ABSTRACT

This paper outlines the design of 3-band tone control and 7-band parametric audio equalizers comprised of a cascade network of second order peak and shelf filters, along with their MATLAB code and App Designer implementations. The 3-band tone control was designed with frequencies of 200 Hz (low shelf), 1kHz (mid peak, Q=1) and 5kHz (high shelf). The 7-band parametric equalizer allows the user to change the filter cut-off/centre frequencies and bandwidth. The implementation allows the user to switch between simple and advanced setups and either choose presets or input their own values.

1. INTRODUCTION

Any system that changes the original audio that goes through it and produces an altered output audio signal can be said to be an audio effects or audio FX system. These effects can be applied to analog sound signals through various electronic circuits or instruments, which may be termed as analog audio effect processors. Simulating these analog effects via computers/digital processing is what is commonly known as digital audio effects processing. Digital audio effects are software tools which take input digital audio signals, modify them according to the desired parameters and deliver output digital audio signals [1]. An analog signal can be converted into digital by using an analog-to-digital converter, and back by using a digital-to-analog converter.

Audio equalizers are one of the most common types of frequency processors that are used by many in everyday life [2]. Equalizers are being used in home sound systems, vehicle sound systems, musical instrument amplifiers and processors, studios, live concerts, PA systems etc. Equalizers are used to boost or cut certain parts of the audio frequency spectrum, which changes the way the original audio sounds.

1.1. Types of Equalizers:

Equalizers can be classified into 4 categories – tone control, graphic, console and parametric [2], but for the purpose of this paper, we will only investigate tone control and parametric equalizers. A 3-band tone control is one of the simplest forms of equalizers which allows the user to boost/cut bass, mid and treble of the sound [2]. It is designed using a low shelf, mid peak, and high shelf filters, and can be usually found in home sound systems and guitar amplifiers. A parametric equalizer however usually has more bands and provides more control to the user to fine tune their sound. It has controls for gain, frequency and bandwidth/Q factor connected to each band [2].
An equalizer is made of a combination of several filters such as low-pass, high-pass and band-pass which can attenuate certain frequencies, but nowadays shelf and peak filters are used in order to boost/cut the desired frequency range while leaving the rest of the spectrum unaffected [5]. A 3-band tone control has 1 low shelf, 1 mid peak, and 1 high shelf filters, while a 7-band parametric equalizer usually has 1 low shelf, 5 peak and 1 high shelf filters. These filters may be connected to each other in cascade, parallel or a hybrid network but for the purpose of this paper, we will only investigate cascade equalizers.
1.2. Cascade Equalizers:

Cascade equalizers operate by sending the input audio signal through a series of filters, as shown in Figure 1.3, to produce the equalized output audio signal. In the frequency domain, any output from a system can be represented as the product of the frequency response of the system and the discrete Fourier transform (DFT) of the input. The overall frequency response of a cascade equalizer is the product of its individual filter responses and can be represented by the following mathematical equation [6]:

$$H_C(e^{j\omega T_s}) = G_0 \prod_{m=1}^{M} H_m(e^{j\omega T_s}), \quad (Eq. 1)$$

Where $H_C(e^{j\omega T_s})$ is the frequency response of a cascade graphic equalizer, $G_0$ is a gain factor, $H_m(e^{j\omega T_s})$ are the individual frequency responses of $M$ equalizing filters, $\omega$ is the radial frequency and $T_s$ is the sampling interval in seconds. The output of the equalizer can thus be represented by the following mathematical expression:

$$Y(k) = X(k)H_C(k), \quad (Eq. 2)$$
Where $H_C(k)$ is the DFT of the system response of a cascade graphic equalizer, $X(k)$ is the DFT of the input signal and $Y(k)$ is the DFT of the output signal. The DFT of a signal $x[n]$ can be calculated using the following formula [7]:

$$X(k) = \sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N}, \quad k = 0,1,2,\ldots,N - 1, \quad (Eq. 3)$$

Where $X(k)$ is the DFT of the signal $x[n]$, $x[n]$ is the input discrete signal and $N$ is the number of points for which the DFT is being calculated. For representation of filter signal flow diagrams, the Z transform is used commonly, which can be calculated using the following formula [8]:

$$X(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n}, \quad (Eq. 4)$$

Where, $X(z)$ is the Z-transform of the input $x(n)$ and $z$ is a complex variable. A typical systems response can be represented in Z-domain in the following manner [8]:

$$\frac{Y(z)}{X(z)} = H(z) = \frac{b_0 + b_1z^{-1} + b_2z^{-2} + \cdots + b_Mz^{-M}}{1 + a_1z^{-1} + a_2z^{-2} + \cdots + a_Nz^{-N}}, \quad (Eq. 5)$$

Where, $Y(z)$ is the Z transform of the output $y[n]$, $X(z)$ is the Z transform of the input $x[n]$, $H(z)$ is the Z transform of the system response, $b_0, b_1, b_2 ... b_M$ are called the numerator coefficients, $a_1, a_2 ... a_N$ are called the denominator coefficients and $M$ & $N$ are the number of zeros (number of values for which $H(z)$ is infinite) and poles (number of values for which $H(z)$ is infinite) respectively. The same can be written in discrete domain in the following way [9]:

$$y[n] = b_0x[n] + b_1x[n-1] + b_2x[n-2] + \cdots + b_Mx[n-M] + a_1y[n-1] + a_2y[n-2] + \cdots + a_Ny[n-N], \quad (Eq. 6)$$

2. LAB WORK

The audio equalizer created in this paper provides a choice to the user to use either a 3-band tone control or a 7-band parametric equalizer. Both the tone control and parametric equalizer are made of a cascade network of filters, whose individual magnitude responses are multiplied to give the final magnitude response of the equalizer. For the tone control, the user is only allowed to change the gains of the filters, while in the parametric equalizer, the user can change the gain, frequency, and bandwidth of the filters.

2.1. Filter Design:

Both the equalizer options have been implemented using a cascade of 2nd order shelf and peak filters. The tone control has 2 shelf and 1 peak filters, while the parametric equalizer has 2 shelf and 5 peak filters. The filter coefficients ($a$ and $b$ coefficients of transfer function as in Eq. 5) were calculated using the following formulae [10]:
2\textsuperscript{nd} Order Shelf Filter:

<table>
<thead>
<tr>
<th></th>
<th>$b_0$</th>
<th>$b_1$</th>
<th>$b_2$</th>
<th>$a_1$</th>
<th>$a_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>LF boost</td>
<td>$\frac{1+\sqrt{2}V_0K+V_0^2K^2}{1+\sqrt{2}K+K^2}$</td>
<td>$\frac{2(V_0K^2-1)}{1+\sqrt{2}K+K^2}$</td>
<td>$\frac{1-\sqrt{2}V_0K+V_0^2K^2}{1+\sqrt{2}K+K^2}$</td>
<td>$\frac{2(K^2-1)}{1+\sqrt{2}K+K^2}$</td>
<td>$\frac{1-\sqrt{2}K+K^2}{1+\sqrt{2}K+K^2}$</td>
</tr>
<tr>
<td>LF cut</td>
<td>$\frac{V_0(1+\sqrt{2}K+K^2)}{V_0+\sqrt{2}V_0K+K^2}$</td>
<td>$\frac{2V_0(K^2-1)}{V_0+\sqrt{2}V_0K+K^2}$</td>
<td>$\frac{V_0(1-\sqrt{2}K+K^2)}{V_0+\sqrt{2}V_0K+K^2}$</td>
<td>$\frac{2(K^2-1)}{V_0+\sqrt{2}V_0K+K^2}$</td>
<td>$\frac{V_0-\sqrt{2}V_0K+K^2}{V_0+\sqrt{2}V_0K+K^2}$</td>
</tr>
<tr>
<td>HF boost</td>
<td>$\frac{V_0(1+\sqrt{2}K+K^2)}{1+\sqrt{2}K^2+V_0K}$</td>
<td>$\frac{2(K^2-V_0)}{1+\sqrt{2}K+K^2}$</td>
<td>$\frac{V_0(1-\sqrt{2}K+K^2)}{1+\sqrt{2}K^2+V_0K}$</td>
<td>$\frac{2(K^2-1)}{1+\sqrt{2}K+K^2}$</td>
<td>$\frac{1-\sqrt{2}K+K^2}{1+\sqrt{2}K+K^2}$</td>
</tr>
<tr>
<td>HF cut</td>
<td>$\frac{V_0(1+\sqrt{2}K+K^2)}{1+\sqrt{2}K+K^2}$</td>
<td>$\frac{2V_0(K^2-1)}{1+\sqrt{2}K+K^2}$</td>
<td>$\frac{V_0(1-\sqrt{2}K+K^2)}{1+\sqrt{2}K+K^2}$</td>
<td>$\frac{2(V_0K^2-1)}{1+\sqrt{2}K+K^2}$</td>
<td>$\frac{1-\sqrt{2}V_0K+V_0^2K^2}{1+\sqrt{2}K+K^2}$</td>
</tr>
</tbody>
</table>

Figure 2.1: Filter coefficients for 2\textsuperscript{nd} order shelf filter, where LF denotes low frequency and HF denotes high frequency [10].

2\textsuperscript{nd} Order Peak Filter:

<table>
<thead>
<tr>
<th></th>
<th>$b_0$</th>
<th>$b_1$</th>
<th>$b_2$</th>
<th>$a_1$</th>
<th>$a_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boost</td>
<td>$\frac{1+\frac{1}{2}K+K^2}{1+\frac{1}{2}K+K^2}$</td>
<td>$\frac{2(K^2-1)}{1+\frac{1}{2}K+K^2}$</td>
<td>$\frac{1-\frac{1}{2}K+K^2}{1+\frac{1}{2}K+K^2}$</td>
<td>$\frac{2(K^2-1)}{1+\frac{1}{2}K+K^2}$</td>
<td>$\frac{1-\frac{1}{2}K+K^2}{1+\frac{1}{2}K+K^2}$</td>
</tr>
<tr>
<td>Cut</td>
<td>$\frac{1+\frac{1}{2}K+K^2}{1+\frac{1}{2}V_0K+K^2}$</td>
<td>$\frac{2(K^2-1)}{1+\frac{1}{2}V_0K+K^2}$</td>
<td>$\frac{1-\frac{1}{2}K+K^2}{1+\frac{1}{2}V_0K+K^2}$</td>
<td>$\frac{2(K^2-1)}{1+\frac{1}{2}V_0K+K^2}$</td>
<td>$\frac{1-\frac{1}{2}V_0K+K^2}{1+\frac{1}{2}V_0K+K^2}$</td>
</tr>
</tbody>
</table>

Figure 2.2: Filter coefficients for 2\textsuperscript{nd} order peak filter [10].

\[ K = \tan \left( \frac{\pi f_c}{f_s} \right), \quad (Eq. 7) \]
\[ V_0 = 10^{G/20}, \quad (Eq. 8) \]
\[ Q = \frac{f_c}{BW}, \quad (Eq. 9) \]

Where, $f_c$ is the cut-off frequency for shelf filter or centre frequency for peak filter, $f_s$ is the sampling frequency and $G$ is the gain of the filter in dB. Q is the quality factor, $f_c$ is the centre frequency and BW is the bandwidth of the peak filter.

Figure 2.3: Signal flow diagram for a 2\textsuperscript{nd} order IIR (infinite impulse response) filter along with the input-output relationship [11].

For 3-band tone control, the following parameters are fixed – $f_c$ is 200Hz for low shelf, $f_c$ is 1kHz & $Q$ is 1 for mid peak, and $f_c$ is 5kHz for high peak. The filter coefficients are calculated based on user inputs or presets for the corresponding parameters in the MATLAB code.
2.2. Equalizer Functioning:

Taking Eq. 1 into consideration, the equalizer was designed to calculate the individual magnitude responses of each constituent filter and multiply them to find the equalizer’s overall magnitude response, which is then applied to the input signal. In MATLAB, this was achieved by first finding the individual frequency responses using the function ‘freqz’ using the calculated filter coefficients as the inputs, calculating the overall equalizer response by multiplying all filter responses, and then using ‘invfreqz’ function to find the filter coefficients of the equalizing filter thus created and applying the function ‘filter’ with these coefficients and the input signal as inputs to create the equalized output signal.

The equalizer has the option of using a simple (3-band tone control) or advanced (7-band parametric equalizer) versions with 15 pre-sets to choose from or a custom pre-set where the user can input the parameters themselves. The equalizer created, as mentioned above, acts basically as another filter, where the input and output relation can be derived from Figure 2.3.

2.3. MATLAB Implementation:

The equalizer has been implemented as a set of functions divided into two parts broadly for simple and advanced equalization. The core of the equalizer is the calculation of various filter
coefficients and finding the overall equalizer’s filter coefficients from them. The functions for these operations have been based on Eq. 1, 7, 8 & 9 and Figure 2.1 & 2.2. The broad process flow of the implementation has been shown in Figure 2.4.

Figure 2.5: Audio Equalizer app in Advanced setting with pre-set ‘Jazz’ selected and the magnitude response display set to Filter (to display individual filter magnitude responses).

The final MATLAB script finds the audio file in MATLAB’s current folder and asks the user to choose any audio file, choose between simple and advanced equalizer types and choose the pre-set for the equalizer. It then calls the relevant equalizer function, displays a comparison between the FFT of input and processed audio files channel-wise, displays the individual and combined filter magnitude responses, plays back the input and processed audio signals and finally saves the processed audio file in either the input audio file’s format or ‘.wav’ format.

In App Designer, the algorithm has been implemented as a standalone desktop application named “Audio Equalizer”. The UI has options to choose presets, switch the equalizer on or off, switch between individual and combined filter magnitude responses, toggle between simple (3-band tone control) and advanced (7-band parametric equalizer) setups, visually change values for gain, frequency and bandwidth, view gain of the audio during playback, load and save audio, and control audio playback.
3. DISCUSSION & CONCLUSION

The designed cascade 3-band tone control and 7-band parametric equalizers created work as expected. The equalizers successfully equalize the input audio signal and the output can be easily verified by comparing the magnitude responses of input and output signals, and by listening to both audio signals. The designed 2\textsuperscript{nd} order filters work pretty well over the frequency range of 20Hz-20kHz, though the peak filters’ magnitude response starts degrading at lower frequency and bandwidth values. The MATLAB function ‘invfreqz’ behaves erratically when the gain for any filter approaches zero; as a corrective measure, any gain value less than 0.01 dB was approximated to zero. The gain meter in the app was implemented to start after a delay of 0.7s to account for the time lag between the command to start audio playback and the playback actually starting. Since the algorithm is only capable of non-real time processing, all features are locked in the app during playback, until the playback ends completely or is stopped using the ‘Stop Audio’ button. The responsivity of the app is dictated by the processing power of the machine being used to run it.
4. REFERENCES


