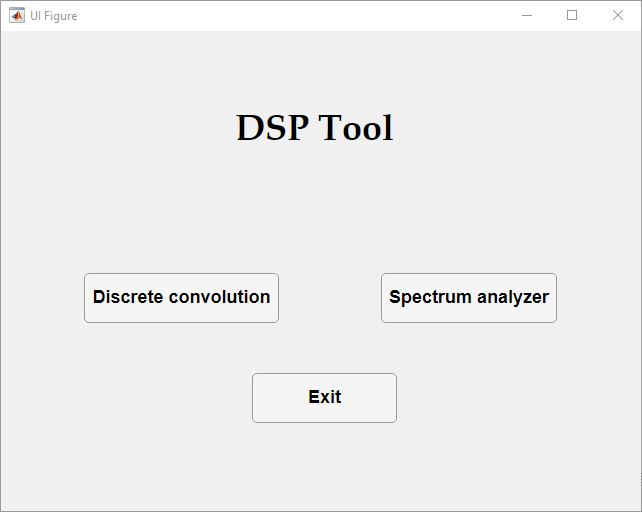
**USER MANUAL FOR DSP TOOL V 1.0**

**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 features presented so for and the rest is to be developed by December 2018.

Figure (1)

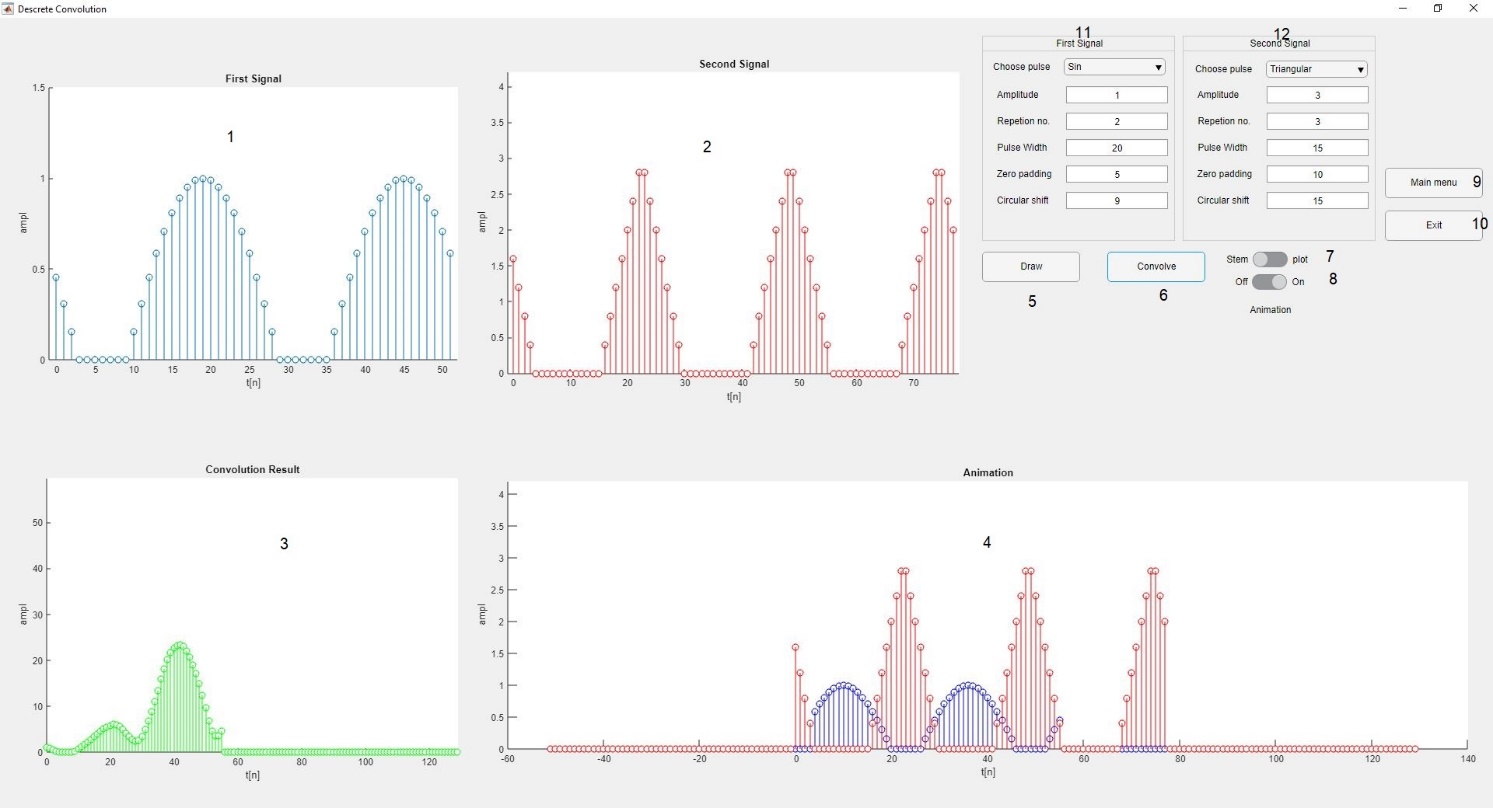
**Discrete Convolution Module**

**Theory:**

This module is made as a tool to visualize and test the convolution operation. Our main focus of the program is discrete convolution which is the same as the continuous convolution but with the summation of a sampled points from the signals. It’s equation as follows:

Where Y[n] is the output signal, X[n] is the input signal and H[n] is the impulse response. Its informal definition is the product of the common area between two signals while the first is flipped and being shifted in time.

**Main Features:**

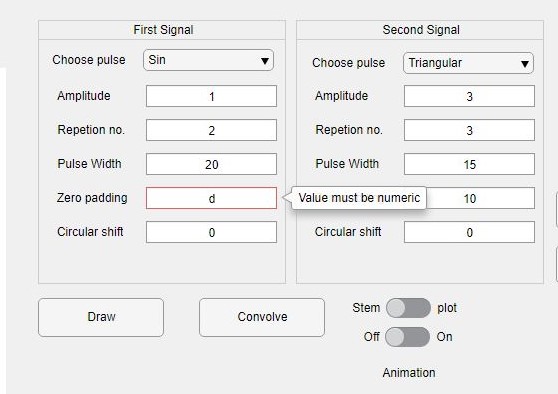
Figure (2)

The features and functionality of each component is mentioned below according to its number in figure (2):

1. The first Signal plot (in blue).
2. The Second Signal plot (in red).
3. The plot that contains the graph of the convolution result.
4. If the animation switch (8) is turned on, this graph will give a real time display for the two signals and its convolution result showing in plot number (3).
5. This button is to update the drawing on the signals plot without calculating the convolution.
6. This button is to convolve the signals according to the state of switch number (8) (animated – non animated).
7. This switch is made to transform all the signals to the continuous form. although the signals are not really continuous, but if the number of pulses is so large, it is more convenient to draw it in the plot mode.
8. This switch is responsible for the animation of the convolution.
9. This button makes you return in the main menu figure (1). It shows a confirmation window before returning as it resets all the data to the initial value.
10. This button Exits directly from the whole program. It shows a confirmation window before returning as it resets all the data to the initial value.
11. And 12 – those panels are responsible for changing the parameters of signal 1 and two respectively. Each field is as follows:

* Pulse type: it contains three. Rectangular, triangular and the positive part of the sin wave.
* Number of repetitions: It takes integer values from the range of (0, ∞). It repeats the signal n times and contaminate them in the graph.
* Pulse width: It takes integer values from the range of (0, ∞). It defines the number of samples in every pulse (before repeating and concatenating).
* Zero padding: It takes integer values from the range of [0, ∞). It defines the number of samples zeros padded after each pulse. It is mainly used to separate each pulse with a convenient number of zeros.
* Circular shift: It takes integer values from the range of (-∞, ∞). It defines the number steps to be circularly shifted.

**Error handling:**

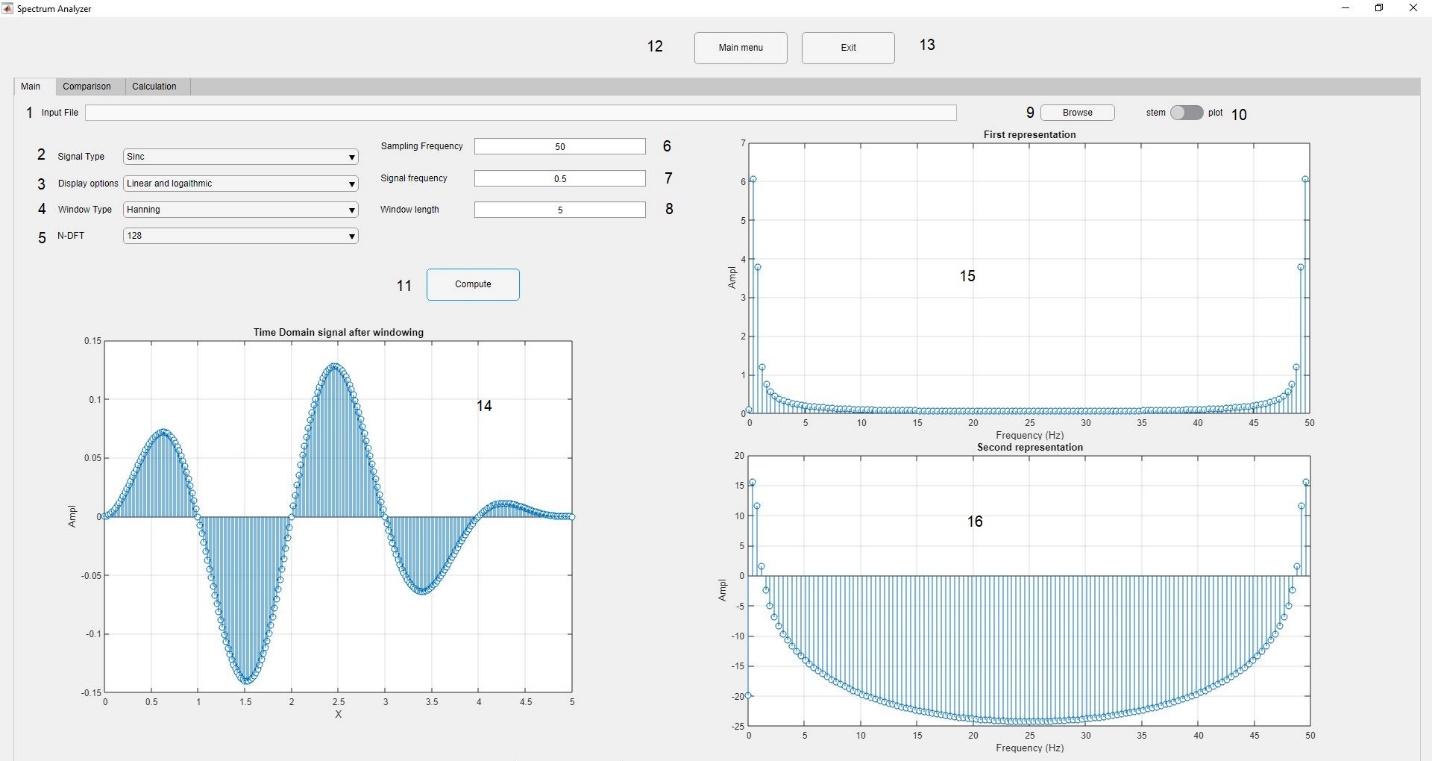
1. If the user enters any value out of the range or non-numeric, the GUI notifies him and returns to the last valid value.

**Spectrum analyzer Module**

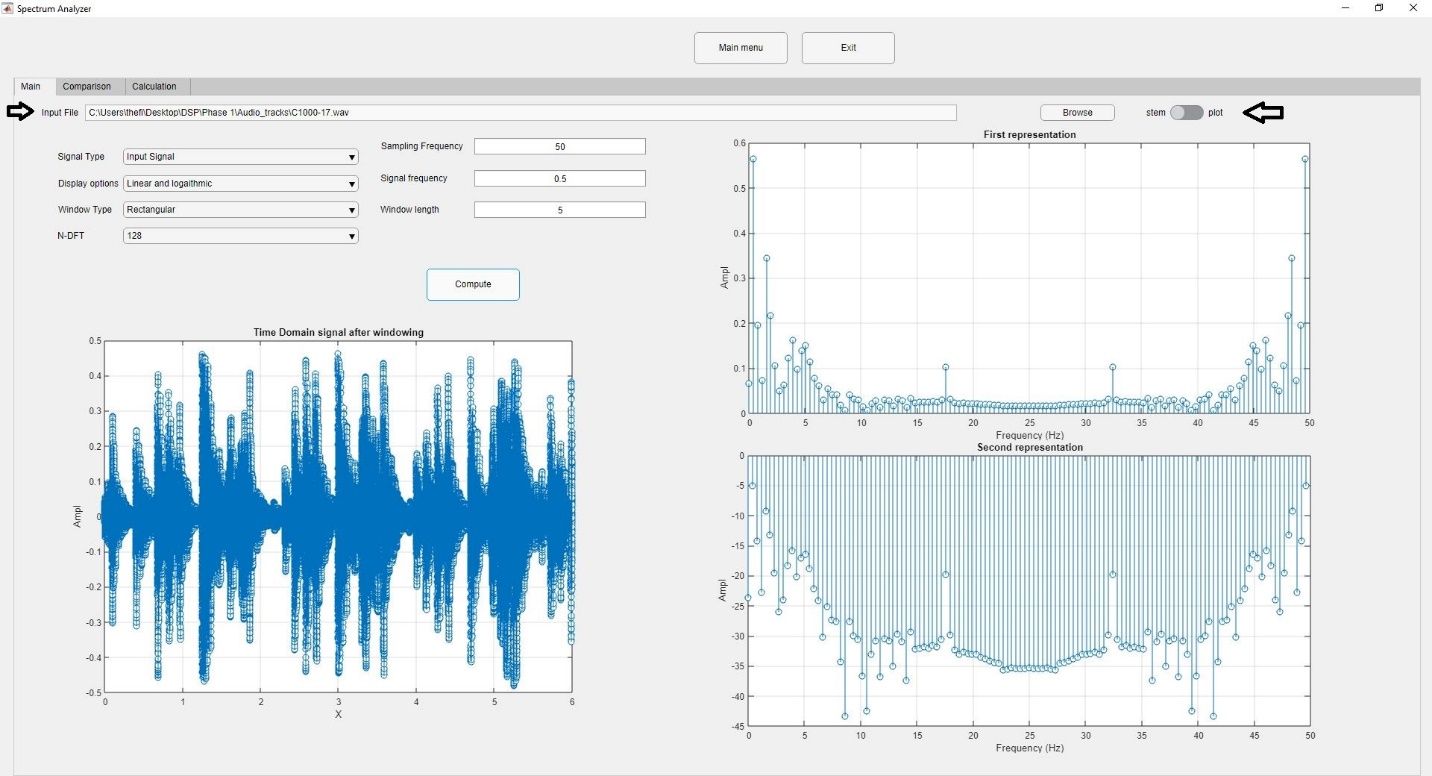
**Theory:**

Spectrum analysis is Essential in almost all DSP applications. This Module is based on the concept fast Fourier transform. It is basically another algorithm to perform Discrete transform. However, the main difference between the DFT and FFT is that the first has a O() while the FFT algorithm has a O(n log(n)) which decrease the time extensively when dealing with large samples. The second core concept is windowing, which is taking a portion of the signal and multiply it with a certain function in order to decrease the frequency leakage as much as possible. This module enables the user to visual and see the frequency spectrum of multiple functions windowed with various types of windows e.g. (hamming, rectangular).

**Main Features (Main mode):**

Figure (3)

1. This Edit field displays the path of the loaded folder.
2. It shows the type of the signal to be displayed (sin – sinc-square-input signal).
3. It shows the type of display (linear and logarithmic-magnitude and phase-real and imaginary). Those are to be displayed in the plots 15 and 16 respectively.
4. It shows the type of window applied. (rectangular- triangular-hamming-hanning)
5. It specifies of the order of fft.
6. It specifies the sampling frequency in Hz. It ranges from (0, ∞).
7. It specifies the Signal frequency in Hz. It ranges from (0, ∞).
8. It specifies the Window size in seconds. It ranges from (0, ∞).
9. This button is to load the audio file from the PC.
10. It is used to switch the drawing of the function in a continuous graph or discrete.

For example, Figure (4) becomes figure (5) after changing the switch and pushing the button. Which make the data more representative and readble.

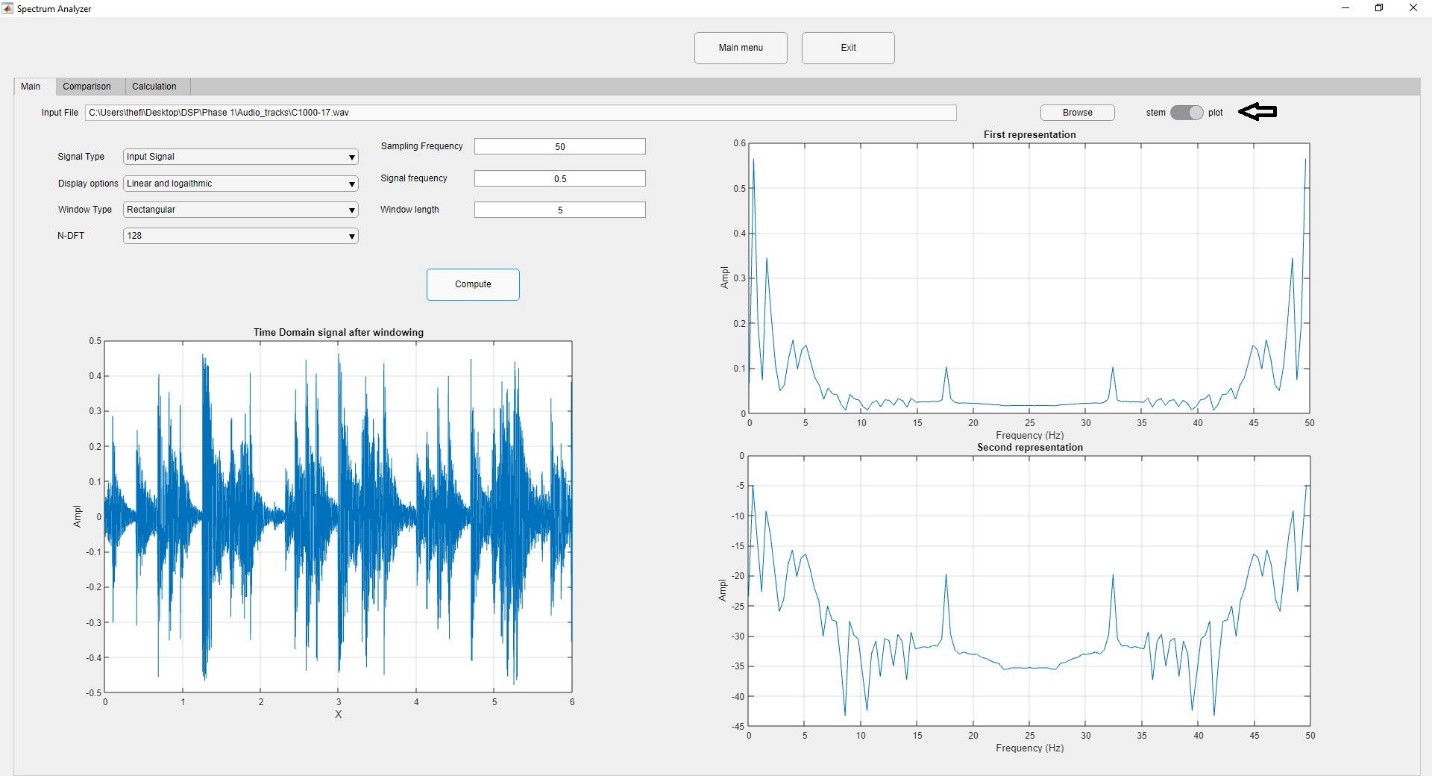
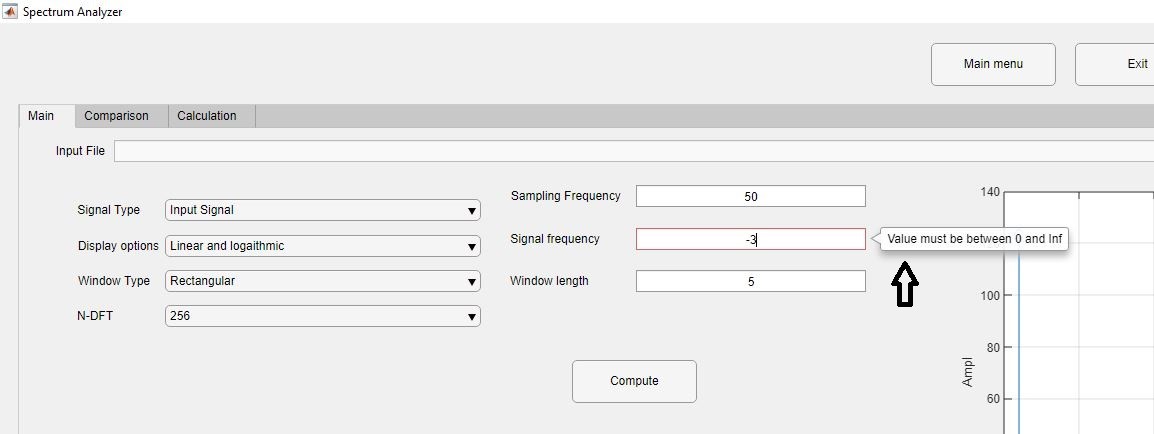
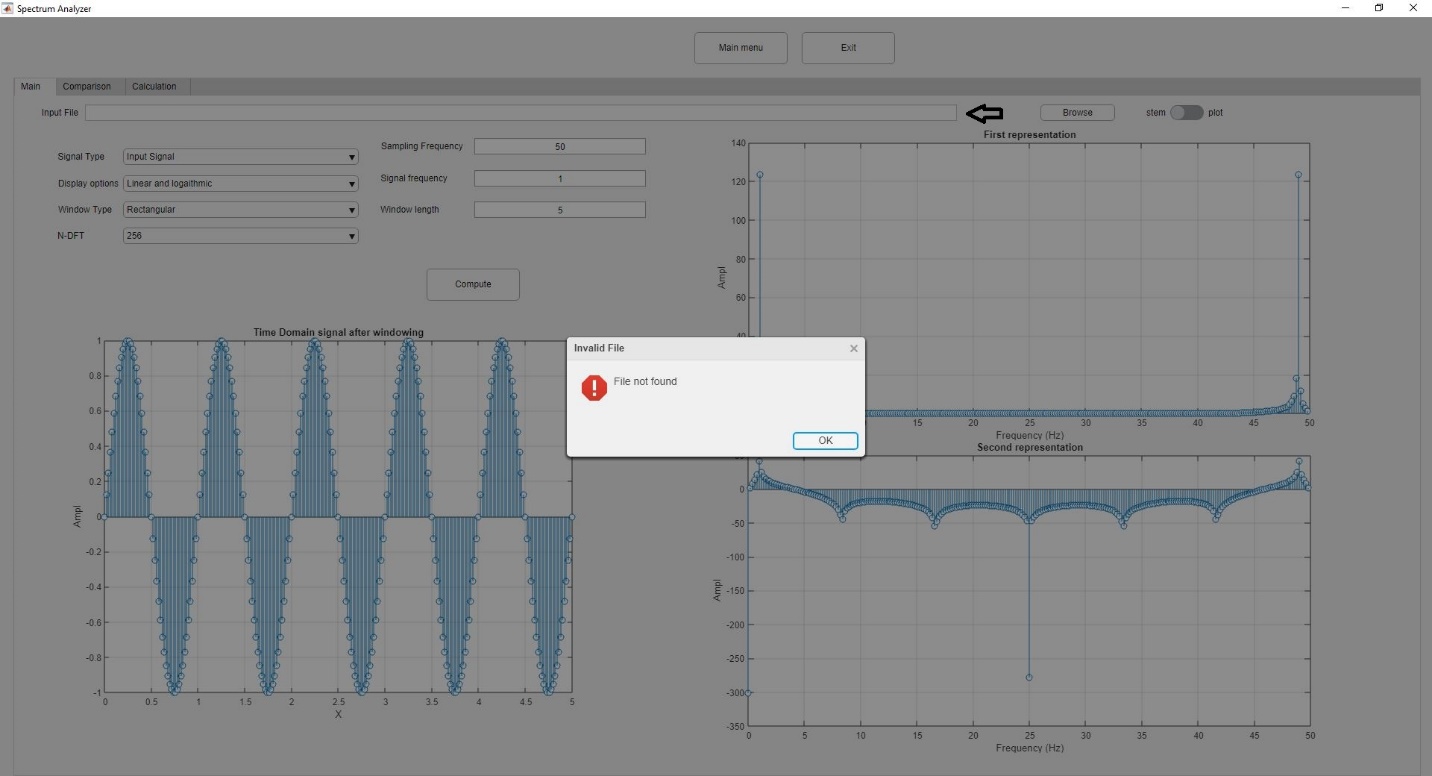
Figure (4)

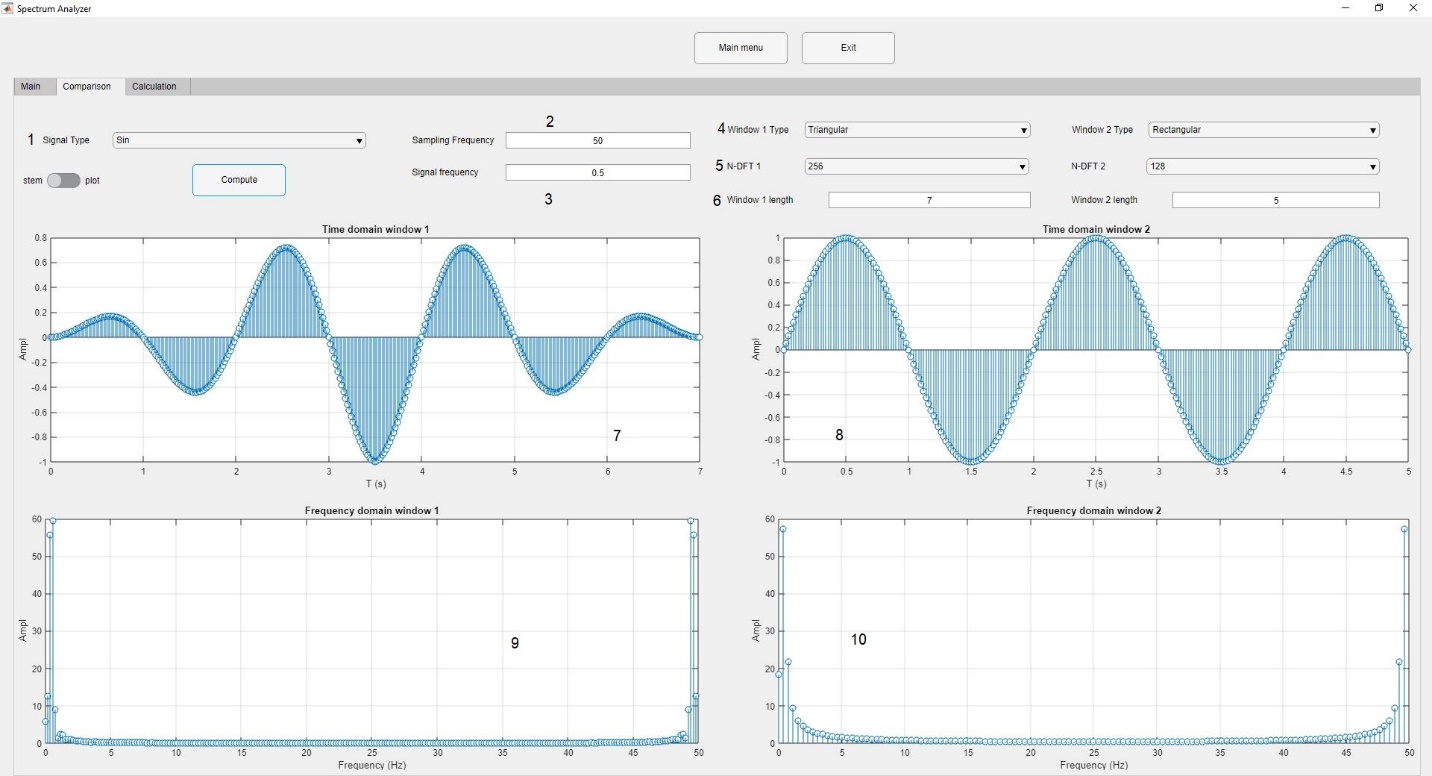
Figure (5)

1. This button initializes the computation of the frequency spectrum.
2. The time domain graph of the signal.
3. The first representation graph of the signals. See drop down menu (3).
4. The second representation graph of the signals. See drop down menu (3).

**Error handling:**

1. If the user enters any value out of the range or non-numeric, the GUI notifies him and returns to the last valid value.
2. The input file field is always disabled so that the user does not enter an invalid path.
3. If the users press computes without entering a valid path file, a warning message appears and draws a normal sin() function.

**Main Features (comparison mode):**

Figure (6)

1. It shows the type of the signal to be displayed (sin – sinc-square-input signal).
2. It specifies the sampling frequency in Hz. It ranges from (0, ∞).
3. It specifies the Signal frequency in Hz. It ranges from (0, ∞).
4. It shows the type of window applied (rectangular- triangular-hamming-hanning).
5. It specifies of the order of fft.
6. It specifies the Window size in seconds. It ranges from (0, ∞).
7. The time domain of the first signal with the first window applied.
8. The frequency domain of the first signal with the first window applied.
9. The time domain of the second signal with the second window applied.
10. The frequency domain of the second signal with the second window applied.