Mathematic Modeling of Optimal Control

1 Introduction

The equation of motion for the constrained robotic manipulator with n degrees of freedom, considering the contact force and the constraints, is given in the joint space as follows:

$$M(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + G(\boldsymbol{q}) = \boldsymbol{u} - F_d\dot{\boldsymbol{q}} - F_\mu \operatorname{sgn}(\dot{\boldsymbol{q}}) \tag{1}$$

where:

- $\boldsymbol{q} \in \mathbb{R}^{n\times 1}$ denotes the joint angles or link displacements of the manipulator
- $M(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is the robot inertia matrix which is symmetric and positive definite
- $C(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{n\times n}$ contains the centripetal and Coriolis terms
- $G(\boldsymbol{q}) \in \mathbb{R}^{n\times 1}$ are the gravity terms
- $\boldsymbol{u} \in \mathbb{R}^{n\times 1}$ denotes the torque or the force
- $F_d \in \mathbb{R}^{n\times n}$ is the damping matrix
- $F_\mu \in \mathbb{R}^{n\times n}$ contains the frictional terms

Let's define $\boldsymbol{x}_1 = \boldsymbol{q}$, $\boldsymbol{x}_2 = \dot{q}$, $\boldsymbol{x} = \binom{\boldsymbol{q}}{\dot{\boldsymbol{q}}}$, and we can rewrite everything as follows:

$$\begin{cases} \dot{\boldsymbol{x}}_1 = \boldsymbol{x}_2 \\ \dot{\boldsymbol{x}}_2 = -M^{-1}C\boldsymbol{x}_2 + M^{-1}(\boldsymbol{u} - G - F_d\boldsymbol{x_2} - F_\mu \operatorname{sgn} \boldsymbol{x_2}) \end{cases} \tag{2}$$

2 Optimal Control

The optimal control will be designed for the trajectory tracking case where the objective is to obtain that the tracking error of the states converges to zero, that is, the states of the robotic arm converge to the desired trajectories.

Reference trajectory Let's say we wanna reach a trajectory specified as $\boldsymbol{q}_d(t)$ and we define $x_d = \binom{q_d}{\dot{q}_d}$. So our tracking error $\tilde{x}$ will be:

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} - \boldsymbol{x}_d \tag{3}$$

With that in mind we can write our cost for this system as follows:

$$J = \tilde{\boldsymbol{x_f}}^T H \tilde{\boldsymbol{x_f}} + \int_{t_i}^{t_f} \left(\tilde{\boldsymbol{x}}^T Q \tilde{\boldsymbol{x}} + \boldsymbol{u}^T R \boldsymbol{u}\right) \tag{4}$$

The Problem itself We can now rewind what we've said until now and define our problem in terms of differential equations.

$$\min_{\boldsymbol{u},\boldsymbol{x}} J(\boldsymbol{u}, \boldsymbol{x}) \quad \text{s.t.} \quad \begin{cases} \dot{\boldsymbol{x}}_1 = \boldsymbol{x}_2 \\ \dot{\boldsymbol{x}}_2 = -M^{-1}C\boldsymbol{x}_2 + M^{-1}(\boldsymbol{u} - G - F_d\boldsymbol{x_2} - F_\mu \operatorname{sgn} \boldsymbol{x_2}) \end{cases} \tag{5}$$

3 What we do

We start from the following set of differential equations:

$$\begin{cases} \dot{\boldsymbol{x}}_1 = \boldsymbol{x}_2 \\ \dot{\boldsymbol{x}}_2 = -M^{-1}C\boldsymbol{x}_2 + M^{-1}(\boldsymbol{u} - G - F_d\boldsymbol{x_2} - F_\mu \operatorname{sgn} \boldsymbol{x_2}) \\ \dot{J} = (\boldsymbol{x} - \boldsymbol{x}_d)^T Q(\boldsymbol{x} - \boldsymbol{x}_d) + \boldsymbol{u}^T R\boldsymbol{u} \end{cases} \implies \begin{pmatrix} \dot{\boldsymbol{x}}_1 \\ \dot{\boldsymbol{x}}_2 \\ \dot{J} \end{pmatrix} = f(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{u}, t) \quad (6)$$

We solve symbolically the differential equation above with RK4 for a single time ste: p, so we get the following function F:

$$\begin{pmatrix} \boldsymbol{x}_f \\ J_f \end{pmatrix} = F(\boldsymbol{x}_i, \boldsymbol{u}_i, J_i, t_i) \quad (7)$$

With that, we can integrate from the initial conditions up to the last time step and get $J(u_0, u_1, ..., u_{N-1})$ and $\boldsymbol{x}$ for each time step, so we can start the optimization imposing the dynamics boundaries.
We also make sure that all our variables stay into the limits specified in the URDF.
Then, we start an optimization algorithm that minimize J and respect the boundaries we defined before. The result will be $\boldsymbol{u}_{opt}$ and $\boldsymbol{x}_{opt}$.