Trajectory Optimization

Fundamentals of Robotics, Lecture 11

by Sergei Savin

Fall 2022

We can define state-space coordinates $\mathbf{x}$ for a mechanical system $\mathbf{H}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{g} = \tau$ as follows:

$$\mathbf{x} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} \tag{1}$$

Defining $\mathbf{u} = \tau$, $\mathbf{S}_q = [\mathbf{I} \;\; \mathbf{0}]$ and $\mathbf{S}_v = [\mathbf{0} \;\; \mathbf{I}]$ we get $\mathbf{q} = \mathbf{S}_q\mathbf{x}$ and $\dot{\mathbf{q}} = \mathbf{S}_v\mathbf{x}$. The state-space dynamics becomes:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \mathbf{S}_v\mathbf{x} \\ \mathbf{H}^{-1}(\mathbf{u} - \mathbf{C}\mathbf{S}_v\mathbf{x} - \mathbf{g}) \end{bmatrix} \tag{2}$$

We know how to control a nonlinear system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$ obtained from the manipulator equations, as long as we want point-to-point or trajectory following control, and the trajectory is provided. But how do we find a feasible trajectory?

Here are aspects of a typical trajectory problems:

1. Drive the system to the desired state $\mathbf{x}^*(t_f)$ by the time t_f.
2. Respect torque limits: $||u_i|| \leq u_{max}$.
3. Respect kinematic constraints: $\dot{q}_{min} \leq \dot{q}_i \leq \dot{q}_{max}$.
4. Respect joint limits: $q_{i,min} \leq q_i \leq q_{i,max}$.

We can formulate the trajectory planning as an *optimal control problem* (OCP):

$$
\begin{aligned}
\underset{\mathbf{x}(t),\mathbf{u}(t)}{\text{minimize}} \quad & \int_{t_0}^{t_f} l\left(\mathbf{x}(t), \mathbf{u}(t)\right) dt + l_f(\mathbf{x}(t_f)), \\
\text{subject to:} \quad & \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \\
& \mathbf{x}_{min} \leq \mathbf{x}(t) \leq \mathbf{x}_{max} \\
& \mathbf{u}_{min} \leq \mathbf{u}(t) \leq \mathbf{u}_{max}
\end{aligned} \tag{3}
$$

This is an optimization problem with continuous variables and there are no solvers that can solve it (in a general case). So, our method is to replace the continuous time variables with a finite number of parameters. This is called *transcription*.

Types of transcription

There are a number of popular ways to transcribe the trajectory. They are often divided into *collocation* and *shooting* methods.

Shooting methods transcribe $\mathbf{u}(t)$ via finite number of parameters (spline coefficients or values of $\mathbf{u}$ at particular time intervals), and then compute $\mathbf{x}(t)$ via integration.

Collocation methods transcribe both $\mathbf{u}(t)$ and $\mathbf{x}(t)$ via finite number of parameters (spline coefficients for $\mathbf{u}(t)$ and $\mathbf{x}(t)$), and use dynamics equations as constraints.

Below is an example of a simple direct collocation, where the $\mathbf{x}(t)$ and $\mathbf{u}(t)$ are discretized at time nodes $t_1,\ ...,\ t_n$ as $(\mathbf{x}_1, \mathbf{u}_1),\ ...,\ (\mathbf{x}_n, \mathbf{u}_n)$.

$$\underset{\substack{\mathbf{x}_1,\ ...,\ \mathbf{x}_n, \\ \mathbf{u}_1,\ ...,\ \mathbf{u}_{n-1}}}{\text{minimize}} \quad \sum_{i=1}^{n-1} \left(\mathbf{x}_i^\top \mathbf{Q}_i \mathbf{x}_i + \mathbf{u}_i^\top \mathbf{R}_i \mathbf{u}_i \right) + (\mathbf{x}_n - \mathbf{x}_d)^\top \mathbf{Q}_n (\mathbf{x}_n - \mathbf{x}_d),$$

$$\text{subject to:} \quad \frac{\mathbf{x}_{i+1} - \mathbf{x}_i}{\Delta t_i} = \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i)$$

$$\mathbf{x}_{min} \leq \mathbf{x}_i \leq \mathbf{x}_{max}$$

$$\mathbf{u}_{min} \leq \mathbf{u}_i \leq \mathbf{u}_{max}$$

Approximating derivative as a difference does not strike us as highly accurate in this case. We can do a little better with the following constraint:

$$\mathbf{x}_{i+1} = \frac{\Delta t_i}{2} (\mathbf{f}(\mathbf{x}_i, \mathbf{u}_i) + \mathbf{f}(\mathbf{x}_{i+1}, \mathbf{u}_{i+1})) \tag{4}$$

Direct collocation - linear discrete systems

In case of a linear discrete system $\mathbf{x}_{i+1} = \mathbf{A}_i \mathbf{x}_i + \mathbf{B}_i \mathbf{u}_i$, direct collocation becomes the obvious choice, equivalent to MPC. With quadratic cost it becomes a finite-time version of LQR:

$$\begin{aligned}
\underset{\substack{\mathbf{x}_1,\ \dots,\ \mathbf{x}_n, \\ \mathbf{u}_1,\ \dots,\ \mathbf{u}_{n-1}}}{\text{minimize}} \quad & \sum_{i=1}^{n-1} \left(\mathbf{x}_i^\top \mathbf{Q}_i \mathbf{x}_i + \mathbf{u}_i^\top \mathbf{R}_i \mathbf{u}_i \right) + (\mathbf{x}_n - \mathbf{x}_d)^\top \mathbf{Q}_n (\mathbf{x}_n - \mathbf{x}_d), \\
\text{subject to:} \quad & \mathbf{x}_{i+1} = \mathbf{A}_i \mathbf{x}_i + \mathbf{B}_i \mathbf{u}_i \\
& \mathbf{x}_{min} \leq \mathbf{x}_i \leq \mathbf{x}_{max} \\
& \mathbf{u}_{min} \leq \mathbf{u}_i \leq \mathbf{u}_{max}
\end{aligned}$$

Note, that if there were no inequality constraints, this problem would be solved analytically.

Single shooting

Below is an example of a simple single shooting, where the $\mathbf{u}(t)$ is discretized at time nodes $t_1, \ ..., \ t_n$ as $\mathbf{u}_1, \ ..., \ \mathbf{u}_n$.

$$
\begin{aligned}
\underset{\mathbf{u}_1, \ ..., \ \mathbf{u}_n}{\text{minimize}} \quad & \sum_{i=1}^{n-1} ||\mathbf{u}_i|| + (\mathbf{x}(t_f) - \mathbf{x}_d)^\top \mathbf{Q}_n (\mathbf{x}(t_f) - \mathbf{x}_d), \\
\text{subject to:} \quad & \mathbf{x} = \int_{t_0}^{t_f} \mathbf{f}(\mathbf{x}, \mathbf{u}) dt \\
& \mathbf{u}_{min} \leq \mathbf{u}_i \leq \mathbf{u}_{max}
\end{aligned}
\tag{5}
$$

Note that here it is possible to use sophisticated integration schemes, but also it is hard to impose state constraints (joint limits, joint velocity limits, obstacle avoidance, etc.).

Read more

Matthew Kelly Intro Trajectory Optimization.

Lecture slides are available via Moodle.

You can help improve these slides at:
github.com/SergeiSa/Fundamentals-of-robotics-2022

Check Moodle for additional links, videos, textbook suggestions.