**USER MANUAL FOR DSP TOOL V 2.0**

**Introduction:**

This project has been made by Khaled Osama Abdel Hamid. Communication and information technology engineering student at Zewail city of science and technology. All the code has been made by App designer, a feature of MATLAB® r2018a. In this report we will introduce all the new features and upgrades done on the app.

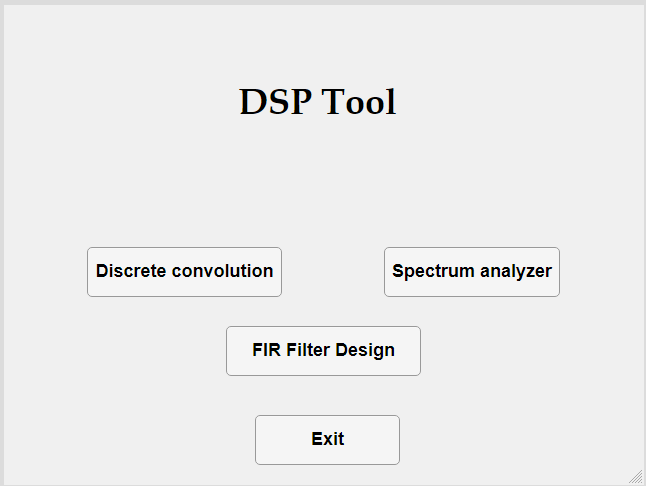


Figure 1

**Spectrum analyzer Module**

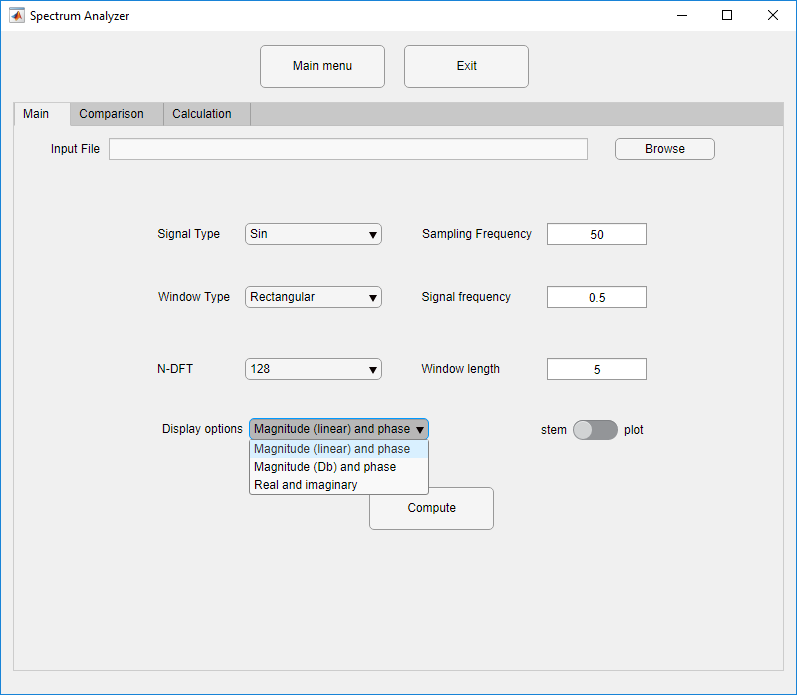


Figure 2 display options

**What’s new :**

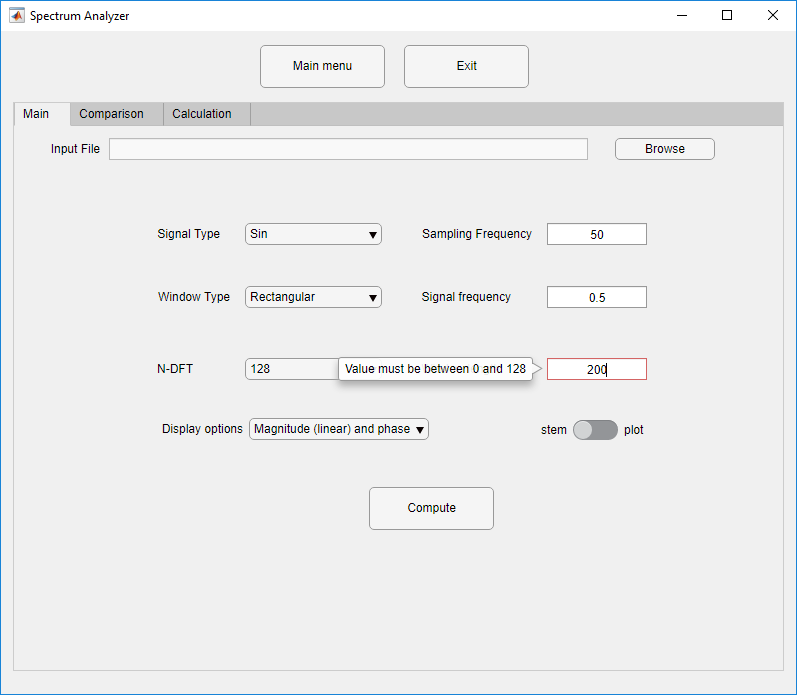
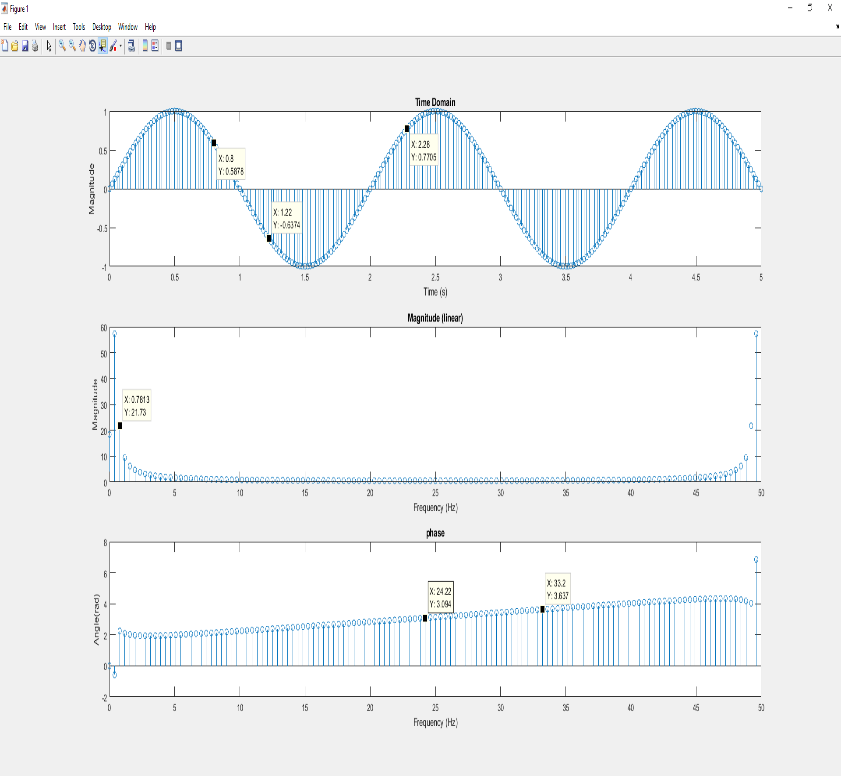
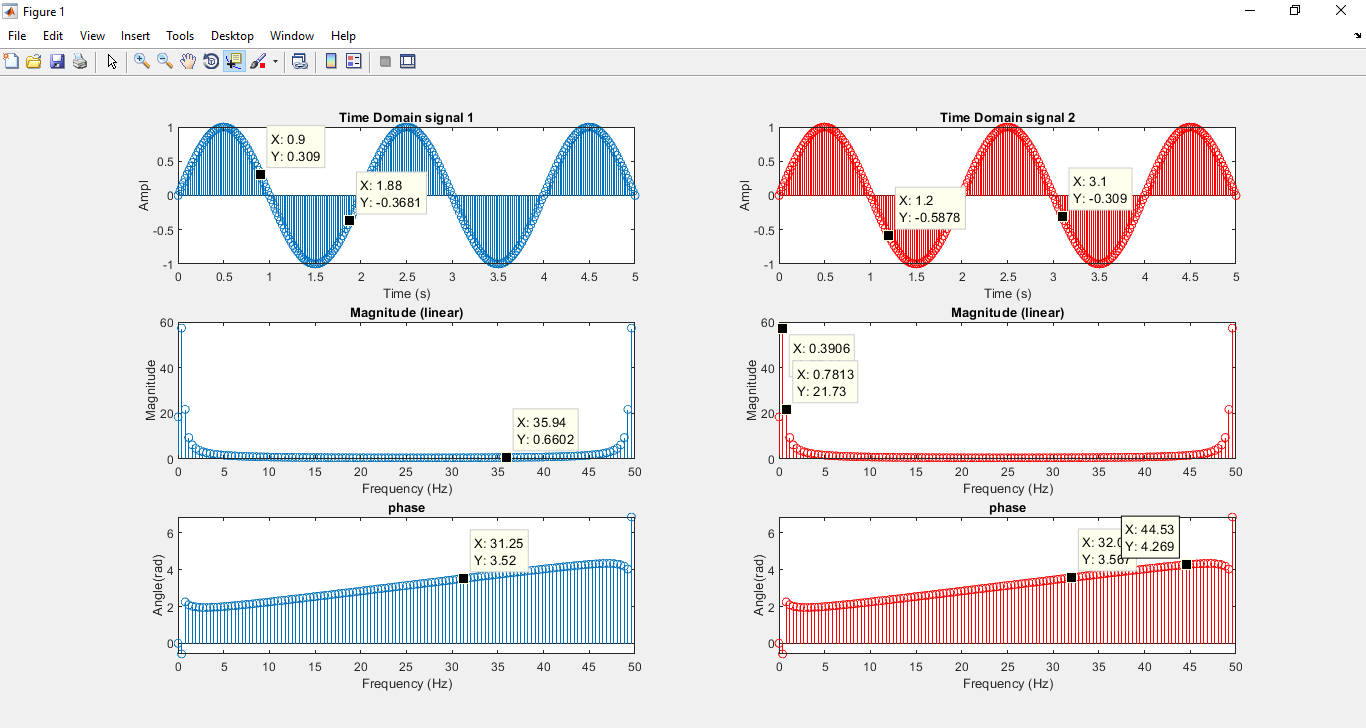
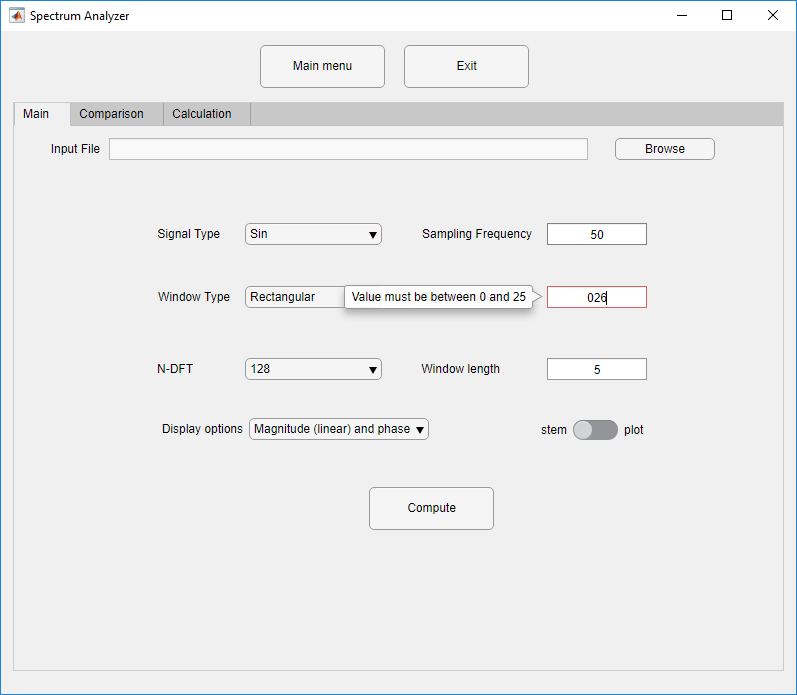
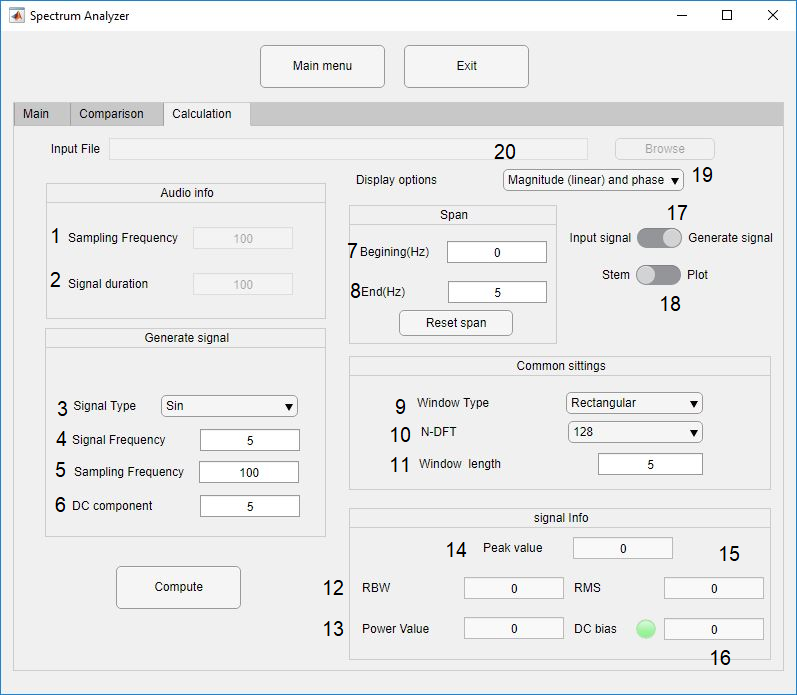
1. Display options changes as seen in the drop down menu above.
2. WE removed the graphs into separate windows for visibility and added markers that you can hold multiple of them at once in all the graphing windows in the whole modules.
3. Now for ALL windows and tabs in this app, the window length will never exceed the NDFT value in order to hold the integrity of the filter.

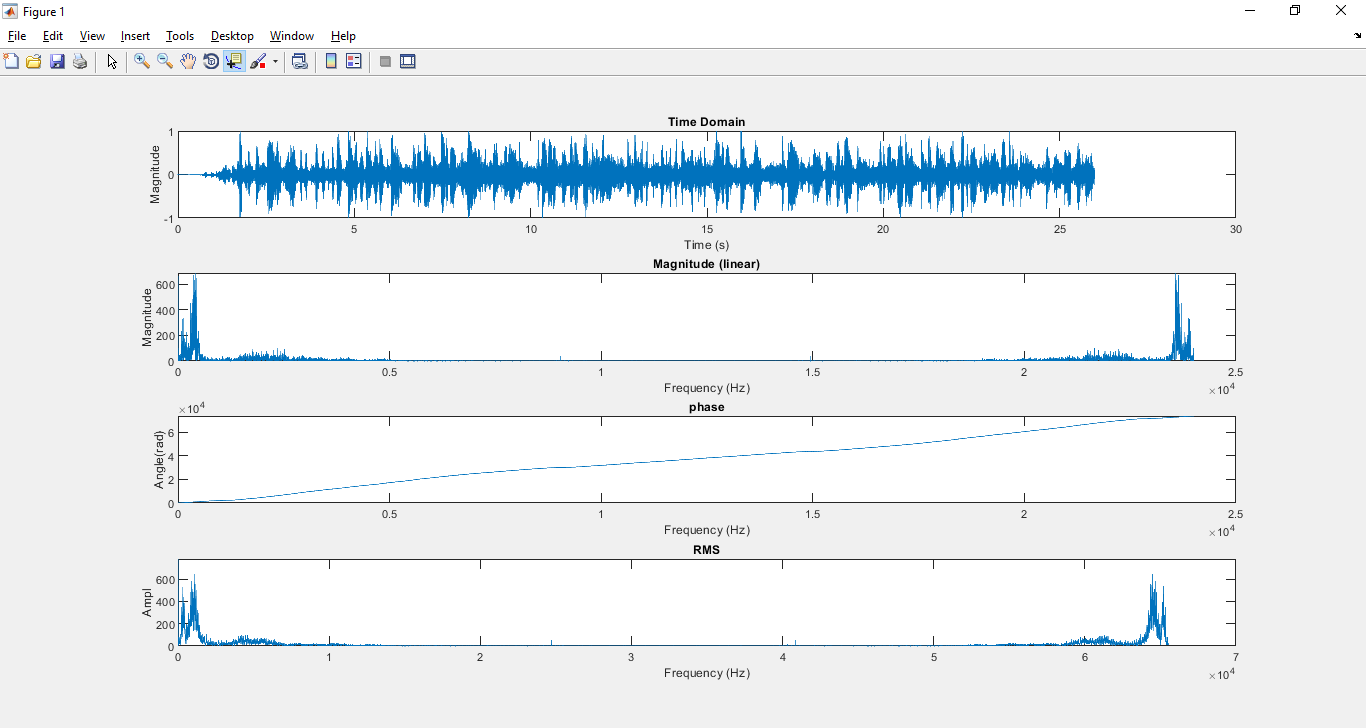
Figure 3-2 multiple markers and color changes in calculation tab

Figure 4- multiple markers addition in the main tab

1. Also, ALL signal frequency values will NEVER exceed sampling frequency /2 , to get any aliased signal with meaningless values.

**Calculation Tab**





**Main Features**

1. The sampling frequency of the audio file (un editable and disabled until you choose a file input option from switch 17).
2. The duration of the audio signal (in seconds) (un editable and disabled until you choose a file input option from switch 17)
3. It shows the type of the signal to be displayed (sin – sinc-square-input signal).
4. It specifies the Signal frequency in Hz. It ranges from (0, ∞).
5. It specifies the sampling frequency in Hz. It ranges from (0, ∞).
6. It adds a DC offset to the signal.
7. It enables you to display the frequency domain of the signal within range (beginning - ending)
8. It enables you to display the frequency domain of the signal within range (beginning - ending)
9. It shows the type of window applied (rectangular- triangular-hamming-hanning).
10. It specifies of the order of fft.
11. It specifies the Window size in seconds. It ranges from (0, ∞).
12. It shows the resolution band width which equals the sampling frequency over the total number of frequency bins. It indicates what is the difference between each frequency bin and it’s neighbor. It is a measure of how accurate the graph is (the less, the better).
13. It shows the power value within the specified span chose above. Its equation is:
14. It shows the absolute maximum value of the signal within the specified span.
15. It is the root mean squared value It’s equation is

We can see that It can be equated by getting the square root of the power.

1. It indicates whether there exist a DC Offset in the signal or not (if it is larger than .001, the lead will be turned on).
2. This switch gives you the choice of generating a signal or getting one from the PC.
3. The RMS graph, which differs than the FT signal when the NDFT is greater than 64 K as it divides the signal into signals each of them has a 2^16 bin and get the fft for each one only the average their whole result, as seen in the graph.

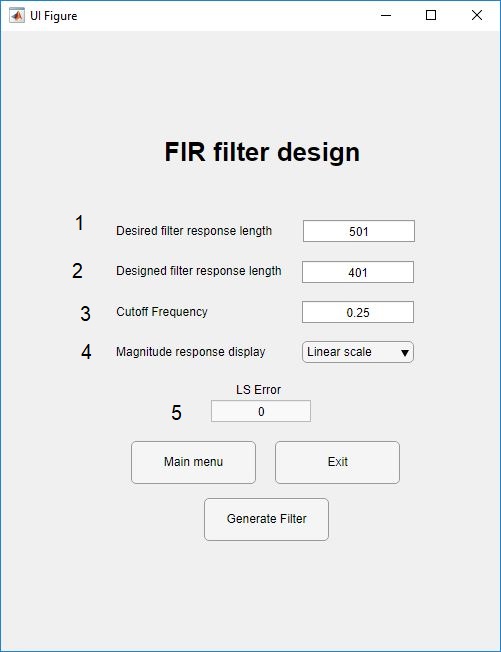
Error handing:

1. the beginning can never exceed the End,
2. The End ranges from the current beginning value and the sampling frequency.
3. All the rest of the usual errors
4. If the span is so small, the RMS and power are zeroed out for stability.

**FIR Filter design module**

**Introduction**

In this module we have implemented a low pass filter which is considered to be one of the most crucial tools in many engineering applications such as noise cancellation, sound editing and almost all the modern telecommunications, the main concept behind it is the convolution of two signals (the to be filtered signal and an impulse response) the impulse response will be our filter. As if you apply an FT to the output, you will get two multiplied signals, the first one is the frequency spectrum of the signal, and the second is the filter profile which multiplies certain frequency bins by some gain and attenuate others. It has many types and implementation techniques. Our main filter of interest is the low pass filter with the least square method. This type does not depend on FFT to get its values, rather it uses inverse matrix multiplications with simple linear algebra in order to obtain The impulse response function.



**Main Features**

1. The user should input the length of the desired filter; it is always greater than the length of the designed filter
2. The user should input the length of the designed filter; it is always less than the length of the designed filter
3. The frequency after which all the frequency content of the signal get attenuated it ranges from 0 to 1.
4. There are two display options (Linear - logarithmic).
5. Outputs The value of the least square (un editable).

Error handing:

1. The designed frequency can never exceed the desired one.
2. The cut off frequency can never exceed one, as it is a ratio of until where the frequencies should be attenuated.
3. If the user enters any even value (the app makes even symmetric- odd filters)

A warning shows and the values is automatically returned to the nearest odd number.

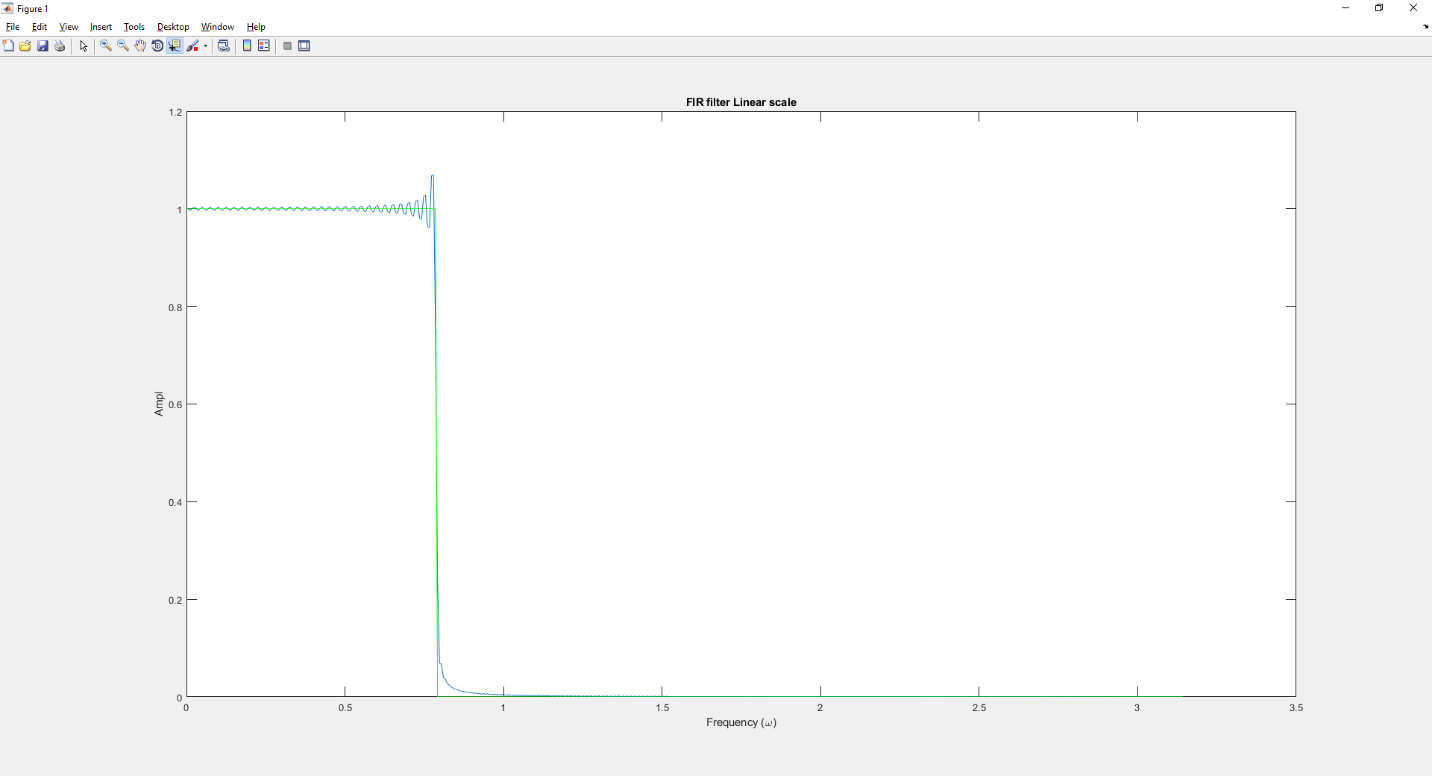


Figure 5 the desired filter (green),drawn with the designed filter (blue)

If you have any questions or enquiries, don’t hesitate to contact:

Email : [s-khaled.usama@zewailcit.edu.eg](mailto:s-khaled.usama@zewailcit.edu.eg)

Phone : +20 01062135803