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

Final exam solutions

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
   $$u(t) = K_{\rm P}x(t) + K_{\rm I}z(t),$$
   where $K_{\rm P} \in \mathbf{R}^{p\times n}$ and $K_{\rm I} \in \mathbf{R}^{p\times m}$. The matrix $K_{\rm P} \in \mathbf{R}^{p\times n}$ is called the *proportional state feedback gain*, and the matrix $K_{\rm I} \in \mathbf{R}^{p\times m}$ is called the *integral output gain*.

   Be sure to explain how to find the two matrices $K_{\rm P}$ and $K_{\rm I}$. 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 $K_{\rm P}$ and $K_{\rm I}$.

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

   Solution:

(a) We augment the system state with $z(t) \in \mathbf{R}^p$, which satisfies $\dot{z} = Cx$, to create a new state

$$\tilde{x}(t) = \left[\begin{array}{c} x(t) \\ z(t) \end{array} \right] \in \mathbf{R}^{n+p}.$$

Then we have

$$\dot{\tilde{x}} = \tilde{A}\tilde{x} + \tilde{B}u, \qquad y = \tilde{C}\tilde{x},$$

where

$$\tilde{A} = \left[\begin{array}{cc} A & 0 \\ C & 0 \end{array} \right], \quad \tilde{B} = \left[\begin{array}{c} B \\ 0 \end{array} \right], \quad \tilde{C} = \left[\begin{array}{cc} C & 0 \end{array} \right].$$

Provided $z(0) = 0$, we have

$$J = \int_0^\infty \left(\tilde{x}(\tau)^T Q \tilde{x}(\tau) + u(\tau)^T R u(\tau) \right) \, d\tau,$$

where

$$Q = \left[\begin{array}{cc} C^T C & 0 \\ 0 & \gamma I \end{array} \right], \quad R = \rho I.$$

Our problem is now an infinite horizon LQR problem, and therefore the optimal control has the form

$$u = \tilde{K}\tilde{x}.$$

This has exactly the required form since

$$u = \tilde{K}\tilde{x} = [K_\mathrm{P} \; K_\mathrm{I}] \left[\begin{array}{c} x \\ z \end{array} \right] = K_\mathrm{P} x(t) + K_\mathrm{I} z(t).$$

We can find the solution by solving the ARE equation

$$\tilde{A}^T P + P\tilde{A} - P\tilde{B}R^{-1}B^T P + Q = 0,$$

to find P, and then setting

$$\tilde{K} = -R^{-1}\tilde{B}^T P.$$

To evaluate J for a specific initial state (of the original system) $x(0) \in \mathbf{R}^n$, we can use

$$J = \left[\begin{array}{c} x(0) \\ 0 \end{array} \right]^T P \left[\begin{array}{c} x(0) \\ 0 \end{array} \right].$$

(Recall that $P \in \mathbf{R}^{(n+p)\times(n+p)}$.) The zero (in $\mathbf{R}^p$) below $x(0)$ corresponds to the fact that the initial condition for the extra state component z, which gives the integral of the output, always starts at 0.

(b) The following Matlab code computes K_P, K_I, and the minimum cost J for $x(0) = (1, 0, 0, 0)$.

```
A=[ 0 1 0 0;
    0 0 1 0;
    0 0 0 1;
    1 -3 7 2];

B=[ 2 -3;
    4  5;
   -1  1;
    1 -4];

C=[1 0 0 0];

rho=1;
gamma=1;

Atilde=[A, zeros(4,1);
        C,  0];
Btilde=[B;zeros(1,2)];
Ctilde=[C 0];

Q=[C'*C ,     zeros(4,1);
   zeros(1,4) ,     gamma];
R=rho*eye(2);

P=are(Atilde,Btilde*inv(R)*Btilde',Q);
K=-inv(R)*Btilde'*P;

KP=K(:,1:4)
KI=K(:,5)

xtilde=[1;0;0;0;0];
J=xtilde'*P*xtilde
```

The output of the code is

```
KP =

   -1.0154   -0.7077    1.1781    0.6912
    0.7817   -0.6282    2.5298    1.4004


KI =
   -0.9604
    0.2785

J = 0.3720
```

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

Solution:

(a) We have

$$\begin{aligned} \dot{V} &= 2\tilde{x}^T P\dot{\tilde{x}} \\ &= 2\tilde{x}^T P(Ax + Bu - A\hat{x} - B\hat{x} - L\mathbf{sat}(Cx - C\hat{x})) \\ &= 2\tilde{x}^T P(A\tilde{x} - L\mathbf{sat}(C\tilde{x})). \end{aligned}$$

When the sensor is not saturated the condition $\dot{V}(\tilde{x}(t)) \leq -\beta V(\tilde{x})$ becomes

$$\tilde{x}^T(A^TP + PA - PLC - C^TL^TP)\tilde{x} + \beta\tilde{x}^TP\tilde{x} \leq 0.$$

This must hold for any $\tilde{x}$ with $|e| = |C\tilde{x}| \leq 1$. In particular, it must hold all small enough $\tilde{x}$, so we must have

$$A^TP + PA - PLC - C^TL^TP + \beta P \leq 0. \qquad (1)$$

(It's also possible to use the S-procedure here; see the end of this solution.)

When the sensor is saturated we need to have

$$2\tilde{x}^TP(A\tilde{x} - L\mathbf{sat}(C\tilde{x})) + \alpha x^TPx = \tilde{x}^T(2PA + \alpha P)\tilde{x} - 2\tilde{x}^TPL\mathbf{sat}(C\tilde{x}) \leq 0. \quad (2)$$

This must hold for all $\tilde{x}$ with $|C\tilde{x}| \geq 1$. Now let's see what happens when $\tilde{x}$ is really big. In this case the quadratic term, $\tilde{x}^T(2PA + \alpha P)\tilde{x}$, is much larger than the other term, which grows linearly with $\tilde{x}$; it follows that we must have $\tilde{x}^T(2PA + \alpha P)\tilde{x} \leq 0$. In other words, we must have

$$A^T P + PA + \alpha P \leq 0. \qquad (3)$$

(Again, it's possible to use the S-procedure here; see the end of this solution.)

Now we will show that if conditions (1) and (3) are satisfied then (2) is satisfied. In fact if

$$2\tilde{x}^T PL\mathbf{sat}(C\tilde{x}) > 0,$$

equation (2) is trivially satisfied. If instead $2\tilde{x}^T PL\mathbf{sat}(C\tilde{x}) \leq 0$, we have

$$\begin{aligned} \tilde{x}^T(2PA + \alpha P)\tilde{x} - 2\tilde{x}^T PL\mathbf{sat}(C\tilde{x}) &\leq \tilde{x}^T(2PA + \alpha P)\tilde{x} - 2\tilde{x}^T PLC\tilde{x} \\ &\leq \tilde{x}^T(2PA + \beta P)\tilde{x} - 2\tilde{x}^T PLC\tilde{x} \\ &\leq 0. \end{aligned}$$

We now have the required conditions on P and L:

$$P > 0, \qquad A^T P + PA + \alpha P \leq 0, \qquad A^T P + PA - PLC - C^T L^T P + \beta P \leq 0.$$

These conditions are not LMIs in P and L together (although they are LMIs in each of P or L, with other one fixed), because there are terms that involve the product of P and L.

But we can use a trick to convert these inequalities into LMIs. We can define a new vector R as

$$R = PL,$$

and rewrite the conditions as

$$\begin{aligned} P &> 0 \\ A^T P + PA + \alpha P &\leq 0 \\ A^T P + PA - RC - C^T R^T + \beta P &\leq 0. \end{aligned}$$

We now have LMIs with variables P and R. Once a solution for P and R has been computed we can find L as

$$L = P^{-1}R.$$

Finally, let's describe how you can use the S-procedure to write the required conditions on P and L. We can express the condition of the sensor not being saturated as $(C\tilde{x})^2 \leq 1$ or equivalently as

$$\begin{bmatrix} \tilde{x} \\ 1 \end{bmatrix}^T \begin{bmatrix} C^T C & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ 1 \end{bmatrix} \leq 0.$$

Therefore we need to have $\tilde{x}^T(A^TP + PA - PLC - C^TL^TP)\tilde{x} + \beta\tilde{x}^TP\tilde{x} \le 0$ or equivalently

$$\begin{bmatrix} \tilde{x} \\ 1 \end{bmatrix}^T \begin{bmatrix} A^TP + PA - PLC - C^TL^TP + \beta P & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ 1 \end{bmatrix} \le 0,$$

whenever $(C\tilde{x})^2 \le 1$. Using the S-procedure for quadratic functions we obtain the equivalent condition

$$\tau \begin{bmatrix} C^TC & 0 \\ 0 & -1 \end{bmatrix} - \begin{bmatrix} A^TP + PA - PLC - C^TL^TP + \beta P & 0 \\ 0 & 0 \end{bmatrix} \ge 0,$$

for some $\tau \ge 0$. Clearly the only possible value of τ is zero and therefore the above condition simplifies to $A^TP + PA - PLC - C^TL^TP + \beta P \le 0$, which is what we had above.

Let's now consider the case when the sensor is saturated. We have two possible situations. For $C\tilde{x} \ge 1$ we need to have $\tilde{x}^T(2PA + \alpha P)\tilde{x} - 2\tilde{x}^TPL \le 0$ and for $C\tilde{x} \le 1$ we need to have $\tilde{x}^T(2PA+\alpha P)\tilde{x}+2\tilde{x}^TPL \le 0$. We will only consider the first of the two conditions; the other one can be handled in the exact same way.

The inequality $C\tilde{x} \le 1$ can we written as

$$\begin{bmatrix} \tilde{x} \\ 1 \end{bmatrix}^T \begin{bmatrix} 0 & C^T/2 \\ C/2 & -1 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ 1 \end{bmatrix} \ge 0$$

and the inequality $\tilde{x}^T(2PA + \alpha P)\tilde{x} - 2\tilde{x}^TPL \le 0$ can be rewritten as

$$\begin{bmatrix} \tilde{x} \\ 1 \end{bmatrix}^T \begin{bmatrix} A^TP + PA + \alpha P & PL \\ L^TP^T & 0 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ 1 \end{bmatrix} \le 0.$$

Using the S-procedure for quadratic functions we obtain the equivalent condition

$$\begin{bmatrix} A^TP + PA + \alpha P & -PL \\ -L^TP^T & 0 \end{bmatrix} + \tau \begin{bmatrix} 0 & C^T/2 \\ C/2 & -1 \end{bmatrix} \le 0,$$

for some $\tau \ge 0$. Notice that the last inequality implies $A^TP + PA + \alpha P \le 0$, which is also what we had above.

(b) Since the LMIs above are homogeneous in P and R we can replace the condition $P > 0$ with $P \ge I$. The following MATLAB code solves for P and L.

```
A=[ 0   1   0   0;
    0   0   1   0;
    0   0   0   1;
  -40  -31 -19   -5];
```

```
B=[ 1,-2;
    3, 1;
   -1, 1;
    1, -1];

C=[1, 2, 1, 3];

beta=2.5;
alpha=1;

cvx_begin sdp
    variable P(4,4) symmetric
    variable R(4,1)
    P>=eye(4);
    A'*P+P*A-R*C-C'*R'+beta*P<=0;
    A'*P+P*A+alpha *P<=0;
cvx_end

cvx_status
cvx_optval;

P
L=P\R
```

The output of the code is

```
P =

   1.0e+03 *

    3.2790    1.8180    0.8327    0.1074
    1.8180    1.3749    0.5852    0.1189
    0.8327    0.5852    0.2638    0.0497
    0.1074    0.1189    0.0497    0.0152

L =

    6.7081
   -8.5296
  -20.1230
   95.3567
```

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.

Solution:

(a) Let $\bar{x}(t) = \mathbf{E}\, x(t)$. Then

$$\begin{aligned}
\bar{x}(t+1) &= \mathbf{E}(A + \Delta(t))x(t) \\
&= \mathbf{E}\, Ax(t) + \mathbf{E}\, \Delta(t)x(t) \\
&= A\bar{x}(t) + \mathbf{E}\, \Delta(t)\, \mathbf{E}\, x(t) \\
&= A\bar{x}(t).
\end{aligned}$$

In the third line here we use the fact that $x(t)$ and $\Delta(t)$ are independent, since $x(t)$ depends only on the random variables $x(0)$, $\Delta(0), \ldots, \Delta(t-1)$, which are independent of $\Delta(t)$. Since $\mathbf{E}\, x(0) = 0$, we have $\bar{x}(t) = 0$ for all t.

Now let's see how see how the covariance propagates.

$$\begin{aligned}
\Sigma(t+1) &= \mathbf{E}\, x(t+1)x(t+1)^T \\
&= \mathbf{E}(A + \Delta(t))x(t)x(t)^T(A + \Delta(t))^T \\
&= \mathbf{E}\, Ax(t)x(t)^T A^T + 2\, \mathbf{E}\, \Delta(t)x(t)x(t)A^T + \mathbf{E}\, \Delta(t)x(t)x(t)^T\Delta(t)^T \\
&= A\Sigma(t)A^T + \mathbf{E}\, \mathbf{E}\left[\Delta(t)x(t)x(t)^T\Delta(t)^T | \Delta(t)\right] \\
&= A\Sigma(t)A^T + \mathbf{E}\, \Delta(t)\Sigma(t)\Delta(t)^T.
\end{aligned}$$

In going from the third to the fourth line, we use

$$\mathbf{E}\, \Delta(t)x(t)x(t)A^T = (\mathbf{E}\, \Delta(t))\left(\mathbf{E}\, x(t)x(t)A^T\right) = 0,$$

by independence of $x(t)$ and $\Delta(t)$.

It remains to work out what $\mathbf{E}\, \Delta(t)\Sigma(t)\Delta(t)^T$ is. Note that the (i,j)th entry of $\Delta(t)\Sigma(t)\Delta(t)^T$ is

$$\begin{aligned}
(\Delta\Sigma\Delta)_{ij} &= \sum_{k=1}^{n}\sum_{l=1}^{n} \Delta_{ik}\Sigma_{kl}\Delta_{lj}^T \\
&= \sum_{k=1}^{n}\sum_{l=1}^{n} \Delta_{ik}\Sigma_{kl}\Delta_{jl},
\end{aligned}$$

where we have dropped the time index for ease of notation. Now we take the expectation of this big sum. Note that $\mathbf{E}\, \Delta_{ik}\Delta_{jl} = \sigma^2$ when $i = j$ and $k = l$; otherwise $\mathbf{E}\, \Delta_{ik}\Delta_{jl} = 0$. Thus $\mathbf{E}(\Delta\Sigma\Delta)_{ij} = 0$ unless $i = j$; and when $i = j$, we have

$$\mathbf{E}(\Delta\Sigma\Delta)_{ii} = \sigma^2\, \mathbf{Tr}\, \Sigma.$$

which implies that

$$\mathbf{E}\, \Delta(t)\Sigma(t)\Delta(t) = \sigma^2(\mathbf{Tr}\, \Sigma(t))I.$$

Thus we have

$$\Sigma(t+1) = A\Sigma(t)A^T + \sigma^2(\mathbf{Tr}\, \Sigma(t))I.$$

In other words, $\psi(X) = AXA^T + \sigma^2(\mathbf{Tr}\, X)I$. Note that ϕ is linear and does indeed reduce to $\mathcal{L}$ when $\sigma = 0$.

We can give a nice interpretation of ψ. The first term is the standard one you'd see when you multiply a random variable by A. The second term, which is always positive, add some extra covariance to $x(t)$.

(b) We will first show by induction that $\Sigma(t)$ is monotonically nondecreasing in σ. Let $\sigma_1 > \sigma_2$, and let $\Sigma_i(t)$ satisfy

$$\Sigma_1(t) = A\Sigma_1(t)A^T + \sigma_1^2(\mathbf{Tr}\,\Sigma_1(t))I \qquad \Sigma_2(t) = A\Sigma_2(t)A^T + \sigma_2^2(\mathbf{Tr}\,\Sigma_2(t))I,$$

with $\Sigma_1(t) = \Sigma_2(t) = \Sigma_0$. Since

$$\begin{aligned} \Sigma_1(1) &= A\Sigma_0A^T + \sigma_1^2(\mathbf{Tr}\,\Sigma_0)I, \\ \Sigma_2(1) &= A\Sigma_0A^T + \sigma_2^2(\mathbf{Tr}\,\Sigma_0)I, \end{aligned}$$

we have

$$\Sigma_1(1) - \Sigma_2(1) = (\sigma_1^2 - \sigma_2^2)(\mathbf{Tr}\,\Sigma_0)I \geq 0$$

because $\sigma_1 > \sigma_2$ and $\Sigma_0 \geq 0$. Now suppose that $\Sigma_1(t) \geq \Sigma_2(t)$. Then

$$\begin{aligned} \Sigma_1(t+1) &= A\Sigma_1(t)A^T + \sigma_1^2(\mathbf{Tr}\,\Sigma_1(t))I, \\ \Sigma_2(t+1) &= A\Sigma_2(t)A^T + \sigma_2^2(\mathbf{Tr}\,\Sigma_2(t))I. \end{aligned}$$

Then

$$\begin{aligned} \Sigma_1(t+1) - \Sigma_2(t+1) &= A(\Sigma_1(t) - \Sigma_2(t))A^T + \sigma_1^2(\mathbf{Tr}\,\Sigma_1(t))I - \sigma_2^2(\mathbf{Tr}\,\Sigma_2(t))I \\ &= A(\Sigma_1(t) - \Sigma_2(t))A^T + (\sigma_1^2 - \sigma_2^2)(\mathbf{Tr}\,\Sigma_1(t))I \\ &\quad + \sigma_2^2(\mathbf{Tr}\,\Sigma_1(t) - \mathbf{Tr}\,\Sigma_2(t))I \\ &\geq 0 \end{aligned}$$

because $\Sigma_1(t) \geq \Sigma(t)$, $\Sigma_1(t) \geq 0$ and $\sigma_1 > \sigma_2$ Thus we have shown that if $\sigma_1 > \sigma_2$ then $\Sigma_1(t) \geq \Sigma_2(t)$ for all t. Hence if for some $\sigma = \sigma_0$, $\Sigma(t) \to 0$ as $t \to 0$ then for all $\sigma \leq \sigma_0$, $\Sigma(t) \to 0$ as well.

To show the system is not mean-square stable for $\sigma \geq 1$, we look at the diagonal of $\Sigma(t)$, which satisfies

$$\mathbf{diag}(\Sigma(t+1)) = \mathbf{diag}(A\Sigma(t)A^T) + \sigma^2(\mathbf{Tr}\,\Sigma(t))\mathbf{1} \geq \sigma^2(\mathbf{Tr}\,\Sigma(t))\mathbf{1},$$

where the inequality is componentwise. It follows that $\mathbf{diag}(\Sigma(t)) \geq \sigma^{2t}(\mathbf{Tr}\,\Sigma(0))\mathbf{1}$. In fact, we can improve the bound: we can show that $\sigma_{\text{crit}} \leq 1/\sqrt{n}$.

(c) Let $s(t) = \text{vec}(\sigma(t))$. We found in the previous part that ψ is linear. This implies that the dynamics of $s(t)$ are governed by the linear dynamical system

$$s(t+1) = \tilde{A}s(t),$$

where $\tilde{A} \in \mathbf{R}^{n^2 \times n^2}$ is the matrix that represents ψ. In fact, there is a formula for $\tilde{A}$ that uses Kroneker products (which you probably don't know, we just mention this for fun):

$$\tilde{A} = A \otimes A + \sigma^2 \mathrm{vec}(I)\mathrm{vec}(I)^T.$$

This doesn't really matter; what's important is that s (or Σ) propagates according to a linear dynamical system, and mean-square stability is nothing more than stability of this LDS. In other words, the original system is mean-square stable if and only if all eigenvalues of $\tilde{A}$ have absolute value smaller than one.

Given a value of σ, we can check whether $\tilde{A}$ is stable by forming it, and checking whether all its eigenvalues have absolute value less than 1.

By the way, you can exploit the fact that $\Sigma(t)$ is symmetric, which means it has only $n(n+1)/2$ free variables in it. This gives a matrix $\bar{A}$ that is $n(n+1)/2 \times n(n+1)/2$, smaller than our $\tilde{A}$.

Some people evaluated stability by just running the iteration $\Sigma(t+1) = \psi(\Sigma(t))$. Others constructed the matrix $\tilde{A}$ and evaluated its eigenvalues.

To find σ_{crit}, we can just try various values of σ between 0 and 1. One way to do this is a bisection algorithm, which maintains an upper bound u and a lower bound $_l$ on σ_{crit}. We start with $u = 1$ and $l = 0$. Then we set $\sigma = (u + l)/2$ and check mean-square stability. If the perturbed system is mean-square stable, we set $l = \sigma$; otherwise we set $u = \sigma$. We repeat these steps until $u - l$ is small enough. (Note that we always have $l \leq \sigma_{\mathrm{crit}} \leq u$.)

The following Matlab script finds σ_{crit} by bisection and simulates the perturbed system for $\sigma = 0.8\sigma_{\mathrm{crit}}$ and $\sigma = 1.2\sigma_{\mathrm{crit}}$.

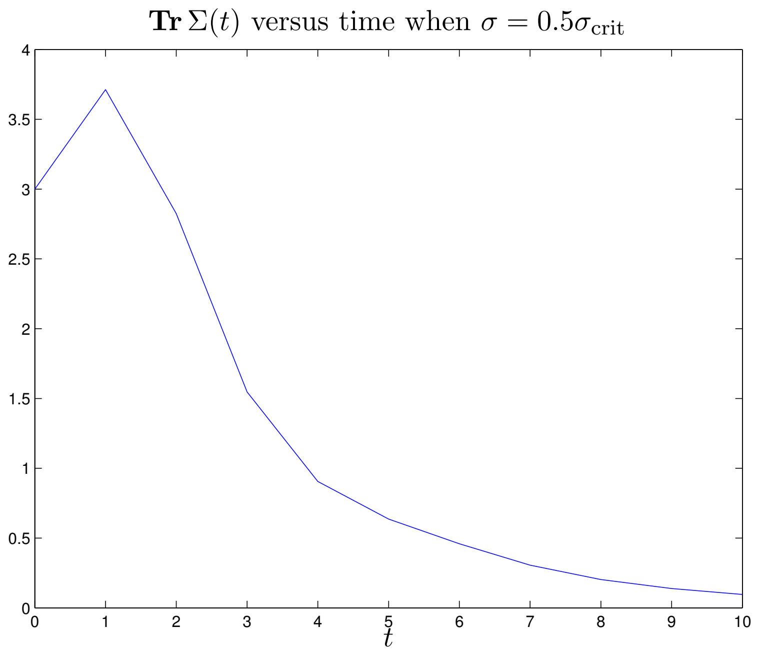
```
close all; clear all;
n = 3;
A = [ 1   1   1; ...
     -0.5 0   0; ...
      0  -0.5 0];
A_tilde = kron(A,A);

I = eye(n);
t = I(:);
z = t*t';

% finding sigma_crit by bisection
s_u = 1;                   % initial upper bound on sigma_crit^2
s_l = 0;                   % initial lower bound on sigma_crit^2
diff = s_u - s_l;
while (abs(diff)>0.001)
    sigma_crit = (s_u + s_l)/2;
    X = A_tilde + sigma_crit.^2*z;
    lambda = max(abs(eig(X)));
    if lambda < 1
        s_l = sigma_crit;
    else
        s_u = sigma_crit;
    end
    diff = s_u - s_l;
end

% simulating the linear system for the covariance matrix
sigma1 = 0.5*sigma_crit;
sigma2 = 2*sigma_crit;
T = 10;

% Case 1: sigma = sigma1
% initializing Sigma(t) with Sigma_0
S1 = I;
s1 = trace(S1);
for k = 1:T
    S1 = A*S1*A' + sigma1^2*trace(S1)*I;
    s1 = [s1 trace(S1)];
end

% Case 2: sigma = sigma2
% initializing Sigma(t) with Sigma_0
S2 = I;
s2 = trace(S2);
for k = 1:T
    S2 = A*S2*A' + sigma2^2*trace(S2)*I;
    s2 = [s2 trace(S2)];
end

figure(1);
plot(0:T,s1)
title('Tr(\Sigma(t)) vs time when \sigma=0.5\sigma_{crit}'); xlabel('t')
figure(2);
plot(0:T,s2')
title('Tr(\Sigma(t)) vs time when \sigma=2\sigma_{crit}'); xlabel('t')

% simulating the perturbed original system
% initializing with x(0) drawn from (0,I)
Nsamples = 3;
```

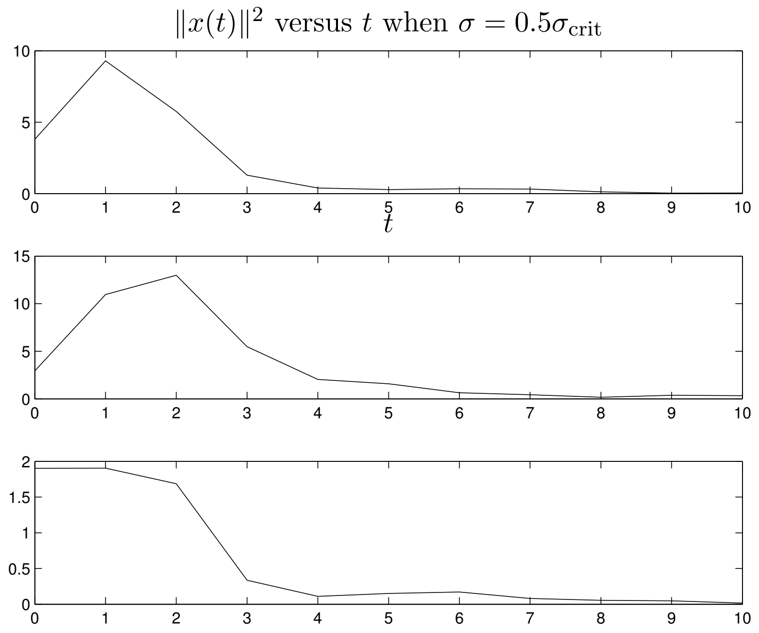
```
norm1 = zeros(Nsamples,T);
norm2 = zeros(Nsamples,T);
for i=1:Nsamples
    x0 = randn(n,1);
    % Case 1: sigma = sigma1
    x1 = x0;
    for k = 1:T+1
        norm1(i,k) = norm(x1)^2;
        Delta = sigma1*randn(n);
        x1 = (A+Delta)*x1;
    end

    % Case 2: sigma = sigma2
    x2 = x0;
    for k = 1:T+1
        norm2(i,k) = norm(x2)^2;
        Delta = sigma2*randn(n);
        x2 = (A+Delta)*x2;
    end
    figure(3);
    subplot(Nsamples,1,i);
    plot(0:T,norm1(i,:))
    figure(4);
    subplot(Nsamples,1,i);
    plot(0:T,norm2(i,:));
end

figure(3);
subplot(Nsamples,1,1)
title('\|x(t)\|^2 versus t when \sigma=0.5\sigma_{crit}');xlabel('t');

figure(4);
subplot(Nsamples,1,1)
title('\|x(t)\|^2 versus t when \sigma=2\sigma_{crit}');xlabel('t');
```

The Matlab script returns

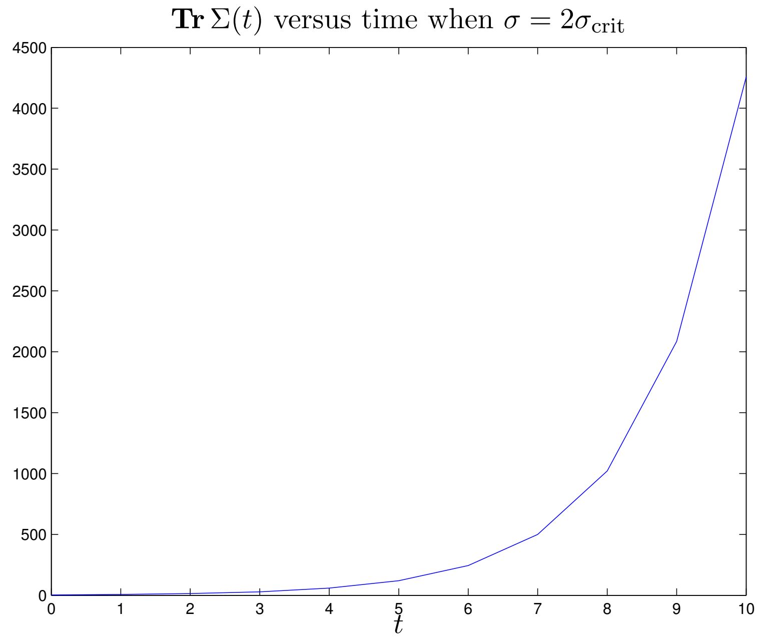
```
sigma_crit =
    0.3076
```

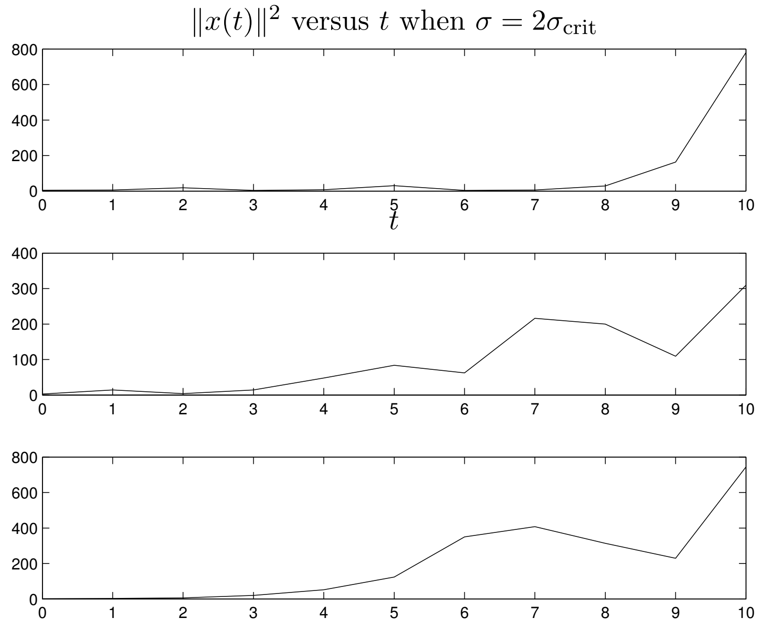
It also plots $\mathbf{Tr}\,\Sigma$ and the squared norm of $x(t)$ for 3 simulated trajectory of the randomly perturbed system in the case where $\sigma = 0.5\sigma_{\text{crit}}$





and in the case where $\sigma = 2\sigma_{\text{crit}}$





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

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

This implies that the ellipsoid $\mathcal{E} = \{z \mid z^T Pz \leq 1\}$ 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

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

Solution:

(a) Let's first look at the conditions under which $\mathcal{B} \subseteq \mathcal{E}$. For this to be the case, all the vertices of $\mathcal{B}$ must be in $\mathcal{E}$. $\mathcal{B}$ has 2^n vertices, which we will call v_i, $i = 1, \ldots, 2^n$. (These vectors are the 'Boolean' vectors, with each entry $+1$ or -1.) Thus the condition under which $\mathcal{B} \subseteq \mathcal{E}$ is

$$v_i^T P v_i \leq 1, \quad i = 1, \ldots, 2^n,$$

a set of linear inequalities in P. Because of symmetry, we only have to consider 2^{n-1} such inequalities.

Now let's look at the conditions under which $\mathcal{E} \subseteq \gamma\mathcal{B}$. To do this we will first prove the hint.

We want to find the maximum value of $c^T z$ subject to $z^T P z \le 1$. Using the substitution $w = P^{1/2}z$, this is equivalent to maximizing $(P^{-1/2}c)^T w$ subject to $\|w\|^2 \le 1$. By Cauchy-Schwarz, the maximizing solution to this problem is given by $w^* = (1/\|P^{-1/2}c\|)P^{-1/2}c$, which gives a maximum value of $\|P^{-1/2}c\|$, establishing the result given in the hint.

Now we turn to the conditions under which $\mathcal{E} \subseteq \gamma\mathcal{B}$. This is the case only if the maximum value of $e_j^T z$, subject to $z^T P z \le 1$, is at most γ, for all j (where e_j denotes the jth unit vector). Using the hint, the conditions are

$$\|P^{-1/2}e_j\| \le \gamma, \quad j = 1, \ldots, n.$$

This can be expressed several other ways. For example, we can write it as

$$\gamma^2 \ge \|P^{-1/2}e_j\|^2 = e_j^T P^{-1} e_j, \quad j = 1, \ldots, n.$$

We can write this compactly as

$$\left(P^{-1}\right)_{jj} \le \gamma^2, \quad j = 1, \ldots, n.$$

(b) The conditions under which $\mathcal{B} \subseteq \mathcal{E}$ are a set of linear inequalities in P and are therefore easy to express as a set of LMIs.

Let's look at the conditions under which $\mathcal{E} \subseteq \gamma\mathcal{B}$. These are of the form $\|P^{-1/2}e_j\| \le \gamma$. Equivalently we must have $\gamma^2 - e_j^T P^{-1} e_j \ge 0$. Since we already require that $P > 0$, because $V(z)$ is a positive definite function, we can use Schur complements to express these inequalities as

$$\begin{bmatrix} \gamma^2 & e_j^T \\ e_j & P \end{bmatrix} \ge 0, \quad j = 1, \ldots, n.$$

This is a set of LMIs in P and γ^2.

Finally, we want a P such that ellipsoids of the form $\mathcal{E}_\alpha = \{z \mid z^T P z \le \alpha\}$ are invariant under $\dot{x}(t) = A(t)x(t)$. Such a P must satisfy the LMIs

$$A_k^T P + P A_k \le 0, \quad k = 1, \ldots, K, \qquad P > 0.$$

Therefore the set of LMIs that P and γ^2 must jointly satisfy is

$$v_i^T P v_i \le 1, \quad i = 1, \ldots, 2^{n-1}$$

$$\begin{bmatrix} \gamma^2 & e_j^T \\ e_j & P \end{bmatrix} \ge 0, \quad j = 1, \ldots, n$$

$$A_k^T P + P A_k \le 0, \quad k = 1, \ldots, K$$

$$P > 0, \quad \gamma^2 \ge 0.$$

By the way, you can use the S-procedure to get a sufficient condition for $\mathcal{B} \subseteq \mathcal{E}$ that does not involve exponentially many inequalities. (This might be needed to get a practical method if, for example, $n = 30$.) We didn't ask you to do this, but we'll tell you what the condition is. We need to have $x^T P x \le 1$ whenever $x_i^2 \le 1$. We write these conditions as

$$\begin{bmatrix} x \\ 1 \end{bmatrix}^T \begin{bmatrix} -P & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix} \ge 0$$

whenever

$$\begin{bmatrix} x \\ 1 \end{bmatrix}^T \begin{bmatrix} -e_1 e_1^T & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix} \ge 0, \quad \ldots, \quad \begin{bmatrix} x \\ 1 \end{bmatrix}^T \begin{bmatrix} -e_n e_n^T & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix} \ge 0.$$

Using the S-procedure, we get a sufficient condition for this to occur: There exists $\tau_1, \ldots, \tau_n \ge 0$ such that

$$\begin{aligned} \begin{bmatrix} -P & 0 \\ 0 & 1 \end{bmatrix} &\ge \tau_1 \begin{bmatrix} -e_1 e_1^T & 0 \\ 0 & 1 \end{bmatrix} + \cdots + \tau_n \begin{bmatrix} -e_n e_n^T & 0 \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} -\mathbf{diag}(\tau) & 0 \\ 0 & \mathbf{1}^T \tau \end{bmatrix}. \end{aligned}$$

So we end up with the sufficient condition for $\mathcal{B} \subseteq \mathcal{E}$: There exists a diagonal matrix D such that

$$P \le D, \qquad \mathbf{Tr}\, D \le 1.$$

(The condition $D \ge 0$ is implied by the first inequality. Also, we can replace the inequality $\mathbf{Tr}\, D \le 1$ with $\mathbf{Tr}\, D = 1$.)

Now, here is the funny part. For our particular problem instance, the S-procedure *happens* to give exactly the right answer, with no conservatism. But that needn't have happened.

(c) Finding the minimizing γ subject to the LMIs found in part (b) is an SDP with variables P and γ^2. For this particular problem instance, it was found that $\gamma_{\min} = 2.56$. The following matlab/CVX code solves this problem

```
clear all

A1 = [0 1 0; 0 0 1; -1 -2 -1];
A2 = [0 1 0; 0 0 1; -2 -4 -7];
A3 = [0 1 0; 0 0 1; -2 -3 -3];

cvx_begin sdp
    variable P(3,3) symmetric
    variable gamma2
```

```
        minimize(gamma2)
        subject to
            P >= 0
            A1'*P + P*A1 <= 0
            A2'*P + P*A2 <= 0
            A3'*P + P*A3 <= 0
            [1 1 1]*P*[1;1;1] <= 1
            [-1 1 1]*P*[-1;1;1] <= 1
            [1 -1 1]*P*[1;-1;1] <= 1
            [1 1 -1]*P*[1;1;-1] <= 1
            [gamma2 [1 0 0]; [1 0 0]' P] >= 0
            [gamma2 [0 1 0]; [0 1 0]' P] >= 0
            [gamma2 [0 0 1]; [0 0 1]' P] >= 0
            gamma2 >= 0
    cvx_end

    gamma = sqrt(gamma2)
```

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 $D^{\rm pf}$ defined by

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

   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 $\lambda_{\rm pf}(A) \le \|DAD^{-1}\|$ for any nonsingular diagonal D; then show that equality holds here when $D = D^{\rm pf}$.

   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.

Solution: For any square matrix X, we have $\rho(X) \leq \|X\|$, where $\rho(X) = \max_i |\lambda_i(X)|$ is the *spectral radius* of X. This inequality is an equality in the case where X is symmetric. Therefore

$$\rho(DAD^{-1}) \leq \|DAD^{-1}\|.$$

Since A and DAD^{-1} share the same eigenvalues (because they are related by a similarity transformation),

$$\rho(DAD^{-1}) = \rho(A) = \lambda_{\rm pf}(A).$$

Combining these we conclude

$$\lambda_{\rm pf}(A) \leq \|DAD^{-1}\|.$$

Thus, you can't do any better than $\lambda_{\rm pf}(A)$ for $\|DAD^{-1}\|$.

Now we'll show that $D^{\rm pf}$ actually achieves this lower bound, and thus is optimal.

Let $D^{\rm pf}$ be such that $D^{\rm pf}_{ii} = (w_i/v_i)^{1/2}$ and let z be the vector defined as $z_i = (w_i v_i)^{1/2}$. Note that $(D^{\rm pf})^{-1}z = v$ and $D^{\rm pf}z = w$. Then

$$\begin{aligned}
(D^{\rm pf})^{-1}A^T(D^{\rm pf})^2A(D^{\rm pf})^{-1}z &= (D^{\rm pf})^{-1}A^T(D^{\rm pf})^2Av \\
&= \lambda_{\rm pf}(A)(D^{\rm pf})^{-1}A^T(D^{\rm pf})^2v \\
&= \lambda_{\rm pf}(A)(D^{\rm pf})^{-1}A^TD^{\rm pf}(D^{\rm pf}v) \\
&= \lambda_{\rm pf}(A)(D^{\rm pf})^{-1}A^TD^{\rm pf}z \\
&= \lambda_{\rm pf}(A)(D^{\rm pf})^{-1}A^Tw \\
&= \lambda_{\rm pf}(A)^2(D^{\rm pf})^{-1}w \\
&= \lambda_{\rm pf}(A)^2z.
\end{aligned}$$

Therefore z is an eigenvector of $(D^{\rm pf})^{-1}A^T(D^{\rm pf})^2A(D^{\rm pf})^{-1}$ associated with the eigenvalue $\lambda_{\rm pf}(A)^2$.

Since z is nonnegative and nonzero, then z is the right Perron-Frobenius eigenvector of $(D^{\rm pf})^{-1}A^T(D^{\rm pf})^2A(D^{\rm pf})^{-1}$ and $\lambda_{\rm pf}(A)^2$ is the Perron-Frobenius eigenvalue of $(D^{\rm pf})^{-1}A^T(D^{\rm pf})^2A(D^{\rm pf})^{-1}$ (this follows directly from problem 9 of homework 9.)

Therefore

$$\rho((D^{\rm pf})^{-1}A^T(D^{\rm pf})^2A(D^{\rm pf})^{-1}) = \lambda_{\rm pf}(A)^2.$$

Since

$$\|D^{\rm pf}A(D^{\rm pf})^{-1}\|^2 = \rho((D^{\rm pf})^{-1}A^T(D^{\rm pf})^2A(D^{\rm pf})^{-1})$$

we conclude

$$\|D^{\rm pf}A(D^{\rm pf})^{-1}\|^2 = \lambda_{\rm pf}(A)^2.$$

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.

Solution:

(a) From $x(t+1) = x(t) + \alpha r(t)$ and $r(t) = b - Ax(t)$ we have

$$x(t+1) = x(t) + \alpha(b - Ax(t)) = (I - \alpha A)x(t) + \alpha b.$$

Let $e(t) = x(t) - \bar{x}$ be the error at iteration t. Then

$$e(t+1) \;=\; x(t+1) - \bar{x}$$

$$\begin{aligned}
&= (I - \alpha A)x(t) - \alpha b - \bar{x} \\
&= (I - \alpha A)x(t) - \alpha A\bar{x} - \bar{x} \\
&= (I - \alpha A)(x(t) - \bar{x}) \\
&= (I - \alpha A)e(t) \\
&= Be(t),
\end{aligned}$$

where $B = I - \alpha A$. Thus, $e(t)$ satisfies a linear recusion; it converges to zero (no matter what the initial error is) if and only if all eigenvalues of B have magnitude smaller than one. But B is symmetric, so this is the same as saying that $\|B\| < 1$. Since the eigenvalues of B are equal to $1 - \alpha\lambda_i$, $i = 1, \ldots, n$, where the λ_i's are the eigenvalues of A, this condition is equivalent to

$$|1 - \alpha\lambda_i| < 1, \quad i = 1, \ldots, n.$$

In other words we need $-1 < 1 - \alpha\lambda_i < 1$ for $i = 1, \ldots, n$. The righthand inequality is satisfied for any $\alpha > 0$, since $A > 0$. The lefthand inequality requires $\alpha < 2/\lambda_{\max}(A)$. So the iteration converges if and only if

$$0 < \alpha < \frac{2}{\lambda_{\max}(A)}.$$

(b) Let ρ be the spectral radius of B, *i.e.*,

$$\rho = \max_{i=1,\ldots,n}|1 - \alpha\lambda_i|.$$

Since our goal is to have the optimal convergence rate (*i.e.*, the fastest convergence possible), we seek to minimize ρ over values of α in $(0, \frac{2}{\lambda_{\max}(A)})$. It is easy to see that the minimum ρ occurs when

$$1 - \alpha\lambda_{\max}(A) = -(1 - \alpha\lambda_{\min}(A)).$$

So the minimum value of ρ occurs when

$$\alpha = \frac{2}{\lambda_{\min}(A) + \lambda_{\max}(A)}.$$

Given this optimal value of α, we can now calculate the optimal ρ as

$$\begin{aligned}
\rho &= 1 - \frac{2\lambda_{\min}(A)}{\lambda_{\min}(A) + \lambda_{\max}(A)} \\
&= \frac{\lambda_{\max}(A) - \lambda_{\min}(A)}{\lambda_{\max}(A) + \lambda_{\min}(A)} \\
&= \frac{\kappa(A) - 1}{\kappa(A) + 1}.
\end{aligned}$$

This is the optimal asymptotic convergence rate and it is achieved by setting $\alpha = \frac{2}{\lambda_{\min}(A)+\lambda_{\max}(A)}$.

(c) Minimizing $\kappa(DAD)$ over diagonal D is equivalent to minimizing δ subject to $\kappa(DAD) \leq \delta$, *i.e.*,

$$\lambda I \leq D^{\text{opt}} A D^{\text{opt}} \leq \lambda \delta I$$

for some $\lambda \geq 0$. This inequality is homogeneous in D^{opt} and λ so it will still hold if scaled by any positive real number. This means we can, without loss of generality, assume that $\lambda = 1$. Then our problem is equivalent to

$$\begin{array}{ll} \text{minimize} & \delta \\ \text{subject to} & I \leq DAD \leq \delta I, \end{array} \tag{4}$$

with variables γ and a diagonal matrix D. We multiply the inequality on the left and right by D^{-1} (which preserves matrix inequalities) to get

$$\begin{array}{ll} \text{minimize} & \delta \\ \text{subject to} & D^{-2} \leq A \leq \delta D^{-2}. \end{array} \tag{5}$$

Now we use the variable $E = D^{-2}$, which is a diagonal, positive definite matrix, and $\gamma = 1/\delta$. This gives the SDP

$$\begin{array}{ll} \text{maximize} & \gamma \\ \text{subject to} & E \leq A \\ & E \geq \gamma A. \end{array} \tag{6}$$

(Note that the second inequality implies E is positive definite.) After solving this SDP, we find the optimal diagonal preconditioner by taking $D^{\text{opt}} = E^{-1/2}$.

(d) The following Matlab script uses `cvx` to solve the SDP derived in part (a)

```
n = 4;
A = [66    68    32    70; ...
     68   106    56    96; ...
     32    56    33    50; ...
     70    96    50    95];

cvx_begin sdp
    variable E(n,n) diagonal
    variable gamma
    maximize (gamma)
    gamma*A <= E
    E <= A
cvx_end

kappa = cond(A)

D_opt = E^(-.5)
```

```
kappa_opt = 1/gamma

D_diag = diag(diag(A).^(-.5))
kappa_diag = cond(D_diag*A*D_diag)
```

The optimal diagonal preconditioner is

```
D_opt =
    0.4728         0         0         0
         0    0.6251         0         0
         0         0    0.8750         0
         0         0         0    0.5696
```

and the condition numbers are

```
kappa =
  150.0660
kappa_diag =
  128.4465
kappa_opt =
  103.9398
```

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 $i(0),\dots,i(t-1)$, 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 $i(0),\dots,i(t-1)$ determines $\Sigma(t|t-1)$; then, $\Sigma(t|t)$ depends on $i(t)$, *i.e.*, which of $C_1,\dots,C_K$ 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 $y(0), y(1),\dots$, 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 $1, 2, 3, 1, 2, 3, \dots$

(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 $t = 0,\dots,50$.

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.

Solution.

(a) Let $\Sigma_i(t|t)$ be the estimation error covariance when $C(t) = C_i$, for all t. To plot the evolution of the MSE with time, we just have to iteratively apply the time and measurement update formulas from the lecture notes.

In order to find the asymptotic value of $\mathbf{Tr}\,\Sigma_i(t|t)$ (which we will denote $\mathbf{Tr}\,\Sigma_{i,\mathrm{ss}}(t|t)$), we first have to solve the DARE

$$\hat{\Sigma}_i = A\hat{\Sigma}_i A^T + W - A\hat{\Sigma}_i C_i^T (C_i \hat{\Sigma}_i C^T + V)^{-1} C_i \hat{\Sigma}_i A^T,$$

and then apply the measurement update formula

$$\Sigma_{i,\mathrm{ss}}(t|t) = \hat{\Sigma}_i - \hat{\Sigma}_i C_i^T (C_i \hat{\Sigma}_i C^T + V)^{-1} C_i \hat{\Sigma}_i.$$

The following matlab code was used for this part of the problem:

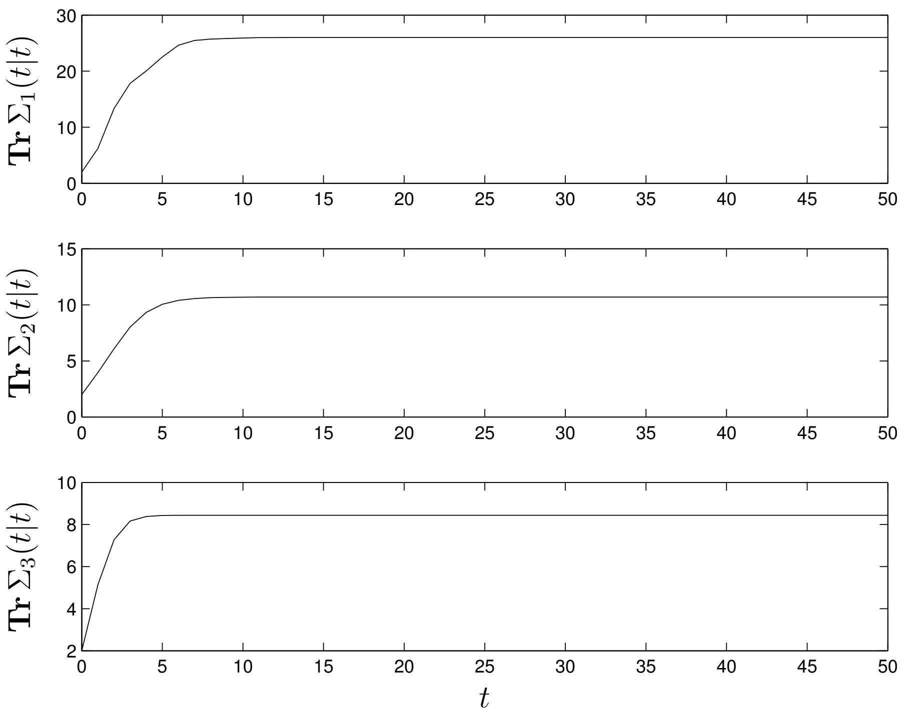
```
sens_data
N = 50;

% Fixed Sensor Policy
Sigmahat1 = Sigma0; Sigmahat2 = Sigma0; Sigmahat3 = Sigma0;
mses1=[]; mses2 = []; mses3 = [];
for n = 1:N+1
    % First sensor
    C = C1;
    % Measurement Update
    Sigma1 = Sigmahat1-Sigmahat1*C'*inv(C*Sigmahat1*C'+V)*...
        C*Sigmahat1;
    % Time Update
    Sigmahat1 = A*Sigma1*A'+W;
    mses1 = [mses1 trace(Sigma1)];

    % Second sensor
    C = C2;
    % Measurement Update
    Sigma2 = Sigmahat2-Sigmahat2*C'*inv(C*Sigmahat2*C'+V)*...
        C*Sigmahat2;
    % Time Update
    Sigmahat2 = A*Sigma2*A'+W;
    mses2 = [mses2 trace(Sigma2)];

    % Third sensor
    C = C3;
    % Measurement Update
    Sigma3 = Sigmahat3-Sigmahat3*C'*inv(C*Sigmahat3*C'+V)*...
        C*Sigmahat3;
    % Time Update
    Sigmahat3 = A*Sigma3*A'+W;
    mses3 = [mses3 trace(Sigma3)];
```

```
end

figure
subplot(3,1,1)
plot(0:N,mses1)
ylabel('mse1')
subplot(3,1,2)
plot(0:N,mses2)
ylabel('mse2')
subplot(3,1,3)
plot(0:N,mses3)
ylabel('mse3')
xlabel('time')

print -deps msefixed.eps

% Find steady-state values
% First sensor
C = C1;
Shat = dare(A',C',W,V);
mse1 = trace(Shat-Shat*C'*inv(C*Shat*C'+V)*C*Shat)

% Second sensor
C = C2;
Shat = dare(A',C',W,V);
mse2 = trace(Shat-Shat*C'*inv(C*Shat*C'+V)*C*Shat)

% Third sensor
C = C3;
Shat = dare(A',C',W,V);
mse3 = trace(Shat-Shat*C'*inv(C*Shat*C'+V)*C*Shat)
```

The steady-state values of the MSE for each i are

$$\mathbf{Tr}\,\Sigma_{1,\mathrm{ss}}(t|t) = 26.01, \qquad \mathbf{Tr}\,\Sigma_{2,\mathrm{ss}}(t|t) = 10.69, \qquad \mathbf{Tr}\,\Sigma_{3,\mathrm{ss}}(t|t) = 8.44.$$

The following plots show the evolution of $\mathbf{Tr}\,\Sigma_i(t|t)$ with time, for each i.

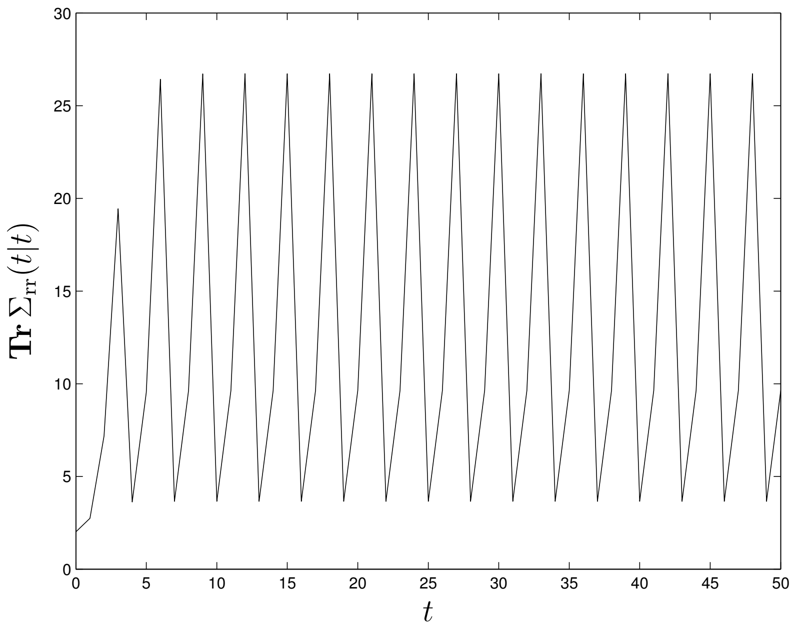


It is evident that the best fixed sensor choice is $C(t) = C_3$, for all t.

(b) Let $\Sigma_{\rm rr}(t|t)$ be the estimation error covariance when using a round-robbin sensor sequence. The following matlab code calculates and plots $\mathbf{Tr}\,\Sigma_{\rm rr}(t|t)$, for $t = 0, \ldots, 50$:

```
% Round robbin
Sigmahat = Sigma0;
mse_rr=[];
time = 1;
while(1)
    % Sensor 1
    C = C1;

    % Measurement Update
    Sigma = Sigmahat-Sigmahat*C'*inv(C*Sigmahat*C'+V)*...
        C*Sigmahat;

    % Time Update
    Sigmahat = A*Sigma*A'+W;

    mse_rr = [mse_rr trace(Sigma)];
    time = time+1;
    if(time>N+1), break; end
    % Sensor 2
    C = C2;

    % Measurement Update
    Sigma = Sigmahat-Sigmahat*C'*inv(C*Sigmahat*C'+V)*...
        C*Sigmahat;

    % Time Update
    Sigmahat = A*Sigma*A'+W;

    mse_rr = [mse_rr trace(Sigma)];
    time = time+1;
    if(time>N+1), break; end

    % Sensor 3
    C = C3;

    % Measurement Update
    Sigma = Sigmahat-Sigmahat*C'*inv(C*Sigmahat*C'+V)*...
        C*Sigmahat;

    % Time Update
    Sigmahat = A*Sigma*A'+W;

    mse_rr = [mse_rr trace(Sigma)];
    time = time+1;
    if(time>N+1), break; end
end

figure
plot(0:N,mse_rr);
ylabel('mserr')
xlabel('time')
print -deps mserr.eps
```

The following plot shows the evolution of $\mathbf{Tr}\,\Sigma_{\rm rr}(t|t)$ with time.

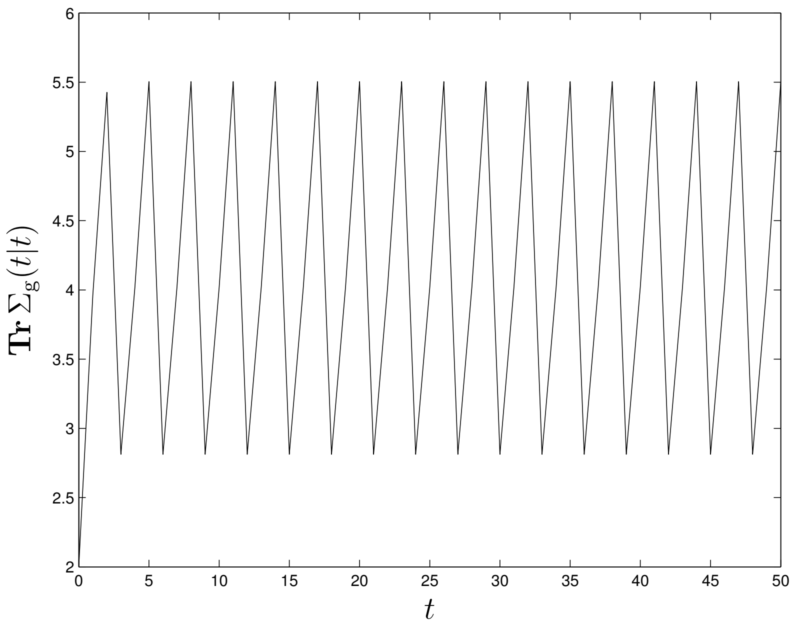


The round-robbin sensor sequence is performing much worse than selecting the best fixed sensor.

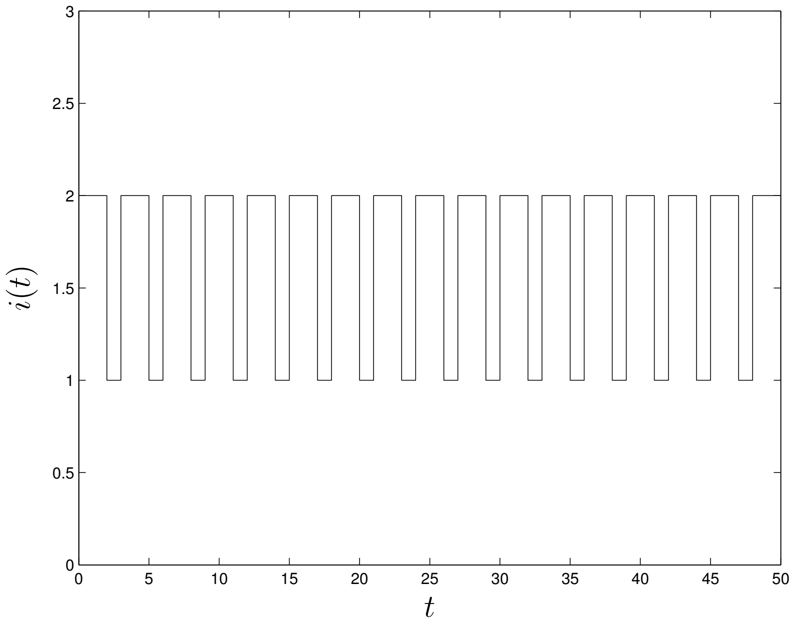
(c) Let $\Sigma_{\rm g}(t|t)$ be the estimation error covariance when using the proposed greedy sensor selection heuristic. The following matlab code calculates and plots $\mathbf{Tr}\,\Sigma_{\rm g}(t|t)$, for $t = 0, \ldots, 50$:

```
% Greedy algorithm
Sigmahat = Sigma0;
mse_g=[];
policy = [];
for n = 1:N+1
    % Measurement Updates
    % First sensor
    C = C1;
    Sigma1 = Sigmahat-Sigmahat*C'*inv(C*Sigmahat*C'+V)*...
        C*Sigmahat;

    % Second sensor
    C = C2;
    Sigma2 = Sigmahat-Sigmahat*C'*inv(C*Sigmahat*C'+V)*...
        C*Sigmahat;

    % Third sensor
    C = C3;
    Sigma3 = Sigmahat-Sigmahat*C'*inv(C*Sigmahat*C'+V)*...
        C*Sigmahat;
    % Greedy sensor selection
    mses = [trace(Sigma1) trace(Sigma2) trace(Sigma3)];
    [min_mse,ind] = min(mses);
    ind = ind(1);
    policy = [policy ind];
    mse_g = [mse_g min_mse];
    switch ind
        case 1
            Sigma = Sigma1;
        case 2
            Sigma = Sigma2;
        case 3
            Sigma = Sigma3;
    end

    % Time update
    Sigmahat = A*Sigma*A'+W;
end

figure
plot(0:N,mse_g);
ylabel('mseg')
xlabel('time')
print -deps mseg.eps

figure
stairs(0:N,policy)
ylabel('policy')
xlabel('time')
axis([0 N 0 3])
print -deps polg.eps
```

The following plot shows the evolution of $\mathbf{Tr}\,\Sigma_{\rm g}(t|t)$ with time.



The following plot shows the sensor sequence $i(t)$ used by the greedy heuristic.



For this particular system, the greedy heuristic is better than using a single fixed sensor with respect to the MSE, since

$$\mathbf{Tr}\,\Sigma_{\mathrm{g}}(t|t) < \mathbf{Tr}\,\Sigma_3(t|t),$$

for all t. It is interesting to note that the sensor sequence used by the heuristic does not contain C_3, which is the optimal fixed sensor.

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.)

Solution: Let's start with some preliminary analysis. The trajectory $x(t)$ can be split into two parts:

$$x(t) = A^t x(0) + \tilde{x}(t),$$

where $\tilde{x}(t)$ is the state trajectory with zero initial condition. Therefore

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

The first part depends only on the initial state, and not on the input u; the second part depends only on the input, and not the initial condition. Note that the first part is always finite, and can be expressed as a linear function of the initial state:

$$\sum_{t=0}^{\infty} \gamma^t q^T A^t x(0) = q^T \sum_{t=0}^{\infty} (\gamma A)^t x(0) = q^T (I - \gamma A)^{-1} x(0).$$

Already we can make an interesting observation: Any input sequence optimal for one value of initial condition $x(0)$ must be optimal for any other initial condition. It follows that the value function V is *affine* in z:

$$V(z) = p^T z + s,$$

where

$$p = (I - \gamma A^T)^{-1} q.$$

This makes some sense: in ordinary LQR, with quadratic state cost, the value function is quadratic in the state; here, with linear state cost, the value function is affine. The number s is the minimum value of J with zero initial state; we still have to find what this is.

Now we apply dynamic programming. The Bellman equation is

$$V(x) = \min_w \left(q^T x + w^T R w + \gamma V(Ax + Bw) \right),$$

which is, using $V(x) = p^T z + s$,

$$p^T x + s = \min_w \left(q^T x + w^T R w + \gamma p^T (Ax + Bw) + \gamma s \right).$$

By taking the derivative with respect to w and setting it to zero we have

$$2Rw^\star + \gamma B^T p = 0,$$

so $w^\star = -(\gamma/2)R^{-1}B^T p$. Thus, we now know the optimal input sequence: it is *constant*, with

$$u^\star(t) = -(\gamma/2)R^{-1}B^T p = -(\gamma/2)R^{-1}B^T (I - \gamma A^T)^{-1} q,$$

for all t. This is very different from the standard LQR problem, in which the optimal input sequence is a linear function of the state.

Let's check things, and evaluate the constant s, by substituting $w^\star = -(\gamma/2)R^{-1}B^T p$ into the Bellman equation:

$$p^T x + s = q^T x - \frac{\gamma^2}{4} p^T B R^{-1} B^T p + \gamma p^T A x + \gamma s.$$

The terms involving x dutifully drop away, and we are left with

$$(1 - \gamma)s = -\frac{\gamma^2}{4} p^T B R^{-1} B^T p.$$

Thus we have

$$\begin{aligned} s &= -\frac{\gamma^2}{4(1-\gamma)} p^T B R^{-1} B^T p \\ &= -\frac{\gamma^2}{4(1-\gamma)} q^T (I - \gamma A)^{-1} B R^{-1} B^T (I - \gamma A^T)^{-1} q. \end{aligned}$$

Now let's discuss some partial and some incorrect solutions. Some people simply assumed (without justification) that V was quadratic, with a linear and constant term as well. That's true, of course, since in fact V is affine, but this is something that should be justified. Using this form for V, you arrive at an ARE with zero constant term. One solution of the ARE is to take P (the quadratic part) to be zero. But few people noted that.

Another method was to augment the system with an extra state that is constant, *i.e.*, satisfies $\dot{z} = 0$. Then you can construct the linear state cost as a quadratic form in the augmented state. But in this form, the state cost is *not* positive semidefinite. But LQR with indefinite cost function is very tricky; in particular the existence and uniqueness of solutions is by no means clear. So you cannot just say “Solve the ARE”; it makes no sense. In general, there can be no solutions, or many; and there’s no reason to believe that whatever solution you might find is the one you want in the Bellman equation. In this case, of course, there is a solution, which in fact has zero quadratic part, and reduces to the simple solution found above. But this would have to be justified.