اصول علم رباتات — اسلاید سیزدهم

Fundamentals of Robotics – Slide 13

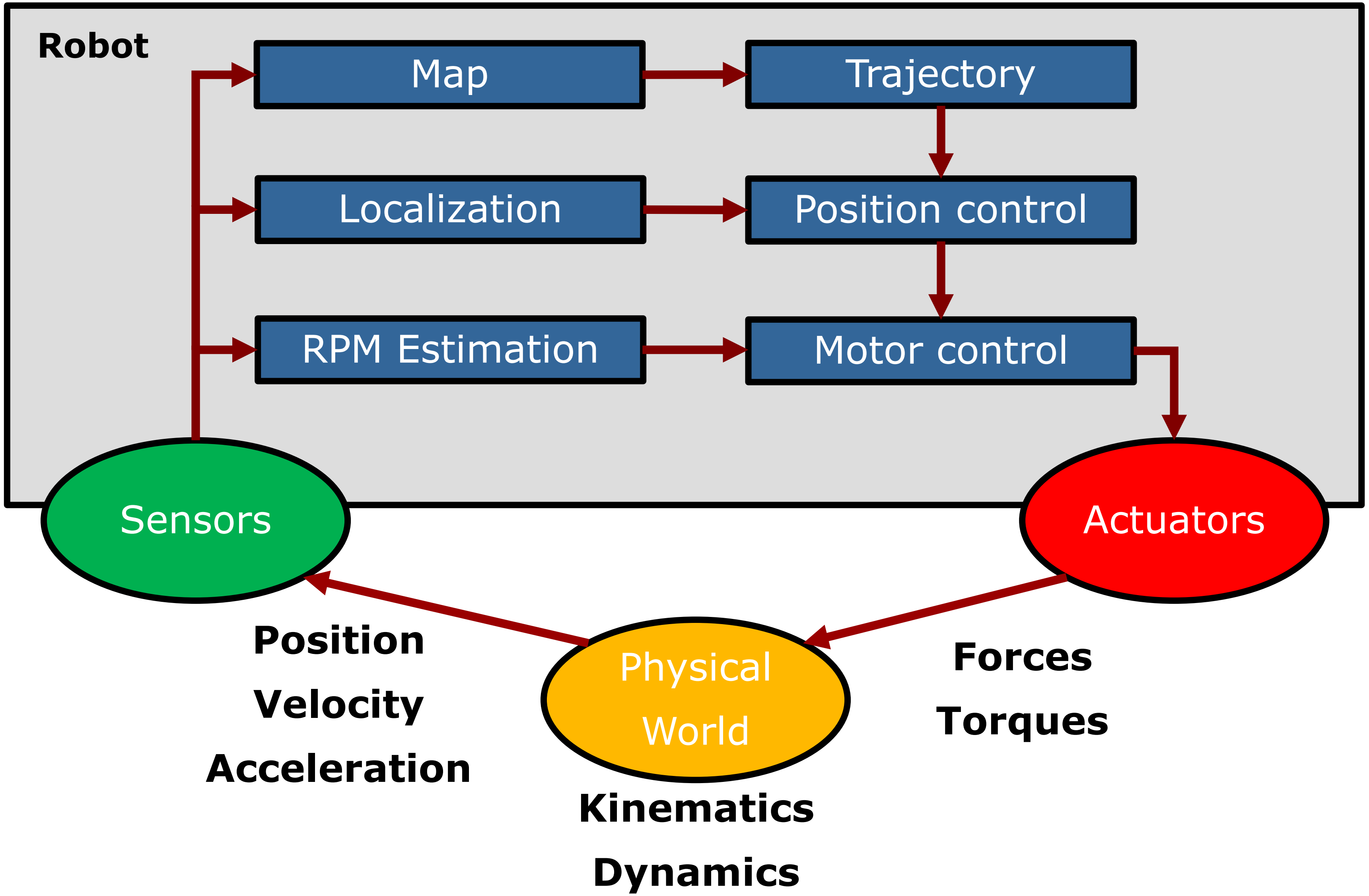
Robot Control (PID)

مهدی جوانمردی

زمستان–بهار ۱۴۰۱

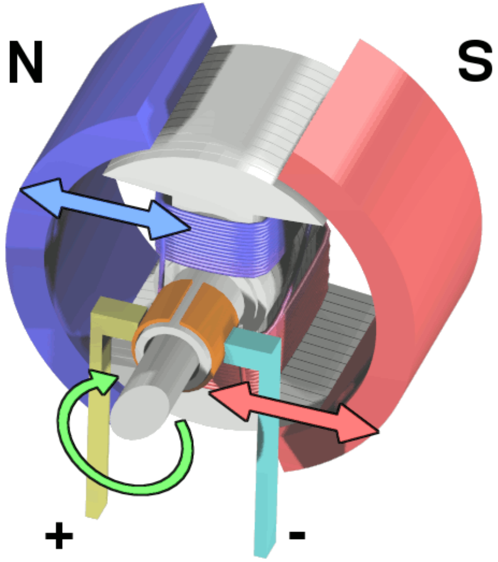
[slides adapted from Cyrill Stachniss with permission]

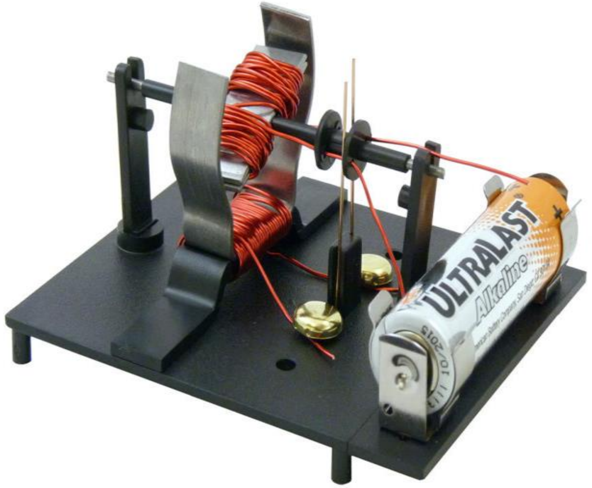
Control Architecture



DC Motors

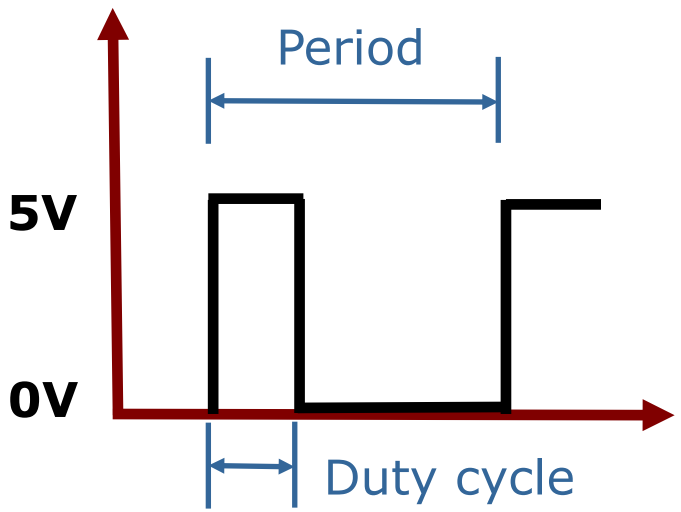
- Stationary permanent magnet
- Electromagnet induces torque
- Split rings + brushes switch the direction of current

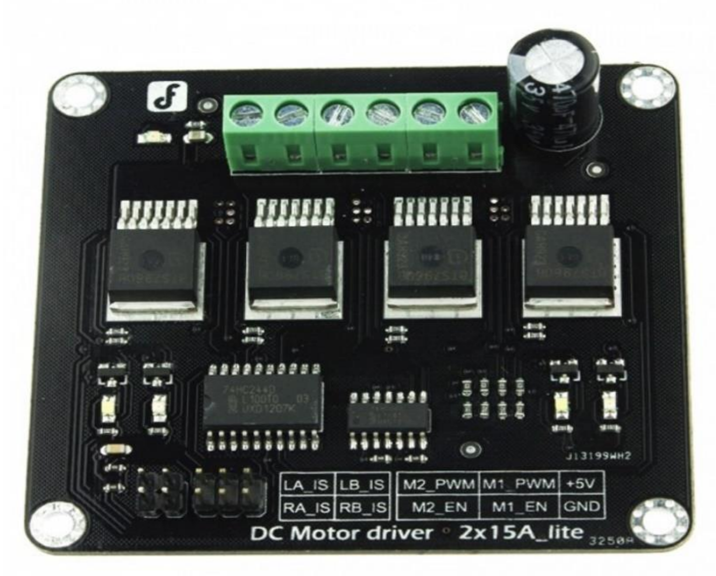




DC motors controllers

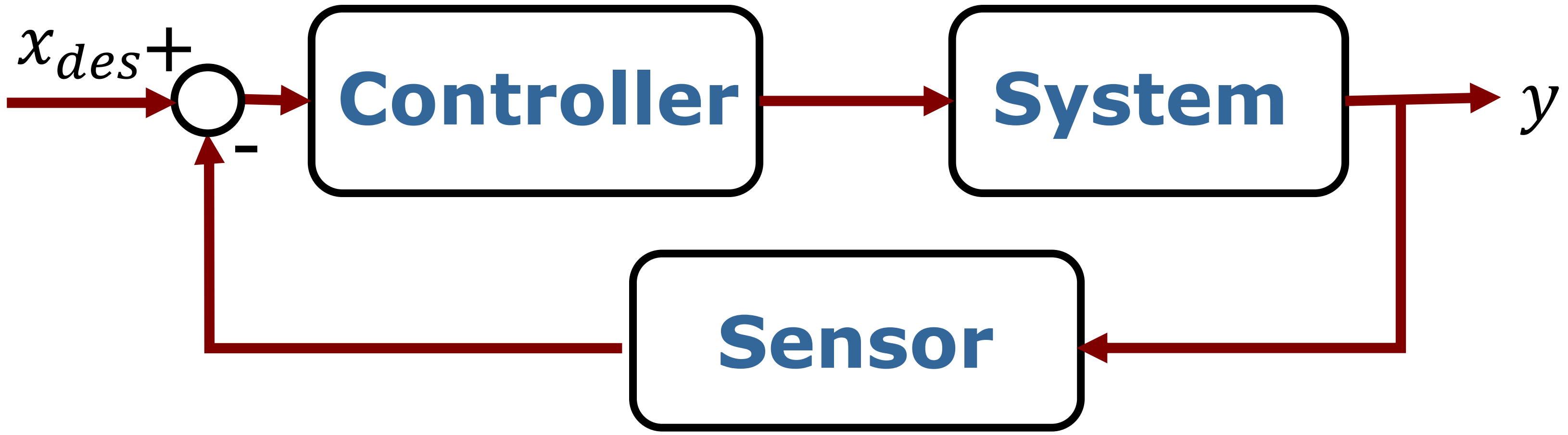
- More power = faster rotation
- How to modulate power using a digital signal?
- Pulse width modulation (PWM)
- Duty cycle = ratio of on time vs period





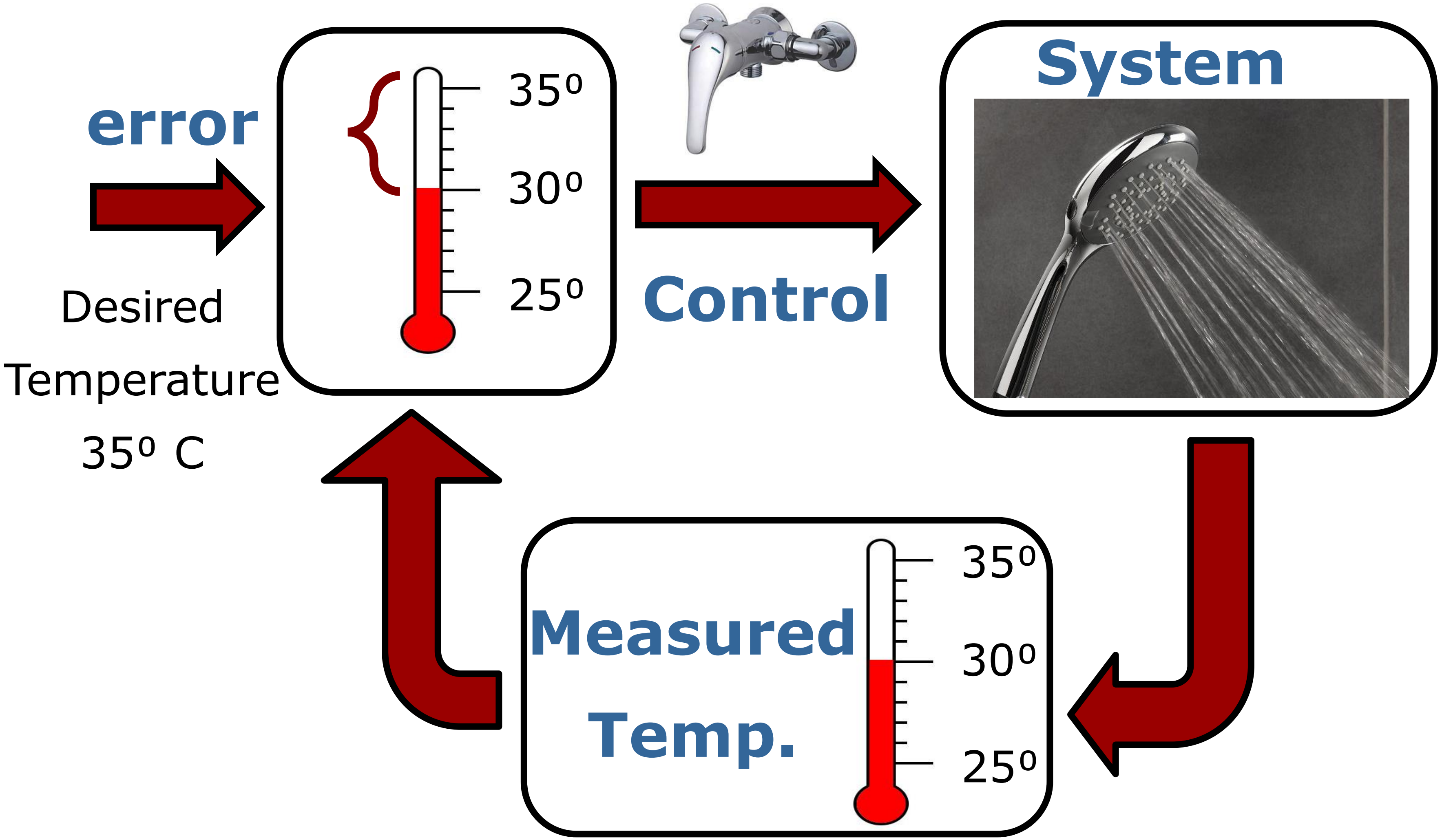
Open loop vs. feedback control

(a) open loop control

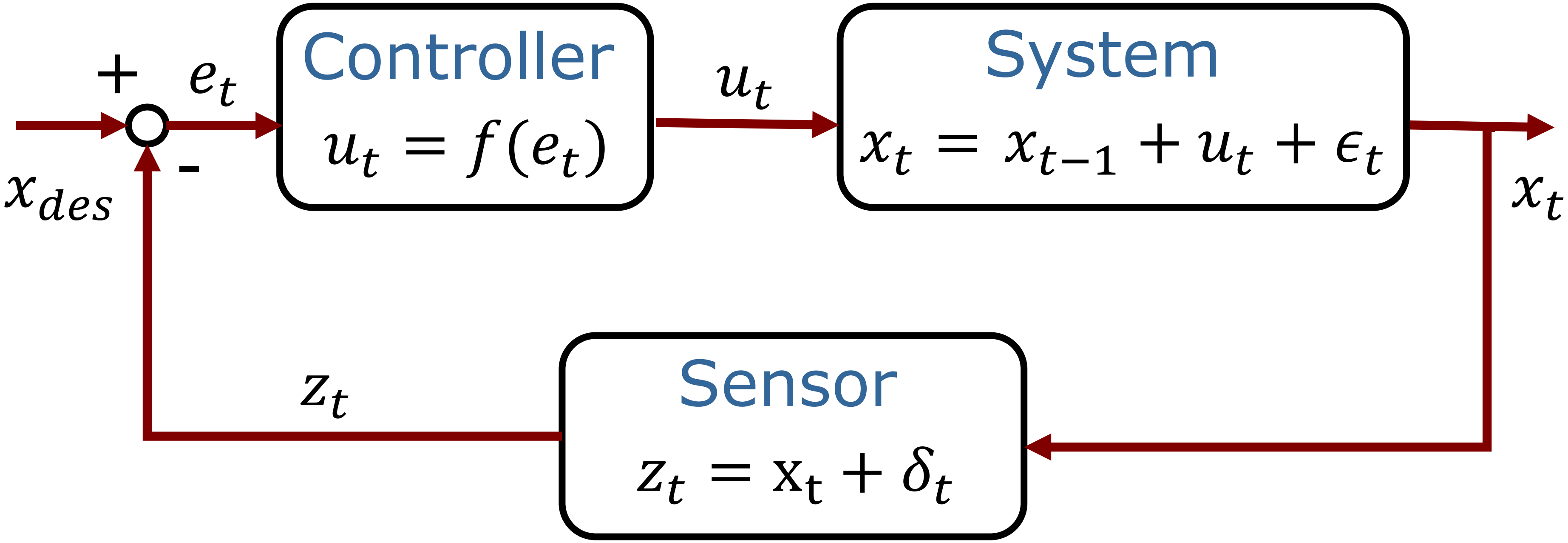


(b) feedback control

Feedback control: An Example

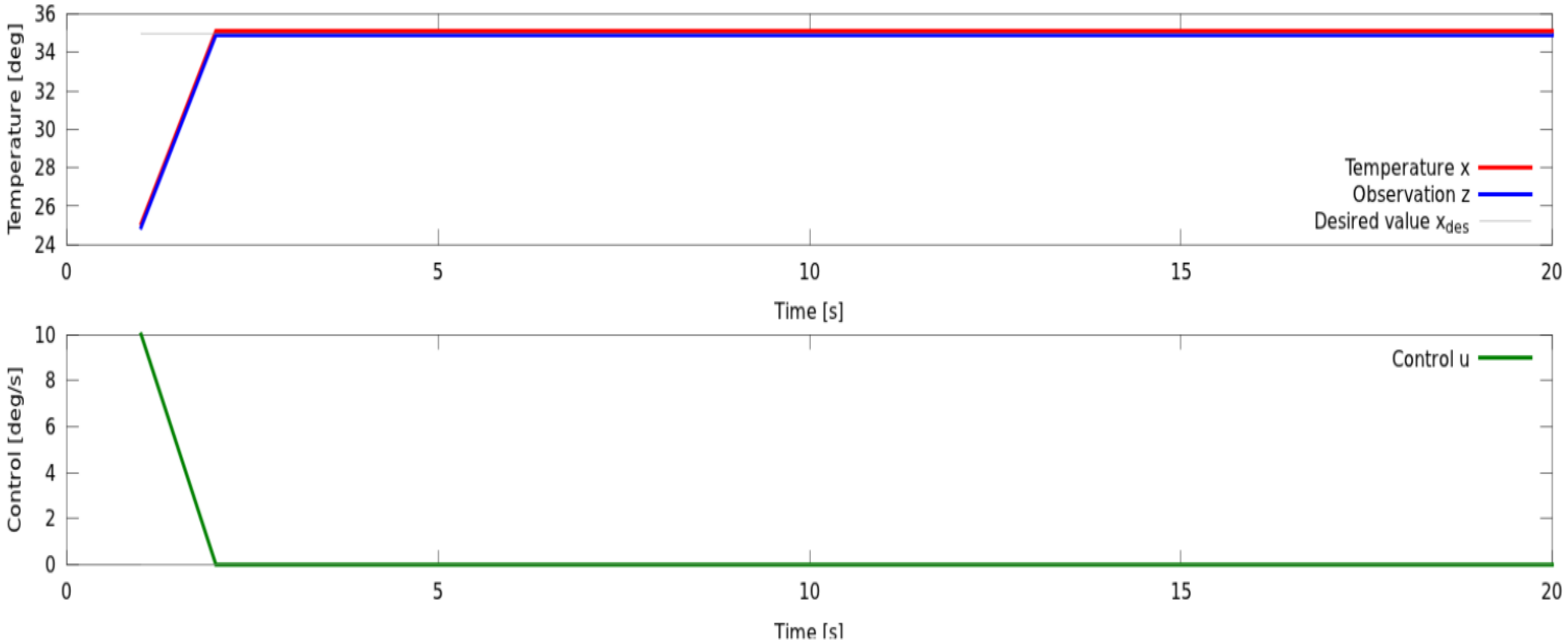


Block Diagram



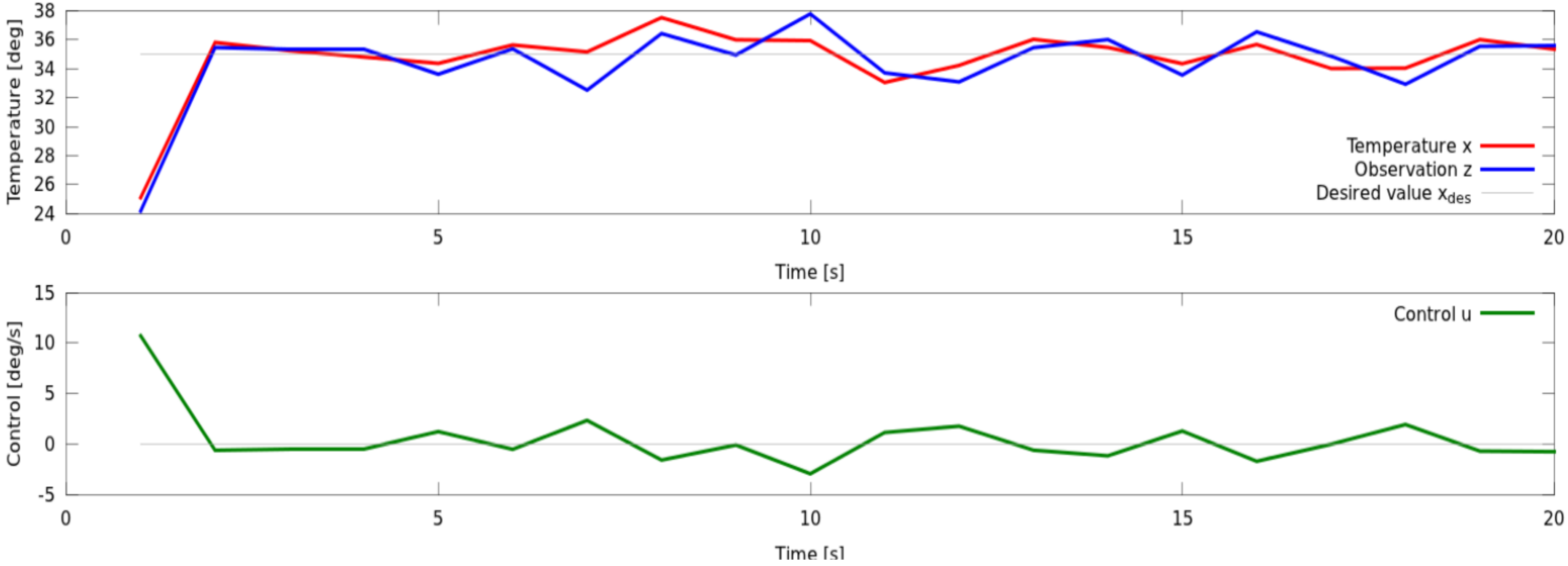
Proportional control

- P-Control: $\boldsymbol{u}_t = K_p \boldsymbol{e}_t$ with $K_p = 1$



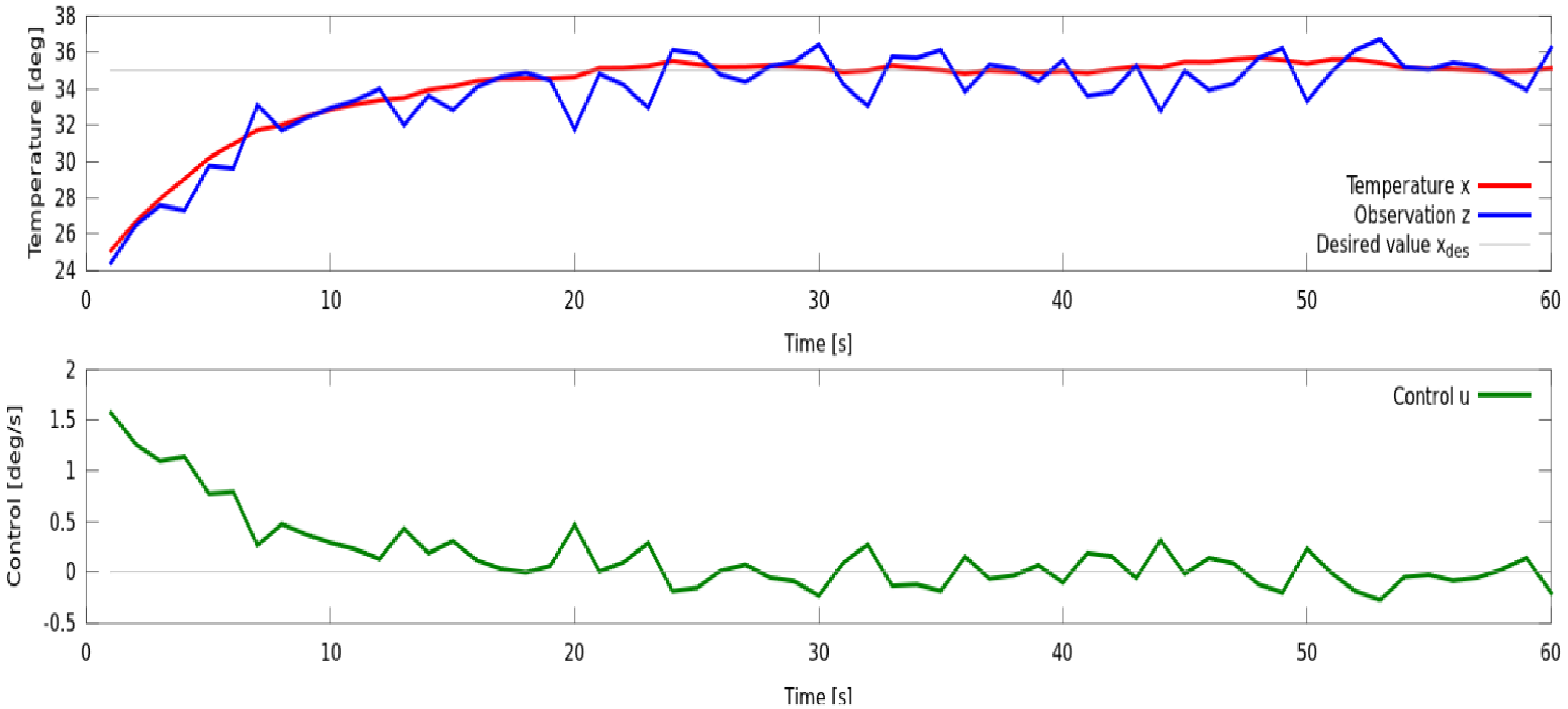
Effect of noise

- How does the noise in system/measurement effect the control?



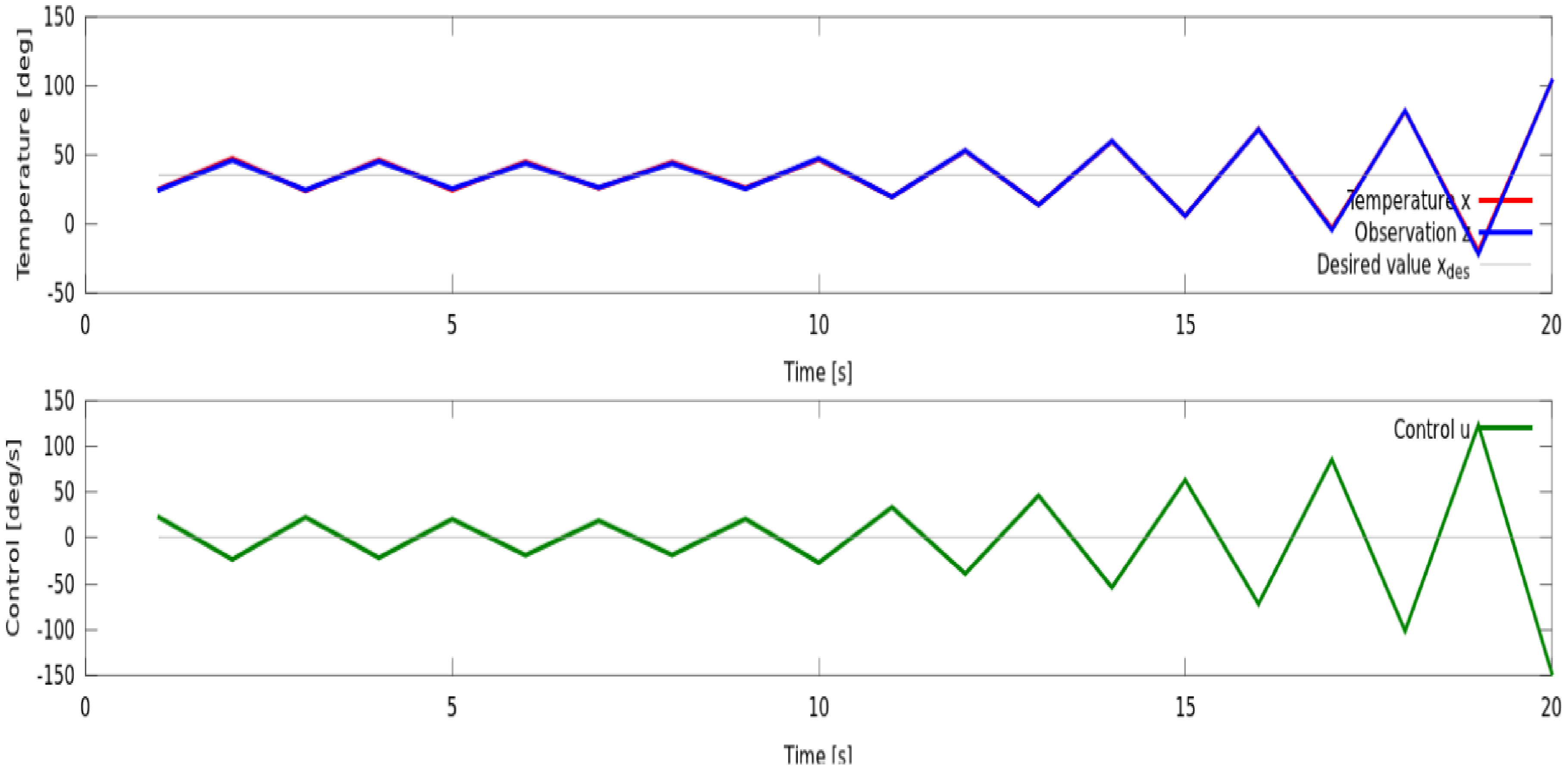
Better control with noise

- Lower gain ($K_p = 0.15$)



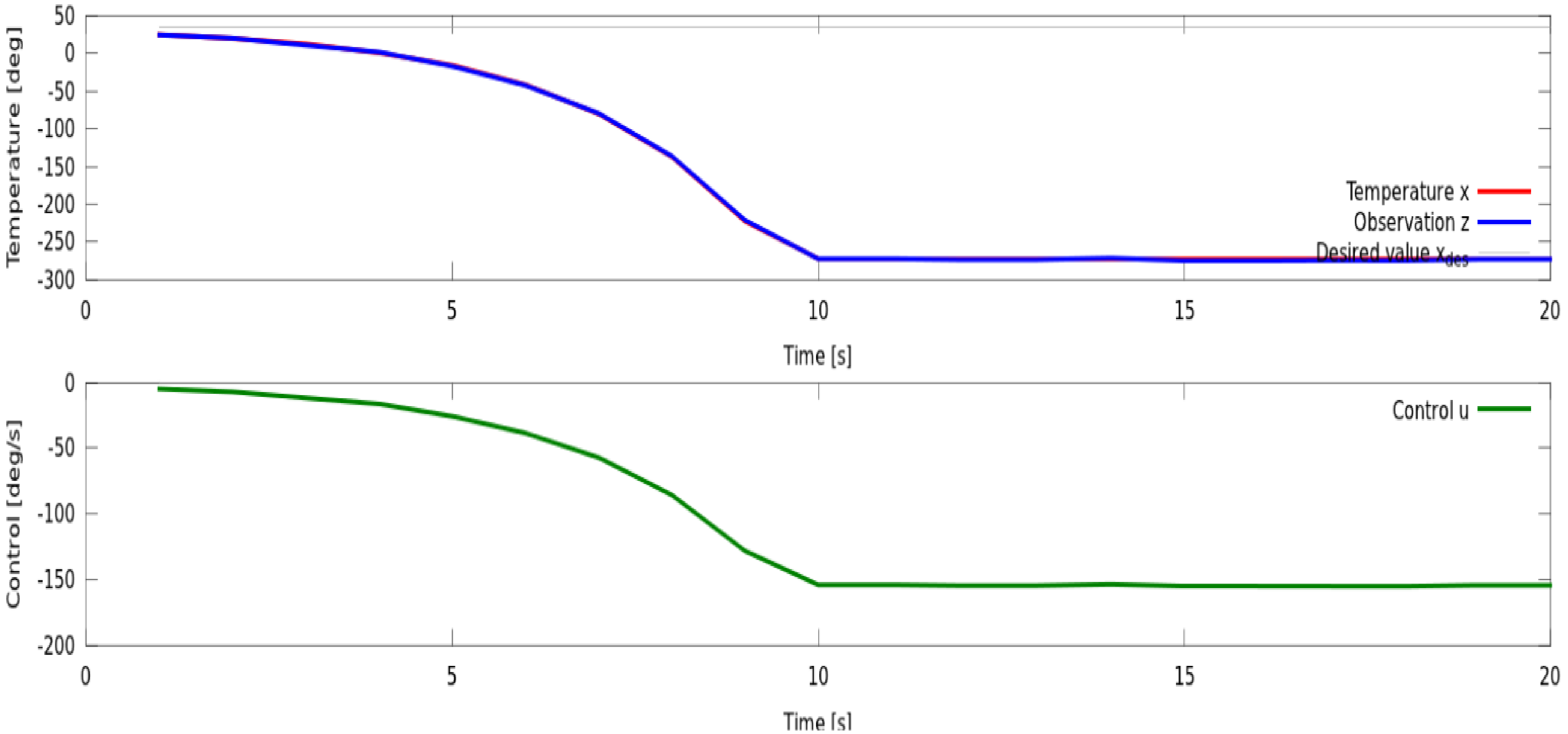
Effect of high gains

- Avoid high gains ($K_p = 2$)



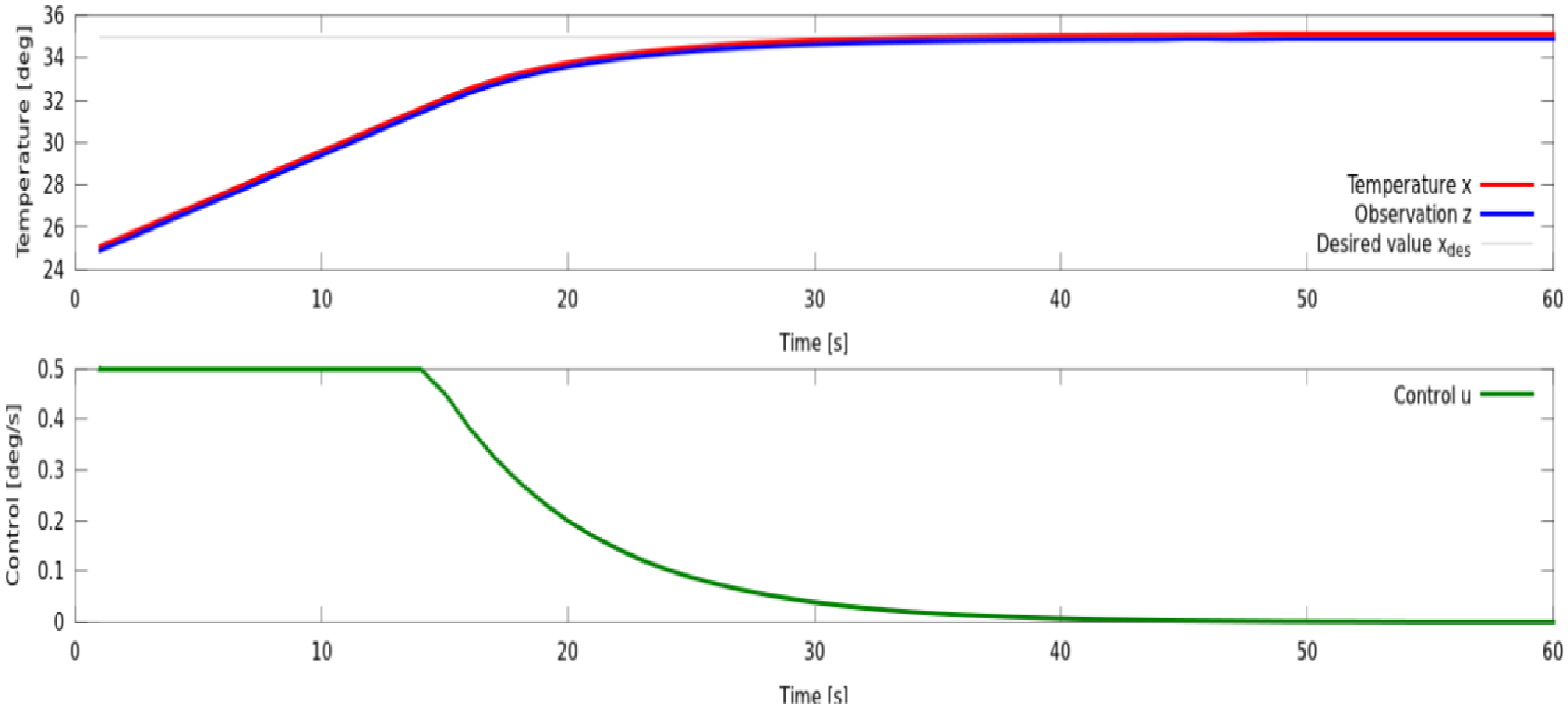
What happens with wrong sign?

- If K is negative …



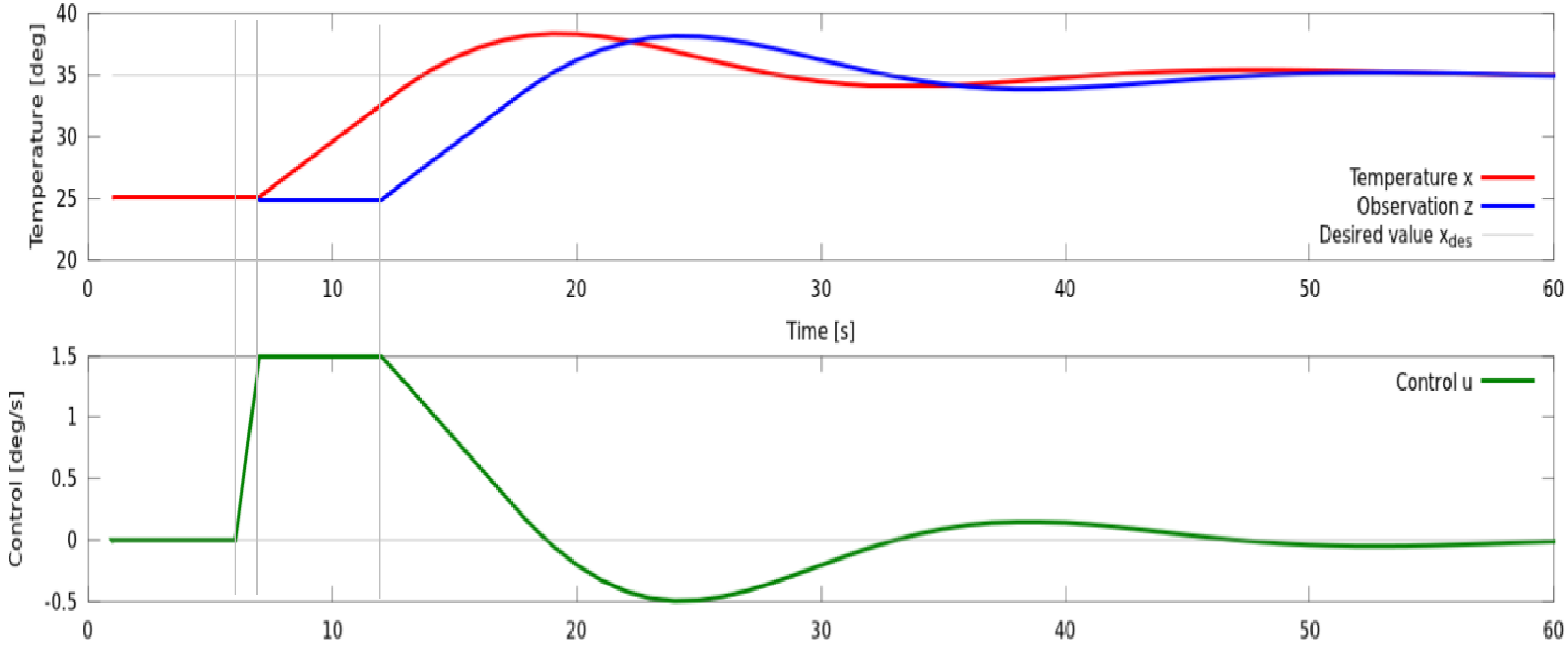
Saturation

- In practice, the set of feasible controls u is bounded



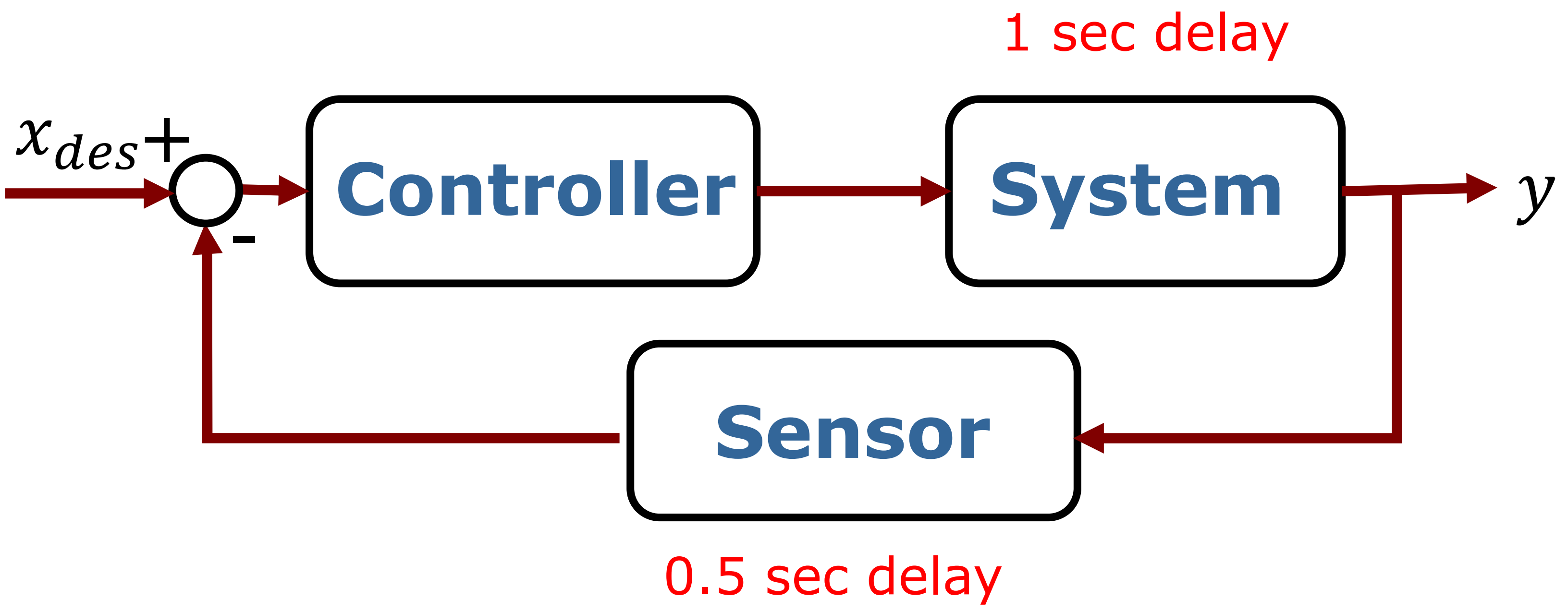
Delays

- Most real world systems have delays
- Can cause overshoots/oscillations



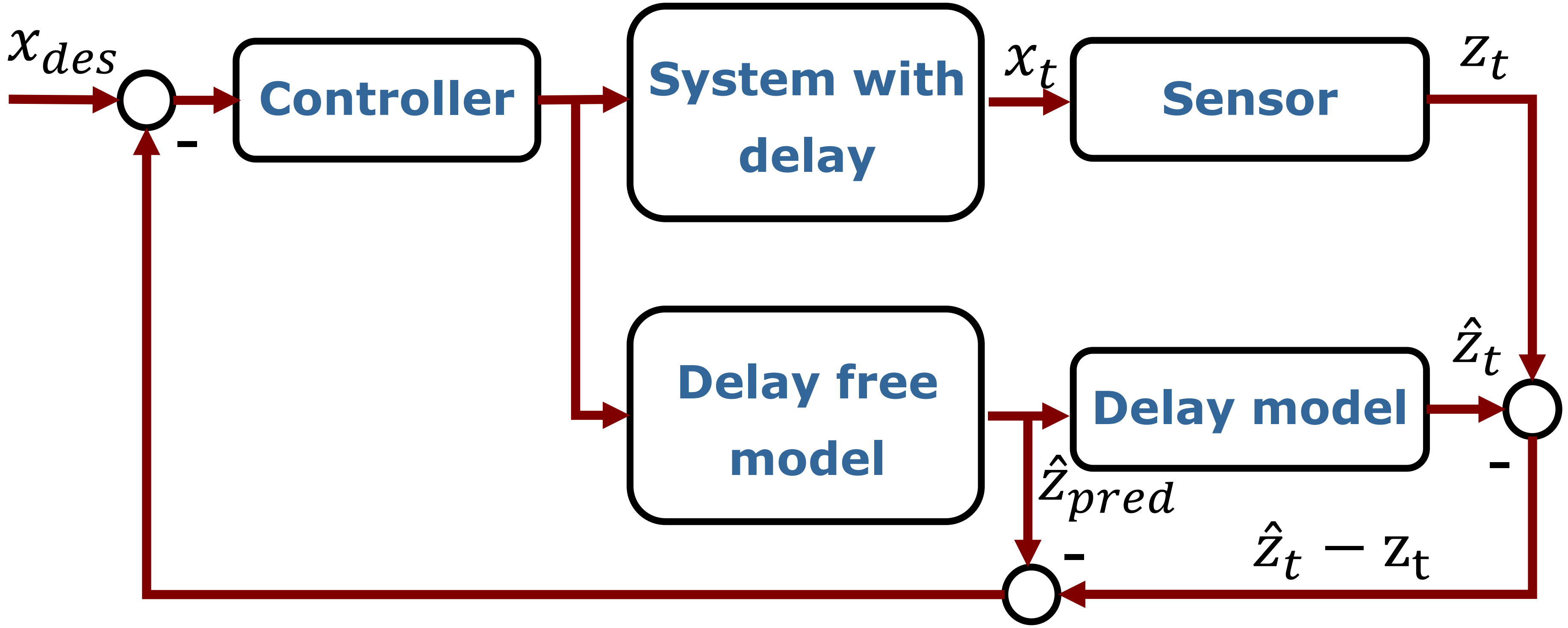
Delays

- What is the total "dead time"?
- Can we distinguish delays caused due to measurement or actuation?



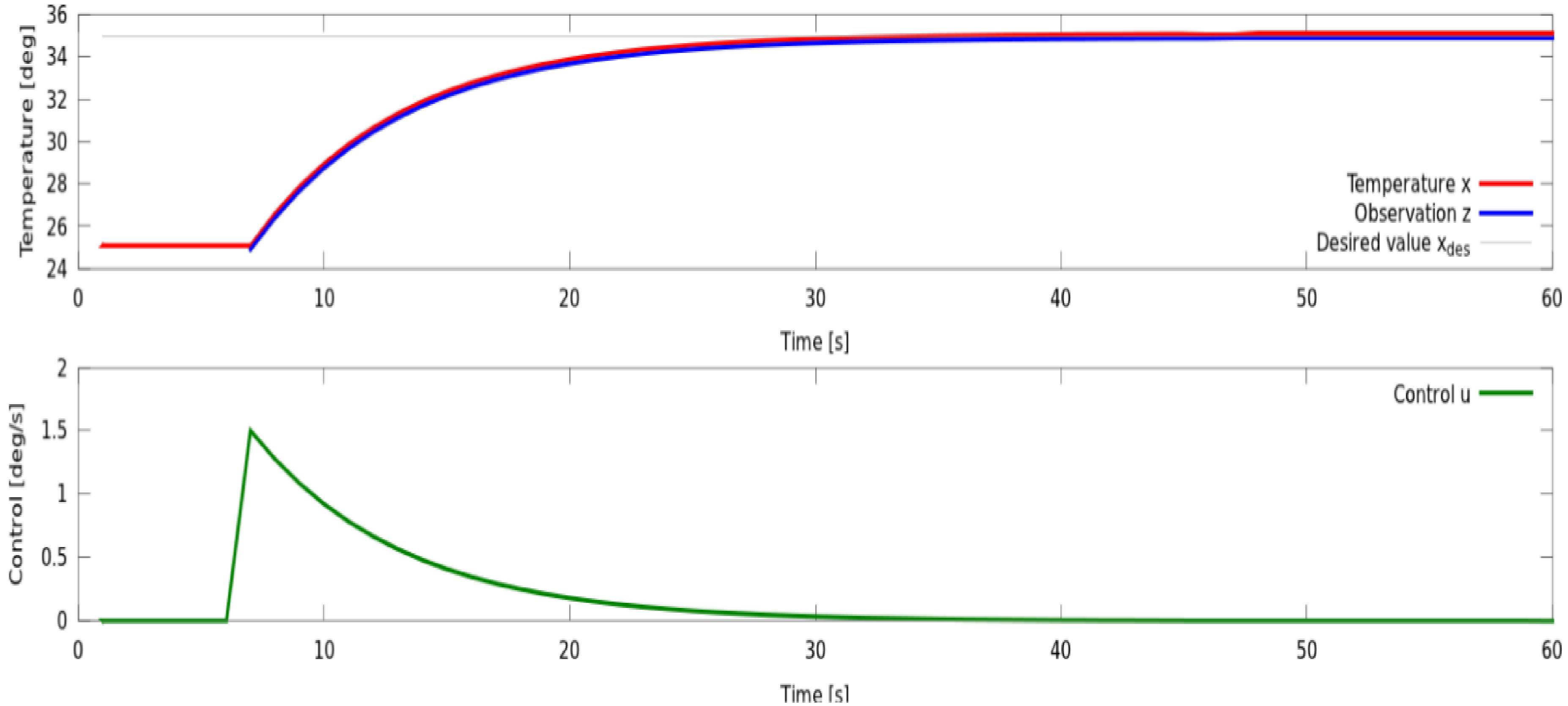
Smith Predictor

- Allows to use higher gains
- Requires accurate system model



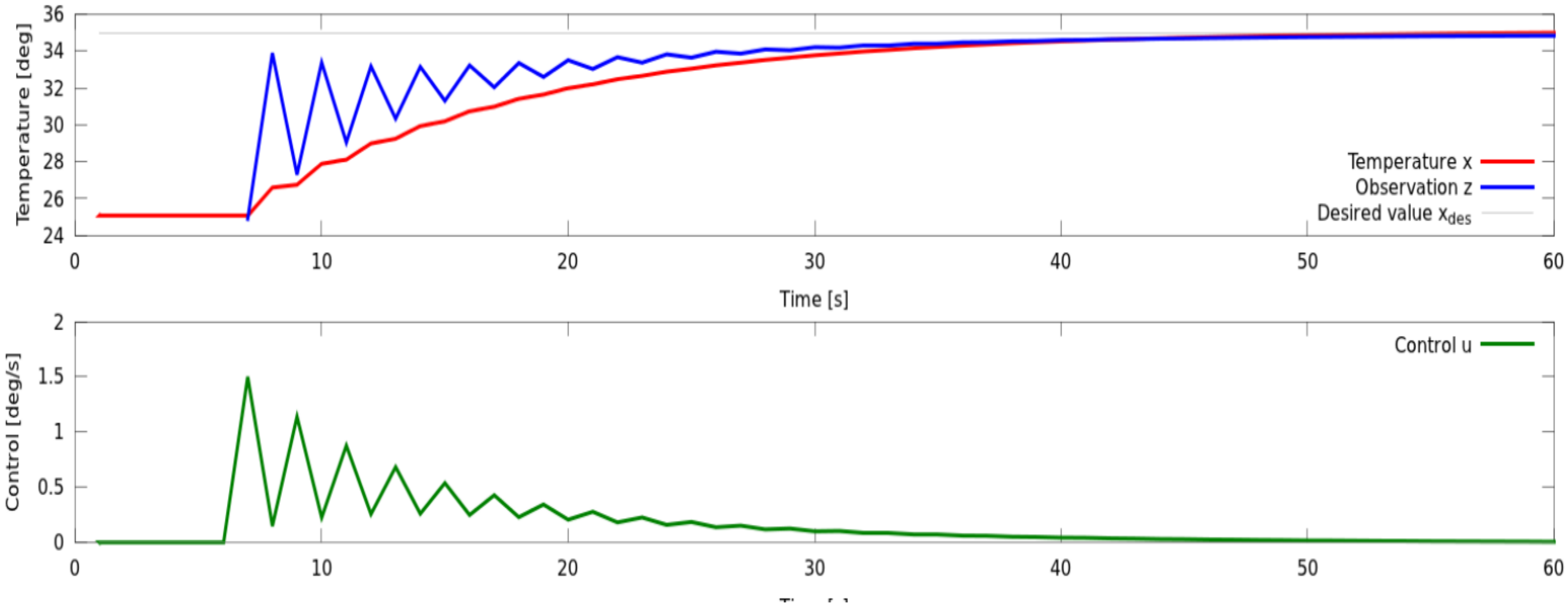
Smith Predictor

- Assume: Known system & delay model
- Results in perfect compensation



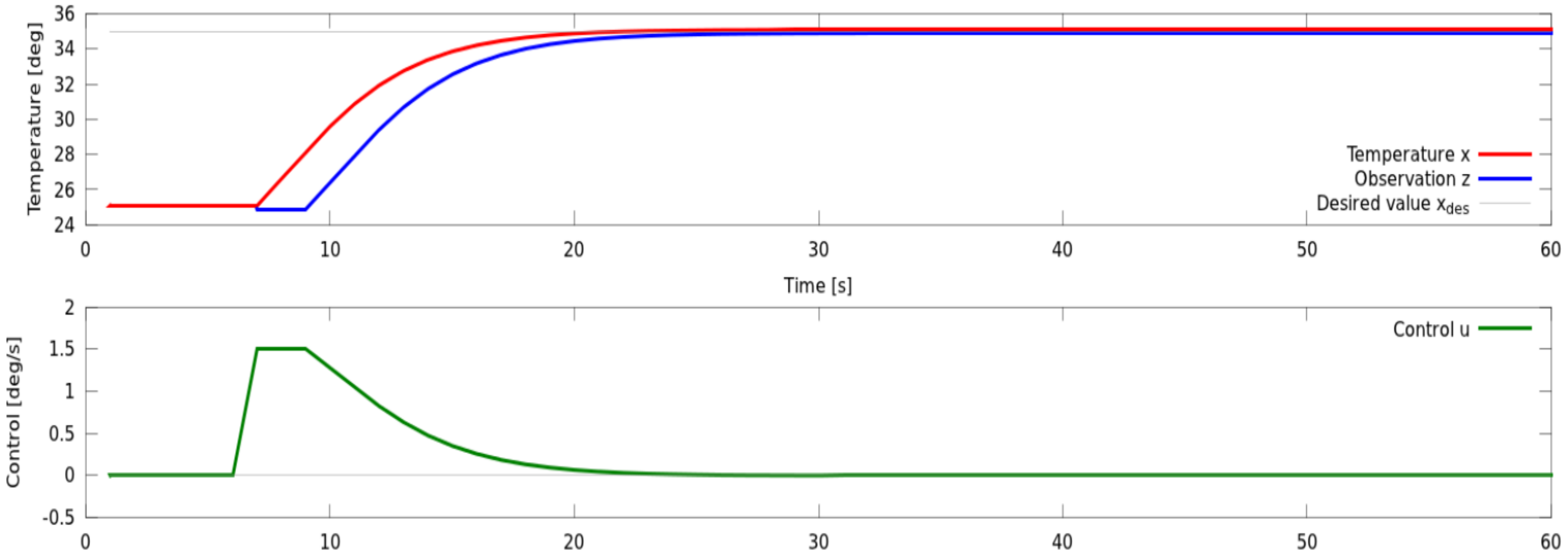
Smith Predictor

- Delay and plant model is not known accurately or is time-varying
- What if delay is **overestimated?**



Smith Predictor

- Delay and plant model is not known accurately or is time-varying
- What if delay is **underestimated?**

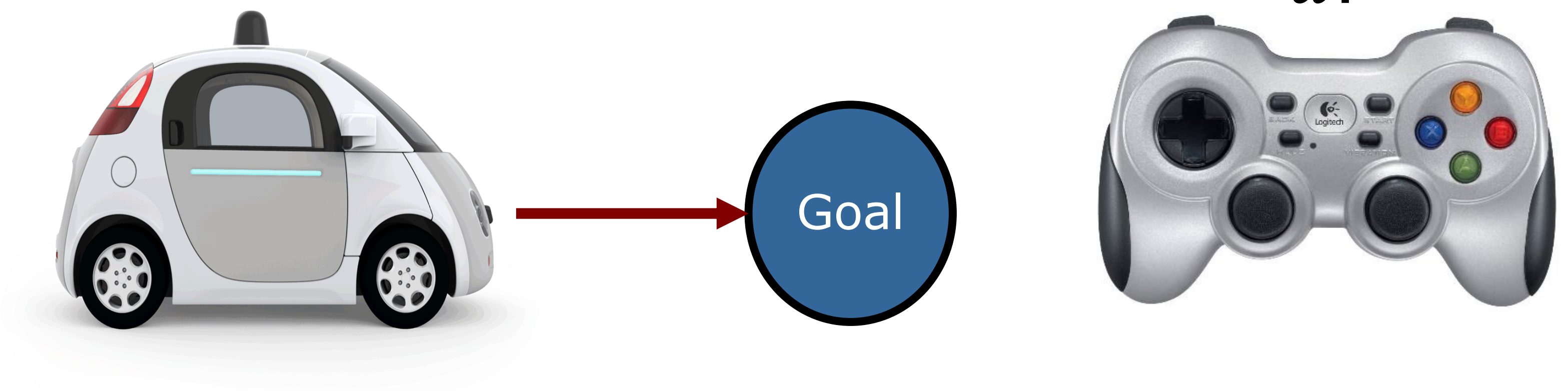


Position Control

Motivation: Position control

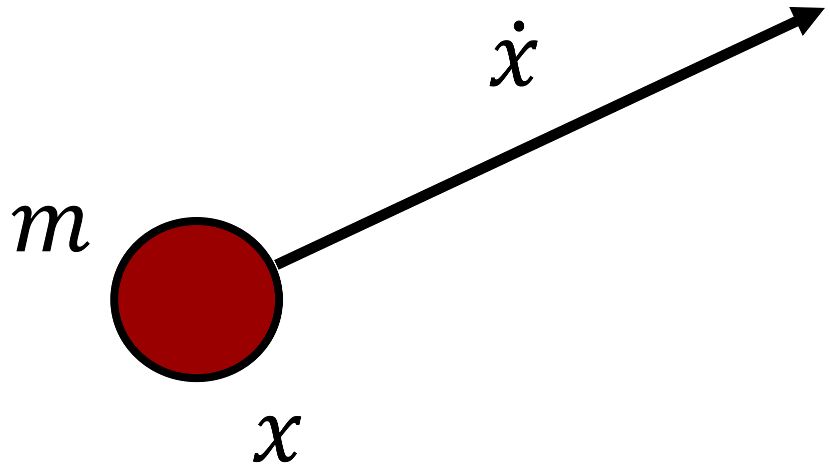
- Move the robot to the desired goal location x_{des}
- How to generate the suitable control signal u ?
- Robot location estimated via sensor measurements z

$u?$



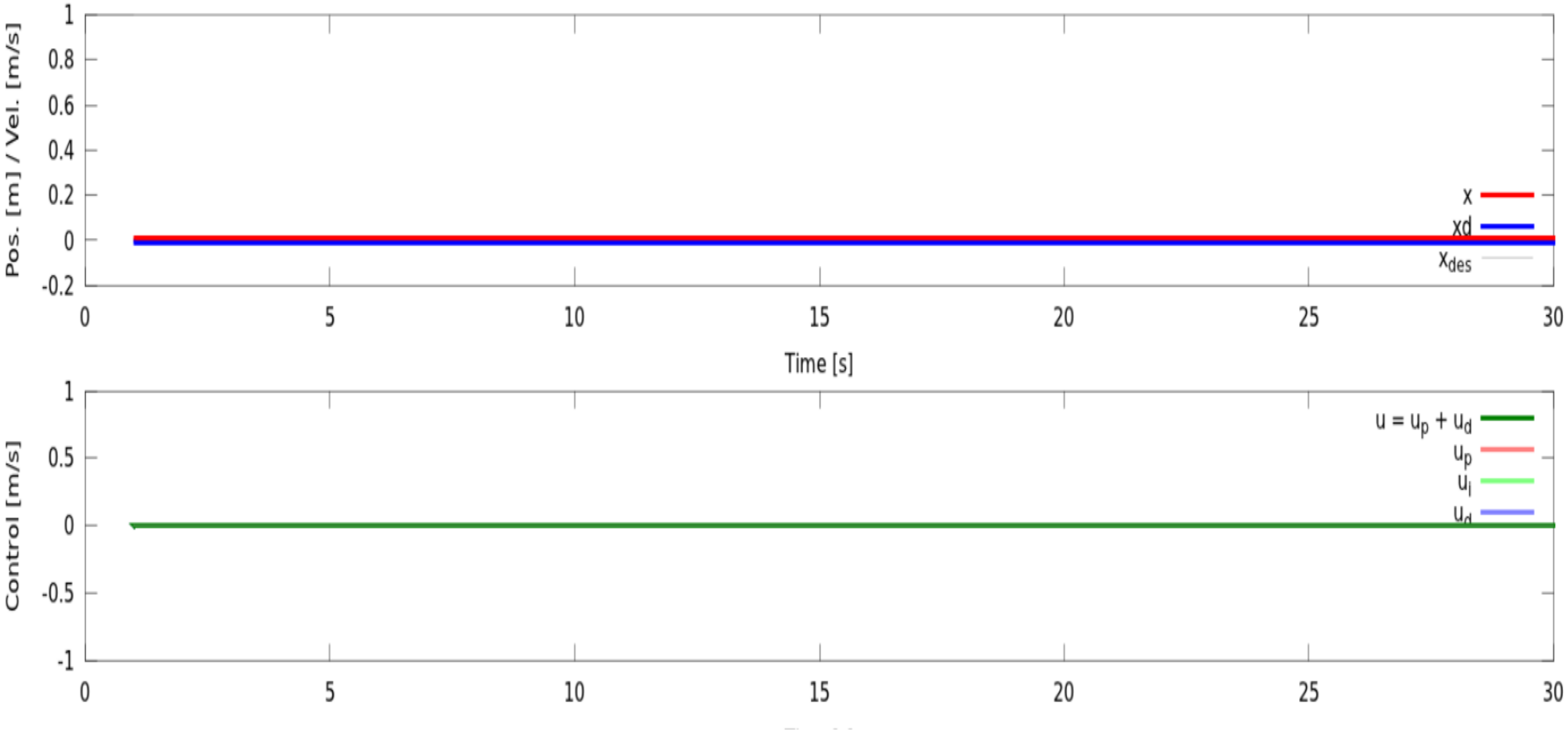
Kinematics of a rigid body

- Consider the robot as a point mass
- Moving freely in 1D space



Kinematics of a rigid body

- System model : $$x_t = x_{t-1} + \dot{x}\Delta t$$
- Initial state: $$x_0 = 0, \dot{x}_0 = 0$$



Position Control Task

- Position control task is to reach the desired position $x_{des} = 1$ and stop there

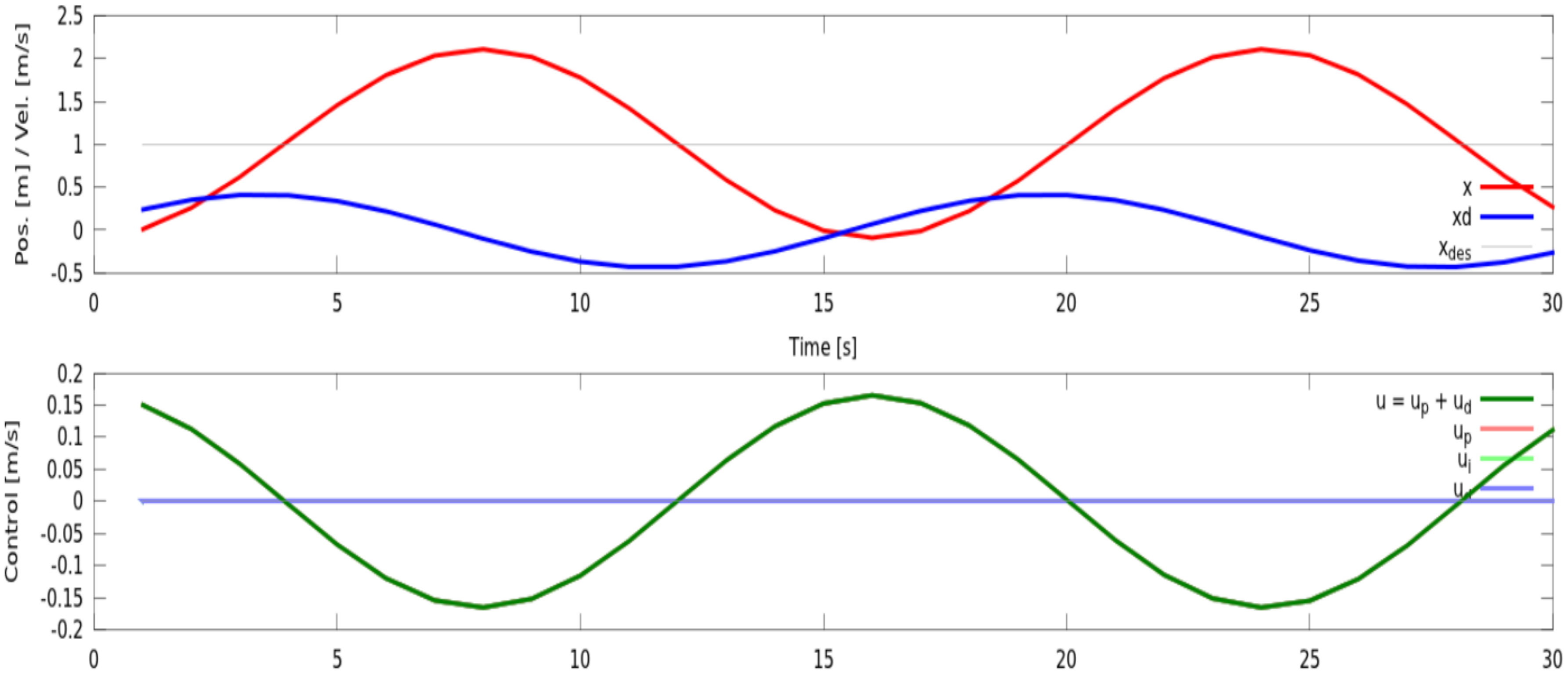
- At each time instant, we apply a control u_t

- What will happen with P control?

P Control

- Proportional control law

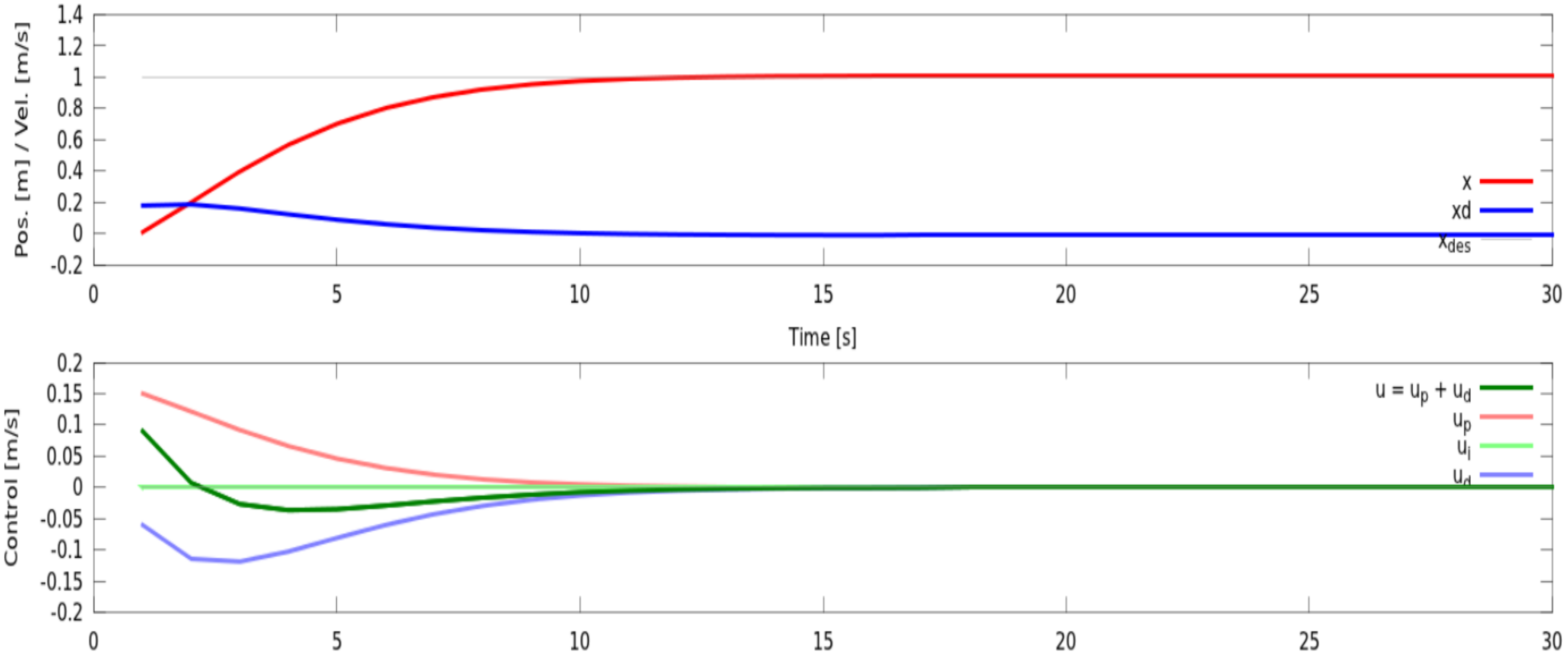
$$u_t = K_p(x_{des} - x_t)$$



PD Control

- Proportional-derivative control law

$$u_t = K_p(x_{des} - x_t) + K_D(\dot{x}_{des} - \dot{x}_t)$$

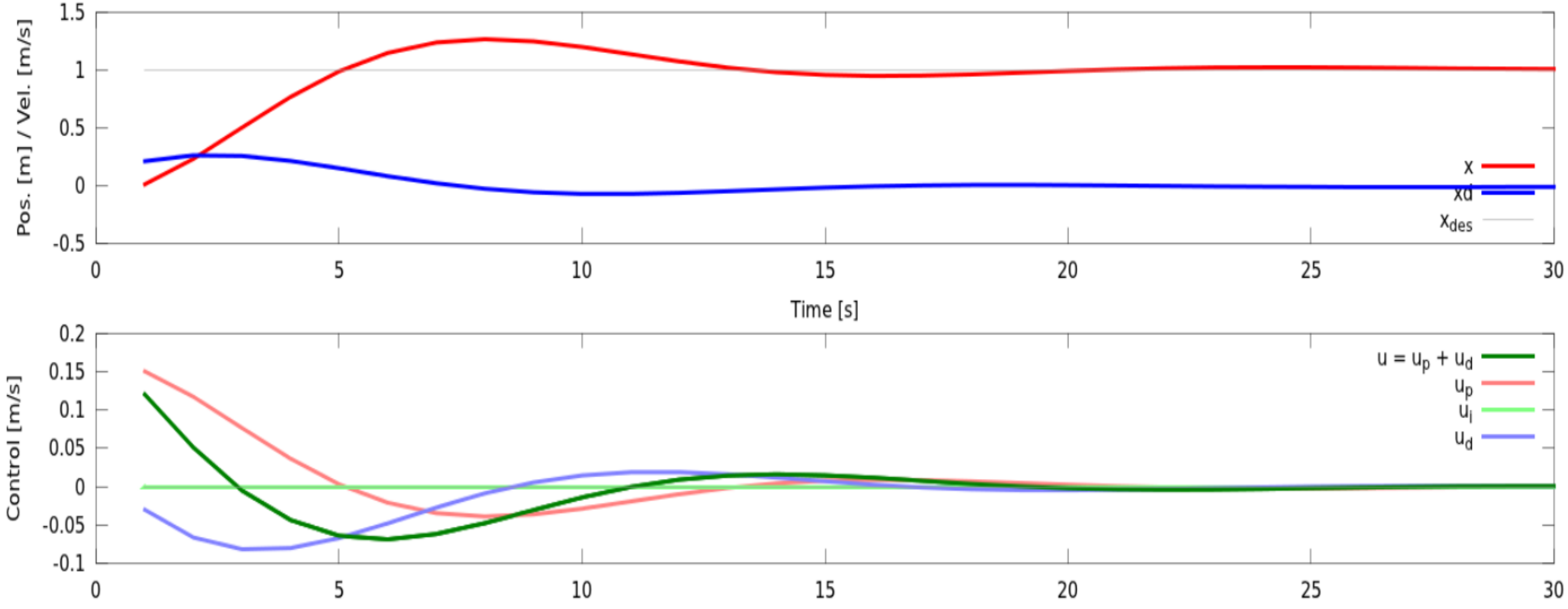


PD Control

- Proportional-derivative control law

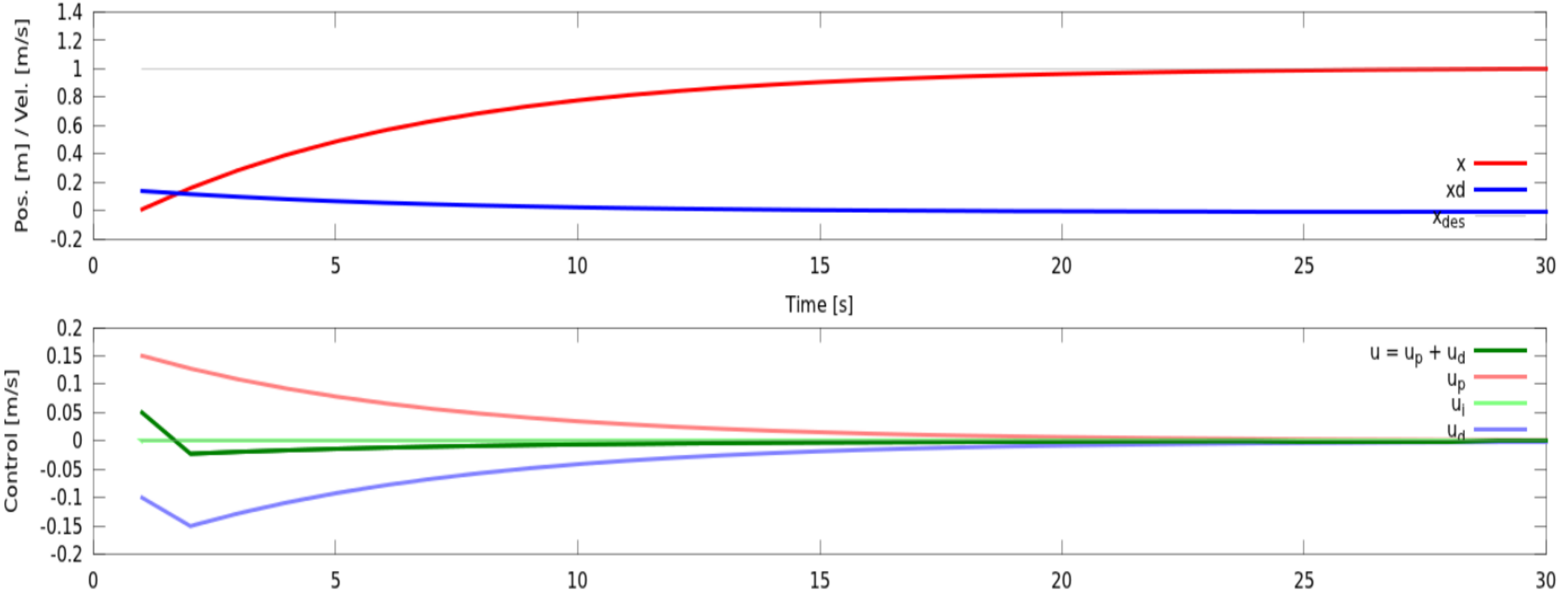
  $$u_t = K_p(x_{des} - x_t) + K_D(\dot{x}_{des} - \dot{x}_t)$$

- What happens with low gains?



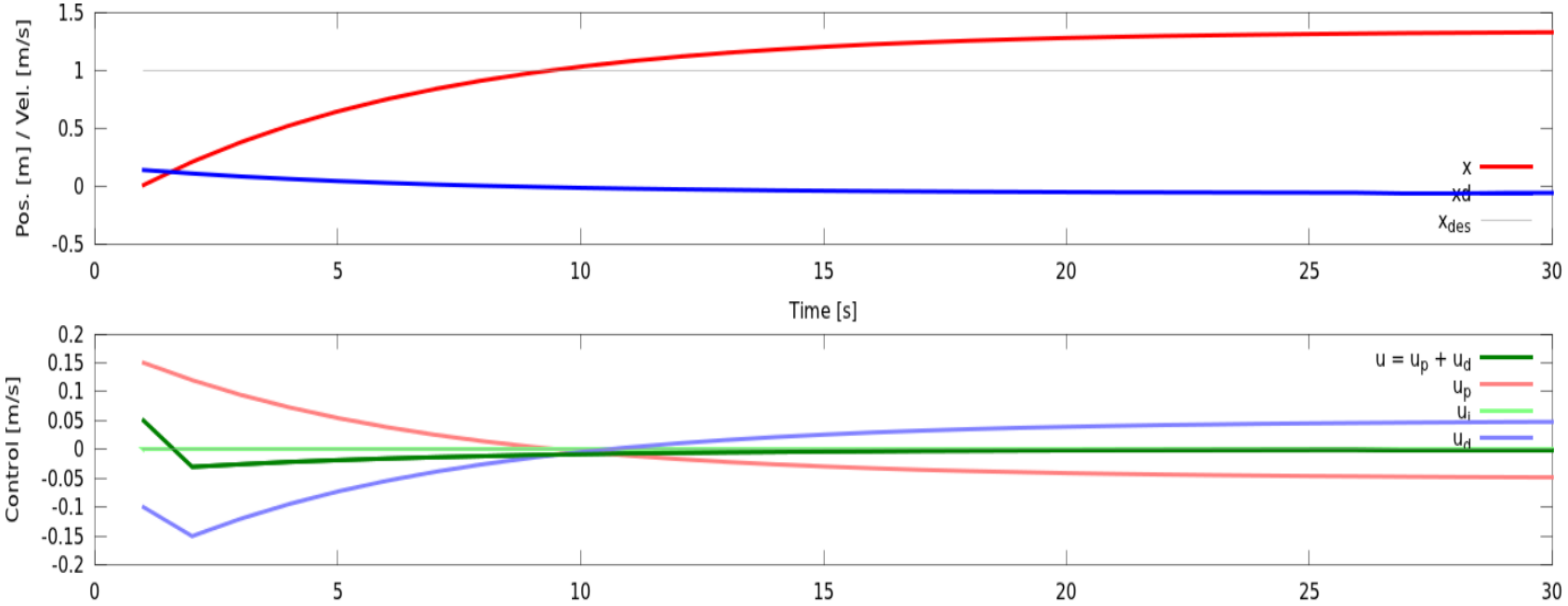
PD Control

- Proportional-derivative control law
  $$u_t = K_p(x_{des} - x_t) + K_D(\dot{x}_{des} - \dot{x}_t)$$
- What happens with high gains?



PD Control

- What happens when there is a systematic bias?
- Ex: gravity is not considered in our drone model



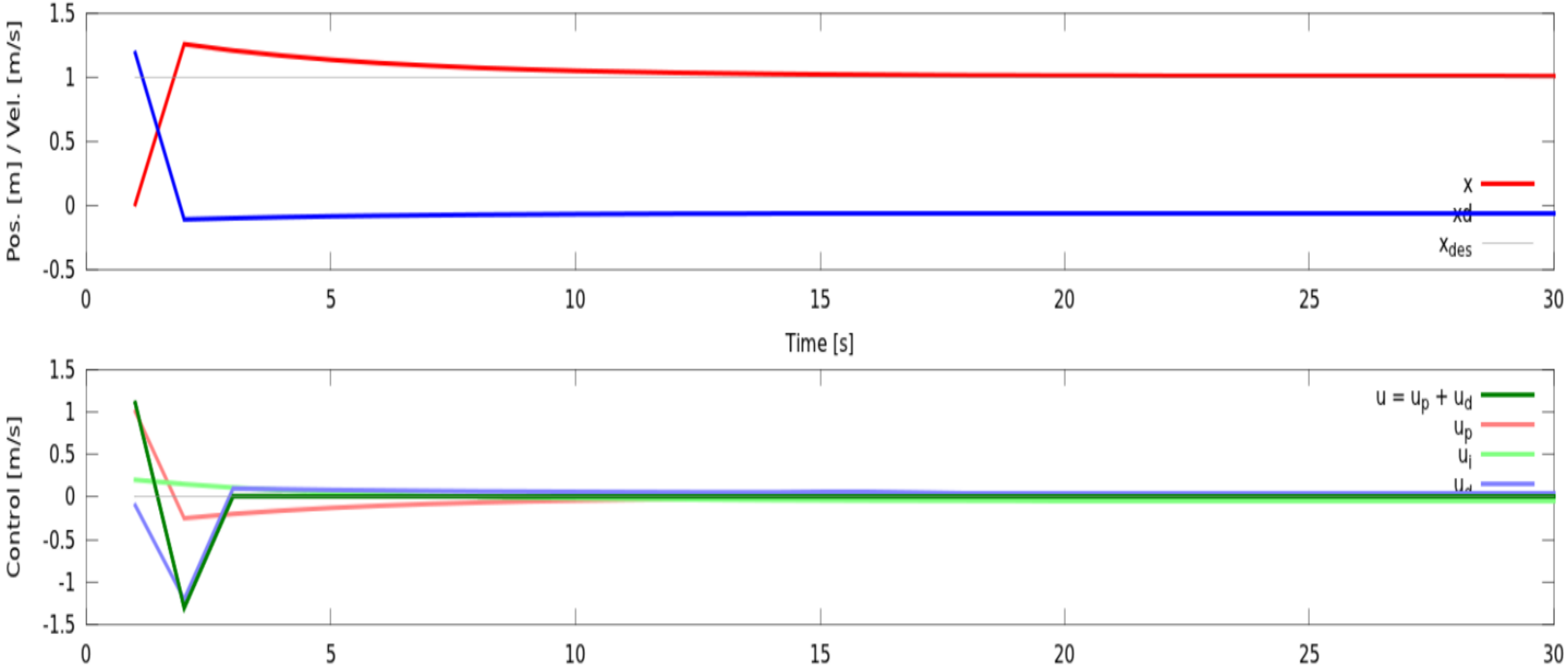
PID Control

- **Idea:** Estimate the systematic error …
    - Also known as steady-state error

$$u_t = K_p(x_{des} - x_t) + K_D(\dot{x}_{des} - \dot{x}_t)$$
$$+K_I \int_0^t (x_{des} - x_t)dt$$

PID Controller

- **Idea:** Estimate the systematic error …



PID Controller

- **Idea:** Estimate the systematic error …

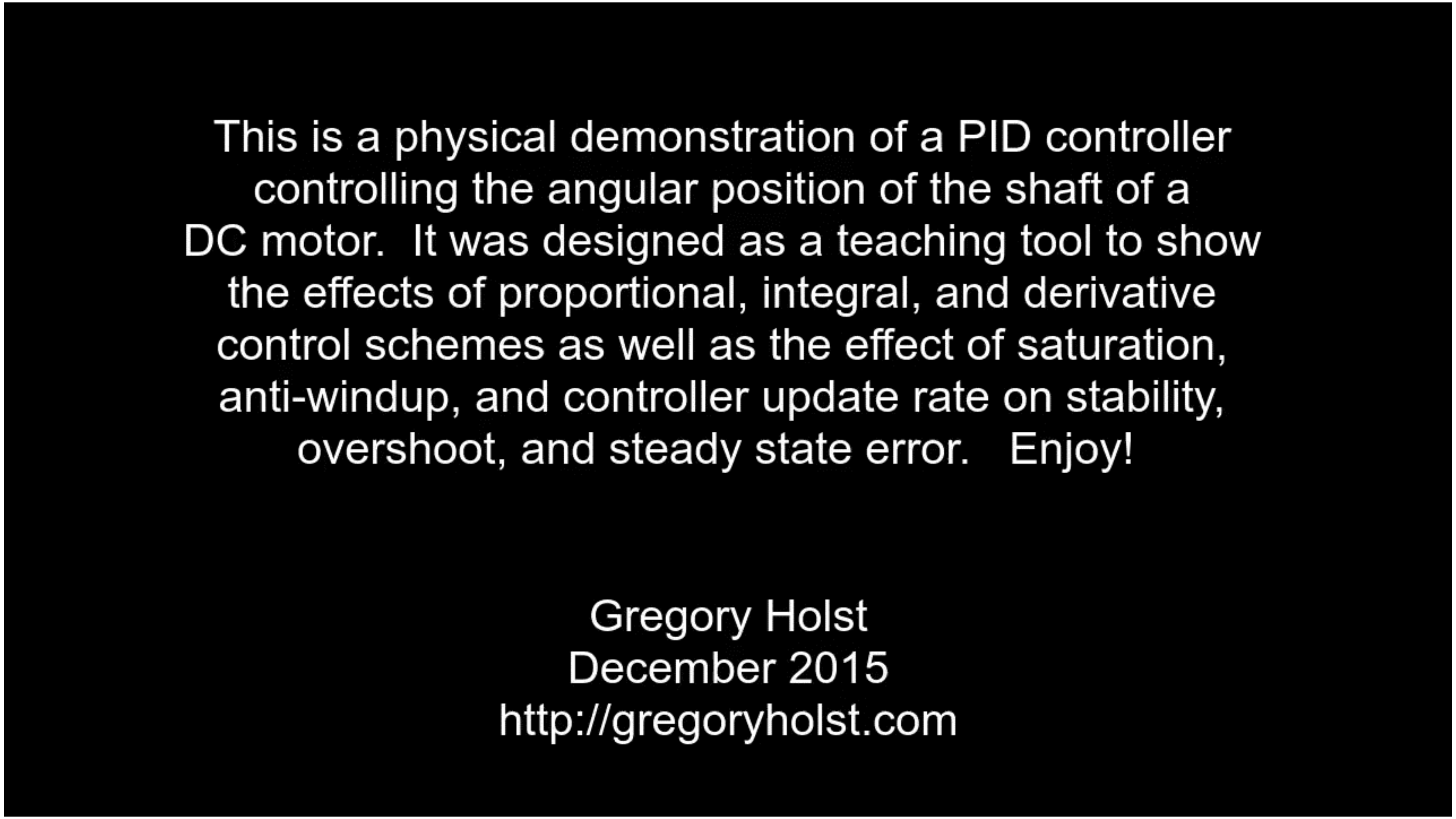
$$u_t = K_p(x_{des} - x_t) + K_D(\dot{x}_{des} - \dot{x}_t) + K_I \int_0^t (x_{des} - x_t)dt$$

- Reasonable for steady state system
- May be dangerous to error build up (wind-up effect)

PID Control - Summary

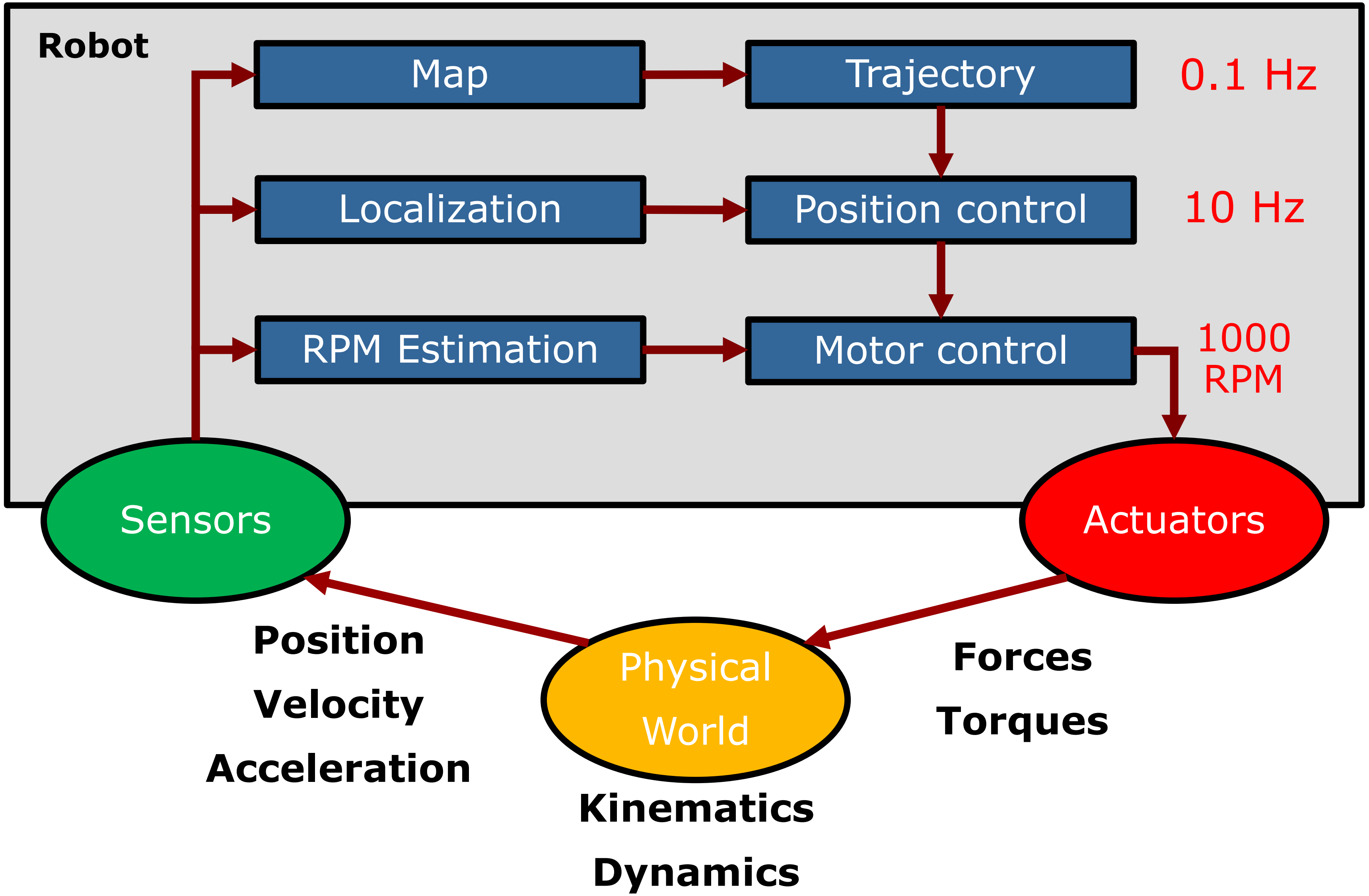
- P = simple proportional control, sufficient in most cases.
- PD = reduce overshoot (e.g. when acceleration can be controlled)
- PI = compensate for systematic error/bias
- PID = combination of the above properties.

PID Controller Demo

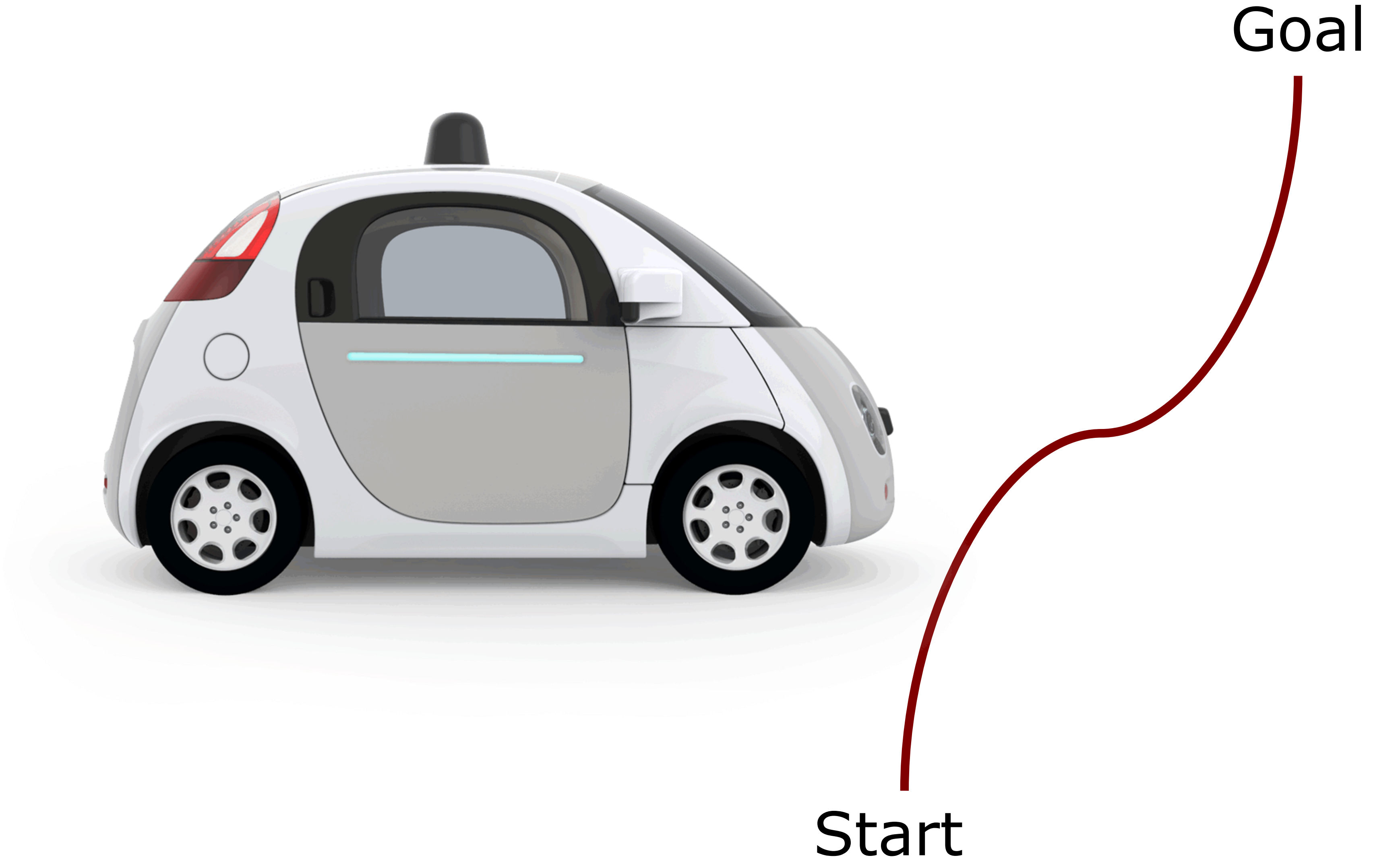


[Video Courtesy: Gregory Holst]

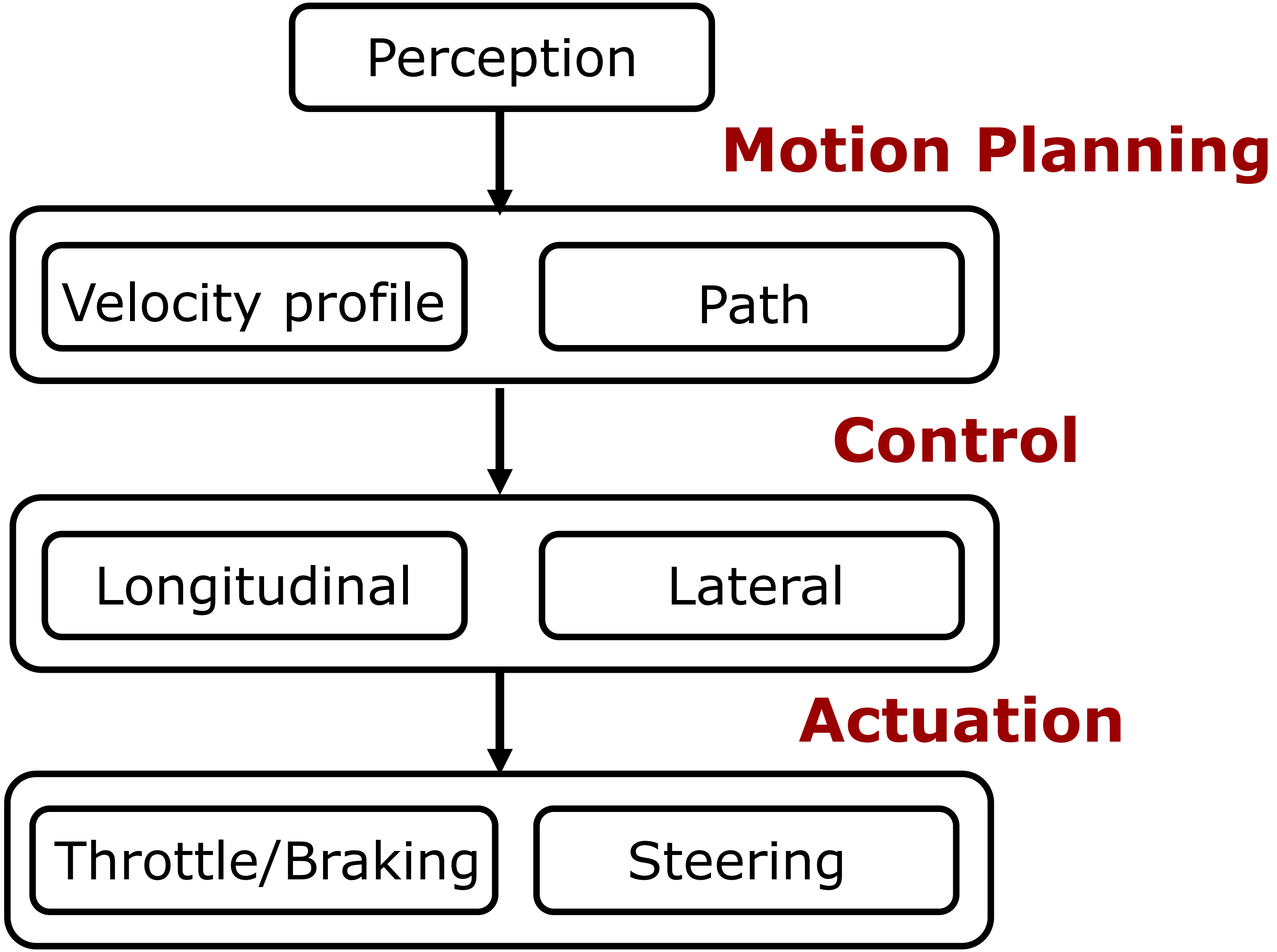
Cascaded control



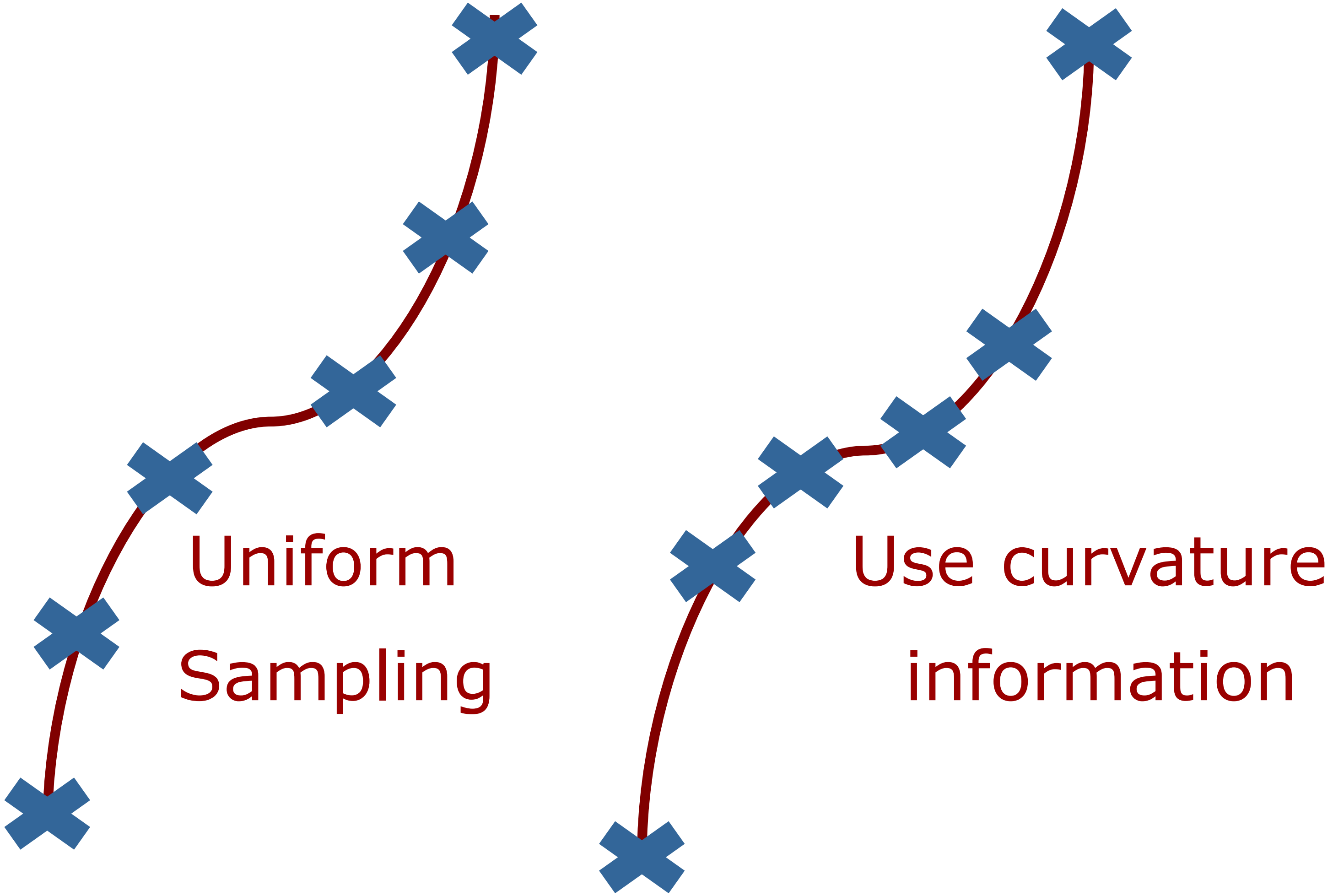
Application: Tracking Trajectory



Control Architecture



Trajectory generation

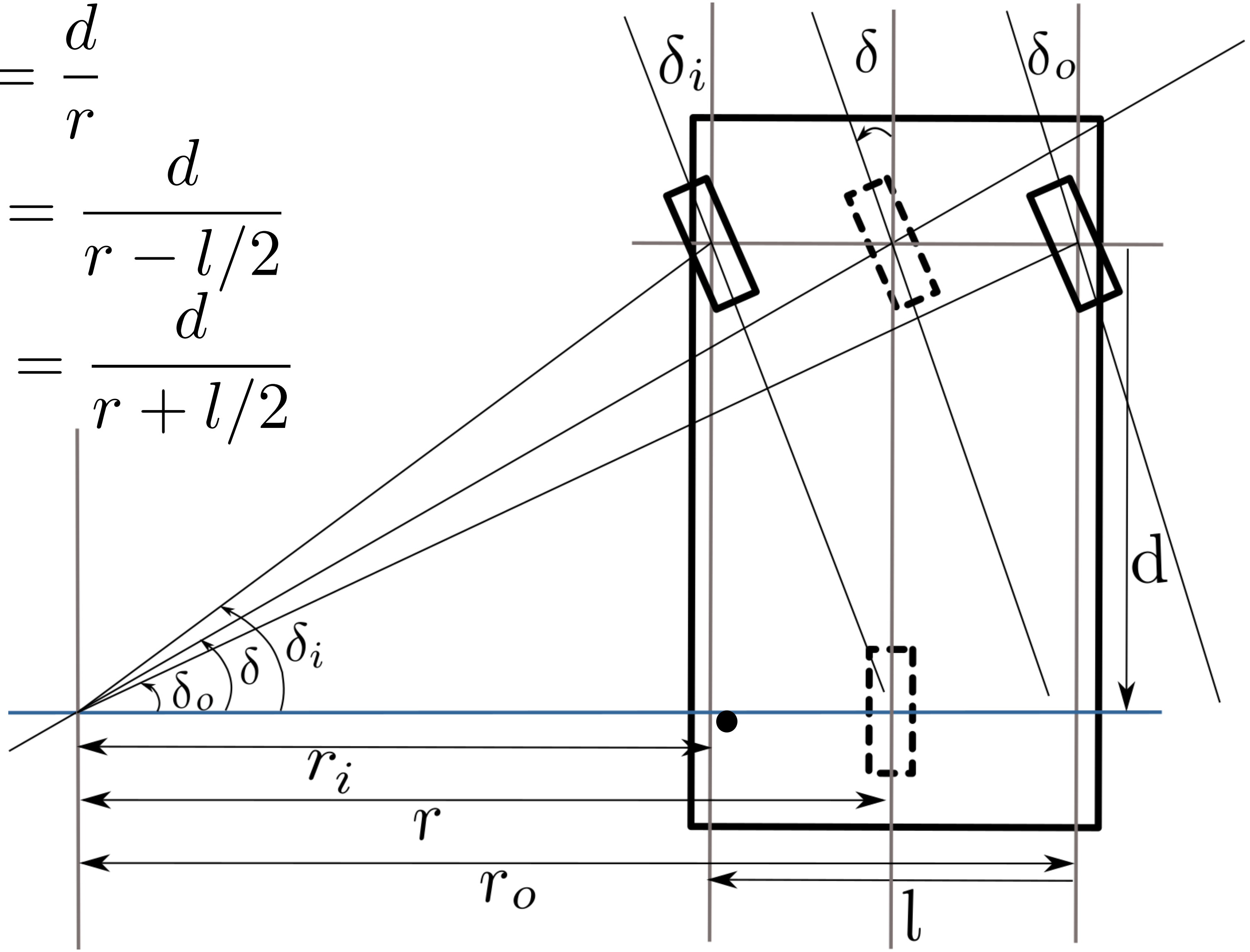


Ackermann Steering

$$\tan\delta = \frac{d}{r}$$

$$\tan\delta_i = \frac{d}{r - l/2}$$

$$\tan\delta_o = \frac{d}{r + l/2}$$



Vehicle Kinematics

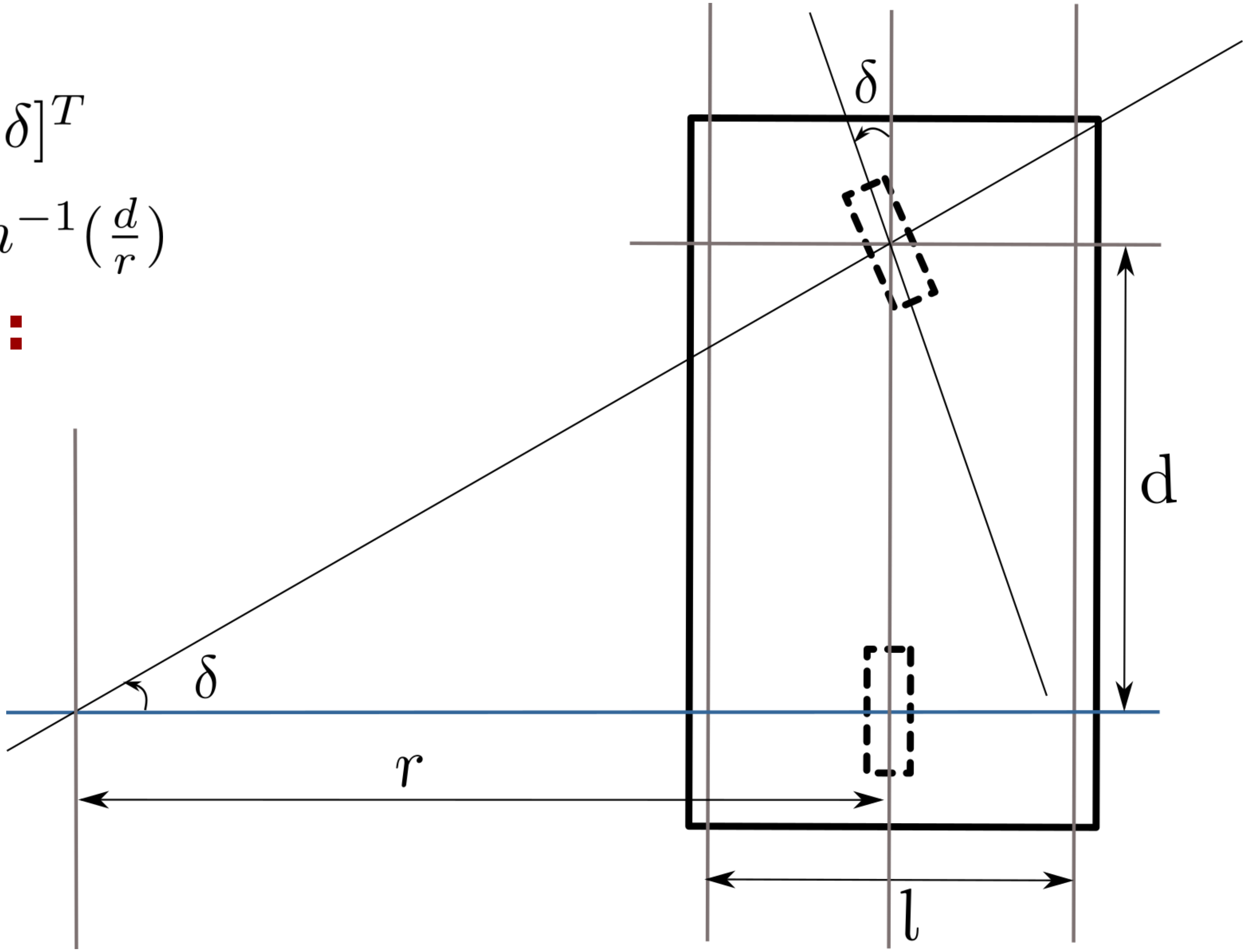
State:

$$[x, y, \theta, \delta]^T$$

$$\delta = tan^{-1}(\frac{d}{r})$$

Control:

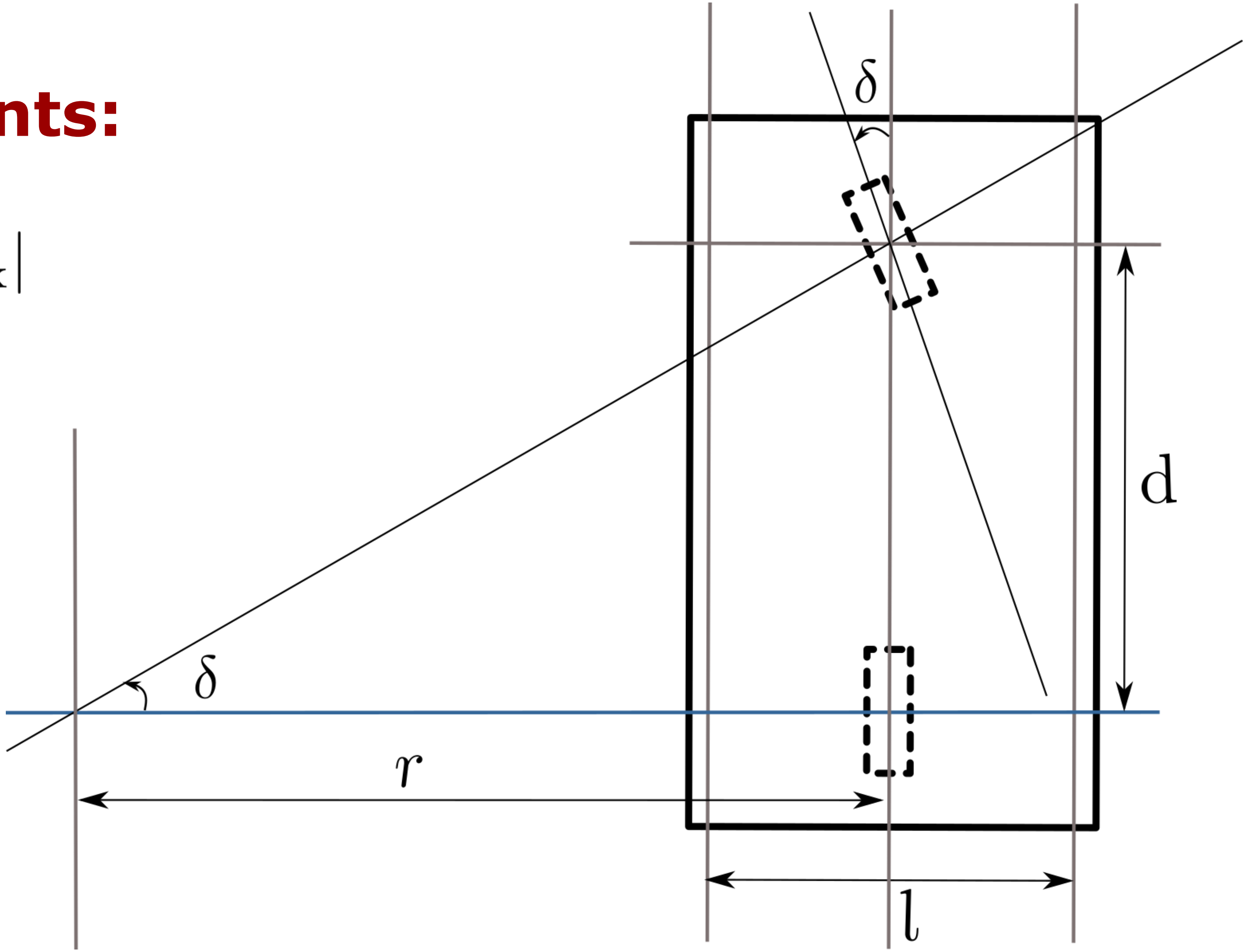
$$[v, \dot{\delta}]^T$$



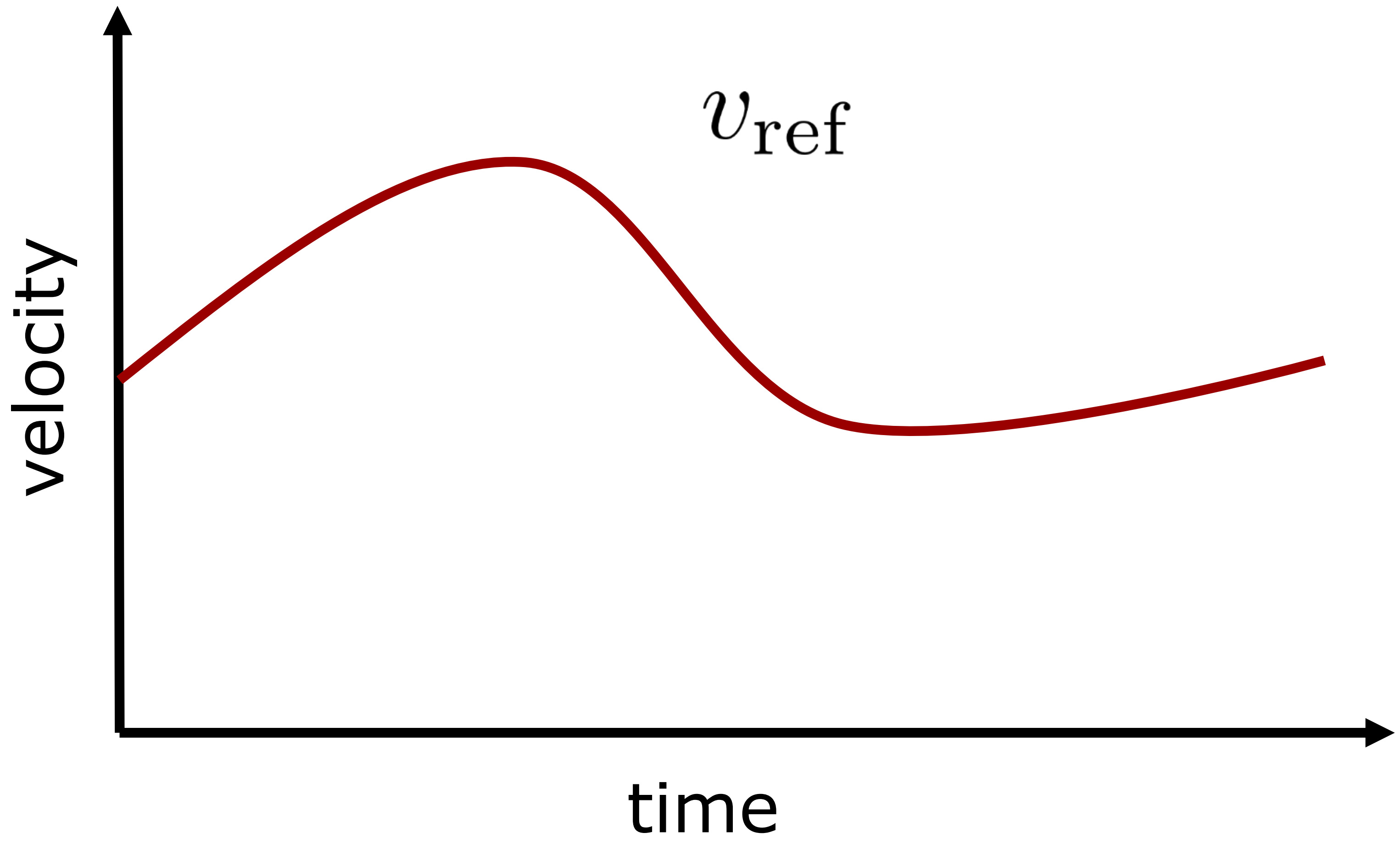
Control

Constraints:

$$v < v_{\max}$$
$$\delta < |\delta_{\max}|$$



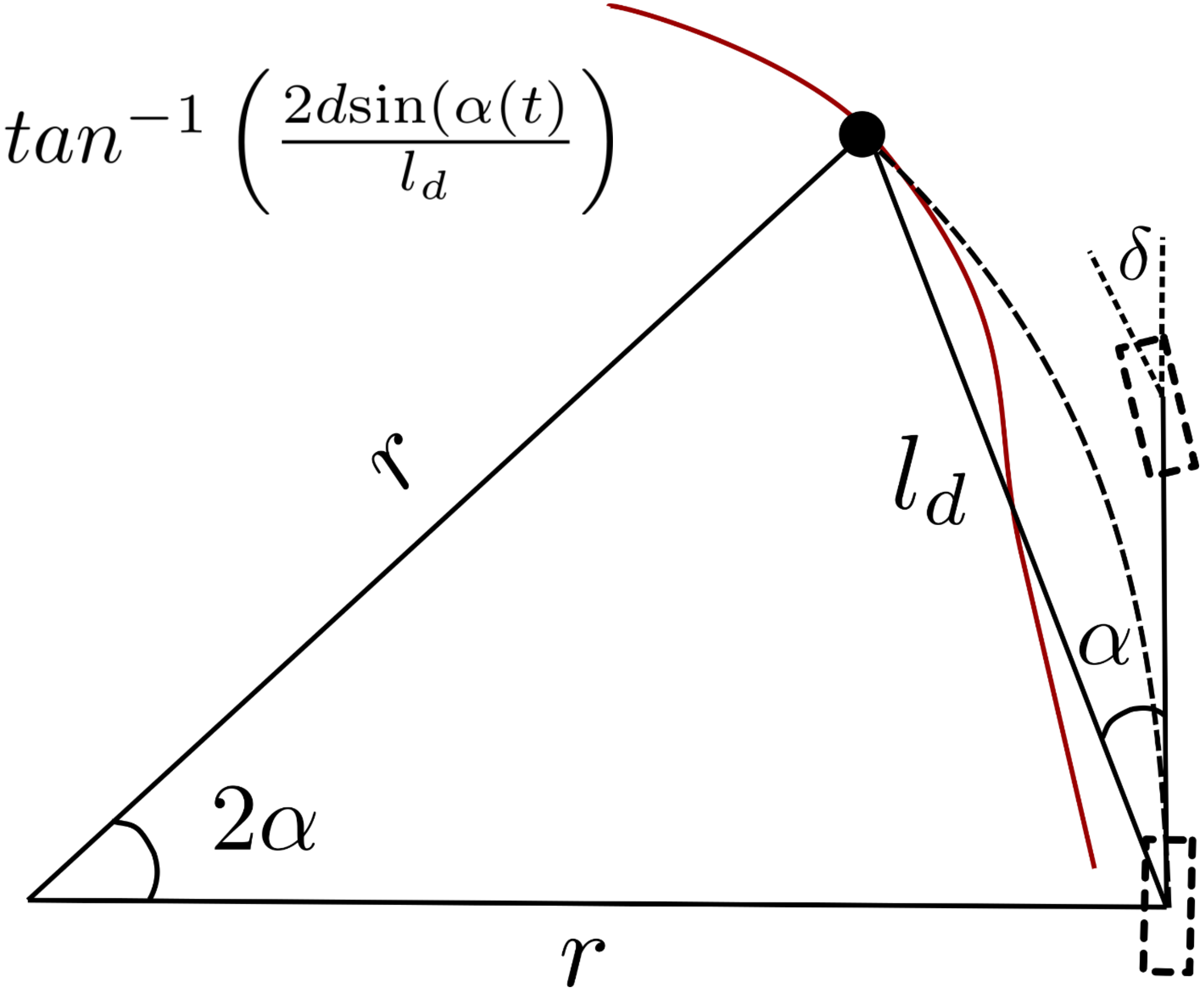
Longitudinal Control



How can we achieve this? PID Controller

Lateral Control

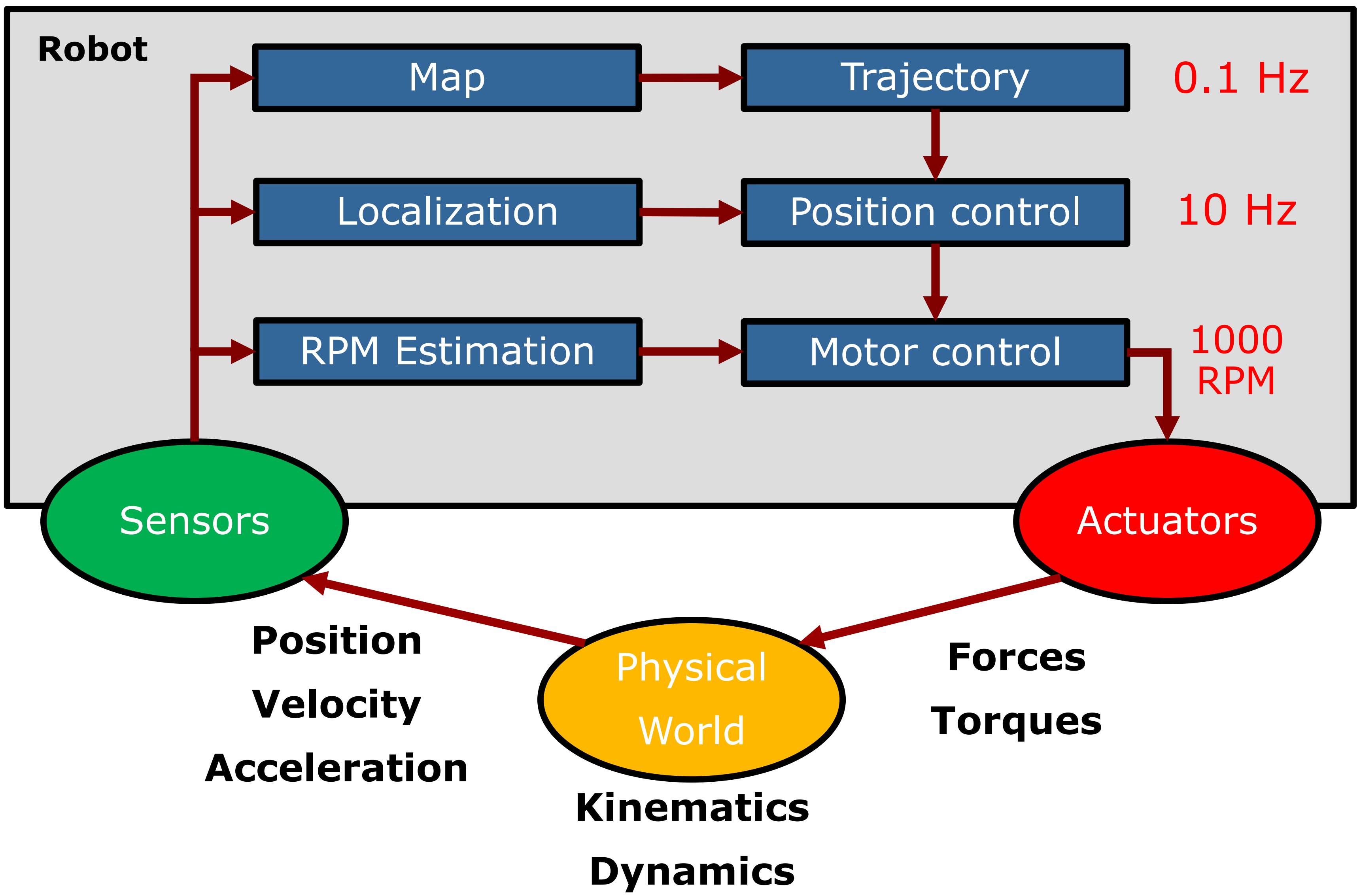
$$\delta = tan^{-1}\left(\frac{2d\sin(\alpha(t))}{l_d}\right)$$



Control in action ...

[MIT racecar]

Cascaded control



Control design goals

- Accuracy
- Safety
- Robustness
- Response time
- Maintenance
- And other application specific goals …

Advanced control techniques

- Optimal control
- Linear Quadratic regulator (LQR)
- Robust control
- Adaptive control
- Failsafe control
- Learning based control
- Many more techniques …

Optimal Control

- Find the controller gives the best performance.
- How to measure performance?
- What would be a **good performance measure**?
  - Minimize the error?
  - Minimize the controls necessary?
  - Combination of both?

Linear Quadratic Regulator

- Discrete-time **linear** system

$$x_{k+1} = Ax_k + Bu_k$$

- **Quadratic** cost function

$$J = \sum(x_k^T Q x_k + u_k^T R u_k)$$

- **Goal** : Finds the control with the lowest cost.

Non-Linear Control

- What if the system has non-linear dynamics?
- Solving a non-linear optimization problem is typically expensive
- Linearize the system and solve as LQR
- Solve non-linear optimization problem for a short horizon
- Results in Model Predictive Controller (MPC)

Non-Linear Control Example

[Dynamic Design Lab, Stanford 2017]

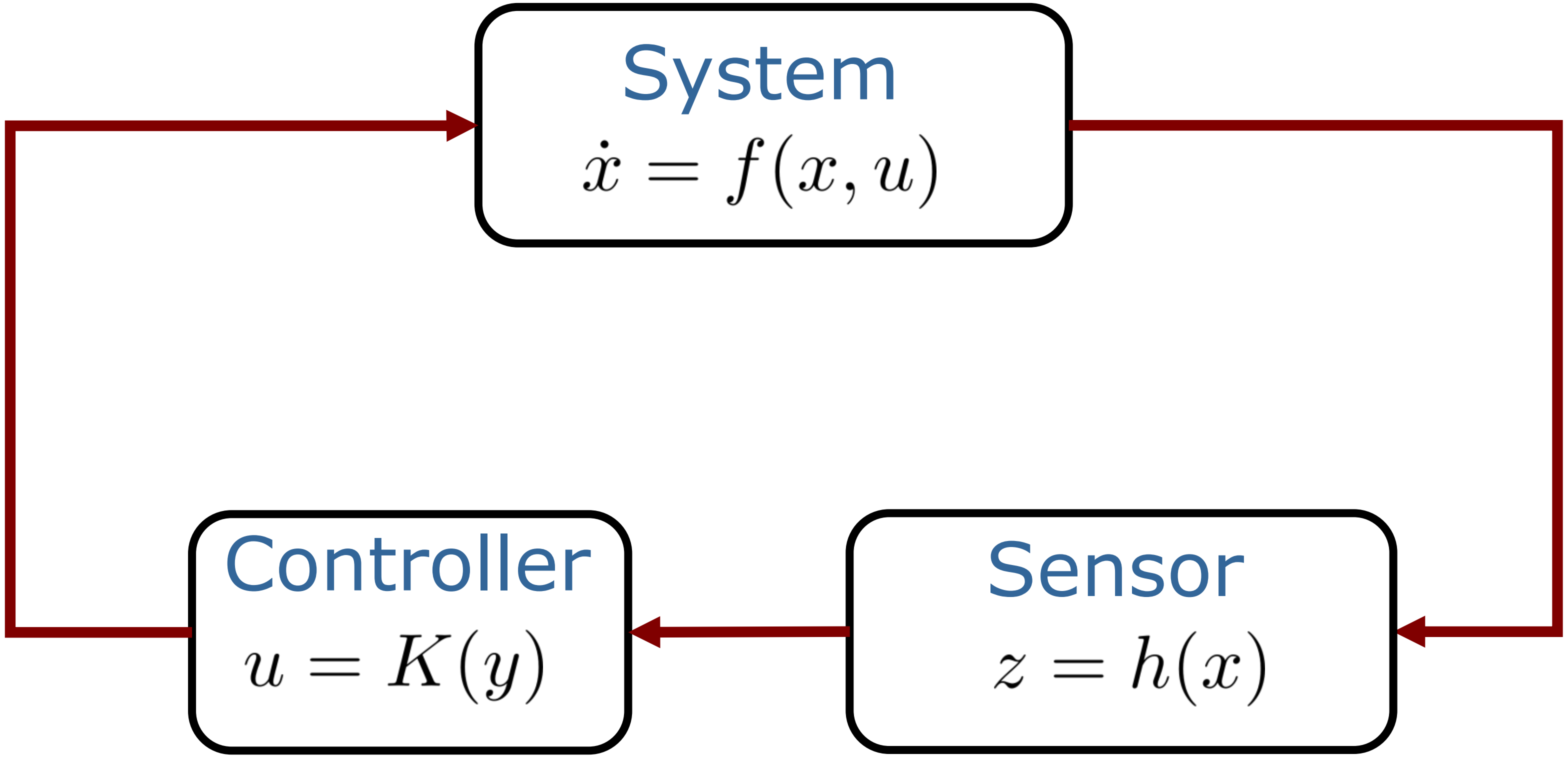
Adaptive control

- **Idea:** change the control law by **estimating system parameter** and update it in real time.
- Adapt coefficients based on meta-observations.
- Deals with time varying/uncertain parameters.
- Ex: Decrease in airplane as the mass decreases due to fuel consumption

Robust control

- **Idea:** Design that explicitly takes uncertainty into account.
- Define a bound for the uncertainty in model parameters.
- Control law guarantees stability as long as the uncertainty is within the bounds.
- But the control law is **static** unlike adaptive control.

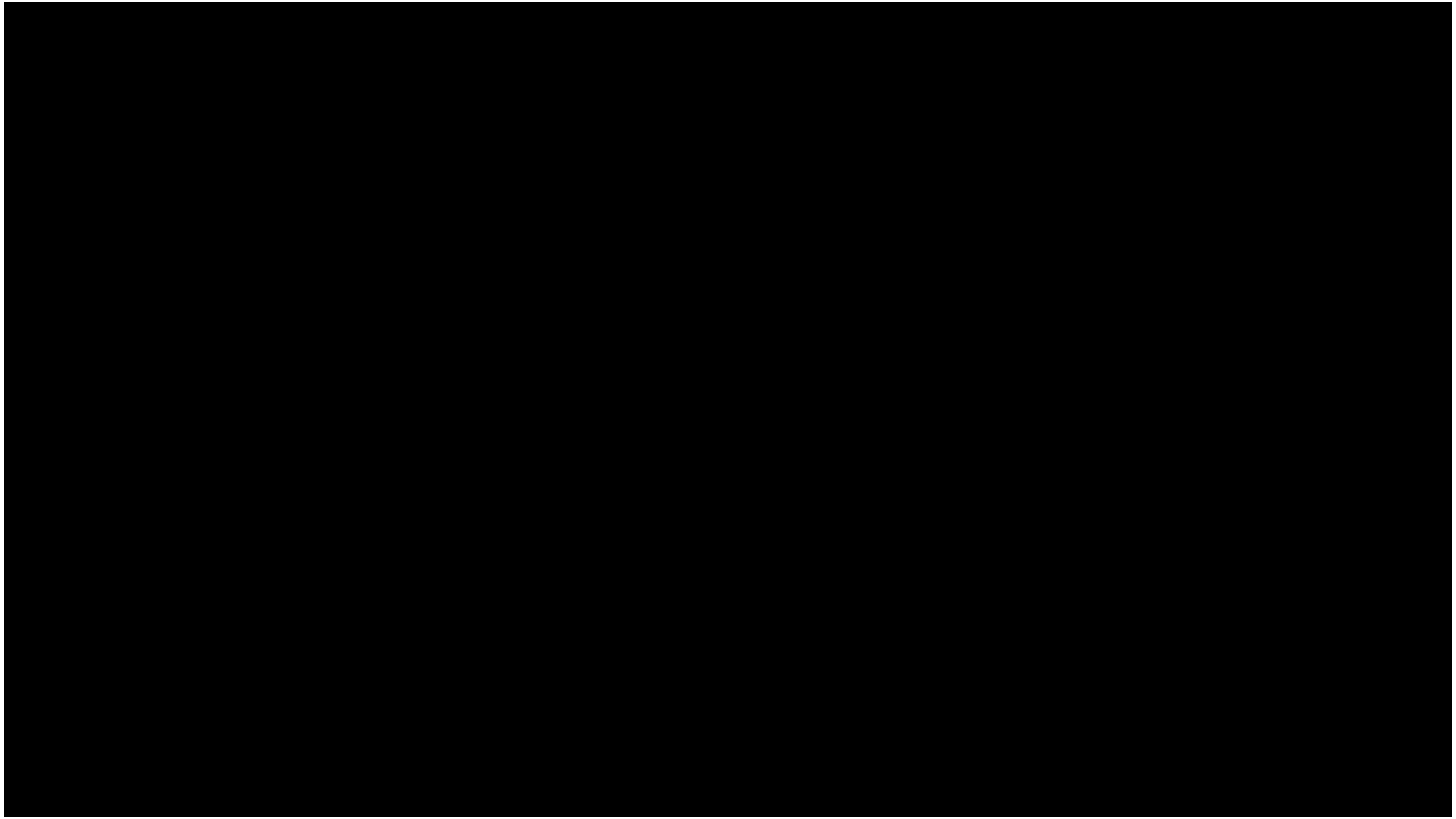
Learning based Control



Learning with experience

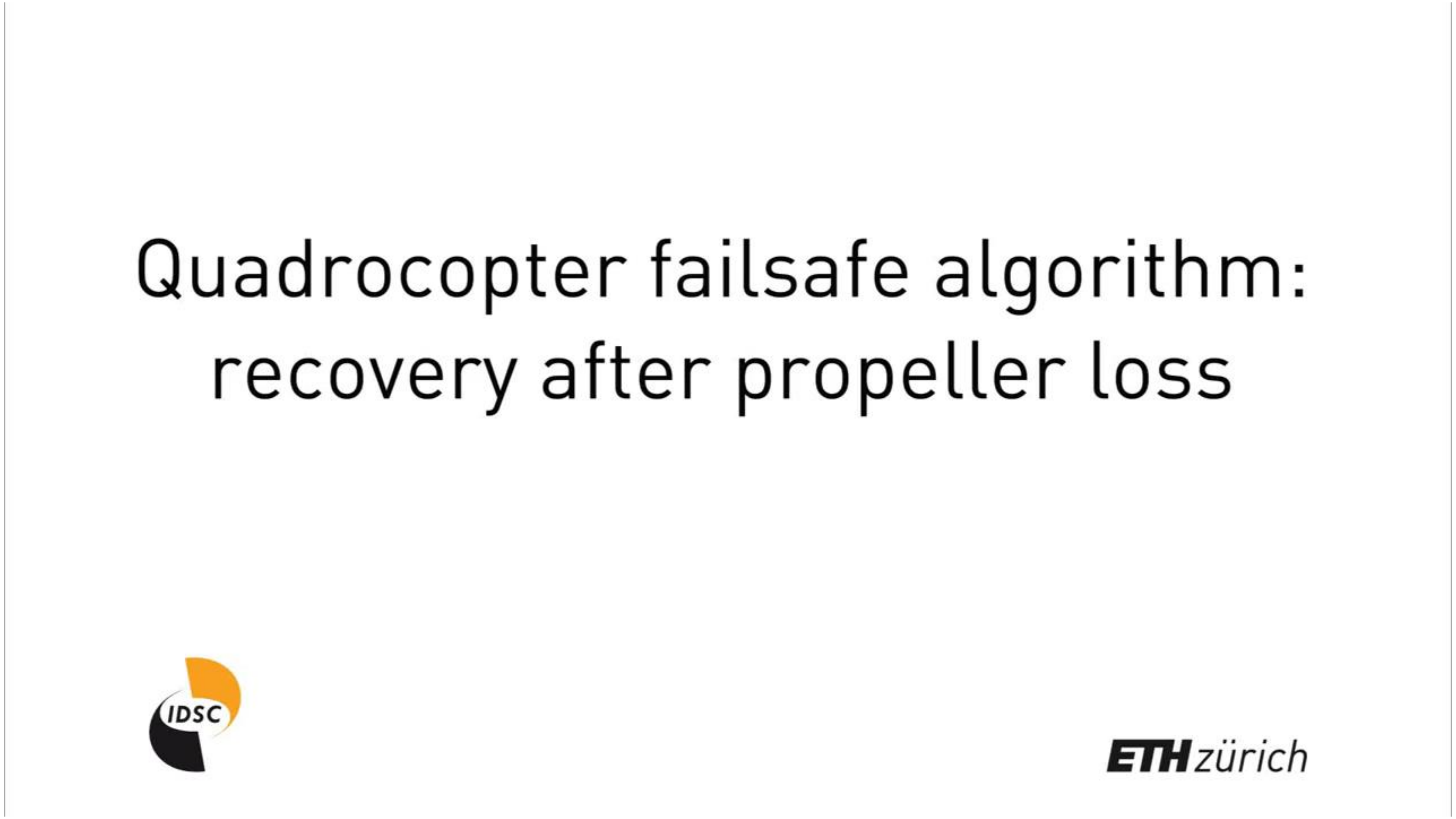
[Schoellig et al. 2012]

Learning from simulations



[Kolter et al, 2010]

Failsafe control



[Mueller et al.]

Summary

- Motors and low level controllers
- Feedback control
- PID Controller
- Path following control
- Some advanced control techniques

Acknowledgements

- “Visual navigation for flying robots” by Dr. Jürgen Strum

Link: https://www.edx.org/course/autonomous-navigation-flying-robots-tumx-autonavx-0

- “Control for Mobile Robots” by Dr. Magnus Egerstedt

Link: https://www.coursera.org/learn/mobile-robot

- “Robotics, Control and Vision” by Dr. Peter Corke

Thank you for your attention