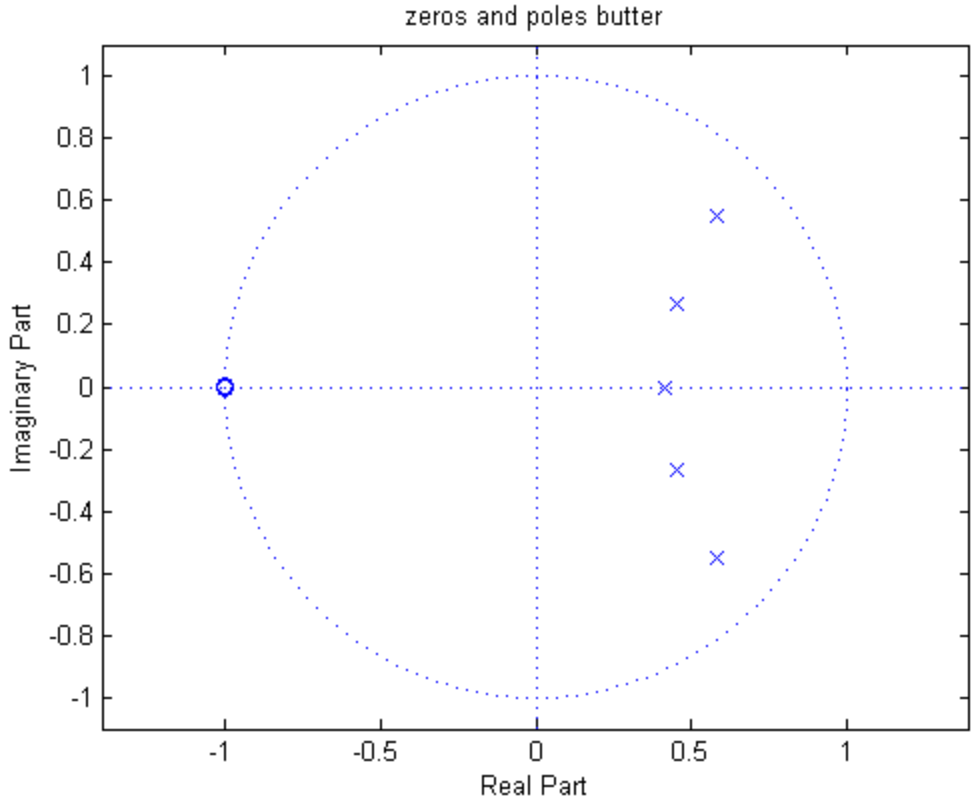
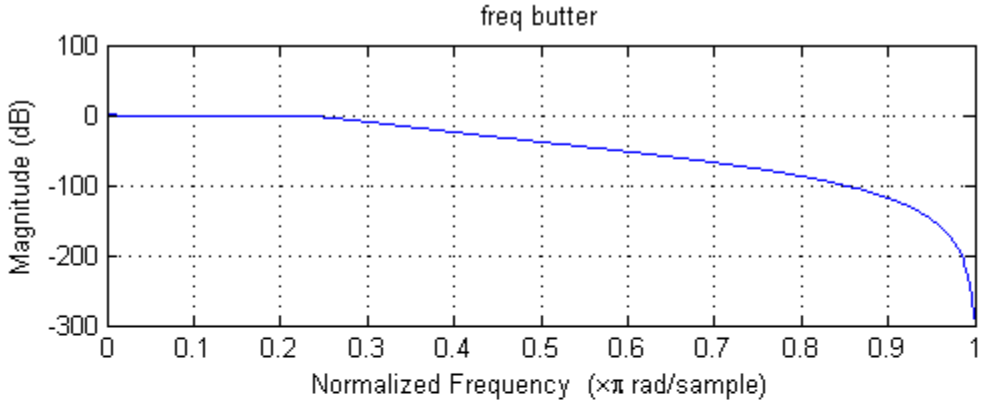
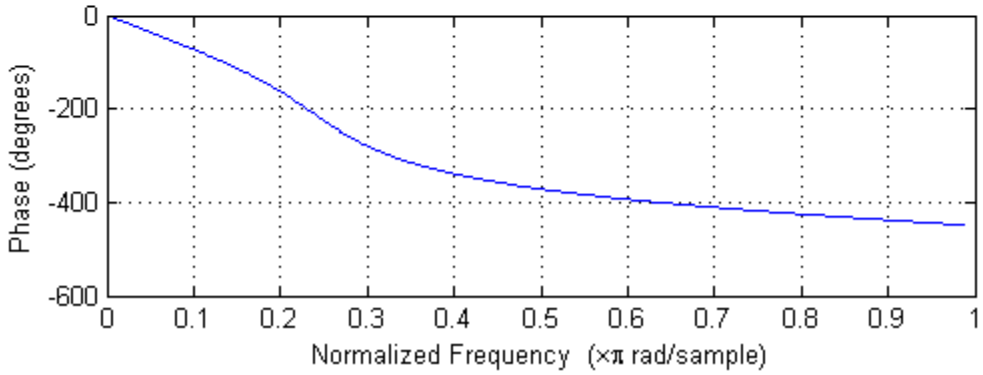
Contents

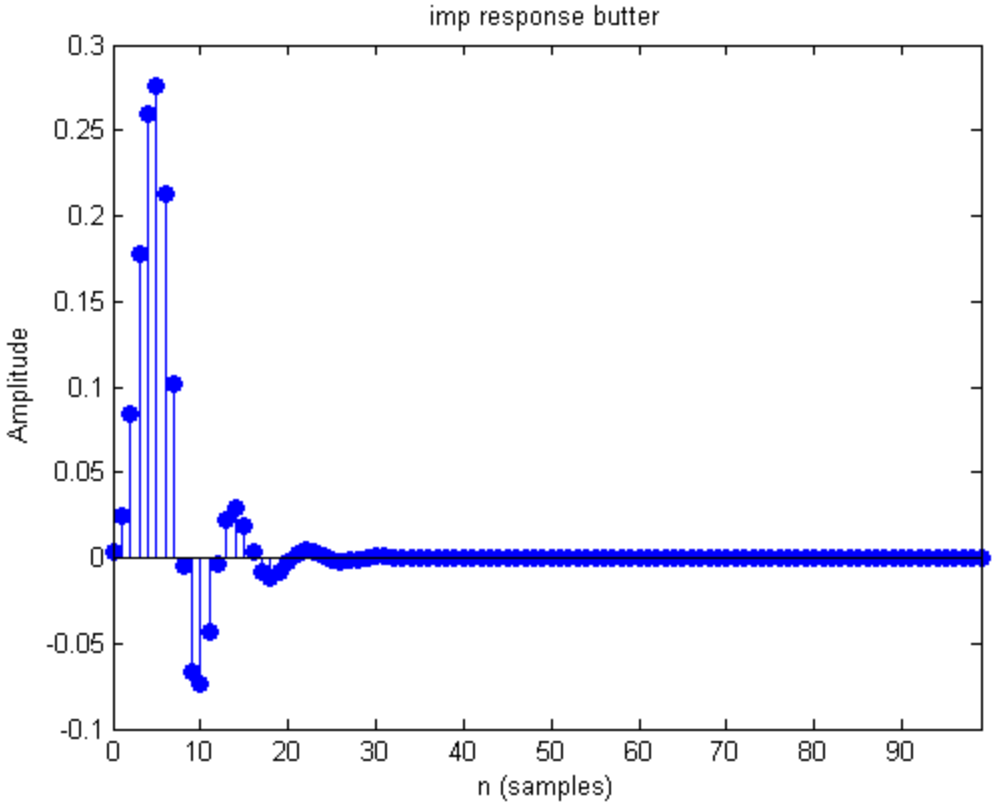
question 1

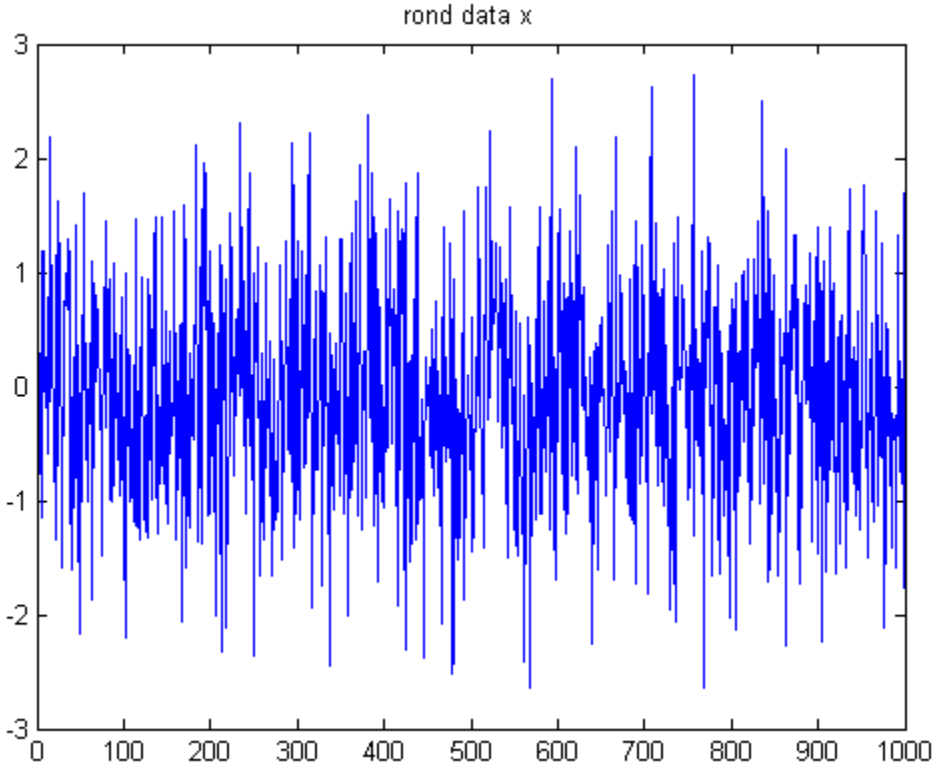
```
rand('seed',72);
n=wgn(1,1000,0);

wc = 1/4;
order = 5;
%%butter
[b,a] = butter(order, wc);

figure(1);
         zplane(b,a);
    title('zeros and poles butter')
figure(2);
         freqz(b,a);
    title('freq butter')

 %%imp response
 figure(3);
 impz(b,a,100)
 title('imp response butter')

 %%rondom butter
%%noise after fillter
butter_filter = filter(b,a,n);

figure(4);
plot(n)
title('rond data x')

x_filter_abs= abs(fftshift(fft(butter_filter)));
figure(5);
plot(x_filter_abs)
title('output from butter')
```

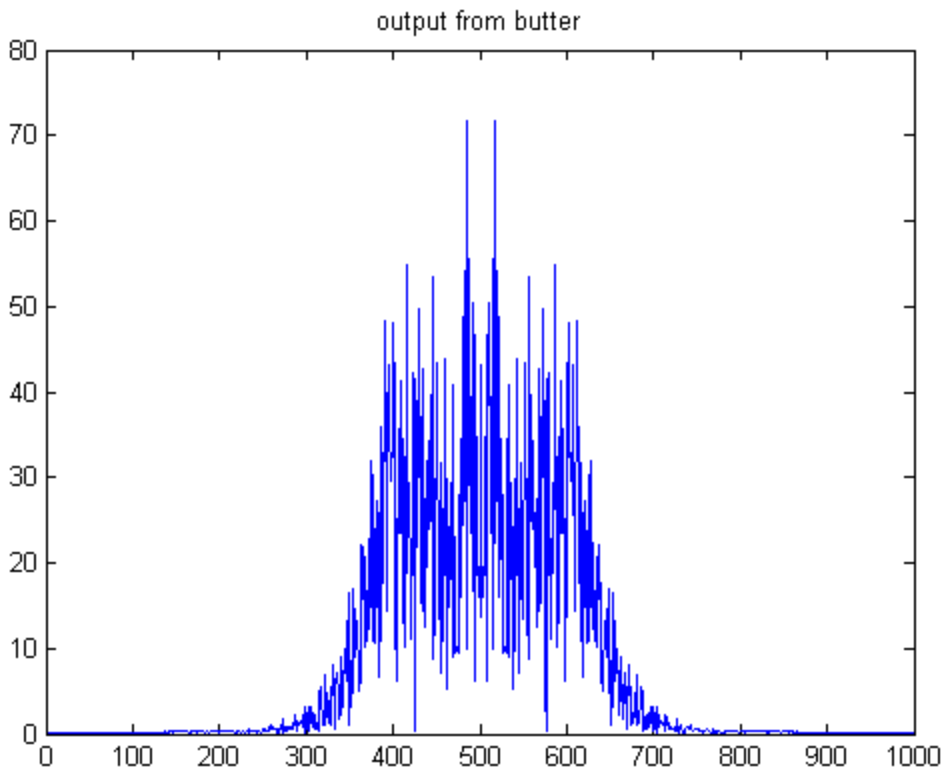

zeros and poles butter
Imaginary Part
Real Part
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.5
0
0.5
1


freq butter
Magnitude (dB)
Normalized Frequency (×π rad/sample)
100
0
-100
-200
-300
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Phase (degrees)
Normalized Frequency (×π rad/sample)
0
-200
-400
-600
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

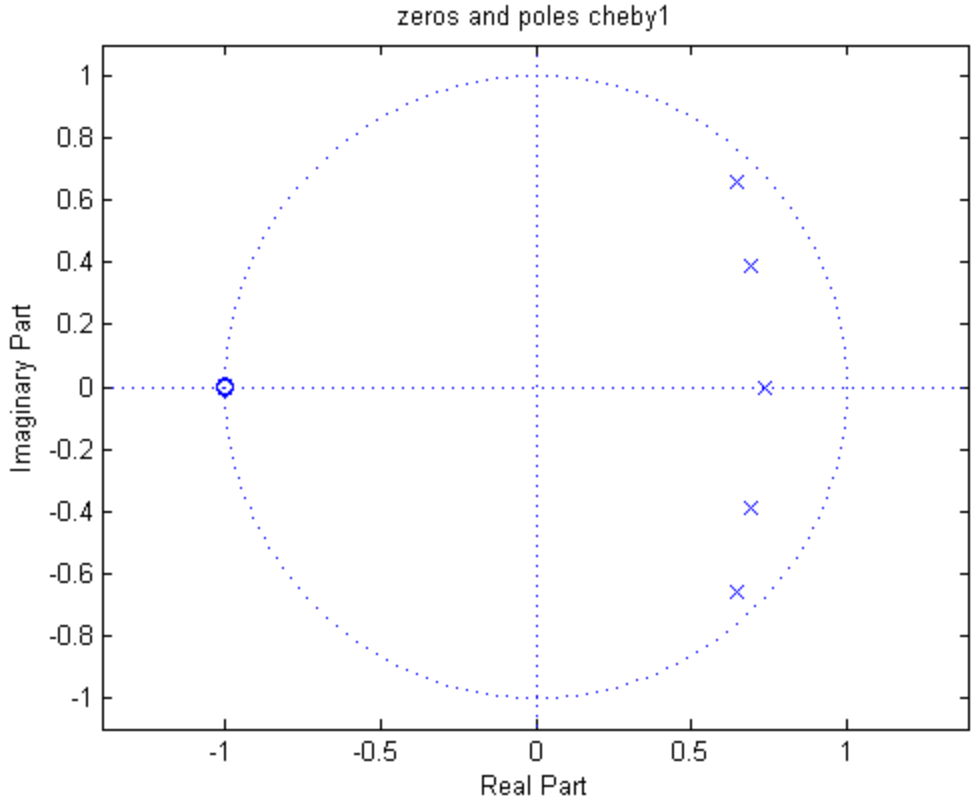

imp response butter
Amplitude
n (samples)
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
0
10
20
30
40
50
60
70
80
90

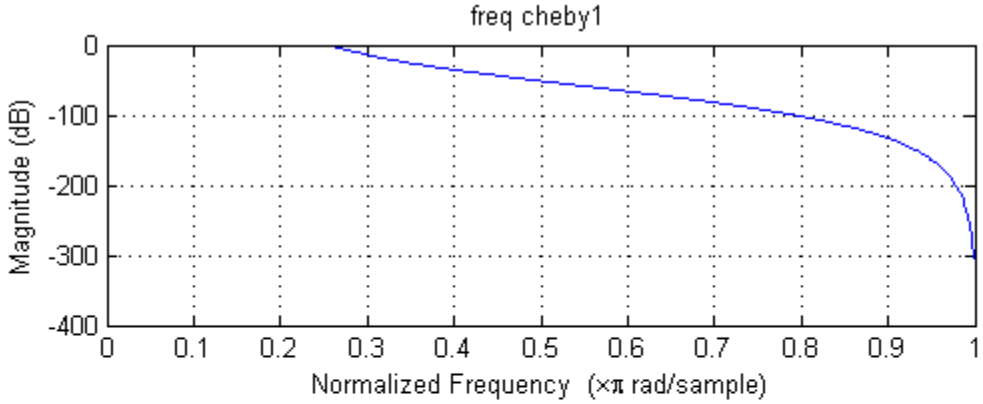

rond data x
3
2
1
0
-1
-2
-3
0
100
200
300
400
500
600
700
800
900
1000

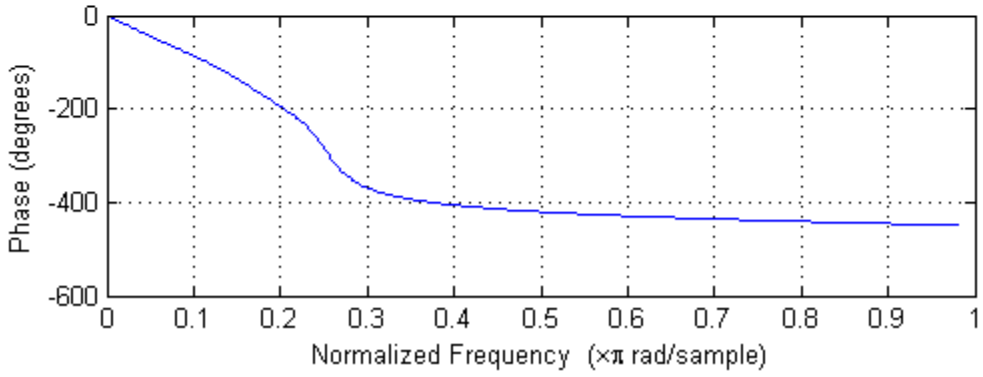


cheby1

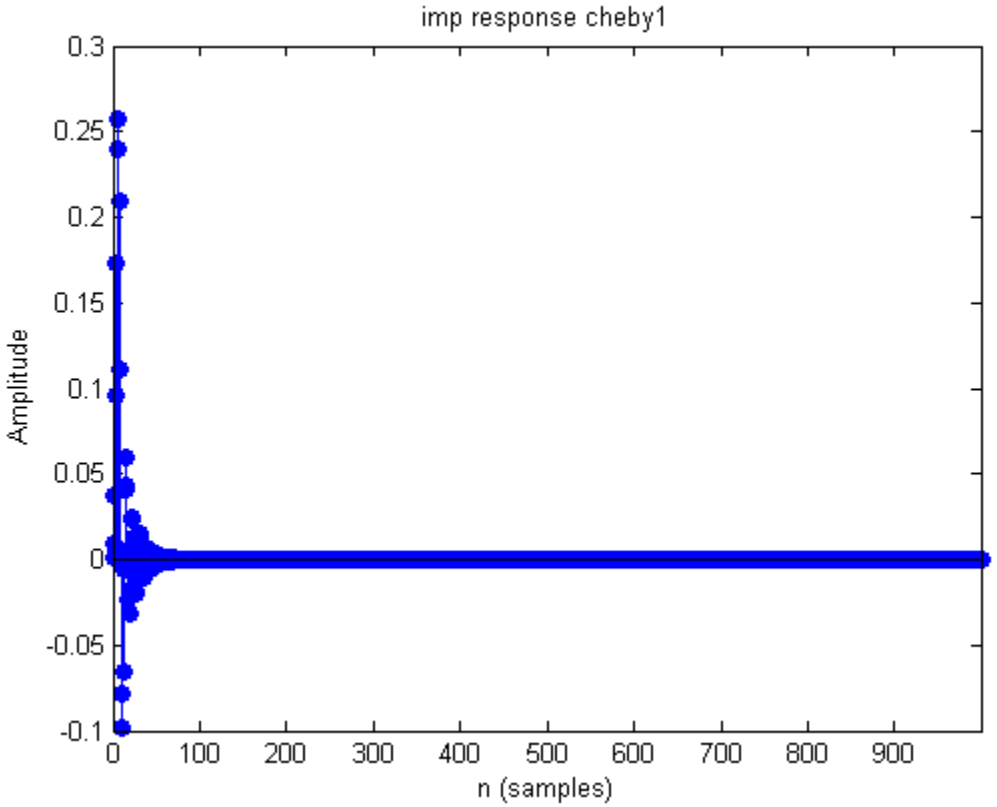
```
n=randn(1,10000);



wc = 1/4;

order = 5;



[z,p]=cheby1(order,0.5,wc);



figure(1);

         zplane(z,p);

    title('zeros and poles cheby1')
figure(2);
         freqz(z,p);
    title('freq cheby1')
```


zeros and poles cheby1
Imaginary Part
Real Part
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.5
0
0.5
1

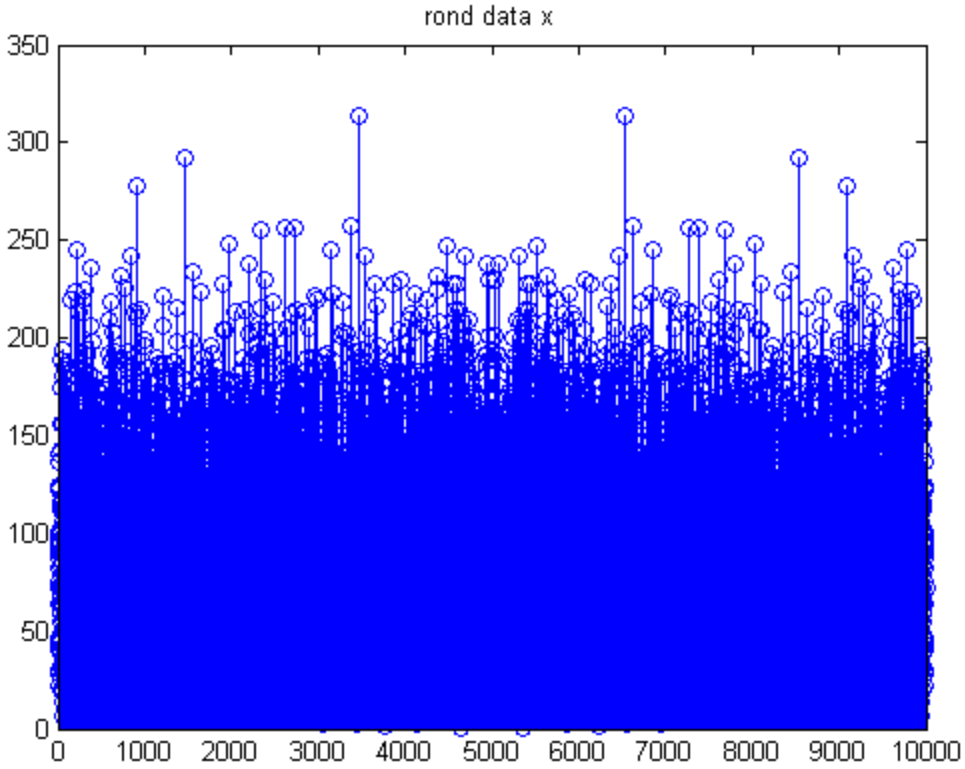

freq cheby1
Magnitude (dB)
Normalized Frequency (×π rad/sample)
0
-100
-200
-300
-400
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

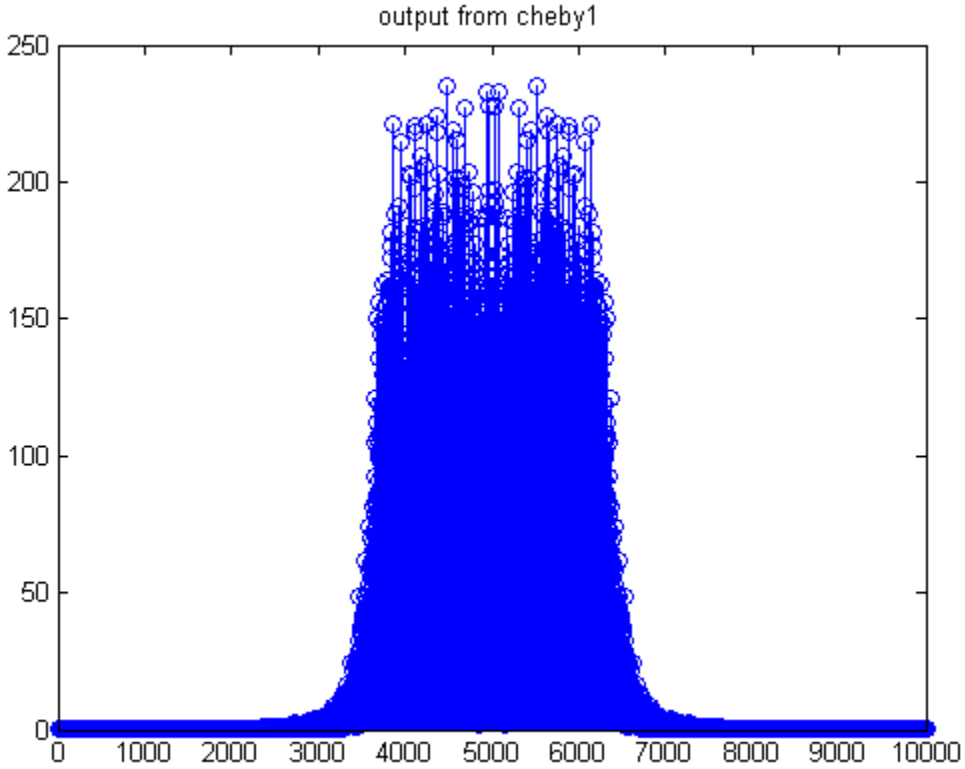

Phase (degrees)
Normalized Frequency (×π rad/sample)
0
-200
-400
-600
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

```
%%imp response
figure(3);
impz(z,p,1000)
title('imp response cheby1')
```



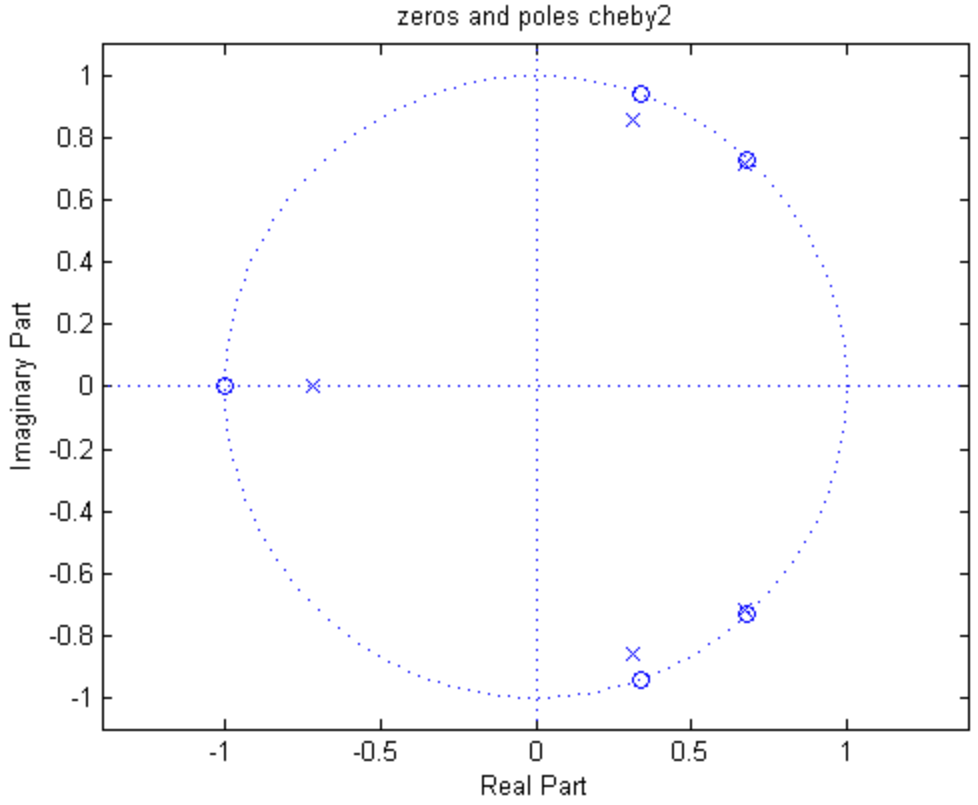

```
 %%rondom cheby1
%%noise after fillter
cheby1_filter = filter(z,p,n);

figure(4);
stem(abs(fftshift(fft(n))))
title('rond data x')

x_filter_abs= abs(fftshift(fft(cheby1_filter)));

figure(5);
stem(x_filter_abs)
title('output from cheby1')
```

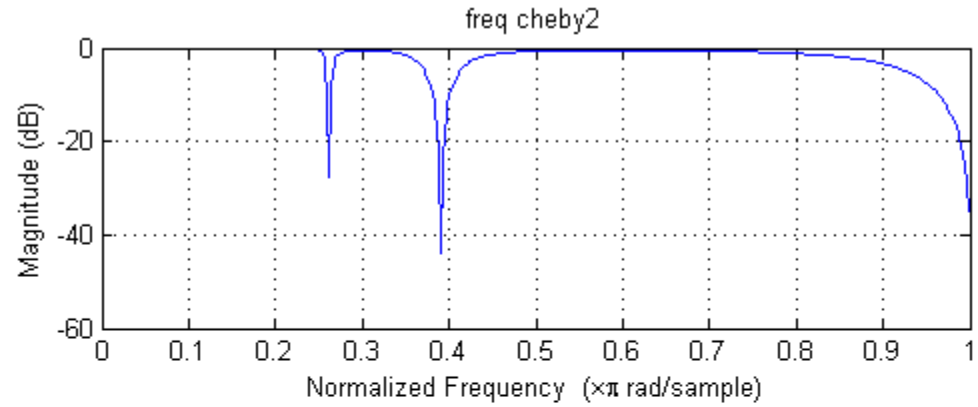

rond data x
350
300
250
200
150
100
50
0
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000

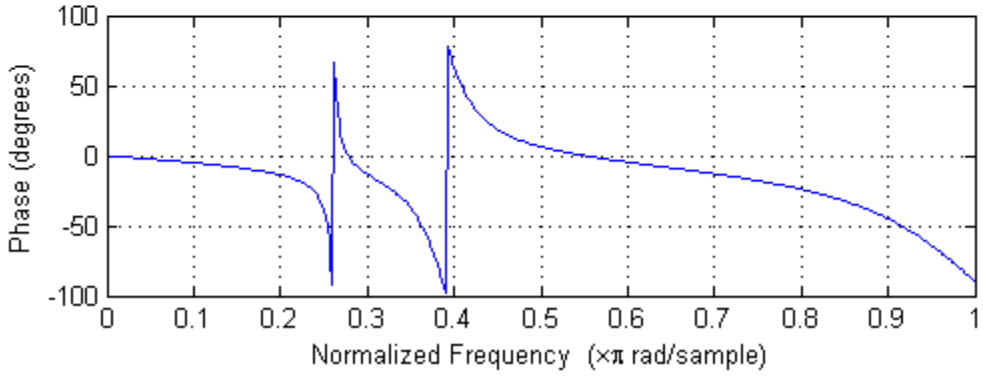

output from cheby1
250
200
150
100
50
0
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000

cheby2

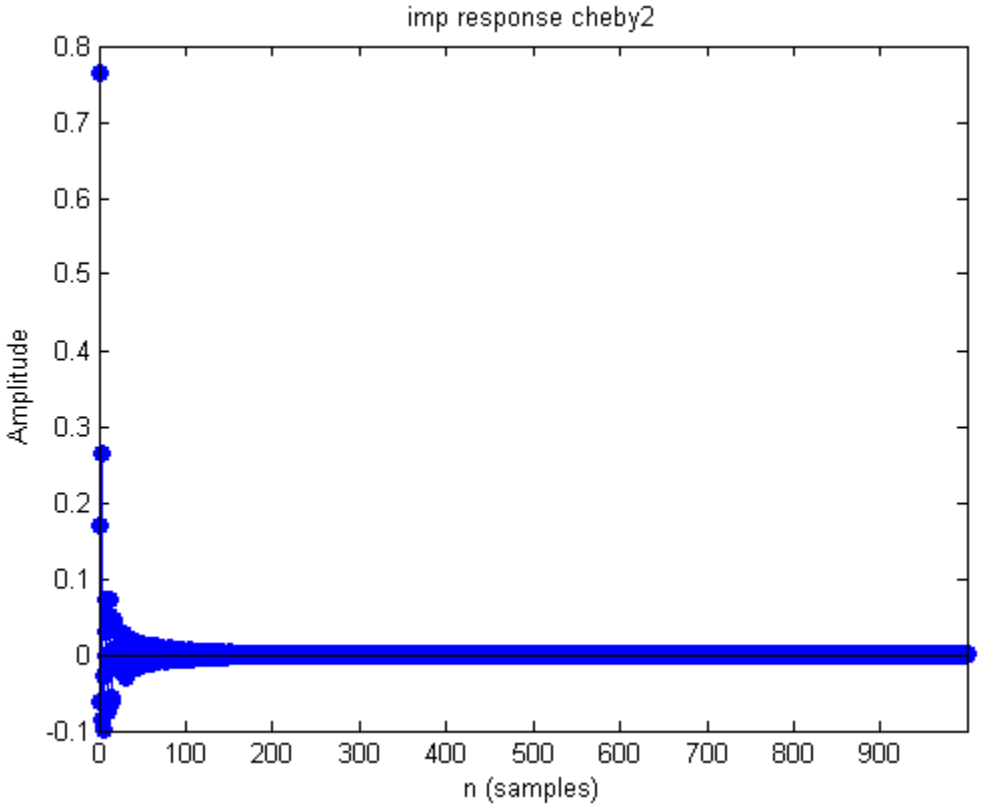
```
n=randn(1,1000);



wc = 1/4;

order = 5;



[z,p]=cheby2(order,0.5,wc);



figure(1);

         zplane(z,p);

    title('zeros and poles cheby2')
figure(2);
         freqz(z,p);
    title('freq cheby2')
```


zeros and poles cheby2
Imaginary Part
Real Part
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.5
0
0.5
1

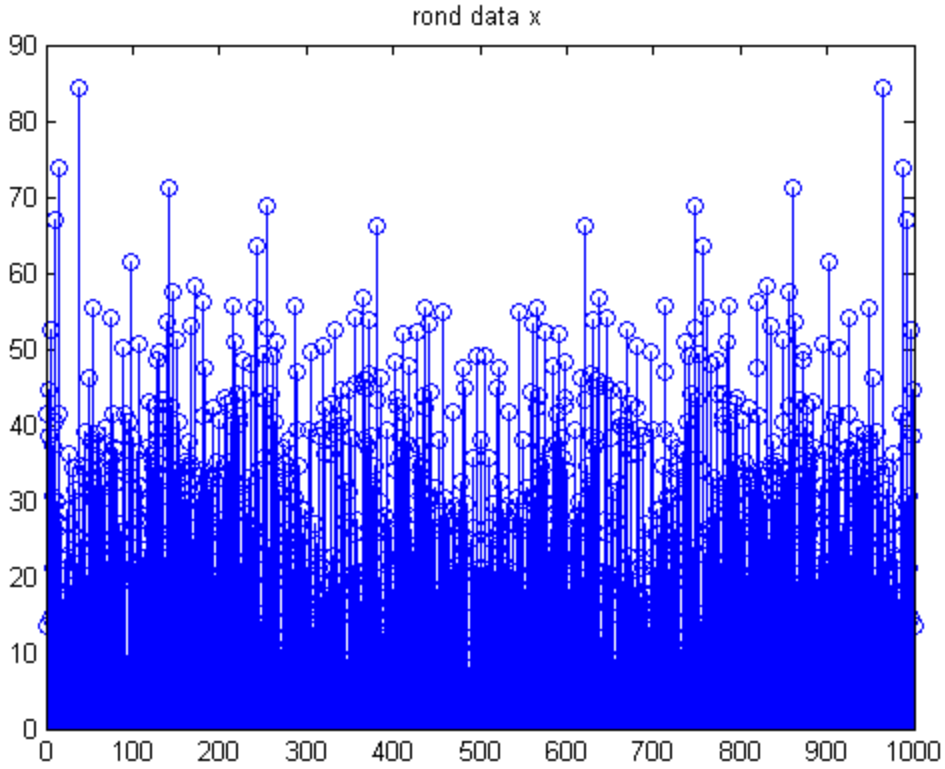

freq cheby2
Magnitude (dB)
Normalized Frequency (×π rad/sample)
0
-20
-40
-60
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

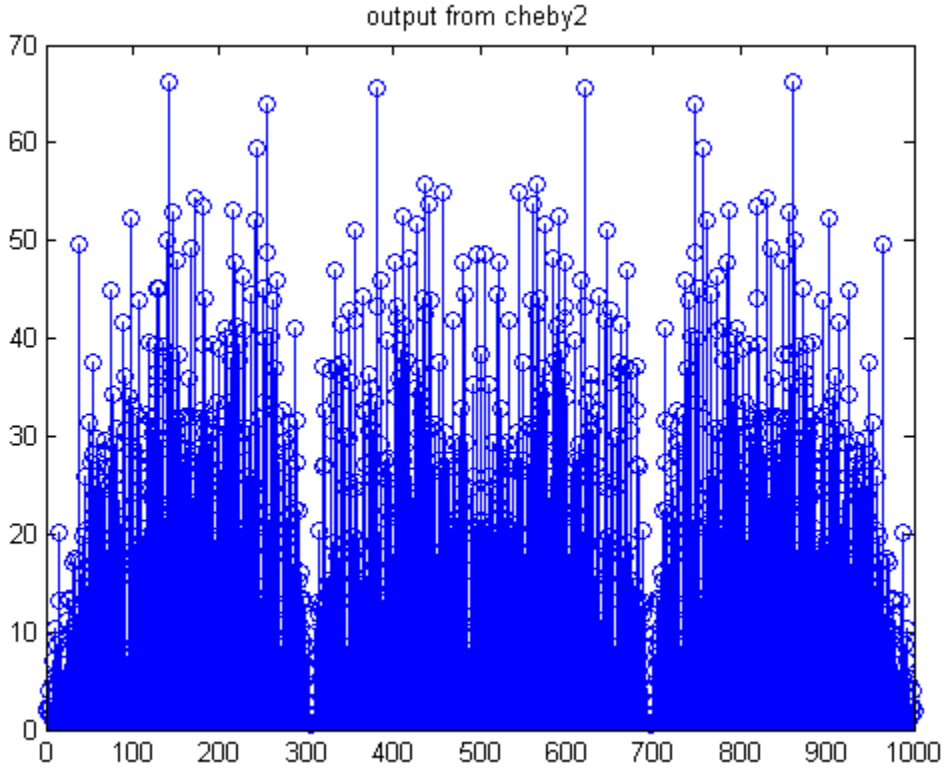

Phase (degrees)
Normalized Frequency (×π rad/sample)
100
50
0
-50
-100
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

```
%%imp response
figure(3);
impz(z,p,1000)
title('imp response cheby2')
```



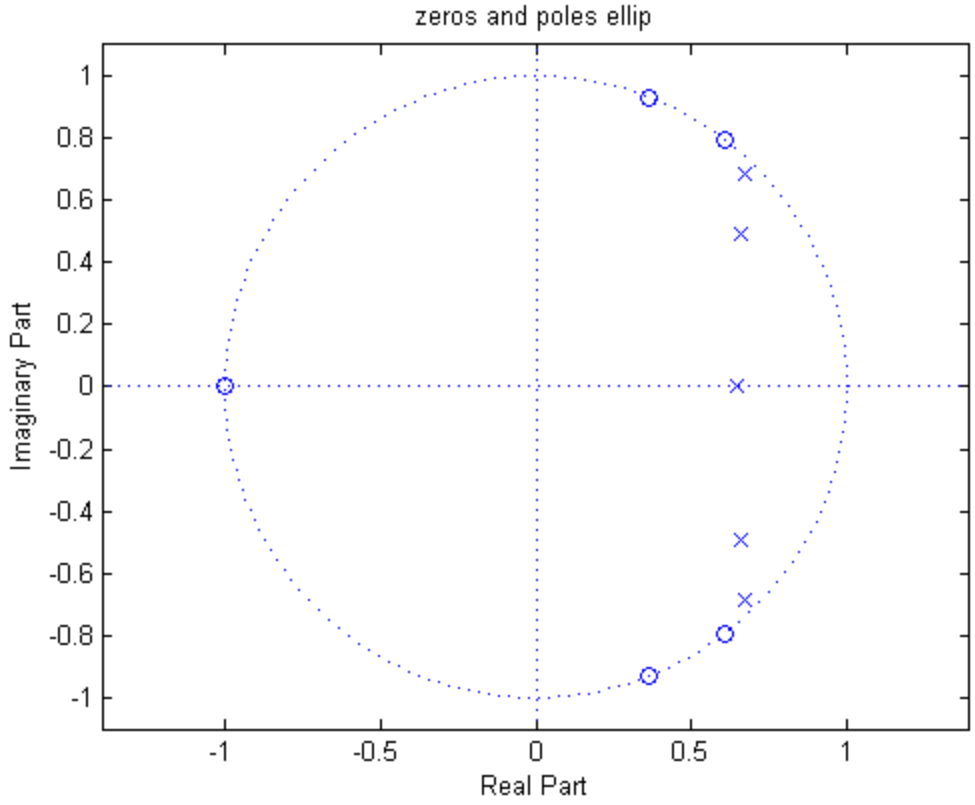

```
 %%rondom cheby2
%%noise after fillter
cheby2_filter = filter(z,p,n);

figure(4);
stem(abs(fftshift(fft(n))))
title('rond data x')

x_filter_abs= abs(fftshift(fft(cheby2_filter)));

figure(5);
stem(x_filter_abs)
title('output from cheby2')
```

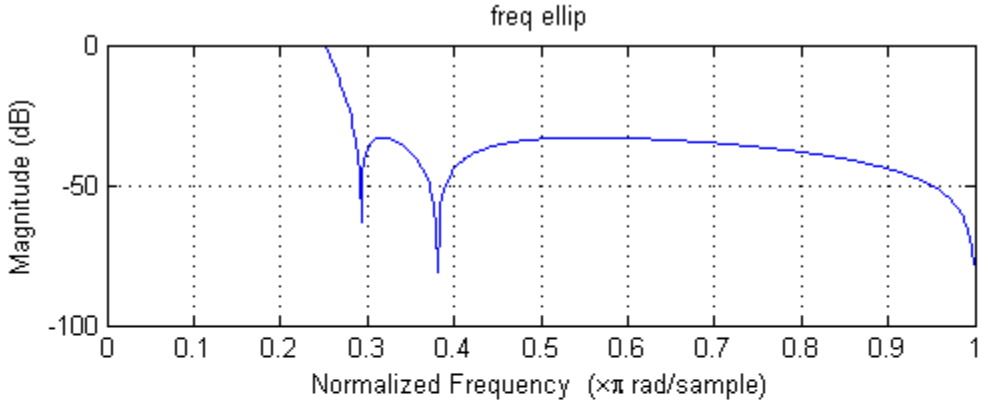

rond data x
90
80
70
60
50
40
30
20
10
0
0
100
200
300
400
500
600
700
800
900
1000

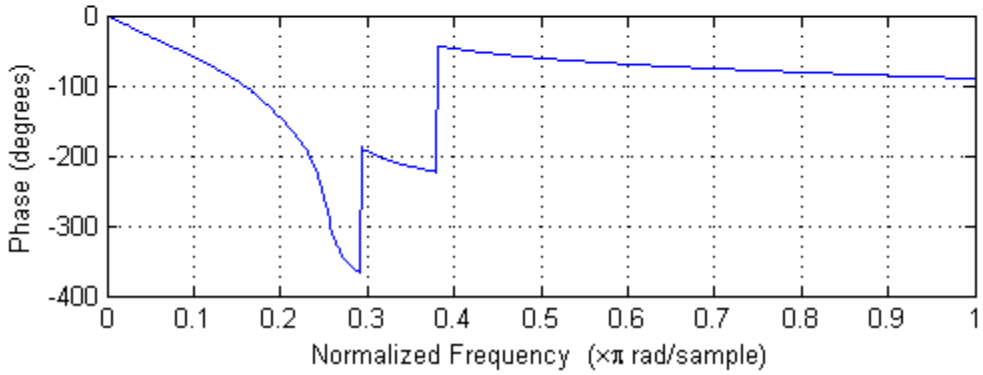

output from cheby2
70
60
50
40
30
20
10
0
0
100
200
300
400
500
600
700
800
900
1000

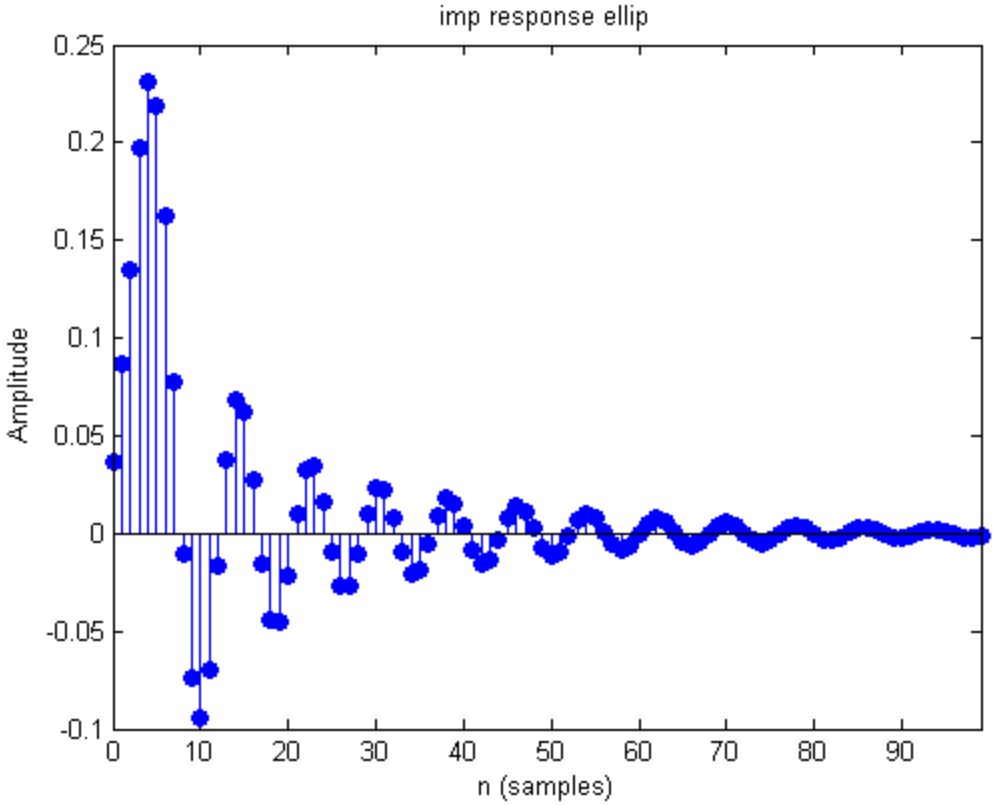
ellip

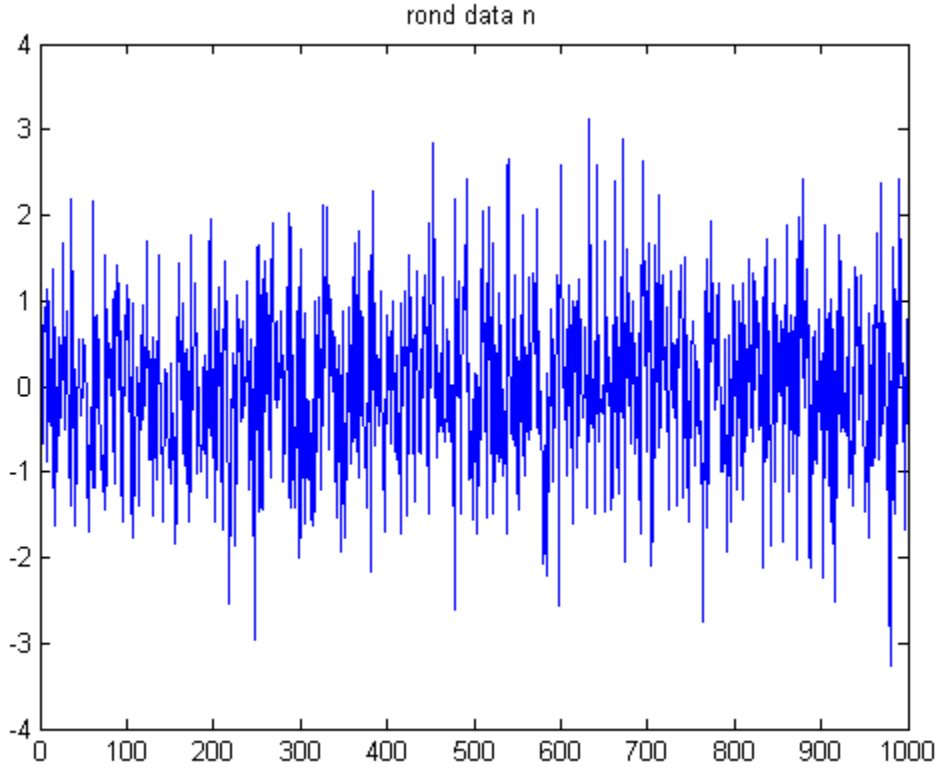
```
n=wgn(1,1000,0);



wc = 1/4;

order = 5;

[b,a] = ellip(order,0.5,33, wc);



figure(1);

         zplane(b,a);

    title('zeros and poles ellip')
figure(2);
         freqz(b,a);
    title('freq ellip')

 %%imp response
 figure(3);
 impz(b,a,100)
 title('imp response ellip')

 %%rondom ellip
%%noise after fillter
ellip_filter = filter(b,a,n);

figure(4);
plot(n)
title('rond data n')

x_filter_abs= abs(fftshift(fft(ellip_filter)));
figure(5);
plot(x_filter_abs)
title('output from ellip')
```

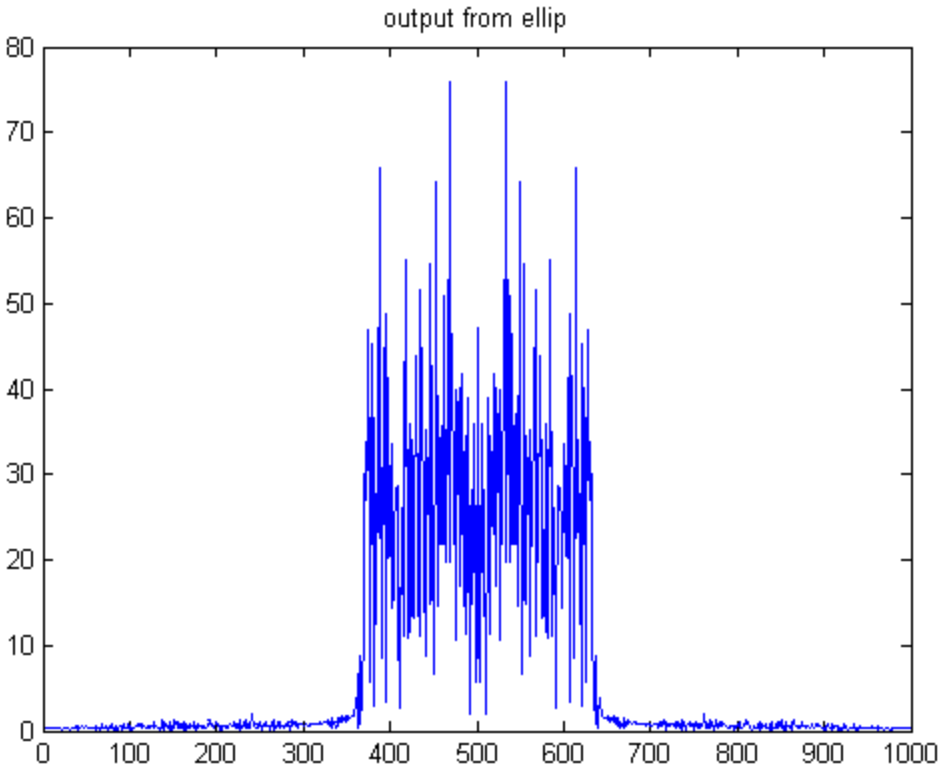

zeros and poles ellip
Imaginary Part
Real Part
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.5
0
0.5
1


freq ellip
Magnitude (dB)
Normalized Frequency (×π rad/sample)
0
-50
-100
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Phase (degrees)
Normalized Frequency (×π rad/sample)
0
-100
-200
-300
-400
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


imp response ellip
Amplitude
n (samples)
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
0
10
20
30
40
50
60
70
80
90

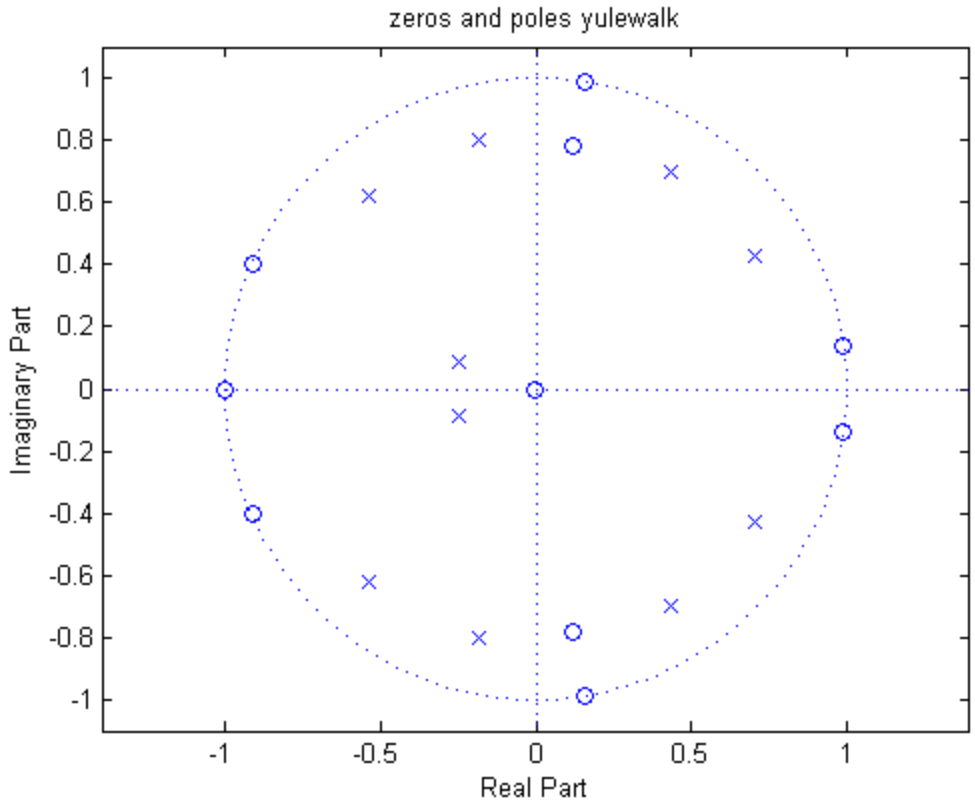

rond data n
4
3
2
1
0
-1
-2
-3
-4
0
100
200
300
400
500
600
700
800
900
1000

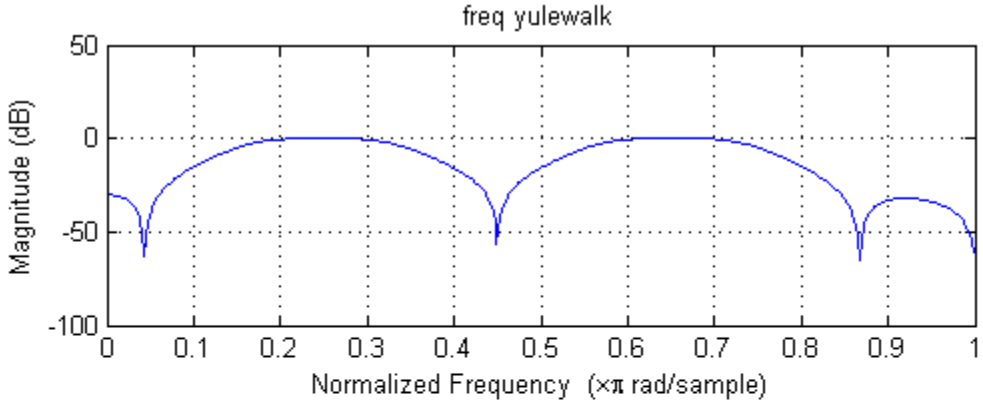


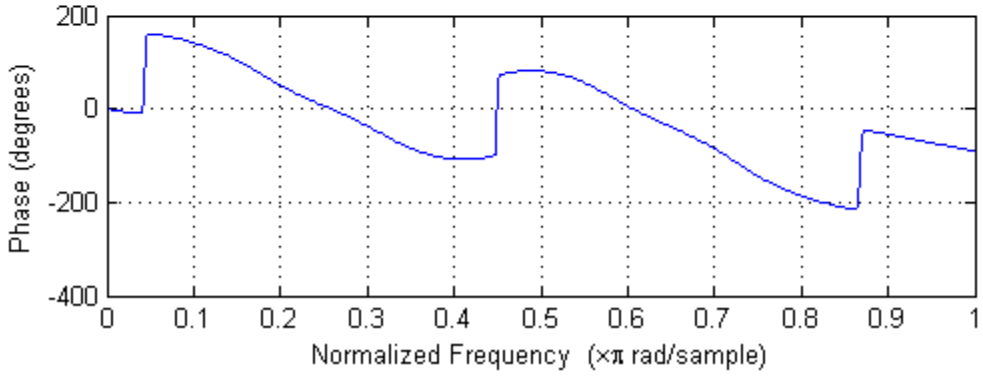
question2-yulewalk

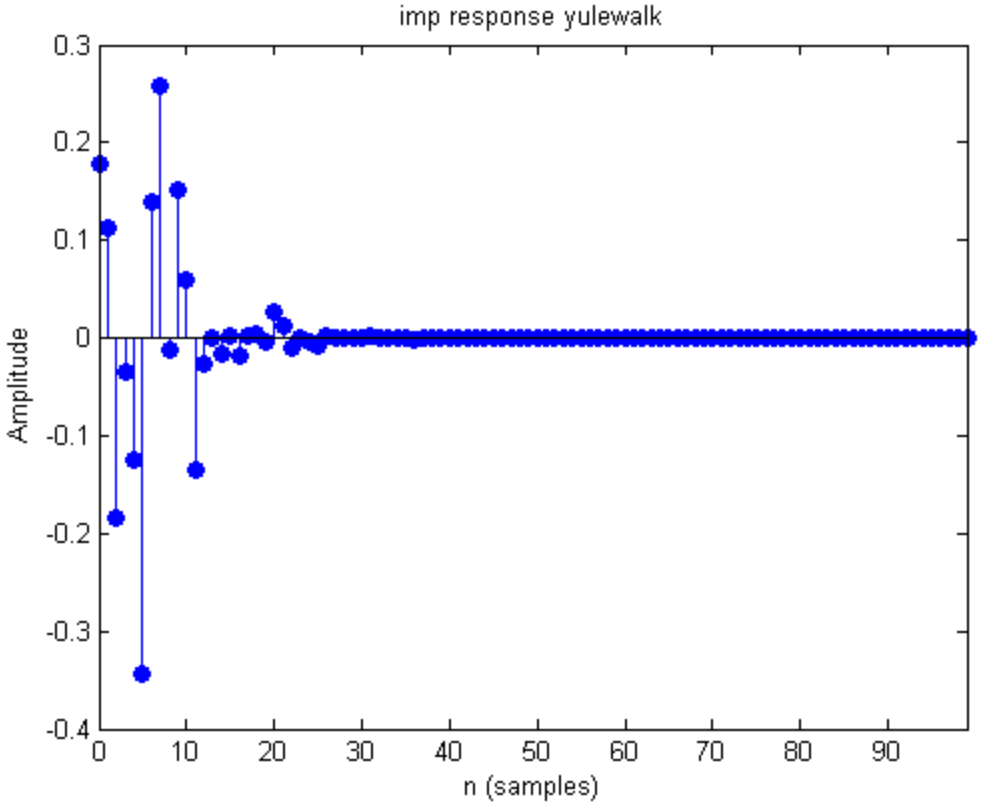
```
n=randn(1,100);



f = [0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 1];

m = [0 0 1 1 0 0 1 1 0 0];



[b,a] = yulewalk(10,f,m);



figure(1);

         zplane(b,a);

    title('zeros and poles yulewalk')
figure(2);
         freqz(b,a);
    title('freq yulewalk')

 %%imp response
 figure(3);
```

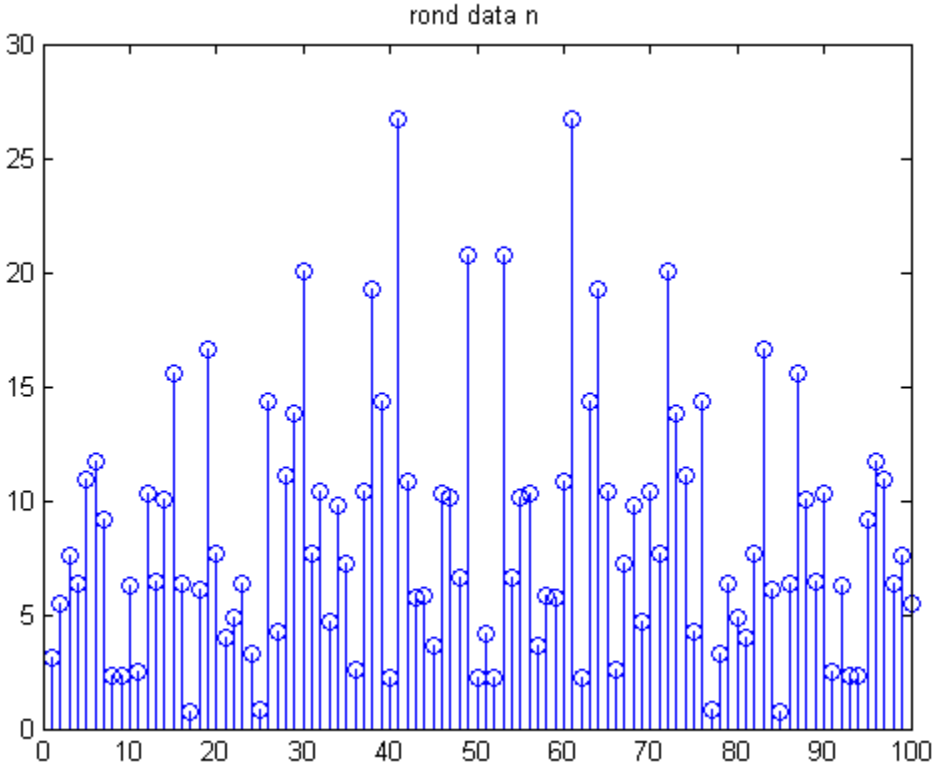
```
 impz(b,a,100)

 title('imp response yulewalk')

 %%rondom butter
%%noise after fillter
yulewalk_filter = filter(b,a,n);

figure(4);
stem(abs(fftshift(fft(n))))
title('rond data n')

x_filter_abs= abs(fftshift(fft(yulewalk_filter)));

figure(5);
stem(x_filter_abs)
title('output from yulewalk')
```

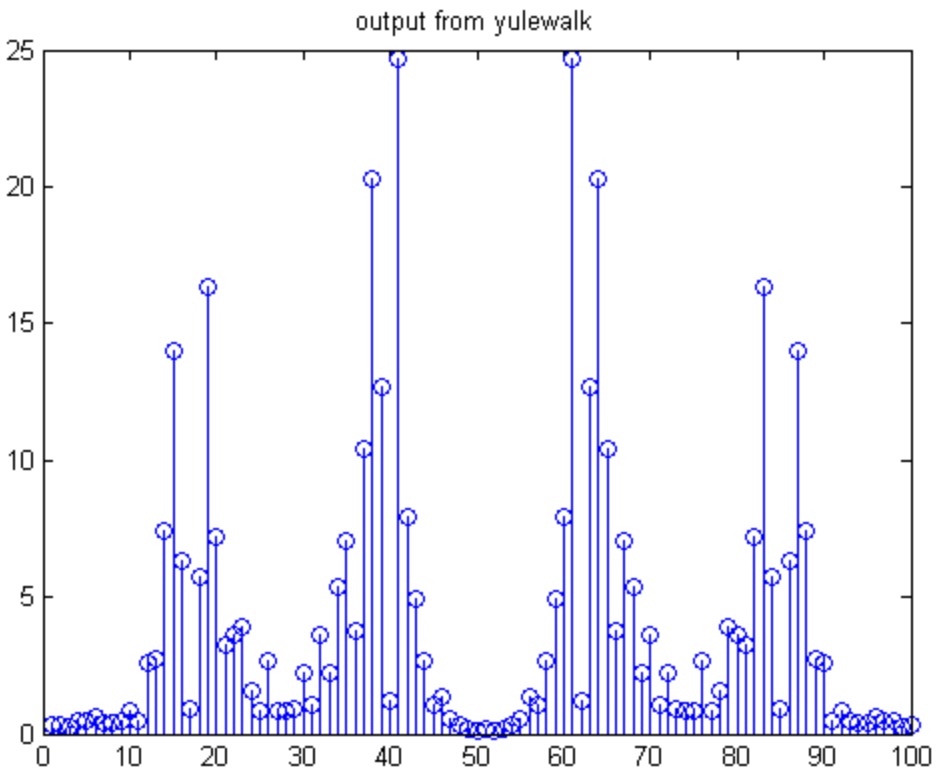

zeros and poles yulewalk
Imaginary Part
Real Part
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.5
0
0.5
1


freq yulewalk
Magnitude (dB)
Normalized Frequency (×π rad/sample)
50
0
-50
-100
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Phase (degrees)
Normalized Frequency (×π rad/sample)
200
0
-200
-400
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


imp response yulewalk
Amplitude
n (samples)
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
0
10
20
30
40
50
60
70
80
90


rond data n
30
25
20
15
10
5
0
0
10
20
30
40
50
60
70
80
90
100



question 3 - tone detection

Contents

step1

the number is : 0 1 0 9 5 0 1 7 7 6 3

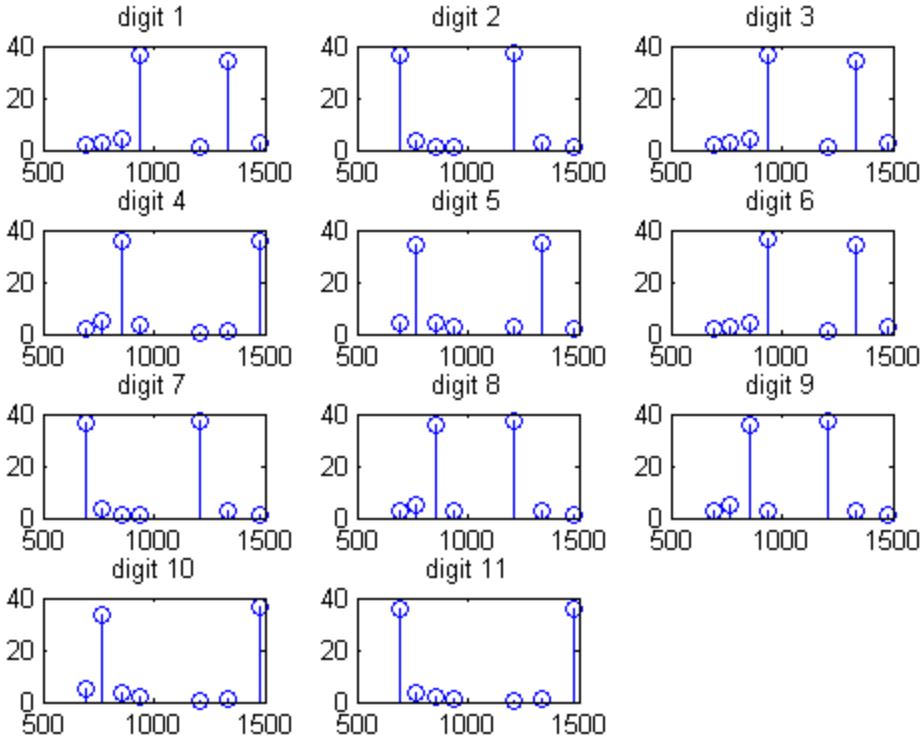
```
Nt = 205;

original_f =[697 770 852 941 1209 1336 1477];

num=[1 2 3;4 5 6 ;7 8 9;10 0 11];



[y,Fs]=audioread('C:\Users\SOUQ
COMPUTER\Downloads\audiocheck.net_dtmf_01095017763.wav');
```

```
number_og_digits=length(y)/1600;
freq_indices = round(original_f/Fs*Nt)+1;
numbers=zeros(1,number_og_digits*2);
```

step2

use goertzel algorithm to detect the db of each freq then take the absolute of them then detect the 2 max and arange them to save the low freq before the high one then plot every digit freq

```
for digits =0:number_og_digits-1;
 first=1+(digits*1600);
 end2=205+(digits*1600);

y2=y(first:end2,:);
dft_data = goertzel(y2,freq_indices);
abs_data=abs(dft_data);

subplot(4,3,1+digits);
stem(original_f,abs_data)
title(['digit ',num2str(digits+1)])

%get max1
[max1,max_index1]=max(abs_data);
abs_data(max_index1)=0;
%get max2
[max2,max_index2]=max(abs_data);
abs_data(max_index1)=0;

%arrange them
if max_index1>max_index2
numbers(2*digits+1)=max_index2;
numbers(2*digits+2)=max_index1;
else
numbers(2*digits+1)=max_index1;
numbers(2*digits+2)=max_index2;
end


end
```

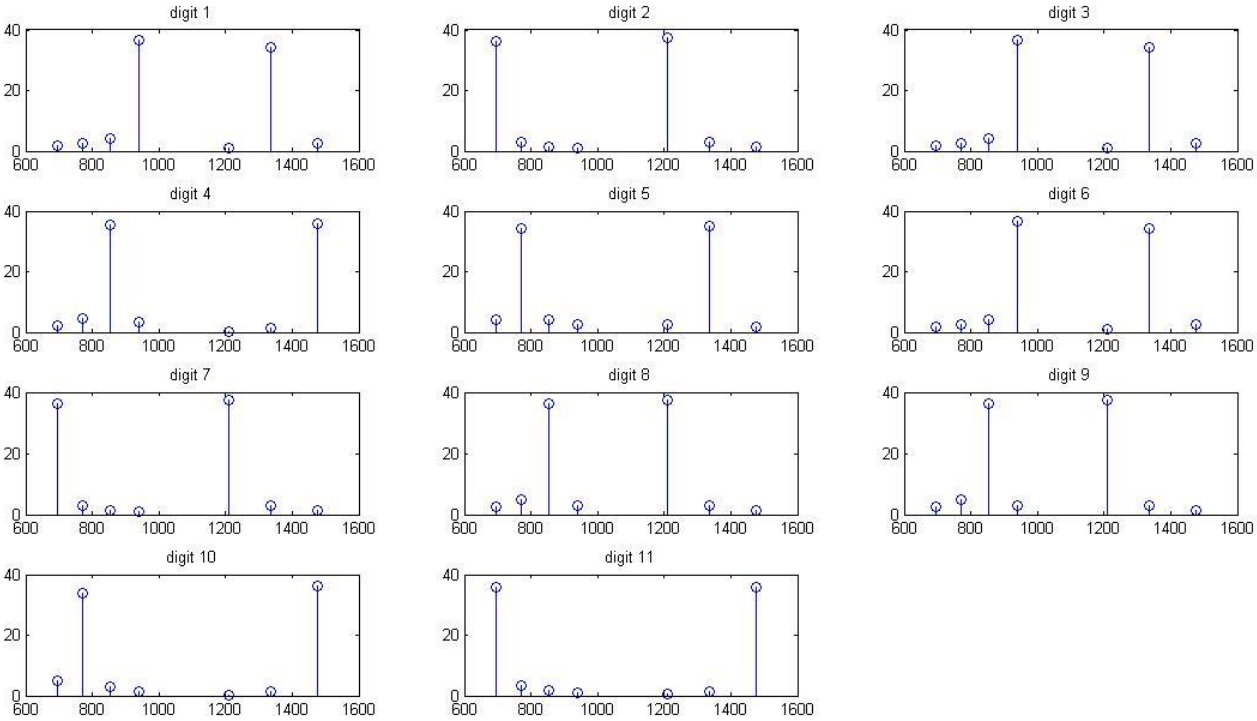


step3

```
%detect the numbers and save them in res

res=zeros(1,number_og_digits);
ind=1;
for index=1:2:(number_og_digits*2)

   res(ind)= num(numbers(index),numbers(index+1)-4);
   ind=ind+1;
end

disp(res)
```

```
 0     1     0     9     5     0     1     7     7     6     3
```

Another clear pic for the frequencies of the detected digits


digit 1
digit 2
digit 3
digit 4
digit 5
digit 6
digit 7
digit 8
digit 9
digit 10
digit 11
0
20
40
600
800
1000
1200
1400
1600