5-3.2 Types of Feedback Controllers

The way feedback controllers make a decision is by solving an equation based on the difference between the controlled variable and the set point. In this section, we examine the most common types of controllers by looking at the equations that describe their operation.

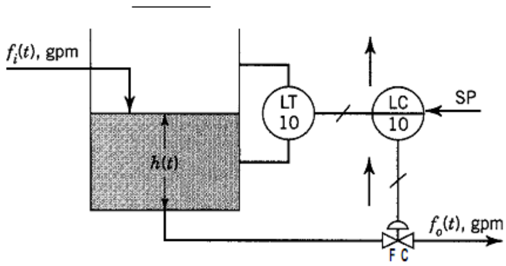

Figure 53.3 Liquid level control loop.

As we saw in Chapter 1, the signals entering and exiting the controllers are either electrical or pneumatic. Even in computer systems, the signals entering from the field are electrical before they are converted, by an analog-to-digital (A/D) converter, to digital signals. Likewise, the signal the computer system sends back to the field is an electrical signal. To help simplify the presentation that follows, we will use all signals in percent. That is, we will speak of 0 to 100% rather than 4 to 20 mA, 3 to 15 psig, or any other type of signal.

As we have said, feedback controllers decide what to do to maintain the controlled variable at set point by solving an equation based on the difference between the set point and the controlled variable. This difference, or error, is computed as

$$e(t) = r(t) - c(t)$$
$$E(t) = R(t) - C(t)$$
$$E(s) = R(s) - C(s)$$

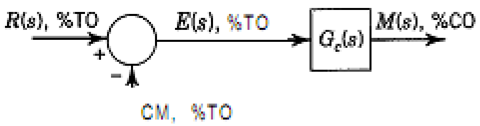

Figure 53.4 Block diagram representation of controller.

Proportional Controller (P)

$$m(t) = \overline{m} + K_c e(t)$$

m(t) = controller output, %CO. The term m(t) is used to stress that as far as the controller is concerned, this output is the manipulated variable.

K_c = controller gain, $\frac{\%CO}{\%TO}$

$\overline{m}$ = bias value, %CO. This is the output from the controller when the error is zero. The value $\overline{m}$ is a constant and is also the output when the controller is switched to manual. It is very often initially set at mid-scale, 50 %CO.

$$\boxed{C_P(s) = \frac{M(s)}{E(s)} = K_c}$$

Proportional-Integral Controller (PI)

$$m(t) = \overline{m} + K_c e(t) + \frac{K_c}{\tau_I}\int e(t)\,dt$$

where τ_I = *integral (or reset) time.*

$$\boxed{C_{PI}(s) = \frac{M(s)}{E(s)} = K_c\left(1 + \frac{1}{\tau_I s}\right) = K_c\left(\frac{\tau_I s + 1}{\tau_I s}\right)}$$

Proportional-Integral-Derivative Controller (PID)

$$m(t) = \overline{m} + K_c e(t) + \frac{K_c}{\tau_I}\int e(t)\,dt + K_c\tau_D\frac{de(t)}{dt}$$

$$C_{PID}(s) = \frac{M(s)}{E(s)} = K_c\left(1 + \frac{1}{\tau_I s} + \tau_D s\right)$$

$$\boxed{C_{PID}(s) = \frac{M(s)}{E(s)} = K_c\left(1 + \frac{1}{\tau_I s} + \tau_D s\right)}$$ **IDEAL PID**

PID controllers are recommended for use in slow processes (processes with multiple time constants or dead time) such as temperature loops, which are usually free of noise.
Fast processes (processes with short time constants) are easily susceptible to process noise. Typical of these fast processes are flow loops and liquid pressure loops. Consider the recording of a flow shown in Fig. 5-3.11.

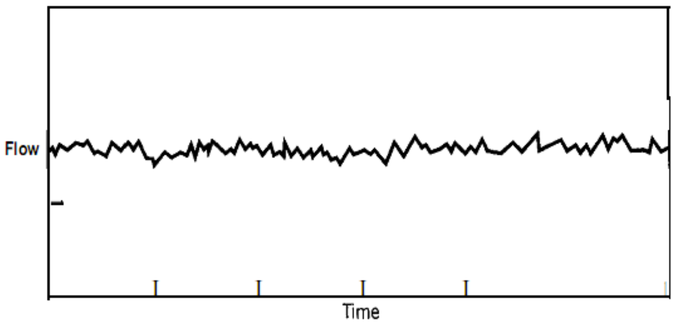

Figure 53.11 Recording of flow.

The application of the derivative mode will only result in the amplification of the noise, because the derivative of the fast changing noise is a large value. Processes with long time constants are usually damped and, consequently, are less susceptible to noise. In the case of a slow process with a noisy transmitter, however, the transmitter must be fixed or the noise filtered before the Actually, when the PID controller is implemented with Eq. 53.16, it does not work very well. To improve the performance of the derivative mode, the algorithm is slightly changed to:

$$C_{PID}(s) = \frac{M(s)}{E(s)} = K_c\left(1 + \frac{1}{\tau_I s} + \left[\underbrace{\frac{1}{(\alpha\tau_D s + 1)}}_{\substack{low-pass \\ filter}}\right]\tau_D s\right)$$

This new term does not affect the performance of the controller because $\alpha\tau_D$ is small.

$$C_{PID}(s) = \frac{M(s)}{E(s)} = K_c\left(\left[\underbrace{\frac{(\alpha + 1)\tau_D s + 1}{(\alpha\tau_D s + 1)}}_{\substack{lead-lag \\ unit}}\right] + \frac{1}{\tau_I s}\right)$$

This transfer function shows the PID controller as a lead/lag unit in parallel with an integration. The net lead of the lead/lag unit is the derivative time.
In analog controllers and many computer-based controllers, the describing transfer function for the PID controllers used is:

$$C_{PID}(s) = \frac{M(s)}{E(s)} = K_c^{prima}\left(\frac{(\alpha + 1)\tau^{prima}{}_D s + 1}{(\alpha\tau^{prima}{}_D s + 1)} + \frac{1}{\tau^{prima}{}_I s}\right)$$ **Real PID, in practice.**

“series PID”
“rate-before-reset”

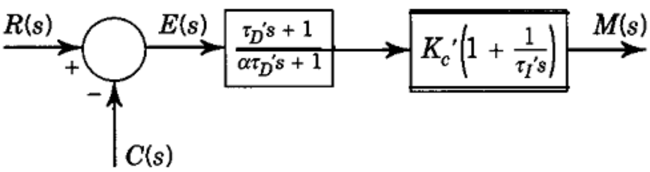

Figure 53.12 Block diagram of PID controller—
Eq. 5-3.19.

In Eq. 5-3.19 the prime notation has been used to indicate that the tuning parameters are not the same as those in Eq. 5-3.16 or Eq. 5-3.17. Using algebraic manipulations with Eq. 5-3.16 and 5-3.19, the following relations can be obtained:

$$K_C^{prima} = K_C\left(0.5 + \sqrt{0.25 - \frac{\tau_D}{\tau_I}}\right)$$

$$\tau_I^{prima} = \tau_I\left(0.5 + \sqrt{0.25 - \frac{\tau_D}{\tau_I}}\right)$$

$$\tau_D^{prima} = \frac{\tau_D}{0.5 + \sqrt{0.25 - \frac{\tau_D}{\tau_I}}}$$

Chapter 6 shows how to obtain the tuning parameters.
The controller described by Eq. 5-3.16 is sometimes referred to as an ideal PID, whereas the controller described by Eq. 5-3.19 is referred to as an actual PID.

To summarize, PID controllers have three tuning parameters: **the gain or proportional band**, **the reset time or reset rate**, and **the rate time**. PID controllers are recommended for processes that are **free of noise**. The advantage of the derivative mode is that it provides anticipation.

Proportional-Derivative Controller (PD)

This controller is used in processes where a proportional controller can be used, where steady-state offset is acceptable, but where some amount of anticipation is desired and no noise is present. The describing equation is:

$$m(t) = \overline{m} + K_c e(t) + K_c \tau_D \frac{de(t)}{dt}$$

and the "ideal" transfer function is

$$\boxed{C_{PD}(s) = \frac{M(s)}{E(s)} = K_c \left(1 + \tau_D s\right)}$$

whereas the "actual," or implemented, transfer function is:

$$\boxed{C_{PD} = \frac{M(s)}{E(s)} = K_C \left[\frac{(1+\alpha)\tau_D s + 1}{\alpha \tau_D s + 1}\right]}$$

5-3.3 Modifications to the PID Controller and Additional Comments

5-3.4 Reset Windup and Its Prevention