Some standard for using big M method

To fully test the big M method and avoid confusion about variable definition, here we define some programming standard including the problem formulation and test items to better manage the code.

This document contains two part. The first part introduces the problem formulation and variable definition. Two formulation methods will be introduced: formulation with substitution and without substitution. The main difference is the number of variables and the relative density of corresponding matrices. The second part stipulates the test items including cost time, optimal solution, etc.

1 Problem Formulation

The original problem is an QPCC which shows as follows.

$$\begin{aligned} \min_{x,u} \quad & x_N Q_f x_N + \sum_{k=1}^{N-1} x_k^T Q x_k + u_k^T R u_k \\ \text{subject to} \quad & x_{k+1} = Ax_k + Bu_k + C\lambda_k, \quad k = 0, \dots, N-1 \\ & 0 \le Dx_{k+1} + \alpha_k \perp \lambda_k \ge 0, \quad k = 0, \dots, N-1 \\ & x_{min} \preceq x_k \preceq x_{max}, \quad k = 0, \dots, N \\ & u_{min} \preceq u_k \preceq u_{max}, \quad k = 0, \dots, N-1 \end{aligned} \tag{1}$$

where $\alpha_k = J(q_k)q_k - f(q_k)$ is a constant and λ_k represents the contact force. No friction is considered here.

Using big M method to replace complementarity constraints, we obtain

$$\begin{aligned} \min_{x,u} \quad & x_N Q_f x_N + \sum_{k=1}^{N-1} x_k^T Q x_k + u_k^T R u_k \\ \text{subject to} \quad & x_{k+1} = Ax_k + Bu_k + C\lambda_k, \quad k = 0, \dots, N-1 \\ & 0 \le Dx_{k+1} + \alpha_k \le M(1 - z_k), \quad k = 0, \dots, N-1 \\ & 0 \le \lambda_k \le Mz_k, \quad k = 0, \dots, N-1 \\ & x_{min} \preceq x_k \preceq x_{max}, \quad k = 0, \dots, N \\ & u_{min} \preceq u_k \preceq u_{max}, \quad k = 0, \dots, N-1 \\ & z_k = \{0, 1\}, \qquad k = 0, \dots, N-1 \end{aligned} \tag{2}$$

where $M = \max\{Dx_{k+1} + \alpha_k, \lambda_k\}, k = 1, \dots, N-1$.

Formulation with substitution measn using the system dynamics to rearrange the objective function and only keeping the input u and contact force λ as decision variables. Formulation without substitution means keeping states x, input u and contact force λ as decision variables. The main

difference as, illustrated before, is the number of variables and the density of corresponding matrices. The main idea behind the optimization problem is MPC. The number of variables depends on the system dimension and the prediction steps. As the prediction steps increase, the number of variables will grow linearly. The formulation without substitution will have more $6N$ variables than with substitution where N is prediction steps. Usually we have $N = 10 \sim 20$, which we think two methods will have a big difference.

Besides, the matrix density is also an issue. For example, the objective matrix of formulation without substitution is merely a block diagonal matrix, while the formulation with substitution has a dense objective matrix, which is not sparse. This may affect the computation time and accuracy. Therefore, we will test both methods and compare the result.

1.1 Formulation without substitution

The decision variables in this method is $X = \begin{bmatrix} x^T & u^T & \lambda^T & z^T \end{bmatrix}^T$, where z is the binary vector introduced for complementarity constraints. x_0 is the initial state and is known.

We ignore the contact force and binary vector first, the objective can be written as

$$J' = x_0^T Q x_0 + \begin{bmatrix} x_1 \\ \vdots \\ x_N \\ u_0 \\ \vdots \\ u_{N-1} \end{bmatrix}^T \begin{bmatrix} Q & & & & & \\ & \ddots & & & & \\ & & Q_f & & & \\ & & & R & & \\ & & & & \ddots & \\ & & & & & R \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_N \\ u_0 \\ \vdots \\ u_{N-1} \end{bmatrix} := \tilde{X}^T S \tilde{X} + x_0^T Q x_0$$

Adding contact force λ and binary vector z, we have

$$J = X^T \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix} X + x_0^T Q x_0 := X^T S_{mat} X + x_0^T Q x_0 \tag{3}$$

Using the system dynamics, we have

$$-\begin{bmatrix} A \\ 0 \\ \vdots \\ 0 \end{bmatrix} x_0 = \begin{bmatrix} -I & 0 & \cdots & 0 & 0 & B & 0 & \cdots & 0 & C & 0 & \cdots & 0 \\ A & -I & \cdots & 0 & 0 & 0 & B & \cdots & 0 & 0 & C & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A & -I & 0 & 0 & \cdots & B & 0 & 0 & \cdots & C \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_N \\ u_0 \\ \vdots \\ u_{N-1} \\ \lambda_0 \\ \vdots \\ \lambda_{N-1} \end{bmatrix}$$

$$\Rightarrow \quad \begin{bmatrix} \tilde{A} & 0 \end{bmatrix} X - b_{mat} x_0 := A_{mat} X - b_{mat} x_0 = 0 \tag{4}$$

Then we focus on the inequality constraints. Notice that (2) is actually an MIQP and we use Gurobi to solve it. Gurobi has the interface for the lower and upper bound of variables, so here we don't need to transform the bound constraints

$$x_{min} \preceq x_k \preceq x_{max}, \quad k = 0, \ldots, N \tag{5a}$$
$$u_{min} \preceq u_k \preceq u_{max}, \quad k = 0, \ldots, N-1 \tag{5b}$$
$$\lambda_k \succeq 0, \quad k = 0, \ldots, N-1 \tag{5c}$$
$$z_k = \{0, 1\}, \quad k = 0, \ldots, N-1 \tag{5d}$$

into affine constraints. All we need to transform is the following.

$$Dx_{k+1} + \alpha_k \succeq 0, \quad k = 0, \dots, N-1 \tag{5e}$$
$$Dx_{k+1} + \alpha_k \preceq M(1 - z_k), \quad k = 0, \dots, N-1 \tag{5f}$$
$$\lambda_k \preceq M z_k, \quad k = 0, \dots, N-1 \tag{5g}$$

(5e) can be written as

$$\left[\begin{bmatrix} D & & \\ & \ddots & \\ & & D \end{bmatrix} \quad 0 \quad 0 \quad 0\right] \begin{bmatrix} x \\ u \\ \lambda \\ z \end{bmatrix} + \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_{N-1} \end{bmatrix} \succeq 0 \quad \Rightarrow \quad D_{mat}X + D_{lim} \succeq 0 \tag{6}$$

(5f) can be written as

$$\left[\begin{bmatrix} D & & \\ & \ddots & \\ & & D \end{bmatrix} \quad 0 \quad 0 \quad MI\right] \begin{bmatrix} x \\ u \\ \lambda \\ z \end{bmatrix} \preceq M\mathbf{1} - \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_{N-1} \end{bmatrix} \quad \Rightarrow \quad E_{mat}X \preceq E_{lim} \tag{7}$$

(5g) can be written as

$$\begin{bmatrix} 0 & 0 & I & -MI \end{bmatrix} \begin{bmatrix} x \\ u \\ \lambda \\ z \end{bmatrix} \preceq 0 \quad \Rightarrow \quad F_{mat}X \preceq 0 \tag{8}$$

To sum up, using (2)-(4), (6)-(8), we obtain the model that can be optimized by Gurobi.

$$\begin{aligned} \min_{X} \quad & X^T S_{mat} X + x_0^T Q x_0 \\ \text{subject to} \quad & A_{mat}X - b_{mat}x_0 = 0 \\ & D_{mat}X + D_{lim} \succeq 0 \\ & E_{mat}X \preceq E_{lim} \\ & F_{mat}X \preceq 0 \\ & x_{min} \preceq x_k \preceq x_{max}, \quad k = 0, \dots, N \\ & u_{min} \preceq u_k \preceq u_{max}, \quad k = 0, \dots, N-1 \\ & \lambda_k \succeq 0, \quad k = 0, \dots, N-1 \\ & z_k = \{0, 1\}, \qquad k = 0, \dots, N-1 \end{aligned} \tag{9}$$

where $X = \begin{bmatrix} x^T & u^T & \lambda^T & z^T \end{bmatrix}^T$.

1.2 Formulation with substitution

The decision variable in this method is $X = \begin{bmatrix} u^T & \lambda^T & z^T \end{bmatrix}$, where z is the binary vector. x_0 is initial state and is known.

First we write the system dynamics.

$$\begin{aligned} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} &= \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix} x_0 + \begin{bmatrix} B & & & \\ AB & B & & \\ \vdots & \vdots & \ddots & \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix} + \begin{bmatrix} C & & & \\ AC & C & & \\ \vdots & \vdots & \ddots & \\ A^{N-1}C & A^{N-2}C & \cdots & C \end{bmatrix} \begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \vdots \\ \lambda_{N-1} \end{bmatrix} \\ &= \tilde{A}x_0 + \tilde{B}u + \tilde{C}\lambda \end{aligned} \tag{10}$$

Like before we ignore the binary vector, and the objective can be written as

$$\begin{aligned} J' &= x_0^T Q x_0 + \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}^T \begin{bmatrix} Q & & \\ & \ddots & \\ & & Q_f \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} + \begin{bmatrix} u_0 \\ \vdots \\ u_{N-1} \end{bmatrix}^T \begin{bmatrix} R & & \\ & \ddots & \\ & & R \end{bmatrix} \begin{bmatrix} u_0 \\ \vdots \\ u_{N-1} \end{bmatrix} \\ &= x_0^T (Q + \tilde{A}^T \tilde{Q} \tilde{A}) x_0 + \begin{bmatrix} u \\ \lambda \end{bmatrix} \begin{bmatrix} \tilde{B}^T \tilde{Q} \tilde{B} + \tilde{R} & \tilde{B}^T \tilde{Q} \tilde{C} \\ \tilde{C}^T \tilde{Q} \tilde{B} & \tilde{C}^T \tilde{Q} \tilde{C} \end{bmatrix} \begin{bmatrix} u \\ \lambda \end{bmatrix} + 2 \begin{bmatrix} x_0^T \tilde{A}^T \tilde{Q} \tilde{B} & x_0^T \tilde{A}^T \tilde{Q} \tilde{C} \end{bmatrix} \begin{bmatrix} u \\ \lambda \end{bmatrix} \\ &= \tilde{X}^T S \tilde{X} + 2 x_0^T c^T \tilde{X} + x_0^T (Q + \tilde{A}^T \tilde{Q} \tilde{A}) x_0 \end{aligned}$$

where $\tilde{Q} = \begin{bmatrix} Q & & \\ & \ddots & \\ & & Q_f \end{bmatrix}, \tilde{R} = \begin{bmatrix} R & & \\ & \ddots & \\ & & R \end{bmatrix}$.

Adding binary vector z, we have

$$\begin{aligned} J =& X^T \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix} X + 2 x_0^T \begin{bmatrix} c^T & 0 \end{bmatrix} X + x_0^T (Q + \tilde{A}^T \tilde{Q} \tilde{A}) x_0 \\ :=& X^T S_{mat} X + 2 x_0^T c_{mat}^T X + x_0^T (Q + \tilde{A}^T \tilde{Q} \tilde{A}) x_0 \end{aligned} \tag{11}$$

Since we use the dynamics to rewrite the objective, there is no need to add the dynamics costraints in the problem. We don't need to transform (5b)-(5d) into affine constraints, too.

(5a) can be written as

$$\begin{aligned} \begin{bmatrix} \begin{bmatrix} I \\ -I \end{bmatrix} & & \\ & \ddots & \\ & & \begin{bmatrix} I \\ -I \end{bmatrix} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} \preceq \begin{bmatrix} \begin{bmatrix} x_{max} \\ -x_{min} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} x_{max} \\ -x_{min} \end{bmatrix} \end{bmatrix} \quad &\Rightarrow \quad \begin{bmatrix} G\tilde{B} & G\tilde{C} & 0 \end{bmatrix} \begin{bmatrix} u \\ \lambda \\ z \end{bmatrix} \preceq g - G\tilde{A} x_0 \\ &\Rightarrow \quad G_{mat} X \preceq G_{lim1} - G_{lim2} x_0 \end{aligned} \tag{12}$$

(5e) can be written as

$$\begin{aligned} \begin{bmatrix} D & & \\ & \ddots & \\ & & D \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} + \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_{N-1} \end{bmatrix} \succeq 0 \Rightarrow \begin{bmatrix} \tilde{D}\tilde{B} & \tilde{D}\tilde{C} & 0 \end{bmatrix} \begin{bmatrix} u \\ \lambda \\ z \end{bmatrix} + \tilde{D}\tilde{A} x_0 + \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_{N-1} \end{bmatrix} \succeq 0 \\ \Rightarrow D_{mat} X + D_{lim1} + D_{lim2} x_0 \succeq 0 \end{aligned} \tag{13}$$

(5f) can be written as

$$\begin{aligned} \tilde{D}\tilde{A} x_0 + \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_{N-1} \end{bmatrix} + \tilde{D}\tilde{B} u + \tilde{D}\tilde{C} \lambda \preceq M\mathbf{1} - Mz \\ \Rightarrow \quad \begin{bmatrix} \tilde{D}\tilde{B} & \tilde{D}\tilde{C} & MI \end{bmatrix} \begin{bmatrix} u \\ \lambda \\ z \end{bmatrix} \preceq M\mathbf{1} - \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_{N-1} \end{bmatrix} - \tilde{D}\tilde{A} x_0 \\ \Rightarrow \quad E_{mat} X \preceq E_{lim1} - E_{lim2} x_0 \end{aligned} \tag{14}$$

(5g) can be written as

$$\begin{bmatrix}0 & I & -MI\end{bmatrix}\begin{bmatrix}u\\ \lambda\\ z\end{bmatrix} \preceq 0 \quad \Rightarrow \quad F_{mat}X \preceq 0 \tag{15}$$

To sum up, using (2), (10)-(15), we obtain the model that can be optimized by Gurobi.

$$\begin{aligned}\min_{X} \quad & X^T S_{mat} X + x_0^T Q x_0 \\ \text{subject to} \quad & D_{mat}X + D_{lim1} + D_{lim2}x_0 \succeq 0 \\ & E_{mat}X \preceq E_{lim1} - E_{lim2} \\ & F_{mat}X \preceq 0 \\ & G_{mat}X \preceq G_{lim1} - G_{lim2}x_0 \\ & u_{min} \preceq u_k \preceq u_{max}, \quad k = 0, \dots, N-1 \\ & \lambda_k \succeq 0, \quad k = 0, \dots, N-1 \\ & z_k = \{0, 1\}, \qquad k = 0, \dots, N-1\end{aligned} \tag{16}$$

where $X = \begin{bmatrix}u^T & \lambda^T & z^T\end{bmatrix}^T$.

1.3 Tracking issues

Suppose we want the robot to reach the desired point x_d, we need to change the objective to be

$$\min_{x,u} \quad (x_n - x_d)^T Q_f (x_n - x_d) + \sum_{k=1}(x_k - x_d)^T Q (x_k - x_d) + u_k^T R u_k$$

Now we consider x_d to be a fixed desired point, then we rewrite the objectve

$$\begin{aligned} J = & \begin{bmatrix}x_1 - x_d\\ \vdots \\ x_N - x_d\end{bmatrix}^T \begin{bmatrix}Q & & \\ & \ddots & \\ & & Q_f\end{bmatrix} \begin{bmatrix}x_1 - x_d\\ \vdots \\ x_N - x_d\end{bmatrix} + \begin{bmatrix}u_0\\ \vdots \\ u_{N-1}\end{bmatrix}^T \begin{bmatrix}R & & \\ & \ddots & \\ & & R\end{bmatrix} \begin{bmatrix}u_0\\ \vdots \\ u_{N-1}\end{bmatrix} \\ & + (x_0 - x_d)^T Q (x_0 - x_d) \\ = & \, x^T \tilde{Q} x - 2\tilde{x}_d^T \tilde{Q} x + u^T \tilde{R} u + \tilde{x}_d^T \tilde{Q} \tilde{x}_d + (x_0 - x_d)^T Q (x_0 - x_d)\end{aligned} \tag{17}$$

where $\tilde{x}_d = kron(\mathbf{1}, x_d)$. Compare (17) with (3) and (11), we can find that we only need to deal with an additional term $-2\tilde{x}_d^T \tilde{Q} x$. The rest part remains the same except the constant term changes slightly.

So for the method without substitution, we have new objective

$$\begin{aligned} \tilde{J} &= X^T S_{mat} X - 2\tilde{x}_d^T \tilde{Q} x + \tilde{x}_d^T \tilde{Q} \tilde{x}_d + (x_d - x_0)^T Q (x_d - x_0) \\ &= X^T S_{mat} X - 2\begin{bmatrix}\tilde{x}_d^T \tilde{Q} & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}x\\ u\\ \lambda\\ z\end{bmatrix} + \tilde{x}_d^T \tilde{Q} \tilde{x}_d + (x_d - x_0)^T Q (x_d - x_0) \\ &= X^T S_{mat} X + 2\tilde{x}_d^T d_{mat}^T X + \tilde{x}_d^T \tilde{Q} \tilde{x}_d + (x_d - x_0)^T Q (x_d - x_0)\end{aligned} \tag{18}$$

For method with substitution, we have new objective

$$\tilde{J} = X^T S_{mat} X + 2x_0^T c_{mat}^T X + x_0^T \tilde{A}^T \tilde{Q} \tilde{A} x_0 - 2\tilde{x}_d^T \tilde{Q} x + \tilde{x}_d^T \tilde{Q} \tilde{x}_d + (x_d - x_0)^T Q (x_d - x_0) \tag{19a}$$

$$
\begin{aligned}
-2\tilde{x}_d^T\tilde{Q}x &= -2\tilde{x}_d^T\tilde{Q}\tilde{A}x_0 - 2\begin{bmatrix}\tilde{x_d}^T\tilde{Q}\tilde{B} & \tilde{x}_d\tilde{Q}\tilde{C} & 0\end{bmatrix}\begin{bmatrix}u\\ \lambda\\ z\end{bmatrix} \\
&= -2\tilde{x}_d^T\tilde{Q}\tilde{A}x_0 + 2\tilde{x}_d^T d_{mat}^T X
\end{aligned} \tag{19b}
$$

From (19a) and (19b), we have

$$
\tilde{J} = X^T S_{mat} X + 2(x_0^T c_{mat}^T + \tilde{x}_d^T d_{mat}^T)X - 2\tilde{x}_d^T\tilde{Q}\tilde{A}x_0 + \tilde{x}_d^T\tilde{Q}\tilde{x}_d + (x_d - x_0)^T Q(x_d - x_0) \tag{20}
$$

2 Test Items

To better test the big M method for complementarity constraints, we will test the following performance for two methods mentioned above.

(1) Modeling time t_m: the time span that formulate the problem before solving it.

(2) Solving time t_s: the time span that Gurobi solves the problem. This indicator can tells which method is more suitable for fast computing.

(3) $N - t_s$ curve: the prediction step vs solving time curve. This curve reveals the relationship as N increases. It is a good indicator for whether the big M method is good enough for largeer size problem or not. Besides, it is also useful for comparison with different methods to solve QPCC like SDP.

(4) error curve: the optimal solution and optimal value error for tow methods mentioned above.

*This document will be constantly updated.