Proportional-Integral-Derivative (PID) Controllers

(Three term controller)

Prof Dr Khalaf S Gaeid

Control Systems Lecture

4th Electrical Department

1. Introduction

- A proportional-integral-derivative controller (PID controller) is a control loop feedback mechanism (controller) widely used in industrial control systems.
- A PID controller calculates an error value as the difference between a measured process variable and a desired set point.
- 90% (or more) of control loops in industry are PID
- Simple control design model → simple controller

1. Introduction

- The controller attempts to minimize the error by adjusting the process through use of a manipulated variable
- The PID controller algorithm involves three separate constant parameters, and is accordingly sometimes called three-term control: the proportional, the integral and derivative values, denoted P, I, and D
- Simply, these values can be interpreted in terms of time: P depends on the present error, I on the accumulation of past errors, and D is a prediction of future errors, based on current rate of change.

- The weighted sum of these three actions is used to adjust the process via a control element such as the position of a control valve, a damper, or the power supplied to a heating element.
- In the absence of knowledge of the underlying process, a PID controller has historically been considered to be the most useful controller.
- By tuning the three parameters in the PID controller algorithm, the controller can provide control action designed for specific process requirements. The response of the controller can be described in terms of the responsiveness of the controller to an error, the degree to which the controller overshoots the setpoint, and the degree of system oscillation.

- Note that the use of the PID algorithm for control does not guarantee optimal control of the system or system stability.

- Some applications may require using only one or two actions to provide the appropriate system control. This is achieved by setting the other parameters to zero. A PID controller will be called a PI, PD, P or I controller in the absence of the respective control actions.
- PI controllers are fairly common, since derivative action is sensitive to measurement noise, whereas the absence of an integral term may prevent the system from reaching its target value due to the control action.

- **2. PID Controller Theory**
- The PID control scheme is named after its three correcting terms, whose sum constitutes the manipulated variable (MV). The proportional, integral, and derivative terms are summed to calculate the output of the PID controller. Defining $u(t)$ as the controller output,
- Controller manufacturers arrange the Proportional, Integral and Derivative modes into three different controller algorithms or controller structures.

- These are called Interactive, Noninteractive, and Parallel algorithms. Some controller manufacturers allow you to choose between different controller algorithms as a configuration option in the controller software.

The PID Algorithms are:

- 1) Interactive Algorithm

$$u(t) = K_c\left[e(t) + \frac{1}{Ti}\int_0^t e(\tau)d\tau\right] \times \left[1 + T_d\,\frac{d}{dt}e(t)\right]$$

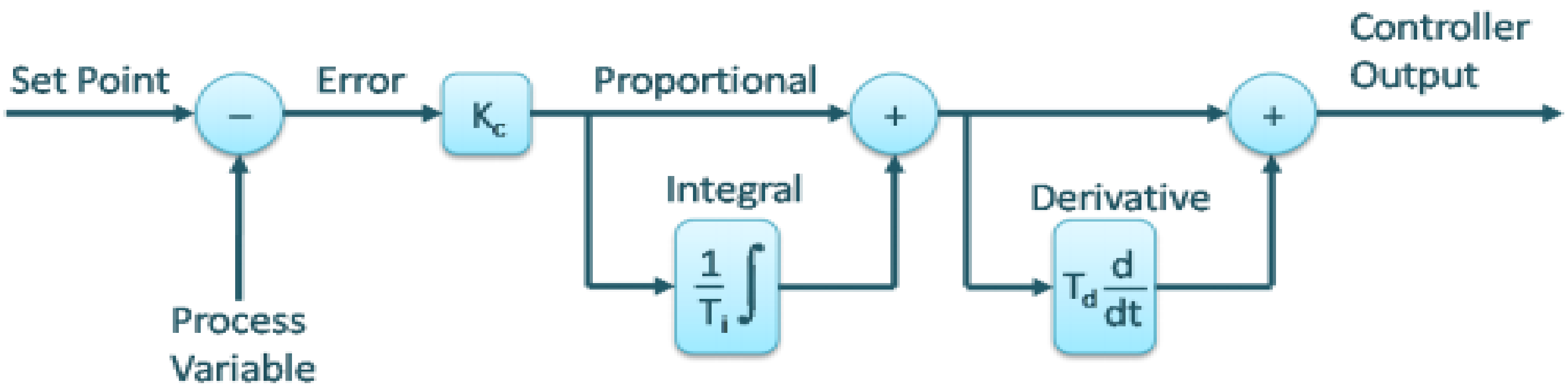


Figure 1: Interactive Algorithm

2) NonInteractive Algorithm

$$u(t) = K_c \left[e(t) + \frac{1}{Ti} \int_0^t e(\tau) d\tau + T_d \frac{d}{dt} e(t) \right]$$

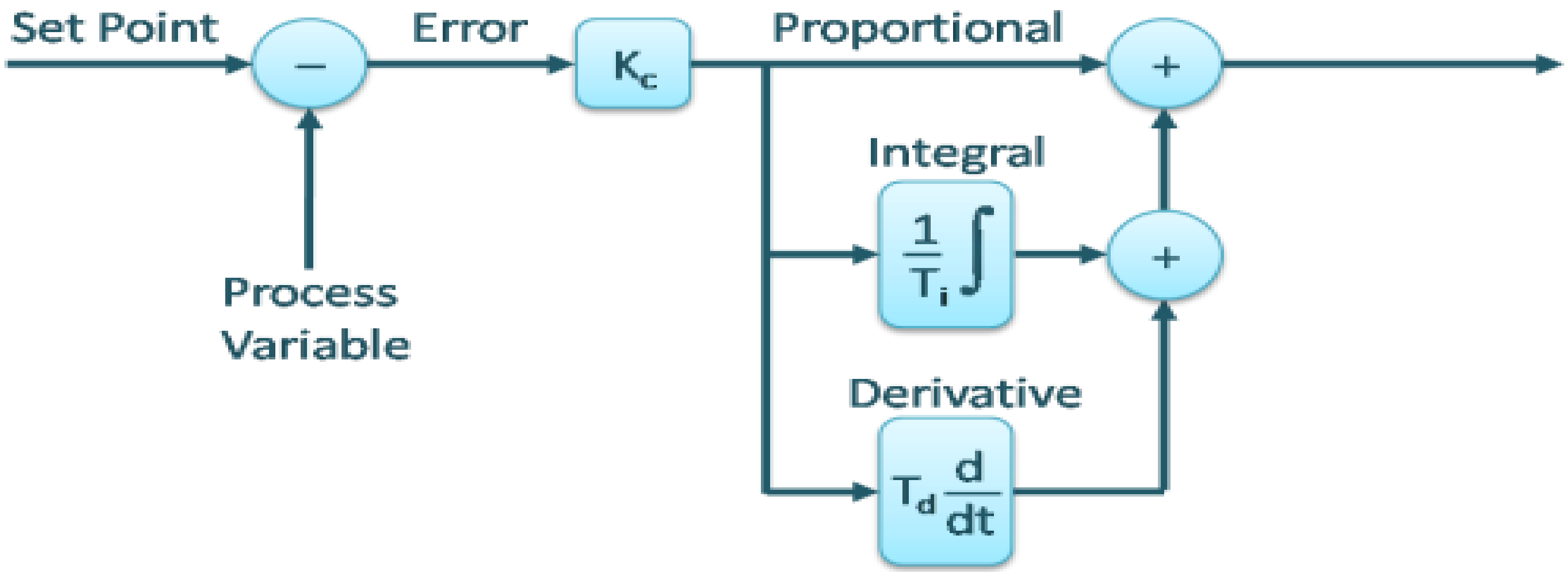


Figure 2: NonIneractive Algorithm

3) Parallel Algorithm

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{d}{dt} e(t)$$

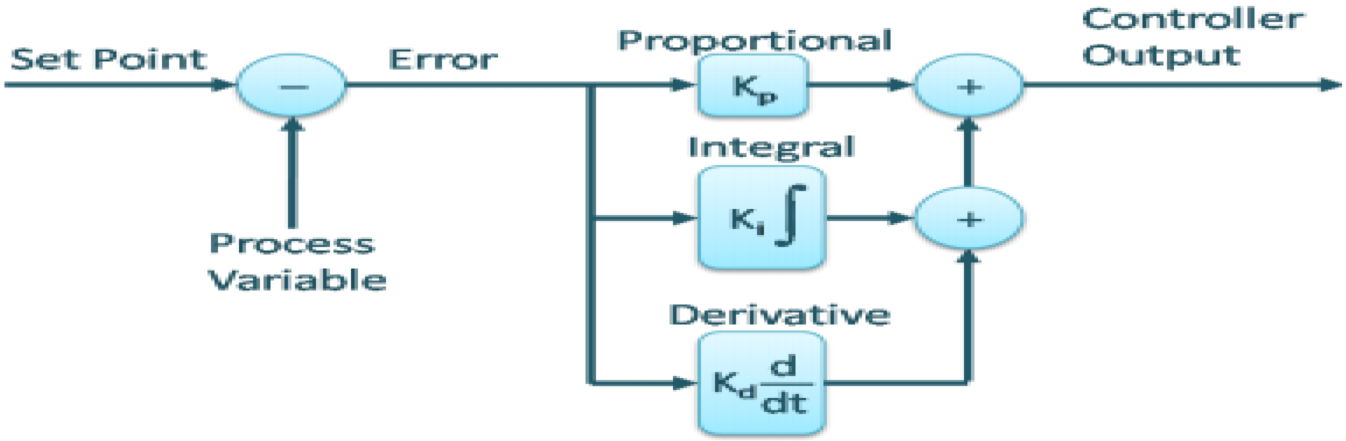


Figure 3: Parallel Algorithm

Where

$K_p = K_c$: *Propotional Gain*

$K_i = \frac{K_c}{T_i}$: *Integral Gain*

$K_d = K_c T_d$: *Derivative Gain*

$e(t) = r(t) - y(t)$

1 Proportional Term

- The proportional term produces an output value that is proportional to the current error value. The proportional response can be adjusted by multiplying the error by a constant K_p, called the proportional gain constant. The proportional term is given by:

$$P_{out} = K_p * e(t)$$

- A high proportional gain results in a large change in the output for a given change in the error. If the proportional gain is too high, the system can become unstable.
- In contrast, a small gain results in a small output response to a large input error, and a less responsive or less sensitive controller

- If the proportional gain is too low, the control action may be too small when responding to system disturbances. Tuning theory and industrial practice indicate that the proportional term should contribute the bulk of the output change

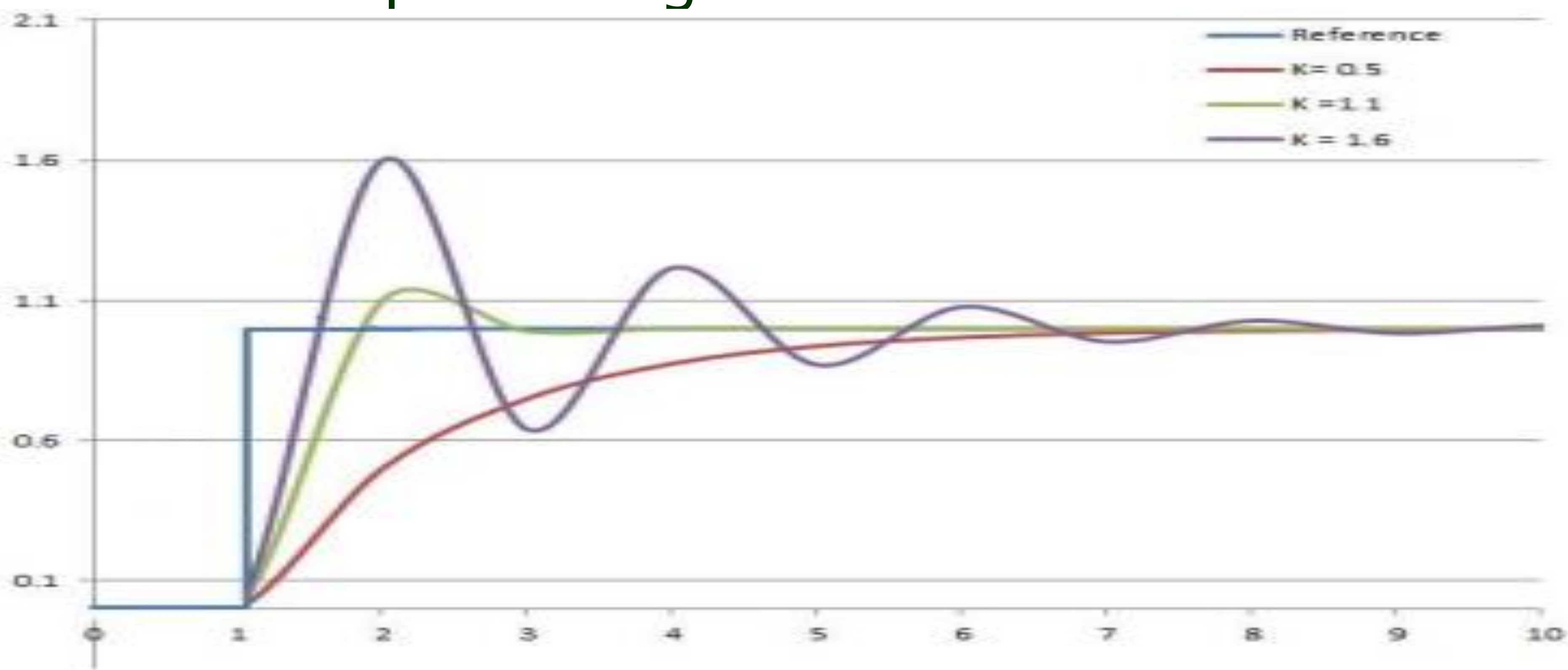


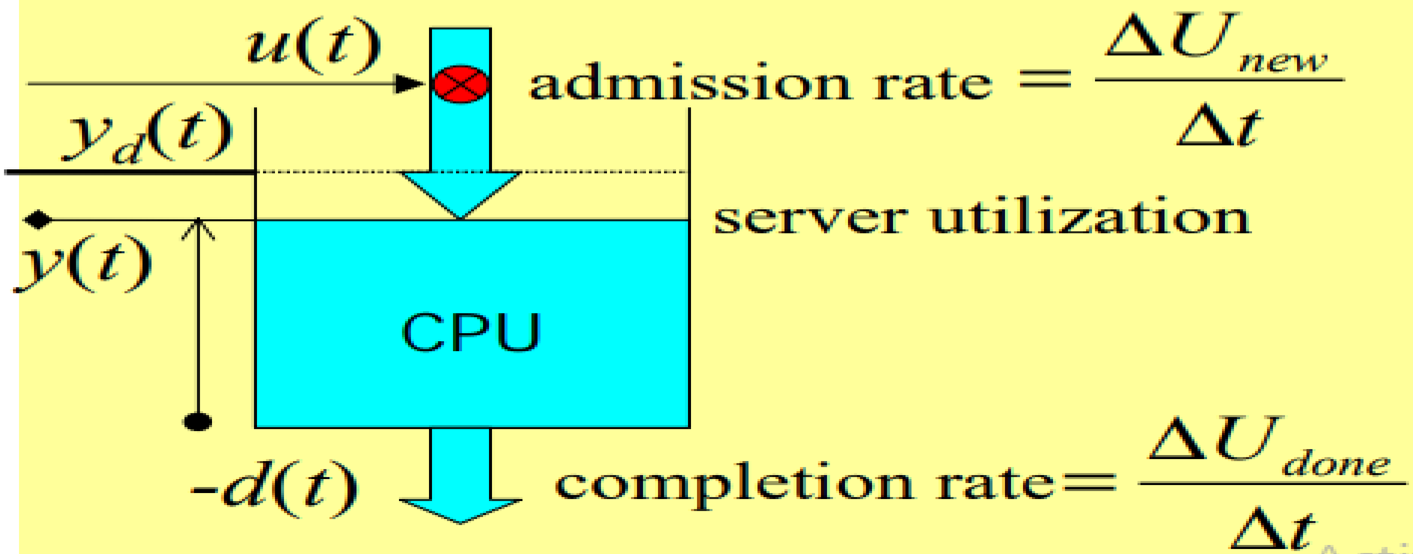
Figure 4: The effect of add K_p (K_i, *and* K_d) held constan

Example:

Utilization control in a video server

Video stream *i*

- processing time $c[i]$, period $p[i]$
- CPU utilization: $U[i]$=c$[i]/p[i]$



Integral Term

- The contribution from the integral term is proportional to both the magnitude of the error and the duration of the error. The integral in a PID controller is the sum of the instantaneous error over time and gives the accumulated offset that should have been corrected previously. The accumulated error is then multiplied by the integral gain $\boldsymbol{Ki}$ and added to the controller output.
- $I_{Out} = K_i * \int e(\tau) d\,\tau$

- The integral term accelerates the movement of the process towards set-point and eliminates the residual steady-state error that occurs with a pure proportional controller. However, since the integral term responds to accumulated errors from the past, it can cause the present value to overshoot the set-point value.

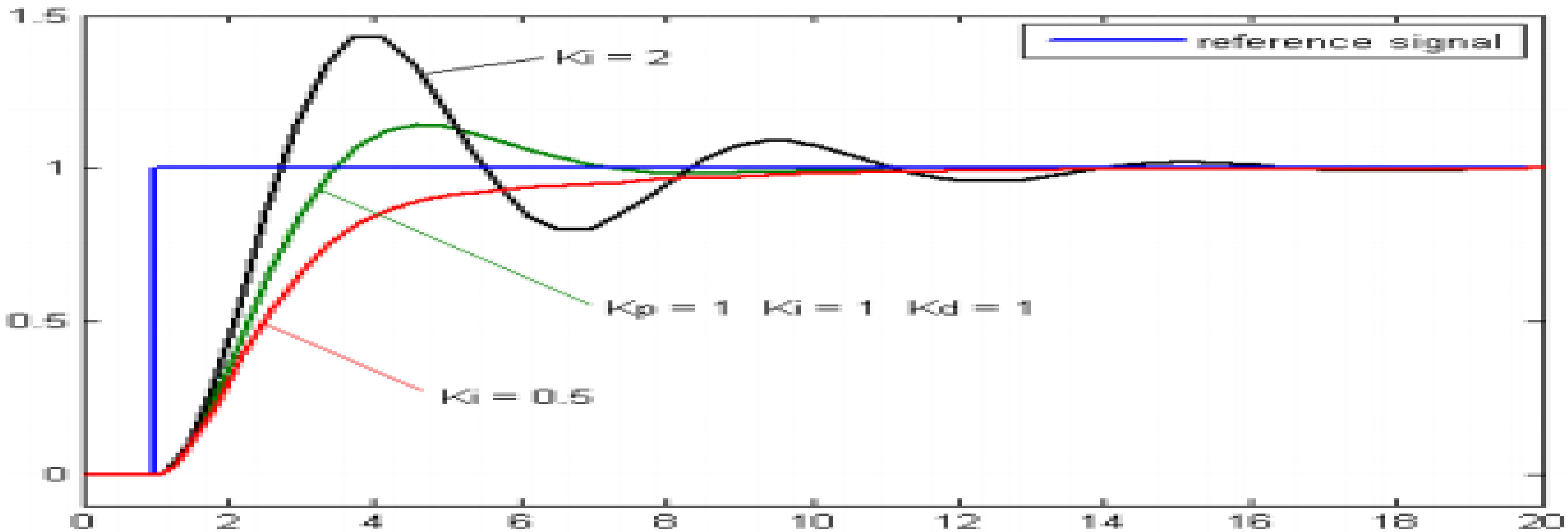
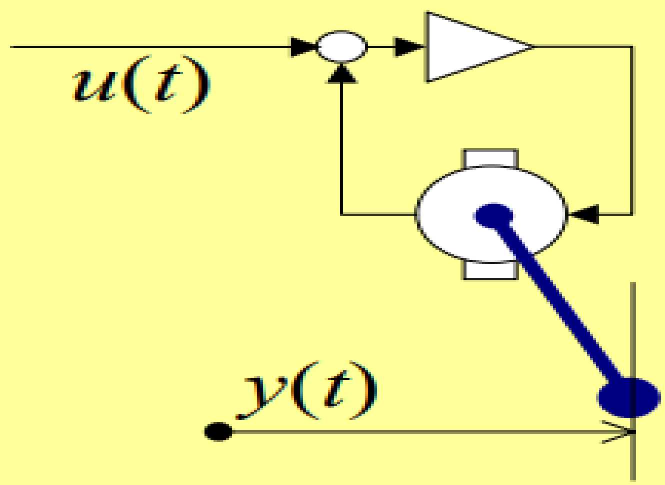


Figure 5: The effect of add K_i (K_p, *and* K_d) held constant

Example:

- Servosystem command



- More:
  - stepper motor
  - flow through a valve
  - motor torque ...

Derivative Term

- The derivative of the process error is calculated by determining the slope of the error over time and multiplying this rate of change by the derivative gain K_d. The magnitude of the contribution of the derivative term to the overall control action is termed the derivative gain, K_d. The derivative term is given by
- $D_{out} = K_d \, de(t)/ \, dt$

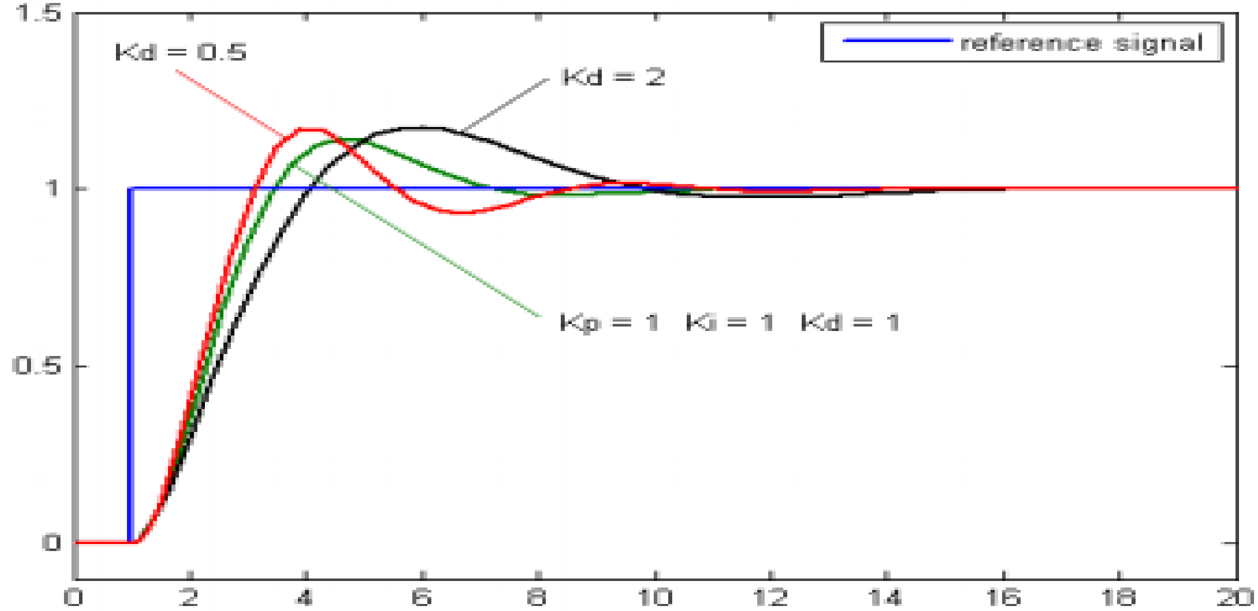


Figure 6 The effect of add K_d (K_p, *and* K_i) held constant

Derivative action predicts system behavior and thus improves settling time and stability of the system. An ideal derivative is not causal, so that implementations of PID controllers include an additional low pass filtering for the derivative term, to limit the high frequency gain and noise. Derivative action is seldom used in practice though - by one estimate in only 20% of deployed controllers- because of its variable impact on system stability in real-world applications.

Table 1: Effect of increasing parameter independently

Parameter	Rise Time	Overshoot	Settling Time	Steady-State Error	Stability
K_p	Decrease	Increase	Small Change	Decrease	Degrade
K_i	Decrease	Increase	Increase	Eliminate	Degrade
K_d	Minor Change	Decrease	Decrease	No Effect	Improve if K_d small

3. Overview of Tuning Methods

- There are several methods for tuning a PID loop. The most effective methods generally involve the development of some form of process model, then choosing P, I, and D based on the dynamic model parameters. Manual tuning methods can be relatively inefficient, particularly if the loops have response times on the order of minutes or longer.

- The choice of method will depend largely on whether or not the loop can be taken "offline" for tuning, and on the response time of the system.
- If the system can be taken offline, the best tuning method often involves subjecting the system to a step change in input, measuring the output as a function of time, and using this response to determine the control parameters.

Table 2: Choosing a Tuning Method

Method	Advantages	Disadvantages
Manual Tuning	No math required , Online	Requires experienced personnel
Ziegler-Nichols	Proven Method, Online	Process upset, some trial-and-error, very aggressive tuning
Cohen-Coon	Good process models	Some math; offline; only good for first-order processes
Software Tools	Consistent tuning; online or offline - can employ computer-automated control system design (CAutoD) techniques;	Some cost or training involved

Usefulness of PID Controls

- Most useful when a mathematical model of the plant is not available
- Many different PID tuning rules available
- Our sources
  - K. Ogata, *Modern Control Engineering*, Fourth Edition, Prentice Hall, 2002, Chapter 10
  - *IEEE Control Systems Magazine,* Feb. 2006, Special issue on PID control

☞ **Proportional-integral-derivative (PID)** control framework is a method to control uncertain systems

Type A PID Control

- Transfer function of PID controller

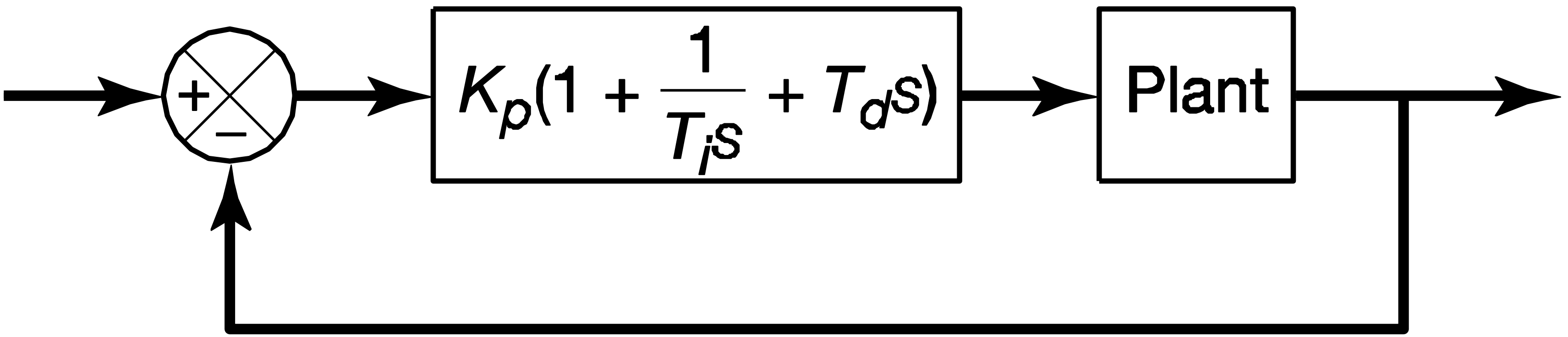
$$G_{PID}(s) = \frac{U(s)}{E(s)} = K_p\left(1 + \frac{1}{T_i s} + T_d s\right)$$

- The three term control signal

$$U(s) = K_p E(s) + K_i \frac{1}{s} E(s) + K_d s E(s)$$

PID-Controlled System

PID controller in forward path

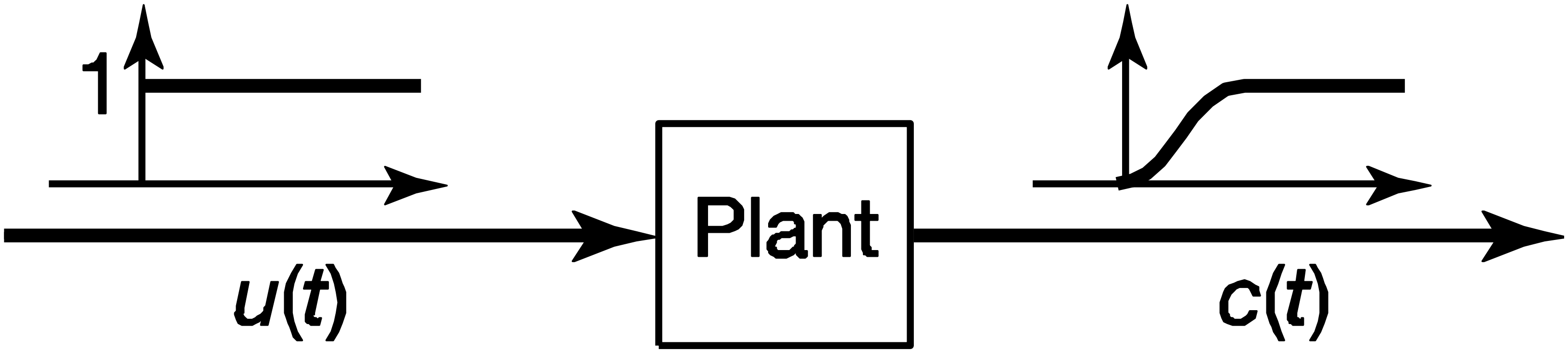


PID Tuning

- Controller tuning---the process of selecting the controller parameters to meet given performance specifications
- PID tuning rules---selecting controller parameter values based on experimental step responses of the controlled plant
- The first PID tuning rules proposed by Ziegler and Nichols in 1942
- Our exposition based on K. Ogata, *Modern Control Engineering*, Prentice Hall, Fourth Edition, 2002, Chapter 10

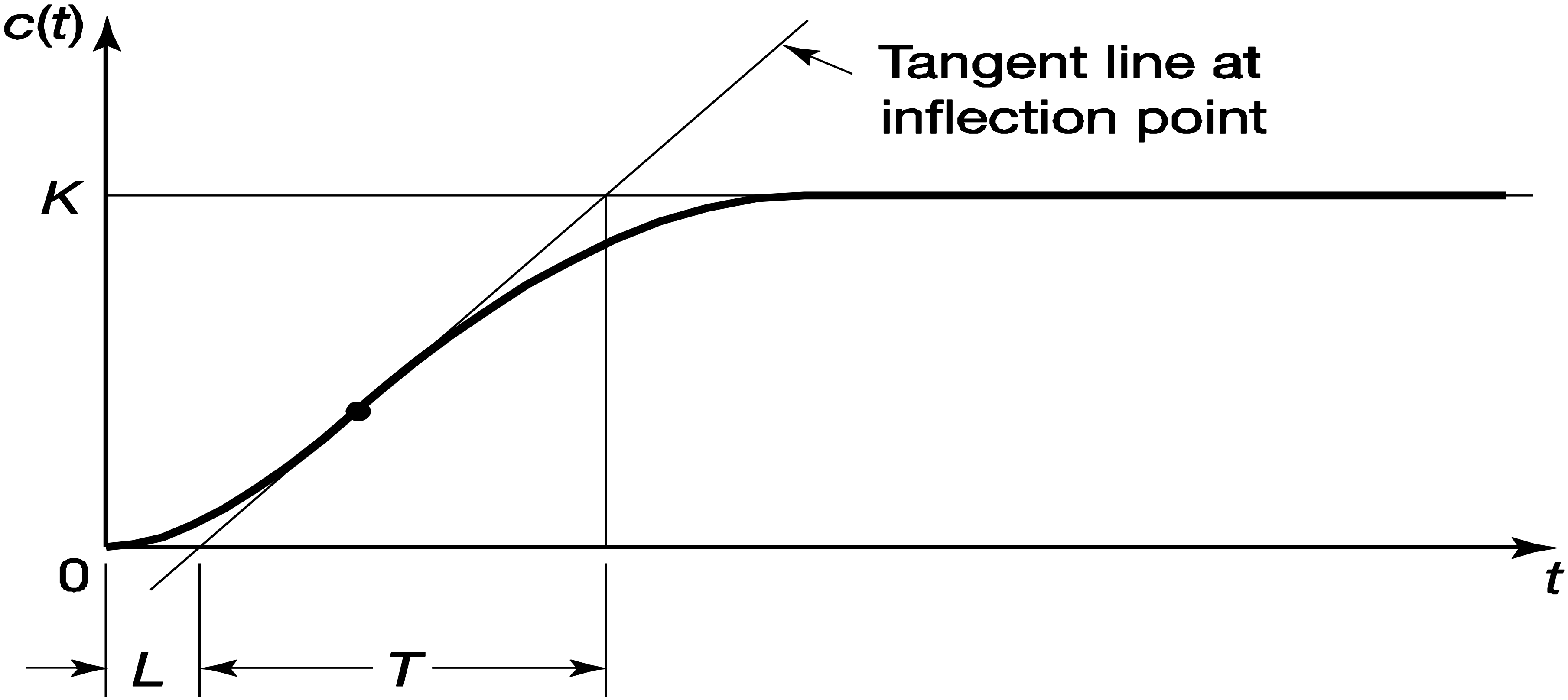
PID Tuning---First Method

Start with obtaining the step response



The S-shaped Step Response

Parameters of the S-shaped step response



The S-Shaped Step Response

- The S-shaped curve may be characterized by two parameters: delay time L and time constant T
- The transfer function of such a plant may be approximated by a first-order system with a transport delay

$$\frac{C(s)}{U(s)} = \frac{Ke^{-Ls}}{Ts+1}$$

PID Tuning---First Method

Table 10–1 Ziegler–Nichols Tuning Rule Based on Step Response of Plant (First Method)

Type of Controller	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0.9\frac{T}{L}$	$\frac{L}{0.3}$	0
PID	$1.2\frac{T}{L}$	$2L$	$0.5L$

Transfer Function of PID Controller Tuned Using the First Method

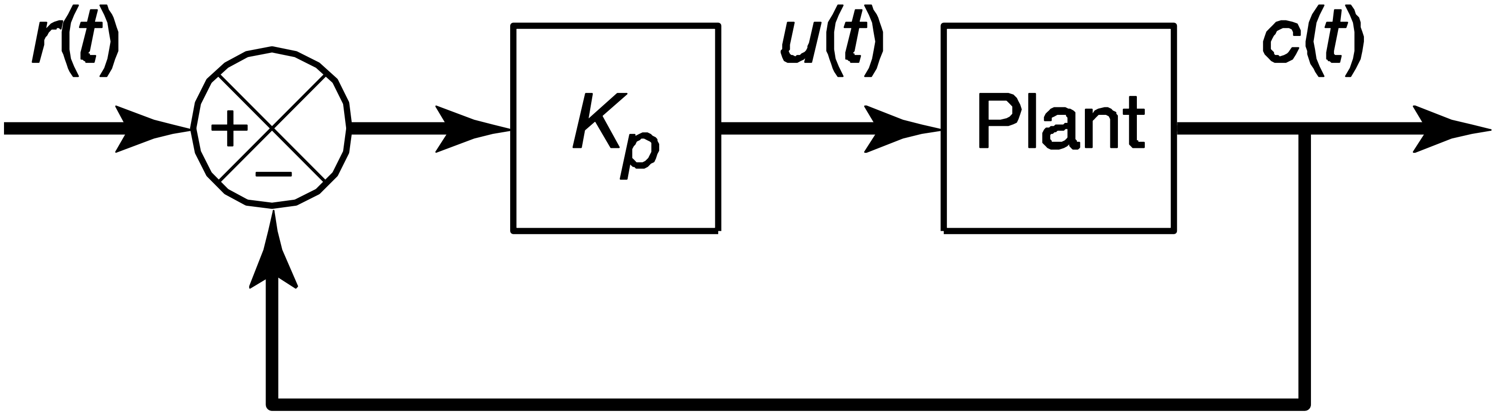
$$G_c(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right)$$

$$= 1.2\frac{T}{L}\left(1 + \frac{1}{2Ls} + 0.5Ls\right)$$

$$= 0.6T\frac{\left(s + \frac{1}{L}\right)^2}{s}$$

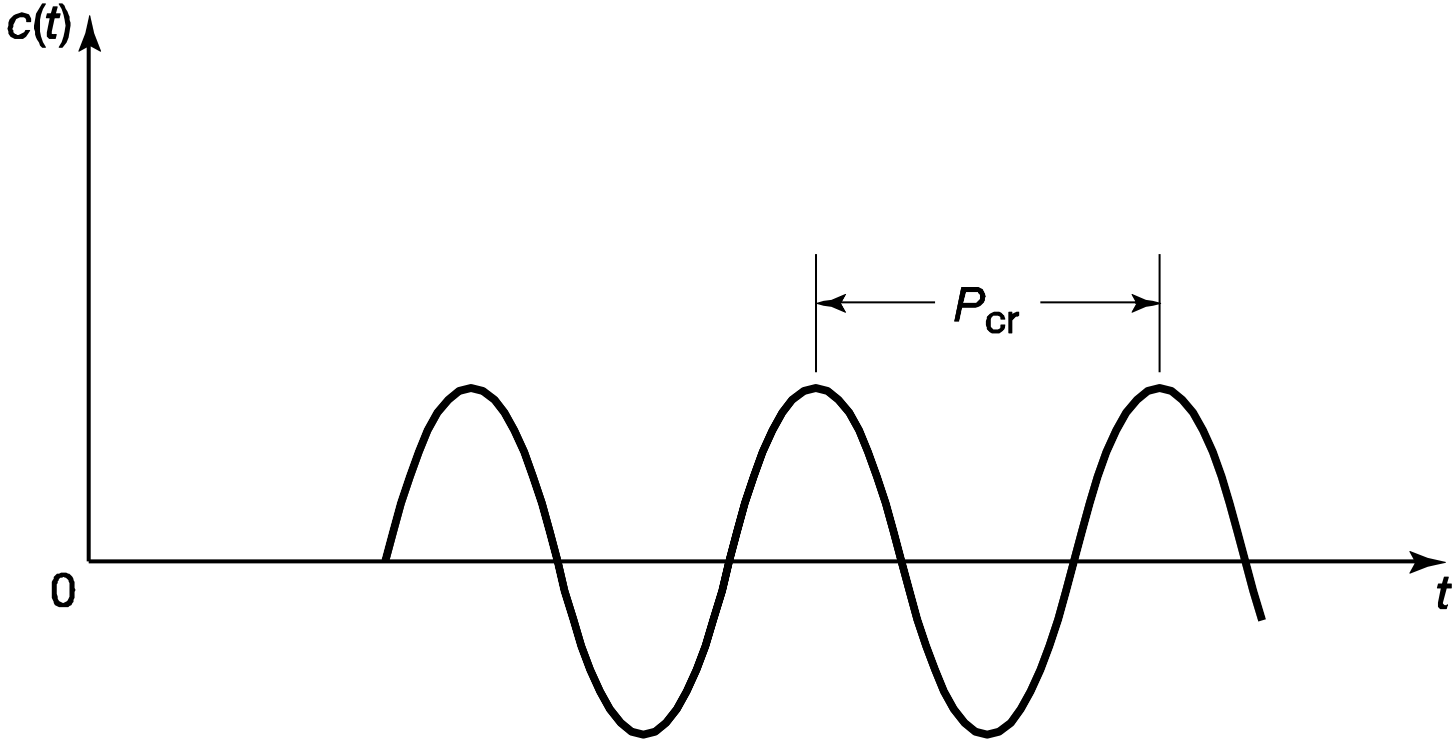
Ziegler-Nichols PID Tuning---Second Method

Use the proportional controller to force sustained oscillations



PID Tuning---Second Method

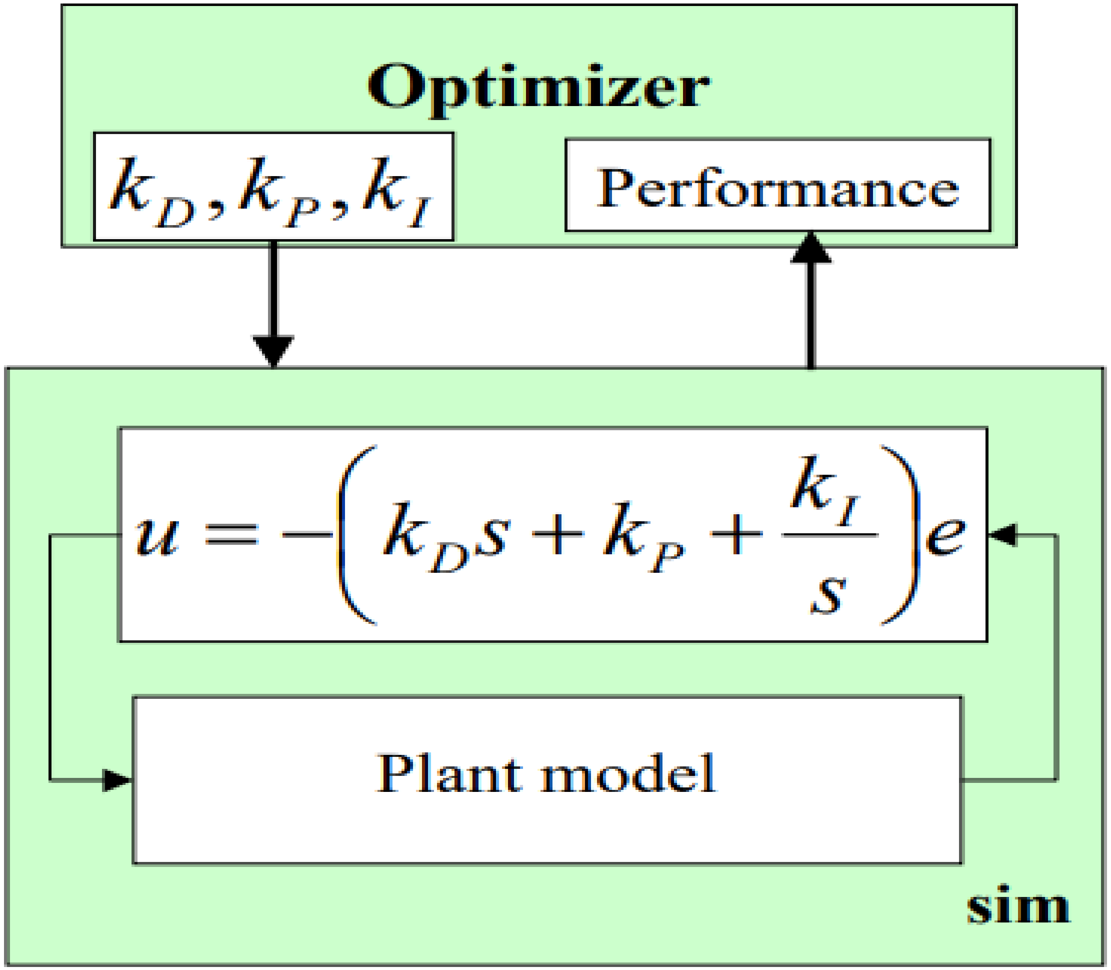
Measure the period of sustained oscillation



PID Tuning

Table 10–2 Ziegler–Nichols Tuning Rule Based on Critical Gain K_{cr} and Critical Period P_{cr} (Second Method)

Type of Controller	K_p	T_i	T_d
P	$0.5K_{cr}$	∞	0
PI	$0.45K_{cr}$	$\frac{1}{1.2}P_{cr}$	0
PID	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$



Optimizer
k_D, k_P, k_I
Performance
$u = -\left(k_D s + k_P + \frac{k_I}{s}\right)e$
Plant model
sim

Transfer Function of PID Controller Tuned Using the Second Method

$$G_c(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right)$$

$$= 0.6K_{cr}\left(1 + \frac{1}{0.5P_{cr}s} + 0.125P_{cr}s\right)$$

$$= 0.075K_{cr}P_{cr}\frac{\left(s + \frac{4}{P_{cr}}\right)^2}{s}$$

EXAMPLE Consider the control system $\frac{c(s)}{U(s)} = \frac{1}{s(s+1)(s+5)}$

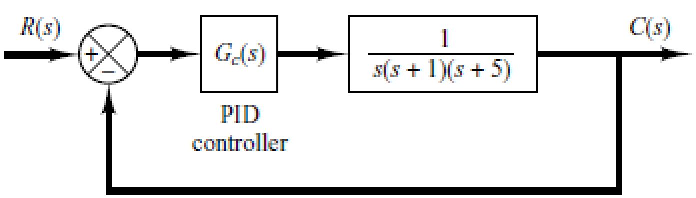
Design a PID controller for the present system, apply a Ziegler–Nichols tuning rule for the determination of the values of parameters and Then obtain a unit-step response curve and check to see if the designed system exhibits approximately 25% maximum overshoot.

$$\frac{C(s)}{R(s)} = \frac{K_p}{s(s+1)(s+5) + K_p}$$

$$s^3 + 6s^2 + 5s + K_p = 0$$

the Routh array becomes as follows:

$$\begin{array}{lcc} s^3 & 1 & 5 \\ s^2 & 6 & K_p \\ s^1 & \dfrac{30 - K_p}{6} & \\ s^0 & K_p & \end{array}$$



$K_{cr}=30$

With gain K_p set equal to $K_{cr}=30$ the characteristic equation becomes

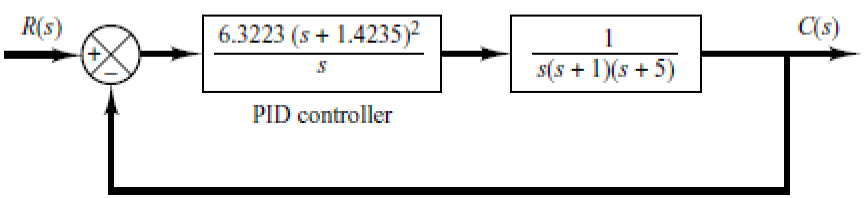
$s^3+6s^2+5s+30=0$

To find the frequency of the sustained oscillation, we substitute s=jw into this characteristic equation as follows:

$(jw)^3+6(jw)^2+5(jw)+30=0$

Gpid(s)= 18 (1 +1/1.405s + 0.35124s)

$6(5-w)^2+jw(5-w^2)=0$



$w=\sqrt{5}$

$P_{cr}=2p/w= 2.8099$

$K_p= 0.6K_{cr}= 18$

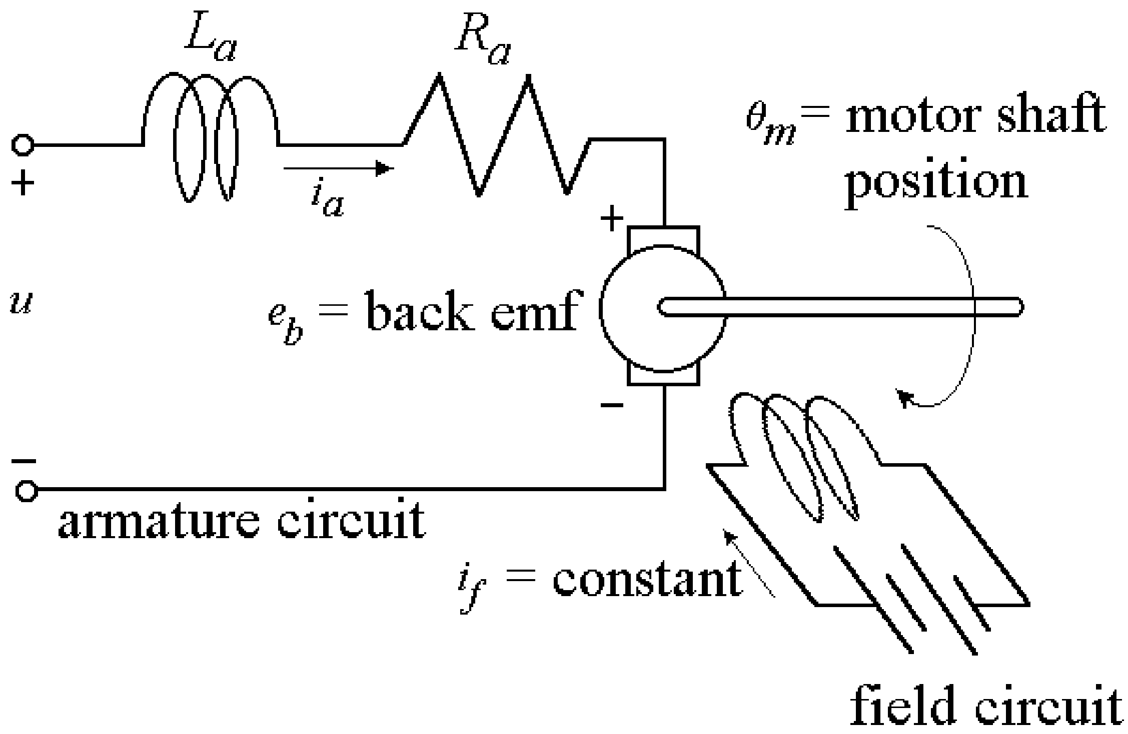
$T_i= 0.5P_{cr}= 1.405$

$T_d= 0.125P_{cr}= 0.35124$

Example 1---PID Controller for DC Motor

- Plant---**Armature-controlled DC motor**; MOTOMATIC system produced by Electro-Craft Corporation
- Design a Type A PID controller and simulate the behavior of the closed-loop system; plot the closed-loop system step response
- Fine tune the controller parameters so that the max overshoot is 25% or less

Modeling the Armature Controlled DC Motor



Transfer Function of the DC Motor System

- Transfer function of the DC motor

$$G_p(s) = \frac{C(s)}{U(s)} = \frac{0.1464}{7.89\times10^{-7}s^3 + 8.25\times10^{-4}s^2 + 0.00172s}$$

where $C(s)$ is the angular displacement of the motor shaft and $U(s)$ is the armature voltage

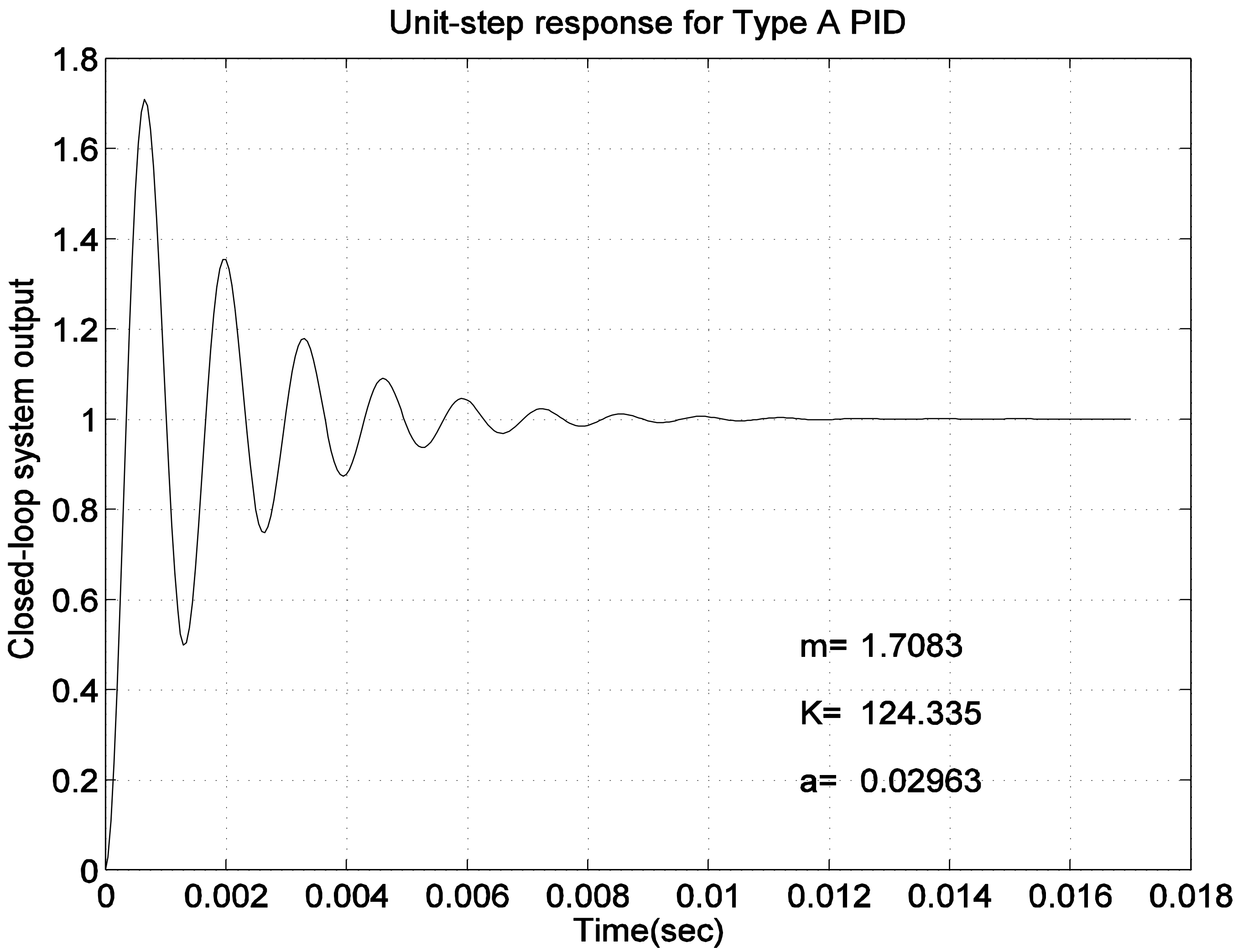
Tuning the Controller Using the Second Method of Ziegler and Nichols

- Determine K_{cr}
- Determine P_{cr}
- Compute the controller parameters

Generating the Step Response

```
t=0:0.00005:.017;
K_cr=12.28; P_cr=135;
K=0.075*K_cr*P_cr; a=4/P_cr;
num1=K*[1 2*a a^2]; den1=[0 1 0];
tf1=tf(num1,den1);
num2=[0 0 0 0.1464];
den2=[7.89e-007 8.25e-004 0.00172 0];
tf2=tf(num2,den2);
tf3=tf1*tf2;
sys=feedback(tf3,1);
y=step(sys,t); m=max(y);
```

Closed-loop System Performance



Example 2 (Based on Ex. 10-3 in Ogata, 2002)

- Use a computational approach to generate an optimal set of the DC motor PID controller's parameters

$$G_c(s) = K\frac{(s+a)^2}{s}$$

- Generate the step response of the closed-loop system

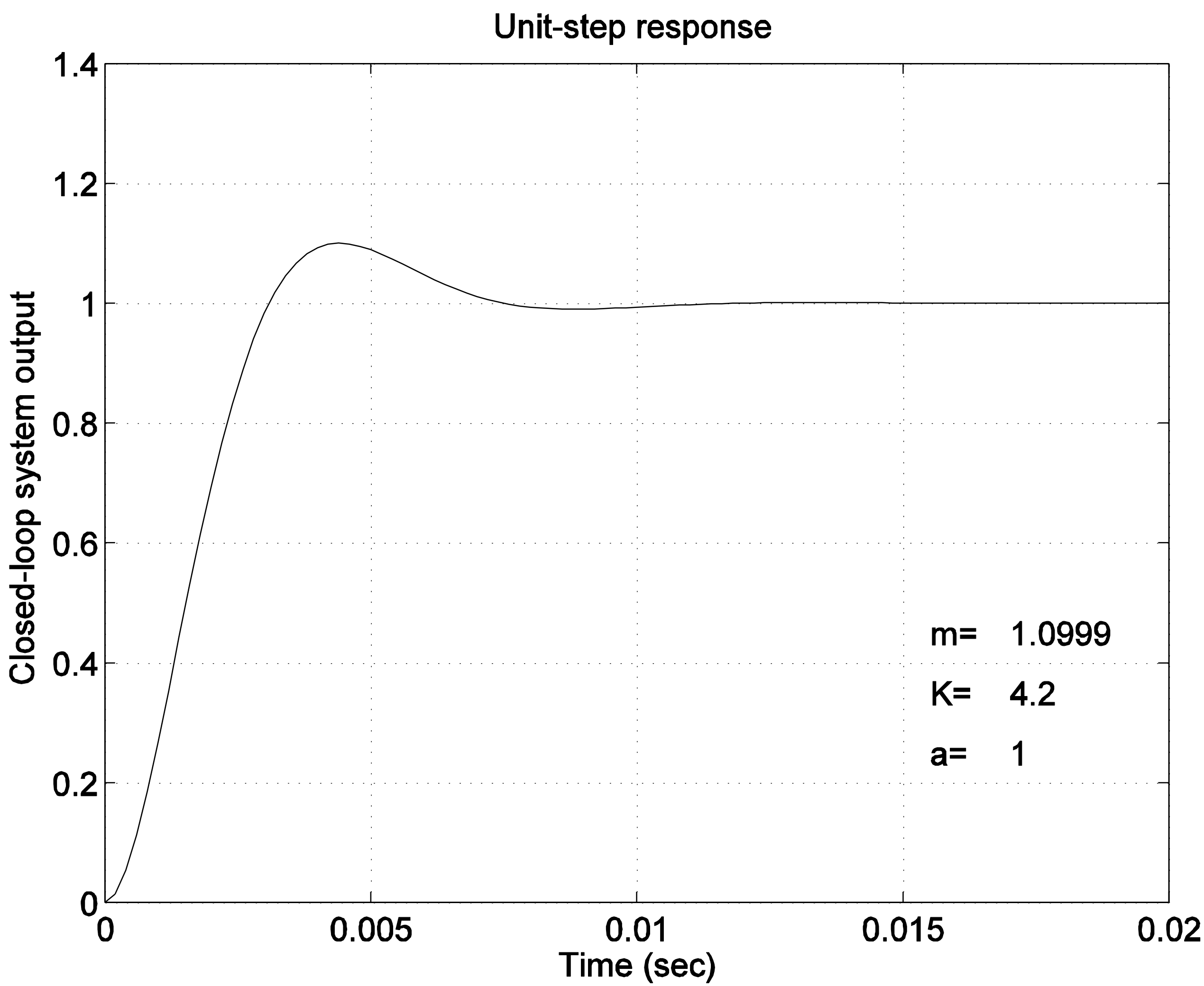
Optimizing PID Parameters

```
t=0:0.0002:0.02;
for K=5:-0.2:2%Outer loop to vary the values of
   %the gain K
    for a=1:-0.01:0.01;%Outer loop to vary the
   %values of the parameter a
        num1=K*[1 2*a a^2]; den1=[0 1 0];
        tf1=tf(num1,den1);
        num2=[0 0 0 0.1464];
        den2=[7.89e-007 8.25e-004 0.00172 0];
        tf2=tf(num2,den2);
        tf3=tf1*tf2;
        sys=feedback(tf3,1);
        y=step(sys,t); m=max(y);
```

Finishing the Optimizing Program

```
if m<1.1 & m>1.05;
            plot(t,y);grid;set(gca,'Fontsize',font)
sol=[K;a;m]
            break % Breaks the inner loop
        end
    end
    if m<1.1 & m>1.05;
        break; %Breaks the outer loop
    end
end
```

Closed-Loop System Performance



Modified PID Control Schemes

- If the reference input is a step, then because of the presence of the derivative term, the controller output will involve an impulse function
- The derivative term also amplifies higher frequency sensor noise
- Replace the pure derivative term with a derivative filter---PIDF controller
- Set-Point Kick---for step reference the PIDF output will involve a sharp pulse function rather than an impulse function

The Derivative Term

- Derivative action is useful for providing a phase lead, to offset phase lag caused by integration term
- Differentiation increases the high-frequency gain
- Pure differentiator is not proper or causal
- 80% of PID controllers in use have the derivative part switched off
- Proper use of the derivative action can increase stability and help maximize the integral gain for better performance

Remedies for Derivative Action---PIDF Controller

- Pure differentiator approximation

$$\frac{T_d s}{1 + \gamma T_d s}$$

where γ is a small parameter, around, 0.1

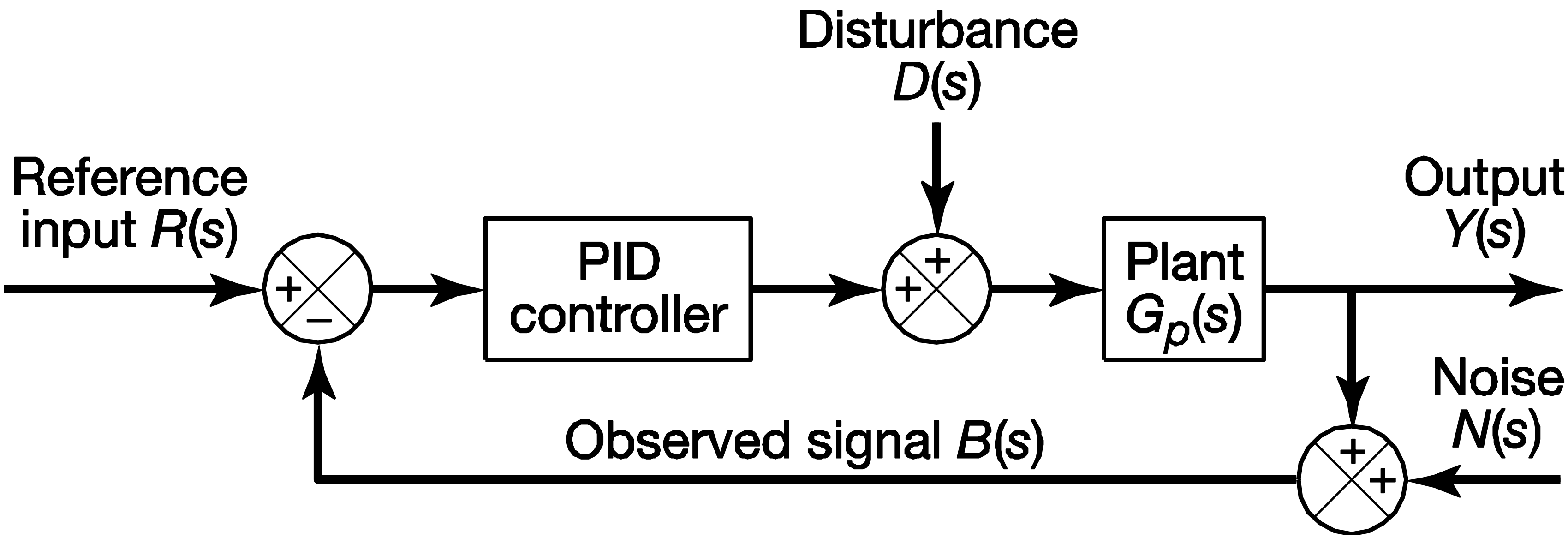
- Pure differentiator cascaded with a first-order low-pass filter

The Set-Point Kick Phenomenon

- If the reference input is a step function, the derivative term will produce an impulse (delta) function in the controller action
- Possible remedy---operate the derivative action only in **the feedback path**; thus differentiation occurs only on the feedback signal and not on the reference signal

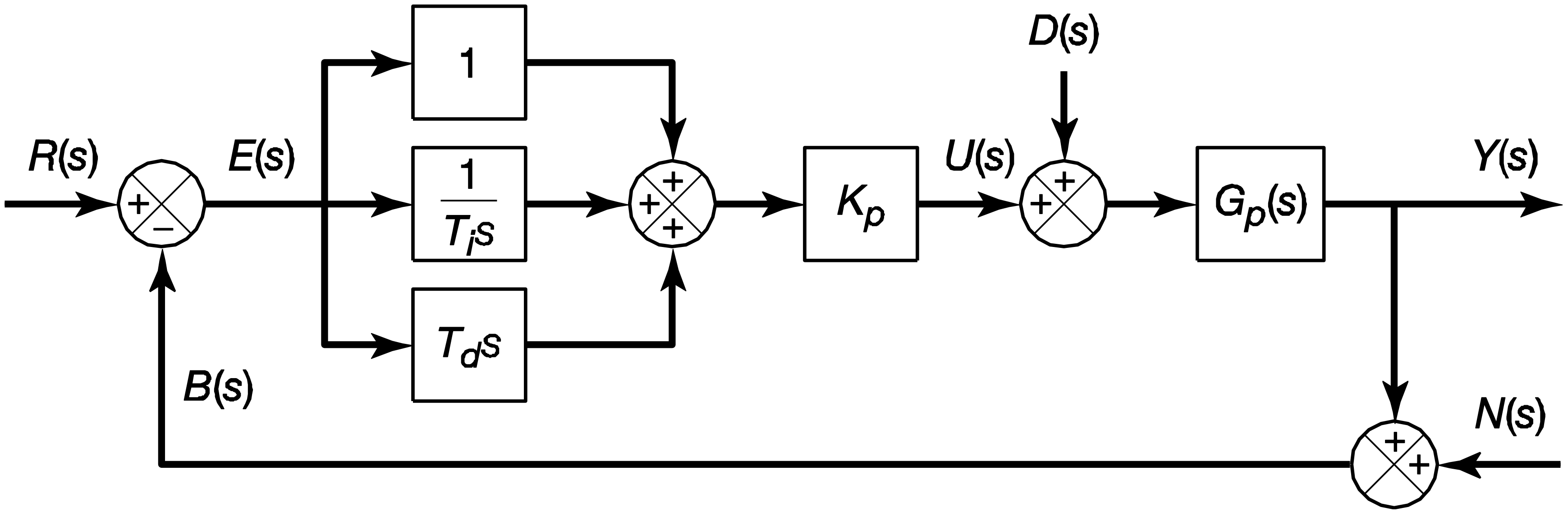
Eliminating the Set-Point Kick

PID controller revisited



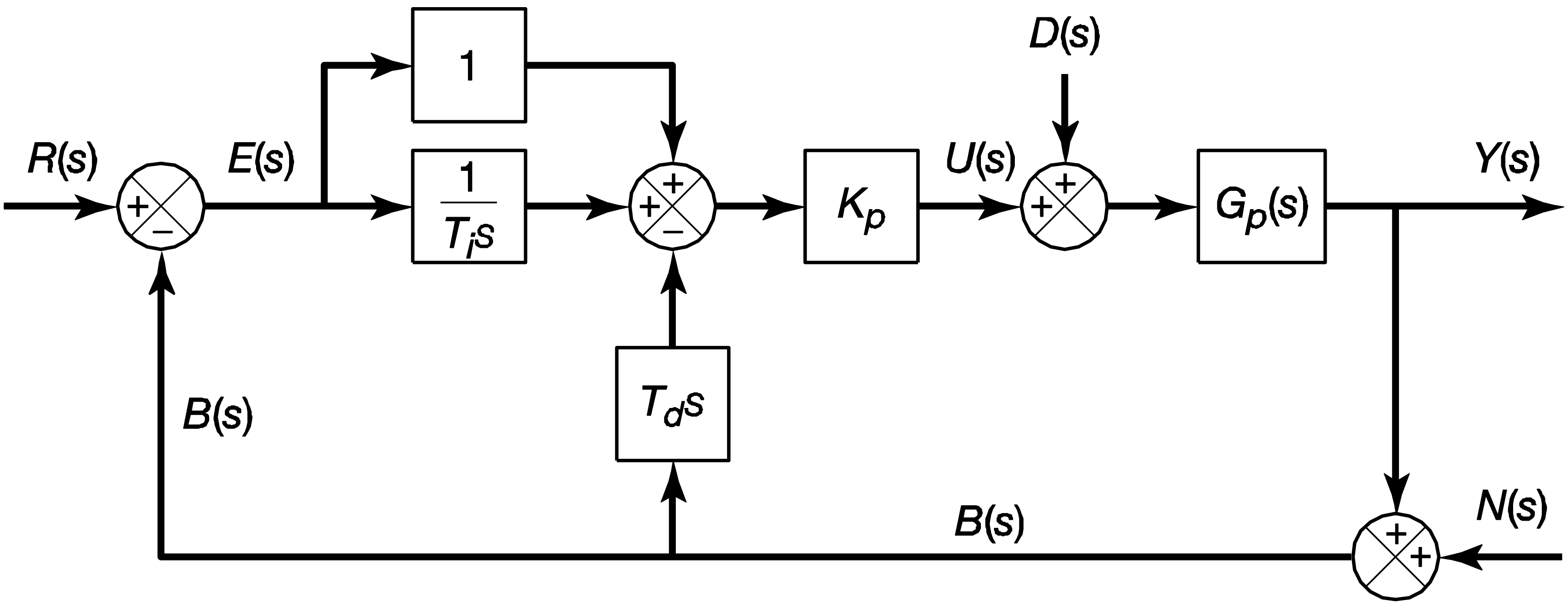
Eliminating the Set-Point Kick---Finding the source of trouble

More detailed view of the PID controller



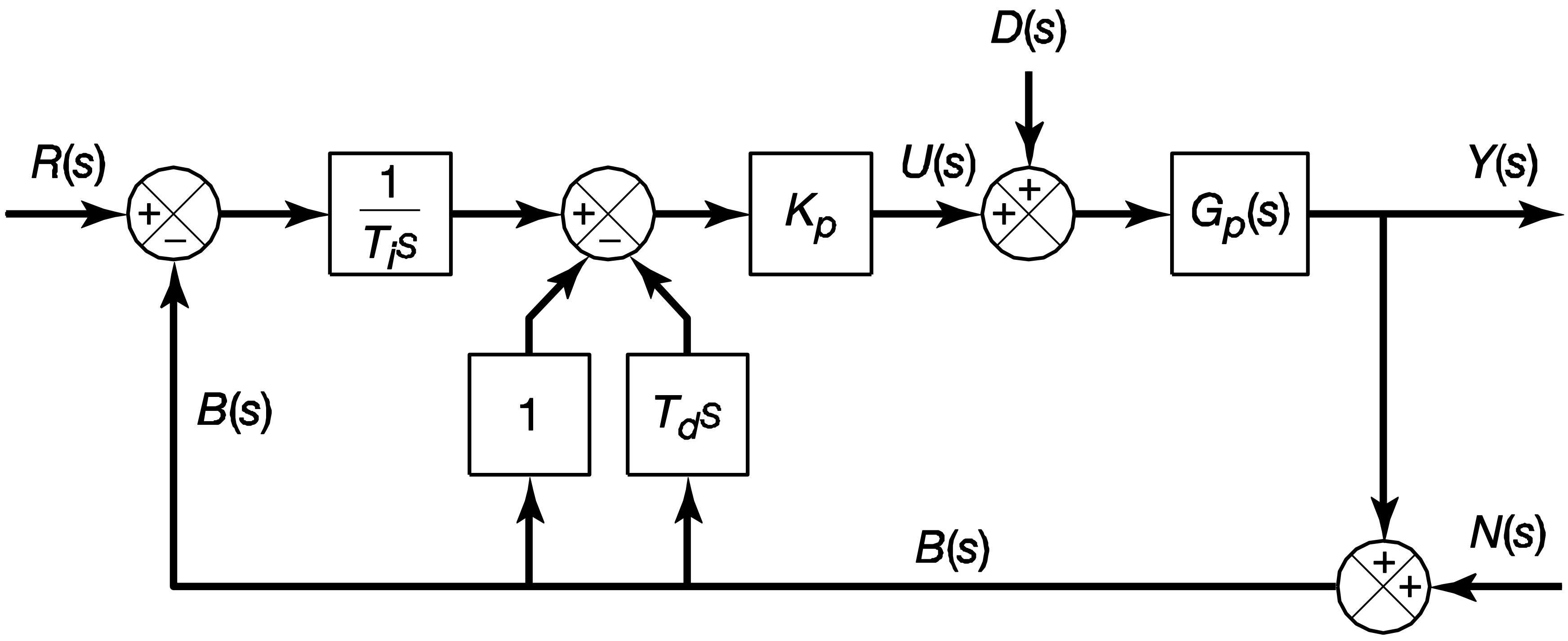
Eliminating the Set-Point Kick---PI-D Control or **Type B PID**

Operate derivative action only in the feedback



I-PD---Moving Proportional and Derivative Action to the Feedback

I-PD control or **Type C PID**



I-PD Equivalent to PID With Input Filter (No Noise)

Closed-loop transfer function $C(s)/R(s)$ of the I-PD-controlled system

$$\frac{C(s)}{R(s)} = \frac{\dfrac{K_p}{T_i s} G_p(s)}{1 + K_p\left(1 + \dfrac{1}{T_i s} + T_d s\right) G_p(s)}$$

PID-Controlled System

- Closed-loop transfer function $C(s)/R(s)$ of the PID-controlled system with input filter

$$\frac{C(s)}{R(s)} = \frac{1}{1 + T_i s + T_i T_d s^2} \frac{K_p \left(1 + \frac{1}{T_i s} + T_d s \right) G_p(s)}{1 + K_p \left(1 + \frac{1}{T_i s} + T_d s \right) G_p(s)}$$

- After manipulations it is the same as the transfer function of the I-PD-controlled closed-loop system

PID, PI-D and I-PD Closed-Loop Transfer Function---No Ref or Noise

In the absence of the reference input and noise signals, the closed-loop transfer function between the disturbance input and the system output is the same for the three types of PID control

$$\frac{Y(s)}{D(s)} = \frac{G_p(s)}{1 + K_p G_p(s)\left(1 + \frac{1}{T_i s} + T_d s\right)}$$

The Three Terms of Proportional-Integral-Derivative (PID) Control

- Proportional term responds immediately to the current tracking error; it cannot achieve the desired setpoint accuracy without an unacceptably large gain. Needs the other terms
- Derivative action reduces transient errors
- Integral term yields zero steady-state error in tracking a constant setpoint. It also rejects constant disturbances

☞ **Proportional-Integral-Derivative (PID)** control provides an efficient solution to many real-world control problems

Industrial PID Controller

- • A box, not an algorithm
- • Auto-tuning functionality: – pre-tune – self-tune
- • Manual/cascade mode switch
- • Bumpless transfer between different modes, setpoint ramp
- • Loop alarms
- • Networked or serial port


Honeywell
1500
1500
MAN AUTO
SET UP
1000
1000

Tune PID Controller Gains

PID tuning is the process of finding the values of proportional, integral, and derivative gains of a PID controller to achieve desired performance and meet design requirements. PID controller tuning appears easy, but finding the set of gains that ensures the best performance of your control system is a complex task. Traditionally, PID controllers are tuned either manually or using rule-based methods. Manual tuning methods are iterative and time-consuming, and if used on hardware, they can cause damage. Rule-based methods also have serious limitations: they do not support certain types of plant models, such as unstable plants, high-order plants, or plants with little or no time delay. You can automatically tune PID controllers to achieve the optimal system design and to meet design requirements, even for plant models that traditional rule-based methods cannot handle well.

Atomated PID tuning workflow involves

- ☐ Identifying plant model from input-output test data
- ☐ Modeling PID controllers in MATLAB using PID objects or in Simulink using PID Controller blocks
- ☐ Automatically tuning PID controller gains and fine-tune your design interactively
- ☐Tuning multiple controllers in batch mode
- ☐Tuning single-input single-output PID controllers as well as multiloop PID controller architectures

we shall first discuss about the PID controller implementation with pneumatic and electronic components and then discuss about the different algorithms those can be used for digital implementation of PID controllers. But before that, we shall discuss about the different schemes of implementation of bumpless transfer and antiwindup actions. These two issues were described in the last lesson. We shall now see, how they can be implemented in actual practice.

we shall first discuss about the PID controller implementation with

1. pneumatic
2. electronic components
3. digital implementation of PID controllers.

But before that, we shall discuss about the different schemes of implementation of bumpless transfer and antiwindup actions. These two issues were described in the last lesson. We shall now see, how they can be implemented in actual practice.

Bumpless Transfer

It is quite normal to set up some processes using manual control initially, and once the process is close to normal operating point, the control is transferred to automatic mode through auto/manual switch. In such cases, in order to avoid any jerk in the process the controller output immediately after the changeover should be identical to the output set in the manual mode. This can be achieved by forcing the integral output at the instant of transfer to balance the proportional and derivative outputs against the previous manual output; i.e.

Integral output = {(previous manual) – (proportional + derivative) output}.

Another way to transfer from Auto to Manual mode in a bumpless manner, the set point may be made equal to the present value of the process variable and then slowly changing the set point to its desired value.

The above features can be easily be implemented if a digital computer is used as a controller.

This provision eliminates the chance of the process receiving sudden jolt during transfer

Prevention of Integration Windup

If there is a sudden large change in set point, the error between the set point and the process output will suddenly shoot up and the integrator output due to this error will build up with time.

As a result, the controller output may exceed the saturation limit of the actuator. This windup, unless prevented may cause continuous oscillation of the process

There exits several methods through which integration windup can be prevented. Before we go to the actual methods, let us consider the input-output characteristics of an actuator as shown in Fig. 1.

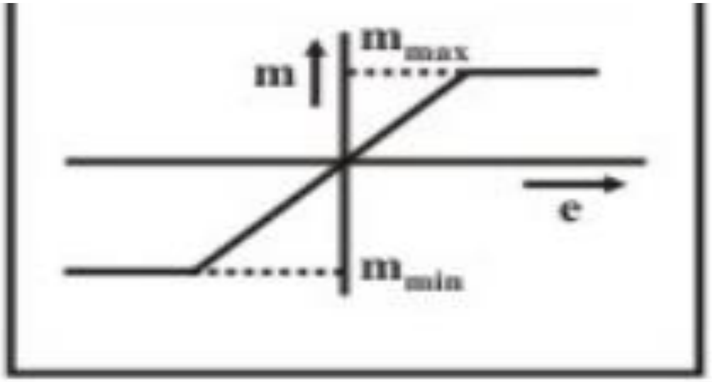


Fig. 1 Typical actuator characteristics

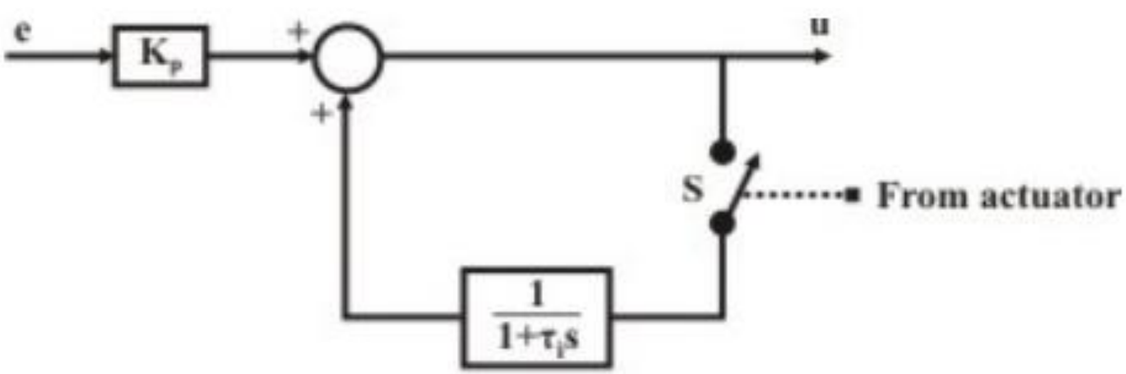


Fig. 2 Scheme for Anti integration windup

Pneumatic Controller

- It has been already mentioned that the early days PID controllers were all pneumatic type. The advantage of pneumatic controllers is its ruggedness, while its major limitation is its slow response. Besides it requires clean and constant pressure air supply. The major components of a pneumatic controller are bellows, flapper nozzle amplifier, air relay and restrictors (valves). The integral and derivative actions are generated by controlling the passage of air flow through restrictors to the bellows.

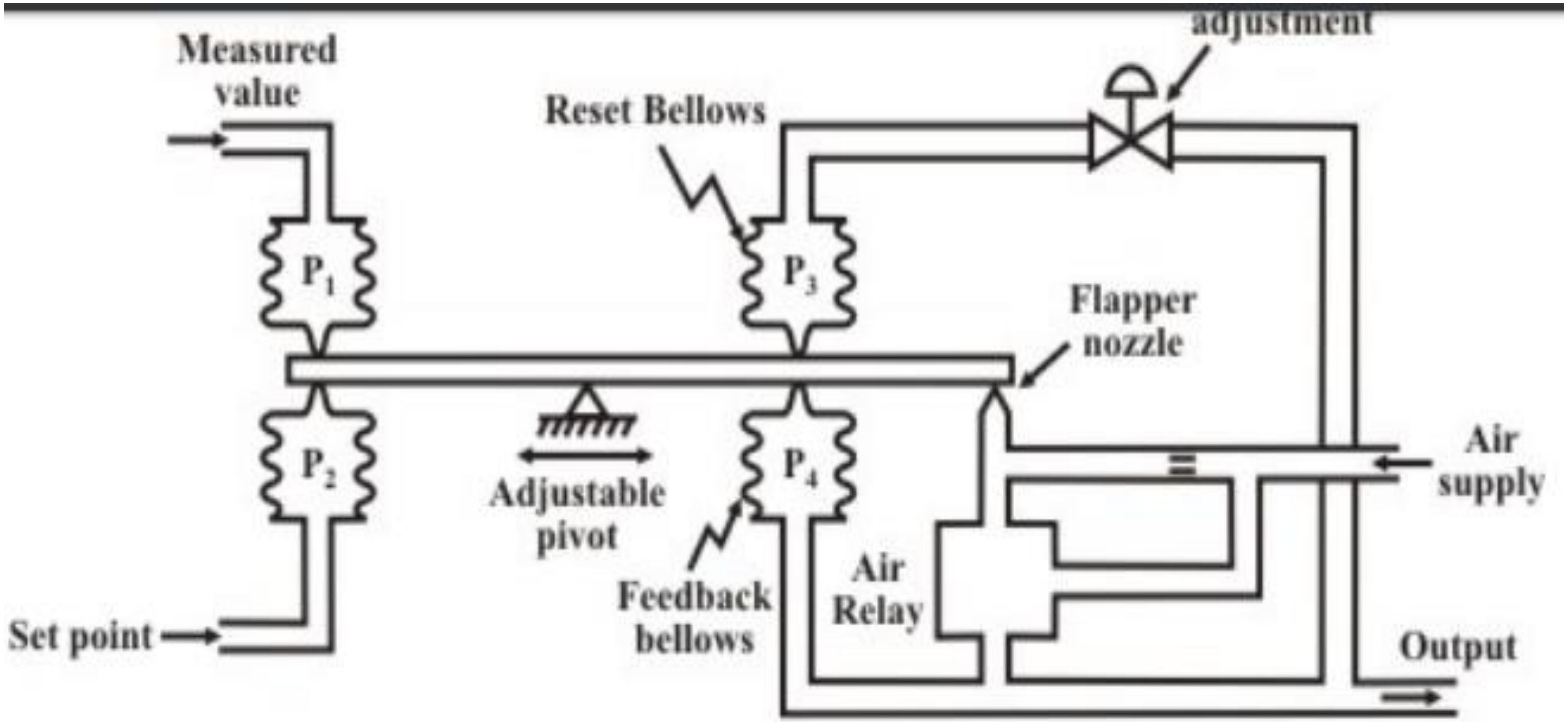


Fig. 4 A Pneumatic PI controller

Electronic PID Controllers

- Electronic PID controllers can be obtained using operational amplifiers and passive components like resistors and capacitors. A typical scheme is shown in Fig. 5. With little calculations, it can be shown that the circuit is capable of delivering the PID actions as:

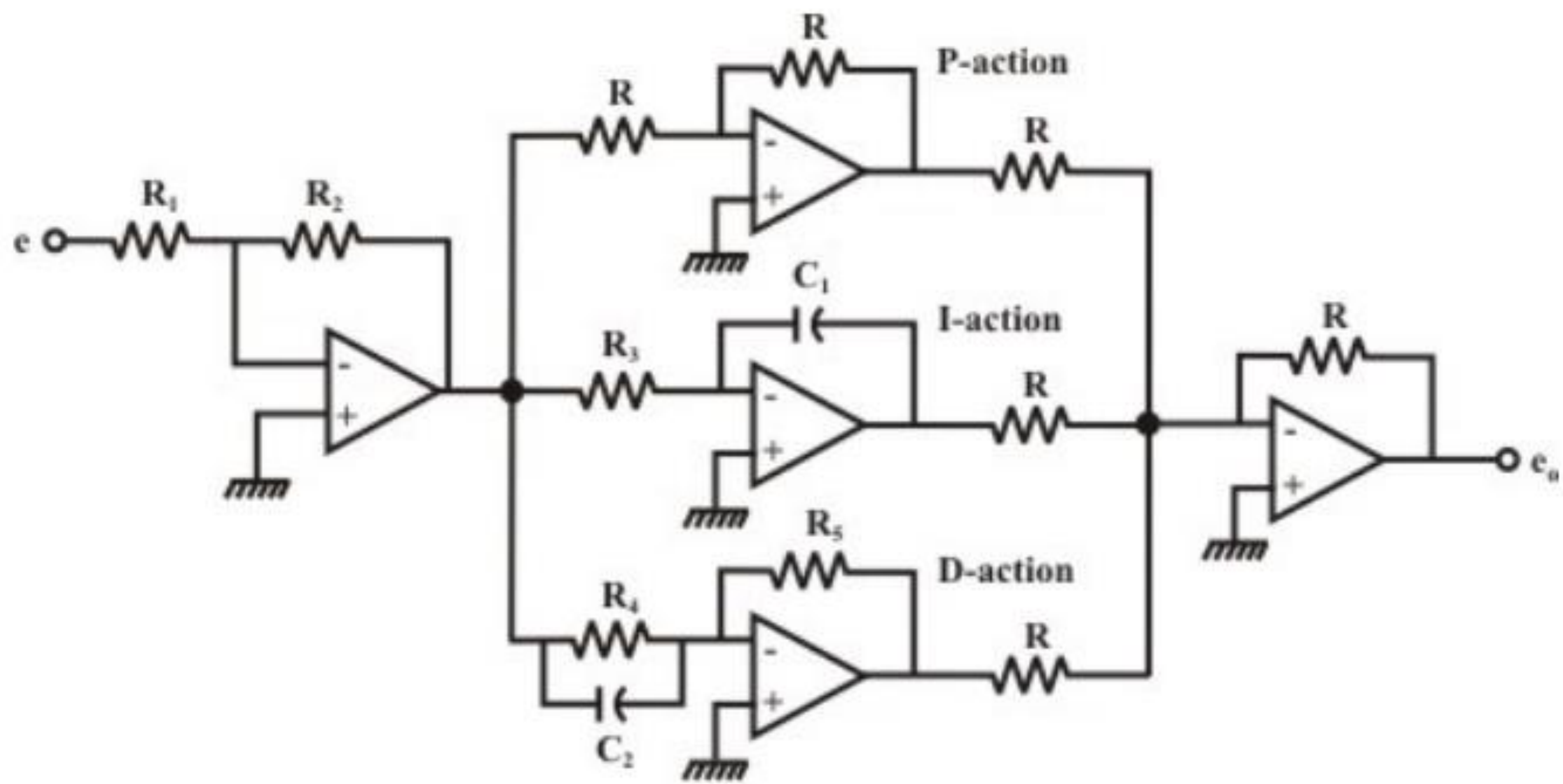


Fig. 5 Electronic PID controller

Digital P-I-D Control

- In the digital control mode, the error signal is first sampled and the controller output is computed numerically through a digital processor.

Review Questions

- **1. What problem is envisaged when a controller is switched from manual control mode to auto mode? Suggest a scheme for overcoming this problem.**
- **2. Explain how the saturation of the actuator affects the performance of a PID controller. Suggest one scheme for overcoming the problem. Explain its operation.**
- **3. Draw the schematic diagram of an electronic PI controller. How the proportional and integral constants can be adjusted.**

- **4. What are the major drawbacks of a pneumatic controller?**
- **5. Explain how a continuous time PID control law can be discretised using velocity algorithm.**
- **6. What are the advantages of velocity algorithm, compared to the position algorithm for digital implementation of PID controllers?**

7.What is a ‘controller’? Where are its application areas?

A controller is an instrument used for controlling a process variable (measurement). Its continuously monitors the error signal and gives a corrective output to the final control element.

8.What is a ‘controller’? Where are its application areas?

A controller is an instrument used for controlling a process variable (measurement). Its continuously monitors the error signal and gives a corrective output to the final control element.

9.Explain what is ‘direct action’ and ‘reverse action’ on a controller?

Direct action :

In a direct acting controller, the output increases when the process measurement (variable) increases.

Reverse action :

In a reverse action controller, the output decreases when the process measurement (variable) increases.

10.What is a gap controller?

A controller whose output changes from minimum to maximum (on-off) and vice-versa when the error signal (deviation) exceeds the set gap depending on the controller action.

11.What is a 'proportional band'? Explain with an example?

It is the range in percentage for which the controller output changes proportionally from minimum to maximum and vice-versa when the measurement deviates from the setpoint.

For example: A controller set at 50% proportional band

The controller output changes from minimum to maximum and vice-versa when the measurement exceeds 25% either side of the setpoint depending on the controller action.

12.What is a 'gain'? Write the relation (formula) between a gain and proportional band?

A controller 'gain' is inversely proportional to its proportional band.

g = (1/p)*100

g = gain

P = proportional band

13.What is a 'reset action'? Explain with an example?

Reset action in a controller is the integration of the proportional action by the set period. The reset action repeats the proportional action's output per the reset time set, until the error signal becomes zero or the output gets saturated.

For example :if the reset action is set for 30 sec.

For a 0.5 volt correction output ;by the proportional action will be repeated by the reset action every 30 secs., until the error signal becomes zero or the output gets saturated.

14.What is a ‘batch’ facility on a controller?

Whenever a deviation persists for a long time the controller output saturates at the maximum of minimum output (-2.5 VDC or + 12.5 VDC/0 kPa or 140 kPa) depending on the controller action. In a normal controller when the process reverts to normal the output takes its time to come into control range.

In a batch controller the output reverts to the control limit (0VDC OR 10.00V DC 20 kPa or 100 kPa) as soon as the deviation enters the batch limits

15.What will be the output (increases or decreases) of a direct action controller when the process goes above the setpoint?

In a direct acting controller, the controller output increases when the process (measurement) goes above the setpoint)

16.What will be the output of a reverse acting controller when the process changes from 50% to 75% where the proportional band is set at 50%, setpoint is set at 50%?

The controller output will be zero.

17.What is a ‘bump-less transfer’ in a controller’s auto/manual change over?

‘Bump-less transfer’ is to eliminate the change in the controller’s output when the controller is changed from auto to manual control and vice-versa.

18.Explain how to change a controller from auto to manual and vise-versa?

Pneumatic controllers :

While taking the controller from auto to manual, the manual output is to be balanced to the auto output and then transfer the auto-manual switch to manual.

While changing the controller from manual to auto, the controller setpoint is matched to the manual output and then auto-manual switch is transferred.

Electronic controllers :

Auto to manual control may be transferred directly as the electronic circuit keeps the auto and manual output matched.

But while changing the controller from manual to auto, the controller setpoint is to match to the process variable and then auto-manual switch is transferred.

19.What type of controller (P, PI, PID) is preferred on the following process control loops?

Pressure :

If the load change is minimum, then a proportional controller is suitable. If a frequent load change is expected then a Proportional + Integral controller is preferred.

Level :

Normally a proportional controller is preferred.

Flow :

Proportional + Integral controller is preferred

Temperature :

Proportional + Integral + Derivative controller is preferred

20.Why is there a direct and reverse action on a controller when the control valves are already having direct (air fail to close)/reverse(air fail to open) actions"?

The type of control valve action requirement on a process line, depends on the protection required on the upstream or downstream of the control valve incase of an air failure.

Depending on the control valve action, the controller action has to be set to control the process.

Summary

- PID control---most widely used control strategy today
- Over 90% of control loops employ PID control, often the derivative gain set to zero (PI control)
- The three terms are intuitive---a non-specialist can grasp the essentials of the PID controller's action. It does not require the operator to be familiar with advanced math to use PID controllers
- Engineers prefer PID controls over untested solutions