Introduction to 2D frictional contact model

1 Modeling

The modeling process for rigid body dynamics with contact and friction has been thoroughly discussed in [1, 2, 3]. The basic idea to simulate such model is to use the time-stepping scheme to discretize the system. Here we first consider the unilateral and inelastic ($\epsilon = 0$) contact. Accoding to all three papers, the discrete model can be written as the fillowing form.

$$M(q_{k+1})(\dot{q}_{k+1} - \dot{q}_k) = n(q_k)c_{n,k+1} + D(q_k)\beta_{k+1} + \delta t[-\nabla V(q_k) + C(q_k, \dot{q}_k)\dot{q}_k + u_k] \quad (1)$$

$$q_{k+1} - q_k = \dot{q}_{k+1}\delta t \quad (2)$$

$$0 \le c_{n,k+1} \perp n(q_k)^T q_{k+1} + \alpha \ge 0 \quad (3)$$

$$0 \le \beta_{k+1} \perp \lambda_{k+1}\mathbf{1} + D(q_k)^T \dot{q}_{k+1} \ge 0 \quad (4)$$

$$0 \le \lambda_{k+1} \perp \mu c_{n,k+1} - \mathbf{1}^T \beta_{k+1} \ge 0 \quad (5)$$

$M(q)$ is mass matrix; the column of $D(q)$ represents the space tangent to the contact; β is the friction force at each direction defined by the column of $D(q)$; $n(q)$ is the surface normal of contact plane, which can be defined as $n(q) = \nabla f(q)$, where $f(q) \ge 0$ refers to the admissible region. $V(q_k)$ is the potential energy. δt is time step, and $\mathbf{1}$ is ones vector with appropriate dimension.

Three points are worth noticing.

1. (3) is from the discretization of the following condition:

   $$f(q_{k+1}) \approx f(q_k) + J(q_k)(q_{k+1} - q_k) \ge 0 \quad \Rightarrow \quad J(q_k)q_{k+1} \ge J(q_k)q_k - f(q_k) = -\alpha$$

   where $J(q_k) = \nabla f(q_k)^T = n(q_k)^T$.

2. The variable $\dot{q}_{k+1}$ in (4) is the relative velocity at contact point. In our model, all contact surface are fixed, which means the relative velocity is the actual velocity of the end effector. Here the model has two prismatic joint, we don't need to map the velocity using Jacobian. If the robot has revolute joints, we have to use Jacobian to find the velocity with respect to the world frame.

3. The column of $D(q)$ represents the tangent space of contact surface. The rigorous description of friction force should be using a friction cone. For computational reasons the polyhedral cone is adopted to approximate the real smooth friction cone. In 2D space, $D(q)$ only has two columns, which is enough to span the tangent space.

The goal is to use MPC to steer the robot to the desired point (this is the first step, the future work will be tajectory tracking). We choose the state as $x = \begin{bmatrix} q \\ \dot{q} \end{bmatrix}$ and define the objective function as

$$J(x, u) = x_f^T Q_f x_f + \sum_{i=0}^{N-1} x^T Q x + u^T R u$$

Then using (1) and (2), we define the system dynamics as follows

$$\begin{bmatrix} I & -\delta t I \\ 0 & I \end{bmatrix} \begin{bmatrix} q_{k+1} \\ \dot{q}_{k+1} \end{bmatrix} = \begin{bmatrix} q_k \\ \dot{q}_k \end{bmatrix} + \begin{bmatrix} 0 \\ M(q_{k+1})^{-1}\delta t \end{bmatrix} u + \begin{bmatrix} 0 & 0 & 0 \\ M(q_{k+1})^{-1}n(q_k) & M(q_{k+1})^{-1}D(q_k) & 0 \end{bmatrix} \begin{bmatrix} c_{n,k+1} \\ \beta_{k+1} \\ \lambda_{k+1} \end{bmatrix}$$
$$+ \begin{bmatrix} 0 \\ M(q_{k+1})^{-1}\delta t \left[C(q_k, \dot{q}_k)\dot{q}_k + \nabla V(q_k)\right] \end{bmatrix} \tag{6}$$

We define $\gamma_{k+1} = \begin{bmatrix} c_{n,k+1} \\ \beta_{k+1} \\ \lambda_{k+1} \end{bmatrix}$, then (6) can be written as

$$x_{k+1} = Ax_k + Bu_k + C\gamma_{k+1} + F \tag{7}$$

Back to (3)-(5), we can write them in the following matrix form:

$$0 \preceq \begin{bmatrix} n(q_k)^T & 0 \\ 0 & D(q_k)^T \\ 0 & 0 \end{bmatrix} \begin{bmatrix} q_k \\ \dot{q}_{k+1} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} \\ \mu & -\mathbf{1}^T & 0 \end{bmatrix} \begin{bmatrix} c_{n,k+1} \\ \beta_{k+1} \\ \lambda_{k+1} \end{bmatrix} + \begin{bmatrix} \alpha \\ 0 \\ 0 \end{bmatrix} \perp \begin{bmatrix} c_{n,k+1} \\ \beta_{k+1} \\ \lambda_{k+1} \end{bmatrix} \succeq 0 \tag{8}$$

Since our simple model is a two-link 2D robot shown as Figure 1, the potential energy and Coriolis force are both zero, which means $F = 0$ in (6).

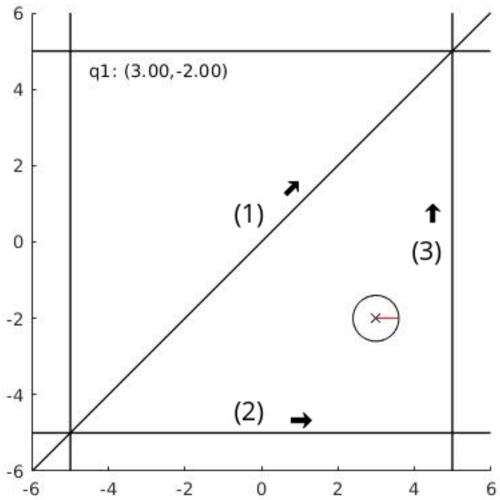

Figure 1: 2D robot model with friction

So the frictional contact with one surface is well defined by (8). The contact description with all three surfaces in Figure 1 is only a repetition of (8). For convenience, we number the contact surface and define the first column in matrix D_i for each surface as Figure 1. Then we write contact force, friction force and dual variable together.

$$0 \leq \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} \perp \begin{bmatrix} \nabla f_1(q_k)^T \\ \nabla f_2(q_k)^T \\ \nabla f_3(q_k)^T \end{bmatrix} q_{k+1} - \begin{bmatrix} f_1(q_k) \\ f_2(q_k) \\ f_3(q_k) \end{bmatrix} \geq 0$$

$$
0 \leq \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} \perp \begin{bmatrix} D_1^T \\ D_2^T \\ D_3^T \end{bmatrix} \dot{q}_{k+1} + \begin{bmatrix} \mathbf{1}_2 & & \\ & \mathbf{1}_2 & \\ & & \mathbf{1}_2 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} \geq 0
$$

$$
0 \leq \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} \perp \begin{bmatrix} \mu & & \\ & \mu & \\ & & \mu \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} - \begin{bmatrix} \mathbf{1}_2^T & & \\ & \mathbf{1}_2^T & \\ & & \mathbf{1}_2^T \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} \geq 0
$$

So we have

$$
\begin{bmatrix} c \\ \beta \\ \lambda \end{bmatrix} \perp \begin{bmatrix} \nabla \tilde{f}^T & 0 \\ 0 & \tilde{D} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} q_{k+1} \\ \dot{q}_{k+1} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \tilde{\mathbf{1}} \\ \mu I & -\tilde{\mathbf{1}}^T & 0 \end{bmatrix} \begin{bmatrix} c \\ \beta \\ \lambda \end{bmatrix} + \begin{bmatrix} \tilde{f} \\ 0 \\ 0 \end{bmatrix} \tag{9}
$$

Let $\gamma = \begin{bmatrix} c^T & \beta^T & \lambda^T \end{bmatrix}^T$, we can write (9) as

$$
0 \leq \gamma_{k+1} \perp Dx_{k+1} + E\gamma_{k+1} + \tilde{\alpha} \geq 0
$$

Therefore, the MPC problem can be written as

$$
\begin{aligned}
\min \quad & J(x,u) = x_f^T Q_f x_f + \sum_{k=0}^{N-1} x_k^T Q x_k + u_k^T R u_k \\
\text{s.t.} \quad & x_{k+1} = Ax_k + Bu_k + C\gamma_{k+1} \quad k = 0, \dots, N-1 \\
& 0 \leq \gamma_{k+1} \perp Dx_{k+1} + E\gamma_{k+1} + \tilde{\alpha} \geq 0, \quad k = 0, \dots\ N-1 \\
& x_{min} \leq x_k \leq x_{max}, \quad k = 0, \dots, N \\
& u_{min} \leq u_k \leq u_{max}, \quad k = 0, \dots, N-1
\end{aligned} \tag{10}
$$

If we want to control the robot to a desired state, then we can simply modify the objective function as follows.

$$
J(x,u) = (x_f - x_d)^T Q_f (x_f - x_d) + \sum_{k=0}^{N-1} (x_k - x_d)^T Q (x_k - x_d) + u_k^T R u_k
$$

2 Code Specification

To better manage code and avoid confusion about variable definitions, some specifications are made here to facilitate code implementation.

We use big M method to represent complementarity constraints. (10) becomes

$$
\begin{aligned}
\min_{x,u} \quad & x_N Q_f x_N + \sum_{k=1}^{N-1} x_k^T Q x_k + u_k^T R u_k \\
\text{subject to} \quad & x_{k+1} = Ax_k + Bu_k + C\gamma_k, \quad k = 0, \dots, N-1 \\
& 0 \leq Dx_{k+1} + E\gamma_{k+1} + \tilde{\alpha}_k \leq M(1 - z_k), \quad k = 0, \dots, N-1 \\
& 0 \leq \gamma_{k+1} \leq Mz_k, \quad k = 0, \dots, N-1 \\
& x_{min} \preceq x_k \preceq x_{max}, \quad k = 0, \dots, N \\
& u_{min} \preceq u_k \preceq u_{max}, \quad k = 0, \dots, N-1 \\
& z_k = \{0, 1\}, \quad k = 0, \dots, N-1
\end{aligned} \tag{11}
$$

where $M = \max\{Dx_{k+1} + E\gamma_{k+1} + \tilde{\alpha}_k, \gamma_{k+1}\}(k = 0, \dots, N-1)$.

So we get an MIQP problem. We choose the decision variable as $X = \begin{bmatrix} x^T & u^T & \lambda^T & z^T \end{bmatrix}^T$. x_0 is known before we solve the problem. Notice that in (11) we have system dynamics, so it is natural to think why don't we replace the states with other variables so that the total number of decision variables will reduce a lot. It is correct, but replacing the states will bring a denser matrix which will actually take longer time to solve. Increasing the number of decision variables leads to sparse matrices, and it is a trade-off. According to previous experience, we keep the states as part of decision variables. Rewrite the objective, we have

$$J = x_0^T Q x_0 + \begin{bmatrix} x_1 \\ \vdots \\ x_N \\ u_0 \\ \vdots \\ u_{N-1} \end{bmatrix}^T \begin{bmatrix} Q & & & & & \\ & \ddots & & & & \\ & & Q_f & & & \\ & & & R & & \\ & & & & \ddots & \\ & & & & & R \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_N \\ u_0 \\ \vdots \\ u_{N-1} \end{bmatrix} := \tilde{X}^T S \tilde{X} + x_0^T Q x_0$$

Adding γ and z, we have

$$J = X^T \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix} X + x_0^T Q x_0 := X^T S_{mat} X + x_0^T Q x_0 \tag{12}$$

Using the system dynamics, we have

$$-\begin{bmatrix} A \\ 0 \\ \vdots \\ 0 \end{bmatrix} x_0 = \begin{bmatrix} -I & 0 & \cdots & 0 & 0 & B & 0 & \cdots & 0 & C & 0 & \cdots & 0 \\ A & -I & \cdots & 0 & 0 & 0 & B & \cdots & 0 & 0 & C & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A & -I & 0 & 0 & \cdots & B & 0 & 0 & \cdots & C \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_N \\ u_0 \\ \vdots \\ u_{N-1} \\ \gamma_1 \\ \vdots \\ \gamma_N \end{bmatrix}$$

$$\Rightarrow \quad \begin{bmatrix} \tilde{A} & 0 \end{bmatrix} X - b_{mat} x_0 := A_{mat} X - b_{mat} x_0 = 0 \tag{13}$$

Gurobi has the interface for the lower and upper bound of variables, so here we don't need to transform the bound constraints.

$$x_{min} \preceq x_k \preceq x_{max}, \quad k = 0, \dots, N \tag{14a}$$
$$u_{min} \preceq u_k \preceq u_{max}, \quad k = 0, \dots, N-1 \tag{14b}$$
$$\gamma_{k+1} \succeq 0, \quad k = 0, \dots, N-1 \tag{14c}$$
$$z_k = \{0, 1\}, \quad k = 0, \dots, N-1 \tag{14d}$$

Then we combine and transform the rest constraints.

$$Dx_{k+1} + E\gamma_{k+1} + \tilde{\alpha}_k \succeq 0, \quad k = 0, \dots, N-1 \tag{14e}$$
$$Dx_{k+1} + E\gamma_{k+1} + \tilde{\alpha}_k \preceq M(1 - z_k), \quad k = 0, \dots, N-1 \tag{14f}$$
$$\gamma_{k+1} \preceq M z_k, \quad k = 0, \dots, N-1 \tag{14g}$$

(14e) can be written as

$$\begin{bmatrix} \begin{bmatrix} D & & \\ & \ddots & \\ & & D \end{bmatrix} & 0 & \begin{bmatrix} E & & \\ & \ddots & \\ & & E \end{bmatrix} & 0 \end{bmatrix} \begin{bmatrix} x \\ u \\ \gamma \\ z \end{bmatrix} + \begin{bmatrix} \tilde{\alpha}_0 \\ \vdots \\ \tilde{\alpha}_{N-1} \end{bmatrix} \succeq 0 \Rightarrow D_{mat}X + D_{lim} \succeq 0 \tag{15}$$

(14f) can be written as

$$\begin{bmatrix} \begin{bmatrix} D & & \\ & \ddots & \\ & & D \end{bmatrix} & 0 & \begin{bmatrix} E & & \\ & \ddots & \\ & & E \end{bmatrix} & MI \end{bmatrix} \begin{bmatrix} x \\ u \\ \gamma \\ z \end{bmatrix} \preceq M\mathbf{1} - \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_{N-1} \end{bmatrix} \Rightarrow E_{mat}X \preceq E_{lim} \tag{16}$$

(14g) can be written as

$$\begin{bmatrix} 0 & 0 & I & -MI \end{bmatrix} \begin{bmatrix} x \\ u \\ \gamma \\ z \end{bmatrix} \preceq 0 \Rightarrow F_{mat}X \preceq 0 \tag{17}$$

To sum up, using (12), (12), (15)-(17), we obtain the model that can be optimized by Gurobi.

$$\begin{aligned} \min_{X} \quad & X^T S_{mat} X + x_0^T Q x_0 \\ \text{subject to} \quad & A_{mat}X - b_{mat}x_0 = 0 \\ & D_{mat}X + D_{lim} \succeq 0 \\ & E_{mat}X \preceq E_{lim} \\ & F_{mat}X \preceq 0 \\ & x_{min} \preceq x_k \preceq x_{max}, \quad k = 0, \ldots, N \\ & u_{min} \preceq u_k \preceq u_{max}, \quad k = 0, \ldots, N-1 \\ & \gamma_{k+1} \succeq 0, \quad k = 0, \ldots, N-1 \\ & z_k = \{0, 1\}, \qquad k = 0, \ldots, N-1 \end{aligned} \tag{18}$$

For the tracking issue, the only change is the objective function.

$$\begin{aligned} J &= (x_n - x_d)^T Q_f (x_n - x_d) + \sum_{k=1} (x_k - x_d)^T Q (x_k - x_d) + u_k^T R u_k \\ &= X^T S_{mat} X - 2\tilde{x}_d^T \tilde{Q} x + \tilde{x}_d^T \tilde{Q} \tilde{x}_d + (x_d - x_0)^T Q (x_d - x_0) \\ &= X^T S_{mat} X - 2 \begin{bmatrix} \tilde{x}_d^T \tilde{Q} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ u \\ \gamma \\ z \end{bmatrix} + \tilde{x}_d^T \tilde{Q} \tilde{x}_d + (x_d - x_0)^T Q (x_d - x_0) \\ &= X^T S_{mat} X + 2\tilde{x}_d^T d_{mat}^T X + \tilde{x}_d^T \tilde{Q} \tilde{x}_d + (x_d - x_0)^T Q (x_d - x_0) \end{aligned} \tag{19}$$

where $\tilde{x}_d = \begin{bmatrix} x_d \\ \vdots \\ x_d \end{bmatrix}$ and $Q = \begin{bmatrix} Q & & \\ & \ddots & \\ & & Q \end{bmatrix}$.

3 Some simulation results

Figure 2 and Figure 3 shows contact and firction force undering the penality $Q = Q_f = I, R = 10I$ for $N = 5$ and $N = 20$ respectively. The goal is to make the robot move from $(-2, -4)$ to $(3, -5)$ which actually cannot be reached.

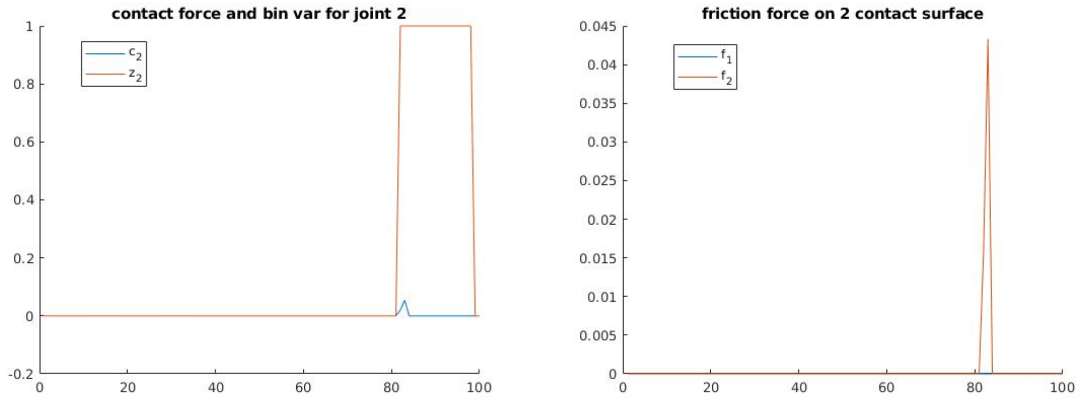

Figure 2: contact and friction force for $N = 5$

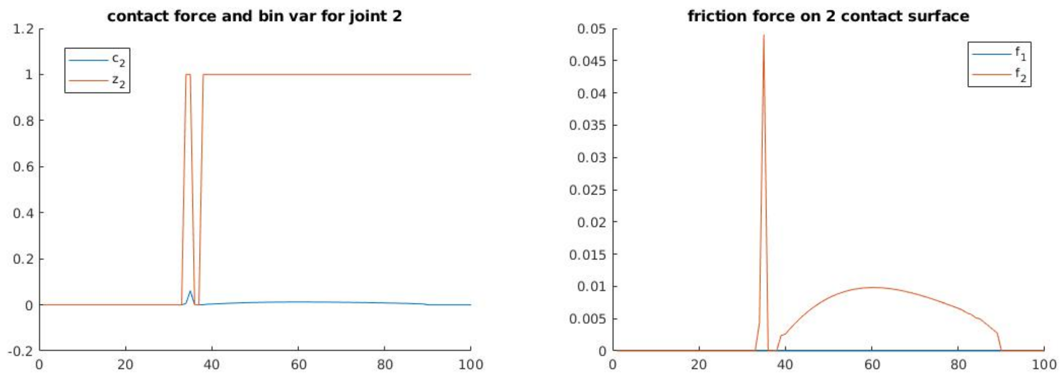

Figure 3: contact and friction force for $N = 20$

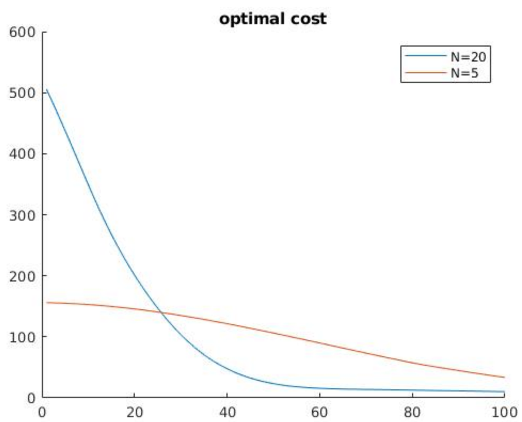

Figure 4: optimal cost for $N = 5$ and $N = 20$

The big R matrix means larger penalty on input, which means we want the robot to use friction as much as possible to stop. The difference is the prediction horizon. Apparently, the longer horizon will make more use of the friction, and the optimal cost is less than the shorter horizon after a few steps. This shows the advantage of MPC. If we predict further, the better solution we can get. Figure 4 shows the optimal cost of two cases. Of course there will be more interesting comparision; we only list one here.

References

[1] D.E. Stewart and J.C. Trinkle. *An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction*. International Journal for Numerical Methods in Engineering, 39(15):26732691, 1996.

[2] M. Anitescu and F.A. Potra. *Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems* Nonlinear Dynamics, 14(3):231247, 1997.

[3] D.E. Stewart *Rigid-body dynamics with friction and impact*. SIAM, 42(1):339, 2000