Lecture 3B.2

Time, Motion, and Trajectories (continued)

Now to look at a concrete setting, let's discuss the problem of designing a minimum jerk trajectory, which makes sense when we have a third-order system. And let's assume that we've specified the starting position and the end position.

Our functional in this case is simply the integral of the square of the jerk and we want to find the best x(t) that minimizes this functional:

Design a trajectory $x(t)$ such that $x(0) = a$, $x(T) = b$

$$x^{\star}(t) = \underset{x(t)}{\operatorname{argmin}} \int_0^T \mathcal{L}\left(\dddot{x}, \ddot{x}, \dot{x}, x, t\right) dt$$

$$\mathcal{L} = \left(\dddot{x}\right)^2$$

We write down the Euler Lagrange equations, and we see that many terms will simply disappear because they don't explicitly depend on $\dddot{x}$:

Euler-Lagrange:

$$\frac{\partial \mathcal{L}}{\partial x} - \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{x}}\right) + \frac{d^2}{dt^2}\left(\frac{\partial \mathcal{L}}{\partial \ddot{x}}\right) - \frac{d^3}{dt^3}\left(\frac{\partial \mathcal{L}}{\partial x^{(3)}}\right) = 0$$

What remains can be simplified into a sixth-order differential equation which can then be solved analytically. The result is this fifth-order polynomial in time:

$$x = c_5 t^5 + c_4 t^4 + c_3 t^3 + c_2 t^2 + c_1 t + c_0$$

What remains is the sub-problem of solving for the coefficients in this fifth-order polynomial.

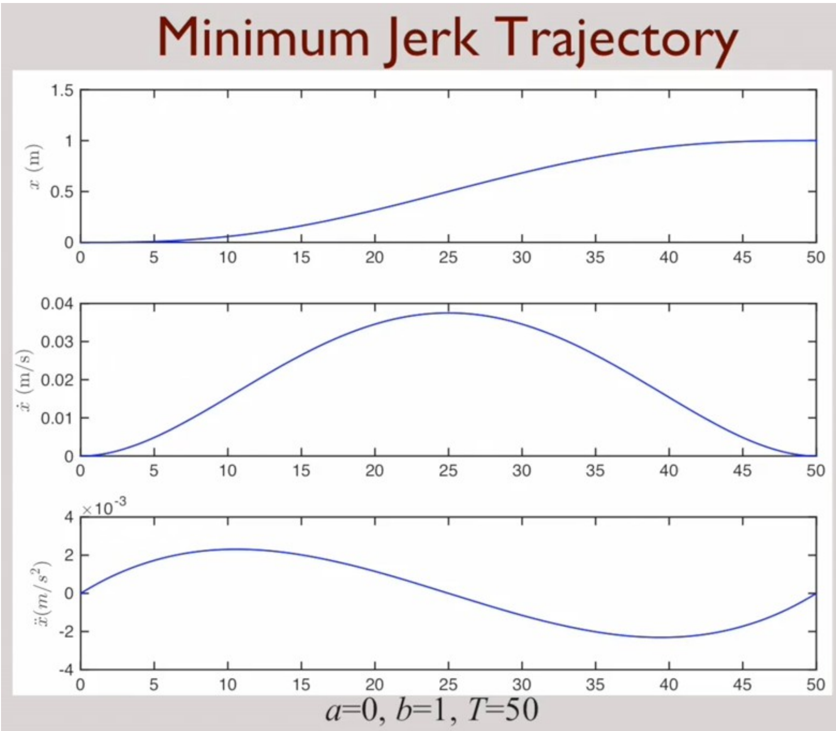
There are six such coefficients. In order to solve for these, we’ll need to specify additional boundary conditions. We've assumed that we know the position at the start and end times, now let's also assume that we know the velocities. In this case, we'll assume that these velocities are zero. Furthermore, let's assume we know the accelerations at times t=0 and t=T, again assuming these accelerations to be zero.

	Position	Velocity	Acceleration
t=0	A	0	0
t=T	b	0	0

The main object of this exercise is to ensure that we have as many boundary conditions as the number of constants we're trying to solve for. Each of these boundary conditions gives us an equation, and we can write down six equations in terms of the unknown constants and the known boundary conditions.

$$\begin{bmatrix} a \\ b \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ T^5 & T^4 & T^3 & T^2 & T & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 5T^4 & 4T^3 & 3T^2 & 2T & 1 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 20T^3 & 12T^2 & 6T & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} c_5 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix}$$

Solving for these constants is now a linear problem. This is what the minimum jerk solution looks like:



In this case, x varies from zero to one, over a total of fifty time-units. The velocity profile is bell shaped starting with zero-velocity, and ending with zero-velocity. It's uni-modal, reaching a peak velocity in the middle, i.e. at 25 time-units. The acceleration looks sinusoidal, starting with zero-acceleration, accelerating to a peak acceleration value, and then decelerating to a peak deceleration value before ending up at zero.

This basic idea can be extended to multiple dimensions. If we're trying to generate trajectories in the x-y plane, then we end up with two Euler Lagrange equations: one for the x-direction and one for the y-direction:

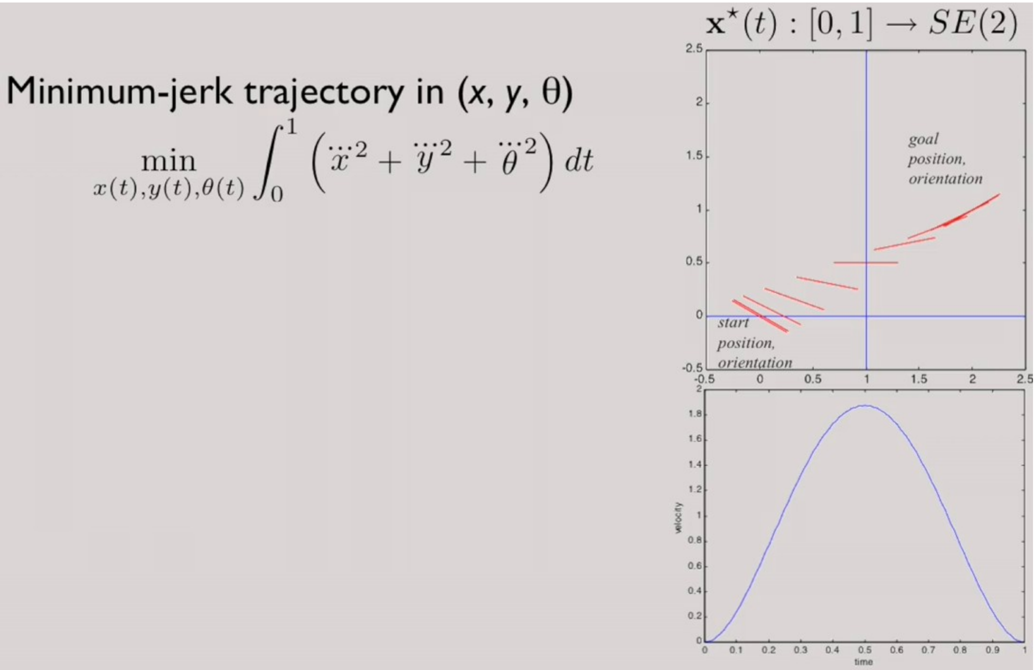
$$(x^\star(t), y^\star(t)) = \arg \min_{x(t),y(t)} \int_0^T \mathcal{L}\left(\dot{x}, \dot{y}, x, y, t\right) dt$$

Euler Lagrange Equation

Necessary condition satisfied by the "optimal" function

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

Here's how we might think about the minimum jerk trajectory in three dimensions:



Here we're looking at a trajectory in the x, y, and θ space. We're assuming that we're given a starting x and y position, a starting orientation, θ, and that we're given a final x, y, and θ. We're trying to find a trajectory that minimizes the functional given by $\ddot{x}^2 + \ddot{y}^2 + \dddot{\theta}^2$. If we use the version of the Euler Lagrange equations for multiple-dimensions, we end up with a very straightforward solution.

We essentially get three fifth-order polynomials, one in the x-direction, one in the y-direction, and one in the θ direction. By plugging in the boundary conditions, we get the on the right. We have the starting position and orientation, the end position and orientation, and a smooth transition from start to finish. The velocity profile is again bell shaped.

It turns out that this problem is also relevant to modelling human-manipulation tasks where:

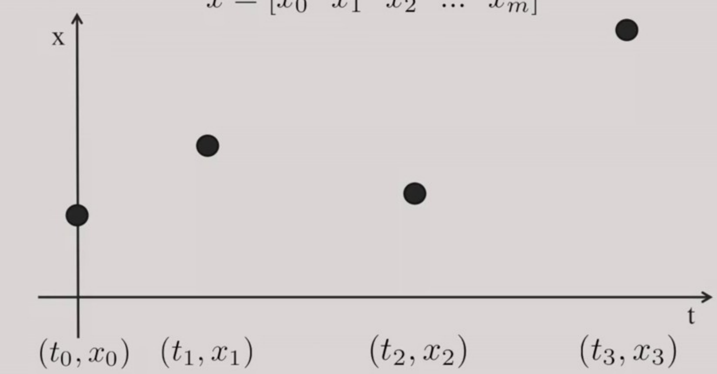
- Noise in the neural control-signal increases with the size of the control-signal.
- Rate-of-change of muscle fibre lengths is critical in relaxed, voluntary motions.

This goes back to a 1985 doctoral dissertation by Tamar Flash. And can be extended to multiple dimensions when two-handed manipulation tasks are considered.

In robotics, it's rarely sufficient to specify just the initial and final positions and orientations. In order to ensure that the trajectory is safe we often want to specify intermediate waypoints. Instead of thinking simply about trajectories with known start and end configurations, we want to think about trajectories that pass through intermediate waypoints.

Design a trajectory x(t) such that:

$$t = [t_0 \;\; t_1 \;\; t_2 \;\; \dots \;\; t_m]^T$$
$$x = [x_0 \;\; x_1 \;\; x_2 \;\; \dots \;\; x_m]^T$$



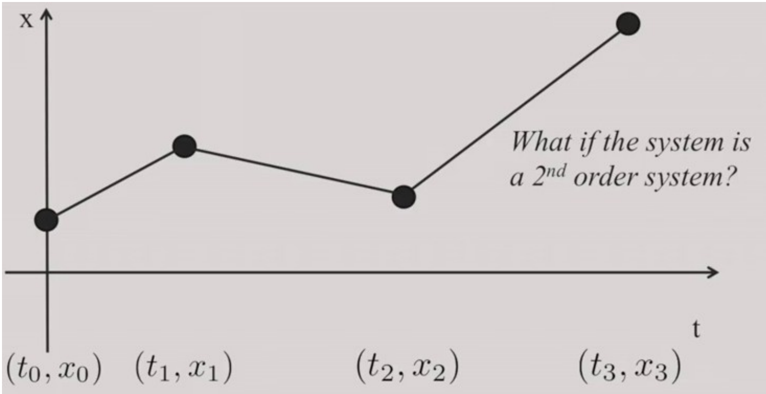
Let's consider the problem. We’re given a set of intermediate time points and positions that we require the trajectory to pass through. The time points are t0, t1, etc., and the corresponding positions are x0, x1, etc. In this case, we assume (m+1) time-points, and (m+1) corresponding positions

A simple way to approach this is to define piecewise continuous-trajectories with one for each time-segment.

Define piecewise continuous trajectory:

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

The obvious way of doing it is to simply connect the waypoints. But this is not very practical.



The result is continuous, but not differentiable. What if the system is a second-order system with inertia? Once we get started on the first segment, it's impossible for the second-order system to make the sharp turn at the kink.

In order to make the trajectory more amenable to higher-order systems, we want to insist that the trajectory is smooth and not have these kinks at the intermediate points. To see how to solve this problem, let's consider a minimum acceleration curve for second-order systems. We look to solve for this function in x(t):

$$\min_{x(t)} \left[\int_{t_0}^{t_1} (\ddot{x}^2) dt + \ldots + \int_{t_{m-1}}^{t_m} (\ddot{x}^2) dt \right]$$

The function now, consists of many terms, each corresponding to a particular segment of the trajectory. From the Euler Lagrange equations, if we solve this, we find that we get cubics for each segment:

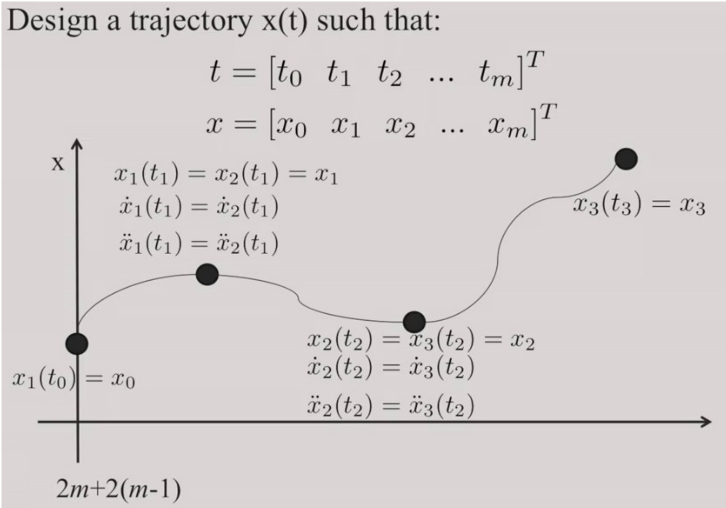
$$x(t) = \begin{cases} x_1(t) = c_{1,3}t^3 + c_{1,2}t^2 + c_{1,1}t + c_{1,0}, & t_0 \leq t < t_1 \\ x_2(t) = c_{2,3}t^3 + c_{2,2}t^2 + c_{2,1}t + c_{2,0}, & t_1 \leq t < t_2 \\ \ldots & \\ x_m(t) = c_{m,3}t^3 + c_{m,2}t^2 + c_{m,1}t + c_{m,0}, & t_{m-1} \leq t < t_m \end{cases}$$

The resulting function or curve is called a ***cubic spline***.

We now know the functional form of each segment, however, we don't have the constants.

There are 4m constants for this cubic spline, each corresponding to a degree of freedom. To determine these constants, we have to specify 4m different boundary-conditions or intermediate conditions.

Going back to our schematic, we're looking for a smooth curve:



We know where the intermediate points are, and that gives us 2m boundary-conditions. Further, we insist that the curve is smooth, more specifically, that it is differentiable up to two times at each of the intermediate points. That gives us an additional 2(m – 1) boundary-conditions. Finally, we insist that the curve starts and ends with specified velocities. In this case, both velocities are taken to be zero. That gives us an further two boundary conditions. In total, this gives us 4m constraints, allowing us to specify the 4m degrees-of-freedom associated with the undetermined constants.

This basic idea can be generalised to n^{th}-order systems. We have to specify appropriate boundary-conditions at the beginning and end. And we have to specify intermediate boundary-conditions. These intermediate boundary-conditions, specify continuity up to the 2(n – 1)-order derivitor.

This leads us to minimum-snap trajectories where n = 4. We're now ready to solve the motion planning problem for quadrotors using minimum snap trajectories.