EECS 127/227AT Optimization Models in Engineering UC Berkeley Spring 2023

Discussion 13

1. LQR and Least Squares

In this question, we consider the time-dependent n-state m-input LQR problem

$$\min_{\vec{x}_t\in\mathbb{R}^n,\vec{u}_t\in\mathbb{R}^m} \sum_{t=0}^{T-1}\left(\vec{x}_t^\top Q_t\vec{x}_t + \vec{u}_t^\top R_t\vec{u}_t\right) + \vec{x}_T^\top Q_T\vec{x}_T \tag{1}$$

$$\text{s.t.}\quad \vec{x}_{t+1} = A\vec{x}_t + B\vec{u}_t,\ t = 1,\dots,T \tag{2}$$

$$\vec{x}_0 = \vec{x}_{\text{init}}, \tag{3}$$

where $Q_t = Q_t^\top \succeq 0$ for all $t = 0,\dots,T$ and $R_t = R_t^\top \succeq 0$ for all $t = 0,\dots,T-1$. Note that this is a minor extension of the standard LQR formulation explored in class — we allow the cost associated with each state $\vec{x}_t$ and input $\vec{u}_t$ to vary by time step. For clarity, note also that $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $Q_t \in \mathbb{R}^{n\times n}$ for all $t = 0,\dots,T$, and $R_t \in \mathbb{R}^{m\times m}$ for all $t = 0,\dots,T-1$. In this problem, we reformulate this calculation as a least squares problem, examine its properties, and compare this solution strategy with others shown in class.

(a) ***Concatenating variables of interest.*** We first make our formulation more concise by concatenating our states and inputs into single vectors and computing the associated matrices.

i. Define full state vector $\vec{x}$ and input vector $\vec{u}$ as follows:

$$\vec{x} = \begin{bmatrix}\vec{x}_0\\ \vec{x}_1\\ \vdots\\ \vec{x}_T\end{bmatrix} \in \mathbb{R}^{n(T+1)}, \qquad \vec{u} = \begin{bmatrix}\vec{u}_0\\ \vec{u}_1\\ \vdots\\ \vec{u}_{T-1}\end{bmatrix} \in \mathbb{R}^{mT}. \tag{4}$$

Show that we can rewrite our LQR objective function as

$$\vec{x}^\top Q\vec{x} + \vec{u}^\top R\vec{u} \tag{5}$$

for some matrices $Q \in \mathbb{R}^{n(T+1)\times n(T+1)}$ and $R \in \mathbb{R}^{mT\times mT}$, which you determine.

Solution: The desired equivalence holds when

$$Q = \begin{bmatrix} Q_0 & 0 & \cdots & \cdots & 0\\ 0 & Q_1 & 0 & \cdots & 0\\ \vdots & & \ddots & & \vdots\\ \vdots & & & \ddots & 0\\ 0 & \cdots & \cdots & 0 & Q_T\end{bmatrix}, \qquad R = \begin{bmatrix} R_0 & 0 & \cdots & \cdots & 0\\ 0 & R_1 & 0 & \cdots & 0\\ \vdots & & \ddots & & \vdots\\ \vdots & & & \ddots & 0\\ 0 & \cdots & \cdots & 0 & R_{T-1}\end{bmatrix}. \tag{6}$$

ii. Show that we can reformulate our constraints (i.e., dynamics) as

$$\vec{x} = G\vec{u} + H\vec{x}_{\text{init}} \tag{7}$$

for some matrices $G \in \mathbb{R}^{n(T+1)\times mT}$ and $H \in \mathbb{R}^{n(T+1)\times n}$, which you determine.

Solution: We can rewrite our dynamics equations iteratively as

$$\vec{x}_0 = \vec{x}_{\text{init}} \tag{8}$$

$$\vec{x}_1 = A\vec{x}_0 + B\vec{u}_0 = A\vec{x}_{\text{init}} + B\vec{u}_0 \tag{9}$$

$$\vec{x}_2 = A\vec{x}_1 + B\vec{u}_1 = A(A\vec{x}_{\text{init}} + B\vec{u}_0) + B\vec{u}_1 = A^2\vec{x}_{\text{init}} + AB\vec{u}_0 + B\vec{u}_1 \tag{10}$$

$$\vdots \tag{11}$$

$$\vec{x}_t = A^t\vec{x}_{\text{init}} + \sum_{s=0}^{t-1} A^{t-s-1}B\vec{u}_s \tag{12}$$

$$\vdots \tag{13}$$

Compiling these equations into matrix form, we have

$$\begin{bmatrix} \vec{x}_0 \\ \vec{x}_1 \\ \vec{x}_2 \\ \vdots \\ \vec{x}_{T-1} \\ \vec{x}_T \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & \cdots & \cdots & \cdots & 0 \\ B & 0 & \cdots & \cdots & 0 \\ AB & B & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ A^{T-2}B & A^{T-3}B & \cdots & B & 0 \\ A^{T-1}B & A^{T-2}B & \cdots & AB & B \end{bmatrix}}_{G} \begin{bmatrix} \vec{u}_0 \\ \vec{u}_1 \\ \vec{u}_2 \\ \vdots \\ \vec{u}_{T-1} \end{bmatrix} + \underbrace{\begin{bmatrix} I \\ A \\ A^2 \\ \vdots \\ A^{T-1} \\ A^T \end{bmatrix}}_{H} \vec{x}_{\text{init}} \tag{14}$$

as desired.

(b) ***Formulating the least squares problem.*** We have now reduced our LQR problem to

$$\min_{\vec{x}\in\mathbb{R}^{n(T+1)},\vec{u}\in\mathbb{R}^{mT}} \vec{x}^\top Q\vec{x} + \vec{u}^\top R\vec{u} \tag{15}$$

$$\text{s.t. } \vec{x} = G\vec{u} + H\vec{x}_{\text{init}}. \tag{16}$$

Rewrite this optimization as an unconstrained least squares problem over $\vec{u}$.

Solution: Our constraint now explicitly defines the value of $\vec{x}$, so we can plug it into the objective function to get equivalent optimization problem

$$\min_{\vec{u}\in\mathbb{R}^{mT}} (G\vec{u} + H\vec{x}_{\text{init}})^\top Q(G\vec{u} + H\vec{x}_{\text{init}}) + \vec{u}^\top R\vec{u} \tag{17}$$

$$= \min_{\vec{u}\in\mathbb{R}^{mT}} \left\|Q^{\frac{1}{2}}G\vec{u} + Q^{\frac{1}{2}}H\vec{x}_{\text{init}}\right\|_2^2 + \left\|R^{\frac{1}{2}}\vec{u}\right\|_2^2 \tag{18}$$

$$= \min_{\vec{u}\in\mathbb{R}^{mT}} \left\|\begin{bmatrix} Q^{\frac{1}{2}}G \\ R^{\frac{1}{2}} \end{bmatrix}\vec{u} + \begin{bmatrix} Q^{\frac{1}{2}}H\vec{x}_{\text{init}} \\ 0 \end{bmatrix}\right\|_2^2 \tag{19}$$

where matrix square-roots $Q^{\frac{1}{2}}$ and $R^{\frac{1}{2}}$ are

$$Q^{\frac{1}{2}} = \begin{bmatrix} Q_0^{\frac{1}{2}} & 0 & \cdots & \cdots & 0 \\ 0 & Q_1^{\frac{1}{2}} & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & Q_T^{\frac{1}{2}} \end{bmatrix}, \quad R^{\frac{1}{2}} = \begin{bmatrix} R_0^{\frac{1}{2}} & 0 & \cdots & \cdots & 0 \\ 0 & R_1^{\frac{1}{2}} & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & R_{T-1}^{\frac{1}{2}} \end{bmatrix}. \tag{20}$$

(Recall that because each of Q's (R's) block diagonal entries Q_t (R_t) are PSD, we can compute a real-valued matrix square root in this manner.) This is now a standard least squares problem over $\vec{u}$ as desired.

(c) ***Analysis.*** We now examine the practicality of using least squares to solve the LQR problem. *NOTE*: This section is meant to provide intuition, not rigorous complexity analysis, and is presented primarily to illustrate the practical utility of different LQR methods. Do not feel obligated to understand the arguments below in detail. Recall from lecture that the LQR problem can also be solved via the following recursive procedure:

(1) Set $P_T = Q_T$, then solve iteratively backward in time for "helper" matrices $P_{T-1}, \ldots, P_0$ via Riccati equation

$$P_t = A^\top (I + P_{t+1} B R_t^{-1} B^\top)^{-1} P_{t+1} A + Q_t \tag{21}$$

(2) Solve iteratively forward in time for optimal $\vec{x}_1, \ldots, \vec{x}_T$ and $\vec{u}, \ldots, \vec{u}_{T-1}$ via

$$\vec{u}_t = -R_t^{-1} B^\top (I + P_{t+1} B R_t^{-1} B^\top)^{-1} P_{t+1} A \vec{x}_t \tag{22}$$

$$\vec{x}_{t+1} = A\vec{x}_t + B\vec{u}_t \tag{23}$$

We will compare this strategy with the least squares formulation developed above.

i. Suppose $n = 2$, $m = 2$, and $T = 10,000$, i.e., we want to solve for the optimal control of a 2-state, 2-input system over a horizon of 10,000 time steps. Which solution method would you use? *HINT: Over long time horizons, computational efficiency is a major concern.*

Solution: If computational efficiency is of primary concern, as with long time horizons, the iterative method explored in class is the better choice. Our least squares problem is of matrix size $n(T + 1) + mT \times mT$ — that's $40,002 \times 20,000$, or over 800 million entries. Computing the solution to such a problem using standard least squares techniques (QR factorization, naive matrix inversion, etc.) would require serious computing power. In contrast, while the iterative method requires computing matrices and vector values at each of the $10,000$ time steps, it requires multiplying and inverted matrices no larger than 2×2, so each computation is effectively constant time, and the overall runtime is several orders of magnitude smaller.

ii. Suppose $n = 2$, $m = 2$, and $T = 100$, and we want to impose constraints on the control values $\vec{u}$ (e.g., each element of the $\vec{u}$ vector must remain between ± 10 units).[1] Which of these formulations might you choose to incorporate such constraints?

Solution: While our iterative formulation is useful for unconstrained problems, it does not readily accommodate constraints of any kind, and is generally most useful when operating when such constraints aren't necessary (e.g., when you want to control a device in an operating range that won't require too much control input from its actuators). Our least squares formulation, on the other hand, is just a QP, so for small enough time horizons (resulting in small enough matrices to be efficiently computable), we can easily tack on arbitrary convex constraints!

2. Can we Use Slack Variables?

So far, we've presented slack variables as a method of converting optimization problems to a desired form, and it may seem like we can always use them. In this question, we take a more nuanced look at when slack variables are helpful and when they are not. For each of the following functions, consider the unconstrained optimization

[1]Constraints like this are common in control problems; our motors/actuators usually can't provide infinite power!

problem

$$p_j^* = \min_{\vec{x} \in \mathbb{R}^n} f_j(\vec{x}) \tag{24}$$

If possible, reformulate each problem into an LP/convex QP/SOCP using slack variables. If not possible, explain why.

(a) $f_1(\vec{w}) = \sum_{i=1}^{n} \left(\max\{0, 1 - y_i \vec{x}_i^\top \vec{w}\}\right)^2 + C \|\vec{w}\|_2^2$ for $C > 0$ and for some given vectors $\vec{x}_i \in \mathbb{R}^d$ for $i = 1, \ldots, n$ and $\vec{y} \in \mathbb{R}^n$ and variable $\vec{w} \in \mathbb{R}^d$.

Solution: We introduce slack variables ζ_i for $i = 1, \ldots, n$ corresponding to the max term. This leads to rewriting the miimization of f_1 as

$$\begin{aligned} \min \sum_{i=1}^{n} \zeta_i^2 + C \|\vec{w}\|_2^2 & \\ \text{s.t.} \quad & \zeta_i \geq 0 \\ & \zeta_i \geq 1 - y_i \vec{x}_i^\top \vec{w} \quad i = 1, \ldots, n \end{aligned}$$

This minimization probem is clearly a convex QP.

(b) $f_2(\vec{x}) = \|A\vec{x} - \vec{y}\|_2 - \|\vec{x}\|_1$.

Solution: While this problem appears almost identical to the one in part (a), it **cannot be written as an SOCP**; in fact, it is not convex. To see this, we can attempt to follow the same procedure as above. First, we introduce our new variables, to generate equivalent problem

$$p_2^* = \min_{\vec{x}, \vec{t} \in \mathbb{R}^n, s \in \mathbb{R}} \quad s - \sum_{i=1}^{n} t_i \tag{25}$$

$$\text{s.t.} \quad \|A\vec{x} - \vec{y}\|_2 = s \tag{26}$$

$$|x_i| = t_i, \quad i = 1, \ldots, n. \tag{27}$$

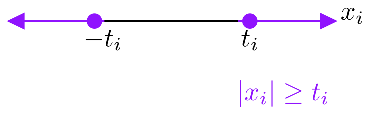
We then relax our equality constraints to inequalities; again, we can perform this step successfully, but we must be cautious, as the sign on each value t_i in the objective function has changed. If we require that $|x_i| \leq t_i$ as before, then our optimization problem is unbounded below, since we can increase t_i arbitrarily high to generate an arbitrarily low value. Instead, we must require that $|x_i| \geq t_i$, giving us optimization problem

$$p_2^* = \min_{\vec{x}, \vec{t} \in \mathbb{R}^n, s \in \mathbb{R}} \quad s - \sum_{i=1}^{n} t_i \tag{28}$$

$$\text{s.t.} \quad \|A\vec{x} - \vec{y}\|_2 \leq s \tag{29}$$

$$|x_i| \geq t_i, \quad i = 1, \ldots, n. \tag{30}$$

While this is a valid (though nonconvex) optimization problem, each of the final constraints $|x_i| \geq t_i$ is not convex:



We therefore cannot linearize them to get an SOCP.