AER 1217 – Autonomy for UAVs

Lecture 2 - Quadrotor Dynamics and Control

Prof. Steven Waslander
(slides by Prof. Angela Schoellig)

Institute for Aerospace Studies
UNIVERSITY OF TORONTO

CONTEXT

- Today’s lecture will be used in the lab next week!

Institute for Aerospace Studies
UNIVERSITY OF TORONTO

AER1217: Winter 2021 Lab 2
February 3, 2022

AER1217: Development of Unmanned Aerial Vehicles

Lab 2: Quadrotor Simulation and Control Design

1 Introduction

This is the second lab in a series designed to complement the lecture material and help you with the course project. This lab requires the design of a quadrotor position controller, and implementation in ROS. It is set up to work with Gazebo, which simulates your interface with the Parrot AR.Drone UAV and the Vicon motion capture system for position and attitude measurements. To be able to complete this lab within your dedicated time slot, you should have preliminary knowledge of ROS, and Python programming.

1. Why quadrotors?

2. Why control?

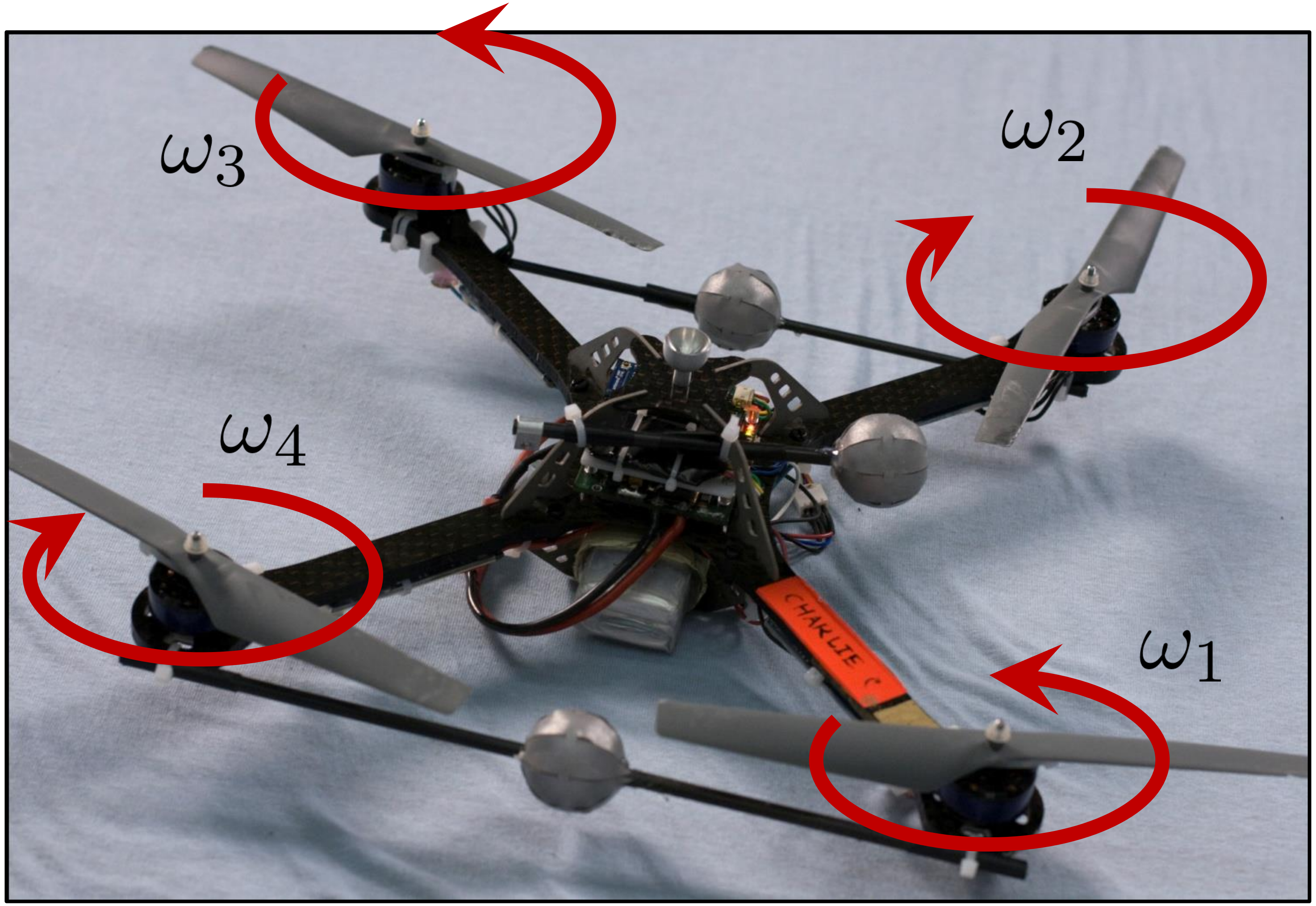
1. Why quadrotors?

WHY QUADROTORS?

1. Rigid frame
2. Four independently controlled motors
3. Move by changing the motor speeds
4. Vertical take-off and landing

- Mechanically simple
- Highly maneuverable

QUADROTOR VEHICLE



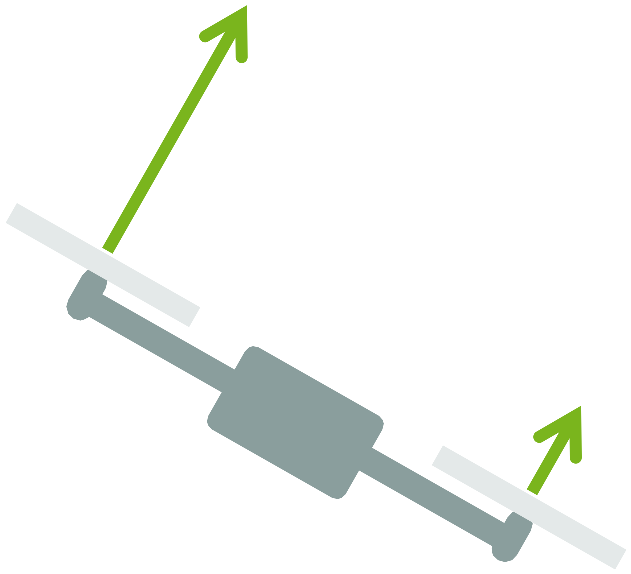
Vary the speeds of the rotors to control:

- the position
- and orientation of the robot

ROLL AND PITCH

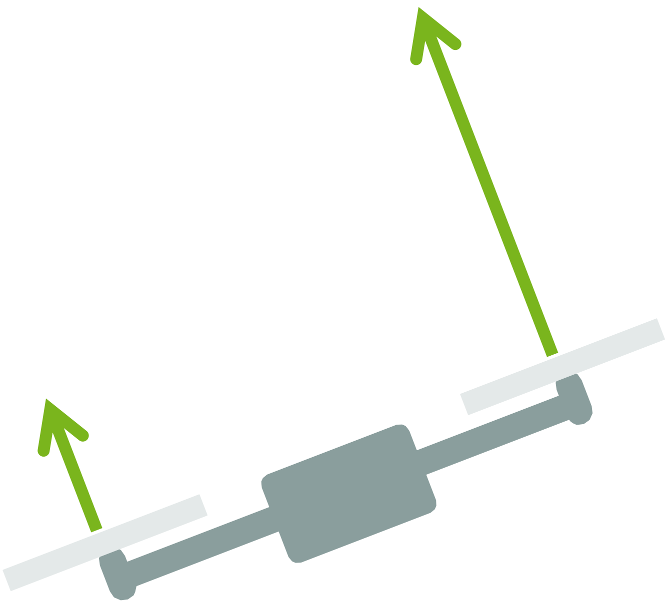
Green arrows represent the **force produced by the motor**, which is roughly **proportional to the squared rotor speed**

ROLL AND PITCH



ROLL AND PITCH

ROLL AND PITCH



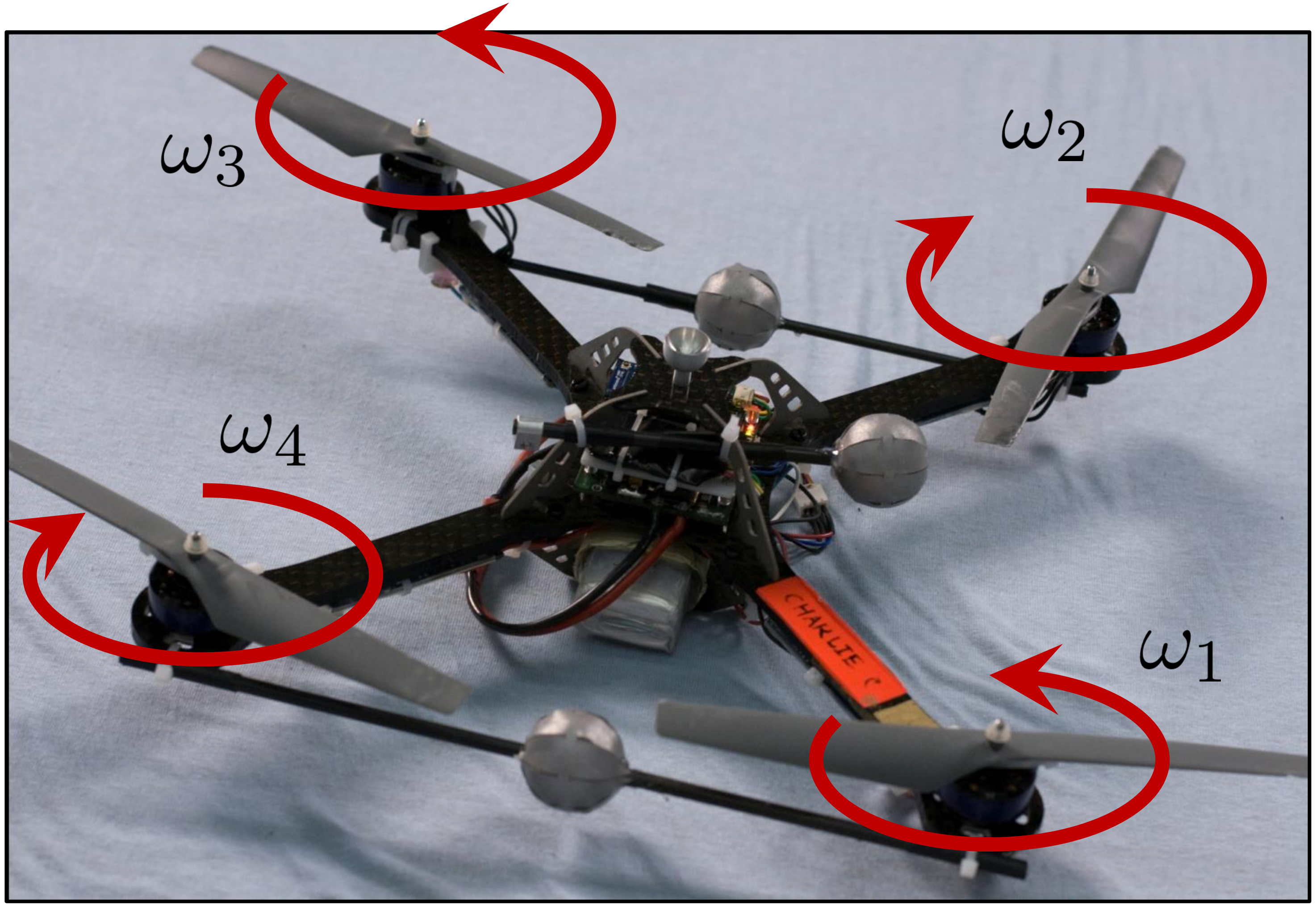
DANCE

youtu.be/NPvGxIBt3Hs

FLIP—up to 1800 degrees/second

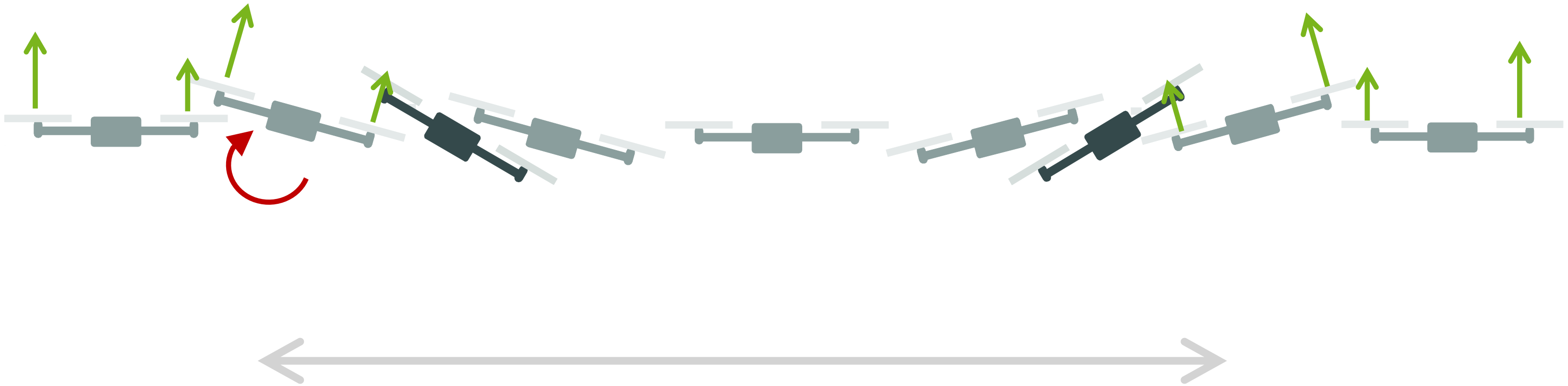
youtu.be/bWExDW9J9sA

YAW



How do you get the robot to steer/yaw?

TRANSLATION



QUESTION

The robot has <u>**6 degrees of freedom**</u>!

1. How many **different ways** can you **rotate or translate** the robot?

2. How many of them are **independent given** that the robot has **4 motors**?

1. Why quadrotors?

2. Why control?

2. Why control?

WHAT IS CONTROLS?

Controls enables **self-regulating systems**

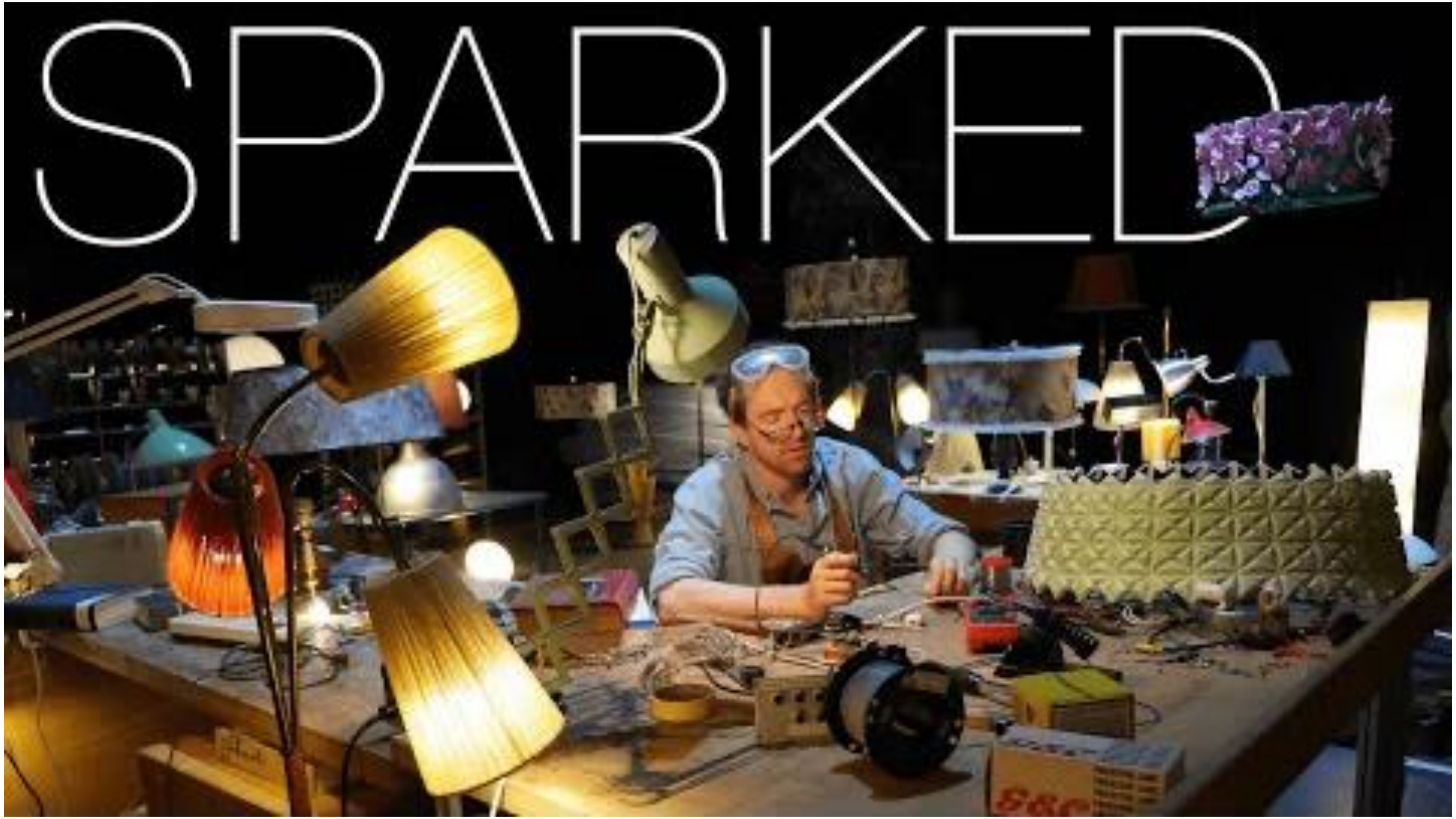
Despite disturbances

That is, machines to achieve a task on their own

DYNAMIC RESPONSE

youtu.be/nQ2ziVW6kts

SPARKED

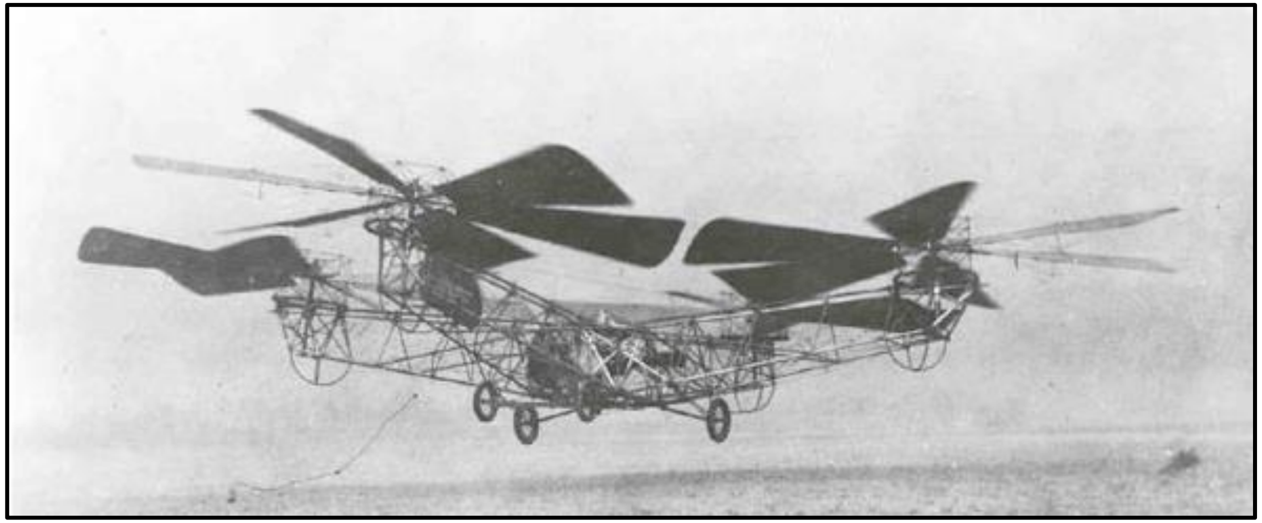


youtu.be/6C8OJsHfmpI

QUADROTOR HISTORY

First successful flight in **1923**

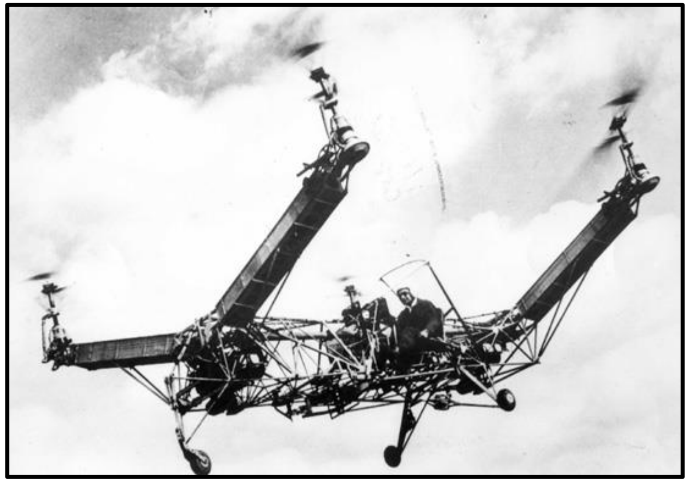
- Design by **George De Bothezat**
- Vertical take-off and landing (**VTOL**)
- **Poor stability, control**



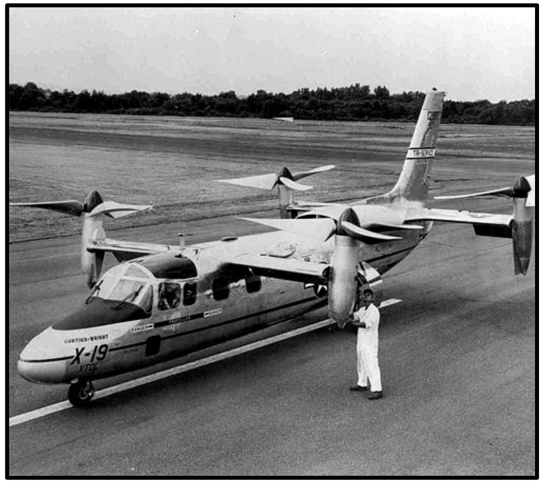
De Bothezat Quadrotor (1923)

Later attempts

- **Used simple feedback control**

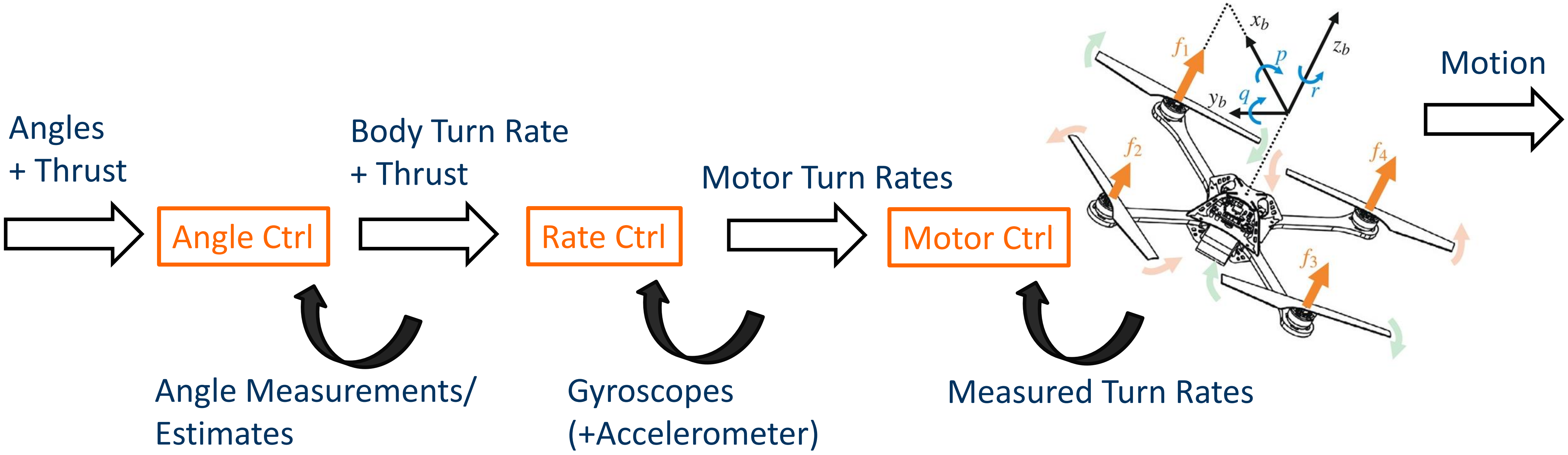


Convertawings Model "A" (1956)

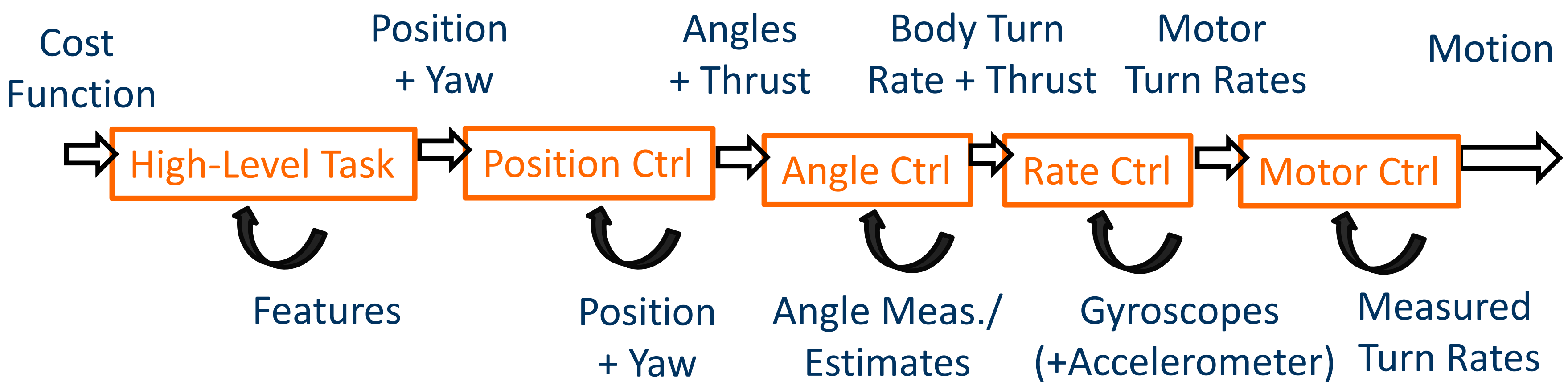


Curtiss-Wright X-19A (1960)

QUADROTOR CONTROL (1 of 2)

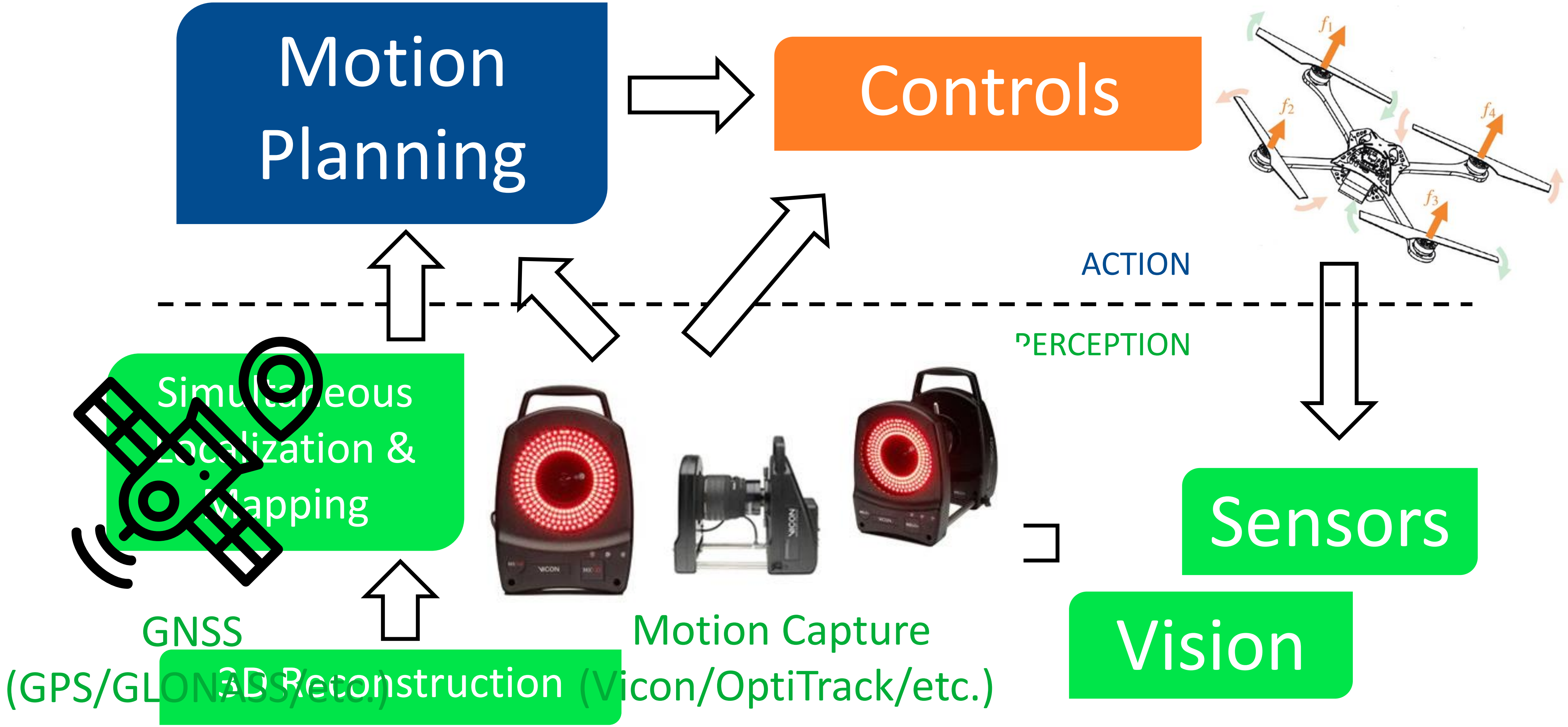


QUADROTOR CONTROL (2 of 2)



Control allows us to focus on the **high-level task**

HOW DOES IT ALL FIT TOGHETER? AN OVERVIEW



My goal for today:

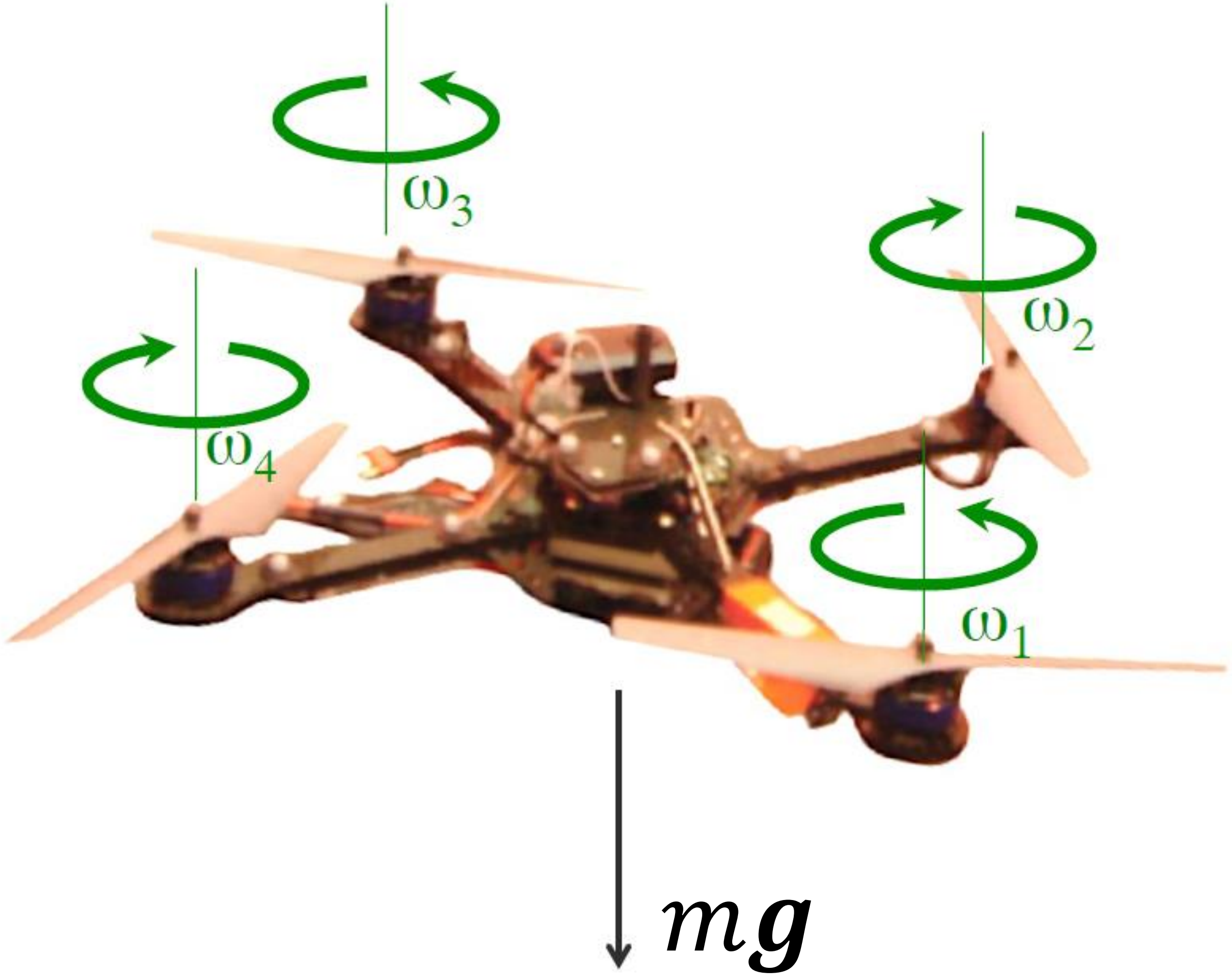
Prepare you to design
your own quadrotor controllers!

OUTLINE

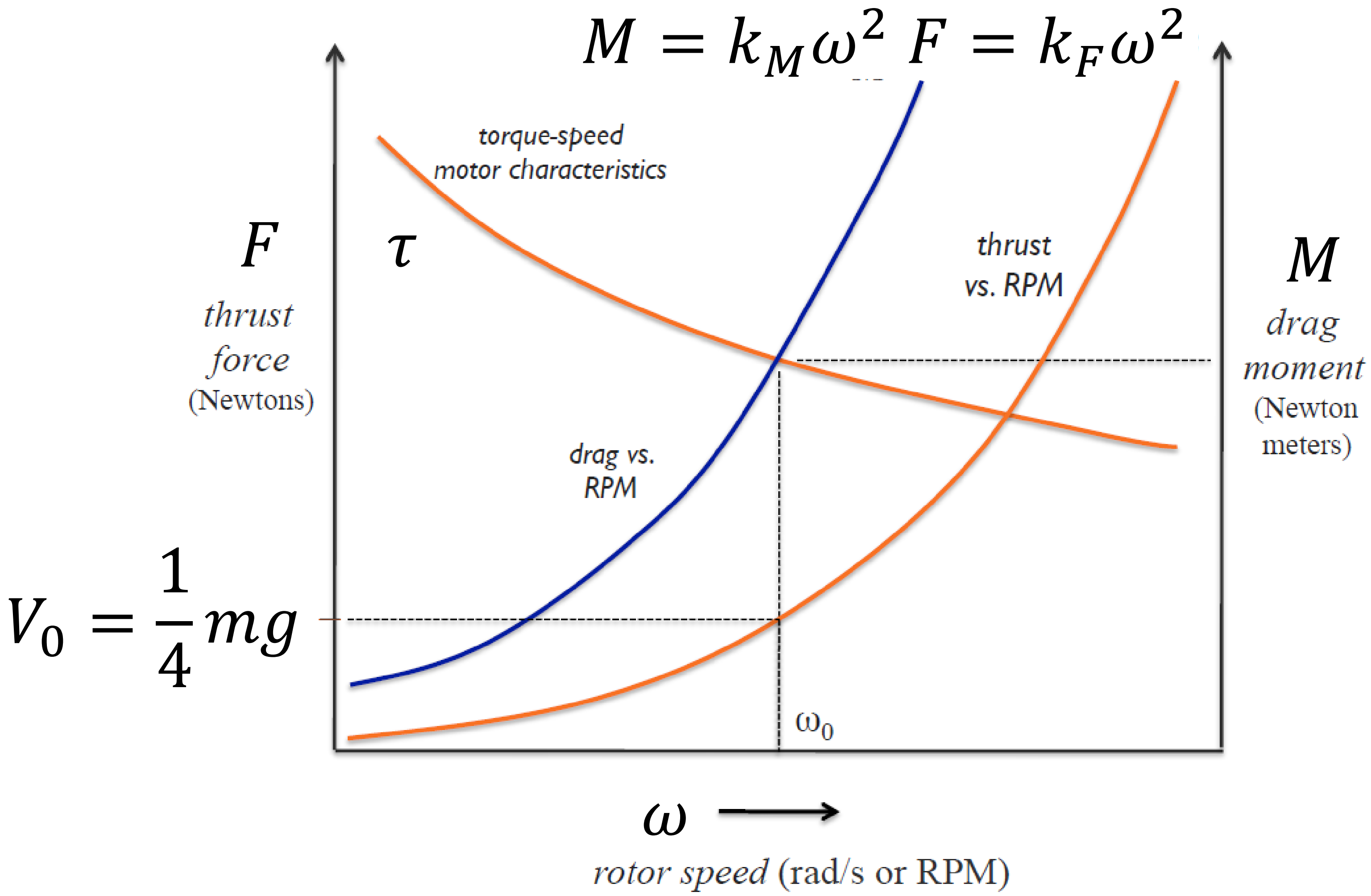
1. **Basic Mechanics**
2. Dynamics & Control of the Vertical Direction (1D)
3. Dynamics & Control in the Vertical Plane (2D)
4. Trajectory Tracking Control (3D)
5. Summary
6. What Can Go Wrong?
7. Learning-Enabled Control

(Several figures/slides are taken from Vijay Kumar's excellent Coursera course on Aerial Robotics)

1. BASIC MECHANICS



1. BASIC MECHANICS: ROTOR PHYSICS



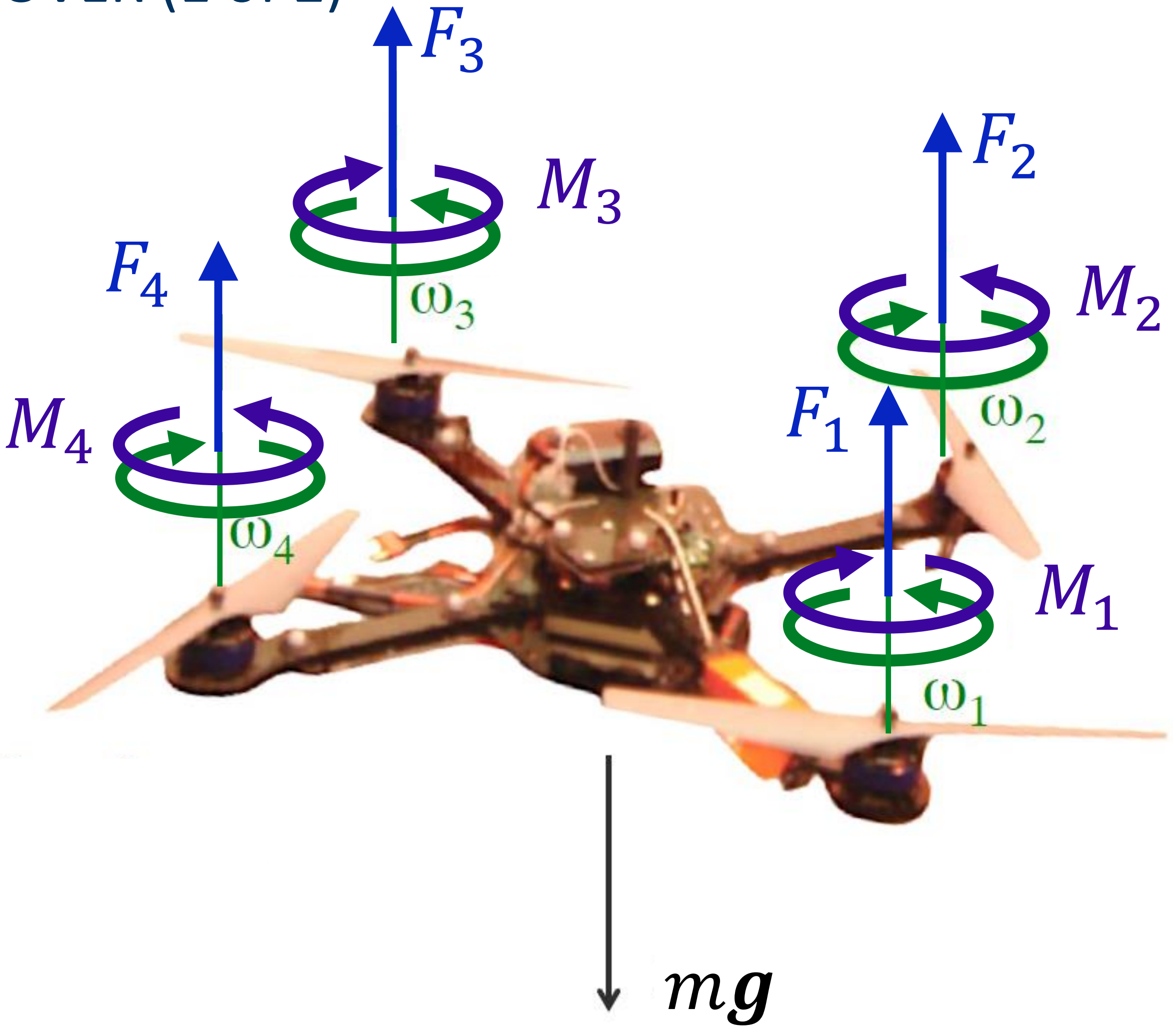
1. BASIC MECHANICS: HOVER (1 of 2)

Motor Speeds

$$k_F \omega_i^2 = \frac{1}{4} mg$$

Motor Torques

$$\tau_i = k_M \omega_i^2$$

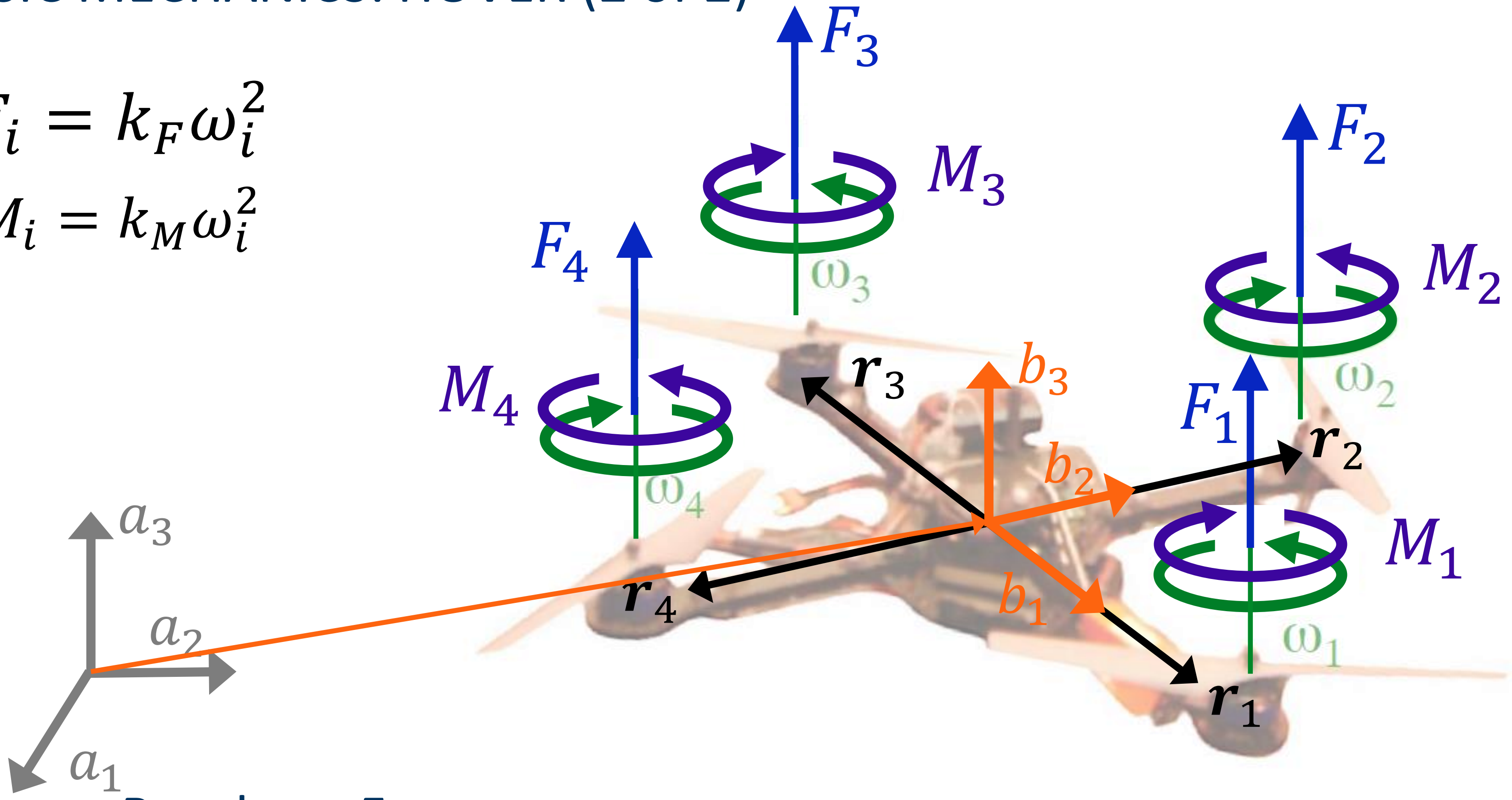


$$F_i = k_F \omega_i^2$$

$$M_i = k_M \omega_i^2$$

$$F_i = k_F \omega_i^2$$

$$M_i = k_M \omega_i^2$$



Resultant Force

$$\boldsymbol{F} = \boldsymbol{F}_1 + \boldsymbol{F}_2 + \boldsymbol{F}_3 + \boldsymbol{F}_4 - mg\boldsymbol{a}_3$$

Resultant Moment

$$\boldsymbol{M} = \boldsymbol{r}_1 \times \boldsymbol{F}_1 + \boldsymbol{r}_2 \times \boldsymbol{F}_2 + \boldsymbol{r}_3 \times \boldsymbol{F}_3 + \boldsymbol{r}_4 \times \boldsymbol{F}_4 + \boldsymbol{M}_1 + \boldsymbol{M}_2 + \boldsymbol{M}_3 + \boldsymbol{M}_4$$

OUTLINE

1. Basic Mechanics
2. **Dynamics & Control of the Vertical Direction (1D)**
3. Dynamics & Control in the Vertical Plane (2D)
4. Trajectory Tracking Control (3D)
5. Summary
6. What Can Go Wrong?
7. Learning-Enabled Control

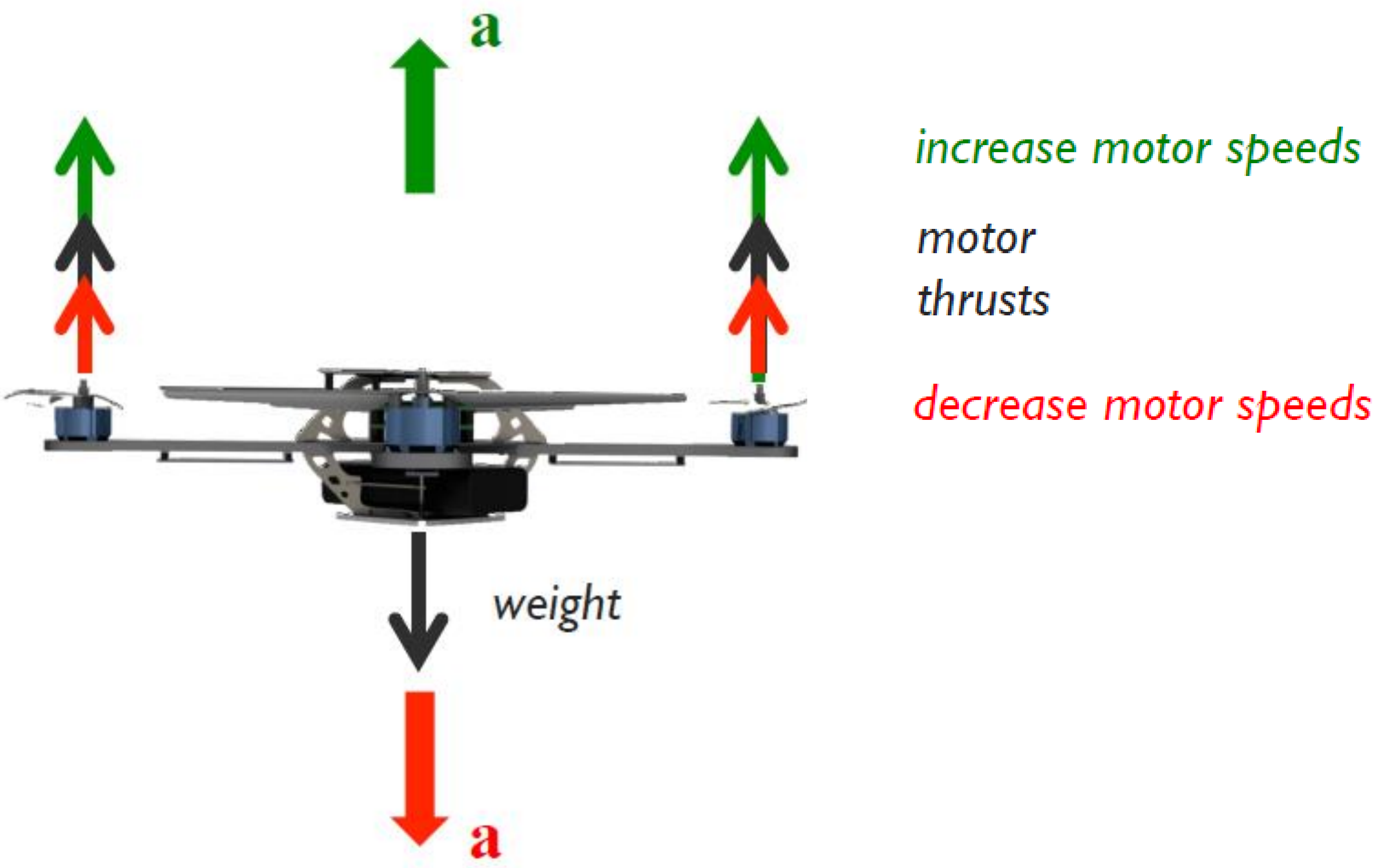
2. DYNAMICS IN THE VERTICAL DIRECTION

Hover:

$$\sum_{i=1}^{4} k_F \omega_i^2 + mg = 0$$

Move:

$$\sum_{i=1}^{4} k_F \omega_i^2 + mg = ma$$



2. CONTROL IN THE VERTICAL DIRECTION (HEIGHT)

x vertical displacement

$$\sum_{i=1}^{4} k_F \omega_i^2 + mg = ma \rightarrow a = \frac{d^2x}{dt^2} = \ddot{x}$$

Define the control input as

$$u = \frac{1}{m}\left[\sum_{i=1}^{4} k_F \omega_i^2 + mg\right]$$

to obtain the second order dynamic system $u = \ddot{x}$.

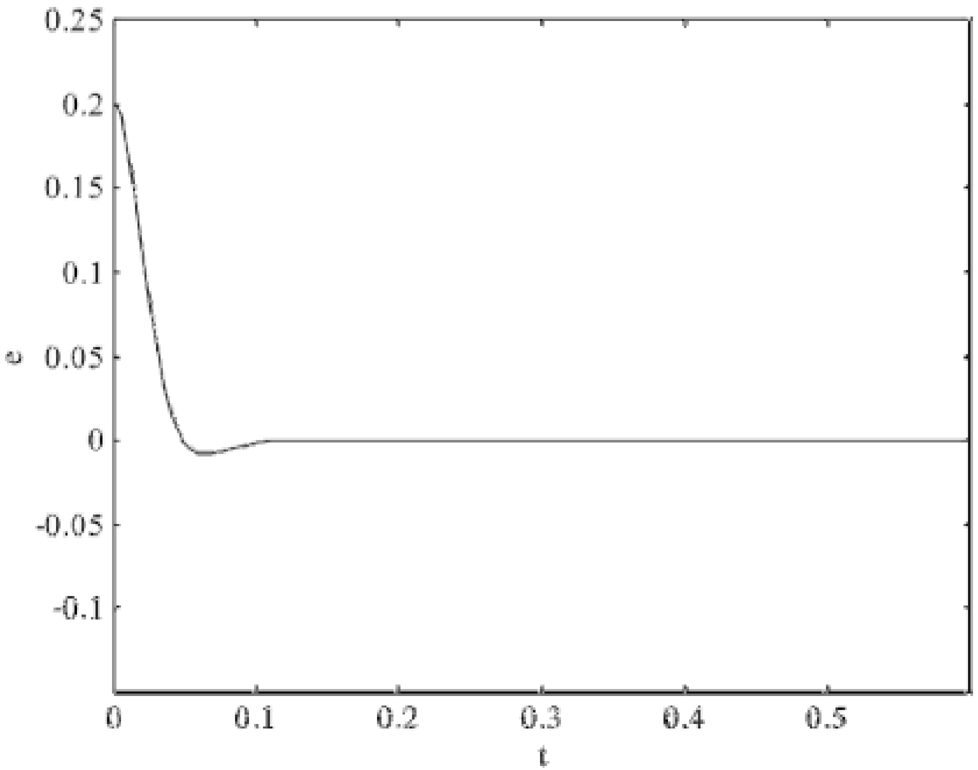
<u>What input drives the robot to the desired position?</u>

2. CONTROL OF A LINEAR, SECOND-ORDER SYSTEM

Problem:

- State, Input $x, u \in \mathbb{R}$
- Plant model $\ddot{x} = u$

Want x to follow the desired trajectory $x^{des}(t)$



General Approach:

Define error $e(t) = x^{des}(t) - x(t)$

Want $e(t)$ to converge exponentially to zero

Strategy:

Find u such that the error converges exponentially to 0

$\ddot{e} + K_v\dot{e} + K_p e = 0 \quad K_p, K_v > 0$

$$u(t) = \ddot{x}^{des}(t) + K_v\dot{e}(t) + K_p e(t)$$

Feedforward Derivative Proportional

PD Control

$$u(t) = \ddot{x}^{des}(t) + K_v \dot{e}(t) + K_p e(t)$$

Proportional control acts like a spring (capacitance) response.
Derivative control is a viscous dashpot (resistance) response.
Large derivative gain makes the system overdamped (slow convergence).

PID Control

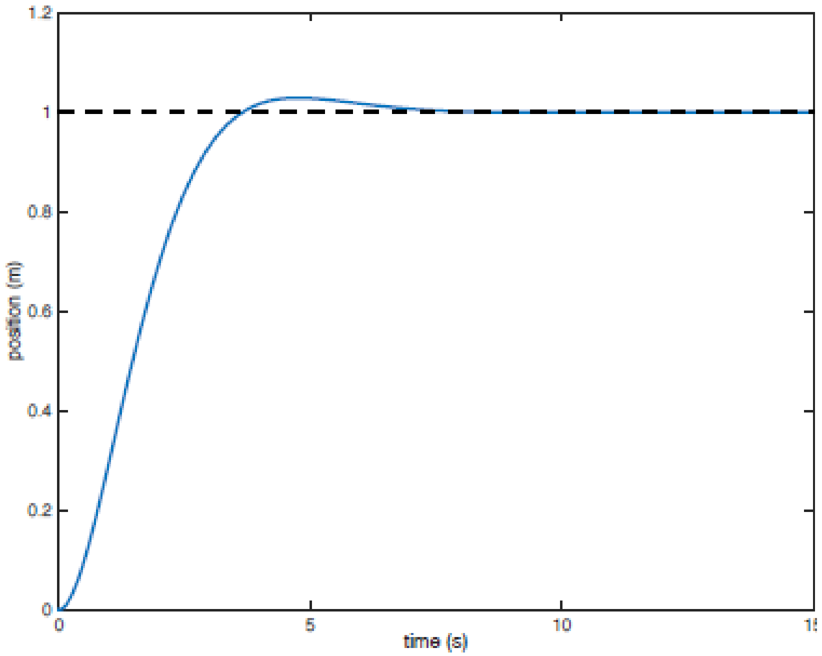
Often advantageous in the presence of disturbances (e.g. wind) or modeling errors (e.g. unknown mass).

$$u(t) = \ddot{x}^{des}(t) + K_v \dot{e}(t) + K_p e(t) + K_i \int_0^t e(\tau) d\tau$$

Integral control makes the steady-state error go to zero.
PID control generates a third-order closed-loop system.

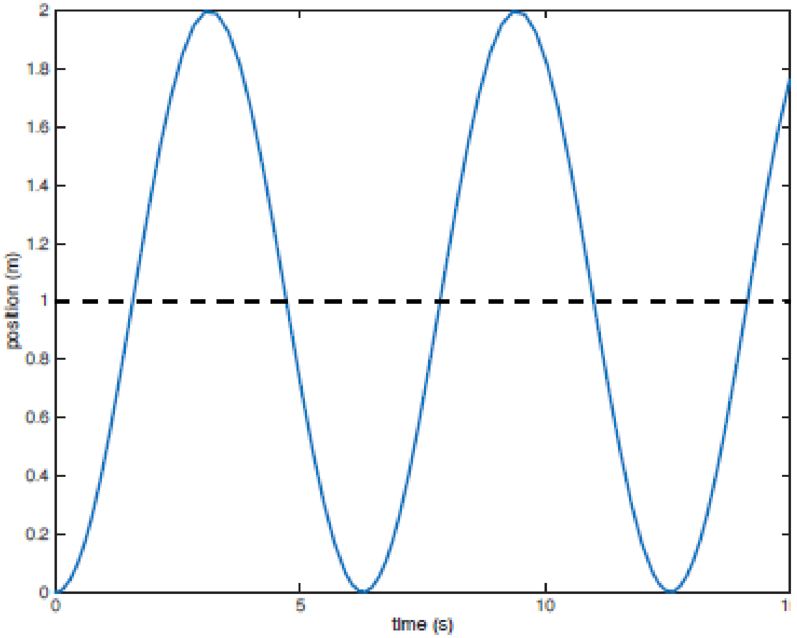
Integral

2. EFFECTS OF DIFFERENT GAINS



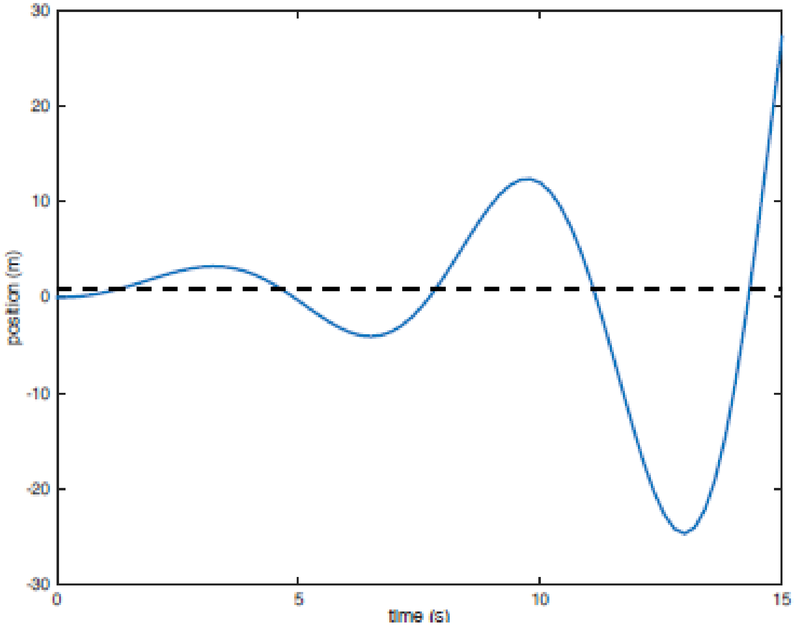
Stable

$$K_p, K_v > 0$$



Marginally Stable

$$K_p > 0, \qquad K_v = 0$$



Unstable

$$K_p \text{ or } K_v < 0$$

2. DIFFERENT PARAMETRIZATIONS

$$\ddot{x} = u,\, u(t) \coloneqq K_p(x^{des} - x) + K_v(\dot{x}^{des} - \dot{x}) + \ddot{x}^{des}$$

$$\implies \quad \ddot{e} + K_v\dot{e} + K_p e = 0$$

$$\implies \quad \boxed{\ddot{e} + 2\zeta\omega_n\dot{e} + \omega_n^2 e = 0}$$

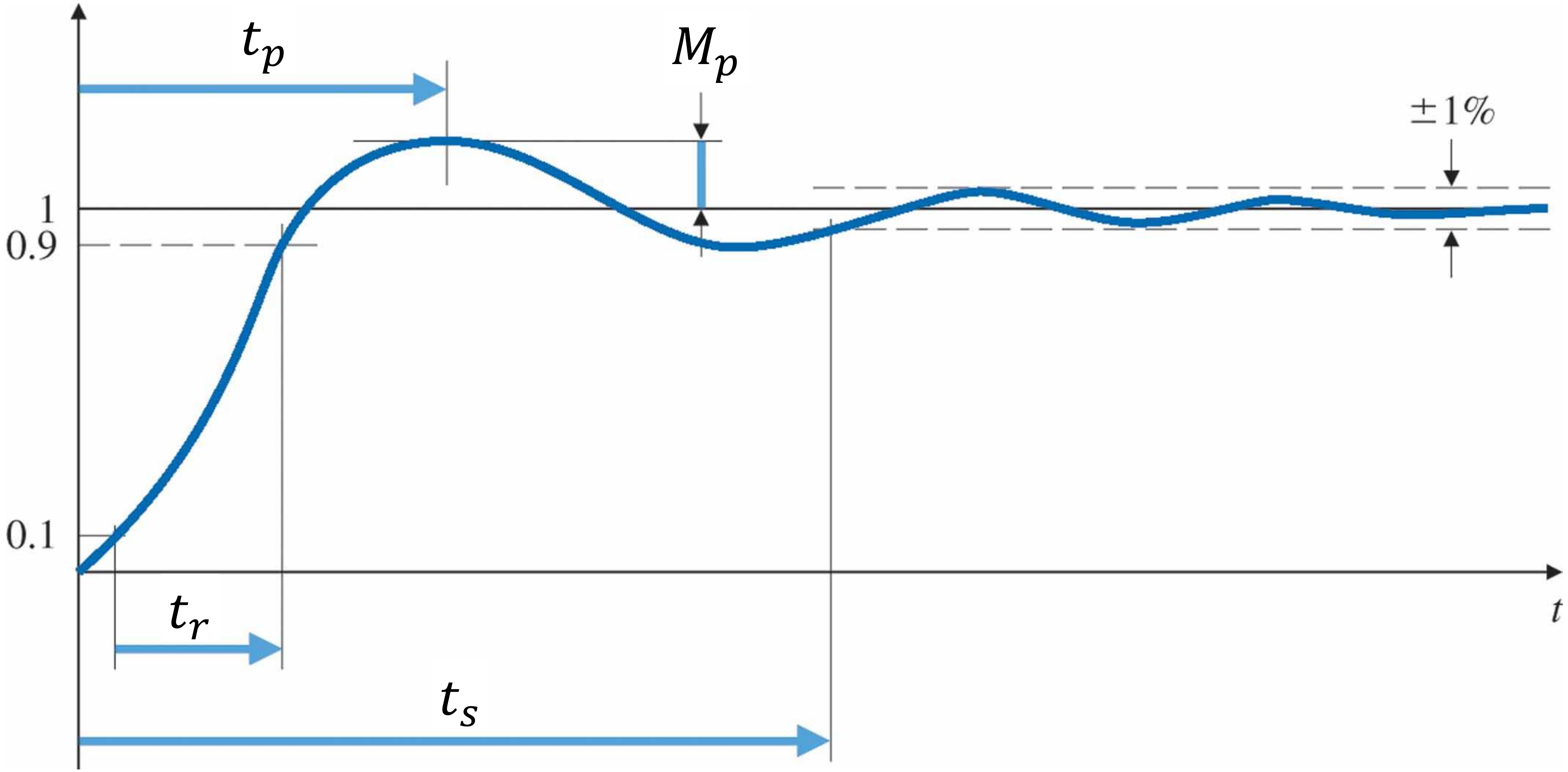
Feedforward term

Intuitive parameterization:

- Damping ratio: $\zeta \in [0.7..1]$
- Natural frequency, related to rise time (10-90%): $\omega_n \approx \frac{t_r}{1.8}$
- Settling time: $t_s \approx \frac{4.6}{\zeta\omega_n}$

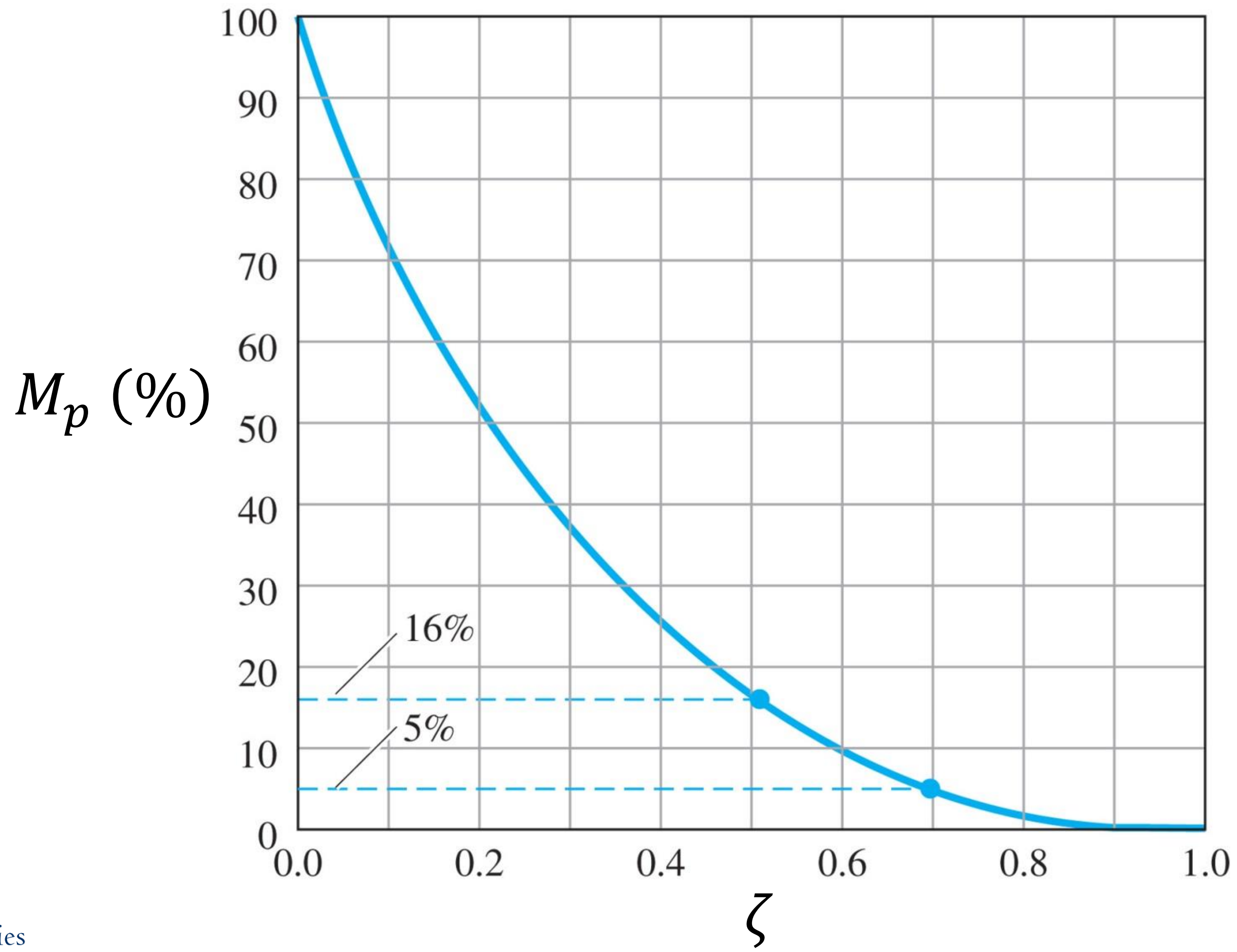
2. SECOND-ORDER CHARACTERISTICS (1 of 3)

Definition of <u>rise time</u> t_r, <u>peak time</u> t_p, <u>settling time</u> t_s, and <u>overshoot</u> M_p:



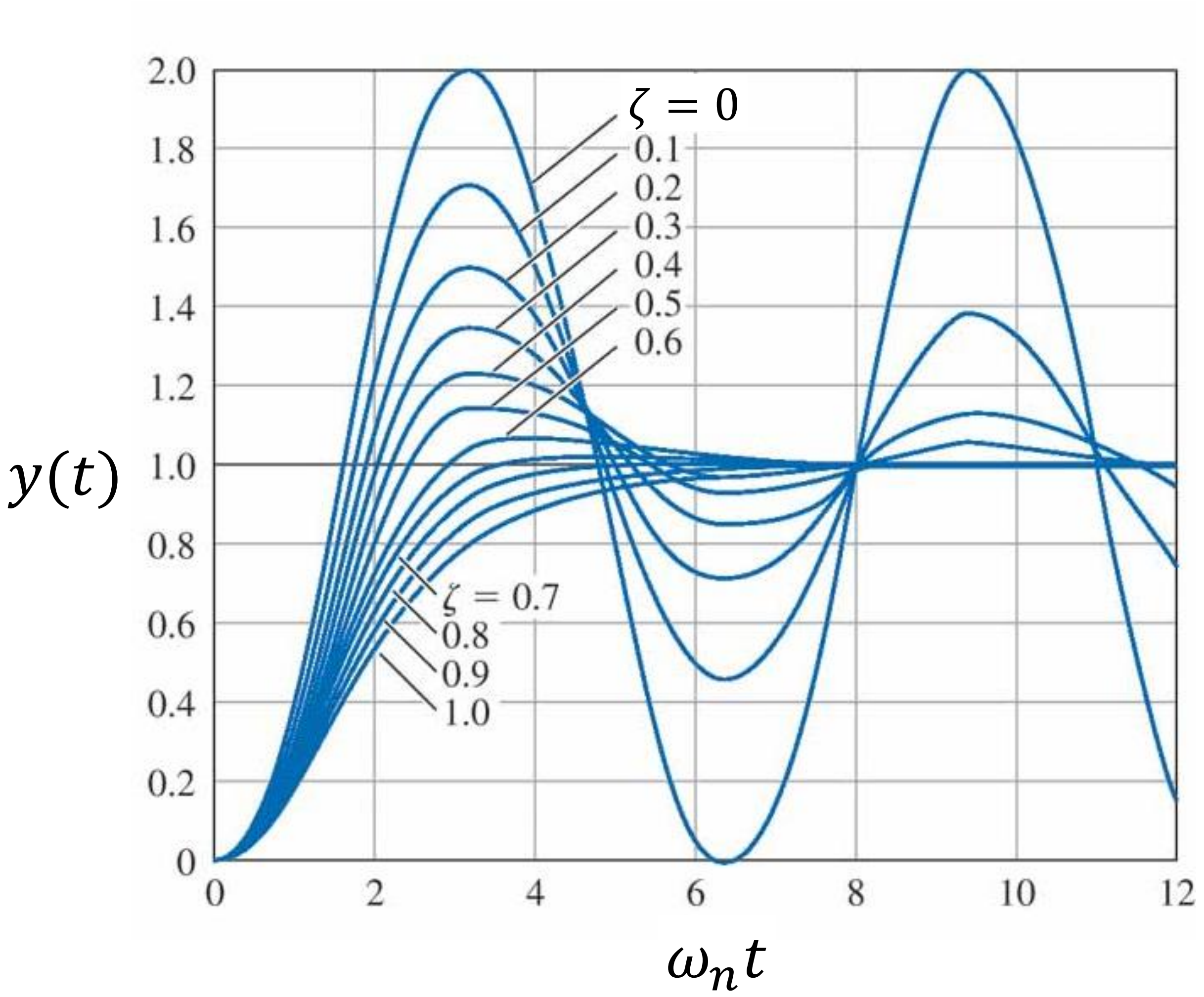
2. SECOND-ORDER CHARACTERISTICS (2 of 3)

Overshoot versus damping ratio for the second-order system:



2. SECOND-ORDER CHARACTERISTICS (3 of 3)

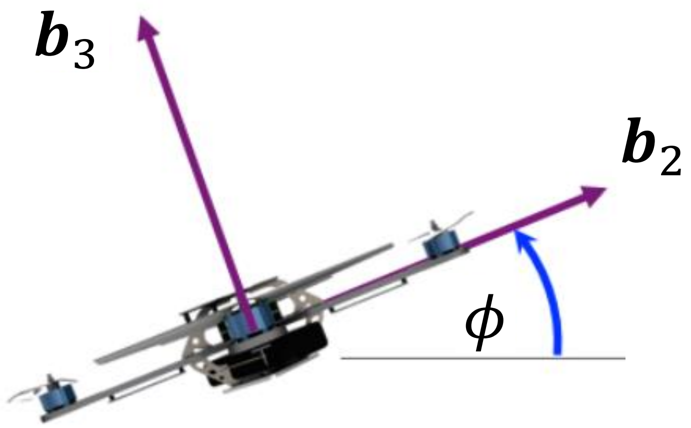
Step responses of second-order systems versus ζ



OUTLINE

1. Basic Mechanics
2. Dynamics & Control of the Vertical Direction (1D)
3. **Dynamics & Control in the Vertical Plane (2D)**
4. Trajectory Tracking Control (3D)
5. Summary
6. What Can Go Wrong?
7. Learning-Enabled Control

3. PLANAR (2D) QUADROTOR MODEL



$$\begin{bmatrix} \ddot{y} \\ \ddot{z} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 \\ -g \\ 0 \end{bmatrix} + \begin{bmatrix} -\frac{1}{m}\sin\phi & 0 \\ \frac{1}{m}\sin\phi & 0 \\ 0 & \frac{1}{I_{zz}} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

State space:

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} y \\ z \\ \phi \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \end{bmatrix}$$

$$\dot{x} = \begin{bmatrix} \dot{y} \\ \dot{z} \\ \dot{\phi} \\ 0 \\ -g \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -\frac{1}{m}\sin\phi & 0 \\ \frac{1}{m}\sin\phi & 0 \\ 0 & \frac{1}{I_{zz}} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

3. LINEARIZED DYNAMIC MODEL

- Equations of motion:

$$\ddot{y} = -\frac{u_1}{m}\sin(\phi)$$

$$\ddot{z} = -g + \frac{u_1}{m}\cos(\phi)$$

$$\ddot{\phi} = \frac{u_2}{I_{zz}}$$

Dynamics are nonlinear

- Equilibrium hover configuration:

$$y_0, z_0, \phi_0 = 0, \qquad u_{1,0} = mg, \qquad u_{2,0} = 0$$

- Linearized dynamics around hover:

$$\ddot{y} = -g\phi$$

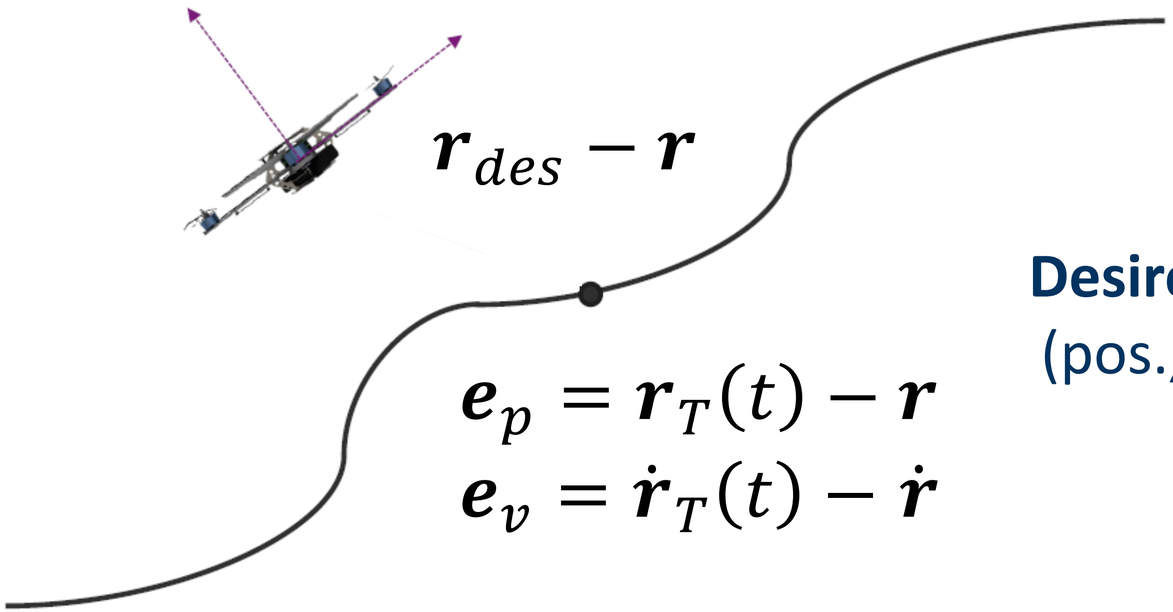
$$\ddot{z} = -g + \frac{u_1}{m}$$

$$\ddot{\phi} = \frac{u_2}{I_{zz}}$$

3. 2D TRAJECTORY TRACKING

Given the differentiable trajectory $\boldsymbol{r}_T(t), \dot{\boldsymbol{r}}_T(t), \ddot{\boldsymbol{r}}_T(t)$

$$\boldsymbol{r}_T(t) = \begin{bmatrix} y(t) \\ z(t) \end{bmatrix}$$



$$\boldsymbol{r}_{des} - \boldsymbol{r}$$

Desired trajectory (pos., vel., accel.)

$$\boldsymbol{e}_p = \boldsymbol{r}_T(t) - \boldsymbol{r}$$
$$\boldsymbol{e}_v = \dot{\boldsymbol{r}}_T(t) - \dot{\boldsymbol{r}}$$

Want error to decay exponentially to 0:

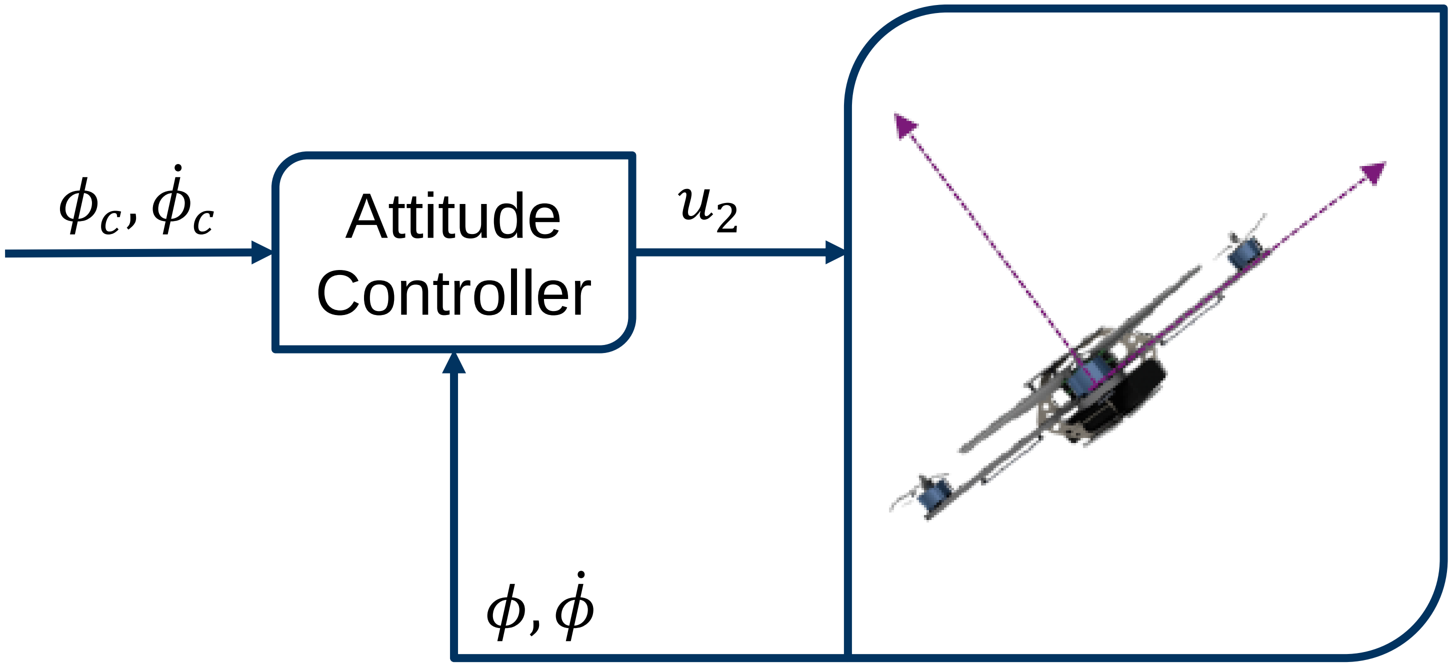
$$(\ddot{\boldsymbol{r}}_T(t) - \ddot{\boldsymbol{r}}_{\boldsymbol{c}}) + K_{d,x}\boldsymbol{e}_v + K_{p,x}\,\boldsymbol{e}_p = 0$$

Commanded acceleration, calculated by the controller

3. NESTED CONTROL STRUCTURE

Want error to decay exponentially to 0:

$$(\ddot{\phi}_c - \ddot{\phi}) + k_{p,\phi}(\phi_c - \phi) + k_{d,\phi}(\dot{\phi}_c - \dot{\phi}) = 0$$



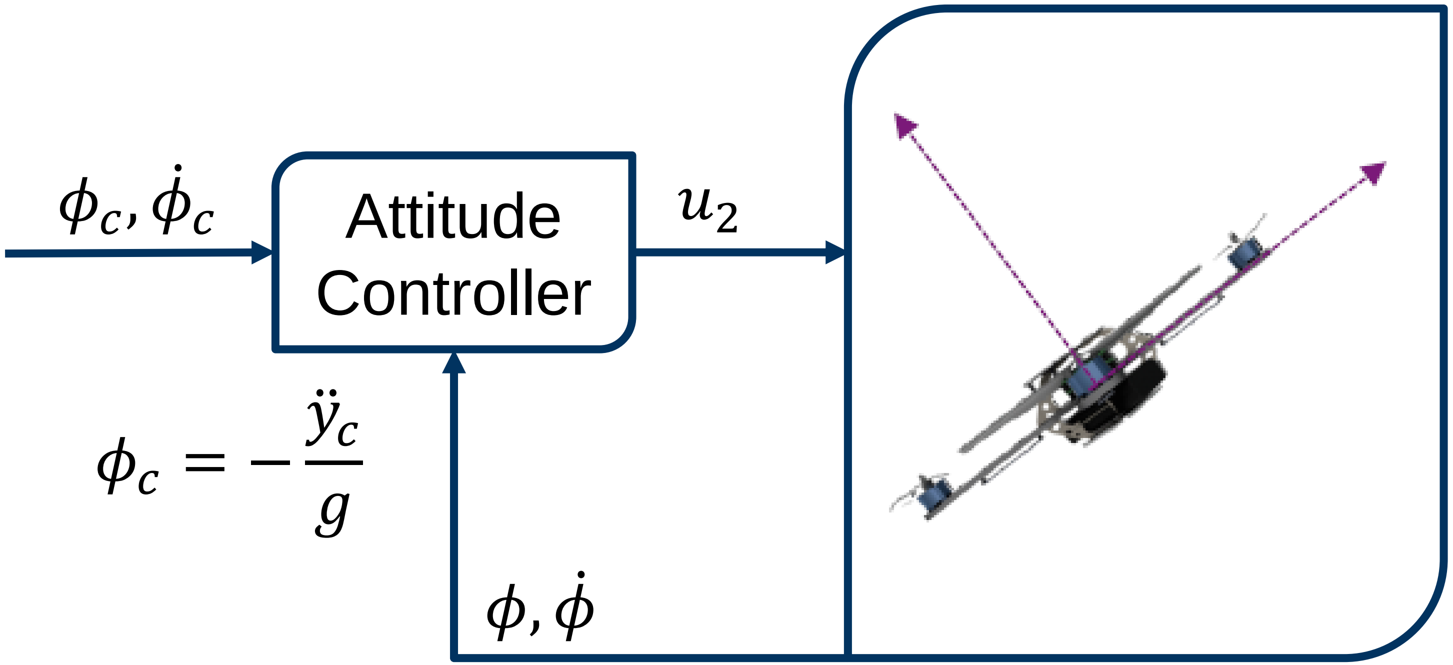
Linearized dynamics:

$$\ddot{\phi} = \frac{u_2}{I_{xx}}$$

$$u_2 = I_{xx}(\ddot{\phi}_c + k_{p,\phi}(\phi_c - \phi) + k_{d,\phi}(\dot{\phi}_c - \dot{\phi}))$$

3. NESTED CONTROL STRUCTURE

$$\phi_c = -\frac{1}{g}\left(\ddot{y}_{des} + k_{d,y}\left(\dot{y}_{des} - \dot{y}\right) + k_{p,y}\left(y_{des} - y\right)\right)$$



$$\phi_c = -\frac{\ddot{y}_c}{g}$$

Linearized dynamics:

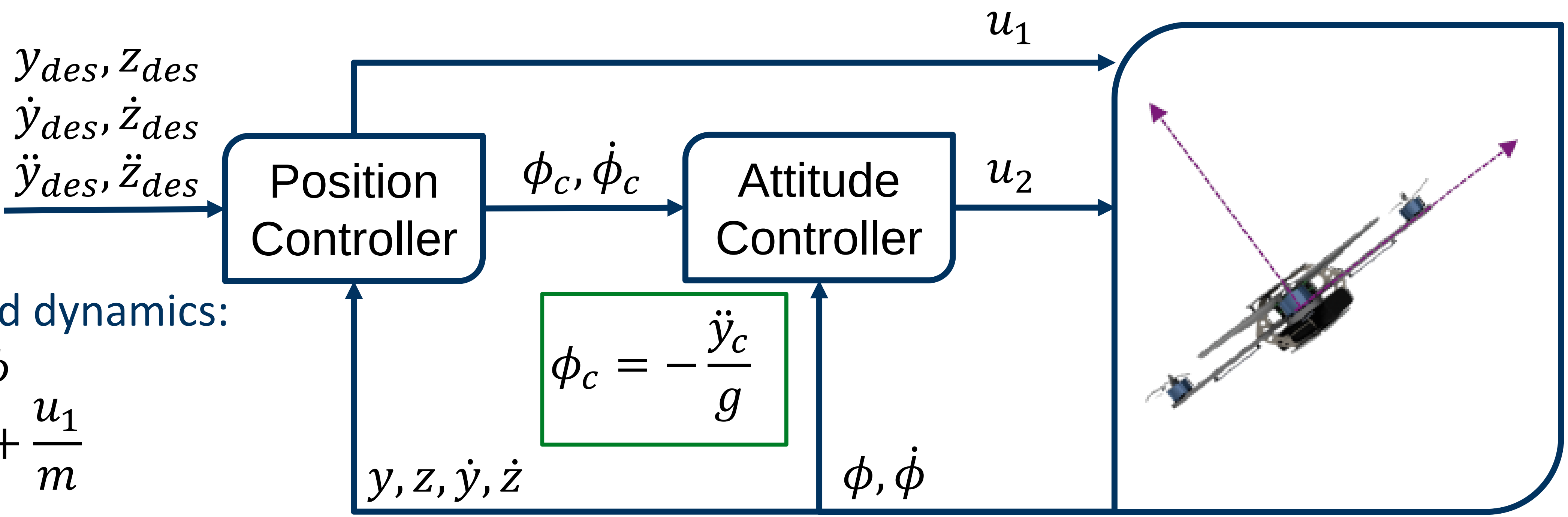
$$\ddot{y} = -g\phi$$

$$\ddot{\phi} = \frac{u_2}{I_{xx}}$$

$$u_2 = I_{xx}\left(\ddot{\phi}_c + k_{p,\phi}\left(\phi_c - \phi\right) + k_{d,\phi}\left(\dot{\phi}_c - \dot{\phi}\right)\right)$$

3. NESTED CONTROL STRUCTURE

$$\boxed{u_1 = m(g + \ddot{z}_{des} + k_{d,z}(\dot{z}_{des} - \dot{z}) + k_{p,z}(z_{des} - z))}$$



Linearized dynamics:

$$\ddot{y} = -g\phi$$

$$\ddot{z} = -g + \frac{u_1}{m}$$

$$\ddot{\phi} = \frac{u_2}{I_{xx}}$$

$$\boxed{u_2 = I_{xx}(\ddot{\phi}_c + k_{p,\phi}(\phi_c - \phi) + k_{d,\phi}(\dot{\phi}_c - \dot{\phi}))}$$

3. CONTROL EQUATIONS

$$u_1 = m(g + \ddot{z}_{des} + k_{d,z}(\dot{z}_{des} - \dot{z}) + k_{p,z}(z_{des} - z))$$

$$u_2 = I_{xx}(\ddot{\phi}_c + k_{p,\phi}(\phi_c - \phi) + k_{d,\phi}(\dot{\phi}_c - \dot{\phi}))$$

Simplification with no performance reduction

$$u_2 = k_{p,\phi}(\phi_c - \phi) + k_{d,\phi}(\dot{\phi}_c - \dot{\phi})$$

$$\phi_c = -\frac{1}{g}(\ddot{y}_{des} + k_{d,y}(\dot{y}_{des} - \dot{y}) + k_{p,y}(y_{des} - y)$$

3. CONTROL EQUATIONS

$$u_1 = m(g + \ddot{z}_{des} + k_{d,z}(\dot{z}_{des} - \dot{z}) + k_{p,z}(z_{des} - z))$$

$$u_2 = k_{p,\phi}(\phi_c - \phi) + k_{d,\phi}(\dot{\phi}_c - \dot{\phi})$$

$$\phi_c = -\frac{1}{g}(\ddot{y}_{des} + k_{d,y}(\dot{y}_{des} - \dot{y}) + k_{p,y}(y_{des} - y)$$

OUTLINE

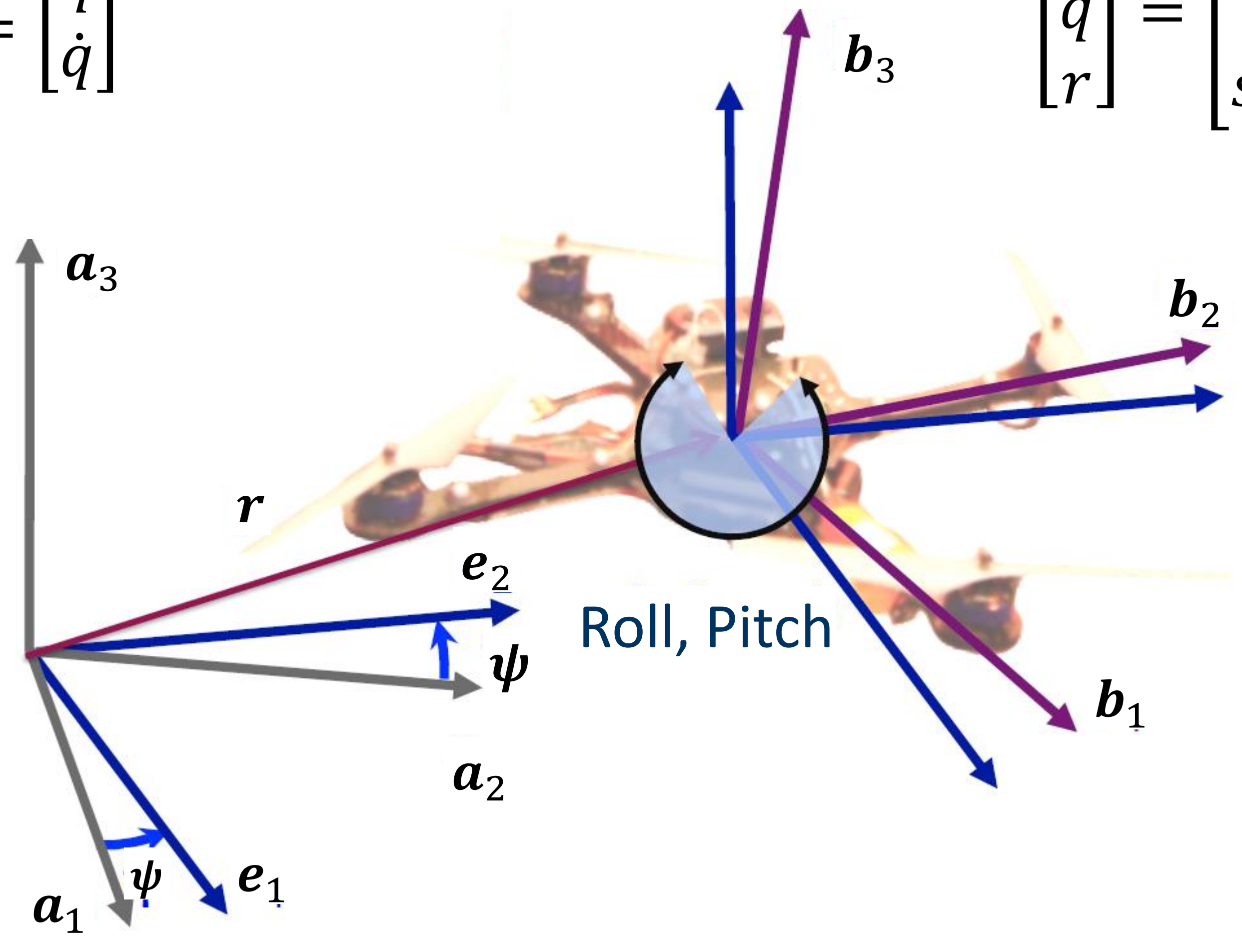
1. Basic Mechanics
2. Dynamics & Control of the Vertical Direction (1D)
3. Dynamics & Control in the Vertical Plane (2D)
4. **Trajectory Tracking Control (3D)**
5. Summary
6. What Can Go Wrong?
7. Learning-Enabled Control

4. 3D VEHICLE MODEL

$$q = \begin{bmatrix} x \\ y \\ z \\ \phi \\ \theta \\ \psi \end{bmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} \boldsymbol{r} \\ \text{Pitch} \\ \text{Roll} \\ \text{Yaw} \end{matrix}, \quad x = \begin{bmatrix} q \\ \dot{q} \end{bmatrix}$$

Angular velocities components in b

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} c\theta & 0 & -c\phi s\theta \\ 0 & 1 & s\phi \\ s\theta & 0 & c\phi c\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

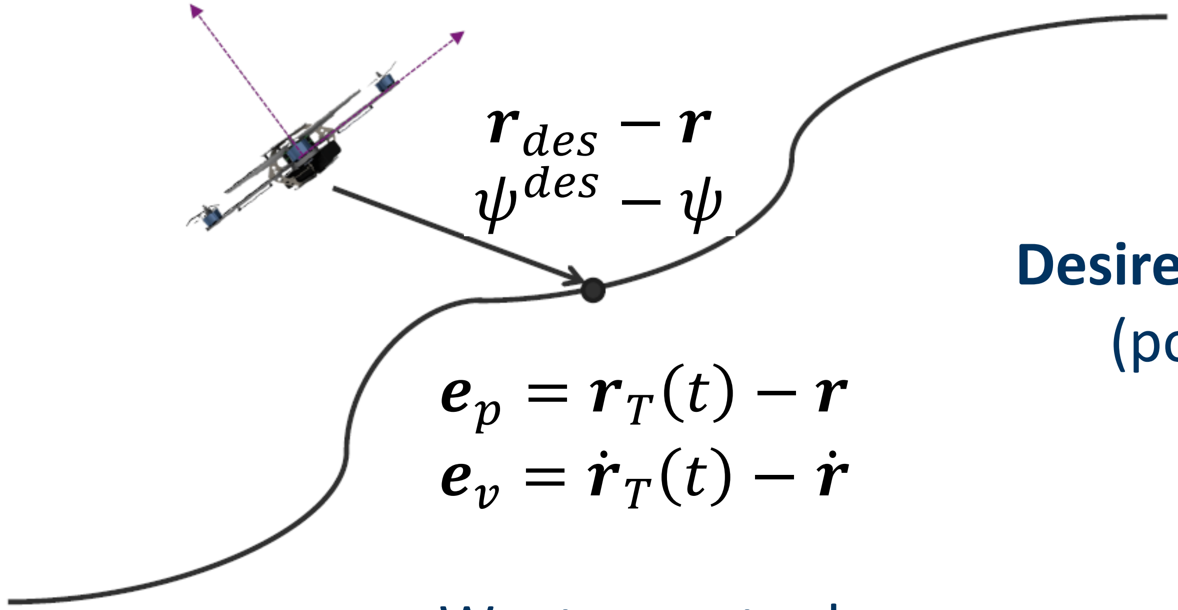


4. CONTROL TASK

Given $\boldsymbol{r}_T(t), \dot{\boldsymbol{r}}_T(t), \ddot{\boldsymbol{r}}_T(t)$ (differentiable)

$\boldsymbol{r}^{des}(t), \dot{\boldsymbol{r}}^{des}(t), \ddot{\boldsymbol{r}}^{des}(t)$

$\psi^{des}(t), \dot{\psi}^{des}(t), \ddot{\psi}^{des}(t)$



$$\boldsymbol{r}_T(t) = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \\ \psi(t) \end{bmatrix}$$

Desired trajectory (pos., yaw)

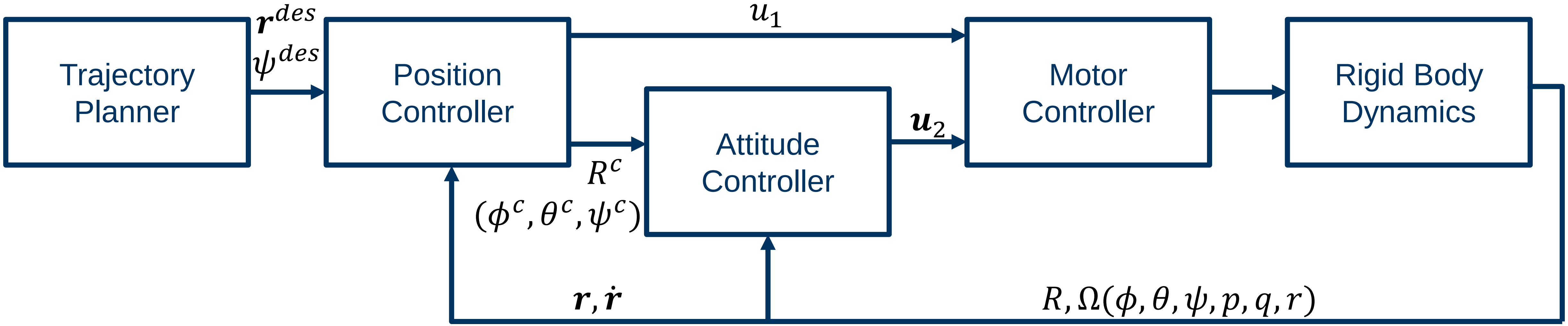
$$\boldsymbol{e}_p = \boldsymbol{r}_T(t) - \boldsymbol{r}$$
$$\boldsymbol{e}_v = \dot{\boldsymbol{r}}_T(t) - \dot{\boldsymbol{r}}$$

Want error to decay exponentially to 0:

$$(\ddot{\boldsymbol{r}}_T(t) - \ddot{\boldsymbol{r}}_{\boldsymbol{c}}) + K_{d,x}\boldsymbol{e}_v + K_{p,x}\,\boldsymbol{e}_p = 0$$

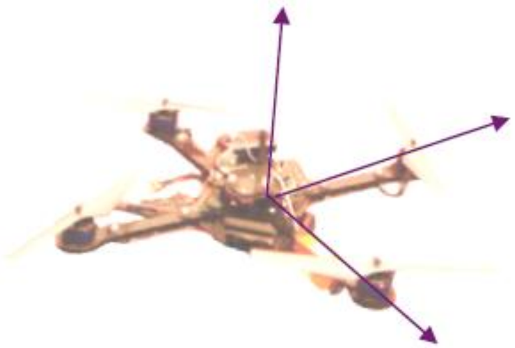
Commanded acceleration, calculated by the controller

4. NESTED CONTROL



$$m\ddot{\mathbf{r}} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ \underbrace{\boxed{F_1 + F_2 + F_3 + F_4}}_{u_1} \end{bmatrix}$$

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \underbrace{\boxed{\begin{bmatrix} L(F_2 - F_4) \\ L(F_3 - F_1) \\ M_1 - M_2 + M_3 - M_4 \end{bmatrix}}}_{\mathbf{u}_2} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$



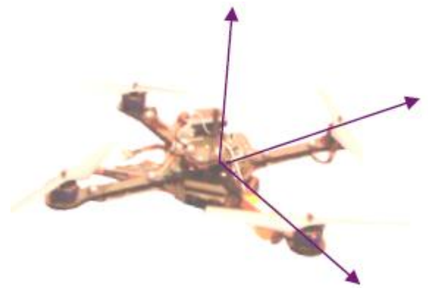
Linearize the dynamics
at the hover configuration

$$(u_1 \sim mg, \theta \sim 0, \phi \sim 0, \psi \sim \psi_0)$$
$$(u_2 \sim 0, p \sim 0, q \sim 0, r \sim 0)$$

$$m\ddot{\boldsymbol{r}} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ F_1 + F_2 + F_3 + F_4 \end{bmatrix}$$

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} L(F_2 - F_4) \\ L(F_3 - F_1) \\ M_1 - M_2 + M_3 - M_4 \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

4. HOVER CONTROL (2 of 5)



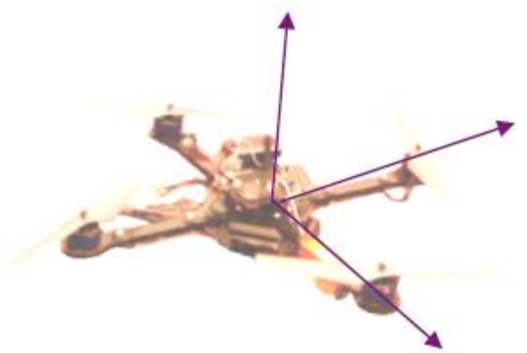
$$m\ddot{\boldsymbol{r}} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ \underbrace{\boxed{F_1 + F_2 + F_3 + F_4}}_{u_1} \end{bmatrix}$$

Linearization

$$(u_1 \sim mg, \theta \sim 0, \phi \sim 0, \psi \sim \psi_0)$$

$$\ddot{r}_1 = \ddot{x} = g(\theta \cos\psi + \phi \sin\psi)$$
$$\ddot{r}_2 = \ddot{y} = g(\theta \sin\psi - \phi \cos\psi)$$

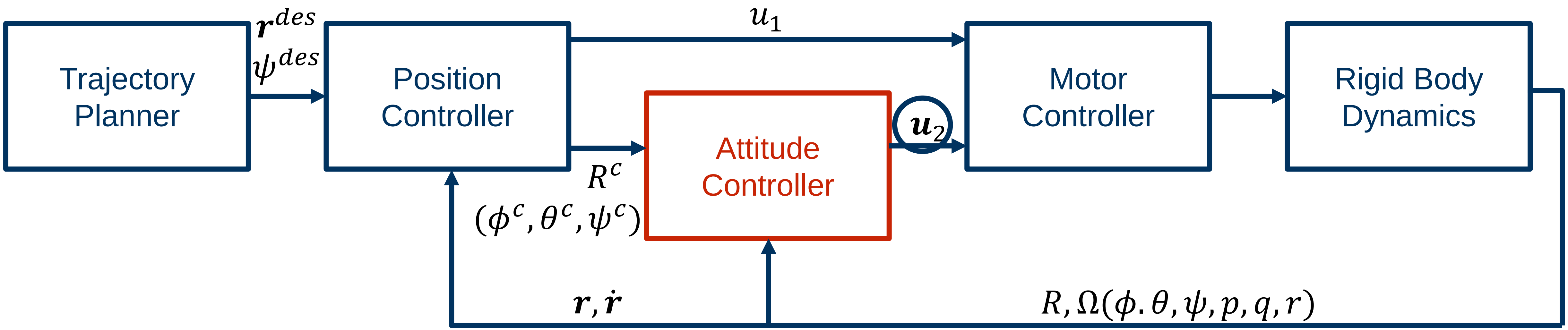
4. HOVER CONTROL (3 of 5)



$$(u_2 \sim 0, p \sim 0, q \sim 0, r \sim 0)$$

$$I\begin{bmatrix}\dot{p}\\ \dot{q}\\ \dot{r}\end{bmatrix} = \underbrace{\begin{bmatrix}L(F_2 - F_4)\\ L(F_3 - F_1)\\ M_1 - M_2 + M_3 - M_4\end{bmatrix}}_{\boldsymbol{u}_2} - \cancelto{0}{\begin{bmatrix}p\\ q\\ r\end{bmatrix} \times I\begin{bmatrix}p\\ q\\ r\end{bmatrix}}$$

4. HOVER CONTROL (4 of 5)

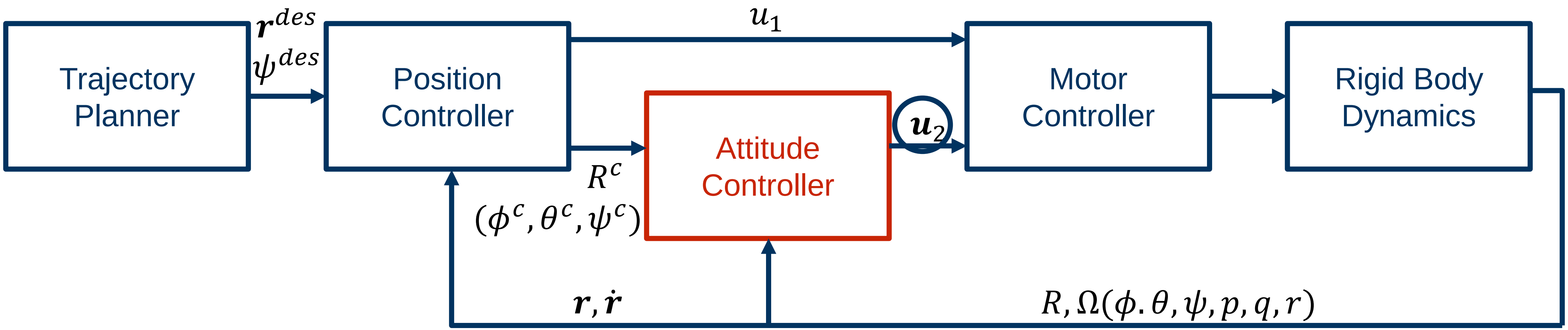


$$\boldsymbol{u}_2 = \begin{bmatrix} k_{p,\phi}(\phi_c - \phi) + k_{d,\phi}(p_c - p) \\ k_{p,\theta}(\theta_c - \theta) + k_{d,\theta}(q_c - q) \\ k_{p,\psi}(\psi_c - \psi) + k_{d,\psi}(r_c - r) \end{bmatrix}$$

Using linearized dynamics:

$$\ddot{r}_1 = \ddot{x} = g(\theta \cos\psi + \phi \sin\psi)$$
$$\ddot{r}_2 = \ddot{y} = g(\theta \sin\psi - \phi \cos\psi)$$

4. HOVER CONTROL (4 of 5)



$$\boldsymbol{u}_2 = \begin{bmatrix} k_{p,\phi}(\phi_c - \phi) + k_{d,\phi}(p_c - p) \\ k_{p,\theta}(\theta_c - \theta) + k_{d,\theta}(q_c - q) \\ k_{p,\psi}(\psi_c - \psi) + k_{d,\psi}(r_c - r) \end{bmatrix}$$

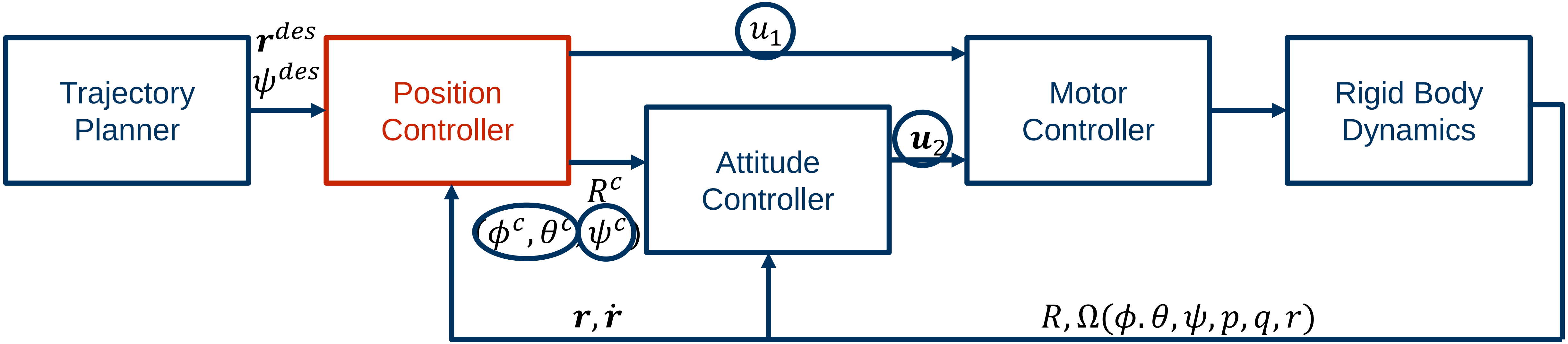
Using linearized dynamics

$$\phi_c = \frac{1}{g}(\ddot{r}_{1,c} \sin\psi_{des} - \ddot{r}_{2,c} \cos\psi_{des})$$

$$\theta_c = \frac{1}{g}(\ddot{r}_{1,c} \cos\psi_{des} - \ddot{r}_{2,c} \sin\psi_{des})$$

$$\psi_c = \psi^{des}$$

4. HOVER CONTROL (5 of 5)



$$\boxed{u_1 = m(g + \ddot{r}_{3,c})}$$

Specified

$$(\ddot{r}_{i,des} - \ddot{r}_{i,c}) + k_{d,i}(\dot{r}_{i,des} - \dot{r}_i) + k_{p,i}(r_{i,des} - r_i) = 0$$

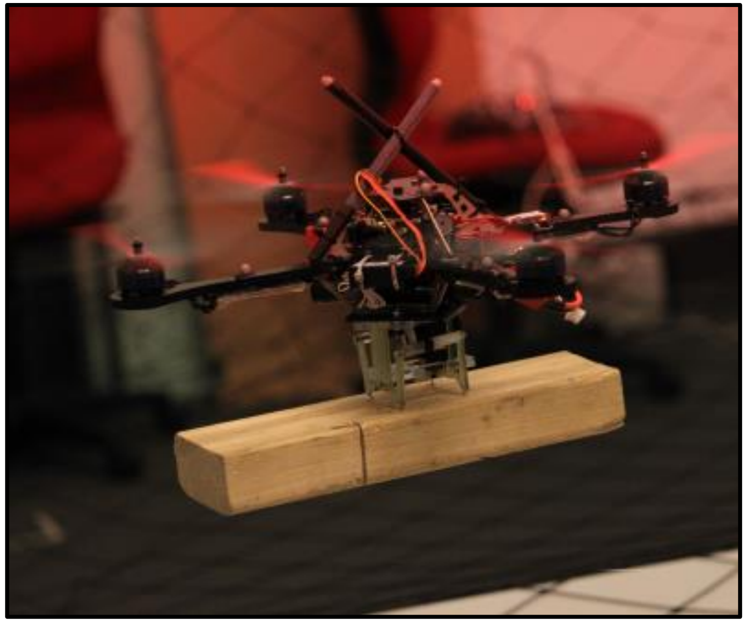
Commanded

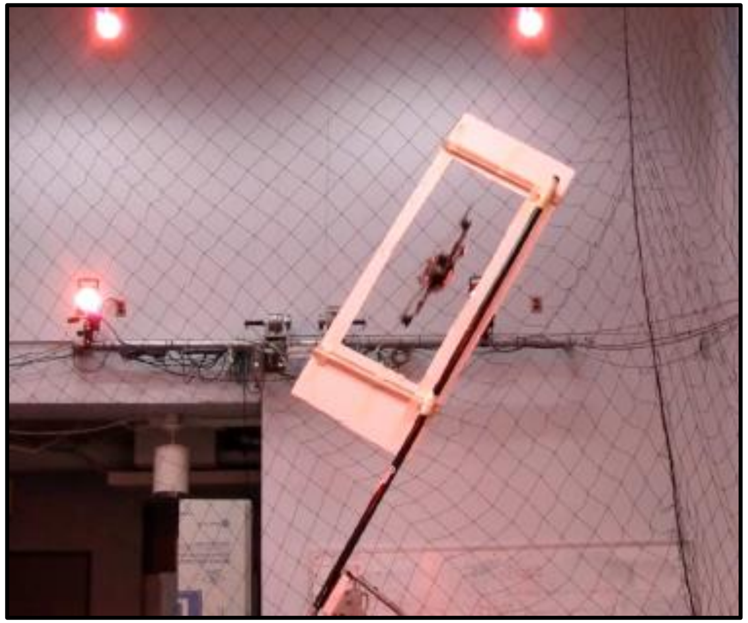
Actual (feedback)

Lupashin, Sergei, et al. "A platform for aerial robotics research and demonstration: The flying machine arena." *Mechatronics* 24.1 (2014): 41-54.

4. LIMITATIONS OF LINEAR CONTROL

Assumptions: roll and pitch angles, and all velocities are close to zero





4. NONLINEAR MODEL (1 of 2)

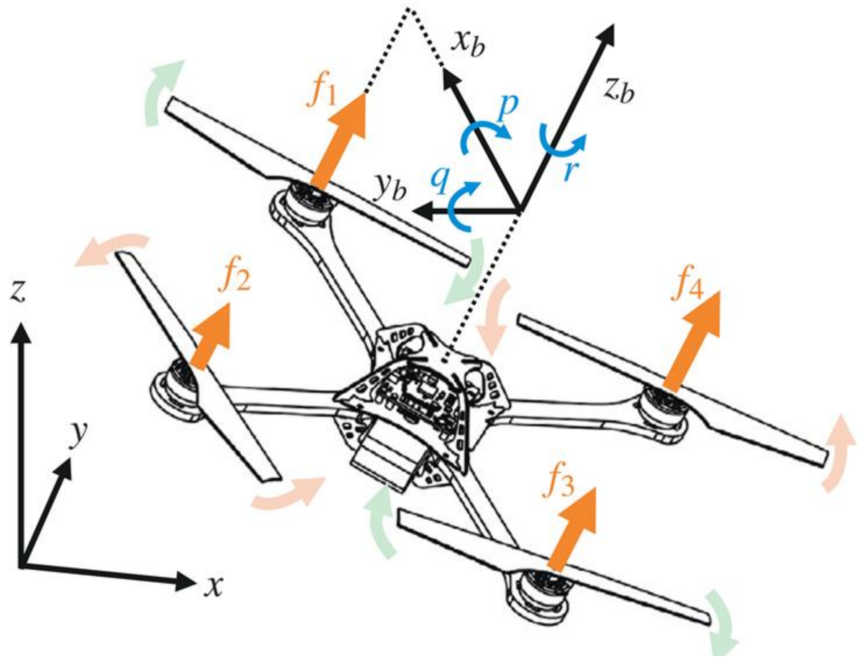
Newton's equations of motion

$$\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix} = \begin{bmatrix}0\\ 0\\ -g\end{bmatrix} + R\begin{bmatrix}0\\ 0\\ c\end{bmatrix}$$

where

$$c = \frac{(F_1 + F_2 + F_3 + F_4)}{m}$$

is mass-normalized.



Maps from body to inertial frame

4. NONLINEAR MODEL (2 of 2)

Newton's equations of motion

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ c \end{bmatrix}$$

where

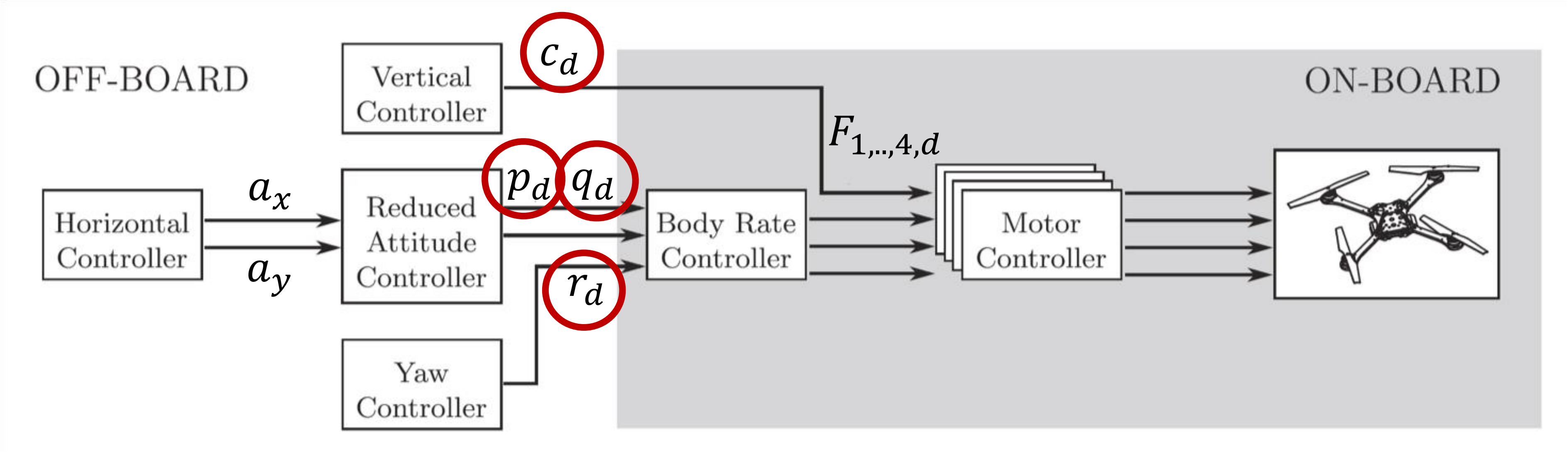
$$c = \frac{(F_1 + F_2 + F_3 + F_4)}{m}$$

is mass-normalized.

Euler's equations of motion

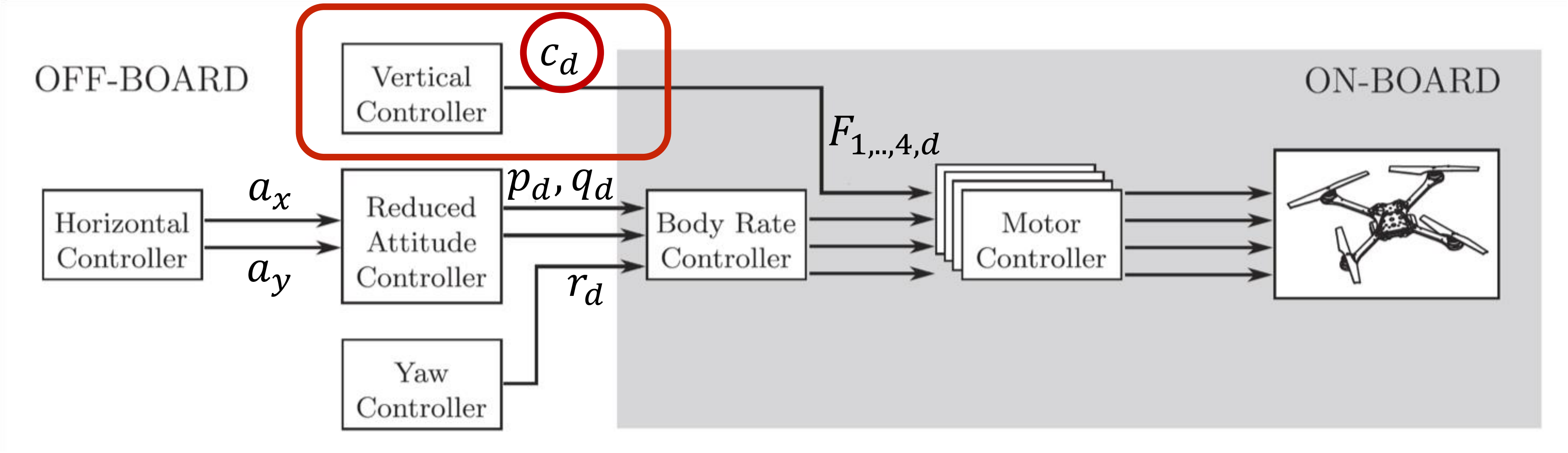
$$\boldsymbol{I} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} L(F_2 - F_4) \\ L(F_3 - F_1) \\ \kappa(F_1 - F_2 + F_3 - F_4) \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \boldsymbol{I} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

4. CONTROL DESIGN OVERVIEW



We need to calculate four inputs for the on-board controllers

4. PART 1: VERTICAL CONTROL



4. PART 1: VERTICAL CONTROL

Recall the equation of motion:

$$\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix} = \begin{bmatrix}0\\ 0\\ -g\end{bmatrix} + \boldsymbol{R}\begin{bmatrix}0\\ 0\\ c\end{bmatrix}$$

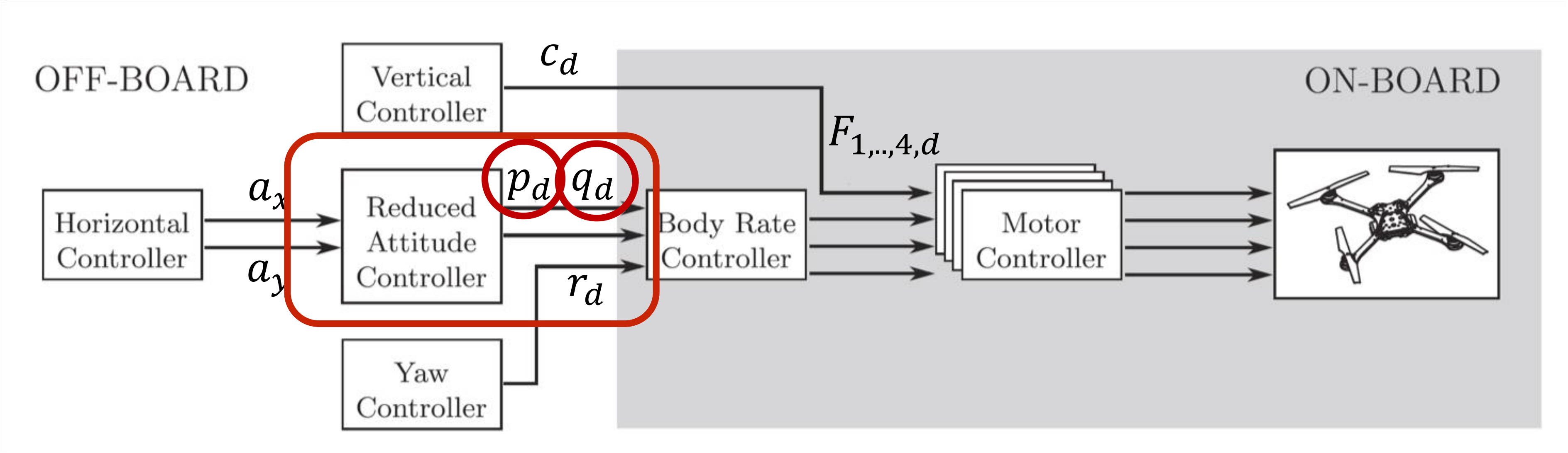
Solve for c :

$$c = \frac{1}{R_{33}}(\ddot{z} + g)$$

Substitute controller for exponentially decaying error:

$$c_d = \frac{1}{R_{33}}\left(\boxed{\omega_{n,z}^2(z_d - z) + 2\zeta_z\omega_{n,z}(\dot{z}_d - \dot{z}) + \ddot{z}_d} + g\right)$$

4. PART 2: LATERAL CONTROL



4. PART 2: LATERAL CONTROL

Horizontal controller to determine accelerations:

$$\ddot{x} = \omega_{n,x}^2(x_d - x) + 2\zeta_x\omega_{n,x}(\dot{x}_d - \dot{x}) + \ddot{x}_d$$
$$\ddot{y} = \omega_{n,y}^2(y_d - y) + 2\zeta_y\omega_{n,y}(\dot{y}_d - \dot{y}) + \ddot{y}_d$$

From the equations of motion:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} R_{13,d} \\ R_{23,d} \end{bmatrix} c_d$$

Shape reaction of matrix entries as a first-order system:

$$\dot{R}_{13,d} = \frac{1}{\tau_{13,d}}\left(R_{13,d} - R_{13}\right) \qquad \dot{R}_{23,d} = \frac{1}{\tau_{23,d}}\left(R_{23,d} - R_{23}\right)$$

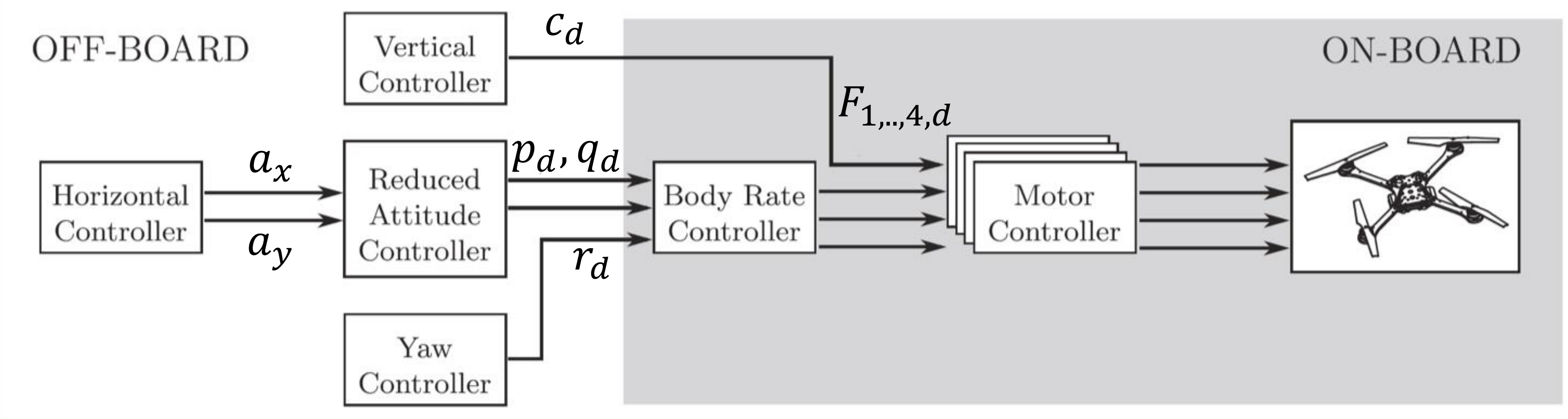
Use rotational kinematics and rate of change calculated above:

$$\begin{bmatrix} p_d \\ q_d \end{bmatrix} = \frac{1}{R_{33}} \begin{bmatrix} R_{21} & -R_{11} \\ R_{22} & -R_{12} \end{bmatrix} \begin{bmatrix} \dot{R}_{13,d} \\ \dot{R}_{23,d} \end{bmatrix}$$

Rotational kinematics:

$$\dot{\boldsymbol{R}} = \boldsymbol{R} \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix}$$

4. OVERALL CONTROL

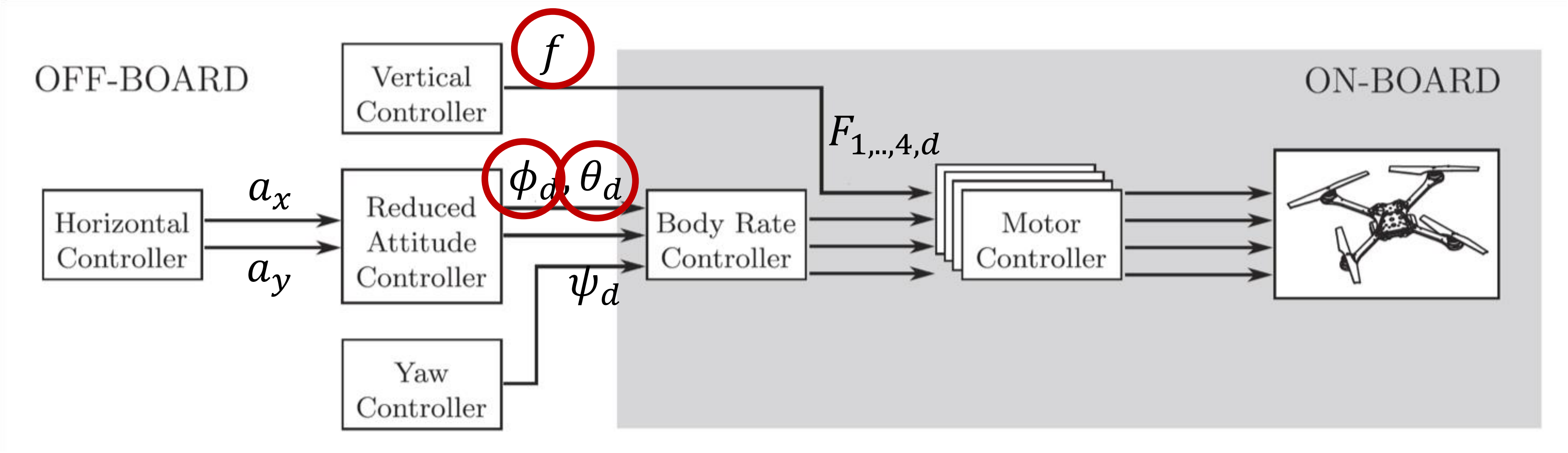


Body rate controller:

$$\boldsymbol{I}\begin{bmatrix}\frac{1}{\tau_p}(p_d-p)\\ \frac{1}{\tau_q}(q_d-q)\\ \frac{1}{\tau_r}(r_d-r)\end{bmatrix}+\begin{bmatrix}p\\q\\r\end{bmatrix}\times\boldsymbol{I}\begin{bmatrix}p\\q\\r\end{bmatrix}=\begin{bmatrix}l(F_{2,d}-F_{4,d})\\ l(F_{3,d}-F_{1,d})\\ \kappa(F_{1,d}-F_{2,d}+F_{3,d}-F_{4,d})\end{bmatrix}$$

$$\left(F_{1,d}-F_{2,d}+F_{3,d}-F_{4,d}\right)=mc_d$$

5. LAB NONLINEAR CONTROLLER



We need to calculate three inputs for the on-board controllers (yaw controller is optional)

Newton's equations of motion

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ c \end{bmatrix}$$

Euler's equations of motion

$$\boldsymbol{I} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} L(F_2 - F_4) \\ L(F_3 - F_1) \\ \kappa(F_1 - F_2 + F_3 - F_4) \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \boldsymbol{I} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

Using $\psi = 0$ to simplify the rotation matrix:

$$R = \begin{bmatrix} . & . & \cos\phi \sin\theta \\ . & . & \sin\phi \\ . & . & \cos\theta \cos\phi \end{bmatrix}$$

5. LAB NONLINEAR CONTROLLER

Newton's equations of motion

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ c \end{bmatrix}$$

$$R = \begin{bmatrix} . & . & \cos\phi \sin\theta \\ . & . & \sin\phi \\ . & . & \cos\theta \cos\phi \end{bmatrix}$$

Hence:

$$\ddot{x} = f \cos\phi \sin\theta$$
$$\ddot{y} = -f \sin\phi$$
$$\ddot{z} = \cos\theta \cos\phi$$

Solve for angles and force:

$$\phi = \sin^{-1}\left(\frac{-\ddot{y}}{f}\right)$$
$$\theta = \sin^{-1}\left(\frac{-\ddot{x}}{f \cos\phi}\right)$$
$$f = \frac{\ddot{z} + g}{\cos\theta \cos\phi}$$

5. LAB NONLINEAR CONTROLLER

Use measurements to calculate the current mass-normalized force:

$$f = \frac{\ddot{z} + g}{\cos\theta \cos\phi}$$

Horizontal controller to determine accelerations:

$$\ddot{x} = \omega_{n,x}^2 (x_d - x) + 2\zeta_x \omega_{n,x} (\dot{x}_d - \dot{x})$$
$$\ddot{y} = \omega_{n,y}^2 (y_d - y) + 2\zeta_y \omega_{n,y} (\dot{y}_d - \dot{y})$$

Calculate commanded roll and pitch:

$$\phi_c = \sin^{-1}\left(\frac{-\ddot{y}}{f}\right)$$
$$\theta_c = \sin^{-1}\left(\frac{-\ddot{x}}{f \cos\phi_c}\right)$$

4. OTHER APPROACHES

- **Linear controller based on linearized system**
  - Hoffmann, Gabriel, et al. "Quadrotor helicopter flight dynamics and control: Theory and experiment." AIAA guidance, navigation and control conference and exhibit. 2007.
  - Gurdan, Daniel, et al. "Energy-efficient autonomous four-rotor flying robot controlled at 1 kHz." Proceedings 2007 IEEE International Conference on Robotics and Automation. IEEE, 2007.
  - Bouabdallah, Samir. Design and control of quadrotors with application to autonomous flying. No. THESIS. Epfl, 2007.
- **LQR**
  - Cowling, Ian D., James F. Whidborne, and Alastair K. Cooke. "Optimal trajectory planning and LQR control for a quadrotor UAV." International Conference on Control. 2006.
- **Feedback Linearization / Backstepping**
  - Madani, Tarek, and Abdelaziz Benallegue. "Backstepping control for a quadrotor helicopter." *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006.
  - Lee, Daewon, H. Jin Kim, and Shankar Sastry. "Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter." *International Journal of control, Automation and systems* 7 (2009): 419-428.
- **Exact linearization, differentially flat system**
  - Mellinger, Daniel, and Vijay Kumar. "Minimum snap trajectory generation and control for quadrotors." 2011 IEEE international conference on robotics and automation. IEEE, 2011.
- **L1 adaptive control**
  - Hovakimyan, Naira, and Chengyu Cao. $\mathcal{L}1$ Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation. Society for Industrial and Applied Mathematics, 2010.

OUTLINE

1. Basic Mechanics
2. Dynamics & Control of the Vertical Direction (1D)
3. Dynamics & Control in the Vertical Plane (2D)
4. Trajectory Tracking Control (3D)
5. **Summary**
6. What Can Go Wrong?
7. Learning-Enabled Control

6. SUMMARY

1. Quadrotor dynamics are nonlinear and coupled

2. Four independent degrees of freedom

3. Controller approach:
   A. Nested controller
   B. “Transform” individual loops into double integrators
   C. Design the closed-loop behavior to resemble a second-order system behavior
   D. Two intuitive parameters to choose for each second-order system

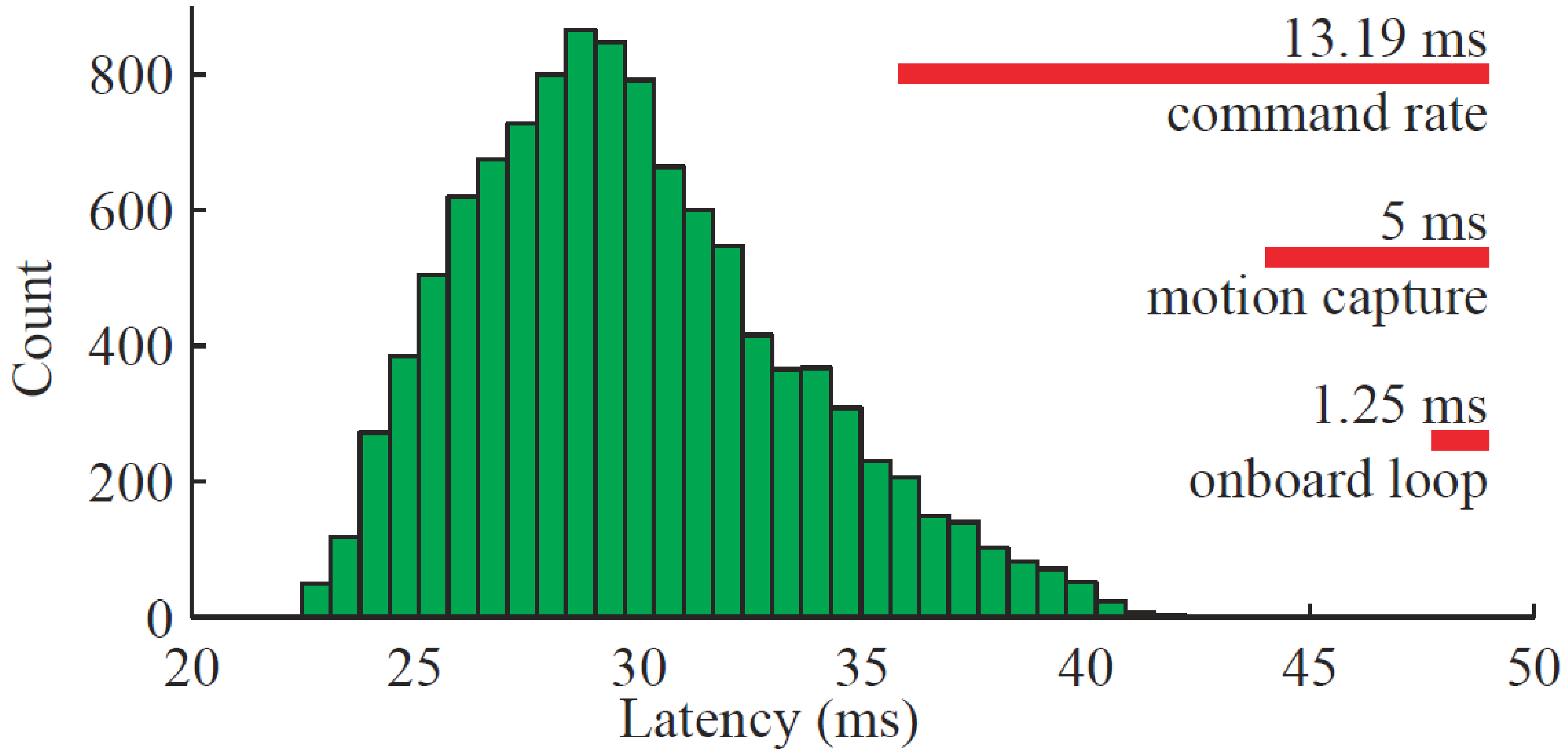
4. There are many more advanced controllers including adaptive and learning controls

OUTLINE

1. Basic Mechanics
2. Dynamics & Control of the Vertical Direction (1D)
3. Dynamics & Control in the Vertical Plane (2D)
4. Trajectory Tracking Control (3D)
5. Summary
6. **What Can Go Wrong?**
7. Learning-Enabled Control

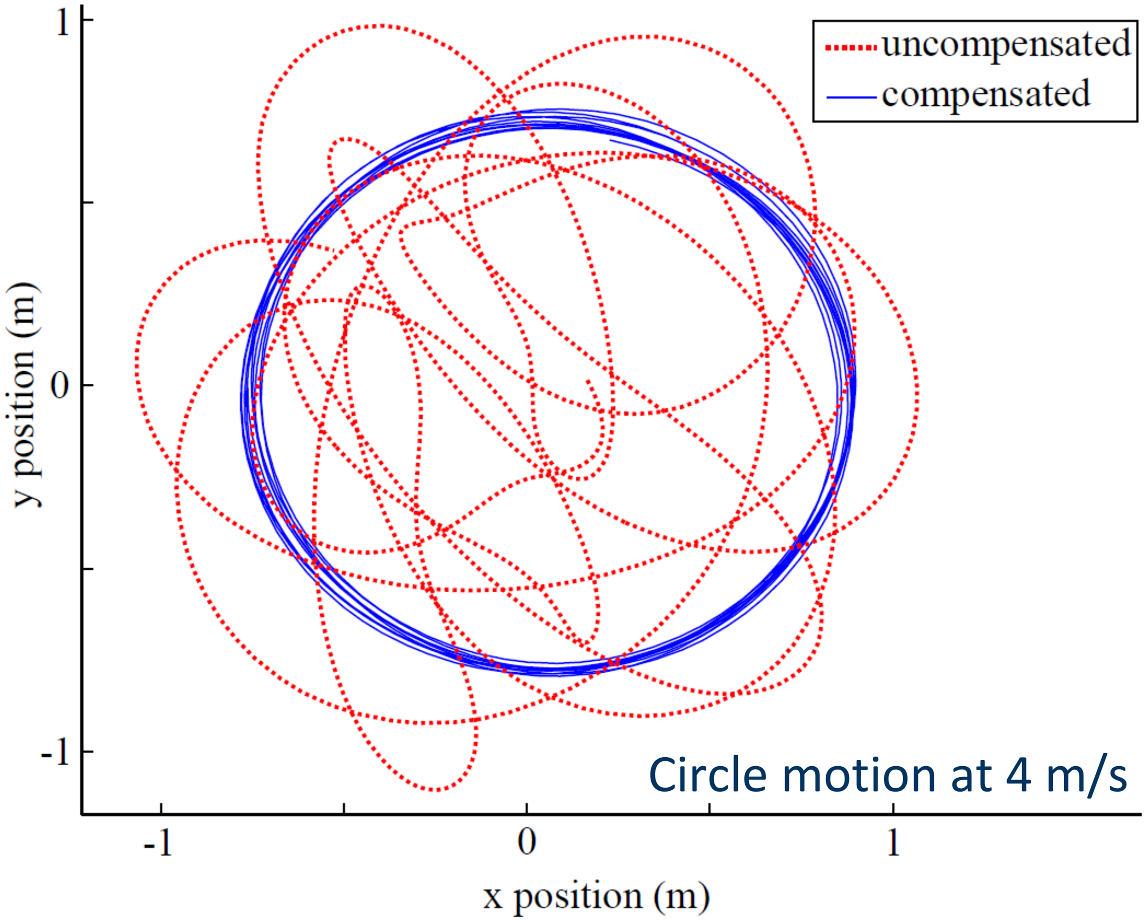
4. LIMITATIONS (from doi.org/10.1016/j.mechatronics.2013.11.006)

- Latency (1 of 3)



4. LIMITATIONS

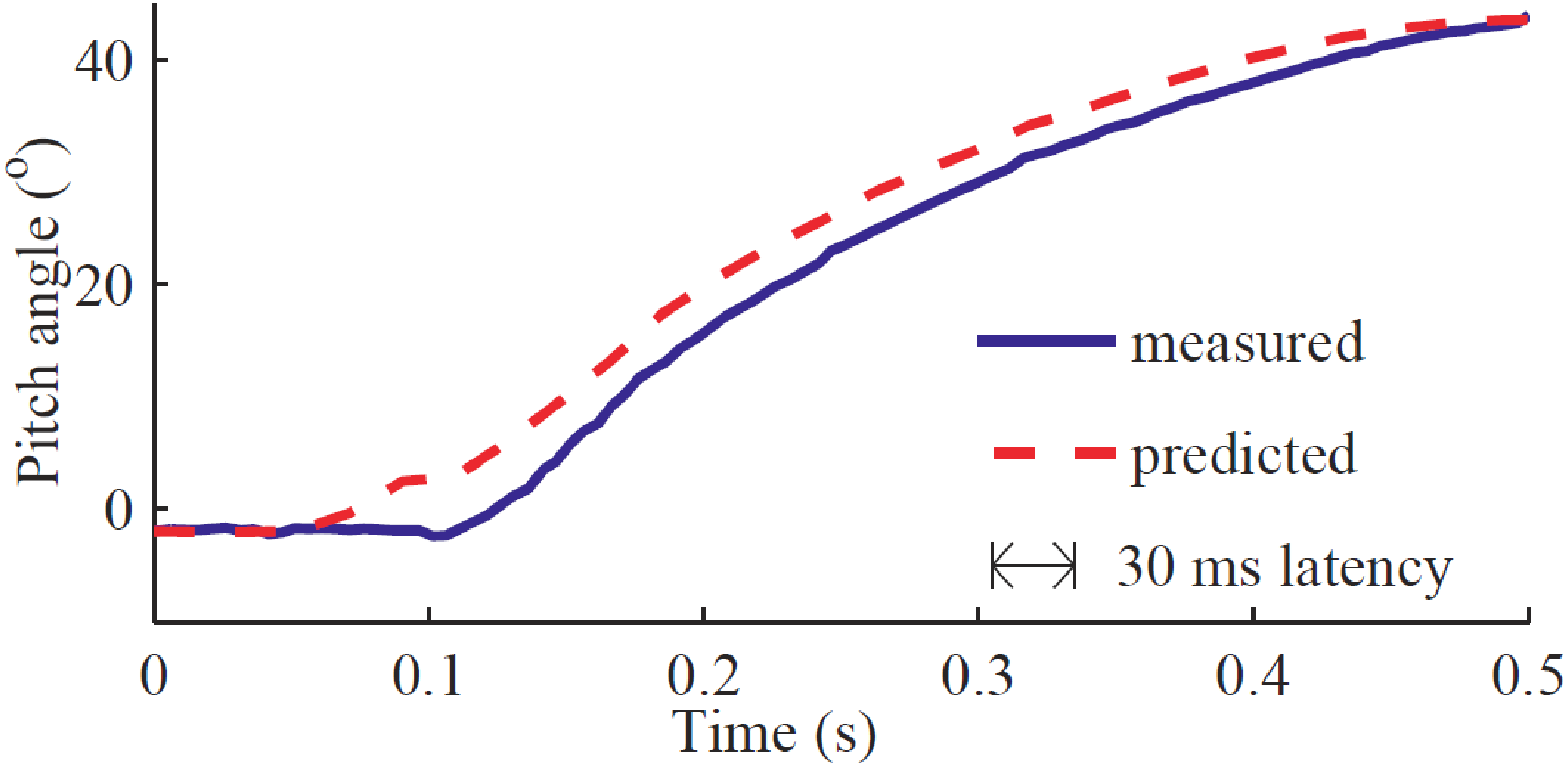
- Latency (2 of 3)



4. LIMITATIONS

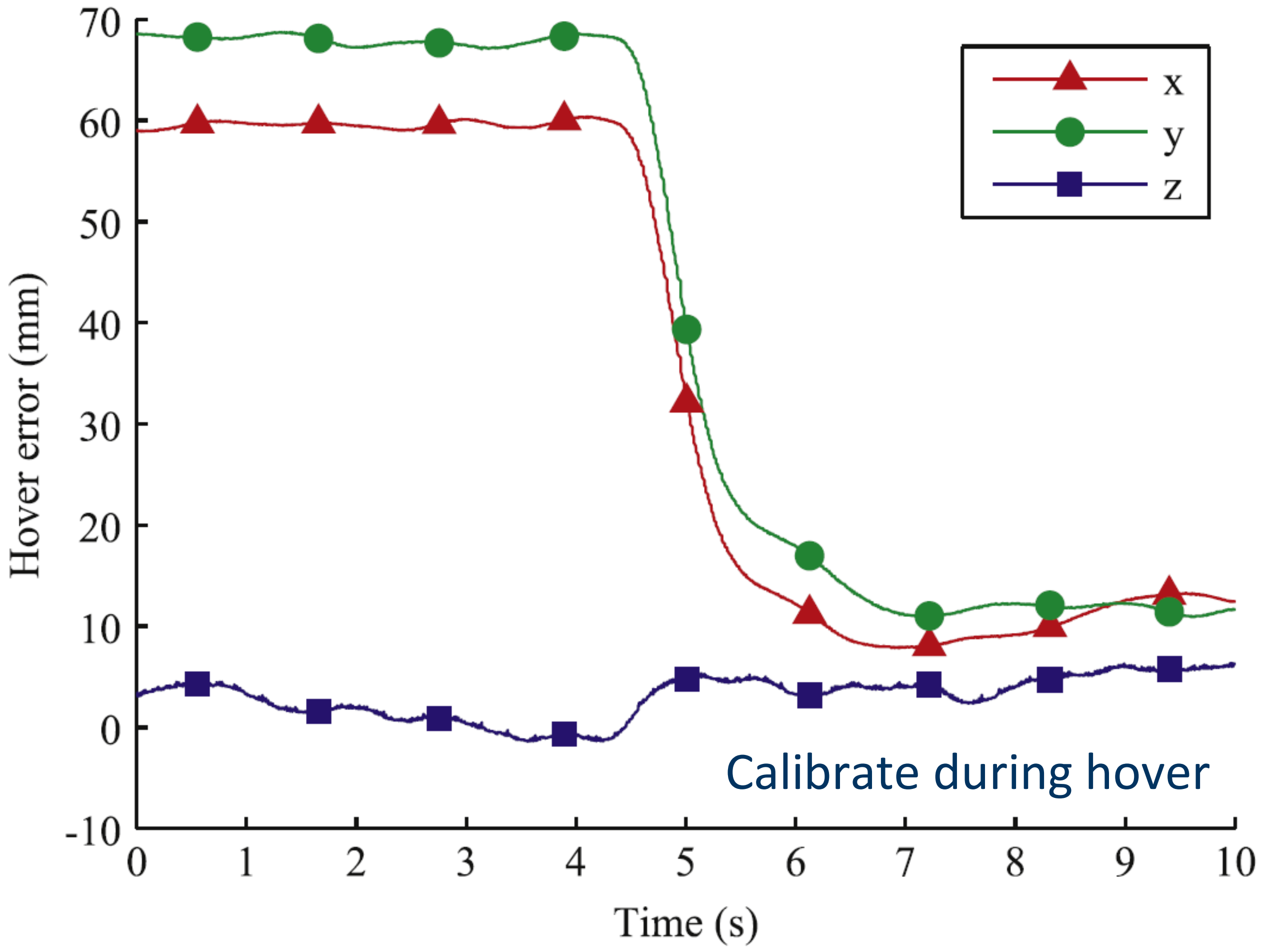
- Latency (3 of 3)

Predict the vehicle position at the time the input arrives at the vehicle



4. LIMITATIONS

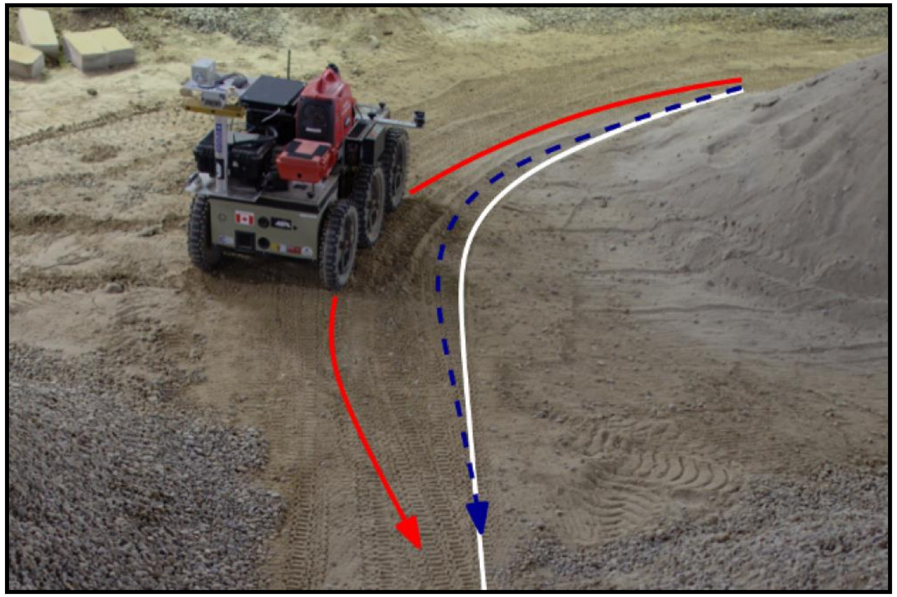
- Offsets



4. LIMITATIONS

- Aggressive Maneuvers

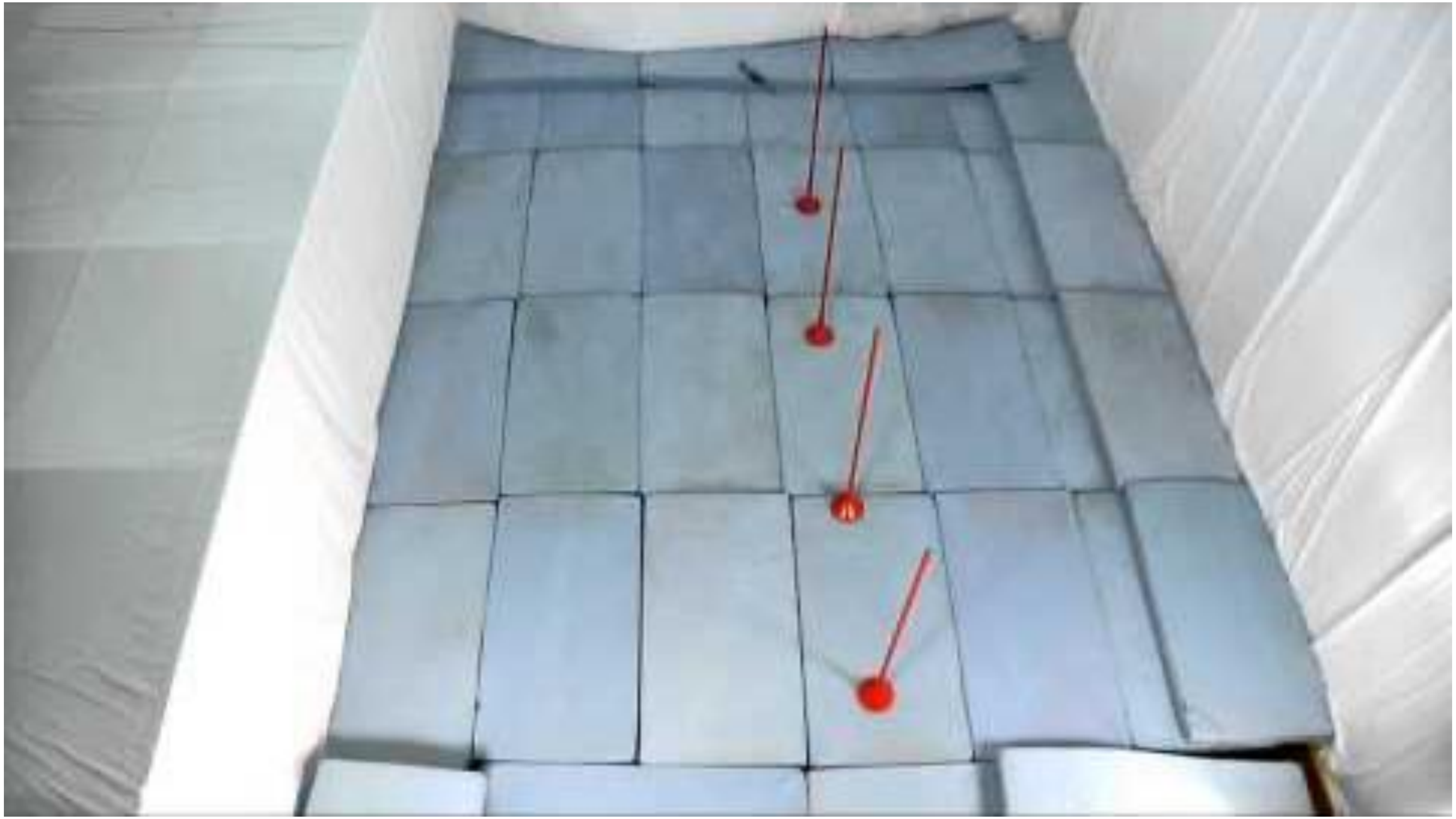
1. Triple flip with a quadrotor
2. Time-optimized slalom
3. Fast path following with a ground vehicle
4. ..



4. TRIPLE ADAPTIVE FLIPS

youtu.be/bWExDW9J9sA

4. SLALOM LEARNING



youtu.be/zHTCsSkmADo

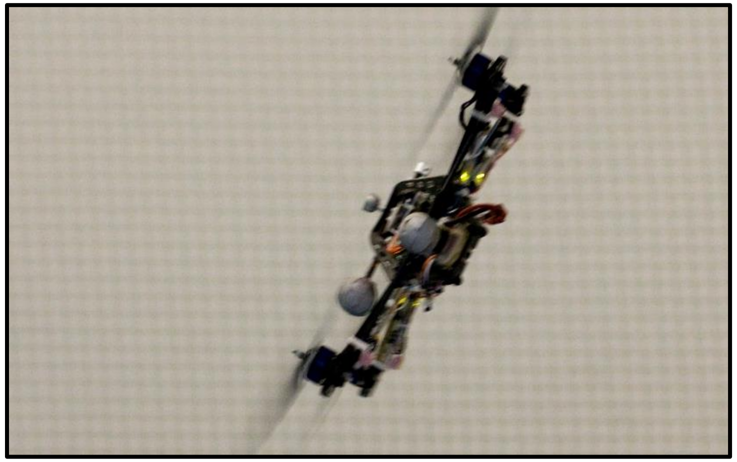
4. VISUAL TEACH AND REPEAT

youtu.be/08_d1HSPADA

4. EXPLANATION

1. Unmodelled dynamics

2. Unknown external disturbances
   - (e.g., environment conditions such as wind, unknown payload, topography or weather)



<u>Model inaccuracies limit achievable performance!</u>

OUTLINE

1. Basic Mechanics
2. Dynamics & Control of the Vertical Direction (1D)
3. Dynamics & Control in the Vertical Plane (2D)
4. Trajectory Tracking Control (3D)
   - What Can Go Wrong?
5. **Learning-Enabled Control**
6. Summary

5. LEARNING-ENABLE CONTROL: RATIONALE

Learning and adaptation enable

- safe, high-performance motions
- in uncontrolled, unknown or changing environments

5. RESEARCH FOCUS

DYNAMICS → CONTROLS → MACHINE LEARNING → ROBOTICS

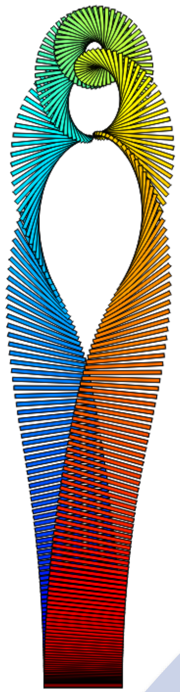
Prior information	+ Current sensor measurement	+ Past experiment data
• Which motions are feasible? • How to plan collision-free motions?	• How to guide the vehicle along a desired path?	• Can the performance be improved by leveraging past data?

Towards robotics applications

4. LEARNING-BASED ROBUST CONTROL

youtu.be/YqhLnCm0KXY

5. DEVELOPMENT PROCESS

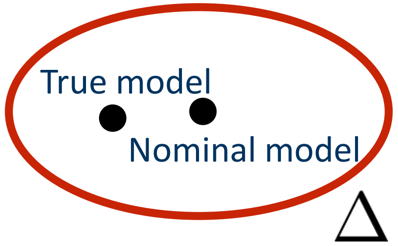


Specific task, adaptation of a few input parameters only

General task, full input trajectory adaptation

Model learning, anytime learning

Learning with safety guarantees



5. CONCLUDING REMARKS

- Safe, reliable, high-performance flight
- of single or multiple aerial vehicles
- in changing and unknown conditions

  using a combination of a-priori models and data

DYNAMICS

CONTROLS

MACHINE LEARNING

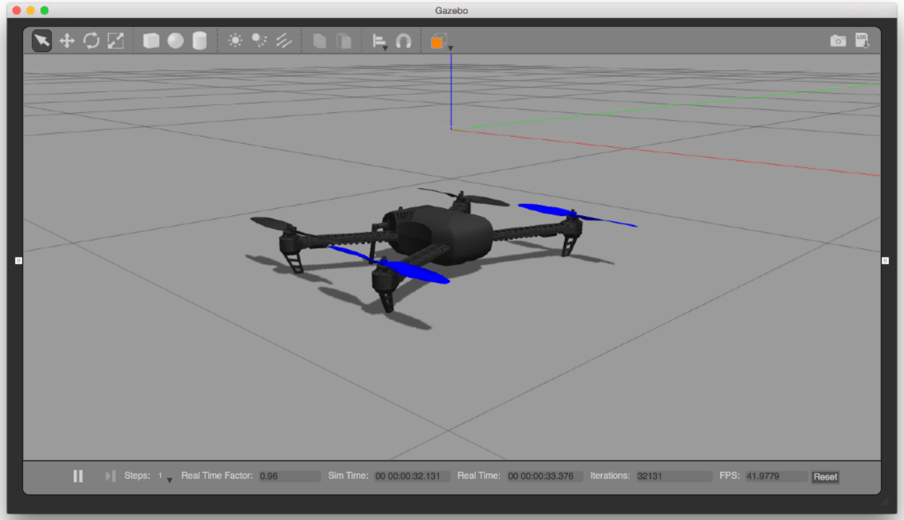
ROBOTICS

UPCOMING LECTURES

The next two lectures will be labs :

1. Quadrotor simulation and position control design

2. From quadrotor simulation to experiment.



THAK YOU, Angela!

For follow-up discussions, please contact:

Prof. Angela P. Schöllig

- email: schoellig@utias.utoronto.ca

Dynamic System Lab

- web: **dynsyslab.org**
- twitter.com/**dynsyslab**
- facebook.com/**DynamicSystemsLab**
- linkedin.com/company/**dynsyslab**
- YT Channel: **Dynamic Systems Lab**