Final exam

This is a 24 hour take-home final exam. Please turn it in at Bytes Cafe in the Packard building 24 hours after you pick it up.

Please read the following instructions carefully.

- You may use any books, notes, or computer programs (*e.g.*, Matlab), but you may not discuss the exam with anyone until March 17, after everyone has taken the exam. The only exception is that you can ask the TAs or Stephen Boyd for clarification, by emailing to the staff email address, `ee363-win0809-staff@lists.stanford.edu`.
- Please check your email a few times during the exam, just in case we need to send out a clarification or other announcement. It's unlikely we'll need to do this, but you never know.
- Since you have 24 hours, we expect your solutions to be legible, neat, and clear. Do not hand in your rough notes, and please try to simplify your solutions as much as you can. We will deduct points from solutions that are technically correct, but much more complicated than they need to be. We will deduct points when the logic is hard to follow.
- Attach the official exam cover page (available when you pick up or drop off the exam) to your exam, and assemble your solutions to the problems in order, *i.e.*, problem 1, problem 2, ..., problem 5. Start each solution on a new page. Do not collect all plots or code (for example) at the end of the exam; plots for problem 3 (say) should be with your solution to problem 3.
- Please make a copy of your exam before handing it in. We have never lost one, but it might occur.
- When a problem involves some computation (say, using Matlab), we do not want just the final answers. We want three things:
  - A clear discussion and justification of your approach and method, in mathematical terms, *i.e.*, using matrices, matrix multiplication, eigenvalues, singular value decomposition, etc. In your mathematical description you *cannot* refer to Matlab operators, such as the backslash operator.
  - A short discussion of how you solved the problem using Matlab (or some equivalent), and the source code that produces the result.
  - The final numerical result. Be sure to show us your verification that your computed solution satisfies whatever properties it is supposed to, at least up to numerical precision. For example, if you compute a symmetric matrix P that is

supposed to satisfy $A^TP + PA < 0$ and $P > 0$, you should verify this using the commands `max(eig(A'*P+PA))` and `min(eig(P))`, and showing us the results. (The first should be negative; the second positive.)

- Some problems require you to download and run a Matlab file to generate the data needed. These files can be found at the URL

  ```
  http://www.stanford.edu/class/ee363/finalmfiles/FILENAME
  ```

  where you should substitute the particular filename (given in the problem) for `FILENAME`. *There are no links on the course web page pointing to these files, so you'll have to type in the whole URL yourself.*

- We are not concerned with fine details such as checking if the constraint qualification holds in an S-procedure argument, or worrying too much about boundary issues, *i.e.*, the distinction between strict and nonstrict inequalities. At the same time, we require complete clarity as to what you are arguing or doing in your solutions.

1. *Finding an invariant ellipsoid containing some points but not others.*

   (a) Explain how to find an ellipsoid $\mathcal{E} \subseteq \mathbf{R}^n$, centered at 0, that is invariant for $\dot{x} = Ax$, and satisfies $z^{(i)} \in \mathcal{E}$, $i = 1, \dots, m$, and $w^{(i)} \notin \mathcal{E}$, $i = 1, \dots, p$. You are given the data A, $z^{(i)}$, and $w^{(i)}$. Your answer can involve any of the methods from the course, *e.g.*, Lyapunov and Riccati equations, LMIs, etc.

   (b) Carry out the method described in part (a) on the numerical instance with data given in `inv_elp_data.m`. There are many valid numerical solutions, so you must show us the script you used, along with corresponding output that verifies that the ellipsoid you found satisfies all the required conditions. For example, to verify that a matrix `Z` is positive definite, say, you can use `min(eig(Z))`, and we need to see the corresponding output (which should be a positive number).

2. *Certifying performance with control-Lyapunov controller.*

   A system has the form
   $$x_{t+1} = A_{i(t)} x_t, \quad t = 1, 2, \dots,$$
   where $x_t \in \mathbf{R}^n$ is the state, $i(t) \in \{1, \dots, k\}$ gives the dynamics matrix used in time period t, and $A_1, \dots, A_k$ are given. In this problem, you can *choose* $i(t)$; you can think of $i(t)$ as the (discrete) input to the system at time period t.

   The following control-Lyapunov scheme is used. We are given a matrix $P = P^T > 0$. At time period t, we choose $i(t)$ to minimize $V(x_{t+1})$, where $V(z) = z^T P z$. (If a tie occurs, you can choose any index which achieves the minimum value.)

   The goal is to certify performance of this control scheme; specifically, to verify that, for any x_t, we have
   $$V(x_{t+1}) \leq \alpha V(x_t),$$
   where α is a given value. (If $\alpha < 1$, this shows the closed-loop system is globally asymptotically stable.) 'Exhaustive simulation' suggests this inequality holds for all $x_t \in \mathbf{R}^n$; your job is to *prove* that it holds.

   (a) Describe a computationally tractable method that can verify the inequality above holds for all $x_t \in \mathbf{R}^n$. Your method does not have to always work; but we will give little or no credit for obvious or silly conditions, such as that $A_i^T P A_i \leq \alpha P$ for some i.

   (b) Carry out your method on the problem instance given in `clf_data.m`. You must submit your Matlab code, along with the corresponding output that verifies your certificate.

3. *Optimal time series interpolation.* A scalar time series $y_0, y_1, \ldots$ is the output of a linear Gauss-Markov process

$$x_{t+1} = Ax_t + w_t, \qquad y_t = Cx_t + v_t, \quad t = 0, 1, 2, \ldots,$$

where $y_t \in \mathbf{R}$, w_t are IID $\mathcal{N}(0, W)$, v_t are IID $\mathcal{N}(0, V)$ (and independent of all w_t), and $x_0 = 0$.

You are given the numbers y_{T_0} and y_{T_1}, with $0 \leq T_0 < T_1 - 1$. Your goal is to estimate y_t for $T_0 < t < T_1$, *i.e.*, to (approximately) interpolate the time series for the samples that occur between T_0 and T_1. Let $\hat{y}_t$ denote the minimum mean-square error estimate of y_t, for $T_0 < t < T_1$, given y_{T_0} and y_{T_1}.

(a) Explain how to find $\hat{y}_t$, as well as the RMS estimation error $\sigma_t = \left(\mathbf{E}(y_t - \hat{y}_t)^2\right)^{1/2}$.

(b) Plot $\hat{y}_t \pm \sigma_t$ for the specific problem instance with

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \qquad C = \begin{bmatrix} 0.5 & -0.8 & 1 \end{bmatrix},$$

$W = 0.1I$, $V = 0.1$, $T_0 = 5$, $T_1 = 35$, $y_{T_0} = -4$ and $y_{T_1} = 4$. (To plot a figure with error bars, you can use the Matlab function `errorbar`.)

4. *Optimal control with random dynamics matrix.* Consider a linear discrete-time system

$$x_{t+1} = A_t x_t + B u_t, \quad t = 1, \ldots, N-1,$$

where

$$A_t = \begin{cases} A^{(1)}, & \text{with probability } 1/2 \\ A^{(2)}, & \text{with probability } 1/2. \end{cases}$$

In other words, at each time period t, the dynamics matrix assumes the value $A^{(1)}$ or the value $A^{(2)}$, with equal probability. The cost J, which is to be minimized, is

$$J = \mathbf{E}\left(\sum_{t=1}^{N-1} \left(x_t^T Q x_t + u_t^T R u_t \right) + x_N^T Q x_N \right),$$

where $Q = Q^T \geq 0$ and $R = R^T > 0$ are state and input cost matrices, respectively. The matrices $A^{(1)}$, $A^{(2)}$, B, Q, and R are given.

We will consider two different scenarios.

- At each time period t, you must commit to (*i.e.*, choose) u_t knowing only x_t, *but not* A_t. (In other words: the input is chosen *before* A_t is revealed.)
- At each time period t, you choose u_t knowing x_t *and* A_t. (In other words: the input is chosen *after* A_t is revealed.)

(In both cases, you do not know $A_{t+1}, \ldots, A_{N-1}$.)

Now, the question.

(a) Show that the optimal control in the first scenario has the form $u_t^\star = K_t x_t$, for some matrices $K_1, \ldots, K_{N-1}$. Explain how to find the matrices $K_1, \ldots, K_{N-1}$.

(b) Show that the optimal control in the second scenario has the form

$$u_t^\star = \begin{cases} K_t^{(1)} x_t, & A_t = A^{(1)}, \\ K_t^{(2)} x_t, & A_t = A^{(2)}, \end{cases}$$

for some matrices $K_1^{(1)}, \ldots, K_{N-1}^{(1)}$, $K_1^{(2)}, \ldots, K_{N-1}^{(2)}$. Explain how to find these matrices.

(c) Find the optimal value of J for the two scenarios for the specific problem instance with

$$A^{(1)} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \quad A^{(2)} = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 1.3 \\ 0 & -2 \\ 1 & 0 \end{bmatrix},$$

$Q = I$, $R = I$, $x_1 = (1, 1, 1)$, and $N = 50$.

5. *Control with unreliable actuators.* A control system is modeled as

$$x_{t+1} = Ax_t + B\Delta_t u_t, \qquad u_t = Fx_t,$$

for $t = 0, 1, \ldots$. Here $x_t \in \mathbf{R}^n$ is the plant state, and $u_t \in \mathbf{R}^m$ is the actuator signal. The matrices A, B, and F, and the initial state x_0, are given.

The matrix Δ_t is used to model actuator failures. It is diagonal, and satisfies $(\Delta_t)_{ii} \in [0, 1]$ for all t and all i, but is otherwise unknown. $(\Delta_t)_{ii} = 1$ means the ith actuator is functioning correctly at time period t; $(\Delta_t)_{ii} = 0$ means the ith actuator has failed completely at time period t; $(\Delta_t)_{ii} = 0.8$, for example, means the ith actuator has suffered a 20% reduction in effectiveness at time period t. (This is sometimes called a *soft failure*.)

In this problem, your job is to establish an upper bound on the objective

$$J = \sum_{t=0}^{\infty} x_t^T Q x_t,$$

where $Q = Q^T > 0$ is given. In other words, you must find a number $\tilde{J}$ such that $J \leq \tilde{J}$ for any allowed values of $\Delta_1, \Delta_2, \ldots$.

(a) Describe an effective (*i.e.*, tractable) method for finding an upper bound on J. You method does not have to give the best possible bound, but if there is the possibility of optimizing your bound, please do so. We will give no credit for trivial bounds such as $\tilde{J} = \infty$, or bounds that are not tractable to compute. Your answer can involve any of the methods from the course, *e.g.*, LQR, Lyapunov and Riccati equations, Kalman filter, LMIs, *etc.*.

(b) Carry out your method for the problem instance with data given in the file `unrelact_data.m`. Report the value of your bound for this problem instance. For comparison, give the value of J when there are no actuator failures (*i.e.*, when $\Delta_t = I$ for all t), and when all actuators completely fail all the time (*i.e.*, when $\Delta_t = 0$ for all t).