AKSI DASAR PENGENDALIAN

PID Control

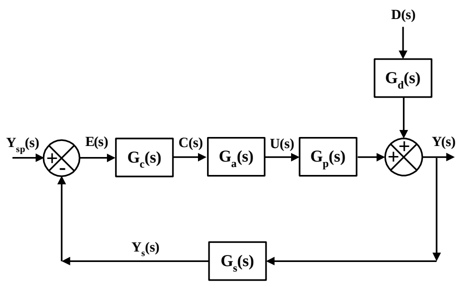
Overall Course Objectives

- Develop the skills necessary to function as an industrial process control engineer.
  - Skills
    - Tuning loops
    - ***Control loop design***
    - Control loop troubleshooting
    - Command of the terminology
  - Fundamental understanding
    - Process dynamics
    - ***Feedback control***

PID Controls

- Most common controller in the CPI.
- Came into use in 1930's with the introduction of pneumatic controllers.
- Extremely flexible and powerful control algorithm when applied properly.

General Feedback Control Loop



Closed Loop Transfer Functions

- From the general feedback control loop and using the properties of transfer functions, the following expressions can be derived:

$$\frac{Y(s)}{Y_{sp}(s)} = \frac{G_p(s)\ G_a(s)\ G_c(s)}{G_p(s)\ G_a(s)\ G_c(s)\ G_s(s) + 1}$$

$$\frac{Y(s)}{D(s)} = \frac{G_d(s)}{G_p(s)\ G_a(s)\ G_c(s)\ G_s(s) + 1}$$

Characteristic Equation

- Since setpoint tracking and disturbance rejection have the same denominator for their closed loop transfer functions, this indicates that both setpoint tracking and disturbance rejection have the same general dynamic behavior.
- The roots of the denominator determine the dynamic characteristics of the closed loop process.
- The **characteristic equation** is given by:

$$G_p(s)\ G_a(s)\ G_c(s)\ G_s(s) + 1 = 0$$

Feedback Control Analysis

- The loop gain ($K_c K_a K_p K_s$) should be positive for stable feedback control.
- An open-loop unstable process can be made stable by applying the proper level of feedback control.

Characteristic Equation Example

- Consider the dynamic behavior of a P-only controller applied to a CST thermal mixer (K_p=1; τ_p=60 sec) where the temperature sensor has a τ_s=20 sec and τ_a is assumed small. Note that $G_c(s)=K_c$.

Substituting into the characteristic equation

$$K_c\left[\frac{1}{60s+1}\right]\left[\frac{1}{20s+1}\right]+1=0$$

After rearranging into the standard form,

$$\tau_p' = \sqrt{\frac{1200}{1+K_c}} \qquad \zeta = \frac{1.15}{\sqrt{1+K_c}}$$

Example Continued- Analysis of the Closed Loop Poles

- When K_c =0, poles are -0.05 and -0.0167 which correspond to the inverse of τ_p and τ_s.
- As K_c is increased from zero, the values of the poles begin to approach one another.
- Critically damped behavior occurs when the poles are equal.
- Underdamped behavior results when K_c is increased further due to the imaginary components in the poles.

In-Class Exercise

- Determine the dynamic behavior of a P-only controller with K_c equal to 1 applied to a first-order process in which the process gain is equal to 2 and the time constant is equal to 22. Assume that $G_s(s)$ is equal to one and $G_a(s)$ behaves as a first-order process with a time constant of 5.

PID Control Algorithm

$$c(t) = c_0 + K_c\left[e(t) + \frac{1}{\tau_I}\int_0^t e(t)dt + \tau_D\frac{de(t)}{dt}\right]$$

where $e(t) = y_{sp} - y_s(t)$

Definition of Terms

- $e(t)$- the error from setpoint [$e(t)=y_{sp}-y_s$].
- K_c- the controller gain is a tuning parameter and largely determines the controller aggressiveness.
- τ_I- the reset time is a tuning parameter and determines the amount of integral action.
- τ_D- the derivative time is a tuning parameter and determines the amount of derivative action.

Transfer Function for a PID Controller

$$G_c(s) = \frac{C(s)}{E(s)} = K_c\left[1 + \frac{1}{\tau_I s} + \tau_D\, s\right]$$

Example for a First Order Process with a PI Controller

$K_c = 2 \quad \tau_I = 10 \quad K_p = 1 \quad \tau_p = 5$

Characteristic Equation :

$$\left[\frac{1}{5s+1}\right]\left[2 + \frac{2}{10s}\right] + 1 = 0$$

Rearranging

$$25s^2 + 15s + 1 = 0$$

$$\tau_p{}' = 5 \qquad \zeta = 1.5$$

Example of a PI Controller Applied to a Second Order Process

$K_c = 1; \ \tau_I = 1; \ K_p = 1; \ \tau_p = 5; \ \zeta = 2$

Characteristic Equation :

$$\left[\frac{1}{25s^2 + 20s + 1}\right]\left[1 + \frac{1}{s}\right] + 1 = 0$$

Rearranging

$$25s^3 + 20s^2 + 2s + 1 = 0$$

$p_1 = -0.764$ and a second order response with $\tau_p = 4.37$ and $\zeta = 0.08$

Properties of Proportional Action

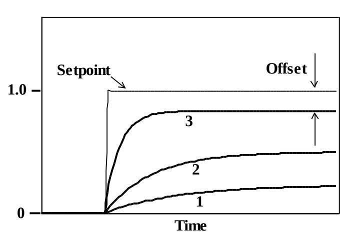
$$c(t) = c_0 + K_c\, e(t)$$

$$G_c(s) = K_c$$

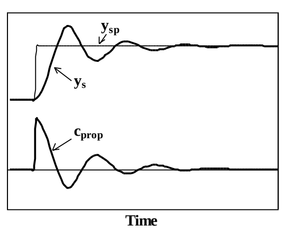
$$\frac{Y(s)}{Y_{sp}(s)} = \frac{\dfrac{K_c K_p}{K_c K_p + 1}}{\dfrac{\tau_p}{K_c K_p + 1} s + 1}$$

- Closed loop transfer function base on a P-only controller applied to a first order process.
- Properties of P control
  - Does not change order of process
  - **Closed loop time constant is smaller than open loop τ_p**
  - Does not eliminate offset.

Offset Resulting from P-only Control



Proportional Action for the Response of a PI Controller



Proportional Action

- The primary benefit of proportional action is that it speedup the response of the process.

Properties of Integral Action

$$c(t) = c_0 + \frac{K_c}{\tau_I}\int_0^t e(t)\ dt$$

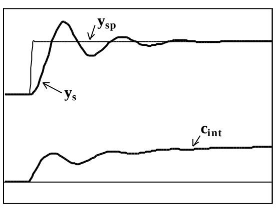
$$\frac{Y(s)}{Y_{sp}(s)} = \frac{1}{\frac{\tau_I\tau_p}{K_cK_p}s^2 + \frac{\tau_I}{K_cK_p}s + 1}$$

$$\tau_p' = \sqrt{\frac{\tau_I\tau_p}{K_cK_p}}$$

$$\zeta = \frac{1}{2}\sqrt{\frac{\tau_I}{\tau_p\ K_cK_p}}$$

- Based on applying an I-only controller to a first order process
- Properties of I control
  - **Offset is eliminated**
  - Increases the order by 1
  - As integral action is increased, the process becomes faster, but at the expense of more sustained oscillations

Integral Action for the Response of a PI Controller



Time

Integral Action

- The primary benefit of integral action is that it removes offset from setpoint.
- In addition, for a PI controller all the steady-state change in the controller output results from integral action.

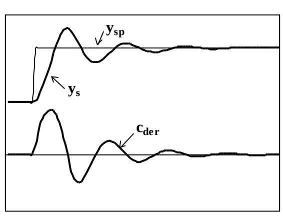
Properties of Derivative Action

$$c(t) = c_0 + K_c\tau_D\frac{de(t)}{dt}$$

$$\frac{Y(s)}{Y_{sp}(s)} = \frac{K_cK_p\tau_D s}{\tau_p^2 s^2 + (\zeta\tau_p + K_cK_p\tau_D)s + 1}$$

- Closed loop transfer function for derivative-only control applied to a second order process.
- Properties of derivative control:
  - Does not change the order of the process
  - Does not eliminate offset
  - **Reduces the oscillatory nature of the feedback response**

Derivative Action for the Response of a PID Controller



Time

Derivative Action

- The primary benefit of derivative action is that it reduces the oscillatory nature of the closed-loop response.

Position Form of the PID Algorithm

$$c(t) = c_0 + K_c\left[e(t) + \frac{1}{\tau_I}\int_0^t e(t)dt + \tau_D \frac{d\,e(t)}{dt}\right]$$

Proportional Band

$$PB = \frac{100\%}{K_c}$$

- Another way to express the controller gain.
- K_c in this formula is dimensionless. That is, the controller output is scaled 0-100% and the error from setpoint is scaled 0-100%.
- In more frequent use 10-15 years ago, but it still appears as an option on DCS's.

Conversion from PB to K_c

- Proportional band is equal to 200%.
- The range of the error from setpoint is 200 psi.
- The controller output range is 0 to 100%.

$$K_c^D = \frac{100\%}{PB} = \frac{100\%}{200\%} = 0.5$$

$$K_c = 0.5\left[\frac{100\%}{200\,\text{psi}}\right] = 0.25\ \%/\text{psi}$$

Conversion from K_c to PB

- Controller gain is equal to 15 %/ºF
- The range of the error from setpoint is 25 ºF.
- The controller output range is 0 to 100%.

$$K_c^D = \left(\frac{15\%}{{}^\circ F}\right)\left(\frac{25^\circ F}{100\%}\right) = 3.75$$

$$PB = \left[\frac{100\%}{3.75}\right] = 26.7\%$$

Digital Equivalent of PID Controller

$$\int_0^\infty e(t)\,dt \approx \sum_{i=1}^{n} e(i\,\Delta t)\,\Delta t$$

- The trapezoidal approximation of the integral.

$$\frac{d\,e(t)}{dt} \approx \frac{e(t) - e(t-\Delta t)}{\Delta t}$$

- Backward difference approximation of the first derivative

Digital Version of PID Control Algorithm

$$c(t) = c_0 + K_c\left[e(t) + \frac{\Delta t}{\tau_I}\sum_{i=1}^{n} e(i\,\Delta t) + \tau_D\,\frac{e(t) - e(t-\Delta t)}{\Delta t}\right]$$

$$n = \frac{t}{\Delta t}$$

Derivation of the Velocity Form of the PID Control Algorithm

$$c(t) = c_0 + K_c\left[e(t) + \frac{\Delta t}{\tau_I}\sum_{i=1}^{n} e(i\,\Delta t) + \tau_D\,\frac{e(t) - e(t-\Delta t)}{\Delta t}\right]$$

$$c(t-\Delta t) = c_0 + K_c\left[e(t-\Delta t) + \frac{\Delta t}{\tau_I}\sum_{i=1}^{n-1} e(i\,\Delta t) + \tau_D\,\frac{e(t-\Delta t) - e(t-2\Delta t)}{\Delta t}\right]$$

$$\Delta c(t) = K_c\left[e(t) - e(t-\Delta t) + \frac{\Delta t\, e(t)}{\tau_I} + \tau_D\left(\frac{e(t) - 2e(t-\Delta t) + e(t-2\Delta t)}{\Delta t}\right)\right]$$

Velocity Form of PID Controller

$$\Delta c(t) = K_c\left[e(t) - e(t-\Delta t) + \frac{\Delta t\ e(t)}{\tau_I} + \tau_D\left(\frac{e(t) - 2e(t-\Delta t) + e(t-2\Delta t)}{\Delta t}\right)\right]$$

$$c(t) = c(t-\Delta t) + \Delta c(t)$$

- Note the difference in proportional, integral, and derivative terms from the position form.
- Velocity form is the form implemented on DCSs.

Correction for Derivative Kick

- Derivative kick occurs when a setpoint change is applied that causes a spike in the derivative of the error from setpoint.
- Derivative kick can be eliminated by replacing the approximation of the derivative based on the error from setpoint with the negative of the approximation of the derivative based on the measured value of the controlled variable, i.e.,

$$-\tau_D\,\frac{y_s(t) - 2y_s(t-\Delta t) + y_s(t-2\,\Delta t)}{\Delta t}$$

Correction for Aggressive Setpoint Tracking

- For certain process, tuning the controller for good disturbance rejection performance results in excessively aggressive action for setpoint changes.
- This problem can be corrected by removing the setpoint from the proportional term. Then setpoint tracking is accomplished by integral action only.

$$K_c\left[e(t) - e(t-\Delta t)\right] \text{ substituted for by } K_c\left[y_s(t-\Delta t) - y_s(t)\right]$$

The Three Versions of the PID Algorithm Offered on DCS's

- (1) The original form in which the proportional, integral, and derivative terms are based on the error from setpoint

$$\Delta c(t) = K_c\left[e(t) - e(t-\Delta t) + \frac{\Delta t\, e(t)}{\tau_I} + \tau_D\left(\frac{e(t) - 2e(t-\Delta t) + e(t-2\Delta t)}{\Delta t}\right)\right]$$

The Three Versions of the PID Algorithm Offered on DCSs

- (2) The form in which the proportional and integral terms are based on the error from setpoint while the derivative-on-measurement is used for the derivative term.

$$\Delta c(t)=K_c\left[e(t)-e(t-\Delta t)+\frac{\Delta t\, e(t)}{\tau_I}-\tau_D\left(\frac{y_s(t)-2y_s(t-\Delta t)+y_s(t-2\Delta t)}{\Delta t}\right)\right]$$

The Three Versions of the PID Algorithm Offered on DCS's

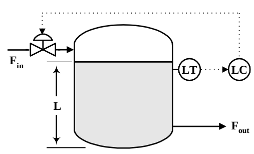
- (3) The form in which the proportional and derivative terms are based on the process measurement and the integral is based on the error from setpoint.

$$\Delta c(t)=K_c\left[y_s(t-\Delta t)-y_s(t)+\frac{\Delta t\, e(t)}{\tau_I}-\tau_D\left(\frac{y_s(t)-2y_s(t-\Delta t)+y_s(t-2\Delta t)}{\Delta t}\right)\right]$$

Guidelines for Selecting Direct and Reverse Acting PID's

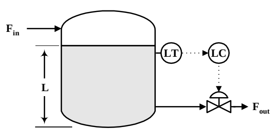
- Consider a direct acting final control element to be positive and reverse to be negative.
- If the sign of the product of the final control element and the process gain is positive, use the reverse acting PID algorithm.
- If the sign of the product is negative, use the direct acting PID algorithm
- If control signal goes to a control valve with a valve positioner, the actuator is considered direct acting.

Level Control Example



- Process gain is positive because when flow in is increased, the level increases.
- If the final control element is direct acting, use reverse acting PID.
- For reverse acting final control element, use direct acting PID.

Level Control Example



- Process gain is negative because when flow out is increased, the level decreases.
- If the final control element is direct acting, use direct acting PID.
- For reverse acting final control element, use reverse acting PID.

In-Class Exercise

- Write the position form of the PID algorithm for Example 3.4, and assume that the control valve on the feed line to the mixer has an air-to-close actuator. Use the form that is not susceptible to derivative kick. Specify whether the controller is a direct-acting or reverse-acting controller.

In-Class Exercise

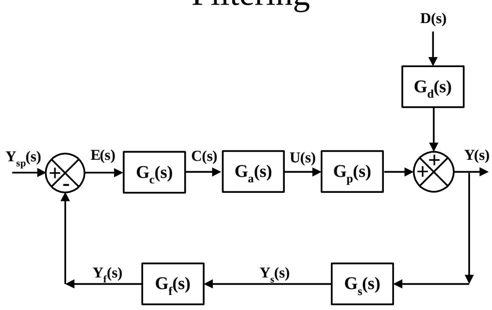
- Write the velocity form of the PID algorithm for Example 3.1, and assume that the control valve on the feed line to the mixer has an air-to-open actuator. Use the form that is not susceptible to derivative kick or proportional kick. Specify whether the controller is a direct-acting or reverse-acting controller.

Filtering the Process Measurement

$$y_f(t) = f\, y_s(t) + (1-f)\, y_f(t-\Delta t)$$

- Filtering reduces the effect of sensor noise by approximating a running average.
- Filtering adds lag when the filtered measurement is used for control.
- Normally, use the minimum amount of filtering necessary.
- *f*- filter factor (0-1)

Feedback Loop with Sensor Filtering



Effect of Filtering on Closed Loop Dynamics

Characteristic equation for P – only controller on first order process with sensor filtering :

$$K_c\left[\frac{K_p}{\tau_p s+1}\right]\left[\frac{1}{\tau_f s+1}\right]+1=0$$

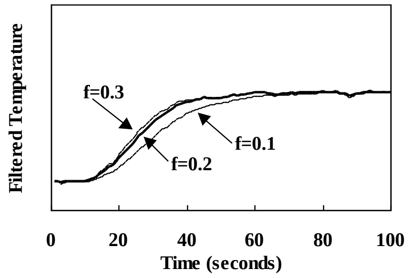
$$\tau_p' = \sqrt{\frac{\tau_p \tau_f}{K_c K_p + 1}}$$

$$\zeta = \frac{\tau_p + \tau_f}{2\sqrt{\tau_p \tau_f (K_c K_p + 1)}}$$

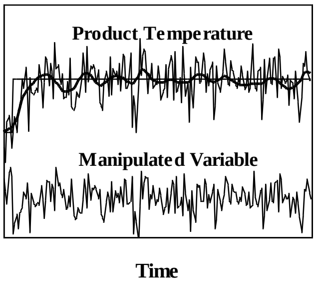
Analysis of Example

- τ_f is equal to $\Delta t\,(1/f-1)$ as f becomes small, τ_f becomes large.
- As τ_f is increased, τ_p' will increase.
- Critical issue is relative magnitude of τ_f compare to τ_p.

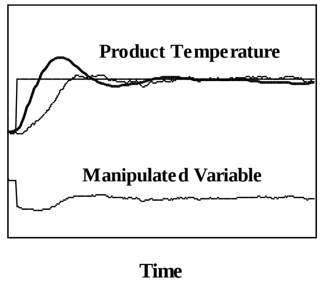
Effect of the Amount of Filtering on the Open Loop Response



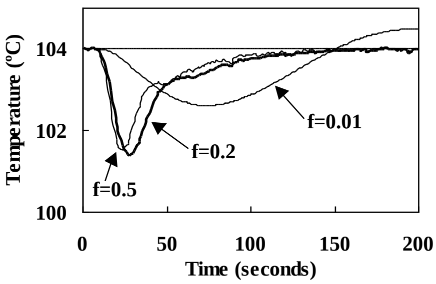
Effect of a Noisy Sensor on Controlled Variable without Filtering



Effect of a Noisy Sensor on Controlled Variable with Filtering



An Example of Too Much and Too Little Filtering



Relationship between Filter Factor (*f*), the Resulting Repeatability Reduction Ratio (*R*) and the Filter Time Constant (τ_f)

$$f = \frac{2}{R^2+1} \quad \text{or} \quad R = \sqrt{\frac{2-f}{f}}$$

$$\tau_f = \Delta t_f \left[\frac{1}{f} - 1 \right]$$

Key Issues for Sensor Filtering

- To reduce the effect of noise (i.e., *R* is increased), *f* must be reduced, which increases the value of τ_f. Filtering slows the closed-loop response significantly as τ_f becomes larger than 10% of τ_p.
- The effect of filtering on the closed-loop response can be reduced by increasing the frequency with which the filter is applied, i.e., reducing $\Delta\tau_f$.

PID Controller Design Issues

- Over 90% of control loops use PI controller.
- P-only: used for fast responding processes that do not require offset free operation (e.g., certain level and pressure controllers)
- PI: used for fast responding processes that require offset free operation (e.g., certain flow, level, pressure, temperature, and composition controllers)

Integrating Processes

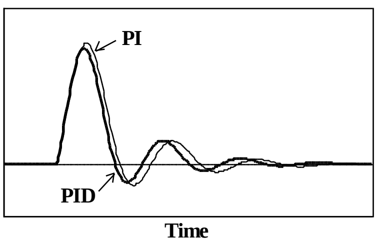
- For integrating processes, P-only control provides offset-free operation. In fact, if as integral action is added to such a case, the control performance degrades.
- Therefore, for integrating processes, P-only control is all that is usually required.

PID Controller Design Issues

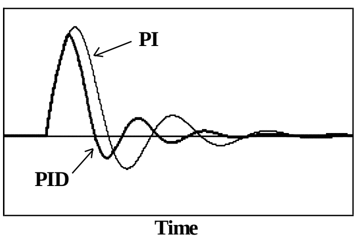
- PID: use for sluggish processes (i.e., a process with large deadtime to time constant ratios) or processes that exhibit severe ringing for PI controllers. PID controllers are applied to certain temperature and composition control loops. Use derivative action when:

$$\frac{\theta_p}{\tau_p} > 1$$

Comparison between PI and PID for a Low θ_p/τ_p Ratio



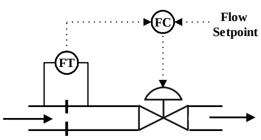
Comparison between PI and PID for a High θ_p/τ_p Ratio



Analysis of Several Commonly Encountered Control Loops

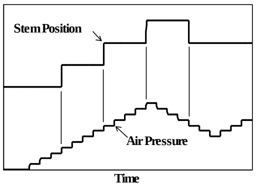
- Flow control loops
- Level control loops
- Pressure control loops
- Temperature control loops
- Composition control loops
- DO control loop
- Biomass controller

Flow Control Loop



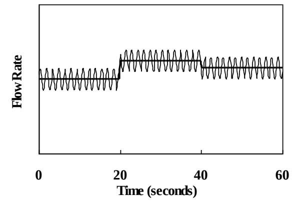
- Since the flow sensor and the process (changes in flow rate for a change in the valve position) are so fast, the dynamics of the flow control loop is controlled by the dynamics of the control valve.
- Almost always use PI controller.

Deadband of a Control Valve



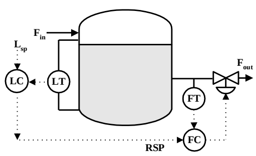
- Deadband of industrial valves is between ±10%-±25%.
- As a result, small changes in the air pressure applied to the valve do not change the flow rate.

Deadband of Flow Control Loop



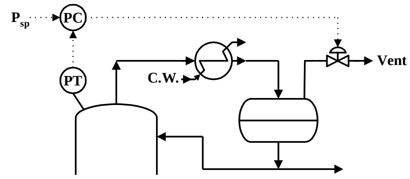
- A control valve (deadband of ±10-25%) in a flow control loop or with a positioner typically has a deadband for the **average flow rate** of less than ±0.5% due to the high frequency opening and closing of the valve around the specified flow rate.

Level Control Loop



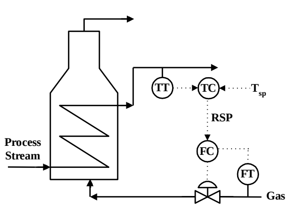
- Dynamics of the sensor and actuator are fast compared to the process.
- Use P-only controller if it is an integrating process.

Pressure Control Process



- The sensor is generally faster than the actuator, which is faster than the process.
- Use P-only controller if it is an integrating process otherwise use a PI controller.

Temperature Control Loop



- The dynamics of the process and sensor are usually slower than the actuator.
- Use a PI controller unless the process is sufficiently sluggish to warrant a PID controller.

Analysis of PI Controller Applied to Typical Temperature Loop

$\tau_I = 30; \quad \tau_p = 60; \quad \tau_s = 20$

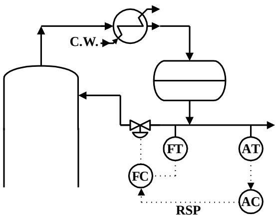
$$K_c \left[1 + \frac{1}{30s}\right] \left[\frac{K_p}{60s + 1}\right] \left[\frac{1}{20s + 1}\right] + 1 = 0$$

K_c	K_p	p_1	τ_p'	ζ
0.02		0.057	325	1.48
0.10		0.056	140	0.74
0.80		0.051	48	0.37
1.50		0.057	28	0.14

Further Analysis of Dynamic of a Typical Temperature Control Loop

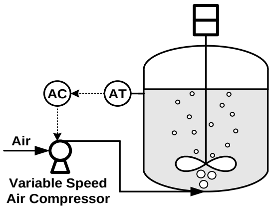
- Note that as the controller gain is increased, i.e., K_cK_p increase, the closed loop time constant becomes smaller.
- Also, note that as the controller gain is increased, the value of ζ decreases.

Composition Control Loop



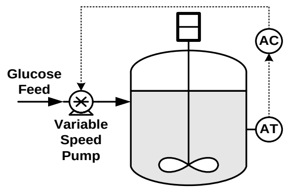
- The process is usually the slowest element followed by the sensor with the actuator being the fastest.
- Use a PI controller unless the process is sufficiently sluggish to warrant a PID controller.

DO Control Loop



- The process and the sensor have approximately the same dynamic response.
- This is a fast responding process for which offset-free operation is desired. Therefore, PI controller should be used.

Biomass Controller



- The process for this system is the slowest element.
- Because the process is a high-order sluggish process, a PID controller is required.

Overview

- The characteristic equation determines the dynamic behavior of a closed loop system
- Proportional, integral, and derivative action each have unique characteristics.
- There are a number of different ways to apply a PID controller.
- Use a PI controller unless offset is not important or if the process is sluggish.
- When analyzing the dynamics of a loop, consider the dynamics of the actuator, the process, and the sensor separately.