Discussion 13

1. LQR and Least Squares

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

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

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

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

where $Q_t = Q_t^\top \succeq 0$ for all $t = 0, \ldots, T$ and $R_t = R_t^\top \succeq 0$ for all $t = 0, \ldots, T-1$. 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 $t = 0, \ldots, T$, and $R_t \in \mathbb{R}^{m\times m}$ for all $t = 0, \ldots, T-1$. 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.

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

$$\vec{x} = G\vec{u} + H\vec{x}_{\text{init}} \tag{6}$$

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.

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

$$\min_{\vec{x}\in\mathbb{R}^{n(T+1)},\vec{u}\in\mathbb{R}^{mT}} \vec{x}^\top Q\vec{x} + \vec{u}^\top R\vec{u} \tag{7}$$

$$\text{s.t. } \vec{x} = G\vec{u} + H\vec{x}_{\text{init}}. \tag{8}$$

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

(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

$$P_t = A^\top(I + P_{t+1}BR_t^{-1}B^\top)^{-1}P_{t+1}A + Q_t \tag{9}$$

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

$$\vec{u}_t = -R_t^{-1}B^\top(I + P_{t+1}BR_t^{-1}B^\top)^{-1}P_{t+1}A\vec{x}_t \tag{10}$$

$$\vec{x}_{t+1} = A\vec{x}_t + B\vec{u}_t \tag{11}$$

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.*

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?

2. Can we Use Slack Variables?

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

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 problem

$$p_j^* = \min_{\vec{x} \in \mathbb{R}^n} \; f_j(\vec{x}) \tag{12}$$

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 $\vec{w} \in \mathbb{R}^d$.

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

