11.2.3 OB251 PID Control Algorithm

The operating systems of the central processing units have an integral PID control algorithm which you can use for your own purposes with the help of organization block OB251.
Before calling OB251, a data block (PID controller DB) containing the controller parameters and other controller-specific data must be opened. The PID control algorithm is called periodically (sampling interval) and generates the manipulated variable. The more closely the sampling interval is observed, the more accurately can the controller fulfill its appointed task. The control parameters specified in the controller DB must be matched to the sampling interval. Typically, timed interrupts are serviced by a time block (OB10 to OB13).
Timed-interrupt OBs can be called at intervals between 10 msec. and 10 minutes. The maximum execution time of the PID control algorithm is 2 msec.

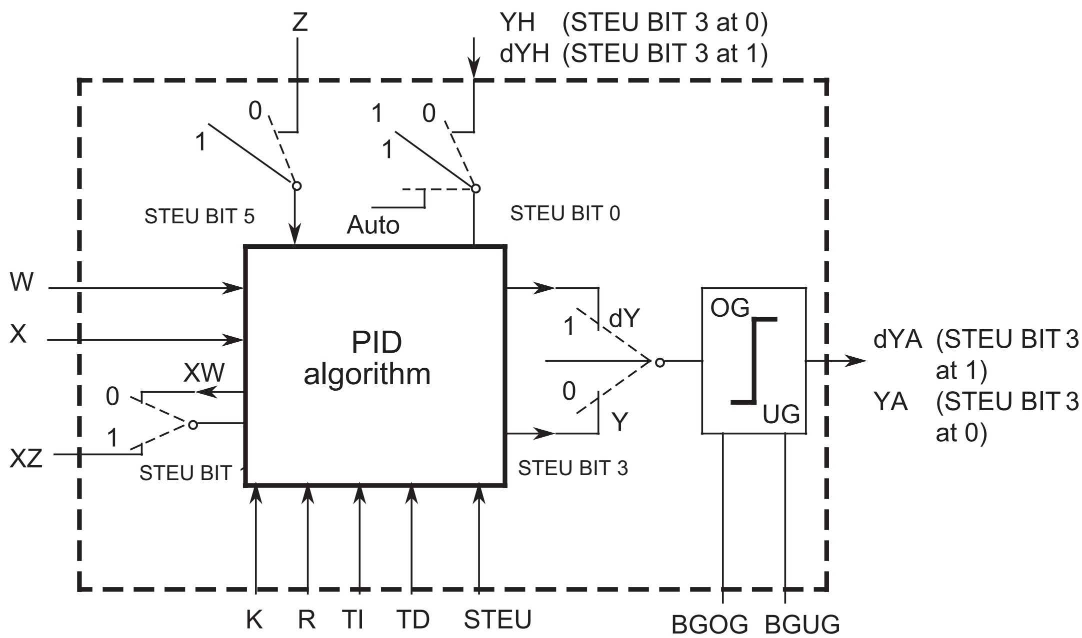

Figure 11-3. Block Diagram of the PID Controller

Legend:

K	=	Proportional coefficient K>0 positive control direction K<0 negative control direction	Y dY YH	= = =	Manipulated variable Correcting increment Value for manual input
R	=	R parameter (usually 1000)	dYH	=	Correcting increment for manual input
TA	=	Sampling interval	BGOG	=	Upper limiting value
TN	=	Integral-action time	BGUG	=	Lower limiting value
TV	=	Derivative-action time	X	=	Actual value
TI	=	TA/TN	Z	=	Disturbance variable
TD	=	TV/TA	XZ	=	Substitute variable for control deviation
STEU	=	Control word	YA	=	Controller output: manipulated variable limited
W	=	Setpoint	dYA	=	Controller output: correcting increment limited
XW	=	Controller difference			

The continuous action controller is designed for controlled systems such as those used in pressure, temperature, or flow rate control.

The "R" parameter sets the proportional component of the PID controller.
If proportional action is required, most controller designs use the value R=1.

The individual proportional-action, integral-action, and derivative-action components can be deactivated via their parameters (R, TI, and TD) by presetting the pertinent data words with zero. This enables you to implement all required controller structures without difficulty, e.g., PI, PD, or PID controllers.

You can forward the system deviation XW or, using the XZ input, any disturbance variable or the inverted actual value X to the differentiator. Specify a negative K value for an inverted control direction.
When the correction information (dY or Y) is at a limit, the integral-action component is automatically deactivated in order not to impair the dynamic response of the controller.

The switch settings in the block diagram are implemented by setting the associated bits in control word STEU when the D controller is initialized.

Table 11-10. Description of the Control Bits in Control Word STEU

Control Bit	Name	Signal State	Description
0	AUTO	0 1	Manual mode The following variables are updated in Manual mode: 1) X_K, XW_{K-1} and PW_{K-1} 2) XZ_K, XZ_{K-1} and PZ_{K-1}, when STEU bit 1=1 3) Z_K and Z_{K-1}, when STEU bit 5=0 Variable dD_{K-1} is set to 0. The algorithm is not computed. Automatic mode
1	XZ EIN	0 1	XW_k is forwarded to the differentiator. The XZ input is ignored. A variable other than XW_k is forwarded to the differentiator.
2	REG AUS	0 1	Normal controller processing When the controller is invoked (OB251), all variables (DW 18 to DW 48) with the exception of K, R, TI, TD, BGOG, BGUG, YH_k and W_k are reset in the controller DB. The controller is deactivated.
3	GESCHW	0 1	Correction algorithm Velocity algorithm
4	HANDART	0 1	When GESCHW=0: Following the transfer to Manual mode, the specified manipulated variable value YA is adjusted exponentially to the manual value in four sampling steps. Additional manual values are then forwarded immediately to the controller output. When GESCHW=1: The manual values are forwarded immediatley to the controller output. The limiting values are in force in Manual mode. When GESCHW=0: The manipulated variable last output is retained. When GESCHW=1: Correction increment dY_K is set to zero.
5	NO Z	0 1	With feedforward control No feedforward control
6 to 15	-		The PID algorithm uses these bits as auxiliary flags.

The control program can be supplied with fixed values or parameters. Parameters are input via the assigned data words. The controller is based on a PID algorithm. Its output signal can be either a manipulated variable (correction algorithm) or a manipulated variable modification (correction rate algorithm).

Correction Rate Algorithm

The relevant correction increment dY_k is computed at instant $t = k \cdot TA$ according to the following formula:

- Without feedforward control (D11.5=1); XW is forwarded to the differentiator (D11.1=0)

$$dY_k = K[(XW_k - XW_{k-1}) R + TI \cdot XW_k + (TD (XW_k - 2XW_{k-1} + XW_{k-2}) + dD_{k-1})]$$
$$= K (dPW_k + dI_k + dD_k)$$

- With feedforward control (D11.5=0); XW is forwarded to the differentiator (D11.1=0)

$$dY_k = K[(XW_k - XW_{k-1}) R + TI \cdot XW_k + (TD (XW_k - 2XW_{k-1} + XW_{k-2}) + dD_{k-1})] + (Z_k - Z_{k-1})$$
$$= K (dPW_k + dI_k + dD_k) + dZ_k$$

- Without feedforward control (D11.5=1); XZ is forwarded to the differentiator (D11.1=1)

$$dY_k = K[(XW_k - XW_{k-1}) R + TI \cdot XW_k + (TD (XZ_k - 2XZ_{k-1} + XZ_{k-2}) + dD_{k-1})]$$
$$= K (dPW_k + dI_k + dD_k)$$

- With feedforward control (D11.5=0); XZ is forwarded to the differentiator (D11.1=1)

$$dY_k = K[(XW_k - XW_{k-1}) R + TI \cdot XW_k + (TD (XZ_k - 2XZ_{k-1} + XZ_{k-2}) + dD_{k-1})] + (Z_k - Z_{k-1})$$
$$= K (dPW_k + dI_k + dD_k) + dZ_k$$

P component   I component   D component   Z component   k: k^{th} sample

When XW_k is applied:	XW_k	=	$W_k - X_k$
	PW_k	=	$XW_k - XW_{k-1}$
	QW_k	=	$PW_k - PW_{k-1}$
		=	$XW_k - 2XW_{k-1} + XW_{k-2}$
When XZ is applied:	PZ_k	=	$XZ_k - XZ_{k-1}$
	QZ_k	=	$PZ_k - PZ_{k-1}$
		=	$XZ_k - 2XZ_{k-1} + XZ_{k-2}$
The result is:	dPW_k	=	$(XW_k - XW_{k-1})R$
	dI_k	=	$TI \cdot XW_k$
	dD_k	=	$(TD \cdot QW_k + dD_{k-1})$ when XW is applied
		=	$(TD \cdot QZ_k + dD_{k-1})$ when XZ is applied
	dZ_k	=	$Z_k - Z_{k-1}$

Correction Algorithm

The formula used to compute the correction rate algorithm is also used to compute the correction algorithm.
In contrast to the correction rate algorithm, however, the sum of all correction increments computed (in DW 48), rather than the correction increment dY_k is output at sampling instant t_k.

At instant t_k , manipulated variable Y_k is computed as follows:

$$Y_k = \sum_{m=0}^{m=k} dY_m$$

Initializing the PID Algorithm

OB251's interface to its environment is the controller DB.
All data needed to compute the next manipulated variable value is stored in this DB. Each controller must have its own controller data block.
The controller-specific data are initialized in a data block that must comprise at least 49 data words.
The CPU goes to STOP with a transfer error (TRAF) if no DB has been opened or if the DB is too short.

Caution

Make sure that the right controller DB has been invoked before calling control algorithm OB251.

Table 11-11. Format of the Controller DB

Data Word	Name	Comments
1	K	Proportional coefficient (- 32 768 to+32 767) for controllers without D component Proportional coefficient (- 1500 to+1500) for controllers with D component[1] K is greater than zero when the control direction is positive, and less than zero when the control direction is negative; the specified value is multiplied with the factor 0.001.
3	R	R parameter (- 32 768 to+32 767) for controllers without D component R parameter (-1500 to+1500) for controllers with D component[1] Normally 1 for controllers with P component: the specified value is multiplied with the factor 0.001.
5	TI	Constant TI (0 to 9999) $TI = \frac{\text{Sampling interval TA}}{\text{Integral-action time TN}}$ The specified value is multiplied with a factor of 0.001.
7	TD	Constant TD (0 to 999) $TD = \frac{\text{Derivative-action time TV}}{\text{Sampling interval TA}}$
9	W	Setpoint (- 2047 to +2047)
11	STEU	Control word (bit pattern)
12	YH	Value for manual operation (- 2047 to +2047)
14	BGOG	Upper limiting value (- 2047 to +2047)
16	BGUG	Lower limiting value (- 2047 to +2047)

1 Larger gains are possible if abrupt changes to the system deviation are sufficiently small. Large changes of the system deviation should therefore be divided up into small changes; e.g. by feeding the setpoint via a ramp function.
The factor 0.001 is an approximate value. The precise value for the factor is 1/1024 or 0.000976.

Table 11-11. Format of the Transfer Block (Continued)

Data Word	Name	Comments
22	X	Actual value (- 2047 to+2047)
24	Z	Disturbance variable (- 2047 to+2047)
29	XZ	Derivative time (- 2047 to+2047)
48	YA	Output variable (- 2047 to+2047)

All parameters (with the exception of the control word STEU) must be specified as 16-bit fixed point numbers.

Note

The PID algorithm uses the data words that are not listed in the table as auxiliary flags.

Initializing and Invoking the PID Controller in the STEP 5 Program

A number of different PID controllers can be implemented by calling OB251 repeatedly, so make sure that the relevant controller DB has been invoked before calling OB251.

Note

Important controller data are stored in the high-order byte of control word DW 11 (DL 11). Make sure that only T DR 11/SU D 11.0 to D 11.7 or RU D 11.0 to D 11.7 operations are used to modify user-specific bits in the control word.

Selecting the Sampling Interval

The value selected as sampling interval must not be excessiveley high in order to be able to use the well-known analog method in the case of digital control loops.
Experience has shown that a sampling interval of approximately 1/10 of the counter constant T_{RKdom}* produces a control result comparable to the equivalent analog result. Dominant system time constantT_{RKdom} determines the step response of the closed control loop.

$$T_A = 1/10 \cdot T_{RKdom}$$

In order to ensure the constancy of the sampling interval, OB251 must always be invoked in the service routine for timed interrupts (OB13).

* T_{RKdom}=Dominant system time constant of the closed control loop

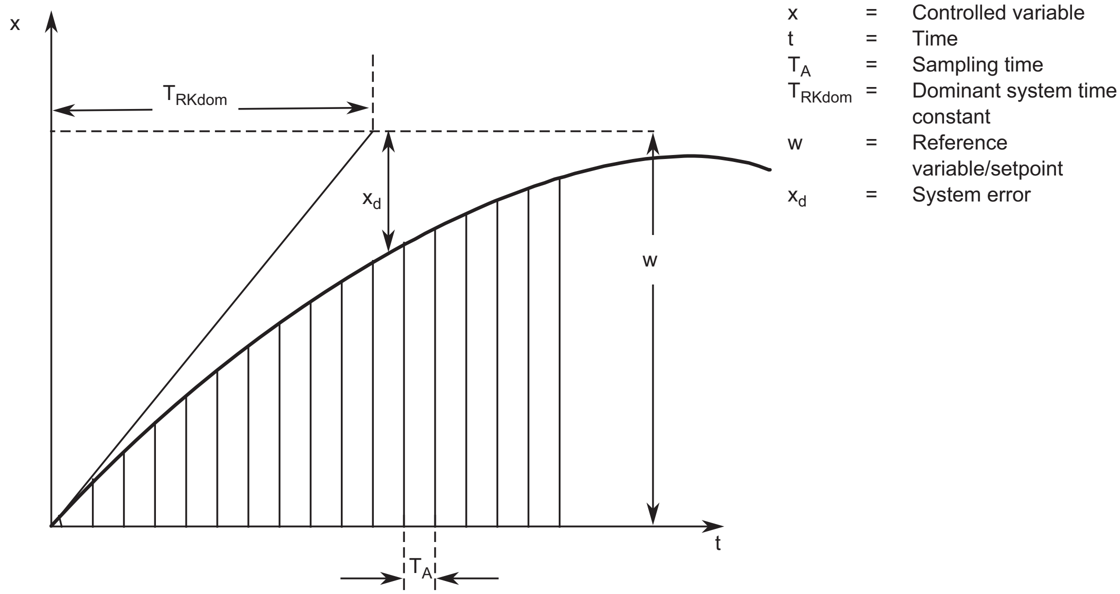

Figure 11-4. Estimating the Dominant System Time Constant (T_{RKdom})