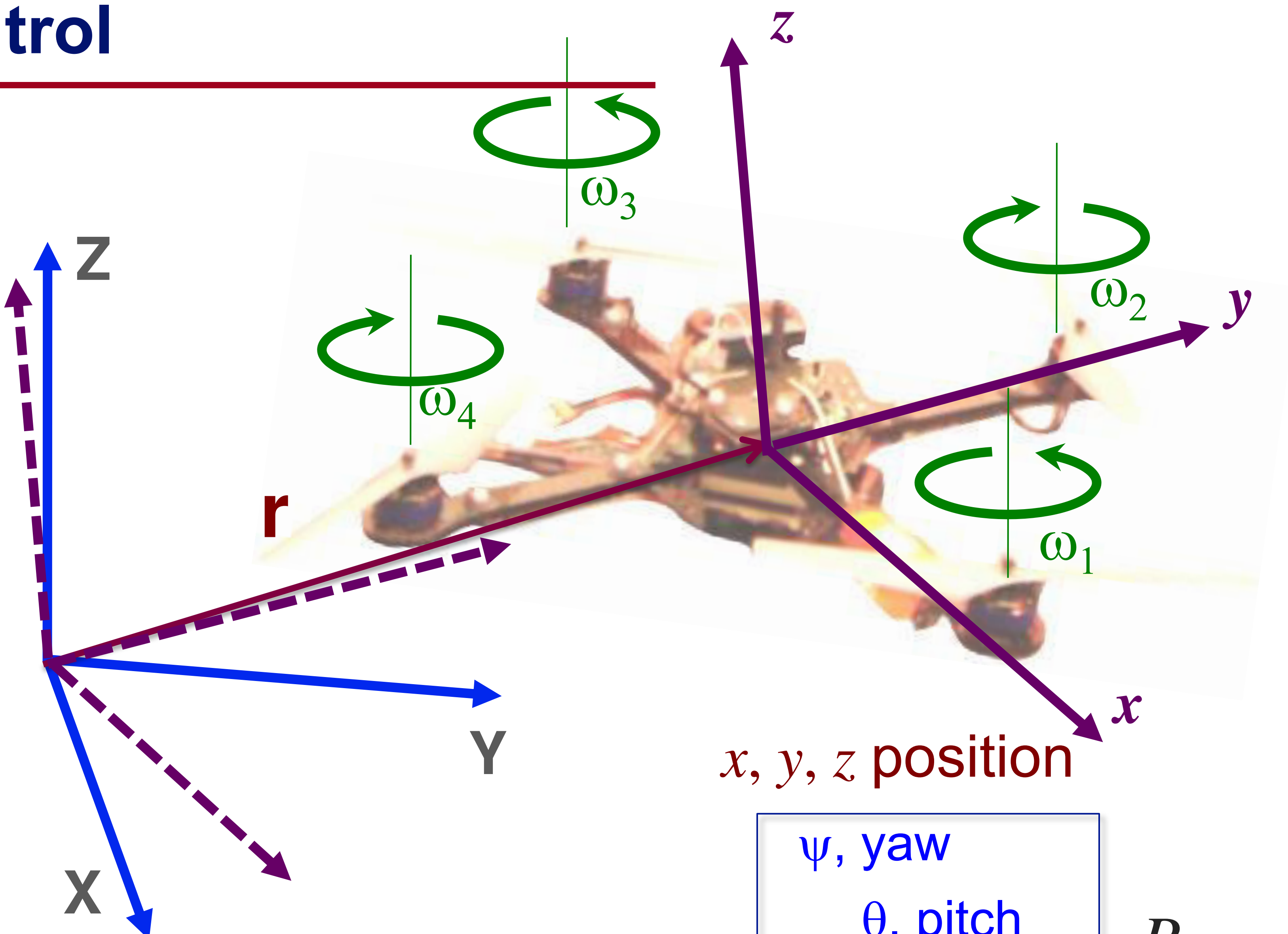
Video 10.1
Vijay Kumar

Control



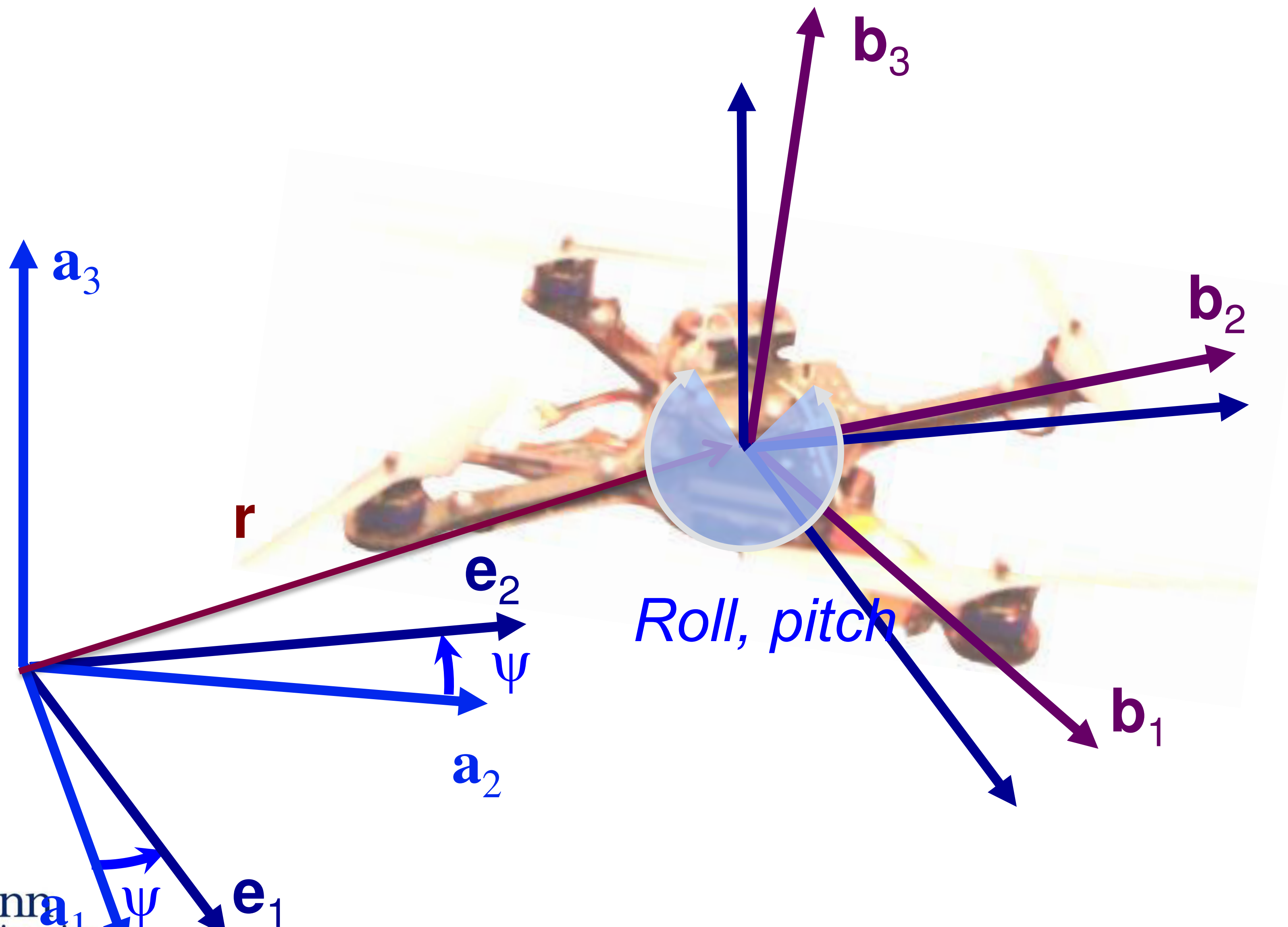
x, y, z position

ψ, yaw
θ, pitch
ϕ, roll

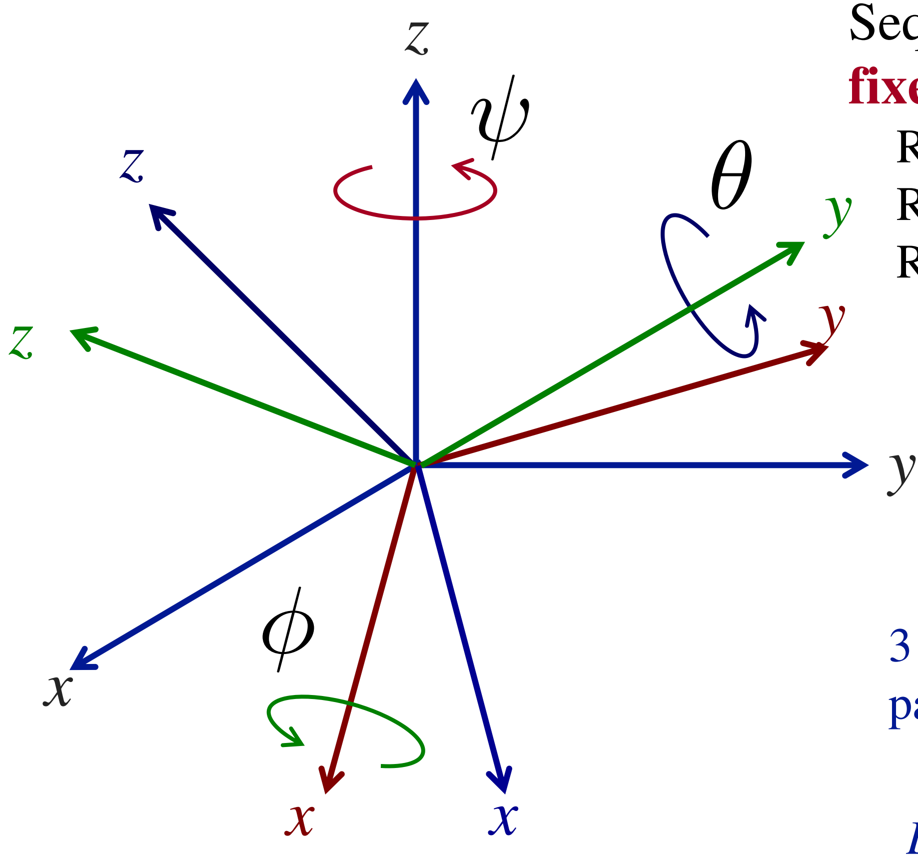
R

Underactuated

Euler Angles



Z-X-Y Euler Angles



Sequence of three rotations about **body-fixed** axes

Rot(z, ψ)	*yaw*
Rot(x, ϕ)	*roll*
Rot(y, θ)	*pitch*

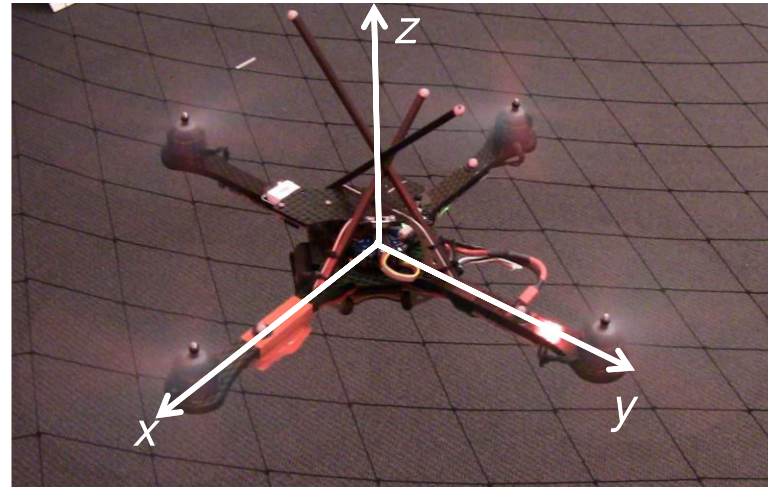
3 Euler angles can be used to parameterize the rotation matrix

But, there are singularities!

$$\mathbf{R} = \mathrm{Rot}(z, \psi) \times \mathrm{Rot}(x, \phi) \times \mathrm{Rot}(y, \theta)$$

Z-X-Y Euler Angles

When are these Euler angles singular?

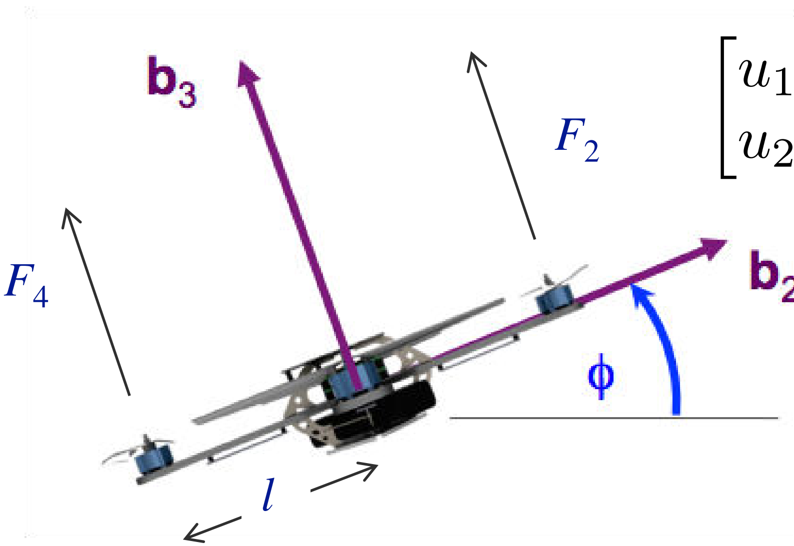


$$\mathbf{R} = \mathrm{Rot}(z, \psi) \times \mathrm{Rot}(x, \phi) \times \mathrm{Rot}(y, \theta)$$

$$R = \begin{bmatrix} c\psi c\theta - s\phi s\psi s\theta & -c\phi s\psi & c\psi s\theta + c\theta s\phi s\psi \\ c\theta s\psi + c\psi s\phi s\theta & c\phi c\psi & s\psi s\theta - c\theta s\phi c\psi \\ -c\phi s\theta & s\phi & c\phi c\theta \end{bmatrix}$$

N. Michael, D. Mellinger, Q. Lindsey, V. Kumar, *The GRASP Multiple Micro-UAV Testbed*, IEEE Robotics & Automation Magazine, vol.17, no.3, pp.56-65, Sept. 2010

Planar Model



$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} F_2 + F_4 \\ (F_2 - F_4)l \end{bmatrix}$$

State Equations

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

Outputs

$$\mathbf{y} = h(x) = \begin{bmatrix} y \\ z \end{bmatrix}$$

Planar Quadrotor

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

$$\mathbf{y} = h(x) = \begin{bmatrix} y \\ z \end{bmatrix}$$

Repeated differentiation of $h(x)$ does not yield explicit dependence on u

The system is not input output linearizable!

Planar Quadrotor

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

$$\mathbf{y} = h(x) = \begin{bmatrix} y \\ z \end{bmatrix}$$

Repeated differentiation of $h(x)$ does not yield explicit dependence on u

Can extend state with higher order derivatives of input

New extended state

$$\bar{x} = \begin{bmatrix} y & z & \phi & \dot{y} & \dot{z} & \dot{\phi} & u_1 & \dot{u}_1 \end{bmatrix}^T$$

New input

$$\bar{u} = \begin{bmatrix} \bar{u}_1 \\ \bar{u}_2 \end{bmatrix} = \begin{bmatrix} \ddot{u}_1 \\ u_2 \end{bmatrix}$$

Penn Engineering

Planar Quadrotor

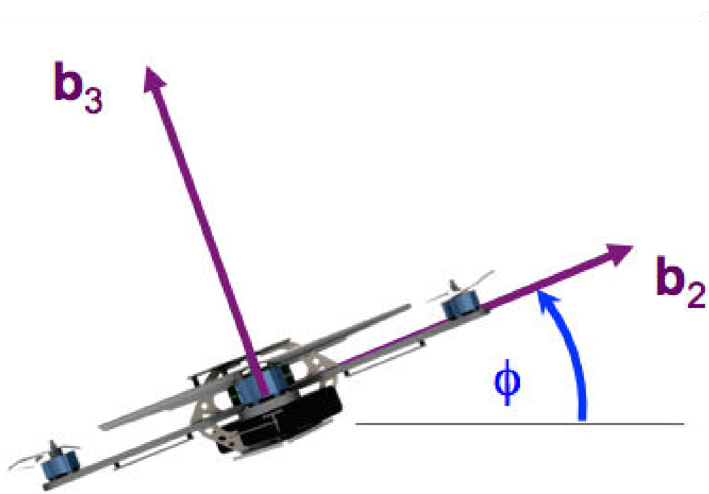
$$\dot{x} = \begin{bmatrix} \dot{y} \\ \dot{z} \\ \dot{\phi} \\ 0 \\ -g \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -\frac{1}{m}\sin\phi & 0 \\ \frac{1}{m}\cos\phi & 0 \\ 0 & \frac{1}{I_{xx}} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$\mathbf{y} = h(x) = \begin{bmatrix} y \\ z \end{bmatrix}$$

$$\dot{\bar{x}} = \underbrace{\begin{bmatrix} \dot{y} \\ \dot{z} \\ \dot{\phi} \\ -\frac{u_1}{m}\sin\phi \\ \frac{u_1}{m}\cos\phi - g \\ 0 \\ \dot{u}_1 \\ 0 \end{bmatrix}}_{\bar{f}(\bar{x})} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{1}{I_{xx}} \\ 0 & 0 \\ 1 & 0 \end{bmatrix}}_{\bar{g}(\bar{x})} \begin{bmatrix} \bar{u}_1 \\ \bar{u}_2 \end{bmatrix}$$

Verify $\mathcal{L}_{\bar{g}}\mathcal{L}_{\bar{f}}^{3}h$ is full rank ($r = 4$)

Relative Degree of Freedom is 4



$$\mathbf{z} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \mathbf{z}_3 \\ \mathbf{z}_4 \end{bmatrix} = \begin{bmatrix} y \\ z \\ \dot{y} \\ \dot{z} \\ \ddot{y} \\ \ddot{z} \\ \dddot{y} \\ \dddot{z} \end{bmatrix}$$

$$\begin{bmatrix} y^{(iv)} \\ z^{(iv)} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} -\sin\phi & -\frac{u_1}{I_{zz}}\cos\phi \\ \cos\phi & -\frac{u_1}{I_{zz}}\sin\phi \end{bmatrix} \begin{bmatrix} \ddot{u}_1 \\ u_2 \end{bmatrix} + \frac{1}{m} \begin{bmatrix} -2\dot{u}_1 \cos\phi\dot{\phi} + u_1\dot{\phi}^2 \sin\phi \\ -2\dot{u}_1 \sin\phi\dot{\phi} - u_1\dot{\phi}^2\cos\phi \end{bmatrix}$$

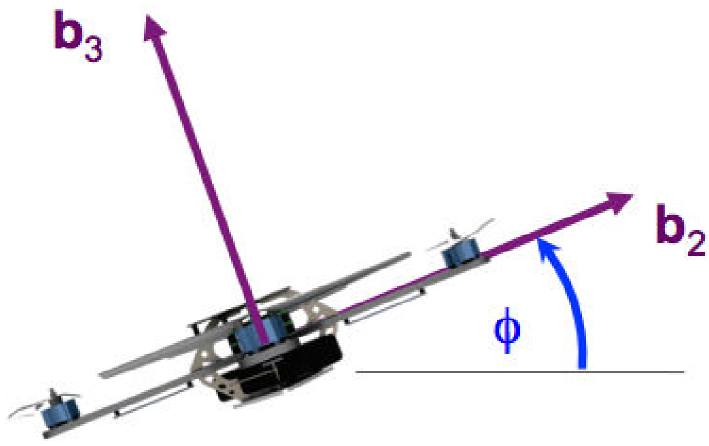
$$h^{(iv)} \qquad \mathcal{L}_{\bar{g}}\mathcal{L}_{\bar{f}}^3 h \qquad \bar{u} \qquad \mathcal{L}_{\bar{f}}^4 h$$

Correction:

Video 10.1 - This slide has been corrected to reflect this change:

Typo @8:20: the (2,1) element in the matrix $\begin{pmatrix} -\sin\phi & -\frac{u_1}{I_{xx}}\cos\phi \\ -\cos\phi & -\frac{u_1}{I_{xx}}\sin\phi \end{pmatrix}$ should be $\cos\phi$ instead of $-\cos\phi$

Dynamic State Feedback



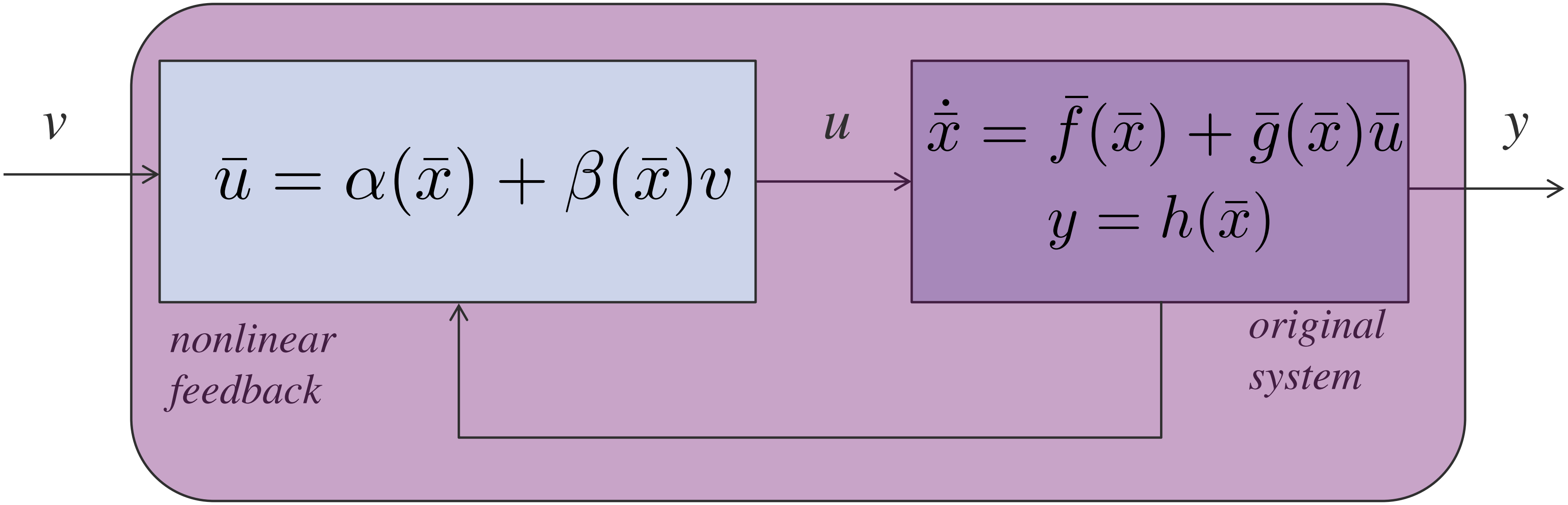
$$\mathbf{z} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \mathbf{z}_3 \\ \mathbf{z}_4 \end{bmatrix} = \begin{bmatrix} y \\ z \\ \dot{y} \\ \dot{z} \\ \ddot{y} \\ \ddot{z} \\ \dddot{y} \\ \dddot{z} \end{bmatrix}$$

$$\begin{bmatrix} y^{(iv)} \\ z^{(iv)} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} -\sin\phi & -\frac{u_1}{I_{zz}}\cos\phi \\ -\cos\phi & -\frac{u_1}{I_{zz}}\sin\phi \end{bmatrix} \begin{bmatrix} \ddot{u}_1 \\ u_2 \end{bmatrix} + \frac{1}{m} \begin{bmatrix} -2\dot{u}_1 \cos\phi\dot{\phi} + u_1\dot{\phi}^2 \sin\phi \\ -2\dot{u}_1 \sin\phi\dot{\phi} - u_1\dot{\phi}^2 \cos\phi \end{bmatrix}$$

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

Input-Output Linearization

$$\bar{x} = \begin{bmatrix} y & z & \phi & \dot{y} & \dot{z} & \dot{\phi} & u_1 & \dot{u}_1 \end{bmatrix}^T$$



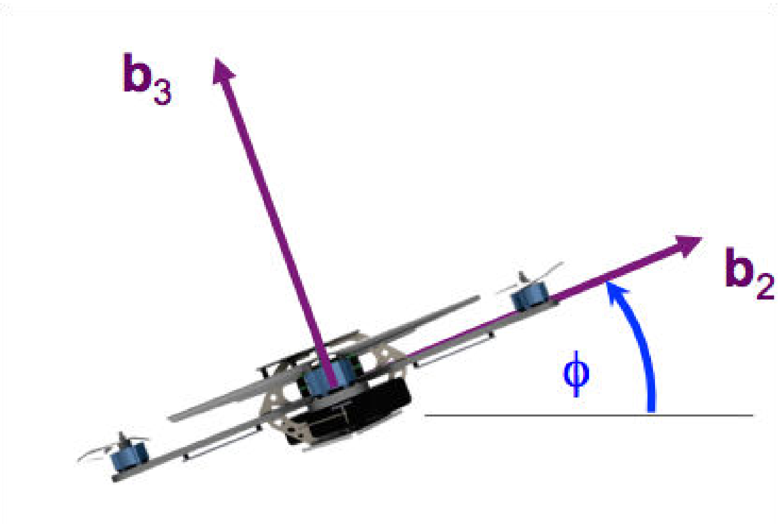
new system

$$y^{(iv)} = v$$

Nonlinear feedback transforms the original nonlinear system to a new linear system

Linearization is exact (distinct from linear approximations to nonlinear systems)

Linear System



$$\mathbf{z} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \mathbf{z}_3 \\ \mathbf{z}_4 \end{bmatrix} = \begin{bmatrix} y \\ z \\ \dot{y} \\ \dot{z} \\ \ddot{y} \\ \ddot{z} \\ \dddot{y} \\ \dddot{z} \end{bmatrix}$$

We can design a linear controller to drive the system along any smooth $\mathbf{z}(t)$

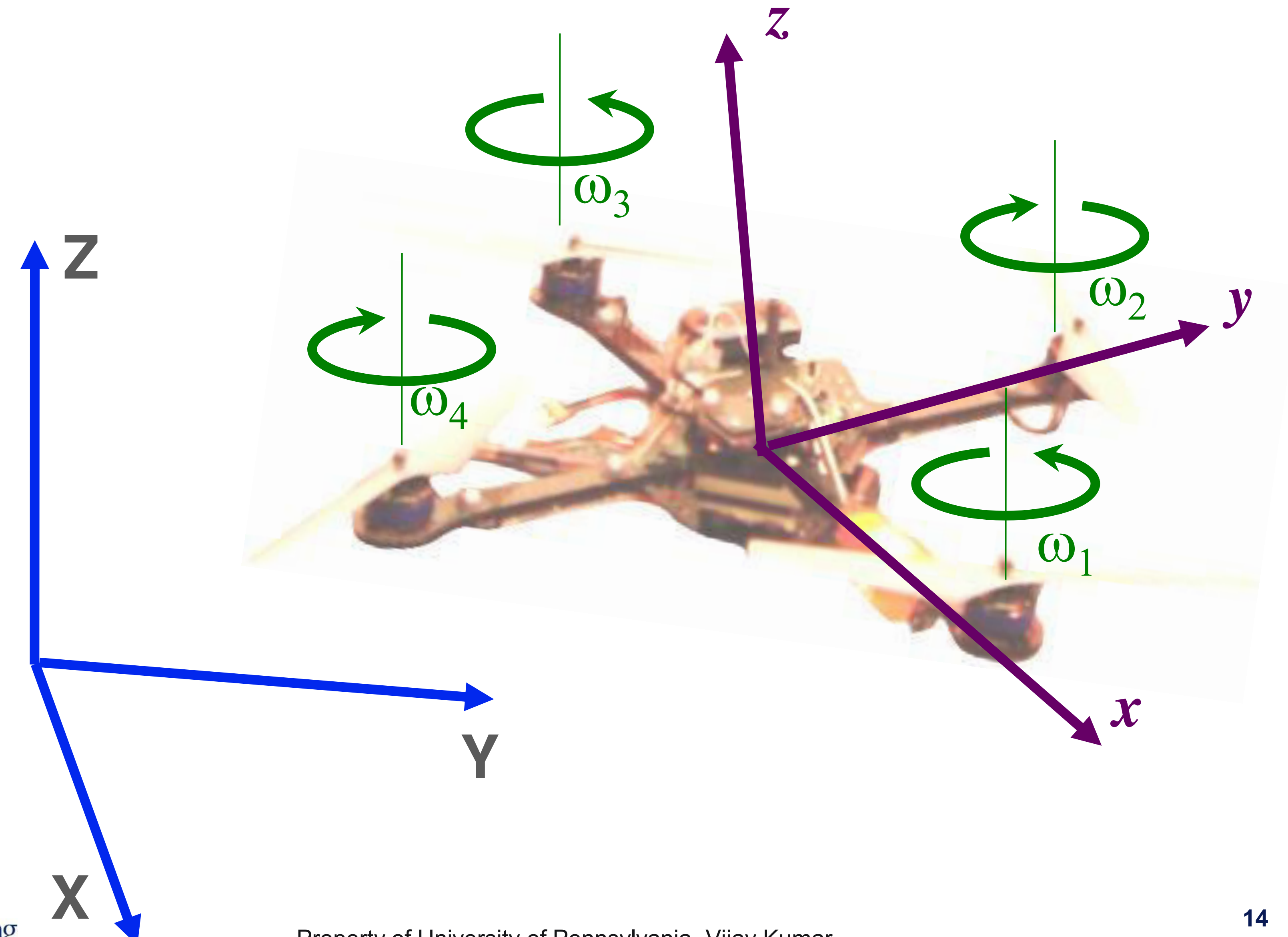
$$\begin{bmatrix} y^{(iv)} \\ z^{(iv)} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} -\sin\phi & -\frac{u_1}{I_{zz}}\cos\phi \\ -\cos\phi & -\frac{u_1}{I_{zz}}\sin\phi \end{bmatrix} \begin{bmatrix} \ddot{u}_1 \\ u_2 \end{bmatrix} + \frac{1}{m} \begin{bmatrix} -2\dot{u}_1\cos\phi\dot{\phi} + u_1\dot{\phi}^2\sin\phi \\ -2\dot{u}_1\sin\phi\dot{\phi} - u_1\dot{\phi}^2\cos\phi \end{bmatrix}$$

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

$$\dot{\mathbf{z}} = \begin{bmatrix} \mathbf{0}_{2\times2} & \mathbf{I}_{2\times2} & \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} \\ \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} & \mathbf{I}_{2\times2} & \mathbf{0}_{2\times2} \\ \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} & \mathbf{I}_{2\times2} \\ \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} \end{bmatrix} \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \mathbf{z}_3 \\ \mathbf{z}_4 \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{2\times2} \\ \mathbf{0}_{2\times2} \\ \mathbf{0}_{2\times2} \\ \mathbf{I}_{2\times2} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

v=Kz

A similar approach can be used for 3-D quadrotors



Video 10.2
Vijay Kumar

Differential Flatness

All state variables and the inputs can be written as smooth functions of *flat outputs* and their derivatives

$$\begin{bmatrix} y \\ z \\ \dot{y} \\ \dot{z} \\ \ddot{y} \\ \ddot{z} \\ \dddot{y} \\ \dddot{z} \\ y^{(iv)} \\ z^{(iv)} \end{bmatrix} \overset{\text{diffeomorphism}}{\longleftrightarrow} \begin{bmatrix} y \\ z \\ \phi \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ u_1 \\ \dot{u}_1 \\ \ddot{u}_1 \\ u_2 \end{bmatrix}$$

Differential Flatness (Murray *et al*, 1995)

Planar Quadrotor

The flat outputs and their derivatives can be written as a function of the state, the inputs, and their derivatives

Flat outputs	State	Input
$\begin{bmatrix} y \\ z \end{bmatrix}$	$\begin{bmatrix} y \\ z \\ \phi \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \end{bmatrix}$	$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$

$$\begin{bmatrix} \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} -\frac{1}{m}\sin\phi \\ \frac{1}{m}\cos\phi \end{bmatrix} u_1$$

$$\begin{bmatrix} y^{(iii)} \\ z^{(iii)} \end{bmatrix} = \frac{1}{m}\begin{bmatrix} -u_1\dot{\phi}\cos\phi - \dot{u}_1\sin\phi \\ -u_1\dot{\phi}\sin\phi + \dot{u}_1\cos\phi \end{bmatrix}$$

$$\begin{bmatrix} y^{(iv)} \\ z^{(iv)} \end{bmatrix} = \frac{1}{m}\begin{bmatrix} -\sin\phi & -\frac{u_1}{I_{xx}}\cos\phi \\ \cos\phi & -\frac{u_1}{I_{xx}}\sin\phi \end{bmatrix}\begin{bmatrix} \ddot{u}_1 \\ u_2 \end{bmatrix} + \frac{1}{m}\begin{bmatrix} -2\dot{u}_1\dot{\phi}\cos\phi + u_1\dot{\phi}^2\sin\phi \\ -2\dot{u}_1\dot{\phi}\sin\phi - u_1\dot{\phi}^2\cos\phi \end{bmatrix}$$

Planar Quadrotor

The state, the inputs, and their derivatives can be written as a function of the flat outputs and their derivatives

Flat outputs

$$\begin{bmatrix} y \\ z \end{bmatrix}$$

State

$$\begin{bmatrix} y \\ z \\ \phi \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \end{bmatrix}$$

Input

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$u_1 = m\left(\ddot{y}^2 + \ddot{z}^2\right)$$

$$\phi = \operatorname{atan2}\left(-\frac{m\ddot{y}}{u_1}, \frac{m\ddot{z}}{u_1}\right)$$

$$\dot{u}_1 = m(-y^{(iii)} \sin\phi + z^{(iii)} \cos\phi)$$

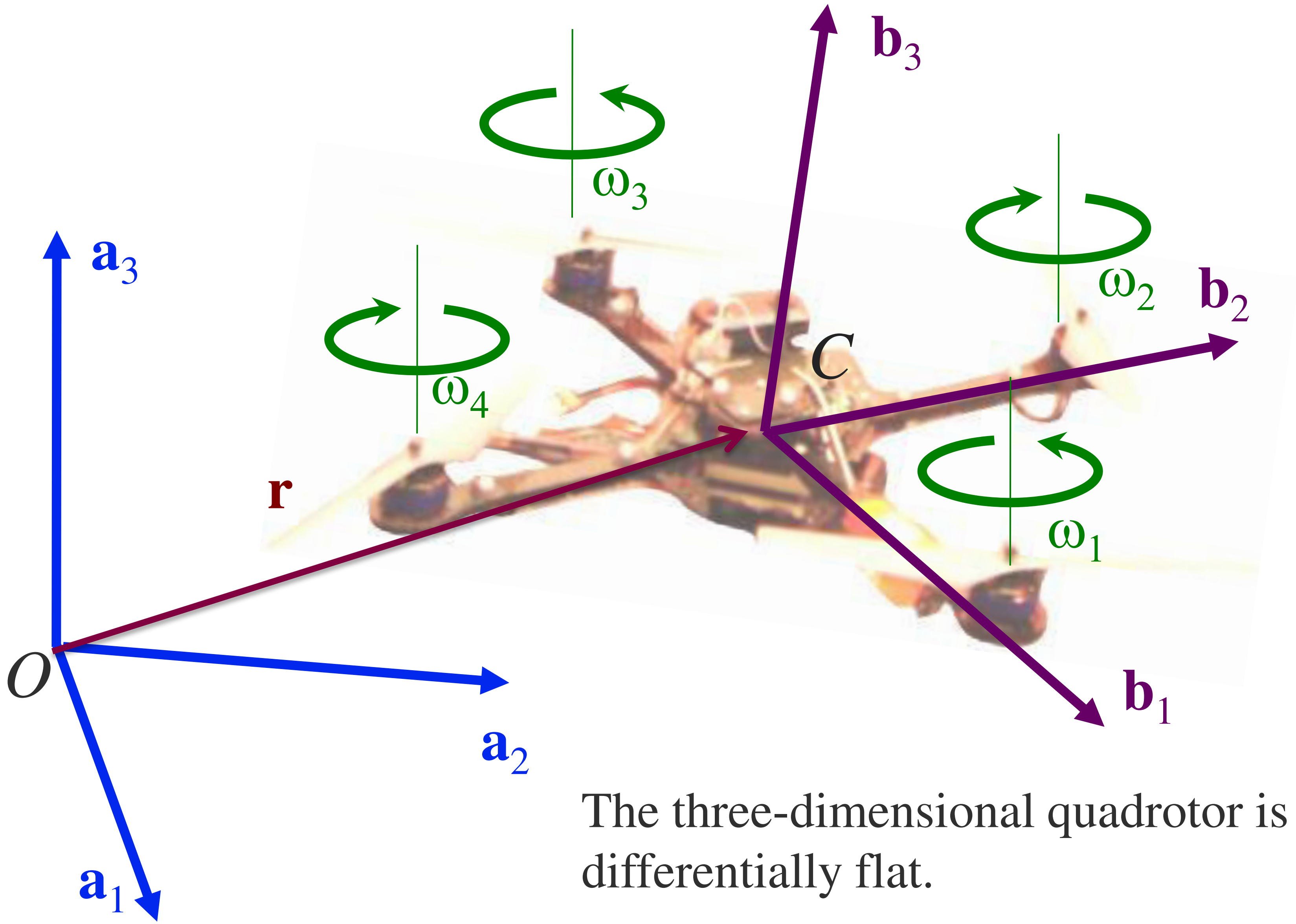
$$\dot{\phi} = \frac{-m}{u_1}\left(y^{(iii)} \cos\phi + z^{(iii)} \sin\phi\right)$$

$$\ddot{u}_1 = \ldots$$

$$\ddot{\phi} = \ldots$$

$$u_2 = \ldots$$

Penn Engineering



The three-dimensional quadrotor is differentially flat.

Differential Flatness (3-D Quadrotor)

Inputs

$u_1, \mathbf{u}_2$

$$u_1 = \sum_{i=1}^{4} F_i \qquad \mathbf{u}_2 = L \begin{bmatrix} 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ \mu & -\mu & \mu & -\mu \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix}$$

State

$(\mathbf{x}, \dot{\mathbf{x}})$

$\mathbf{r}_T$

$$\begin{bmatrix} \mathbf{r} \\ \mathbf{v} \\ \mathbf{a} \\ \mathbf{j} \\ \mathbf{s} \\ \psi \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix} \longleftrightarrow \begin{bmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \\ u_1 \\ \dot{u}_1 \\ \ddot{u}_1 \\ \mathbf{u}_2 \end{bmatrix}$$

jerk ($\mathbf{j}$), *snap* ($\mathbf{s}$), *yaw* (ψ)

$$u_1 = m(a_3 - \mathbf{g}.\mathbf{b}_3)$$

$$\dot{u}_1 = m j_3$$

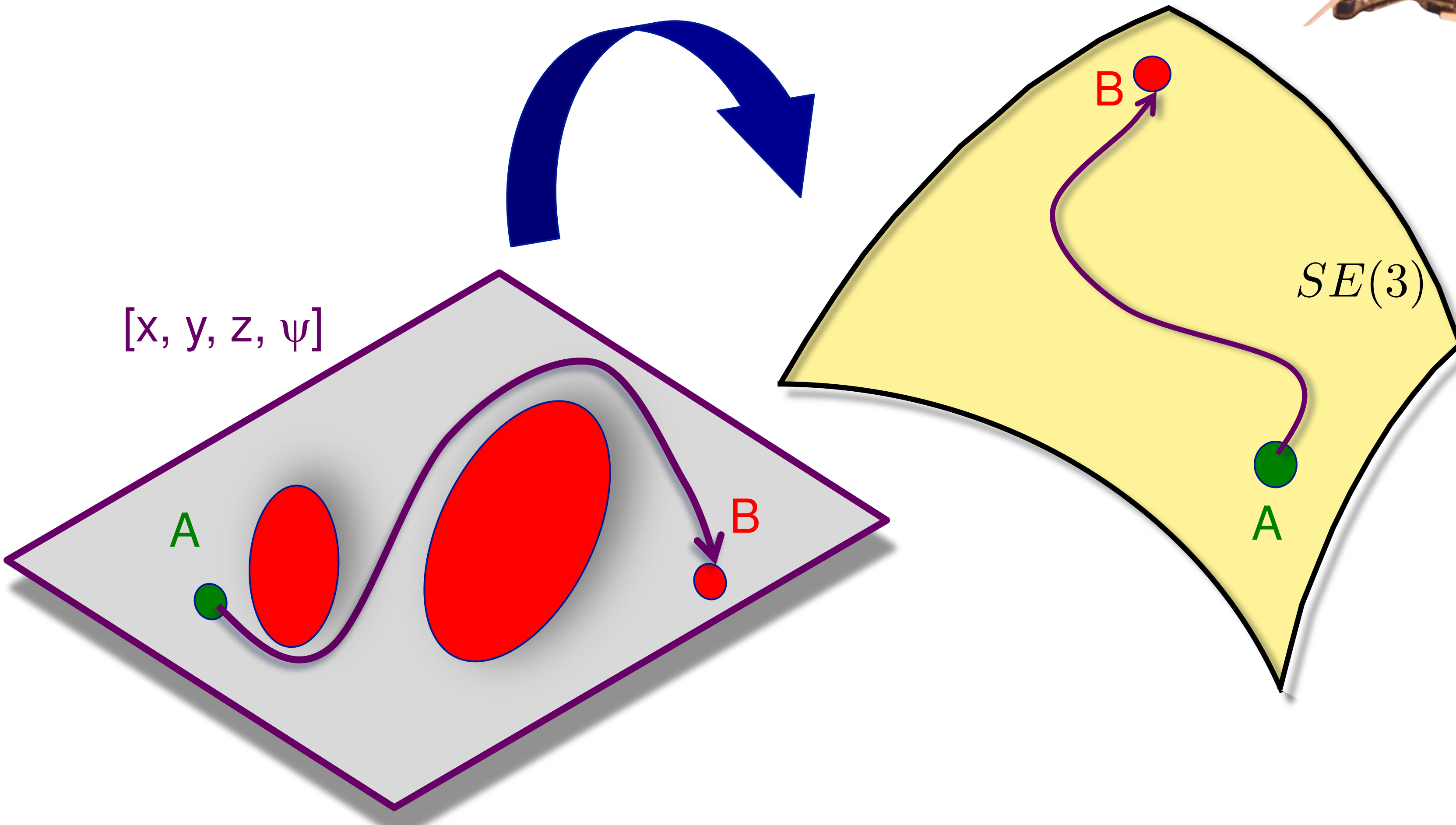
$$\ddot{u}_1 = m s_3 + u_1(q^2 + rp)$$

$$p = \frac{-m j_2}{u_1}$$

$$q = \frac{m j_1}{u_1}$$

D. Mellinger and V. Kumar, "Minimum Snap Trajectory Generation and Control for Quadrotors," *Proc. IEEE Int. Conf. on Robotics and Automation*. May, 2011.

Trajectory Planner



$$\begin{bmatrix} x \\ y \\ z \\ \theta \\ \phi \\ \psi \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{\phi} \end{bmatrix}$$

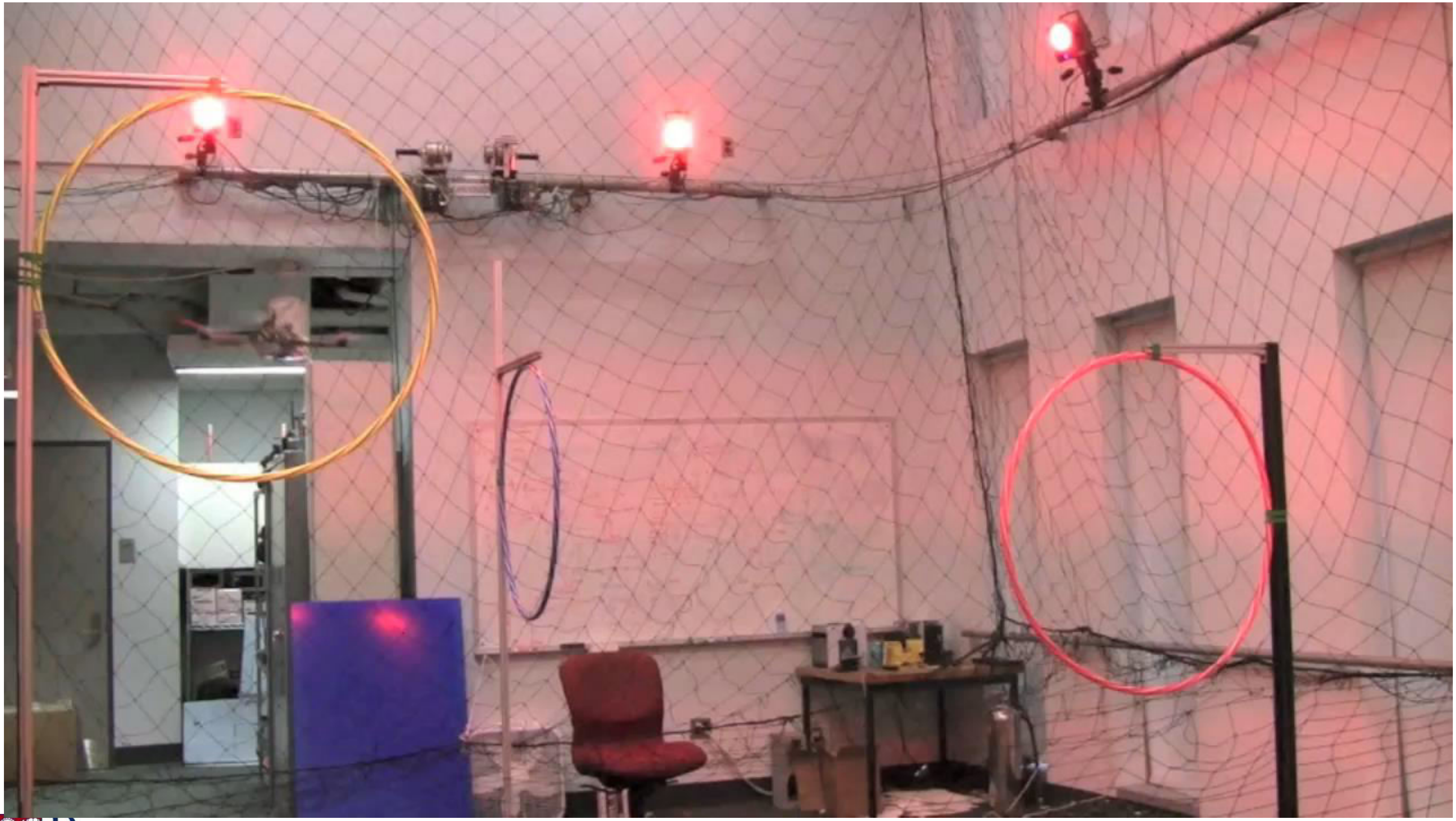
Minimum snap trajectory

$$\min_{\sigma(t)} \int_0^T \alpha \| \ddot{\ddot{\mathbf{r}}}(t) \|^2 + \beta \ddot{\psi}(t)^2 dt$$

Penn Engineering

Minimum Snap Trajectory

Minimum Snap Trajectory

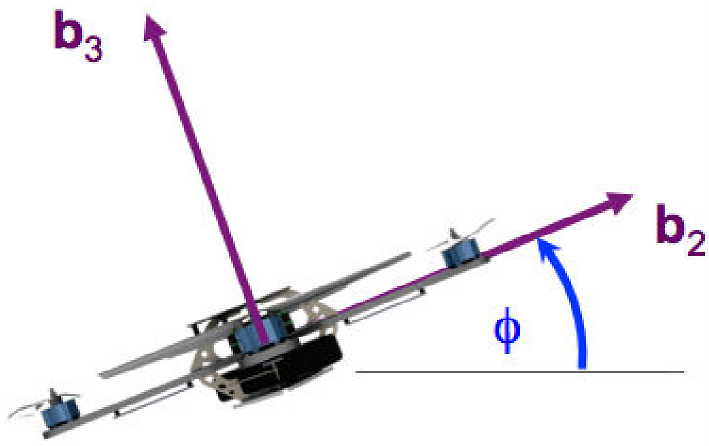


Penn Engineering

ONLINE LEARNING

Video 10.3
Vijay Kumar

Input Output Linearization



$$\mathbf{z} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \mathbf{z}_3 \\ \mathbf{z}_4 \end{bmatrix} = \begin{bmatrix} y \\ z \\ \dot{y} \\ \dot{z} \\ \ddot{y} \\ \ddot{z} \\ \dddot{y} \\ \dddot{z} \end{bmatrix}$$

$$\begin{bmatrix} y^{(iv)} \\ z^{(iv)} \end{bmatrix} = \frac{1}{m}\begin{bmatrix} -\sin\phi & -\frac{u_1}{I_{zz}}\cos\phi \\ -\cos\phi & -\frac{u_1}{I_{zz}}\sin\phi \end{bmatrix} \begin{bmatrix} \ddot{u}_1 \\ u_2 \end{bmatrix} + \frac{1}{m}\begin{bmatrix} -2\dot{u}_1\cos\phi\dot{\phi} + u_1\dot{\phi}^2\sin\phi \\ -2\dot{u}_1\sin\phi\dot{\phi} - u_1\dot{\phi}^2\cos\phi \end{bmatrix}$$

$h^{(iv)}$ $\qquad \mathcal{L}_{\bar{g}}\mathcal{L}_{\bar{f}}^3 h$ $\qquad \bar{u}$ $\qquad \mathcal{L}_{\bar{f}}^4 h$

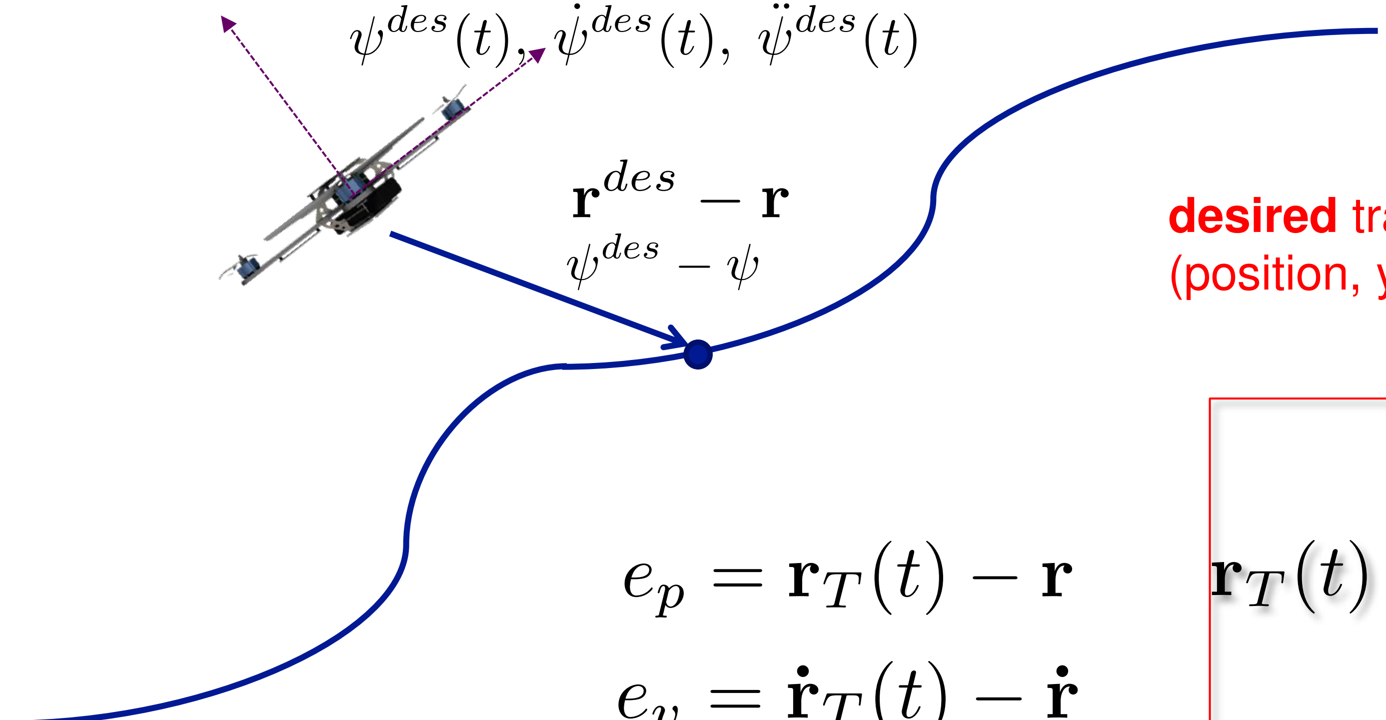
$$\bar{u} = \left(\mathcal{L}_{\bar{g}}\mathcal{L}_{\bar{f}}^3 h\right)^{-1} v - \left(\mathcal{L}_{\bar{g}}\mathcal{L}_{\bar{f}}^3 h\right)^{-1} \mathcal{L}_{\bar{f}}^4 h$$

Trajectory Tracking

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

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

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



$$e_p = \mathbf{r}_T(t) - \mathbf{r}$$

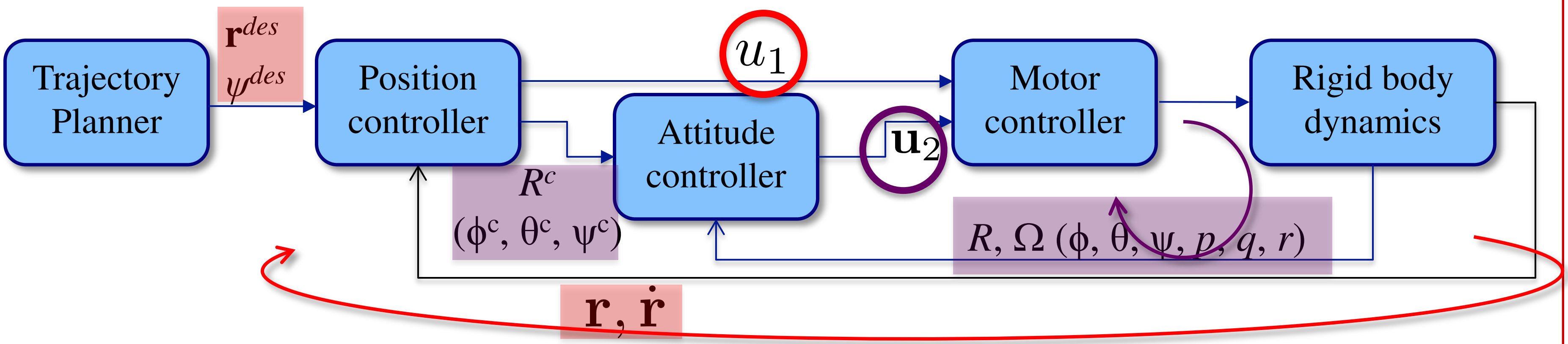
$$e_v = \dot{\mathbf{r}}_T(t) - \dot{\mathbf{r}}$$

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

Want $(\ddot{\mathbf{r}}_T(t) - \ddot{\mathbf{r}}_c) + k_{d,x} e_v + k_{p,x} e_p = 0$

Commanded acceleration, calculated by the controller

Penn Engineering



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

u_1

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

$\mathbf{u}_2$

Trajectory Tracking

$$r_T(t) = \begin{bmatrix} x^{des}(t) \\ y^{des}(t) \\ z^{des}(t) \\ \psi^{des}(t) \end{bmatrix}$$

$\mathbf{t}$

$$u_1 = m\left(\ddot{r}^{des} + K_v \mathbf{e}_{\dot{r}} + K_p \mathbf{e}_r + g\mathbf{a}_3\right).\mathbf{R}\mathbf{b}_3$$

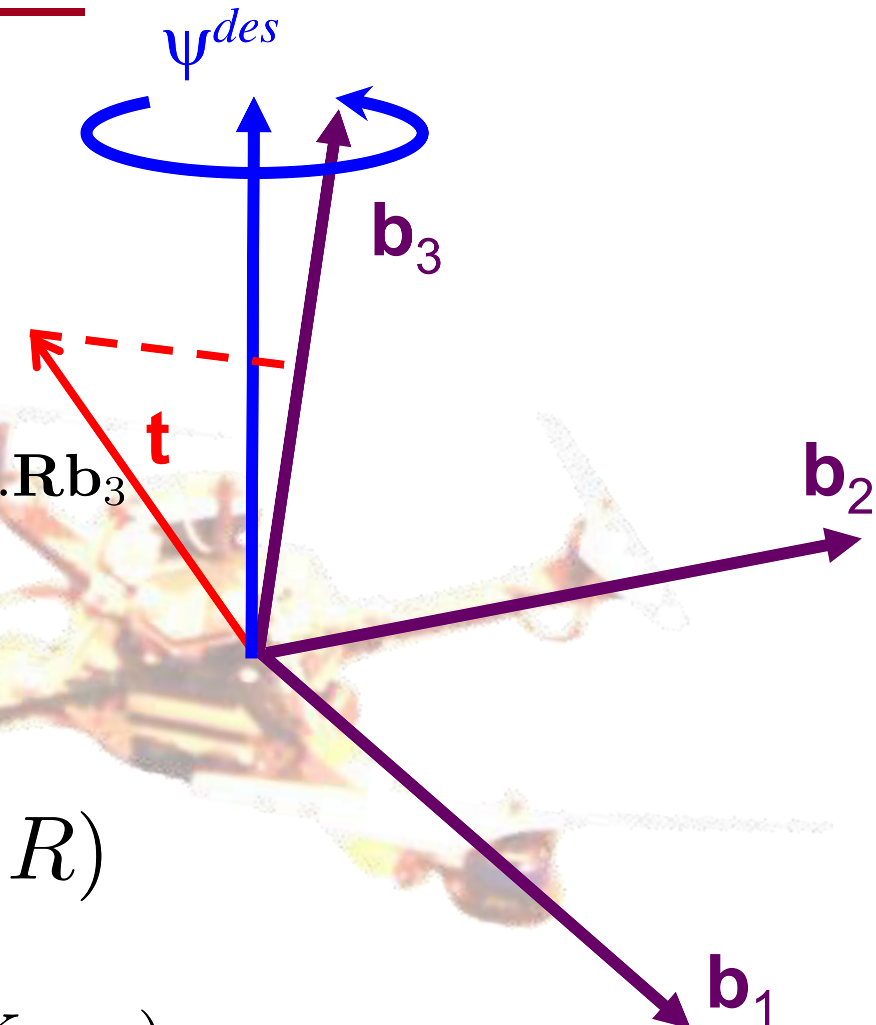
$$\mathbf{R}^{des}\mathbf{b}_3 = \frac{\mathbf{t}}{\|\mathbf{t}\|}$$

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

$$R^{des}\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$e_R(R^{des}, R)$$

$$\mathbf{u}_2 = \omega \times \mathbf{I}\omega + \mathbf{I}\left(-K_R \mathbf{e}_R - K_\omega \mathbf{e}_\omega\right)$$



How to determine $\mathbf{R}^{des}$?

You are given two pieces of information

$$\mathbf{R}^{des}\mathbf{b}_3 = \frac{\mathbf{t}}{\|\mathbf{t}\|}$$

$$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \rightarrow \mathbf{b}_3$$

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

You know that the rotation matrix has the form

$$\mathbf{R} = \begin{bmatrix} c\psi c\theta - s\phi s\psi s\theta & -c\phi s\psi & c\psi s\theta + c\theta s\phi s\psi \\ c\theta s\psi + c\psi s\phi s\theta & c\phi c\psi & s\psi s\theta - c\theta s\phi c\psi \\ -c\phi s\theta & s\phi & c\phi c\theta \end{bmatrix}$$

You should be able to find the roll and pitch angles.

How to calculate the error $\mathbf{e}_R(\mathbf{R}^{des}, \mathbf{R})$?

Cannot simply take the difference of two rotation matrices

What is the magnitude of the rotation required to go from the current orientation to the desired orientation?

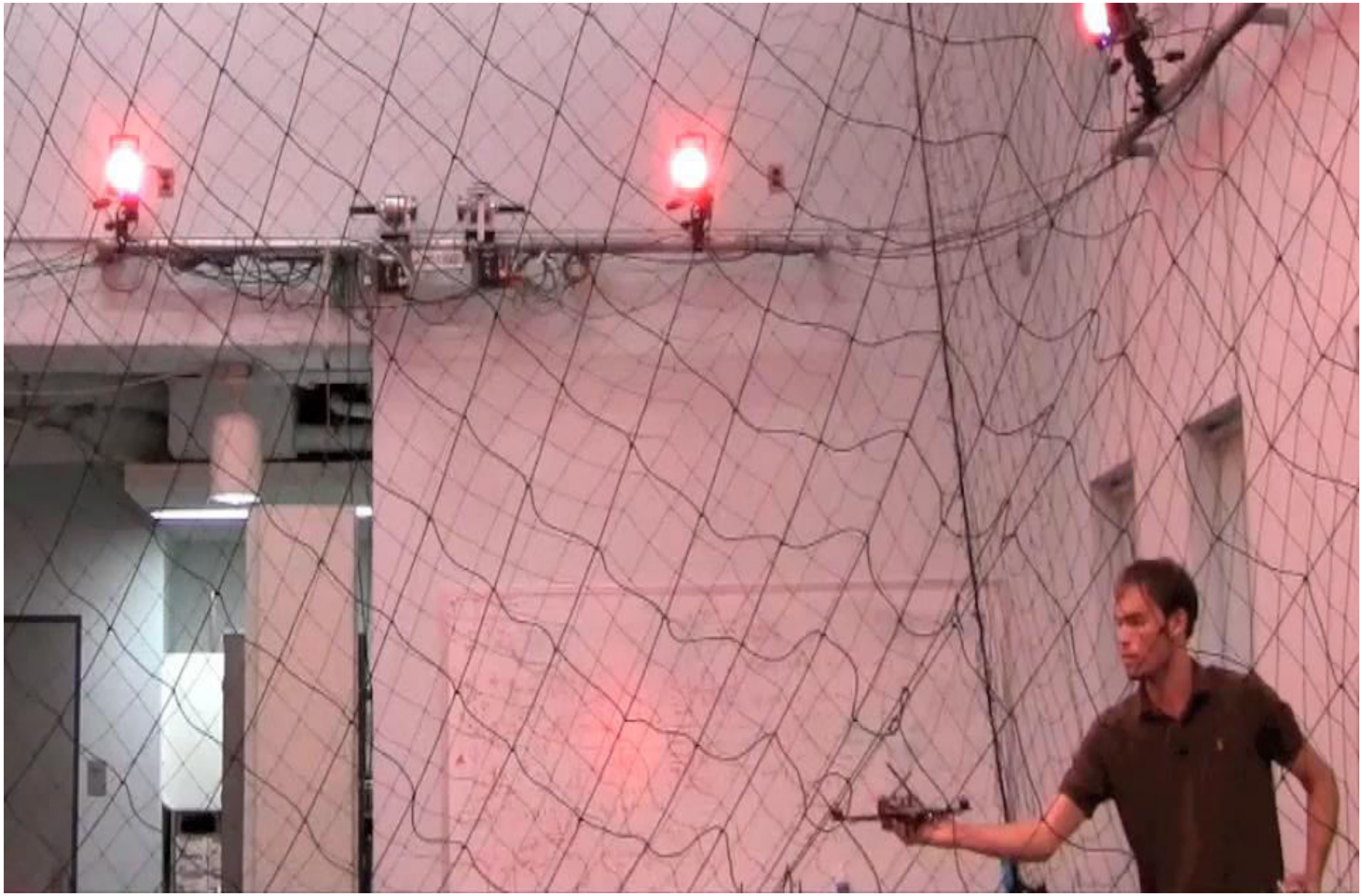
$$\mathbf{R} \rightarrow \mathbf{R}^{des}$$

The required rotation is

$$\Delta R = \mathbf{R}^T \mathbf{R}^{des}$$

The angle and axis of rotation can be determined using Rodrigues formula

Asymptotic Stability



T. Lee, M. Leoky, and N. H. McClamroch, Geometric tracking control of a quadrotor UAV on SE(3), *IEEE Conference on Decision and Control*, 2010.

D. Mellinger and V. Kumar, "Minimum Snap Trajectory Generation and Control for Quadrotors," *Proc. IEEE Int. Conf. on Robotics and Automation*. May, 2011.