EE363 Prof. S. Boyd
3/17/06–3/18/06 or 3/18/06–3/19/06

Final exam

This is a 24 hour take-home final exam. Please return it in room 244 of 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 20, 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.
- Please check your email a few times during the exam, just in case we need to send out a clarification or other announcement.
- Please address email inquiries to `ee363-win0506-staff@lists.stanford.edu`. This forwards the mail to the professor and the TAs. In particular, please do not use Stephen Boyd's or the TAs' individual email addresses.
- 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 8. Start each solution on a new page.
- **We are asking you to do only SIX out of the EIGHT problems. You MUST indicate on the exam cover page which SIX problems you did.** If you don't, we'll just read your first six problems. (All problems have equal weight.)
- 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/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.*

1. *Integral control design via LQR.* We consider the LDS $\dot{x} = Ax + Bu$, $y = Cx$, where $A \in \mathbf{R}^{n\times n}$, $B \in \mathbf{R}^{n\times p}$, and $C \in \mathbf{R}^{m\times n}$. You can think of $y(t)$ as some sort of deviation or error, that we'd like to drive to zero by a good choice of u.

   We define the signal z to be

$$z(t) = \int_0^t y(\tau)\, d\tau,$$

   which is the running integral (componentwise) of the error signal. Now we define the cost function

$$J = \int_0^\infty \left(y(\tau)^T y(\tau) + \rho u(\tau)^T u(\tau) + \gamma z(\tau)^T z(\tau) \right)\, d\tau,$$

   where ρ and γ are given positive constants. This is like an LQR cost function, with the addition of a term that penalizes the integral of the norm squared of the integral of the error. (Do not say that out loud too quickly.)

   (a) Show that the input u that minimizes J can be expressed in the form

$$u(t) = K_{\mathrm{P}} x(t) + K_{\mathrm{I}} z(t),$$

   where $K_{\mathrm{P}} \in \mathbf{R}^{p\times n}$ and $K_{\mathrm{I}} \in \mathbf{R}^{p\times m}$. The matrix $K_{\mathrm{P}} \in \mathbf{R}^{p\times n}$ is called the *proportional state feedback gain*, and the matrix $K_{\mathrm{I}} \in \mathbf{R}^{p\times m}$ is called the *integral output gain*.

   Be sure to explain how to find the two matrices K_{P} and K_{I}. You can use any of the ideas we've used in the course: Lyapunov and Sylvester equations, Riccati equations, LMIs, SDPs, etc.

   (b) Now consider the specific case

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & -3 & 7 & 2 \end{bmatrix}, \qquad B = \begin{bmatrix} 2 & -3 \\ 4 & 5 \\ -1 & 1 \\ 1 & -4 \end{bmatrix}, \qquad C = [1\ 0\ 0\ 0].$$

   with cost parameters $\rho = \gamma = 1$. Find K_{P} and K_{I}.

   Evaluate J (with the optimal u) for $x(0) = (1, 0, 0, 0)$.

2. *Observer with saturating sensor.* We consider a system of the form

$$\dot{x} = Ax + Bu, \qquad y = Cx,$$

where $A \in \mathbf{R}^{n\times n}$, $B \in \mathbf{R}^{n\times m}$, and $C \in \mathbf{R}^{1\times n}$. To estimate the state $x(t)$, based on observation of the input signal u and a measurement related to the scalar output signal y, we use a replica of the system, with an extra feedback input e:

$$\dot{\hat{x}} = A\hat{x} + Bu + Le, \qquad \hat{y} = C\hat{x},$$

where $L \in \mathbf{R}^n$ is the observer feedback gain. In the usual setup, $e(t) = y(t) - \hat{y}(t)$, *i.e.*, e is the output prediction error, and L is chosen so that $A - LC$ is stable. This ensures that the state estimation error $\tilde{x}(t) = x(t) - \hat{x}(t)$ converges to zero, since $\dot{\tilde{x}} = (A - LC)\tilde{x}$.

In this case, however, the measurement we have is the *saturated* output prediction error,

$$e(t) = \mathbf{sat}(y(t) - \hat{y}(t)).$$

(The signals e, y, and $\hat{y}$ are all scalar.) In other words, we get the actual prediction error when it is no more than one in magnitude; otherwise, we get only the sign of the prediction error.

The goal is to find an observer feedback gain L that gives quick convergence of the state estimation error $\tilde{x}(t)$ to zero, for any input u.

You can convince yourself (and it is true) that if A is not stable, then no matter how you choose L, we won't have $\tilde{x}(t) \to 0$ for all $x(0)$, $\hat{x}(0)$, and inputs u. So we're going to need A to be stable. This means that one possible choice for L is zero, in which case the state estimation error satisfies the stable linear dynamics $\dot{\tilde{x}} = A\tilde{x}$, and therefore converges to zero. (This is called an *open-loop* observer.) By using a nonzero L, we hope to speed up the estimator convergence, at least when the output error measurement isn't saturated. When the output prediction error isn't saturated, the state estimation error dynamics is given by $\dot{\tilde{x}} = (A - LC)\tilde{x}$, so we'd like to choose L to make these dynamics fast, or at least, faster than the open-loop dynamics A.

To certify the convergence of $\tilde{x}(t)$ to zero, we also seek a quadratic Lyapunov function $V(\tilde{x}) = \tilde{x}^T P\tilde{x}$ with the following properties:

- $P > 0$.
- $\dot{V}(\tilde{x}) < -\alpha V(\tilde{x})$ for all $x \neq \hat{x}$ and u.
- $\dot{V}(\tilde{x}) \leq -\beta V(\tilde{x})$ for all x, $\hat{x}$, and u, provided $|y(t) - \hat{y}(t)| \leq 1$.

Here, $\beta \geq \alpha > 0$ are given parameters that specify required convergence rates for the state estimation error in the saturated and unsaturated cases, respectively.

Note that the first two conditions above ensure that $\tilde{x}(t) \to 0$ with an exponential rate at least $\alpha/2$. The last condition guarantees that in the unsaturated mode, the

convergence is exponential, with rate at least $\beta/2$. The choice of α is limited by the dynamics of A (specifically, we must have $\alpha < -2\max_i \Re\lambda_i(A)$). Presumably, we have $\beta > -2\max_i \Re\lambda_i(A)$, which means that when saturation doesn't occur, the observer converges faster than an open-loop observer.

(a) Explain how to find P and L, given the problem data A, B, C, α, and β, or determine that no such matrices exist, using LMIs. Describe your method clearly and concisely; we won't read pages and pages of discussion.

(b) Carry out your method for the specific problem instance

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -40 & -31 & -19 & -5 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & -2 \\ 3 & 1 \\ -1 & 1 \\ 1 & -1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 2 & 1 & 3 \end{bmatrix},$$

with $\alpha = 1$, $\beta = 2.5$.

3. *Mean-square stability of a randomly perturbed linear system.* We consider the time-varying linear dynamical system

$$x(t+1) = (A + \Delta(t))x(t),$$

where $A \in \mathbf{R}^{n\times n}$, and $\Delta(t)$ is a random matrix with IID entries, $\Delta_{ij}(t) \sim \mathcal{N}(0, \sigma^2)$, with $\Delta(t)$ and $\Delta(s)$ independent for $t \neq s$. The initial state is also random, independent of $\Delta(t)$, with $\mathbf{E}\, x(0) = 0$, $\mathbf{E}\, x(0)x(0)^T = \Sigma(0)$. Such a system is sometimes called one with *multiplicative* noise (since the noise terms multiply the state), to distinguish it from the Gauss-Markov model, which is a linear dynamical system with *additive* noise (since the noise terms are added to the state).

It is easy to show that $\mathbf{E}\, x(t) = 0$ for all t. We let $\Sigma(t)$ denote $\mathbf{E}\, x(t)x(t)^T$, the covariance of $x(t)$. The system is said to be *mean-square stable* if $\Sigma(t) \to 0$ as $t \to \infty$, for any initial state covariance $\Sigma(0)$.

(a) Show that $\Sigma(t)$ satisfies a *linear* recursion, *i.e.*,

$$\Sigma(t+1) = \psi(\Sigma(t)),$$

where ψ is a linear function that maps symmetric matrices to symmetric matrices. Give an explicit description of the function ψ, in terms of A and σ.

Hint: Be sure to check that ψ reduces to the Lyapunov operator $\mathcal{L}(X) = AXA^T$ when $\sigma = 0$.

(b) Show that if the randomly perturbed system is mean-square stable for σ_0, then it is mean-square stable for any σ with $\sigma \leq \sigma_0$. (Of course σ_0 and σ are both nonnegative here.)

Show that if $\sigma \geq 1$, the system is not mean-square stable (for any A).

Thus, there exists a critical value $\sigma_{\text{crit}} \in [0, 1]$ such that the randomly perturbed linear system is mean-square stable for $\sigma < \sigma_{\text{crit}}$. (We allow the possibility $\sigma_{\text{crit}} = 0$, which means the system is mean-square stable for no value of σ; this happens, for example, when A is not stable.)

(c) Find σ_{crit} for the specific case

$$A = \begin{bmatrix} 1 & 1 & 1 \\ -0.5 & 0 & 0 \\ 0 & -0.5 & 0 \end{bmatrix}.$$

You can do this numerically, if you like; we'd like σ_{crit} to two significant figures. Of course, you must explain your method.

Plot $\mathbf{E}\, \|x(t)\|^2 = \mathbf{Tr}\, \Sigma(t)$ versus t, for $t = 0, \ldots, 10$, with $\Sigma(0) = I$, and $\sigma = 0.5\sigma_{\text{crit}}$. Repeat for $\sigma = 2\sigma_{\text{crit}}$.

Simulate a few trajectories x versus t, for $t = 0, \ldots, 10$, with $\sigma = 0.5\sigma_{\text{crit}}$, and also for $\sigma = 2\sigma_{\text{crit}}$.

Remark/warning: The distribution of $x(t)$, for $t \geq 1$, is highly non Gaussian, so it's possible that when you simulate a trajectory, you won't see what you're expecting (*i.e.*, divergence or convergence). If this happens, simulate another one.

4. *Bounds on individual state variables using a quadratic Lyapunov function.* Consider the time-varying LDS $\dot{x}(t) = A(t)x(t)$, where $A(t) \in \{A_1, \ldots, A_K\}$ and $x(t) \in \mathbf{R}^n$. In this problem your goal is to guarantee that $|x_i(t)| \leq \gamma$, $i = 1, \ldots, n$, for all $t \geq 0$, provided $|x_i(0)| \leq 1$, $i = 1, \ldots, n$. In other words, the goal is to show that if a trajectory starts in the unit box $\mathcal{B} = \{z \mid |z_i| \leq 1,\ i = 1, \ldots, n\}$, it cannot leave $\gamma\mathcal{B}$, the unit box scaled by the factor γ. (Clearly we must have $\gamma \geq 1$.)

   You will provide such a guarantee using the positive definite quadratic Lyapunov function $V(z) = z^T P z$. We require $\dot{V}(z) \leq 0$ for all z, and for any possible value of $A(t)$, which is equivalent to the set of of LMIs

$$A_i^T P + P A_i \leq 0, \quad i = 1, \ldots, K.$$

   This implies that the ellipsoid $\mathcal{E} = \{z \mid z^T P z \leq 1\}$ is invariant. If in addition $\mathcal{B} \subseteq \mathcal{E} \subseteq \gamma\mathcal{B}$, the desired property holds. (Why?)

   (a) Find the exact conditions on P under which

$$\mathcal{B} \subseteq \mathcal{E} \subseteq \gamma\mathcal{B}$$

   holds, and express them as a set of LMIs in P and $\lambda = \gamma^2$. Your condition can involve a large number of inequalities (say, 2^n).

   Hints:

   - $\mathcal{B} \subseteq \mathcal{E}$ holds if and only if the vertices of $\mathcal{B}$ are in $\mathcal{E}$. (Please justify this statement.)
   - To handle $\mathcal{E} \subseteq \gamma\mathcal{B}$, first show that $\max\{c^T z \mid z \in \mathcal{E}\} = \|P^{-1/2}c\|$. Now use Schur complements.

   (b) Now we return to the main problem. Show how to find the smallest value of γ, along with an associated P, that guarantees $x(t) \in \gamma\mathcal{B}$ for $t \geq 0$, whenever $x(0) \in \mathcal{B}$. You should reduce this problem to an SDP in a suitable set of variables.

   (c) Now consider the specific case

$$A_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -2 & -1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & -4 & -7 \end{bmatrix}, \quad A_3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & -3 & -3 \end{bmatrix}.$$

   Find the smallest value of γ, and an associated P, that satisfy the conditions found in part (a), as well

$$A_i^T P + P A_i \leq 0, \quad i = 1, \ldots, K.$$

5. *Optimal scaling of a positive matrix.* Suppose $A \in \mathbf{R}^{n\times n}$. We consider the problem of finding a nonsingular diagonal matrix D such that $\|DAD^{-1}\|$ is minimized. (The norm here is the matrix norm, *i.e.*, the maximum singular value.)

   You've encountered this problem (or a closely related one) several times. For example, in problem 6 of homework 7, you showed that the system $x(t+1) = \mathbf{sat}(Ax(t))$ is globally asymptotically stable if there exists a nonsingular diagonal D such that $\|DAD^{-1}\| < 1$. In problem 7 of homework 8, you showed how to find such a D using LMIs.

   In this problem, we consider the case where A is entrywise positive. For this case we can give a simple formula for the optimal D in terms of the left and right Perron-Frobenius eigenvectors of A. Show that the diagonal matrix D^{pf} defined by

$$D^{\mathrm{pf}}_{ii} = (w_i/v_i)^{1/2},$$

   where v and w are the right and left Perron-Frobenius eigenvectors of A, minimizes $\|DAD^{-1}\|$ over all nonsingular diagonal matrices.

   Since we've given you the 'answer', all you have to do is give the reasoning; your score will depend on how clear your argument is. (Clear does not mean long, by the way.)

   Hint: Start by showing that $\lambda_{\mathrm{pf}}(A) \leq \|DAD^{-1}\|$ for any nonsingular diagonal D; then show that equality holds here when $D = D^{\mathrm{pf}}$.

   Remark: You can find the optimal diagonal scaling for any matrix (not necessarily positive) by solving an SDP. A heuristic for finding a good scaling, if not the best, is to use $D_{ii} = (w_i/v_i)^{1/2}$, where v and w are the right and left Perron-Frobenius eigenvectors of $|A|$. By the result above, this heuristic gives the optimal scaling when A is elementwise positive.

6. *Optimal diagonal pre-conditioning.*

Part I. Iterative methods. We consider the problem of solving a set of linear equations $Ax = b$, where $A \in \mathbf{R}^{n\times n}$ is symmetric and positive definite. (Problems like this arise in several contexts, for example solving a least-squares problem, or an elliptic PDE.) When n is small enough (no more than a few thousand) we can easily solve the equations using standard methods. When n is very large (say, millions) we usually have to resort to an iterative method, which solves $Ax = b$ by generating a sequence $x(1), x(2), \ldots$ that converges to the solution $\bar{x} = A^{-1}b$. Each step of an iterative method involves multiplication of a vector by A, so we don't even have to form the matrix A and store it; it's good enough to have a fast method for multiplying A by a vector.

The simplest iterative method for solving $Ax = b$ starts with any $x(0)$, and then carries out the following iteration for $t = 0, 1, \ldots$

$$\begin{array}{ll} r(t) := b - Ax(t) & \text{(Evaluate the residual)} \\ x(t+1) := x(t) + \alpha r(t) & \text{(Update } x\text{)} \end{array}$$

Here $\alpha \in \mathbf{R}$ is an algorithm parameter.

(a) For what values of α does $x(t)$ converge to $\bar{x}$, for any b and any $x(0)$?
Hint: Express the answer in terms of the eigenvalues of A.

(b) Find the value of α that results in fastest asymptotic convergence of $x(t)$ to $\bar{x}$. Show that with this optimal value, the asymptotic convergence rate is given by

$$\frac{\kappa(A) - 1}{\kappa(A) + 1},$$

where $\kappa(A)$ is the condition number of A, *i.e.*, $\kappa(A) = \lambda_{\max}(A)/\lambda_{\min}(A)$.
Remarks.

- In practice α is chosen based on estimates or bounds on the eigenvalues of A, and not on the exact eigenvalues, which are typically unknown.
- There are many other (and often better) iterative methods for solving $Ax = b$. For example the conjugate gradient algorithm yields asymptotic convergence rate

$$\frac{\kappa(A)^{1/2} - 1}{\kappa(A)^{1/2} + 1},$$

which for large κ yields much faster convergence than the simple iterative method described above. It is general rule of thumb (and not always true) that the smaller the condition number $\kappa(A)$ is, the faster iterative methods converge.

Part II. Pre-conditioning.

Let D be a diagonal nonsingular matrix. Instead of solving $Ax = b$, we can solve the equation $DADy = Db$, to get $\bar{y}$, and then find $\bar{x}$ from $\bar{x} = D\bar{y}$. To solve $DADy = Db$ we use an iterative method, which in each iteration requires multiplication of a vector by DAD. But that's easy to do, provided it's easy to multiply by A: given the vector, we first multiply by D, then A, and then D again.

Why do this? Because if we choose D properly, the condition number of DAD can be considerably smaller than the condition number of A, so convergence of the iterative method for solving the pre-conditioned system $DADy = Db$ is much faster than the convergence of the iterative method applied to the original system $Ax = b$.

One widely used choice for the diagonal pre-conditioning matrix is

$$D_{ii}^{\text{diag}} = A_{ii}^{-1/2}, \quad i = 1, \ldots, n.$$

Using this pre-conditioner, the diagonal entries of DAD are all one.

Now we ask the question: given (symmetric, positive definite) $A \in \mathbf{R}^{n\times n}$, can you find the *optimal* diagonal pre-conditioner, *i.e.*, the diagonal nonsingular matrix D^{opt} that minimizes $\kappa(DAD)$?

(c) Show how to find D^{opt} using LMIs or SDP.

Remark: In most cases this is not a practical method, since solving the LMIs or SDP to find D^{opt} is much harder than solving $Ax = b$.

(d) Find D^{opt} for

$$A = \begin{bmatrix} 66 & 68 & 32 & 70 \\ 68 & 106 & 56 & 96 \\ 32 & 56 & 33 & 50 \\ 70 & 96 & 50 & 95 \end{bmatrix}.$$

Compare the condition numbers of A, $D^{\text{diag}}AD^{\text{diag}}$ and $D^{\text{opt}}AD^{\text{opt}}$.

Warning: Use `cond(X)` or `max(eig(X))/min(eig(X))` to compute the condition number of a symmetric positive definite matrix `X` in Matlab. Don't use `1/rcond(X)`, which will give you wrong results because `rcond` computes an *estimate* of the inverse of the condition number of a matrix.

7. *Simultaneous sensor selection and state estimation.* We consider a standard state estimation setup:

$$x(t+1) = Ax(t) + w(t), \qquad y(t) = C(t)x(t) + v(t),$$

where $A \in \mathbf{R}^{n\times n}$ is constant, but $C(t)$ can vary with time. The process and measurement noise are independent of each other and the initial state $x(0)$, with

$$x(0) \sim \mathcal{N}(0, \Sigma_0), \qquad w(t) \sim \mathcal{N}(0, W), \qquad v(t) \sim \mathcal{N}(0, V).$$

The standard formulas for the Kalman filter allow you to compute the next state prediction $\hat{x}(t|t-1)$, current state prediction $\hat{x}(t|t)$, and the associated prediction error covariances $\Sigma(t|t-1)$ and $\Sigma(t|t)$.

Now we are going to introduce a twist. The measurement matrix $C(t)$ is one of K possible values, *i.e.*, $C(t) \in \{C_1, \ldots, C_K\}$. In other words, at each time t, we have $C(t) = C_{i(t)}$. The sequence $i(t)$ specifies which of the K possible measurements is taken at time t. For example, the sequence $2, 2, \ldots$ means that $C(t) = C_2$ for all t; the sequence

$$1, 2, \ldots, K,\ 1, 2 \ldots, K,\ \ldots$$

is called *round-robbin*: we cycle through the possible measurements, in order, over and over again.

Here's the interesting part: *you* get to choose the measurement sequence $i(0), i(1), \ldots,$. You will use the following greedy algorithm. You will choose the sequence in order; having chosen $i(0), \ldots, i(t-1)$, you will choose $i(t)$ so as to minimize the mean-square prediction error associated with $\hat{x}(t|t)$. This is the same as choosing $i(t)$ so that $\mathbf{Tr}\,\Sigma(t|t)$ is minimized. Roughly speaking, at each step, you choose the sensor that results in the smallest mean-square state prediction error, given the sensor choices you've made so far, plus the one you're choosing.

Let's be very clear about this method for choosing $i(t)$. The choice of $i(0), \ldots, i(t-1)$ determines $\Sigma(t|t-1)$; then, $\Sigma(t|t)$ depends on $i(t)$, *i.e.*, which of $C_1, \ldots, C_K$ is chosen as $C(t)$. Among these K choices, you pick the one that minimizes $\mathbf{Tr}\,\Sigma(t|t)$.

This method does not require knowledge of the actual measurements $y(0), y(1), \ldots$, so we can determine the sequence of measurements we are going to make *before any data have been received.* In particular, the sequence can be determined ahead of time (at least up to some large value of t), and stored in a file.

Now we get to the question. You will work with the specific system with

$$A = \begin{bmatrix} -0.6 & 0.8 & 0.5 \\ -0.1 & 1.5 & -1.1 \\ 1.1 & 0.4 & -0.2 \end{bmatrix}, \qquad W = I, \qquad V = 0.1^2, \qquad \Sigma_0 = I,$$

and $K = 3$ with

$$C_1 = \begin{bmatrix} 0.74 & -0.21 & -0.64 \end{bmatrix}, \qquad C_2 = \begin{bmatrix} 0.37 & 0.86 & 0.37 \end{bmatrix}, \qquad C_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}.$$

(a) *Using one sensor.* Plot the mean-square current state prediction error $\mathbf{Tr}\,\Sigma(t|t)$ versus t, for the three special cases when $C(t) = C_1$ for all t, $C(t) = C_2$ for all t, and $C(t) = C_3$ for all t.

(b) *Round-robbin.* Plot the mean-square current state prediction error $\mathbf{Tr}\,\Sigma(t|t)$ versus t, using sensor sequence $1, 2, 3, 1, 2, 3, \ldots$.

(c) *Greedy sensor selection.* Find the specific sensor sequence generated by the algorithm described above. Show us the sequence, by plotting $i(t)$ versus t. Plot the resulting mean-square estimation error, $\mathbf{Tr}\,\Sigma(t|t)$, versus t. Briefly compare the results to what you found in parts (a) and (b).

In all three parts, you can show the plots over the interval $t = 0, \ldots, 50$.

To save you some time, we have created the file `sens_data.m`, which contains the problem data. The file also contains two lines, currently commented out, that implement a generic Kalman filter measurement and time update. You're welcome to use these, or to use or write your own.

8. *LQR with linear state cost.* We consider the stable system $x(t+1) = Ax(t) + Bu(t)$, where $A \in \mathbf{R}^{n\times n}$ and $B \in \mathbf{R}^{n\times m}$. We define the cost function J as

$$J = \sum_{t=0}^{\infty} \gamma^t \left(q^T x(t) + u(t)^T R u(t) \right),$$

where $q \in \mathbf{R}^n$, $R > 0$, and $\gamma < 1$. This is like a discounted LQR cost, except that here the cost associated with the state is *linear*, and not quadratic, as it is in LQR.

We define the cost-to-go or value function $V : \mathbf{R}^n \to \mathbf{R}$ as

$$V(z) = \inf_u J,$$

where the infimum (or minimum, if you like) is over all input sequences u, with $x(0) = z$.

Find V. If you can give an explicit formula for V, do so. If not, explain how to find V (*e.g.*, by solving a Riccati or other equation). Give the optimal input sequence $u^\star$. (You can give $u^\star(t)$ as a function of the state $x^\star(t)$, if you like.)