Trajectory Optimization

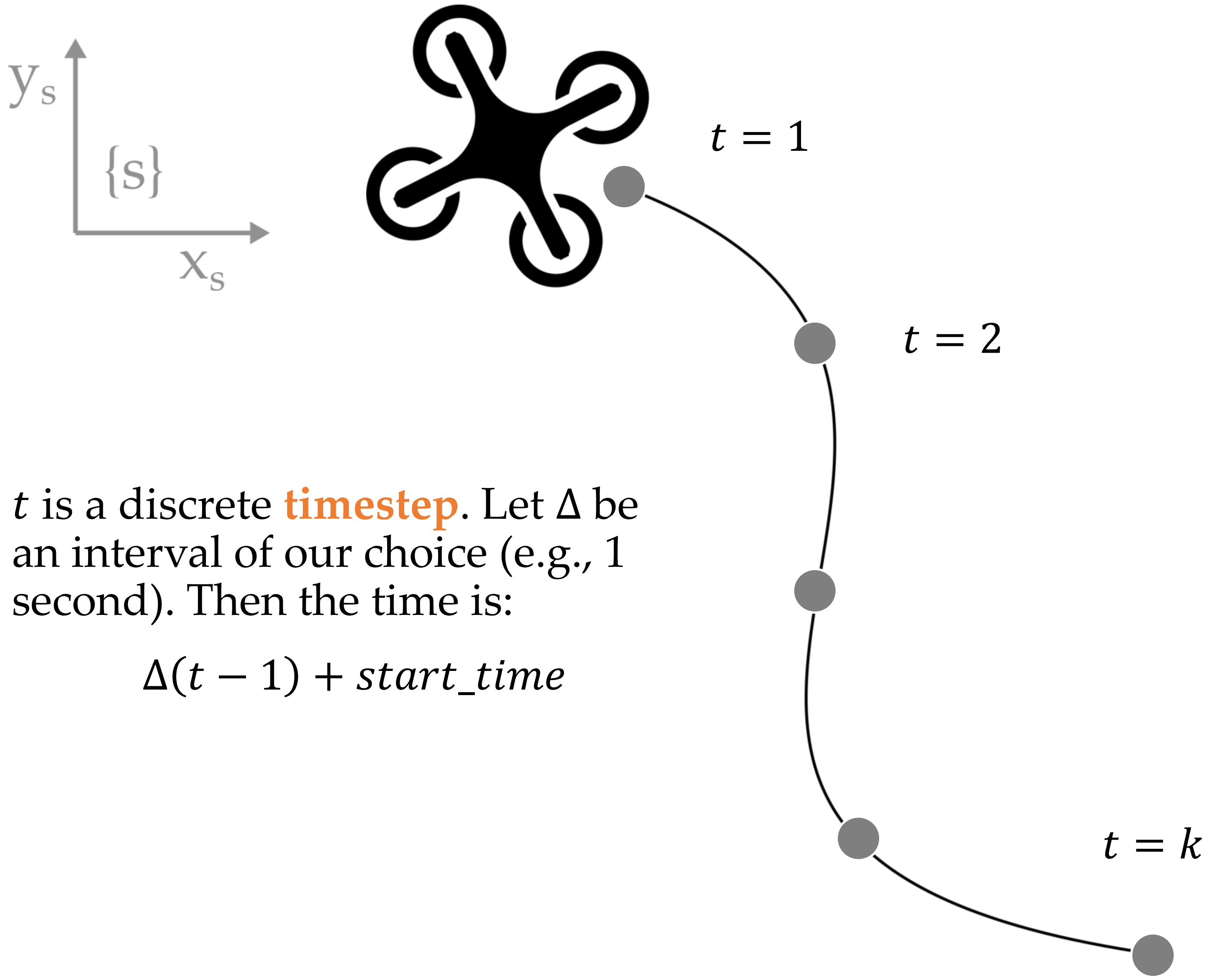
Reading: Modern Robotics 10.7


5

This Lecture

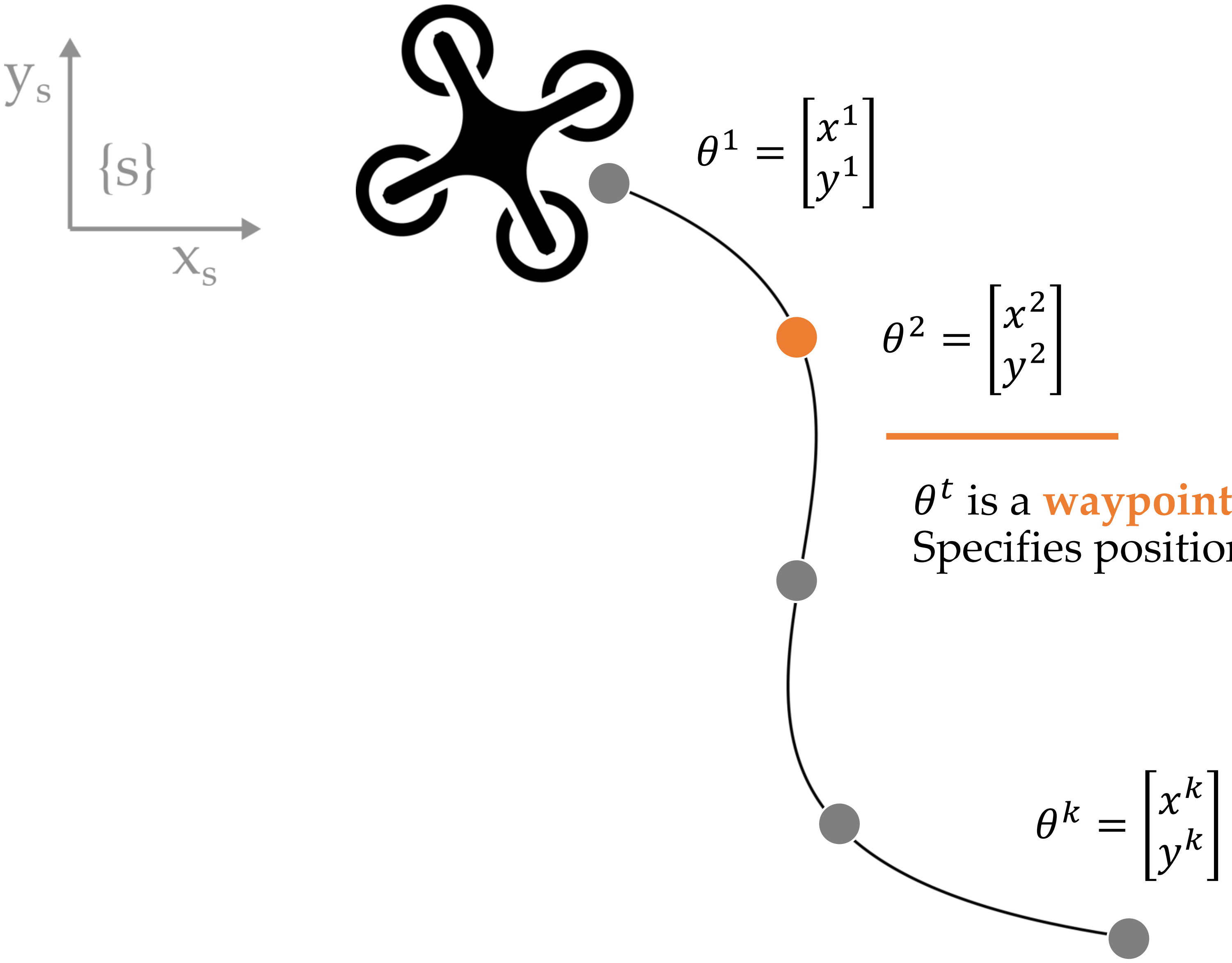
- What is a trajectory?
- How do we perform trajectory optimization?
- What are pros and cons of trajectory optimization?

What is a trajectory?

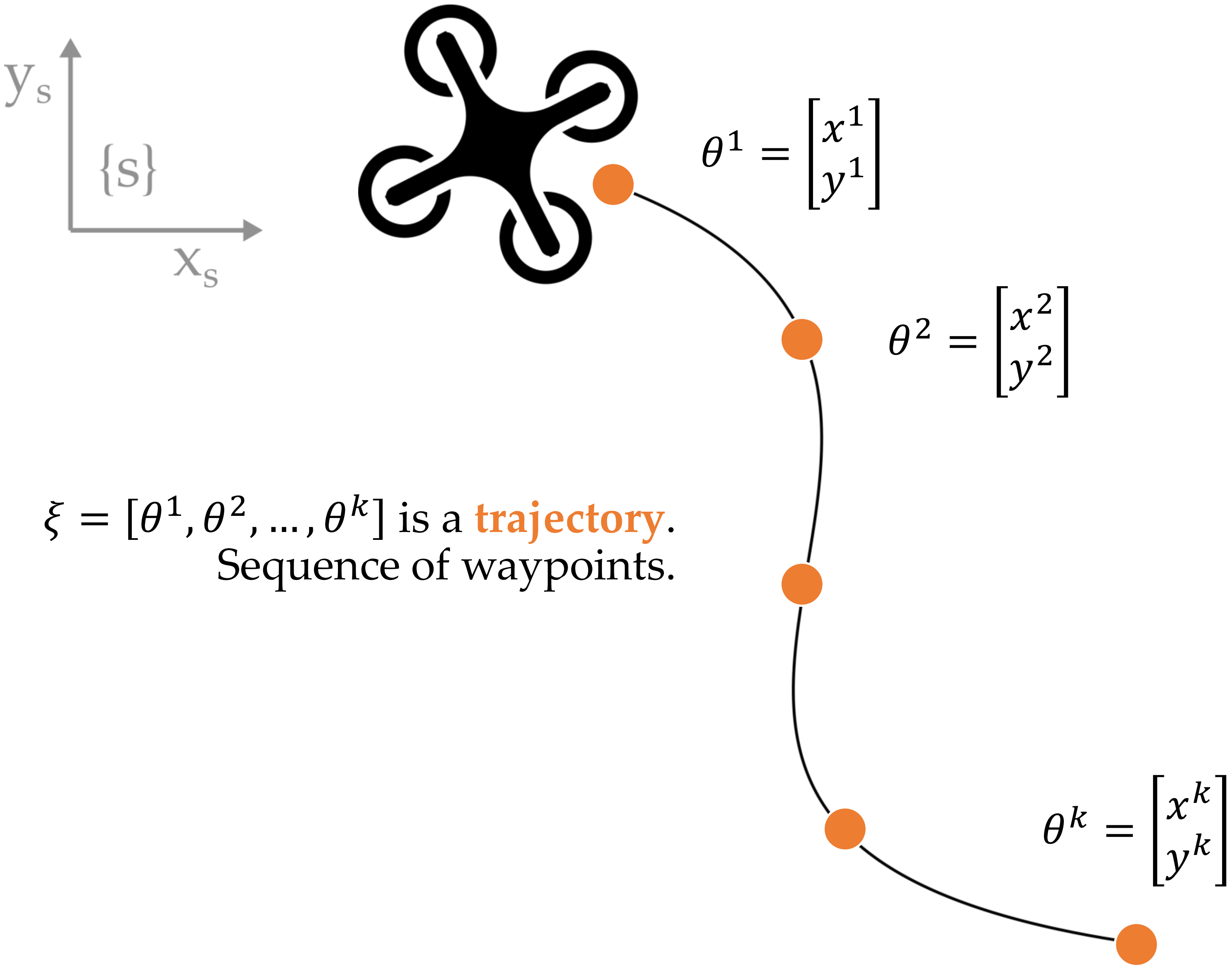


t is a discrete **timestep**. Let Δ be an interval of our choice (e.g., 1 second). Then the time is:

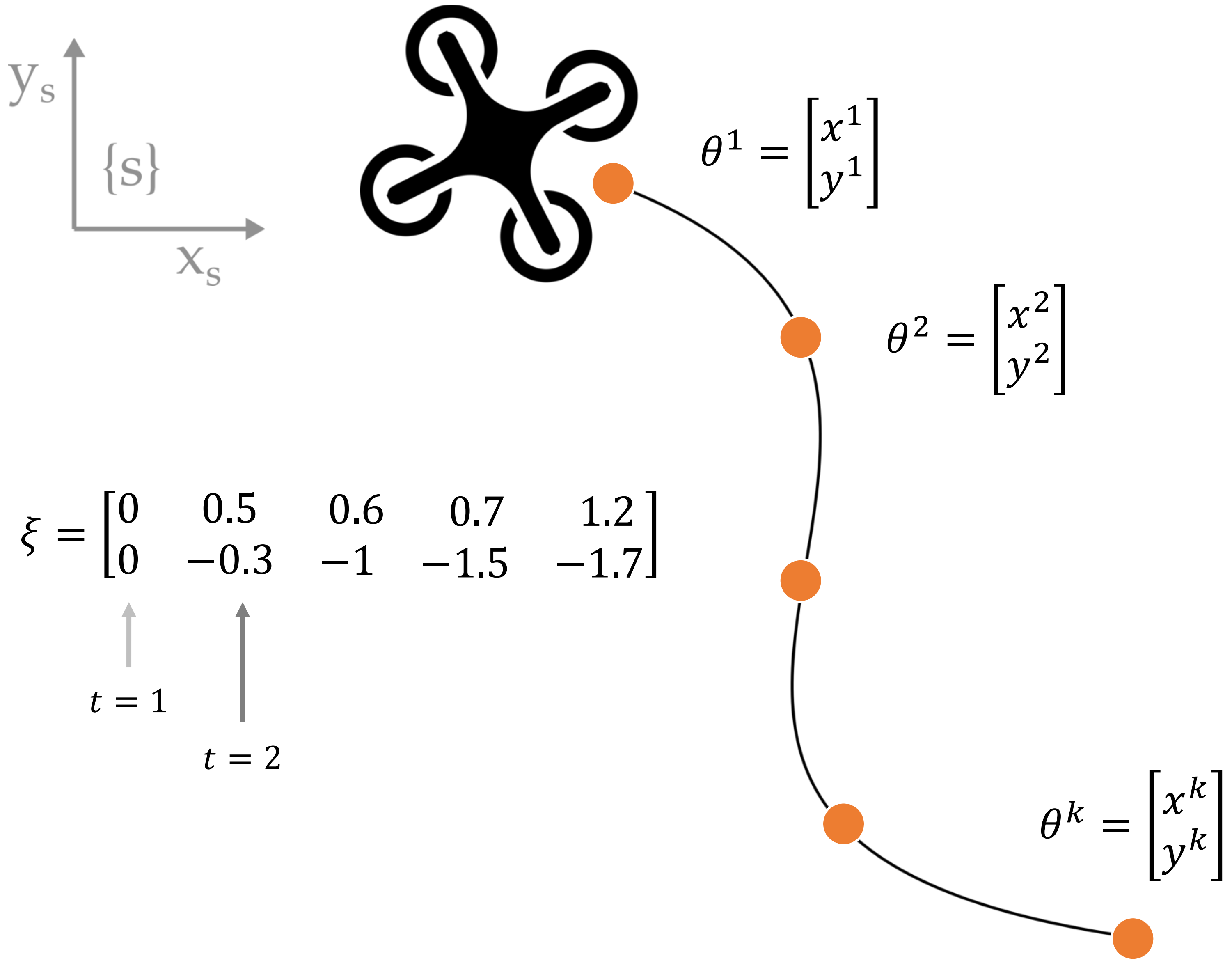
$$\Delta(t-1) + start\_time$$



θ^t is a **waypoint**.
Specifies position at timestep.



$\xi = [\theta^1, \theta^2, \dots, \theta^k]$ is a **trajectory**.
Sequence of waypoints.



y_s
$\{s\}$
x_s
$\theta^1 = \begin{bmatrix} x^1 \\ y^1 \end{bmatrix}$
$\theta^2 = \begin{bmatrix} x^2 \\ y^2 \end{bmatrix}$
$\theta^k = \begin{bmatrix} x^k \\ y^k \end{bmatrix}$
$\xi = \begin{bmatrix} 0 & 0.5 & 0.6 & 0.7 & 1.2 \\ 0 & -0.3 & -1 & -1.5 & -1.7 \end{bmatrix}$
$t = 1$
$t = 2$

Trajectory

A **trajectory** $\boldsymbol{\xi}$ is a sequence of waypoints.

We can write ξ as a **matrix**:

$$\xi = [\theta^1 \quad \theta^2 \quad \cdots \quad \theta^k]$$

- Robot has n joints so that $\theta \in \mathbb{R}^n$
- Trajectory has k waypoints
- Dimensions of ξ are $n \times k$

Trajectory

A **trajectory** $\boldsymbol{\xi}$ is a sequence of waypoints.

We can also write ξ as a **vector**:

$$\vec{\xi} = \begin{bmatrix} \theta^1 \\ \theta^2 \\ \vdots \\ \theta^k \end{bmatrix}$$

- Let's refer to this as $\vec{\xi}$
- Vector of length $n \cdot k$

Motion planning via **trajectory optimization**

Nonlinear Optimization

Solve for the trajectory ξ that:

$$\min C(\xi)$$
$$\text{s.t. } A\vec{\xi} = b$$

Nonlinear Optimization

Solve for the trajectory ξ that:

$$\min C(\xi)$$
$$\textbf{s.t. } \boldsymbol{A\vec{\xi} = b}$$

Constrain initial and final waypoints to the start and goal.

$$A = \begin{array}{cc} n \cdot k & \\ \begin{bmatrix} I_{n\times n} & 0_{n\times n(k-1)} \\ 0_{n\times n(k-1)} & I_{n\times n} \end{bmatrix} & 2n \end{array}$$

$$A\vec{\xi} = \begin{bmatrix} \theta^1 \\ \theta^k \end{bmatrix}, \qquad b = \begin{bmatrix} \theta_{start} \\ \theta_{goal} \end{bmatrix}$$

Nonlinear Optimization

Solve for the trajectory ξ that:

$$\min \boldsymbol{C}(\boldsymbol{\xi})$$

$$\text{s.t. } A\vec{\xi} = b$$

Minimize a cost function. For example, minimize the trajectory length and collisions with obstacles.

$$C(\xi) = \sum_{t=2}^{k} \underbrace{U_{rep}(\theta^t)}_{\textit{collisions}} + \underbrace{\|\theta^t - \theta^{t-1}\|^2}_{\textit{length}}$$

Nonlinear Optimization

Solve for the trajectory ξ that:

$$\mathbf{min}\, \boldsymbol{C}(\boldsymbol{\xi})$$

$$\text{s.t. } A\vec{\xi} = b$$

Minimize a cost function. For example, minimize the trajectory length and collisions with obstacles.

$$C(\xi) = \sum_{t=2}^{k} U_{rep}(\theta^t) + \|\theta^t - \theta^{t-1}\|^2$$

$$U_{rep}(\theta) = 0 \text{ if } \|c - \theta\| > r$$

$$U_{rep}(\theta) = \frac{1}{2}\gamma\left(\frac{1}{\|c-\theta\|} - \frac{1}{r}\right)^2 \text{ if } \|c - \theta\| \leq r$$

Sum across obstacles in environment

Nonlinear Optimization

- Given θ_{start} and θ_{goal} and initial guess ξ^0
- Given differentiable cost function $C : \Xi \rightarrow \mathbb{R}$

For $i = 0, 1, 2, \ldots$

$$\xi^{i+1} \leftarrow \xi^i - \alpha \nabla C\left(\xi^i\right)$$

$$\text{s.t. } A\vec{\xi} = b$$

In practice, use **nonlinear programming solver** (fmincon in matlab)

fmincon

Find minimum of constrained nonlinear multivariable function

R2022b

Syntax

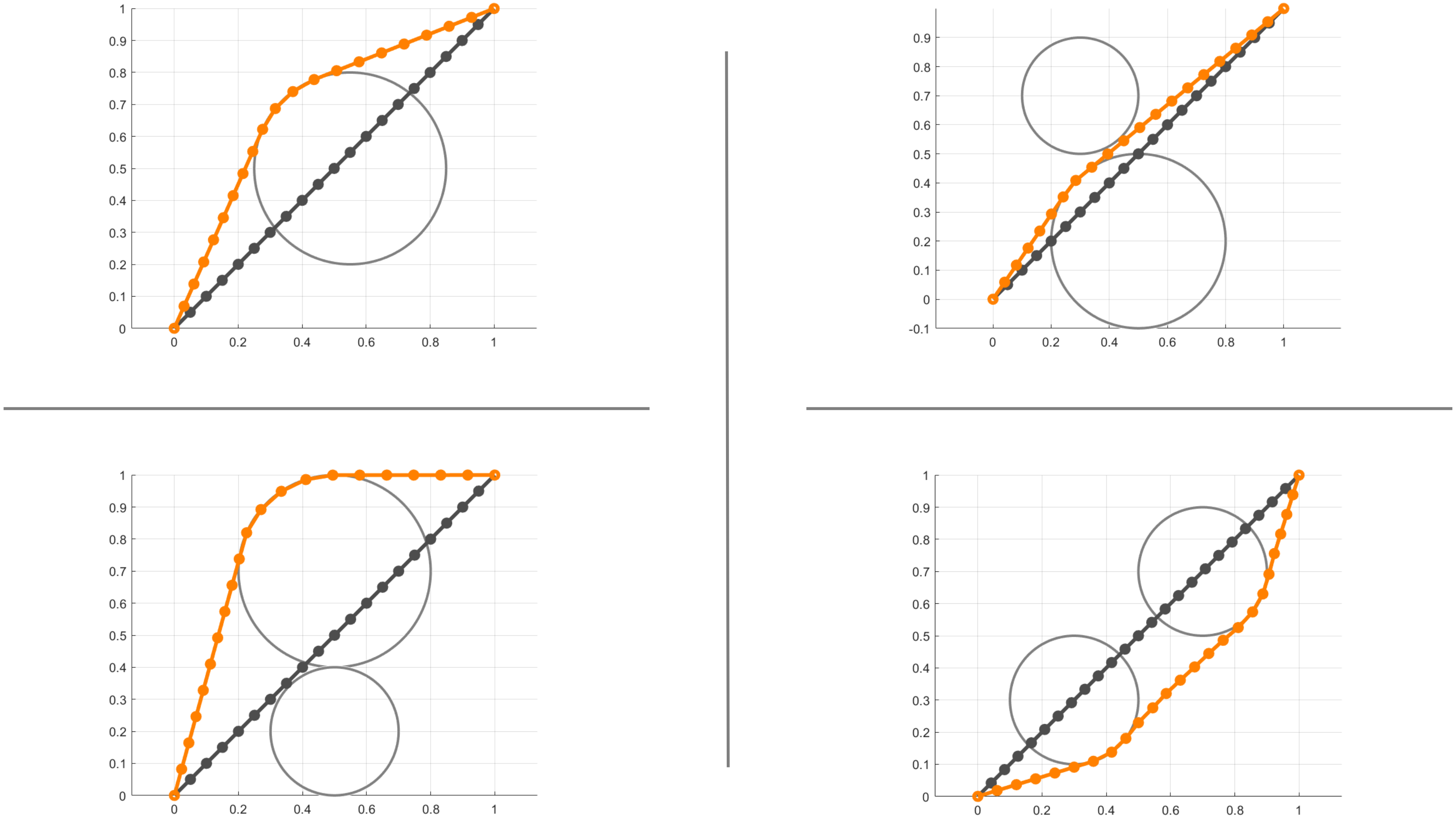
```
x = fmincon(fun,x0,A,b)
x = fmincon(fun,x0,A,b,Aeq,beq)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)
x = fmincon(problem)
[x,fval] = fmincon( __ )
[x,fval,exitflag,output] = fmincon( __ )
[x,fval,exitflag,output,lambda,grad,hessian] = fmincon( __ )
```

Description

Nonlinear programming solver.

Finds the minimum of a problem specified by

$$\min_x f(x) \text{ such that } \begin{cases} c(x) \leq 0 \\ ceq(x) = 0 \\ A \cdot x \leq b \\ Aeq \cdot x = beq \\ lb \leq x \leq ub, \end{cases}$$

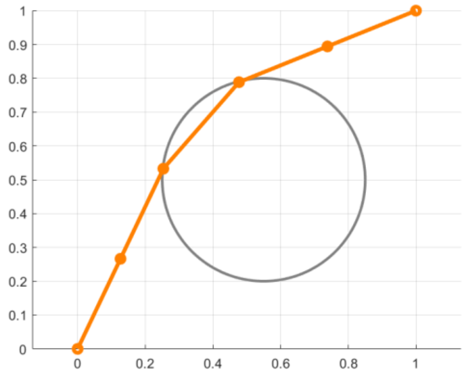

1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0 0.2 0.4 0.6 0.8 1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
0 0.2 0.4 0.6 0.8 1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0 0.2 0.4 0.6 0.8 1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0 0.2 0.4 0.6 0.8 1

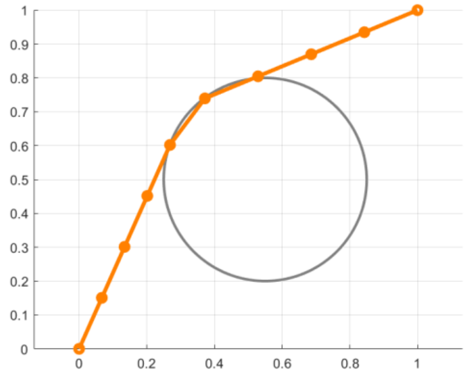
What are some **pros and cons** of trajectory optimization?

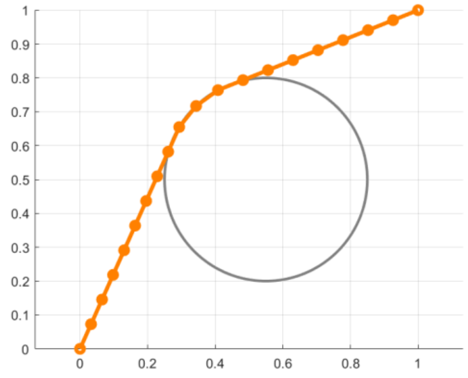
Pros and Cons

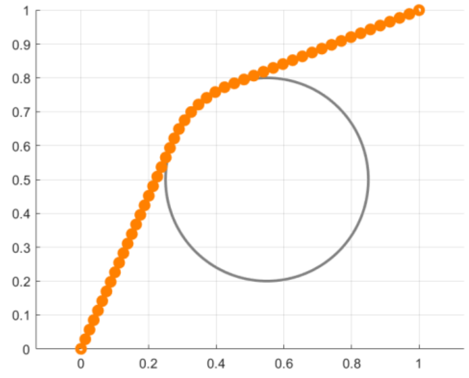
Advantage:

Produces a *smooth* trajectory as the number of timesteps increases.





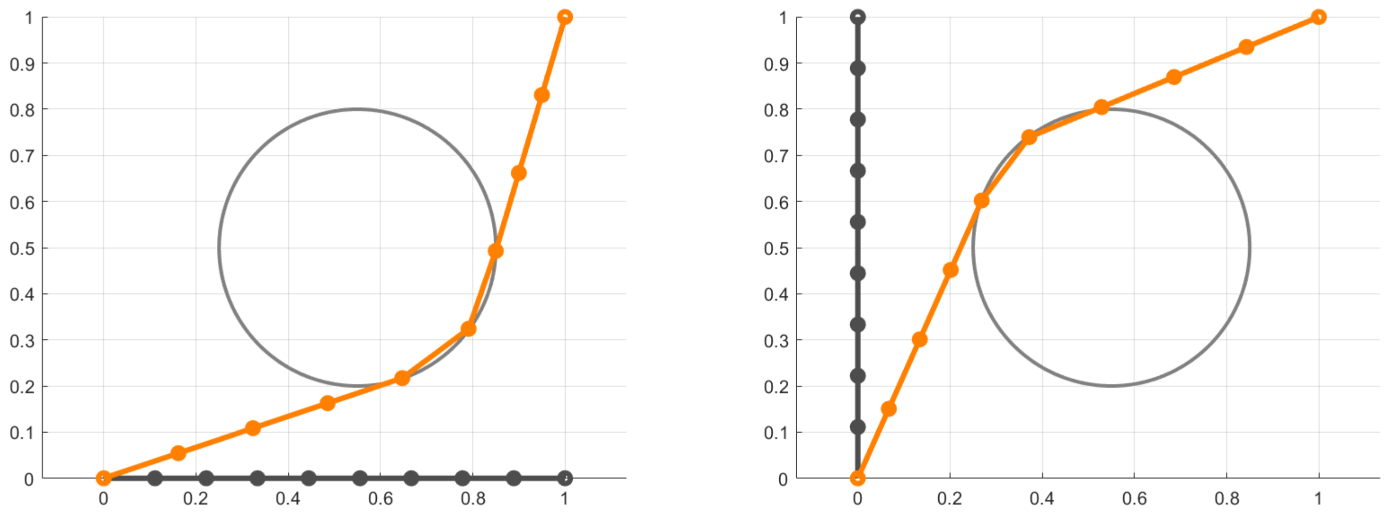




Pros and Cons

Advantage / Disadvantage:

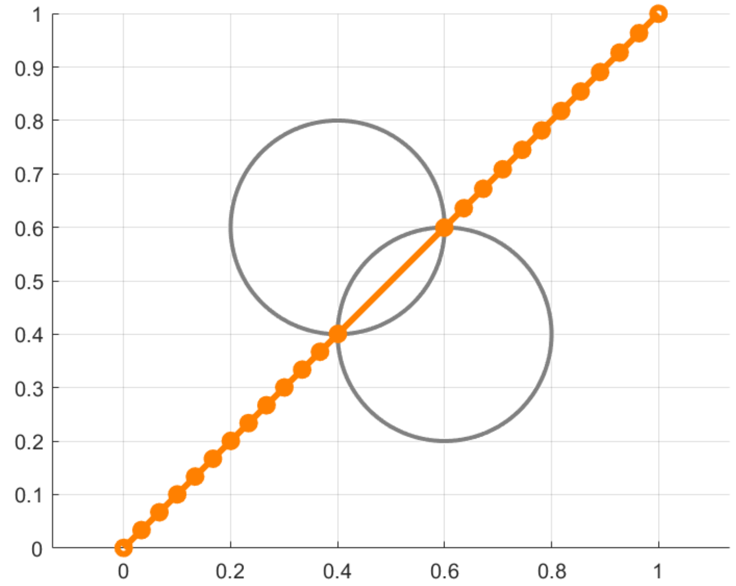
Uses gradient descent — solution depends on initial guess ξ^0



Pros and Cons

Disadvantage:

Uses gradient descent — can get stuck in local minimum



Pros and Cons

Disadvantage:

Uses gradient descent — can get stuck in local minimum

Solutions:

- Sample multiple initial trajectories
- Use alternate planner to get feasible path, then smooth with trajectory optimization

This Lecture

- What is a trajectory?
- How do we perform trajectory optimization?
- What are pros and cons of trajectory optimization?

Next Lecture

- Sampling-based motion planning