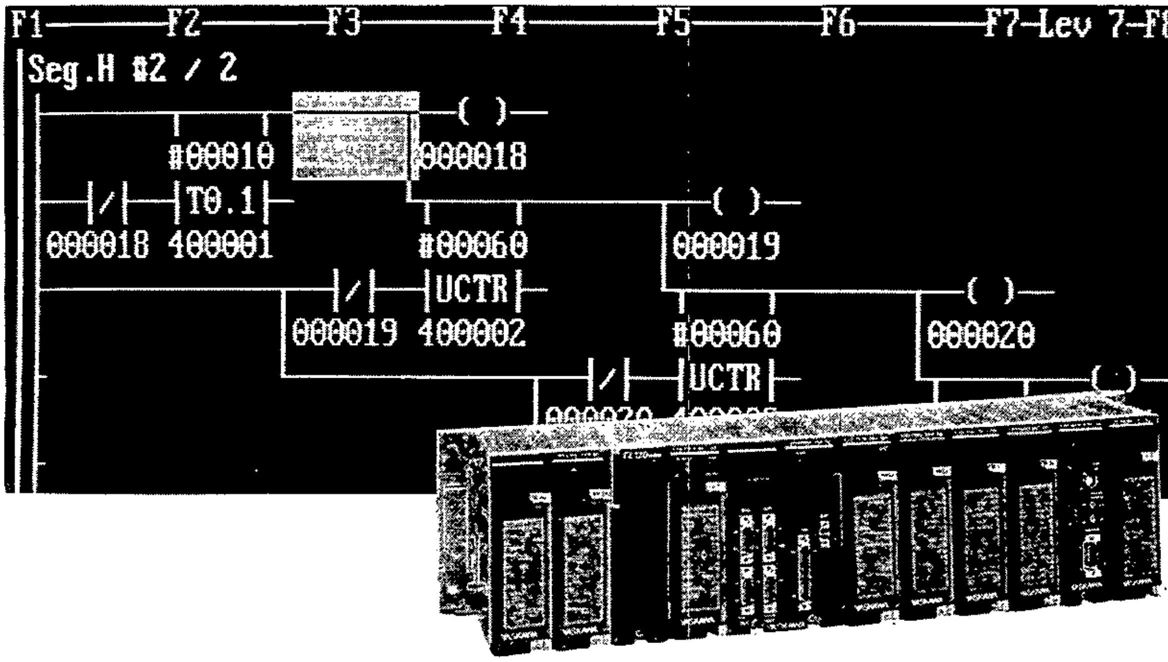
YASKAWA

MEMOCON GL120, GL130
SOFTWARE USER'S MANUAL
VOL.4



YASKAWA

MANUAL NO. SIEZ-C825-20.14

Manual Contents

This manual describes the ladder logic programming instructions used to program the MEMOCON GL120 and GL130 Programmable Controllers (PLCs). Please read this manual carefully and be sure you understand the information provided before attempting to program a MEMOCON PLC.

Visual Aids

The following aids are used to indicate certain types of information for easier reference.

☞	Indicates references for additional information.
IMPORTANT	Indicates important information that should be memorized.
◀EXAMPLE▶	Indicates application examples.
INFO	Indicates supplemental information.
▶ SUMMARY	Indicates a summary of the important points of explanations.
Note	Indicates inputs, operations, and other information required for correct operation but that will not cause damage to the device.
TERMS	Indicates definitions of terms used in the manual.

NOTICE

The following conventions are used to indicate precautions in this manual. Failure to heed precautions provided in this manual can result in injury to people or damage to the products.

⚠ **WARNING**	Indicates precautions that, if not heeded, could possibly result in loss of life or serious injury.
⚠ **Caution**	Indicates precautions that, if not heeded, could result in relatively serious or minor injury, damage to the product, or faulty operation.

CONTENTS

APPENDIX

Introduction and Precautions

This chapter introduces general information, including basic information and precautions for the use of this manual and the software. **You must read this chapter before attempting to read the rest of the manual or using the product.**

I.1 Overview of Manual

- This manual describes the program control instructions used to create ladder logic programs for MEMOCON GL120 and GL130 Programmable Controllers. Basic, expansion math, communications, and motion control instructions are described in other manuals.

- Read this manual carefully in order to use the instructions properly. Also, keep this manual in a safe place so that it can be used whenever necessary.

- Refer to the following related manuals.

Manual Name	Manual No.	Content
MEMOCON GL120, GL130 Hardware User's Manual	SIEZ-C825-20.1	Describes system configuration devices, functions, specifications, and application methods for the GL120 and GL130.
MEMOCON GL120 CPU10 Module User's Manual	SIEZ-C825-20.1-1	Describes the functions, specifications, and usage of the CPU10 Module.
MEMOCON GL120 CPU21 Module User's Manual	SIEZ-C825-20.1-2	Describes the functions, specifications, usage, and extended memory access instructions of the CPU21 Module.
MEMOCON GL130 CPU35 Module User's Manual	SIEZ-C825-20.1-3	Describes the functions, specifications, and usage of the CPU35 Module.
MEMOCON GL120, GL130 Software User's Manual, Vol.1	SIEZ-C825-20.11	Describes the following for the GL120 and GL130: 1) Operating principles, 2) I/O allocation, 3) Overview of instructions and 4) Instruction processing times.
MEMOCON GL120, GL130 Software User's Manual, Vol.2	SIEZ-C825-20.12	Describes the programming instructions used to create ladder logic programs for the GL120 and GL130.
MEMOCON GL120, GL130 Software User's Manual, Vol.3	SIEZ-C825-20.13	Describes the expansion math instructions (e.g., floating point math instructions) for the GL120 and GL130.
MEMOCON GL120, GL130 PC Link Module User's Manual	SIEZ-C825-70.4	Describes the PC Link Module and the FBUS communications instruction for the GL120 and GL130.
MEMOCON GL120, GL130 MEMOBUS PLUS User's Manual	SIEZ-C825-70.5	Describes the MEMOBUS PLUS and the MSTR communications instruction.
MEMOCON GL120, GL130 COM Instructions User's Manual	SIEZ-C825-70.14	Describes the COM communications instruction.

Manual Name	Manual No.	Content
MEMOCON GL120, GL130 Motion Module MC20 Software User's Manual	SIEZ-C825-20.52	Describes the motion control instructions for the GL120 and GL130.
MEMOCON GL120, GL130 Motion Module MC20 Hardware User's Manual	SIEZ-C825-20.51	Describes the design and maintenance of Motion Modules.
MEMOCON GL120, GL130 P120 Programming Panel (MEMOSOFT) User's Manual	SIEZ-C825-60.7	Describes the functions, specifications and usage of the P120Programming Panel with MEMOSOFT.
MEMOCON, GL120, GL130 MEMOSOFT for DOS User's Manual	SIEZ-C825-60.10	Describes the functions and operational methods for MEMOSOFT for DOS.
MEMOCON GL120, GL130 120 Series I/O Modules User's Manual	SIEZ-C825-20.22	Describes the functions, specifications and usage of the 120-Series I/O Modules.

- Thoroughly check the specifications and conditions or restrictions of the product before using it.

I.2 Safety Precautions

- MEMOCON was not designed or manufactured for use in devices or systems that concern human lives. Users who intend to use the product described in this manual for special purposes such as devices or systems relating to transportation, medical, space aviation, atomic power control, or underwater use must contact Yaskawa Electric Corporation beforehand.

- This product has been manufactured under strict quality control guidelines. However, if this product is to be installed in any location in which a failure of MEMOCON involves a life and death situation or in a facility where failure may cause a serious accident, safety devices MUST be installed to minimize the likelihood of any accident.

- Any illustrations, photographs, or examples used in this manual are provided as examples only and may not apply to all product to which this manual is applicable.

- The products and specifications described in this manual or the content and presentation of the manual may be changed without notice to improve the product and/or the manual. A new version of the manual will be re-released under a revised document number when any changes are made.

- Contact your Yaskawa representative or a Yaskawa office listed on the back of this manual to order a new manual whenever this manual is damaged or lost. Please provide the document number listed on the front cover of this manual when ordering.

- Contact your Yaskawa representative or a Yaskawa office listed on the back of this manual to order new nameplates whenever a nameplate becomes worn or damaged.

- Yaskawa cannot make any quality guarantee for products which have been modified. Yaskawa assumes no responsibility for any injury or damage caused by a modified product.

I.3 Using this Manual

This manual is written for the following personnel:

- Personnel responsible for designing GL120 or GL130 ladder programs.
- Personnel responsible for testing GL120 or GL130 ladder programs.
- Personnel responsible for debugging GL120 or GL130 ladder programs during trial operation.
- Personnel responsible for maintaining GL120 or GL130 ladder programs.
- **Basic Terms**

  In this manual, the following terms have the meanings described below.

  - **PLC = Programmable (Logic) Controller**
  - **PP = Programming Panel**
  - **GL120, GL130 = MEMOCON GL120 and/or MEMOCON GL130 Programmable Controller**

- **Technical Terms**

  The bold technical terms in this manual are briefly explained in the **Glossary** provided at the bottom of the page. An example is shown below.

Glossary

The following types of terms are described.

- Specific sequence control terms required for explanation of functions.
- Terms that are specific to programmable controllers and electronic devices.

I.4 Reference Numbers

The types and ranges of reference numbers that can be used with each instruction are provided in this manual under the heading *Structural Elements*, as shown in the following example. These reference numbers are specified as follows:

1) The ranges listed in the tables are for initial values.

2) Two systems are used for reference numbers for holding registers and link registers: Reference numbers beginning with numbers are called numeric reference numbers and those beginning with letters are called lettered reference numbers. In the *Structural Elements* tables, the lettered reference numbers are given in parentheses.

Structural Elements of PID2

Element	Meaning	Possible Settings	
Top (S)	Source reference number	Holding register: Link register:	400001 to 409979 (W00001 to W09979) R10001 to R11004 or R20001 to R21004
Middle (P)	Pointer reference number	Holding register: Link register:	400001 to 409991 (W00001 to W09991) R10001 to R11016 or R20001 to R21016
Bottom (Z)	Destination table size	Constant:	#00001 to #00255

1

Preparing to Configure a PID System

This chapter explains the Modules and Cables necessary to configure a PID system.

1.1 MEMOSOFT Preparations

1) Prepare the following programming software to program the PID circuit using a PC.

a) DOS

MEMOSOFT for DOS (model: FMSGL-DV3) is necessary.

b) P120-series Programming Panel

MEMOSOFT for P120 (model: FMSGL-PP3) is necessary.

1.2 Preparing Connection Cables

1) Select the necessary cables from *Tables 1.1, 1.2* and *1.3* according to the PLC and computer.

a) Connecting via MEMOBUS

Table 1.1 MEMOBUS–Computer Cables

PLC	DOS Computer Cable	
	Model	Cable Length
MEMOCON GL120, GL130	JZMSZ-120W0202-03	2.5 m
	JZMSZ-120W0202-15	15 m

Note JZMSZ-120W0200-□□ D-sub connector (25 pin) cable
JZMSZ-120W0201-□□ Half-pitch (14 pin) cable

Table 1.2 MEMOBUS–P120 Programming Panel Cables

PLC	P120-Series Panel Cable	
	Model	Cable Length
MEMOCON GL120, GL130	JZMSZ-120W0203-03	2.5 m
	JZMSZ-120W0203-15	15 m

b) Connecting via MEMOBUS PLUS

Table 1.3 MEMOBUS PLUS–P120/DOS

PLC	DOS Computer/ P120-series Panel Cable	
	Model	Cable Length
MEMOCON GL120, GL130	JZMSZ-120W0800-03	2.5 m
	JZMSZ-120W0800-15	15 m

Note (a) It is necessary to install a MEMOBUS PLUS Communications Board (model: SA85).

(b) The cables shown in *Table 1.3* are used for 1:1 communications between the GL120/GL130 and SA85.

1.3 CPU Preparations

1) Configuring circuits for one loop of PID control requires an Analog Input Module to input the process variable and an Analog Output Module to output the manipulated variable, as shown in *Figure 1.1* below.

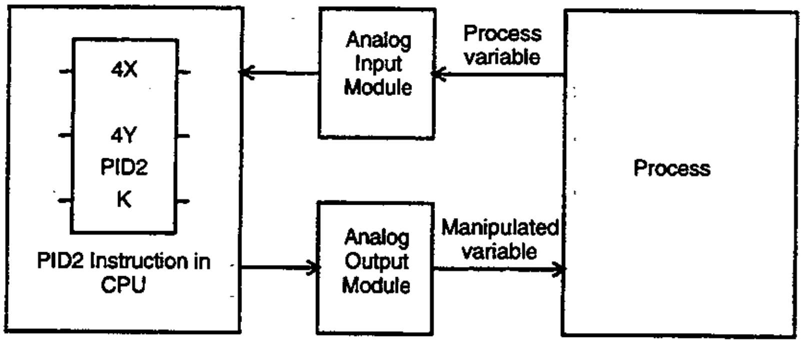


Figure 1.1 PID Loop Configuration

2) **Configuration**

The GL120 and GL130 systems are configured using Modules. Configuring circuits for one loop of PID control requires at least a Power Supply Module, a CPU Module, an Analog Input Module, and Analog Output Module, and a Mounting Base (to which the Modules are mounted), as shown in *Figure 1.2.*

A list of the model numbers and specifications of each Module is provided in *Table 1.4* below.

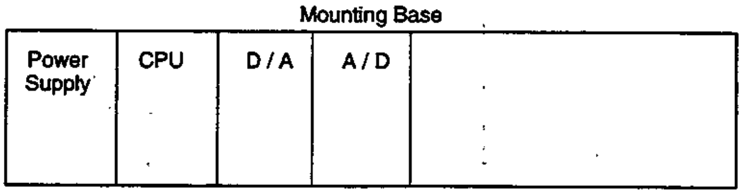


Figure 1.2 Basic GL120 and GL130 Configuration

Table 1.4 GL120 and GL130 Modules

Module	Module Description	Model	Specifications
Power Supply Module	PS10	JRMSP-120CPS11300	Input: 100/200 VAC Output: 7 A, 5 VDC
	PS05	JRMSP-120CPS11100	Input: 100/200 VAC Output: 3 A, 5 VDC
	PS11	JRMSP-120CPS21300	Input: 24 VDC Output: 7 A, 5 VDC
	PS06	JRMSP-120CPS21100	Input: 24 VDC Output: 3 A, 5 VDC
CPU Module	CPU10	DDSCR-120CPU14200	Program: 8 kW, DIO:1,024 points, RIO: 512 registers
	CPU20	DDSCR-120CPU34100	Program: 16 kW, DIO:1,024 points, RIO: 512 registers
	CPU21	DDSCR-120CPU34110	Program: 16 kW, DIO:1,024 points, RIO: 512 registers
	CPU30	DDSCR-130CPU54100	Program: 32 kW, DIO: 4,096 points, RIO: 512 registers
	CPU35	DDSCR-130CPU54110	Program: 40 kW, DIO: 4,096 points, RIO: 512 registers
Analog Output Module (converts digital to analog)	D/A-VOL-2CH	JAMSC-120AVO01000	–10 V to +10 V, 12 bits, 2 channels
	D/A 0-10V 2CH	JAMSC-120AVO01100	0 V to +10 V, 12 bits, 2 channels
	D/A 0-5V 2CH	JAMSC-120AVO01200	0 V to +5 V, 12 bits, 2 channels
	D/A-CUR-2CH	JAMSC-120ACO01000	4 mA to 20 mA, 12 bits, 2 channels
Analog Input Module (converts analog to digital)	A/D-VOL-4CH	JAMSC-120AVI02000	–10 V to +10 V, 12 bits, 4 channels
	A/D 0-10V 4CH	JAMSC-120AVI02100	0 V to +10 V, 12 bits, 4 channels
	A/D-CUR-4CH	JAMSC-120ACI02000	4 mA to 20 mA, 12 bits, 4 channels
Mounting Base	MB06	JRMSI-120XBP00600	6-slot Mounting Base
	MB08	JRMSI-120XBP00800	8-slot Mounting Base
	MB10	JRMSI-120XBP01000	10-slot Mounting Base
	MB12	JRMSI-120XBP01200	12-slot Mounting Base
	MB16	JRMSI-120XBP01600	16-slot Mounting Base

3) Analog I/O Module Allocation

- Refer to section 6.3 in the *MEMOCON GL120, GL130 MEMOSOFT for DOS User's Manual* (SIEZ-C825-60.10)

- Refer to section 6.3 in the *MEMOCON GL120, GL130 Programming Panel P120 (MEMOSOFT* ***English*** *version) User's Manual* (SIEZ-C825-60.7).

- Refer to chapter 3 in the *MEMOCON GL120, GL130 120-Series I/O Modules User's Manual* (SIEZ-C825-20.22).

4) Analog I/O Module Wiring

- Refer to chapter 4 in the *MEMOCON GL120, GL130 Hardware User's Manual* (SIEZ-C825-20.1).

- Refer to the *MEMOCON GL120, GL130 120-Series I/O Modules User's Manual* (SIEZ-C825-20.22).

Specifications

2

This chapter explains the basics of PID control and the details of the registers used in the ladder logic program.

2.1 PID Overview

This section explains the basic operation of PID control.

2.1.1 PID System Overview

1) PID control involves adding to the process the manipulated variable calculated using P (proportional), I (integral), and D (derivative) control for the difference (process error) between the actual process value and the set point, thereby approaching the set point.

Note PID is an abbreviation for Proportional Integral Derivative.

a) Closed–loop Control

Closed-loop control is a system which measures, analyses, and adjusts the process error from the set point in order to achieve and maintain an ideal process state. Closed-loop control can be programming in the ladder logic program using a PID control block called PID2.

b) Set Point (SP) and Process Variable (PV)

In a PID2 block, the set point is abbreviated as SP, and the process variable is abbreviated as PV. The difference between SP and PV is referred to as the process error (E). Process error (E) is defined as follows:

$$E = SP - PV$$

E is used in a control calculations that generate a manipulated variable (MV) to control the process so that PV = SP, i.e., E = 0.

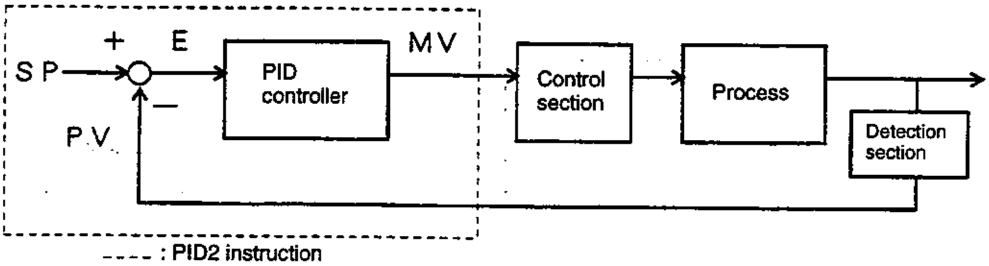


Figure 2.1 Closed-loop Control System

Note This manual discriminates between different versions of the set point (SP) and process variable (PV) as follows:
Internal units: SP value, PV value
Engineering units: Sp value, Pv value

2.1.2 Proportional (P) Control

1) In P control, the manipulated variables is determined by applying a proportional constant K_1 to the process error (E) and adding a bias.

$$MV = K_1 \times E + Bias$$

2) Here, the reciprocal of the constant K_1 is used as a percentage instead of K_1. It is expressed in terms of the proportional band PB.

$$K_1 = 100/PB$$

If the proportional constant K_1 is increased (or the proportional band PB is decreased), a large output will be produced even if the process error is small, allowing a quick response and making it possible to reduce the constant process error. If K_1 is increased further, however, it will cause oscillations and eventually will result in a continuous oscillation.

Even if proportional control is applied, a constant process error will remain between PV and SP. While it is possible to eliminate the process error by adjusting the bias manually (manual reset), normally the process status changes as a result of other system variables, so if the bias is kept fixed, the process error will occur again. This status is the limit of proportional control.

INFO

1) The proportional band determines the effective change width for the process error and manipulated variable. The proportional band is expressed as a percentage of the change in the process error needed for the manipulated variable to change by the effective change width under proportional control.

2) During manual operation, the values of integral terms in registers 4Y+3, 4Y+4, and 4Y+5 are always used even when integral action is not selected. This is necessary for bumpless switching between manual and automatic operation. (Refer to *2.6.4 Bumpless Switching Between Manual and Automatic.*) If bumpless switching is not used, these registers must always be kept clear.

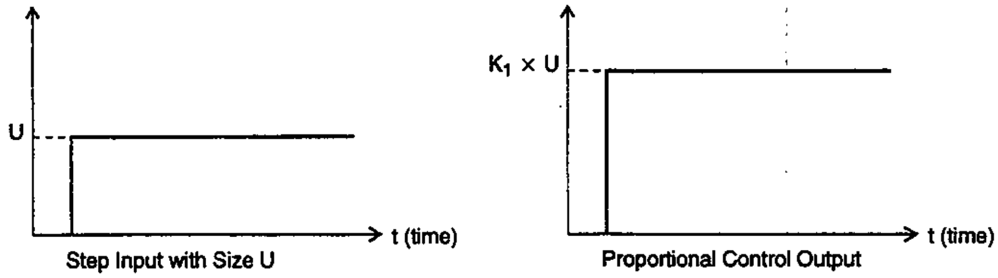


Figure 2.2 Proportional Control Step Response

2.1.3 Proportional Integral (PI) Control

1) By adding an integration function to the control expression, it is possible to eliminate the process error without manually changing the bias.

$$MV = K_1\ (E + K_2 \int Edt\ )$$

2) It is possible for PI control to eliminate the process error by integrating process error as a function of time. K_2 is the integral constant. As long as E is not 0, the integrator continues to increase (or decrease) its value, thereby adjusting the MV. This can be continued until the process error is eliminated. Because the objective of integral action is eliminating the constant process error, it is also referred to as reset action.

3) The strength of integral action is expressed by the integral constant K_2. If K_2 is increased, the output of the integral action will change greatly. If it is too large, it will cause oscillation, and if it is too small, it will take a long time for the process error to be eliminated.

INFO The integral constant K_2 (reset rate) is the reciprocal of the reset time constant. Taking the reset time constant as T_1, we get the relation

$$K_2 = 1/T_1$$

The reset time constant is the time taken from the step input until the proportional action equals the integral action.

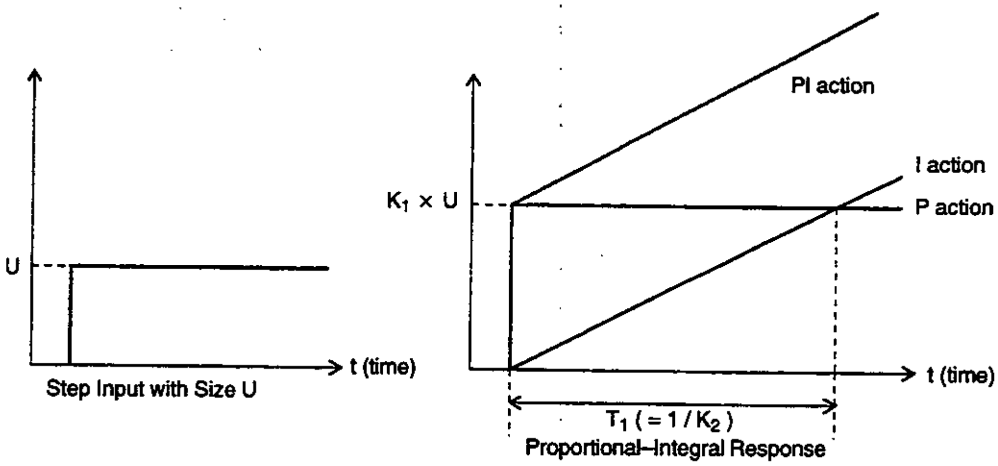


Figure 2.3 Reset Time Constant Definition

2.1.4 Proportional Integral Derivative (PID) Control

1) A derivative action is added to eliminate as quickly as possible process error generated by disturbance and to eliminate instability in delay resulting from the integral action.

$$MV = K_1\ (E + K_2 \cdot \int Edt - K_3 \cdot dPV/dt\ )$$

2) K_3 is a derivative constant. dPV is the change in PV over time dt. Therefore, if PV is fixed, the derivative action will not operate. Derivative action is referred to as rate action because it is in proportion to the rate of change in the signal.

3) If the derivative constant K_3 is increased, the output of the derivative action will also increase. When K_3 is too large, oscillation will occur easily.

INFO The derivative constant K_3 (rate time constant) is the time taken from lamp input until the proportional action equals the derivative action.

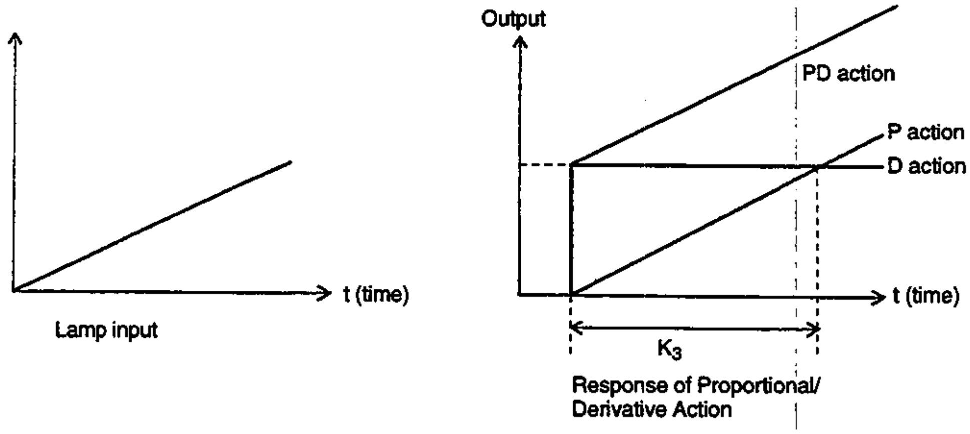


Figure 2.4 Rate Time Constant Definition

2.2 PID Calculations

▌ This section explains PID calculations.

2.2.1 General PID Calculations

1) The general equation for PID calculations (error value differentiation) of continuous analog values is shown at (a) below. The relationships between the elements in the calculations is shown in *Figure 2.5.*

$$MV(t) = \frac{100}{PB}\{E(t) + \frac{1}{Ti}\int E(t)dt + Td\frac{dE(t)}{dt}\} + M_0 \text{ .... (a)}$$

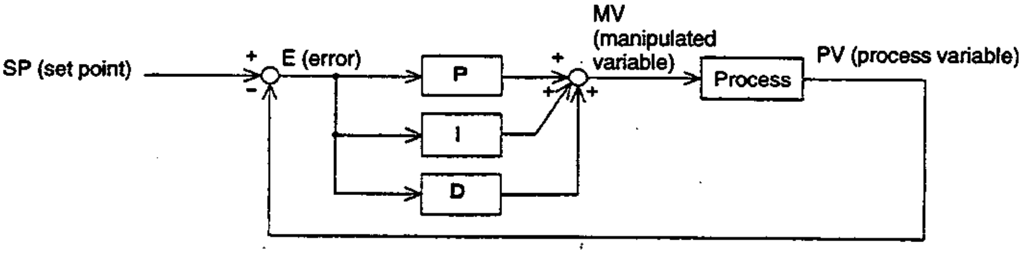


Figure 2.5 Error Value Differentiation

IMPORTANT

2) GL120 and GL130 use process variable differentiation, in which the derivative effect works not on process error but on the process variable. The equation for calculating process variable differentiation is shown at (b) below. The relationships between the elements in the calculations are shown in *Figure 2.6.*

$$MV(t) = \frac{100}{PB}\{E(t) + \frac{1}{Ti}\int E(t)\ dt - Td\frac{dPV(t)}{dt}\} + M_0 \text{ .... (b)}$$

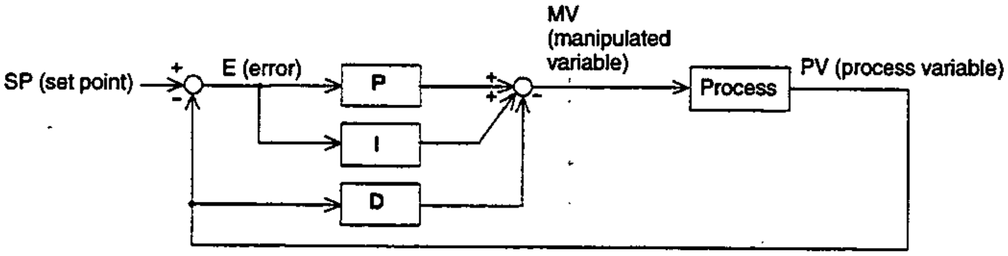


Figure 2.6 Process Variable Differentiation

For (a) and (b) above,

t: Time (s)

MV(t): Manipulated variable

E(t): Process error

PV(t): Process variable

E(t) = SP(t) – PV(t) SP(t): Set point

PB: Proportional band [%]

Ti: Reset time constant [s]

Td: Rate time constant [s]

M_0: Bias

2.2.2 PID Calculations for GL120 and GL130 PLCs

1) In GL120 and GL130 PLCs, the process variable differentiation equation at (b) is digitalized and calculated using the following approximate expression (integral → sum, derivative → differential).

$$MV(t_n) = K_1 [E(t_n) + \frac{K_2}{60} \Sigma E(t_n) \cdot Ts - \boxed{\text{Filter}}\ 60K_3\{PV(t_n) - PV(t_{n-1})\} \times \frac{1}{Ts}\ ] + M_0 \qquad \ldots\ldots \text{(c)}$$

tn: Time for the nth sampling cycle [s]

Ts: Sampling cycle [s] tn = nTs

K_1: Proportional gain $K_1 = 100/PB$ PB: Proportional band [%]

K_2: Reset rate [repeats/min] $K_2 = 60/Ti$

K_3: Rate time constant [min] $K_3 = Td/60$

M_0: Bias

Filter: Filter constant (RGL)

2.3 Derivative Term

▌ This chapter explains the derivative term and its features

2.3.1 The Effects of Process Variable Differentiation

1) Process variable differentiation is suited for controlling set points that are set to a constant value.

2) Set points are often changed manually, and generally their rate of change is much faster than the rate of change of the manipulated variable in response to disturbance. With normal error value differentiation, the control response deteriorates because the derivative action is too responsive to changes in the set point.

3) In process variable differentiation, control action against disturbance is exactly the same as that for error value differentiation. Differentiation does not work against changes in set points, however, so differentiation is never too responsive.

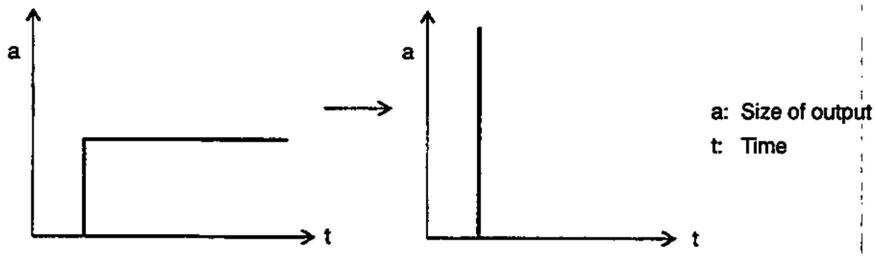
4) In process variable differentiation, changes in set point become changes in the manipulated variable, and they are input to the differential-type derivative action. In this case, however, changes pass through a delay in the control system, so changes are slowed down, and do not have any adverse effects.

2.3.2 Filters

1) For a step input, for example, if derivative calculations are used, a response like that shown in *Figure 2.7 (b)* will result. This is referred to as exact differentiation. Theoretically it is possible to produce a pulse wave output of unlimited size with a time amplitude of 0, but with waveforms with this line shape, the external control unit will not respond no matter how high it is. In addition, the effective range of the control signal is exceeded, and the functions of the derivative action will not only be unable to be utilized fully, but they also become hazardous.

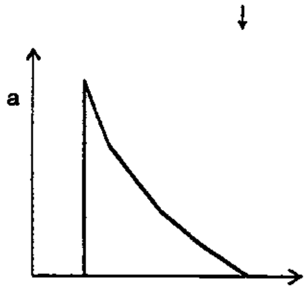
2) To prevent this, a low-pass filter (initial delay filter) that cuts the high frequency components before derivatives are input, and inexact differentiation is used. (Refer to *Figure 2.8.*) Stated simply, it entails somewhat slowing the changes in output generated by differentiation (giving it a time amplitude), then providing an upper limit to output, thereby making derivative action work effectively.

3) Although the rate gain limit (RGL) constant is used as a filter constant, the smaller the value of this RGL constant becomes, the more effect it has as a low-pass filter. The step response changes shown in *Figure 2.7* and *Figure 2.9* are according to the filter constant RGL of the low-pass filter and the size of the rate time constant of the derivative term.



(a) Step Input

(b) Step Response for Exact Differentiation



(c) Step response for Inexact Differentiation

Figure 2.7 Step Response for Derivative Terms

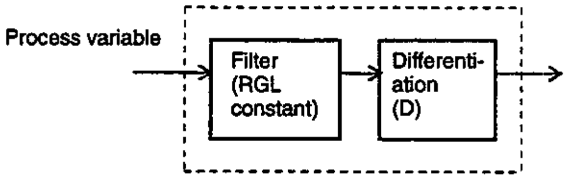
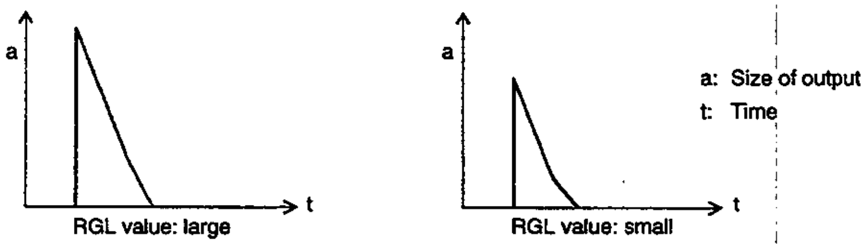
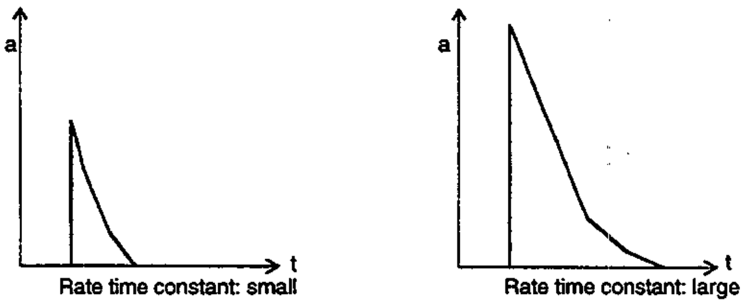


Figure 2.8 Inexact Differentiation



(a) Filter Constant for Inexact Differentiation: RGL



(b) Rate Time Constant for Inexact Differentiation: T_d

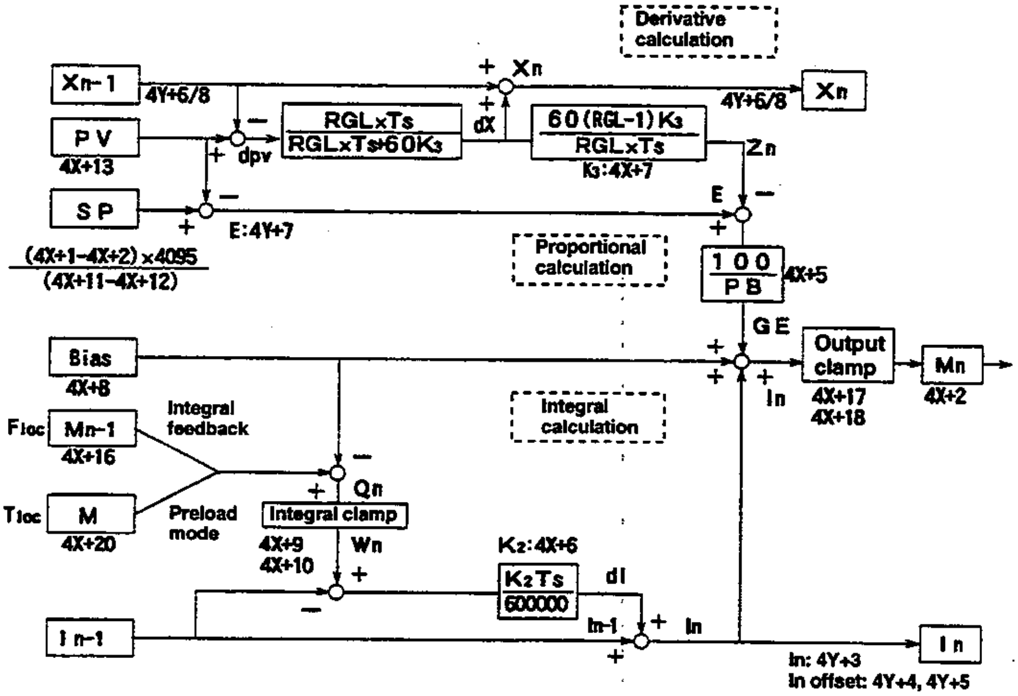
Figure 2.9 Step Response to Changes in Filter Constant (RGL) for Inexact Differentiation and Rate Time Constant

2.4 Block Diagram for PID2 Instruction

This section shows the block diagram of calculation of PID2 instructions.

PID Calculation Block Diagram

The PID2 instruction executes PID calculations using algorithms corresponding to the block diagram shown below. (*Figure 2.10*)



E:	Process error	
SP:	Set point	(0 to 4095)
PV:	Process variable	(0 to 4095)
X:	Filtered PV	
K_1:	Proportional gain constant (100/PB)	
K_2:	Integral gain constant (reset rate)	(0 to 9999) × 0.01/min
K_3:	Derivative gain constant	(0 to 9999) × 0.01min
RGL:	Derivative gain filter constant	(2 to 30)
Ts:	Sampling time	(1 to 255) × 0.01 s
PB:	Proportional band	(5% to 500%)
Bias:	Loop output bias	(0 to 4095)
M:	Loop output	
GE:	PD term output	
Z:	D term output	
Qn:	Bias-eliminated loop output	
F:	Feedback value	(0 to 4095)
I:	I term output	

Note Integral terms are calculated by integrating the difference between actual output (M) and the integral value (I). This has the same effect as subjecting the process error to integral calculation.

Figure 2.10 PID2 Instruction Execution Block Diagram

2.5 PID2 Ladder Logic Instruction

This chapter explains the ladder logic I/O, and the contents of the top, middle and bottom registers of the PID2 instruction.

2.5.1 Function

1) The PID instruction for GL120 and GL130 PLCs perform PID calculations using the equation given below (equation from *2.2.2*) using the values set in holding registers.

$$MV(t_n) = K_1\left[E(t_n) + \frac{K_2}{60}\Sigma E(t_n) \cdot Ts - \boxed{\text{Filter}}\,60K_3\{PV(t_n) - PV(t_{n-1})\} \times \frac{1}{Ts}\right] + M_o$$

tn:	Time in the nth sampling cycle [s]		
Ts:	Sampling cycle [s]	tn = nTs	
K_1:	Proportional gain	$K_1 = 100/PB$	PB: Proportional band [%]
K_2:	Reset rate [repeats/min]	$K_2 = 60/Ti$	
K_3:	Rate time constant [min]	$K_3 = Td/60$	
M_o:	Bias		
Filter:	Filter constant (RGL)		

2.5.2 Structure of PID2 Instruction

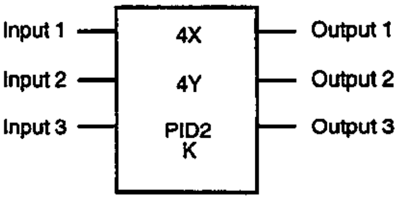


Figure 2.11 Structure of PID2 Instruction

Table 2.1 Structural Elements of PID2 Instruction

Element	Meaning	Possible Settings			
Top (S)	Source reference no.	Holding registers	400001 to 409979 (W00001 to W09979)	Link registers	R10001 to R11004 R20001 to R21004
Middle (P)	Pointer reference no.	Holding registers	400001 to 409991 (W00001 to W09991)	Link registers	R10001 to R11016 R20001 to R21016
Bottom (Z)	Size of destination table	Constant	#00001 to #00255		

1) Size of Elements

PID2 requires three elements (top, middle, and bottom) in the vertical direction on the network.

a) Top Element

The top element, 4X, provides the starting reference number of the source table in which PID parameters are stored. Refer to *Table 2.5* for the meanings of the data set in each register.

b) Middle Element

The middle element, 4Y, represents the starting reference number of the destination table that stores the results of PID calculation results. Refer to *Table 2.6* for the meanings of the data set in each register.

c) Bottom Element

The bottom K sets the sampling cycle of the PID calculation in 0.1-s units. Values can be set from 1 to 255 (0.1 s to 25.5 s)

2.5.3 Specifying Operation

1) Inputs

Table 2.2 PID2 Inputs

Input	ON	OFF
Input 1	Automatic (AUTO) mode	Manual (MAN) mode
Input 2	Integral preload mode ON	Integral preload mode OFF
Input 3	Reverse output MV (manipulated variable) is decreased by increase in E (process error).	Direct output MV (manipulated variable) is increased by increase in E (process error).

2) Outputs

Table 2.3 PID2 Outputs

Output	Conditions that Turn it ON
Output 1	Error in calculation parameters or node lockout
Output 2	Pv (process variable) ≥ CH (process variable upper limit alarm value)
Output 3	Pv (process variable) ≤ CL (process variable lower limit alarm value)

3) Operating Mode

Any of three operating modes, P control, PI control, or PID control, can be set for the PID2 instruction using the parameters set in the source table, as shown in *Table 2.4.*

Refer to *Table 2.5* for parameter details.

Table 2.4 PID Operating Mode

PID Operating Mode	Source Table (Starting Register = Value of 4X)			
	4X+5: Proportional Band	4X+6: 1/Reset Time Constant	4X+7: Rate Time Constant	4X+8: Bias Value
P	5 to 500	0	0	1 to 4095
PI	5 to 500	1 to 9999	0	0
PID	5 to 500	1 to 9999	1 to 9999	0

IMPORTANT

Note (a) The actual reset rate (1/reset time constant) is the set numeric value multiplied by 0.01, i.e., between 0.01 and 99.99 [repeats/min]. Set the reset rate to 0 if it is not used.

(b) The actual rate time constant is the set numeric value multiplied by 0.01, i.e., between 0.01 and 99.99 [min]. Set the rate time constant to 0 if it is not used.

◀EXAMPLE▶

If PID control is carried out with the proportional band PB set to 10 [%], the reset time constant Ti set to 60 [s], and the rate time constant Td set to 15 [s],

$$K_1 = 100/PB,\ PB = 10\ [\%]$$
$$K_2 = 60/Ti = 1\ [repeats/min.]$$
$$K_3 = Td/60 = 0.25\ [min.]$$

Therefore, set 4X+5 to 10, 4X+6 to 100, and 4X+7 to 25.

2.5.4 Register Contents

1) Source Table

The source table is a group of 21 consecutive registers that starts with the holding register specified for 4X. It is used to set the various parameters necessary for PID calculations. *Table 2.5* shows the meanings of the parameters set in the holding registers.

In the register address column of the table, 4X+n refers to the nth holding register counting from the holding register 4X, i.e., from the starting reference number. For example, if 4X is 400100, then 4X+5 is 400105.

In the preset yes/no column, those values marked "Yes" are variable constants, and are numbers which are set before carrying out PID calculations. These values can be varied in a different ladder logic circuit while the PID circuit is operating.

2) Destination Table

The destination table is a group of 9 consecutive registers that starts with the holding register specified for 4Y. It stores the PID calculation error data.

Table 2.6 shows the meanings of the data set in each holding register. In the register address column of the table, 4Y+n refers to the nth holding register counting from the holding register 4Y, i.e., from the starting reference number. For example, if 4Y is 400200, then 4Y+9 is 400209.

Table 2.5 Source Table

Register Address	Contents	Meaning	Range of Numerical Values	Unit	Preset Yes/No
4X	Scaled Pv	The process variable scaled into engineering units, which is updated with each scan. The process variable (4X+13) is scaled from the settings for the upper limit (4X+11) and lower limit (4X+12) in engineering units. The equation is as follows: **Pv** = {(4X+13)/4095} x {(4X+11) – (4X+12)} + (4X+12)	0000 to 9999	Engineering units	No
4X+1	Set point Sp	The set point expressed in engineering units. The following relation must be met. SL (4X+12) < **Sp** < SH (4X+11)	0000 to 9999	Engineering units	Yes
4X+2	Output MV	The manipulated variable. It is clamped (limited) within the range 0 to 4095. This value is used as the output to the Analog Output Module. It can also be used as the input to other destinations, such as cascade loops. It is updated each time a loop is solved.	0000 to 4095	Internal units	See note.
4X+3	Upper limit alarm value; CH (process variable)	The upper limit alarm value of Pv (process variable). If CH is exceeded ladder logic output 2 turns ON. Sp (4X+1) < **CH** ≤ SH (4X+11)	0000 to 9999	Engineering units	Yes
4X+4	Lower limit alarm value; CL (process variable)	The lower limit alarm value of Pv (process variable). If CL is exceeded, ladder logic output 3 turns ON. SL (4X+12) ≤ **CL** < Sp(4X+1)	0000 to 9999	Engineering units	Yes
4X+5	Proportional band PB	The parameter for P (proportional term). The smaller this value, the more effective proportional control is.	0005 to 0500	%	Yes
4X+6	Reset rate K_2	The reset rate (1/reset time constant) parameter for I (integral term). The greater this value, the more effective integral action. The actual value is the value multiplied by 0.01, and the range is 0 to 99.99 [repeats/min]. If it is set to a value greater than 9999, PID calculation is not performed. Set this value to 0 when integral control is not being performed.	0000 to 9999	Repeats/ min	Yes

Register Address	Contents	Meaning	Range of Numerical Values	Unit	Preset Yes/No
4X+7	Rate time constant K_3	The rate time constant parameter for D (derivative term). The larger this value, the more effective derivative action. The actual value is the value multiplied by 0.01, and it is 0 to 99.99 [min]. If it is set to a value larger than 9999, PID calculation is not performed. Set this value to 0 if integral calculation is not being performed.	0000 to 9999	Min	Yes
4X+8	Bias value M_0	The bias added to MV (manipulated variable).	0000 to 4095	Internal units	Yes
4X+9	Integral wind-up upper limit RH	The integral wind-up upper limit. It stores the output upper limit when performing anti-reset wind-up. If the integral value exceeds this value, integral calculation is not performed. It is normally set to the maximum of 4095.	0000 to 4095	Internal units	Yes
4X+10	Integral wind-up lower limit RL	The integral wind-up lower limit. It stores the output lower limit when performing anti-reset wind-up. It is normally set to the maximum value of 0.	0000 to 4095	Internal units	Yes
4X+11	Engineering units upper limit SH	The upper limit in engineering units. It stores the upper limit of engineering unit process variables. For example, if the upper limit is 10 to 500°C, the value would be set to 500. This value corresponds to an analog input value of 4095.	0000 to 9999	Engineering units	Yes
4X+12	Engineering units lower limit SL	The lower limit of engineering units. It stores the lower limit of engineering unit process variables. For example, if the lower limit is 10 to 500°C, the value would be set to 10. This value corresponds to the analog input value of 0.	0000 to 9999	Engineering units	Yes
4X+13	Process variable (internal units) PV	The process variable input from the A/D converter.	0000 to 4095	Internal units	No
4X+14	Pointer of PID loop counter register OFS	Used to set the pointer value for the PID loop counter. The pointer value tells the register number which stores the number of loops solved in each scan in odd numbers from 400000. For example, if the number of solved loops is stored in register 401236, 1236 will be set in 4X+14. Set the pointer to 0 if it is not being used.	0000 to 9999	—	Yes
4X+15	Maximum number of loops (loops/scan) MAX	Used to set the maximum number of PID loops to be solved each scan. If the contents of 4X+14 are 0, 4X+15 is set to 0.	0000 to 9999	—	Yes
4X+16	Pointer for reset feedback input RP	Used to set the pointer value for reset feedback input. The pointer value is expressed in odd numbers from 400000. For example, if the feedback register (Fr) is 402250, 4X+16 is set to 2250. It is usually set to 4X+2. This Fr value is used in integral calculations. Fr value is in internal units from 0 to 4095.	0000 to 9999	—	Yes
4X+17	Output limit upper limit CLH	Used to set the output clamp value which indicates the upper limit of the manipulated variable (MV). It is usually set to 4095.	0000 to 4095	Internal units	Yes

Register Address	Contents	Meaning	Range of Numerical Values	Unit	Preset Yes/No
4X+18	Output limit lower limit CLL	Used to set the output clamp value which indicates the lower limit of the manipulated variable (MV). It is usually set to 0.	0000 to 4095	Internal units	Yes
4X+19	RGL constant RGL	Used to set the RGL (Rate Gain Limit) constant. This value is the setting for the RGL derivative filter, and can be set from 2 to 30. The smaller the value, the more effect the filter has.	0002 to 0030	—	Yes
4X+20	Derivative preload register pointer TP	Used to set the derivative preload input pointer. For example, if the derivative preload register (Tr) is 400956, this register is set to 956. When input 1 is ON (automatic mode) and input 2 is ON (integral preload mode) is ON, the contents of the Tr register are used in derivative calculations.	0000 to 9999	—	Yes

Note During manual operation, the value can be altered to control the output. During automatic operation, PID calculation values are updated each time a loop is solved.

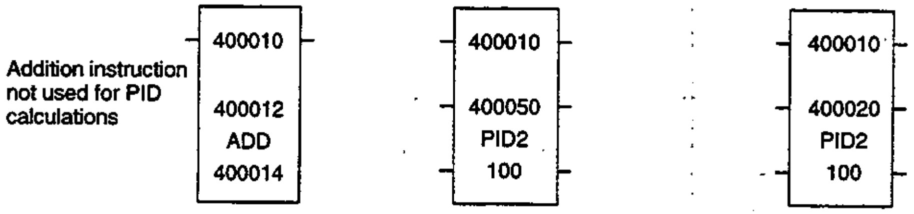
IMPORTANT

Using the PID2 Instructions

a) Check to be sure that other instructions are not written to the same holding registers as those used in the source table.

b) Check that other instructions are not written to the same holding registers as those used in the destination table.

c) Check that the source table and destination table do not overlap.



Addition instruction not used for PID calculations

Duplicate usage of source table registers

Overlapping source and destination tables

Figure 2.12 Example of Using Incorrect Register

d) Check that the settings within the source table satisfy the following relationships.

$$0 \le \underset{\text{Engineering units upper limit}}{(4X+12)} \le \underset{\text{Lower limit alarm value}}{(4X+4)} \le \underset{\text{Set point}}{(4X+1)} \le \underset{\text{Upper limit alarm value}}{(4X+3)} \le \underset{\text{Engineering units upper limit}}{(4X+11)} \le 9999$$

$$0 \le \underset{\text{Output clamp lower limit}}{(4X+18)} \le \underset{\text{Output clamp upper limit}}{(4X+17)} \le 4095 \qquad 0 \le \underset{\text{Integral wind-up lower limit}}{(4X+10)} \le \underset{\text{Integral wind-up upper limit}}{(4X+9)} \le 4095$$

$$2 \le \underset{\text{RGL}}{(4X+19)} \le 30$$

Table 2.6 Destination Table

Register Address	Register Contents
4Y	**Loop Status Register** 16 Refer to Note (1) 15 Top input automatic/manual status ON: 1, OFF: 0 14 Middle input integral pre-load mode status ON: 1, OFF: 0 13 Bottom input reverse output/direct output status ON: 1, OFF: 0 12 Calculation sign bit (Negative: 1, Positive: 0) Refer to Note (2). 11 Process variable vs integral wind-up upper limit value (RH, RL) Exceeds: 1, Normal: 0 10 Always 0. 9 Always 1. 8 Process error sign bit. Positive (SP≥PV): 0, negative (SP<PV): 1 7 4X+14 (PID loop counter pointer) pointer value Normal: 1, Abnormal or not used: 0 6 Stop solving PID loop: 1, Solve: 0 (always 0 in manual) 5 Wind-down mode: 1, Normal: 0 (Integral function recovered when anti-reset wind-up function changes from enabled to disabled.) 4 Request solving PID loop passing through the sampling cycle k (normally 0, 1 to request solving). Refer to Note (3). 3 Bottom output ON (Pv lower limit alarm) ON: 1, OFF: 0 2 Middle output ON (Pv upper limit alarm) ON: 1, OFF: 0 1 Top output ON (node lockout or parameter error) ON: 1, OFF: 0

IMPORTANT

<table>
<tr><th>Register Address</th><th colspan="3">Register Contents</th></tr>
<tr><td>4Y
(cont'd)</td><td colspan="3">Note (1) Bit 16 is set to "1" when solving a PID loop (CPU is STOP→RUN, preparation of PID2 instructions online) or after initialization is completed.
In the clearing (initialization) of this bit and the initial solving, the following operations are carried out during one scan.
The contents of the loop status register are reset.
The current value of the real time clock is stored in holding register 4Y+2.
The contents of holding registers 4Y+3, 4Y+4, and 4Y+5 (registers related to integral calculations) are cleared to 0.
The contents of holding register 4X+13 (process variable) are multiplied by 8, and stored in holding register 4Y+6.
Holding register 4Y+7 (process error) and 4Y+8 (sampling time) are cleared.
(2) This bit is used to determine the sign for internal calculations, and the status of the bit will sometimes alternate between 0 and 1 when calculations are being performed normally.
(3) If the sampling cycle comes and there is a request to perform calculations, this will be set to 1. Once it is solved it returns to 0. As a result, even when calculations are being performed normally, the status will alternate between 1 and 0. If the 0 status does not change, however, a lockout will occur. If a lockout occurs, calculations for the PID2 instructions will not be performed, and bit 1 and bit 6 will change to 1. (Refer to <i>Table 2.5</i> for further information on lockouts.)</td></tr>
<tr><td rowspan="28">4Y+1</td><td colspan="3">Error Status</td></tr>
<tr><td>Error Code</td><td>Meaning</td><td>Register in Source Table to Check</td></tr>
<tr><td>0000</td><td>No error</td><td>—</td></tr>
<tr><td>0001</td><td>SP (set point in engineering units) exceeds 9999</td><td>4X+1</td></tr>
<tr><td>0002</td><td>CH (upper limit alarm value) exceeds 9999</td><td>4X+3</td></tr>
<tr><td>0003</td><td>CL (lower limit alarm value) exceeds 9999</td><td>4X+4</td></tr>
<tr><td>0004</td><td>PB (proportional band) is less than 5</td><td>4X+5</td></tr>
<tr><td>0005</td><td>PB exceeds 500</td><td>4X+5</td></tr>
<tr><td>0006</td><td>K_2 (reset rate) exceeds 99.99 [repeats/min]</td><td>4X+6</td></tr>
<tr><td>0007</td><td>K_3 (rate time constant) exceeds 99.99 [min]</td><td>4X+7</td></tr>
<tr><td>0008</td><td>M_o (bias) exceeds 4095</td><td>4X+8</td></tr>
<tr><td>0009</td><td>RH (integral wind-up upper limit) exceeds 4095</td><td>4X+9</td></tr>
<tr><td>0010</td><td>RL (integral wind-up lower limit) exceeds 4095</td><td>4X+10</td></tr>
<tr><td>0011</td><td>SH (engineering units upper limit) exceeds 9999</td><td>4X+11</td></tr>
<tr><td>0012</td><td>SL (engineering units lower limit) exceeds 9999</td><td>4X+12</td></tr>
<tr><td>0013</td><td>SH ≤ SL (normally SH > SL)</td><td>4X+11 and 4X+12</td></tr>
<tr><td>0014</td><td>Sp exceeds SH (normally Sp ≤ SH)</td><td>4X+1 and 4X+11</td></tr>
<tr><td>0015</td><td>Sp is less than SL (normally Sp ≥ SL)</td><td>4X+1 and 4X+12</td></tr>
<tr><td>0016</td><td>N (maximum number of loops/scan) exceeds 9999</td><td>4X+15</td></tr>
<tr><td>0017</td><td>RP (reset feedback pointer) exceeds 9999</td><td>4X+16</td></tr>
<tr><td>0018</td><td>CLH (output clamp upper limit) exceeds 4095</td><td>4X+17</td></tr>
<tr><td>0019</td><td>CLL (output clamp lower limit) exceeds 4095</td><td>4X+18</td></tr>
<tr><td>0020</td><td>CLL > CLH (normally CLL ≤ CLH)</td><td>4X+17 and 4X+18</td></tr>
<tr><td>0021</td><td>RGL is less than 2</td><td>4X+19</td></tr>
<tr><td>0022</td><td>RGL exceeds 30</td><td>4X+19</td></tr>
<tr><td>0023</td><td>When input 2 is ON in automatic mode, TP (integral preload input pointer) is out of range</td><td>4X+20</td></tr>
<tr><td>0024</td><td>When input 2 is ON in automatic mode, TP is 0</td><td>4X+20</td></tr>
<tr><td>0025</td><td>Node is lockout (scan is too fast)</td><td>See note.</td></tr>
</table>

Register Address	Register Contents		
4Y+1 (cont'd)	0026	F (PID loop counter pointer) is 0	4X+14 and 4X+15
	0027	F is out of range	4X+14 and 4X+15
	Note	If lockouts (parameters are normal and output 1 is ON) occur frequently, increase the MAX setting (maximum value for number of PID loops solved in each scan) in 4X+15. Slow down the scan time using a lockout or a constant sweep. Lockouts may also occur if the counter register is not cleared.	
4Y+2	Loop timer register. After the loop calculations have been completed, a realtime clock is read, and the time is stored in this register. The difference between the value stored in this register and the realtime clock becomes the elapse time. Loop calculations are carried out in scans in which elapse time ≥ sampling time (K × 10).		
4Y+3	Calculation values for integral terms	Output value of integral terms (internal units)	
4Y+4		Compensation term 1 for output value of integral term	
4Y+5		Compensation term 2 for output value of integral term	
4Y+6	PV multiplied by 8 (numerical value of register 4X+13 multiplied by 8). This value is used in derivative calculations.		
4Y+7	The absolute value of process error $\lvert SP - PV \rvert$. The sign bit of the process error is shown in the 8th bit of 4Y+1. This value is updated after loop calculations (internal units).		
4Y+8	The sampling time set in the bottom of the ladder logic is shown in units of 10 ms (used by the system).		

2.6 Top Register Contents

This section explains in detail the contents of the top register.

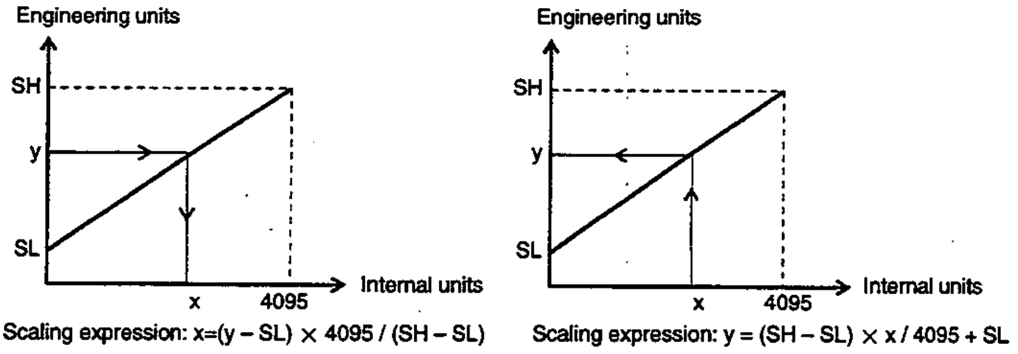
2.6.1 Scaling

1) PID calculations are carried out with all constants and parameters within the range of internal units (no units) of full scale 12 bits (0 to 4095).

   Variables and parameters relating to processes are expressed in engineering units (°C, m/min, kg/cm^2, etc.). The scaling between these is performed automatically, as shown in *Figure 2.13,* on the scaling allocated in engineering units as long as the lower limits are stored in the source table.

INFO The results of scaling operations are truncated at the decimal point.

4X: Scaled Pv
4X+11: Engineering units upper limit (SH)
4X+12: Engineering units lower limit (SL)



Scaling expression: x=(y – SL) × 4095 / (SH – SL)

Scaling expression: y = (SH – SL) × x / 4095 + SL

(a) Scaling: Engineering Units → Internal Units

(b) Scaling: Internal Units → Engineering Units

Figure 2.13 Scaling

◀EXAMPLE▶ **PV Scaling**

a) Scaling: Engineering Units → Internal units

$$\underset{(PV)}{\boxed{4X+13}} = \left(\underset{(Pv)}{\boxed{4X\quad}} - \underset{(SL)}{\boxed{4X+12}}\right) \times 4095 / \left(\underset{(SH)}{\boxed{4X+11}} - \underset{(SL)}{\boxed{4X+12}}\right)$$

b) Scaling: Internal Units → Engineering Units

$$\underset{(Pv)}{\boxed{4X}} = \underset{(PV)}{\boxed{4X+13}} \times (\underset{(SH)}{\boxed{4X+11}} - \underset{(SL)}{\boxed{4X+12}}) / 4095 + \underset{(SL)}{\boxed{4X+12}}$$

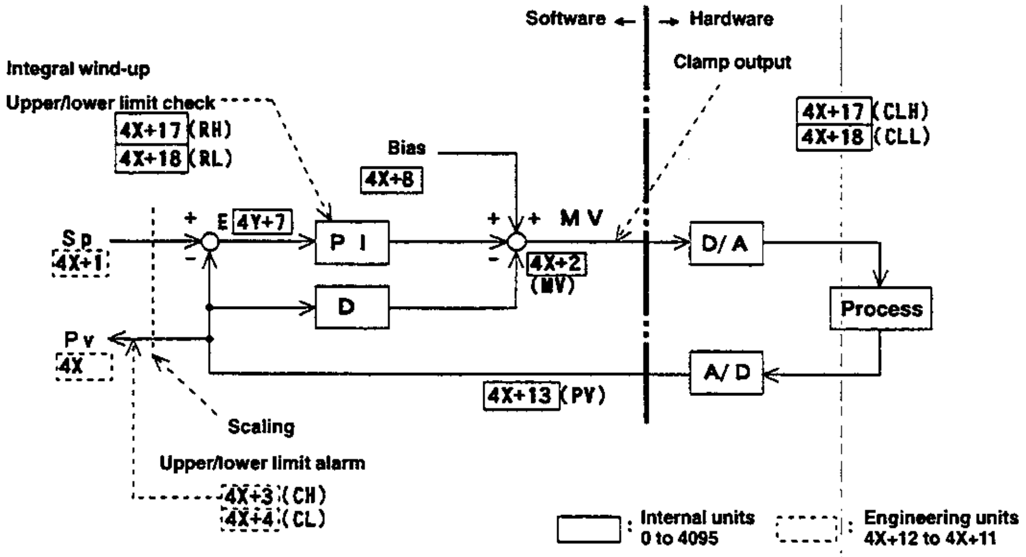


Figure 2.14 Scaling and Range Check

2.6.2 Anti-reset Wind-up

1) If the set point changes suddenly during startup, the process error will also increase suddenly, causing the value of integral terms to become extremely large, and the manipulated variable will be saturated. When the derivative of the process error is saturated, it's called reset wind-up or integral wind-up.

2) The following describes only the integral action in an open loop, and the behavior when the process error input changes, as shown in *Figure 2.15 (a)*.

   a) If there are no saturated values, the manipulated variable moves from 0→1→2→3→4. When the integral variable increases, the manipulated variable increases proportionally, and when the integral variable decreases, the manipulated variable decreases proportionally, fulfilling the original operation of integration.

   b) If the manipulated variable is saturated, as shown in *Figure 2.15 (b)*, the output of the integrator goes from 0→1→2→3→4 as before, but the actual manipulated variable goes from 0→1→5→3→4.

   Here, the original operation does not increase the manipulated variable in proportion to the increase in the integral variable of the process error between 1 and 5, but this is caused by saturated values and is unavoidable. Between 5 and 3, the manipulated variable does not decrease despite the decrease in the integral variable of process error. This is because the integrator output is wound up from 1 to 2, and the manipulated variable will not decrease in proportion to the decrease in the integral variable of the process error until wound back from 2 to 3. The original operation is not achieved.

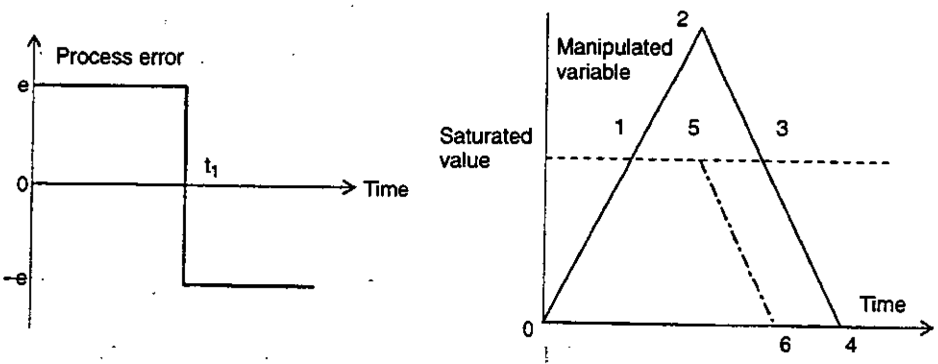
c) The same applies when incorporated in a feedback control system, the recovery of the original operation is delayed only for the period of winding back, so the process error setting is delayed, resulting in an increase in the tendency to overshoot.

d) As a countermeasure, when the saturation value is reached, the integral action in the direction that would exceed that value is stopped (integration in the direction which returns to the unsaturated area is continued). Integration is stopped at 1, and the integrator output moves from 1 to 5 in the same way as the manipulated variable. After that, the original operation is recovered at 5, where the decrease in process error starts, then the manipulated variable moves though 5 to 6. This operation is referred to as anti-reset wind-up. When this operation is unnecessary, set the upper limit to 4095 and the lower limit to 0.

Integral wind-up upper limit: RH
(register: 4X+9)

Integral wind-up lower limit: RL
(register: 4X+10)

$0 \le RL \le RH \le 4095$



(a) (b)

Figure 2.15 Reset Wind-up

2.6.3 Output Control Upper Limit, Lower Limit

MV (output value) is clamped (limited) by the upper/lower limits.

Output limiting upper limit: CLH
(register: 4X+17)

Output limiting lower limit: CLL
(register: 4X+18)

$0 \leq CLL \leq CLH \leq 4095$

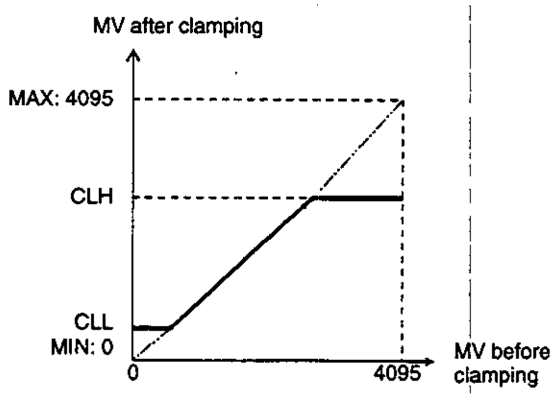


Figure 2.16 Output Limiting Upper/Lower Limits

2.6.4 Bumpless Switching Between Manual and Automatic

1) Generally when starting control actions, the operator controls the adjustment valve of the adjustment gauge manually. After that, it is switched to automatic operation when the status of process has stabilized to some extent. When switching between manual and automatic, the manipulated variable output can change discontinuously, and there is a danger that the process (control object) will be affected dramatically by these changes. Bumpless switching helps to eliminate such discontinuous changes.

2) Bumpless switching is defined below.

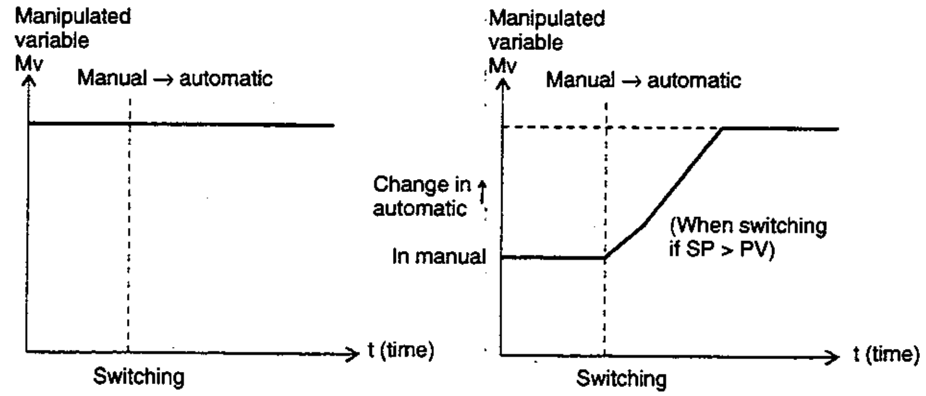
a) Balanced Bumpless Switching

Balanced bumpless switching refers to switching when the process error is maintained at zero by manual operation, i.e., switching in a balanced state. We will use this as the limited meaning of bumpless switching. (*Figure 2.17* (*a*))

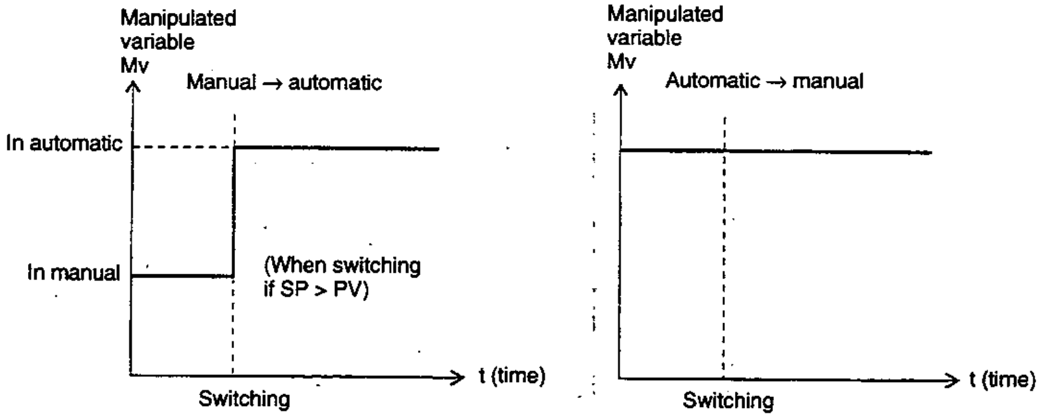
b) Balanceless Bumpless Switching

When the balance in a) above is not assumed, it is referred to as balanceless bumpless switching. (*Figure 2.17 (b)*) When switching from manual to automatic the change is continuous, and PID calculations causes a move towards the set point.

3) When switching from automatic to manual, the value immediately before switching to manual remains in the output register (MV), allowing switching to take place smoothly while maintaining that value. (*Figure 2.17 (d)*)



(a) Balanced Bumpless Switching **(b) Balanceless Bumpless Switching**



(c) Without Bumpless Switching **(d) Switching from Automatic to Manual**

Figure 2.17 Manipulated Variable at Manual/Automatic Switching

4) PID2 instructions can switch as follows:

IMPORTANT If using bumpless switching, set the reset feedback input pointer (4X+16) to output register (4X+2). (Refer to *Table 2.5.*)

a) P Control

Balanced bumpless switching can be performed. Set the SP to the target value, set the PV to the SP using manual operation, and then switch to automatic operation. (*Figure 2.17 (a)*)

When switching from manual operation to automatic operation, switching can be carried out with the manipulated variable fixed at a particular value even if there is still a process error while in manual operation. Even if there is process error between the SP and PV, the PV will not change towards SP.

During manual operation, the difference between the calculated output value and the manual output value is calculated, and then that difference is used as the compensation in automatic mode. This is because it is calculated internally so that it matches the output value during manual operation. Therefore, if the SP is changed after switching to automatic operation, the PV will also change to maintain the same compensation value.

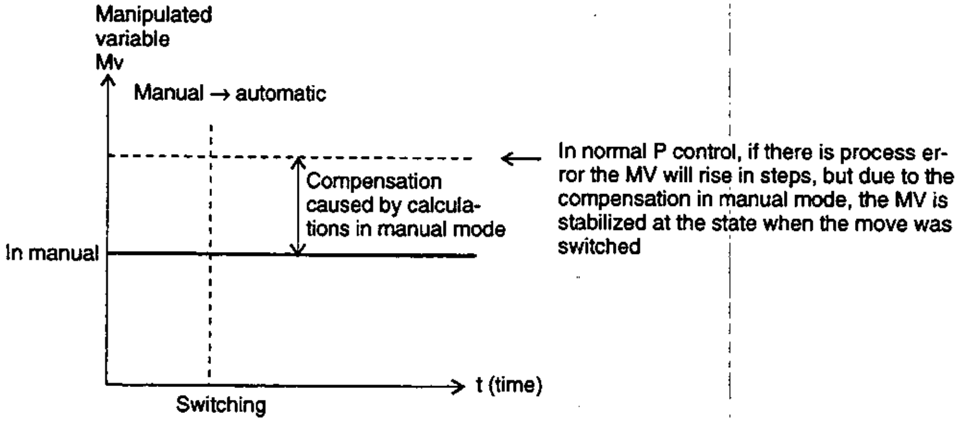


Figure 2.18 When Switching from Manual/Automatic: SP > PV

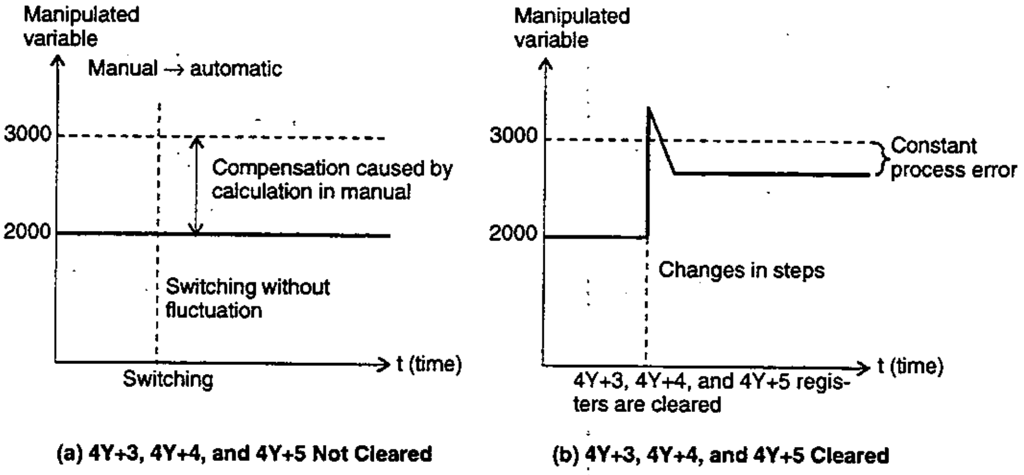
If balanced bumpless switching is not carried out, 4Y+3, 4Y+4 and 4Y+5 must be cleared. When switching to automatic has been completed, the PV is not stable in the same state, but changes discontinuously towards the SP. (*Figure 2.17 (c)*) Because there is no I control, however, a constant process error remains.

◀**EXAMPLE**▶

For example, if MV in manual mode is set to 2000, the PV will also approach 2000. The SP is set to 3000, and the mode is switched to automatic (PB = 100, internal units = engineering units, bias = 2000). The PV value does not move towards the SP, but stabilizes at the value set during manual operation (PV ≅ 2000, SP = 3000). (Refer to *Figure 2.19 (a).*)

In this state, if 4Y+3, 4Y+4, and 4Y+5 registers are cleared, the PV will increase to near the SP. (Compensation values have been cleared, so the P control works so that the PV equals the SP.)

Accordingly, in switching from manual to automatic when using P control, switching should be carried out after setting the SP to the target value of 3000 and manually setting the MV to near 3000 (PV = 3000) (state at which process error = 0).



(a) 4Y+3, 4Y+4, and 4Y+5 Not Cleared **(b) 4Y+3, 4Y+4, and 4Y+5 Cleared**

Figure 2.19 P Control Manual → Automatic Switching Example

Note P control is not affected by the ON or OFF state of input 2 (integral preload mode).

b) PI Control

Balanceless bumpless switching is possible under PI control. If switching from manual to automatic with a process error, the integral action will move the PV towards the SP. (See *Figure 2.17 (b)*.)

Even if switching to automatic while manually maintaining a balance between SP values and PV values, it will switch smoothly. (See Balanced Bumpless Switching: *Figure 2.17 (a)*.)

◀EXAMPLE▶

1) Integral Preload Mode (Input 2: ON) Usage Example

In the same way as for P control, when switching from manual to automatic the PV will not move towards the SP, but will be maintained at the state they were before switching. In this case, the integral preload mode of input 2 is used. Integral preload mode switches the integral input register to the register set in 4X+20 (integral preload register: Tr) with both input 1 and input 2 ON. Apart from that, the register set in 4X+16 (feedback register: Fr) is used. (Refer to *Figure 3.3* and *Table 2.4.*)

If input 2 is switched OFF after switching to automatic and is in a stable state, the PV will move towards the SP. This is because the normal integral action switches to the feedback register and operates.

2) Using Integral Preload Mode

a) Not Using a Bias

Set the pointer (register number) of 4Y+3 register to the 4X+20 register.

b) Using a Bias

Store values that have added the bias and 4Y+3 in a register and set the pointer of that register (register number) in 4X+20 register.

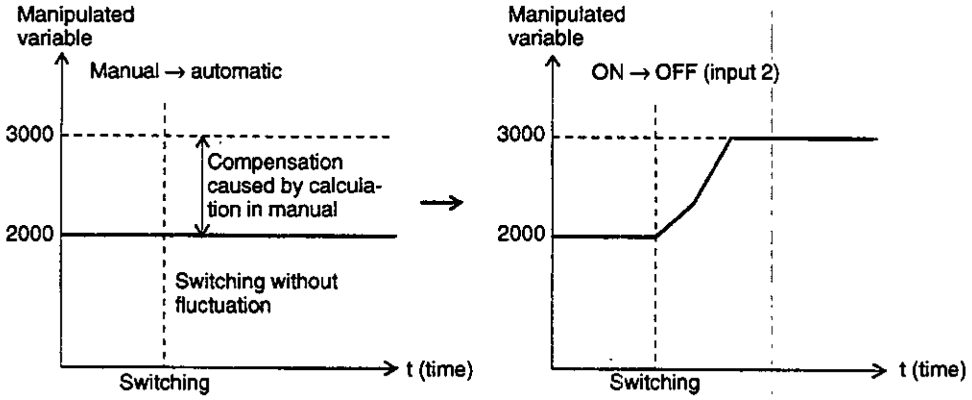
Note If the following conditions are not satisfied after switching to automatic operation, the PV may not stabilize (although it will not change dramatically or discontinuously), but may change little by little towards the SP.

4Y+3 must be between 0 to 4095.

The output value must be stable when switching to automatic.

For example, if MV is manually set to 2000, the PV will also move to a value near 2000. After that, set the SP to 3000 and switch to automatic. (PB = 100, K_2 = 1.00, internal units = engineering units, bias = 0, set 4Y+3 register number (= Y+3) to 4X+20, input 2 = ON) The PV will not move towards the SP, but will stabilize at the value it had when switched (PV ≅ 2000, SP = 3000). (See *Figure 2.20 (a).*)

If input 2 is turned OFF in this state, it will switch from the integral preload register to the feedback register, and the normal integral action will work to change the PV so they conform with the SP. (See *Figure 2.20 (b).*)



(a) Integral Preload Mode ON **(b) Integral Preload Mode OFF**

Figure 2.20 Integral Preload Register Usage Example

◀EXAMPLE▶ **Calculations for PID2 in Manual Mode**

Manual operation calculates using the equation (1) below.

Equation: $$\Delta FV = FV - GE - Bias \ldots\ldots\ldots\ldots \quad (1)$$

FV: Contents of feedback register

GE: Proportional terms + derivative terms output value

Bias: Bias value

The value of ΔFV is set in 4Y+3. This 4Y+3 is the register into which integral term output value is input in automatic operation. In manual operation, integral calculation is not performed, and calculation values, such as those shown above, are input into the 4Y+3 integral term output register.

The feedback register (Tr) is the register to which 4X+16 is set, and FV expresses its contents. Normally, 4X+2 (output: MV) is set to the 4X+16 register, so, as the following example, FV = MV. If equation (1) is rewritten FV = MV it becomes expression (1)'.

Equation: $\Delta MV = MV - GE - Bias$ .... (1)'

Calculation output value MV' under manual operation is as shown in expression (2).

$$MV' = GE + Bias \quad \text{..........} \quad (2)$$

Equation (1)' can be rewritten as equation (3).

Equation: $\Delta MV = MV - MV'$ .... (3)

For example,
Proportional band = 50%
Bias = 1000
MV = 2000
SP = 2200

In order to simplify the example, there will be no derivative action, and engineering units will be matched to internal units.

In manual operation, MV will be set to 2000, and PV to 1980.

$$\begin{aligned} E &= SP - PV = 2200 - 1980 \\ &= 220 \end{aligned}$$

$$\begin{aligned} \Delta MV &= MV - GE - Bias \\ &= MV - (100/PB) \times E - Bias \\ &= 2000 - (100/50) \times 220 - 1000 \\ &= 2000 - 2 \times 220 - 1000 \\ &= 2000 - 440 - 1000 \\ &= 560 \end{aligned}$$

This ΔMV (= 560) value is set in 4Y+3.

1) Not Using Derivative Action

After switching from manual to automatic operation, the calculations for the PID2 instruction are as shown in equation (4) below.

Equation: $$\begin{aligned} MV &= GE + Bias + \Delta MV \quad \text{........} \quad (4) \\ &= 440 + 1000 + 560 \\ &= 2000 \end{aligned}$$

Therefore, the results of calculations in automatic mode are equal to the output value (MV = 2000) in manual mode.

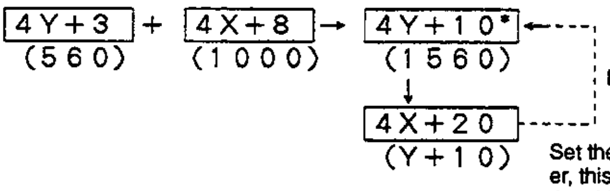
2) Using Derivative Action

Even if switching from manual to automatic as shown in (a), the ladder logic input 2 and 4X+20 integral preload register are used to keep the calculation values constant.

◀EXAMPLE▶ **Setting to 4X+20 Register**

a) Turn ON input 2 in advance in manual mode.

b) The reference number of the register that has the value of the register contents of 4Y+3 (integral calculation term) register contents added to the contents of 4X+8 (bias) is set to 4X+20 (integral preload register). When bias is not used, set the register number of 4Y+3 to 4X+20.



4Y+10 register must be a register not used for anything else

Set the reference number of 4Y+10. However, this shall be the number which represents the 4Y+10 register with the leading 4 removed.

Compensation terms like those as shown above (the contents of the register set to 4X+20) disable integral action, even if an integral constant is set when switched to automatic operation. The value of integral term I is the constant value 4Y+3 = 560,

$$\begin{aligned} MV &= GE + Bias + 1 \\ &= 440 + 1000 + 560 \\ &= 2000 \end{aligned}$$

Therefore, the results of calculations in automatic operation are equal to the output values in manual operation (MV = 2000).

Execution Timing and Monitoring

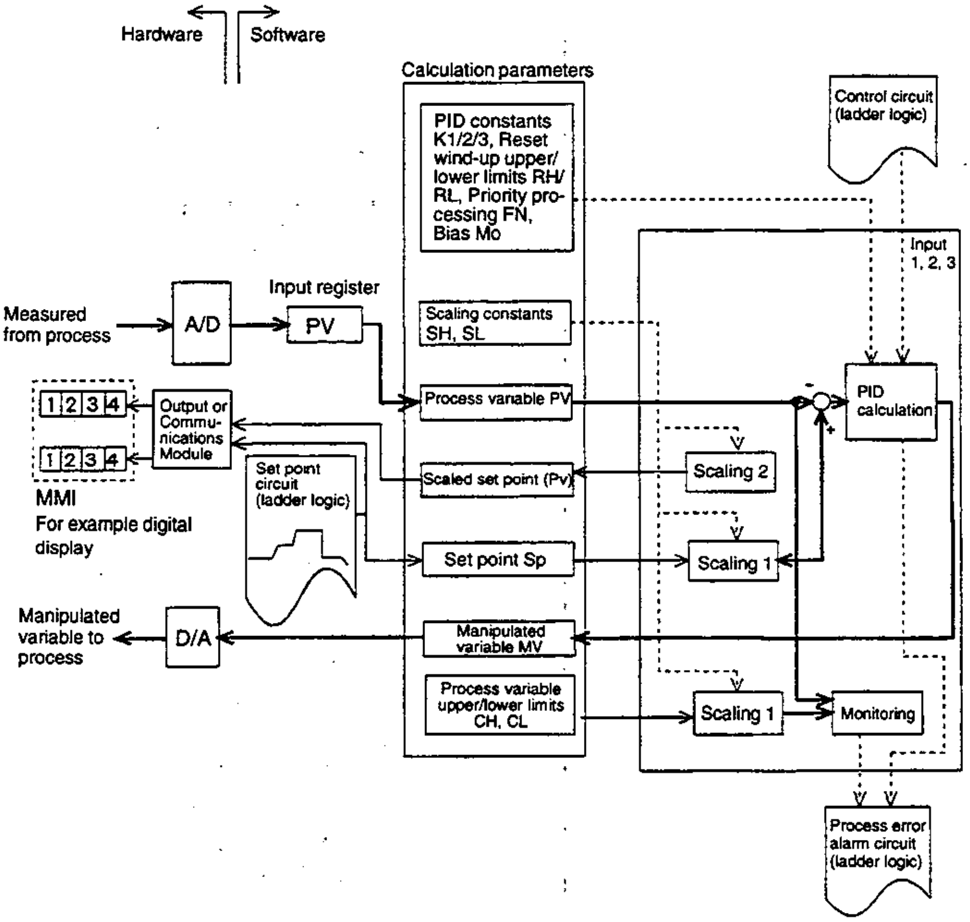
3

This chapter describes the execution timing, monitoring operations and processing priority for multiple PID2 instructions.

3.1 Data Flow for PID2 Instructions

- This section explains the data flow for PID2.

1) The relations of calculation parameters and the data flow for PID2 instructions are shown in *Figure 3.1.*



Note (1) Scaling 1: Engineering units → internal units
Scaling 2: Internal units → engineering units

(2) ➔ Number of internal units (e.g., 0 to 4095)
→ Number of engineering units (e.g., 0°C to 1000°C)

Figure 3.1 PID Control System and Data Flow

3.2 Ladder Logic I/O Functionality

This section explains functions in the ladder logic I/O.

3.2.1 Ladder Logic Inputs

The following functions are provided by the ON/OFF status of input 1, input 2, and input 3.

1) Input 1: Automatic/Manual Mode

Input 1: ON .... Automatic mode
OFF ... Manual mode

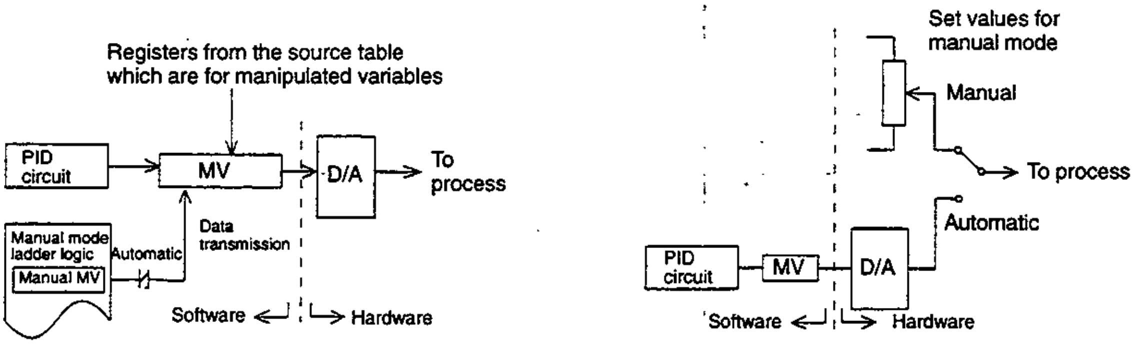
In automatic mode, PID calculations are executed according to calculation parameters, and the results are output as the manipulated variable (Mv). The manipulated variable is output to the top element register (4X+2).

Scaling and monitoring is performed in manual mode. Part of PID calculations are also executed in manual mode to enable bumpless switching from manual to automatic operation. The results are output to the middle element register (4Y+3). As shown in *Figure 3.2 (c)*, however, the PID calculation section and the manipulated variable output register (4X+2) are separate, so manipulated variables are not altered.

IMPORTANT

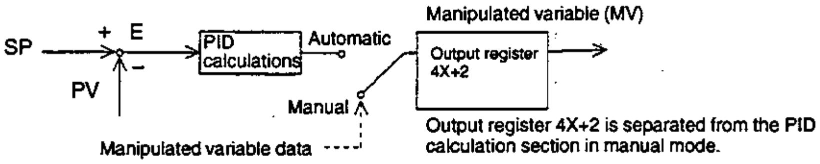
Note (a) After switching from automatic to manual, the value calculated immediately before switching remain as the manipulated variable (MV). Accordingly, when providing the manipulated variable in manual mode inside the PLC, it is necessary to prepare separate ladder logic circuits for the manual mode, as shown in *Figure 3.2 (a)*. These ladder logic circuits may, for example, convert engineering units of the manual process variable into internal units and provide the manipulated variable (MV). *Figure 3.2 (b)* shows an example in which circuits for the manual mode outside the PLC system was provided.

(b) To switch bumplessly from manual to automatic, the value set in manual in the upper element register (4X+2), as shown in *Figure 3.2 (a)*, is used as the manipulated variable (MV) so there is no particular problem. If the manipulated variable is set outside, as shown in *Figure 3.2 (b)*, however, it is unknown what value will enter the upper element register, so bumpless switching cannot be carried out. It is therefore necessary to set the value to the PID instruction's top element register that have been converted into internal units.



(a) Manual Mode Settings Generated by Ladder Logic

(b) Manual Mode Settings Generated Outside the PLC

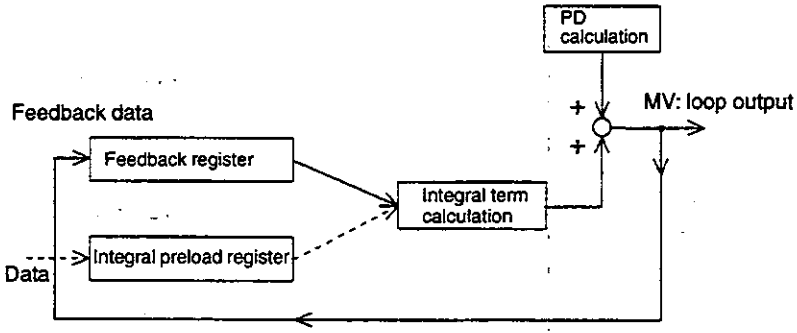


(c) Relation Between Automatic/Manual Modes and Output Register 4X+2

Figure 3.2 Manual Mode Set Values

2) Switching to the Integral Preload Register

When input 1 and input 2 are both ON, the system switches from the feedback register to the integral preload register. This register is related only to integral terms, and determines whether feedback values or other values will be used in the integral calculation. Refer to the block diagram *Figure 3.3* below. Refer to *2.6.4 Bumpless Switching Between Manual and Automatic.*



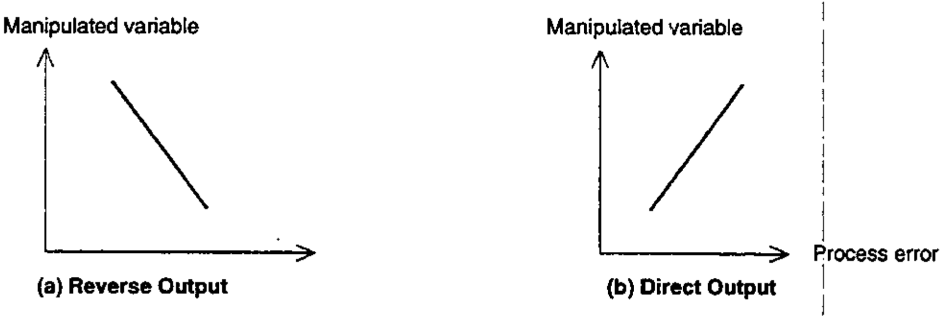
---- : Use only when input 1 and input 2 are ON

Figure 3.3 Switching Registers

3) Reverse/Direct Output (Input 3)

Input 3: ON .... Reverse output
OFF ... Direct output

Input 3 defines the relationship between an increase/decrease in the process error and an increase/decrease in the manipulated variable. (See *Figure 3.4.*) Either of the following two relationships can be selected according to the characteristics of the external actuator that operates on the manipulated variable from the PLC, and the fail-safe measures for the system.



(a) Reverse Output (b) Direct Output

Figure 3.4 Reverse Output/Direct Output

3.2.2 Ladder Logic Outputs

Outputs 1, 2, and 3 turn ON to indicate the following status. (Refer to *3.4 Monitoring Operations* for details.)

1) Output 1 ON

Output 1 turns ON and PID calculation is not performed when the following alarms are in effect.

a) When the calculation parameters are incorrect

b) When it was not executed within two sampling cycles (node lockout)

2) Output 2 ON

Output 2 turns ON when the value set in 4X+3 (Pv upper limit alarm value) is exceeded. The Pv is not limited, it is not clamped and only an alarm is given.

3) Output 3 ON

Output 3 turns ON when the value set in 4X+4 (Pv lower limit alarm value) is exceeded. The Pv is not limited, it is not clamped and only an alarm is given.

3.3 Execution Timing

This section explains the timing of PID calculations and sampling cycles.

1) Sampling Cycles and Execution of Calculations

Calculations are carried out using the following equation (equation (c) from *2.2.2*) using PID constants K_1, K_2, K_3 and sampling cycle Ts.

$$MV(t_n) = K_1 \times [E(t_n) + \frac{K_2}{60}\Sigma E(t_n) \times Ts - \boxed{\text{Filter}}\ 60K_3\{PV(t_n) - Pv(t_{n-1})\} \times \frac{1}{Ts}] + M_o$$

Sampling cycles and the execution of calculations is explained below. (Refer to *Figure 3.5.*)

a) The CPU's realtime clock is used to time the sampling cycle. Each of the PID circuits has interval timer driven by this realtime clock.

b) When the PID circuit is solved during the scanning cycle, if
(elapse time of interval timer) ≥ (set sampling cycle),
then the operations concerning the PID circuit (PID calculation, control, monitoring, priority processing) will be executed. The interval timer will be reset immediately afterwards.

c) If the above equation is not satisfied when the PID circuit is solved, only monitoring of parameters and scaling will be performed. This is referred to as non-execution. In this case, the interval timer is not reset, and timing continues.

Note Execution and non-execution are not related to ON/OFF status of input 1, input 2, and input 3.

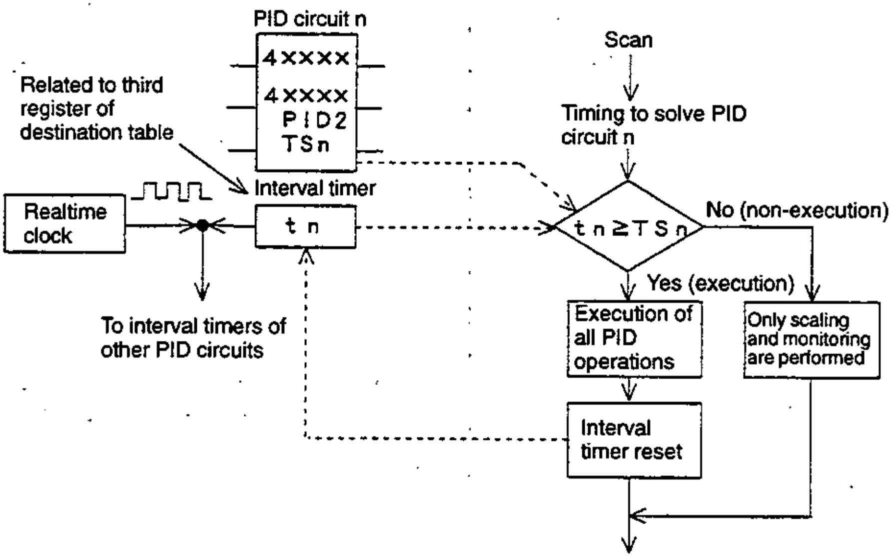


Figure 3.5 Execution and Non-execution

d) The timing of the interval timer is independent of scanning, but timing to solve the PID circuit is affected by the scan time. Rather than the interval timer value (actual time), the value set in the function block (sampling cycle) is used as Ts in equation (c) of *2.2.2.* Consequently, the larger the sampling cycle is in relation to scan time, the better the accuracy of PID calculations. (*Figure 3.6*)

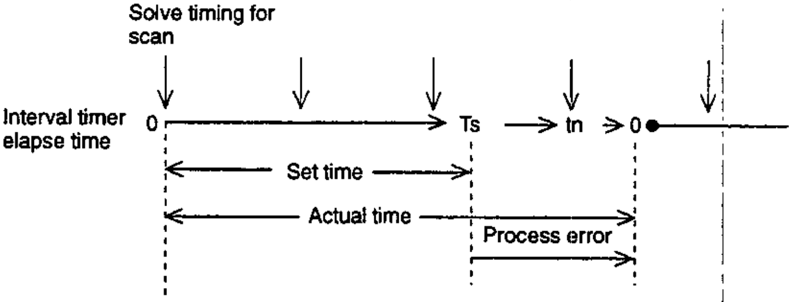


Figure 3.6 Interval Timer

3.4 Monitoring Operations

This section explains the system monitoring operations for each scan.

1) Monitoring Upper/Lower Limit of Process Variables

Pv (engineering units) will be used for the process variable. Supposing the process variable upper limit is CH, and the lower limit is CL (engineering units), outputs are as shown in *Figure 3.7.*

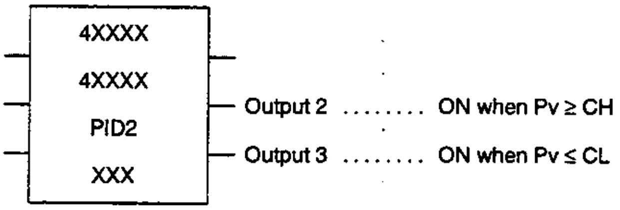


Figure 3.7 Ladder Logic Block

This monitoring is performed without relation to execution/non-execution of PID calculations, or the ON/OFF status of input 1, input 2, and input 3. Outputs 2 and 3 are only alarm outputs as a result of monitoring, and the process variable or manipulated variable are not limited.

2) Calculation Execution Impossible Alarm

In the following two cases, output 1 turns ON and PID calculations are not performed.

a) **Incorrect Calculation Parameters**

Error codes are displayed in the ladder logic middle register (4Y+1). Change the parameters according to the contents of these process errors. (Refer to *Table 2.5.*)

IMPORTANT

Note (a) In ladder logic output 1, a range check is not carried out for the upper/lower limit set values of output value (MV) and process variable (Pv).

- Output values (MV) are subject to clamping by the upper/lower limit set values and an error is not generated.
- The process variable (Pv) can be checked for the upper/lower limit set values using ladder logic output 2 and ladder logic output 3.

These checks are performed with no relation to execution, non-execution, or the ON/OFF status of inputs 1, 2, and 3.

Note (b) If output 1 is turned ON due to incorrect calculation parameters, it is necessary to initialize operation by setting the contents of the registers for the initialization flag to 0 and starting from the initial state after correcting the parameters. Initialization clears the LSB bit (bit number 16) of the 4Y register. (Refer to *Table 2.5.*)

b) **Execution Not Performed within Two Sampling Cycles (Node Lockout)**

PID calculations may be delayed when the priority processing discussed in the next section is performed. When output 1 is turned ON despite the calculation parameters being correct, it is necessary to increase the maximum number of loops per scan. (Refer to *3.5.3 Avoiding Node Lockout.*)

Checks concerning the node lockout are performed in the scan that is executed, and these are not related to the ON/OFF status of inputs 1, 2, and 3. Initialization is also necessary when this problem occurs.

3.5 Priority Processing

This section explains priority processing when using more than one PID2 instruction.

3.5.1 Priority Processing

1) The processing time for the PID circuits may increase the scan time when there are many loops (= number of PID circuits). The following methods can be used to prevent this from occurring.

a) Specify the maximum number of loops to be executed in one scan.

b) Divide loop groups into high-priority groups and low-priority groups.

When doing so, high-priority groups are always executed in the scan in which they are read, but some of the loops in the low-priority group will not be executed.

Execution scans:	Processed by sampling time (Ts) (PID calculation, control, monitoring, priority processing)
Non-execution scans:	Processed between sampling time (Ts) (scaling, parameter monitoring)

2) To perform priority processing, each of the PID circuit calculation parameter tables contains two holding registers (4X+14 and 4X+15 in *Figure 8*).

Register Address	Symbol	Register Contents
4X+14	OFS	Maximum number of loops N executed per pointer scan of the loop counter
4X+15	MAX	

3) The loop counter counts how many loops have been executed in each scan. To execute a) above, one holding register common to all groups is allocated. To execute b) above, one holding register is allocated to the high-priority group and another one to the low-priority group.

The OFS does not act as a counter, but stores the pointer value of the loop counter (the value indicates which number holding register it is using with 400000 as a base).

For example, suppose 400123 was allocated as the loop counter for a), then the pointer value F = 123 would be stored as the OFS for all PID circuits.

Note As a loop counter, as long as there is no duplication with other counters, it is possible to allocate any holding register. MAX stores the maximum number of loops N that can be executed in the execution scan.

The maximum number of loops executed in MAX indicates the number of loops in the execution scan. Loops in the non-execution scan are not included in this number. Loops in the non-execution scan and loops that were not executed in the execution scan are processed in the non-execution scan.

If the priority scan operation is not being used, input O for the MAX and OFS of all loops.

3.5.2 Priority Processing Example

An example of using the priority processing operation is shown below.

◀**EXAMPLE**▶ Suppose the number of loops is 20, and the loops (PID circuits) are L_1, L_2....L_{20}, The corresponding OFS and MAX are OFS_1, OFS_2,...., OFS_{20}, and MAX_1, MAX_2,....MAX_{20}, and their respective contents are F_1, F_2,...., F_{20}, N_1, N_2,....N_{20}. (Refer to *Figure 3.8.*)

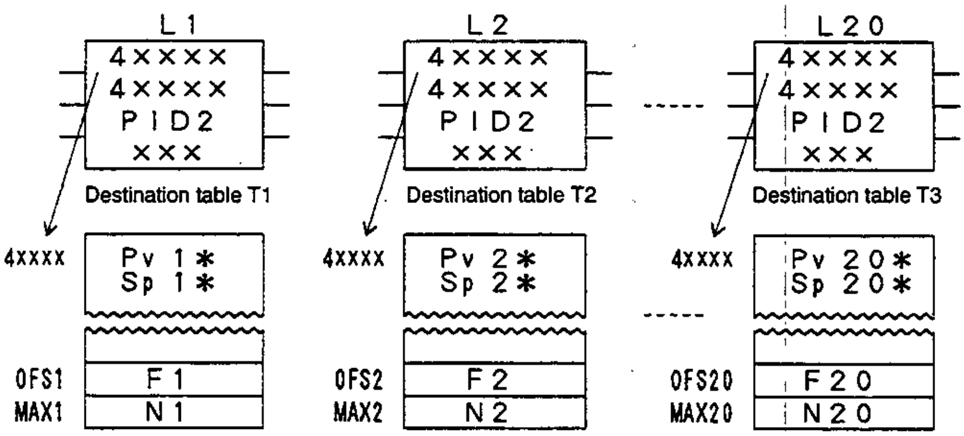


Figure 3.8 Priority Processing

1) **No Priority Processing**

Preset the following:

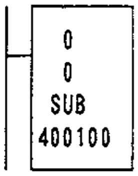
$$F_1 = F_2 = \cdots\cdots = F_{20} = 0$$
$$N_1 = N_2 = \cdots\cdots = N_{20} = 0$$

Although it depends on the values of the sampling cycles of each loop, 20 loops may be executed in the same scan in some cases. Node lockout will not be generated in any case.

2) **Maximum Loops per Scan: 7**

a) Decide on a holding register that will act as the common loop counter for 20 loops. This will be 400100 in this example.

b) Because the offset is 100 (400000 + 100 = 400100), store the following values:
$F_1 = F_2 = ... = F_{20} = 100$



c) Store the following values $N_1 = N_2 = ... = N_{20} = 7$.

d) In a network before the network containing L_1, L_2, ... L_{20}, create a circuit that clears the contents of 400100 at each scan.

Each time a loop is executed in a particular scan, the contents of 400100 is incremented. When 400100 is 7, that is, after the first 7 groups in the execution cycle (this is not necessarily L_1 to L_7) are executed, the following groups (loops with lower network solving order) are not executed even if they are in the execution cycle. Instead they are processed in the non-execution scan.

The loop counter is reset with each scan (by the SUB instruction mentioned earlier), so the maximum of 7 loops is executed in each scan. Supposing the scan time and sampling cycle were the same, and all sampling cycles were the same, only the 7 loops at the top of the solving order are executed, and the remaining 13 loops are always subject to a node lockout.

3) **Always Executing Four Specified Loops between L_1 and L_{20} (e.g., L_2, L_5, L_{11}, and L_{16}) When in the Execution Cycle (Maximum Loops per Scan = 7)**

a) Suppose the loop counter for the high-priority group (L_2, L_5, L_{11}, and L_{16}) is 400101, and the counter for the remaining 16 loops is 400102.

b) $F_2 = F_5 = F_{11} = F_{16} = 101$
The remaining F values are set to 102.

c) $N_2 = N_5 = N_{11} = N_{16} = 4$
The remaining N values are set to 3 (7 – 4 = 3).

d) Create a circuit which clears the contents of 400101 and 400102 with each scan in a network before the network in which L_1, L_2, ... L_{20} are stored.

If in a particular scan, the high-priority group of loops is in the execution cycle, the loops are executed, and three (maximum) of the remaining execution scan loops from the 16 low-priority groups are executed.

Depending on the way in which sampling cycles are specified some specific loops may always be left unexecuted as shown in 2) above.

3.5.3 Avoiding Node Lockout

1) The following measures are available to prevent node lockout from occurring.

a) Use priority processing for loops with short sampling cycles.

For example, if the scan time is about 100 ms, the sampling cycle of loop Ln is 100 ms, and the sampling cycle of loop Lm is 1 s, loop Ln may not be executed in each scan, and node lockouts will occur. Even if loop = Lm is not executed in the execution scan, a node lockout will not occur as long as it is executed within the following 9 scans.

b) Within the restrictions of the maximum number of loops per scan, the sampling cycles for each loop should be kept different as much as possible.

Appendix A

Confirming PID Operation

A.1 Confirming PID Operation

1) The methods for confirming PID calculations on a PLC not connected to the control process are described below.

2) It is not absolutely necessary to perform these tests in the actual processing, but we recommend that they be performed in the following cases.

   - To study and learn about the PLC's PID control or to learn about how the PID constants work.

   - When actual testing does not go well, to check whether the PID system is operating correctly.

3) The confirmation tests are open loop tests. The block diagram is shown in *Figure A.1* below.

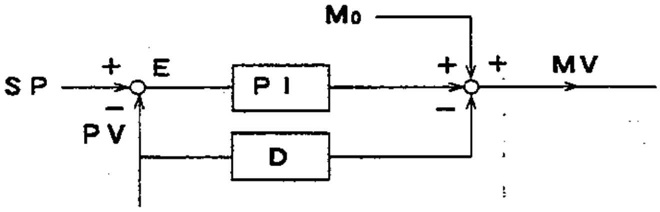


Figure A.1 PID Instruction Open Loop

Set the process variable to 0, and use the test to see how the manipulated variable (MV) changes when PID constants K_1, K_2, and K_3 are changed in relation to the set point SP.

4) Parameters not related to this test are fixed to those shown below.

$CH = RH = SH = 4095$
$CL = RL = SL = 0$
$F = N = 0$

The sampling cycle is set to 0.1 s.

A.2 P Operation

◀EXAMPLE▶

1) With $K_2 = K_3 = 0$
and Bias Mo = 2047

Change the SP (contents in the second register of the source table) in steps 0 → SP → 0 as shown in *Figure A.2.*

In this case,

$$MV = \frac{100}{PB} E + M_o$$
$$= \frac{100}{PB} SP + M_o$$

Example: Supposing $SP = 100$ and $PB = 50$

$$MV = \frac{100}{50} \times 100 + 2047 = \ \ldots\ldots. \text{Direct}$$
$$MV = -\frac{100}{50} \times 100 + 2047 = \ldots\ldots. \text{Reverse}$$

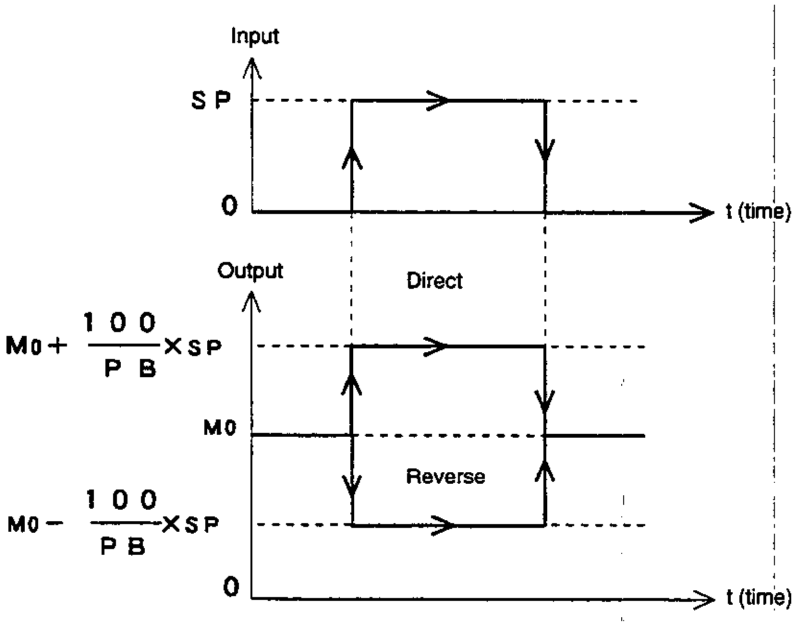


Figure A.2 P Control and Direct/Reverse Operation

A.3 PI Operation

◀EXAMPLE▶ 1) Set $K_3 = 0$ and $Mo = 0$, and change SP in steps. (Set to direct.)

At this time,

$$MV = \frac{100}{PB}\left\{E + \frac{K_2}{60}\Sigma E \cdot Ts\right\}$$

When $t < t_1$ $E = SP = 0$ $MV = 0$

When $t_1 \leq t$ $E = SP = 100$

$$MV = \frac{100}{PB}\left\{SP + \frac{K_2}{60} \times SP(t - t_1)\right\}$$

Example 1: $PB = 100$, $K_2 = 1.0$ ($Ti = 60$), $SP = 100$

When $t < t_1$ $E = SP = 0$ $MV = 0$

When $t_1 \leq t$ $E = SP = 100$

$$\begin{aligned} MV &= \frac{100}{100}\left\{100 + \frac{1.0}{60} \times 100(t - t_1)\right\} \\ &= 100 + \frac{100}{60}(t - t_1) \end{aligned}$$

Example 2: $PB = 50$, $K_2 = 2.0$ ($Ti = 30$), $SP = 100$

When $t < t_1$ $E = SP = 0$ $MV = 0$

When $t_1 \leq t$ $E = SP = 100$

$$\begin{aligned} MV &= \frac{100}{50}\left\{100 + \frac{2.0}{60} \times 100(t - t_1)\right\} \\ &= 200 + \frac{400}{60}(t - t_1) \end{aligned}$$

The two examples above are illustrated in *Figure A.3.*

Note The results are the same even if sampling cycle Ts is changed.

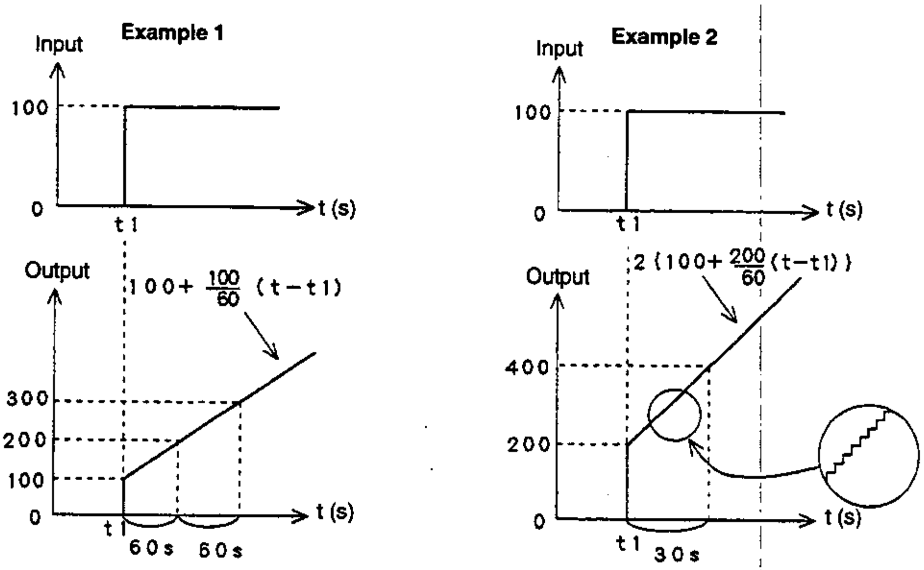


Figure A.3 PD Control Response Example

Changes by Integration

The changes resulting from integral control and the changes resulting from proportional control in the reset time constant (Ti) are equal. (Definition of reset time constant)

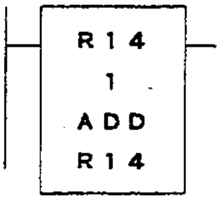
K_2 (repeats/min) indicates how many times per minute in integral control a value the same as the change resulting from proportional control is repeated.

A.4 PD Operation

◀EXAMPLE▶

1) Differentiation is carried out on the process variable, so an operation test in which PV = 0 is impossible. With this in mind, set SP = 0, and move the PV towards the SV.

   It is difficult to gain an understanding of the characteristics of differentiation when the values have been changed in steps, so it is set to a variable (lamp function) that increases proportional to time. For example, if PV = 10t, ladder logic like that in *Figure A.4* can be created.



If R14 sets the scan time of the 14th register of the source table to 100 ms,

PV = 10t

Figure A.4 Lamp Input Example

2) Suppose $K_2 = 0$, $SP = 0$, $M_0 = 0$, and $PV = 10t$. (In reverse operation) then, E = −10t

$$
\begin{aligned}
MV &= -\frac{100}{PB}\left\{En - 60K_3(E_n - E_{n-1}) \times \frac{1}{T_s}\right\} \\
&= \frac{100}{PB}\{10t + 60K_3 \times 10\}
\end{aligned}
$$

Example 1: $PB = 100$, $K_3 = 1.00$ ($Td = 60$ *seconds*)

$$MV = \frac{100}{100}\{10t + 60 \times 1.0 \times 10\} = 10t + 600$$

Example 2: $PB = 50$, $K_3 = 0.50$ ($Td = 30$ *seconds*)

$$MV = \frac{100}{50}\{10t + 60 \times 0.5 \times 10\} = 20t + 600$$

The two examples above are illustrated in *Figure A.5.*

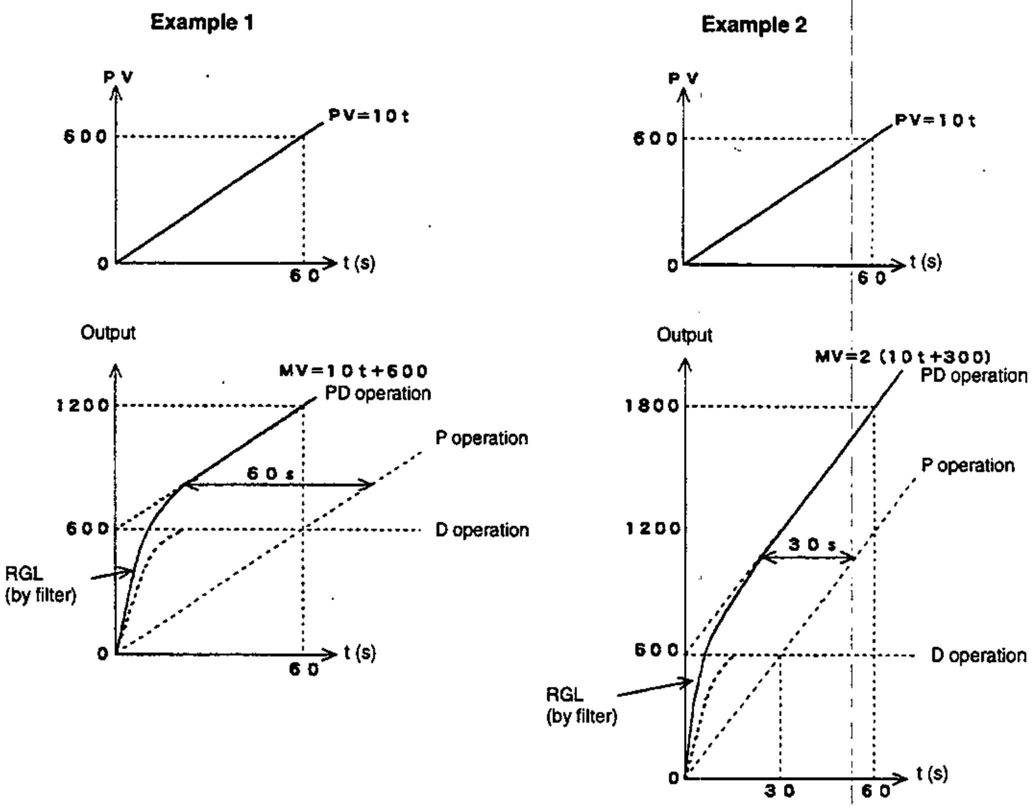


Figure A.5 PD Operation Response Example

3) The manipulated variable MV changes by the amount of the rate time constant (Td). The results are the same even if the sampling cycle is changed.

INDEX

MEMOCON GL120, GL130
SOFTWARE USER'S MANUAL
VOL.4

TOKYO OFFICE
New Pier Takeshiba South Tower, 1-16-1, Kaigan, Minatoku, Tokyo 105-6891 Japan
Phone 81-3-5402-4511 Fax 81-3-5402-4580

YASKAWA ELECTRIC AMERICA, INC.
2121 Norman Drive South, Waukegan, IL 60085, U.S.A.
Phone 1-847-887-7000 Fax 1-847-887-7370

MOTOMAN INC. HEADQUARTERS
805 Liberty Lane West Carrollton, OH 45449, U.S.A.
Phone 1-937-847-6200 Fax 1-937-847-6277

YASKAWA ELÉTRICO DO BRASIL COMÉRCIO LTDA.
Avenida Fagundes Filho, 620 Bairro Saude-Sao Pāulo-SP, Brazil CEP: 04304-000
Phone 55-11-5071-2552 Fax 55-11-5581-8795

YASKAWA ELECTRIC EUROPE GmbH
Am Kronberger Hang 2, 65824 Schwalbach, Germany
Phone 49-6196-569-300 Fax 49-6196-888-301

Motoman Robotics Europe AB
Box 504 S38525 Torsås, Sweden
Phone 46-486-48800 Fax 46-486-41410

Motoman Robotec GmbH
Kammerfeldstraße 1, 85391 Allershausen, Germany
Phone 49-8166-900 Fax 49-8166-9039

YASKAWA ELECTRIC UK LTD.
1 Hunt Hill Orchardton Woods Cumbernauld, G68 9LF, United Kingdom
Phone 44-1236-735000 Fax 44-1236-458182

YASKAWA ELECTRIC KOREA CORPORATION
Kfpa Bldg #1201, 35-4 Youido-dong, Yeongdungpo-Ku, Seoul 150-010, Korea
Phone 82-2-784-7844 Fax 82-2-784-8495

YASKAWA ELECTRIC (SINGAPORE) PTE. LTD.
151 Lorong Chuan, #04-01, New Tech Park Singapore 556741, Singapore
Phone 65-282-3003 Fax 65-289-3003

YASKAWA ELECTRIC (SHANGHAI) CO., LTD.
4F No.18 Aona Road, Waigaoqiao Free Trade Zone, Pudong New Area, Shanghai 200131, China
Phone 86-21-5866-3470 Fax 86-21-5866-3869

YATEC ENGINEERING CORPORATION
Shen Hsiang Tang Sung Chiang Building 10F 146 Sung Chiang Road, Taipei, Taiwan
Phone 886-2-2563-0010 Fax 886-2-2567-4677

YASKAWA ELECTRIC (HK) COMPANY LIMITED
Rm. 2909-10, Hong Kong Plaza, 186-191 Connaught Road West, Hong Kong
Phone 852-2803-2385 Fax 852-2547-5773

BEIJING OFFICE
Room No. 301 Office Building of Beijing International Club, 21
Jianguomenwai Avenue, Beijing 100020, China
Phone 86-10-6532-1850 Fax 86-10-6532-1851

TAIPEI OFFICE
Shen Hsiang Tang Sung Chiang Building 10F 146 Sung Chiang Road, Taipei, Taiwan
Phone 886-2-2563-0010 Fax 886-2-2567-4677

SHANGHAI YASKAWA-TONGJI M & E CO., LTD.
27 Hui He Road Shanghai China 200437
Phone 86-21-6531-4242 Fax 86-21-6553-6060

BEIJING YASKAWA BEIKE AUTOMATION ENGINEERING CO., LTD.
30 Xue Yuan Road, Haidian, Beijing P.R. China Post Code: 100083
Phone 86-10-6233-2782 Fax 86-10-6232-1536

SHOUGANG MOTOMAN ROBOT CO., LTD.
7, Yongchang-North Street, Beijing Economic Technological Investment & Development Area,
Beijing 100076, P.R. China
Phone 86-10-6788-0551 Fax 86-10-6788-2878

YASKAWA ELECTRIC CORPORATION

YASKAWA

MANUAL NO. SIEZ-C825-20.14

99-7③
95-81192, 95-C82-031