

Basic of servomotor control

NPM Nippon Pulse Motor Co.,Ltd.

Table of contents

I. Connection and operation differences in stepper motors and servomotors	1
Stepper motor	1
Servo motor	1
II. Advantages and disadvantages of stepper motors and servomotors	2
1. Stepper motor	2
2. Servomotor	2
III. What is a deflection counter?	3
IV. Output signals from an encoder	4
1. A phase and B phase signals	4
2. Z phase	4
V. Multiplication	5
(1) 1x	5
(2) 2x	5
(3) 4x	5
VI. INP (in-position) signal	6
* Setting to the delay operation complete status until the INP signal goes ON.	7
* Setting to enable output of an INT signal when the INP signal goes ON.	7
* Setting the INP signal input logic	7
VII. ERC (deflection counter clear) signal	8
* Condition settings for outputting the ERC signal.	8
* Setting the output logic of the ERC signal	9
* Set the ERC signal output length	9
VIII. ALM (alarm) signal	10
* Setting for the stop method when the ALM signal is ON	10
* Setting the input logic for the ALM signal	10
* Read the cause of the ALM signal	10

Basic of servomotor control

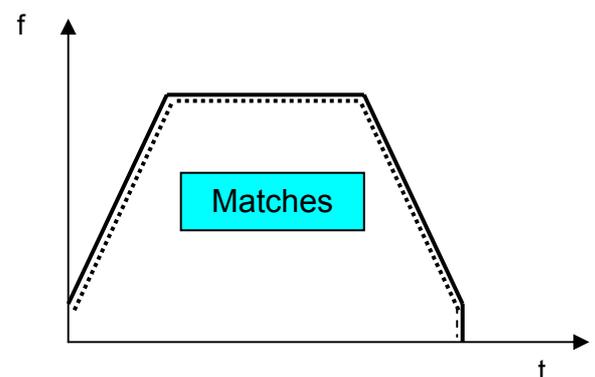
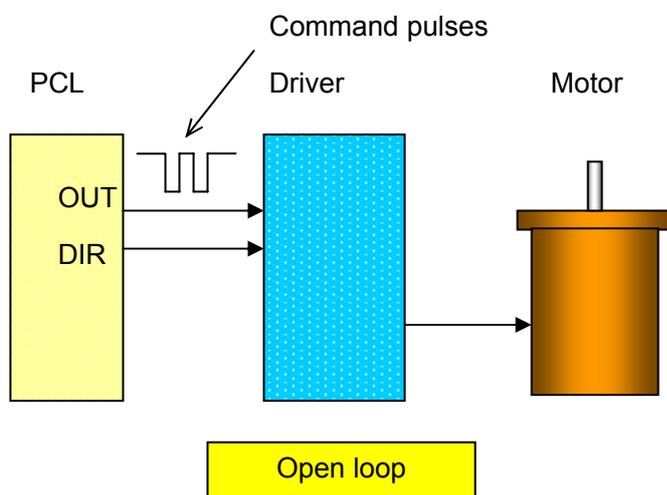
This report explains the difference between a servomotor and a stepper motor when connected to a servo driver. It covers the terms used in controlling the pulse train supplied to servomotors by a PCL series controller.

Please note that this pamphlet does not explain the principles of operation or the design of the motors and drivers themselves.

- The difference between a stepper motor and a servomotor configuration is shown below. The design and construction of the motors are also different.
- Stepper motor operation is synchronized by command pulse signals output from the PCL or a pulse generator (strictly speaking it follows the pulses). In contrast, servomotor operation lags behind the command pulses.

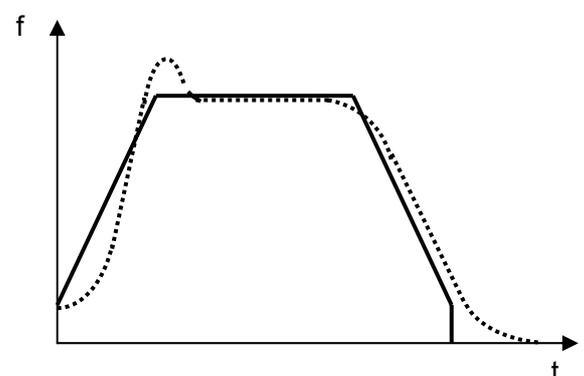
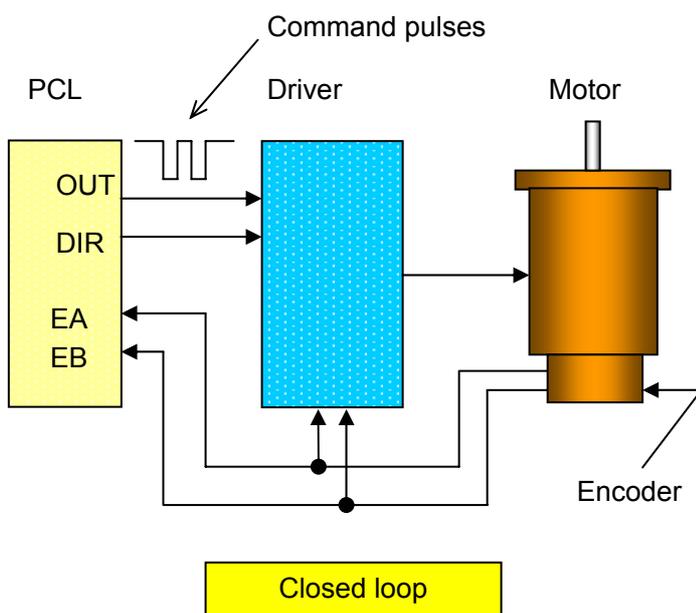
1. Connection and operation differences in stepper motors and servomotors

Stepper motor



Operates by following the command pulses.

Servomotor



II. Advantages and disadvantages of stepper motors and servomotors

1. Stepper motor

<Advantages>

- (1) Since stepper motor operation is synchronized with the command pulse signals from a pulse generator such as the PCL, they are suitable for precise control of their rotation.
- (2) Lower cost.

<Disadvantages>

- (1) Basically, the current flow from a driver to the motor coil cannot be increased or decreased during operation. Therefore, if the motor is loaded with a heavier load than the motor's designed torque characteristic, it will get out of step with the pulses.
- (2) Stepper motors produce more noise and vibration than servomotors.
- (3) Stepper motors are not suitable for high-speed rotation.

2. Servomotor

<Advantages>

- (1) If a heavy load is placed on the motor, the driver will increase the current to the motor coil as it attempts to rotate the motor. Basically, there is no out-of-step condition. (However, too heavy a load may cause an error.)
- (2) High-speed operation is possible.

<Disadvantages>

- (1) Since the servomotor tries to rotate according to the command pulses, but lags behind, it is not suitable for precision control of rotation.
- (2) Higher cost.
- (3) When stopped, the motor's rotor continues to move back and forth one pulse, so that it is not suitable if you need to prevent vibration.

Both motors have advantages and disadvantages. The selection of which type to use requires careful consideration of the application's specifications.

Below is a summary of the comparison of stepper motors and servomotors

	Stepper motor	Servomotor
Drive circuit	Simple. The user can fabricate it. (The customer just purchases a motor, creates a control circuit and starts the motor rotating.)	Since the design is very complicated, it is not possible to fabricate your own driving circuit. (Therefore the motor and driver are sold as a set.)
Noise and vibration	Significant	Very little
Speed	Slow (1000 to 2000 rpm maximum)	Faster (3000 to 5000 rpm maximum)
Out-of-step condition	Possible (will not run if too heavy a load is applied)	Not possible (will rotate even if a heavier load is applied)
Control method	Open loop (no encoder)	Closed loop (uses an encoder)
Price of motor and driver	Cheap (100 to 200 US\$ per set) (If you produce your own driver, the cost will be only a few 10 US\$)	Expensive (more than 400 US\$)
Resolution (single-step angle)	2-phase PM model: 7.5° (48 ppr) 2-phase HB model: 1.8° (200 ppr) or 0.9° (400 ppr) 5-phase HB model: 0.72° (500 ppr) or 0.36° (1,000 ppr)	Depends on the encoder's resolution. Generally speaking, 0.36° (1,000 ppr) to 0.036° (10,000 ppr)

* ppr = Pulses per revolution

III. What is a deflection counter?

The servomotor rotation lags behind the command pulses from the PCL. This means that when the PCL completes outputting a number of pulses equivalent to the preset amount, the encoder will take some time to return all of the pulses. That is why the servo driver includes a "deflection counter."

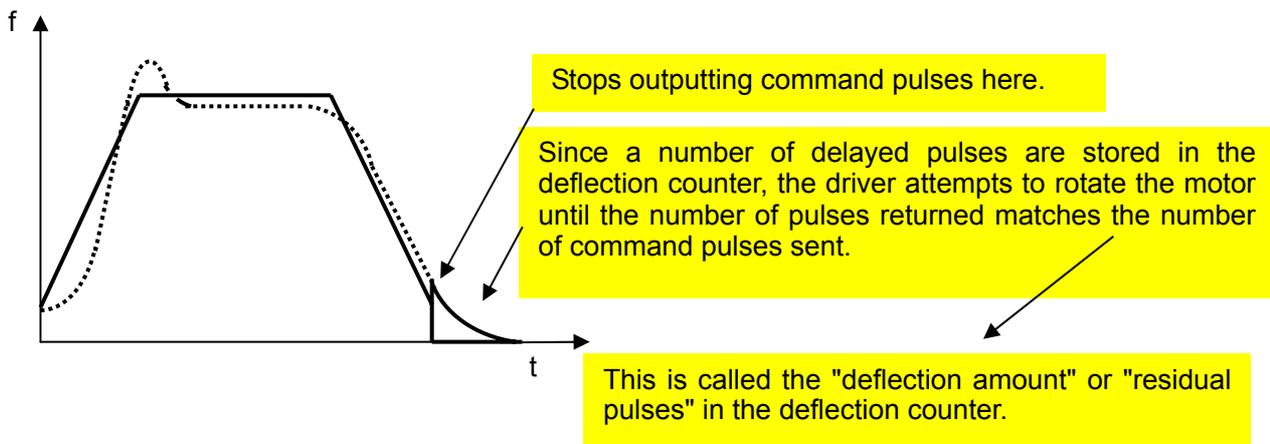
=> This counter compares the number of command pulses from the PCL and the number of pulses returned from the encoder.

=> If the number of pulses returned from the encoder is smaller than the number of command pulses output, the driver will try to rotate the motor some more.

If the number of pulses returned from the encoder is larger than the number of command pulses output, the driver will attempt to run the motor backward.

When the number of command pulses output from the PCL and the number of pulses returned from the encoder match, the motor stops.

(In other words, the driver attempts to rotate the motor until the deflection counter is zero.)

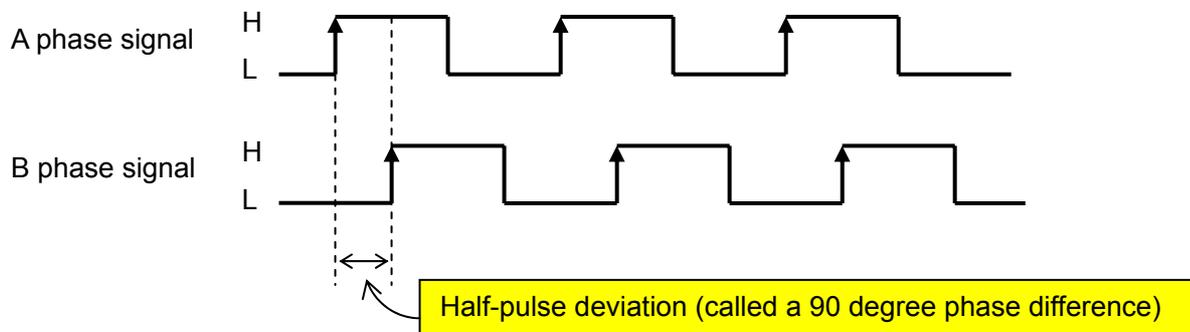


IV. Output signals from an encoder

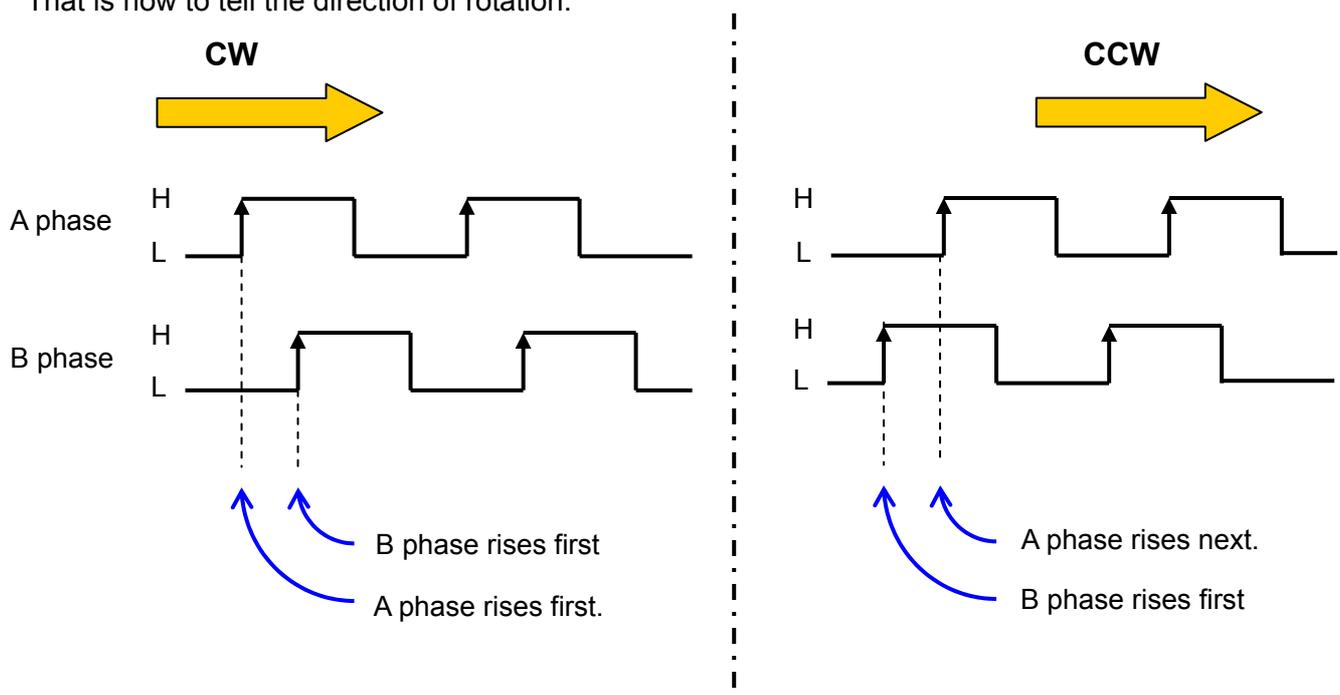
An encoder is a kind of pulse generator. It outputs three types of pulse signals: A phase, B phase, and Z phase (Index signal)

1. A phase and B phase signals

In order to make the pulse per rotation resolution finer and to set the direction of rotation, two pulse trains with the same cycle length are phase shifted.



This half-pulse deviation is the key. For example, if the A phase pulse rises first, this means the motor rotation is clockwise (CW). If the B phase rises first, this means the rotation is counter-clockwise (CCW). That is how to tell the direction of rotation.



2. Z phase (Index signal)

In our example, we will assume that the encoder has 1000 pulse per revolution (1000 ppr) and that it outputs 1000 A and B phase pulses (1000 rising edges per revolution). But, it outputs a Z phase pulse only once per one revolution.

If you want to execute a zero return precisely, a stop using only the ORG sensor may cause a deviation of plus or minus a few pulses each time. Therefore, after the ORG turns ON, count a specified number of Z phase signals and make this the official zero position.

Supply these A phase, B phase, and Z phase signals from the encoder to the PCL.

(The corresponding terminal names on the PCL50xx series, PCL61xx/PCL60xx series are: EA, EB, and EZ)

V. Multiplication

Since stepper motors have full step and half step operation modes, the number of encoder divisions can be enhanced.

For example, an encoder which puts out a nominal 1000 pulses per revolution (1000 ppr) can be multiplied as follow:

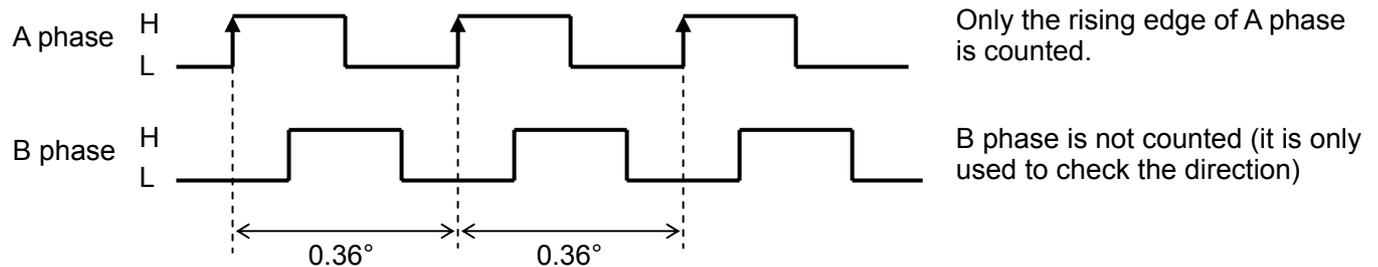
- 1x => 1000ppr = 0.36°/ pulse

- 2x => 2000ppr = 0.18°/ pulse

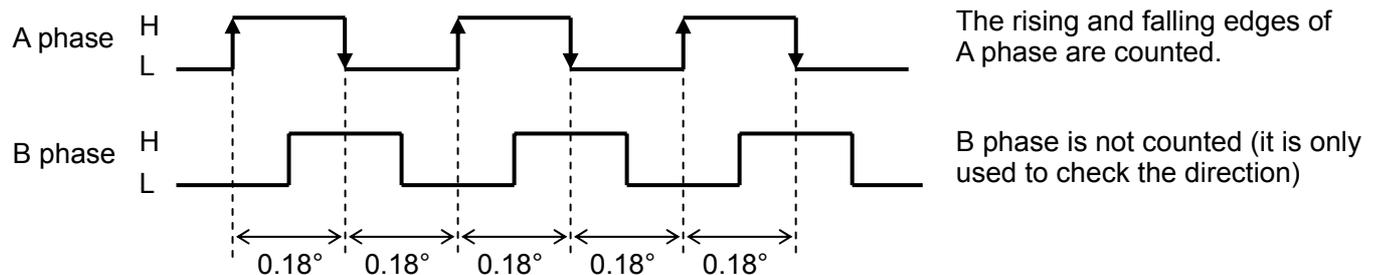
- 4x => 4000ppr = 0.09°/ pulse

(The multiplication rate is specified in the servo driver.)

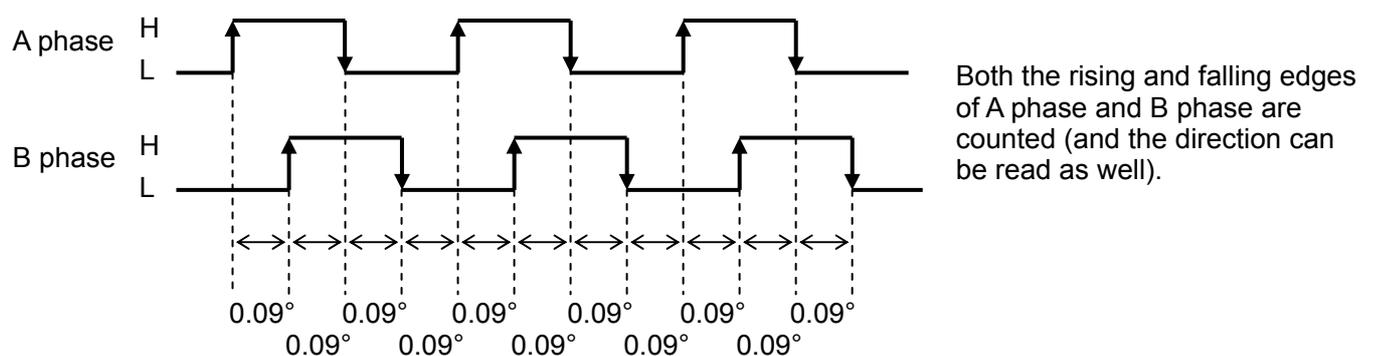
(1) 1x



(2) 2x



(3) 4x

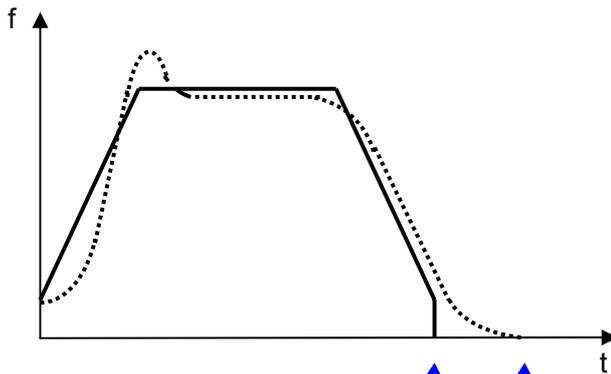


You can specify the multiplication rate to apply (1x, 2x, and 4x) in the PCL. Shown below are the bits used by 3 typical PCL models.

* Multiplication settings of the encoder A/B phase signals

- PCL3013/5014: Set bits 10 and 11 in R16 (environment setting register 1)
- PCL61xx series: Set bits 16 and 17 in RENV2 (environment setting register2) (Bit names: EIM0 and EIM1)
- PCL60xx series: Set bits 20 and 21 in RENV2 (environment setting register2) (Bit names: EIM0 and EIM1)

VI. INP (in-position) signal



Normally the INT is output here.
 However, we want to output the INT signal here.

When controlling a stepper motor the PCL can be made to output an INT signal (interrupt) after all the pulses have been output. However, the servomotor rotation lags behind the command pulses. So, after the PCL has output all of its command pulses, the servomotor is still rotating. When the motor reaches a position equivalent to the number of command pulses, the motor stops rotating (the deflection is 0). In order to indicate that the motor has actually reached the desired position, the servo driver outputs an in-position signal. The PCL is equipped with an input terminal to receive this in-position signal. When this signal is enabled, the completion of the operation can be delayed until the PCL has received this signal.

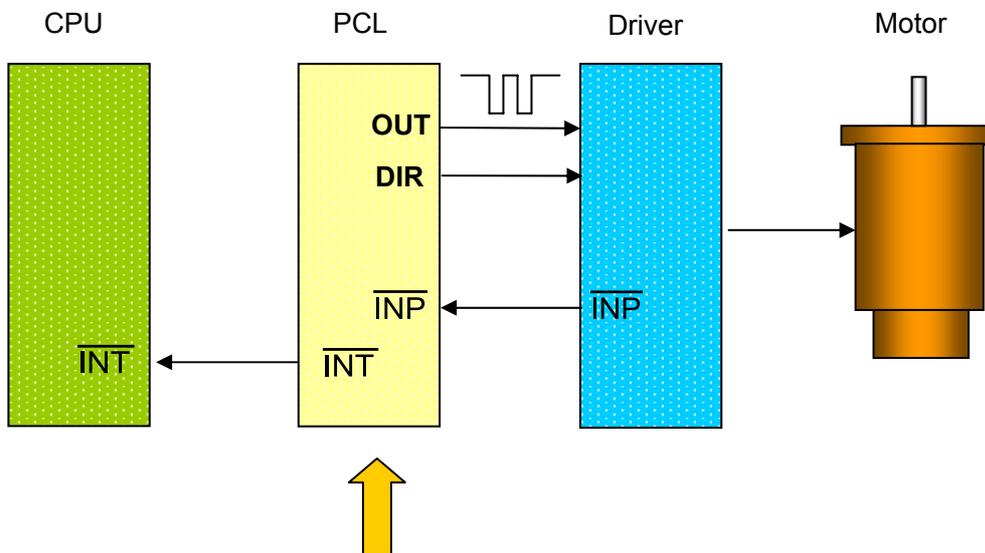
When controlling a stepper motor, you must disable this function since the stepper motor and driver do not produce an in-position signal.

The previous example described how the INT signal is output. However, the PCL can also delay the output of the following signals until the motor has stopped.

- (1) The moment when the BSY signal goes HIGH (this signal is LOW during operation).
- (2) The moment when the operating status changes to "stopped".

However, when the EL or ALM signal goes ON and the motor stops, or a zero return is complete, the INT signal cannot be delayed.

The PCL can be set to identify LOW or HIGH as the in-position signal coming in from a driver.



The PCL completes operation when it receives the INP signal from the servo driver and it outputs an INT signal to the CPU. (When INT signal output is enabled.)

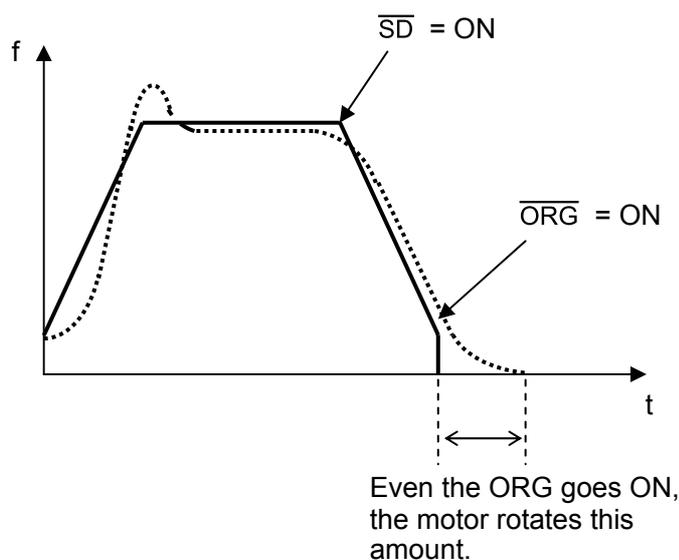
Described below are individual settings related to the INP signal for three typical PCL series.

- * Setting to the delay operation complete status until the INP signal goes ON.
 - PCL3013/5014: Set bit 3 in the control mode buffer to 1
 - PCL61xx series: Set bit 9 in PRMD (RMD) (operation mode register) (Bit name MINP) to 1
 - PCL60xx series: Set bit 9 in PRMD (RMD) operation mode register) (Bit name MINP) to 1

- * Setting to enable output of an INT signal when the INP signal goes ON.
 - PCL3013/5014
 - (1) When the preset operation is complete: Set bit 1 in R8 (environment setting register 3) to 1.
 - (2) During deceleration stop: Set bit 0 in R8 (environment setting register 3) to 1.
 - PCL61xx series: Set bit 0 in RIRQ (event interrupt cause setting register) (bit name IREN) to 1
 - PCL60xx series: Set bit 0 in RIRQ (event interrupt cause setting register) (bit name IREN) to 1
- * The PCL61xx/60xx series output an INT signal when the motor stops normally, regardless of the INP signal setting.

- * Setting the INP signal input logic
 - PCL3013/5014: Set bit 3 in R6 (environment setting register 1).
 - PCL61xx series: Set bit 22 in RENV1 (environment setting 1 register) (Bit name INPL)
 - PCL60xx series: Set bit 22 in RENV1 (environment setting 1 register) (Bit name INPL)

VII. ERC (deflection counter clear) signal



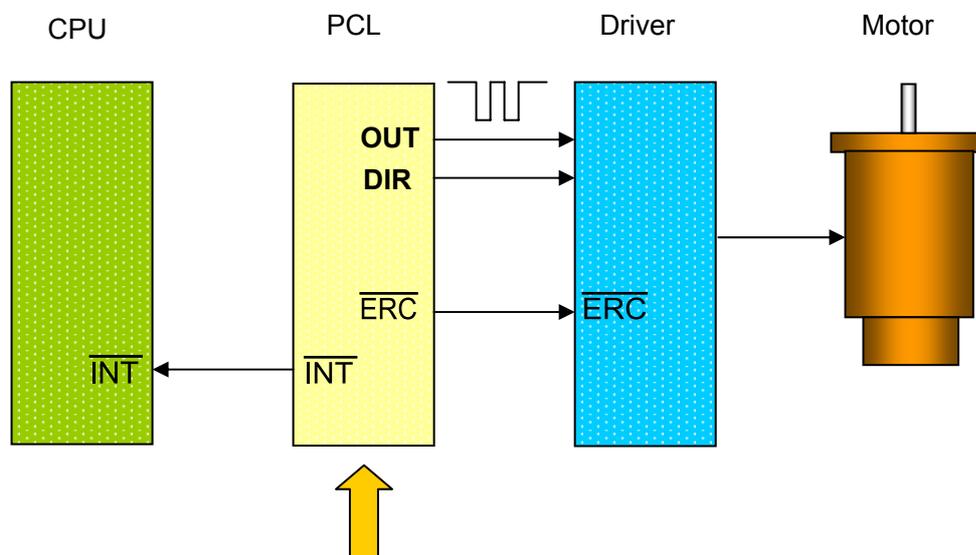
Although the \overline{INP} signal indicating completion of the operation can be delayed by the PCL until the motor stops turning, the \overline{ERC} signal will forcibly stop the motor rotation when the command pulses stop being output, even if there are some deflection pulses left over.

In positioning operations (preset operation), it is essential to rotate the motor for the full number of command pulses, whether the motor rotation is delayed or not. However, there are cases where you may want to stop the motor immediately, when the command pulses stopped being output.

For example, if an immediate stop or emergency stop command is written, or if the \overline{EL} or \overline{ORG} signal goes ON, it is especially important to stop the motor instantly. It is always an emergency when \overline{EL} goes ON. In this case, the motor should be stopped immediately, even if there are a number of deflection pulses left over.

When the \overline{ORG} signal goes ON, this means the device is at the zero position. However, if there are number of deflection pulses left, the motor will keep rotating and therefore precision zero positioning cannot be done.

To stop the motor immediately, even if there are deflection pulses left, you should set the deflection counter to zero. In other words, the \overline{ERC} signal sets the deflection counter to zero.



The PCL outputs the \overline{ERC} signal to the servo driver when it is through outputting command pulses. The PCL outputs the \overline{INT} signal to the CPU to indicate that the motor has stopped.

Described below are individual settings related to the \overline{ERC} signal for three typical PCL series.

* Condition settings for outputting the \overline{ERC} signal.

1. When the emergency stop command is written

- PCL3013/5014: Output unconditionally (emergency stop command: 63h)
- PCL61xx series: Output unconditionally (emergency stop command 05h)

2. Output the $\overline{\text{ERC}}$ signal using a command
- PCL3013/5014: Unavailable
 - PCL61xx series: Write control command "24h."
3. Settings to output the $\overline{\text{ERC}}$ signal automatically in the following conditions.
- PCL3013/5014
Automatic output when the PCL stops operation because of the $\overline{\text{EL}}$ or $\overline{\text{ALM}}$ signal, or a zero return stop:
Set bit 9 in R6 (environment setting register 1) to 1.
 - PCL61xx series
 - (1) When stopped by the $\overline{\text{EL}}/\overline{\text{ALM}}$ signal, or an emergency stop signal:
Set bit 10 (EROE) in RENV1 (environment setting 1 register) to 1.
 - (2) When the zero return is complete:
Set bit 11 (EROR) in RENV1 (environment setting 1 register) to 1.
 - PCL60xx series
 - (1) When stopped by the $\overline{\text{EL}}/\overline{\text{ALM}}$ signal, or an emergency stop signal:
Set bit 10 (EROE) in RENV1 (environment setting 1 register) to 1.
 - (2) When the zero return is complete:
Set bit 11 (EROR) in RENV1 (environment setting 1 register) to 1.
- * Setting the output logic of the $\overline{\text{ERC}}$ signal
- PCL3013/5014: Fixed to negative logic (active LOW)
 - PCL61xx series: Set bit 15 in RENV1 (environment 1 setting register) (Bit name ERCL)
 - PCL60xx series: Set bit 15 in RENV1 (environment 1 setting register) (Bit name ERCL)
- * Set the $\overline{\text{ERC}}$ signal output length
- PCL3013/5014: Fixed to 200 ms
 - PCL61xx series: Set bits 12 to 14 in RENV1 (environment setting 1 register)
(Can be set to one of 8 lengths using bits EPW0 to EPW2)
 - PCL61xx series: Set bits 12 to 14 in RENV1 (environment setting 1 register)
(Can be set to one of 8 lengths using bits EPW0 to EPW2.)

Although an $\overline{\text{INT}}$ signal is not output by clearing the deflection counter, when the counter is cleared, the motor stops. By determining the cause of the stop, an $\overline{\text{INT}}$ signal can be output to indicate that the counter has been cleared.

- * The $\overline{\text{ERC}}$ signal is referred to as the " $\overline{\text{CLR}}$ " signal in the PCL-240 family. The $\overline{\text{CLR}}$ terminal name is also used in PCL50xx/60xx series. However, this terminal is not used to clear the deflection counter. It is a signal used to clear the up/down counter.

VIII. ALM (alarm) signal

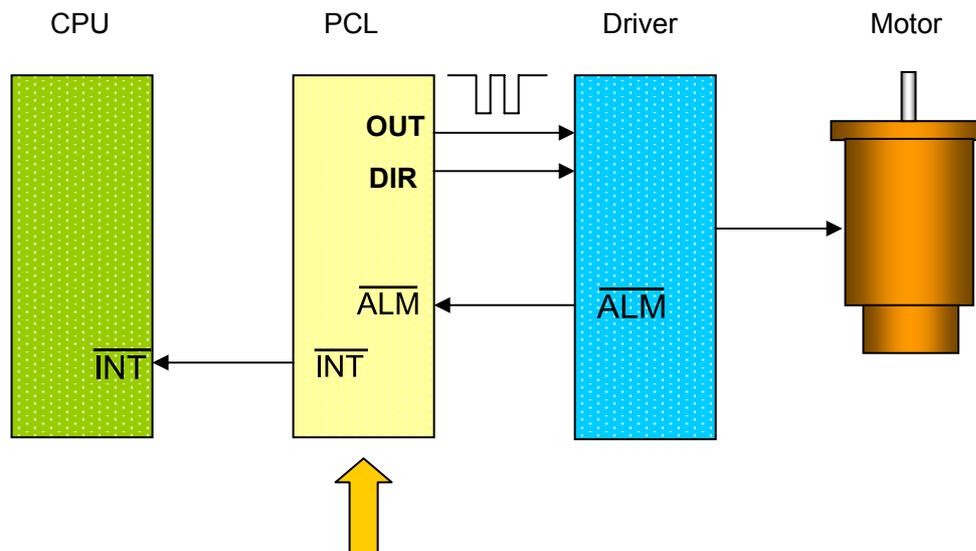
Like the $\overline{\text{INP}}$ signal, the $\overline{\text{ALM}}$ is a signal output from a servo driver.

This signal is output when any of the abnormalities below occur in the servo driver or motor.

- (1) The deflection amount becomes abnormally large.
- (2) Excess current is flowing through the motor (instantaneous measurement, or at certain intervals)
- (3) There is a temperature error
- (4) There is a power supply voltage error

The alarm signal associated with each error varies with the servo driver.

Generally, the $\overline{\text{ALM}}$ signal from a servo driver is kept ON until it is cleared. The PCL cannot start operation, even if a start command is written, unless the cause of the $\overline{\text{ALM}}$ is cleared and the servo driver turns OFF the $\overline{\text{ALM}}$ signal. (The PCL is not equipped with a release function.) If a start command is written while this signal is ON, the PCL will not operate and it will only keep outputting an $\overline{\text{INT}}$ signal. Also, the PCL can select whether the $\overline{\text{ALM}}$ signal from the driver is active LOW (negative logic) or active HIGH (positive logic).



No matter whether operating or stopped, the PCL will output the $\overline{\text{INT}}$ signal to the CPU when it receives an $\overline{\text{ALM}}$ signal from the servo driver. Unless the $\overline{\text{ALM}}$ signal is turned OFF by the driver, the motor cannot be started, even if a start command is written.

The settings related to the $\overline{\text{ALM}}$ signal are described below for three typical PCL series models.

* Setting for the stop method when the $\overline{\text{ALM}}$ signal is ON

- PCL3013/5014: Immediate stop
- PCL61xx series: Set bit 8 in RENV1 (environment setting 1 register)
Select whether to use an immediate stop or a deceleration stop with the ALML bit
- PCL60xx series: Set bit 8 in RENV1 (environment setting 1 register)
Select whether to use an immediate stop or a deceleration stop with the ALML bit

* Setting the input logic for the $\overline{\text{ALM}}$ signal

- PCL3013/5014: Set bit 6 in R6 (environment setting register 1)
- PCL61xx series: Set bit 9 in RENV1 (environment setting 1 register) (Bit name: ALML)
- PCL60xx series: Set bit 9 in RENV1 (environment setting 1 register) (Bit name: ALML)

* Read the cause of the $\overline{\text{ALM}}$ signal

- PCL3013/5014: Use interrupt code "0Bh" to read the interrupt
- PCL61xx series: Set bit 2 in the REST (error interrupt cause) register (bit name: ESAL) to 1
- PCL60xx series: Set bit 7 in the REST (error interrupt cause) register (bit name: ESAL) to 1

That completes the basic description of servomotor control.
NPM's PCL series is suitable for use with pulse train input servo drivers.

To learn more about the PCL series, also see the pamphlet "Basic instructions for the PCL series."



NPM Nippon Pulse Motor Co., Ltd.

Asia/Europe: Nippon Pulse Motor Co., Ltd.

2-16-13 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan

TEL: 81-3-3813-8841

FAX: 81-3-3813-7049

Web: <http://www.pulsemotor.com>

E-mail: int-l@npm.co.jp

North/South America: Nippon Pulse America, Inc.

1073 East Main Street, Radford, VA 24141, U.S.A.

TEL: 1-540-633-1677

FAX: 1-540-633-7049

Web: <http://www.nipponplse.com>

E-mail: info@nipponpulse.com