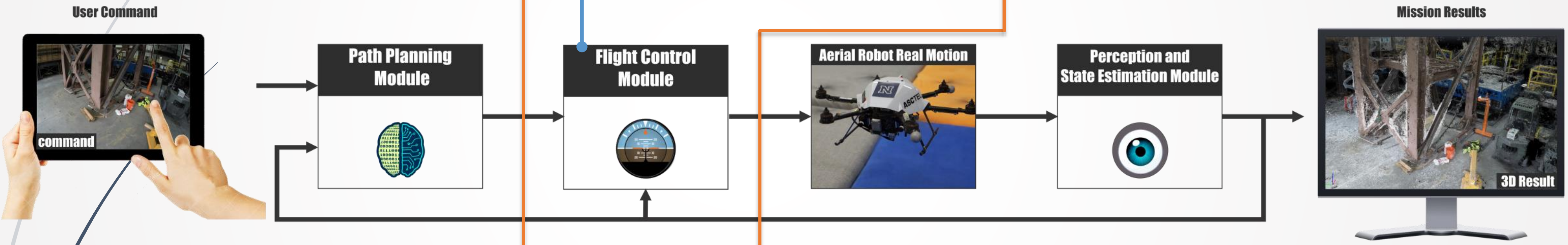
CS491/691: Introduction to Aerial Robotics

Topic: Flight Controls Introduction

Dr. Kostas Alexis (CSE)

The Aerial Robot Loop

- How to design the flight control of an aerial robot and program its autopilot.



Section 3 of our course

MAV Dynamics

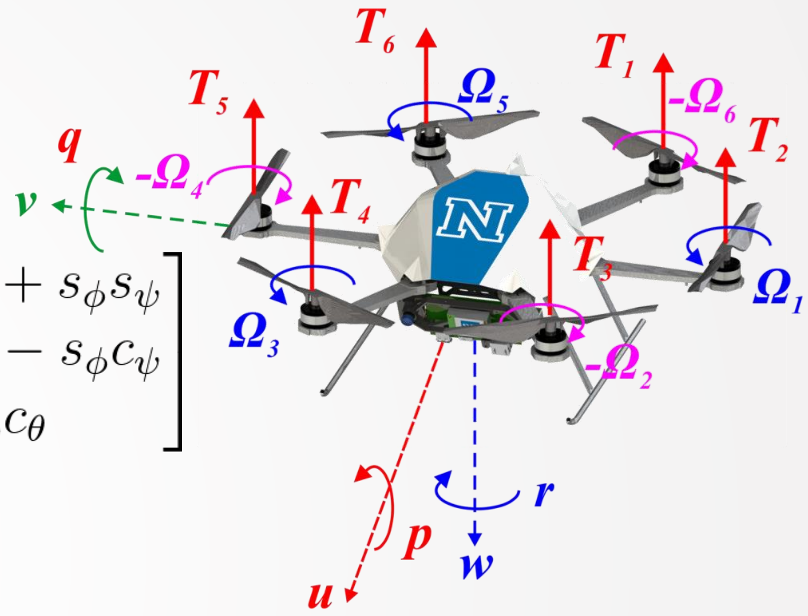
- To append the forces and moments we need to combine their formulation with

$$\begin{bmatrix}\dot{p}_n\\ \dot{p}_e\\ \dot{p}_d\end{bmatrix} = \mathcal{R}_b^v \begin{bmatrix}u\\ v\\ w\end{bmatrix}, \quad \mathcal{R}_b^v = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix}$$

$$\begin{bmatrix}\dot{u}\\ \dot{v}\\ \dot{w}\end{bmatrix} = \begin{bmatrix}rv - qw\\ pw - ru\\ qu - pv\end{bmatrix} + \frac{1}{m}\begin{bmatrix}f_x\\ f_y\\ f_z\end{bmatrix}$$

$$\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix}1 & \sin\phi\tan\theta & \cos\phi\tan\theta\\ 0 & \cos\phi & -\sin\phi\\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta\end{bmatrix}\begin{bmatrix}p\\ q\\ r\end{bmatrix}$$

$$\begin{bmatrix}\dot{p}\\ \dot{q}\\ \dot{r}\end{bmatrix} = \begin{bmatrix}\frac{J_y - J_z}{J_x} qr\\ \frac{J_z - J_x}{J_y} pr\\ \frac{J_x - J_y}{J_z}\end{bmatrix}\begin{bmatrix}p\\ q\\ r\end{bmatrix} + \begin{bmatrix}\frac{1}{J_x} M_x\\ \frac{1}{J_y} M_y\\ \frac{1}{J_z} M_z\end{bmatrix}$$

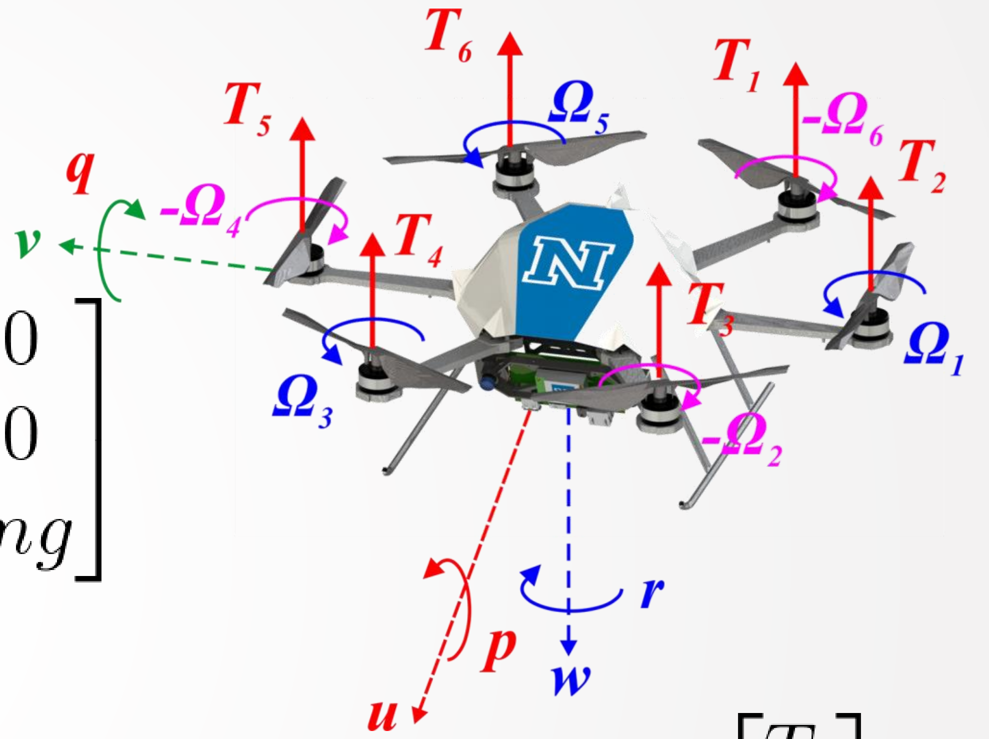


- ***Next step: append the MAV forces and moments***

MAV Dynamics

- MAV forces in the body frame:

$$\mathbf{f}_b = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^{6} T_i \end{bmatrix} - \mathcal{R}_v^b \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}$$



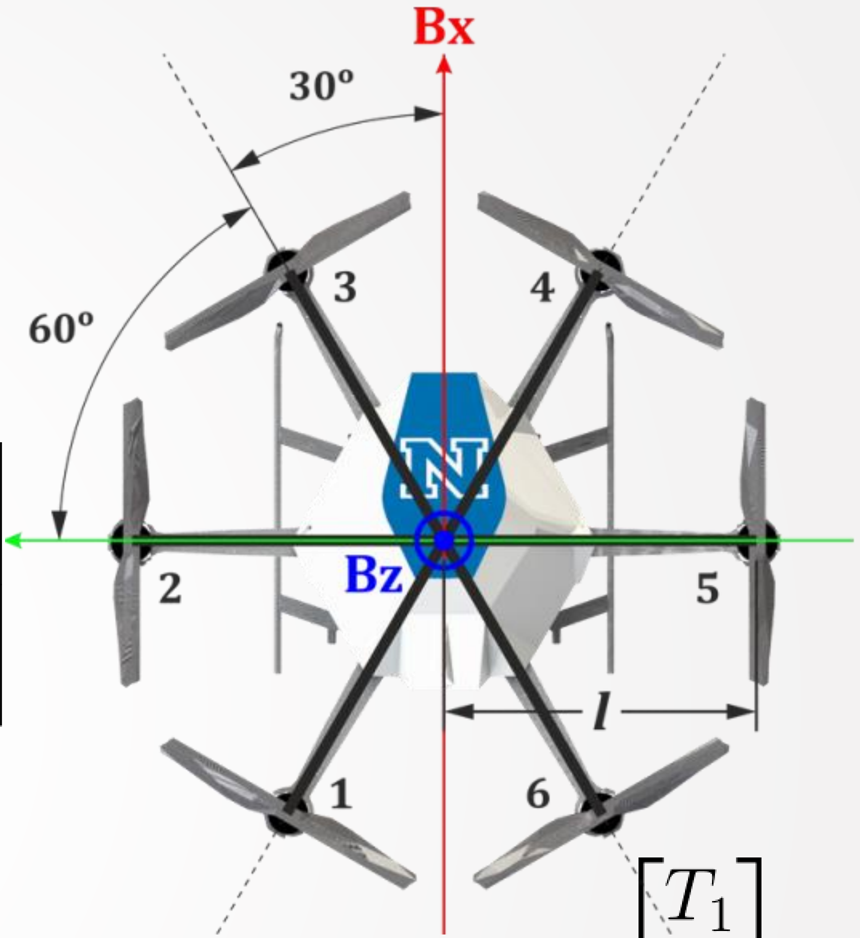
- Moments in the body frame:

$$\mathbf{m}_b = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} ls_{30} & l & ls_{30} & -ls_{30} & -l & ls_{30} \\ -lc_{60} & 0 & lc_{60} & lc_{60} & 0 & -lc_{60} \\ -k_m & k_m & -k_m & k_m & -k_m & k_m \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \end{bmatrix}$$

MAV Dynamics

- MAV forces in the body frame:

$$\mathbf{f}_b = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^{6} T_i \end{bmatrix} - \mathcal{R}_v^b \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}$$



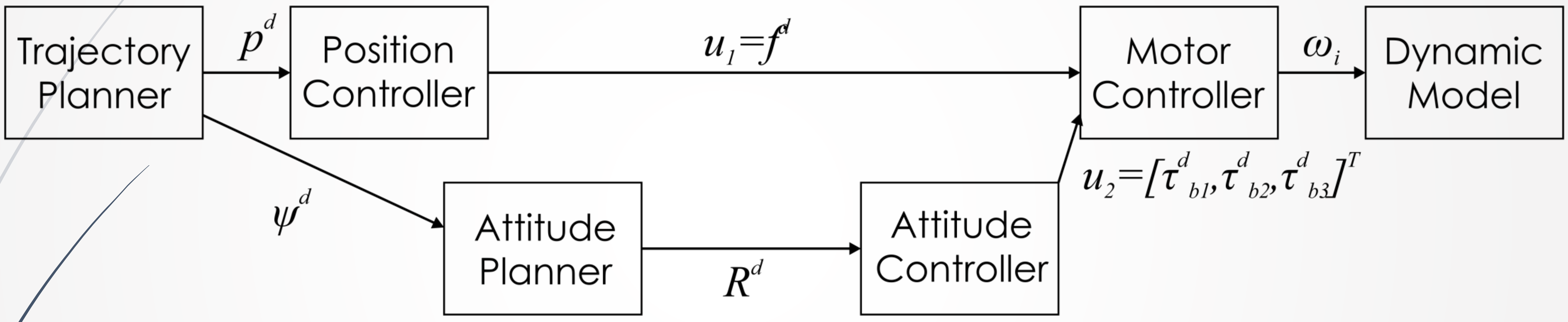
- Moments in the body frame:

$$\mathbf{m}_b = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} ls_{30} & l & ls_{30} & -ls_{30} & -l & ls_{30} \\ -lc_{60} & 0 & lc_{60} & lc_{60} & 0 & -lc_{60} \\ -k_m & k_m & -k_m & k_m & -k_m & k_m \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \end{bmatrix}$$

MAV Dynamics

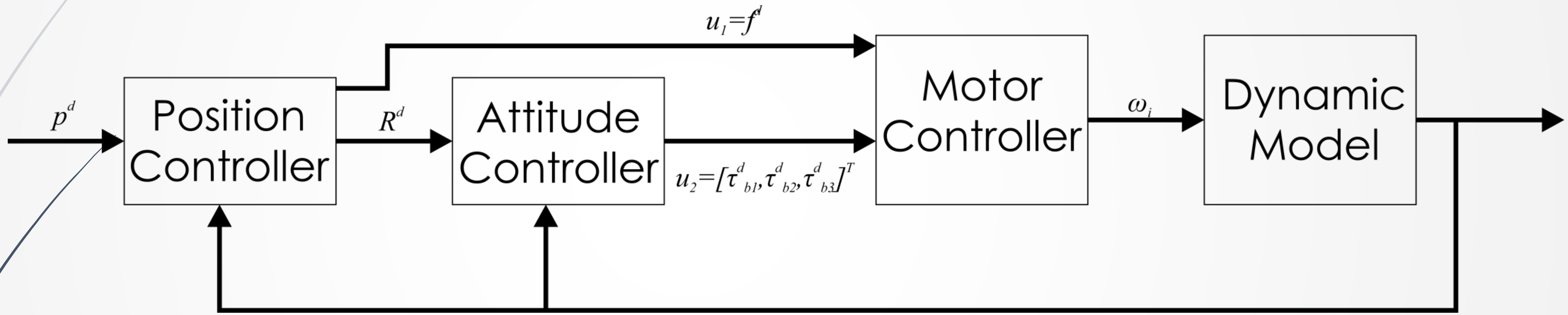
- Exchangeable symbols:
  - M with τ for Moments
  - F with f for forces
  - $\mathrm{x,y,z}$ when used for body-axis with $\mathrm{b_1,b_2,b_3}$

Control System Block Diagram



- There are simpler

Control System Block Diagram



- Simplified loop

Attitude Control

$$\mathbf{u}_2 = -k_R \mathbf{e}_R - k_\omega \mathbf{e}_\omega$$

$$\mathbf{e}_\omega = \omega - \omega^d$$

$$\mathbf{e}_R = \frac{1}{2}[(\mathbf{R}^d)^T \mathbf{R} - \mathbf{R}^T \mathbf{R}^d]^\vee$$

- Where the vee map $^\vee : \mathcal{SO}(3) \to \mathbb{R}^3$ is the inverse of the hat map.

The Hat Map and its Inverse

- The *hat* map

$$\wedge : \mathbb{R}^3 \rightarrow \mathcal{SO}(3)$$

Transforms a vector in $\mathbb{R}^3$ to a 3x3 skew-symmetric matrix that

$$\hat{\mathbf{x}}\mathbf{y} = \mathbf{x} \times \mathbf{y}$$

for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^3$. The inverse of the hat map is denoted by the vee map:

$$\vee : \mathcal{SO}(3) \rightarrow \mathbb{R}^3$$

Attitude Control

- Linearize the nonlinear model around hover:

$$R_0 = R\left(\phi_0 = 0,\ \theta_0 = 0,\ \psi_0\right)$$

$$R^d = R_z\left(\psi_0 + \Delta\psi\right) R_{yx}\left(\Delta\phi, \Delta\theta\right)$$

- Rotation error metric:

$$\mathbf{e}_R = \frac{1}{2}\left(\left(R^d\right)^{\mathrm{T}} R_0 - R_0^{\mathrm{T}} R^d\right)^{\vee}$$

After Linearization:

$$\approx \begin{bmatrix} 0 & \Delta\psi & -\Delta\theta \\ -\Delta\psi & 0 & -\Delta\phi \\ \Delta\theta & -\Delta\phi & 0 \end{bmatrix}^{\vee}$$

$$= \left[\Delta\phi,\ \Delta\theta,\ \Delta\psi\right]^{\mathrm{T}}$$

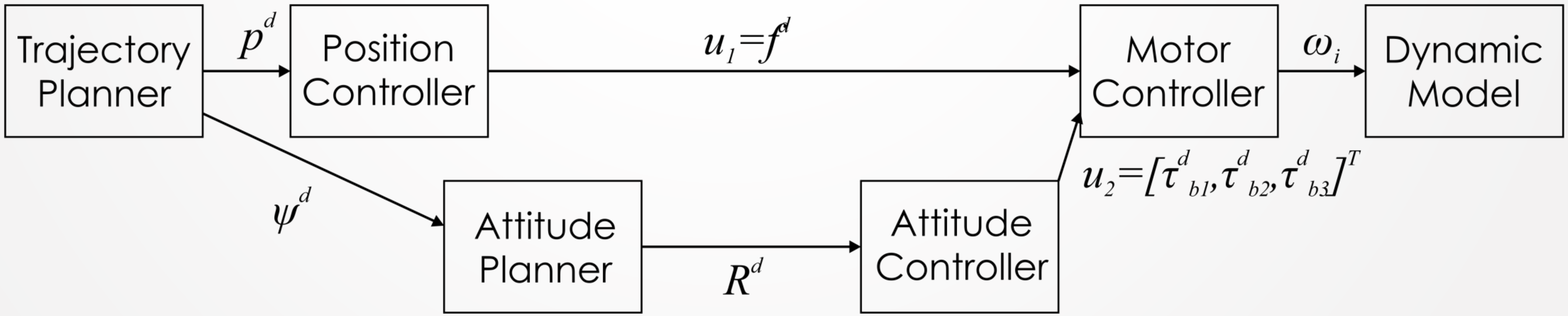
Attitude Control

- PD Control Law:

$$\mathbf{u_2} = -k_R \mathbf{e}_R - k_\omega \mathbf{e}_\omega$$

$$\mathbf{e}_R = [\Delta\phi,\ \Delta\theta,\ \Delta\psi]^{\mathrm{T}}$$

$$\mathbf{e}_\omega = \omega - \omega^d$$



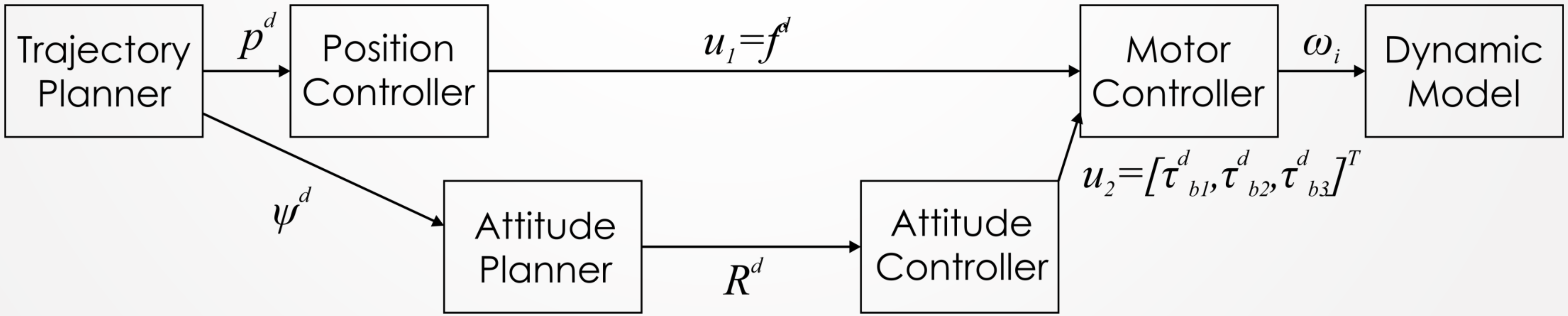
Position Control

- PD Control Law:

$$\mathbf{e}_a + k_d\mathbf{e}_v + k_p\mathbf{e}_p = 0$$

- Linearize the nonlinear model around hover:
  - Nominal Input:

$$\mathbf{u}_1 = mg$$
$$\mathbf{u}_2 = \mathbb{0}_{3\times 1}$$



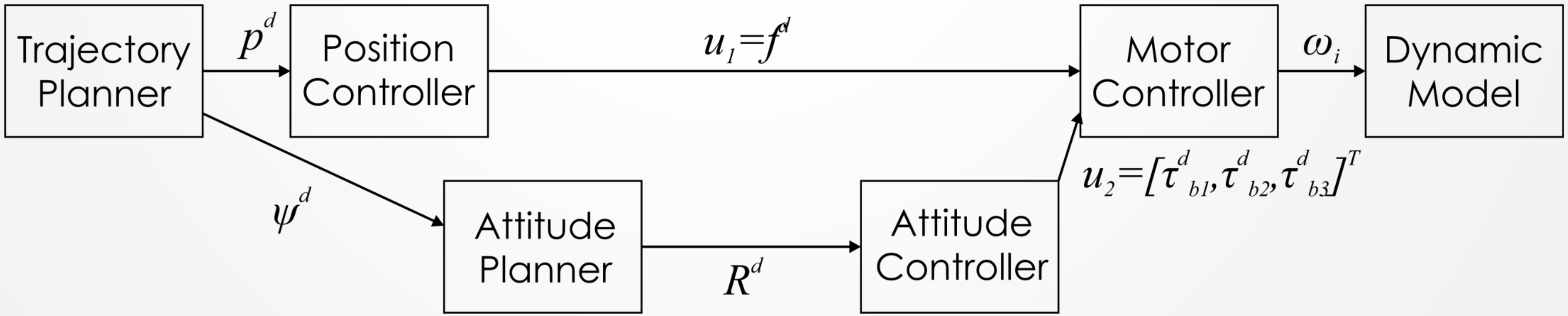
Position Control

- PD Control Law:

$$u_1 = m\mathbf{b}_3^{\mathrm{T}}\left(\mathbf{g} + \mathbf{a}^d + K_d\mathbf{e}_v + K_p\mathbf{e}_p\right)$$

$$\mathbf{e}_p = \mathbf{p} - \mathbf{p}^d$$

$$\mathbf{e}_v = \mathbf{v} - \mathbf{v}^d$$



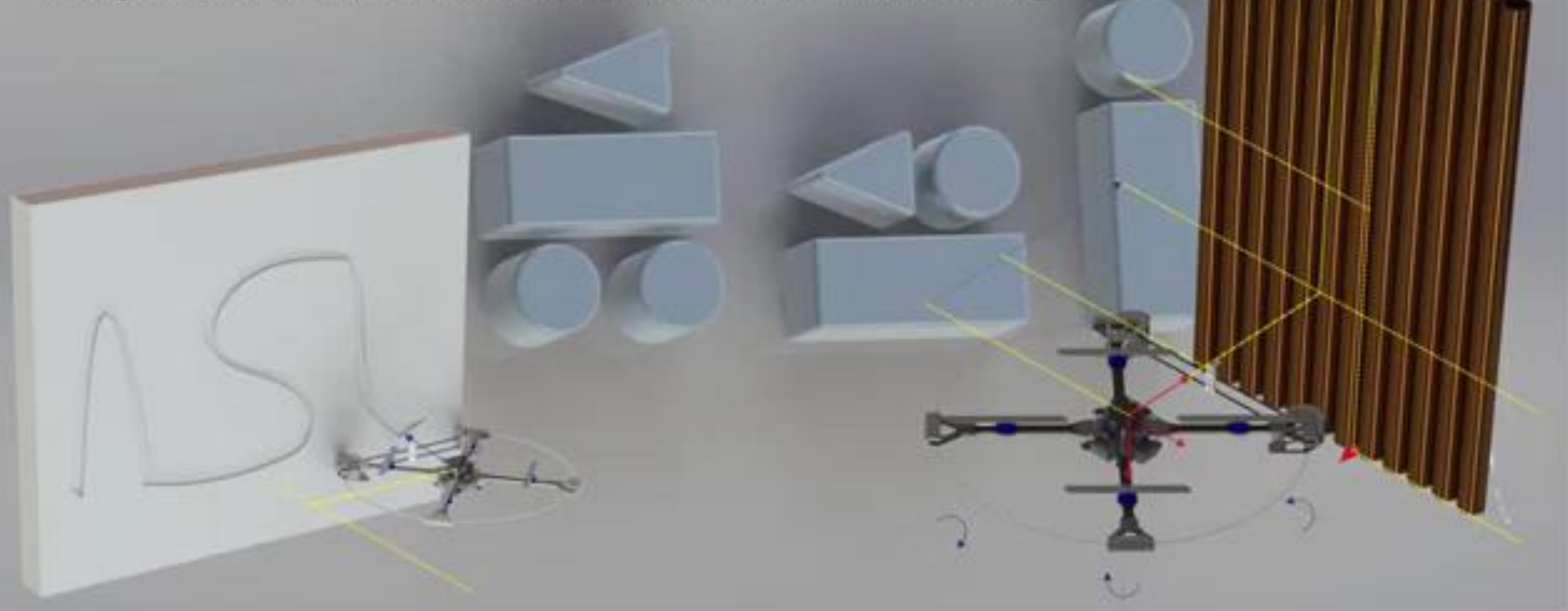
Fast Nonlinear Model Predictive Control for Multicopter Attitude Tracking on SO(3)

Mina Kamel, Kostas Alexis, Markus Achtelik and Roland Siegwart

Position tracking without one propeller

Hybrid Predictive Control for Aerial Robotic Physical Interaction towards Inspection Operations

Georgios Darivianakis, Kostas Alexis, Michael Burri, Roland Siegwart



Very Simple Example

- Control the decoupled roll dynamics of a multirotor aerial robot

$$\begin{bmatrix} \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 1/J_x \end{bmatrix} M_x$$

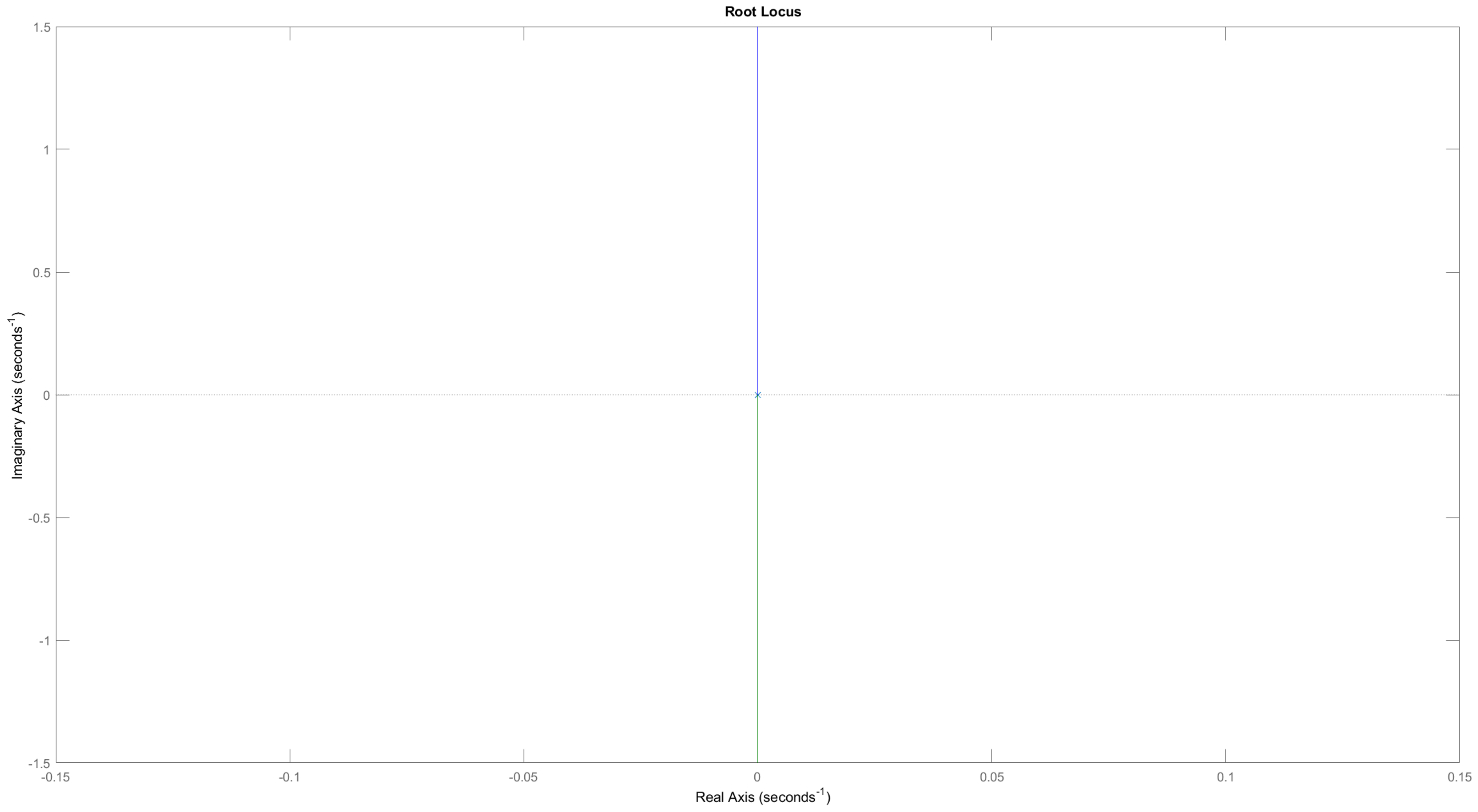
Very Simple Example

- Similar transfer function example:

```
%%  Extremely Oversimplified Example of a Bad Design

dummy_tf = tf(1,[1 0 0]);

dummy_ctrl = 1;

rlocus(dummy_ctrl*dummy_tf);

%%  Extremely Oversimplified Example of a Design that Pretends to Work

nondummy_ctrl = 1 + 0.1*tf([1 0],[0.0001 1]);

rlocus(nondummy_ctrl*dummy_tf);
```

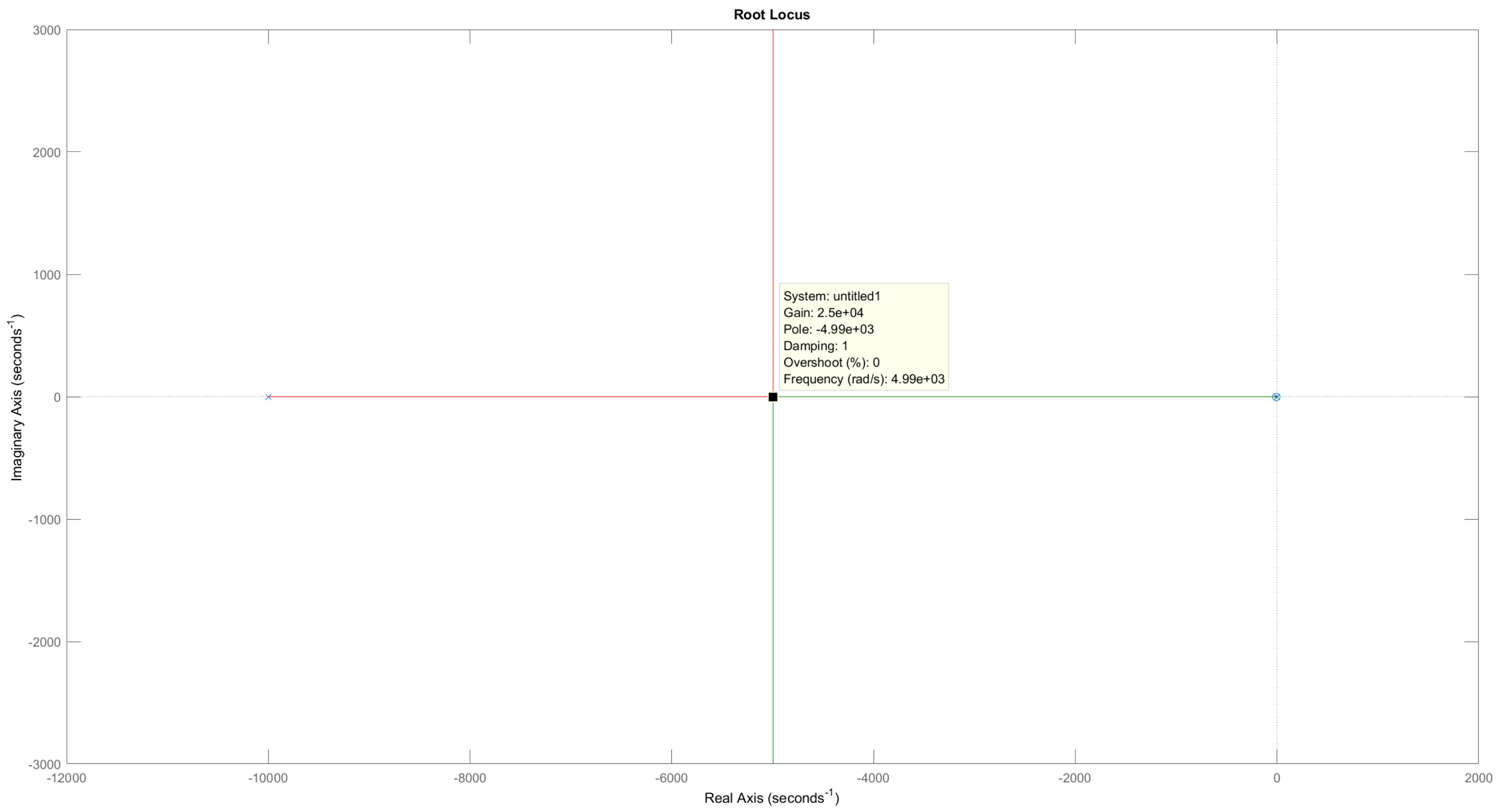
Very Simple Example

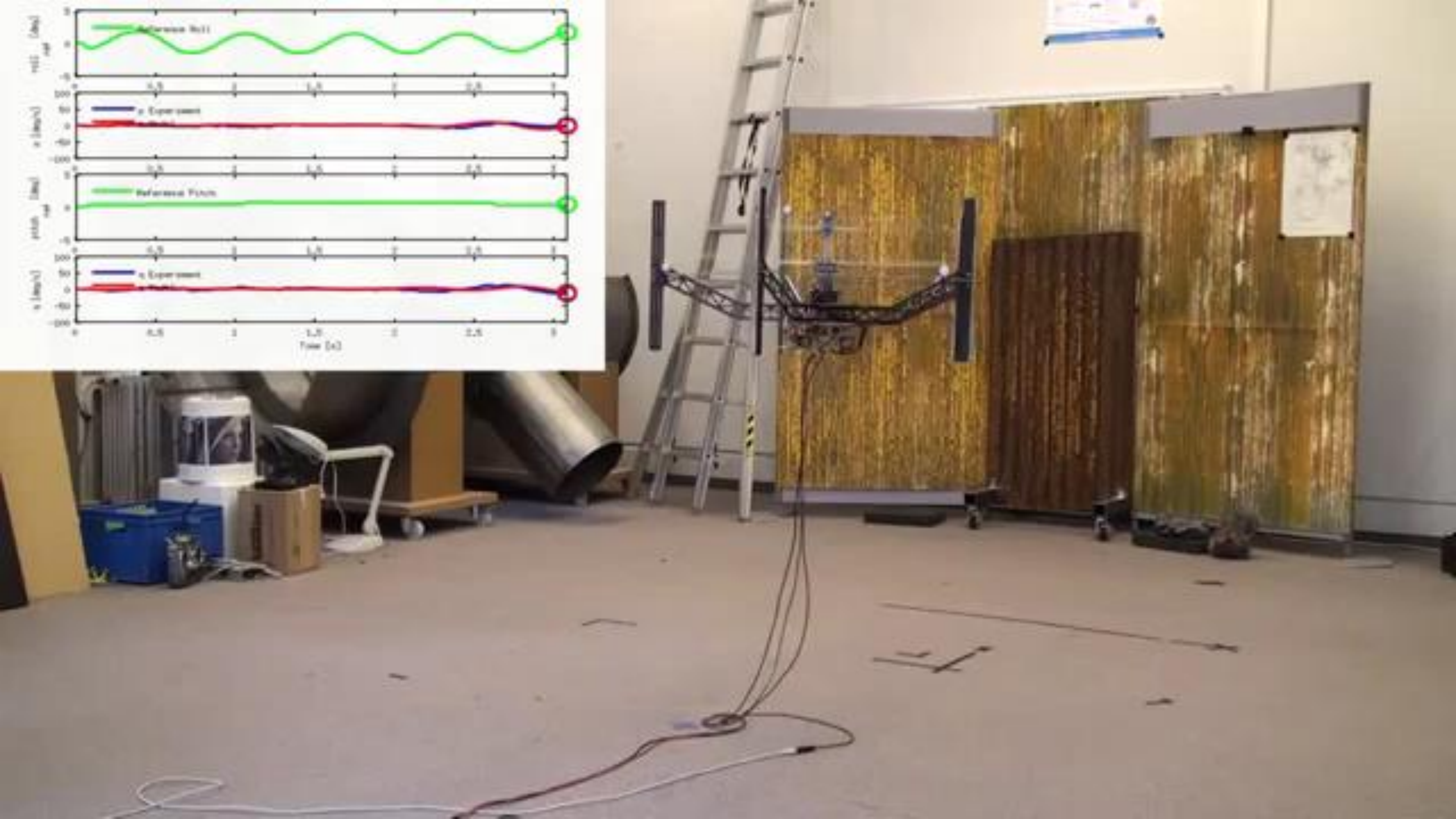
- How to fail terribly:



Very Simple Example

- How to think you did it OK before actually testing in practice:




Reference Roll
p Experiment
Reference Pitch
q Experiment
Time [s]

AtlantikSolar

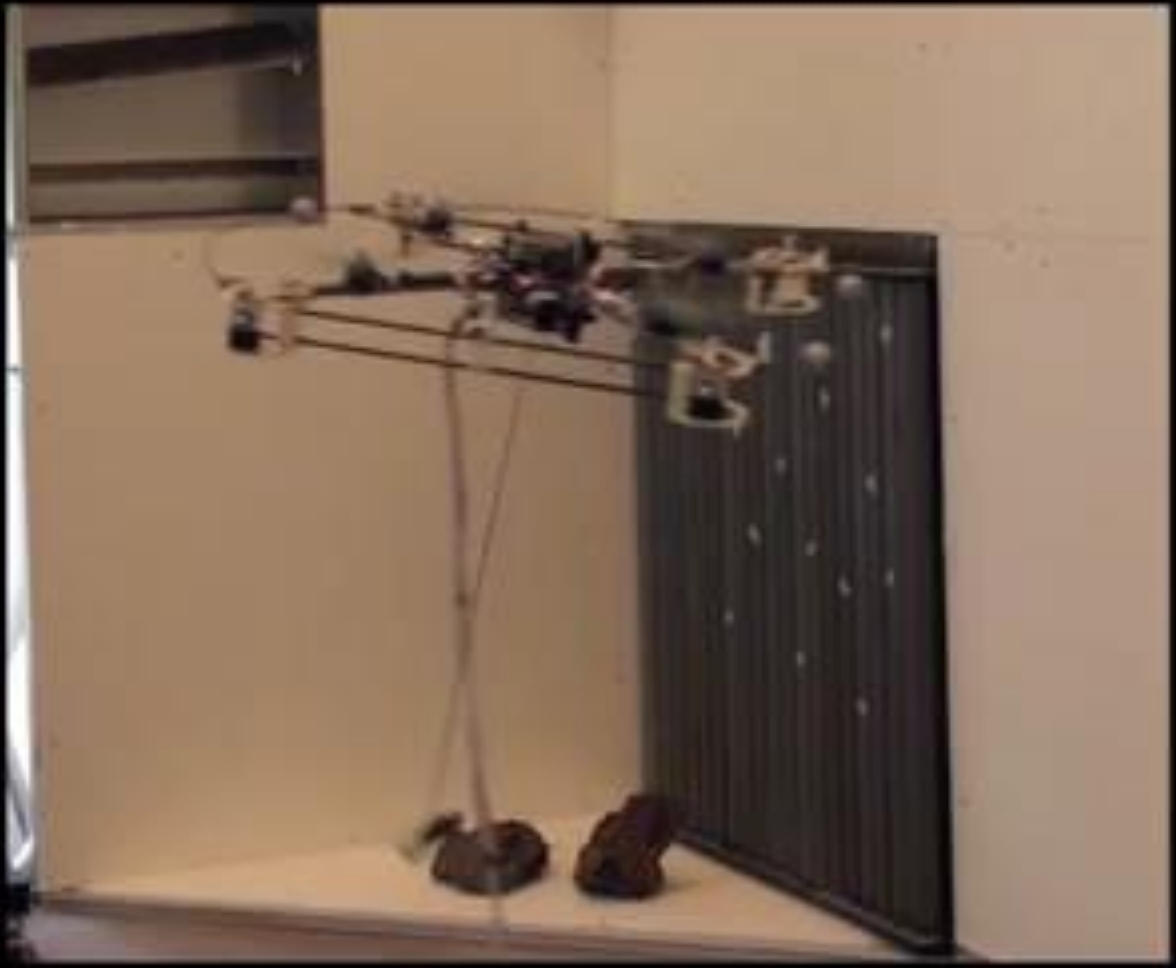
Test Flight Days #9&#10
(May 5th & 6th 2014)

Aircraft: AtlantikSolar UAV Prototype
Location: Tuggen, Switzerland
Flights performed: 5
Tests: Autopilot Waypoint-following

Autonomous Systems Lab

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Slung Load Operations - disturbance of the Load Test-case using the ASLquad



Vehicle: ASLquad

Type of trajectory: Position hold

Nature of disturbance: Disturbance introduced from a 0.16kg slung load which is externally disturbed

Corresponding results using the UPAT-TTR unmanned rotorcraft are also presented later on in the same video sequence

ETH Zürich

AERO WORKS

ETH Manipulator

Find out more

- http://www.kostasalexis.com/pid-control.html
- http://www.kostasalexis.com/lqr-control.html
- http://www.kostasalexis.com/linear-model-predictive-control.html
- http://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum&section=ControlStateSpace

- http://www.kostasalexis.com/literature-and-links.html

Thank you!

Please ask your question!