Soft Servo
SYSTEMS, INC

Reference Manual
for
ServoWorks CNC Parameters and Functions

Revision 3.56

Warning

The product described herein has the potential – through misuse, inattention, or lack of understanding – to create conditions that could result in personal injury, damage to equipment, or damage to the product(s) described herein. Machinery in motion and high-power, high-current servo drives can be dangerous; potentially hazardous situations such as runaway motors could result in death; burning or other serious personal injury to personnel; damage to equipment or machinery; or economic loss if procedures aren't followed properly. Soft Servo Systems, Inc. assumes no liability for any personal injury, property damage, losses or claims arising from misapplication of its products. In no event shall Soft Servo Systems, Inc. or its suppliers be liable to you or any other person for any incidental collateral, special or consequential damages to machines or products, including without limitation, property damage, damages for loss of profits, loss of customers, loss of goodwill, work stoppage, data loss, computer failure or malfunction claims by any party other than you, or any and all similar damages or loss even if Soft Servo Systems, Inc., its suppliers, or its agent has been advised of the possibility of such damages.

It is therefore necessary for any and all personnel involved in the installation, maintenance, or use of these products to thoroughly read this manual and related manuals and understand their contents. Soft Servo Systems, Inc. stands ready to answer any questions or clarify any confusion related to these products in as timely a manner as possible.

The selection and application of Soft Servo Systems, Inc.'s products remain the responsibility of the equipment designer or end user. Soft Servo Systems, Inc. accepts no responsibility for the way its controls are incorporated into a machine tool or factory automation setting. Any documentation and warnings provided by Soft Servo Systems, Inc. must be promptly provided to any end users.

This document is based on information that was available at the time of publication. All efforts have been made to ensure that this document is accurate and complete. However, due to the widely varying uses of this product, and the variety of software and hardware configurations possible in connection with these uses, the information contained in this manual does not purport to cover every possible situation, contingency or variation in hardware or software configuration that could possibly arise in connection with the installation, maintenance, and use of the products described herein. Soft Servo Systems, Inc. assumes no obligations of notice to holders of this document with respect to changes subsequently made. Under no circumstances will Soft Servo Systems, Inc. be liable for any damages or injuries resulting from any defect or omission in this manual.

Soft Servo Systems, Inc. makes no representation or warranty, expressed, implied, or statutory with respect to, and assumes no responsibility for the accuracy, completeness, sufficiency, or usefulness of the information contained herein. NO IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS OF PURPOSE SHALL APPLY.

Important Notice

Table of Contents

List of Tables

List of Figures

Chapter 1: ServoWorks CNC Parameters

1.1 Overview

Parameters are an integral part of all ServoWorks CNC systems. This reference manual is designed to guide you in setting the optimal NC and machine parameters, home parameters, motor and drive parameters, servo control parameters, compensation parameters, macro function parameters, etc. for your ServoWorks CNC system, to get the best performance from your machine tool.

PLEASE NOTE THAT SINCE THERE ARE MANY SERVOWORKS CNC PRODUCTS, YOUR CNC APPLICATION MAY NOT INCLUDE EVERY PARAMETER LISTED IN THIS DOCUMENT.

1.2 Backing Up Final Parameter Settings After Tuning

All current software settings, including ServoWorks CNC parameter settings, are contained in the Windows registry. Therefore, when you have set all of your ServoWorks CNC parameters and completed your installation, setup and integration (especially the important process of tuning your system), we highly recommend that you back up the GMC and Windows application folders in the HKEY_CLASSES_ROOT\ServoWorks key of the Windows registry. This will save you time and money if you need to recreate your optimized software environment for any reason. This is also useful for creating an identical control system with a new PC.

To back up your current parameter settings, you must export two folders to two Windows files (as explained in *Section 4.2: Exporting Current Parameter Settings to Windows Files* in the *Windows Registry Reference Manual for ServoWorks CNC Products and SMP Series General Motion Control Products*). [We recommend saving these files somewhere other than your PC.]

1.3 SDK Parameter Name Mapping

This manual covers both the parameters used by ServoWorks CNC HMI applications, and the parameters available through the ServoWorks Development Kit (SDK). The following tables map the ServoWorks CNC parameter names to the variables used in the ServoWorks Development Kit (SDK):

Parameter Type	Parameter Name	Corresponding Variable Name in SDK
Servo Control Parameters	Distance Per Encoder Revolution	DisPerEncRev
	Machine Unit (Minimum Resolution)	MachineUnit
	Over Position Error Protection Limit – Moving	OverPosErrServoOffLimit_Moving
	Over Position Error Protection Limit – Stopped	OverPosErrServoOffLimit_Stopped
	In Position Width	InPositionWidth
	Hard Limit Switch Action	LimitSwitchAction
	Smoothing Time – Cutting	SmoothingTime_Regular_Cutting
	Smoothing Time – Rapid	SmoothingTime_Regular_Rapid
	Smoothing Time – Manual	SmoothingTime_Regular_Manual
	Look Ahead Acc/Dec Time	SmoothingTime_LookAhead
	Position Loop Gain	PosLoopGain
	Position Loop Integral Control Enable	PosLoopIntEnable
	Position Loop Integral Time Constant	PosLoopIntTimeConst
	Position Loop Integral Saturation	PosLoopIntSaturation
	Velocity Feedforward Enable	VelocityFFEnable
	Velocity Feedforward Percentage	VelocityFFpct
	E-STOP Type	EStopType
	Limit Switch Type	LimitSwitchType
	Smoothing Mode – Cutting	SmoothingMode_Cutting
	Smoothing Mode – Rapid	SmoothingMode_Rapid
	Smoothing Mode – Manual	SmoothingMode_Manual

Table 1-1: Parameter Name Mapping (1 of 4)

Parameter Type	Parameter Name	Corresponding Variable Name in SDK
NC & Machine Parameters	Plus Stroke	PlusStroke
	Minus Stroke	MinusStroke
	Backlash Value	Backlash
	Pitch Origin	PitchOrigin
	Pitch Interval	PitchInterval
	Rapid Feedrate	RapidFeedrate
	Jog Feedrate	JogFeedrate
	Home Type	HomeType
	Home Switch Type	HomeSwitchType
	Home Direction	HomeDirection
	Home Position	HomePosition
	Home Shift	HomeShift
	Grid Search Speed	HomeSpeed
	Home Reverse Distance	HomeReverseDistance
	Home Reverse Dwell Time	HomeReverseDwellTime
	Axis Type	AxisType
	Always Search for Home	AlwaysSearchHome
	Rotary ST Rotating Type	RotarySingleTurnType
	Enable Integer Programming with Machine Unit	IntProgEnable
	Number of DC Modules	NoOfDC_Modules
	Number of IM Modules	NoOfIM_Modules
	Enable HandWheel / HandWheel Type	HandwheelType
	Accumulate HandWheel Pulse (HandWheel Pulse Accumulation)	HandwheelPulseAccumulation
	G00 Perform Linear Interpolation (Rapid Feed Type)	RapidFeedType
	Unit / Default Program Unit	DefaultProgramUnit
	Reference Position #2	RefPoint_2
	Reference Position #3	RefPoint_3

Table 1-2: Parameter Name Mapping (2 of 4)

Parameter Type	Parameter Name	Corresponding Variable Name in SDK
NC & Machine Parameters (continued)	Reference Position #4	RefPoint_4
	Home Switch Search Speed	HomeSwitchSearchSpeed
	Maximum Acceleration/Deceleration	MaxAccDecRate
	Look Ahead Smoothing Type	SmoothingType_LookAhead
	Look Ahead Smoothing Mode	SmoothingMode_LookAhead
	Look Ahead Smoothing Buffer Size	SmoothingBuffer_LookAhead
	Look Ahead Smoothing End Check Limit	SmoothingEndCheckLimit_LookAhead
	Look Ahead Smoothing Limit	SmoothingLimit_LookAhead
	Look Ahead Smoothing Factor	SmoothingFactor_LookAhead
	Retract Vector	RetractVector
	Shift Direction	ShiftDirection
	Sync Slave Axis Compensation Gain	SynSlaveCompGain
	Sync Master Axis	SynAssociatedToAxis
	Over Position Error Sync Limit – Moving	OverPosErrSynLimit_Rpd
	Over Position Error Sync Limit – Stopped	OverPosErrSynLimit_Stp
	Dry Run Feedrate	DryRunFeedrate
	Circle Error Allowance	CircleErrorLimit
	Corner Deceleration Control – Corner Deceleration	CornerDecEnabled
	Corner Deceleration Control – Corner Angle	CornerDecAngle
	Corner Deceleration Control – Corner Speed Limit	CornerDecSpeedLimit
	Corner Deceleration Control – Corner Tolerance Compensation Enable	CornerDecInPosEnabled
	Corner Deceleration Control – Corner Tolerance	CornerDecInPosLength
	G and M Code Order	GMCodeOrder
	Velocity Control in Circular Interpolation Enable	CirDecEnabled
	Velocity Control in Circular Interpolation – Maximum Acceleration	CirDecAccDec_Min

Table 1-3: Parameter Name Mapping (3 of 4)

Parameter Type	Parameter Name	Corresponding Variable Name in SDK
NC & Machine Parameters (continued)	Velocity Control in Circular Interpolation – Minimum Feedrate	CirDecSpeedLimit
	TCP – Arm Length (Axis 5)	TCPArmLen
	Home Switch Source	SwitchHomeInMode
	Limit Switch Source	SwitchLimitInMode
	Normal Direction Control – Rotary Axis No.	NormDirRotaryAxisNum
	Normal Direction Control – Rotation Feedrate	NormDirFeed
	Normal Direction Control – Angle Limit	NormDirAngleLmt
	Normal Direction Control – Length Limit	NormDirLengthLmt
Servo Drive Parameters	Rated Velocity	RatedVelocity
	Peak Velocity	PeakVelocity
	Encoder Resolution	EncoderResolution
	Encoder Polarity	EncoderPolarity
	Encoder Type	EncoderType
	Motor Polarity	MotorPolarity
	Servo Drive Velocity Sensitivity	DriveVelSensitivity
Macro Function Parameters	S Code Setting	SCall
	T Code Setting	TCall
	M Code Settings	MCall
	G Code Settings	GCall

Table 1-4: Parameter Name Mapping (4 of 4)

Chapter 2: Machine Parameters

2.1 Overview

The parameters included in this chapter are for axis configuration, I/O configuration, handwheel settings, tool center point settings, and home switch and limit switch source selection.

2.2 Axis Configuration Parameters

2.2.1 Axis Name

Description

The letter name to be used for the axis.

Valid Values: X, Y, Z, A, B, C, D, E, U, V, W

Default Values: X for Axis 1, Y for Axis 2, Z for Axis 3, A for Axis 4, B for Axis 5, and C for the spindle axis

Limitations

You can choose any unique value of the valid letters for each axis, except that Axis 4 of the first servo interface module (DC-155) must be used for a spindle, and that axis name must be either “S” or “C.”

PLC axes are named “P1,” “P2,” etc. and cannot be changed. Sync slave axes are named after their slave axis (i.e. the sync slave axis to Axis X would be named “X” also).

2.2.2 Axis Type

Description

The selection of axis control type or spindle drive type in your ServoWorks CNC system.

Valid Values for CNC Axes: Unused, Normal, Rotary, Rotary ST, Sync Slave, PLC Axis

Valid Values for Spindles: Unused, Servo Spindle, Inverter Spindle

Meaning of Values

Unused – The axis or spindle is not controlled by any means. The servo drive or spindle drive will never be enabled, and the all the control functions related to this axis will remain disabled.

Normal – The typical control type: a rotary motor geared or coupled to a ball screw system, driving a linear table

Rotary – A rotary motor driving a rotary mechanism, such as a rotary table

Rotary Single Turn (Rotary ST) – A rotary motor driving a rotary mechanism, such as a rotary table. When commanded by an absolute position (including homing), it will always turn within one rotation only (either from -180 to +180 or 0 to 360), and it will always choose the shortest path within that single turn to reach the absolute angular position. [For instance, a command of 1000 degrees (two full turns + 280 degrees) will result in a reverse direction turn of 80 degrees (1000 degrees - 360 degrees - 360 degrees = 280 degrees in one direction, so the shorter path of 80 degrees in the reverse direction).]

Sync Slave – An axis synchronized to another axis for dual axis control

PLC Axis – A PLC axis for individual, independent positioning; controlled by PLC functions and an axis control module

Servo Spindle – A servo motor and drive used as a spindle. A servo spindle axis type can be used for either open loop (S axis) spindle control (velocity control) by default, or closed loop (C axis) spindle control (position control) by using an M19 code. [Referred to in ServoWorks MotionLite as “Spindle_V”.] For open loop control, encoder feedback is not required – the encoder does not need to be connected.

Inverter Spindle – An inverter drive – for velocity control only, accepting positive voltages only – it can only be used for open loop spindle control. Its direction is controlled with PLC or a hardware port, as it can only send a unipolar (positive) signal.

Default Value: Unused

Notes

1) When you change the axis type to “Sync Slave,” you need to restart the ServoWorks CNC application (ServoWorks MC-Quad, ServoWorks S-100M, etc.) to synchronize the encoders of the master and slave axes, and to make synchronization control effective.

2) The “Rotary” and “Rotary ST” axis types will respond to position commands differently. The ending positions will be the same, but the movement to get there will be different. For example, if you command a position of 3600 degrees and start from 360 degrees, the “Rotary” axis will make 10 turns (10 x 360° = 3600°). The “Rotary ST” axis would not turn at all, as it is already at 360°.

3) The “Rotary” and “Rotary ST” axis types are different from a spindle axis, in that you can control the position. Only the speed and voltage of an inverter spindle can be controlled.

2.2.3 Rotary Position Display Range

Description

The position data display range for rotary axes.

Valid Values: 0, 1

Meaning of Values

0 – Display between 0.00000 ~ 359.99999 degrees
1 – Display between -179.99999 ~ 180.00000 degrees

Default Value: 0

Limits

The rotary position display range must be the same for coordinated master and slave axes when synchronous control is used.

Example

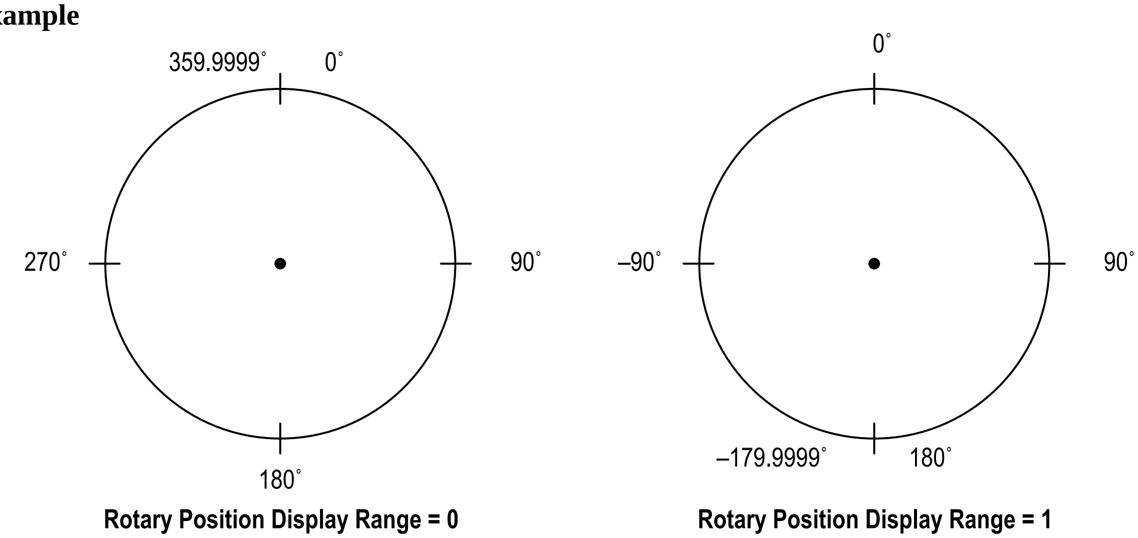


Figure 2-1: Rotary Position Display Range Example

2.2.4 Rotary ST Rotating Type

Description

The rotation type for rotary ST axes.

Valid Values: 0, 1

Meaning of Values

0 – Shorter – The rotary ST axis moves to the commanded absolute position taking the shortest path.
1 – Direction – The rotary ST axis moves to the commanded absolute position, taking the positive (clockwise) direction if the commanded absolute position is positive and taking the negative (counterclockwise) direction if the commanded absolute position is negative. Also, the commanded absolute position is interpreted as its absolute value in determining the target position (e.g. -60.0 degrees is interpreted as +60.0 degrees in determining the target position).

Default Value: 0

Example

With Rotary ST Rotating Type = 0, G00 X300. will cause the X axis to move backwards 60 degrees to reach the 300 degree position. This is because the shortest path to the 300 degree position is backwards 60 degrees.

With Rotary ST Rotating Type = 1, G00 X300. will cause the X axis to move forwards 300 degrees to reach the 300 degree position. This is because the positive commanded absolute position causes the axis to move forwards until it reaches the target position of 300 degrees.

With Rotary ST Rotating Type = 0, G00 X-300. will cause the X axis to move forwards 60 degrees to reach the -300 degree = +60 degree position. This is because the shortest path to the -300 degree position is forwards 60 degrees (since -300 degrees is equivalent to 60 degrees).

With Rotary ST Rotating Type = 1, G00 X-300. will cause the X axis to move backward 60 degrees to reach the 300 degree position. This is because the negative commanded absolute position causes the axis to move backwards until it reaches the target position of 300 degrees. The target position is 300 degrees because the absolute value of -300 degrees is 300 degrees.

2.3 I/O Configuration Parameters

2.3.1 Number of DC Modules

Description

Number of DC-155 servo interface modules used in the ServoWorks CNC system.

Range of Valid Values: 1 – 4

Default Value: 1

Note

This value applies to the VersioBus II interface system only.

2.3.2 Number of IM Modules

Description

Number of general I/O modules (IM-305s) used in the ServoWorks CNC system.

Range of Valid Values: 0 – 4

Default Value: 0

Note

This value applies to the VersioBus II interface system, or to any other interface system that includes optional VersioBus II I/O.

2.4 HandWheel Parameters

2.4.1 Enable HandWheel / HandWheel Type

Description

The type of handwheel used in the ServoWorks CNC system.

Valid Values: 0, 1, 2, 3, 4, 6

Meaning of Values

0 – None – no handwheel is used
1 – VersioBus II – a quadrature-encoder handwheel is used with a VersioBus II adapter card (HW-100 handwheel or another handwheel)
2 – ML Counter – a MECHATROLINK counter for encoder input for a handwheel
3 – SW MPG – Ormec IEEE 1394 servo drive encoder input for a handwheel
4 – FXI-40 handwheel
6 – EtherCAT Handwheel

Default Value: 0

2.4.2 Accumulate HandWheel Pulse (Handwheel Pulse Accumulation)

Description

Whether or not to accumulate the excessive handwheel pulses.

When the handwheel is in use, the rotation of the handwheel will generate the NC movement pulses to the commanded axis. If the speed of generated pulses exceeds the rapid feedrate of the commanded axis, the excessive pulses could be either accumulated or discarded.

Valid Values: 0, 1

Meaning of Values

0 – Do not accumulate the excessive handwheel pulses
1 – Accumulate excessive handwheel pulses

Default Value: 1

2.5 TCP Parameters

2.5.1 Arm Length (Axis 5)

Description

The arm length to use for TCP (tool center point) calculations.

Measured in Units of: mm or inches

Range of Valid Values: 0 – 999,999.9 mm or 0 – 999.999.9 inches

Default Value: 0.0 mm

Notes

See *Section 6.3.18 : Tool Center Point (TCP) Control / Control Cancel (G43.2, G49)* in the *Part Programming Manual for ServoWorks S-100M, S-120M and S-140M* for a discussion of TCP and how this parameter is used.

2.6 Home Switch and Limit Switch Source Selection Parameters

2.6.1 Home Switch Source

Description

The source of home switches in the network.

Valid Values: 0, 1

Meaning of Values

0 – On Servo – On servo home switches are used.
1 – PLC (G196.x) – PLC signals at address G196.x function as home switches.

Default Value: 0

2.6.2 Limit Switch Source

Description

The source of limit switches in the network.

Valid Values: 0, 1

Meaning of Values

0 – On Servo – On servo limit switches are used.
1 – PLC (G114.x, G116.x) – PLC signals at addresses G114.x and G116.x function as positive and negative limit switches, respectively.

Default Value: 0

Chapter 3: NC Settings Parameters

3.1 Overview

The parameters included in this chapter are for NC settings, canned cycle parameters and NC programming options.

3.2 NC Settings Parameters

3.2.1 Distance per Encoder Revolution

Description of Distance Per Encoder Revolution for CNC Axes

1) The linear movement (distance) produced by the actuator's (e.g. NC machine's) final output upon one complete revolution of the encoder

2) The linear movement (distance) produced by the actuator's (e.g. NC machine's) final output when the number of encoder pulses received is equal to encoder resolution

Description of Distance Per Encoder Revolution for Spindles (Spindle Rotation Per Encoder Revolution)

The spindle rotation angle per encoder revolution.

Measured in Units of: mm, inches or degrees

Range of Valid Values: 0 – 999,999.9 mm, 0 – 999,999.9 inches or 0 – 999,999.9 degrees

Default Value: 8.192 mm, 0.322520 inches or 8.192 degrees

3.2.2 Machine Unit (Minimum Resolution)

Description

The least input increment – the minimum unit of linear movement (distance) that the actuator (e.g. NC machine) can be commanded to move.

Measured in Units of: mm, inches or degrees

Range of Valid Values: 10^{-15} – 999,999.9 mm, 10^{-14} – 999,999.9 inches or 10^{-15} – 999,999.9 degrees

Default Value: 0.001 mm, 0.00003937 inches or 0.001 degrees

Usage

This parameter is mainly used for three purposes:

1) In ServoWorks MC-Quad Position Mode, it determines the minimum distance increment value

2) In HandWheel Mode, it determines the minimum distance increment value for each handwheel encoder pulse

3) In Auto Mode, it determines the unit of distance or position commanded in the NC program

3.2.3 In Position Width

Description

Half width of a window centered at the command position. When the actual position falls into this window, it is in position.

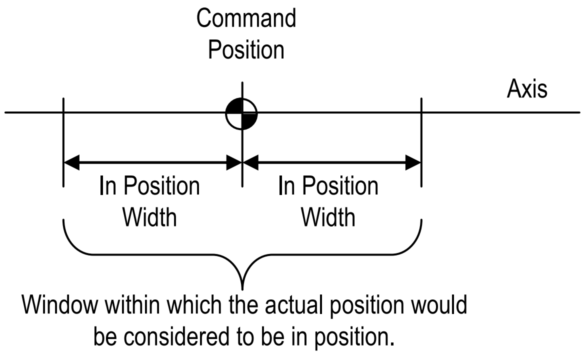


Figure 3-1: In Position Width

Measured in Units of: mm, inches or degrees

Range of Valid Values: 0 – 999,999.9 mm, 0 – 999,999.9 inches or 0 – 999,999.9 degrees

Default Value: 0.100 mm, 0.003937 or 0.100 degrees

Discussion

In position check is performed at the end of rapid movement G00 in the Auto mode, and also used for at home check after a home operation is completed.

Variations

In position width is measured in encoder counts for ServoWorks MotionLite.

3.2.4 Over Position Error Protection Limit – Moving

Description

The position error limit for an axis at which the Emergency Stop will be triggered while that axis is moving. (When the NC machine is in moving status, the emergency stop will be triggered if the position error exceeds this value.)

[NOTE: Position error is the difference between the command position and the actual position.]

Measured in Units of: mm, inches or degrees

Range of Valid Values: 0 – 999,999.9 mm, 0 – 999,999.9 inches or 0 – 999,999.9 degrees

Default Value: 10.000 mm, 0.393701 inches or 10.000 degrees

Variations

Over position error protection limit – moving is measured in encoder counts for ServoWorks MotionLite.

3.2.5 Over Position Error Protection Limit – Stopped

Description

The position error limit for an axis at which the emergency stop will be triggered while that axis is stopped (i.e. if the encoder feedback for that axis changes while the axis is stopped).

Measured in Units of: mm, inches or degrees

Range of Valid Values: 0 – 999,999.9 mm, 0 – 999,999.9 inches or 0 – 999,999.9 degrees

Default Value: 0.500 mm, 0.019685 inches or 0.500 degrees

Variations

Over position error protection limit – stopped is measured in encoder counts for ServoWorks MotionLite.

3.3 Canned Cycle Parameters

3.3.1 Shift Direction

Description

Axes shift direction during canned cycles (G76/G87). See the *Part Programming Manual for ServoWorks S-100M, S-120M and S-140M*.

Valid Values: 0, 1, –1

Meaning of Values

0 – No shift
1 – Shift in the positive direction
–1 – Shift in the negative direction

Default Value: 0

3.3.2 Retract Vector

Description

The retract distance during canned cycles, such as G73 and G83. See the *Part Programming Manual for ServoWorks S-100M, S-120M and S-140M*.

Measured in Units of: mm or inches

Range of Valid Values: 0 – 999,999.9 mm or 0 – 999,999.9 inches

Default Value: 7.000 mm or 0.275591 inches

Examples

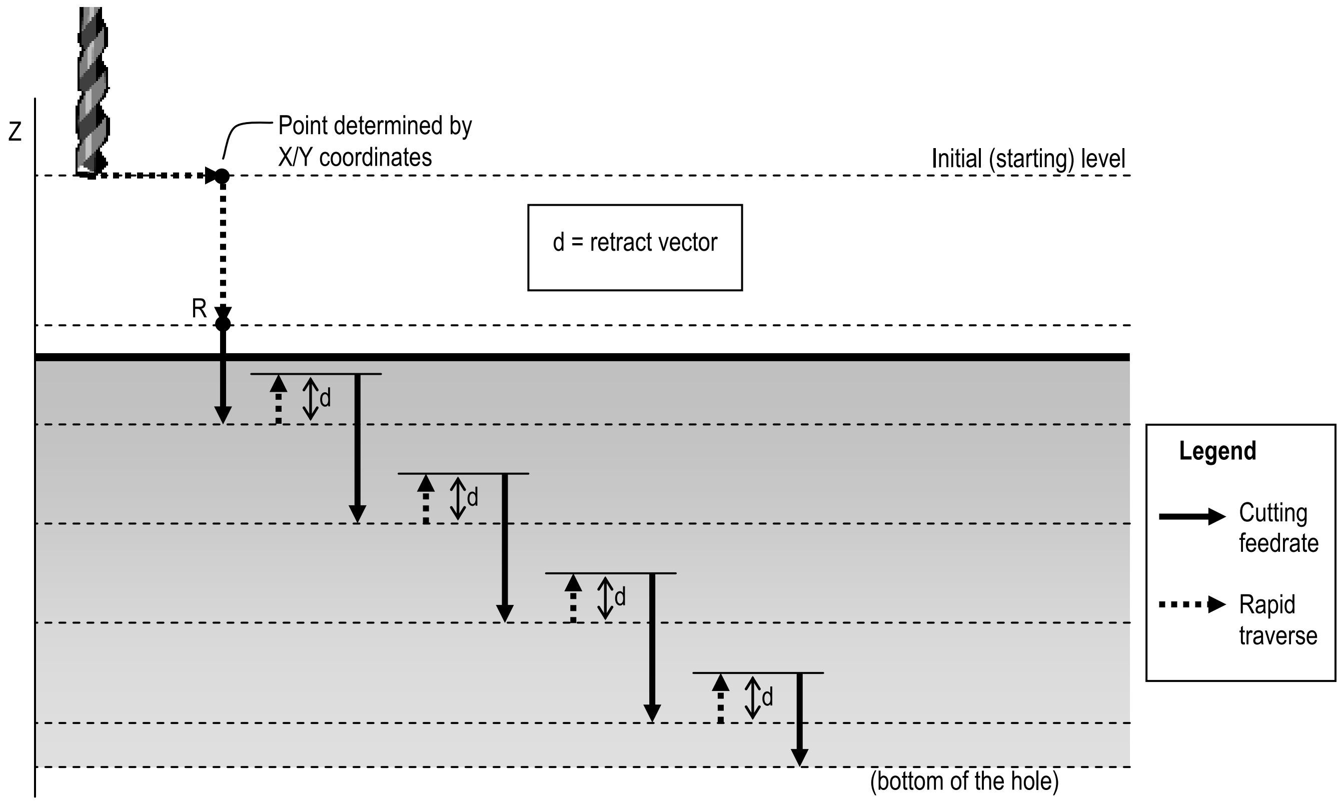


Figure 3-2: G73 Parameters

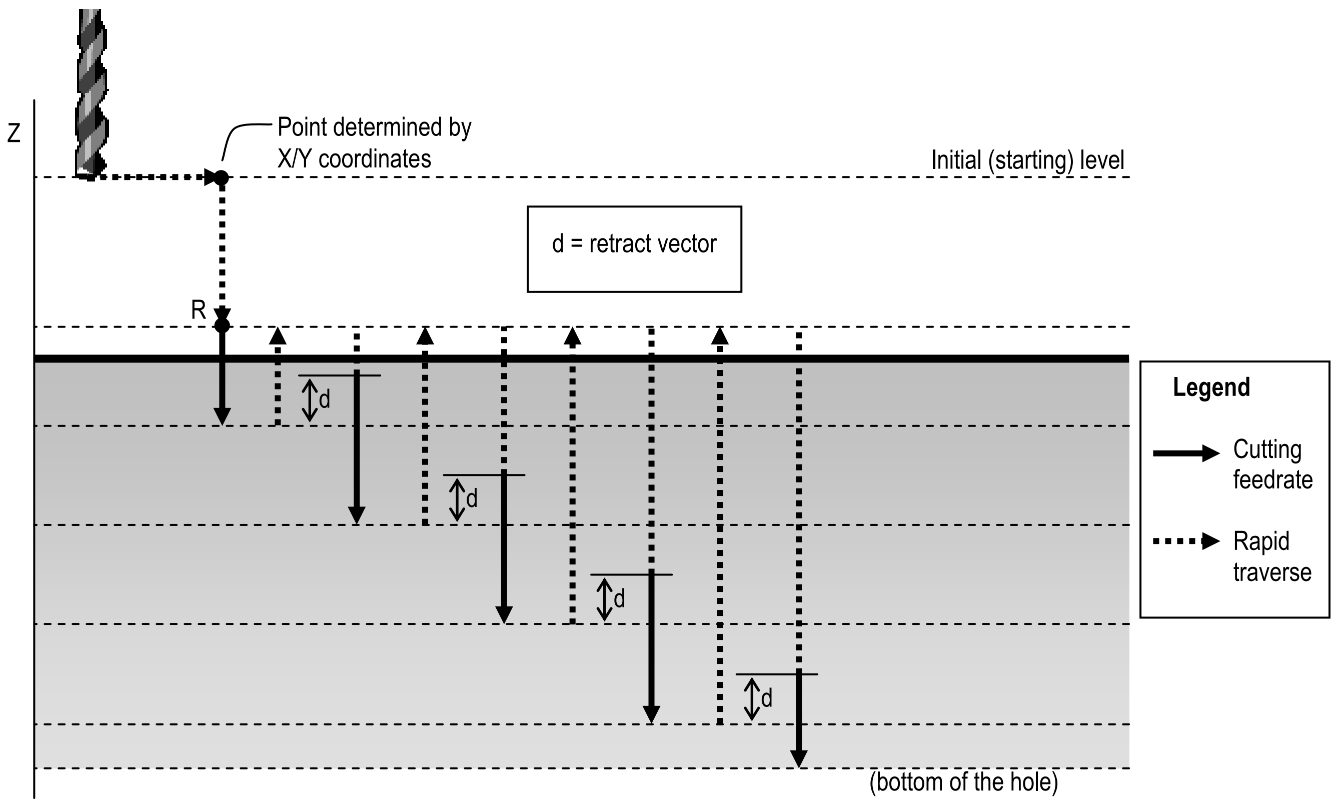


Figure 3-3: G83 Parameters

3.4 NC Programming Options

3.4.1 Enable Integer Programming with Machine Unit

Description

Enabling or disabling of integer programming with the machine unit.

When integer programming is disabled, both integer and floating-point values in the G-code program are treated as either mm or inch (depending on G20/G21 inch/metric setting in the NC program).

When integer programming is enabled, floating-point values are still treated as either mm or inch (depending on G20/G21 inch/metric setting in the NC program), but integer values are treated as the multiple of the "Machine Unit" parameter (in the "Servo Control Parameter" settings).

Please refer to the following table:

DATA TYPE	DATA UNIT WHEN INTEGER PROGRAMMING IS DISABLED	DATA UNIT WHEN INTEGER PROGRAMMING IS ENABLED
Integer	mm or inch	machine unit parameter setting
Floating-Point	mm or inch	mm or inch

Table 3-1: Integer Programming with Machine Unit Enable for Integer and Floating-Point Data Types

Valid Values: 0, 1

Meaning of Values

0 – Integer programming with the machine unit disabled
1 – Integer programming with the machine unit enabled

Default Value: 0

Example

With Integer Programming with Machine Unit Enable = 0, the "14" in "G00 X14 Y14.5" would be interpreted as 14.0 mm or 14.0 inches, and the "14.5" would be interpreted as 14.5 mm or 14.5 inches.

With Integer Programming with Machine Unit Enable = 1, the "14" in "G00 X14 Y14.5" would be interpreted as 14.0 machine units, and the "14.5" would still be interpreted as 14.5 mm or 14.5 inches. If the machine unit were set to 0.001 mm, then "X14" would be converted to 0.014 mm.

NOTE

This parameter may also be referred to as "Integer Programming with Machine Unit Enable."

3.4.2 G00 Perform Linear Interpolation (Rapid Feed Type)

Description

Rapid movement feedrate type. This parameter affects how G00 or G53 codes are executed.

When the machine is in rapid movement (commanded by G00 or G53), each commanded axis could either move in its rapid feedrate or move in a coordinated manner same as linear interpolation. In the latter case, the vector feedrate of the linear movement will be automatically maximized within the constraint of each axis' rapid feedrate.

Valid Values: 0, 1

Meaning of Values

0 – G00 perform linear interpolation disabled – G00 or G53 will be executed as normal rapid traverse, with no interpolation
1 – G00 perform linear interpolation enabled – G00 or G53 will be executed with linear interpolation at the highest possible speed (constrained by the rapid feedrate of each axis)

Default Value: 0

3.4.3 Unit / Default Program Unit

Description

The default NC program unit when neither G20 (inch programming) nor G21 (mm programming) is specified.

Valid Values: 0, 1

Meaning of Values

0 – mm programming
1 – Inch programming

Default Value: 0

3.4.4 G and M Code Order

Description

The order in which G and M codes in the same block of code are executed.

Valid Values: 0, 1, 2

Meaning of Values

0 – M -> G – M code is executed first.
1 – G -> M – G code is executed first.
2 – Same time – G code and M code are executed concurrently.

Default Value: 0

Notes

See *Section 5.2: Examples of MF/DEN Timing* in the *LadderWorks PLC Reference Manual* for an example of how this parameter affects the MF and DEN PLC signals and ultimately the tool path.

3.4.5 Circle Error Allowance

Description

The maximum difference between the specified target position and the calculated end position during circular interpolation before throwing an error.

Measured in Units of: mm or inches

Range of Valid Values: 0 – 999mm

Default Value: 0.1mm

Discussion

In a circular interpolation block, a target position is specified in addition to the radius or the center point. In many cases, the end point calculated from the specified radius or center point is not exactly equal to the specified target position because of the limited number of decimal digits one can use (or decides to use) to specify the radius or center point. This parameter defines a tolerance for this difference, below which the radius or center point is automatically adjusted to draw a clean arc that ends at exactly the target position.

Refer to *Section 6.3.3 : Circular Interpolation (G02, G03)* in the *Part Programming Manual for ServoWorks S-100M, S-120M and S-140M* for a more detailed discussion of this parameter.

3.4.6 Direct Dout Address

Description

The starting address of the 2-byte long region within the PLC output address space to control using Direct Digital Output.

Range of Valid Values: 80-98

Default Value: 80

Discussion

Direct Digital Output allows the user to control a 16-bit region of the PLC output address space using G code. This parameter is used to specify the starting address of the 2-byte region to use with Direct Digital Output. The range of PLC output address that can be used with this feature is Y80-Y98. As an example, if Direct Dout Address is set to 81, the output address space Y81.0~Y82.7 will be controlled by Direct Digital Output.

Refer to *Chapter 10: Direct Digital Output* in the *Part Programming Manual for ServoWorks S-100M, S-120M and S-140M* for more information on Direct Digital Output.

Chapter 4: Feedrate Settings Parameters

4.1 Overview

The parameters included in this chapter are for the various feedrates used by ServoWorks CNC applications.

4.2 Jog Feedrate

Description

Axis feedrate to be used in Jog Mode.

Measured in Units of: mm/min, inches/min or deg/min

Range of Valid Values: 0 – Rapid Feedrate

Default Value: 1,000.0 mm/min, 39.370079 inches/min or 1,000.0 deg/min

Limits

The jog feedrate must be the same for coordinated master and slave axes when synchronous control is used.

4.3 Rapid Feedrate

Description

Maximum allowable axis feedrate on the machine. This is the speed that will be used for rapid positioning (G00).

Measured in Units of: mm/min, inches/min or deg/min

Range of Valid Values: 0 – 999,999,999.9 mm/min, 0 – 999,999,999.9 inches or 0 – 999,999,999.9 deg/min

Default Value: 3,000.0 mm/min, 118.110236 inches/min or 3,000.0 deg/min

Limits

The rapid feedrate must be the same for coordinated master and slave axes when synchronous control is used.

4.4 Dry Run Feedrate

Description

The default dry run axis feedrate to be used in Auto Mode or MDI Mode when the Dry Run switch is selected.

Measured in Units of: mm/min or inches/min

Range of Valid Values: 0 – 999,999.9 mm/min or 0 – 999,999.9 inches/min

Default Value: 2,000.0 mm/min or 78.740157 inches/min

Note

This parameter does not apply to spindles.
Dry run feedrate is used while the dry run button is selected. Dry run feedrate immediately changes Jog or Rapid feedrate when dry run turns off.

Chapter 5: Motor/Servo Drive Parameters

5.1 Overview

The parameters included in this chapter are related to the servo motor and servo drive of each axis. The value of each parameter should match the actual specification of the servo motor or drive, as provided by the servo motor or drive manufacturer.

5.2 Rated Velocity

Description

Rated motor rotational speed (as specified by the motor manufacturer).

Measured in Units of: RPM (revolutions per minute)

Range of Valid Values: 0 – 999,999,999.9 RPM

Default Value: 3,000.0 RPM

Note

This parameter may also be referred to as "Rated Speed."

5.3 Peak Velocity

Description

Maximum allowable motor rotational speed (as specified by the motor manufacturer).

Measured in Units of: RPM (revolutions per minute)

Range of Valid Values: 0 – 999,999,999.9 RPM

Default Value: 4,500.0 RPM

Note

This parameter may also be referred to as "Peak Speed."

5.4 Encoder Resolution

Description

Number of encoder pulses per motor shaft revolution. (1 encoder pulse = 4 encoder counts for encoders with quadrature output).

Measured in Units of: pulses per revolution

Range of Valid Values: 1 – 999,999,999

Default Value: 2,048 pulses per revolution

⚠ CAUTION

For the MECHATROLINK interface system, the "Encoder Resolution" setting should be ¼ the value of the motor specification. For example, if the motor specifies encoder resolution as 8192 pulses/revolution, the "Encoder Resolution" parameter setting in the ServoWorks CNC software should be 8192 / 4 = 2048.

NOTE: You MUST divide the value of the motor specification by 4 when setting the "Encoder Resolution" parameter, or you will get unexpected results.

5.5 Encoder Polarity

Description

How the encoder feedback value is counted: in normal direction or in reversed (up/down) direction.

Valid Values: 1, -1

Meaning of Values

1 – Normal encoder value counting (positive)
–1 – Reversed encoder value counting (negative)

Default Value: 1

Discussion

Encoder polarity has to match motor polarity; otherwise the servo loop control will fail immediately.

Note

This parameter may also be referred to as "Encoder Direction Change."

5.6 Servo Drive Velocity Sensitivity

Description

Input/output sensitivity of the velocity mode servo drive.

$$\text{Servo Drive Velocity Sensitivity} = \frac{\text{Rated Velocity}}{\text{Positive Analog Input Range}}$$

The positive analog input range is the range of positive values of voltage input that will be accepted by the motor drive.

Measured in Units of: RPM per Volt

Range of Valid Values: 0 – 999,999,999.9 RPM per Volt

Default Value: 300.0 RPM per Volt

5.7 Motor Polarity

Description

How the analog command to the servo drive is sent: in normal polarity or reversed polarity.

Valid Values: 1, -1

Meaning of Values

1 – Normal analog command to the servo drive (positive)
-1 – Reversed analog command to the servo drive (negative)

Default Value: 1

Discussion

Motor polarity has to match encoder polarity; otherwise the servo loop control will fail immediately.

Note

This parameter may also be referred to as “DAC Direction Change,” where “DAC” refers to “Digital-to-Analog Converter.”

5.8 Encoder Type

Description

The type of encoder used in each axis.

Valid Values: 0, 1

Meaning of Values

0 – Incremental encoder
1 – Absolute encoder

Default Value: 0

Chapter 6: Servo Control Parameters

6.1 Overview

The parameters included in this chapter are related to the servo loop settings of each axis.

6.2 Position Loop Gain

Description

The position loop gain, shown in the following diagram as K_p:

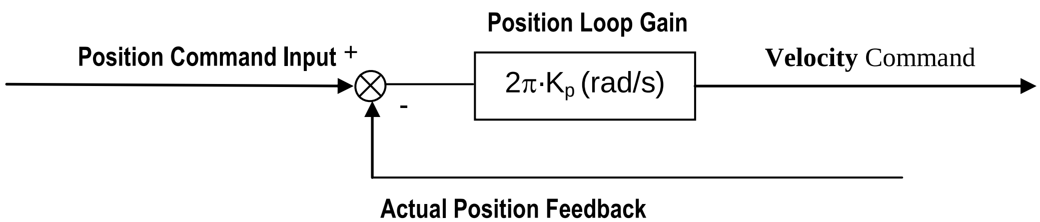


Figure 6-1: Position Loop Gain

Measured in Units of: Hz

Range of Valid Values: 0 – 500.0 Hz

Default Value: 20.0 Hz

Discussion

A higher value would result in stiffer (or firmer) position control, faster response and smaller servo lag, but would be more likely to cause vibration or oscillation.

Note

This parameter may also be referred to as "Position Loop Control Bandwidth."

CAUTION

For the MECHATROLINK interface system, there is a special consideration for setting the "Position Loop Gain" parameter. Please note that although the unit is "Hz", the actual value you need to set for SGDS series drives **must be 10 times the value in the unit of Hz** (because of the way the Sigma III series drives interpret the data). For example, if you'd like to set the position loop gain to 10 Hz, the value you need to set is 100 Hz. But for Sigma II series drives, the position loop gain setting should be the original value (if you'd like to set the position loop gain to 10 Hz, the value you need to set is 10 Hz).

NOTE: If you set the "Position Loop Gain" parameter without multiplying it by 10 for a Sigma III servo drive, you will get less precision position control and slower response.

6.3 Position Loop Integral Control Enable

Description

Enabling or disabling of position loop integral control.

Valid Values: 0, 1

Meaning of Values

0 – Position loop integral control disabled (Proportional control only)
1 – Position loop integral control enabled (Proportional-Integral control)

Default Value: 0

Discussion

Position loop integral control helps decrease or eliminate servo lag, but at the cost of possible overshooting and/or oscillation.

6.4 Position Loop Integral Time Constant

Description

The time constant which determines the response of position loop integral control, shown in the following diagram as T_i, where K_p is the position loop gain:

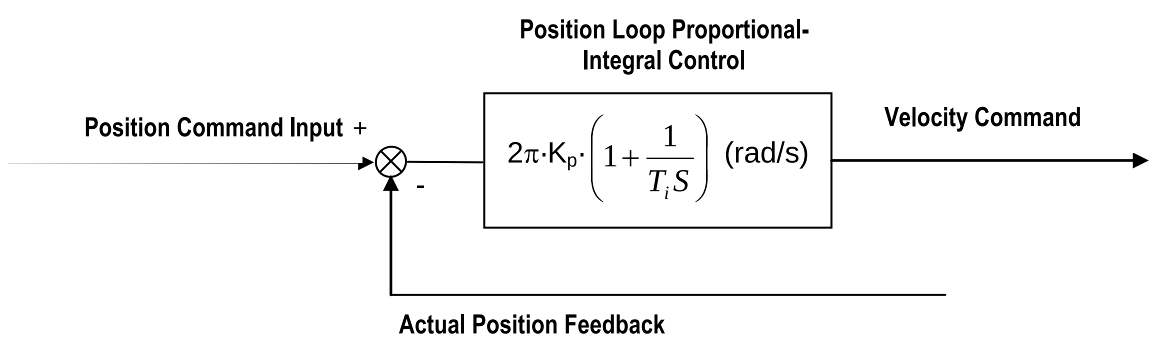


Figure 6-2: Position Loop Integral Time Constant

Measured in Units of: ms

Range of Valid Values: 0 – 999,999,999.9 ms

Default Value: 1,000,000.0 ms

Discussion

A lower value would result in faster response, but would be more likely to cause oscillation.

6.5 Position Loop Integral Saturation

Description

The maximum value for position error integration. Beyond this value, position error due to integration is disregarded.

Measured in Units of: mm, inches or degrees

Range of Valid Values: 0 – 999,999.9 mm, 0 – 999,999.9 inches or 0 – 999,999.9 degrees

Default Value: 1.000 mm, 0.039370 inches or 1.000 degrees

Discussion

A higher value would help minimize servo lag, but would lead to more significant overshooting.

Variations

Position loop integral saturation is measured in encoder counts for ServoWorks MotionLite.

6.7 Velocity Feedforward Enable

Description

Enabling or disabling of velocity feedforward (the process of using calculations to anticipate the likely value of axis velocity at some point in the immediate future, comparing the calculated velocity to the desired velocity, and using that information to influence the axis velocity).

Valid Values: 0, 1

Meaning of Values

0 – Velocity feedforward disabled
1 – Velocity feedforward enabled

Default Value: 0

Discussion

Velocity feedforward will decrease servo lag in the case when velocity command is constantly changing, e.g. in circular interpolation.

6.8 Velocity Feedforward Percentage

Description

The following equations explain the usage of this parameter.

$$\text{Velocity Feedforward} = \frac{\text{Position Command}[n] - \text{Position Command}[n-1]}{\text{Position Sampling Time}} * K$$

Where:

n = position loop sample counter

K = velocity feedforward percentage $*$ K′

K′ is a constant.

Measured in Units of: percent

Range of Valid Values: 0 – 100

Default Value: 0

Chapter 7: Smoothing Parameters (Acceleration/Deceleration)

7.1 Overview of Smoothing Theory

Smoothing is a way of specifying a velocity profile for servo motion to follow, since response to a command to move at a certain feedrate is not instant. Velocity profiles can be specified as linear, bell shaped or exponential. Illustrations of the three types of acceleration / deceleration smoothing filters follow:

7.1.1 Linear Smoothing

Where V_c is command velocity output and T is smoothing time.

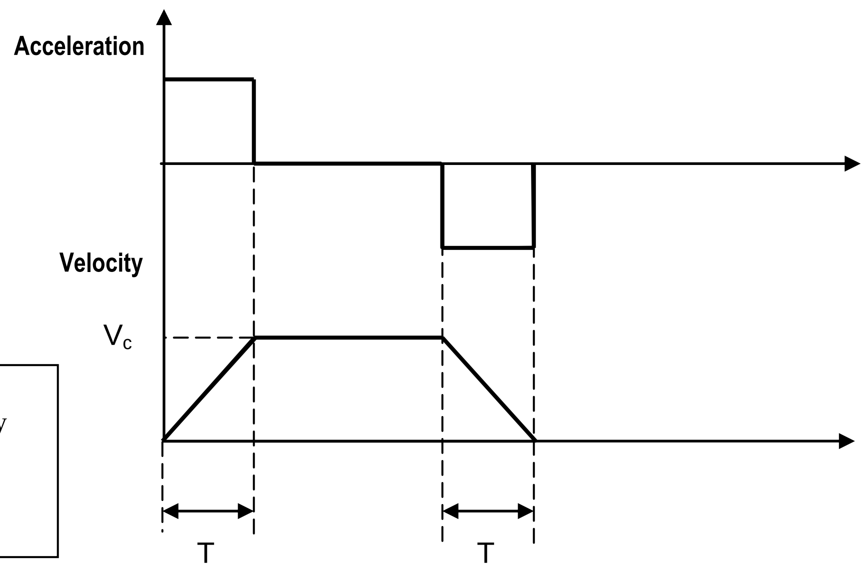


Figure 7-1: Linear Smoothing

7.1.2 Bell-Shaped Smoothing

Where V_c is command velocity output and T is the smoothing time.

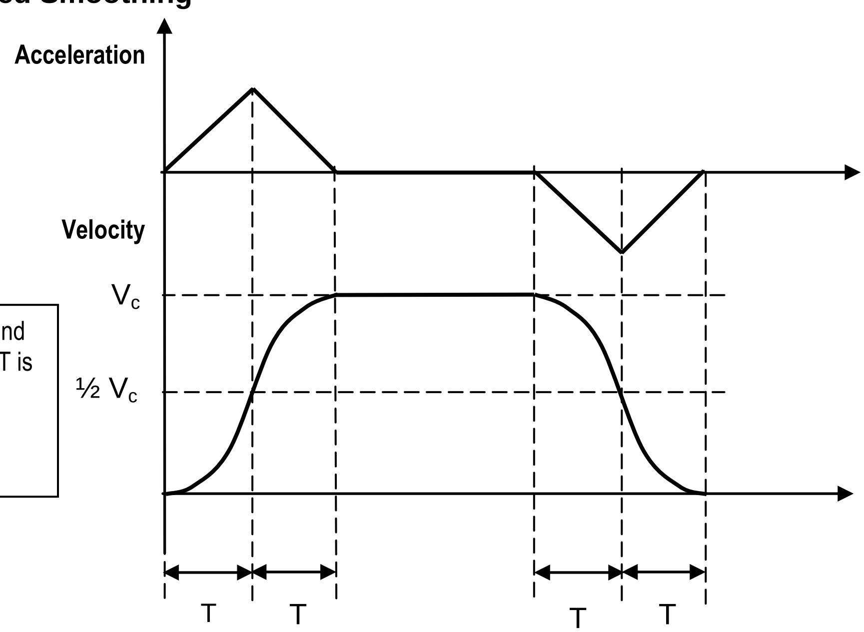


Figure 7-2: Bell-Shaped Smoothing

7.1.3 Exponential Smoothing

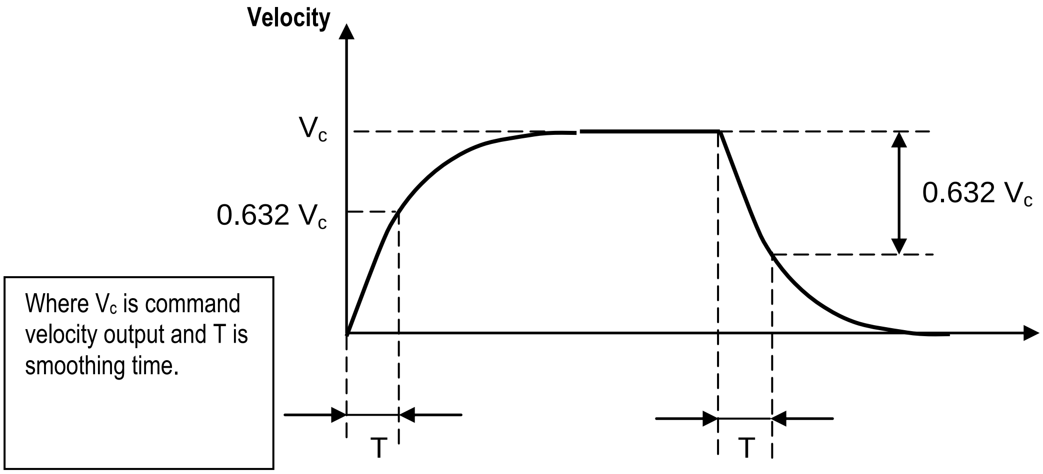


Figure 7-3: Exponential Smoothing

7.2 Smoothing Parameters

7.2.1 Smoothing Time – Cutting

Description

The motion profile smoothing filter time for cutting (G01, G02, G03, G02.3 or G03.3).

Measured in Units of: ms

Range of Valid Values: 0 – 5,000 ms

Default Value: 20 ms

NOTE

This parameter may also be referred to as “Smoothing Constant” or “Regular Smoothing Time.”

Limits

The smoothing time – cutting must be the same for coordinated master and slave axes when synchronous control is used.

7.2.2 Smoothing Time – Rapid

Description

The motion profile smoothing filter time for G00 and for Rapid Mode.

Measured in Units of: ms

Range of Valid Values: 0 – 5,000 ms

Default Value: 20 ms

NOTE

This parameter may also be referred to as "Smoothing Constant" or "Regular Smoothing Time."

Limits

The smoothing time – rapid must be the same for coordinated master and slave axes when synchronous control is used.

7.2.3 Smoothing Time – Manual

Description

The motion profile smoothing filter time for the following manual modes: Continuous Jog Mode, Incremental Jog Mode, Position Mode, HandWheel Mode and Home Mode.

Measured in Units of: ms

Range of Valid Values: 0 – 5,000 ms

Default Value: 20 ms

NOTE

This parameter may also be referred to as "Smoothing Constant" or "Regular Smoothing Time."

Limits

The smoothing time – manual must be the same for coordinated master and slave axes when synchronous control is used.

7.2.4 Smoothing Mode – Cutting

Description

The motion profile smoothing mode for cutting (G01, G02, G03, G02.3 or G03.3).

Valid Values: 0, 1, 2, 3

Meaning of Values

0 – No smoothing
1 – Linear smoothing
2 – Bell-shaped smoothing
3 – Exponential smoothing

Default Value: 3

Limits

The smoothing mode – cutting must be the same for coordinated master and slave axes when synchronous control is used.

Notes

See *Section 7.1: Overview of Smoothing Theory* for illustrations of the meaning of values 1, 2 and 3 (linear, bell-shaped and exponential smoothing).

7.2.5 Smoothing Mode – Rapid

Description

The motion profile smoothing mode for G00 and for Rapid Mode.

Valid Values: 0, 1, 2, 3

Meaning of Values

0 – No smoothing
1 – Linear smoothing
2 – Bell-shaped smoothing
3 – Exponential smoothing

Default Value: 1

Limits

The smoothing mode – rapid must be the same for coordinated master and slave axes when synchronous control is used.

Notes

See *Section 7.1: Overview of Smoothing Theory* for illustrations of the meaning of values 1, 2 and 3 (linear, bell-shaped and exponential smoothing).

7.2.6 Smoothing Mode – Manual

Description

The motion profile smoothing mode for the following manual modes: Continuous Jog Mode, Incremental Jog Mode, Position Mode, HandWheel Mode and Home Mode.

Valid Values: 0, 1, 2, 3

Meaning of Values

0 – No smoothing
1 – Linear smoothing
2 – Bell-shaped smoothing
3 – Exponential smoothing

Default Value: 1

Limits

The smoothing mode – manual must be the same for coordinated master and slave axes when synchronous control is used.

Notes

See *Section 7.1: Overview of Smoothing Theory* for illustrations of the meaning of values 1, 2 and 3 (linear, bell-shaped and exponential smoothing).

Chapter 8: Safety Parameters

8.1 Overview

The parameters included in this chapter are for hard limits, emergency stop and soft limits.

8.2 Safety Parameters

8.2.1 Limit Switch Type

Description

The type of limit switch in the ServoWorks CNC system.

Valid Values: 0, 1

Meaning of Values

0 – Active closed – the limit switch has "On" status when the switch is closed, and "Off" status when the switch is open
1 – Active open – the limit switch has "On" status when the switch is open, and "Off" status when the switch is closed

Default Value: 0

Notes

For safety reasons, it is recommended that you use the "Active open" type of limit switch, because it could detect malfunctions such as a broken wire, etc.

8.2.2 Hard Limit Switch Action

Description

The action to be taken when any of the hardware limit switches is triggered.

Valid Values: 0, 1

Meaning of Values

0 – Emergency Stop (shut down servo drives)
1 – Stop all motion (but do not shut down servo drives)

Default Value: 0

8.2.3 E-STOP Type

Description

Determines the behavior of each axis when Emergency Stop is triggered.

Valid Values: 0, 1

Meaning of Values

0 – Servo Off: When the Emergency Stop is triggered, the servo drive for this axis will be immediately turned off
1 – Dec/Servo Off: When the Emergency Stop is triggered, the axis will decelerate until velocity command becomes zero, then the servo drive for this axis will be turned off

Default Value: 0

8.3 Soft Limit Parameters

8.3.1 Plus Stroke

Description

Soft limit (software stroke limit) for positive axis travel. The plus stroke is effective only after a home operation has completed, because the home operation sets the machine origin from which the plus stroke is measured.

Measured in Units of: mm, inches or degrees

Range of Valid Values: 0 – 999,999.9 mm, 0 – 999,999.9 inches or 0 – 999,999.9 degrees

Default Value: 1,000.0 mm, 39.370079 inches or 1,000.0 degrees

Limits

- The plus stroke value must be greater than the home position value. Therefore, if the home position is set to zero, the plus stroke value would need to be a positive value.
- The plus stroke must be the same for coordinated master and slave axes when synchronous control is used.
- The plus stroke value must be greater than the minus stroke value.

Warning

If the “plus stroke” is set to “0,” this means that the positive soft limit of the axis will be invalid – there will be no limitation to travel from the software.

8.3.2 Minus Stroke

Description

Soft limit (software stroke limit) for negative axis travel. The minus stroke is effective only after a home operation has completed, because the home operation sets the machine origin from which the minus stroke is measured.

Measured in Units of: mm, inches or degrees

Range of Valid Values: -999,999.9 – 0 mm, -999,999.9 – 0 inches or -999,999.9 – 0 degrees

Default Value: -1,000.0 mm, -39.370079 inches or -1,000.0 degrees

Limits

- The minus stroke must be the same for coordinated master and slave axes when synchronous control is used.
- The minus stroke value must be less than the plus stroke value.

Warning

If the “minus stroke” is set to “0,” this means that the negative soft limit of the axis will be invalid – there will be no limitation to travel from the software.

Chapter 9: Home Parameters

9.1 Overview

This chapter describes the parameters related to the homing operation and reference positions, and provides time charts for several homing operation examples.

9.2 Home Type

Description

The method of finding the home position used for a home operation.

Valid Values: 0, 1, 2, 3, 4, 5, 6

Meaning of Values

0 – "Z Pulse" – Find the nearest grid

1 – "HSOffZP" – High-speed (at the home switch search speed) positioning until reaching the home switch, then decelerate to grid search speed and move at that speed until the home switch is untripped (is back to normal), then find the nearest grid

2 – "HSOnZP" – High-speed (at the home switch search speed) positioning until reaching the home switch, then decelerate to grid search speed and find the nearest grid

3 – "HSRevZP" – High-speed (at the home switch search speed) positioning until reaching the home switch, then decelerate to grid search speed; move the "home reverse distance" (a specified distance that defines a starting point to reverse direction); dwell for a specified time (defined by the "home reverse dwell time" parameter); reverse direction until the home switch is untripped (is back to normal) and find the nearest grid

4 – "LSRevZP" – High-speed (at the home switch search speed) positioning until reaching the limit switch, then decelerate to zero, reverse direction and accelerate to the grid search speed; move at the grid search speed until the limit switch is untripped (back to normal) and find the nearest grid

5 – "HSOn" – Home switch on: high-speed (at the home switch search speed) positioning until reaching the home switch, then decelerate to the grid search speed and apply the home shift.

6 – "CurPos" – Make the current position the home position (i.e. wherever the axis is when the home button is pushed becomes the home position – the machine doesn't move during this homing operation). This applies to the home button in the software (i.e. "Home Axis 1"), or the "Home" button on the operator's panel.

Default Value: 0

Limits

The home type must be the same for coordinated master and slave axes when synchronous control is used.

9.3 Home Direction

9.3.1 Home Direction for CNC Axes

Description

The direction that the machine initially moves during a home operation.

Valid Values: 1, -1

Meaning of Values

1 – Home to the positive direction
-1 – Home to the negative direction

Default Value: 1

Limits

The home direction must be the same for coordinated master and slave axes when synchronous control is used.

9.3.2 Home Direction for Spindles (Orientation Direction)

Description

The direction of spindle rotation for the spindle orientation operation.

Valid Values: 1, -1

Meaning of Values

1 – Clockwise
-1 – Counterclockwise

Default Value: 1

9.4 Home Switch Type

Description

This parameter specifies whether the home switch is electrically or mechanically configured to close or open when activated.

Valid Values: 0, 1

Meaning of Values

0 – Active closed (i.e. normal state is open, switch closes when activated)
1 – Active open (i.e. normal state is closed, switch opens when activated)

Default Value: 1

9.5 Home Shift

Description of Home Shift for CNC Axes

The distance commanded after the Z-Pulse is found.

Description of Home Shift for Spindles (Orientation Shift)

The angle from the spindle encoder Z-pulse position to the orientation end position.

Measured in Units of: mm, inches or degrees

Range of Valid Values: -999,999.9 – 999,999.9 mm, -999,999.9 – 999,999.9 inches or -999,999.9 – 999,999.9 degrees

Default Value: 1.000 mm, 0.039370 inches or 1.000 degrees

9.6 Home Position

Description

Desired machine position when a home operation is complete, i.e. the machine is at the home position. (In other words, the machine coordinates you want to set for the physical location of the home position.) Because the home position is relative to the machine origin, setting the home position also sets the machine origin.

Measured in Units of: mm, inches or degrees

Range of Valid Values: -999,999.9 – 999,999.9 mm, -999,999.9 – 999,999.9 inches or -999,999.9 – 999,999.9 degrees

Default Value: 0.000 mm, 0.000 inches or 0.000 degrees

Limits

The home position must be the same for coordinated master and slave axes when synchronous control is used.

9.7 Home Reverse Dwell Time

Description

This command specifies a delay in the homing procedure.

This parameter only applies when the "Home Type" value is 3 (HS Rev ZP, or "Home Switch Reverse Z Pulse"). As part of the homing procedure for this home type, the axis dwells for a specified time after reversing direction upon tripping the home switch. See the homing chart in *Section 9.16.9: Home Type = 3, Home Direction = 1, Home Shift < 0* for more information.

Measured in Units of: milliseconds

Range of Valid Values: 0 – 9,999 ms

Default Value: 100 ms

Limits

This parameter does not normally apply to any Home Type except Home Type #3 (HS Rev ZP, or "Home Switch Reverse Z Pulse"). However, this parameter is also used for Home Type #2 and Home Type #5 when a homing procedure is initiated and the current machine position is at the home switch – see *Section 9.15.2: When the Starting Location for a Homing Procedure is On the Home Switch* for more information.

9.8 Home Reverse Distance

Description

This command specifies a starting point for a reversal in the direction of axis movement, for a specified reverse distance past reaching the home switch, as part of the homing procedure. This parameter only applies when the "Home Type" value is 3 (HS Rev ZP, or "Home Switch Reverse Z Pulse"). See the homing chart in *Section 9.16.9: Home Type = 3, Home Direction = 1, Home Shift < 0* for more information.

Measured in Units of: mm, inches or degrees

Range of Valid Values: 0 –999.9 mm, 0 – 39.366 inches or 0 –999.9 degrees

Default Value: 1.0 mm, 0.039370 inches or 1.0 degrees

Limits

This parameter does not normally apply to any Home Type except Home Type #3 (HS Rev ZP, or "Home Switch Reverse Z Pulse"). However, this parameter is also used for Home Type #2 and Home Type #5 when a homing procedure is initiated and the current machine position is at the home switch – see *Section 9.15.2: When the Starting Location for a Homing Procedure is On the Home Switch* for more information.9.9 Grid Search Speed

Description of Home Feedrate for CNC Axes

This is the feedrate that will be used for the home operation, for moving the axes when searching for the grid (i.e. "low speed homing").

Description of Home Feedrate for Spindles (Orientation Speed)

The rotation speed of the spindle for the spindle orientation operation.

Measured in Units of: mm/min, inches/min or deg/min

Range of Valid Values: 0 – Rapid Feedrate

Default Value: 300.0 mm/min, 11.811024 inches/min or 300.00 deg/min

Limits

The home feedrate must be the same for coordinated master and slave axes when synchronous control is used.

Notes

This parameter is sometimes referred to as "home feedrate."

9.10 Home Switch Search Speed

Description

The speed at which the axes move when searching for the home switch, when performing a long distance home shift (greater than the distance per encoder revolution), and when performing a reference point return (i.e. "high speed homing").

Measured in Units of: mm/min, inches/min or deg/min

Range of Valid Values: 0 – Rapid Feedrate

Default Value: 1,000.0 mm/min, 39.370079 inches/min or 1,000.00 deg/min

Limits

The home switch search speed must be the same for coordinated master and slave axes when synchronous control is used.

9.11 Reference Position #2

Description

The second reference point (home) position.

Measured in Units of: mm, inches or degrees

Range of Valid Values: -999,999.9 – 999,999.9 mm, -999,999.9 – 999,999.9 inches or -999,999.9 – 999,999.9 degrees

Default Value: 0.000 mm, 0.000 inches or 0.000 degrees

Limits

Reference position #2 must be the same for coordinated master and slave axes when synchronous control is used.

9.12 Reference Position #3

Description

The third reference point (home) position.

Measured in Units of: mm, inches or degrees

Range of Valid Values: -999,999.9 – 999,999.9 mm, -999,999.9 – 999,999.9 inches or -999,999.9 – 999,999.9 degrees

Default Value: 0.000 mm, 0.000 inches or 0.000 degrees

Limits

Reference position #3 must be the same for coordinated master and slave axes when synchronous control is used.

9.13 Reference Position #4

Description

The fourth reference point (home) position.

Measured in Units of: mm, inches or degrees

Range of Valid Values: -999,999.9 – 999,999.9 mm, -999,999.9 – 999,999.9 inches or -999,999.9 – 999,999.9 degrees

Default Value: 0.000 mm, 0.000 inches or 0.000 degrees

Limits

Reference position #4 must be the same for coordinated master and slave axes when synchronous control is used.

9.14 Always Search for Home

Description

Whether or not to perform a full homing operation for subsequent homing operations.

Valid Values: 0, 1

Meaning of Values

0 – Auto – perform a full homing operation (find the home switch, find the Z pulse, establish the coordinate system) the first time a homing operation is commanded; for subsequent homing operations, move to the home position without finding the home switch, Z pulse, etc.

1 – Always– perform a full homing operation (find the home switch, find the Z pulse, establish the coordinate system) every time a homing operation is commanded

Default Value: 0

Notes

This parameter is sometimes referred to as "perform home search."

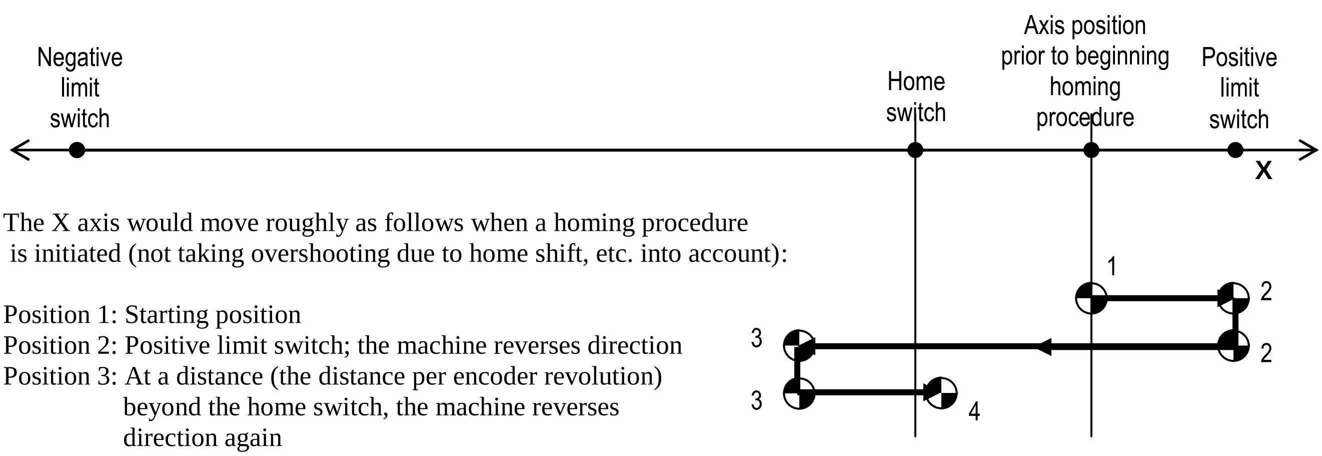
9.15 Homing Procedure Protections

9.15.1 When the Starting Location for a Homing Procedure is Between the Home Switch and a Limit Switch, with the Home Direction Towards the Limit Switch

The machine is set to move in a given initial direction during a homing operation (set by the "Home Direction" parameter – see *Section 9.3: Home Direction*). If the current axis position is between the home switch and a limit switch, AND the Home Direction sets the initial direction such that the axis starts the homing procedure by moving away from the home switch, then the axis will reach a limit switch before it reaches a home switch.

There is a built-in procedure to avoid causing an Emergency Stop when the limit switch is reached (which would require operator input to reset and continue). Instead, the homing procedure is designed to automatically reverse direction, go past the home switch, and start the homing procedure again. With this complete homing procedure, the home position is always found.

For example, assume the current machine setup for the X axis is as follows, and the "Home Direction" is "1" (home to the positive direction):



The X axis would move roughly as follows when a homing procedure is initiated (not taking overshooting due to home shift, etc. into account):

Position 1: Starting position
Position 2: Positive limit switch; the machine reverses direction
Position 3: At a distance (the distance per encoder revolution) beyond the home switch, the machine reverses direction again
Position 4: Home position (varies depending upon the specified "Home Type" and the Z pulse location)

Figure 9-1: Example of When the Starting Location for a Homing Procedure is Between the Home Switch and a Limit Switch, with the Home Direction Towards the Limit Switch

9.15.2 When the Starting Location for a Homing Procedure is On the Home Switch

When the starting location for a homing procedure is exactly at the home switch, there is a built-in protection that saves time and avoids the procedure described in *Section 9.15.1: When the Starting Location for a Homing Procedure is Between the Home Switch and a Limit Switch, with the Home Direction Towards the Limit Switch*.

When the starting location for a homing procedure is exactly at the home switch, the homing procedure is designed to automatically reverse direction (from the specified "Home Direction," and to move in the reverse direction at the grid search speed until the machine is past the home switch (the home switch is untripped), then the machine continues to move the distance specified by the "Home Reverse Distance" parameter. The machine dwells for the length of time specified by the "Home Reverse Dwell Time" parameter, then initiates the homing procedure from this new position. This saves time by avoiding the need to move the machine all the way to the limit switch and back.

<u>NOTE</u>: This protection only applies to Home Types 2 and 5.

9.16 Time Charts of the Home Operation for Different Home Type, Home Direction and Home Shift Settings

The following time charts show the homing operations for different combinations of home type, home direction and home shift settings.

Of particular interest are two settings:

1) Home shift = 0

   For this setting of a zero home shift, the machine must actually overshoot the Z-pulse by ¼ of the distance per encoder revolution (enough distance to eliminate backlash and ensure contact between the threads of the leadscrew and the gear), and then return in the opposite direction to come to a stop back at the Z pulse. [NOTE: See *Section 3.2.1: Distance per Encoder Revolution* for more information on the “distance per encoder revolution” parameter.]

2) 0 < Home shift < deceleration distance

   The deceleration distance from movement at the home feedrate to a complete stop is dependent on the smoothing time:

   deceleration distance = ½ (home feedrate)(smoothing time)

   For this setting of a home shift that is greater than zero, but less than the deceleration distance, the machine must actually overshoot the Z-pulse by the home shift distance plus ¼ of the distance per encoder revolution (see *Section 3.2.1: Distance Per Encoder Revolution*), and then return in the opposite direction to come to a stop at the distance from the Z-pulse specified by the home shift setting.

Please note that some of the following graphs plot velocity against time, not against position; other graphs plot position against time. Pay attention to axes labels.

9.16.1 Home Type = 0, Home Direction = 1, Home Shift = 0

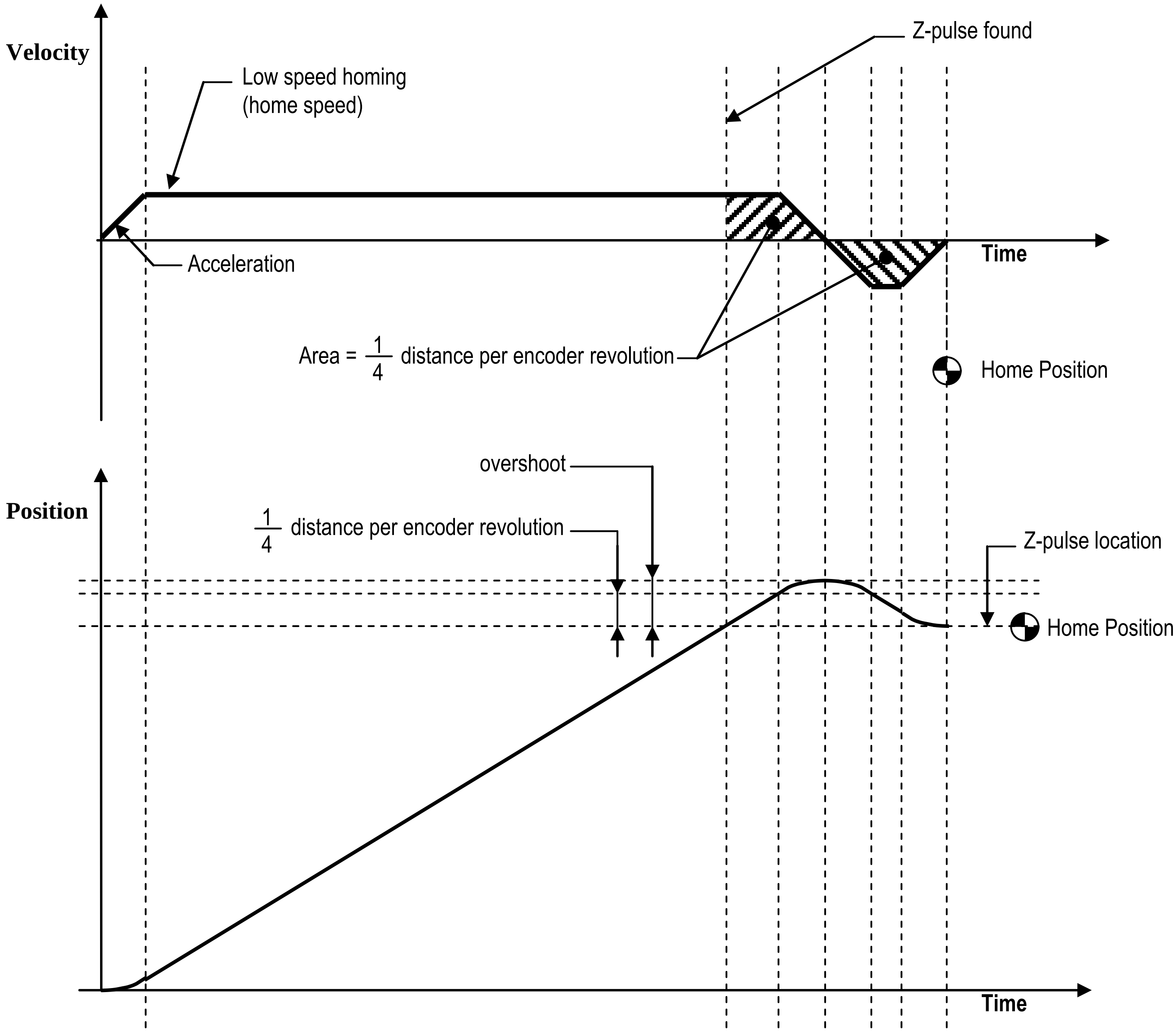


Figure 9-2: Home Operation Time Chart Example #1

9.16.2 Home Type = 0, Home Direction = 1, Home Shift > 0 and Home Shift < Deceleration Distance

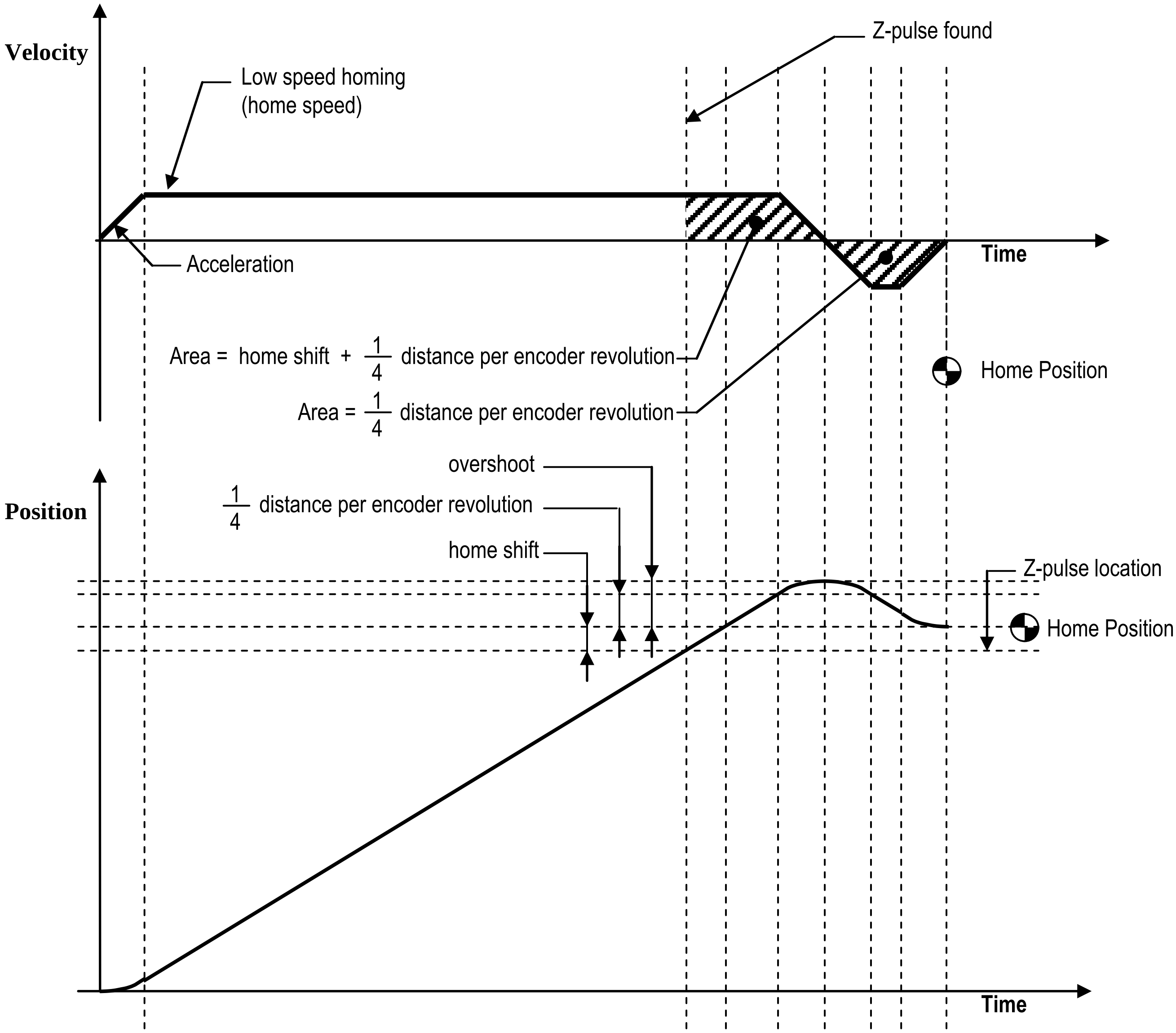


Figure 9-3: Home Operation Time Chart Example #2

9.16.3 Home Type = 0, Home Direction = 1, Home Shift > 0 and Home Shift < Distance per Encoder Revolution

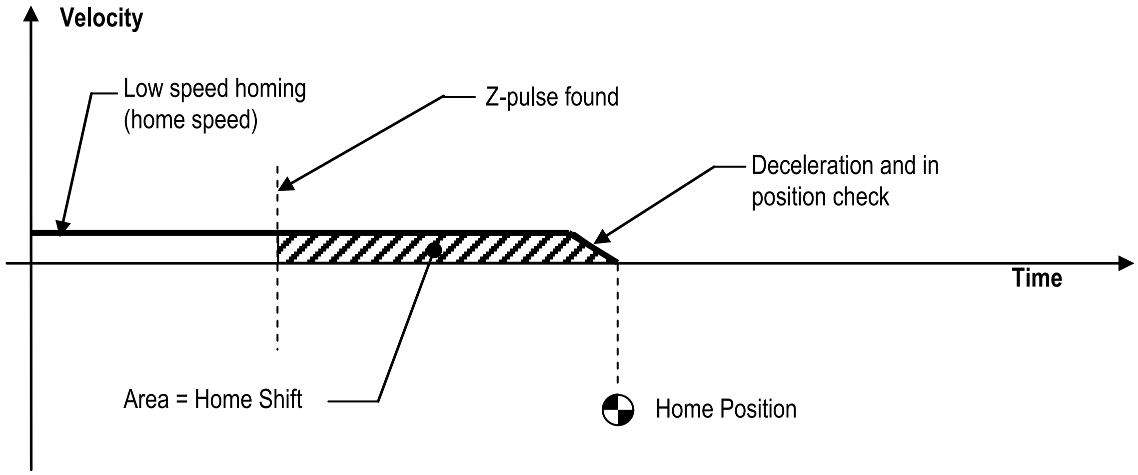


Figure 9-4: Home Operation Time Chart Example #3

9.16.4 Home Type = 0, Home Direction = 1, Home Shift > 0 and Home Shift > Distance per Encoder Revolution

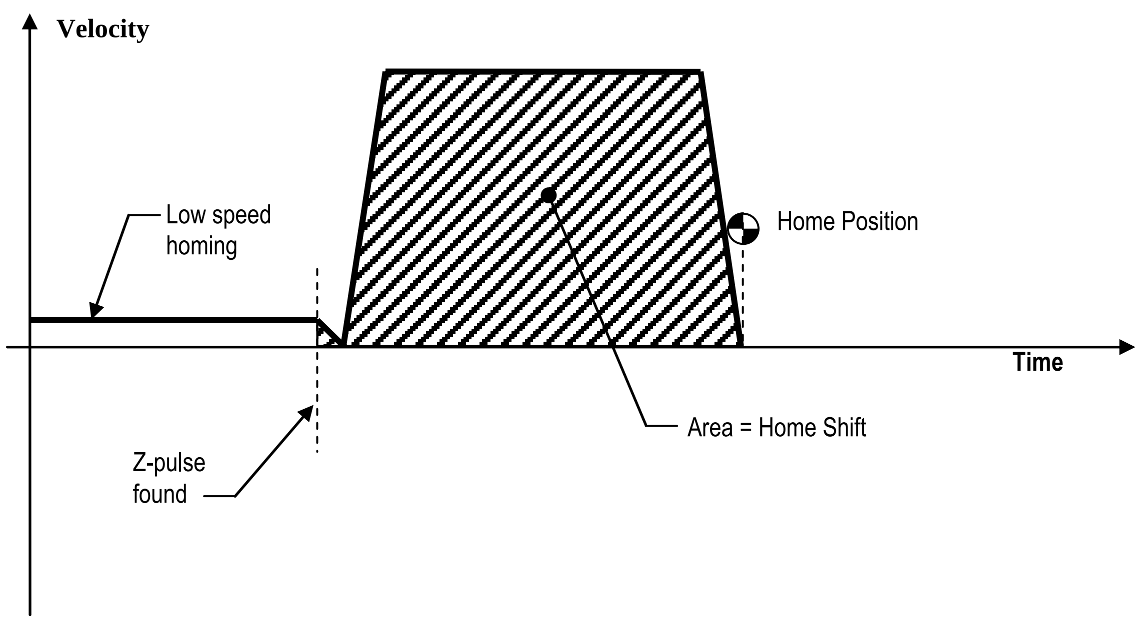


Figure 9-5: Home Operation Time Chart Example #4

9.16.5 Home Type = 0, Home Direction = 1, Home Shift < 0 and |Home Shift|< Distance per Encoder Revolution

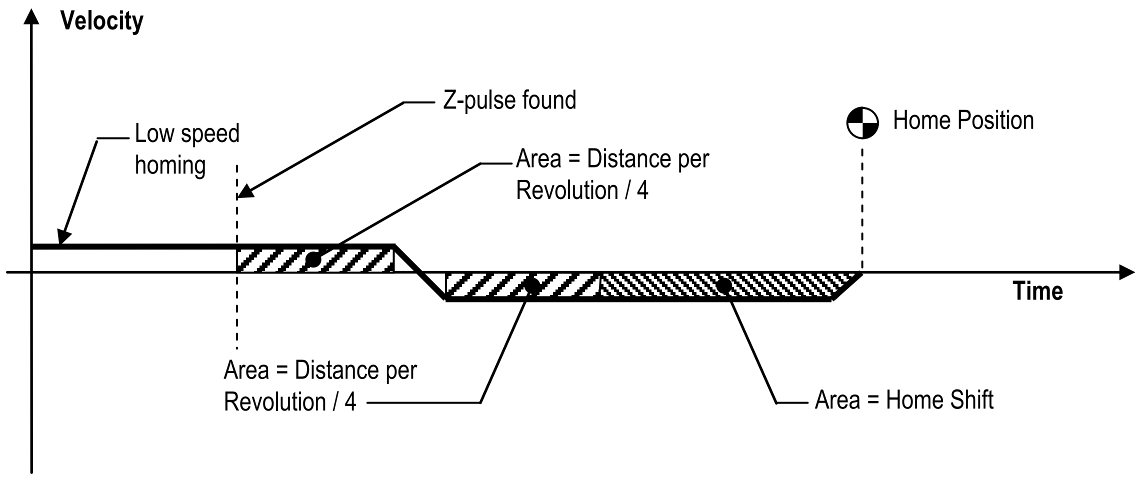


Figure 9-6: Home Operation Time Chart Example #5

<u>NOTE</u>: The purpose for moving the additional ¼ Distance per Revolution forward and backward is to avoid the friction effect when the Home Shift is in a very small amount.

9.16.6 Home Type = 0, Home Direction = 1, Home Shift < 0 and |Home Shift| > Distance per Encoder Revolution

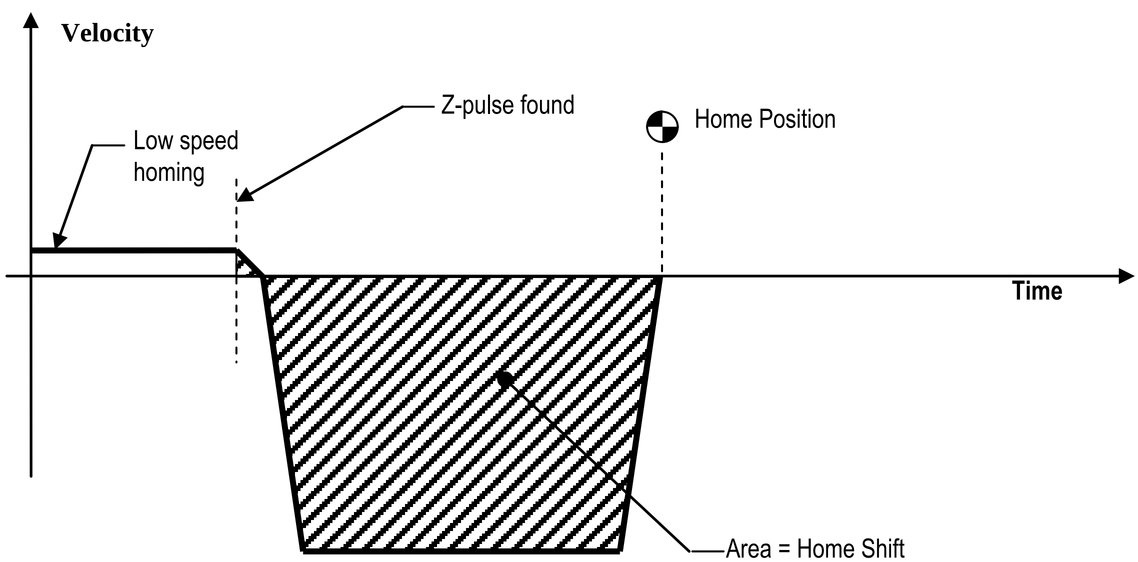


Figure 9-7: Home Operation Time Chart Example #6

9.16.7 Home Type = 1, Home Direction = 1, Home Shift = 0

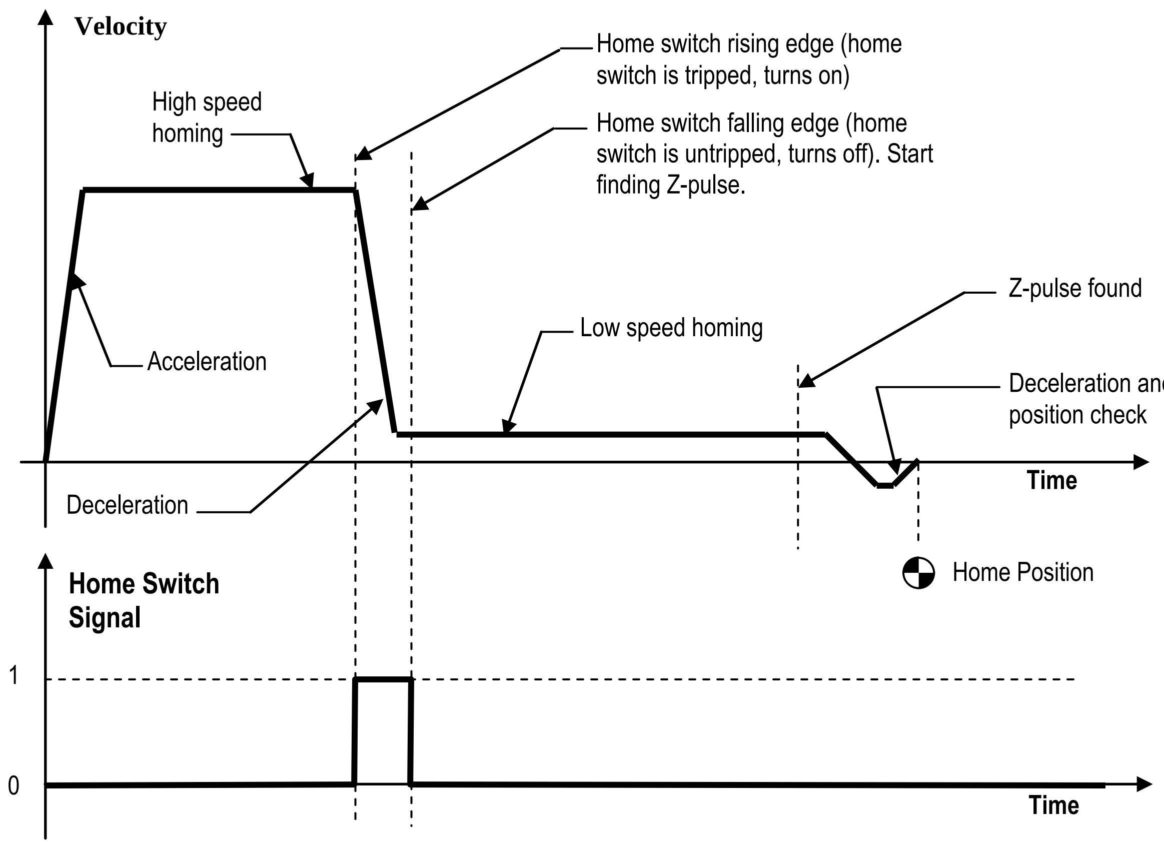


Figure 9-8: Home Operation Time Chart Example #7

NOTE: Other combinations of home direction and home shift settings for Home Type =1 are very similar to Home Type = 0, *after* high-speed homing is performed, the home switch falling edge is encountered, and deceleration to low speed homing has occurred.

For Home Type = 1, Home Direction = 1, Home Shift > 0 and Home Shift < Distance per Encoder Revolution, see Section 7.16.3, which shows what occurs after low speed homing.

Home Type = 1, Home Direction = 1, Home Shift > 0 and Home Shift > Distance per Encoder Revolution, see Section 7.16.4, which shows what occurs after low speed homing.

Home Type = 1, Home Direction = 1, Home Shift < 0 and |Home Shift|< Distance per Encoder Revolution, see Section 7.16.5, which shows what occurs after low speed homing.

Home Type = 1, Home Direction = 1, Home Shift < 0 and |Home Shift|> Distance per Encoder Revolution, see Section 7.16.6, which shows what occurs after low speed homing.

9.16.8 Home Type = 2, Home Direction = 1, Home Shift = 0

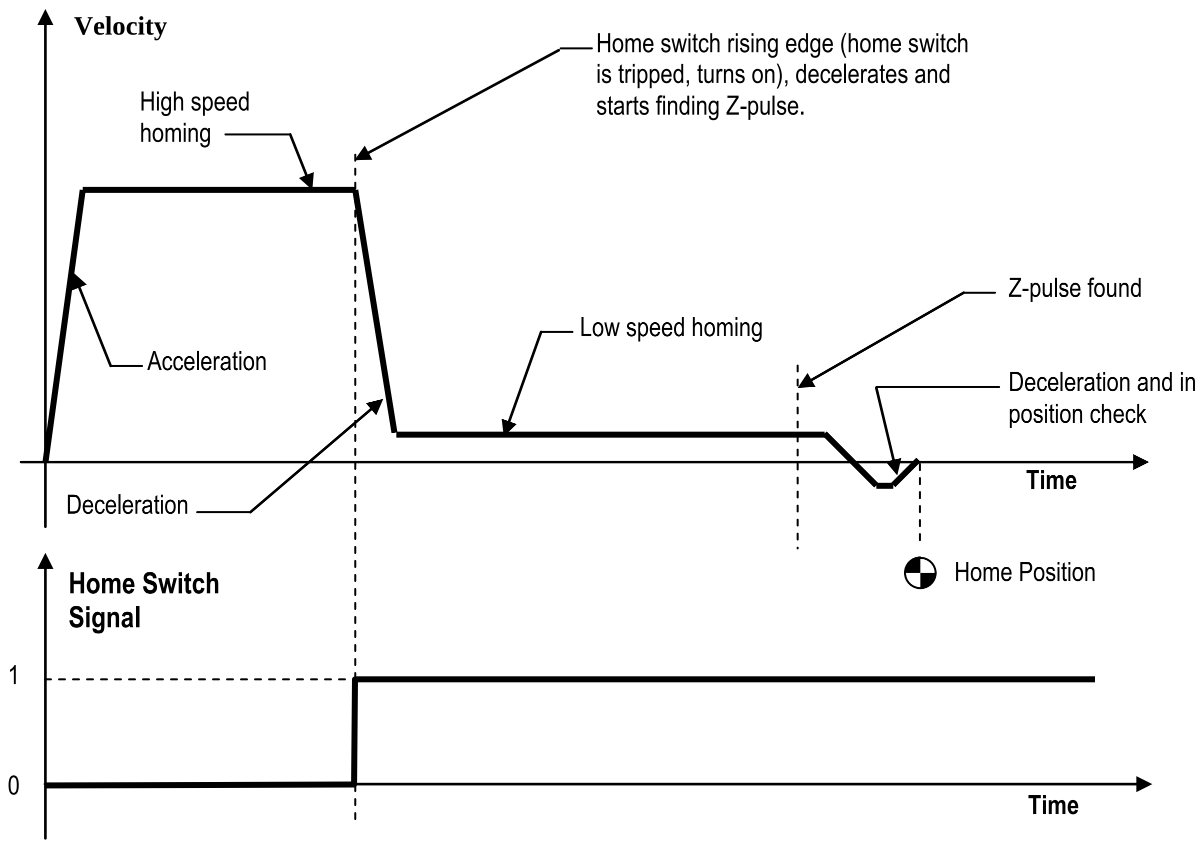


Figure 9-9: Home Operation Time Chart Example #8

9.16.9 Home Type = 3, Home Direction = 1, Home Shift < 0 and Home Shift < Distance per Encoder Revolution

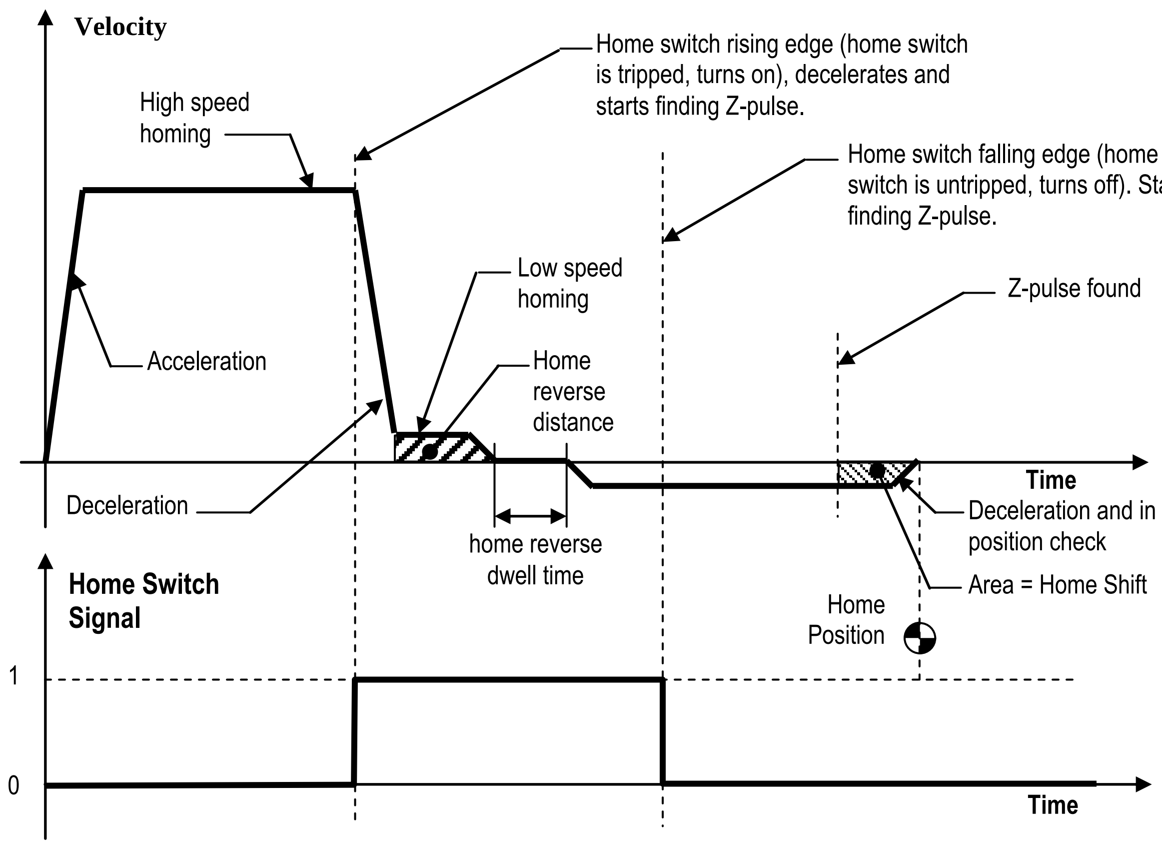


Figure 9-10: Home Operation Time Chart Example #9

9.16.10 Home Type = 4, Home Direction = 1, Home Shift > 0 and Home Shift < Distance per Encoder Revolution

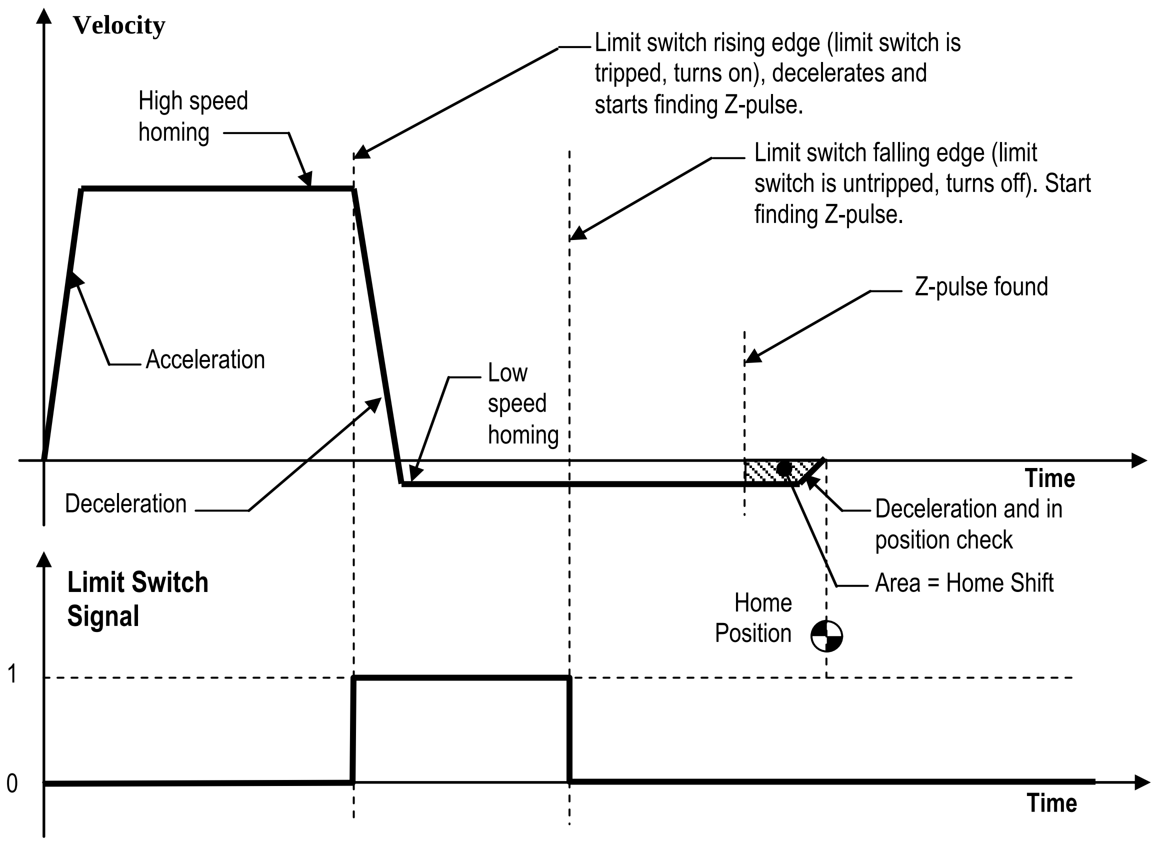


Figure 9-11: Home Operation Time Chart Example #10

9.16.11 Home Type = 5, Home Direction = 1, Home Shift > 0 and Home Shift < Distance per Encoder Revolution

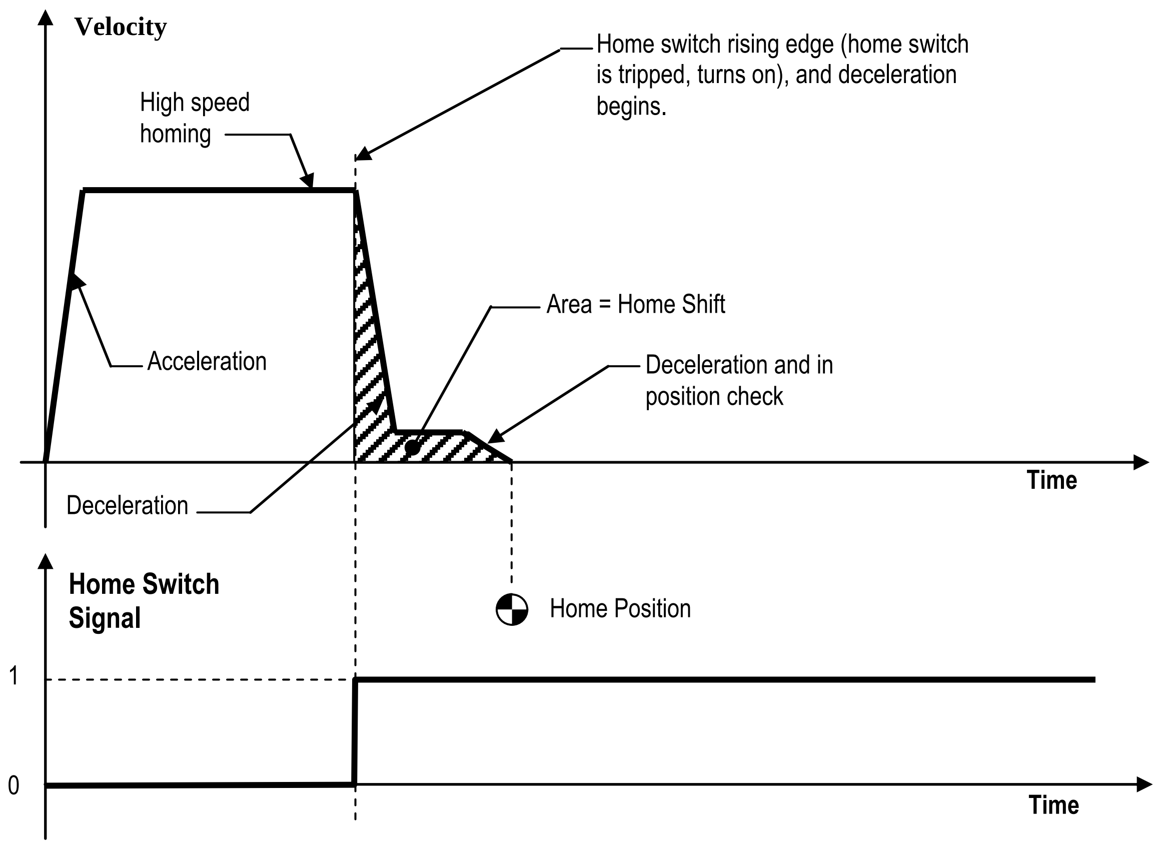


Figure 9-12: Home Operation Time Chart Example #11

Chapter 10: Synchronization Control Parameters and Usage

10.1 Overview

This chapter explains how to set up your ServoWorks CNC system for synchronization control of multiple axes, including setting the parameters related to synchronization control.

To drive a large gantry machine, a command for one axis can drive two (or more) motors synchronously. Moreover, for synchronization error compensation, feedback information from each motor allows a positional difference (synchronization error) between the two motors to be detected. If a synchronization error beyond a set value occurs, an alarm is issued to stop movement along the axes.

The reference axis for synchronization error compensation is referred to as a master axis (M axis), while the axis to which compensation is applied is referred to as a slave axis (S axis).

ServoWorks CNC synchronization control supports up to 3 pairs of Master/Slave axes or 1 master axis followed by up to 3 slave axes, depending upon the application configuration.

If the agreement between the positions of the master and slave axes is lost when the system power is turned off, the ServoWorks controller matches up the difference between them. After performing a follow-up at power on, the controller sends the difference to the slave axis to adjust its initial position feedback such that it agrees with that of the master axis. (No real operation occurs at this point.)

10.2 Sync Master Axis

Description

The master axis number with which the slave axis is associated.

Valid Values: 0 –16

Default Value: 0 (None)

Meaning of Values

0 – None, not a sync axis
1 ~ 16 – Master Axis Number

Note

This parameter may also be referred to as “sync slave axis associated to axis.”

10.3 Sync Slave Axis Compensation Gain

Description

The compensation gain of the slave axis, when there is sync error between the master axis position and the slave axis position (actual).

Valid Values: 0 – 500%

Default Value: 50%

10.4 Over Position Error Sync Limit – Moving

Description

The position error limit between the master axis and the slave axis at which the Emergency Stop will be triggered while these axes are moving. (When the NC is in moving status, the Emergency Stop will be triggered if the sync position error exceeds this value.)

[NOTE: Position error is the difference between the actual position of the master axis and the actual position of the slave axis.]

Measured in Units of: mm or inches

Range of Valid Values: 0 – 999,999.9 mm or 0 – 999,999.9 inches

Default Value: 6.0 mm or 0.236220 inches

10.5 Over Position Error Sync Limit – Stopped

Description

The position error limit between the master axis and the slave axis at which the Emergency Stop will be triggered while these axes are stopped. (When the NC status is “Stopped,” the Emergency Stop will be triggered if the sync position error exceeds this value.)

Measured in Units of: mm or inches

Range of Valid Values: 0 – 999,999.9 mm or 0 – 999,999.9 inches

Default Value: 1.0 mm or 0.039370 inches

10.6 Sync Control on Startup

Description

Enabling or disabling of synchronization control between master and slave axes on the startup of a ServoWorks CNC application.

Valid Values: 0, 1

Meaning of Values

0 – Synchronization control disabled on application startup
1 – Synchronization control enabled on application startup

Default Value: 0

Warning

If neither “sync control on startup” nor “sync control on reset” is enabled, synchronization control will not be engaged and slave axes will not move.

10.7 Sync Control on Reset

Description

Enabling or disabling of synchronization control between master and slave axes on the press of the “Reset” button in a ServoWorks CNC application.

Valid Values: 0, 1

Meaning of Values

0 – Synchronization control disabled on the press of the “Reset” button
1 – Synchronization control enabled on the press of the “Reset” button

Default Value: 0

Warning

If neither “sync control on startup” nor “sync control on reset” is enabled, synchronization control will not be engaged and slave axes will not move.

10.8 Synchronization Functions

10.8.1 Overview

A movement along an axis can be executed simply by executing a move command specified for that axis or by synchronizing the movement with another axis. Either of these two types can be selected by means of parameter configuration.

In synchronous operation, commands can be specified for a master axis, and then the slave axis moves along the tool with the master axis accordingly. The function can synchronize both automatic and manual operations. Also, the following functions are provided.

10.8.2 Synchronization Error Check Function

The synchronization error check function is based on machine coordination. The difference between the servo position of the master axis and that of the slave axis is monitored constantly. If the function detects a difference greater than or equal to a preset value (calculated in the following figure), a software E-Stop stops the machine.

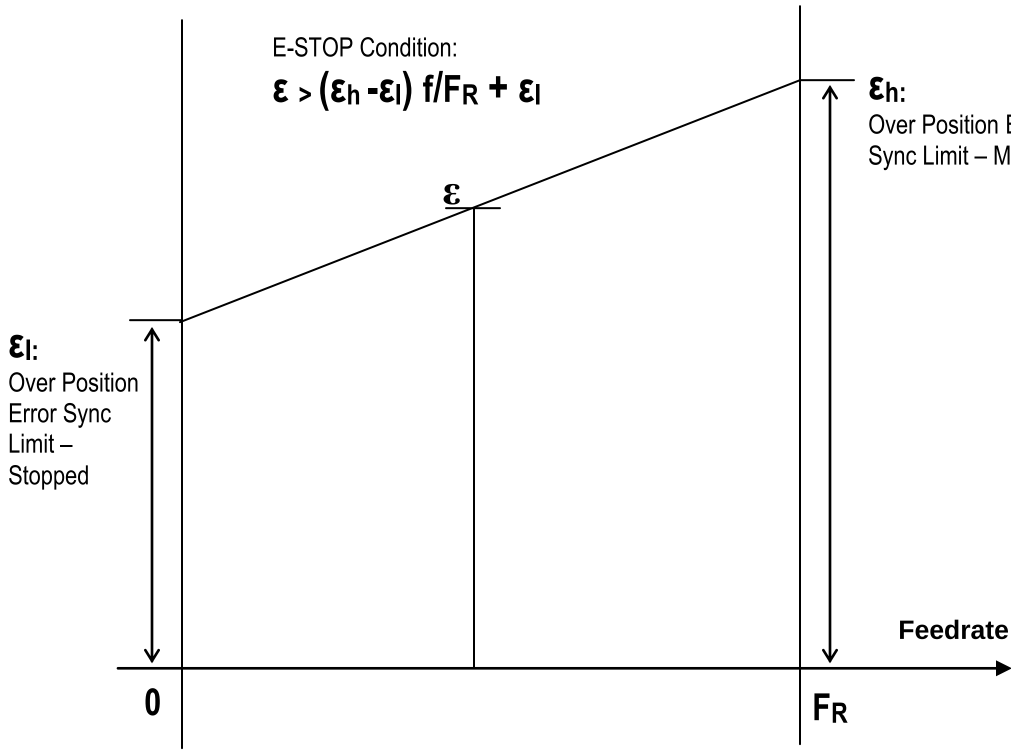


Figure 10-1: Sync Error Limit Formula

10.8.3 Synchronization Compensation Function

The synchronization compensation function is enabled as soon as the ServoWorks CNC Engine is powered on and there is synchronization deviation (see the figure on the following page). If the error accumulates (if the synchronization compensation function can't help the slave axis catch up with the master axis) and exceeds a certain set value (calculated above), a software E-STOP is triggered. Each "Reset" call will allow the slave axis position to be adjusted automatically (with a 1 second try) by the synchronization compensation function.

Synchronization Compensation Calculation (Proportional Control):

$$SynComp = SyncErr * K_{syn}$$

SynComp: Synchronization Compensation, the input to slave position command (calculated)

SyncErr: Synchronization Error, the difference between the position feedback from master and slave axes

K_{syn}: Sync Slave Axis Compensation Gain (%) – see *Section 10.3: Sync Slave Axis Compensation Gain*

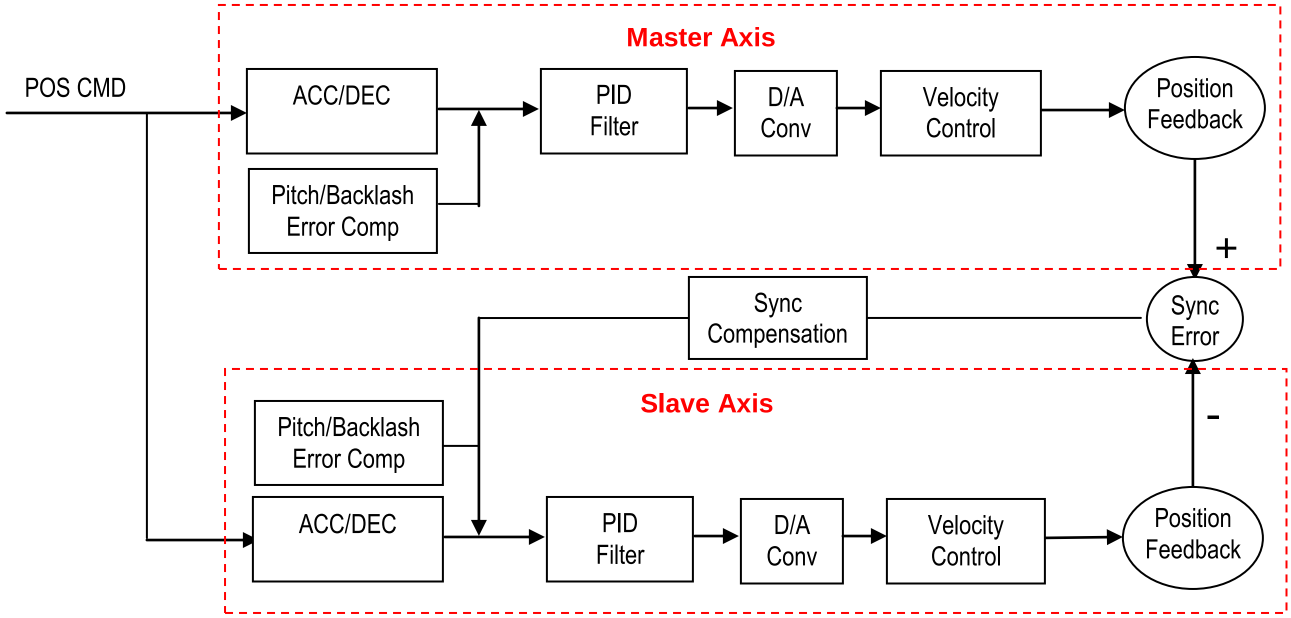


Figure 10-2: Synchronization Compensation

10.8.4 Homing Operation

In the homing operation with synchronized axes, the master axis only searches the home position and the slave axis simply follows the movement of the master axis. Homing master axis will move slave axis as much as master axis is moved. It does not mean the slave axis moves its reference point, thus, to move the slave axis to its reference point, you will need to do homing independently.

You are able to choose whether the master and the slave axes are synchronized when the GUI start, or reset. Not synchronize the master and the slave axes will allow you to execute homing independently. However, you should check that the machine is enough flexible to search each of reference point.

10.9 ServoWorks CNC Parameters Configuration for Synchronous Control

10.9.1 Overview

In addition to the four parameters that are dedicated to synchronous control (described in Sections 4.2 – 4.5), there are other considerations for parameters in a ServoWorks CNC system that includes synchronous control.

All Motor/Servo Drive and Servo Control parameters can be different for the master axis and all slave axes except for Smoothing Time. However, due to the master/slave relationship, some NC/Machine parameters are required to be preset as listed in the following sections.

NOTE: Machine error compensation, including backlash compensation and pitch error compensation, should be tuned before pairing synchronous axes.

10.9.2 Calculating the "Home Shift" Parameter for All Synchronous Axes

The "Home Shift" in Master/Slave will be used to align the master/slave pair. The limits for the value of "Home Shift" for all synchronous axes should be calculated with the following formula:

$$DPR + SMT * SPD_{gs} * 2 \geq HS_{m/s} \geq SMT * SPD_{gs} * 2$$

DPR: Distance Per Revolution (motors)

SMT: Smoothing Time

SPD_{gs}: Grid Search Speed

$HS_{m/s}$: Home Shift (master or slave axis)

10.9.3 Parameters with Dedicated Settings for Synchronization Control

- Axis Type: the master axis type must be "Normal" or "Rotary"; the slave axis type must be "Slave." [NOTE: In Configuration Mode, the ServoWorks CNC Engine must be off when the "Axis Type" is switched to "Slave."]

- Sync Master Axis: the master axis value must be "0"; the slave axis value should be the number of the master axis to which it is synchronized

- Sync Slave Axis Compensation Gain: this value applies to slave axes only (0%–500%)

- Over Position Error Sync Limit – Moving: this value applies to slave axes only

- Over Position Error Sync Limit – Stop: this value applies to slave axes only

10.9.4 Parameters Requiring Identical Settings for Master Axis & Slave Axis

For the following parameters, the master axis and all slave axes must have the exact same value:

- Smoothing Time
- Rotary Position Display Range
- Plus Stroke
- Minus Stroke
- Jog Feedrate
- Rapid Feedrate
- Home Type
- Home Direction
- Home Switch Type
- Home Position
- Home Reverse Dwell Time
- Home Reverse Distance
- Grid Search Speed
- Home Switch Search Speed
- Reference Position #2
- Reference Position #3
- Reference Position #4
- Always Search for Home
- Encoder Type

10.9.5 Parameters That Can Have Independent Settings for Master Axis & Slave Axis

The following parameters can have different values for the master axis and all slave axes:

- Backlash value
- Pitch Origin
- Pitch Interval
- Home Shift

10.10 Other Function Implementations for Synchronous Control

10.10.1 Machine Lock/InterLock

The machine lock or interlocks command for the master axis will also activate its slave axis, while all commands to the slave axes will be ignored.

10.10.2 Work Coordinates Selection

The work coordinates preset will be ignored for all slave axes. Only master sets will be called.

Chapter 11: Machine Compensation Parameters

11.1 Overview

11.1.1 What is Machine Compensation?

Machine compensation is a function to correct leadscrew backlash. Backlash is lost motion after reversing direction, due to the axial free motion between the ball nut and the ball screw (the gap between the threads of the leadscrew and the gear).

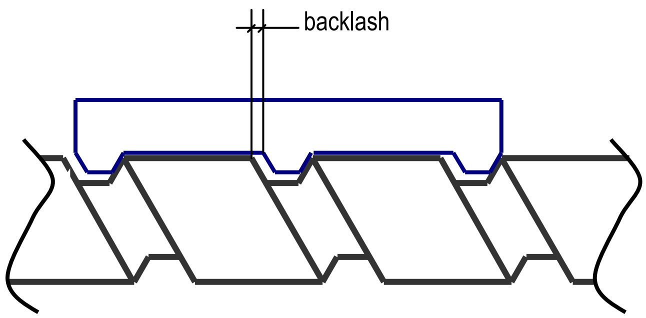


Figure 11-1: Backlash

11.1.2 Machine Compensation Types

There are three different types of machine compensation: One Time, Time, and Distance:

- One Time: Upon reversing direction, this mode adds an offset value, which is set in the parameter "Backlash Value." The advantage of this mode is its simplicity and effectiveness.
- Time: Upon reversing direction, this mode adds an offset value that linearly increases from 0 to "Backlash Value" in the time set in the parameter "Backlash Compensation Distribution." Since the machine compensates for backlash gradually, the effects of overshooting are not as sharply pronounced. A disadvantage is that the cut path will be a function of the feedrate.
- Distance: Upon reversing direction, this mode sets the backlash offset in three steps. In the first step, this mode adds the offset value set in the parameter "Backlash Value (Low)" until the traveled distance reaches the parameter "Backlash Distance (Low)." Then, the offset value linearly increases from "Backlash Value (Low)" to "Backlash Value" during the distance that the axis reaches a total traveled distance equal to the parameter "Backlash Distance." Finally, after the total traveled distance reaches the "Backlash Distance," the offset value is held at the "Backlash Value." See Figure 11-2 for a graphical representation of this mode. This mode reduces the effects of overshooting as with time mode, but has the advantage of the cut path not being as strong a function of the feedrate. The disadvantage of this mode is its relative complexity.

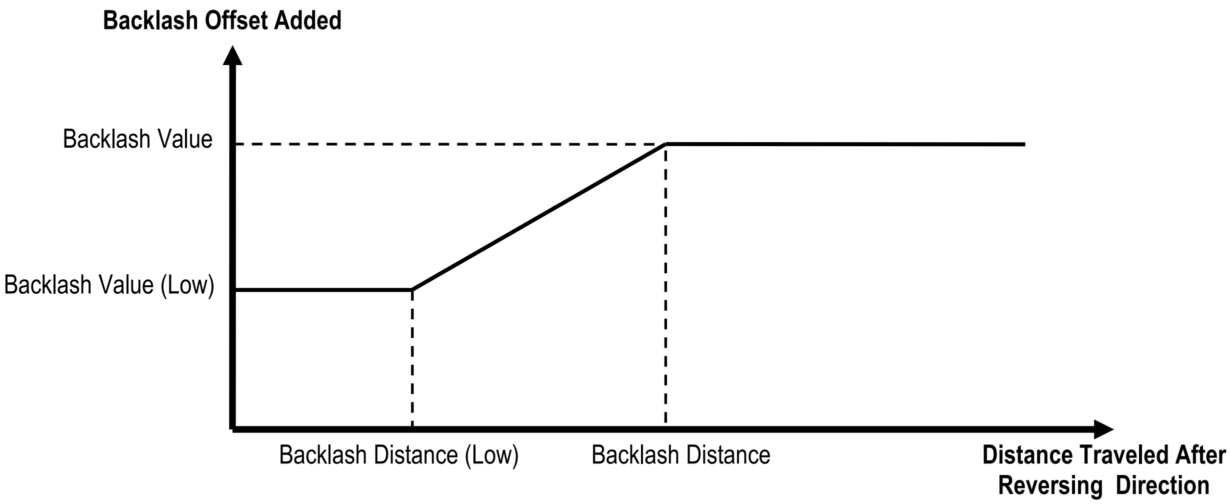


Figure 11-2: Machine Compensation in Distance Mode

11.2.1 Backlash Compensation Type

Description

This value sets the type of machine compensation.

Valid Values: One Time, Time, Distance

Meaning of Values

One Time – Machine Compensation works in One Time mode.
Time – Machine Compensation works in Time mode.
Distance – Machine Compensation works in Distance mode.

Default Value: Time

NOTE:
Machine compensation is effective only after a homing operation is complete.

11.2.2 Backlash Value

Description

The value is set to compensate for backlash. This axis' backlash value should be the lost motion value which is provided by the leadscrew manufacturer.

Measured in Units of: mm, inches or degrees

Range of Valid Values: 0 – 999,999.9 mm, 0 – 999,999.9 inches or 0 – 999,999.9 degrees

Default Value: 0.000 mm, 0.000 inches or 0.000 degrees (disables backlash compensation)

NOTE:
Machine compensation is effective only after a homing operation is complete.

NOTE:
This parameter is also known as "reverse error compensation."

11.2.3 Backlash Comp. Distribution

Description

In Time mode, this value determines the period of time over which the backlash offset, upon reversing direction, linearly increases from 0 to the "Backlash Value" parameter.

Measured in Units of: ms

Range of Valid Values: 0 – 10,000 ms

Default Value: 0.000 ms

11.2.4 Backlash Value (Low)

Description

In Distance mode, this value determines the backlash offset added upon reversing direction. This backlash offset is applied until the traveled distance reaches "Backlash Distance (Low)."

Measured in Units of: mm, inches or degrees

Range of Valid Values: 0 – Backlash Value

Default Value: 0.000 mm, 0.000inches or 0.000degrees

11.2.5 Backlash Distance

Description

In Distance mode, this value determines the total distance traveled, after reversing direction, at which the full "Backlash Value" is added.

Measured in Units of: mm, inches or degrees

Range of Valid Values: 0 – 999,999.9 mm, 0 – 999,999.9 inches or 0 – 999,999.9 degrees

Default Value: 0.000 mm, 0.000inches or 0.000degrees

11.2.6 Backlash Distance (Low)

Description

In Distance mode, this value determines the distance traveled, after reversing direction, at which the added backlash offset starts increasing linearly from "Backlash Value (Low)" to "Backlash Value."

Measured in Units of: mm, inches or degrees

Range of Valid Values: 0 – Backlash Distance

Default Value: 0.000 mm, 0.000inches or 0.000degrees

Chapter 12: Pitch Error Compensation Parameters and Usage

12.1 Overview

12.1.1 What is Pitch Error Compensation?

Pitch error compensation is calibration compensation using pitch intervals to correct for imperfections in the ballscrew of the feed motor.

Pitch error compensation occurs at any position on the axis, by interpolating between the pitch errors at the two nearest pitch error measurement points.

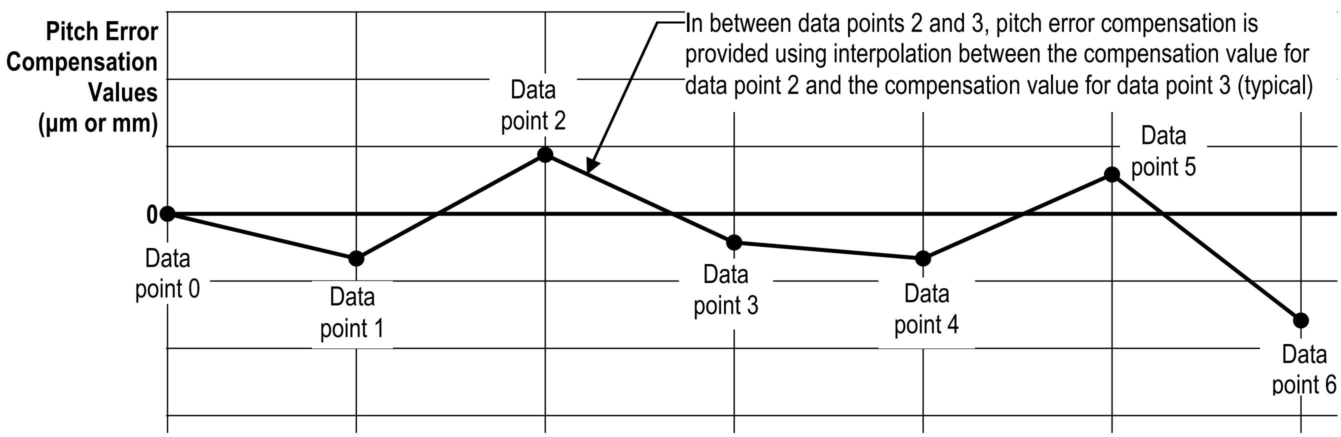


Figure 12-1: Linear Interpolation for Pitch Error Compensation Between Pitch Error Compensation Data Points

12.1.2 Setting Up Pitch Error Compensation

In order to set up your machine to perform pitch error compensation correctly, you must understand the two NC/machine parameters related to pitch error compensation, and you must enter pitch error compensation data in your ServoWorks CNC application or set up your stored pitch error compensation in a certain file and format.

The starting point for pitch error compensation (data point 0) is the left most point, and must always have a value of zero. This starting point must always be less than the machine origin, but cannot be less than the minus stroke. The starting point is defined as the zero calibration point. [Refer to the examples later in this chapter.]

There are a maximum of 1024 points of stored pitch error values for each axis, so set your pitch interval accordingly.

12.1.3 Limitations

Pitch error compensation is effective only after a homing operation is complete.

12.2 Pitch Error Compensation Parameters

12.2.1 Pitch Origin

Description

Index value of the machine origin (relative to the pitch error measurement starting position) among the pitch error measurement points.

$$\text{Pitch Origin} = \frac{\text{Relative Distance from Machine Origin to the Pitch Error Measurement Starting Position}}{\text{Pitch Interval}}$$

Range of Valid Values: 0 – 1,023

Default Value: 0.0

Note

This parameter may also be referred to as "Pitch Origin Index" or "Pitch Index."

12.2.2 Pitch Interval

Description

Interval distance between the two points where the pitch error is measured.

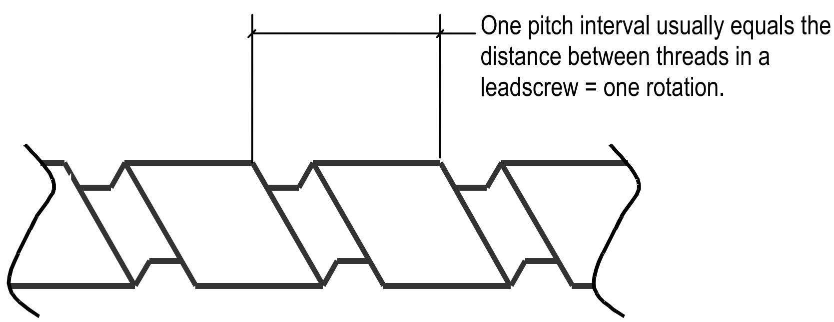


Figure 12-2: Pitch Interval

Measured in Units of: mm or inches

Range of Valid Values: 0 – 999,999.9 mm or 0 – 999,999.9 inches

Default Value: 10.000 mm or 0.393701 inches

Discussion

There are a maximum of 1024 points of stored pitch error values for each axis, so set your pitch interval accordingly.

Note

This parameter may also be referred to as "Pitch Length."

12.3 Pitch Error Compensation Values

Description

A pitch error compensation value is the difference between the expected (theoretical, specified, ideal) position and the actual (measured, real) position of a pitch interval. Both values are relative to the home position. Pitch error compensation values are machine set-up data, for an individual machine. Entering this data is equivalent to calibrating an individual machine.

Measured in Units of: µm (ServoWorks MC-Quad and customized ServoWorks CNC applications)
mm (ServoWorks S-100M, S-120M and S-140M)

Range of Valid Values: 0 – 99,999.999 µm (ServoWorks MC-Quad and customized ServoWorks CNC applications)
0 – 999,999.999 mm (ServoWorks S-100M, S-120M and S-140M)

Default Value: 0 µm (ServoWorks MC-Quad and customized ServoWorks CNC applications)
0 mm (ServoWorks S-100M, S-120M and S-140M)

Note

Pitch error compensation is measured from the home position.

Example

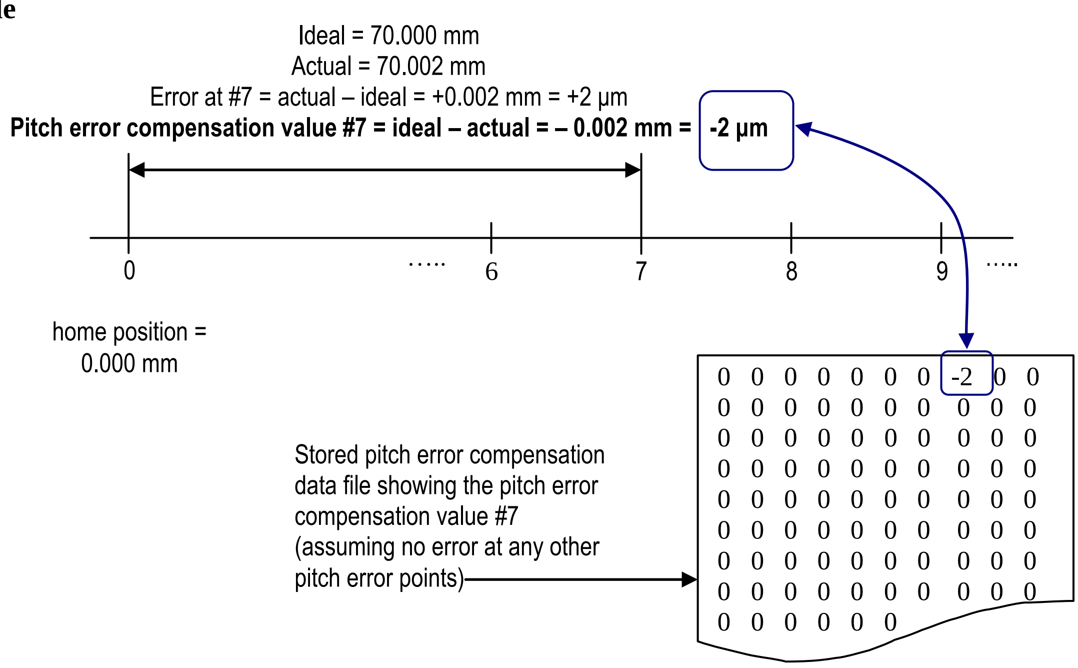


Figure 12-3: Pitch Error Compensation Example Calculation

12.4 Stored Pitch Error Compensation Data

Pitch error compensation data is stored differently depending upon which application you are using, as explained in the following two sections.

12.4.1 Stored Pitch Error Compensation Data for ServoWorks MC-Quad and Customized ServoWorks CNC Applications

Location and File Name for Stored Pitch Error Compensation Data

Currently, ServoWorks MC-Quad does not provide an interface for setting pitch error compensation data, so the pitch error compensation data must be saved in a text file with specific name, and located at specific location:

- Location: Under the ServoWorks application folder (e.g. C:\Program Files\SoftServo\MC-Quad), there is an "ini" folder. The stored pitch error compensation data file must be located in this "ini" folder.
- File name: The store pitch error compensation data file must be named: StoredPitchErr.dat
- Example of complete file address: "C:\Program Files\SoftServo\MC-Quad\ini\StoredPitchErr.dat"

NOTES:

1) For ServoWorks Development Kit (SDK) users, the pitch error compensation data file name and location is specified by the "sssSetPitch" API function call, and does not need to comply with the restrictions above.
2) If the pitch error compensation data file is modified while the ServoWorks CNC application is running, the change will not be effective until the next time the ServoWorks CNC application starts.

Required Format for Stored Pitch Error Compensation Data

Inside the stored pitch error compensation data file for ServoWorks MC-Quad or customized ServoWorks CNC applications, the data must have the following format:

Measurement value for the starting point for Axis 1 (must have the value of zero)

```
Axis_1

0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0

Axis_2

0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
```

Measurement value for the starting point for Axis 2 (must have the value of zero)

Figure 12-4: Required Format for Stored Pitch Error Compensation Data

As shown in the previous figure, the data file is divided into sections by the “Axis_Number” symbol. Each section contains the pitch error compensation measurement data for each corresponding axis, specified by the axis number. The data area contains the measured pitch error value at each pitch error measurement point, and the data must be arranged in a sequence from left to right, top to bottom.

Pitch error values for ServoWorks MC-Quad and for customized ServoWorks CNC applications are measured in units of micrometers. For each axis, the maximum number of measured pitch error data points is 1024 (0 through 1023).

12.4.2 Stored Pitch Error Compensation Data for ServoWorks S-100M, S-120M and S-140M

ServoWorks S-100M, S-120M and S-140M each provide an interface for setting pitch error compensation data right in the application, and this data is saved in the Windows registry. Therefore, no special pitch error compensation data file is required. In fact, if any StoredPitchErr.dat exists, it is ignored by ServoWorks S-100M, S-120M and S-140M.

The pitch error compensation data interface appears as follows:

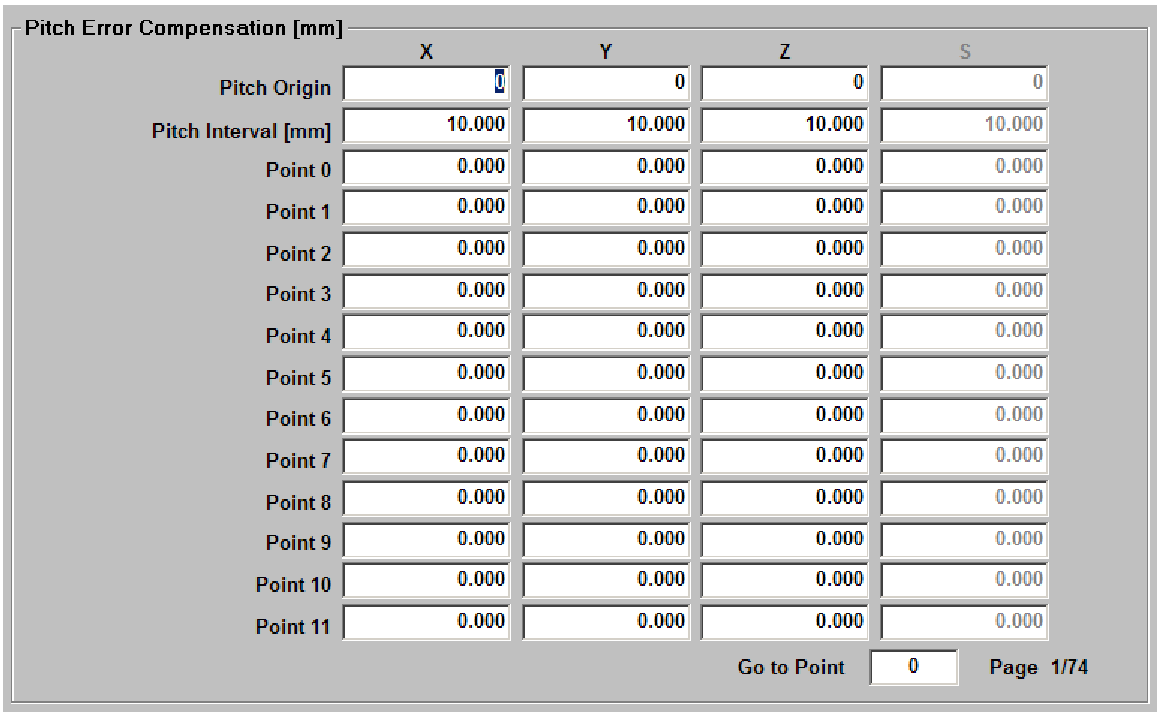


Figure 12-5: Pitch Error Compensation Data Screen in Configuration Mode of the ServoWorks S-140M Series

Pitch error values for ServoWorks S-100M, S-120M and S-140M are measured in units of millimeters. For each axis, the maximum number of measured pitch error data points is 1024 (0 through 1023).

12.5 Examples of Pitch Error Compensation Setup

12.5.1 Example #1: Simple Example

Let's assume a machine has the following parameters for the X axis:

- Soft limits: minus stroke = –100.0 mm, and plus stroke = 100.0 mm.
- The pitch error compensation measurement starting position is at –100.0 mm (the minus stroke).
- The pitch interval is 50 mm.

Then, the pitch origin can be calculated as:

$$\text{Pitch Origin} = \frac{\text{Relative Distance from Machine Origin to the Pitch Error Measurement Starting Position}}{\text{Pitch Interval}} = \frac{0.0 - (-100.0\text{ mm})}{50.0\text{ mm}} = \frac{100.0\text{ mm}}{50.0\text{ mm}} = 2.0$$

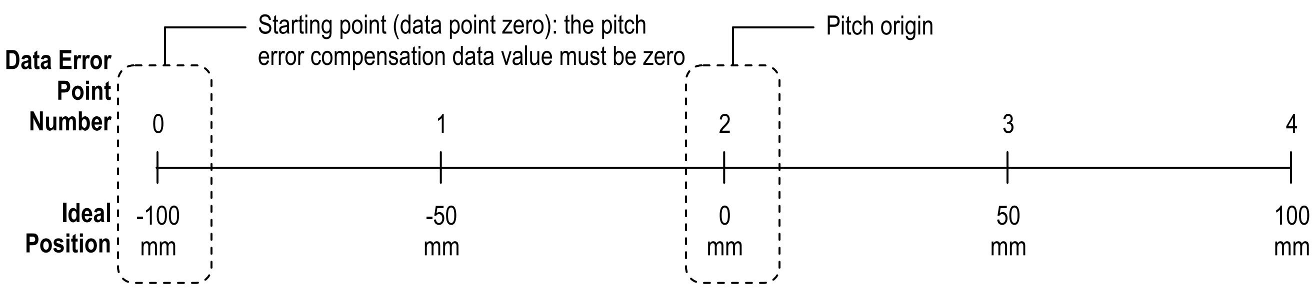


Figure 12-6: Pitch Error Compensation Setup Example #1: Schematic

In the stored pitch error compensation data file, in the "Axis_1" section, the first data corresponds to pitch error measurement position at –100.0 mm (the starting point), and the data value must be 0. Then the next data value is the measured pitch error at position –50.0 mm, and so forth. An example data file follows:

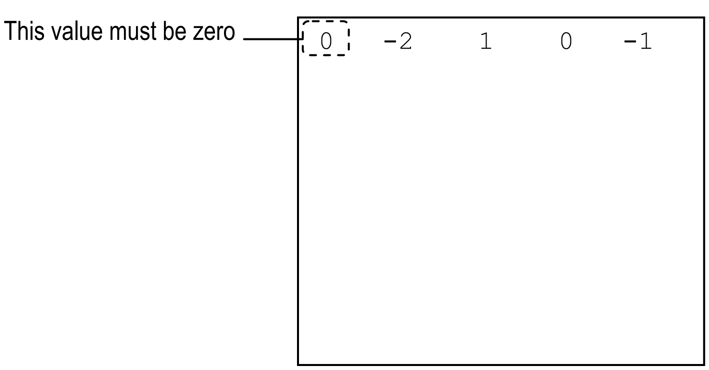


Figure 12-7: Pitch Error Compensation Setup Example #1: Data File

12.5.2 Example #2: Example with a Pitch Origin In Between Two Data Points

Let's assume a machine has the following parameters for the X axis:

- Soft limits: minus stroke = –100.0 mm, and plus stroke = 100.0 mm.
- The pitch error compensation measurement starting position is at –100.0 mm (the minus stroke).
- The pitch interval is 40 mm.

Then, the pitch origin can be calculated as:

$$\text{Pitch Origin} = \frac{\text{Relative Distance from Machine Origin to the Pitch Error Measurement Starting Position}}{\text{Pitch Interval}} = \frac{0.0 - (-100.0\text{ mm})}{40.0\text{ mm}} = \frac{100.0\text{ mm}}{40.0\text{ mm}} = 2.5$$

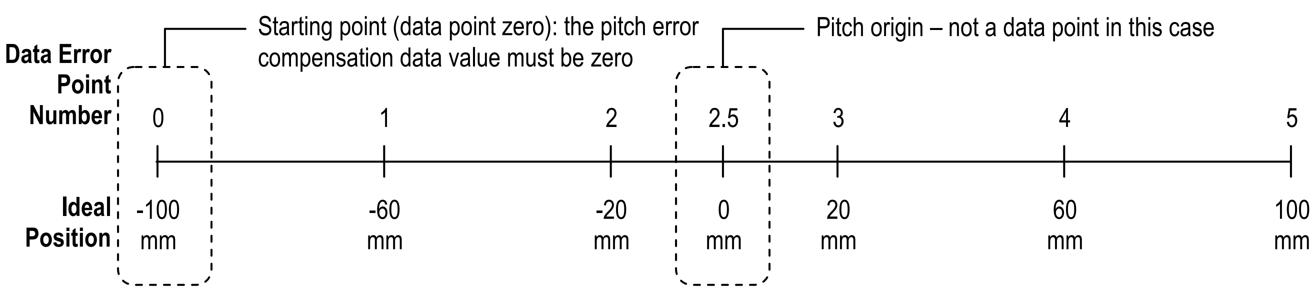


Figure 12-8: Pitch Error Compensation Setup Example #2: Schematic

In the stored pitch error compensation data file, in the "Axis_1" section, the first data corresponds to pitch error measurement position at –1000.0 mm (the starting point), and the data value must be 0. Then the next data value is the measured pitch error at position –60.0 mm, and so forth. An example data file follows:

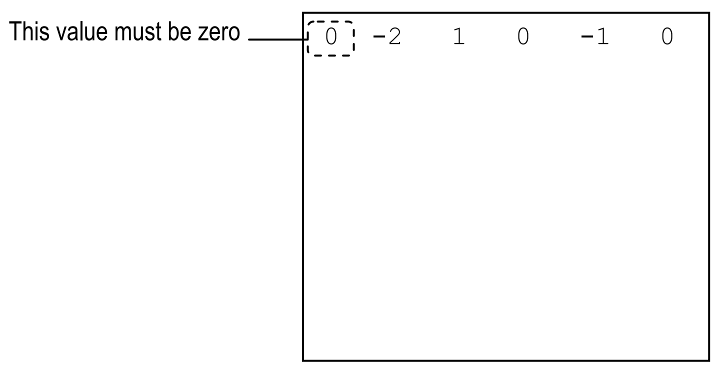


Figure 12-9: Pitch Error Compensation Setup Example #2: Data File

12.5.3 Example #3

Let's assume a machine has the following parameters for the X axis:

- Soft limits: minus stroke = –1000.0 mm, and plus stroke = 1000.0 mm.
- The pitch error compensation measurement starting position is at –1000.0 mm (the minus stroke).
- The pitch interval is 10 mm.

Then, the pitch origin can be calculated as:

$$\text{Pitch Origin} = \frac{\text{Relative Distance from Machine Origin to the Pitch Error Measurement Starting Position}}{\text{Pitch Interval}} = \frac{0.0 - (-1000.0\text{ mm})}{10.0\text{ mm}} = \frac{1000.0\text{ mm}}{10.0\text{ mm}} = 100.0$$

In the stored pitch error compensation data file, in the "Axis_1" section, the first data corresponds to pitch error measurement position at –1000.0 mm (the starting point), and the data value must be 0. Then the next data value is the measured pitch error at position –990.0 mm, and the next one is measured pitch error at position –980.0 mm, and so on and so forth.

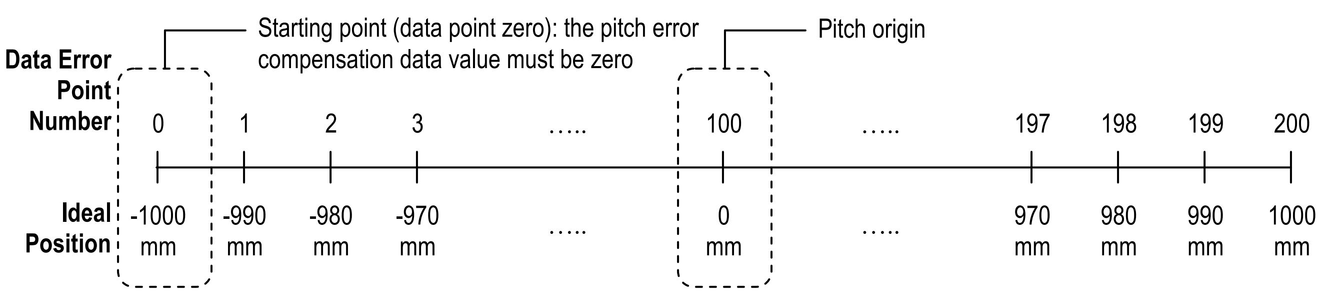


Figure 12-10: Pitch Error Compensation Setup Example #3: Schematic

Chapter 13: Tool Compensation Parameters

13.1 Overview of Tool Offsets

Tool offsets are the difference between the actual tool position and the theoretical tool position.

Each tool offset is a set of distances. When T codes are executed in a part program, the program position is actually shifted to account for tool offset (as opposed to compensating each actual movement command in the part program to account for tool offset).

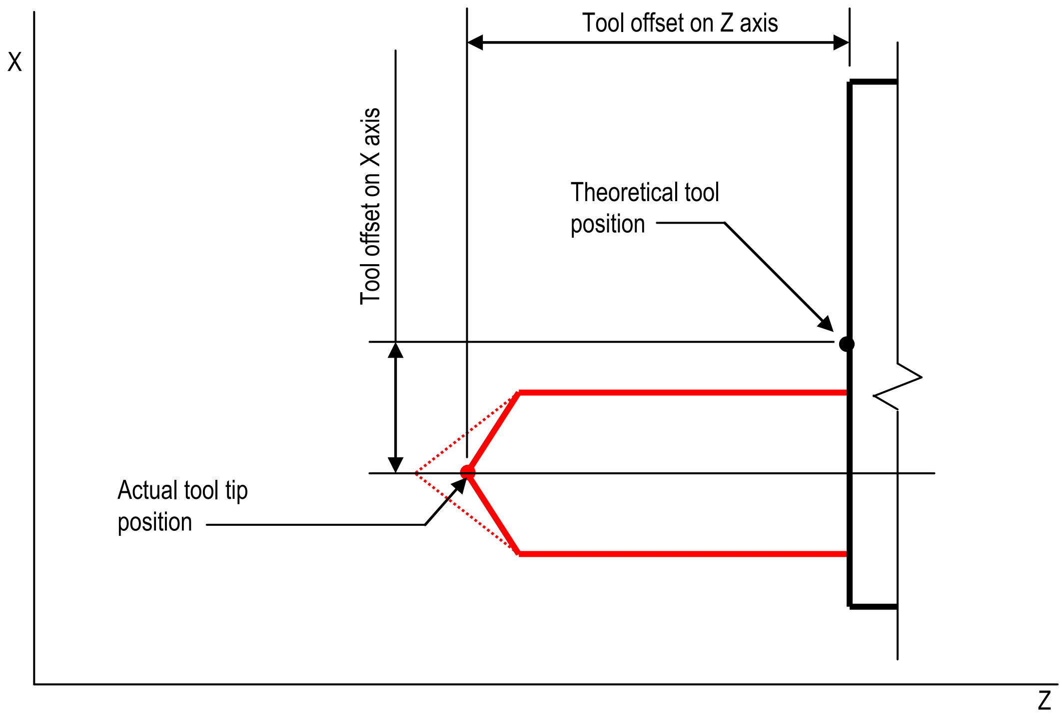


Figure 13-1: Tool Offsets

Tool offsets can be divided into two types: geometry offsets and wear offsets. The total tool offset is the sum of the geometry offset and the wear offset.

The geometry offset compensates for the tool mounting position and for the tool shape, as shown in the following figure:

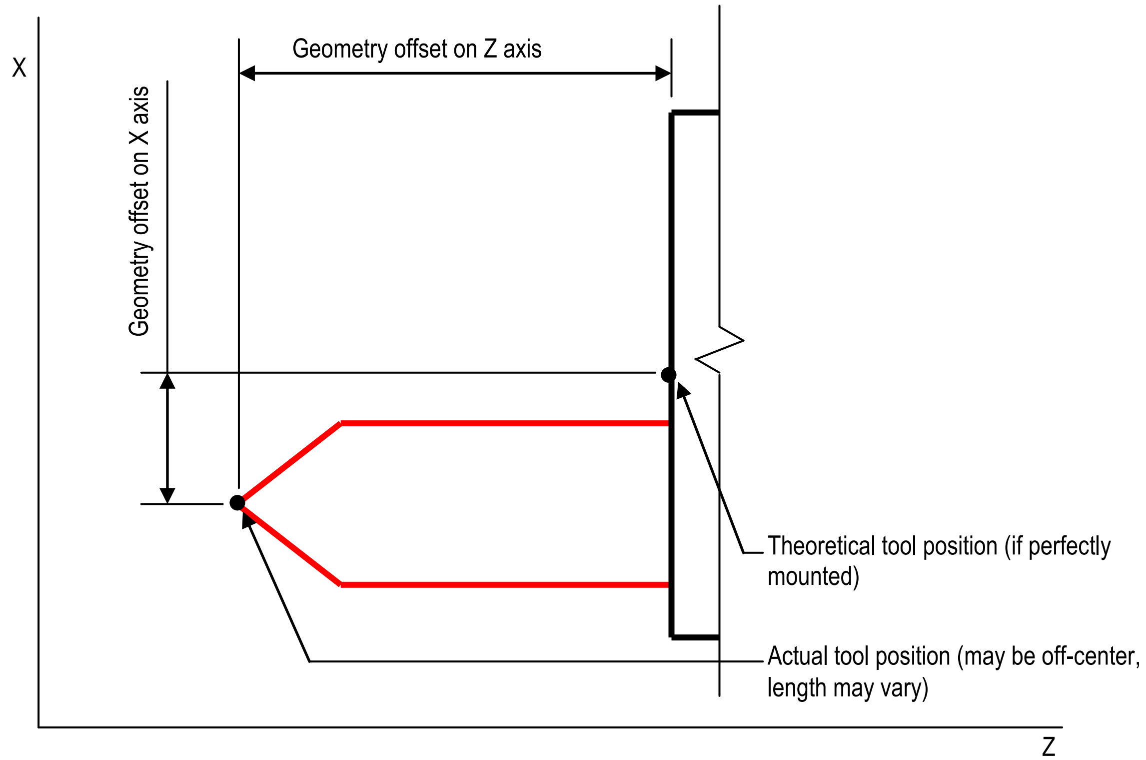


Figure 13-2: Tool Geometry Offsets

The wear offset compensates for wear to the tool tip, as shown in the following figure. Wear is continually changing as tools are used in production.

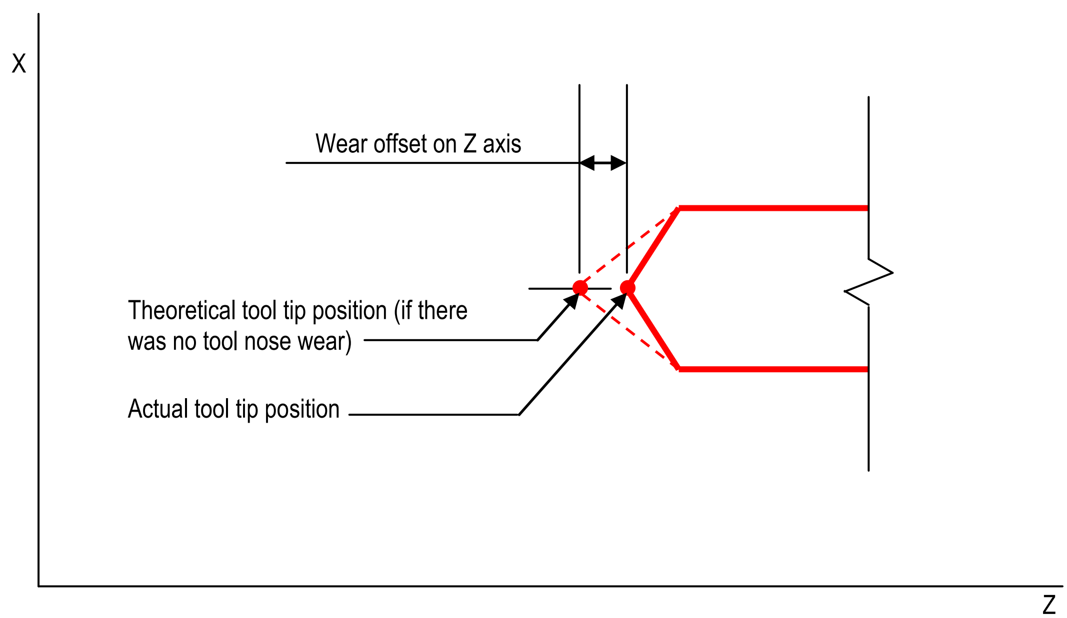


Figure 13-3: Tool Wear Offsets

13.2 Tool Radius Compensation Parameters

13.2.1 Tool Radius Compensation

Description

Using wear data only for tool radius compensation, or using both wear data and geometry data for tool radius compensation.

Valid Values: 0, 1

Meaning of Values

0 – Use wear data only
1 – Use geometry and wear data

Default Value: 0

13.2.2 Tool Radius Compensation Startup/Cancel Type

Description

This parameter determines the type of operation of tool radius compensation.

Valid Values: 0, 2

Meaning of Values

0 – Tool Length Offset Type A (when tool radius offset is applied or removed, the offset will be applied or removed during the next commanded movement)
2 – Tool Length Offset Type C (when tool radius offset is applied or removed, the offset will be applied or removed before the next commanded movement)

Default Value: 0

13.3 Tool Length Compensation Parameters

13.3.1 Tool Length Calibration Position

Description

ServoWorks S-100M, S-120M and S-140M provide a "MEASURE" function for calculating tool length geometry compensation. The tool length geometry compensation value is changed to the current Z axis program position minus the tool length calibration position when the "MEASURE" button is pressed.

Measured in Units of: mm or inches

Range of Valid Values: 0 – 999,999.9 mm or 0 – 999,999.9 inches

Default Value: 0.000 mm or 0.000 inches (the tool tip is at the workpiece)

Discussion

This is helpful when a buffer material is put in between the tool tip and the workpiece (part), to avoid damage to the workpiece. This tool length calibration position parameter accounts for the buffer material.

13.3.2 Tool Length Compensation Type

Description

This parameter determines the type of operation of tool length compensation.

Valid Values: 0, 1, 2

Meaning of Values

0 – Tool Length Offset Type A (for tool length compensations in the Z-axis direction)
1 – Tool Length Offset Type B (for tool length compensations in the X-, Y- or Z-axis directions)
2 – Tool Length Offset Type C (for tool length compensations along any specified axis)

Default Value: 0

13.3.3 G37 Limit

Description

This parameter specifies the valid range of tool length offset calibration values for automatic tool length compensation calibration (G37).

Measured in Units of: mm or inches

Range of Valid Values: 0 – 999,999.9 mm or 0 – 999,999.9 inches

Default Value: 5.000 mm or 0.19685 inches

Discussion

The valid range of tool length offset calibration values is between the positive G37 Limit and the negative G37 Limit, as measured from the theoretical surface position. For example, if the G37 Limit is 2.0 mm, then the tool length offset calibration value must be less than 2.0 mm and greater than -2.0 mm. In ServoWorks S-100M/S-120M/S-140M, an error message will be displayed if the surface detection signal is triggered before the axis reaches the surface position minus the G37 Limit, or if the axis reaches the surface position plus G37 Limit without the surface detection signal having been triggered.

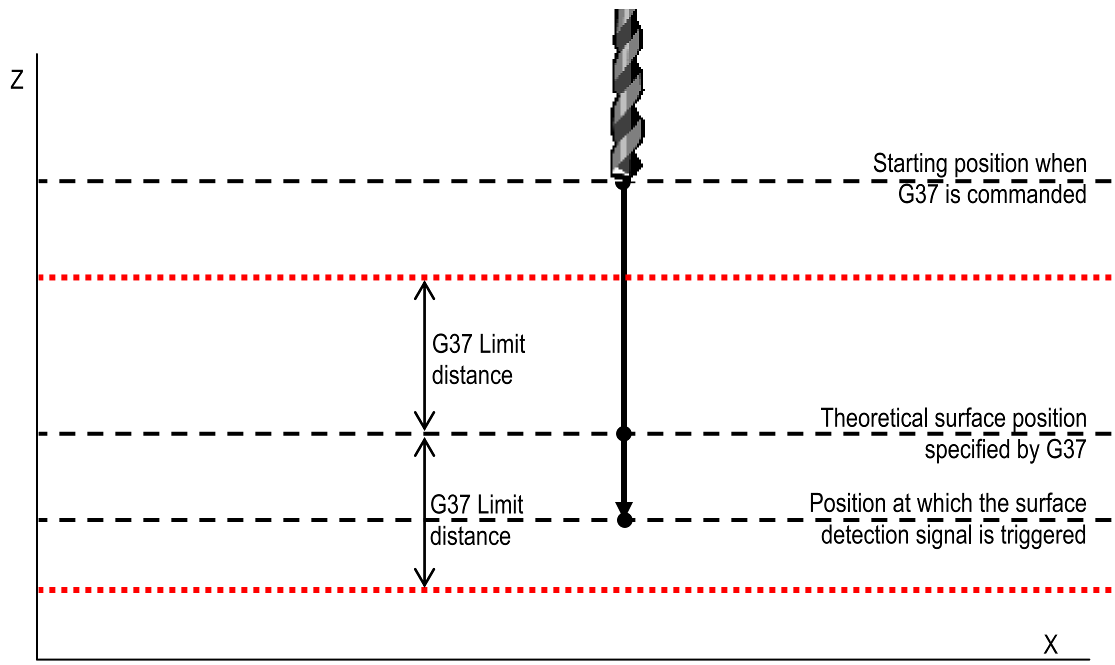


Figure 13-4: G37 Limit

13.3.4 G37 Ratio

Description

This parameter specifies the fraction of the movement toward the theoretical surface position that is executed at the rapid traverse rate during automatic tool length compensation calibration (G37).

Range of Valid Values: 0 – 1

Default Value: 0.5

Discussion

When G37 is executed, the distance from the starting position to the theoretical surface position (specified by G37) is broken into two movements at two feedrates. The first movement is executed at the rapid traverse rate for that axis. The second movement is executed at the G37 Speed. The G37 Ratio specifies the fraction of the distance from the starting position to the theoretical surface position at which movement occurs at the rapid traverse feedrate. The remainder of the distance from the starting position to the theoretical surface position occurs at the G37 Speed. [NOTE: The distance from the theoretical surface position to where the surface detection signal is triggered, is always at the G37 feedrate.]

If the G37 Ratio is 0, no movement will occur at the rapid traverse feedrate; the entire movement toward the surface position will be at the G37 Speed. Then the axis will continue at the G37 speed until the surface detection signal is triggered.

If the G37 Ratio is 1, the axis will move the entire distance from the starting position to the theoretical surface position at the rapid traverse feedrate, and assuming no surface detection signal was detected prior to the theoretical surface position, continue moving at the G37 Speed until the surface detection signal is triggered.

If the G37 Ratio is 0.5, the axis will move half the distance from the starting position to the theoretical surface position at the rapid traverse feedrate, then (and assuming no surface detection signal was detected at this point) continue moving at the G37 Speed until the surface detection signal is triggered.

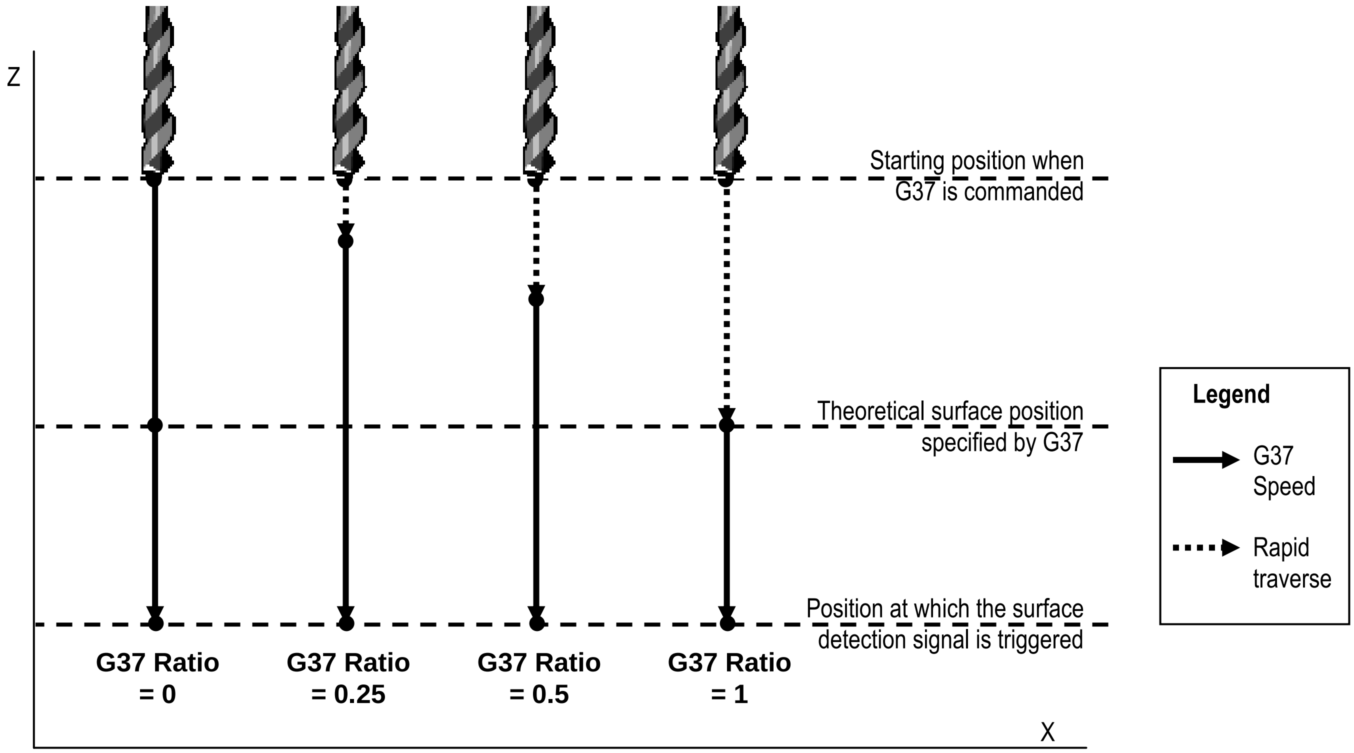


Figure 13-5: G37 Execution Showing Different G37 Ratios

13.3.5 G37 Speed

Description

This parameter specifies the feedrate at which the surface detection signal is searched during automatic tool length compensation calibration (G37).

Measured in Units of: mm/min or inches/min

Range of Valid Values: 0 – 999,999.9 mm/min or 0 – 999,999.9 inches/min

Default Value: 1000.000 mm/min or 39.370079 inches/min

Discussion

Some fraction of the initial movement toward the surface may not be at the G37 Speed, but at the rapid traverse feedrate, as dictated by the G37 Ratio.

13.3.5 G37 Switch Type

Description

This parameter specifies whether the surface detection signal switch for automatic tool length compensation calibration (G37) is active closed or active open.

Measured in Units of: mm/min or inches/min

Valid Values: 0, 1

Meaning of Values

0 – Active Closed
1 – Active Open

Default Value: 0

Chapter 14: Macro Function Parameters and Usage

14.1 Overview of Macros and Customized Macro Calls

The ServoWorks S-100M series (S-100M, S-120M and S-140M) provide a function that allows you to create special, user-defined G, M, S and T codes, wherein the ServoWorks CNC controller can be set up to associate custom G, M, S and T codes with corresponding macro programs.

A macro is one instruction that is included in a part program and that represents a sequence of simpler instructions, known as a macro program. In a way, macro programs are like simple programs or batch files. ServoWorks CNC systems support sophisticated custom macro programs that allow you to use variables; flow control structures such as loops; and mathematical and logical operations. In this way, you can use program logic to eliminate redundant programming of machine functions, such as tool changes. A typical automated tool change (ATC) macro program includes motion commands, M codes and T codes.

Macro programs are written in the ServoWorks macro programming language, which is similar to the BASIC programming language. See the *ServoWorks CNC Macro Programming Manual* for more information.

You can define G, M, S and T codes as specialized codes by using macro parameters to associate these codes with specific macro programs. For S and T codes, the specified macro program applies to ALL S and T codes. For M and G codes, you define specific M and G codes (up to ten of each) that you associate with different macro programs.

When the "Enable Custom G/M/S/T Macro Calls" parameter is set to "Enabled," the ServoWorks RealTime DLL searches part program files before starting G code execution and generates (creates) a temporary file (with a user-defined file name and location), in which every special G, M, S and T code (defined by the macro function parameters) is replaced with a G65 simple macro call to the specified macro program, and each macro program is appended to this temporary file as a subroutine. S and T codes are kept as parameters (arguments) in any new line involving special S or T codes that invoke a macro program.

NOTE: When the "Enable Custom G/M/S/T Macro Calls" parameter is set to "Enabled," the temporary file is created and executed even if there are no special G, M, S or T codes in the part program.

14.2 Advantages of Using Customized G, M, S and T Codes

Custom macros significantly simplify automated processes (such as automatic tool changes) and reduce part programming time by allowing reuse of macro programs, saving time and reducing operator errors. For instance, for ATC, you can have one custom macro for one type of tool change, and just change the variables (such as the tool number) you assign when you call the custom macro.

This allows you to access programs through a network, and run NC programs through a network. Controllers can work together to create a custom macro library (with customized ATC subroutines, for instance) that can be maintained, updated and shared, to save time and money.

14.3 Macro Function Parameters

14.3.1 Enable Custom G/M/S/T Macro Calls

Description

Enabling or disabling of custom G, M, S and T macro calls.

Valid Values: 0, 1

Meaning of Values

0 – Custom G, M, S and T code macro calls disabled
1 – Custom G, M, S and T code macro calls enabled

Default Value: 0

14.3.2 Macro Program Folder (Full Path)

Description

The directory (folder) where all the user-created macro programs related to custom G, M, S or T code macro calls must be stored. The ServoWorks CNC controller will search this folder for all macro programs.

Required Format

You must include the full path for this folder, including the drive.

Example

C:\Program Files\SoftServo\S-100M\ncdata\Macro\

Notes

The path for this folder can be on a local drive or on a network.

Limitations

This folder must already exist. If you type the path of a folder that does not exist, you will get an error message similar to the following:

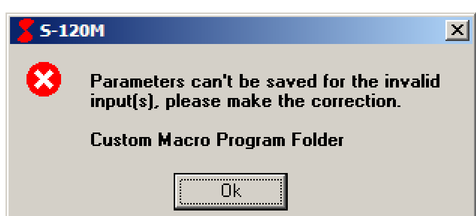


Figure 14-1: Error in Naming Macro Program Folder

14.3.3 Output File Name (Full Path)

Description

The name and location for the temporary NC program file that is created by and executed by the ServoWorks G-Code Parser. This temporary file consists of the part program file with the following changes:

- All lines of code that contain special S, T, M or G codes are replaced with G65 simple macro calls to the corresponding user-defined macro programs
- These user-defined macro programs are appended to this temporary part program file as subroutines

Required Format

You must include the full path for this file, including the drive.

Example

C:\Program Files\SoftServo\S-100M\ncdata\Macro\tempdata.dat

Limitations

- The location for this temporary file must be in a local folder, not on a network or a non-system drive.
- This folder must already exist. If you type the path of a folder that does not exist, you will get an error message similar to the following:

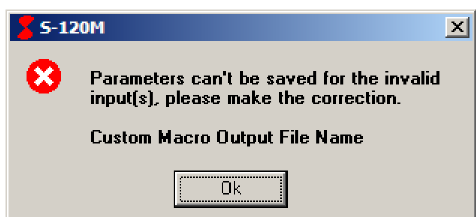


Figure 14-2: Error in Naming Macro Output File Name

Notes

- When the “Enable Custom G/M/S/T Macro Calls” is set to “Enabled,” this temporary file is created whether or not there are any special G, M, S or T codes.
- Because this temporary part program file is executed by the ServoWorks G-Code parser and not the original part program file, M98 blocks will look for subprograms in the folder containing the temporary file and not the original part program file. To have M98 blocks look for subprograms in the folder containing the original part program file, set the output file to be in the same folder as the original part program file.

14.3.4 S Code Setting

Description

The name of the macro program to be called for each instance of an S code in the user's part program.

Required Format

An integer between 1 and 999999999 (with no commas).

Example

3333 (This would cause all S codes to invoke a macro program file named "O3333.dat")

Limitations

- You cannot call a macro program with an S code from within a macro program. All S codes in macro programs are treated as ordinary S codes.

  For example, if S1000 is defined as a custom S code macro call to O1234.dat, if "S500" appears in O1234.dat, it will NOT be treated as a macro call.

- No nesting of calls using S codes is allowed. In other words, if S codes are defined to call a macro program, any S codes in that macro program (subroutine) will be treated as ordinary S codes, and will not call that macro program again.

- All S codes in subprograms called with M98 are treated as ordinary S codes.

Notes

The S code is kept as a parameter (argument) in the new line that invokes the macro program. For example, the line of code "S10" would be replaced with "G65 P3333 S10."

14.3.5 T Code Setting

Description

The name of the macro program to be called for each instance of a T code in the user's part program.

Required Format

An integer between 1 and 999999999 (with no commas).

Example

3333 (This would cause all T codes to invoke a macro program file named "O3333.dat")

Limitations

- You cannot call a macro program with a T code from within a macro program. All T codes in macro programs are treated as ordinary T codes.

  For example, if T1000 is defined as a custom T code macro call to O1234.dat, if "T500" appears in O1234.dat, it will NOT be treated as a macro call.

- No nesting of calls using T codes is allowed. In other words, if T codes are defined to call a macro program, any T codes in that macro program (subroutine) will be treated as ordinary T codes, and will not call that macro program again.

- All T codes in subprograms called with M98 are treated as ordinary T codes.

Notes

The T code is kept as a parameter (argument) in the new line that invokes the macro program. For example, the line of code "T10" in the original NC part program would be replaced with "G65 P3333 T10."

In the O3333.dat macro program, the code should be written using the variable "T#20," because "#20" is the variable number that corresponds to address letter T (see Table 6-2 in the *ServoWorks CNC Macro Programming Manual*). When the parameter "T10" is passed into the O3333.dat macro program as an argument, T#20 will be replaced with T10. In this way, the O3333.dat macro program can support all T codes, with the T codes defined by the tool number called in the original NC program.

14.3.6 M Code Settings

Description

The M code settings actually have two parts:

1) The name of the M code that invokes a macro program file (that defines this custom M code).

2) The name of the macro program to be called for each instance of that specific M code in the user's part program.

Required Format

The format for an M code can be an integer, or an integer with one digit after the decimal point. This is helpful, since you can name special M codes as non-integer numbers to set them apart from normal, predefined M codes which are always an integer value.

The name of the macro program should be an integer between 1 and 999999999 (with no commas).

Example

The name of the M code could be "3" (which defines "M03" as a customized M code), or could be "3.1" (which defines "M03.1" or "M3.1" as a customized M code).

The name of the macro program could be "3333" (which would cause the special M code to invoke a macro program file named "O3333.dat")

Limitations

- No nesting of calls using any specially defined M codes is allowed. In other words, if a special M code is defined to call a macro program, any instances of that same M code in that macro program (subroutine) will be treated as an ordinary M code, and will not call that macro program again.

  However, if a DIFFERENT special M code is used in a macro program (different than the special M code that called that macro program), then it WILL call its associated macro program from within the macro program.

  For example, if M3.1 is defined as a custom M code macro call to O3131.dat, if "M3.1" appears in O3131.dat, it will NOT be treated as a macro call. But if M4.1 is defined as a custom M code macro call to O4141.dat, and M4.1 appears in O3131.dat, it WILL call O4141.dat from O3131.dat.

- All M codes in subprograms called with M98 are treated as ordinary M codes.

14.3.7 G Code Settings

Description

The G code settings actually have two parts:

1) The name of the G code that invokes a macro program file (that defines this custom G code).

2) The name of the macro program to be called for each instance of that specific G code in the user's part program.

Required Format

The format for a G code can be an integer, or an integer with one digit after the decimal point. This is helpful, since you can name special M codes as non-integer numbers to set them apart from normal, predefined M codes which are always an integer value.

The name of the macro program should be an integer between 1 and 999999999 (with no commas).

Example

The name of the G code could be "23" (which defines "G23" as a customized G code), or could be "23.1" (which defines "G23.1" as a customized G code).

The name of the macro program could be "3333" (which would cause the special G code to invoke a macro program file named "O3333.dat")

Limitations

- No nesting of calls using any specially defined G codes is allowed. In other words, if a special G code is defined to call a macro program, any instances of that same G code in that macro program (subroutine) will be treated as an ordinary G code, and will not call that macro program again.

  However, if a DIFFERENT special G code is used in a macro program (different than the special G code that called that macro program), then it WILL call its associated macro program from within the macro program.

  For example, if G23.1 is defined as a custom G code macro call to O231.dat, if "G23.1" appears in O231.dat, it will NOT be treated as a macro call. But if G24.1 is defined as a custom G code macro call to O241.dat, and G24.1 appears in O231.dat, it WILL call O241.dat from O231.dat.

- All G codes in subprograms called with M98 are treated as ordinary G codes.

14.4 ATC Function Example Using Customized G, M and T Macro Calls

Following is an example of an ATC function that uses a customized G, M and T codes to trigger tool changes in the executing part program.

First, the macro function parameters are set as follows:

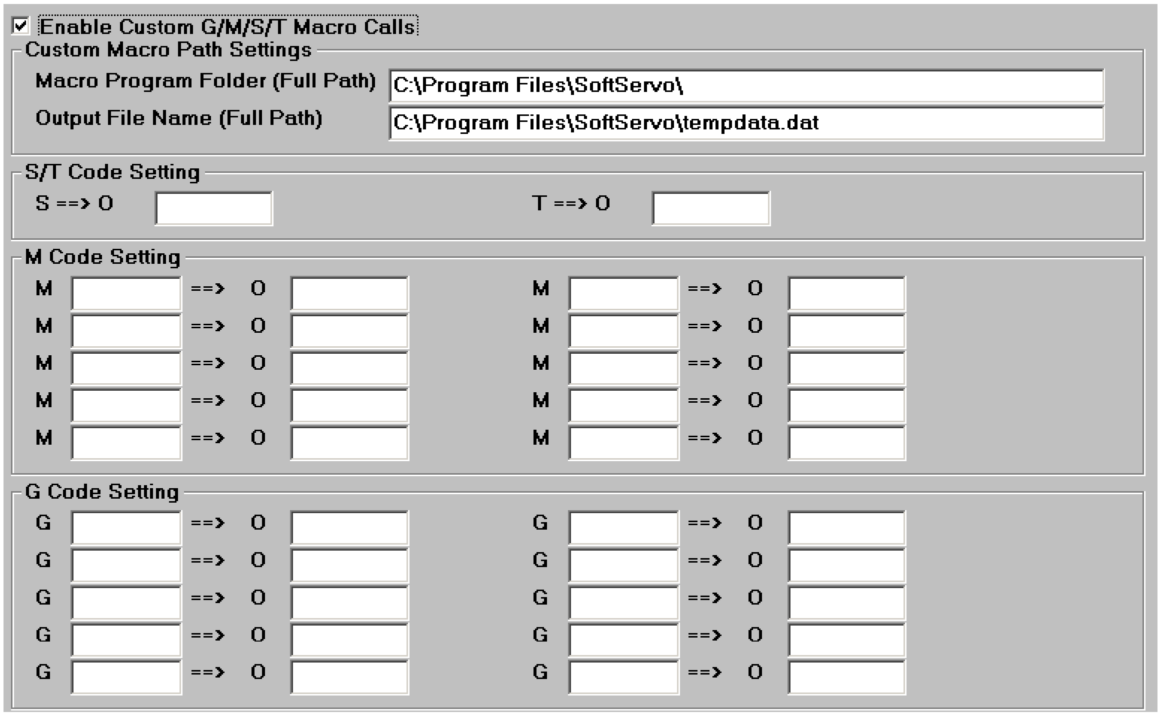


Figure 14-3: Example Macro Parameter Settings – Configuration Mode of the ServoWorks S-140M Series

Then, for each customized G, M, S or T code, you must write a customized macro program (.dat file) that resides in the macro program folder (C:\Program Files\SoftServo\S-100M\ncdata\Macro\ in this case). In this example, the customized macro programs are as follows:

```
G90                        (absolute programming)
G1 X-50.0 Y100.0 F2000     (linear interpolation command)
X-50.00 Y-200.0 F4000
M99                        (return to main program)
```

Figure 14-4: Example O1111.dat File

```
G90                         (absolute programming)
X100.0 Y-200.0
X0.0 Y0.0
M99                         (return to main program)
```

Figure 14-5: Example O2222.dat File

```
G90                         (absolute programming)
M03 S1000                   (spindle clockwise at 1000 RPMs)
M99                         (return to main program)
```

Figure 14-6: Example O3333.dat File

Write a part program file that uses these specialized S, T, M and G codes (G21.2, M03 and T20, in this example). One such example part program follows:

```
G53 X0.0 Y0.0
G92 X0.0 Y0.0
G021.2
X50.0 Y200.0
G1 X-50.0 Y100.0 F2000
X-50.005 Y-200.0 F4000
/G0
X-50.0 Y400.0
X100.0 Y100.0
X100.0 Y100.0
M03
G91G01 X50.0 Y0.002 F500.0
X-50.0 Y-0.001
G 0 X100.0 Y-0.001
G31 X-100.0 Y100.0 F2000.0
X100.0 Y100.0
G 4 P1000
G92 X50.0 Y20.0
G04 X2.0
G90X-100.0 Y400.0
X-50.0 Y100.0
T20
M1
X100.0 Y-200.0
\X-100.0 Y400.0
X0.0 Y0.0
M02
```

Figure 14-7: Example ATCTest.dat File

When the "Enable Custom G/M/S/T Macro Calls" parameter is set to "Enabled," the ServoWorks G-Code Parser searches the ATCTest.dat file before starting G code execution. It generates (creates) a temporary file (with the "Output File Name", in this case C:\Program Files\SoftServo\S-100M\ncdata\Macro\tempdata.dat), in which every special S, T, M or G code is replaced with a G65 simple macro call to the specified macro program, and each macro program is appended to the "tempdata.dat" file as a subroutine, as shown:

```
G53 X0.0 Y0.0
G92 X0.0 Y0.0
G021.2
X50.0 Y200.0
G1 X-50.0 Y100.0 F2000
X-50.005 Y-200.0 F4000
/G0
X-50.0 Y400.0
X100.0 Y100.0
X100.0 Y100.0
M03
G91G01 X50.0 Y0.002 F500.0
X-50.0 Y-0.001
G 0 X100.0 Y-0.001
G31 X-100.0 Y100.0 F2000.0
X100.0 Y100.0
G 4 P1000
G92 X50.0 Y20.0
G04 X2.0
G90X-100.0 Y400.0
X-50.0 Y100.0
T20
M1
X100.0 Y-200.0
\X-100.0 Y400.0
X0.0 Y0.0
M02
```

ATCTest.dat

```
G53 X0.0 Y0.0
G92 X0.0 Y0.0
G65 P1111
X50.0 Y200.0
G1 X-50.0 Y100.0 F2000
X-50.005 Y-200.0 F4000
/G0
X-50.0 Y400.0
X100.0 Y100.0
X100.0 Y100.0
G65 P3333
G91G01 X50.0 Y0.002 F500.0
X-50.0 Y-0.001
G 0 X100.0 Y-0.001
G31 X-100.0 Y100.0 F2000.0
X100.0 Y100.0
G 4 P1000
G92 X50.0 Y20.0
G04 X2.0
G90X-100.0 Y400.0
X-50.0 Y100.0
G65 P2222 T20
M1
X100.0 Y-200.0
\X-100.0 Y400.0
X0.0 Y0.0
M02

O1111
G90
G1 X-50.0 Y100.0 F2000
X-50.00 Y-200.0 F4000
M99

O3333
G90
M03 S1000
M99

O2222
G90
X100.0 Y-200.0
X0.0 Y0.0
M99
```

tempdata.dat

Figure 14-8: Example of ATC: Transformation of ATCTest.dat into the tempdata.dat File

The tempdata.dat file becomes the file actually executed by the ServoWorks G-Code Parser, NOT the ATCTest.dat file.

Chapter 15: Cutting Speed Adjustment Parameters

15.1 Corner Deceleration Control Parameters and Usage

15.1.1 Overview of Corner Deceleration Control

The higher the feedrate when cutting corners, the more rounded off the corners will be due to the effects of smoothing. To get sharp corners without the corner deceleration control function, the feedrate for a part program would need to be determined by the feedrate required at the corners (slowing down the machining process), or the programmer would need to manually program a change in feedrate before and after each corner (complicating the programming process).

Corner Deceleration Control is a function to control the corner cutting feedrate to create sharper, more accurate corners while still maintaining high feedrates away from corners. There are two aspects to this function, as explained below.

When this function is enabled, the angle at the end of each block of code is compared to the angle at the beginning of the subsequent block of code, and the change in angle between the two blocks is calculated. If this change in angle (corner) is less than a user-specified "corner angle," then two things happen:

1) If "corner deceleration" is enabled, the cutting feedrate near the end of the first block (at a user-specified distance from the end of the first block, known as "corner tolerance") is automatically decelerated to a user-specified corner feedrate known as the "corner speed limit." The work-corner is processed at corner speed limit to increase the working accuracy to that of low feedrate processing.

2) If "corner tolerance compensation" is enabled, the tool decelerates at the end point of the first block, and performs an in-position check to make sure the difference between the actual position and the programmed position is less than the "In Position Width" parameter (see *Section 3.2.3: In Position Width*) or the "Corner Tolerance" parameter, before starting the execution of the next block. This is similar to Exact Stop Check Mode, except that rather than performing a position check at the end of every block, the in position check only occurs at corners with a change in angle greater than a user-specified angle.

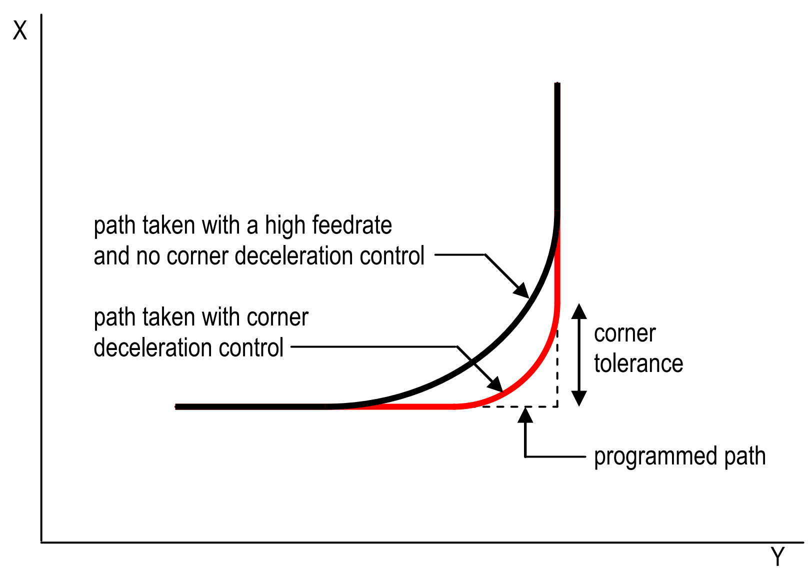


Figure 15-1: Corner Deceleration Control

15.1.2 Limitations of Corner Deceleration Control

The Corner Deceleration Control function can only be used in the XY plane (G17).

The Corner Deceleration Control function checks the angle between two blocks of code when determining whether to decelerate or perform an in-position check. If the NC program consists of many short interpolation segments, there may be cases where Corner Deceleration Control will not trigger when the operator expects it to, due to the corner being traversed across multiple blocks of code. Consider the programmed path in the following figure:

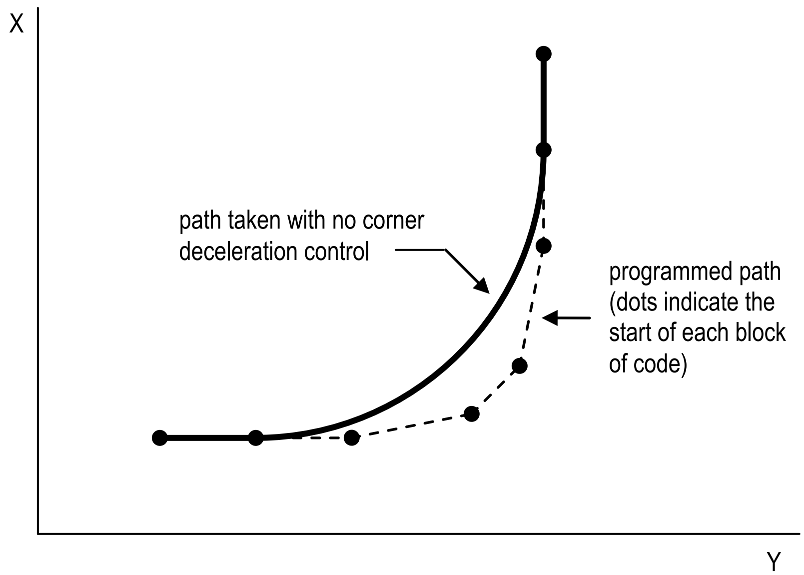


Figure 15-2: Corner Deceleration Control Limitation

This path appears to traverse a 90 degree corner, and the operator may set the “Corner Angle” parameter to 91, expecting Corner Deceleration Control to trigger. However, Corner Deceleration Control will never trigger along this path, as no two consecutive blocks of code form an angle of less than 91 degrees. The “Corner Angle” should be set sufficiently high to handle such cases.

If the above limitations make the Corner Deceleration Control function unsuitable, consider using the DLACC function, which has a similar functionality but without the above limitations (see *Chapter 17: Dynamic Look-Ahead Contour Control Parameters and Usage*).

15.1.3 Corner Deceleration

Description

Enabling or disabling of the corner deceleration control function.

Valid Values: Enabled, Disabled

Meaning of Values

Angle – Corner deceleration control enabled
Disabled – Corner deceleration control disabled

Default Value: Disabled

Warning

This function overlaps with the DLACC function (see *Chapter 17: Dynamic Look-Ahead Contour Control Parameters and Usage*). If you are using the DLACC function, Corner Deceleration must be set to "Disabled." Otherwise, the two functions will interfere and result in an undesirable path.

15.1.4 Corner Angle

Description

The maximum change in angle (between two blocks of code) for which the corner deceleration control will be applied. If the change in angle between two blocks of code is less than or equal to the corner angle, it will not be considered a corner, no change in feedrate will occur as a result of corner deceleration control, and no corner in position check will occur.

The change in angle between the two blocks is calculated by comparing the angle at the end of the block of code to the angle at the beginning of the subsequent block of code, as shown in the following figure. For blocks of code specifying circular movements, the angle is calculated using the tangent to the circular movement.

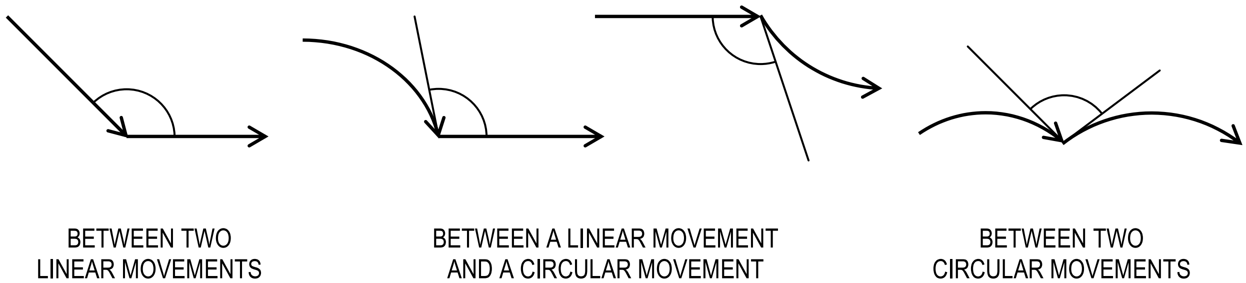


Figure 15-3: Determination of Change in Angle between Two Blocks

Measured in Units of: degrees

Range of Valid Values: 0 ≤ corner angle < 180.0 degrees

Default Value: 90.0 degrees

15.1.5 Corner Speed Limit

Description

This is the maximum coordinated feedrate to be used when turning a corner.

Measured in Units of: mm/min, inches/min

Range of Valid Values: 0 < corner speed limit < Rapid Feedrate (see *Section 3.2.2: Rapid Feedrate*)

Default Value: 500 mm/min, 19.685 inches

Limitations

The corner speed limit must not be zero.

15.1.6 Corner Tolerance Compensation Enable

Description

Enabling or disabling of an in position check at the end of a block that has been determined to be a corner. If the change in angle between two blocks of code is less than or equal to the corner angle, the in position check will be performed at the end of the first block of code comprising that corner, to ensure that corner tolerance has not been exceeded. [See *Section 15.1.7: Corner Tolerance*.]

Valid Values: Enabled, Disabled

Meaning of Values

Enabled – Corner tolerance compensation enabled
Disabled – Corner tolerance compensation disabled

Default Value: Disabled

Notes

This parameter is also referred to as "corner in position check enable."

Warning

This function overlaps with the DLACC function (see *Chapter 17: Dynamic Look-Ahead Contour Control Parameters and Usage*). If you are using the DLACC function, Corner Tolerance Compensation must be set to "Disabled." Otherwise, the two functions will interfere and result in an undesirable path.

15.1.7 Corner Tolerance

Description

This is the vector length along the direction of movement from the programmed position at the end of the block. Unless the block of code commands movement along only one axis, this length will have components in two axes as shown:

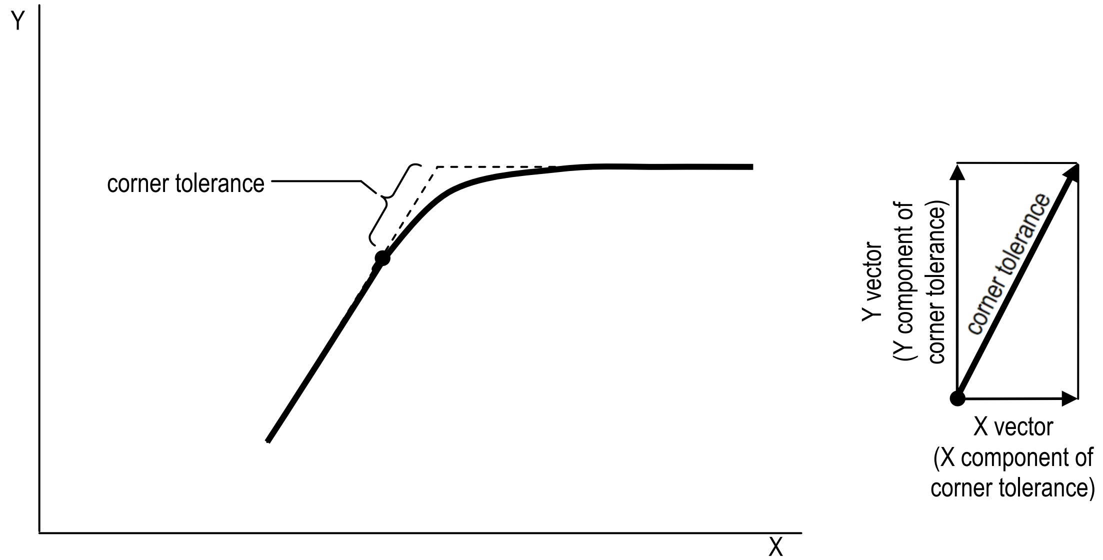


Figure 15-4: Corner Tolerance

Measured in Units of: mm, inches

Range of Valid Values: 0 – 999,999.9 mm, 0 – 999,999.9 inches

Default Value: 0.5 mm, 0.019685 inch

15.2 Velocity Control in Circular Interpolation Parameters and Usage

15.2.1 Overview of Velocity Control in Circular Interpolation

This function automatically adjusts (reduces) circular interpolation speed according to the circle radius and the maximum allowable machine acceleration/ deceleration, to meet the specified trajectory accuracy. The feedrate is clamped during circular interpolation G code commands when the specified feedrate would result in a radial deviation greater than the maximum allowable deviation, determined by an equation that varies depending upon what type of smoothing is specified.

This function is also known as "circular speed clamping."

15.2.2 Technical Explanation of Velocity Control in Circular Interpolation

The formulas to approximate the radial deviation Δ_r vary based on the type of smoothing. The equations that follow using the following nomenclature:

Δ_r: radial deviation (mm)
T_1: smoothing time (sec)
T_2: servo motor time constant (sec)
V: feedrate (mm/sec)
a: radial acceleration (mm/sec^2)

Radial deviation Δ_r for an exponential filter for smoothing is as follows:

$$\Delta_r = \frac{1}{2}\left(T_1^2 + T_2^2\right)\left(\frac{V^2}{r}\right) = \frac{1}{2}\left(T_1^2 + T_2^2\right) a \qquad \textbf{EQ. \#1}$$

Radial deviation Δ_r for a linear filter for smoothing is as follows:

$$\Delta_r = \frac{1}{2}\left(\frac{1}{12} x T_1^2 + T_2^2\right)\left(\frac{V^2}{r}\right) = \frac{1}{2}\left(\frac{1}{12} x T_1^2 + T_2^2\right) a \qquad \textbf{EQ. \#2}$$

Radial deviation Δ_r for a bell-shaped filter for smoothing is as follows:

$$\Delta_r = \frac{1}{2}\left(\frac{1}{24} x T_1^2 + T_2^2\right)\left(\frac{V^2}{r}\right) = \frac{1}{2}\left(\frac{1}{24} x T_1^2 + T_2^2\right) a \qquad \textbf{EQ. \#3}$$

However, the value T_2 could theoretically be approximated to 0 if the servo drives' position loop gain is infinitely increased using velocity feedforward and other methods. This means that if we let the servo drives' position loop gain be infinite, the above equations can be reduced to the following:

Radial deviation Δ_r for an exponential filter for smoothing is reduced as follows:

$$\Delta_r = \frac{1}{2}\left(\frac{T_1^2 V^2}{r}\right) = \frac{1}{2} T_1^2 a \qquad \textbf{EQ. \#4}$$

Radial deviation Δ_r for a linear filter for smoothing is reduced as follows:

$$\Delta_r = \frac{1}{24}\left(\frac{T_1^2 V^2}{r}\right) = \frac{1}{24} T_1^2 a \qquad \textbf{EQ. \#5}$$

Radial deviation Δ_r for a bell-shaped filter for smoothing is reduced as follows:

$$\Delta_r = \frac{1}{48}\left(\frac{T_1^2 V^2}{r}\right) = \frac{1}{48} T_1^2\, a \qquad \textbf{EQ. \#6}$$

From the above equations, the maximum radial acceleration "A" (mm/sec^2) with respect to the maximum radial deviation "Δ_R" (mm) can be calculated using the following equations:

Maximum radial acceleration for an exponential filter for smoothing is as follows:

$$A = \frac{2\Delta_R}{T_1^2} \qquad \textbf{EQ. \#7}$$

Maximum radial acceleration for a linear filter for smoothing is as follows:

$$A = \frac{24\Delta_R}{T_1^2} \qquad \textbf{EQ. \#8}$$

Maximum radial acceleration for a bell-shaped filter for smoothing is as follows:

$$A = \frac{48\Delta_R}{T_1^2} \qquad \textbf{EQ. \#9}$$

In our velocity control for circular interpolation function, we use this maximum acceleration as a parameter, and clamp the feedrate during circular interpolation according to the following equation:

$$V = \sqrt{A\,r} \qquad \textbf{EQ. \#10}$$

To reiterate:

V: feedrate (mm/sec)
A: maximum radial acceleration (mm/sec)
r: circle radius

15.2.3 Velocity Control in Circular Interpolation Enable

Description

Enabling or disabling of the velocity control in circular interpolation function.

Valid Values: Enabled, Disabled

Meaning of Values

Enabled – Velocity control in circular interpolation enabled
Disabled – Velocity control in circular interpolation disabled

Default Value: Disabled

Warning

This function overlaps with the DLACC function (see *Chapter 17: Dynamic Look-Ahead Contour Control Parameters and Usage*). If you are using the DLACC function, Velocity Control in Circular Interpolation must be set to "Disabled." Otherwise, the two functions will interfere and result in an undesirable path.

15.2.4 Maximum Acceleration

Description

This value is the maximum radial acceleration limit. It is used in conjunction with the radius of the circle to determine the reduced velocity for circular interpolation. By adjusting this parameter, you can adjust the maximum radial deviation during circular interpolation. [NOTE: Maximum Acceleration is denoted as "A" in Equation #10.]

Measured in Units of: mm/sec^2, inches/ sec^2

Range of Valid Values: 0 – 999,999,999.9 mm/ sec^2, 0 – 999,999,999.9 inches/ sec^2

Default Value: 100 mm/sec^2, 3.937 inches/sec^2

15.2.5 Minimum Feedrate

Description

In the case of a very small specified radius during circular interpolation, it is possible that the velocity control in circular interpolation function could reduce the circular interpolation to an extremely low and undesirable velocity. This parameter allows you to set limits on how low the feedrate can be reduced. The velocity control in circular interpolation function is limited by this minimum feedrate – the feedrate during circular interpolation can't be reduced to less than this minimum feedrate.

Measured in Units of: mm/min, inches/min

Range of Valid Values: 0 – 999,999,999.9 mm/min, 0 – 999,999,999.9 inches/min

Default Value: 1,000 mm/min, 39.370 inches

15.2.6 Warnings Regarding Velocity Control in Circular Interpolation

To maintain high accuracy during circular interpolation using this function, you must not rely wholly on the simple equation (Equation #10) used to calculate the clamped feedrate, but tune the machine using the "maximum acceleration" parameter, known as "A." Note that we assumed that T_2 was equal to zero in Equations #7, 8 and 9. However, it is physically impossible for T_2 to be actually zero. What is usually called an optimally tuned servo drive is one for which the position loop time constant is optimized, not one for which the position loop time constant is zero. Using servo drives optimally tuned for sophisticated velocity feedforward functions, you can make the time constant close to zero, but never equal to zero.

In addition, errors and deviations during cutting are attributable not only to CNC and to servo drives, but also to the stiffness and the straightness of the machinery itself. The formulas that our function is based on – Equations #1, 2 and 3 – are also only approximations, and if the feedrate (V) decreases, this approximation becomes weak. Thus, it is important that you rely not only on these equations, but actually tune this function by trying a series of values for the "maximum acceleration" parameter to obtain the optimum result for your machine.

15.2.7 Calculation Example of Velocity Control in Circular Interpolation

The following example assumes these conditions:

- Smoothing Type = Exponential
- Smoothing Time = 200 ms
- Radius = 100 mm
- Goal: maximum radial deviation of 0.1 mm

From Equation #1,

$$A = 2\left(\frac{r}{T_1^2 + T_2^2}\right) = 2\left(\frac{0.1\text{ mm}}{(0.2\text{ sec})^2 + 0^2}\right) = 5\text{ mm/sec}^2 \qquad \textbf{EQ. \#11}$$

From the above equation, we obtain the value for which the maximum acceleration parameter should be set. Setting it to the above value of "5," the cutting feedrate will be clamped to speed calculated as follows:

From Equation #10,

$$V = \sqrt{(5\text{ mm/sec}^2)(100\text{ mm})} = 22.36\text{ mm/sec}$$

$$= 1342\text{ mm/min} \qquad \textbf{EQ. \#12}$$

Chapter 16: Normal Direction Control

16.1 Overview of Normal Direction Control

Normal direction control is a function that keeps the tool oriented at a constant angle relative to the cutting direction.

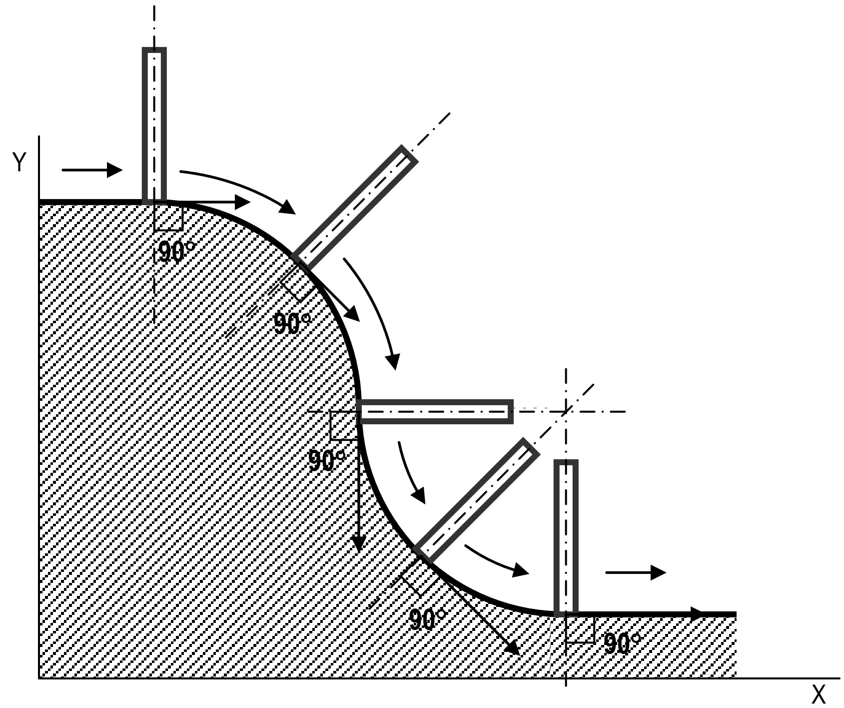


Figure 16-1: Sample Tool Movement during Normal Direction Control

Without normal direction control, if you wanted the tool to point at a constant angle relative to the cutting direction, you would have to manually insert commands to rotate the tool axis into linear and circular interpolation G codes, by finding the angle that the tool will face before and after interpolation, in addition to the direction the tool should turn during the interpolation, for each line of G code. With normal direction control, you can write NC programs that will automatically insert the appropriate rotation parameters into any interpolation G codes that are commanded.

G40.1: Normal direction control cancel
G41.1: Normal direction control left
G42.1: Normal direction control right

The angle for normal direction control is measured from the positive X axis, in the counterclockwise direction, as shown in the figure below:

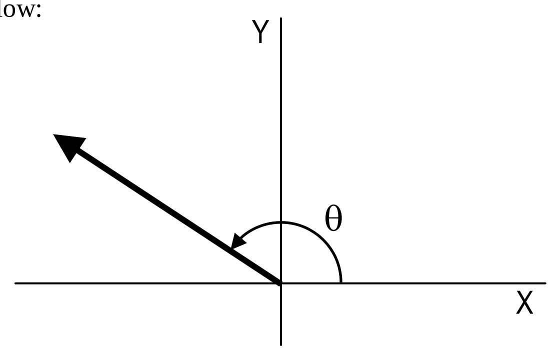


Figure 16-2: Definition of Angle for Normal Direction Control

The “left” and “right” in the two modes of normal direction control designate the side the tool is at, relative to the substrate being cut, from the perspective of looking in the direction of motion. For example, when cutting straight eastwards using G01X100, if the tool is on the north and the substrate on the south, G41.1 should be used, while if the tool is on the south and the substrate on the north, G42.1 should be used. When normal direction control is on, the tool face will be perpendicular to the cutting direction at all times.

G40.1 is the default mode when none of the G40.1, G41.1 and G41.1codes has been programmed. The required format in a part program is as follows:

```
G41.1 H##
...
G40.1
```

OR

```
G42.1 H##
...
G40.1
```

16.2 Normal Direction Control Parameters

16.2.1 Rotary Axis No.

Description

This number designates which axis controls the angle of the tool.

Valid Values: 1, 2, 3, 4, 5, 6, 7, 8

Meaning of Values

1 – Axis 1
2 – Axis 2
3 – Axis 3
4 – Spindle Axis
5 – Axis 4
6 – Axis 5
7 – Axis 6
8 – Axis 7

Default Value: 5

Limitation

The "Axis type" of the tool must be "Rotary" or "Rotary ST."

Warning

Take care to note that setting "Rotary Axis No." to "7" does not indicate Axis #7, it selects Axis #6. (See above.)

16.2.2 Rotation Feedrate

Description

This specifies the feedrate in effect when normal direction control causes the rotary axis to rotate.

Measured in Units of: degrees/min

Range of Valid Values: 0 – Rapid Feedrate

Default Value: 1000.0 degrees/min

Discussion

Note that this parameter is only used when the rotary axis rotates in position, such as at a sharp corner or the starting point of an arc. When there is continuous rotation together with tool movement, which normally happens during circular interpolation, the X and Y axes feedrates override this parameter.

16.2.3 Angle Limit

Description

This specifies the minimum angle of rotation that can be added to a block by the normal direction control function. All angular rotations less than the Angle Limit are ignored.

Measured in Units of: degrees

Range of Valid Values: 0 – 360 degrees

Default Value: 10.0 degrees

Discussion

This parameter is useful when small rotations will not affect the operation of the machine tool and can be ignored. Note that the ignored angles accumulate in that the angle of rotation becomes larger and larger as more angles are ignored for sequences of angles of the same concavity.

For example, when cutting an octagon from the outside, the tool normally would rotate 45° per corner. If the Angle Limit is set to 60°, the first corner is ignored because a rotation of 45° is less than 60°. However, on the second corner, the tool rotation would now be 90° since the first rotation was ignored and hence the tool must rotate 45° + 45° to achieve the target orientation. Thus, a 90° rotation is performed on this second corner.

NOTE: An Angle Limit above 180° will have the same effect as an Angle Limit of 180°.

Example Demonstrating the Effects of Various Values of the Angle Limit Parameter

The following example demonstrates the effects of varying values for the Angle Limit parameter on the rotations that the tool undergoes while cutting.

The shape shown in the following figure will be cut using various values for the Angle Limit parameter. The shape is cut in the clockwise direction starting from the point marked “Start.”

In the first instance, we will demonstrate what happens when the Angle Limit is set to 0° (see the following figure). At the starting position, the tool is pointing in the 0° direction. First, the normal direction control function moves the rotary axis to 270° before drawing the first line. After the first line is drawn, the rotary axis rotates 45° to the 225° position to place the tool perpendicular to the second line. At the end of the second line, the tool is rotated 135° to the 90° position. At the end of the third line, the tool is rotated 90° to the 0° position. At the end of the fourth line, the tool returns to the starting position.

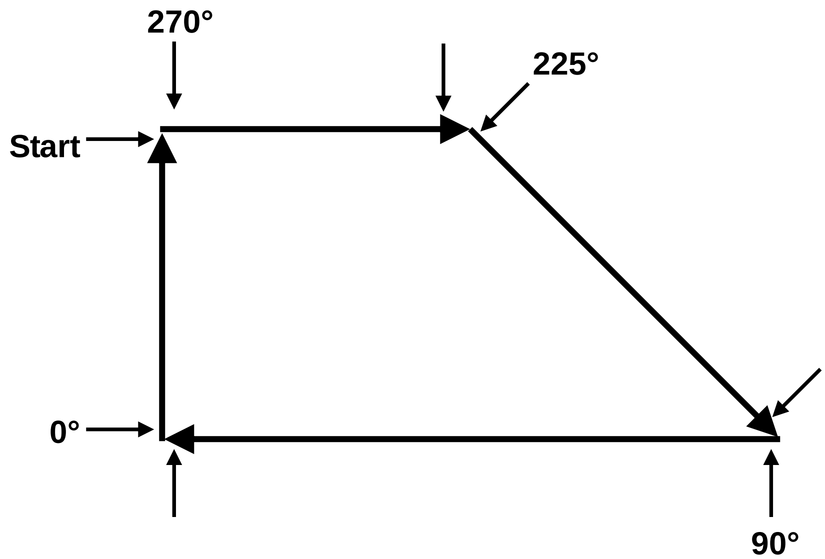


Figure 16-3: Normal Direction Control Example of Angle Limit – Figure 1 of 4

Since no rotations less than 45° are used to cut this shape, an Angle Limit value of less than 45° will have no effect, cutting the shape in the manner described above.

Let's consider a second instance when the Angle Limit is set to 46° (shown in the following figure). The rotation between the first and second line is ignored since it is only 45°. At the corner of the second and third lines, the tool rotates 180° from the 270° position to the 90° position. The rotation at the corner of the third and fourth lines remains unaffected, since the rotation is 90°, greater than the 46° Angle Limit.

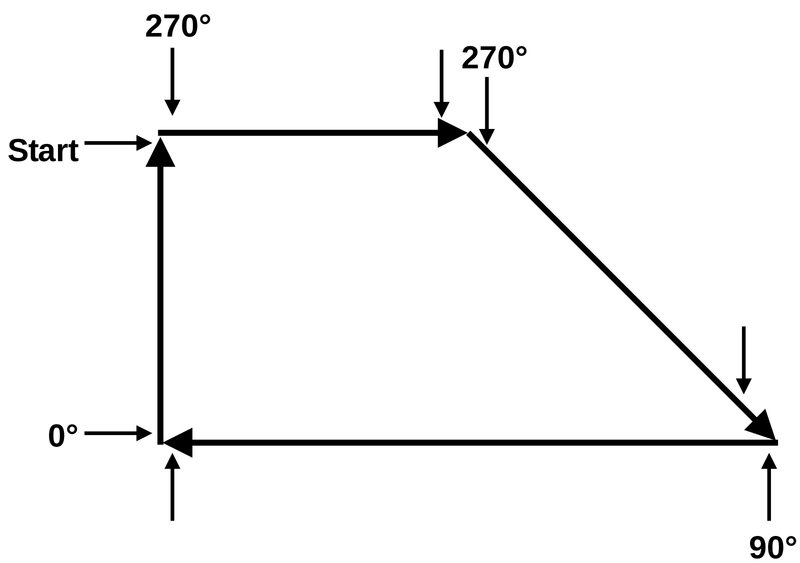


Figure 16-4: Normal Direction Control Example of Angle Limit – Figure 2 of 4

Let's consider a third instance when the Angle Limit is set to 91°. An Angle Limit of 91° is slightly more complicated. Since the first rotation at the start of the first line is a 90° rotation, the rotation is ignored. This means that at the corner of the first and second lines, where there normally is a 45° rotation, the tool must rotate an extra 90° for a total of 135°. Since 135° is greater than the Angle Limit of 91°, a rotation is made at that point. At the corner of the second and third lines, a 135° rotation is performed. At the corner of the third and fourth lines, a 90° rotation normally occurs, but since 90° is less than the Angle Limit, no rotation occurs. At the end of the run, the rotary axis faces 90°. This example is a strange case, probably only useful for illustrative purposes.

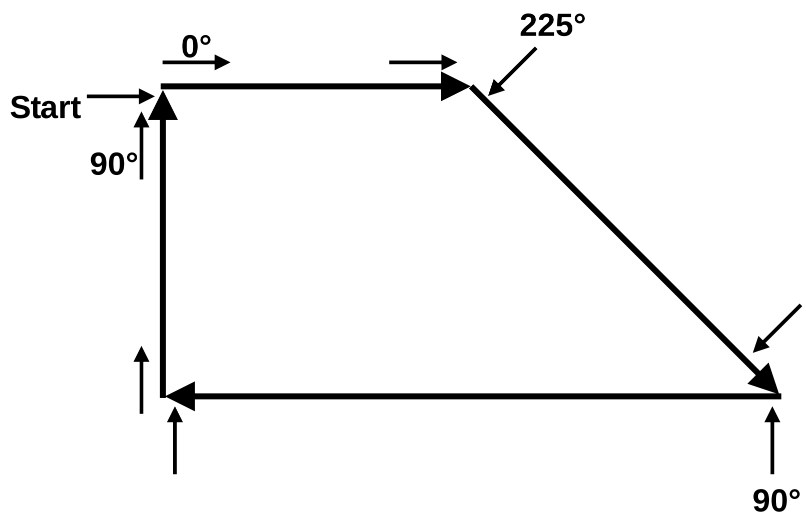


Figure 16-5: Normal Direction Control Example of Angle Limit – Figure 3 of 4

Finally, let's consider a fourth instance when the Angle Limit is set to 181°. We will get no angle movement at all. This is because all angle movements can be performed in 180° or less (if we take the shorter path), so at no point can there be a rotation of over 181°.

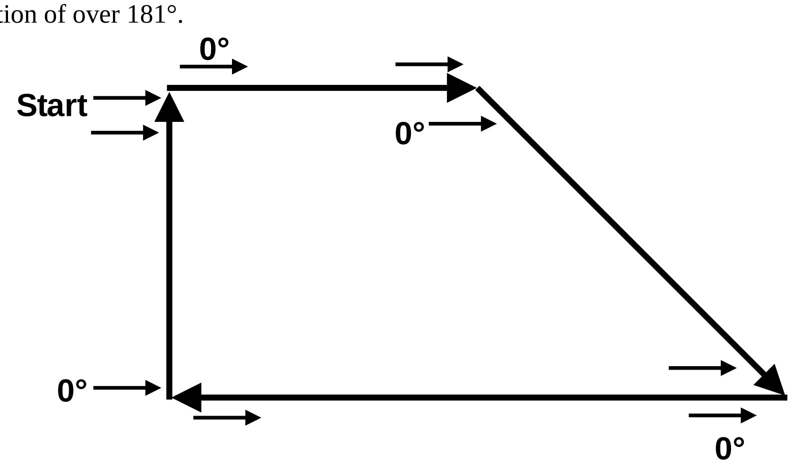


Figure 16-6: Normal Direction Control Example of Angle Limit – Figure 4 of 4

16.2.4 Length Limit

Description

This specifies the length of a line (including ones cut using circular interpolation) for which angular rotation should occur. For G code blocks that cut a shorter distance than the Length Limit, no angular rotation will be inserted by normal direction control function.

Measured in Units of: mm, inches

Range of Valid Values: 0 – 999,999.9 mm or 0 – 999,999.9 inches

Default Value: 10.0 mm

Discussion

This parameter is very similar to the Angle Limit parameter.]

16.3 Examples Demonstrating the Difference Between Normal Direction Control Left and Normal Direction Control Right

Assume G41.1 is called, and then a circle is cut. The arc starts at the top of the circle and continues clockwise to the 10 o'clock to 11 o'clock position of the circle. The arrows on the figure indicate the direction that the machine tool is pointing. First, after the machine moves to the starting point of the arc, the rotary axis is rotated to 270° to position the tool perpendicular to the arc. Notice that the tool is now situated so that it is pointing toward the arc from the left side of the view toward the direction of movement. Normal direction control modifies the block drawing the arc appropriately so that the rotary axis rotates in a way that it is always pointing perpendicular to the arc, as labeled in the following figure. At the end of the arc, the tool has rotated to the 315° position.

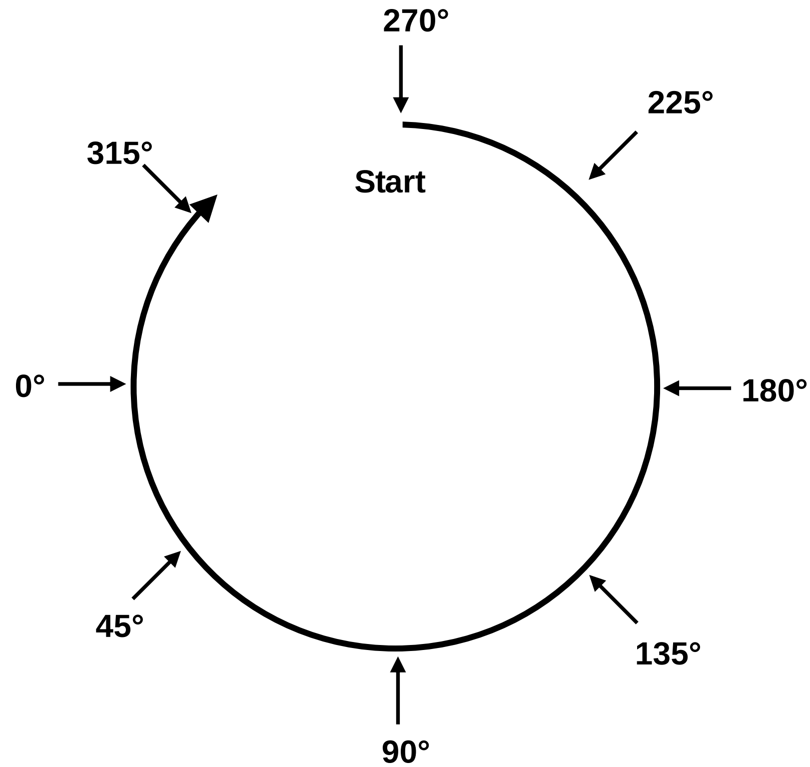


Figure 16-7: Normal Direction Control, Example of G41.1

What happens if we cut the same arc using G42.1 instead of G41.1? The tool is designated to be at the right side of the view toward the direction of movement. This means that the arc is cut from the inside. At the starting point, the rotary axis rotates to 90°, and as the arc is cut, the angle decreases to 0° and further down from 360° to 135°, where the arc is completed. Notice that at all times, the tool is facing perpendicular to the substrate.

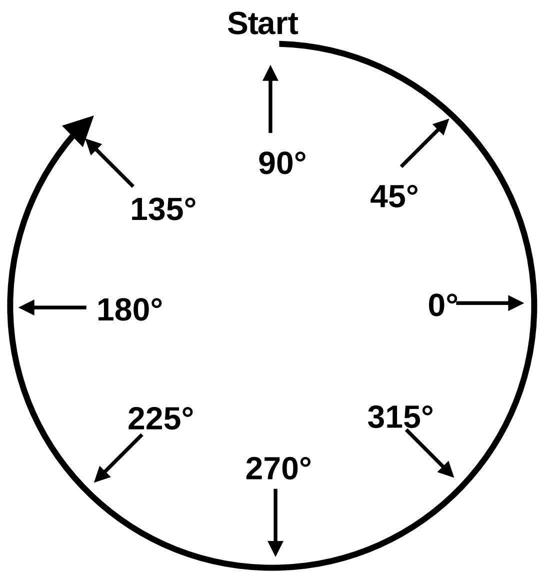


Figure 16-8: Normal Direction Control, Example of G42.1

It is important to realize that G42.1 does not necessarily always cut inside the shape and G41.1 the outside of the shape. Indeed, if we cut the arc in the other direction, from the 10 o'clock to 11o'clock direction counterclockwise to the 12 o'clock position, G41.1 would position the tool inside the arc and G42.1 would position the tool outside the arc.

Chapter 17: Dynamic Look-Ahead Contour Control Parameters and Usage

17.1 Overview

17.1.1 DLACC Description

Soft Servo Systems' three-dimensional Dynamic Look-Ahead Contour Control (DLACC) is designed for high-speed, precision machining. The main advantage of DLACC is to generate an actual tool path that follows the programmed path with dramatically improved accuracy over conventional methods, while at the same time optimizing the feedrate along the tool path for the maximum possible feedrate with controlled precision. The use of Soft Servo's Systems' DLACC function results in the ability to run bigger and heavier machines at faster feedrates with higher precision.

In the conventional method of generating a tool path (no DLACC), there are two important factors at work:

1) Interpolation among multiple moving axes for a programmed tool path.

2) Smoothing filters for proper acceleration and deceleration of each axis (the programmed path is put through a "smoothing filter" to avoid jerky motion).

In general, in a normal cutting process, a long smoothing time will result in smooth motion, but will also result in inaccuracy of the tool trajectory due to different acceleration/deceleration among axes most of the time, especially for very complex tool paths. A shorter smoothing time will result in unsmooth and shaky motion due to all kinds of limitations of any given machine.

There are always conflicts between these two factors of multi-axis interpolation and smoothing filters. The smoothing filter conveniently controls acceleration/ deceleration for each axis with a smoothing time constant. But the drawback of the smoothing filter is that the movement of each axis is delayed without coordination with other axes. In other words, the integrity of interpolation between each axis is broken, and the accuracy of the tool path deteriorates.

Let's take a simple example of a corner in the X-Y plane:

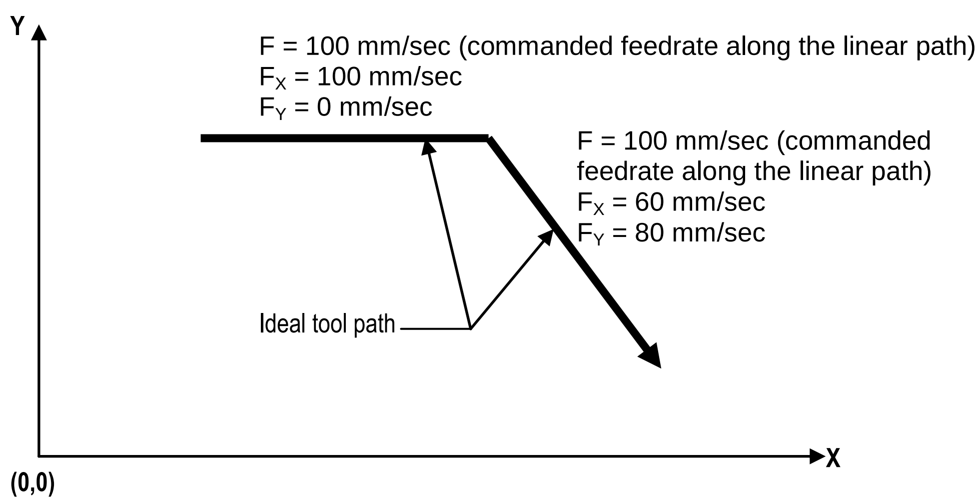


Figure 17-1: Example of Different Accelerations/Decelerations for Different Axes at a Corner

At the corner, the two axes will have significantly different accelerations/decelerations. Axis X will be decelerating from 100 mm/sec to 60 mm/sec, and Axis Y will be accelerating from 0 to 80 mm/sec. Even if the X and Y axes each have the same smoothing time (let's say 10 ms) and the same linear smoothing mode, X experiences a deceleration of 4000 ms/sec^2 while Y experiences an acceleration of 8000 ms/sec^2. This will result in a less accurate tool path, as the interpolation between the X and Y axes is lost at the corner. If the two axes have different smoothing times and/or smoothing modes, this further deteriorates the interpolation between the two axes at the corner.

Soft Servo Systems' advanced DLACC technology effectively solves the conflict between multi-axis interpolation and smoothing filters, and produces accurate tool paths with optimized feedrates. The acceleration/deceleration of each axis is controlled along the programmed path, i.e. coordinated with interpolation, resulting in an accurately generated tool path that faithfully follows the programmed path without jerky motion, and optimized acceleration/deceleration of each axis to allow the maximum cutting feedrate.

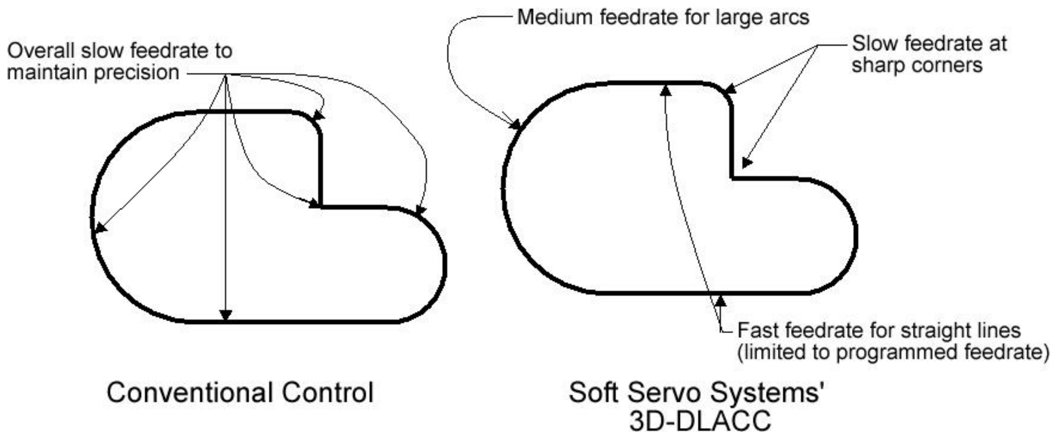


Figure 17-2: Comparison of Soft Servo Systems' DLACC with Conventional Controls (2 of 2)

A circle can be thought of as a series of corners. A programmed circle is broken into numerous tiny linear segments; between each linear segment is a corner. The following figure depicts tool path accuracy for a circle with and without DLACC:

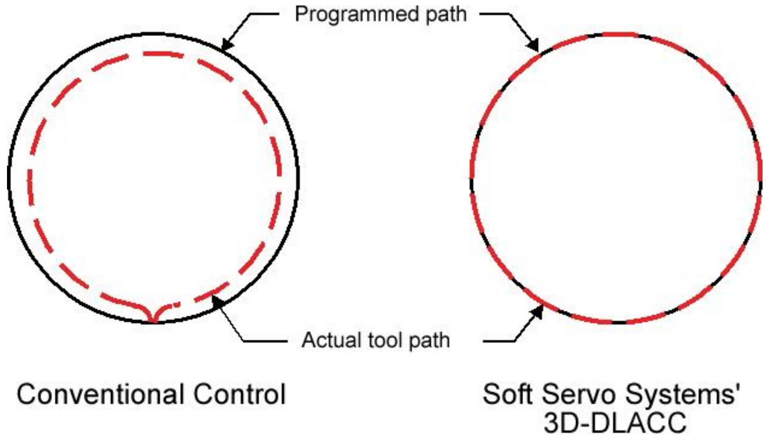


Figure 17-3: Comparison of Soft Servo Systems' DLACC with Conventional Controls (1 of 2)

17.1.2 DLACC Advantages

The DLACC developed by Soft Servo Systems solves problems with close points, corner rounding, gouging and overshooting:

1) Three-dimensional surfaces and complex, free-form shapes can be designed with CAD/CAM to produce a motion program consisting of many, many linear movements over a fine mesh of very closely spaced points to produce a highly finished surface. With such tiny movements, and closely bunched points, there are constant slight direction changes that must be made in rapid succession. These small linear movements can be optimized with DLACC to move the tool over the milling surface as quickly as possible while not sacrificing accuracy.

2) For motion program commands using circular, helical or exponential interpolation, or milling long linear segments, DLACC provides calculated deceleration at corners and other changes in direction, to eliminate corner rounding and to get the most accurate possible tool path.

3) DLACC produces sharp corners by preventing overshooting of the tool path with deceleration as necessary for changes in direction (while still maximizing feedrates on long segments), so the tool isn't moving too quickly to "make the corner." Without DLACC, there is the potential for a tool that is moving at a high feedrate to overshoot the tool path when changing direction, leading to gouging of the part – one block of data is executed before the previous block of data is quite complete because the tool is moving too quickly. If the control looks ahead and sees no deviation in the upcoming tool path, no deceleration is required. If the control looks ahead and sees a 90º deviation in the upcoming tool path, it may decelerate the tool to a total or near total stop to produce an accurate corner.

The major advantages of Soft Servo Systems' DLACC are as follows:

1) Acceleration/deceleration along the programmed path (before interpolation), which ensures the highest possible tool path accuracy at a given interpolation cycle time. DLACC has an advanced and complicated algorithm for acceleration/deceleration before interpolation, maximizing speed and making the best use of the machine's Maximum Acceleration/Deceleration, resulting in higher performance.

2) DLACC looks ahead 500 to 2000 cycles: for a 1 ms interpolation rate (i.e. the VersioBus II interface system), this results in up to 2 seconds of preprocessed look-ahead, with 1000 cyclic blocks every second. For a 4 ms interpolation rate (MECHATROLINK interface system for ServoWorks CNC products), this results in up to 8 seconds of preprocessed look-ahead, with 250 blocks every second (or 2000 blocks every 8 seconds). [See Table 17-1 for more examples.] This configurable number of cycles allows you to fine-tune the DLACC preprocessing time for your system, resulting in greater efficiency and more consistent performance. This allows you to avoid looking too far ahead (wasting computing power), and allows you to make sure DLACC is looking ahead long enough to decelerate enough when needed.

3) Adjustable look ahead smoothing filter before interpolation. As discussed, smoothing filters, necessary in the absence of DLACC, deteriorate the precision of multi-axis interpolation and tool path accuracy. While the look ahead smoothing filter before interpolation has the same effect of losing accuracy, it also provides the benefit of a higher feedrate because of the smoothed-out sharp corners in the tool path. Specifically, a long Look Ahead Acc/Dec Time results in smoother motion and faster feedrates, with a less accurate tool path; a shorter Look Ahead Acc/Dec Time results in a more accurate tool path, with slower feedrates. To reach the highest possible accuracy while still maximizing feedrates whenever possible, the smoothing filter for DLACC can be set to "None" (not recommended) or set to a very short smoothing time.

   Such flexibility – allowing a choice among prioritizing accuracy, prioritizing speed, or finding a balance between accuracy and speed that optimizes cutting efficiency (increasing speed by sacrificing some accuracy within an allowable tolerance) – provides a convenient tool for the experienced user. Automatic feedrate control maximizes the trajectory feedrate within the constraints of Maximum Acceleration/ Deceleration Rate and the maximum speed of each axis.

17.1.3 How DLACC Works

In a normal cutting process (no DLACC), one G-code block is processed, and executed immediately, before the next block is processed or executed.

With DLACC technology, we introduce the concept of "cyclic blocks." Cyclic blocks are different from G-code blocks, which are individual lines of code in a G-code program. Soft Servo Systems' DLACC works by processing one G-code block and breaking that G-code block down into many cyclic blocks (as determined by the Look Ahead Smoothing Buffer Size parameter). These cyclic blocks are smaller blocks, and will not be executed immediately. Rather, the cyclic blocks are put into a buffer and reviewed again for optimal feedrate and tool trajectory, evaluated according to DLACC parameters. Typically, the cyclic blocks will be consumed (executed) in one cycle, when they are pushed out of the cyclic block buffer.

DLACC works as follows:

1) The G-code blocks are processed and cyclic blocks are generated by breaking the movement commanded by the G-code block into cyclic blocks. Each cyclic block is the component of the G-code block that can be completed in one cycle.

   As an extremely simple single-axis example, for a 1 ms interpolation rate, cyclic block buffer size of 2000, and the G-code block "G01 X10.0 F100," to move axis X 10.0 mm at 100 mm/min takes 6000 cycles:

$$\frac{10.0\text{ mm}}{\left(\frac{100\text{ mm}}{\text{min}}\right)\left(\frac{1\text{ min}}{60\text{ sec}}\right)\left(\frac{1\text{ sec}}{1000\text{ ms}}\right)\left(\frac{1\text{ ms}}{1\text{ cycle}}\right)} = 6{,}000\text{ cycles}$$

   That's 6000 cyclic blocks, with each cyclic block initially equivalent to 10 mm/6000 cyclic blocks = 1.66666 µm. The first 2000 cyclic blocks are put into the cyclic block buffer.

2) These cyclic blocks are revised (recalculated) by going through a look ahead smoothing filter as an initial estimation of the tool path. The Look Ahead Acc/Dec Time parameter (usually a very small value) is used to form an ideal tool trajectory around a corner without considering acceleration or deceleration. NOTE: The ideal tool trajectory is not usually the same as the programmed tool trajectory, as shown in the following figure:

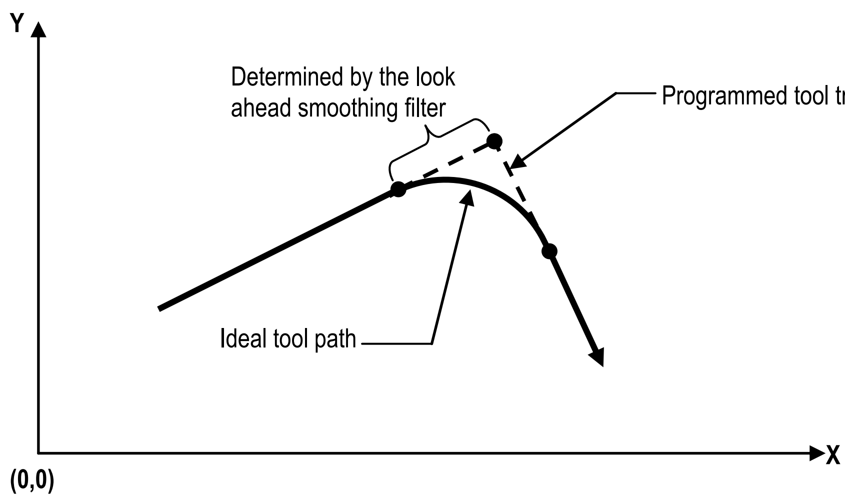


Figure 17-4: Ideal Tool Path

3) The cyclic blocks are stacked up in a cyclic block buffer that holds as many as 2000 cyclic blocks. [This number is configurable, and is determined by the Look Ahead Smoothing Buffer Size parameter.]

4) The DLACC algorithm is constantly checking the acceleration/deceleration rate of each axis for each cyclic block, to determine if the Maximum Acceleration/Deceleration parameter has been exceeded. [This would typically indicate a corner or a sharp turn or a circle.] If the Maximum Acceleration/Deceleration parameter *has* been exceeded, then the DLACC algorithm works backwards from that cycle, through the cyclic block buffer, adjusting acceleration or deceleration in previous blocks until the programmed path no longer exceeds the Maximum Acceleration/Deceleration parameter, and calculating an optimal feedrate along the designated trajectory, thus keeping the accuracy of the tool path.

5) After the cyclic block buffer is full, the execution of these cyclic blocks begins. Every executed cyclic block is replaced with a new cyclic block immediately, to keep the pre-processed cyclic block buffer full.

6) Every time a new cyclic block is filled in, it will be evaluated in consideration with existing blocks in the cyclic block buffer to calculate an optimal feedrate along the designated trajectory. If modification is necessary because the Maximum Acceleration/Deceleration is exceeded along any axis, the feedrate is recalculated to attain an acceptable acceleration/deceleration rate along the programmed path, i.e. coordinated with interpolation.

7) DLACC executes modal or coordinate G-code commands (F, G90/G91, G54, etc.), and mixes them in the next block using block rollover, to avoid any interruption of the motion by keeping the cyclic block buffer full.

8) DLACC executes M codes or nonmodal G code commands (such as G04, dwell at full stop) only after the Smoothing End Check Signal is given, so as to complete any motion from previous blocks <u>before</u> executing M codes or nonmodal G codes.

   The Smoothing End Check Signal is used by the DLACC algorithm to keep track of when smoothing is essentially completed. Smoothing filters cause motion to take longer to be completed than it would without a smoothing filter; in general, the motion in the smoothing filter needs to be completed before M codes and nonmodal G code commands are executed. However, sometimes – particularly with exponential smoothing filters – smoothing takes too long to wait for it to be completed in its entirety before executing subsequent M code or nonmodal G code commands. (See *Section 17.3.4: Exponential Smoothing Mode* for a graphic that demonstrates this.) For this reason, DLACC provides a parameter you can use called the Look Ahead Smoothing End Check Limit. This value is a distance that determines when it's okay to give the Smoothing End Check Signal, even before the entire motion in the smoothing filter is completed. [<u>NOTE</u>: Even after the Smoothing End Check Signal is given, this distance left in the smoothing filter is still completed, even while an M code or nonmodal G code is being executed.]

17.1.4 Illustrations of DLACC Concepts

DLACC uses block rollover to avoid interruption of motion, as shown:

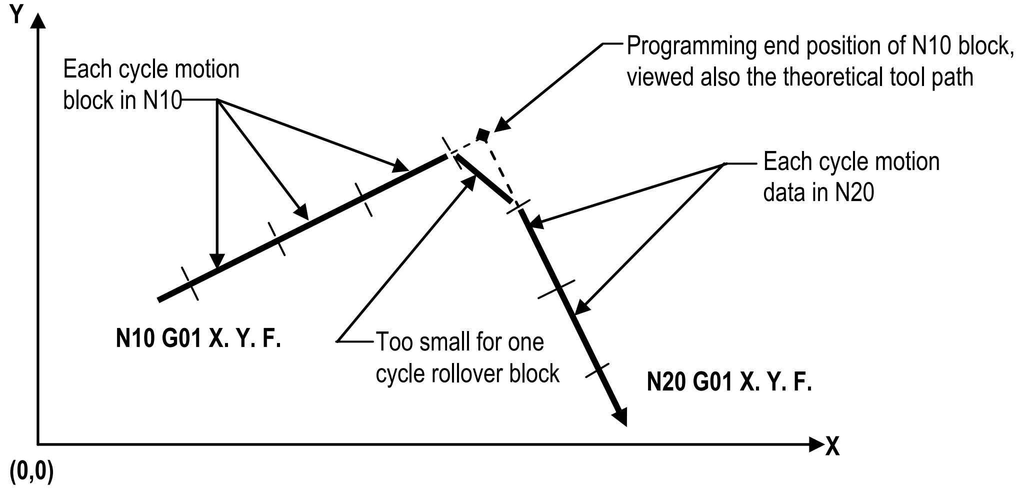


Figure 17-5: Illustration of Block Rollover in DLACC Execution, Before the Look Ahead Smoothing Filter is Engaged

DLACC uses the look ahead smoothing filter to calculate an ideal tool path that is different than the theoretical programmed tool path:

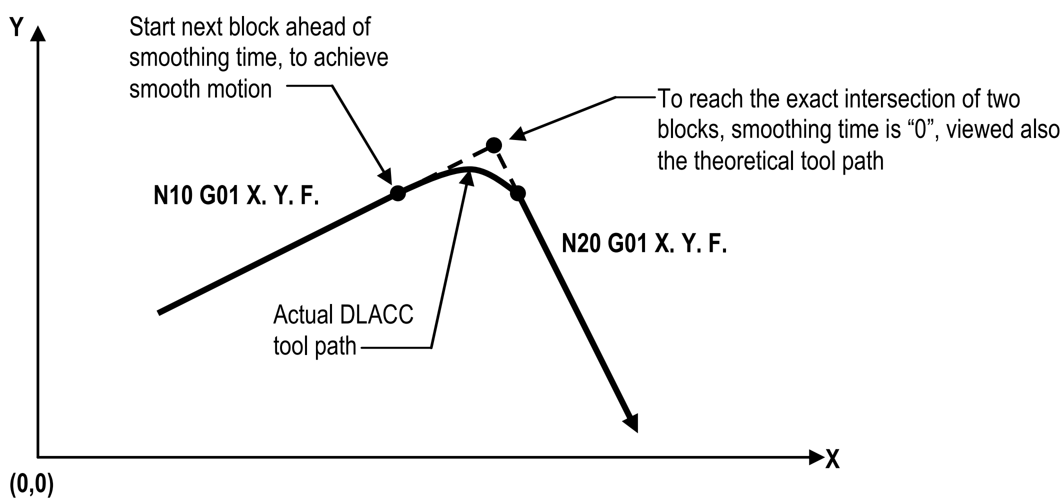


Figure 17-6: Illustration of Smoothing Time in DLACC Execution, After the Look Ahead Smoothing Filter is Engaged

17.1.5 Computer Configuration Requirements

Due to the significant amount of computation required by the DLACC, we suggest the following computer configuration for DLACC:

- CPU: Intel Pentium IV 2 GHz or faster
- Memory: 1 GB or more

17.2 DLACC Parameters

⚠ CAUTION

If your servos have built-in velocity feedforward or torque feedforward functions, they must be disabled before using DLACC. DLACC generates precise commands with frequent acceleration and deceleration, which may result in overshooting or other undesirable motion when combined with built-in velocity feedforward or torque feedforward functions.

17.2.1 Maximum Acceleration/Deceleration

Description

The maximum acceleration/deceleration rate allowed for each axis. This value is used as a "trigger" value – when exceeded, the DLACC algorithm works backwards from the cyclic block that exceeded the maximum acceleration/ deceleration rate, and readjusts the feedrates in previous cyclic blocks until the maximum acceleration/deceleration rate is no longer exceeded.

Measured in Units of: meters/sec^2 or feet/sec^2 or kdeg/sec^2

Range of Valid Values: 0.0001 – 99,999.9999 meters/sec^2 or 0.0001 – 99,999.9999 feet/sec^2 or 0.0001 – 99,999.9999 kdeg/sec^2

Default Value: 1.0 m/s^2 or 3.280840 feet/sec^2 or 1.0 kdeg/s^2

Notes

- This parameter is mostly based on machine characteristics. It is used to determine if the cyclic block is acceptable in terms of acceleration/deceleration, or to calculate the optimal vector speed along the programmed path in each cycle.

- This parameter is also known as the "Maximum Acceleration/Deceleration Rate."

Warning

The Corner Deceleration (see *Section 15.1.3: Corner Deceleration*), Corner Tolerance Compensation (see *Section 15.1.6: Corner Tolerance Compensation Enable*) and Velocity Control in Circular Interpolation (see *Section 15.2.3: Velocity Control in Circular Interpolation Enable*) functions overlap with the DLACC function. If you are using the DLACC function, Corner Deceleration, Corner Tolerance Compensation and Velocity Control in Circular Interpolation must be set to "Disabled." Otherwise, the functions will interfere and result in an undesirable path.

17.2.2 Look Ahead Acc/Dec Time

Description

The smoothing filter time constant for Dynamic Look Ahead Contour Control. It is used by the smoothing filter function to determine how an axis changes velocity over time, from one commanded velocity to another commanded velocity. See *Section 17.3: DLACC Smoothing Modes* for illustrations of how the Look Ahead Acc/Dec Time is used in each smoothing mode type.

The Look Ahead Acc/Dec Time is used to balance between accuracy and smoothness/cutting speed. As this value increases, the motion profile becomes smoother, the maximum cutting speed increases, but the accuracy decreases. As this value decreases, the accuracy increases, but the motion profile becomes less smooth, and the maximum cutting speed decreases.

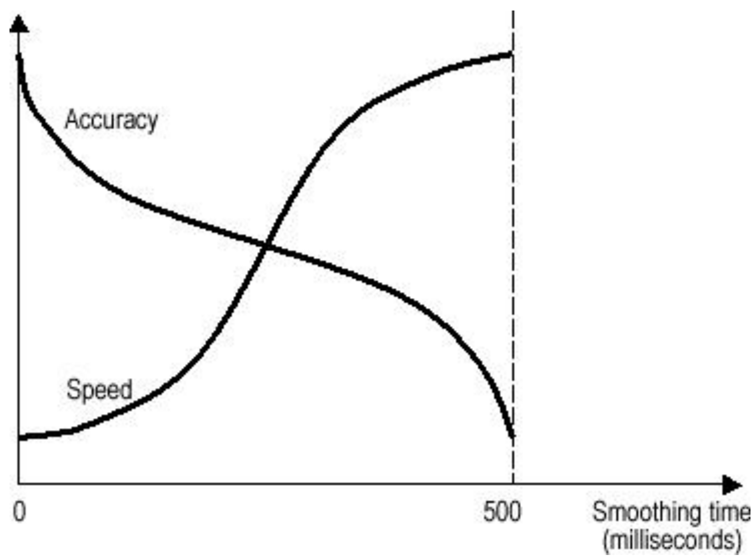


Figure 17-7: Balance Between Accuracy and Speed With Respect to Smoothing Time

Measured in Units of: ms (in integers)

Range of Valid Values: 0 – Look Ahead Smoothing Buffer Size

Default Value: 10 ms

Notes

The Look Ahead Acc/Dec Time is used to determine the ideal tool path.

"0" means no Look Ahead Acc/Dec Time – in this case**,** the targeted tool path is the same as programmed tool path.

A smaller value usually results in a smaller (more accurate) corner, but also more deceleration coming into a corner and more acceleration coming out of a corner. DLACC will adjust affected cyclic blocks to re-distribute feedrate accordingly, so the maximum acceleration/deceleration isn't exceeded.

A larger value usually results in smoother motion, but some inaccuracy of the tool path corner (a larger corner).

In some applications, this parameter may be called "Look Ahead Acc/Dec Time."

17.2.3 Look Ahead Smoothing Limit

Description

This parameter is used when the Look Ahead Smoothing Type is set to “Advanced.” In this case, the smoothing mode is assumed to be “jerk control,” regardless of the actual setting of the Look Ahead Smoothing Mode. This parameter is used to balance block rollover and corner error, by checking to see if the variation of acceleration/deceleration is too big in any one cycle (which would indicate jerky motion), and adjusting the acceleration/deceleration in each cycle if that is the case.

Measured in Units of: millimeters

Valid Values: 0.00001 – 99,999.99999 mm

Default Value: 0.01 mm

Notes

This parameter is only used when the Look Ahead Smoothing Type is set to “Advanced.” If the Look Ahead Smoothing Type is set to “Original,” then this parameter is not used by the DLACC algorithm.

This parameter is for protection in the case of high speeds and longer cycle times. If the block rollover value is smaller than this value, it will be rolled over into the next cyclic block (and combined with that other cyclic block); otherwise, it will be left as one cyclic block by itself.

17.2.4 Look Ahead Smoothing Factor

Description

This value determines the jerk control motion profile of acceleration/deceleration. See Figure 17-13 for an illustration of how this value is used.

Measured in Units of: Integers

Valid Values: 0.0 - 1.0

Default Value: 0.5

Note

This value is only used in jerk control smoothing mode. If the Look Ahead Smoothing Mode is set to “linear,” “exponential” or “bell-shaped,” this parameter will not be used by the DLACC algorithm.

17.2.5 Look Ahead Smoothing End Check Limit (MU)

Description

This limit is a distance that is used to determine when the Smoothing End Check Signal is given.

The Smoothing End Check Signal is used by the DLACC algorithm to keep track of when smoothing is essentially completed, so that M codes or nonmodal G codes start to be executed when and only if the Smoothing End Check Signal indicates that smoothing is completed enough. The movement of the Look Ahead Smoothing End Check Limit distance is still completed, it's just completed AFTER the Smoothing End Check Signal is given, and may be completed concurrently with the execution of an M code or a nonmodal G codes.

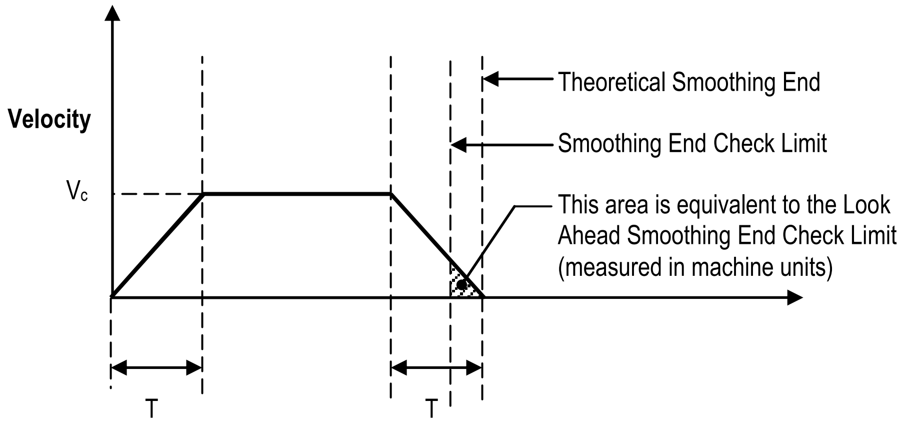


Figure 17-8: Example of Smoothing End Check Limit for Linear Smoothing

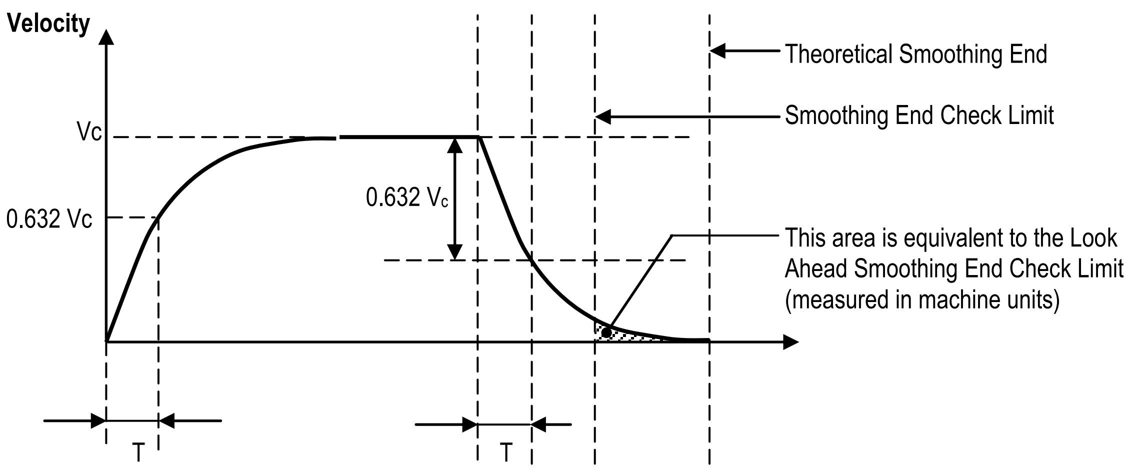


Figure 17-9: Example of Smoothing End Check Limit for Exponential Smoothing

Measured in Units of: Machine Units [see *Section 3.2.2: Machine Unit (Minimum Resolution)*]

Default Value: 500 machine units

Valid Values: 1 – 2000 machine units

Notes

Smoothing filters cause motion to take longer to be completed than it would without a smoothing filter; and in general, the motion in the smoothing filter needs to be completed before M codes and nonmodal G code commands are executed. However, sometimes – particularly with exponential smoothing filters – smoothing takes too long to wait for it to be completed in its entirety before executing subsequent M code or nonmodal G code commands. (See *Section 17.3.4: Exponential Smoothing Mode* for a graphic that demonstrates this.)

This value determines when it's okay to give the Smoothing End Check Signal, even before the entire motion in the smoothing filter is completed. [NOTE: Even after the Smoothing End Check Signal is given, any motion left in the smoothing filter is still completed, even while an M code or nonmodal G code is being executed.]

The smaller the value of this limit, the longer the waiting time for the Smoothing End Check Signal to be given, especially for exponential smoothing filters.

The Smoothing End Check Signal is only really useful when beginning the execution of M codes or nonmodal G codes. If continuous modal or coordinate G codes are given, DLACC uses block rollover and the Smoothing End Check Signal is not checked, so the Look Ahead Smoothing End Check Limit doesn't come into play.

17.2.6 Look Ahead Smoothing Buffer Size

Description

This parameter is used to define the total number of cyclic blocks (small section in each communication cycle) to be pre-processed in DLACC (i.e. the size of the cyclic block buffer).

Valid Values: 200 – 2000

Default Value: 1000

Notes

More cyclic blocks in the cyclic block buffer will provide longer pre-processing and give more room to adjust feedrates if necessary; however, it takes more CPU resources.

Here are some examples of how different values for different interpolation rates result in different preprocessing times:

Interpolation Rate (Cycle Time)	Look Ahead Smoothing Buffer Size	DLACC Preprocessing Time
1 ms	200	200 ms (0.2 seconds)
1 ms	2,000	2,000 ms (2 seconds)
2 ms	200	400 ms (0.4 seconds)
2 ms	2,000	4,000 ms (4 seconds)
4 ms	200	800 ms (0.8 seconds)
4 ms	2,000	8,000 ms (8 seconds)

Table 17-1: Examples of Look Ahead Smoothing Buffer Size and DLACC Preprocessing Time

17.2.7 Look Ahead Smoothing Type

Description

This parameter allows one more level of complexity in the DLACC function. Instead of just optimizing feedrates and trajectories to avoid exceeding the Maximum Acceleration/Deceleration parameter, the DLACC algorithm takes into account the rate of change of acceleration/deceleration, and uses a complex algorithm to optimize that rate of change of acceleration/deceleration. The DLACC function will adjust cyclic blocks in the cyclic block buffer to avoid big changes in acceleration/deceleration.

Valid Values: Original, Advanced

Default Value: Original

Notes

The “Original” mode uses the maximum acceleration/deceleration rate to check all corners to get the optimal feedrate, and re-distributes (interpolate) among motion axes. The DLACC algorithm uses the Look Ahead Smoothing Mode, whatever you have set that to be.

The “Advanced” mode assumes that the smoothing mode is “jerk control,” regardless of the actual setting of the Look Ahead Smoothing Mode parameter (see the following section), and checks to see if the variation of acceleration/deceleration is too big (jerk motion) in any cyclic block. If this is the case, the DLACC algorithm will adjust the acceleration/deceleration in each cycle dynamically.

In general, the “Advanced” mode requires more processing time of the CPU than the “Original” mode, and will add some cutting time, but it allows a higher setting for the Maximum Acceleration/Deceleration.

17.2.8 Look Ahead Smoothing Mode

Description

Valid values: None, Linear, Exponential, Bell-Shape, Jerk-Control (Extended Bell-Shape)

Default value: Exponential

Notes

This parameter is used together with the Look Ahead Acc/Dec Time. Different modes will result in different velocity profiles at corners, if they are within the allowable Maximum Acceleration/Deceleration (explained in *Section 17.2.1: Maximum Acceleration/Deceleration*.)

See *Section 17.3: DLACC Smoothing Modes* for illustrations of each smoothing mode type.

The “None” value indicates no smoothing, and so provides the best possible accuracy, but at the cost of high feedrates.

17.3 DLACC Smoothing Modes

17.3.1 Overview of DLACC Smoothing Modes

Acceleration/deceleration filters are shown in the following figures, which use the following nomenclature:

T: Look Ahead Acc/Dec Time
V_c: Command Velocity

When smoothing = "none," no acceleration or deceleration takes place on the velocity command. [This is theoretical; obviously an axis can't go from 0 to some velocity without some acceleration, but there is no control over how that acceleration occurs.]

17.3.2 Linear Smoothing Mode

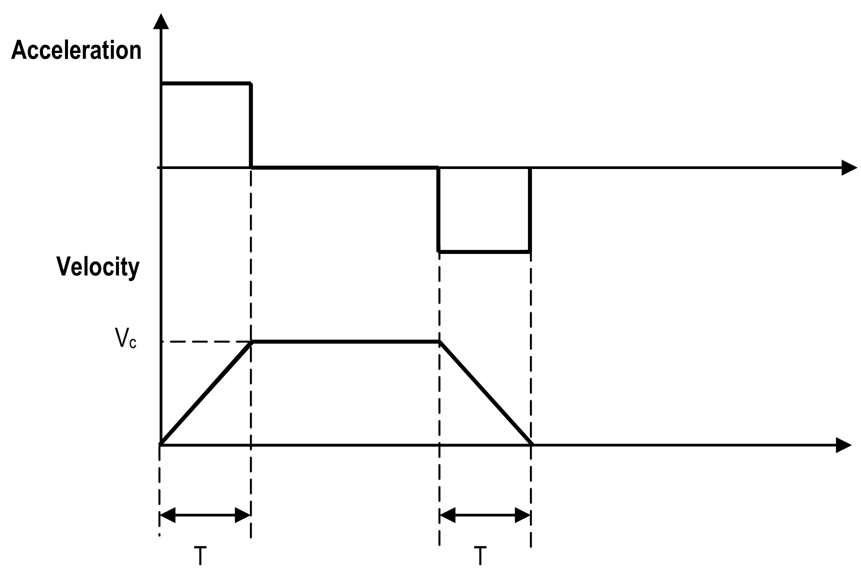


Figure 17-10: Linear Smoothing in DLACC

17.3.3 Bell-Shaped Smoothing Mode

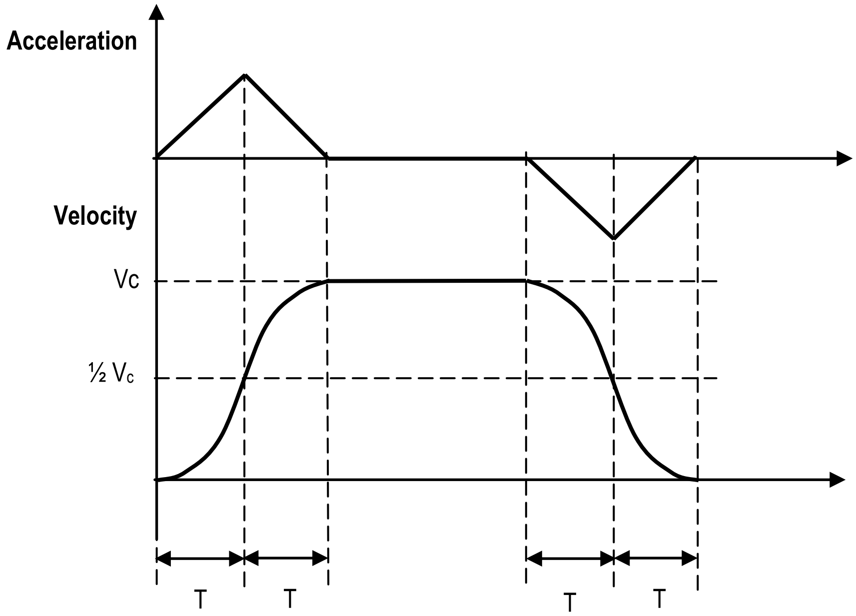


Figure 17-11: Bell-Shaped Smoothing in DLACC

17.3.4 Exponential Smoothing Mode

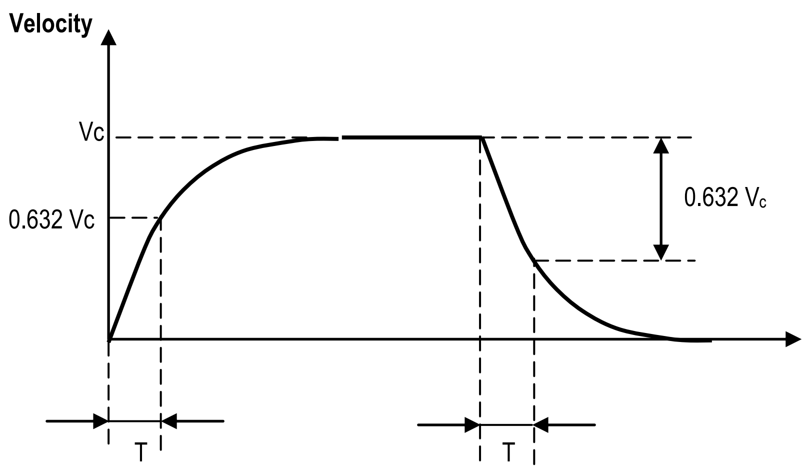


Figure 17-12: Exponential Smoothing in DLACC

17.3.5 Jerk Control Smoothing Mode

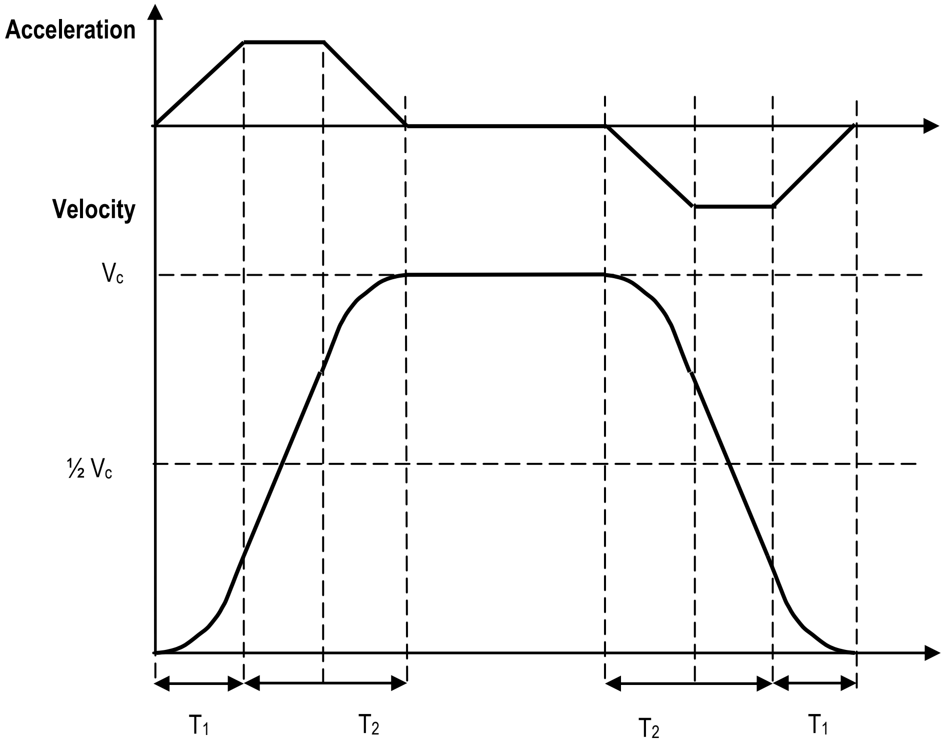


Figure 17-13: Jerk Control (Extended Bell-Shaped) Smoothing Where $T_1 < T_2$

T: Look Ahead Acc/Dec Time (see *Section 17.2.2: Look Ahead Acc/Dec Time*)
V_c: Command Velocity
T_1 = (T)(Look Ahead Smoothing Factor)
T_2 = (T)(2.0- Look Ahead Smoothing Factor)

See *Section 17.2.8: Look Ahead Smoothing Factor* for more information regarding the Look Ahead Smoothing Factor.

If the Look Ahead Smoothing Factor is 0, T_2=2T, and acceleration is a constant, so it is same as linear smoothing with 2T as the Look Ahead Acc/Dec Time. (See Figure 17-10.)

If the smoothing factor is 1, T_1 = T2 =T, so it is same as bell-shaped smoothing. (See Figure 17-11.)

17.4 DLACC Example

There are only four possibilities for the effective usage of G05/G08:

G05 P10000 to turn ON Dynamic Look-Ahead Contour Control
G05 P0 to turn OFF Dynamic Look-Ahead Contour Control

G08 P1 to turn ON Dynamic Look-Ahead Contour Control
G08 P0 to turn OFF Dynamic Look-Ahead Contour Control
(G05 P10000 is equivalent to G08 P1, and G05 P0 is equivalent to G08 P0)

Here is a simple example to demonstrate DLACC usage:

```
G90G00X0Y0 (X and Y axes return to zero)
G08 P1 (DLACC ON)
G91G01F20000.0 (Incremental linear feed)
X50.0Y100.0
X50.0Y-100.0
X50.0Y100.0
X50.0Y-100.0
G08 P0 (DLACC OFF)
G04 X1.0 (Dwell)
G90G00X0.0Y0.0 (X and Y axes return to zero)
M02 (Program end)
```

The program starts the motion from X-Y plane program zero point, turns on DLACC, performs “saw teeth” shape movements, then turns off DLACC, and finally returns to program zero point after a dwell.

17.5 DLACC Limitations

Since acceleration and deceleration is performed before interpolation, and the motion profile is pre-processed by DLACC, if Cycle Stop is commanded during the DLACC movement, the machine will stop without deceleration. In other words, the machine will come to a jerky stop. Therefore, it is highly recommended that cycle stop not be used during DLACC movement. It is also recommended that the following commands not be used when DLACC is on: Single Block, Optional Stop, Optional Skip, and Dry Run.

17.6 DLACC Supported Programming Codes

17.6.1 Supported M and T Codes

All M and T codes are supported, and can be commanded when DLACC is on. This includes M98/M99 subprogram calls.

17.6.2 Supported G Codes

The following G codes can be commanded when DLACC is on:

G00	Rapid traverse
G01	Linear interpolation
G02, G03	CW/CCW circular or helical interpolation
G04	Dwell
G17-G19	XY/ZX/YZ plane selection
G20, G21	Inch/metric data input
G28, G29	Automatic return to/from reference point
G30	Automatic return to 2^{nd}, 3^{rd}, 4^{th} reference points
G50, G51	Scaling off/on
G50.1, G51.1	Mirror image off/on
G52	Local coordinate system selection
G53	Machine coordinate system selection
G54-G59	Workpiece coordinate system 1-6 selection
G54.1	Additional workpiece coordinate system selection
G61	Exact stop check mode
G64	Cutting mode
G65	Simple macro call
G68, G69	Coordinate system rotation on/off
G90	Absolute command programming
G91	Incremental command programming
G92	Workpiece coordinate programming
G310, G311	Linear interpolation feedrate include/exclude rotary axes

⚠ CAUTION

NO G CODES OTHER THAN THOSE LISTED ABOVE SHOULD BE COMMANDED WITH DLACC. If an incompatible G code (a G code not listed above) is used with DLACC, the motion program will stop at the first instance of an incompatible G code, and you will see an error message.

Chapter 18: Quadrant Protrusion Compensation

NOTE: This function is only available in MC-Quad.

18.1 Overview

18.1.1 Quadrant Protrusion Compensation Description

Quadrant Protrusion Compensation is a function designed to avoid protrusions that can appear when cutting circles, caused by lost motion that can occur due to backlash or friction when a motor reverses direction (when the path crosses from one quadrant to another).

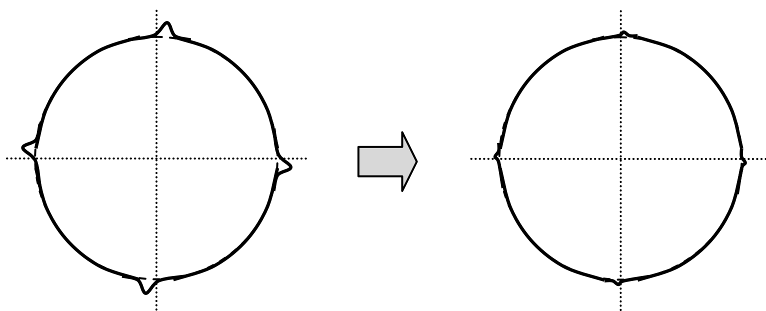


Figure 18-1: Quadrant Protrusion Compensation

The principle of Quadrant Protrusion Compensation is to apply a velocity offset in the shape of a square wave impulse in the direction opposite to the protrusions to make the cut path closer to the ideal path. For example, in Figure 18-2, Quadrant Protrusion Compensation is used while cutting the bottom of a circle in the counterclockwise direction.

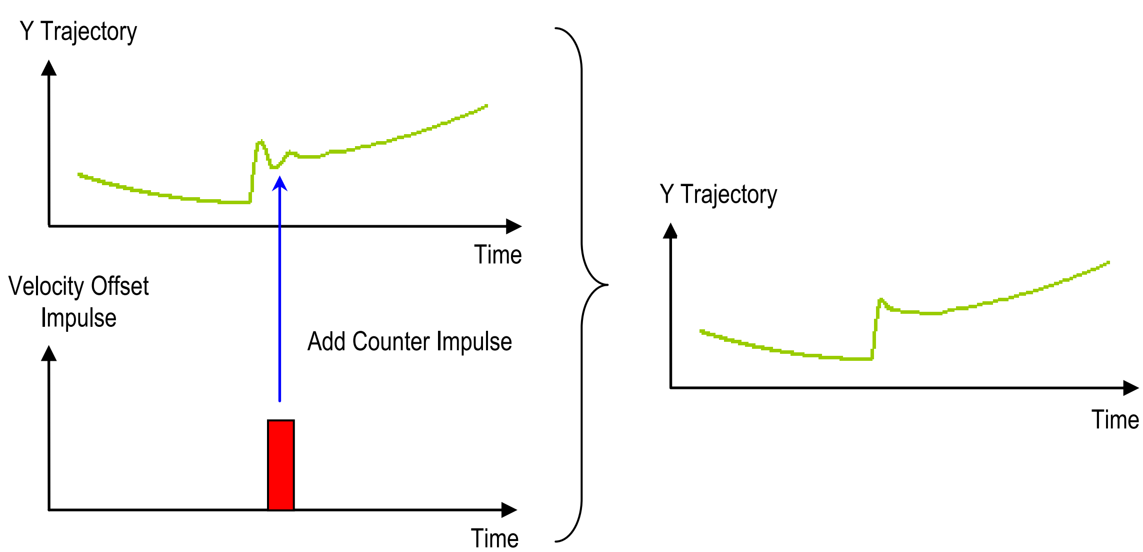


Figure 18-2: Principle of Quadrant Protrusion Compensation

18.2 To Set the Pattern of Velocity Offset Impulse

18.2.1 Idea of Velocity Offset Impulse

Parameters that describe the impulse are height, width, and timing.

- The height of the impulse represents the velocity offset, and increases with the feedrate. You will need to set the parameters for minimum velocity, maximum velocity, and a profile factor that affects the feedrate vs. impulse height curve. (See Figure 18-3).
- The width of the impulse represents the duration of the impulse, and is determined by parameters for impulse duration.
- The timing at which the impulse is applied (the amount of time between when the motor reverses direction and when the impulse is applied) is determined by parameters for impulse delay.

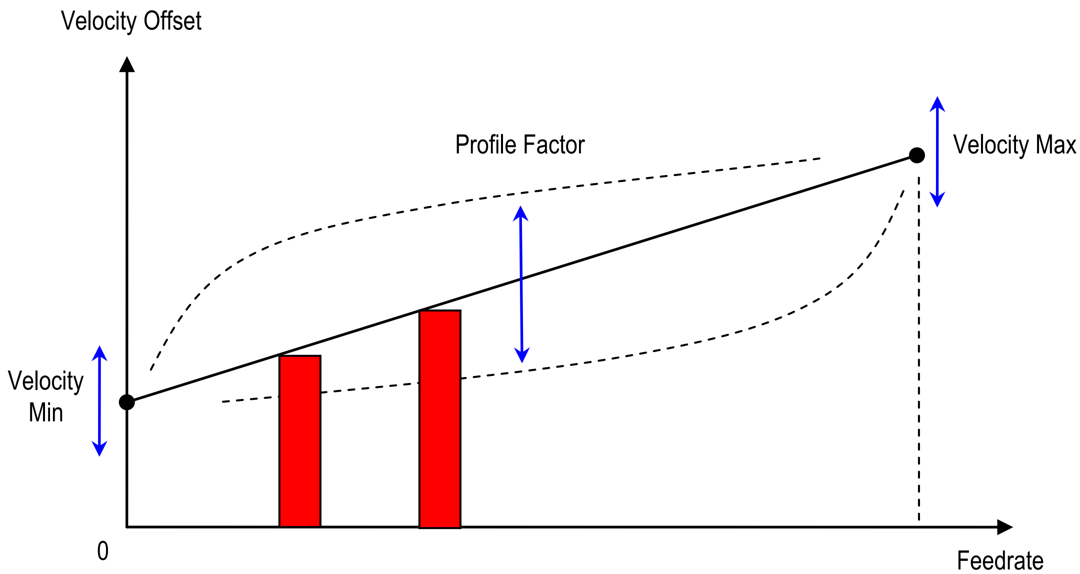


Figure 18-3: Setting the Velocity Offset Impulse Parameters

18.3 Quadrant Protrusion Compensation Parameters

18.3.1 Quadrant Protrusion Compensation

Description

This value is set to “1” (checked) to enable, or is set to “0” (unchecked) to disable Quadrant Protrusion Compensation for each of X, Y, Z, and A axis.

Valid Values: 0, 1

Meaning of Values

0 – Disable Quadrant Protrusion Compensation for target axis.
1 – Enable Quadrant Protrusion Compensation for target axis.

Default Value: 0

18.3.2 Handwheel Interposition

Description

This value is set to “1” (checked) to enable, or is set to “0” (unchecked) to disable Quadrant Protrusion Compensation for each of X, Y, Z, and A axis when Handwheel (manual pulse) mode is selected.

Valid Values: 0, 1

Meaning of Values

0 – Disable Quadrant Protrusion Compensation for target axis during Handwheel mode.
1 – Enable Quadrant Protrusion Compensation for target axis during Handwheel mode.

Default Value: 0

18.3.3 When the program is stopped

Description

This value is set to “1” (checked) to enable, or is set to “0” (unchecked) to disable Quadrant Protrusion Compensation for X, Y, Z, and A axis after the part program ends or is stopped.

Valid Values: 0, 1

Meaning of Values

0 – Disable Quadrant Protrusion Compensation for target axis when the part program ends or is stopped.
1 – Enable Quadrant Protrusion Compensation for target axis when the part program ends or is stopped.

Default Value: 0

18.3.4 Plus Direction Velocity Min

Description

This value is the minimum velocity offset applied by the velocity offset impulse when the motor changes its turning direction from the axis' minus direction to the axis' plus direction.

Measured in Units of: mm/sec

Range of Valid Values: 0 – 10,000 mm/ sec

Default Value: 10 mm/sec

NOTE: Plus value applies when the axis velocity turns from minus to plus.

18.3.5 Minus Direction Velocity Min

Description

This value is the minimum velocity offset applied by the velocity offset impulse when the motor changes its turning direction from the axis' plus direction to the axis' minus direction.

Measured in Units of: mm/sec

Range of Valid Values: 0 – 10,000 mm/ sec

Default Value: 10 mm/sec

NOTE: Minus value applies when the axis velocity turns from plus to minus.

18.3.6 Plus Direction Velocity Max

Description

This value is the maximum velocity offset applied by the velocity offset impulse when the motor changes its turning direction from the axis' minus direction to the axis' plus direction.

Measured in Units of: mm/sec

Range of Valid Values: 0 – 100,000 mm/ sec

Default Value: 100 mm/sec

NOTE: Plus value applies when the axis velocity turns from minus to plus.

18.3.7 Minus Direction Velocity Max

Description

This value is the maximum velocity offset applied by the velocity offset impulse when the motor changes its turning direction from the axis' plus direction to the axis' minus direction.

Measured in Units of: mm/sec

Range of Valid Values: 0 – 100,000 mm/ sec

Default Value: 100 mm/sec

NOTE: Minus value applies when the axis velocity turns from plus to minus.

18.3.8 Plus Direction Profile Factor

Description

This value determines the curvature of the feedrate vs. velocity offset curve, as shown in Table 18-1 and Figure 18-4. A positive value will cause the curve to bend down and a negative value will cause the curve to bend up, with the absolute value determining how much the curve bends up or down. This value applies to the case when the motor changes its turning direction from the axis' minus direction to the axis' plus direction.

Range of Valid Values: -100 – 100

Default Value: 0

NOTE: Plus value applies when the axis velocity turns from minus to plus.

Profile Factor	Path in Figure 18-4
< 0	Path 1
= 0	Path 2
> 0	Path 3

Table 18-1: Relation of Profile Factor and Path

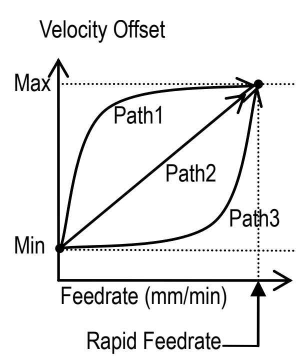


Figure 18-4: Velocity Path

18.3.9 Minus Direction Profile Factor

Description

This value determines the curvature of the feedrate vs. velocity offset curve, as shown in Table 18-1 and Figure 18-4. A positive value will cause the curve to bend down and a negative value will cause the curve to bend up, with the absolute value determining how much the curve bends up or down. This value applies to the case when the motor changes its turning direction from the axis' plus direction to the axis' minus direction.

Range of Valid Values: -100 – 100

Default Value: 0

NOTE: Minus value applies when the axis velocity turns from plus to minus.

18.3.10 Plus Direction Duration

Description

This value is the duration of the velocity offset impulse. This value applies to the case when the motor changes its turning direction from the axis' minus direction to the axis' plus direction.

Measured in Units of: ms

Range of Valid Values: 1 – 100 ms

Default Value: 5 ms

NOTE: Plus value applies when the axis velocity turns from minus to plus.

18.3.11 Minus Direction Duration

Description

This value is the duration of the velocity offset impulse. This value applies to the case when the motor changes its turning direction from the axis' plus direction to the axis' minus direction.

Measured in Units of: ms

Range of Valid Values: 1 – 100 ms

Default Value: 5 ms

NOTE: Minus value applies when the axis velocity turns from plus to minus.

18.3.12 Plus Direction Delay

Description

This value is the delay between when the motor changes direction and when the velocity offset impulse is applied. This value applies to the case when the motor changes its turning direction from the axis' minus direction to the axis' plus direction.

Measured in Units of: ms

Range of Valid Values: 0 – 1000 ms

Default Value: 0 ms

NOTE: Plus value applies when the axis velocity turns from minus to plus.

18.3.13 Minus Direction Delay

Description

This value is the delay between when the motor changes direction and when the velocity offset impulse is applied. This value applies to the case when the motor changes its turning direction from the axis' plus direction to the axis' minus direction.

Measured in Units of: ms

Range of Valid Values: 0 – 1000 ms

Default Value: 0 ms

NOTE: Minus value applies when the axis velocity turns from plus to minus.

18.3.14 Plus Direction Polarity

Description

When this value is set to "1," the direction of the velocity offset is in the normal direction (outwards from the circle). When this value is set to "-1," the direction of the velocity offset is in the reverse direction (inwards from the circle). This value applies to the case when the motor changes its turning direction from the axis' minus direction to the axis' plus direction.

Valid Values: 1, -1

Meaning of Values:

1 – Velocity offset in outward direction.
-1 – Velocity offset in inward direction.

Default Value: 1

NOTE: Plus value applies when the axis velocity turns from minus to plus.

18.3.15 Minus Direction Polarity

Description

When this value is set to "1," the direction of the velocity offset is in the normal direction (outwards from the circle). When this value is set to "-1," the direction of the velocity offset is in the reverse direction (inwards from the circle). This value applies to the case when the motor changes its turning direction from the axis' plus direction to the axis' minus direction.

Valid Values: 1, -1

Meaning of Values:

1 – Velocity offset in outward direction.
-1 – Velocity offset in inward direction.

Default Value: 1

NOTE: Minus value applies when the axis velocity turns from plus to minus.

Chapter 19: Stored Straight Error Compensation

19.1 Overview

19.1.1 What is Stored Straight Error Compensation

Stored Straight Error Compensation is a function that adds offsets to compensate for errors in the straightness of each axis, such as those that result from axes that are not perfectly perpendicular to each other. Straightness errors can be more pronounced in machines with longer strokes.

Stored Straight Error Compensation works between a moving axis and a compensating axis. The moving axis contains offset points, which are points at which offset values for the compensating axis can be specified. For each offset point, the corresponding offset value is added to the compensating axis when the moving axis is at that offset point. In between offset points, an offset obtained by interpolating between the offset points is added to the compensating axis (see Section 19.1.2). Offset points are specified using pitch error points (see *Chapter 12: Pitch Error Compensation Parameters and Usage*), and are affected by pitch origin and pitch interval parameter settings. The units of offset points are the number of pitch intervals and the units of offset values are micrometers.

19.1.2 Calculation of Straight Error Compensation

This section describes how offsets for the compensating axis are calculated from the offset points and offset values. Figure 19-1 shows a sample setup where a, b, c, and d are offset points and A, B, C, and D are the corresponding offset values.

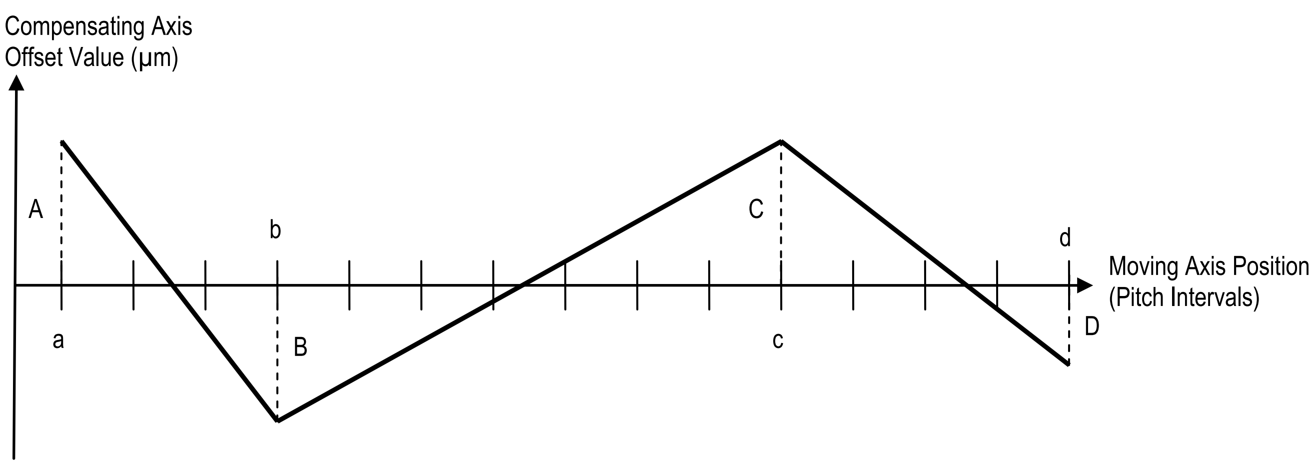


Figure 19-1 Stored Straight Error Compensation Offset Sample

The following function calculates the offset added to the compensating axis at a point x in between offset points b and c:

Offset Value (x) = (C – B) / (c – b) * (x – b) + B

Normally, the first offset point (offset point a in this example) should be the pitch error point at the most negative point of the moving axis and the last offset point (offset point d in this example) should be the pitch error point at the most positive point of the moving axis.

19.2 Stored Straight Error Compensation Parameters

19.2.1 Stored Straight Error Compensation Enable

Description

This value is set to "1" (checked) to enable, or is set to "0" (unchecked) to disable this function. Disabling this function will stop the ServoWorks CNC Engine from loading the "StoredStraightErr.dat" file. You must restart the ServoWorks CNC application for this parameter to take effect.

Valid Values: 0, 1

Meaning of Values

0 – Disable Stored Straight Error Compensation.
1 – Enable Stored Straight Error Compensation.

Default Value: 0

NOTE: This parameter takes effect only after restarting the ServoWorks CNC application.

19.3 Stored Straight Error Compensation DAT File

The ServoWorks CNC Engine will load the offset points and offset values from the file “StoredStraightErr.dat” file during initialization. This file is located in the folder “C:\Program Files\Softservo\S-100M\ini” (for the S-100M; replace “S-100M” in the file path with the ServoWorks CNC application that you are using to find the file in your system). The “StoredStraightErr.dat” file is only read when the ServoWorks CNC application is initializing; thus, you must restart the ServoWorks CNC application after editing the text file for the changes to take effect. An example of this file is shown in Figure 19-2.

```
StoredStraightErr.dat - Notepad
File  Edit  Format  View  Help
AxisMove=1
        2          5        7          9%
AxisComp=2
    -10.00      10.00    -10.00      10.00      0.00      0.00      0.00      0.00
AxisComp=3
      0.00       0.00      0.00       0.00      0.00      0.00      0.00    | 0.00
```

Figure 19-2 StoredStraightErr.dat example

19.2.2 Moving Axis and Points

The first two lines designate the moving axis and the offset points. In the first line, “AxisMove=1” specifies that axis 1 is the moving axis. In the second line, the offset points are specified using pitch error points. For example, in the above “StoredStraightErr.dat” file, the offset points are set to pitch error points 2, 5, 7, and 9. The actual positions of the offset points on the moving axis can be calculated using:

Offset position = Pitch Origin + Pitch Interval * Value

For example, if the “Pitch Origin” is 0 and the “Pitch Interval” is 10mm, the offset points will be located at 20mm, 50mm, 70mm, and 90mm on the moving axis. For additional information on how to set the pitch error compensation parameters, please refer to “*Chapter 12: Pitch Error Compensation*.”

You can set a maximum of 8 offset points. Use the “%” symbol to terminate the sequence of offset points if you have less than 8 offset points.

19.2.3 Compensating Axis and Values

The remaining lines are used to set up the offset values of the compensating axes. The lines “AxisComp=2” and “AxisComp=3” signify that axis 2 and axis 3 are compensating axes. The sequence of numbers after each compensating axis specifies the offset values. The units for each entry are in micrometers. For example, the offset values for axis 2 (when the axis 1, the moving axis, is at positions 20mm, 50mm, 70mm, and 90mm) are -10µm, 10µm, -10µm, and 10µm.

NOTE: The units of offset values are in micrometers.

NOTE: Stored Straight Error Compensation is enabled only after a homing operation.

Chapter 20: Switch Buttons

NOTE: This function is only available in the S-100M series of ServoWorks CNC applications.

20.1 Overview

This function will change six of the buttons displayed on the right side of the S-100M series GUI to report the status of I/O bits. Each button will contain a LED icon that shows whether the I/O bit assigned to the button is currently on (green) or off (red). Moreover, when a button is assigned an output bit, the output bit can be toggled by clicking on the button. To enable this button, click on the CHG BUTTON icon on the main screen of the S-100M series GUI.

20.2 Switch Buttons Parameters

20.2.1 I/O Type

Description

This value is set to “Disable,” “Input,” or “Output” to determine whether the button is disabled, enabled with an input bit assigned to it, or enabled with an output bit assigned to it.

Valid Values: Disable, Input, and Output

Meaning of Values

Disable – Disable the button, no I/O bit is assigned.
Input – Enable the button with an input bit assigned. The input bit address and a label for the button can be set using the “Label” and “I/O Address” settings.
Output – Enable the button with an output bit assigned. The output bit address and a label for the button can be set using the “Label” and “I/O Address” settings.

Default Value: Disable

20.2.2 Label

Description

This value is the caption shown on the button. Setting this blank will cause the GUI to not show any caption on the button, but the next time the application is restarted, the caption will be set to “BUTTON #” (where # is the button number).

Values: Any combination of alphabets and numbers.

Default Value: Button # (# is the button number, 1 to 6)

20.2.3 I/O Address

Description

This value is the bit address of the I/O bit assigned to the button. The I/O bit must fall within the PLC addresses X0.0 to X99.7 for input bits and Y0.0 to Y99.7 for output bits. Invalid addresses will be ignored and a gray LED will be shown on the button. Refer to the “*LadderWorks PLC I/O Mapping for ServoWorks MC-Quad and the ServoWorks S-100M Series*” manual for more information on these PLC addresses.

Range of Value: X0.0 – X99.7, Y0.0 – Y99.7

Default Value: 0.0

Index

E

F

G

H

I

J

L

M

N

T

U

V

W

Z