Signal Analysis in both time and frequency domain using MATLAB

Lab # 04

Fall 2023

CSE-402L Digital Signal Processing Lab

Submitted by: **Ali Asghar**

Registration No.: **21PWCSE2059**

Class Section: **C**

"On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work."

Submitted to:

Dr. Yasir Saleem Afridi

Date:

25^{th} October 2023

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Procedure:

1. Will generate the signal of different frequencies say , 10,20,30,40,50,60 Hz (one second duration) using Matlab as shown in figure 1 and transform the same signal in frequency domain using Fourier transform and will compare the frequencies with the time domain signal as shown in figure 2.

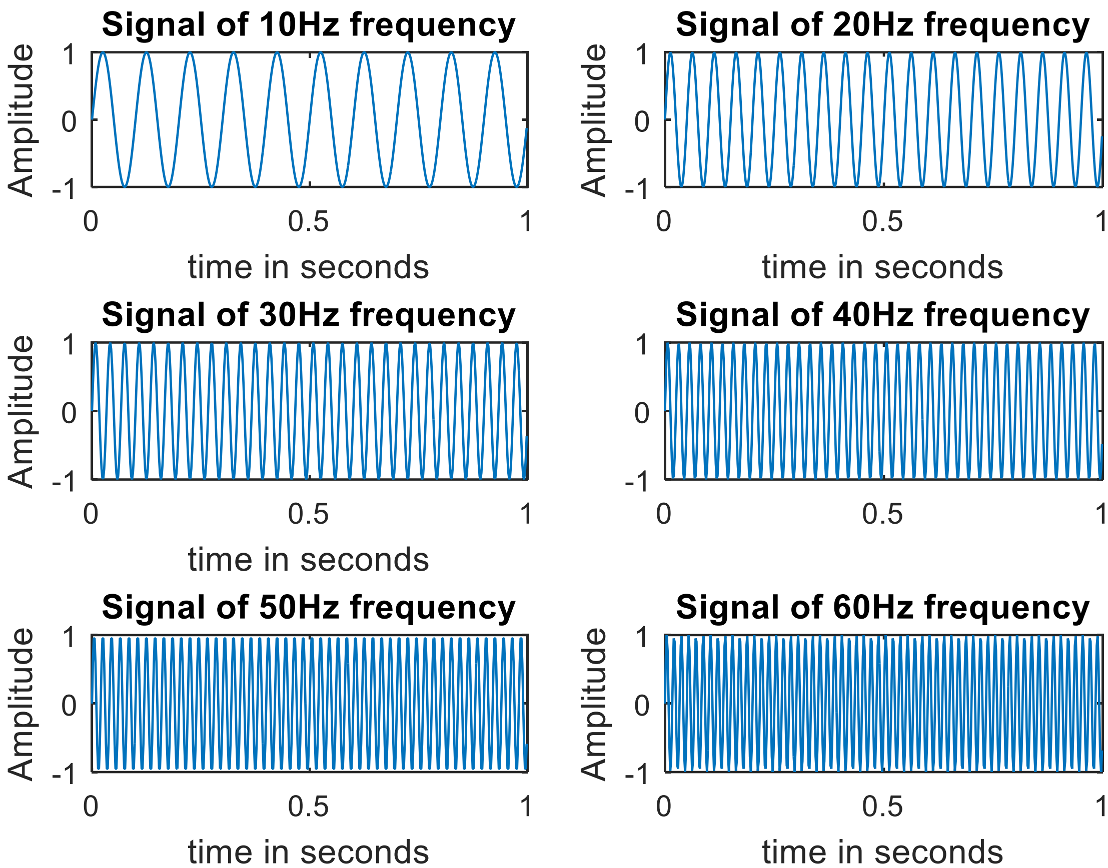


Figure 1: *Time Domain Representation*

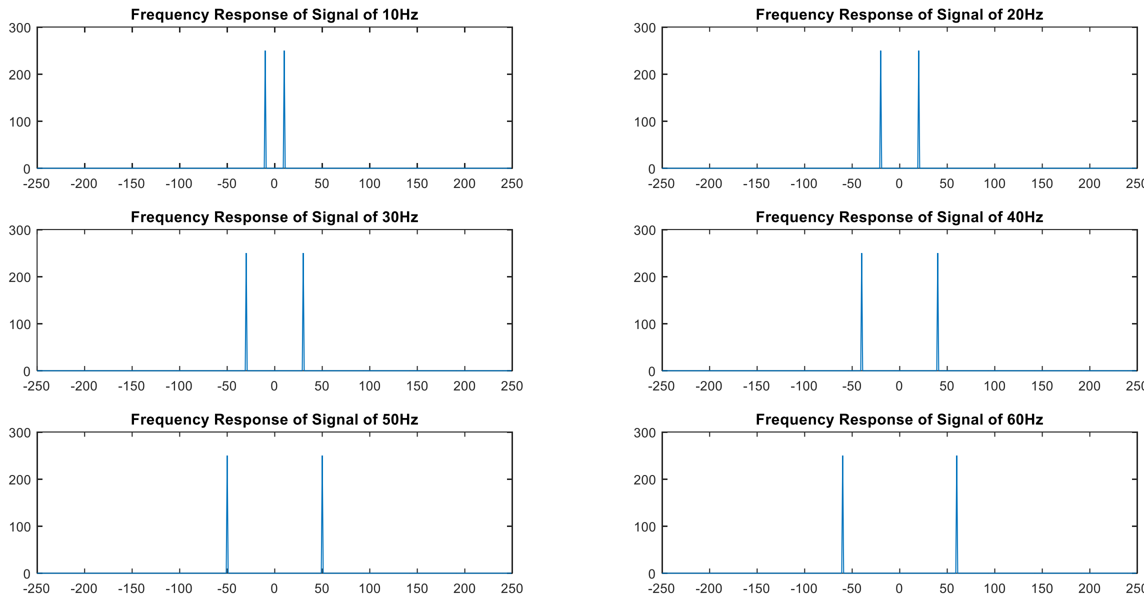


Figure 2: *Frequency Domain Representation*

2. **Compare** the Figures 1 and 2 (Generated by your code)
3. Add all the signals generated in step 1 and get a composite signal. (which may be considered as a voice signal)
4. Plot the time and frequency domain representation of the composite signal as shown in Figure 3

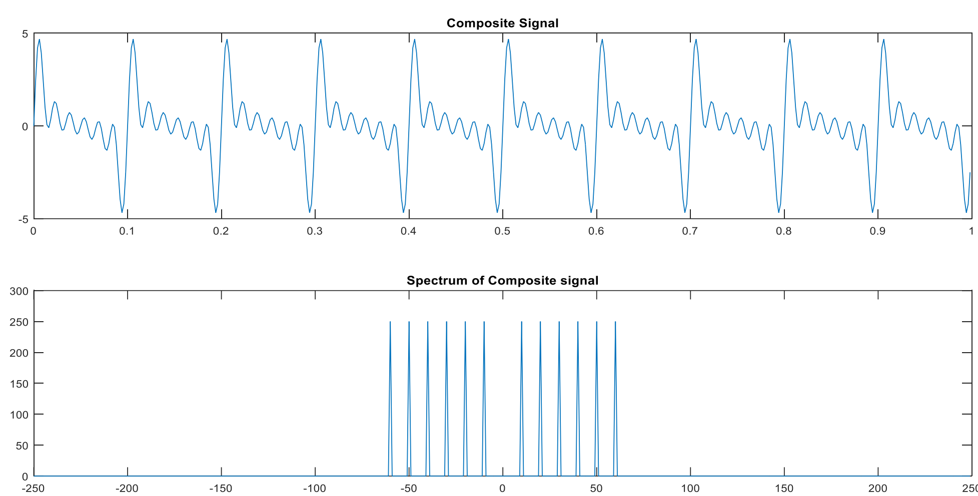


Figure 3: *Composite Signal in time and frequency domain*

5. **Compare/Confirm** that you are getting all the frequency generated in step 1 above.
6. Generate some unwanted signal having frequencies say 80Hz and 100Hz (assume these signals represent noise) and different amplitudes say 0.5 and 0.7

7. Obtain both time and frequency representation of noise and confirm they have different power as shown in Figure 4

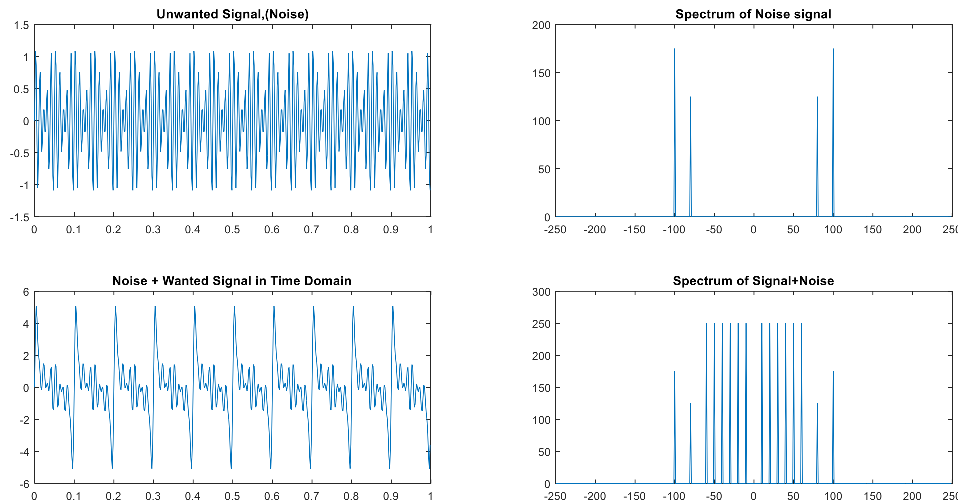


Figure 4: *Time and Frequency domain representation of Noise and Signal+Noise*

8. Add the noise to the composite signal (assume the noise is added to the signal during transmission) and obtain frequency spectrum.
9. Final Remarks/Conclusion.

Task 01:

Code:

Editor - D:\Uni\DSP Lab\Lab 04\Task1.m

Task1.m

```
t=0:0.001:1;
f=10;

figure(1);
for i = 1:6
    y = sin(2*pi*f*i*t);
    subplot(3,2,i);
    plot(t,y)
    title_str =  ['Signal of ',num2str(i*10),'Hz frequency'];
    title(title_str);
    xlabel('Time');
    ylabel('Amplitude');
    grid on;
    hold on
end

figure(2);
w=-(length(y)-1)/2:(length(y) -1)/2;

for j = 1:6
    y = sin(2*pi*f*j*t);
    x_f = fft(y, length(y));


    x_fshift = fftshift(x_f);
```

```
    subplot(3,2,j);
    plot(w,abs(x_fshift))
    title_str =  ['Frequency Response of signal of ',' ',num2str(j*10),'Hz'];
    title(title_str);
    xlabel('Frequency');
    ylabel('Amplitude');
    grid on;
    hold on
end
```

Output:

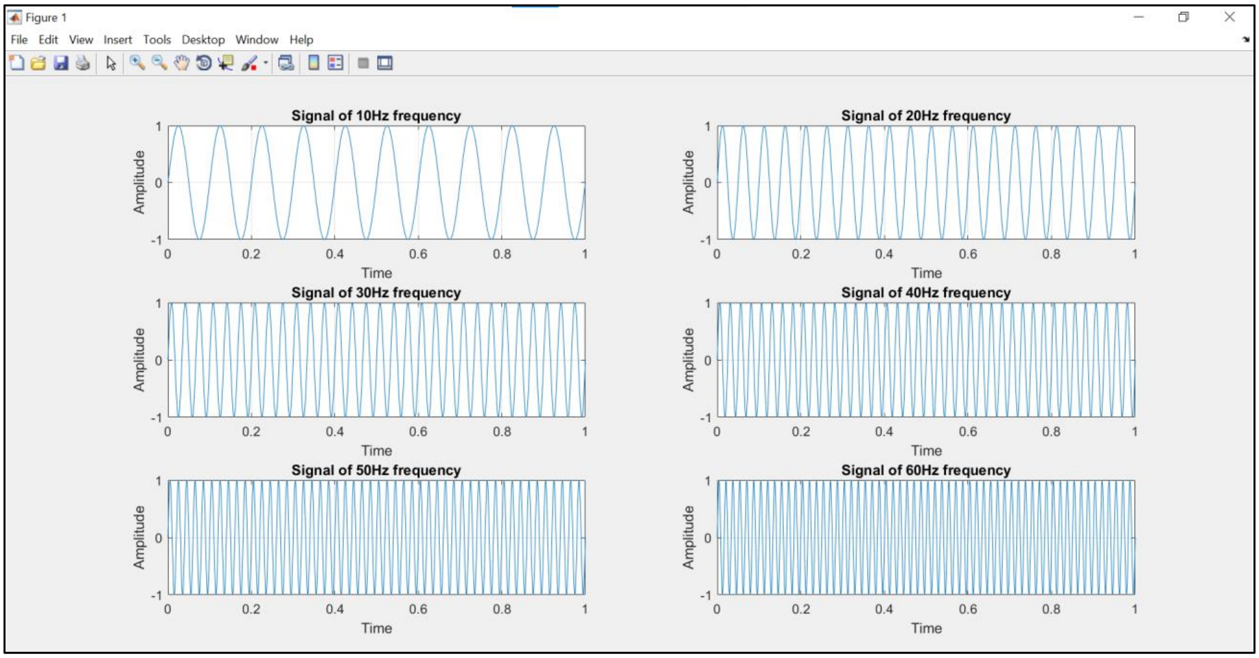


Figure 1-1: Time Domain Signals Generated in MATLAB

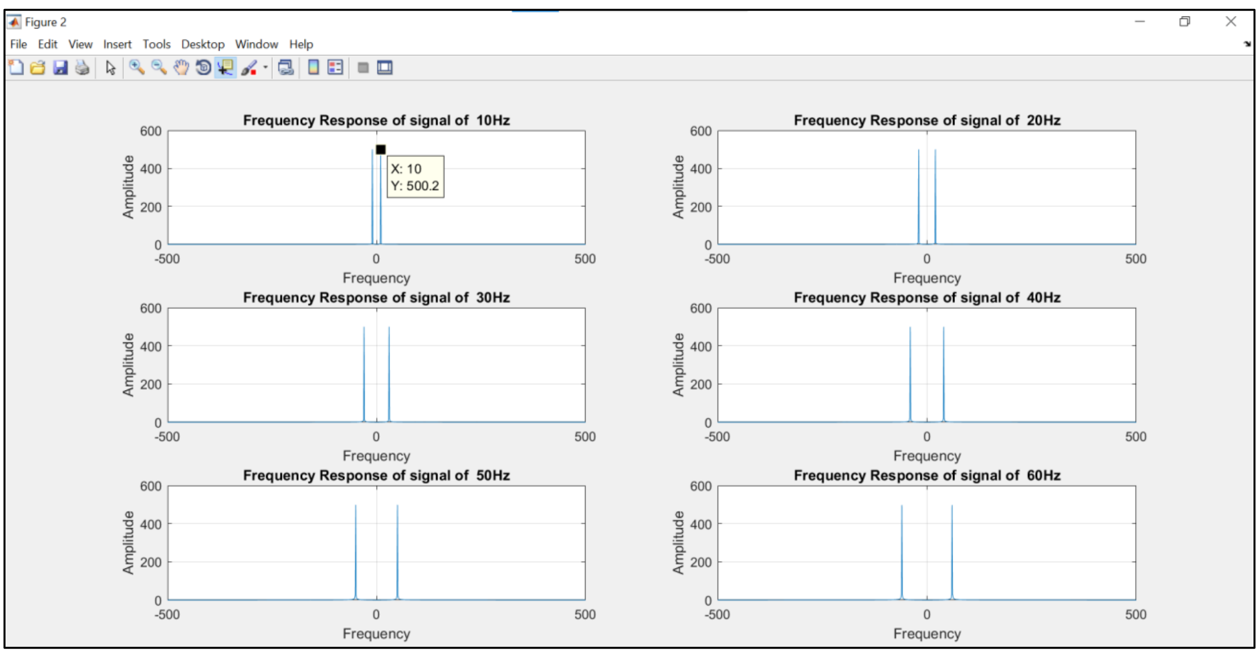


Figure 1-2: Frequency Domain of Generated Signals of Figure 1-1

Task 02:

Figure 1 (Time-Domain Representation):

In Figure 1, I created six subplots, each showing a sine wave with a different frequency. The time-domain signal is plotted in each subplot, and the title of each subplot reflects the frequency of the sine wave. The x-axis represents time (in seconds), and the y-axis represents amplitude.

Figure 2 (Frequency-Domain Representation):

In Figure 2, again I created six subplots, each showing the frequency-domain representation of a sine wave with a different frequency. The magnitude of the frequency components is plotted in each subplot. The title of each subplot reflects the frequency of the sine wave. The x-axis represents frequency (in Hz), and the y-axis represents the magnitude of the frequency components.

Major Difference:

The major difference among Figure 1 and Figure 2 is the function of plot. The time domain plot in figure 1 is represented by a sinusoidal function which has a certain frequency, amplitude and phase. The Frequency domain plot is represented by a shifted unit impulse along the spectrum of frequencies. The value of unit impulse at certain frequencies represents the frequency of the corresponding sinusoid. The frequency spectrum also contains negative frequency. In Practical life, negative frequency does not exist for a signal.

Task 03 & 04:

Code:

Editor - D:\Uni\DSP Lab\Lab 04\Task2.m

Task1.m | Task2.m

```
t=0:0.001:1;
f=10;

for i = 1:6
    y = y + sin(2*pi*f*i*t) ;
end

subplot(2,1,1)
plot(t,y)
title('Composite SIgnal');
xlabel('Time');
ylabel('Amplitude');
```

```
x_f = fft(y);
x_fshift = fftshift(x_f);

w=-(length(y)-1)/2:(length(y) -1)/2;

subplot(2,1,2)
plot(w,abs(x_fshift))
title('Fourier Transform of Composite Signal');
xlabel('Frequency');
ylabel('Amplitude');
grid on;
```

Output:

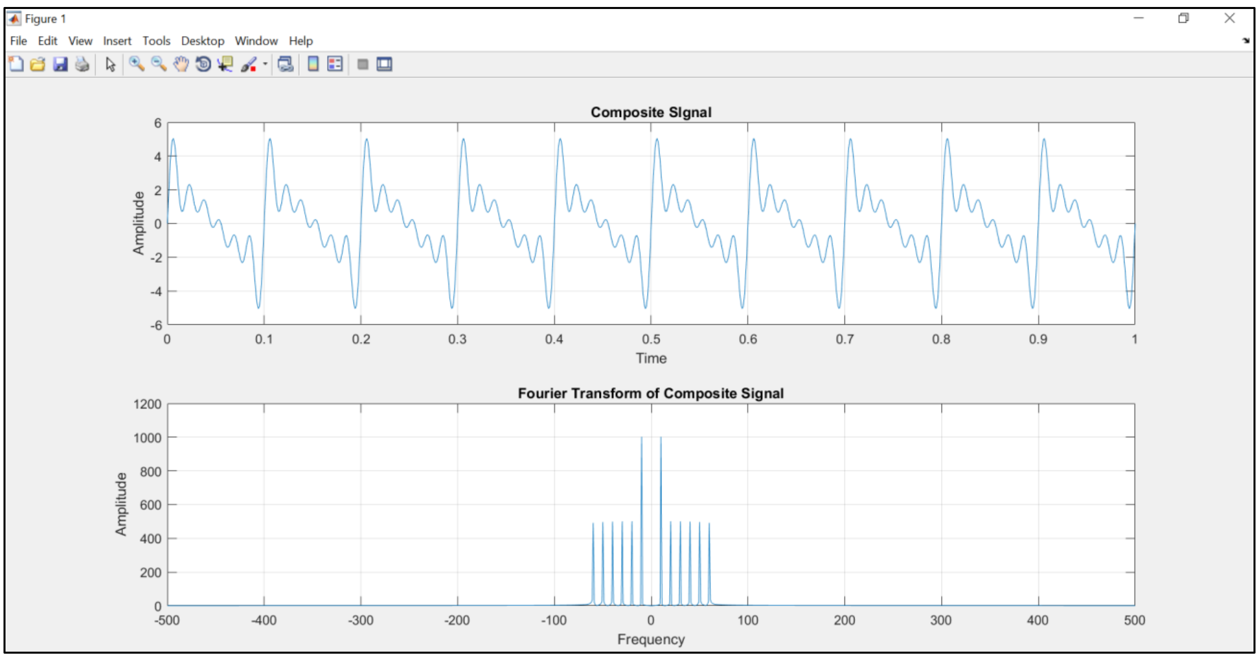
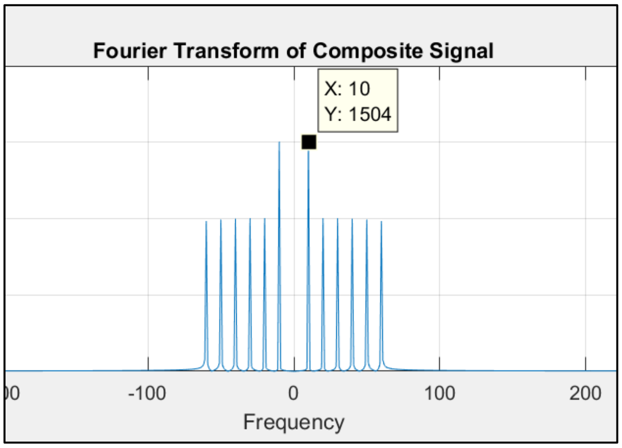
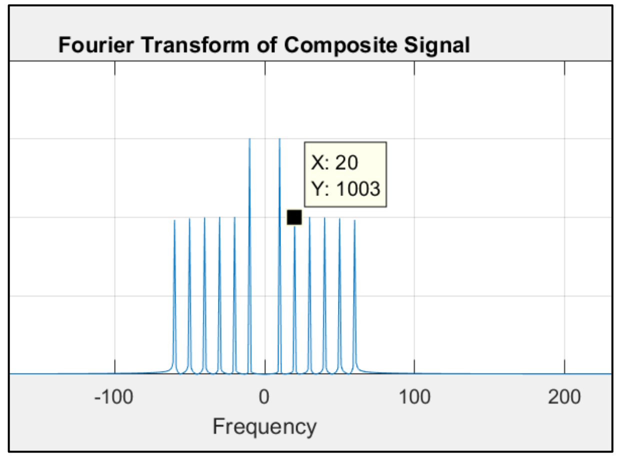
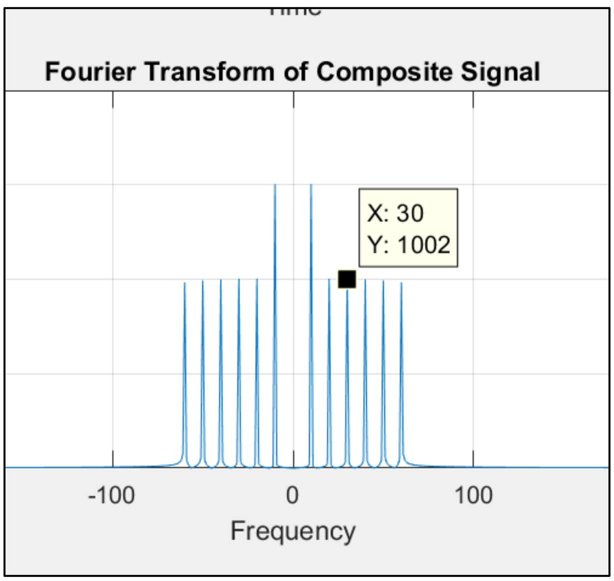


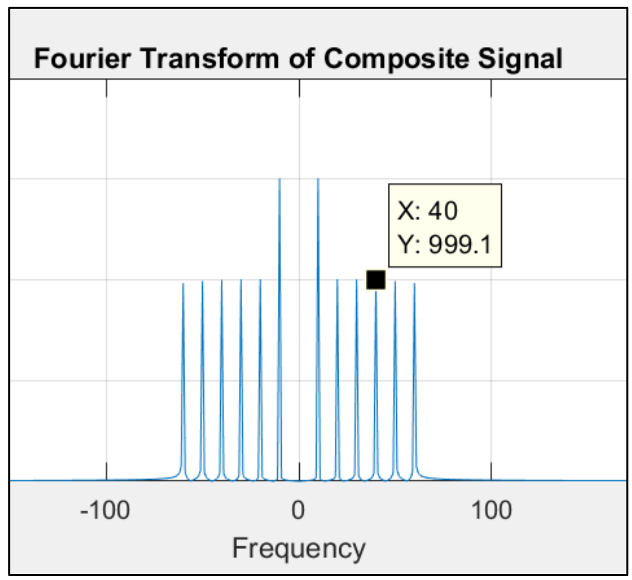
Figure 2-1: Time and Frequency Domain of Generated Composite Signal

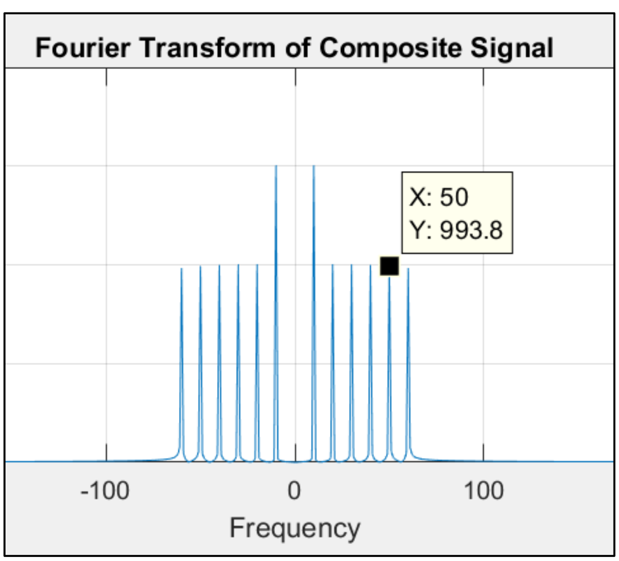
Task 05:

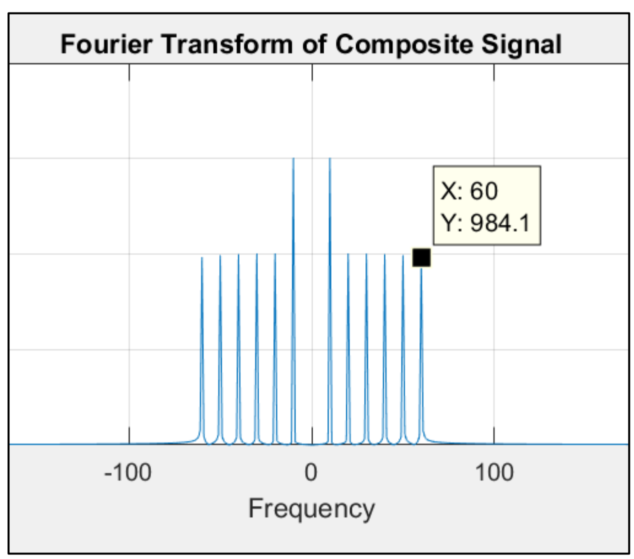

Fourier Transform of Composite Signal
X: 10
Y: 1504
-100
0
100
200
Frequency


Fourier Transform of Composite Signal
X: 20
Y: 1003
-100
0
100
200
Frequency


Fourier Transform of Composite Signal
X: 30
Y: 1002
-100
0
100
Frequency


Fourier Transform of Composite Signal
X: 40
Y: 999.1
-100
0
100
Frequency


Fourier Transform of Composite Signal
X: 50
Y: 993.8
-100
0
100
Frequency


Fourier Transform of Composite Signal
X: 60
Y: 984.1
-100
0
100
Frequency

Task 06 & 07:

Code:

```
t=0:0.001:1;
f1=80;
f2=100;

y1 = 0.5 * sin(2*pi*f1*t);
y2 = 0.7 * sin(2*pi*f2*t);
y = y1 + y2;
figure(1)
subplot(2,1,1)
plot(t,y);
title('Composite Signal');
xlabel('Time');
ylabel('Amplitude');
grid on;

w=-(length(y)-1)/2:(length(y) -1)/2;
x_f = fft(y);
x_fshift = fftshift(x_f);

subplot(2,1,2)
plot(w,abs(x_fshift))
title('Fourier Transform of Composite Signal');
xlabel('Frequency');
ylabel('Amplitude');
grid on;
```

Output:

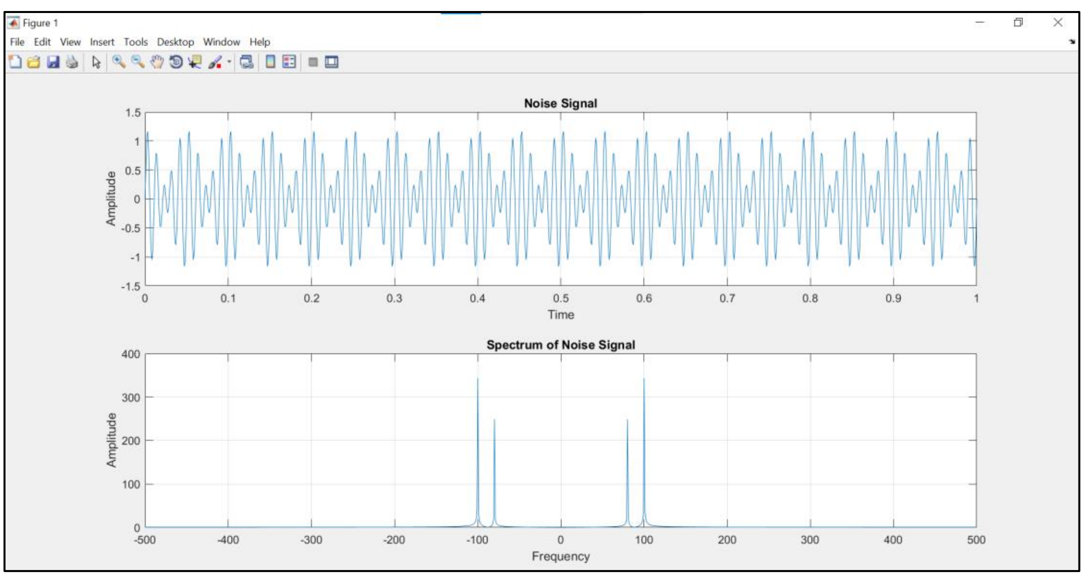


Figure 3-1: Time and Frequency Domain of Generated Noise Signal

Task 08:

Code:

Task1.m Task2.m Task3.m Task4.m

```
t=0:0.001:1;
f1=80;
f2=100;
f= 10;

y1 = 0.5 * sin(2*pi*f1*t);
y2 = 0.7 * sin(2*pi*f2*t);
y = y1 + y2;

for i = 1:6
    x = x + sin(2*pi*f*i*t) ;
end

figure(1)
subplot(2,2,1)
plot(t,y);
title('Noise');
xlabel('Time');
ylabel('Amplitude');
grid on;

subplot(2,2,3)
plot(t,x);
title('Noise + Wanted Signal in Time Domain');
xlabel('Time');
```

```
xlabel('Time');
ylabel('Amplitude');
grid on;

w=-(length(y)-1)/2:(length(y) -1)/2;
y_f = fft(y);
y_fshift = fftshift(y_f);

x_f = fft(x);
x_fshift = fftshift(x_f);

subplot(2,2,2)
plot(w,abs(y_fshift))
title('Spectrum of Noise Signal');
xlabel('Frequency');
ylabel('Amplitude');
grid on;


subplot(2,2,4)
plot(w,abs(x_fshift))
title('Spectrum of Signal + Noise');
xlabel('Frequency');
ylabel('Amplitude');
grid on;
```

Output:

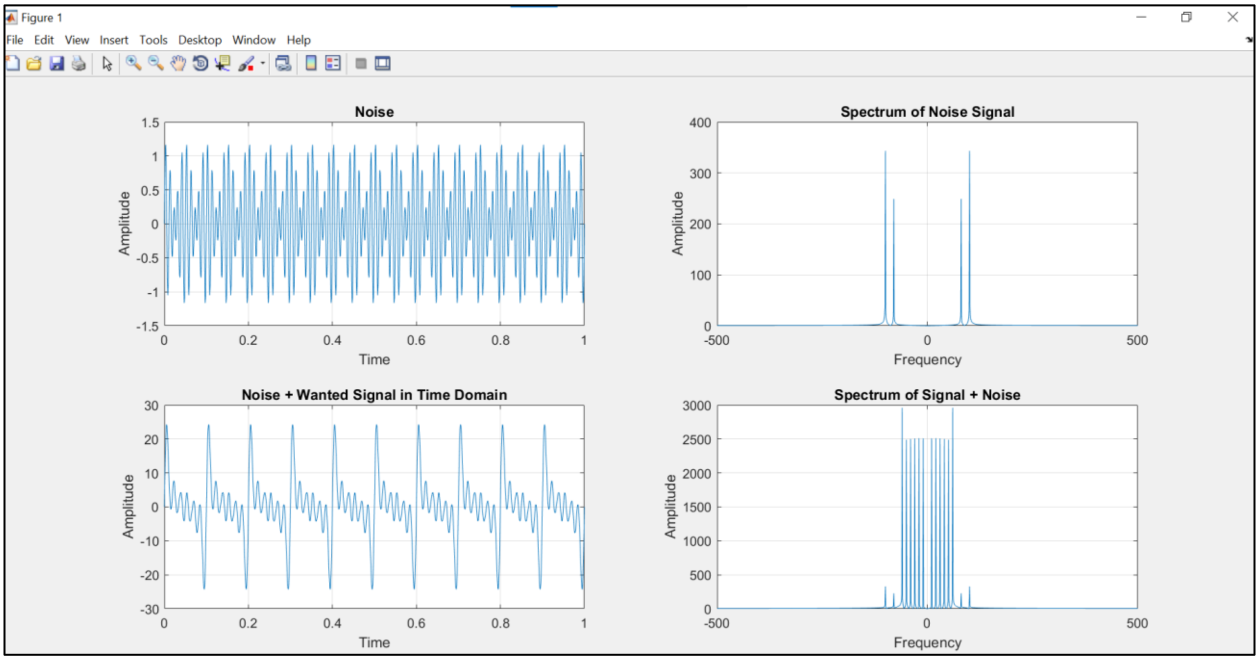


Figure 4-1: Time and Frequency Domain of Noise and Noise + Wanted Signal

Final Remarks:

In this lab I learned how to transform a time domain signal into frequency domain using fft() function. fft computes the Fourier Transform of a time domain signal. In order to correctly display the spectrum of given time domain signal, we need to shift the zero frequency component to the center of array. Then we can iterate from minus extreme through zero and to plus extreme. fftshift() function helps us in this regard. It shifts and re arranges the output of fft() and brings the zero frequency component to the center of the array.