Control System Design

Introduction to Control Theory

- Order of a system
- Linear Time Invariant (LTI) systems
  - Single integrator (kinematic)
  - Double integrator
  - Feedforward, feedback control
- Controller Design
  - Gain tuning
- Linear controller for a quadrotor
  - Planar quadrotor
  - Project 1 Phase 2 introduction

Control of a simple first-order system

Problem

State, input

$$x,\ u\ \in \mathbb{R}$$

Kinematic plant model

$$\dot{x} = u$$

Want x to follow 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

$$\dot{e} + K_p e = 0 \qquad K_P > 0$$

$$u(t) = \underbrace{\dot{x}^{des}(t)}_{\text{Feedforward}} + \underbrace{K_P e(t)}_{\text{Proportional}}$$

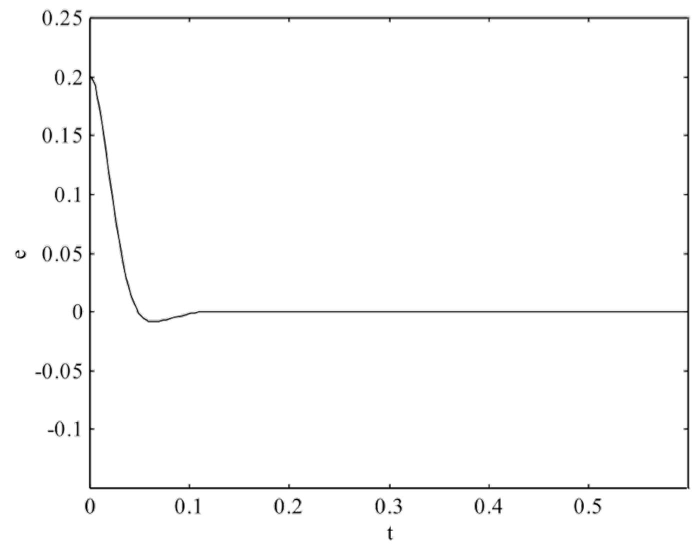
Control of a simple second-order system

Problem

State, input $x,\ u\ \in \mathbb{R}$

Kinematic plant model $\ddot{x} = u$

Want x to follow 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

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

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

Feedforward Derivative Proportional

Control for trajectory tracking in a simple second-order system

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 and the system converges slowly

PID control

In the presence of disturbances or modeling errors, it is often advantageous to use PID control

$$u(t) = \ddot{x}^{des}(t) + K_V \dot{e}(t) + K_P e(t) + K_I \underbrace{\int_0^t e(\tau) d\tau}_{\text{Integral}}$$

PID control generates a third-order closed-loop system

Integral control makes the steady-state error go to zero

Penn Engineering

Model-based control for trajectory tracking

Disadvantages of PID or PD control schemes

$$m\ddot{x}(t) + b\dot{x}(t) + kx(t) = f(t)$$

- performance will depend on the model
- need to tune gains to maximize performance

Model based control law

$$f(t) = \overset{\text{model based}}{m}\underbrace{\left(\ddot{x}_d(t) + k_p e(t) + k_v \dot{e}(t)\right)}_{\text{feedforward + PD feedback}} + \underbrace{b\dot{x}(t) + kx(t)}_{\text{model based}}$$

Two parts of a model based scheme

– model based part

- cancel the dynamics of the system
- specific to the model

– servo based part

- use PID or PD with feedforward to drive errors to zero
- <u>independent</u> of the model of the system

Model-based control for trajectory tracking

Model

$$m\ddot{x}(t)+b\dot{x}(t)+kx(t)=f(t)$$

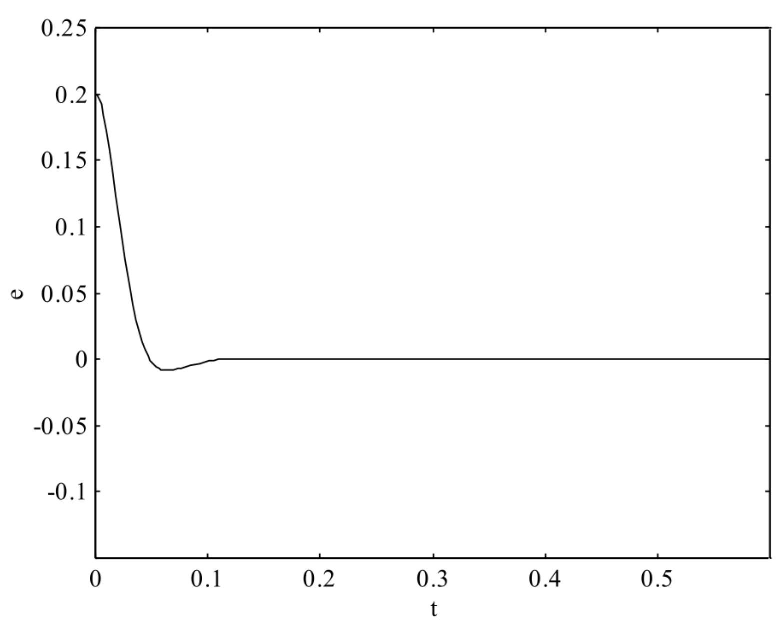
Model based control law

model based

$$f(t)=m\underbrace{\left(\ddot{x}_d(t)+k_p e(t)+k_v \dot{e}(t)\right)}_{\text{servo}}+b\dot{x}(t)+kx(t)$$

Performance

$$\ddot{e}+k_v\dot{e}+k_p e=0$$



Model-based control for trajectory tracking

- Advantage
  - decomposes the control law into
    - model-dependent part (depends on the knowledge of the model)
    - model-independent part (servo control, gains are independent of the model)
- Disadvantage

Model based control law (based on estimates of model parameters)

$$f(t) = \hat{m}\left(\ddot{x}_d(t) + k_p e(t) + k_v \dot{e}(t)\right) + \hat{b}\dot{x}(t) + \hat{k}x(t)$$

estimates

Ideal performance

$$\ddot{e} + k_v \dot{e} + k_p e = 0$$

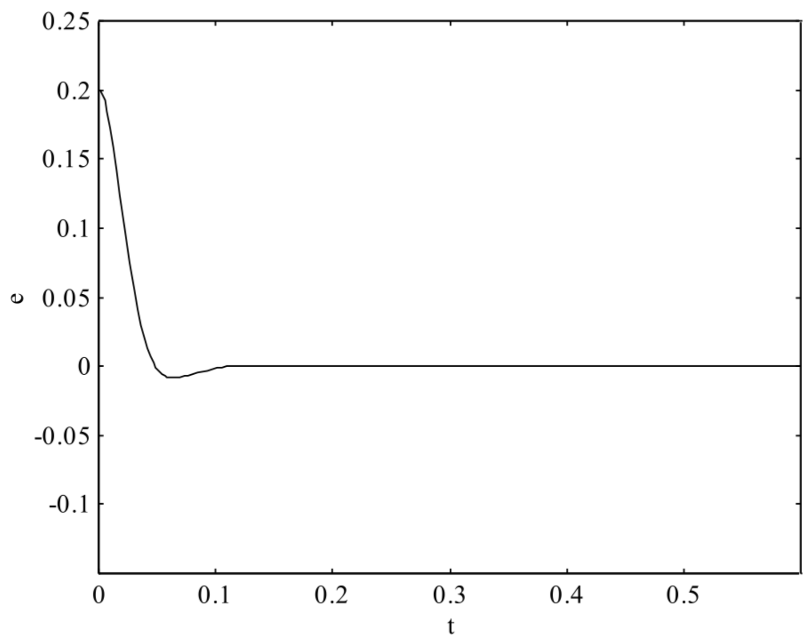
Actual performance

$$\ddot{e} + k_v \dot{e} + k_p e = \left(\tfrac{m}{\hat{m}} - 1\right)\ddot{x} + \tfrac{b-\hat{b}}{\hat{m}}\dot{x} + \tfrac{k-\hat{k}}{\hat{m}}x$$

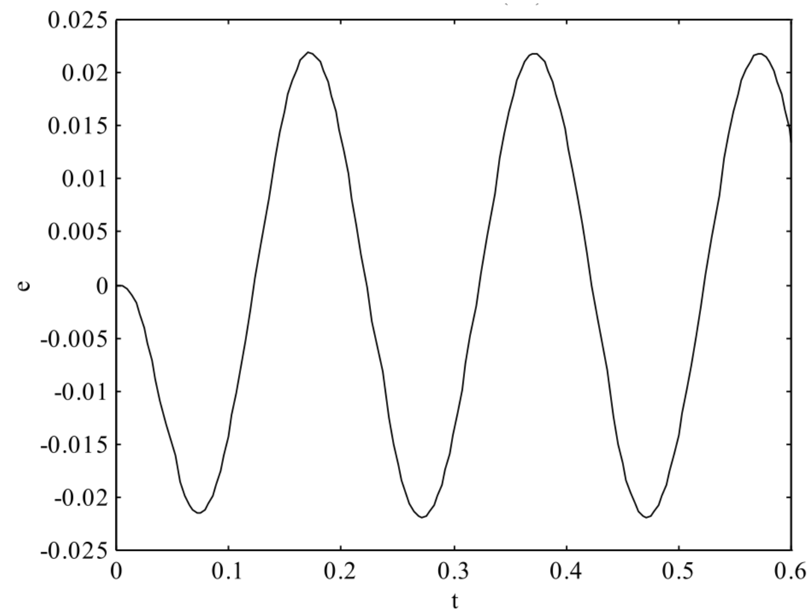
1. Error term will not go exponentially to zero
2. Right hand side is a forcing function driving the error away from zero

Model-based control for trajectory tracking

$$\ddot{e} + k_v \dot{e} + k_p e = \boxed{\left(\frac{m}{\hat{m}} - 1\right)\ddot{x} + \frac{b-\hat{b}}{\hat{m}}\dot{x} + \frac{k-\hat{k}}{\hat{m}}x} \quad f_p(t)$$



Perfect model



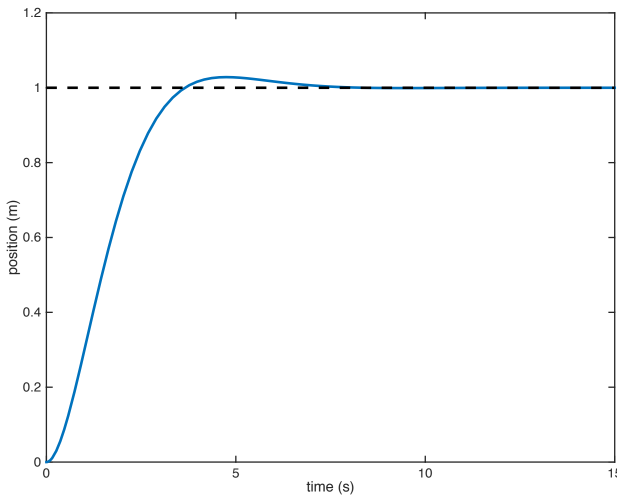
Imperfect model, 10% errors in parameters

Not all is lost however

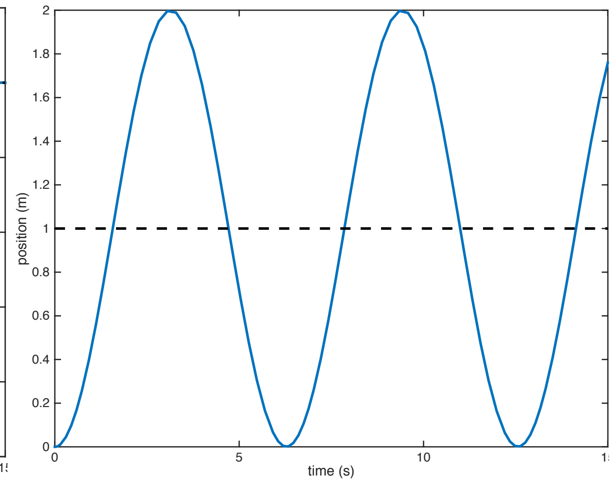
- Treat f_p as a perturbation or a disturbance force
- If $\max_t f_p(t) < M$ we can prove that the error $e(t)$ is also bounded

Gain Tuning

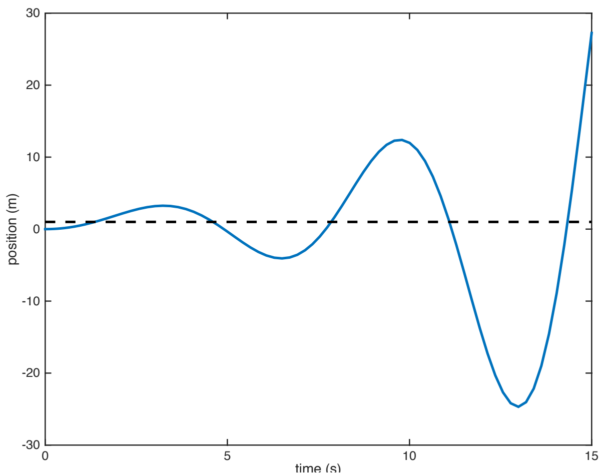
Manual Tuning



Stable

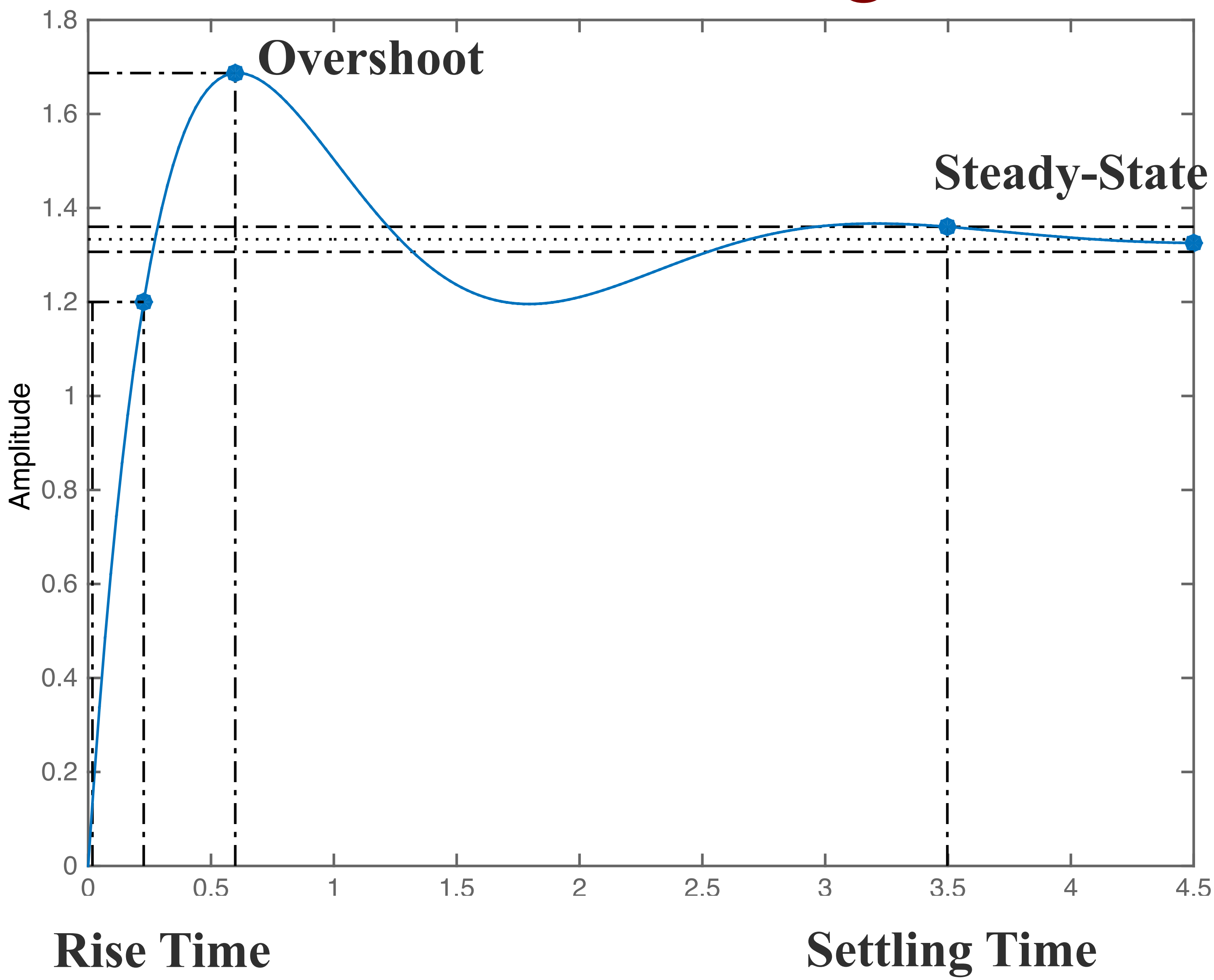


Marginally Stable



Unstable

Manual Tuning



Manual Tuning

Parameter Increased	K_p	K_d	K_i
Rise Time	Decrease	-	Decrease
Overshoot	Increase	Decrease	Increase
Settling Time	-	Decrease	Increase
Steady-State Error	Decrease	n/a	Eliminate

Ziegler-Nichols Method

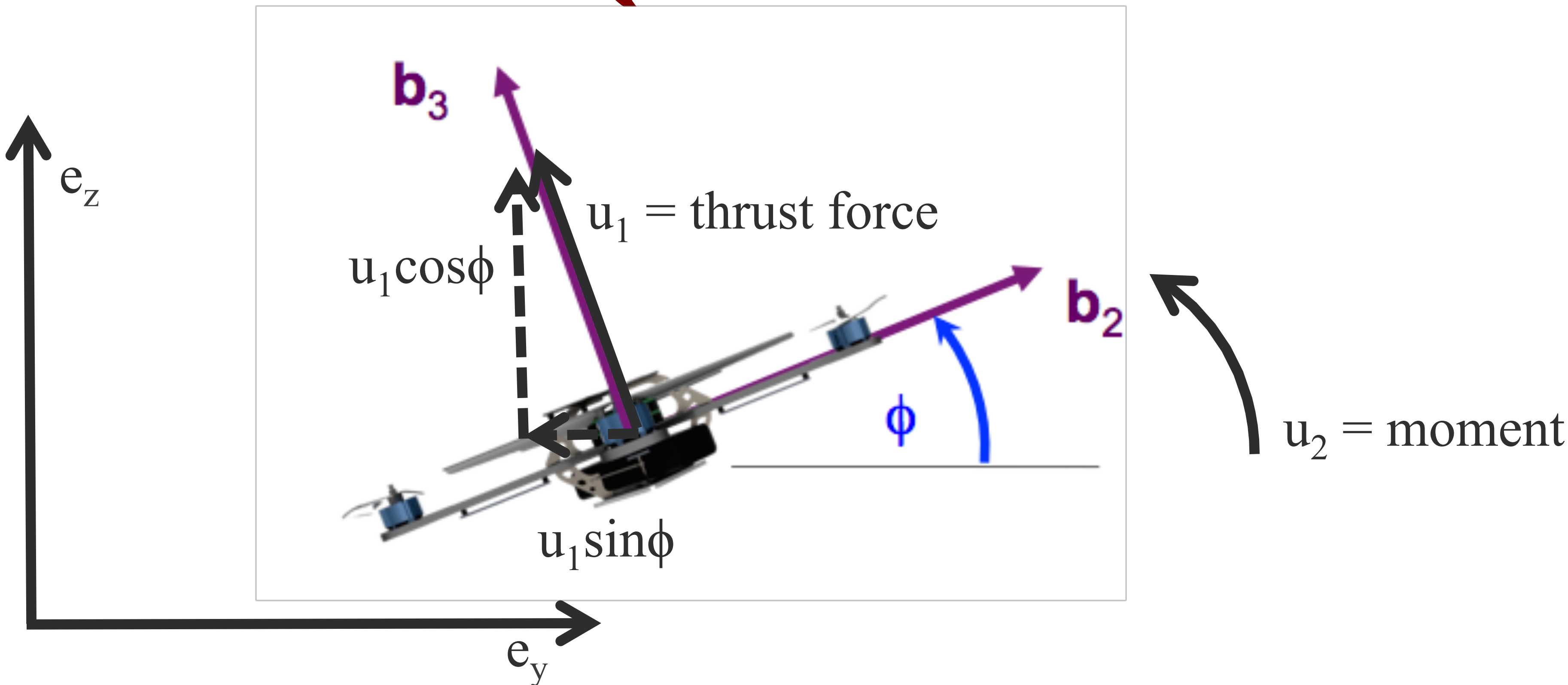
Heuristic for tuning gains

1. Set $K_i = K_d = 0$
2. Increase K_p until ultimate gain, K_u, when output starts to oscillate
3. Find the oscillation period T_u at K_u
4. Set gains according to:

Controller	K_p	K_d	K_i
P	$0.50K_u$	-	-
PD	$0.80K_u$	$K_pT_u/8$	-
PID	$0.60K_u$	$2K_p/T_u$	$K_pT_u/8$

Application to Quadrotors

Planar Quadrotor Model



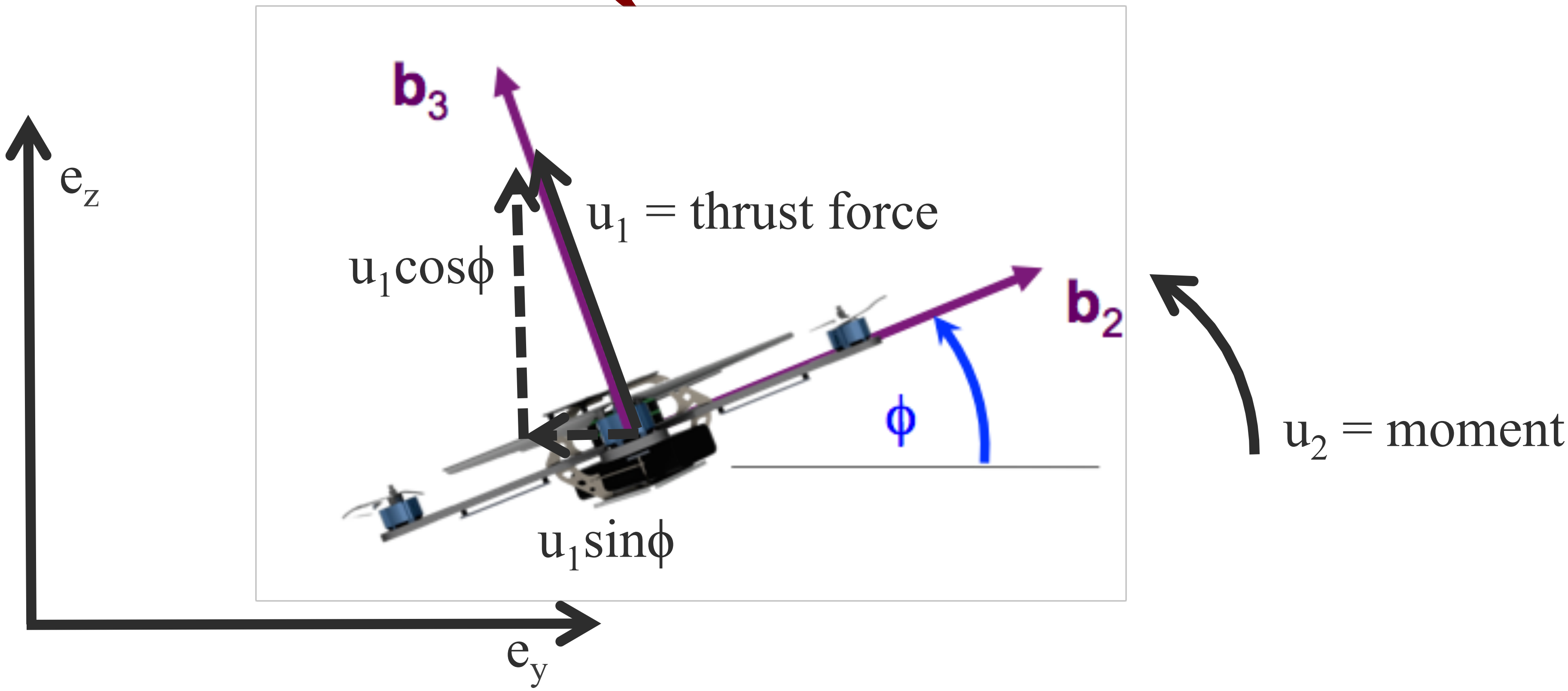
$$\mathbf{x} = [y \quad z \quad \phi \quad \dot{y} \quad \dot{z} \quad \dot{\phi}]^T$$

$$\sum \mathbf{F}_y = -u_1 \sin(\phi) = m\ddot{y}$$

$$\sum \mathbf{F}_z = -mg + u_1 \cos(\phi) = m\ddot{z}$$

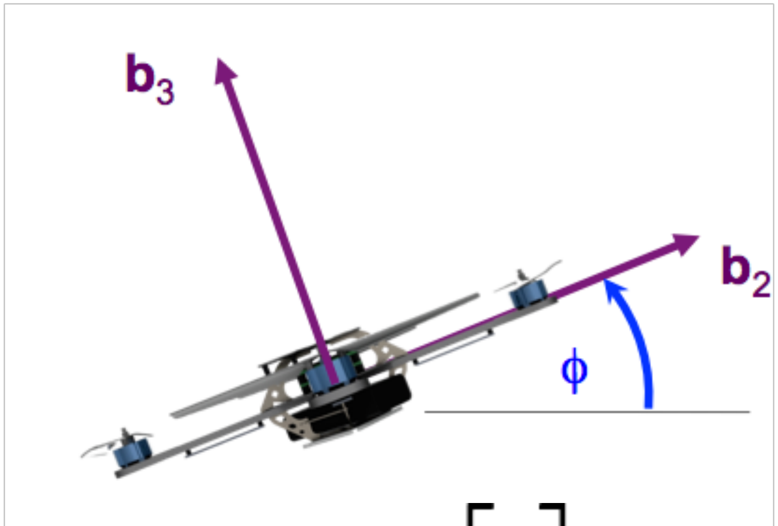
$$\mathbf{M} = u_2 = I_{xx}\ddot{\phi}$$

Planar Quadrotor Model



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

Planar Quadrotor Model



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

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

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

Penn Engineering

Linearized Dynamic Model

Nonlinear dynamics

$$\ddot{y} = -\frac{u_1}{m}\sin(\phi)$$
$$\ddot{z} = -g + \frac{u_1}{m}\cos(\phi)$$
$$\ddot{\phi} = \frac{u_2}{I_{xx}}$$

Equilibrium hover configuration

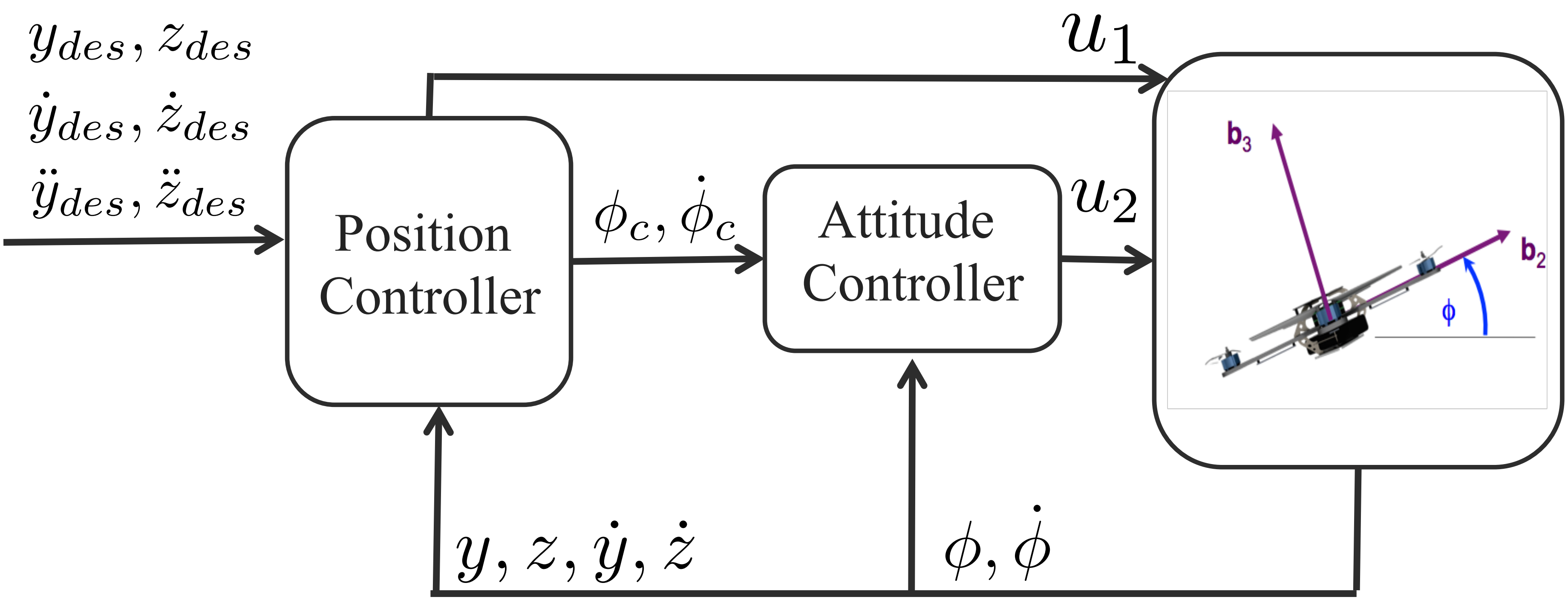
$$y_0, z_0, \phi_0 = 0, u_{1,0} = mg, u_{2,0} = 0,$$

Linearized dynamics

$$\ddot{y} = -g\phi$$
$$\ddot{z} = -g + \frac{u_1}{m}$$
$$\ddot{\phi} = \frac{u_2}{I_{xx}}$$

Penn Engineering

Nested Control Structure



Control Equations

Lateral dynamics

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

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

Attitude control

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

Position control - determining $\phi_c, \dot{\phi}_c$

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

$$\dot{\phi}_c = 0$$

Control Equations

Vertical dynamics

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

Z-position control

$$u_1 = m(g + \ddot{z}_c)$$

Control Equations

Determining $\ddot{y}_c, \ddot{z}_c$

$$\mathbf{x} = [y \quad z]^T$$

$$\ddot{\mathbf{e}} + k_{d,x}\dot{\mathbf{e}}_{des} + k_{p,x}\mathbf{e} = 0$$

Control Equations

Determining $\ddot{y}_c, \ddot{z}_c$

$$\mathbf{x} = [y \quad z]^T$$

$$(\ddot{\mathbf{x}}_{des} - \ddot{\mathbf{x}}_c) + k_{d,x}(\dot{\mathbf{x}}_{des} - \dot{\mathbf{x}}) + k_{p,x}(\mathbf{x}_{des} - \mathbf{x}) = 0$$

Specified by trajectory

Actual states

Note:

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

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

Model-based Feedforward Feedback

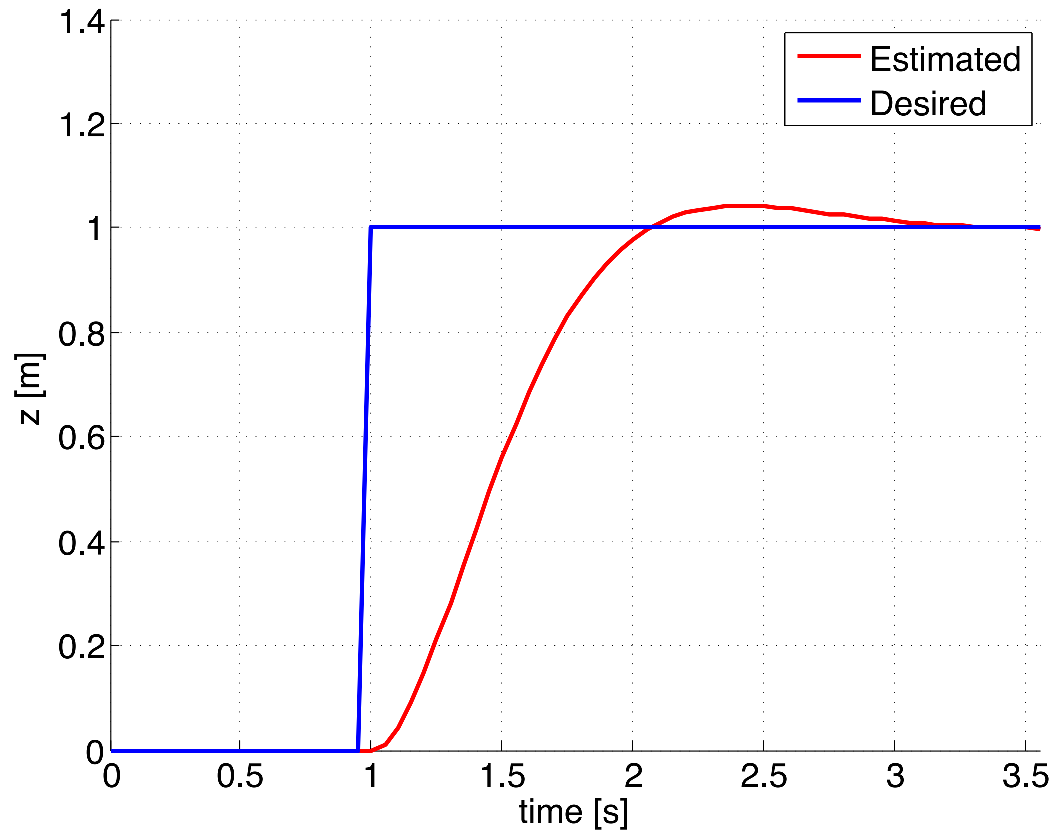
Control Equations

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

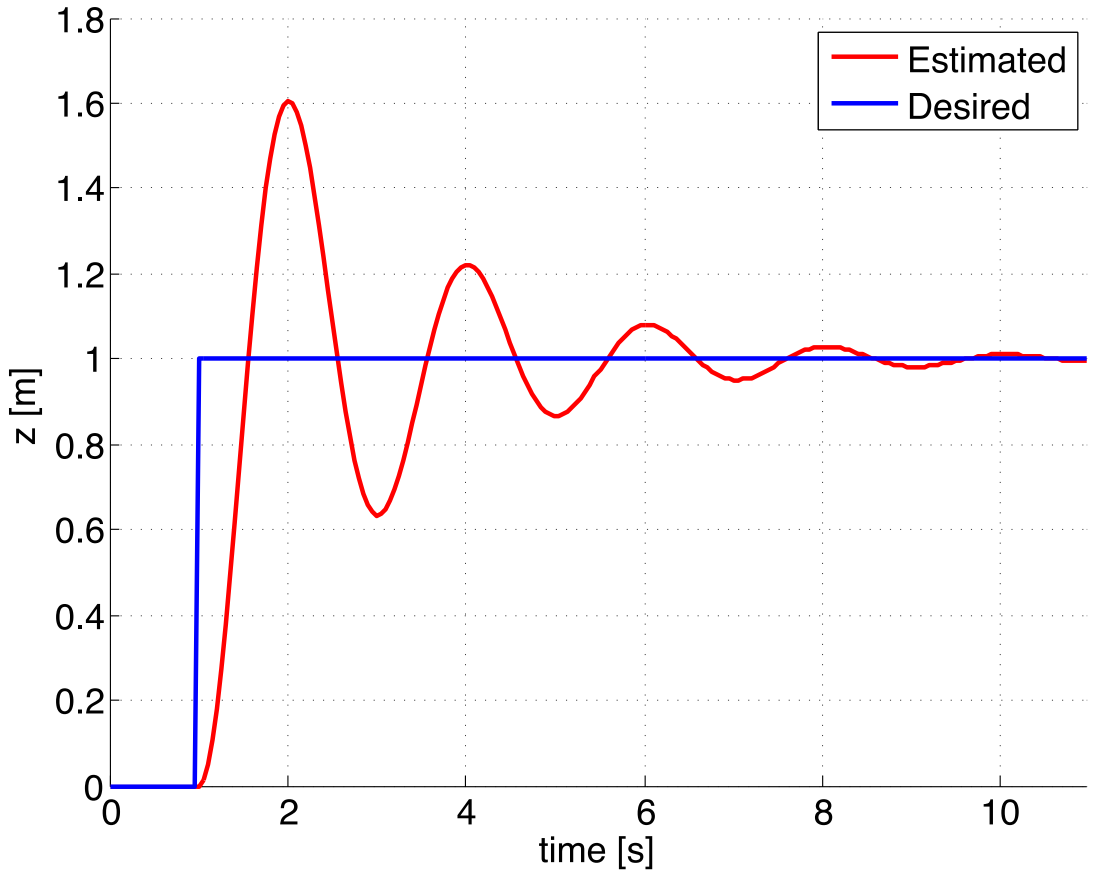
$$u_2 = k_{p,\phi}(\phi_c - \phi) + k_{d,\phi}(\dot{\phi}_c - \dot{\phi})$$

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

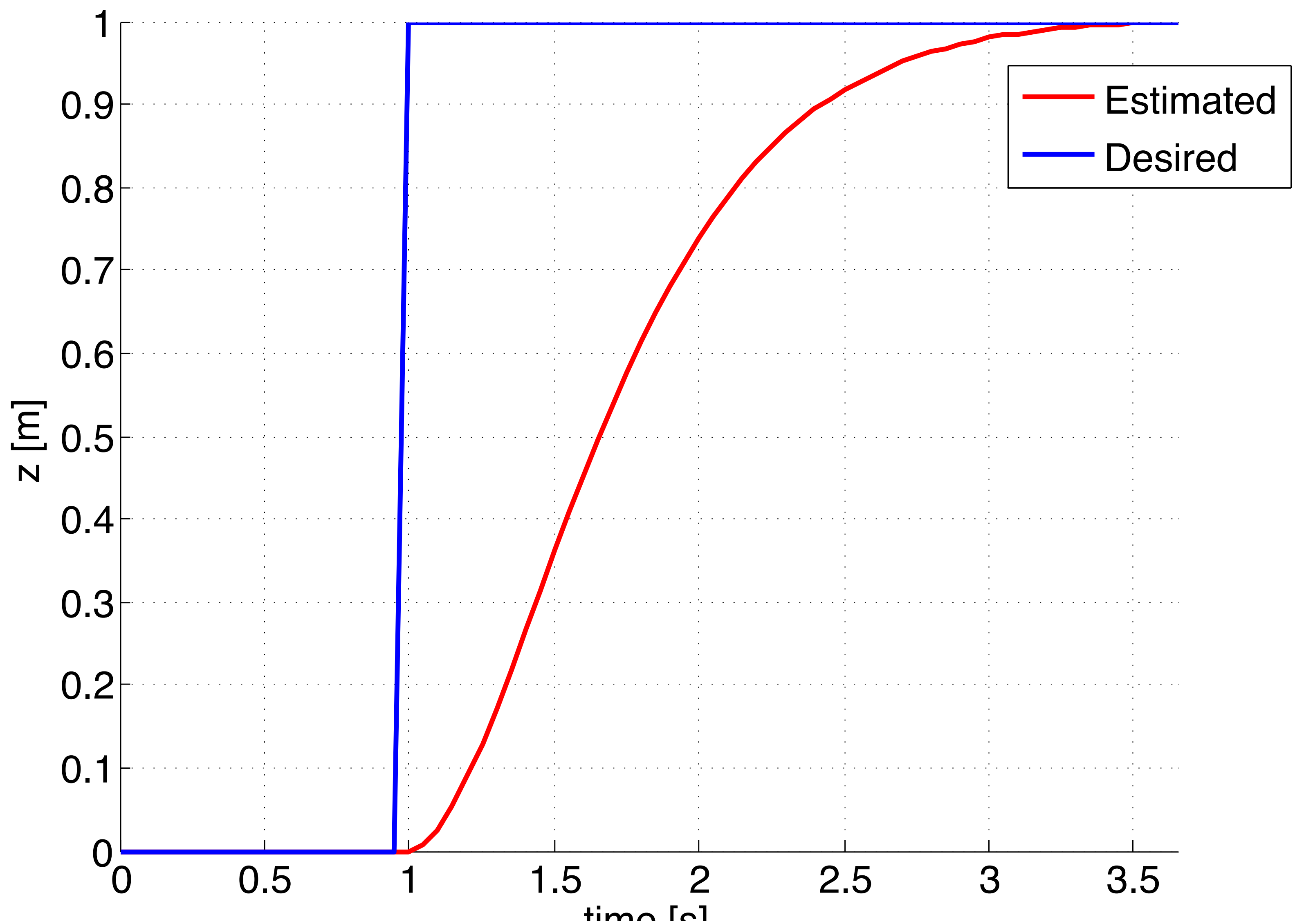
PD Controller



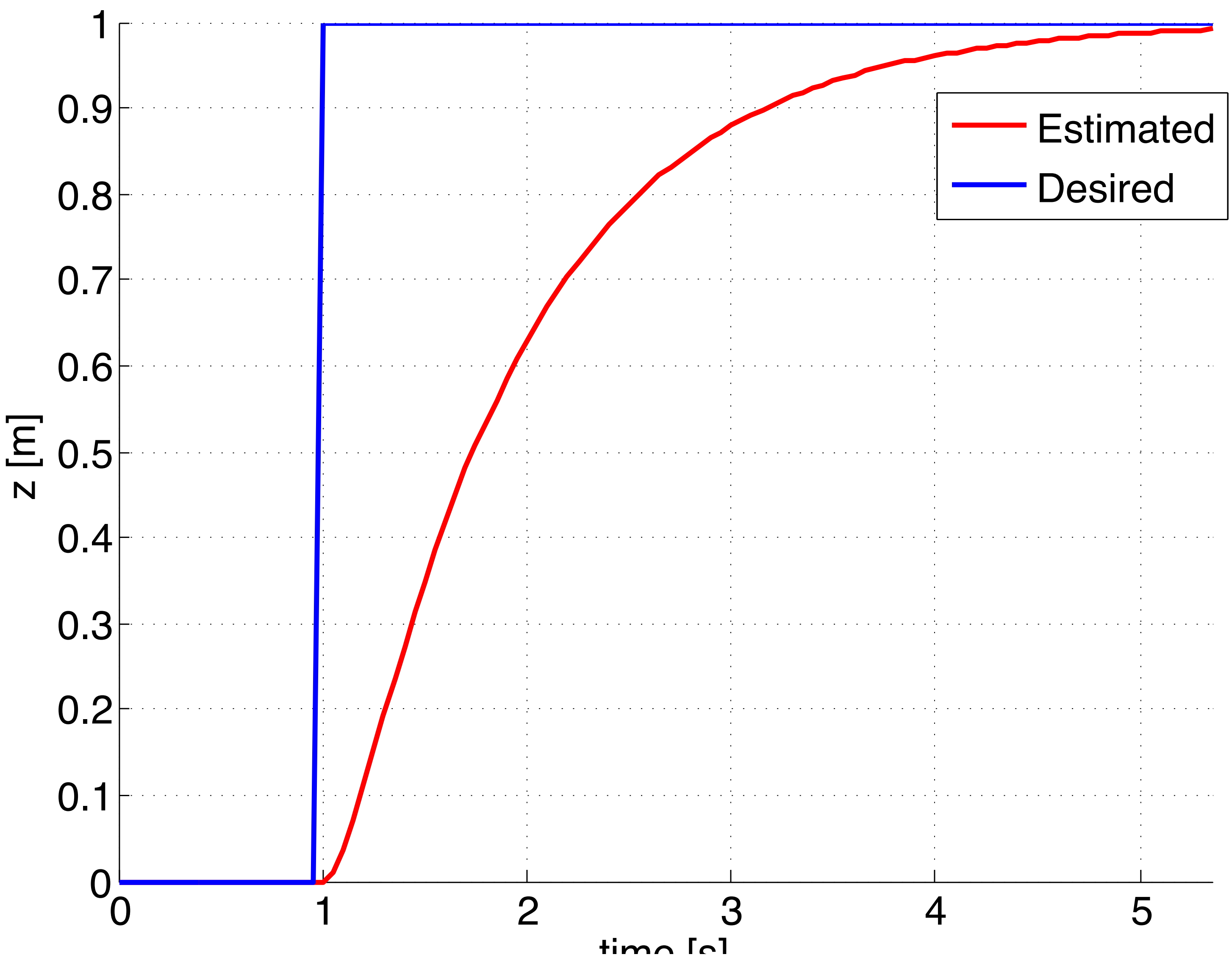
High K_p



Low K_p (soft response)

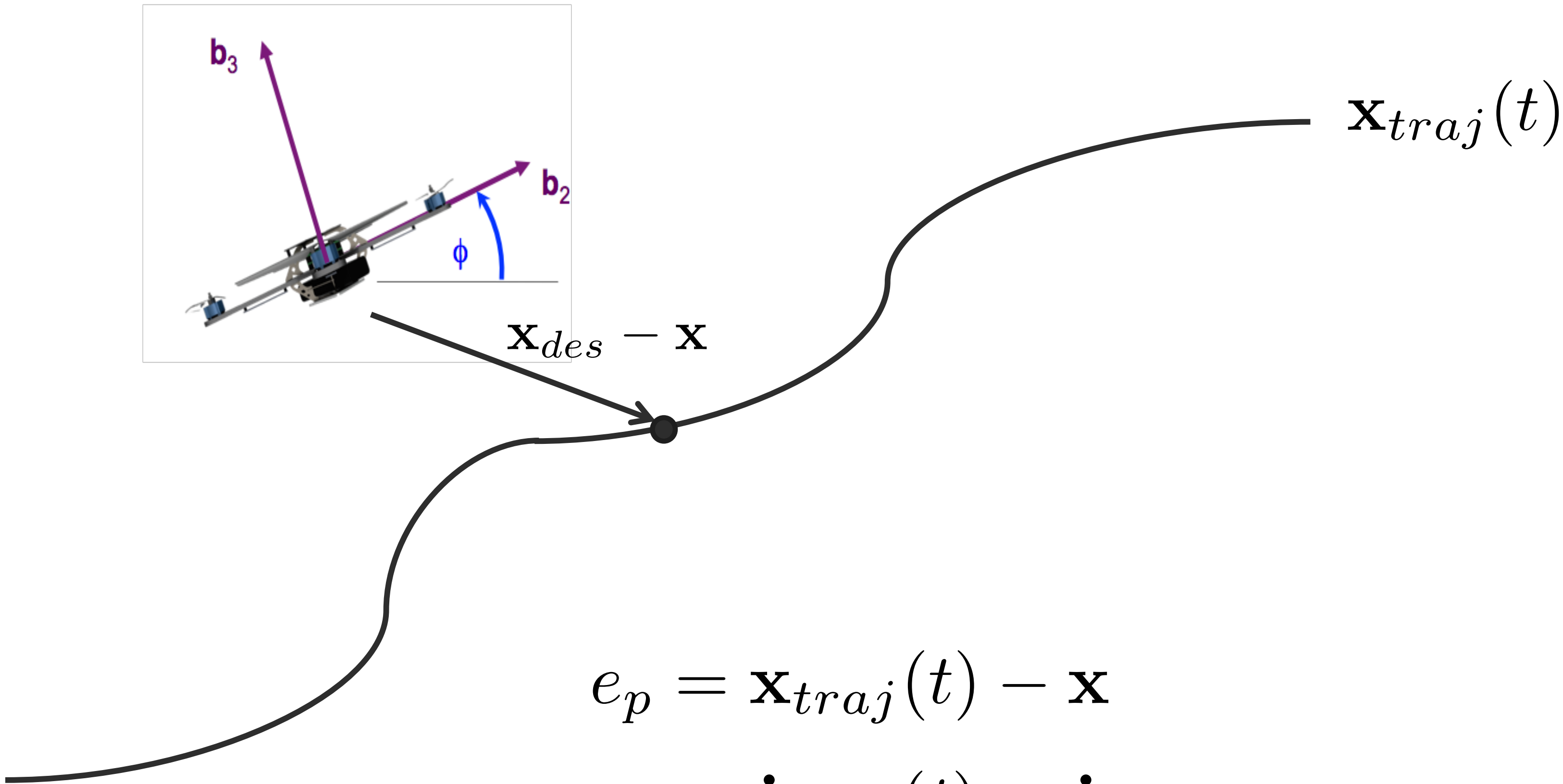


High K_d (overdamped)



Trajectory Tracking

Given $\mathbf{x}_{traj}(t), \dot{\mathbf{x}}_{traj}(t), \ddot{\mathbf{x}}_{traj}(t)$



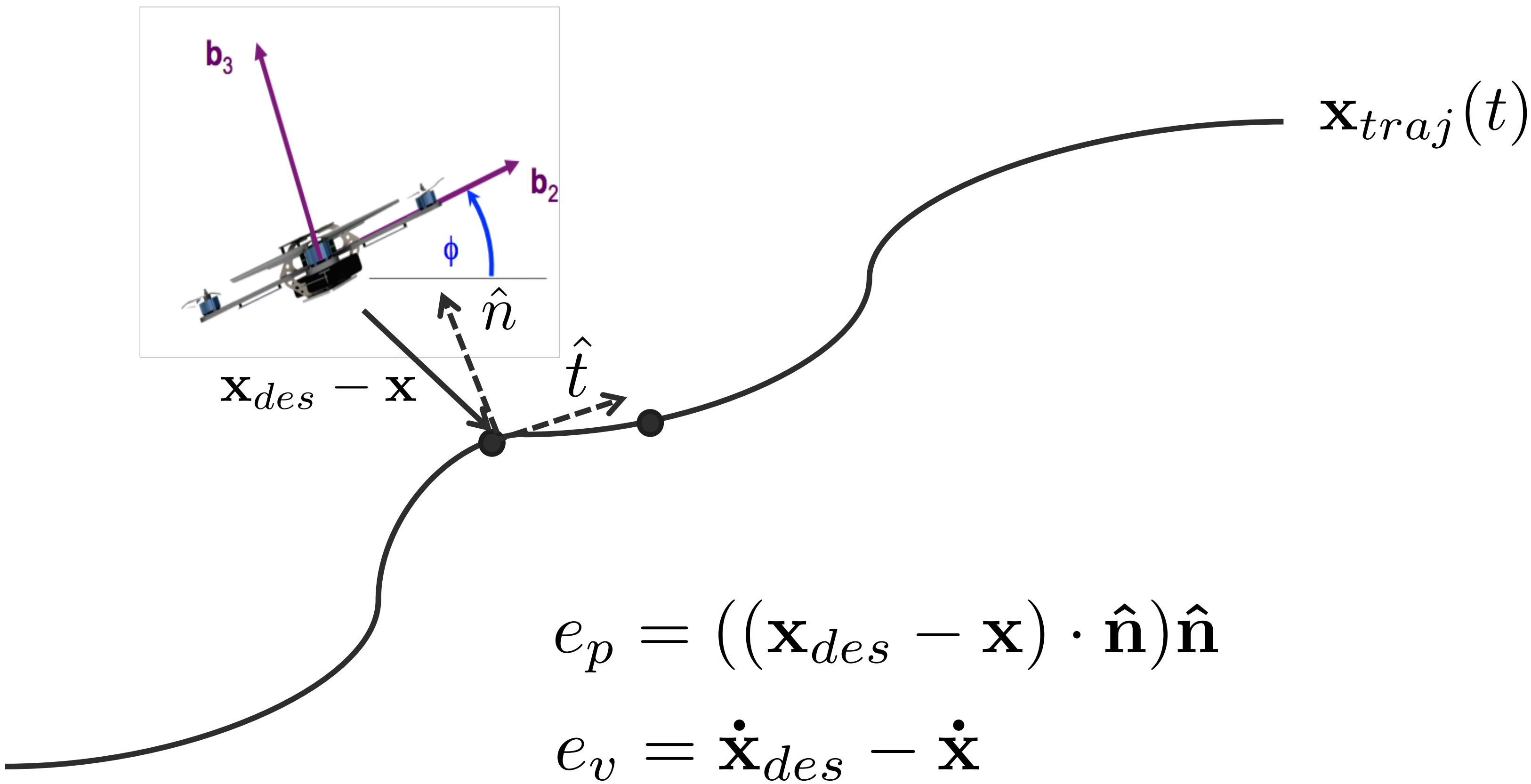
$$e_p = \mathbf{x}_{traj}(t) - \mathbf{x}$$

$$e_v = \dot{\mathbf{x}}_{traj}(t) - \dot{\mathbf{x}}$$

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

Trajectory Tracking

Given $\mathbf{x}_{traj}(t), \dot{\mathbf{x}}_{traj}(t), \ddot{\mathbf{x}}_{traj}(t)$



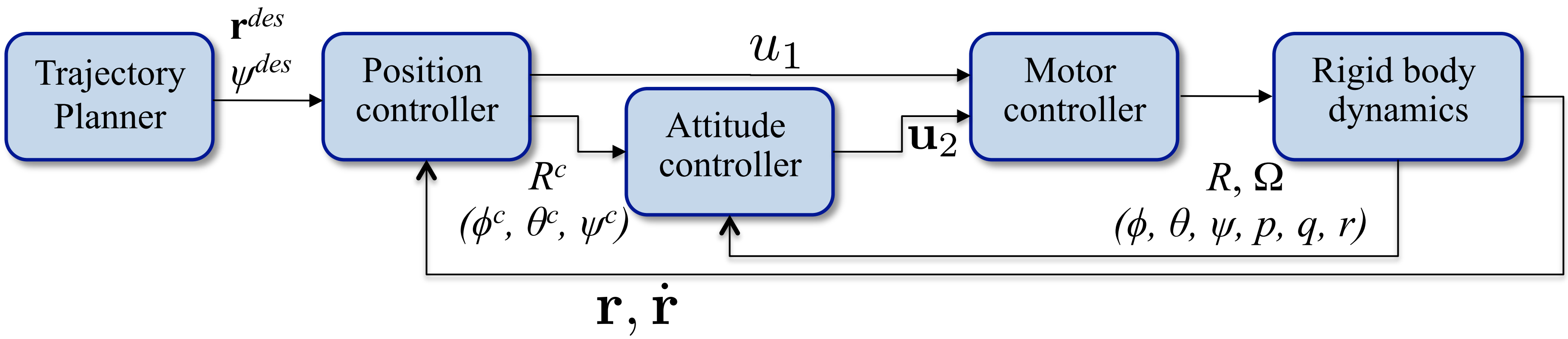
$$e_p = ((\mathbf{x}_{des} - \mathbf{x}) \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}$$

$$e_v = \dot{\mathbf{x}}_{des} - \dot{\mathbf{x}}$$

$$(\ddot{\mathbf{x}}_{des} - \ddot{\mathbf{x}}_c) + k_{d,x} e_v + k_{p,x} e_p = 0$$

Penn Engineering

3-D Quadrotor

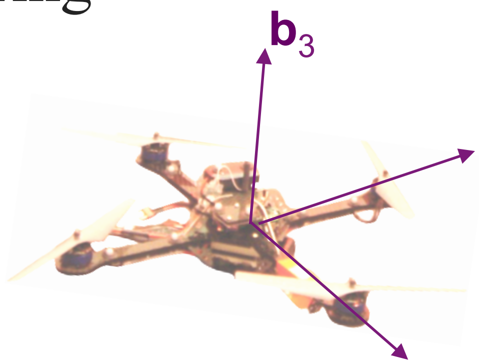


$$m\ddot{\mathbf{r}} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ \underbrace{\boxed{F_1 + F_2 + F_3 + F_4}}_{u_1} \end{bmatrix}$$

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

Penn Engineering

Control for Hovering



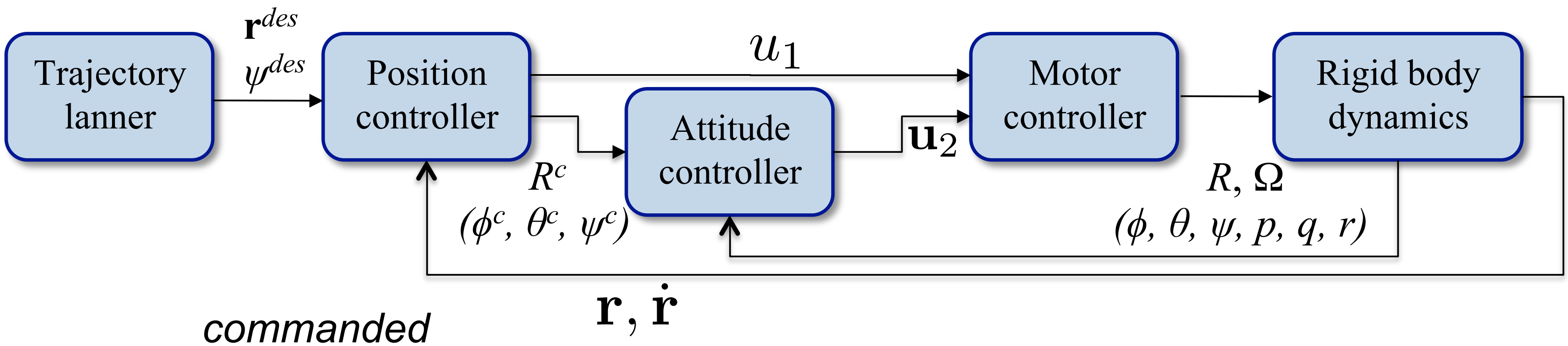
$$m\ddot{\mathbf{r}} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ \underbrace{\boxed{F_1 + F_2 + F_3 + F_4}}_{u_1} \end{bmatrix}$$

Linearization

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

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

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



commanded

actual (feedback)

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

specified

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

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

$$\theta_c = \frac{1}{g}(\ddot{r}_{1,c}\cos\psi + \ddot{r}_{2,c}\sin\psi)$$

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