Chapter 7: Motion Control

Table of Contents

Chapter 7: Motion Control 7–1
Introduction 7–2
7.1 - PR mode description 7–2
7.1.1 - Shared PR parameters 7–4
7.1.2 - Monitoring variables of PR mode 7–6
7.1.3 - Motion Control commands 7–9
7.1.4 - Overview of the PR procedure 7–36
7.1.5 - Trigger methods for the PR command 7–42
7.1.6 - PR procedure execution flow 7–46
7.2 - Application of motion control 7–58
7.2.1 - Data array 7–58
7.2.2 - High-speed position capturing function (Capture) 7–63
7.2.3 - High-speed position comparing function (Compare) 7–69
7.3 - E-Cam 7–75
7.3.1 - Source signal for the master axis 7–76
7.3.2 - Clutch engagement and disengagement 7–80
7.3.3 - E-Cam gears and curve scaling 7–86
7.3.4 - E-Cam curve 7–89
7.3.5 - E-Cam curve and PR command overlapping 7–97
7.3.6 - Troubleshooting for E-Cam 7–99
7.3.7 - Rotary Shear 7–101
7.3.8 - Flying Shear 7–121
7.3.9 - Macro 7–133
7.3.10 - Auxiliary function 7–143

Wiring
Parameters
DI/DO Codes
Monitoring
Alarms

Introduction

This chapter introduces internal motion commands in the SureServo2 drive in PR mode. In this mode, commands are generated based on the internal commands of the servo drive. Various motion commands are available, including homing, speed, position, parameter writing, arithmetic operation, and jump. Other motion control functions such as high-speed position capture (Capture), high-speed position compare (Compare), and E-Cam are also available. This chapter contains detailed description of each command type.

7.1 - PR mode description

In PR mode, the servo drive automatically generates the motion commands. Apart from the basic arithmetic operation commands, the drive saves all parameter settings in the parameter file in the servo drive. Thus changing parameter values simultaneously changes the PR commands. The drive provides 100 path setting sets, which include the homing method, Position command, Speed command, Jump command, Write command, Index Positioning command, and arithmetic operation commands.

Except for arithmetic operations, the properties and corresponding data for each PR path are set by parameters. You can find information for all PR parameters in the descriptions of Group 6 and 7 in Chapter 8. For example, PR#1 path is defined by two parameters: P6.002 and P6.003. P6.002 specifies the properties for PR#1, such as the PR command type, whether to interrupt and whether to auto-execute. The next PR. P6.003 is subject to change based on the properties set in P6.002. If P6.002 is set to a Speed command, then P6.003 specifies the target speed. When P6.002 is set to a Jump command, then P6.003 specifies the target PR. The parameters for the PR#2 path are P6.004 and P6.005 and they work the same way as P6.002 and P6.003. The same is true for the rest of PR paths. See Figure 7-1.

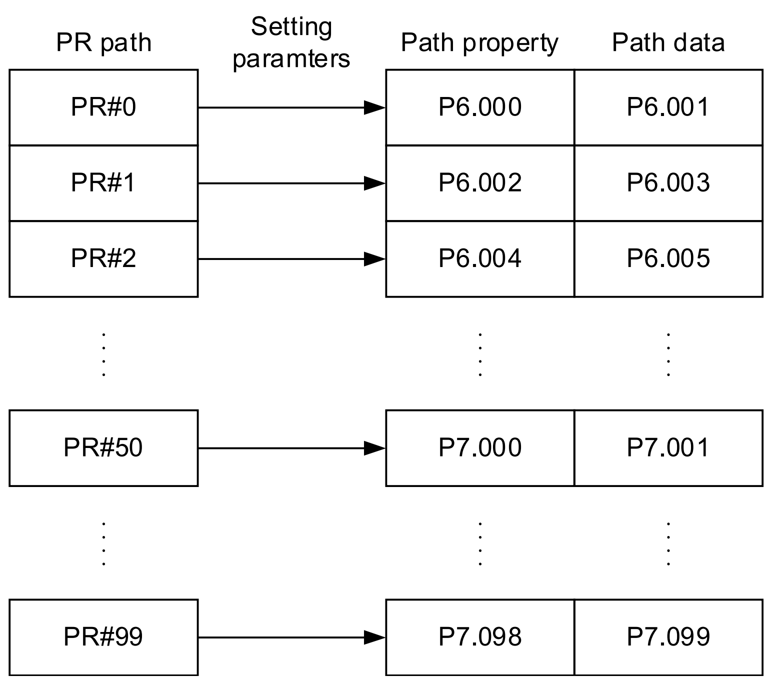


Figure 7-1 Setting parameters for each PR path

In the SureServo2 Pro software, when you select the PR to be edited in PR mode, the corresponding parameters appear at the top of the window. See Figure 7-2. If you select PR#1, P6.002 and P6.003 appear at the top in the editing section (see P6.002 and P6.003 in Table 7-1 for example). The PR property and its data content differ in accordance with the motion command type. The small green box in the top left of the screen contains the current value for P5.007 (command and status of the current PR path). For more information about Motion Control mode, please refer to Section 7.1.3.

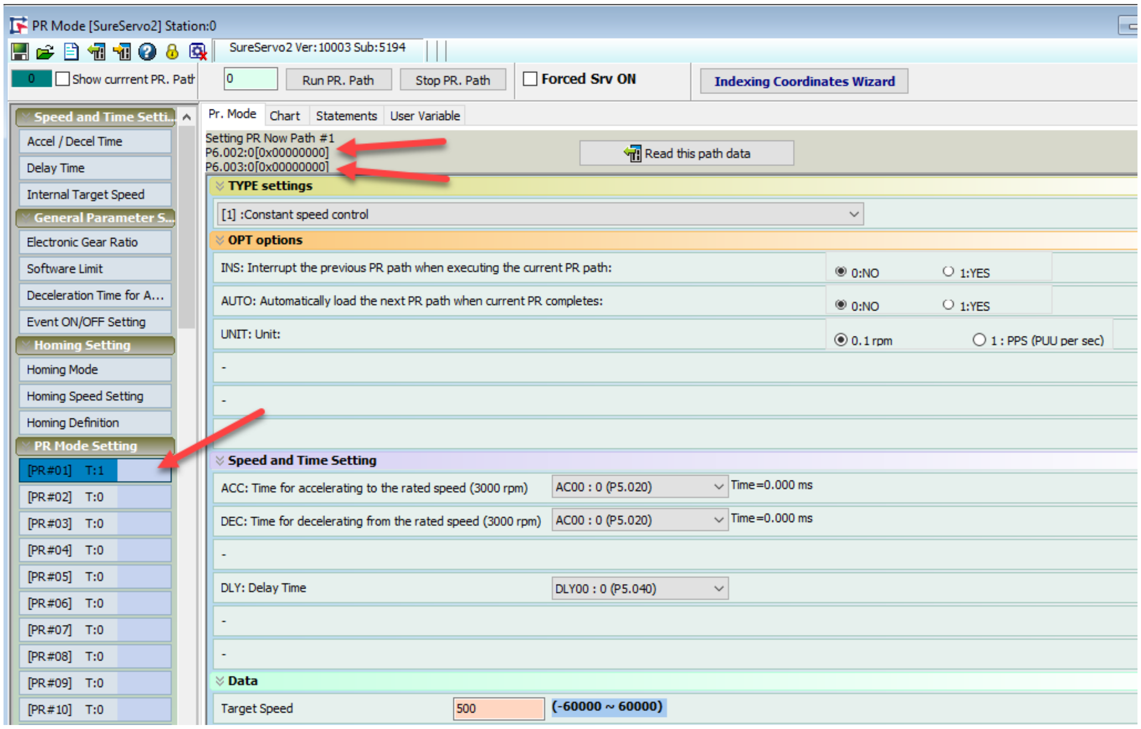


Figure 7-2 PR mode interface in SureServo2 Pro

Table 7-1 Example of PR#1 property and data content

BIT / PR#1	31–28	27–24	23–20	19–16	15–12	11–8	7–4	3–0
P6.002								TYPE
P6.003	Data content (32-bit)							

TYPE: Control command mode

TYPE No.	Command Mode
1	SPEED: speed control
2	SINGLE: positioning control; stop once positioning is completed.
3	AUTO: positioning control; execute the next PR path once positioning is completed.
7	JUMP: jump to the specified path.
8	WRITE: write a constant, parameter value, data array, or monitoring variable to another parameter or data array..
0xA	INDEX: index positioning control
0xB	STATEMENT: statement / arithmetic operations

SureServo2 Pro software provides an editing interface for PR programming (see Figure 7-3). It is easier to set PR paths in SureServo2 Pro, where you can set the options for command triggering, command types and other properties. In the Chart tab you can view the sequence flow in a graphical manner.

Note: You must set the arithmetic operations statements in the software, you cannot enter arithmetic operations or statements using the keypad.

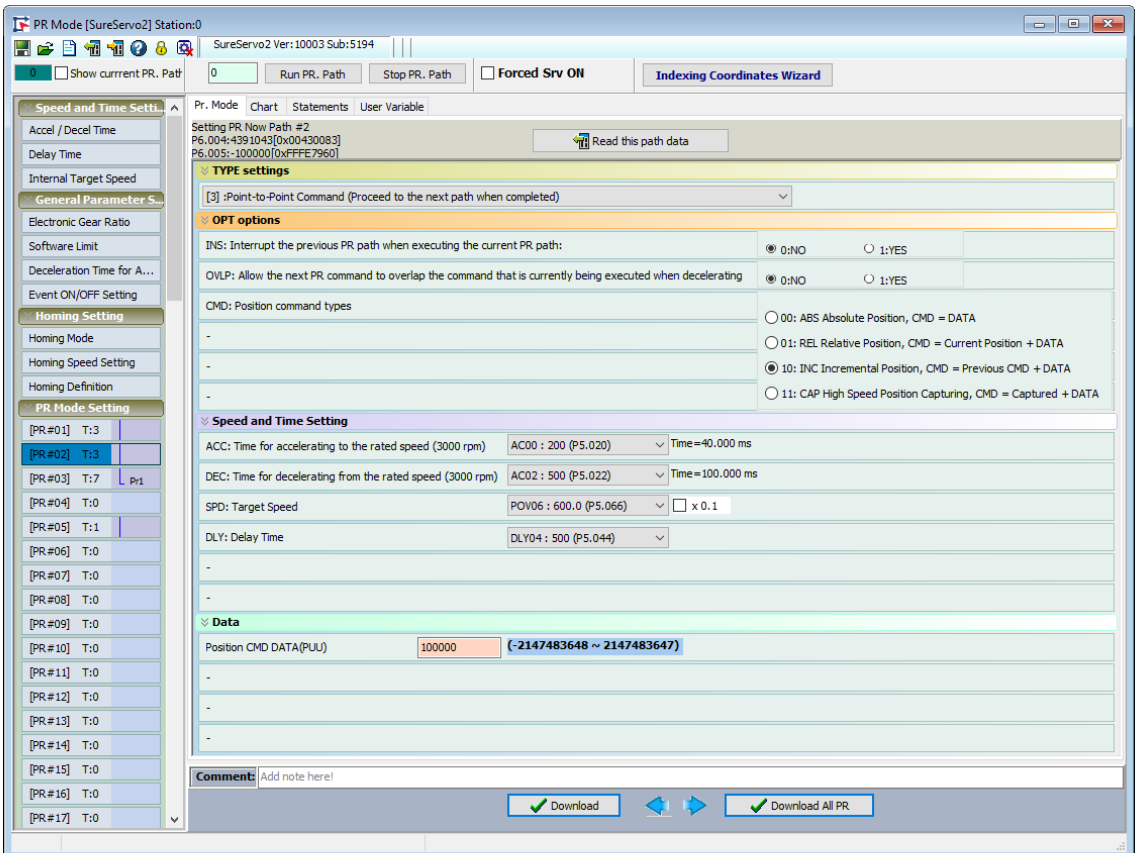


Figure 7-3 PR display in SureServo2 Pro

7.1.1 - Shared PR Parameters

The SureServo2 drive provides 16 acceleration / deceleration time settings (P5.020–P5.035), 16 delay time settings (P5.040–P5.055), and 16 target speed settings (P5.060–P5.075) for you to set the PR paths (as shown in Figure 7-4). The Internal Target Speed or POV (Position Option Velocity) registers are only used for point-to-point and index positioning moves. If you change a parameter that is used by multiple PR paths, then all PR paths using this parameter are changed as well. Please be aware of this when setting PR paths so as to avoid any danger or damage to the machine. For example, if multiple PR commands use the target speed setting from P5.060, when you change the value of P5.060, those PR commands' target speed are also changed.

SureServo2 Pro also provides a user-friendly interface for this shared PR parameter function (see Figure 7-5). In this data window, the acceleration / deceleration time is set based on the length of time for the motor to accelerate from 0 to 3000rpm or to decelerate from 3000rpm to 0. For instance, if acceleration time is set to 50ms then it will take 50ms to reach 3000rpms. If target speed for the motion command is 1500rpm, then the acceleration time to reach 1500rpm is 25ms even though the acceleration time is set to 50ms. The acceleration / deceleration time is a fixed slope, and the slope does not change when you change target speed parameter values.

Wiring
Parameters
DI/DO Codes
Monitoring
Alarms

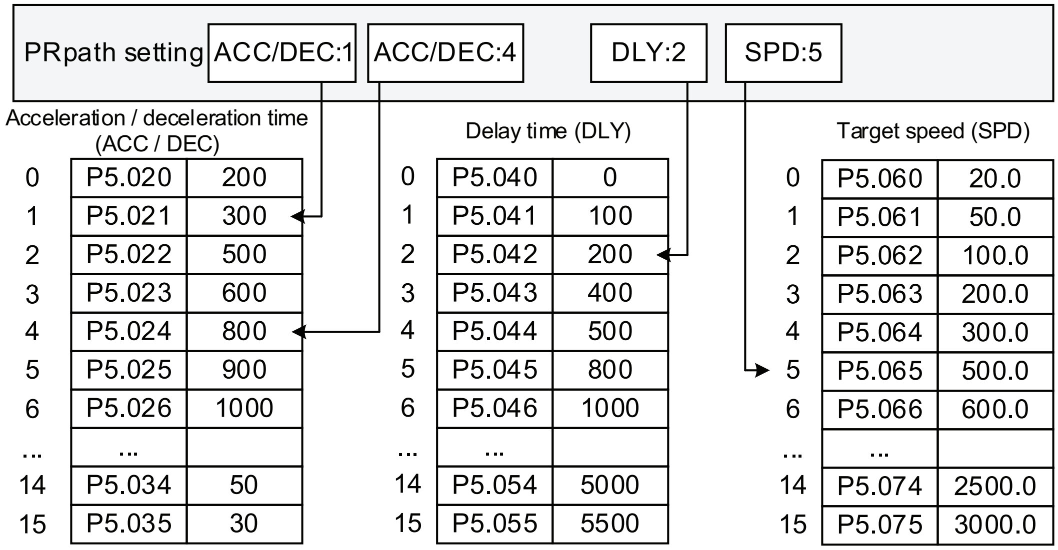


Figure 7-4 Shared parameter data for PR paths

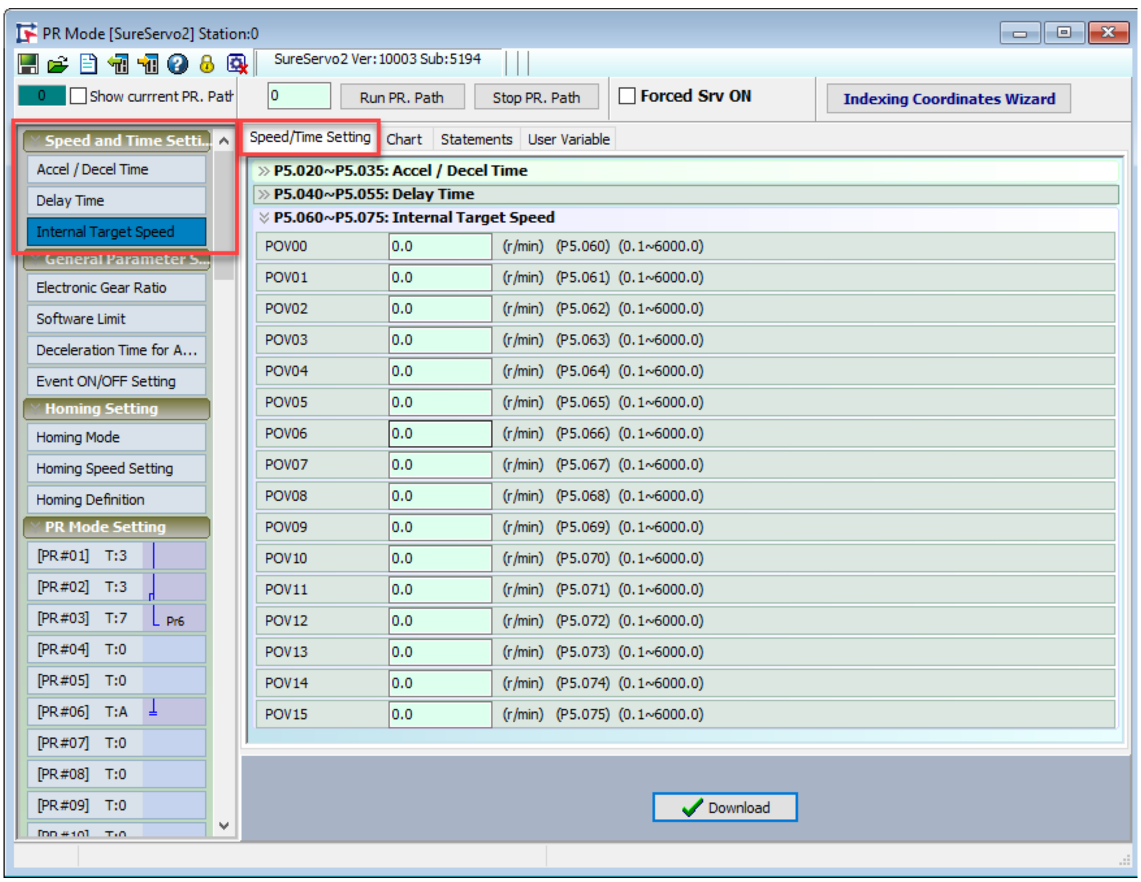


Figure 7-5 SureServo2 Pro interface for shared PR parameter data

7.1.2 - Monitoring variables of PR mode

PR mode provides four monitoring variables for the servo command and feedback: command position (PUU), PR command end register, feedback position (PUU), and position error (PUU). PUU stands for Pulse User Units and is determined by the E-Gear ratio (P1.044 and P1.045). These can be monitored in P0.002 and are described below:

1) PR command end register: monitoring variable code 064 (40h). The target position of the PR command, simplified as **Cmd_E** (Command End). When a command is triggered, the servo drive calculates the target position and then updates the PR command end register.
2) Command position (PUU): monitoring variable code 001. The target position of the motion command generated per scan cycle during servo operation (updated every 1ms), simplified as **Cmd_O** (Command Operation).
3) Position error (PUU): monitoring variable code 002. The deviation between the command position (PUU) and the feedback position (PUU), simplified as **Err_PUU** (Error PUU).
4) Feedback position (PUU): monitoring variable code 000. The feedback position (coordinates) for the motor, simplified as **Fb_PUU** (Feedback PUU).

How these four monitoring variables work is shown in Figure 7-6 After the servo issues a Position command, the servo sets the position of Cmd_E once the target position data is acquired. The motor operates to the target position based on the PR path setting. Cmd_O calculates the amount of command deviation in each fixed cycle and sends it to the servo drive, where it is treated as a dynamic command. Fb_PUU is motor's feedback position and Err_PUU is the deviation of Cmd_O minus Fb_PUU.

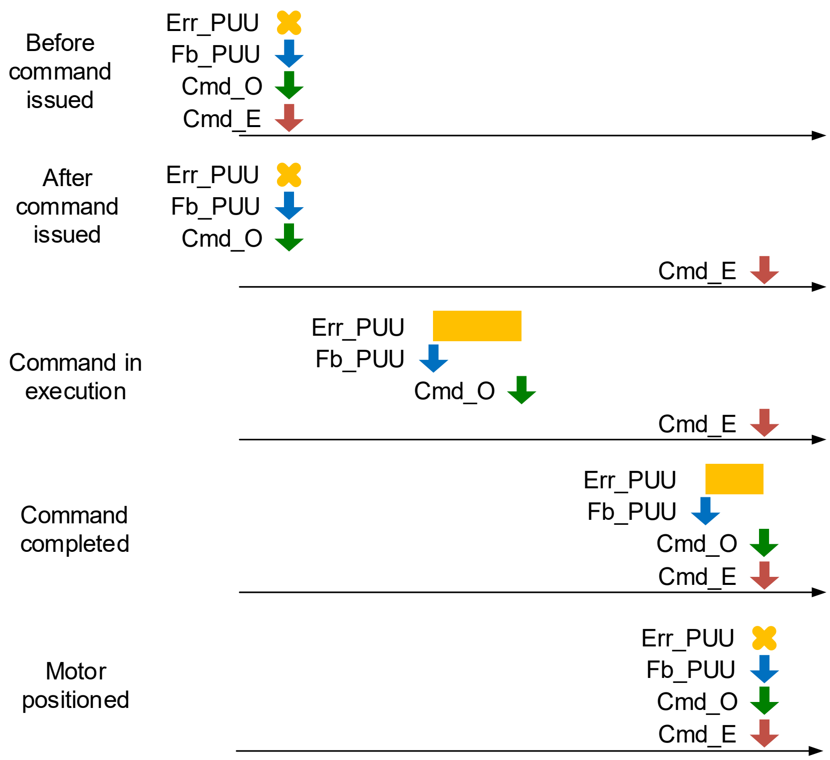


Figure 7-6 Timing diagram for PR mode monitoring variables

The detailed command behavior of each stage is illustrated in Figure 7-7. Cmd_E is the endpoint specified by the command; it is set when the PR path is triggered. Fb_PUU is the feedback position, that is motor's actual position. Divide this motion command into slices and take one of them as an example. Cmd_O is the target of this command section and Err_PUU is the deviation between the target position and the feedback position.

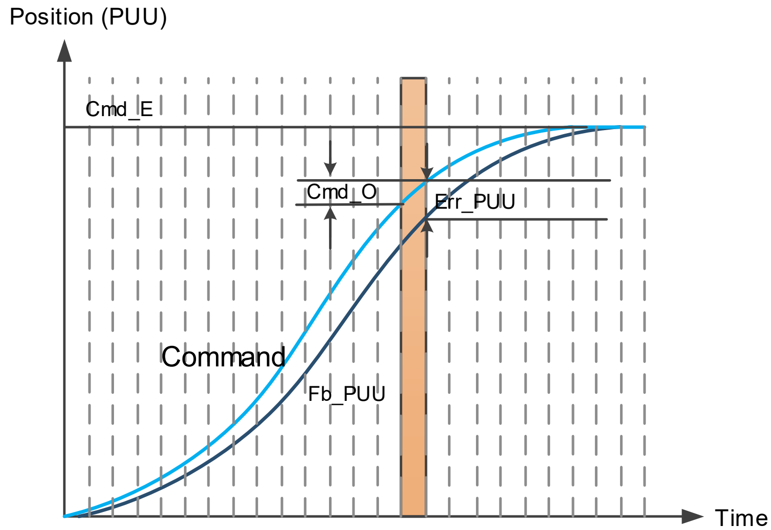


Figure 7-7 Monitoring variables status when executing a command in PR mode

You can use the digital inputs (DI) to trigger PR paths and digital outputs (DO) to monitor PR paths (refer to section 8.4.9 and 8.4.10 for DI and DO function descriptions). When you trigger the motion command with DI.CTRG [0x08], the servo drive operates based on the command from the internal registers. See section 7.1.5 for all the trigger methods for the PR paths. Once the execution is completed, DO.Cmd_OK [0x15] is set to on. When the position deviation (pulse number) becomes smaller than the set amount in P1.054, DO.TPOS [0x05] (Motor reaches the target position) is set to on. When both DO.TPOS and DO.Cmd_OK signals are on the servo outputs the MC_OK [0x17] signal to signify that it has completed this PR path. The operation is as shown in Figure 7-8. If you have set a delay time in this PR and the position deviation (pulse number) is smaller than the value of P1.054, DO.TPOS [0x05] is set to on.

When the delay time is over, DO.Cmd_OK [0x15] (PR position command complete) is set to on. After both of the above mentioned DO signals are on, DO.MC_OK [0x17] (Servo procedure complete) is set to on to signify it has completed this PR path, as shown in Figure 7-9.

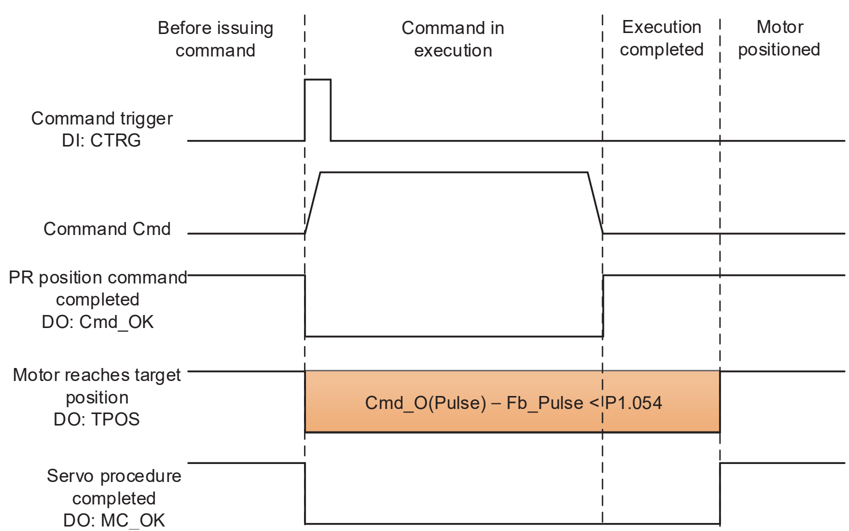


Figure 7-8 Operation of DI signals in PR mode

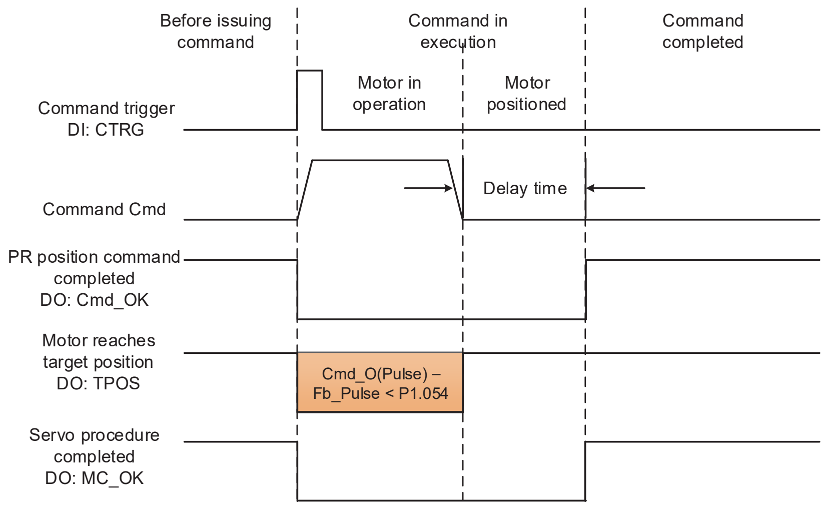


Figure 7-9 Operation of DO signals in PR mode

7.1.3 - Motion Control commands

The SureServo2 drive provides 100 path sets that can include homing methods, speed, position, path jumping, parameter writing, index positioning, and arithmetic operations. The following sections detail each command type.

Homing methods

The SureServo2 provides 11 homing methods in PR mode. They include home sensor (Origin Sensor = DI.ORG 0x24), limit sensor, and torque-level (collision point). Sub-selections such as whether to refer to the Z pulse and the limit signal as the trigger are available, with more than 30 combinations possible. The homing method is specified by P5.004 and the homing definition is determined by P6.000. Bit function bit is listed below.

P5.004	***Homing Methods***		***Address: 0508H 0509H***
Default:	0x0	Control mode:	PR
Unit:	-	Setting range:	0 – 0x128
Format:	HEX	Data size:	16-bit

Settings:

0000

U Z Y X

X	Homing method	Z	Limit setting
Y	Z pulse setting	U	Reserved

Definition of each setting value:

<table>
<tr><th>U</th><th>Z</th><th>Y</th><th>X</th></tr>
<tr><td>Reserved</td><td>Limit setting</td><td>Z pulse setting</td><td>Homing method</td></tr>
<tr><td rowspan="12">-</td><td>0 – 1</td><td>0 – 2</td><td>0 – A</td></tr>
<tr><td>-</td><td rowspan="2">Y = 0: return to Z pulse
Y = 1: go forward to Z pulse
Y = 2: do not look for Z pulse</td><td>X = 0: homing in forward direction and define PL (Positive Limit Sensor) as homing origin</td></tr>
<tr><td>-</td><td>X = 1: homing in reverse direction and define NL (Negative Limit Sensor) as homing origin</td></tr>
<tr><td rowspan="9">When limit sensor triggered:
Z = 0: show error
Z = 1: reverse direction</td><td rowspan="7">Y = 0: return to Z pulse
Y = 1: go forward to Z pulse
Y = 2: do not look for Z pulse</td><td>X = 2: homing in forward direction, DI.ORG: OFF→ON as homing origin</td></tr>
<tr><td>X = 3: homing in reverse direction, DI.ORG: OFF→ON as homing origin</td></tr>
<tr><td>X = 4: look for Z pulse in forward direction and define it as homing origin</td></tr>
<tr><td>X = 5: look for Z pulse in reverse direction and define it as homing origin</td></tr>
<tr><td>X = 6: homing in forward direction, DI.ORG: ON→OFF as homing origin</td></tr>
<tr><td>X = 7: homing in reverse direction, DI.ORG: ON→OFF as homing origin</td></tr>
<tr><td>X = 8: define current position as the origin</td></tr>
<tr><td rowspan="2">Y = 0: return to Z pulse
Y = 1: do not look for Z pulse</td><td>X = 9: look for the collision point in forward direction and define it as the origin</td></tr>
<tr><td>X = A: look for the collision point in reverse direction and define it as the origin</td></tr>
</table>

P6.000	***Homing Definition***		***Address: 0600H 0601H***
Default:	0x00000000	Control mode:	PR
Unit:	-	Setting range:	0x00000000 – 0x00000063
Format:	HEX	Data size:	32-bit

Settings:

Homing definition:

h0020 (D C B A)   L052A (U Z YX)

A	DEC2: deceleration time selection of second homing velocity	***YX***	PATH: path type
B	DLY: select 0 – F for delay time	***Z***	ACC: select 0 – F for acceleration time
C	N/A	***U***	DEC1: deceleration time selection of first homing velocity
D	BOOT	**-**	-

- *YX: PATH: path type*
  0x0: stop: homing complete and stop.
  0x1 – 0x63: auto: homing complete and execute the specified path (Path#1 – Path#99).
- *Z: ACC: select 0 – F for acceleration time*
  0 – F: corresponds to P5.020 – P5.035
- *U: DEC1: deceleration time selection of first homing velocity*
  0 – F: corresponds to P5.020 – P5.035
- *A: DEC2: deceleration time selection of second homing velocity*
  0 – F: corresponds to P5.020 – P5.035
- *B: DLY: select 0 – F for delay time*
  0 – F: corresponds to P5.040 – P5.055
- *D: BOOT: when the drive is powered on decide whether to initiate the homing routine or not.*
  0: do not execute homing
  1: execute homing automatically (servo must be enabled, SON, before homing will occur)

Apart from the above definitions, the related settings for homing also include:

1) P5.004 homing methods.
2) P5.005 – P5.006 speed setting of searching for the origin.
3) P6.001: Origin Definition (ORG_DEF) is the coordinate of the origin and may not be 0.

Notes:

1) After the origin is found (sensor or Z), it has to decelerate to a stop. The stop position exceeds the origin by a short distance. The motor will decel to another PUU value but the origin definition value (P6.001) was assigned at the Z pulse.
   If returning to the origin is not needed, set PATH to 0;
   If returning to the origin is needed, set PATH to a non-zero value and have that PATH execute an absolute move to the Origin Definition entered in P6.001.
   Example:
   Upon completion of P6.000 = 0x1, automatically execute Path#1.
   Set from absolute position (ABS) to 0 as the route of Path#1 (set P6.002 & P6.003).
2) If the origin is found (sensor or Z), and you want it to move an offset S and define the coordinate as P after moving, then PATH = non-zero and set ORG_DEF = P - S, and this absolute Position command = P.

The PR Homing mode includes the function for setting the origin offset. You can define any point on the coordinate axis as the reference origin, which does not have to be 0. Once you define the reference origin, the system can create the coordinate system for the motion axis. See Figure 7-10. The coordinate for the reference origin is 2000 (P6.001 = 2000). The motor passes by the reference origin and then stops at coordinate 1477. From the coordinate system that it created, the system automatically calculates the position of 0 point. As soon as the PR motion command is issued, the motor moves to the specified position.

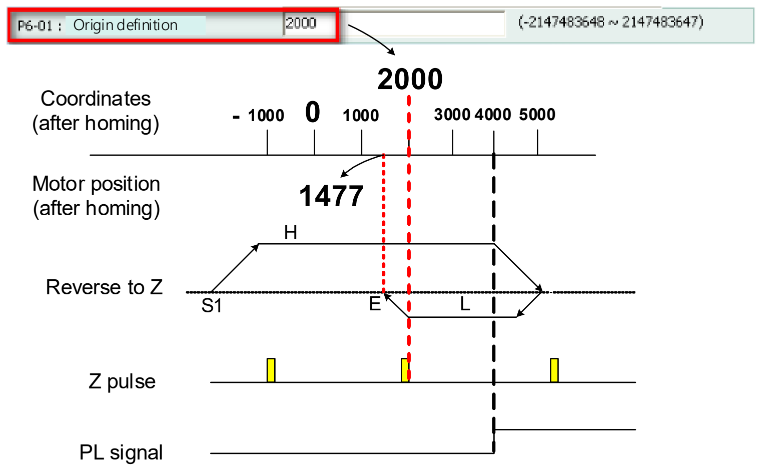


Figure 7-10 Origin Definition

P6.001	***Origin Definition***		***Address: 0602H 0603H***
Default:	0	Control mode:	PR
Unit:	-	Setting range:	-2147483648 – 2147483647
Format:	DEC	Data size:	32-bit

Settings:

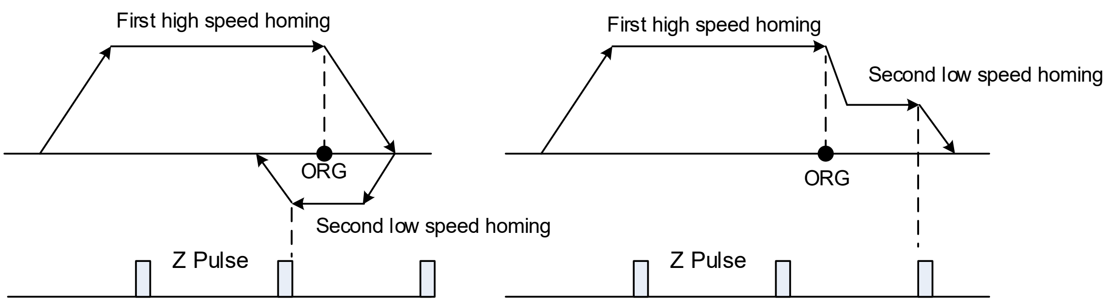
Origin definition (Home definition). This is the axis coordinate value (offset) that you want for this axis to be at the origin sensor. For example, if you use the rising edge of the Z pulse as the home origin, you can make that point on the coordinate system 3000 instead of 0 by entering 3000 in P6.001. In this example the Home Origin is 3000 instead of 0.

The homing procedure goes through two stages: high speed and low speed. Homing starts in high speed, seeking the reference point (such as the limit switch and ORG signal). Once the servo detects the reference point, the motor runs at low speed to find the reference point accurately (such as the Z pulse). The speeds for the two stages are defined by P5.005 and P5.006.

P5.005	***High Speed Homing (First Speed Setting)***			***Address: 050AH 050BH***
Operation interface:	Panel / software	Communication	Control mode:	PR
Default:	100.0	1000	Data size:	32-bit
Unit:	1 rpm	0.1 rpm		
Setting range:	0.1 – 2000.0	1 – 20000		
Format:	DEC	DEC	-	-
Example:	1.5 = 1.5 rpm	15 = 1.5 rpm	-	-

Settings:

The first speed setting for high speed homing.



P5.006	***Low Speed Homing (Second Speed Setting)***			***Address: 050CH 050DH***
Operation interface:	Panel / software	Communication	Control mode:	PR
Default:	20.0	200	Data size:	32-bit
Unit:	1 rpm	0.1 rpm		
Setting range:	0.1 – 500.0	1 – 5000		
Format:	DEC	DEC	-	-
Example:	1.5 = 1.5 rpm	150 = 1.5 rpm	-	-

Settings:

The second speed setting for low speed homing.

You can set the homing parameters in the PR mode homing screen in SureServo2 Pro, including the homing methods, homing definition, and homing speed (see Figure 7-11).

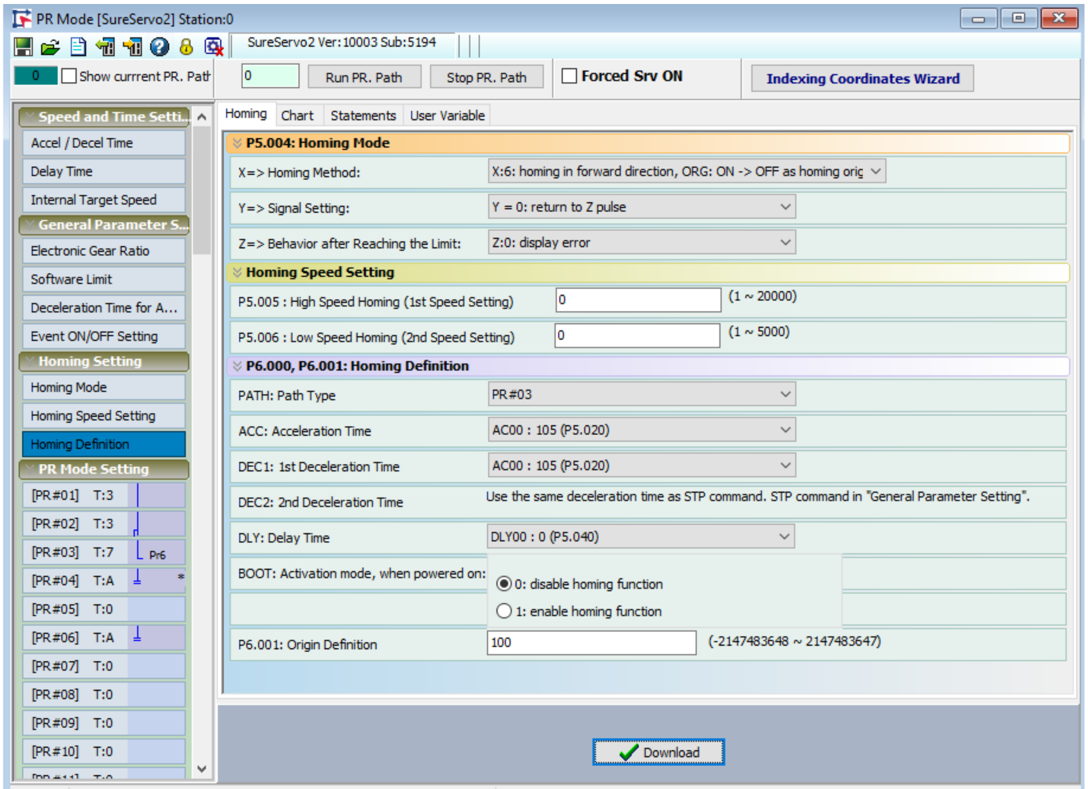
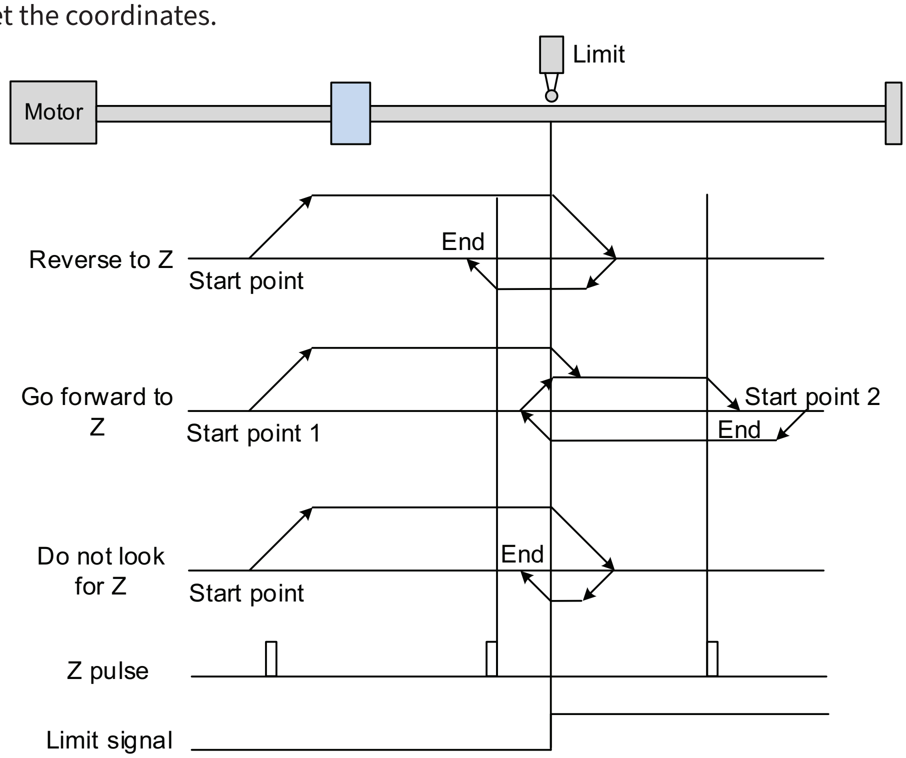


Figure 7-11 Homing screen in SureServo2 Pro

The following describes the homing methods supported by the SureServo2 drive. They can be categorized into six types based on their reference points.

1) **Referencing the limit:** this homing method uses the positive or negative limit as the reference point. When the limit is detected, you can choose to look for the Z pulse and use it as the homing reference point. Changing the starting position does not change the homing result. The drive always looks for the setting reference point so as to correctly reset the coordinates.

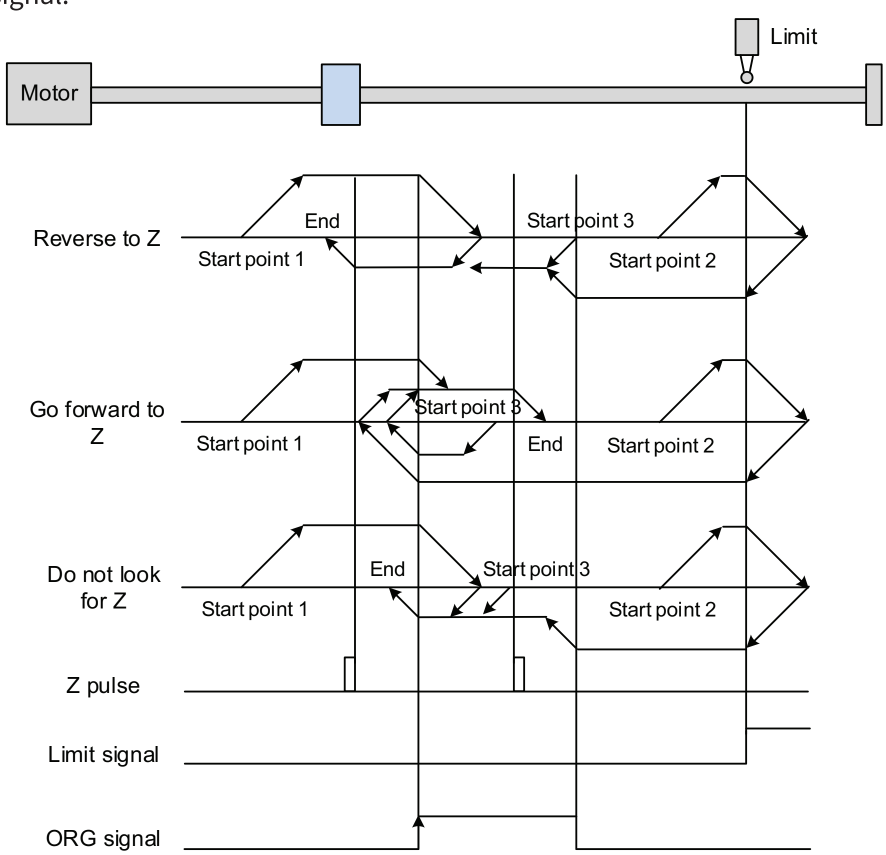


In the example above (looking for the Z pulse), the servo motor operates at high speed (first speed value) and then decelerates once it reaches the limit (rising-edge triggers the signal). Then the motor switches to low speed (second speed value) to look for the Z pulse. When the motor finds the Z pulse, it decelerates and stops, completing the homing procedure.

If you set the motor to look for the Z pulse and the limit signal remains un-triggered (low, Start point 1), the servo motor operates at high speed (first speed setting) and then decelerates once it reaches the limit (rising-edge triggers the signal). Then the motor switches to low speed (second speed setting) to look for the Z pulse. When the motor finds the Z pulse, it decelerates and stops, completing the homing procedure. The motor will decel to another PUU value but the origin definition value (P6.001) was assigned at the Z pulse. If the setting is to look for the Z pulse and the limit signal is triggered (high, Start point 2), the servo motor returns to look for the falling-edge trigger signal at low speed (second speed setting). Once it is found, the servo motor starts to look for the Z pulse and decelerates to stop when it finds the Z pulse., completing the homing. In conclusion, the set origin is at the same position after homing with the same condition regardless of the location of the starting point.

If you set the motor to not look for the Z pulse, the servo motor first operates at high speed (first speed setting) and then decelerates to a stop once rising-edge limit signal is triggered. Then the motor changes to low speed (second speed setting) to look for rising-edge signal. Once it finds the rising-edge signal, the motor decelerates to a stop, completing the homing.

2) **Referencing the rising-edge signal of the home sensor:** This method uses the rising-edge of the home sensor (ORG) signal as the reference point. You have the option of using the Z pulse as the reference point of the origin when the home sensor detects the signal.



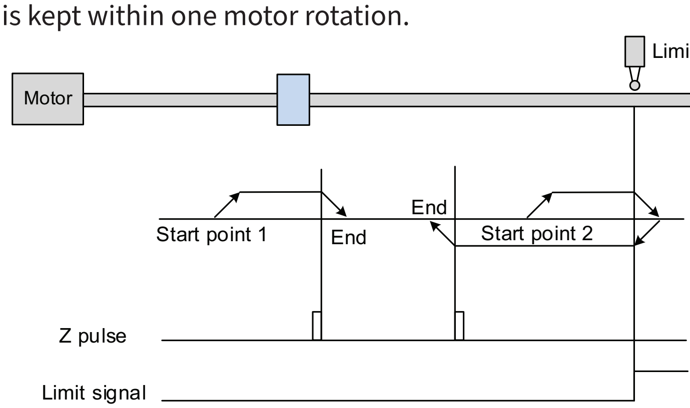
The figure above shows an example of reversing to look for the Z pulse. If the home sensor signal (DI.ORG sensor) for the start position is untriggered (low, Start point 1), the servo motor operates at high speed (first speed setting) until it reaches the rising-edge of ORG signal. Then it decelerates, switching to low speed (second speed setting) and reverses to look for the Z pulse. When the motor finds the Z pulse, it decelerates to a stop, completing the homing.

If the ORG signal at the start point is untriggered and the current position is relatively closer to the limit switch (Start point 2), then the servo motor operates at high speed (first speed setting). You can choose whether to show an error (P5.004.Z=0) or reverse the running direction (P5.004.Z=1) when it reaches the limit switch. If you choose to reverse the rotation direction, the servo motor keeps rotating in reverse direction. Once the motor reaches the limit switch, it changes to low speed (second speed setting) and operates until the ORG signal switches to low. Next, it starts to look for the Z pulse. When the motor finds the Z pulse, it decelerates to a stop, completing the homing.

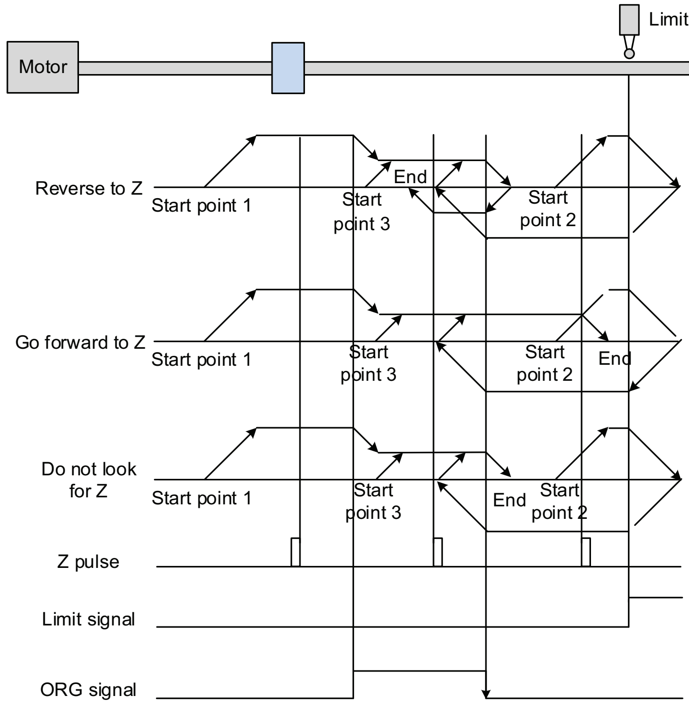
If the ORG signal is triggered (high, Start point 3), the motor reverses with low speed (second speed setting) and after the ORG signal switches to low, the motor returns to look for the Z pulse. Once the Z pulse is found, homing is complete.

If you set the server motor to look for the Z pulse, or not to look for the Z pulse in the forward direction (this is similar to the first method mentioned above, going in the reverse direction or not to look for the pulse Z), please refer to the timing diagram above.

3) **Referencing the Z pulse:** This method uses the Z pulse as the reference origin. One Z pulse is generated per rotation of the motor. This method is only suitable when the operation is kept within one motor rotation.



4) **Referencing the falling-edge of the ORG signal:** This method uses the falling-edge signal of the home sensor as the reference origin. You can choose whether or not to use Z pulse as the reference origin after the ORG signal is detected.



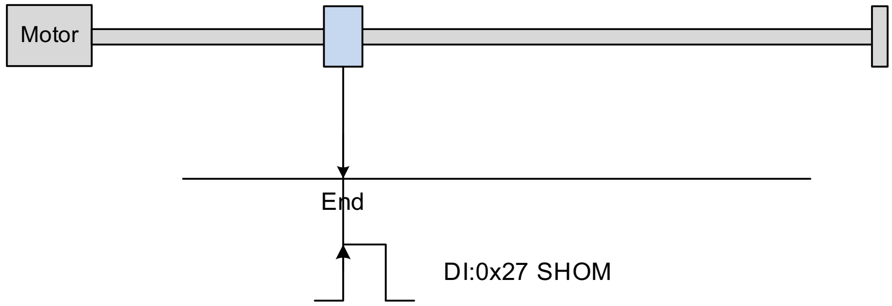
See the above example of looking for the Z pulse. If ORG signal is un-triggered at the start point (low, Start point 1), the servo motor runs at high speed (first speed setting) until reaching the rising edge of the ORG signal. Then it decelerates and switches to low speed until the ORG signal is off (low). Next, it reverses to look for the Z pulse and decelerates to a stop once it finds the Z pulse, completing the homing.

If ORG signal is un-triggered at the start point and is closer to the limit switch (Start point 2), the motor runs at high speed (first speed setting). You can set whether to show an error or reverse the running direction when it reaches the limit switch. If you set it to reverse direction, the motor operates in reverse to reach the ORG signal. Once it reaches the ORG signal, it decelerates and runs at low speed (second speed setting) until it reaches the falling edge of ORG signal. Then it reverses to look for Z pulse. When found, the servo decelerates to a stop, completing the homing.

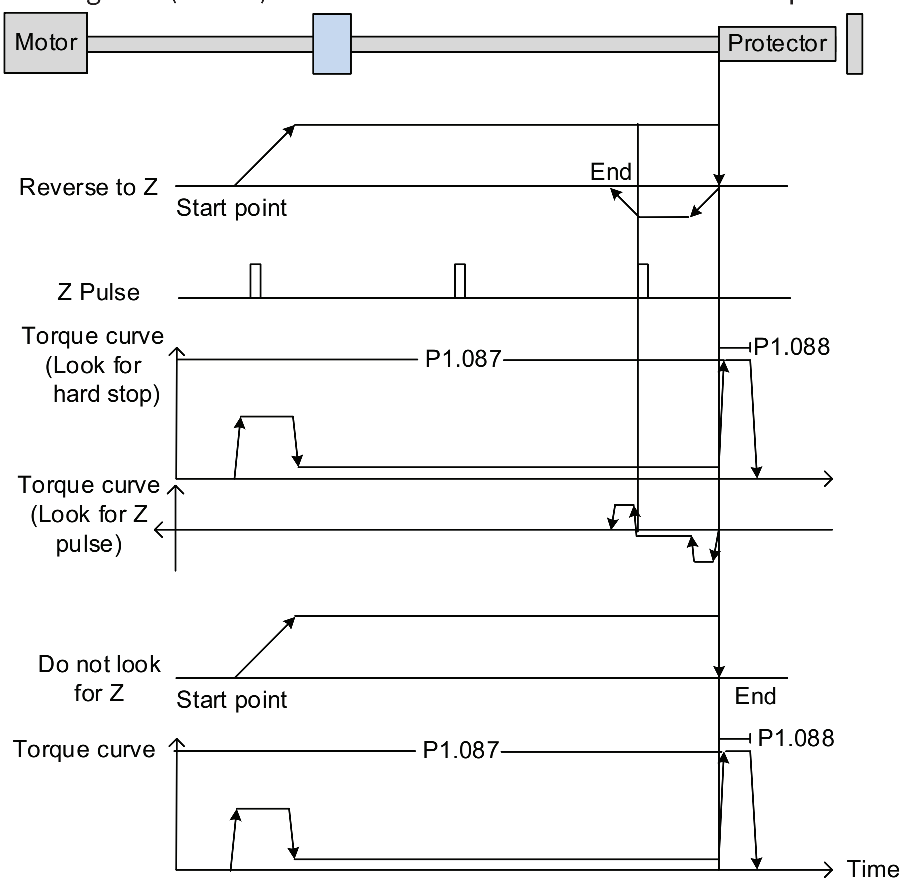
If the ORG signal is triggered at the start point (high, Start point 3), the servo motor operates at low speed (second speed setting) in the forward direction until the ORG signal switches to low. Finally, the motor reverses to look for the Z pulse and decelerates to a stop, completing the homing.

If you set it to look for the Z pulse or to not to look for the Z pulse in the forward direction, which is similar to the first setting mentioned above (going in the reverse direction or not to look for the Z pulse), please refer to the timing diagram above.

5) **Referencing the current position as the origin:** This method uses the motor's current position as the reference origin. As long as the homing procedure is triggered and the motor remains still, then coordinate positioning is completed.



6) **Referencing the torque limit:** This method uses the motor's stop position as the origin by referring to a mechanical limit on the machine, the torque level setting (P1.087), and the retaining time (P1.088). You can also choose whether to use the Z pulse as the origin.



In the figure above that uses looking for the Z pulse, the motor runs at the high speed (first speed setting). Then the servo outputs a greater current to resist the external force once the motor touches the collision protector. When the motor torque reaches the set limit (P1.087) and the output duration is longer than the time setting (P1.088), the motor runs in the reverse direction to look for the Z pulse at low speed (second speed setting). Once the motor finds the Z pulse, it decelerates to a stop, completing the homing.

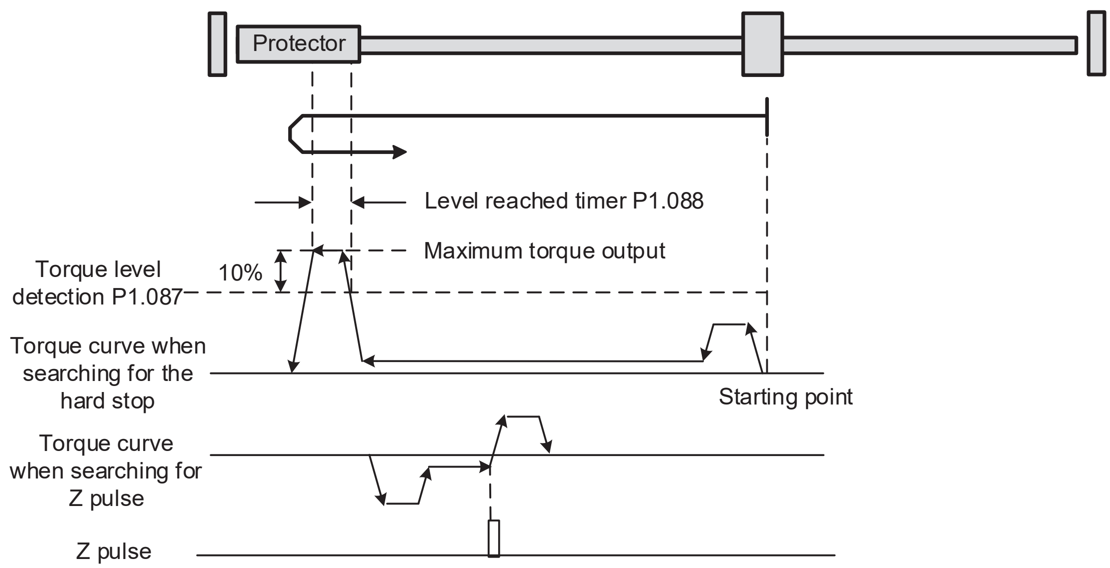
If you set it not to look for the Z pulse, the servo motor runs at high speed (first speed setting) until it touches the collision protector. Then the servo outputs a greater current to resist the external force. When the motor torque reaches the set limit (P1.087) and the output duration is longer than the time setting (P1.088), the motor stops, completing the homing.

The following table lists the parameters for the torque limit (P1.087) and the torque limit time (P1.088):

P1.087	***Torque Limit (Torque Level)***		***Address: 01AEH 01AFH***
Default:	1	Control mode:	PR
Unit:	%	Setting range:	1 – 300
Format:	DEC	Data size:	16-bit

Settings:

The Torque limit is only for Torque limit homing mode. As shown in the following diagram, when the homing command is triggered, the motor runs in one direction until it reaches the collision protector. After reaching the collision protector, the servo drive outputs a larger motor current to counter the external force from the collision protector. The servo drive uses the motor current and the Torque limit time to determine homing, and then it runs in the opposite direction to find the Z pulse.



Note: The actual maximum torque output of the motor is 10% greater than the detected torque level (P1.087). For example, set P1.087 to 50%, then the maximum torque output of the motor is 60%.

P1.088	Torque limit time		Address: 01B0H 01B1H
Default:	2000	Control mode:	PR
Unit:	millisecond	Setting range:	2 – 2000
Format:	DEC	Data size:	16-bit

Settings:

Set the Torque limit time for Torque limit homing mode. Please refer to P1.087 for the timing diagram of Torque limit homing mode.

PR Monitoring Variables

As mentioned in Section 7.1.2, the PR mode provides four monitoring variables for you to monitor the servo commands and feedback status. These are Command position PUU (Cmd_O), PR command end register (Cmd_E), Feedback position PUU (Fb_PUU), and Position error PUU (Err_PUU). Before homing completes, the command end register (Cmd_E) cannot be calculated because the coordinate system can only be created after homing is completed, and the target position remains unknown after the Homing command is issued. This is why the status of each monitoring variables is different during homing. In Homing command's default setting, the contents of Cmd_E and Cmd_O are identical. After it finds the reference origin in the coordinate system, it sets the content of Cmd_E to the coordinate of the reference origin. However, once it finds the reference origin, it still requires some distance for motor to decelerate to a stop. Meanwhile, Cmd_O continues to issue commands. If no other PR commands are issued after homing (other than the Position command), the contents of the final command position (Cmd_O) and command end position (Cmd_E) will be different. See Figure 7-12.

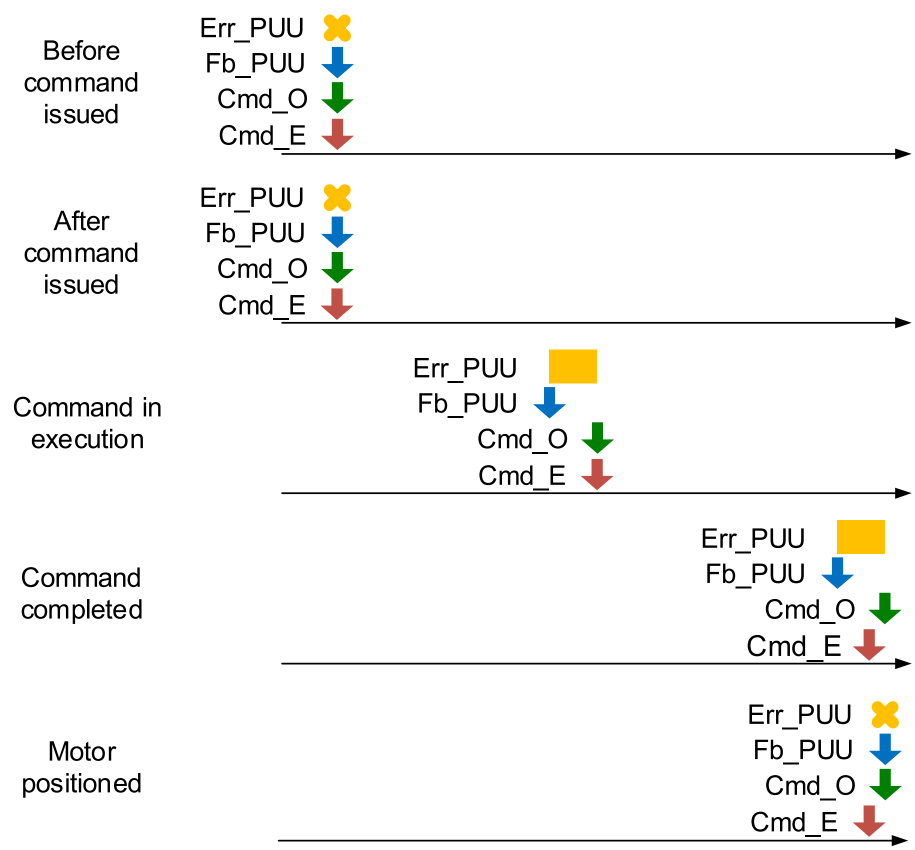


Figure 7-12 Homing mode and monitoring variables

Wiring
Parameters
DI/DO Codes
Monitoring
Alarms

Constant Speed Control

The PR mode provides a speed control function. The following parameters are available for PR speed setting: acceleration / deceleration time, delay time, and target speed. You can easily set the Speed command in the PR mode screen in SureServo2 Pro. See Figure 7-13.

INS (or Insert) is an interrupt command that interrupts the previous motion command. Please refer to Section 7.1.6 for more details. AUTO is a command that automatically loads the next PR path. It executes the next PR path when the current PR path completes. In addition, you can set the target speed with two unit options, which are 0.1 rpm and 1 PPS ranging from -6000 rpm–6000 rpm. You select the ACC/DEC acceleration / deceleration time with the shared PR parameters. The software calculates and displays the required duration for accelerating from 0 to the target speed. DLY is the delay time that is determined by the shared PR parameters. In Constant Speed Control the Delay starts counting after the Target Speed has been reached and only if the AUTO selection is on. If the PR path is not set up to automatically proceed to the next path then the Delay does not apply.

See Figure 7-14 for the effects of the parameters for the PR mode speed control. Table 7-2 shows the bit function when speed control is in operation.

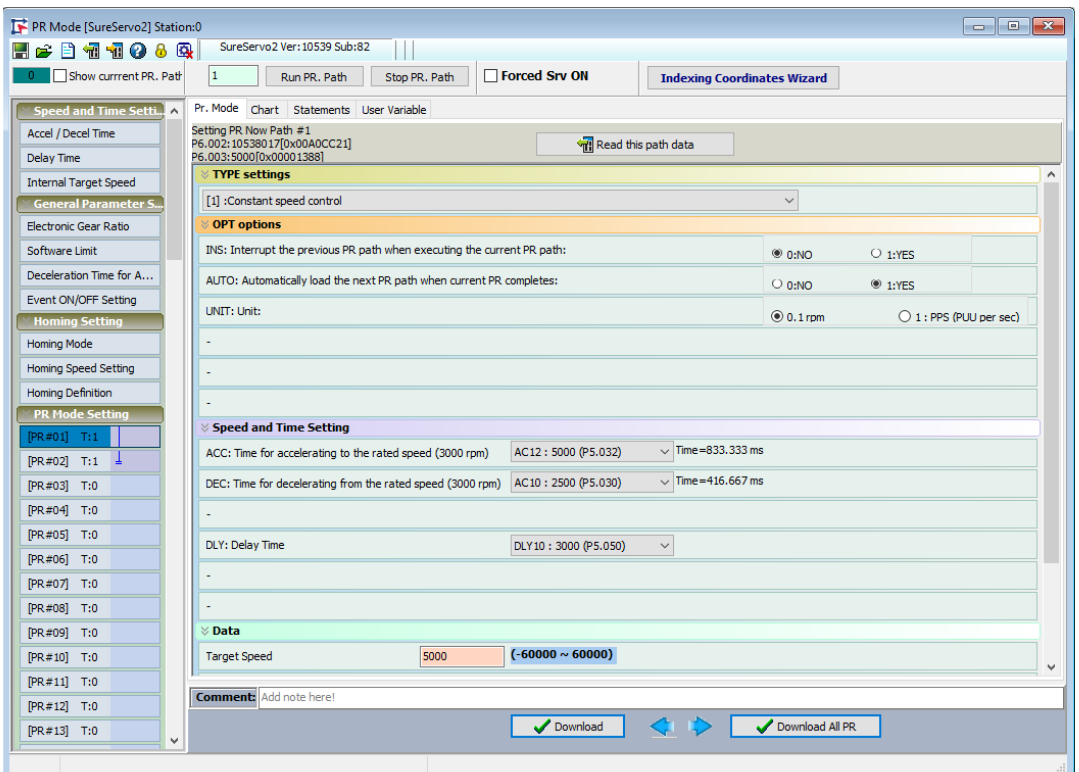


Figure 7-13 PR mode Speed screen in SureServo2 Pro

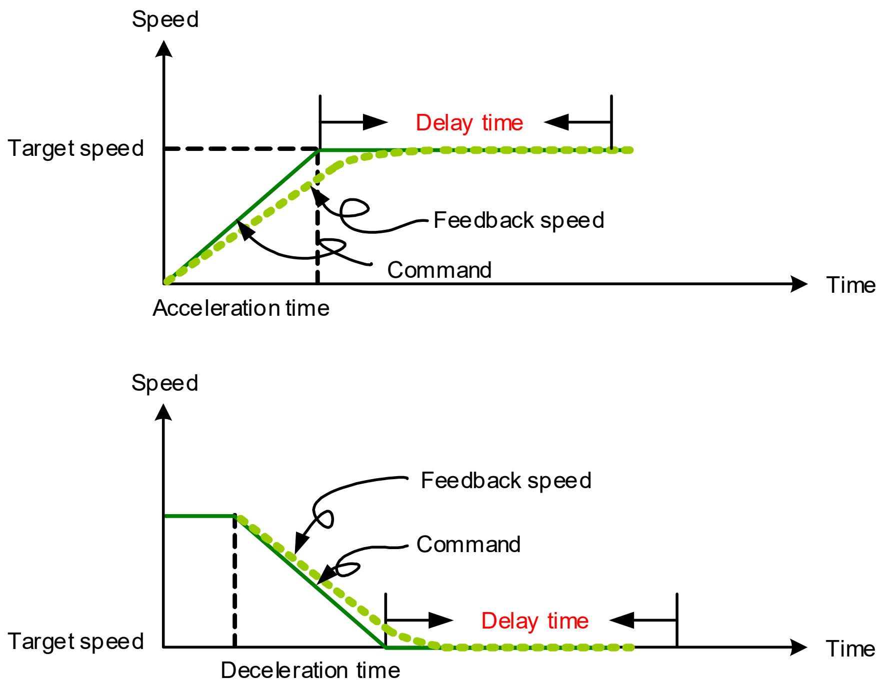


Figure 7-14 Parameters for PR mode speed control

Table 7-2 Bit function of PR speed control

PR Parameters	D	C	B	A	U	Z	Y	X
Command Type	-	-	DLY	-	DEC	ACC	OPT	1
Data Content	Target speed [0.1 rpm / PPS]							

Notes:

1) Y OPT: option

BIT	3	2	1	0
Command Type	-	UNIT	AUTO	INS

INS: Interrupt command that interrupts the previous motion command.
AUTO: automatically load the next PR command when the current one is completed.
UNIT: speed unit selection; 0 signifies 0.1 rpm and 1 signifies PPS.

2) Z, U: ACC / DEC: acceleration / deceleration time, set by P5.020 – P5.035.
3) B: DLY: delay time, set by P5.040 – P5.055.

Point-to-Point Command

PR mode includes a position control function. The Position command is user-defined and its unit is PUU. There are two command types: Mode 2 and Mode 3. In Mode 2 (Point-to-Point Command) the command signifies that it stops once the command is completed. In Mode 3 (Point-to-Point Command & proceed to next path) the command signifies that the next PR path is automatically executed. You use the same method to set the value for these modes in SureServo2 Pro. See Figure 7-15.

Wiring
Parameters
DI/DO Codes
Monitoring
Alarms

In Figure 7-15, INS stands for the interrupt command that interrupts the previous motion command. OVLP stands for the overlap command that allows the next PR command to overlap the command that is currently being executed when decelerating. If you apply this this function, setting the delay time to 0 is suggested (please refer to Section 7.1.6). ACC / DEC is the acceleration / deceleration time determined by the shared PR parameters. The software calculates and displays the required time to accelerate from 0 to the target speed. SPD is the target speed specified by the shared PR parameters. You can choose whether it is multiplied by 0.1. DLY is the delay time specified by the shared PR parameters and it is defined by the command; in other words, once the target position is reached, the delay time starts counting.

The Position command for PR mode is illustrated in Figure 7-16. Table 7-3 lists the bit functions of position control.

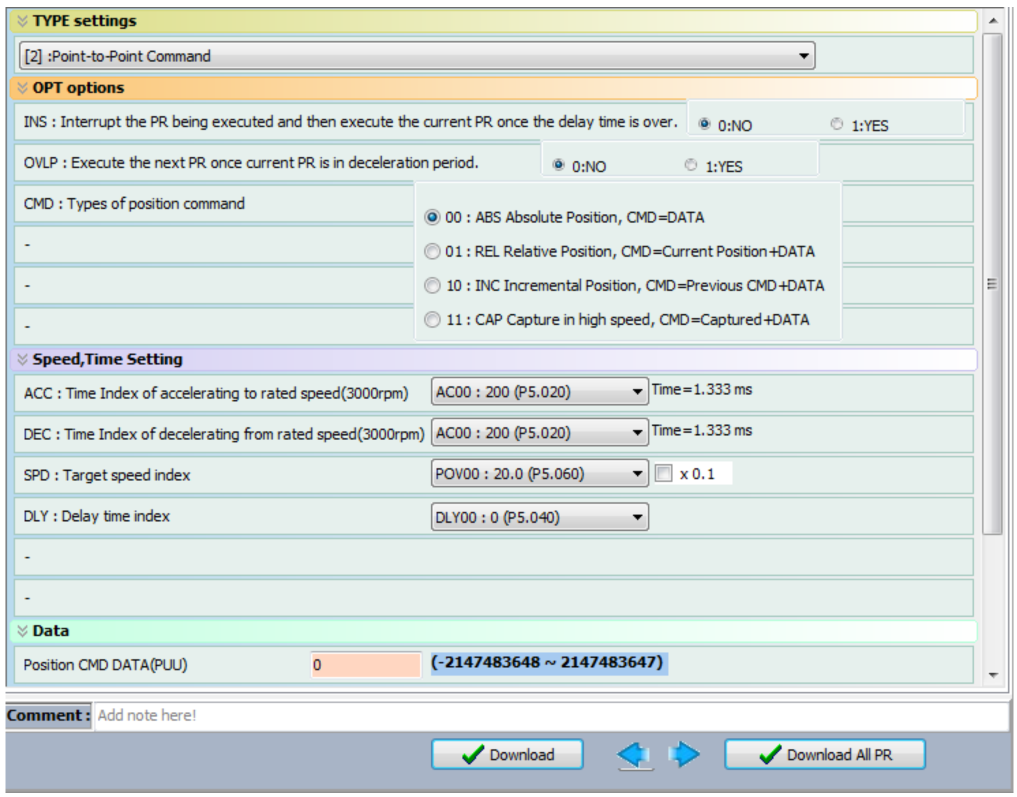


Figure 7-15 PR mode position interface of SureServo2 Pro

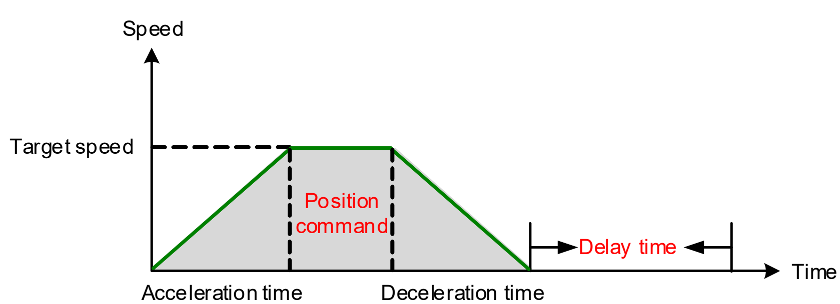


Figure 7-16 Parameters for PR mode position setting

There are four types of position commands for the PR mode. You can choose the position command according to the application requirements. The functions of each type are described in the examples below. Please note that the condition in these examples is that a position command is still being executed and another type of command is inserted. To see how the position commands are combined, please see Figure 7-17.

1) ***Absolute Position Command*** (ABS): when executed, the target position value equals the absolute command value. In the figure, an ABS command with the value of 60000 PUU is inserted in the previous PR path with setting target position of 60000 PUU on the coordinate axis.
2) ***Relative Position Command*** (REL): when executed, the target position value is the motor's current position value plus the position command value. In the figure, a REL command with the value on 60000 PUU is inserted in the previous PR path. The target position is the motor's current position (20000 PUU) plus the relative position command (60000 PUU), which equals 80000 PUU in the coordinate system. The target position specified by the original command is overwritten. When using REL or INC in a PR path as an interrupt (I) then they behave the same way. They both act as a REL command. See Figure 7-49 for this behavior.
3) ***Incremental Command*** (INC): when executed, the target position is the previous target position value plus the current position command value. In the example below, an INC command with the value of 60000 PUU is inserted in the previous PR path. The target position is the previous target position value 30000 PUU plus the relative position command 60000 PUU, which equals 90000 PUU. The previous destination specified by the previous command is combined to define the new one. When using REL or INC in a PR path as an interrupt (I) then they behave the same way. They both act as a REL command. See Figure 7-49 for this behavior.
4) ***High-Speed Position Capturing Command*** (CAP): when executed, the target position is the last position acquired by the Capture function plus the position command value. Please refer to Section 7.2.2 for more on the high-speed position capturing function. In the following example, a high-speed capturing command with the value of 60000 PUU is inserted in the previous PR path. The target position value is the captured position value of 10000 PUU plus the relative command of 60000 PUU, which equals 70000 PUU. The target position specified by the original command is omitted.

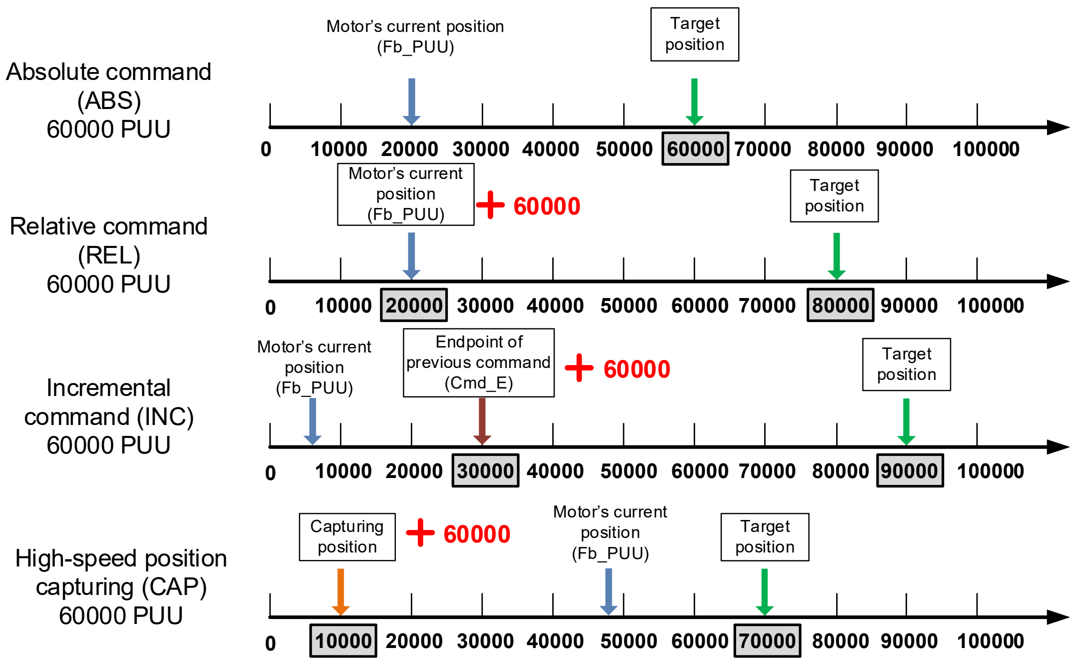


Figure 7-17 Four types of position command

Table 7-3 Bit functions of parameters of PR mode Position control

PR Parameters	***D***	***C***	***B***	***A***	***U***	***Z***	***Y***	***X***
Command Type	-	-	DLY	SPD	DEC	ACC	OPT	2 or 3
Data content	Target position [PUU]							

Notes:

1) Y: OPT: option

BIT	***3***	***2***	***1***	***0***	***Description***
Command Type	CMD		OVLP	INS	-
Data Content	0	0	-	-	ABS (absolute positioning)
	0	1			REL (relative positioning)
	1	0			INC (incremental positioning)
	1	1			CAP (high-speed position capturing)

INS: Interrupt command interrupts the previous motion command.
OVLP: allow overlapping of the next command
CMD: Position command selection

2) Z, U: ACC / DEC: acceleration / deceleration time, set by P5.020 – P5.035.
3) A: SPD: internal target speed number, set by P5.060 – P5.075.
4) B: DLY: delay time, set by P5.040 – P5.055.

Jump to the Specific Path

PR mode includes a Jump command. It can call any PR paths or form PR paths into a loop, as shown in Figure 7-18. You can specify the PR path number to jump to in the PR mode screen in SureServo2 Pro (see Figure 7-19). INS stands for the interrupt command that interrupts the previous motion command. You can find more information in Section 7.1.6. DLY is the delay time determined by shared PR parameters. Once a Jump command is issued, the servo drive starts counting the delay time then jumps to the specific path. Available target PR numbers are PR#00 – PR#99. Table 7-4 shows the functions of each bit when executing a Jump command.

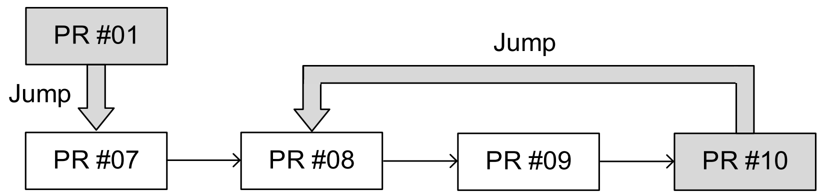


Figure 7-18 Jump command in PR mode

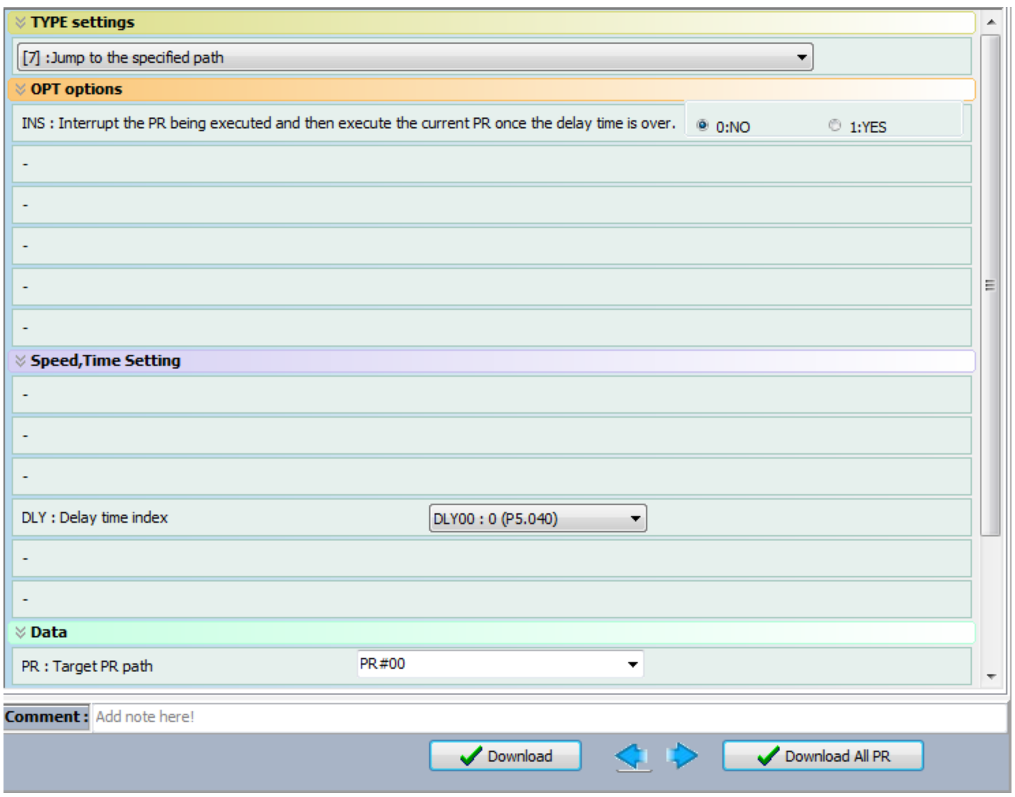


Figure 7-19 Using PR mode Jump command in SureServo2 Pro

Table 7-4 Bit function of PR Jump command

PR Parameters	D	C	B	A	U	Z	Y	X
Command Type	-	-	DLY	-	-	-	OPT	7
Data Content	Jump to target PR path (0 – 99)							

Notes:

1) Y: OPT: option

BIT	3	2	1	0
Command Type	-	-	-	INS

INS: Interrupt command; interrupts the previous motion command.

2) B: DLY: delay time, which is set in P5.040 – P5.055.

Write command

PR mode includes a Write command. It can write constants, parameters, data array elements, and monitoring variables to the specified parameters or to data array elements. You can write a parameter to a specified path in the PR mode screen in SureServo2 Pro (see Figure 7-20). INS is an interrupt command, which interrupts the previous motion command. Refer to Section 7.1.6 for more details. AUTO command automatically loads and executes the next PR once the current PR completes. ROM command writes parameters to both RAM and EEPROM at the same time. Writing to non-volatile memory function is also available; however, frequent writes shortens the life of the EEPROM. DLY is the delay time selected by shared PR parameters. Once the write command is issued the write happens immediately and the delay is applied after the data is written. Table 7-5 shows the functions of each bit when executing a Write command.

Writing Target	Data Source
Parameter	Constant
Data array	Parameter
-	Data array
-	Monitoring variables

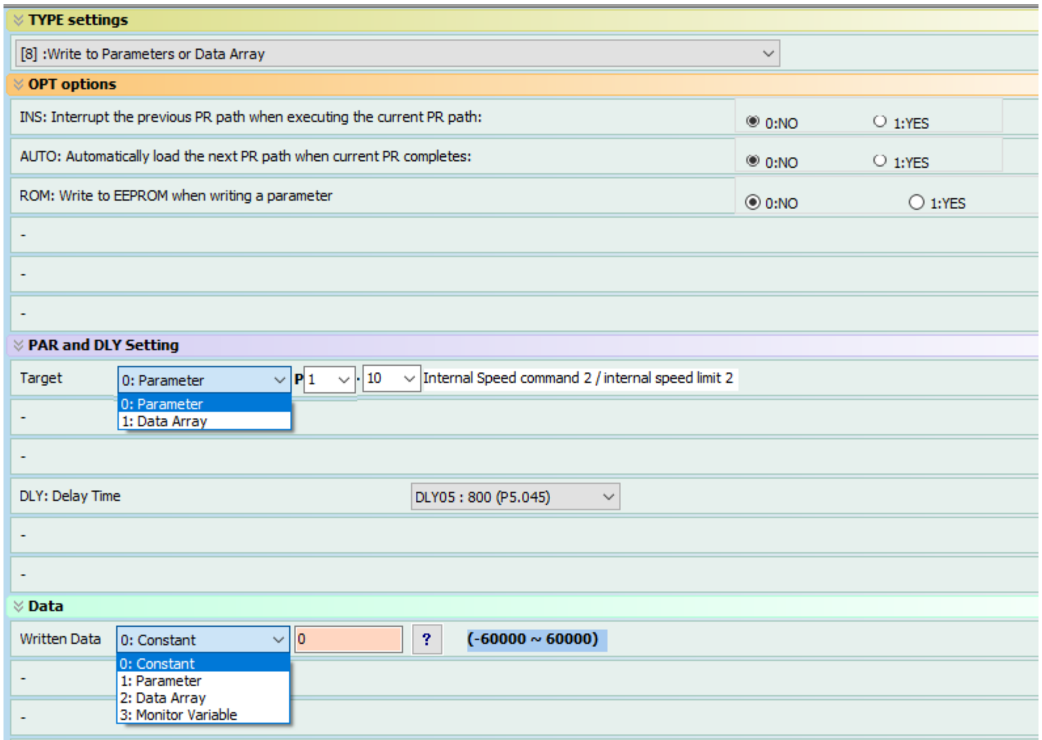


Figure 7-20 Using PR Write command in SureServo2 Pro

Table 7-5 Bit function for PR Write command

PR Parameters	***D***	***C***	***B***	***A***	***U***	***Z***	***Y***	***X***
Command Type	0	SOUR_ DEST	DLY	DESTINATION			OPT	8
Data Content	SOURCE							

Notes:

1) Y: OPT: option

BIT	***3***	***2***	***1***	***0***
Command Type	-	ROM	AUTO	INS

INS: Interrupt command interrupts the previous motion command.
AUTO: once the current PR is completed, automatically load the next command.
ROM: write data to RAM and EEPROM at the same time. This function is only available when the writing Target is a Parameter, it is not available when the writing Target is a Data Array.

2) B: DLY: delay time, which is set in P5.040 – P5.055.

Wiring
Parameters
DI/DO Codes
Monitoring
Alarms

3) C: SOUR_DEST: data source and data format to be written.

BIT	3	2	1	0	Description	
Command Type	SOUR		-	DEST	Data source	Writing target
Data Content	0	0	0	0	Constant	Parameter
	0	1		0	Parameter	Parameter
	1	0		0	Data array	Parameter
	1	1		0	Monitoring variable	Parameter
	0	0		1	Constant	Data array
	0	1		1	Parameter	Data array
	1	0		1	Data array	Data array
	1	1		1	Monitoring variable	Data array

4) Z, U, A: DESTINATION: destination

	A	U	Z
Target: Parameter	Parameter group	Parameter number	
Target: Data array	Data array number		

5) SOURCE: data source setting

	D	C	B	A	U	Z	Y	X
Data Source: Constant	Constant data							
Data source: Parameter	-					Parameter group	Parameter number	
Data source: Data array	-					Data array number		
Data source: Monitoring variable	-						Monitoring variable number	

Index Position command

PR mode includes an Index Position command, which creates an index coordinate system. The system must have completed the homing routine before indexing can occur. This command positions the motor within the indexing coordinates. Unlike other feedback positions in global coordinate system, index positioning is able to divide the total moving distance of one index into the number of paths required by the application (see Figure 7-21). Please refer to Chapter 10 for absolute position or if position counter overflows occur due to index positioning command. You can start the index positioning in the **Index Coordinates Setting Wizard** in the PR screen in SureServo2 Pro (see Figure 7-22). As shown in the example, the start PR path is set to 1, the Path Size (number of index locations) is set to 8, and total moving distance (Indexing Coordinate Scale) is 100000 PUU. When you click **OK**, the software automatically writes position command 0 PUU to PR#01, 12500 PUU to PR#02, 25000 PUU to PR#03, and so on up to PR#08. When the index position reaches 100000 PUU, it automatically returns to 0 PUU. The index path parameters will automatically populate PR1-PRn. If you want to leave open spaces in between each indexed PR path you can enter a value of how many empty PR registers you need by using “Set the interval between two paths” option if you want to have other actions occur after the index moves.

In addition, you can modify each individual index position in each PR path as needed, as shown in Figure 7-23. INS stands for the interrupt command that interrupts the previous motion command (see Section 7.1.6.). OVLP stands for the overlap command that allows the next PR command to overlap the current one during deceleration. If you use OVLP, setting the delay time to 0 is recommended (refer to Section 7.1.6). DIR sets the rotation direction with options of forward (always runs forward), backward (always runs backward), and shortest distance. The movement is illustrated in Figure 7-24. S_LOW is the speed unit with options of 0.1 rpm or 0.01 rpm. AUTO automatically loads and executes the next PR path when current PR completes. ACC / DEC is the acceleration / deceleration time setting determined by shared PR parameters. SPD is the target speed set by the PR shared parameters. DLY is the delay time defined by the command from controller; meaning that when motor reaches the target position, the servo drive starts counting the delay time. Position command is the target position of each index segment. Please note that the setting range must be smaller than the total index moving distance (P2.052). Table 7-6 shows the bit function for the Index Position command. If you use the index function, please execute homing in order to create the coordinate system so that the origin of the motor’s feedback position and that of the motor’s index position can be identical. If you do not execute homing, AL237 occurs.

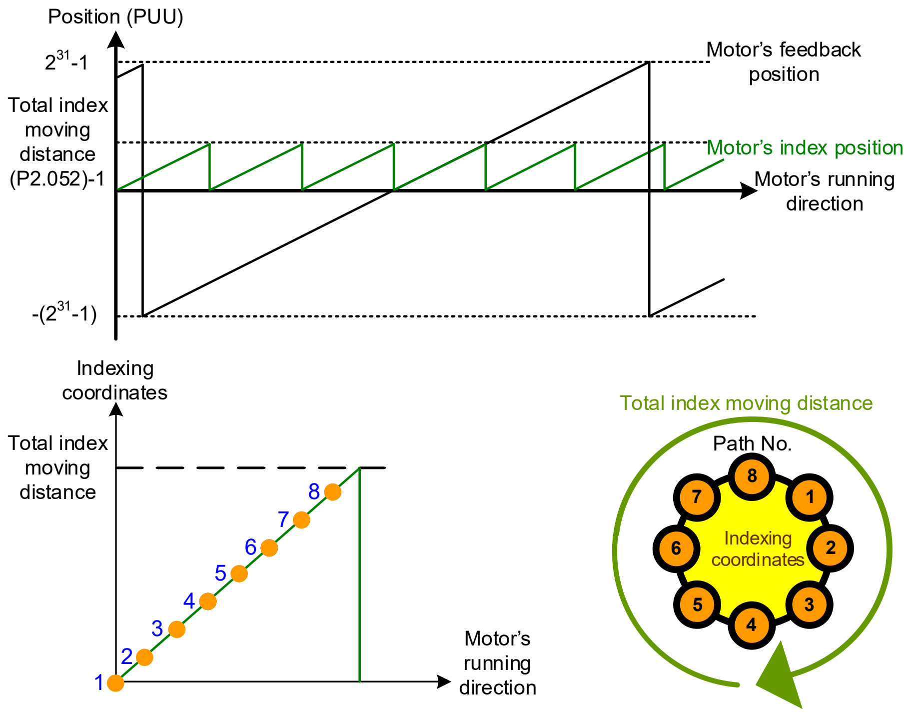


Figure 7-21 PR mode indexing coordinates

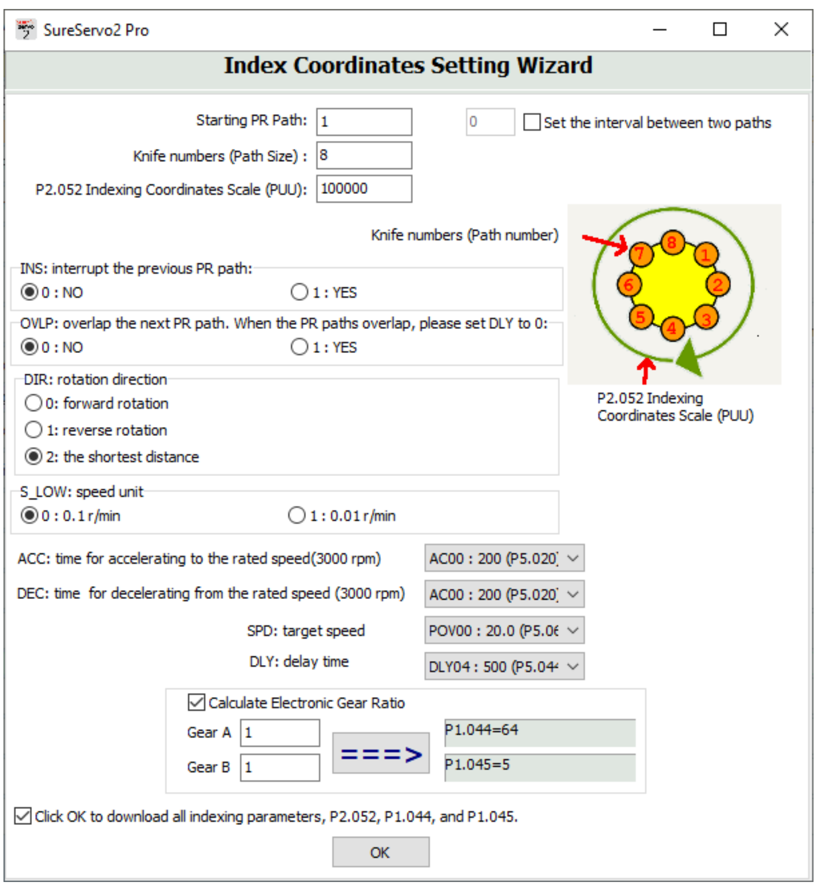


Figure 7-22 Indexing Coordinates Setting Wizard in PR mode

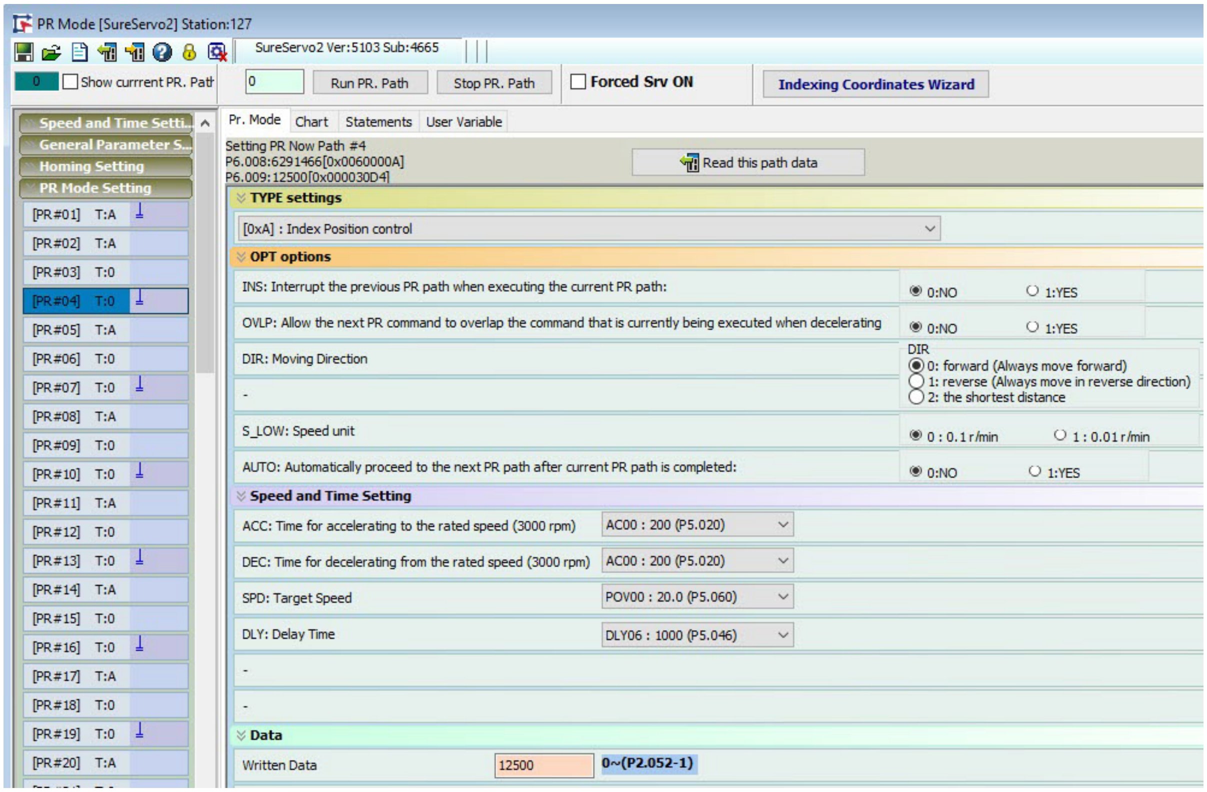


Figure 7-23 PR mode Index Position screen in SureServo2 Pro

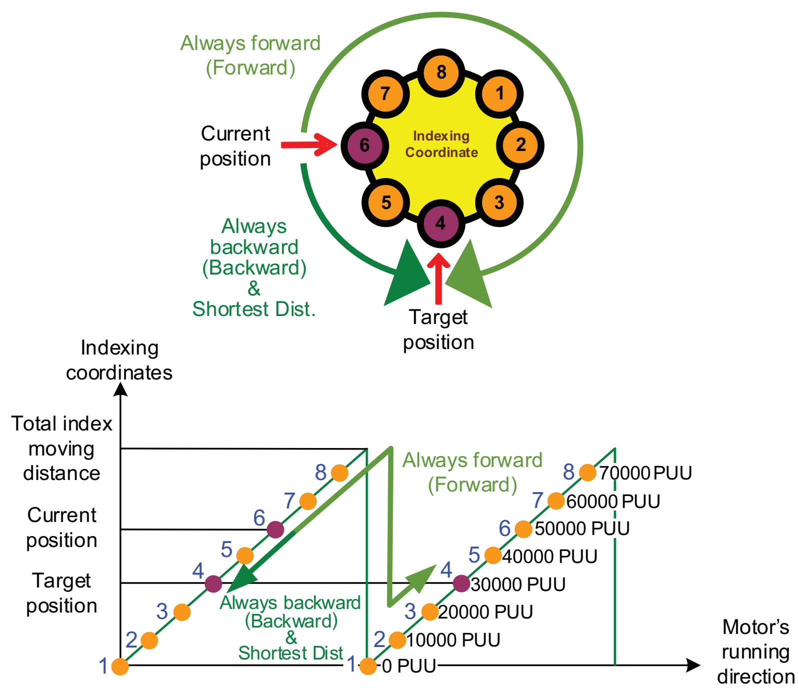


Figure 7-24 Motor's operation direction and indexing coordinates

Table 7-6 Bit function for the Index Position command

BIT / PR Parameters	D	C	B	A	U	Z	Y	X
Command Type	-	OPT2	DLY	SPD	DEC	ACC	OPT	0xA
Data Content	Index Position command [PUU](0 – P2.052-1)							

Notes:

1) Y: OPT: option

<table>
<tr><th>BIT</th><th>3</th><th>2</th><th>1</th><th>0</th><th>Description</th></tr>
<tr><td>Command Type</td><td colspan="2">DIR</td><td>OVLP</td><td>INS</td><td>-</td></tr>
<tr><td rowspan="4">Data Content</td><td>0</td><td>0</td><td rowspan="4">-</td><td rowspan="4">-</td><td>Always goes forward (Forward)</td></tr>
<tr><td>0</td><td>1</td><td>Always goes backward (Backward)</td></tr>
<tr><td>1</td><td>0</td><td>Shortest distance</td></tr>
<tr><td>1</td><td>1</td><td>-</td></tr>
</table>

INS: Interrupt command interrupts the previous motion command.
OVLP: allow overlapping of the next command

2) C: OPT2: Option 2

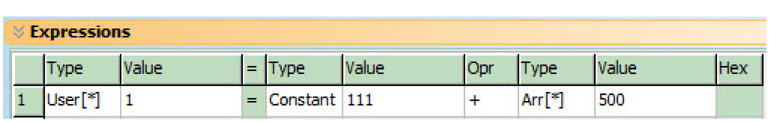
BIT	3	2	1	0
Command Type	-	AUTO	-	S_LOW

S_LOW: speed unit options, 0 stands for 0.1 rpm and 1 for 0.01 rpm.
AUTO: automatically load the next PR command when the current one is completed.

3) Z, U: ACC / DEC: acceleration / deceleration time set by P5.020 – P5.035.
4) A: SPD: delay time, set by P5.060 – P5.075.
5) B: DLY: delay time, set by P5.040 – P5.055.

Statement (Arithmetic operations)

PR mode has arithmetic operations commands, including addition, subtraction, multiplication, division, OR, AND, MOD (obtain remainder), and logic conditions. The available operands are user variables, parameters, data arrays, monitoring variables, and constants. Among them, the user variable is the register only for arithmetic operations. There are 64 sets of user variables with a data size of 32-bits. The data size of a constant is also 32-bits. The user variable index number is selected in the Value column [3] of the Expressions block. This is not the data value, only the index number. In order to assign a data value to a User[*] index you must write an expression (equation) to populate it. Below is an expression that assigns a value of 111 plus the data value stored in array element 500 into User Index 1. When the PR path containing this expression is executed the values of 111+Array Element 500 data will be populated into User[1].



Expressions

	Type	Value	=	Type	Value	Opr	Type	Value	Hex
1	User[*]	1	=	Constant	111	+	Arr[*]	500	

After all arithmetic operation commands are executed, you can set a jump condition in the path so that execution jumps to different PR path and then continue or stop once the operation is done. A Statement Type in a PR path can execute expressions and procedures in the same PR execution cycle with the Expression being evaluated before the Procedure, or the PR path can execute either just an Expression or a Procedure. Multiple Expressions can be entered but only one Procedure (condition). The Statement section is the result of the equations entered in the Expression and the Procedure section.

The arithmetic operation commands support negative numbers operations but not floating point operations. Negative numbers are calculated by “two’s compliment”. Figure 7-25 is the Arithmetic Operations screen in SureServo2 Pro. **Expressions, Procedures, and Statements must be created in SureServo2 Pro. Do not use the keypad or communication channels to enter arithmetic operations.** Once you complete the arithmetic operation, click **Download All PR** to write all PR paths to the servo drive.

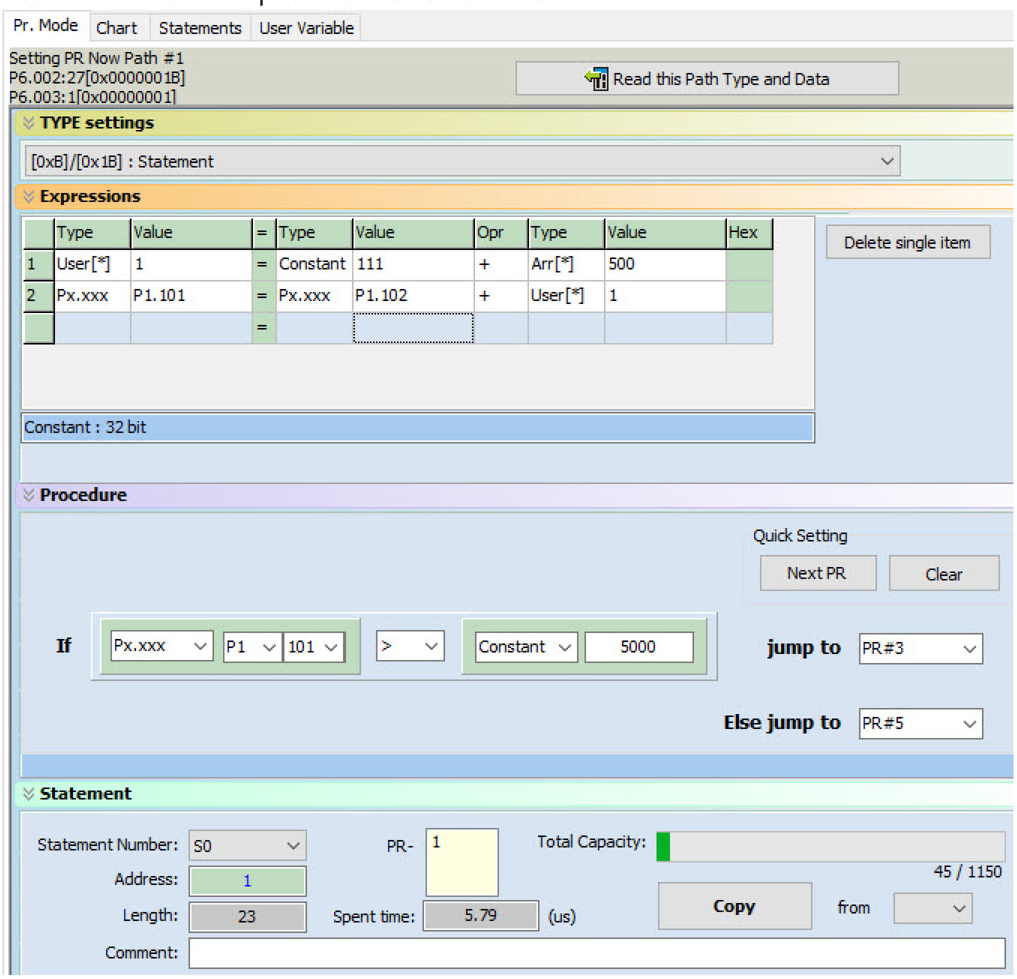


Figure 7-25 PR Arithmetic Operations screen in SureServo2 Pro

Wiring
Parameters
DI/DO Codes
Monitoring
Alarms

1) Expressions Section: supports addition, subtraction, multiplication, division, AND, OR, and MOD operation as well as logical operations for multiple data. Table 7-7 shows the supported operators and calculation data with data format in DEC and HEX.
   After all expressions have been evaluated the Statement Path now evaluates the Procedure condition.

Table 7-7 Description of each field in the Expression section

Data to be Written	=	Calculation Data	Operator	Calculation Data
User variable (User[0-63])		User variable (User[0-63])	Addition (+) Subtraction (-) Multiplication (*) Division (/) Modulus (%) And (&) Or (|)	User variable (User[0-63])
		Constant (Constant)		
Parameter (PX.XXX)		Data array (Arr[0-799])		Constant (Constant)
Data array (Arr[0-799])		Parameter (PX.XXX)		Data array (Arr[0-799])
		Monitoring variable (Mon[*])		

2) Procedure Section: uses the IF statement to determine whether the user-defined condition is fulfilled. If true, jump to the next specified PR path; if false, jump to the other specified PR path. If you click **Next PR** in **Quick Setting**, the software automatically inputs the condition and then jumps to the next PR path. If you leave this section blank, then the PR procedure will stop once the basic operation is done. See Table 7-8 for data formats and operators.

Table 7-8 Field description for the Procedure setting section

Data Format	Operator	Data Format
User variable (User[0-63])	Greater than (>) Greater than or equal to(≥) Less than (<) Less than or equal to (≤) Equal to (=) Not equal to (≠)	User variable (User[0-63])
Constant (Constant)		
Data array (Arr[0-799])		Constant (Constant)
Parameter (PX.XXX)		Data array (Arr[0-799])
Monitoring variable (Mon[*])		

3) Statement Section: this section includes statements and memory capacity. Statements save the data from the expression and procedure sections. Data in the expression and procedure sections of the same statement always remain identical and can be shared by multiple PR paths. If data in those two sections are different, then the data is saved to another statement. The time required to execute the statement is shown in the **Spend Time** field. **Total Capacity** shows the servo drive memory capacity; basic operations cannot be performed if there is no memory space available. The **Statements** tab is shown in Figure 7-26. The upper section displays all the statements and the lower section displays the operations in each statement and the values.

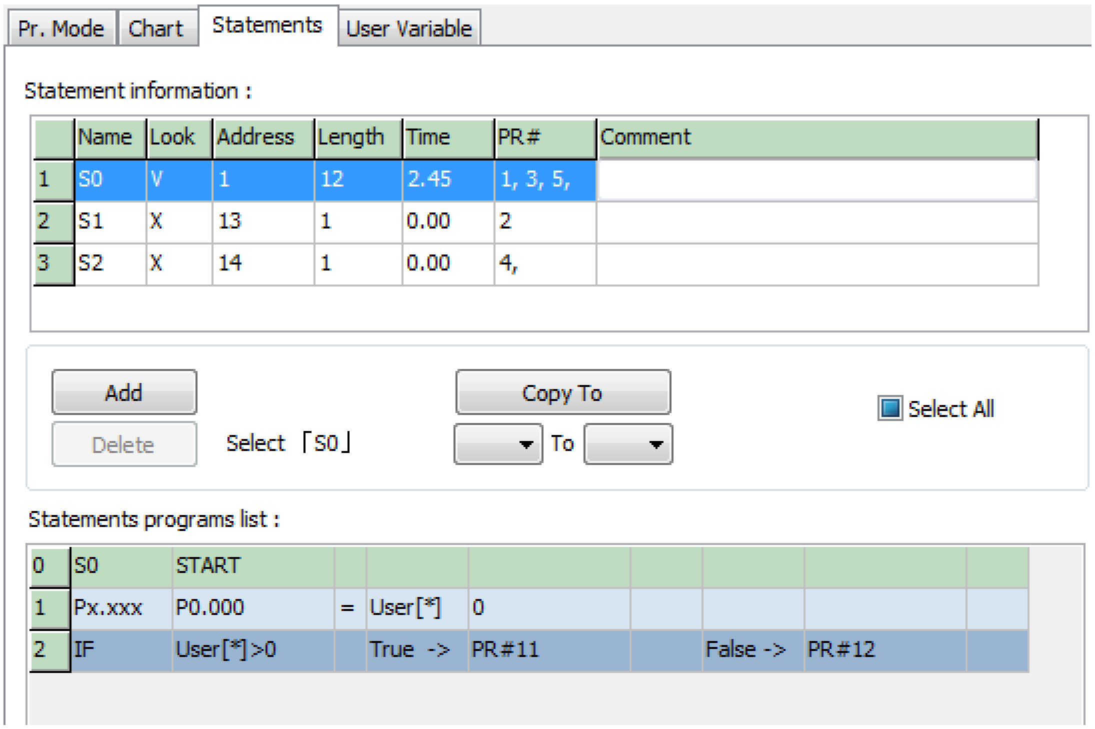


Figure 7-26 Statements tab in SureServo2 Pro

7.1.4 - Overview of the PR procedure

In PR mode, there are seven types of commands. To understand how the PR procedure works, SureServo2 Pro presents the execution order and calling sequence of all PR procedures. First, symbols and contents in the PR figure are shown. This includes five parts: PR path number, execution type, command type, next PR command, and command data. See Figure 7-27.

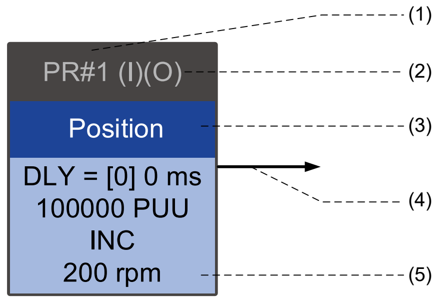


Figure 7-27 PR display

1) Number: the PR number, ranging from PR#0 to PR#99 (100 sets of PR paths).
2) Command execution (property): (B) Execute homing when power on; (O) Command overlap; (R) write data to EEPROM; (I) command interrupt.
3) Command type: there are six types of PR procedure commands: homing, speed, position, writing, jumping, and arithmetic operations. The color displayed in this section depends on the command type.
4) Next procedure command: if followed by a PR command, the arrow points to the specified PR path.
5) Command information: displays the details of this PR path. The color depends on the information types.

The following sections illustrate each command type and its presentation.

Homing methods

In the display of homing methods, PR#0 is always the homing procedure, which is marked as "Homing". See Figure 7-28.

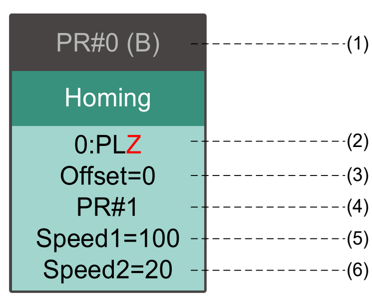


Figure 7-28 Homing methods display

1) Activation mode (Boot): to execute homing when the drive is in Servo On state, it displays (B); if homing is not required, then no information is displayed.

2) Method Selection: homing methods and Z pulse setting are shown in the table below. F signifies running forward; R signifies running in reverse; ORG signifies origin; CUR signifies current position; BUMP represents the collision point.

Homing methods	***Y = 0: reverse to look for Z pulse Y = 1: go forward to look for Z pulse***	***Y = 2: do not look for Z pulse***
X = 0: homing in forward direction with PL as the homing origin	0: PLZ	0: PL
X = 1: homing in reverse direction with NL as the homing origin	1: NLZ	1: NL
X = 2: homing in forward direction with ORG (when it switches from off to on state) as the homing origin	2: F_ORGZ	2: F_ORG
X = 3: homing in reverse direction with ORG (when it switches from off to on state) as the homing origin	3: R_ORGZ	3: R_ORG
X = 4: look for the Z pulse in forward direction with it as the homing origin	4: F_Z	
X = 5: look for the Z pulse in reverse direction with it as the homing origin	5: R_Z	
X = 6: homing in forward direction with ORG (when it switches from on to off state) as the homing origin	6: F_ORGZ	6: F_ORG
X = 7: homing in reverse direction with ORG (when it switches from on to off state) as the homing origin	7: R_ORGZ	7: R_ORG
X = 8: use the current point as the origin	8: CUR	
X = 9: look for collision point in forward direction and use it as the origin	9: F_BUMPZ	9: F_BUMP
X = A: look for collision point in reverse direction and use it as the origin	A: R_BUMPZ	A: R_BUMP

3) Offset: origin offset, P6.001
4) Path: next PR path to be executed after homing
5) Homing at high speed: first homing speed, P5.005.
6) Homing at low speed: second homing speed, P5.006.

Speed command

You can use the Speed command in any PR paths (PR#1 – PR#99). It is marked as “Speed”. See Figure 7-29.

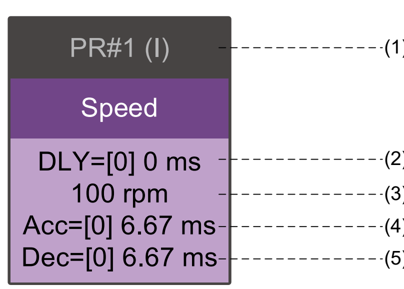


Figure 7-29 Speed command display

1) Command execution type: a Speed command can interrupt (INS) the previous PR path. If the Interrupt function is enabled, it displays (I); if not, no information is displayed.
2) Delay time (DLY): determined by shared PR parameters. It is defined by a command from the controller; the servo drive starts counting the delay time once it reaches the target speed.
3) Target speed: the set target speed.
4) Acceleration time (ACC): determined by shared PR parameters; length of time to reach the target speed from stopped.
5) Deceleration time (DEC): determined by shared PR parameters; length of time to decelerate from target speed to stopped.

Position command

You can use the Position command in any PR paths (PR#1 – PR#99). It is marked as "Position", and includes the options to "Stop once position control completed" and "Load the next path once position control completed". The only difference is that "Load the next path once position control completed" shows an arrow pointing to the next PR. See Figure 7-30.

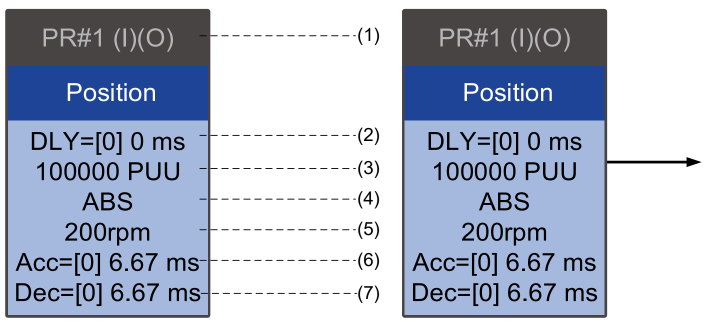


Figure 7-30 Position command display

1) Command execution type: a Position command can interrupt (INS) the previous PR path. If the Interrupt function is enabled, it displays (I); if not, no information is displayed. The Position command can overlap (OVLP) the next PR path. If delay time is set to 0 when this function is enabled, it displays (O). If the Overlap function is not used, no information is displayed.
2) Delay time (DLY): determined by shared PR parameters. It is defined by a command from the controller. The servo drive starts counting the delay time once it reaches the target position.
3) Target position: the set target position.
4) Position command type: "ABS" means an absolute positioning command ; "REL" means relative positioning; "INC" means incremental positioning; "CAP" means high speed position capture.
5) Target speed: determined by shared PR parameters.
6) Acceleration time (ACC): determined by shared PR parameters; the length of time to reach the target speed from stopped.
7) Deceleration time (DEC): determined by shared PR parameters; the length of time to decelerate from target speed to stopped.

Jump command

You can use the Jump command in any PR paths (PR#1–PR#99). It is marked as "Jump" and followed by an arrow pointing to the next PR path. See Figure 7-31.

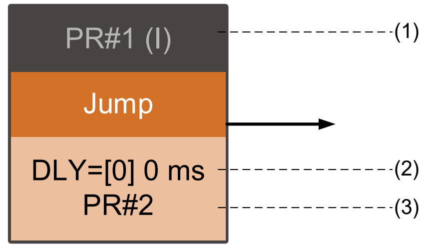


Figure 7-31 Jump command display

1) Command execution type: the Jump command can interrupt (INS) the previous PR path. If the Interrupt function is enabled, it displays (I); if not, no information is displayed.
2) Delay time (DLY): determined by shared PR parameters.
3) Target PR number: the target PR number.

Write command

You can use the Write command in any PR paths (PR#1 – PR#99). It is marked as "Write". See Figure 7-32.

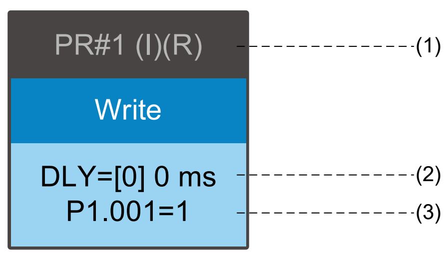


Figure 7-32 Write command display

1) Command execution type: a write command can interrupt (INS) the previous PR path. If the Interrupt function is enabled, it displays (I); if not, no information is displayed. You can determine whether to write the data to EEPROM. If writing data to EEPROM is required, it shows (R); if not, no information is displayed.
2) Delay time (DLY): determined by shared PR parameters.
3) Writing target and data source: the corresponding target and data sources are shown in the table below. Please note that constants can be written in DEC or HEX format.

Writing Target	***Data Source***
Parameter (PX.XXX)	Constant
Data array (Arr[#])	Parameter (PX.XXX)
-	Data array (Arr[#])
-	Monitoring variable (Mon[#])

Index position command

You can use the Indexing Position command in any PR paths (PR#1–PR#99). The number of PR paths is determined by the index number. It is marked as “Index Position”. See Figure 7-33.

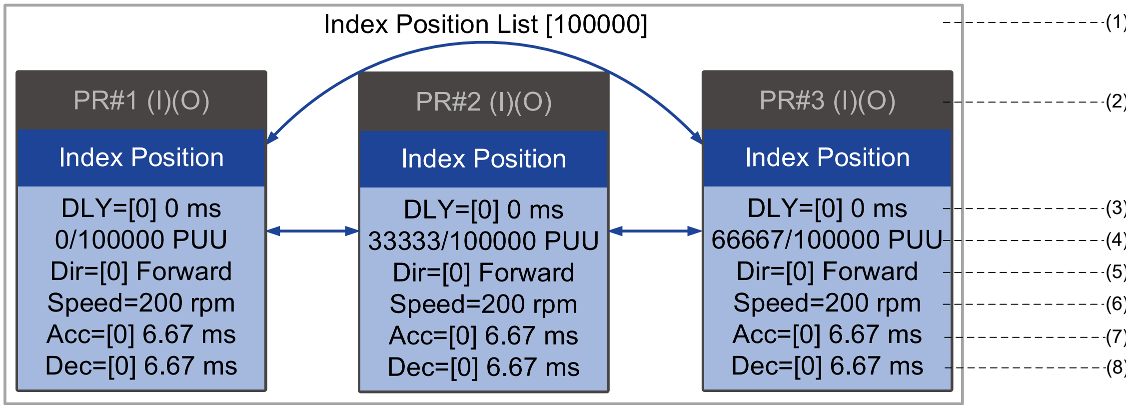


Figure 7-33 Indexing Position command display

1) Indexing Position command section: the number of the index position. It shows the total moving distance at the top using double arrows to show that the motor can run reciprocally between each target position in each PR path.
2) Command execution type: an Index Position command can interrupt (INS) the previous PR path. If the Interrupt function is enabled, it displays (I); if not, no information is displayed. The Index Position command can overlap (OVLP) the next PR path. If delay time is set to 0 when this function is enabled, it displays (O). If the Overlap function is not used, no information is displayed.
3) Delay time (DLY): determined by shared PR parameters. It is defined by a command from the controller. The servo drive starts counting the delay time once it reaches the target position.
4) Position command: the numerator is the position of this PR path; the denominator is the total moving distance of this indexing Position command, which is set by P2.052.
5) Rotation direction (Dir): available options are “Rotation forward (Forward)”, “Rotation in reverse (Reverse)” and “Rotation with the shortest distance (Shortest)”.
6) Target speed: determined by shared PR parameters.
7) Acceleration time (ACC): determined by shared PR parameters; the length of time to reach the target speed from stopped.
8) Deceleration time (DEC): determined by shared PR parameters; the length of time to decelerate from target speed to stopped.

Statement (Arithmetic operation)

You can use arithmetic operations and statements in any PR paths (PR#1 – PR#99). It is marked as “Statement”. When the condition is fulfilled, an arrow pointing to the next PR path appears with a solid line; if the condition is unfulfilled, an arrow pointing to the next PR appears with a dotted line; Or you can choose to execute the next PR path and stop once the execution is completed. See Figure 7-34.

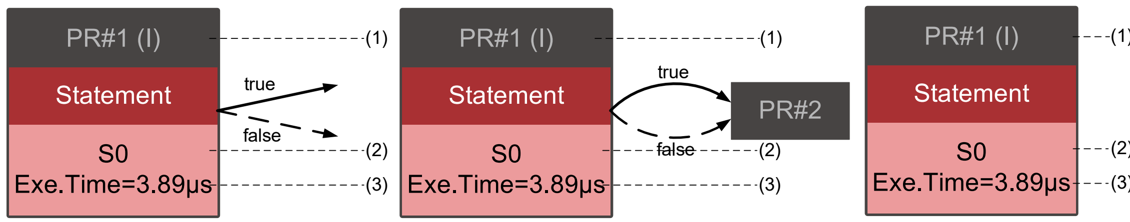


Figure 7-34 Arithmetic operation display

1) Command execution type: an arithmetic operation command can interrupt (INS) the previous PR path. If the Interrupt function is enabled, it displays (I); if not, no information is displayed.
2) Statement number: displays the statement number used in the PR path.
3) Execution time (Exe. Time): the time required to execute the arithmetic operation.

7.1.5 - Trigger methods for the PR command

There are seven types of PR triggering methods. They are DI-triggered, Event-triggered, P5.007-triggered (for Modbus/ModTCP and EtherNet/IP Explicit communication), P5.122-triggered (for EtherNet/IP *Implicit* communications), Capture-triggered (high-speed position capturing), Compare-triggered (high-speed position comparing), and E-Cam-triggered. You can choose the most suitable triggering method according to the applications and requirements.

Digital input (DI) triggering

You can choose the PR path to be executed by using the internal registers (Position command Bit0 – Bit6) and use a command to trigger the selected PR path. Bits 0–6 are used to create a binary number that calls the PR path. Before using DI-triggering commands, you must define the 8 sets of DI functions, which are [0x11]POS0, [0x12]POS1, [0x13]POS2, [0x1A]POS3, [0x1B]POS4, [0x1C]POS5, [0x1E]POS6, and [0x08]CTRG (refer to section 8.4.9). You can also set this in the Digital IO / Jog Control window of SureServo2 Pro, as shown in Figure 7-35.

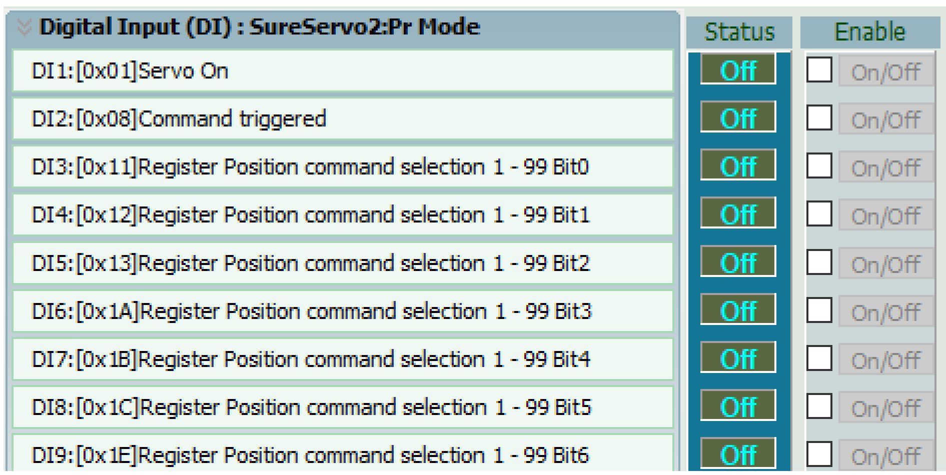


Figure 7-35 I/O screen in SureServo2 Pro

Select the PR number to be executed based on the on / off status of DI.POS0–6 and use DI.CTRG to trigger the specified PR path. See Figure 7-35 for an example.

Table 7-9 Use DI to select the PR path to be triggered

Position Command	POS 6	POS 5	POS 4	POS 3	POS 2	POS 1	POS 0	CTRG	Parameter
Homing	0	0	0	0	0	0	0	↑	P6.000 P6.001
PR#1	0	0	0	0	0	0	1	↑	P6.002 P6.003
~									
PR#50	0	1	1	0	0	1	0	↑	P6.098 P6.099
PR#51	0	1	1	0	0	1	1	↑	P7.000 P7.001
~									
PR#99	1	1	0	0	0	1	1	↑	P7.098 P7.099

In addition, there are two sets of DI for special functions: [0x27] Enable Homing and [0x46] motor stop. If the former is triggered, the servo drive executes the homing routine based on the homing setting. If the latter is triggered, the servo drive stops the motor. You can use the I/O screen in SureServo2 Pro to set these functions, as shown in Figure 7-36.

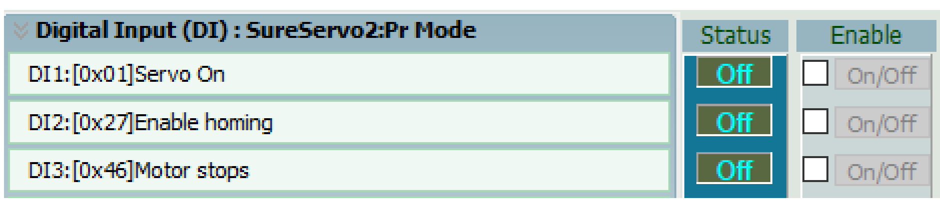


Figure 7-36 I/O screen in SureServo2 Pro

Wiring
Parameters
DI/DO Codes
Monitoring
Alarms

Wiring
Parameters
DI/DO Codes
Monitoring
Alarms

Event triggering

You can use Event-triggered commands 1 – 4 to execute the specified PR path. You can select two types of Event triggering: rising-edge trigger and falling-edge trigger. Parameters P5.098 (rising edge events) and P5.099 (falling edge events) are used to configure what events trigger what PR paths. The range of PR path numbers that you can specify is from 51 – 63 (see example in Figure 7-37). Before using the Event-trigger for PR command, you must define the DI functions, which are [0x39] Event-trigger command 1, [0x3A] Event-trigger command 2, [0x3B] Event-trigger command 3, and [0x3C] Event-trigger command 4 (see section 8.4.9). You can use SureServo2 Pro to set the I/O triggering as shown in Figure 7-38.

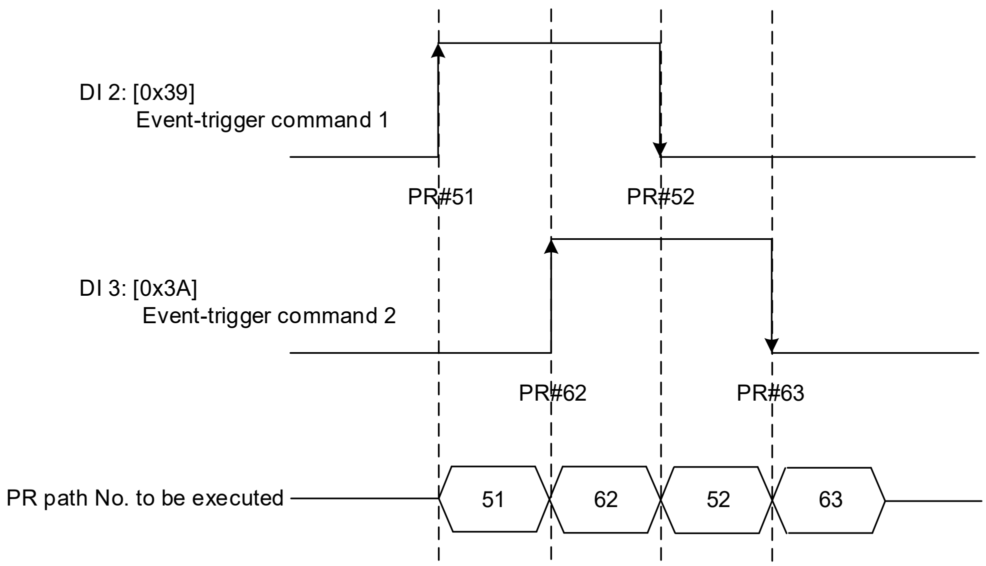


Figure 7-37 Example of Event triggering timing diagram

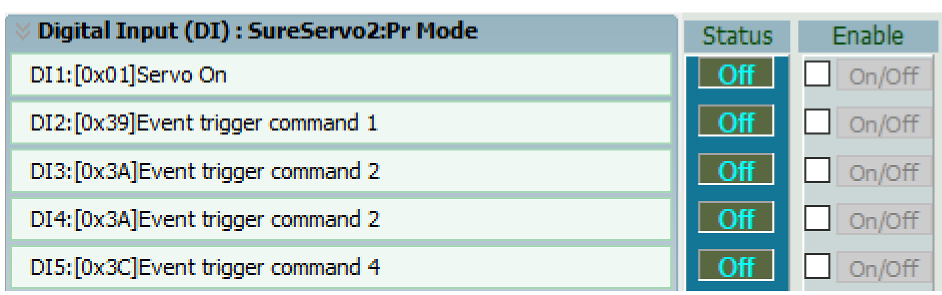


Figure 7-38 I/O screen in SureServo2 Pro

You can set the rising-edge trigger for the desired PR path with P5.098 while you can set the falling-edge trigger with P5.099. Please refer to Chapter 8 for more details. You can also set the Event trigger of PR in SureServo2 Pro (see Figure 7-39).

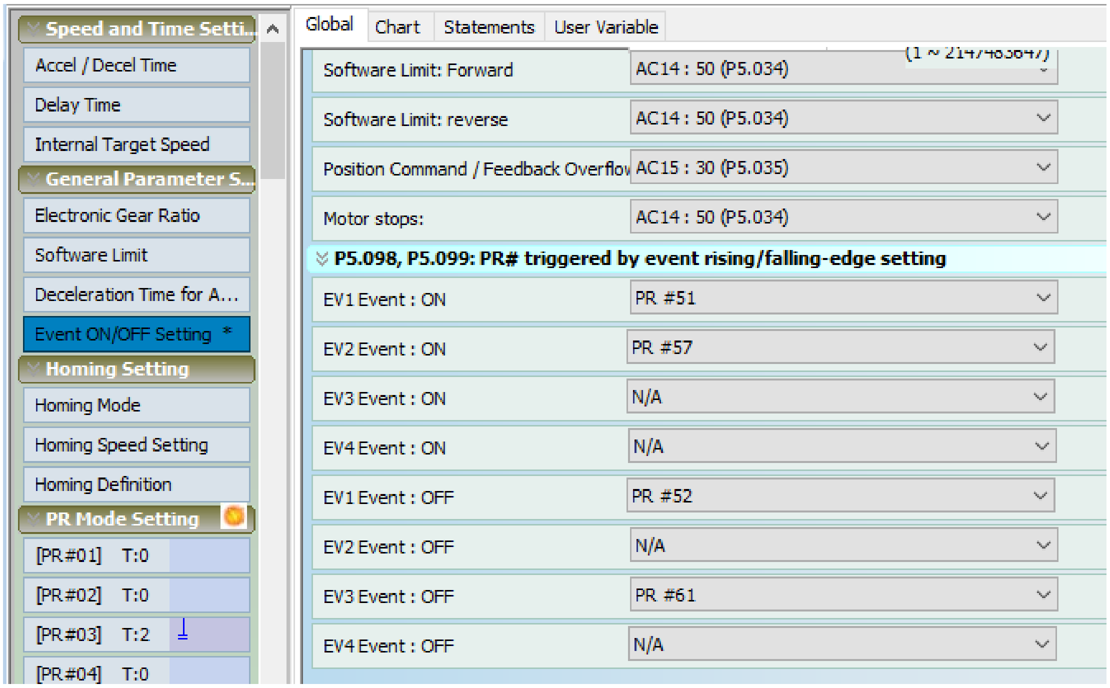


Figure 7-39 Event On/Off screen in SureServo2 Pro

PR command trigger register (P5.007)

You can write the PR number to be executed in P5.007 to make the servo drive execute the specified PR path. If you write 0 to the PR Command Trigger register, the servo drive executes homing. If you write 1 – 99 to the PR Command Trigger register, the servo drive executes the specified PR path. If you write 1000, the servo drive stops executing PR commands. You can find more information in the description of P5.007 in Chapter 8.

Special trigger method

You can use High-speed position capturing (Capture), High-speed position comparing (Compare), and the E-Cam function to trigger the specified PR path. When the capturing completes, you can set Bit3 of P5.039.X to trigger or not trigger PR#50, or set Bit12 of P5.059 to trigger or not trigger PR#45 once the last data is compared. If the E-Cam disengagement setting is 2, 4, or 6, use P5.088.BA to write the PR path number. Please refer to Section 7.2 for Capture, Compare, and E-Cam functions.

Triggering method	***Setting bit***	***Trigger PR path***
High-speed position capturing (Capture)	P5.039.X Bit3	PR#50
High-speed position comparing (Compare)	P5.059.U Bit0	PR#45
E-Cam	P5.088.BA	User-defined

7.1.6 - PR Procedure Execution Flow

The SureServo2 drive updates the command status every 1ms. Figure 7-40 illustrates the PR procedure execution flow and how the servo drive deals with PR commands. Once a PR procedure is triggered, it goes through three phases, which are PR queue, PR executor, and motion command generator.

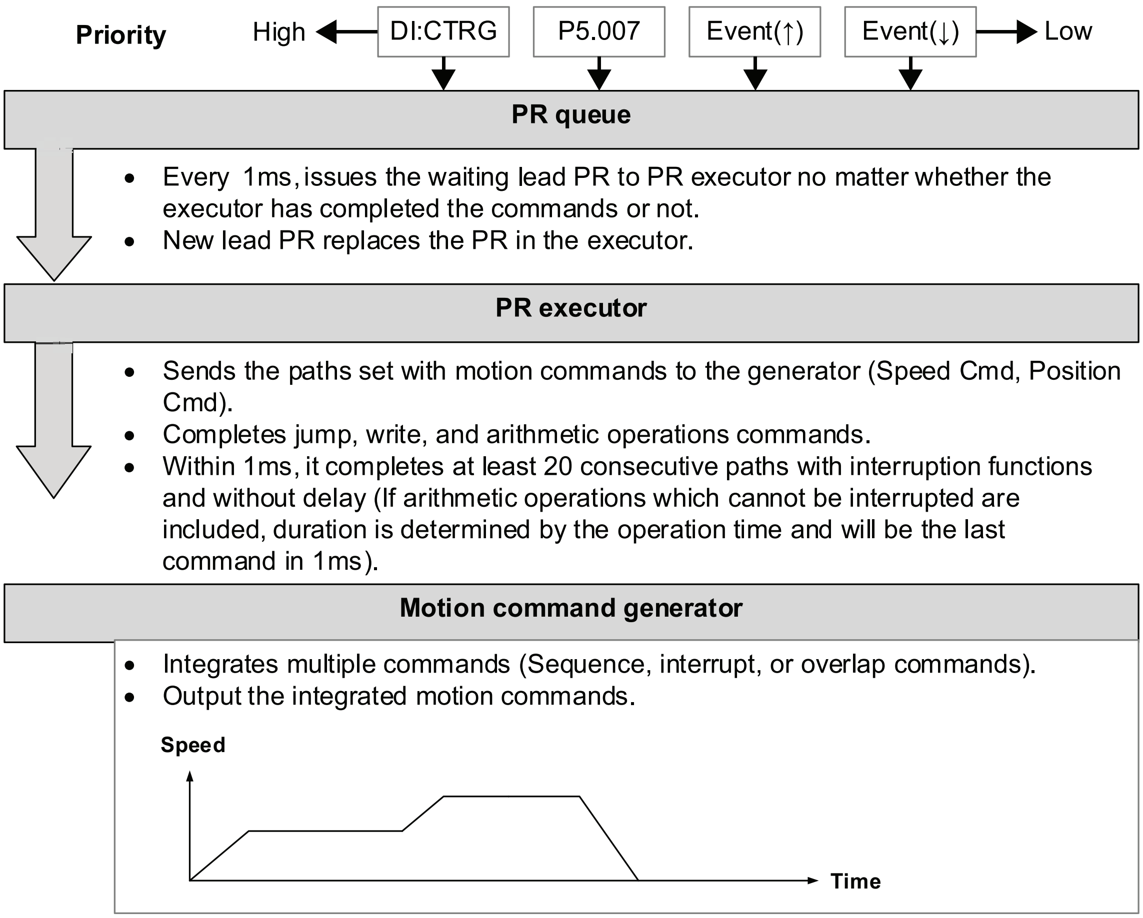


Figure 7-40 PR execution flow in the SureServo2 drive

- *Trigger mechanism*
  The trigger mechanism is as mentioned in Section 7.1.5. There are three trigger methods. A PR procedure is executed as long as a trigger signal is activated. When two different trigger methods are used for one PR procedure within the same millisecond, the priority is as follows: DI trigger (DI.CTRG) > PR command trigger register (P5.007) > Rising-edge event trigger (Event ↑) > Falling-edge event trigger (Event↓). Within this millisecond, commands with higher priority are executed first and then the lower priority commands are arranged in the next ms. If three trigger commands are generated in the same millisecond, the third is not added to the PR queue.

- *PR queue*
  The triggered PR path is the lead PR. The PR group it leads goes into the PR queue to wait for prioritization. In each millisecond, the servo drive sends the lead PR and the PR group it leads to the PR executor no matter whether a PR path is being executed. Therefore, as long as a PR path is triggered, the PR queue collects it and sends it to the executor.

- *PR executor*
  Once the PR executor receives the lead PR and its PR group, the PR group in execution is replaced immediately. If a PR group includes motion commands, such as Speed commands and Position commands, then the PR executor sends them to the motion command generator. PR paths with Write or Jump commands are completed at the moment when the PR executor reads the command, and thus they do not enter the motion command generator. The arithmetic operations commands are executed when entering the PR executor; however, the execution time varies with the computing duration and the next command cannot interrupt during computing. The PR executor can consecutively complete at least 20 PR paths with interrupt commands (INS) (without delay times) within 1ms. If there is a PR path that it has not completed within 1ms, and a new PR group is sent to the executor by the queue, the new PR group then replaces the previous PR group. In other words, instead of executing the PR group that hasn't been completed, the executor starts executing the new PR group. However, if a new PR group hasn't been sent to the executor yet, the executor continues to execute the unfinished PR path.

- *Motion command generator*
  Motion commands include the Speed and Position commands. The PR executor sends this type of command to the motion command generator. This generator has a buffer for temporarily storing the next motion command and all motion commands are integrated here. Motion commands can be executed as soon as they enter the generator. If another motion command (with interrupt setting) also enters the generator, it is integrated with the current command in the generator and the integration is based on the motion command settings. The settings include whether multiple motion commands are sequence commands, and whether it is set with the Overlap or Interrupt function. All integration varies with each PR path setting.

Sequence command

The configurable commands in PR path are the motion commands, which are the Position and Speed commands. A sequence command is a motion command without an Overlap or Interrupt function. The following command start to be executed only after the delay set in the previous command. Regarding Position commands, the delay time starts to count after the target position is reached. For Speed command, the delay time counting starts after the target speed is reached.

- *Position command followed by a Position command*
  When the PR executor receives two consecutive Position commands, if they do not have Interrupt or Overlap functions, the PR executor issues the first Position command to the motion command generator, and the generator starts the first part of position control. After the first Position command completes, if no delay time is set, the PR executor issues the second Position command for the generator to start the second part of position control (see Figure 7-41).
  If the first Position command includes a delay, the PR executor starts counting the delay time right after the motor reaches the target position. Then it issues the second Position command to the generator for the second part of position control as shown in Figure 7-42).

Wiring
Parameters
DI/DO Codes
Monitoring
Alarms

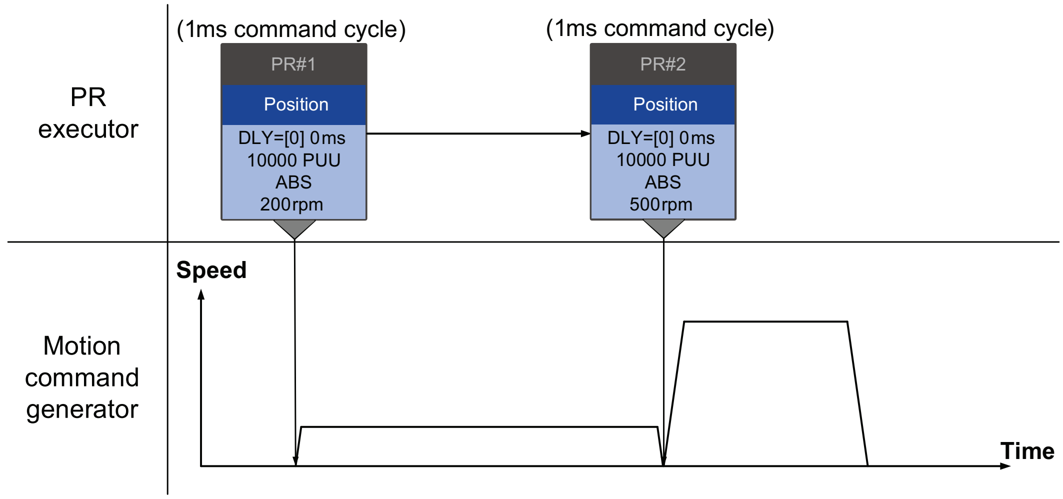


Figure 7-41 Position command without delay

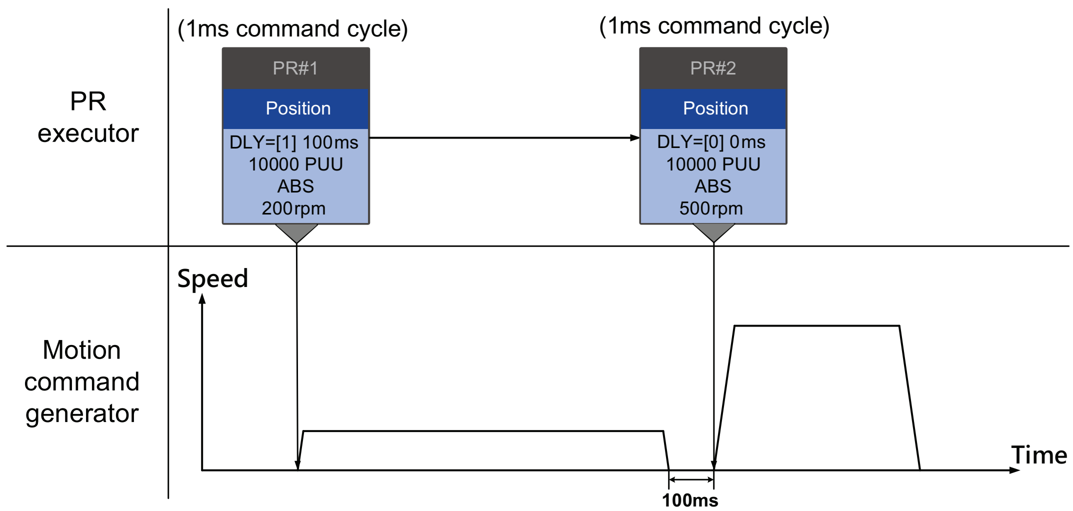


Figure 7-42 Position command with delay

- *Speed command followed by a Speed command*
  When the PR executor receives two consecutive Speed commands, if they do not have Interrupt or Overlap functions, the PR executor issues the first Speed command to the motion command generator, and the generator starts the first part of speed control. After the first Speed command completes, if no delay time is set, the PR executor issues the second Speed command to the generator to start the second part of speed control (see Figure 7-43).
  If the first Speed command includes a delay, the PR executor starts counting the delay time right after the motor reaches the target speed. Then it issues the second Speed command to the generator for the second part of speed control as shown in Figure 7-44.

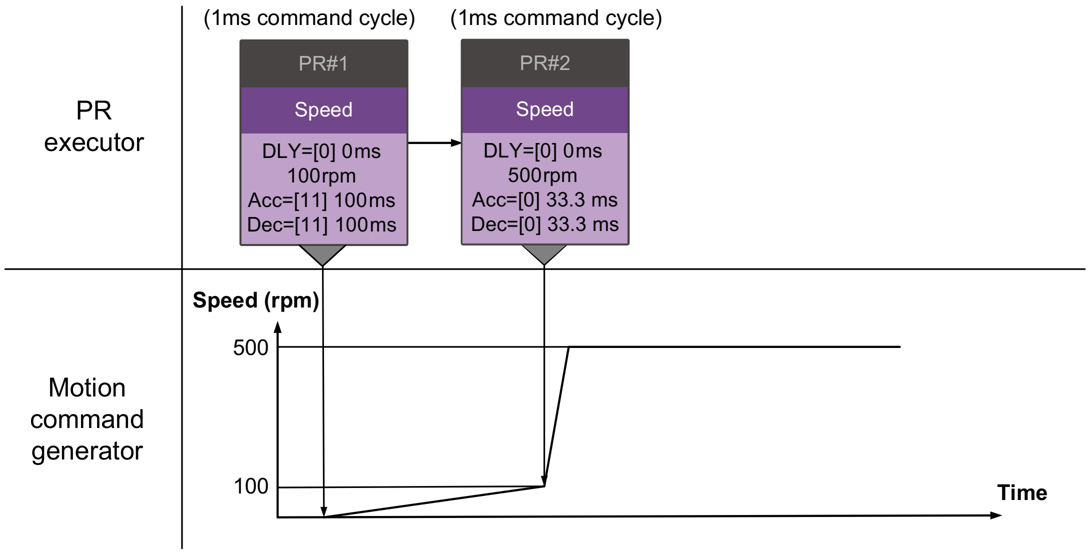


Figure 7-43 Speed command without delay

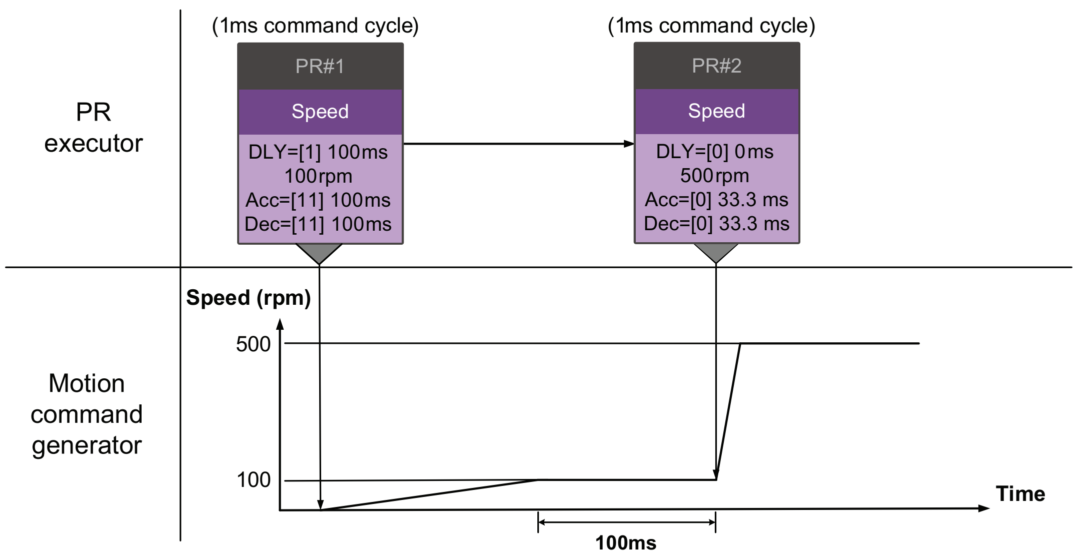


Figure 7-44 Speed command with delay

- *Multiple commands*
  The PR queue updates commands every millisecond. For a motion command, the PR queue sends the next command to the generator only after the previous command completes. Jump or Write commands are executed in the PR queue immediately.
  As shown in Figure 7-45, in the first millisecond, the PR queue receives a Position command and it sends this command to the motion command generator, causing the generator to execute the command. In the second millisecond, the PR queue receives a Write command and executes it immediately. In the third millisecond, the PR queue receives a Jump command and executes it immediately as well. The Write and Jump commands are not sent to the motion command generator since the PR queue and the generator can execute commands independently. In the fourth millisecond, the PR queue receives a Position command. After the first Position command is completed, the PR executor sends it to the generator and the generator starts executing it immediately.

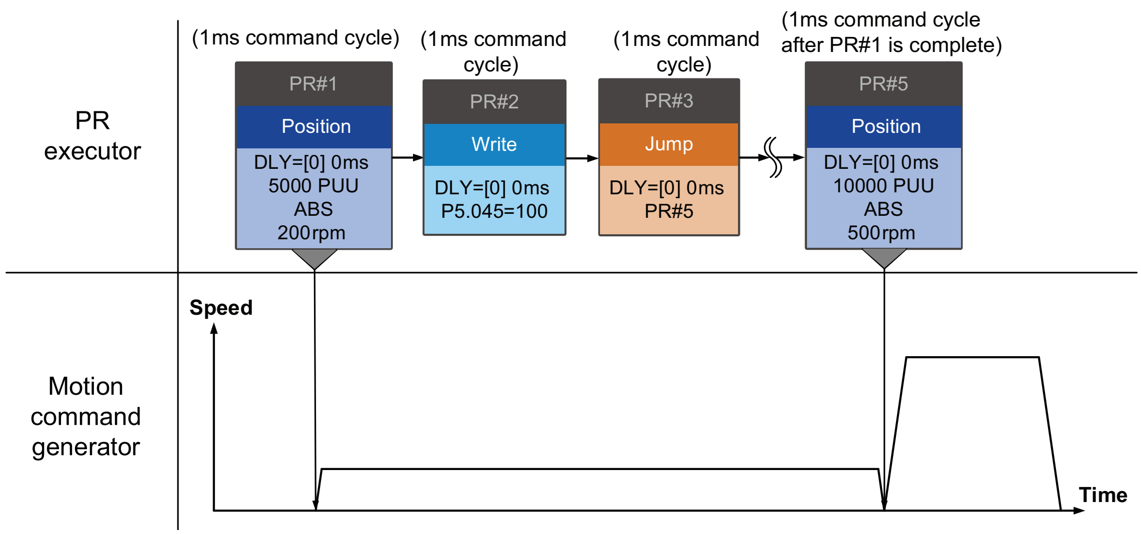


Figure 7-45 Sequence command – Multiple commands

Command interruption

Interruption (INS) causes a command in execution to be replaced or integrated. The results of the interruption differ based on the command types. The next command replaces the previous command. There are two types of interruption: internal and external, as shown in Figure 7-46.

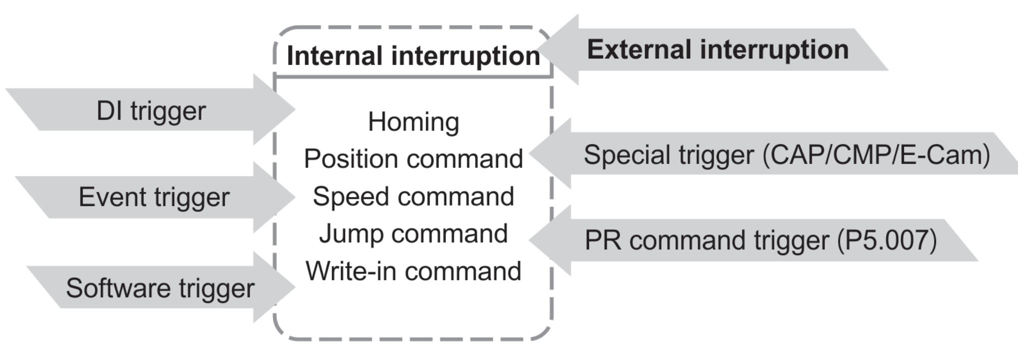


Figure 7-46 Internal and external interruption

Internal Interruption

For a series of PR paths, if one PR path includes an AUTO function (auto-execute the next path), the system reads the next path after reading the current path. If the current path includes a delay, the next path is read after delay time is over. Meanwhile, if the next path includes an Interrupt function (which has a higher execution priority) the servo drive immediately executes the interrupt command. It replaces the un-executed part in the previous path with the next or integrates the commands in the execution of the previous path.

- *Position command ► Position command (I) ► Position command*
  When the PR executor receives three consecutive Position commands with an interrupt in the second command, the executor treats the first and the second Position commands as one PR group. Since the first Position command is not executed, the executor replaces the first command with the second. It only sends the second command to the motion command generator for execution. After the second command is completed, the executor sends the third command to the generator (see Figure 7-47).
  If the first command includes a delay, then the PR executor sends the first command to the generator and then starts counting the delay time. After the delay is over, the PR executor then sends the second command and the generator starts the second part of position control. While the first command is still being executed, it is integrated with the second command. The integration is slightly different from what is described in Section 7.1.3. Please refer to the note below. Once the second command is completed, the executor sends the third command to the generator for execution (see Figure 7-48).

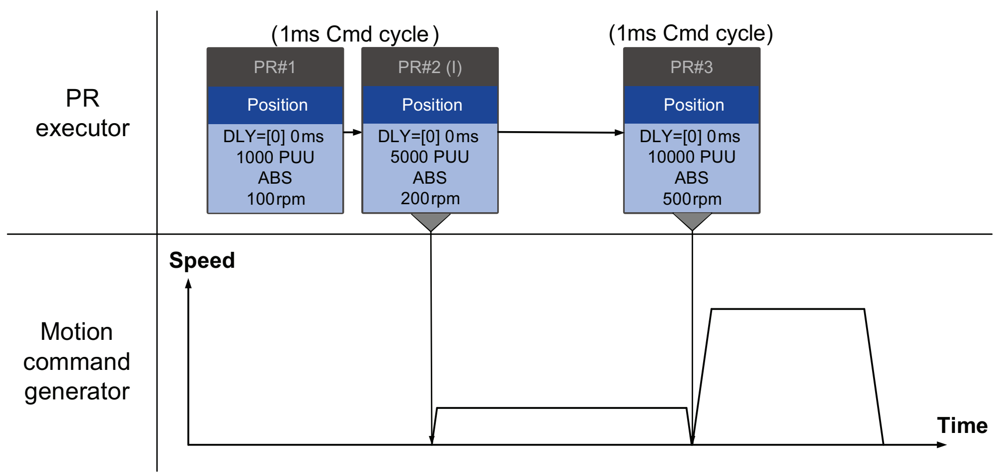


Figure 7-47 Position command without delay

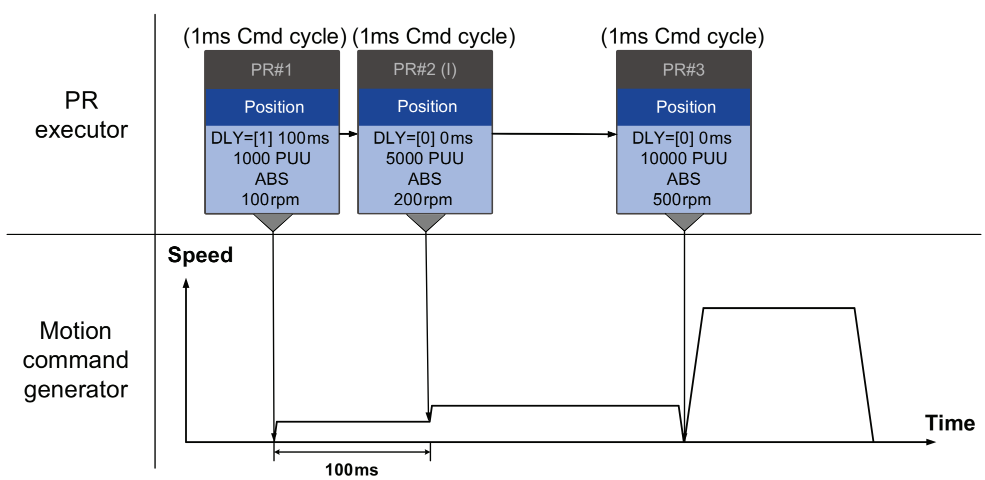


Figure 7-48 Position command with delay

NOTE: *The integration for internal interrupt position command is slightly different from what is described in Section 7.1.3. The way REL and INC commands work is the identical. The target position is the previous target position plus the current position. See the example below. The rest of the integration method is the same as mentioned in Section 7.1.3.*

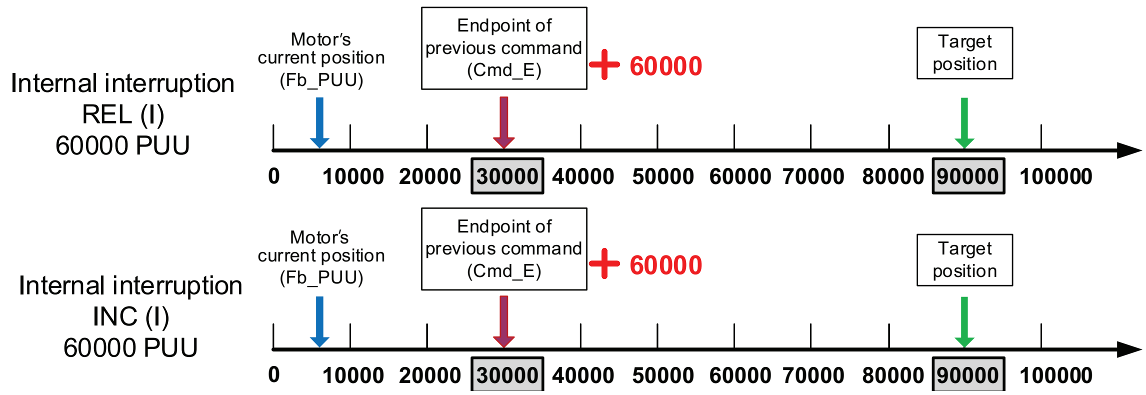


Figure 7-49 Example of relative and incremental position command for internal interruption

- *Speed command ► Speed command (I) ► Speed command*
  When the PR executor receives three consecutive Speed commands with an interrupt in the second command, the executor treats the first and the second as one PR group. Since the first Speed command is not executed, the executor replaces the first command with the second. It only sends the second command to the motion command generator for execution. After the second command is completed, the executor sends the third command to the generator (see Figure 7-50).
  If the first command includes a delay, then the PR executor sends the first command to the generator and then starts counting the delay time. After the delay is over, it then sends the second command and the generator starts the second part of speed control. While the first command is still being executed, it is integrated with the second command. Once the second command is completed, the executor sends the third to the generator for execution (see Figure 7-51).

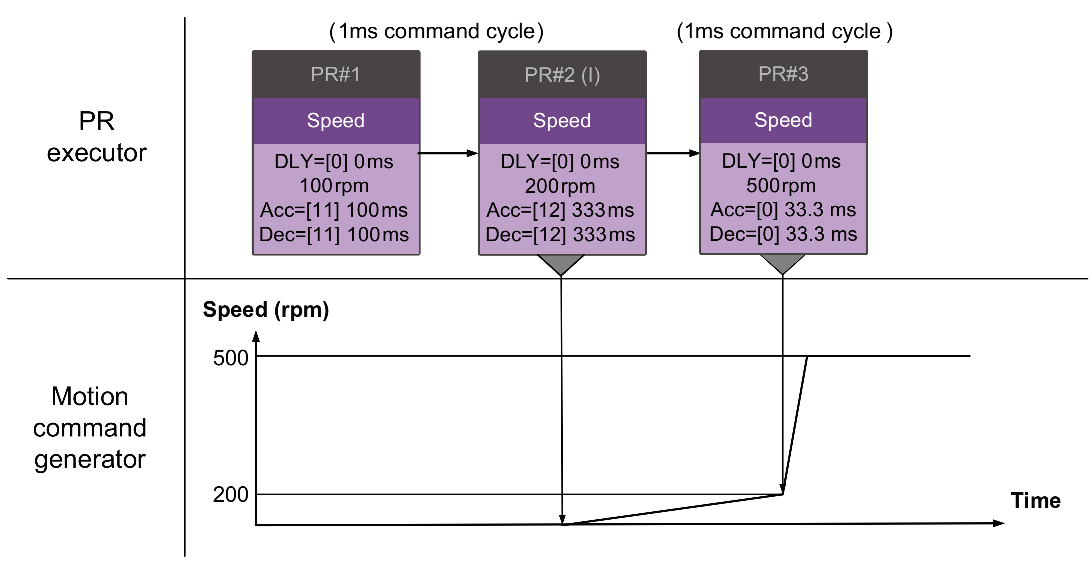


Figure 7-50 Speed command without delay

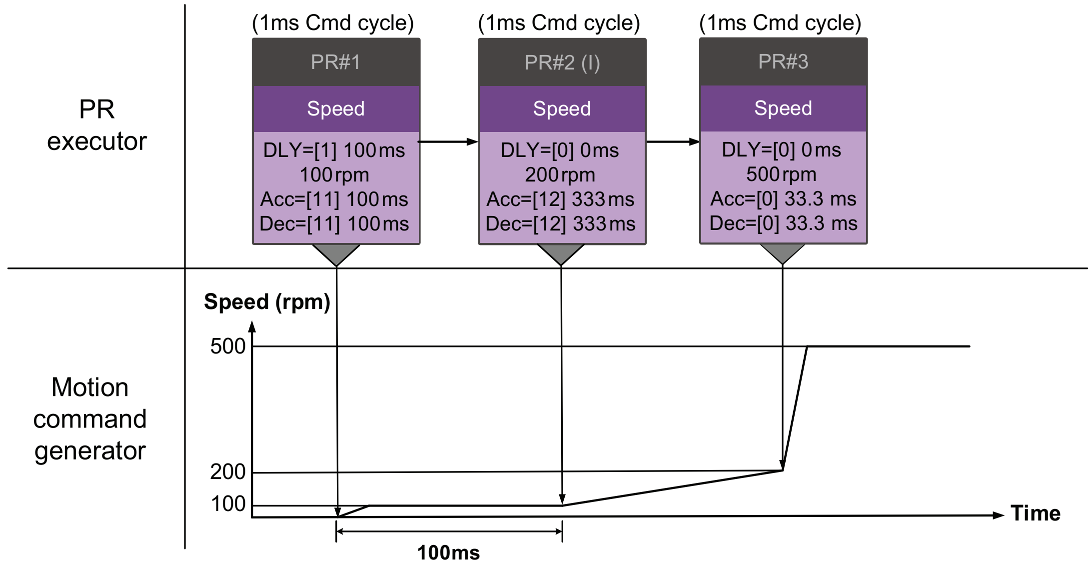


Figure 7-51 Speed command with delay

- *Multiple interrupt commands*
  The PR queue updates once every 1ms. If all PR paths include an Interrupt function, the queue can read at least 20 PR paths in 1ms, and these paths are called a PR group.
  If this PR group has multiple motion commands, the PR queue only sends the last command it receives to the motion command generator for execution. Therefore, in a PR group, only one PR path with motion command is executed. The latter motion command directly replaces the former, whereas Jump and Write commands are executed as soon as they are received by the PR queue (see Figure 7-52).
  If one of the PR paths includes a delay, the PR queue schedules all paths on the basis of this PR path. The prior path(s) including a delay becomes as the first PR group, and what follows is the second PR group. Thus, this PR procedure can execute up to two PR paths with motion commands, as shown in Figure 7-53.

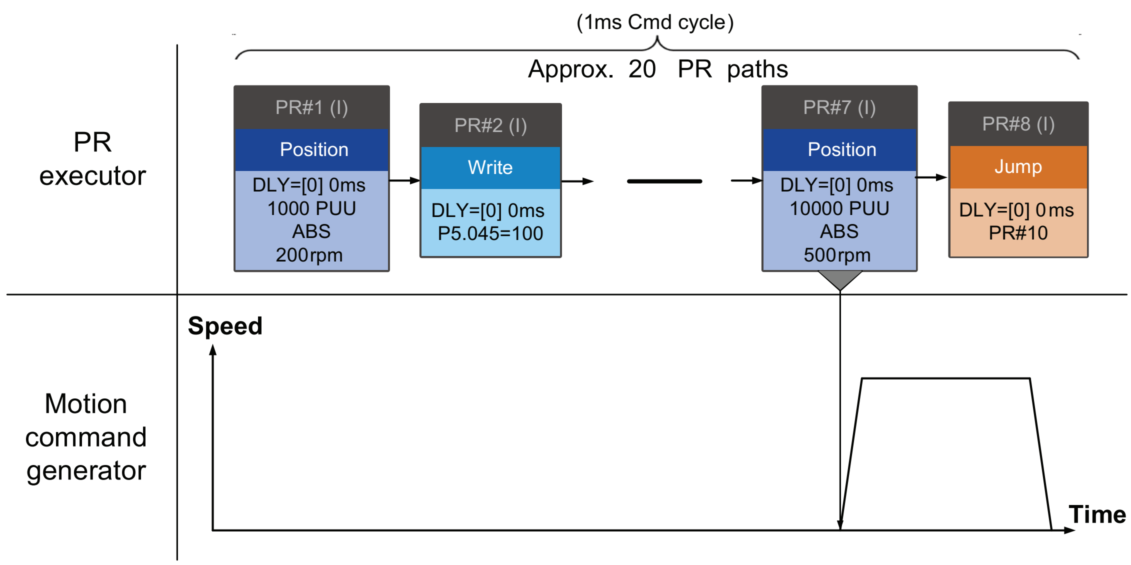


Figure 7-52 Multiple commands without delay

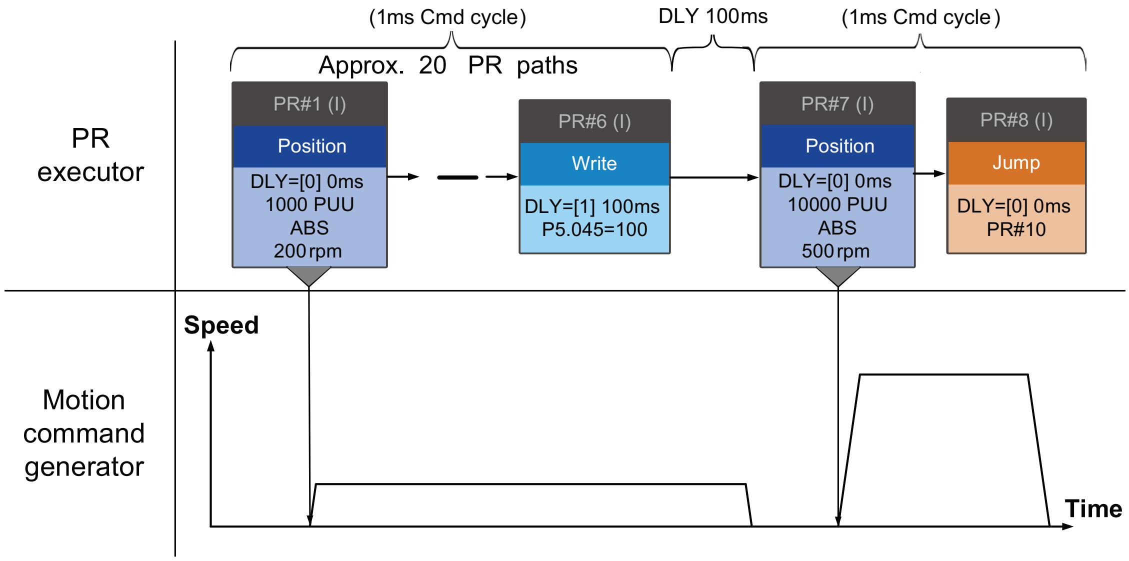


Figure 7-53 Multiple commands with delay

External Interruption

If an external interruption is encountered, it uses the PR Command trigger method to execute another PR path (refer to Section 7.1.5 for PR trigger methods). When the PR queue receives a PR path with an Interrupt function, it sends this path to the motion command generator immediately and changes the path in execution. Note that a delay does not change the result of an external interruption. That is, once the PR queue receives an external interruption command, the motion commands in the latter part are executed by the generator and integrated with the previous commands. The external interruption is as shown in Figure Figure 7-54.

If a PR path with external interruption enters the PR executor, the executor sends this Position command immediately to the generator so that the motor can run in accordance with the interruption. The motor uses the settings that integrate with the former motion commands when running. The methods of integration are described in Section 7.1.3. Similarly, an external interruption affects Speed and Position commands the same way and the same is true for multiple commands. See Figure 7-55 for an example.

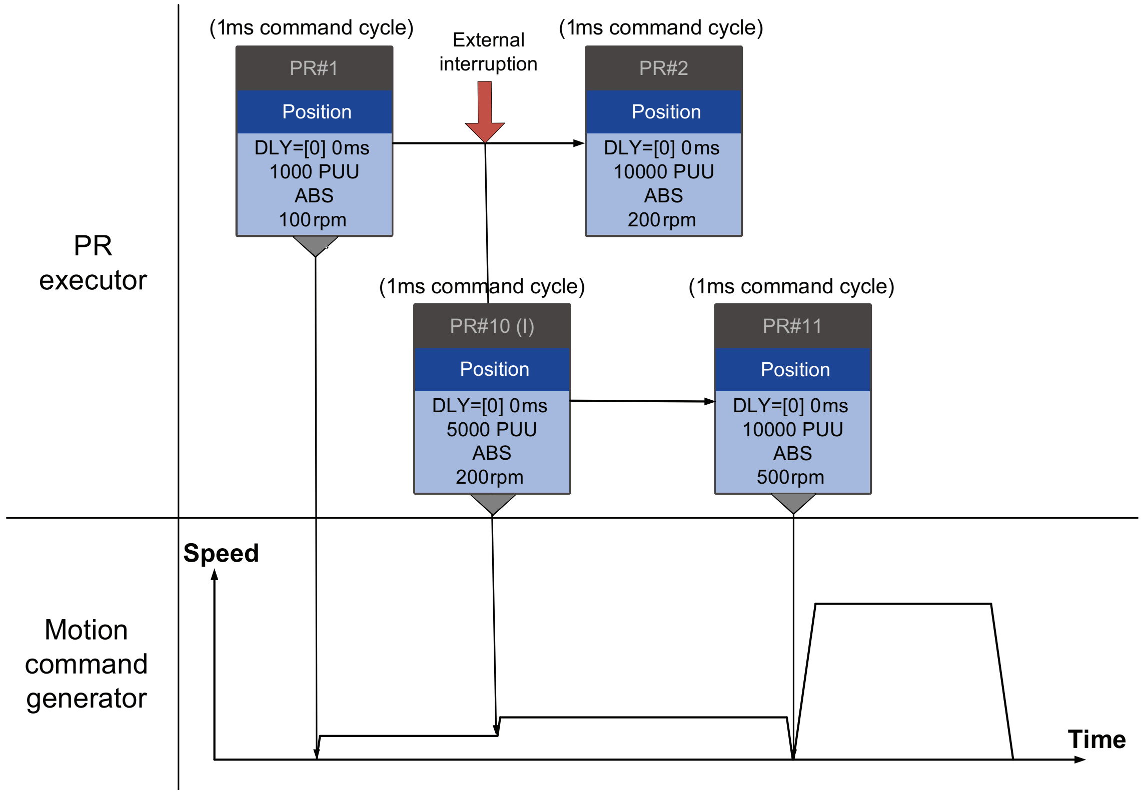


Figure 7-54 External interruption – Position command

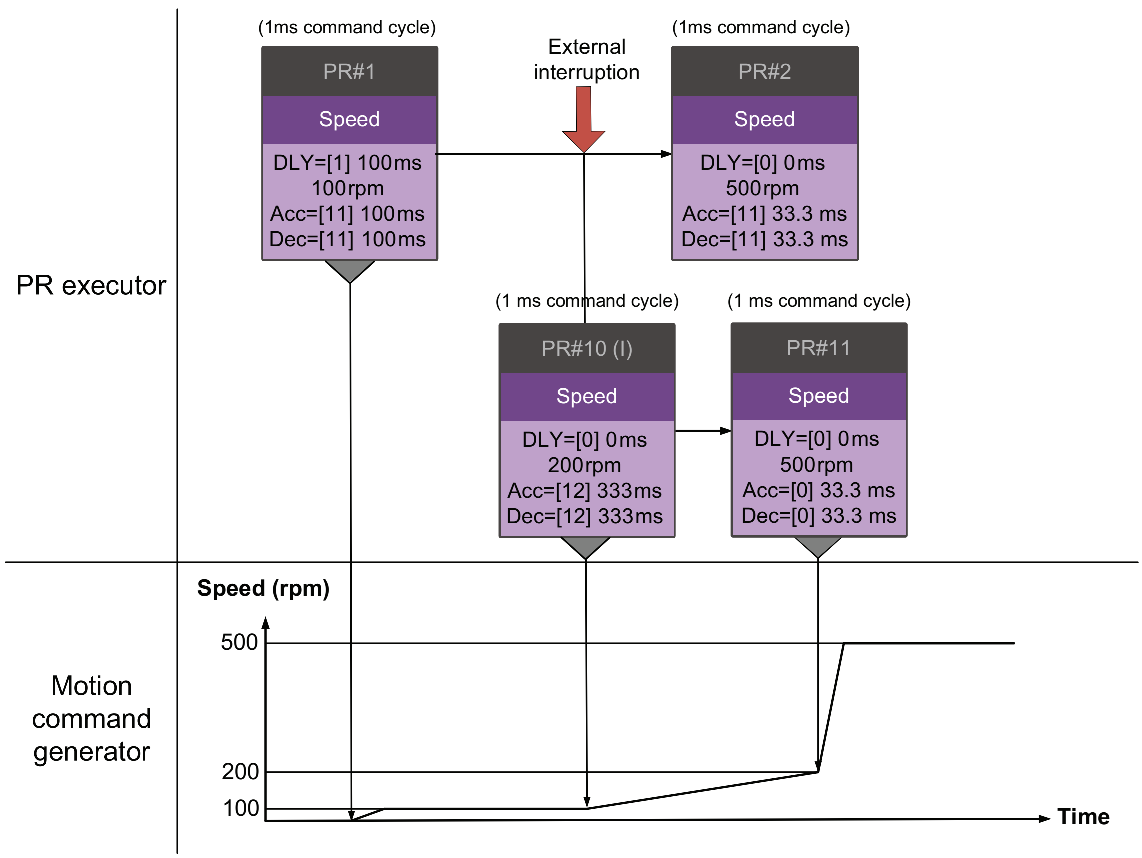


Figure 7-55 External interruption – Speed command

Overlap command

If the previous position command includes an Overlap function; it allows the next command to be executed while the previous motion is decelerating, thus achieving a continuous motion. This is also referred to as a blended move or a blended motion. When you use an Overlap command, the delay time is still effective. The delay time starts to count from the command's start point; however, in order to have the commands transition smoothly, setting the delay time of the previous command to 0 is suggested. In addition, if deceleration time of the previous command is identical to acceleration time of the next, the transition between commands can be very smooth, avoiding discontinuous speed during transition (see Figure 7-56 and Figure 7-57).

An Interrupt command has a higher priority than an Overlap command. Thus, when you set an Overlap function in the current Position command, and the next motion command includes an Interrupt function, only the command with the Interrupt function is executed.

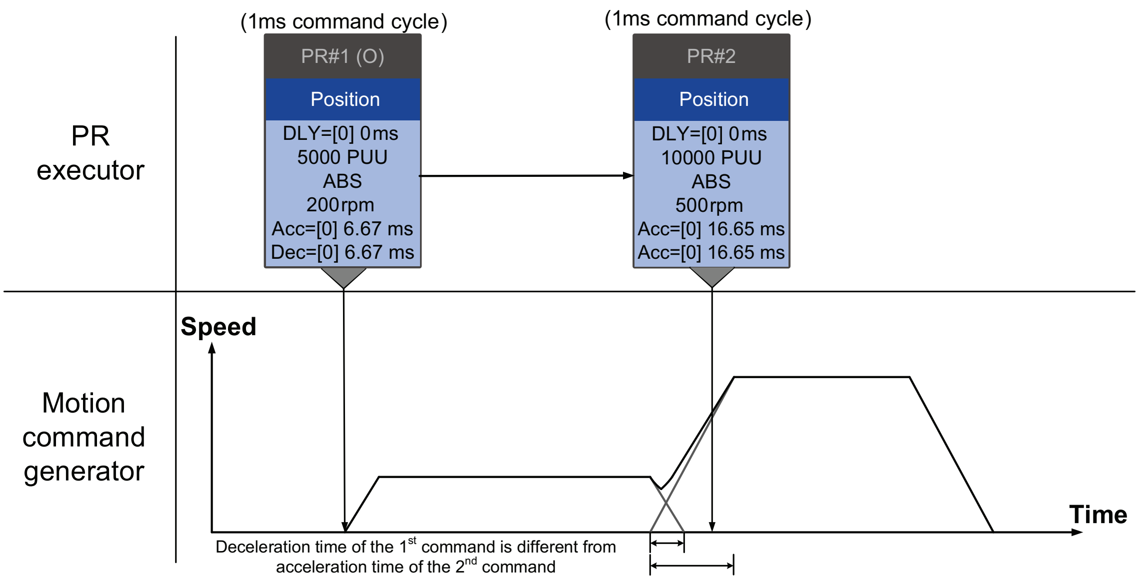


Figure 7-56 Overlap command - Acceleration and deceleration time are different

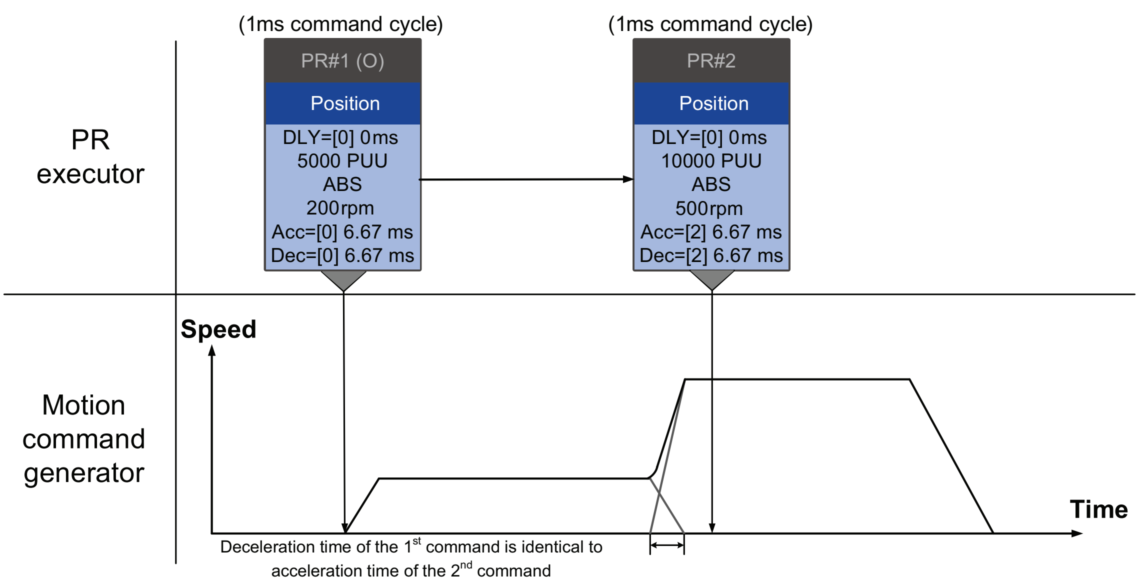


Figure 7-57 Overlap command - Acceleration and deceleration time are identical

Arithmetic operations (Statement)

You can regard arithmetic operation commands as combinations of Write commands and Jump commands. Thus, the execution priority is the same as these two types of commands, which are executed by PR executor. Arithmetic operation commands can interrupt the previous command but cannot be interrupted by the following command. This ensures that all arithmetic operations are completed before the PR paths enter the PR queue. In other words, for a series of PR paths with both arithmetic operations and Interrupt functions, only the arithmetic operations commands of this PR path are executed in the first ms. The rest are sent to the PR queue in the next ms.

Therefore, the jump target PR number specified by the path with arithmetic operations is executed in the next ms (see Figure 7-58). If you have entered the triggering parameter in the Statement section, such as PR command trigger register (P5.007) (which has the highest execution priority and is processed as an external interruption) after the arithmetic operations are done, the path specified by PR command trigger register is executed in the next ms. The logic condition commands are not executed (see Figure 7-59).

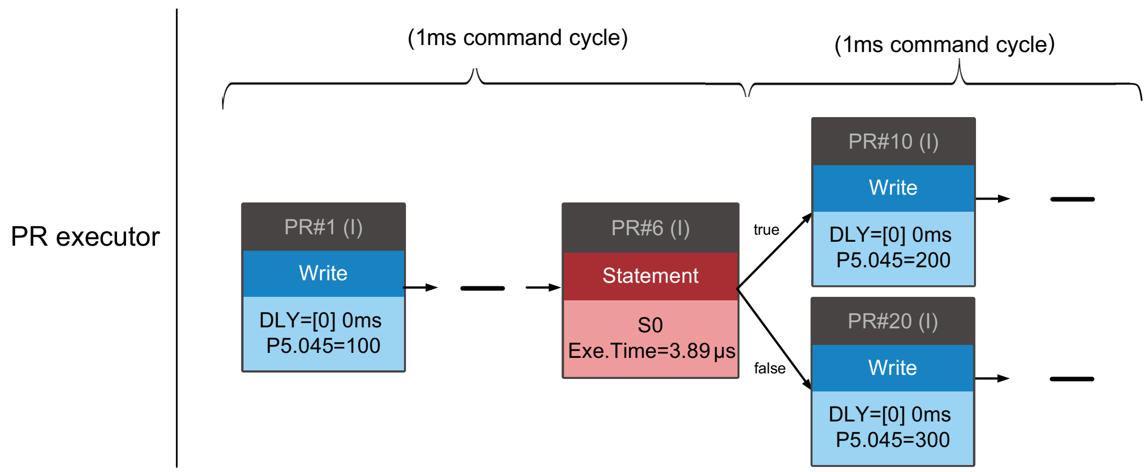


Figure 7-58 Multiple commands with arithmetic operations

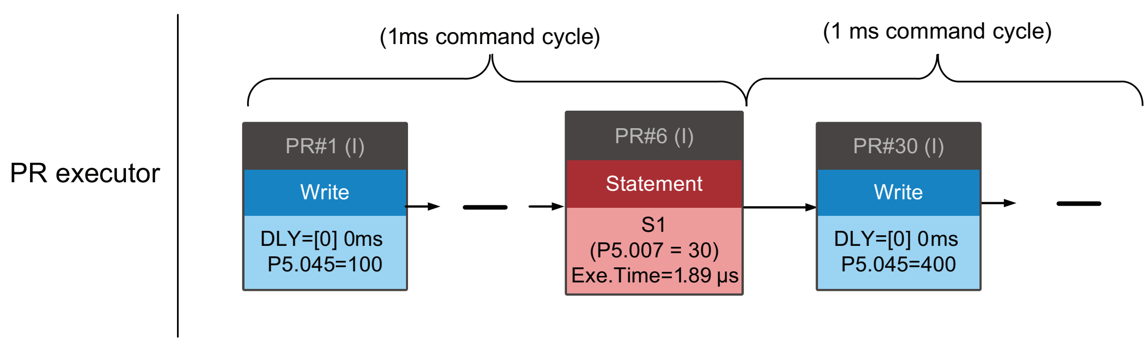


Figure 7-59 Writing trigger command in Statement section

7.2 - Application of motion control

Applications of motion control in the SureServo2 drive include high-speed position capture (Capture), high-speed position comparing (Compare), and E-Cam. High-speed position capturing uses digital input (DI7) to capture the motor's feedback position instantly and store this position in the data array. For high-speed position comparing, it writes the specified motor position to the data array and outputs a high-speed digital signal (DO4) once the motor feedback position reaches this specified position. The purpose of E-Cam is to create an E-Cam curve according to the correlation between the Master and the Slave, and then store the curve in the data array. The Slave axis refers to the Master axis position and moves to the position specified by the E-Cam. You can find more details about the setting and how it works in the following sections.

7.2.1 - Data array

The data array can store up to 800 32-bit sets of data (0–799). You can use it to store the high-speed capture data and high-speed compare data as well as the E-Cam curves. You have to segment the space for these three functions as their individual spaces are not defined by default. This prevents overwriting or accidentally changing any data. You can set P2.008 to 30 and then 35 or use SureServo2 Pro to write the data to EEPROM; otherwise, the data is not saved after you turn the power off. SureServo2 Pro includes a user-friendly screen for reading and writing the data array. See the following figure.

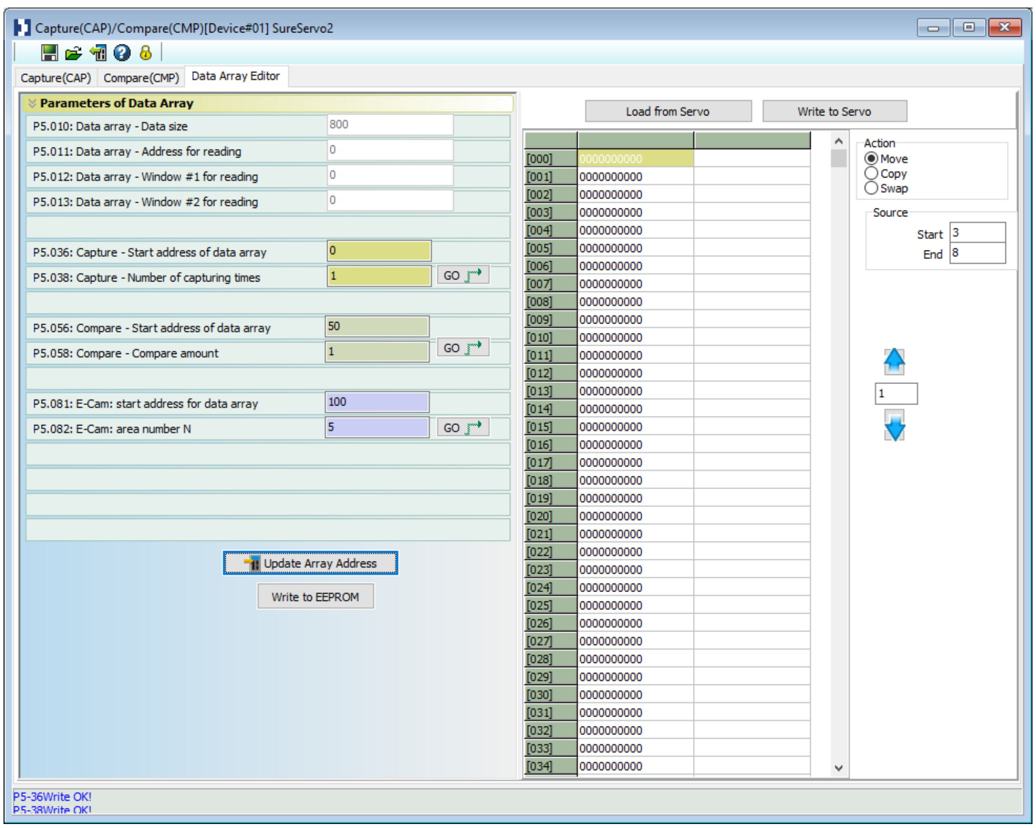


Figure 7-60 Data Array screen in SureServo2 Pro

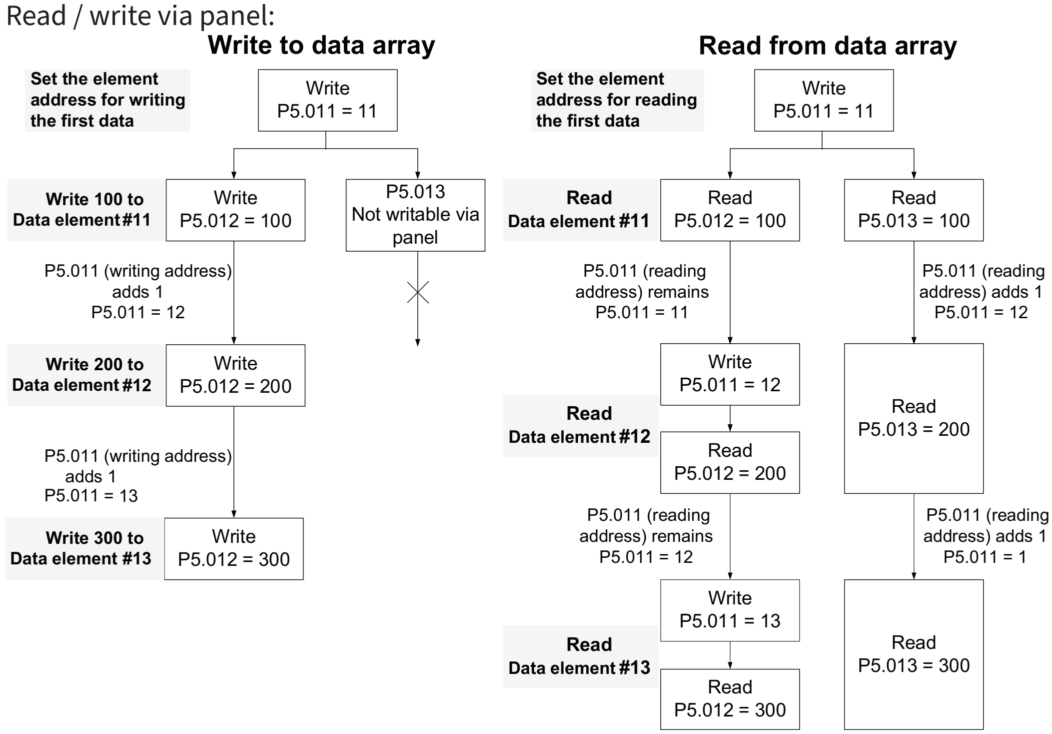
You can use the panel, communication, or SureServo2 Pro to read data from, or write data to, the data array. The first parameter group for reading and writing the data array is P5.011, P5.012, and P5.013. P5.011 specifies the address for reading and writing the value of the elements for the data array. P5.012 and P5.013 are for reading and writing the actual data contents to the element address in the data array. You can use both to read and write, but the behaviors after reading and writing differ. Please refer to Table 7-10. The second parameter group for reading and writing the data array is P5.011 and P5.100–P5.103. P5.100 reads data from or writes data to the data array element address set by P5.011. P5.101 reads data from or writes data to the data array element address following the address set by P5.011. P5.102 and P5.103 work the same way. If the address value accumulates and exceeds the maximum of 799, the returned address is 0. You can find more details in Table 7-11.

Table 7-10 Group 1 – reading / writing the data array

Parameter	***Description***		
P5.011 Address for reading / writing	Specify the address of the first element for reading from or writing to the data array		
Data Values for Reading / Writing	***by***	***Behavior After Reading***	***Behavior After Writing***
P5.012 Data Value #1 for reading / writing	Panel	Value of P5.011 does not increment 1	Value of P5.011 increments 1
	Communication / SureServo2 Pro	Value of P5.011 increments 1	Value of P5.011 increments 1
P5.013 Data Value #2 for reading / writing	Panel	Value of P5.011 increments 1	Cannot be written with the drive panel
	Communication / SureServo2 Pro	Value of P5.011 increments 1	Value of P5.011 increments 1

Example: when using the drive panel or communication for reading from or writing to the data array, input values to the data array address in sequence as follows: Data array #11 = 100, Data array #12 = 200, Data array #13 = 300. Then the data is read in sequence.

1) Read / write via panel:



2) Read and write using communication:
Reading and writing using communication requires Modbus RTU, MODTCP, or EtherNet/IP (Explicit only) communications. You can use the communication command 0x10 to write consecutively, 0x06 to write single data, and 0x03 to read consecutive data. First, use a consecutive writing command to write 100 to Data element #11, 200 to Data element #12, and 300 to Data element #13. When reading, use a single data writing command to set the start address as Data element #11, then use a consecutive reading command to read P5.011–P5.013 (Data element #11 and #12). This reads two values, so P5.011 is incremented by 2 and then it reads Data element #13.

Writing to Data Array

Packet	Modbus Function Code	Start Address	Data Size	P5.011		P5.012		P5.013	
				Low byte	High byte	Low byte	High byte	Low byte	High byte
1	0x10	P5.011	6 words	11	0	100	0	200	0
2	0x10	P5.011	6 words	13	0	300	0	0	0

Reading Data Array

Packet	Modbus Function Code	Start Address	Data size	P5.011		P5.012		P5.013	
				Low-byte	High-byte	Low byte	High byte	Low byte	High byte
4	0x06	P5.011	-	11	0	-	-	-	-
5	0x03	P5.011	6 words	11	0	100	0	200	0
6	0x03	P5.011	6 words	13	0	300	0	0	0

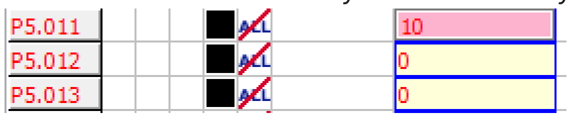
Table 7-11 Group 2 – reading and writing the data array

Parameter	***Description***	***Example 1***	***Example 2***
P5.011 Read / write address	Specify the address for reading from or writing to the data array	200	797

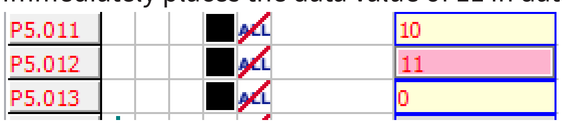
Parameter	***Description***	***Example 1***		***Example 2***	
		Address	***Content***	***Address***	***Content***
P5.100 Data value #3 for reading / writing	Read from or write to the address specified by P5.011.	200	1234	797	5678
P5.101 Data element - Data Value #4 for reading / writing	Read from or write to the first address following the address specified by P5.011.	201	2345	798	6789
P5.102 Data element - Data Value #5 for reading / writing	Read from or write to the second address following the address specified by P5.011.	202	3456	799	7890
P5.103 Data element - Data Value #6 for reading / writing	Read from or write to the third address following the address specified by P5.011.	203	4567	x	0

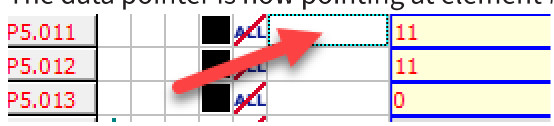
Data Array Intro Example

Set P2.008 to 30 and then 35 or use SureServo2 Pro to write the data to EEPROM; otherwise, the data is not saved after you turn the power off. This is only needed if you want to retain the data array after a power cycle. The below method uses the SureServo2 Pro Parameter Editor to read and write data to the array.

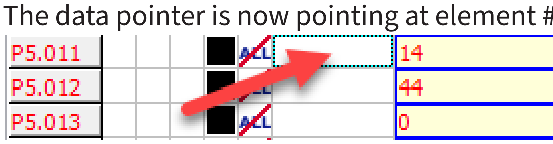
1) P5.010 = 10 -> Enter
   a) Enters the Element number you want the Array to start at with P5.011.
   b) 

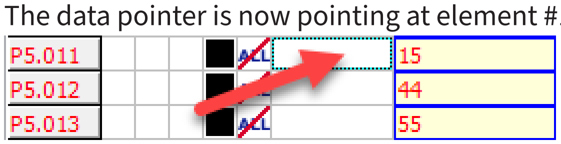
Group 1 Data Element Value 1 and 2 (P5.012, P5.013) can be written to the array starting at the P5.011 address and the element address will auto increment 1 address value.

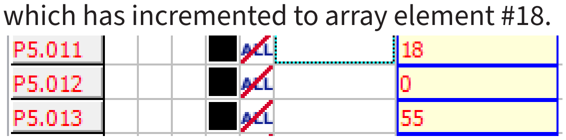
2) P5.012 = 11 -> Enter
   a) Immediately places the data value of 11 in data array element #10.
   b) 

3) Click in the “Read Parameter” box for P5.011 to read the new value of the array pointer.
   a) The data pointer is now pointing at element #11.
   b) 

4) P5.012 = 22 -> Enter, P5.012 = 33 -> Enter, P5.012 = 44 -> Enter.
   a) Immediately places the data value of:
      i) 22 in data array element #11
      ii) 33 in data array element #12
      iii) 44 in data array element #13

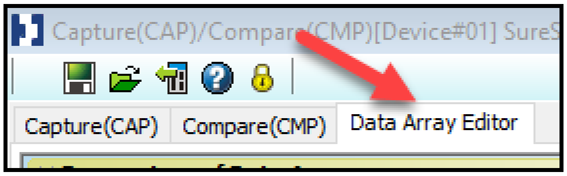
5) Click in the “Read Parameter” box for P5.011 to read the new value of the array pointer.
   a) The data pointer is now pointing at element #14.
   b) 

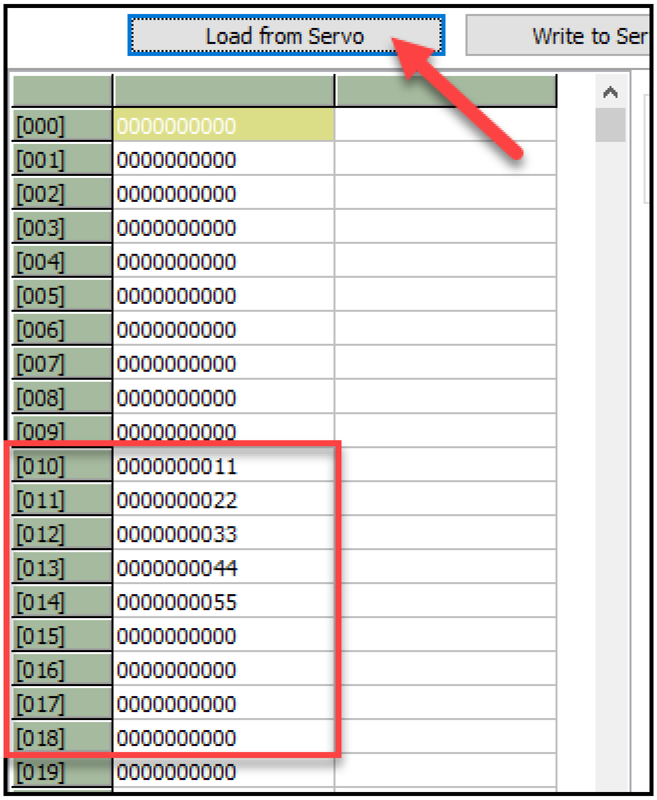
6) P5.013 = 55 -> Enter (Not P5.012) then Click in the “Read Parameter” box for P5.011 to read the new value of the array pointer.
   a) Immediately places the data value of 55 in data array element #14
   b) The data pointer is now pointing at element #15.
   c) 

7) Click in the “Read Parameter” box for P5.012 three times.
   a) This will execute 3 reads from P5.012. Each read will increment the array data pointer which has incremented to array element #18.
   b) 

8) To verify what is in the data array, open the “High Speed Position Capture/Compare” window and select the “Data Array Editor”. Click “Load from Servo” to update the array in SV2-PRO. You can now see the values you manually entered.

a) 

b) 

c) 

9) The method above walks the user through the understanding of how data is written into different block of the 800 element data array. Group 1 uses P5.012 and P5.013. Group 2 behaves the same using P5.100, P5.101, P5.102, and P5.103. This method is designed for keypad entry or PLC communication population. If you were to actually use SM-PRO you can just enter it in the “High Speed Position Capture/Compare” window and type in the values into the Data Array click “Write to Servo”. SureServo2 Pro does not allow csv import of data arrays.

Group 2 Data Element Values 3, 4, 5, and 6 (P5.100, P5.101, P5.102, and P5.103) can be written to the array starting at the P5.011 address but the element address will NOT auto increment. Instead, they are written as follows:

- *Data in P5.100 -> written to array element value of P5.011*
- *Data in P5.101 -> written to array element value of P5.011 +1*
- *Data in P5.102 -> written to array element value of P5.011 +2*
- *Data in P5.103 -> written to array element value of P5.011 +3*

7.2.2 - High-speed position capturing function (Capture)

The high-speed position capturing function (CAP) uses the external-triggered high speed digital input DI7 (with execution time of only 5μs) to capture the position data of the motion axis and store it in the data array for further motion control. As the Capture function is executed by the hardware, there is no lag in the software, and it is able to capture the motion axis' position accurately. While the Capture function is enabled, DI7 is automatically defined as the Capture signal. DI7 cannot be user defined through the software as the Capture Signal. Whatever function DI7 was previously defined as gets overwritten once the Capture function is enabled. Once the Capture function is disabled, DI7 is automatically reset to Disable (0x00), DI7 will not revert back to the previously user defined value. High Speed Capture is only available in PR Mode.

The flow chart for high-speed position capturing is shown in Figure 7-61. The relevant parameters are defined as follows. P5.036 stores the start position for capturing in the data array; if it is not set, the default start position is #0. P5.038 sets the capturing amount. The amount has to be greater than 0, otherwise the Capture function is not executed. P1.019.X enables the cycle mode. When the last data is captured, the capturing amount is reset to 0 (P5.038 = 0), and the next cycle starts automatically to capture the set capturing amount. However, the start capturing position is still determined by P5.036; that is, the captured data in previous cycle is replaced by the data captured in the next cycle. P5.039 enables and disables the Capture function and other settings. See the following table for more information. To avoid capturing multiple false position data, you can use P1.020 to set the masking range for capturing to avoid false triggers due to EMI noise or a double bounce of the capture trigger DI7. This prevents the same position data being captured repeatedly because capturing more than once is not allowed in the masked area. You can set the Capture function in SureServo2 Pro, as shown in Figure 7-62.

P5.039	Bit	Function	Description
X	0	Enable Capture function	When P5.038 > 0 and bit 0 = 1, the capturing starts and DO.CAP_OK is off. Each time a position is captured, value of P5.038 is decremented by 1. When P5.038 = 0, it means the capturing is finished, DO.CAP_OK is on, and bit 0 is reset to 0. If bit 0 is already 1, the written value must not be 1; you must write 0 to disable the Capture function.
	1	Reset position when first data is captured	If bit 1 = 1, after the first data is captured, set Capture axis' position to the value of P5.076.
	2	Enable Compare function after first data is captured. The Compare Function is only supported in the Capture Trigger.	If bit 2 = 1, when the first data is captured, the Compare function is enabled. Activate Compare Bit P5.059.X bit 0 = 1 and P5.058 (Remaining Counts) resets to the previous value. If the Compare function is already enabled, then this bit function is ignored.
	3	Execute PR#50 after the last data is captured	If bit 3 = 1, execute PR#50 once all data are captured.
Y	-	Source of Capture axis	0: disabled 1: auxiliary encoder (CN5) 2: pulse command (CN1) 3: main motor encoder (CN2)
Z	-	Trigger logic	0: NO (normally open) 1: NC (normally closed)
U	-	Minimum trigger interval (ms)	-

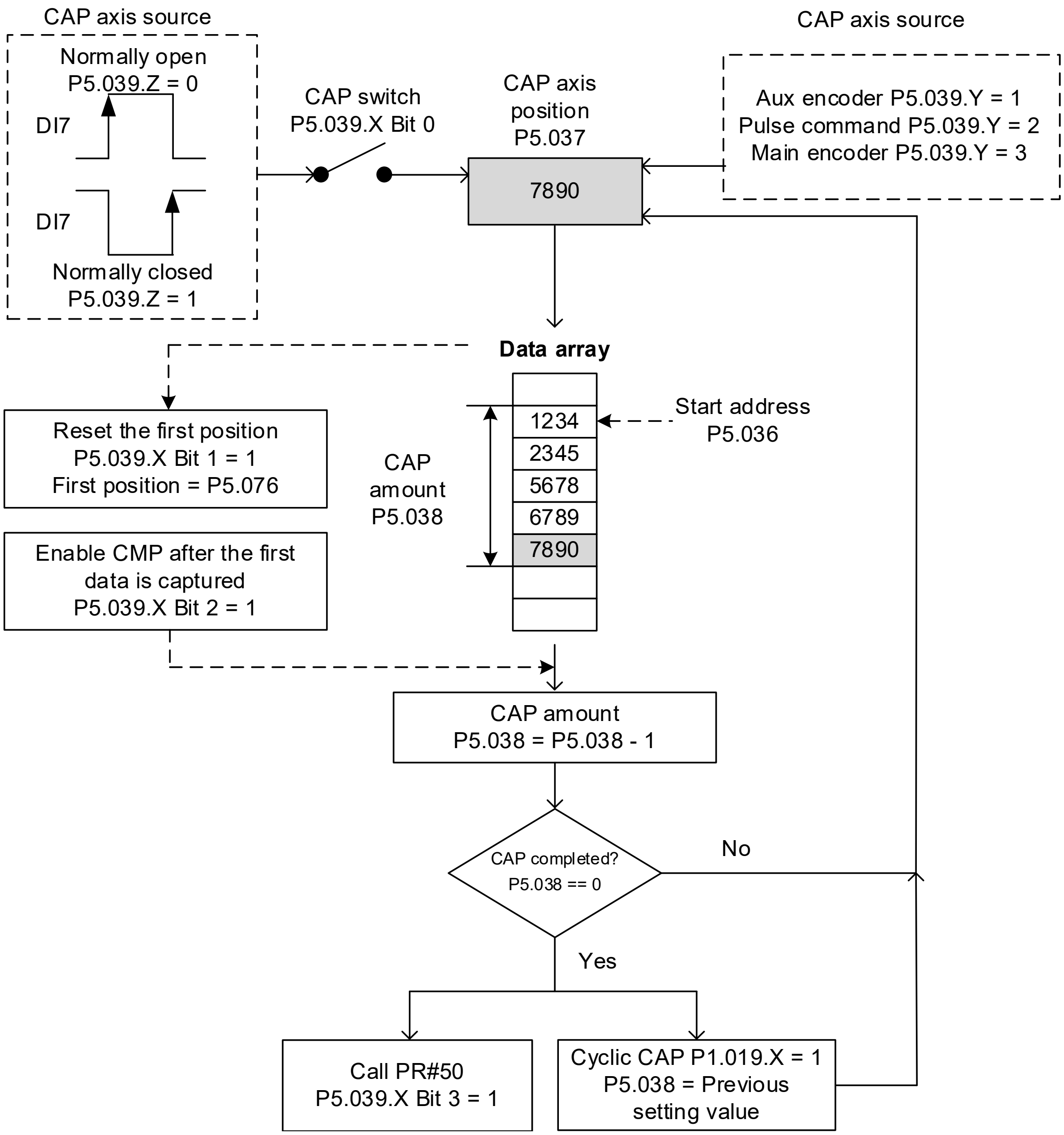


Figure 7-61 Flow chart for high-speed position capturing

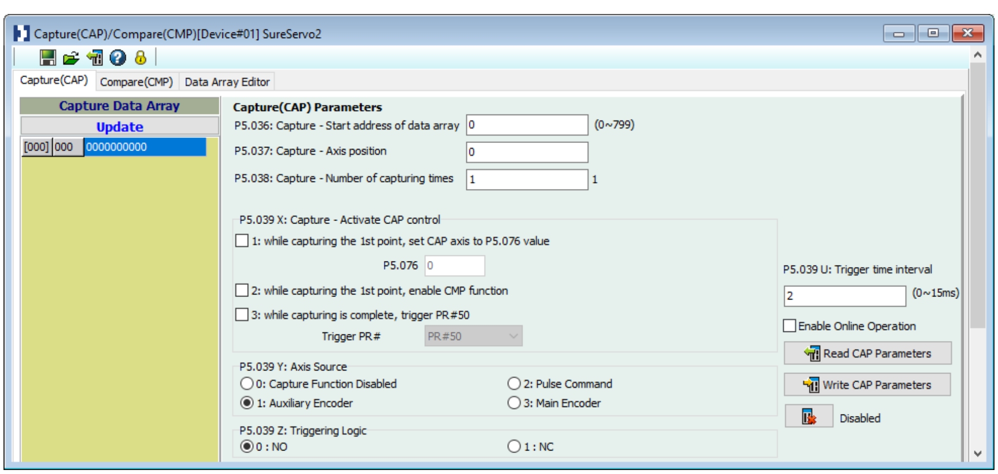


Figure 7-62 Capture Function screen in SureServo2 Pro

Walk Through Capture Example Intro

Use PR path programming to use the motion commands with the Capture function. Use Write commands to set the high-speed position Capture function, as well as to execute motion commands once the Capture cycle is completed. See the example in Figure 7-63.

1) Reset the drive to defaults (P2.008=10), cycle power and set limit switch inputs and OVRD stop as needed.
2) Set P1.001 = 1 (PR Mode).
3) Set Homing Mode to P5.0004.X = 8 (Define Current Position as the Origin) Since the motor's encoder is configured to be the capture axis (PR#6 below), the Capture Axis position (P5.037) is reset to 0 during the homing routine.
4) Configure the PR paths:
   - *PR#1 confirms that the Capture function is disabled (P5.039.X Bit 0 = 0).*
   - *PR#2 sets the start position of data array to #000.*
   - *PR#3 sets the capturing amount to 3.*
   - *PR#4 sets the capturing axis' position to 0 for the first capture point.*
   - *PR#5 sets the cyclic capture mode with delay time of 1 second to ensure that the next PR path can be executed with the Capture function.*
   - *PR#6:*
     i) Enables the Capture function
     ii) Resets the first point to the value of P5.076
     iii) Once the data is captured the drive proceeds to path PR#50
     iv) Selects the motor's encoder as the capturing axis
     v) Uses 'normally open' contact as the trigger logic
     vi) And sets a trigger interval of 2ms
   - *PR#7 sets the Speed command to 50 rpm.*
   - *PR#50 sets the move command (50000 PUU) per the last captured position once the CAP is completed.*
   - *PR#51 resets speed to 50 rpm.*

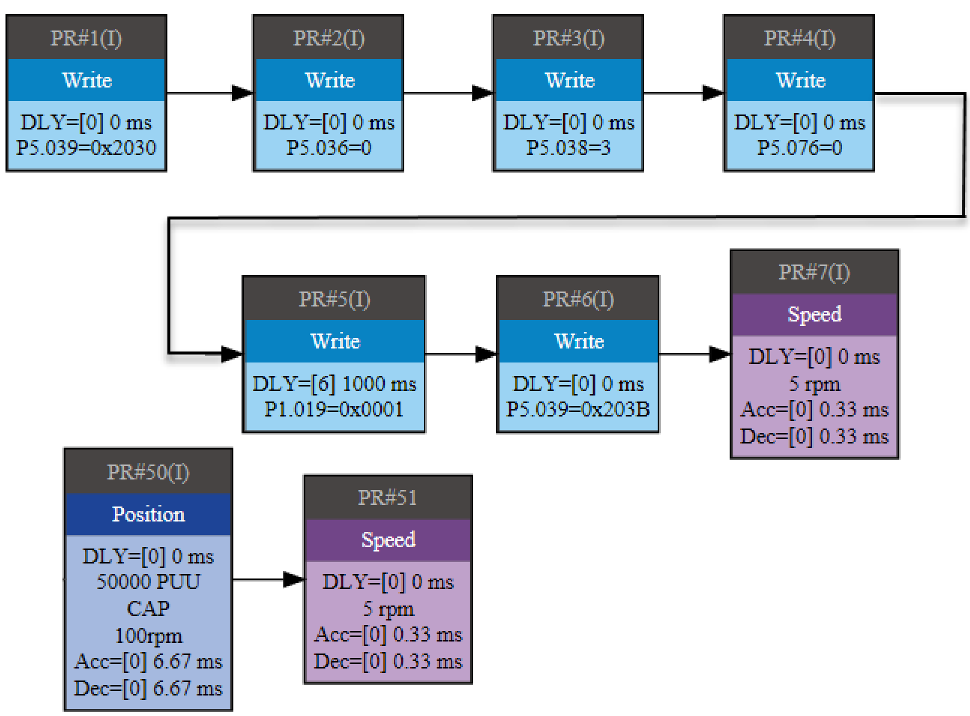
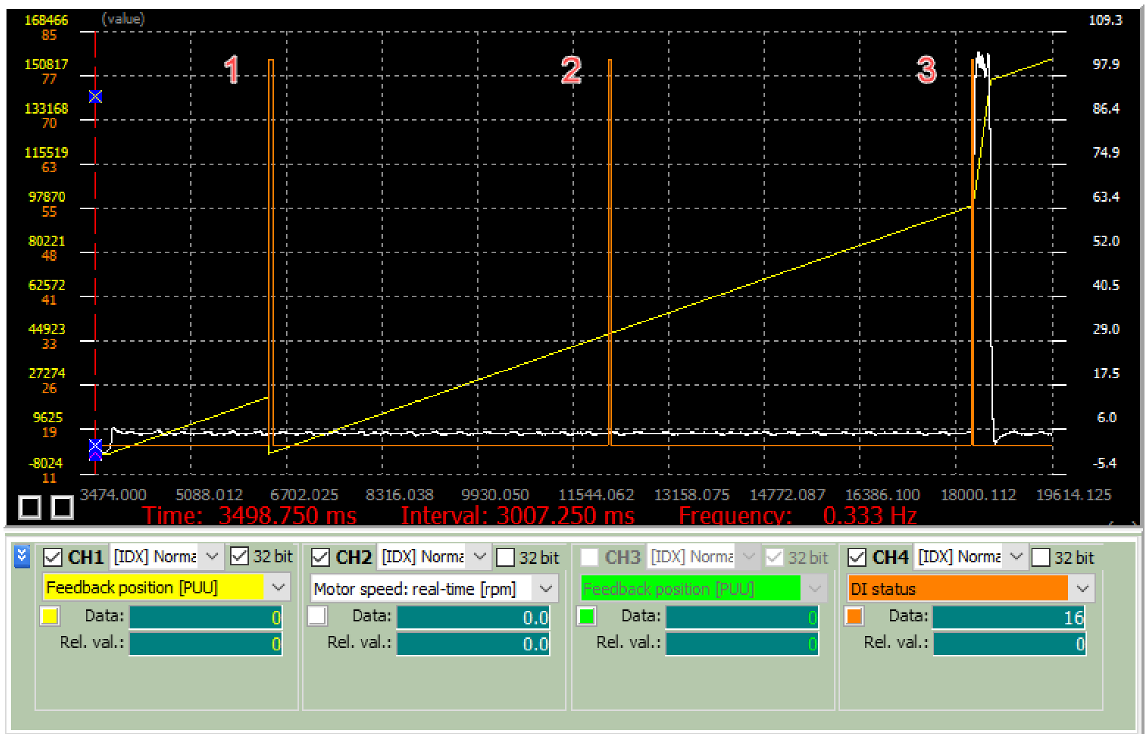


Figure 7-63 PR path with application of high-speed capture function

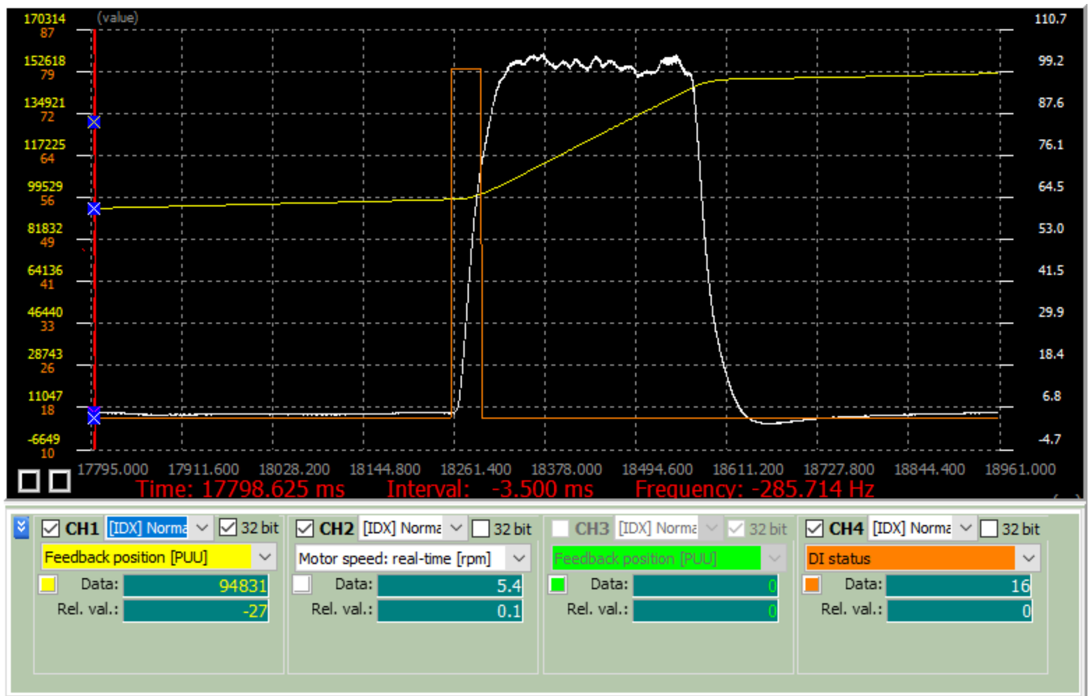
5) Set up the Scope in SureServo2 Pro to easily visualize what is happening.
   a) CH1 = Feedback Position (PUU)
      i) Be sure to check the 32 bit box. One revolution is 100,000 PUU. If left at 16 bit then the scope Y axis will roll over at 65535 total pulses. This will cause CH3 to be void since the 16 bits of CH3 will be used for the CH1 32 bit value.
   b) CH2 = Motor speed: real time (rpm). Uncheck the 32 bit box.
   c) CH4 = DI status.
6) Enable the servo.
7) Run PR path #0 from the PR Mode Editor window to home the axis.
8) Click Run on the Scope window.
9) Run PR path #1 from the PR Mode Editor window to run the PR path registers.
10) Trigger DI7 3 times within the scopes time capture limit.
11) Stop the scope and stop the PR path.

You should have a scope capture like the one below after you adjust your channel axis min and maxes.

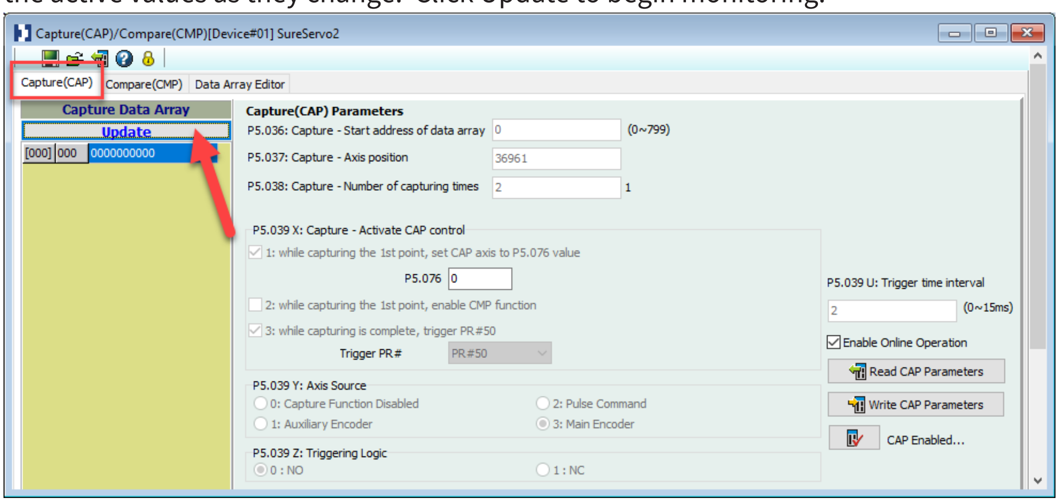


The above scope capture shows three input pulses from DI7. DI status is 80 decimal (50 hex) which represents DI5 (Set to disabled in this example) and DI7 are on. On the first pulse, the Position of the capture axis is reset to 0. This was configured by P5.039.X bit 1=1 that defines after the first data is captured, set Capture axis' position to the value of P5.076.

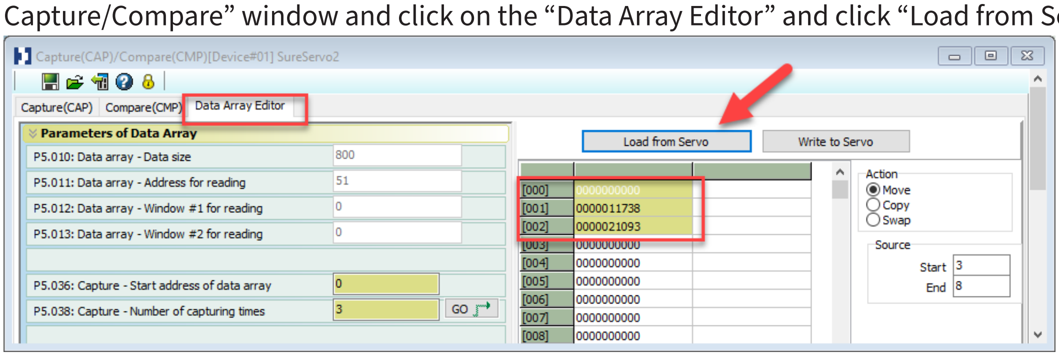
Since P5.038 = 3 (three data points for capture cycle) on the 3rd pulse the capture path is called (PR#50). Below you can see a zoomed in view of the third pulse. The scope shows the motor speed jump to 100 rpm for 50,000 PUU then returns to 5 rpm.



To view the captured data in SV2-PRO, open the “High Speed Position Capture/Compare” window and click on the “Capture (CAP)” tab during operation. Here you can view several of the active values as they change. Click Update to begin monitoring.



To view the actual values as they appear in the Data Array, open the “High Speed Position Capture/Compare” window and click on the “Data Array Editor” and click “Load from Servo”.



From Figure 7-64, you see that after DI7 is triggered, the capturing axis is reset to 0 and the data stored in data element #000. P5.039.X bit1 = 1 tells the Capture event to reset the position after the first data point is captured. P5.076=0 defines that 0 will be the first data value written. When DI7 is triggered the second and third time, the position data is written to the data array element #001 and #002. Once the first capture cycle is completed, DO: [0x16]CAP_OK is set to on which then calls path PR#50 (high speed position capture command) and then proceeds to path PR#51 (motion with fixed speed). PR#50 is automatically called because P5.039.X Bit3 = 1. Then, the servo drive continues executing the next Capture cycle and sets DO: CAP_OK is set to off and the capturing amount is then set to 3 since P5.038 = 3. When DI7 is triggered for the fourth time, the capture axis' position is not reset according to the value in P5.076; the position data of the capturing axis is written to #000 again. Therefore, the data written in the previous cycle is replaced. At the moment DI7 is triggered the fifth and sixth time, the position of the capturing axis is stored in data array #101 and #102. As soon as the second capture cycle is finished, DO: [0x16]CAP_OK is set to on and then calls PR#50 (high-speed position capture command) and then PR#51 (motion with fixed speed).

When using Cyclic Capture mode (P1.019.X = 1), the Reset function is only valid for the first cycle. Executing the PR path is valid for every cycle; in other words, every time a cycle ends, PR#50 is executed. The first position data captured in every cycle is written to the data element number set by P5.036, and then the other data is written to sequential elements. So, position data written in the previous cycle is always replaced by the position data of the next cycle.

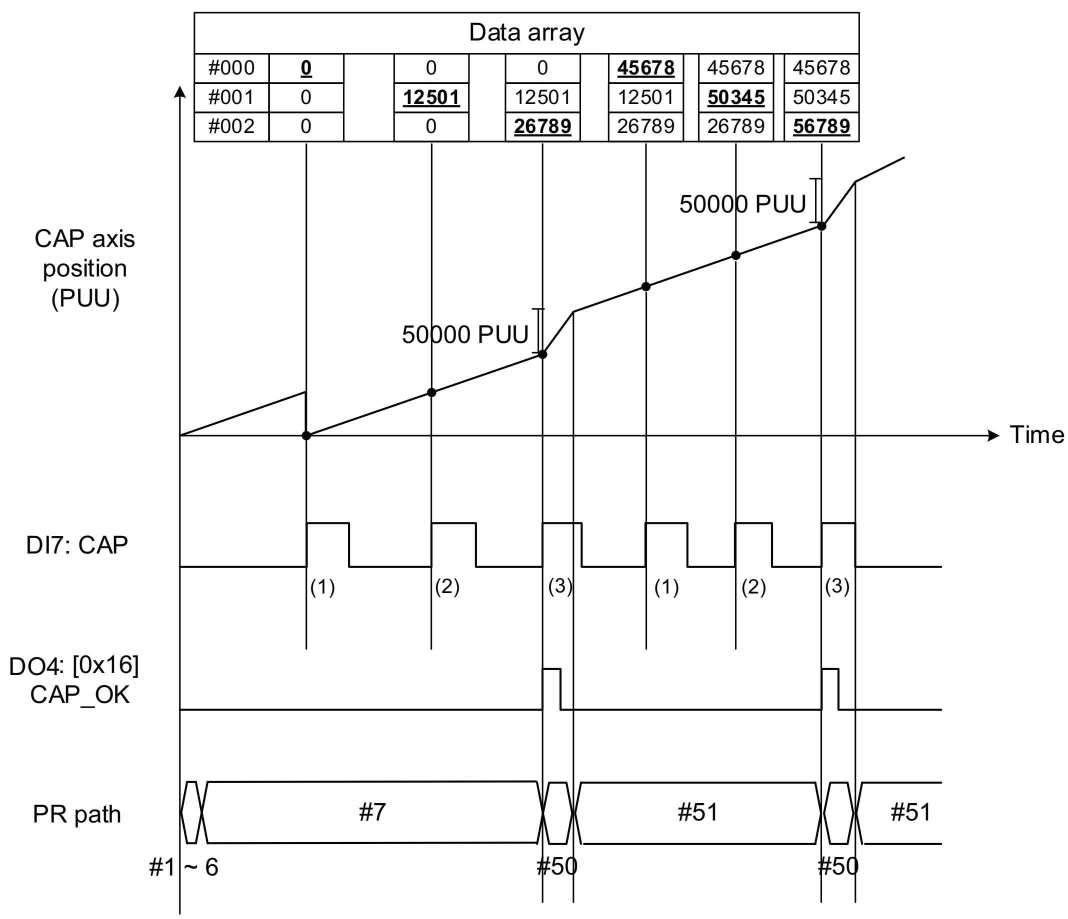


Figure 7-64 Application example for high-speed capturing function

7.2.3 - High-speed position comparing function (Compare)

The purpose of high-speed position comparing (CMP) is to compare the instant position of the motion axis with the value saved in the data array. When the compare condition is fulfilled (DO4 with execution time of only 5μs), a high-speed digital signal is sent immediately for user application motion control. Since this function is carried out by the hardware, there is no lag in the software and the position compare is more accurate on high speed motion axes. When the Compare function is enabled, the servo drive takes control of DO4, which cannot be user-defined, it is automatically defined when the Compare function is active.

As shown in Figure 7-65 Flow chart for the high-speed Compare function, P5.056 stores the start position of the data array for comparing (default is #50 in the data array). You must write the position data to be compared to the data array before comparing. P5.058 is the data size to be compared and must be greater than 0 or the function is invalid. P5.059 is the switch of the Compare function and for other settings. You can find more information in the table below. Please note that when the comparing source is the encoder output terminals of CN1, the pulse resolution of the comparing axis is set by P1.046 (numerator) and P1.097 (denominator) with default value of 2500 and 0 respectively; that is, the moving distance of the comparing axis is 10000 PUU per rotation of motor. The comparing position in the data array can be shifted using P1.023 (non-volatile) and P1.024 (volatile). You can reset P1.024 to 0 after the shift, and you can enable this function with P1.019.Z. You can also set the Compare function through SureServo2 Pro, as shown in Figure 7-66.

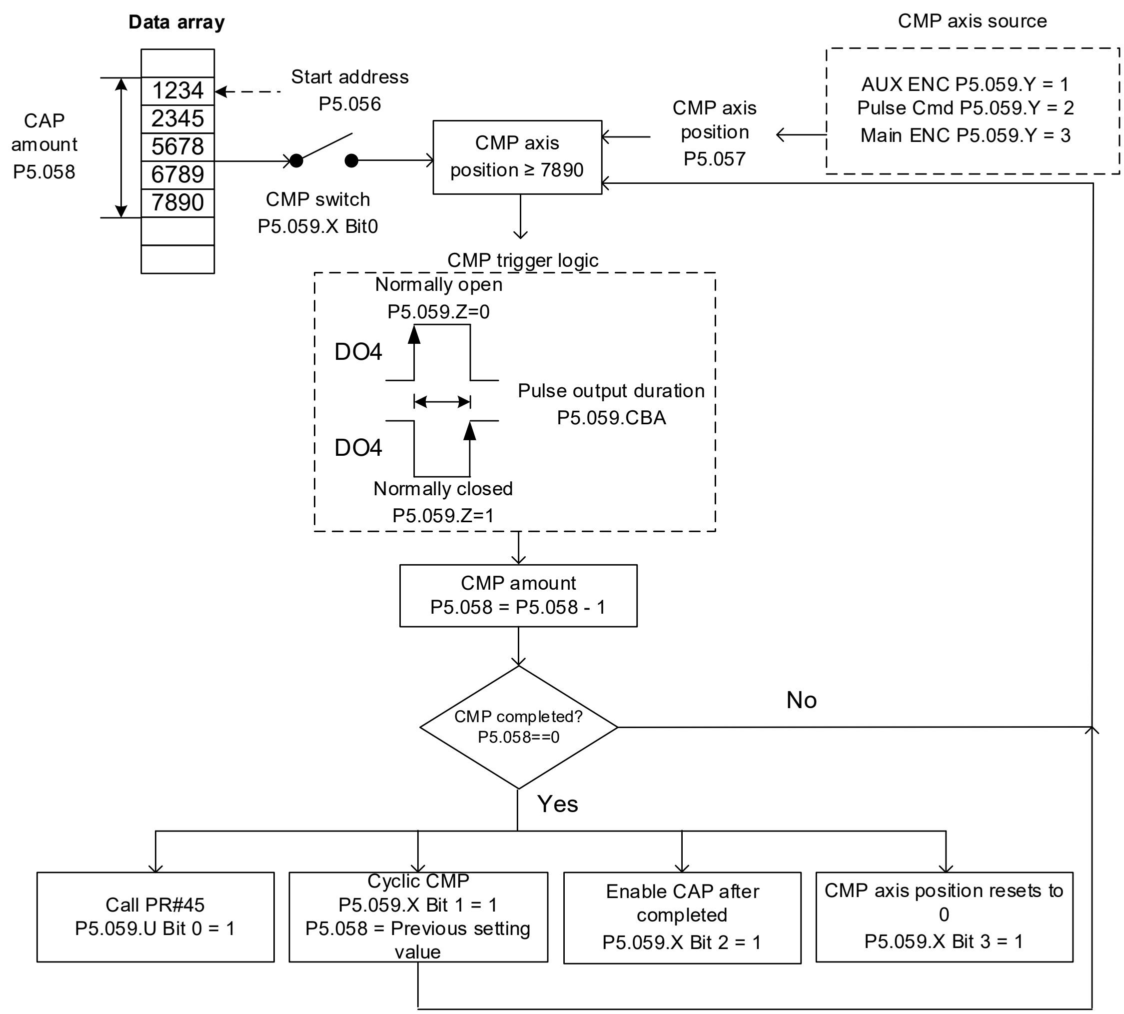


Figure 7-65 Flow chart for the high-speed position Compare function

Wiring
Parameters
DI/DO Codes
Monitoring
Alarms

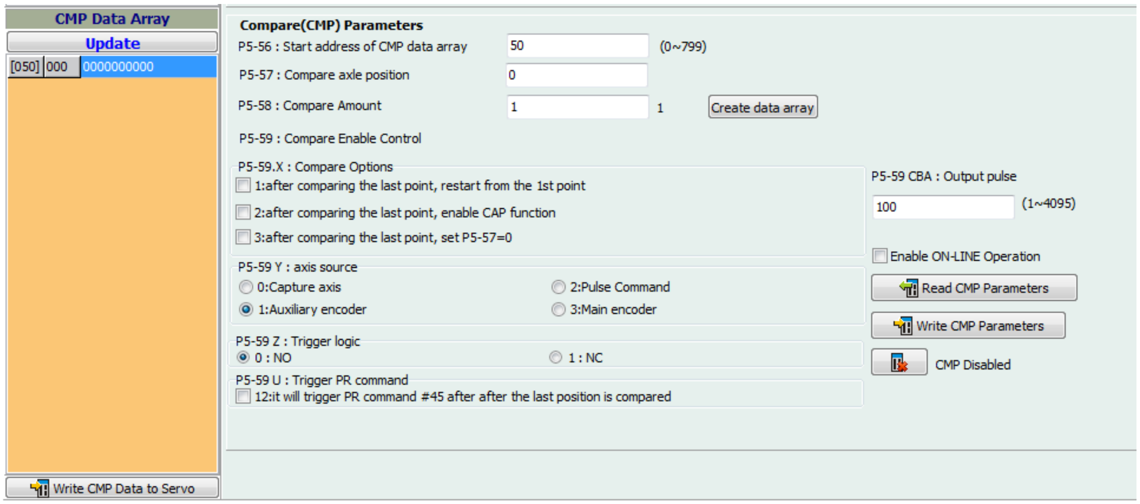


Figure 7-66 Compare Function screen in SureServo2 Pro

P5.059	Bit	Function	Description
X	0	Enable high-speed position compare function	When P5.058 is greater than 0 and bit 0 is set to 1, the comparing starts. The value of P5.058 decreases 1 every time a point in data array is compared. When P5.058 reaches 0, bit 0 is automatically cleared to 0. If bit 0 is 1, the new value to be written cannot be 1; you can only write 0 to disable the Compare function.
	1	Cycle mode	If bit 1 is set to 1 and all compare procedures are completed, P5.058 resets to the setting value and then the compare procedure starts again.
	2	Enable Capture function after data compared	If bit 2 is 1, after all comparing is done, enable the Capture function (Set P5.039.X bit 0 to 1, and set the previous value to P5.038 as the data size to be captured); if Capture function has been enabled, then this function is invalid.
	3	Reset position for the comparing axis to 0	If bit 3 is 1, set P5.057 to 0 once comparing is completed, the position for the comparing axis is reset to 0.
Y	-	Source setting of comparing axis	0: capturing axis 1: auxiliary encoder (CN5) 2: pulse command (CN1) 3: motor encoder (CN2) If capturing axis is selected, the source of the capturing axis (P5.039.Y) cannot be changed. If the motor encoder is selected, the pulse resolution is determined by P1.046 (Encoder pulse number output setting) and P1.097.
Z	-	Trigger logic	0: NO (normally open); 1 ： NC (normally closed)
U	-	Trigger PR path	If bit 0 is set to 1, PR#45 is triggered once the last data is compared.
CBA	-	Pulse output duration (ms)	-

Walk Through Compare Example Intro

Use PR path programming to use motion commands with the Compare function. You can use Write commands to edit the contents of the data array and set the high-speed position Compare function, as well as executing a motion command. As shown in Figure 7-67, you set the numerator (P1.046) and denominator (P1.097) for the encoder's pulse output (the default is based on the comparing axis runs of 10000 pulses per rotation of the motor).

1) Reset the drive to defaults (P2.008=10), cycle power and set limit switch inputs and OVRD stop as needed.
2) Set P1.001 = 1 (PR Mode).
3) Set Homing Mode to P5.0004.X = 8 (Define Current Position as the Origin) Since the motor's encoder is configured to be the capture axis (PR#6 below), the Capture Axis position (P5.037) is reset to 0 during the homing routine.
4) Configure the PR paths:
   - *PR#1-3 uses write commands to populate values into data array #50 - 52.*
   - *PR#4 confirms that the Compare function is disabled (P5.059.X Bit 0 = 0), Compare axis source is the motor's encoder (P5.059.Y=3) and the output pulse width is 100mSec.*
   - *PR#5 sets the start position to #50.*
   - *PR#6 sets the comparing data array size to 3*
   - *PR#7 resets the compare axis position to 0 and sets a delay of 1 ms to ensure that the PR path using the Compare function can be executed.*
   - *PR#8.*
     - i) Enables the Compare function in Cycle mode which resets the comparing axis to 0 on the first pulse of each cycle
     - ii) Selects CN2 (motor encoder) as the capturing axis source
     - iii) Selects the motor's encoder as the capturing axis
     - iv) Uses 'normally closed' contact as the trigger logic
     - v) Sets a (DO4) pulse output duration of 100 ms
     - vi) After the comparison is complete execute PR#45
   - *PR#9 initiates a Speed command of 50 rpm.*
   - *PR#45 initiates an Incremental command of 50000 (5000.0) PUU which is equivalent to a half turn of the motor*
   - *PR#46 resets the Speed command setting at 50rpm.*

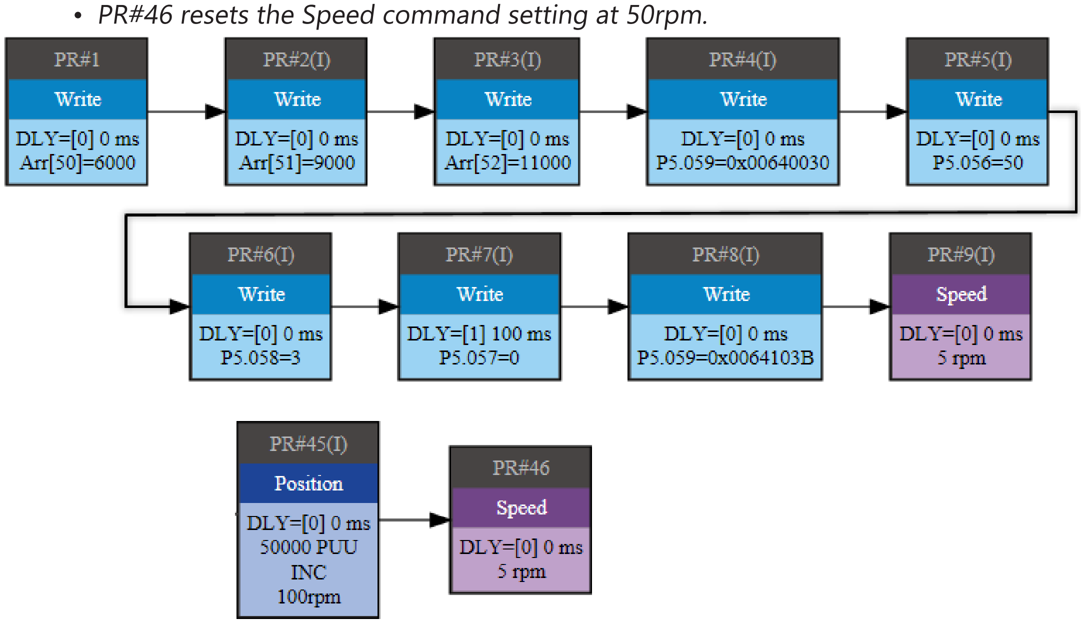
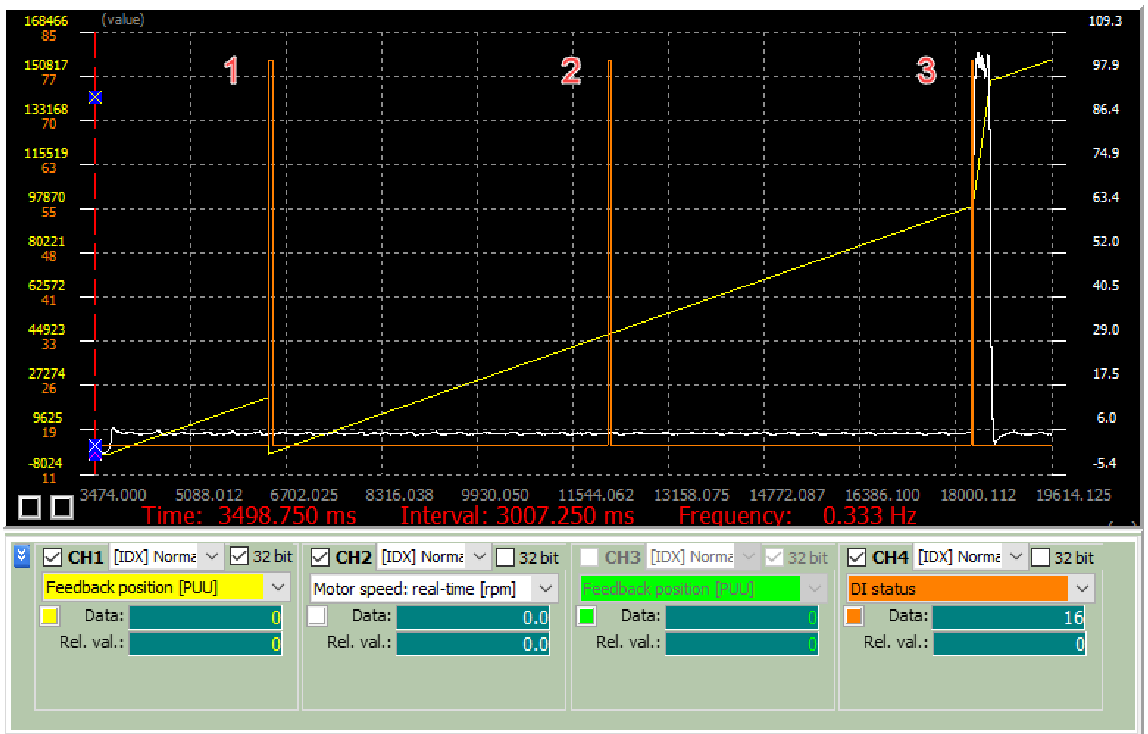


Figure 7-67 PR path using the Compare function

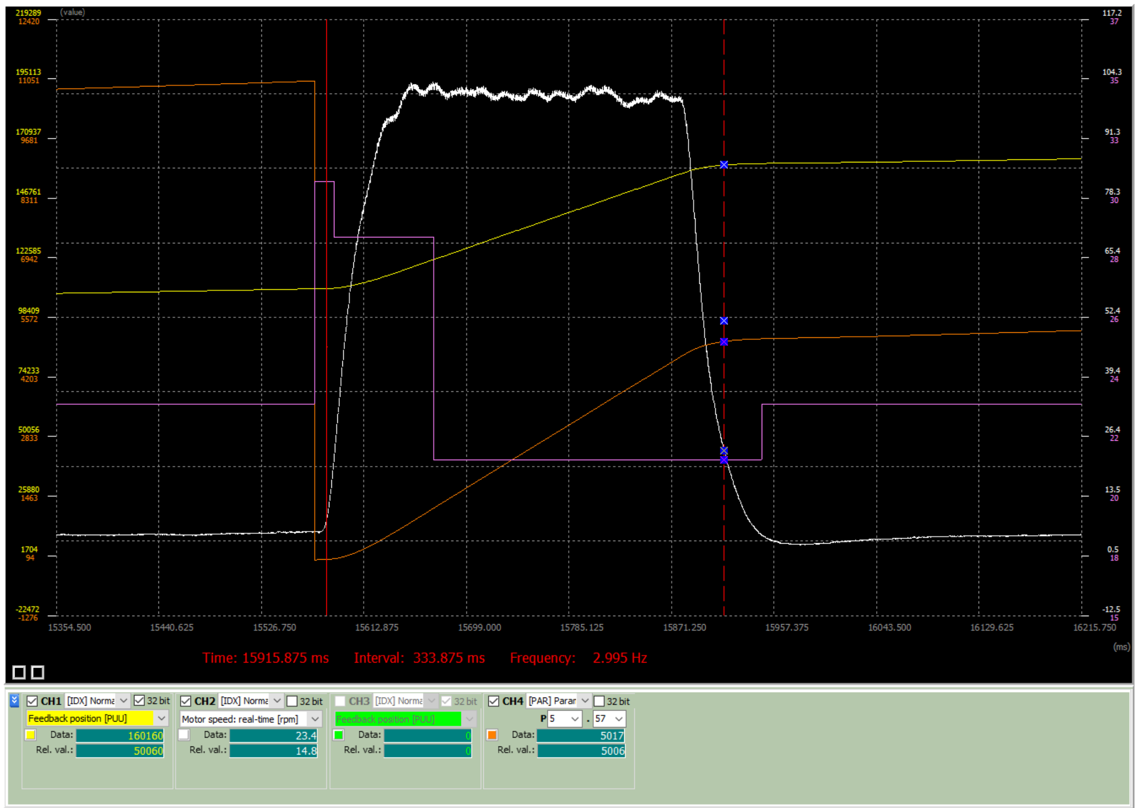
5) Set up the Scope in SureServo2 Pro to easily visualize what is happening.
   a) CH1 = Feedback Position (PUU)
      i) Be sure to check the 32 bit box. One revolution is 100,000 PUU. If left at 16 bit then the scope Y axis will roll over at 65535 total pulses. This will cause CH3 to be void since the 16 bits of CH3 will be used for the CH1 32 bit value.
   b) CH2 = Motor speed: real time (rpm)
   c) CH4 = P5.057 (compare axis position)
   d) CH5 = DO status
6) Enable the servo
7) Run PR path #0 from the PR Mode Editor window to home the axis.
8) Click Run on the Scope window.
9) Run PR path #1 from the PR Mode Editor window to run the PR path registers.
10) DO4 will pulse every time the compare axis is equal to the next element in the array. On the third pulse the compare axis position will reset to 0 and call PR45.
11) Stop the scope after three DO4 pulses.

You should have a scope capture like the one below after you adjust your channel axis min and maxes.

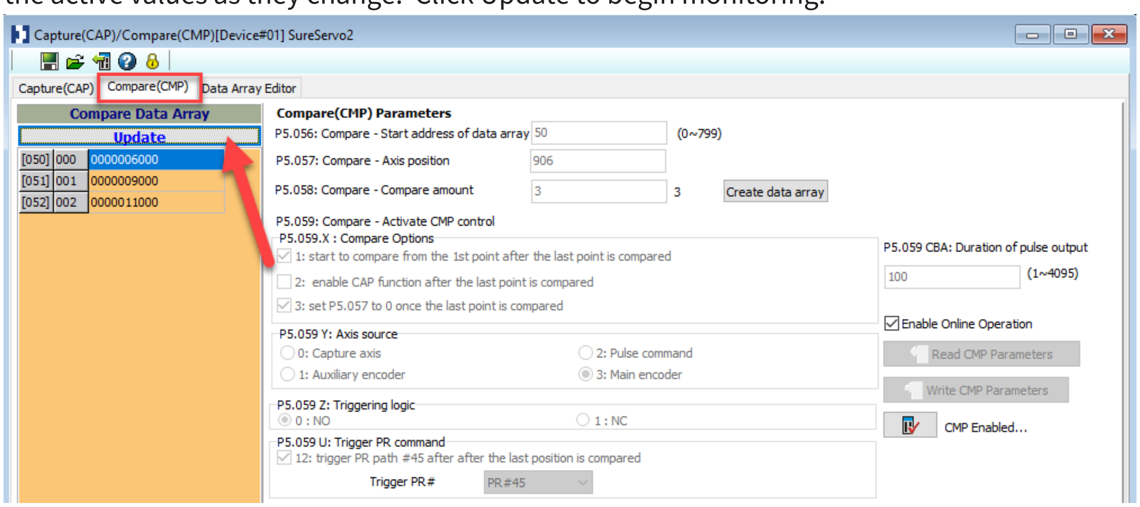


The above scope capture shows 3 DO4 pulses and on the 3 one the compare axis is reset to 0 then the position move 50,000 PUU as a speed of 100 rpm. On the first pulse, the Position of the compare axis is reset to 0. This was configured by P5.059.X bit 3=1 that defines after the compare is complete, set the compare axis' position to 0.

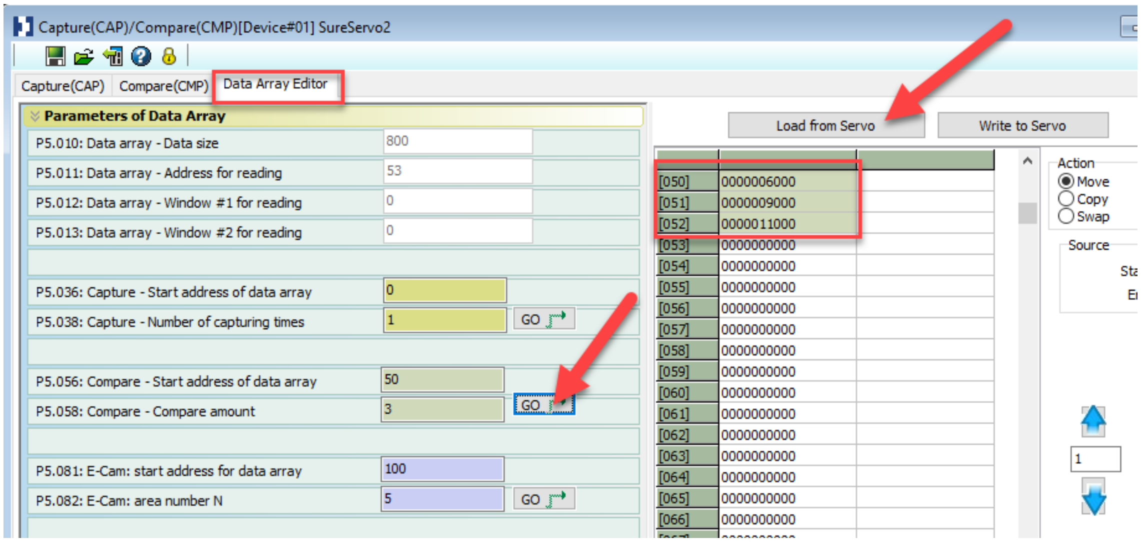
Below you can see a zoomed in view of the third pulse. The scope shows the motor speed jump to 100 rpm for 50,000 PUU then returns to 5 rpm. The capture axis gets reset to 0 but the motors actual encoder feedback pulses continue to increase.



To view the compare data in SV2-PRO, open the “High Speed Position Capture/Compare” window and click on the “Compare (CMP)” tab during operation. Here you can view several of the active values as they change. Click Update to begin monitoring.



To view the actual values as they appear in the Data Array, open the "High Speed Position Capture/Compare" window and click on the "Data Array Editor" and click "Load from Servo".



Looking at another example of three data points or 20000, 30000, 40000. From Figure 7-68, you see that when the comparing axis runs to 20000 pulses, it is identical to the contents of data array #50 and the first DO4 pulse is set. When the comparing axis runs to 30000 pulses, it is identical to the contents of data array #51 and the second DO4 pulse is set. While comparing axis runs to 40000 pulses, it is identical to the contents of data array #52 and the third DO4 pulse is set. Once the first cycle completes, the comparing axis resets to 0 and executes PR#45 (Incremental command 50000 (5000.0) PUU), which is equivalent to a half turn of the motor. Therefore, the comparing axis outputs 5000 pulses, and after the position command completes, it executes the Speed command. Then the next comparing cycle starts. This is the same as the first cycle, and the comparing axis outputs DO4 signal at 20000, 30000, and 40000 pulses respectively and then it resets to 0 and executes PR#45.

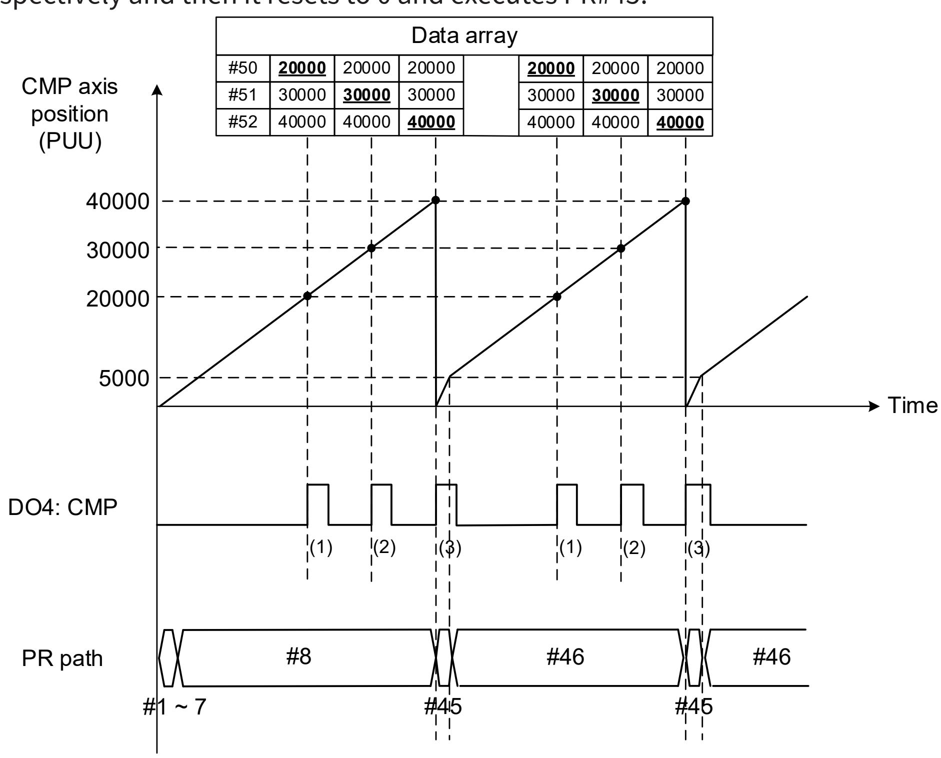
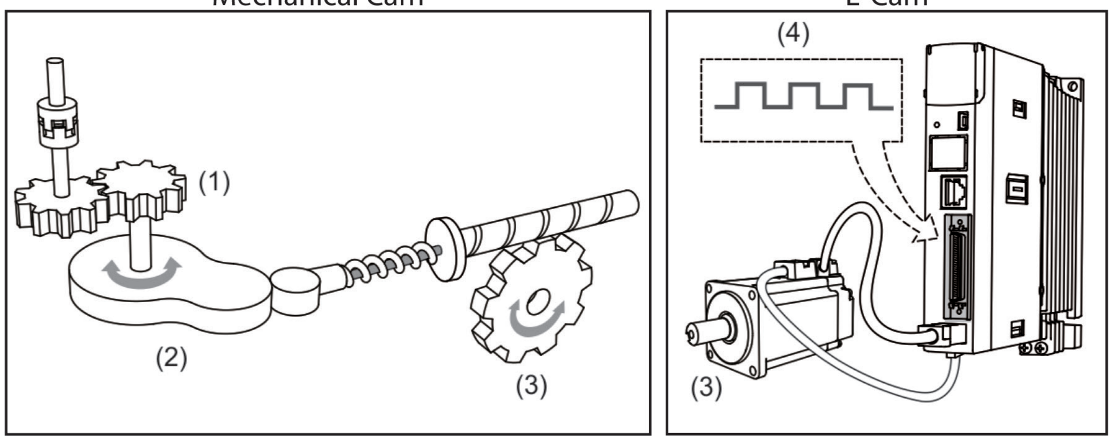


Figure 7-68 Timing of the Compare function

7.3 - E-Cam

The E-Cam system uses the mathematical formula to plan the relative following motion path based on the master-slave operation, which can replace the actual mechanical cams and overcome the limitation of the mechanical cam shapes. You can use the E-Cam function as long as it is a master-slave application and their positions can translate into a mathematical formula. The mechanical cam and E-Cam are shown in Figure 7-69.



(1) Master axis input; (2) Mechanical cam; (3) Slave axis output; (4) E-Cam master axis input

Figure 7-69 Mechanical cam and E-Cam

The E-Cam function is only available in PR mode (P1.001 = 1). The slave axis operates based on the cam curve; the positions of the master and slave correspond to a mathematical function. The master axis sends pulses to the slave axis so the slave axis runs according to the corresponding E-Cam curve, as shown in Figure 7-70. P5.088.X can enable or disable the E-Cam function. When this function is enabled, the servo drive determines the clutch engagement and disengagement timings. Figure 7-71 introduces the E-Cam parameters with a mechanical cam illustration. See the detailed settings in the following section.

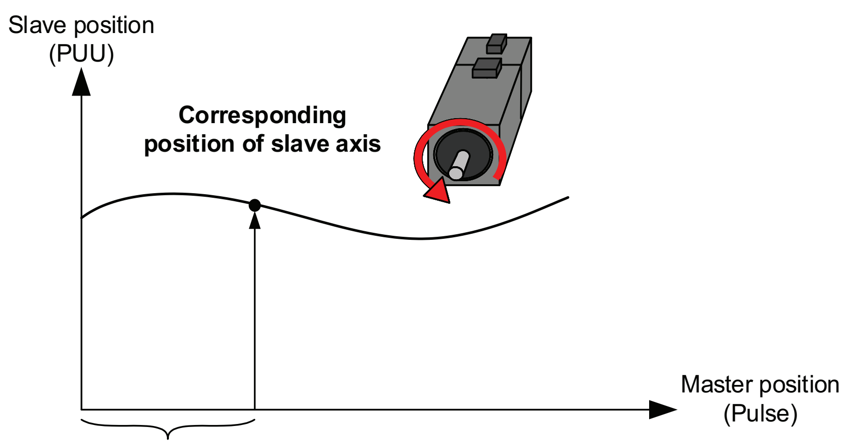
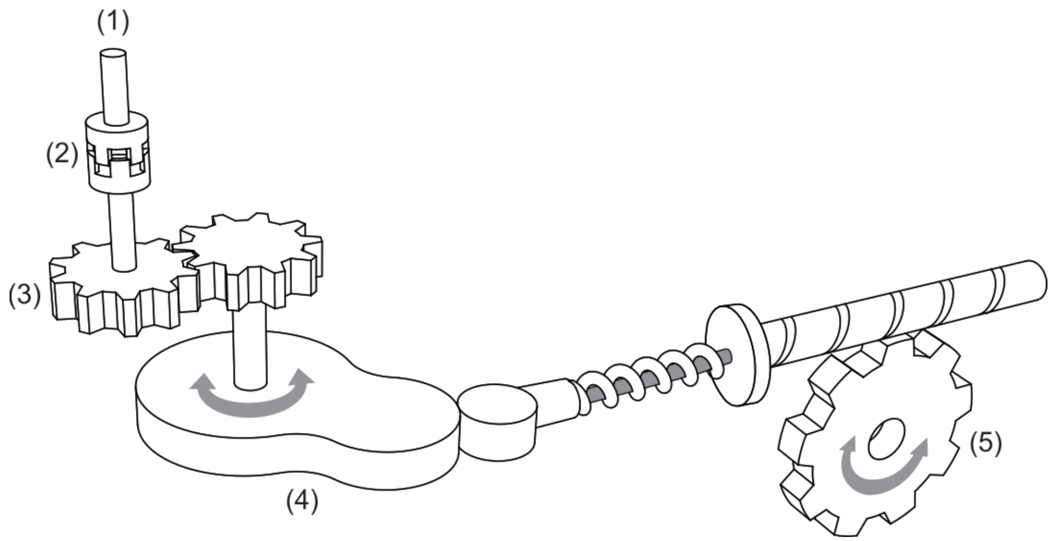


Figure 7-70 E-Cam curve



(1) Master axis: P5.088.Y: command source that acts as the master axis
(2) Clutch: P5.088.UZ, P5.087, and P5.089: engagement and disengagement timing control
(3) E-Gear of master axis: P5.083 and P5.084: command pulse resolution
(4) E-Cam curve: P5.081, P5.082, and P5.085: position correlations of master and slave axes; P5.019: scaling
(5) E-Gear of slave axis: P1.044 and P1.045: output signal resolution

Figure 7-71 Use E-Cam servo parameters to simulate mechanical cams

7.3.1 - Source signal for the master axis

When using the E-Cam function, you must specify the signal source for the master axis signaling, which can be an external encoder, PLC, or another servo drive to name a few. The SureServo2 servo drives support seven source types for the master axis. You can use P5.088.Y to select the master axis signal source and monitor the master axis position with P5.086.

1) Capture axis: when P5.088.Y = 0, it uses the source set in P5.039.Y (Capture function source) as the master axis signal source. You can read the value of P5.037 to display the axis position of Capture pulse source.
2) Encoder: when P5.088.Y = 1, it uses CN5 external encoder signals as the master axis signal source. You can read the value of P5.017 to acquire the master axis position.
3) Pulse input: when P5.088.Y = 2, it uses pulses input through CN1 as the master axis signal source. You can read the value of P5.018 to acquire the master axis position.
4) PR command: when P5.088.Y = 3, it uses the PR motion control command as the master axis signal source.
5) Time axis (1ms): when P5.088.Y = 4, it uses the pulse signal generated per millisecond from the servo drive as the master axis signal source.
6) P5.088.Y = 5 is Reserved.
7) Analog speed channel: when P5.088.Y = 6, it uses the analog speed command as the master axis signal source; 10V corresponds to the frequency of 1 M pulse/s.

SureServo2 provides two sets of DO, [0x18] CAM_AREA1 and [0x1A] CAM_AREA2, which specify the current E-Cam operation position (in respect of the master axis). The first set is set by P5.090 and P5.091; the second set is set by P2.078 and P2.079, as shown in Figure 7-72. For detailed settings, refer to Chapter 8.

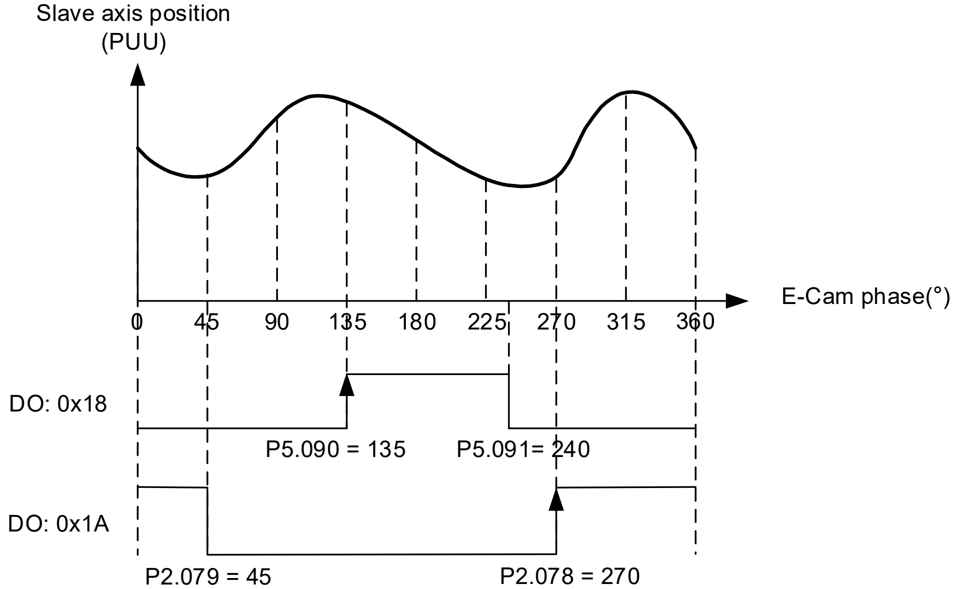


Figure 7-72 DO is on when the clutch engages

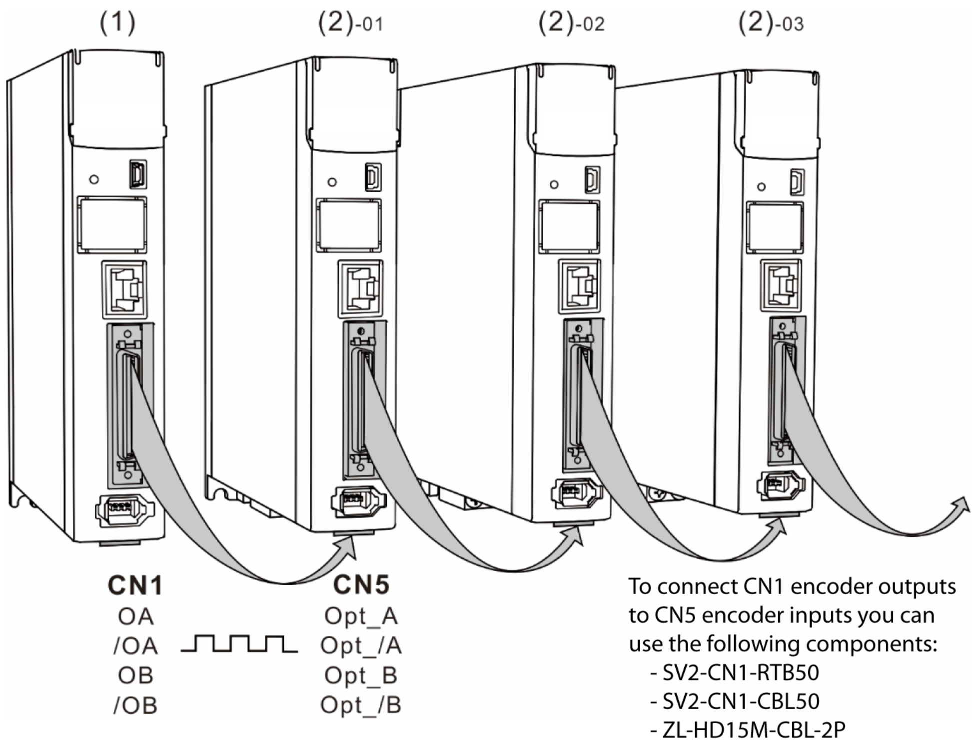
To get the data for the master axis, use four monitoring variables, which are Accumulative pulse of master axis, Incremental pulse of master axis, Pulse of master axis (lead pulse), and Position of master axis. The following is the detail description for the four monitoring variables.

1) Accumulative pulse of master axis: monitoring variable code 059(3Bh); the accumulative pulse number of the E-Cam master axis. Same as P5.086.
2) Incremental pulse of master axis: monitoring variable code 060(3Ch); the incremental pulse number of the E-Cam master axis generated per 1ms.
3) Pulse of master axis (lead pulse): monitoring variable code 061(3Dh); when the clutch is engaged, the master axis disengagement pulse number (P5.089) decrements to 0 and then the clutch disengages; when the clutch is disengaged, the master axis lead pulse number (P5.087 or P5.092) decrements to 0 and then the clutch engages.
4) Position of master axis: monitoring variable code 062(3Eh); the position of the E-Cam master axis.

Pulse Bypass Function

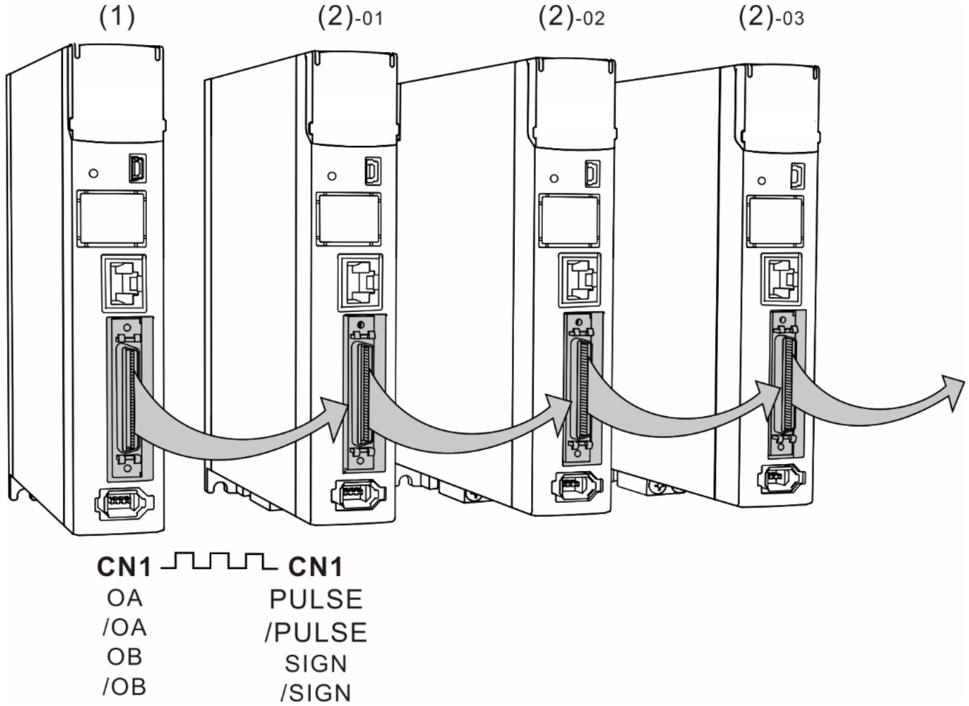
When using the E-Cam and pulse bypass functions, the servo drive can receive pulse signals and send these signals to the next servo axis, so multiple slave axes can refer to the same master axis signals. In addition, signals transmitted through the servo drives are not attenuated because the servo drive amplifies the signals to the strength they should have during output. For example, if the signal input is 4.5V, it becomes 5V when output. Since there is electrical resistance in the wire, take the signal attenuation into account and use twisted-pair shielded wires. If the signals transmitted to the servo drive are attenuated to the level that the servo cannot identify, use a cable with thicker gauge or a shorter signal cable. If not considering the signal delay caused by cables, the delay time of each servo drive is 50ns.

On the SureServo2 servo drives, the pulse output pins are OA, /OA, OB, and /OB of CN1 only; pulses can be input through CN1 or CN5 to the servo drive. Use P1.074.Y to set the output signal source for the servo drive. If selecting CN5 as the pulse input channel, as shown in Figure 7-73, then you must set P1.074.Y to 1 for each slave axis (servo drive) to have CN5 receive pulses. If selecting CN1 as the pulse input channel, as shown in Figure 7-74, then you must set P1.074.Y to 2 for each slave axis (servo drive) to have CN1 receive pulses.



(1) Master axis; (2) Slave axes 1, 2, and 3.

Figure 7-73 Pulse bypass function: CN1 output / CN5 input



(1) Master axis; (2) Slave axes 1, 2, and 3.

Figure 7-74 Pulse bypass function: CN1 output / CN1 input

7.3.2 - Clutch engagement and disengagement

When the E-Cam function is enabled, the E-Cam clutch status determines whether the slave axis operates based on the signals sent from the master axis. While cams are engaged, the slave axis operates according to the received master axis pulses and the E-Cam curve; when cams are disengaged, the slave axis does not operate according to the E-Cam curve even if it receives the master axis pulses. The clutch engagement and disengagement timings are described as follows. Note that if the drive is not in PR mode, then E-Cam will not engage.

Clutch engagement condition

After the E-Cam function is enabled, the slave axis operates according to the master axis signals and E-Cam curve only when the clutch is engaged as shown in Figure 7-75. The timing for clutch engagement can be specified with P5.088.Z. The SureServo2 provides three condition options for clutch engagement timing:

1) Engage immediately (P5.088.Z = 0): the clutch engages immediately as soon as the E-Cam function is enabled. The slave axis operates according to the E-Cam curve and the master axis signals.
2) Engagement control with DI (P5.088.Z = 1): trigger DI.CAM[0x36] to have the clutch engaged. When this DI is triggered, the clutch remains engaged until the disengagement condition is met.
3) Engagement control with high-speed capturing (P5.088.Z = 2): when the master axis source is the Capture axis and the first position data is captured and the clutch is engaged immediately. The Capture signal is triggered by DI7. Different from triggering DI.CAM[0x36] for clutch engagement, the high-speed capturing function only takes 5 µs to have the clutch engaged using the DI7, making the system timing control more precise. When the Capture function is engaged, DI7 automatically changes functions to the Capture trigger.

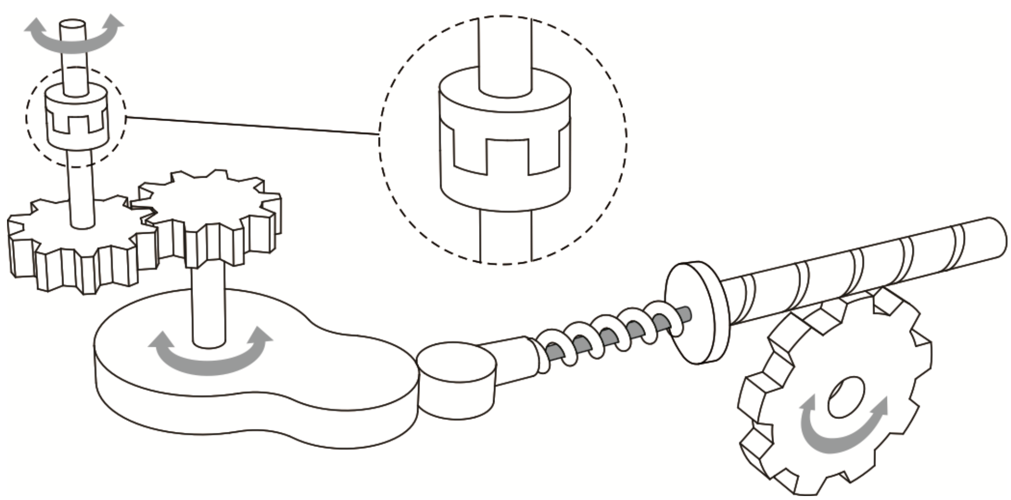


Figure 7-75 Clutch engagement

In addition, you can use P5.087 to set the initial lead pulse of the master axis before engagement. That is, once the engagement condition is met, the master axis needs to reach the set lead pulse number first and then the clutch is engaged as shown in Figure 7-76.

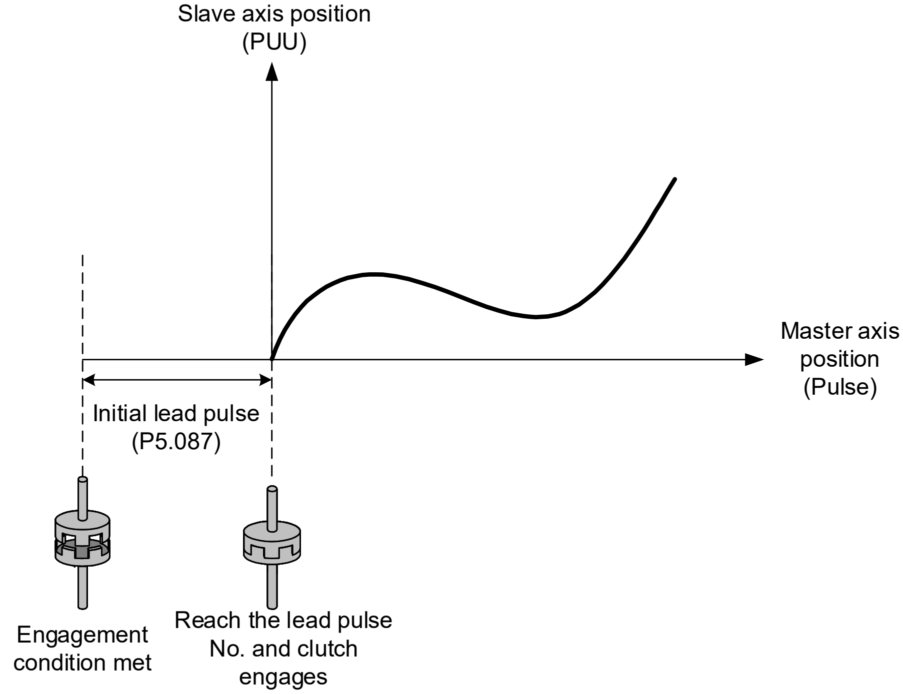


Figure 7-76 Initial lead pulse before clutch engagement

Clutch disengagement condition

When the E-Cam function is enabled and the clutch is engaged, the slave axis operates based on the E-Cam curve and master axis signals. When the slave axis completes the motion, you can directly disable the E-Cam function or disengage the clutch to stop the slave axis motion. While the clutch is disengaged, the slave remains stationary regardless of the master axis motion as shown in Figure 7-77.

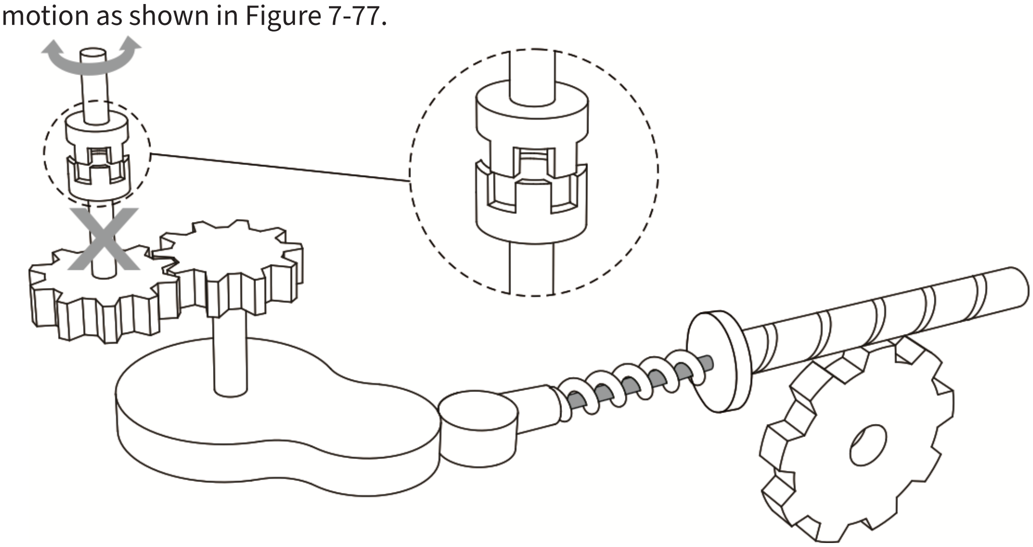


Figure 7-77Clutch disengagement

You can use P5.088.U to select the disengagement condition depending on the applications. The drive provides five condition options for clutch disengagement timing.

1) Remain engaged (P5.088.U = 0): the clutch remains engaged unless the E-Cam function is disabled.
2) Disengagement control with DI (P5.088.U = 1): switch the DI.CAM[0x36] to off to have the clutch disengaged. When this DI remains off, the clutch remains disengaged and the E-Cam system is in stop state.
3) Immediate stop after disengagement (P5.088.U = 2): when the clutch is engaged and the pulse number of the master axis reaches the value set in P5.089, the clutch disengages, the slave axis stops immediately, and the E-Cam system is in stop state as shown in Figure 7-78. This function is suitable for applications that require the slave axis to accurately stop at the specified position.

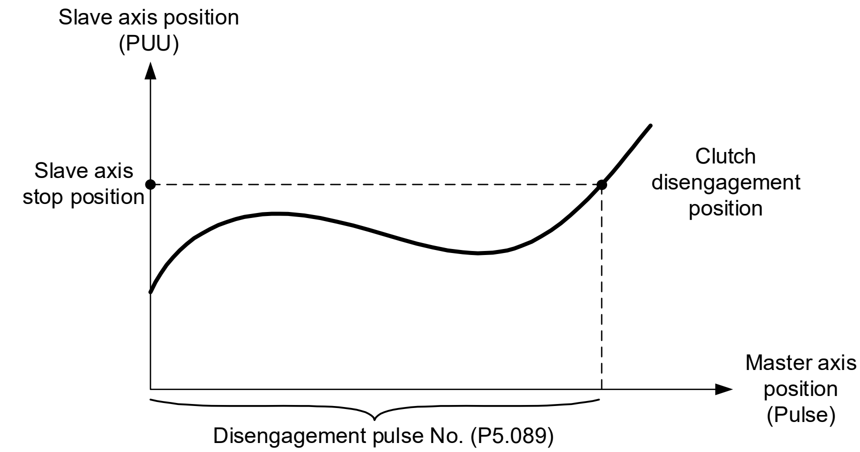


Figure 7-78 Disengagement timing: slave axis stops right after clutch disengagement

4) Decelerate to stop after disengagement (P5.088.U = 6): when the clutch is engaged and the pulse number of the master axis reaches the value set in P5.089, the clutch disengages, the slave axis decelerates to stop, and the E-Cam system is in stop state as shown in Figure 7-79. This function is suitable for applications that require the slave axis to slowly decelerate to stop.

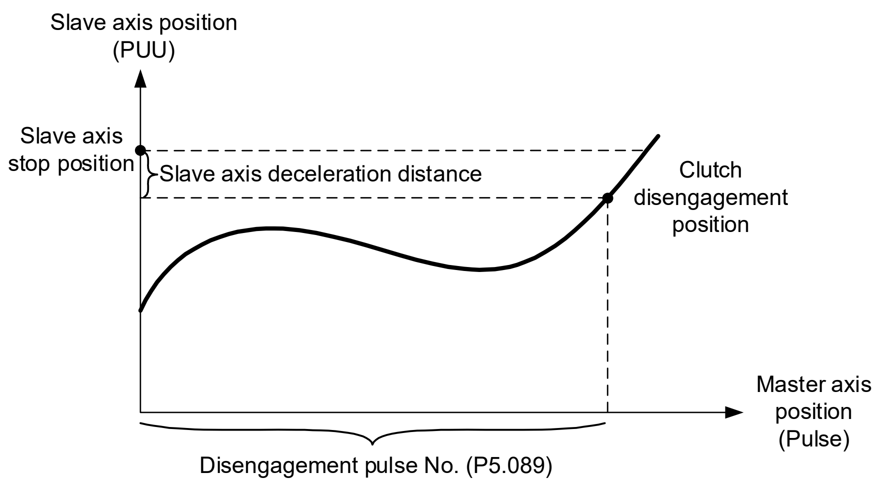


Figure 7-79 Disengagement timing: slave axis decelerates to stop after clutch disengagement

5) Enter cyclic mode after disengagement (P5.088.U = 4): when the clutch is engaged and the pulse number of the master axis reaches the value set in P5.089, the clutch disengages and the master and slave axes enter the cyclic mode. Then, the E-Cam system goes into the pre-engage state and waits for the master axis pulse to reach the number set in P5.092. Next, the clutch re-engages and operation of the next cycle starts. See Figure 7-80.

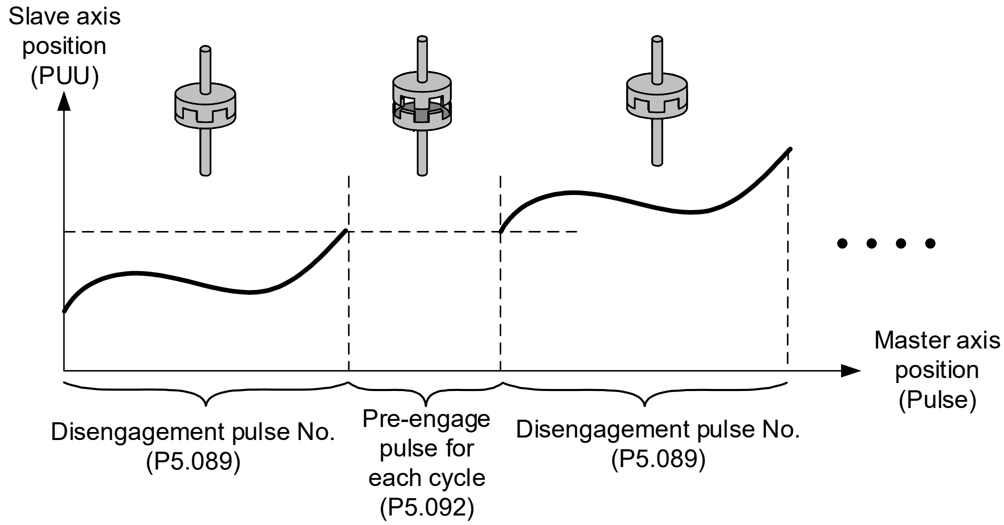


Figure 7-80 Disengagement timing: cyclic mode

NOTE: The “pre-engage pulse for each cycle” and the “initial lead pulse” are different. The “initial lead pulse” is valid only for the first engagement whereas the “pre-engage pulse for each cycle” is effective before each engagement cycle. You can see how these two work together in Figure 7-81.

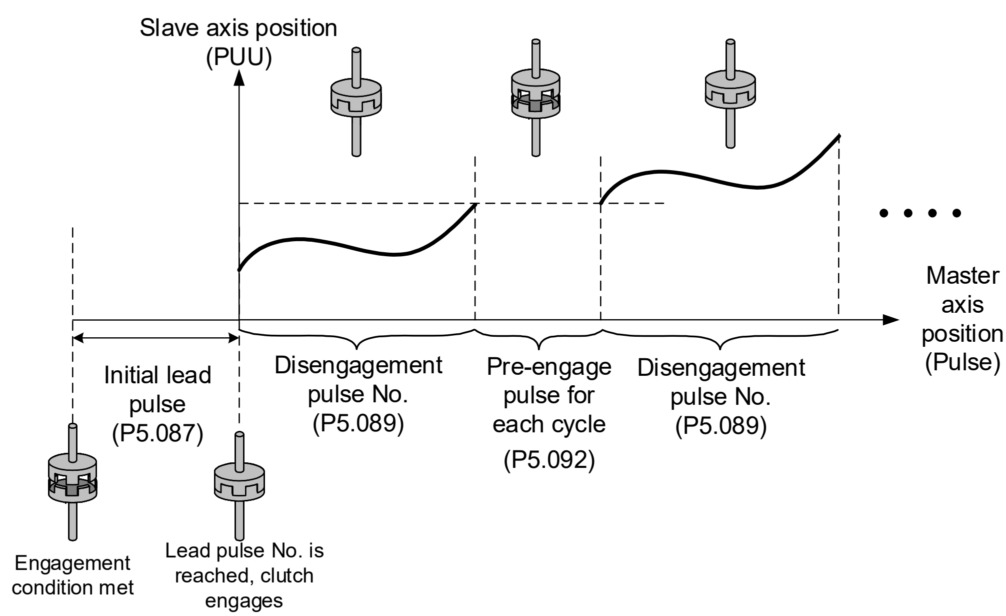


Figure 7-81 “Initial lead pulse” and “Pre-engage pulse for each cycle”

There are three available options for how the E-cam behaves after the clutch disengagement condition has been met (choose one of the following): “Immediate stop after disengagement”, “Decelerate to stop after disengagement”, or “Enter cyclic mode after disengagement”. If you select the disengagement options such as “Disengagement control with DI”, “Immediate stop after disengagement”, or “Decelerate to stop after disengagement” (P5.088.U = 1, 2, or 6), you can activate the function for disabling the E-Cam after clutch disengagement (P5.088.U = 8). This is the same as setting P5.088.X to 0; however, you cannot set it individually; you need to use one of the three options (P5.088.U = 1, 2, or 6) when setting P5.088.U to 8. You can stop the slave axis by disengaging the clutch or disabling the E-Cam. Regardless of the current state of the E-Cam system (stop, engaged, or disabled), you need to enable the E-Cam function to operate it. When the clutch is disengaged, although the slave axis is stopped, the slave axis continues to monitor the signals sent from the master axis as the E-Cam system remains operating. Settings for clutch disengagement timing and disabling E-Cam function are as follows:

P5.088.U value	*Clutch disengagement condition*	*System status after disengagement*
0	Condition 0: remains engaged unless E-cam function is disabled	-
1	Condition 1: DI.CAM Off. Disengages when DI (DI: 0x36) is off	0: stop
2	Condition 2: disengages when the master axis pulse number reaches the setting value of P5.089, and the slave axis stops immediately (sign indicates the direction)	0: stop
4	Condition 4: disengages when the master axis pulse number reaches the setting value of P5.089 and the master and slave axes enters the cyclic mode. When the pre-engaged pulse number for each cycle (P5.092) is reached, the clutch re-engages	2: pre-engage
6	Condition 6: disengages when the master axis pulse number reaches the setting value of P5.089, and the slave axis decelerates to stop	0: stop

NOTE: For the system status after disengagement, refer to the following sections.

You can choose one of the three disengagement conditions for the PR path after the clutch disengagement, which are “2: Immediate stop after disengagement”, “6: Decelerate to stop after disengagement”, and “4: Enter cyclic mode after disengagement” (P5.088.U = 2, 6, or 4). Write the PR number in hexadecimal to P5.088.BA. If this value is 0, it means no PR path is executed after the disengagement. In addition, if you use the setting “Enter cyclic mode after disengagement (P5.088.U = 4)” and specify the following PR path, as the E-Cam function does not have an interruption setting, the slave axis carries on to the next cycle until the motion set in the PR path is complete.

E-Cam system status

The E-Cam system has three states once the E-Cam is enabled, Stop, Engage, and Pre-engage. When the E-Cam function is enabled, you can use P5.088.D to promptly monitor the system's current status. The following section explains each state, as shown in Figure 7-82.

1) Stop state (P5.088.D = 0): the clutch is disengaged and the system continues to check the engagement condition (P5.088.Z). If the engagement condition is met and the initial lead pulse (P5.087) is not set, the clutch engages. If you have set the initial lead pulse, the system enters the Pre-engage state. When the E-Cam function is disabled, the system is also in the stop state.
2) Engage state (P5.088.D = 1): the clutch is engaged and the system continues to check the disengagement condition (P5.088.U). If one of the three disengagement conditions, "Disengagement control with DI", "Immediate stop after disengagement, or "Decelerate to stop after disengagement" (P5.088.U = 1, 2, or 6) is met, the system is stopped. If the condition "Enter cyclic mode after disengagement (P5.088.U = 4)" is met, the system enters the Pre-engage state.
3) Pre-engage state (P5.088.D = 2): the clutch is disengaged. If pulses from the master axis reach the initial lead pulse number or the pre-engaged pulse number for each cycle, the clutch engages and the system enters the Engage state.

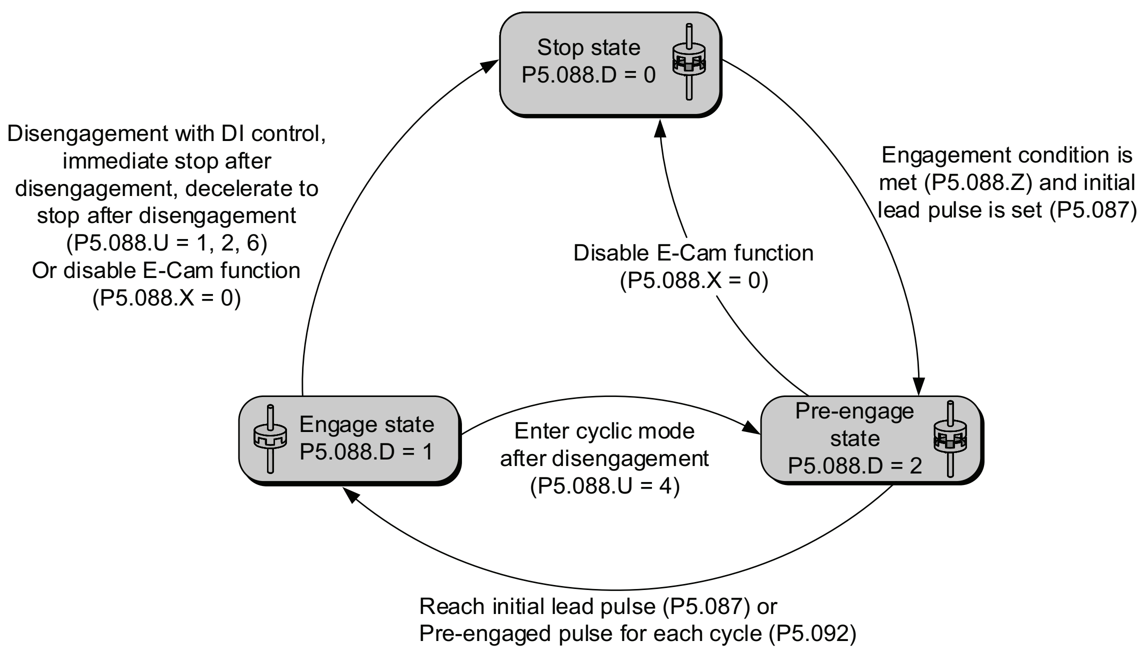
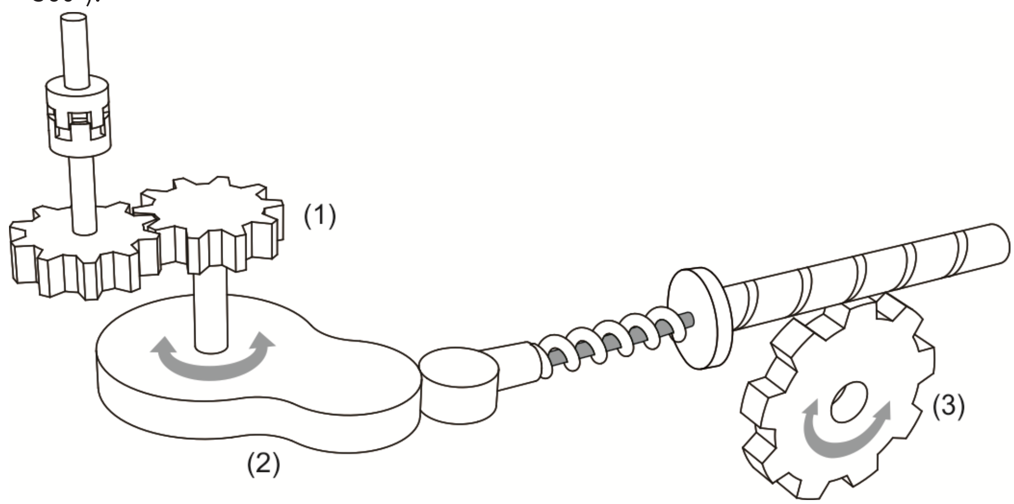


Figure 7-82 E-Cam system status

7.3.3 - E-Cam gears and curve scaling

In the E-Cam system, two sets of electronic gearing can determine the E-Cam motion, which are electronic gearing of the master axis and electronic gearing of the slave axis. The electronic gearing of the slave axis determines the motion ratio of all motion commands and is determined by P1.044 and P1.045. Changing this electronic gearing ratio will change the E-Cam motion and motion commands in PT and PR modes as well. Therefore, if you need to change the E-Cam curve scaling, changing P1.044 and P1.045 is not suggested. Use P5.019 to scale this electronic gear ratio.

The electronic gearing of the master axis is only for the E-Cam system and can change the pulse command resolution of the master axis. The Master axis E-Gear parameters are P5.083 and P5.084. When the slave axis receives the pulse number defined by P5.084 from the master axis, E-Cam rotates the number of cycles defined by P5.083 (one cycle of E-Cam = rotate from 0° – 360°).



(1) Electronic gearing of master axis: P5.083 and P5.084 for command pulse resolution. This is how many pulses you want to have for one E-Cam cycle.
(2) Electronic gearing curve: P5.019 for scaling
(3) Electronic gearing of slave axis: P1.044 and P1.045 for output signal resolution

Figure 7-83 E-Cam gear ratio

The following example illustrates how the command resolution is adjusted: assume that the original master axis pulse number for one cycle is 10000 pulses as shown in Figure 7-84. If this master axis E-Gear ratio becomes larger (P5.084 increases or P5.083 decreases), then the master axis pulse unit corresponds to a narrower E-Cam phase, making the master axis pulse command resolution higher. When the master axis E-Gear ratio becomes smaller (P5.084 decreases or P5.083 increases), the master axis pulse unit corresponds to a wider E-Cam phase, making the master axis pulse command resolution lower. In common applications, P5.083 is set to 1 and P5.084 is for specifying the required master axis pulse number for the E-Cam to operate one cycle. If the required master axis pulse number has decimal places, you can adjust the value of P5.083. For example, when the required master axis pulse for operating one cycle is 517.5, you can set P5.083 to 2 and P5.084 to 1035.

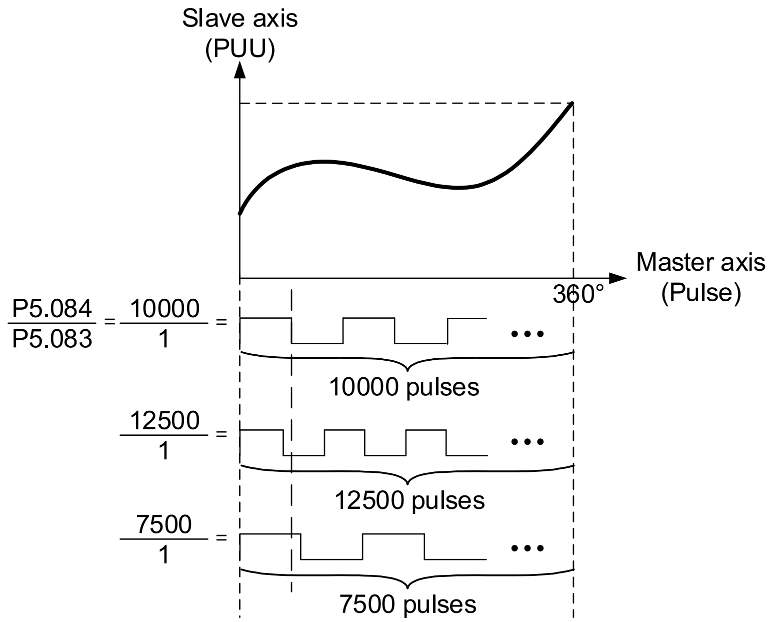


Figure 7-84 E-gear ratio for E-Cam

To change the slave axis motion path proportionally in E-Cam applications, it is suggested to use P5.019 to scale the E-Cam curve. This parameter is only effective to the E-Cam system and does not affect other motions in the servo system. As shown in Figure 7-85, if P5.019 = 2, the output curve of the slave axis is two times of the original. If P5.019 = 0.5, the output curve of the slave axis is 0.5 times of the original. If P5.019 = -1, the positive and negative outputs of the slave axis are reversed.

P5.088.X Bit 2 can specify the effective timing for the E-Cam curve scaling with the options of taking effect immediately and after clutch re-engagement. For example, after adjusting the cutting length in flying shear applications, you can use this parameter to determine when to make the modified E-Cam curve scaling take effect. If the clutch is set to remain engaged and modifying the cutting length is required, set P5.088.X Bit 2 to 1 (modified E-Cam curve scaling is effective immediately).

If you set P5.088.X Bit 2 to 0 to have the modified E-Cam curve scaling take effect upon the next engagement, then the cutting length changes upon the next clutch engagement. For details about flying shear applications, refer to Section 7.3.8.

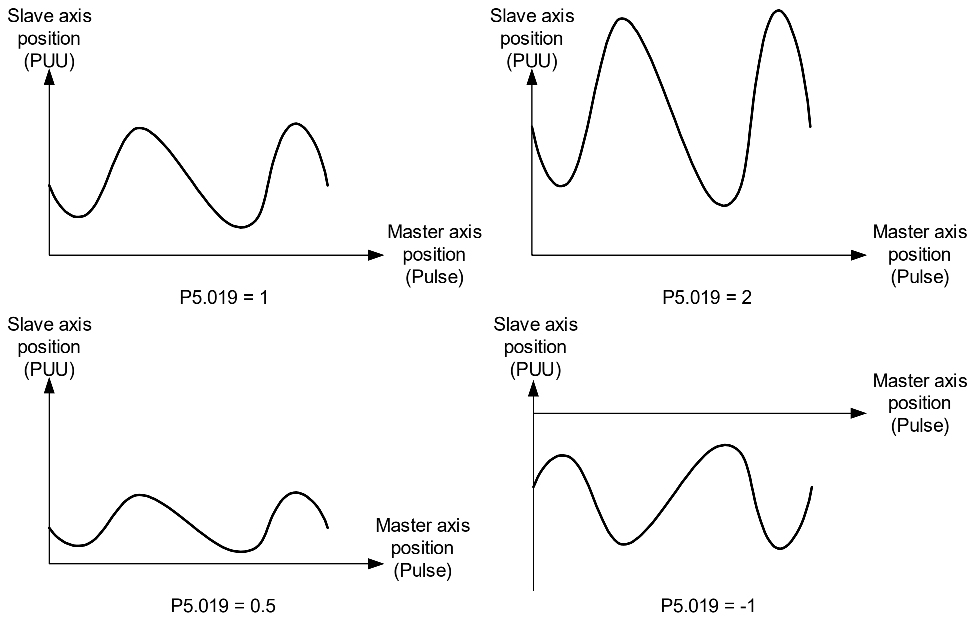
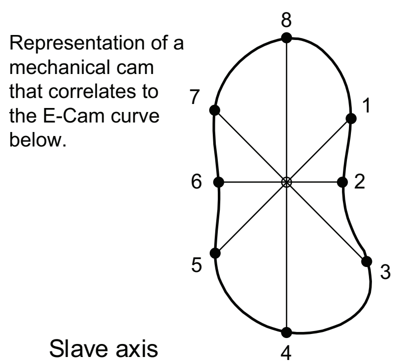


Figure 7-85 E-Cam curve scaling

7.3.4 - E-Cam curve

E-Cam curve is created by the mathematic function based on positions of the master and slave axes. There are various ways to create the table. You can use other 3rd party mathematic tools (software) or use the tabulation interface for industry-specific applications provided in SureServo2 Pro in the E-CAM editor window. Regardless of the tabulation methods, the software coverts the mathematic function into position data and stores them in the data array. To enter an E-Cam table made from 3rd party software, right-click on the Cam table in the E-Cam Editor window in SureServo2 Pro and select “Import Points”. One E-Cam curve can have up to 721 sets of data (divided 720 times). It means the highest resolution is 0.5 degrees. As long as the total point number is within the data array’s maximum 800, the array can store multiple E-Cam curves. The slave axis curve between two data points will be interpolated with a cubic curve to smooth the motion at each point.

See Figure 7-86 for example. If using E-Cams to replace mechanical cams, divide the mechanical cam into several segments. The more the segments, the higher the precision. In this example, the cam is divided into 8 segments, and each interval is 45 degrees (this is for reference only; you need to delicately segment the cam in the actual application, or the path will be distorted). Then, enter the distance between the cam shaft and the points (1 – 8) on the cam edge to the data array. The start point 0 degree and the last point 360 degrees are identical but you must enter both of the two points to the data array so the E-Cam can completely go around the mechanical cam for one cycle. Therefore, you need to enter 9 sets of data to complete the E-Cam curve table.



Point	Data array	
1	#50	30000
2	#51	20000
3	#52	40000
4	#53	45000
5	#54	30000
6	#55	20000
7	#56	20000
8	#57	50000
1	#58	30000

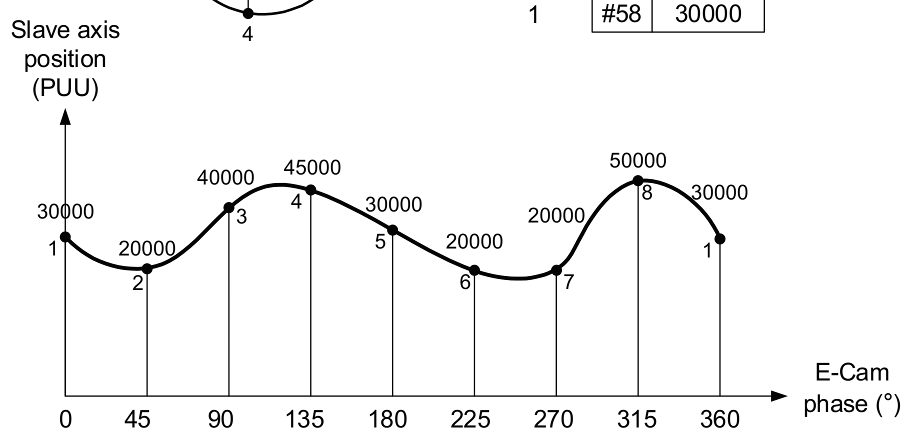


Figure 7-86 Example of creating E-Cam curve table

You can use SureServo2 Pro to create the E-Cam curve. Click **E-CAM** in the software function list and the E-CAM Editor appears, as shown in Figure 7-87.

In the first page of the editing window, select the method to create the E-Cam curve table. There are seven options, [Manual], [Speed Fitting], [Rotary Shear - W/O Sync. Zone], [Rotary Shear - Fixed Sync. Zone], [Rotary Shear - Adjustable Sync. Zone], [Cubic Curve], and [Rotary Shear - Printing Machine].

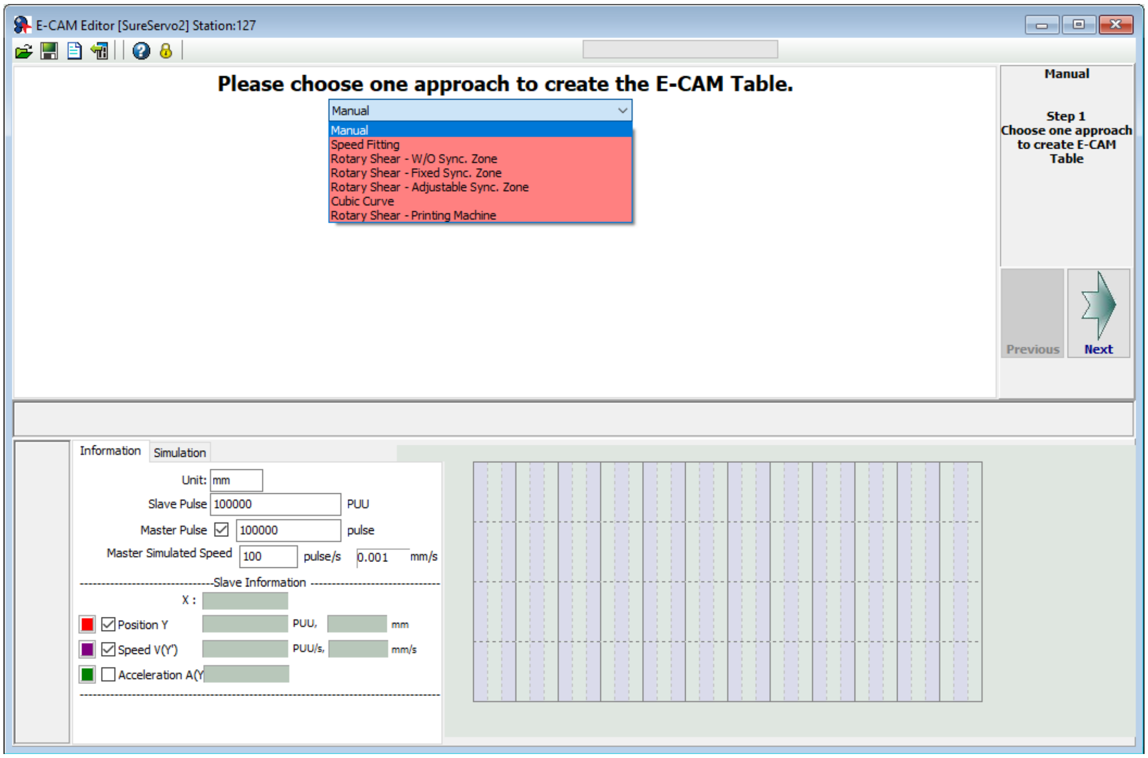


Figure 7-87 SureServo2 Pro E-Cam setting interface

Manual

When you use other software to create the table, the software presents the final results with position data and then imports these data to the table to complete the E-Cam curve. As the example shown in Figure 7-86, the mechanical cams can be replaced by E-Cams. It creates the E-Cam curve by using the angles of the mechanical cams corresponding to the distances between the cam shaft and edge, which is to establish the correlation of the angles and slave axis positions. The setting interface for manually creating E-Cam tables in SureServo2 Pro is shown in Figure 7-88. The following illustrates the steps to manually create the table:

1) Set the E-Cam segment number: an E-Cam can be divided into up to 720 segments (721 points). For a 360-degree cycle, every 0.5 degrees corresponds to a slave axis position. The more the points, the higher the resolution and the more delicate the curve. Appropriately allocate the resources for the curve resolution and data array to set the E-Cam segment number.
2) Create curve table: after setting the E-Cam segment number, click Create Table, and the software equally segments the 360-degree E-Cam and automatically fills in the angle data to the table. When you set n points for the E-Cam segment number, the table has n+1 columns.

3) Fill in the slave axis positions: fill in the corresponding position in PUU for each segment. Click Draw. The software automatically generates a simulated E-Cam curve and cam position, speed, and acceleration curve. When manually creating the table, pay more attention to the speed continuation of the slave axis to avoid machine vibration or motor overload caused by speed discontinuation.
4) Download E-Cam curve: once the E-Cam curve is confirmed, click Download Table to write the E-Cam curve to the data array. If you click Write Table Data to EEPROM, the data array is written to EEPROM and is non-volatile.

NOTE: Unless you write to EEPROM, the E-Cam table is cleared on a power cycle and the E-Cam will not run without an E-Cam table.

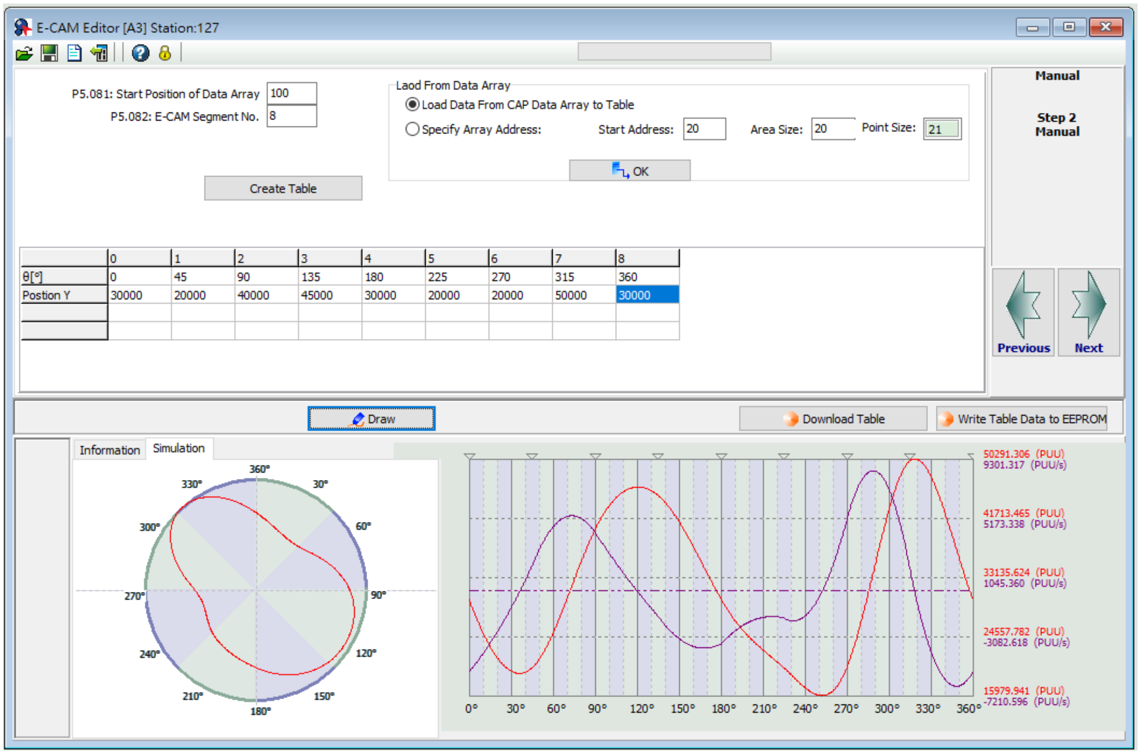


Figure 7-88 SureServo2 Pro Manual setting interface for creating E-Cam table

When using a third-party software (such as Excel) to create the table, save all position data as a text file (.txt) and use the Space, Tab, Enter keys, vertical bar "|", or a comma "," to separate the position data of each point.

If you start the E-CAM Editor in SureServo2 Pro and choose to manually create the table, first specify the number of E-Cam segments (P5.082), and click **Create Table**, then the table displays the E-Cam phase corresponding to the E-Cam segments. Right-click the table, select **Import points**, and then click **Browse** after the window for importing point data appears. Open the text file that has been saved, and select the separator you use in the saved file for **Separate symbol**. Then, click **OK** to complete loading the text file. Next, click **Draw** to have the software generate the E-Cam curve based on the E-Cam positions. You can also export the position data as a text file.

Right-click the table, and select **Batch change the values**, including increment, decrement, addition (+), deduction (-), multiplication (*), division (/), copy, and exchange functions for users to change the E-Cam curve quickly. There are also functions for inserting and deleting single position data on the right-click menu. The setting interface for creating E-Cam tables with a third-party software in SureServo2 Pro is shown in Figure 7-89.

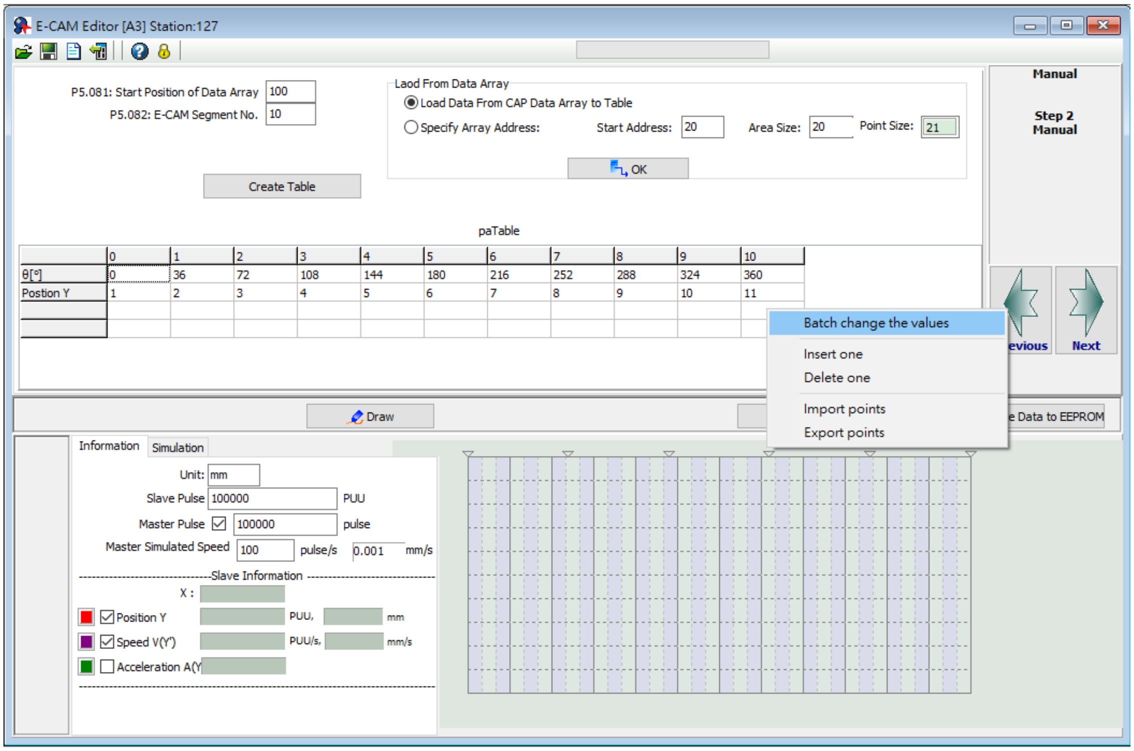


Figure 7-89 Use third-party software to create E-Cam curves

Speed Fitting

When the application requires the motions of both the master and slave axes to keep the same speed or have the same correspondence relations, you can use the speed fitting method to create the E-Cam curve. With this method, the E-Cam cycle is divided into five zones, which are the waiting, acceleration, synchronous speed, deceleration, and stop zones as shown in Figure 7-90. You can adjust the proportion of each zone as needed. All five zones added up must equal 100%. This is 100% of the 360 degree Cam profile.

This E-Cam curve is designed based on the positions. It plans the corresponding speed of the master and slave axes based on the relationship between the position change and speed within a given time. The setting interface for creating E-Cam curves with the speed fitting method in SureServo2 Pro is shown in Figure 7-91. Steps to create the table with the speed fitting method are as follows:

1) Plan the E-Cam curve: determine the proportions of the waiting, acceleration, synchronous speed, deceleration, and stop zones within an E-Cam cycle.
2) Set the total moving distance (lead): set the total moving distance of the slave axis within one cycle in the unit of PUU.
3) Set the S-curve: set smoothness at the transition points of the position curve. The higher the value, the smoother the motor acceleration or deceleration, and the longer the operation cycle. The S-curve setting value is usually the same or smaller than the stop zone point number. See section 6.2.4 for more on S-curve settings.
4) Download E-Cam curve: once the E-Cam curve is confirmed, click Download Table to write the E-Cam curve to the data array. If you click Write Table Data to EEPROM, the data array is written to EEPROM and is non-volatile.

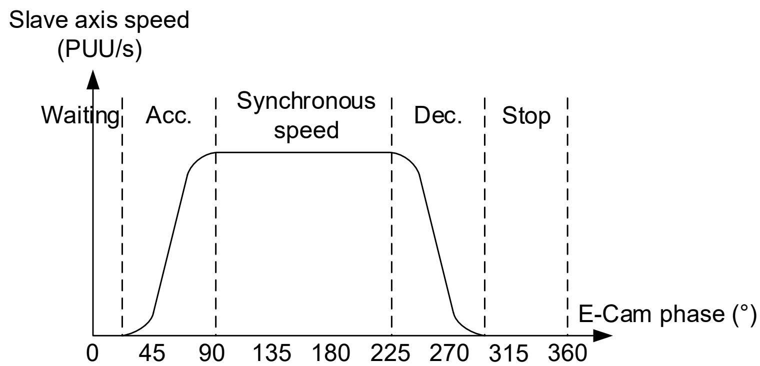


Figure 7-90 Speed fitting zone definition

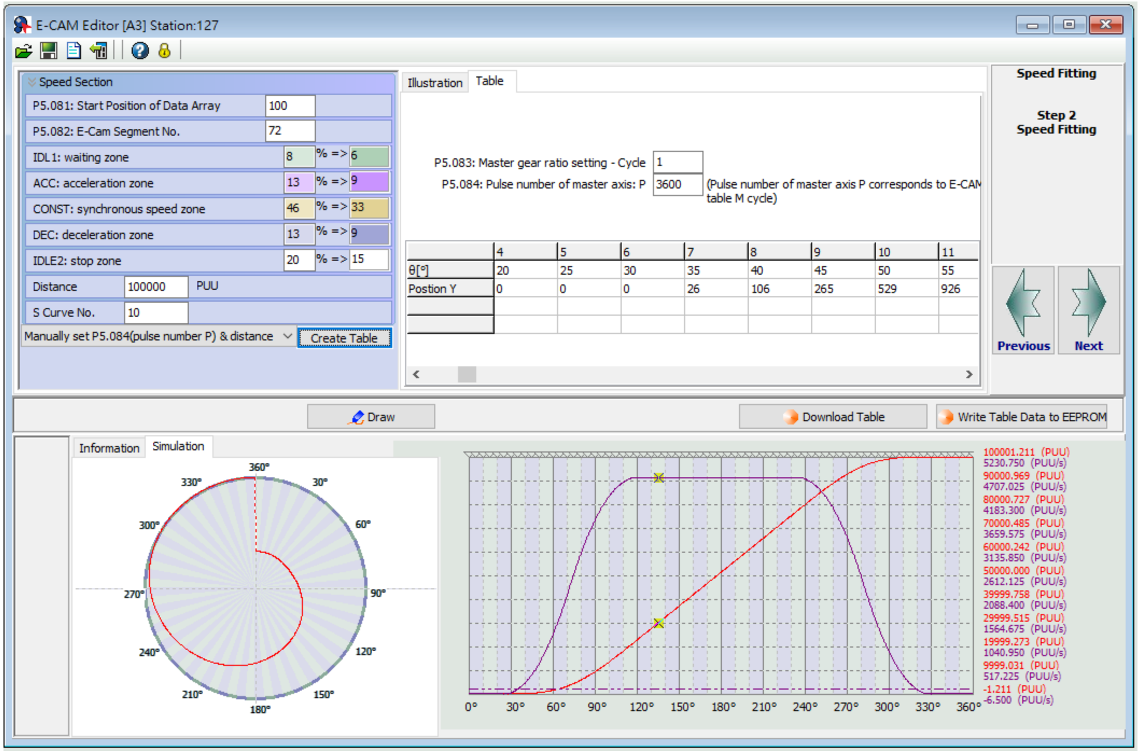


Figure 7-91 SureServo2 Pro Speed fitting setting interface for creating E-Cam table

Cubic Curve

If the master and slave axes operate only based on the corresponding positions, such as the point-to-point relation, you can use the cubic curve method to create an E-Cam curve. If using the cubic curve method to create the table, simply enter the E-Cam phase and the corresponding slave axis positions to have the tabulation tool automatically connect the points and optimize the curve. In some applications, you might need a point-to-point cam motion trajectory such as a constant line or curve, then you can use the cubic curve method to modify the curve, and set the start angle N1 (the angle departing from the start point) and the end angle N2 (the angle arriving at the target point) as needed, as shown in Figure 7-92. There are three types of curves for creating the table:

1) Constant speed: a constant-speed linear trajectory connecting two sets of cam point data; the start and end angles are unadjustable.
2) Uniform acceleration: a uniform incremental or decremental curve in single direction. Only the start angle is adjustable.
3) Cubic curve: the start and end angles are both adjustable. Changing the angles also changes the speed when departing from the start point and arriving at the target point. Note that improper angle setting leads to drastic speed change which causes machine vibration.

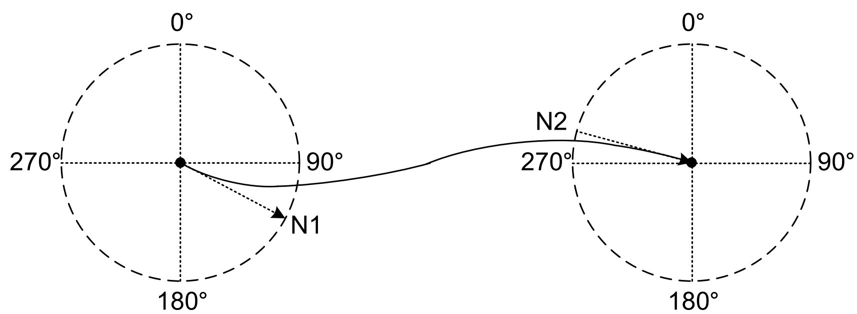


Figure 7-92 Start angle and end angle

Figure 7-93 is the cubic curve setting interface for creating E-Cam tables in SureServo2 Pro. The steps to create the table with the cubic curve method are as follows:

1) Set the E-Cam curve: set the E-Cam phase, slave axis position, curve type, start and end angles in Cubic Data. You can drag the transition points in the cubic curve simulation to change the corresponding data of each point. When you drag, insert, or delete the transition points, the corresponding cubic data contents are promptly changed. However, if you directly enter or select the desired content in the cubic data, you must click Create Cubic Curve to change the cubic curve simulation.
2) E-Cam table setting: when completing the transition point setting, set the sampling angle and click Convert to E-Cam table, so the software will fill in the sampling data to the E-Cam table. The more the sampling points, the more accurate the cam shapes. If the setting value for the slave axis is too small, vibration might occur because the decimal value is rounded off. Use P5.019, E-Cam curve scaling, to keep decimals in the table to reduce zigzags of the curve and generate an E-Cam with higher precision.
3) Download E-Cam curve: make sure the E-Cam curve is correct and click Download Table to write the E-Cam curve to the data array. If you click Write Table Data to EEPROM, the data array is written to EEPROM and is non-volatile.

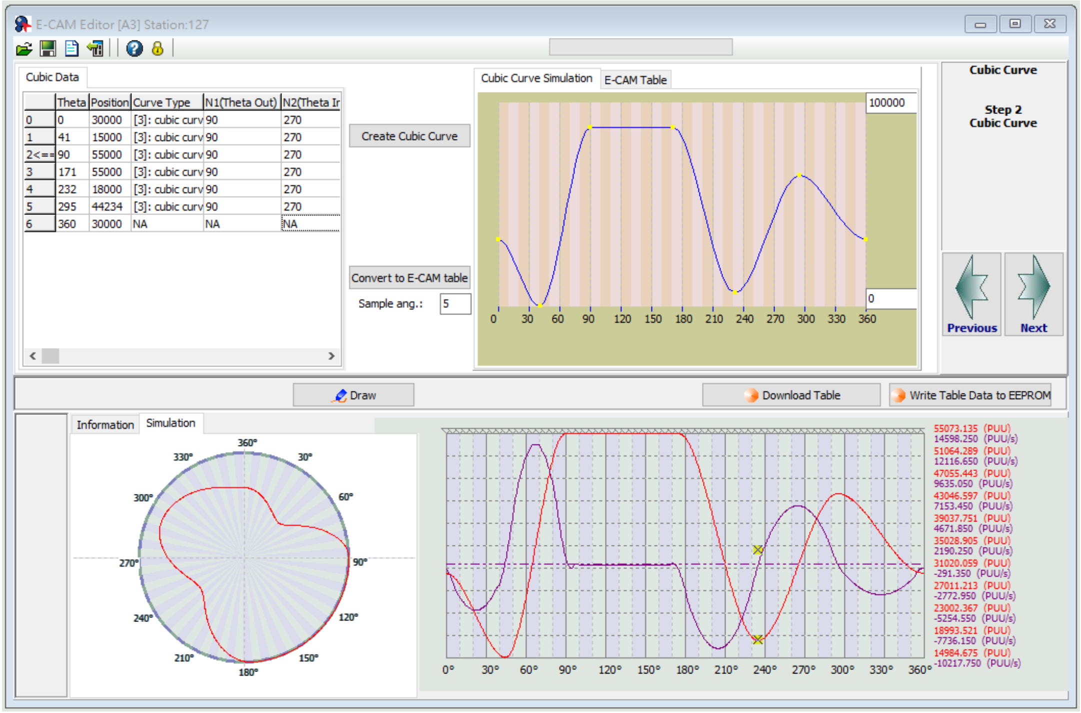


Figure 7-93 SureServo2 Pro Cubic curve setting interface for creating E-Cam table

Create E-Cam table for rotary shear applications

For rotary shear applications, SureServo2 Pro provides three methods for creating rotary shear curves, [Rotary Shear - W/O Syn. Speed Zone], [Rotary Shear - Fixed Sync. Zone], and [Rotary Shear - Adjustable Sync. Zone]. The only difference between these methods is the synchronous speed zone for the master and slave axes, which is adjusted based on the types of the machining cutter and motion. [Rotary Shear - Printing Machine] is for creating curves for printing machines. In addition, you can also use macro #7 to create E-Cam curves for the rotary shear. For detailed settings, refer to Section 7.3.7. For more informatoin on Macros, refer to section 7.3.9.

7.3.5 - E-Cam curve and PR command overlapping

When the E-Cam curve is operating, if you trigger a PR path of incremental position command, the E-Cam command is overlapped with the PR command. As shown in the upper part of Figure 7-94, the moving direction of the slave axis is the same as that set in the incremental position command. When the slave axis is moving at 300 rpm and you trigger an incremental position command with the target speed of 200 rpm in the same direction, the slave axis overlaps the PR incremental position command with the E-Cam command and completes the 5000 PUU incremental position command at the target speed of 500 rpm. As shown in the lower part of Figure 7-94, the moving direction of the slave axis is opposite to that set in the incremental position command. When the slave axis is moving at 300 rpm and you trigger an incremental position command with the target speed of 200 rpm in the reverse direction, the slave axis executes the E-Cam command at the target speed of 100 rpm. Then, it resumes the original speed after the incremental position command of -5000 PUU is executed completely.

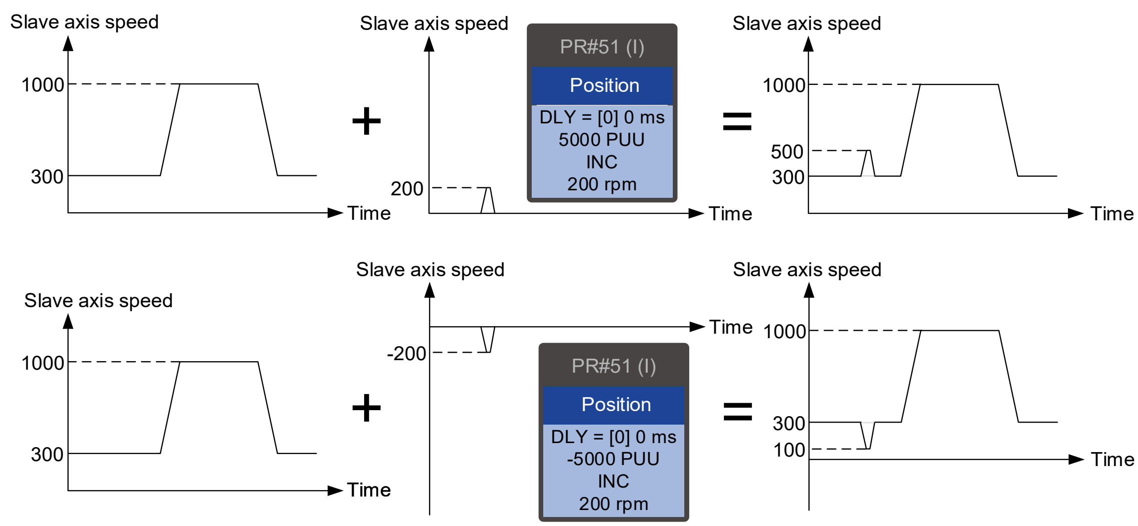


Figure 7-94 Overlapping of E-Cam command and PR incremental position command

To change the E-Cam phase when E-Cam is operating, use the PR incremental position command. Both the phase alignment function and macro for the rotary shear, which align the E-Cam phase, are completed by this command overlapping method. For more about this function, refer to Sections 7.3.7 and 7.3.9.

Take the triple-axis synchronous printing machine shown in Figure 7-95 for example. The material feeder is the master axis sending pulse signals to have the three slave axes operate based on the same E-Cam curve. The pulse bypass feature in P1.074.U allows for a pulse train to come in to one drive and be repeated back out on CN1 (OA, /OA, OB, /OB). Generally, the E-Cam phases of the three axes must be consistent. If inconsistent, use the command overlapping function to correct the E-Cam phase. To shift the phase in the forward direction, set the forward incremental command. To shift the phase in the reverse direction, set the reverse incremental command.

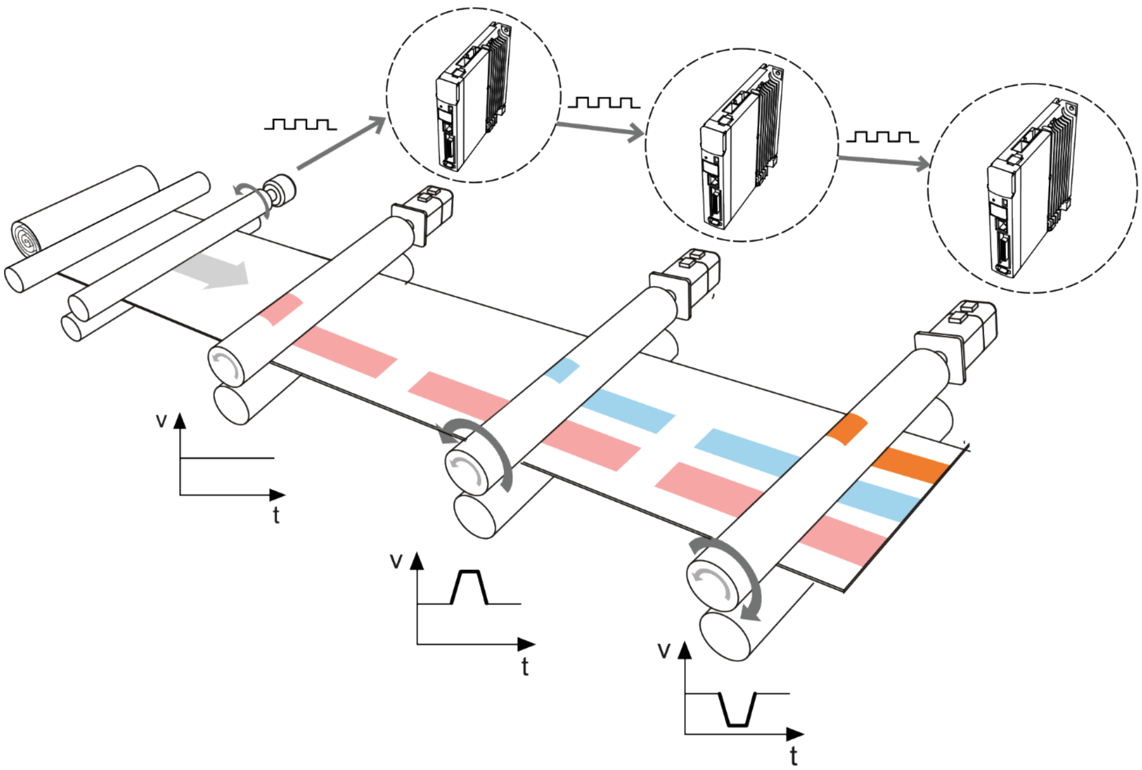


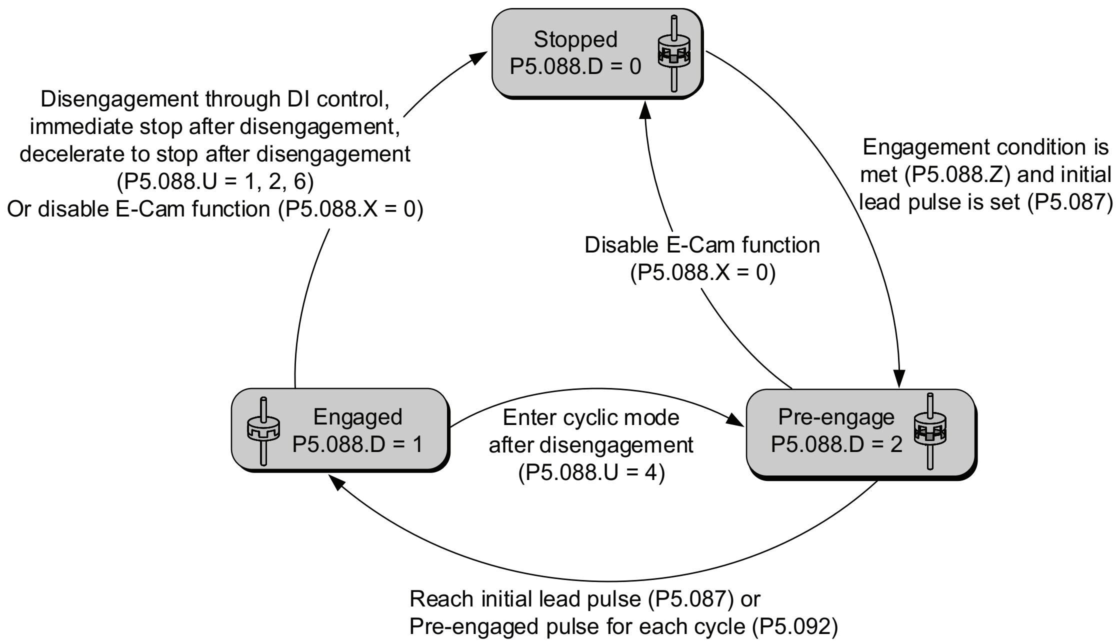
Figure 7-95 E-Cam phase alignment function

7.3.6 - Troubleshooting for E-Cam

If the E-Cam cannot operate normally, follow these steps to troubleshoot:

1) Servo drive control mode: make sure the control mode is PR mode and the system is in the Servo On state.
2) Pulse source of master axis: check the setting of P5.088.Y for the source of the master axis. Use P5.086 or monitoring variable 059 to read the pulse count of the master axis. If selecting CN1 as the input connector, use P5.018 to monitor the pulse count; if selecting CN5 as the input connector, use P5.017 to monitor the pulse count. Please note that this method is only applicable to the external input. When applying E-Cam function, it is suggested to use signal input externally as the source of the master axis; virtual signal which generates from the servo drive is for testing purposes only.
3) E-Cam curve: read the E-Cam curve data in the data array by opening the High Speed Position window and selecting the Data Array Editor tab, then click Load from Servo. Make sure the E-Cam curve is correct and check the settings for P5.081 (the start address of the E-Cam data array) and P5.082 (E-Cam segment number). The total number of points of E-Cam is the value of P5.082+1. If incorrect, the system is unable to carry out the setting of every point on the E-Cam curve. If the data is not written to EEPROM, then the data is lost on power cycle.
4) E-Cam gear ratio and scaling of E-Cam curve: check the master axis E-Gear ratio (P5.084 / 5.083) and the slave axis E-Gear ratio (P1.044 / P1.045). Check the E-Cam curve scaling (P5.019). If the proportion is set too small, the motor operation is too subtle to be monitored even when E-Cam is operating. In this case, use the scope in the SureServo2 Pro to see if the motor is slightly rotating.
5) Clutch status: read P5.088.D to obtain the current status of the clutch.
If P5.088.D = 0, it means the clutch is disengaged. Check the engagement setting (P5.088.Z). If P5.088.D = 1, it means the clutch is engaged and the slave axis operates based on the pulses from the master axis. If the disengagement condition is determined by the DI (P5.088.U = 1), check the timing for triggering the DI to on and off. If the disengagement condition is set to “Immediate stop after disengagement” (P5.088.U = 2) or “Decelerate to stop after disengagement” (P5.088.U = 6), check the pulse number of disengaging time (P5.089).
6) When E-Cam is enabled (P5.088.X = 1), the pulse number count from the master axis can be accessed by P5.086. Its value must be an increasing value. If the value is not increasing, please reverse the pulse direction (not the motor’s operating direction). Value of P5.086 has to be an increasing value; otherwise, E-Cam axis will not be able to operate based on the E-Cam curve.

7) If P5.088.D = 2, it means the clutch is in the pre-engage status. Check the setting for the initial lead pulse before engaged (P5.087). The clutch engages only when receiving the set number of pulses in the forward direction from the master axis. If the received pulses are in the reverse direction, modify the setting according to the master axis pulse source:
   a) Master axis pulse source: change the encoder output polarity for the servo drive (P1.003).
   b) Master axis pulse input from CN5: change the auxiliary encoder feedback direction (P1.074.Z).
   c) Master axis pulse input from CN1: directly modify the wirings by exchanging the wirings for the A and B phase signals.



7.3.7 - Rotary Shear

The rotary shear system is a system that combines the material feeder and cutter; the cutter cuts simultaneously when materials are fed as shown in Figure 7-96. Similar systems are widely used in different applications, such as cutting machines, printing machines, and packing machines. In this example, the material feeder is the master axis in the E-Cam system. When the master axis operates, it simultaneously sends pulse commands to the slave axis (cutter).

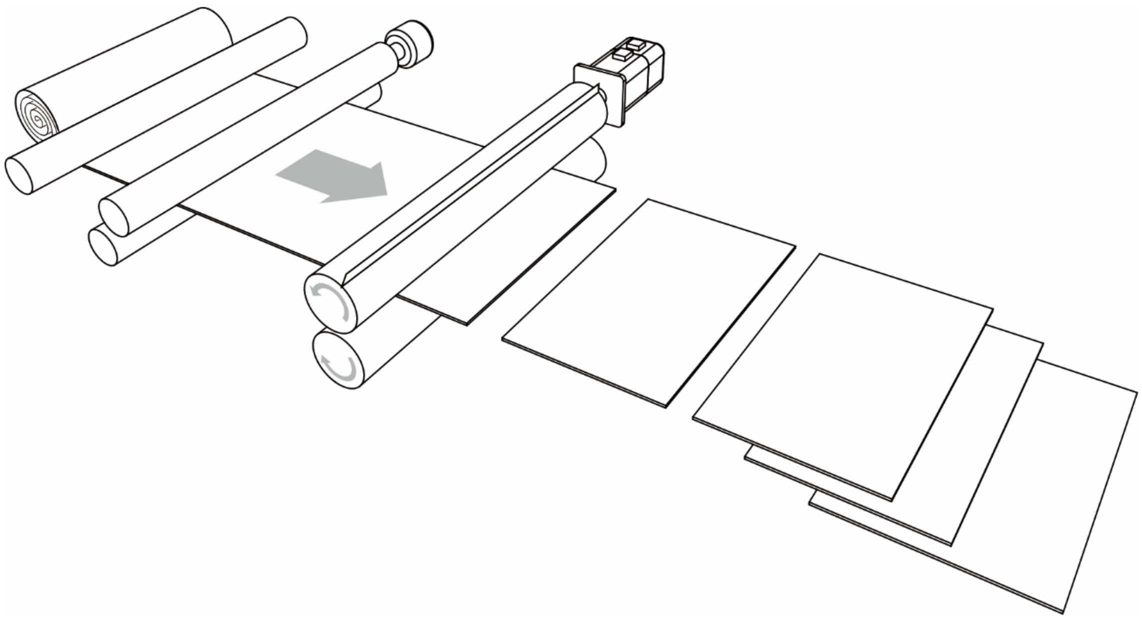
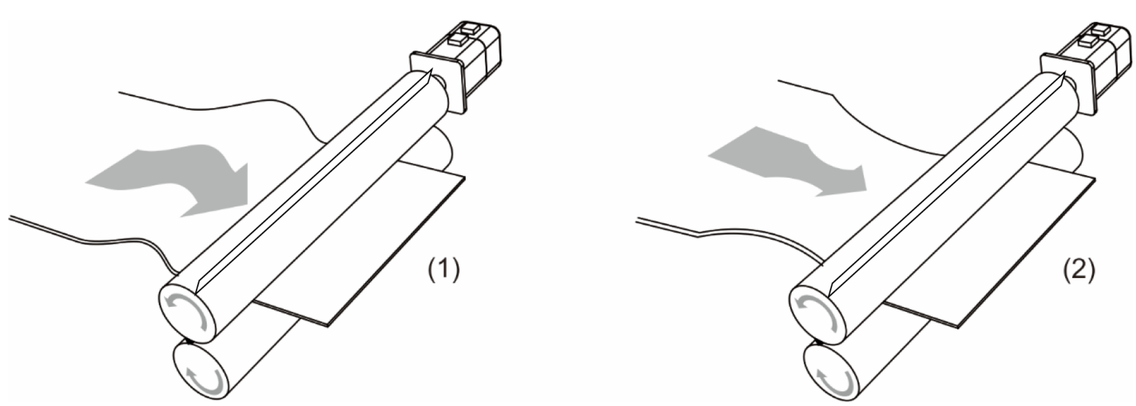


Figure 7-96 Rotary shear system - cutting machine

Apart from the requirement for calculating the correct cutting length, the operation speeds of both the material feeder and cutter have to be the same during cutting. This stage in the E-Cam curve is called the synchronous speed zone. During cutting, if the material feeder runs too fast, the material might be crushed or piled in front of the cutters, as shown in Figure 7-97 option (1).
If the feeder runs too slow, the cutter might over-stretch the material, causing the distortion of the material as shown in Figure 7-97 option (2).



(1) Material feeder runs too fast; (2) Material feeder runs too slow

Figure 7-97 Inconsistent speed in synchronous speed zone for cutting machine operation

E-Cam curve

In the E-Cam curve for the rotary shear system, apart from the requirement that the cutter cuts at the right position, it is important that the master axis and slave axis run at the same speed, which means the relative speed is zero, so the the materials are not over-stretched when cutting. In terms of cutters, wider cutters require larger synchronous speed zone (as shown in Figure 7-98). These wider cutters can be heat sealers or jaw/crimp cutters. The proportion of this zone is determined by the cutting length instead of the cutter width, as shown in Figure 7-99.

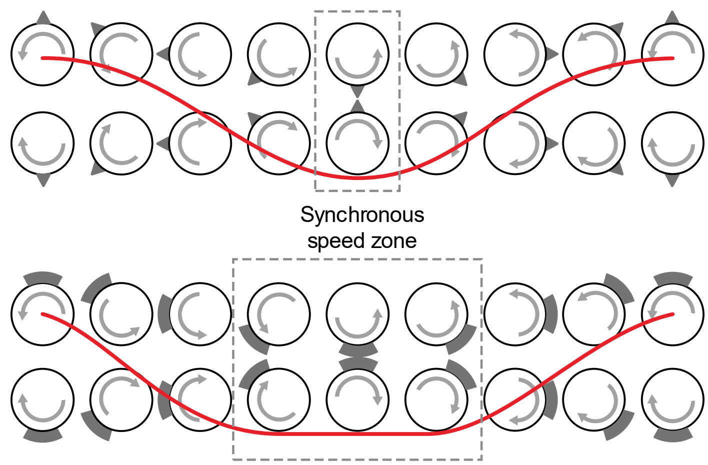


Figure 7-98 Cutter type and synchronous speed zone

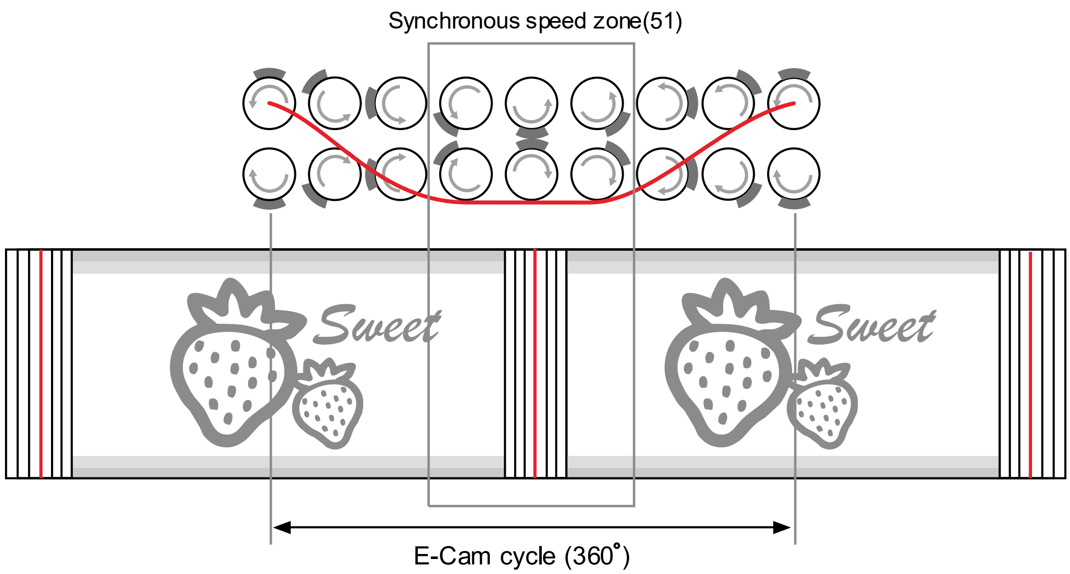
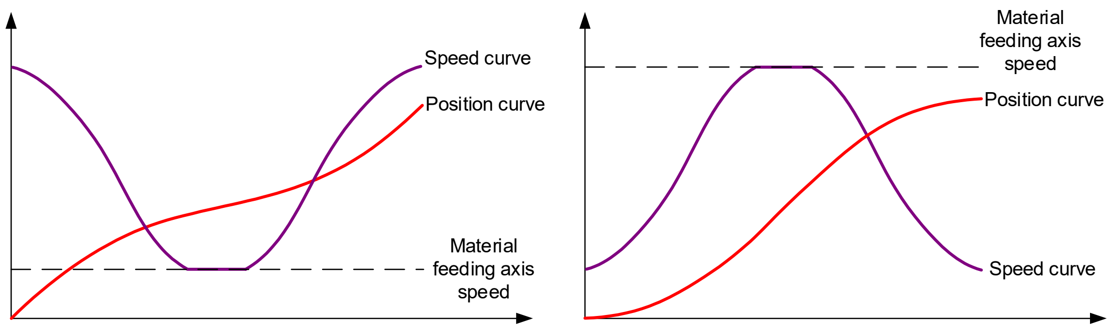


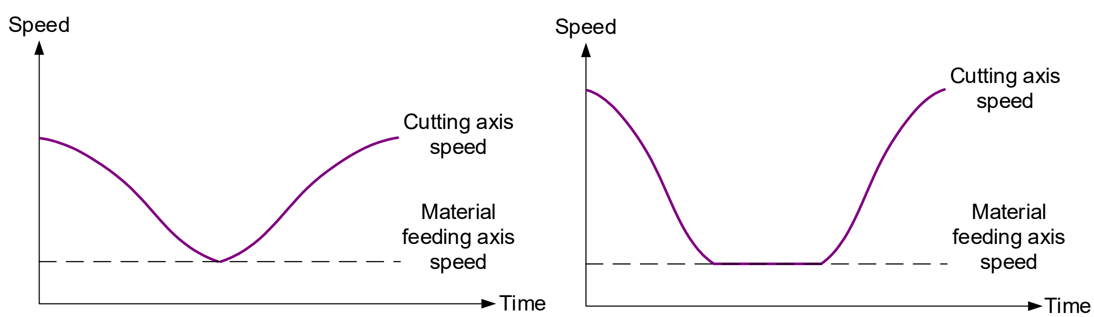
Figure 7-99 Definition of synchronous speed zone

For E-Cam curves used in the rotary shear applications, apart from the requirements of synchronous speed and fixed length, the speed has to be stable. The arc length between cutter ends (Figure 7-103) and the cutting length proportion will determine the speed variation. The larger the value, the greater the variation. If each arc length between cutters is longer than the cutting length, the motor speed is faster than the master axis before entering the synchronous speed zone. So, the motor needs to decelerate to the master axis speed, as shown in Figure 7-100 (a). If the arc length between cutters is shorter than the cutting length, the motor speed is slower than the master axis before entering the synchronous speed zone. Therefore, the motor needs to accelerate to the master axis speed, as shown in Figure 7-100 (b).



(a) Arc length between cutter ends > cutting length
(b) Arc length between cutter ends < cutting length
Figure 7-100 Correlations between the cutting length, speed, and arc length between cutter ends

You can adjust the cutting length by changing the cutter rotation speed. However, the larger the synchronous speed zone, the less flexibility to adjust the cutting length. As shown in Figure 7-101, the cutting length is the same, meaning the rotation distances of the pointed cutter and wide blade cutter are the same (measure of the speed curve area). When you use a wide blade cutter, the non-synchronous speed zone is larger and requires an abrupt acceleration or deceleration, which may easily make the motor reach the maximum torque limit. When the cutter cannot cut the material with a shorter length due to the cutter speed or the maximum current limit, increase the cutter number to shorten the operation distance per cutting, making the cutter slower and the current output lower.

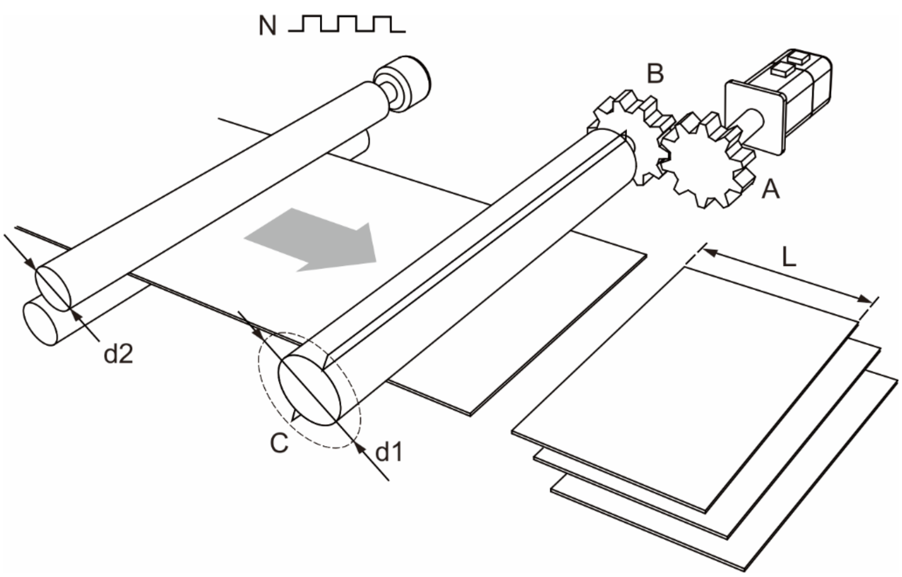


(a) Pointed cutter (small synchronous speed zone)
(b) Wide blade cutter (large synchronous speed zone)
Figure 7-101 Size of the synchronous speed zone and motor speed

To avoid drastic speed change during operation, when plotting the system, take the arc length between cutter ends, cutting length, and the synchronous speed zone into consideration. The synchronous speed zone is fixed based on the machining requirements for the materials; whereas the cutting length is determined by the acceleration and deceleration zones. Therefore, in addition to the cutter diameter, the cutter number can determine the speed and alleviate the speed variation caused by the acceleration or deceleration of the motor, which makes the system operation smoother. You can also use a lower power motor to have a more cost-effective servo system.

Create an E-Cam curve

You can plot the E-Cam curve for the rotary shear system through SureServo2 Pro or macros (see section 7.3.9)of the servo drive. There are four methods available to create E-Cam curves for rotary shear applications. [Rotary Shear - W/O Sync. Zone], [Rotary Shear - Fixed Sync. zone], and [Rotary Shear - Adjustable Sync. Zone] are designed for common rotary shear applications. The differences among the three are whether there is a synchronous speed zone and whether this zone is adjustable. [Rotary Shear - Printing Machine] is specially designed for creating curves for printing machines. Two macros provided by the servo drive are available for creating E-Cam curves. Macro #6 can create E-Cam curves with fixed synchronous speed zone and Macro #7 can create E-Cam curves with adjustable synchronous speed zone for rotary shear applications. The required setting parameters and the rotary shear mechanical structure are shown in Figure 7-102.



N: pulse number per encoder revolution
A: gear teeth number of the motor
B: gear teeth number of the cutter
L: cutting length
C: number of cutters
d1: cutter diameter
d2: encoder roller diameter

Figure 7-102 Example application of a rotary shear

Create E-Cam curves without synchronous speed zone

This kind of E-Cam curve is only suitable for applications using pointed cutters and can only be created by SureServo2 Pro. The setting interface is shown in Figure 7-104. The specification settings labeled in Figure 7-102 for the rotary shear are as follows:

1) Gear teeth ratio: set the gear teeth number of the motor (A) and gear teeth number of the cutter (B).
2) Number of cutters (C): set the cutter number based on the rotary shear mechanism.
3) Cutter diameter (d1): set the cutter diameter based on the rotary shear mechanism. The cutter radius is the distance from the cutter shaft center to the cutter end; the cutter diameter is two times of the cutter radius. This value does not change with the cutter number and the software will calculate the circumference drawn by the cutter end.
4) Encoder roller diameter (d2) and encoder pulse number (N): set the diameter and pulse per encoder revolution. The command resolution can be calculated with these two values.
   If you know the master axis gear ratio, entering the encoder diameter and pulse number are not required. You can simply input the values for P5.083 and P5.084.
5) PUU number per motor revolution: set the PUU number per motor revolution (slave axis) after being calculated with the E-Gear ratio (P1.044 / P1.045).
6) Cutting length (L): set the material cutting length. To avoid generating an unsuitable rotary shear curve, the software automatically limits the cutting length by referring to the ratio (R) of the cutting length (L) and the arc length between cutter ends (a); R = L / a; R = 0.3 – 3.

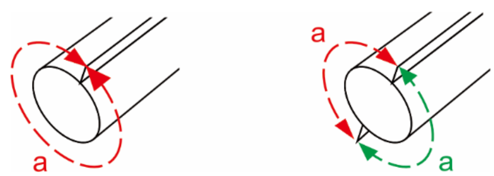


Figure 7-103 Arc length between cutter ends

7) Speed compensation (Vc): in some rotary shear applications, the speeds of the master and cutter axis are different during cutting; so you can use speed compensation to change the speed of the cutter axis. In the cutting zone, if the speed compensation value is positive, the cutter axis speed is faster than the master axis; if the speed compensation value is negative, the cutter axis speed is slower than the master axis.

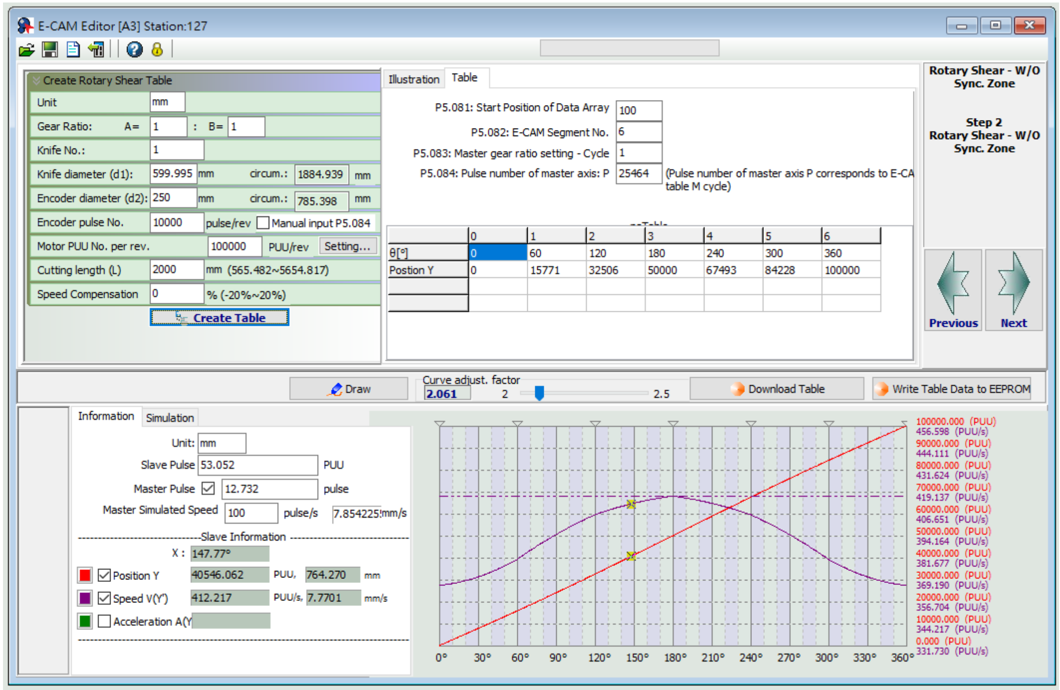


Figure 7-104 SureServo2 Pro setting interface for [Rotary Shear - W/O Sync. Zone]

Create E-Cam curve with fixed synchronous speed zone

This method allows you to create a rotary shear curve with fixed synchronous speed zone, which range is fixed to 51°. You can use SureServo2 Pro to create the table, which parameter setting is similar to the curve for rotary shears without synchronous speed zone, as shown in Figure 7-105. The software automatically limits the cutting length by referring to the ratio (R) of the cutting length (L) and the arc length between cutter ends (a); R = L / a; R = 0.07 - 2.5.

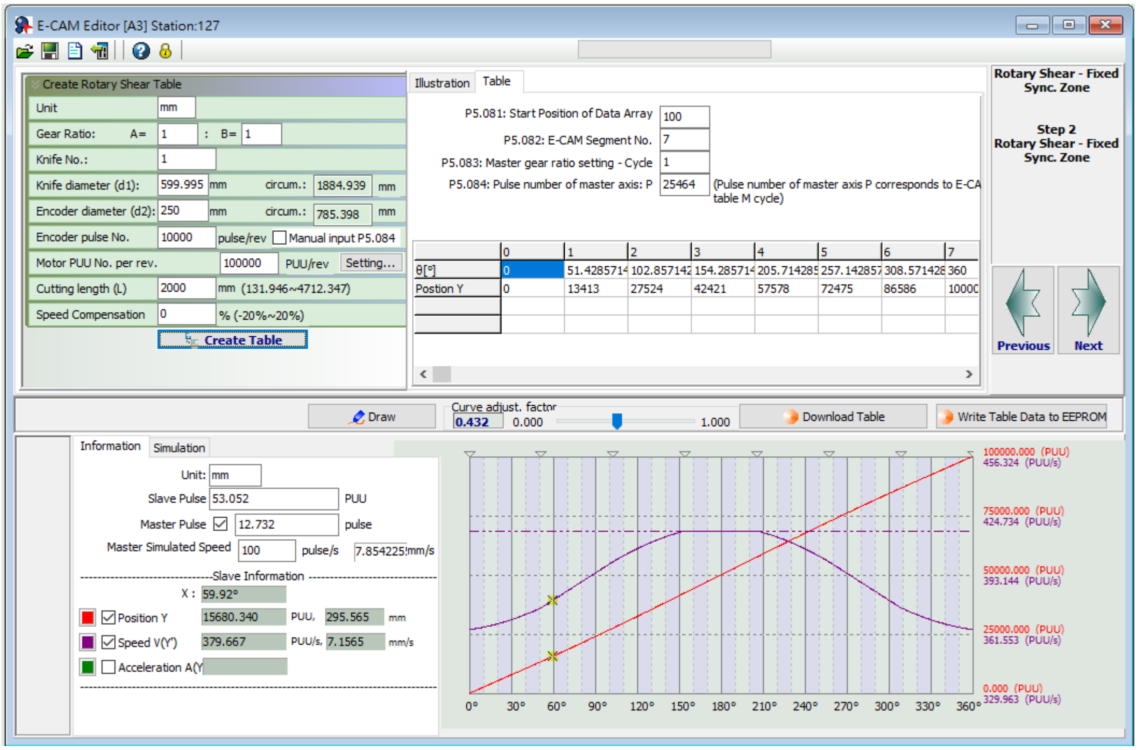


Figure 7-105 SureServo2 Pro setting interface for [Rotary Shear- Fixed Sync. Zone]

Create E-Cam curve with adjustable synchronous speed zone

This table creation method is for generating an E-Cam curve with adjustable synchronous speed zone. Use SureServo2 Pro to create the table. The parameter setting for the rotary shear curve is similar to the setting of that without the synchronous speed zone, as shown in Figure 7-106. The software automatically limits the cutting length by referring to the ratio (R) of the cutting length (L) and the arc length between cutter ends (a); R = L / a.

1.88 > R × Speed compensation (V_C).

The difference from the rotary shear curves without the synchronous speed zone is you can plot the acceleration, synchronous speed, and S-curve zones for the curves with adjustable synchronous speed zone. If the deceleration zone size is the same as the acceleration zone, the software defines the remaining part as the waiting zone.

Figure 7-106 SureServo2 Pro setting interface for [Rotary Shear - Adjustable Sync. Zone]

You can use Macro #7 of the servo drive to create the table with the same method. The advantage of using the macro to create an E-Cam curve is that when changing the cutting length is required, you can create a new curve simply by modifying the parameters. It is very friendly for those applications that require frequent modification of the cutting length. The setting steps are as follows:

1) Set the start position for the data array: use P5.081 to specify the data array start position for E-Cam and use P5.085 to set the E-Cam's segment number for clutch engagement. When using Macro #7, the range for the number of allowed E-Cam segments (P5.082) is 30–72. It is suggested to set number of segments to 72 for the optimal resolution of 5°.
2) Set the system E-Gear ratio: set the system E-Gear ratio with P1.044 and P1.045.

3) Set the E-Cam's gear ratio and curve scaling: set the pulse number required for the cutting length, which is P5.083 = 1 and P5.084 =

$$\frac{\text{Pulse number per turn of the master axis encoder } N}{\pi \; x \text{ master axis encoder diameter } d2 \text{ (mm)}} \; x \text{ Cutting Length L (mm)}$$

Use P5.019 to specify the E-Cam curve scaling.

4) Set the parameters for the E-Cam curve zones: specify the size of the waiting, acceleration, synchronous speed, and S-curve zones.
   - *P5.093.DCBA sets the S-curve level (S) with the range of 1 - 4; the calculation for the corresponding angle (S°) is as follows.*
   - *P5.093.UZYX sets the angle (W) of the waiting zone with the range of -1° to 170° in hexadecimal. If you enter -1 (0xFFFF), it means the cutter speed is 0 in the waiting zone and the angle for the waiting zone is calculated by the servo drive.*
   - *P5.094 sets the angle (Y) of the synchronous speed zone with the range of 0° - 330° in decimal format.*
   - *The acceleration zone is automatically calculated by the servo drive, as shown in the following formula:*

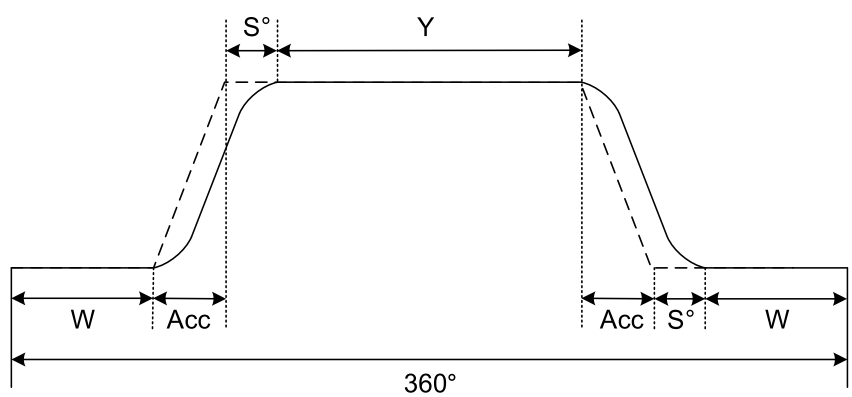
$$S° = \frac{2^S \text{ x } 360°}{\text{E – Cam total segment number } P5.082}$$

$$360° = 2W + 2Acc + 2S° + Y$$

As the synchronous speed zone is adjustable, there are limitations when using Macro #7 to create the waiting zone of the E-Cam curve. The conditions are as follows:

$$\hat{W} \text{ (minimum waiting zone)} = 180° + \frac{360°}{P5.082} - \frac{360°}{R} + \frac{Y}{2}$$

If the waiting zone (W) < minimum waiting zone (Ŵ), the error 0xF07A occurs, and you must increase the waiting zone or decrease the synchronous speed zone. If the waiting zone (W) = minimum waiting zone (Ŵ), the cutter speed is 0 at the waiting zone. If the waiting zone (W) > minimum waiting zone (Ŵ), the cutter axis speed is greater than 0 at the waiting zone.



5) Set the parameters for creating the E-Cam curve table (refer to Figure 7-102):
   - *P5.095.DCBA = Motor gear teeth number (A) cutter number (C) in decimal format.*

- *P5.095.UZYX = Cutter gear teeth number (B) in decimal format. For example, if the motor gear teeth number A = 10, cutter number C = 1, and cutter gear teeth number B = 1, then P5.095 = 0x000A0001 (HEX), but since the parameter only accepts decimal format, you need to set P5.095 to 655361 (DEC); P5.096 =*

$$\frac{\text{Cutting length } L \text{ (mm)}}{\pi \text{ x cutter diameter } d1 \text{ (mm)}} \text{ x Cutter number } C \text{ x Speed compensation } V_c \text{ x } 1000000$$

If Vc = 1, there is no speed compensation. If Vc = 0.9, the speed of the cutter axis in the synchronous speed zone is 0.9 times of the master axis speed. If Vc = 1.1, the speed of the cutter axis in the synchronous speed zone is 1.1 times of the master axis speed.

6) Enable Macro #7: write 0x0007 to P5.097 to enable Macro #7. Read P5.097 and if it returns 0x1007, it means using macro for table creation is successful. If any of the following failure codes shows, modify the setting according to the description.

Failure Code	***Description***
0xF071	When the clutch is engaged, the E-Cam table cannot be created.
0xF072	Degree of synchronous area of P5.094 exceeds the range (0 - 330).
0xF073	S-curve level of P5.093.DCBA (HEX) exceeds the range (1 - 4).
0xF074	Degree of waiting zone of P5.093.UZYX (HEX) exceeds the range (-1 to 170).
0xF075	Data of P5.096 exceeds the range (50000 - 5000000).
0xF076	E-Cam segment number of P5.082 exceeds the range (30 - 72).
0xF077	P5.081 data array start position exceeds the array length.
0xF078	The set values of P1.044 and P1.045 for the E-Gear ratio is too high. Decrease the values of P1.044 and P1.045, but maintain the same ratio, for example: adjust 167772160 : 1000000 to 16777216 : 100000.
0xF079	Acceleration degree is too small. Decrease the value for the waiting zone, synchronous speed zone, or S-curve level.
0xF07A	Waiting zone < minimum waiting zone. Increase the value for the waiting zone or decrease the value for the synchronous speed zone.

The following method helps you to test the maximum border condition and create the E-Cam curve successfully when using the HMI or controller to create the E-Cam table.
If the ratio (R) of the cutting length (L) and the arc length between cutter ends (a) is 0.05 - 1.09 (R = 0.05 - 1.09), and the E-Cam segment number P5.082 = 72, follow the parameter setting procedure to create the table with Macro #7.

1) Set the waiting zone (W) and synchronous speed zone (Y) based on the S-curve level:

S-curve Level (S)	***Range of Waiting Zone (W) and Synchronous Speed Zone (Y)***
1	0° ≤ W ≤ 75°; 0° ≤ Y ≤ 150°
	0° ≤ 2W+Y ≤ 300°
2	0° ≤ W ≤ 70°; 0° ≤ Y ≤ 150°
	0° ≤ 2W+Y ≤ 290°
3	0° ≤ W ≤ 55°; 0° ≤ Y ≤ 110°
	0° ≤ 2W+Y ≤ 220°
4	0° ≤ W ≤ 25°; 0° ≤ Y ≤ 30°
	0° ≤ 2W+Y ≤ 80°

2) Write the corresponding parameters:
   - *P5.093.DCBA = S-curve level (S)*
   - *P5.093.UZYX = Angle of the waiting zone (W) set in hexadecimal.*
   - *P5.094 = Synchronous speed zone (Y); other parameter settings for curve table creation are the same as Step 5 mentioned above. Set P5.097 to 7 to enable Macro #7.*

If the ratio (R) of the cutting length (L) and the arc length between cutter ends (a) is 1.1 - 5 (R = 1.1 - 5), and the E-Cam segment number P5.082 = 72, follow the parameter setting procedure to create the table with Macro #7.

1) Set the S-curve level: use macro parameter P5.093.DCBA to set the S-curve level (S) with the range of 1 - 4.
2) Set the synchronous speed zone: use macro parameter P5.094 to set the angle of the synchronous speed zone (Y). Its angle must be greater than 0. As shown in the following formula, when the synchronous speed zone (Y_{Max}) is less than 0 degree, decrease the S-curve level.

$$Y_{Max} = \frac{360}{R} - 5 \text{ x } (3+2^{s+1}) \geq 0$$

3) Set the waiting zone: use macro parameter P5.093.UZYX in hexadecimal to set the angle of the waiting zone (W). The calculation is as follows.

$$W\,(Hex) = 180 - \frac{180}{R} - \frac{5 \text{ x } (2^{s+1} - 1)}{2}$$

When cutting, if the cutter speed is faster than the material feeder, it means the speed compensation (V_C) is greater than 1. Proceed to Step 4 and re-plan the synchronous speed and waiting zones based on the speed compensation requirements. If the cutter is slower than the material feeder, it means the speed compensation is equal to or less than 1. Proceed to Step 6, set the mechanism related parameters, and enable Macro #7 to complete the E-Cam curve creation.

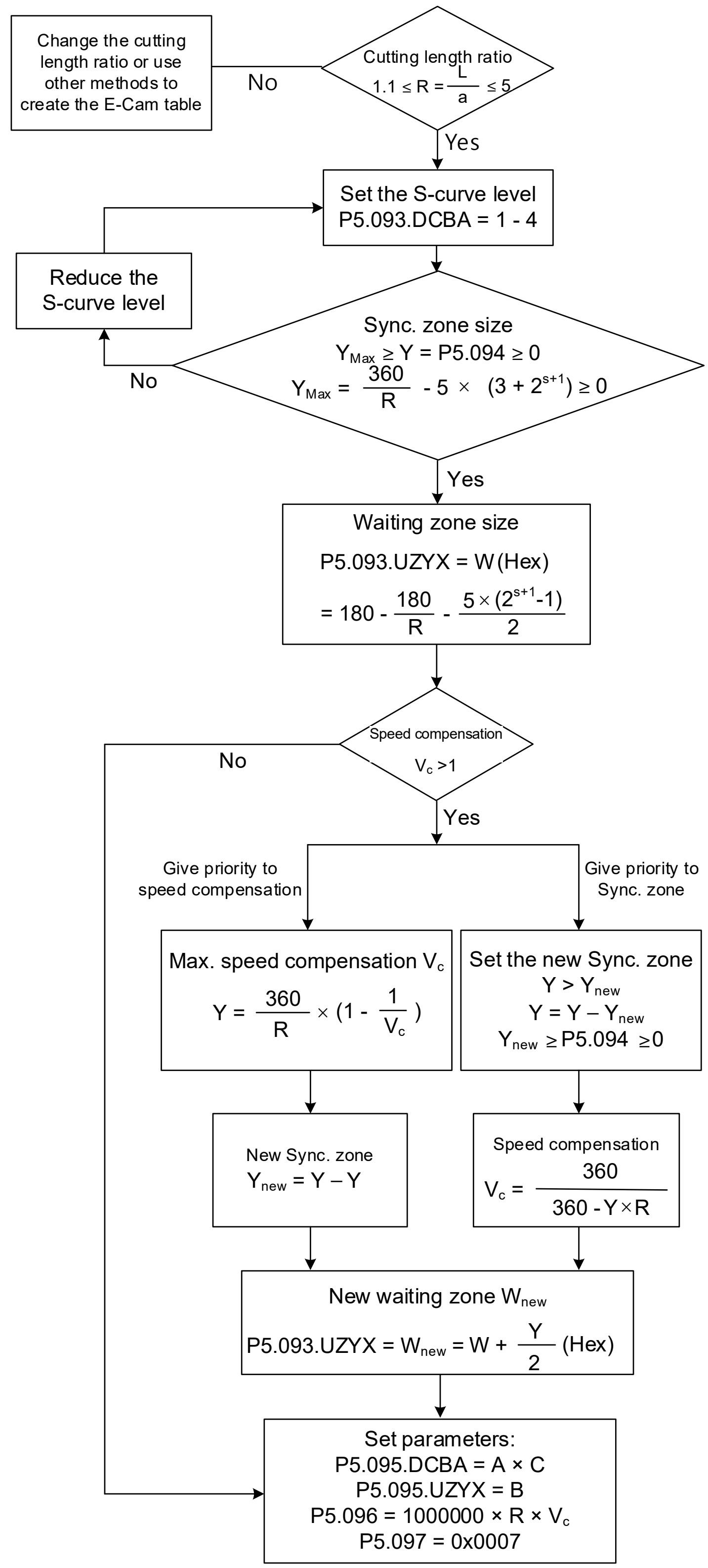
4) Taking adjusting the cutter speed as the priority, set the required maximum speed compensation and use this compensation value to re-calculate the Sync zone (Y_{new}) and the Waiting zone (W_{new}). Taking the Sync zone size as the priority, set the Sync zone (Y_{new}) and use the size of this Sync zone to re-calculate the speed compensation value and the Waiting zone (W_{new}). Refer to the following formulas to calculate the synchronous speed zone and speed compensation value.

Priority	***Calculation 1***	***Calculation 2***
Adjust cutter speed	Use the speed compensation (V_C) to calculate the variation of the Sync zone (ΔY): $\Delta Y = \frac{360}{R} \text{ x } (1 - \frac{1}{V_c})$	New Sync zone size (Y_{new}): $Y_{new} = Y - \Delta Y$
Sync zone size	Use the new Sync zone size (Ynew) to calculate the variation of the Sync zone (ΔY): $Y > Y_{new}$ $\Delta Y = Y - Y_{new}$ Y_{new} P5.094 0	Speed compensation (V_C): $V_c = \frac{360}{360 - \Delta Y \text{ x } R}$

5) Set the new waiting zone size: use the macro parameter P5.093.UZYX in hexadecimal to set the re-calculated waiting zone angle (W_{new}), as shown in the following formula.

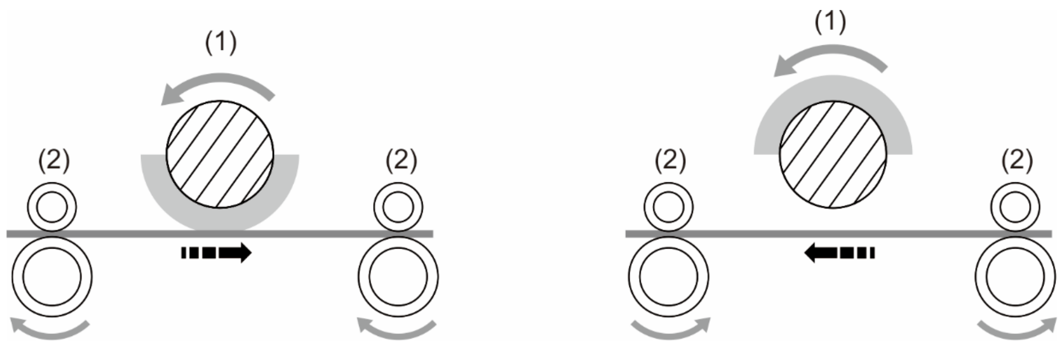
$$W_{new} = W + \frac{\Delta Y}{2}$$

6) Set the parameters related to the mechanism and enable Macro #7: use the macro parameter P5.095.DCBA in hexadecimal to set the motor gear number (A) Cutter number (C). Use the macro parameter P5.095.UZYX in hexadecimal to set the cutter gear number (B). Set P5.096 to 1000000 Ratio of the cutting length and arc length between two cutter ends (R) Speed compensation (V_C). Set P5.097 = 0x0007 to enable Macro #7 to complete the E-Cam curve for the adjustable synchronous speed zone.



Create E-Cam curve for periodic intermittent printer

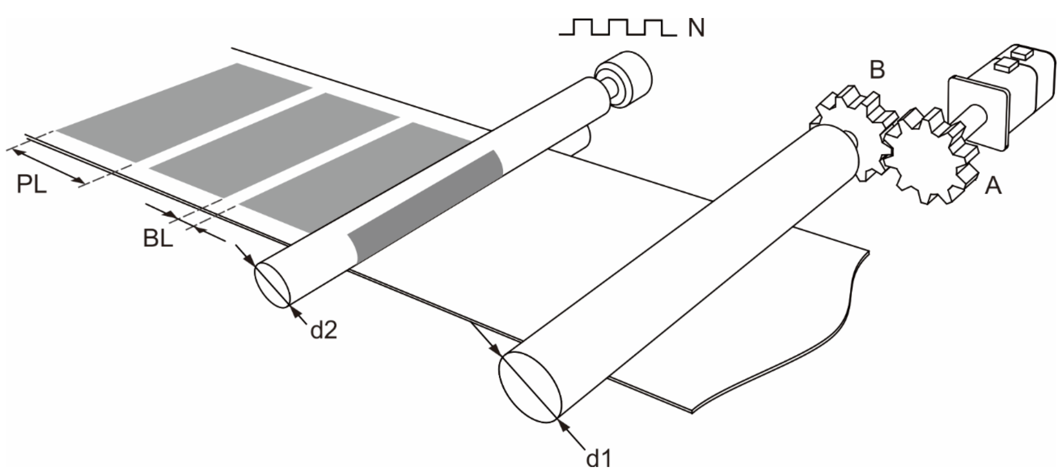
Use this type of E-Cam curve when the printing material length is limited and full print cannot be carried out. This type of curve helps to save the material, decreasing the interval between each print pattern by retracting the material when the printing plate detaches from the material. In the case of paper printing, the printing axis rotates at a fixed speed in a single direction. When the printing plate and paper come into contact, the paper and printing axis starts printing with the same linear speed, as shown in Figure 7-107 (a). Once the printing motion stops and during the interval when the printing plate and paper are separated, the drive roll starts to decelerate and move in the opposite direction until reaching the specified zone, as shown in Figure 7-107 (b). Afterwards, the drive roll resumes operation in the printing direction. When the printing plate and paper come into contact again, the paper and printing axis resume a synchronous relation and make the next print. Regarding the E-Cam curve for intermittent printing machines, the parameters that you must set and the corresponding relation with the printing machine mechanical structure are shown in Figure 7-108.



a) Printing motion b) Retraction motion

1) Printing axis; (2) Drive roll

Figure 7-107 Intermittent printing machine

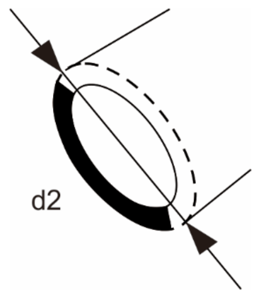


N: number of pulses per printing axis revolution
A: number of gear teeth on motor; B: number of gear teeth on material feeder
PL: print length; BL: blank length
d1: drive roll diameter; d2: printing axis diameter

Figure 7-108 Printing machine mechanical structure

You can create the E-Cam curve table for intermittent printer by using the SureServo2 Pro software. Figure 7-109 shows the user setting interface. The printer specification settings are as follows.

1) Gear ratio: set the number of motor gear teeth (A) and number of material feeder gear teeth (B).
2) Print length (PL) and blank length (BL): set the print length and blank length.
3) Drive roll diameter (d1): set the drive roll diameter for conveying the material.
4) Printing axis diameter (d2) and printing axis pulse number (N): set the diameter of master printing axis and number of pulses per revolution.



5) Number of PUU per motor revolution: set the number of PUU per motor revolution of the drive roll after E-Gear ratio conversion (P1.044 / P1.045).

The ratio of printing axis circumference to printing zone length ($R = \frac{\pi \text{ x } d2}{PL + BL}$) must exceed 1 when using E-Cam curves for intermittent printing machines in order to save the material. Calculate the synchronous speed zone angle with the formula $Y = \frac{PL}{\pi \text{ x } d2} \text{ x } 360^{\circ}$. You can also adjust the waiting zone angle and S-curve angle in the advanced settings. Increase the angle for the synchronous speed zone to increase the zone size. This ensures the printing axis and drive roll are in the uniform operation speed and stable during printing motions to achieve higher quality printing.

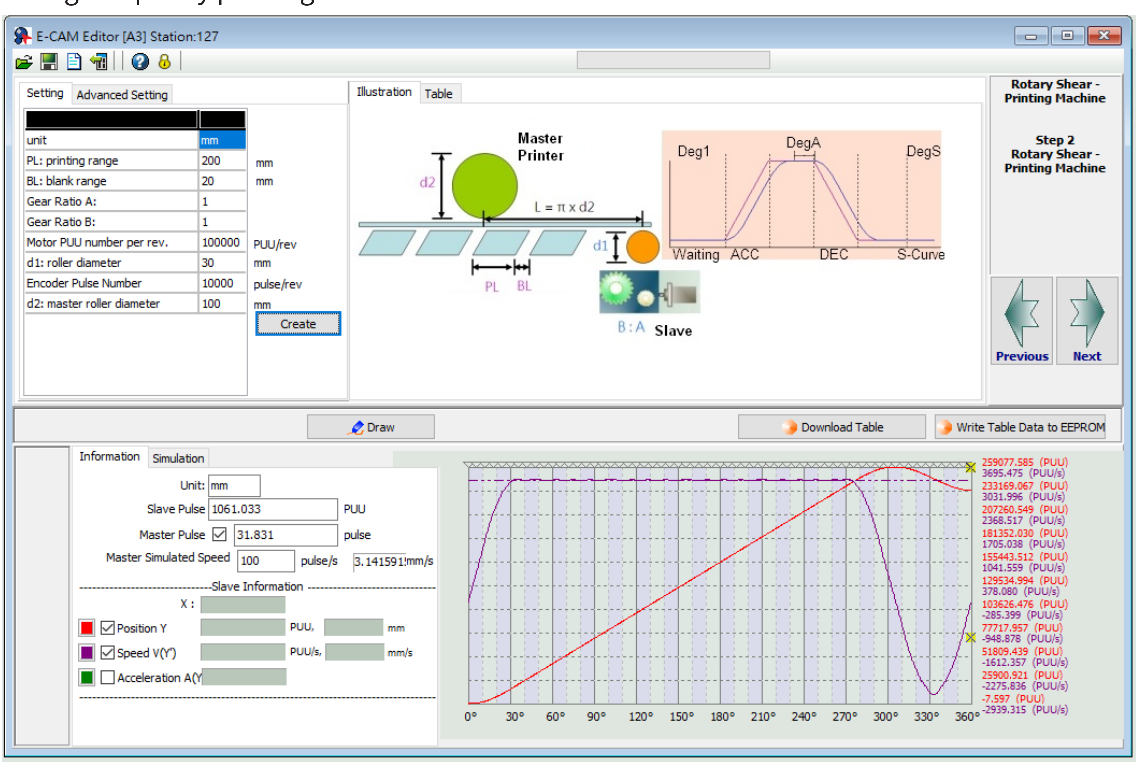


Figure 7-109 SureServo2 Pro rotary shear - intermittent printing machine setting interface

E-Cam phase alignment

The E-Cam phase alignment function is another compensation method provided by the servo drive. You need to first set the phase for the E-Cam phase alignment and the compensation detection position for the external sensor. In each cycle, everytime the E-Cam operates to the detection position of the external sensor, the servo drive starts comparing the actual phase with the correct phase and then calculates the deviation of the slave axis. Then, the servo drive writes this deviation amount to the PR program for immediate or later compensation (user-defined), which is completed with the E-Cam and PR command overlapping function introduced in Section 7.3.5. The compensation mechanism for the E-Cam phase alignment is shown in Figure 7-110.

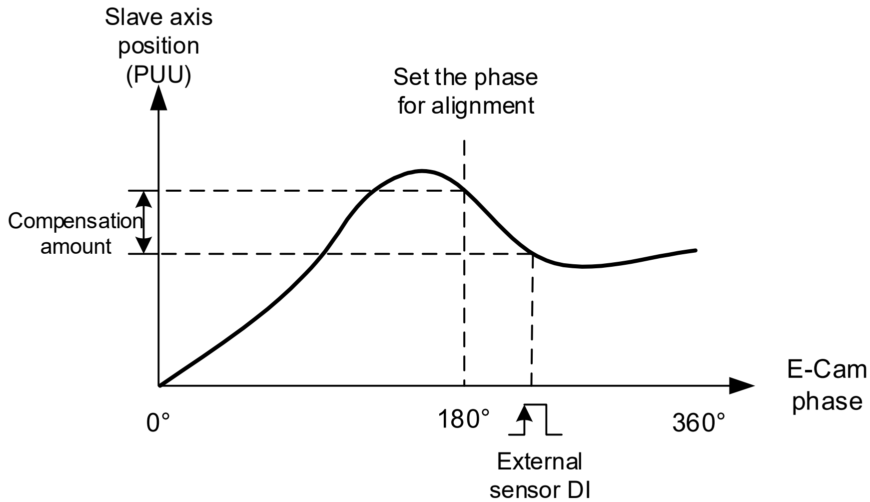


Figure 7-110 Compensation mechanism for phase alignment

E-Cam phase alignment setting steps

When using the E-Cam phase alignment compensation, set the parameters for the DI (digital input), phase alignment position, and compensation level. The flow chart is shown in Figure 7-112. The steps to set the E-Cam phase alignment function are as follows.

1) Presetting: create and download the E-Cam curve to the servo drive. Set the E-Gear ratio, including the system E-Gear ratio (P1.044 and P1.045), E-Cam gear ratio (P5.083 and P5.084), and E-Cam curve scaling (P5.019). Complete the settings relevant to the E-Cam, including the start address of the data array (P5.081), E-Cam total segment number (P5.082), and the E-Cam segment number for engagement (P5.085).
2) DI relevant settings: connect the external sensor with the DI point. Define this DI as [0x35]ALGN. Since both the DI and the sensor have delay, the captured phase position might also delay. Use P2.074 to set the delay time compensation of the DI as follows:

$$P2.074 = P2.009 \text{ (DI filter time)} + \text{Sensor delay time}$$

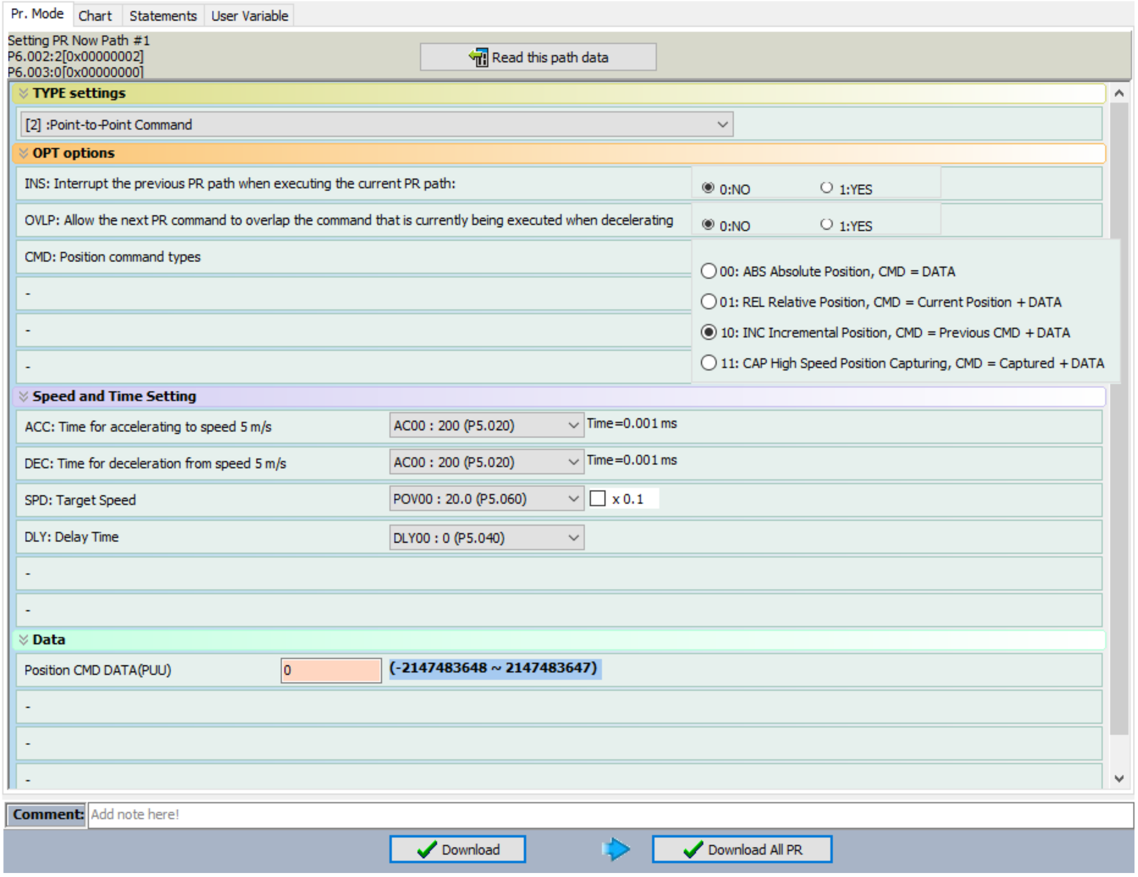
To prevent mistakenly triggering the DI, set P2.073.DC in hexadecimal to specify the masking zone proportion (%). The master axis pulse must exceed the masking area before the next phase alignment starts. This function is only applicable to the applications with forward direction pulse input.
The formula is as follows:

$$\text{Masking zone (pulse)} = \frac{P5.084}{P5.083} \text{ x } P2.073.DC(\%)$$

3) E-Cam phase alignment setting: P2.075 sets the alignment position for E-Cam phase alignment. The unit is the pulse number of the master axis, which corresponds to the specified E-Cam phase after conversion. For example, if the pulse number per E-Cam curve cycle is 36000, then P5.083 = 1 and P5.084 = 36000. If you set P2.075 to 18000, then upon the DI receives the signal, the system starts comparing the slave axis actual position and the set E-Cam position of 180° and then calculates the required compensation value. If you set P2.075 to 10000, when the DI receives the signal, the system starts comparing the slave axis actual position and the set E-Cam position of 100° and then calculates the required compensation value. Use the monitoring variable 063(3Fh) to monitor the slave axis actual position.
4) Filter setting: to keep the operation smooth and decrease the position deviation caused by the noise of the external sensor, when the distances between each mark read by the sensor are equal, you can use the filter to improve the stability of the phase alignment. P2.073.YX specifies the filter range (%) in hexadecimal. If the deviation is smaller than this set value, the filter function is effective. If the deviation is greater than this value, it means the deviation is greater and requires immediate compensation. P2.076.Y specifies the filter intensity. Use the monitoring variable 085(55h) to monitor the E-Cam phase deviation in percentage, which unit is 0.1%. If this value is 10, it means the deviation is 1%, which is 3.6°.
5) Compensation direction setting: P2.076.UZ specifies the allowable forward alignment rate in hexadecimal. Set it to 0% to perform the alignment always in the reverse direction; set 50% to align the phase with the shortest path; set 100% to perform the alignment always in the forward direction. When you select “always forward” or “always reverse” for the alignment, you must set the maximum correction rate to avoid excessive compensation. Generally, using the shortest path for alignment is recommended. If the application has set the reverse inhibit condition and the deviation is sometimes positive and sometimes negative, use with P1.022.U the reverse inhibit function.
6) Maximum correction rate setting: when the alignment deviation is too great, the compensation amount might thus be great which causes motor vibration or even overload. P2.073.UZ in hexadecimal sets the maximum correction rate and gradually compensates the deviation in stages to alleviate the motor vibration, but it requires longer time to complete the alignment compensation. The formula for maximum correction amount per time is as follows:

$$\text{Max. correction rate per time (pulse)} = \frac{P5.084}{P5.083} \text{ x } P2.073.\text{UZ}(\%)$$

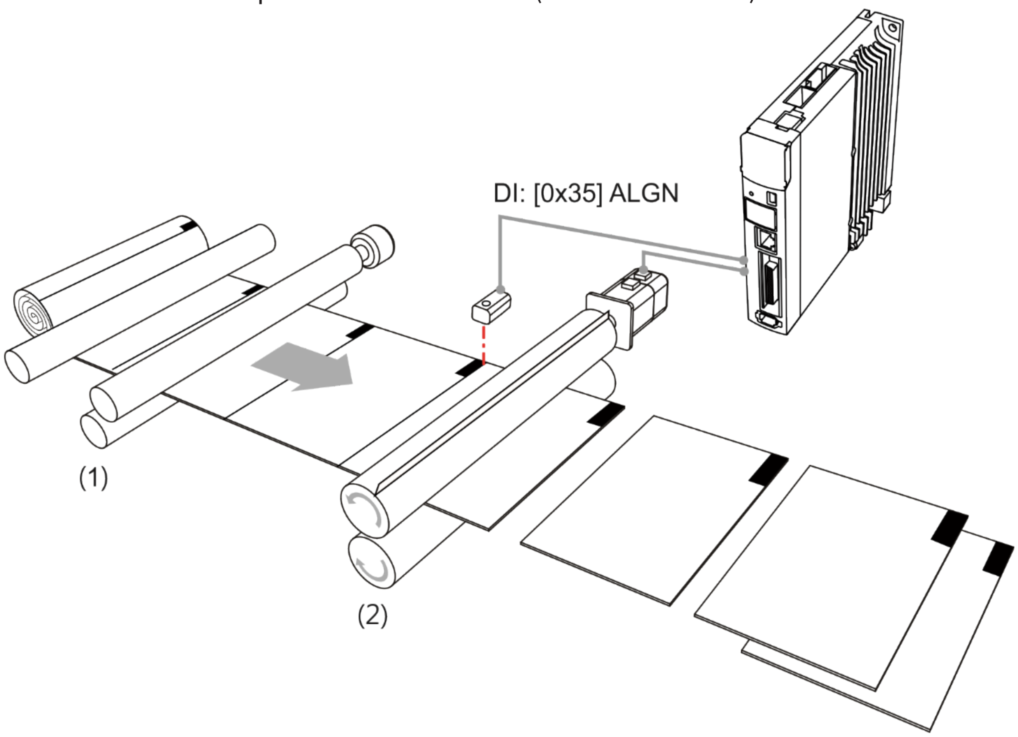
7) PR path setting: the compensation amount for the slave axis is stored in the PR number specified by P2.073.BA. When the slave axis requires compensation, the system can trigger this PR path at the proper timing. When using the E-Cam phase alignment function, set the following for the specified PR: [Point-to-Point Command], [INC Incremental position], [OVLP: allow the next PR command to overlap the command that is currently executed when decelerating], and the appropriate speed and acceleration. Setting the position command is not required because it is automatically set by the E-Cam phase alignment function.



8) E-Cam phase alignment setting: it is set by P2.076.X. P2.076.X Bit 0 enables or disables the alignment function. After this function is enabled, it starts operating as soon as the servo drive receives the DI signal. P2.076.X Bit 1 sets whether to immediately trigger the PR command and calls this PR command when the clutch disengages (P5.088.BA). P2.076.X Bit 2 selects the phase alignment method depending on whether the mark is on the compensated motion axis.

If the mark is on the non-compensation motion axis and when E-Cam phase alignment is operating, the following mark position is unchanged. If the mark is on the compensation motion axis and when E-Cam phase alignment is operating, the following mark position is changed, as shown in Figure 7-111.

A) Mark on the non-compensation motion axis (P2.076.X Bit 2 = 0)



B) Mark on the compensation motion axis (P2.076.X Bit 2 = 1)

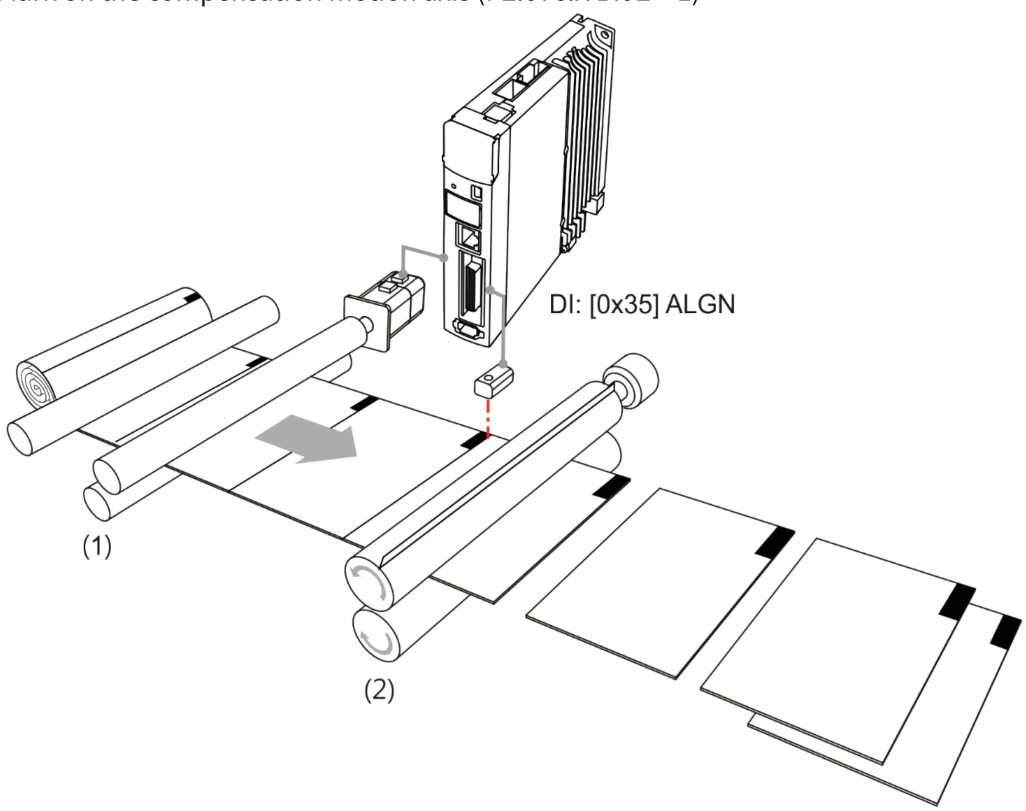
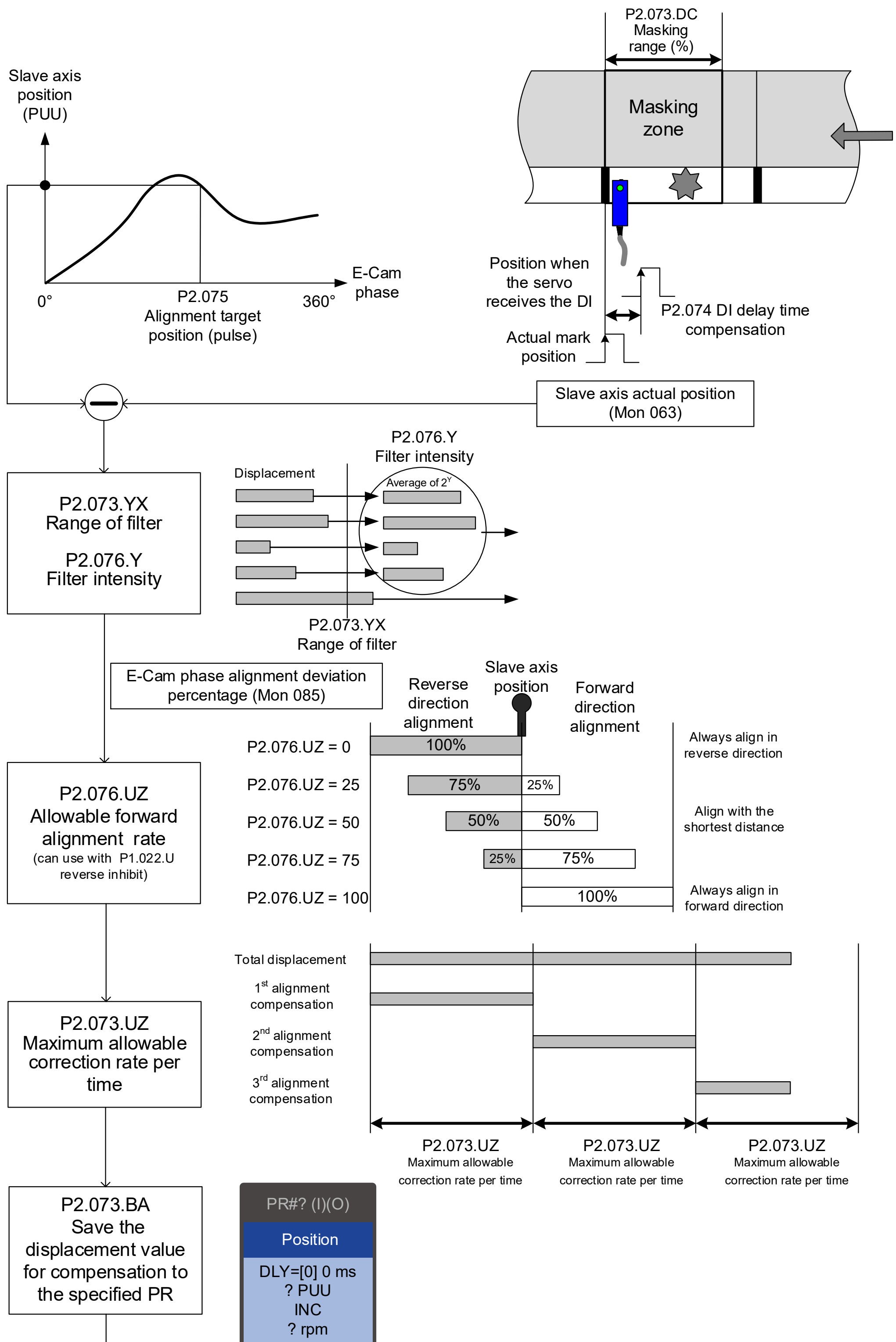


Figure 7-111 E-Cam phase alignment



P2.076.X	0	1
Bit0 Phase alignment switch	Disable	Enable
Bit1 Trigger compensation PR immediately	No	Yes
Bit2 Position of the mark	Alignment mark is on the non-compensation motion axis	Alignment mark is on the compensation motion axis

Figure 7-112 Phase alignment setting procedure

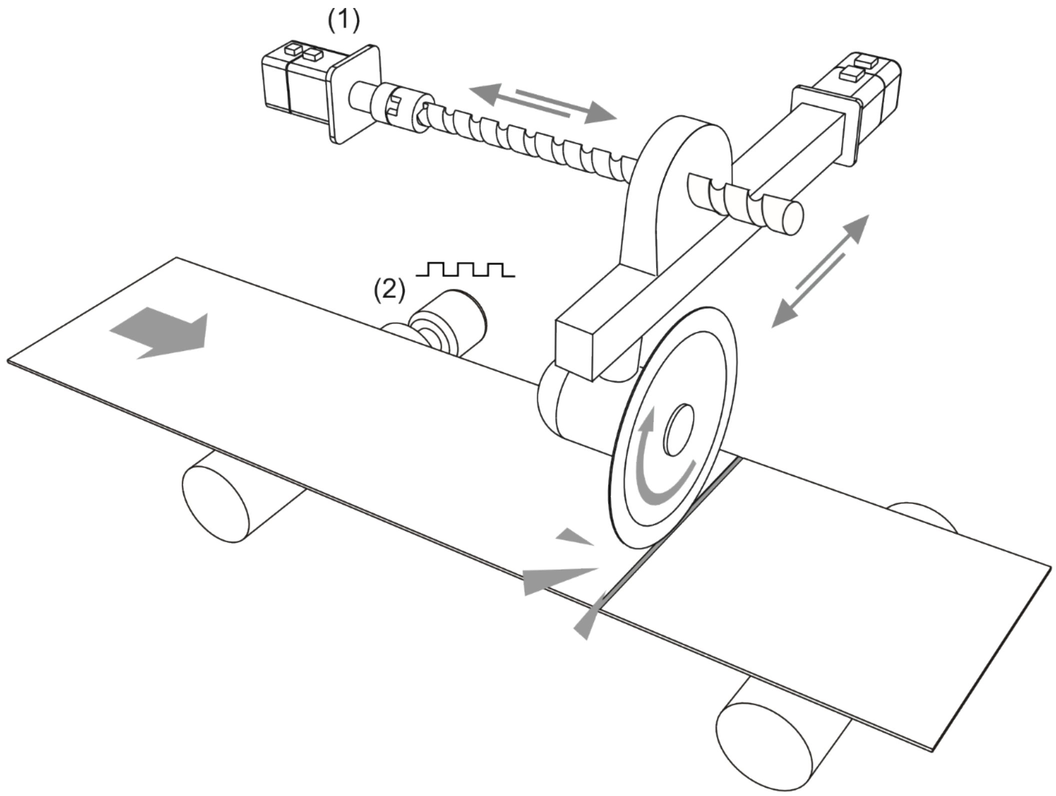
Differences between positioning with synchronous Capture axis and E-Cam phase alignment

The synchronous Capture axis (not currently available with SureServo2) and E-Cam phase alignment are both commonly used compensation approaches for the rotary shear system. In real applications, you can use these two together. The feature differences of the two are as follows.

Item	***Synchronous Capture axis***	***E-Cam phase alignment***
Correction method	Corrects master axis pulses.	Uses the PR incremental position command to correct the slave axis position.
Digital input (DI)	High-speed DI7 (CAP) only.	Uses DI.ALGN most of the time. Uses Macro #E if using high-speed DI7 (CAP) is required.
Marking position	On the non-compensation motion axis.	On any of the axes (compensation or non-compensation).
Equal space marking	Available.	Available and can be used with the filter.
Random marking	Not available.	When using the high-speed DI7 (CAP) with Macro #E, using the filter is not suggested. Keep the distance between the sensor and cutter within the cutting length per cut.

7.3.8 - Flying Shear

The flying shear system is a dynamic cutting system of which feeder continues to operate. Therefore, the cutting and feeding axes have to be synchronous during cutting. The synchronous speed duration should allow the cutter to finish cutting and return to the right position to avoid damaging the cutter or materials, as shown in Figure 7-113. Common applications include cutting machines, filling machines, and labeling machines. Different from rotary shear, the compensation methods using synchronous Capture axis and phase alignment are not applicable to flying shear applications. This avoids machine damage caused by the compensation in the synchronous area.



(1) Cutting axis (slave axis); (2) Material feeder (master axis)

Figure 7-113 Flying shear system

The application of flying shear is divided into two types according to the clutch engagement time. The first type is fully engaged. Its E-Cam curve includes the acceleration zone, synchronous speed zone, deceleration zone, and reset zone. The slave axis is completely controlled by the E-Cam system. The second type is partially engaged: the E-Cam operation is triggered by the signal, and the E-Cam curve has no reset zone. After one cycle of E-Cam operation, the clutch disengages and uses the PR command to reset. Then, the E-Cam waits for the next trigger signal. As shown in Figure 7-114 using a cutting machine as an example, the material feeder is the master axis and the cutting axis is the slave axis. The feeder maintains at a constant speed, and the cutting axis operates according to the E-Cam curve or PR command.

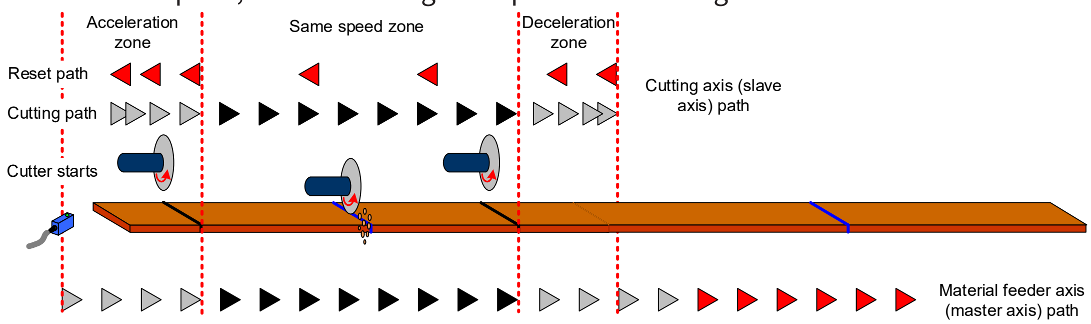


Figure 7-114 Cutting path and operation cycle of the cutting machine

Fully engaged

The fully engaged E-Cam application is suitable for cutting operations without marks. Its E-Cam curve includes the acceleration zone, synchronous speed zone, deceleration zone, and reset zone. The master axis operates at a constant speed. The slave axis operates according to the E-Cam curve and the cutting is complete in the synchronous speed zone. In each cycle, the slave axis starts from the acceleration zone. To avoid wasting materials in the first cycle, set the initial lead pulse before engaged (P5.087). The setting value is the total pulse number of the synchronous speed zone, deceleration zone, and reset zone. If the cutting sensor and material are not aligned during the first cutting, you need to add the offset pulse number of the sensor.

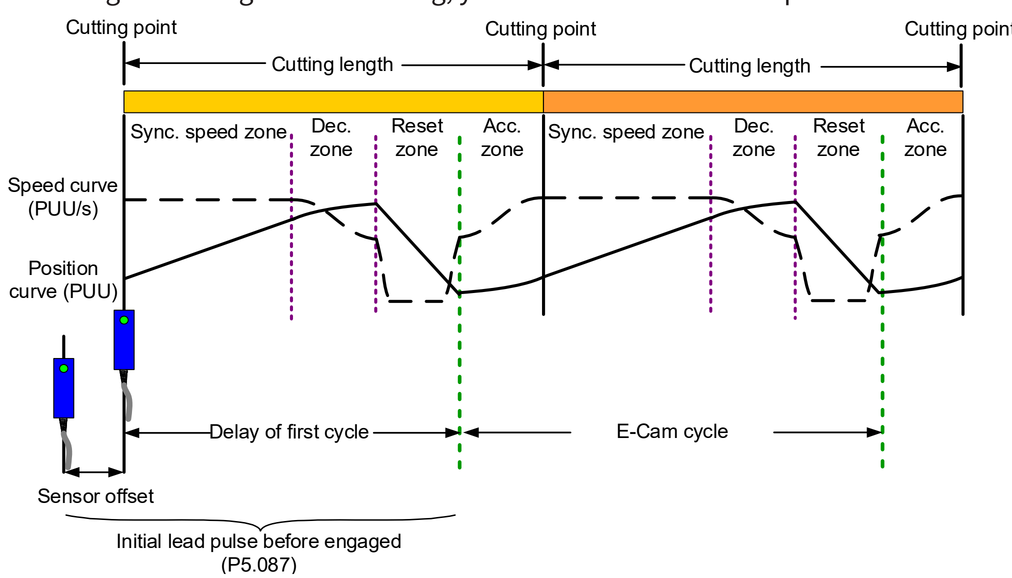
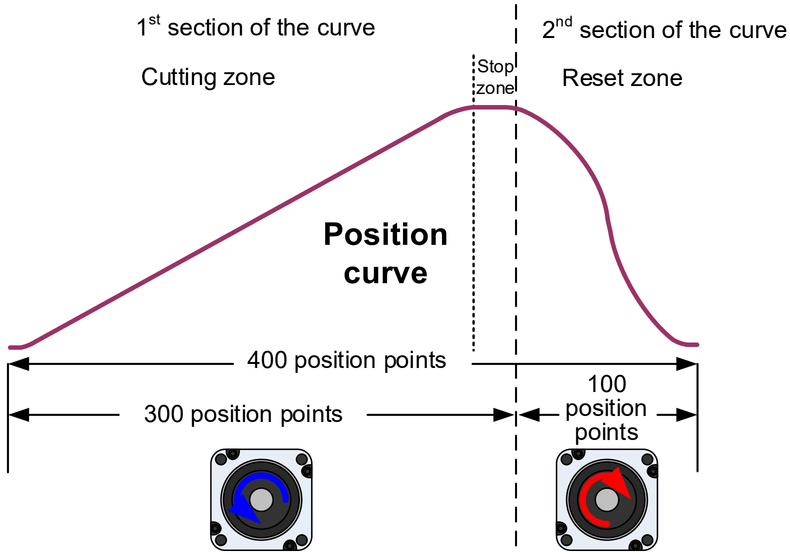


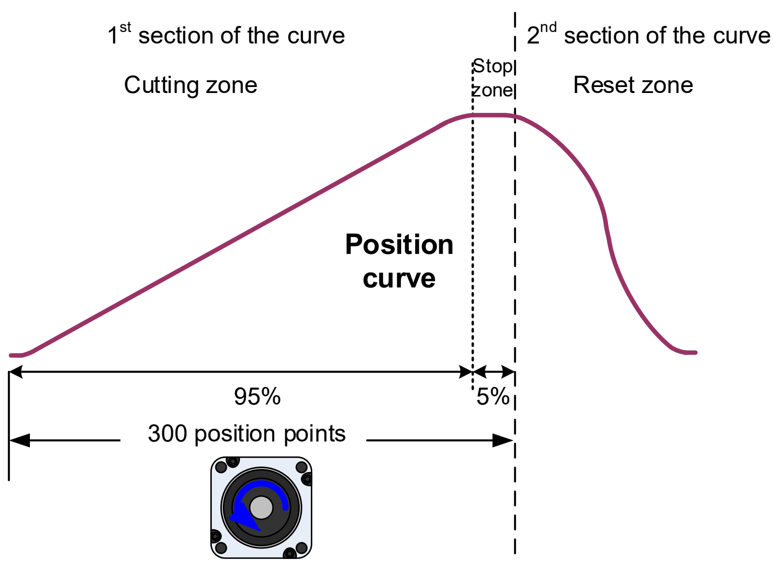
Figure 7-115 Fully engaged E-Cam cycle of the cutting machine

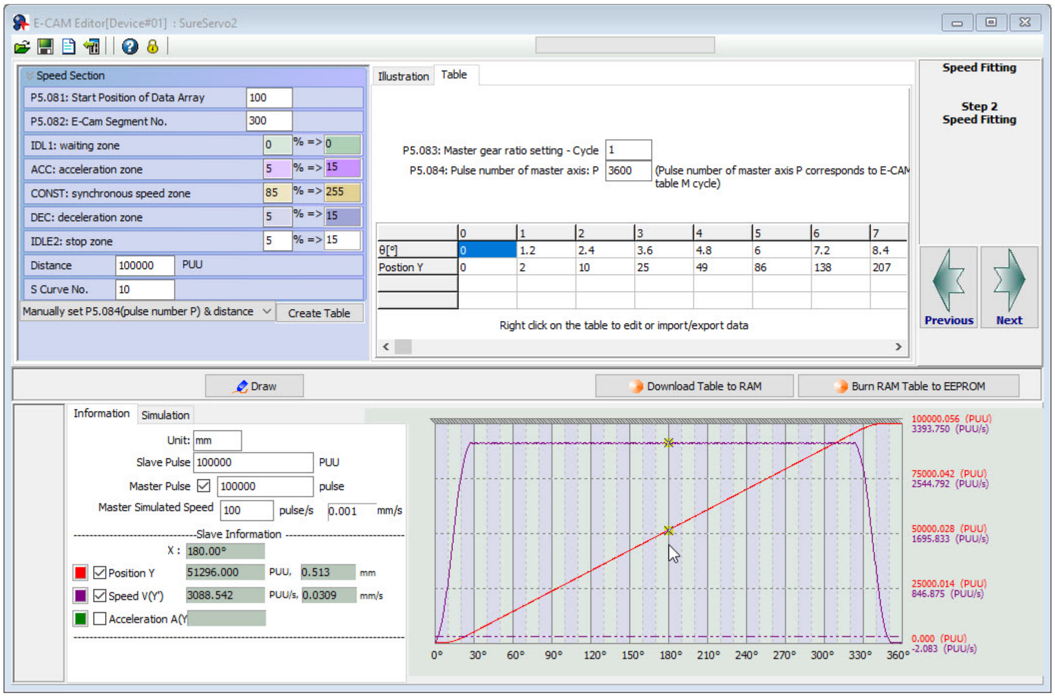
You can use the “Speed Fitting Creation” in SureServo2 Pro to create the E-Cam curve. However, this method can only generate E-Cam curves with single operation direction. You need to create the curves for the cutting zone and reset zone respectively. Then, combine the two curves with the “Manually create a table” function. The operation steps are as follows:

1) Curve planning: segment the cutting zone, stop zone, and reset zone. Because the cutting zone is in the first segment of the curve, there will be more position points plotted in the first segment to ensure the flying shear can complete the cutting in the cutting zone. In the following example, the cutting zone and stop zone are segmented as the first section of the curve with 300 position points. The second section of the curve is the reset zone with 100 position points. So, the curve is composed of 400 position points.

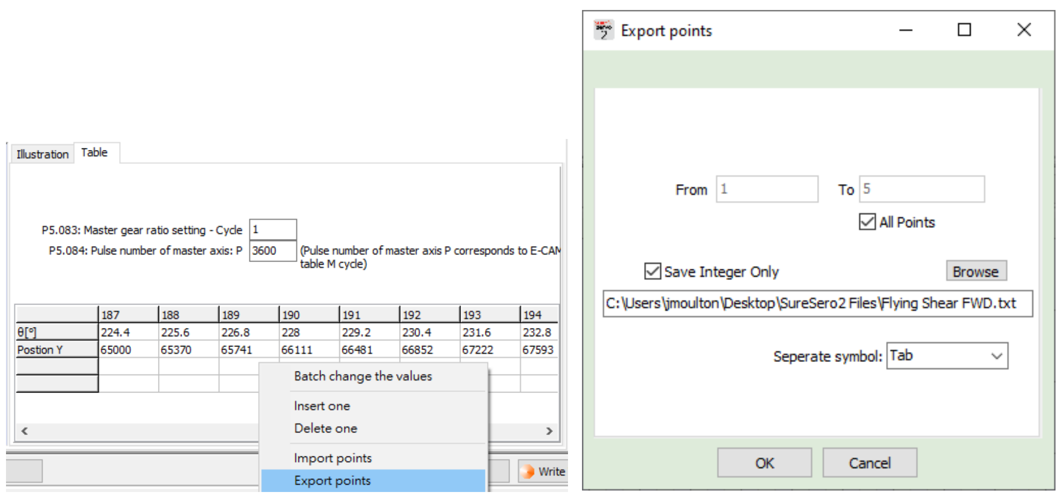


2) Plan and create the first section of the E-Cam curve: this section has a total of 300 position points including the cutting zone and stop zone, so set the E-Cam total segment number N (P5.082) to 300. This example sets the cutting zone to 95% (including acceleration zone of 5%, synchronous speed zone of 85%, and deceleration zone of 5%) which is 285 position points. The stop zone is 5%, which is 15 position points. After setting the required lead distance of the slave axis, click **Create Table**.

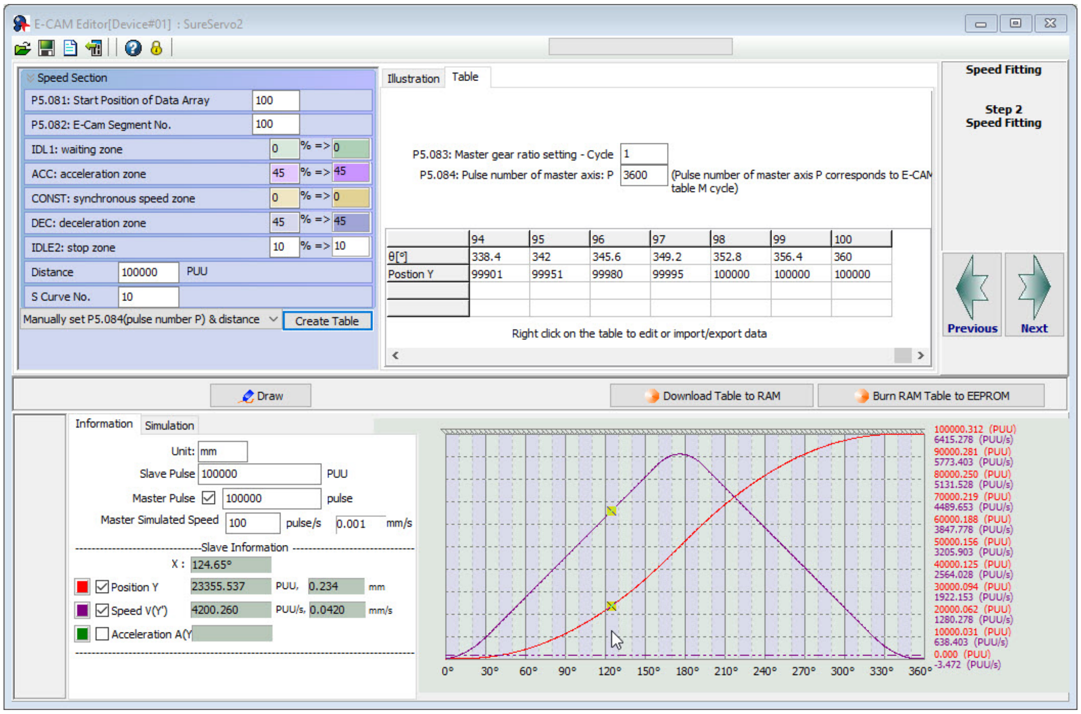




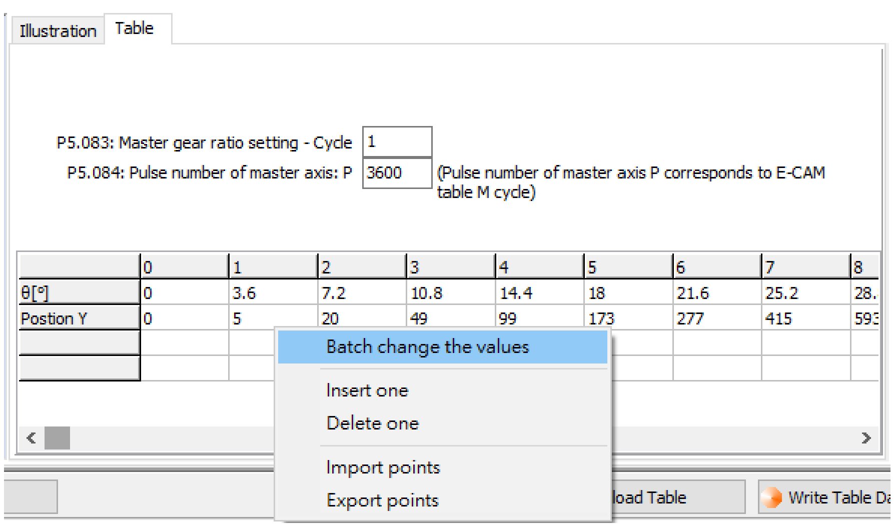
3) Export the curve data of the first section: right-click on the table, select **Export points** and a window appears. Select the check box for **All points** and specify the saving location, then click **OK** to save. And empty .txt file may need to be created first for some Windows versions.

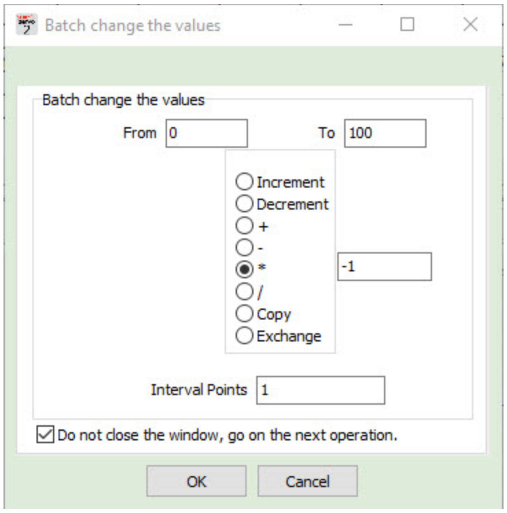


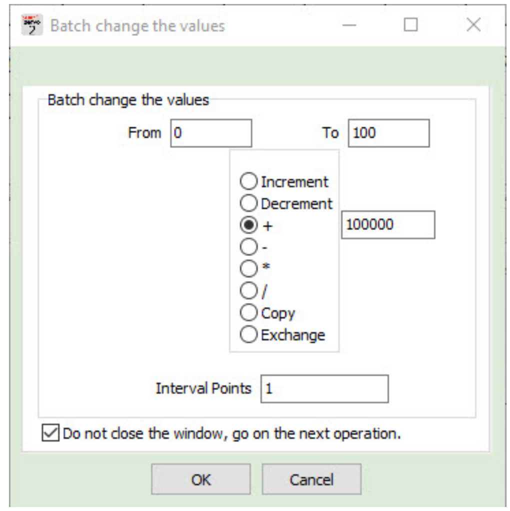
4) Plan and create the second section of the E-Cam curve: the reset zone has 100 position points in total. You must set the E-Cam total segment number N (P5.082) to 100. Since the curves created by this function are all in forward direction, you must first create a curve, and then reverse it to complete the curve for the reset zone.

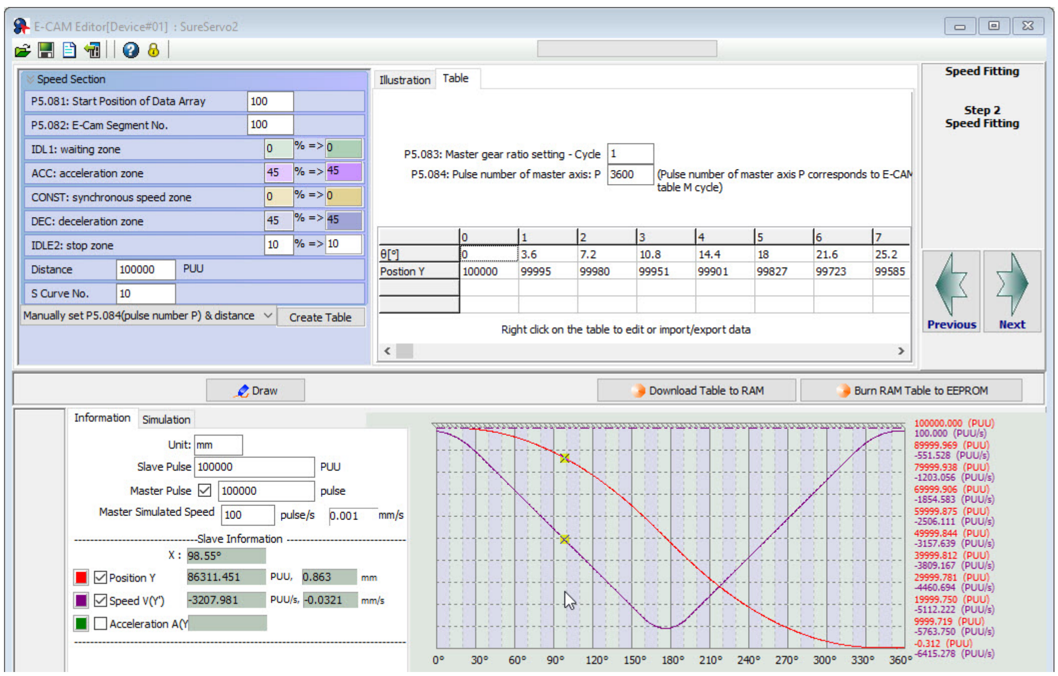


After creating the forward direction E-Cam curve, right-click on the table, select **Batch change the values**, and fill in 0 to 100 in the pop-up window. Select "*" (multiplication), fill in "-1", and select the check box for **Don't close, continue the next operation**, then the curve direction reverses from forward to backward. Then, select "+" (plus), fill in "100000" for the lead distance of 100,000 PUU, so that the initial value of this curve section smoothly coincides with the final value of the previous curve section. Click **OK**, then click on **Draw** to complete the reversed direction of the curve.

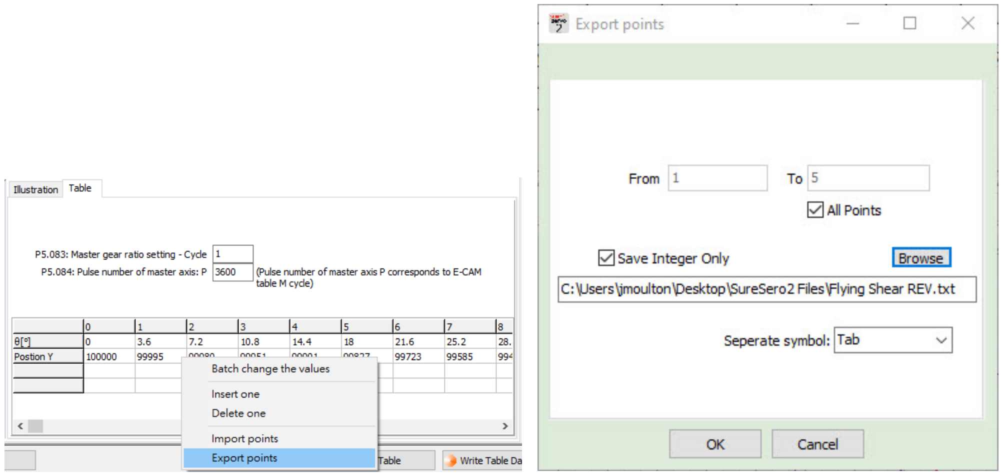



Batch change the values
Batch change the values
From 0
To 100
Increment
Decrement
+
-
*
/
Copy
Exchange
-1
Interval Points 1
Do not close the window, go on the next operation.
OK
Cancel

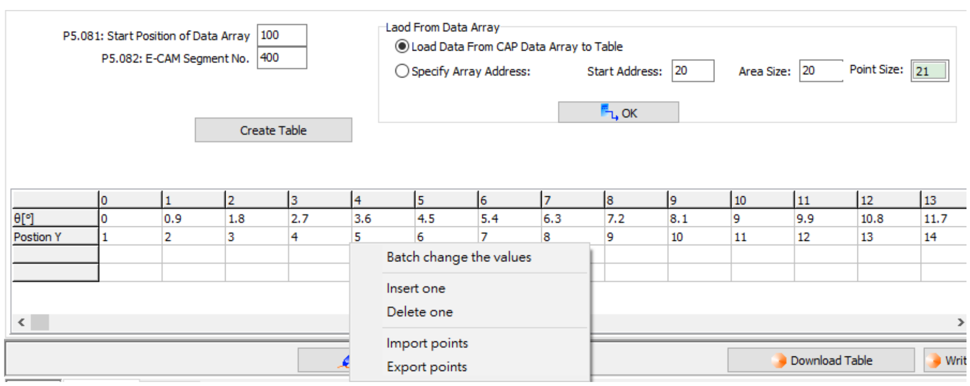

Batch change the values
Batch change the values
From 0
To 100
Increment
Decrement
+
100000
-
*
/
Copy
Exchange
Interval Points 1
Do not close the window, go on the next operation.
OK
Cancel

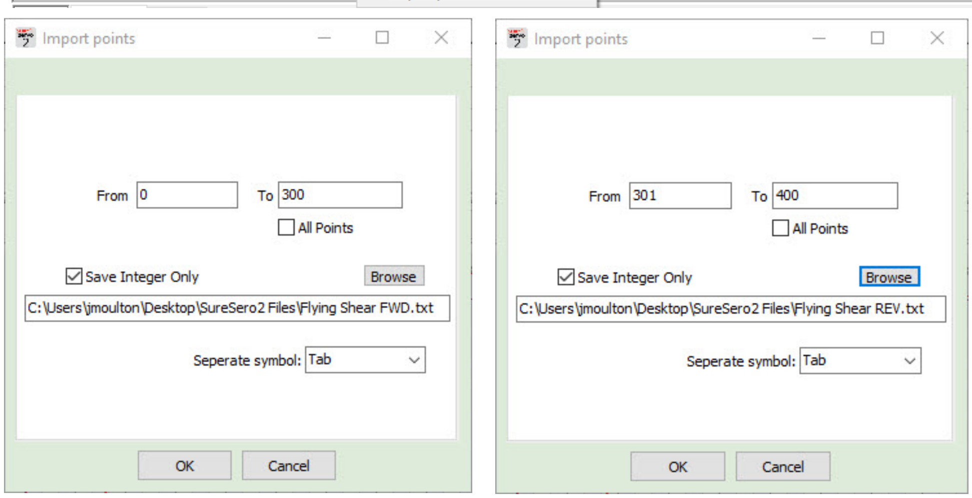

E-CAM Editor[Device#01] : SureServo2
Speed Section
P5.081: Start Position of Data Array 100
P5.082: E-Cam Segment No. 100
IDL1: waiting zone
ACC: acceleration zone
CONST: synchronous speed zone
DEC: deceleration zone
IDLE2: stop zone
Distance 100000 PUU
S Curve No. 10
Manually set P5.084(pulse number P) & distance
Create Table
Illustration
Table
P5.083: Master gear ratio setting - Cycle 1
P5.084: Pulse number of master axis: P 3600
(Pulse number of master axis P corresponds to E-CAM table M cycle)
Right click on the table to edit or import/export data
Speed Fitting
Step 2 Speed Fitting
Previous
Next
Draw
Download Table to RAM
Burn RAM Table to EEPROM
Information
Simulation
Unit: mm
Slave Pulse 100000 PUU
Master Pulse 100000 pulse
Master Simulated Speed 100 pulse/s 0.001 mm/s
Slave Information
X : 98.55°
Position Y 86311.451 PUU, 0.863 mm
Speed V(Y') -3207.981 PUU/s, -0.0321 mm/s
Acceleration A(Y
0° 30° 60° 90° 120° 150° 180° 210° 240° 270° 300° 330° 360°

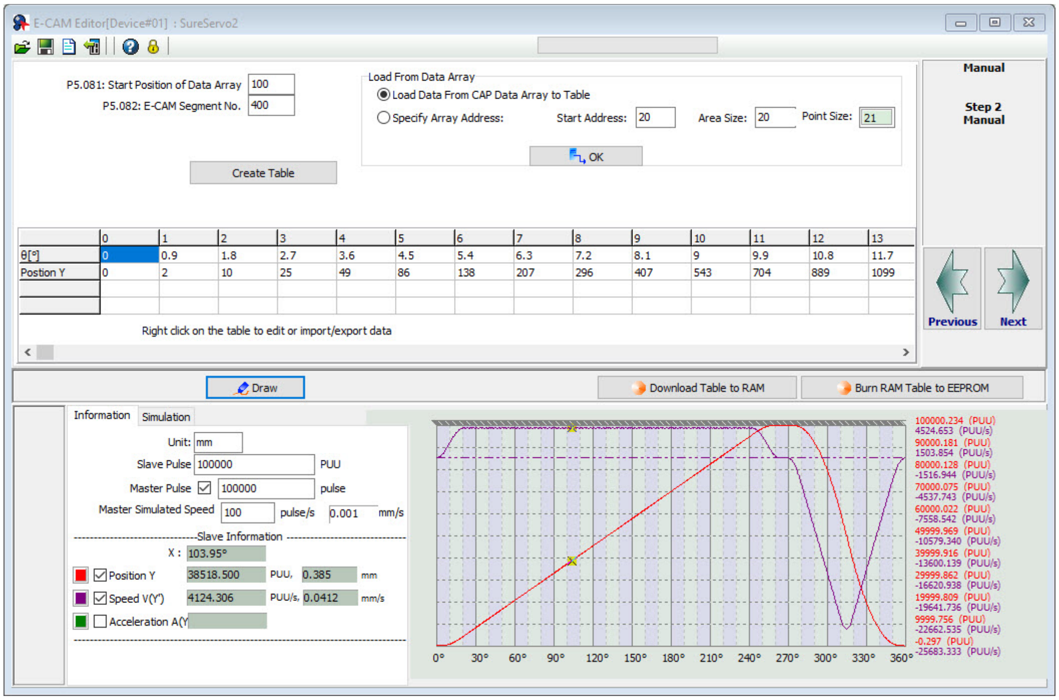
5) Export the curve data of the second section: right-click on the table, select **Export points** and a window appears. Select the check box for **All points** and specify the save location, then click **OK** to save. Ensure the file path also has the correct file name at the end of the path (for example: ...\retractzone.txt).



6) Combine E-Cam curves: go back to the previous screen and use “Manually create a table”. The E-cam curve has a total of 400 position points, so you need to set the E-Cam total segment number N (P5.082) to 400. Click **Create Table** and a table of 400 position points is generated. Right-click on the table, select **Import points**, fill in “1” to “300” in the pop-up window, select the first section of the curve, then click **OK**. Click **Draw** to see the imported points. Follow the same steps as previous, but fill in “301” to “400” in the pop-up window, select the second section of the curve, then click **Draw** to complete the E-Cam curve of the fully engaged mode. Save the .ecd file to preserve the CAM table that was just created.



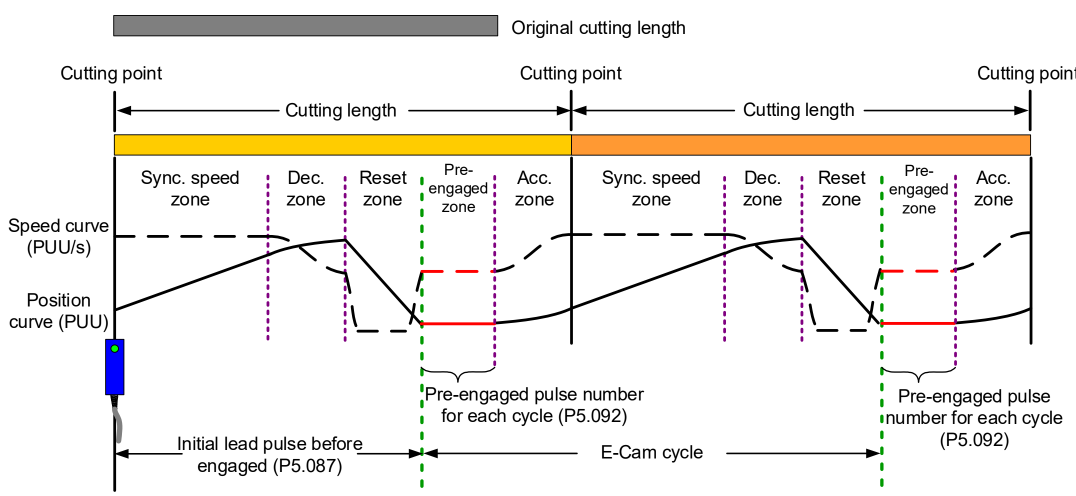




The operation of the fully engaged mode is based on the E-Cam curve. The E-Cam curve is more complex and more difficult to create, so if the cutting length changes, you can only modify the cutting length by setting the pre-engaged pulse number for each cycle (P5.092) or adjust the E-Cam gear ratio (P5.084 / P5.083) and curve scaling (P5.019).

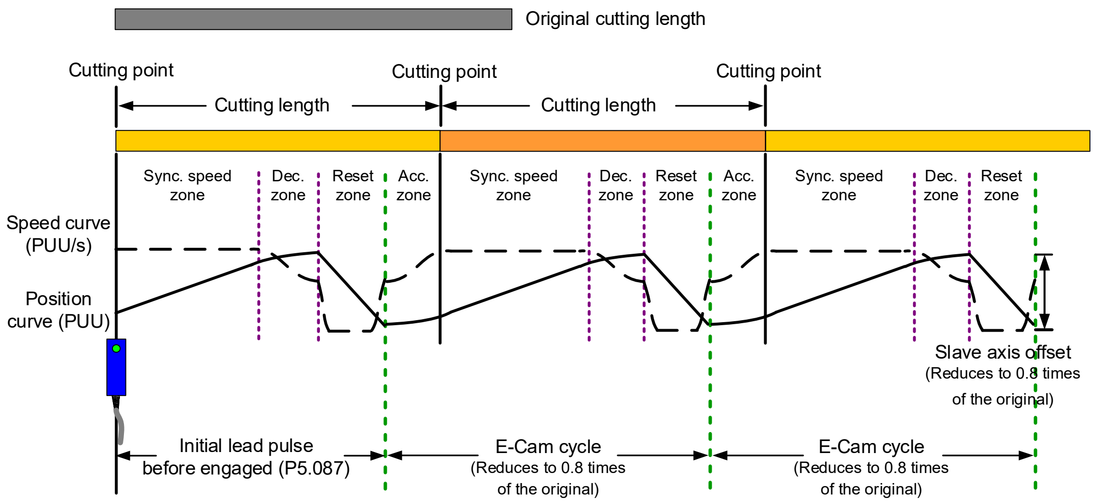
Cutting length is greater than the E-Cam curve operating length

Select to enter cyclic mode after disengaging (P5.088.U = 4) and set the pre-engaged pulse number for each cycle (P5.092). When the E-Cam enters the pre-engaged status, the material feeder continues operating, but the cutting axis stops. The cutting axis resumes operation until the pre-engaged pulse number for each cycle is met. The greater the pre-engaged pulse number for each cycle, the longer the cutting length.



Cutting length is less than the E-Cam curve operating length

Reduce the E-Cam master axis pulse number and slave axis moving distance proportionally. For example, if you reduce the master axis pulse number for each cycle (P5.084 / P5.083) to 0.8 times of the original, the moving distance of the slave axis should also reduce 0.8 times. You can use the E-Cam curve scaling (P5.019) to reduce the setting by 0.8 times to shorten the cutting length. However, this method also reduces the synchronous speed zone, so make sure that the cutting action can be completed in the synchronous speed zone. This method is not recommended for applications when the cutting length is greater than the E-Cam curve operating length. Because when you increase the moving distance of the slave axis, the machine may not have sufficient distance to complete the slave axis motion which can lead to collision.



Partially engaged

This is applicable for cutting operations with or without marks. For the cutting operation with marks, use the Capture function to have the E-Cam engaged. For the cutting without marks, use the Compare function to generate virtual marks for the Capture function to capture the position data. The E-Cam curve includes the acceleration zone, synchronous speed zone, and deceleration zone. After the E-Cam curve (acceleration, synchronous speed, deceleration zones) is executed, the clutch disengages. The reset zone is completed with a PR command and then it waits for the next trigger signal to have the clutch engaged. With this method, you can create an E-Cam curve with a larger synchronous speed zone based on the maximum moving distance of the cutter. This is suitable for applications with the cutting length smaller than the waiting zone. For applications with the cutting length greater than the maximum moving distance of the cutter, disengage the E-Cam and have the material feeder continue to operate, and then the E-Cam re-engages when the servo receives the trigger signal.

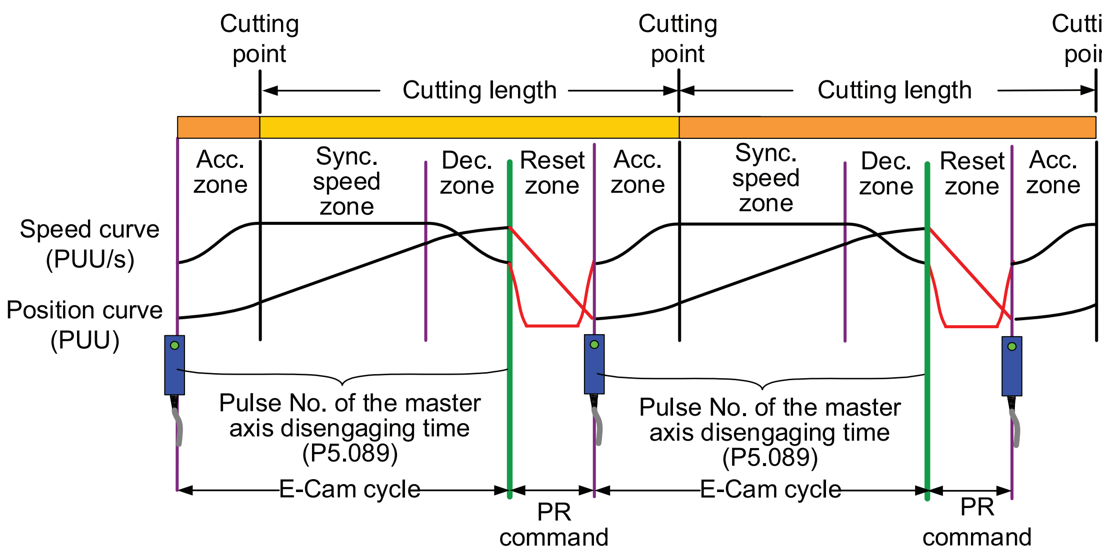
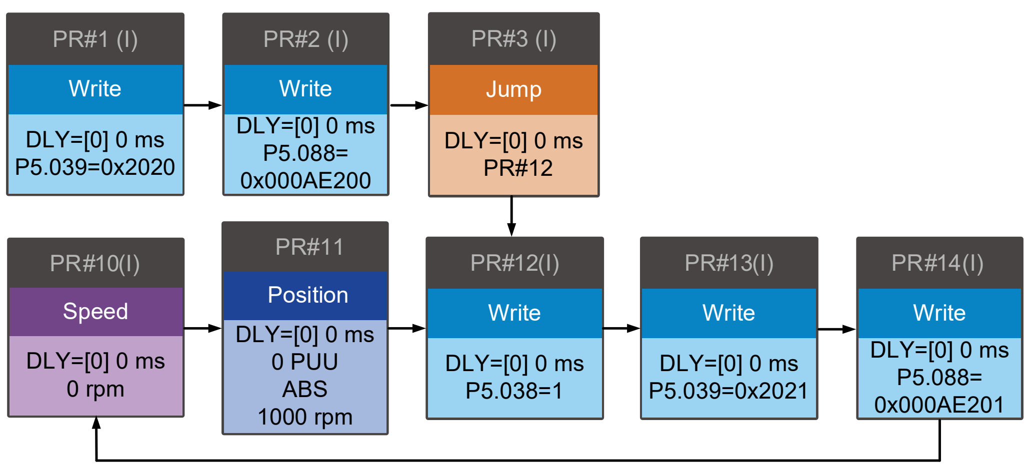


Figure 7-116 E-Cam cycle of the partially engaged cutting machine

After creating an E-Cam curve based on the maximum moving distance of the cutting axis, set the master pulse number of disengaging time (P5.089) according to the cutting length. After reaching the disengaging pulse number or receving the cutting complete signal, the clutch disengages and continues with a zero-speed PR speed command to stop the cutter. Then, it uses another PR position command to return the cutter to the initial position, as shown in Figure 7-117. The setting methods are as follows:

1) Master axis signal source: set P5.088.Y to 0; it means the source of the master axis is the capturing axis. In the Capture function, this capturing axis refers to the setting of P5.039.Y for the signal source of the master axis.
2) Engagement condition: set P5.088.Z to 2; it means the clutch engages as soon as the first data is captured and the signal is input through DI7 to the servo drive.
3) Disengagement condition: set P5.088.U to E; it means the clutch disengages when the master axis pulse number reaches the pulse number set in P5.089, the slave axis decelerates to stop, and the E-Cam function is disabled.
4) To set the subsequent PR procedure after the clutch disengages, set the PR number to be executed in P5.088.BA in hexadecimal.
5) Set the PR procedures:
   - *Procedure 1: set the PR commands for execution when the cutting machine is activated. PR#1 confirms the Capture function is disabled. PR#2 confirms the E-Cam function is disabled. PR#3 jumps to PR#12. PR#12 sets the capturing amount to 1. PR#13 enables the Capture function. PR#14 enables the E-Cam function.*
   - *Procedure 2: set the subsequent PR commands after the clutch disengages. PR#10 sets the zero-speed command to stop the cutting axis. PR#11 sets the position command to return the cutting axis to the initial position.*



Note: P5.088.BA causes a jump PR#10(A) when disengaging.

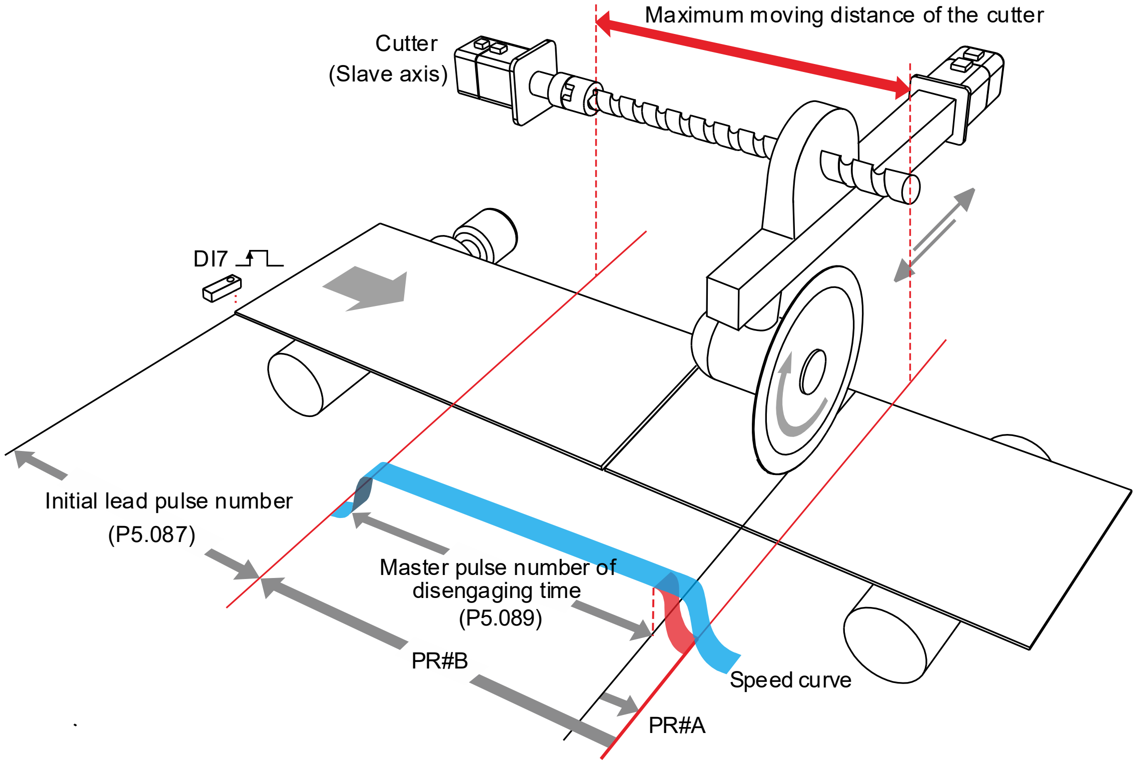


Figure 7-117 Operation of partially engaged cutting axis

7.3.9 - Macro

In real applications, the macro commands cater to different needs during E-Cam operation, such as the requirements for stopping and resuming the operation after an alarm occurs, the phase alignment at the initial operation stage, or phase modification and pausing the cycle during operation. You can use the following macros to complete the tasks. Use P5.097 to enable the macro. Input the values for P5.093 - P5.096 based on the macro requirements.

Macro	Function	Application Requirement
Macro #5	Sets the pulse number for the master axis gear ratio (P5.084) and the pulse number for disengagement (P5.089) as the same.	The cams disengage after one cycle.
Macro #8	When the clutch engages, the E-Cam curve scaling (P5.019) takes effect one time.	Prompt change in scaling.
Macro #C	When the clutch engages, sets the master axis pulse phase when the motor remains unmoved.	Precise control of the clutch engagement position.
Macro #D	When the slave axis position is not in the corresponding E-Cam curve, calculates the position correction amount and writes this amount to the PR incremental position command.	Slave axis position offset correction for E-Cam cycles.
Macro #E	Uses high-speed DI7 to perform E-Cam phase alignment, calculates the compensation amount, and writes this amount to the PR incremental position command.	Non-cyclic marking function.
Macro #F	When the master axis stops and the clutch disengages, moves the slave axis to the specified position and then back to the original position.	Evacuating the damaged material due to mis-cutting.
Macro #10	Carries on the operation after the slave axis stops for one cycle.	Empty pack prevention mechanism.

Macro #C - Change the engagement position and operate in forward direction until the disengagement condition is met

When the clutch is engaged, this macro immediately changes the master axis position and automatically calculates the remaining pulse number in the cycle. When the E-Cam cycle is complete, the clutch disengages based on the set disengagement condition (P5.088.U).
This macro can be used for setting the initial engagement position for the master axis and you can select any of the master axis position to engage. The precision level is higher when you use Macro #C than using P5.085 to select the section from the E-Cam table for engagement. When using this macro, the master axis should stay stationary. Wait for the macro to complete before operating the master axis. The operation is shown in Figure 7-118.

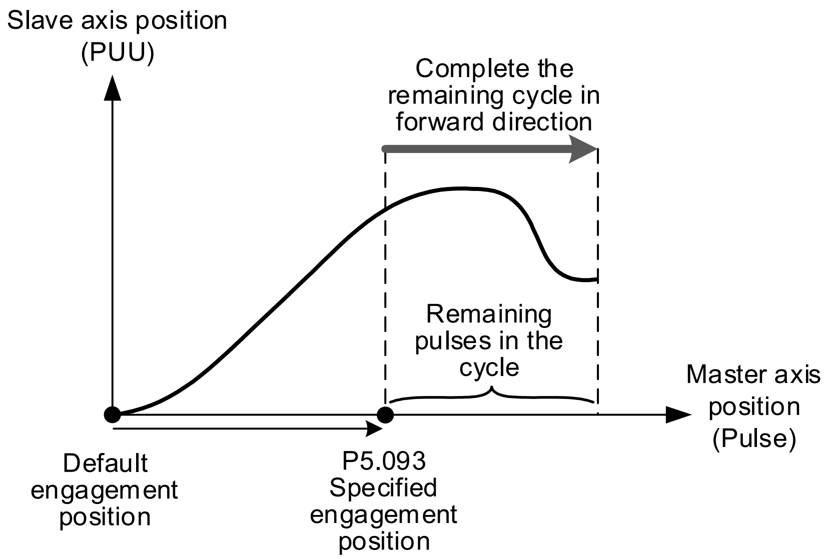


Figure 7-118 Macro #C operation

Setting steps:

1) The E-Cams engage and the master axis stops.
2) Set the disengagement condition (P5.088.U).
3) Set the engagement position: use P5.093 to write the master axis engagement position (pulse) in hexadecimal and use monitoring variable 062(3Eh) to monitor the current master axis position. The specified range for the new engagement master axis position is:

$$0 \le \text{P5.093 (Pulse)} < \frac{P5.084}{P5.083}$$

4) Enable Macro #C: set P5.097 = 0x000C to enable Macro #C. Read P5.097 and if it returns 0x100C, it means the macro execution is successful. If any of the following failure codes shows, modify the setting according to the description.

Failure Code	*Description*
0xF0C1	When executing this macro command, the clutch is not in the engaged status.
0xF0C2	The engagement position specified in P5.093 exceeds the range (must be ≥ 0).
0xF0C3	The engagement position specified in P5.093 exceeds the range (must be < (P5.084/P5.083)).

Macro #D - calculate the deviation between the current slave axis position and index coordinate for PR positioning

When the slave axis position is not at the E-Cam curve corresponding position, this macro finds the slave axis position corresponding to the master axis position. Next, it calculates the deviation between this value and the current motor position, and writes the deviation to the PR incremental position command. You can trigger the specified PR and move the motor of the slave axis to the position corresponding to the master axis position. This macro is suitable for the cyclic motion which starts from the same point. In other words, the mechanism returns to the start point each cycle; and the slave axis moving distance is the same as the total index moving distance. You can monitor the index coordinate in PUU with monitoring variable 091(5Bh). The operation is shown in Figure 7-119.

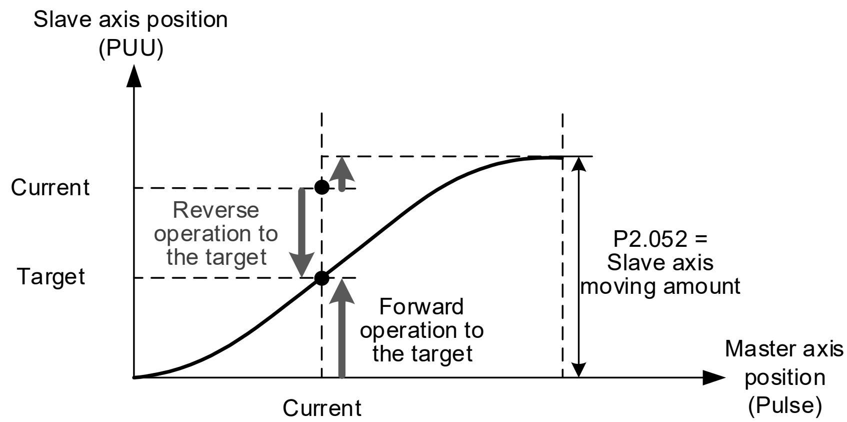
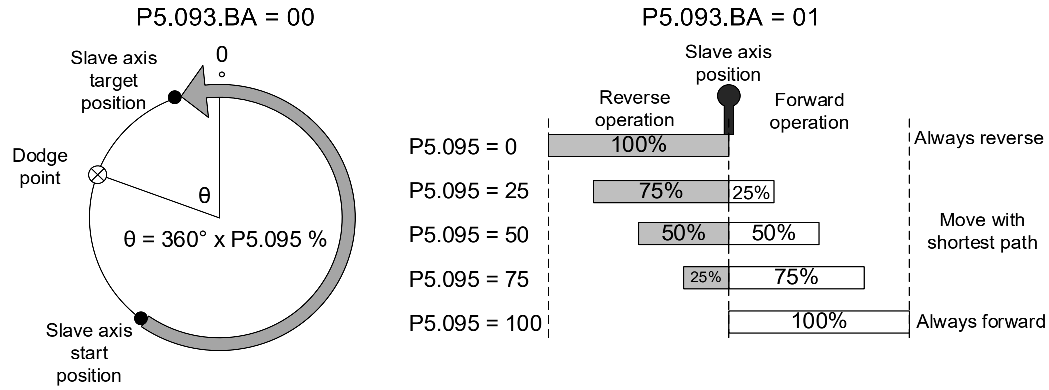


Figure 7-119 Macro #D operation

Setting steps:

1) Set P5.088.X Bit1 to 1 to keep the clutch engaged when Servo Off and engage the E-Cam.
2) Set the total index moving distance to equal the moving distance of the slave axis per cycle (P2.052 = slave axis moving distance ECAM_H).
3) Set the E-Cam scaling to 1 (P5.019 = 1).
4) Set the initial engagement position: align the start point of 0 degree in the E-Cam curve table with the index coordinate origin.
5) Set the PR number to save the deviation: specify PR#1 - 99 in hexadecimal.
   Set P5.093.YX = 0x01 - 0x63, and set this PR as an incremental position command.
6) Select the direction control type: set P5.093.BA = 00 to use the dodge point for controlling the forward and reverse directions. Set P5.093.BA = 01 to use the allowable forward rate for controlling the forward and reverse directions.
7) Set the reverse inhibit function: set P5.093.CD = 0 to disable the reverse inhibit function. Set P5.093.CD = 1 to enable the reverse inhibit function.
8) Set the dodge point or allowable forward rate: if using the dodge point for direction control, set P5.095 to 0 - 100% for the dodge position. If using the allowable forward rate for direction control, set P5.095 to 0 - 100% as the allowable forward rate. Refer to the following figure.



9) Enable Macro #D: set P5.097 = 0x000D to enable Macro #D. Read P5.097 and if it returns 0x100D, it means the macro execution is successful. If any of the following failure codes shows, modify the setting according to the description.

Failure Code	*Description*
0xF0D1	When executing this macro command, the clutch is not in the engaged status.
0xF0D2	PR number specified by P5.093.YX exceeds the range (0x01 - 0x63).
0xF0D3	P5.095 the dodge point or allowable forward rate exceeds the range (0 - 100%).
0xF0D5	Position correction value does not exist. This macro command might be triggered twice.
0xF0D6	E-Cam did not remain engaged when servo is off, so when servo switches to the on state again, E-Cam is not engaged.
0xF0D7	Slave axis moving distance does not equal the total index moving distance (ECAM_H ≠ P2.052).
0xF0D8	E-Cam curve scaling does not equal 1 (P5.019 ≠ 1).
0xF0D9	P5.093.BA forward / reverse direction setting exceeds the range (00 - 01).
0xF0DA	P5.093.DC reverse inhibit setting exceeds the range (00 - 01).
0xF0DB	The reverse inhibit function has failed. Do not use macro command #D and #10 consecutively.

Macro #E - PR positioning using E-Cam correction amount

When the clutch engages, this macro sets the master axis engagement position (pulse) and calculates the required correction amount for the slave axis to complete positioning for one time. Next, it writes this correction amount to the PR incremental position command to execute.

You can trigger this PR command to have the slave axis operate to the corresponding target position at the proper time. In actual applications, you can connect the external sensor to the servo drive DI, and use an event trigger to enable Macro #E. Then, the macro calculates the correction amount and writes this value to the specified PR program. This macro is suitable for applications with random markings. The operation is shown in Figure 7-120.

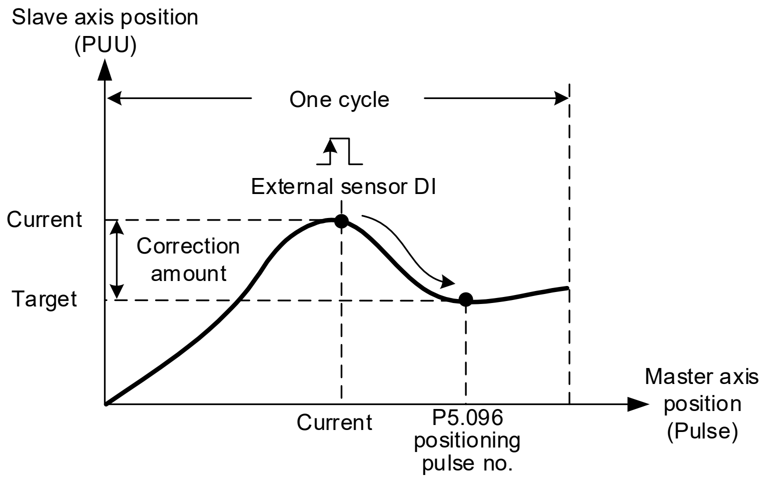
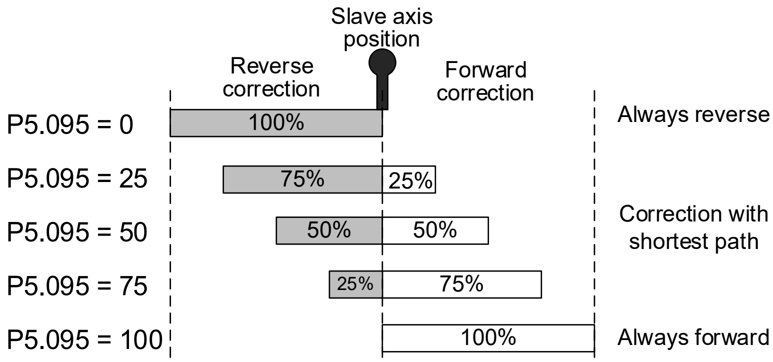


Figure 7-120 Operation of Macro #E

Setting steps:

1) Engage the clutch.
2) Set the PR number to save the correction amount: specify PR#1 - 99 in hexadecimal. Set P5.093.YX = 0x01 - 0x63, and set this PR as an incremental position command.
3) Set the maximum correction rate: specify the maximum correction rate of 0 - 100% in hexadecimal. When P5.093.UZ = 0x00 - 0x64%, it limits the correctable range to avoid over-correction per time and causing machine vibration.
4) Set the PR trigger timing: set P5.093.A to 1 to immediately trigger the PR command for correction. Set P5.093.A to 0 to manually trigger the PR command.
5) Set the mark position: set P5.093.B to 0 to mark on other motion axis and the following mark positions are not changed when positioning. Set P5.093.B to 1 to mark on the motion axis for compensation, but this changes the following mark positions when positioning.
6) Set the triggering method: set P5.093.C to 0 to use the general DI with event triggering. Set P5.093.C to 1 to use the high speed DI7 with Capture function as the triggering method; meanwhile, set the pulse source of the master axis (P5.088.Y = 0) as the Capture axis. When the last position data is captured, execute PR#50 (P5.039.X Bit3 = 1) to perform the compensation. This is suitable for high precision applications.
7) Set the compensation for the DI time delay: set P5.094 as -25000 to 25000 (µs) to compensate the delay time for the sensor and the signal transmission.
8) Set the allowable forward rate: set P5.095 to 0 - 100% to specify the allowable forward rotation rate.



9) Set the positioning pulse number: use P5.096 to set the pulse number (position) of the master axis for positioning. The setting range is as follows:

$$0 \leq \text{P5.096 (Pulse)} < \frac{P5.084}{P5.083}$$

10) Enable Macro #E: set P5.097 = 0x000E to enable Macro #E. Read P5.097 and if it returns 0x100E, it means the macro execution is successful. If any of the following failure codes shows, modify the setting according to the description.

Failure Code	***Description***
0xF0E1	When executing this macro command, the clutch is not in the engaged status.
0xF0E2	PR number specified by P5.093.YX exceeds the range (0x01 - 0x63).
0xF0E3	P5.093.UZ maximum correctable rate exceeds the range (0x00 - 0x64%).
0xF0E4	P5.094 DI delay time compensation exceeds the range (-25000 to 25000 µs).
0xF0E5	P5.095 allowable forward rate exceeds the range (0 - 100%).
0xF0E6	P5.096\ pulse number (position) of the master axis for positioning exceeds the range (0 ≤ P5.096).
0xF0E7	P5.093 setting value exceeds the range (0x0000 - 0x0111).
0xF0E8	When using DI7 with the Capture function for triggering (P5.093.C = 1), the Capture axis has to be the source pulse of the master axis (P5.088.Y = 0).
0xF0E9	When using DI7 with the Capture function for triggering (P5.093.C = 1), execute PR#50 (P5.039.X Bit3 = 1) for compensation after the last data is captured.

Macro #F - use the deviation between the current slave axis position and the target position for PR positioning

When the master axis stops but the E-Cams remain engaged, this macro can move the slave axis to the specified position and then return it to the original position. The specified position is specified with the master axis pulse number. After Macro #F is triggered, the servo calculates the required moving amount for the slave axis to move to the specified position and writes this moving amount to the two PR incremental position commands (onward and return trips). Trigger the onward trip PR command, and the slave axis moves to the target position. Trigger the return PR command, and the slave axis returns to the original position. This macro is suitable for applications that require moving the slave axis while the system or the master axis is stopped. The operation is shown in Figure 7-121.

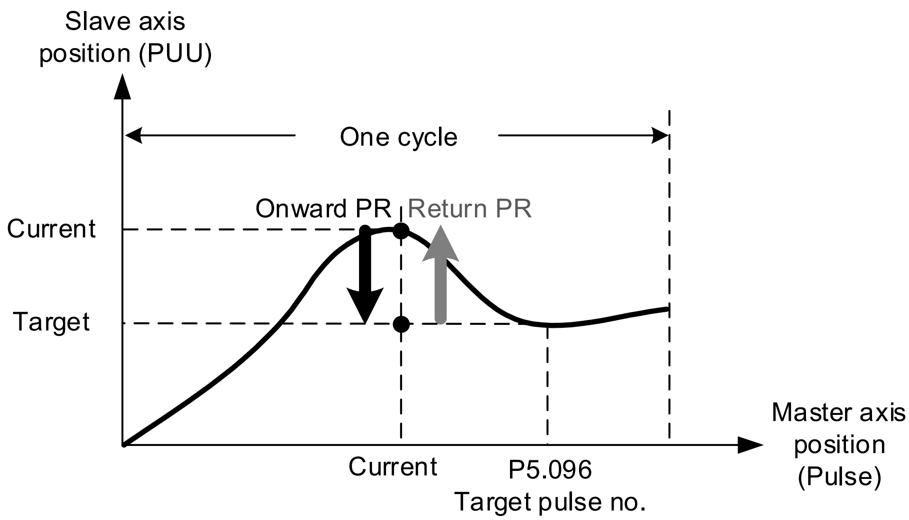
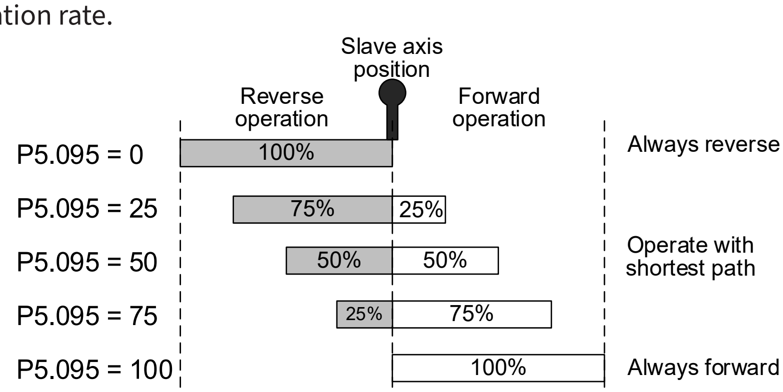


Figure 7-121 Macro #F operation

Setting steps:

1) The master axis stops and the clutch is engaged.
2) Set the onward and return trip PR numbers: specify any of the PR from PR#1 - 99 in hexadecimal as the onward trip PR command. Set P5.093.YX = 0x01 - 0x63 and set this PR as the incremental position command. Set any of the PR from PR#1 - 99 as the return trip PR command. Set P5.093.UZ = 0x01 - 0x63 and set this PR as the incremental position command. Do not use the same PR number at the same time.
3) Set the allowable forward rate: set P5.095 to 0 - 100% to specify the allowable forward rotation rate.



4) Set the target pulse number: use P5.096 to specify the master axis pulse number of the target position, which range is as follows:

$$0 \le P5.096 \text{ (Pulse)} < \frac{P5.084}{P5.083}$$

5) Enable Macro #F: set P5.097 = 0x000F to enable Macro #F. Read P5.097 and if it returns 0x100F, it means the macro execution is successful. If any of the following failure codes shows, modify the setting according to the description.

Failure Code	***Description***
0xF0F1	When executing this macro command, the clutch is not in the engaged status.
0xF0F2	PR number of onward trip specified by P5.093.YX exceeds the range (0x01 - 0x63).
0xF0F3	PR number of return trip specified by P5.093.UZ exceeds the range (0x01 - 0x63).
0xF0F5	P5.095 allowable forward rate exceeds the range (0 - 100%).
0xF0F6	P5.096 master axis pulse number of the target position exceeds the range (0 ≤ P5.096).

Macro #10 - the slave axis immediately pauses for one cycle

When the clutch is engaged and the slave axis operates in forward direction, this macro can stop one cycle of the slave axis operation and then the operation resumes. To stop for multiple cycles, consecutively trigger Macro #10 for a number of times. The servo drive records the number of times Macro #10 is triggered and the slave axis will stop for the number of cycles accordingly. When using this macro, use P1.022 PR special filter and set P1.022.YX acceleration time limit (the required time for the motor to accelerate from 0 to 3000 rpm, that range is 10–1270 ms).

If the acceleration or deceleration time is shorter than the acceleration time limit, then the filter takes effect and smooths the acceleration or deceleration process, preventing the command from changing too drastically and machine vibration. The following error caused by the smooth command will be compensated after the command changes become moderate, so the final position does not deviate. This macro is usually used for the empty pack prevention function on the packing machine. The operation is shown in Figure 7-122.

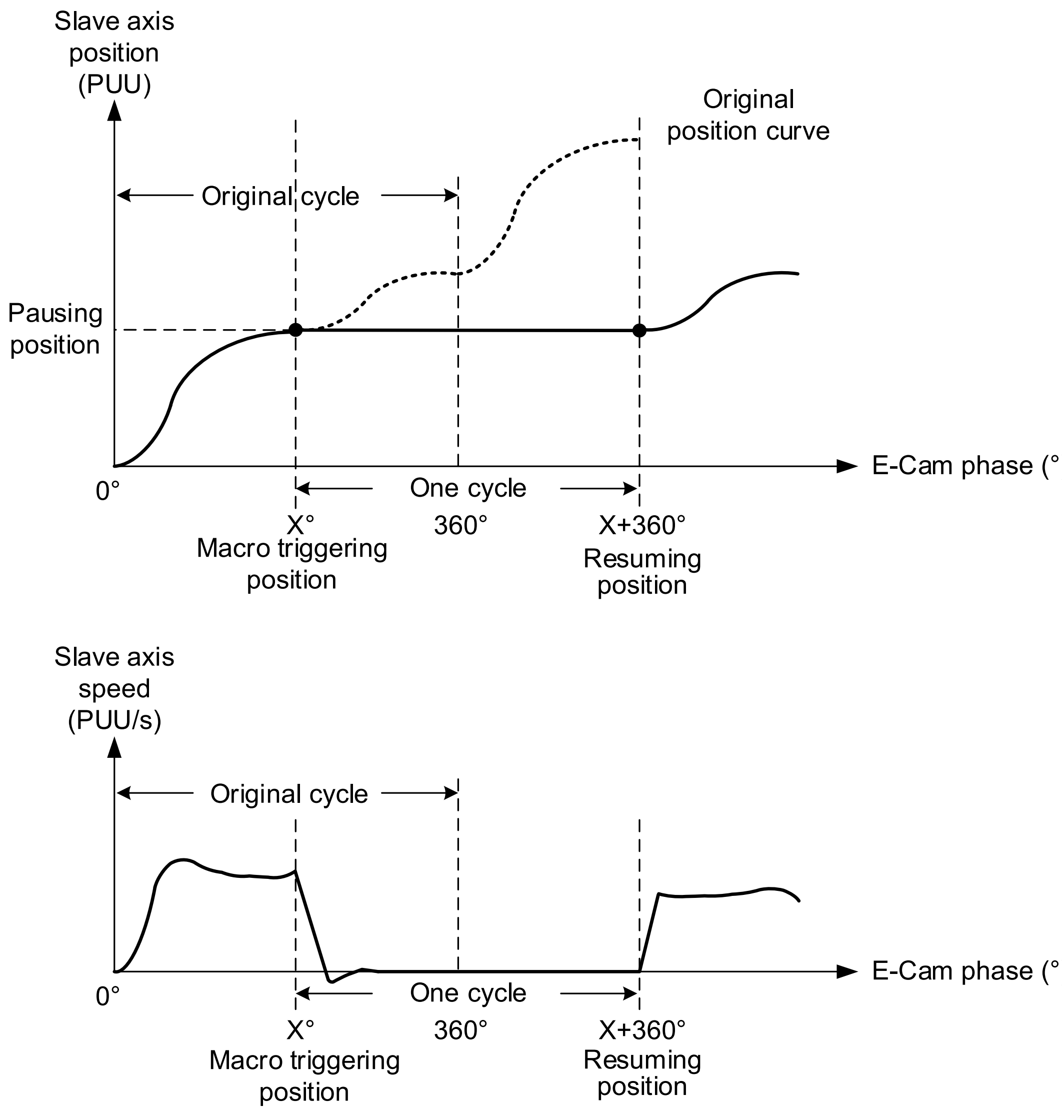


Figure 7-122 Macro #10 operation

Setting steps:

1) Engage the clutch.
2) Set P1.022.YX the acceleration time limit. If the reverse inhibit is required, set P1.022.U.
3) Enable Macro #10: set P5.097 = 0x0010 to enable Macro #10. Read P5.097 and if it returns 0x1010, it means the macro execution is successful. If any of the following failure codes shows, modify the setting according to the description.

Failure Code	***Description***
0xF101	When executing this macro command, the clutch is not in the engaged status.
0xF102	Set P5.093 to 0.
0xF103	The slave axis must operate in forward direction. Check the E-Cam curve and P5.019 E-Cam curve scaling.
0xF104	Accumulated pause distance exceeds 2^{31}. Do not execute this macro command consecutively.

NOTE: *This function is accumulative. If the command is triggered for N times consecutively, it pauses the E-Cam for N cycles. Note that the accumulated pause distance cannot exceed the range. When the pause cycle is complete, the slave axis continues to operate and the accumulated pause distance is cleared to 0.*

Wiring
Parameters
DI/DO Codes
Monitoring
Alarms

7.3.10 - Auxiliary function

Following error compensation

There are two factors causing the following error. The first is the servo error, which is generated by the position loop and can be eliminated by the position integral compensation (P2.053).

The second is the command processing delay, which is the delay caused by the filter or command. For the general point-to-point motion, the servo waits for the positioning complete signal and then proceed to the next command. This does not generate too much following error and affect the motion. However, for E-Cam applications, you must reduce the following error, or the E-Cam phase can deviate, and thus reducing the machining precision.

To enable the following error compensation function, set P1.036 to 1. Meanwhile, set P1.008 (Position command smoothing constant) to 0ms. Enable the position command moving filter (P1.068) and set the value to less than 10ms. Set the position integral compensation (P2.053) to less than 50. If you are not satisfied with the performance in the acceleration or deceleration stage, adjust the command response gain (P2.089) to reduce the following error. To have better performance in the synchronous speed zone, set the additional compensation time (P1.017) to compensate the deviation. The formula is as follows.

Compensation distance = P1.017 (additional compensation time) current motor speed

Excluding the following error caused by the machine, if the error is proportional to the speed (for example: 100 rpm with an error of 0.01%; 1000 rpm with an error of 0.1%), it could be caused by the electrical delay. In this case, use P1.018 and P1.021 to compensate the E-Cam phase. The compensation mechanism is as follows.

Compensation amount (pulse) = P1.018 (compensation time) x [Master axis pulse frequency (Kpps) – P1.021 (Minimum frequency of pulse compensation for the E-Cam master axis)]

The master axis pulse frequency can be monitored with monitoring variable 060(3Ch), which value has to be greater than the minimum compensation frequency.

Virtual master axis

During E-Cam operation, if there is a following error in the slave axis, use the virtual master axis to correct the cam phase. Virtual master axis operation is as shown in Figure 7-123.

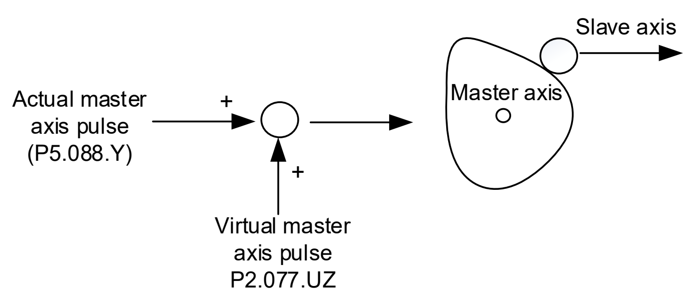


Figure 7-123 Virtual master axis operation

Use P2.077 to set the virtual master axis function. P2.077.X can mask the actual master axis pulses and determine how the virtual master axis pulses are generated, as shown in the following table.

X	***Function***	***Actual Master Axis Pulse***	***Virtual Master Axis Pulse***	***Description***
0	Function disabled	Receive actual master axis pulse	Disable	Slave axis operates based on the actual master axis pulses.
1	Mask the master axis pulses	Masked		Slave axis stops operating, but the masked master axis pulse number continues to be stored in the internal variable.
2	Continuous forward running		Enable	Command source is the virtual pulse frequency set in P2.077.UZ (unit: Kpps). This virtual pulse function continues to operate. To stop it, set X to 1.
3	Continuous reverse running			
4	Forward JOG			Command source is the virtual pulse number set in P2.077.UZ (unit: pulse). This function only refers to the pulse number set in P2.077.UZ.
5	Reverse JOG			
6 - 8	-	-	-	-
9	Master axis pulse masked	Receive actual master axis pulse	Disable	Slave axis operates based on the actual master axis pulses. The master axis pulse number continues to be stored in the internal variable.
A	Continuous forward running		Enable	Command source is the frequency transmitted by the actual master axis (P5.088 Y) plus the virtual pulse frequency in Kpps set by P2.077.UZ. This virtual pulse function continues to operate. To stop it, set X to 9.
B	Continuous reverse running			
C	Forward JOG			Command source is the pulse transmitted by the actual master axis (P5.088.Y) plus the virtual pulse number in pulses set by P2.077.UZ. This function is often used for dynamic adjustment.
D	Reverse JOG			

P2.077.Y sets whether to write the pulse number of the virtual master axis to P5.087 (initial lead pulse before engaged).

- *When the setting of P2.077.Y is changed from 0 to 1, write the pulse number of the virtual master axis to P5.087.*
- *When the setting of P2.077.Y is changed from 0 to 2, write the pulse number of the virtual master axis to P5.087 and store in EEPROM as non-volatile data.*
- *When the setting of P2.077.Y is changed from 0 to 7, write the pulse number of the virtual master axis plus the pulse number of one cycle to P5.087 and store in EEPROM as non-volatile data. The value written to P5.087 has to be positive. When the pulse number of the virtual master axis is negative, the system automatically makes it a positive number by adding the master axis pulses of one or multiple cycles and then write this value to P5.087.*
- *The virtual master pulse number set in P2.077.UZ is hexadecimal. If selecting continuous forward or reverse operation, the unit is Kpps. If selecting jog operation in forward or reverse direction, the unit is pulse.*

BLANK
PAGE