EE363 Homework 5

Spring 2026

22.1000. Controlling a mixing tank. Consider a mixing tank with cold and hot water supplies at constant supply temperatures T_C and T_H, respectively. Let q_C and q_H denote the input flow rate for each supply and treat them as control inputs. Let h and T be the height and temperature of the water in the tank; these are our state variables. The differential equations governing this system are

$$\dot{h} = \frac{1}{\alpha}\left(q_C + q_H - \beta\sqrt{h}\right)$$
$$\dot{T} = \frac{1}{\alpha h}\left(q_C(T_C - T) + q_H(T_H - T)\right)$$

where α is the area of the tank and the term $\beta\sqrt{h}$, where β is a constant, dictates the rate at which water is drained. Using the standard state and input notation, let $x_1 = h, x_2 = T, u_1 = q_C, u_2 = q_H$.

a) Suppose we want the steady-state values to be $(h^{\mathrm{eq}}, T^{\mathrm{eq}})$, where $h^{\mathrm{eq}} > 0$ and $T^{\mathrm{eq}} \in [T_C, T_H]$. What are the corresponding steady-state input values, $u_1^{\mathrm{eq}}, u_2^{\mathrm{eq}}$?

b) We can linearize this system around the equilibrium $(h^{eq}, T^{eq}, u_1^{eq}, u_2^{eq})$ using standard techniques:

$$A = \left.\frac{\partial f}{\partial x}\right|_{eq}, \quad B = \left.\frac{\partial f}{\partial u}\right|_{eq}$$

The linearized matrices are

$$A = \begin{bmatrix} -\frac{\beta}{2\alpha\sqrt{h^{eq}}} & 0 \\ 0 & -\frac{\beta}{\alpha\sqrt{h^{eq}}} \end{bmatrix}, \quad B = \frac{1}{\alpha}\begin{bmatrix} 1 & 1 \\ \frac{T_C - T^{eq}}{h^{eq}} & \frac{T_H - T^{eq}}{h^{eq}} \end{bmatrix}$$

Note that A is diagonal, which reflects that height and temperature dynamics decouple at steady-state.

Suppose $T_C = 10°, T_H = 90°, \alpha = 3\mathrm{m}^2, \beta = 0.155\mathrm{m}^{5/2}/\mathrm{s}$, and $h^\star = 1\mathrm{m}, T^\star = 25°$. Evaluate A and B in part (b) for these values. Report the eigenvalues of A (you should do this numerically in your favorite programming language). Is the system (A, B) controllable at these values?

c) Design a controller of the form $\tilde{u} = L\tilde{x}$ for the values in part (c) using the pole placement method. Compute feedback gains L using your favorite programming language for the following closed-loop eigenvalue locations:

$$\{-0.1, -0.2\}, \quad \{-0.3, -0.6\}, \quad \{-1, -2\}.$$

Hint: in Python you can use `scipy.signal.place_poles`.

d) Optional: simulate the nonlinear model for each controller, You may observe some non-physical behavior (such as $h(t) \leq 0$ or negative flow rates. Why do you think this occurs?

Solution.

a) At equilibrium, $\dot{h} = 0$ and $\dot{T} = 0$.

From $\dot{h} = 0$:

$$\frac{1}{\alpha}\left(u_1^{eq} + u_2^{eq} - \beta\sqrt{h^{eq}}\right) = 0$$
$$\Rightarrow u_1^{eq} + u_2^{eq} = \beta\sqrt{h^{eq}}$$

From $\dot{T} = 0$:

$$\frac{1}{\alpha h^{eq}}\left(u_1^{eq}(T_C - T^{eq}) + u_2^{eq}(T_H - T^{eq})\right) = 0$$
$$\Rightarrow u_1^{eq}(T_C - T^{eq}) + u_2^{eq}(T_H - T^{eq}) = 0$$

From the second equation: $u_1^{eq} = u_2^{eq} \cdot \dfrac{T^{eq} - T_H}{T_C - T^{eq}}$ Substituting into the first:

$$u_2^{eq} = \beta\sqrt{h^{eq}} \cdot \frac{T^{eq} - T_C}{T_H - T_C}, \qquad u_1^{eq} = \beta\sqrt{h^{eq}} \cdot \frac{T_H - T^{eq}}{T_H - T_C}$$

Interpretation: both inputs are non-negative since $T^{eq} \in [T_C, T_H]$, and they blend hot/cold water in proportion to how far T^{eq} is from each supply temperature.

b) Here is a short Python code to compute the eigenvalues and check controllability of (A, B): The system (A, B) is controllable if and only if $\mathcal{C} = \begin{bmatrix} B & AB \end{bmatrix}$ has full rank (2). Output of our code:

```
Eigenvalues of A: [-0.02583333 -0.05166667]
Rank of controllability matrix: 2 (system size: 2)
Controllable: True
```

c) The closed-loop dynamics become:

$$\dot{\tilde{x}} = (A + BL)\tilde{x}$$

Here's an example using `scipy.signal.place_poles` to compute L for each set of desired eigenvalues.

```
from scipy.signal import place_poles

pole_sets = [[-0.1, -0.2], [-0.3, -0.6], [-1, -2]]

for poles in pole_sets:
    result = place_poles(A, B, poles)
    L = result.gain_matrix
    cl_eigenvalues = np.linalg.eigvals(A + B @ L)
    print(f"Desired poles: {poles}")
    print(f"L =\n{L}")
    print(f"Closed-loop eigenvalues: {cl_eigenvalues}\n")
```

The three pole choices represent increasingly aggressive controllers — faster convergence to steady-state but requiring larger inputs. The gains L will scale accordingly.

22.1200. Deadbeat control. Consider the discrete-time system

$$x_{k+1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0//1 \end{bmatrix} u_k.$$

Define $A_{\text{cl}} = A + BL$.

A state feedback law that places all eigenvalues of of a discrete-time system at zero is called a deadbeat controller. For an n^{th} order system, this ensures $A_{\text{cl}}^n = 0$ thus, closed-loop trajectories converge to zero in finite time.

Finite-time convergence can't be achieved in continuous time with linear state feedback because the matrix exponential $\exp A_{\text{cl}} t$ is never exactly zero at any finite time. Instead, trajectories approach the origin asymptotically. (An important caveat is that deadbeat control requires large feedback gains, increasing sensitivity to modeling errors and noise.)

a) Design a state feedback control $u_k = L x_k$ that places both closed-loop eigenvalues at zero.

b) Show that $A_{\text{cl}}^2 = 0$. What does this imply for the closed-loop trajectories?

Solution.

a) The characteristic polynomial of A_{cl} is:

$$\det(\lambda I - A_{\text{cl}}) = \lambda^2 - (2 + \ell_2)\lambda + (1 + \ell_2 - \ell_1)$$

For zero eigenvalues, the desired characteristic polynomial is $\lambda^2 = 0$.

$$2 + \ell_2 = 0 \implies \ell_2 = -2$$

$$1 + \ell_2 - \ell_1 = 0 \implies \ell_1 = 1 + \ell_2 = -1$$

Thus,

$$L = \begin{bmatrix} -1 & -2 \end{bmatrix}.$$

b) With this gain, the closed-loop matrix is:

$$A_{\text{cl}} = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$$

Computing A_{cl}^2:

$$A_{\text{cl}}^2 = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

What does this imply about the trajectories? The closed-loop dynamics are $x_{k+1} = A_{cl} x_k$, so $x_k = A_{cl}^k x_0$. Since $A_{cl}^2 = 0$, for any initial condition x_0:

$$x_k = 0 \quad \text{for all } k \geq 2$$

The state reaches the origin in at most 2 steps, regardless of x_0.

23.1000. Linear quadratic state tracking. We consider the system $x_{t+1} = Ax_t + Bu_t$. In the conventional LQR problem the goal is to make both the state and the input small. In this problem we study a generalization in which we want the state to follow a desired (possibly nonzero) trajectory as closely as possible. To do this we penalize the *deviations* of the state from the desired trajectory, *i.e.*, $x_t - x_t^d$, using the following cost function:

$$J = \sum_{\tau=0}^{N} \left(x_\tau - x_\tau^d\right)^\top Q \left(x_\tau - x_\tau^d\right) + \sum_{\tau=0}^{N-1} u_\tau^\top R u_\tau,$$

where we assume $Q = Q^\top \succeq 0$ and $R = R^\top \succ 0$. (The desired trajectory x_τ^d is given.) Compared with the standard LQR obejctive, we have an extra linear term (in x) and a constant term.

In this problem you will use dynamic programming to show that the cost-to-go function $V_t(z)$ for this problem has the form

$$z^\top P_t z + 2q_t^\top z + r_t,$$

with $P_t = P_t^\top \succeq 0$. (*i.e.*, it has quadratic, linear, and constant terms.)

a) Show that $V_n(z)$ has the given form.

b) Assuming $V_{t+1}(z)$ has the given form, show that the optimal input at time t can be written as

$$u_t^\star = K_t x_t + g_t,$$

where

$$K_t = -\left(R + B^\top P_{t+1} B\right)^{-1} B^\top P_{t+1} A, \quad g_t = -\left(R + B^\top P_{t_1} B\right)^{-1} B^\top q_{t+1}.$$

In other words, $u_t^\star$ is an affine (linear plus constant) function of the state x_t.

c) Use backward induction to show that $V_0(z), \dots, V_N(z)$ all have the given form. Verify that

$$\begin{aligned} P_t &= Q + A^\top P_{t+1} A - A^\top P_{t+1} B \left(R + B^\top P_{t+1} B\right)^{-1} B^\top P_{t+1} A, \\ q_t &= (A + BK_t)^\top q_{t+1} - Q x_t^d, \\ r_t &= r_{t+1} + x_t^d Q x_t^d + q_{t+1}^\top B g_t, \end{aligned}$$

for $t = 0, \dots, N-1$.

Solution.

a) We can write

$$\begin{aligned} V_N(z) &= (z - x_N^d)^\top Q (z - x_N^d) \\ &= z^\top Q z - 2x_N^d Q z + x_N^d Q x_N^d \\ &= z^\top P_N z + 2q_N^\top z + r_N, \end{aligned}$$

where $P_N = Q$, $q_N = -Q^\top x_N^d$ and $r_N = x_N^d Q x_N^d$.

b) Assuming $V_{t+1}(z)$ has the given form, the Bellman equation can be written as

$$\begin{aligned}V_t(z) &= \min_w \left\{ \left(z - x_t^d\right)^\top Q \left(z - x_t^d\right) + w^\top R w + V_{t+1}(Az + Bw) \right\} \\ &= \left(z - x_t^d\right)^\top Q \left(z - x_t^d\right) \\ &\quad + \min_w \left\{ w^\top R w + (Az + Bw)^\top P_{t+1} (Az + Bw) + 2q_{t+1}^\top (Az + Bw) + r_{t+1} \right\}.\end{aligned}$$

To find the w that minimizes the above expression, we set its derivative with respect to w equal to zero. This gives

$$2Rw + 2B^\top P_{t+1}(Az + Bw) + 2B^\top q_{t+1} = 0,$$

and so

$$u_t^\star = w^\star = -\left(R + B^\top P_{t+1} B\right)^{-1} \left(B^\top q_{t+1} + B^\top P A z\right) = K_t z + g_t,$$

where

$$K_t = -\left(R + B^\top P_{t+1} B\right)^{-1} B^\top P_{t+1} A$$

, and

$$g_t = -\left(R + B_{t+1}^\top B\right)^{-1} B^\top q_{t+1}.$$

c) We can write $V_t(z)$ as

$$\begin{aligned}V_t(z) &= z^\top \left(Q + A^\top P_{t+1} A\right) z + 2\left(A^\top q_{t+1} - x_t^d\right)^\top z + x_t^d Q x_t^d + r_{t+1} \\ &\quad + \min_w \left\{ w^\top \left(R + B^\top P_{t+1} B\right) w + 2\left(B^\top P_{t+1} A z \ + B^\top q_{t+1}\right)^\top w \right\}.\end{aligned}$$

Substituting $w^\star$ into the above expression gives

$$\begin{aligned}V_t(z) &= z^\top \left(Q + A^\top P_{t+1} A\right) z + 2\left(A^\top q_{t+1} - x_t^d\right)^\top z + x_t^d Q x_t^d + r_{t+1} \\ &\quad + \left(B^\top q_{t+1} + B^\top P_{t+1} A z\right)^\top \left(R + B^\top P_{t+1}\right)^{-1} \left(B^\top q_{t+1} + B^\top P_{t+1} A z\right) \\ &\quad - 2\left(B^\top q_{t+1} + B^\top P_{t+1} A z\right)^\top \left(R + B^\top P_{t+1} B\right)^{-1} \left(B^\top q_{t+1} + B^\top P_{t+1} A z\right) \\ &= z^\top \left(Q + A^\top P_{t+1} A\right) z + 2\left(A^\top q_{t+1} - x_t^d\right)^\top z + x_t^d Q x_t^d + r_{t+1} \\ &\quad - \left(B^\top q_{t+1} + B^\top P_{t_1} A z\right)^\top \left(R + B^\top P_{t+1} B\right)^{-1} \left(B^\top q_{t+1} + B^\top P_{t+1} A z\right).\end{aligned}$$

After rearranging and collecting terms we get

$$\begin{aligned}V_t(z) &= z^\top \left(Q + A^\top P_{t+1} A - A^\top P_{t+1} B \left(R + B^\top P_{t+1} B\right)^{-1} B^\top P_{t+1} A\right) z \\ &\quad + 2\left(A^\top q_{t+1} - A^\top P_{t+1} B \left(R + B^\top P_{t+1} B\right)^{-1} B^\top q_{t+1} - x_t^d\right)^\top z \\ &\quad + r_{t+1} + x_t^d Q x_t^d - q_{t+1}^\top B \left(R + B^\top P_{t+1} B\right)^{-1} B^\top q_{t+1} \\ &= z^\top P_t z + 2q_t^\top z + r_t,\end{aligned}$$

where

$$
\begin{aligned}
P_t &= Q + A^\top P_{t+1} A - A^\top P_{t+1} B \left(R + B^\top P_{t+1} B\right)^{-1} B^\top P_{t+1} A, \\
q_t &= A^\top q_{t+1} - A^\top P_{t+1} B \left(R + B^\top P_{t+1} B\right)^{-1} B^\top q_{t+1} - x_t^d \\
&= (A - BK)^\top q_{t+1} - x_t^d, \\
r_t &= r_{t+1} + x_t^d Q x_t^d - q_{t+1}^\top B \left(R + B^\top P_{t+1} B\right)^{-1} B^\top q_{t+1} \\
&= r_{t+1} + x_t^d Q x_t^d + q_{t+1}^\top B g_t.
\end{aligned}
$$

Thus we have shown that if $V_{t+1}(z)$ has the given form, then $V_t(z)$ also has the given form. Since $V_N(z)$ has this form (from part (a)), by induction, $V_0(z), \ldots, V_N(z)$ all have the given form.

23.1100. When does a finite-horizon LQR problem have a time-invariant optimal state feedback gain?. Consider a discrete-time LQR problem with horizon $t = N$, with optimal input $u(t) = K_t x(t)$. Is there a choice of Q_f (that is symmetric and positive semidefinite) for which K_t is constant, *i.e.*, $K_0 = \cdots = K_{N-1}$?

Solution. K_t is defined by

$$
K_t = -\left(R + B^\top P_{t+1} B\right)^{-1} B^\top P_{t+1} A.
$$

Therefore, if $P_1 = \cdots = P_n$, we have $K_0 = \cdots = K_{N-1}$. Since $Q_f = P_N$, and for t far from N we have that P_t converges to the steady-state solution, we see that to have constant P_t, we must have Q_f equal to the steady-state solution P_{ss}. With this choice, we have $P_t = P_{\text{ss}}$ for all t, and therefore $K_t = K_{\text{ss}}$ for all t.