

Lecture 7: Trajectory Optimisation Part III

Scribes: Daniel Filan and Davis Foote

7.1 Summary

This lecture will have two sections: first, an explanation of why and how to change the inner product implicitly used to define functional gradient descent, and secondly, a discussion of how student paper presentations will work, and why they will work that way.

7.2 Changing the Inner Product

For ease of understanding, we will think about our trajectories as discretised vectors, whose i^{th} component is the position in configuration space of the trajectory at the i^{th} point in (discrete) time. This vector will look something like

$$\zeta = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \vdots \\ q_N \end{bmatrix} \quad (7.1)$$

Since we won't consider changing the start or end points of the trajectory, we leave out the initial configuration q_0 and the final configuration q_{N+1} . We should also remember that each element of this vector is itself a vector, rather than simply a scalar.

So far, we have been using the Euclidean inner product, defined as

$$\langle \zeta_1, \zeta_2 \rangle_E = \zeta_1^\top \zeta_2 = q_{1,1}^\top q_{2,1} + \cdots + q_{1,N}^\top q_{2,N} \quad (7.2)$$

Why would we want to change this? Well, an inner product defines a norm by $\|\zeta\| = \sqrt{\langle \zeta, \zeta \rangle}$, and a norm defines a distance metric defined by $\text{distance}(\zeta_1, \zeta_2) = \|\zeta_1 - \zeta_2\|$. Therefore, we might ask ourselves whether the Euclidean metric induced by the Euclidean inner product is any good.

7.2.1 Why the Euclidean metric isn't good

Consider three trajectories of a robot with one degree of freedom:

DIAGRAM HERE

$$a = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, b = \begin{bmatrix} 0 \\ 0 \\ 10 \\ 0 \\ 0 \end{bmatrix}, \text{ and } c = \begin{bmatrix} 0 \\ 5 \\ 10 \\ 5 \\ 0 \end{bmatrix} \quad (7.3)$$

Which is closer to a : b or c ? Well,

$$\|a - b\|_E^2 = \langle b, b \rangle_E = 10 \times 10 = 100 \quad (7.4)$$

and

$$\|a - c\|_E^2 = \langle c, c \rangle_E = 5 \times 5 + 10 \times 10 + 5 \times 5 = 150 \quad (7.5)$$

Therefore, according to the Euclidean metric, b is closer to a than c is. However, intuitively, there's something wrong with this. a involves very smooth motion, b is super jerky, and c involves a bit of a jerk, but less so than b . The Euclidean metric can't notice this because it doesn't take the orderings of elements of the vector into account, and therefore can't talk about speeds, but it would be nice if we could take time into account somehow.

Can we give a more rigorous argument for why we should change metric?

7.2.1.1 Detour 1

The formula for gradient descent, as we know, is $\xi_{i+1} = \xi_i - (1/\alpha)\nabla_{\xi_i}\mathcal{U}$. But where does this come from?

Suppose instead that we approximate \mathcal{U} by its first order Taylor expansion, and try to minimise that. Then, we would use the rule

$$\xi_{i+1} = \arg \min_{\xi} \left\{ \mathcal{U}[\xi_i] + \nabla_{\xi_i}\mathcal{U}^\top (\xi - \xi_i) \right\} \quad (7.6)$$

However, we know that this Taylor expansion will be off if we get far away in norm from ξ_i , so we add a regularisation term to penalise this:

$$\xi_{i+1} = \arg \min_{\xi} \left\{ \mathcal{U}[\xi_i] + \nabla_{\xi_i}\mathcal{U}^\top (\xi - \xi_i) + \frac{1}{2}\alpha\|\xi - \xi_i\|_E^2 \right\} \quad (7.7)$$

This is a quadratic problem in ξ , so we can BLAH BLAH