EE263 homework 9 solutions

14.16 *Frobenius norm of a matrix.* The Frobenius norm of a matrix $A \in \mathbf{R}^{n\times n}$ is defined as $\|A\|_\mathrm{F} = \sqrt{\mathbf{Tr}\, A^TA}$. (Recall $\mathbf{Tr}$ is the trace of a matrix, *i.e.*, the sum of the diagonal entries.)

(a) Show that

$$\|A\|_\mathrm{F} = \left(\sum_{i,j} |A_{ij}|^2 \right)^{1/2}.$$

Thus the Frobenius norm is simply the Euclidean norm of the matrix when it is considered as an element of $\mathbf{R}^{n^2}$. Note also that it is much easier to compute the Frobenius norm of a matrix than the (spectral) norm (*i.e.*, maximum singular value).

(b) Show that if U and V are orthogonal, then $\|UA\|_\mathrm{F} = \|AV\|_\mathrm{F} = \|A\|_\mathrm{F}$. Thus the Frobenius norm is not changed by a pre- or post- orthogonal transformation.

(c) Show that $\|A\|_\mathrm{F} = \sqrt{\sigma_1^2 + \cdots + \sigma_r^2}$, where $\sigma_1, \ldots, \sigma_r$ are the singular values of A. Then show that $\sigma_{\max}(A) \le \|A\|_\mathrm{F} \le \sqrt{r}\sigma_{\max}(A)$. In particular, $\|Ax\| \le \|A\|_\mathrm{F}\|x\|$ for all x.

Solution:

(a) Simply by definition

$$\|A\|_\mathrm{F}^2 = \mathbf{Tr}\, A^TA = \sum_i [A^TA]_{ii} = \sum_i \left(\sum_j A^T_{ij} A_{ji} \right) = \sum_{i,j} A_{ij}^2.$$

(b) First note that $\|UA\|_\mathrm{F} = \|A\|_\mathrm{F}$ because

$$\|UA\|_\mathrm{F}^2 = \mathbf{Tr}(UA)^T(UA) = \mathbf{Tr}\, A^TU^TUA = \mathbf{Tr}\, A^TA = \|A\|_\mathrm{F}^2.$$

and $\|AV\|_\mathrm{F} = \|A\|_\mathrm{F}$ since

$$\|AV\|_\mathrm{F}^2 = \mathbf{Tr}(AV)^T(AV) = \mathbf{Tr}(AV)(AV)^T = \mathbf{Tr}\, AVV^TA^T = \mathbf{Tr}\, AA^T = \mathbf{Tr}\, A^TA = \|A\|_\mathrm{F}^2$$

where we have used the fact that $\mathbf{Tr}\, XY = \mathbf{Tr}\, YX$.

(c) We start with the full SVD of $A = U\Sigma V^T$. By the previous problem,

$$\|A\|_\mathrm{F} = \|U^T\Sigma V\|_\mathrm{F} = \|\Sigma V\|_\mathrm{F} = \|\Sigma\|_\mathrm{F} = \sqrt{\sigma_1^2 + \cdots + \sigma_r^2}.$$

Since $\sigma_2^2, \ldots, \sigma_r^2 \ge 0$, we have $\sigma_1 \le \sqrt{\sigma_1^2 + \cdots + \sigma_r^2} = \|A\|_\mathrm{F}$. Next, since $\sigma_2, \ldots, \sigma_r \le \sigma_1$, we have $\|A\|_\mathrm{F} = \sqrt{\sigma_1^2 + \cdots + \sigma_r^2} \le \sqrt{\sigma_1^2 + \cdots + \sigma_1^2} = \sqrt{r}\sigma_1$.

14.26 *Optimal time compression equalizer.* We are given the (finite) impulse response of a communications channel, *i.e.*, the real numbers

$$c_1, c_2, \dots, c_n.$$

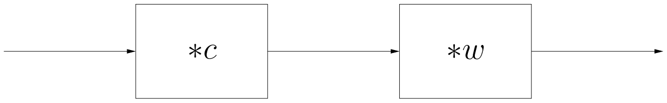
Our goal is to design the (finite) impulse response of an equalizer, *i.e.*, the real numbers

$$w_1, w_2, \dots, w_n.$$

(To make things simple, the equalizer has the same length as the channel.) The equalized channel response h is given by the convolution of w and c, *i.e.*,

$$h_i = \sum_{j=1}^{i-1} w_j c_{i-j}, \quad i = 2, \dots, 2n.$$

This is shown below.



The goal is to choose w so that most of the energy of the equalized impulse response h is *concentrated* within k samples of $t = n+1$, where $k < n-1$ is given. To define this formally, we first define the total energy of the equalized response as

$$E_{\text{tot}} = \sum_{i=2}^{2n} h_i^2,$$

and the energy in the desired time interval as

$$E_{\text{des}} = \sum_{i=n+1-k}^{n+1+k} h_i^2.$$

For any w for which $E_{\text{tot}} > 0$, we define the *desired to total energy ratio*, or DTE, as $\text{DTE} = E_{\text{des}}/E_{\text{tot}}$. Thus number is clearly between 0 and 1; it tells us what fraction of the energy in h is contained in the time interval $t = n+1-k, \dots, t = n+1+k$. You can assume that h is such that for any $w \neq 0$, we have $E_{\text{tot}} > 0$.

(a) How do you find a $w \neq 0$ that maximizes DTE? You must give a very clear description of your method, and explain why it works. Your description and justification must be *very clear.* You can appeal to any concepts used in the class, *e.g.*, least-squares, least-norm, eigenvalues and eigenvectors, singular values and singular vectors, matrix exponential, and so on.

(b) Carry out your method for time compression length $k = 1$ on the data found in `time_comp_data.m`. Plot your solution w, the equalized response h, and give the DTE for your w.

Please note: You do not need to know anything about equalizers, communications channels, or even convolution; everything you need to solve this problem is clearly defined in the problem statement. **Solution:**

(a) First we are going to find expressions for E_{tot} and E_{des}. We can write $h = Aw$, where A is the Toeplitz matrix

$$A = \begin{bmatrix} c_1 & & & \\ c_2 & c_1 & & \\ & & \ddots & \\ c_n & c_{n-1} & \cdots & c_1 \\ & c_n & \cdots & c_2 \\ & & \ddots & \\ & & & c_n \end{bmatrix}.$$

Let's define the subvector of the equalized response corresonding to the desired time interval as

$$\bar{h} = \begin{bmatrix} h_{n+1-k} \\ \vdots \\ h_{n+1+k} \end{bmatrix}.$$

Then we have $\bar{h} = Bw$, where B is the Toeplitz matrix consisting of the $n - k, \dots, n+k$ rows of A. Now we have

$$E_{\text{tot}} = \|h\|^2 = h^T h = w^T A^T A w, \qquad E_{\text{des}} = \|\bar{h}\|^2 = \bar{h}^T \bar{h} = w^T B^T B w.$$

The assumption that $E_{\text{tot}} > 0$ for any $w \neq 0$ means that $A^T A > 0$. Now we can state the problem in simple matrix form. We need to find w that maximizes the ratio of quadratic forms

$$d = \frac{w^T B^T B w}{w^T A^T A w}.$$

That's close to, but not exactly, a problem we have studied. We know how to solve this problem in the case $A^T A = I$. So let's reduce our problem to this one. Let's define $z = (A^T A)^{1/2} w$, so we have $w = (A^T A)^{-1/2} z$. We can express the DTE d in terms of z as

$$d = \frac{w^T B^T B w}{w^T A^T A w} = \frac{z^T (A^T A)^{-1/2} B^T B (A^T A)^{-1/2} z}{z^T z}.$$

Now we do know how to maximize this ratio. Its maximum value is

$$d_{\max} = \lambda_{\max}\left((A^T A)^{-1/2} B^T B (A^T A)^{-1/2} \right),$$

and the value of z that maximizes the ratio is v, the eigenvector associated with the maximum eigenvalue above. To find the value of w that maximizes DTE, we multiply v by $(A^T A)^{-1/2}$. Here's the summary:

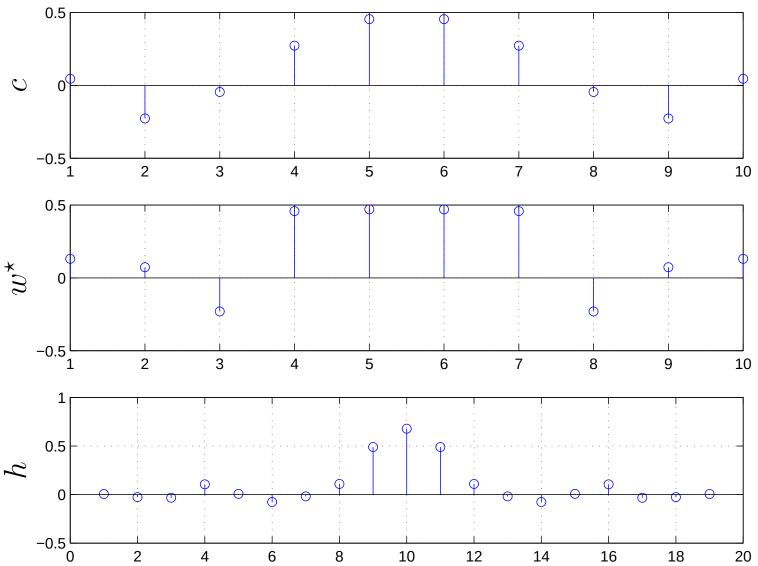


Figure 1: Time compression equilizer.

- Form the symmetric matrix $C = (A^TA)^{-1/2}B^TB(A^TA)^{-1/2}$.
- Let v be the eigenvector of C associated with its largest eigenvalue $\lambda_{\max}$.
- Let $w^\star = (A^TA)^{-1/2}z$.

The same algorithm can also be expressed in terms of the SVD of A and B. Many students came up with heuristics for (approximately) solving this problem, ranging from iterative least-squares, regularization, etc. Some of these methods even came up with an answer close to the correct answer. But we still took off a number of points, since the goal is to describe a correct method, not just to get the particular numerical answer (in this case). One heuristic is to simply take w as the right singular vector of B associated with its largest signular value. This does well (at least for this example) but isn't correct.

(b) When we carry out the procedure described above on the given problem instance with time-compression parameter $k = 1$, we obtain the optimal DTE $d_{\max} = 0.9375$. Thus, 93.75% of the energy in the equalized impulse response h is concentrated within the window of 3 samples around the impulse center. This can also be seen in figure 1, where we plotted the channel impulse response c, the equalizer $w^\star$, and the equalized impulse response h (also see the Matlab code below).

```
time_comp_data;  % defines channel impulse response
n = length(c);
k = 1; % time compression window length
A = toeplitz([c;zeros(n-1,1)], [c(1);zeros(n-1,1)]);
```

```
B = A(n-k:n+k,:);
D = inv(sqrtm(A'*A))*B'*B*inv(sqrtm(A'*A));
[vmax dmax] = eigs(D,1);
wmax = inv(sqrtm(A'*A))*vmax;
figure; subplot(311), stem(c,'o'); ylabel('c'); xlabel('n'); grid;
subplot(312), stem(wmax,'o'); ylabel('w'); xlabel('n'); grid;
subplot(313), stem(A*wmax,'o'); ylabel('h'); xlabel('n'); grid;
print -depsc time_comp_eq.eps
```

15.2 *Condition number.* Show that $\kappa(A) = 1$ if and only if A is a multiple of an orthogonal matrix. Thus the best conditioned matrices are precisely (scaled) orthogonal matrices. *Solution:*
Let us assume $\kappa(A) = 1$; we will show that A is a multiple of an orthogonal matrix. If $\kappa(A) = 1$, then $\sigma_{\min} = \sigma_{\max}$; so $\Sigma = \sigma_{\max} I$, and

$$A = U\Sigma V^T = \sigma_{\max}(UV^T), \quad AA^T = A^T A = \sigma_{\max}^2 I.$$

Thus, A is a scaled orthogonal matrix. Now let us assume that $A = \alpha U$, where U is an orthogonal matrix and $\alpha \in \mathbf{R}$; we will show that $\kappa(A) = 1$. Since

$$A = \alpha U = U(|\alpha| I)\mathrm{sgn}(\alpha) I,$$

then the above equation gives an SVD of A, in which $\Sigma = |\alpha| I$ and $V = \mathrm{sgn}(\alpha) I$ is an orthogonal matrix. So $\sigma_{\min} = \sigma_{\max} = |\alpha|$, and thus $\kappa(A) = 1$.

15.3 *Tightness of the condition number sensitivity bound.* Suppose A is invertible, $Ax = y$, and $A(x+\delta x) = y+\delta y$. In the lecture notes we showed that $\|\delta x\|/\|x\| \leq \kappa(A)\|\delta y\|/\|y\|$. Show that this bound is not conservative, *i.e.*, there are x, y, δx, and δy such that equality holds. *Conclusion:* the bound on relative error can be taken on, if the data x is in a particularly unlucky direction and the data error δx is in (another) unlucky direction. *Solution:*
Assume $A = U\Sigma V^T$, then $A^{-1} = V\Sigma^{-1}U^T$. Now, let $y = u_1$ and $\delta y = u_n$; then

$$\begin{array}{lcl} x = A^{-1}y = v_1/\sigma_{\max} & \Longrightarrow & \|x\| = \|v_1\|/\sigma_{\max} = 1/\sigma_{\max} \\ \delta x = A^{-1}\delta y = v_n/\sigma_{\min} & \Longrightarrow & \|\delta x\| = \|v_n\|/\sigma_{\min} = 1/\sigma_{\min}. \end{array}$$

Since by our construction $\|\delta y\|/\|y\| = 1$, we have

$$\frac{\|\delta x\|}{\|x\|} = \frac{\sigma_{\max}}{\sigma_{\min}} = \kappa(A)\frac{\|\delta y\|}{\|y\|}.$$

15.6 *Detecting linear relations.* Suppose we have N measurements $y_1, \ldots, y_N$ of a vector signal $x_1, \ldots, x_N \in \mathbf{R}^n$:

$$y_i = x_i + d_i, \ i = 1, \ldots, N.$$

Here d_i is some small measurement or sensor noise. We hypothesize that there is a linear relation among the components of the vector signal x, *i.e.*, there is a nonzero vector q such that $q^T x_i = 0$, $i = 1, \ldots, N$. The geometric interpretation is that all of the vectors x_i lie in the hyperplane $q^T x = 0$. We will assume that $\|q\| = 1$, which does not affect the linear relation. Even if the x_i's do lie in a hyperplane $q^T x = 0$, our measurements y_i will not; we will have $q^T y_i = q^T d_i$. These numbers are small, assuming the measurement noise is small. So the problem of determing whether or not there is a linear relation among the components of the vectors x_i comes down to finding out whether or not there is a unit-norm vector q such that $q^T y_i$, $i = 1, \ldots, N$, are all small. We can view this problem geometrically as well. Assuming that the x_i's all lie in the hyperplane $q^T x = 0$, and the d_i's are small, the y_i's will all lie close to the hyperplane. Thus a scatter plot of the y_i's will reveal a sort of flat cloud, concentrated near the hyperplane $q^T x = 0$. Indeed, for any z and $\|q\| = 1$, $|q^T z|$ is the distance from the vector z to the hyperplane $q^T x = 0$. So we seek a vector q, $\|q\| = 1$, such that all the measurements $y_1, \ldots, y_N$ lie close to the hyperplane $q^T x = 0$ (that is, $q^T y_i$ are all small). How can we determine if there is such a vector, and what is its value? We define the following normalized measure:

$$\rho = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(q^T y_i)^2} \Big/ \sqrt{\frac{1}{N}\sum_{i=1}^{N}\|y_i\|^2}.$$

This measure is simply the ratio between the *root mean square distance* of the vectors to the hyperplane $q^T x = 0$ and the *root mean square length* of the vectors. If ρ is small, it means that the measurements lie close to the hyperplane $q^T x = 0$. Obviously, ρ depends on q. Here is the problem: explain how to find the minimum value of ρ over all unit-norm vectors q, and the unit-norm vector q that achieves this minimum, given the data set $y_1, \ldots, y_N$. *Solution:*

Let us collect the measurements into one large $n \times N$ matrix, $Y := [y_1 \ \cdots \ y_N]$. Then the formula for the confidence measurement ρ becomes

$$\rho = \frac{\sqrt{q^T Y Y^T q}}{\sqrt{\mathbf{Tr}\, Y^T Y}} = \frac{\|Y^T q\|}{\sqrt{\mathbf{Tr}\, Y^T Y}}.$$

Now there are two cases to consider: $N < n$, and $N \geq n$. If $N < n$, then it is *always* possible to determine a q such that $Y^T q = 0$; indeed, for $N < n - 1$, there are an infinite number of them. Of course, this is not a very interesting case; it suggests that too few measurements have been taken. On the other hand, if $N \geq n$, then we can express the minimum cost in terms of σ_n. Let $Y = U \Sigma V^T$ be the full SVD of Y, where $U \in \mathbf{R}^{n \times n}$, $\Sigma \in \mathbf{R}^{n \times N}$, and $V \in \mathbf{R}^{N \times N}$. Then

$$\rho = \frac{\|V \Sigma^T U^T q\|}{\sqrt{\mathbf{Tr}\, V \Sigma^2 V^T}} = \frac{\|\Sigma^T U^T q\|}{\sqrt{\sum_{i=1}^{n} \sigma_i^2(Y)}} \geq \frac{\sigma_n(Y)}{\sqrt{\sum_{i=1}^{n} \sigma_i^2(Y)}}.$$

Equality is achieved if $q = u_n$, because $\Sigma^T U^T q = \Sigma^T e_n = \sigma_n e_n$. Therefore, the n-th singular vector is the minimizing q. Of course, if Y is rank-deficient, then $\sigma_n = \rho = 0$, but in practice this would be unlikely.

15.8 Consider the system $\dot{x} = Ax$ with

$$A = \begin{bmatrix} 0.3132 & 0.3566 & 0.2545 & 0.2579 & 0.2063 \\ -0.0897 & 0.2913 & 0.1888 & 0.4392 & 0.1470 \\ 0.0845 & 0.2433 & -0.5888 & -0.0407 & 0.1744 \\ 0.2478 & -0.1875 & 0.2233 & 0.3126 & -0.6711 \\ 0.1744 & 0.2315 & -0.1004 & -0.2111 & 0.0428 \end{bmatrix}.$$

(a) Find the initial state $x(0) \in \mathbf{R}^5$ satisfying $\|x(0)\| = 1$ such that $\|x(3)\|$ is maximum. In other words, find an initial condition of unit norm that produces the *largest* state at $t = 3$.

(b) Find the initial state $x(0) \in \mathbf{R}^5$ satisfying $\|x(0)\| = 1$ such that $\|x(3)\|$ is minimum.

To save you the trouble of typing in the matrix A, you can find it on the web page in the file `max_min_init_state.m`. *Solution:*
Recall that $x(t) = e^{At}x(0)$ where $e^{tA} \in \mathbf{R}^{n\times n}$. Suppose $e^{tA} = U\Sigma V^T$ is a full SVD of e^{tA}. Now, the maximum and minimum values for $x(t)$ for $\|x(0)\| = 1$ are obtained when $x(0)$ is equal to v_1 and v_n respectively. Using the following Matlab function we are able to compute the maximum and minimum values for $x(t)$ given t and A subject to $\|x(0)\| = 1$:

```
function [xmin,xmax]=solution(A,t) [U,S,V]=svd(expm(A*t));
[vrows,vcols]=size(V); xmax=V(:,1); xmin=V(:,vcols);
```

(a) For $t = 3$ the value of $x(0)$ with $\|x(0)\| = 1$ that maximizes $\|x(t)\|$ is found to be

$$x_{\max} = \begin{bmatrix} 0.4413 \\ 0.2706 \\ 0.2837 \\ 0.7356 \\ -0.3324 \end{bmatrix}.$$

The maximum value for $\|x(3)\|$ is simply $\sigma_{\max}(e^{3A}) = 8.2706$.

(b) The value of $x(0)$ with $\|x(0)\| = 1$ that minimizes $\|x(t)\|$ becomes

$$x_{\min} = \begin{bmatrix} 0.1939 \\ 0.2062 \\ -0.9333 \\ 0.0735 \\ -0.2085 \end{bmatrix}.$$

The minimum value for $\|x(3)\|$ is simply $\sigma_{\min}(e^{3A}) = 0.1468$.

15.10 *Optimal binary signalling.* We consider a communication system given by

$$y(t) = Au(t) + v(t), \quad t = 0, 1, \ldots.$$

Here

- $u(t) \in \mathbf{R}^n$ is the transmitted (vector) signal at time t
- $y(t) \in \mathbf{R}^m$ is the received (vector) signal at time t
- $v(t) \in \mathbf{R}^m$ is noise at time t
- $t = 0,\ 1,\ \ldots$ is the (discrete) time

Note that the system has no memory: $y(t)$ depends only on $u(t)$. For the noise, we assume that $\|v(t)\| < V_{\max}$. Other than this maximum value for the norm, we know nothing about the noise (for example, we do not assume it is random). We consider binary signalling, which means that at each time t, the transmitter sends one of two signals, *i.e.*, we have either $u(t) = s_1 \in \mathbf{R}^n$ or $u(t) = s_2 \in \mathbf{R}^n$. The receiver then guesses which of the two signals was sent, based on $y(t)$. The process of guessing which signal was sent, based on the received signal $y(t)$, is called *decoding.* In this problem we are only interested in the case when the communication is completely reliable, which means that the receiver's estimate of which signal was sent is always correct, no matter what $v(t)$ is (provided $\|v(t)\| < V_{\max}$, of course). Intuition suggests that this is possible when $V_{\max}$ is small enough.

(a) Your job is to design the signal constellation, *i.e.*, the vectors $s_1 \in \mathbf{R}^n$ and $s_2 \in \mathbf{R}^n$, and the associated (reliable) decoding algorithm used by the receiver. Your signal constellation should minimize the maximum transmitter power, *i.e.*,

$$P_{\max} = \max\{\|s_1\|, \|s_2\|\}.$$

You must describe:

- your analysis of this problem,
- how you come up with s_1 and s_2,
- the exact decoding algorithm used,
- how you know that the decoding algorithm is reliable, *i.e.*, the receiver's guess of which signal was sent is always correct.

(b) The `file opt_bin_data.m` contains the matrix A and the scalar $V_{\max}$. Using your findings from part 1, determine the optimal signal constellation.

Solution.

Part 1. The first thing to do is to figure out when the receiver is able to correctly decode, no matter what the noise is. The received signal is either $As_1 + v$, or $As_2 + v$. As v ranges over all possible values, *i.e.*, $\|v\| < V_{\max}$, $As_i + v$ traces out the interior of a ball in $\mathbf{R}^n$, centered at the point As_i, with radius $V_{\max}$. If the two open balls intersect then we cannot do perfect decoding, no matter how fancy a method we use, because there is some received signal that could have come from either original signal. So the key to being able to perfectly decode is that the two (open) balls must not intersect. It's easy to say when two (open) balls don't intersect: it's only if the distance between the centers is greater than or equal to the sum of the radii. Here that means that we must have

$$\|As_1 - As_2\| \geq 2V_{\max}.$$

If this condition holds, we can do perfect, error free decoding; if it does not, then we cannot do perfect decoding. Before moving on to the design of s_1 and s_2, let's discuss how decoding can be done when the condition above holds. The simplest method is to choose s_1 when the received signal y satisfies $\|y - As_1\| < V_{\max}$, and to choose s_2 when satisfies $\|y - As_2\| < V_{\max}$. According to our analysis above, exactly one of these two conditions holds. There are several other methods, all variations on the one above, that work. We can, for example, choose s_1 when $\|y - As_1\| < \|y - As_2\|$, and s_2 otherwise. As_1 is what we'd receive if there were no noise, and s_1 is transmitted; As_2 is what we'd receive if s_2 were transmitted. The decoding scheme described above is very natural: we choose the transmitted signal for which the associated received signal is closest to what we actually received. This decoder also turns out to be the same as a so-called linear decoder. Squaring the condition, we get $\|y - As_1\|^2 < \|y - As_2\|^2$, which expands to

$$\|y\|^2 - 2(As_1)^T y + \|As_1\|^2 < \|y\|^2 - 2(As_2)^T y + \|As_2\|^2$$

as the condition for decoding s_1. This is the same as

$$(s_1 - s_2)A^T y > \|As_2\|^2 - \|As_1\|^2.$$

In other words, we form the inner product of y with a certain vector, and if it is above a threshold, we decode s_1. Enough for decoding. Our problem is now to find s_1 and s_2 that satisfy the decoding condition $\|As_1 - As_2\| \geq 2V_{\max}$, and minimizes the maximum transmitter power $P_{\max} = \max\{\|s_1\|, \|s_2\|\}$. If we fix the maximum transmitter power, then s_1 and s_2 lie in a ball. The (noise free) received signals As_1 and As_2 lie in an ellipsoid. We'd like the points in the ellipsoid that are farthest apart. But that's easy: they are the opposite sides of the ellipsoid along the major semi-axis. In other words, $As_1 = -As_2 = \alpha u_1$, where u_1 is the left singular vector of A associated with the largest singular value. This means that

$$s_1 = \alpha v_1, \qquad s_2 = -\alpha v_1,$$

where v_1 is the left singular vector of A (associated with the largest singular value). The decoding condition is:

$$\begin{array}{rcl} 2V_{\max} & \leq & \|As_1 - As_2\| \\ & = & \|A\alpha v_1 - A(-\alpha v_1)\| \\ & = & |\alpha|\sigma_1\|u_1 - (-u_1)\| \\ & = & 2|\alpha|\sigma_1. \end{array}$$

So we can choose for α the smallest value that satisfies the decoding condition,

$$\alpha = \frac{V_{\max}}{\sigma_1},$$

so our signal constellation is

$$s_1 = \frac{V_{\max}}{\sigma_1} v_1, \qquad s_2 = -\frac{V_{\max}}{\sigma_1} v_1.$$

This results in a transmitter power of

$$P_{\max} = \alpha = \frac{V_{\max}}{\sigma_1}.$$

Part 2. For our matrix, the maximum singular value is unique so there is only one solution (although it is arbitrary which signal is chosen to be s_1 and which is s_2). The following matlab code can be used to find the optimal signal constellation.

```
[U S V]=svd(A);               % SVD decomposition of A
v1=V(:,1);                    % Select vector associated with
alpha=Vmax/(S(1,1));          % Find scaling to assure detection
s1=alpha*v1;
s2=-alpha*v1;
```

The optimal signal constellation is

$$\begin{bmatrix} -0.06 \\ -0.04 \\ -0.03 \\ -0.07 \\ -0.05 \end{bmatrix}, \qquad \begin{bmatrix} 0.06 \\ 0.04 \\ 0.03 \\ 0.07 \\ 0.05 \end{bmatrix}.$$

15.11 *Some input optimization problems.* In this problem we consider the system $x(t+1) = Ax(t) + Bu(t)$, with

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix}, \qquad x(0) = \begin{bmatrix} 1 \\ 0 \\ -1 \\ 1 \end{bmatrix}.$$

(a) *Least-norm input to steer state to zero in minimum time.* Find the minimum T, $T_{\min}$, such that $x(T) = 0$ is possible. Among all $(u(0), u(1), \ldots u(T_{\min} - 1))$ that steer $x(0)$ to $x(T_{\min}) = 0$, find the one of minimum norm, *i.e.*, the one that minimizes

$$J_{T_{\min}} = \|u(0)\|^2 + \cdots + \|u(T_{\min} - 1)\|^2.$$

Give the minimum value of $J_{T_{\min}}$ achieved.

(b) *Least-norm input to achieve* $\|x(10)\| \leq 0.1$. In lecture we worked out the least-norm input that drives the state exactly to zero at $t = 10$. Suppose instead we only require the state to be *small* at $t = 10$, for example, $\|x(10)\| \leq 0.1$. Find $u(0), u(1), \ldots, u(9)$ that minimize

$$J_9 = \|u(0)\|^2 + \cdots + \|u(9)\|^2$$

subject to the condition $\|x(10)\| \leq 0.1$. Give the value of J_9 achieved by your input.

Solution:

(a) The first task is to find the smallest T such that $x(T) = 0$ is possible. This is the smallest T such that $-A^T x(0) \in \text{range}(\mathcal{C}_T)$, where $\mathcal{C}_T$ is the reachability matrix defined in the lectures. (Ooops we have a terrible symbol clash here — by A^T we mean A to the Tth power, not transpose.) We start with $T = 1$ by checking if $-Ax(0)$ is in range(B). If so, then we can have $x(1) = 0$; if not we try $T = 2$ by checking if $-A^2x(0)$ is in range$([B\ AB])$. If so, then we can have $x(2) = 0$; if not we try $T = 3$. If we fail at $T = 4$, then we know there is no input (of any length) that drives the state to zero. There are many ways to check whether a vector g is in the range of a matrix F, *e.g.*, QR factorization of $[F\ g]$. We'll use the simple method based on checking the ranks of F and $[F\ g]$.

```
>>rank([B A*x0])-rank(B)
ans = 1
>>rank([B A*B A^2*x0])-rank([B A*B])
ans = 0
```

We see that $T_{\min} = 2$ is the smallest T that works; we can have $x(2) = 0$, but $x(1) = 0$ is impossible. Now let's look at finding the minimum norm $U = [u(1)^T\ u(0)^T]^T$ that achieves $x(2) = 0$. Such U's satisfy the linear equation

$$[B\ AB]U = -A^2x(0).$$

The rank of $[B\ AB]$ is three, so it is singular. This means several things: first of all, we are lucky that $-A^2x(0)$ is in the range (which is only a three dimensional subspace of $\mathbf{R}^4$); second, there is a whole one-dimensional *line* of U's that satisfy the equation. So here we have a need for the general pseudo-inverse; we cannot use the formula for the least-norm solution that works for full-rank matrices (go ahead and try it to see what happens!). The solution is simply $U = -[B\ AB]^\dagger A^2x(0)$. You can compute this manually using the SVD, or using the command `pinv()` in Matlab:

```
>>U_minT=pinv([B A*B])* (-A^2*x0)
U_minT =
0.5000
```

```
-0.0000
0.5000
-1.00000
```

The minimum value of $J_{T_{\min}}$ is 1.5. One error made by a fair number of people was to find the smallest T such that $\mathcal{C}_T$ is rank 4. The answer, which is 3, gives the smallest time at which *any* state can be steered to zero. But for the specific initial state we gave, it could be done in two steps. Another error we saw was not recognizing that the minimum energy u could not be calculated by the usual formula that involves $(\mathcal{C}_T\mathcal{C}_T^T)^{-1}$; this matrix is singular. Unfortunately, Matlab was able to find the correct answer even though the formula is meaningless. (We say 'unfortunately' because we don't think people who invert singular matrices should get away with it.)

(b) *Least-norm input to achieve* $\|x(10)\| \leq 0.1$. There's no way to solve this directly, in one shot. The only way to solve it is via the trade-off between $\|x(10)\|^2$ and $\|U\|^2$. We define the cost function $J = \|x(10)\|^2 + \rho\|U\|^2$, where $\rho \geq 0$ is some parameter or weight. By varying ρ, and minimizing this cost at each step, we obtain the optimal trade-off curve between the input cost and final state cost. We then simply find the value of ρ that just achieves $\|x(10)\|^2 = 0.1^2$. Some people used Lagrange multipliers. In fact this is equivalent; the parameter ρ is a Lagrange multiplier. To minimize the cost function J, we express it as

$$J = \left\| \begin{bmatrix} \mathcal{C}_{10} \\ \sqrt{\rho} I \end{bmatrix} U + \begin{bmatrix} A^{10} \\ 0 \end{bmatrix} x(0) \right\|^2,$$

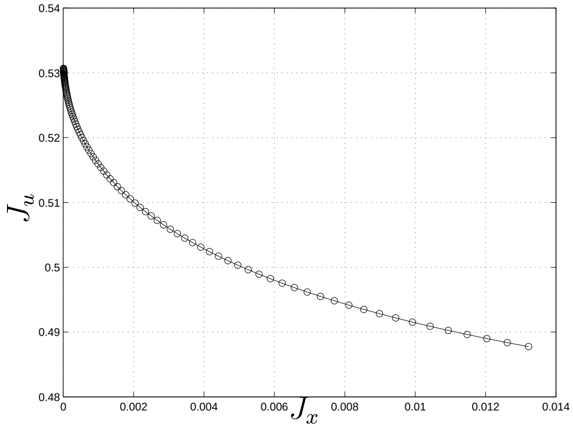
so minimizing J is a least-squares problem; the resulting optimal U is just

$$U = -(\mathcal{C}_{10}^T\mathcal{C}_{10} + \rho I)^{-1}\mathcal{C}_{10}^T A^{10} x(0).$$

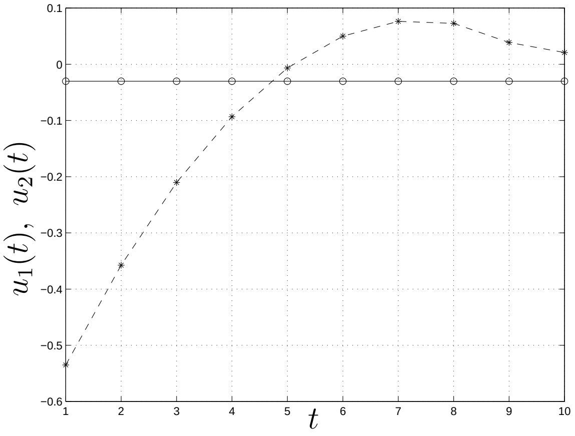
It's also possible to solve this problem via LQR (but the code is much longer ...). To solve the original problem we vary ρ until we have $\|x(10)\| = 0.1$. In Matlab:

```
C10=B; AA=A; for i=1:9,
C10 = [C10 AA*B];
AA = A*AA;
end; rho=logspace(1,-1,100); for i=1:40,
U = -inv(C10'*C10+r(i)*eye(20))*C10'*A^10*x0;
J9(i) = norm(U)^2;   % input cost
x10 = A^10*x0+C10*U;
Jx(i) = norm(x10)^2;  % final state cost
if Jx(i)<=0.01,
U
J9(i)
break;
end;
end;
```

The resulting trade-off curve is as follows:



The value of ρ that yields $\|x(10)\| = 0.1$ (*i.e.*, $J_u = 0.1^2$) is $\rho = .76$. The resulting input is shown below:



This input has $J_9 = .4915$ (which is the minimum possible value). Many people attempted to use SVDs. The idea was to find what state to hit at $t = 10$ to minimize the input energy. They then reasoned that the SVD of $\mathcal{C}_{10}$ would give a direction 'easy to get to' is a good heuristic, but doesn't solve the problem. A few people used SVDs and survived, by using Lagrange multipliers. But that's much more complicated than necessary: first of all the SVD was not needed, and second, we didn't use or cover Lagrange multipliers in this class ...