16-662: Robot Autonomy

Lecture 2: Modeling and Control

Nathan Michael

January 14, 2015

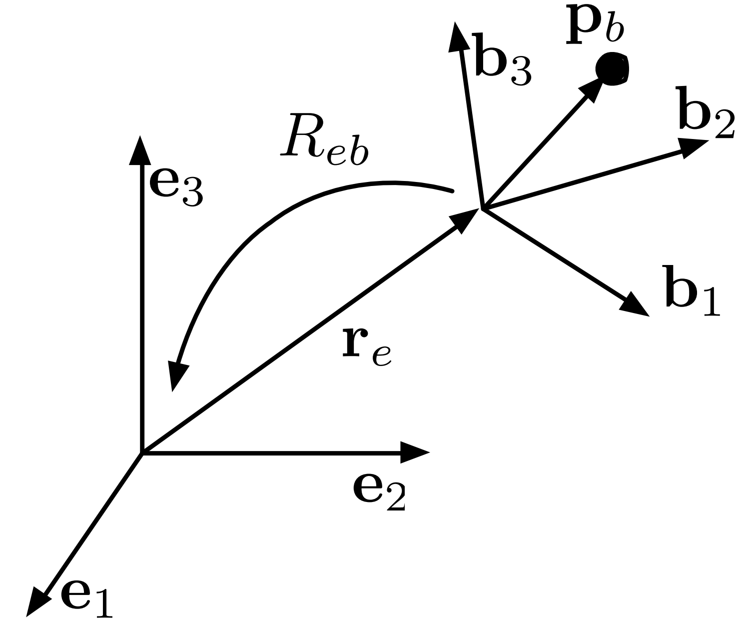
Lecture Outline

- Model
  - Propeller model
  - Force and moments generation
  - Equations of motion
  - Simplifying assumptions
- Control
  - Concept review
  - Attitude control (inner loop)
  - Position control (outer loop)
  - Gain selection
- Project 1A discussion and suggestions

Rigid-Body Transform

$$\mathbf{p}_e = R_{eb}\mathbf{p}_b + \mathbf{r}_e$$

rotation translation



Rotation (special orthogonal group):

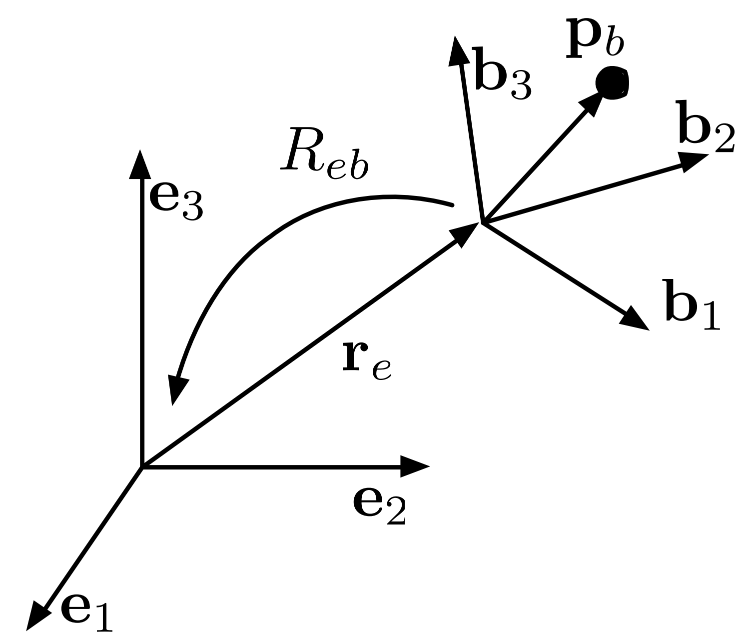
$$SO(3) = \{A \in R^3 | ATA = I, \det(A) = 1\}$$

Parameterizations:

- Euler angles
- Quaternions

Rigid-Body Transform

Euler angle parameterization of rotation:



$$R_{eb} = R_z(\psi) R_y(\theta) R_x(\phi)$$

yaw pitch roll

ZYX (321) form

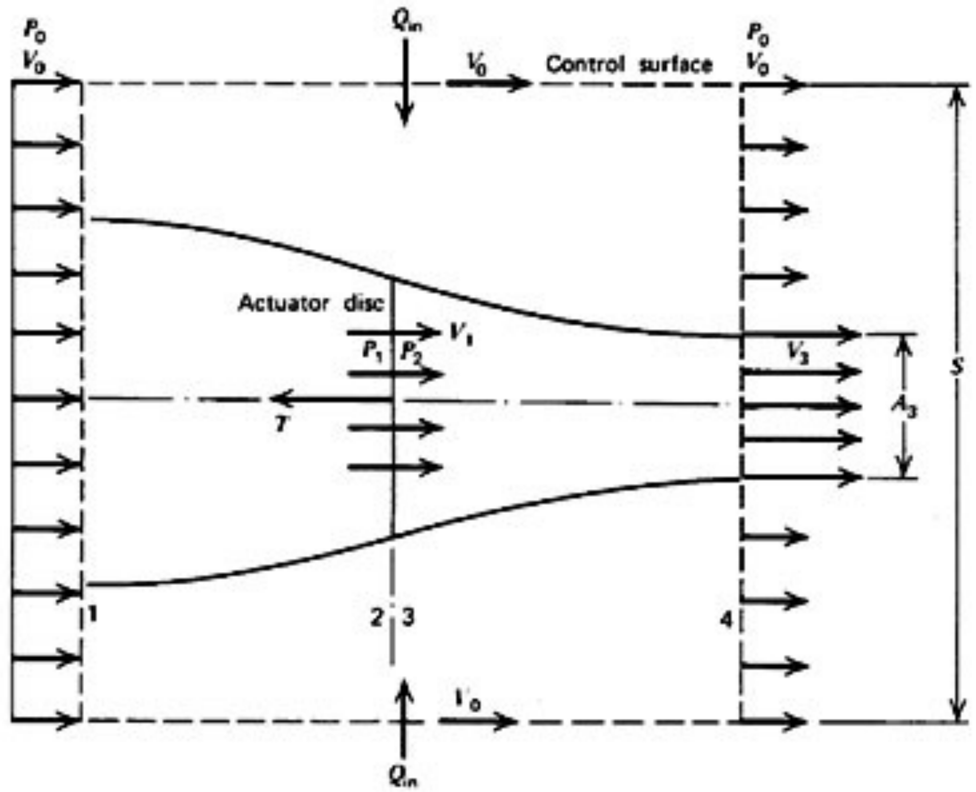
$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix} \quad R_y(\theta) = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \quad R_z(\psi) = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Coordinates (6 DoF):

$$\mathbf{q} = [x,\ y,\ z,\ \phi,\ \theta,\ \psi]^{\mathrm{T}}$$

Propeller and Motor Model

From dimensional analysis:



(McCormick, 1979)

air density (kg/m^3)

diameter (m)

Thrust $$T = k_T \rho n^2 D^4$$

speed (rev/s)

Moment $$Q = k_Q \rho n^2 D^5$$

Motor is a controlled sub-system (model as first-order):

$$\dot{w}_i = k_m \left(w_i^d - w_i \right)$$

Overloaded notation: $\{\omega_x,\, \omega_y,\, \omega_z\}$

Note labels vs enumeration (x vs i)

THE ROBOTICS INSTITUTE

Newton-Euler Equations

total force   mass   linear acceleration

linear velocity

$$\begin{bmatrix} \mathbf{F} \\ \tau \end{bmatrix} = \begin{bmatrix} m\mathbb{1}_3 & \mathbb{0}_3 \\ \mathbb{0}_3 & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \alpha \end{bmatrix} + \begin{bmatrix} \omega \times m\mathbf{v} \\ \omega \times \mathbf{I}_3\omega \end{bmatrix}$$

total torque

inertia tensor

angular velocity

angular acceleration

We could apply the *Euler-Lagrange* equations to the *Lagrangian* for the system but go directly to this form to save time.

Newton-Euler Equations

$$\begin{bmatrix} \mathbf{F} \\ \tau \end{bmatrix} = \begin{bmatrix} m\mathbb{1}_3 & \mathbb{0}_3 \\ \mathbb{0}_3 & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \alpha \end{bmatrix} + \begin{bmatrix} \omega \times m\mathbf{v} \\ \omega \times \mathbf{I}_3\omega \end{bmatrix}$$

Recall matrix form (from 16-642):

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau$$

forces/torques

inertia (mass)

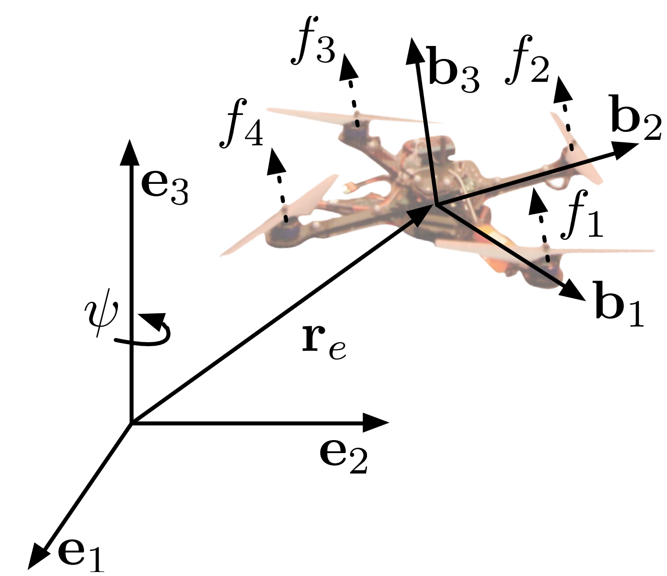
gravity

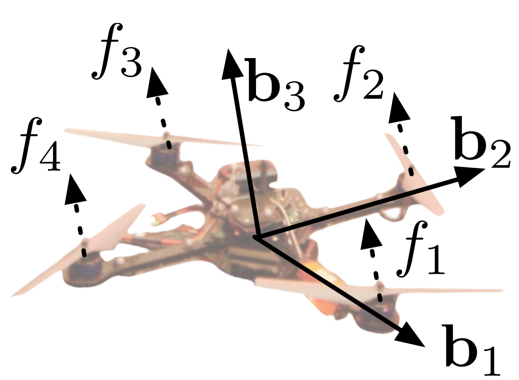
Coriolis/centrifugal

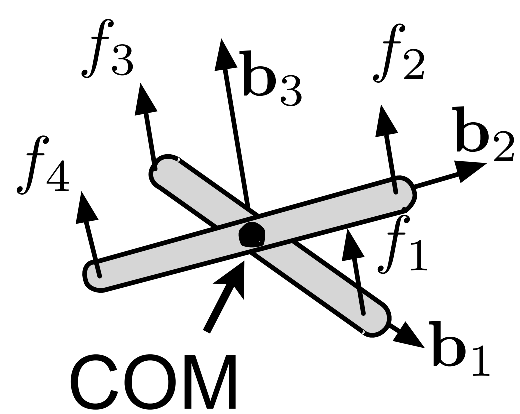
Force-Moment Generation

Newton-Euler equations:

$$\begin{bmatrix} \mathbf{F} \\ \tau \end{bmatrix} = \begin{bmatrix} m\mathbb{1}_3 & \mathbb{0}_3 \\ \mathbb{0}_3 & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \alpha \end{bmatrix} + \begin{bmatrix} \omega \times m\mathbf{v} \\ \omega \times \mathbf{I}_3\omega \end{bmatrix}$$







Total force:

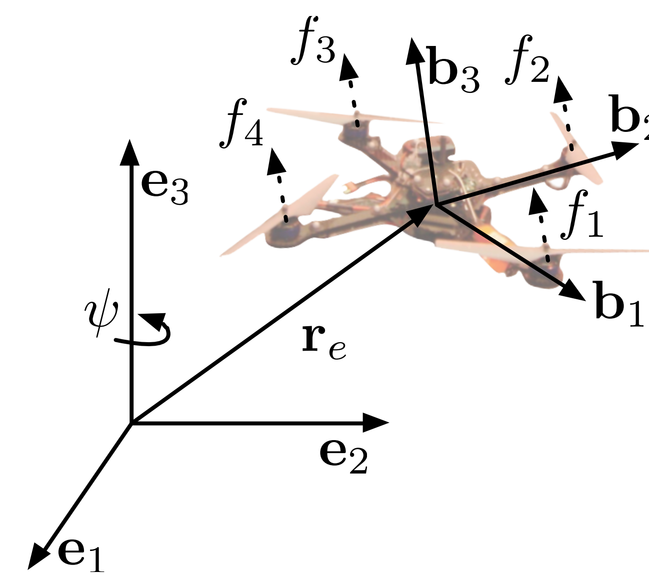
Body: $f = \sum_{i=1}^{4} f_i \xrightarrow[\text{along } \mathbf{b}_3]{} \mathbf{F}_b = \begin{bmatrix} 0 \\ 0 \\ f \end{bmatrix}$

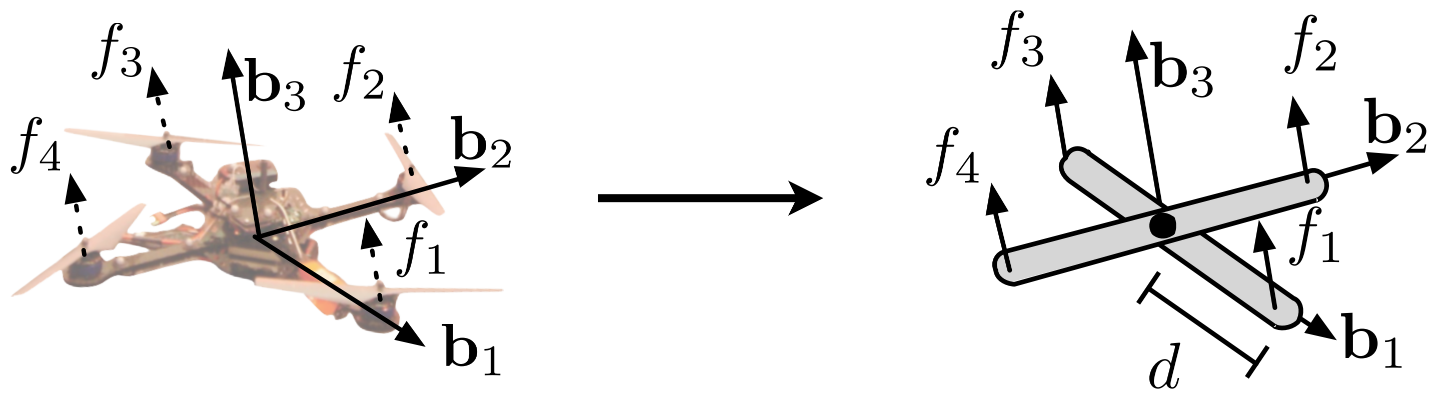
Inertial: $\mathbf{F}_e = R_{eb}\mathbf{F}_b - m\mathbf{g} \longleftarrow$ gravity

Force-Moment Generation

Newton-Euler equations:

$$\begin{bmatrix} \mathbf{F} \\ \tau \end{bmatrix} = \begin{bmatrix} m\mathbb{1}_3 & \mathbb{0}_3 \\ \mathbb{0}_3 & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \alpha \end{bmatrix} + \begin{bmatrix} \omega \times m\mathbf{v} \\ \omega \times \mathbf{I}_3\omega \end{bmatrix}$$



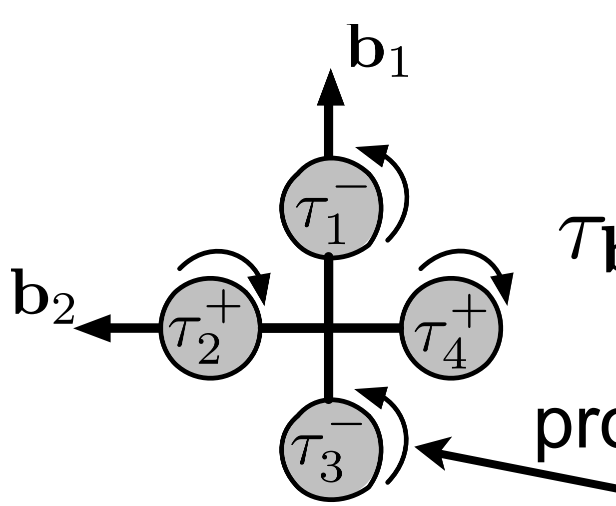


Total torque:

Recall: $\tau = \mathbf{r} \times \mathbf{F}$

$$\tau_{\mathbf{b}_1} = d\,(f_2 - f_4)$$

$$\tau_{\mathbf{b}_2} = d\,(f_3 - f_1)$$



propeller direction of rotation

induced moments

$$\tau_{\mathbf{b}_3} = -\tau_1 + \tau_2 - \tau_3 + \tau_4$$

THE ROBOTICS INSTITUTE

Equations of Motion (Summary)

Simplifying assumption: aligned center of mass and body frame

$$\begin{bmatrix} m\mathbb{1}_3 & \mathbb{0}_3 \\ \mathbb{0}_3 & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \alpha \end{bmatrix} + \begin{bmatrix} \cancel{\omega \times m\mathbf{v}} \\ \omega \times \mathbf{I}_3\omega \end{bmatrix} = \begin{bmatrix} \mathbf{F}_e \\ \tau \end{bmatrix} = \begin{bmatrix} R_{eb}\mathbf{F}_b - m\mathbf{g} \\ \left[\tau_{\mathbf{b}_1}, \tau_{\mathbf{b}_2}, \tau_{\mathbf{b}_3}\right]^{\mathrm{T}} \end{bmatrix}$$

$$\mathbf{F}_e = R_{eb}\mathbf{F}_b - m\mathbf{g}$$

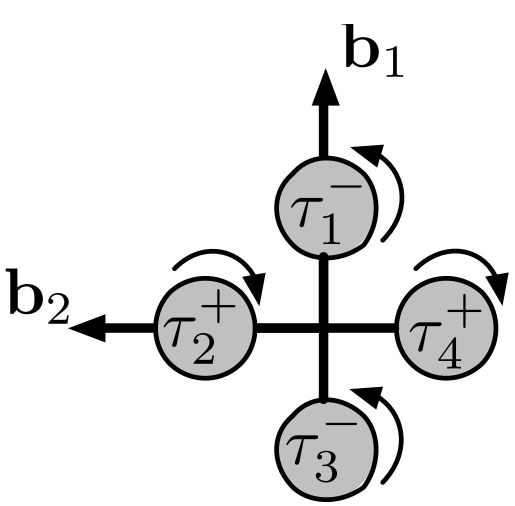
$$\mathbf{F}_b = \begin{bmatrix} 0 \\ 0 \\ f \end{bmatrix}$$

$$\begin{aligned} \tau_{\mathbf{b}_1} &= d\left(f_2 - f_4\right) \\ \tau_{\mathbf{b}_2} &= d\left(f_3 - f_1\right) \\ \tau_{\mathbf{b}_3} &= -\tau_1 + \tau_2 - \tau_3 + \tau_4 \end{aligned}$$

Prop model:

$$f_i = c_{\mathrm{T}}\bar{\omega}_i^2$$

$$\tau_i = \pm c_{\mathrm{Q}}\bar{\omega}_i^2$$



$$\begin{bmatrix} f \\ \tau_{\mathbf{b}_1} \\ \tau_{\mathbf{b}_2} \\ \tau_{\mathbf{b}_3} \end{bmatrix} = \begin{bmatrix} c_{\mathrm{T}} & c_{\mathrm{T}} & c_{\mathrm{T}} & c_{\mathrm{T}} \\ 0 & dc_{\mathrm{T}} & 0 & -dc_{\mathrm{T}} \\ -dc_{\mathrm{T}} & 0 & dc_{\mathrm{T}} & 0 \\ -c_{\mathrm{Q}} & c_{\mathrm{Q}} & -c_{\mathrm{Q}} & c_{\mathrm{Q}} \end{bmatrix} \begin{bmatrix} \bar{w}_1^2 \\ \bar{w}_2^2 \\ \bar{w}_3^2 \\ \bar{w}_4^2 \end{bmatrix}$$

Simplifying Model Assumptions

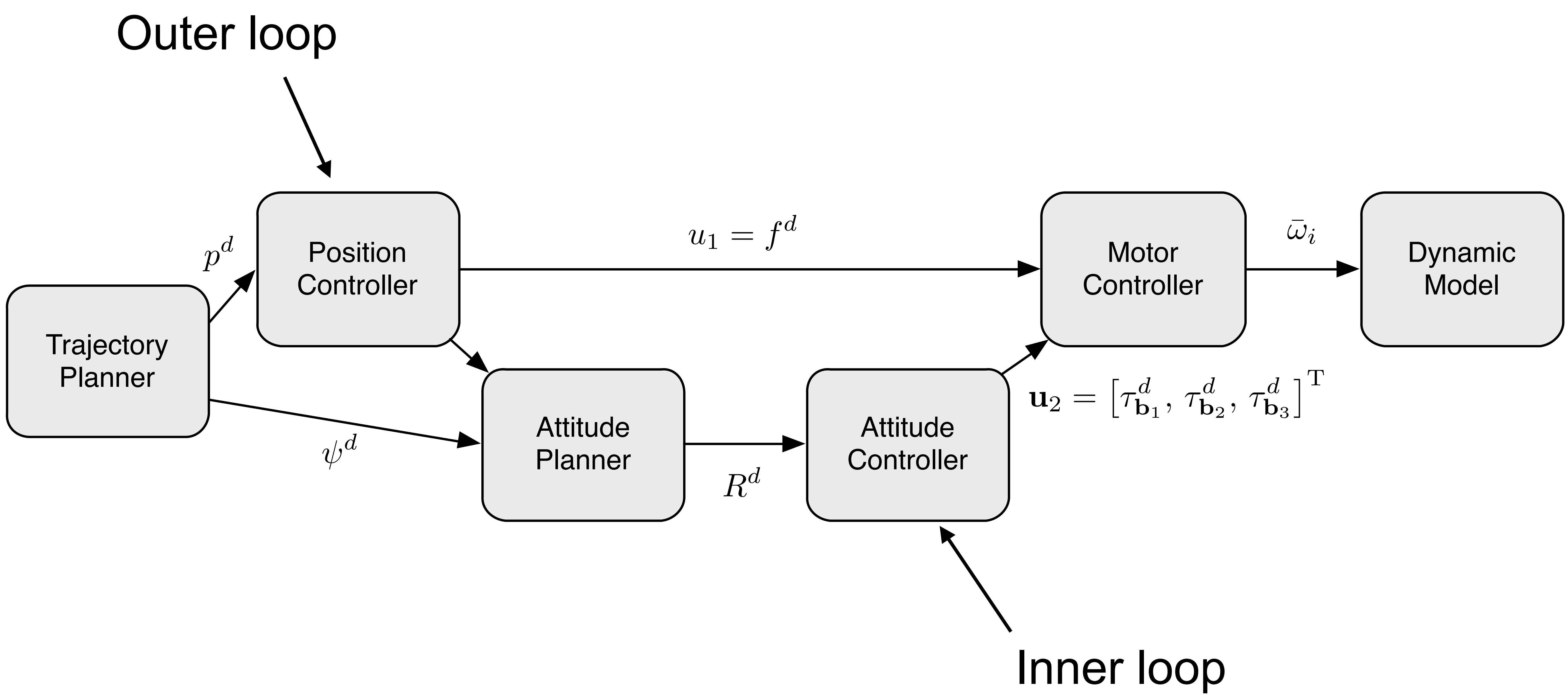
- Aligned center of mass and body frame
- Motor follows first-order model
- Ignore:
  - Power model (battery properties)
  - Rotor/blade flapping
  - Drag:
    - Induced: advancing blade generates more lift than retreating blade
    - Momentum: induced velocity of airflow through rotor
    - Parasitic: non-lifting (e.g., body) surfaces
    - Profile: transverse velocity of the rotor blade moving through air

For a more complete description of power and drag effects:

M. Bangura and R. Mahony, “Nonlinear Dynamic Modeling for High Performance Control of a Quadrotor,” presented at the Proc. of Australasian Conf. on Robot. and Autom., Wellington, New Zealand, 2012.

THE ROBOTICS INSTITUTE

Control System Diagram



R. Mahony, V. Kumar, and P. Corke. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. IEEE Robot. Autom. Mag., 19(3):20–32, Sept. 2012.

Proportional-Derivative Feedback Control

From 16-642 (or undergrad), define the linear time invariant (LTI) system:

$$\dot{x} = Ax + Bu$$

with the control law (regulator):

$$u = -Kx$$

The closed loop system is:

$$\dot{x} = (A - BK)x$$

Or with respect to a constant (or slowly varying) reference:

$$u = -Kx \longrightarrow u = -K(x - x^d) = -Ke_x$$

For a second order system:

$$u = -K_p e_x - K_d \dot{e}_x$$

Linearization of Nonlinear Model

Given a nonlinear system:

$$\dot{x} = f(x, u)$$
$$y = h(x)$$

We can approximate the system as locally linear (about an equilibrium point)

$$z = x - x_e$$

The system becomes:

$$\dot{z} = \dot{x} = f(z + z_e) = f_z(z)$$
$$y = h(z + x_e) = h_z(z)$$

THE ROBOTICS INSTITUTE

Linearization of Nonlinear Model

The linear approximation is:

$$\dot{z} = Az + Bu$$

$$A = \left.\frac{\partial f_z}{\partial z}\right|_{\substack{z=x_e\\u=u_e}} \qquad B = \left.\frac{\partial f_z}{\partial u}\right|_{\substack{z=x_e\\u=u_e}}$$

Assume identity feedback for now (so *y=h(x)* is already linear).

Linearize about hover:

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

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

$$\mathbf{F}_{b,0} = \left[0,\ 0,\ mg\right]^{\mathrm{T}}$$

Attitude Control

PD control law:

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

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

Rotation error metric:

nonlinear

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

hat operator:

$$\hat{\mathbf{a}} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

vee operator:

$$(\hat{\mathbf{a}})^{\vee} = \mathbf{a}$$

Attitude Control

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 $\longrightarrow$

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

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

THE ROBOTICS INSTITUTE

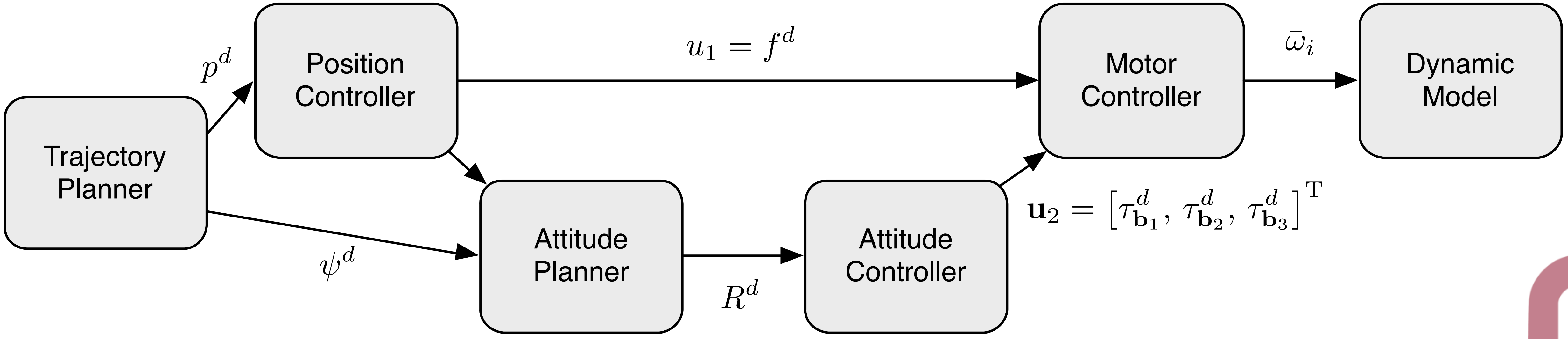
Attitude Control

PD control law:

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

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

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



THE ROBOTICS INSTITUTE

Position Control

Linearized model:

$$\ddot{x} = g\left(\Delta\theta\cos\psi_0 + \Delta\phi\sin\psi_0\right)$$

$$\ddot{y} = g\left(\Delta\theta\sin\psi_0 - \Delta\phi\cos\psi_0\right)$$

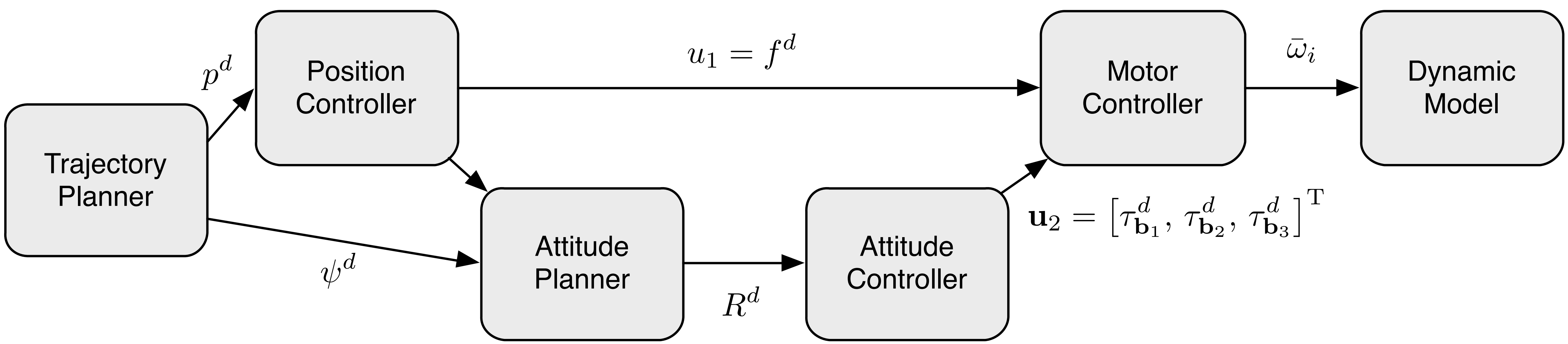
$$\ddot{z} = \frac{1}{m}u_1 - g$$

Mathematica derivation

Position Control

PD control law:

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



How do we pick the gains?

Proportional-Derivative Feedback Control

Recall for a second order system:

$$\ddot{x} + 2\gamma\omega_n\dot{x} + \omega_n^2 x = 0$$

Three response cases:

Under-damped:

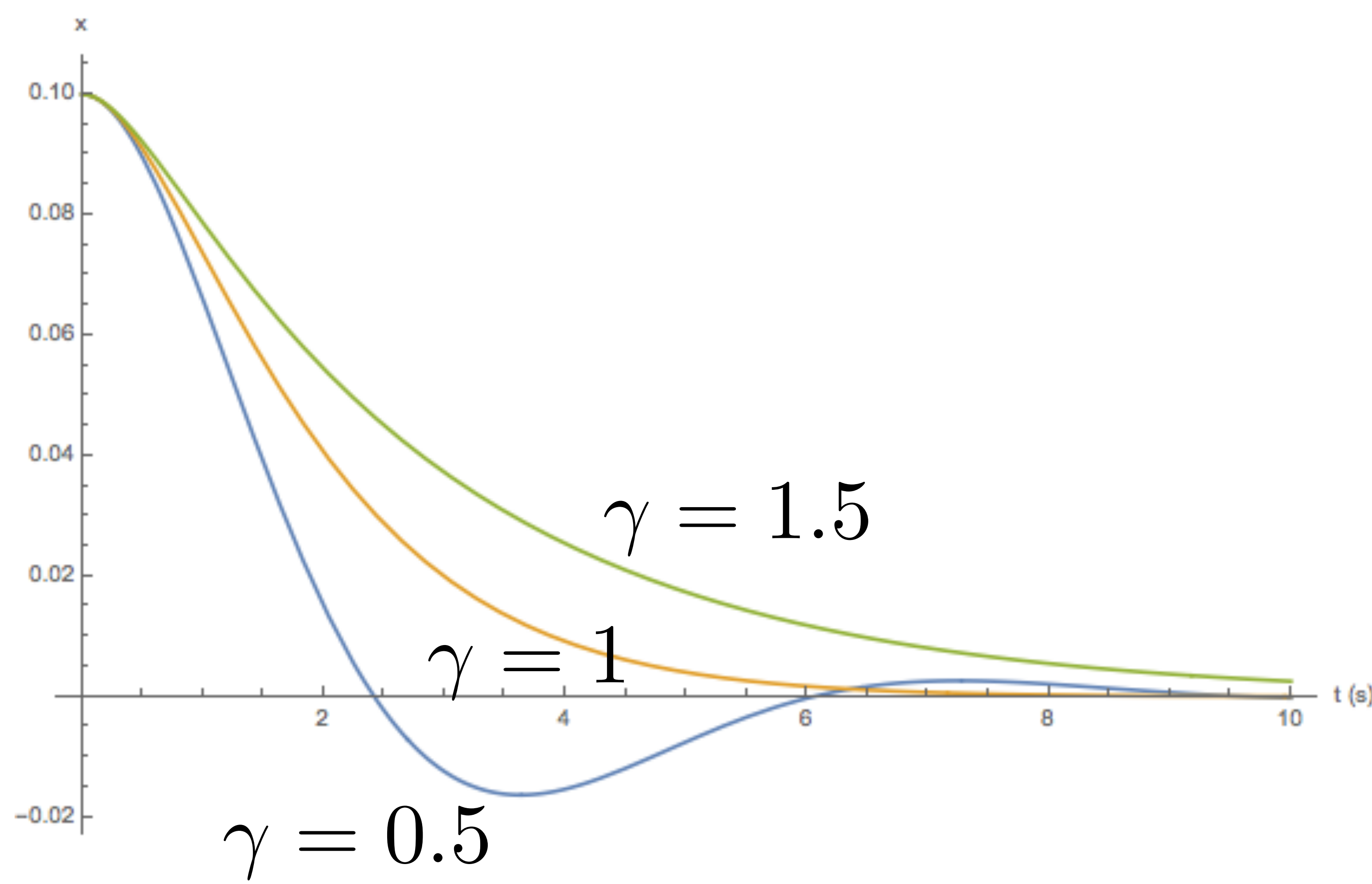
$$0 \leq \gamma < 1$$

Critically damped:

$$\gamma = 1$$

Over-damped:

$$\gamma > 1$$



Model error response as second order and design accordingly

THE ROBOTICS INSTITUTE

LQR (alternative gain selection approach)

Penalty for not getting to goal

$$J(\mathbf{x}, \mathbf{u}) = \int_0^\infty \left(\mathbf{x}^\mathrm{T} Q \mathbf{x} + \mathbf{u}^\mathrm{T} R \mathbf{u}\right) dt$$

Penalty for expending energy