Audio Equalizer

Prepared by:

Name	ID	Group
Youssef Samuel Nachaat Labib	6978	2
Youssef Amr Ismail Othman	6913	2
Arsany Mousa Fathy Rezk	6927	2

Introduction:

By adjusting a variety of different frequency signals, equalizer compensates or optimizes system deficiencies, equalizer is used in voice, communication systems, mechanical vibration, fault diagnosis and many other fields. Traditional analog equalizer has low precision, phase nonlinearity and more distortion characteristics. On the basis of this, digital equalization is maintaining an ultimate quality.

As a result, this program has been made to work as an equalizer using digital (FIR and IIR) filters in addition to a visual signal representation in both time and frequency domains.

The program has the functionality to filter the given bands:

(0-170 Hz) -(170-310Hz) -(310-600 Hz) -(600-1000 Hz) -(1-3 kHz)-(3-6 kHz) -(6-12 kHz) -(12-14 kHz) -(14-16 kHz)

Using FIR filters and IIR filters with ability also to change the gains applied on any band.

Additionally, 2 possible representations are available (Time domain representation) and (Frequency domain representation).

Moreover, impulse, step, and frequency responses in addition to zeros and poles of each filter are available to be shown when needed.

And here is an illustration for the code and a user manual with sample runs to explain the implementation and usage of the program.

• Main template (project.fig)
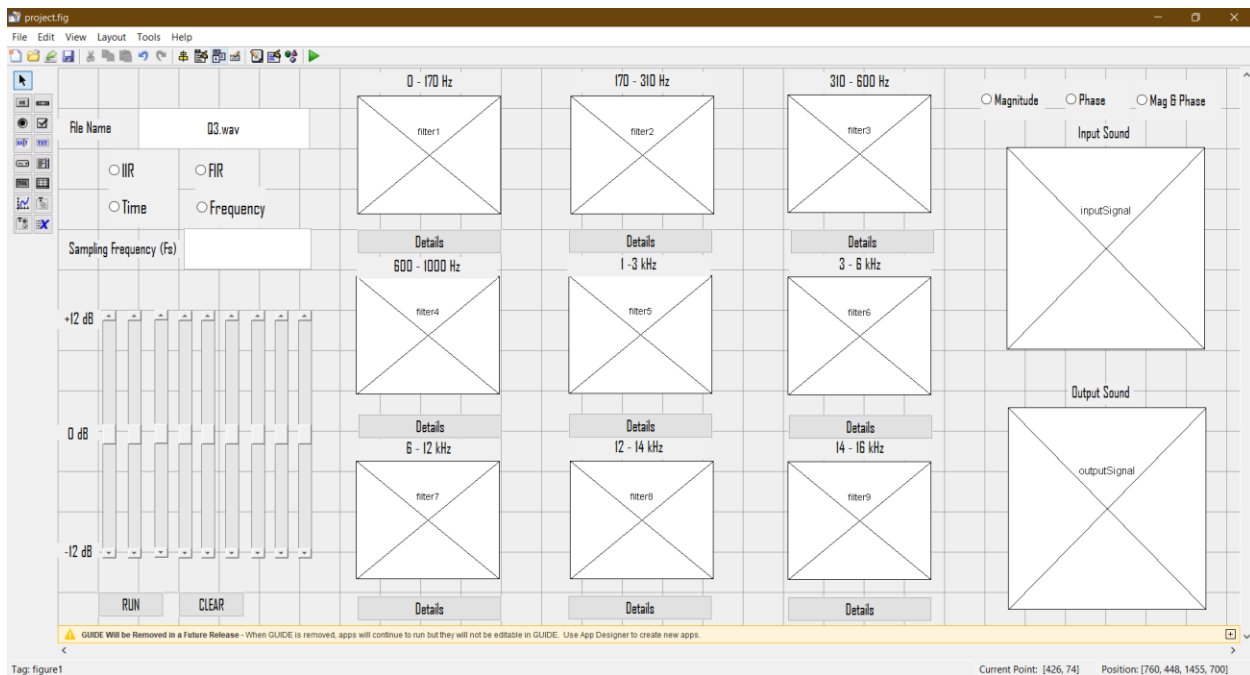
• IIR button

The IIR radio button works in a toggling manner together with the FIR button. When the IIR button (tagged iir) is pressed, it forces the FIR button (tagged fir) to be logic 0. The button is sensed using the get method and is set to a logic 0 if the IIR button is pressed with the set method. Logical value of the button is to be read when the RUN button is pressed (RUN callback function).

• FIR button

The FIR radio button works in the same manner as the IIR button. When the FIR button is pressed, it forces the IIR button to be logic 0.

Logical value of the button is to be read when the RUN button is pressed (RUN callback function).

• Time button

The Time radio button works in a toggling manner together with the Frequency button. The same way IIR button works with the FIR button. When the Time button (tagged timebutton) is pressed, it forces the Frequency button (tagged freqbutton) to be logic 0. Logical value of the button is to be read when the RUN button is pressed (RUN callback function).

• Frequency button

The Frequency radio button works in the same manner as the Time button. When the Frequency button is pressed, it forces the Time button to be logic 0. Logical value of the button is to be read when the RUN button is pressed (RUN callback function).

• RUN button

The run button is responsible of almost all the functions of the program.

The name of the file containing the sound is entered by the user in the specified text field and read in the program using get method which read the string written in the handle with tag 'filepath'.

Then, the audio file is read using audioread method and stored in the vector y with its initial sampling frequency in 'org_fs'.

The user specifies the value of the sampling frequency of the output in the field with tag 'samplingFrequency'. If this field is empty, the sampling frequency of the output will be same as the input, else it is equal to the value entered by the user. In both cases, the vector 'fs' contains the value of the sampling frequency of the output.

In the program, it is required to design 9 filters. These filters can be IIR filter or FIR filters according to the decision made by the user. The decision is made by selecting either the FIR radio button or the IIR one. That's why, in the code, the value of the IIR radio button with tag 'iir' is read and stored in vector 'radio_iir'.

The 9 sliders shown in the figure are used for changing the gain of any of the filters. The range of each is from -12dB to +12dB. The value of each slider is also read and stored in variables (g1-g9). To convert the value of the gain from dB to Watt, the following equation is used:

$$\text{Gain}_{(Watt)} = 10 \wedge (\text{Gain}_{(dB)} / 20)$$

The user can decide whether to plot the outputs in the frequency or time domain. For this purpose, two other radio buttons exists, read and stored in vectors: 'radio_timebutton', 'radio_freqbutton'.

In case of plotting in frequency domain, the user can see the input and the output signals in one of the 3 following ways:

- Plotting the magnitude in frequency domain.
- Plotting the phase in frequency domain.
- Plotting both, magnitude, and phase in the same figure.

That's why 3 radio buttons will appear in case that the user has decided to plot in frequency domain and selected its radio button. The default is the third option, both magnitude and phase. The value of the first two buttons is stored in vectors: 'radio_magbutton', 'radio_phasebutton'.

A variable 'duration' contains the duration of the sound which is equal to number of samples of the input sound / sampling frequency. This variable is used to construct the x-axis for plotting the sound in time domain.

'fmax' is variable containing the maximum frequency allowed according to Nyquist Rule (fs >= 2fmax). To get the normalized frequency the following equation is used:

Normalized frequency (f_n) = f_{cut_off} / f_{max}

For IIR filters the Butterworth approximation is used. To design each filter, butter method is used and takes as input the order, and the cut off frequency of the filter. In case of band pass filters, it is given a vector containing the first and second cut off frequencies.

Here are the nine IIR filters designed:

Filter	Order	Type	Fc$_1$ (Hz)	Fc$_2$ (Hz)
1	6	LPF	170	-
2	4	BPF	170	310
3	4	BPF	310	600
4	2	BPF	600	1000
5	6	BPF	1000	3000
6	10	BPF	3000	6000
7	14	BPF	6000	12000
8	12	BPF	12000	14000
9	10	BPF	14000	16000

As an example of how the butter method is used, let's take the first low pass filter. Its order equals 6 and cut off frequency equals 170 Hz. So, to get the coefficients of the numerator and the denominator of the transfer function of this filter the following line of code is used:

[b1, a1] = butter (N1, 170/fmax)

It must be noted that in IIR filter, butter method takes as input half of the filter's order. (except for low pass filters)

For FIR filters fir1 method is used and takes as input the order, and the cut off frequency of the filter. All the FIR filters have an order of 450.

The difference here that the denominator of the transfer function = 1, because no feedback.

After designing the filters, the coefficients of the numerator of each transfer function are multiplied by the gain entered by the user.

Now, output of each filter is ready to be computed using filter method. Nine outputs are calculated by entering the input sound to each filter.

To get the output sound, all these outputs are just added to each other.

According to the decision of the user, the 9 outputs are plotted either in time or frequency domain.

In time domain the method plot is used for all figures, but before plotting, all the axes of the GUI are cleared using cla command.

In frequency domain, the 9 outputs of the filters are plotted with their magnitude and phase on the same figure. The frequency response of each filter is computed using freqz command, magnitude using abs and phase with angle command. The total input and output sound can be plotted in 3 ways as mentioned before.

Finally, the output sound is written into a file using audiowrite command and played using sound command.

• Magnitude button

The Magnitude button together with the Phase and Mag & Phase buttons are set to be visible when the Frequency button mentioned above is at logic 1 when the RUN button is pressed. The three buttons allow the user to interface the input and output sound signal in the way he prefers. By pressing the Magnitude button, the magnitude spectrum will be showed to the user. The 2 other buttons will be

forced to a logic 0. A press on the RUN button is then needed to trigger a change on the axes.

• Phase button

By pressing the Phase button, the phase spectrum will be showed to the user. The 2 other buttons will be forced to a logic 0. A press on the RUN button is then needed to trigger a change on the axes.

• Mag & Phase button

The default case for the spectra is to show both the magnitude and the phase together on the same axes. So, when the user first chooses the frequency display (By pressing the Frequency button), the Mag & Phase button will be set at a logic 1 and the other 2 buttons set at a logic 0. The user can then choose the viewing option he would like between this and the other 2 options.

• CLEAR button

The CLEAR button is used to reset the interface window to its initial state by clearing all what has been written on it. It initially sets the visibility of the 3 buttons associated with the frequency display off. Then the radio buttons present (IIR, FIR, Time, and Frequency) are given a logic value of 0. The sampling frequency input box is reset to be blank. The gain sliders are then set to the initial value of 0 dB of gain followed by clearing all the axes using the cla method.

- ## Details button

When the Details button (Present beneath each filter output display) is pressed, the user will be prompted to a new window showing him more details regarding that filter.

The template for the window that will be output is shown in the following figure:

- ## Details template (details.fig)
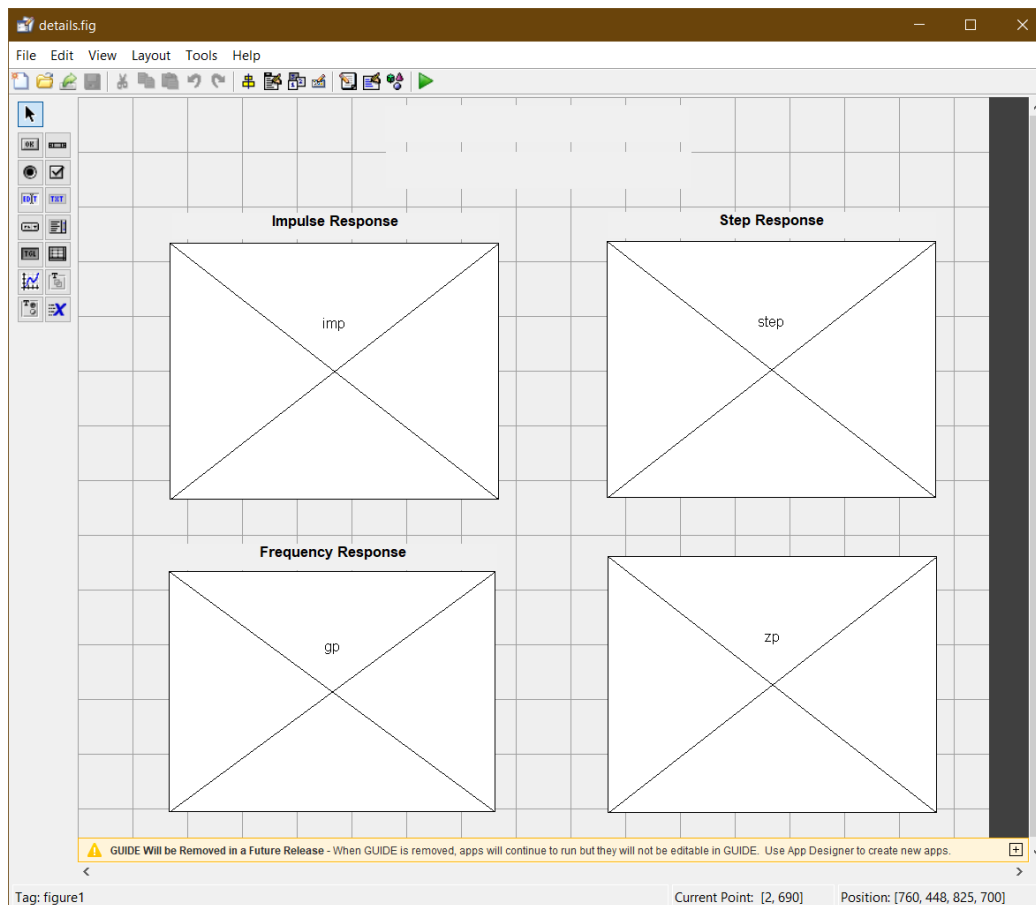
Let us take an instance when the Details button of the second filter is pressed (170 - 310 Hz). All what will be mentioned will happen to all other filters differing only in the values of the arguments passed.

We will use the function setappdata to store the data in the callback function of Details button. That is to share this data between the project.fig UI and the details.fig UI. The arguments passed are (1) a string stating the interval of the filter, in our case '170 - 310 Hz', (2) the order of the filter, (3) the numerator coefficients of the filter transfer function after multiplying by the gain input, (4) the denominator coefficients of the transfer function, (5) the sampling frequency used on the sound file. The .m file of the details.fig UI will then be called after setting all the data mentioned above.

The .m file of the details.fig UI will then retrieve all the arguments passed using the getappdata function. The order and the filter interval will be displayed in the upper interval of the screen. The numerator and the denominator together with the sampling time (1 / Sampling Frequency) will be used as arguments for tf function to create a transfer function model. The impulse response of the system modelled will be stemmed in the axes corresponding to the impulse response using the impz function. The step response will be stemmed using the stepz function in its corresponding axes as well. To display the poles and zeros of the filter, the pzmap function is used on the system. The magnitude response and the phase response of the system is displayed on the same axes with 2 separate Y-axes scaling. The function used for the frequency response is freqz, it returns frequency response vector. We take its magnitude using the abs

function and convert to dB by the formula mentioned in the RUN button callback function, the phase response is known by applying the angle function on the vector. Both responses are then plotted in the axes corresponding to them.
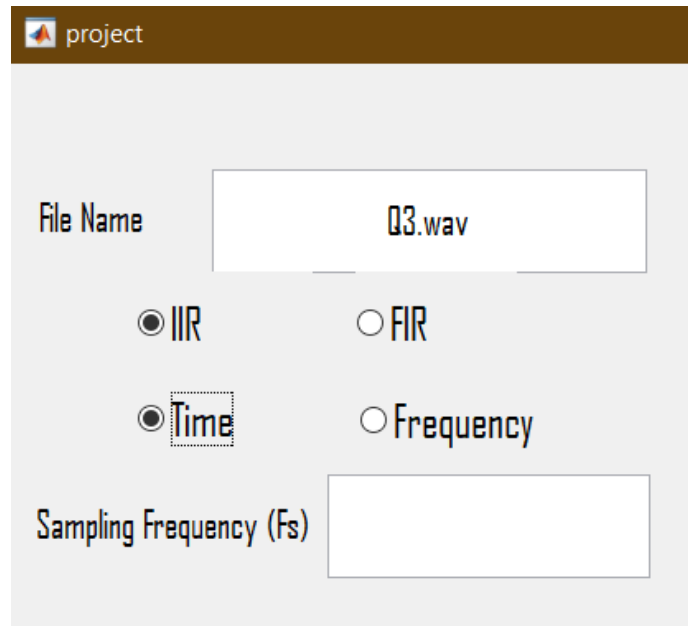
• User Manual and Sample Runs

After opening the program, the user sees this interface:
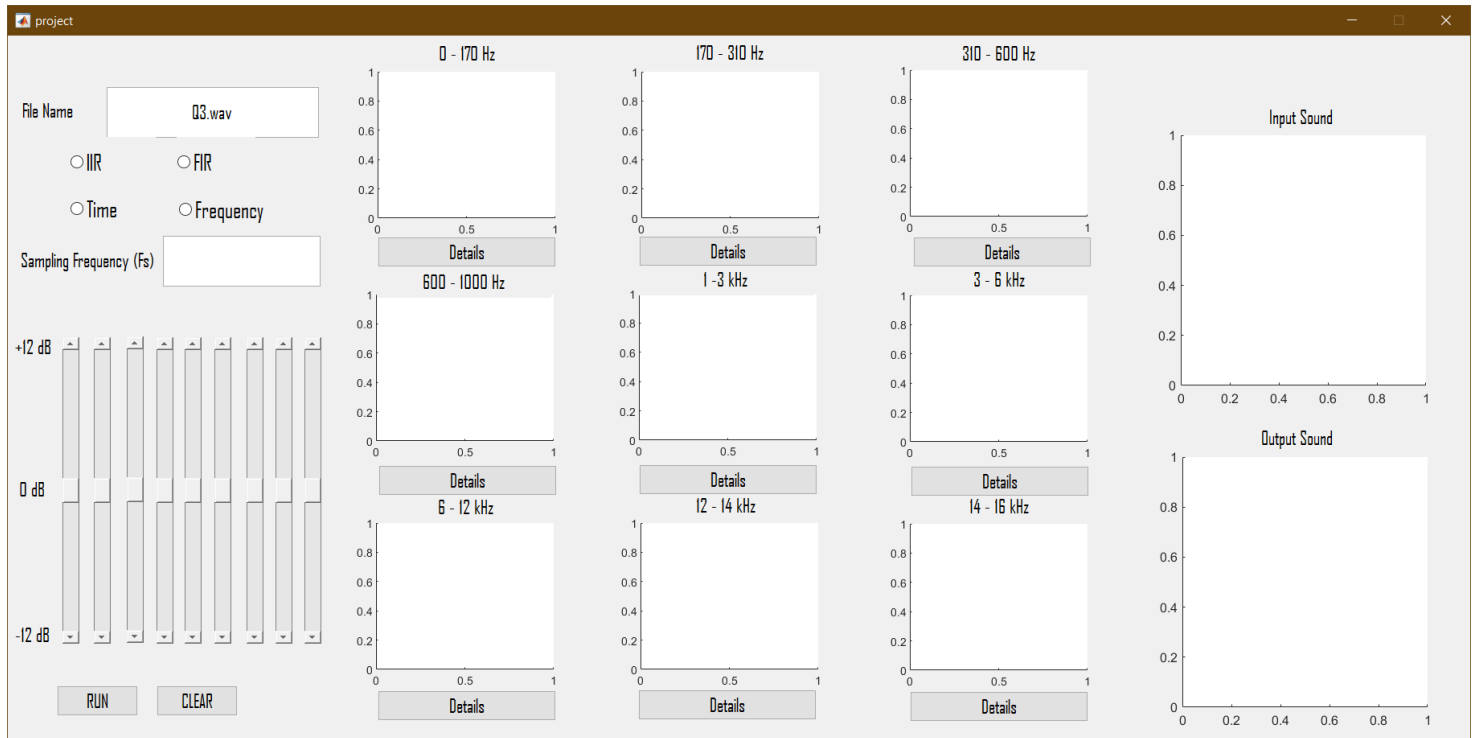
To show time representation of the signal after applying IIR filters, the user has to set IIR and time radio buttons on.
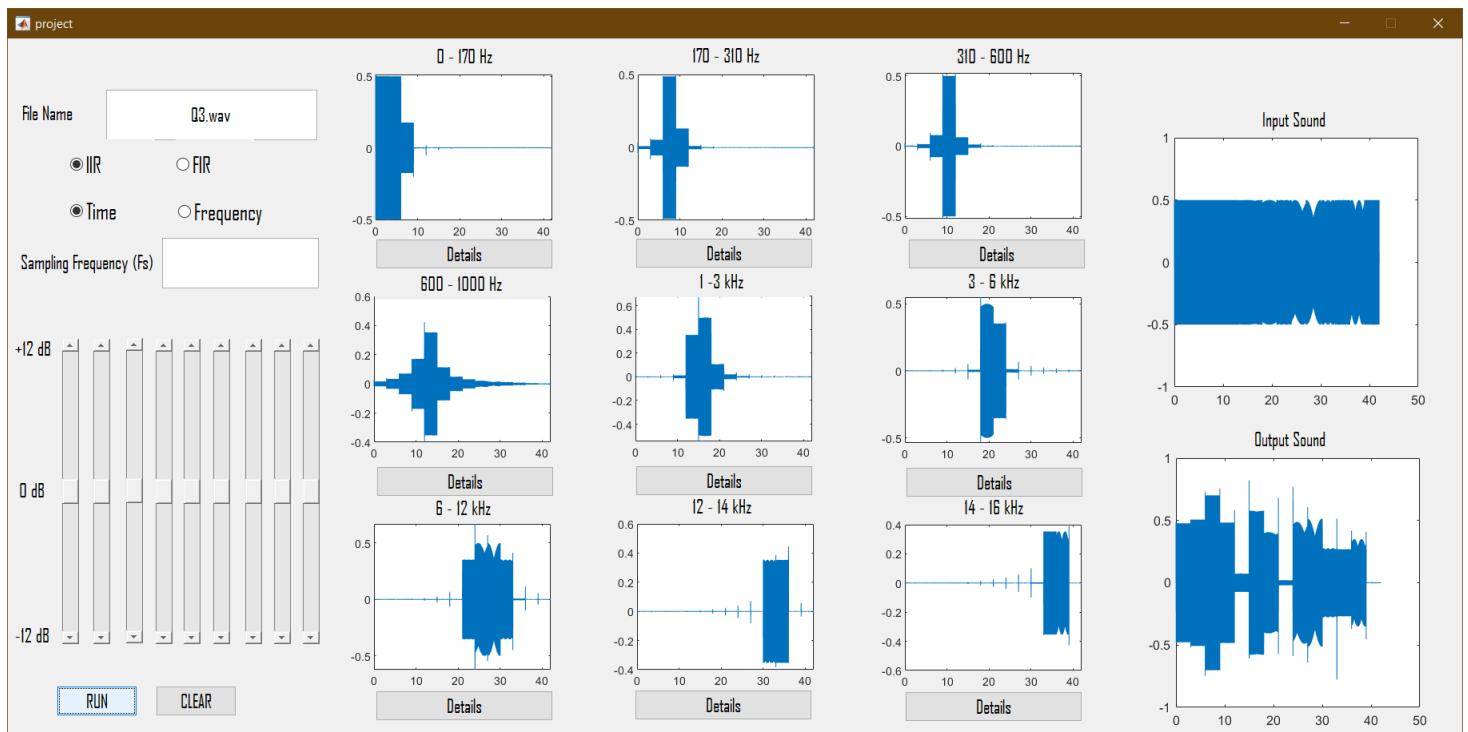
As shown:
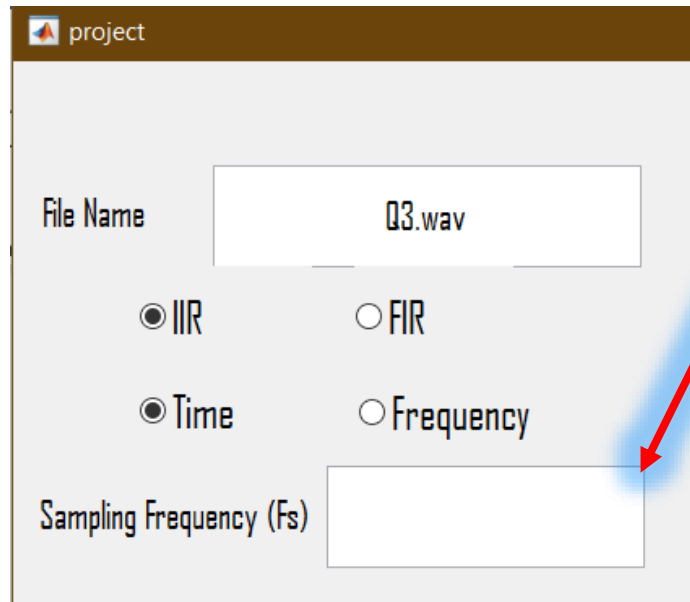
Then click on Run button:

IIR (time domain)

To change the sampling frequency of the output, the user has to write his new fs in the given text box:
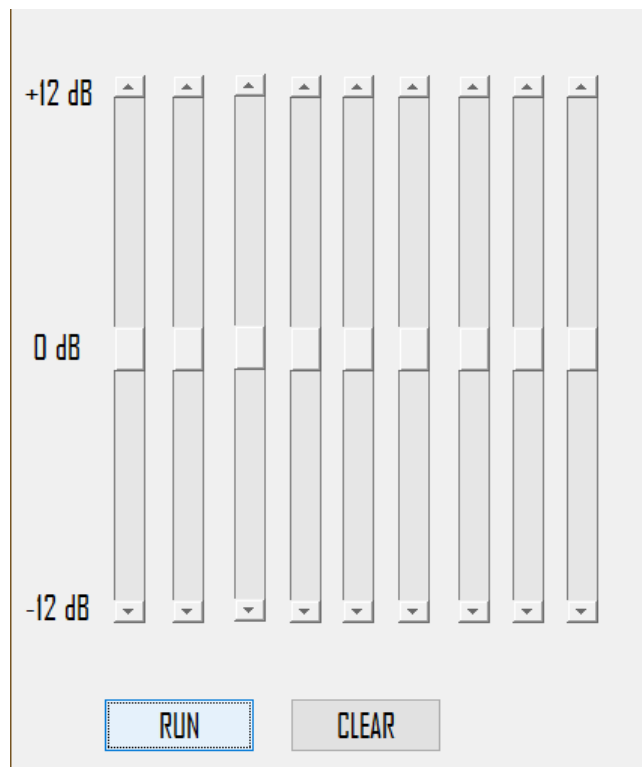
To change the gain of any band by applying the new gain to the used filter for that band you could use these sliders:

Using those ways described you could visualize the output you want by using either IIR filters or FIR filters in frequency or in time domain as shown:

FIR (time domain)
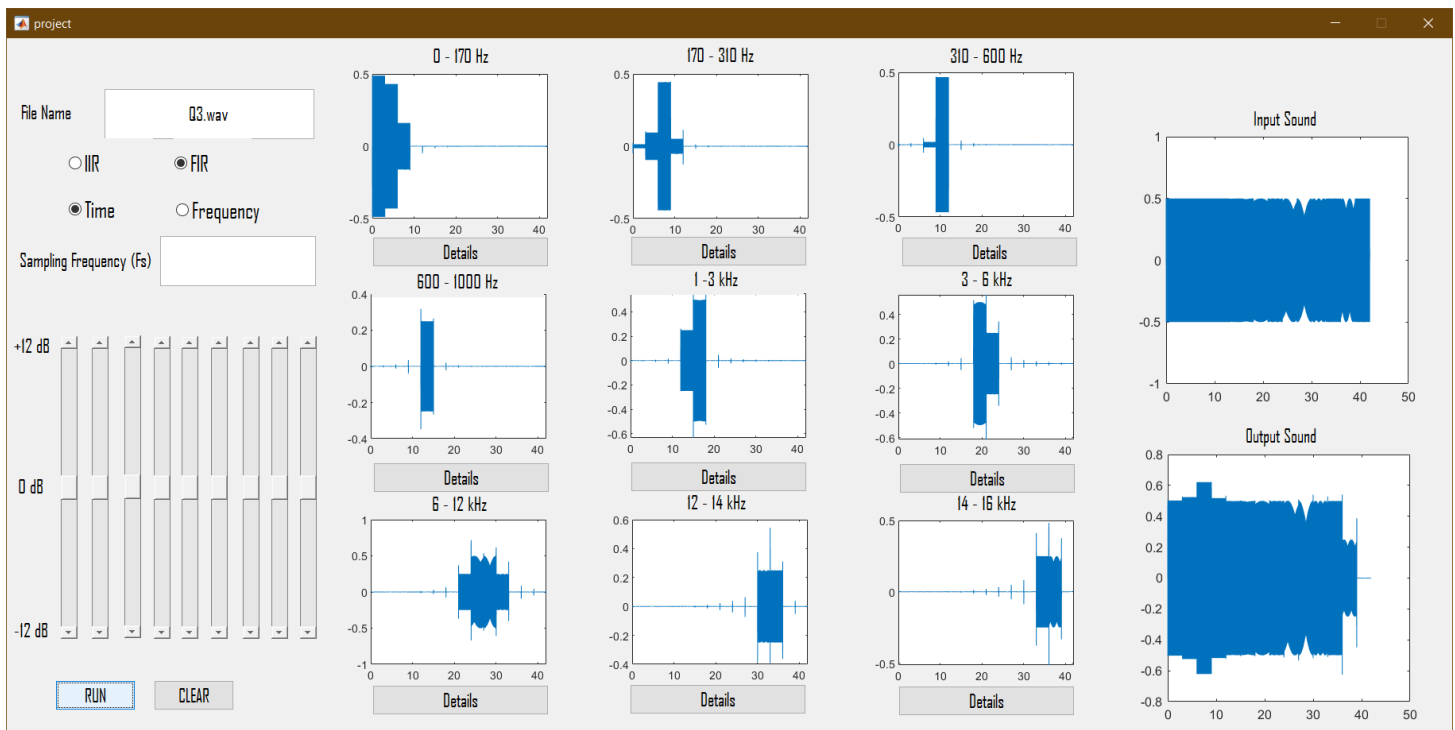
In frequency domain representation, the user will see both magnitude and phase representation of both input and output signals as a default but he can see one of them only if he selected one of these options seen here (Note: These options only appear when frequency button is selected):
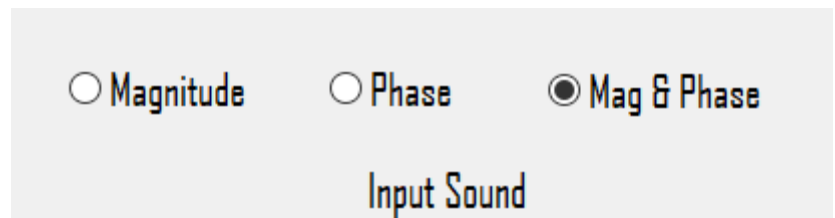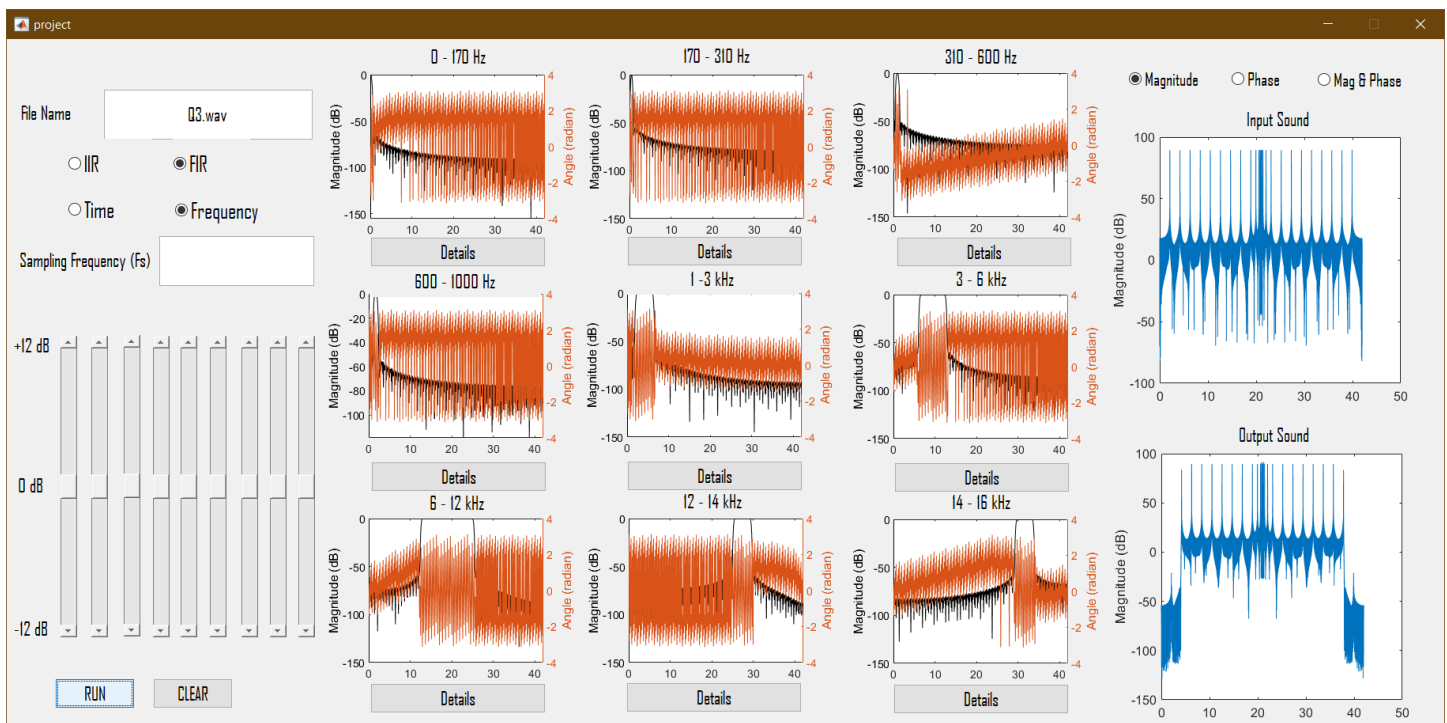

IIR (Frequency domain (magnitude & phase))
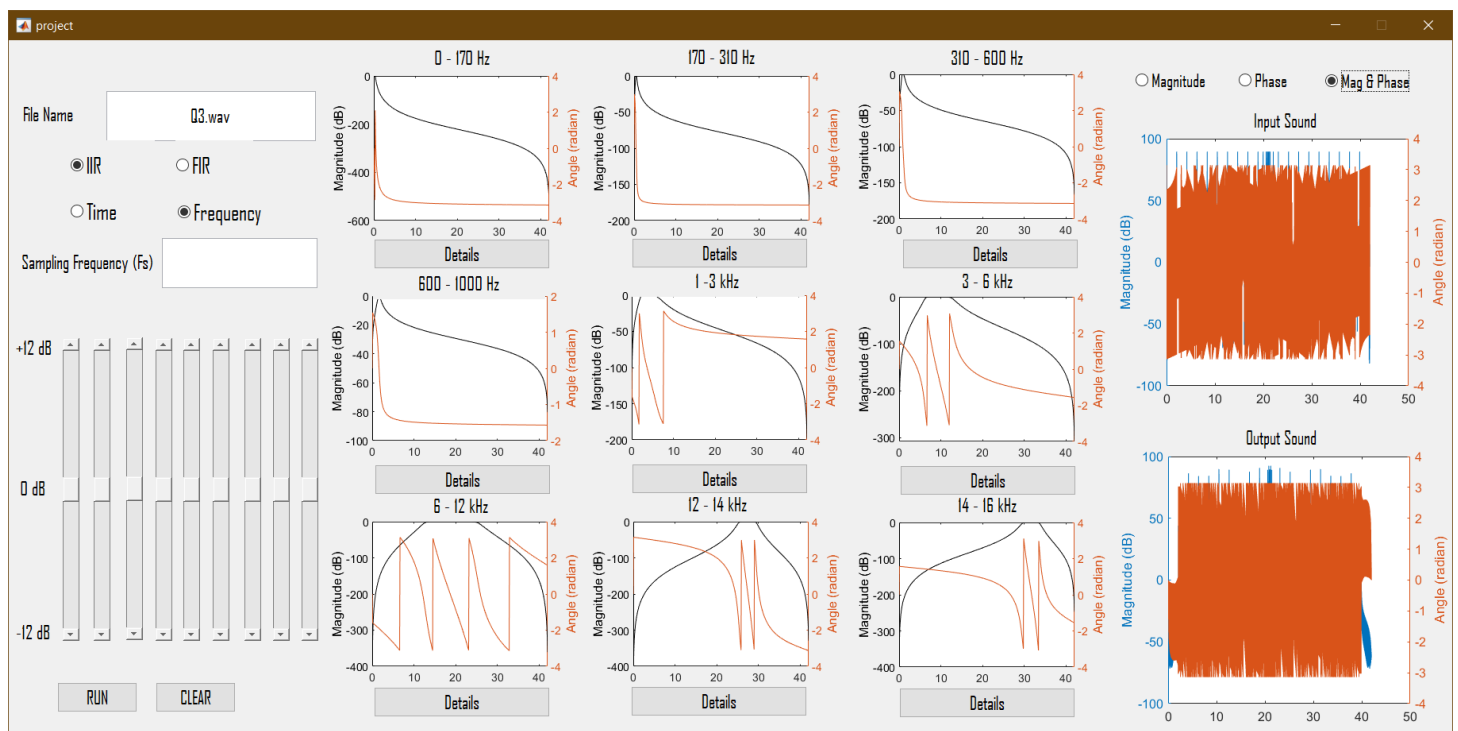
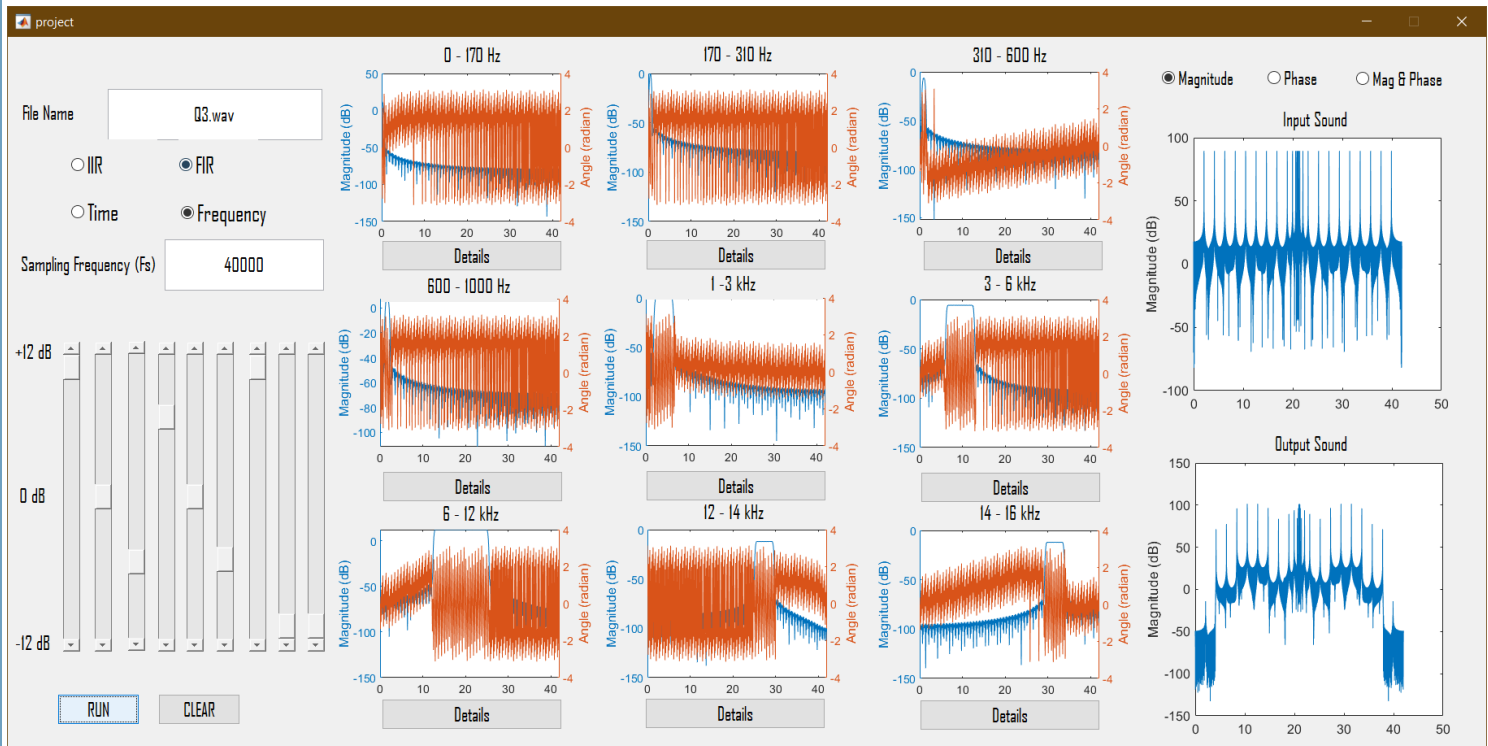# IIR (Frequency domain (phase only))
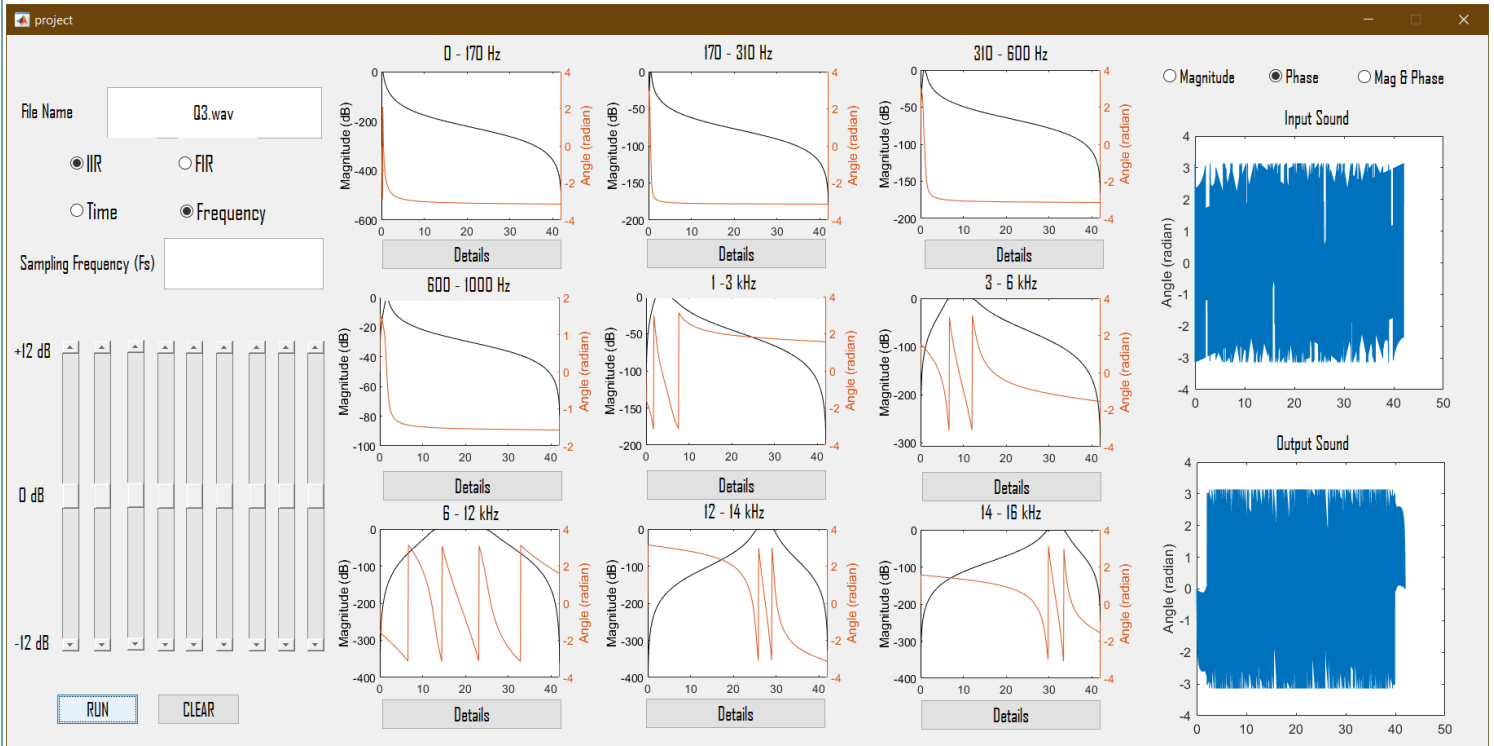
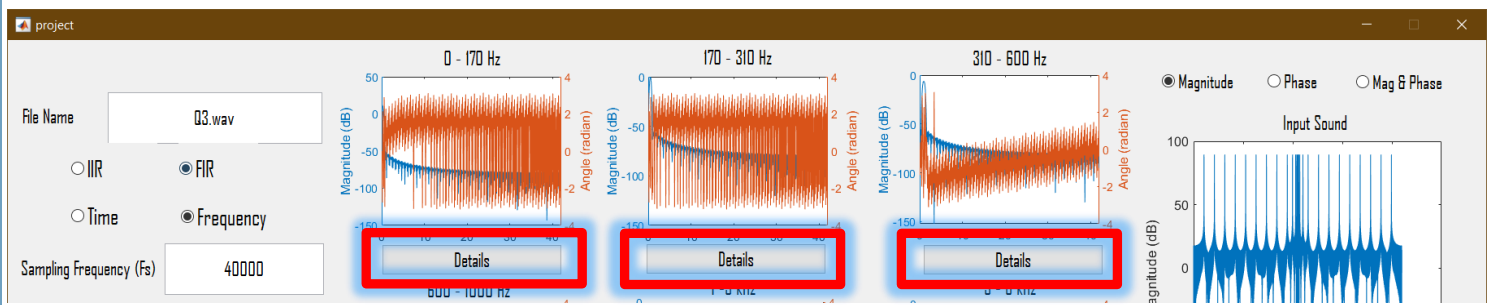# FIR (Frequency domain (magnitude only))
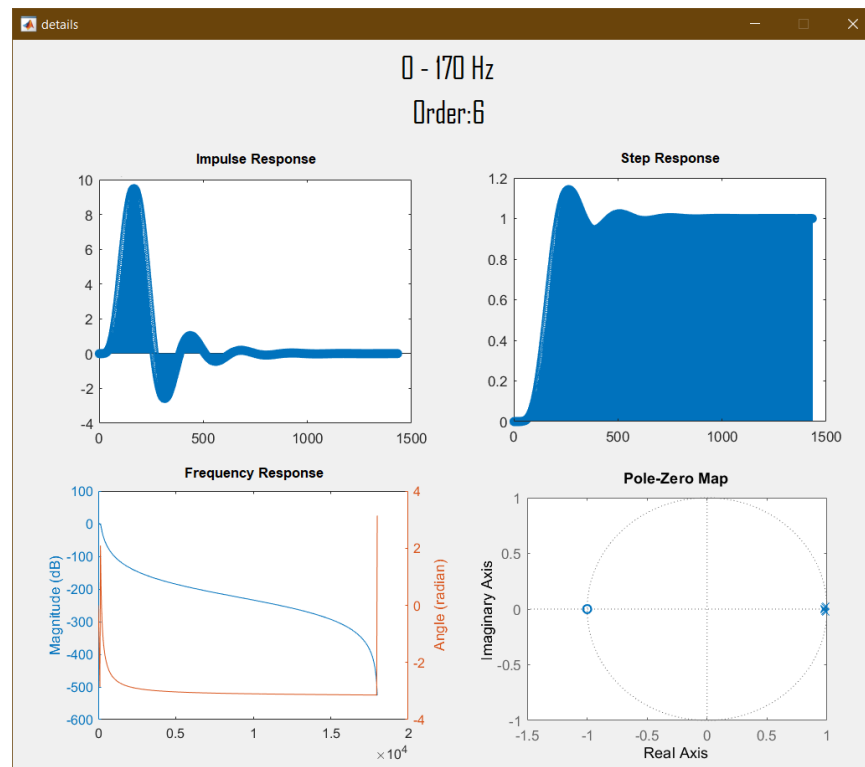
✚ FIR (Frequency domain (magnitude only))

(After changing the gains and the sampling frequency)

To show any filter (Impulse, Step, zeros and poles, Frequency) responses click the button detail under the required filter:

➕ IIR filter for the 1st band (0 – 170 Hz)

➕ FIR filter for the 6th band (3 – 6 kHz)