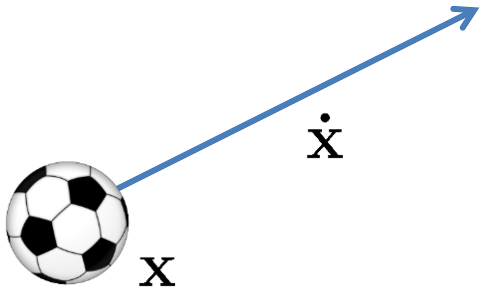
Autonomous Navigation for Flying Robots

Lecture 4.4 :
PID Control

Jürgen Sturm
Technische Universität München

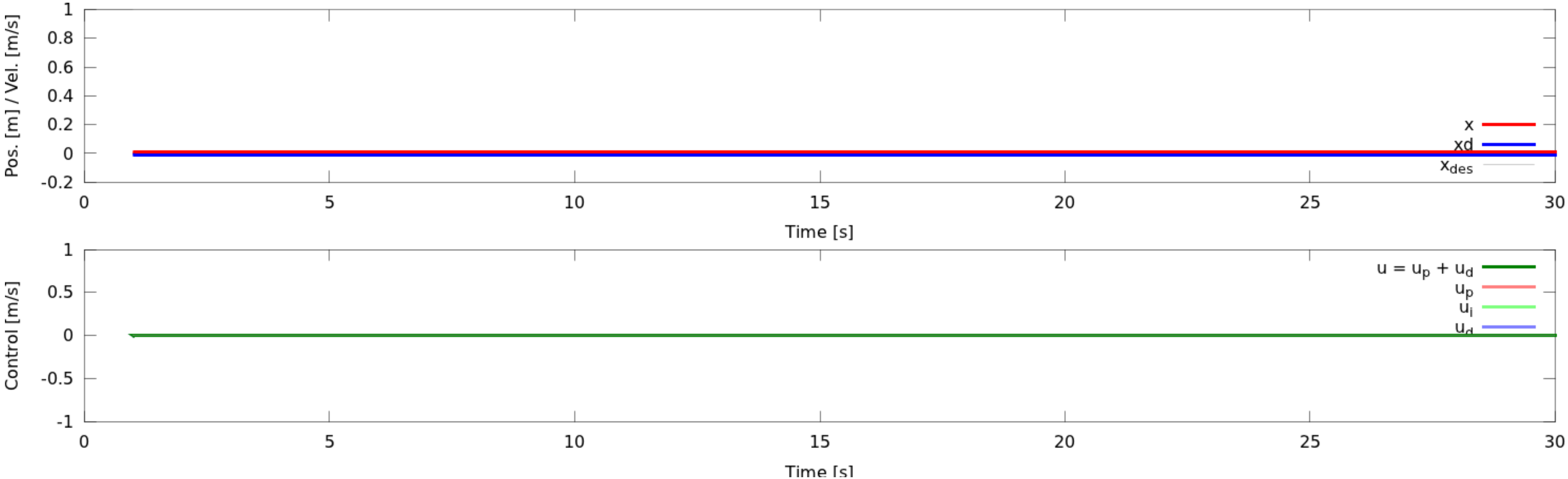
Rigid Body Kinematics

- Consider a rigid body
- Free floating in 1D space, no gravity



Rigid Body Kinematics

- System model $\mathbf{x}_t = \mathbf{x}_{t-1} + \dot{\mathbf{x}}$
- Initial state $\mathbf{x}_0 = 0, \dot{\mathbf{x}}_0 = 0$



Rigid Body Kinematics

- In each time instant, we can apply a force $\mathbf{F}_t \propto \mathbf{u}_t$
- Results in acceleration $\ddot{\mathbf{x}}_t = \mathbf{F}_t / m$
- Desired position $\mathbf{x}_{\text{des}} = 1$

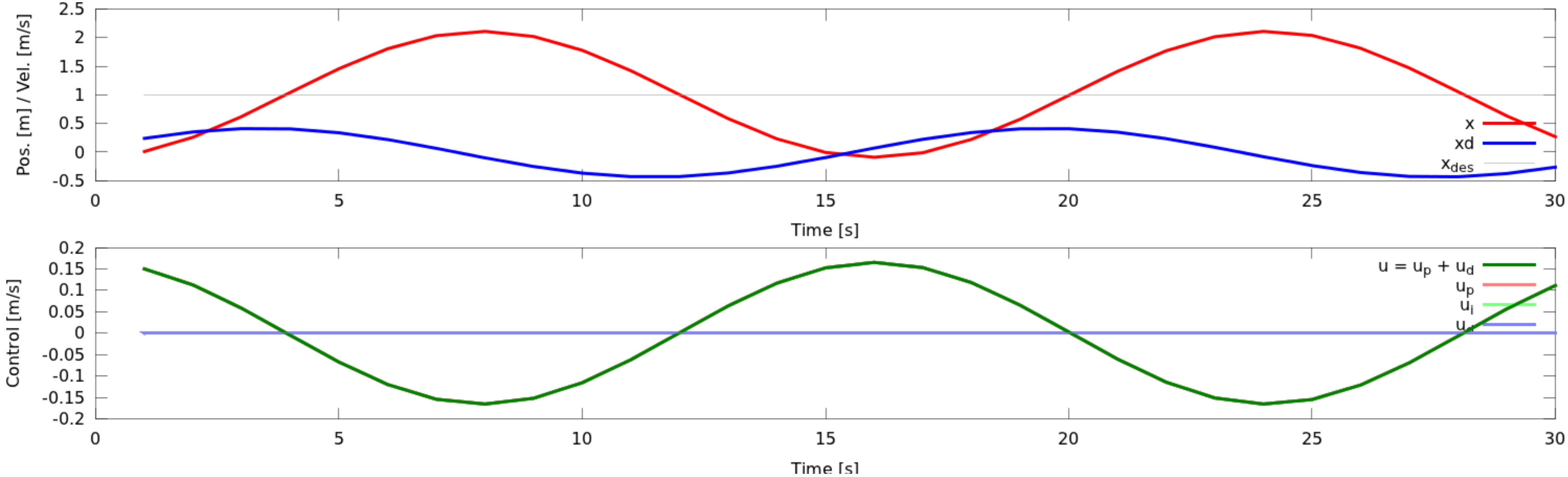
- What will happen if we apply P-control?

$$\mathbf{u}_t = K(\mathbf{x}_{\text{des}} - \mathbf{x}_{t-1})$$

P Control

- Control law

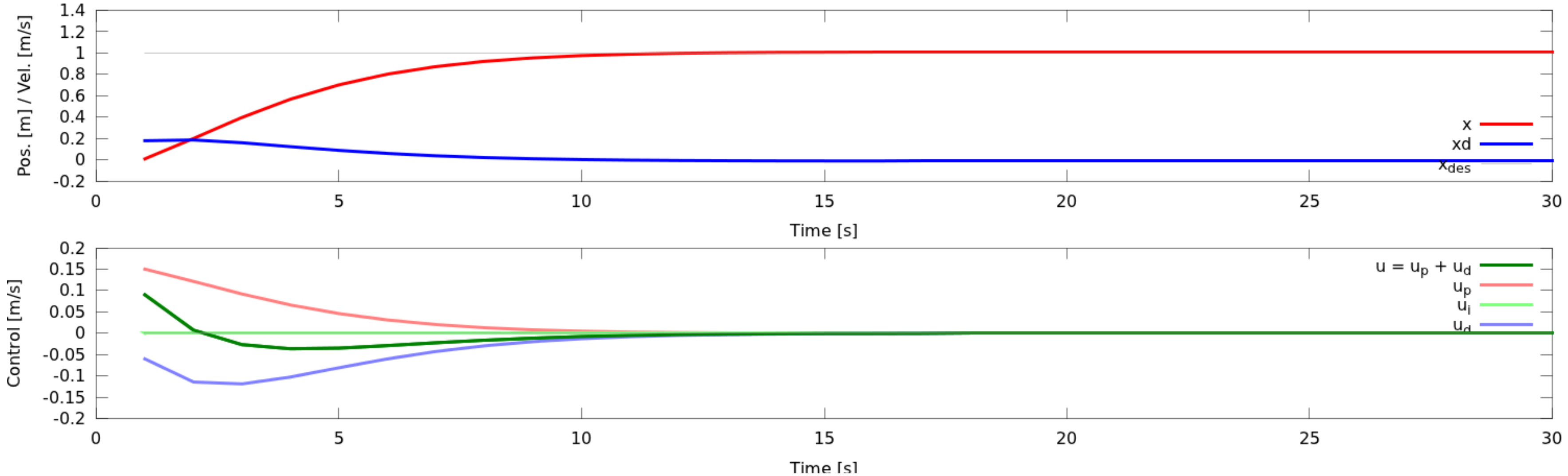
$$\mathbf{u}_t = K(\mathbf{x}_{\text{des}} - \mathbf{x}_{t-1})$$



PD Control

- Proportional-derivative control

$$\mathbf{u}_t = K_P(\mathbf{x}_{\text{des}} - \mathbf{x}_{t-1}) + K_D(\dot{\mathbf{x}}_{\text{des}} - \dot{\mathbf{x}}_{t-1})$$

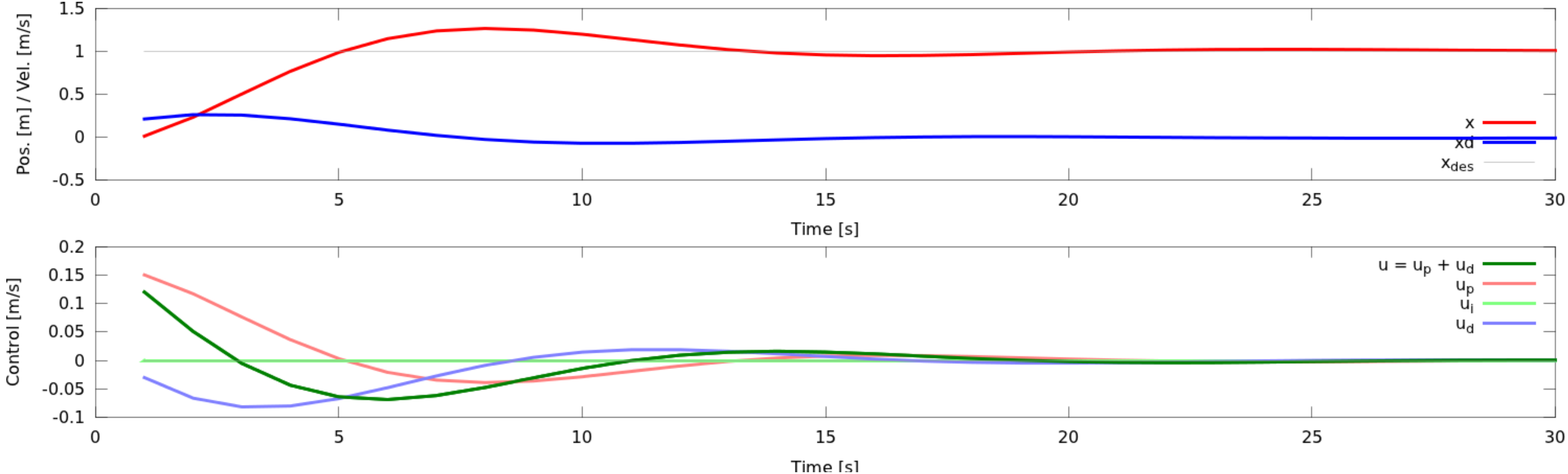


PD Control

- Proportional-derivative control

$$\mathbf{u}_t = K_P(\mathbf{x}_{\text{des}} - \mathbf{x}_{t-1}) + K_D(\dot{\mathbf{x}}_{\text{des}} - \dot{\mathbf{x}}_{t-1})$$

- What if we set lower gains for K_D?

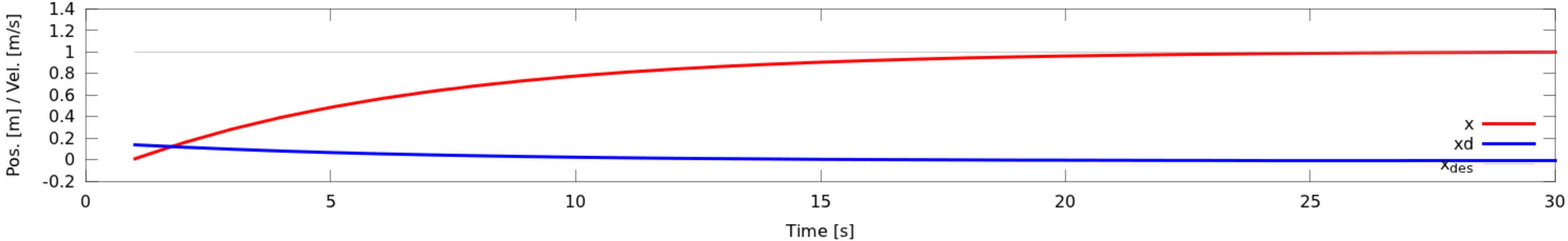


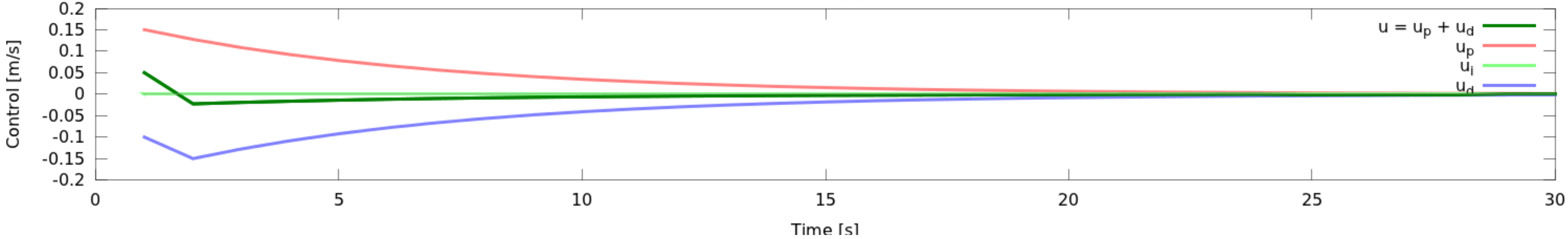
PD Control

- Proportional-derivative control

$$\mathbf{u}_t = K_P(\mathbf{x}_{\text{des}} - \mathbf{x}_{t-1}) + K_D(\dot{\mathbf{x}}_{\text{des}} - \dot{\mathbf{x}}_{t-1})$$

- What if we set higher gains for K_D?

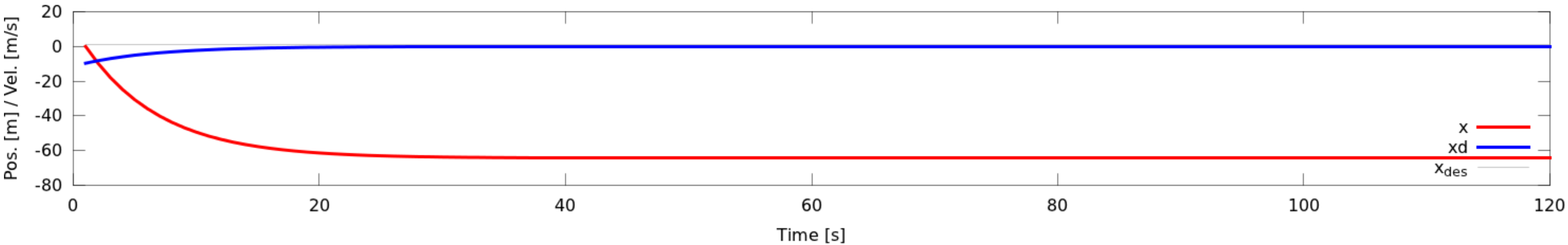


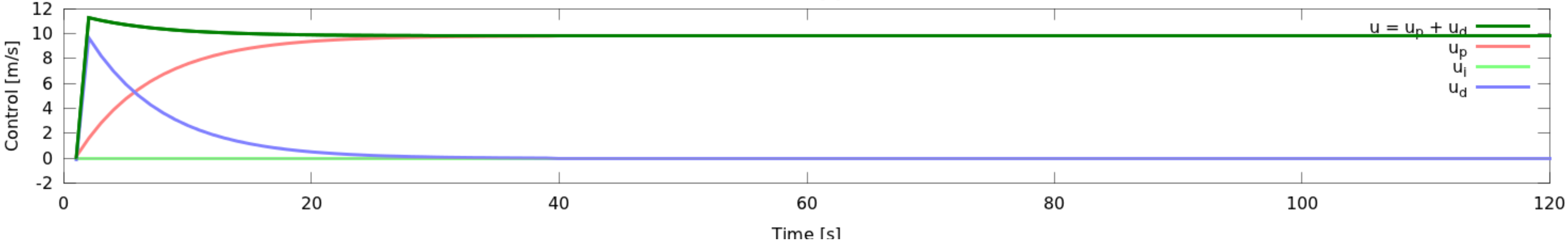


PD Control

- What happens when we add gravity?

$$\ddot{\mathbf{x}}_t = (\mathbf{F}_t + \mathbf{F}_{\text{grav}})/m$$



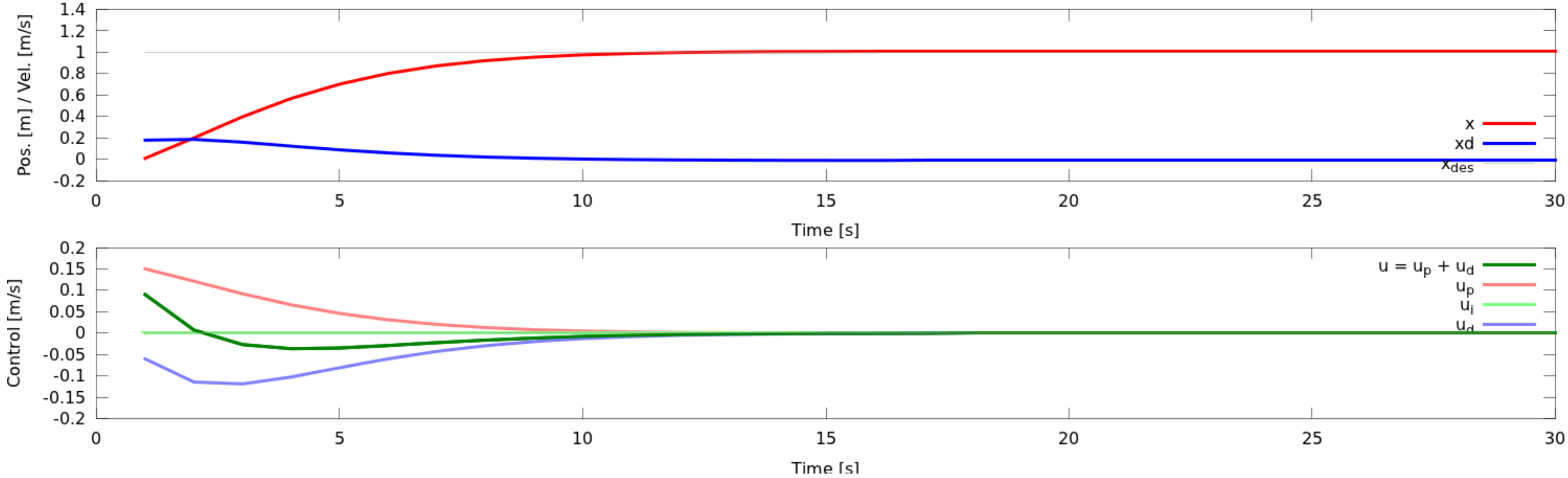


Gravity compensation

- Add as an additional term in the control law

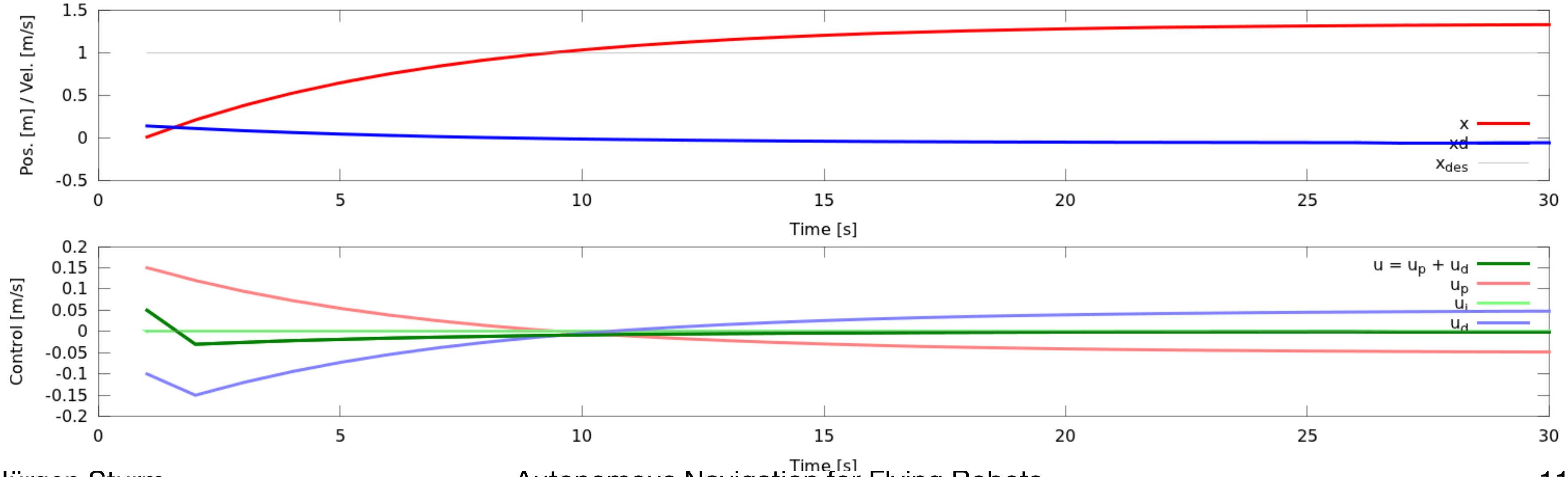
$$\mathbf{u}_t = K_P(\mathbf{x}_{\text{des}} - \mathbf{x}_{t-1}) + K_D(\dot{\mathbf{x}}_{\text{des}} - \dot{\mathbf{x}}_{t-1}) - \mathbf{F}_{\text{grav}}$$

- Any known (inverse) dynamics can be included



PD Control

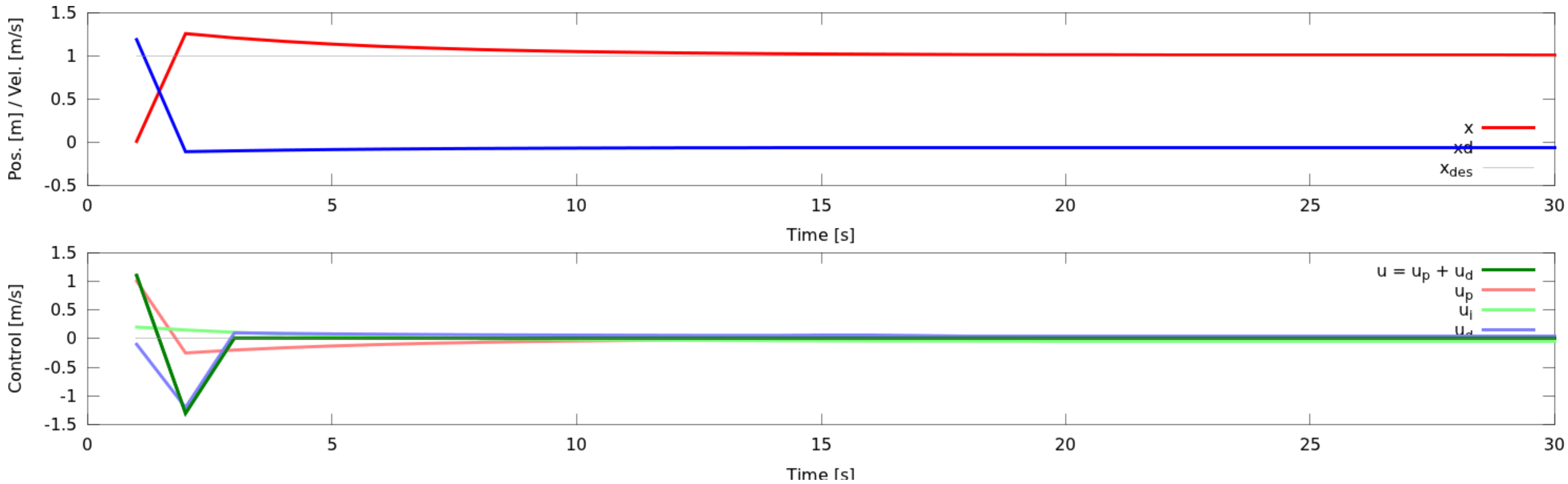
- What happens when we have systematic errors? (control/sensor noise with non-zero mean)
- Example: unbalanced quadrotor, wind, …



PID Control

- Idea: Estimate the system error (bias) by integrating error

$$\mathbf{u}_t = K_P(\mathbf{x}_{\text{des}} - \mathbf{x}_{t-1}) + K_D(\dot{\mathbf{x}}_{\text{des}} - \dot{\mathbf{x}}_{t-1}) + K_I \int_0^t \mathbf{x}_{\text{des}} - \mathbf{x}_{t'-1} \mathrm{d}t'$$



PID Control

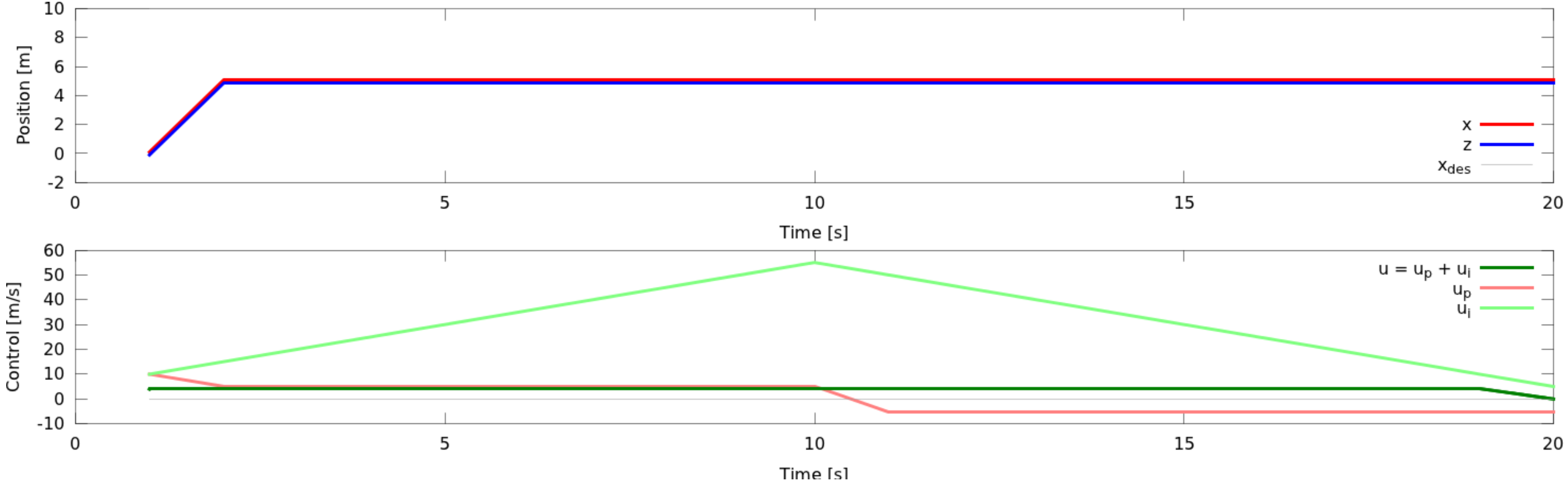
- Idea: Estimate the system error (bias) by integrating error

$$\mathbf{u}_t = K_P(\mathbf{x}_{\text{des}} - \mathbf{x}_{t-1}) + K_D(\dot{\mathbf{x}}_{\text{des}} - \dot{\mathbf{x}}_{t-1}) + K_I \int_0^t \mathbf{x}_{\text{des}} - \mathbf{x}_{t'-1} \mathrm{d}t'$$

- For steady state systems, this can be reasonable
- Otherwise, it may create havoc or even disaster (wind-up effect)

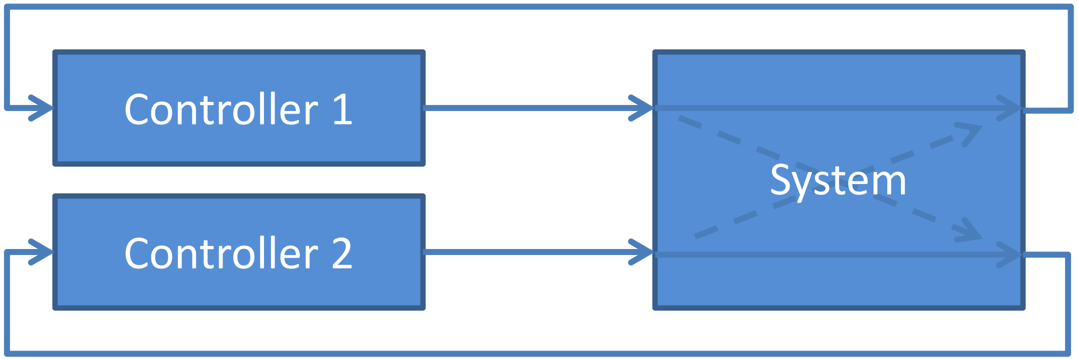
Example: Wind-up effect

- Quadrotor gets stuck in a tree → does not reach steady state
- What is the effect on the I-term?



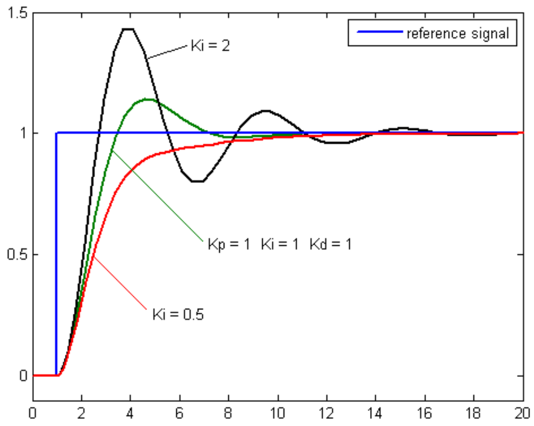
De-coupled Control

- So far, we considered only single-input, single-output systems (SISO)
- Real systems have multiple inputs + outputs
- MIMO (multiple-input, multiple-output)
- In practice, control is often de-coupled

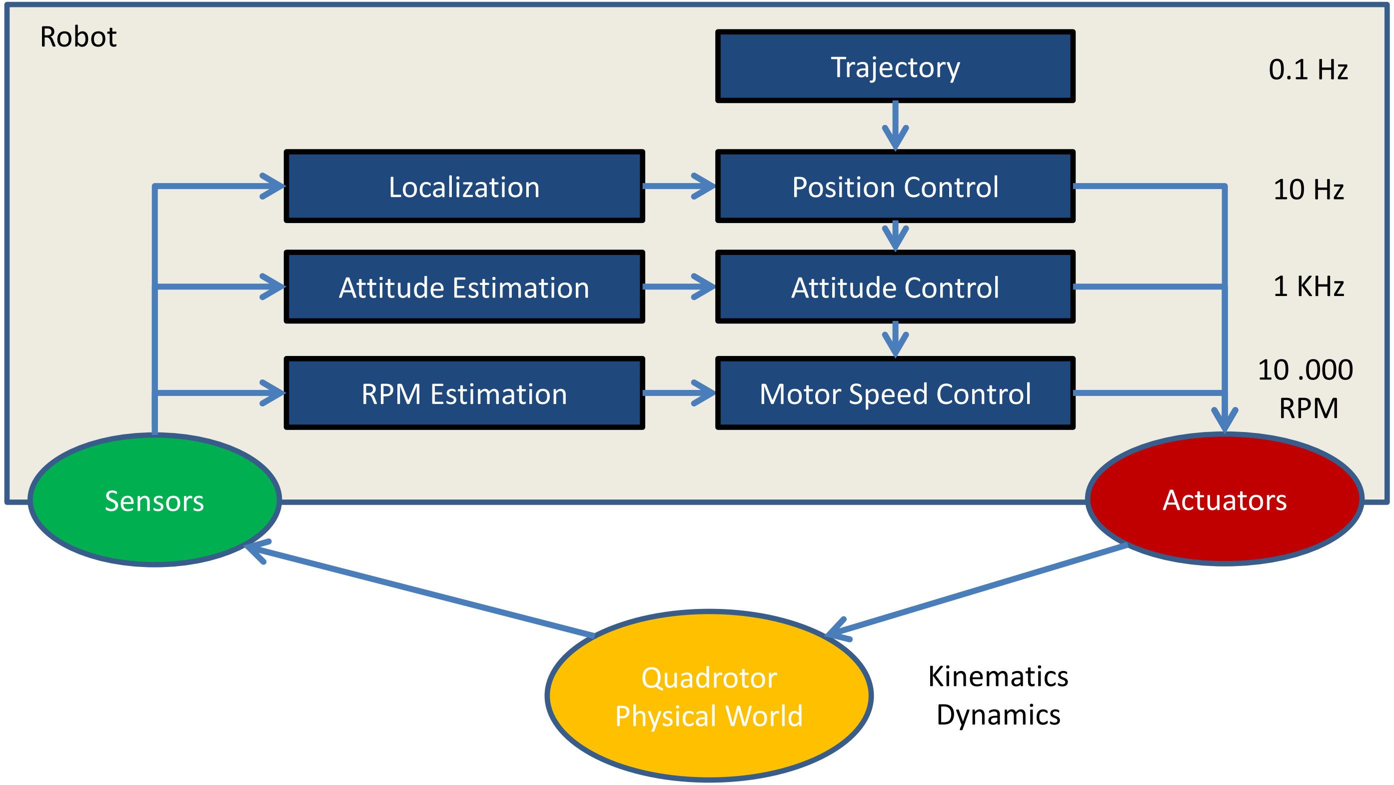


How to Choose the Coefficients?

- Gains too large: overshooting, oscillations
- Gains too small: long time to converge
- Heuristic methods exist
- In practice, often tuned manually



Cascaded Control



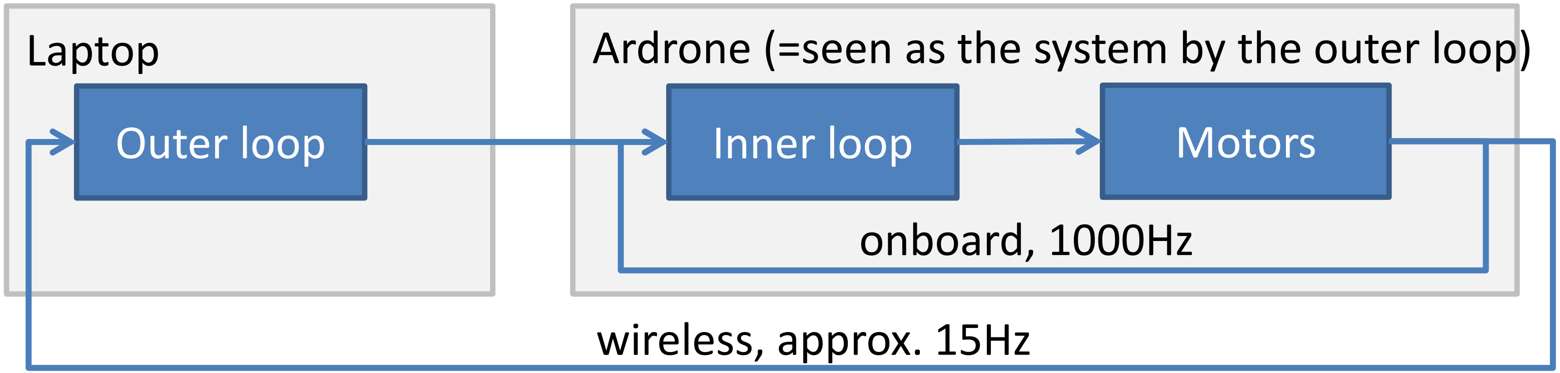
Assumptions of Cascaded Control

- Dynamics of inner loops is so fast that it is not visible from outer loops
- Dynamics of outer loops is so slow that it appears as static to the inner loops

Example: Ardrone

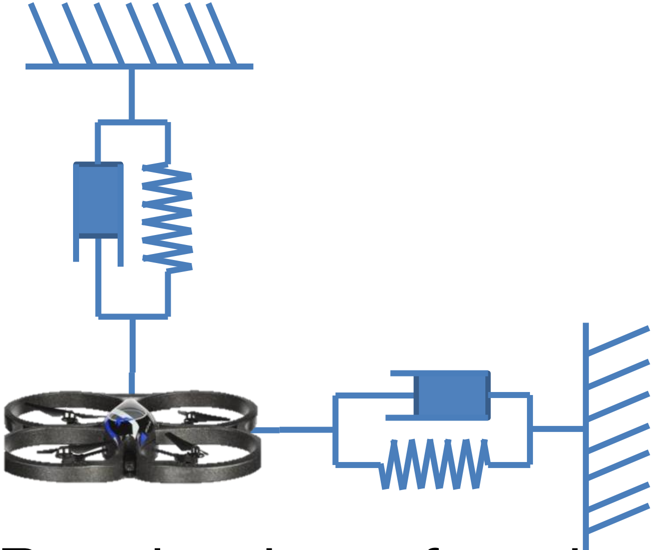
Cascaded control

- Inner loop runs on embedded PC and controls attitude
- Outer loop runs externally and implements position control



Mechanical Equivalent

- PD Control is equivalent to adding spring-dampers between the desired values and the current position



- Run the demo from http://wiki.ros.org/tum_ardrone

Lessons Learned

- P – proportional term
- I – integral term
- D – derivative term
- De-coupled control
- Cascaded control