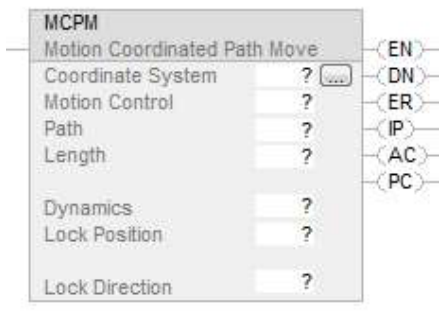
# Motion Coordinated Path Move (MCPM)

This information applies to the Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, and GuardLogix 5580 controllers.

Use the MCPM instruction to start a multi-dimensional coordinated path move for the specified Primary axes (X, Y, Z) and orientation axes (Rx, Ry, Rz) of a Cartesian coordinate system. Use this instruction to program Cartesian moves on robots with orientation control.

## Available Languages

### Ladder Diagram




### Function Block

This instruction is not available in function block.

### Structured Text

MCPM (CoordinateSystem, MotionControl, Path, Length, Dynamics, LockPosition, LockDirection);

## Operands

**Important:** Do not use the same tag name for more than one instruction in the same program. Do not write to any instruction output tag under any circumstances.

**ATTENTION:** If instruction operands are changed while in Run mode, the pending edits must be accepted and the controller mode cycled from Program to Run for the changes to take effect.

## Configuration

The following table provides the operands used to configure the instruction. These operands cannot be changed at runtime.

| Operand | Data Type | Format | Description |
|---|---|---|---|
| Coordinate System | COORDINATE_SYSTEM | Tag | Cartesian coordinate system used to program the moves. |
| Motion Control | MOTION_INSTRUCTION | Tag | Structure used to access instruction status parameters. |

For further information of configuring coordinate systems refer to the Motion Coordinate System User Manual publication MOTION-UM002.

### Inputs

Search

- Quick Start Steps
- Logix Designer
- Module Information
- Instruction Set
  - Logix 5000 Controllers Instruction and Application Considerations
  - Logix Designer Application Instruction Set
  - Interpret the Attribute Tables
  - Array Concepts
  - CIP Axis Attributes
  - Module Configuration Attributes
  - Bit Addressing
  - Common Attributes
  - Data Conversions
  - Elementary data types
  - LINT data types
  - Floating Point Values
  - Immediate values
  - Index Through Arrays
  - Math Status Flags
  - Motion Error Codes (.ERR)
  - Structures
  - Equipment Sequence instructions
  - Equipment Phase Instructions
  - Alarm Instructions
  - Advanced Math Instructions
  - Array (File)/Misc Instructions
  - Array (File)/Shift Instructions
  - ASCII Conversion Instructions
  - ASCII Serial Port Instructions
  - ASCII String Instructions
  - Bit Instructions
  - Compare Instructions
  - Debug Instructions
  - Drives Instructions
  - Drive Safety Instructions
  - For/Break Instructions
  - Filter Instructions
  - Function Block Attributes
  - Structured Text Attributes
  - Compute/Math Instructions
  - Move/Logical Instructions
  - Input/Output Instructions
  - License Instructions
  - Math Conversion Instructions
  - Metal Form Instructions
  - Motion Configuration

## Inputs

The following table explains the instruction inputs. The inputs may be field device signals from input devices or derived from user logic.

| Operand | Data Type | Format | Description |
|---|---|---|---|
| Path | PATH_DATA | Tag | See PATH_DATA Structure |
| Length | DINT | Immediate Tag | This input is immediate and indicates the length of the PATH_DATA input.<br><br>**Tip:** Set the length to 1. Values greater than 1 are reserved for future use. |
| Dynamics | DYNAMICS_DATA | Tag | See DYNAMICS_DATA Structure |
| Lock Position | REAL | Tag | Position on the Master Axis where a Slave should start following the master after the move has been initiated on the slave axis coordinate system.<br><br>**Tip:** Lock Position is only valid when the MCPM instruction is used in Master Driven Speed Control mode. |
| Lock Direction | UINT32 | Tag | Specifies the conditions when the Lock should be activated.<br><br>**Tip:** Lock Direction is only valid when the MCPM instruction is used in Master Driven Speed Control mode. |

## PATH_DATA Structure

| Operand | Scroll, List, or Check Box | Data Type | Default | Notes |
|---|---|---|---|---|
| Interpolation Type | List of Point-to-Point (0)<br>Continuous Path Linear (1) | DINT | 0 | Point-to-point move[2]<br><br>Continuous Path Linear<br><br>See Interpolation Type related topics below. |

- Logical and Move Instructions
- Program Control Instructions
- Sequencer Instructions
- Special Instructions
- Timer and Counter Instructions
- Trigonometric Instructions
- Process Control Instructions
- Select/Limit Instructions
- Sequential Function Chart (SFC) Instructions
- Statistical Instructions
- Safety Instructions
- Studio 5000 Logix Designer Glossary


| | | | | |
|---|---|---|---|---|
| Position [X, Y, Z, Rx, Ry, Rz] | List of constant or variable | REAL [9] 1 | 0 | [X,Y.Z] in Coordination Units,<br>[Rx, Ry, Rz] X-Y-Z fixed angle format in degrees<br>Index 0: X<br>1: Y<br>2: Z<br>3: Rx<br>4: Ry<br>5: Rz<br>6: *[2]<br>7: *[2]<br>8: *[2]<br>See Position related topics in the following section. |
| Robot Configuration | List of Bit values :<br>**Bit0 –** Robot Configuration Change(1)/Same(0)<br>**Bit1 –** Lefty(1)/Righty(0)<br>**Bit2 –** Above(1)/Below(0)<br>**Bit3 –** Flip(1)/No flip(0) | DINT | 0 | **Bit 0 to 3 –** Applies only to Articulated and SCARA geometries. Set to zero for Delta geometries.<br>See Robot configuration in section below. |
| Turns Counters | List of variable | INT 16 [4] | 0 | Index 0: J1<br>1: J4<br>2: J6<br>3 [2]<br>Joint axes turns counters. Each integer is a signed value (± 127). |
| Move Type | List of Absolute (0), Incremental (1) | DINT | 0 | Select the move type.<br>See the MCLM instruction for more information on this operand. |
| Termination Type | List of Actual Tolerance(0), No Settle(1) Command Tolerance(6) | DINT | 0 | See the MCLM instruction for more information on this operand. |
| Command Tolerance Linear | List of constant or variable | REAL | 0 | Used for Cartesian Primary axes position only.<br>Coordination Units Linear |

# DYNAMICS_DATA_STRUCTURE

| Operand | Scroll, List, or Check Box | Data Type | Default | Notes |
|---|---|---|---|---|
| Units Mode | List of 0 = % of Maximum<br>1 = Coord Units (per) | DINT | 0 | See Units Mode section below. |
| Time Units | List of 0 = Seconds<br>1 = Master Units | DINT | 0 | Not applicable if Percentage of Maximum is selected as unit mode.<br>Applies to speed, acceleration, and deceleration only.<br>See Time Units section below. |
| Profile | List of 0= Trapezoidal<br>1= S-Curve | REAL | 0 | [1]<br>See Profile section below. |
| Speed | List of Constant or Value | REAL | 0 | [1]<br>% of Maximum or Coordination Units/Time Units. |
| Acceleration | List of Constant or Value | REAL | 0 | [1]<br>% of Maximum or Coordination Units/Time Units$^2$. |
| Deceleration | List of Constant or Value | REAL | 0 | [1]<br>% of Maximum or Coordination Units/Time Units$^2$. |
| Acceleration Jerk | List of Constant or Value | REAL | 0 | % of Time accelerating<br>For all axes always<br>See Acceleration Jerk section below.<br>Applies to acceleration and orientation acceleration. |
| Deceleration Jerk | List of Constant or Value | REAL | 0 | % of Time deceleration<br>For all axes always.<br>Applies to deceleration and orientation deceleration. |
| Orientation Speed | List of Constant or Value | REAL | 0 | % of Maximum of orientation speed in coordinate system configuration or Degrees/Time Unit. |
| Orientation Acceleration | List of Constant or Value | REAL | 0 | % of Maximum orientation acceleration in coordinate system configuration or Degrees/Time Unit2. |

| | | | | |
|---|---|---|---|---|
| | | | | |
| Orientation Deceleration | List of Constant or Value | REAL | 0 | % of Maximum orientation deceleration in coordinate system configuration or Degrees/Tim Unit2. |

[1] The units mode specifies the units for dynamics parameters of the coordinated move and not the individual axes. The two selections give an option for the user to program directly in Coordination Units or % of the maximum configured in the coordinate system tag. All the primary axes are configured in different user units, such as mm, inches, and cm, while the orientation axes units are in degrees.

[2] Reserved for future use.

# Path Data

**Interpolation Type**

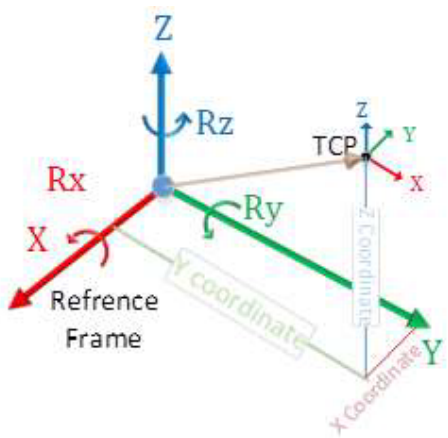
- **Point to Point**: In this type of motion the end position of the Tool Center point (TCP) is designated but the path used to reach the end position is irrelevant. This is generally the quickest way to move the TCP to a destination position. This type of interpolation will be available only for the Articulated and SCARA type geometries in future releases.
- **Continuous Path Linear**: In this type of motion the TCP moves from a starting position to a commanded end point along a straight line. During the linear move the orientation of the TCP changes continuously from the start orientation position to the commanded orientation position.

# Position

This is a one dimensional array, whose dimension is defined to be at least equivalent to the number of axes specified in the coordinate system, 6 in Cartesian coordinate system where X, Y, Z, Rx, Ry, Rz are values of the TCP with reference to a reference frame. The Position array defines either the new absolute or incremental position.

Below is an example of TCP point which has translation on X, Y, and Z axis followed with rotation on Z axis (Rz = 90°) resulting in TCP's X axis aligning with Y axis of reference frame.

See Configuring Cartesian XYZRxRyRz Coordinate System for more details on Cartesian coordinate system.




# Robot configuration

A robot can reach an end destination position with different joint positions and, consequently different robot configurations. If the user wants the robot to move to a position with a specific configuration in continuous path (CP) mode, specify the end position along with the desired pose of the robot.

- RIGHTY versus LEFTY: For future use
- ABOVE versus BELOW: For future use
- FLIP versus NOFLIP: For future use

# Dynamics Data

Dynamics Data

## Profile

- Trapezoidal31 (0): When the value is set to 0 the profile setting takes precedence over the acceleration Jerk and deceleration Jerk value (values are defaulted to 0%), as a result the velocity profile is always trapezoidal. See Profile topic in MCLM for more details.
- S Curve (1): When the value is set to 1 the acceleration and deceleration jerk values are taken into account. The planner attempts to achieve the acceleration and deceleration value calculated from the dynamics settings. Refer to the Profile information in the MCLM instruction topic for further details.

## Units Mode

- % of Maximum (0): When Units Mode is selected to be percentage of Maximum then the 6 dynamics parameters (as shown in 1st column in table below) are programmed to be percentage of those defined in dynamics tab of the coordinate system (as shown in 2nd column in table below).
  For Example if Units Mode is set to percentage of Maximum, the Speed is set to 50, and the Vector. Maximum Speed is set to 100mm/sec: the maximum linear speed of the MCLM could be up to 50% of the Maximum speed, i.e. 50mm/sec. This option is available for only CP moves ( Support for PTP in future).

| Dynamics Data Parameter | Coordinate System Parameters Specification |
|---|---|
| Speed | Vector. Maximum Speed. Primary |
| Acceleration | Vector. Maximum Acceleration. Primary |
| Deceleration | Vector. Maximum Deceleration. Primary. |
| Orientation Speed | Vector Maximum Speed Orientation |
| Orientation Acceleration | Vector Maximum Acceleration Orientation |
| Orientation Deceleration | Vector Maximum Deceleration Orientation |

- Coordination Units (1): The speed for 6 parameters (as shown in 1st Column in table above) are programming units of Coordination Units per Time Unit, as defined by the Time Units parameter.
  - Speed is programmed in Coordination Units per Second or Coordination Units per Master Unit.
  - Acceleration and Deceleration in Coordination Units per Seconds$^2$ or Coordination Units per second Master Units$^2$.
  - Orientation Speed is programmed in Degrees per Second or Degrees per Master Unit.
  - Orientation Acceleration and Deceleration in Degrees per Seconds$^2$ or Degrees per Master Units$^2$

## Time Units

- Seconds (0): Time units is in seconds
- Master Units (1): In a Master Slave configuration, the Speed/Acceleration/Deceleration of Master determines the final Primary and Orientation speed/Acceleration/Deceleration of the MCPM instruction.

For the example consider that a MDCC establishes a linkage between Master axis and slave Coordinate system. The Master axis is being moved by an MAM instruction and an MCPM instruction is executing: The resultant Speed/Acceleration/Deceleration value can be calculated as shown below.

- Formula for velocity conversion from [Units/Master Units] to [Units/Seconds]:
  $V_S[SU/TU] = V_S[SU/MU] * VM[MU/TU]$
  If the Slave is running at 3 Degrees/MU and master is running at 2 MU/sec then the slave speed is: 3*2 = 6 Degrees/Second.
- Formula for acceleration and deceleration conversion from [Units/Master Units2] to [Units/Seconds2]:
  $aS[SU/TU2] = aS[SU/MU2] * VM^2 [MU/TU] + VS[SU/MU] * aM[MU/TU2]$
  - If the Slave is accelerating at 5 Degrees/MU2 and master is running at constant speed 2 MU/sec (aM=0), the slave acceleration would be: 5*2*2 = 20 Degrees/Second2.
  - If the Slave is accelerating at 5 Degrees/MU2, its instantaneous speed is 3 Degrees/MU and the Master is running at speed 2 MU /sec and acceleration 4 MU /sec2 then the slave acceleration is: 5*2*2 + 3*4 = 32 Degrees/Second2.

# Acceleration Jerk

This parameter specifies the percentage of Time, of the Acceleration that the MCPM path should use, to compute the acceleration jerk rate of the MCPM path move. This is always programmed in units of percentage of time.

See Jerk Unit topic below on how to convert percentage of Time to engineering units.

See Acceleration Jerk topic in MCLM document for more details

# Deceleration Jerk

This parameter specifies the percentage of of Time for the Deceleration that the MCPM path should use to compute the deceleration jerk rate of the MCPM path move. This is always programmed in units of percentage of time.

See Jerk Units topic below on how to convert percentage of Time to engineering units.

See Deceleration Jerk topic in MCLM document for more details

# Jerk Unit: percentage of time (% of time)

If you want to convert % of Time to engineering units, use these equations.

- For Acceleration jerk calculation in Engineering Units, use this calculation:

$$J_a\left[\frac{EU}{S^3}\right] = \frac{a_{max}^2\left[\frac{EU}{S^2}\right]}{V_{max}\left[\frac{EU}{S}\right]}\left(\frac{200}{J_a\ [\%\ of\ Time]} - 1\right)$$

- For deceleration jerk calculation in Engineering Units, use this calculation:

$$J_d\left[\frac{EU}{S^3}\right] = \frac{d_{max}^2\left[\frac{EU}{S^2}\right]}{V_{max}\left[\frac{EU}{S}\right]}\left(\frac{200}{J_d\ [\%\ of\ Time]} - 1\right)$$

# Orientation Speed

Orientation speed is a vector speed, its value is applied as a single orientation vector composed of Rx Ry and Rz as orthogonal components. For example if the Orientation speed is 5 Deg/Sec.

$$\sqrt{V_{Rx}^2 + V_{Ry}^2 + V_{Rz}^2} = \sqrt{25} = 5 \text{ Degrees/Sec}$$

# Orientation Acceleration and Deceleration

Orientation acceleration/deceleration is a vector value, its value is applied as a single orientation vector composed of Rx, Ry and Rz as orthogonal components. For example if

the Orientation acceleration or deceleration is set to 5 Deg/Sec$^2$, then

$$\sqrt{A_{Rx}^2 + A_{Ry}^2 + A_{Rz}^2} = \sqrt{25} = 5 \text{ Deg/Sec}^2$$

# Lock Direction

Specifies the direction that the master axis must be moving when it crosses the Lock Position for the lock to be activated.

**Tip:** When the Time Unit is set to Master units the value set in this field should be the direction of Master. Refer to the MCLM instruction for further information on the Lock Direction operand.

# Lock Position

Specifies the master axis position where the slave will become locked to the master axis.

# Outputs

N/A

# Affects Math Status Flags

No

# Major/Minor Faults

None specific to this instruction. See Index Through Arrays for array-indexing faults.

# Execution

# Ladder Diagram

| Condition/State | Action Taken |
|---|---|
| Prescan | Same as Rung-condition-in is false. |
| Rung-condition-in is false | The .EN, .DN, and .ER are cleared to false. |
| Rung-condition-in is true and .EN bit is false | The .EN bit is set to true and the instruction executes. |
| Rung-condition-in is true and .EN bit is true | N/A |
| Postscan | N/A |

# Structured Text

| Condition/State | Action Taken |
|---|---|
| Prescan | See Prescan in the Ladder Diagram table. |
| Normal execution | See Rung-condition-in is false, followed by Rung-condition-in is true in the Ladder Diagram table. |
| Postscan | See Postscan in the Ladder Diagram table. |

# Error Codes

See Motion Error Codes (.ERR) for motion instructions.

# Extended Error Codes

Extended Error codes provide additional instruction-specific information for the Error Codes that are generic to other instructions. See Motion Error Codes (ERR) for Motion Instructions. Extended Error Codes meaning depends on the Error Codes they are associated with.

| **Error Code (.ERR)** | **Extended Error Code (.EXERR)** | **Description** |
|---|---|---|
| 7 | 0 thru 5 | Shutdown State Error<br><br>For a motion coordinated instruction, look at the extended error code (EXERR). It identifies which axis caused the error.<br>Example: If EXERR is zero, check the axis for dimension zero. |
| 11 | 0 thru 5 | Axis Not Configured<br><br>For single axis instructions:<br><br>The Extended Error code for MAG, MDAC, MAPC, MAM, MAJ, MATC, and MCD is defined as:<br><br>1 = Slave axis<br><br>2 = Master Axis<br><br>For the MAM, MCD, and MAJ instructions in time driven mode, the axis being moved is a slave axis.<br><br>For multi-axes instructions:<br><br>The Extended Error code for MCPM, MDCC, MCLM, MCCM, and MCCD is defined as: The axis number in the coordinate system where<br><br>0 = 1st axis<br><br>2 = Master Axis or 3rd Slave Axis |
| 13 | 2, 3, 5 | Parameter Out of Range<br><br>An EXERR = 0 means the first operand of the instruction is outside its range. |
| 16 | 0 thru 5 | Homing is in process on an axis.<br><br>Extended error code indicates which axis caused the error. |
| 25 | 0 thru 5 | You attempted to execute an instruction that is not correct.<br><br>Extended error code indicates which axis caused the error<br><br>**Tip:** MCPM returns this error if the axis is configured in torque mode. |

| | | |
|---|---|---|
| 53 | 0 thru 5 | Axis is inhibited<br><br>For single axis instructions, the Extended Error code for MAG, MDAC, MAPC, MAM, MAJ, MATC, and MCD is defined as:<br><br>1 = Slave axis<br><br>2 = Master Axis<br><br>For the MAM, MCD, and MAJ instructions in time driven mode, the axis being moved is a slave axis.<br><br>For multi-axes instructions, the Extended Error code for MCPM, MDCC, MCLM, MCCM, and MCCD is defined as:<br><br>The axis number in the coordinate system where<br><br>0 = 1st axis<br>2 = Master Axis or 3rd Slave Axis |
| 65 | 0 thru 5 | Axis Position overflow<br><br>The range for position depends on the conversion constant of the axis.<br><br>Maximum positive position = 2,147,483,647 / conversion constant of the axis.<br><br>Maximum negative position = -2,147,483,648 / conversion constant of the axis.<br><br>Select a conversion constant of 2,097,152 counts/inch. In this case:<br><br>• Maximum positive position = 2,147,483,647 / 2,097,152 counts/inch = 1023 inches.<br>• Maximum negative position = -2,147,483,648 / 2,097,152 counts/inch = -1023 inches.<br>For a motion coordinated instruction, look at the extended error code (EXERR). It identifies which axis caused the error. |
| 76 | 0 thru 5 | Maximum deceleration jerk is set to zero<br><br>You cannot start motion that uses an S-curve profile if the maximum deceleration jerk for the axis is zero.<br>(EXERR). It identifies which axis caused the error. |
| 138 | 0 | MCPM Path Data Invalid Value<br><br>MCPM Path Data Interpolation Type<br><br>If the interpolation type is set to anything other than 0 or 1, the instruction will report this error. |

| | | |
|---|---|---|
| 138 | 1 | MCPM Path Data has Invalid Value.<br><br>The instruction will report error if either of the below conditions are true.<br><br>• If the end position is either infinity or NAN<br>• If kinematic transform is active<br>  ◦ If end position Rx is not between +/-180 deg for absolute moves<br>  ◦ If end position Rz is not between +/-180 deg for absolute moves<br>  ◦ If end position Ry is not between +/-90 deg for absolute moves<br>• If kinematic transform is active AND robot Geometry is Delta J1J2J6 (Delta 3 axes) or Delta J1J2J3J6 (Delta 4 axes)<br>  ◦ If end position Rx is not 180° for absolute moves<br>  ◦ If end position Ry is not 0° for absolute moves<br>• If kinematic transform is active AND robot Geometry is Delta J1J2J3J4J5 (Delta 5 axis)<br>  ◦ If end position Rx is not 0 or 180° for absolute moves. |
| 138 | 2 | MCPM Path Data Invalid Value in MCPM Path Data Robot Config<br>If an invalid Robot Configuration is specified (any bits other than but 0,1,2,3 are set), the instruction will report this error. |
| 138 | 3 | MCPM Path Data Invalid Value in MCPM Path Data Turns Counter<br>If turns counter value is not within +/-127, the instruction will report this error. |
| 138 | 4 | MCPM Path Data Invalid Value in MCPM Path Data Move Type<br>If the move type is not either absolute or incremental, the instruction will report this error. |
| 138 | 5 | MCPM Path Data Invalid Value in MCPM Path Data TT Type<br>If any term types other than 0, 1 and 6 are specified, the instruction will report this error. |
| 138 | 6 | MCPM Path Data Invalid Value in MCPM Path Data CMD TOL<br>If negative command tolerance is specified or INF or NAN is specified, the instruction will report this error. |
| 139 | 0 | MCPM Dyn Data Invalid Value in MCPM Dyn Data Units Mode<br>If Units Mode is specified anything other than 0, 1, or 2, instruction will report this error. |
| 139 | 1 | MCPM Dyn Data Invalid Value in MCPM Dyn Data Time Units<br>If Time Units is specified anything other than 0 or 1, the instruction will report this error. |
| 139 | 2 | MCPM Dyn Data Invalid Value in MCPM Dyn Data Profile<br>If Profile is specified anything other than 0(trap) or 1(s-curve), the instruction will report this error. |
| 139 | 3 | MCPM Dyn Data Invalid Value in MCPM Dyn Data Speed<br>If the speed is negative or INF or NAN, the instruction will report this error. |
| 139 | 4 | MCPM Dyn Data Invalid Value in MCPM Dyn Data Accel<br>If the accel is zero or negative or INF or NAN, the instruction will |

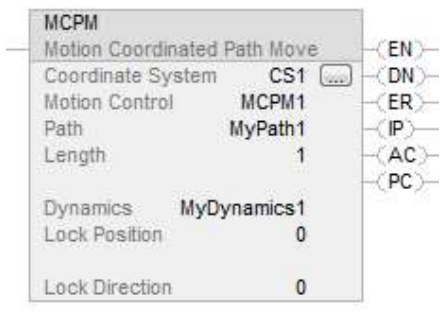
| | | |
|---|---|---|
| | | report this error. |
| 139 | 5 | MCPM Dyn Data Invalid Value in MCPM Dyn Data Decel<br>If the decel is zero or negative or INF or NAN, the instruction will report this error. |
| 139 | 6 | MCPM Dyn Data Invalid Value in MCPM Dyn Data Accel Jerk<br><br>If Jerk Units are %time and if the accel jerk is INF, NAN, less than or equal to zero or value greater than 100%, then the instruction will error.<br><br>**Tip:** These errors apply only for s-curve. |
| 139 | 7 | MCPM Dyn Data Invalid Value<br><br>MCPM Dyn Data Decel jerk<br><br>If Jerk Units are %time, and if the decel jerk is NAN and less than or equal to zero, then the instruction will report error.<br><br>**Tip:** This error only applies for s-curve. |
| 139 | 8 | MCPM Dyn Data Invalid Value<br><br>MCPM Dyn Data Orientation Speed<br><br>If the orientation speed is negative or INF or NAN, the instruction will report this error. |
| 139 | 9 | MCPM Dyn Data Invalid Value<br><br>MCPM Dyn Data Orientation Accel<br><br>If the orientation accel is zero or negative or INF or NAN, the instruction will report this error. |
| 139 | 10 | MCPM Dyn Data Invalid Value<br><br>MCPM Dyn Data Orientation Decel<br><br>If the orientation decel is zero or negative or INF or NAN, the instruction will report this error. |

# Diagnostic Codes and Corrective Actions

N/A

# Example

# Ladder Diagram




# Structured Text

MCPM (CS1, MCPM1, MyPath1, 1, MyDynamics1, 0, 0);

**Tip:** For further information on creating geometries with orientation support, see the Motion Coordinate System User Manual publication MOTION-UM002.

# See also

Structured Text Syntax

Index Through Arrays

Motion Error Codes (.ERR)

Motion Coordinated Transform with Orientation (MCTO)

Define coordinate system frames



How are we doing?