Motion Coordinate System

Catalog Numbers 1756-HYD02, 1756-M02AE, 1756-M02AS, 1756-M03SE, 1756-M08SE, 1756-M16SE, 1768-M04SE
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).
<table>
<thead>
<tr>
<th>Table of Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary of Changes</td>
</tr>
<tr>
<td>Preface</td>
</tr>
<tr>
<td>Motion Coordinate Instructions</td>
</tr>
<tr>
<td>Where to Find Sample Projects</td>
</tr>
<tr>
<td>Additional Resources</td>
</tr>
</tbody>
</table>

### Create and Configure a Coordinate System

- Create a Coordinate System | 14 |
- Coordinate System Wizard Dialog Boxes | 17 |
- Edit Coordinate System Properties | 18 |
  - General Tab | 19 |
  - Geometry Tab | 21 |
  - Units Tab | 22 |
  - Offsets Tab | 23 |
  - Joints Tab | 24 |
  - Dynamics Tab | 25 |
  - Dynamics Tab Manual Adjust | 27 |
- Motion Planner Tab | 28 |
- Tag Tab | 29 |

### Chapter 2

- Program an MCLM Instruction | 32 |
- Blended Moves and Termination Types | 32 |
  - Example Ladder Diagram for Blended Instructions | 33 |
- Bit State Diagrams for Blended Moves | 35 |
  - Bit States at Transition Points of Blended Move by Using Actual Tolerance or No Settle | 35 |
  - Bit States at Transition Points of Blended Move by Using No Decel | 36 |
  - Bit States at Transition Points of Blended Move by Using Command Tolerance | 37 |
  - Bit States at Transition Points of Blended Move by Using Follow Contour Velocity Constrained or Unconstrained | 38 |
- Choose a Termination Type | 39 |
  - Velocity Profiles for Collinear Moves | 41 |
  - Symmetric Profiles | 43 |
  - Triangular Velocity Profile | 45 |
  - Blending Moves at Different Speeds | 46 |

### Chapter 3

- Motion Calculate Transform Position (MCTP) | 47 |
- Motion Coordinated Shutdown Reset (MCSR) | 47 |
- Useful Terms | 48 |
# Gather Information about Your Robot

Summary of Kinematic Steps .................................................. 48
Determine the Coordinate System Type ................................. 50

## Chapter 4

Configure an Articulated Independent Robot .......................... 53

Methods to Establish a Reference Frame ................................. 55
  Method 1 - Establishing a Reference Frame ....................... 56
  Method 2 - Establishing a Reference Frame ....................... 57

Chapter 4

Reference Frame ...................................................................... 53
Methods to Establish a Reference Frame ................................. 55
  Method 1 - Establishing a Reference Frame ....................... 56
  Method 2 - Establishing a Reference Frame ....................... 57

Work Envelope ........................................................................ 57
Configuration Parameters ...................................................... 59
  Link Lengths ..................................................................... 60
  Base Offsets ..................................................................... 61
  End-effector Offsets .......................................................... 62

Delta Robot Geometries .......................................................... 62

Configure a Delta Three-dimensional Robot ........................... 62
  Establish the Reference Frame for a Delta Three-dimensional Robot ...................... 64
  Calibrate a Delta Three-dimensional Robot ........................................... 64
  Alternate Method for Calibrating a Delta Three-dimensional Robot ...................... 65
  Configure Zero Angle Orientations for Delta Three-dimensional Robot ................... 65

Identify the Work Envelope for a Delta Three-dimensional Robot ...................... 67
  Define Configuration Parameters for a Delta Three-dimensional Robot ................... 69

Configure a Delta Two-dimensional Robot .............................. 71
  Establish the Reference Frame for a Delta Two-dimensional Robot ...................... 72
  Calibrate a Delta Two-dimensional Robot ........................................... 73

Define Configuration Parameters for a Delta Two-dimensional Robot ...................... 74

Configure a SCARA Delta Robot .............................................. 76
  Establish the Reference Frame for a SCARA Delta Robot ................................. 76
  Calibrate a SCARA Delta Robot ............................................... 77

Define Configuration Parameters for a SCARA Delta Robot ................................. 78

Configure a Delta Robot with a Negative X1b Offset .................. 80

Arm Solutions ........................................................................ 81
# Table of Contents

- Left-Arm and Right-Arm Solutions for Two-Axes Robots 
- Solution Mirroring for Three-dimensional Robots 
- Activating Kinematics 
- Change the Robot Arm Solution 
- Plan for Singularity 
- Encounter a No-solution Position 
- Configure a SCARA Independent Robot 
- Establish the Reference Frame for a SCARA Independent Robot 
- Identify the Work Envelope for a SCARA Independent Robot 
- Define Configuration Parameters for a SCARA Independent Robot 
- Error Conditions 
- Monitor Status Bits for Kinematics 

## Chapter 5

### Configure an Articulated Dependent Robot

- Reference Frame 
- Methods to Establish a Reference Frame
  - Method 1 - Establishing a Reference Frame 
  - Method 2 - Establishing a Reference Frame 
- Work Envelope 
- Configuration Parameters 
  - Link Lengths 
  - Base Offsets 
  - End-effector Offsets 

## Chapter 6

### Configure a Cartesian Gantry Robot

- Establish the Reference Frame for a Cartesian Gantry Robot 
- Identify the Work Envelope for a Cartesian Gantry Robot 
- Define Configuration Parameters for a Cartesian Gantry Robot 

## Chapter 7

### Configure a Cartesian H-bot

- About Cartesian H-bots 
- Establish the Reference Frame for a Cartesian H-bot 
- Identify the Work Envelope for a Cartesian H-bot 
- Define Configuration Parameters for a Cartesian H-bot
# Chapter 8

Configure Camming

Camming Concepts ................................................. 107
Mechanical Camming ............................................. 107
Electronic Camming ............................................... 108
Cam Profiles ......................................................... 109
  Position Cam Profile ........................................... 109
  Time Cam Profile ............................................... 110
  Calculating a Cam Profile .................................... 111
Using Common Cam Profiles .................................... 111
  Acceleration Cam Profile ...................................... 111
  Run Cam Profile ................................................. 112
  Deceleration Cam Profile ..................................... 113
  Dwell Cam Profile .............................................. 113
  Behavior of Pending Cams .................................... 114
Scaling Cams ......................................................... 115
  Scaling Position Cam Profiles ............................... 115
  Scaling Time Cam Profiles .................................... 116
Cam Execution Modes ............................................. 117
  Execution Schedule ............................................ 117
  MAPC Instruction ............................................... 117
  MATC Instruction .............................................. 121
Pending Cams ......................................................... 122

Index ................................................................. 127
This manual contains new and updated information as indicated in the following table.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moved the Motion Coordinated Instruction Appendix to (MCLM, MCCM, MCCD, MCS, MCSR, MCT, MCTP, MDCC) MOTION-RM002</td>
<td>—</td>
</tr>
<tr>
<td>Move the Coordinate System Attributes Appendix to Online Help</td>
<td>—</td>
</tr>
<tr>
<td>Moved the Error Codes (ERR) for Coordinate Motion Instructions Appendix to MOTION-RM002</td>
<td>—</td>
</tr>
<tr>
<td>Added Motion Coordinate Instructions section to Preface</td>
<td>9</td>
</tr>
<tr>
<td>Added the Camming Appendix from MOTION-RM002 to Chapter 8 of this manual</td>
<td>107</td>
</tr>
</tbody>
</table>
Notes:
Preface

Motion Coordinate Instructions

This manual provides information on how to configure various coordinated motion applications. Appendix A provides detailed information about the coordinated motion instructions. See the Additional Resources section for information configuration and startup of Sercos and analog motion or Integrated Motion on EtherNet/IP networks.

Use this table to choose a motion coordinated instruction. Detailed information about these coordinate instructions can be found in the Logix5000™ Controllers Motion Instructions Reference Manual, publication MOTION-RM002.

<table>
<thead>
<tr>
<th>If you want to</th>
<th>Use this instruction</th>
<th>Available in these languages</th>
</tr>
</thead>
</table>
| Initiate a single or multi-dimensional linear coordinated move for the specified axes within a Cartesian coordinate system. | Motion Coordinated Linear Move (MCLM) | • Relay ladder  
• Structured text |
| Initiate a two- or three-dimensional circular coordinated move for the specified axes within a Cartesian coordinate system. | Motion Coordinated Circular Move (MCCM) | • Relay ladder  
• Structured text |
| Initiate a change in path dynamics for coordinate motion active on the specified coordinate system. | Motion Coordinated Change Dynamics (MCCD) | • Relay ladder  
• Structured text |
| Stop the axes of a coordinate system or cancel a transform. | Motion Coordinated Stop (MCS) | • Relay ladder  
• Structured text |
| Initiate a controlled shutdown of all of the axes of the specified coordinate system. | Motion Coordinated Shutdown (MCSD) | • Relay ladder  
• Structured text |
| Start a transform that links two coordinate systems together. | Motion Coordinated Transform (MCT)\(^{(1)}\) | • Relay ladder  
• Structured text |
| Calculate the position of one coordinate system with respect to another coordinate system. | Motion Calculate Transform Position (MCTP)\(^{(1)}\) | • Relay ladder  
• Structured text |
| 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) | • Relay ladder  
• Structured text |

\(^{(1)}\) You cannot use this instruction with SoftLogix™ controllers.

Where to Find Sample Projects

Use the Studio 5000 Logix Designer® application Start Page (Alt F9) to find the sample projects.
The default location of the Rockwell Automation sample project is:
C:\Users\Public\Documents\Studio 5000\Samples\ENU\n
There is a PDF file that is named Vendor Sample Projects on the Start Page that explains how to work with the sample projects. Free sample code is available at: http://samplecode.rockwellautomation.com/.

**Additional Resources**

These documents contain additional information concerning related products from Rockwell Automation.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sercos and Analog Motion Configuration and Startup User Manual</td>
<td>Provides installation instructions for the Analog Encoder (AE) Serve Module, Catalog Number 1756-M02AE.</td>
</tr>
<tr>
<td>Logix5000 Controllers Motion Instructions Reference Manual</td>
<td>Provides installation instructions for the ControlLogix SERCOS interface modules, Catalog Number 1756-M02SE, 1756-M08SE, 1756-M16SE, 1756-M08SEG.</td>
</tr>
<tr>
<td>Integrated Motion on the Ethernet/IP Network:</td>
<td>Provides installation instructions for the CompactLogix SERCOS interface Module, Catalog Number 1768-M04SE.</td>
</tr>
<tr>
<td>Logix5000 Controllers Common Procedures,</td>
<td>Provides detailed and comprehensive information about how to program a Logix5000 controller.</td>
</tr>
<tr>
<td>Logix5000 Controllers General Instructions Reference Manual</td>
<td>Provides detailed and comprehensive information about how to program a Logix5000 controller.</td>
</tr>
<tr>
<td>Logix5000 Controllers Advanced Process Control and Drives</td>
<td>Provides detailed and comprehensive information about how to program a Logix5000 controller.</td>
</tr>
<tr>
<td>ControlLogix System User Manual</td>
<td>Provides detailed and comprehensive information about how to program a Logix5000 controller.</td>
</tr>
<tr>
<td>GuardLogix 5570 Controllers User Manual</td>
<td>Provides detailed and comprehensive information about how to program a Logix5000 controller.</td>
</tr>
<tr>
<td>GuardLogix 5570 and Compact GuardLogix 5370 Controller Systems</td>
<td>Provides detailed and comprehensive information about how to program a Logix5000 controller.</td>
</tr>
<tr>
<td>Analog Encoder (AE) Servo Module Installation Instructions</td>
<td>Provides detailed and comprehensive information about how to program a Logix5000 controller.</td>
</tr>
<tr>
<td>CompactLogix SERCOS interface Module Installation Instructions</td>
<td>Provides detailed and comprehensive information about how to program a Logix5000 controller.</td>
</tr>
<tr>
<td>Industrial Automation Wiring and Grounding Guidelines,</td>
<td>Provides detailed and comprehensive information about how to program a Logix5000 controller.</td>
</tr>
<tr>
<td>Product Certifications website,</td>
<td>Provides detailed and comprehensive information about how to program a Logix5000 controller.</td>
</tr>
</tbody>
</table>
You can view or download publications at http://www.rockwellautomation.com/literature/. To order paper copies of technical documentation, contact your local Allen-Bradley distributor or Rockwell Automation sales representative.
Create and Configure a Coordinate System

In the Studio 5000 Logix Designer® application, you use the Coordinate System tag to configure a coordinate system. A coordinate system is a grouping of one or more primary and ancillary axes that you create to generate coordinated motion.

You can configure the coordinate system with one, two, or three dimensions. The programming software supports these types of geometry:

- Cartesian
- Articulated Dependent
- Articulated Independent
- Selective Compliant Assembly Robot Arm (SCARA) Independent
- Delta three-dimensional
- Delta two-dimensional
- SCARA Delta

Figure 1 - Coordinate Systems with Orthogonal Axes
Create a Coordinate System

Use the Coordinate System tag to set the attribute values that the Multi-Axis Coordinated Motion instructions use in your motion applications. The Coordinate System tag must exist before you can run any of the Multi-Axis Coordinated Motion instructions.

Now you make the following configurations:

- Introduce the COORDINATE_SYSTEM data type
- Associate the coordinate system to a Motion Group
- Associate the axes to the coordinate system
- Set the dimension
- Define the values that are later used by the operands of the Multi-Axis Motion Instructions

The information included when you configure the Coordinate System tag defines the values for Coordination Units, Maximum Speed, Maximum Acceleration, Maximum Deceleration, Actual Position Tolerance, and Command Position Tolerance.
Follow these steps to create a coordinate system.

1. Right-click the motion group in the Controller Organizer.
2. Choose New Coordinate System.

The New Tag dialog box appears.

Use the parameter descriptions in Table 1 to help you configure your new tag.
Click Open COORDINATE_SYSTEM Configuration to display the wizard that guides you through the process of configuring a coordinate system. You can also right-click the tag and choose Properties to access the configuration wizard.
Create and Configure a Coordinate System

Chapter 1

Coordinate System Wizard Dialog Boxes

The Coordinate System Wizard takes you through the Coordinate System Properties dialog boxes. It is not necessary to use the Wizard dialogs to configure your coordinate system. Once it has been created, you can access the Coordinate System Properties dialog box by choosing Properties of the menu. See Edit Coordinate System Properties on page 18 for detailed information on how to enter configuration information.

Table 2 - Coordinate System Dialog Box Descriptions

<table>
<thead>
<tr>
<th>Wizard or Dialog Box</th>
<th>Description</th>
</tr>
</thead>
</table>
| **General**          | The General dialog box lets you:  
  • associate the tag to a Motion Group.  
  • enter the coordinate system type.  
  • select the Dimension for the tag (that is, the number of associated axes).  
  • specify the number of dimensions to transform.  
  • enter the associated axis information.  
  • choose whether to update Actual Position values of the coordinate system automatically during operation.  
  This dialog box has the same fields as the General tab found under Coordinate System Properties. |
| **Geometry**         | The Geometry dialog box allows you to configure key attributes that are related to non-Cartesian geometry and shows the bitmap of the associated geometry. |
| **Offset**           | The Offset dialog box allows you to configure the offsets for the base and end effector. This dialog box shows the bitmaps for the offsets related to the geometry. |
| **Units**            | The Units dialog box allows you to determine the units that define the coordinate system. At this dialog box, you define the Coordination Units and the Conversion Ratios. This dialog box has the same fields as the Units tab found under Coordinate System Properties. |
| **Dynamics**         | Use the Dynamics dialog box for entering the Vector values used for Maximum Speed, Maximum Acceleration, and Maximum Deceleration. It is also used for entering the Actual and Command Position Tolerance values. This dialog box has the same fields as the Dynamics tab found under Coordinate System Properties. |
| **Manual Adjust**    | The Manual Adjust button is inactive when creating a Coordinate System tag via the Wizard dialog boxes. It is active on the Dynamics tab of the Coordinate System Properties dialog box. It is described in detail in the Editing Coordinate System Properties later in this chapter. |
| **Tag**              | The Tag dialog box allows you to rename your Tag, edit your description, and review the Tag Type, Data Type, and Scope information. The only fields that you can edit on the Tag dialog box are Name and Description. These fields are the same fields as on the New Tag dialog box and the Coordinate System Properties Tag tab. |
Edit Coordinate System Properties

Create your Coordinate System in the New Tag dialog box, and then configure it. You can make your configuration selections from the Coordinate System Properties dialog box.

You can also use the Coordinate System Properties dialog boxes to edit an existing Coordinate System tag. These dialog boxes have a series of tabs that access a specific dialog box for configuring the different facets of the Coordinate System. Make the appropriate entries for each of the fields. An asterisk appears on the tab to indicate that changes have been made but not implemented. Click Apply to save your selections.

**TIP** When you configure your coordinate system, some fields can be unavailable (dimmed) because of choices you made in the New Tag dialog box.

In the Controller Organizer, right-click the coordinate system to edit and choose Coordinate System Properties from the pull-down menu.

The Coordinate System Properties General dialog box appears.

The name of the Coordinate System tag that is being edited appears in the title bar to the right of Coordinate System Properties.
General Tab

Use this tab to do the following for a coordinate system:

- Assign the coordinate system, or terminate the assignment of a coordinate system, to a Motion Group.
- Choose the type of coordinate system you are configuring.
- Change the number of dimensions, that is, the number of axes.
- Specify the number of axes to transform.
- Assign axes to the coordinate system tag.
- Enable/Disable automatic updating of the tag.

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

Table 3 - General Tab Field Descriptions

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion Group</td>
<td>Motion Group is where you can select and display the Motion Group to which the Coordinate System is associated. A Coordinate System that is assigned to a Motion Group appears in the Motion Groups branch of the Controller Organizer, under the selected Motion Group subbranch. If you select &lt;none&gt;, it terminates the Motion Group association, and moves the coordinate system to the Ungrouped Axes subbranch of the Motions Groups branch.</td>
</tr>
<tr>
<td>Ellipsis (…)</td>
<td>Ellipsis opens the Motion Group Properties dialog box for the Assigned Motion Group where you can edit the Motion Group properties. If no Motion Group is assigned to this coordinate system, this dialog box is unavailable.</td>
</tr>
<tr>
<td>New Group</td>
<td>New Group opens the New Tag dialog box where you can create a Motion Group tag. This option is enabled only if no Motion Group tag has been created.</td>
</tr>
<tr>
<td>Type</td>
<td>Type selects and displays the type of coordinate system (robot type) in the Motion Group. Available choices are Cartesian, Articulated Dependent, Articulated Independent, SCARA Independent, Delta, and SCARA Delta. The type of coordinate system you choose in this field changes the configuration tabs that are available.</td>
</tr>
<tr>
<td>Dimension</td>
<td>Enter the coordinate system dimensions, that is, the number of axes, that this coordinated system is to support. The options are 1, 2, or 3 to keep with its support of a maximum of three axes. Changes in the Dimension spin also reflect in the Axis Grid by either expanding or contracting the number of fields available. Data is set back to the defaults for any axis that is removed from the Axis Grid due to reducing the Dimension field.</td>
</tr>
<tr>
<td>Transform Dimension</td>
<td>Enter the number of axes in the coordinate system that you want to transform. The options are 1, 2, or 3 to keep with its support of a maximum of three axes. The number of axes that you transform must be equal to or less than the specified coordinate system dimensions. The transform function always begins at the first axis. For example, if you have specified that the coordinate system has three axes, but indicate only that two axes be transformed, then axes 1 and 2 are transformed. In other words, you cannot specify that only axes number 2 and number 3 are to be transformed.</td>
</tr>
<tr>
<td>Axis Grid</td>
<td>The Axis Grid is where you associate axes to the Coordinate System. There are five columns in the Axis Grid that provide information about the axes in relation to the Coordinate System.</td>
</tr>
<tr>
<td>[] (Brackets)</td>
<td>The Brackets column displays the indices in tag arrays used with the current coordinate system. The tag arrays that are used in multi-axis coordinated motion instructions map to axes by using these indices.</td>
</tr>
<tr>
<td>Coordinate</td>
<td>The text in this column X1, X2, or X3 (depending on the entry to the Dimension field) is used as a cross-reference to the axes in the grid. For a Cartesian system, the mapping is simple.</td>
</tr>
</tbody>
</table>
### Chapter 1
Create and Configure a Coordinate System

#### Table 3 - General Tab Field Descriptions

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Axis Name**                 | The Axis Name column is a list of combo boxes (the Dimension field determines this number) used to assign axes to the coordinate system.  
|                               | The pull-down lists display all Base Tag axes defined in the project. (Alias Tag axes do not display in the pull-down list.) They can be axes that are associated with the motion group, axes associated with other coordinated systems, or axes from the Ungrouped Axes folder. Choose an axis from the pull-down list. The default is <none>. It is possible to assign fewer axes to the coordinate system than the maximum for the Dimension field. However, you receive a warning when you verify the coordinate system and, if left in that state, the instruction generates a runtime error. You can assign an axis only once in a coordinate system. Ungrouped axes also generate a runtime error. |
| **Ellipsis (...)**            | The Ellipsis in this column takes you to the Axis Properties pages for the axis listed in the row.                                               |
| **Coordination Mode**         | The Coordination Mode column indicates the axes that are used in the velocity vector calculations. If the type of coordinate system is specified as Cartesian, then Primary axes are used in these calculations. For non-Cartesian coordinate systems, the coordination mode for the axes defaults to Ancillary. |
| **Enable Coordinate System Auto Tag Update** | The Enable Coordinate System Auto Tag Update checkbox lets you determine whether the Actual Position values of the current coordinated system are automatically updated during operation. Use the checkbox to enable this feature. The Coordinate System Auto Tag Update feature can ease your programming burden if you must add GSV statements to the program to get the desired result. However, by enabling this feature, the Coarse Update rate is increased. Whether to use the Coordinate System Auto Tag Update feature depends upon the trade-offs between ease in programming and increase in execution time. You can lower the execution time if you enable this feature in initial system programming to formulate the process and then disable it and enter the GSV statements in your program. If you enable this feature, it can result in some performance penalty. |
Geometry Tab

The Geometry tab of the Coordinate System Properties is where you can specify the link lengths and zero angle orientation values for articulated robotic arms.

![Geometry Tab](image)

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

**Link Lengths Box**

The Link Length box displays fields to let you specify a value for the length of each link in an articulated robotic arm (coordinate system). The measurement units configured for the affiliated Cartesian coordinate system defines the measurement units for the articulated 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 by using the same measurement units as the linked Cartesian coordinate system. For example, the manufacturer specifies the robot link lengths by using millimeter units and you want to configure the robot by using inches. You must convert the millimeter link measurements to inches and enter the values in the appropriate link length fields.

**IMPORTANT** Be sure that the link lengths that are specified for an articulated coordinate system are in the same measurement units as the affiliated Cartesian coordinate system. Your system does not work properly if you are using different measurement units.
The number of fields available for configuration in the link lengths box is determined by the combination of the following:

- Values that are entered on the General tab for the type of coordinate system
- Total coordinate system dimensions
- Transform dimensions

The link identifiers are L1 and L2 in the corresponding graphic. These fields are not configurable for a Cartesian coordinate system.

**Zero Angle Orientations Box**

The zero-angle orientation is the rotational offset of the individual joint axes. If applicable, enter the offset value in degrees for each joint axis. The coordinate dimension value entered on the General tab determines the number of available fields. The angle identifiers are Z1, Z2, and Z3 in the corresponding graphic.

**Units Tab**

The Units tab of the Coordinate System Properties is where you determine the units that define the coordinate system. This dialog box is where you define the Coordination Units and the Conversion Ratios.

**Coordination Units**

The Coordination Units field lets you define the units to be 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. Enter units that are relevant to your application and maximize ease of use. When you change the Coordination Units, the second portion of the Coordination Ratio Units automatically changes to reflect the new units. Coordination Units is the default.
Axis Grid

The Axis Grid of the Units dialog box displays the axis names that are associated with the coordinate system, the conversion ratio, and the units that are used to measure the conversion ratio.

Table 4 - Units Tab Description

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Axis Name</td>
<td>The Axis Name column contains the names of the axes assigned to the coordinate system in the General dialog box. These names appear in the order that they were configured into the current coordinate system. You cannot edit this column from this dialog box.</td>
</tr>
<tr>
<td>Conversion Ratio</td>
<td>The Conversion Ratio column defines the relationship of axis position units to coordination units for each axis. For example, there is a program the position units for an axis are in millimeters and the axis is associated with a coordinate system whose units are in inches. 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. The numerator can be entered as a float or an integer. The denominator must be entered only as an integer.</td>
</tr>
<tr>
<td>Conversion Ratio Units</td>
<td>The Conversion Ratio Units column displays the axis position units to coordination units used. The Axis Position units are defined in the Axis Properties – Units dialog box and the coordination units are defined in Coordinated System Properties – Units dialog box. These values are dynamically updated when changes are made to either axis position units or coordination units.</td>
</tr>
</tbody>
</table>

Offsets Tab

The Offsets tab of the Coordinate System Properties dialog box is where you define the end effector and base offset values for the robotic arm. This tab shows the top and/or sides view of a typical robotic arm, based on the type of coordinate system and coordinate Transform dimension values specified on the General tab. The number of axes associated with the coordinate system determines the number of available offset fields in each box.
When specifying the end effector and base offset values, be sure that the values are calculated by using the same measurement units as the linked Cartesian coordinate system.

For example, the manufacturer specifies the robot offset by using millimeter units and you want to configure the robot by using inches. You must convert the millimeter link measurements to inches and enter the values in the appropriate offset fields.

**End Effector Offsets Box**

The end effector offset value specifies the dimensions of the end effector. The correct end effector offsets are typically available from the manufacturer. The end effector indicators are X1e, X2e, and X3e in the corresponding graphic.

**Base Offsets Box**

The Logix Designer kinematics internal equations define the robot origin relative to the first joint of the robotic arm. Sometimes the robot manufacturer specifies the origin at another location. The difference between these two locations is the base offsets value. The correct base offset values are typically available from the robot manufacturer. The base offset indicators are X1b, X2b, and X3b in the corresponding graphic.

**Joints Tab**

The Joints tab is accessible only if you are configuring or editing an articulated coordinate system. This dialog box is where you define the Joint Conversion Ratios. Joint axis units are always specified in degrees.

**Table 5 - Joints Tab Field Descriptions**

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Axis Name</td>
<td>The Axis Name column displays the names of the axes associated to the coordinate system. The names appear in the order that they were configured into the coordinate system. This field is a read-only field.</td>
</tr>
<tr>
<td>Joint Ratio</td>
<td>The Joint Ratio column (shown in white) is divided into two columns that define the relationship between the axis position units to the joint axis units. The left-half of the Joint Ratio column is a configurable field that lets you specify a value for the axis position units (numerator). The right-half of the Joint Ratio column is a configurable field that lets you specify a value for the joint axis units (denominator). Keep in mind that Joint axis units are always specified as degrees.</td>
</tr>
<tr>
<td>Joint Units</td>
<td>The Joint Units column is a read-only field that displays the configured axis position units to the joint units. The Axis Position units are defined in the Axis Properties – Units dialog box. Joint units are always defined as degrees.</td>
</tr>
</tbody>
</table>
If you are configuring a Cartesian coordinate system, go to the Dynamics tab to access the Coordinate System Properties Dynamics dialog box.

![Coordinate System Properties - joint_coordinate_system](image)

**Dynamics Tab**

The Dynamics dialog box is accessible only if you are configuring a Cartesian coordinate system. The Dynamics tab is for entering the Vector values used for Maximum Speed, Maximum Acceleration, Maximum Deceleration, Maximum Acceleration Jerk and Maximum Deceleration Jerk. It is also used for entering the Actual and Command Position Tolerance values.

![Coordinate System Properties - my_coordinate_system](image)

**Vector Box**

In the Vector box, values are entered for Maximum Speed, Maximum Acceleration, Maximum Deceleration, Maximum Acceleration Jerk, and Maximum Deceleration Jerk. The values are used by the Coordinated Motion instructions in calculations when their operands are expressed as percent of Maximum. The Coordination Units to the right of the edit boxes automatically change when the coordination units are redefined in the Units dialog box.
### Table 6 - Dynamics Tab Field Descriptions

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Speed</td>
<td>Enter the value for Maximum Speed to be used by the Coordinated Motion instructions in calculating vector speed when speed is expressed as a percent of maximum.</td>
</tr>
<tr>
<td>Maximum Acceleration</td>
<td>Enter the value for Maximum Acceleration to be used by the Coordinated Motion instructions. This value determines the acceleration rate to apply to the coordinate system vector when acceleration is expressed as a percent of maximum.</td>
</tr>
<tr>
<td>Maximum Deceleration</td>
<td>Enter the value for Maximum Deceleration to be 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 nonzero value to achieve any motion by using the coordinate system.</td>
</tr>
<tr>
<td>Maximum Acceleration Jerk</td>
<td>The jerk parameters only apply to S-Curve profile moves by using these instructions:</td>
</tr>
<tr>
<td></td>
<td>• MCS</td>
</tr>
<tr>
<td></td>
<td>• MCCC</td>
</tr>
<tr>
<td></td>
<td>• MCCM</td>
</tr>
<tr>
<td></td>
<td>• MCLM</td>
</tr>
<tr>
<td></td>
<td>The Maximum Acceleration Jerk rate of the coordinate system, in Coordination Units/second^2, defaults to 100% of the maximum acceleration time. The speed and acceleration rate for this calculation are defined previously.</td>
</tr>
</tbody>
</table>
|                             | \[
|                             |   \frac{\text{MaxAccel}}{\text{Speed}} = \text{Maximum Acceleration Jerk}\]
|                             | The Maximum Accel Jerk value that you enter 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^2, 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 has Jerk limiting. This is the default mode. An S-Curve move with 0% of Time results in a trapezoidal profile and have 0% Jerk limiting. If set manually, enter the value in units=Coordination Units/second^2 units. You can also use the Calculate button to view this value in terms of units=% of Time. |
| Maximum Deceleration Jerk   | The jerk parameters only apply to S-Curve profile moves by using these instructions:                                                                                                                         |
|                             | • MCS                                                                                                                                            |
|                             | • MCCC                                                                                                                                          |
|                             | • MCCM                                                                                                                                          |
|                             | • MCLM                                                                                                                                          |
|                             | The Maximum Deceleration Jerk rate of the coordinate system, in Coordination Units/second^2, defaults to 100% of the maximum deceleration time. The speed and deceleration rate for the calculation are defined previously. |
|                             | \[
|                             |   \frac{\text{MaxDecel}}{\text{Speed}} = \text{Maximum Deceleration Jerk}\]
|                             | The Maximum Decel Jerk value that you enter 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^2, 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 has Jerk limiting. Thus, the default mode. An S-Curve move with 0% of Time results in a trapezoidal profile and has 0% Jerk limiting. If set manually, enter the value in units=Coordination Units/second^2 units. You can also use the optional Calculate button to view the value in terms of units=% of Time. |
**Position Tolerance Box**

In the Position Tolerance Box, values are entered for Actual and Command Position Tolerance values. See the Logix5000™ Motion Controllers Instructions Reference Manual, publication MOTION-RM002, for more information regarding the use of Actual and Command Position Tolerance.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual</td>
<td>Enter the value in coordination units for Actual Position to be used by Coordinated Motion instructions when they have a Termination Type of Actual Tolerance.</td>
</tr>
<tr>
<td>Command</td>
<td>Enter the value in coordination units for Command Position to be used by Coordinated Motion instructions when they have a Termination Type of Command Tolerance.</td>
</tr>
</tbody>
</table>

**Manual Adjust Button**

The Manual Adjust button on the Coordinate System Dynamics tab accesses the Manual Adjust Properties dialog box. The Manual Adjust button is enabled only when there are no pending edits on the properties dialog box.

**Dynamics Tab Manual Adjust**

At this dialog box, you can change the Vector and Position Tolerance values.

![Dynamics Tab Manual Adjust](image)

These changes can be made either online or offline. The blue arrows to the right of the fields indicate that they are immediate commit fields. This indication means that the values in those fields are immediately updated to the controller if online or to the project file if offline.

**Reset**

Reset reloads the values that were present at the time this dialog box was entered. The blue arrow to the right of Reset means that the values are immediately reset when you click Reset.
Motion Planner Tab

The Motion Planner dialog box is accessible only if you are configuring a Cartesian coordinate system. The Motion Planner tab is used to enable or disable Master Delay Compensation, enable or disable Master Position Filter, and to enter the bandwidth for Master Position Filter.

Table 7 - Motion Planner Tab Field Descriptions

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master Delay Compensation</td>
<td>Check or clear this box to enable or disable Master Delay Compensation, respectively. This value is used to balance the delay time between reading the Master Axis command position and applying the associated slave command position to the servo loop of the slave. This feature makes sure 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 this box to disable Master Delay Compensation. The default setting is Enabled. 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 can be beneficial to disable the Master Delay Compensation feature to eliminate the disturbances introduced to the Slave Axis. Master Delay Compensation, even if the box is checked, is not applied in cases where a Slave Axis is gearing or camming to the command position of the Master Axis because there is no need to compensate for master position delay.</td>
</tr>
<tr>
<td>Enable Master Position Filter</td>
<td>Check or clear this box to enable or disable Master Position Filter, respectively. The default is cleared (disabled). Master Position Filter, when enabled, effectively 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. When this box is checked, the Master Position Filter Bandwidth box is enabled.</td>
</tr>
<tr>
<td>Master Position Filter Bandwidth</td>
<td>The Master Position Filter Bandwidth field is enabled when the Enable Master Position Filter checkbox is checked. This field controls the bandwidth for master position filtering. Enter a value in Hz in this field to set the bandwidth for the Master Position Filter. A value of zero for Master Position Filter Bandwidth effectively disables the master position filtering.</td>
</tr>
</tbody>
</table>
Tag Tab

The Tag tab is for reviewing your Tag information and renaming the tag or editing the description.

Use this tab to modify the name and description of the coordinate system. When you are online, all parameters on this tab transition to a read-only state, and cannot be modified. If you go online before you save your changes, all pending changes revert to their previously saved state.

Table 8 - Tag Tab Field Descriptions

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name displays the name of the current tag. You can rename the tag now. The name can be up to 40 characters and can include letters, numbers, and underscores (_). When you rename a tag, the new name replaces the old one in the Controller Organizer after you click OK or Apply.</td>
</tr>
<tr>
<td>Description</td>
<td>Description displays the description of the current tag, if any is available. You can edit this description. The edited description replaces the existing description when you click OK or Apply.</td>
</tr>
<tr>
<td>Tag Type</td>
<td>Tag Type indicates the type of the current Coordinate System tag. This type can be either a base or an alias. The field is not editable and is for informational purposes only.</td>
</tr>
<tr>
<td>Data Type</td>
<td>Data Type displays the data type of the current Coordinate System tag, which is always COORDINATE_SYSTEM. This field cannot be edited and is for informational purposes only.</td>
</tr>
<tr>
<td>Scope</td>
<td>Scope displays the scope of the current Coordinate System tag. The scope for a Coordinate System tag can be only controller scope. This field is not editable and is for informational purposes only.</td>
</tr>
<tr>
<td>External Access</td>
<td>External Access displays the parameter that is chosen in the New Tag dialog box for whether the tag has Read/Write, Read Only, or no (None) access from external applications such as HMIs.</td>
</tr>
</tbody>
</table>
Notes:
Configure a Cartesian Coordinate System

Use the multi-axis coordinated motion instructions to perform linear and circular moves in single and multidimensional spaces. A Cartesian coordinate system in Logix Designer application can include one, two, or three axes.

Figure 3 - Coordinate Systems with Orthogonal Axes

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program an MCLM Instruction</td>
<td>32</td>
</tr>
<tr>
<td>Blended Moves and Termination Types</td>
<td>32</td>
</tr>
<tr>
<td>Bit State Diagrams for Blended Moves</td>
<td>35</td>
</tr>
<tr>
<td>Choose a Termination Type</td>
<td>39</td>
</tr>
</tbody>
</table>

Use the MCLM instruction to start a single or multi-dimensional linear coordinated move. See the Logix5000™ Motion Controllers Instructions Reference Manual, publication MOTION-RM002, for more information.

Use the MCCM instruction to initiate a two or three-dimensional circular coordinated move for the specified axes. See the Logix5000 Motion Controllers Instructions Reference Manual, publication MOTION-RM002, for more information.
## Program an MCLM Instruction

The following are the steps to program and test an MCLM instruction.

1. **Configure motion axes in Logix Designer application.**

   The maximum number of axes that can be associated with one Coordinate System is limited to three axes.

2. **Create a Coordinate System Tag**

   The number of Coordinate System tags that can be created is 32. This number is based on the fact that a maximum of 32 axes can be assigned to a motion group and in the current implementation. Because only one motion group can be created, the number of axes that can be created is 32.

3. **Program an MCLM.**

   The Motion Coordinated Linear Move (MCLM) instruction performs a linear move by using up to three axes in a Cartesian coordinate system. As with all moves, you must specify, for example, absolute or incremental, or speed. Speed is based on the vector move distance as shown in this figure.

   \[
   V = \sqrt{v_x^2 + v_y^2}
   \]

   One dimension array determines the position. The coordinate system selected determines array length. For a (2) Axis Cartesian System, each endpoint requires (2) words; for a (3) Axis Cartesian System, each axis requires (3) words. We create a position array shortly for clarification. An array can consist of multiple endpoint coordinates that multiple coordinated move instructions can use.

## Blended Moves and Termination Types

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

- MovePendingStatus
- MovePendingQueueFullStatus
The MCLM and MCCM instructions reference a coordinate system called Coordinate_System_1 (cs1). For example, the following ladder diagram uses coordinate system cs1 to blend Move1 into Move2.

**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, then the bits are set by using these parameters.

Table 9 - Bit Parameters

<table>
<thead>
<tr>
<th>When</th>
<th>Then</th>
</tr>
</thead>
</table>
| 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

For further information on how to select a termination type, refer to Choose a Termination Type on page 39.
Bit State Diagrams for Blended Moves

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

**Bit States at Transition Points of Blended Move by Using Actual Tolerance or No Settle**

![Diagram showing bit states at transition points]

<table>
<thead>
<tr>
<th>Table 10</th>
<th>Bit Status at Transition Points with Actual Tolerance or No Settle Termination Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit</td>
<td>TP1</td>
</tr>
<tr>
<td>Move1.DN</td>
<td>T</td>
</tr>
<tr>
<td>Move1.IP</td>
<td>T</td>
</tr>
<tr>
<td>Move1.AC</td>
<td>T</td>
</tr>
<tr>
<td>Move1.PC</td>
<td>F</td>
</tr>
<tr>
<td>Move2.DN</td>
<td>T</td>
</tr>
<tr>
<td>Move2.IP</td>
<td>T</td>
</tr>
<tr>
<td>Move2.AC</td>
<td>F</td>
</tr>
<tr>
<td>Move2.PC</td>
<td>F</td>
</tr>
<tr>
<td>cs1.MoveTransitionStatus</td>
<td>F</td>
</tr>
<tr>
<td>cs1.MovePendingStatus</td>
<td>T</td>
</tr>
<tr>
<td>cs1.MovePendingQueueFullStatus</td>
<td>T</td>
</tr>
</tbody>
</table>
Bit States at Transition Points of Blended Move by Using No Decel

Table 11 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 Move 1 instruction. If Move 1 and Move 2 are collinear, then Move 1.PC is true at TP3 (the programmed end point of first move).

Table 11 - Bit Status with No Decel Termination Type

<table>
<thead>
<tr>
<th>Bit</th>
<th>TP1</th>
<th>TP2</th>
<th>TP3</th>
<th>TP4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Move1.DN</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Move1.IP</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>Move1.AC</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>Move1.PC</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>Move2.DN</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Move2.IP</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>Move2.AC</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>Move2.PC</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>cs1.MoveTransitionStatus</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>cs1.MovePendingStatus</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>cs1.MovePendingQueueFullStatus</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>
**Bit States at Transition Points of Blended Move by Using Command Tolerance**

Table 12 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.

### Table 12 - Bit Status with Command Tolerance Termination Type

<table>
<thead>
<tr>
<th>Bit</th>
<th>TP1</th>
<th>TP2</th>
<th>TP3</th>
<th>TP4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Move1.DN</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Move1.IP</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>Move1.AC</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>Move1.PC</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Move2.DN</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Move2.IP</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>Move2.AC</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>Move2.PC</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>cs1.MoveTransitionStatus</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>cs1.MovePendingStatus</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>cs1.MovePendingQueueFullStatus</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>
Bit States at Transition Points of Blended Move by Using Follow Contour Velocity Constrained or Unconstrained

Table 13 shows the bits status at the transition points.

Table 13 - Bit Status with Contour Velocity Constrained or Unconstrained Termination Type

<table>
<thead>
<tr>
<th>Bit</th>
<th>TP1</th>
<th>TP2</th>
<th>TP3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Move1.DN</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Move1.IP</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>Move1.AC</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>Move1.PC</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Move2.DN</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Move2.IP</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>Move2.AC</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>Move2.PC</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>cs1.MoveTransitionStatus</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>cs1.MovePendingStatus</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>cs1.MovePendingQueueFullStatus</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>
### 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.

1. Choose a termination type.

<table>
<thead>
<tr>
<th>If You Want the Axes to (vector speeds)</th>
<th>And You Want the Instruction to Complete When the</th>
<th>Then Use this Termination Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop between moves</td>
<td>Both of these events occur:</td>
<td>0 - Actual Tolerance</td>
</tr>
<tr>
<td></td>
<td>• Command position equals target position.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• The vector distance between the target and actual positions is less than or equal to the Actual Position Tolerance of the coordinate system.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Command position equals the target position.</td>
<td>1 - No Settle</td>
</tr>
<tr>
<td>Keep the speed constant except between moves</td>
<td>Command position gets within the Command Position Tolerance of the coordinate system.</td>
<td>2 - Command Tolerance</td>
</tr>
<tr>
<td></td>
<td>Axes get to the point at which they must decelerate at the deceleration rate.</td>
<td>3 - No Decel</td>
</tr>
<tr>
<td>Transition into or out of a circle without stopping</td>
<td>Command position gets within the Command Position Tolerance of the coordinate system.</td>
<td>4 - Follow Contour Velocity Constrained</td>
</tr>
<tr>
<td>Accelerate or decelerate across multiple moves</td>
<td>The command position gets within the Command Position Tolerance of the coordinate system.</td>
<td>5 - Follow Contour Velocity Unconstrained</td>
</tr>
<tr>
<td>Use a specified Command Tolerance</td>
<td></td>
<td>6 - Command Tolerance Programmed</td>
</tr>
</tbody>
</table>
2. Make sure this selection is the right choice for you.

<table>
<thead>
<tr>
<th>Termination Type</th>
<th>Example Path</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0 - Actual Tolerance     | ![Graph](image1) | The instruction stays active until both of these events occur:  
• 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            | ![Graph](image2) | 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 | ![Graph](image3) | 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. |

<table>
<thead>
<tr>
<th>Logix Designer Application Compares</th>
<th>To the</th>
<th>And Uses the</th>
<th>For the</th>
</tr>
</thead>
<tbody>
<tr>
<td>100% of the configured length of the first instruction by using a Command Tolerance termination type</td>
<td>Configured Command Tolerance for the coordinate system</td>
<td>Shorter of the two lengths</td>
<td>Command Tolerance length used for the first instruction</td>
</tr>
<tr>
<td>100% of the configured length of the last move instruction by using a Command Tolerance termination type</td>
<td>Configured Command Tolerance for the coordinate system</td>
<td>Shorter of the two lengths</td>
<td>Command Tolerance length used for the next to last instruction</td>
</tr>
<tr>
<td>50% of each of the lengths of all other move instructions</td>
<td>Configured Command Tolerance for the coordinate system</td>
<td>Shorter of the two lengths</td>
<td>Command Tolerance length used for each individual instruction</td>
</tr>
</tbody>
</table>

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.
**Important Considerations**

If you stop a move by using an MCS or by changing the speed to zero with an MCCD during a blend and then resume the move by reprogramming the move or by using another MCCD, it deviates from the path that displayed 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 is most likely a slight deviation.

**Velocity Profiles for Collinear Moves**

Collinear moves are moves 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**

Figure 4 shows the velocity profile of two collinear moves by 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.
Configure a Cartesian Coordinate System

Chapter 2

Configure a Cartesian Coordinate System

**Figure 4 - 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**

The .PC bit is set, MCLM1 is over

Command Tolerance Point

MCLM1

MCLM2

Programmed endpoint of MCLM1 instruction

Position

**Figure 5** shows the velocity profile of two collinear moves by 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.

**Figure 5 - 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**

The .PC bit is set, MCLM1 is Over

MCLM1

MCLM2

Programmed Endpoint of MCLM1 instruction

Position

**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.
Configure a Cartesian Coordinate System

Chapter 2

Figure 6 - 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.

Figure 7 - 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 if they are programmed symmetrically.
To make sure that your trajectory is symmetric, you must terminate any sequence of moves by either Termination Types 0 or 1. 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 or TT6 as the last move in a profile (or the reversal point) is safe. However, the resulting trajectory from A to B cannot 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 make sure of a symmetric profile.
**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.

![Triangular Velocity Profile Diagram]

Then, use termination type 5. The other termination types can prevent you from getting to the speed you want.

<table>
<thead>
<tr>
<th>Termination Types 2, 3, 4, or 6</th>
<th>Termination Type 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>You want to get to this speed.</td>
<td>You calculate the acceleration.</td>
</tr>
<tr>
<td>But the axes have to decelerate before they get there.</td>
<td>And you must also calculate the starting speed for each move during deceleration.</td>
</tr>
<tr>
<td>The length of each move determines its maximum speed. As a result, the axes do not reach a speed that causes them to overshoot the target position during deceleration.</td>
<td>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.</td>
</tr>
</tbody>
</table>
## Blending Moves at Different Speeds

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

<table>
<thead>
<tr>
<th>If the Next Move is</th>
<th>And the Termination Type of the First Move is</th>
<th>Then</th>
</tr>
</thead>
</table>
| Slower              | 2 - Command Tolerance  
3 - No Decel  
4 - Contour Velocity Constrained  
5 - Contour Velocity Unconstrained  
6 - Command Tolerance Programmed | Vector speed  
Target position of first move |
| Faster              | 2 - Command Tolerance  
3 - No Decel  
6 - Command Tolerance Programmed | Vector speed  
Target position of first move |
|                      | 4 - Contour Velocity Constrained  
5 - Contour Velocity Unconstrained | Vector speed  
Target position of first move |
Configure Kinematics Coordinate Systems

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Useful Terms</td>
<td>48</td>
</tr>
<tr>
<td>Gather Information about Your Robot</td>
<td>48</td>
</tr>
<tr>
<td>Summary of Kinematic Steps</td>
<td>48</td>
</tr>
<tr>
<td>Determine the Coordinate System Type</td>
<td>50</td>
</tr>
</tbody>
</table>

This chapter provides you with the information you need when using the Kinematics functionality within Logix Designer application. This chapter also provides you with guidelines for robot-specific applications.

Kinematics coordinate systems use two instructions, the Motion Calculate Transform Position (MCTP) and the Motion Coordinated Shutdown Reset (MCSR).

**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.

**Motion Coordinated Shutdown Reset (MCSR)**

Use the Motion Coordinated Shutdown Reset (MCSR) instruction to reset all axes in a coordinate system. The MCSR instruction resets the axes from a shutdown state to an axis ready state. This instruction also clears any axis faults.

**ATTENTION:** Use each tag for the motion control attribute of instructions only once. Re-use of the motion control tag in other instructions can cause unintended operation. This can result in damage to equipment or personal injury.
**Useful Terms**

Understanding the terms used in this chapter enables you to properly configure your robot.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forward Kinematics</td>
<td>The solution of source positions given the target positions. In practice, requires computing the Cartesian positions given the Joint positions.</td>
</tr>
<tr>
<td>Forward Transform</td>
<td>The solution of source positions given target positions.</td>
</tr>
<tr>
<td>Inverse Kinematics</td>
<td>The solution of joint positions given Cartesian positions. Typically, converts Cartesian positions to joint positions.</td>
</tr>
<tr>
<td>Inverse Transform</td>
<td>The solution of target positions given source positions.</td>
</tr>
<tr>
<td>Joint axis</td>
<td>A rotary robotic coordinate axis typically having overtravel rather than rollover limits.</td>
</tr>
<tr>
<td>Kinematics</td>
<td>The family of mathematical equations that convert positions back and forth between two linked geometries.</td>
</tr>
<tr>
<td>Orientation</td>
<td>Robotic term for directional attitude or rotation about a point in Cartesian (3D) space. Orientation is expressed as three ordered rotations</td>
</tr>
<tr>
<td></td>
<td>around the X, Y, and Z Cartesian axes.</td>
</tr>
<tr>
<td>Reference frame</td>
<td>An imaginary Cartesian coordinate system used to define a Cartesian origin and reference orientation.</td>
</tr>
<tr>
<td>Source system</td>
<td>One of two coordinate systems used in a Kinematics transform and having special properties. When connected to a target system by means of</td>
</tr>
<tr>
<td></td>
<td>a Kinematics transform, motion commanded at the source system's inputs produces motion at both the source and target system's outputs</td>
</tr>
<tr>
<td></td>
<td>(if the physical axes are connected).</td>
</tr>
<tr>
<td>Target system</td>
<td>One of two coordinate systems used in a Kinematics transform and having special properties. When connected to a source system by means of</td>
</tr>
<tr>
<td></td>
<td>a Kinematics transform, motion commanded at the target system's inputs produces motion in both the source and target system's outputs</td>
</tr>
<tr>
<td></td>
<td>(if the physical axes are connected).</td>
</tr>
<tr>
<td>Tool Center Point</td>
<td>All Kinematics programmed position (motion) is based on the Tool Center Point (TCP).</td>
</tr>
<tr>
<td></td>
<td>To determine the TCP, you must enter information on these Logix Designer application tabs:</td>
</tr>
<tr>
<td></td>
<td>• Geometry - Enter values for Link Length (linear displacement), Zero Angle Orientation (angular rotation), and Base Offsets. These values, in</td>
</tr>
<tr>
<td></td>
<td>combination with the selected Geometry type, defines the resulting Geometry’s end-of-arm position.</td>
</tr>
<tr>
<td></td>
<td>• Offsets - Enter value for End-effector offset; these are included when establishing the final TCP position.</td>
</tr>
<tr>
<td>Transform</td>
<td>General term for conversion equations that map values in one coordinate space to values in another coordinate space.</td>
</tr>
<tr>
<td>Translation</td>
<td>Robotic term for a linear movement or offset in Cartesian (three-dimensional) space. Translation describes the distance between two</td>
</tr>
<tr>
<td></td>
<td>Cartesian points.</td>
</tr>
<tr>
<td>Zero Angle Offset</td>
<td>Offset on a rotary axis in the Joint Coordinate system between where the Kinematics equations were derived and where you want your zero</td>
</tr>
<tr>
<td></td>
<td>position to be.</td>
</tr>
</tbody>
</table>

**Gather Information about Your Robot**

Before you begin the configuration steps for the Kinematics transformation function, you need to gather specific information about your robot and application parameters. Specifications for your robot can be found in the documentation provided by the manufacturer; other required information is application dependent. You need to know this information before you begin configuring motion control.

- Robot geometry type
- Zero angle orientation
- Work envelope
- Link lengths
- Base offsets
- End-effector offsets
- Arm solution

**Summary of Kinematic Steps**

After you create a Joint (target) coordinate system tag for your Motion control project, there are general steps to follow for Kinematics.
Configure Kinematics Coordinate Systems

Chapter 3

1. Determine and then configure the type of coordinate system you need for your robot.

For help in determining your coordinate system type, see page 50.

2. Establish the Joint-to-Cartesian reference frame relationship.

For more information regarding the joint-to-Cartesian reference frame, see the section about the type of robot you are using.

**ATTENTION:** The correct relationship between the Joint reference frame and the Cartesian reference frame must be established. Failure to do this can allow your robot to move to unexpected positions causing machine damage and/or injury or death to personnel.

3. Calibrate your robot (if applicable).

4. Identify your robot work envelope.

5. Determine and then configure the following parameters:
   - Link lengths
   - Base offsets
   - End-effector offsets

6. Create the source and target coordinate systems.

7. Save the project.

8. Download the Kinematic project to the controller and then use the MCT instruction to link the Joint coordinate system to the Cartesian coordinate system.

The Joint-to-Cartesian reference frame relationship is automatically established by the controller after the Joint coordinate system
parameters (link lengths, base offsets, and end-effector offsets) are configured and the MCT instruction is enabled. For additional information about the MCT or MCTP instructions, see the Logix5000™ Controllers Motion Instructions, publication MOTION-RM002.

For detailed steps about Creating and Configuring a Coordinate System, see on Create and Configure a Coordinate System page 13.

**Determine the Coordinate System Type**

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

<table>
<thead>
<tr>
<th>If your robot looks similar to</th>
<th>Your Coordinate System type is</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Articulated Independent" /></td>
<td>Articulated Independent&lt;br&gt;For configuration information, see page 53.</td>
</tr>
<tr>
<td><img src="image" alt="Articulated Dependent" /></td>
<td>Articulated Dependent&lt;br&gt;For configuration information, see page 91.</td>
</tr>
<tr>
<td><img src="image" alt="Cartesian" /></td>
<td>Cartesian&lt;br&gt;This illustration shows a typical Gantry machine.&lt;br&gt;For configuration information, see page 101.</td>
</tr>
<tr>
<td>If your robot looks similar to</td>
<td>Your Coordinate System type is</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td><img src="image" alt="H-bot diagram" /></td>
<td>Cartesian</td>
</tr>
<tr>
<td></td>
<td>This illustration shows a typical H-bot.</td>
</tr>
<tr>
<td></td>
<td>For configuration information see page 103.</td>
</tr>
<tr>
<td><img src="image" alt="SCARA Independent diagram" /></td>
<td>SCARA Independent</td>
</tr>
<tr>
<td></td>
<td>For configuration information, see page 84.</td>
</tr>
</tbody>
</table>
### Chapter 3  Configure Kinematics Coordinate Systems

<table>
<thead>
<tr>
<th>If your robot looks similar to</th>
<th>Your Coordinate System type is</th>
</tr>
</thead>
</table>
| ![Three-dimensional Delta](image1) | Three-dimensional Delta  
For configuration information, see page 62. |
| ![Two-dimensional Delta](image2) | Two-dimensional Delta  
For configuration information, see page 71. |
| ![SCARA Delta](image3) | SCARA Delta  
For configuration information, see page 76. |
Configure an Articulated Independent Robot

Use these guidelines when configuring an Articulated Independent robot.

ATTENTION: 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.
1. Set and enable the soft travel limits.
2. 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.

ATTENTION: 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.

Reference Frame

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

The reference frame for an Articulated Independent robot is at the base of the robot, as shown in Figure 9.
Before you begin establishing the Joint-to-Cartesian reference frame relationship, it is important to know some information about the Kinematic mathematical equations used in the controllers. The equations were written as if the Articulated Independent robot joints were positioned as shown in Figure 10.

- +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 your robot is physically in this position, the Logix Designer application Actual Position tags for the axes must be:
- J1 = 0
- J2 = 0
- J3 = 0
When your robot is physically in this position, the Logix Designer application Actual Position tags for the axes must be:

- J1 = 0
- J2 = 90°
- J3 = -90°

Figure 11 - Articulated Independent 3

If your robot’s physical position and joint angle values cannot match those shown in either figures above, then use one of the Alternate Methods for Establishing the Joint-to-Cartesian reference frame relationship.

Methods to Establish a Reference Frame

The following methods let you establish a reference frame for an Articulated Independent robot.

<table>
<thead>
<tr>
<th>For each</th>
<th>Use one of these methods to establish the reference frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incremental axis</td>
<td>Each time the robot’s power is cycled.</td>
</tr>
<tr>
<td>Absolute axis</td>
<td>Only when you establish absolute home.</td>
</tr>
</tbody>
</table>

- Method 1 - establishes a Zero Angle Orientation and lets the configured travel limits and home position on the joint axes remain operational. Use this method if you are 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 a MRP instruction to redefine the axes position to align with the Joint reference frame. This method can require the soft travel limits to be adjusted to the new reference frame.
Method 1 - Establishing a Reference Frame

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 as shown in Articulated Independent 3 on page 55, the Logix Designer application Actual Position tag values are equal to:

- J1 = 0
- J2 = 90°
- J3 = -90°

If the Logix Designer application Positions tags do not correspond to these values, configure the Zero Angle Orientation for the joint or joints that do not correspond.

<table>
<thead>
<tr>
<th>If the Logix Designer application read-out values are</th>
<th>Set the Zero Angle Orientations on the Coordinate System Properties dialog to</th>
</tr>
</thead>
<tbody>
<tr>
<td>J1 = 10, J2 = 80, J3 = -85</td>
<td>Z1 = -10, Z2 = 10, Z3 = -5</td>
</tr>
</tbody>
</table>

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

Figure 12 - Setting the Zero Angle Orientations
Method 2 - Establishing a Reference Frame

Position the robot so that:

- Link1 is parallel to the X3 axis.
- Link2 is parallel to X1 axis.

Program a 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 ControlLogix controller after the Joint coordinate system parameters (link lengths, base offsets, and end-effector offsets) are configured and the MCT instruction is enabled.

Work Envelope

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 having 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 is not always a complete sphere.
Chapter 4  Configure an Articulated Independent Robot

If the range-of-motion values for the articulated robot are

<table>
<thead>
<tr>
<th>J1</th>
<th>J2</th>
<th>J3</th>
<th>L1</th>
<th>L2</th>
</tr>
</thead>
<tbody>
<tr>
<td>±170</td>
<td>0 to 180</td>
<td>±100</td>
<td>10</td>
<td>12</td>
</tr>
</tbody>
</table>

Typically, the work envelope is

![Diagram of work envelope](image)

**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°.
Configuration Parameters

Logix Designer application can be configured for control of robots with varying reach and payload capacities. As a result, it is very important to know the configuration parameter values for your robot including:

- 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 entered into the Configuration Parameters dialog use the same measurement units.

Figure 13 illustrates the typical configuration parameters for an Articulated Independent robot.

**Figure 13 - Typical Configuration Parameters for an Articulated Independent Robot**

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

- X3e2 = 1.5 inches
- L2 = 12 inches
- X1e = 2 inches
- L1 = 12 inches
- X3b = 4.0 inches
- X1b = 3.0 inches
- X3e = -X3e1 + X3e2
- X3e = -3 + 1.5
- X3e = -1.5 inches

Robot Origin

Tool reference frame

X3

L2 = 12 inches

X1e = 2 inches

-X3e1 = 3.0 inches
Link Lengths

Link lengths are the rigid mechanical bodies attached at joints.

<table>
<thead>
<tr>
<th>For an articulated independent robot with</th>
<th>The length of</th>
<th>Is equal to the value of the distance between</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 dimensions</td>
<td>L1, L2</td>
<td>J1 and J2, J2 and the end-effector</td>
</tr>
<tr>
<td>3 dimensions</td>
<td>L1, L2</td>
<td>J2 and J3, J3 and the end-effector</td>
</tr>
</tbody>
</table>

**Figure 14 - Example of Link Lengths for an Articulated Independent Robot**

Enter the Link Length values.

For the robot shown in Typical Configuration Parameters for an Articulated Independent Robot, the Link Length values are:
- L1 = 10.0
- L2 = 12.0
**Base Offsets**

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. Enter the values for the base offsets in the X1b and X3b fields of the Coordinate System Properties dialog.

**Figure 15 - Example of Base Offsets for an Articulated Independent Robot**

Enter the Base Offset values. For the robot shown in our example, the Base Offset values are:
- \( X_{1b} = 3.0 \)
- \( X_{3b} = 4.0 \)
End-effector Offsets

The robot can have an end-effector attached to the end of robot link L2. If there is an attached end-effector, then you must configure the end-effector offset value on the Coordinate System Properties dialog. 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 as shown in Figure 13 on page 59. You can account for this value when computing the X3e end-effector offset value. In Figure 13, 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.

Delta Robot Geometries

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

Figure 17 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 Figure 17 is a three-degree of freedom robot with an optional fourth degree of freedom used to rotate a part at the tool tip. In Logix Designer application, the first three degrees of freedom are configured as three joint axes \(J_1, J_2, J_3\) in the robot's coordinate system. The three joint axes are either:

- directly programmed in joint space.
- automatically controlled by the embedded Kinematics software in 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 are identical in that they each have a single top link arm \(L_1\) and a parallelogram two-bar link assembly \(L_2\).

As each axis \(J_1, J_2, J_3\) is rotated, the TCP of the gripper moves correspondingly in \((X_1, X_2, X_3)\) direction. The gripper remains vertical along the \(X_3\) axis while its position is translated to \((X_1, X_2, X_3)\) space by the mechanical action of the parallelograms in each of the three forearm assemblies. The mechanical connections of the parallelograms via spherical joints ensure that the top and bottom plates remain parallel to each other.

You program the TCP to an \((X_1, X_2, X_3)\) coordinate, then Logix Designer application computes the commands necessary for each of the joints \((J_1, J_2, J_3)\) to move the gripper linearly from the current \((X_1, X_2, X_3)\) position to the programmed \((X_1, X_2, X_3)\) position, at the programmed vector dynamics.

When each top link \(L_1\) moves downward, its corresponding joint axis \((J_1, J_2,\) or \(J_3)\) 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 either a linear or rotary, independent axis.
Chapter 4 Configure an Articulated Independent Robot

Establish the Reference Frame for a Delta Three-dimensional Robot

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

Calibrate a Delta Three-dimensional Robot

Use these steps to calibrate your robot.

1. Obtain the angle values from the robot manufacturer for J1, J2, and J3 at the calibration position. These values are used to establish the reference position.

2. Move all joints to the calibration position by either 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 these:
   a. Use a Motion Redefine Position instruction (MRP) 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 of this procedure and execute a Motion Axis Home instruction (MAH) for each joint axis.

4. Move each joint to an absolute position of 0.0. Verify that each joint position reads $0^\circ$ and that the respective L1 is in a horizontal position. If L1 is not in a horizontal position, then see the alternate method for calibrating a Delta three-dimensional robot.
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, and then perform one of the following:
- Use a MRP instruction to set all the joint angles to 0° at this position.
- Configure the values for the Zero Angle Offsets to be equal to the values of the joints when 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 you must properly configure the Zero Angle Orientation values on the Geometry tab of the Target Coordinate System Properties dialog to align your joint angle positions with the internal equations.

For example, if your Delta robot is mounted so that the joints attached at the top plate are homed at 30° in the positive direction below horizontal (see Figure 18) and you want the Logix Designer application readout values to be zero in this position, then you must configure the Zero Angle Orientation values to -30° on the Geometry tab of the Target Coordinate System Properties dialog (see Figure 19).

Figure 18 - Delta Robot with Joints Homed at 30°
Figure 19 - 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 can be described as looking similar to 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 your Delta three-dimensional robot, see the documentation provided by the robot manufacturer.

We recommend that you program the robot within a rectangular solid defined inside the robot’s work zone. The rectangular solid can be 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. You can check the position in the event task.

To avoid issues with singularity positions, 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.

During each scan, Logix Designer application evaluates the joint positions in the forward and inverse kinematics routines to be sure that they do not violate the computed maximum positive and maximum negative joint limits.
Homing or moving a joint axis to a position beyond a computed joint limit and then invoking a MCT instruction, results in an error 67 (Invalid Transform position). For more information regarding error codes, see the Logix5000™ Controllers Motion Instructions Reference Manual, publication MOTION-RM002.

Maximum Positive Joint Limit Condition

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

Figure 21 - Maximum Positive Joint Limit Condition - L1 and L2 are Collinear
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).

Define Configuration Parameters for a Delta Three-dimensional Robot

Logix Designer application can be configured for control of robots with varying reach and payload capacities. As a result, it is very important to know the configuration parameter values for your robot including:

- Link lengths.
- Base offsets.
- End-effector offsets.

The configuration 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 by using the same measurement units.
**Link Lengths**

Link lengths are the rigid mechanical bodies attached at the rotational joints. The three-dimensional Delta robot geometry has three link pairs each 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.

**Figure 23 - Three-dimensional Delta Robot - Link Lengths Configuration Screen**

![Configuration Screen](image)

**Base Offsets**

There is one base offset value (X1b) available for the three-dimensional Delta robot geometry. Enter the value equal to the distance from the origin of the robot coordinate system to one of the actuator joints.
**End-effector Offsets**

The two end-effector offsets available for the three-dimensional Delta robot geometry are as follows. Offset values are always positive numbers.

- $X_{1e}$ is the distance from the center of the moving plate to the lower spherical joints of the parallel arms.
- $X_{3e}$ is the distance from the base plate to the TCP of the gripper.

**Figure 24 - Configuring the Base Offset and End-effector Offsets for a Three-dimensional Delta Robot**

Configure a Delta Two-dimensional Robot

**Figure 25** shows a two-dimensional Delta robot that moves in two-dimensional Cartesian space.

**Figure 25 - Two-dimensional Delta Robot**
This robot has two rotary joints that move the gripper in the \((X_1, X_2)\) 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 \(X_2\) axis and its position is translated in Cartesian space \((X_1, X_2)\) by mechanical parallelograms in each forearm assembly. The two joints, \(J_1\) and \(J_2\), are actuated joints. The joints between links \(L_1\) and \(L_2\) and between \(L_2\) and the base plate are unactuated joints.

Each joint is rotated independently to move the gripper to a programmed \((X_1, X_2)\) position. As each joint axis (\(J_1\) or \(J_2\) or \(J_1\) and \(J_2\)) is rotated, the TCP of the gripper moves correspondingly in the \(X_1\) or \(X_2\) direction or \(X_1\) and \(X_2\) direction. You can program the TCP to a \((X_1, X_2)\) coordinate, then 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 \((X_1, X_2)\) position to the programmed \((X_1, X_2)\) position.

The two joint axes \((\text{J}_1, \text{J}_2)\) of the robot are configured as linear axes.

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

**Establish the Reference Frame for a Delta Two-dimensional Robot**

The reference frame for the two-dimensional Delta geometry is at the center of the fixed top plate. When the angles of joints \(\text{J}_1\) and \(\text{J}_2\) are both at \(0^\circ\), each of the two \(L_1\) links is along the \(X_1\) axis. One \(L_1\) link is pointing in the positive \(X_1\) direction, the other in the negative \(X_1\) direction.

When the right-hand link \(L_1\) moves downward, joint \(\text{J}_1\) is assumed to be rotating in the positive direction and when \(L_1\) moves upward, the \(\text{J}_1\) is assumed to be moving in the negative direction. When the left-hand link \(L_1\) moves downward, joint \(\text{J}_2\) is assumed to be rotating in the positive direction and when left-hand \(L_1\) moves upward, the \(\text{J}_2\) is assumed to be moving in the negative direction.
Calibrate a Delta Two-dimensional Robot

The method used to calibrate a Delta two-dimensional robot is the same as the method used for calibrating a Delta three-dimensional robot. The only difference is the number of axes used. For more information about calibration, see Calibrate a Delta Three-dimensional Robot on page 64.

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.

We recommend that you define the program parameters for the two-dimensional Delta robot within a rectangle (dotted lines in Figure 27) inside the work zone of the robot. The rectangle can be defined 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. You can check the position in the event task.

To avoid problems with singularity positions, 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.

<table>
<thead>
<tr>
<th>For More Information About</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum positive joint limits</td>
<td>68</td>
</tr>
<tr>
<td>Maximum negative joint limits</td>
<td>69</td>
</tr>
</tbody>
</table>

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 about error codes, see the Logix5000® Controllers Motion Instructions Reference Manual, publication MOTION-RM002.

**Define Configuration Parameters for a Delta Two-dimensional Robot**

You can configure Logix Designer application for control of robots with varying reach and payload capacities. As a result, it is very important to know the configuration parameter values for your robot including:

- Link lengths.
- Base offsets.
- End-effector offsets.

The configuration 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 by using the same measurement units.
**Link Lengths**

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.

**Figure 28 - Two-dimensional Delta Robot - Link Lengths Configuration Screen**

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

**Base Offsets**

There is one base offset (X1b) available for the two-dimensional Delta robot geometry. Enter the value equal to the distance from the origin of the robot coordinate system to one of the actuator joints.
End-effector Offsets

There are two end-effector offsets available for the two-dimensional Delta robot geometry. The value for X1e is the offset distance from the center of the lower plate to the lower spherical joints of the parallel arms. The distance from the lower plate to the TCP of the gripper is the value for X2e.

Figure 29 - Delta Two-dimensional Robot - Base and End-effector Offsets

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).

Figure 30 - SCARA Delta Robot

Establish the Reference Frame for a SCARA Delta Robot

The reference frame for the SCARA Delta robot is at the center of the base 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 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 is orthogonal to the X1-X2 plane pointing up. The linear axis always moves in the X3 direction.

When configuring a SCARA Delta robot in Logix Designer application, keep the following in mind.

- **Configure both** 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**

The method used to calibrate a SCARA Delta robot is the same as the method used for calibrating a Delta three-dimensional robot. The only difference is the number of axes used. For more information about calibration, see [Calibrate a Delta Three-dimensional Robot](#) on page 64.
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 defines the height of the solid region.

We recommend that you program the SCARA Delta robot within a rectangular solid defined inside the robots work zone. The rectangular solid can be 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. You can check the position in the event task.

To avoid problems with singularity positions, Logix Designer application internally calculates the joint limits for the Delta robot geometries.

<table>
<thead>
<tr>
<th>For More Information About</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum positive joint limits</td>
<td>68</td>
</tr>
<tr>
<td>Maximum negative joint limits</td>
<td>69</td>
</tr>
</tbody>
</table>

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 about error codes, see the Logix5000 Controllers Motion Instructions Reference Manual, publication MOTION-RM002.

Define Configuration Parameters for a SCARA Delta Robot

Logix Designer application can be configured for control of robots with varying reach and payload capacities. As a result, it is very important to know the configuration parameter values for your robot including:

- Link lengths.
- Base offset.
- End-effector offset.

The configuration 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 by using the same measurement units.
**Link Lengths**

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.

**Base Offset**

There is one base offset (X1b) available for the SCARA Delta robot geometry. Enter the value equal to the distance from the origin of the robot coordinate system to one of the actuator joints. The base offset value is always a positive number.

**End-effector Offset**

There is one end-effector offset (X1e) available for the SCARA Delta robot geometry. Enter 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 value is always a positive number.

**Figure 32 - SCARA Delta End-effector and Base Offset**
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 both 2D and 3D delta geometries. For example, a mechanical 2D delta robot that uses a negative X1b offset has a mechanical configuration like the one shown here.

![Diagram of a Delta Robot with negative X1b offset]

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. Therefore, the base offset X1b is measured to be a value of -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 example in the previous figure is shown here.

![Logix Designer configuration window]

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

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 shown in Figure 33). One solution satisfies the equations for a right-armed robot, the other solution satisfies the equations for a left-armed robot.

Figure 33 - Right Arm and Left Arm Solutions

Solution Mirroring for Three-dimensional Robots

For a three-dimensional Articulated Independent robot, there are four possible 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 two solutions are mirror image solutions where J1 is rotated 180°.

### Activating Kinematics

**ATTENTION:** Be sure to choose an arm solution before activating the Kinematic function. Failure to do so can result in machine damage and/or serious injury or death to personnel.

Before activating Kinematics, configure the robot in a left-arm or right-arm solution. The robot stays in the same configuration in which it was activated as it is moved in Cartesian or source coordinate mode. If activated in a fully-extended-arm mode (this is, neither a left-arm nor a right-arm solution), the system chooses a left-arm solution.
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. The robot arm solution changes again (if required) when another joint move is made.

Example: Suppose, you want to move the robot from position A \((x_1,y_1)\) to position B \((x_2,y_2)\) (see the next figure). At position A, the system is in a left arm solution. Programming a Cartesian move from A \((x_1,y_1)\) to B \((x_2,y_2)\) means that the system moves along the straight line (see the illustration) 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 \(\Theta_1\) to \(\Theta_2\) and a joint move in J2 from \(\alpha_1\) to \(\alpha_2\).

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.

ATTENTION: Avoid programming your robot towards a singularity position when programming in Cartesian mode. The velocity of the robot increases very rapidly as it approaches a singularity position and can result in injury or death to personnel.
Chapter 4 Configure an Articulated Independent Robot

**Encounter a No-solution Position**

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

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.

**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.

*Figure 34 - SCARA Independent Robot Reference Frame*
The internal Kinematic equations are written as if the start position for the SCARA Independent robot joints are as shown in Figure 35.

**Figure 35 - Joint and Link Start Position that Kinematics Equations use for the SCARA Independent Robots**

- +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 Configure an Articulated Independent Robot on page 53.

When configuring the parameters for the Source coordinate system and the Target coordinate system for a SCARA Independent robot, keep the following information in mind:

- Set the transform dimension value 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 Configure an Articulated Independent Robot on page 53.
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

Logix Designer application can be configured for control of robots with varying reach and payload capacities. As a result, it is very important to know the configuration parameter values for your robot including:

- Link lengths.
- Base offsets.
- End-effector offsets.

The configuration 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 by using the same measurement units.

Figure 38 illustrates the typical configuration parameters for a SCARA Independent robot.

**Figure 38 - SCARA Independent**
**Link Lengths**

Link lengths are the rigid mechanical bodies attached at joints.

**Figure 39 - Configuring Link Lengths for a SCARA Independent Robot**

Enter the Link Length values. For the robot shown in SCARA Independent above, the Link Length values are:
- $L_1 = 20$
- $L_2 = 40$

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

**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 more information about error codes, see the Logix5000 Controllers Motion Instructions Reference Manual, publication MOTION-RM002.
## Monitor Status Bits for Kinematics

You can monitor the status of the Kinematics functions by using Logix Designer application status bits.

<table>
<thead>
<tr>
<th>To see if</th>
<th>Check this tag</th>
<th>And this bit</th>
<th>For</th>
</tr>
</thead>
<tbody>
<tr>
<td>A coordinate system is the source of an active transform</td>
<td>Coordinate system</td>
<td>TransformSourceStatus</td>
<td>On</td>
</tr>
<tr>
<td>A coordinate system is the target of an active transform</td>
<td>Coordinate system</td>
<td>TransformTargetStatus</td>
<td>On</td>
</tr>
<tr>
<td>An axis is part of an active transform</td>
<td>Axis</td>
<td>TransformStateStatus</td>
<td>On</td>
</tr>
<tr>
<td>An axis is moving because of a transform</td>
<td>Axis</td>
<td>ControlledByTransformStatus</td>
<td>On</td>
</tr>
</tbody>
</table>
Chapter 4 Configure an Articulated Independent Robot

Notes:
Configure an Articulated Dependent Robot

The Articulated Dependent robot has motors for the elbow and the shoulder at the base of the robot. The dependent link controls J3 at the elbow. Use these guidelines when configuring an Articulated Dependent robot.

---

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

1. Set and enable the soft travel limits.
2. 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.

---

**Reference Frame**

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

---

**ATTENTION:** 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 Dependent robot is at the base of the robot as shown in Figure 40.
Before you begin establishing the Joint-to-Cartesian reference frame relationship, it is important to know some information about how the Kinematic mathematical equations in the ControlLogix® controllers were written. The equations were written as if the Articulated Dependent robot joints were positioned as shown in Figure 40.

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

- J1 = 0
- J2 = 0
- J3 = 0
When your robot is physically in this position, the Logix Designer application Actual Position tags for the axes must be:

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

**Figure 42 - Articulated Dependent 3**

If the physical position and joint angle values of your robot cannot match those shown in **Figure 41** or in **Figure 42**, use one of the methods that are outlined in this section to establish the Joint-to-Cartesian reference frame relationship.

---

**ATTENTION:** Failure to properly establish the correct reference frame for your robot can cause the robotic arm to move to unexpected positions potentially resulting in damage to property or injury to personnel.

---

### Methods to Establish a Reference Frame

The following methods let you establish a reference frame for an Articulated Independent robot.

<table>
<thead>
<tr>
<th></th>
<th>Use one of these methods to establish the reference frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incremental axis</td>
<td>Each time the power of the robot is cycled.</td>
</tr>
<tr>
<td>Absolute axis</td>
<td>Only when you establish absolute home.</td>
</tr>
</tbody>
</table>

- **Method 1** - establishes a Zero Angle Orientation and lets the configured travel limits and home position on the joint axes remain operational. Use this method if you are operating the axes between the travel limits determined before programming a Motion Redefine Position (MRP) instruction and want these travel limits to stay operational.
- **Method 2** - uses a Motion Redefine Position (MRP) instruction to redefine the axes position to align with the Joint reference frame. This method can require the soft travel limits to be adjusted to the new reference frame.
Method 1 - Establishing a Reference Frame

Each axis for the robot has the mechanical hard stop in each of the positive and negative directions. Manually move or press each axis 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 as shown in Figure 42, the Logix Designer application values for the Actual Position tags are:

- J1 = 0
- J2 = 90°
- J3 = 0°

If the Logix Designer application Actual Position tags do not show these values, configure the Zero Angle Orientation for the joint or joints that do not correspond.

<table>
<thead>
<tr>
<th>If the Logix Designer application read-out values are</th>
<th>Set the Zero Angle Orientations on the Coordinate System Properties dialog to</th>
</tr>
</thead>
<tbody>
<tr>
<td>J1 = 10</td>
<td>Z1 = -10</td>
</tr>
<tr>
<td>J2 = 80</td>
<td>Z2 = 10</td>
</tr>
<tr>
<td>J3 = 5</td>
<td>Z3 = -5</td>
</tr>
</tbody>
</table>
Set the Zero Angle Orientations.

**Method 2 - Establishing a Reference Frame**

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 to with the following values 0, 90, and 0°.

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

The work envelope is the three-dimensional region of space that defines the reaching boundaries for the robot arm. The work envelope of an articulated robot is ideally a complete sphere having an inner radius equal to \(|L_1 - L_2|\) and outer radius equal to \(|L_1 + L_2|\). However, due to the range of motion limitations on individual joints, the work envelope is not always a complete sphere.

If the range-of-motion values for the articulated robot are

<table>
<thead>
<tr>
<th>J1</th>
<th>J2</th>
<th>J3</th>
<th>L1</th>
<th>L2</th>
</tr>
</thead>
<tbody>
<tr>
<td>±170</td>
<td>0...180</td>
<td>±60</td>
<td>10</td>
<td>12</td>
</tr>
</tbody>
</table>

Typically, the work envelope is

- **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°.
**Configuration Parameters**

Logix Designer application can be configured for control of robots with varied reach and payload capacities. As a result, it is important to know the configuration parameter values for your robot including:

- 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 by using the same measurement units.

Figure 44 illustrates the typical configuration parameters for an Articulated Dependent robot.

**Figure 44 - Articulated Dependent 4**

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

**Link Lengths**

Link lengths are the rigid mechanical bodies attached at joints.

<table>
<thead>
<tr>
<th>For an articulated dependent robot with</th>
<th>The length of</th>
<th>Is equal to the value of the distance between</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two-dimensions</td>
<td>L1, L2</td>
<td>J1 and J2, J2 and the end-effector</td>
</tr>
<tr>
<td>Three-dimensions</td>
<td>L1, L2</td>
<td>J2 and J3, J3 and the end-effector</td>
</tr>
</tbody>
</table>
Figure 45 - Example of Link Lengths for an Articulated Dependent Robot

Enter the Link Length values. For the robot shown in our example, the Link Length values are:
• \( L_1 = 10.0 \)
• \( L_2 = 12.0 \)

Figure 46 - Example of Base Offsets for an Articulated Independent Robot

Enter the Base Offset values. For the robot shown in our example, the Base Offset values are:
• \( X_{1b} = 3.0 \)
• \( X_{3b} = 4.0 \)

Base Offsets

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. Enter the values for the base offsets in the \( X_{1b} \) and \( X_{3b} \) fields of the Coordinate System Properties dialog.
End-effector Offsets

The robot can have an end-effector attached to the end of robot link L2. If there is an attached end-effector, then you must configure the end-effector offset value on the Coordinate System Properties dialog. The end-effector offsets are defined regarding the tool reference frame at the tool tip.

Figure 47 - Example of End-effector Values for an Articulated Independent Robot

Enter the end-effector offset values. For the robot shown in our example, the end-effector values are:
- $X_{1e} = 2.0$
- $X_{3e} = -3.0$
Notes:
Configure a Cartesian Gantry Robot

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Establish the Reference Frame for a Cartesian Gantry Robot</td>
<td>101</td>
</tr>
<tr>
<td>Identify the Work Envelope for a Cartesian Gantry Robot</td>
<td>101</td>
</tr>
<tr>
<td>Define Configuration Parameters for a Cartesian Gantry Robot</td>
<td>102</td>
</tr>
</tbody>
</table>

Use these guidelines when configuring a Cartesian Gantry robot.

**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.

![Cartesian Reference Frame](image)

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. You can 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 Logix5000™ 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

You do not need to define the link lengths, base offset, or end-effector offset configuration parameters for a Cartesian Gantry robot.
Configure a Cartesian H-bot

About Cartesian H-bots

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.

Figure 49 - Cartesian H-bot

In Figure 49, 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 enables it to move at a 45° angle to the axes when either 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 both rotated clockwise at the same speed, then the machine moves along a horizontal line
• Motors A and B are both 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 \((X1=0, X2=14.142)\).

While this configuration can be very confusing for a programmer, utilizing the Logix Designer application Kinematics function configured with two Cartesian coordinate systems and a \(-45^\circ\) rotation easily performs the function.

To configure two Cartesian coordinate systems, Coordinate system 1 (CS1) and Coordinate system 2 (CS2), each containing two linear axes, use the following steps.

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 \(-45^\circ\) rotation introduced by the Kinematics, counteracts the \(45^\circ\) 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 \(X1_{\text{virt}}=10, X2_{\text{virt}}=5\) moves to a real mechanical position of \(X1=10, X2=5\).

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. Do not rotate the angular rotation of the reference frame for this robot because the angular rotation vector is used to achieve the \(45^\circ\) rotation required for 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

You do not need to define the link lengths, base offset, or end-effector offset configuration parameters for a Cartesian H-bot.
Notes:
Configure Camming

This chapter describes camming concepts. You use the motion coordinated instructions to move as many as three axes in a coordinate system. Descriptions of these instructions are in the Logix5000™ Controllers Motion Instructions Reference Manual, publication MOTION-RM002.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camming Concepts</td>
<td>107</td>
</tr>
<tr>
<td>Cam Profiles</td>
<td>109</td>
</tr>
<tr>
<td>Scaling Cams</td>
<td>115</td>
</tr>
<tr>
<td>Cam Execution Modes</td>
<td>117</td>
</tr>
<tr>
<td>Execution Schedule</td>
<td>117</td>
</tr>
<tr>
<td>Pending Cams</td>
<td>122</td>
</tr>
</tbody>
</table>

Camming Concepts

Camming is the process of coordinating the movement of a master axis and a slave axis, where the movement of one is dependent on the movement of the other.

There are two types of camming:
- Mechanical Camming
- Electronic Camming

**IMPORTANT** Logix5000 motion control applications use electronic camming. A brief description of mechanical camming provides you an understanding of both types of camming. However, the remainder of this publication describes how to use electronic camming in your motion control application.

Mechanical Camming

In mechanical camming, the master axis functions as a cam. A cam is an eccentric wheel that is 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.
Chapter 8 Configure Camming

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.

This figure illustrates a mechanical cam turning in a clockwise manner and the affect it has on a follower that is physically connected to it.

Figure 50 - Mechanical Cams

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 or 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. An array of cam elements represents a cam profile. The point pair that is 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**

### 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 that is 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. When the master axis reaches a specific position, the slave axis moves to its specific corresponding point, as defined in the table of points for the cam profile.

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 Logix5000 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. The linear or cubic interpolation between adjacent points determines the slave axis position if the following is true:

- The current master position or time does not correspond exactly with a point in the cam array that is used to generate the cam profile.

In this way, the smoothest possible slave motion is provided. The MCCP instruction accomplishes this condition 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 that is 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 condition allows electronic camming motions to be superimposed with jog, or move profiles to create complex motion and synchronization.

**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 Logix5000 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. The linear or cubic interpolation between the adjacent points determines the slave axis position if the following is true:

- The current master time value does not correspond exactly with a point in the cam table that is associated with the cam profile.
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 condition allows electronic camming motions to be superimposed with jog, or move profiles to create complex motion and synchronization.

**Calculating 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.

**Using 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.

**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.
Figure 51 - Acceleration Cam Profile

Run Cam Profile

A run cam profile determines the movement of a slave axis. This process begins when the master axis reaches a specific position and remains steady until the end of the cam profile. Figure 52 illustrates a sample run cam profile in the Logix Designer programming software cam editor.

Figure 52 - Run Cam Profile
Deceleration Cam Profile

A deceleration cam profile determines the deceleration of a slave axis from a particular position. Figure 53 illustrates a sample deceleration cam profile in the Logix Designer programming software cam editor.

Figure 53 - Deceleration Cam Profile

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. Figure 54 illustrates a sample dwell cam profile in the Logix Designer programming software cam editor.
Figure 54 - Dwell Cam Profile

Behavior of Pending Cams

If you want to run one profile and then pend another one, you must 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 ½ Cycle = 1 Cycle. Both of these actions 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.
Scaling Cams

You can use the scaling feature to determine the general form of the motion profile with one 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, in position cam profiles, than when it is used with an MATC instruction, in time cam profiles.

Scaling Position Cam Profiles

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, as shown in Figure 55.

The master and slave values that the cam profile array defines take on the position units of the master and slave axes respectively. This process occurs when an MAPC instruction specifies a position cam profile array. By contrast, the Master and Slave Scaling parameters are ‘unit-less’ values that are 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.

If you increase the Master Scaling value of a position cam profile, it decreases the velocities and accelerations of the profile. However, if you increase the slave 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 Master Scaling and Slave Scaling values must be equal. For example, if the Slave Scaling value of a profile is 2, the Master Scaling value must also be 2. This requirement is to maintain approximately equal velocities and accelerations during execution of the scaled position cam.

**ATTENTION:** 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.

**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 MA TC 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.

**Figure 56 - Scaling Time Cam Profile**

![Diagram showing scaling of time cam profiles](image)

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.
Configure Camming  Chapter 8

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.

---

**Cam Execution Modes**

Cam execution modes determine if the cam profile is executed only one time or repeatedly. You must configure the Execution Mode parameter on an MAPC or MATC instruction.

**Table 14 - Execution Mode Descriptions**

<table>
<thead>
<tr>
<th>Execution Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Once</td>
<td>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.</td>
</tr>
<tr>
<td>Continuous</td>
<td>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.</td>
</tr>
<tr>
<td>Persistent(1)</td>
<td>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.</td>
</tr>
</tbody>
</table>

(1) This selection is only available on the MAPC instruction.

---

**Execution Schedule**

The Execution Schedule parameter controls the execution of an instruction. You must 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.

**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.

Figure 57 - Immediate Execution

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 diagram Figure 58 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.

**Figure 58 - Changing the Cam Lock Position**

---

**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 on page 122](#).

**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.
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.

**MATC Instruction**

An MATC instruction uses one of two Execution Schedule parameters:

- Immediate
- Pending

**Immediate**

By default, the MATC instruction is scheduled to execute immediately because the default setting of Execution Schedule is Immediate. 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 Figure 60. 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.

**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 graphic *Pending Cam Execution*. 
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 Figure 62.

**Figure 62 - Pending Position Cam**
Notes:
Index

A
acceleration 111
acceleration cam 111
Arm Solution
definition of configuring 81
Articulated Dependent
base offsets 98
define configuration parameters 97
define effector offsets 99
establish the reference frame 91
establish the reference frame alternate methods 93
identify the work envelope 96
link lengths 97
Articulated Independent
base offsets 61
configuration parameters 59
define effector offsets 62
establish reference frame 53, 59
establish reference frame methods 55
identify the work envelope 57
link lengths 60
axis
inhibit 47, 53
B
behavior 114
bidirectional 117
C
calculating 111
cam 115, 118
master axis 109
cam execution modes 117
cam pending 122
cam profile 109, 111
cam profile array 115
cam profile editor 111
cam profile scaling time 116
cam shape 108
Cartesian Gantry
configuration parameters 102
establish reference frame 101
identify the work envelope 101
Cartesian H-bot
configuration parameters 105
establish reference frame 104
identify the work envelope 104
changing 118
coefficients
calculating 110
Collinear Moves
velocity profiles
termination types 41
common cam profiles 111
Configure 62, 65
Coordinate System Dialog Boxes
Dynamics 17
General 17
Geometry 17
Manual Adjust 17
Offset 17
Tag 17
Units 17
Coordinate System Properties
Dynamics Tab 25
Manual Adjust 27
Manual Adjust Button 27
Position Tolerance Box, Actu 27
Position Tolerance Box, Command 27
Vector Box 25
Vector Box, Maximum Acceleration 26
Vector Box, Maximum Acceleration Jerk 26
Vector Box, Maximum Deceleration 26
Vector Box, Maximum Deceleration Jerk 26
Vector Box, Maximum Speed 26
Editing 18
General Tab 19
Axis Grid 19
Axis Name 20
Brackets 19
Coordinate 19
Coordinating Mode 20
Dimension 19
Ellipsis 20
Ellipsis button 19
Enable Coordinate System Auto Tag Update 20
Motion Group 19
New Group button 19
Transform Dimension 19
Type 19
Geometry Tab 21
Link Lengths 21
zero angle orientations box 22
Joints Tab 24
Joint Ratio 24
Joint Units 24
Offsets Tab 23
Base Offsets 24
Tag Tab 29
Data Type 29
Description 29
Name 29
Scope 29
Tag Type 29
Units Tab
Axis Grid 23
Axis Grid, Axis Name 23
Axis Grid, Conversion Ratio 23
Axis Grid, Conversion Ratio Units 23
Coordination Units 22
Coordinate system properties
Offsets Tab
End Effector 24
cubic 110
Index

D
deceleration 111
deceleration cam 113
Delta 62
Delta Robot
  Maximum Negative Joint Limit Condition 69
  Maximum Positive Joint Limit Condition 68, 69, 71, 72, 73, 74, 76, 78
  types
  configure 62
Delta three-dimensional
  configuration parameters 69
  configure 62
  maximum positive joint limit condition 68
  reference frame 64
  work envelope 67
  zero angle orientation 65
Delta two-dimensional
  configuration parameters 74
  configure 71
  establish the reference frame 72
  work envelope 73
dimension 115
dwell 111
dwell cam 113

E
electronic 107
  electronic camming 108
End Effector Offsets
determining 98
error 121, 122, 124
error code 23 122
  establish the reference frame 64
  execution 123
  execution schedule parameter 117, 121

F
forward only 117

G
Geometry
  of robot 50
  Geometry Tab
  link lengths 21
  zero angle orientations box 22

I
identify the work envelope 67
illegal dynamic change 121, 122, 124
immediate 117, 121
  immediate execution 118
inhibit
  axis 47, 53
  interpolated 110
interpolation 110

J
jog
  execution of 110

K
Kinematics
  activating 82
  arm solutions 81, 83
  arm solutions for two axes robots 81
  Articulated Independent 53
  changing arm solutions 83
  determine Coordinate system type 50
  no solution 84
  singularity 83
  solution mirroring 81
  terms 48
kinematics
  See multi-axis coordinated motion instructions

L
linear 110
lock position 118
Logix Designer Cam Profile Editor 123

M
MAPC 109, 114, 115, 117, 118, 120, 121, 123, 125
  cam profile 111
  master 118
  master axis 108, 119, 125
    execution mode 117
    position values 109
    time cam profile 110
  master axis position 123
  master axis time 123
MATC 115, 117, 121, 123, 125
  cam profile 111
  time cam 110
Maximum Acceleration 26, 28
MCCP 110, 111
MCLM
Examples
  Blending Different Speeds 46
MCTP 47
mechanical
  107
  mechanical camming 107
Motion Axis Position Cam (MAPC) 109
  changing the cam lock position 118
  forward only, reverse only, or bidirectional execution 119
  immediate execution 118
motion coordinated instructions
See multi-axis coordinated motion instructions

MRP 120

Multi-Axis Coordinated Motion Instructions
107
Introduction 13
MCSR 47
Motion Coordinated Change Dynamics (MCOD) 9
Motion Coordinated Circular Move (MCCM) 9
Operands Termination Type 39, 41, 43
Operands Termination Type Actual Tolerance 39, 41, 43
Operands Termination Type Command Tolerance 39, 41, 43
Operands Termination Type Follow Contour Velocity Constrained 39, 41, 43
Operands Termination Type No Decel 39, 41, 43
Operands Termination Type No Settle 39, 41, 43
Motion Coordinated Linear Move (MCLM) 9
Operands Termination Type 39, 41, 43
Operands Termination Type Follow Contour Velocity Unconstrained 39, 41, 43
Symmetric Profiles 43
Velocity Profiles 41
Motion Coordinated Shutdown (MCSD) 9
Motion Coordinated Shutdown Reset (MCSR)
Description 47
Motion Coordinated Stop (MCS) 9

N
Naming a Coordinate System 14
non-linear motion 109

P
pending 117, 119, 121, 122, 125
position cam 125
pending cam 114, 123
point pair 108
position cam 115, 118, 123, 125
position cam profile 109
position-lock cams 109
profile 111

R
reciprocating motion 107
reverse only 117
run 111
run cam 112

S
slave axis 125
scaling 115
SCARA Delta
configuration parameters 78
establish the reference frame 76
identify the work envelope 78
SCARA Independent
reference frame 84, 86
Selective Compliant Assembly Robot Arm
configuration parameters 87
configure 81
establish reference frame 84
identify work envelope 86
link lengths 88
Singularity
planning for definition of 83
slave axis 107, 111, 119
acceleration cam profile 111
execution mode 117
move process 110
position values 109
time cam profile 110
slave axis position 123
status bit 118
Studio 5000 Logix Designer application 9
Symmetric Profiles
paths of 43

T
time cam profile 109, 110
time cam status 122
types of camming 107

V
Velocity Profiles
of collinear moves 41
triangular 45
Notes:
Rockwell Automation Support

Use the following resources to access support information.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct Dial Codes</td>
<td>Find the Direct Dial Code for your product. Use the code to route your call directly to a technical support engineer.</td>
<td><a href="http://www.rockwellautomation.com/global/support/direct-dial.page">http://www.rockwellautomation.com/global/support/direct-dial.page</a></td>
</tr>
</tbody>
</table>

Documentation Feedback

Your comments will help us serve your documentation needs better. If you have any suggestions on how to improve this document, complete the How Are We Doing? form at http://literature.rockwellautomation.com/idc/groups/literature/documents/du/ra-du002_en-e.pdf.


Allen-Bradley, CompactLogix, ControlLogix, FactoryTalk, GuardLogix, Logix5000, Rockwell Automation, Rockwell Software, SoftLogix, and Studio 5000 Logix Designer are trademarks of Rockwell Automation, Inc. Trademarks not belonging to Rockwell Automation are property of their respective companies.

Rockwell Otomasyon Ticaret A.Ş., Kar Plaza İş Merkezi E Blok Kat:6 34752 İçerenköy, İstanbul, Tel: +90 (216) 5698400

www.rockwellautomation.com