 25mm. (In this setup, D3 can also be used in place of D4).	A3 = 50 D4 = 25
Joint 4 and Joint 5 are intersecting each other so **D3** and **A4** offset values are zero.	D3 = 0 A4 = 0
From Joint 5 to EOA is measured as **D5** = 75 mm.	D5 = 75

Enter these offset values on the **Offsets** tab in the **Coordinate System Properties** dialog box.



See also

Coupling between J4 J5 axis on page 246

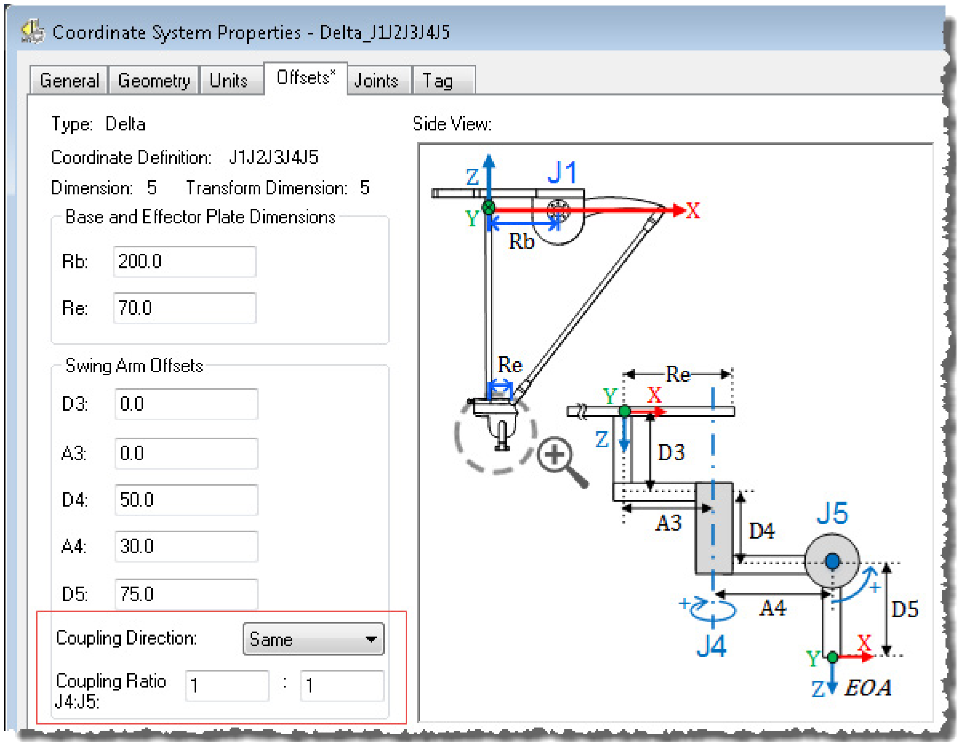
Configuration parameters for Delta J1J2J3J4J5 robot on page 240

Configure Zero Angle Orientations for Delta J1J2J3J4J5 robot on page 248

Coupling between J4 and J5 axis

Some five dimensional Delta robots have a mechanical coupling between the J4 and J5 axis. Rotation of the Swing Arm causes the tilt movement on D5 offset link. To compensate for this tilt motion, move the J5 axis in the same or opposite direction of the J4 axis move with relative gear ratio.

Configure the gear ratio as **Coupling Ratio J4:J5** and gear direction as **Coupling Direction** on the **Offsets** tab in the **Coordinate System Properties** dialog box.



Refer to manufacturer's manual for coupling relationship between J4 and J5 axis.

Tip: The Coupling attributes apply only to the Delta J1J2J3J4J5 robot.

Coupling Direction

This parameter indicates the direction of the coupling between J4 and J5. There are 3 options to choose from:

- **<none>** - No coupling relation between J4 and J5.
- **Same** - Coupling between J4 and J5 is in same direction, that is, J4 positive rotation causes the tilt motion in the same direction of the positive J5 motion.
- **Opposite** - Coupling between J4 and J5 is in opposite direction, that is, J4 positive rotation causes the tilt motion in the opposite direction of the positive J5 motion.

Coupling Ratio J4:J5

The parameter is only available when **Coupling Direction** is set to **Same** or **Opposite**. It includes a Swing Arm Coupling Ratio Numerator and a Swing Arm Coupling Ratio Denominator.

$$Coupling\ Ratio = \frac{\text{Joint 4}}{\text{Joint 5}} = \frac{Swing\ Arm\ Coupling\ Ratio\ Numerator}{Swing\ Arm\ Coupling\ Ratio\ Denominator}$$

The **Numerator** is the first value of the Coupling Ratio parameter. It represents J4 axis rotation as a reference for J5 axis move.

The **Denominator** is the second value of the Coupling Ratio parameter. It represents J5 axis rotation caused by J4.

For example, if the J4 axis is moving by 10 degrees (or revs) and causes the 5 degrees (or revs) of tilt movement, then the coupling ratio between J4:J5 should be set as 2:1.

Tip: Both rotations should be measured in same units (degree or rev.) The **Numerator** and **Denominator** default to 1 and cannot be set to zero.

See also

Configuration parameters for Delta J1J2J3J4J5 robot on page 240

Configure Zero Angle Orientations for Delta J1J2J3J4J5 robot

For Delta robot geometries, the internal transformation equations in the Logix Designer application assume:

- Joints (J1, J2, and J3) are at 0° when link L1 is horizontal in the XY plane.
- As each top link (L1) moves downward, its corresponding joint axis (J1, J2, or J3) is rotating in the positive direction.
- Joint 4 axis of rotation is aligned with Z axis and Joint 5 axis or rotation is aligned with Y axis of the base frame. When J4 and J5 is at 0°, End of Arm (EOA) frame is rotated by 180° on Rx (Z axis pointing down) with respect to base frame.

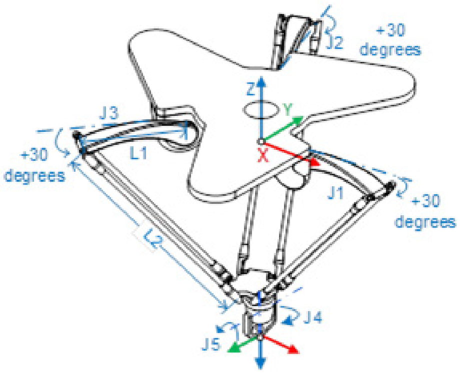
To have joints J1, J2, and J3 angular positions to be any value other than 0° when L1 is horizontal, then configure the **Zero Angle Orientation** values on the **Geometry** tab in the **Coordinate System Properties** dialog box to align the joint angle positions with the internal equations.

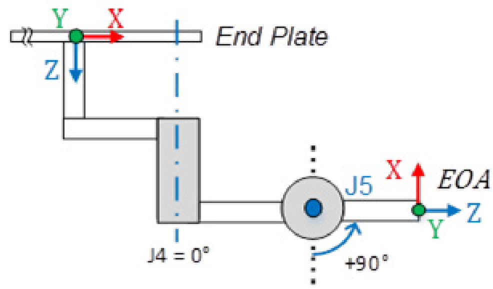
For example, if the Delta robot is mounted so that the joints attached at the top plate are homed at 30° in the positive direction below horizontal and you want the readout value in the application to be zero in this position, then enter -30° in the **Z1**, **Z2**, and **Z3** parameters on **Geometry** tab.

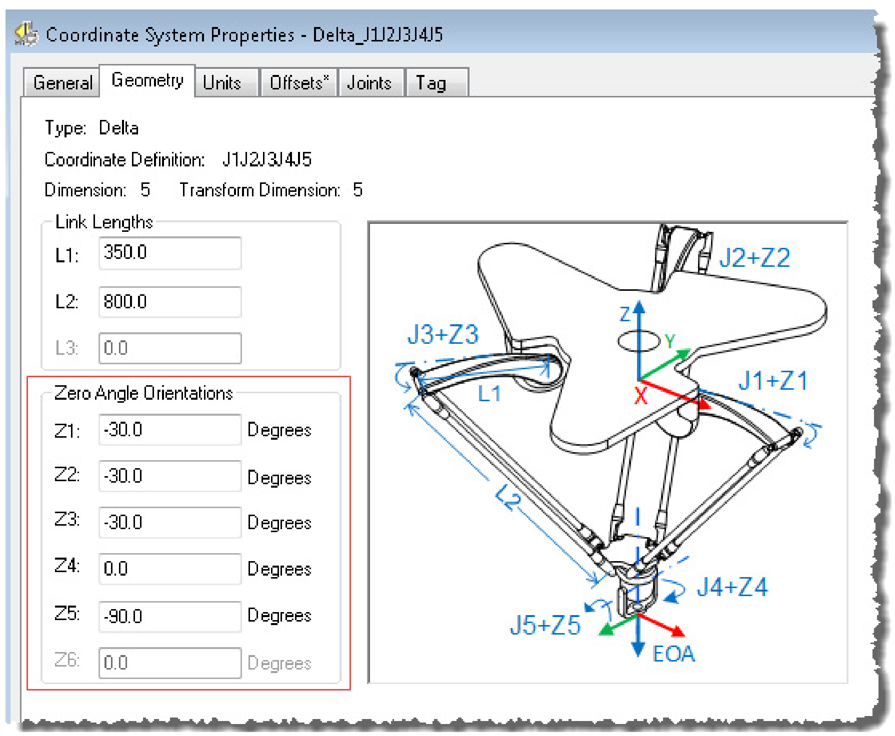
If you want the Joint 5 axis position set as a 0° when D5 link is at horizontal position (shown in the image below), then enter -90° in the **Z5** parameter for

Joint 5. The **Z4** offset can be used to set Joint 4 axis other than default 0° position.

Example of Zero Angle Orientation set up in 5-axis Delta robot







See also

[Configuration parameters for Delta J1J2J3J4J5 robot](#) on [page 240](#)

[Link Lengths for Delta J1J2J3J4J5 robot](#) on [page 241](#)

[Base and Effector Plate dimensions for Delta J1J2J3J4J5 robot](#) on [page 242](#)

[Swing Arm Offsets for Delta J1J2J3J4J5 robot](#) on [page 243](#)

Identify the work envelope for Delta J1J2J3J4J5 robot

The work envelope is the three-dimensional region of space that defines the reaching boundaries for the robot arm. The typical work envelope for a Delta robot looks similar to a plane in the upper region, with sides similar to a hexagonal prism and the lower portion similar to a sphere. For more detailed information regarding the work envelope of the five-dimensional Delta robot, refer to the documentation provided by the robot manufacturer.

Program the robot within a rectangular solid defined inside the robot's work zone. The rectangular solid is defined by the positive and negative dimensions of the X, Y, Z virtual source axes. Be sure that the robot position does not go outside the rectangular solid. Check the position in the event task triggered by the execution of the Motion Group task.

To avoid issues with the singularity positions, the Motion Coordinated Transform with Orientation (MCTO) instruction internally calculates the joint limits for the Delta robot geometries. When an MCTO instruction is invoked for the first time, the maximum positive and maximum negative joint limits are internally calculated based upon the Link Lengths and Offset values entered on the **Geometry** and **Offsets** tabs of the **Coordinate System Properties** dialog box.

For more information about the maximum positive and maximum negative joint limits, refer to:

- Maximum Joint Limit Conditions
- Work and Tool Frame Offset Limits.

During each scan, the joint positions are checked to ensure that they are within the maximum and minimum joint limits.

Homing or moving a joint axis to a position beyond a computed joint limit and then invoking an MCTO instruction results in an error 67 (Invalid Transform position). For more information regarding error codes, refer to the MCTO instruction in the online help or the Logix 5000 Controllers Motion Instructions Reference Manual, publication MOTION-RM002.

See also

Maximum joint limit condition for Delta J1J2J3J4J5 robot on page 250

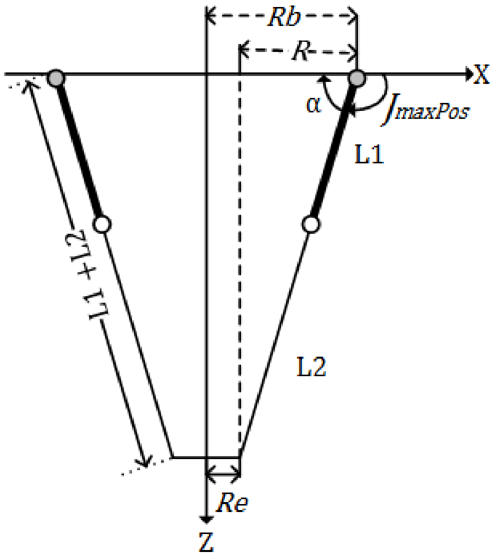
Work and Tool Frame offset limits for Delta J1J2J3J4J5 robot on page 253

Maximum joint limit condition for Delta J1J2J3J4J5 robot

Use these guidelines to determine the maximum joint limit conditions for the five-dimensional robot.

Maximum J1, J2, J3 Positive joint limit condition

The derivations for the maximum positive joint apply to the condition when L1 and L2 are collinear.



Maximum Positive Joint Limit Position

R = absolute value of (Rb - Re)

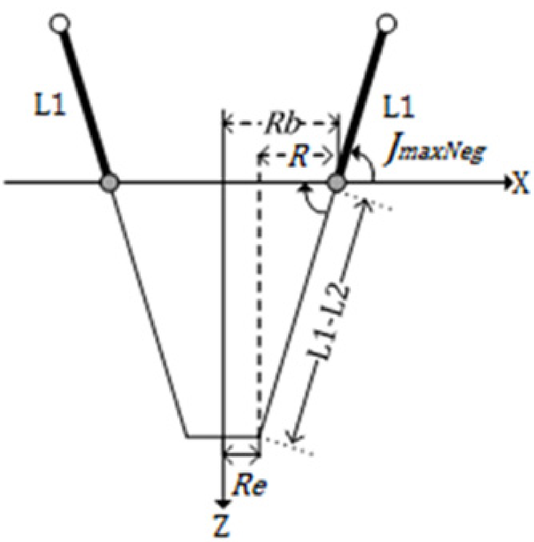
$$\alpha = \cos^{-1}\left(\frac{R}{L1 + L2}\right)$$

JmaxPos = 180 - α

Maximum J1, J2, J3 negative joint limit condition

The derivations for the maximum negative joint limit apply to the condition when L1 and L2 are folded back on top of each other.

R is computed by using the base and end-effector offsets values (Rb and Re).



Maximum Negative Joint Limit Condition

R = absolute value of (Rb - Re)

$$\text{JmaxNeg} = -\cos^{-1}\left(\frac{R}{L1 - L2}\right)$$

Maximum J4 joint limit condition

J4 axis is the rotational axis that could have multiple turns. The maximum number of turns supported is +/-127. Maximum positive and negative range is checked based on number of turns supported on J4.

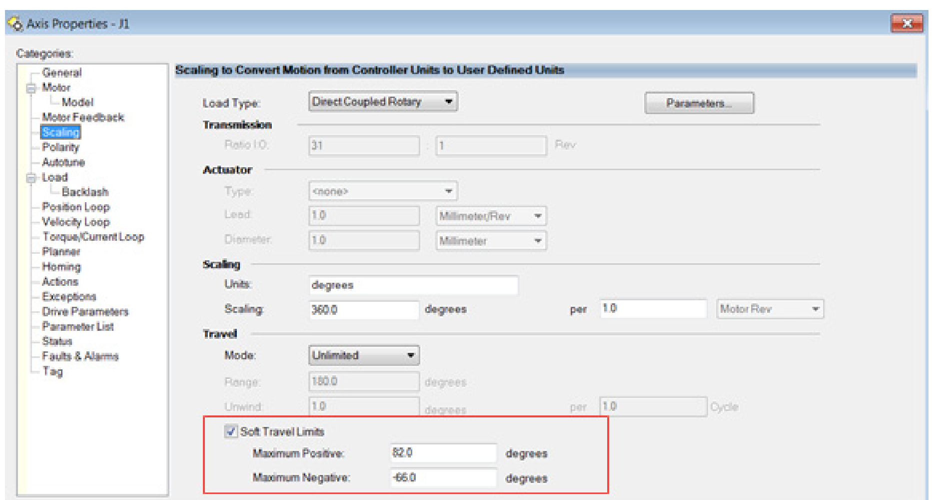
Maximum J5 joint limit condition

The maximum positive and negative limit of J5 axis is restricted between -179° to +179° to avoid singularity conditions. Actually tilt motion of the Swing Arm is restricted with -/+179° range.

In case of mechanical coupling, the maximum limit of J5 axis is computed based on J4 axis limit. J5 axis can move beyond this -/+ 179° range but the effective Swing Arm tilt motion is restricted between +/- 179°. For example, if J4:J5 coupling ratio is 2:1 and J4 range is -/+720°, then J5 can move up to -/+360° to compensate for coupling effect.

Configure the joint limits

Refer to robot manufacturer's data sheet to compute the range of J1, J2, J3, J4, and J5 axes. These limits are set as a **Soft Travel Limit** on the **Scaling** tab in the **Axis Properties** dialog box.



See also

[Identify the Work Envelope for Delta J1J2J3J4J5 robot](#) on [page 250](#)

Work and Tool Frame offset limits for Delta J1J2J3J4J5 robot

The work envelope for the 5-axis Delta robot relies on the Work and Tool Frame offset values defined in the MCTO instruction. The target end position range changes based on the Work and Tool Frame offsets.

In the Delta robot, the End plate is always parallel to the Base plate and the 5-axis Delta robot can reach up to limited orientation positions. Work and Tool frame offset values are limited up to reachable work envelope. The following offset values are allowed for Work and Tool frames. The MCTO instruction generates error 148 for invalid offset values.

- Offset values on X, Y, Z and Rz axis are allowed for the Work Frame offsets. Rx and Ry offsets are restricted and must be set to 0°. Specify these offsets through the **WorkFrame** parameter in the MCTO instruction.
- Offset values on X, Y, Z and Ry axis are allowed for the Tool Frame offsets. Rx and Rz offsets are restricted and must be set to 0°. Specify these offsets through the **ToolFrame** parameter in the MCTO instruction.

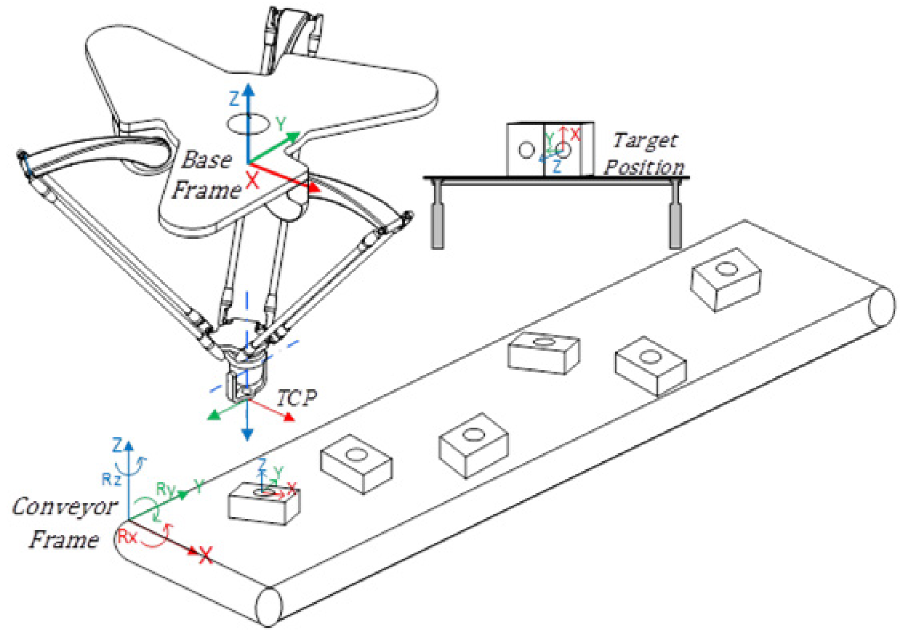
See also

Identify the Work Envelope for Delta J1J2J3J4J5 robot on page 250

Example of a Pick and Place application for Delta J1J2J3J4J5 robot

The following image is an example of a typical pick and place application with the Delta robot. It illustrates how the 5-axis Delta robot picks up the boxes from the conveyor and places them on the table with different orientations on Ry and Rz axis, assuming that all target positions are reachable for the 5-axis Delta robot.

Conveyor coordinate system frame is used as a reference frame for this application. Positions of all boxes on the conveyor are measured using this reference frame.



Work Frame offsets set the distance from the robot's base frame to the conveyor reference frame. For example, if the XYZ offsets between robot base frame to conveyor reference frame is (-200, -100, and -1000) and the orientation offset on Rz is -30°, then set the work frame offset as [-200,-100,-1000, 0, 0,-30] in the Motion Coordinated Transform with Orientation (MCTO) instruction.

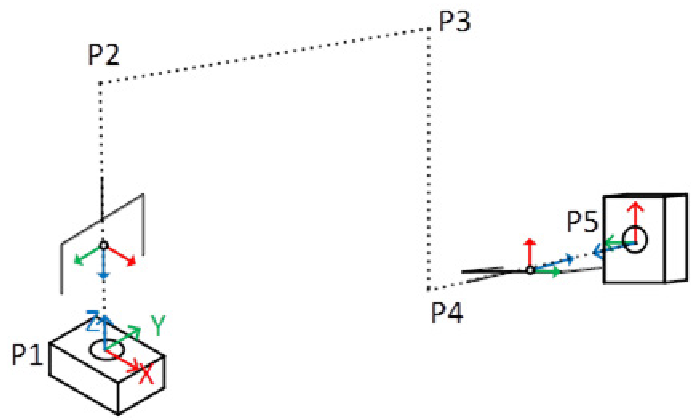
Configure the robot by entering the Link lengths, Base and Effector plate dimensions, and Swing Arm offsets in the **Coordinate System Properties** dialog box.

The following image shows Pick & Place path details from the conveyor to the table. The object is picked from point P1 and moved on the Z axis to P2. During the horizontal move from point P2 to P3, Ry and Rz orientation positions are changed and it will maintain that orientation during P4 and P5 move.

- Positions of different boxes from the conveyor frame are used as a target position in the Motion Coordinated Path Move (MCPM) instruction. For example, first box's XYZ position form the conveyor is (200, 200, 50) and it is rotated by 30° on Rz axis so P1 position is programmed as [200, 200, 50, 180, 0, 30] in MCPM instruction.
- During point P2 to P3 move, Rz value at TCP changes from 30° to 90° and Ry value changes from 0° to -90.
- Boxes are placed on a table with different Rx, Ry and Rz orientations. For example, first box's XYZ position form the conveyor is (400, 500, 100) and it is rotated by -90 on Ry and Rz axis so P5 position is programmed as [400, 500, 100, 0, -90, -90] in the MCPM instruction.

Tip: Here Rx, Ry and Rz orientation positions are measured using fixed frame XYZ Euler angle notation, where Ry range is +/- 90 and it will rollover. Rx and Rz values will flip at Ry rollover positions.

- This cycle is repeated for other boxes coming on the Conveyor with different XYZ positions and Rz orientations.

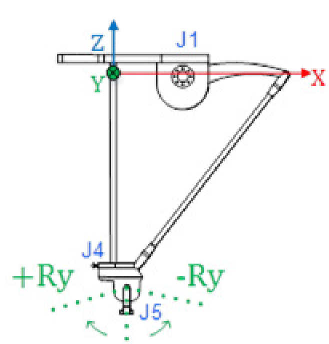


Different target positions for Pick and Place application

Position	X	Y	Z	Rx	Ry	Rz
P1	200	200	50	180	0	30
P2	200	200	200	180	0	30
P3	400	400	200	0	-90	-90
P4	400	400	100	0	-90	-90
P5	400	500	100	0	-90	90

MCPM mirror image orientation axis behavior

Many robot geometries supported in ControlLogix integrated kinematics transformations do not have enough degrees of freedom to support orientation motion in the Ry axis, to include SCARA J1J2J3J6 and the Delta J1J2J3J6. Some robot geometries, like the Delta J1J2J3J4J5, do support orientation moves in the Ry axis. Systems like these allow for programmed motion on the Ry axis position, which exhibits a mirror image orientation behavior. This introduces some notable changes in how orientation moves of such systems are specified.



Tips:
- Mirror image behavior occurs only when Motion Coordinated Transform with Orientation (MCTO) transforms are active.
- The mirror image position data assumes no Tool or Work frame orientation offsets are applied.
- Ry orientation on the Delta J1J2J3J4J5 has opposite sign of J5 joint position. See Configuring the Delta J1J2J3J4J5 Coordinate System for more details.

Important: Avoid using the Motion Axis Move (MAM) instruction with orientation axes to prevent unintended motion on the machine. It does not take into account the Euler angle rollover specifications or the Ry mirror orientation effect when planning motion on these axes.

See also

Mirror image Ry orientation on page 256

Example of mirror image and flip behavior on Rx and Rz axes on page 258

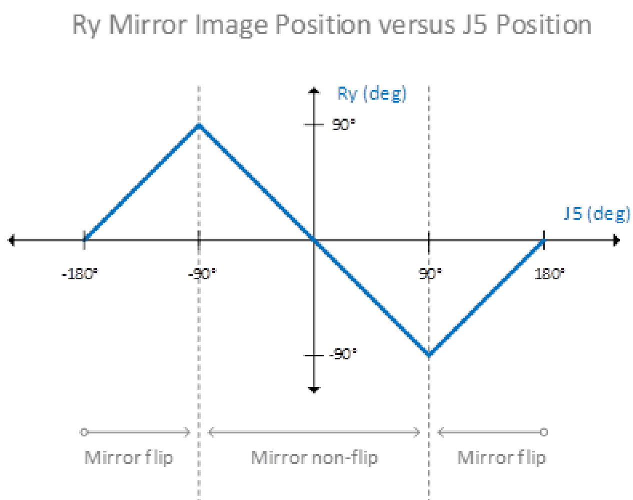
Mirror orientation restrictions on page 259

Use MCPM to program Ry absolute moves for geometries with mirror image position on page 259

Configure a Delta J1J2J3J4J5 coordinate system on page 223

Mirror image Ry orientation

Ry is limited to +/- 90° per Euler angle rules. Refer to Orientation Specification for information about XYZ Fixed angles and Euler Angles Representation. Mirror image refers to the way the Ry position trend looks with respect to +/- 90°.



When the J5 axis position is in the range of -90.0° > J5 > +90.0°, the Ry axis position correlates inversely to J5 axis position. This range of operation is referred to as the mirror non-flip region, and is similar in behavior to the Rz/J4 transform position relationship.

When the J5 axis crosses the ninety degree boundary, the Ry axis position no longer tracks the inverse of J5. Instead the Ry position reflects a positive correlation with J5. This range of operation is referred to as mirror flip region.

See also

[Rx axis position in mirror non-flip and flip regions on page 257](#)

Rz axis position in mirror non-flip and flip regions on page 258

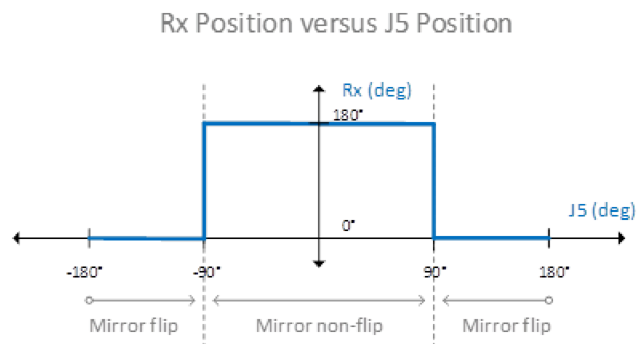
Orientation specification on page 171

Rx axis position in mirror non-flip and mirror flip regions

For certain geometries, such as the Delta J1J2J3J4J5, there is no direct control over the Rx axis. Instead, the value of Rx can be one of two discrete values depending on the J5/Ry position:

Region	Rx position
Mirror non-flip	180.0°
Mirror flip	0.0°

This is shown graphically as follows.



Important: Per Euler angle convention, -180.0° is equal to +180.0° and is also a valid Rx position in the mirror non-flip region. However, due to limitations imposed to support J4 turns counter, this value is not permitted for use in specifying Rx position.

See also

Rz axis position in the mirror non-flip and mirror flip regions on page 258

Mirror image Ry orientation on page 256

Rz axis position in mirror non-flip and mirror flip regions

Robot geometries that exhibit the mirror image Ry position behavior have an impact on the Rz position depending on which region the Ry axis is operating. This relationship is shown in the following table.

Region	J4 range	Rz position
mirror non-flip	-180° <= J4 < 180°	-(J4)
mirror flip	0 <= J4 < 180°	-(J4) + 180.0°
mirror flip	-180° <= J4 < 0	-(J4) - 180.0°

Tip: The Rz flip in position does not result in any motion on the J4 axis.

See also

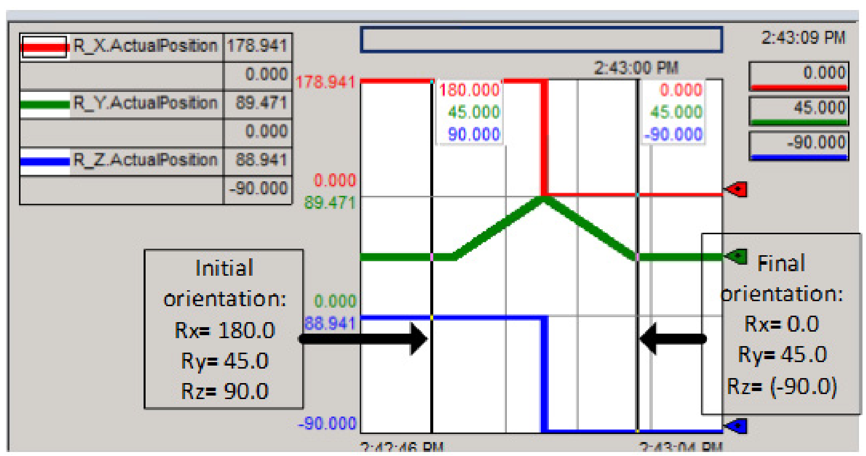
Mirror image Ry orientation on page 256

Rx axis position in mirror non-flip and flip regions on page 257

Example of mirror image and flip behavior on Rx and Rz axes on page 258

Example of mirror image and flip behavior on Rx and Rz axes

The following trend shows the Ry mirror image orientation and associated flip behavior on Rx and Rz axes.



The move that is demonstrated in the example is a pure Ry move from 45.0° in the mirror non-flip region (Rx = 180.0°) in a positive direction ending at 45.0° in the rollover region (Rx = 0°).

Tips:

- The flip of Rx and Rz values as Ry crosses the mirror boundary at 90°.
- No motion is commanded on Rx or Rz, only Ry.

Tip: To use these Kinematic sample projects, on the **Help** menu, click **Vendor Sample Projects** and then click the **Motion** category. The Rockwell Automation sample project's default location is: c:\Users\Public\Public Documents\Studio 5000\Sample\ENU\v<current_release>\Rockwell Automation

Mirror orientation restrictions

The following orientation angle specifications are not allowed in Logix Designer application due to singularity conditions involving multiple solutions or other scenarios involving Euler angle specification:

- The orientation [Rx = 180.0°, Ry = 90.0°] is mathematically correct but is not allowed in Logix Designer application due to ambiguity with the [Rx = 0.0°, Ry = 90.0°] specification. **Always use Rx = 0.0° when specifying Ry = 90.0°**.
- An absolute orientation move starting at [Rx = 180.0,° Ry = 0.0°] and ending at [Rx = 0.0°, Ry = 0.0°] is not allowed. See example 6 in Use MCPM to program Ry absolute moves for geometries with mirror image position.
- Shortest rotary path moves for Ry is not allowed when both start and end orientation lies in the mirror flip region. See example 6 in Use MCPM to program Ry absolute moves for geometries with mirror image position.

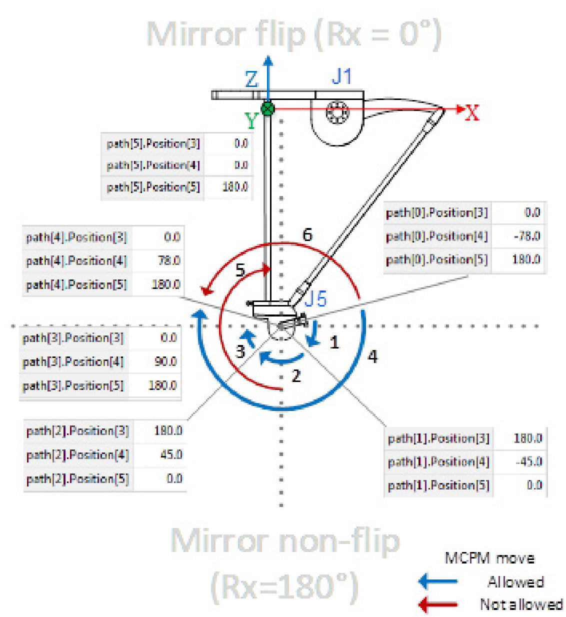
See also

Use MCPM to program Ry absolute moves for geometries with mirror image position on page 259

Use MCPM to program Ry absolute moves for geometries with mirror image position

Below is the side view of the Delta J1J2J3J4J5 arm. It illustrates Ry moves using the absolute position to specify the end of the move.

The blue arrows [1-4] indicate absolute moves that are allowed. The red arrows [5-6] indicate absolute moves that are not allowed.



The following examples are limited to absolute moves since incremental moves for orientation axes with mirror image are not impacted like absolute moves. The absolute orientation for starting and end positions are specified using the notation [Rx, Ry, Rz]. Also, the examples limit actual motion to the J5 axis (due to Ry) to demonstrate the mirror image effect on Rx and Rz without generating actual changes in orientation in those dimensions.

Example	Start Region	End Region	Notes
1	Mirror flip	Mirror non-flip	Starting orientation [Rx=0, Ry=(-78), Rz=180], with Motion Coordinated Path Move (MCPM) move to orientation [Rx=180, Ry=(-45), Rz=0]. The resultant move is +57° on Ry (-57° on J5), and Rx flips from 0° to 180° and Rz flips from 180° to 0° when Ry crosses the minus -90° boundary.
2	Mirror non-flip	Mirror non-flip	Starting orientation [Rx=180, Ry=(-45), Rz=0], with MCPM move to orientation [Rx=180, Ry=45, Rz=0]. The resultant move is +90° on Ry (-90° for J5). No boundary is crossed and thus no flip in value for Rx or Rz.
3	Mirror non-flip	Mirror flip	Starting orientation [Rx=180, Ry=45, Rz=0], with MCPM move to orientation [Rx=0, Ry=90, Rz=180]. The resultant move is +45° on Ry (-45° on J5). The positive 90° boundary cross causes a flip on Rx and Rz. See Mirror orientation restrictions for more on specifying Ry = 90 orientation.
4	Mirror flip	Mirror flip	Starting orientation [Rx=0, Ry=(-78), Rz=180] with MCPM move to orientation [Rx=0, Ry=78, Rz=180]. The resultant move takes the longest rotary path move to avoid travel through 0° in the Mirror flip region, or +204° on Ry (-204° for J5). Shortest rotary path move for Ry is not allowed in the Mirror flip region.
5	--	---	This is a very specific case involving a move from home position [Rx=180, Ry=0] to absolute position [Rx=0, Ry=0]. This move is not allowed due to ambiguity of the direction of travel (either positive or negative direction would be correct, yet indeterminate from the absolute orientation specified). **Tip:** An incremental Ry move of distance 180° is allowed here - the direction of the move is explicitly specified by the sign of the distance parameter.
6	Mirror flip	Mirror flip	Shortest rotary path move for Ry is not allowed in the Mirror flip region. Example 4 shows how such a move is planned. **Tip:** Incremental moves are not limited like absolute moves are. However, such incremental Ry moves will encounter transformation error when attempting to cross zero degrees (J5 = +/- 180°) in the Mirror flip region.

See also

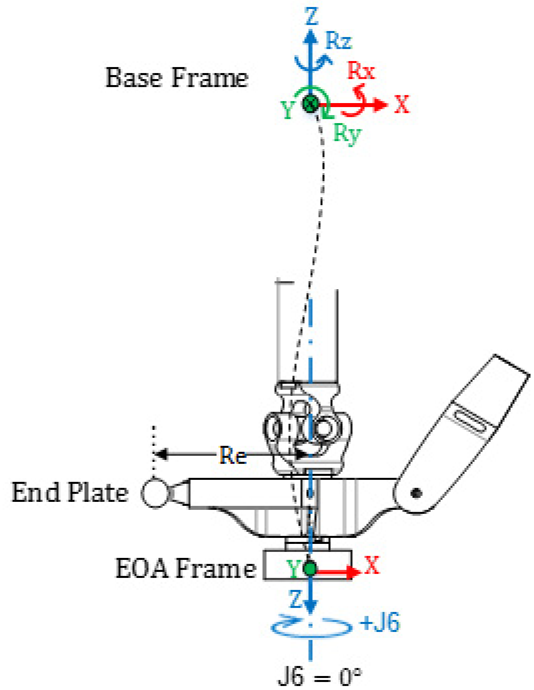
Mirror orientation restrictions on page 259

Configure and program turns counters

Use the MCTO instruction to establish a bidirectional transform between Cartesian and robot system with coordinates that are joint axes of a robot.

The Cartesian system coordinates are defined by XYZ translation coordinates and RxRyRz orientation coordinates in the fixed angle convention.

The robots have geometrical configurations where typically the joint axes are not orthogonal. The geometrical configurations are specified by coordinate system type, such as Delta. The coordinate definition attribute further specifies how many joint axes in the Robot coordinate system, such as J1,J2,J3,J6. This diagram shows the details of a Delta J1J2J3J6 robot with the base Cartesian coordinate system and four joint axes, which form the non-Cartesian coordinate system.



Cartesian and joint target points for Delta J1J2J3J6 robot system

A point in space may be described in two different ways; as a set of Cartesian coordinates (Euclidean space) and as a set of robot joint angles (joint space).

Since there is no rotation on Rx and Ry Orientation axis, only program the Rx orientation value to 180°. The Ry orientation is always 0°, and program the Rz orientation values within fixed XYZ Euler Angle range of Rz, that is, within +/-180°.

Joint axes for J1, J2 and J3 are typically configured as linear axis with over-travel limits. The J6 joint axis is also typically configured as a linear axis with over-travel limits.

Tips:
- For transformations to work correctly, be sure to establish the reference frame for the joint coordinate system first. For the Delta J1J2J3J6 and Delta J1J2J3J4J5 robots, the normal reference positions for J1, J2 and J3 axes are homed to 0° when the J1, J2 and J3 links are horizontal. The J6 axis is homed to 0° when it is parallel to J1 link.
- The J6 rotation is opposite to Rz rotation with respect to the robot base frame.

Once the robot reference frame is established, move the robot to a position in joint space, if needed, before enabling the MCTO instruction. After enabling the MCTO instruction, a bidirectional transform link is established so that, if the Cartesian coordinate is commanded to move to Cartesian coordinate target, the robot moves to Cartesian target coordinates along a linear path. Similarly, if the robot joint coordinate system is commanded to move to joint coordinate target, the robot moves to target joint coordinates along a non-Cartesian path. As the MCTO instruction is enabled, the system maintains the coordinate system related data (that is Cartesian position) for Cartesian and robot coordinate systems.

Turns counter

As shown in the previous diagram, positive orientation rotation for Rz is counterclockwise around the Z axis of the robot base frame. However, the positive rotation for J6 axis is clockwise around the Z axis of the robot base frame which is opposite to Rz axis rotation.

With the 3D Delta robot system since there is no rotation possible around X and Y axis of base frame, the only rotation possible is around Z axis. As a result, the Cartesian coordinate system can be described with the following translation and orientation specifications:
X, Y, Z: [-inf,+inf]
Rx: [180.0]
Ry: [0.0]
Rz: [-179.999, +180.0]

The Rz target position is the rotation around base Z axis and so any rotation can be specified with a range of +/- 180° with one exception of -180°. As 180° and -180° is the same point, the system does not allow specification of -180° as Rz target point.

However, this specification will not be complete as J6 axis can rotate more than one turn. The system handles this functionality by adding an additional turns counter specification for each target point specification.

Co-relating Rz axis with J6 axis and turns counter

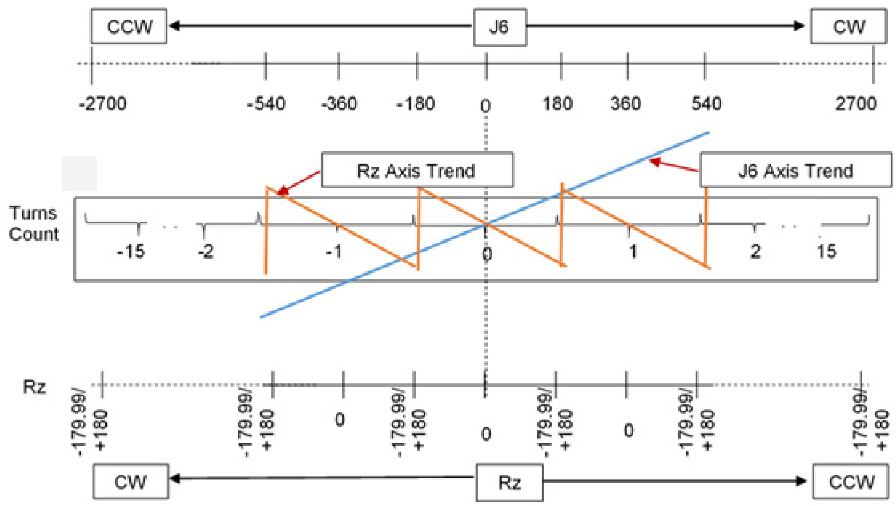
This diagram explains how Rz and turns counter varies with J6 (assuming that the work frame offset, the tool frame offset and the zero angle offset on J6 are 0). J6 is a linear axis and for example can have total travel of 15 revolutions with for example a range from -7.5*360 = -2700 to +7.5*360 = +2700°. As a result, physically the J6 can have multiple turns and have an attribute of turns counter which keeps track of number of the turns associated with the current position of J6 axis. When J6 crosses the 180° point in the CW direction, turns counter is incremented and Rz flips from -180° to 180° and when J6 goes past the 180° point in the CCW direction, turns counter is decremented and Rz flips from 180.0001° to -179.9999°.

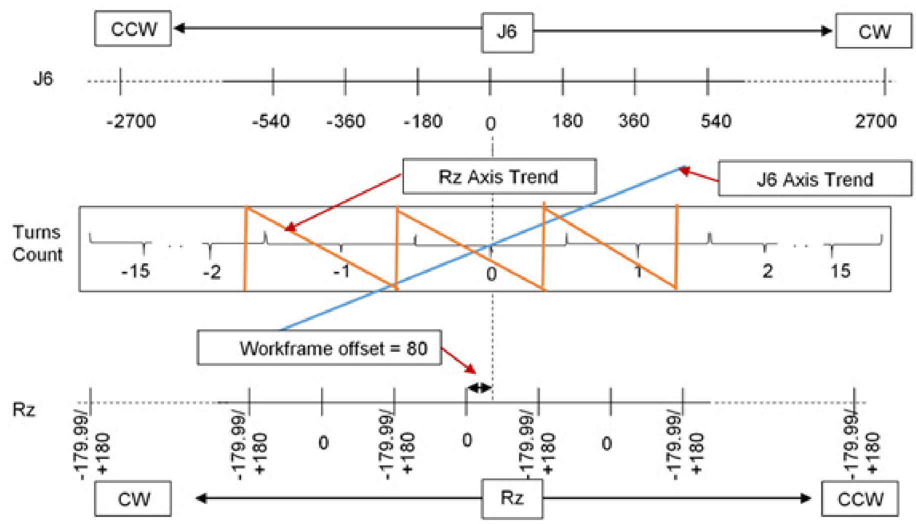
The range of turns counter is limited to +/-127 but the actual max number of turns is geometry dependent. The 3 Turns Counters are elements of a single array attribute of the target coordinate system which contain J1, J4 or J6 axes turns counters.

Tips:

- If Rz reaches the point 180° but does not cross it, it does not flip and stays at 180°. If Rz reaches the point -180°, it flips to +180°.
- If either the work frame or the tool frame offset on Rz is not 0, turns counters still increment when J6 crosses the 180° point, but Rz is flipped when J6 crosses the (180° + offset on Rz) point. In other words flip is shifted by offset on Rz as shown. See below for details.

Rz, J6 axis position and turns counter trends and tables





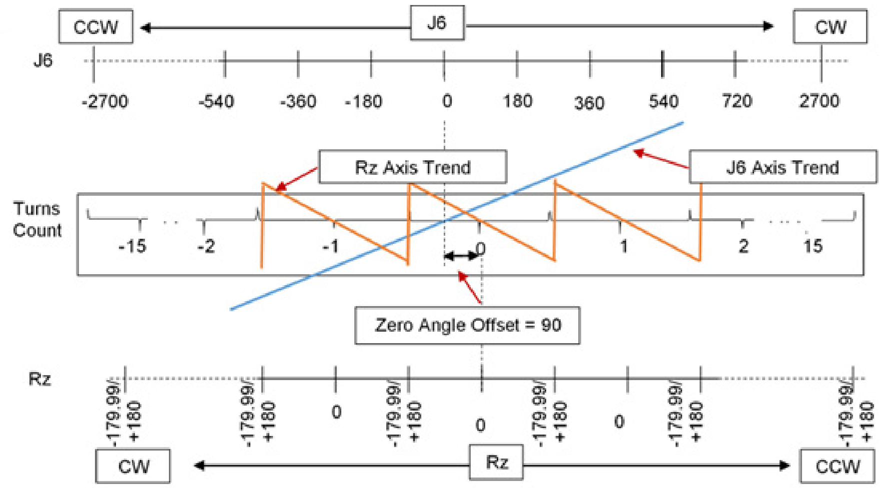


Table of Rz, turns counter and J6 values that are shown in the trends in figures above.

Rz	Turns Counter of J6	J6 (if zero angle offset = 0°) and (Rz work Offset = 0°)	J6 (if zero angle offset = 0°) and (Rz work offset = 80°)	J6 (if zero angle offset = 90°) and (work Offset = 0°)
+179.9999	2	540.0001	460.0001	630.0001
+180	2	540	460	630
-179.9999	1	539.9999	459.9999	629.9999
---	---	---		---
0	1	360	280	450
---	---	---		---
+179.9999	1	180.0001	100.0001	270.0001
+180	1	180	100	270
-179.9999	0	179.9999	99.9999	269.9999
---	---	---		---
0	0	0	-80	90
---	---	---		---
+179.9999	0	-179.9999	-259.9999	-89.9999
+180	0	-180	-260	-90
-179.9999	-1	-180.0001	-260.0001	-90.0001

See also

Program example for turns counter on page 264

Program example for turns counter

The following is an example for programming a turns counter.

Configure Cartesian and robot coordinate systems

Refer to configuring Cartesian and robot coordinate systems for details of configuring the two coordinate systems that are used for the turns counter application example. The example uses the Delta J1J2J3J4J5 robot system.

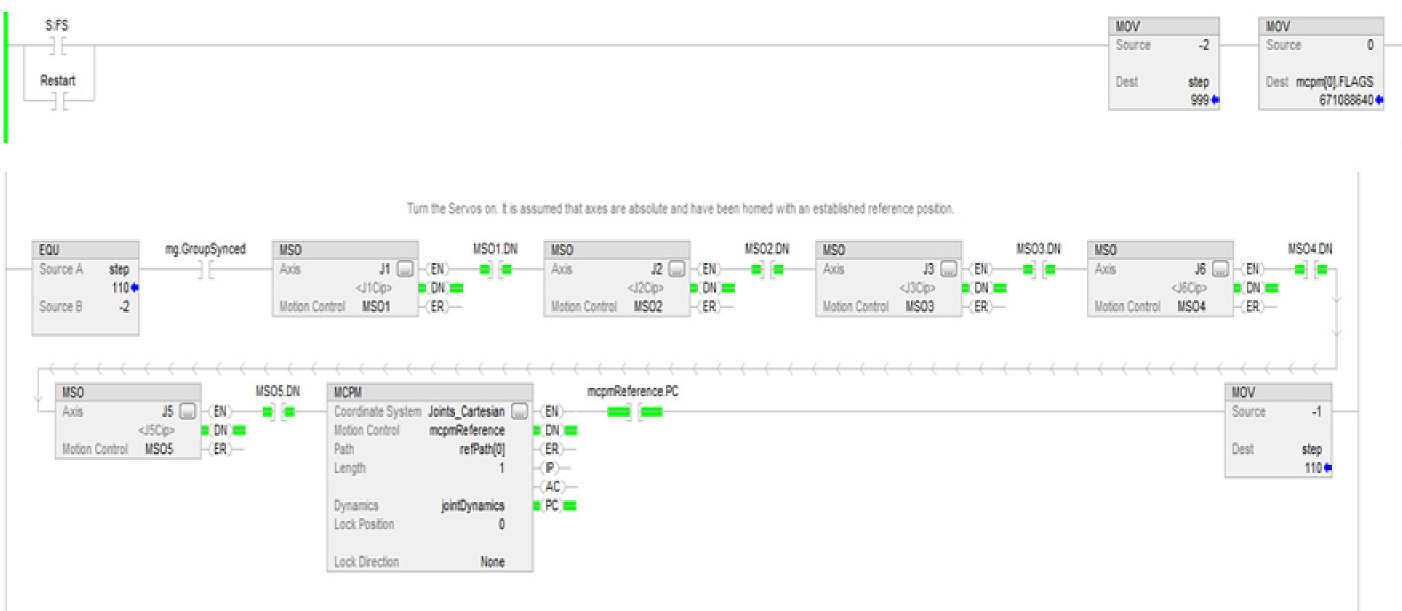
In this example, the source Cartesian coordinate system has six virtual axes X,Y,Z,Rx,Ry,Rz. The robot coordinate system has five real axes (J1,J2,J3,J4,J5). The example uses the MCTO instruction to establish the bidirectional transform relationship between these coordinate systems.

The example also contains a Joint Cartesian coordinate system for moving to a joint coordinate target point to establish initial positions or other joint positions. The Joint Cartesian systems has six axes (J1,J2,J3,J4,J5,J6). The J6 is a virtual axis, while the rest are real axes.

Tip: The Joint Cartesian coordinate system described here is not intended for use as the Cartesian coordinate system operand of the MCTO instruction.

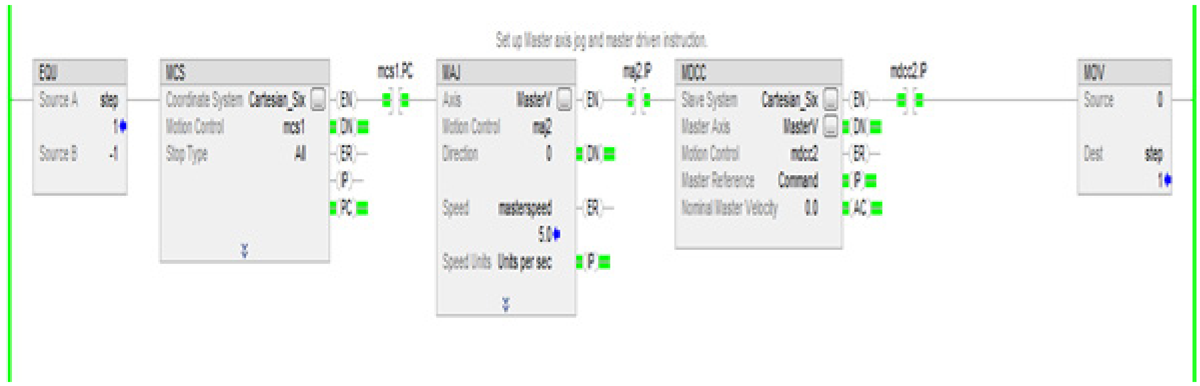
Align Cartesian and Robot Coordinate systems

The following ladder logic illustrates moving the robot coordinate system to an initial position before enabling the transformation. The transformation sets up the robot to a known position.



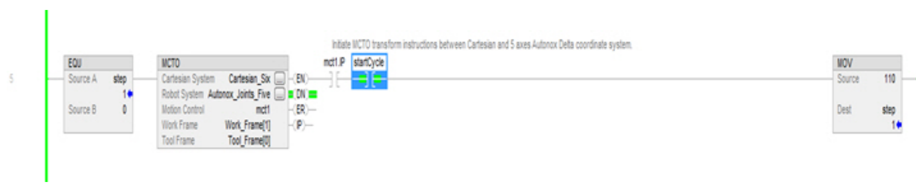
Set up Master Driven instructions for Cartesian dynamics control

This ladder logic illustrates setting up the Master Driven Speed Control (MDCC) instruction and jogging the master axis for the application.



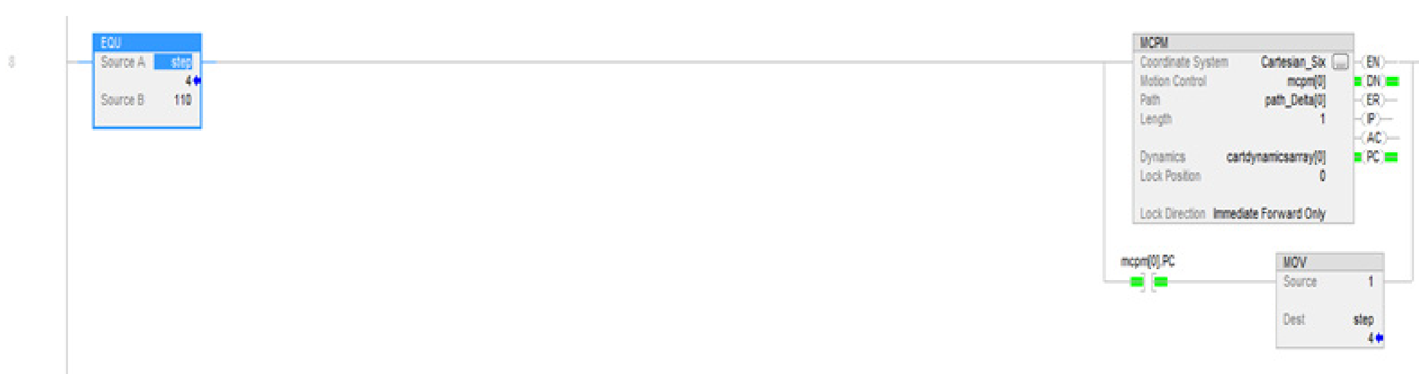
Initiate Transform instructions

This ladder logic illustrates enabling the transform instruction between the source Cartesian coordinate system and target 5 axis Delta robot system.



Move the source side to the desired target positions using MCPM path data with turns counter specifications

Refer to this ladder logic to command the robot to move to a target point in the Cartesian space specified by an element of an array of PATH_DATA points. See MCPM programming instructions and sample programs for details on ladder logic to move the robot through a series of such points.

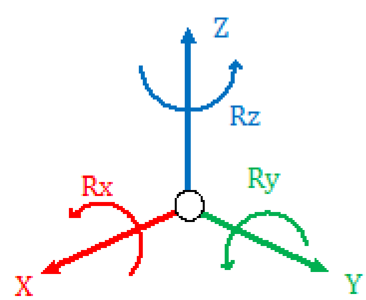


Name	Scope	Value	Force Mask	Description
◢ path_Delta[0]	Controller	{...}	{...}	
▸ path_Delta[0].InterpolationType	Controller	1		
◢ path_Delta[0].Position	Controller	{...}	{...}	
path_Delta[0].Position[0]	Controller	25.0		
path_Delta[0].Position[1]	Controller	25.0		
path_Delta[0].Position[2]	Controller	-1100.0		
path_Delta[0].Position[3]	Controller	180.0		
path_Delta[0].Position[4]	Controller	0.0		
path_Delta[0].Position[5]	Controller	45.0		
path_Delta[0].Position[6]	Controller	0.0		
path_Delta[0].Position[7]	Controller	0.0		
path_Delta[0].Position[8]	Controller	0.0		
▸ path_Delta[0].RobotConfiguration	Controller	0		
◢ path_Delta[0].TurnsCounters	Controller	{...}	{...}	
▸ path_Delta[0].TurnsCounters[0]	Controller	0		
▸ path_Delta[0].TurnsCounters[1]	Controller	1		
▸ path_Delta[0].TurnsCounters[2]	Controller	0		
▸ path_Delta[0].TurnsCounters[3]	Controller	0		
▸ path_Delta[0].MoveType	Controller	0		
▸ path_Delta[0].TerminationType	Controller	1		
path_Delta[0].CommandToleranceLinear	Controller	0.0		

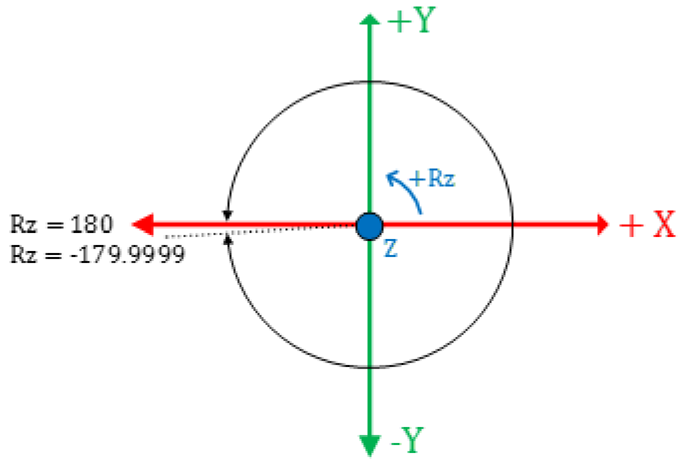
Program the MCPM target points as absolute move - MoveType = 0

The target position and orientation of any point defined has six coordinates XYZRxRyRz.

The translation coordinates are the coordinates of target point with respect to the base coordinate systems. The orientation coordinates are fixed angle rotations first around X axis followed by second rotation around Y axis of the fixed robot base frame and third rotation around Z axis of the fixed robot base frame.



The target specification typically has Rx = 180°, Ry = 0° and Rz equal to desired orientation. The Rz rotations have a range of +180° to -179.9999° as shown in this diagram that illustrates the top view from Z positive axis looking at the origin.

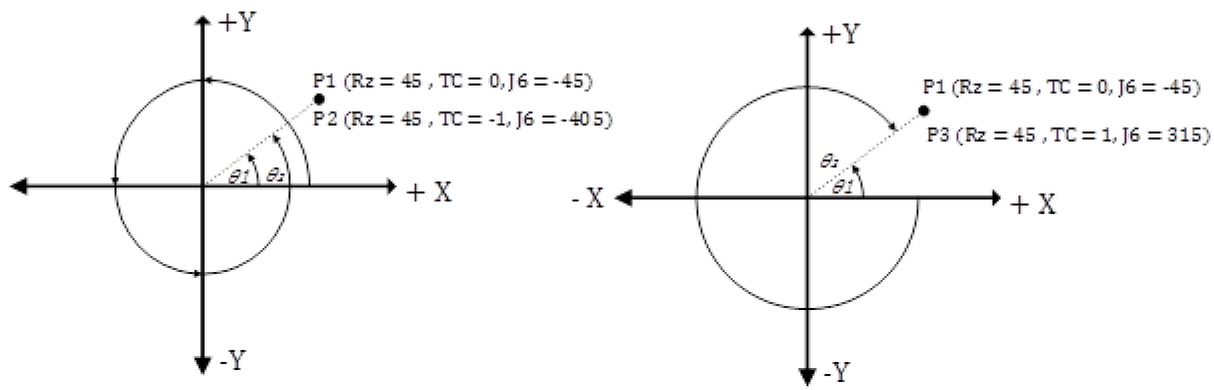


The orientation for any target point can be fully specified by Rx = 180°, Ry = 0° and Rz orientation in the range of +180° to -179.9999°.

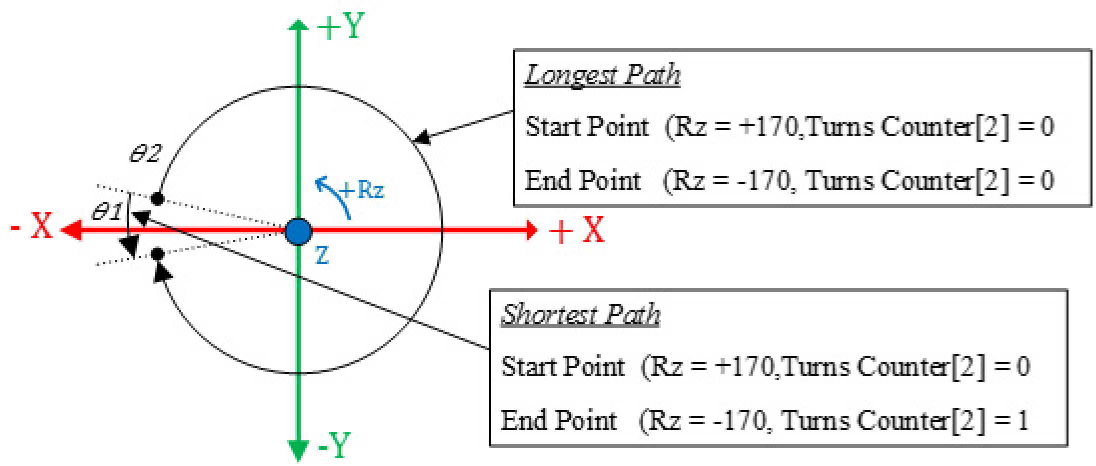
The turns counter is associated with Rz rotation and J6 axis for Delta J1J2J6 and Delta J1J2J3J6 robots. For Delta J1J2J3J4J5, the turns counter is associated with Rz rotation and J4 axis. The J6 or J4 axis rotates multiple revolutions around the Z axis shown in the previous diagram.

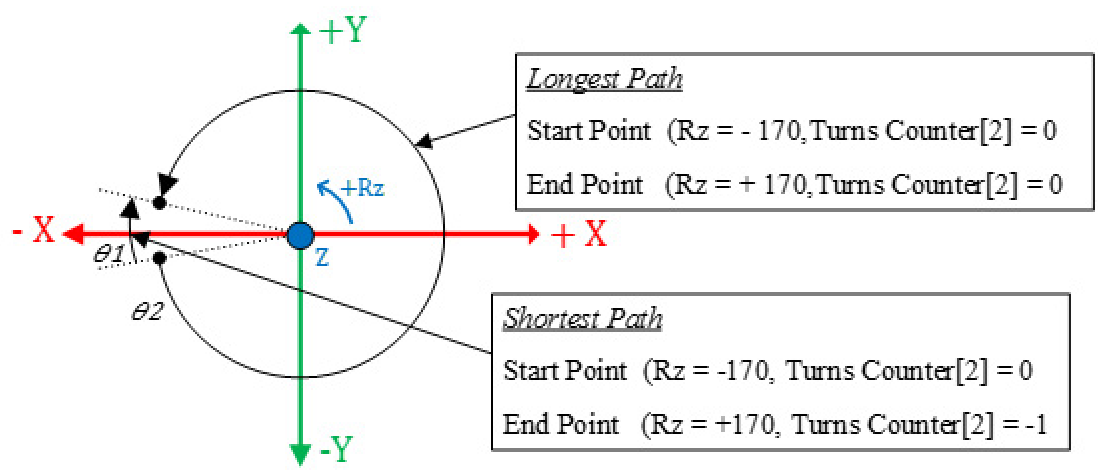
To fully specify the correct orientation, the Rz orientation must specify the desired orientation with which turn of joint axis. For example, +45° with turns counter 0 and + 45° with turns counter 1 and +45° with turns counter -1 are the same orientation but they are 360° apart from joint angle rotation point of view. Any point in the joint travel needs an additional turns counter specification for the Cartesian target point specification. See the following diagrams that show the 45° point with different turns.

Tip: Turns counters are only valid if MCTO is enabled on the Cartesian coordinate system. MCPM with nonzero turns counter will error if the MCTO is not enabled on the Cartesian coordinate system.



For programming the multi-turn axis, such as J6 for Delta J1J2J3J6, specify the shortest or longest path for J6 axis by specifying the Rz position and turns counter. See the following diagram for absolute moves.





The trends and tables show the complete specification of Cartesian target point for joint angles in the span of J6 travel.

These PATH_DATA points show typical target point specification for the MCPM instructions for the rung input in an excel spreadsheet for Delta J1J2J3J6 as absolute move with turns counter.

Position [0]	Position [1]	Position [2]	Position [3]	Position [4]	Position [5]	TurnsCounters[0]	TurnsCounters[1]	TurnsCounters[2]	RobotConfiguration	MoveType	InterpolationType	TerminationType
0	0	-782	180	0	90	0	0	0	0	0	1	1
0	0	-782	180	0	90	0	1	0	0	0	1	1
0	0	-782	180	0	180	0	0	0	0	0	1	1
0	0	-782	180	0	180	0	2	0	0	0	1	1
0	0	-782	180	0	-127	0	0	0	0	0	1	1
0	0	-782	180	0	-127	0	2	0	0	0	1	1
0	0	-782	180	0	-179.99	0	0	0	0	0	1	1
0	0	-782	180	0	-179.999	0	-3	0	0	0	1	1

Program the MCPM target points in incremental mode - MoveType = 1

The incremental moves are programmed differently and are not restricted to +/- 180°. Program multiple turns using just positive or negative displacements more than one turn. The system also enforces turns counters set to 0 in incremental move.

These PATH_DATA points show typical target point specification for the MCPM instructions for the rung input in an excel spreadsheet for Delta J1J2J3J6 as incremental move with turns counter.

Position [0]	Position [1]	Position [2]	Position [3]	Position [4]	Position [5]	TurnsCounters[0]	TurnsCounters[1]	TurnsCounters[2]	RobotConfiguration	MoveType	InterpolationType	TerminationType		
0	0	0	0	0	180	0	0	0	0	1	1	1	0	0
0	0	0	0	0	2520	0	0	0	0	1	1	1	0	0
0	0	0	0	0	-2520	0	0	0	0	1	1	1	0	0
0	0	0	0	0	45720	0	0	0	0	1	1	1	0	0
0	0	0	0	0	-45720	0	0	0	0	1	1	1	0	0
0	0	0	0	0	2340.01	0	0	0	0	1	1	1	0	0
0	0	0	0	0	2340.01	0	0	0	0	1	1	1	0	0
0	0	0	0	0	-4680.02	0	0	0	0	1	1	1	0	0
0	0	0	0	0	180	0	0	0	0	1	1	1	0	0
0	0	0	0	0	-360	0	0	0	0	1	1	1	0	0
0	0	0	0	0	180	0	0	0	0	1	1	1	0	0
0	0	0	0	0	-287	0	0	0	0	1	1	1	0	0

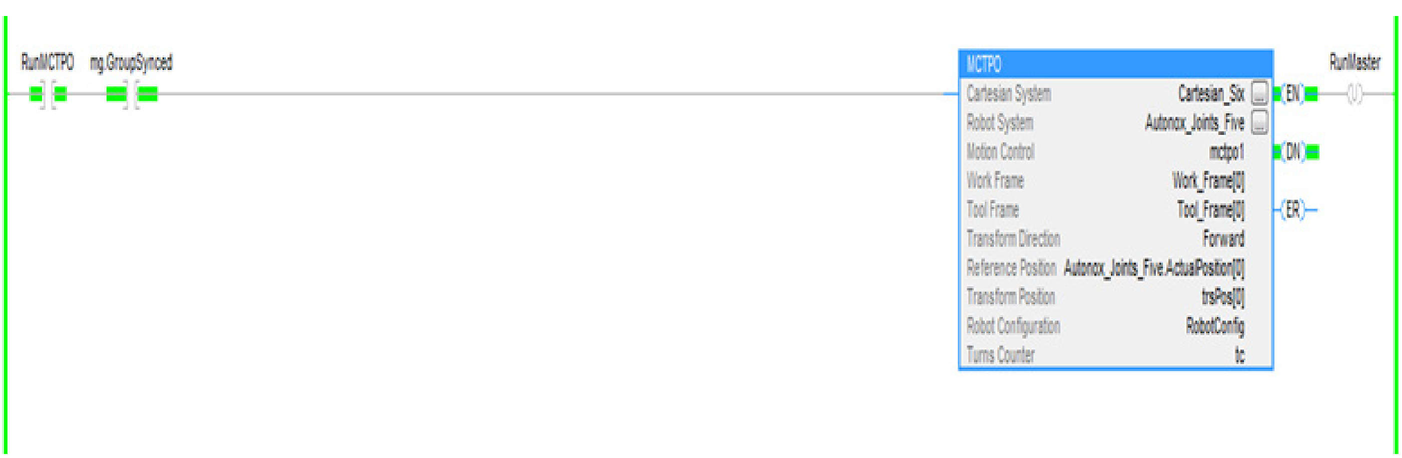
Teach positions for PATH_DATA target points for MCPM instructions using Coordinate System turns counter data

This section explains entering target points for turns counter. The system has turns counter template attributes for coordinate systems tag which keep track of turns counter once the MCTO is enabled on the coordinate system. If MCTO is not enabled these field get set to +128. The following figure shows the template information with the MCTO enabled. At any point the robot can be moved to desired position using HMI panel and the turns counter data along with Cartesian data can be used to program the target point for the MCPM move.

◢ Autonox_Joints_Five.TurnsCounters	{...}	{...}	Decimal	INT[4]
▸ Autonox_Joints_Five.TurnsCounters[0]	128		Decimal	INT
▸ Autonox_Joints_Five.TurnsCounters[1]	-2		Decimal	INT
▸ Autonox_Joints_Five.TurnsCounters[2]	128		Decimal	INT
▸ Autonox_Joints_Five.TurnsCounters[3]	0		Decimal	INT
▸ Autonox_Joints_Five.RobotConfiguration	0		Decimal	DINT

Getting positions for PATH_DATA target points for MCPM instruction using MCTPO turns counter data

Sometimes after powerup or shutdown, only joint positions are known while continuing from the current position. Use the MCTPO instruction to transform a point in joint target point to Cartesian target point by executing the MCTPO instruction perform a forward transform. At any point, use the MCTPO instruction to retrieve pertinent information like position, configuration, and turns counter. Use this data to program the target Cartesian point for MCPM Cartesian move. The following rung shows typical set up for MCTPO instruction.



Tip: To use this Kinematic sample projects, on the **Help** menu, click **Vendor Sample Projects** and then click the **Motion** category.
The Rockwell Automation sample project's default location is:
c:\Users\Public\Public Documents\Studio 5000\Sample\ENU\v<current_release>\Rockwell Automation

See also

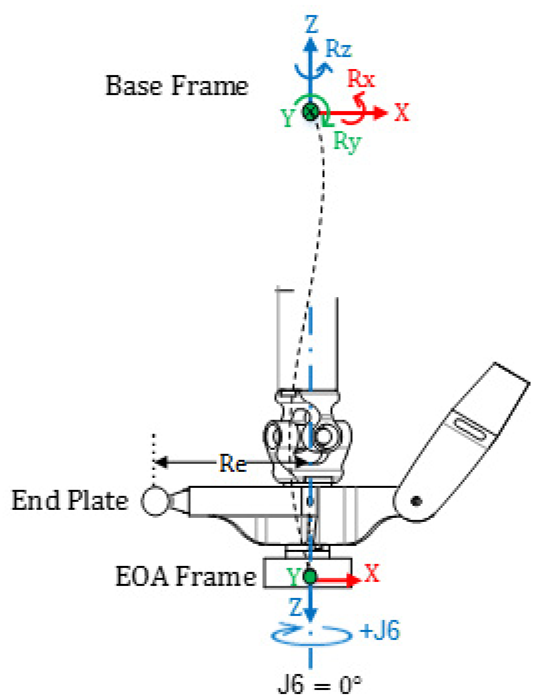
Configure and program turns counters on page 260

Configure and program turns counters

Use the MCTO instruction to establish a bidirectional transform between Cartesian and robot system with coordinates that are joint axes of a robot.

The Cartesian system coordinates are defined by XYZ translation coordinates and RxRyRz orientation coordinates in the fixed angle convention.

The robots have geometrical configurations where typically the joint axes are not orthogonal. The geometrical configurations are specified by coordinate system type, such as Delta. The coordinate definition attribute further specifies how many joint axes in the Robot coordinate system, such as J1,J2,J3,J6. This diagram shows the details of a Delta J1J2J3J6 robot with the base Cartesian coordinate system and four joint axes, which form the non-Cartesian coordinate system.



Cartesian and joint target points for Delta J1J2J3J6 robot system

A point in space may be described in two different ways; as a set of Cartesian coordinates (Euclidean space) and as a set of robot joint angles (joint space).

Since there is no rotation on Rx and Ry Orientation axis, only program the Rx orientation value to 180°. The Ry orientation is always 0°, and program the Rz orientation values within fixed XYZ Euler Angle range of Rz, that is, within +/-180°.

Joint axes for J1, J2 and J3 are typically configured as linear axis with over-travel limits. The J6 joint axis is also typically configured as a linear axis with over-travel limits.

Tips:
- For transformations to work correctly, be sure to establish the reference frame for the joint coordinate system first. For the Delta J1J2J3J6 and Delta J1J2J3J4J5 robots, the normal reference positions for J1, J2 and J3 axes are homed to 0° when the J1, J2 and J3 links are horizontal. The J6 axis is homed to 0° when it is parallel to J1 link.
- The J6 rotation is opposite to Rz rotation with respect to the robot base frame.

Once the robot reference frame is established, move the robot to a position in joint space, if needed, before enabling the MCTO instruction. After enabling the MCTO instruction, a bidirectional transform link is established so that, if the Cartesian coordinate is commanded to move to Cartesian coordinate target, the robot moves to Cartesian target coordinates along a linear path. Similarly, if the robot joint coordinate system is commanded to move to joint coordinate target, the robot moves to target joint coordinates along a non-Cartesian path. As the MCTO instruction is enabled, the system maintains the coordinate system related data (that is Cartesian position) for Cartesian and robot coordinate systems.

Turns counter

As shown in the previous diagram, positive orientation rotation for Rz is counterclockwise around the Z axis of the robot base frame. However, the positive rotation for J6 axis is clockwise around the Z axis of the robot base frame which is opposite to Rz axis rotation.

With the 3D Delta robot system since there is no rotation possible around X and Y axis of base frame, the only rotation possible is around Z axis. As a result, the Cartesian coordinate system can be described with the following translation and orientation specifications:
X, Y, Z: [-inf,+inf]
Rx: [180.0]
Ry: [0.0]
Rz: [-179.999, +180.0]

The Rz target position is the rotation around base Z axis and so any rotation can be specified with a range of +/- 180° with one exception of -180°. As 180° and -180° is the same point, the system does not allow specification of -180° as Rz target point.

However, this specification will not be complete as J6 axis can rotate more than one turn. The system handles this functionality by adding an additional turns counter specification for each target point specification.

Co-relating Rz axis with J6 axis and turns counter

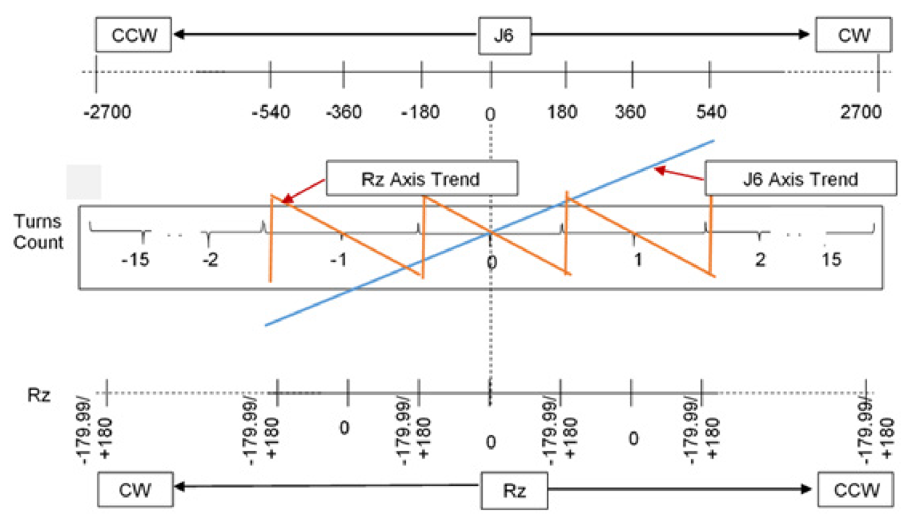
This diagram explains how Rz and turns counter varies with J6 (assuming that the work frame offset, the tool frame offset and the zero angle offset on J6 are 0). J6 is a linear axis and for example can have total travel of 15 revolutions with for example a range from -7.5*360 = -2700 to +7.5*360 = +2700°. As a result, physically the J6 can have multiple turns and have an attribute of turns counter which keeps track of number of the turns associated with the current position of J6 axis. When J6 crosses the 180° point in the CW direction, turns counter is incremented and Rz flips from -180° to 180° and when J6 goes past the 180° point in the CCW direction, turns counter is decremented and Rz flips from 180.0001° to -179.9999°.

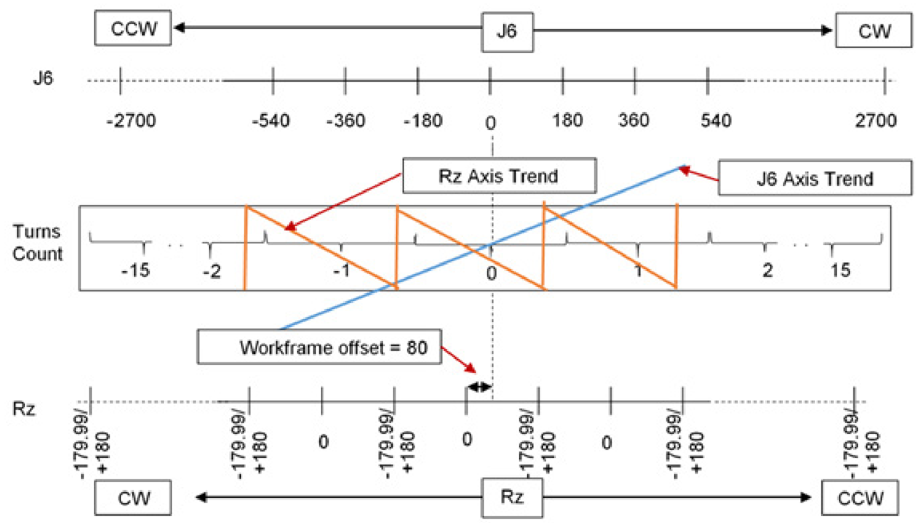
The range of turns counter is limited to +/-127 but the actual max number of turns is geometry dependent. The 3 Turns Counters are elements of a single array attribute of the target coordinate system which contain J1, J4 or J6 axes turns counters.

Tips:

- If Rz reaches the point 180° but does not cross it, it does not flip and stays at 180°. If Rz reaches the point -180°, it flips to +180°.
- If either the work frame or the tool frame offset on Rz is not 0, turns counters still increment when J6 crosses the 180° point, but Rz is flipped when J6 crosses the (180° + offset on Rz) point. In other words flip is shifted by offset on Rz as shown. See below for details.

Rz, J6 axis position and turns counter trends and tables





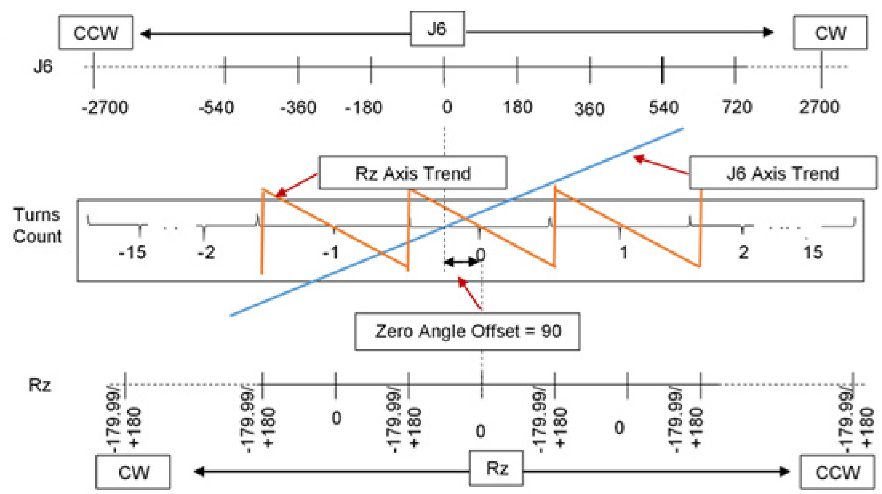


Table of Rz, turns counter and J6 values that are shown in the trends in figures above.

Rz	Turns Counter of J6	J6 (if zero angle offset = 0°) and (Rz work Offset = 0°)	J6 (if zero angle offset = 0°) and (Rz work offset = 80°)	J6 (if zero angle offset = 90°) and (work Offset = 0°)
+179.9999	2	540.0001	460.0001	630.0001
+180	2	540	460	630
-179.9999	1	539.9999	459.9999	629.9999
---	---	---		---
0	1	360	280	450
---	---	---		---
+179.9999	1	180.0001	100.0001	270.0001
+180	1	180	100	270
-179.9999	0	179.9999	99.9999	269.9999
---	---	---		---

Rz	Turns Counter of J6	J6 (if zero angle offset = 0°) and (Rz work Offset = 0°)	J6 (if zero angle offset = 0°) and (Rz work offset = 80°)	J6 (if zero angle offset = 90°) and (work Offset = 0°)
0	0	0	-80	90
---	---	---		---
+179.9999	0	-179.9999	-259.9999	-89.9999
+180	0	-180	-260	-90
-179.9999	-1	-180.0001	-260.0001	-90.0001

See also

Program example for turns counter on page 264

Program example for turns counter

The following is an example for programming a turns counter.

Configure Cartesian and robot coordinate systems

Refer to configuring Cartesian and robot coordinate systems for details of configuring the two coordinate systems that are used for the turns counter application example. The example uses the Delta J1J2J3J4J5 robot system.

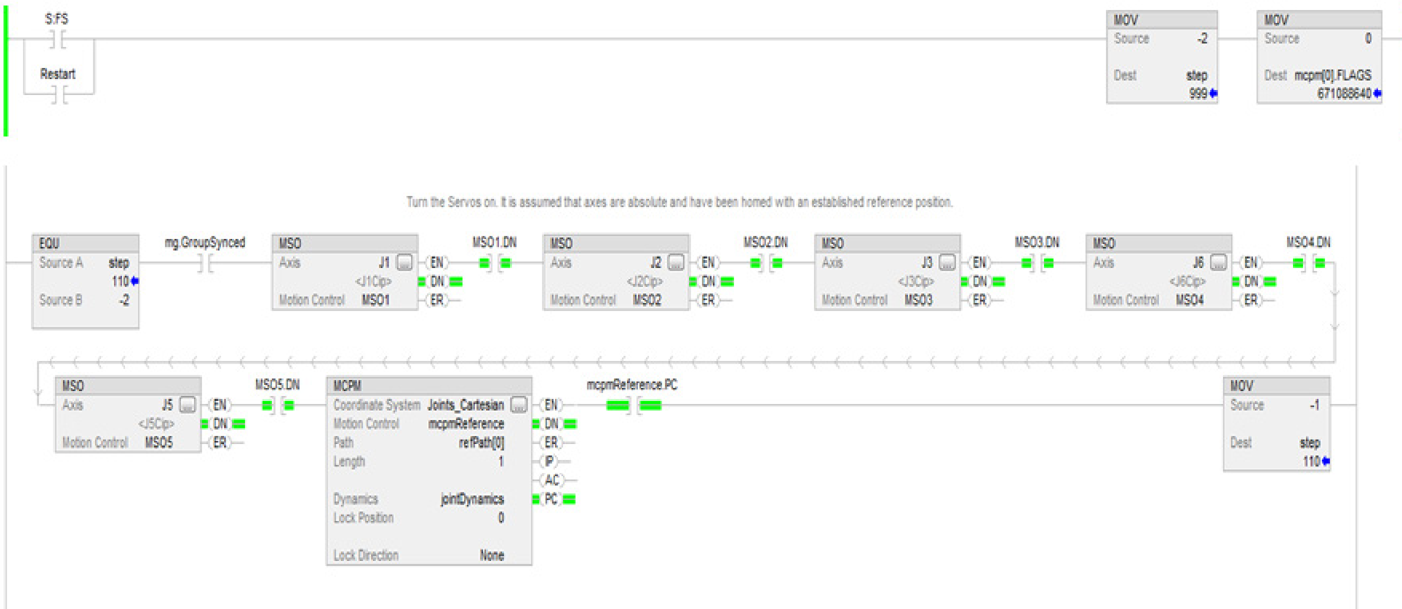
In this example, the source Cartesian coordinate system has six virtual axes X,Y,Z,Rx,Ry,Rz. The robot coordinate system has five real axes (J1,J2,J3,J4,J5). The example uses the MCTO instruction to establish the bidirectional transform relationship between these coordinate systems.

The example also contains a Joint Cartesian coordinate system for moving to a joint coordinate target point to establish initial positions or other joint positions. The Joint Cartesian systems has six axes (J1,J2,J3,J4,J5,J6). The J6 is a virtual axis, while the rest are real axes.

Tip: The Joint Cartesian coordinate system described here is not intended for use as the Cartesian coordinate system operand of the MCTO instruction.

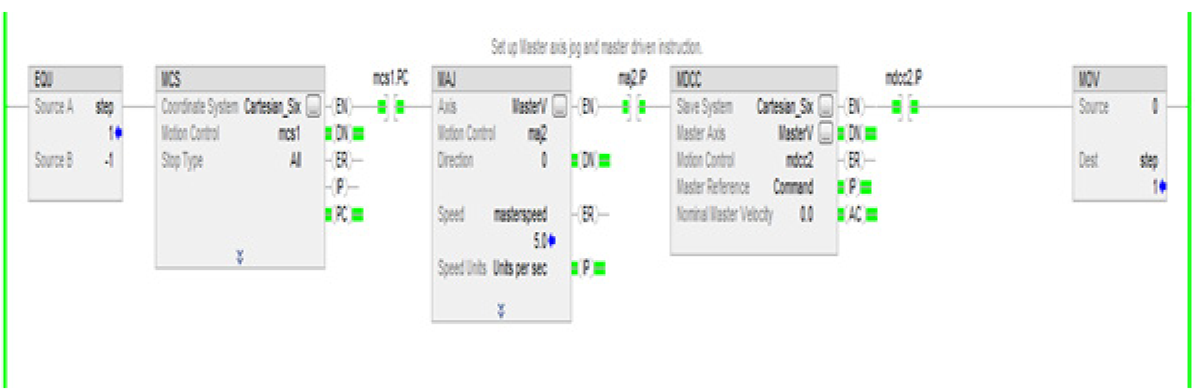
Align Cartesian and Robot Coordinate systems

The following ladder logic illustrates moving the robot coordinate system to an initial position before enabling the transformation. The transformation sets up the robot to a known position.



Set up Master Driven instructions for Cartesian dynamics control

This ladder logic illustrates setting up the Master Driven Speed Control (MDCC) instruction and jogging the master axis for the application.

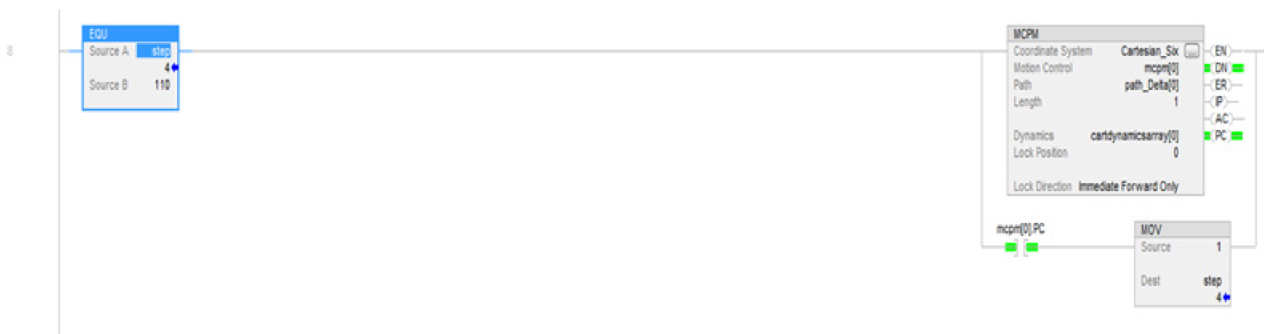


Initiate Transform instructions

This ladder logic illustrates enabling the transform instruction between the source Cartesian coordinate system and target 5 axis Delta robot system.

Move the source side to the desired target positions using MCPM path data with turns counter specifications

Refer to this ladder logic to command the robot to move to a target point in the Cartesian space specified by an element of an array of PATH_DATA points. See MCPM programming instructions and sample programs for details on ladder logic to move the robot through a series of such points.

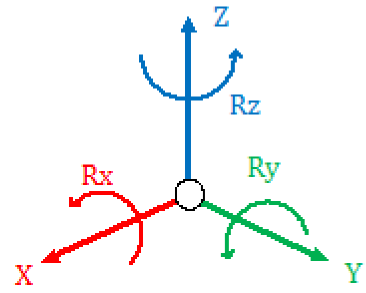


Name	Scope	Value	Force Mask	Description
path_Delta[0]	Controller	{...}	{...}	
path_Delta[0].InterpolationType	Controller	1		
path_Delta[0].Position	Controller	{...}	{...}	
path_Delta[0].Position[0]	Controller	25.0		
path_Delta[0].Position[1]	Controller	25.0		
path_Delta[0].Position[2]	Controller	-1100.0		
path_Delta[0].Position[3]	Controller	180.0		
path_Delta[0].Position[4]	Controller	0.0		
path_Delta[0].Position[5]	Controller	45.0		
path_Delta[0].Position[6]	Controller	0.0		
path_Delta[0].Position[7]	Controller	0.0		
path_Delta[0].Position[8]	Controller	0.0		
path_Delta[0].RobotConfiguration	Controller	0		
path_Delta[0].TurnsCounters	Controller	{...}	{...}	
path_Delta[0].TurnsCounters[0]	Controller	0		
path_Delta[0].TurnsCounters[1]	Controller	1		
path_Delta[0].TurnsCounters[2]	Controller	0		
path_Delta[0].TurnsCounters[3]	Controller	0		
path_Delta[0].MoveType	Controller	0		
path_Delta[0].TerminationType	Controller	1		
path_Delta[0].CommandToleranceLinear	Controller	0.0		

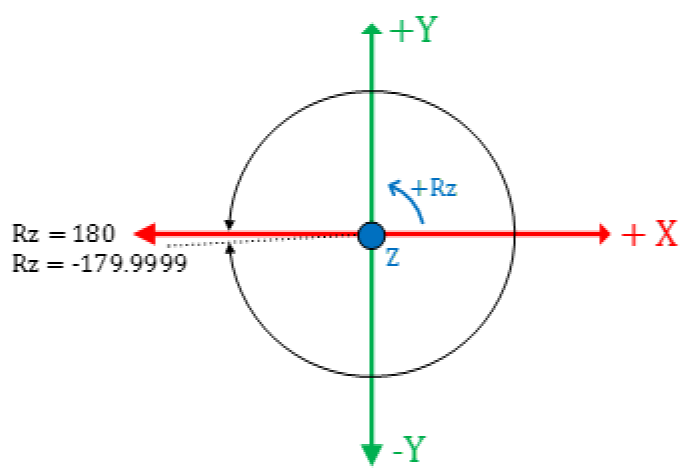
Program the MCPM target points as absolute move - MoveType = 0

The target position and orientation of any point defined has six coordinates XYZRxRyRz.

The translation coordinates are the coordinates of target point with respect to the base coordinate systems. The orientation coordinates are fixed angle rotations first around X axis followed by second rotation around Y axis of the fixed robot base frame and third rotation around Z axis of the fixed robot base frame.



The target specification typically has Rx = 180°, Ry = 0° and Rz equal to desired orientation. The Rz rotations have a range of +180° to -179.9999° as shown in this diagram that illustrates the top view from Z positive axis looking at the origin.

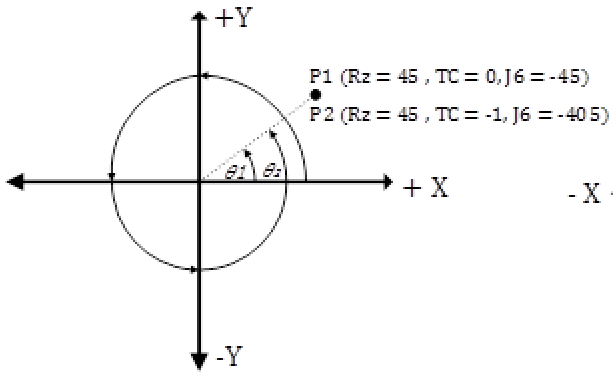


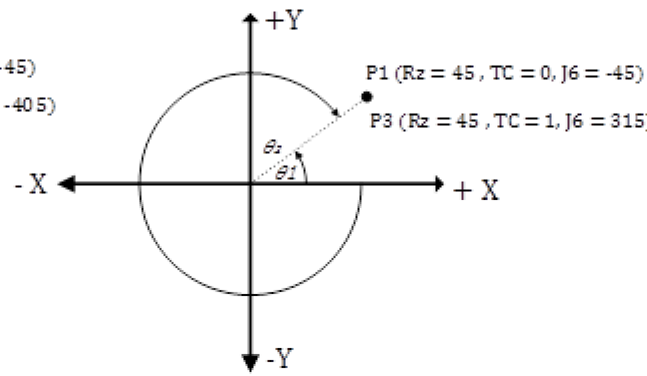
The orientation for any target point can be fully specified by Rx = 180°, Ry = 0° and Rz orientation in the range of +180° to -179.9999°.

The turns counter is associated with Rz rotation and J6 axis for Delta J1J2J6 and Delta J1J2J3J6 robots. For Delta J1J2J3J4J5, the turns counter is associated with Rz rotation and J4 axis. The J6 or J4 axis rotates multiple revolutions around the Z axis shown in the previous diagram.

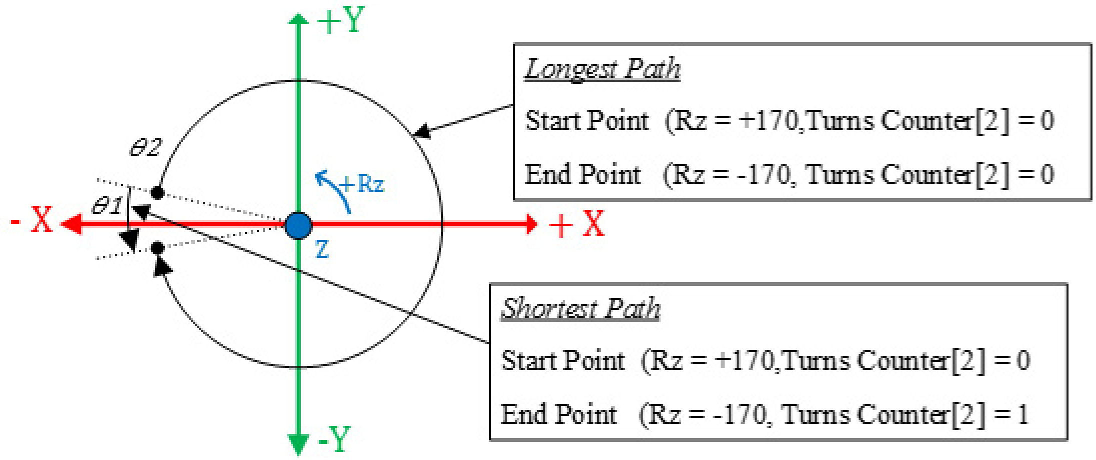
To fully specify the correct orientation, the Rz orientation must specify the desired orientation with which turn of joint axis. For example, +45° with turns counter 0 and + 45° with turns counter 1 and +45° with turns counter -1 are the same orientation but they are 360° apart from joint angle rotation point of view. Any point in the joint travel needs an additional turns counter specification for the Cartesian target point specification. See the following diagrams that show the 45° point with different turns.

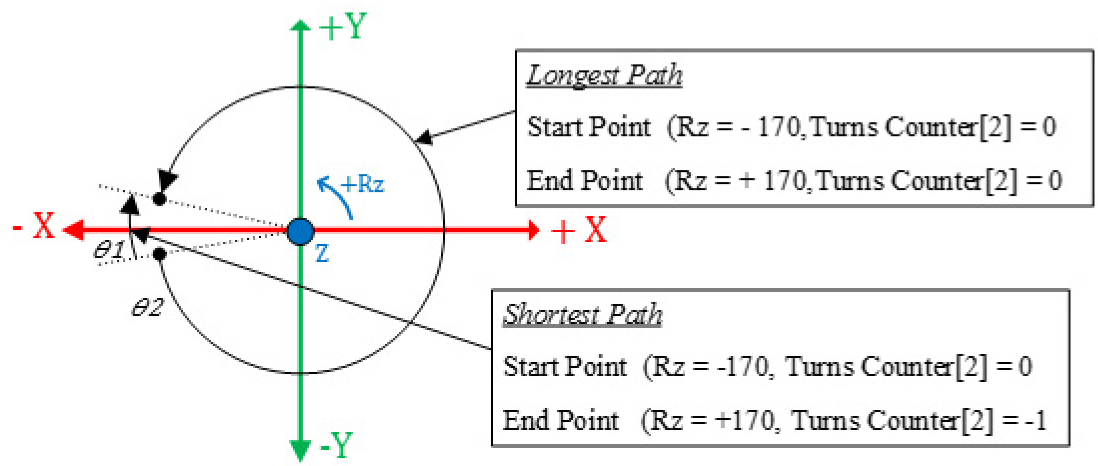
Tip: Turns counters are only valid if MCTO is enabled on the Cartesian coordinate system. MCPM with nonzero turns counter will error if the MCTO is not enabled on the Cartesian coordinate system.





For programming the multi-turn axis, such as J6 for Delta J1J2J3J6, specify the shortest or longest path for J6 axis by specifying the Rz position and turns counter. See the following diagram for absolute moves.





The trends and tables show the complete specification of Cartesian target point for joint angles in the span of J6 travel.

These PATH_DATA points show typical target point specification for the MCPM instructions for the rung input in an excel spreadsheet for Delta J1J2J3J6 as absolute move with turns counter.

Position [0]	Position [1]	Position [2]	Position [3]	Position [4]	Position [5]	TurnsCounters[0]	TurnsCounters[1]	TurnsCounters[2]	RobotConfiguration	MoveType	InterpolationType	TerminationType
0	0	-782	180	0	90	0	0	0	0	0	1	1
0	0	-782	180	0	90	0	1	0	0	0	1	1
0	0	-782	180	0	180	0	0	0	0	0	1	1
0	0	-782	180	0	180	0	2	0	0	0	1	1
0	0	-782	180	0	-127	0	0	0	0	0	1	1
0	0	-782	180	0	-127	0	2	0	0	0	1	1
0	0	-782	180	0	-179.99	0	0	0	0	0	1	1
0	0	-782	180	0	-179.999	0	-3	0	0	0	1	1

Program the MCPM target points in incremental mode - MoveType = 1

The incremental moves are programmed differently and are not restricted to +/- 180°. Program multiple turns using just positive or negative displacements more than one turn. The system also enforces turns counters set to 0 in incremental move.

These PATH_DATA points show typical target point specification for the MCPM instructions for the rung input in an excel spreadsheet for Delta J1J2J3J6 as incremental move with turns counter.

Position [0]	Position [1]	Position [2]	Position [3]	Position [4]	Position [5]	TurnsCounters[0]	TurnsCounters[1]	TurnsCounters[2]	RobotConfiguration	MoveType	InterpolationType	TerminationType		
0	0	0	0	0	180	0	0	0	0	1	1	1	0	0
0	0	0	0	0	2520	0	0	0	0	1	1	1	0	0
0	0	0	0	0	-2520	0	0	0	0	1	1	1	0	0
0	0	0	0	0	45720	0	0	0	0	1	1	1	0	0
0	0	0	0	0	-45720	0	0	0	0	1	1	1	0	0
0	0	0	0	0	2340.01	0	0	0	0	1	1	1	0	0
0	0	0	0	0	2340.01	0	0	0	0	1	1	1	0	0
0	0	0	0	0	-4680.02	0	0	0	0	1	1	1	0	0
0	0	0	0	0	180	0	0	0	0	1	1	1	0	0
0	0	0	0	0	-360	0	0	0	0	1	1	1	0	0
0	0	0	0	0	180	0	0	0	0	1	1	1	0	0
0	0	0	0	0	-287	0	0	0	0	1	1	1	0	0

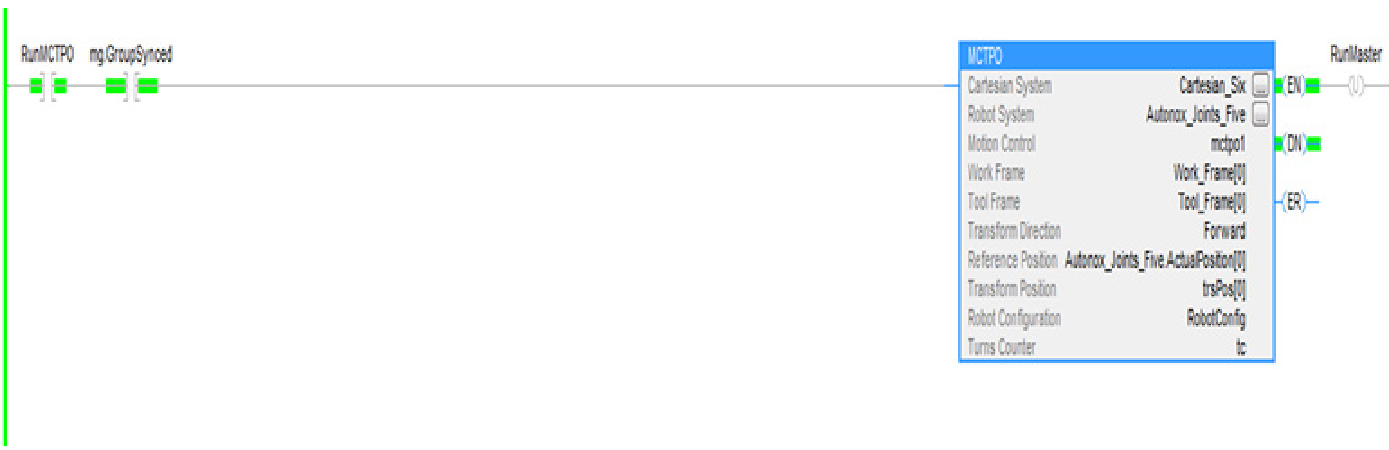
Teach positions for PATH_DATA target points for MCPM instructions using Coordinate System turns counter data

This section explains entering target points for turns counter. The system has turns counter template attributes for coordinate systems tag which keep track of turns counter once the MCTO is enabled on the coordinate system. If MCTO is not enabled these field get set to +128. The following figure shows the template information with the MCTO enabled. At any point the robot can be moved to desired position using HMI panel and the turns counter data along with Cartesian data can be used to program the target point for the MCPM move.

Autonox_Joints_Five.TurnsCounters	{...}	{...}	Decimal	INT[4]
Autonox_Joints_Five.TurnsCounters[0]	128		Decimal	INT
Autonox_Joints_Five.TurnsCounters[1]	-2		Decimal	INT
Autonox_Joints_Five.TurnsCounters[2]	128		Decimal	INT
Autonox_Joints_Five.TurnsCounters[3]	0		Decimal	INT
Autonox_Joints_Five.RobotConfiguration	0		Decimal	DINT

Getting positions for PATH_DATA target points for MCPM instruction using MCTPO turns counter data

Sometimes after powerup or shutdown, only joint positions are known while continuing from the current position. Use the MCTPO instruction to transform a point in joint target point to Cartesian target point by executing the MCTPO instruction perform a forward transform. At any point, use the MCTPO instruction to retrieve pertinent information like position, configuration, and turns counter. Use this data to program the target Cartesian point for MCPM Cartesian move. The following rung shows typical set up for MCTPO instruction.



Tip: To use this Kinematic sample projects, on the **Help** menu, click **Vendor Sample Projects** and then click the **Motion** category.
The Rockwell Automation sample project's default location is:
c:\Users\Public\Public Documents\Studio 5000\Sample\ENU\v<current_release>\Rockwell Automation

See also

Configure and program turns counters on page 260

Configure Camming

This information describes camming concepts. Use the motion coordinated instructions to move up to three axes in a coordinate system. Descriptions of these instructions are in the Logix 5000 Controllers Motion Instructions Reference Manual, publication MOTION-RM002.

See also

Caming concepts on page 285

Cam Profiles on page 286

Use Common Cam Profiles on page 288

Scaling cams on page 293

Execution Schedule on page 295

Camming concepts

Camming is the process of coordinating the movement of two axes, a master axis, and a slave axis, where the movement of one is completely dependent on the movement of the other.

There are two types of camming:

- Mechanical camming
- Electronic camming

See also

Mechanical camming on page 285

Electronic camming on page 286

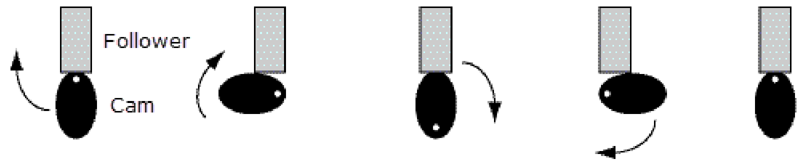
Mechanical camming

In mechanical camming, the master axis functions as a cam. A cam is an eccentric wheel mounted on a rotating shaft and used to produce variable or reciprocating motion in another engaged part, that is, the slave axis. The slave axis is also known as a follower assembly.

Mechanical camming has the following characteristics:

- There is a physical connection between the cam and the follower.
- The follower conforms to the cam shape as the cam unit rotates.
- Motion is limited by the cam shape.

The following illustrates a mechanical cam turning in a clockwise manner and the affect it has on a follower that is physically connected to it.



Electronic camming

Electronic camming is an electronic replacement for a mechanical camming. In this case, there is still a master axis that produces variable and reciprocating motion in a slave axis. However, electronic camming coordinates the movement of the two separate axes without a physical connection between them. There is no physical cam or follower assembly. In addition to removing the physical connection between axes, electronic camming:

- Creates coordinated motion profiles that are functions of the time o relative position of another axis.
- Allows you to configure higher cam velocities.
- Is defined by using a 'point pair' table of values. This table is a master axis set of point positioning values and a corresponding set of slave axis point positioning values.

The user-defined position point array causes one closed-loop axis to move with another open or closed-loop axis.

Cam Profiles

A cam profile is a representation of non-linear motion, that is, a motion profile that includes a start point, end point, and all points and segments in between. A cam profile is represented by an array of cam elements. The point pair used in a cam profile determines slave axis movement in response to master axis positions or times.

In a motion control application, you can use two different types of general cam profiles to accomplish electronic camming:

- Position Cam Profile
- Time Cam Profile

See also

Position Cam Profile on page 286

Time Cam Profile on page 287

Position Cam Profile

Position-lock cams provide the capability of implementing non-linear electronic gearing relationships between two axes based on a Cam Profile. Upon execution of this instruction, the axis specified as the slave is synchronized with the axis designated as the master. A position cam profile is defined by using a table of points that contains the following information:

- An array of master axis position values

- An array of slave axis position values

The master axis position values correspond to the slave axis position values. In other words, when the master axis reaches a specific position, the slave axis moves to its specific corresponding point, as defined in the cam profile's table of points.

Additionally, a position cam profile does the following:

- Provides the capability of implementing non-linear electronic gearing relationships between two axes
- Does not use maximum velocity, acceleration, or deceleration limits

Position cam profiles are used with Motion Axis Position Cam (MAPC) instructions. Upon execution of this instruction, the slave axis is synchronized with the master axis. See the Logix 5000 Controllers Motion Instructions Reference Manual, publication MOTION-RM002 for more information on how to configure the position cam profile in an MAPC instruction.

Linear and Cubic Interpolation

The resultant calculated cam profiles are fully interpolated. This means that if the current master position or time does not correspond exactly with a point in the cam array used to generate the cam profile, the slave axis position is determined by linear or cubic interpolation between adjacent points. In this way, the smoothest possible slave motion is provided. The MCCP instruction accomplishes this by calculating coefficients to a polynomial equation that determines slave position as a function of master position or time.

Each point in the cam array used to generate the position cam profile can be configured for linear or cubic interpolation. Electronic camming remains active through any subsequent execution of jog, or move processes for the slave axis. This allows electronic camming motions to be superimposed with jog, or move profiles to create complex motion and synchronization.

See also

Cam Profiles on page 286

Time Cam Profile

A time cam profile functions similarly to a cam drum driven by a constant speed motor. A time cam profile is also defined by using a table of points. However, with the time cam profile, the table contains the following information:

- An array of master axis time values
- An array of slave axis position values

The master axis time values correspond to slave axis position value. When the master axis reaches a specific point in time, the slave axis moves to a specific position as configured in the cam profile.

Time cam profiles are used with Motion Axis Time Cam (MATC) instructions.

Upon execution of this instruction, the slave axis is synchronized with the master axis.

See the Logix 5000 Controllers Motion Instructions Reference Manual, publication MOTION-RM002 for more information on how to configure the position cam profile in an MATC instruction.

Linear and Cubic Interpolation

Time cams are fully interpolated. This means that if the current master time value does not correspond exactly with a point in the cam table associated with the cam profile, the slave axis position is determined by linear or cubic interpolation between the adjacent points. In this way, the smoothest possible slave motion is provided. Each point in the cam array that was used to generate the time cam profile can be configured for linear or cubic interpolation. Electronic camming remains active through any subsequent execution of jog, or move processes for the slave axis. This allows electronic camming motions to be superimposed with jog, or move profiles to create complex motion and synchronization.

See also

Cam Profiles on page 286

Calculate a Cam Profile

You can use a Motion Calculate Cam Profile (MCCP) instruction to calculate a cam profile based on an array of cam points. You can establish an array of cam points programmatically or by using the Logix Designer software Cam Profile Editor. Each cam point in the cam array consists of a slave position value, a master position (position cam) or time (time cam) value, and an interpolation type (linear or cubic). An MAPC or MATC instruction can use the resulting cam profile to govern the motion of a slave axis according to master position or time.

See also

Cam Profiles on page 286

Use Common Cam Profiles

There are four common cam profiles that can be used as position cam or time cam profiles:

- Acceleration Cam Profile

- Run Cam Profile
- Deceleration Cam Profile
- Dwell Cam Profile

Cam profiles are configured for each required slave axis change of position, as corresponds to specific master axis position or time positions.

See also

Acceleration Cam Profile on page 289

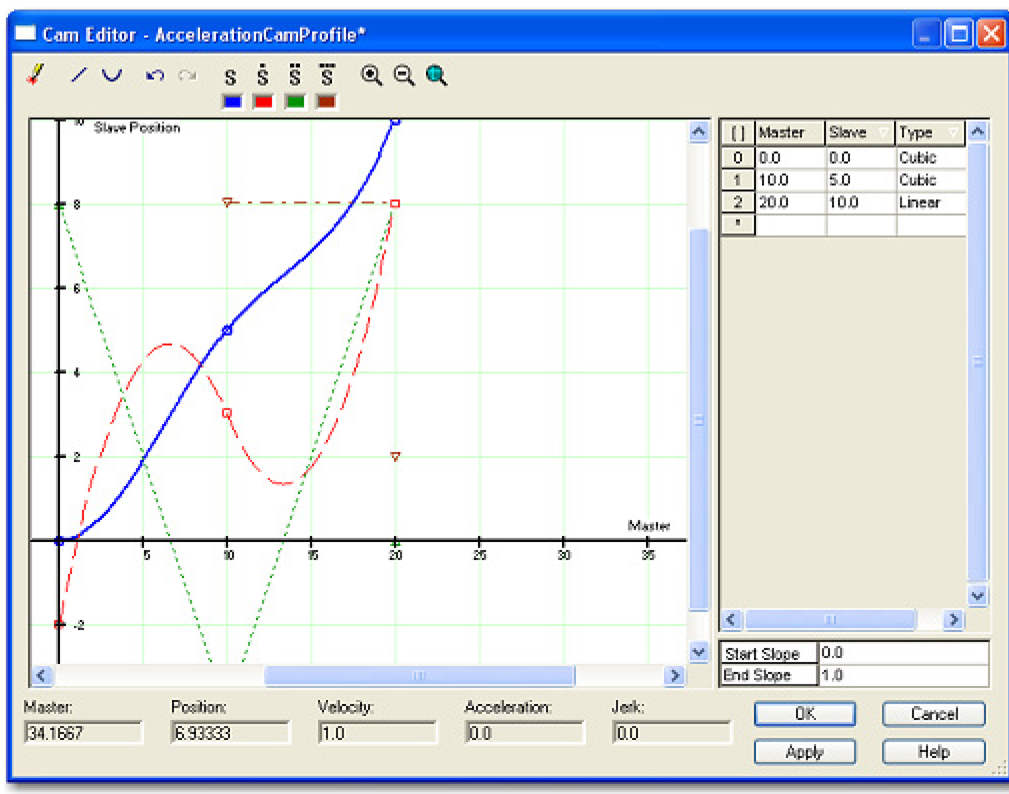
Run Cam Profile on page 289

Deceleration Cam Profile on page 291

Dwell Cam Profile on page 292

Acceleration Cam Profile

An acceleration cam profile determines a slave axis acceleration to a particular position. This graphic illustrates a sample acceleration cam profile in the Logix Designer programming software Cam Editor.

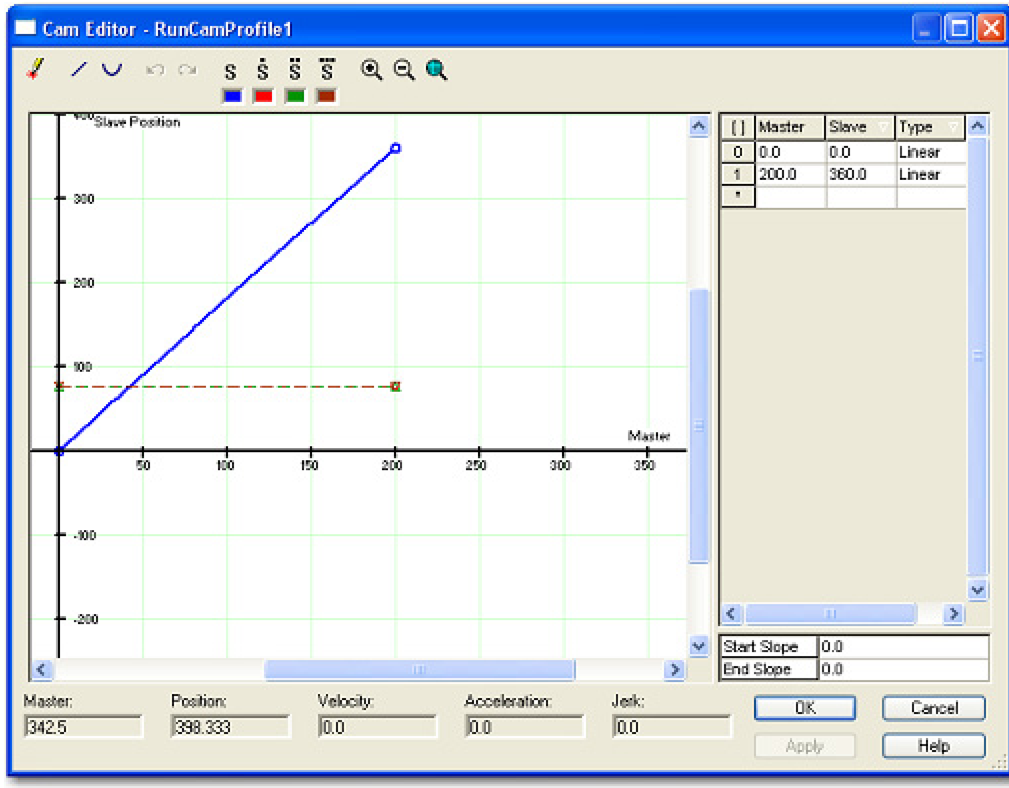


See also

Use Common Cam Profiles on page 288

Run Cam Profile

A run cam profile determines a slave axis' movement that begins when the master axis reaches a specific position and remains steady until the end of the cam profile. This graphic illustrates a sample run cam profile in the Logix Designer programming software Cam Editor.

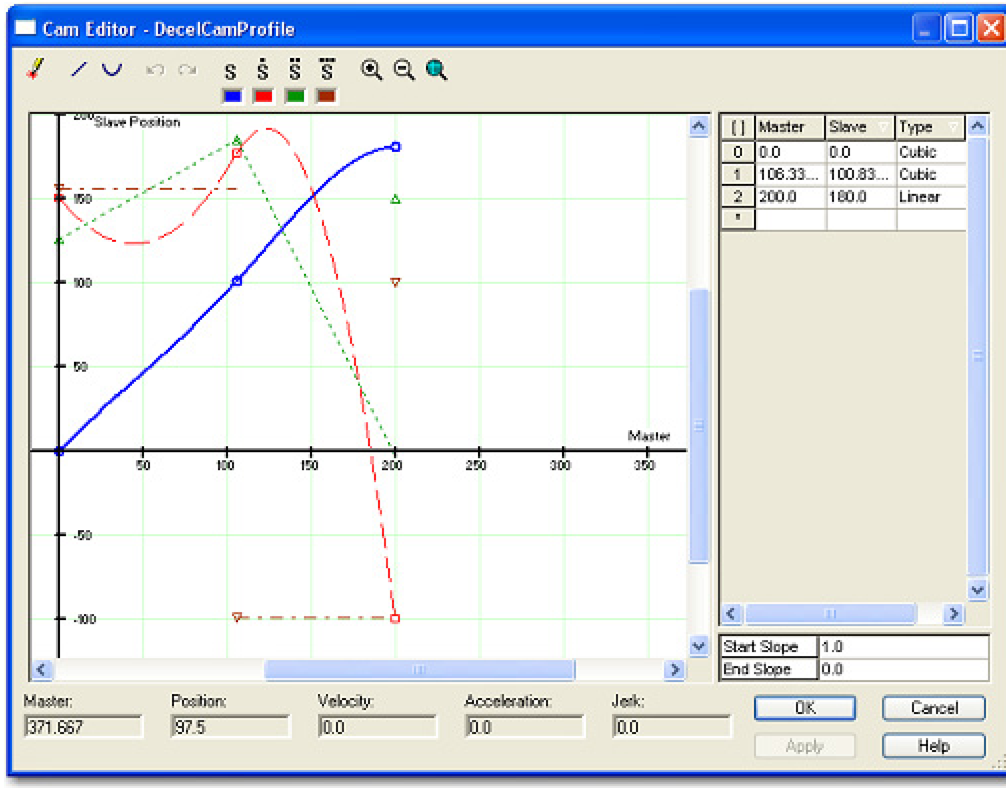


See also

Use Common Cam Profiles on page 288

Deceleration Cam Profile

A deceleration cam profile determines a slave axis' deceleration from a particular position. This graphic illustrates a sample deceleration cam profile in the Logix Designer programming software Cam Editor.

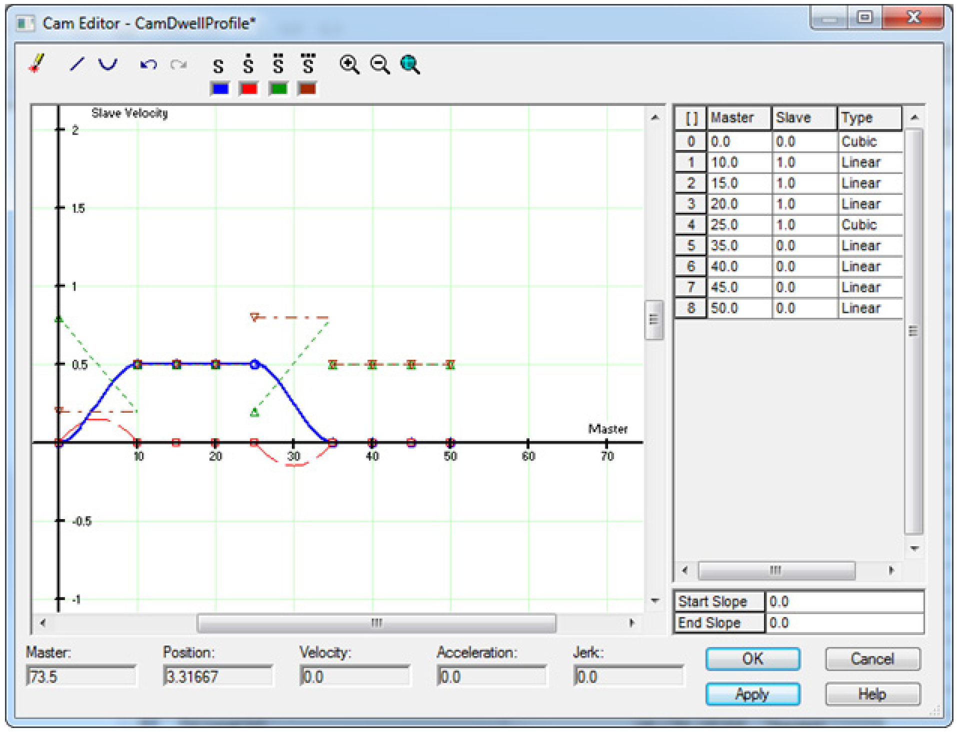


See also

Use Common Cam Profiles on page 288

Dwell Cam Profile

A dwell cam profile stops all slave axis movement until another cam profile begins operation. Typically, a dwell cam profile follows a deceleration cam profile. This graphic illustrates a sample dwell cam profile in the Logix Designer programming software cam editor.



See also

Use Common Cam Profiles on page 288

Behavior of Pending Cams

If you want to run one profile and then pend another one, you need to execute the MAPC instructions in the right order.

For example, if you want to run only one slave cycle, start with the Accel_Profile and pend the Decel_Profile immediately, that results in 2 x 1/2 Cycle = 1 Cycle.

These are executed at the same point in time:

- Set the execution schedule in the MAPC instruction for Acceleration as Immediate.
- Set the Deceleration to Pending.

Execution Schedule: Pending

See also

Use Common Cam Profiles on page 288

Scaling cams

You can use the scaling feature to determine the general form of the motion profile with a single stored cam profile. With this feature, one standard cam profile can be used to generate a family of specific cam profiles. Scaling works slightly differently when it is used with an MAPC instruction, that is, in position cam profiles, than when it is used with an MATC instruction, that is, in time cam profiles.

See also

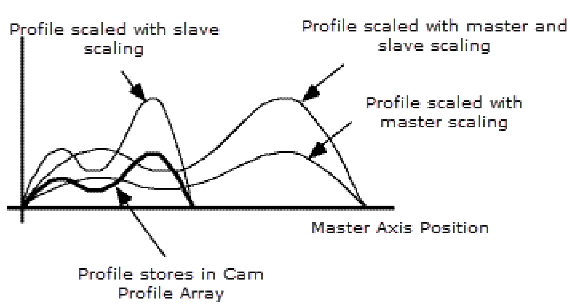
Scaling Position Cam Profile on page 293

Scaling Time Cam Profiles on page 294

Scaling Position Cam Profile

A position cam profile can be scaled in both the master dimension and slave dimension when it is executed. The scaling parameters are then used to define the total master or slave travel over which the profile is executed.

When an MAPC instruction specifies a position cam profile array, the master and slave values defined by the cam profile array take on the position units of the master and slave axes respectively. By contrast, the Master and Slave Scaling parameters are 'unit-less' values that are simply used as multipliers to the cam profile.



By default, both the Master Scaling and Slave Scaling parameters are set to 1. To scale a position cam profile, enter a Master Scaling or Slave Scaling value other than 1. Increasing the Master Scaling value of a position cam profile decreases the velocities and accelerations of the profile. However, increasing the slave scaling value increases the velocities and accelerations of the profile.

To maintain the velocities and accelerations of the scaled profile approximately equal to those of the unscaled profile, the Master Scaling and Slave Scaling values should be equal. For example, if the Slave Scaling value of a profile is 2, the Master Scaling value should also be 2 to maintain approximately equal velocities and accelerations during execution of the scaled position cam.

Important:	Decreasing the Master Scaling value or increasing the Slave Scaling value of a position cam increases the required velocities and accelerations of the profile. This can cause a motion fault if the capabilities of the drive system are exceeded.

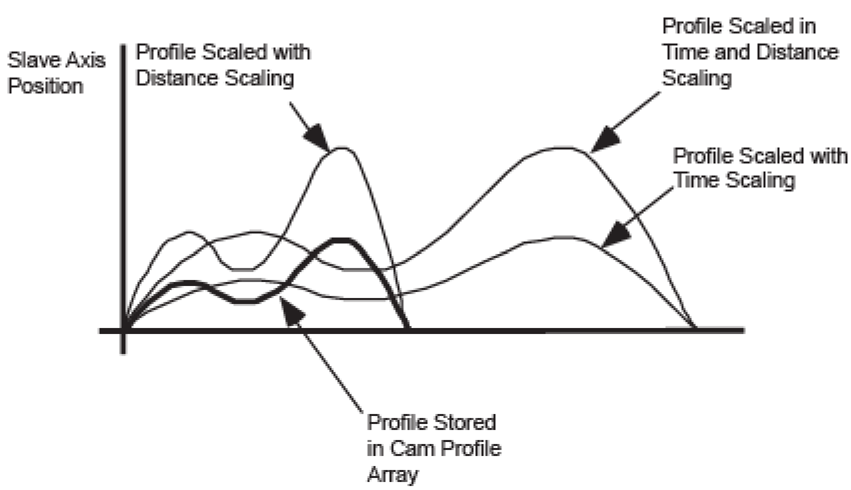
See also

Scaling Time Cam Profiles on page 294

Scaling cams on page 293

Scaling Time Cam Profiles

A time cam profile can be scaled in both time and distance when it is executed. The master coordinate values that the cam profile array defines take on the time units and the slave values take on the units of the slave axis. This process occurs when an MATC instruction specifies a time cam profile array. By contrast, the Time and Distance Scaling parameters are 'unitless' values that are used as multipliers to the cam profile.



By default, both the Time and Distance Scaling parameters are set to 1. To scale a time cam profile, enter a Time Scaling or Distance Scaling value other than 1. If you increase the Time Scaling value of a time cam profile, it decreases the velocities and accelerations of the profile. However, if you increase the Distance Scaling value, it increases the velocities and accelerations of the profile.

To maintain the velocities and accelerations of the scaled profile approximately equal to the values of the unscaled profile, the Time Scaling and Distance Scaling values must be equal. For example, if the Distance Scaling value of a profile is 2, the Time Scaling value must also be 2. This requirement is to maintain approximately equal velocities and accelerations during execution of the scaled time cam.

Important:	If you decrease the Time Scaling value or increase the Distance Scaling of a time cam, it increases the required velocities and accelerations of the profile. This action can cause a motion fault if the capabilities of the drive system are exceeded.

See also

Scaling Position Cam Profile on page 293

Scaling cams on page 293

Cam Execution Modes

Cam execution modes determine if the cam profile is executed only one time or repeatedly. Configure the Execution Mode parameter on an MAPC or MATC instruction.

Execution Mode	Description
Once	Cam motion of slave axis starts only when the master axis moves into the range defined by the start and end points of the cam profile. When the master axis moves beyond the defined range, cam motion on the slave axis stops and the Process Complete bit is set. Slave motion does not resume if the master axis moves back into the cam profile range.
Continuous	Once started, the cam profile is executed indefinitely. In this mode, the master and slave positions are unwound when the position of the master axis moves outside the profile range. This unwinding causes the cam profile to repeat. This feature is useful in rotary applications where it is necessary that the cam position runs continuously in a rotary or reciprocating fashion.
Persistent[1]	The cam motion of the slave axis proceeds only when the master axis moves within the range defined by the start and end points of the cam profile. When the master axis moves beyond the range of the profile, cam motion on the slave axis stops. Cam motion only resumes when the master moves back into the profile range specified by the start and end points.

[1] This section is only available on the MAPC instruction.

Execution Schedule

The Execution Schedule parameter controls the execution of an instruction. Configure the Execution Schedule parameter on an MAPC or MATC instruction. The Execution Schedule selections are different depending on which instruction, that is, the MAPC instruction or the MATC instruction, you are using.

See also

Execution Schedule for the MAPC Instruction on page 296

Execution Schedule for the MATC Instruction on page 299

Execution Schedule for the MAPC Instruction

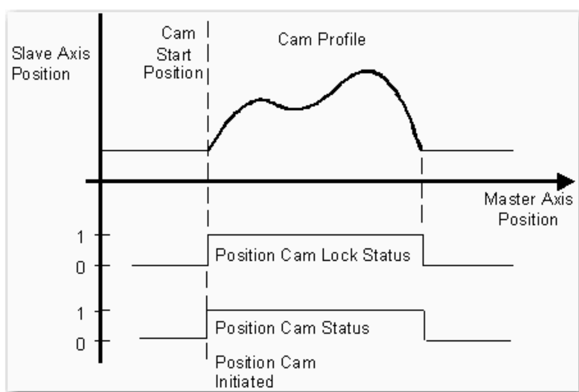
The Execution Schedule parameter selections are the following:

- Immediate
- Pending
- Forward Only
- Reverse Only
- Bidirectional

Immediate

By default, the MAPC instruction is scheduled to execute Immediately. In this case, there is no delay to the enabling of the position camming process and the

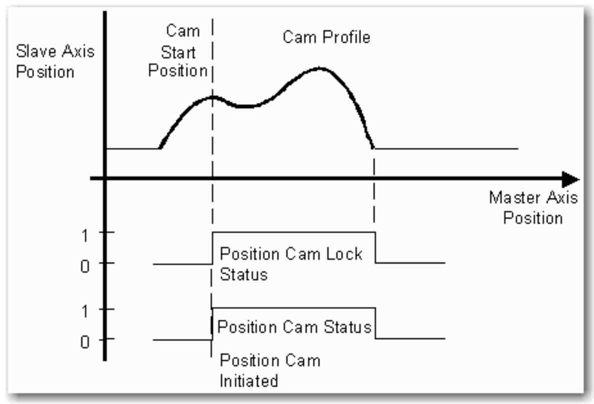
Master Lock Position parameter is irrelevant. The slave axis is immediately locked to the master axis, which begins at the Cam Lock Position of the specific cam profile. When the MAPC instruction is executed, the camming process is initiated on the specified slave axis. The Position Cam Status bit in the Motion Status word of the slave axis is also set. If the Execution Schedule is Immediate, the slave axis is immediately locked to the master according to the specified Cam Profile. The fact that the Position Cam Lock Status bit for the specified slave axis is also set indicates this condition.



Changing the Cam Lock Position on an MAPC Immediate Execution Schedule

The Cam Lock Position parameter of the MAPC instruction determines the starting location within the cam profile when the slave locks to the master. Typically, the Cam Lock Position is set to the beginning of the cam profile. Because the starting point of most cam tables is 0, the Cam Lock Position is typically set to 0. Alternatively, the Cam Lock Position can be set to any position within the master range of the cam profile. If a Cam Lock Position is specified that is out of this range, the MAPC instruction errors.

The following diagram shows the effect of specifying a Cam Lock Position value other than the starting point of the cam table. In this case, the value represents a position within the cam profile itself. Be careful not to define a Cam Start Point that results in a velocity or acceleration discontinuity to the slave axis if the master axis is moving.



Pending

The execution of an MAPC instruction can be deferred pending completion of a currently executing position cam. You can use Execution Schedule selection of Pending to blend two position cam profiles together without stopping motion. This Execution Schedule selection of Pending is fully described in Pending Cams topic.

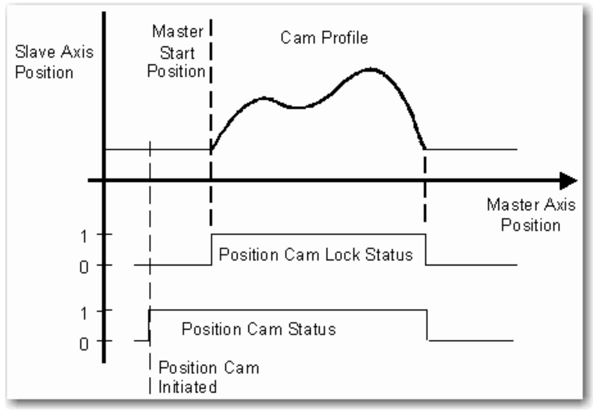
Forward Only, Reverse Only, or Bidirectional Execution Schedules

The slave axis is not locked to the master until the master axis satisfies the condition that is specified when the Execution Schedule parameter is set to any of the following parameters:

- Forward only
- Reverse only
- Bidirectional

With any of these selections, the camming process monitors the master axis to determine when the master axis passes the specified Master Lock Position in the specified direction. In a rotary axis configuration, this lock criterion is still valid, independent of the turns count.

Important: The cam profile generator monitors the master axis based on the absolute position reference system in effect before the redefine position operation.
This process only occurs if the position reference of the master axis is redefined with a Motion Redefine Position (MRP) instruction after the MAPC instruction executes but before the lock condition is satisfied.



The Position Cam Status bit of the Motion Status word for specified slave axis is set. This process occurs when the absolute position of the master axis passes the specified Master Lock Position in the specified direction. Slave axis motion is then initiated according to the specified cam profile starting at the specified Cam Lock Position of the cam profile.

From this point on, only the **incremental change** in the master axis position determines the corresponding slave axis position from the defined cam profile. This condition is important for applications where the master axis is a rotary axis because the position cam is then unaffected by the position unwind process.

When the master axis moves out of the range that the cam profile defines, if Execution Mode is Once, the following occur:

- It clears the Position Cam Lock Status
- It clears the Position Cam Status bits of the Motion Status word

This Motion Status bit condition indicates that the cam process has completed. This fact is also reflected in the bit leg behavior of the associated MAPC instruction, PC bit set, and IP bit clear.

The master axis can change direction and the slave axis reverses accordingly. This process occurs after position cam motion is started when the master axis passes the specified Master Lock Position in either the Forward Only or Reverse Only direction.

If an MAPC instruction is executed on a slave axis that is actively position camming, an Illegal Dynamic Change error is generated (error code 23). However, this error does not occur if the Execution Schedule is Pending.

See also

Execution Schedule on page 295

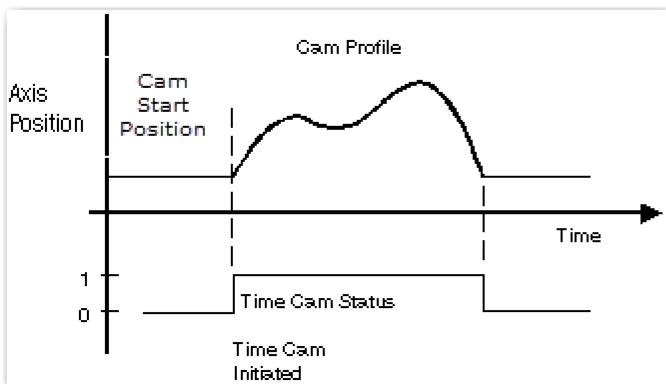
Execution Schedule for the MATC Instruction

An MATC instruction uses one of two Execution Schedule parameters:

- Immediate
- Pending

Immediate

Since the default setting of Execution Schedule is Immediate, the MATC instruction executes immediately. In this case, there is no delay to the enabling of the time camming process. When the MATC instruction is executed, the camming process is initiated on the specified axis. The Time Cam Status bit in the Motion Status word for the axis is also set. This process is shown in the following figure. If the Execution Schedule parameter is set to Immediate, the axis is immediately locked to the time master coordinate according to the specified Cam Profile.



If an MATC instruction is executed on an axis that is already actively time camming, an Illegal Dynamic Change error is generated (error code 23). The only exception for this occurrence is if the Execution Schedule is specified as pending.

Pending

The execution of a MATC instruction can be deferred pending completion of a currently executing time cam profile. You can use Execution Schedule selection of Pending to blend two time cam profiles together without stopping motion.

See also

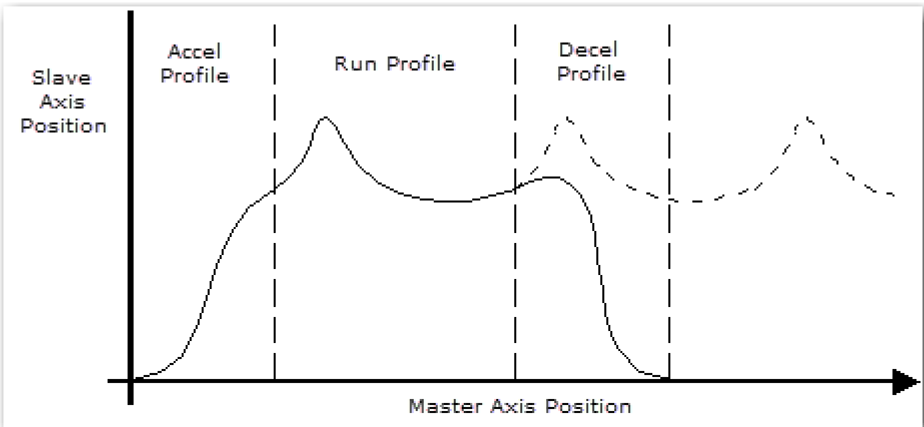
Execution Schedule on page 295

Pending Cams

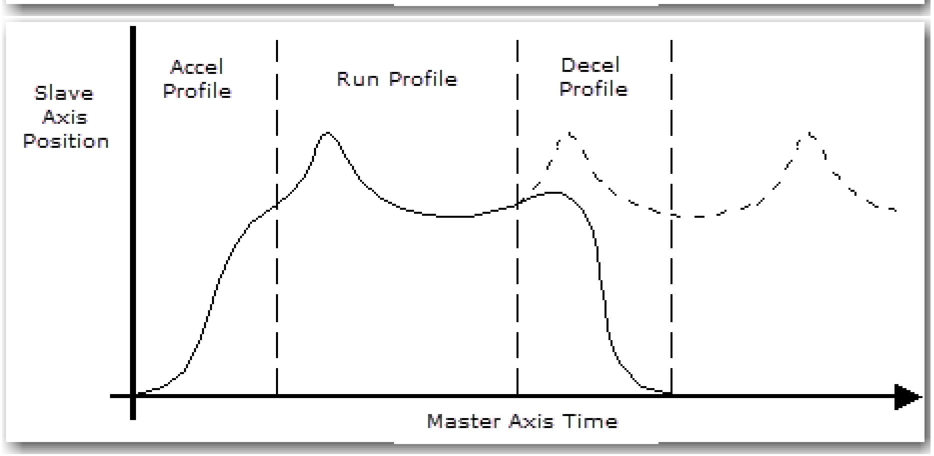
Cam pending is a technique that lets the blending of one cam profile together with another without stopping either master or slave axis movement. An Execution Schedule selection of Pending can thus be used to blend two position cam profiles together without stopping motion.

The Pending execution feature is useful when the axis must be accelerated up to speed by using a specific velocity profile. When this acceleration profile is done, it must be smoothly blended into the operating cam profile, which is typically executed continuously. To stop the slave axis, the operating cam profile is smoothly blended into a deceleration profile such that the axis stops at a known location, as shown in this diagram.

MAPC Instruction



MATC Instruction



By executing the position cam profile as a Pending cam profile while the current profile is still executing, the appropriate cam profile parameters are configured ahead of time. This condition makes the transition from the current profile to the pending profile seamless. Synchronization between the master and slave axes is maintained. To make sure of smooth motion across the transition, however, the profiles must be designed as follows. No position, velocity, or acceleration discontinuities can exist between the end of the current profile and the start of the new one. This process is done by using the Logix Designer Cam Profile Editor.

Once a pending position cam instruction has been executed, the new cam profile takes effect automatically (and becomes the current profile). This process occurs when the master axis passes through either the start or end point of the current profile. If the current cam is configured to execute once, the new profile is initiated at the completion of the current cam profile. The PC bit of the currently active instruction (either MAPC or MATC) is also set.

If the current cam is configured to execute continuously, the new profile is initiated at the completion of the current pass through the current cam profile. The IP bit of the currently active instruction is also cleared. The motion controller tracks the master axis position or time, depending on which instruction is used. The slave axis position relative to the first profile at the time of the change and uses this information to maintain synchronization between the profiles.

If the Execution Schedule of an instruction is set to Immediate and a position or time cam profile is in process, the instruction errs. In this case, the instruction generates an Illegal Dynamic Change error, error code 23, in the programming software. This error even occurs when the axis is waiting to lock onto the master axis. If an Execution Schedule of Pending is selected without a corresponding position or time cam profile in progress, the instruction executes. However, no camming motion occurs until another instruction with a non-pending Execution Schedule is initiated. This process allows pending cam profiles to be preloaded before executing the initial cam. This method addresses cases where immediate cams would finish before the pending cam could be reliably loaded.

The Position or Time Cam Pending Status bit of the Motion Status word for the specified slave axis is set to 1 (true). This process occurs after a Pending position cam has been configured. When the pending (new) profile is initiated and becomes the current profile, Position or Time Cam Pending Status bit is immediately cleared as shown in this diagram.

Index

Rockwell Automation support

Use these resources to access support information.

Technical Support Center	Find help with how-to videos, FAQs, chat, user forums, and product notification updates.	rok.auto/support
Knowledgebase	Access Knowledgebase articles.	rok.auto/knowledgebase
Local Technical Support Phone Numbers	Locate the telephone number for your country.	rok.auto/phonesupport
Literature Library	Find installation instructions, manuals, brochures, and technical data publications.	rok.auto/literature
Product Compatibility and Download Center (PCDC)	Get help determining how products interact, check features and capabilities, and find associated firmware.	rok.auto/pcdc

Documentation feedback

Your comments help us serve your documentation needs better. If you have any suggestions on how to improve our content, complete the form at rok.auto/docfeedback.

Waste Electrical and Electronic Equipment (WEEE)

At the end of life, this equipment should be collected separately from any unsorted municipal waste.

Rockwell Automation maintains current product environmental information on its website at rok.auto/pec.

Allen-Bradley, expanding human possibility, Logix, Rockwell Automation, and Rockwell Software are trademarks of Rockwell Automation, Inc.

EtherNet/IP is a trademark of ODVA, Inc.

Trademarks not belonging to Rockwell Automation are property of their respective companies.

Rockwell Otomayson Ticaret A.Ş. Kar Plaza İş Merkezi E Blok Kat:6 34752, İçerenkÖy, İstanbul, Tel: +90 (216) 5698400 EEE YÖnetmeliğine Uygundur

Connect with us.

rockwellautomation.com — expanding **human possibility**™

AMERICAS: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
EUROPE/MIDDLE EAST/AFRICA: Rockwell Automation NV, Pegasus Park, De Kleetlaan 12a, 1831 Diegem, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
ASIA PACIFIC: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Rockwell Automation Publication MOTION-UM002I-EN-P - March 2022
Supersedes Publication MOTION-UM002H-EN-P - November 2021

Copyright © 2022 Rockwell Automation Technologies, Inc. All Rights Reserved. Printed in the U.S.A.