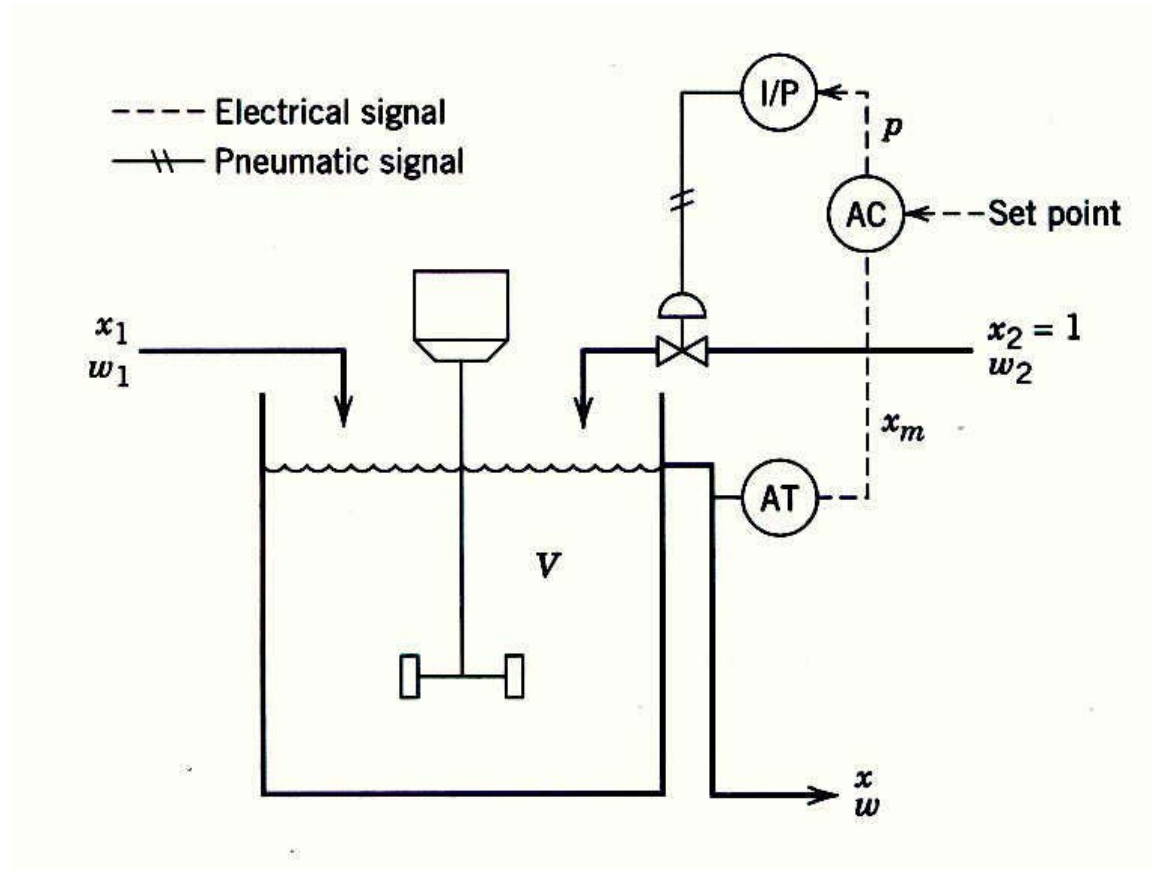# Feedback Controllers

Figure 8.1 Schematic diagram for a stirred-tank blending system.

Basic Control Modes

Next we consider the three basic control modes starting with the simplest mode, *proportional control.*

Proportional Control

In feedback control, the objective is to reduce the error signal to zero where

$$e(t) = y_{sp}(t) - y_m(t) \qquad (8\text{-}1)$$

and

$$e(t) \quad = \quad \text{error signal}$$

$$y_{sp}(t) \quad = \quad \text{set point}$$

$$y_m(t) \quad = \quad \text{measured value of the controlled variable}$$

(or equivalent signal from the sensor/transmitter)

Although Eq. 8-1 indicates that the set point can be time-varying, in many process control problems it is kept constant for long periods of time.

For proportional control, the controller output is proportional to the error signal,

$$p(t) = \bar{p} + K_c e(t) \qquad (8\text{-}2)$$

where:

$$p(t) = \text{controller output}$$

$$\bar{p} \quad = \text{bias (steady-state) value}$$

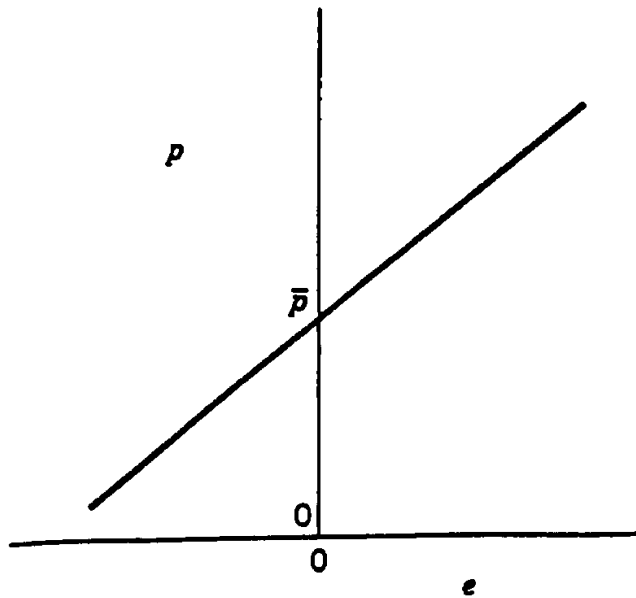$$K_c \quad = \text{controller gain (usually dimensionless)}$$

Figure 8.4. Proportional control: ideal behavior (slope of line = K_c).
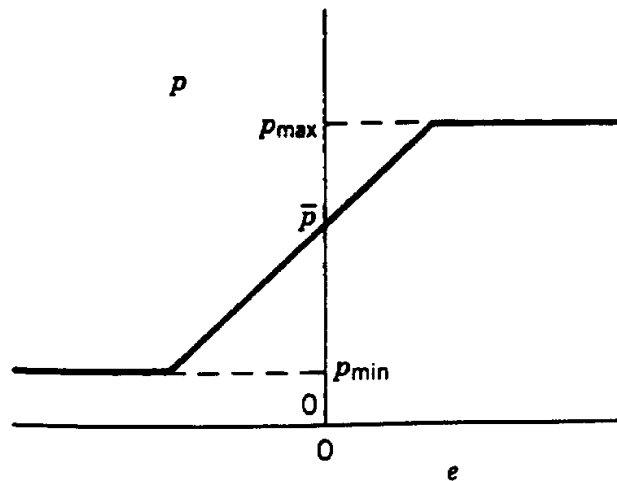

Figure 8.5. Proportional control: actual behavior.

The key concepts behind proportional control are the following:

1. The controller gain can be adjusted to make the controller output changes as sensitive as desired to deviations between set point and controlled variable;

2. the sign of K_c can be chosed to make the controller output increase (or decrease) as the error signal increases.

For proportional controllers, bias $\bar{p}$ can be adjusted, a procedure referred to as *manual reset*.

Some controllers have a proportional band setting instead of a controller gain. The *proportional band PB* (in %) is defined as

$$PB \triangleq \frac{100\%}{K_c} \qquad\qquad (8\text{-}3)$$

In order to derive the transfer function for an ideal proportional controller (without saturation limits), define a deviation variable $p'(t)$ as

$$p'(t) \triangleq p(t) - \bar{p} \qquad (8\text{-}4)$$

Then Eq. 8-2 can be written as

$$p'(t) = K_c e(t) \qquad (8\text{-}5)$$

The transfer function for proportional-only control:

$$\frac{P'(s)}{E(s)} = K_c \qquad (8\text{-}6)$$

An inherent disadvantage of proportional-only control is that a steady-state error occurs after a set-point change or a sustained disturbance.

Integral Control

For integral control action, the controller output depends on the integral of the error signal over time,

$$p(t) = \bar{p} + \frac{1}{\tau_I} \int_0^t e(t^*)\,dt^* \qquad (8\text{-}7)$$

where τ_I, an adjustable parameter referred to as the integral time or reset time, has units of time.

Integral control action is widely used because it provides an important practical advantage, the elimination of offset. Consequently, integral control action is normally used in conjunction with proportional control as the *proportional-integral (PI)* controller:

$$p(t) = \bar{p} + K_c \left(e(t) + \frac{1}{\tau_I} \int_0^t e(t^*)\,dt^* \right) \qquad (8\text{-}8)$$

The corresponding transfer function for the PI controller in Eq. 8-8 is given by

$$\frac{P'(s)}{E(s)} = K_c \left(1 + \frac{1}{\tau_I s} \right) = K_c \left(\frac{\tau_I s + 1}{\tau_I s} \right) \qquad (8\text{-}9)$$

Some commercial controllers are calibrated in terms of $1/\tau_I$ (repeats per minute) rather than τ_I (minutes, or minutes per repeat).

Reset Windup

- An inherent disadvantage of integral control action is a phenomenon known as *reset windup* or *integral windup*.

- Recall that the integral mode causes the controller output to change as long as $e(t^*) \neq 0$ in Eq. 8-8.

- When a sustained error occurs, the integral term becomes quite large and the controller output eventually saturates.

- Further buildup of the integral term while the controller is saturated is referred to as reset windup or *integral windup.*

Derivative Control

The function of derivative control action is to anticipate the future behavior of the error signal by considering its rate of change.

- The anticipatory strategy used by the experienced operator can be incorporated in automatic controllers by making the controller output proportional to the rate of change of the error signal or the controlled variable.

- Thus, for *ideal* derivative action,

$$p(t) = \bar{p} + \tau_D \frac{de(t)}{dt} \qquad (8\text{-}10)$$

where τ_D, the derivative time, has units of time.

For example, an ideal PD controller has the transfer function:

$$\frac{P'(s)}{E(s)} = K_c \left(1 + \tau_D s \right) \qquad (8\text{-}11)$$

- By providing anticipatory control action, the derivative mode tends to stabilize the controlled process.

- Unfortunately, the ideal proportional-derivative control algorithm in Eq. 8-10 is *physically unrealizable* because it cannot be implemented exactly.

- For analog controllers, the transfer function in (8-11) can be approximated by

$$\frac{P'(s)}{E(s)} = K_c \left(1 + \frac{\tau_D s}{\alpha \tau_D s + 1} \right) \qquad (8\text{-}12)$$

  where the constant α typically has a value between 0.05 and 0.2, with 0.1 being a common choice.

- In Eq. 8-12 the derivative term includes a *derivative mode filter* (also called a *derivative filter*) that reduces the sensitivity of the control calculations to high-frequency noise in the measurement.

Proportional-Integral-Derivative (PID) Control

Now we consider the combination of the proportional, integral, and derivative control modes as a PID controller.

- Many variations of PID control are used in practice.

- Next, we consider the three most common forms.

Parallel Form of PID Control

The *parallel form* of the PID control algorithm (without a derivative filter) is given by

$$p(t) = \bar{p} + K_c \left[e(t) + \frac{1}{\tau_I} \int_0^t e(t^*)\, dt^* + \tau_D \frac{de(t)}{dt} \right] \qquad (8\text{-}13)$$

The corresponding transfer function is:

$$\frac{P'(s)}{E(s)} = K_c \left[1 + \frac{1}{\tau_I s} + \tau_D s \right] \qquad (8\text{-}14)$$

Series Form of PID Control

Historically, it was convenient to construct early analog controllers (both electronic and pneumatic) so that a PI element and a PD element operated in series.

Commercial versions of the series-form controller have a derivative filter that is applied to either the derivative term, as in Eq. 8-12, or to the PD term, as in Eq. 8-15:

$$\frac{P'(s)}{E(s)} = K_c \left(\frac{\tau_I s + 1}{\tau_I s} \right) \left(\frac{\tau_D s + 1}{\alpha \tau_D s + 1} \right) \qquad (8\text{-}15)$$

Expanded Form of PID Control

In addition to the well-known series and parallel forms, the *expanded form* of PID control in Eq. 8-16 is sometimes used:

$$p(t) = \bar{p} + K_c e(t) + K_I \int_0^t e(t^*)\, dt^* + K_D \frac{de(t)}{dt} \qquad (8\text{-}16)$$

Features of PID Controllers

Elimination of Derivative and Proportional Kick

- One disadvantage of the previous PID controllers is that a sudden change in set point (and hence the error, *e*) will cause the derivative term momentarily to become very large and thus provide a *derivative kick* to the final control element.

- This sudden change is undesirable and can be avoided by basing the derivative action on the measurement, y_m, rather than on the error signal, e.

- We illustrate the elimination of derivative kick by considering the parallel form of PID control in Eq. 8-13.

- Replacing de/dt by $-dy_m/dt$ gives

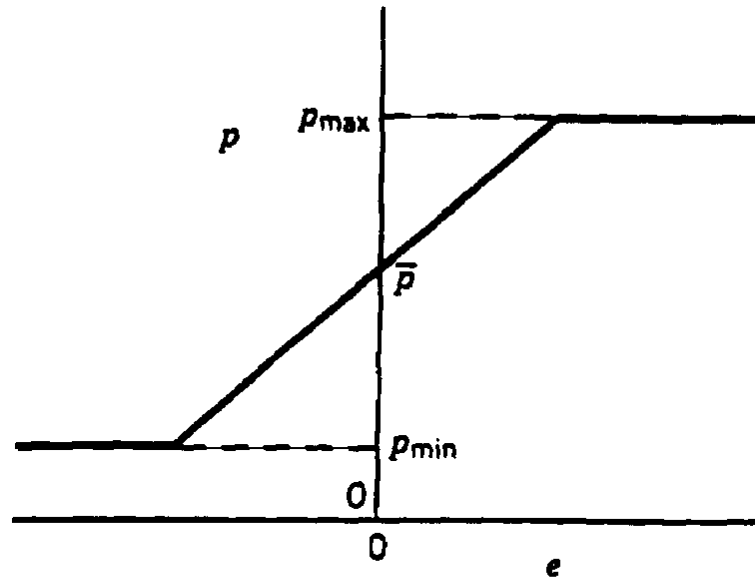$$p(t) = \bar{p} + K_c\left[e(t) + \frac{1}{\tau_I}\int_0^t e(t*)\,dt* - \tau_D\frac{dy_m(t)}{dt}\right] \quad (8\text{-}17)$$

Reverse or Direct Action

- The controller gain can be made either negative or positive.

15

- For proportional control, when $K_c > 0$, the controller output $p(t)$ increases as its input signal $y_m(t)$ decreases, as can be seen by combining Eqs. 8-2 and 8-1:

$$p(t) - \overline{p} = K_c \left[y_{sp}(t) - y_m(t) \right] \qquad \text{(8-22)}$$

- This controller is an example of a *reverse-acting* controller.

- When $K_c < 0$, the controller is said to be *direct acting* because the controller output increases as the input increases.

- Equations 8-2 through 8-16 describe how controllers perform during the *automatic mode* of operation.

- However, in certain situations the plant operator may decide to override the automatic mode and adjust the controller output manually.
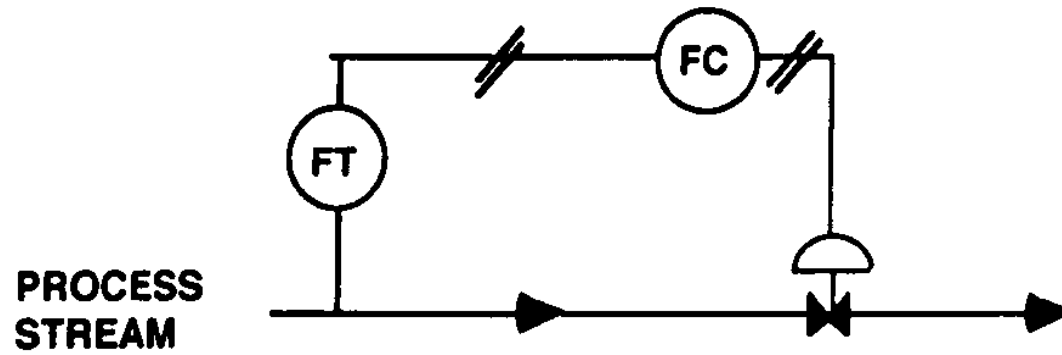
Figure 8.11 Reverse and direct-acting proportional controllers. (a) reverse acting ($K_c > 0$. (b) direct acting ($K_c < 0$)

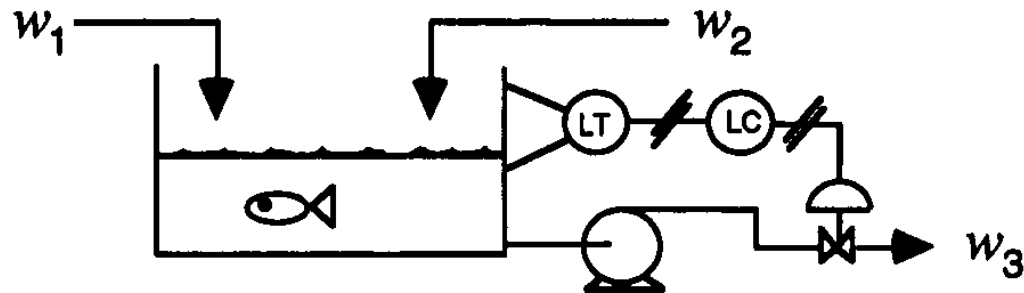- ***Example:*** Flow Control Loop

Assume FT is direct-acting.

1. Air-to-open (fail close) valve ==> ?
2. Air-to-close (fail open) valve ==>   ?
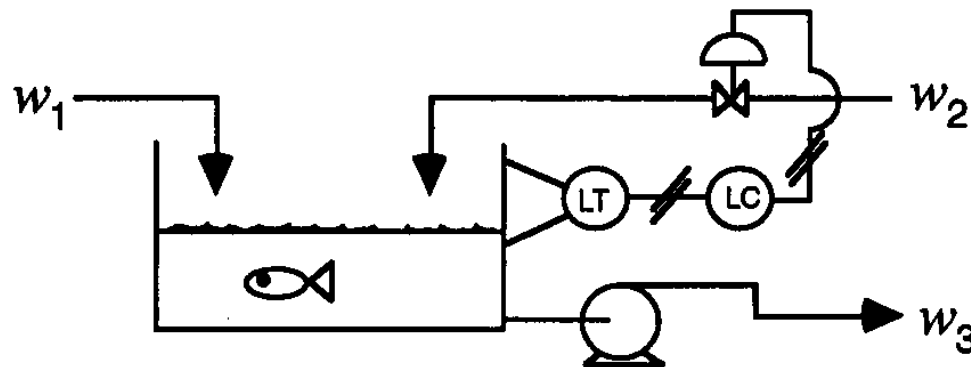
Automatic and Manual Control Modes

- Automatic Mode

   Controller output, p(t), depends on *e(t),* controller constants, and type of controller used.
   (PI *vs.* PID etc.)

- Manual Mode

   Controller output, p(t), is adjusted manually.

- Manual Mode is very useful when unusual

   conditions exist:

      plant start-up

      plant shut-down

      emergencies

- Percentage of controllers "on manual" ??
   (30% in 2001, Honeywell survey)

Example: Liquid Level Control

- Control valves are air-to-open
- Level transmitters are direct acting

System I

System II

Questions: 1. Type of controller action?
2. What type of fish?

On-Off Controllers

- Simple
- Cheap
- Used In residential heating and domestic refrigerators
- Limited use in process control due to continuous cycling of controlled variable $\Rightarrow$ excessive wear on control valve.
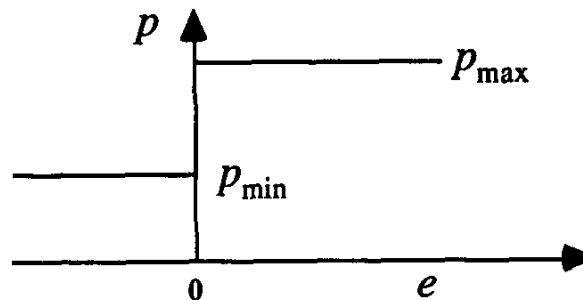
On-Off Controllers (continued)

Synonyms:

"two-position" or "bang-bang" controllers.

$$p(t) = \begin{cases} p_{max} & \text{if} & e > 0 \\ p_{min} & \text{if} & e < 0 \end{cases} \qquad \textbf{ideal case}$$
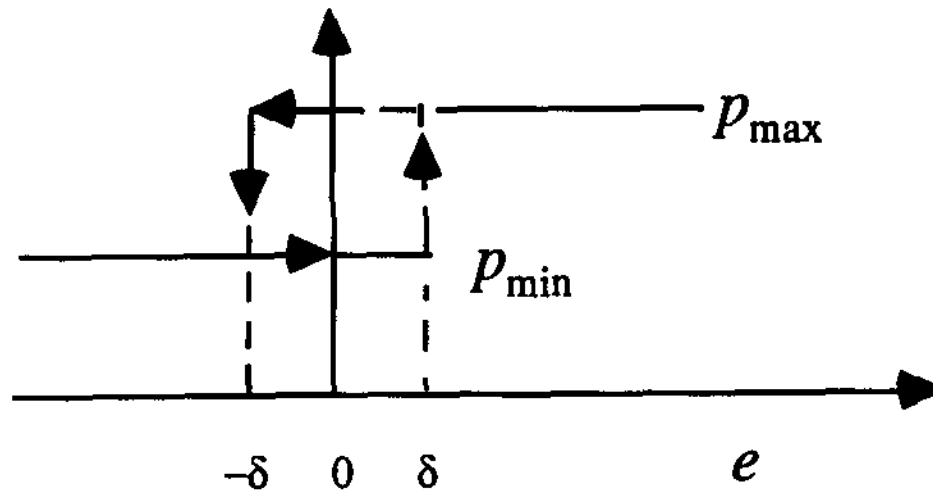
p_{max} **is the "on" value**
p_{min} **is the "off" value**

Controller output has two possible values.

Practical case (dead band)

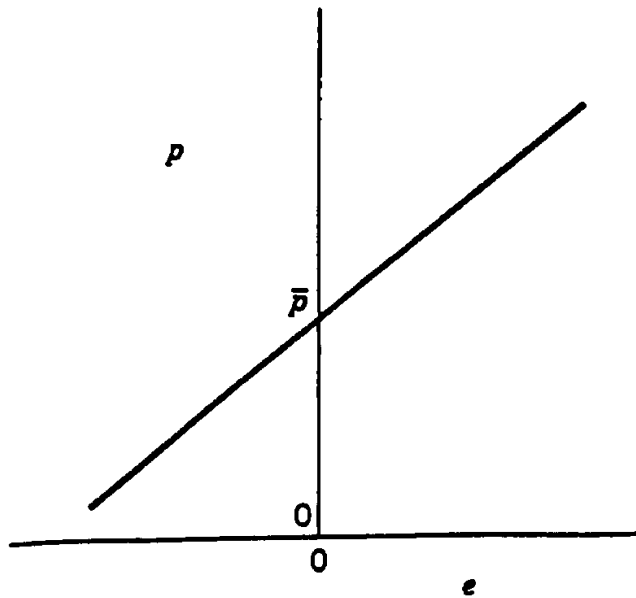$$p(t) = \begin{cases} P_{max} & for & e > \delta \\ P_{min} & for & e < -\delta \end{cases}$$

Figure 8.4. Proportional control: ideal behavior (slope of line $= K_c$).
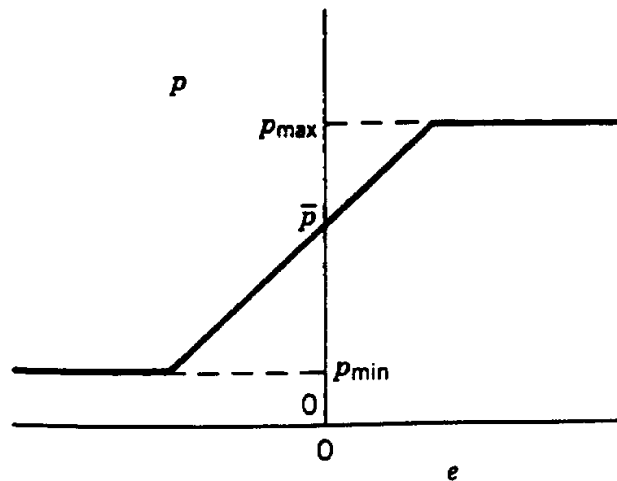

Figure 8.5. Proportional control: actual behavior.

Proportional-Integral (PI) Control

$$p(t) = \bar{p} + K_c \left[e(t) + \frac{1}{\tau_I} \int_0^t e(t') dt' \right]$$
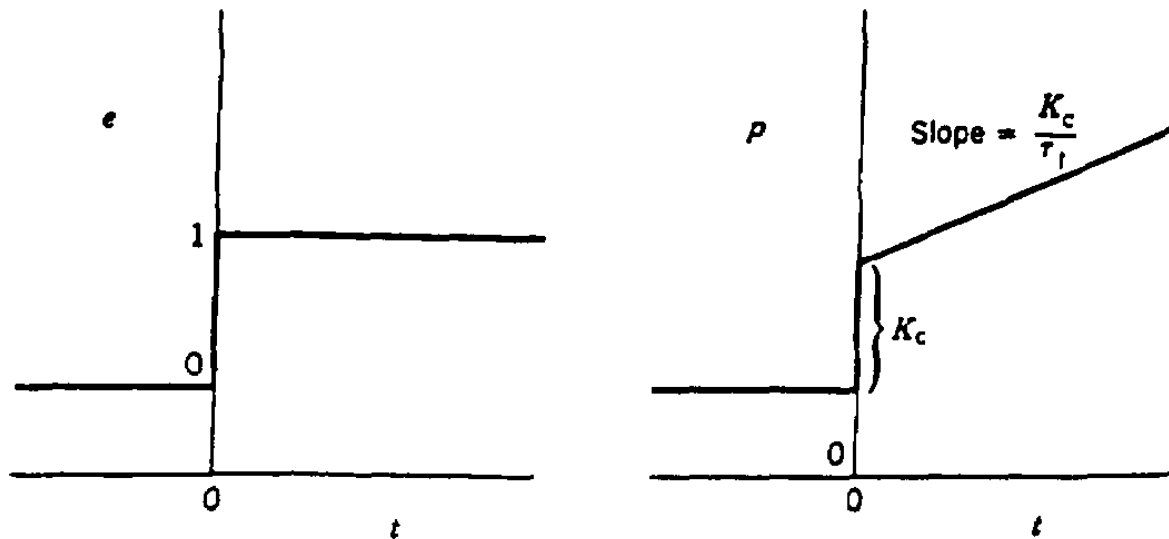
- Response to unit step change in e:

Figure 8.6. Response of proportional-integral controller to unit step change in $e(t)$.

- Integral action eliminates steady-state error (i.e., offset) Why??? $e \neq 0 \Rightarrow p$ is changing with time until $e = 0$, where p reaches steady state.

- Transfer function for PI control

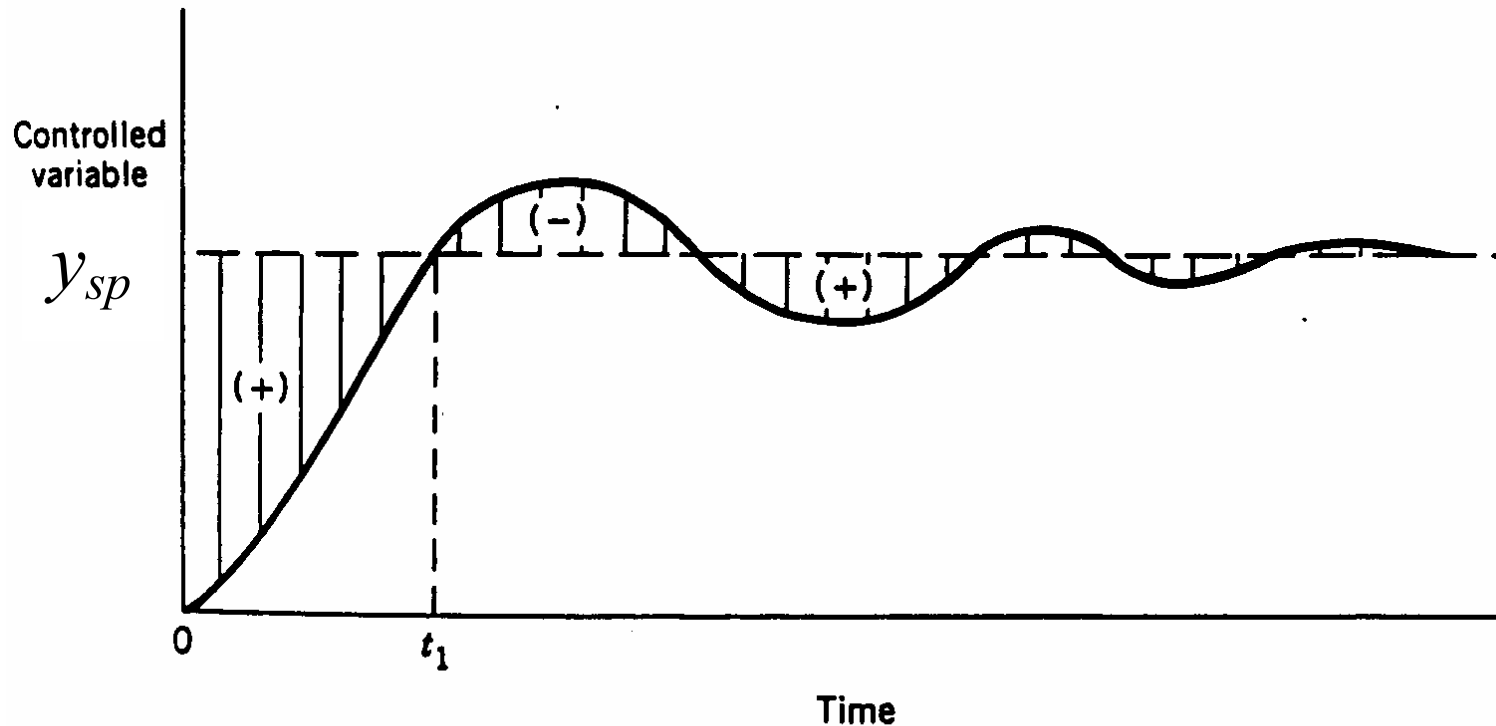$$\frac{P'(s)}{E(s)} = K_c\left(1 + \frac{1}{\tau_I s}\right)$$

Controlled variable

y_{sp}

(+)

(−)

(+)

0   t_1

Time

Figure 8.7.   Reset windup during a set-point change.

26

- Some controllers are calibrated in $1/\tau_I$ ("repeats per minute") instead of τ_I .

- For PI controllers, $\bar{p}$ is not adjustable.

Derivative Control Action

- Ideal derivative action

$$p(t) = \bar{p} + \tau_D \frac{de}{dt}$$

- Used to improve dynamic response of the controlled variable
- Derivative kick (use db/dt)
- Use alone?

PID Controller

- Ideal controller

- Transfer function (ideal)

$$p(t) = \bar{p} + K_c \left[e(t) + \frac{1}{\tau_I} \int_0^t e(t')dt' + \tau_D \frac{de}{dt} \right]$$

$$\frac{P'(s)}{E(s)} = K_c \left(1 + \frac{1}{\tau_I s} + \tau_D s \right)$$

- Transfer function (actual)

$$\frac{P'(s)}{E(s)} = K_c \left(\frac{\tau_I s + 1}{\tau_I s} \right) \left(\underbrace{\frac{\tau_D s + 1}{\alpha \tau_D s + 1}}_{\text{lead / lag units}} \right)$$

α = small number (0.05 to 0.20)

Controller Comparison

P   -   Simplest controller to tune (K_c).
         -   Offset with sustained disturbance or setpoint
             change.

PI   -   More complicated to tune (K_c, τ_I) .
         -   Better performance than P
         -   No offset
         -   Most popular FB controller

PID   -   Most complicated to tune (K_c, τ_I, τ_D) .
         -   Better performance than PI
         -   No offset
         -   Derivative action may be affected by noise

Typical Response of Feedback Control Systems

Consider response of a controlled system after a sustained disturbance occurs (e.g., step change in the disturbance variable)
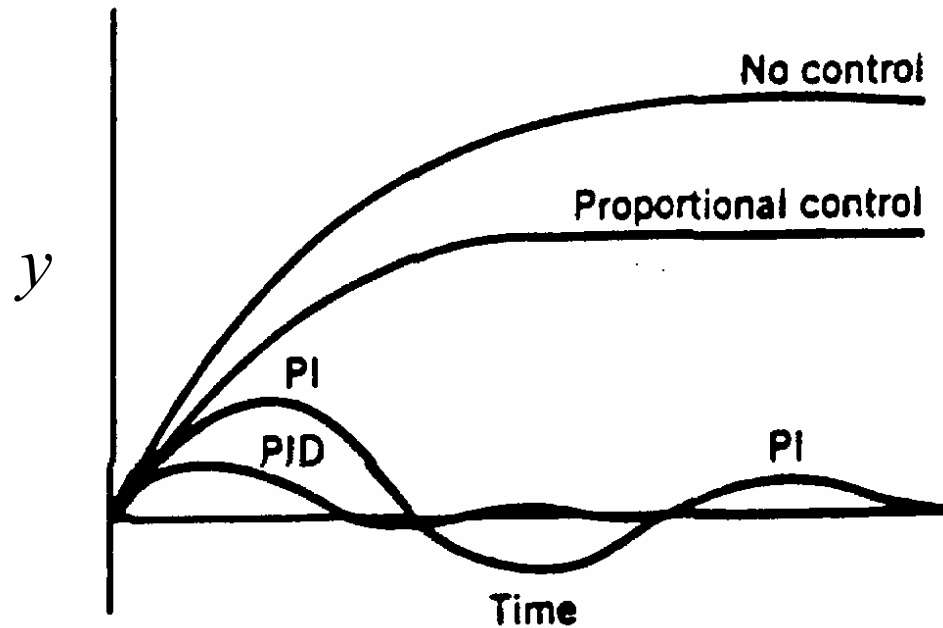

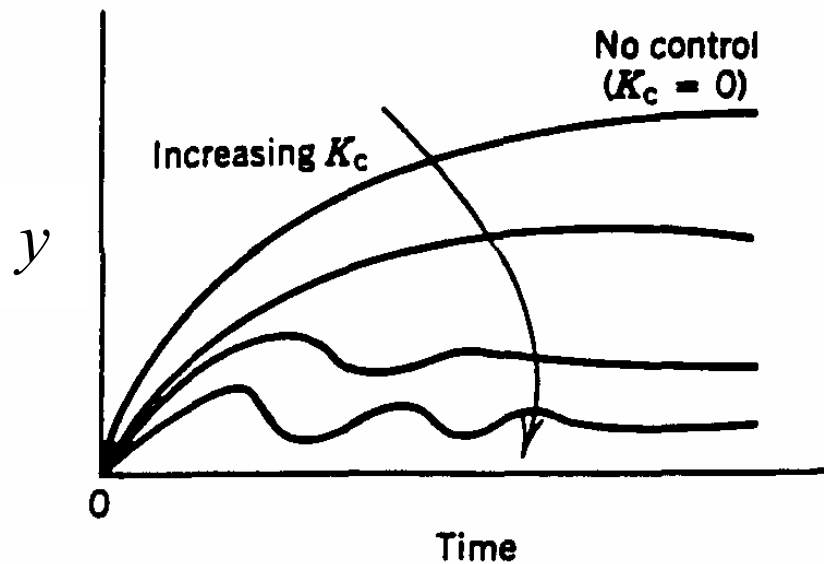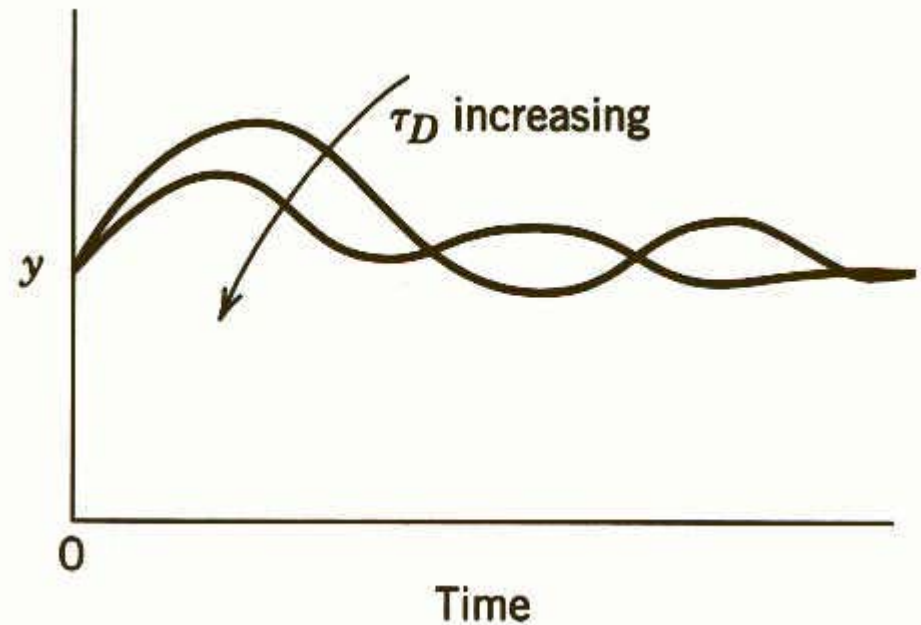Figure 8.12. Typical process responses with feedback control.

Figure 8.13. Proportional control: effect of controller gain.

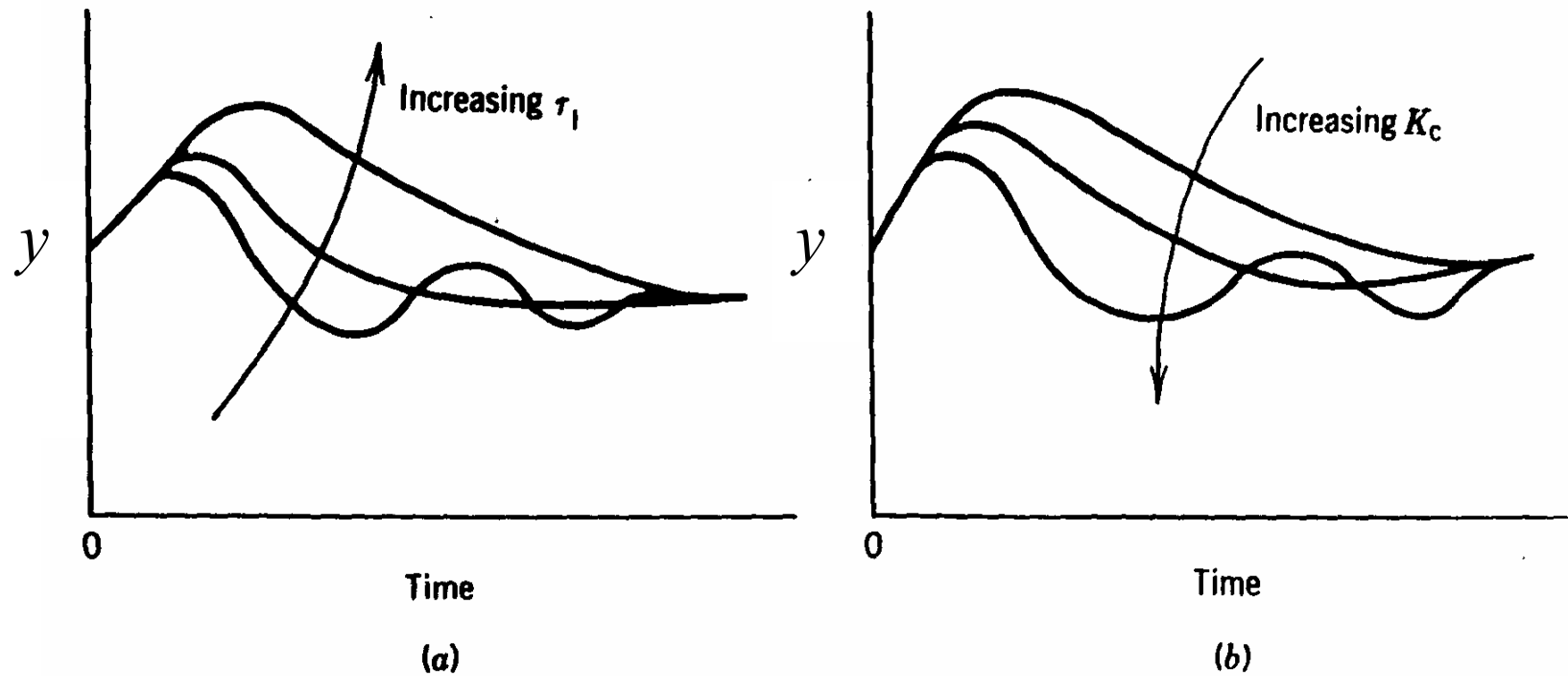Figure 8.15. PID control: effect of derivative time.

Figure 8.14. PI control: (a) effect of reset time (b) effect of controller gain.

Position and Velocity Algorithms for Digital PID Control

A straight forward way of deriving a digital version of the parallel form of the PID controller (Eq. 8-13) is to replace the integral and derivative terms by finite difference approximations,

$$\int_0^t e(t^*)\,dt \approx \sum_{j=1}^{k} e_j \Delta t \qquad (8\text{-}24)$$

$$\frac{de}{dt} \approx \frac{e_k - e_{k-1}}{\Delta t} \qquad (8\text{-}25)$$

where:

Δt = the sampling period (the time between successive measurements of the controlled variable)

e_k = error at the kth sampling instant for $k = 1, 2, \ldots$

There are two alternative forms of the digital PID control equation, the position form and the velocity form. Substituting (8-24) and (8-25) into (8-13), gives the *position form*,

$$p_k = \bar{p} + K_c \left[e_k + \frac{\Delta t}{\tau_1} \sum_{j=1}^{k} e_j + \frac{\tau_D}{\Delta t} \left(e_k - e_{k-1} \right) \right] \qquad (8\text{-}26)$$

Where p_k is the controller output at the kth sampling instant. The other symbols in Eq. 8-26 have the same meaning as in Eq. 8-13. Equation 8-26 is referred to as the position form of the PID control algorithm because the actual value of the controller output is calculated.

In the *velocity form*, the change in controller output is calculated. The velocity form can be derived by writing the position form of (8-26) for the (*k*-1) sampling instant:

$$p_k = \bar{p} + K_c \left[e_k + \frac{\Delta t}{\tau_1} \sum_{j=1}^{k} e_j + \frac{\tau_D}{\Delta t} \left(e_k - e_{k-1} \right) \right] \qquad \text{(8-26)}$$

Note that the summation still begins at $j = 1$ because it is assumed that the process is at the desired steady state for $j \leq 0$ and thus $e_j = 0$ for $j \leq 0$. Subtracting (8-27) from (8-26) gives the velocity form of the digital PID algorithm:

$$\Delta p_k = p_k - p_{k-1} = K_c \left[\left(e_k - e_{k-1} \right) + \frac{\Delta t}{\tau_I} e_k + \frac{\tau_D}{\Delta t} \left(e_k - 2e_{k-1} + e_{k-2} \right) \right]$$

$$\text{(8-28)}$$

The velocity form has three advantages over the position form:

1. It inherently contains anti-reset windup because the summation of errors is not explicitly calculated.

2. This output is expressed in a form, Δp_k, that can be utilized directly by some final control elements, such as a control valve driven by a pulsed stepping motor.

3. For the velocity algorithm, transferring the controller from manual to automatic mode does not require any initialization of the output ($\bar{p}$ in Eq. 8-26). However, the control valve (or other final control element) should be placed in the appropriate position prior to the transfer.