EECS 844 – Fall 2017

Exam 1 Cover page*

Each student is expected to complete the exam individually using only course notes, the book, and technical literature, and without aid from outside sources.

Aside from the most general conversation of the exam material, I assert that I have neither provided help nor accepted help from another student in completing this exam. As such, the work herein is mine and mine alone.

__________________________________ __________

Signature Date

MY SOLUTIONS

__________________________________ __________

Name (printed) Student ID #

*** Attach as cover page to completed exam.**

1. For the cost functions below, determine the derivative with respect to $\mathbf{w}^*$. You can assume that $\mathbf{R}$ is PDH as needed. (*Note: the chain rule still holds*)

a) $J(\mathbf{w}) = \tan\left(\mathbf{w}^H \mathbf{R}\mathbf{w}\right)$

b) $J(\mathbf{w}) = \|\mathbf{w}\|_1$

c) $J(\mathbf{w}) = \left|\mathbf{b}^H \mathbf{w}\right|^2$

d) $J(\mathbf{w}) = \dfrac{\mathbf{w}^H \mathbf{R}\mathbf{w}}{\left|\mathbf{w}^H \mathbf{a}\right|^2}$

Solutions:

Comment 1: See Sect. 4.1 of 2012 edition of *The Matrix Cookbook* (available free online and I encourage you all to get a copy). The complex chain rule is the standard chain rule if involving analytic functions (which means the function is complex differentiable). In short, we don't need to consider the complicated form of the chain rule for these problems.

Comment 2: Use parentheses in places where two adjacent vectors are multiplied but have incompatible dimensions. For example, $(\mathbf{b}^H\mathbf{w})\mathbf{b}$ is correct, but in $\mathbf{b}^H\mathbf{w}\mathbf{b}$ the multiplication of the last two vectors cannot be performed due to dimensionality. Of course, rearranging as $\mathbf{b}\mathbf{b}^H\mathbf{w}$ is a feasible multiplication without parentheses. Use this formalism to be technically correct and also to avoid possible mistakes.

Comment 3: When expressing math via a word processor (like you would in a paper as opposed to handwritten), make scalars italic & lowercase, make vectors bold & lowercase, and make matrices bold & uppercase. Vectors and matrices are not italicized according to general signal processing convention.

a)

$$\begin{aligned}\frac{dJ(\mathbf{w})}{d\mathbf{w}^*} &= \frac{d}{d\mathbf{w}^*}\left[\tan\left(\mathbf{w}^H\mathbf{R}\mathbf{w}\right)\right] \\ &= \mathbf{R}\mathbf{w}\sec^2\left(\mathbf{w}^H\mathbf{R}\mathbf{w}\right)\end{aligned}$$

b)

$$\begin{aligned}\frac{dJ(\mathbf{w})}{d\mathbf{w}^*} &= \frac{d}{d\mathbf{w}^*}\left[\|\mathbf{w}\|_1\right] \\ &= \frac{d}{d\mathbf{w}^*}\left[\sum_{n=1}^{N}|w_n|\right] \\ &= \frac{d}{d\mathbf{w}^*}\left[\sum_{n=1}^{N}\left(w_n^* w_n\right)^{1/2}\right]\end{aligned}$$

the *k*th component of which is

$$\frac{d}{dw_k^*}\left[\sum_{n=1}^{N}\left(w_n^* w_n\right)^{1/2}\right] = 0.5 w_k \left(w_k^* w_k\right)^{-1/2}.$$

We could write this in compact form as

$$\frac{dJ(\mathbf{w})}{d\mathbf{w}^*} = 0.5\mathbf{B}\mathbf{w}$$

where **B** is a diagonal matrix with $\left(w_k^* w_k\right)^{-1/2} = 1/|w_k|$ in the (n,n) element.

c)

$$\begin{aligned}\frac{dJ(\mathbf{w})}{d\mathbf{w}^*} &= \frac{d}{d\mathbf{w}^*}\left[\left|\mathbf{b}^H\mathbf{w}\right|^2\right] \\ &= \frac{d}{d\mathbf{w}^*}\left[\left(\mathbf{b}^H\mathbf{w}\right)\left(\mathbf{b}^H\mathbf{w}\right)^*\right] \\ &= \frac{d}{d\mathbf{w}^*}\left[\left(\mathbf{b}^H\mathbf{w}\right)\left(\mathbf{w}^H\mathbf{b}\right)\right] \\ &= \left(\mathbf{b}^H\mathbf{w}\right)\mathbf{b}\end{aligned}$$

d)

$$\frac{dJ(\mathbf{w})}{d\mathbf{w}^*} = \frac{d}{d\mathbf{w}^*}\left[\frac{\mathbf{w}^H\mathbf{R}\mathbf{w}}{\left|\mathbf{w}^H\mathbf{a}\right|^2}\right]$$

$$= \frac{d}{d\mathbf{w}^*}\left[\frac{\mathbf{w}^H\mathbf{R}\mathbf{w}}{\left(\mathbf{w}^H\mathbf{a}\right)\left(\mathbf{w}^H\mathbf{a}\right)^*}\right]$$

$$= \frac{d}{d\mathbf{w}^*}\left[\frac{\mathbf{w}^H\mathbf{R}\mathbf{w}}{\left(\mathbf{w}^H\mathbf{a}\right)\left(\mathbf{a}^H\mathbf{w}\right)}\right]$$

$$= \frac{\left(\mathbf{w}^H\mathbf{a}\right)\left(\mathbf{a}^H\mathbf{w}\right)\frac{d}{d\mathbf{w}^*}\left[\mathbf{w}^H\mathbf{R}\mathbf{w}\right] - \left(\mathbf{w}^H\mathbf{R}\mathbf{w}\right)\frac{d}{d\mathbf{w}^*}\left[\left(\mathbf{w}^H\mathbf{a}\right)\left(\mathbf{a}^H\mathbf{w}\right)\right]}{\left[\left(\mathbf{w}^H\mathbf{a}\right)\left(\mathbf{a}^H\mathbf{w}\right)\right]^2}$$

$$= \frac{\left(\mathbf{w}^H\mathbf{a}\right)\left(\mathbf{a}^H\mathbf{w}\right)\mathbf{R}\mathbf{w} - \left(\mathbf{w}^H\mathbf{R}\mathbf{w}\right)\left(\mathbf{a}^H\mathbf{w}\right)\mathbf{a}}{\left|\mathbf{w}^H\mathbf{a}\right|^2\left(\mathbf{w}^H\mathbf{a}\right)\left(\mathbf{a}^H\mathbf{w}\right)}$$

$$= \frac{\left(\mathbf{w}^H\mathbf{a}\right)\mathbf{R}\mathbf{w} - \left(\mathbf{w}^H\mathbf{R}\mathbf{w}\right)\mathbf{a}}{\left|\mathbf{w}^H\mathbf{a}\right|^2\left(\mathbf{w}^H\mathbf{a}\right)}$$

2. For the length K time-series data in P2.mat, use Appendix A to form the $M \times N$ delay-shifted snapshot matrix $\mathbf{X}$ for $M = 20$ (length of each snapshot and also the number of rows in $\mathbf{X}$) and $N = K - M + 1$. Starting with the 1st column in $\mathbf{X}$, estimate the correlation matrix using different numbers of snapshots as

$$\mathbf{R}_n = \left(\frac{1}{n}\right)\sum_{\ell=1}^{n} \mathbf{x}(M+\ell-1)\,\mathbf{x}^H(M+\ell-1) \quad \text{for} \quad n = 1, 2, \cdots, N .$$

For each of these correlation matrices, determine the eigenvalues having the maximum and minimum absolute values. As a function of n, plot (in dB) the maximum eigenvalues. Likewise plot (in dB) the minimum eigenvalues as a function of n. Note that eigenvalues are in units of power in this case.

What can we infer regarding the condition number and invertibility of the correlation matrix as a function of the number of snapshots used to estimate it?

Solution:

The maximum and minimum eigenvalues of $\mathbf{R}$ as a function of the number of snapshots are plotted in Fig. 2.1. Note that, which theoretically these eigenvalues should be real and non-negative, numerical effects can introduce small imaginary and/or negative values when the number of snapshots is very low (or when the matrix is rank deficient).

We observe in Fig. 2.1 that when the number of snapshots is less than $M = 20$ (the size of $\mathbf{R}$), the maximum and minimum eigenvalues change significantly. In fact, for $n < M$ the minimum eigenvalue is essentially zero (around –150 dB) which means the matrix is not invertible. For $n > M$ the minimum eigenvalue settles quickly while the maximum eigenvalue still fluctuates some, though this effect tends to not be much of an issue. For independent and identically distributed (IID) data, the rule is $n > 2M$ snapshots are needed for the estimate of $\mathbf{R}$ to get within 3 dB of optimum (this statement comes from a form of interference cancellation known as maximum SINR filtering).

Comment 1: Note that the 'eig' function in Matlab often (but not always!) provides the eigenvalues in descending order (with associated eigenvectors ordered accordingly). This is something you want to check for.

Comment 2: Note that correlation and covariance are associated with power (not amplitude) so you would plot eigenvalues using 10 log and not 20 log.

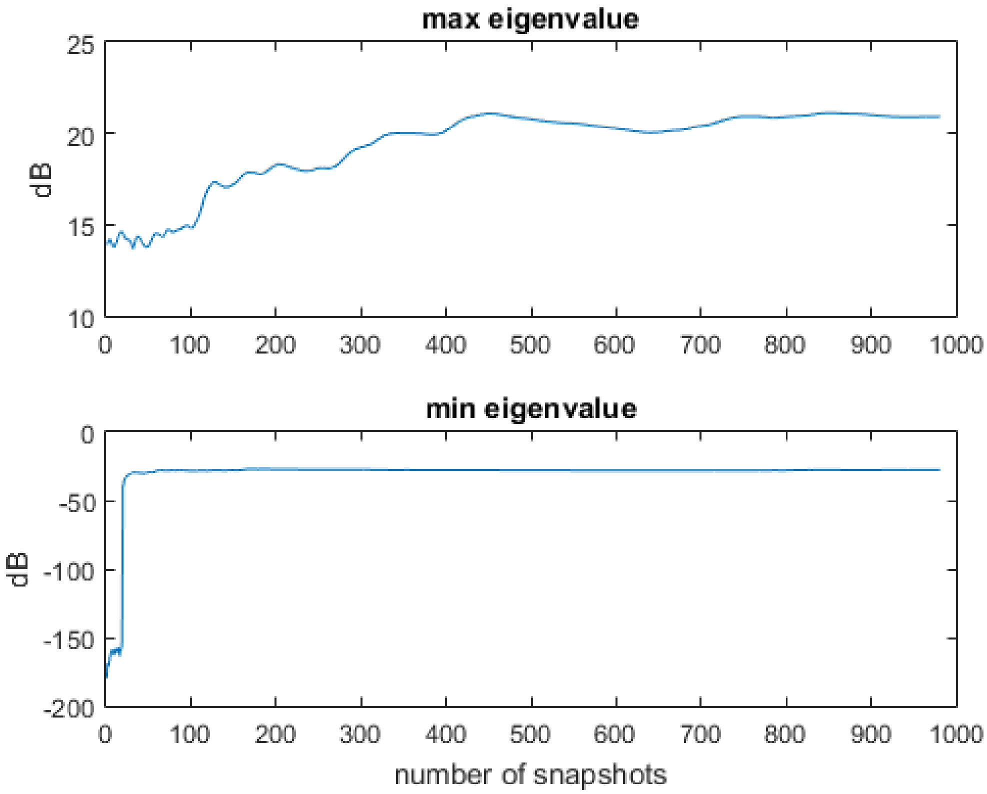

Figure 2.1. Maximum and minimum eigenvalues of **R** as a function of the number of snapshots

Matlab Code for Problem 2

```
clear all;
load P2
N = length(x);

M = 20;

for jm = 1:M
    X(M-jm+1,:) = x(jm:jm+N-M).';
end;
[M,K] = size(X);

for n = 1:K
    R = (1/n).*X(:,1:n)*X(:,1:n)';
    [V,D] = eig(R);
    lam_max(n) = max(abs(diag(D)));
    lam_min(n) = min(abs(diag(D)));
end;

figure(21)
subplot(2,1,1)
plot(1:K,10*log10(lam_max));
ylabel('dB')
title('max eigenvalue')
subplot(2,1,2)
plot(1:K,10*log10(lam_min));
ylabel('dB')
title('min eigenvalue')
xlabel('number of snapshots')
```

3. Repeat problem 2 using "diagonally loading" of each correlation matrix as

$$\mathbf{R}_n = \left[\left(\frac{1}{n} \right) \sum_{\ell=1}^{n} \mathbf{x}(M+\ell-1)\, \mathbf{x}^H(M+\ell-1) \right] + \sigma^2 \mathbf{I} \quad \text{for} \quad n = 1, 2, \cdots, N$$

where $\mathbf{I}$ is an $M \times M$ identity matrix and setting $\sigma^2 = 1$ (in one case) and $\sigma^2 = 10$ (in another case), and $\sigma^2 = 100$ (in the last case). Like problem 2, plot the maximum and minimum eigenvalue magnitudes (in dB) for these three cases along with the previous unloaded case. What do you observe in comparison to the Problem 2 results? What are the implications to the condition number and matrix invertibility as a function of the number of snapshots when diagonal loading is used? What is the impact of different values of the loading factor σ^2?

<u>Solution:</u>

The maximum and minimum eigenvalues for the unloaded (from Prob. 2) and diagonally loaded versions of $\mathbf{R}$ are plotted in Fig. 3.1. For the three diagonally loaded cases the minimum eigenvalue appears to simply become the loading factor (it is not exact but it is very close). In contrast, the loading appears to offset the maximum eigenvalue according to the value of the loading factor, with $\sigma^2 = 100$ having a significant effect. Since all three of these cases have non-zero minimum eigenvalues, they are invertible regardless of the number of snapshots used and thus will have a manageable condition number as well.

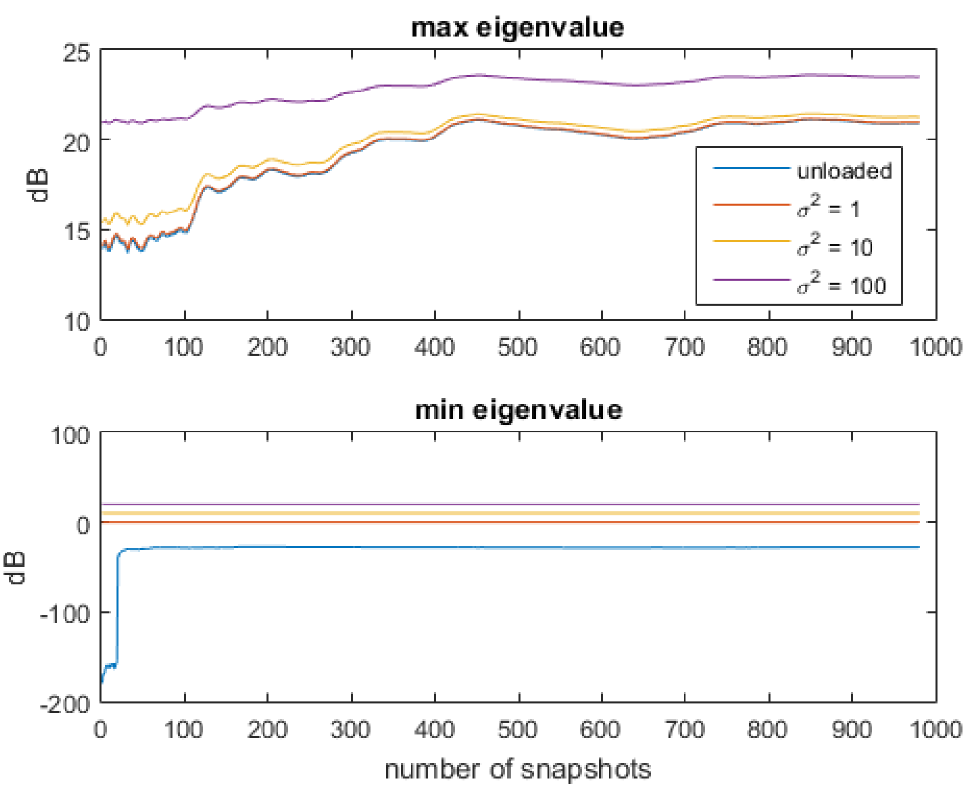

Figure 3.1. Maximum and minimum eigenvalues of $\mathbf{R}$ as a function of the number of snapshots for diagonally loaded cases

Matlab Code for Problem 3

```
clear all;
load P2
N = length(x);

M = 20;

for jm = 1:M
    X(M-jm+1,:) = x(jm:jm+N-M).';
end;
[M,K] = size(X);

for n = 1:K
    R1 = (1/n).*X(:,1:n)*X(:,1:n)' + eye(M,M);
    [V1,D1] = eig(R1);
    lam_max1(n) = max(abs(diag(D1)));
    lam_min1(n) = min(abs(diag(D1)));

    R10 = (1/n).*X(:,1:n)*X(:,1:n)' + 10.*eye(M,M);
    [V10,D10] = eig(R10);
    lam_max10(n) = max(abs(diag(D10)));
    lam_min10(n) = min(abs(diag(D10)));

    R100 = (1/n).*X(:,1:n)*X(:,1:n)' + 100.*eye(M,M);
    [V100,D100] = eig(R100);
    lam_max100(n) = max(abs(diag(D100)));
    lam_min100(n) = min(abs(diag(D100)));
end;

figure(31)
subplot(2,1,1)
plot(1:K,10*log10(lam_max),1:K,10*log10(lam_max1),1:K,10*log10(lam_max1
0),1:K,10*log10(lam_max100));
ylabel('dB')
title('max eigenvalue')
legend('unloaded','\sigma^{2} = 1','\sigma^{2} = 10','\sigma^{2} =
100')
subplot(2,1,2)
plot(1:K,10*log10(lam_min),1:K,10*log10(lam_min1),1:K,10*log10(lam_min1
0),1:K,10*log10(lam_min100));
ylabel('dB')
title('min eigenvalue')
xlabel('number of snapshots')
```

4. Again using the data from Problem 2, for $n = N$ (all the snapshots) plot the complete set of eigenvalues (there are $M = 20$) for each of the four different estimates of the correlation matrix (i.e. unloaded from Prob. 2 and for each of the three different loading factors from Prob. 3). What do you observe occurring as the loading factor is increased?

<u>Solution:</u>

Each set of M eigenvalues is plotted in Fig. 4.1 where it is observed that the original unloaded case (from Prob. 2) has the highest dynamic range. The three diagonally loaded cases flatten out the smaller eigenvalues at a minimum level that is commensurate with the respective loading factors of 0 dB, 10 dB, and 20 dB (i.e. 1, 10, and 100). We therefore see that diagonally loading provides a way to control the condition number and thus invertibility of a correlation matrix, though it can come at the cost of lost information in the correlation matrix if the loading factor is too large (it dominates the smaller eigenvalues).

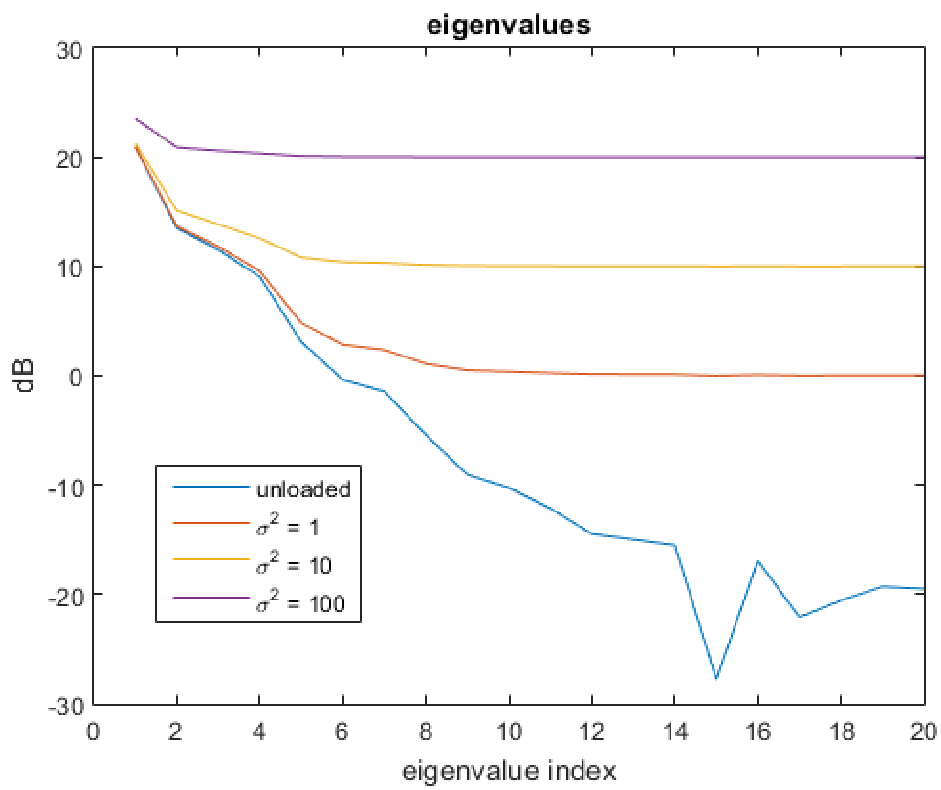

Figure 4.1. Eigenvalues of **R** for different diagonal loading factors

Matlab Code for Problem 4

```
clear all;
load P2
N = length(x);

M = 20;

for jm = 1:M
    X(M-jm+1,:) = x(jm:jm+N-M).';
end;
[M,K] = size(X);

R = (1/n).*X(:,1:n)*X(:,1:n)';
[V,D] = eig(R);

R1 = (1/n).*X(:,1:n)*X(:,1:n)' + eye(M,M);
[V1,D1] = eig(R1);

R10 = (1/n).*X(:,1:n)*X(:,1:n)' + 10.*eye(M,M);
[V10,D10] = eig(R10);

R100 = (1/n).*X(:,1:n)*X(:,1:n)' + 100.*eye(M,M);
[V100,D100] = eig(R100);


figure(41)
plot(1:M,10*log10(diag(D)),1:M,10*log10(diag(D1)),1:M,10*log10(diag(D10
)),1:M,10*log10(diag(D100)));
ylabel('dB')
title('eigenvalues')
legend('unloaded','\sigma^{2} = 1','\sigma^{2} = 10','\sigma^{2} =
100')
xlabel('eigenvalue index')
```

5. The dataset P5.mat contains N = 200 time samples collected from an M = 15 element antenna array (this is spatial data). This data is already collected into snapshot form, with each snapshot (of length M) corresponding to the samples obtained from the antenna elements at a single time instant. Using the complete set of snapshots, estimate the spatial correlation matrix $\mathbf{R}$ and plot its eigenvalues (in dB). What do you observe about the eigenvalue structure?

Solution:

The spatial correlation matrix corresponding to a 15 element antenna array has a dimensionality of 15×15. The spatial snapshot matrix $\mathbf{X}$ is 15×200, with each column representing the samples captured by the array at a given time instant. In other words, $\mathbf{X}$ is oriented as "number of antenna elements" $\times$ "number of time samples". Therefore, computing $\mathbf{R} = (1/N)\, \mathbf{X}\mathbf{X}^H$ produces the spatial correlation matrix. The eigenvalues of $\mathbf{R}$ are plotted in Fig. 5.1. The relatively flat region looks like what we observed previously when diagonal loading was used, though none was employed here. This effect represents a clear delineation between a "signal + noise" subspace (the larger eigenvalues) and a "noise only" subspace (the relatively flat region of small eigenvalues). This effect is not specific to spatial data, since it could likewise occur for temporal data if there is sufficient signal structure present (and if our "model order" is large enough to capture it).

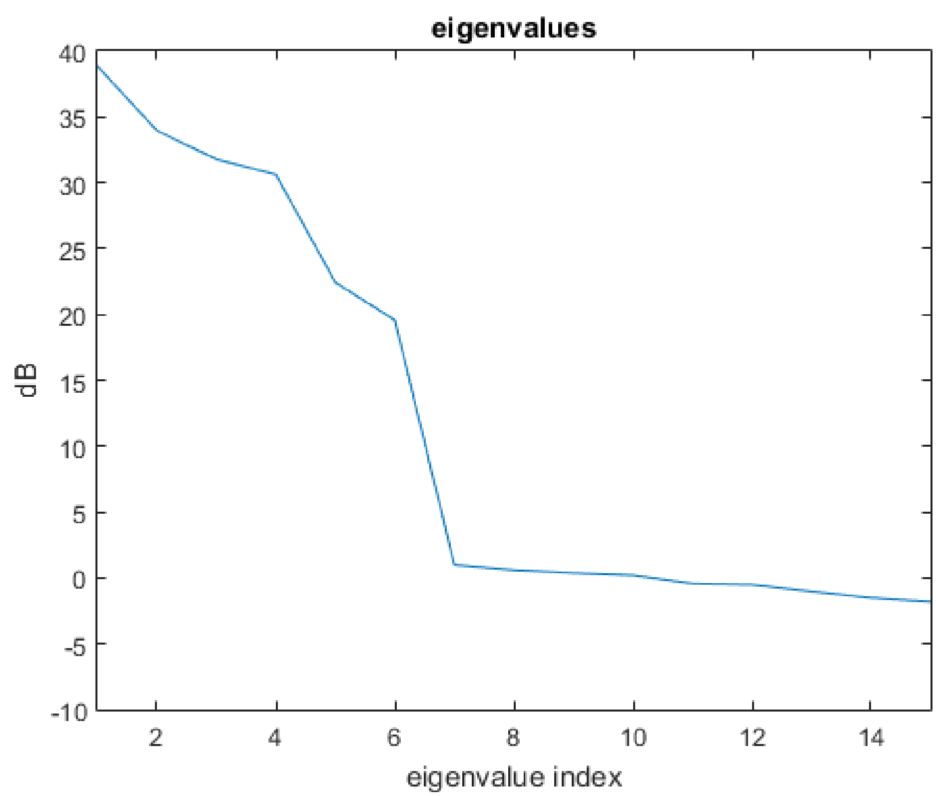

Figure 5.1. Eigenvalues of $\mathbf{R}$ for spatial data

Matlab Code for Problem 5

```
clear all
load P5

[M N] = size(X);

R = X*X'./N;
[V,D] = eig(R);

figure(51)
plot(1:M,10*log10(flipud(diag(D))));
axis([1 M -10 40])
ylabel('dB')
title('eigenvalues')
xlabel('eigenvalue index')
```

6. The dataset P6.mat contains two length M = 14 antenna array filters (i.e. beamformers) for a uniform linear array whose elements are separated by a half-wavelength. The filter 'w_non_adap' is a non-adaptive filter while the filter 'w_adap' is an adaptive filter. Using Appendices B and C, plot the beampatterns of these two filters in terms of electrical angle and spatial angle (plot in dB). Discuss what you observe.

<u>Solution:</u>

The beampatterns in terms of electrical angle and spatial angle are plotted in Figs. 6-1 and 6-2, respectively. The adaptive response appears to be poorer since it has slightly higher sidelobes. However, the data used to form the adaptive filter contains interference sources at electrical angles −175.7°, −129.6°, +33.9°, and +161.5°, for which Fig. 6-1 show nulls in the adaptive beampattern corresponding to these angles. The spatial null in the region of electrical angle +34° is particularly prominent.

The seemingly higher sidelobes in the adaptive response actually do not degrade performance since no signals are presently arriving from those directions (or else they would have been present in the data). Of course, there is an assumption here that the data is sufficiently stationary that the interference does not change locations and new sources do not appear.

The nonlinear relationship between electrical angle and spatial angle (due to the sin(•) function) can also be observed. The electrical angle sidelobes are uniformly spaced while the spatial sidelobes are nonlinear in nature (narrower near boresight and wider near endfire).

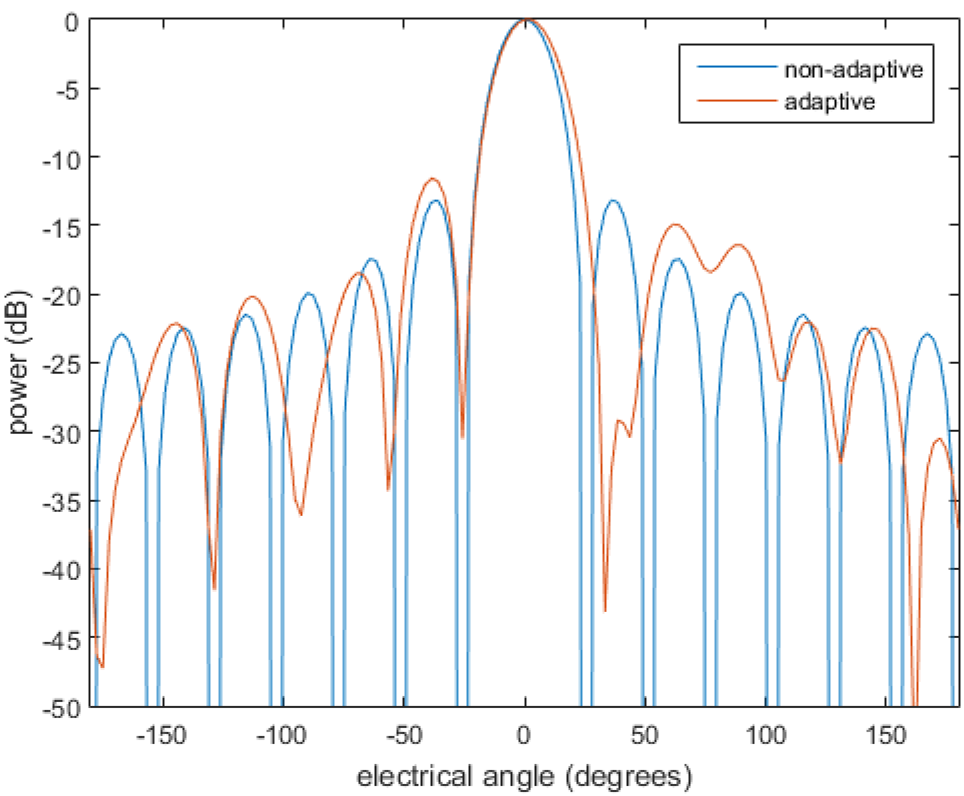

Figure 6.1. Beampatterns in terms of electrical angle ($-\pi \leq \theta \leq \pi$).

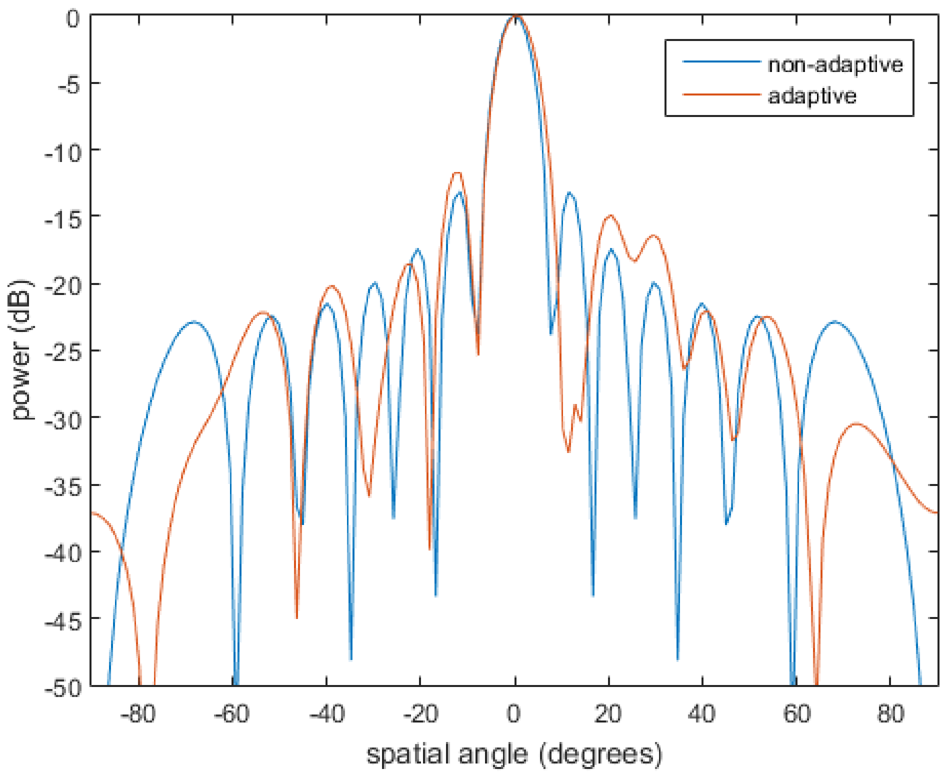

Figure 6.2. Beampatterns in terms of spatial angle ($-\pi/2 \leq \phi \leq \pi/2$).

Matlab Code for Problem 6

```
clear all
load P6

M = length(w_adap);

angs_spat = linspace(-90,90,M*10+1);
angs_elec = linspace(-180,180,M*10+1);

for theta_i = 1:length(angs_spat)
    s_spat = (exp(-j*pi*sin((angs_spat(theta_i)*pi/180)).*[0:M-1]).');
    s_elec = (exp(-j*(angs_elec(theta_i)*pi/180).*[0:M-1]).');
    S_spat(1:M,theta_i) = s_spat;
    S_elec(1:M,theta_i) = s_elec;
end;

pattern_nonadap_spat = abs(S_spat'*w_non_adap);
pattern_nonadap_elec = abs(S_elec'*w_non_adap);
pattern_adap_spat = abs(S_spat'*w_adap);
pattern_adap_elec = abs(S_elec'*w_adap);

figure(61)
plot(angs_elec,20*log10(pattern_nonadap_elec),angs_elec,20*log10(patter
n_adap_elec))
axis([-180 180 -50 0])
xlabel('electrical angle (degrees)')
ylabel('power (dB)')
legend('non-adaptive','adaptive')

figure(62)
plot(angs_spat,20*log10(pattern_nonadap_spat),angs_spat,20*log10(patter
n_adap_spat))
axis([-90 90 -50 0])
xlabel('spatial angle (degrees)')
ylabel('power (dB)')
legend('non-adaptive','adaptive')
```

7. For each of the cost functions in Problem 1, apply the linear constraint $\mathbf{w}^H\mathbf{s}=1$ and <u>solve for the complex Lagrange multiplier</u> (you do <u>not</u> need to solve for the filter). You can assume all matrices are PDH. (*In each case, pre-multiply by* $\mathbf{s}^H$ *when solving*)

<u>Solutions:</u>

For all four cases, define $c(\mathbf{w})=\mathbf{w}^H\mathbf{s}-1$.

Reminder: You <u>cannot</u> divide by a vector or matrix. To solve for λ (after taking the derivative) pre-multiply by $\mathbf{s}^H$ to obtain an inner product that results in scalar terms that can then be manipulated (i.e. divided by).

a) Supplement the cost function with the constraint as

$$h(\mathbf{w})=\tan\left(\mathbf{w}^H\mathbf{R}\mathbf{w}\right)+\mathrm{Re}\left\{\lambda^*\left(\mathbf{w}^H\mathbf{s}-1\right)\right\}$$

so

$$\begin{aligned}\frac{dh(\mathbf{w})}{d\mathbf{w}^*}&=\frac{d}{d\mathbf{w}^*}\left[\tan\left(\mathbf{w}^H\mathbf{R}\mathbf{w}\right)+\mathrm{Re}\left\{\lambda^*\left(\mathbf{w}^H\mathbf{s}-1\right)\right\}\right]\\&=\mathbf{R}\mathbf{w}\sec^2\left(\mathbf{w}^H\mathbf{R}\mathbf{w}\right)+0.5\lambda^*\mathbf{s}=\mathbf{0}_{M\times 1}\end{aligned}.$$

Rearrange this equation, pre-multiply by $\mathbf{s}^H$, and then solve for the Lagrange multiplier:

$$\lambda=-\sec^2\left(\mathbf{w}^H\mathbf{R}\mathbf{w}\right)\left(\frac{\mathbf{w}^H\mathbf{R}\mathbf{s}}{0.5\mathbf{s}^H\mathbf{s}}\right)$$

where we have used $\left(\mathbf{s}^H\mathbf{R}\mathbf{w}\right)^*=\left(\mathbf{s}^H\mathbf{R}\mathbf{w}\right)^H=\mathbf{w}^H\mathbf{R}\mathbf{s}$ since the term in parentheses is a scalar and noting that $\mathbf{R}=\mathbf{R}^H$ since assumed to be PDH.

b) Supplement the cost function with the constraint as

$$h(\mathbf{w}) = \|\mathbf{w}\|_1 + \mathrm{Re}\left\{\lambda^* \left(\mathbf{w}^H \mathbf{s} - 1\right)\right\}$$

so

$$\begin{aligned}\frac{dh(\mathbf{w})}{d\mathbf{w}^*} &= \frac{d}{d\mathbf{w}^*}\left[\|\mathbf{w}\|_1 + \mathrm{Re}\left\{\lambda^* \left(\mathbf{w}^H \mathbf{s} - 1\right)\right\}\right] \\ &= 0.5\mathbf{B}\mathbf{w} + 0.5\lambda^* \mathbf{s} = \mathbf{0}_{M\times 1}\end{aligned}$$

Rearrange this equation, pre-multiply by $\mathbf{s}^H$, and then solve for the Lagrange multiplier:

$$\lambda = \frac{-\mathbf{w}^H \mathbf{B}\mathbf{s}}{\left(\mathbf{s}^H \mathbf{s}\right)}$$

noting that $\mathbf{B} = \mathbf{B}^*$ since it is diagonal and real-valued.

c) Supplement the cost function with the constraint as

$$h(\mathbf{w}) = \left|\mathbf{b}^H \mathbf{w}\right|^2 + \mathrm{Re}\left\{\lambda^* \left(\mathbf{w}^H \mathbf{s} - 1\right)\right\}$$

so

$$\begin{aligned}\frac{dh(\mathbf{w})}{d\mathbf{w}^*} &= \frac{d}{d\mathbf{w}^*}\left[\left|\mathbf{b}^H \mathbf{w}\right|^2 + \mathrm{Re}\left\{\lambda^* \left(\mathbf{w}^H \mathbf{s} - 1\right)\right\}\right] \\ &= \left(\mathbf{b}^H \mathbf{w}\right)\mathbf{b} + 0.5\lambda^* \mathbf{s} = \mathbf{0}_{M\times 1}\end{aligned}$$

Rearrange this equation, pre-multiply by $\mathbf{s}^H$, and then solve for the Lagrange multiplier:

$$\lambda = \frac{-\left(\mathbf{w}^H \mathbf{b}\right)\left(\mathbf{b}^H \mathbf{s}\right)}{\left(0.5\mathbf{s}^H \mathbf{s}\right)}$$

d) Supplement the cost function with the constraint as

$$h(\mathbf{w}) = \frac{\mathbf{w}^H \mathbf{R} \mathbf{w}}{\left|\mathbf{w}^H \mathbf{a}\right|^2} + \operatorname{Re}\left\{\lambda^* \left(\mathbf{w}^H \mathbf{s} - 1\right)\right\}$$

so

$$\frac{dh(\mathbf{w})}{d\mathbf{w}^*} = \frac{d}{d\mathbf{w}^*}\left[\frac{\mathbf{w}^H \mathbf{R} \mathbf{w}}{\left|\mathbf{w}^H \mathbf{a}\right|^2} + \operatorname{Re}\left\{\lambda^* \left(\mathbf{w}^H \mathbf{s} - 1\right)\right\}\right]$$
$$= \frac{\left(\mathbf{w}^H \mathbf{a}\right)\mathbf{R}\mathbf{w} \; - \; \left(\mathbf{w}^H \mathbf{R} \mathbf{w}\right)\mathbf{a}}{\left|\mathbf{w}^H \mathbf{a}\right|^2 \left(\mathbf{w}^H \mathbf{a}\right)} + 0.5\lambda^* \mathbf{s} = \mathbf{0}_{M \times 1}$$

Rearrange this equation, pre-multiply by $\mathbf{s}^H$, and then solve for the Lagrange multiplier:

$$\lambda = \frac{-\left(\mathbf{a}^H \mathbf{w}\right)\mathbf{w}^H \mathbf{R} \mathbf{s} \; + \; \left(\mathbf{w}^H \mathbf{R} \mathbf{w}\right)\left(\mathbf{a}^H \mathbf{s}\right)}{\left|\mathbf{w}^H \mathbf{a}\right|^2 \left(\mathbf{a}^H \mathbf{w}\right)\left(0.5 \mathbf{s}^H \mathbf{s}\right)}.$$