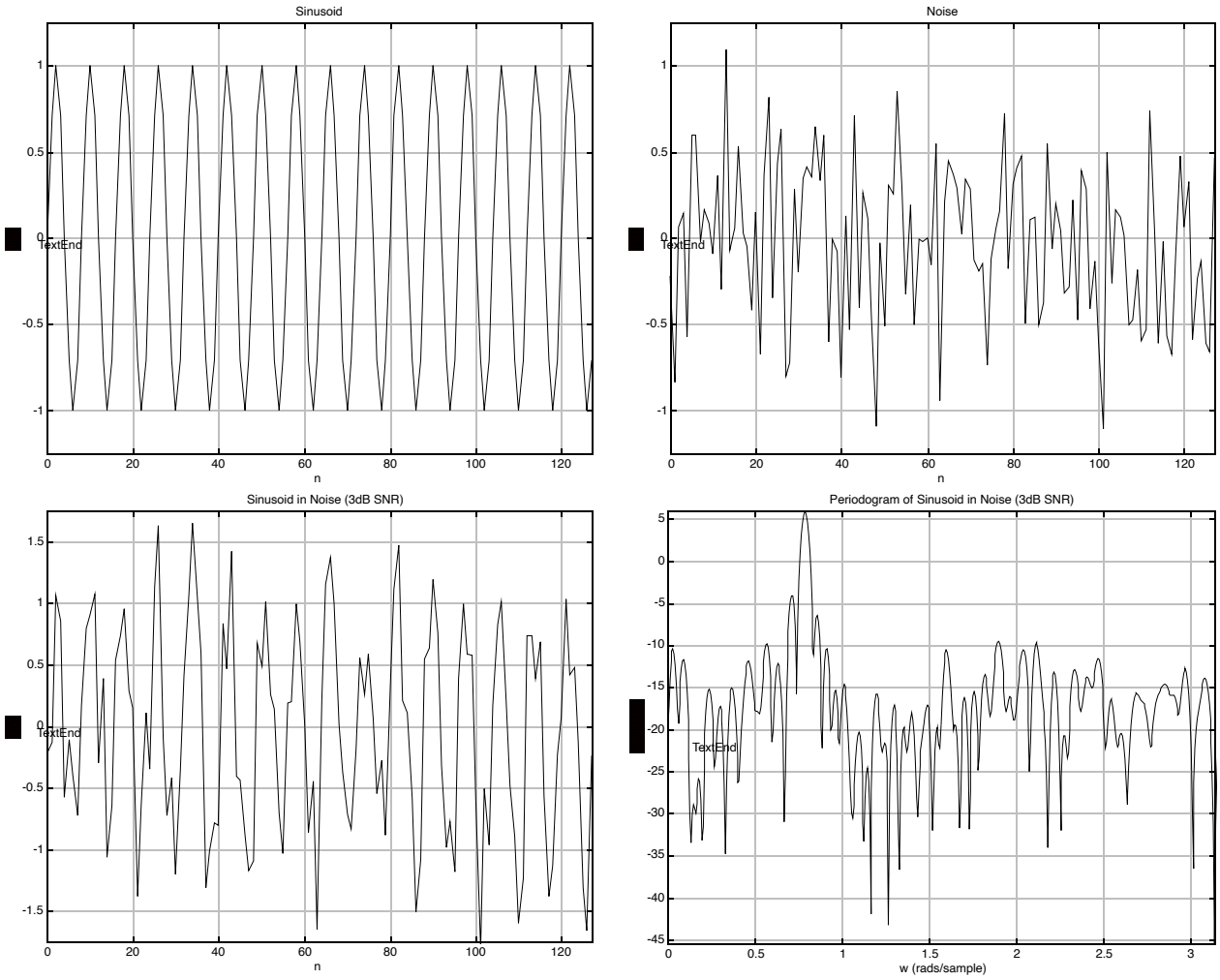
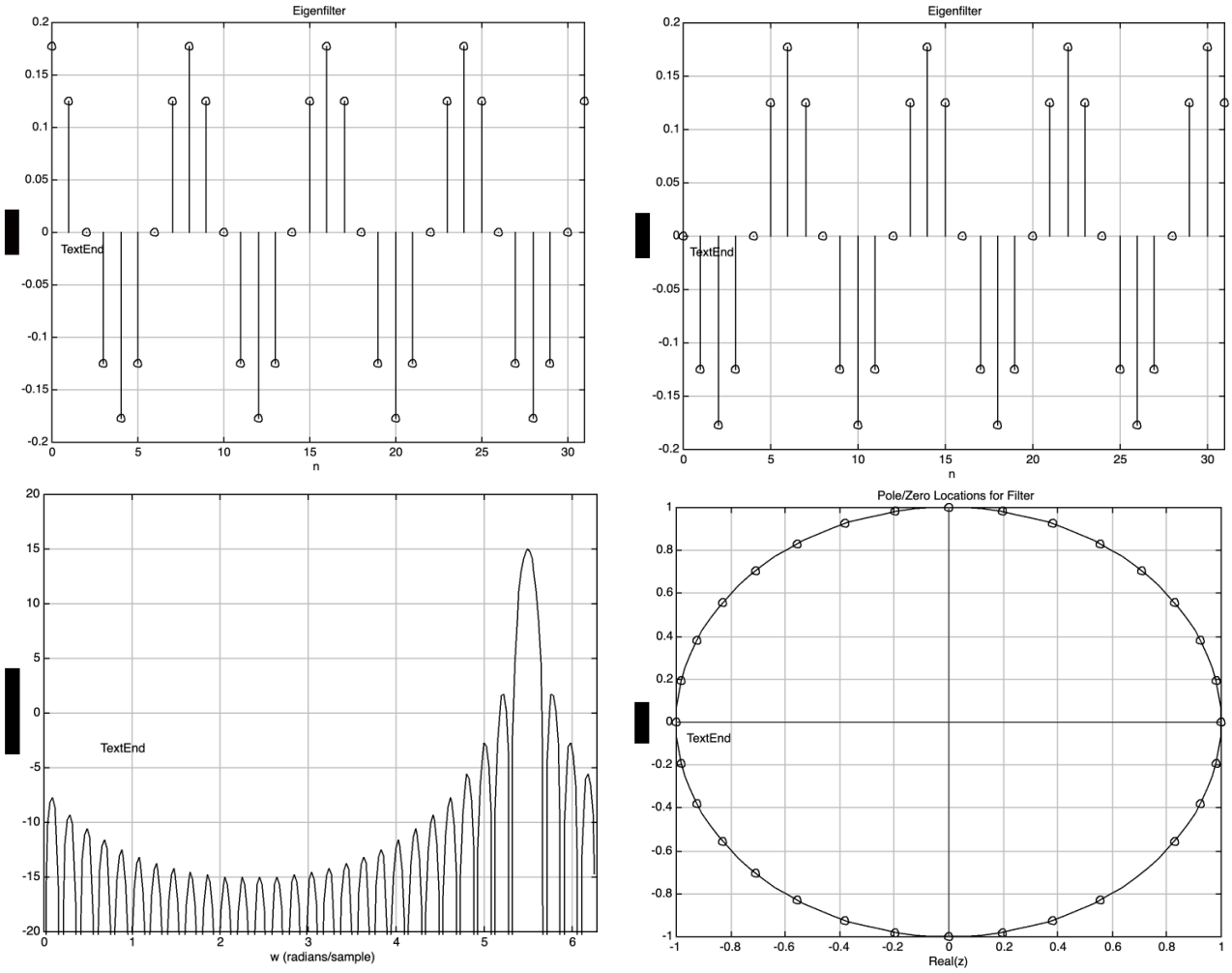
Solution #2 Eigenanalysis

1) (a) (See attached)

(b)



2)
(a)

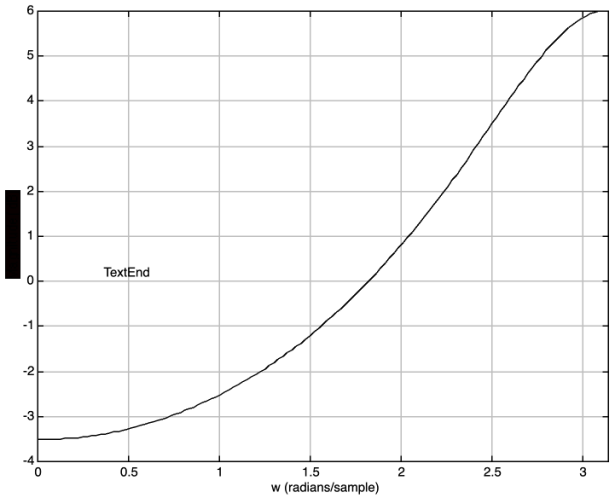


Clearly, this filter is a peaking filter (see magnitude response plot and pole/zero plot) with the peak at $7\pi / 4$. When the convolution is performed, the output of the filter peaks at $\pi / 4$.

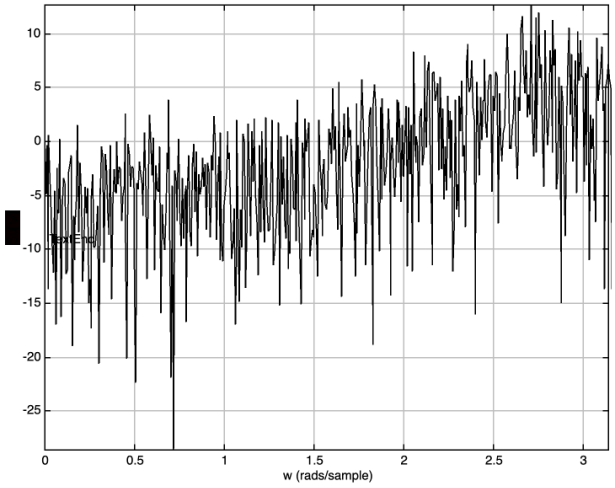
(b) (This part was stated incorrectly in the homework. The eigenfilter should be applied to a noisy, complex sinusoid rather than the noisy sinusoid of Prob. 1b)

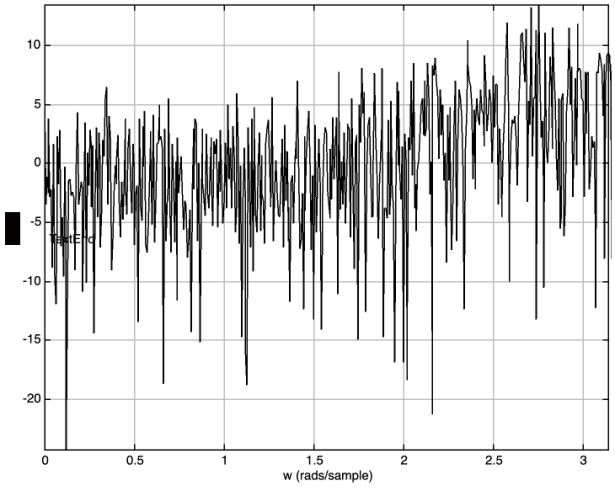
(c) With a longer filter we have more degrees of freedom and would be able to make a peaking filter with a narrower passband.

3)
(a)

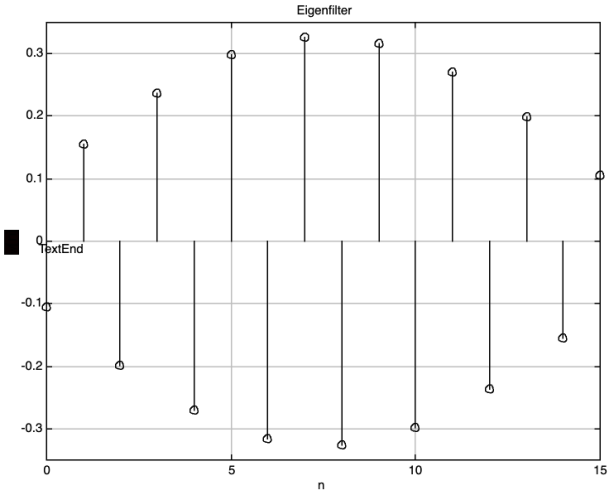


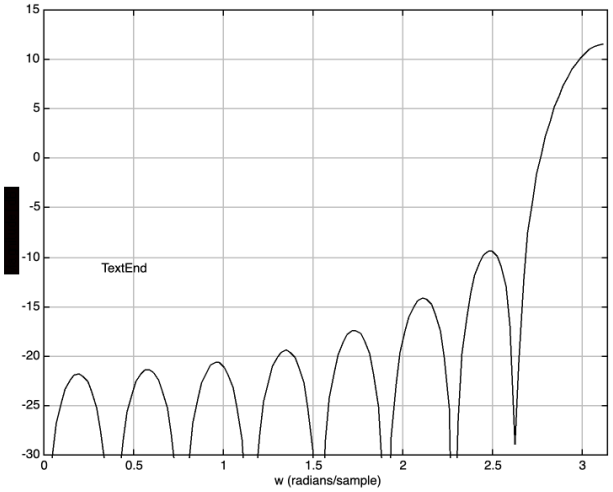
(b)

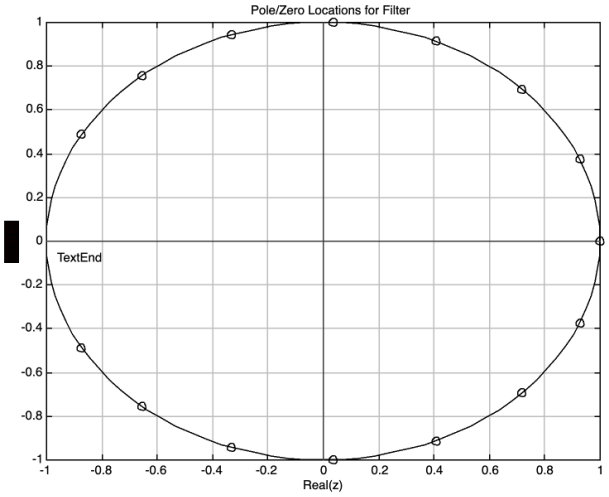




(c)

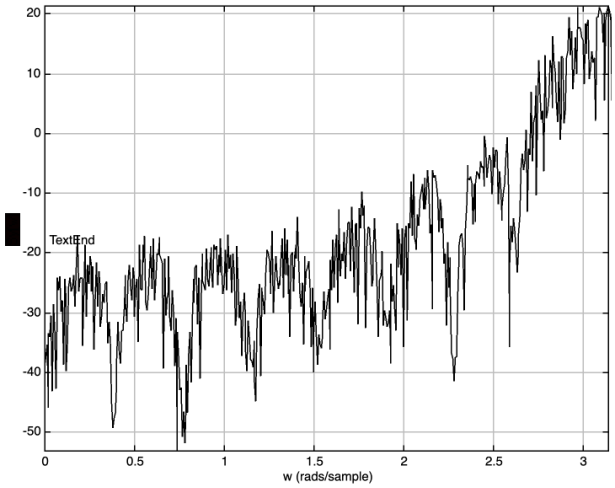






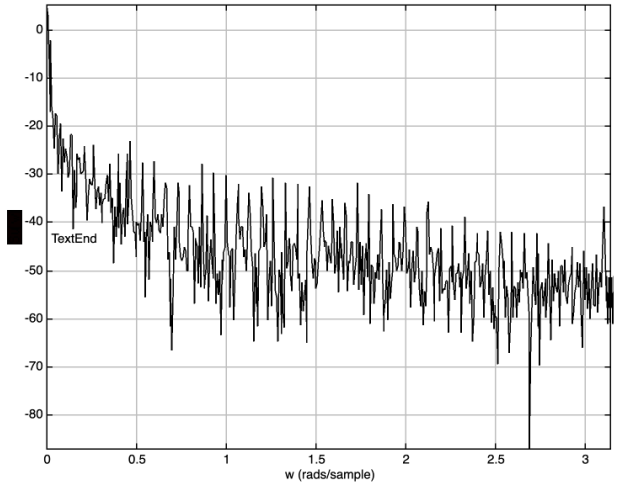
The eigenfilter is highpass which is exactly like the AR model.

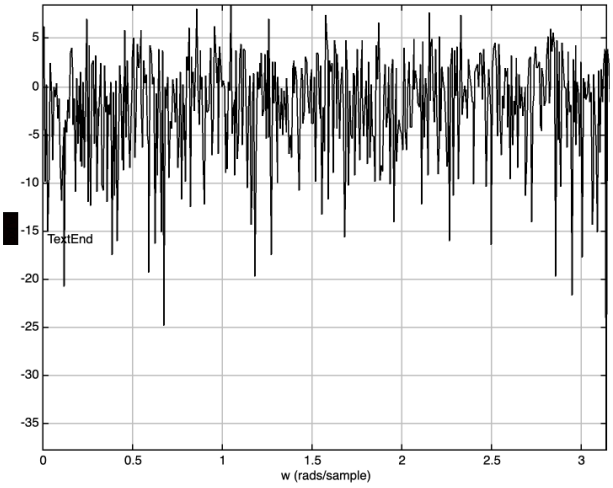
(d)



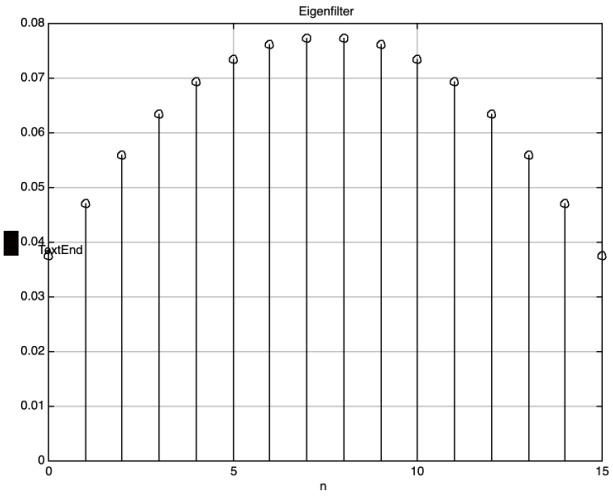
4)

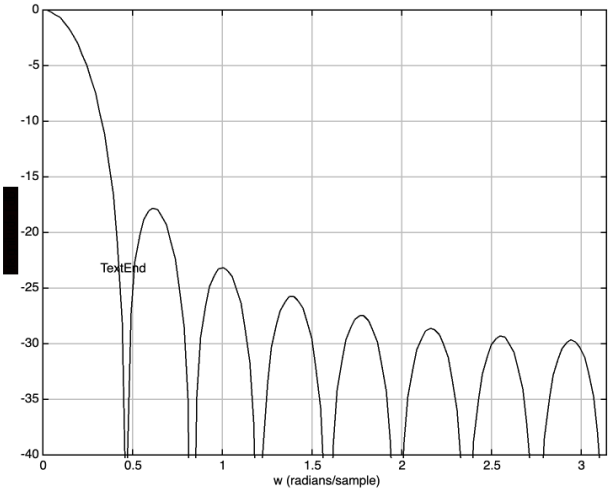
(a)


TextEnd
w (rads/sample)

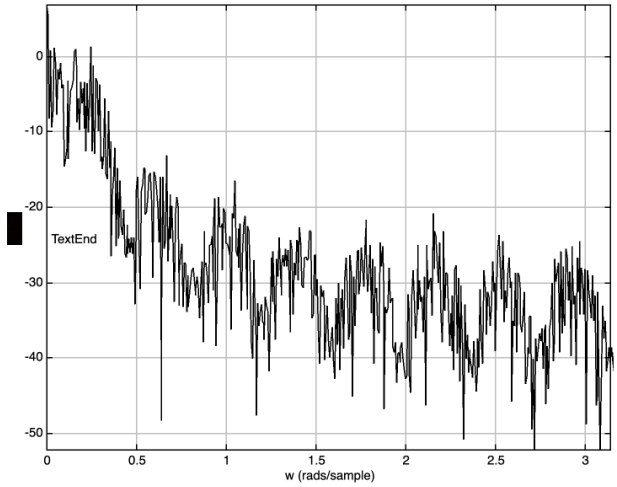

TextEnd
w (rads/sample)

(b)


Eigenfilter
TextEnd
n


TextEnd
w (radians/sample)

(c)


TextEnd
w (rads/sample)

---

5)
(a) Other speech signals will have different **R** and **p**.

```
R =
    1.0000    0.9202    0.7682    0.5982
    0.9202    1.0000    0.9202    0.7682
    0.7682    0.9202    1.0000    0.9202
    0.5982    0.7682    0.9202    1.0000

p =
    1.9468
    2.0188
    1.9044
    1.6504
```

(b)

```
w =
    0.9998
    0.5001
    0.4000
    0.3000
```

(c)

```
w =
    1.0015
    0.3439
    0.8186
```

With the shorter filter, **w** minimizes the squared error.

(d)

```
w =
    0.9998
    0.5001
    0.4000
    0.3001
   -0.0000
```

With the longer filter, **w** needs only match **h** (which minimizes the squared error) and other coefficients are set to zero.

```
% EE594 - Fall 2002 - Homework #2

%---
% 1
%---
% b
N = 128; %signal length
L = 1024; % periodogram length
w0 = pi/4;
SNR = 3;  % in dB
randn('state',0);

n = [0:N-1]';
u = sin(w0.*n);
figure(1);
plot(n,u,'k');
axis([0 N-1 -1.25 1.25]);
title('Sinusoid');
ylabel('u(n)');
xlabel('n');
grid;

sigma_u2 = cov(u);  % 1/2 for a sinusoid
sigma_v2 = sigma_u2*10^(-SNR/10)
v = randn(N,1) .* sqrt(sigma_v2);
figure(2);
plot(n,v,'k');
axis([0 N-1 -1.25 1.25]);
title('Noise');
ylabel('v(n)');
xlabel('n');
grid;

x = u+v;
figure(3);
plot(n,x,'k');
title('Sinusoid in Noise (3dB SNR)');
axis([0 N-1 -1.75 1.75]);
ylabel('x(n)');
xlabel('n');
grid;

Sx = periodogram(x,L);
figure(4);
periodogram_plot(Sx);
title('Periodogram of Sinusoid in Noise (3dB SNR)');

% c
N = 128;
w0 = pi/4;
SNR = -10;  % in dB
randn('state',0);

n = [0:N-1]';
u = sin(w0.*n);
figure;
plot(n,u,'k');
```

```
title('Sinusoid');
ylabel('u(n)');
xlabel('n');
grid;

sigma_u2 = cov(u);  % 1/2 for a sinusoid
sigma_v2 = sigma_u2*10^(-SNR/10)
v = randn(N,1) .* sqrt(sigma_v2);
figure;
plot(n,v,'k');
title('Noise');
ylabel('v(n)');
xlabel('n');
grid;

x = u+v;
figure;
plot(n,x,'k');
title('Sinusoid in Noise (-10dB SNR)');
ylabel('x(n)');
xlabel('n');
grid;

Sx = periodogram(x,8*N);
figure;
periodogram_plot(Sx);
title('Periodogram of Sinusoid in Noise (-10dB SNR)');

%---
% 2
%---
% a
w0 = pi/4;
M = 32;
k = [1:M-1]';
r = [1;exp(j*w0.*k)];
R = toeplitz(conj(r),r);
[evecs,evals] = eig(R);
[tmp1,tmp2] = find(evals==max(max((evals))));
w = evecs(:,tmp1(1));
w = conj(w);  % beats me why this is needed to get spectral peak @ w0!!!

figure(1);
stem([0:M-1],real(w),'k');
axis([0 M-1 -0.2 0.2]);
title('Eigenfilter');
ylabel('Re[w(n)]');
xlabel('n');
grid;
figure(2);
stem([0:M-1],imag(w),'k')
axis([0 M-1 -0.2 0.2]);
title('Eigenfilter');
ylabel('Im[w(n)]');
xlabel('n');
grid;
```

```
fr_plot(w,1,2*pi,0);

figure(5)
polezero(w,1)

% b
% Assume x is a signal from Prob. 1b.
y = filter(w,1,x);  % filter noisy sinusoid;
Sy = periodogram(y,L);
figure;
periodogram_plot(Sy);


%---
% 3
%---
a = [1 0.5]';

% a
fr_plot(1,a);
axis([0 pi -4 6])

% b
N = 16000;
SNR = 0;
u = AR_synthesizer(a,N,1);  % AR process
sigma_u2m = cov(u);
sigma_v2 = sigma_u2*10^(-SNR/10);
v = randn(N,1) .* sqrt(sigma_v2);
x = u + v;
Su = periodogram(u,L);
figure;
periodogram_plot(Su);
ylabel('Su(w)');
Sx = periodogram(x,L);
figure;
periodogram_plot(Sx);
ylabel('Sx(w)');

% c
M = 16;
r = correlation(u,u,M-1);
R = toeplitz(r);
[evecs,evals] = eig(R);
w = evecs(:,M);
% w = w ./ sum(w);  % set noise gain to 1

stem([0:M-1],w,'k');
axis([0 M-1 -0.35 0.35]);
title('Eigenfilter');
ylabel('w(n)');
xlabel('n');
grid;

fr_plot(w,1);

polezero(w,1)
```

```
% d
y = filter(w,1,x);  % filter noisy sinusoid;
Sy = periodogram(y,L);
figure;
periodogram_plot(Sy);
ylabel('Sy(w)');

%---
% 4
%---
[u,fs,bits] = wavread('deleon.wav');
u = u - mean(u);
u = u ./ sqrt(cov(u));  % normalize to unit variance (will distort on
playback)

% a
L= 1024;
randn('seed',0);
SNR = 0; % in dB
sigma_v2 = 10^(-SNR/10);
v = sqrt(sigma_v2)*randn(length(u),1);
x = u + v;  % speech plus noise at prescribed SNR
Su = periodogram(u,L);
figure;
periodogram_plot(Su);
ylabel('Su(w)');
Sx = periodogram(x,L);
figure;
periodogram_plot(Sx);
ylabel('Sx(w)');
x = x ./ max(abs(x));
sound(x);

% b
M = 16;
r = correlation(u,u,M-1);
R = toeplitz(r);
[evecs,evals] = eig(R);
w = evecs(:,M);
w = w ./ sum(w);  % set noise gain to 1
stem([0:M-1],w,'k');
% axis([0 M-1 -0.35 0.35]);
title('Eigenfilter');
ylabel('w(n)');
xlabel('n');
grid;

fr_plot(w,1);

% c
y = filter(w,1,x);
Sy = periodogram(y,L);
figure;
periodogram_plot(Sy);
ylabel('Sy(w)');
y = y ./ max(abs(y));
```

```
sound(y);

%---
% 5
%---
[u,fs,bits] = wavread('deleon.wav');
u = u - mean(u);
u = u ./ sqrt(cov(u));  % normalize to unit variance (will distort on
playback)

h = [1 0.5 0.4 0.3]';
d = filter(h,1,u);  % filter speech to get a desired signal

% a
M = 3;
r = correlation(u,u,M);
R = toeplitz(r)
p = correlation(u,d,M)

% b
w = inv(R)*p

% c
M = 2;
r = correlation(u,u,M);
R = toeplitz(r);
p = correlation(u,d,M);
w = inv(R)*p;

% d
M = 4;
r = correlation(u,u,M);
R = toeplitz(r);
p = correlation(u,d,M);
w = inv(R)*p;
```