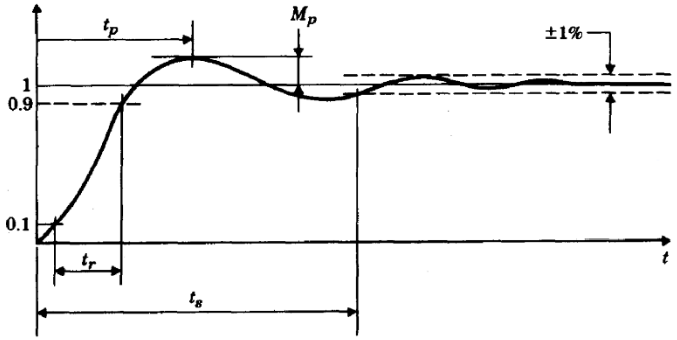
Aalto-yliopisto
Aalto-universitetet
Aalto University

Control systems

KON-C2004 Mechatronics Basics

Raine Viitala 5.11.2024

Lecture topics & learning outcomes

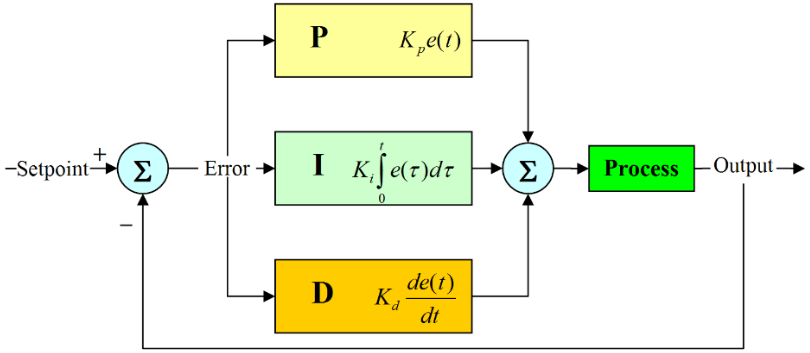


System modeling and response

- Know what system modelling is and analyse responses in simple cases

Open & closed loop control

- Understand the difference between open and closed loop control



PID controller

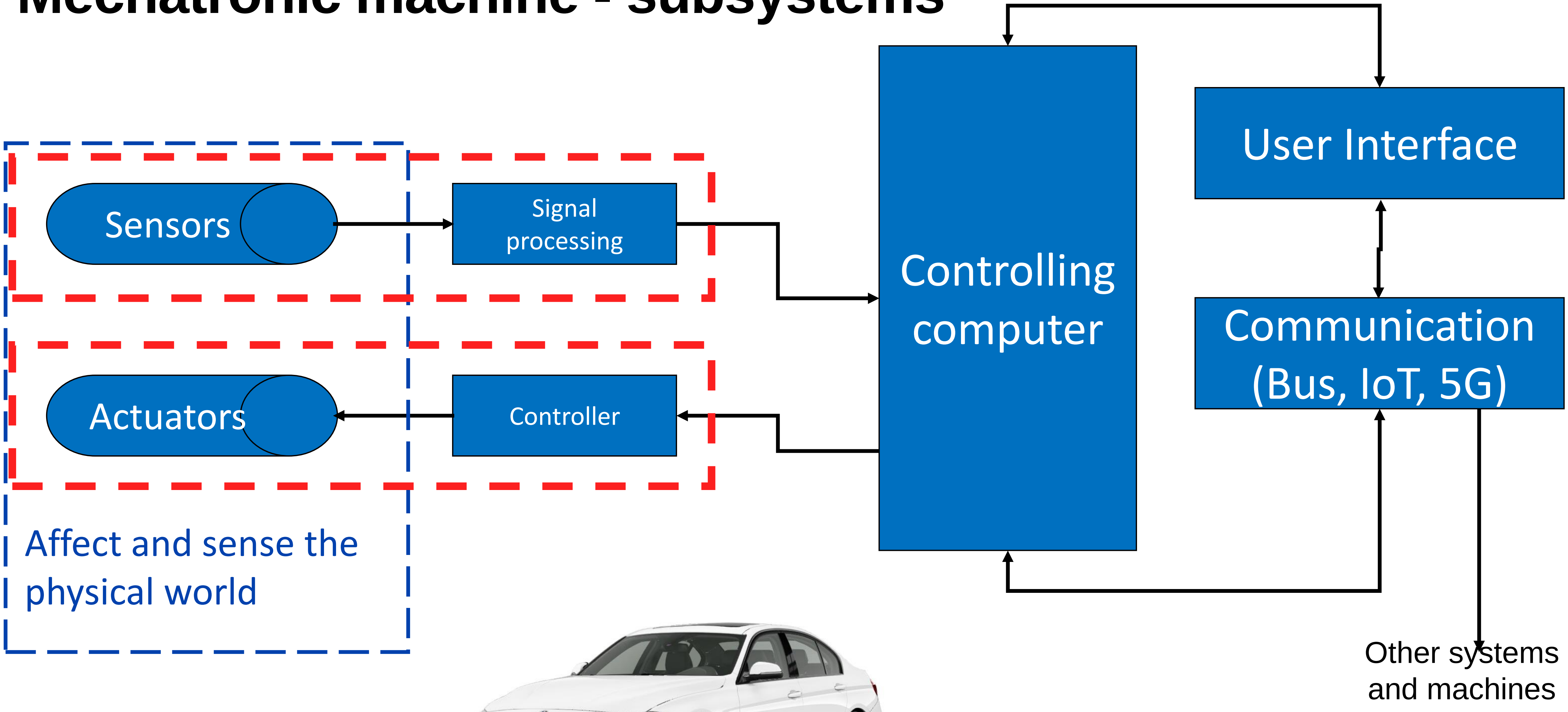
- Understand and trial and error tune PID controller

Servo motors

- Know what servo motor is

A!

Mechatronic machine - subsystems

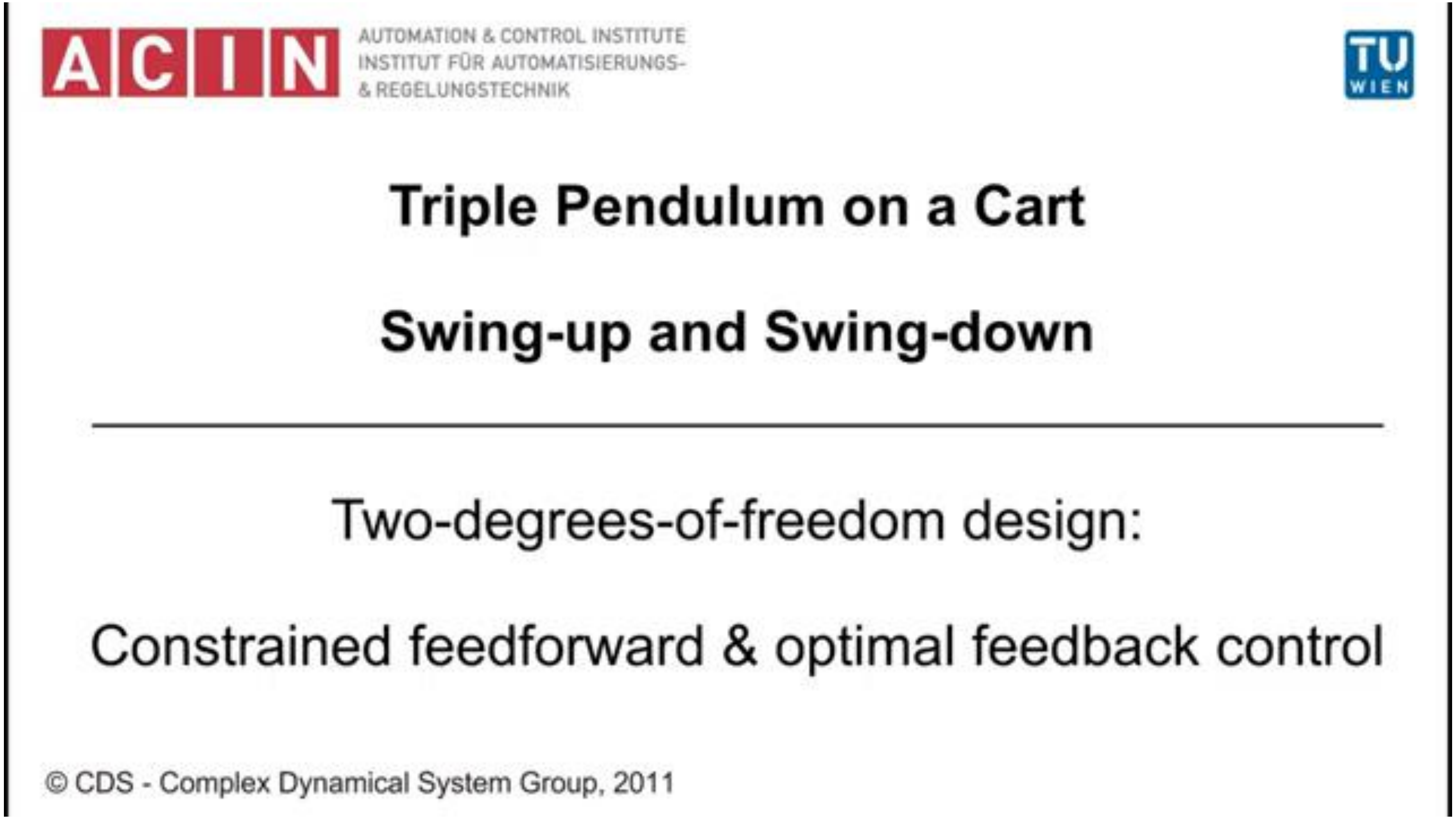


A!

Example

Triple inverted pendulum

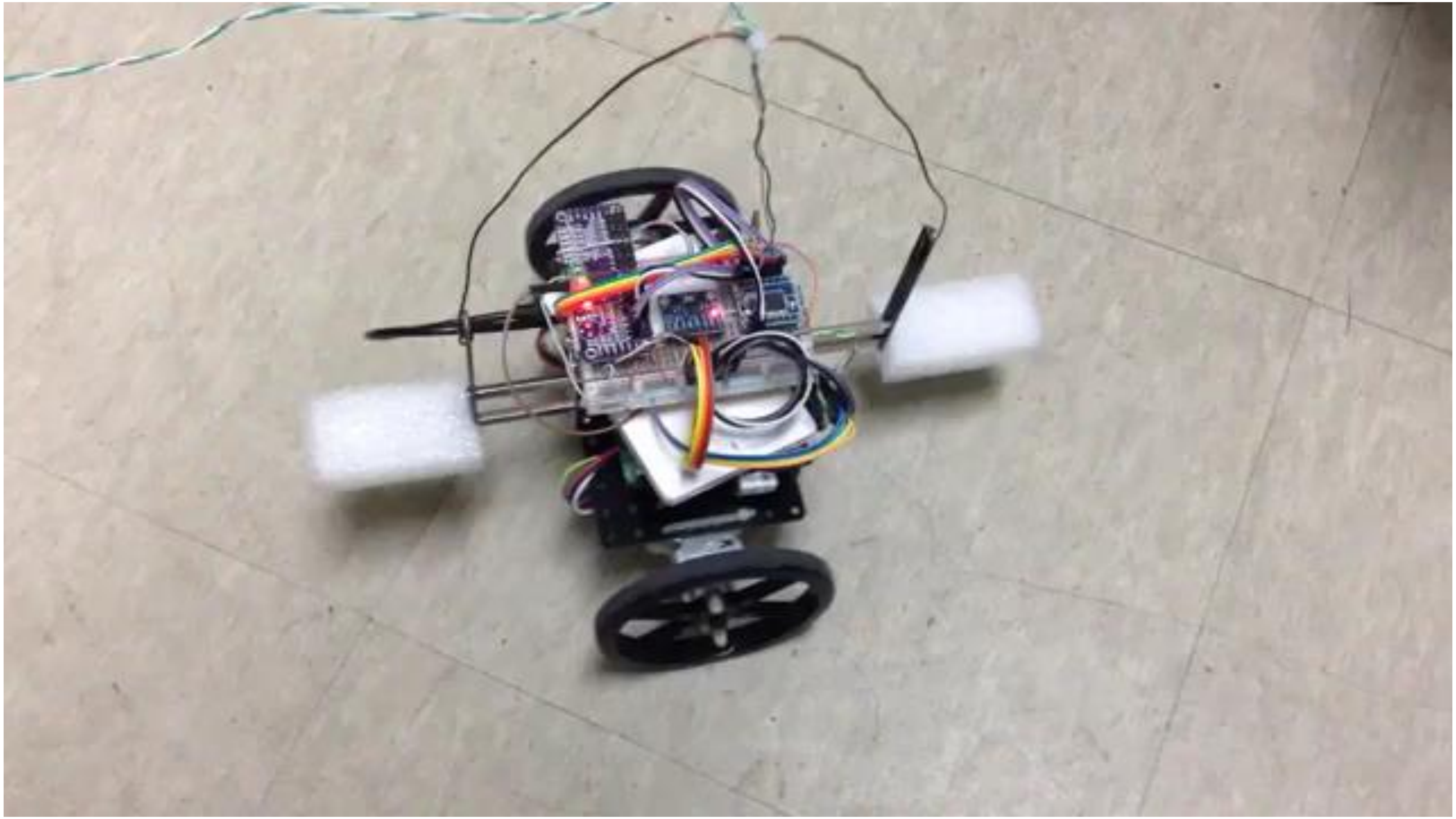
No actuators in rotational joints, only angle measurement



Unstable controller

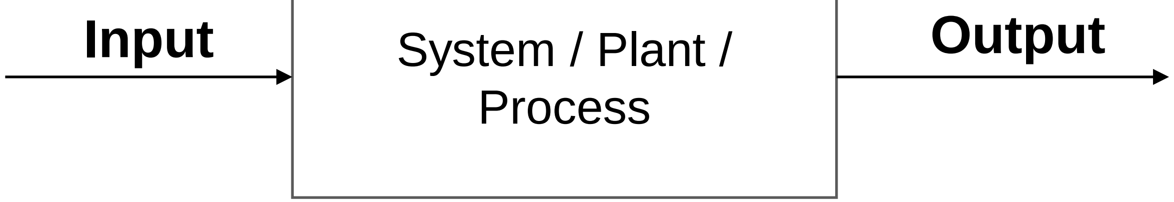
A!

Unstable controller 2



System response

System



Mathematical models of systems

Zero order

=time independent

$$Ax_{out} = Bx_{in}$$

$$U = 1.45x$$

Ideal sensor

First order

$$A_1 \frac{dx_{out}}{dt} + A_0 x_{out} = B_0 x_{in}$$

$$C\frac{dV}{dt} + \frac{V}{R} = 0$$

RC-circuit

Second order

$$A\frac{d^2\, x_{out}}{dt^2} + B\frac{dx_{out}}{dt} + x_{out} = Cx_{in}$$

$$\mathrm{m}\frac{d^2x}{dt^2} + b\frac{dx}{dt} + kx = F$$

A!

Firsto order system: DC motor in terms of angular velocity
Laplace analysis to solve differential equations in analytical form

Electrical in time domain

$$U = L_A \frac{dI}{dt} + R_A I + K_E \omega$$

Mechanical in time domain

$$T = K_T I = (J_R + J_L)\frac{d\omega}{dt} + D\omega + T_f + T_L$$

Laplace transformation to Laplace domain

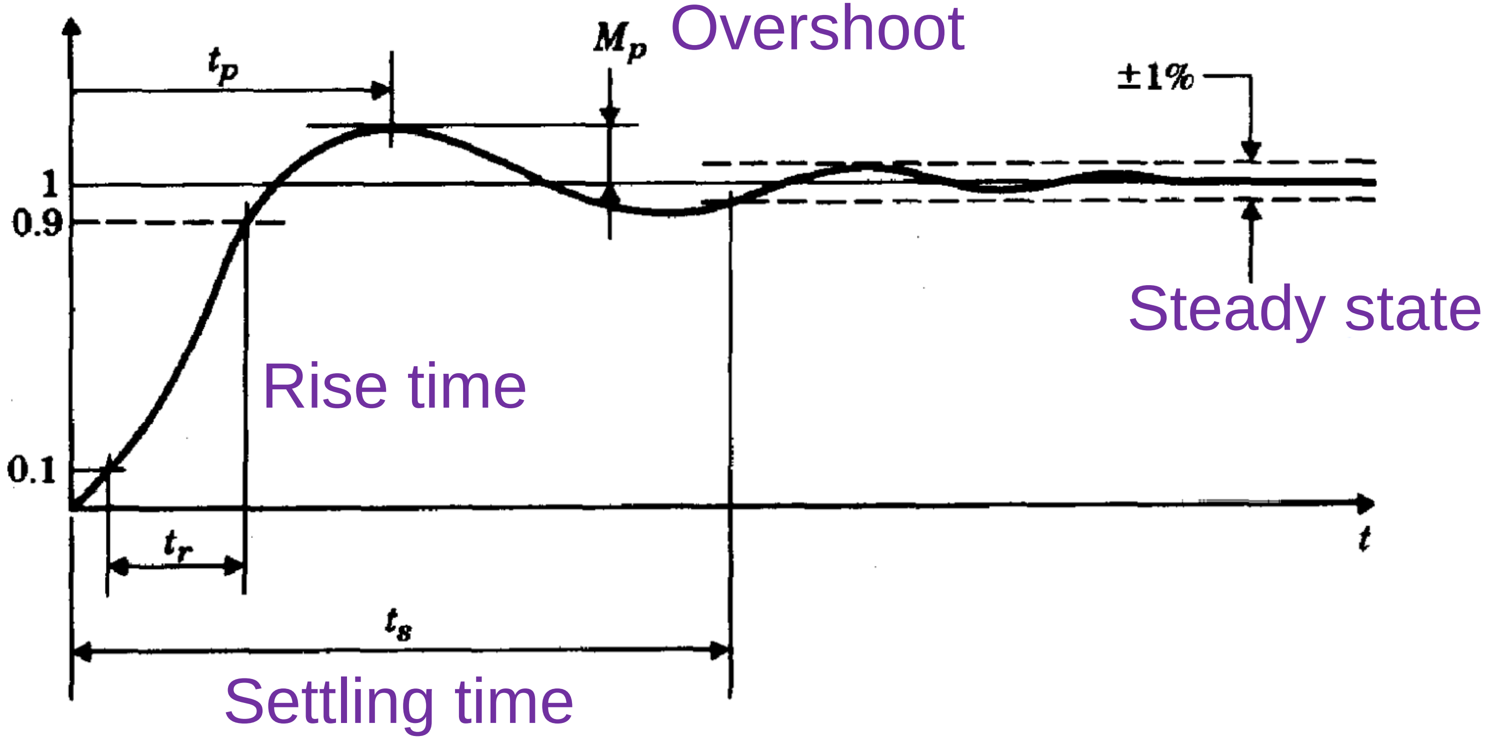
$$V(s) = LsI + RI + K_e\omega$$
$$K_t I(s) = Js\omega + \mathrm{D}\omega$$

Transfer function

$$\frac{\omega(\mathrm{s})}{V(s)} = \frac{K_t}{(Js + D)(Ls + R) + K_t K_e} = \frac{K_t}{Js^2 + (RJ + DL)s + DR + K_t K_e}$$

Step response of a system

This is what we put in



Parameters

Time constant $\boldsymbol{\tau}$ – Represents 0-63 % rise time

Damping ratio $\boldsymbol{\zeta}$

- Damping ratio for underdamped < 1, critically damped = 1, overdamped > 1
- Unstable when negative

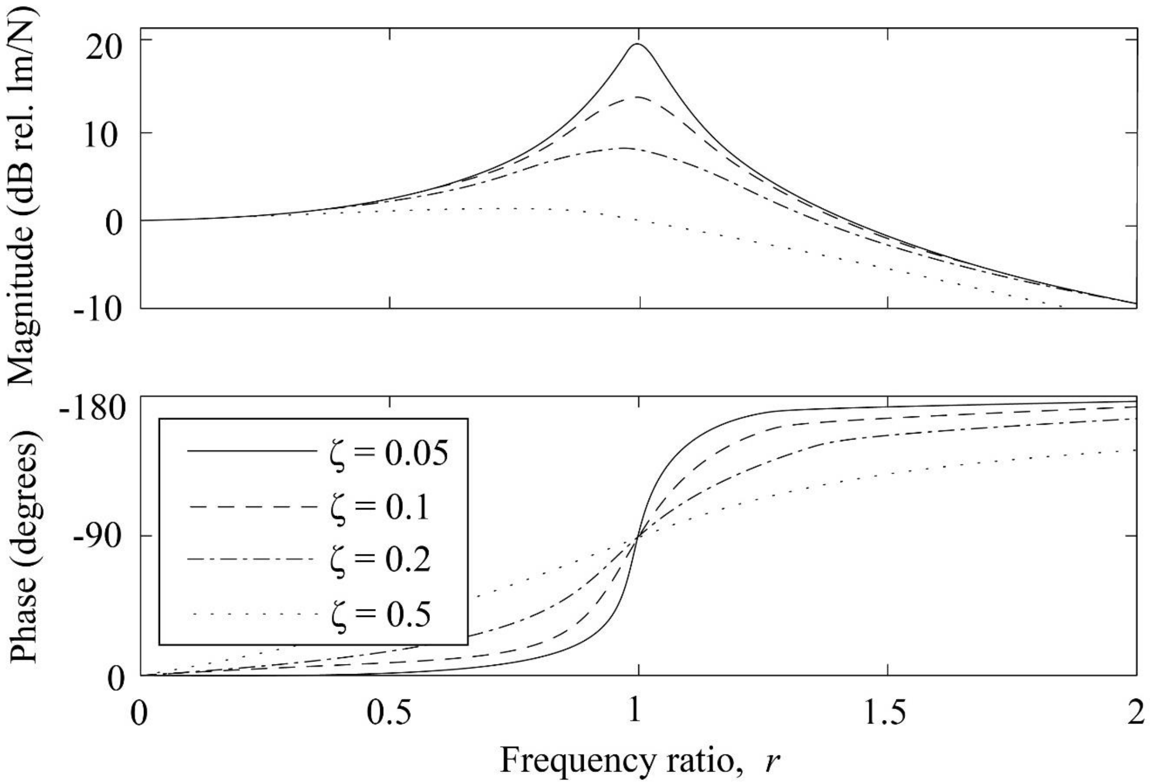
Natural frequency $\boldsymbol{\omega_0}$

- Frequency at which system oscillates without driving or damping force
- Resonance happens when external force excites the system at natural frequency
  - At natural frequency, the energy transfer to the excited system is at its maximum

Resonance

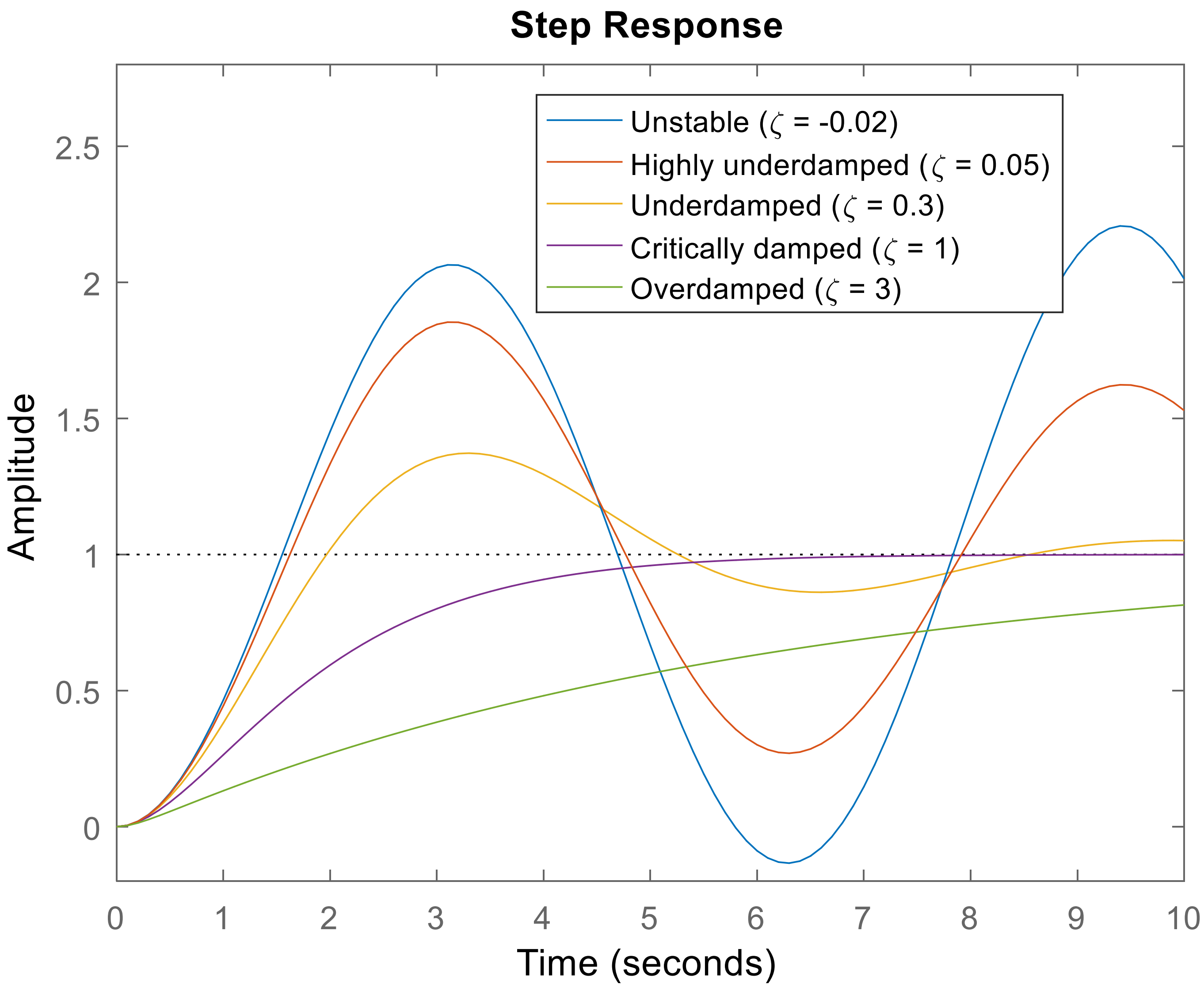
A!

Resonance 2



Magnitude (dB rel. 1m/N)
20
10
0
-10
Phase (degrees)
-180
-90
0
ζ = 0.05
ζ = 0.1
ζ = 0.2
ζ = 0.5
0
0.5
1
1.5
2
Frequency ratio, r

System damping

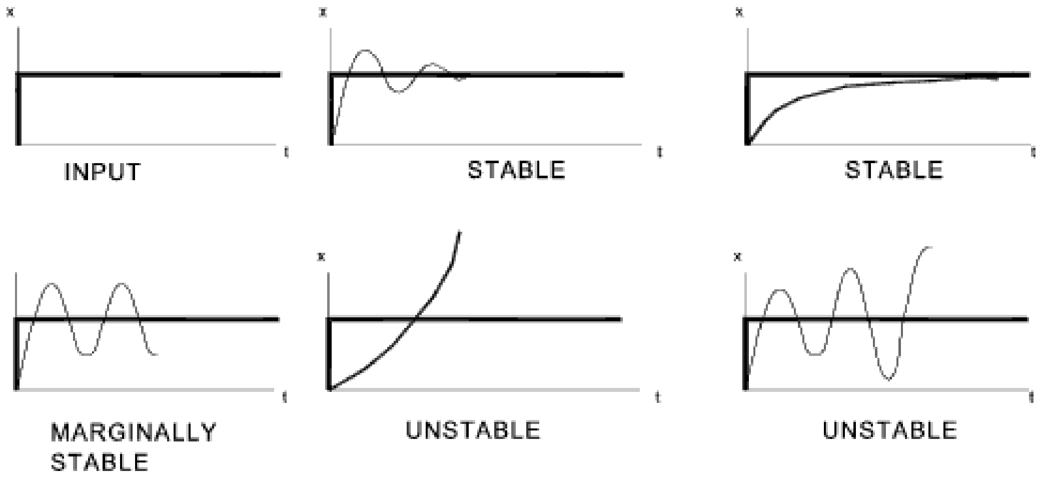


Stability

System is stable when bounded input always causes a bounded output

Problems are caused by

- Inherently unstable systems
- Unstable controller
- Delay in control loop
- Too high control gains
- Positive feedback



Control systems

Requirements for a control system

The output of the process follows the desired value with some

- Steady state accuracy
- Transient accuracy
  - Time constant, overshoot
- Stability

And the controller requires reasonable

- Computational effort
- Amount of knowledge of the process
- Effort to tune

Open loop control

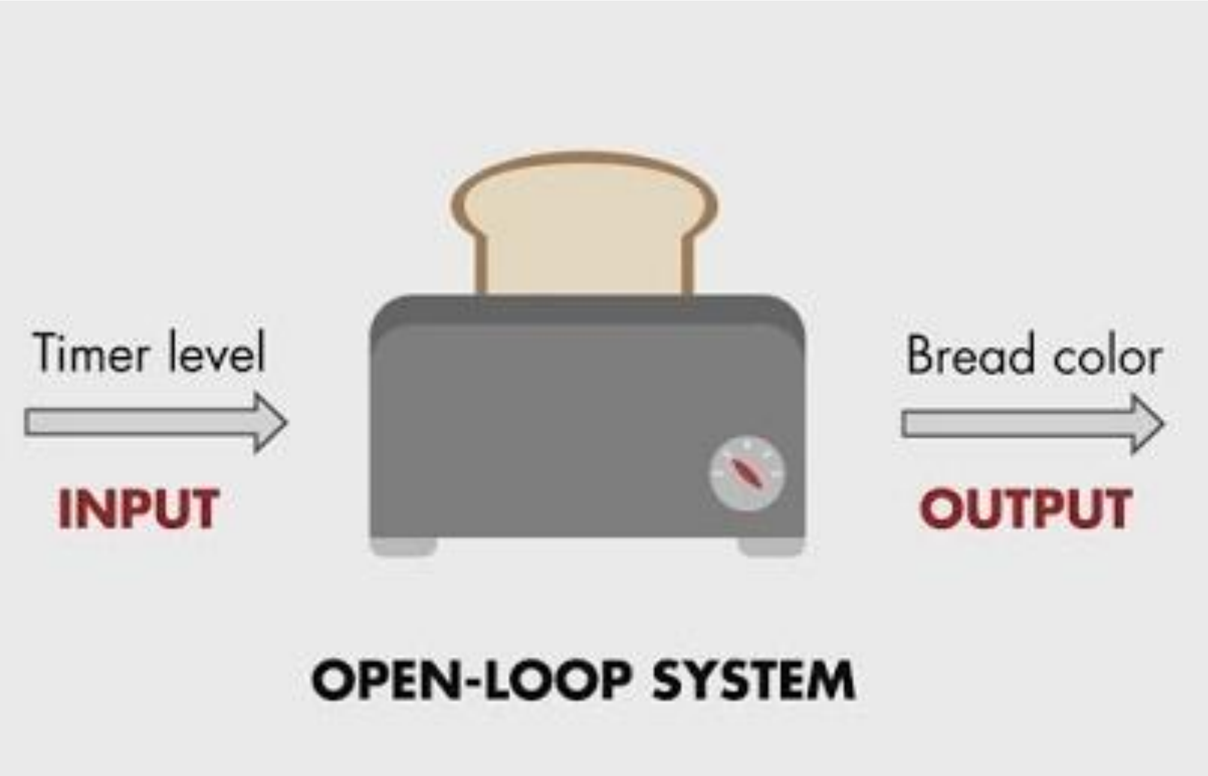
Controlling the p ... ate

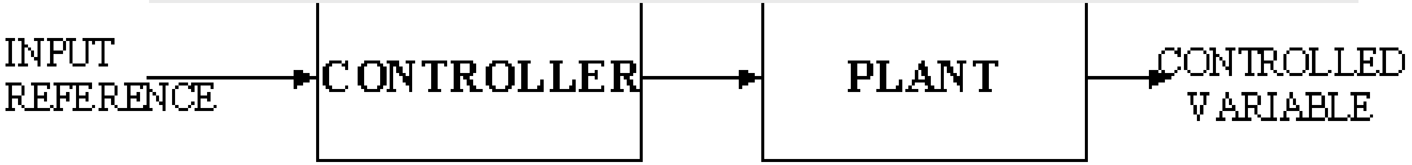
Controller alway

Does not reject

- External forces
- Aging of devices (c
- Changes in enviror

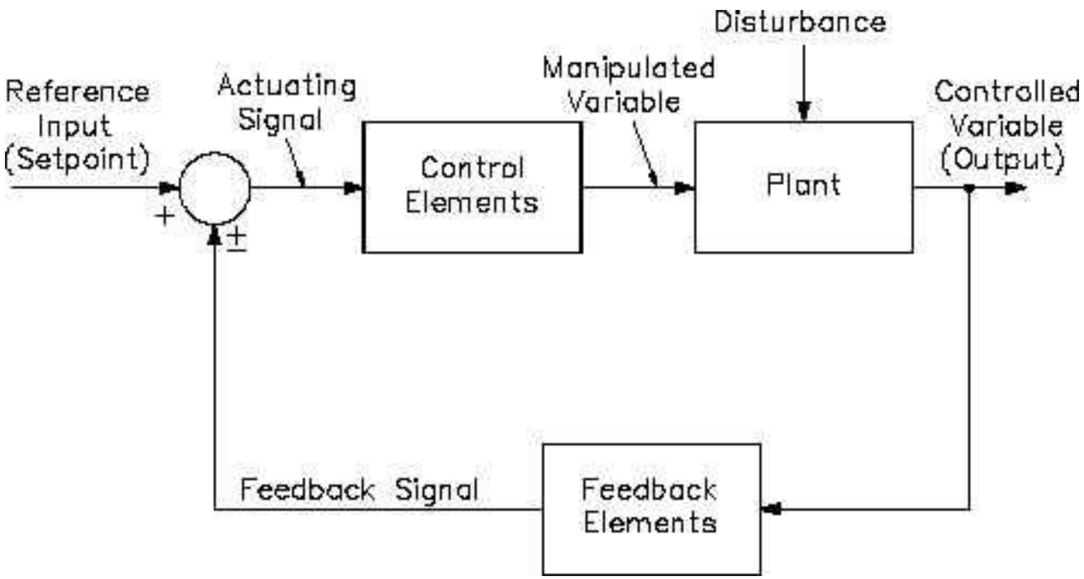
No sensors requ

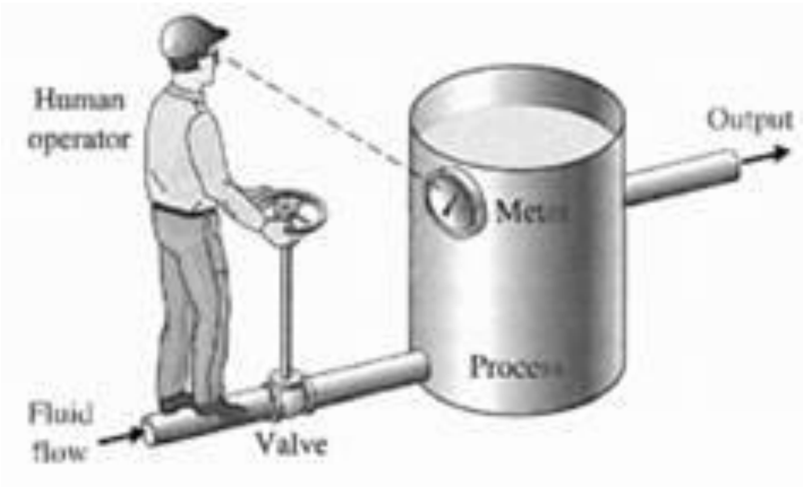




Closed loop (feedback) control

- Modifying the control signal using information of the current output of the process
- Negative feedback stabilizes system

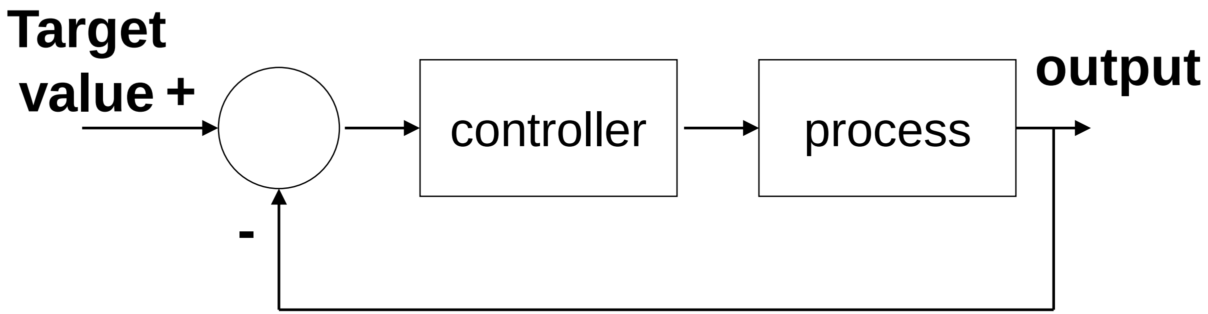


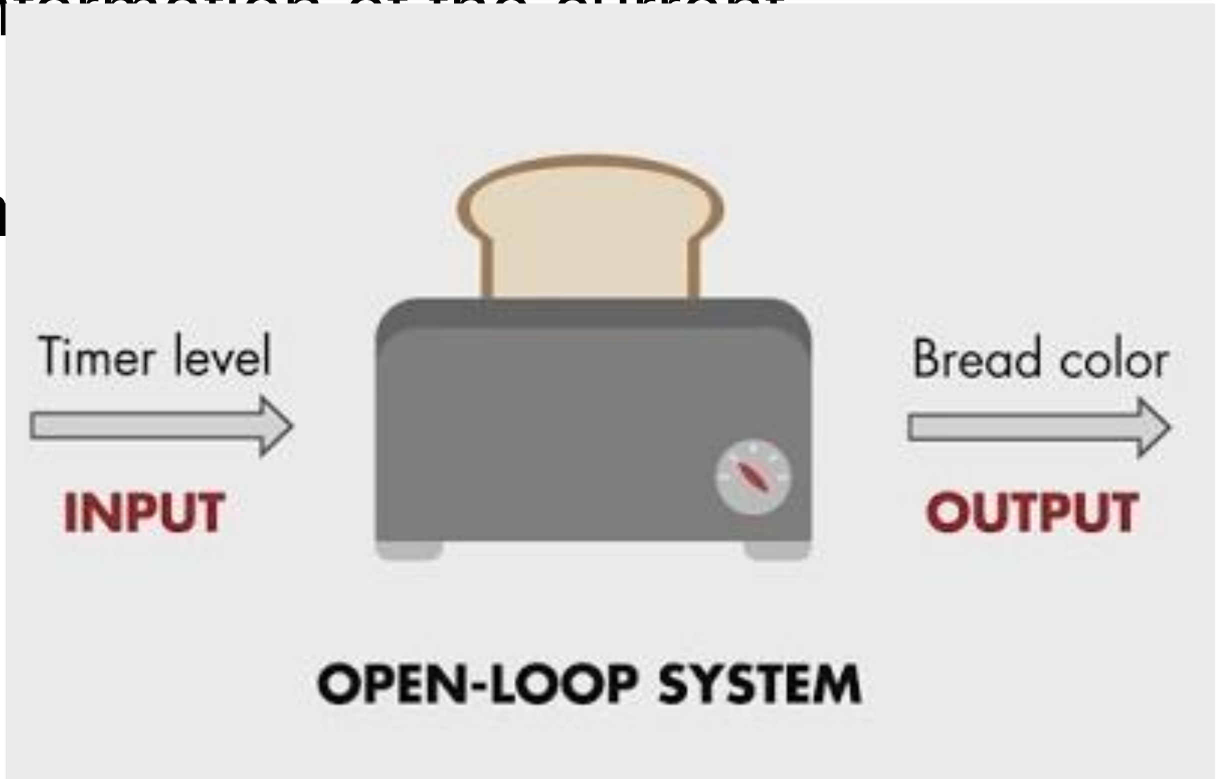


A!

Closed loop (feedback) control

- Modifying the control signal using in
output of the process
- Negative feedback stabilizes systen



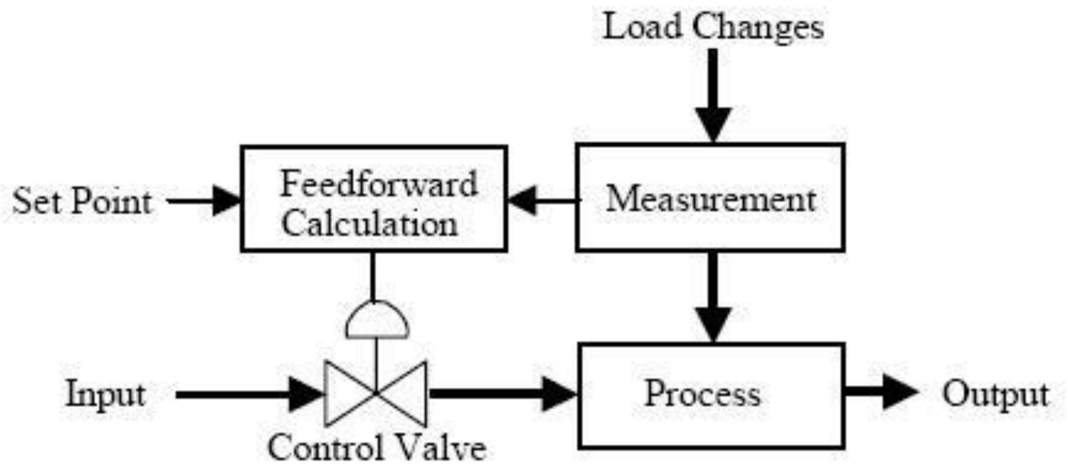


How to make this closed loop??

Feed forward control

Modifying the control signal according to some external parameters

- Power steering in a car – speed affects in addition the the position



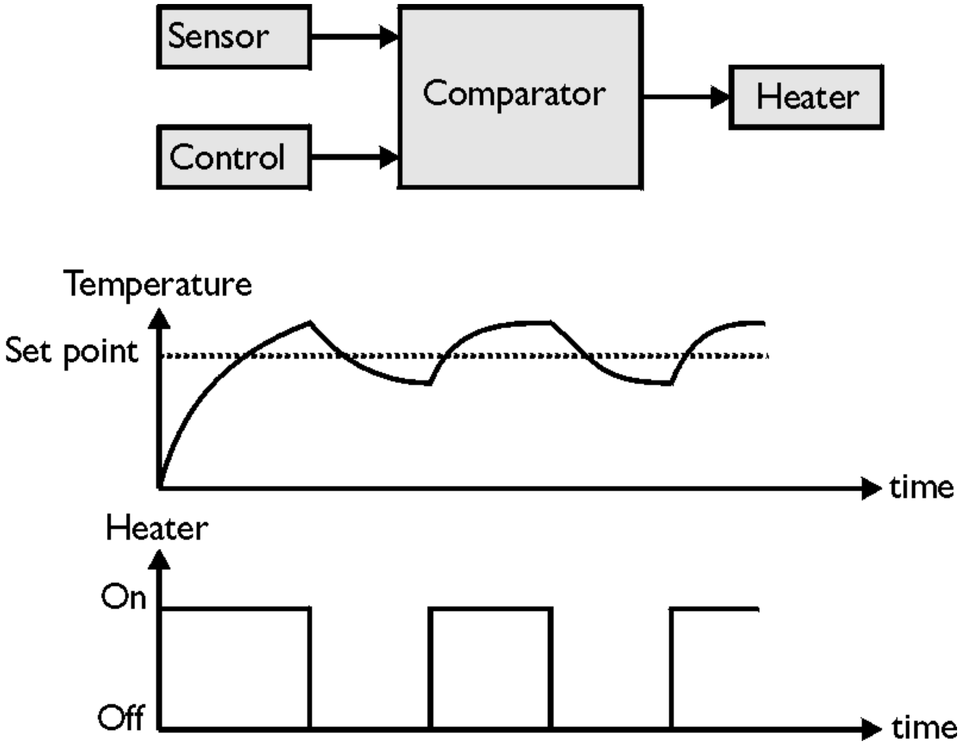
Bang bang controller

Actuator at full positive or negative power

Very simple and often fastest possible response

In some cases requires a system model for accuracy

Requires hysteresis to limit switching



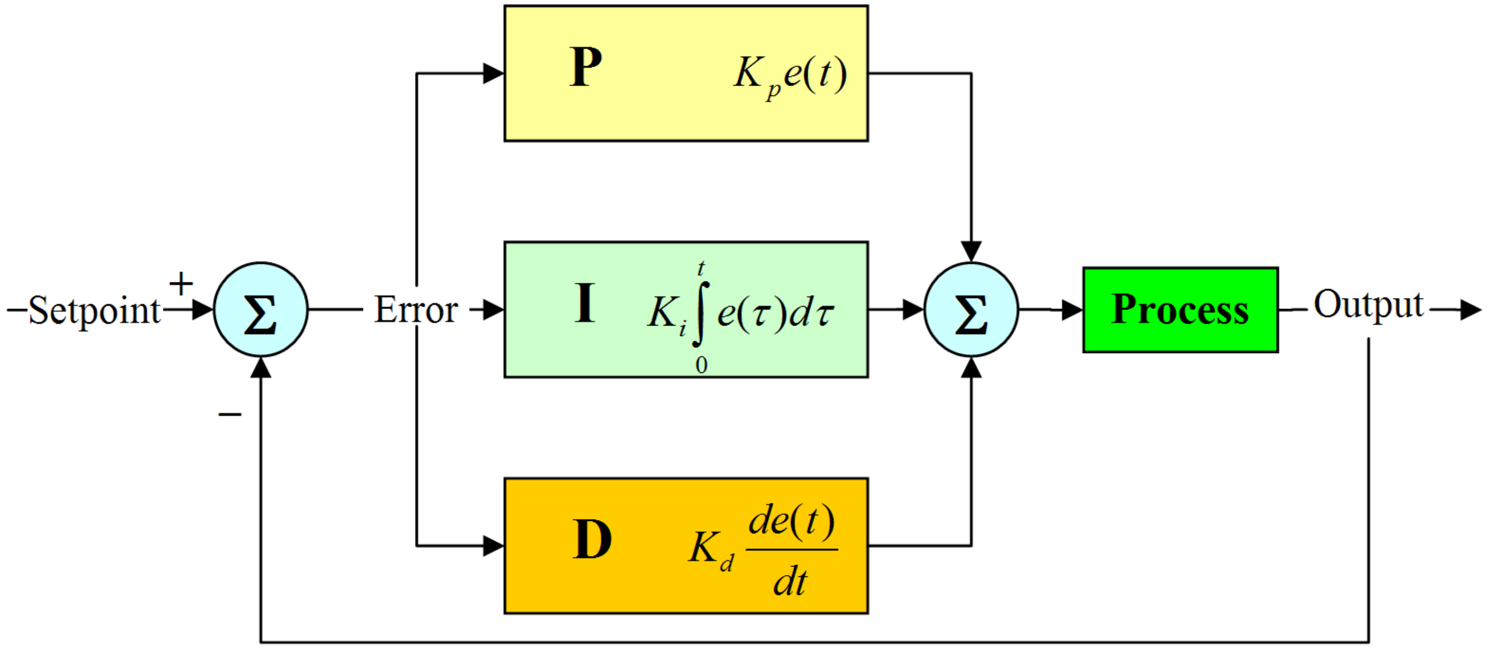
PID controller

Back to the example

A!

PID controller

Proportional, Integral and Derivative terms



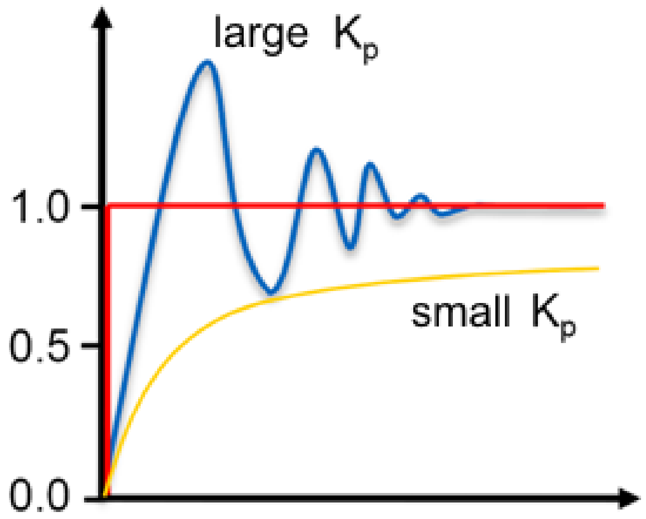
PID controller (P term)

Output proportional to error signal

Large gain K_p -> fast response, overshoot, oscillation

Small gain K_p -> slow response, large steady-state error

$$\boxed{K_p e(t)}$$



PID controller (I term)

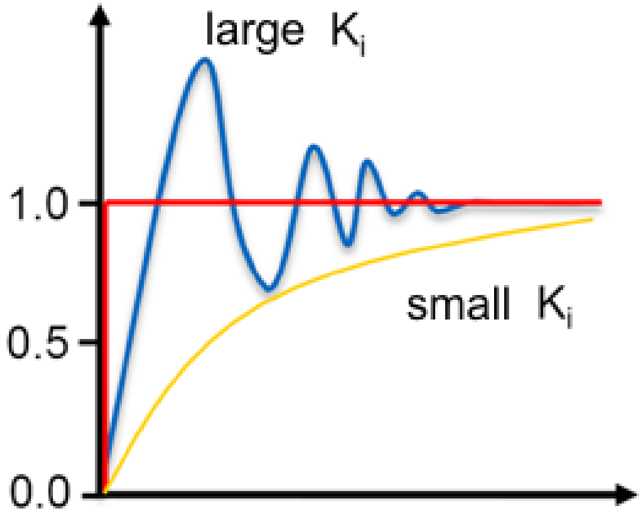
Output proportional to time integral of error

- Makes steady state error go to zero

Large K_I -> error decreases quickly, overshoot, oscillation

Small K_I -> error decreases slowly

$$\boxed{K_i \int e(t)dt}$$



A!

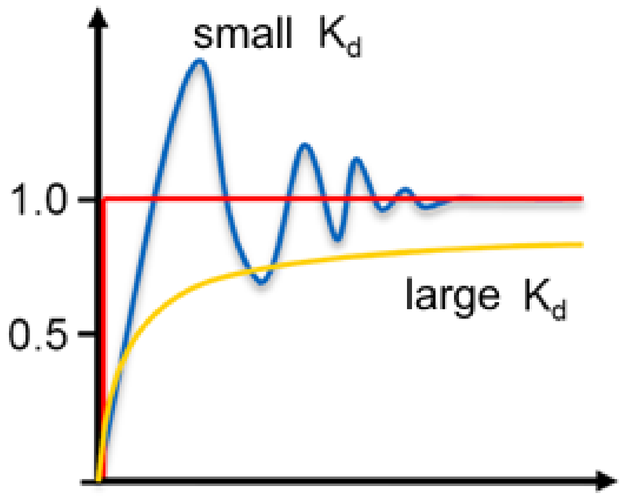
PID controller (D term)

Output proportional to change rate of the error

- K_d can dampen the response

If signal is noisy K_d may increase the interference

$$\boxed{K_d \frac{d}{dt} e(t)}$$



A!

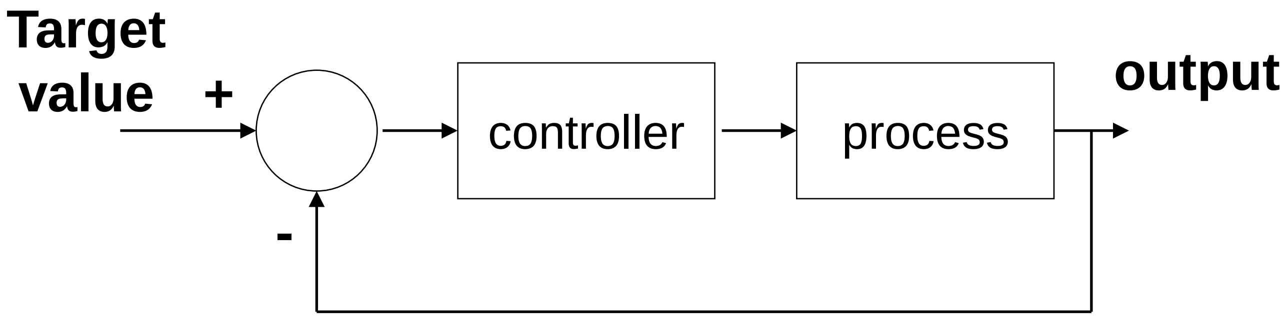
Parameter effects, warning, rough generalizations

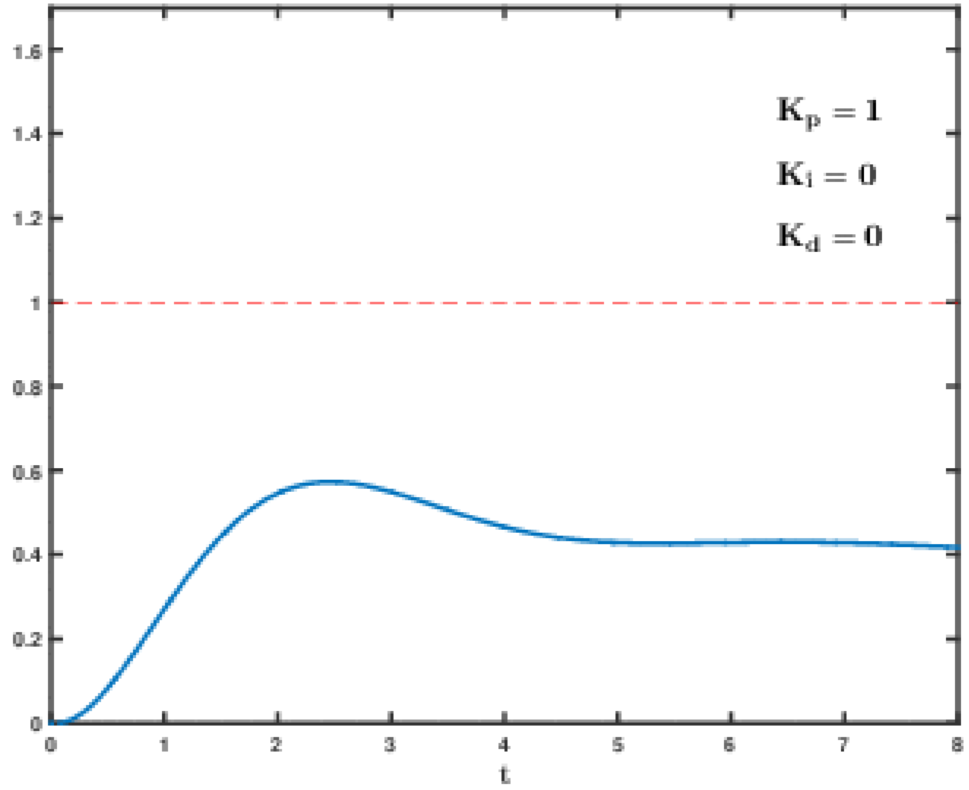
	Overshoot	Steady state error	Stability
K_P	+	-	-
K_I	+	-> 0	-
K_D	-	-	+/-

A!

Parameter effects

This is what we put in





Lecture task: PID-controller

Discuss in groups of 2-4 the operating principles of a PID-controller. If your colleague does not understand some part of it, teach!

- Why PID is cool and nice in many engineering applications?
- Negative feedback
- Proportional part
- Integral part
- Derivative part

If there is something unclear, write it down, and I will try to answer

PID controller tuning

How to find the right gains?

Experimenting with a real system – might be dangerous

- Apply input and observe

Experimenting with a system model – might be tedious

- Simulink model – curves – adjust gains

Iterative optimization with a system model – requires cost function

- For example a Simulink model with Matlab optimization functions

Analytical optimization with mathematical system model – might be difficult

- Transfer function – calculus

Example: Microcontroller code for PI controller

```
if(PID_on){
    reference = read_adc(0)/4;                          //Read reference from a potentiometer
    error = reference-speed;                            //Calculate error

    proportional = P*error;                             //Proportional part of control
    integral = integral + I*error;                      //Integral part of control
        control = proportional+integral;                //Controller output

    if(control<0){direction = 1;}else{direction = 0;}   //Choose motor direction
}

PORTB.3 = direction;                                    //Write to direction register
OCR1A = control;                                        //Write to PWM register
```

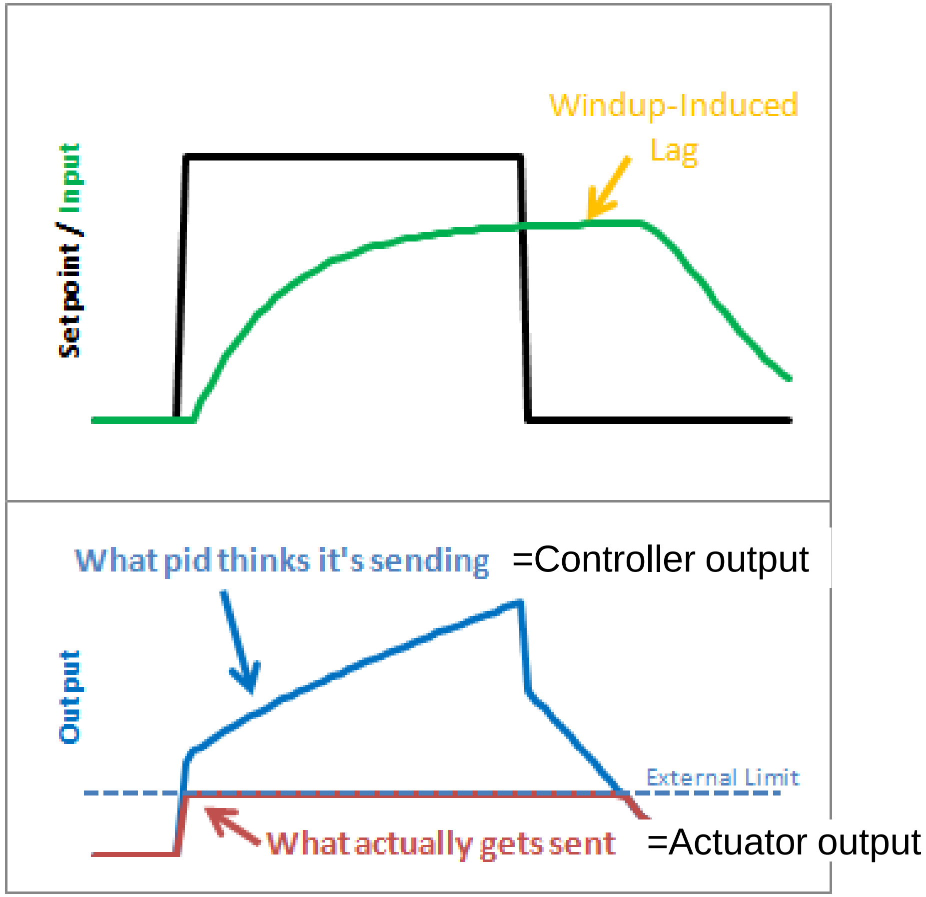
Actuator range and rate limitations

Real actuators have range and speed limitations

Actuator saturates -> modify control law

Anti-windup compensator

- Limits integral growth



A!

Analog vs digital controllers

Controllers implemented with digital devices do not work continuously

Analog controller can be implemented with operational amps

- High frequency, integration differentiation, summing etc
- Most controllers digital, some vefy fast control loops analog

Results slightly different in analog and digital controllers

Slightly different analysis methods

Control rate

In digital technology, not even time is continuous

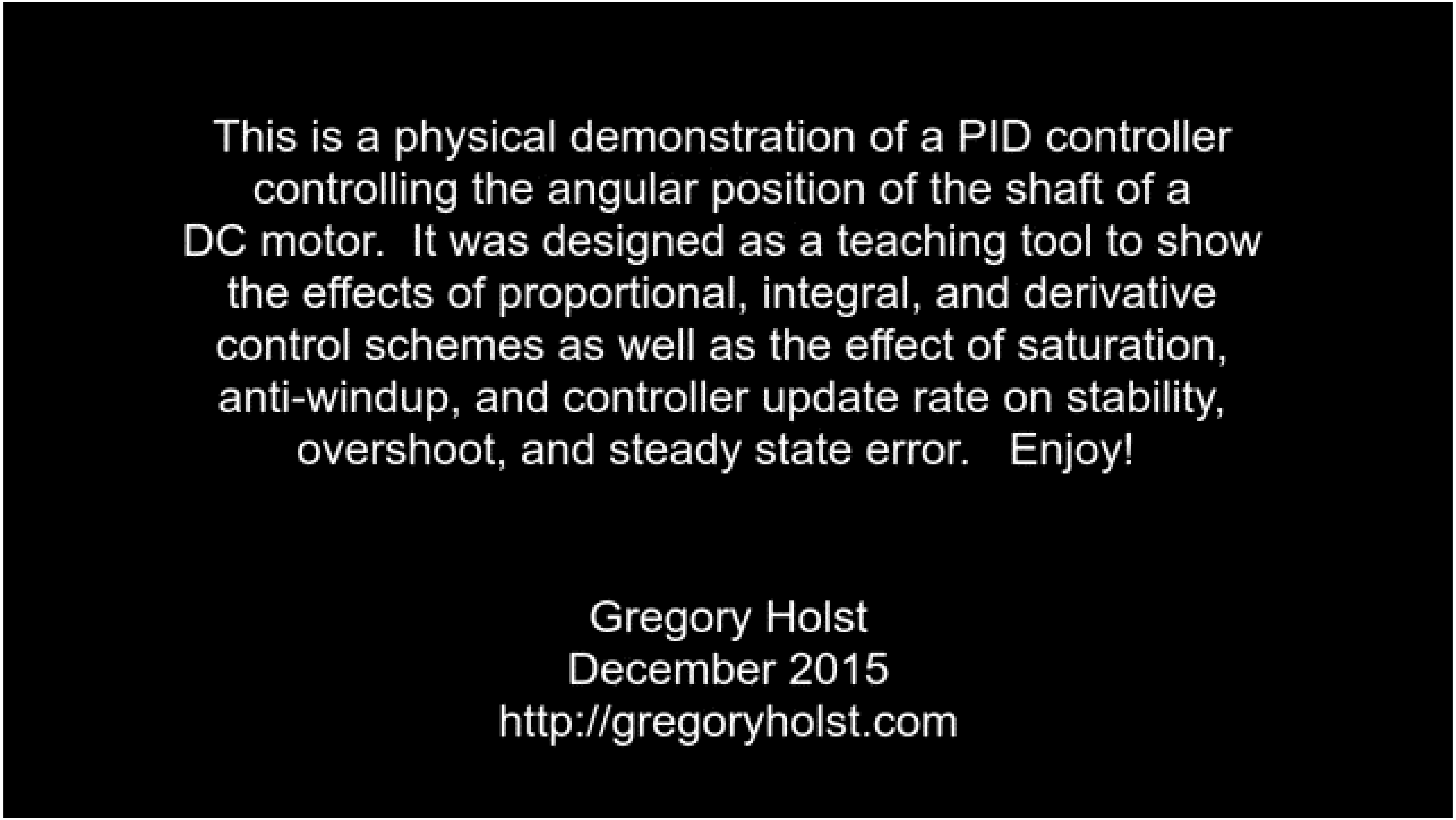
All tasks are done at certain intervals

- Measurements are either averaged or momentary
- Control algorithm calculated at constant and small enough control intervals

Control rate depends on time constant of the system

- Rise time from one state to another minimum 5-10 control loops
  - For a motor for example every 10 ms is often enough

PID position control demo



Other controllers

Model-based control

- Using a model of the system to predict its behaviour

Adaptive control

- Modifying controller parameters according to output

Optimal control

- Optimizing a controller to minimize a cost function

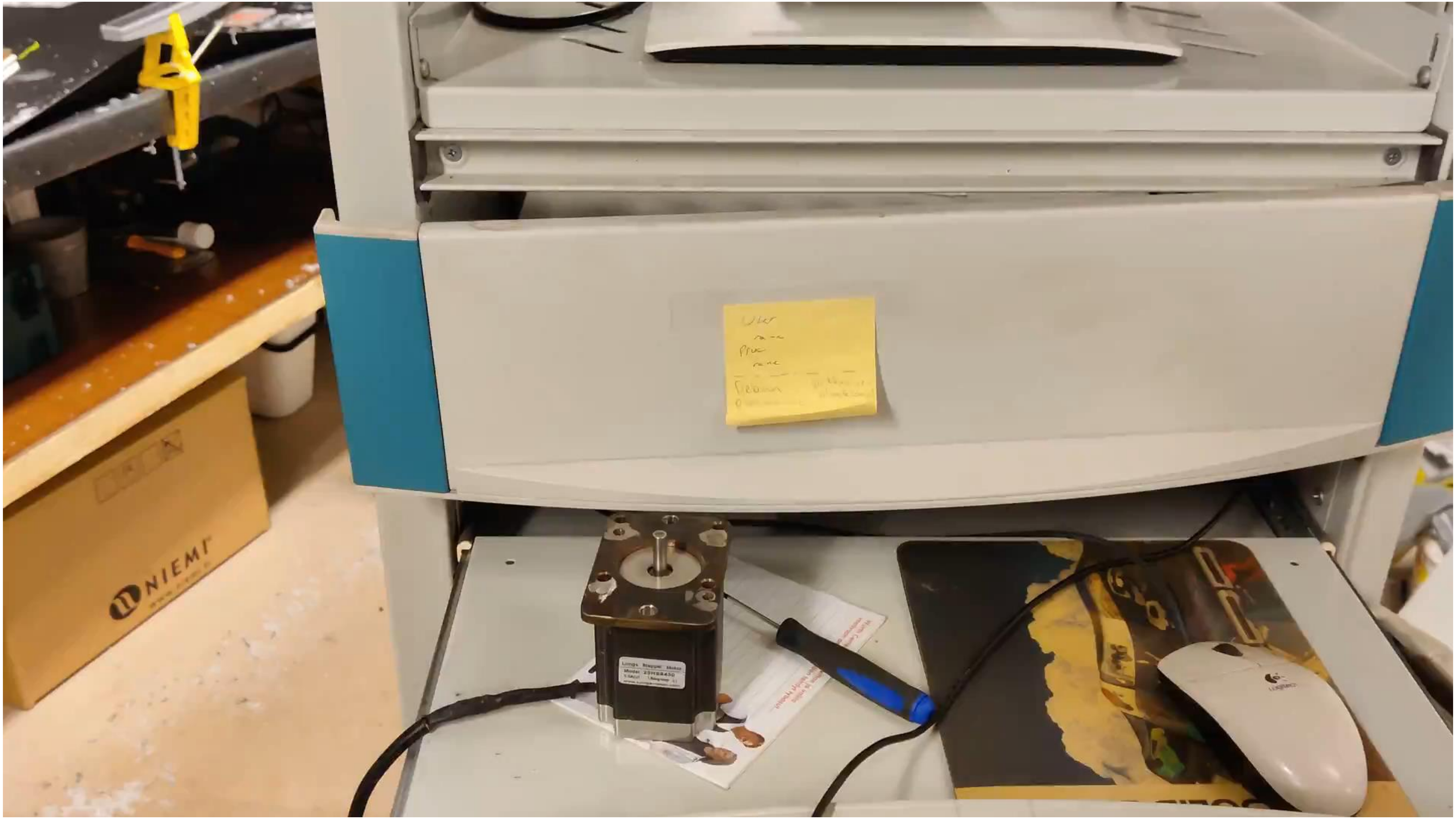
Robust control

- Designing a controller that works with disturbances

Fuzzy control

- Intuitive rule based controllers

Servo motors


NIEMI

Servo

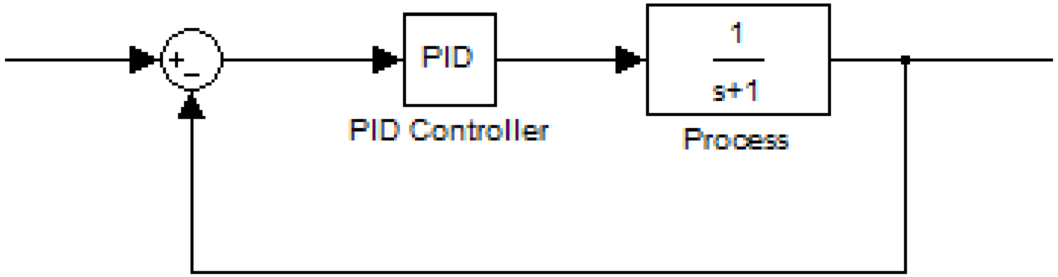
The origin of the word is believed to come from the French "*Le Servomoteur*" or the slavemotor, first used by J. J. L. Farcot in 1868 to describe hydraulic and steam engines for use in ship steering.

Latin servus = slave

Servomotor control system

Goal:

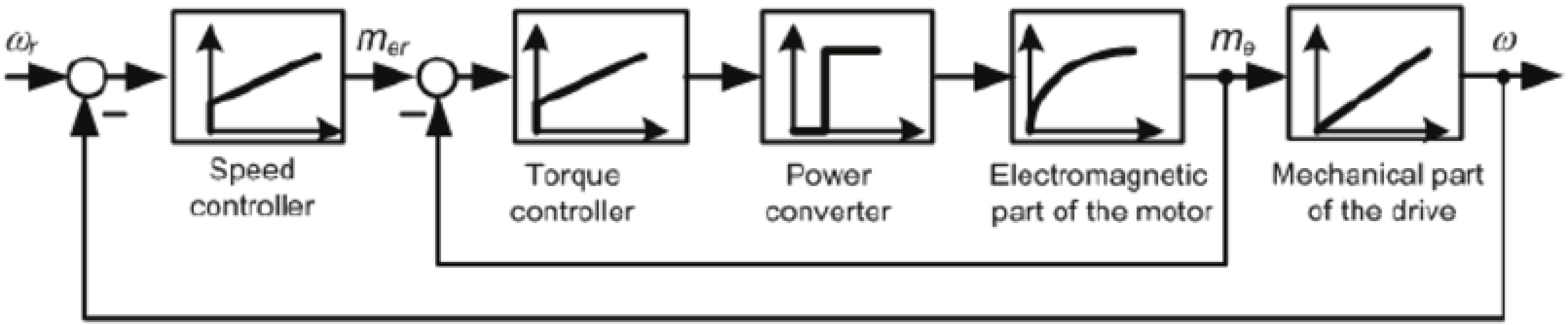
- Motor that spins at the required speed regardless of load or disturbances
  - Velocity feedback
- Actuator that keeps at or goes to the required position regardless of load or disturbances
  - Position feedback



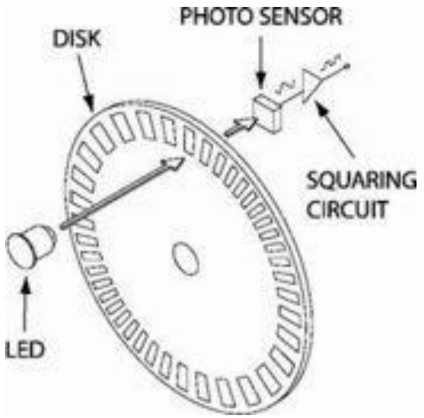
Cascade control

Control loop inside a control loop

- Outer loop for speed control
- Inner for torque (current)



Sensors for a servo motor

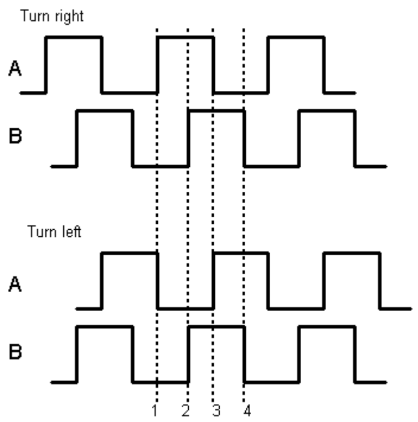


Encoders

- Incremental or absolute
- Optical, magnetic etc.

Resolvers & Synchros

- Rotating transformers which measure position



Tachogenerators

- ”Inverse motor”, measures rotating speed

Potentiometers

- Variable voltage divider

Motors for servo systems

DC (brushed or brushless)

AC (synchronous or asynchronous)

Integrated feedback sensors

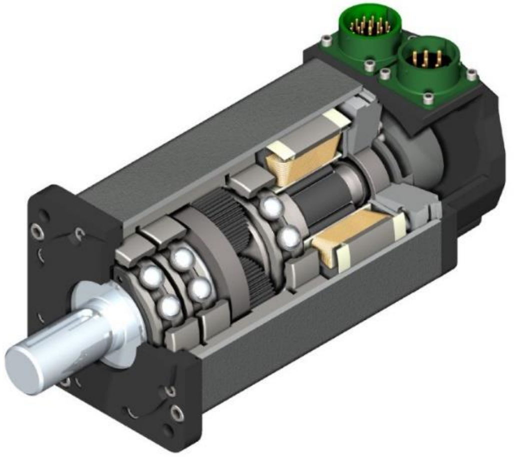
Ability to tolerate short term overloads

Low inductance

- Small electric time constant

Low rotor inertial mass

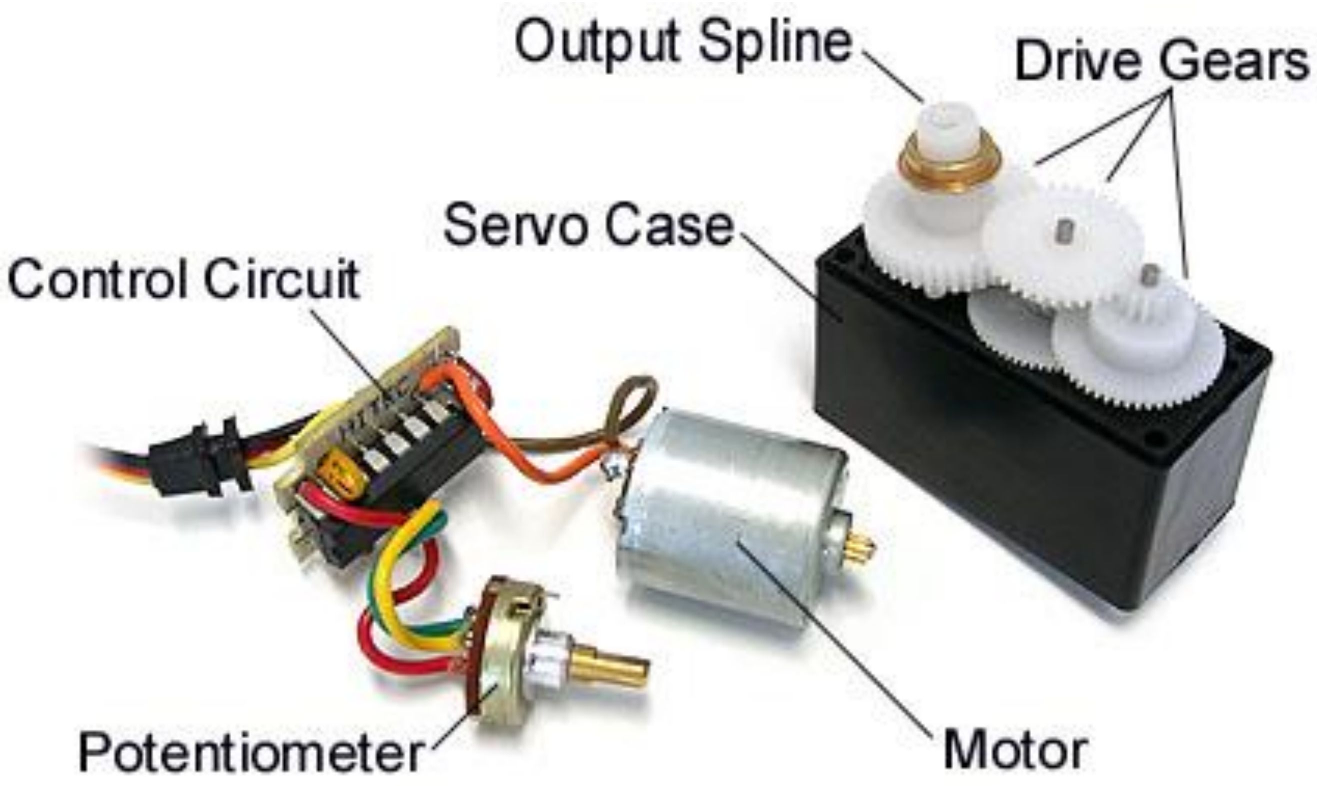
- Small mechanical time constant



Example: RC-servo

Built in components:

- Motor
- Reduction gear
- Potentiometer
- Control circuit
- Driver circuit



Application examples

Industrial robots

- E.g. ABB IRB 7600
- Payload up to 630 kg
- Positioning accuracy 0.1 mm

Machining centers

- Rotation to linear motion with ball screws
- Positioning accuracy 0.005 mm

Summary

A controller increases the performance of a system

Feedback control is used to reject disturbances

Feedback control can stabilize unstable systems

- Feedback control can also destabilize stable systems

A PID controller is easy to implement

- Every mechatronic engineer should understand it

Servomotor is a feedback controlled motor

- Servomotor is not a motor type