EE365 Homework 6

1. *LQR with random dynamics matrix.* We consider the dynamical system

$$x_{t+1} = A_t x_t + B u_t + w_t, \quad t = 0, 1, \dots,$$

where $x_t \in \mathbb{R}^n$, $u_t \in \mathbb{R}^m$. We assume that w_t are IID, with $w_t \sim \mathcal{N}(0, W)$. Unlike the standard LQR problem, the dynamics matrices A_t are random, with the form

$$A_t = A^{\text{nom}} + \Delta_t, \quad t = 0, 1, \dots,$$

where $A^{\text{nom}} \in \mathbb{R}^{n \times n}$ is known, and Δ_t are IID, independent of all w_t, with $(\Delta_t)_{ij}$ uniform on $[-\epsilon, \epsilon]$. In other words, the entries of A_t are independent, with $(A_t)_{ij}$ uniformly distributed on $[A^{\text{nom}}_{ij} - \epsilon, A^{\text{nom}}_{ij} + \epsilon]$.

The (time-invariant) stage cost is $(1/2)(x^T Q x + u^T R u)$, with $Q \geq 0$ and $R > 0$. Let $K^\star \in \mathbb{R}^{m \times n}$ denote the feedback gain matrix for the policy $\mu^\star(x) = K^\star x$ that minimizes average stage cost. (You can assume that it exists and is unique.)

(a) Explain how to find $K^\star$ (in the limit) using value iteration. Be as explicit as you can be about the iteration that yields $K^\star$ (in the limit). (Yes, we are asking you to work out the 19th century style formulas.)

(b) Consider the specific problem instance with $n = 3$, $m = 1$, $Q = I$, $R = 1$, $W = I$, $\epsilon = 0.3$, and

$$A^{\text{nom}} = \begin{bmatrix} 0.9 & -0.2 & 0.4 \\ -0.1 & -0.9 & 0.3 \\ -0.8 & -0.1 & 1.2 \end{bmatrix}, \quad B = \begin{bmatrix} 0.1 \\ -0.1 \\ 0.2 \end{bmatrix}.$$

Find $K^\star$ and the optimal average stage cost $J^\star$ (or more accurately, very good approximations of them, by iteration).

2. *Bi-directional supply chain via LQR.* In this problem we manage the flow of a single commodity across a chain consisting of n warehouses and n links connecting the warehouses in a chain. The warehouses and links are numbered $1, \dots, n$. Each warehouse has a target amount of the commodity to be stored. The state variable $(x_t)_i$ is the *deviation* of the amount of the commodity stored at warehouse i at time t from the target amount, so it can have positive or negative values.

Link i goes from warehouse $i-1$ to warehouse i, for $i = 2, \dots, n$, and link 1 goes from an external supply into warehouse 1. The link flows $(u_t)_i$ can have either sign. When $(u_t)_i$ is positive, it means that an amount $(u_t)_i$ of the commodity is shipped in the direction of the link; when $(u_t)_i$ is negative, it means that an amount $-(u_t)_i$ of the commodity is shipped in the direction opposite the link. (Since the commodity can be shipped either direction along the supply chain, we call it a bi-directional supply chain.)

A demand d_t is removed from warehouse n. This too can be positive or negative, with positive meaning the commodity is taken out of warehouse n and negative meaning it is put into warehouse n.

The dynamics is

$$(x_{t+1})_i = (x_t)_i + (u_t)_i - (u_t)_{i+1}, \quad i = 1, \ldots, n-1,$$

and, for warehouse n,

$$(x_{t+1})_n = (x_t)_n + (u_t)_n - d_t.$$

We assume that d_t are independent with $d_t \sim \mathcal{N}(0, \sigma^2)$.

The cost to transport the commodity across link i is $(\rho/2)(u_t)_i^2$, where $\rho > 0$ is a parameter. The cost at warehouse i for deviating from the target storage amount is $(1/2)(x_t)_i^2$. The total stage cost is therefore $(1/2)\|x_t\|_2^2 + (\rho/2)\|u_t\|_2^2$. The goal is to find a policy $u_t = \mu(x_t)$ that minimizes average stage cost.

Finally, we get to the question. Consider the specific case with $n = 4$, $\sigma = 1$ and $\rho = 0.1$. Give the optimal policy, and report the associated average stage cost. Simulate the system to verify your answer.

3. *Linear-quadratic stochastic control.* Consider a discrete-time dynamical system with random linear dynamics:

$$x_{t+1} = A_t(w_t)x_t + B_t(w_t)u_t + c_t(w_t), \qquad t = 0, \ldots, T-1; \qquad x_0 = 0.$$

where $x_t \in \mathbb{R}^n$ is the state, $u_t \in \mathbb{R}^m$ is the input, and $w_t \in \{1, \ldots, k\}$ is the disturbance. Suppose the system incurs random quadratic costs:

$$g_t(x_t, u_t, w_t) = \frac{1}{2} \begin{bmatrix} x_t \\ u_t \end{bmatrix}^T P_t(w_t) \begin{bmatrix} x_t \\ u_t \end{bmatrix} + q_t(w_t)^T \begin{bmatrix} x_t \\ u_t \end{bmatrix} + \frac{1}{2} r_t(w_t), \qquad t = 0, \ldots, T-1,$$

$$g_T(x_T) = \frac{1}{2} x_T^T P_T x_T + q_T^T x_T + \frac{1}{2} r_T,$$

where $P_0, \ldots, P_{T-1} \succeq 0$ and $P_T \succ 0$. In this problem you will write a `MATLAB` implementation of the calculus of affine and quadratic functions, and use your code to solve a linear-quadratic stochastic control problem.

(a) Download the `MATLAB` files `linear_function.m` and `quadratic_function.m`. The file `linear_function.m` gives a complete implementation of the subset of the calculus of linear functions that we need for value iteration. (Actually, it contains much more than we need; the extra code is intended to give you an example of how to implement a class, and you should examine it carefully.) The file `quadratic_function.m` gives a partial implementation of the subset of the calculus of linear functions that we need for value iteration. You need to complete the following five functions.

- `quadratic_plus_quadratic`, compute the sum of two quadratic functions
- `constant_times_quadratic`, compute a scalar multiple of a quadratic functions

- `evaluate_quadratic`, evaluate a quadratic function at a point
- `quadratic_precompose_linear`, precompose a quadratic function with a linear function
- `partial_minimization`, compute the partial minimization of a strictly convex quadratic

Note that the class is structured so that `MATLAB`'s default operators are overloaded: for example, we can write `qf1 + qf2` to compute the sum of quadratic functions `qf1` and `qf2`, and we can write `qf(lf)` for the composition of the quadratic function `qf` and the linear function `lf`. This overloading is handled in the starter code that we give you, and you do not need to worry about it, although you are encouraged to read the starter code thoroughly in order to understand how it works. You can create a quadratic function using the constructor: `qf = quadratic_function(P,q,r)`. You do not need to do any error checking or input validation.

(b) The file `lqsc_example_data.m` defines an instance of a linear-quadratic stochastic control problem. In particular, it defines the following variables.

- `n`, the dimension of the state space
- `m`, the dimension of the input space
- `k`, the number of values taken by the disturbances
- `T`, the time horizon
- `p`, a $T \times k$ array such that `p(t+1,w)` is the probability that $w_t = w$
- `A`, `B`, `c`, `P`, `q` and `r`, $T \times k$ cell arrays such that, for example, `A{t+1,wt}` is $A_t(w_t)$
- `pT`, `qT` and `rT`, the parameters of the terminal cost function
- `x0`, the initial state

Use your code from (a) to find the optimal controllers and value functions for this instance of the linear-quadratic stochastic control problem. Report the optimal expected total cost $J^\star$.

Hint. You can store the controllers and value functions in cell arrays.