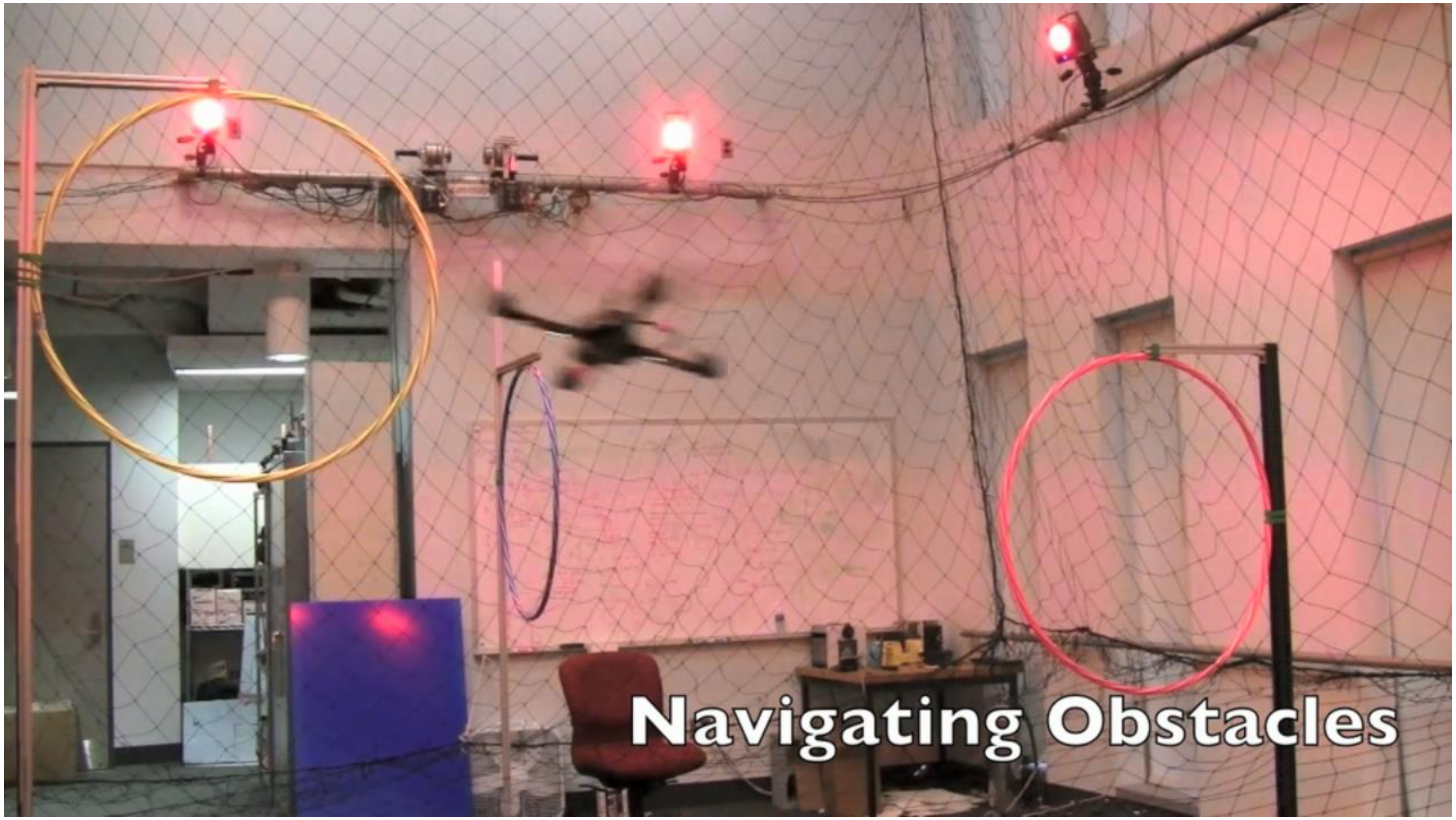
MEAM 620

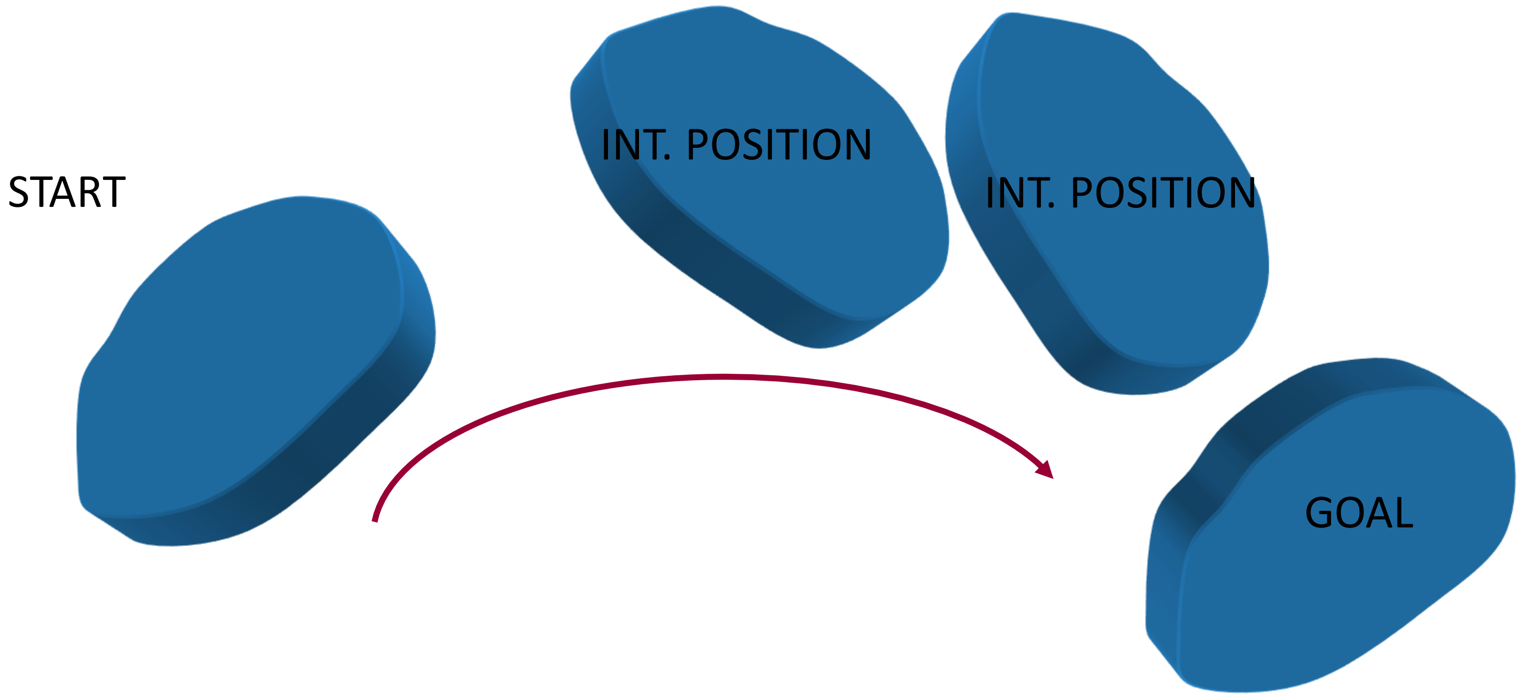
TIME, MOTION, AND TRAJECTORIES

Penn Engineering | GRASP Laboratory
General Robotics, Automation, Sensing & Perception Lab

Aggressive Trajectories



Smooth Trajectories



General Problem

Start and goal states

- Position, orientation, velocity, etc.

Waypoints

- Position, orientation, velocity, etc.

Smoothness criterion

- Generally means minimizing rate of change of “input”

Order of the system (n)

- Order of the system determines the input
- Boundary conditions on $(n-1)^{th}$ order and lower derivatives

Calculus of Variations

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

Function　　　　Functional

Examples

- Shortest distance (geometry)

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

- Fermat's principle (optics)

$$x^*(t) = \underset{x(t)}{\operatorname{argmin}} \int_0^T 1dt$$

- Principle of least action (mechanics)

$$x^*(t) = \underset{x(t)}{\operatorname{argmin}} \int_0^T T(\dot{x}, x, t) - V(\dot{x}, x, t)dt$$

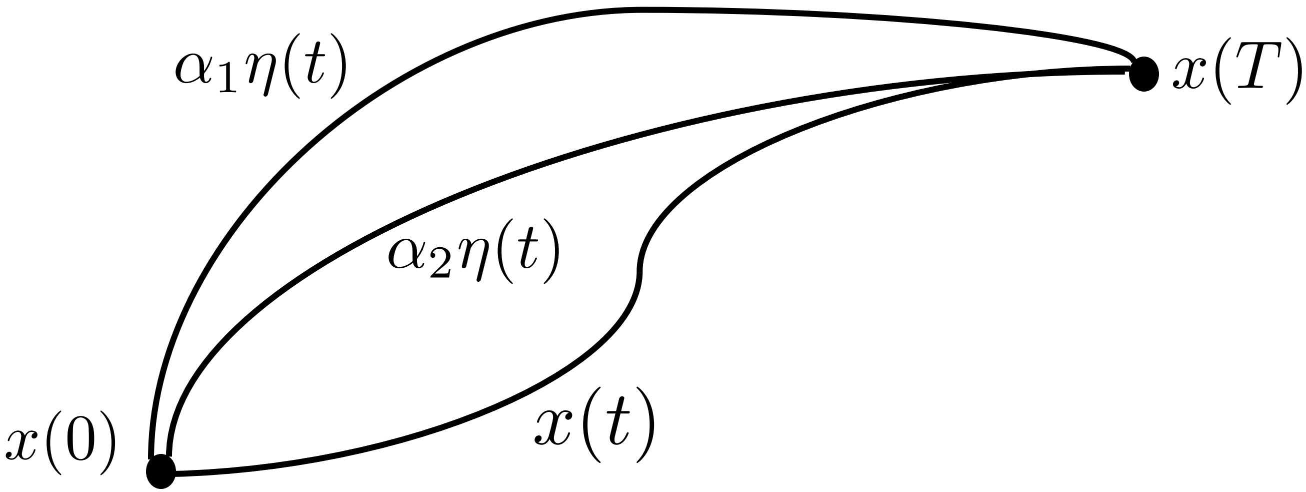
Calculus of Variations

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

Assume $x(t)$ is the optimal function

Any other path from $x(0)$ to $x(T)$ can be written as

$x(t) + \alpha\, \eta(t)$ s.t. $\eta(0) = \eta(T) = 0$



Calculus of Variations

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

Since $x(t)$ is the optimal function it must be that

$$\left.\frac{d}{d\alpha}\right|_{\alpha=0} \int_0^T L(\dot{x} + \alpha\dot{\eta}(t), x(t) + \alpha\eta(t), t)dt = 0$$

Euler Lagrange Equation

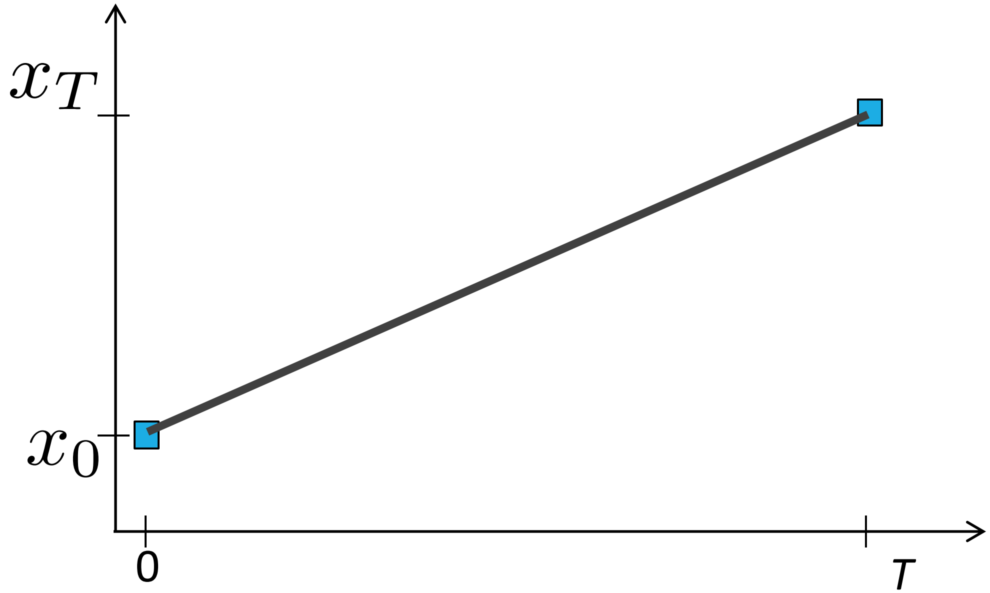
- Necessary condition satisfied by the optimal function
- $\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} = 0$

Smooth Trajectories ($n = 1$)

$\dot{x} = u$

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

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



Smooth Trajectories $(n = 1)$

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

$$L(\dot{x}, x, t) = \dot{x}^2$$

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

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

- Shortest paths are straight lines

Smooth Trajectories (n)

$x^{(n)} = u$

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

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

$\dot{x}(0) = v_0, \dot{x}(T) = v_T$

$\vdots$

$x^{(n)}(0) = x_0^{(n)}, x^{(n)}(T) = x_T^{(n)}$

Euler-Lagrange Equation

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

Euler Lagrange Equation

- Necessary condition satisfied by the optimal function
- $\frac{\partial L}{\partial x} - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) + \frac{d^2}{dt^2}\left(\frac{\partial L}{\partial \ddot{x}}\right) - \cdots + (-1)^n \frac{d^n}{dt^n}\left(\frac{\partial L}{\partial x^{(n)}}\right) = 0$

Smooth Trajectories

$n = 1$, shortest distance (minimum velocity)

$n = 2$, minimum acceleration

$n = 3$, minimum jerk

$n = 4$, minimum snap

$n = 5$, minimum crackle

$n = 6$, minimum pop

$L = \left(x^{(n)}\right)^2$ leads to $x(t) = c_0 + c_1 t + \cdots + c_{2n-1} t^{2n-1}$

Extensions to Multiple Variables

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

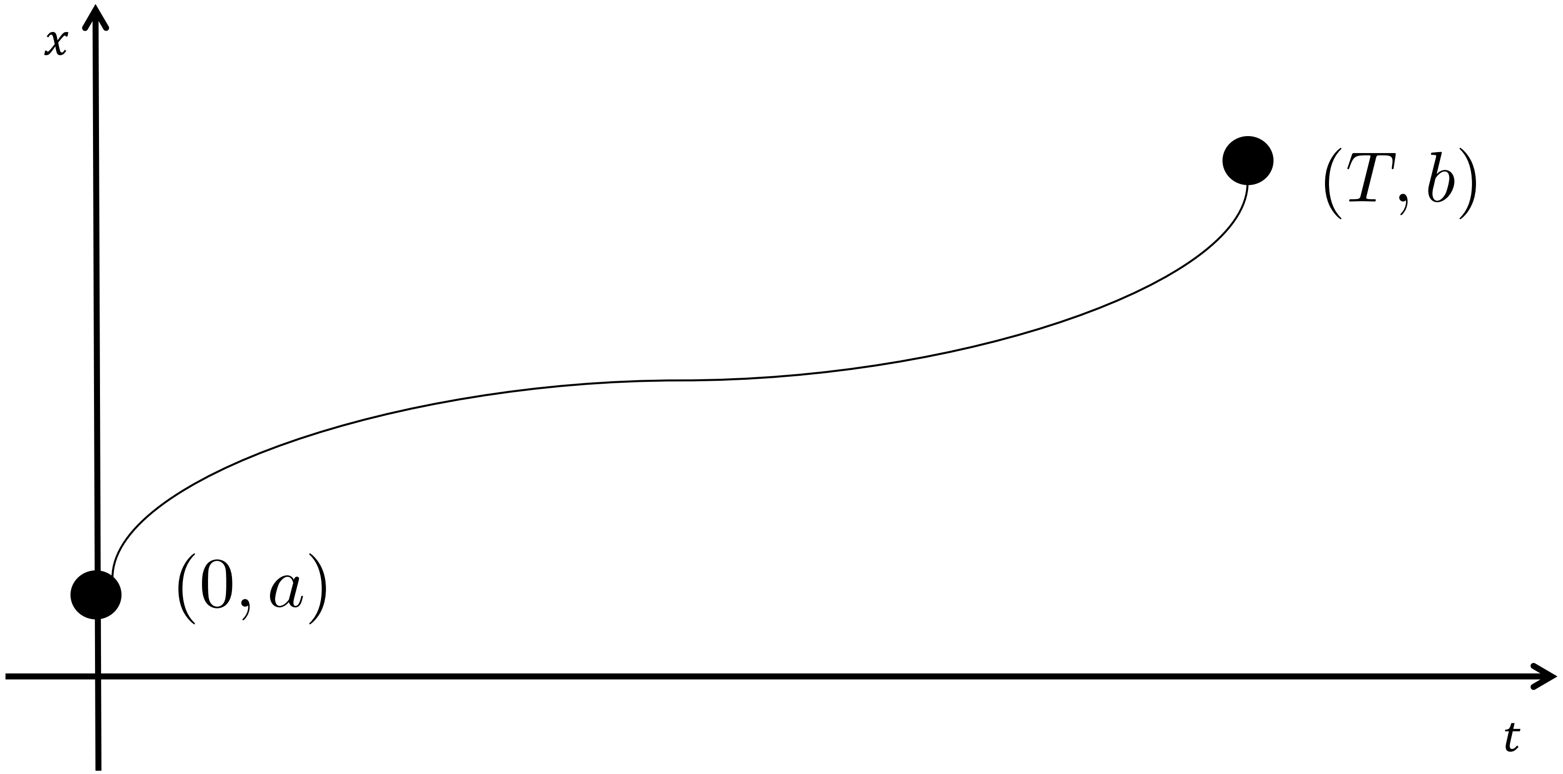
Euler Lagrange Equations

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

Waypoint Navigation

Smooth 1D Trajectories

Design a trajectory $x(t)$ s.t. $x(0) = a$ and $x(T) = b$



Minimum Acceleration Trajectory

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

Euler Lagrange Equation

- Necessary condition satisfied by the optimal function
- $\frac{\partial L}{\partial x} - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) + \frac{d^2}{dt^2}\left(\frac{\partial L}{\partial \ddot{x}}\right) = 0$
- $x^{(4)} = 0$
- $x(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3$

How do we find the c_i?

Solving for Coefficients

Boundary conditions:

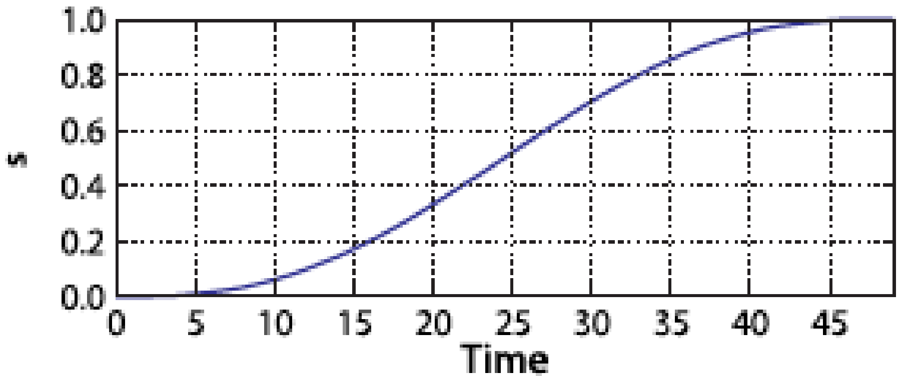
	Position	Velocity
$t = 0$	a	0
$t = T$	b	0

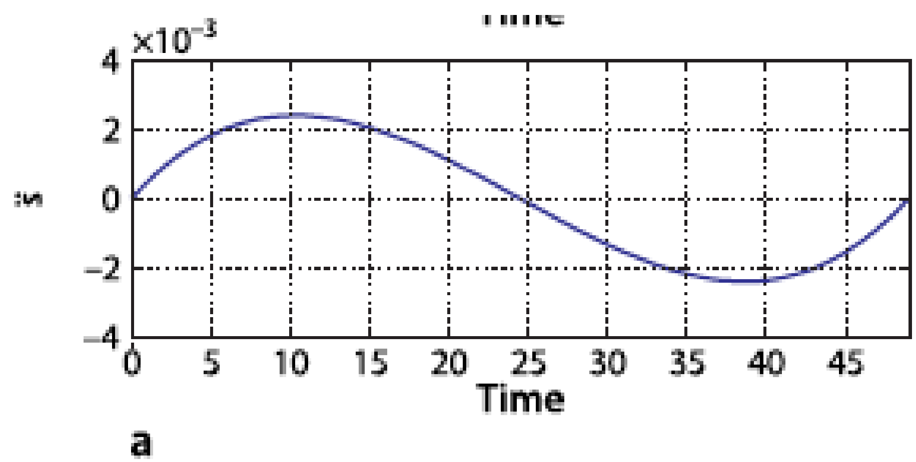
Solve:

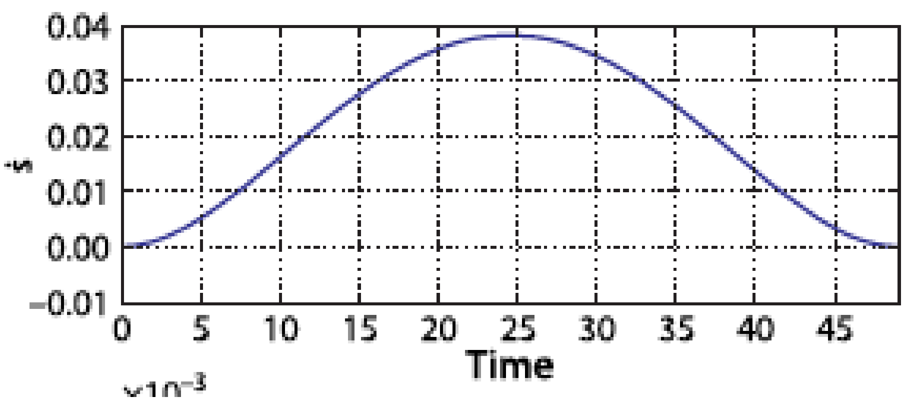
- Recall $x(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3$

$$\begin{bmatrix} a \\ b \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & T & T^2 & T^3 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2T & 3T^2 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

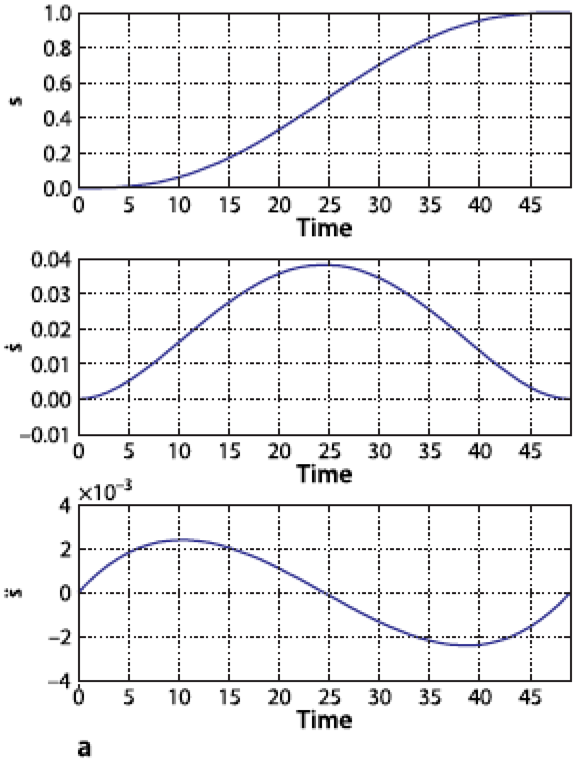
Motion Profiles

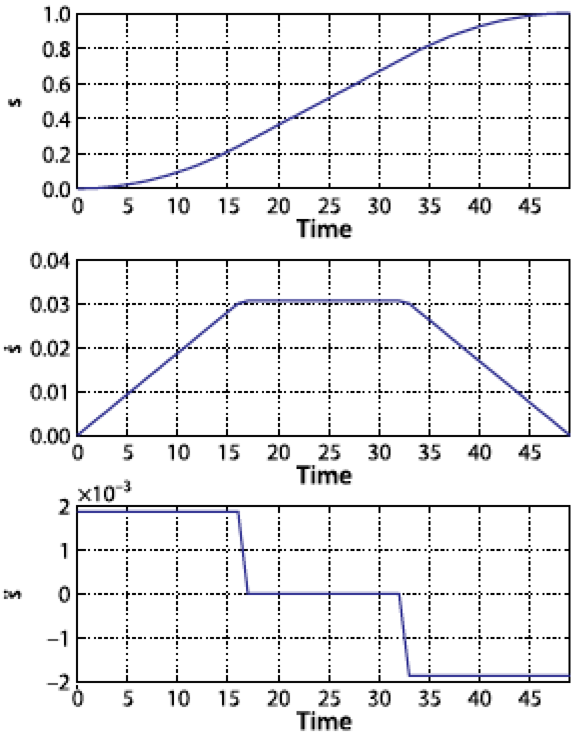






Bang-(Coast)-Bang Trajectories



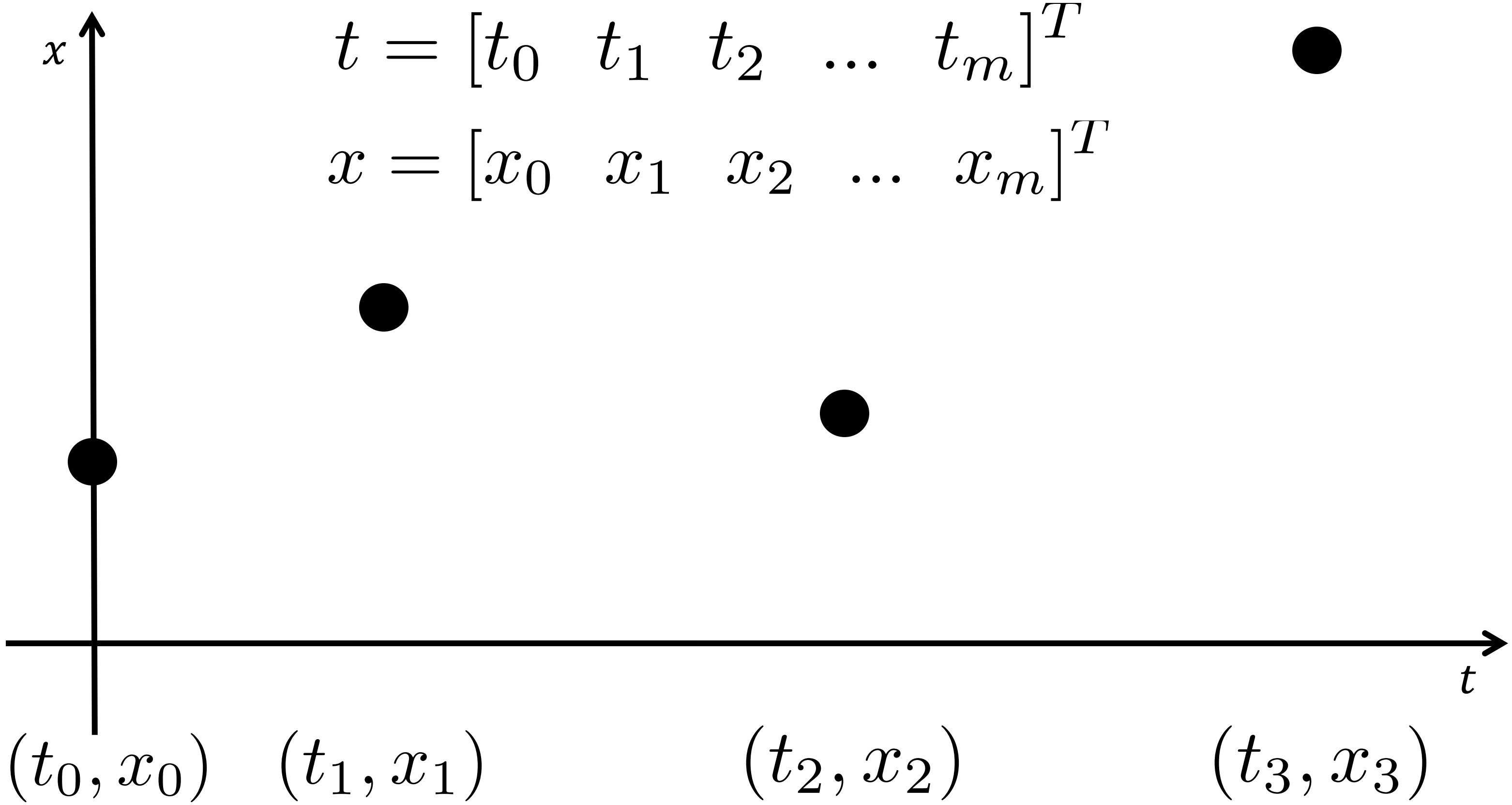


Multi-Segment 1D Trajectories

Design a trajectory such that

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

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



Multi-Segment 1D Trajectories

Design a trajectory such that

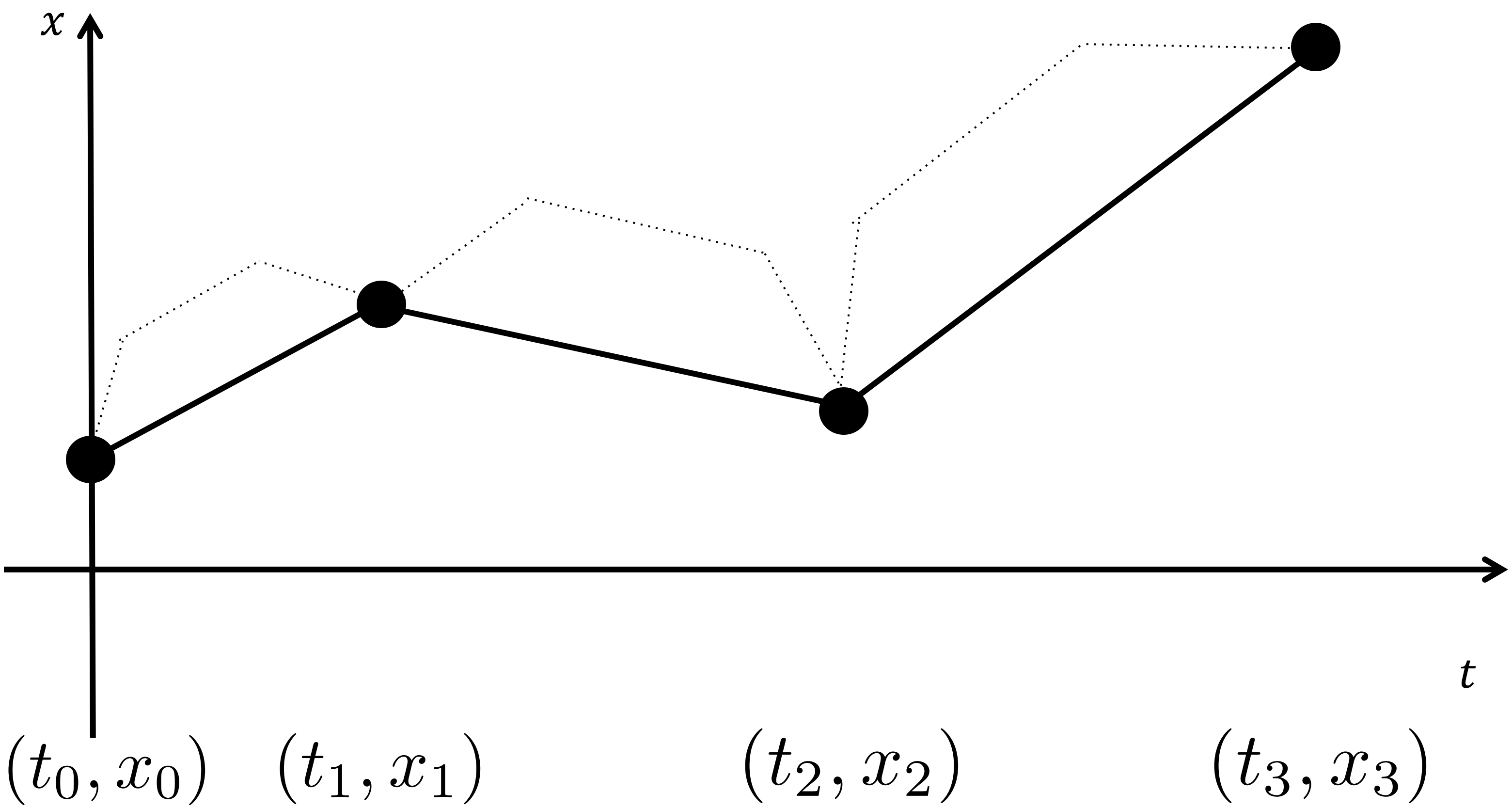
$$t = [t_0 \quad t_1 \quad t_2 \quad \dots \quad t_m]^T$$

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

Define a piecewise trajectory

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

Bang-(Coast)-Bang Segments



Cubic Spline

Design a trajectory such that

$$t = \begin{bmatrix} t_0 & t_1 & t_2 & \dots & t_m \end{bmatrix}^T$$

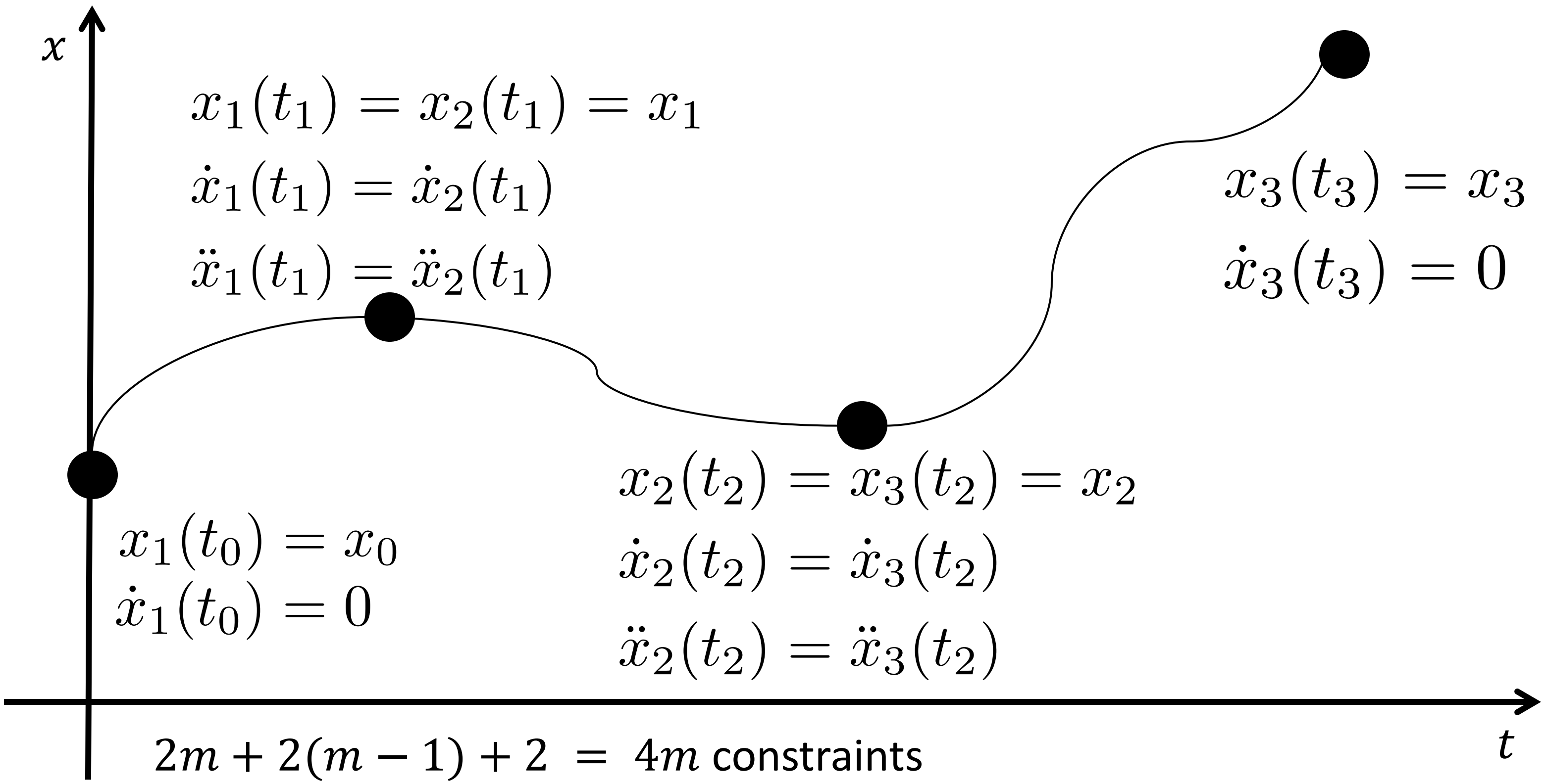
$$x = \begin{bmatrix} x_0 & x_1 & x_2 & \dots & x_m \end{bmatrix}^T$$

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

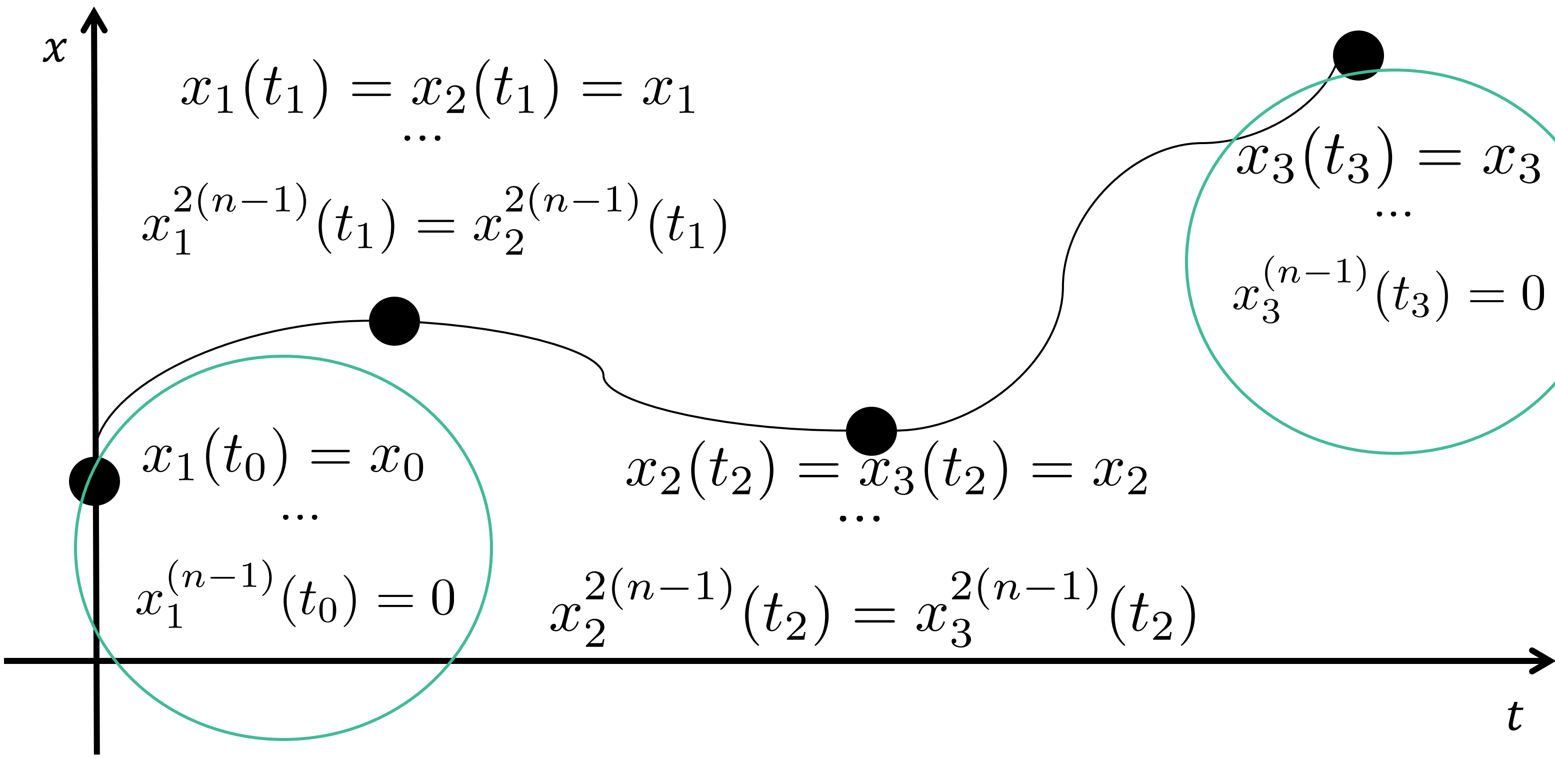
$4m$ degrees of freedom

Cubic Spline

Second order system $L(\ddot{x}, \dot{x}, x, t)$



Spline for nth Order System



$2nm$ constraints

Multi-Segment Multi-Dimensional Trajectories

Bang-(Coast)-Bang trajectories

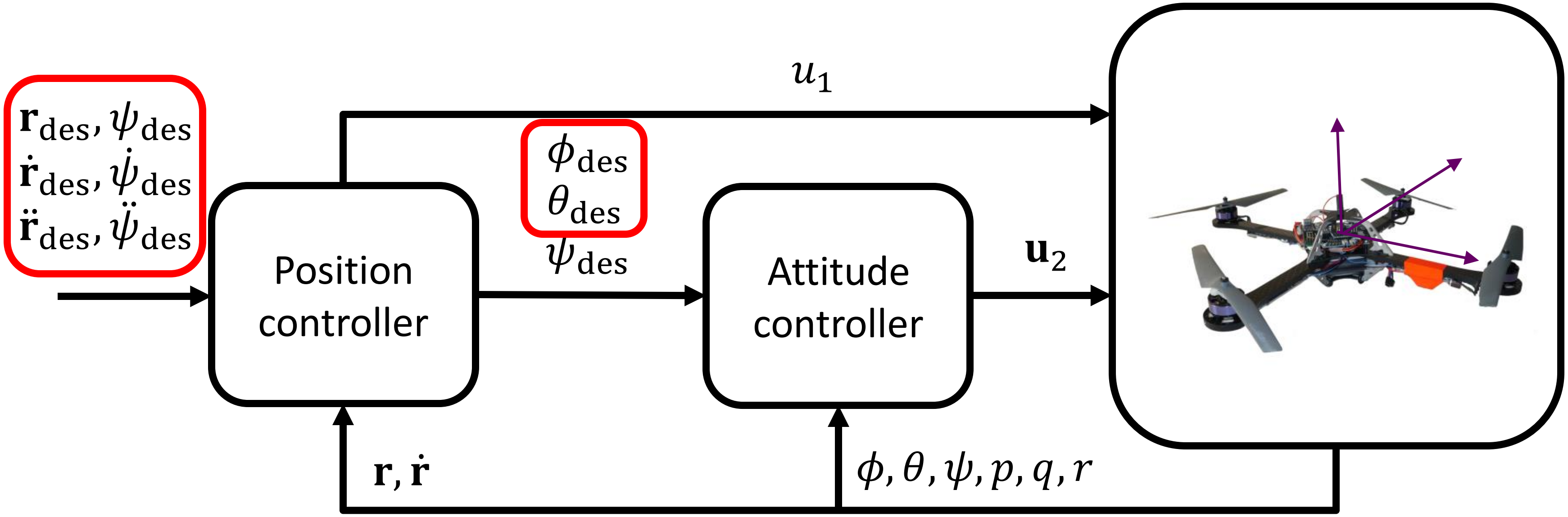
- Project the desired acceleration/velocity profile along the straight line connecting two waypoints

Polynomial trajectories

- Solve each dimension independently
- Euler-Lagrange equations decouple
- Make sure time constraints are the same

Application to Quadrotors

Quadrotor Control



$$m\ddot{\mathbf{r}} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ u_1 \end{bmatrix} \qquad I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

Differential Flatness

Flat systems have a *flat output*, which can be used to explicitly express all states and inputs in terms of the flat output and a finite number of its derivatives

- Implications: can plan in a lower dimensional space

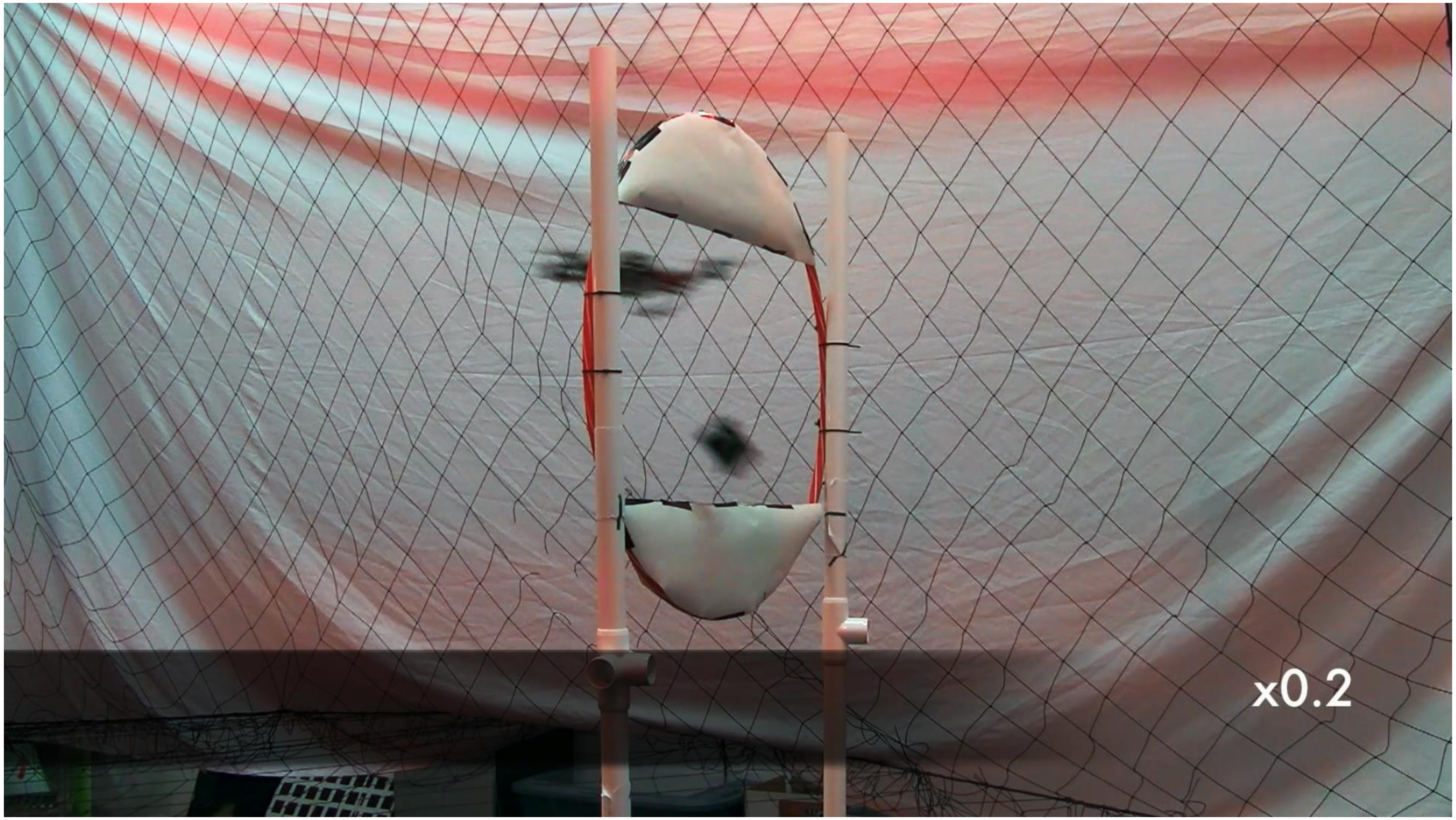
Quadrotors are differentially flat

- Flat outputs x, y, z, ψ
- The flat system is 4th order

See Daniel Mellinger and Vijay Kumar. "Minimum snap trajectory generation and control for quadrotors." *IEEE International Conference on Robotics and Automation (ICRA)*. 2011 for further details

Aerial Grasping

Payload Manipulation



Project 1

PHASE 3

Project 1 Phase 3

Due Tuesday Feb 16 at 11:59pm

Work individually

Combine Phases 1 and 2

- Generate safe and short paths through the environment (Phase 2)
- Smooth the path (this lecture)
- Follow the path (Phase 1)

Project 1 Phase 3

Will be running more tests than we give you

- Giving you a few baseline tests
- Please make your own!

Test locally before using turnin

- Your code will be given 10 minutes to run

Grading

- If you complete all of the tests you will earn at least 60%
  - If you collide with an obstacle, you will get a 0 on that test!
- Rest will depend on speed and accuracy