Aalto University
School of Engineering

Mechatronics Machine Design (MMD)

MEC-E5001
Lecture 4
On Jan 28, 2020
Kari Tammi, Associate Professor

Learning goals, this lecture, this week

Mechatronic control, hardware, theory

Controller design: PID, basics of advanced controls, and soft computing

Laboratory exercise:

IoT sensor configuration exercise

Learning goals, exercises this week

Cascaded control loops

What is PID controller and what the terms mean

Integrator anti-windup

The feedforward: What? Why?

Example: Linear motor model

Loop specs. (Internal Model Control)

Note: control engineering involves deeply in stability and optimality. We mainly omit those questions

General on controls and hardware

Aalto University
School of Engineering

a) Open-loop and b) closed-loop control

Q: Why to use closed-loop control?

A: To improve accuracy

A: Less prone to modelling errors

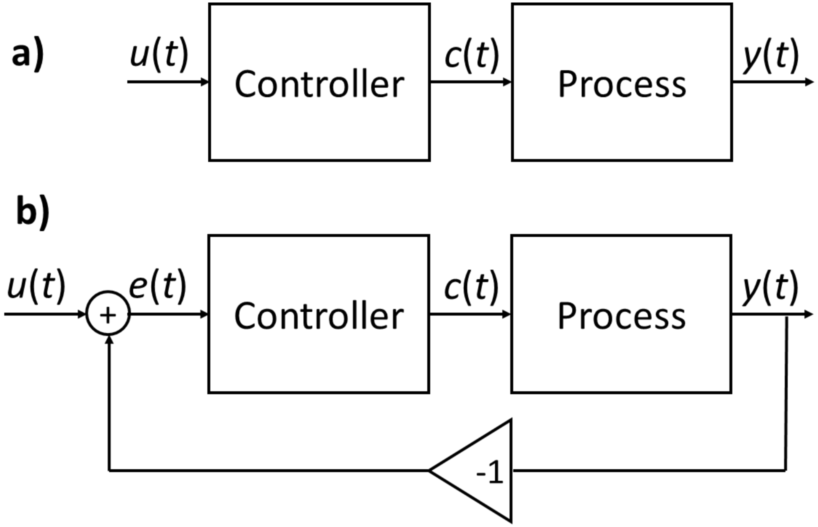
A: any other?

Q: Any challenges with closed loop?

A: Requires sensors, more cables

A: Stability may be issue

A: A bit slower



Stability can be studied by examining closed loop system poles (roots of characteristic polynomial 1+C*P=0)

Control hardware

Traditionally analogue (continuous time)

Today almost exclusively digital (discrete time)

- Continuous-time analysis and theory mostly holds

Fast analogue loops exists in digital controls

- For instance fast current control loops (wiki)

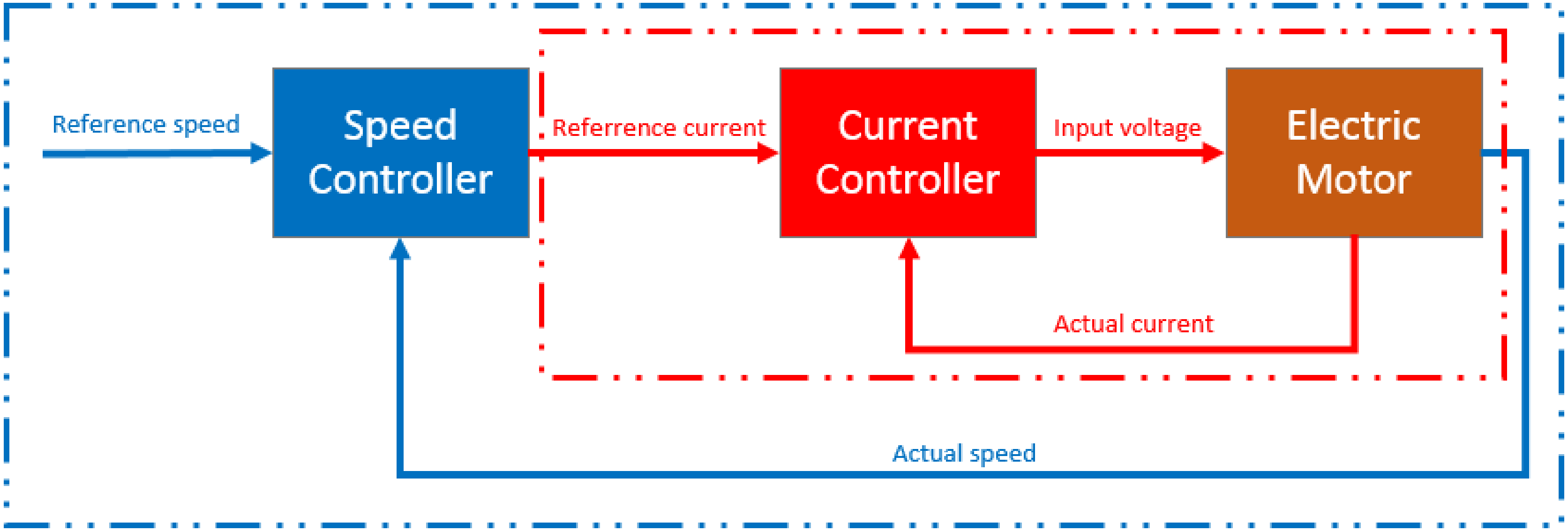
Control are more and more connected over bus (e.g. CAN, Ethernet, …) communication

Often cascaded (multiple levels, fast – slow)

Give examples of multi-level controls

Cascaded control - example

Cascaded Speed control of an electric motor



Very fast electric control loop: Eliminates small errors that happen fast

Slow mechanical control loop: Eliminates large errors that happen slowly

Digital control hardware, examples

PLC (wiki)

PC (desktop, industrial, mini, …)

Microcontroller (µC) (wiki)

DSP (wiki)

FPGA (wiki)

When a controller is good?

Technical measures: stability, robustness, rise time, time constant, overshoot, settling time, steady-state error

Techno-economical measures: cost, easy to tune, possibly automatic tuning, computationally inexpensive, updatable, possibly adaptive

PID control

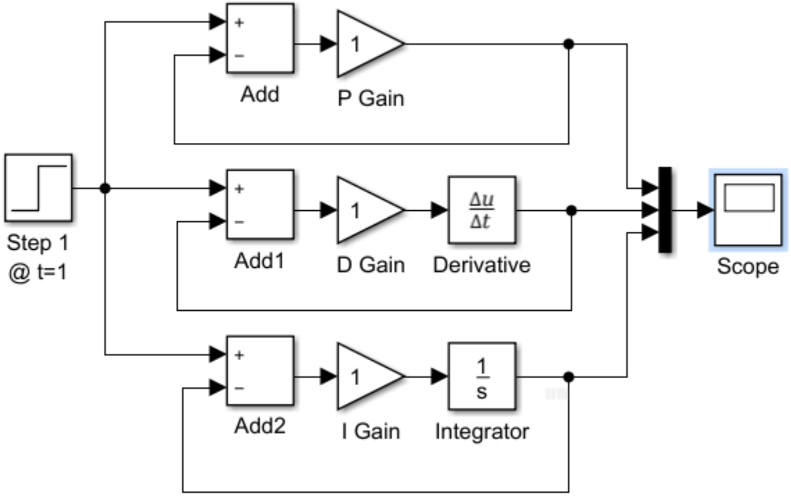
PID controller (wiki)

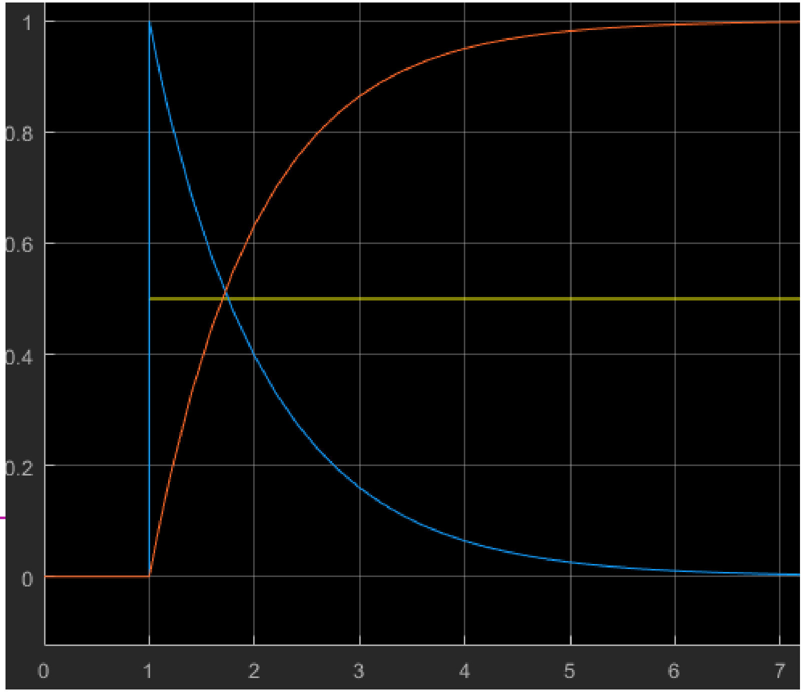
Reacts to error signal

- P: proportional term
- I: integral (steady-state error)
- D: derivative (change) term

Notes:

- Individual responses of P, I, and D terms
- Plant = 1 currently, usually a frequency dependent transfer function
- Ready made PID blocks in Matlab are often preferred (derivative filtering, anti-windup etc. included)





Which output is P, I, and D component?

PID controller – multiple ways for tuning

Trial and error

Experience (~some P term, then I term, possibly a bit D)

Physics based (e.g. D can be necessary for stability)

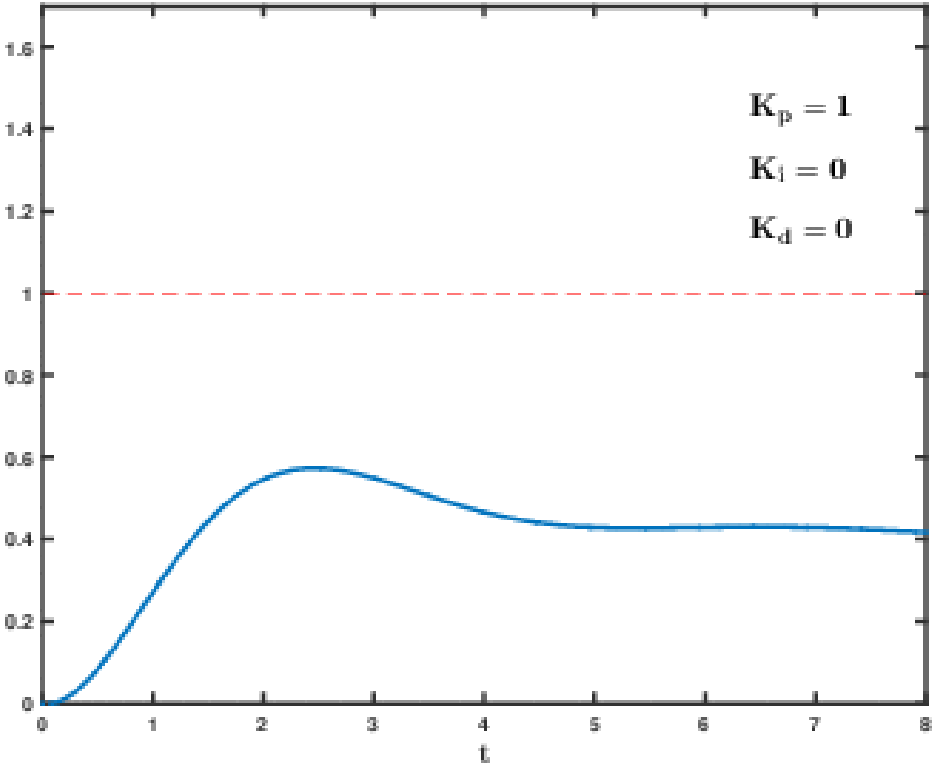
Closed-loop pole placement (roots of char. polynomial)

Optimisation methods (classical or soft computing)

Ziegler Nichols (wiki, may be difficult to realise in many mechatronic systems)

+ many other methods

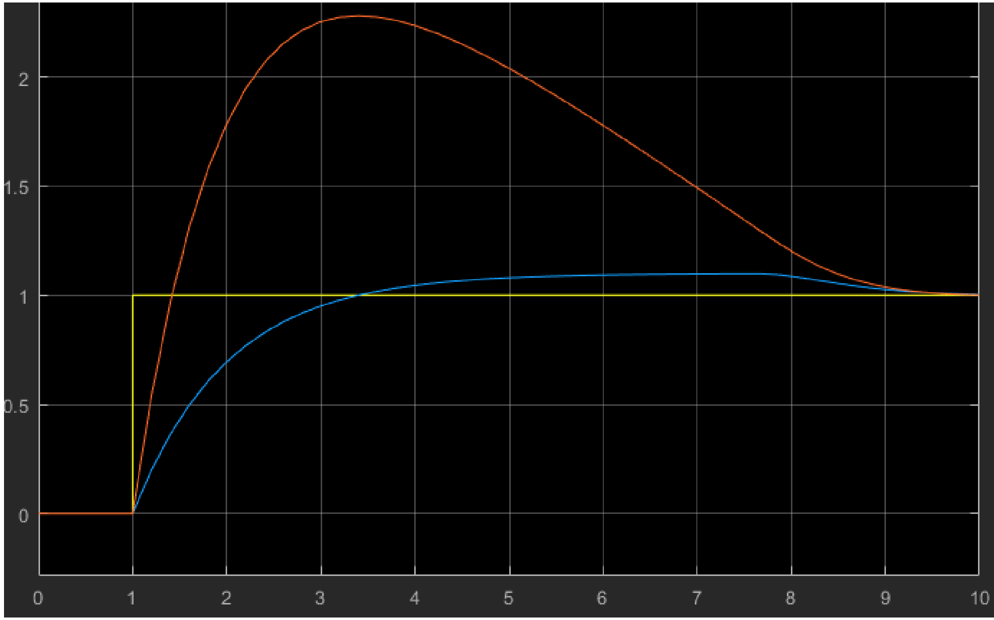
PID tuning demo (wiki)

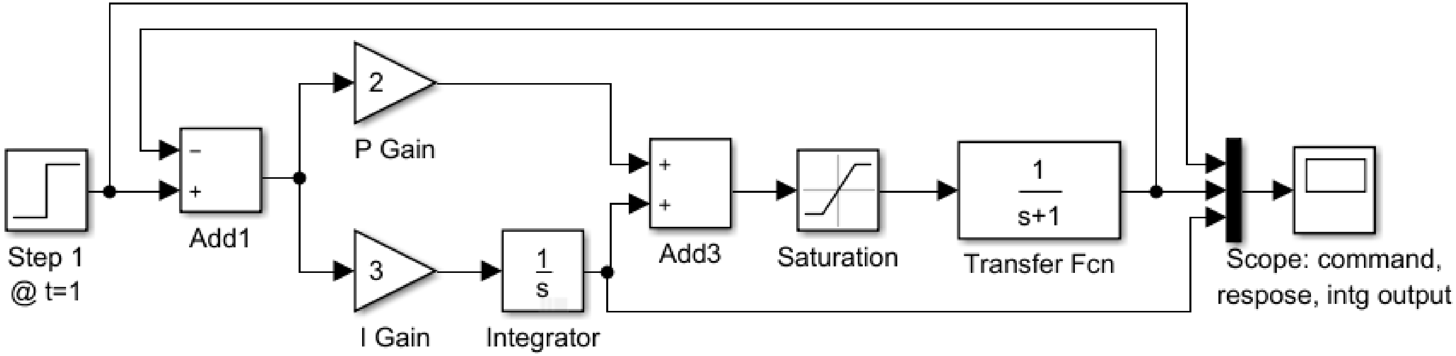


Integrator windup

If the control value is not reached, integrator may cause overshoot and eventually instability

Windup can happen e.g. due to saturation, weak actuator, broken sensor, plant jammed,…

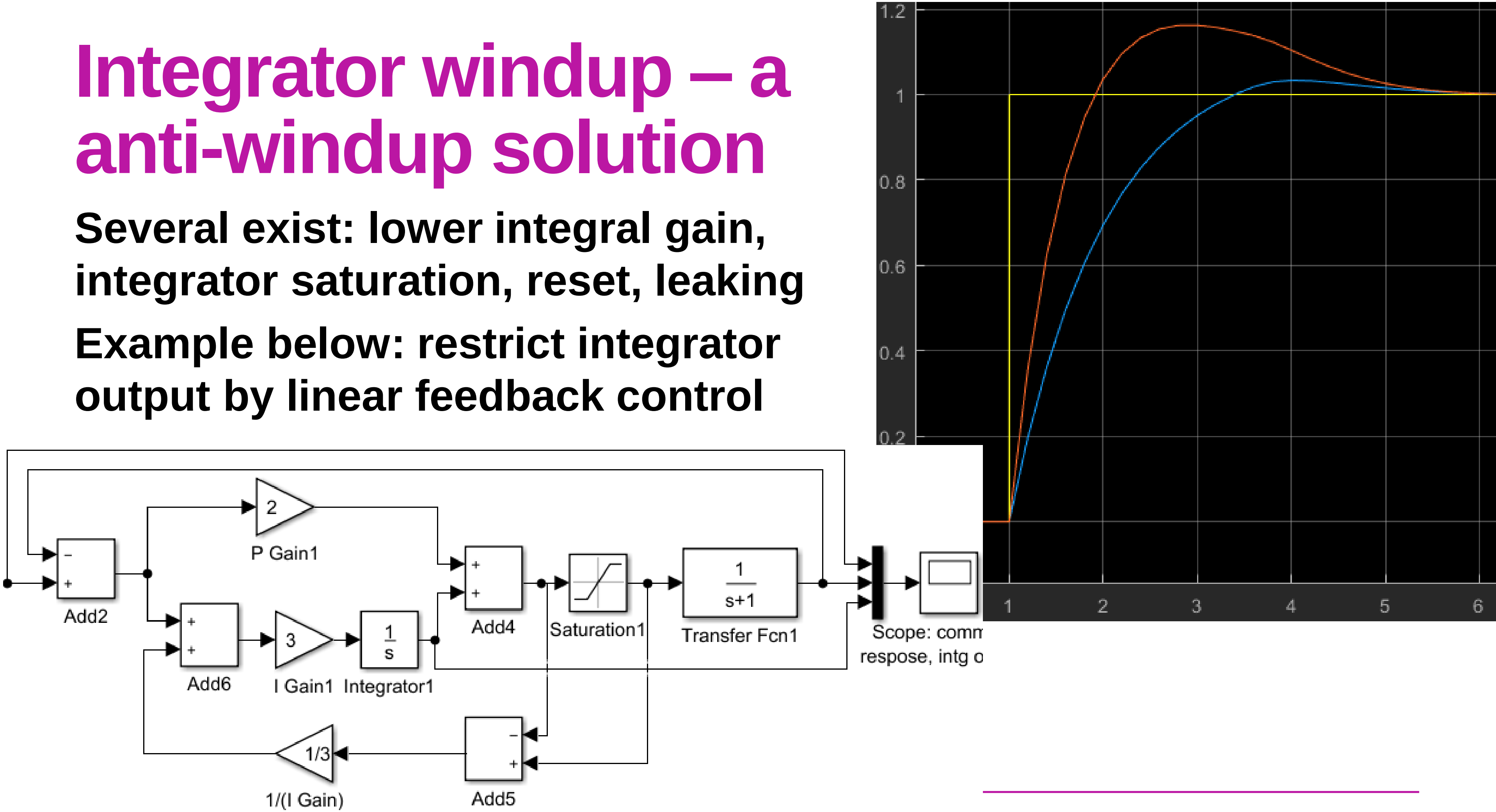




Integrator windup – a anti-windup solution

Several exist: lower integral gain, integrator saturation, reset, leaking

Example below: restrict integrator output by linear feedback control



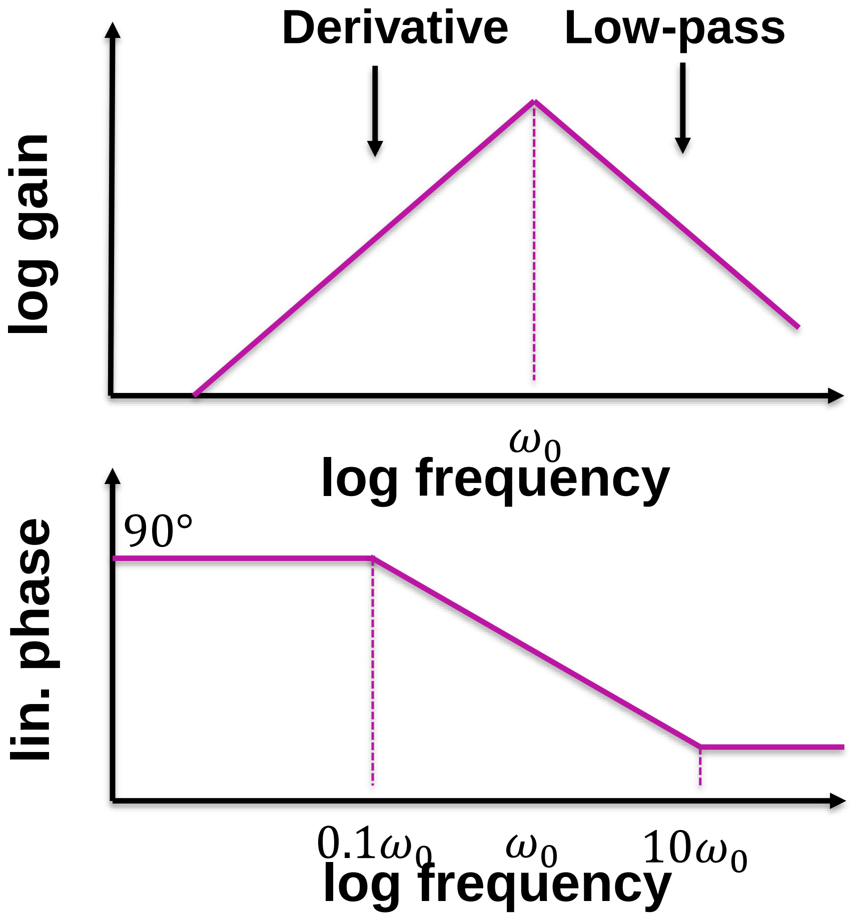
Derivative noise

Derivative control increases high-frequency control actions

Noise present at high frequencies is also increased

In practice, low-pass filter applied with derivative controller

For instance, ready made Simulink blocks have filtering feature



Controller synthesis (applicable on PID, but also many other controllers)

Aalto University
School of Engineering

Control synthesis methods

Classical

- **LQG (wiki)**
- **Loop shaping (H infinity) (wiki)**
- **Robust control (wiki)**

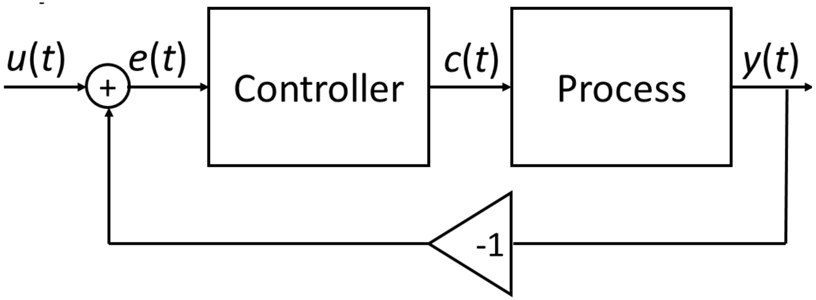
Soft computing (and/or nature inspire)

- **Neural networks (wiki)**
- **Fuzzy control (wiki)**

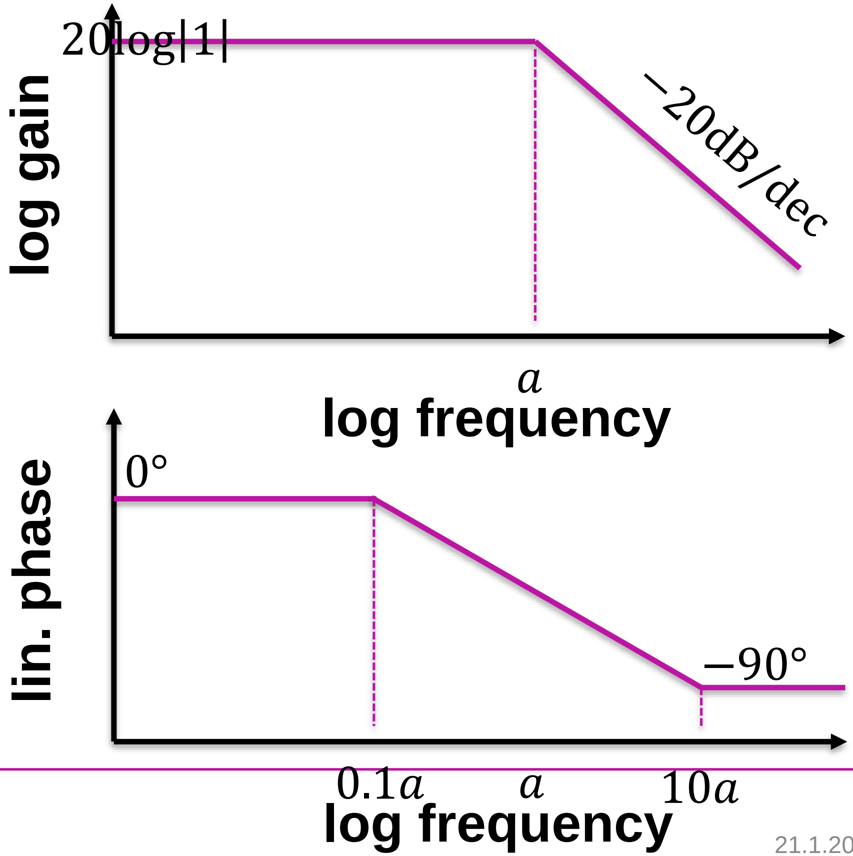
+ many other

Synthesis example – closed loop properties specified (1/2)

Derive controller to fulfil closed loop specification



$$C(s) = ?, P(s) = \frac{1}{s}$$



Synthesis example – closed loop properties specified (1/2)

$$C(s) = ?, P(s) = \frac{1}{s}$$

$$\frac{Y(s)}{U(s)} = \frac{a}{1+a}$$

$$\frac{Y(s)}{U(s)} = \frac{C(s)P(s)}{1+C(s)P(s)} = \frac{a}{s+a}$$

$$\frac{Y(s)}{U(s)} = \frac{C(s)1/s}{1+C(s)1/s} = \frac{a}{s+a}$$

$$\frac{Y(s)}{U(s)} = \frac{C(s)}{s+C(s)} = \frac{a}{s+a}$$

$$C(s) = a$$

Result is proportional controller (gain only)

Note: the spec may not be realisable (2 equations vs. one controller)

Aalto University
School of Engineering

https://en.wikipedia.org/wiki/Internal_model_(motor_control)

a) Feedback and b) feedback & feedforward control topologies

Q: Why to use feedforward control?

A: To react faster

A: To have dedicated tuning for regulation and tracking problems

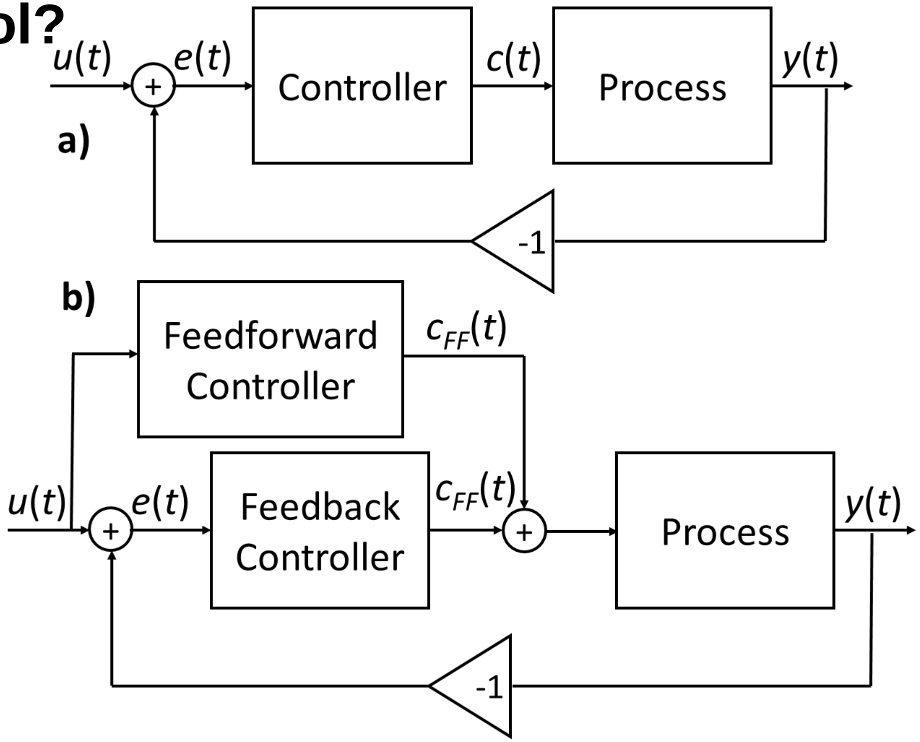
A: Any other?

Q: Challenges with feedforward?

A: More complicated

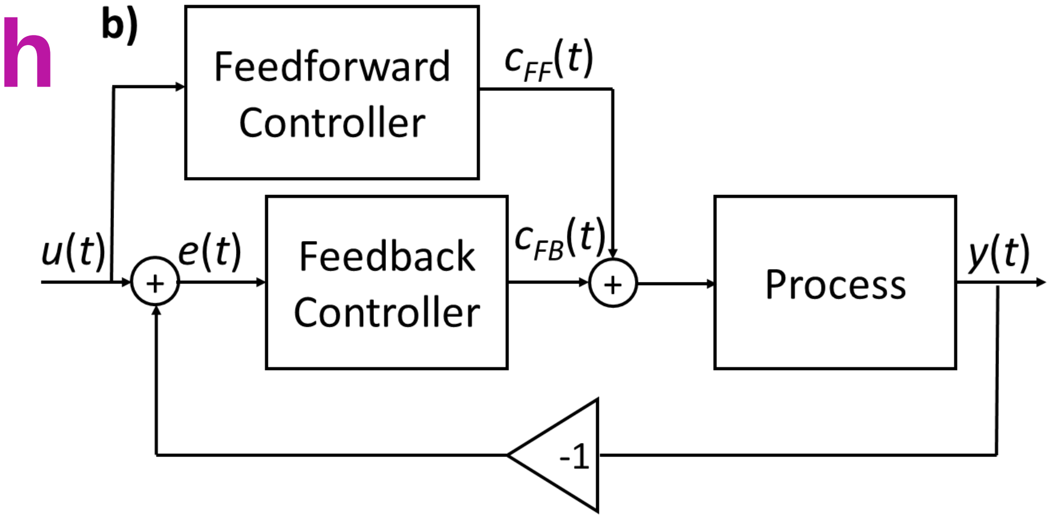
A: More tuning, more work

A: Stability?



Many topologies for feedforward, and control systems in general exist. a) and b) are commonly used examples

Feedback and feedforward control – a possible approach



$C_{FF}(s)$, $C_{FB}(s), P(s)$

$$\frac{Y(s)}{U(s)} = \frac{C_{FF}(s)P(s) + C_{FB}(s)P(s)}{1 + C_{FB}(s)P(s)}$$

$$C_{FF}(s) \approx \left(P(s)\right)^{-1}$$

$$\frac{Y(s)}{U(s)} \approx \frac{1 + C_{FB}(s)P(s)}{1 + C_{FB}(s)P(s)} \approx 1$$

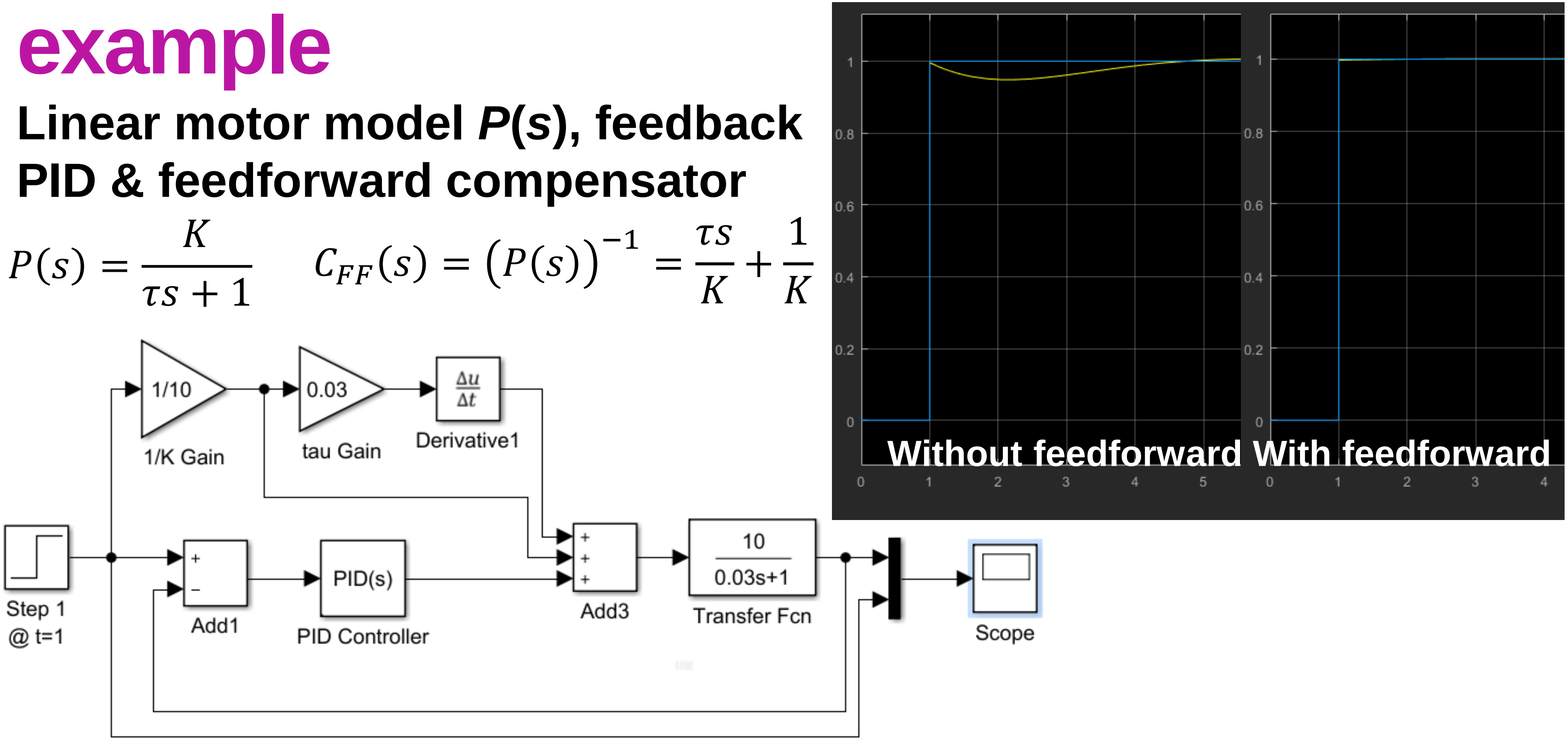
If we can find approximation of process (or plant) inverse for feedforward path it can make tracking very fast & accurate

Finding inverse is problematic in general case, but works with simple examples

Feedback and feedforward control – example

Linear motor model *P*(*s*), feedback PID & feedforward compensator

$$P(s) = \frac{K}{\tau s + 1} \qquad C_{FF}(s) = \left(P(s)\right)^{-1} = \frac{\tau s}{K} + \frac{1}{K}$$



Without feedforward With feedforward

http://www.controleng.com/single-article/feed-forwards-augment-pid-control/e6dcd10be91de5c64e075ac84749e77d.html

Group work (and lecture quiz)

Aalto University
School of Engineering

Group work & lecture quiz 4

Discuss with your pair. Write down your answers and use them to answer lecture quiz today.

1. Explain why closed-loop control reduces control errors. Prove it by deriving the closed-loop transfer function and analysing the relation between the output *y* and reference *u* (1 point).

$$\frac{y}{u} =$$

2. Design a proportional speed controller for a wheel with inertia *J*. For a unit step in speed command, 63 % of the command value needs to be achieved in 1 second. Derive the proportional gain *K* (1 point).

$$T = J\dot{\omega}$$

3. Write a code of PID controller for a digital controller. Use programming/scripting language you know and document/explain the code (1 point).