Course of
"Industrial Control System Security"
2024/25

Industrial controllers

Prof. Francesco Montefusco

Department of Economics, Law, Cybersecurity, and Sports Sciences

Università degli studi di Napoli Parthenope

francesco.montefusco@uniparthenope.it

Team code**: 09tkpu5**

Implementation of control systems

- As mentioned, an automatic controller compares the actual value of the controlled output with the reference signal, compute the error signal, and produces a control signal with the aim to achieve an error within the limits set by the specifications/requirements.

- The strategy used by the controller to calculate the control signal defines the so-called type of **control action**.

- The simplest industrial controllers are classified into two categories, based on the type of control action they perform:

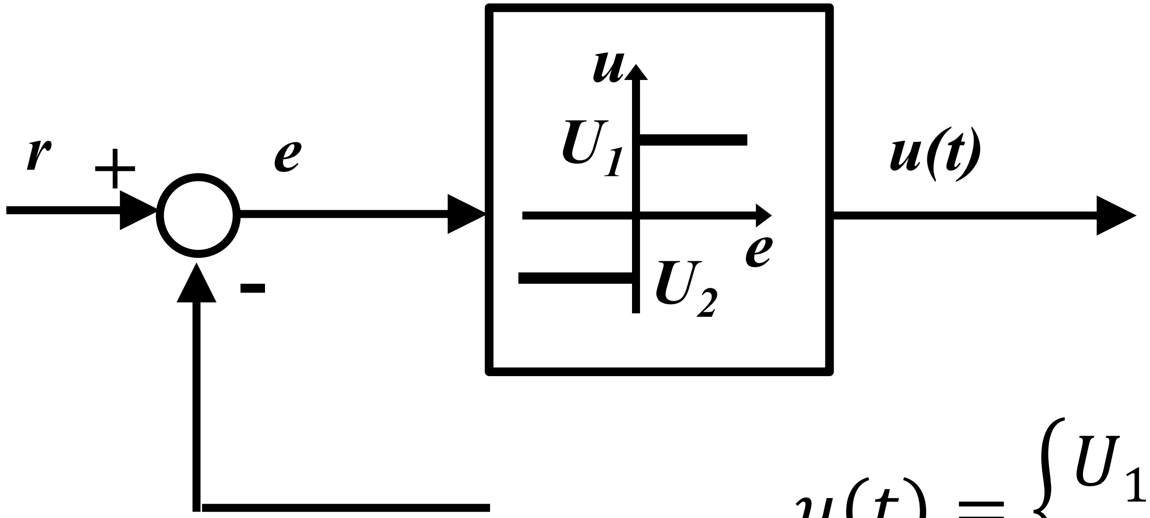
- Two-position controllers, also called **on-off controllers** or **relays**;

- **Proportional-Integral-Derivative controllers**, also called **PID** controllers or standard **regulators**.

Relays

- These controllers generally have only two operating conditions; these, in many cases, are "**on**" and "**off**".
- They can be interpreted as simple automatic switches, are easy to make, reliable and inexpensive.
- For this reason, they find wide application both in industrial systems (level regulation systems, temperature regulation in industrial ovens, etc.) and in domestic applications (refrigerator, water heater, heating system, etc.)

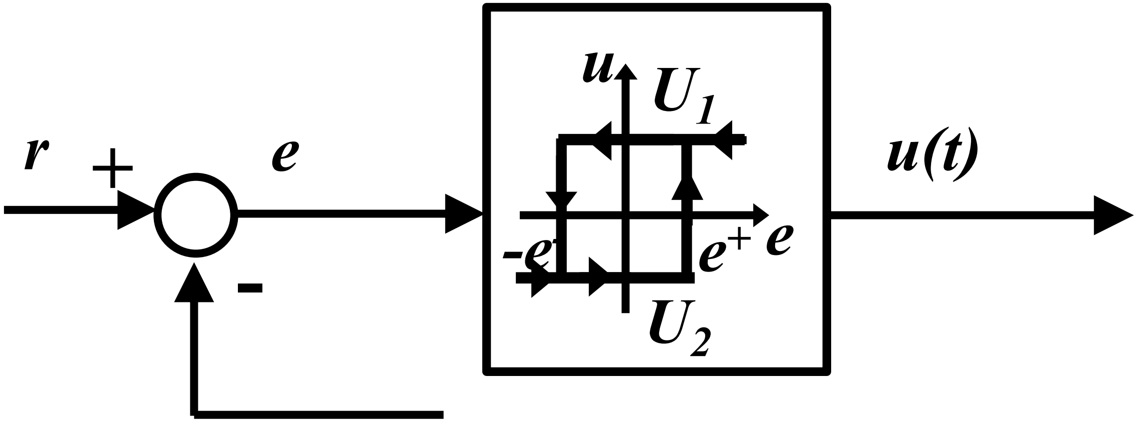
Input- output relationship of the ideal relay



$$u(t) = \begin{cases} U_1 & \text{if} \quad e(t) > 0 \\ U_2 & \text{if} \quad e(t) < 0 \end{cases}$$

Issues for the ideal relay

- The ideal relay tends to switch at a high frequency, keeping the error around zero.
- Such high-frequency switching should be avoided as it would cause considerable wear to both the relay itself and the other devises, reducing their lifespan.
- To avoid this inconvenience, so-called hysteresis relays are used.



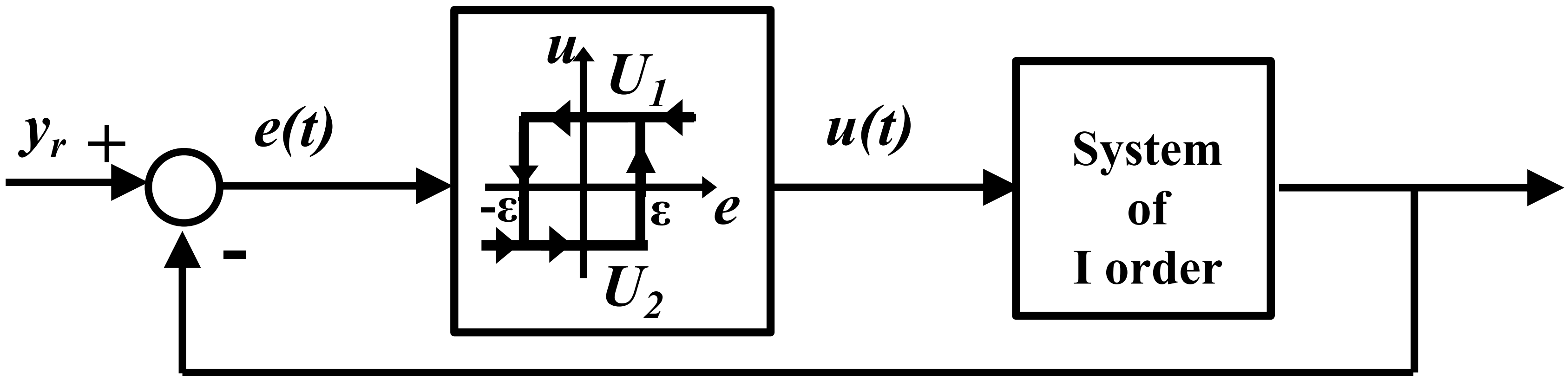
- In this controller, switching from $\boldsymbol{U_2}$ to $\boldsymbol{U_1}$ occurs not when the error exceeds zero, but when it is higher than threshold value $\boldsymbol{e^+}$; similarly, switching from U_1 to U_2 occurs when the error is lower than $\boldsymbol{-e^-}$.

- In this way, a finite switching frequency is obtained, and by choosing the threshold values appropriately, it is possible to impose acceptable switching frequency values.

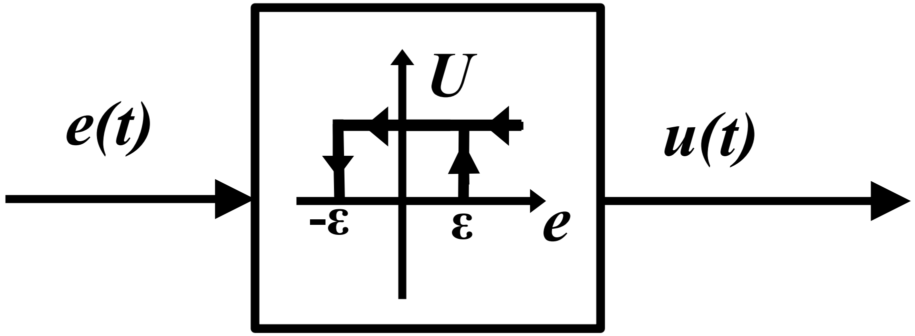
- By the way, the error will not remain strictly at zero, but will oscillate between values "close" to $\boldsymbol{-e^-}$ and $\boldsymbol{e^+}$.

- These values ($\boldsymbol{-e}$ and $\boldsymbol{e^+}$) are called ***differential thresholds***.

Design of a relay for a plant of first order

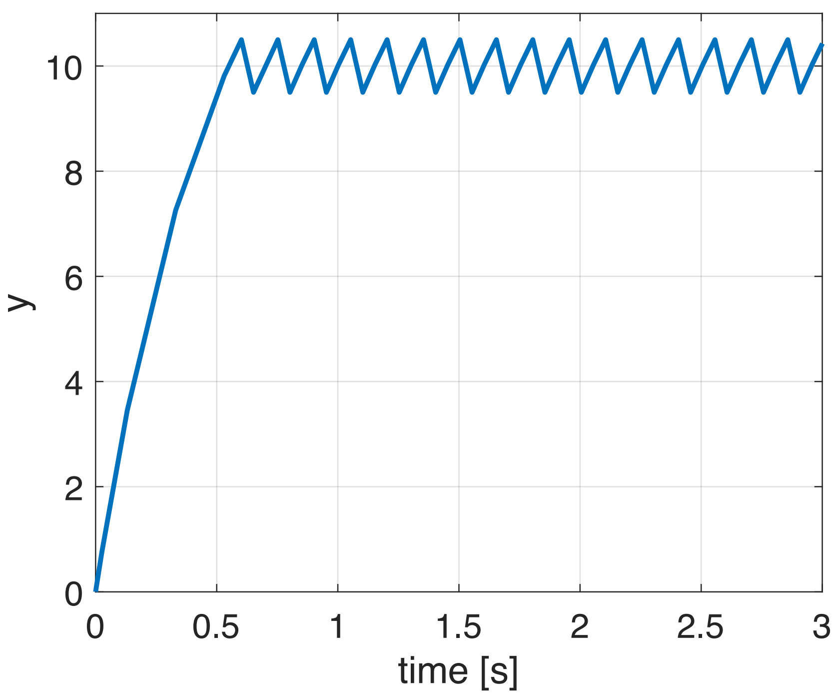


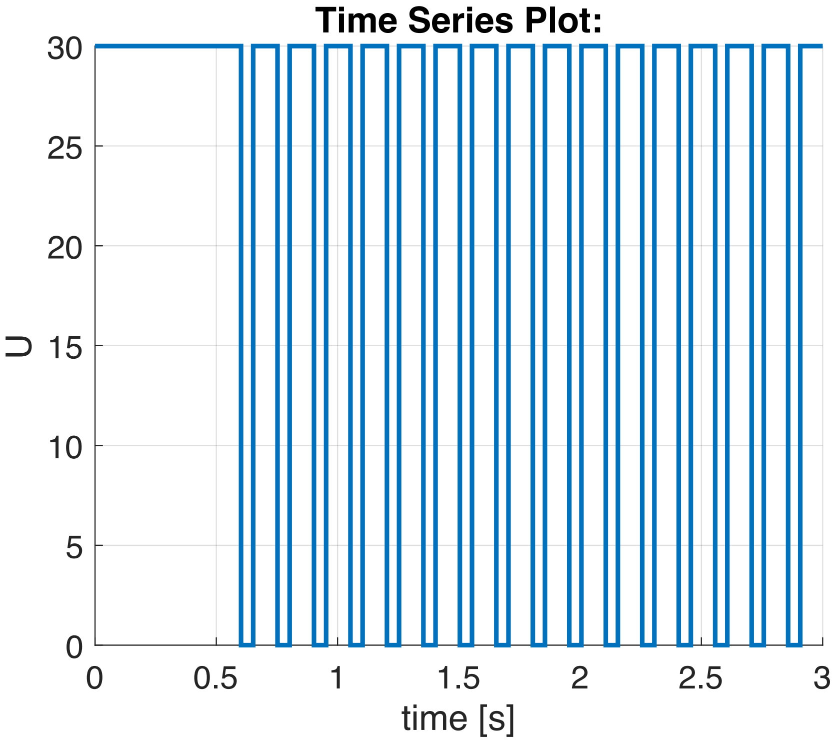
- The design parameters of this controller are $U_1=U$ ($U_2=0$) and ε.



- In order to guarantee relay switching, the following condition needs to be satisfied: $G_0U>y_r+\varepsilon$

- G_0=1/2, τ=1/2, y_r=10; ⇒ ε=5%y_r =0.5, U=30





Parametri della risposta - 1/4

- Vediamo come i parametri di funzionamento del sistema, ossia il tempo di salita e l'ampiezza degli intervalli di chiusura e apertura del relè sono legati ai parametri di progetto del controllore.

- Per il tempo di salita da 0 al 100% di y_r si ha:

$$T_s = \tau \log \frac{GU}{GU - y_r} = \tau \log \frac{\sigma}{\sigma - 1}$$

$$\sigma = \frac{GU}{y_r} \quad \text{è il fattore di "sovralimentazione" del sistema di controllo}$$

σ	1.25	1.5	2	3
T_s	$1.6\,\tau$	$1.1\,\tau$	$0.69\,\tau$	$0.41\,\tau$

Parametri della risposta – 2/4

- Per quanto riguarda T_{on} e T_{off} si ha:

$$T_{on} = \tau \log \frac{GU - y_r + \varepsilon}{GU - y_r - \varepsilon}$$

$$T_{off} = \tau \log \frac{y_r + \varepsilon}{y_r - \varepsilon}$$

- Se $\sigma>2$ allora $T_{on}<T_{off}$.
- Se $\varepsilon \to 0$ sia T_{on} *che* $T_{off} \to 0$ e quindi la frequenza di commutazione tende all'infinito.

Parametri della risposta – 3/4

- Detto $T=T_{on}+T_{off}$ il "periodo" del segnale di controllo e $\rho=T_{on}\ /\ T_{off}$ il "coefficiente di parzializzazione", valori tipici di tali parametri in funzione del fattore di sovralimentazione σ per il caso di $\varepsilon=5\%y_r$ e $\varepsilon=10\%y_r$, sono riportati nelle seguenti tabelle.

σ	1.25	1.5	2	3
T	$0.51\,\tau$	$0.30\,\tau$	$0.20\,\tau$	$0.15\,\tau$
ρ	80%	67%	50%	33%

Tabella 4.2 Valori di T e ρ nel caso di $\varepsilon=5\%y_r$

σ	1.25	1.5	2	3
T	$1.05\,\tau$	$0.61\,\tau$	$0.40\,\tau$	$0.30\,\tau$
ρ	80%	67%	50%	33%

Tabella 4.3 Valori di T e ρ nel caso di $\varepsilon=10\%y_r$

Parametri della risposta – 4/4

- Si conclude che:
  - per avere piccoli valori del tempo di salita T_s bisogna aumentare il fattore di sovralimentazione σ; ciò comporta un aumento della frequenza di commutazione e il sovradimensionamento dell'attuatore;
  - per avere piccoli errori, bisogna diminuire la soglia differenziale, e, di conseguenza, tollerare frequenze di commutazione più elevate.

Design of relay in case of disturbances

- By assuming a constant disturbance in the process, which decrease the steady-state response by y_d.
- In this case, the condition that guarantees the switching of the relay is given by

$$GU - y_d > y_r + \varepsilon$$

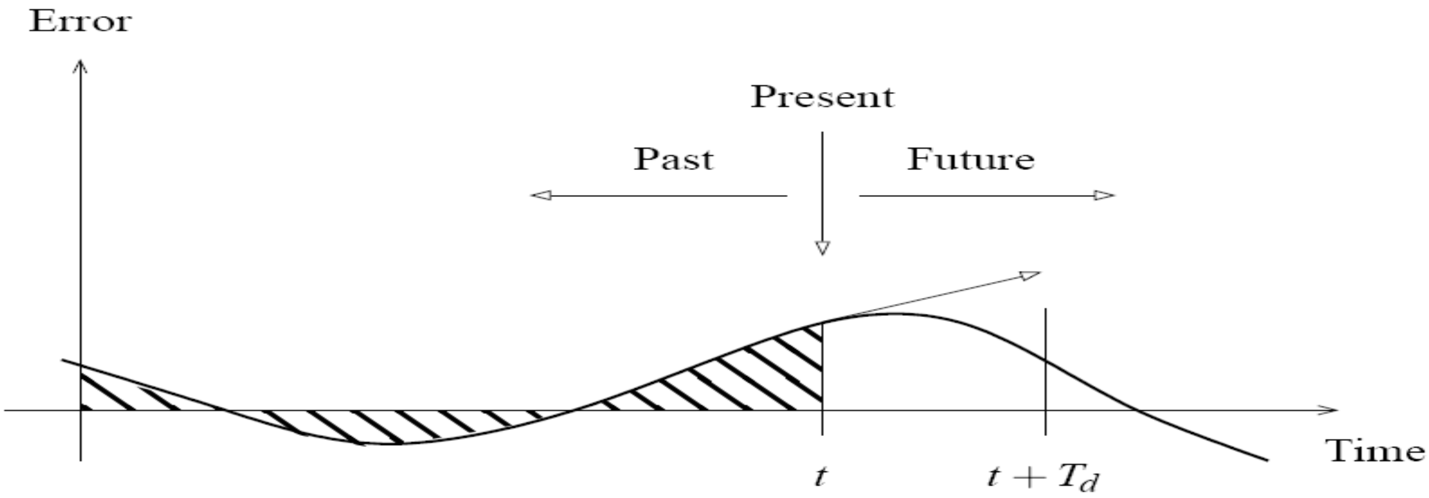
- Then, in case of disturbances, the control level U must be higher value than the case with no disturbances (if $y_d>0$).

PID controller

- A PID controller is characterized by a Proportional-Integral-Derivative control actions with respect to the tracking error $e(t) = r(t) - y(t)$.

- A PID can be written in the time domain as

$$u(t) = K_P e(t) + K_I \int e(t)dt + K_D \frac{de(t)}{dt}$$



PID controller

- A PID, defined in the time domain by

$$u(t) = K_P e(t) + K_I \int e(t)dt + K_D \frac{de(t)}{dt}$$

it is usually defined in the form

$$u(t) = K_P \left(e(t) + \frac{K_I}{K_P} \int e(t)dt + \frac{K_D}{K_P} \frac{de(t)}{dt} \right)$$

$$= K_P \left(e(t) + \frac{1}{T_I} \int e(t)dt + T_D \frac{de(t)}{dt} \right)$$

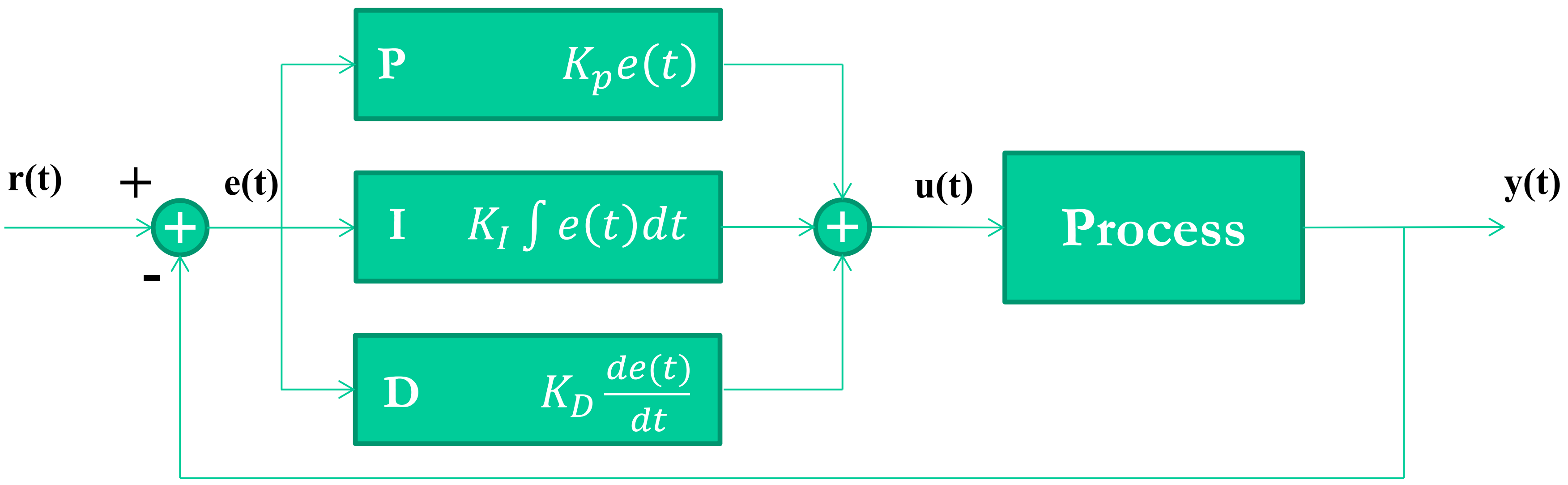
where $T_I = \frac{K_P}{K_I}$ (Integral time) and $T_D = \frac{K_D}{K_P}$ (Derivative time)

PID controller in the time domain

- A PID controller in the time domain form,

$$u(t) = K_P e(t) + K_I \int e(t)dt + K_D \frac{de(t)}{dt}$$

can be represented as

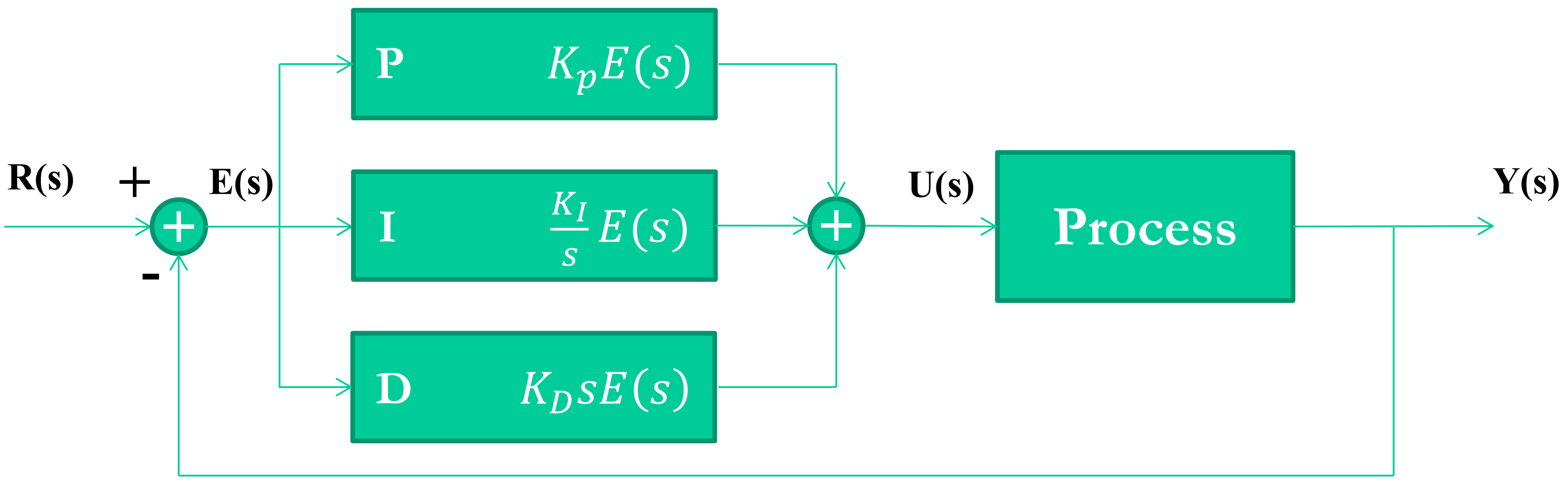


PID controller in the Laplace domain

- A PID controller defined in the Laplace domain as

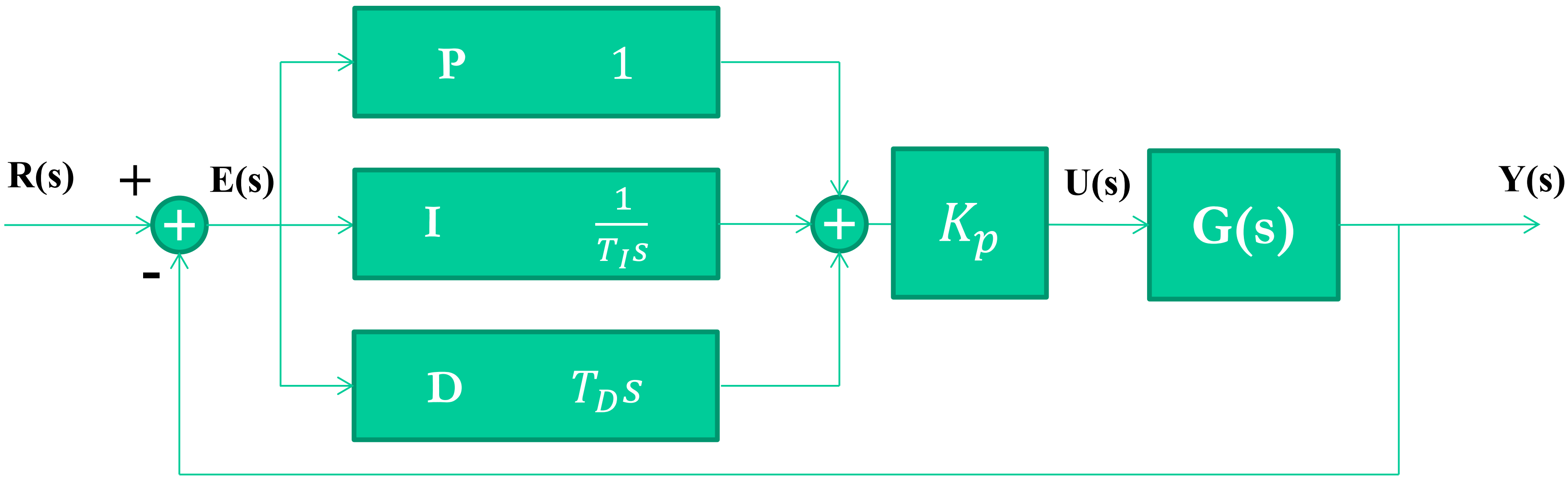
$$U(s) = K_p E(s) + \frac{K_I}{s} E(s) + K_D s E(s)$$

can be represented as



- Or by $u(t) = K_p\left(e(t) + \frac{1}{T_I}\int e(t)dt + T_D\frac{de(t)}{dt}\right)$

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



PID controller

- Usually, only a subset of the possible PID control actions are implemented.
- In particular we have
  - ***Proportional controller (P)***
  - ***Integral controller (I)***
  - ***Proportional-Integral controller (PI)***
  - ***Proportional-Derivative controller (PD)***
  - ***Proportional-Integral-Derivative controller (PID)***

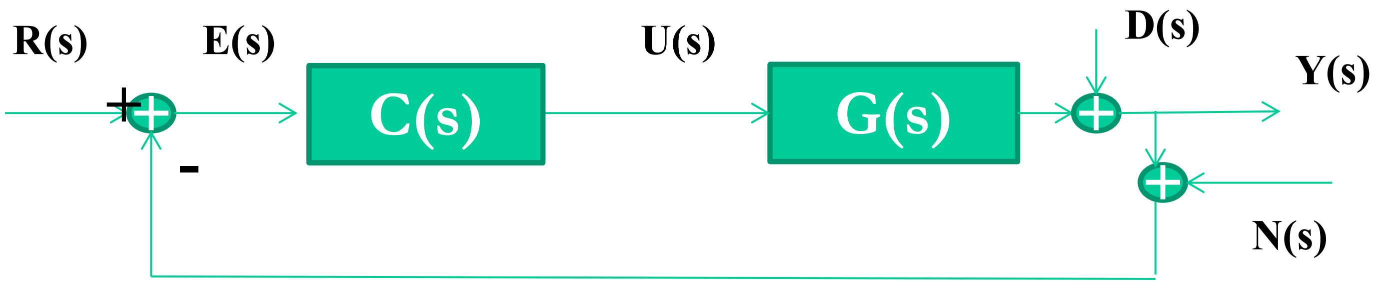
P controller

- The P controller can be written as

$$u(t) = K_p e(t) \rightarrow U(s) = K_p E(s) \rightarrow \frac{U(s)}{E(s)} = C(s) = K_p$$

- **P controllers are used to reduce the steady-state error** when
  - the integral action is not required for the steady-state performance
  - the bandwidth can be increased without violating the other requirements

- Let us consider a closed loop system in the form

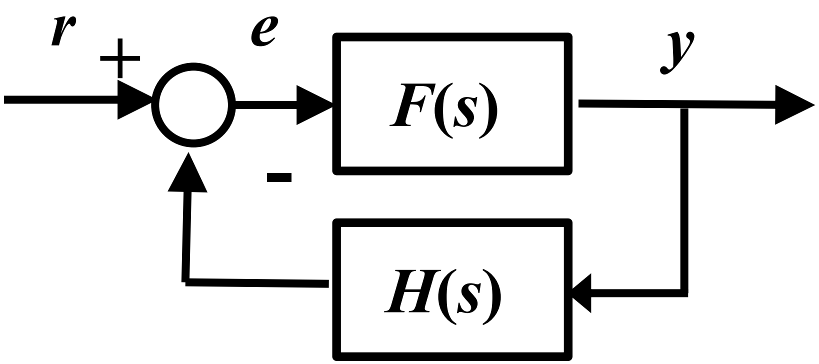


where

$$G(s) = \frac{G_0}{(1 + s\tau)}$$

$$C(s) = K_p$$

$$\boldsymbol{Y}(\boldsymbol{s}) = W_r(s)\boldsymbol{R}(\boldsymbol{s}) + W_{\boldsymbol{d}}(s)\boldsymbol{R}(\boldsymbol{s}) + W_{\boldsymbol{n}}(s)\boldsymbol{R}(\boldsymbol{s})$$



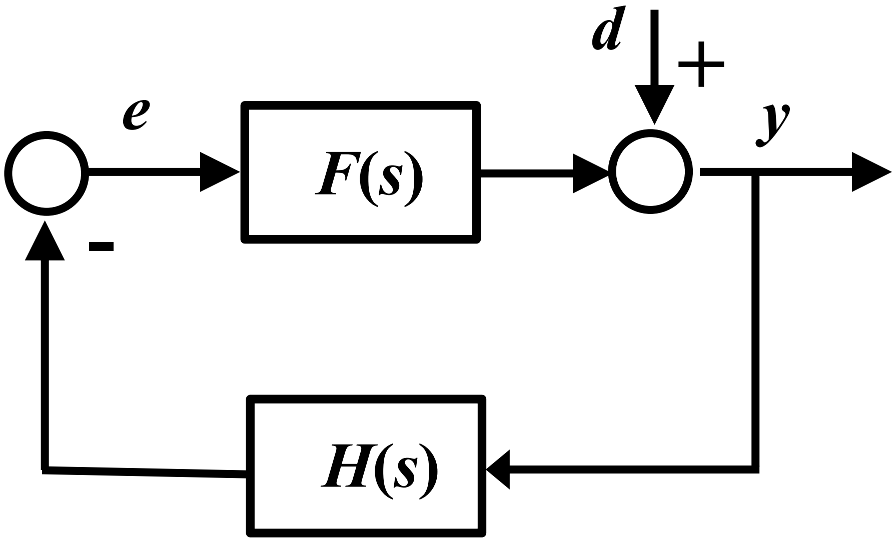
$$W_r(s) = \frac{Y(s)}{R(s)} = \frac{F(s)}{1 + F(s)H(s)}.$$

Here, $H(s)$=**1**, $F(s) = K_p \frac{G_0}{1+\tau s}$,

$$W_r(s) = \frac{K_p \frac{G_0}{1+\tau s}}{1 + K_p \frac{G_0}{1+\tau s}} = \frac{K_p G_0}{1 + K_p G_0 + \tau s}$$

$$= \frac{K_p G_0}{(1 + K_p G_0)\left(1 + \frac{\tau s}{1 + K_p G_0}\right)} = \frac{\frac{K_p G_0}{(1 + K_p G_0)}}{\left(1 + \frac{\tau s}{1 + K_p G_0}\right)}$$

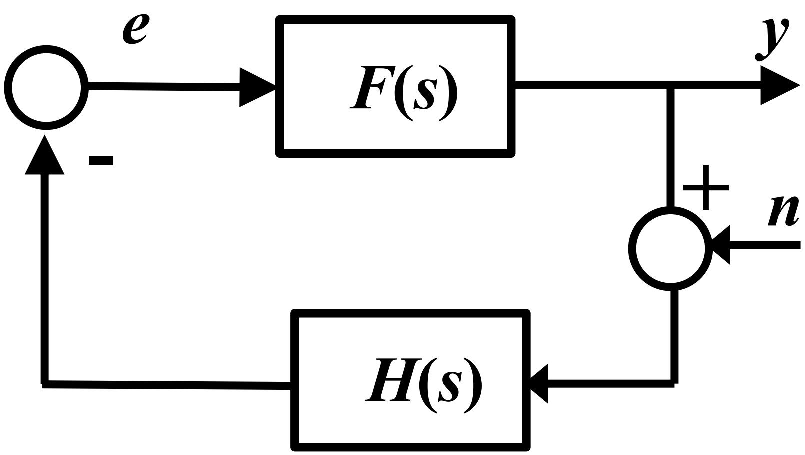
Relationship between Y(s) and D(s)



$$W_d(s) = \frac{Y(s)}{D(s)} = \frac{1}{1 + F(s)H(s)}.$$

$$W_d(s) = \frac{1}{1 + K_p \frac{G_0}{1+\tau s}} = \frac{1+\tau s}{1 + K_p G_0 + \tau s}$$

$$= \frac{1+\tau s}{1 + K_p G_0 + \tau s} = \frac{1+\tau s}{(1 + K_p G_0)\left(1 + \frac{\tau s}{1 + K_p G_0}\right)}$$



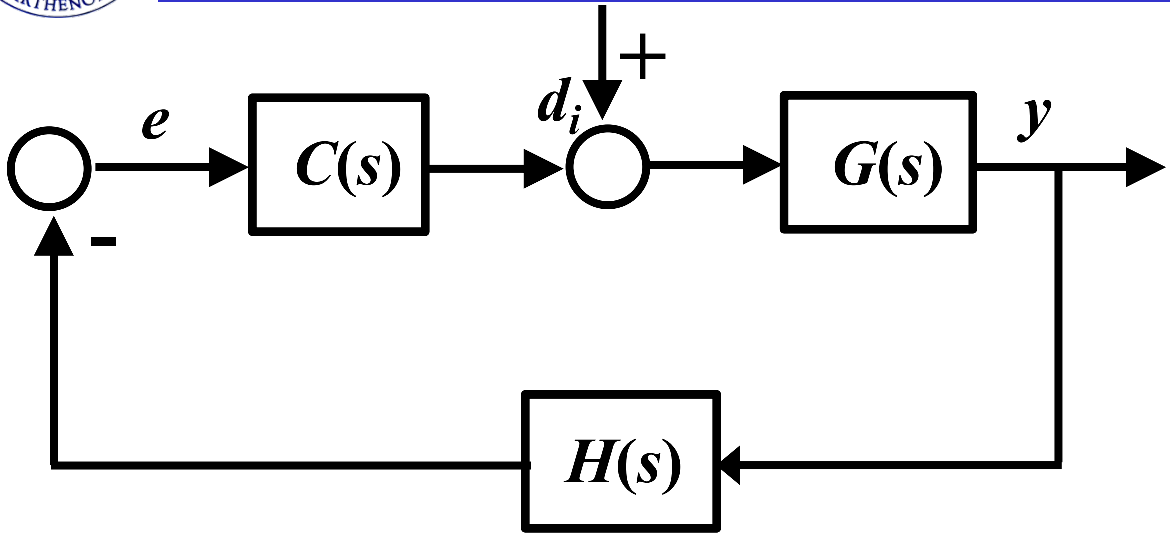
$$W_n(s) = \frac{Y(s)}{N(s)} = -\frac{F(s)H(s)}{1+F(s)H(s)}.$$

Note $W_n(s) = -W_r(s)$

$$\Rightarrow W_n(s) = -\frac{K_p \frac{G_0}{1+\tau s}}{1+K_p \frac{G_0}{1+\tau s}} = -\frac{K_p G_0}{1+K_p G_0 + \tau s}$$

$$= -\frac{K_p G_0}{(1+K_p G_0)\left(1+\frac{\tau s}{1+K_p G_0}\right)} = -\frac{\frac{K_p G_0}{(1+K_p G_0)}}{\left(1+\frac{\tau s}{1+K_p G_0}\right)}$$

In case of disturbance as input to the plant

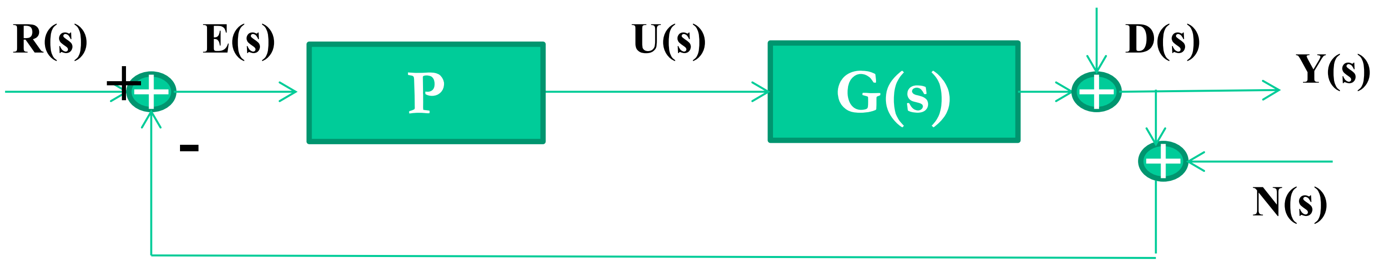


$$W_{d_i}(s) = \frac{Y(s)}{D_i(s)} = \frac{G(s)}{1 + C(s)G(s)H(s)}.$$

$$\Rightarrow \quad W_{d_i}(s) = \frac{\frac{G_0}{1+\tau s}}{1 + K_p \frac{G_0}{1+\tau s}} = \frac{G_0}{1 + K_p G_0 + \tau s}$$

$$= \frac{G_0}{(1 + K_p G_0)\left(1 + \frac{\tau s}{1 + K_p G_0}\right)} = \frac{\frac{G_0}{1 + K_p G_0}}{1 + \frac{\tau s}{1 + K_p G_0}}$$

- Then,



$$\boldsymbol{Y(s)} = W_r(s)\boldsymbol{R(s)} + W_{\boldsymbol{d}}(s)\boldsymbol{D(s)} + W_{\boldsymbol{n}}(s)\boldsymbol{N(s)}$$

$$= \frac{\frac{K_pG_0}{(1+K_pG_0)}}{\left(1+\frac{\tau s}{1+K_pG_0}\right)}R(s) + \frac{1+\tau s}{(1+K_pG_0)\left(1+\frac{\tau s}{1+K_pG_0}\right)}D(s)$$

$$-\frac{\frac{K_pG_0}{(1+K_pG_0)}}{\left(1+\frac{\tau s}{1+K_pG_0}\right)}N(s)$$

Hence,

$$Y(s) = W_r(s)\boldsymbol{R}(\boldsymbol{s}) + W_{\boldsymbol{d}}(s)\boldsymbol{D}(\boldsymbol{s}) + W_{\boldsymbol{n}}(s)\boldsymbol{N}(\boldsymbol{s}) =$$
$$W_r(s)\boldsymbol{R}(\boldsymbol{s}) + W_{\boldsymbol{d}}(s)\boldsymbol{D}(\boldsymbol{s}) - W_{\boldsymbol{r}}(s)\boldsymbol{N}(\boldsymbol{s})$$

The performance of the closed loop system have been classified in

- ***Tracking of the reference input*** $\rightarrow \boldsymbol{W_r(s) \cong 1}$
- ***Rejection of the disturbs*** $\rightarrow \boldsymbol{W_d(s) \cong 0}$

However, the noise is not filtered by the system, $W_{\boldsymbol{n}}(s) = -W_{\boldsymbol{r}}(s)$

Real control performance

- To overcome this problem the control theory takes advantage to the fact that the input signals have usually different intervals of frequency
  - ***Reference and disturbs at low frequencies***
  - ***Noise at high frequencies***
- So,
  - $\boldsymbol{W_r(s) \cong 1}$ **and** $\boldsymbol{W_d(s) \cong 0}$ **at low frequencies**
  - $\boldsymbol{W_r(s) \cong 0}$ **at high frequencies**

Steady state performance wrt a step reference

- By assuming a step reference signal $r(t) = R_0 1(t)$, and using the final value theorem,

$$y_\infty = \lim_{t\to\infty} y(t) = \lim_{s\to 0} sY(s) = \lim_{s\to 0} sW_r(s)\frac{R_0}{s} = \frac{G_0 K_p}{1 + G_0 K_p} R_0$$

- Then, the steady error e_∞,

$$e_\infty = R_0 - y_\infty = \frac{1}{1 + G_0 K_p} R_0$$

- And the relative error, $e_{\infty,r}$

$$e_{\infty,r} = \frac{1}{1 + G_0 K_p}$$

Steady state performance wrt a step disturbance

- By assuming a step disturbance signal $r(t) = D_0 1(t)$, and using the final value theorem,

$$y_\infty = \lim_{t\to\infty} y(t) = \lim_{s\to 0} sY(s) = \lim_{s\to 0} sW_d(s)\frac{D_0}{s} = \frac{1}{1+G_0K_p}D_0$$

- At steady state, the effect of the step disturbance of amplitude D_0 on the output is attenuated by the factor $\frac{1}{1+G_0K_p}$.

- If $G_0\, K_p$ >>1 and D_0 is of the same order of magnitude as R_0, then $y_\infty \approx R_0$, therefore the effect of the disturbance is negligible.

Transient performance wrt a step disturbance

- The closed-loop control system is a first order system as the plant to be controlled, indeed

$$W_r = \frac{\frac{K_p G_0}{(1 + K_p G_0)}}{\left(1 + \frac{\tau s}{1 + K_p G_0}\right)}$$

with time constant, τ_w,

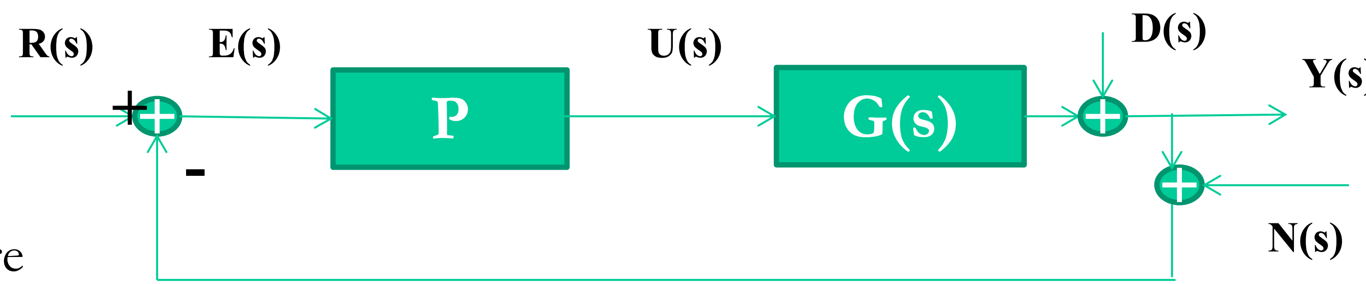
$$\tau_w = \frac{\tau}{1 + K_p G_0}$$

- The the closed-loop control system is faster than the plant to be controlled ($\tau_w \ll \tau$, if $K_p G_0 \gg 1$)

- Note, however, that the control signal $\boldsymbol{u}$ is very high: at time $t = 0, u = K_p(R_0 + D_0)$

P controller with a plant of second order

- Let us consider a closed loop system in the form



where

$$G(s) = \frac{G_0}{\frac{s^2}{\omega_n^2} + \frac{2\zeta}{\omega_n}s + 1}$$

$$\Rightarrow \quad W_r(s) = \frac{\frac{G_0K_p}{1+G_0K_p}}{\left(\frac{s^2}{(1+G_0K_p)\omega_n^2} + \frac{2\zeta}{(1+G_0K_p)\omega_n}s + 1\right)}$$

- $\omega_{nc} = \omega_n\sqrt{1+G_0K_p}$
- $\zeta_c = {}^{\zeta}/_{\sqrt{1+G_0K_p}}$

Steady state and transient performance wrt a step reference

- The steady-state error, e_∞ w.r.t. a step reference and step disturbance, will be the same as for the first order system.
- Once again, G_0K_p >>1 in order to achieve steady-state precision.
- It should be noted, however, that such a choice in this case involves a notable decrease in ζ_c and therefore a high overshoot.
- In practice, overshoots of less than or equal of 30% are tolerated, and therefore $\zeta_c \geq 0.35$. This last constraint, however, can determine a not very high proportional gain, resulting in lower steady state performance.
- About the transient, if $\zeta_c \leq 1$, the settling time is independent of K_p:

$$t_{s5\%} = \frac{3}{\zeta_c \omega_{nc}} = \frac{3}{\zeta \omega_n}$$

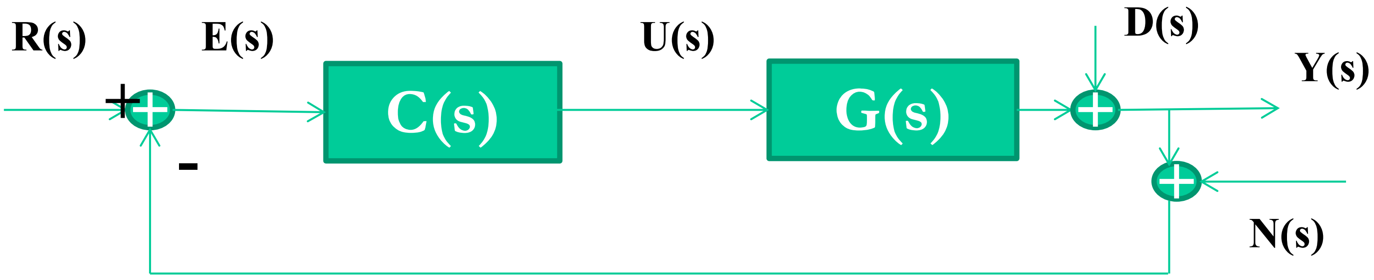
I controller

- The integral (I) action can be written as

$$u(t) = K_I \int e(t)dt \rightarrow U(s) = \frac{K_I E(s)}{s} \rightarrow \frac{U(s)}{E(s)} = C(s) = \frac{K_I}{s}$$

- **I controllers are used to reduce or eliminate the steady-state error**

- Let us consider a closed loop system in the form



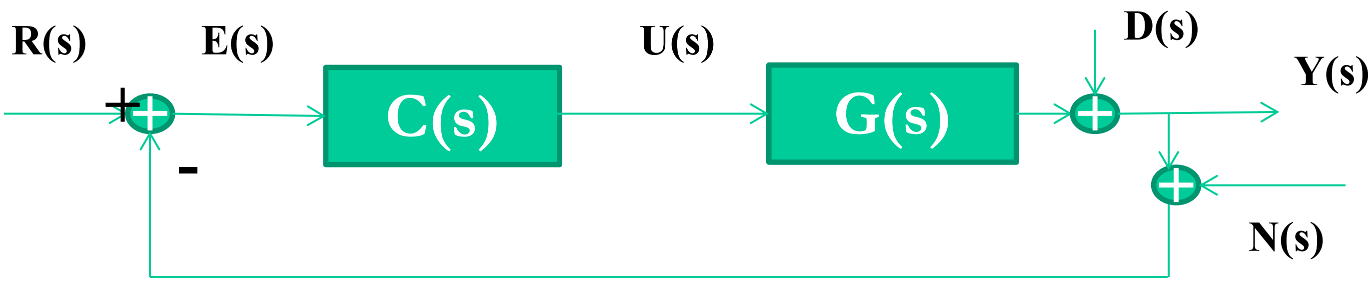
where

$$G(s) = \frac{G_0}{(1 + s\tau)}, C(s) = \frac{K_I}{s}$$

$$W_r(s) = \frac{1}{\left(\frac{\tau}{G_0 K_I} s^2 + \frac{1}{G_0 K_I} s + 1\right)}$$

- $\omega_{nc} = \sqrt{G_0 K_I / \tau}$
- $\zeta_c = \frac{1}{2\sqrt{G_0 K_I \tau}}$

- Let us consider a closed loop system in the form



where

$$G(s) = \frac{G_0}{(1+s\tau)}, C(s) = \frac{K_I}{s}$$

$$\boldsymbol{W_d}(s) = \frac{\tau s^2 + s}{\left(\frac{\tau}{G_0 K_{\boldsymbol{I}}} s^2 + \frac{1}{G_0 K_I} s + 1\right)} = \frac{s(1+s\tau)}{\left(\frac{\tau}{G_0 K_{\boldsymbol{I}}} s^2 + \frac{1}{G_0 K_I} s + 1\right)}$$

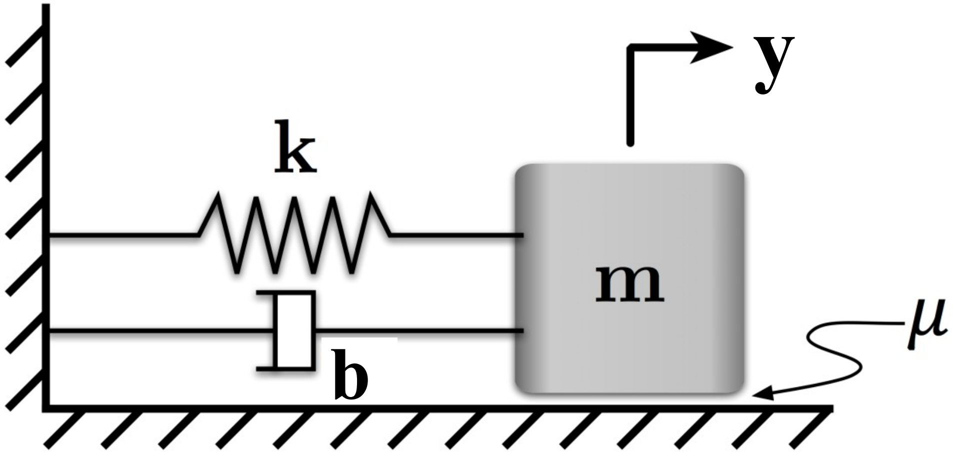
Steady state and transient performance wrt a step reference

- The steady-state error, e_∞ w.r.t. a reference and step disturbance, will be null, i.e. $e_\infty = 0, y_\infty = R_0$
- Increasing the gain of the integral action, K_I, determines a decrease of ζ_c and therefore the closed loop response exhibits a more pronounced oscillation behavior with higher overshoot.
- By maintaining overshoots $\leq$ 30%, then $\zeta_c \geq$ 0.35, i.e. $K_I < 1/(4G\tau\zeta_c^2)$.
- About the transient, if $\zeta_c \leq 1$, the settling time is independent of K_i:

$$t_{s5\%} = \frac{3}{\zeta_c \omega_{nc}} = 6\tau$$

- Let us consider a mass-spring-damper system
- Indicate with $s(t)$ the movement of the mass with respect to a reference position and with $f(t)$ an external control force applied to the mass
- Said $\mathrm{y}(\mathrm{t}) = s(t)$ and $\mathrm{u}(t) = f(t)$, the system can be written as

$$m\ddot{y} + b\dot{y} + ky = u$$



P controller: example

- The aim of the closed loop control is to bring the mass in a new constant position, that is $r(t) = R_0 1(t)$
- Let us assume to use a proportional controller taking care of both the transient behavior and the steady-state performance $u(t) = K_p e(t) = K_p\big(R_0 - y(t)\big)$
- The closed loop equation can be written as

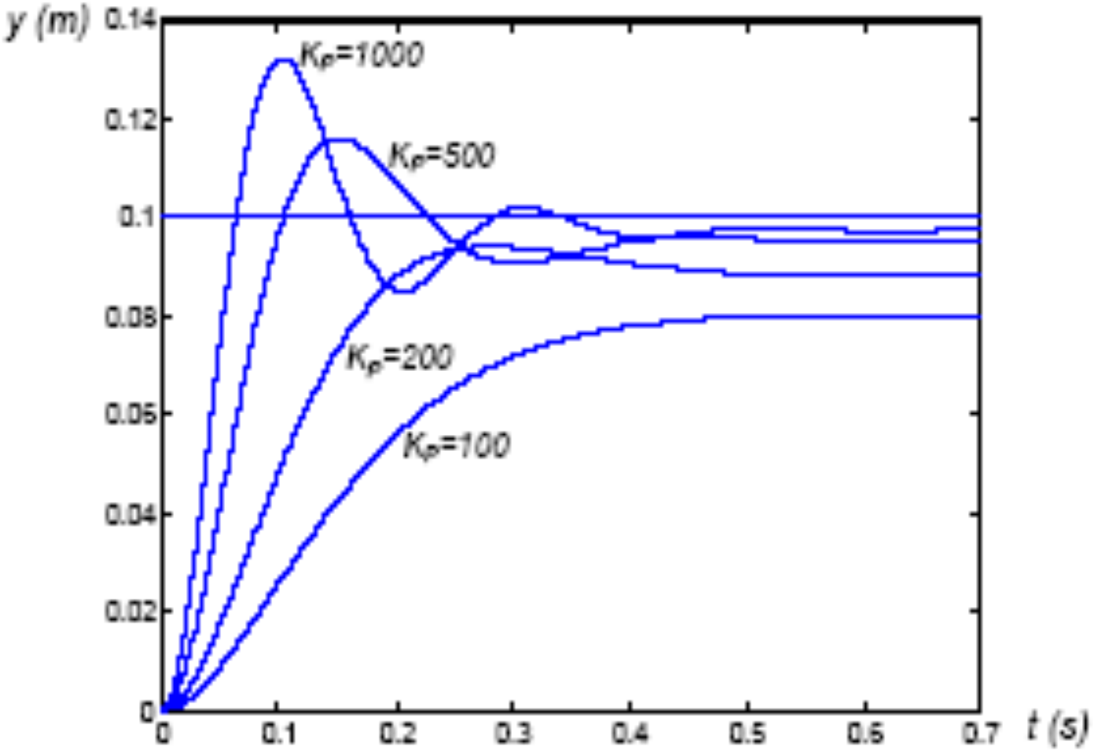
$$m\ddot{y} + b\dot{y} + (k + K_P)y = K_P R_0$$

$$\Rightarrow W(s) = \frac{Y(s)}{R(s)} = \frac{K_P}{ms^2 + bs + k + K_P}$$

$$W_0 = W(0) = \frac{K_p}{k + K_p}, \qquad \zeta_c = \frac{b}{2\sqrt{k + K_p}}$$

P controller: example

- $m = 1 \text{ Kg}$
- $k = 25 \text{ N/m}$
- $b = 20 \text{ Ns/m}$
- $R_0 = 0.1 \text{ m}$



I controller: example

- Let us consider again the tracking problem of the mass-spring-damper system

$$m\ddot{y} + b\dot{y} + ky = u$$

- Let us assume to use an integral controller taking care of both the transient behavior and the steady-state performance

$$u(t) = K_I \int e(t)dt = K_I \int \big(R_0 - y(t)\big)dt$$

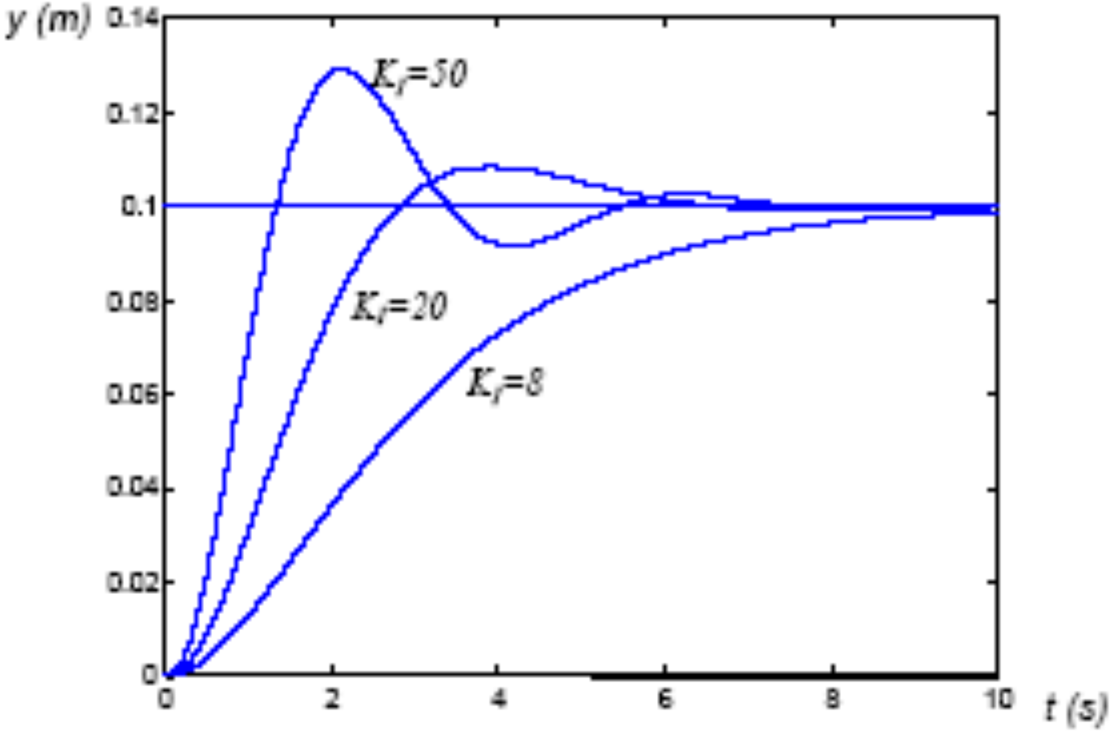
- By substituting the equation of $u(t)$ and assuming u as a step signal

$$m\dddot{e} + b\ddot{e} + k\dot{e} + K_I e = 0$$

- If the choice of K_I is made in such a way as to guarantee that the system is asymptotically stable, then the error tends to zero and therefore the controlled output tracks at infinity the value of the reference signal.

I controller: example

- $m = 1\ \text{Kg}$
- $k = 25\ \text{N/m}$
- $b = 20\ \text{Ns/m}$
- $R_0 = 0.1\ \text{m}$



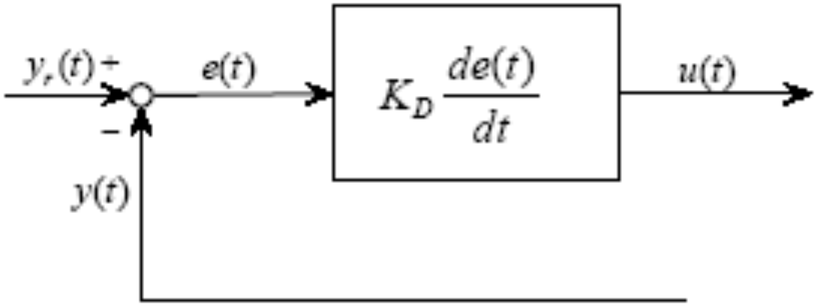
Derivative (D) controller

- The PID integral action can be written as

$$u(t) = K_D \frac{de(t)}{dt} \rightarrow \qquad U(s) = K_D s E(s)$$

- It should be noted that a pure Derivative controller is not Physically Realizable. Indeed, it is realized by adding a pole at a very high frequency, so that to not much modify the performance of the ideal D controller.

- The control signal is proportional to the derivative of the tracking error



- The coefficient K_D is said derivative gain

D controller: example

- Let us consider again the tracking problem of the mass-spring-damper system

$$m\ddot{y} + b\dot{y} + ky = u$$

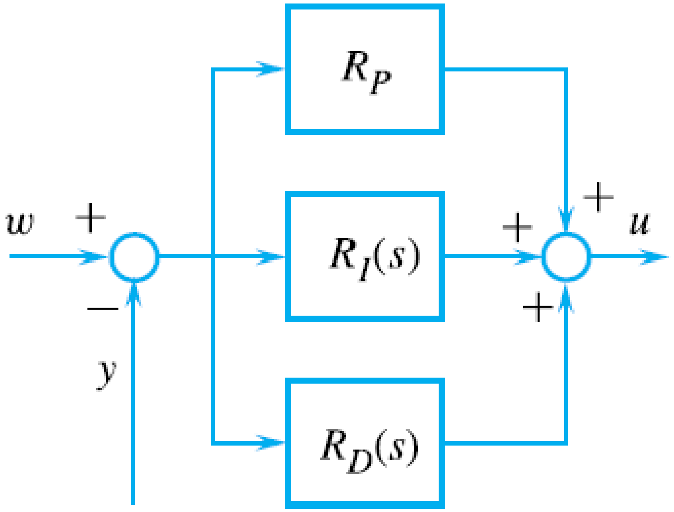
- Let us assume to use a derivative controller taking care of both the transient behavior and the steady-state performance

$$u(t) = K_D \frac{de(t)}{dt} = K_D \frac{d\left(R_0 - y(t)\right)}{dt}$$

- The closed loop equation can be written as

$$m\ddot{y} + (b + K_D)\dot{y} + ky = 0$$

- The effect of the derivative action is
  - Null steady-state output
  - An increase of the damping coefficient $\zeta = \dfrac{b + K_D}{2\sqrt{km}}$



- $R_P = K_P$
- $R_I(s) = \frac{K_I}{s}$ **or** $R_I(s) = \frac{K_P}{T_I s}$, with $T_I = \frac{K_P}{K_I}$
- $R_D(s) = K_D s$; $R_D^a(s) = \frac{K_D s}{1+\frac{K_D}{K_P N}s} = \frac{K_P T_D s}{1+\frac{T_D}{N}s}$, with $T_D = \frac{K_D}{K_P}$

Ideal PID:

$$\boldsymbol{R}_{PID}(s) = K_P + \frac{K_I}{s} + K_D s = \frac{K_D s^2 + K_P s + K_I}{s}$$

or

$$\boldsymbol{R}_{PID}(s) = K_P\left(1 + \frac{1}{T_I s} + T_D s\right) = \frac{K_P}{T_I}\frac{(1 + T_I s + T_I T_D s^2)}{s}$$

Real PID:

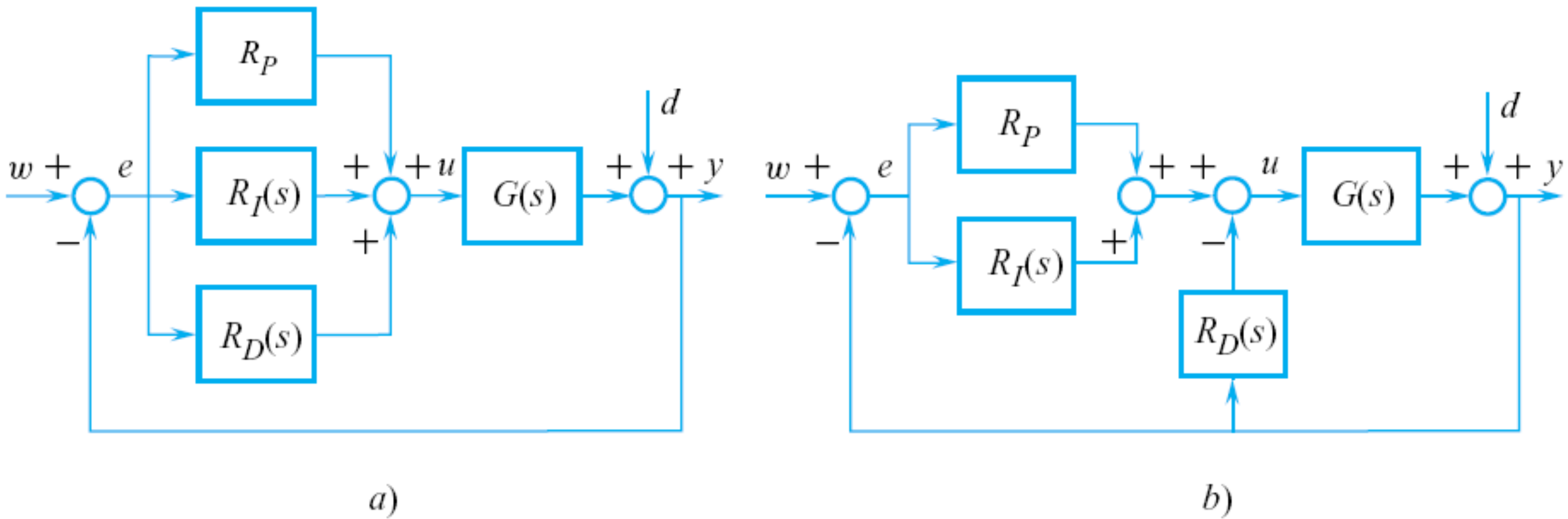
$$\boldsymbol{R}_{PID}^{a}(s) = K_P + \frac{K_I}{s} + \frac{K_D s}{1 + \frac{K_D}{K_P N} s}$$

$$\boldsymbol{R}_{PID}^{a}(s) = K_P\left(1 + \frac{1}{T_I s} + \frac{T_D s}{1 + \frac{T_D}{N} s}\right)$$

PID controller: issues

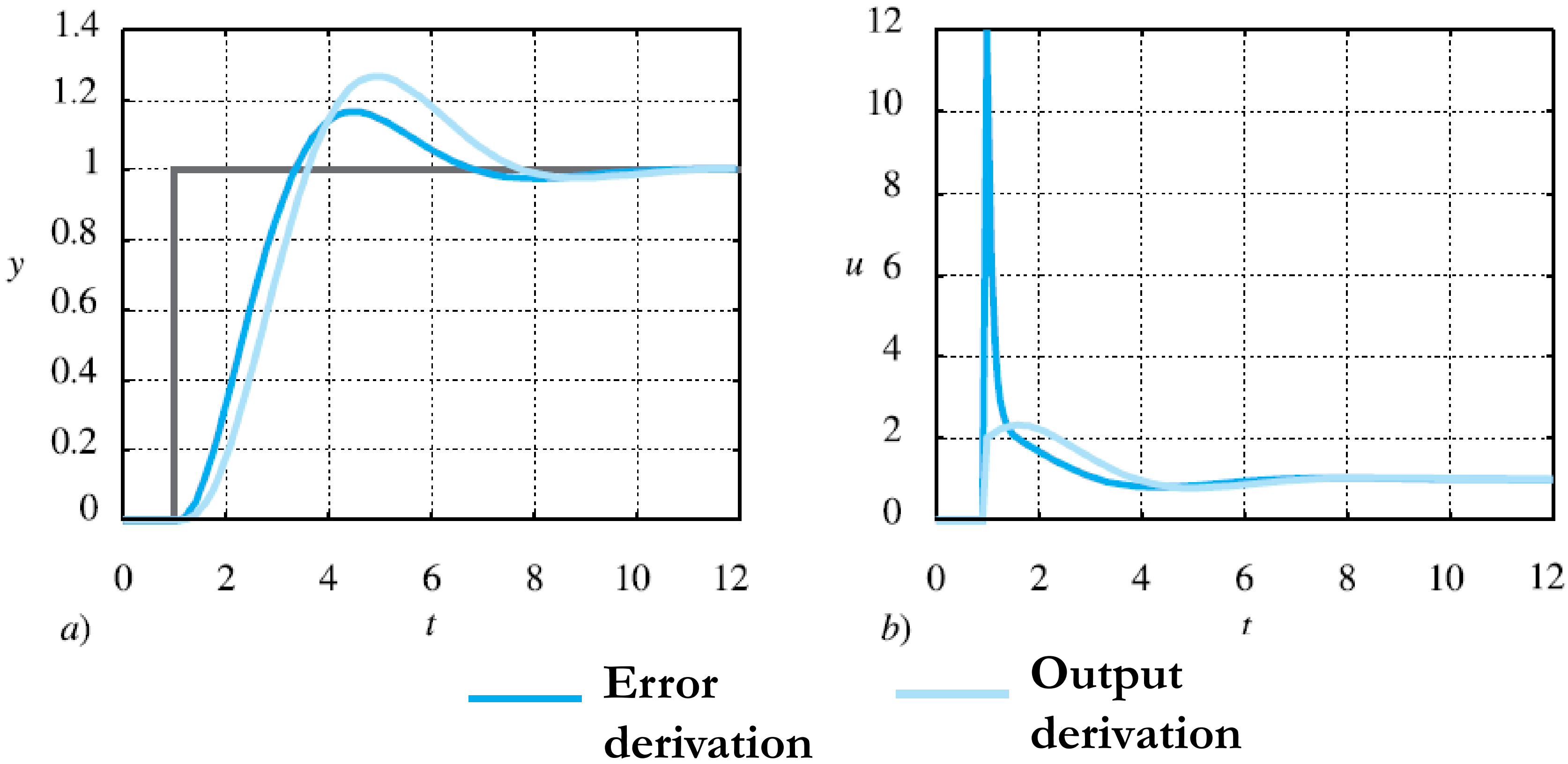
- Limitation of derivative action
  - In the classical feedback control scheme, the derivative action is performed on the error e(t)
  - In the presence of a step reference signal *r*(t), the control variable u(t) assumes an impulsive behavior
  - This sudden variation can cause saturation of the actuator and an operation regime far from the linearity conditions for which the regulator is designed

PID controller: limitation of the derivative action



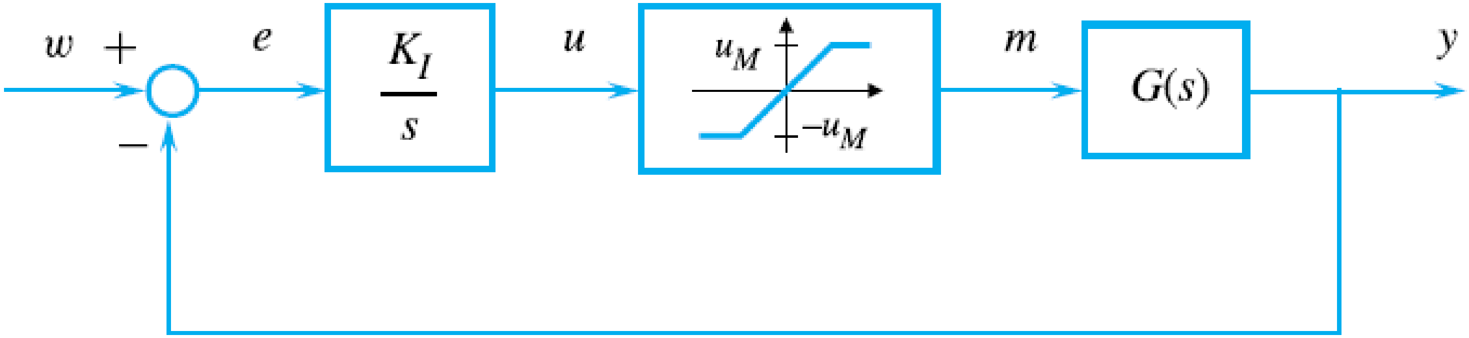
Error derivation **Output derivation**

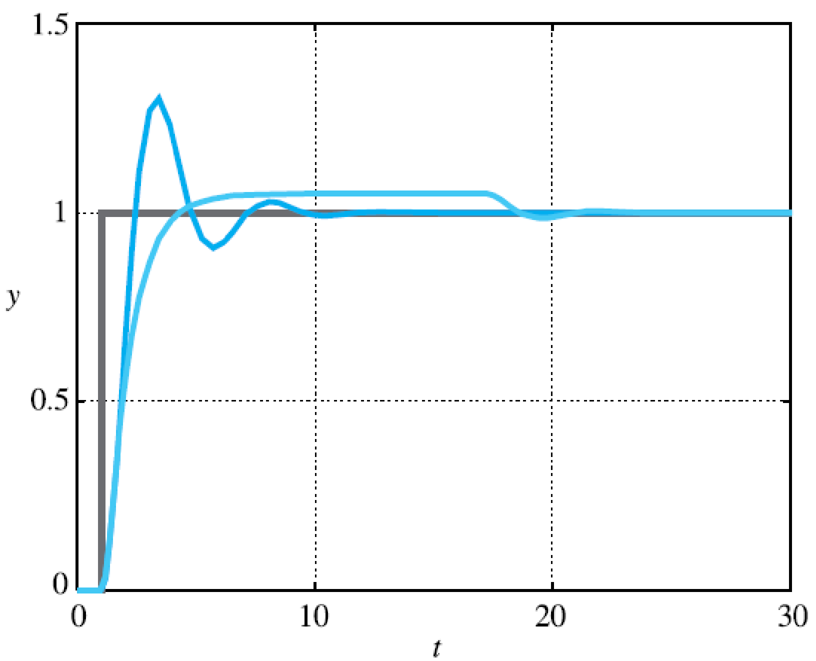
PID controller: error vs output derivation



$$G(s) = \frac{1}{(s+1)^3} \qquad R_{PID}(s) = \frac{(s+1)^2}{s}, \text{ with } \boldsymbol{K_D = K_I = 1, K_P = 2}$$

- Actuator: nonlinear behavior (saturation)





linear case

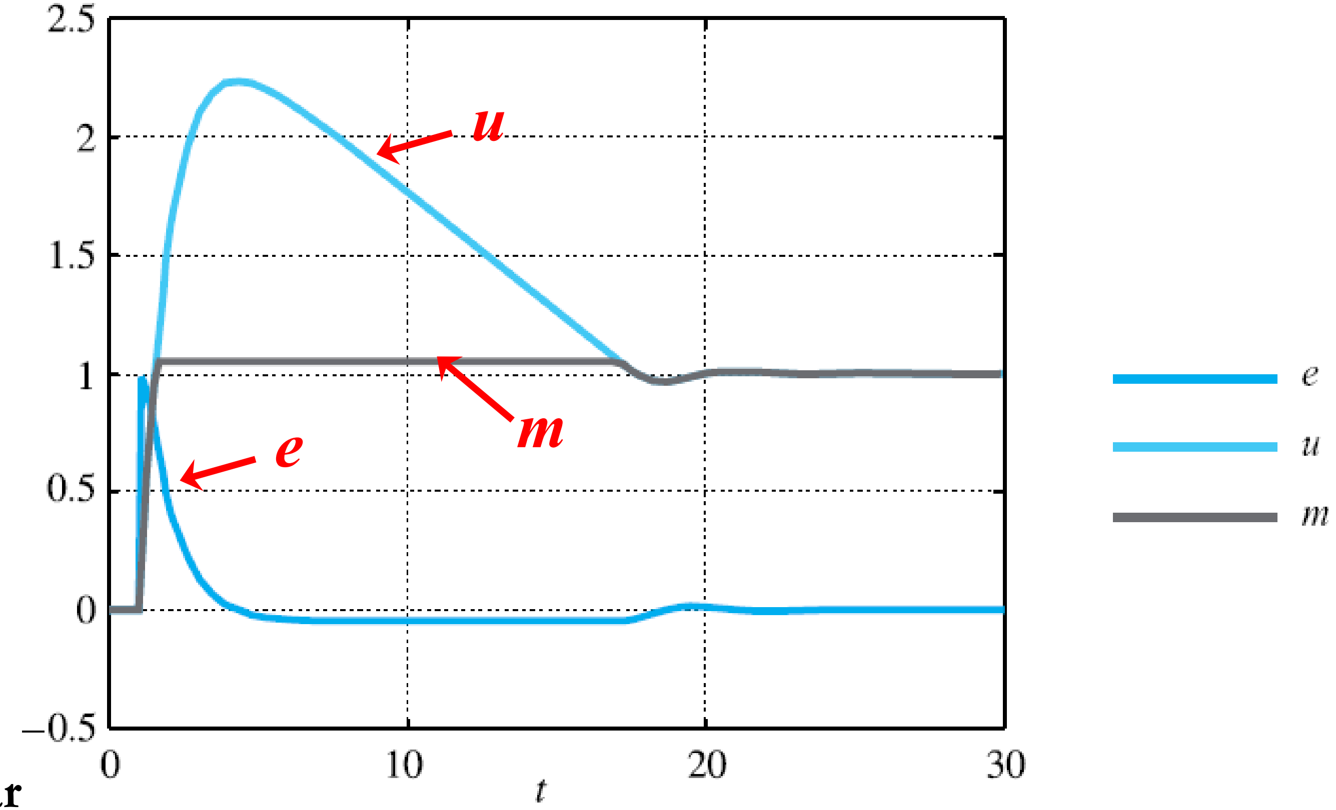
nonlinear (saturating) case

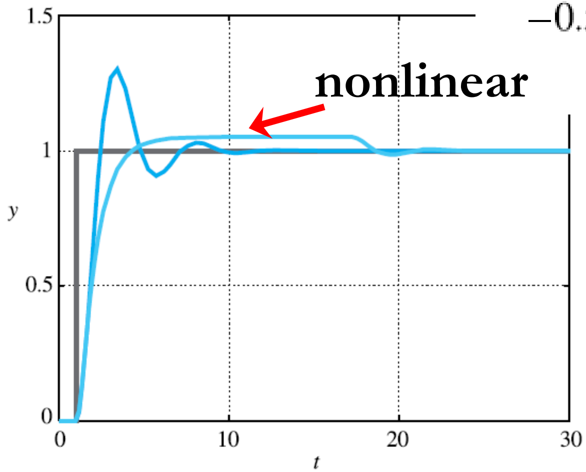
$$G(s) = \frac{1}{s+1}$$

$$K_I = 2$$

$$u_M = 1.05$$

PID controller: integral windup

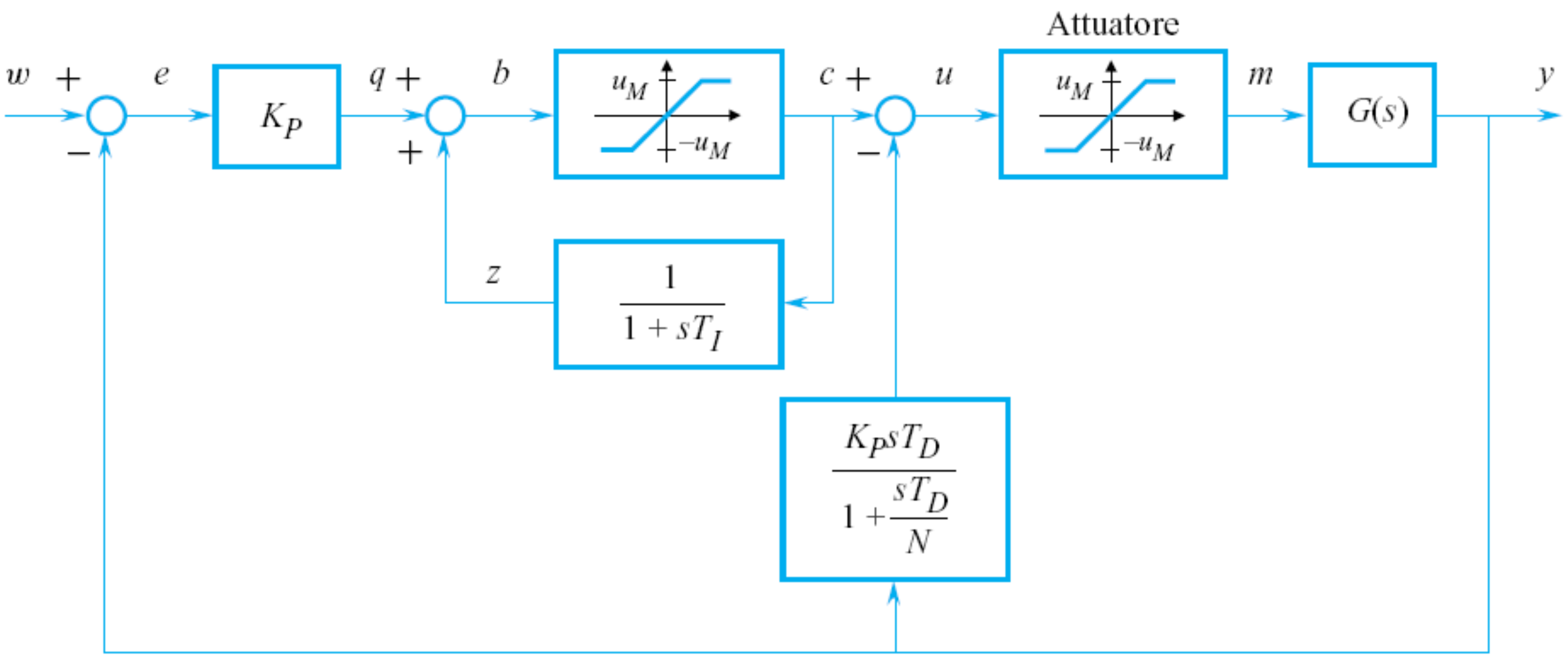




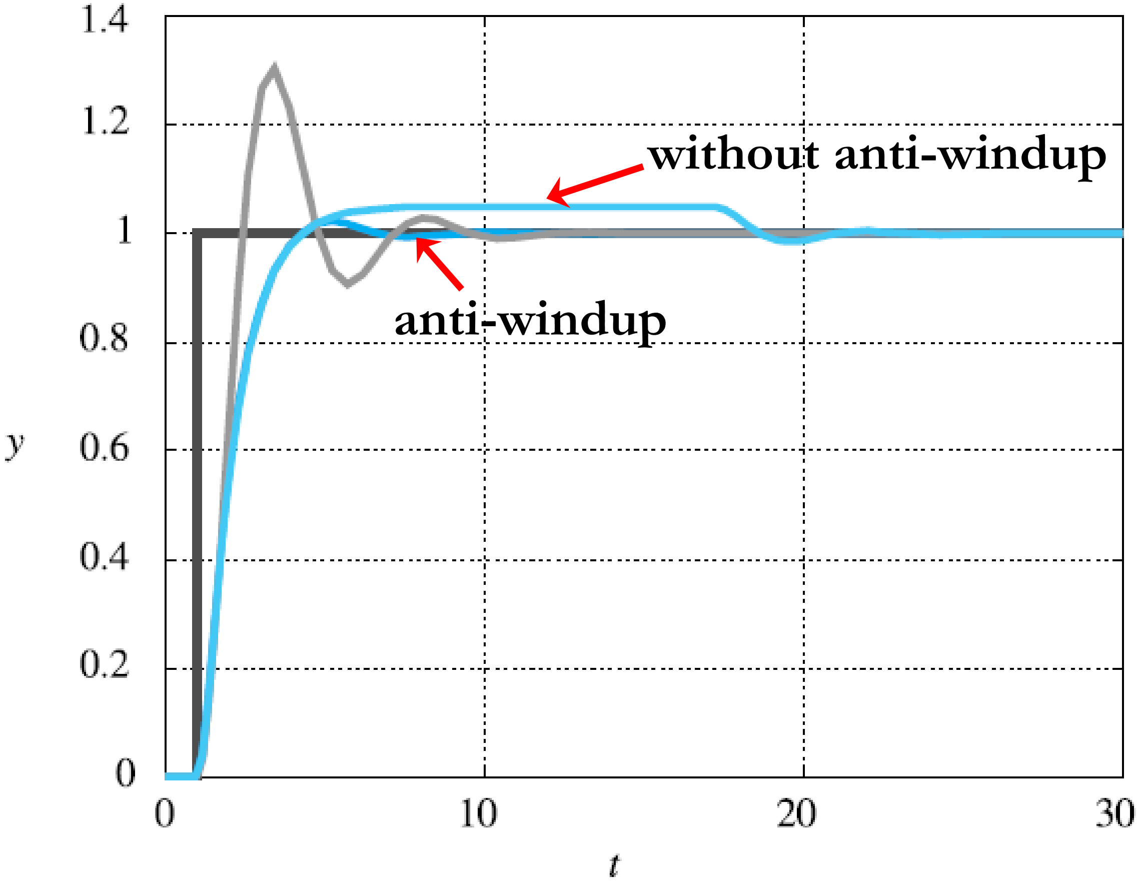
linear case

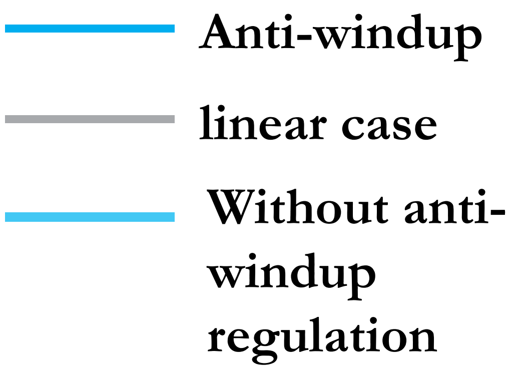
nonlinear (saturating) case

PID controller: anti-windup for integrator

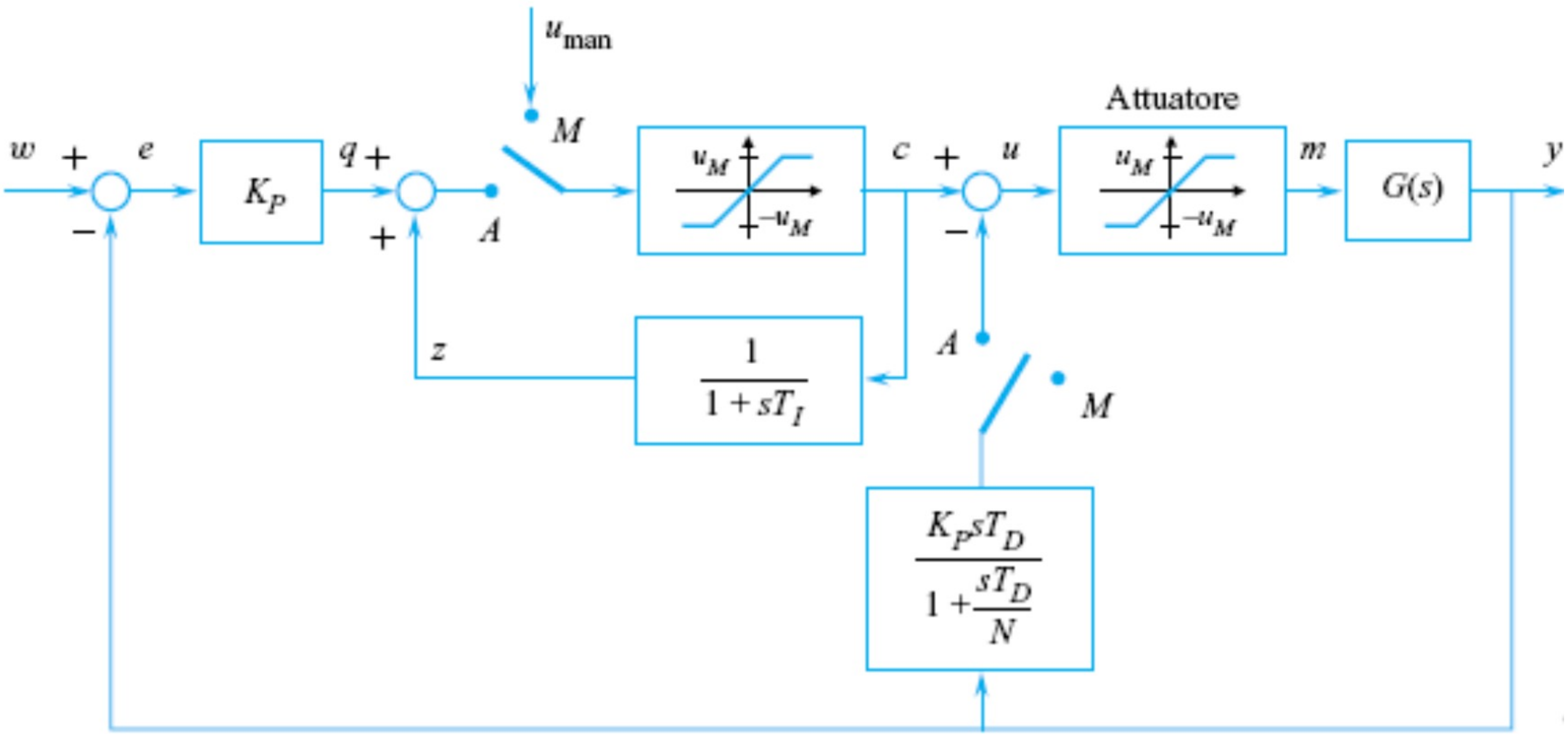


PID controller: anti-windup for integrator

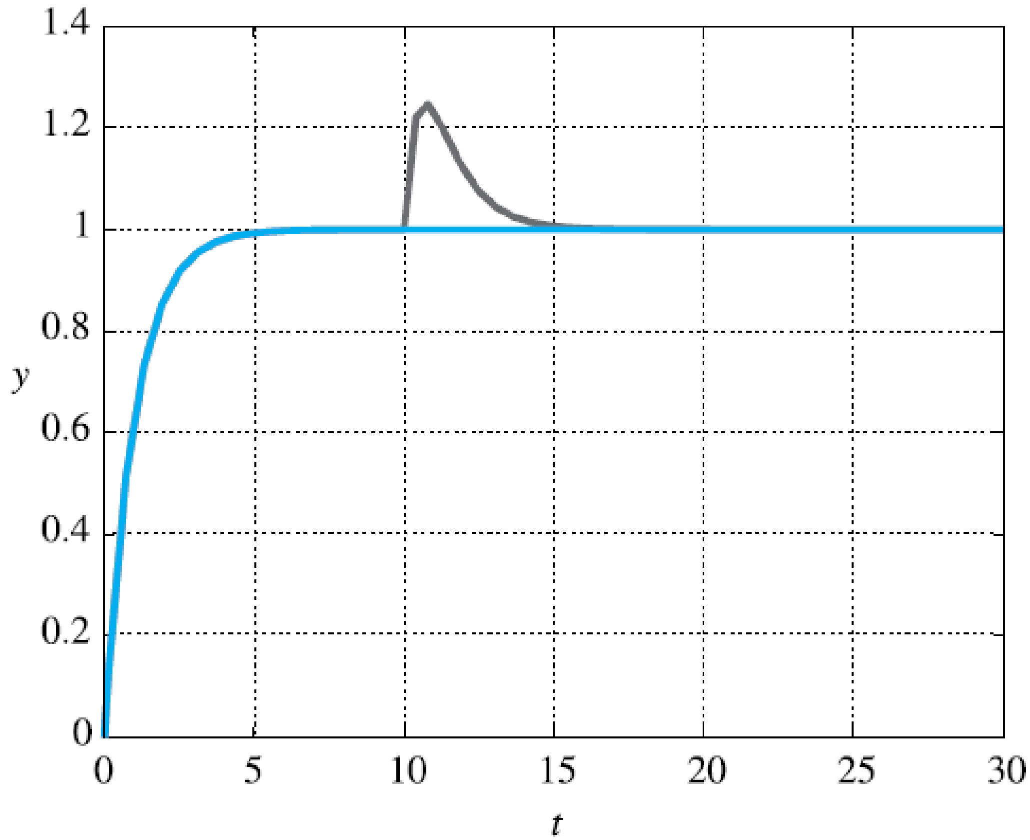




PID controller: bumpless transfer



PID controller: bumpless transfer



bumpless control transfer when switching from manual control to PID control

Without bumpless control

Ziegler-Nichols tuning methods

- The **Ziegler–Nichols tuning method** is a heuristic method of tuning a PID controller.
- Controller tuning is the process of determining the controller parameters which produce the desired output.
- It allows the optimization of a closed loop performance and minimizes the error between the variable of the process and its set point.
- The Ziegler-Nichols methods are trial and error method often used when the mathematical model of the system is not available
- The Ziegler-Nichols methods can be used for both **closed** and **open loop systems**

Ziegler-Nichols closed loop method

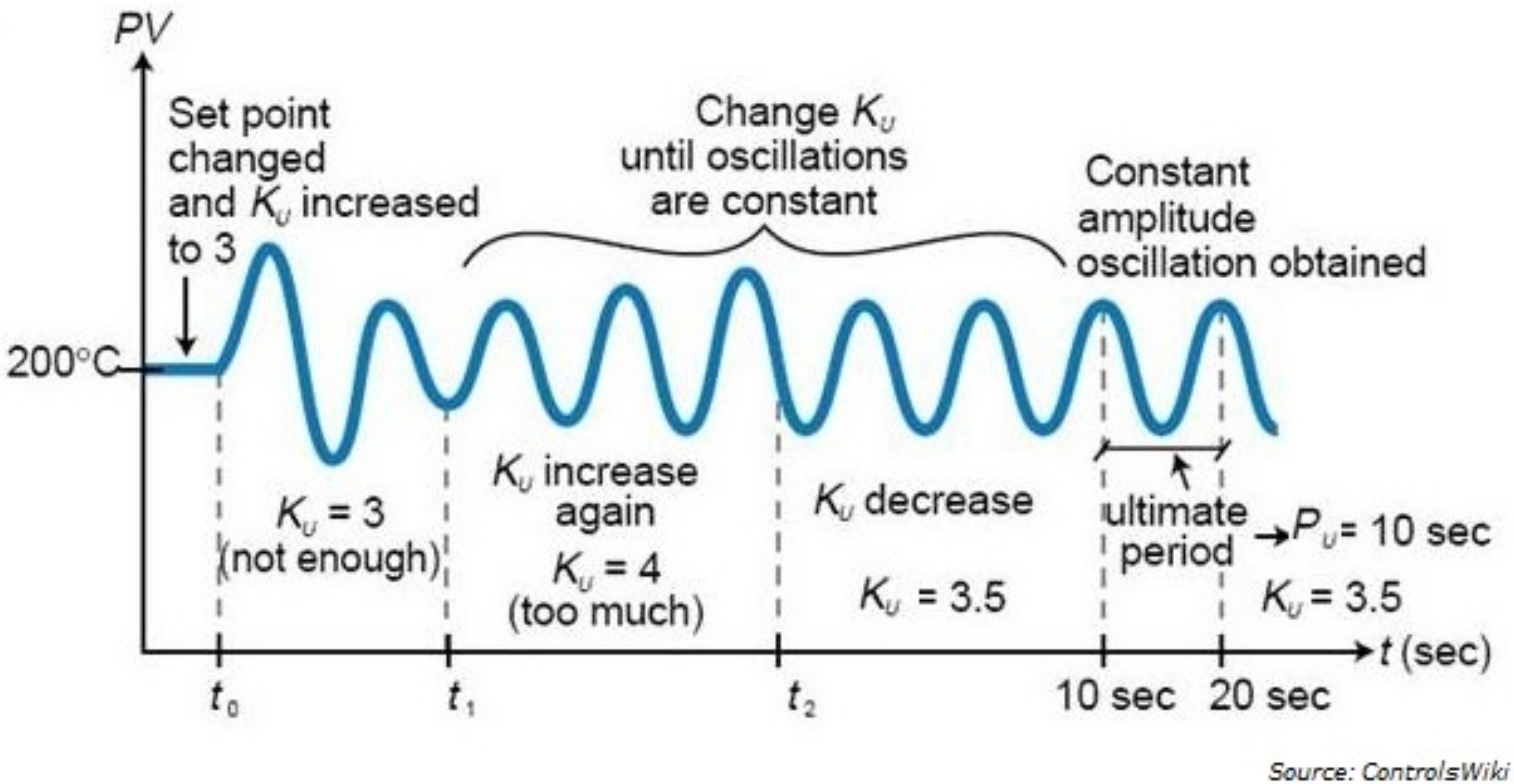
- The Ziegler-Nichols closed loop method is a simple method of tuning PID controllers by means of a number of tests carried out on the control system in closed loop with the PID controller

- It can be used only with regularly stable systems: closed loop systems moving toward the instability when the proportional gain increases

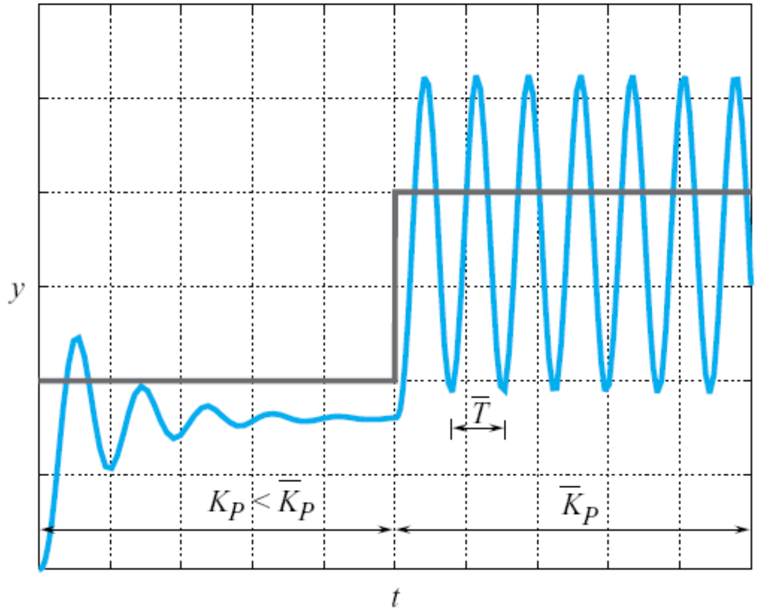
- Let us define **ultimate gain $\bar{K}_P$** the proportional gain which gives stable and consistent oscillations for closed loop systems

- $\bar{K}_P$ is found experimentally by starting from a small value of K_P and adjusting upwards until consistent oscillations are obtained. The integral and derivative actions are set to zero.

- Another important value associated with the ultimate gain $\bar{K}_P$ is **the ultimate period $\bar{T}$**. The ultimate period is the time required to complete one full oscillation while the system is at steady state


PV
Set point
changed
and K_u increased
to 3
Change K_u
until oscillations
are constant
Constant
amplitude
oscillation obtained
200°C
K_u = 3
(not enough)
K_u increase
again
K_u = 4
(too much)
K_u decrease
K_u = 3.5
ultimate
period
→P_u= 10 sec
K_u = 3.5
t (sec)
t_0
t_1
t_2
10 sec
20 sec
Source: ControlsWiki

- The parameters, $\bar{K}_P$ and $\bar{T}$ are used to find the loop-tuning constants of the controllers (P, PI, or PID)



	K_P	T_I	T_D
P	$0.5\bar{K}_P$		
PI	$0.45\bar{K}_P$	$0.8\bar{T}$	
PID	$0.6\bar{K}_P$	$0.5\bar{T}$	$0.125\bar{T}$

Ziegler-Nichols open loop method

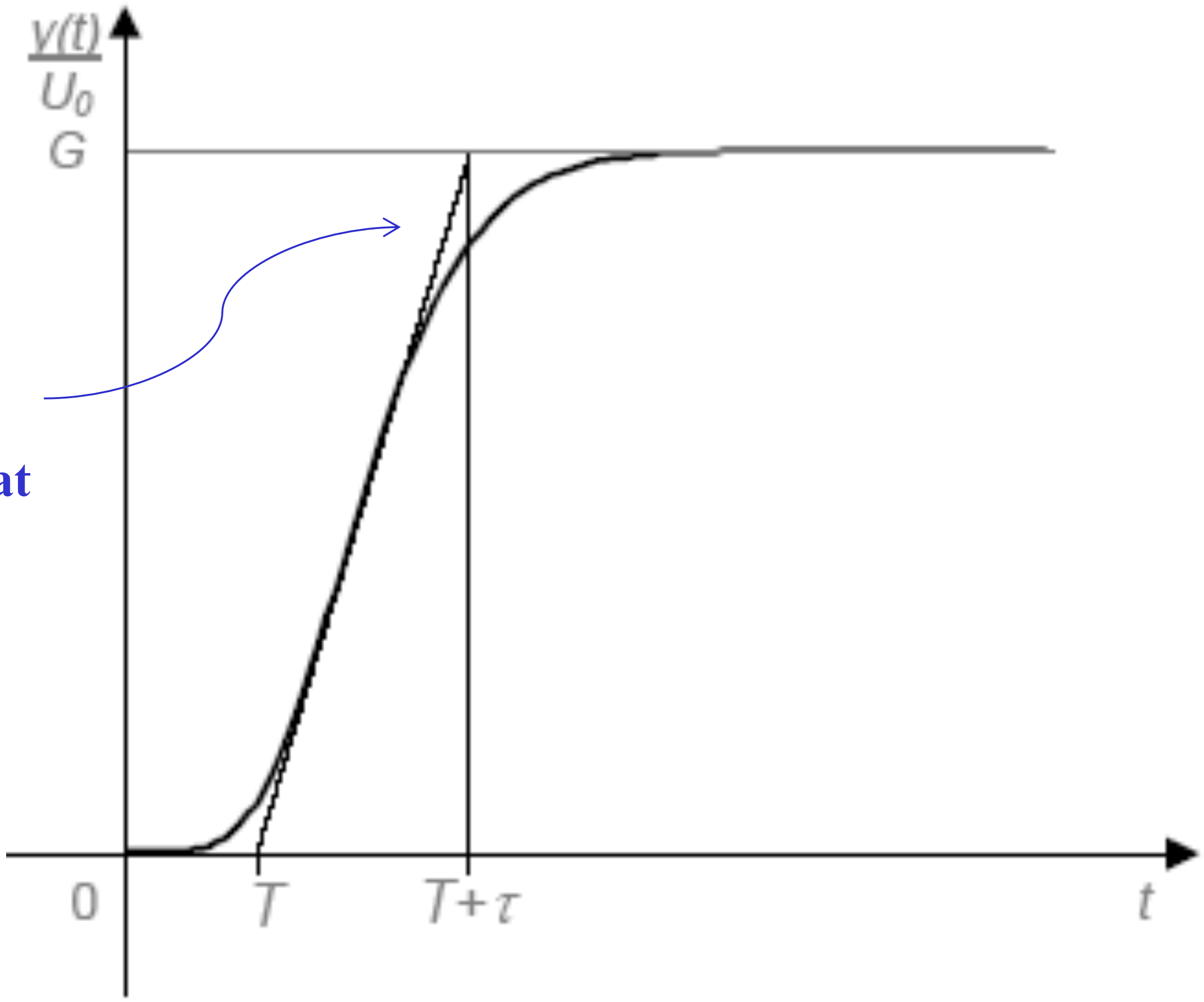
- The Ziegler-Nichols open-loop method is also referred to as a process reaction method, because it tests the open-loop reaction of the process to a change in the control variable output.

- The Ziegler-Nichols open loop method can be applied on processes whose step response doesn't oscillate.

- To use the Ziegler-Nichols open-loop tuning method, you must perform the following steps:
  - Evaluate the open loop unitary step response of the process
  - From the process reaction curve, determine
    - **the transportation lag or dead time** T,
    - **the equivalent time constant** τ
    - **the steady state value** G of the step response

Ziegler-Nichols open loop method

Tangent to the step response at the point of inflection



Ziegler-Nichols open loop method

Dead time T: time interval defined by the intersection of the tangent to the step response at the point of inflection and the time-axis

Equivalent time constant τ: Time interval from T to the intersection of the tangent to the step response at the point of inflection and the line indicating the steady-state value G

Ziegler-Nichols open loop method

- The parameters, ***T***, ***τ*** and ***G*** are used to find the loop-tuning constants of the controllers (P, PI, or PID)

	K_P	T_I	T_D
P	$\frac{\tau}{TG}$		
PI	$\frac{0.9\tau}{TG}$	$3T$	
PID	$\frac{1.2\tau}{TG}$	$2T$	$0.5T$

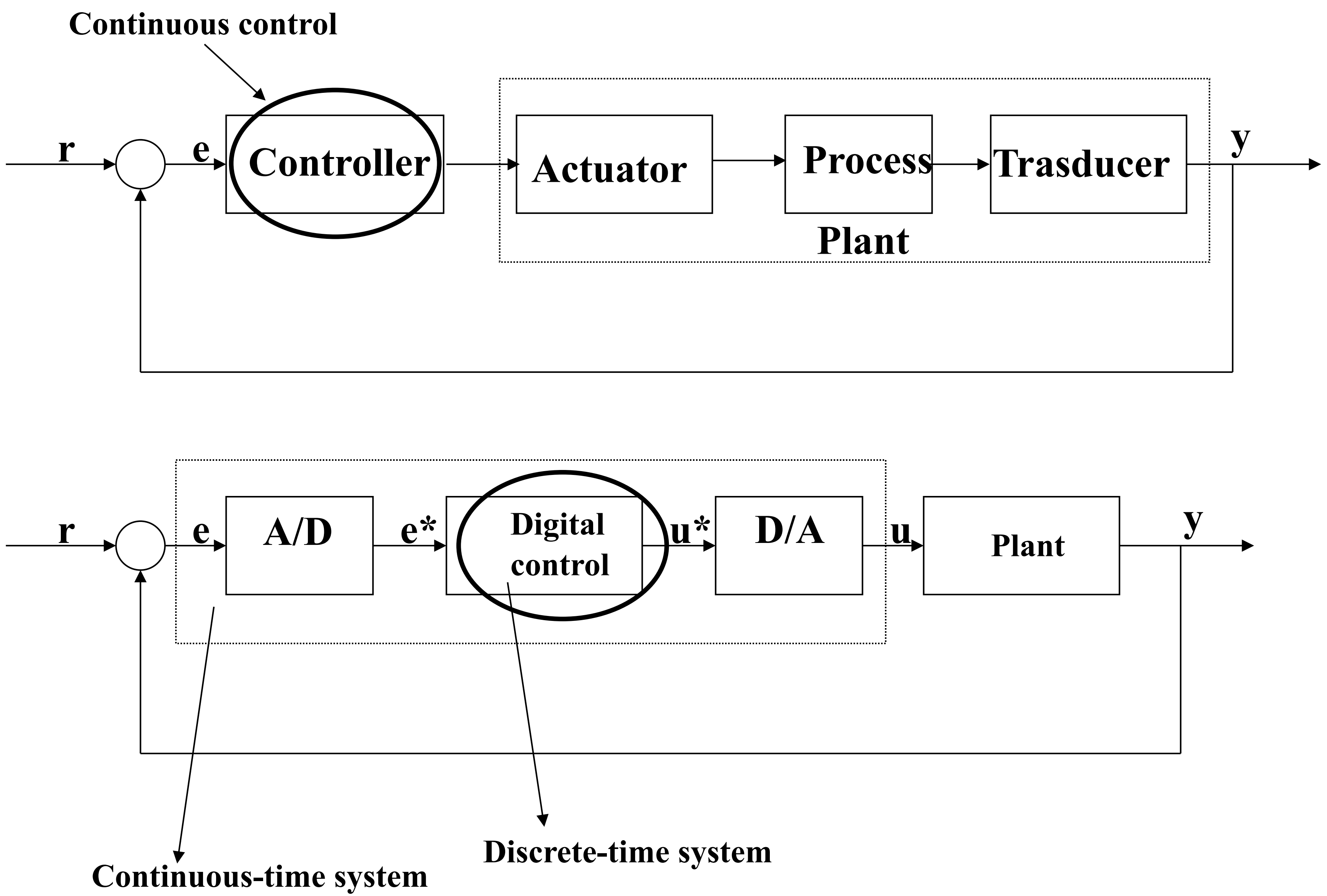
Other approaches for PID tuning

- By using optimization algorithms and numerical simulations it is possible to get the PID parameters that minimize the following cost indexes:
  - IAE (Integral Absolute Error): $\int |e| dt$
  - ISE (Integral Squared Error): $\int |e|^2 dt$
  - ITAE (Integral Time multiplied by Absolute Error): $\int t|e|dt\, dt$
  - ITSE (Integral Time multiplied by Squared Error): $\int t|e|^2 dt$
- Depending on the chosen index, the error (IAE, ISE) is weighted more than the final part of the transient (ITAE, ITSE).

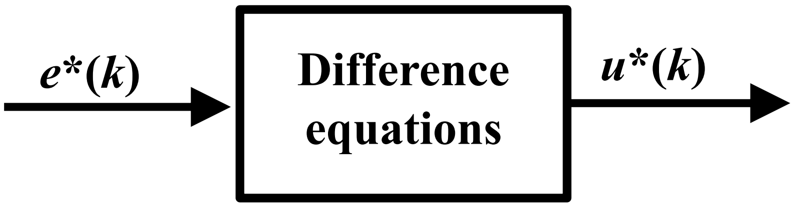
PID implementation

- PID controllers are widely used because they can be implemented using different technologies
  - Pneumatics
  - Hydraulics
  - Electrics
  - Digital electronics
- In the following, we will focus on the latter type
- We will assume that we have designed a continuous-time PID controller

Continuous vs. digital

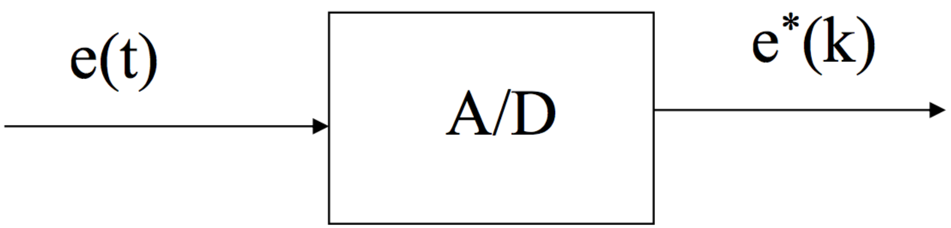


- Implementation of the digital control:



Analog-to-digital converter (ADC, A/D, or A-to-D)

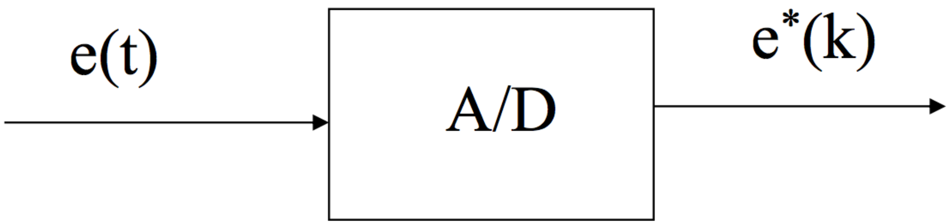
- The digital controller is a discrete-time system and the plant to be controlled is a continuous-time system.
- It is needed a device that transforms a continuous signal into a discrete one.



- Such device is the analog-to-digital converter (A/D).

Ideal sampler

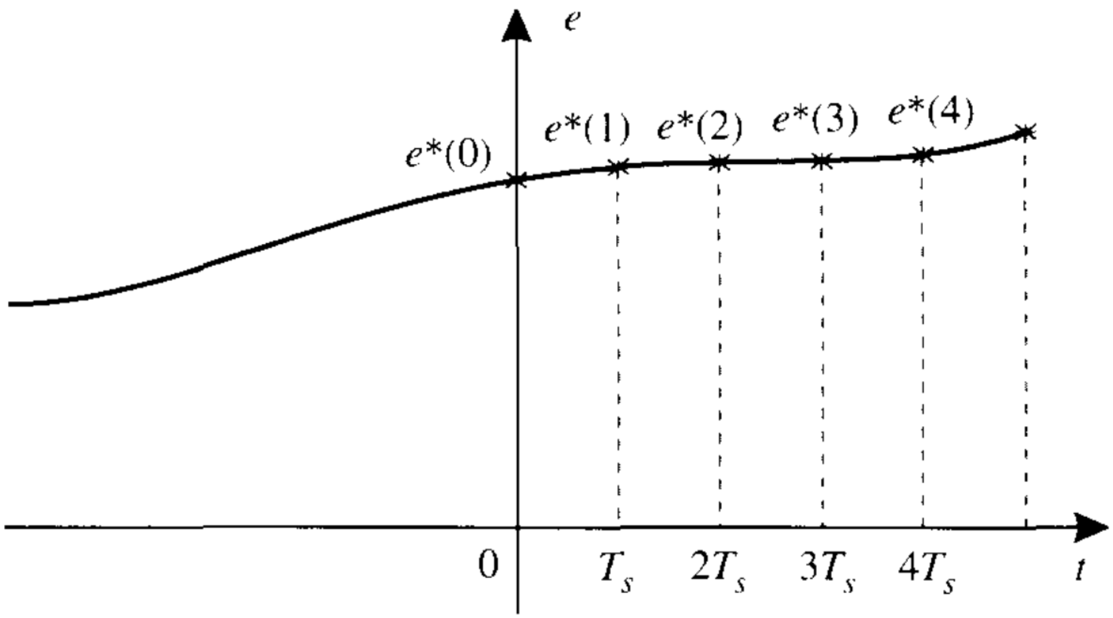
- The most common analog-to-digital converter is the sampler, which does the following

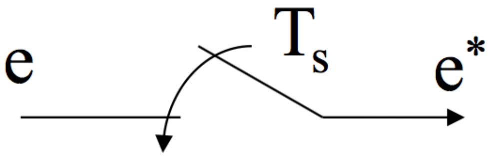


$$e^{*}(k) = e(kT_{s})$$

- Periodic sampling: the sampling instants are equally spaced, or k, i.e. $t_k = kT_s$ (k=0,1,2,..), with T_s representing the sampling time.
- The hold circuit holds the value of the sampled signal over a specified period of time.

Sampling operation





- $f_S = \frac{1}{T_S}$
- $\omega_S = 2\pi f_S = \frac{2\pi}{T_S}$.

Sampling operation

- The common problem when sampling a signal is the loss of information.

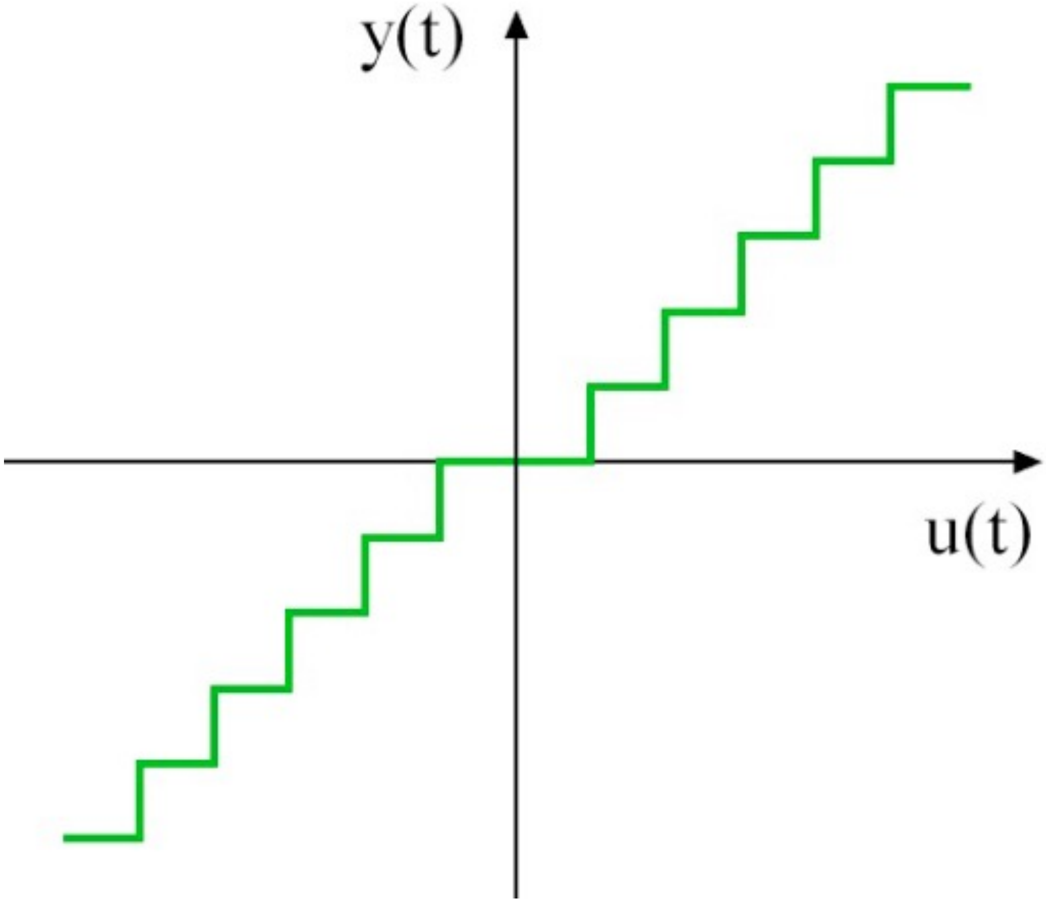
- Indeed, it is obvious that the same signal $e^*(k)$ can be generated by infinite continuous-time functions $e(t)$.

- Hence, given a signal $e^*(k)$ it is impossible to go back to the original signal $e(t)$.

Quantization

- The sampler defined above is ideal.
- It is assumed that in the sampling instants the value of e^* coincides with that of e.
- $e^*(k)$ is represented by a finite number of discrete states (by a numerical code)
- The process of representing a continuous or analog signal into a set of discrete state is called (amplitude) quantization.
- The output state of each quantized sample is then described by a numerical code (such a binary code): this process is called encoding.

Quantization



- The standard number system used for processing digital system is the binary number system
- n bits available, 2^n amplitude levels represented
- The quantization operation introduces a nonlinearity in the system
- When the number of digits of the binary representation is high enough, it is possible to neglect the effect of quantization

Digital-to-analog converter (DAC, D/A, or D-to-A)

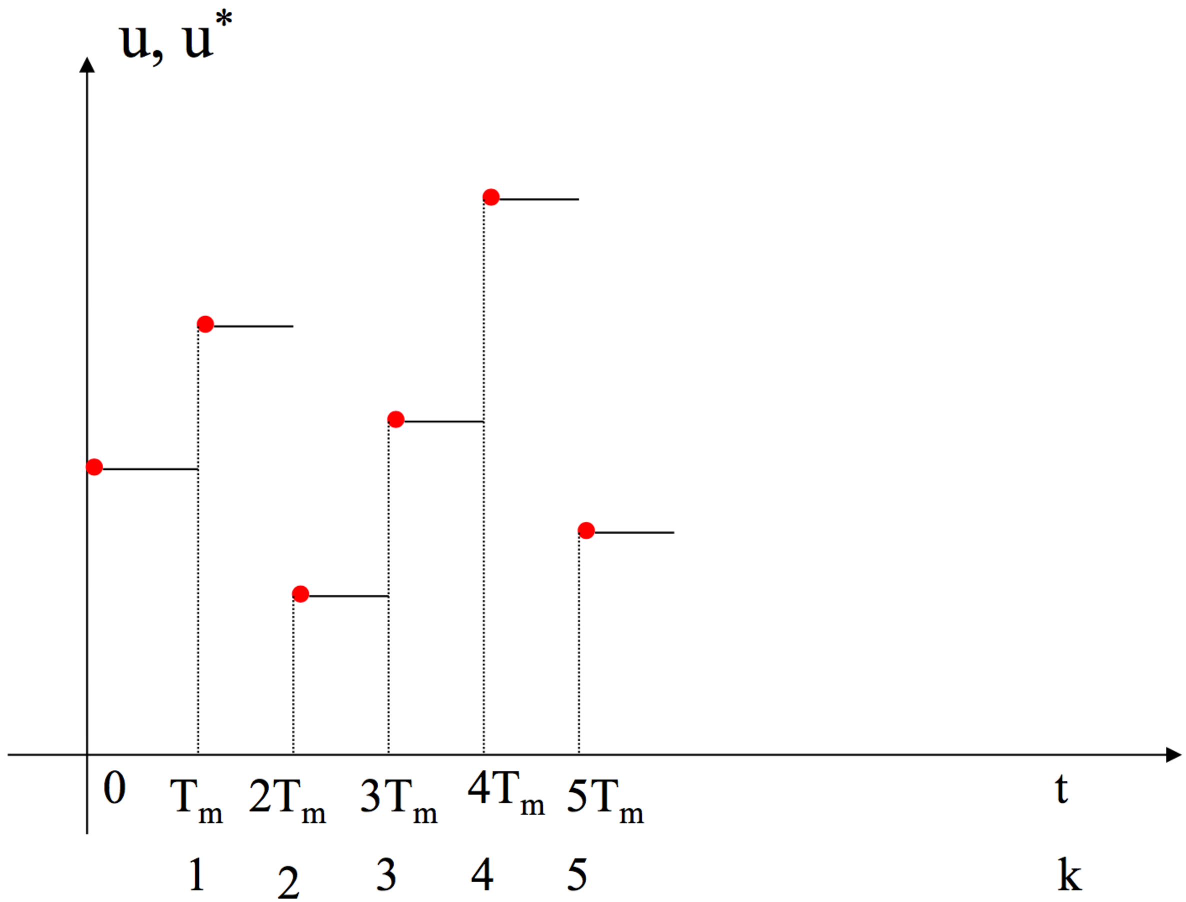
- It is a device that transforms a digital input (binary numbers) to an analog output.

- The most commonly used D/A converter is the zero order hold (ZOH), which operates as follows:

$$u(t) = u^{*}(k) \quad t \in [kT_m, (k+1)T_m]$$

- T_m is the sample time

ZOH circuit



Shannon's Theorem

In order for an analog signal (*e*(*t*)) to be reconstructed from its sampled version (*e(*k*)), by Shannon's theorem, it must have a strictly limited bandwidth and $\omega_S > 2\omega_B$ (with ω_B signal bandwidth).**

Euler's method – Difference equation for derivative

- ***y(t)***, **analog** signal; ***T*** is called the **sample period**; ***y(kT)***, the **sampled** signal with *k* integer value; it is often written simply as ***y(k)*** - we called this type of variable a **discrete signal**.
- From the definition of a derivative

$$\dot{y} = \lim_{\delta t \to 0} \frac{\delta y}{\delta t}$$

Even if δt is not quite equal to zero

$$\dot{y}(k) = \dot{y}(kT) \cong \frac{y\big((k+1)T\big) - y(kT)}{(k+1)T - kT} = \frac{y(k+1) - y(k)}{T}$$

This approximation can be used in place of all the derivatives that appear in the controller differential equations to arrive at a set of equation (called **difference equations**) that can be solved by a digital computer, respectively with time steps of length *T*.

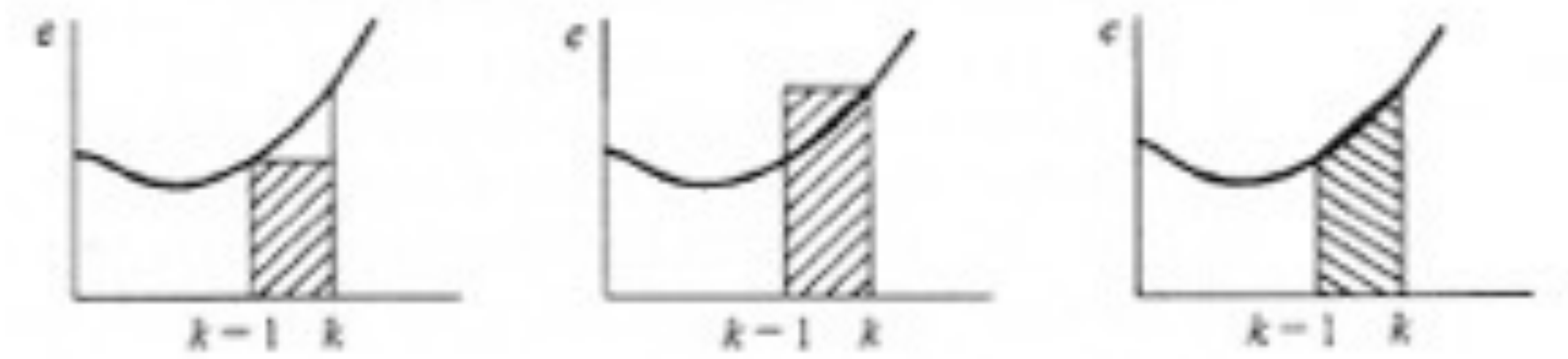
For systems having bandwidths of a few Hertz, sample rate are often on the order of 100 Hz, so that sample peridios are on the order of 10 msec and errors from this approximation can be quite small.

Difference equation for integration

We wish to compute an approximation for the integral of $\boldsymbol{e}(t)$ (numerical algorithms as discrete time systems).

Assume that we have an approximation for the integral from zero to the time $t_{k\text{-}1}$ (i.e. k-1) that is $u_{k\text{-}1}$.

We need an approximation of the area under the curve $\boldsymbol{e}(t)$ between k-1 and k (sampling period constant, i.e. $\boldsymbol{T}$)



$$u(k) = u(k-1) + Te(k-1)$$

rectangle of height $e(k-1)$

called **forward rectangular rule (or Euler's method)**

$$u(k) = u(k-1) + Te(k)$$

rectangle of height $e(k)$

called **backward rectangular rule (or Euler's method)**

$$u(k) = u(k-1) + \frac{T}{2}\left(e(k) + e(k-1)\right)$$

Trapezoid rule

Digital PID – position form

- By assuming T_c the sampling time of the error signal $e(t)$
- Discrete-time approximation of the controller

$$u_n = K_p e_n + \frac{K_p T_c}{T_i} \sum_{k=0}^{n} e_k + \frac{K_p T_d}{T_c} \left(e_n - e_{n-1}\right)$$

- Integral as summation of areas of rectangles
- Derivative as incremental ratio
- With this implementation (called position form), all samples for $k=0,\ldots,n$ should be stored in memory

Digital PID – velocity form

- By defining the deviation between the control action at step *n* and that at step *n*-1, we obtain the velocity form

$$\Delta u_n = u_n - u_{n-1} = K_p\left(e_n - e_{n-1}\right) + \frac{K_p T_c}{T_i} e_n + \frac{K_p T_d}{T_c}\left(e_n - 2e_{n-1} + e_{n-2}\right)$$

$$u_n = u_{n-1} + \Delta u_n$$

Digital PID – implementation by MCU

- Digital PID as an algorithm performed by a microcontroller (MCU), a microprocessor system on a single integrated circuit, including the processor itself along with memory and programmable input/output peripherals (**embedded system**).

- Here, a possible pseudo-code

```
previous_error = 0
integral = 0
loop:
error = setpoint - measured_value
integral = integral + error*dt
derivative = (error - previous_error)/dt
output = Kp*error + Ki*integral + Kd*derivative
previous_error = error
wait(dt)
goto loop
```

where

Kp - proportional gain; Ki - integral gain; Kd - derivative gain; dt - loop interval time

Arduino

- Arduino is a family of single-board devices, which includes, in the basic version
  - a microcontroller
  - Digital I/O
  - Analog-to-Digital Converters (ADC)
  - PWM outputs
  - USB connection for programming/power supply/data exchange
  - Timer
  - ICSP (In-Circuit Serial Programming) interface
- The system is also supported by a very simple software development environment, based on a language oriented to physical computing

Arduino UNO

- Atmel ATmega328 8-bit MCU
- 14 digital I/O (6 can work as PWM outputs)
- Clock: 16 MHz
- 6 analog inputs (10-bit ADC)
- Operating voltage: 5 V
- Supply voltage: 7-12 V
- Max continuous current on I/O pin: 40 mA
- Max continuous current on 3.3 V pin: 50 mA
- Flash memory: 32 KB
- SRAM: 2 KB
- EEPROM: 1 KB
- Price: €20