BASIC CONTROL SYSTEMS

6 PID CONTROL

HANSHU YU

NOVEMBER 2025

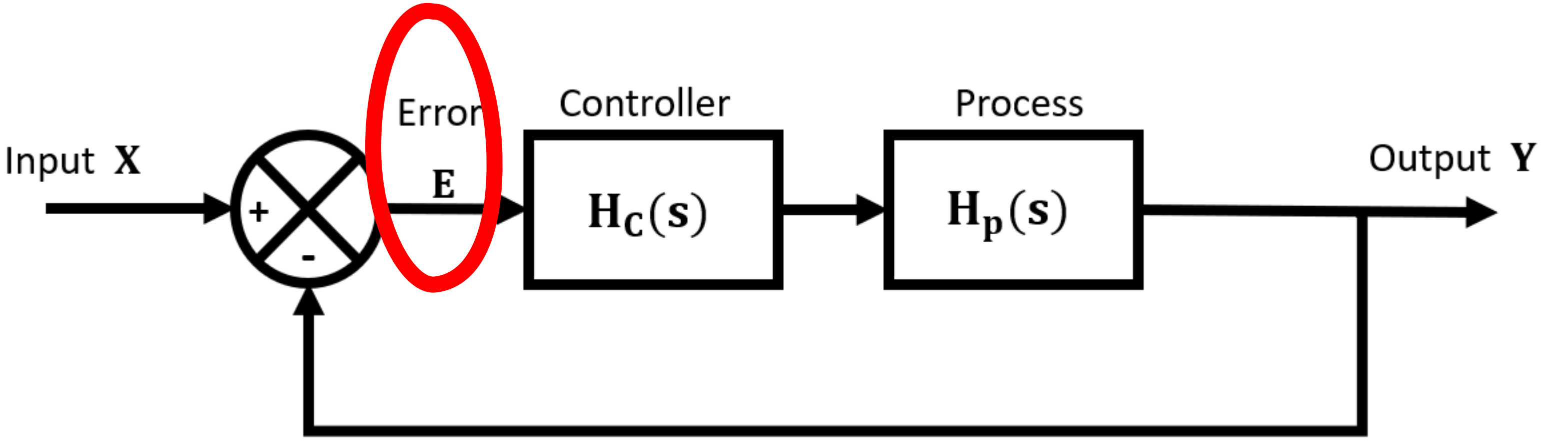
WHERE STUDENTS MATTER

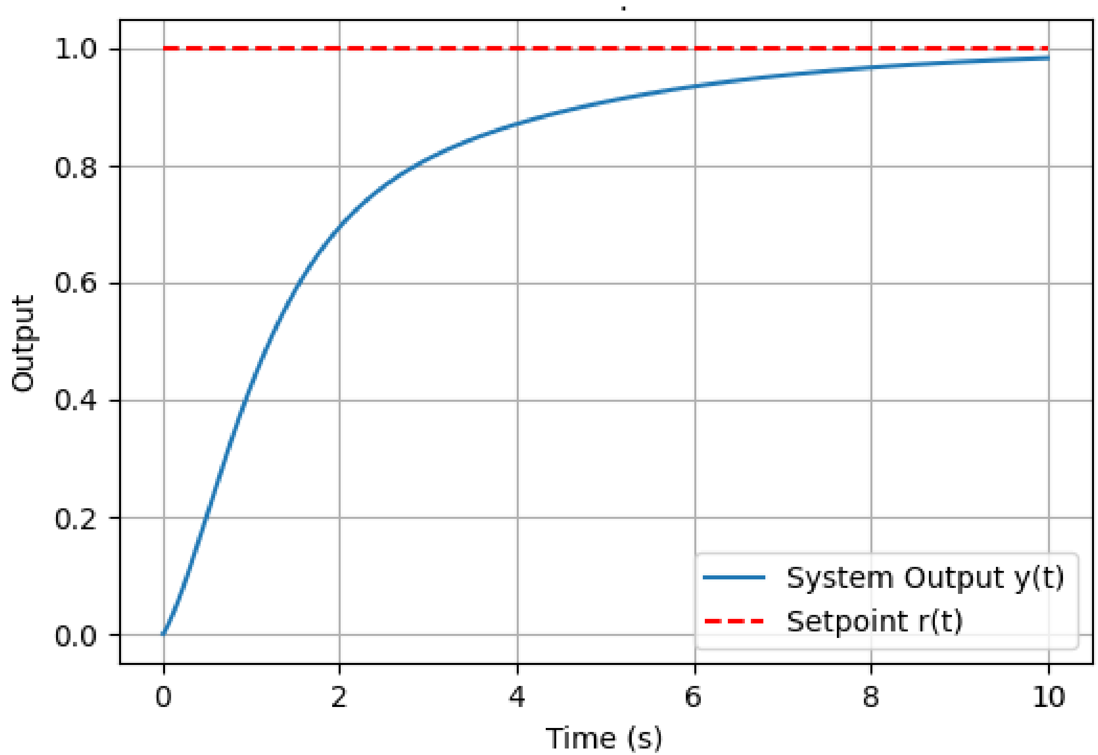
Introduction

- The PI(D)-type is the most popular controller in process control (over 80%)
- Good for linear process control
- Relatively easy to understand (important reason for wide popularity)
- But still, in reality many of the PID-control loops are poorly tuned...

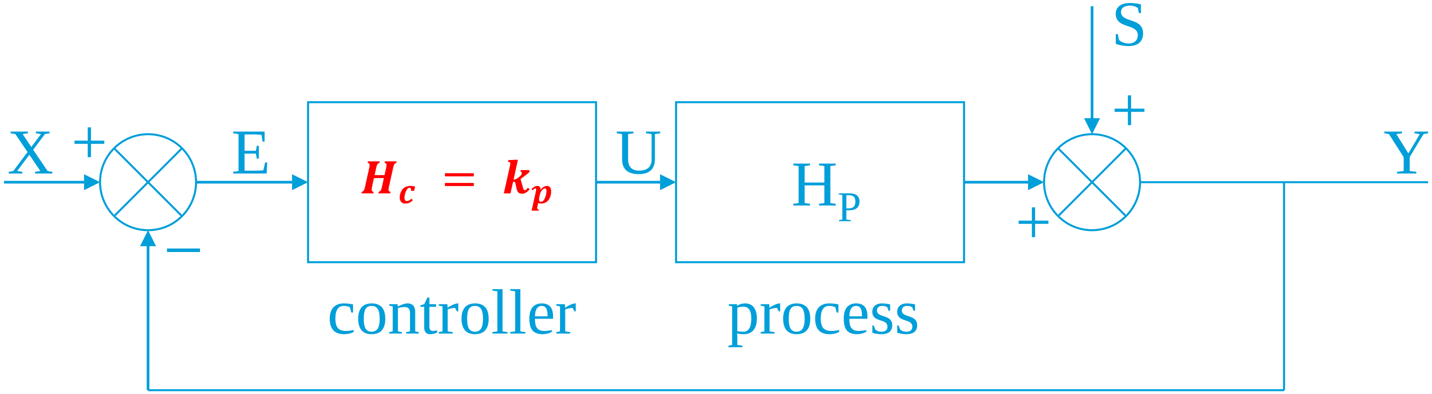
...in this lecture we will study the design and tuning of a PID-controller

Controlled processes





Controller: P (= proportional)



For convenience we assume $S = 0$

$$E = X - Y \;; \quad U = EH_c \;; \quad Y = UH_p$$

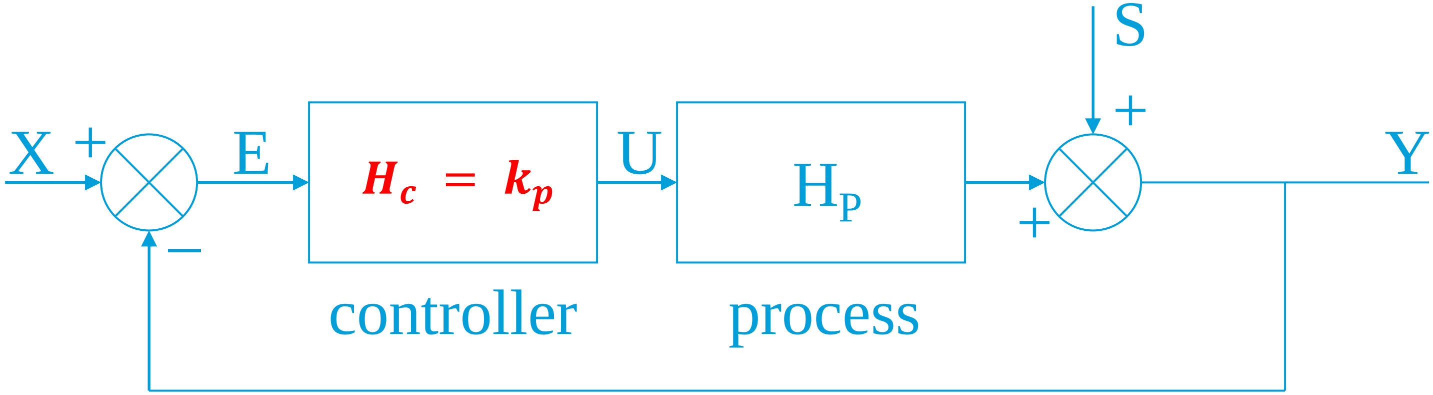
Together we get:

$$E = X - EH_cH_p$$

$$\boldsymbol{E = \frac{X}{1 + k_pH_p}}$$

If $X \neq 0$, then the only thing that the controller can do to make $E \rightarrow 0$ is to make $\boldsymbol{k_p} \rightarrow \infty$

Controller: P (= proportional)



This controller gives the basic control feedback loop.

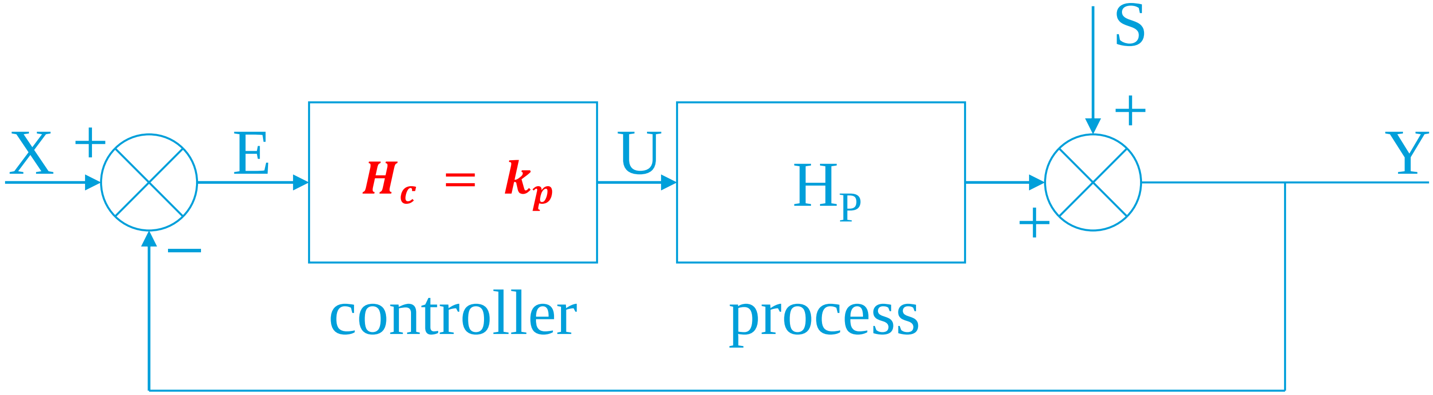
Advantage:

- The system reacts faster on deviations (faster than open loop)

Disadvantages:

- A possible instability at too high K-values
- An overshoot which is too large
- Steady-state error

Controller: P & steady-state error



S

X + E $H_c = k_p$ U H_P + + Y

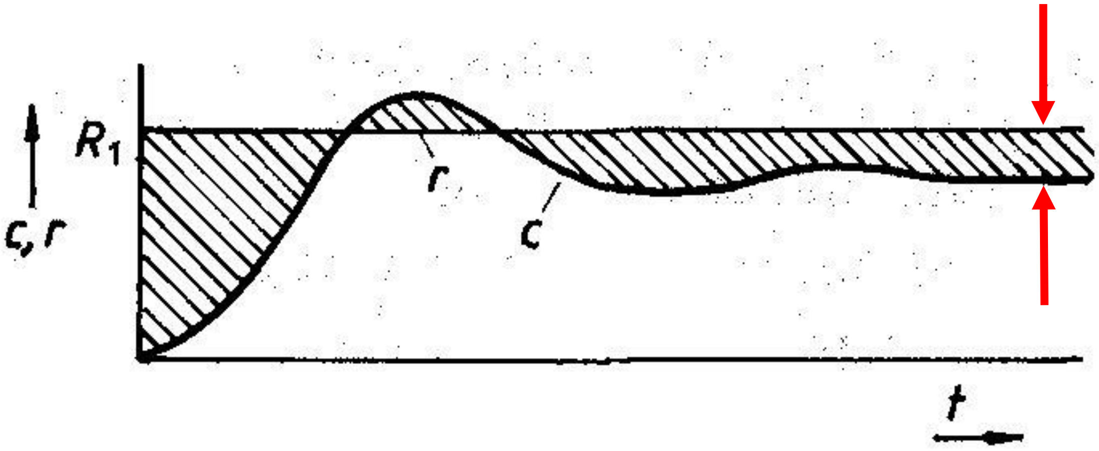
−

controller

process

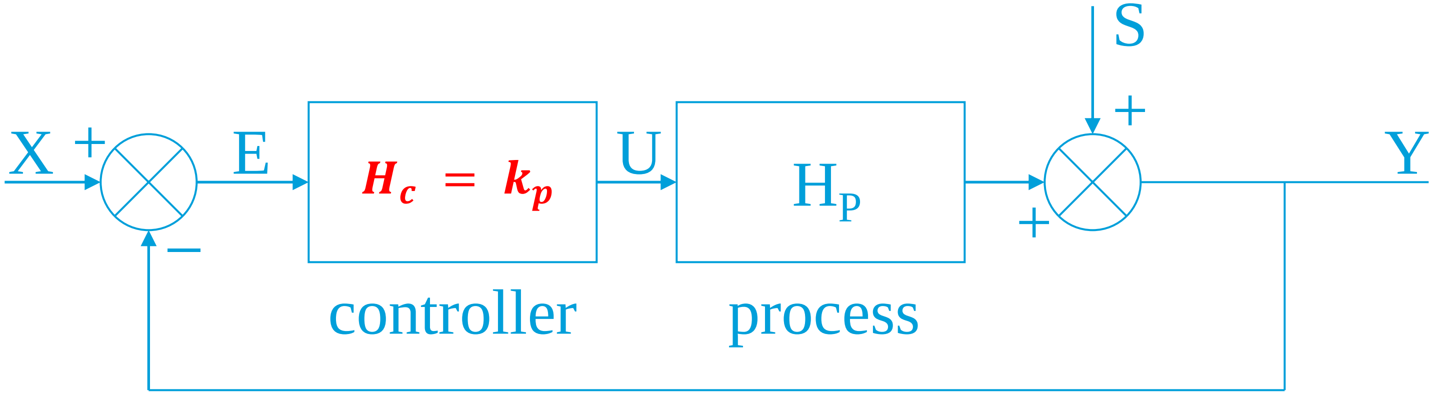
What happens with only a P-controller (or gain) and a step input?

Steady state error!



desired output *R* is different from the measured output *C*

Controller: P & steady-state error



Steady-state error! How big is the error?

Final value theorem

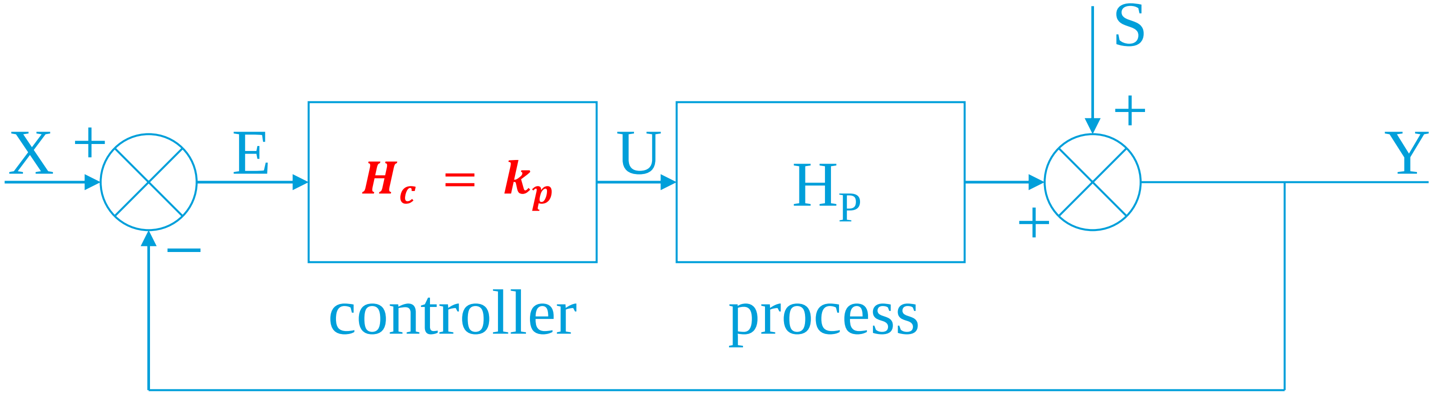
$$\boxed{\lim_{t\to\infty} e(t) = \lim_{s\to 0} s\,E(s)} = \lim_{s\to 0} s \frac{X(s)}{1 + H_c(s)H_P(s)}$$

Assume $X(s) = \frac{1}{s}$ (step input)

$$\lim_{s\to 0} s\,E(s) = \lim_{s\to 0} \frac{s\frac{1}{s}}{1 + k_p H_p(s)} = \lim_{s\to 0} \frac{1}{1 + k_p H_p(s)}$$

What do we do next????

Controller: P & steady-state error



Steady-state error! How big is the error?

Let's take a look at $H_p(s)$.

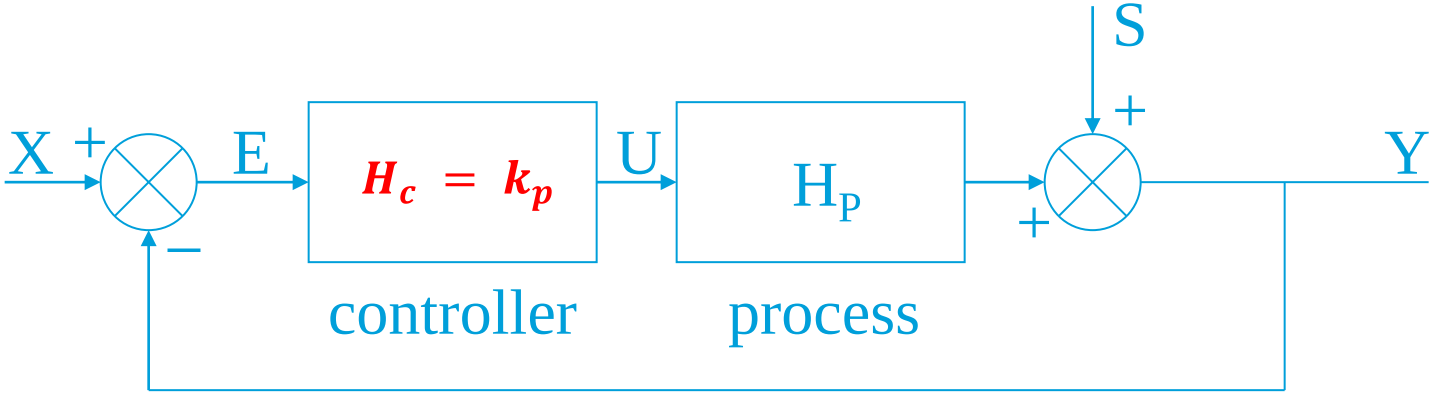
Although we make no assumption to $H_p(s)$ but we still can infer something…

Recall that we can write $H_p(s)$ in this format:

$$H_p(s) = K_{DC} \cdot \frac{(\frac{1}{z_1}s-1)(\frac{1}{z_2}s-1)\ldots..(\frac{1}{z_{m-1}}s-1)(\frac{1}{z_m}s-1)}{(\frac{1}{p_1}s-1)(\frac{1}{p_1}s-1)\ldots..(\frac{1}{p_{n-1}}s-1)(\frac{1}{p_n}s-1)}$$

$$\mathrm{K_{DC}} = \frac{b_m \prod_{k=0}^{m} z_k}{a_n \prod_{q=0}^{m} p_q}$$

Controller: P & steady-state error



Steady-state error! How big is the error?

Let’s take a look at $H_p(s)$.

Although we make no assumption to $H_p(s)$ but we still can infer something…

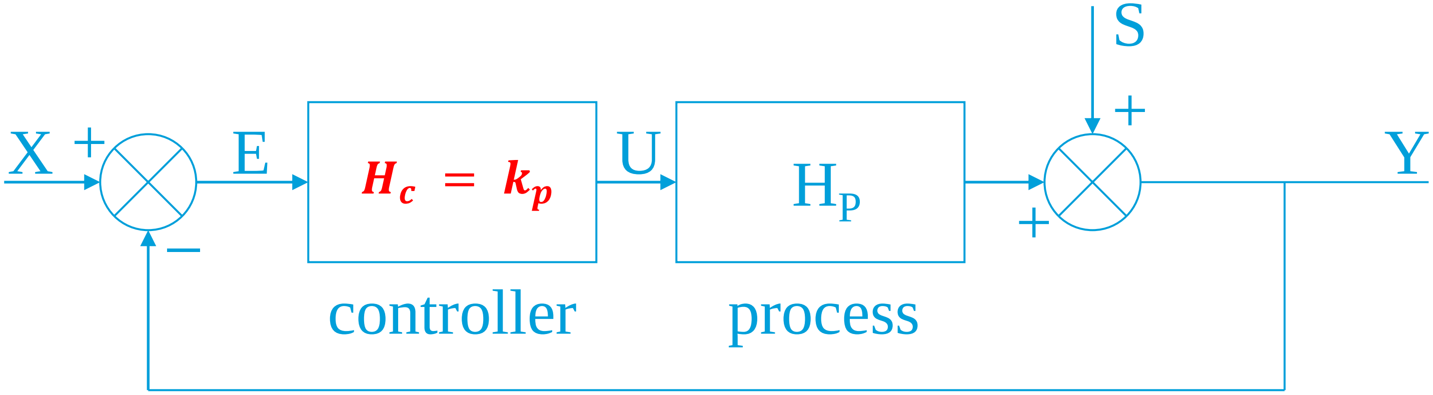
Recall that we can write $H_p(s)$ in this format:

$$\boxed{H_p(s) = K_{DC}} \cdot \cancel{\frac{(\frac{1}{z_1}s - 1)(\frac{1}{z_2}s - 1)\ldots..(\frac{1}{z_{m-1}}s - 1)(\frac{1}{z_m}s - 1)}{(\frac{1}{p_1}s - 1)(\frac{1}{p_1}s - 1)\ldots..(\frac{1}{p_{n-1}}s - 1)(\frac{1}{p_n}s - 1)}}$$

$$s \to 0$$

$$\mathrm{K_{DC}} = \frac{b_m \prod_{k=0}^{m} z_k}{a_n \prod_{q=0}^{m} p_q}$$

Controller: P & steady-state error



Steady-state error! How big is the error?

Final value theorem

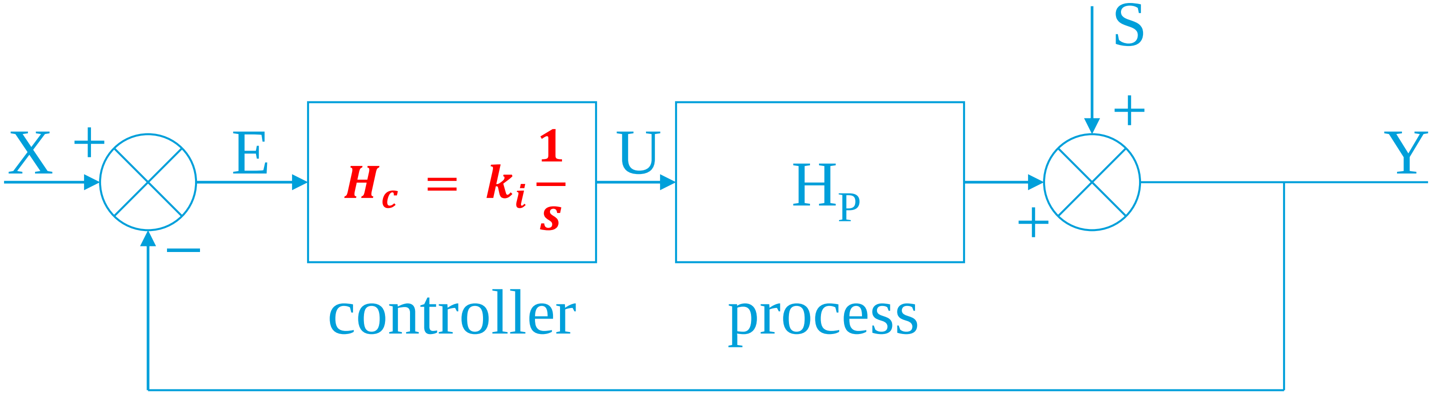
$$\boxed{\lim_{t\to\infty} e(t) = \lim_{s\to 0} s\,E(s)} = \lim_{s\to 0} s\,\frac{X(s)}{1 + H_c(s)H_P(s)}$$

Assume $X(s) = \frac{1}{s}$ (step input)

$$\lim_{s\to 0} s\,E(s) = \lim_{s\to 0} \frac{s\frac{1}{s}}{1 + k_p H_p(s)} = \lim_{s\to 0} \frac{1}{1 + k_p H_p(s)}$$

Steady state error $= \dfrac{1}{1 + k_p \cdot K_{DC}}$

Controller: I (integral)



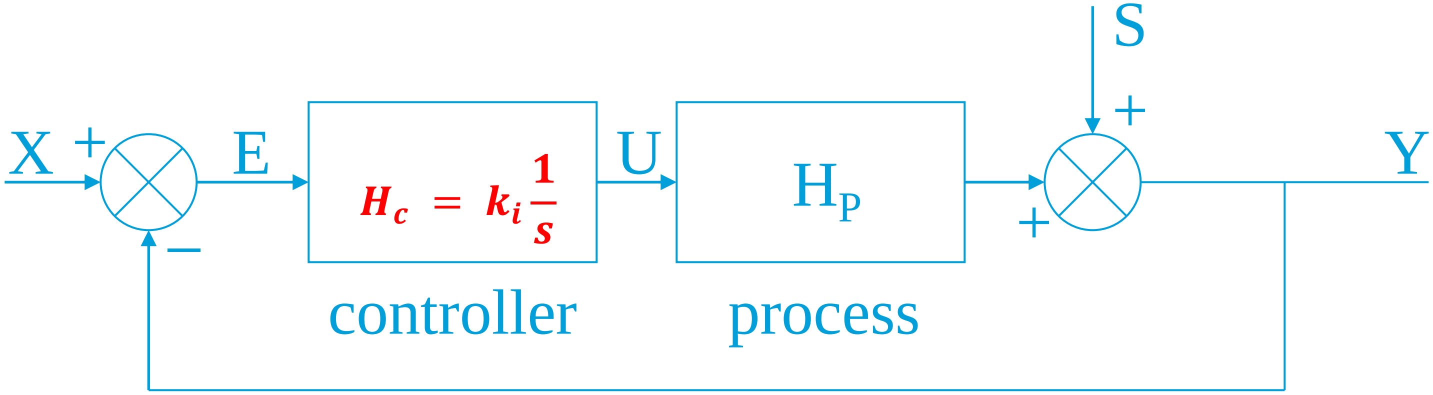
Advantage:

- eliminates the steady-state error of the P-controller

Disadvantages:

- a possible instability at too large k_i -values
- too slow at too small k_i -values

Controller: I & steady-state error



Steady state error? How big is the error?

Final value theorem

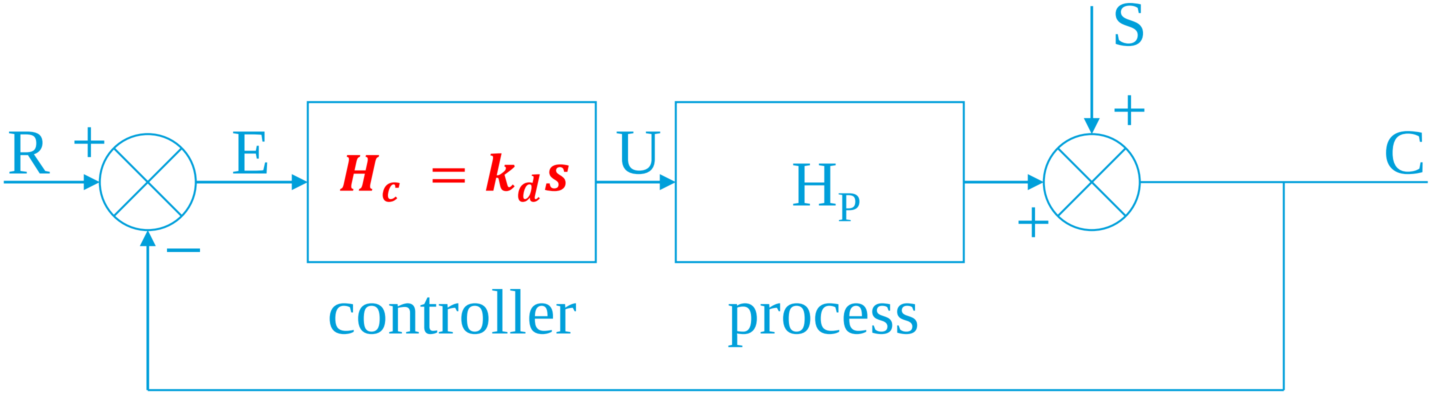
$$\boxed{\lim_{t\to\infty} e(t) = \lim_{s\to 0} s\,E(s)} = \lim_{s\to 0} s \frac{X(s)}{1 + H_c(s)H_P(s)}$$

Assume $X(s) = \frac{1}{s}$ step input

$$\lim_{s\to 0} s\,E(s) = \lim_{s\to 0} \frac{s\frac{1}{s}}{1 + \frac{k_i H_p(s)}{s}} = \lim_{s\to 0} \frac{s}{s + k_i H_p(s)} = \frac{0}{0 + k_i K_{DC}}$$

Steady state error $= \mathbf{0}$

Controller: D (= derivative)



This controller is never used alone…

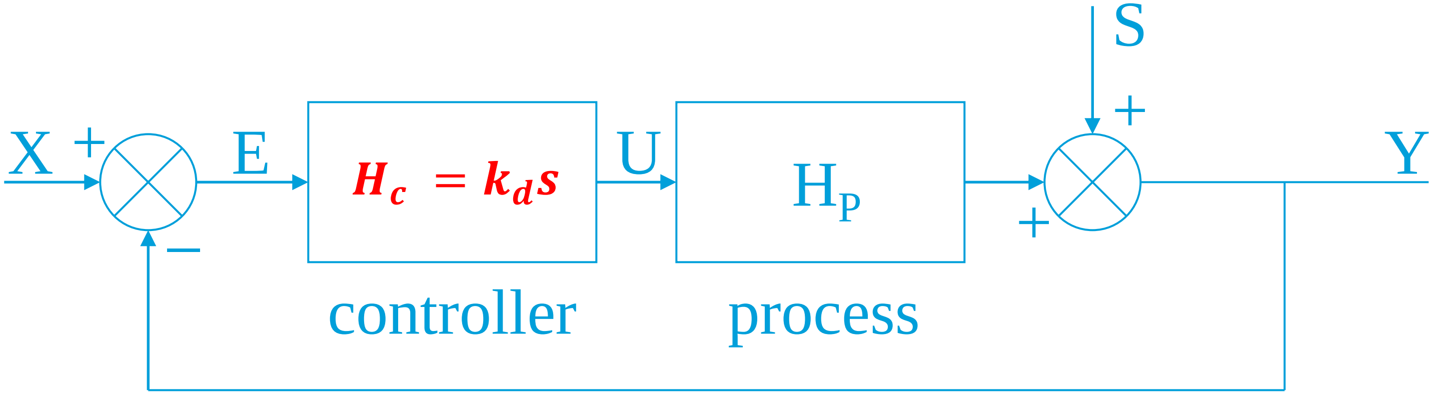
Advantage:

- a D-action has most of the time a stabilising effect on the control loop
- makes the dynamics of the response better (faster)

Disadvantages:

- A possible unstable behaviour at too large k_d -values
- Vulnerable to noise

Controller: D & steady-state error



Steady-state error! How big is the error?

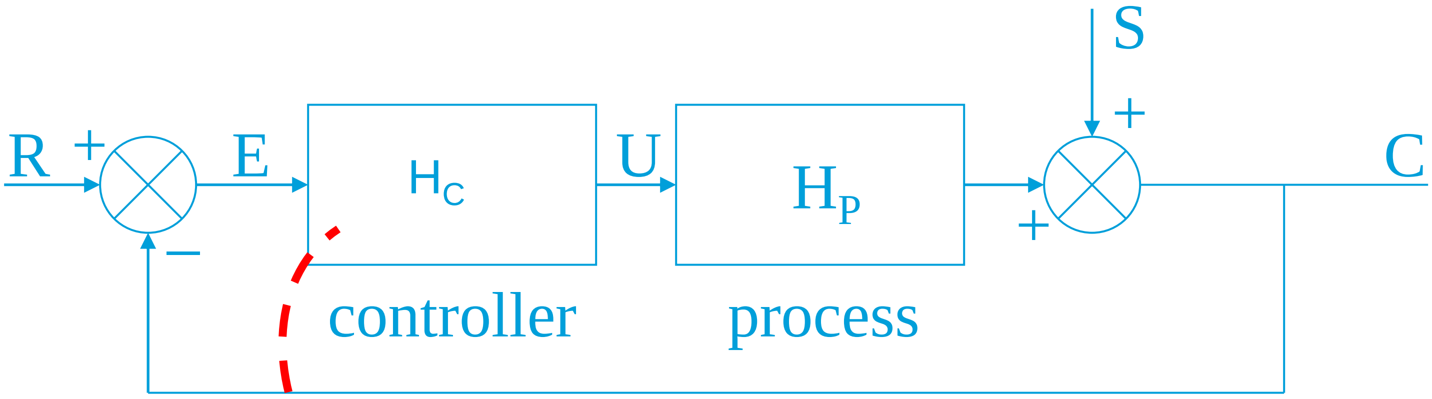
Final value theorem
$$\boxed{\lim_{t\to\infty} e(t) = \lim_{s\to 0} s\,E(s)} = \lim_{s\to 0} s\,\frac{X(s)}{1 + H_c(s)H_P(s)}$$

Assume $X(s) = \frac{1}{s}$ (step input)

$$\lim_{s\to 0} s\,E(s) = \lim_{s\to 0} \frac{s\frac{1}{s}}{1 + k_d s H_p(s)} = \lim_{s\to 0} \frac{1}{1 + k_d s H_p(s)}$$

Steady state error $= \frac{1}{1+0} = 1$

Controller: PI (proportional + integral)

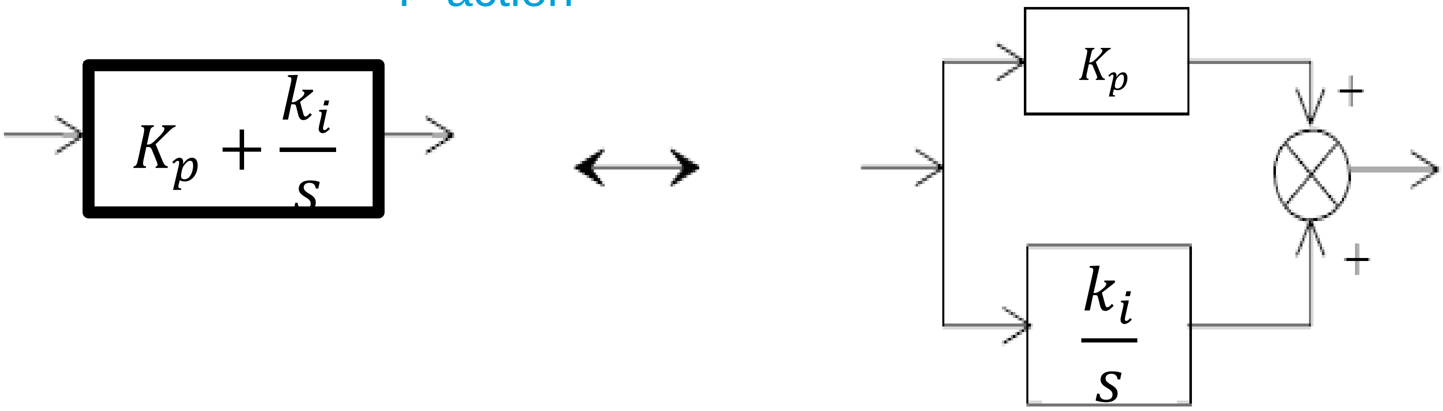


$$H_C = k_p + \frac{k_i}{s}$$

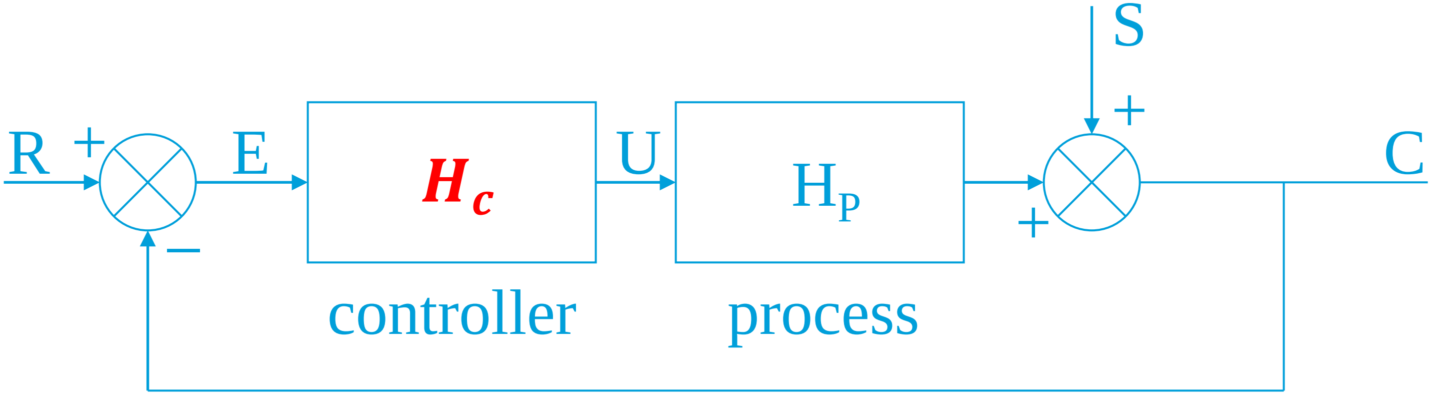
I-action

P-action

Parallel Structure



Controller: PD (proportional + derivative)



$H_C = k_p + k_d s$

Advantage:

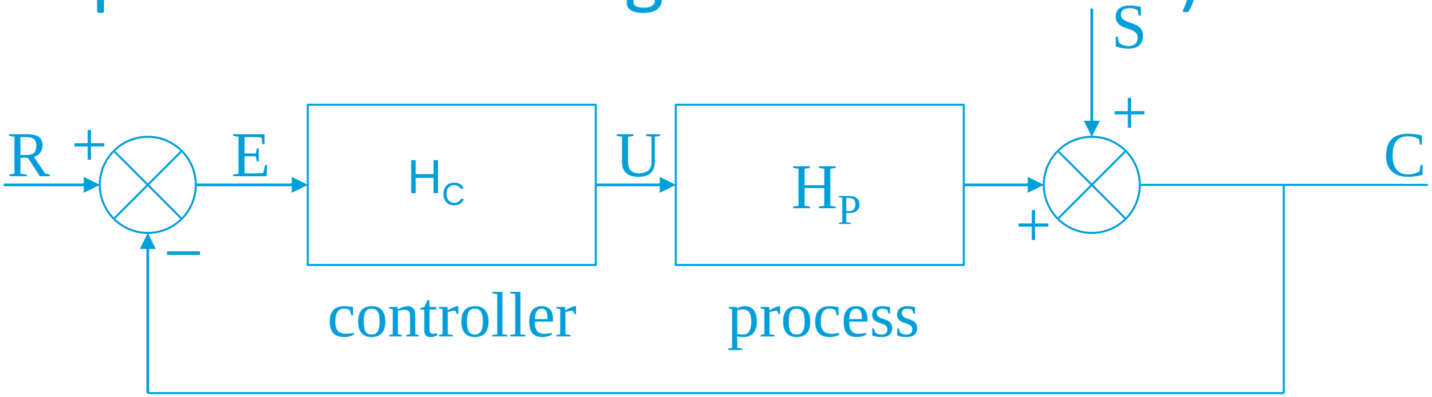
- combination of P-action and D-action

Disadvantages:

- Steady state error

Controller: PID

(proportional + integral + derivative)



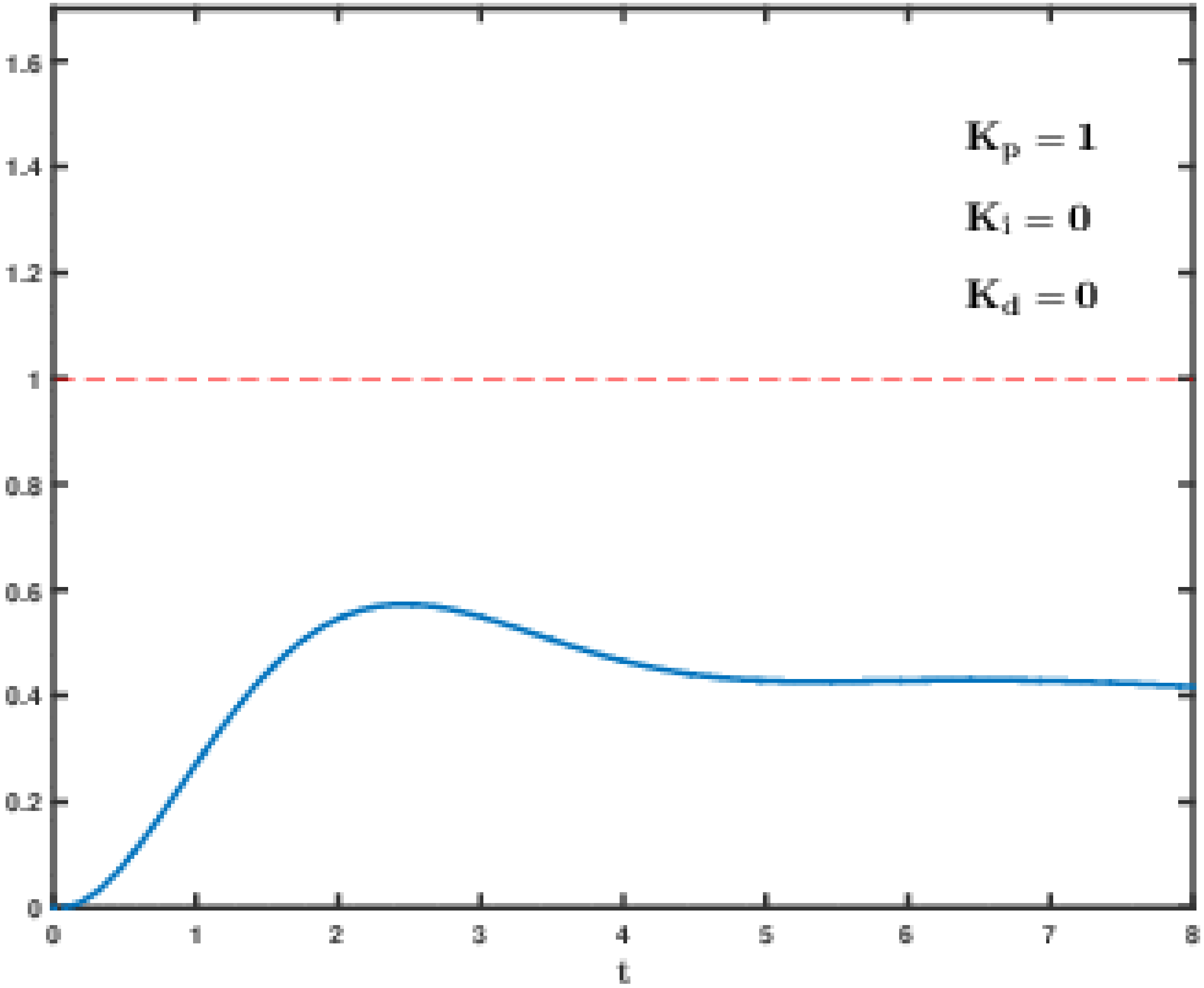
$$H_C = k_p + \frac{k_i}{s} + k_d s$$

Advantage:

- combination of P-action, I-action and D-action
- this is the most flexible controller

However, it is always better to use an easier controller (like P, PI or PD) if you do not need a certain action.

PID actions



Attention!

Different implementations of the PID-controller

1. We use the parallel controller (Ziegler)

$$H_C(s) = k_p + \frac{k_i}{s} + k_d s$$

$$= K_c\left(1 + \tau_d s + \frac{1}{\tau_i s}\right)$$

$$K_c \frac{(1+\tau_d)s + \frac{1}{\tau_i}}{\tau_i s}$$

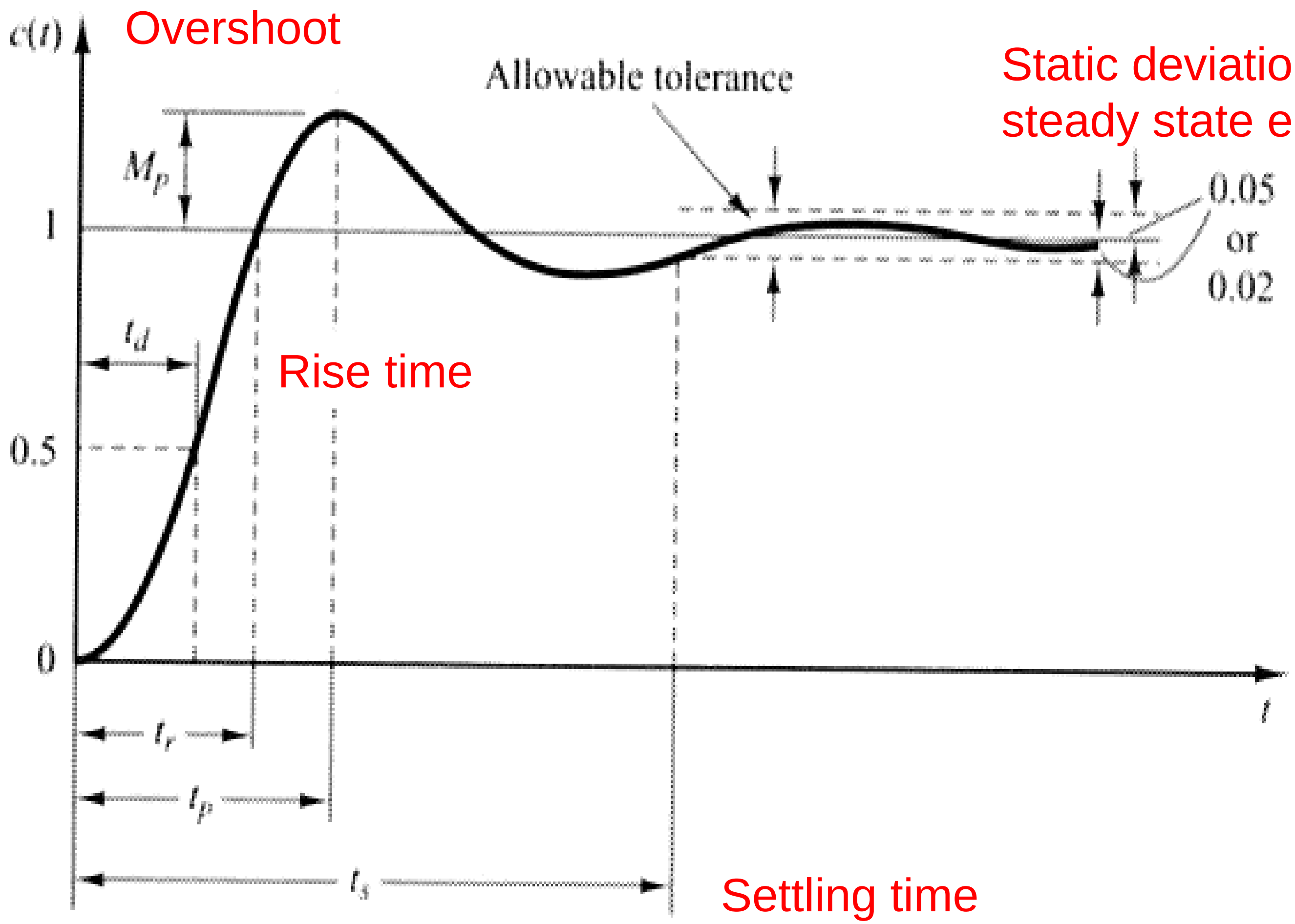
2. However, other implementations also exists, for example:

$$H_C(s) = K_c\left(\frac{\tau_d s + 1}{\frac{\tau_d}{5}s + 1} + \frac{1}{\tau_i s}\right)$$

3. The structure of the ‘default’ controller in Matlab/Simulink is again different

“IDEAL” CONTROLLER

What do you want for all these specifications?



Overshoot

Static deviation or steady state error

Rise time

Settling time

Figure 5-8 from Ogata

“Ideal” controller

The “ideal” controlled system will have:

1. rise time: short
2. settling time: short
3. overshoot:
   - no undershoot or
   - else a certain maximum value, for example 10%
4. steady state error
   - none (preferable) or
   - else as little as possible

It is possible to specify these parameters to tune a controller.

However, in this course we will use practical tuning rules…

Practical tuning rules for a PID-controller

1. Step response method of Ziegler and Nichols

- to be used after process analysis of the step input response
- based upon a delayed 1st order process description

$$H_P(s) = \frac{a}{\tau s} e^{-\tau s}$$

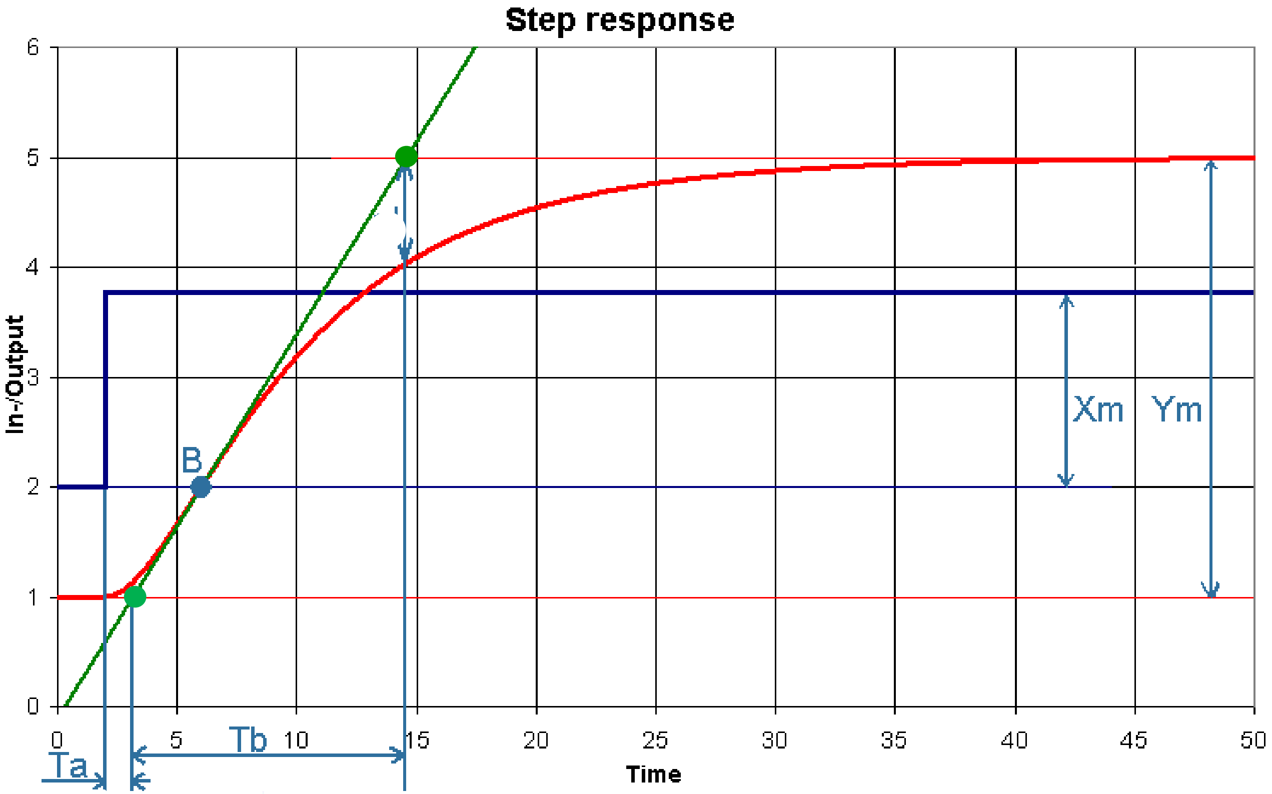
2. Oscillation method of Ziegler and Nichols

- oscillate the process by a proportional feed-back
- not directly based on a process discussed in the presentation about process analysis but it can be used for every process which is at least of the second order

2. Delayed first order process

Method of approximation

1. Determine the point of inflection B
2. Draw tangent through the point of inflection
3. Determine X_m, Y_m, T_a, T_b



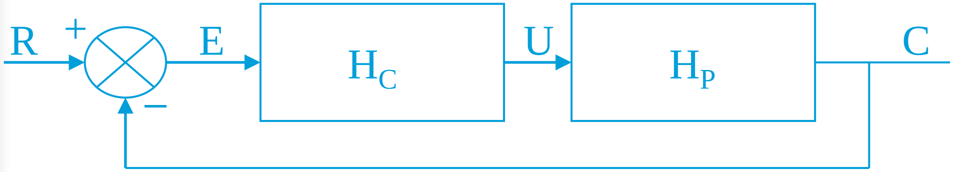
steady state gain $K_P = \dfrac{Y_m}{X_m}$

$$\tau_P = T_b$$

$$\tau_v = T_a$$

$$H_P(s) = \frac{K_P e^{-\tau_v s}}{\tau_p s + 1}$$

Step response by Ziegler & Nichols



$$H_C(s) = K_c\left(1 + \tau_d s + \frac{1}{\tau_i s}\right)$$

Delayed 1st order process and parallel controller

	K_c	τ_i	τ_d
P-Controller	$\frac{T_b}{K_p T_a}$		
PI-Controller	$\frac{0.9T_b}{K_p T_a}$	$3.3T_a$	
PID-Controller	$\frac{1.2T_b}{K_p T_a}$	$2T_a$	$0.5T_a$

Step response by Ziegler & Nichols

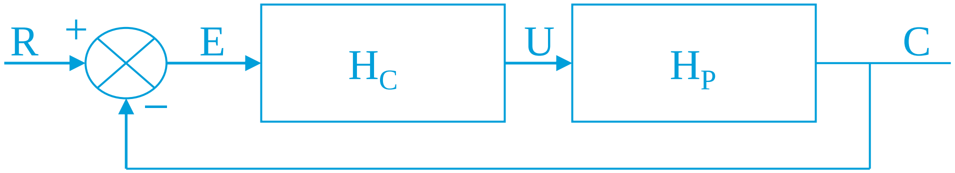
Pros:

- Simple and effective.

Cons:

- Further fine tuning needed.
- Settings are aggressive, might result in large overshoot and oscillatory behaviour.
- If the delay is dominant, then the performance is poor.
- Sensitive to parameter variation and relies on accuracy of the step response measurement, in reality often this is very noisy. Perfect measurements are often either impossible or too expensive.

Oscillation method Ziegler & Nichols

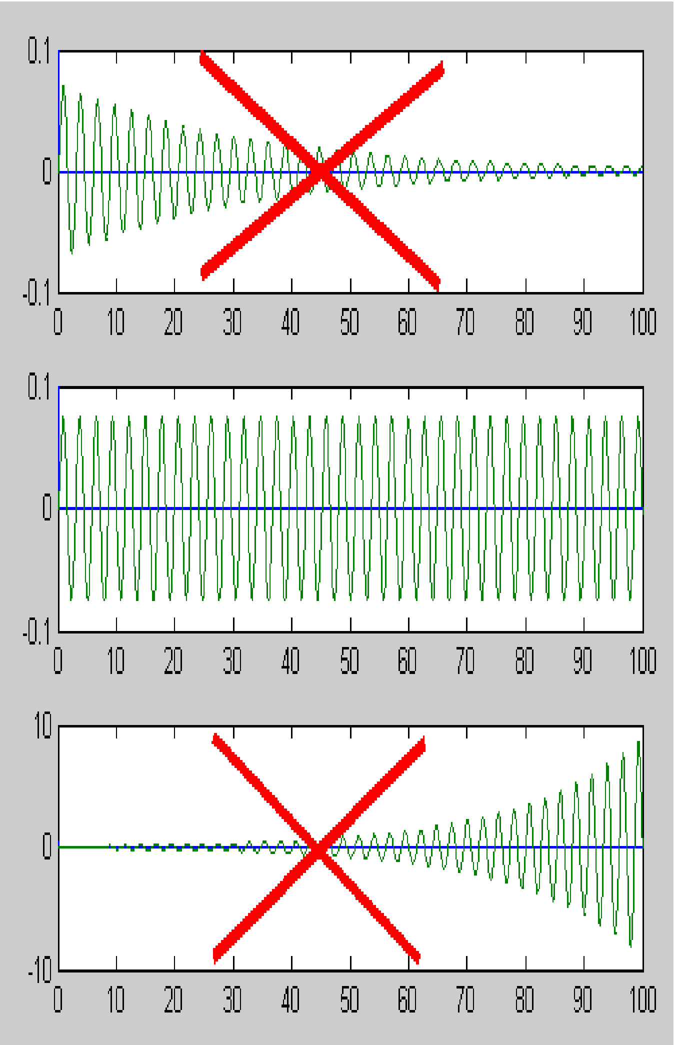


Calculate the controller in 4 steps:

1. Take only a gain K_c as controller;

   $H_C(s) = K_c$

2. Increase K_c until the process starts oscillating

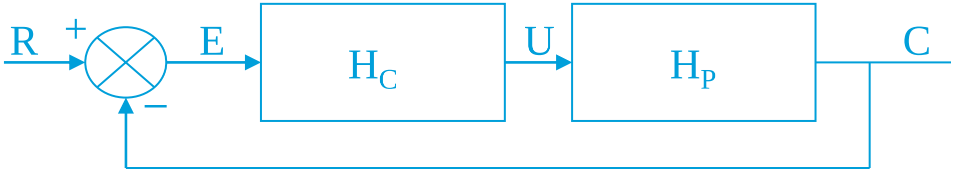


The output should not decrease (upper figure), nor increase (lower figure), but should have a constant amplitude and frequency (middle figure)

3. Read the K_c value, this is called K_b the boundary gain

4. Find the period T of the oscillation

Oscillation method Ziegler & Nichols



$$H_C(s) = K_c\left(1 + \tau_d s + \frac{1}{\tau_i s}\right)$$

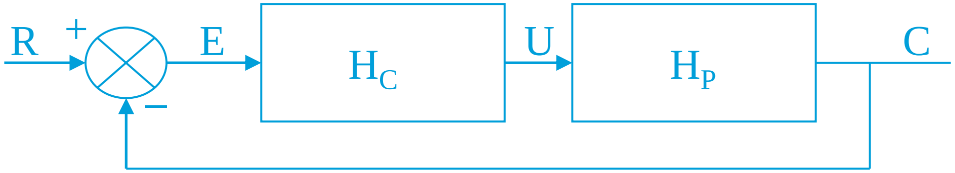
K_b is the boundary gain

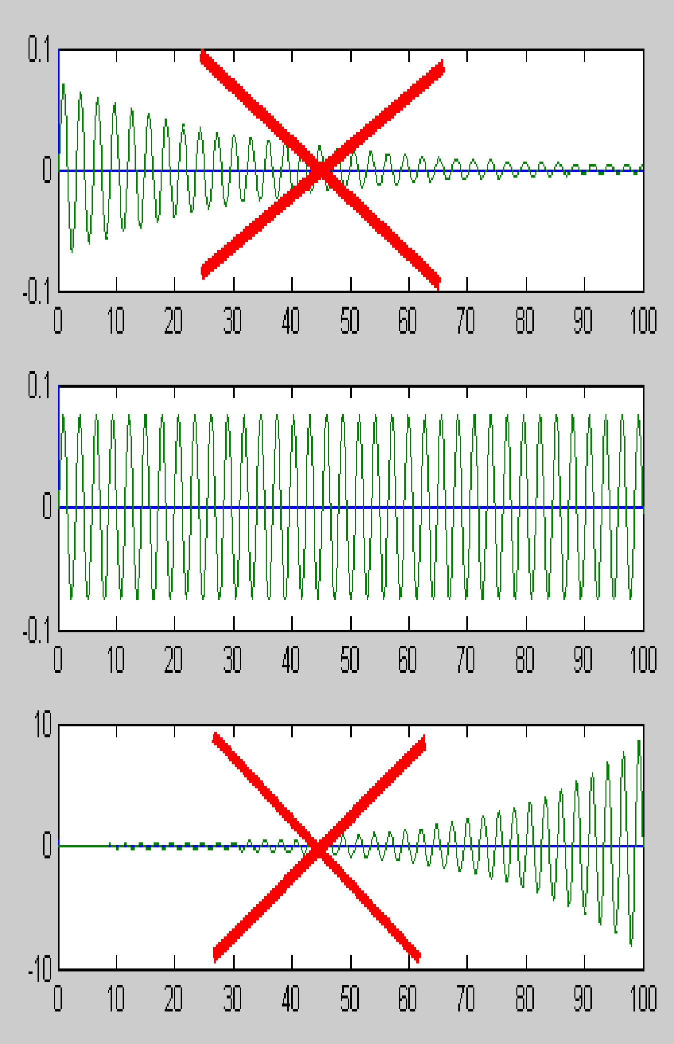
T is the oscillation periodical time at K_b

Second order process and parallel controller

	K_c	τ_i	τ_d
P-Controller	$0.5K_b$		
PI-Controller	$0.45K_b$	$\frac{T}{1.2}$	
PID-Controller	$0.6K_b$	$0.5T$	$0.125T$

Oscillation method Ziegler & Nichols





Pros:

Simple, convenient, effective, systematic!

Cons:

The system is driven towards instability, this is often dangerous and costly in practice. The resulting closed loop behavior can be very different depending on the actual process dynamics.

Other practical tuning rules for a PID-controller

- Åström and Hägglund
  - Based on Nyquist curve (We will explore what Nyquist curve is next week.)
- Setpoint weighting
- Direct pole placement
- Dominant pole design
- Optimization based method
  - LQR
- …

SUMMARY

- The functionality, pros and cons of P, I, and D controllers
- Parallel structure of PID controller
- Ziegler-Nichols tuning method of PID controllers

HOMEWORK

Stage ONE exercises:

- Problem 2
- Problem 3