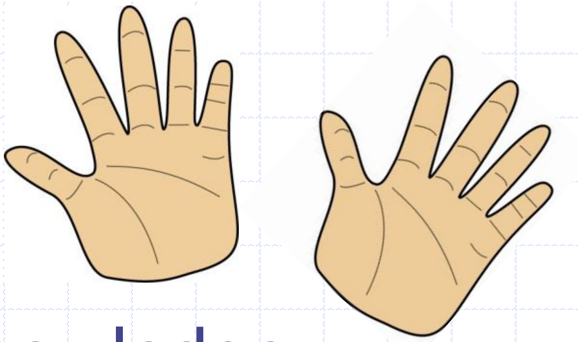
CESE4030
Embedded Systems Laboratory

Technology

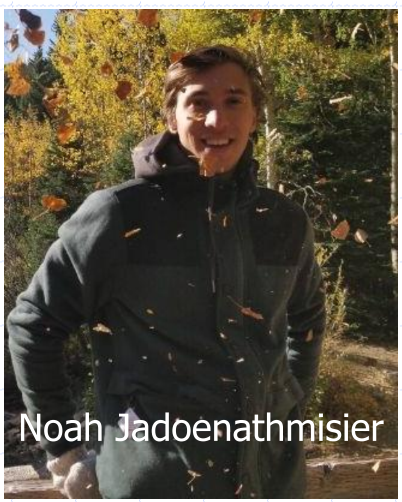
Introduction to Control ~~Theory~~

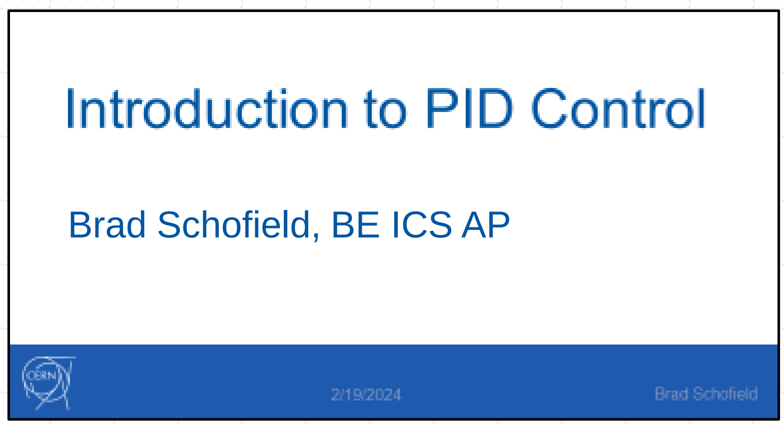
Disclaimer / acknowledgements

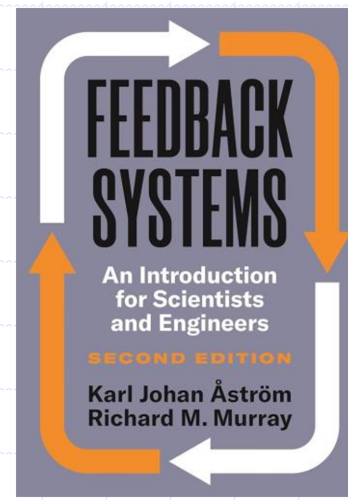
- Koen is a computer scientist



- Lecture is based on experts' knowledge

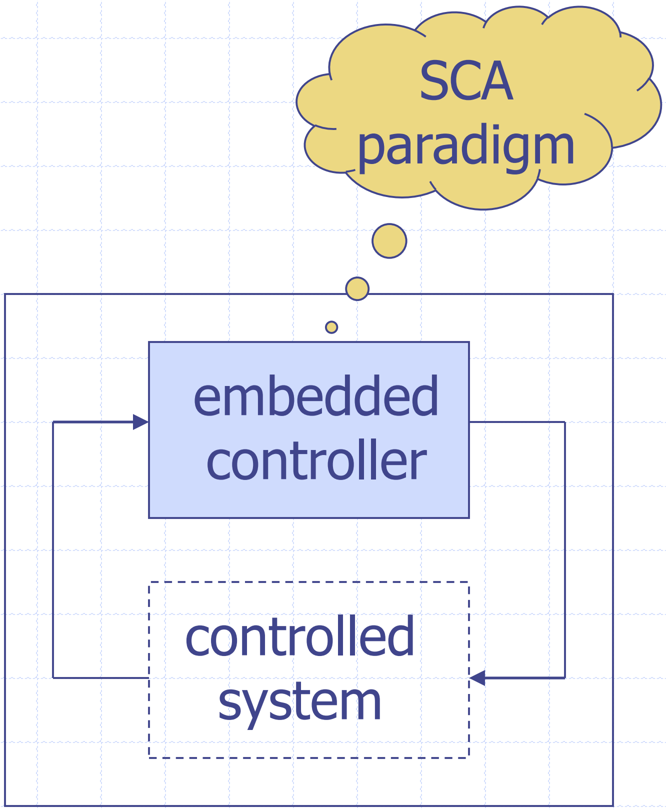




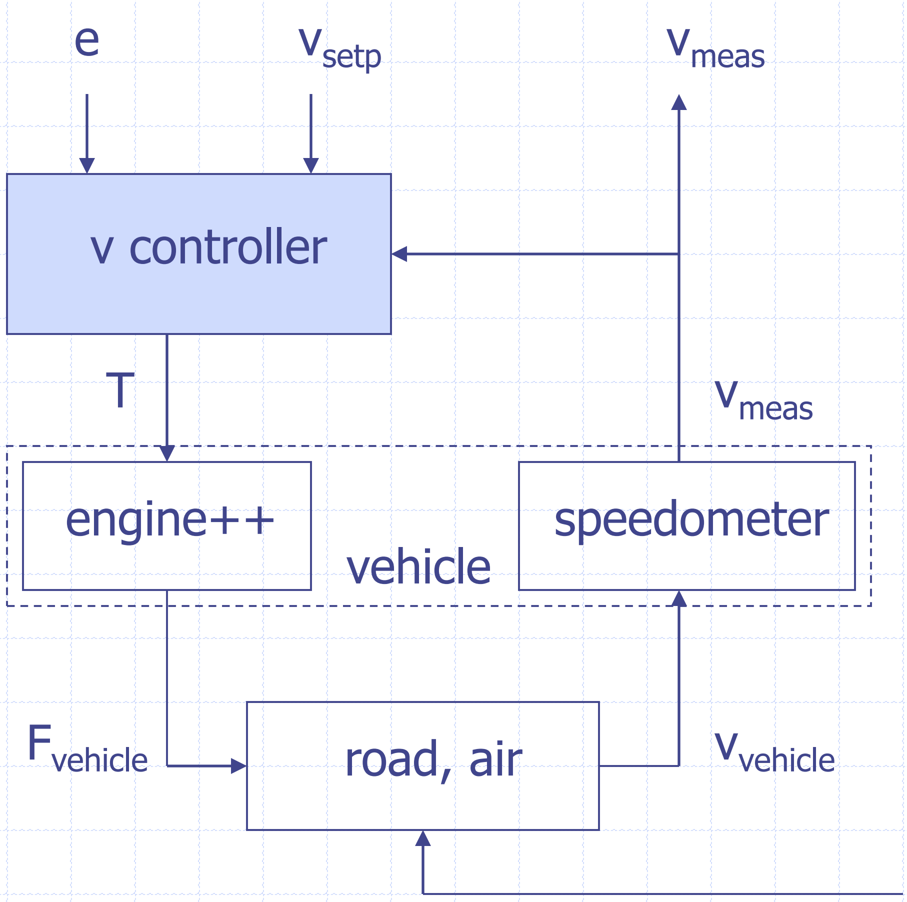


Control is Everywhere

- Automotive
- Aerospace
- Plant Control
- Climate Control
- Health Care
- Copiers, Wafer Scanners
- Model Quad Rotors ...



Cruise Control



e = enable [0/1]
T = throttle [%]
F = thrust [N]
v = velocity [m/s]

v_{setp} = setpoint
v_{meas} = measured
$v_{vehicle}$ = actual

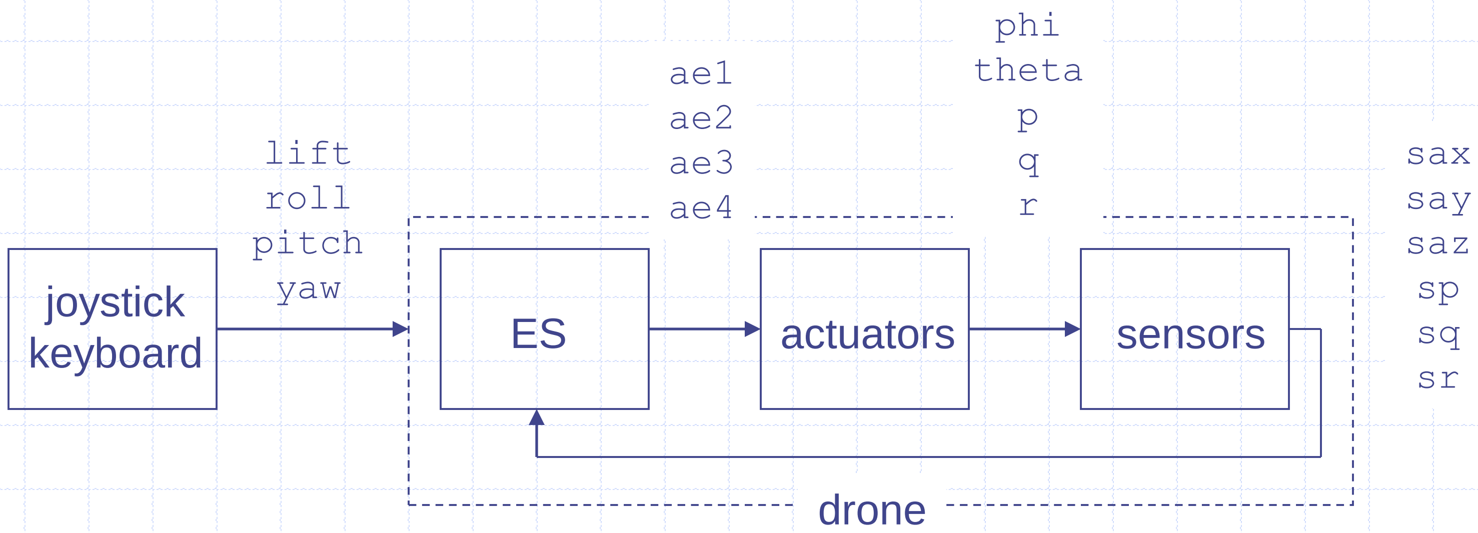
disturbances (slope, wind)

Objectives of this Crash Course

- Appreciate the benefits of control
- Understand basic control principles
- Communicate with control engineers

- Get you up to speed to do the QR control

Drone: Control Circuit



control loop example (yaw rate):

```
eps = yaw - sr;                // measure deviation
N_needed = P * eps;            // compute ctl action
ae1 .. ae4 = f(N_needed);      // actuate, see slide 9
```

Introduction to PID Control

Brad Schofield, BE ICS AP

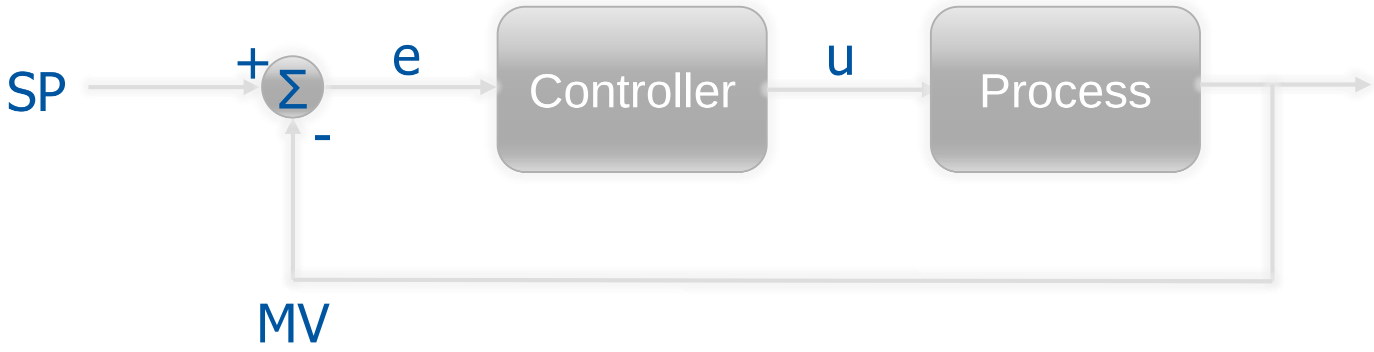
What is PID Control?

- Let's take a step back… What is **control**?

- **Control** is just making a dynamic process behave in the way we want

- We need 3 things to do this:
  - A way to **influence** the process
  - A way to see how the process **behaves**
  - A way to **define** how we want it to behave

The Closed Loop

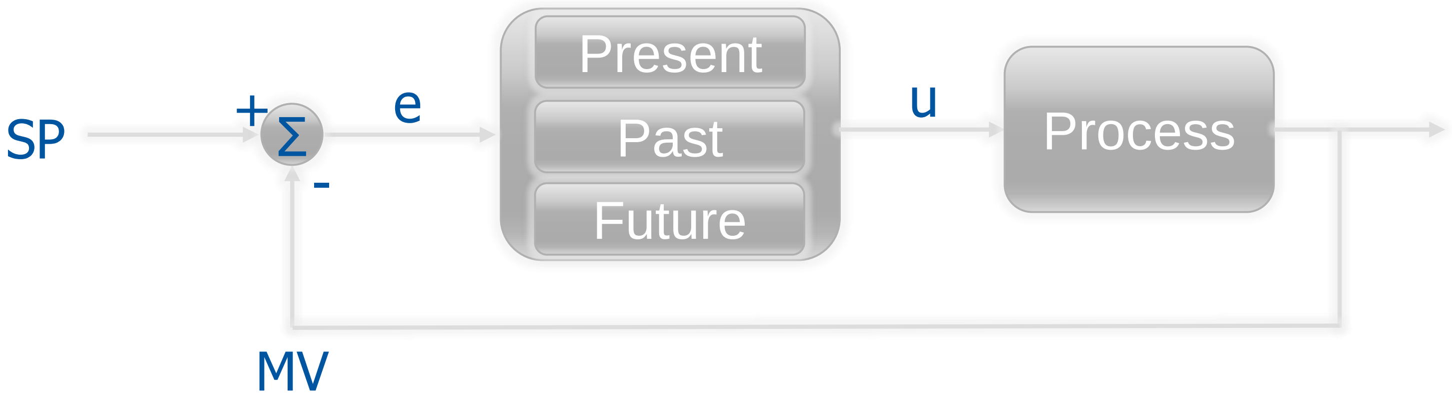


This is the ‘classic’ closed loop block diagram representation of a control system

A Dynamic Controller

- We said that since the process is dynamic (dependent on inputs made at different times), it makes sense that the controller should be too
- How do we usually think of time?
  - ‘Present’
  - ‘Past’
  - ‘Future’

Splitting the Controller



The ‘Present’

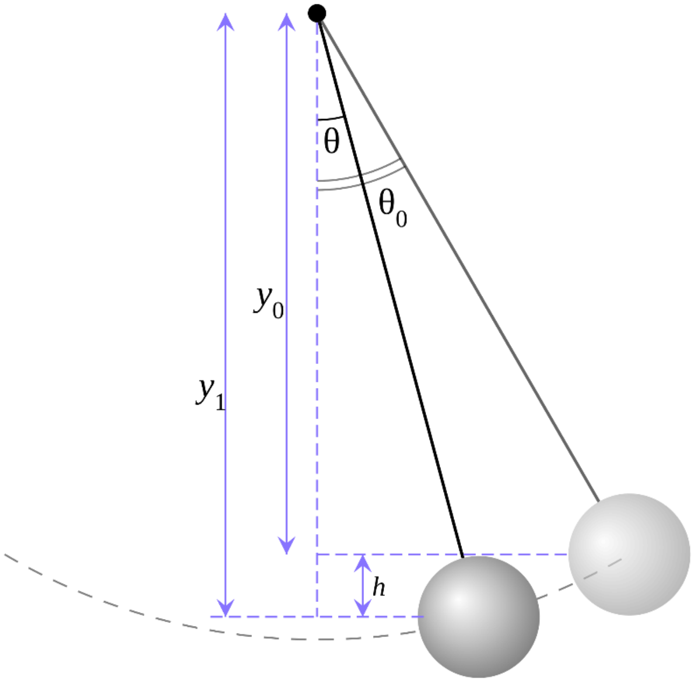
- This part of the controller is only concerned with what the **error** is **now**
- Let’s take a simple law: let the control signal be **proportional** to the error:

$$\boldsymbol{u = k_p \times e}$$

Is Proportional Control enough?

- Intuitively it seems like it should be fine on its own: when the error is big, the control input is big to correct it. As the error reduced so does the control input.
- But there are problems…

Problem 1: oscillations

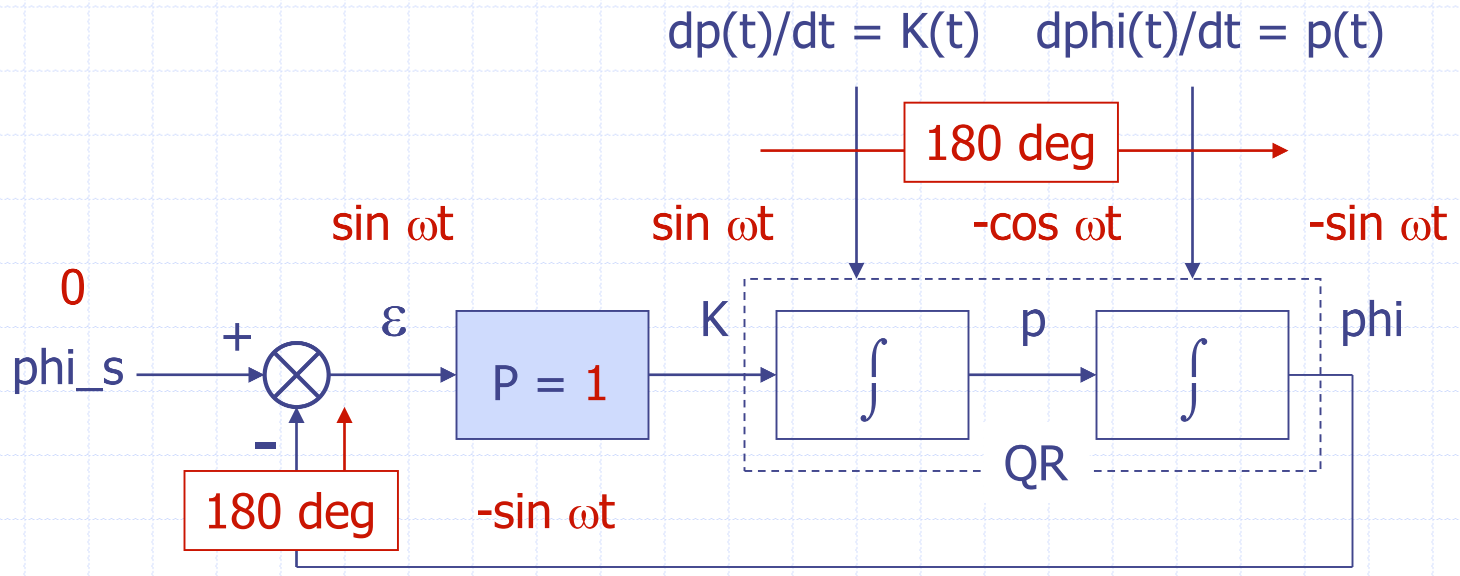


Think of a pendulum.

If the setpoint is hanging straight down, then gravity acts as a proportional controller for the position…

Pendulum will **oscillate**!

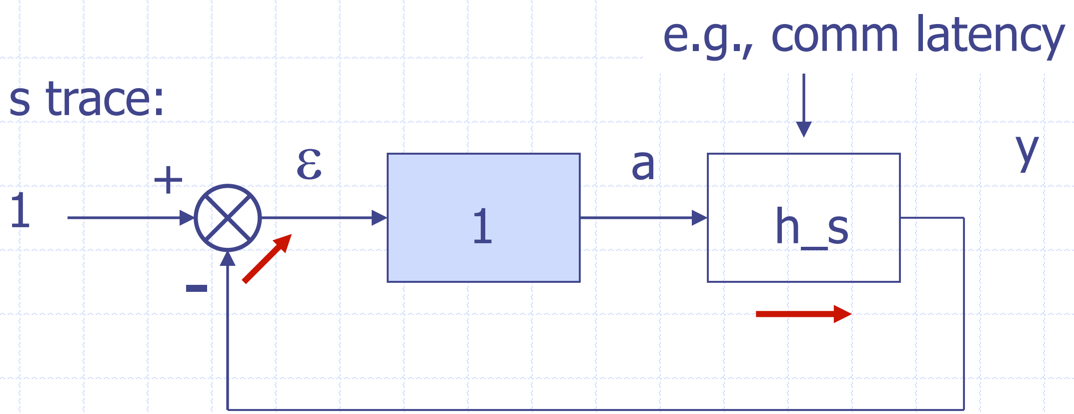
Example 1: Integrator Systems

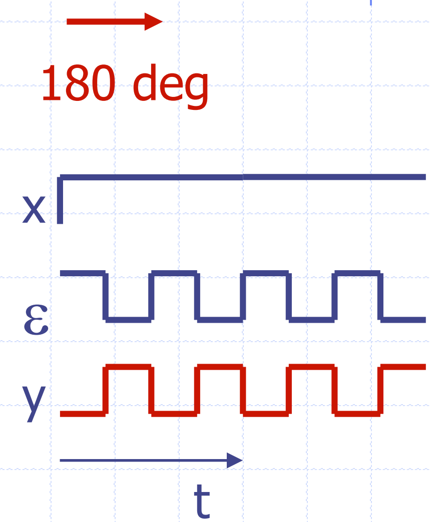


P ≥ 1: instability!
Cause: each integration adds 90 deg phase lag
So 2 integrators use up all 180 deg budget!

Example 2: Time Latency

0.5 s trace:





Let h_s: y(t) = a(t-0.5) (i.e., 0.5s delay)
Phase lag of 180 deg at 1 Hz causes instability

Phase Lag: examples

h_s

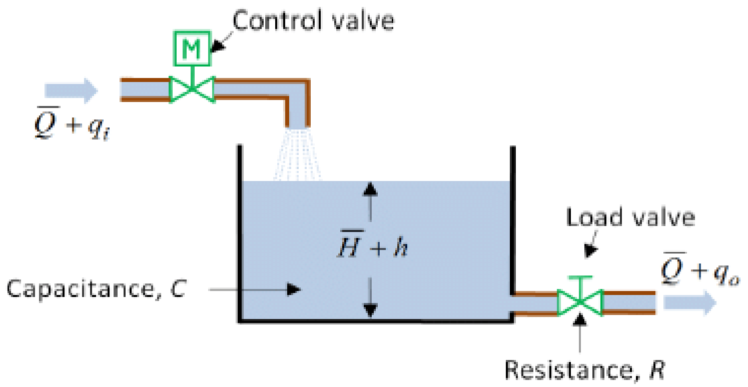
- Integration (90 deg):
  - speed -> position, flow -> volume
- First-order system (up to 90 deg):
  - lamp, heating, car velocity, ...
- N-th order system (up to N*90 deg):
  - compositions of 1st-order systems, missiles
- Delay systems (unlimited):
  - humans, computers, sample times, cables, air

Need control theory to analyze, e.g., control stability

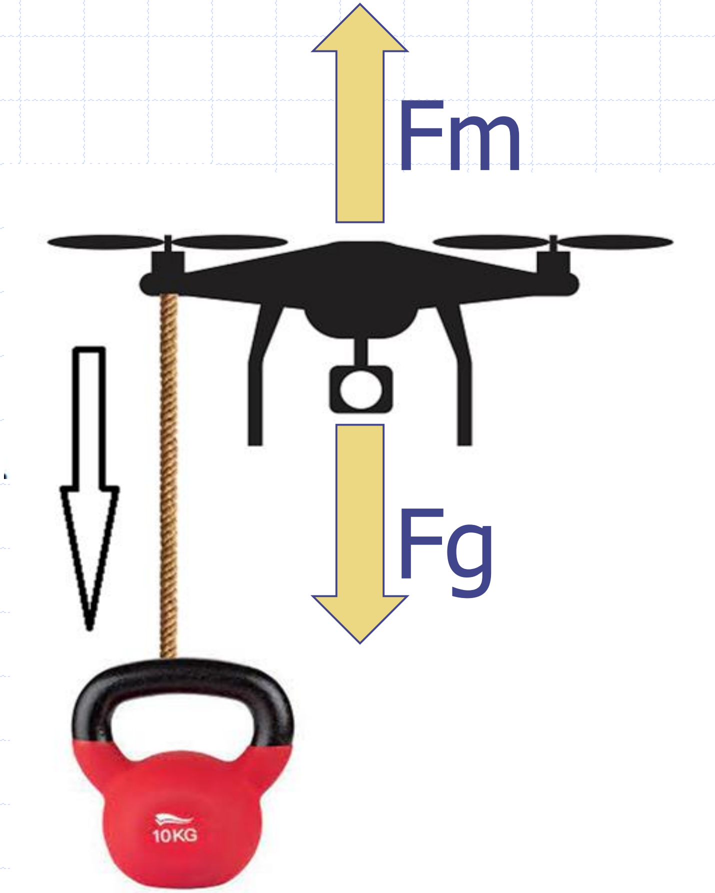
Problem 2: steady state error

- What happens when the error is zero?

- Causes problems if we need to have a nonzero control value while at our setpoint



Example: external disturbance



The solution to everything

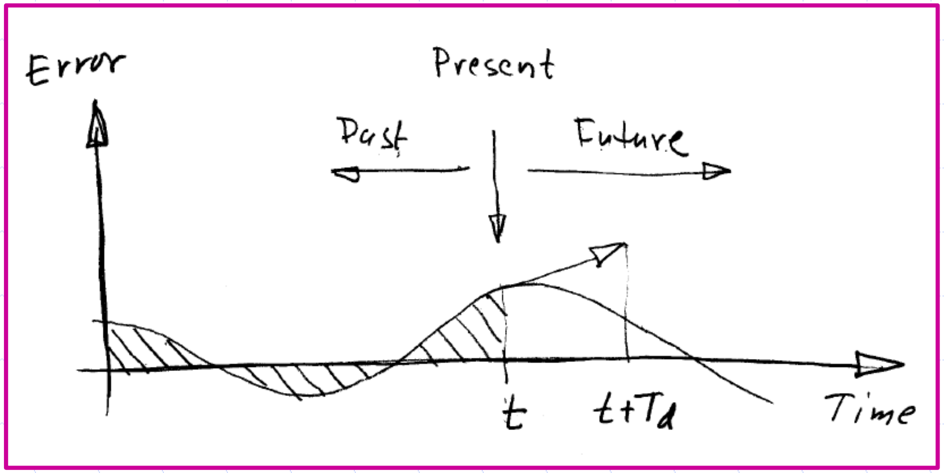
PID CONTROL

PID in a nutshell

The textbook version of the PID controller is

$$u(t) = ke(t) + k_i \int_0^t e(\tau)d\tau + k_d \frac{de}{dt}$$

$$u(t) = k\left(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d \frac{de(t)}{dt}\right)$$



PID control

- P: address errors
- D: address oscillations
- I: address steady state


Who is the boss?

PID tuning

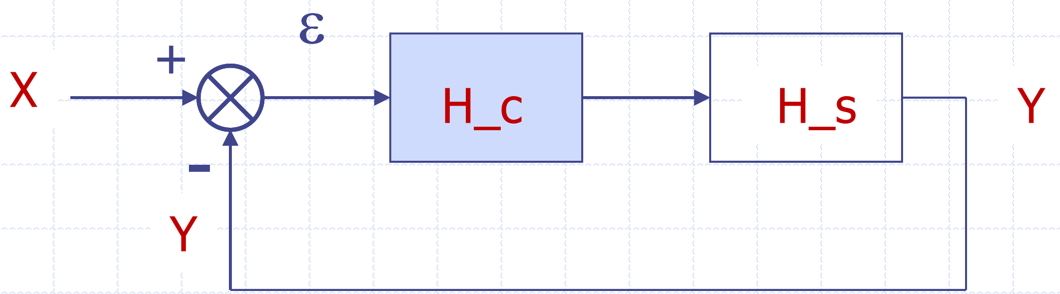
- P: address errors
- D: address oscillations
- I: address steady state

Integral windup

Max the error

Who is the boss?

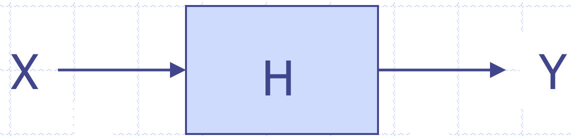
Classical Control Theory



Describe x(t), y(t), h_c(t), h_s(t) in terms of their Laplace transforms X(s), Y(s), H_c(s), H_s(s), respectively

$$\boxed{L[f(t)] = F(s) = \int_0^\infty f(t) e^{-st} dt}$$

Classical Control Theory

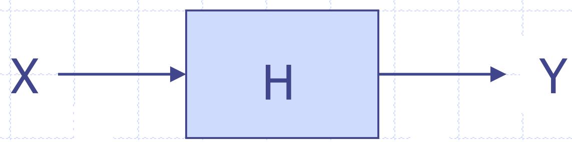


For linear system h it holds $Y(s) = H(s) \cdot X(s)$
(i.e. composition in time domain reduces to multiplication in the Laplace domain). This allows for easy analysis.

Laplace cheat sheet

- $L[a] = a/s$
- $L[a\ t] = a / s^2$
- $L[a\ f + b\ g] = a\ L[f\ ] + b\ L[g]$
  $= a\ F(s) + b\ G(s)$
- $L[f\ '\ ] = s\ L[f\ ] - f(0)$
  $= s\ F(s) - f(0)$

Control System Analysis



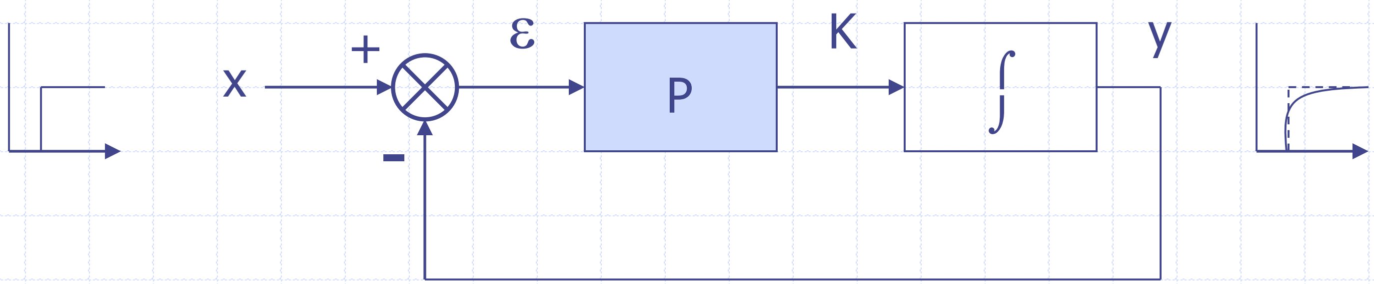
zeros

$$Y(s) = H(s)\,X(s) \qquad\qquad H(s) = \frac{Y(s)}{X(s)}$$

poles

Stability: Re(roots H(s)) < 0
Im(roots H(s)) small

Example: Rate Control



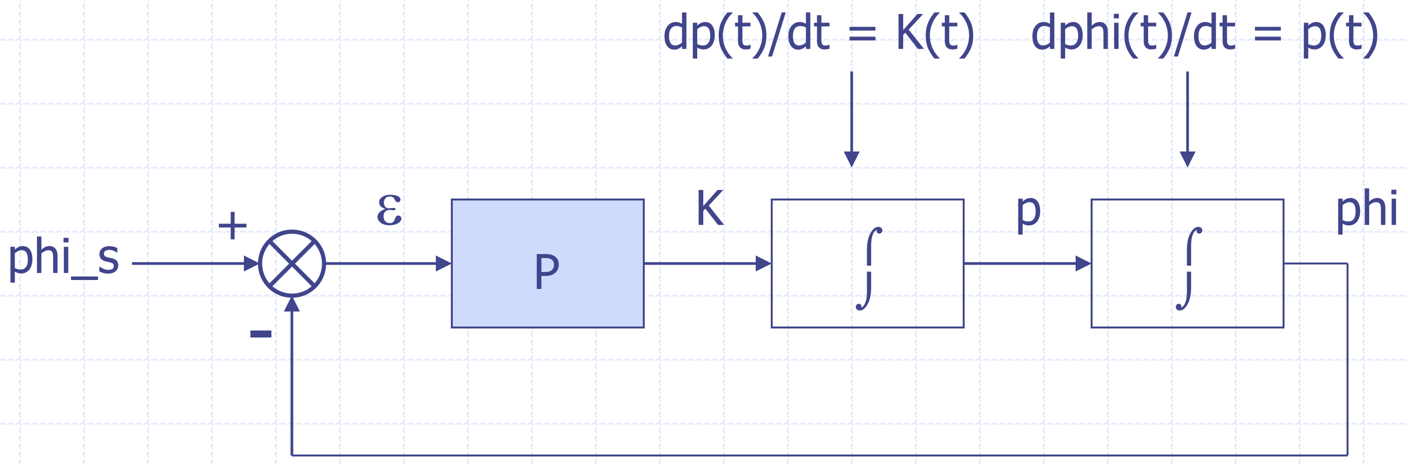
$Y(s) = P\,H(s)\,(X(s) - Y(s))$
$Y(s) = (P\,H(s) / (1 + P\,H(s)))\,X(s) = H_{PC}(s)\,X(s)$

$H(s) = 1/s$
$H_{PC}(s) = (P/s) / (1 + P/s) = P / (s + P)$

First-order system with time constant 1/P
(root: $s = -P \Rightarrow Re < 0, Im = 0$) so stable

Angle control using P controller

P controller for roll angle:

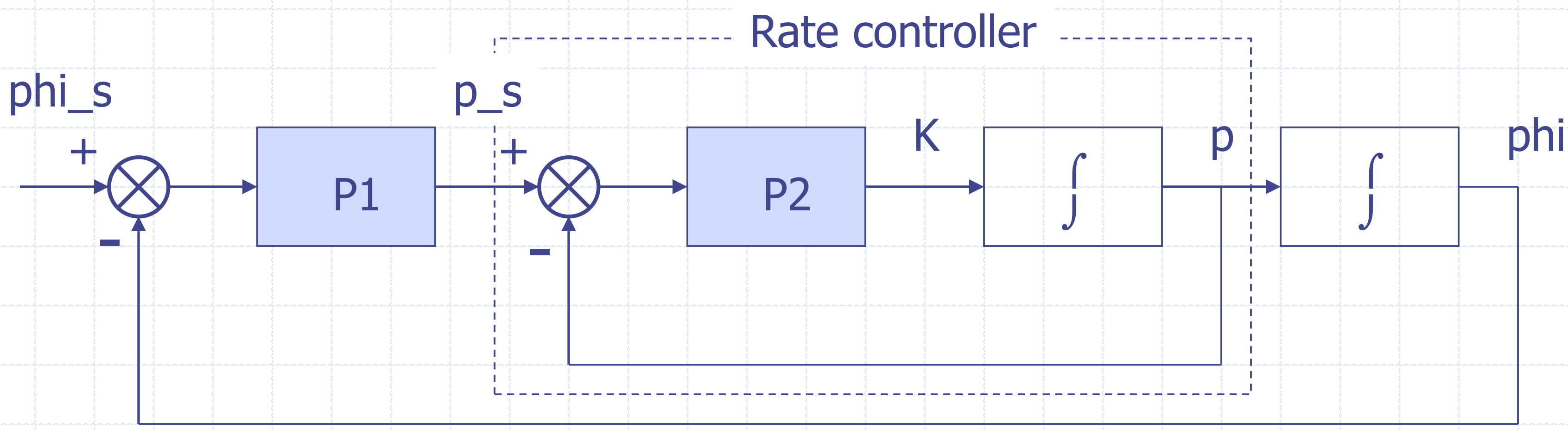


P < 1: useless control performance

P ≥ 1: instability

Angle control using cascaded P control

Embedded rate controller “neutralizes” one integrator



Cascaded P Controller: stable (for not too high P1 and P2! and P2 >> P1)

Summary

- Feedback control offers many advantages
- Is ubiquitous (cars, planes, missiles, QRs ..)
- Potential stability problems
- Need control theory
- This was merely introduction into the field
- Get a feel by applying to QR!