Basic Properties of Feedback

Part D: The Classical Three- Term Controllers

Basic Operations of a Feedback Control

Think of what goes on in domestic hot water thermostat:

- The temperature of the water is measured.
- Comparison of the measured and the required values provides an error, *e.g.* “*too hot’ or ‘too cold*’.
- On the basis of error, a control algorithm decides what to do.

  → *Such an algorithm might be:*

  - *If the temperature is too high then turn the heater off.*
  - *If it is too low then turn the heater on*
- The adjustment chosen by the control algorithm is applied to some adjustable variable, *such as the power input to the water heater.*

Feedback Control Properties

- ***A feedback control system seeks to bring the measured quantity to its required value or set-point.***
- The control system *does not need to know why the measured value is not currently what is required*, only that is so.
- There are two *possible causes of such a disparity*:
  - The system has been disturbed.
  - The set point has changed. In the absence of external disturbance, a change in set point will introduce an error. The control system will act until the measured quantity reach its new set point.

The PID Algorithm

- The **PID algorithm** is the *most popular feedback controller algorithm used*. It is a robust easily understood algorithm that can provide *excellent control performance* despite the varied dynamic characteristics of processes.

- As the name suggests, the **PID algorithm** consists of ***three basic modes***:

the *Proportional* mode,

the *Integral* mode

& the *Derivative* mode.

P, PI or PID Controller

- When utilizing the PID algorithm, it is necessary to decide which modes are to be used (P, I or D) and then ***specify the parameters (or settings) for each mode used***.

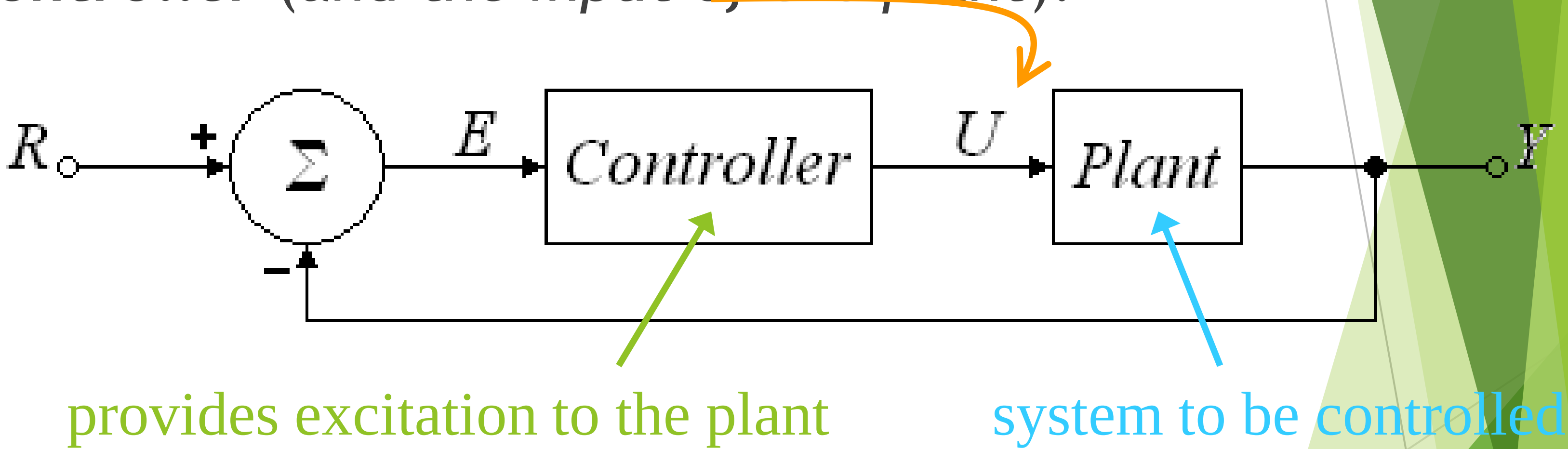
- Generally, three basic algorithms are used: ***P, PI or PID***.

- Controllers are designed to eliminate the need for continuous operator attention.

$\rightarrow$ *Cruise control in a car and a house thermostat* are common examples of *how controllers are used to automatically adjust some variable to hold a measurement (or* ***process variable****) to a desired variable (or* ***set-point****)*

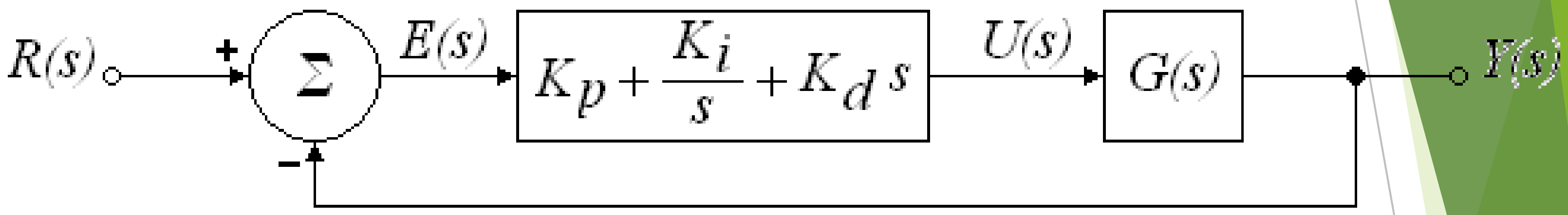
Controller Output

- The *variable being controlled* is the ***output of the controller*** (and the *input of the plant*):



- The output of the controller will change in response to a change in measurement or set-point (that said a change in the tracking error)

PID Controller



- In the *s*-domain, the PID controller may be represented as:

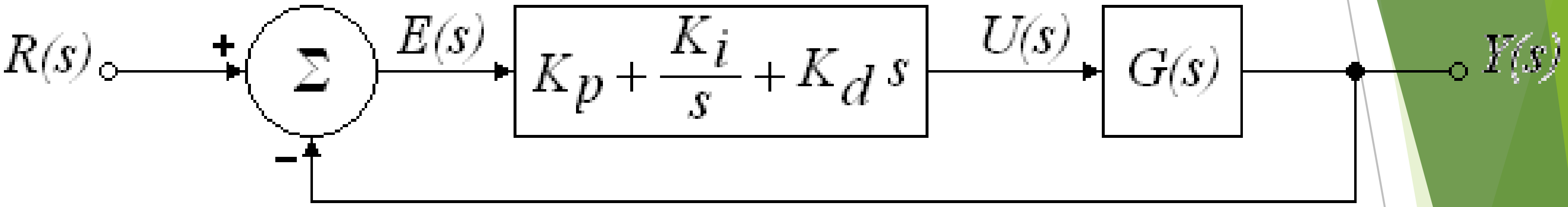
$$U(s) = \left(K_p + \frac{K_i}{s} + K_d s \right) E(s)$$

- In the time domain:

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

proportional gain integral gain derivative gain

PID Controller



- In the time domain:

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

- The signal *u(t)* will be sent to the plant, and a new output *y(t)* will be obtained. This new output *y(t)* will be sent back to the sensor again to find the new error signal *e(t)*. The controllers takes this new error signal and computes its derivative and its integral gain. This process goes on and on.

Definitions

- In the time domain:

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

$$= K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(t)dt + T_d \frac{de(t)}{dt} \right)$$

integral time constant

derivative time constant

$$\text{where} \quad T_i = \frac{K_p}{K_i}, \qquad T_d = \frac{K_d}{K_i}$$

derivative gain

proportional gain

integral gain

Controller Effects

- A **proportional controller** (P) ***reduces error responses to disturbances***, but *still allows a steady-state error.*
- When the controller includes a term proportional to the integral of the error (I), then the ***steady state error to a constant input is eliminated***, although typically *at the cost of deterioration in the dynamic response*.
- A **derivative control** typically ***makes the system better damped and more stable***.

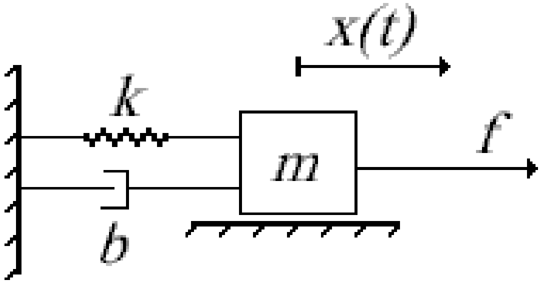
Closed-loop Response

	Rise time	Maximum overshoot	Settling time	Steady-state error
P	**Decrease**	Increase	Small change	**Decrease**
I	**Decrease**	Increase	Increase	**Eliminate**
D	Small change	**Decrease**	**Decrease**	Small change

- Note that these correlations may not be exactly accurate, because P, I and D gains are dependent of each other.

Example problem of PID

- Suppose we have a simple mass, spring, damper problem.



- The dynamic model is such as:

$$m\ddot{x} + b\dot{x} + kx = f$$

- Taking the Laplace Transform, we obtain:

$$ms^2X(s) + bsX(s) + kX(s) = F(s)$$

- The Transfer function is then given by:

$$\frac{X(s)}{F(s)} = \frac{1}{ms^2 + bs + k}$$

Example problem (cont'd)

- Let

$$m = 1kg,\ b = 10N.s/m,\ k = 20N/m,\ f = 1N$$

- By plugging these values in the transfer function:

$$\frac{X(s)}{F(s)} = \frac{1}{s^2 + 10s + 20}$$

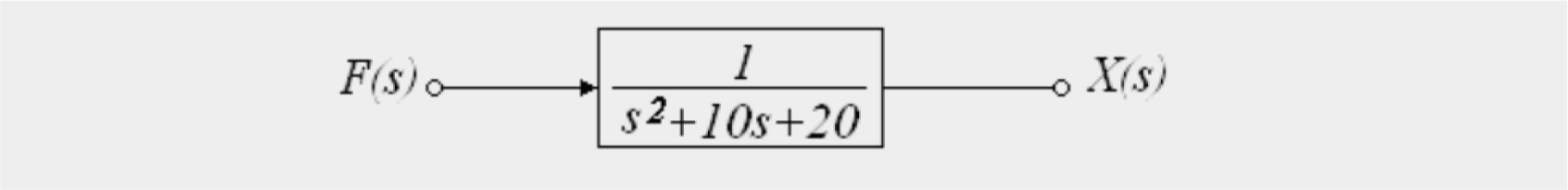
- The goal of this problem is to show you how each of $K_p, K_i\ and\ K_d$ contribute to obtain:

fast rise time,

minimum overshoot,

no steady-state error.

Ex (cont'd): No controller



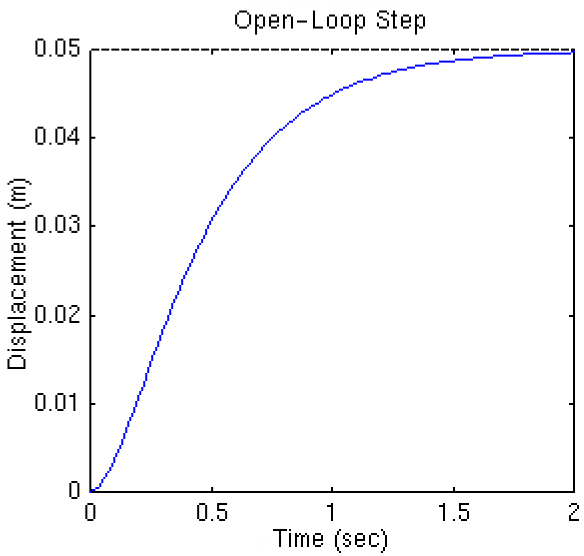
- The (***open***) loop transfer function is given by:

$$\frac{X(s)}{F(s)} = \frac{1}{s^2 + 10s + 20}$$

- The steady-state value for the output is:

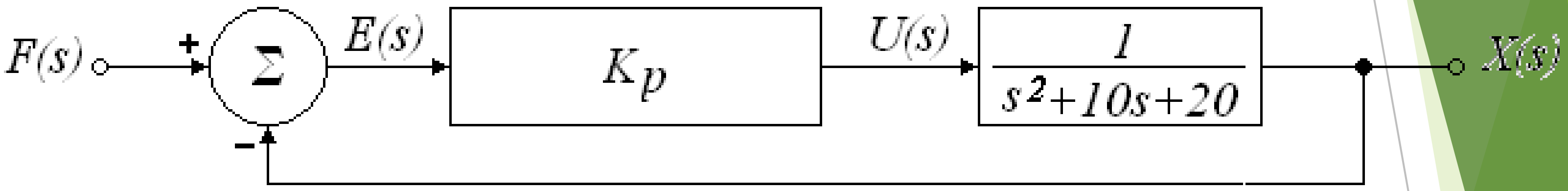
$$x_{ss} = \lim_{t \to \infty} x(t) = \lim_{s \to 0} sX(s) = \lim_{s \to 0} sF(s)\frac{X(s)}{F(s)} = \frac{1}{20}$$

Ex (cont'd): Open-loop step response



- 1/20=0.05 is the *final value* of the output to an *unit* step input.
- This corresponds to a **steady-state error of 95%, *quite large!***
- The ***settling time is about 1.5 sec***.

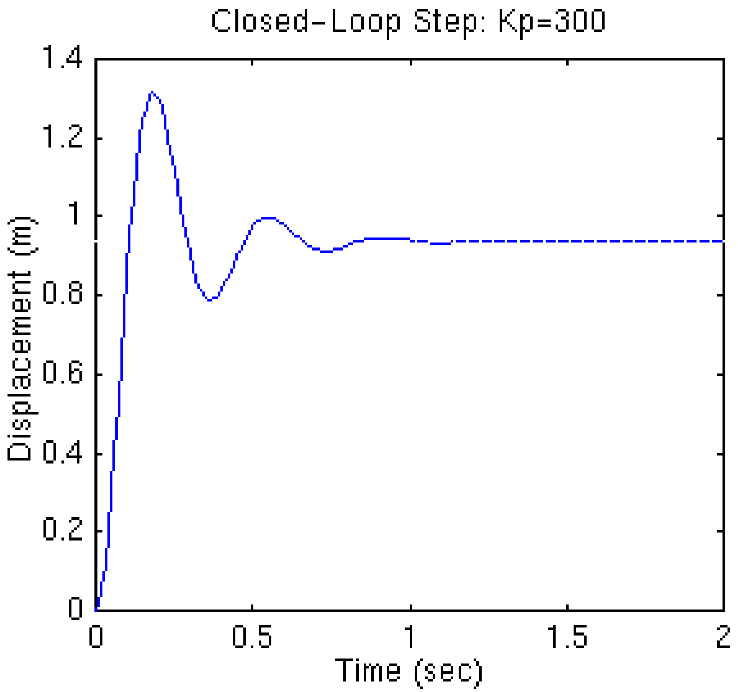
Ex (cont'd): Proportional Controller



- The closed loop transfer function is given by:

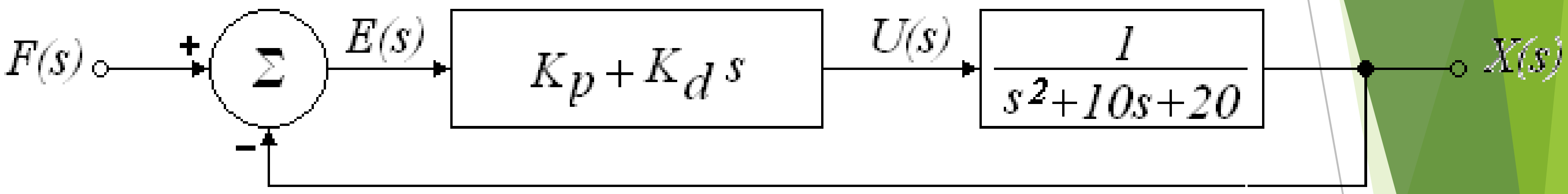
$$\frac{X(s)}{F(s)} = \frac{\dfrac{K_p}{s^2+10s+20}}{1+\dfrac{K_p}{s^2+10s+20}} = \frac{K_p}{s^2+10s+(20+K_p)}$$

Ex (cont'd): Proportional control



- Let $K_p = 300$
- The above plot shows that the ***proportional controller reduced both the rise time and the steady-state error***, *increased the overshoot*, and *decreased the settling time by small amount*.

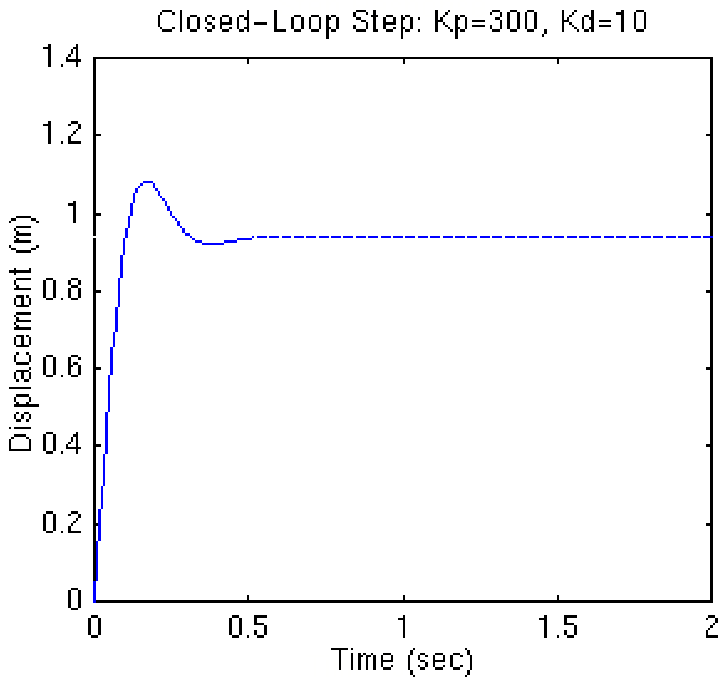
Ex (cont'd): PD Controller



- The closed loop transfer function is given by:

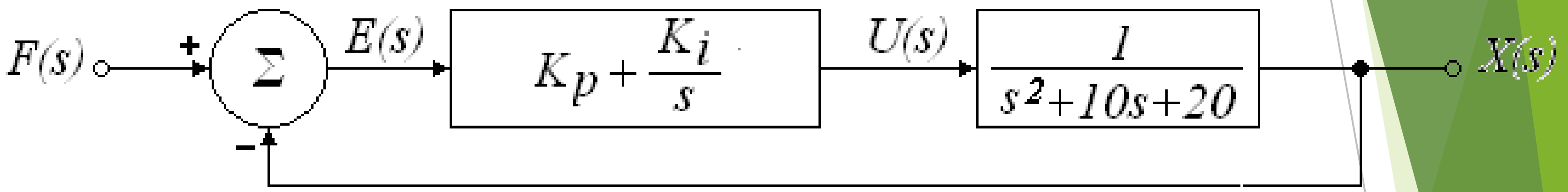
$$\frac{X(s)}{F(s)} = \frac{\dfrac{K_p + K_d s}{s^2 + 10s + 20}}{1 + \dfrac{K_p + K_d s}{s^2 + 10s + 20}} = \frac{K_p + K_d s}{s^2 + (10 + K_d)s + (20 + K_p)}$$

Ex (cont'd): PD control



- Let $K_p = 300, \ K_d = 10$
- This plot shows that the ***proportional derivative controller reduced both the overshoot and the settling time***, and *had small effect on the rise time and the steady-state error*.

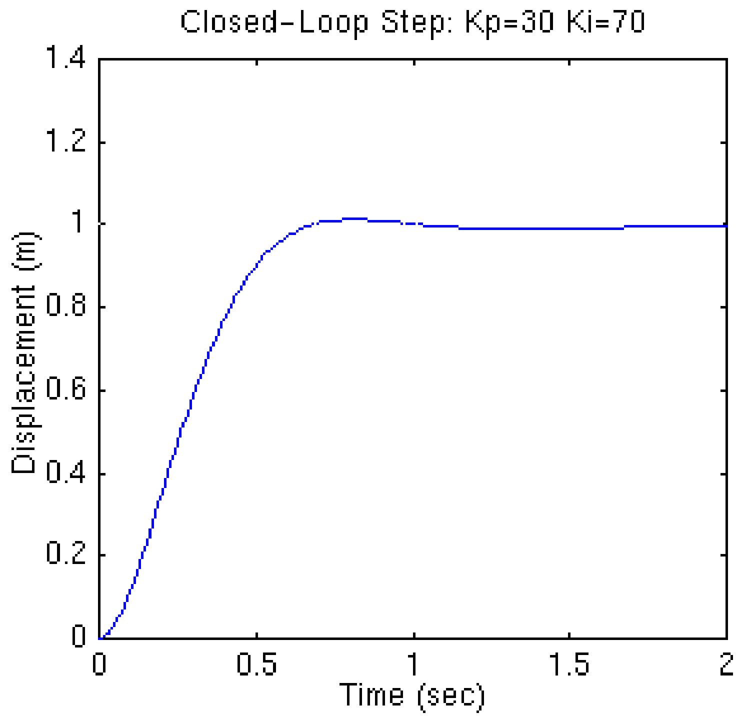
Ex (cont'd): PI Controller



- The closed loop transfer function is given by:

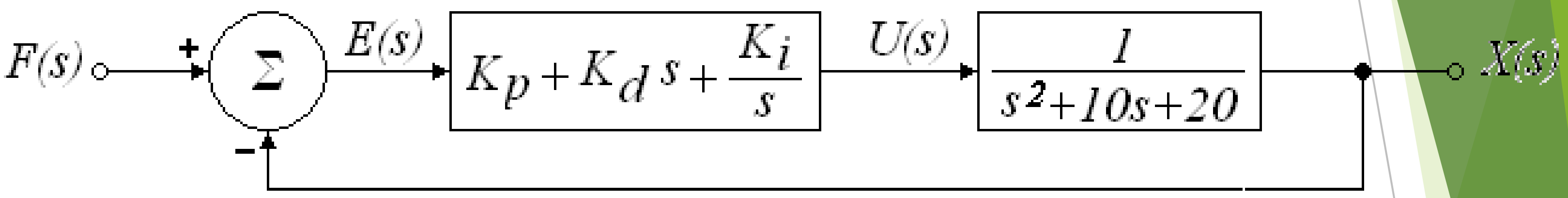
$$\frac{X(s)}{F(s)} = \frac{\dfrac{K_p + K_i / s}{s^2 + 10s + 20}}{1 + \dfrac{K_p + K_i / s}{s^2 + 10s + 20}} = \frac{K_p s + K_i}{s^3 + 10s^2 + (20 + K_p)s + K_i}$$

Ex (cont'd): PI Controller



- Let $K_p = 30, \ K_i = 70$
- We have reduced the proportional gain because the integral controller also ***reduces the rise time and increases the overshoot*** as the proportional controller does (double effect).
- The above response shows that the ***integral controller eliminated the steady-state error***.

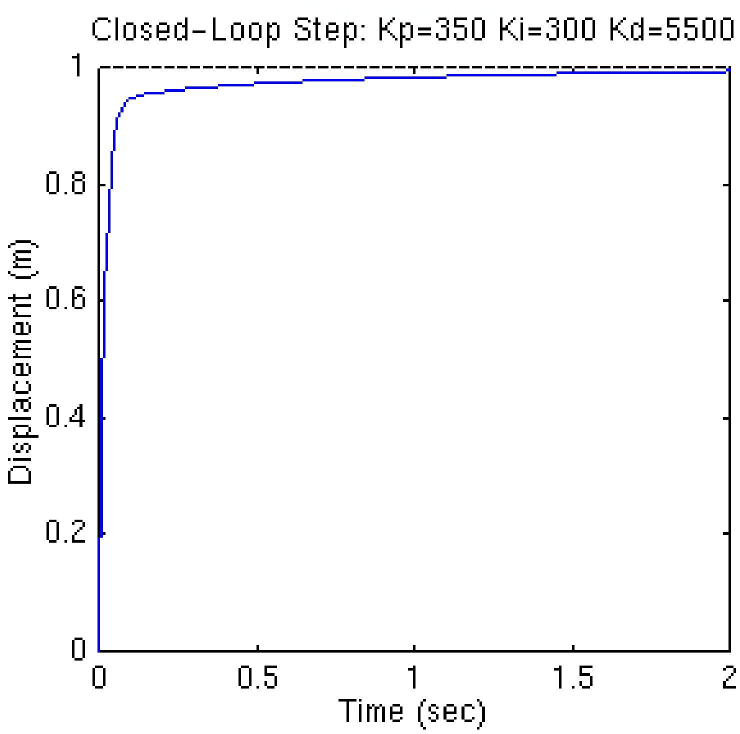
Ex (cont'd): PID Controller



- The closed loop transfer function is given by:

$$\frac{X(s)}{F(s)}=\frac{\dfrac{K_p+K_d s+K_i/s}{s^2+10s+20}}{1+\dfrac{K_p+K_d s+K_i/s}{s^2+10s+20}}=\frac{K_d s^2+K_p s+K_i}{s^3+(10+K_d)s^2+(20+K_p)s+K_i}$$

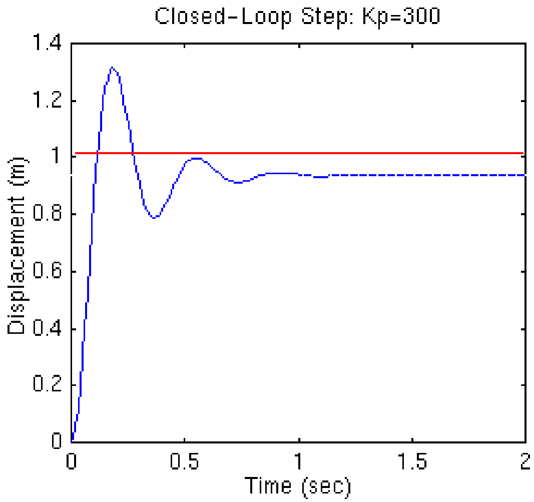
Ex (cont'd): PID Controller



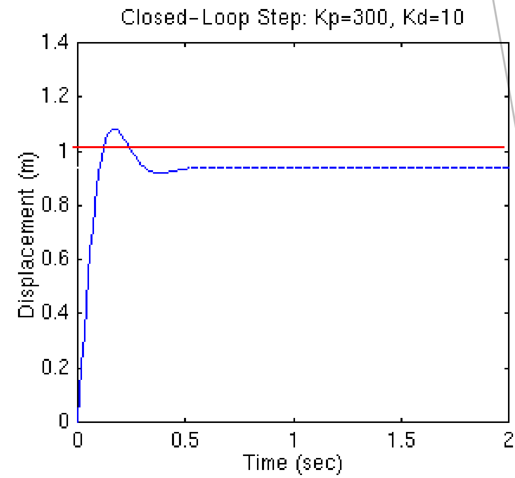
- Let $K_p = 350,\ K_i = 300,$ $K_d = 5500$
- Now, we have obtained the system with **no overshoot, fast rise time, and no steady-state error.**

Ex (cont'd): Summary

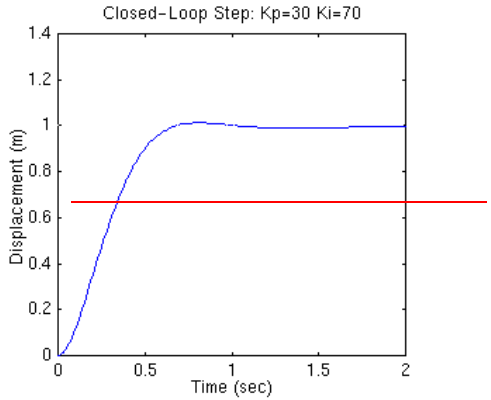
P



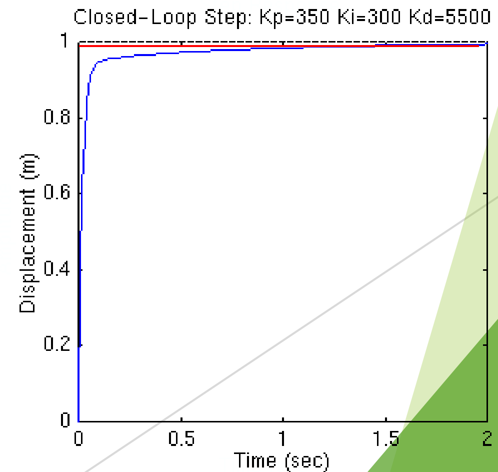
PD



PI



PID



PID Controller Functions

- Output feedback

  → from ***P**roportional action*

  compare output with set-point

- Eliminate steady-state offset (=error)

  → from ***I**ntegral action*

  apply constant control even when error is zero

- Anticipation

  → From ***D**erivative action*

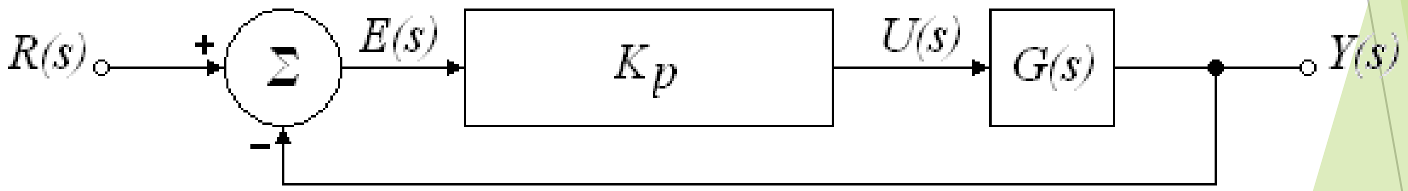
  react to rapid rate of change before errors grows too big

Proportional Controller

- Pure gain (or ***attenuation***) since:

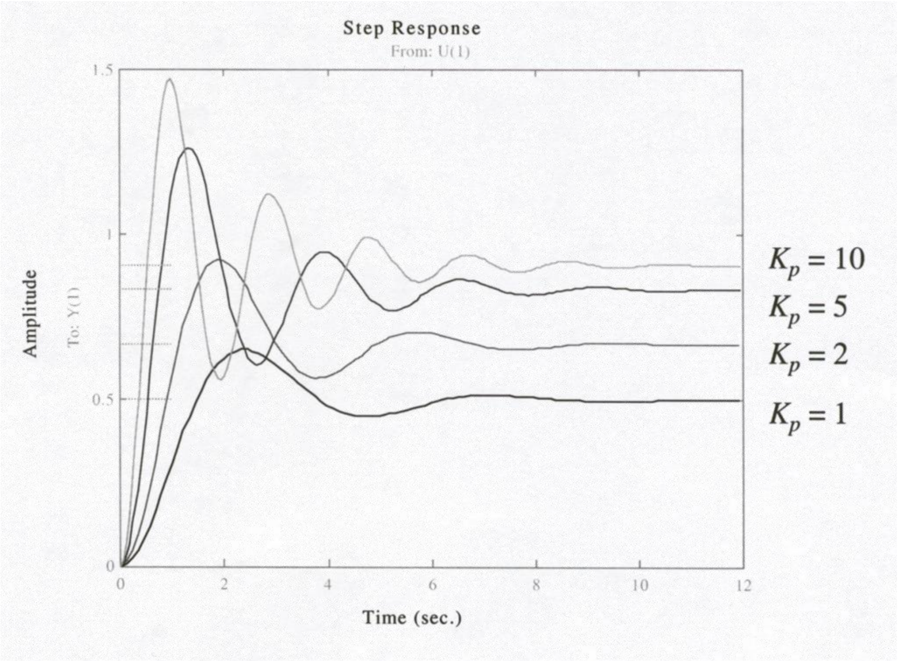
  the controller input is error

  the controller output is a proportional gain



$$E(s)K_p = U(s) \Rightarrow u(t) = K_p e(t)$$

Change in gain in P controller



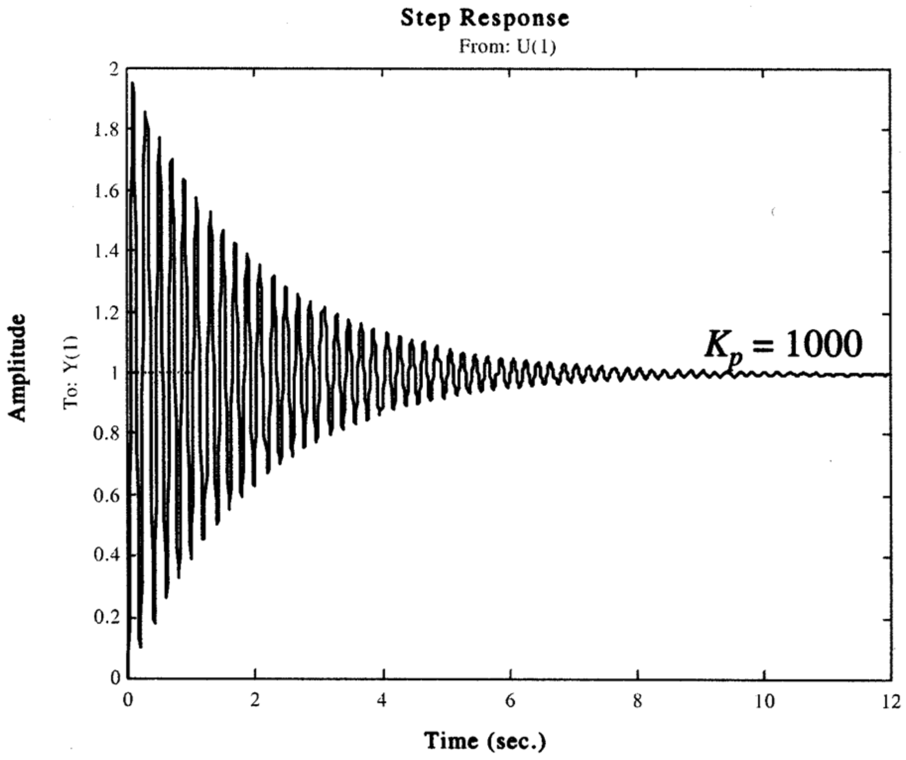
- **Increase in gain:**

$\rightarrow$ Upgrade both steady-state and transient responses

$\rightarrow$ Reduce steady-state error

$\rightarrow$ ***Reduce stability!***

P Controller with *high* gain



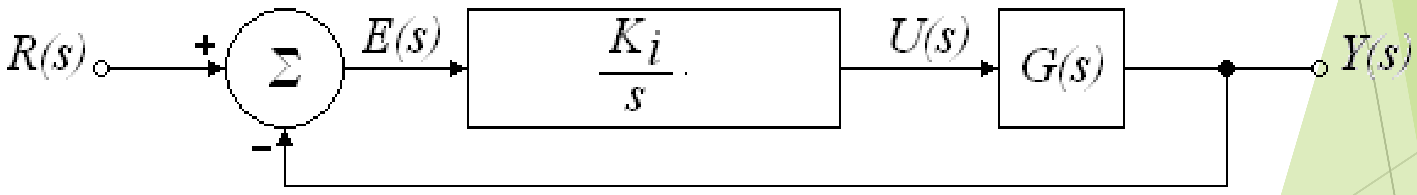
Integral Controller

- Integral of error with a constant gain

$\rightarrow$ increase the system type by 1

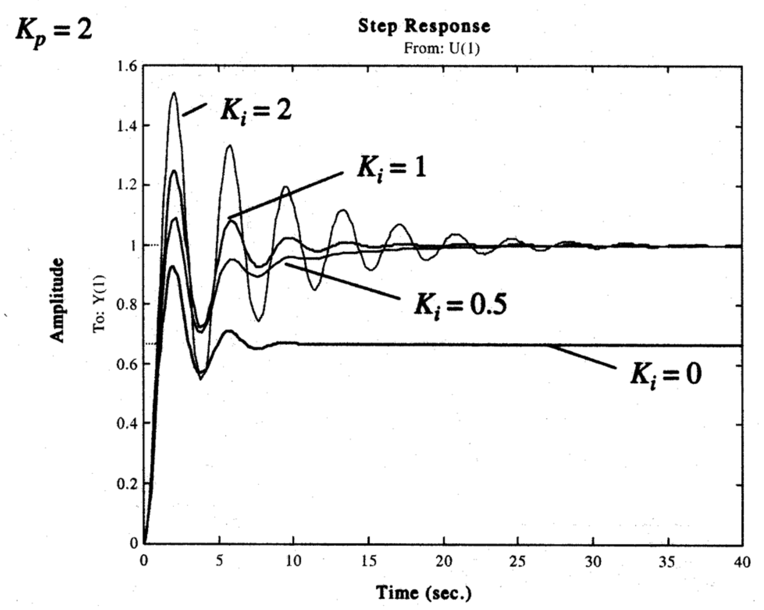
$\rightarrow$ *eliminate steady-state error for a unit step input*

$\rightarrow$ amplify overshoot and oscillations



$$E(s)\frac{K_i}{s} = U(s) \Rightarrow u(t) = K_i \int_0^t e(t)dt$$

Change in gain for PI controller



- **Increase in gain:**

  → Do not upgrade steady-state responses

  → Increase slightly settling time

  → ***Increase oscillations and overshoot!***

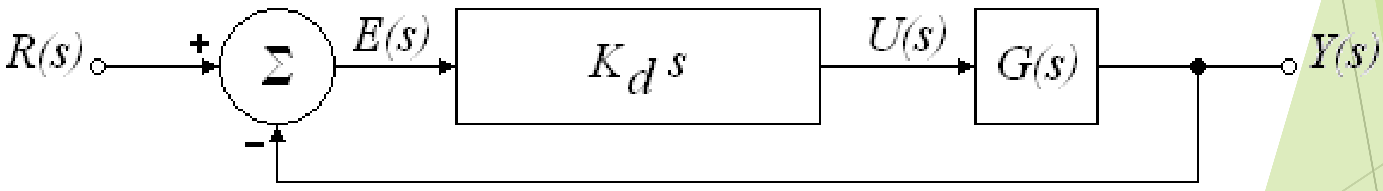
Derivative Controller

- Differentiation of error with a constant gain

  → detect rapid change in output

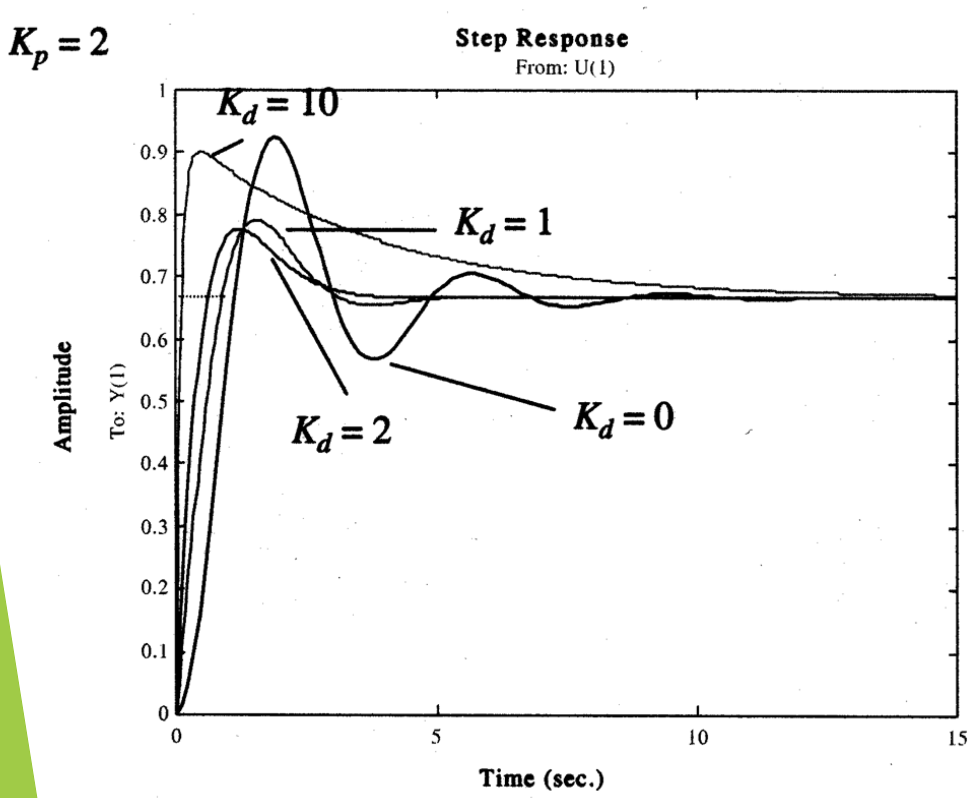
  → *reduce overshoot and oscillation*

  → do not affect the steady-state response



$$E(s)K_d s = U(s) \Rightarrow u(t) = K_d \frac{de(t)}{dt}$$

Effect of change for gain PD controller



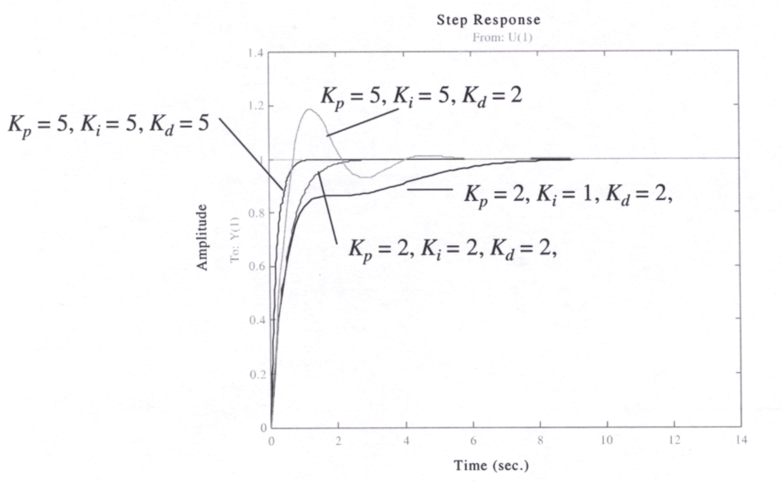
- **Increase in gain:**

  → Upgrade transient response

  → Decrease the peak and rise time

  → ***Increase overshoot and settling time!***

Changes in gains for PID Controller



Conclusions

- **Increasing the proportional feedback gain** ***reduces steady-state errors***, but high gains almost always ***destabilize the system*.**
- **Integral control** provides ***robust reduction in steady-state errors***, but often ***makes the system less stable***.
- **Derivative control** usually ***increases damping and improves stability***, but has almost ***no effect on the steady state error***
- These ***3 kinds of control combined*** from the **classical PID controller**

Conclusion - PID

- The standard PID controller is described by the equation:

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

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

Application of PID Control

- PID regulators provide reasonable control of most industrial processes, provided that the performance demands is not too high.
- PI control are generally adequate when plant/process dynamics are essentially of 1st-order.
- PID control are generally ok if dominant plant dynamics are of 2nd-order.
- More elaborate control strategies needed if process has long time delays, or lightly-damped vibrational modes