Lecture 34 – Trajectory Generation

$\boxed{\text{Book Chapter 7}}$

Roadmap:

- Overview
- General Considerations
- Joint Space Trajectories
  - Cubic Polynomials
  - Trapezoidal Velocity Profiles

Overview:

Trajectory: A time evolution of motion variables

Joint-Space trajectory: $\theta(t)$ [and perhaps derivatives $\dot{\theta}$, $\ddot{\theta}$]

Task-Space / End effector trajectory: ${}^{0}T_e(t)$ and perhaps ${}^{0}\omega_e$, ${}^{0}v_e$

General Considerations

- Design (by a human or computer)
- Storage (within a computer)

this lecture

- Execution (via control)

Given desired final pose ${}^0T_e^f$, current θ_0, total time t_f

... think of as many way as possible to connect intitial & Final points

① • ${}^0T_e(\theta_0)$ - then draw straight line in cartesian space

- How to draw a "Straight line" for rotation matrices?

② • θ_f based on IK then move from θ_0 to θ_f w/ constant velocity

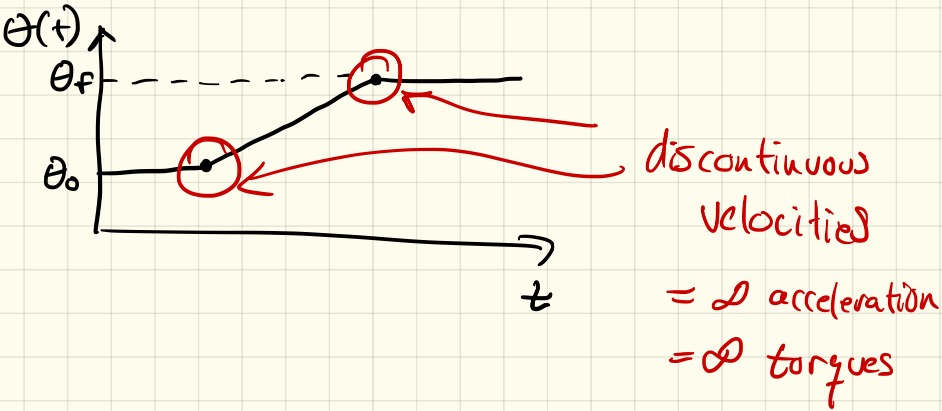
• Design a set of segments (task space or joint space) use ① or ②

• Spline curve!

The Simplest trajectory Design: Straight line in joint space

- Given desired final pose ${}^0T_e^f$ and current θ_0
- Solve IK problem $\quad \mathrm{InvKin}({}^0T_e^f) \to \theta_f$
- Interpolate

$$\theta(t) = \theta_0 + (\theta_f - \theta_0)\, t/t_f \qquad 0 \le t \le t_f$$



- Presents a problem for execution

Designing Smooth Motions:

- Suppose some initial State $\theta_0, \dot{\theta}_0$ and a desired Final State $\theta_f, \dot{\theta}_f$. Rather than a line $(a_0 + a_1 t)$ we could use a polynomial

$$\theta(t) = a_n t^n + a_{n-1} t^{n-1} + \dots + a_1 t + a_0$$

$$a_3 t^3 + a_2 t^2 + a_1 t + a_0$$

- The minimum order to meet endpoint constraints? 3rd (4 coefficients)

- Cubic polynomial also called a cubic Spline

- Coefficients easily stored

Example: Suppose $\theta_0 = 15°$, $\theta_f = 75°$, $\dot{\theta}_0 = \dot{\theta}_f = 0$ deg/sec

Design a 3 second cubic spline trajectory to meet these constraints

$$\theta(t) = a_3 t^3 + a_2 t^2 + a_1 t + a_0$$

$$\theta(0) = \theta_0 = a_0 \Rightarrow a_0 = 15°$$

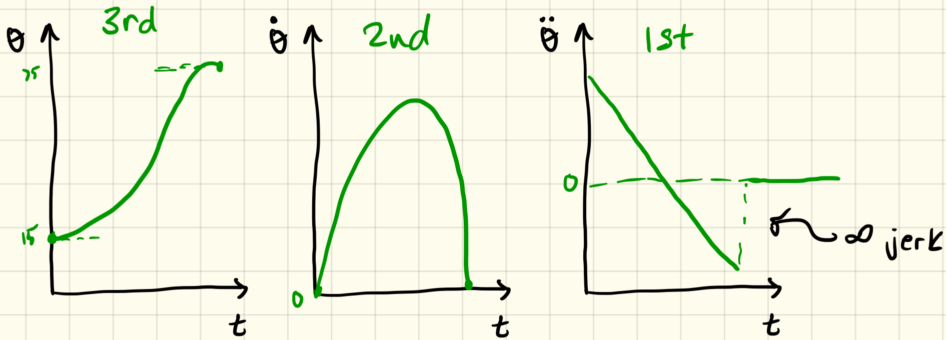
$$\dot{\theta}(0) = \dot{\theta}_0 = a_1 \Rightarrow a_1 = 0°/sec$$

$$\theta(3) = 27a_3 + 9a_2 + 15° = 75°$$

$$\dot{\theta}(3) = 27a_3 + 6a_2 = 0°/sec$$

$$a_2 = 20°/s^2 \qquad a_3 = -4.4\bar{4}°/s^3$$

Result:



Cubic Splines In general

$$\theta(t) = a_3 t^3 + a_2 t^2 + a_1 t_1 + a_0$$

- $\theta_0, \dot{\theta}_0, \theta_f, \dot{\theta}_f, t_f$

Computer Project 2

$$\boxed{\begin{array}{ll} a_0 = \theta_0 & a_2 = \frac{3}{t_f^2}(\theta_f - \theta_0) - \frac{2}{t_f}\dot{\theta}_0 - \frac{1}{t_f}\dot{\theta}_f \\ a_1 = \dot{\theta}_0 & a_3 = -\frac{2}{t_f^3}(\theta_f - \theta_0) + \frac{1}{t_f^2}(\dot{\theta}_f + \dot{\theta}_0) \end{array}}$$

Via Points & combinations of cubic Splines

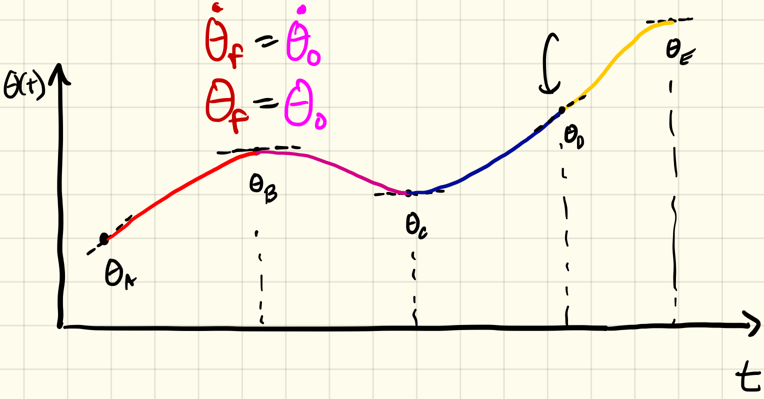
- Rather than simply a final pose, intermediate points "Via points" may be supplied
  - To avoid obstacles
  - To improve appearance (e.g. theme park robots)

$\ddot{\theta}_f \neq \ddot{\theta}_0$

$\dot{\theta}_f = \dot{\theta}_0$

$\theta_f = \theta_0$

Intermediate slopes set heuristically

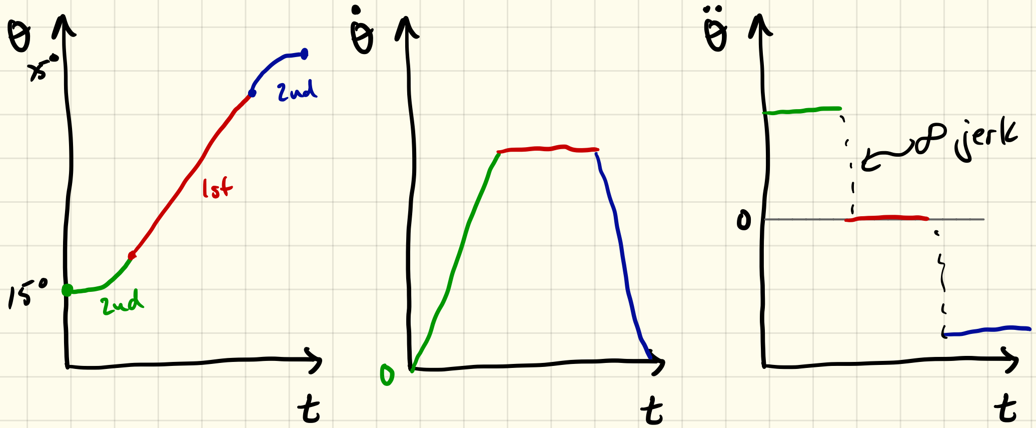


Each color:

- Separate cubic spline
- Linked to next spline with velocity / position continuity

Common Option: Linear trajectories w/ Parabolic blends

- Especially common w/ industrial servo controllers
- Also called trapezoidal velocity profiles
- Efficiently stored on the computer



Recap:

- Designing Trajectories in joint space
  - Cubic Splines
  - Linear interpolation w/ parabolic blends
  - Can use output of IK to design via points and final goal