Control Systems

Lecture 7

PID Design

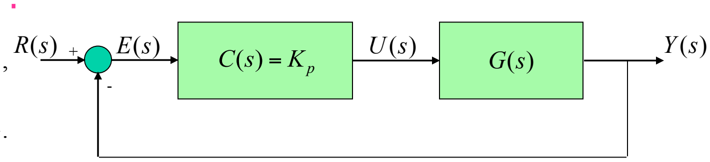
PID Controller

PID: Proportional – Integral – Derivative

P Controller:

$$\frac{Y(s)}{R(s)} = \frac{C(s)G(s)}{1+C(s)G(s)},$$

$$\frac{E(s)}{R(s)} = \frac{1}{1+C(s)G(s)}.$$



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

Step Reference:

$$R(s) = \frac{1}{s} \Rightarrow e_{ss} = \lim_{s\to 0} sE(s) = \lim_{s\to 0} s \frac{1}{1+K_p G(s)} \frac{1}{s} = \frac{1}{1+K_p G(0)}$$

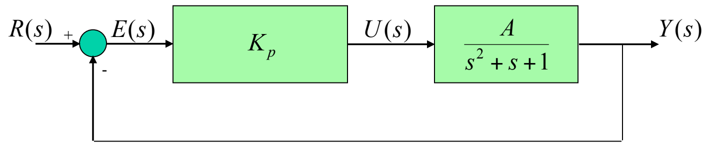
$e_{ss} = 0 \Leftrightarrow K_p G(0) \to \infty$ True when:

- Proportional gain is high
- Plant has a pole at the origin

High gain proportional feedback (needed for good tracking) results in underdamped (or even unstable) transients.

PID Controller

P Controller: Example (P_controller.m)



$$\frac{Y(s)}{R(s)} = \frac{K_p G(s)}{1+K_p G(s)} = \frac{K_p A}{s^2+s+(1+K_p A)}$$

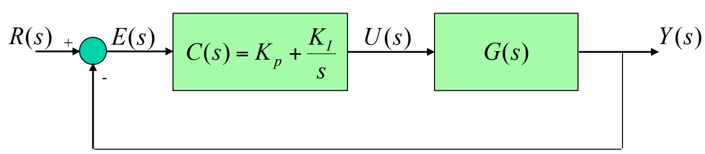
$$\left.\begin{aligned}\omega_n^2 &= 1+K_p A\\ 2\zeta\omega_n &= 1\end{aligned}\right. \Rightarrow \zeta = \frac{1}{2\omega_n} = \frac{1}{2\sqrt{1+K_p A}} \xrightarrow[K_p\to\infty]{} 0$$

✓ Underdamped transient for large proportional gain
✓ Steady state error for small proportional gain

PID Controller

PI Controller:

$$\frac{Y(s)}{R(s)} = \frac{C(s)G(s)}{1+C(s)G(s)},$$
$$\frac{E(s)}{R(s)} = \frac{1}{1+C(s)G(s)}.$$



$$u(t) = K_p e(t) + K_I \int_0^t e(\tau)d\tau, \qquad U(s) = \left(K_p + \frac{K_I}{s}\right)E(s)$$

Step Reference:

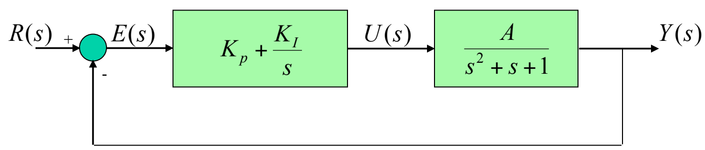
$$R(s) = \frac{1}{s} \Rightarrow e_{ss} = \lim_{s\to 0} sE(s) = \lim_{s\to 0} s\frac{1}{1+\left(K_p+\frac{K_I}{s}\right)G(s)}\frac{1}{s} = \lim_{s\to 0}\frac{1}{1+\left(K_p+\frac{K_I}{s}\right)G(s)} = 0$$

- It does not matter the value of the proportional gain
- Plant does not need to have a pole at the origin. The controller has it!

Integral control achieves perfect steady state reference tracking!!!
Note that this is valid even for $K_p=0$ as long as $K_i\neq 0$

PID Controller

PI Controller: Example (PI_controller.m)



$$\frac{Y(s)}{R(s)} = \frac{\left(K_p + \frac{K_I}{s}\right)G(s)}{1 + \left(K_p + \frac{K_I}{s}\right)G(s)} = \frac{\left(K_p s + K_I\right)A}{s^3 + s^2 + (1 + K_p A)s + K_I A}$$

DANGER: for large K_i the characteristic equation has roots in the RHP

$$s^3 + s^2 + (1 + K_p A)s + K_I A = 0$$

Analysis by Routh's Criterion

PID Controller

PI Controller: Example (PI_controller.m)

$$s^3 + s^2 + (1 + K_p A)s + K_I A = 0$$

Necessary Conditions: $1 + K_p A > 0, K_I A > 0$

This is satisfied because $A > 0, K_p > 0, K_I > 0$

Routh's Conditions:

s^3	1	$1 + K_p A$
s^2	1	$K_I A$
s^1	$1 + K_p A - K_I A$	
s^0	$K_I A$	

$$1 + K_p A - K_I A > 0$$

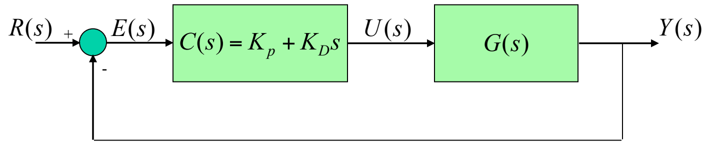
$$\Downarrow$$

$$K_I < K_P + \frac{1}{A}$$

PID Controller

PD Controller:

$$\frac{Y(s)}{R(s)} = \frac{C(s)G(s)}{1+C(s)G(s)},$$
$$\frac{E(s)}{R(s)} = \frac{1}{1+C(s)G(s)}.$$



$$u(t) = K_p e(t) + K_D \frac{de(t)}{dt}, \qquad U(s) = (K_p + K_D s)E(s)$$

Step Reference:

$$R(s) = \frac{1}{s} \Rightarrow e_{ss} = \lim_{s\to 0} sE(s) = \lim_{s\to 0} s \frac{1}{1+(K_p+K_D s)G(s)} \frac{1}{s} = \frac{1}{1+K_p G(0)}$$

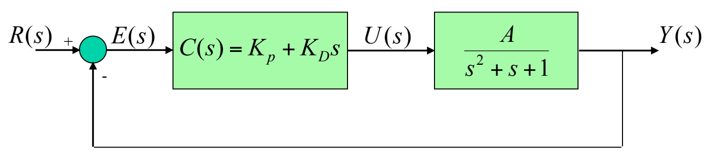
$$e_{ss} = 0 \Leftrightarrow K_p G(0) \to \infty$$

True when:
- Proportional gain is high
- Plant has a pole at the origin

PD controller fixes problems with stability and damping by adding "anticipative" action

PID Controller

PD Controller: Example (PD_controller.m)



$$\frac{Y(s)}{R(s)} = \frac{(K_p + K_D s)G(s)}{1+(K_p+K_D s)G(s)} = \frac{A(K_p + K_D s)}{s^2 + (1+K_D A)s + (1+K_p A)}$$

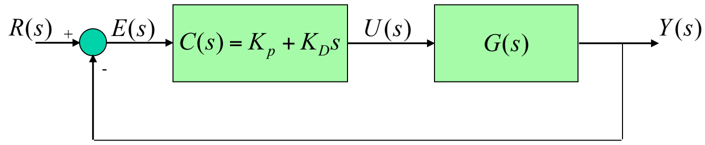
$$\omega_n^2 = 1 + K_p A$$
$$2\zeta\omega_n = 1 + K_D A \qquad \Rightarrow \zeta = \frac{1+K_D A}{2\omega_n} = \frac{1+K_D A}{2\sqrt{1+K_p A}}$$

- ✓ The damping can be increased now independently of K_p
- ✓ The steady state error can be minimized by a large K_p

PID Controller

PD Controller:

$$\frac{Y(s)}{R(s)} = \frac{C(s)G(s)}{1+C(s)G(s)},$$
$$\frac{E(s)}{R(s)} = \frac{1}{1+C(s)G(s)}.$$



$$u(t) = K_p e(t) + K_D \frac{de(t)}{dt}, \qquad U(s) = \left(K_p + K_D s\right)E(s)$$

NOTE: cannot apply pure differentiation.
In practice,

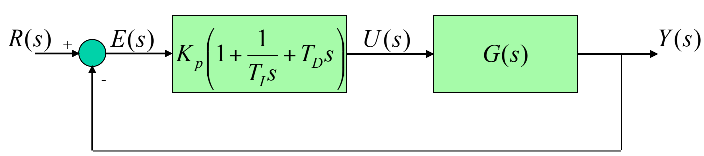
$$K_D s$$

is implemented as

$$\frac{K_D s}{\tau_D s + 1}$$

PID Controller

PID: Proportional – Integral – Derivative



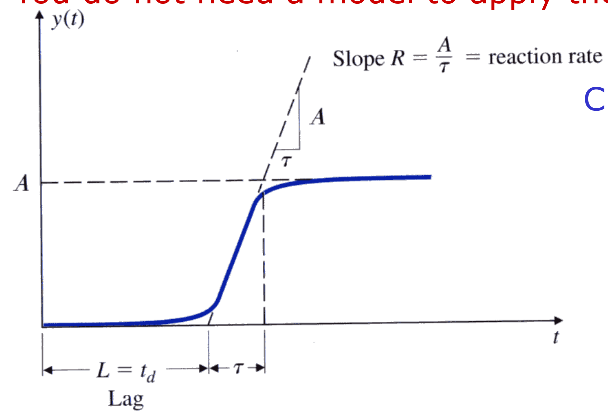
$$u(t) = K_p\left[e(t) + \frac{1}{T_I}\int_0^d e(\tau)d\tau + T_D\frac{de(t)}{dt}\right] \quad K_I = \frac{K_p}{T_I}, K_D = K_p T_D$$

$$\frac{U(s)}{E(s)} = K_p\left(1 + \frac{1}{T_I s} + T_D s\right)$$

PID Controller: Example (PID_controller.m)

PID Controller: Ziegler-Nichols Tuning

- Empirical method (no proof that it works well but it works well for simple systems)
- Only for stable plants
- You do not need a model to apply the method

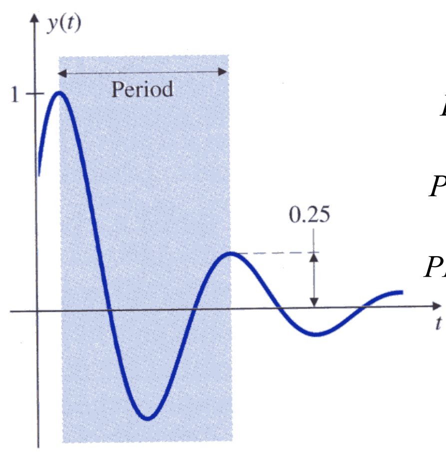


Class of plants:

$$\frac{Y(s)}{U(s)} = \frac{Ke^{-t_d s}}{\tau s + 1}$$

PID Controller: Ziegler-Nichols Tuning

METHOD 1: Based on step response, tuning to decay ratio of 0.25.



Tuning Table:

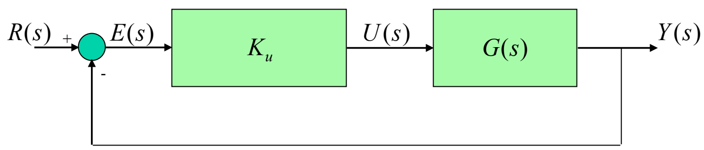
P: $K_p = \frac{\tau}{t_d}$

PD: $K_p = 0.9\frac{\tau}{t_d}, T_I = \frac{t_d}{0.3}$

PID: $K_p = 1.2\frac{\tau}{t_d}, T_I = 2t_d, T_D = 0.5t_d$

PID Controller: Ziegler-Nichols Tuning

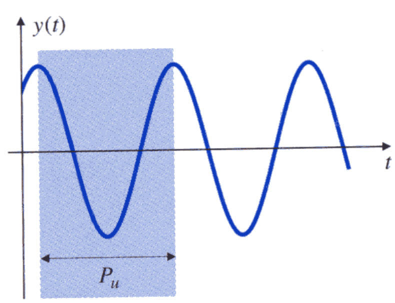
METHOD 2: Based on limit of stability, ultimate sensitivity method.



- Increase the constant gain K_u until the response becomes purely oscillatory (no decay – marginally stable – pure imaginary poles)
- Measure the period of oscillation P_u

PID Controller: Ziegler-Nichols Tuning

METHOD 2: Based on limit of stability, ultimate sensitivity method.



Tuning Table:

$$P: \quad K_p = 0.5K_u$$

$$PD: \quad K_p = 0.45K_u, T_I = \frac{P_u}{1.2}$$

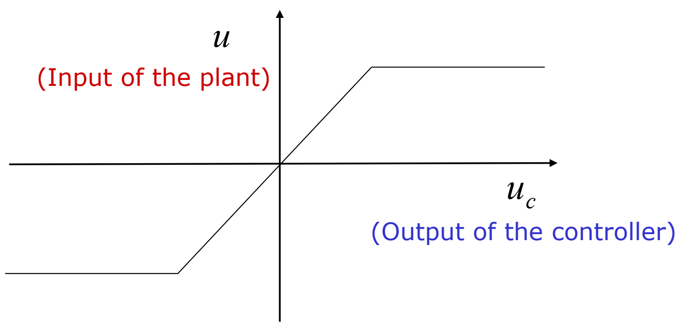
$$PID: \quad K_p = 0.6K_u, T_I = \frac{P_u}{2}, T_D = \frac{P_u}{8}$$

The Tuning Tables are the same if you make:

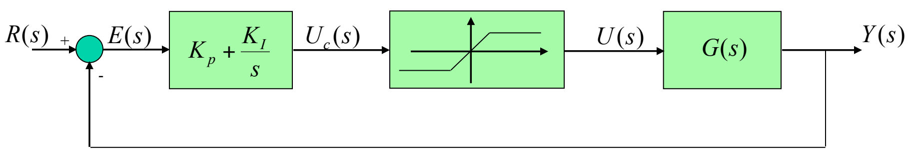
$$K_u = 2\frac{\tau}{t_d}, P_u = 4t_d$$

PID Controller: Saturation

Actuator Saturates:
- valve (fully open)
- aircraft rudder (fully deflected)



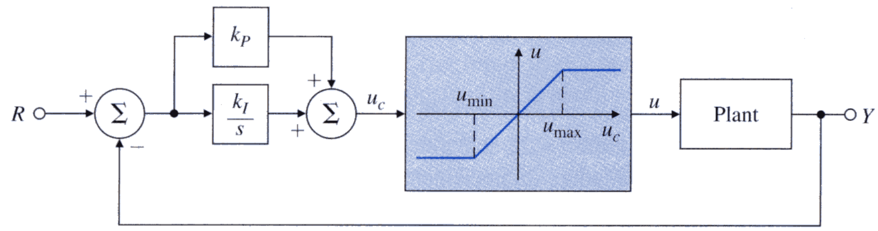
PID Controller: Saturation



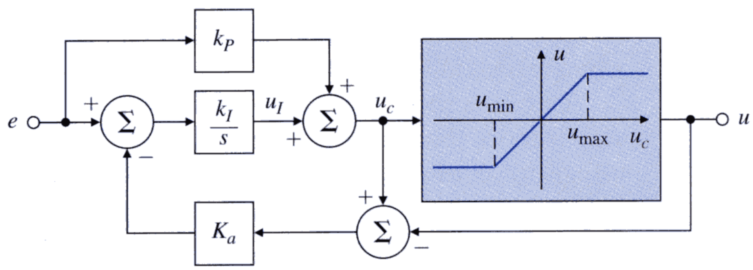
What happens?
- large step input in r
- large e
- large $u_c \rightarrow u$ saturates
- eventually e becomes small
- u_c still large because the integrator is "charged"
- u still at maximum
- y overshoots for a long time

PID Controller: Saturation

Plant without Anti-Windup:

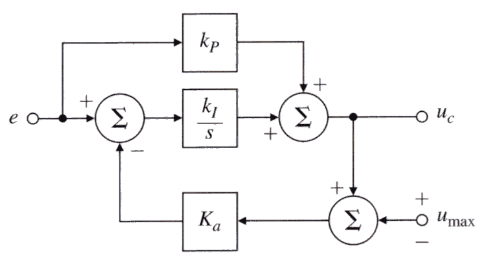


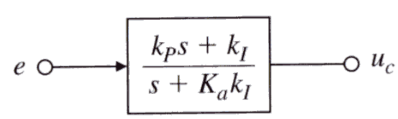
Plant with Anti-Windup:



PID Controller: Saturation

In saturation, the plant behaves as:





For large K_a, this is a system with very low gain and very fast decay rate, i.e., the integration is turned off.

Saturation/Antiwindup: Example (Antiwindup_sim.mdl)