Topics Covered:

- Trajectory Design
- Cubic Polynomials / Splines
- Manipulator Trajectories

Additional Reading:

- LP Chapter 9
- Craig Chapter 7

Note about Matrix Exponential

There's been a few questions on how to compute the exponential map by hand. One way to do it is the formula we introduced earlier in this course:

$$\exp(\hat{\xi}\theta) = \begin{bmatrix} e^{[\omega]_\times \theta} & (I - e^{\hat{\omega}\theta})[\omega]_\times \hat{v} + \omega\omega^T v\theta \\ 0 & 1 \end{bmatrix}$$

where $\exp([\omega]_\times \theta)$ is computed using Rodrigues' formula (and assuming $\|\omega\| = 1$):

$$e^{[\omega]_\times \theta} = I + \sin(\theta)[\omega]_\times + (1 - \cos(\theta))[\omega]_\times^2$$

However, as we saw with our computation of twists, this formula can be simplified if we know that our associated joint is a revolute joint. In this case, our exponential map simplifies to:

$$e^{\hat{\xi}\theta} = \begin{bmatrix} e^{[\omega]_\times \theta} & (I - e^{[\omega]_\times \theta})q \\ 0 & 1 \end{bmatrix}$$

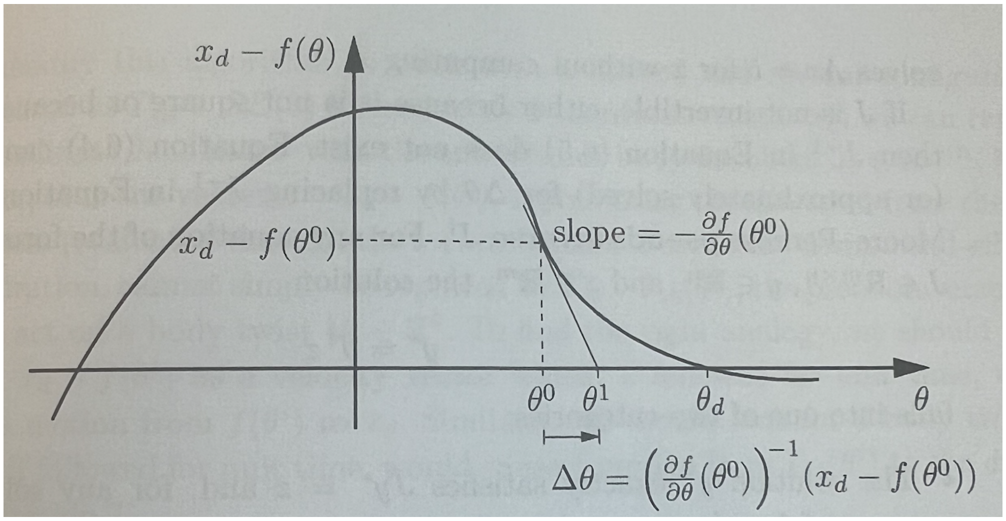
where q is the position vector from the origin to any point on the rotation axis associated with ω. And again, $\exp([\omega]_\times \theta)$ is computed using Rodrigues' formula. Alternatively, if the rotation axis is about a principle axis, this expression can be obtained using our traditional rotation matrices.

If the joint is prismatic, the exponential map simplifies to:

$$\exp(\hat{\xi}\theta) = \begin{bmatrix} I & v\theta \\ 0 & 1 \end{bmatrix}$$

Note about Newton-Raphson Method

There have also been some questions about how to implement the Newton-Raphson method visually. There was one example from the notes:



As shown, the gradient step is the same as following the tangent line of the curve (evaluated at the point $g(\theta_k)$) to the point where it intersects the x-axis. This intuition comes from the following derivation:

$$\begin{aligned} \frac{\partial g(\theta_1)}{\partial \theta} &= \frac{g(\theta_2) - g(\theta_1)}{\theta_2 - \theta_1} \\ &= \end{aligned}$$

Since we're trying to find the point where $g(\theta_2) \approx 0$, we can take this to be zero. We will also replace $\theta_2 - \theta_1$ with $\Delta\theta$:

$$\begin{aligned} \frac{\partial g(\theta_1)}{\partial \theta} = \frac{\partial (x_d - f(\theta))}{\partial \theta} = \frac{-\partial f}{\partial \theta_1} &= \frac{0 - (x_d - f(\theta_1))}{\Delta\theta} \\ \frac{\partial f}{\partial \theta_1} &= \frac{x_d - f(\theta_1)}{\Delta\theta} \\ \Delta\theta &= \left(\frac{\partial f(\theta_1)}{\partial \theta_1}\right)^{-1} (x_d - f(\theta_1)) \end{aligned}$$

So this gradient step is the same thing as following the tangent line to the point where it intersects the x-axis.

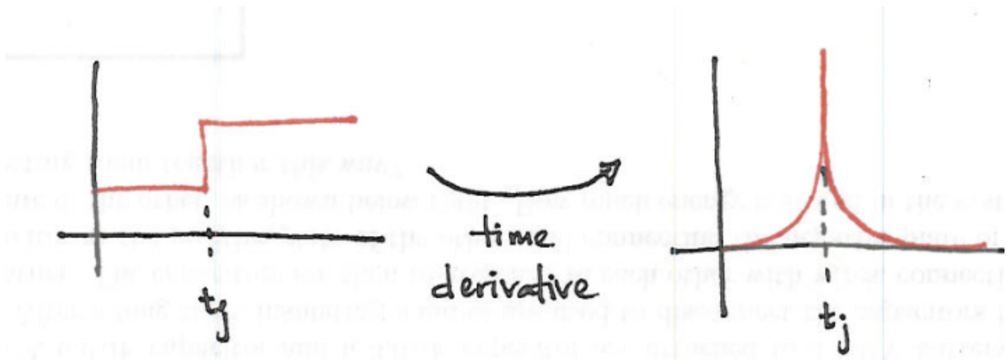
Introduction to Trajectory Design

Goal: to design realistic/feasible trajectories connecting different configurations

- joint configurations
- group configurations
- both levels? $\rightarrow$ sometimes not possible using only one configuration space. may need to utilize both.

Trajectory Design

- want sufficiently smooth trajectories (continuous derivatives). No jerky motions which cause wear, induce vibrations, excite resonances, etc. The delta function has an infinite derivative

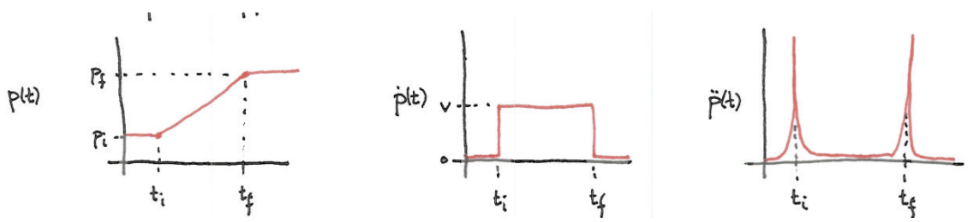


  at t_j. In frequency domain, excites all frequencies.

- some trajectories may require intermediate points between the initial and final configurations. (known as: way points, via points, knot points)

How should trajectories be generated/designed?

- The simplest approach is linear



- need to manage velocity better
- let's add velocity constraints on initial and final configurations

Cubic Polynomials / Splines

A polynomial is defined as:
$$p(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

Why cubic? Because of the number of constraints we are imposing

So, minimally, a cubic polynomial is required.

4 constraints on the polynomial:
1. initial point p_i
2. final point p_f
3. initial velocity $\dot{p}_i$
4. final velocity $\dot{p}_f$

To simply connect two points, constraints are

$$\begin{aligned} p(0) &= p_i, & \dot{p}(0) &= 0 \\ p(t_f) &= p_f, & \dot{p}(t_f) &= 0 \end{aligned}$$

Note:

$$\begin{aligned} \dot{p}(t) &= a_1 + 2a_2 t + 3a_3 t^2 \\ \ddot{p}(t) &= 2a_2 + 6a_3 t \end{aligned}$$

Thus, we can solve for the polynomial coefficients:

$$\begin{aligned} p(0) &= a_0 && = p_i \\ p(t_f) &= a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 = p_f && = p_f \\ \dot{p}(0) &= a_1 && = 0 \\ \dot{p}(t_f) &= a_1 + 2a_2 t_f + 3a_3 t_f^2 = 0 && = 0 \end{aligned}$$

Putting these equations into matrix form yields:

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & t_f & t_f^2 & t_f^3 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2t_f & 3t_f^2 \end{bmatrix}}_{A(t_f)} \vec{a} = \vec{p}_0, \quad \text{where } \vec{a} = \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{Bmatrix}, \ \vec{p}_0 = \begin{Bmatrix} p_i \\ p_f \\ 0 \\ 0 \end{Bmatrix}$$

By inverting this matrix, we can directly obtain a matrix form of the polynomial coefficients:

$$\vec{a} = P(t_f)\vec{p}_0, \ \text{where } P(t_f) = A^{-1}(t_f)$$

$$P(t_f) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3/t_f^2 & 3/t_f^2 & -2/t_f & -1/t_f \\ 2/t_f^3 & -2/t_f^3 & 1/t_f^2 & 1/t_f^2 \end{bmatrix}$$

but, notice that $\vec{p}_0$ has zeros in the last two rows, thus the last two columns of $P(t_f)$ are not needed.

$$\vec{a} = P_{simp}(t_f) \begin{Bmatrix} p_i \\ p_f \end{Bmatrix}, \ \text{where } P_{simp}(t_f) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ -3/t_f^2 & 3/t_f^2 \\ 2/t_f^3 & -2/t_f^3 \end{bmatrix}$$

worked out individually, this is

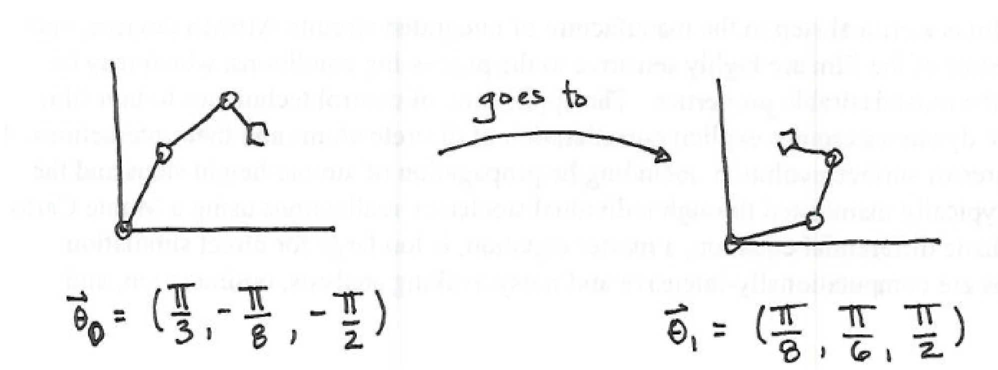
$$a_0 = p_i \qquad\qquad a_2 = \frac{3}{t_f^2}(p_f - p_i)$$

$$a_1 = 0 \qquad\qquad a_3 = \frac{2}{t_f^3}(p_i - p_f)$$

What value should t_f take?

- certainly should be short enough so that trajectory doesn't take forever.
- not too short though: actuator limits, end-effector velocity/acceleration limits
- have decent initial guess based on limits, $v_{lim} \cdot t = d$ (v_{lim} is the limit/nominal velocity, t is the unknown time to solve, and d is the distance)

Example



Goal: join initial and final joint configurations with trajectory whose velocity does not exceed $\pi/4$ rad/sec.

$$t_f = \frac{d}{v_{max}} = \frac{\pi \text{ rad}}{\pi/4 \text{ rad/sec}} = 4\text{sec}$$

We can solving for each joint angle polynomial by plugging the values for $\theta_i(0)$ and $\theta_i(t_f)$ into the equation we obtained ($\theta_i(t) = p_i + 0t + \frac{3}{t_f^2}(p_f - p_i)t^2 + \frac{2}{t_f^3}(p_i - p_f)t^3$):

$$
\begin{aligned}
\theta_1(t) &= \frac{\pi}{3} + \frac{3}{4^2}\left(\frac{\pi}{8} - \frac{\pi}{3}\right)t^2 + \frac{2}{4^3}\left(\frac{\pi}{3} - \frac{\pi}{8}\right)t^3 \\
&= \frac{\pi}{3} + \frac{3}{16}\frac{-5\pi}{24}t^2 + \frac{2}{64}\frac{5\pi}{24}t^3 \\
&= \frac{\pi}{3} - \frac{5\pi}{128}t^2 + \frac{5\pi}{768}t^3 \\
\theta_2(t) &= -\frac{\pi}{8} + \frac{3}{4^2}\left(\frac{\pi}{6} + \frac{\pi}{8}\right)t^2 + \frac{2}{4^3}\left(-\frac{\pi}{8} - \frac{\pi}{6}\right)t^3 \\
&= -\frac{\pi}{8} + \frac{3}{16}\frac{7\pi}{24}t^2 + \frac{2}{64}\frac{-7}{24}t^3 \\
&= -\frac{\pi}{8} + \frac{7\pi}{128}t^2 - \frac{7}{768}t^3 \\
\theta_3(t) &= -\frac{\pi}{2} + \frac{3}{4^2}\left(\frac{\pi}{2} + \frac{\pi}{2}\right)t^2 + \frac{2}{4^3}\left(-\frac{\pi}{2} - \frac{\pi}{2}\right)t^3 \\
&= -\frac{\pi}{2} + \frac{3\pi}{16}t^2 - \frac{\pi}{32}t^3
\end{aligned}
$$

where each of these polynomials are enforced for $t \in [0, 4]$.

If we examine $\dot{\theta}_3(t)$, it actually goes more than $\pi/4$ rad/sec (due to initial acceleration and final deceleration).

Extending t_f to 7 seconds reduces the velocity below $\pi/4$ rad/sec:

$$
\begin{aligned}
\theta_1(t) &= \frac{\pi}{3} - \frac{5\pi}{392}t^2 + \frac{5\pi}{4116}t^3 \\
\theta_2(t) &= -\frac{\pi}{8} + \frac{\pi}{5}t^2 - \frac{\pi}{588}t^3 \\
\theta_3(t) &= -\frac{\pi}{2} + \frac{3\pi}{49}t^2 - \frac{2\pi}{343}t^3
\end{aligned}
$$

where now the polynomials are for $t \in [0, 7]$.