Homework 6

Solutions

6.1

consider

$$x_1[n] = \begin{cases} 1 & 0 \le n \le 99 \\ 0 & \text{else} \end{cases}$$

$$x_2[n] = \begin{cases} 1 & 0 \le n \le 9 \\ 0 & \text{else} \end{cases}$$

a, b, c -> see code

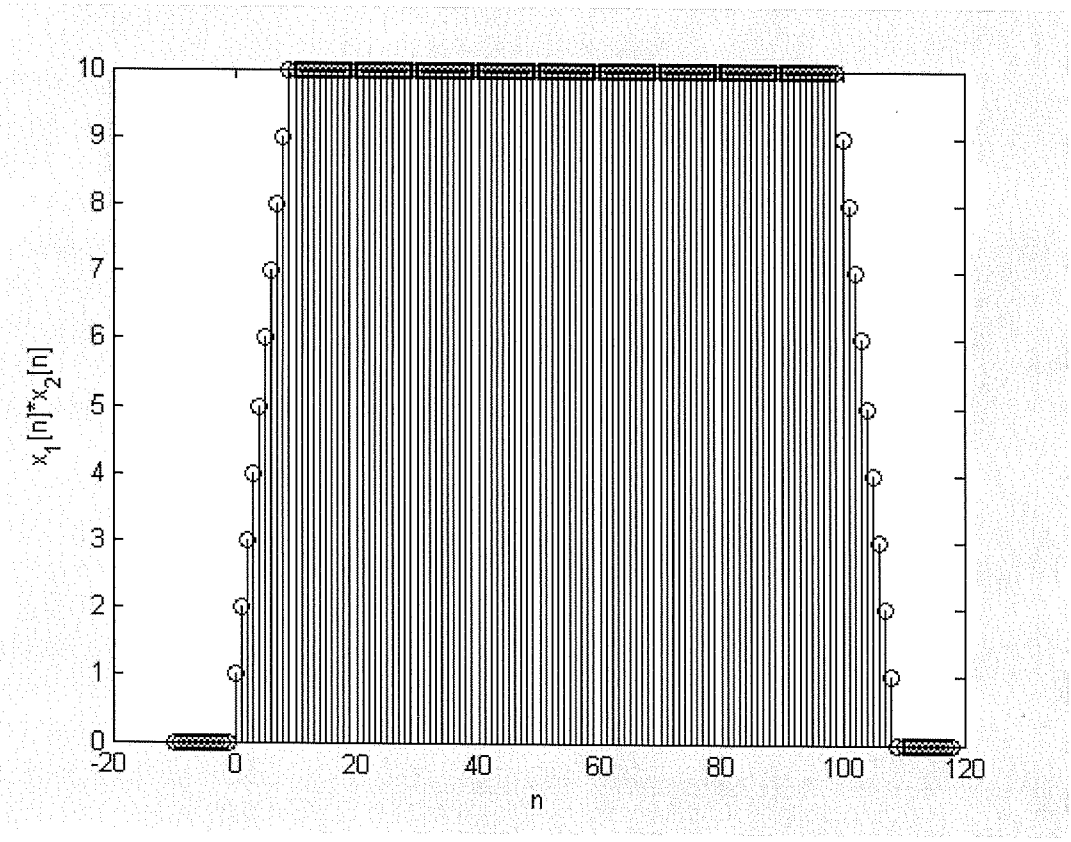
Contents

Homework 6 Problem 1

```
clear
clc
close all

x1 = ones(1,100);
x2 = ones(1,10);
n1 = [0:1:99];
n2 = [0:1:9];
```

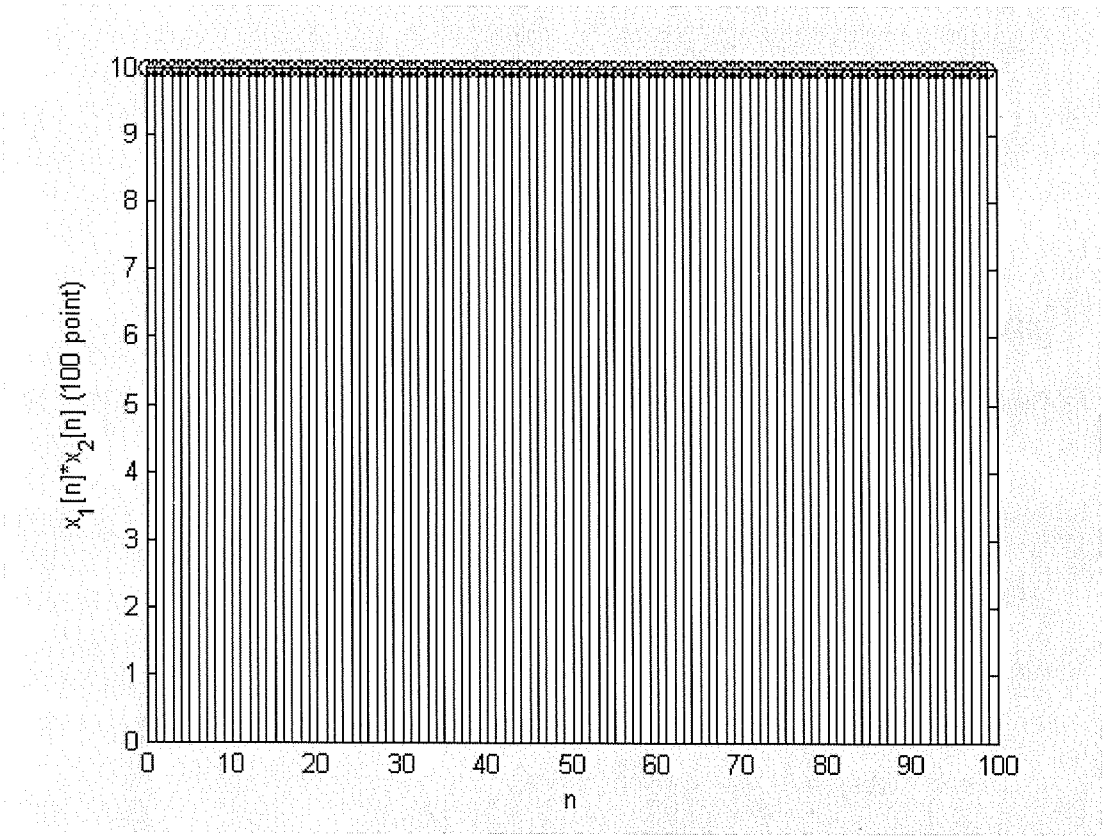
Problem 1a) - implement a linear convolution

```
pad = zeros(1,min(length(x1),length(x2)));
x1_pad = [pad x1 pad];
y = conv(x1_pad,x2);
n3 = [-length(pad):1:-length(pad)+length(y)-1];
stem(n3,y);
xlabel('n')
ylabel('x_1[n]*x_2[n]')
```



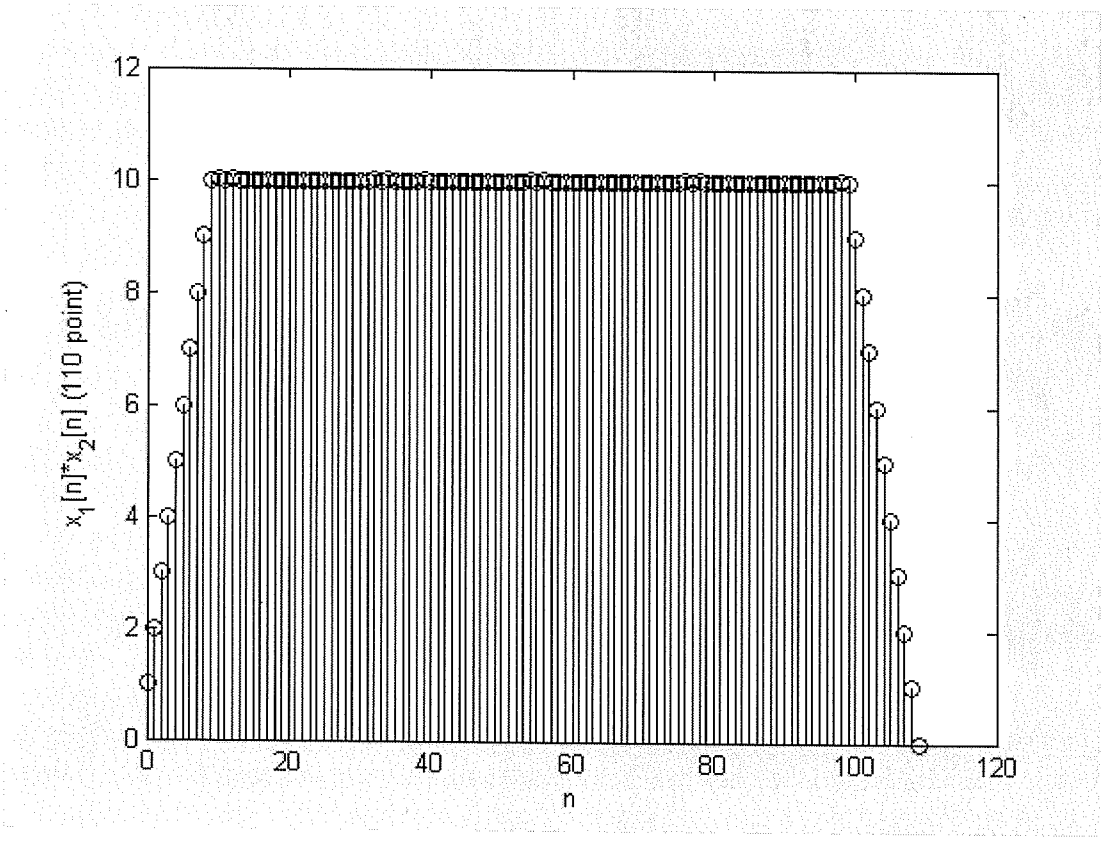
Problem 1b) -

```
X1 = fft(x1,100);
X2 = fft(x2,100);
figure
stem([0:1:99],ifft(X1.*X2))
xlabel('n')
ylabel('x_1[n]*x_2[n] (100 point)')
```



Problem 1c) -

```
X1 = fft(x1,110);
X2 = fft(x2,110);
figure
stem([0:1:109],ifft(X1.*X2))
xlabel('n')
ylabel('x_1[n]*x_2[n] (110 point)')
```



Published with MATLAB® 7.6

6.2

a) code

b) $K = 6$

c) $\Omega = \frac{2\pi K}{NT} = 2\pi f$

$$F = \frac{K}{NT} = \frac{6}{122} \cdot 4 = 0.197\,Hz$$

d) see code

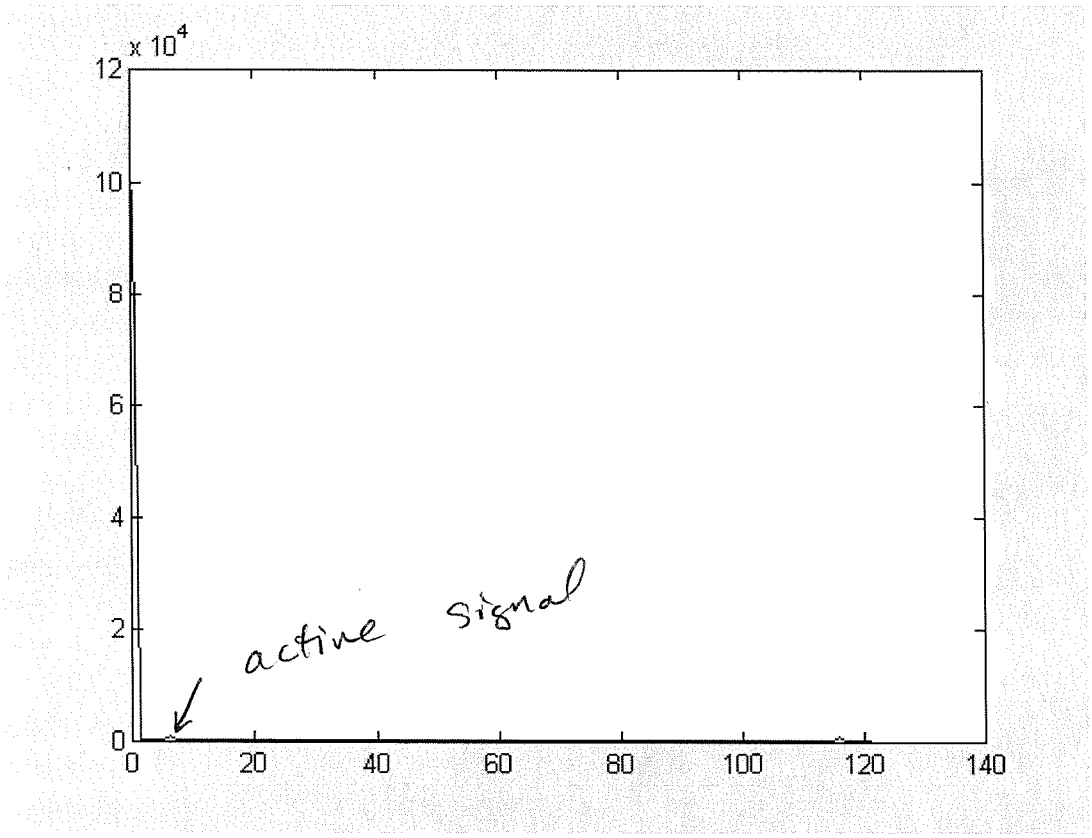
Contents

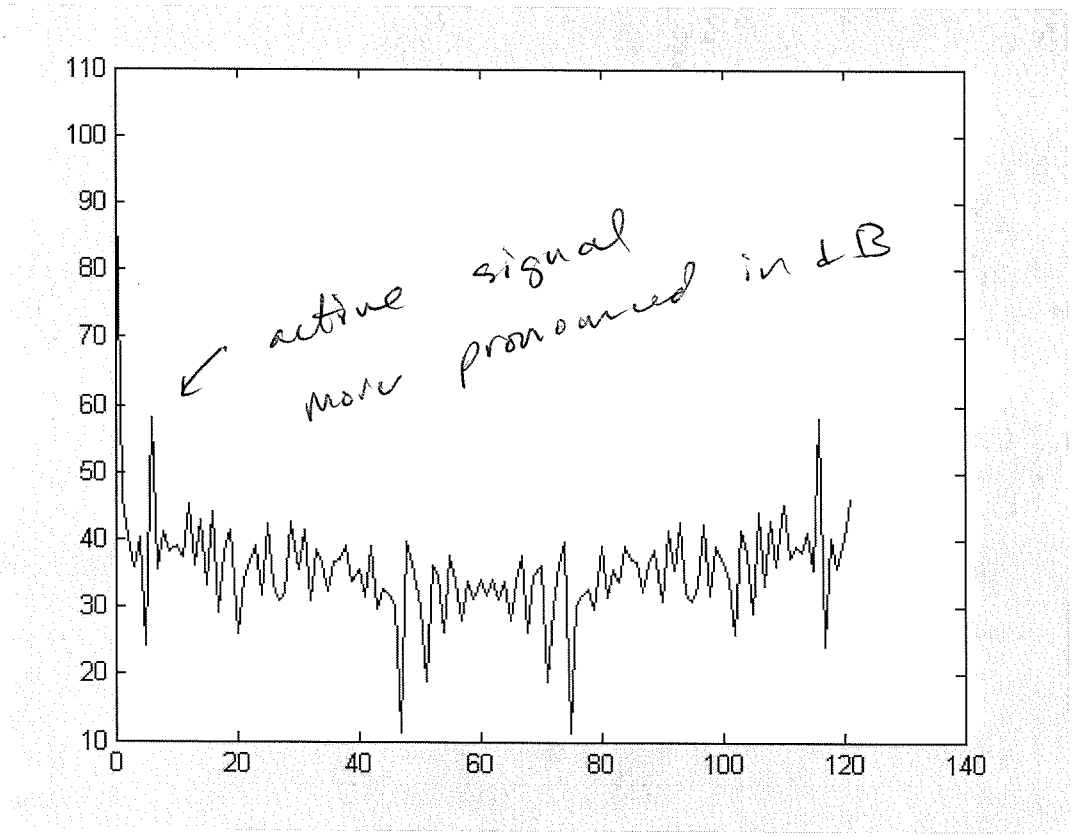
Homework 6 problem 2

```
clear
clc
close all

load fmrisig.mat
```

Problem 6.2a,b)

```
S = fft(s);
k = [0:1:length(S)-1];
plot(k,abs(S))


%plot fft(S) in dB since this will make strong component stand out (not
%required)
figure
S_dB = 20*log10(abs(S));
plot(k,S_dB)

% Dominant component at k = 6
```





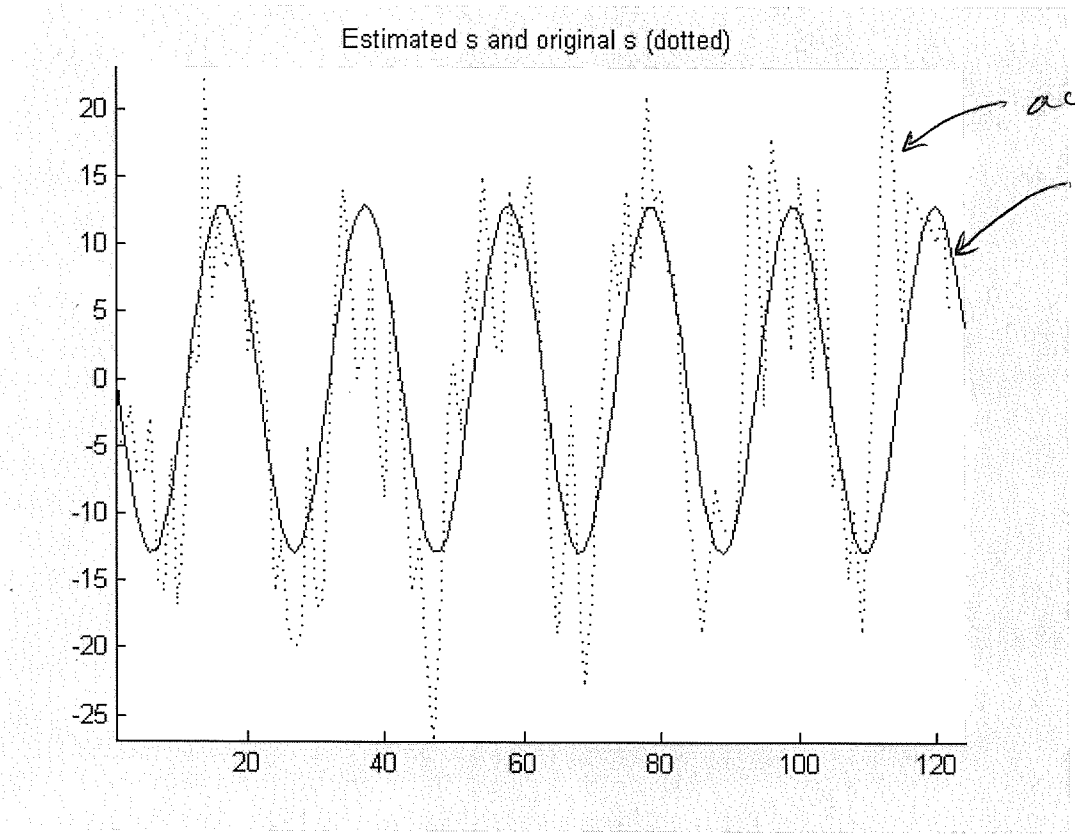
Problem 6.2c)

```
f = 6/122*4
```

```
f =

    0.1967
```

Problem 6.2d)

```
s_estimate = ifft([zeros(1,6) S(7) zeros(1,111) S(117)  zeros(1,5)]);
figure
hold on
plot(s_estimate)
%Remove DC component for plotting
plot(s-mean(s),':')
axis tight
title('Estimated s and original s (dotted)')
```



Published with MATLAB® 7.6

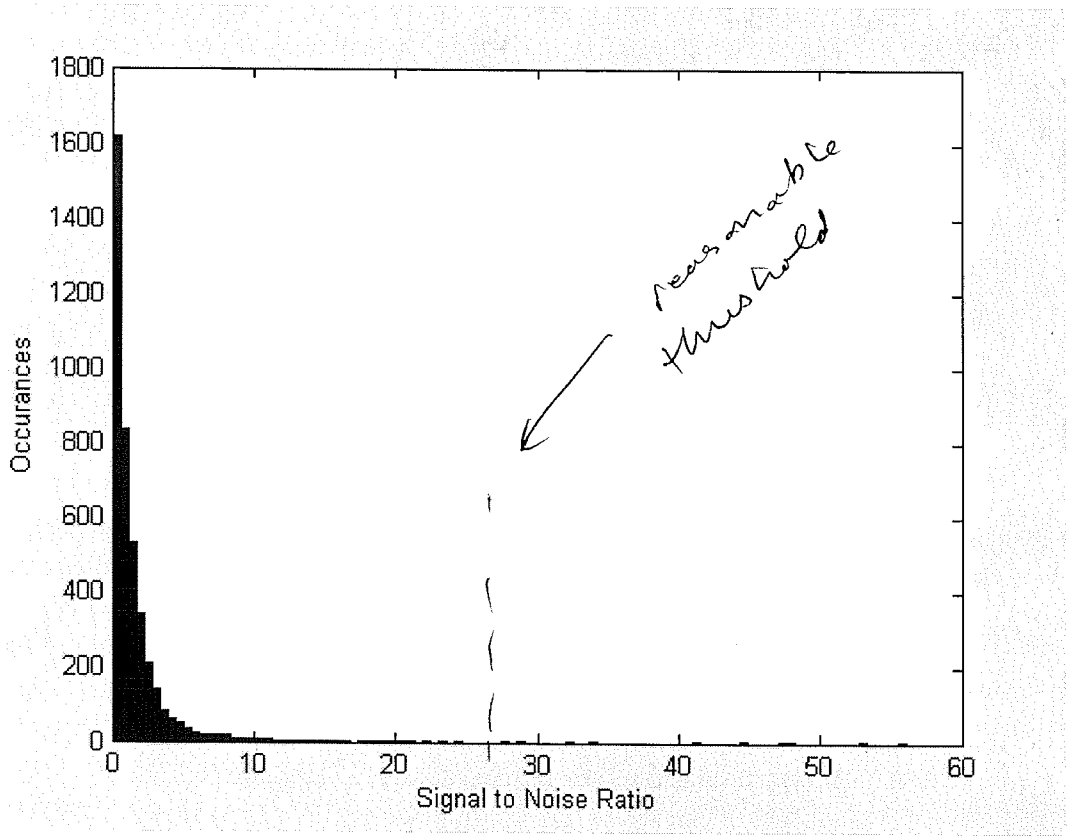
Contents

Homework 6 problem 3

```
clear
clc
close all

load fmri.mat
```

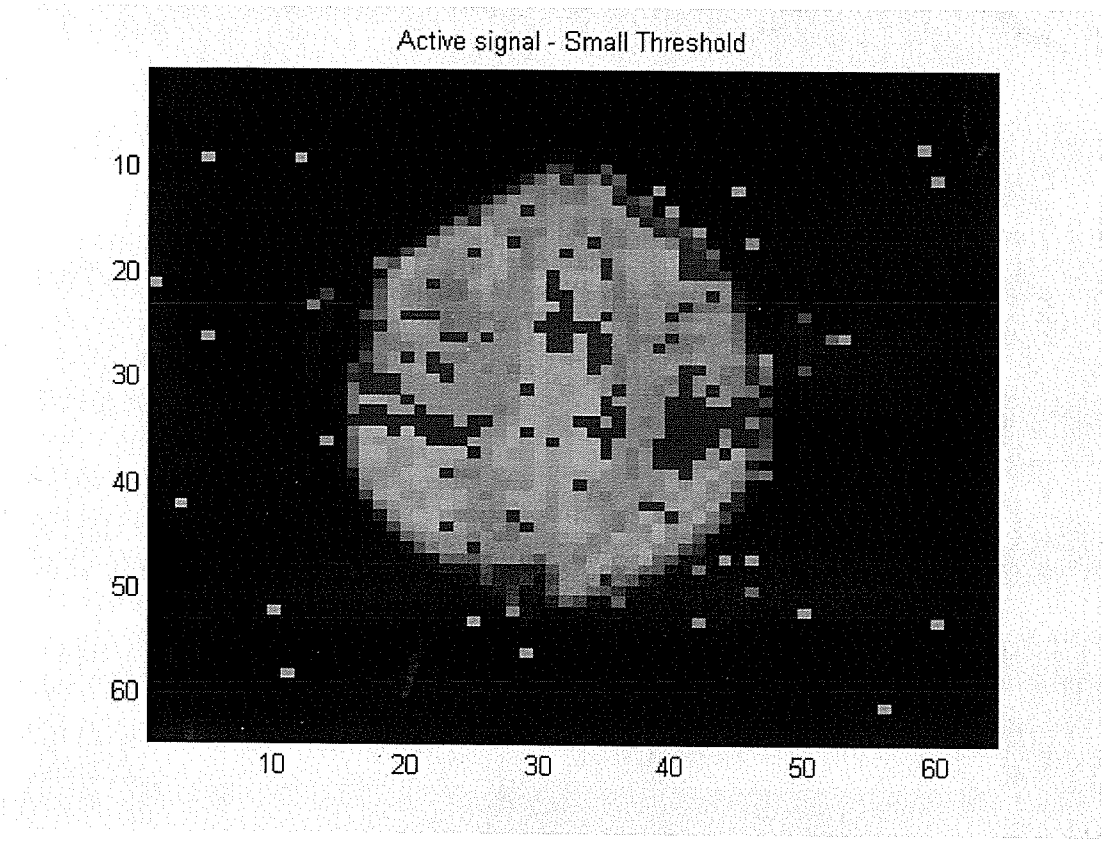
Problem 3 a

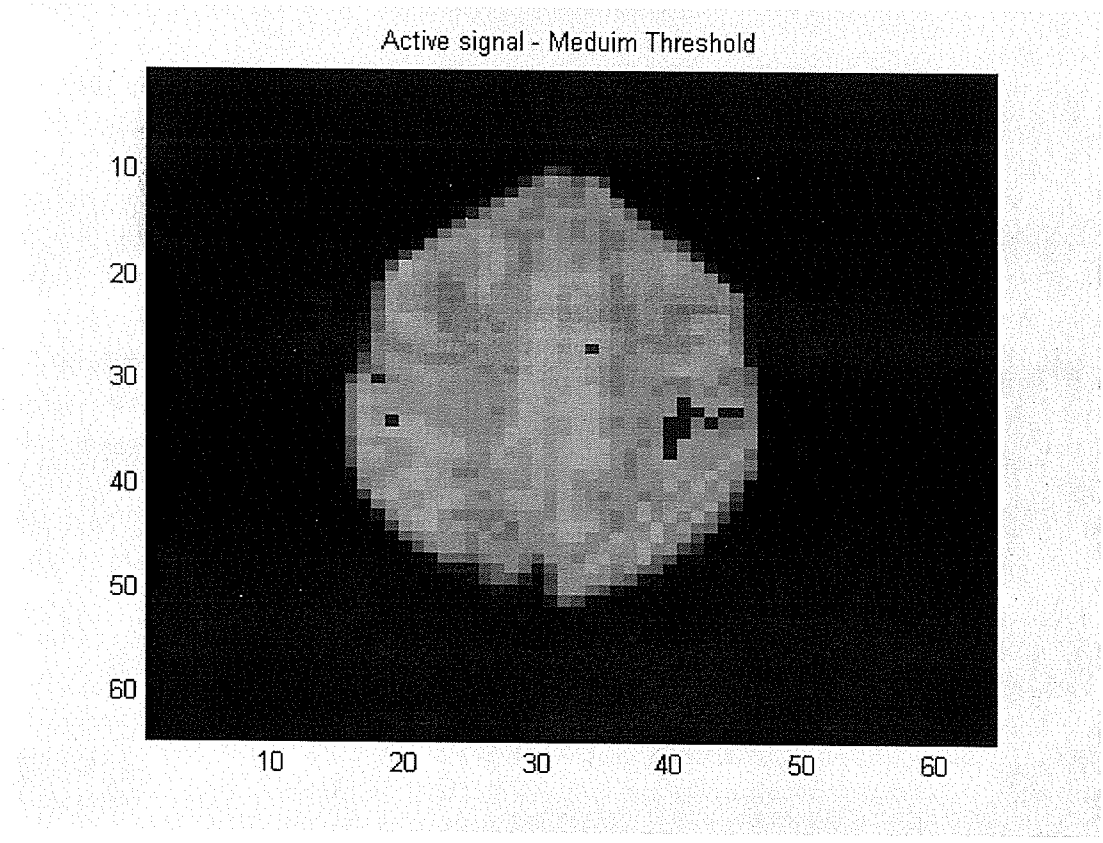
```
% we need to take DFT of each pixel as a function of time, and keep k = 7
% and k = 117 (from problem 2)

for i = 1:1:64
    for j = 1:1:64
        s = x(:,i,j);
        s = s-mean(s);
        S = fft(s);
        S_7(i,j) = abs(S(7))^2;
        %remove signal at k = 7 before finding noise
        S(7) = 0;
        noise(i,j) = mean(abs(S).^2);
    end
end

%Now we have noise power and signal power at each pixel

noise(noise==0) = eps;  %avoid divide by zero errors
SNR = S_7./noise;

% Make a histogram of SNRs to help determine threshold
figure
hist(reshape(SNR,1,[]),100)
xlabel('Signal to Noise Ratio')
ylabel('Occurances')

%From histogram, we see an appropriate threshold might be ~30
```

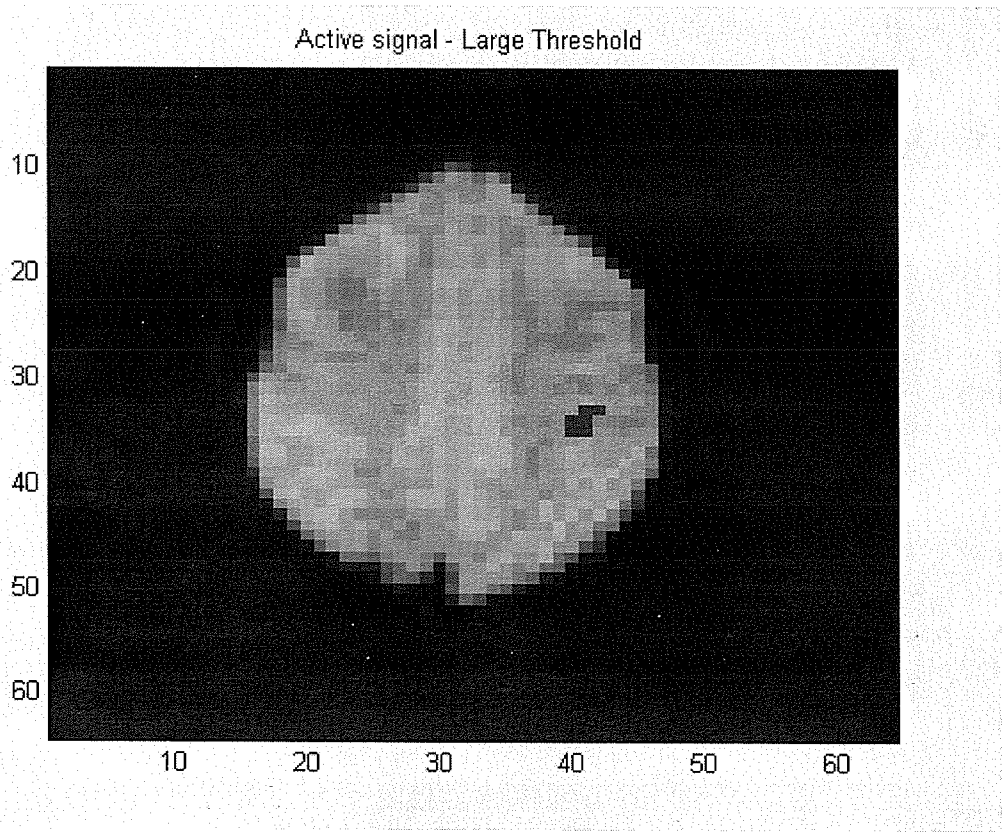


6.3 b - Small, medium and large thresholds

```
thres_s = 5;
thres_m = 20;
thres_l = 40;

active_s = SNR;
active_s(active_s<thres_s) = 0;
active_s(active_s>thres_s) = 1;

active_m = SNR;
active_m(active_m<thres_m) = 0;
active_m(active_m>thres_m) = 1;

active_l = SNR;
active_l(active_l<thres_l) = 0;
active_l(active_l>thres_l) = 1;


im = reshape(x(1,:,:),64,64);
figure
image(im/50 + active_s*50)
title('Active signal - Small Threshold')

figure
image(im/50 + active_m*50)
title('Active signal - Meduim Threshold')

figure
image(im/50 + active_l*50)
title('Active signal - Large Threshold')
```

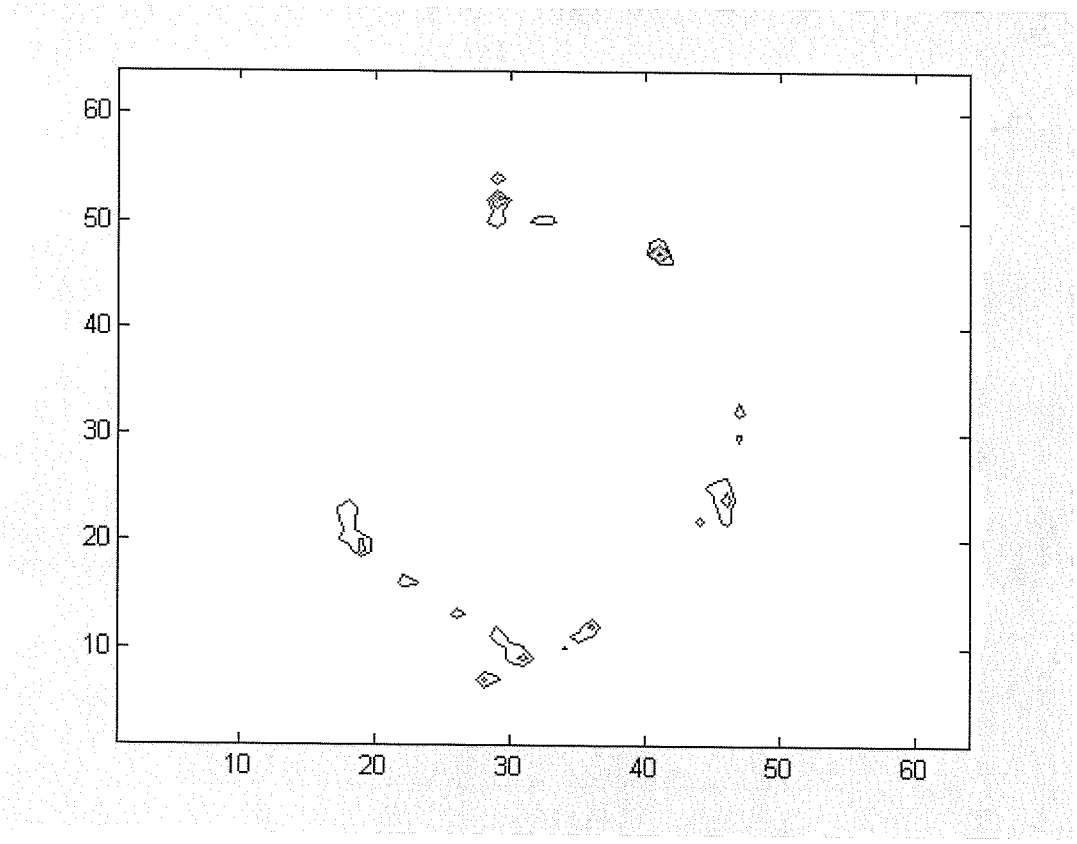

Active signal - Small Threshold
10
20
30
40
50
60
10
20
30
40
50
60


Active signal - Meduim Threshold
10
20
30
40
50
60
10
20
30
40
50
60



Problem 6 c

```
figure
contour(noise)
```



Published with MATLAB® 7.6

↑

Contour of Noise →

Clearly noise power depends on location.

We may get a 'stronger' signal + noise reading at some locations.

Therefore, it is better to look at signal to noise ratio at a given pixel.