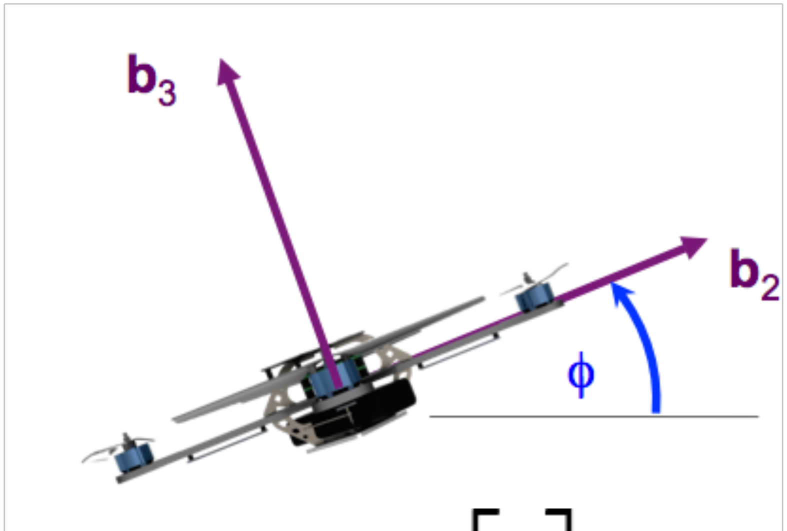
Planar Quadrotor

Planar Quadrotor Model



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

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

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

Linearized Dynamic Model

Equations of motion

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

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

Dynamics are nonlinear

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

Equilibrium hover configuration

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

Linearized dynamics

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

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

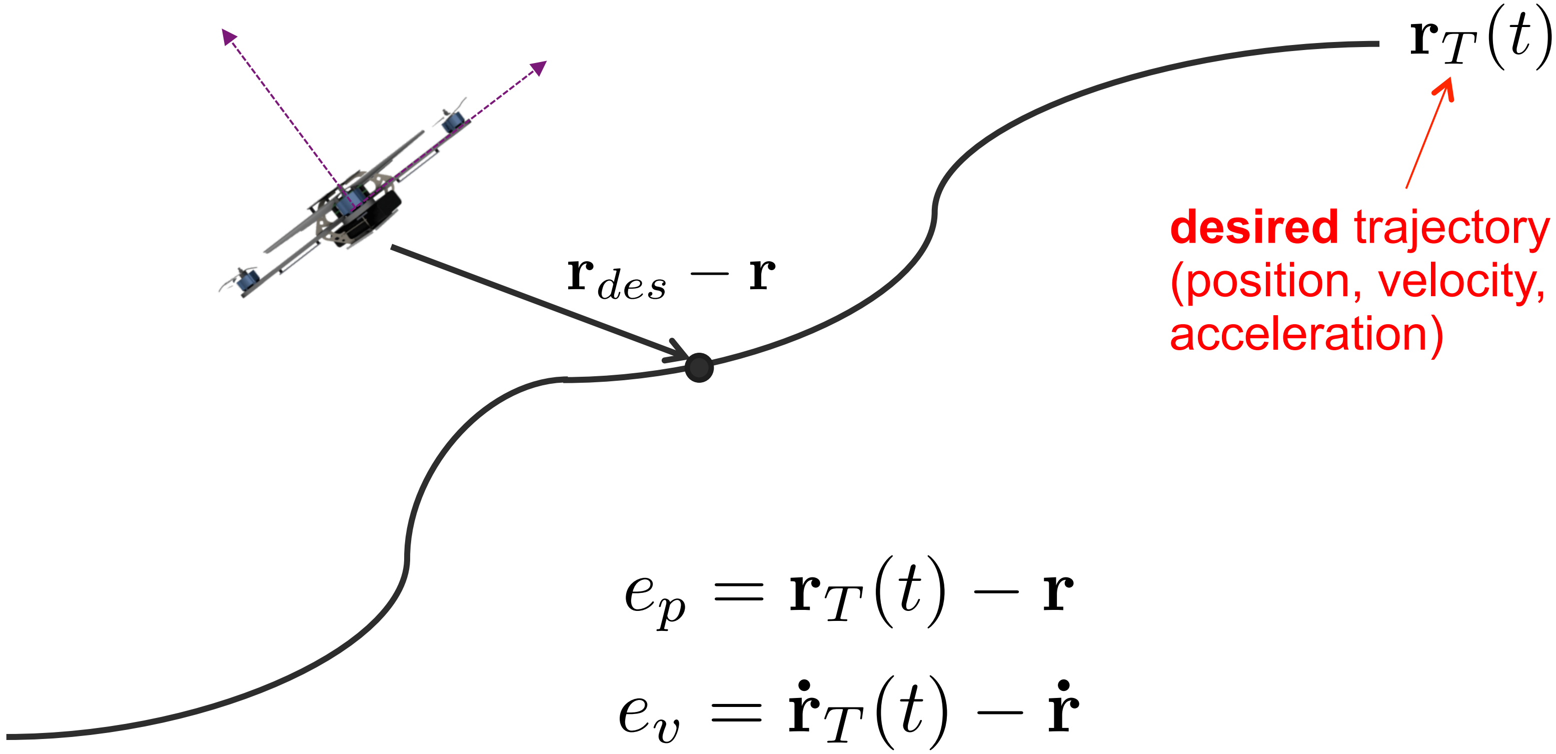
$$\ddot{\phi} = \frac{u_2}{I_{xx}}$$

Penn Engineering

Trajectory Tracking

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

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



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

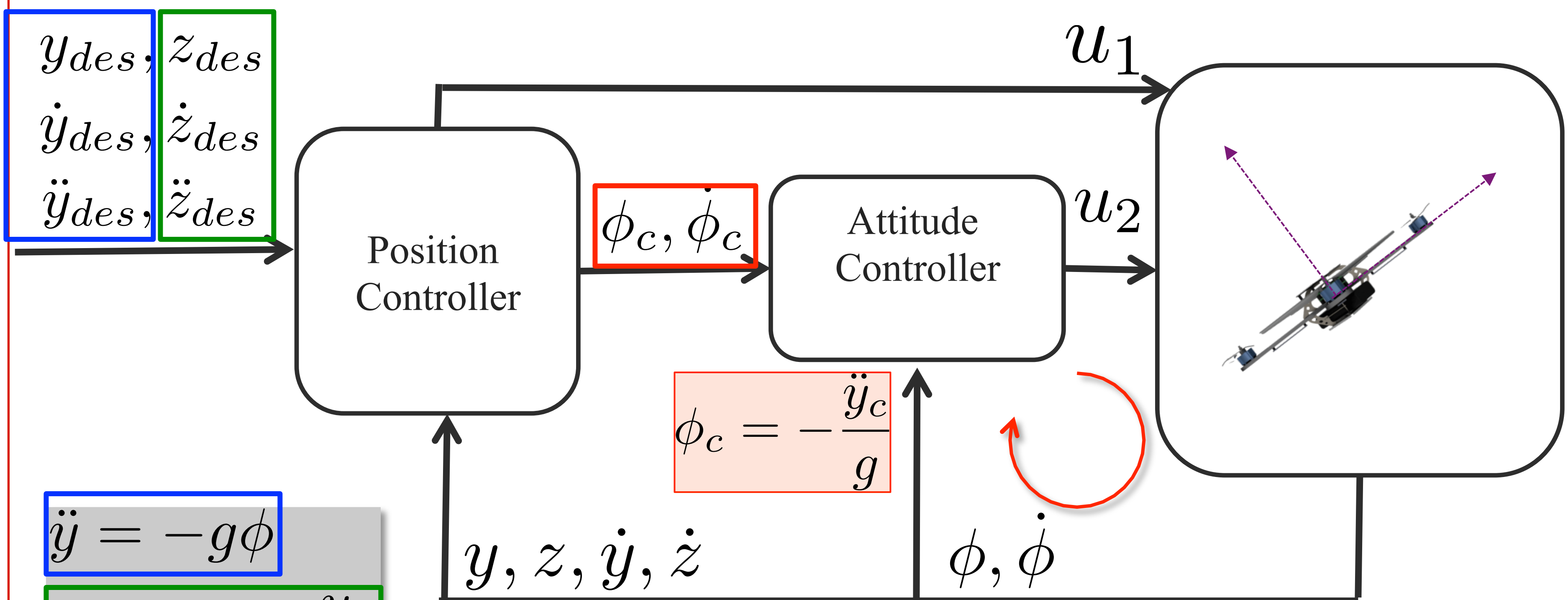
$$e_v = \dot{\mathbf{r}}_T(t) - \dot{\mathbf{r}}$$

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

Nested Control Structure

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



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

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

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

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

Penn Engineering

Control Equations

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

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

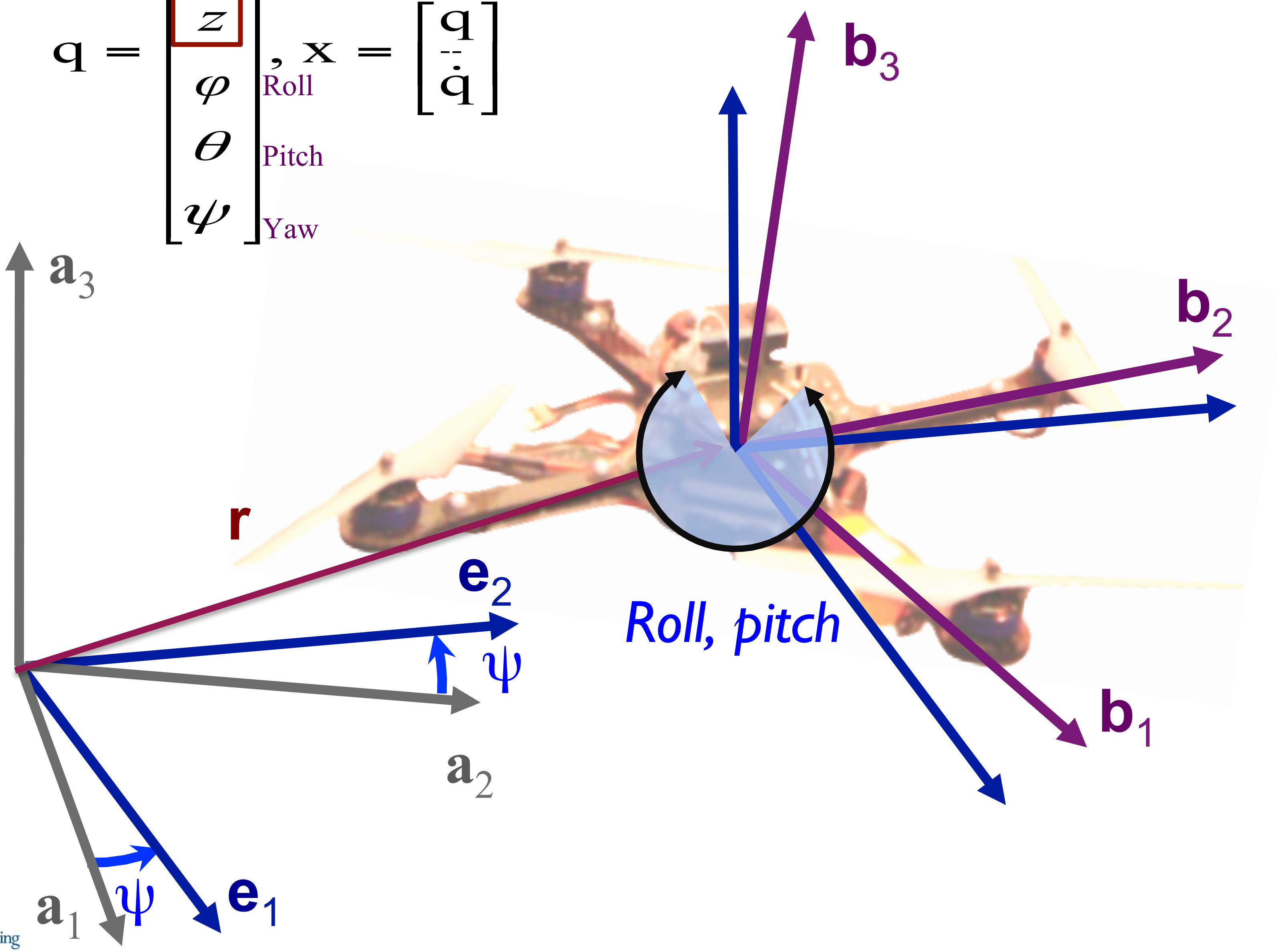
$$\phi_c = -\frac{1}{g}(\ddot{y}_{des} + k_{d,y}(\dot{y}_{des} - \dot{y}) + k_{p,y}(y_{des} - y))$$

3-D Quadrotor

$$\mathbf{q} = \begin{bmatrix} \boxed{\begin{matrix} x \\ y \\ z \end{matrix}}^{\,\mathbf{r}} \\ \varphi \\ \theta \\ \psi \end{bmatrix} \begin{matrix} \\ \\ \\ \text{Roll} \\ \text{Pitch} \\ \text{Yaw} \end{matrix}, \quad \mathbf{x} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix}$$

Angular velocity components in B

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

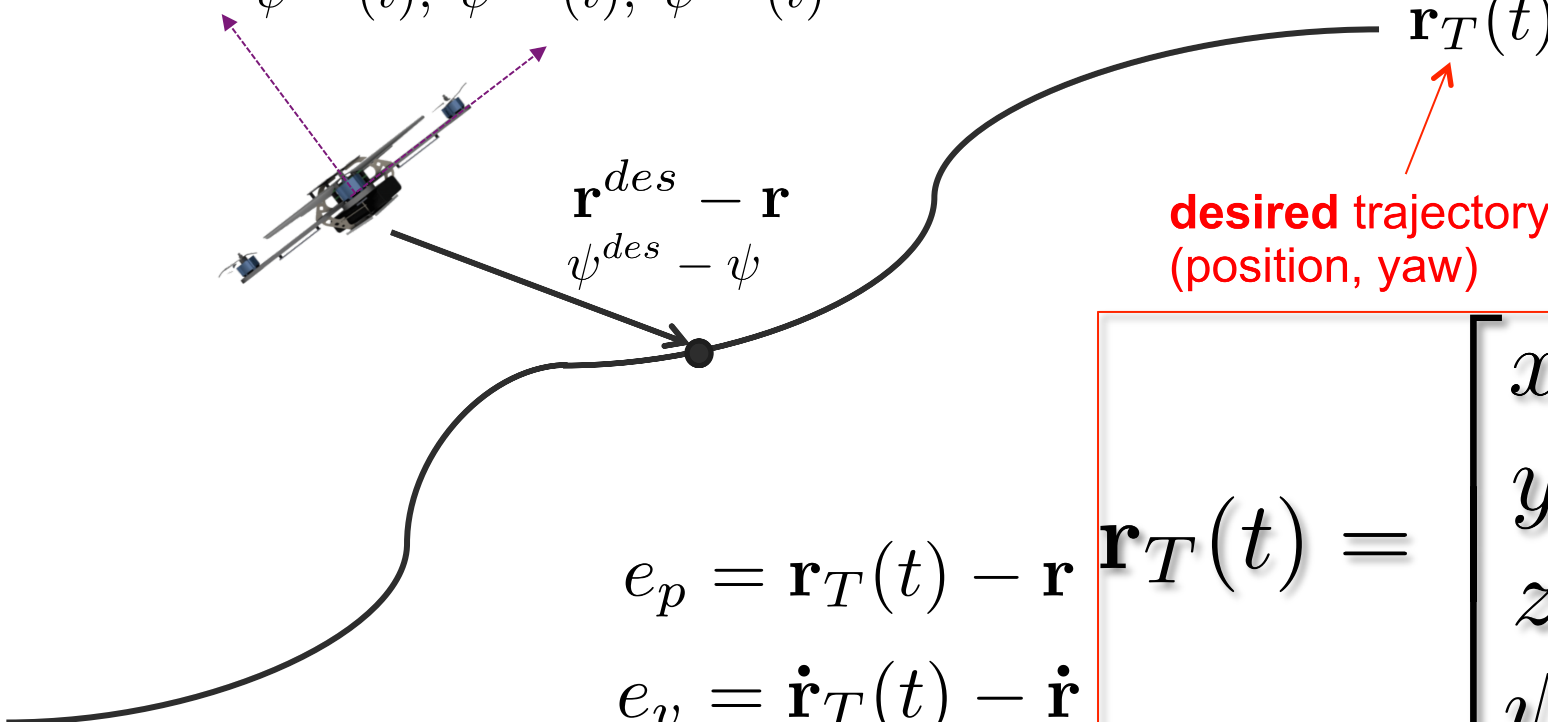


Trajectory Tracking in 3 Dimensions

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

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

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



desired trajectory (position, yaw)

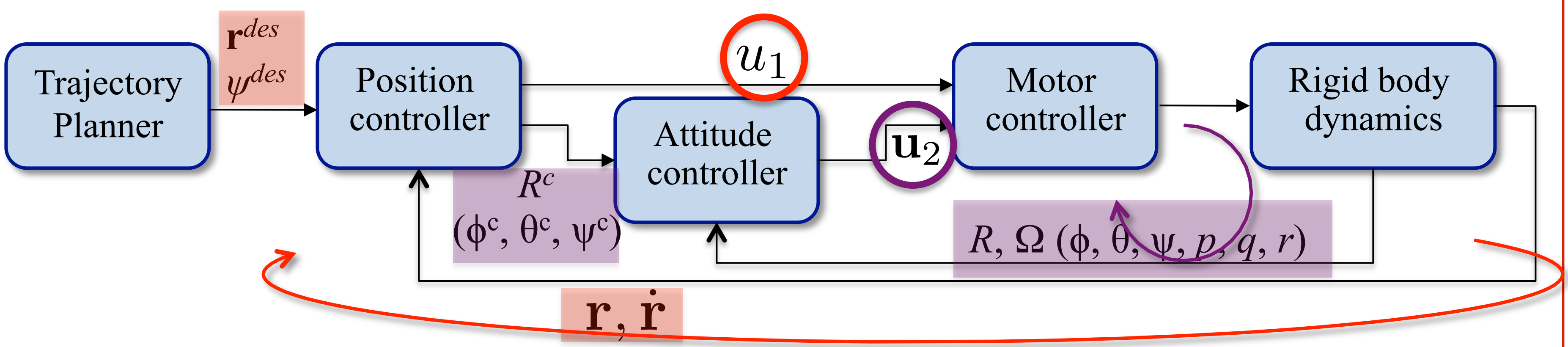
$$\mathbf{r}_T(t) = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \\ \psi(t) \end{bmatrix}$$

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

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

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



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

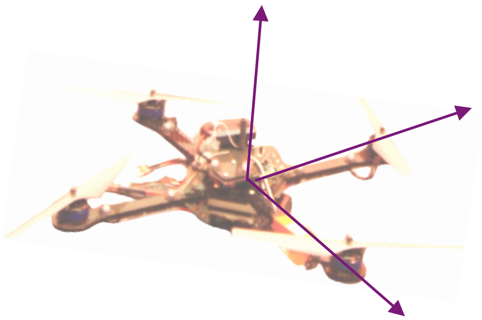
u_1

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

$\mathbf{u}_2$

Penn Engineering

Control for Hovering



Linearize the dynamics at the hover configuration

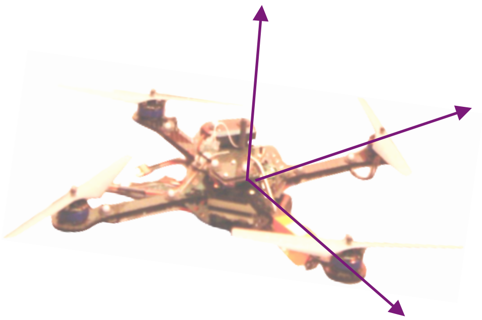
$$(u_1 \sim mg, \theta \sim 0, \phi \sim 0, \psi \sim \psi_0)$$

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

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

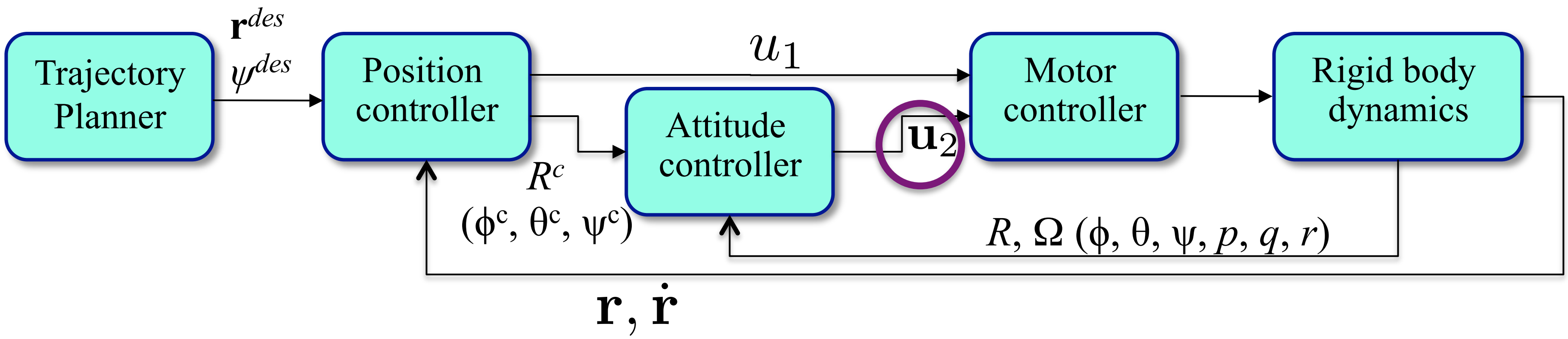
$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \underbrace{\begin{bmatrix} L(F_2 - F_4) \\ L(F_3 - F_1) \\ M_1 - M_2 + M_3 - M_4 \end{bmatrix}}_{\mathbf{u}_2} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

Control for Hovering



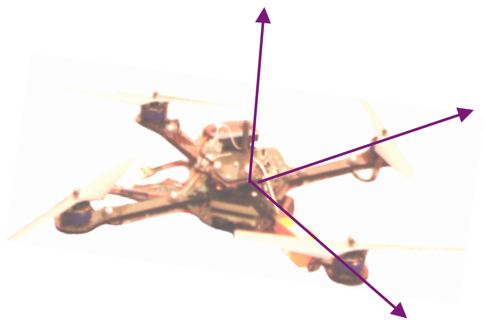
$$\left(u_2 \sim 0, p \sim 0, q \sim 0, r \sim 0\right)$$

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



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

Control for Hovering



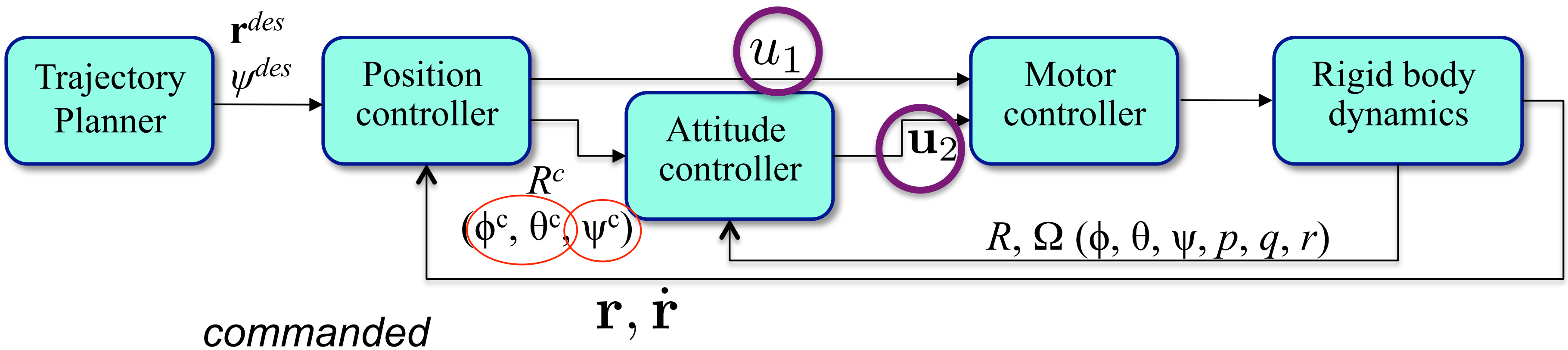
$$m\ddot{\mathbf{r}} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ \underbrace{\boxed{F_1 + F_2 + F_3 + F_4}}_{u_1} \end{bmatrix}$$

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

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

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

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

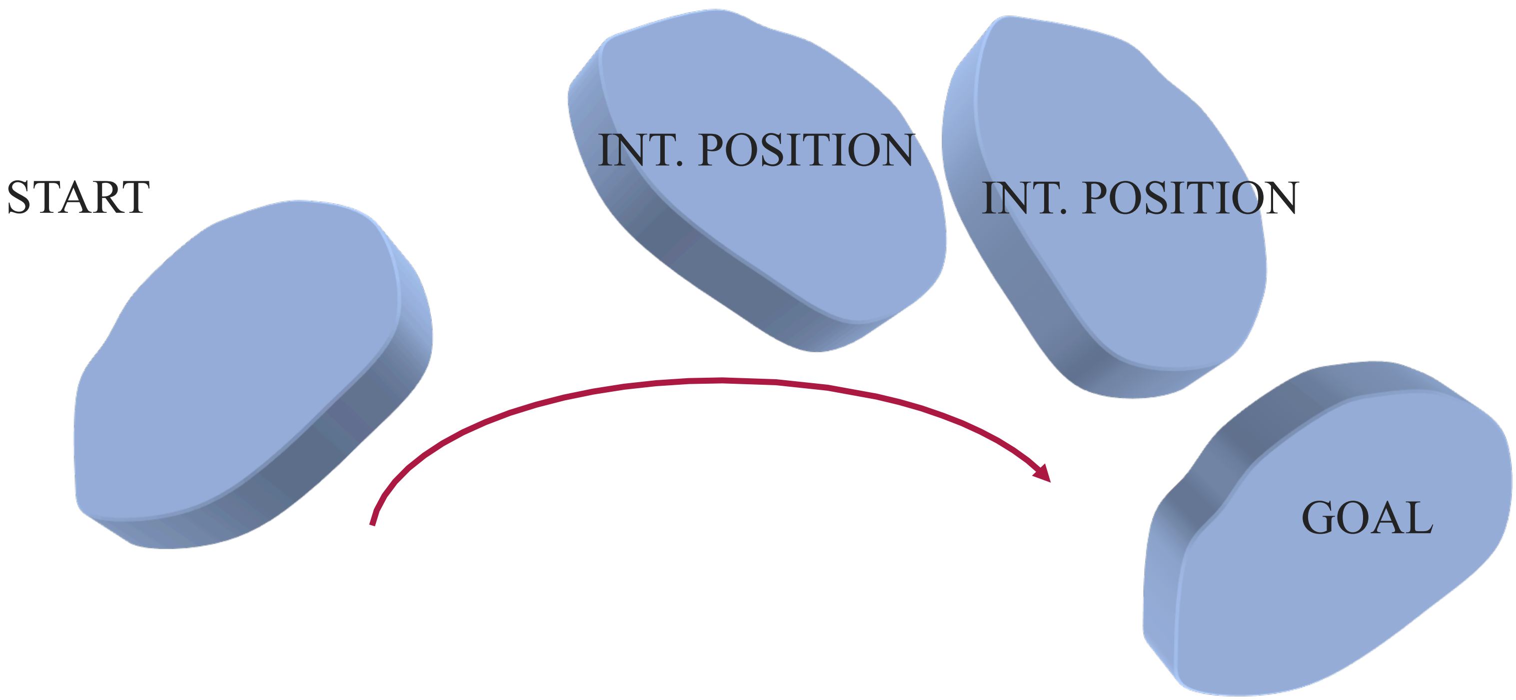
$$\psi_c = \psi^{des}$$

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

Penn Engineering

Time, Motion and Trajectories

Smooth three dimensional trajectories



Applications

- Trajectory generation in robotics
- Planning trajectories for quad rotors

Penn Engineering

General Set up

- Start, goal positions (orientations)
- Waypoint positions (orientations)
- Smoothness criterion
  - Generally translates to minimizing rate of change of "input"
- Order of the system (n)
  - Order of the system determines the input
  - Boundary conditions on $(n\text{-}1)^{\text{th}}$ order and lower derivatives

Calculus of Variations

$$x^{\star}(t) = \underset{x(t)}{\operatorname{argmin}} \int_0^T \mathcal{L}\left(\dot{x}, x, t\right) dt$$

functional

function

Examples

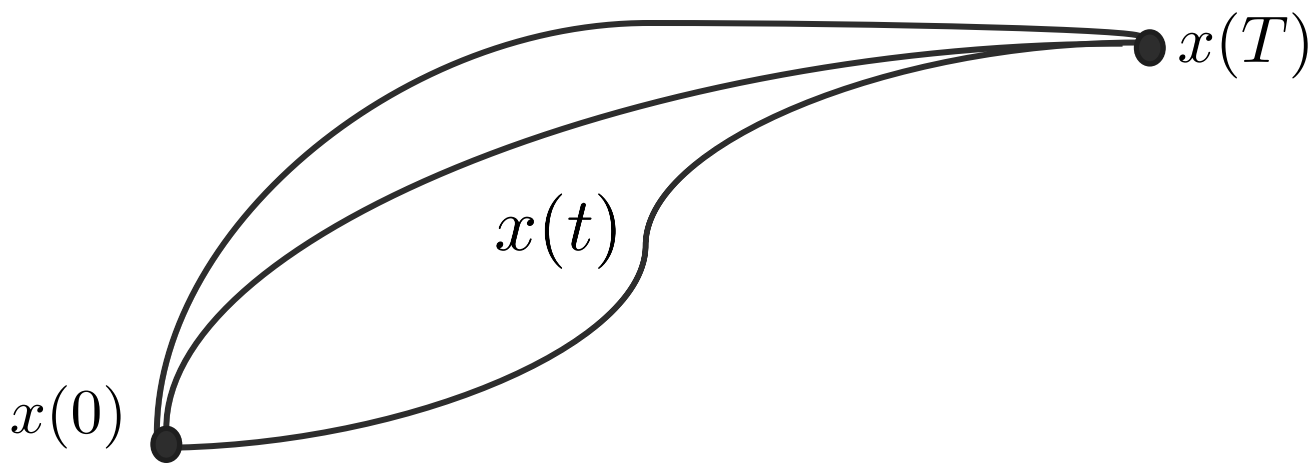
- Shortest distance path (geometry) $x^{\star}(t) = \arg\min_{x(t)} \int_0^T \dot{x}^2 dt$
- Fermat's principle (optics) $x^{\star}(t) = \underset{x(t)}{\operatorname{argmin}} \int_0^T 1 dt$
- Principle of least action (mechanics) $x^{\star}(t) = \underset{x(t)}{\operatorname{argmin}} \int_0^T T\left(\dot{x}, x, t\right) - V\left(x, t\right) dt$

Penn Engineering

Calculus of Variations

$$x^{\star}(t) = \underset{x(t)}{\operatorname{argmin}} \int_0^T \mathcal{L}(\dot{x}, x, t)\, dt$$

Consider the set of all differentiable curves, $x(t)$, with a given $x(0)$ and $x(T)$.



Calculus of Variations

$$x^\star(t) = \underset{x(t)}{\operatorname{argmin}} \int_0^T \mathcal{L}\left(\dot{x}, x, t\right) dt$$

Euler Lagrange Equation

Necessary condition satisfied by the "optimal" function $x(t)$

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{x}}\right) - \frac{\partial \mathcal{L}}{\partial x} = 0$$

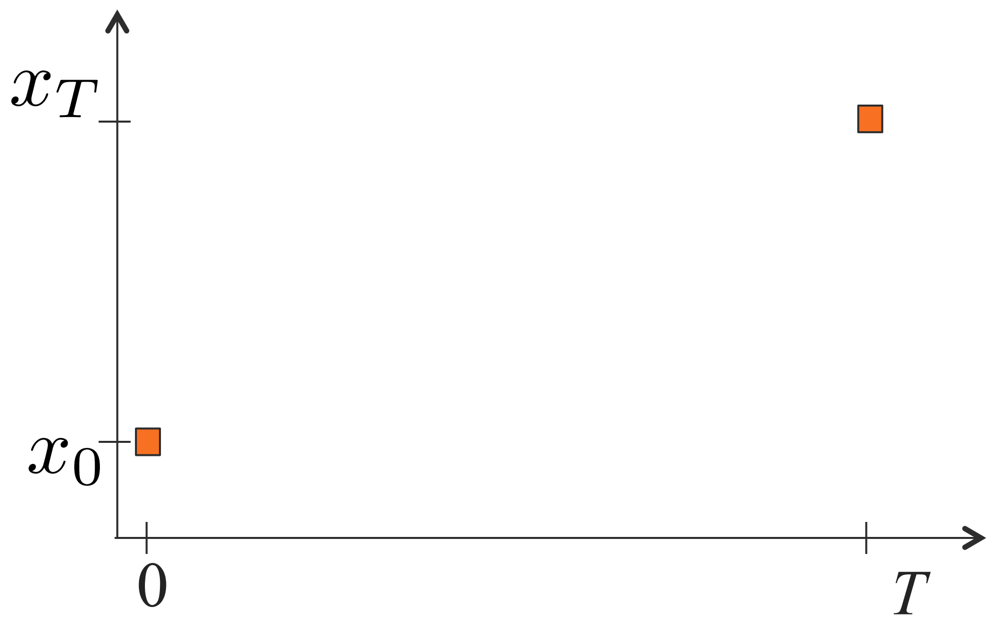
Smooth trajectories (n=1)

$$x^\star(t) = \arg\min_{x(t)} \int_0^T \dot{x}^2 dt$$

$$x(0) = x_0, \;\; x(T) = x_T$$

input

$$u = \dot{x}$$



Smooth trajectories (*n*=1)

$$x^{\star}(t) = \arg\min_{x(t)} \int_0^T \dot{x}^2 dt$$

Euler Lagrange Equation

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{x}}\right) - \frac{\partial \mathcal{L}}{\partial x} = 0$$

$$\mathcal{L}(\dot{x}, x, t) = (\dot{x})^2 \quad \Rightarrow \quad \ddot{x} = 0$$

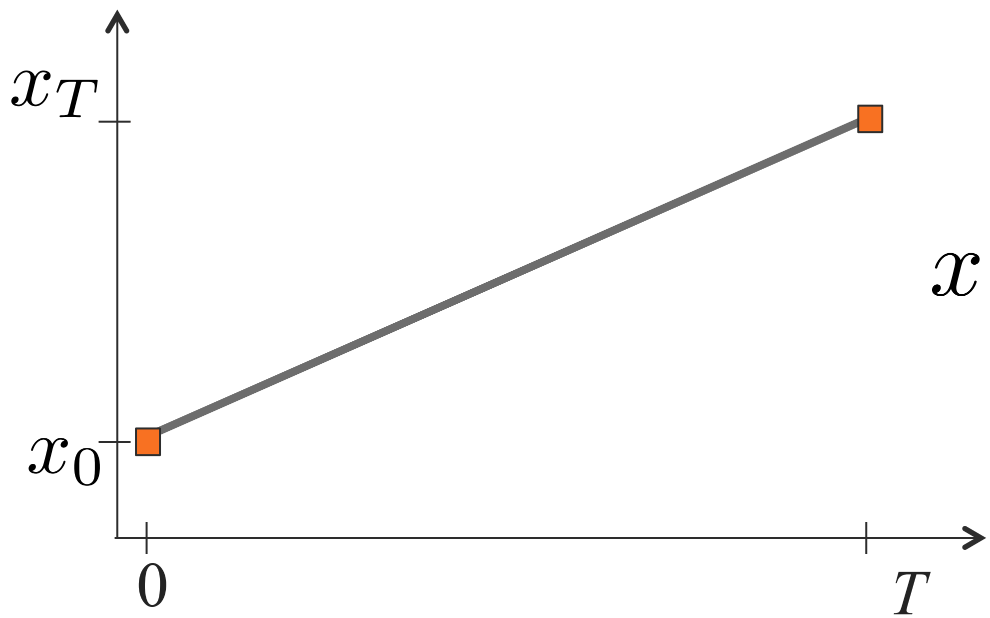
$$x = c_1 t + c_0$$

Penn Engineering

Smooth trajectories (*n*=1)

$$x^\star(t) = \arg\min_{x(t)} \int_0^T \dot{x}^2 dt$$

$$x(0) = x_0, \; x(T) = x_T$$



$$x = c_1 t + c_0$$

Smooth trajectories (general *n*)

$$x^\star(t) = \underset{x(t)}{\operatorname{argmin}} \int_0^T \left(x^{(n)}\right)^2 dt$$

input

$u = x^{(n)}$

Euler-Lagrange Equation

$$x^\star(t) = \underset{x(t)}{\operatorname{argmin}} \int_0^T \mathcal{L}\left(x^{(n)}, x^{(n-1)}, \ldots, \dot{x}, x, t\right) dt$$

Euler Lagrange Equation

Necessary condition satisfied by the "optimal" function

$$\frac{\partial \mathcal{L}}{\partial x} - \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{x}}\right) + \frac{d^2}{dt^2}\left(\frac{\partial \mathcal{L}}{\partial \ddot{x}}\right) + \ldots + (-1)^n \frac{d^n}{dt^n}\left(\frac{\partial \mathcal{L}}{\partial x^{(n)}}\right) = 0$$

Smooth Trajectories

$$x^\star(t) = \underset{x(t)}{\operatorname{argmin}} \int_0^T \left(x^{(n)}\right)^2 dt$$

- n=1, shortest distance $\boxed{\text{velocity}}$
- n=2, minimum $\boxed{\text{acceleration}}$
- n=3, minimum $\boxed{\text{jerk}}$
- n=4, minimum $\boxed{\text{snap}}$

n – order of system
n^{th} derivative is input

Smooth Trajectories

$$x^{\star}(t) = \underset{x(t)}{\operatorname{argmin}} \int_0^T \left(x^{(n)}\right)^2 dt$$

- *n*=1, shortest distance $\boxed{\text{velocity}}$
- *n*=2, minimum $\boxed{\text{acceleration}}$
- *n*=3, minimum $\boxed{\text{jerk}}$
- *n*=4, minimum $\boxed{\text{snap}}$

Why is the minimum velocity curve also the shortest distance curve?

Minimum Jerk Trajectory

Design a trajectory $x(t)$ such that $x(0) = a$, $x(T) = b$

$$x^{\star}(t) = \underset{x(t)}{\operatorname{argmin}} \int_0^T \mathcal{L}\left(\dddot{x}, \ddot{x}, \dot{x}, x, t\right) dt$$

$$\mathcal{L} = \left(\dddot{x}\right)^2$$

Euler-Lagrange:

$$\boxed{\frac{\partial \mathcal{L}}{\partial x} - \cancelto{0}{\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{x}}\right)} + \frac{d^2}{dt^2}\left(\frac{\partial \mathcal{L}}{\partial \ddot{x}}\right)} \boxed{- \frac{d^3}{dt^3}\left(\frac{\partial \mathcal{L}}{\partial x^{(3)}}\right)} = 0$$

$$x^{(6)} = 0$$

$$x = c_5 t^5 + c_4 t^4 + c_3 t^3 + c_2 t^2 + c_1 t + c_0$$

Solving for Coefficients

$$x = c_5 t^5 + c_4 t^4 + c_3 t^3 + c_2 t^2 + c_1 t + c_0$$

Boundary conditions:

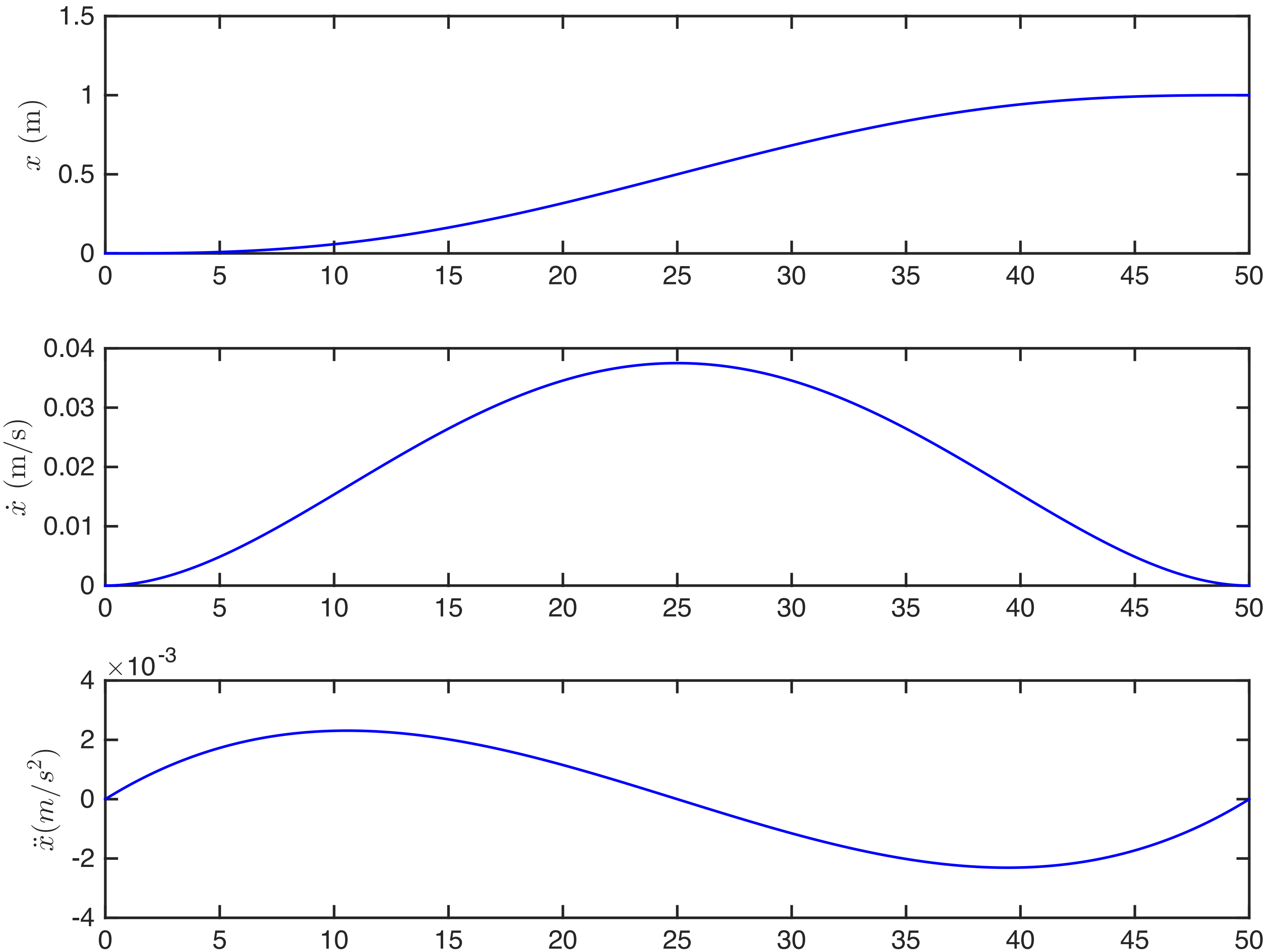
	Position	**Velocity**	**Acceleration**
$t = 0$	a	0	0
$t = T$	b	0	0

Solve:

$$\begin{bmatrix} a \\ b \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ T^5 & T^4 & T^3 & T^2 & T & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 5T^4 & 4T^3 & 3T^2 & 2T & 1 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 20T^3 & 12T^2 & 6T & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} c_5 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix}$$

Penn Engineering

Minimum Jerk Trajectory



a=0, b=1, T=50

Extensions to multiple dimensions

$$(x^\star(t), y^\star(t)) = \arg \min_{x(t),y(t)} \int_0^T \mathcal{L}\left(\dot{x}, \dot{y}, x, y, t\right) dt$$

Euler Lagrange Equation

Necessary condition satisfied by the "optimal" function

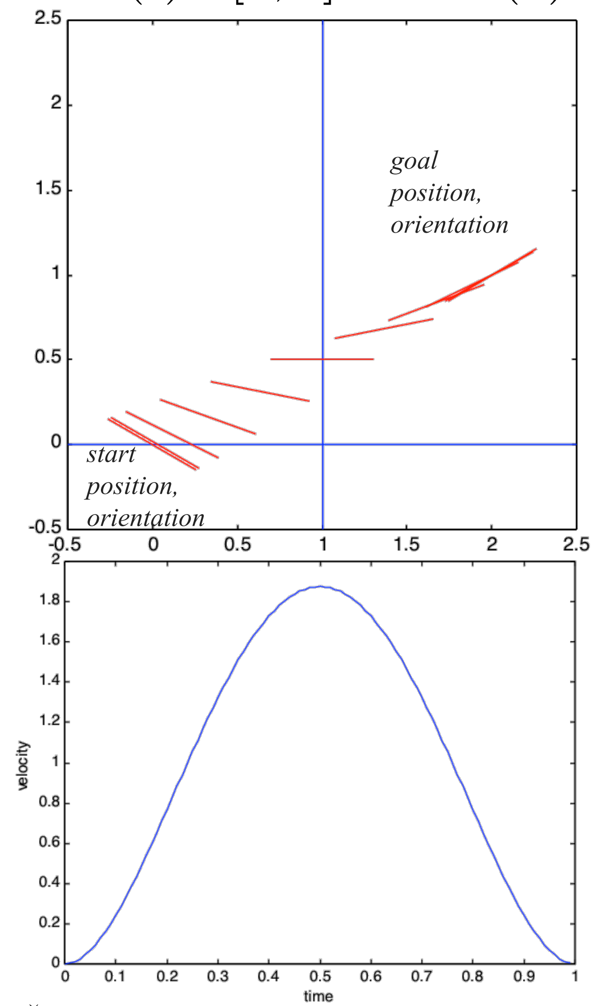
$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{x}}\right) - \frac{\partial \mathcal{L}}{\partial x} = 0$$
$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{y}}\right) - \frac{\partial \mathcal{L}}{\partial y} = 0$$

Minimum Jerk for Planar Motions

Minimum-jerk trajectory in (x, y, θ)

$$\min_{x(t),y(t),\theta(t)} \int_0^1 \left(\dddot{x}^2 + \dddot{y}^2 + \dddot{\theta}^2\right) dt$$

$$\mathbf{x}^\star(t) : [0,1] \rightarrow SE(2)$$



Human two-handed manipulation tasks

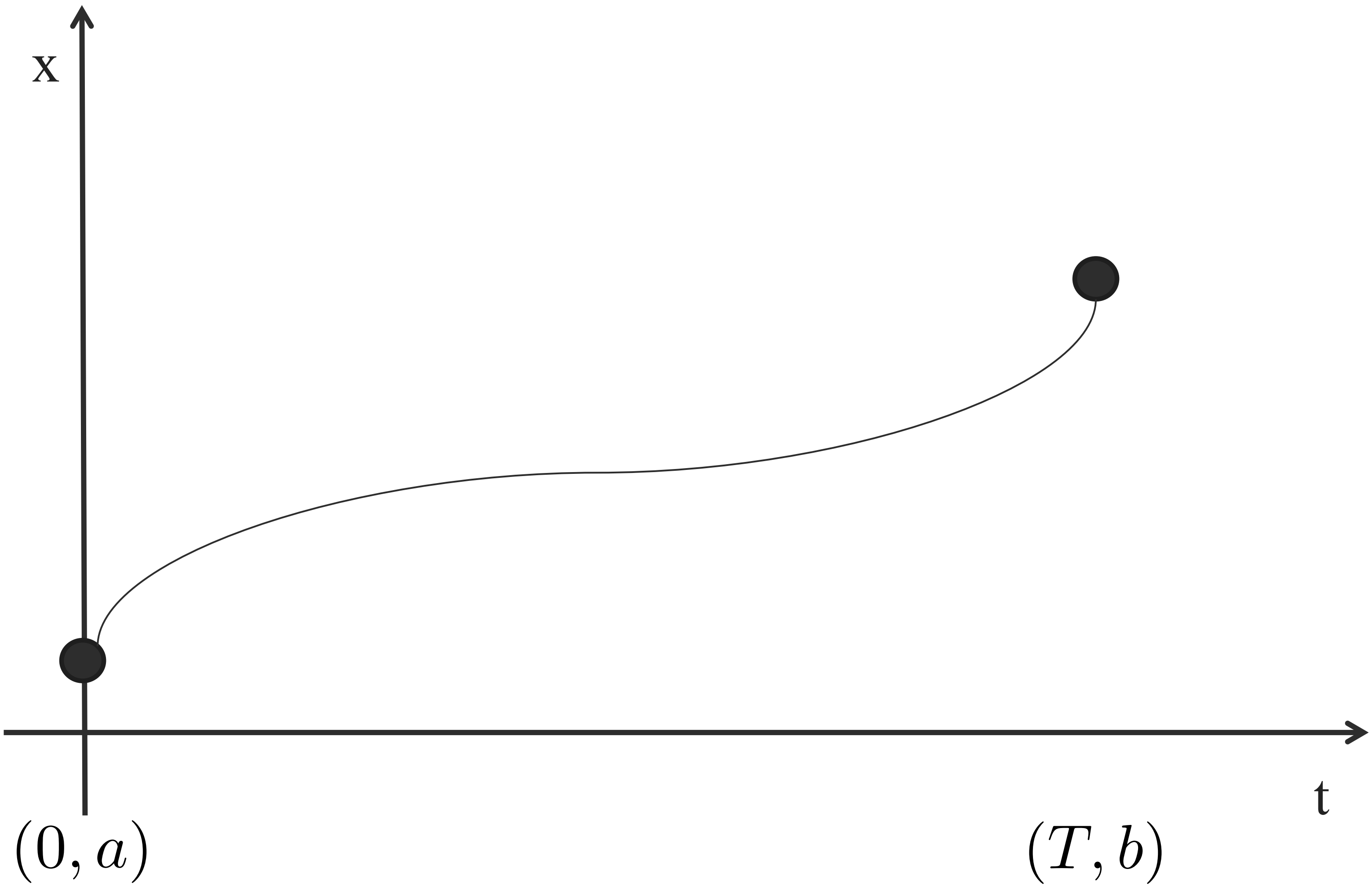
- Noise in the neural control signal increases with size of the control signal
- Rate of change of muscle fiber lengths is critical in relaxed, voluntary motions

G.J. Garvin, M. Žefran, E.A. Henis, V. Kumar, Two-arm trajectory planning in a manipulation task, *Biological Cybernetics,* January 1997, Volume 76, Issue 1, pp 53-62

Waypoint Navigation

Smooth 1D Trajectories

Design a trajectory $x(t)$ such that $x(0) = a$, $x(T) = b$



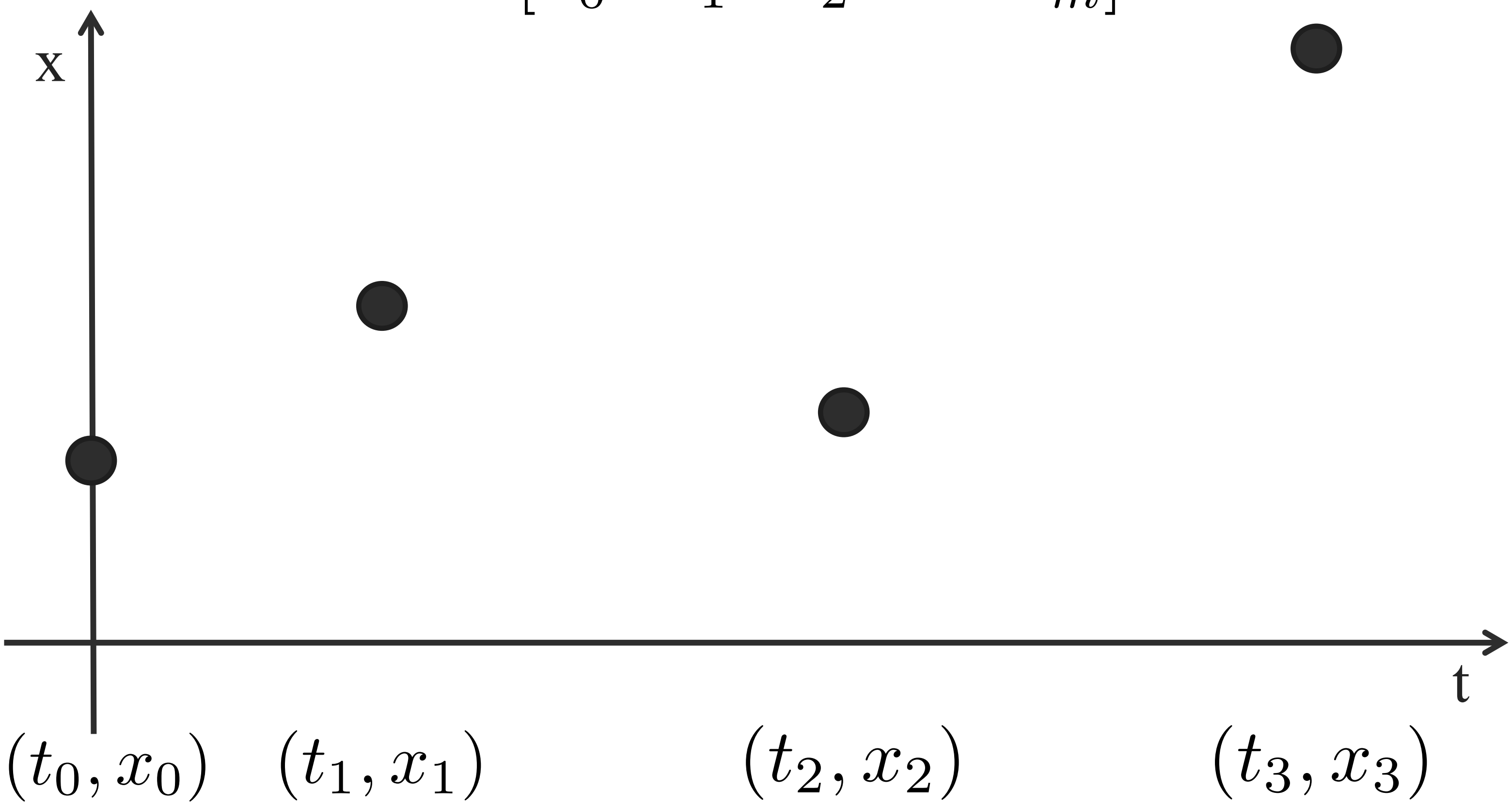
Penn Engineering

Multi-Segment 1D Trajectories

Design a trajectory x(t) such that:

$$t = [t_0 \quad t_1 \quad t_2 \quad \ldots \quad t_m]^T$$

$$x = [x_0 \quad x_1 \quad x_2 \quad \ldots \quad x_m]^T$$



Multi-Segment 1D Trajectories

Design a trajectory x(t) such that:

$$t = [t_0 \quad t_1 \quad t_2 \quad \dots \quad t_m]^T$$

$$x = [x_0 \quad x_1 \quad x_2 \quad \dots \quad x_m]^T$$

Define piecewise continuous trajectory:

$$x(t) = \begin{cases} x_1(t), & t_0 \leq t < t_1 \\ x_2(t), & t_1 \leq t < t_2 \\ \dots \\ x_m(t), & t_{m-1} \leq t < t_m \end{cases}$$

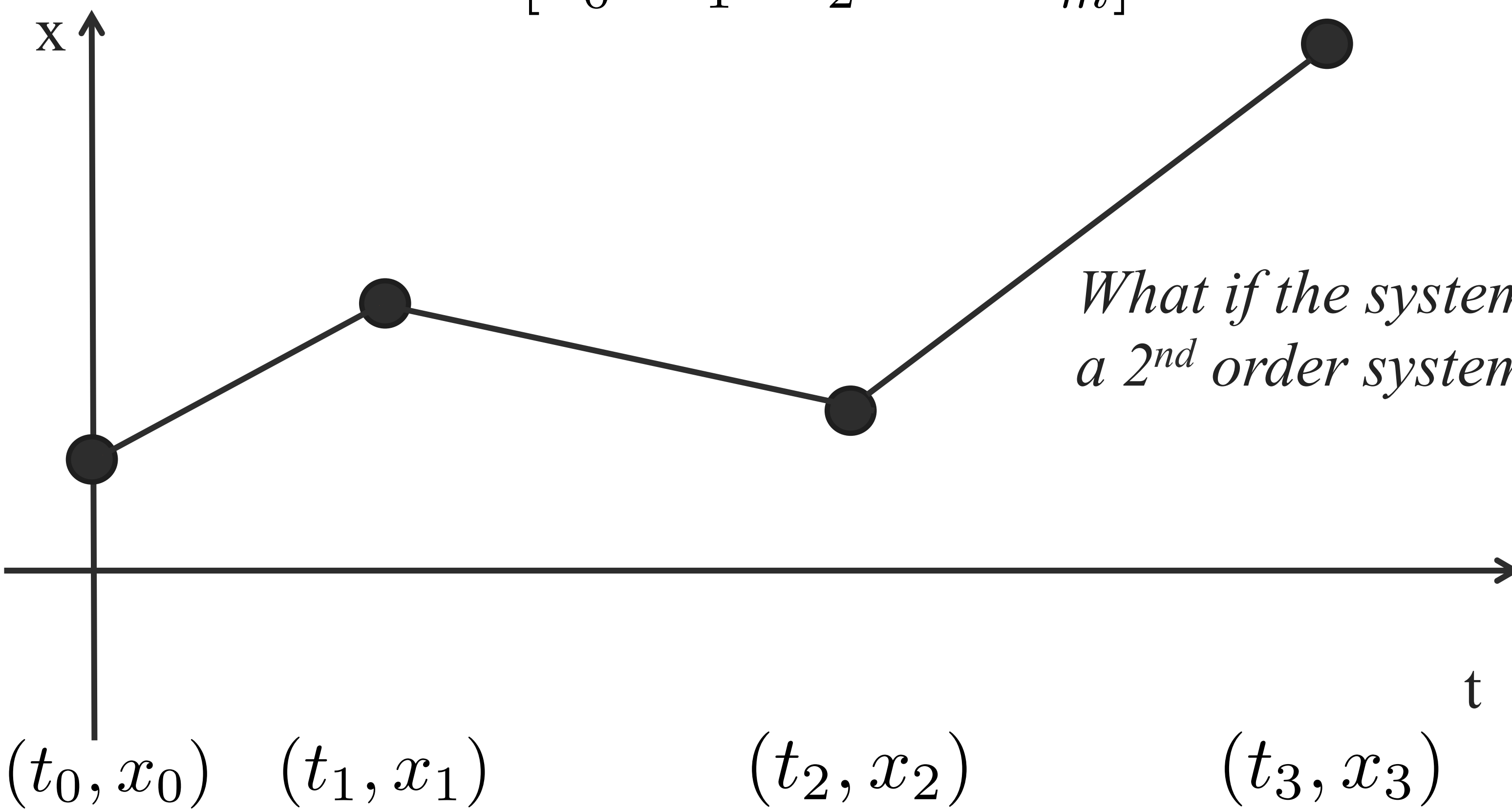
Penn Engineering

Continuous but not Differentiable

Design a trajectory $x(t)$ such that:

$$t = [t_0 \quad t_1 \quad t_2 \quad \ldots \quad t_m]^T$$

$$x = [x_0 \quad x_1 \quad x_2 \quad \ldots \quad x_m]^T$$



What if the system is a 2nd order system?

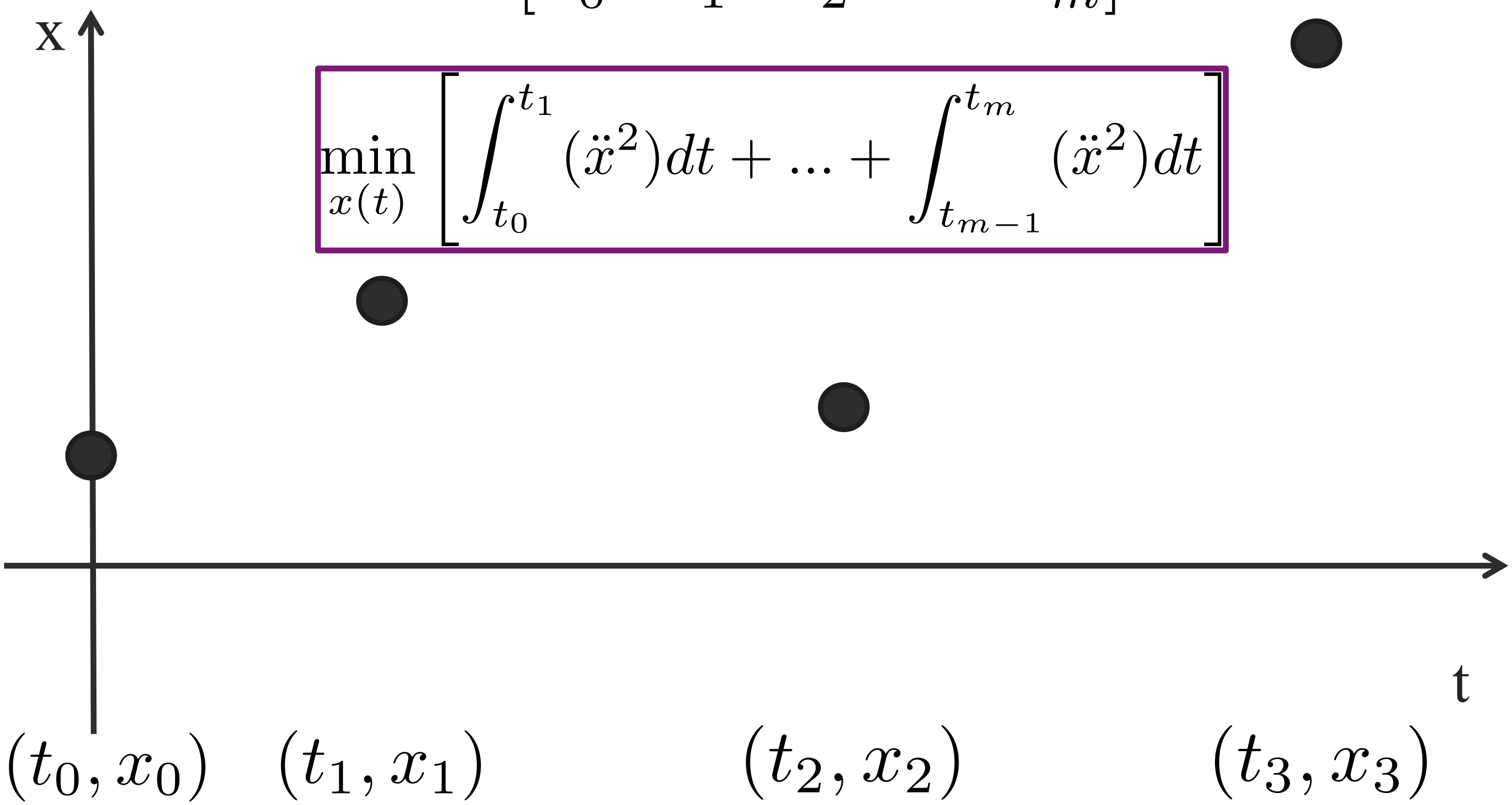
Minimum Acceleration Curve for 2nd Order Systems

Design a trajectory $x(t)$ such that:

$$t = [t_0 \quad t_1 \quad t_2 \quad \ldots \quad t_m]^T$$

$$x = [x_0 \quad x_1 \quad x_2 \quad \ldots \quad x_m]^T$$

$$\boxed{\min_{x(t)} \left[\int_{t_0}^{t_1} (\ddot{x}^2) dt + \ldots + \int_{t_{m-1}}^{t_m} (\ddot{x}^2) dt \right]}$$



Cubic Spline

Design a trajectory x(t) such that:

$$t = [t_0 \quad t_1 \quad t_2 \quad \dots \quad t_m]^T$$

$$x = [x_0 \quad x_1 \quad x_2 \quad \dots \quad x_m]^T$$

$$\boxed{\min_{x(t)} \left[\int_{t_0}^{t_1} (\ddot{x}^2) dt + \dots + \int_{t_{m-1}}^{t_m} (\ddot{x}^2) dt \right]}$$

$$x(t) = \begin{cases} x_1(t) = c_{1,3}t^3 + c_{1,2}t^2 + c_{1,1}t + c_{1,0}, & t_0 \le t < t_1 \\ x_2(t) = c_{2,3}t^3 + c_{2,2}t^2 + c_{2,1}t + c_{2,0}, & t_1 \le t < t_2 \\ \dots & \\ x_m(t) = c_{m,3}t^3 + c_{m,2}t^2 + c_{m,1}t + c_{m,0}, & t_{m-1} \le t < t_m \end{cases}$$

$4m$ degrees of freedom

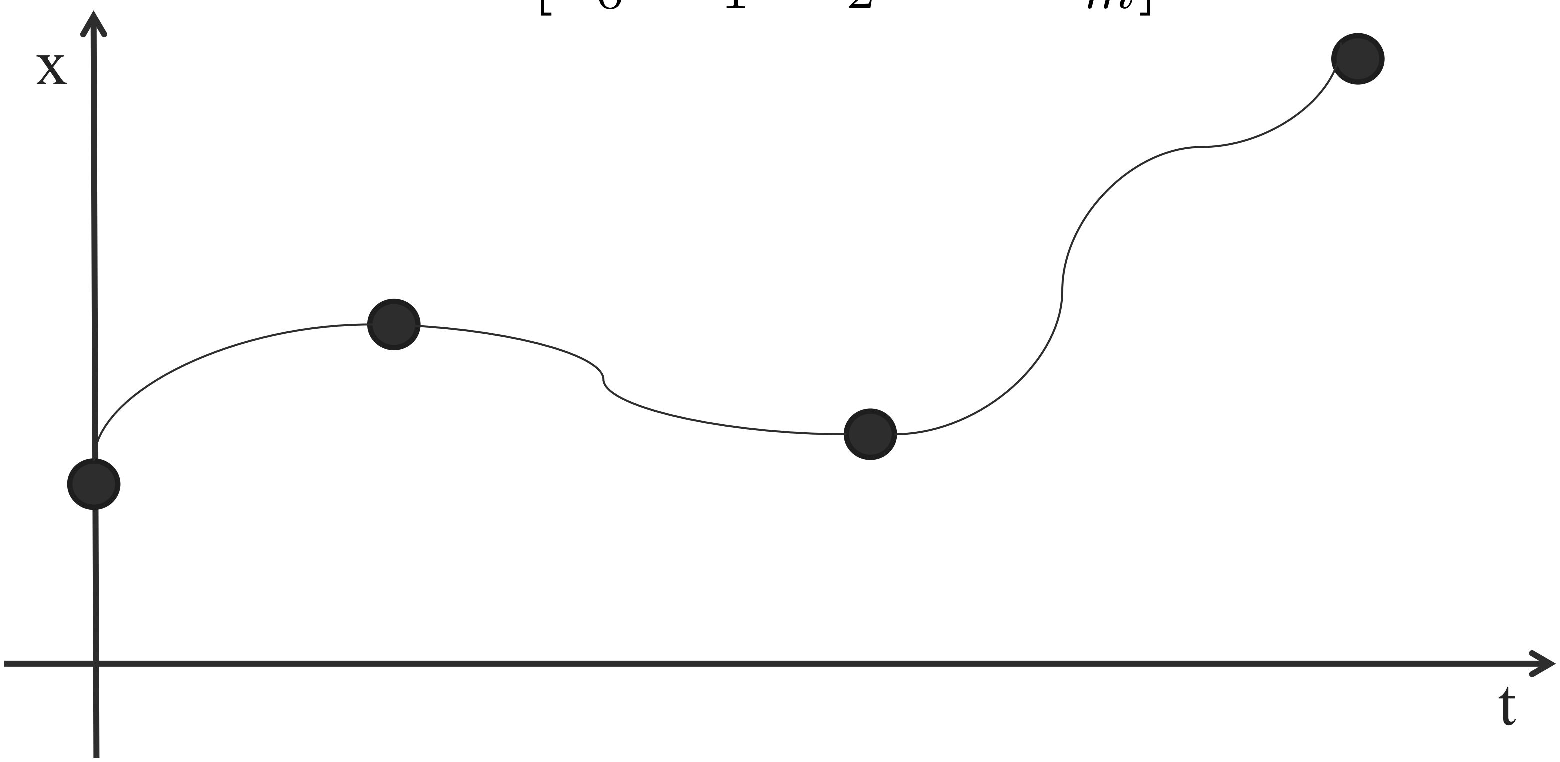
Penn Engineering

Cubic Spline

Design a trajectory x(t) such that:

$$t = [t_0 \quad t_1 \quad t_2 \quad \ldots \quad t_m]^T$$

$$x = [x_0 \quad x_1 \quad x_2 \quad \ldots \quad x_m]^T$$

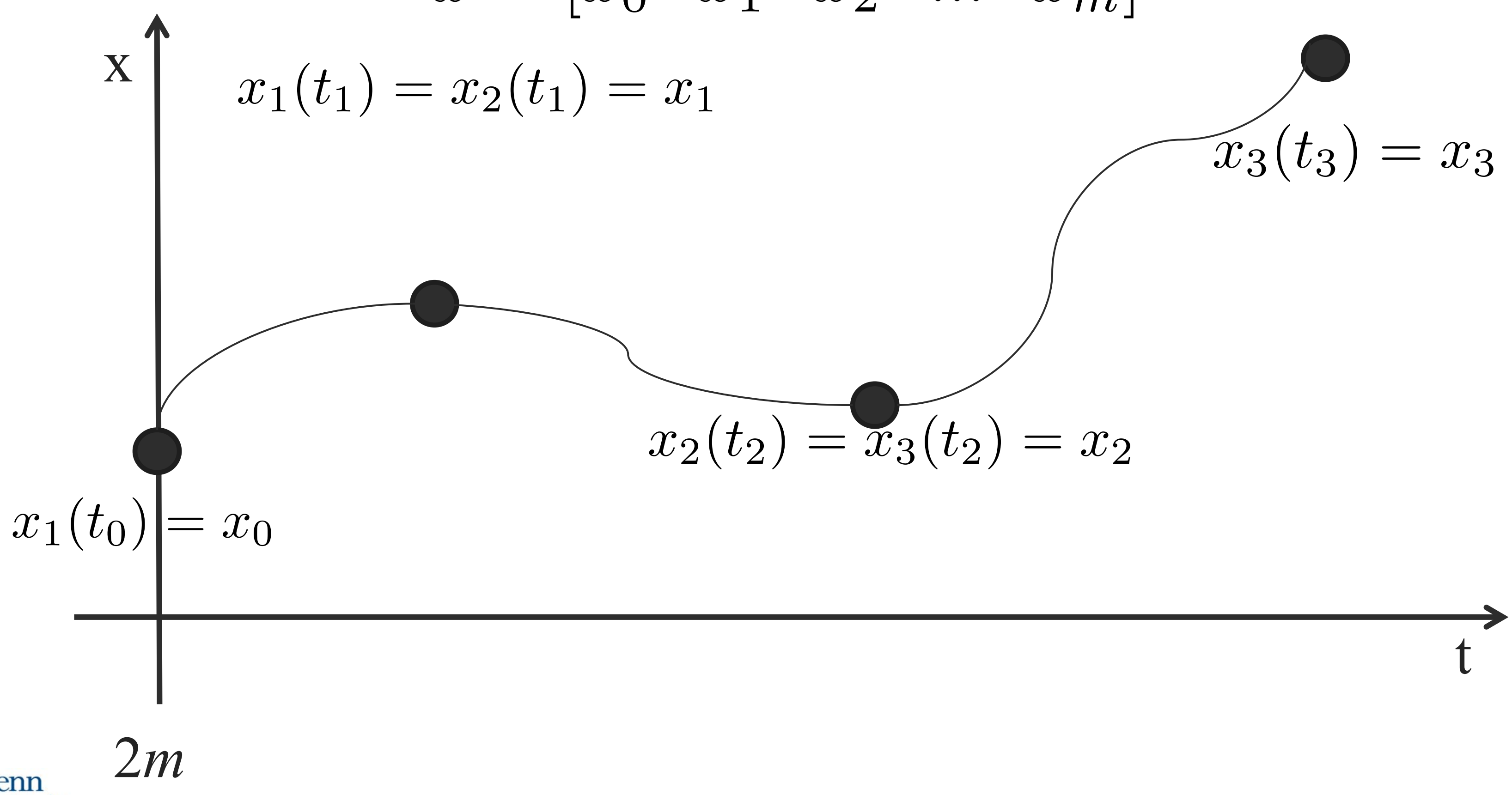


Cubic Spline

Design a trajectory x(t) such that:

$$t = [t_0 \quad t_1 \quad t_2 \quad \dots \quad t_m]^T$$

$$x = [x_0 \quad x_1 \quad x_2 \quad \dots \quad x_m]^T$$

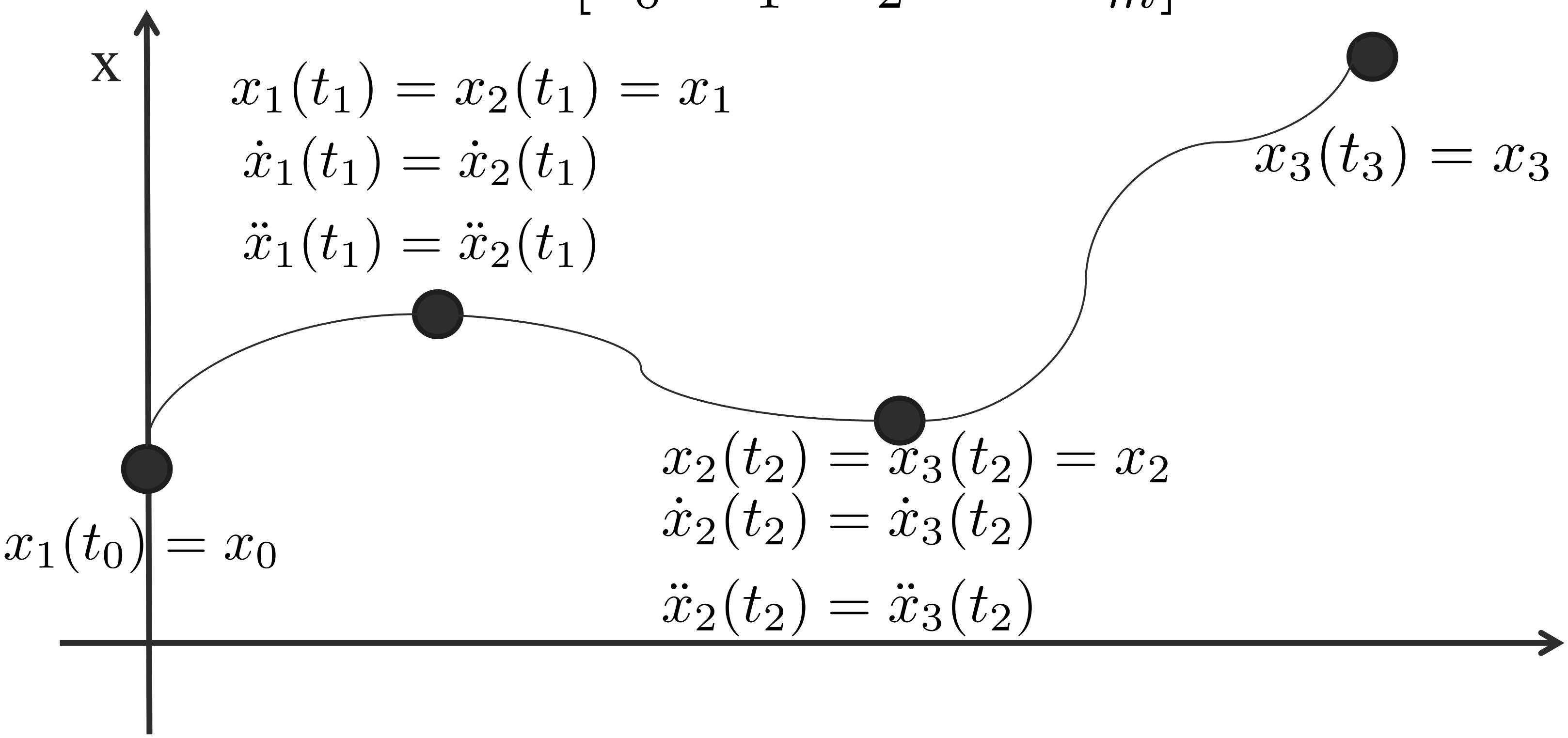


Cubic Spline

Design a trajectory x(t) such that:

$$t = [t_0 \quad t_1 \quad t_2 \quad \dots \quad t_m]^T$$

$$x = [x_0 \quad x_1 \quad x_2 \quad \dots \quad x_m]^T$$



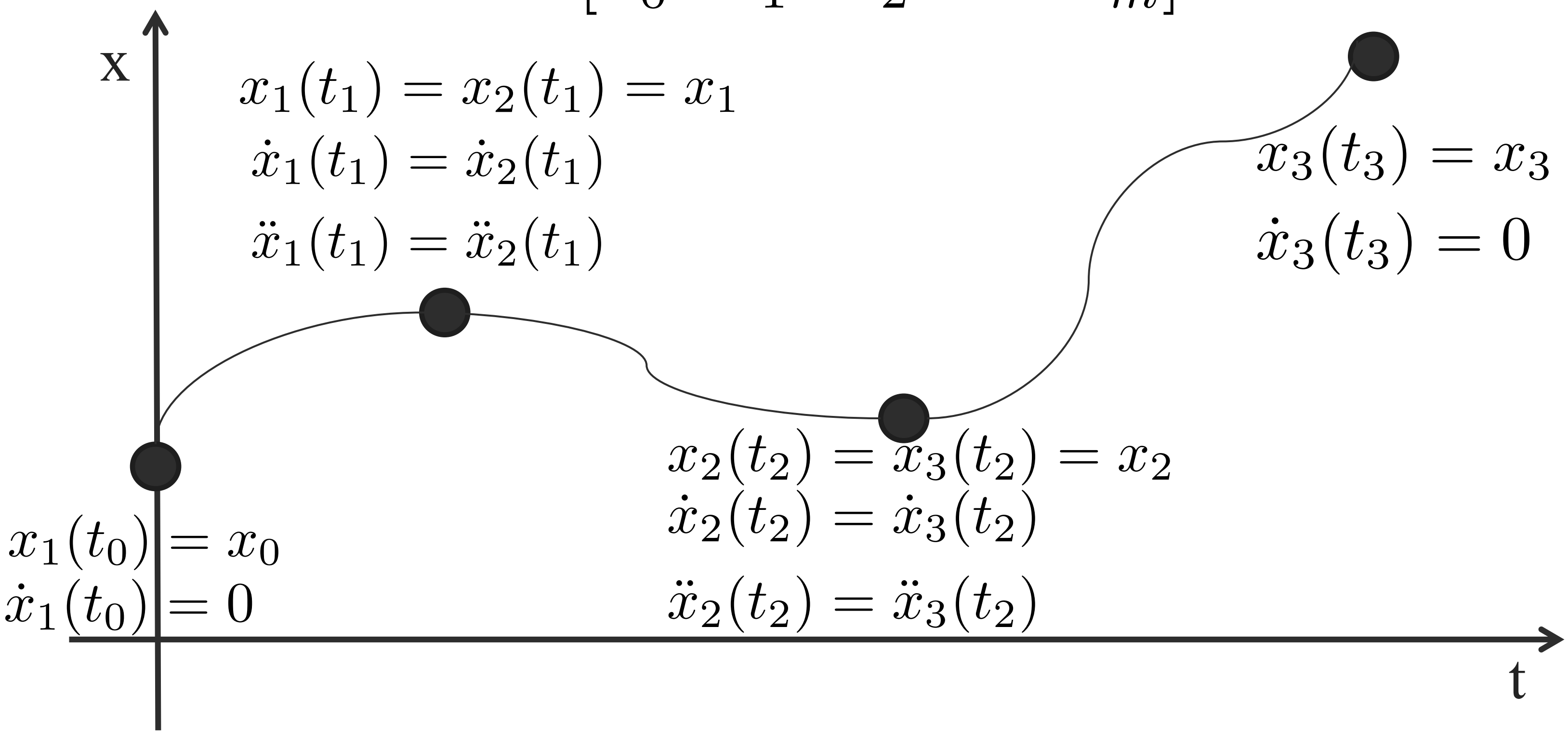
$2m+2(m-1)$

Cubic Spline

Design a trajectory x(t) such that:

$$t = [t_0 \quad t_1 \quad t_2 \quad \ldots \quad t_m]^T$$

$$x = [x_0 \quad x_1 \quad x_2 \quad \ldots \quad x_m]^T$$



$2m+2(m-1)+2 = 4m$ constraints

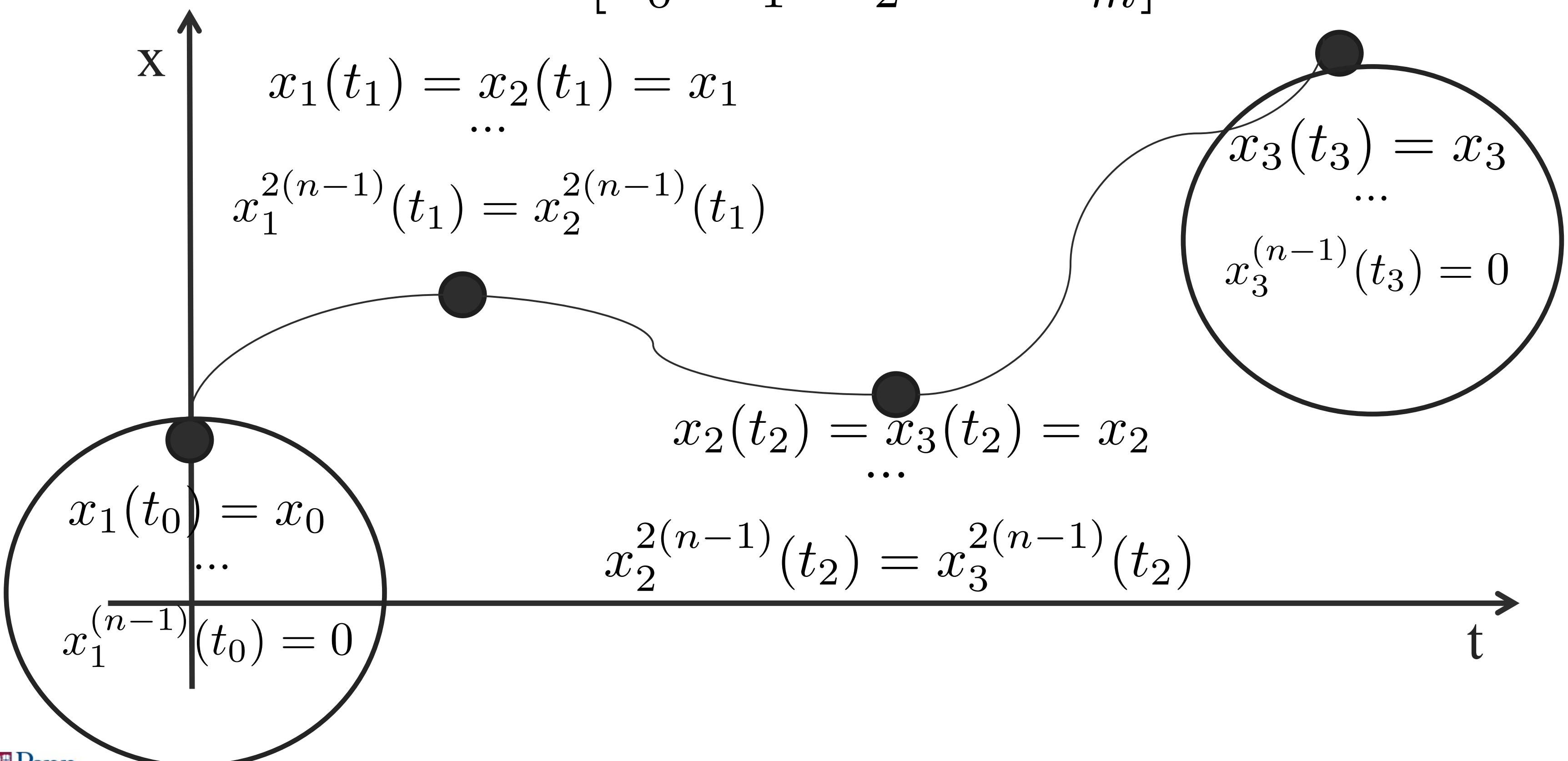
Penn Engineering

Spline for nth order system

Design a trajectory x(t) such that:

$$t = [t_0 \quad t_1 \quad t_2 \quad \dots \quad t_m]^T$$

$$x = [x_0 \quad x_1 \quad x_2 \quad \dots \quad x_m]^T$$

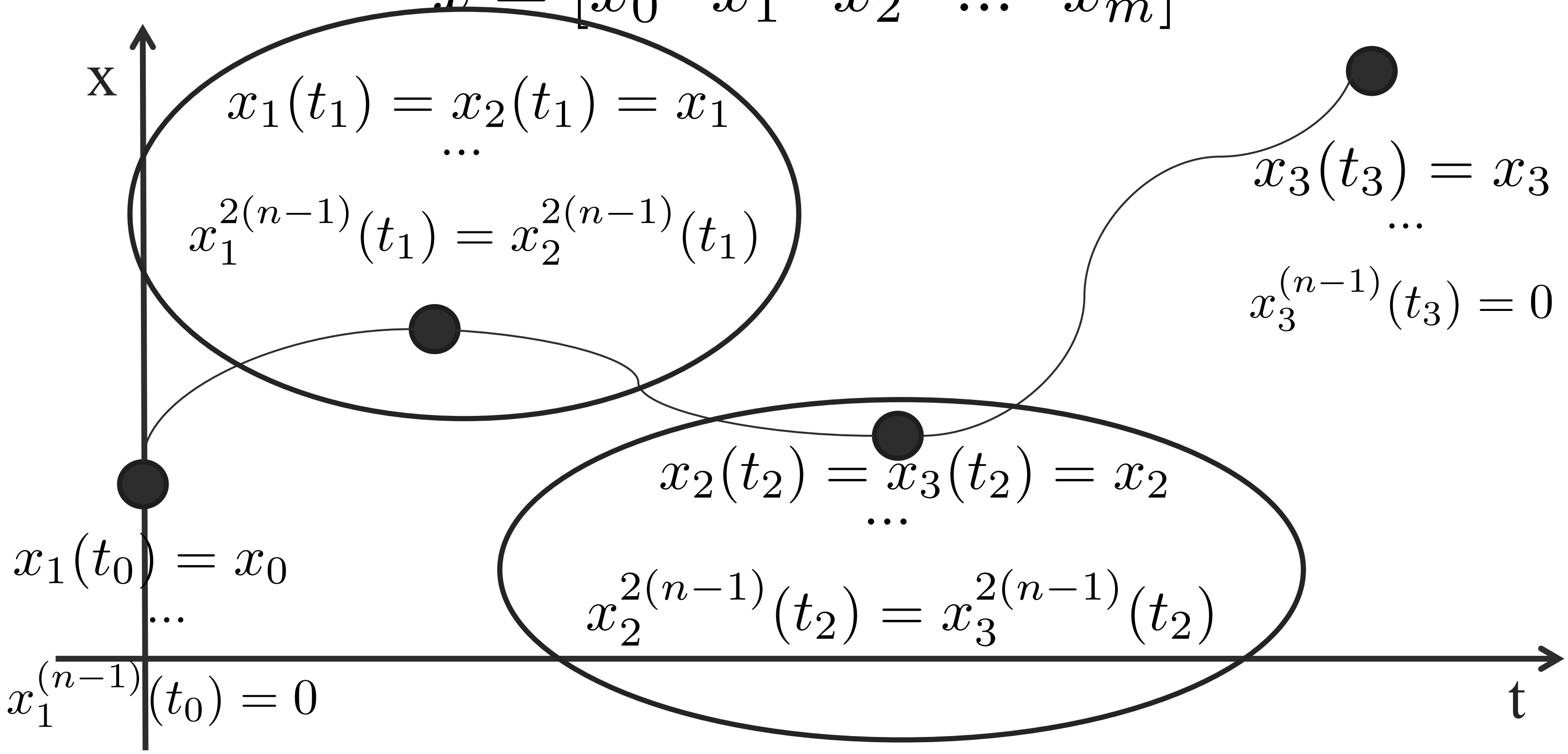


Spline for nth order system

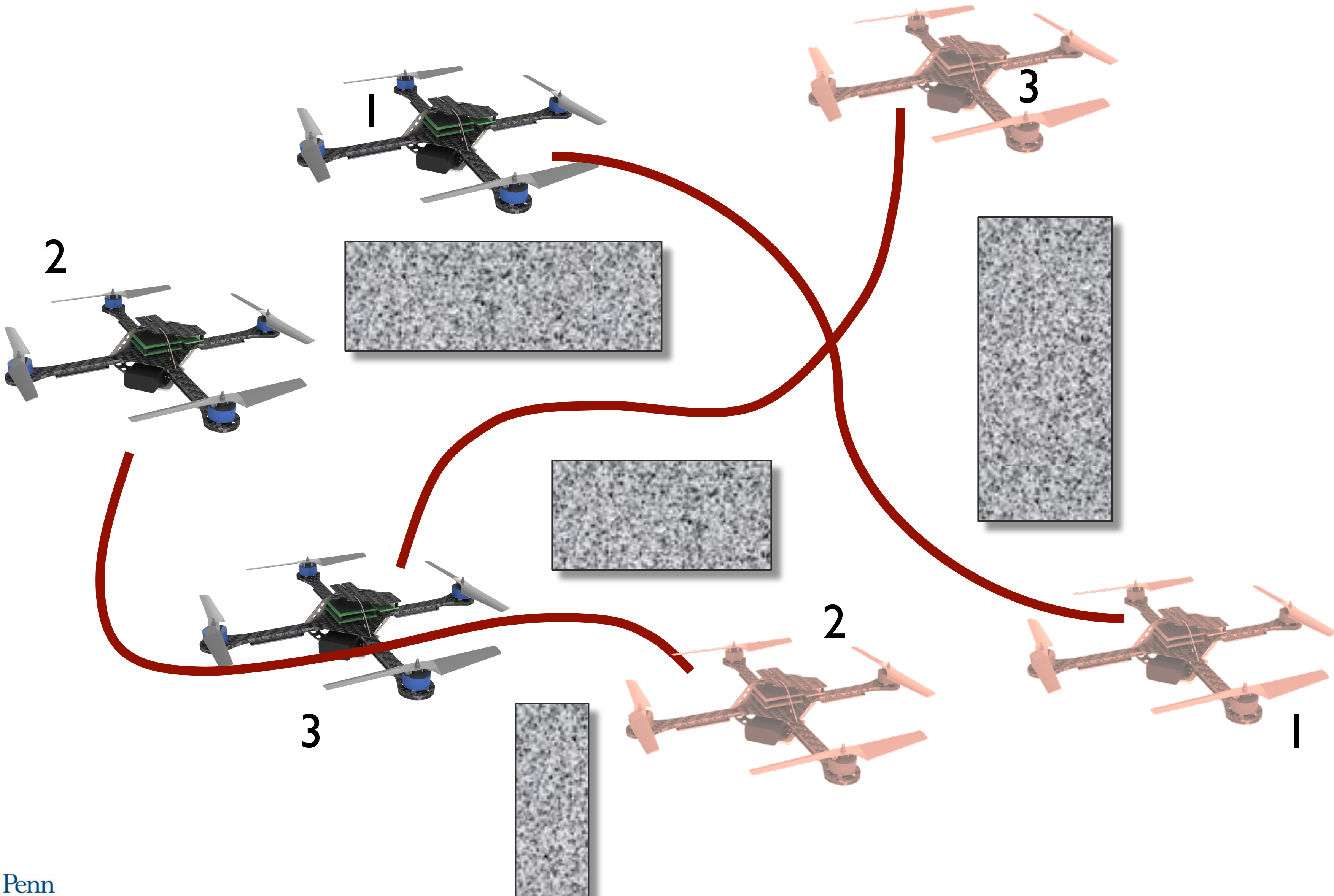
Design a trajectory x(t) such that:

$$t = [t_0 \quad t_1 \quad t_2 \quad \dots \quad t_m]^T$$

$$x = [x_0 \quad x_1 \quad x_2 \quad \dots \quad x_m]^T$$

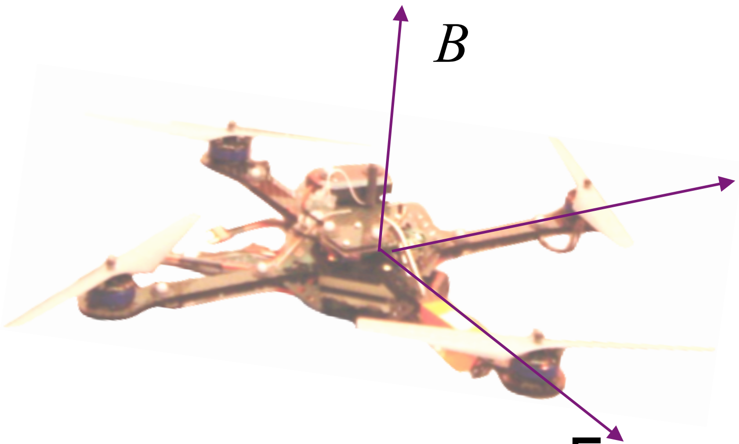


Motion Planning of Quadrotors



Motion Planning for Quadrotors

Penn Engineering



Newton-Euler Equations

$${}^{A}\boldsymbol{\omega}^{B} = p\,\mathbf{b}_1 + q\,\mathbf{b}_2 + r\,\mathbf{b}_3$$

Rotation of thrust vector from B to A

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

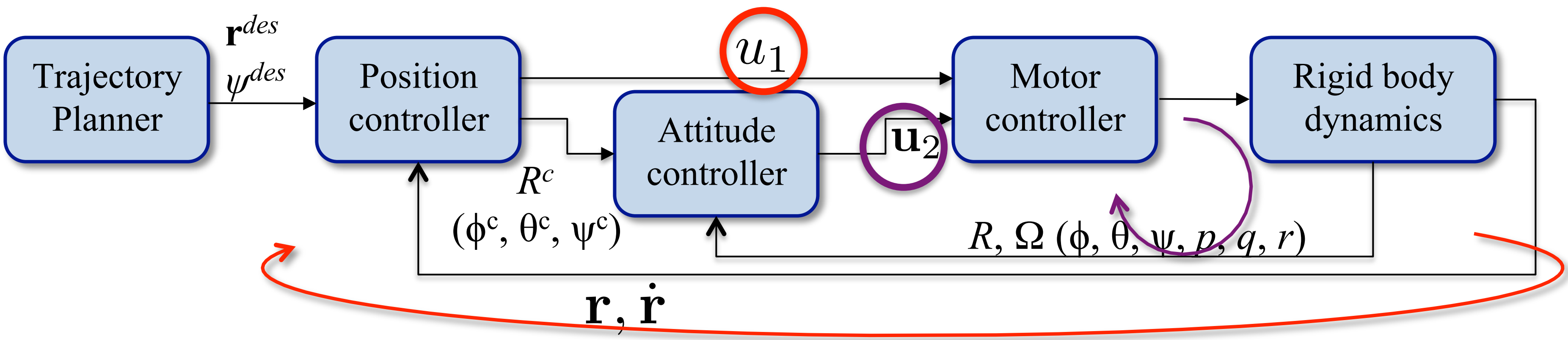
Components in the inertial frame along $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{a}_3$

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

Components in the body frame along $\mathbf{b}_1$, $\mathbf{b}_2$, and $\mathbf{b}_3$, the principal axes

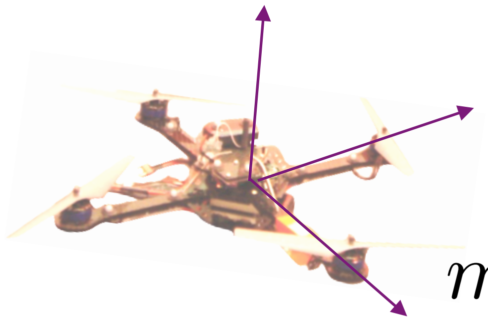
Penn Engineering

Position Control



Position control loop relies on an inner attitude control loop

The fourth derivative of position depends on $\mathbf{u}_2$



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

$\boxed{R(\theta, \phi, \psi)}$

$\boldsymbol{u}_1$

The second derivative of position depends on u_1

The second derivative of the rotation matrix depends on $\mathbf{u}_2$

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

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

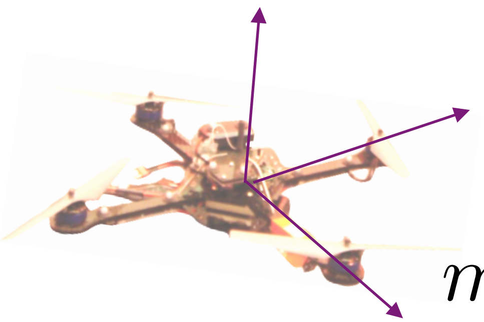
$\mathbf{u}_2$

Penn Engineering

Linearized Model

$(\theta \sim 0, \phi \sim 0, \psi \sim 0)$

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



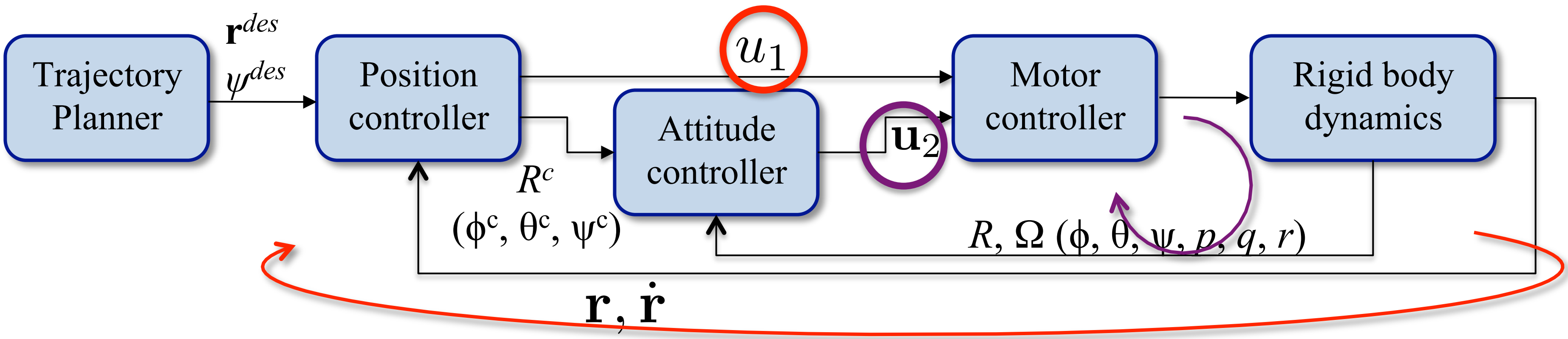
$$m\ddot{\mathbf{r}} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + \overset{I}{\cancel{R}} \begin{bmatrix} 0 \\ 0 \\ \underbrace{F_1 + F_2 + F_3 + F_4}_{u_1} \end{bmatrix}$$

$I \quad \boxed{R(\theta, \phi, \psi)}$

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

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

Minimum Snap Trajectory



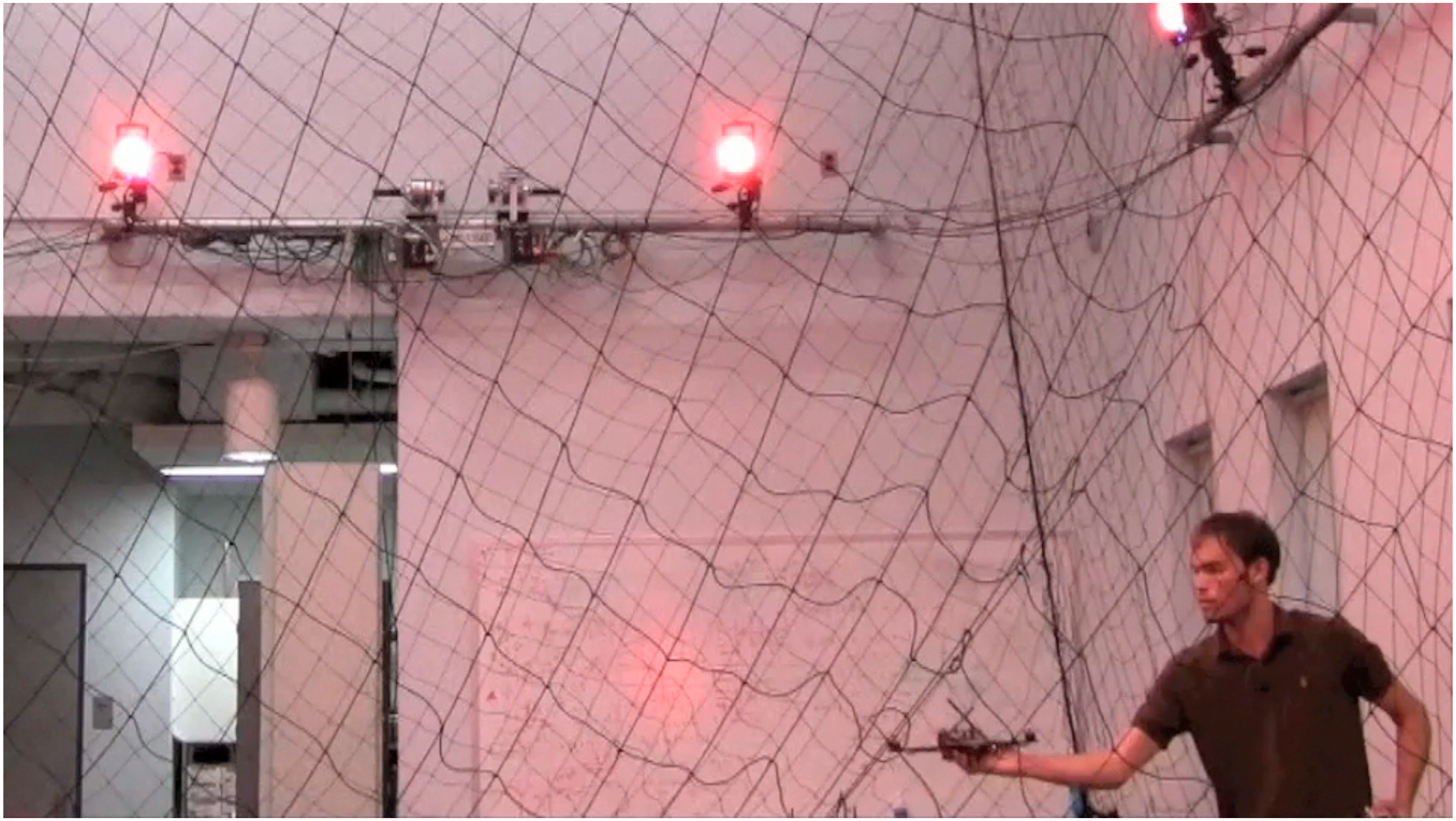
The position control system is a fourth order system

Want trajectories that can be differentiated four times

Minimum Snap Trajectory

$$x^\star(t) = \arg\min_{x(t)} \int_0^T \left(x^{(iv)}\right)^2 dt$$

Inner Attitude Control Loop

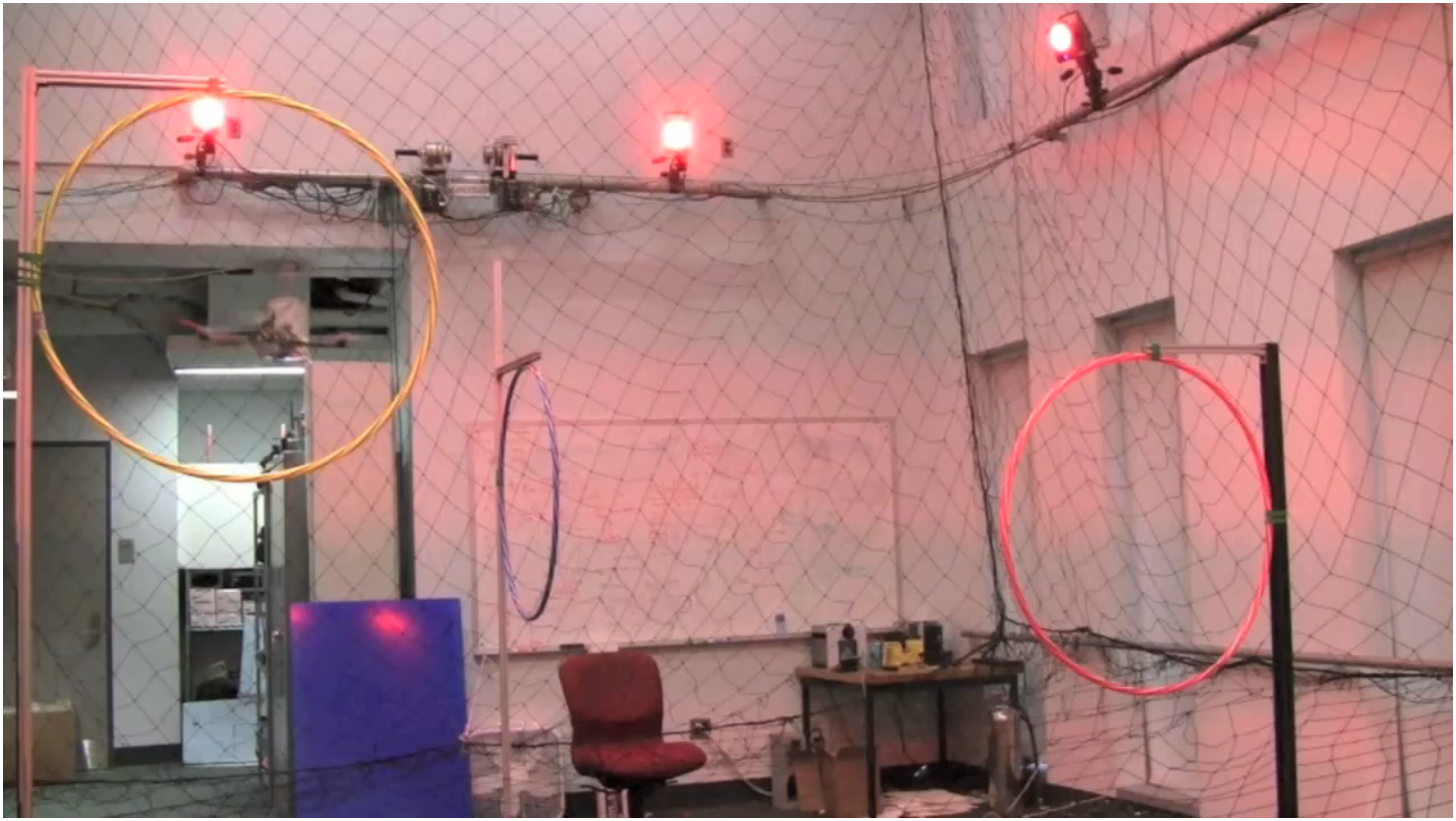


Daniel Mellinger, Nathan Michael, and Vijay Kumar. Trajectory Generation and Control for Precise Aggressive Maneuvers with Quadrotors. *International Journal of Robotics Research*, Apr. 2012.

Minimum Snap Trajectories

D. Mellinger and V. Kumar, "Minimum Snap Trajectory Generation and Control for Quadrotors," *Proc. IEEE International Conference on Robotics and Automation*. Shanghai, China, May, 2011.

Automated Synthesis of Trajectories



D. Mellinger and V. Kumar, "Minimum Snap Trajectory Generation and Control for Quadrotors," *Proc. IEEE International Conference on Robotics and Automation*. Shanghai, China, May, 2011.

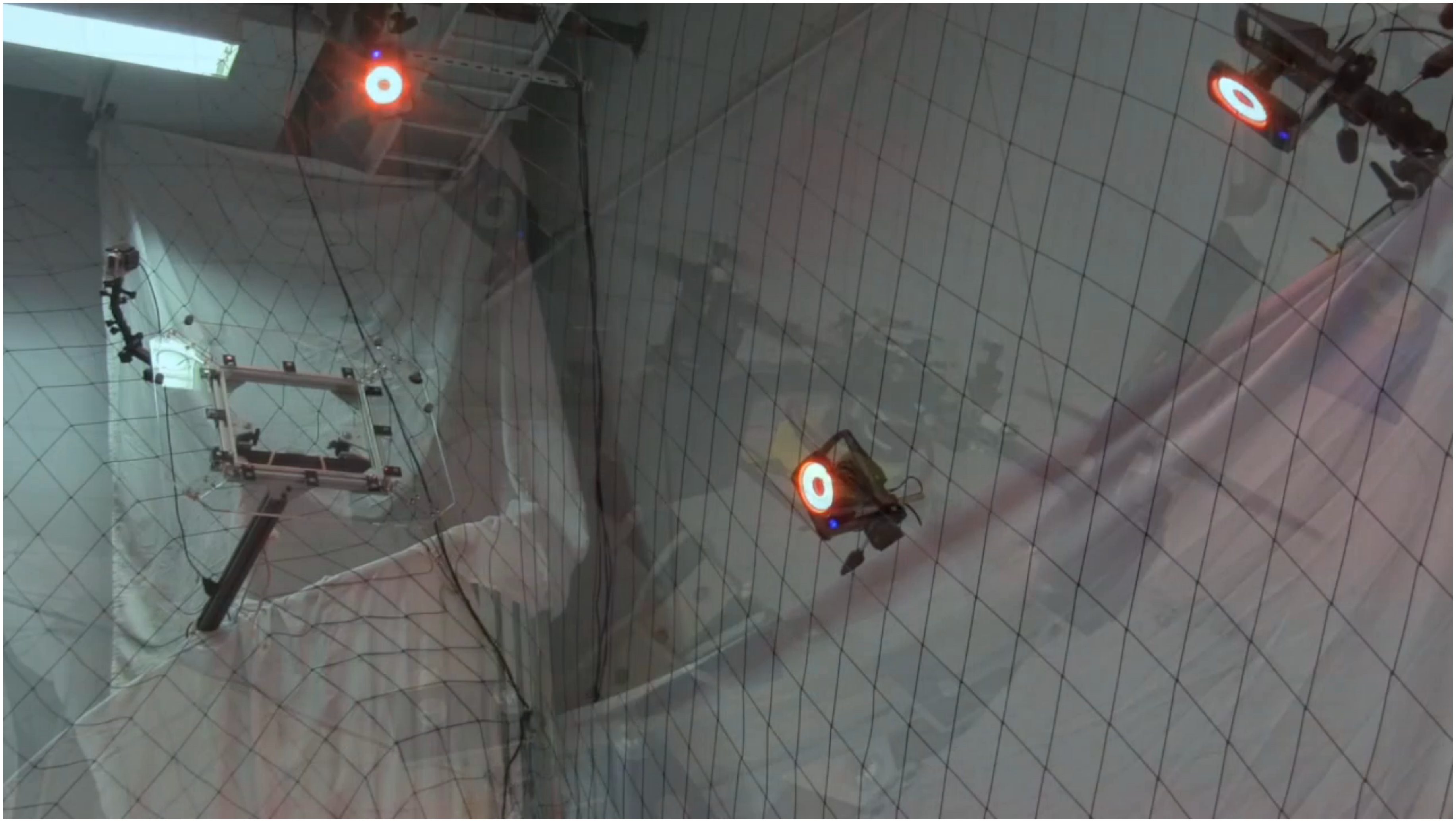
Penn Engineering

Aerial Grasping and Manipulation

Justin Thomas, Joe Polin, Koushil Sreenath, and Vijay Kumar, "Avian-inspired grasping for quadrotor micro UAVs," *ASME International Design Engineering Technical Conference* (IDETC), Portland, Oregon, August 2013.

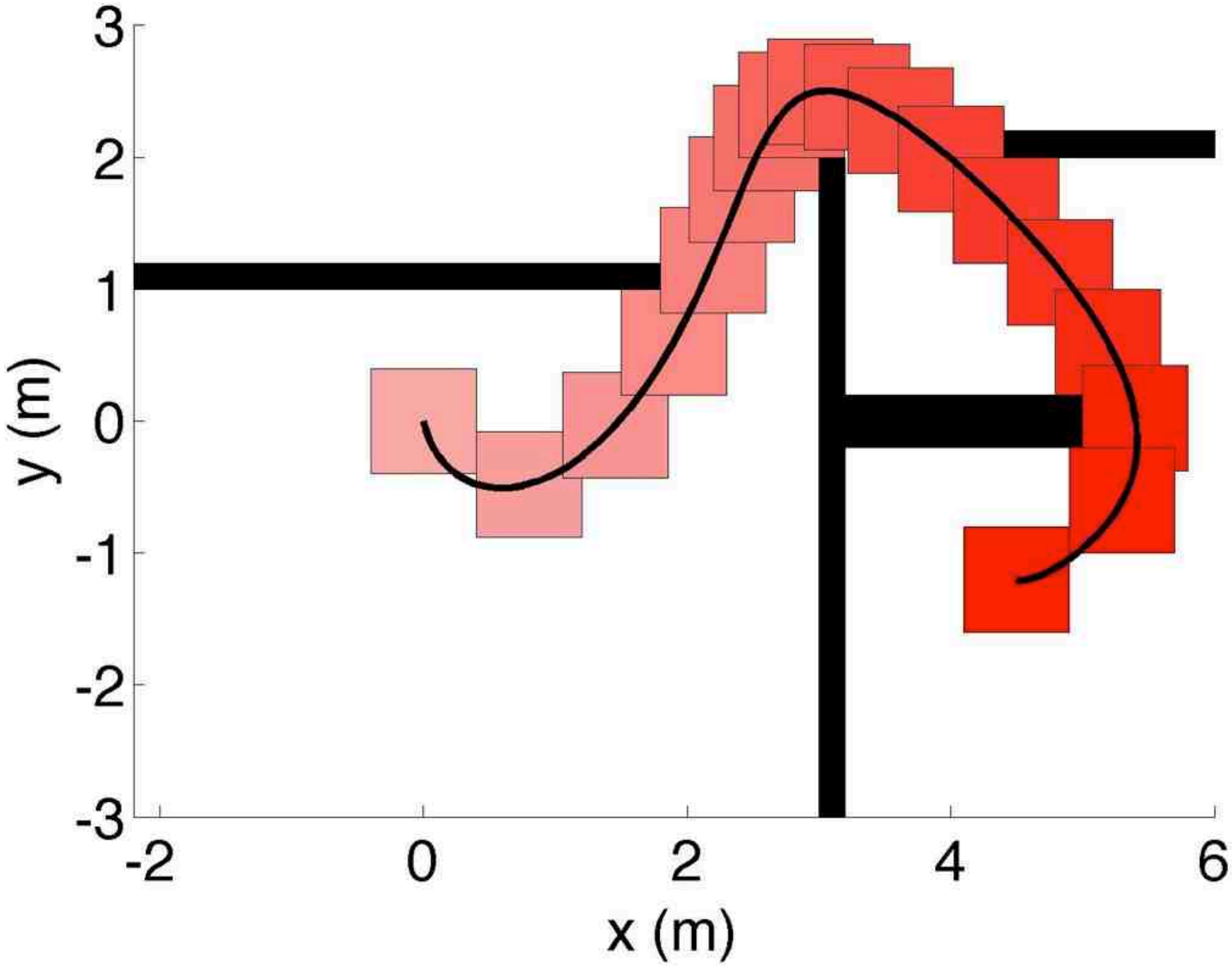
Penn Engineering

Perching



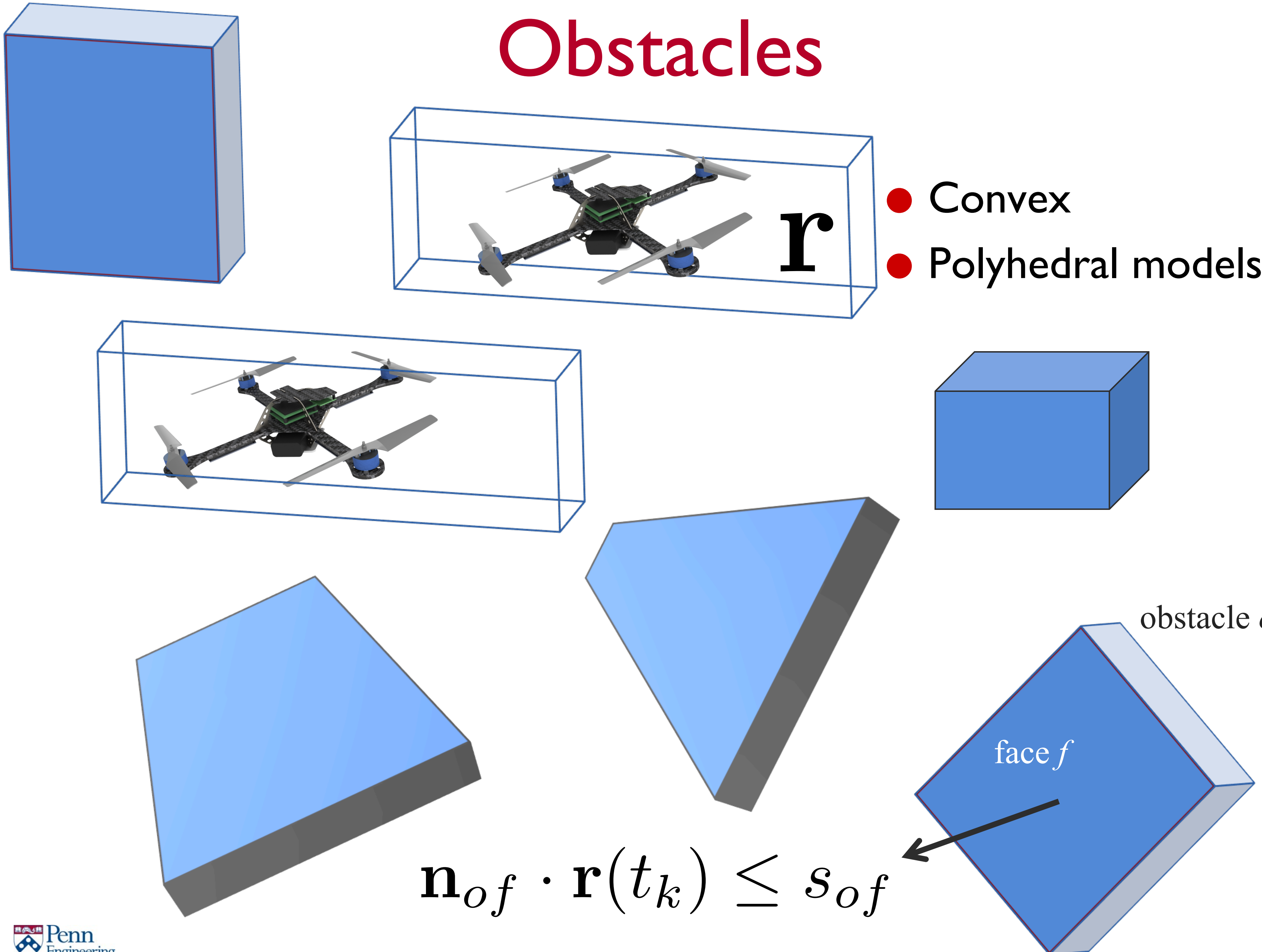
J. Thomas, G. Loianno, M. Pope, E. W. Hawkes, M. A. Estrada, H. Jiang, M. R. Cutkosky, and V. Kumar, "Planning and Control of Aggressive Maneuvers for Perching on Inclined and Vertical Surfaces," in *International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE)*, Boston MA, August 2015.

Min Snap Trajectory with Constraints



Obstacles

- Convex
- Polyhedral models

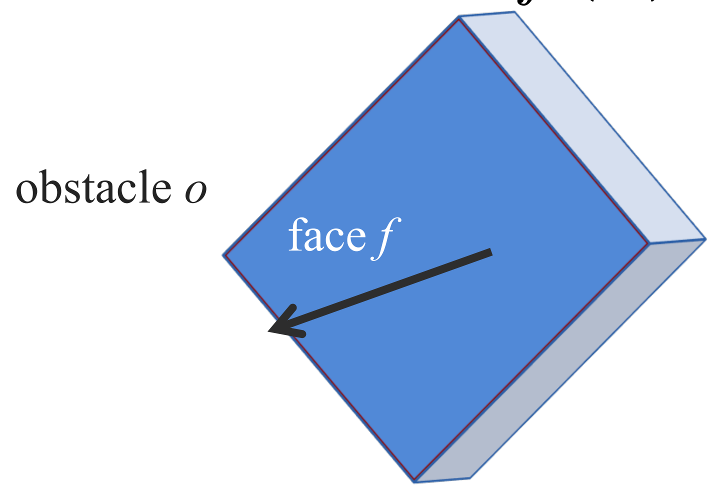


$$\mathbf{n}_{of} \cdot \mathbf{r}(t_k) \leq s_{of}$$

Integer Constraints for Obstacle Avoidance

$$\mathbf{n}_{of} \cdot \mathbf{r}(t_k) \leq s_{of} + M b_{ofk}, \; \forall f = 1, \ldots, n_f(o)$$

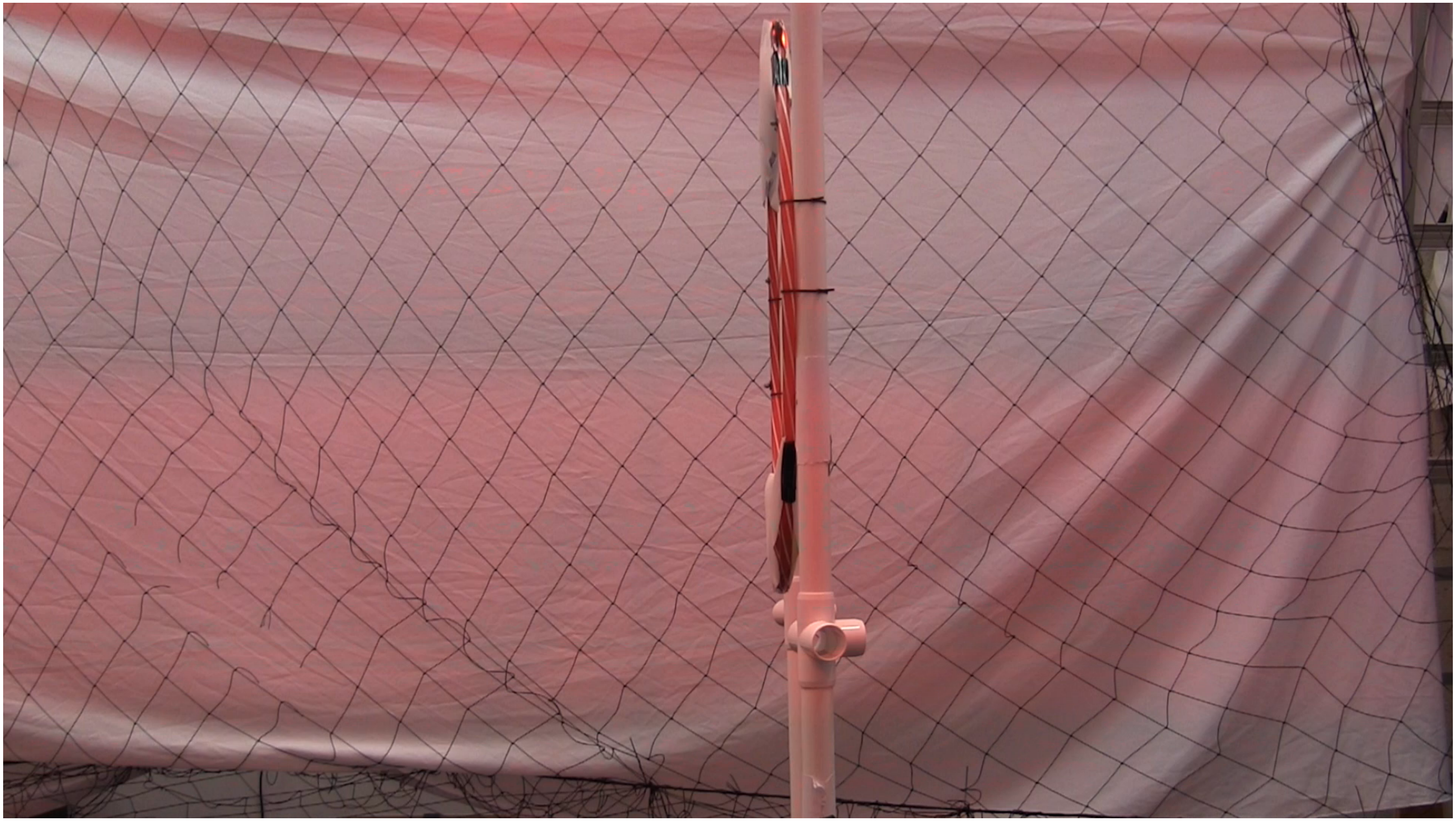
o	obstacle
n_f	number of faces
t_k	*kth* time instant
b_{ofk}	binary variable
M	large positive constant



$$\mathbf{n}_{of} \cdot \mathbf{r}(t_k) \leq s_{of}$$

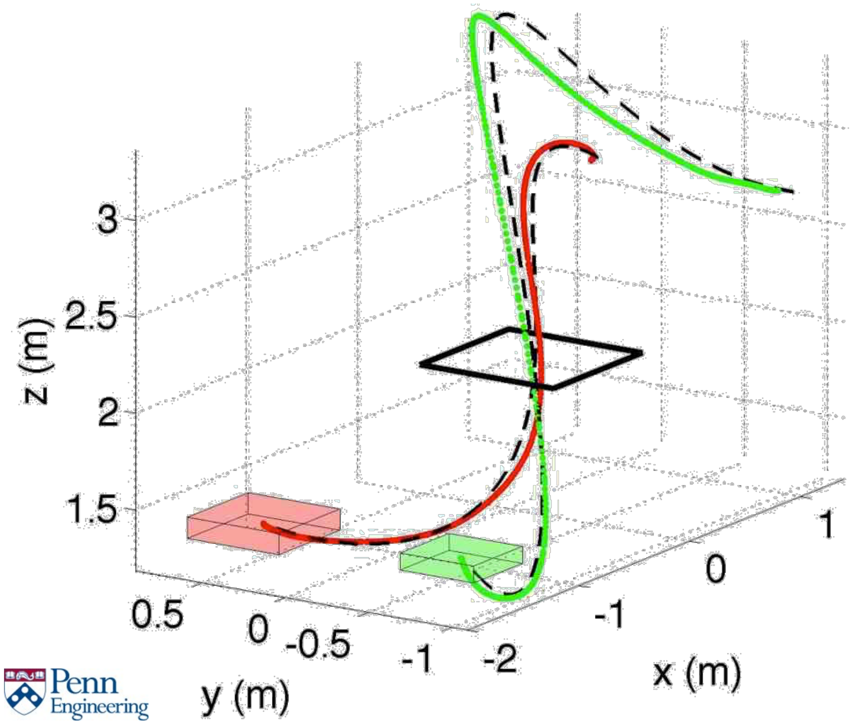
$$\sum_{f=1}^{n_f(o)} b_{ofk} \leq n_f(o) - 1$$

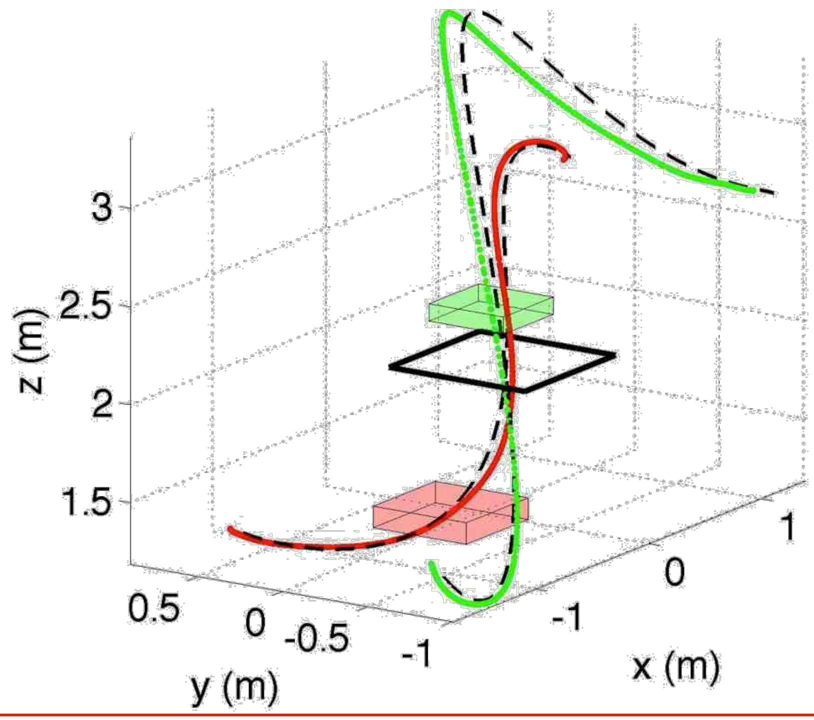
Transporting Suspended Payloads



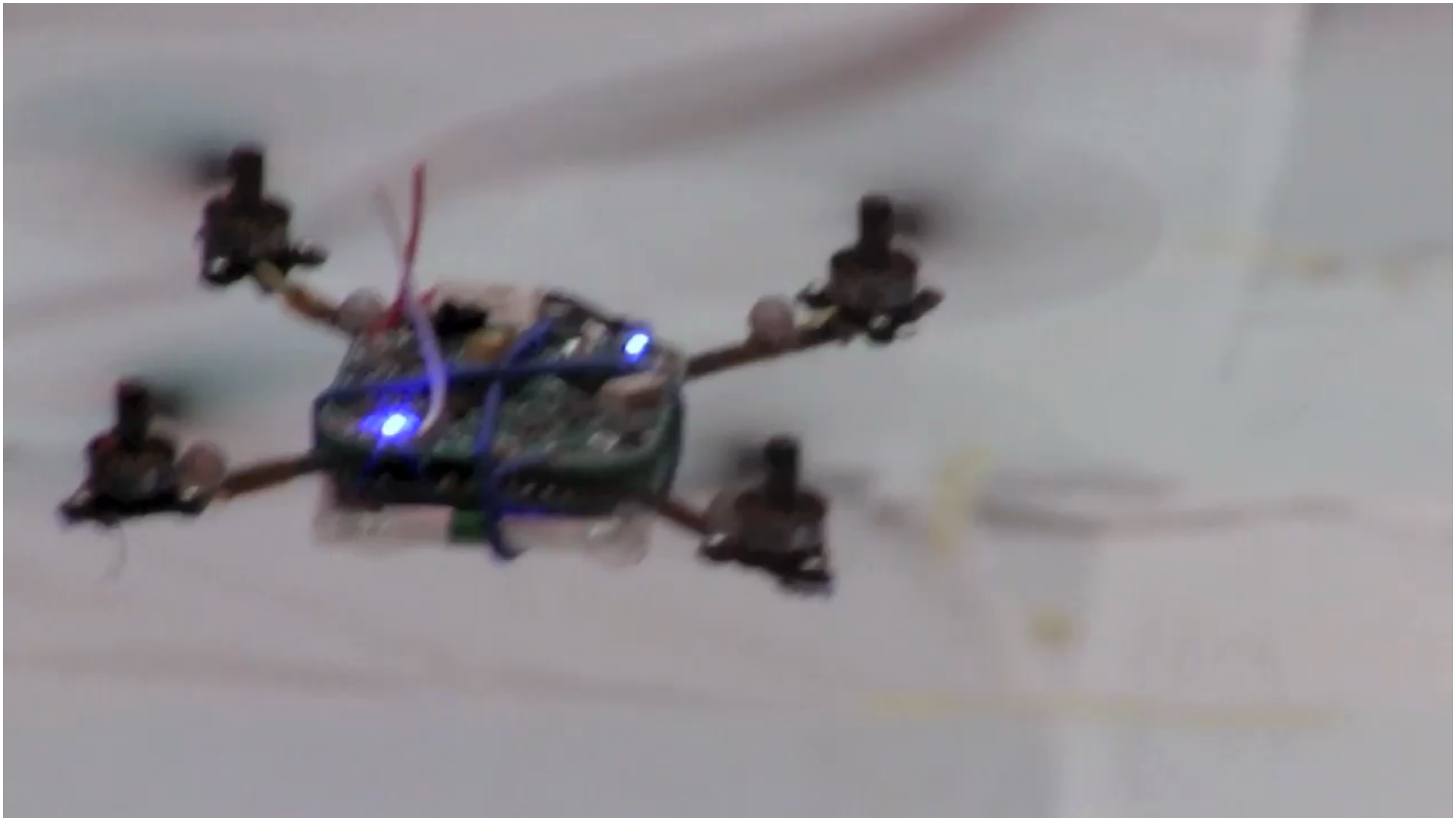
S. Tang and V. Kumar, "Mixed Integer Quadratic Program Trajectory Generation for a Quadrotor with a Cable-Suspended Payload," *in IEEE International Conference on Robotics and Automation*, May 2015.

Results





Penn Engineering



Aleksandr Kushleyev, Daniel Mellinger, Caitlin Powers, Vijay Kumar, “Towards a swarm of agile micro quadrotors,” *Autonomous Robots*, Vol. 35, No. 4, Pg. 287-300, 2013.

Minimum Velocity Trajectories from the Euler-Lagrange Equations

Penn Engineering

Minimum Velocity Trajectory

Find the function x(t) such that:

$$x^\star(t) = \underset{x(t)}{\operatorname{argmin}} \int_0^T \mathcal{L}\left(\dot{x}, x, t\right) dt$$

$$= \underset{x(t)}{\operatorname{argmin}} \int_0^T \dot{x}^2 dt$$

Euler-Lagrange equation:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{x}}\right) - \frac{\partial \mathcal{L}}{\partial x} = 0$$

Minimum Velocity Trajectory

Euler-Lagrange equation:

$$\boxed{\frac{d}{dt}}\boxed{\left(\frac{\partial \mathcal{L}}{\partial \dot{x}}\right)} - \boxed{\frac{\partial \mathcal{L}}{\partial x}} = 0$$

Cost function: $\mathcal{L}\left(\dot{x}, x, t\right) = \left(\dot{x}\right)^2$

Euler-Lagrange terms:

$$\left(\frac{\partial \mathcal{L}}{\partial x}\right) = 0$$

← No x appears in $\mathcal{L}$

$$\left(\frac{\partial \mathcal{L}}{\partial \dot{x}}\right) = 2\dot{x}$$

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{x}}\right) = \frac{d}{dt}(2\dot{x}) = 2\ddot{x}$$

Penn Engineering

Minimum Velocity Trajectory

Euler-Lagrange equation:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{x}}\right) - \frac{\partial \mathcal{L}}{\partial x} = 0$$

Euler-Lagrange terms: $2\ddot{x} - 0 = 0 \longrightarrow 2\ddot{x} = 0 \longrightarrow \ddot{x} = 0$

Integrate to get the velocity: $\dot{x} = c_1$

Integrate to get position: $x(t) = c_1 t + c_0$

Solving for Coefficients of Minimum Jerk Trajectories

Minimum Jerk Trajectory

$$x^\star(t) = \underset{x(t)}{\operatorname{argmin}} \int_0^T \mathcal{L}\left(\dddot{x}, \ddot{x}, \dot{x}, x, t\right) dt = \underset{x(t)}{\operatorname{argmin}} \int_0^T \dddot{x}^2 dt$$

We can solve the Euler-Lagrange equation:

$$\frac{\partial \mathcal{L}}{\partial x} - \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{x}}\right) + \frac{d^2}{dt^2}\left(\frac{\partial \mathcal{L}}{\partial \ddot{x}}\right) - \frac{d^3}{dt^3}\left(\frac{\partial \mathcal{L}}{\partial x^{(3)}}\right) = 0$$

to get the condition:

$$x^{(6)} = 0$$

Thus, we want a trajectory of the form:

$$x(t) = c_5 t^5 + c_4 t^4 + c_3 t^3 + c_2 t^2 + c_1 t + c_0$$

Penn Engineering

Solving for Coefficients

Boundary conditions:

	Position	Velocity	Acceleration
t = 0	a	0	0
t = T	b	0	0

Position constraints: $x(t) = c_5 t^5 + c_4 t^4 + c_3 t^3 + c_2 t^2 + c_1 t + c_0$

$$x(0) = c_0 = a$$

$$x(T) = c_5 (T)^5 + c_4 (T)^4 + c_3 (T)^3 + c_2 (T)^2 + c_1 (T) + c_0 = b$$

Penn Engineering

Solving for Coefficients

Position constraints in matrix form:

$$x(0) = c_0 = a$$

$$\underbrace{\underbrace{\begin{bmatrix}0 & 0 & 0 & 0 & 0 & 1\end{bmatrix}}_{1 \times 6}\underbrace{\begin{bmatrix}c_5\\ c_4\\ c_3\\ c_2\\ c_1\\ c_0\end{bmatrix}}_{6 \times 1}}_{1 \times 1} = a$$

Solving for Coefficients

Position constraints in matrix form:

$$x(0) = c_0 = a$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_5 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix} = a$$

Penn Engineering

Solving for Coefficients

Position constraints in matrix form:

$$x(T) = c_5(T)^5 + c_4(T)^4 + c_3(T)^3 + c_2(T)^2 + c_1(T) + c_0 = b$$

$$\begin{bmatrix} T^5 & T^4 & T^3 & T^2 & T & 1 \end{bmatrix} \begin{bmatrix} c_5 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix} = b$$

Solving for Coefficients

Boundary conditions:

	Position	Velocity	Acceleration
t = 0	a	0	0
t = T	b	0	0

Velocity constraints: $\dot{x}(t) = 5c_5t^4 + 4c_4t^3 + 3c_3t^2 + 2c_2t + c_1$

$$\dot{x}(0) = c_1 = 0$$

$$\dot{x}(T) = 5c_5(T)^4 + 4c_4(T)^3 + 3c_3(T)^2 + 2c_2(T) + c_1 = 0$$

Solving for Coefficients

Velocity constraints in matrix form:

$$\dot{x}(0) = c_1 = 0$$

$$\dot{x}(T) = 5c_5(T)^4 + 4c_4(T)^3 + 3c_3(T)^2 + 2c_2(T) + c_1 = 0$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} c_5 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix} = 0$$

$$\begin{bmatrix} 5T^4 & 4T^3 & 3T^2 & 2T & 1 & 0 \end{bmatrix} \begin{bmatrix} c_5 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix} = 0$$

Penn Engineering

Solving for Coefficients

Boundary conditions:

	Position	Velocity	Acceleration
t = 0	a	0	0
t = T	b	0	0

Acceleration constraints: $\ddot{x}(t) = 20c_5t^3 + 12c_4t^2 + 6c_3t^2 + 2c_2$

$$\ddot{x}(0) = 2c_2 = 0$$

$$\ddot{x}(T) = 20c_5(T)^3 + 12c_4(T)^2 + 6c_3(T)^2 + 2c_2 = 0$$

Penn Engineering

Solving for Coefficients

Acceleration constraints in matrix form:

$$\ddot{x}(0) = 2c_2 = 0$$

$$\ddot{x}(T) = 20c_5(T)^3 + 12c_4(T)^2 + 6c_3(T)^2 + 2c_2 = 0$$

$$\begin{bmatrix} 0 & 0 & 0 & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} c_5 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix} = 0$$

$$\begin{bmatrix} 20T^3 & 12T^2 & 6T & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} c_5 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix} = 0$$

Penn Engineering

Solving for Coefficients

Boundary conditions:

	Position	Velocity	Acceleration
t = 0	a	0	0
t = T	b	0	0

Combine constraints into one matrix expression:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ T^5 & T^4 & T^3 & T^2 & T & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 5T^4 & 4T^3 & 3T^2 & 2T & 1 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 20T^3 & 12T^2 & 6T & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} c_5 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix} = \begin{bmatrix} a \\ b \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Example 1: Find the Minimum Jerk Trajectory

Find the minimum jerk trajectory with boundary conditions:

	Position	Velocity	Acceleration
t = 0	a = 0	0	0
t = T = 1	b = 5	0	0

$$Ax = b$$

$$x = A^{-1}b$$

$$\underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 5 & 4 & 3 & 2 & 1 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 20 & 12 & 6 & 2 & 0 & 0 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} c_5 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} 0 \\ 5 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{b}$$

Penn Engineering

Example 1: Find the Minimum Jerk Trajectory

$$x = \begin{bmatrix} 30 \\ -75 \\ 50 \\ 0 \\ 0 \\ 0 \end{bmatrix} \longleftrightarrow \begin{bmatrix} c_5 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix}$$

$$x(t) = 30t^5 - 75t^4 + 50t^3$$

Example 1: Find the Minimum Jerk Trajectory

We can verify that this trajectory does in fact satisfy all boundary conditions:

	Position	Velocity	Acceleration
t = 0	a = 0 ✓	0	0
t = T = 1	b = 5 ✓	0	0

$$x(t) = 30t^5 - 75t^4 + 50t^3$$

$$x(0) = 0$$

$$x(1) = 30(1)^5 - 75(1)^4 + 50(1)^3 = 5$$

Penn Engineering

Example 1: Find the Minimum Jerk Trajectory

We can verify that this trajectory does in fact satisfy all boundary conditions:

	Position	Velocity	Acceleration
t = 0	a = 0 ✓	0 ✓	0
t = T = 1	b = 5 ✓	0 ✓	0

$$\dot{x}(t) = 150t^4 - 300t^3 + 150t^2$$

$$\dot{x}(0) = 0$$

$$\dot{x}(1) = 150 - 300 + 150 = 0$$

Penn Engineering

Example 1: Find the Minimum Jerk Trajectory

We can verify that this trajectory does in fact satisfy all boundary conditions:

	Position	Velocity	Acceleration
t = 0	a = 0 ✓	0 ✓	0 ✓
t = T = 1	b = 5 ✓	0 ✓	0 ✓

$$\ddot{x}(t) = 600t^3 - 900t^2 + 300t$$

$$\ddot{x}(0) = 0$$

$$\ddot{x}(1) = 600 - 900 + 300 = 0$$

Find the Minimum Jerk Trajectory

The order of coefficients in the matrix of unknowns matters.

$$\begin{bmatrix} T^5 & T^4 & T^3 & T^2 & T & 1 \end{bmatrix} \begin{bmatrix} c_5 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix} = b$$

Find the Minimum Jerk Trajectory

The order of coefficients in the matrix of unknowns matters.

$$\begin{bmatrix} 1 & T & T^2 & T^3 & T^4 & T^5 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} = b$$

Find the Minimum Jerk Trajectory

The order of coefficients in the matrix of unknowns matters.

$$\begin{bmatrix} T^4 & T & T^2 & T^5 & T^3 & 1 \end{bmatrix} \begin{bmatrix} c_4 \\ c_1 \\ c_2 \\ c_5 \\ c_3 \\ c_0 \end{bmatrix} = b$$

Minimum Velocity Trajectories

Minimum Velocity Trajectory

Why is the minimum velocity curve also the shortest distance curve?

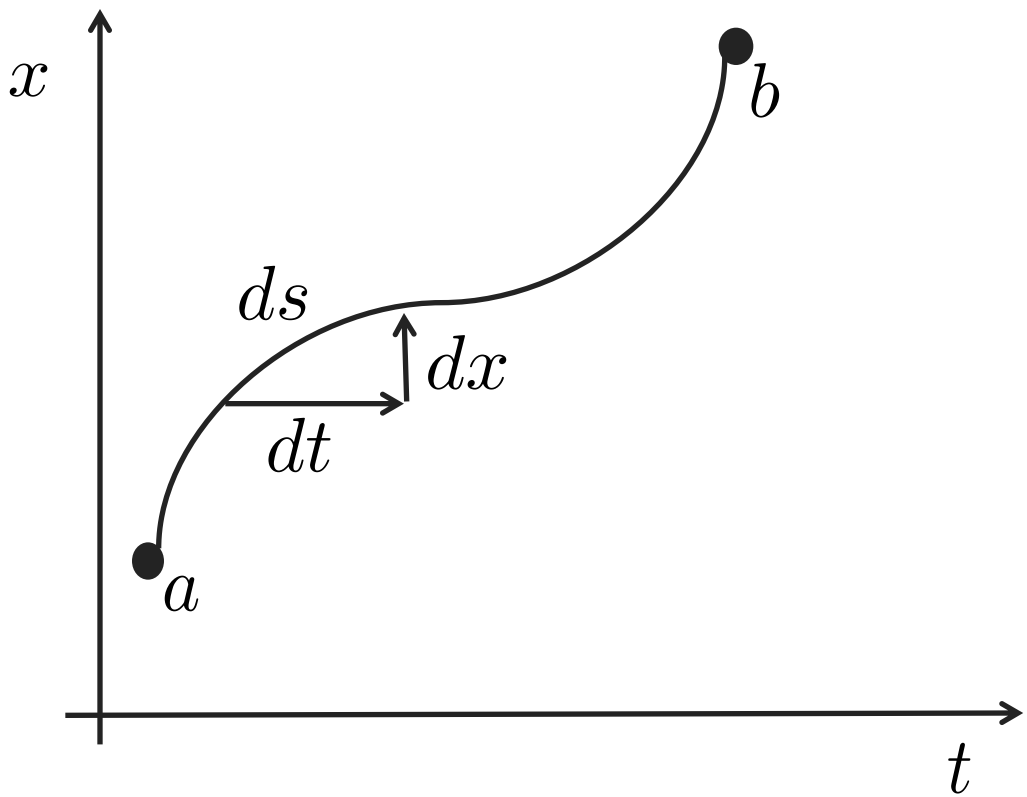
To get the minimum velocity trajectory, we solved:

$$x^{\star}(t) = \underset{x(t)}{\operatorname{argmin}} \int_0^T \dot{x}^2 dt$$

From the Euler-Lagrange equations, the solution is:

$$x(t) = c_1 t + c_0$$

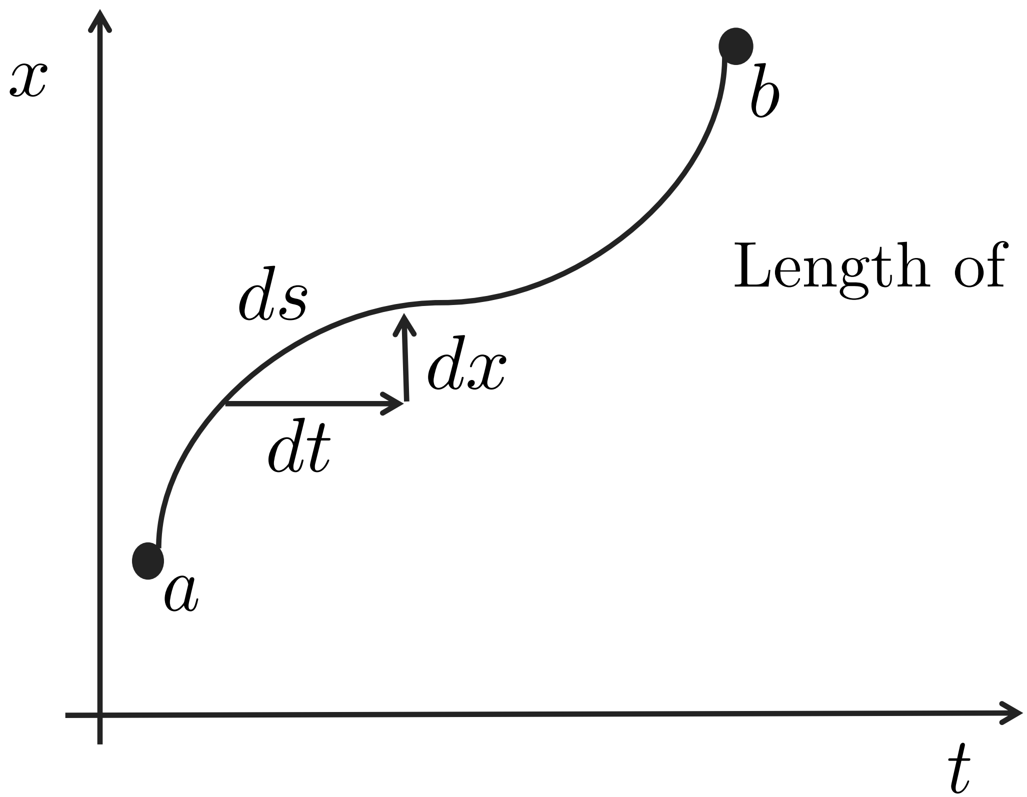
Minimum Distance Trajectory



$$
\begin{aligned}
ds &= \sqrt{dt^2 + dx^2} \\
&= \sqrt{1 + \left(\frac{dx}{dt}\right)^2}\, dt \\
&= \sqrt{1 + \dot{x}^2}\, dt
\end{aligned}
$$

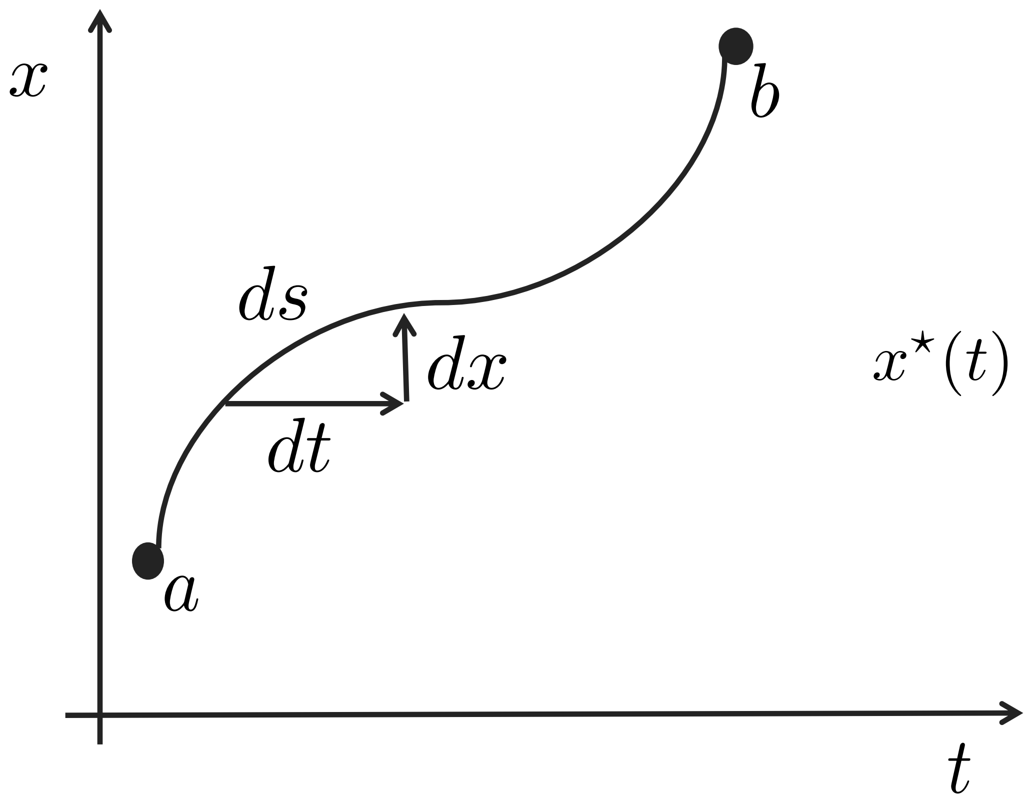
Penn Engineering

Minimum Distance Trajectory



$$\text{Length of curve} = \int ds$$
$$= \int_0^T \sqrt{1 + \dot{x}^2} dt$$

Minimum Distance Trajectory



$$x^\star(t) = \underset{x(t)}{\operatorname{argmin}} \int_0^T \sqrt{1 + \dot{x}^2}\,dt$$

Minimum Distance Trajectory

$$x^\star(t) = \underset{x(t)}{\operatorname{argmin}} \int_0^T \sqrt{1+\dot{x}^2}\,dt$$

$$\mathcal{L}(\dot{x}, x, t) = \sqrt{1+\dot{x}^2}$$

Euler-Lagrange equation:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{x}}\right) - \frac{\partial \mathcal{L}}{\partial x} = 0$$

Minimum Distance Trajectory

Euler-Lagrange equation:

$$\boxed{\frac{d}{dt}}\boxed{\left(\frac{\partial \mathcal{L}}{\partial \dot{x}}\right)} - \boxed{\frac{\partial \mathcal{L}}{\partial x}} = 0$$

Cost-function: $\mathcal{L}(\dot{x}, x, t) = \sqrt{1 + \dot{x}^2}$

Euler-Lagrange terms:

$$\left(\frac{\partial \mathcal{L}}{\partial x}\right) = 0$$ ← No x appears in $\mathcal{L}$

$$\left(\frac{\partial \mathcal{L}}{\partial \dot{x}}\right) = \frac{\dot{x}}{\sqrt{1 + \dot{x}^2}}$$

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{x}}\right) = \frac{d}{dt}\left(\frac{\dot{x}}{\sqrt{1 + \dot{x}^2}}\right)$$

Minimum Distance Trajectory

Euler-Lagrange equation:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{x}}\right) - \frac{\partial \mathcal{L}}{\partial x} = 0$$

Euler-Lagrange terms: $\frac{d}{dt}\left(\frac{\dot{x}}{\sqrt{1+\dot{x}^2}}\right) = 0$

Integrate to get velocity: $\frac{\dot{x}}{\sqrt{1+\dot{x}^2}} = K \longrightarrow \dot{x} = \sqrt{\frac{K^2}{1-K^2}} = c_1$

Integrate to get position: $x(t) = c_1 t + c_0 \longleftarrow$ Same as minimum velocity solution

Linearization of Quadrotor Equations of Motion

Quadrotor Equations of Motion

Linear momentum balance:

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

Angular momentum balance:

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

Quadrotor Equations of Motion

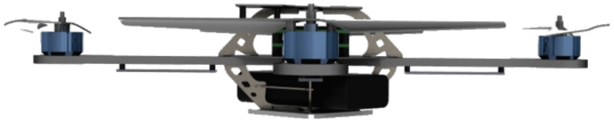
Linear momentum balance:

$$m\ddot{\mathbf{r}} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ u_1 \end{bmatrix}$$

Angular momentum balance:

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} u_{2x} \\ u_{2y} \\ u_{2z} \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

Equilibrium Hover Configuration



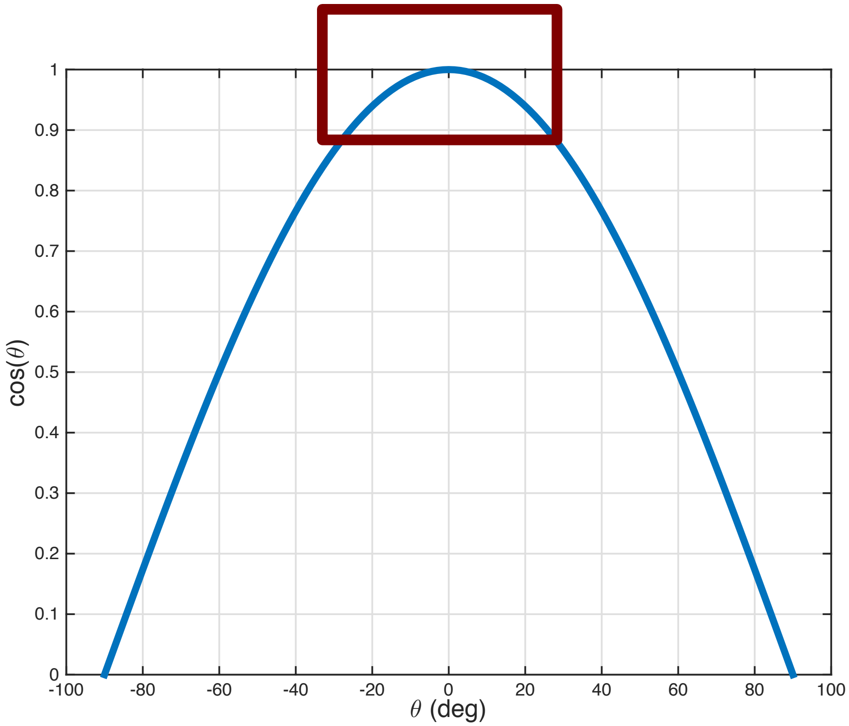
$$\mathbf{r} = \mathbf{r}_0, \theta = \phi = 0, \psi = \psi_0$$

$$\dot{\mathbf{r}} = 0, \dot{\theta} = \dot{\phi} = \dot{\psi} = 0$$

Penn Engineering

Linearization of Trigonometric Functions

What is the value of $\cos(\theta)$ near $\theta = 0$?



Can be approximated with the Taylor Series:

$\cos(\theta)$

$$\approx \cos(\theta)|_{\theta=0} + \frac{d\cos(\theta)}{d\theta}|_{\theta=0}\theta$$

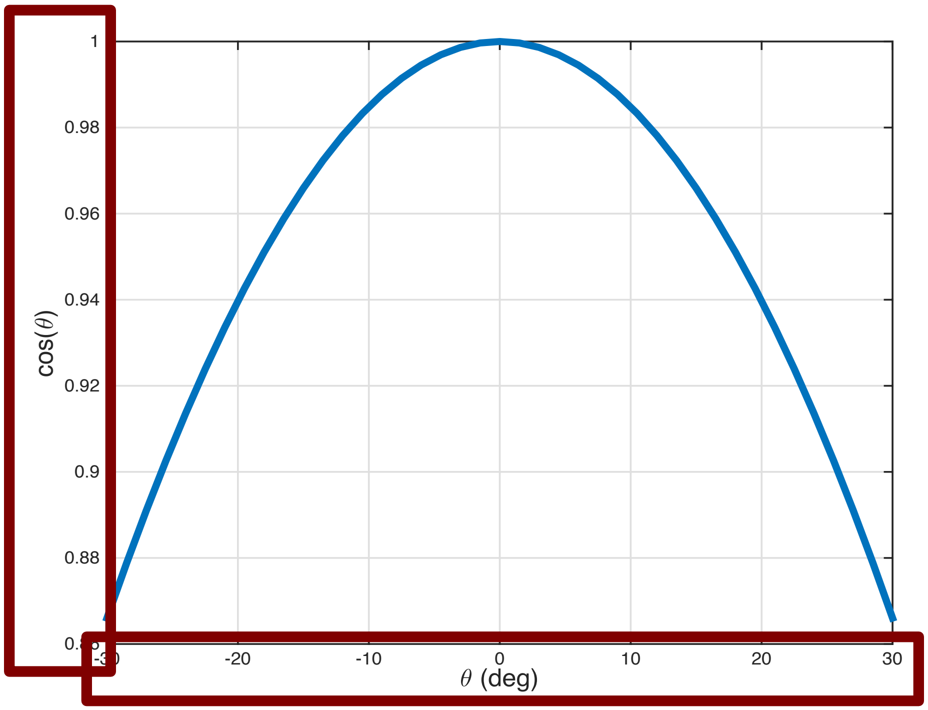
$$+ \text{ higher order terms}$$

$$\approx 1 - \sin(\theta)|_{\theta=0}\theta$$

$$\approx 1$$

Linearization of Trigonometric Functions

What is the value of $\cos(\theta)$ near $\theta = 0$?



Can be approximated with the Taylor Series:

$$\cos(\theta)$$

$$\approx \cos(\theta)|_{\theta=0} + \frac{d\cos(\theta)}{d\theta}|_{\theta=0}\theta$$

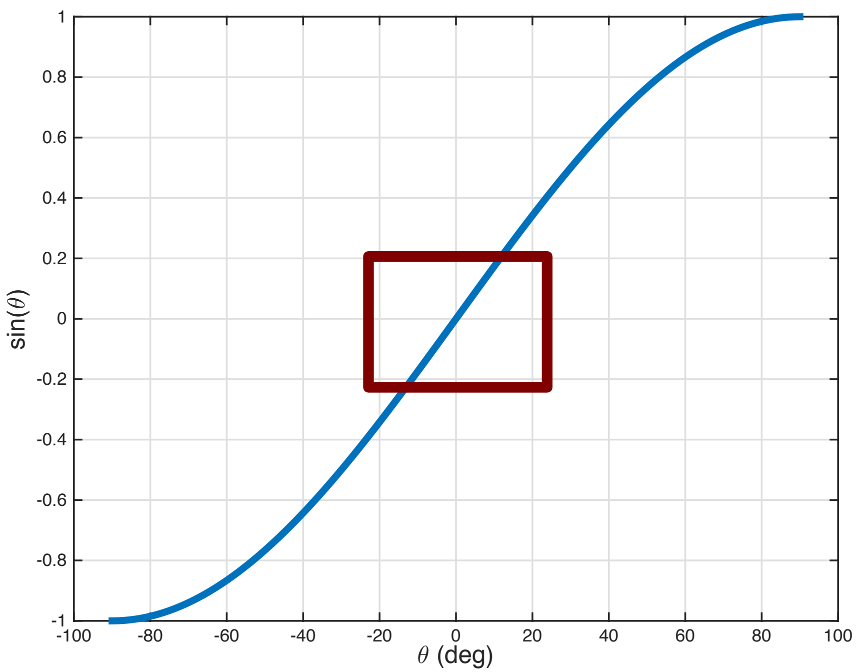
$$+ \text{ higher order terms}$$

$$\approx 1 - \sin(\theta)|_{\theta=0}\theta$$

$$\approx 1$$

Linearization of Trigonometric Functions

What is the value of $\sin(\theta)$ near $\theta = 0$?



Can be approximated with the Taylor Series:

$$\sin(\theta)$$

$$\approx \sin(\theta)|_{\theta=0} + \frac{d\sin(\theta)}{d\theta}|_{\theta=0}\theta$$

$$+ \text{higher order terms}$$

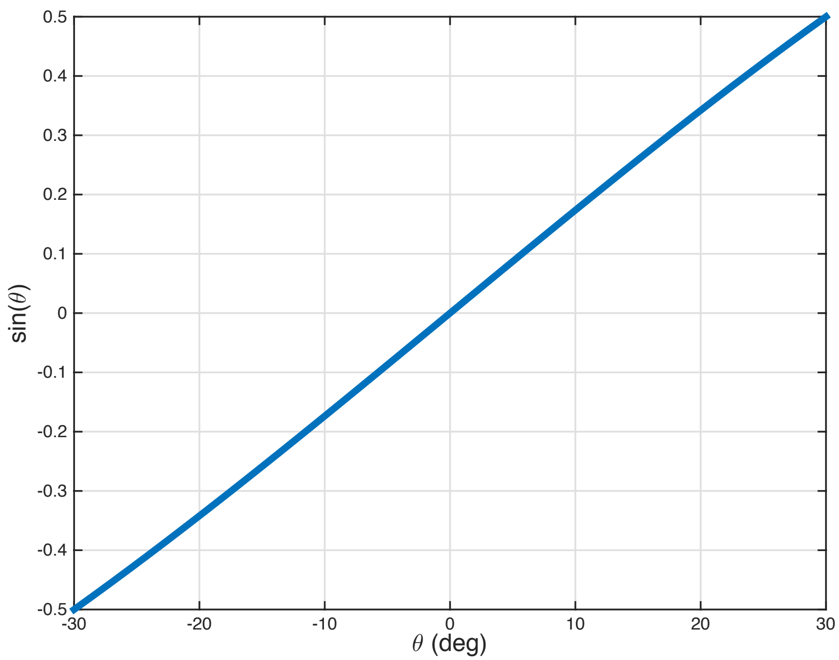
$$\approx 0 + \cos(\theta)|_{\theta=0}\theta$$

$$\approx \theta$$

Penn Engineering

Linearization of Trigonometric Functions

What is the value of $\sin(\theta)$ near $\theta = 0$?



sine function looks linear around $\theta = 0$

Can be approximated with the Taylor Series:

$$\sin(\theta)$$

$$\approx \sin(\theta)|_{\theta=0} + \frac{d\sin(\theta)}{d\theta}|_{\theta=0}\theta$$

$$+ \text{higher order terms}$$

$$\approx 0 + \cos(\theta)|_{\theta=0}\theta$$

$$\approx \theta$$

Penn Engineering

Linearized Equations of Motion

What are the equations of motion of the quadrotor when it is near the equilibrium hover configuration?

$$\mathbf{r} \approx \mathbf{r}_0, \theta \approx \phi \approx 0, \psi \approx \psi_0$$

$$\dot{\mathbf{r}} \approx 0, \dot{\theta} \approx \dot{\phi} \approx \dot{\psi} \approx 0$$

Linear Momentum Equation Near Hover

$$m\ddot{\mathbf{r}} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ u_1 \end{bmatrix}$$

Linear Momentum Equation Near Hover

$$m\ddot{\mathbf{r}} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + \begin{bmatrix} c\psi c\theta - s\phi s\psi s\theta & -c\phi s\psi & c\psi s\theta + c\theta s\phi s\psi \\ c\theta s\psi + c\psi s\theta s\phi & c\phi c\psi & s\psi s\theta - c\psi c\theta s\phi \\ -c\phi s\theta & s\phi & c\phi c\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ u_1 \end{bmatrix}$$

Linear Momentum Equation Near Hover

$$m\ddot{x} = (c\psi s\theta + c\theta s\phi s\psi)\, u_1$$
$$m\ddot{y} = (s\psi s\theta - c\psi c\theta s\phi)\, u_1$$
$$m\ddot{z} = -mg + (c\phi c\theta)\, u_1$$

Substituting in the approximation:

$$\sin(\theta) \approx \theta, \sin(\phi) \approx \phi, \cos(\theta) \approx \cos(\phi) \approx 1$$

Linear Momentum Equation Near Hover

$$m\ddot{x} = (\theta c\psi + \phi s\psi)\, u_1$$
$$m\ddot{y} = (\theta s\psi - \phi c\psi)\, u_1$$
$$m\ddot{z} = -mg + u_1$$

The second derivative of position is proportional to u_1 !

Angular Rates Near Hover

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

Angular Rates Near Hover

$$p = \dot{\phi} c\theta - \dot{\psi} c\phi s\theta$$

$$q = \dot{\theta} + \dot{\psi} s\phi$$

$$r = \dot{\phi} s\theta + \dot{\psi} c\phi c\theta$$

Substituting in the approximation:

$$\sin(\theta) \approx \theta, \sin(\phi) \approx \phi, \cos(\theta) \approx \cos(\phi) \approx 1$$

Penn Engineering

Angular Rates Near Hover

$$p = \dot{\phi} - \dot{\psi}\theta$$

$$q = \dot{\theta} + \dot{\psi}\phi$$

$$r = \dot{\phi}\theta + \dot{\psi}$$

Substituting in the approximation:

$$\dot{\psi}\theta \approx \dot{\psi}\phi \approx \dot{\phi}\theta \approx 0$$

Higher order terms: Product of two terms around 0 is approximately 0.

Angular Rates Near Hover

$$p = \dot{\phi}$$

$$q = \dot{\theta}$$

$$r = \dot{\psi}$$

Angular Momentum Equation Near Hover

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} u_{2x} \\ u_{2y} \\ u_{2z} \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

Substituting in the approximation:

$$I_{xy} \approx I_{yx} \approx I_{xz} \approx I_{zx} \approx I_{yz} \approx I_{zy} \approx 0$$

Angular Momentum Equation Near Hover

$$\begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} =$$

$$\begin{bmatrix} u_{2x} \\ u_{2y} \\ u_{2z} \end{bmatrix} - \begin{bmatrix} 0 & r & -q \\ -r & 0 & p \\ q & -p & 0 \end{bmatrix} \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

Penn Engineering

Angular Momentum Equation Near Hover

$$I_{xx}\dot{p} = u_{2x} - I_{yy}qr + I_{zz}qr$$
$$I_{yy}\dot{q} = u_{2y} + I_{xx}pr - I_{zz}pr$$
$$I_{zz}\dot{r} = u_{2z} - I_{xx}pq + I_{yy}pq$$

Substituting in the approximation:

$$qr \approx pr \approx pq$$
$$\approx \dot{\theta}\dot{\psi} \approx \dot{\phi}\dot{\psi} \approx \dot{\phi}\dot{\theta} \approx 0$$

Higher order terms: Product of two terms around 0 is approximately 0.

Angular Momentum Equation Near Hover

$$I_{xx}\dot{p} = u_{2x}$$

$$I_{yy}\dot{q} = u_{2y}$$

$$I_{zz}\dot{r} = u_{2z}$$

Substituting in the approximation:

$$p \approx \dot{\phi}$$

$$q \approx \dot{\theta}$$

$$r \approx \dot{\psi}$$

Angular Momentum Equation Near Hover

$$\ddot{\phi} = \frac{u_{2x}}{I_{xx}}$$

$$\ddot{\theta} = \frac{u_{2y}}{I_{yy}}$$

$$\ddot{\psi} = \frac{u_{2z}}{I_{zz}}$$

Penn Engineering

Equations of Motion

Recall the linearized linear momentum equation:

$$m\ddot{x} = (\theta c\psi + \phi s\psi)\, u_1$$

Differentiating the equation:

$$m\dddot{x} = (\theta c\psi + \phi s\psi)\, \dot{u}_1 + \left(\dot{\theta} c\psi - \theta s\psi\dot{\psi} + \dot{\phi} s\psi + \phi c\psi\dot{\psi}\right) u_1$$

Differentiating again:

$$m\ddddot{x} = (\theta c\psi + \phi s\psi)\, \ddot{u}_1 + 2\left(\dot{\theta} c\psi - \theta s\psi\dot{\psi} + \dot{\phi} s\psi + \phi c\psi\dot{\psi}\right) \dot{u}_1 +$$
$$\left(\ddot{\theta} c\psi - \dot{\theta} s\psi\dot{\psi} - \theta s\psi\ddot{\psi} - \theta c\psi\dot{\psi}^2 + \ddot{\phi} s\psi + \dot{\phi} c\psi\dot{\psi} + \phi c\psi\ddot{\psi} - \phi c\psi\dot{\psi}^2\right) u_1$$

Equations of Motion

Substituting in the approximation:

$$\ddot{\phi} = \frac{u_{2x}}{I_{xx}}, \ddot{\theta} = \frac{u_{2y}}{I_{yy}}, \ddot{\psi} = \frac{u_{2z}}{I_{zz}}$$

The linear momentum equation becomes:

$$m\ddddot{x} = \ldots + \left(\frac{u_{2y}}{I_{yy}}c\psi + \frac{u_{2z}}{I_{zz}}\theta(c\psi - s\psi) + \frac{u_{2x}}{I_{xx}}s\psi\right)u_1$$

The fourth derivative of position is proportional to u_2 !