Autonomous Control Strategy

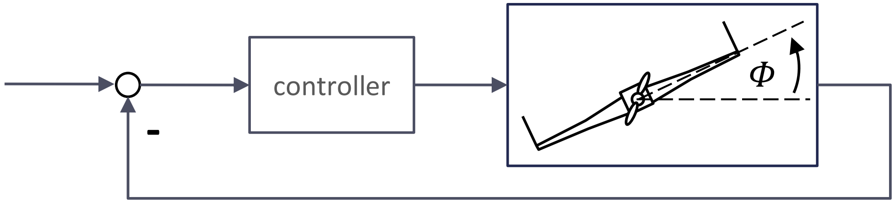
Prof. Dr. Jörg Fischer

Hochschule Offenburg
offenburg.university

Agenda

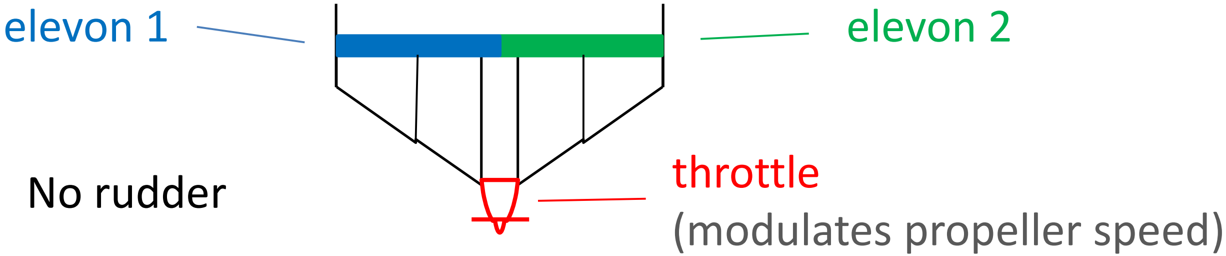
- **Control Structure of the Delta Drone**
  - Control Inputs
  - Lateral and Longitudinal control

- **Lateral Control**
  - Roll Controller
  - Horizontal Trajectory Controller
    - Pure Pursuite and Line of Sight
    - Carrot Chasing
    - Nonlinear Guidance Law

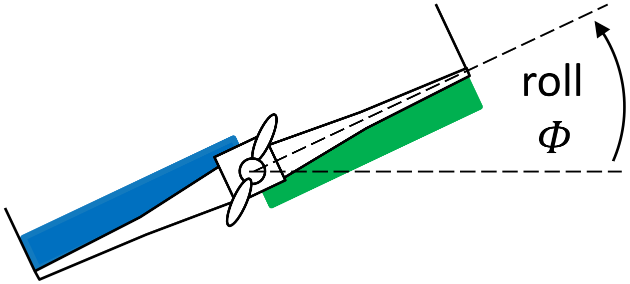


The „Delta“ Drone

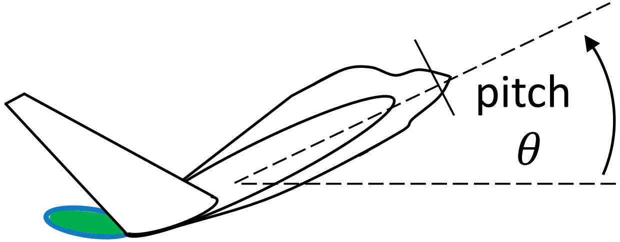
Control Inputs



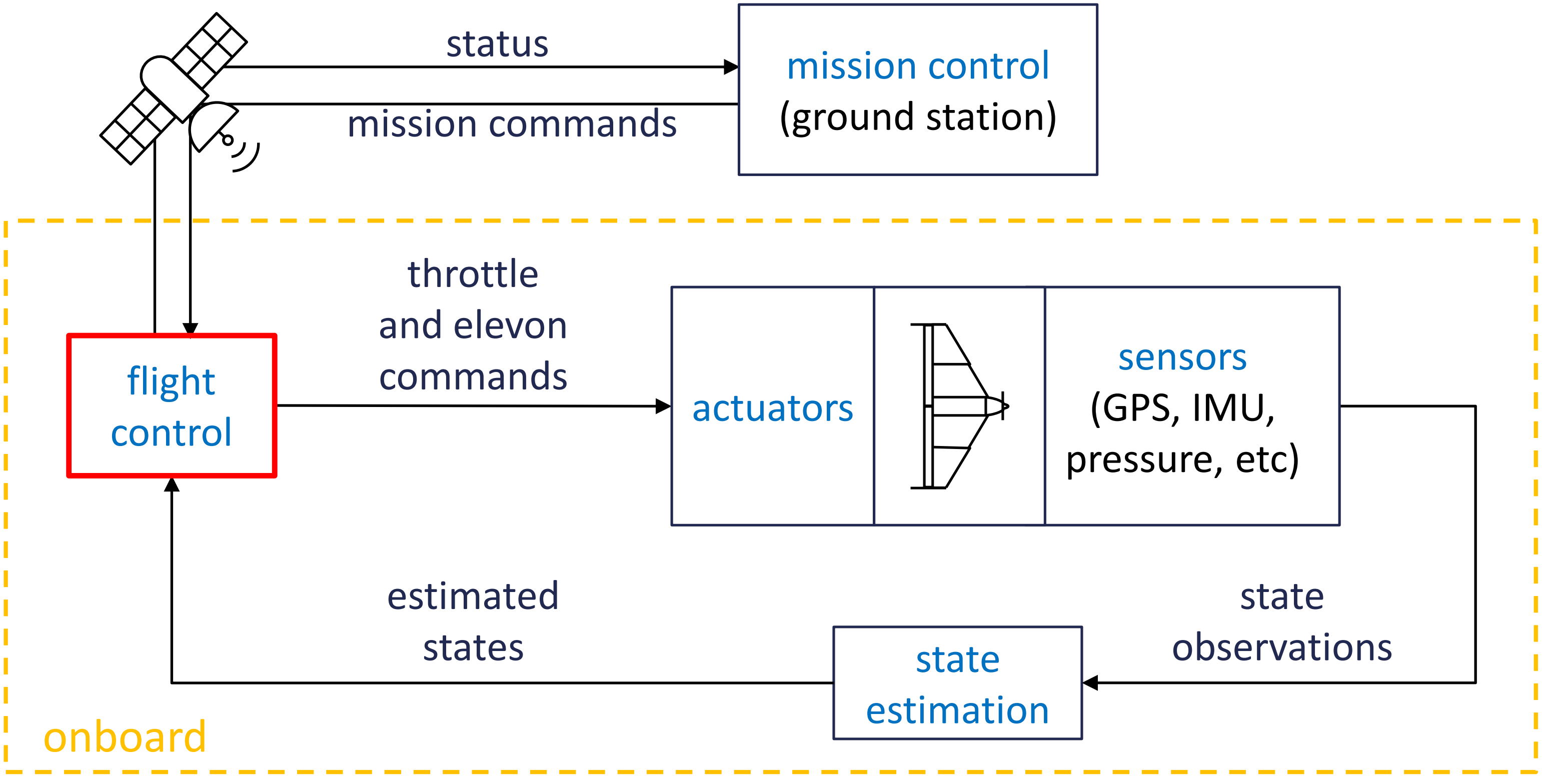
differential mode of elevons ≙ aileron



common mode of elevons ≙ elevator

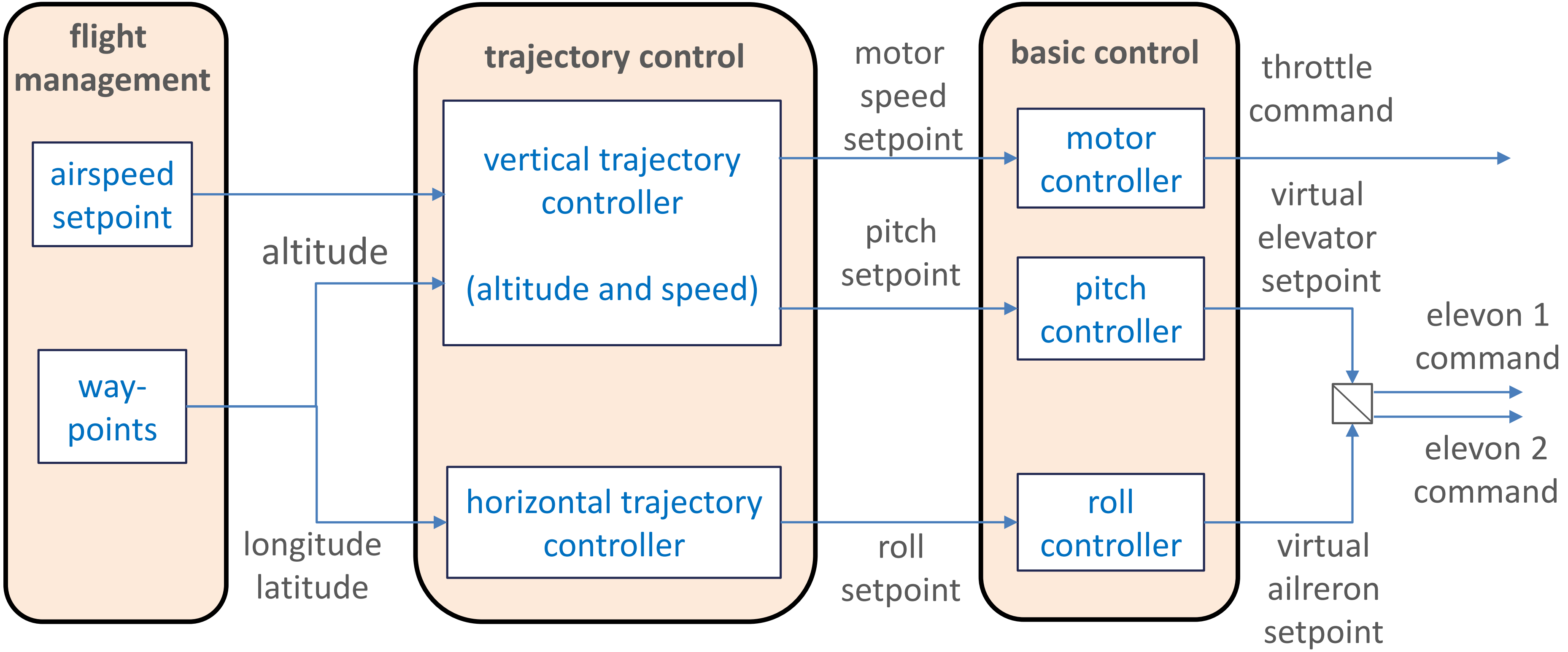


Control Structure
Overview

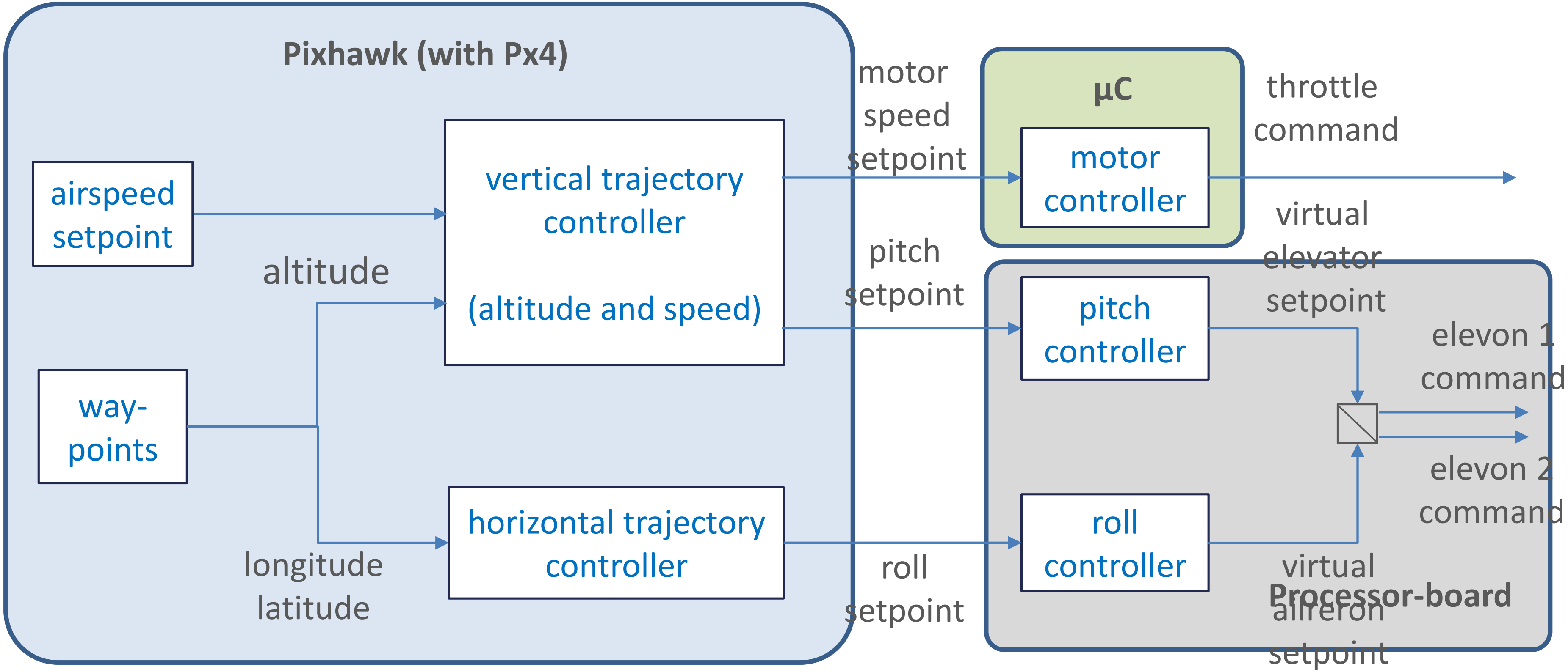


Control Structure

Flight Controller

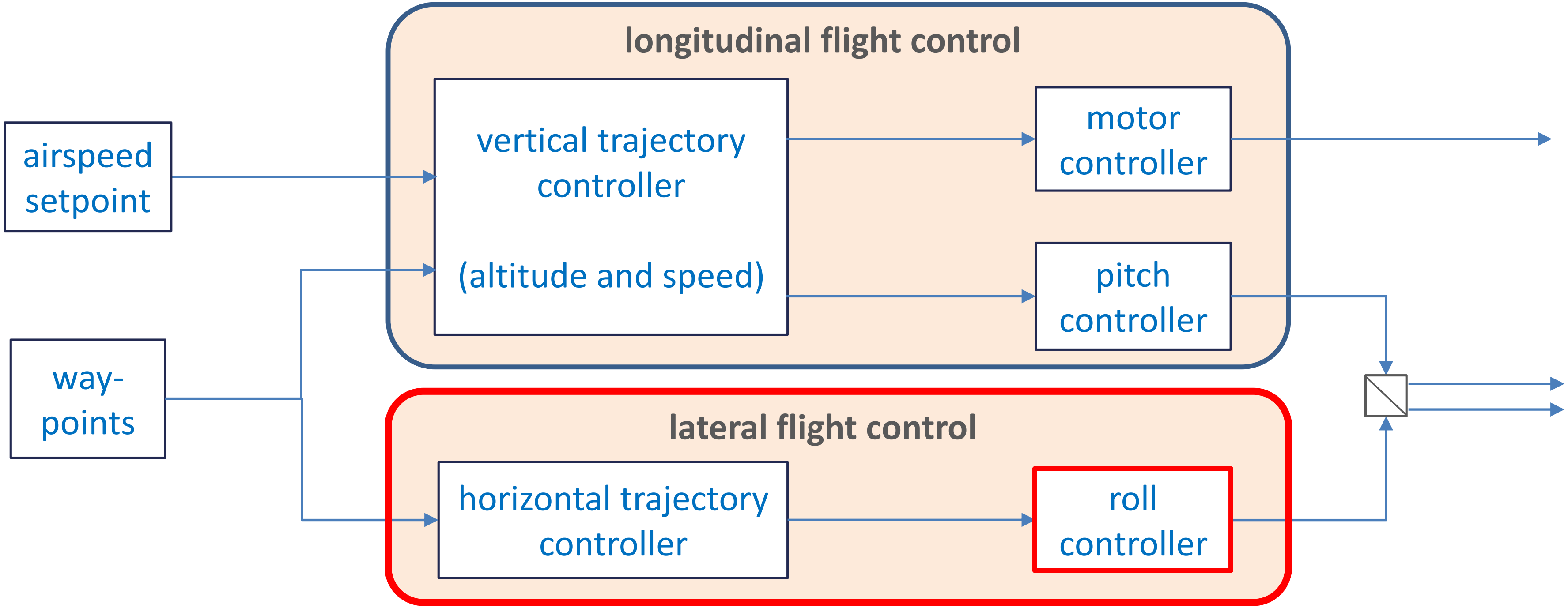


Control Structure
Flight Controller



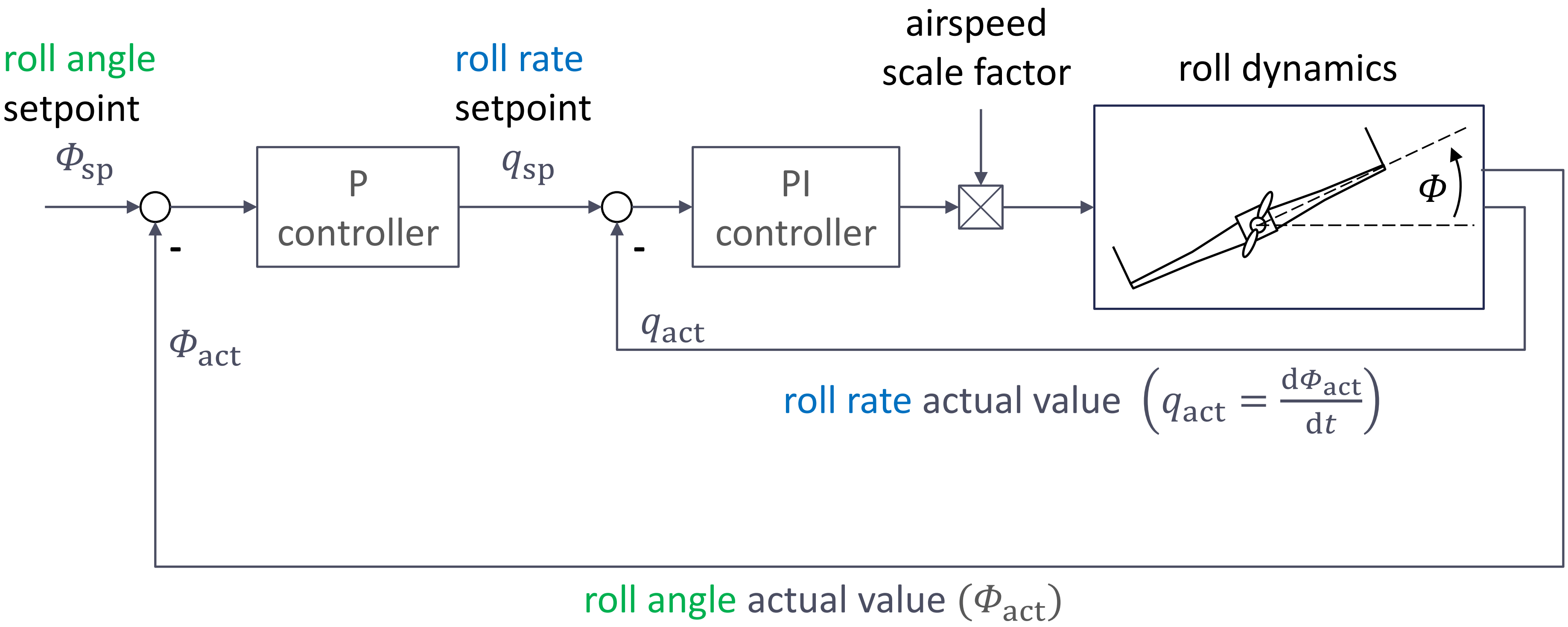
Control Structure

Flight Controller



Roll Controller

Structure

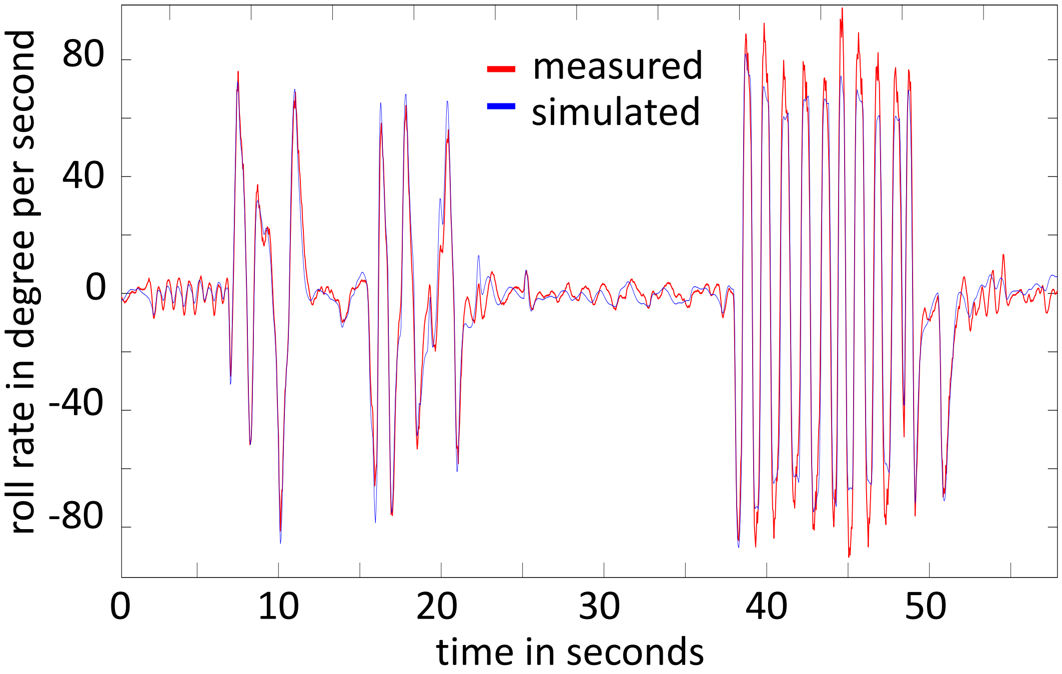


Roll Controller

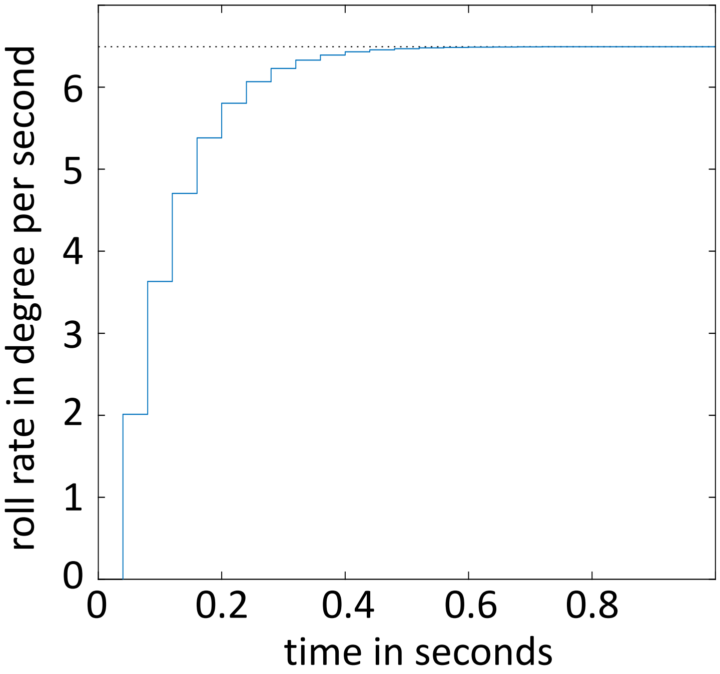
Model of roll dynamics

roll dynamics identified from data
(2nd order ARX model)

validation of identified model



step response of model

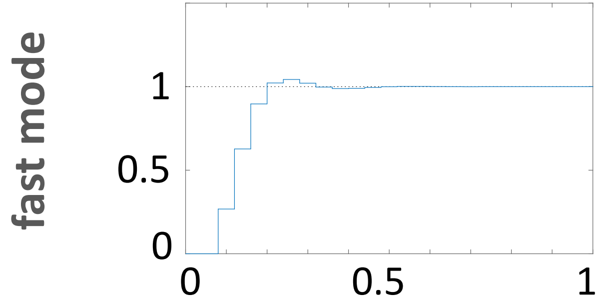


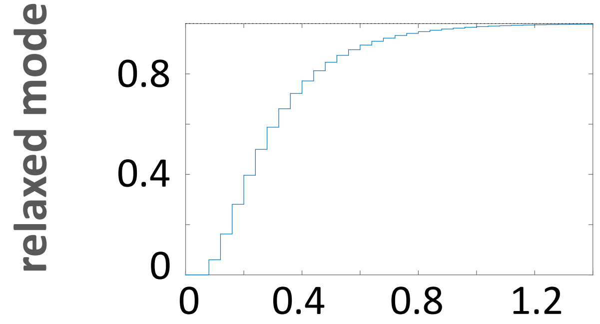
Roll controller

Tuning and Simulation

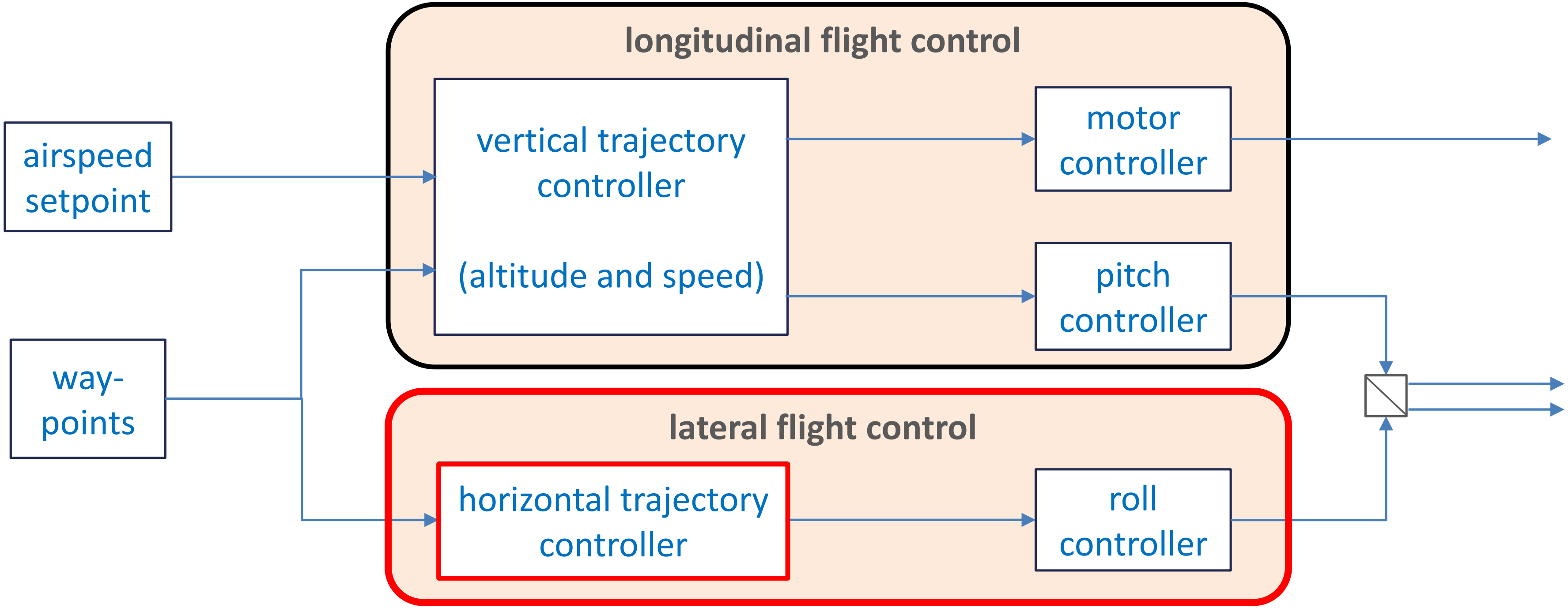
x-axis: time in seconds
y-axis: in degree

roll setpoint to roll





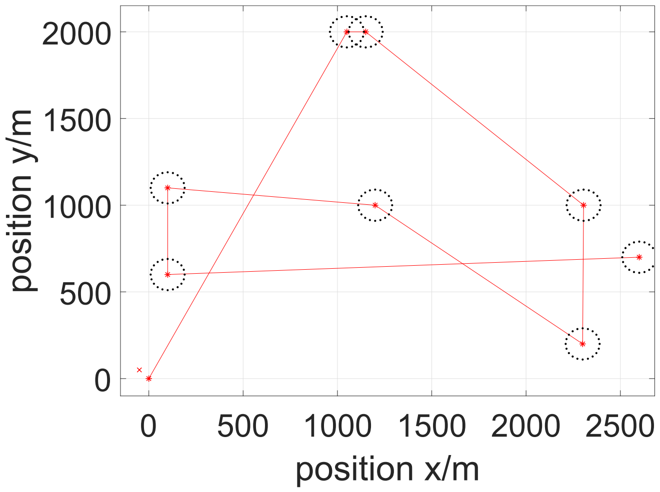
Control Structure
Flight Controller



Horizontal Trajectory Control

Comparision of three methods

- Pure Pursuit and Line of Sight Guidance (PLOS)
- Carrot Chasing
- Nonlinear Guidance Law
  (implemented in Px4)



Horizontal Trajectory Control
PLOS - Algorithm

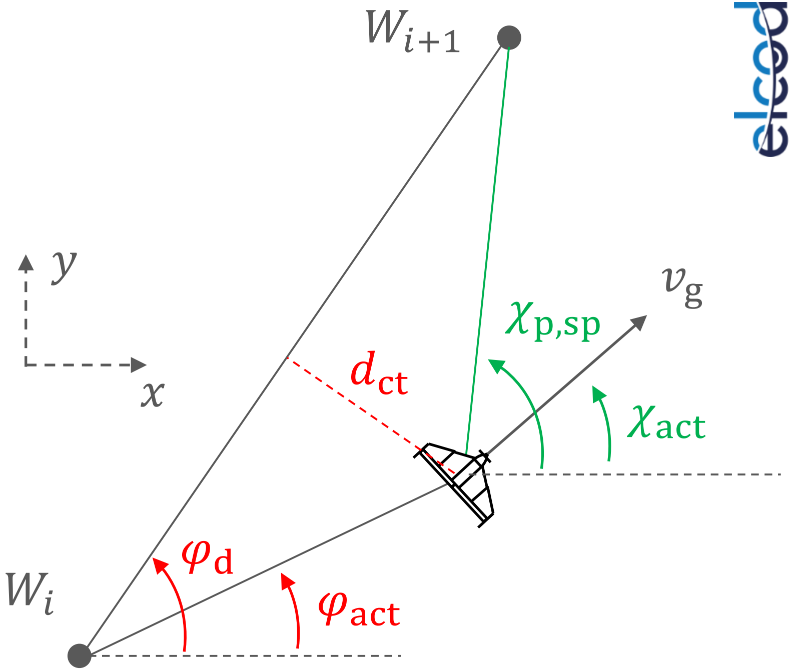
- Combines two strategies
  - pure pursuit
    drives drone to the next waypoint
  - line-of-sight (LOS) guidance
    decreases cross track error (d_{ct})

- Algorithm
  1. Calculate course setpoint

$$\chi_{\mathrm{sp}} = k_{\mathrm{P}} \cdot \left(\chi_{\mathrm{p,sp}} - \chi_{\mathrm{act}}\right) + k_{\mathrm{LOS}} \cdot d_{\mathrm{ct}} \cdot \sin(\varphi_{\mathrm{d}} - \varphi_{\mathrm{act}})$$

  2. Calculate roll setpoint via P-controller

$$\Phi_{\mathrm{sp}} = k\left(\chi_{\mathrm{sp}} - \chi_{\mathrm{act}}\right)$$

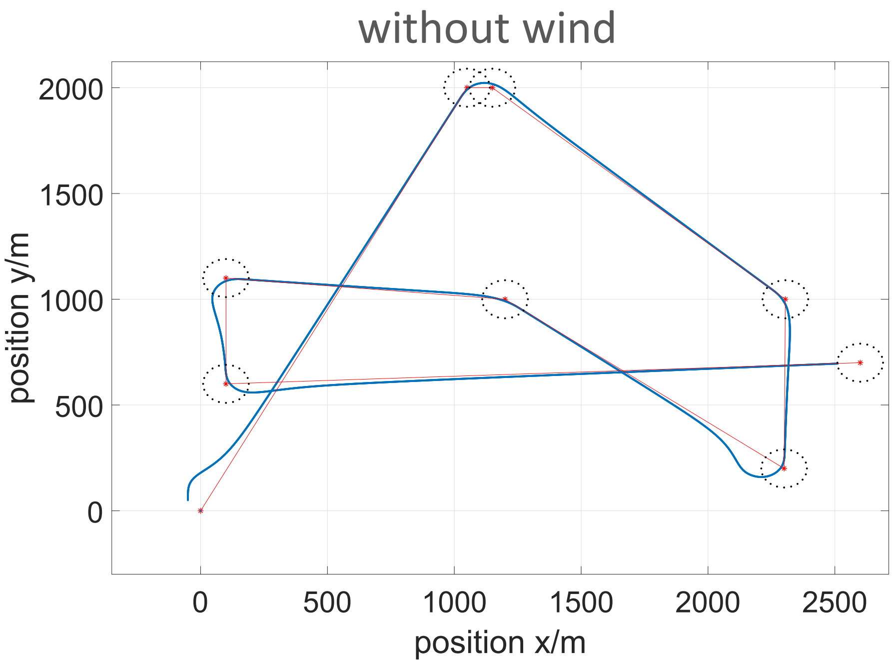


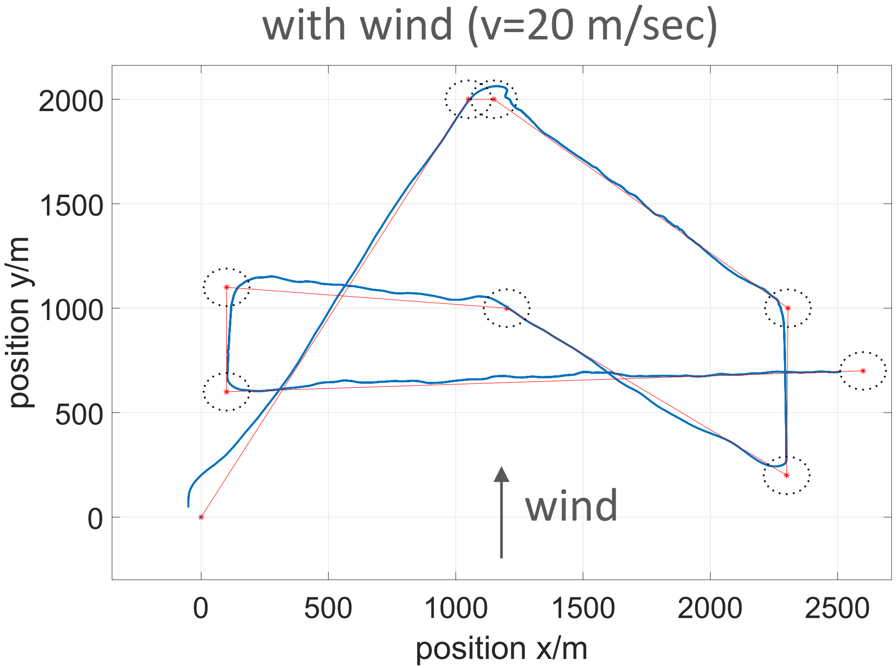
- Tuning parameters
  k_{p}, k_{LOS}

Horizontal Trajectory Control

PLOS - Simulation

waypoint
switch distance
track
flight path





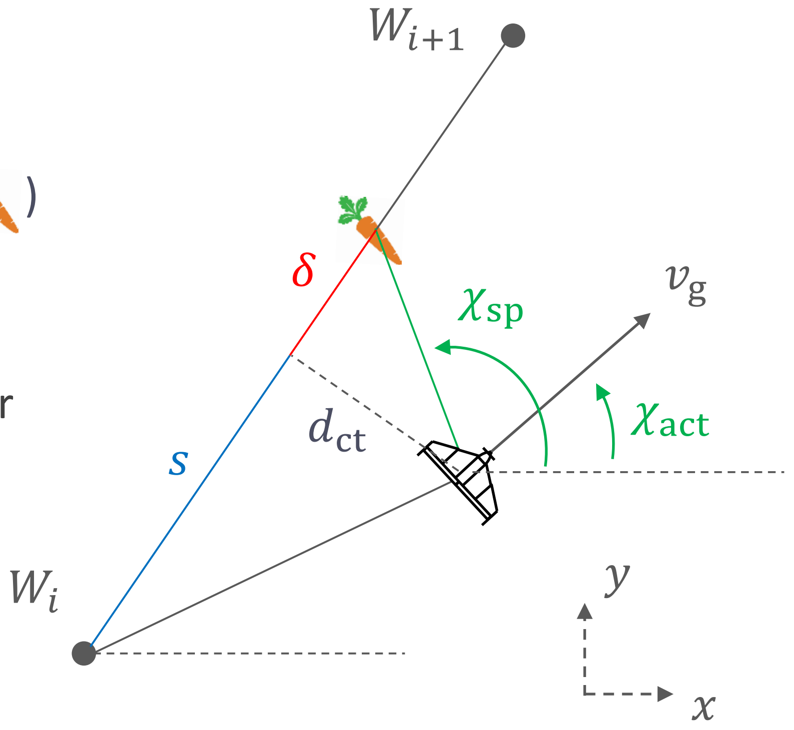
Horizontal Trajectory Control

Carrot Chasing - Algorithm

- Algorithm:
  1. Calculate (receding) virtual target point () located at distance s + δ from W_i on LOS
  2. Calculate course towards carrot (χ_{sp})
  3. Calculate roll setpoint $\left(\Phi_{\mathrm{sp}}\right)$ via P-controller

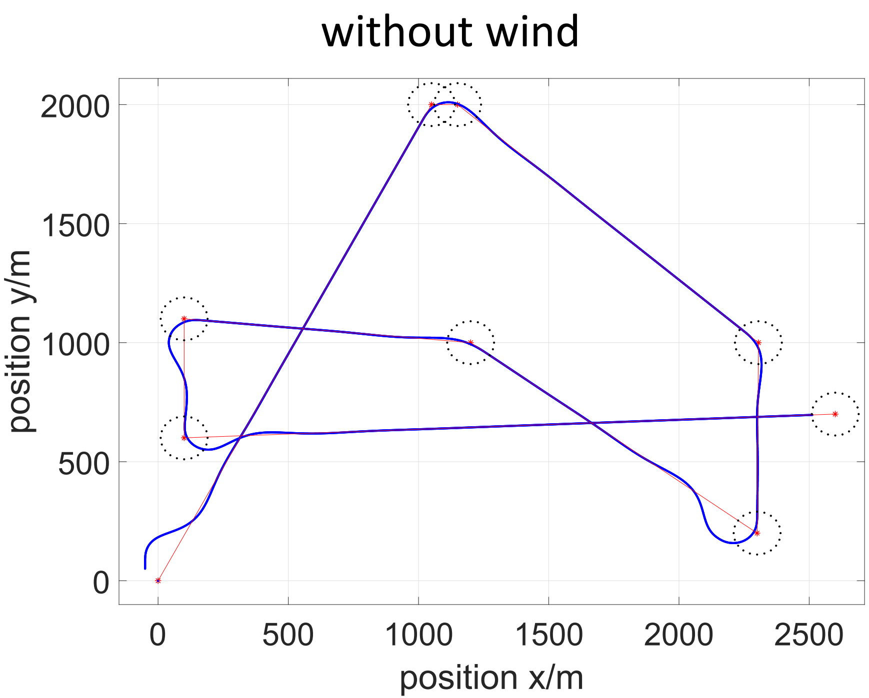
$$\Phi_{\mathrm{sp}} = k\left(\chi_{\mathrm{sp}} - \chi_{\mathrm{act}}\right)$$

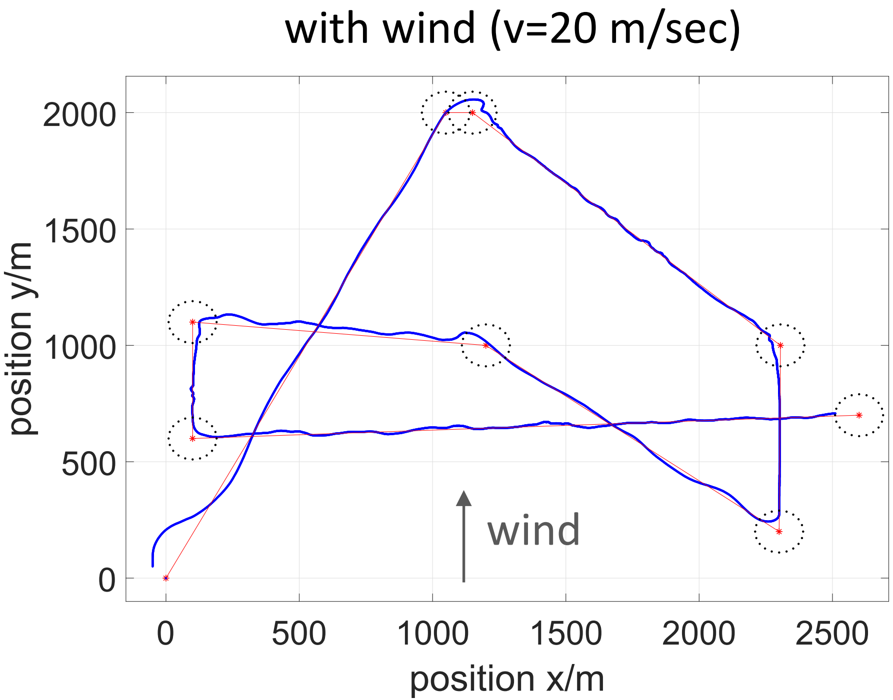
- Tuning parameters:
  $k, \ \delta$



Horizontal Trajectory Control

Carrot Chasing - Simulation





Horizontal Trajectory Control

Nonlinear Guidance Law – Algorithm

- Algorithm
  1. Calculate receding virtual target point (VTL) located at distance L_1 from drone on LOS
  2. Calculate angle η
  3. Calculate desired lateral acceleration

$$a_{\text{lat,des}} = 2 \cdot \frac{v_g^2}{L_1} \cdot \sin(\eta)$$

  4. C

- Tuning parameter: L_1

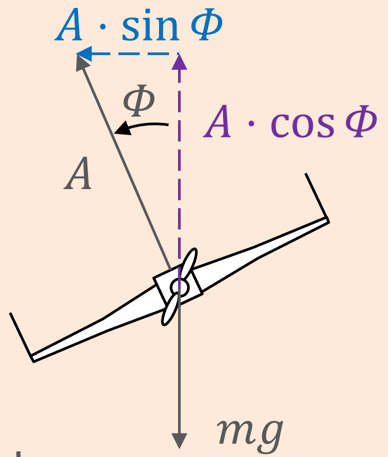
W_{i+1}

Why? This would move the drone along

$$a_{\text{centripetal}} = \frac{v_g^2}{R}$$

$$L_1 = 2R \cdot \sin(\eta) \quad \Leftrightarrow \quad \frac{1}{R} = \frac{2}{L_1} \cdot \sin(\eta)$$

Why? In a coordinated turn holds

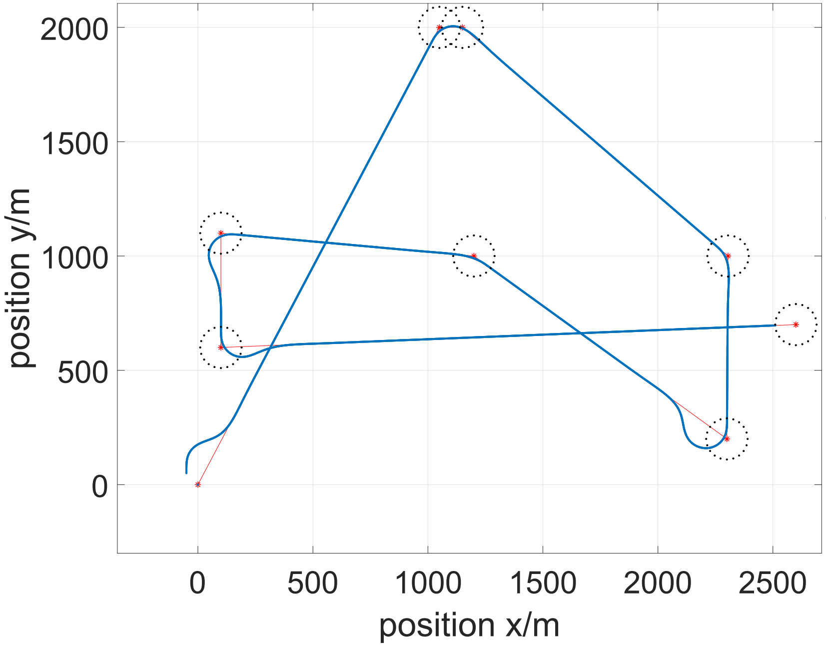


$$\left.\begin{array}{l} A \cdot \cos(\Phi) \overset{!}{=} mg \\ A \cdot \sin(\Phi) = m a_{\text{lat}} \end{array}\right\} \tan(\Phi) = \frac{a_{\text{lat}}}{g}$$

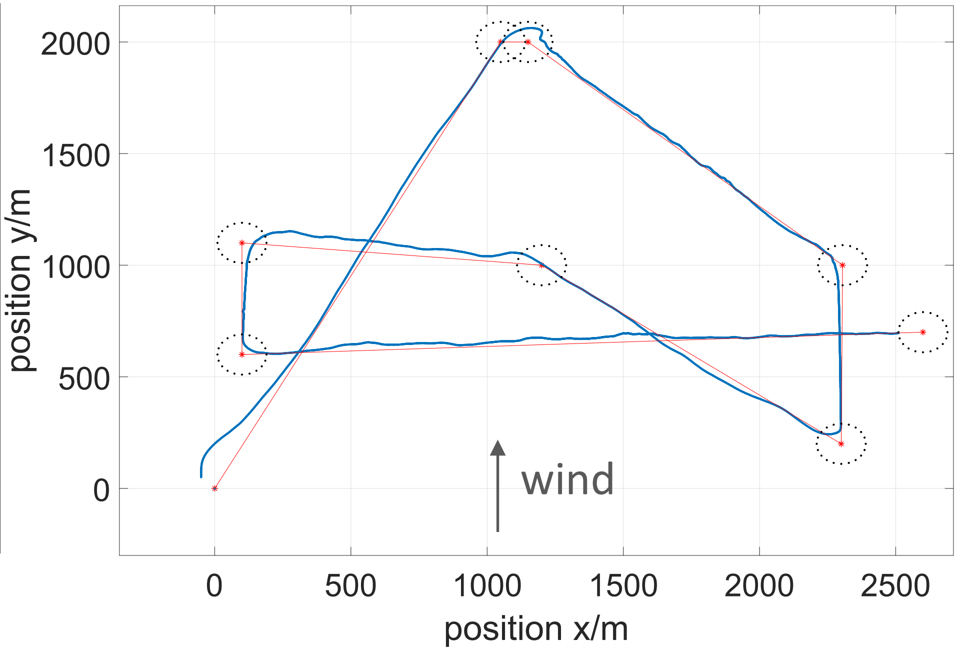
Horizontal Trajectory Control

Nonlinear Guidance Law - Simulation

without wind



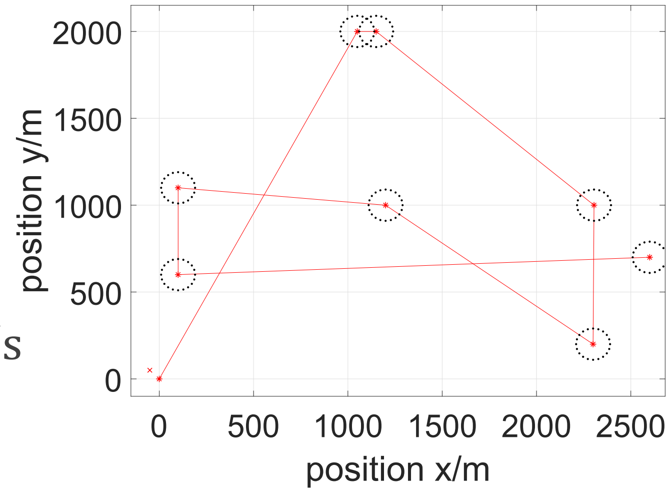
with wind (v=20 m/sec)



Horizontal Trajectory Control

Comparison of the algorithms (1)

- 1000 Monte Carlo Simulation runs
- Drone starts at point (-50, 50)
- airspeed setpoint is 30 m/s
- wind is filtered white noise with $v_{\text{mean}} = 20\ \text{m/s}$
- wind direction changes every 30 s
- performance metric



control energy

$$U = \sum_{t=0}^{t=T} \text{aileron(t)}^2$$

tracking error

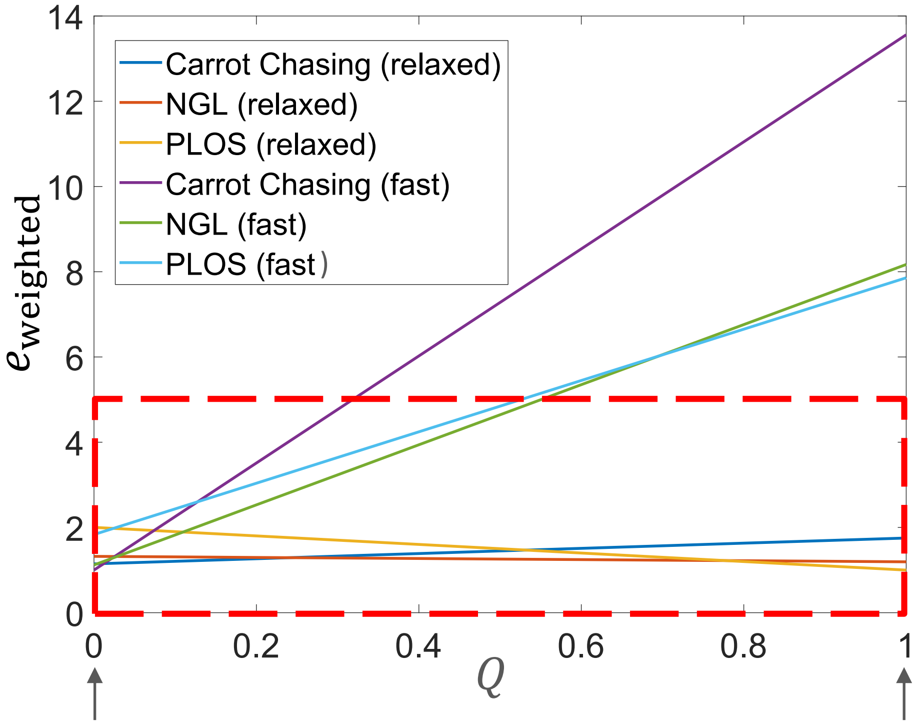
$$E_{\text{CT}} = \sum_{t=0}^{t=T} \text{crossTrackError(t)}^2$$

normalizing

$$e_{\text{weighted}}(Q) = Q\bar{U} + (1-\text{Q})\ \bar{E}_{\text{CT}}$$ with Q = 0…1

Horizontal Trajectory Control

Comparison of the algorithms (2)



only cross track error

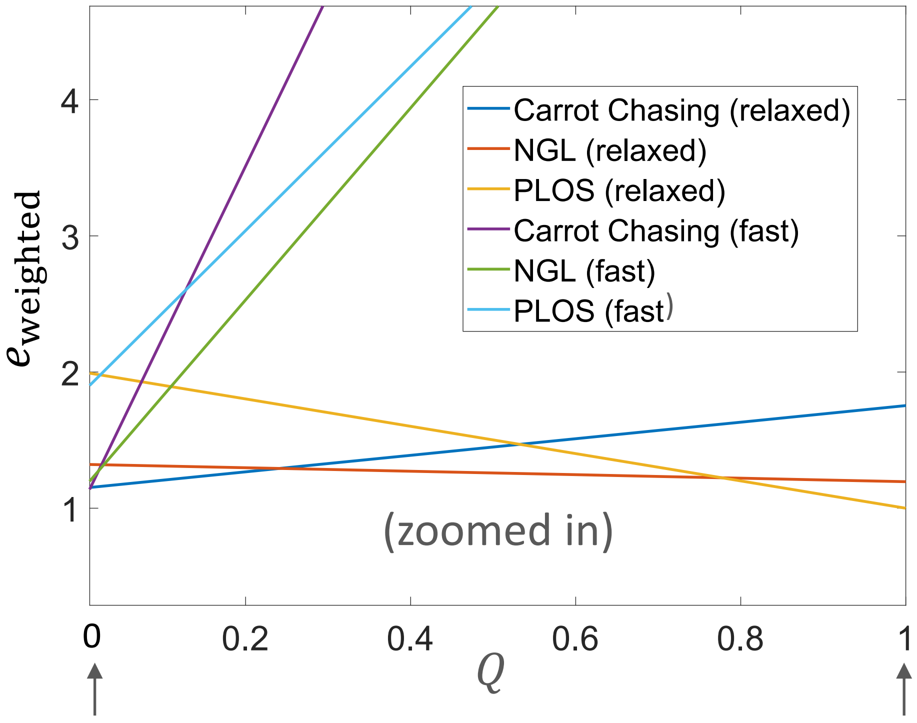
only control effort

Resuts for fast mode

- Fast mode needs much more control effort but reduces mean cross track error only be a factor of 2
- NGL and PLOS better than Carrot Chasing
- NGL slightly more efficient than PLOS

Horizontal Trajectory Control

Comparison of the algorithms (2)



only cross track error

only control effort

Resuts for relaxed mode

- PLOS uses least control but has high cross track error
- Carrot Chasing has least cross track error but uses much control effort
- NGL efficient compromise

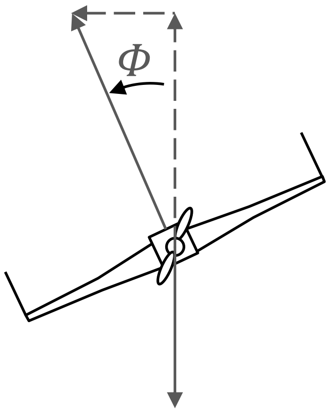
⇒ NGL chosen

Conclusion

- **Overview** of Control Structure

- **Lateral control**
  - Roll controller
  - Trajectory controller
    - PLOS
    - Carrot Chasing
    - NGL

- **NGL** gives best compromise between control effort and cross track error



Thank you for your attention!