What is Trajectory Optimization?

June 13, 2025

The Fundamental Problem

Trajectory optimization answers: **"What is the best way for my robot to move from A to B?"**

Unlike path planning (which only considers geometry) or time scaling (which assumes a fixed path), trajectory optimization simultaneously determines:

- **Where to go** (spatial trajectory)
- **When to be there** (temporal profile)
- **How to get there** (control inputs)

Mathematical Framework

Goal: Find optimal state trajectory $\vec{x}(t)$ and control inputs $\vec{u}(t)$ that minimize a cost function while satisfying constraints.

General Form:

$$\min_{\vec{x}(\cdot),\vec{u}(\cdot)} J = \underbrace{\phi(\vec{x}(T))}_{\text{Terminal Cost}} + \underbrace{\int_0^T L(\vec{x}(t),\vec{u}(t))dt}_{\text{Running Cost}}$$

Subject to:

$$\dot{\vec{x}}(t) = f(\vec{x}(t),\vec{u}(t)) \qquad \text{(Dynamics)}$$
$$g(\vec{x}(t),\vec{u}(t)) \leq 0 \qquad \text{(Path constraints)}$$
$$\psi(\vec{x}(0),\vec{x}(T)) = 0 \qquad \text{(Boundary conditions)}$$

Example 1: Point Mass Robot - Minimum Time Problem

Problem Setup

A point mass robot needs to move from $(0,0)$ to $(10,5)$ in minimum time.

State: $\vec{x} = [x, y, \dot{x}, \dot{y}]^T$
Control: $\vec{u} = [u_x, u_y]^T$
Dynamics:

$$\dot{\vec{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ u_x \\ u_y \end{bmatrix}$$

Constraints:

- Control limits: $|\vec{u}| \leq u_{\max} = 2 \text{ m/s}^2$
- Velocity limits: $|\dot{\vec{x}}|, |\dot{\vec{y}}| \leq 5$ m/s

Boundary Conditions:

$$\vec{x}(0) = [0, 0, 0, 0]^T$$
$$\vec{x}(T) = [10, 5, 0, 0]^T$$

Cost Function: For minimum time: $J = T$ (equivalently, $L(\vec{x}, \vec{u}) = 1$)

Analytical Insight

This is a bang-bang control problem. Optimal solution:

- Accelerate at maximum toward target
- Decelerate at maximum to arrive at rest

Phase 1:

$$\vec{u} = u_{\max} \cdot \frac{\vec{d}}{|\vec{d}|}, \quad \vec{d} = [10, 5]^T$$

Phase 2:

$$\vec{u} = -u_{\max} \cdot \frac{\vec{v}}{|\vec{v}|}, \quad \vec{v} = \text{current velocity}$$

Numerical Solution

Normalized direction:

$$\hat{d} = \frac{[10, 5]}{\sqrt{125}} = [0.894, 0.447]$$

Phase 1 (Acceleration):

$$\vec{u} = 2 \cdot [0.894, 0.447] = [1.789, 0.894]$$

$$t_1 = \sqrt{\frac{|\vec{d}|}{|\vec{u}|}} = \sqrt{\frac{\sqrt{125}}{2}} \approx 2.5 \text{ seconds}$$

Phase 2 (Deceleration):

$$\vec{u} = -[1.789, 0.894], \quad t_2 = t_1 = 2.5 \text{ seconds}$$

Total time: $T = 5$ seconds

Example 2: Inverted Pendulum - Energy Optimal Swing-Up

Problem Setup

Swing up an inverted pendulum from hanging to upright with minimum energy.
State: $\vec{x} = [\theta, \dot{\theta}]^T$
Control: u (torque)
Dynamics:

$$\dot{\vec{x}} = \begin{bmatrix} \dot{\theta} \\ \frac{1}{I}(u - mgl\sin\theta - b\dot{\theta}) \end{bmatrix}$$

Parameters:

- $m = 1$ kg
- $l = 1$ m
- $I = ml^2 = 1$ kg·m^2
- $g = 9.81$ m/s^2
- $b = 0.1$ (damping)

Boundary Conditions:

$$\vec{x}(0) = [\pi, 0]^T$$
$$\vec{x}(T) = [0, 0]^T$$

Cost Function:

$$J = \int_0^T u^2 \, dt$$

This penalizes large control efforts, leading to smooth, efficient motion.

Solution Approach Using Direct Collocation

Step 1: Discretize Time

Divide $[0, T]$ into N segments:

$$t_k = k \cdot \frac{T}{N}, \quad k = 0, 1, \ldots, N$$

Step 2: Discrete Variables

- States: $\vec{x}_k \approx \vec{x}(t_k)$
- Controls: $u_k \approx u(t_k)$
- Time step: $h = T/N$

Step 3: Discrete Dynamics

Using Euler integration:

$$\vec{x}_{k+1} = \vec{x}_k + h \cdot f(\vec{x}_k, u_k)$$

Step 4: Optimization Problem

$$\min_{\{\vec{x}_k, u_k\}} \sum_{k=0}^{N-1} h \cdot u_k^2$$

Subject to:

$$\begin{aligned} \vec{x}_{k+1} - \vec{x}_k - h \cdot f(\vec{x}_k, u_k) &= 0 \quad \text{(Dynamics)} \\ \vec{x}_0 &= [\pi, 0]^T \\ \vec{x}_N &= [0, 0]^T \\ |u_k| &\leq u_{\max} \end{aligned}$$