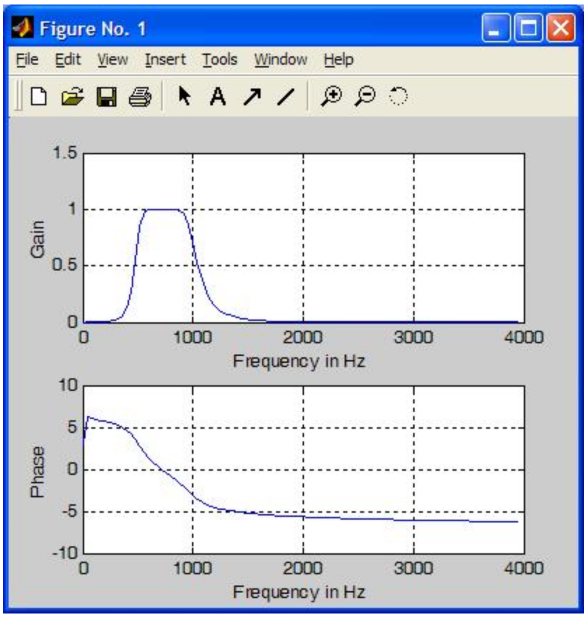
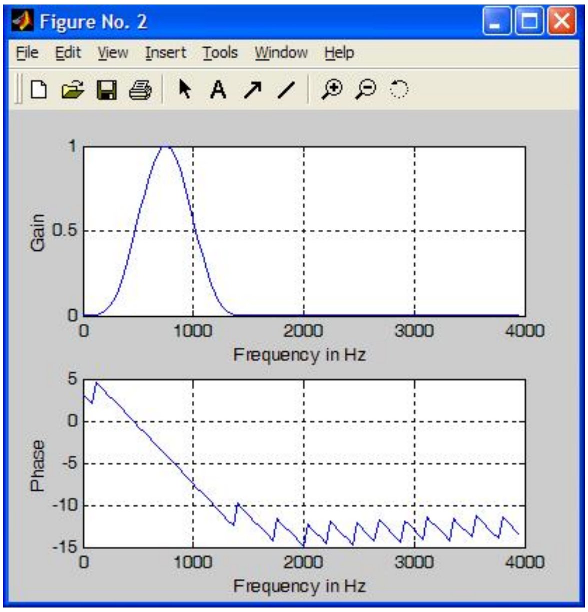
No.	Problem Examples

1	Design IIR filter (Butterworth – 8^{th} Orders), f_{CL}=500Hz, f_{CH}=1000Hz, f_s=8000Hz.
	Solution:
	To find a & b coefficients:-
	[b,a]= butter(4,[500 1000]/4000);
	zplane(b,a);
	help freqz
	[h,f]=freqz(b,a,100,8000);
	subplot(211),plot(f,abs(h));grid;xlabel('Frequency in Hz');ylabel('Gain')
	subplot(212),plot(f,unwrap(angle(h)));grid;xlabel('Frequency in Hz');ylabel('Phase');
	

2	Design FIR filter (35^{th} Orders), f_{CL}=500Hz, f_{CH}=1000Hz, f_s=8000Hz.
	To find b coefficients:-
	[b]= fir1(35,[500 1000]/4000); - by default in FIR a = 1
	zplane(b,1);
	help freqz
	[h,f]=freqz(b,1,100,8000);
	subplot(211),plot(f,abs(h));grid;xlabel('Frequency in Hz');ylabel('Gain')
	subplot(212),plot(f,unwrap(angle(h)));grid;xlabel('Frequency in Hz');ylabel('Phase');
	

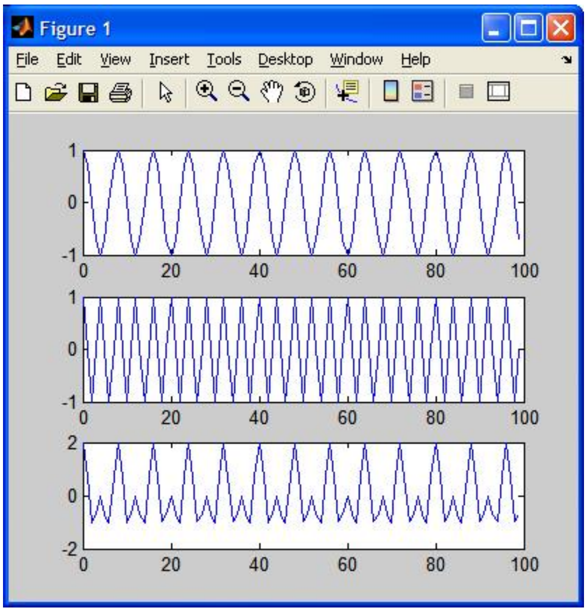
3

<<<<<Play around with the basic signal processing>>>>>

x0(1:2000)=cos(2*pi*1000*((1:2000)-1)/8000); - generate 1000Hz signal
sound(x0,8000); - create sound of the above signal
subplot(311),plot((1:100)-1,x0(1:100)); - plot the signal

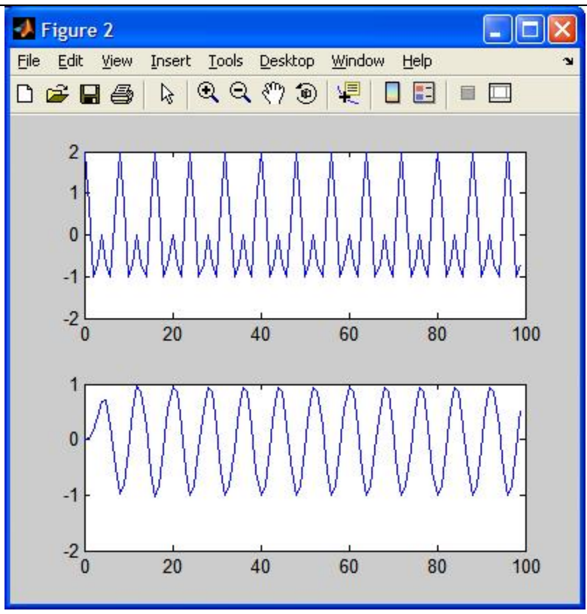
-another frequency
x1(1:2000)=cos(2*pi*2000*((1:2000)-1)/8000); - generate 2000Hz signal
sound(x1,8000); - create sound of the above signal
subplot(312),plot((1:100)-1,x1(1:100)); - plot the signal

-combination of both signals (in time domain)
x2=x0+x1;
sound(x2,8000);
subplot(313),plot((1:100)-1,x2(1:100));



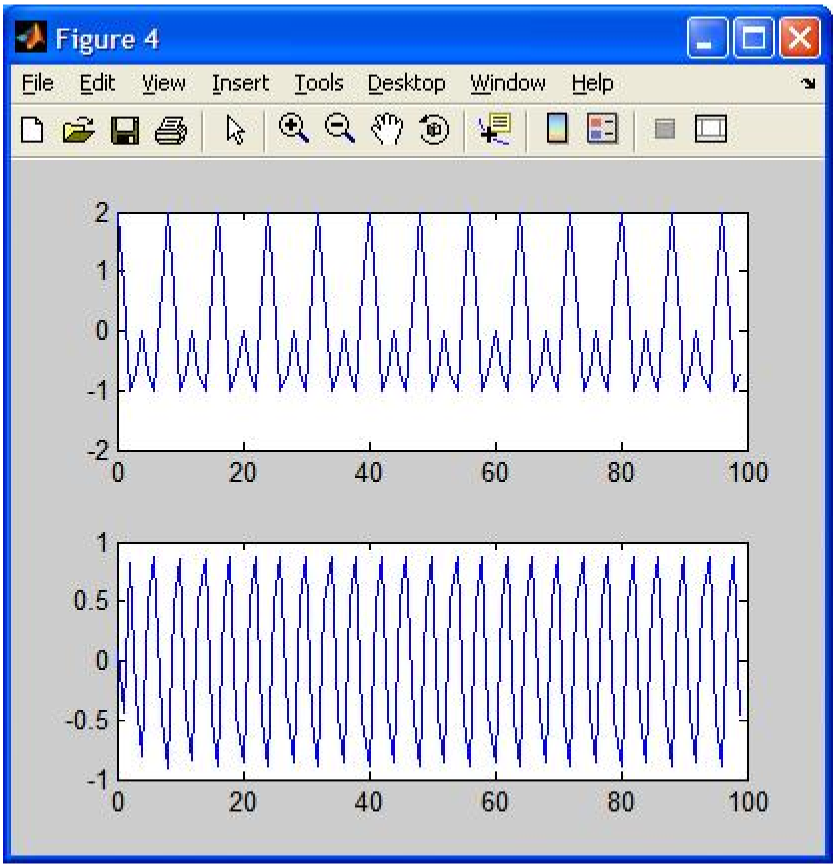
****
-introduce an IIR Butterworth filter for filtering signal x2 with a 1500Hz low-pass filter

[b0,a0]= butter(8,1500/4000);
y0=filter(b0,a0,x2);
subplot(211),plot((1:100)-1,x2(1:100)); - before low pass filter
subplot(212),plot((1:100)-1,y0(1:100)); - after low pass filter



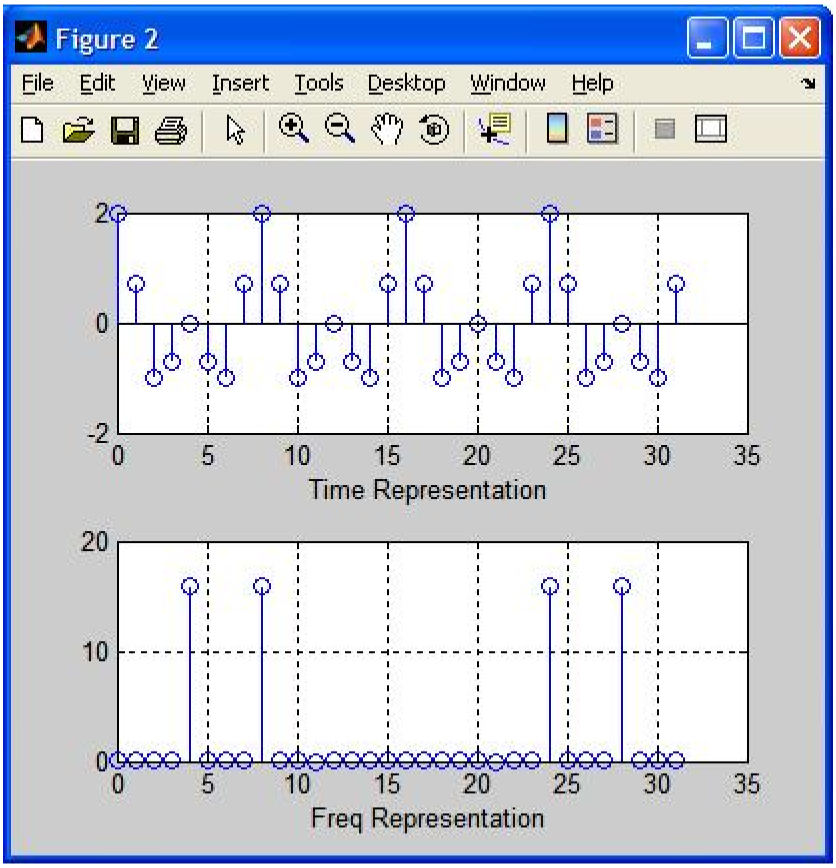
-introduce an IIR Butterworth filter for filtering signal x2 with 1500Hz high-pass filter

```
[b1,a1]= butter(8,1500/4000,'high');
y1=filter(b1,a1,x2);
subplot(211),plot((1:100)-1,x2(1:100)); -
subplot(212),plot((1:100)-1,y1(1:100));
```



4 **DFT and FFT**

```
X(1:32)=abs(fft(x2(1:32)));
subplot(211),stem((1:32)-1,x2(1:32));xlabel('Time Representation');grid
subplot(212),stem((1:32)-1,X(1:32));xlabel('Freq Representation');grid
```

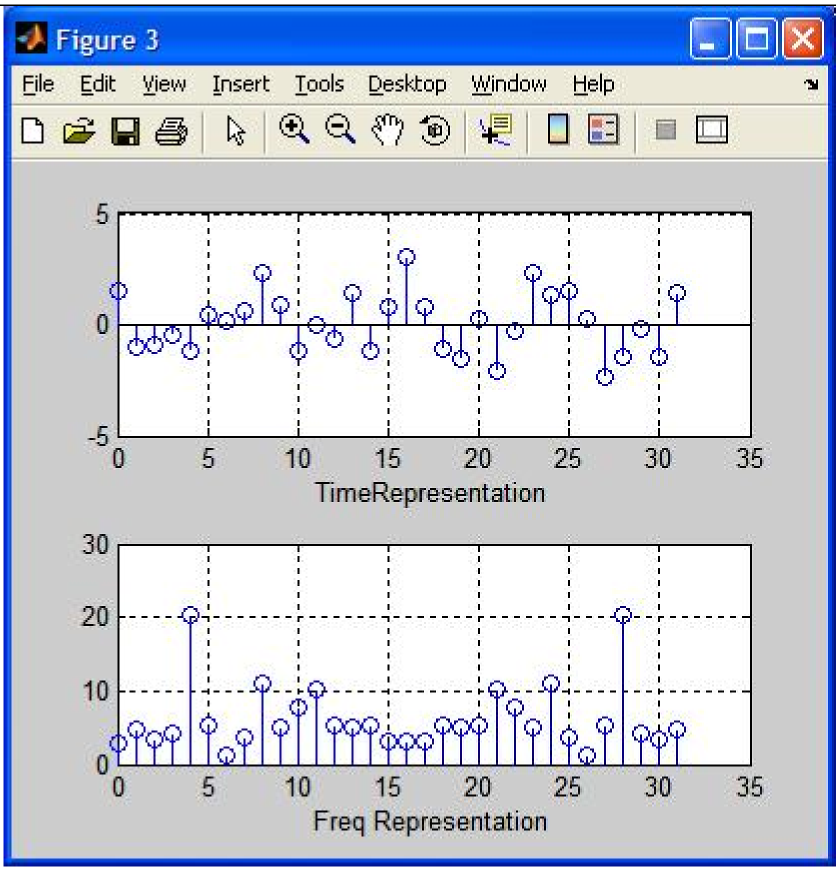


This graph is based on 32 points. Take half of the points, N=16. For point 0 to 15, the signal is the True Spectrum. For point 16 to 32, the signal is the replicated signal.

From the FFT graph, we can actually determine the frequency of the signal rather then determine the frequency of the signal using time domain ($F_s=1/T_s$). The signals can be a combination of multiple frequency signals.

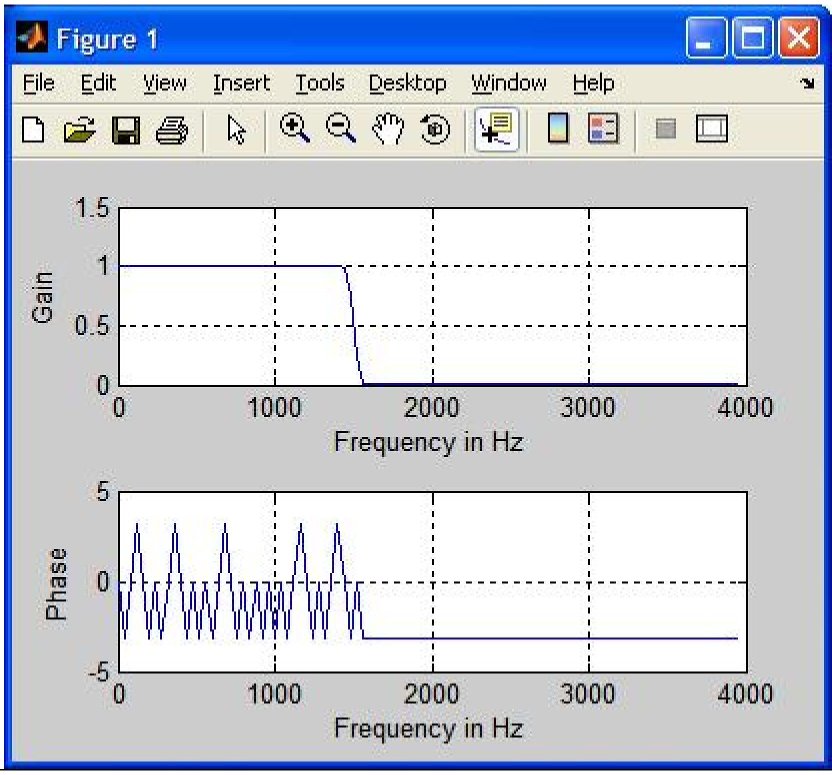
Lets introduce some noise:

```
n=randn(1,2000);
 x3=x2+n;
X3(1:32)=abs(fft(x3(1:32)));
figure; subplot(211),stem((1:32)-1,x3(1:32));xlabel('TimeRepresentation');grid
subplot(212),stem((1:32)-1,X3(1:32));xlabel('Freq Representation');grid
```

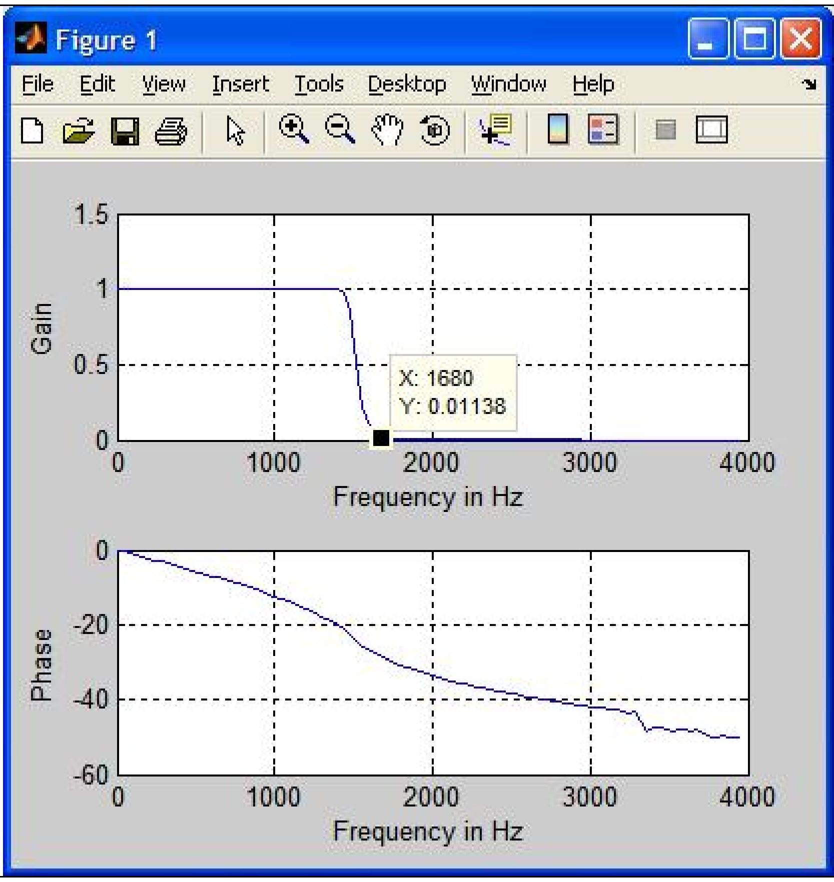
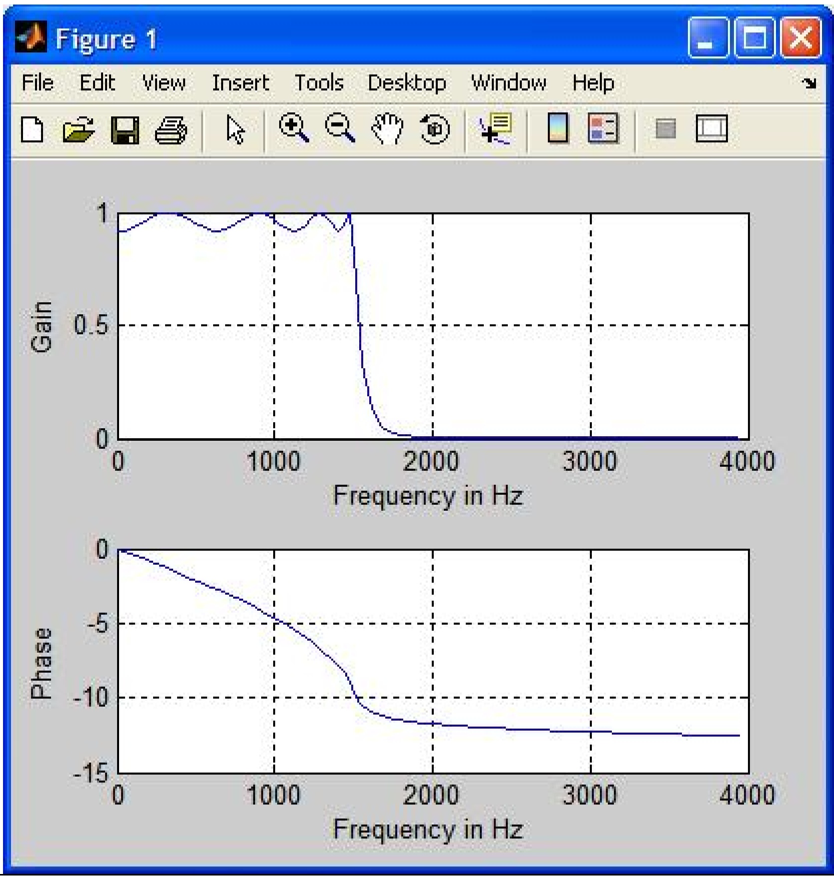
	 From the FFT graph, we still can distinguish the frequencies compared to visualize it in time domain. *MATLAB request any filtration must be done in time domain but the design stage must be in frequency domain (for DSP cases)
5	Hands-on DSP Board (C5416) File: “InOutCodec” – input from signal generator and send out a signal File: “IIRasgenerator” – DSP as signal generator, creating sinusoidal signal File:” IIRfilter” – to test IIR filter upon a given sinusoid input File: “FIRfilter” – to test FIR filter upon a given sinusoid input :Unfortunately, the DSK board only support up to 25 order. Any FIR filter with order higher than 25, the resulting output will be distorted. This is due to higher order system, the DSP needs to calculate longer mathematical equation. FIR is suitable if the system can works with delays. Otherwise, IIR is recommended. -to find an unknown filter cut-off frequency, check the output response, tune input until output is 0.7071V @ 1.414Vp-p (3dB point)!
6	Case-Studies: -read file *.wave using matlab y0=wavread(‘y0.wav’) -write file *.wave using matlab wavwrite(y1f,'c:\Y1f.wav');

<table>
<tr><th>No.</th><th>Problems</th></tr>
<tr><td>1a</td><td>
Design a low-pass filter with the following specifications:
F_s=8000Hz, f_{CL}=1500Hz, Transition Band <200Hz

Solutions:
[b]= fir1(200,[1500]/4000,'low');
zplane(b,1);
[h,f]=freqz(b,1,100,8000);
subplot(211),plot(f,abs(h));grid;xlabel('Frequency in Hz');ylabel('Gain');
subplot(212),plot(f,unwrap(angle(h)));grid;xlabel('Frequency in Hz');ylabel('Phase');



</td></tr>
<tr><td>1b</td><td>
[b,a]= butter(30,[1500]/4000,'low');
[h,f]=freqz(b,a,100,8000);
subplot(211),plot(f,abs(h));grid;xlabel('Frequency in Hz');ylabel('Gain');
subplot(212),plot(f,unwrap(angle(h)));grid;xlabel('Frequency in Hz');ylabel('Phase');
</td></tr>
</table>

<table>
<tr><td></td><td>
</td></tr>
<tr><td>1c</td><td>[b,a]=cheby1(8,0.75,1500/4000);
[h,f]=freqz(b,a,100,8000);
subplot(211),plot(f,abs(h));grid;xlabel('Frequency in Hz');ylabel('Gain');
subplot(212),plot(f,unwrap(angle(h)));grid;xlabel('Frequency in Hz');ylabel('Phase');

</td></tr>
</table>