ECSE512 Digital Signal Processing

Fall 2020 Term Project Report

Multi-Band Digital Audio Tone Control Unit Design using Dual Recursive Running Sum Filters

Yuanzhe Gong

260677071

yuanzhe.gong@mail.mcgill.ca

Yi Zhu

260716006

yi.zhu6@mail.mcgill.ca

Abstract:

Due to the non-linear phase in the infinite impulse response (IIR) filter and high computational complexity in the finite impulse response (FIR) filter, the audio tone control unit in the multimedia systems is usually implemented in analogue form. However, the analogue form tone control introduces extra costs in the system of analogue and digital transformation. In this report, several papers related to digital audio tone control unit with a FIR filter structure and dual recursive running sum (DRRS) algorithm are reviewed. Furthermore, a three-band tone control unit is designed in MATLAB based on the transfer function provided in the reference. A five-band tone control unit is implemented further to investigate the DRRS structure in the tone control unit, and a MATLAB based application is developed for a better user-friendly interface. As a result, an attenuation of more than -30dB on the sidelobe level and a linear phase is achieved. Wave samples and filter samples are tested and added in the Appendix.

Contents

1. Introduction..2
2. Background 3
3. Proposed Approach 7
3.1 Five-Band DRRS Tone Control Algorithm 7
3.2 MATLAB Code Implementation...9
3.2.1 Calculate Filter Length based on Input Band Cutoff Frequencies 9
3.2.2 Dual Recursive Running Sum Function..9
3.2.3 Five-Band Equalizer 9
3.2.4 Main Function and Input 10
3.3 PC Interface..10
4. Results and Discussion..11
5. Conclusion 14
References...15
Appendix I 16
Appendix II..17
Appendix III 19
Appendix IV 22

1. Introduction

Nowadays, audio system applications have been broadly deployed in the daily life of people. In general, a standard audio system can be decomposed into five separate units responsible for five consecutive functions: 1) acquisition of source signals; 2) storage of audio signals; 3) signal processing; 4) signal transmission; 5) output reproduction [1]. As audio signals are electronic representations of sound waves traveling in the air, different audio effects, such as synthesis, noise-canceling, and tone changing, could be carried out through the audio signal processing by various kinds of manipulating. The implementation of tone control function in the audio signal processing significantly benefits audio users by allowing them to boost or attenuate the frequency of a specific audio signal to an ideal value to achieve an ideal outcome. For example, a tone control system can help speakers or headphones to compensate for the inadequate bass response to make the output melody optimized. The process of tone control, also known as equalization, is accomplished by increasing or cutting off specific frequency ranges or band of the input audio signal. A typical tone control unit, also known as an equalizer, applies a hardware filter onto the input audio signal to realize the frequency range modification.

Due to high computational complexity and difficulties, the tone control unit in most audio systems is still in analogue form even though audio signals are generally implemented in digital format [2]. Therefore, an extra function unit that transfers signal between analogue form and digital form is needed, and this unit can introduce a lot of additional cost and function limitations caused. To break the bottleneck caused by these analog systems, a digital tone control system is required. Besides, by implementing all function units in digital forms, the processing performance of audio systems will also be considerably enhanced.

When choosing the types of fundamental filter type in the tone control unit, a linear phase in operation is essential for the audio system. For example, in stereo playback, the phase information provides a sense of the relative position of the sources, and the linear phase can also manipulate all frequencies by the same or a relative amount, thereby maximally preserving waveshape and original information. Due to this fact, infinite impulse response (IIR) filters, which usually have a nonlinear phase, are not suitable for the tone control units. Therefore, the realization of the digital audio tone control function becomes the search for an efficient finite impulse response (FIR) filter that can achieve low computational complexity in the digital audio tone control process.

Several researchers have proposed implementing a FIR filter with various extra techniques to achieve the linear phase operation in the tone control system with low complexity. In [3], a linear phase FIR filter system is implemented in an audio tone control system using the frequency-response masking technique. The low arithmetic complexity of the FIR filter system at the cost of an increase in the number of delay units is realized by using a multiplication-free fourth-order low-pass FIR filter. Moreover, in [4], a new structure, namely recursive running sum structure (RRS), is proposed. The utilization of RRS on linear phase signal processing reduces the arithmetic operation, which can further lower the cost of implementation for the audio system. The RRS filter has a very efficient filter structure as it only requires two summations for one input. However single RRS filter does not provide enough attenuation on the side lobe level. Therefore, two RRSs can be cascaded to form a dual recursive running sum (DRRS) structure to further enhance the filter performance by attenuating the sidelobe level to -30dB or more.

In this report, we illustrate our proposed design of a pure digital audio tone control unit using a multi-band FIR filter with a DRRS technique based on the MATLAB platform. The background information of the RRS and DRRS filter will be first introduced. Related methodologies and formulas will also be given. The details of our proposed design will then be demonstrated in the next section. Experimental testing and results will then be discussed in the fourth section, while instructions for using the proposed application and extra equalizer samples are placed in the Appendix section.

2. Background

The RRS filter, an efficient implementation of a simple FIR digital filter with unity coefficients and performs only two additions to carry out output, has an excellent noise-removal performance with low computational cost [4, 5]. To get the formula of the RRS filter, we first consider a linear phase FIR filter with the following general z-domain transfer function:

$$H(z) = \sum_{i=0}^{K} z^{-i} \tag{1}$$

To simplify this transfer function, which reveals the sum of a geometric series, and avoid complex additions, equation 1 can be then written as:

$$H(z) = \frac{1 - z^{-(K+1)}}{1 - z^{-1}} \tag{2}$$

The equation 2 shows a wide known RRS structure. By replacing the $K + 1$ term by filter length L_1, the formula of the RRS filter in the z-domain is then given by:

$$F(z) = \frac{1 - z^{-L_1}}{1 - z^{-1}} \tag{3}$$

By taking the inverse z-transform, we have the formula in the time-domain as:

$$y(n) = x(n) + y(n-1) - x(n - L_1) \tag{4}$$

From equation 4 we can see an RRS filter requires one adder, one subtracter, one-unit delay, and a L_1 samples delay. The hardware implementation of the RRS filter can be obtained from equation 4 and is shown in Fig.1.

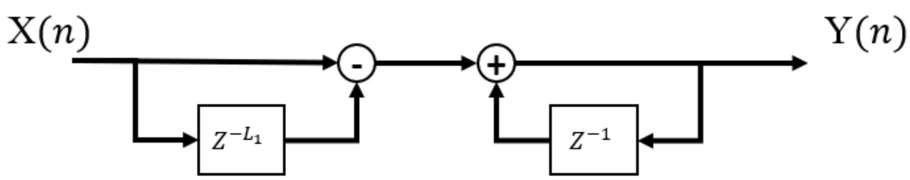


Fig. 1 Hardware realization of RRS filter

The first deep null position of the RSS filter will be created at frequency f_{null}, which is at:

$$f_{null} = \frac{1}{TL_1} \quad (5)[6]$$

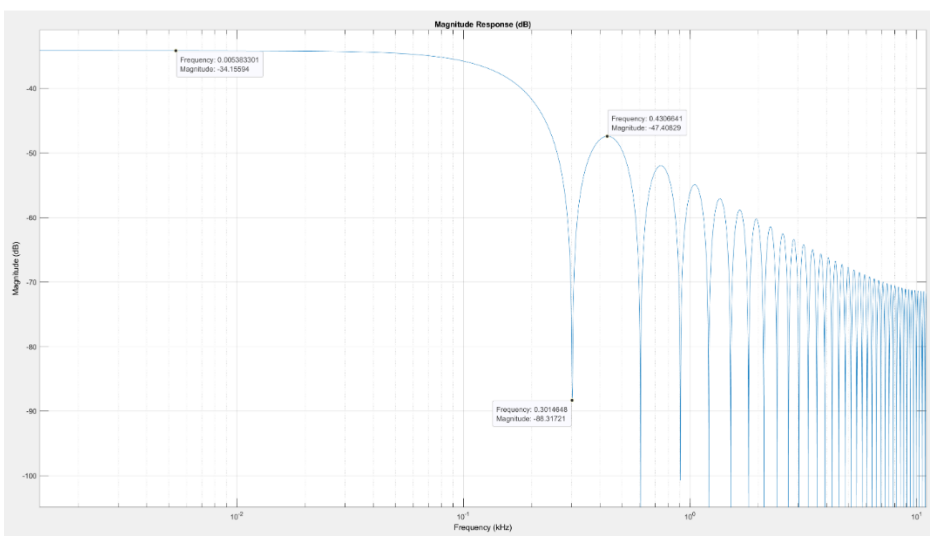


Fig. 2 Magnitude response of a low-pass filter (with single RRS filter structure)

However, a single RRS filter can only achieve about -13dB attenuation at the cutoff on the highest side lobe, which is too large in the audio tone control system [6]. To test the performance of a single RRS filter, a low-pass filter is implemented in MATLAB, and the magnitude response is shown in Fig. 2. It has been observed that the attenuation between the low-frequency band and the highest sidelobe level is -13dB, which result is consistent with results in [6].

Moreover, as its compact and simple nature, the RRS filter is considered to be an ideal candidate for a prefilter. Thus, a dual RRS filter design composed of one RRS filter with coefficient L_1 and another cascaded RRS filter with length coefficient L_2 is proposed. Cascaded RRS filter can provide an increased stopband attenuation, and the sensitivity to coefficient quantization will also be reduced [7]. The formula of the cascaded RRS filter in the z-domain is similar as shown in equation 3, given L_1, a L_2 is chosen as:

$$L_2 = int\left\{\frac{L_1}{\sqrt{2}}\right\} \tag{6}$$

Where $int\{.\}$ refers to the closest integer value to the result. The z-domain transfer function cascaded filter is shown in equation 7.

$$H(z) = \frac{1 - z^{-L_1}}{1 - z^{-1}} * \frac{1 - z^{-L_2}}{1 - z^{-1}} \tag{7}$$

Thus, the based DRRS system has the frequency response as shown:

$$F\left(e^{j\pi fT}\right) = \frac{L_1 L_2 \sin(\pi f T L_1)\sin(\pi f T L_2)}{[\sin(\pi f T)]^2} e^{-j2\pi fT(L_1+L_2-2)} \tag{8}$$

This DRRS filter is also implemented in MATLAB, and the magnitude response is shown in Fig. 3. It can be found that the attenuation between the low passband and highest sidelobe level is improved to more than -30dB.

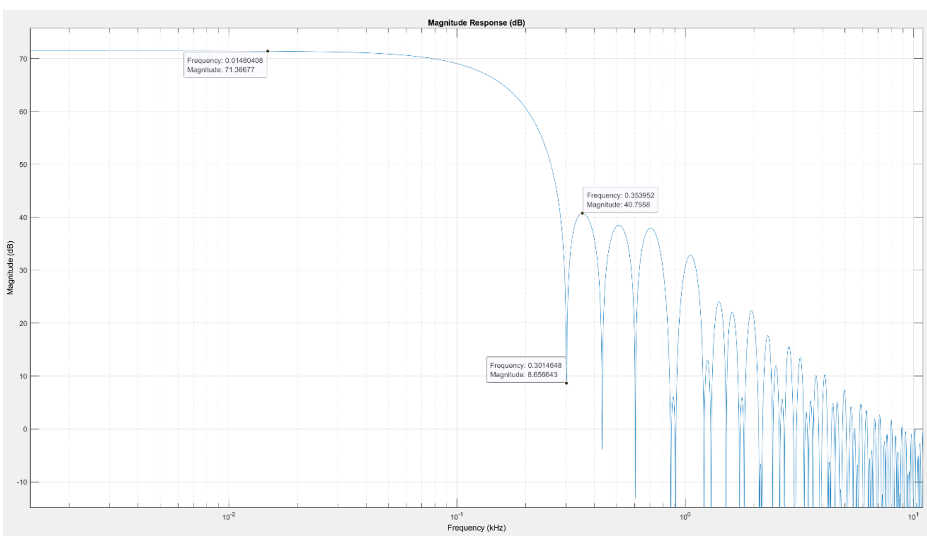


Fig. 3 Magnitude response of a DDRS filter (with DRRS filter structure)

Previous researchers have also proposed the implementation of DRRS in audio tone control systems. A detailed example is shown in [2], where the digital tone control unit consists of one high-pass $DRRS(H1, H2)$ and one low-pass $DRRS(L1, L2)$, and the pass band can be achieved by shifting the high and low pass filter and use $DRRS(H1, H2)$-$DRRS(L1, L2)$. The transfer function of the overall tone control filter network is given by:

$$\begin{aligned} F_{net} &= F_H(z) + F_B(z) + F_L(z) \\ &= G_H z^{-[(L_1+L_2)/2]+1}\left[z^{-[(H_1+H_2)/2]+1} - k_H DRRS(H_1, H_2)\right] \\ &- +\left[k_H z^{-[(L_1+L_2)/2]+1} DRRS(H_1, H_2) - k_L z^{-[(H_1+H_2)/2]+1} DRRS(L_1, L_2)\right] \\ &+ G_L k_L z^{-[(H_1+H_2)/2]+1} DRRS(L_1, L_2) \end{aligned} \tag{9}$$

In terms of linearity, three delay blocks are implemented to ensure the same phases of the adders. A simple logic circuit of the complete system is shown in Fig. 4.

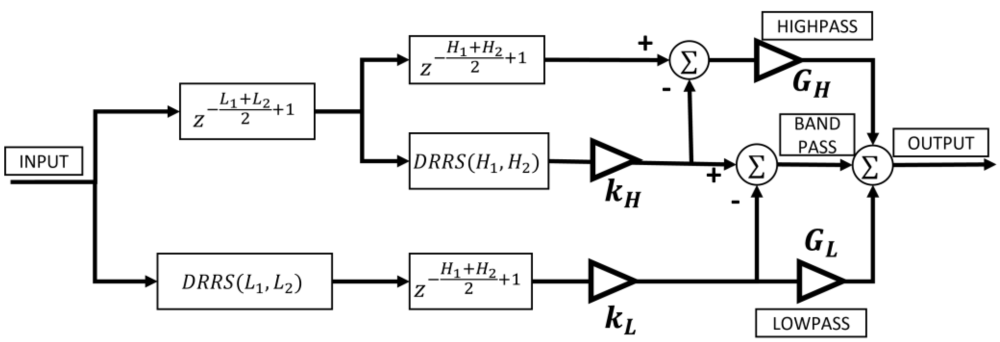


Fig. 4 Logical circuit of three-band DRRS filter implemented for audio tone control

Where k_L and k_H are used to compensate the DC gain to unity,

$$k_L = \frac{1}{L_1 L_2} \, and \, k_H = \frac{1}{H_1 H_2} \tag{10}$$

The values H_1, H_2, L_1, and L_2 are chosen based on the high-pass and low-pass cutoff frequencies and the sampling frequency. G_H and G_L are two gains in which values can be adjusted by the user to achieve various high-frequency and low-frequency gain of the system. Besides, from the equation 9 above we can also see that when $G_L = 1 \, and \, G_H = 1$, we will get:

$$F_{net} = z^{-[(L_1+L_2+H_1+H_2)/2]+2} \tag{11}$$

Therefore, no frequency response distortion is produced when both high-frequency gain (G_H) and low-frequency gain (G_L) are chosen to be 1. The three-band equalizer function is implemented in MATLAB file named *threebandequalizer.m*.

3. Proposed Approach

3.1 Five-Band DRRS Tone Control Algorithm

Our proposed design allows for five-band equalization with adjustable gains in dB for each frequency band. Users will be able to boost or attenuate selected frequencies.

With the DRRS filter, the sidelobe is attenuated to -30dB. Similar to the idea of the three-band tone control unit, we expand the tone control unit to five bands. Five delay blocks are added in five branches to ensure the same phase of the adders. To avoid a half-delay unit, $(L_1 + L_2)$, $(B_1 + B_2)$, $(B_{11} + B_{22})$, and $(H_1 + H_2)$ should be even number. The Scale factor $k = \frac{1}{filterLength1*filterLength2}$ is added behind the DRRS filter to scale the DC gain to unity. Five separate gains are added in five bands to enable users to control the gain level of each band. Fig. 5a shows the block logical circuit of our proposed five band DRRS filter and Fig. 5b shows the band diagram and the position of user input cutoff frequency of the proposed method.

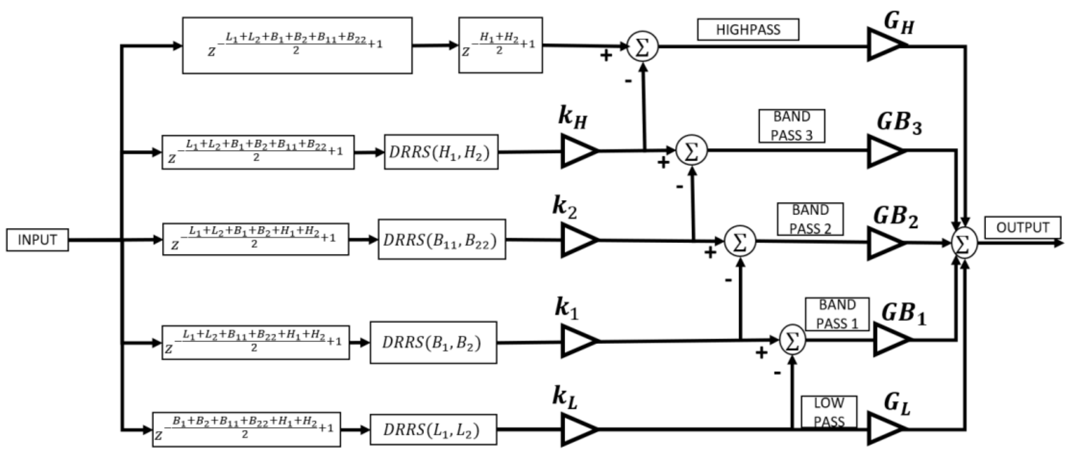


Fig. 5a Logical circuit of the five-band DRRS filter implemented for audio tone control

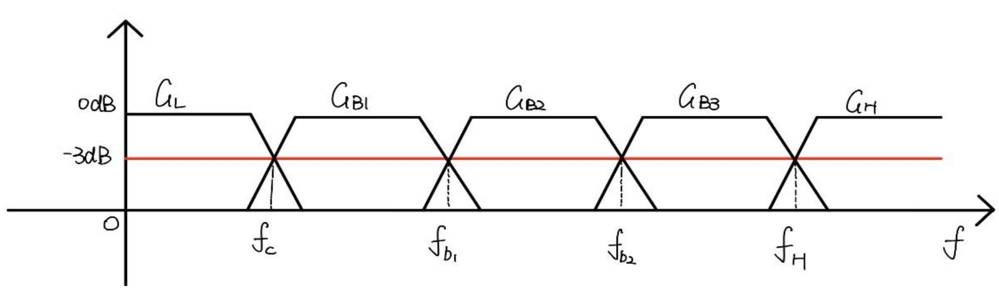


Fig. 5b Band diagram of five-band filter with $G_H = G_{B3} = G_{B2} = G_{B1} = G_L = 0dB$

As shown in Fig. 5a and Fig. 5b, after defining four cutoff frequencies where the target nulls exist, five-band can be achieved by following phase shifting and DRRS filter:

$$F_{net} = F_H(z) + F_{B3}(z) + F_{B2}(z) + F_{B1}(z) + F_L(z) \tag{12}$$

$$F_H(z) = G_H * z^{-\left[\frac{L_1+L_2+B_1+B_2+B_{11}+B_{22}}{2}\right]+1} * \left[z^{-[(H_1+H_2)/2]+1} - k_H * DRRS(H_1, H_2)\right] \tag{13}$$

$$F_{B3}(z) = G_{B3} * \left[k_H * z^{-\left[\frac{L_1+L_2+B_1+B_2+B_{11}+B_{22}}{2}\right]+1} * DRRS(H_1, H_2) - k_2 * z^{-\left[\frac{L_1+L_2+B_1+B_2+H_1+H_2}{2}\right]+1} * DRRS(B_{11}, B_{22})\right] \tag{14}$$

$$F_{B2}(z) = G_{B2} * \left[k_2 * z^{-\left[\frac{L_1+L_2+B_1+B_2+H_1+H_2}{2}\right]+1} * DRRS(B_{11}, B_{22}) - k_1 * z^{-\left[\frac{L_1+L_2+B_{11}+B_{22}+H_1+H_2}{2}\right]+1} * DRRS(B_1, B_2)\right] \tag{15}$$

$$F_{B1}(z) = G_{B1} * \left[k_1 * z^{-\left[\frac{L_1+L_2+B_{11}+B_{22}+H_1+H_2}{2}\right]+1} * DRRS(B_1, B_2) - k_L * z^{-\left[\frac{B_1+B_2+B_{11}+B_{22}+H_1+H_2}{2}\right]+1} * DRRS(L_1, L_2)\right] \tag{16}$$

$$F_L(z) = G_L * \left[k_L * z^{-\left[\frac{B_1+B_2+B_{11}+B_{22}+H_1+H_2}{2}\right]+1} * DRRS(L_1, L_2)\right] \tag{17}$$

$F_H(z), F_{B3}(z), F_{B2}(z), F_{B1}(z),$ and $F_L(z)$ are one high-pass filter (13), three bandpass filters (14-16), and one low-pass filter (17), respectively. The phase shift before the DRRS is added to assure a linear phase in the adder. When gains of all band equal to 1, $G_H = G_{B3} = G_{B2} = G_{B1} = G_L = 1\ or\ 0dB$,

$$F_{net} = z^{-\left[\frac{L_1+L_2+B_1+B_2+B_{11}+B_{22}+H_1+H_2}{2}\right]+2} \tag{18}$$

Therefore, there is no distortion after applying the filter to the input signal. The filter is plotted in MATLAB when the gains are set to unit gain.

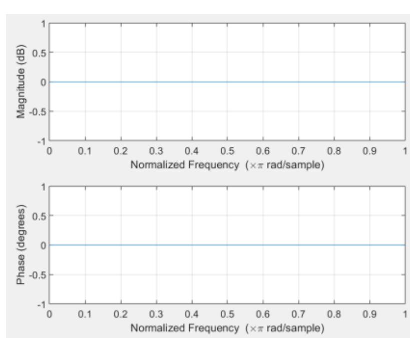


Fig. 5c Magnitude and phase response plot with $G_H = G_{B3} = G_{B2} = G_{B1} = G_L = 1$

As shown in Fig. 5c, the magnitude response plot is a straight line at 0dB (which represents a unit gain), and the phase response is also a straight line at zero degrees.

3.2 MATLAB Code Implementation

3.2.1 Calculate Filter Length based on Input Band Cutoff Frequencies

```
function y=getodd(x)
y = 2*floor(x/2)+1;
end

function bandLength=solveLength(fc,fs)
bandLength=getodd(1/(fc*2/fs));
%bandLength=getodd(1/(fc/fs));
end
```

Fig. 6 MATLAB code for filter length calculation

There are two functions to calculate the filter length, the first one is *solveLength()* which used the equation 6 to find the nearest odd integer value, which can be used as the order of the FIR filter and *getodd()* is the auxiliary function to make sure the value found is the closest odd integer so that the sum of $-\frac{L_1+L_2}{2}+1$ can be assured to be an integer.

3.2.2 Dual Recursive Running Sum Function

```
function result1=Drrs(L1,L2)
syms z
  result1=(1-z^(-L1))/(1-z^(-1))*(1-z^(-L2))/(1-z^(-1));
end
```

Fig. 7 MATLAB code for DRRS function

After first filter length parameter L_1 is found, the L_2 is determined by $L_2 = getodd(\frac{L_2}{\sqrt{2}})$. The DRRS function cascade two low-pass recursive running filter with order L_1 and L_2. In z-domain, the two cascade systems are multiplied with each other and the frequency response of the total DRRS function is shown in equation 7.

3.2.3 Five-Band Equalizer

The five-band equalizer consists of one low-pass filter with filter length L_1 and L_2, one high-pass filter with length H_1 and H_2, and three band pass filter between. The input gains are transferred from dB scale to linear scale.

```matlab
function result2=Equalizer(L1,B1,B11,H1,GL,GB1,GB2,GB3,GH)
syms z
L2=getodd(L1/sqrt(2))
H2=getodd(H1/sqrt(2))
B2=getodd(B1/sqrt(2))
B22=getodd(B11/sqrt(2))
kH=1/(H1*H2);
kL=1/(L1*L2);
k1=1/(B1*B2);
k2=1/(B11*B22);
% change the log scale gain input to linear scale
GL=10^(GL/20);
GB1=10^(GB1/20);
GB2=10^(GB2/20);
GB3=10^(GB3/20);
GH=10^(GH/20);
result2=GH*z^(-[L1+L2+B1+B2+B11+B22]/2+1)*[z^(-[H1+H2]/2+1)-kH*Drrs(H1,H2)]...
        +GB3*[kH*z^(-[L1+L2+B1+B2+B11+B22]/2+1)*Drrs(H1,H2)-k2*z^(-[H1+H2+B1+B2+L1+L2]/2+1)*Drrs(B11,B22)]...
        +GB2*[k2*z^(-[H1+H2+B1+B2+L1+L2]/2+1)*Drrs(B11,B22)-k1*z^(-[H1+H2+L1+L2+B11+B22]/2+1)*Drrs(B1,B2)]...
        +GB1*[k1*z^(-[H1+H2+L1+L2+B11+B22]/2+1)*Drrs(B1,B2)-kL*z^(-[H1+H2+B1+B2+B11+B22]/2+1)*Drrs(L1,L2)]...
        +GL*kL*z^(-[H1+H2+B1+B2+B11+B22]/2+1)*Drrs(L1,L2);
end
```

Fig. 8 MATLAB code for the five-band equalizer

To make sure that all parts of the filter have a linear phase, an extra delay (or phase shift) is added in front of each DRRS filter as shown in equation (12-17). When all bands have unit gain, $F_{net}(z) = z^{-\frac{L_1+L_2+B_1+B_2+B_{11}+B_{22}+H_1+H_2}{2}+2}$, thus no frequency distortion is produced. In addition, amplifier parameter k added to scale the DC gain of the DRRSs to unity for each band case where $k = \frac{1}{filterLength1*filterLength}$.

3.2.4 Main Function and Input

The *audioread()* function is utilized to read the amplitude and sampling frequency from the audio file. Then the filter generated by the five-band equalizer will be applied to the input data. The input signal will be convolved with impulse response of the design filter. The result filtered signal will be written to a target audio file using the *audiowrite()* function.

In order to show the five-band equalizer graphically, Fast Fourier Transform will transfer the original signal and filtered signal into the frequency domain, and the data will be plotted in the frequency domain in our application. Complete MATLAB code implemented in this five-band tone control system is attached separately and illustrated in Appendix IV as well.

3.3 PC Interface

Using MATLAB App Designer software, we implemented the DRRS algorithm into a five-band digital audio tone control application with adjustable gains (in dB). The developed application consists of a user interface where users could input the desirable WAV format audio signal file and manipulate essential parameters, such as cutoff frequencies and gains for different passes. The application will then modify the input audio signal accordingly and generate the filtered audio signal file in WAV format. Based on user's desire, the visualized original signal, filtered signal, or both could be plotted in frequency domain linear or dB scale and saved to the local destination. This application can be installed and run-on MATLAB software. A screenshot of the developed application is shown in Fig. 9a, while detailed instructions for using this user-friendly program are illustrated in Appendix I.

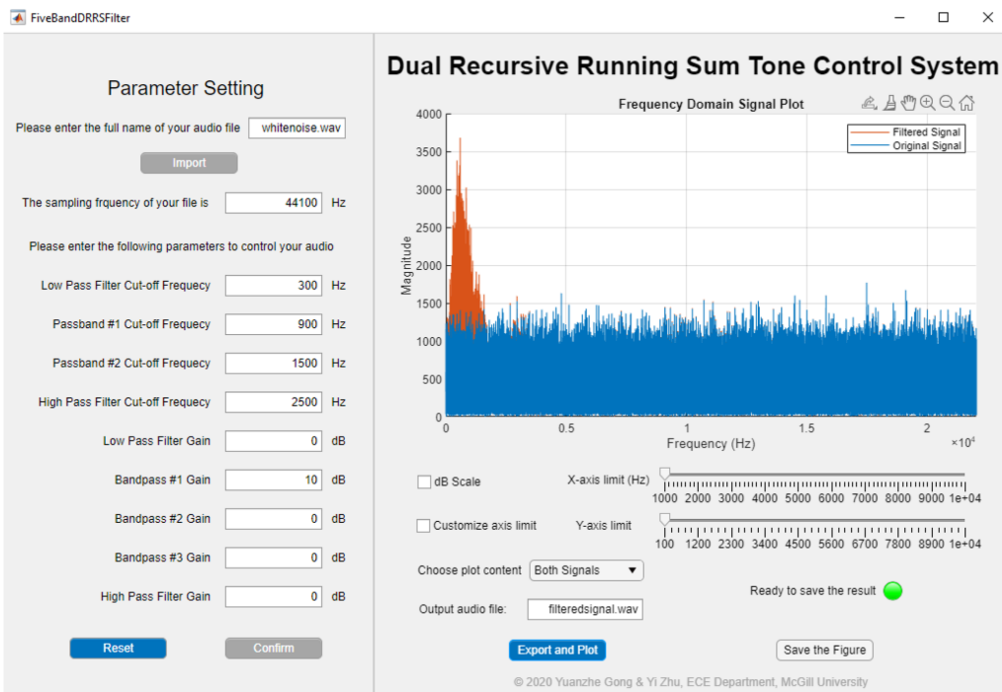


Fig. 9a MATLAB-based five-band equalizer interface

4. Results and Discussion

As shown in Fig. 9a, a white noise audio file is processed using the proposed dual recursive running sum tone control system. The low-pass filter cutoff frequency is set to be 300Hz, while the high-pass filter cutoff frequency is set to be 2500Hz. Two passband cutoff frequencies are set to be 900Hz and 1500Hz, respectively. The second passband gain is modified to be 3, while others remain 1. The result filtered signal (orange) and the original signal (blue) is shown in Figure 10a. It can be observed that the band from 300Hz to 900Hz is amplified almost 3 times and other bands of the filtered signal almost overlap with the original signal. The rest of test samples are shown in Appendix II.

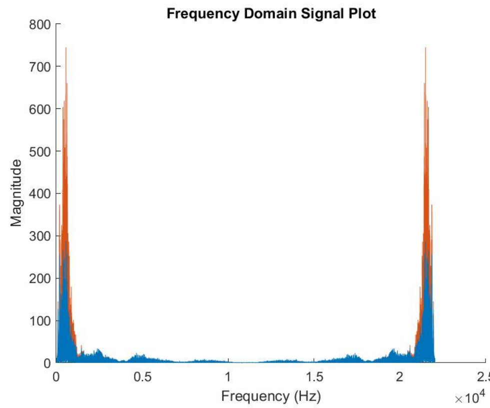
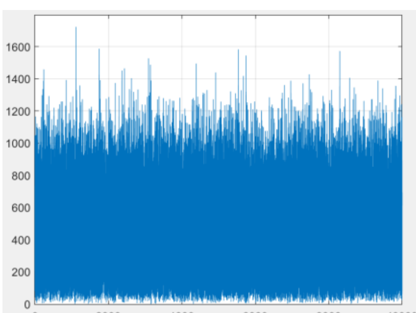
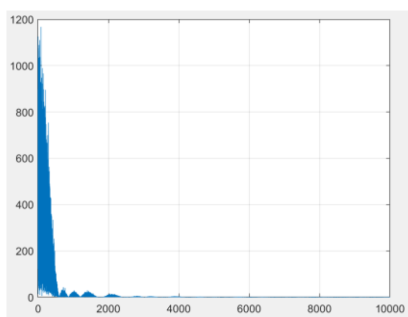


Fig. 10a Signal plot of sample1.wave in linear scale with parameters: $f_s = 22050Hz, f_L = 300Hz, f_{B1} = 900Hz, f_{B2} = 1500Hz, f_H = 2500Hz, (G_L, G_{B1}, G_{B2}, G_{B3}, G_H) = (0,10,0,0,0)$

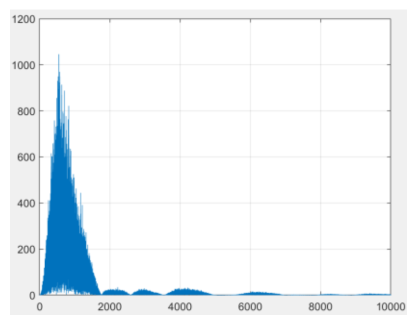
To show the manipulations of all five bands clearly, one set of five-band equalizer samples is shown in Fig. 10b. A white noise wave file is used as an input file, and the set of cutoff frequency is set to $\{300, 900, 1500, 2500\}Hz$, and in each test case, only one of five bands has a unit gain, and all other gains are set to zero (which should eliminate the signal). The amplification and cutting off shift between five bands can be observed.



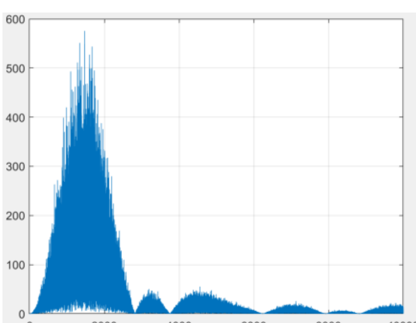
White-noise input sample



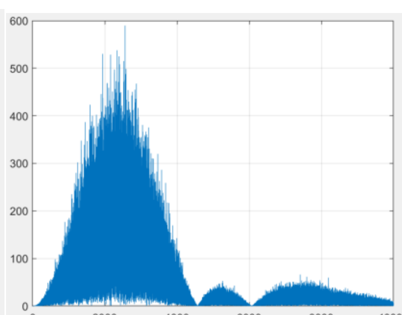
$(G_L, G_{B1}, G_{B2}, G_{B3}, G_H) = (1,0,0,0,0)$



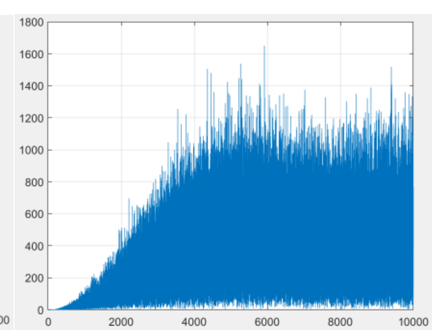
$(G_L, G_{B1}, G_{B2}, G_{B3}, G_H) = (0,1,0,0,0)$



$(G_L, G_{B1}, G_{B2}, G_{B3}, G_H) = (0,0,1,0,0)$



$(G_L, G_{B1}, G_{B2}, G_{B3}, G_H) = (0,0,0,1,0)$



$(G_L, G_{B1}, G_{B2}, G_{B3}, G_H) = (0,0,0,0,1)$

Fig. 10b Filtered output signal sample with five band equalizers in linear scale

To test the filter design process, a DRRS low pass filter is designed and plotted in Fig. 11a, where $(G_L, G_{B1}, G_{B2}, G_{B3}, G_H) = (1,0,0,0,0)$ dB is shown. The rest of test results are shown in the Appendix III. It can be observed that in all test cases the linear phase is achieved, that the frequency responses of all five cases are straight line. Ripples exist in the stop band area. The change in the transition band is relatively slow which means the width of transition band is large and adjustment in one band can influence other bands. The wide transition band will cause the leakage of amplification which means when the user changes the gain of one band, the bands next to it will also be changed. At the same time, the pass band is not wide and flat enough, which can potentially make the unequal attenuation in target bands.

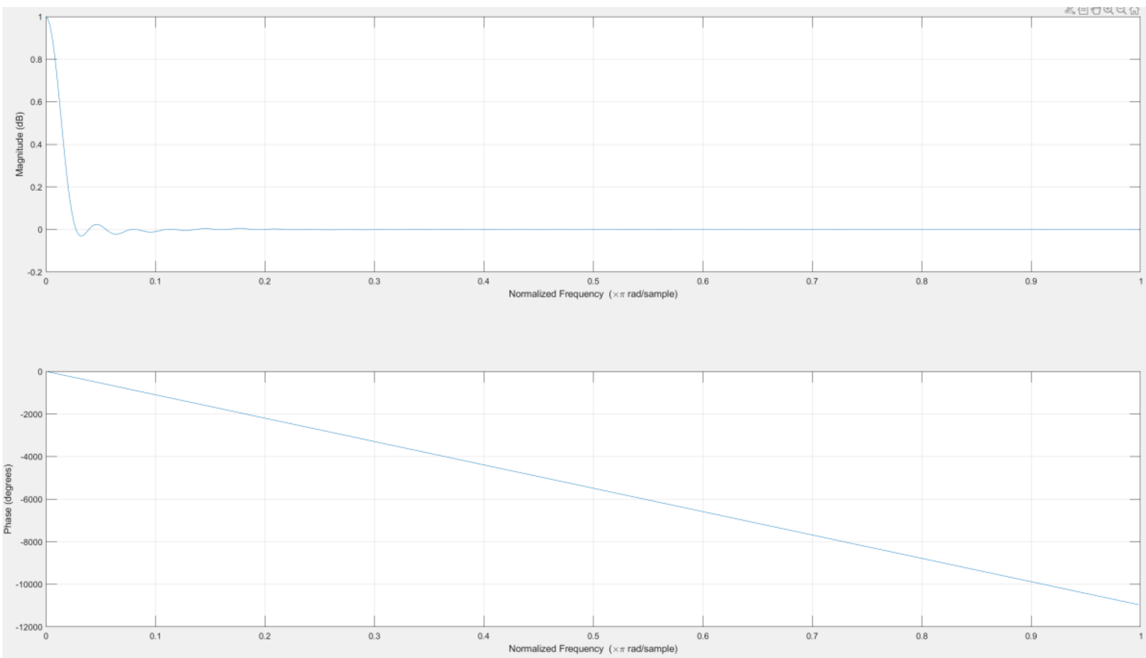


Fig. 11a Filter Magnitude and Frequency Response with input $(G_L, G_{B1}, G_{B2}, G_{B3}, G_H)$ = (1,0,0,0,0) dB

Since in the proposed approach, the whole band is divided into five separate bands, the length of each pass band will be relatively narrow. Bands will influence with each other more serious. This interaction will cause another issue, that the target band cannot achieve the set gain level. In our test cases, the three passband's gain will be lower than the set value, and the difference depends on the input cutoff frequencies and gains.

5. Conclusion

Same as the author claimed in [2], the DRRS algorithm offers users a choice of FIR filter with low computation cost and complexity. This algorithm can be easily implemented and adjusted for different scenarios. A linear phase and a unit DC gain can be achieved by adding phase shift and DC gain compensation coefficient before the filter. By cascading two FIR filter, a -30dB attenuation between the target band and highest side lobe level is achieved. A five-band equalizer based on this DRRS algorithm is implemented and tested. Besides, an easy-use MATLAB-based user interface (Application) has also been created. The user can easily tune the tone by giving the cutoff frequencies and target band gain.

At the same time, we found that DRRS filter has a quite wide transition band, and this fact will cause different bands interact with each other. At the same time, the three middle pass bands are not flat and wide enough. These issues become very serious when the whole band is divided into five and each path band has a relative narrow width. This fact makes the accurate control of audio bands impossible. To improve the five-band equalizer, some more advanced FIR filter with narrow transition band and wider stable pass band should be implemented.

References

[1] P. Berkhout and L. Eggermont, "Digital audio systems," in *IEEE ASSP Magazine*, vol. 2, no. 4, pp. 45-67, October 1985.

[2] R. H. Yang, S. B. Chiah and W. Y. Chan, "Design and implementation of a digital audio tone control unit using an efficient FIR filter structure," *Proceedings of Digital Processing Applications (TENCON '96)*, Perth, WA, Australia, 1996, pp. 273-277.

[3] Y. Lian and Y. C. Lim, "Linear-phase digital audio tone control using multiplication-free FIR filter," *Journal of the Audio Engineering Society,* vol. 41, no. 10, pp. 791-794, 1993.

[4] Yong Lian and Poh Choo Ho, "ECG noise reduction using multiplier-free FIR digital filters," *Proceedings 7th International Conference on Signal Processing, 2004. Proceedings. ICSP '04. 2004.*, Beijing, 2004, pp. 2198-2201 vol.3.

[5] H. Patel, S. Trivedi, R. Neelkanthan and V. R. Gujraty, "A Robust UART Architecture Based on Recursive Running Sum Filter for Better Noise Performance," *20th International Conference on VLSI Design held jointly with 6th International Conference on Embedded Systems (VLSID'07)*, Bangalore, 2007, pp. 819-823.

[6] Rong-Huan Yang, "Linear-phase digital audio tone control using dual RRS structure," in *Electronics Letters*, vol. 25, no. 5, pp. 360-362, 2 March 1989.

[7] J. Adams and A. Willson, "Some efficient digital prefilter structures," in *IEEE Transactions on Circuits and Systems*, vol. 31, no. 3, pp. 260-266, March 1984.

Appendix I

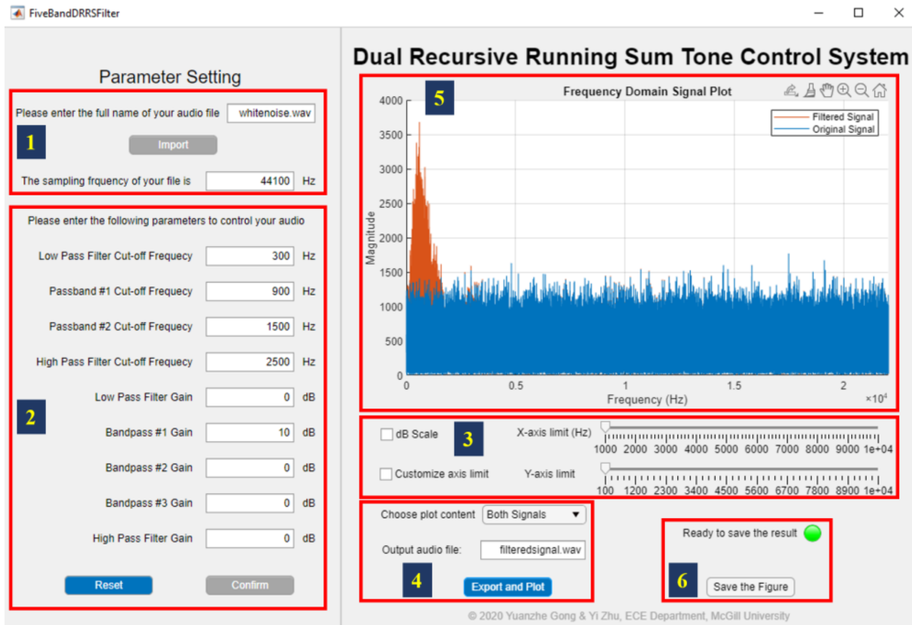


Fig. 9b Detailed divisions of the MATLAB-based five-band equalizer interface

As can be seen circled and labeled in Fig. 9b, this interface is divided into six sections. Step by step instructions for using this five-band tone control system are as following:

- In section 1, input the complete file name of the target WAV audio signal, including ".wav". The file should be under the same folder as the current MATLAB path.
- The sampling frequency of imported audio signal will be displayed in the unit of Hz in below.
- Input other parameters required in section 2 and click on the "Confirm" button when finished.
- In section 3, based on users' desire, check the "dB Scale" box for plotting the output in dB (or default plotting in linear scale) or check the "Customize axis limit" and move the slider on the right to modify the graph limits for both X- and Y-axis (or the signal graph will be plotted with default axis limits, in which the maximum frequency plotted will be half of the sampling frequency, while the maximum magnitude plotter will be based on input gain).
- In section 4, choose to plot the original signal, filtered signal, or both signals on the same graph.
- Based on users' desire, modify the name of the output filtered audio signal file.
- In section 5, click on the "Export and Plot" button to generate the filtered audio signal file in the current MATLAB path, and plot the selected signal in the graphic area accordingly.
- In section 6, after the green lamp is on, click on the "Save the Figure" button to save the signal plot into any preferable local destinations with desired image formats.
- Click on the "Reset" button in section 2 to clear out all input files, parameters, and signal plots.
- Users could select another audio file and manipulate it by re-performing the steps above.

Appendix II

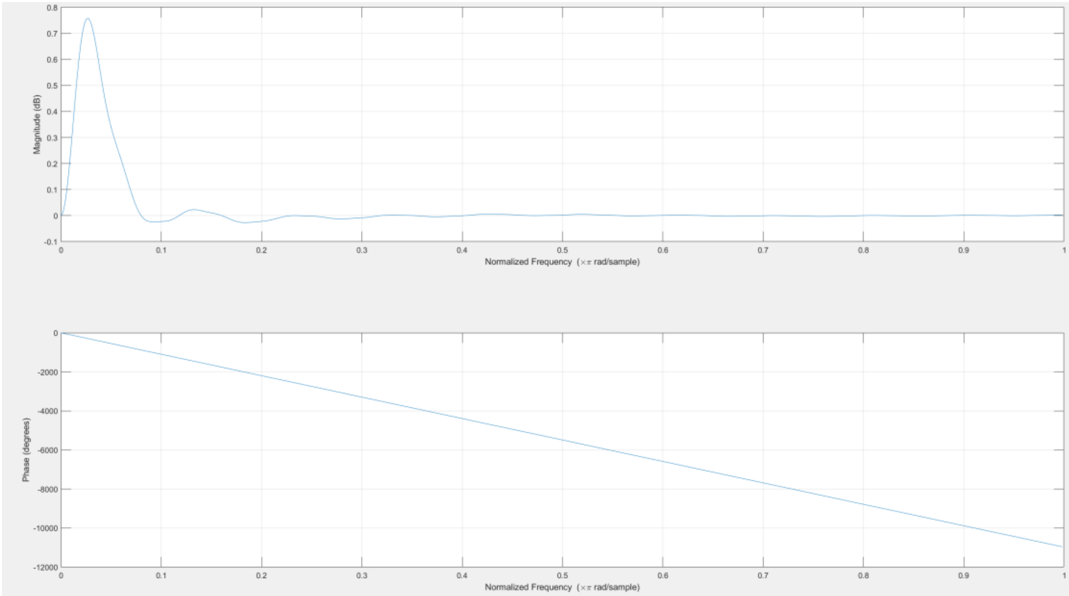


Fig.11b Filter Magnitude and Frequency Response with input $(G_L, G_{B1}, G_{B2}, G_{B3}, G_H) = (0,1,0,0,0)$ dB

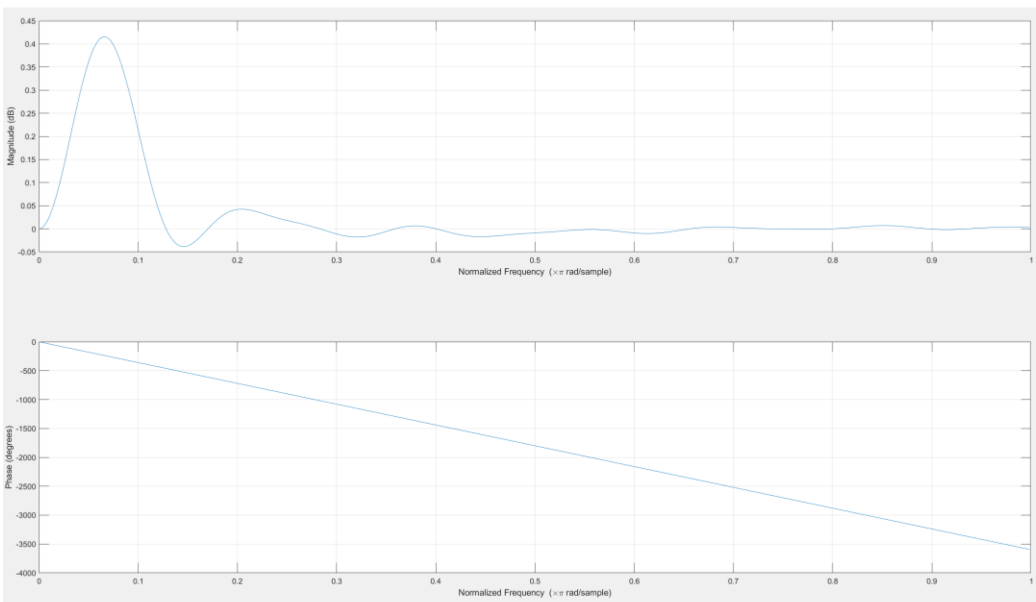


Fig.11c Filter Magnitude and Frequency Response with input $(G_L, G_{B1}, G_{B2}, G_{B3}, G_H) = (0,0,1,0,0)$ dB

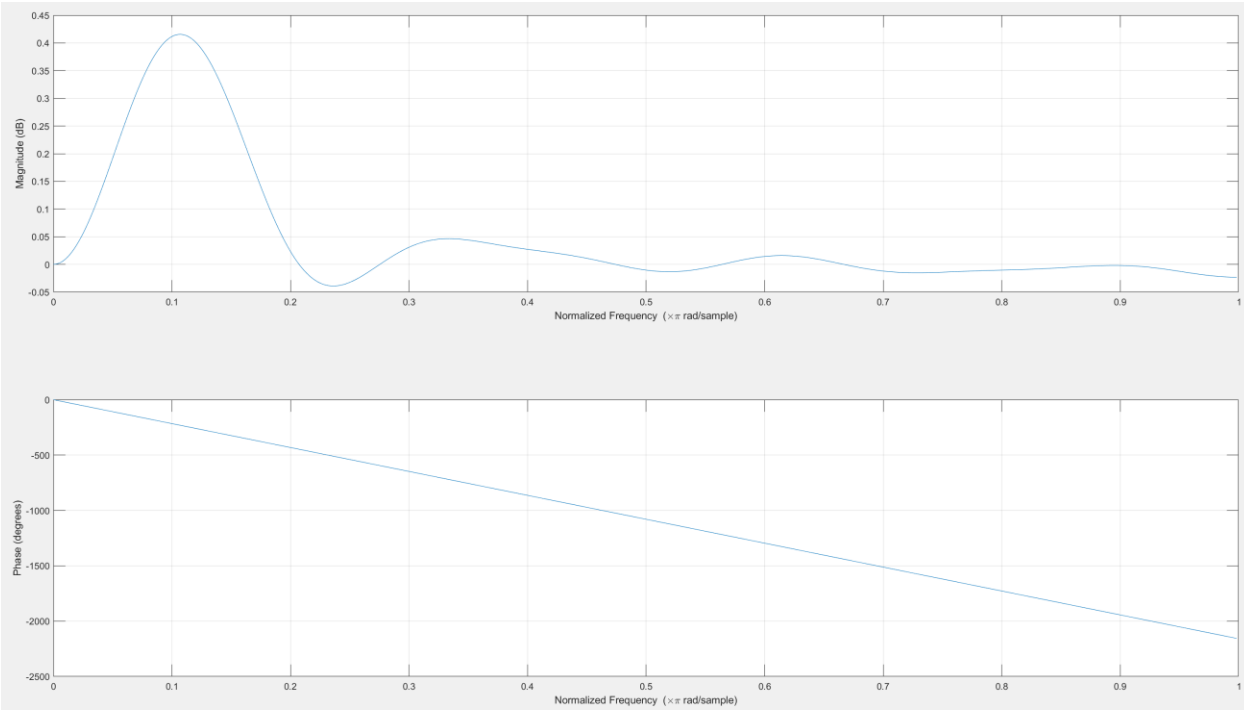


Fig.11d Filter Magnitude and Frequency Response with input $(G_L, G_{B1}, G_{B2}, G_{B3}, G_H) = (0,0,0,1,0)$ dB

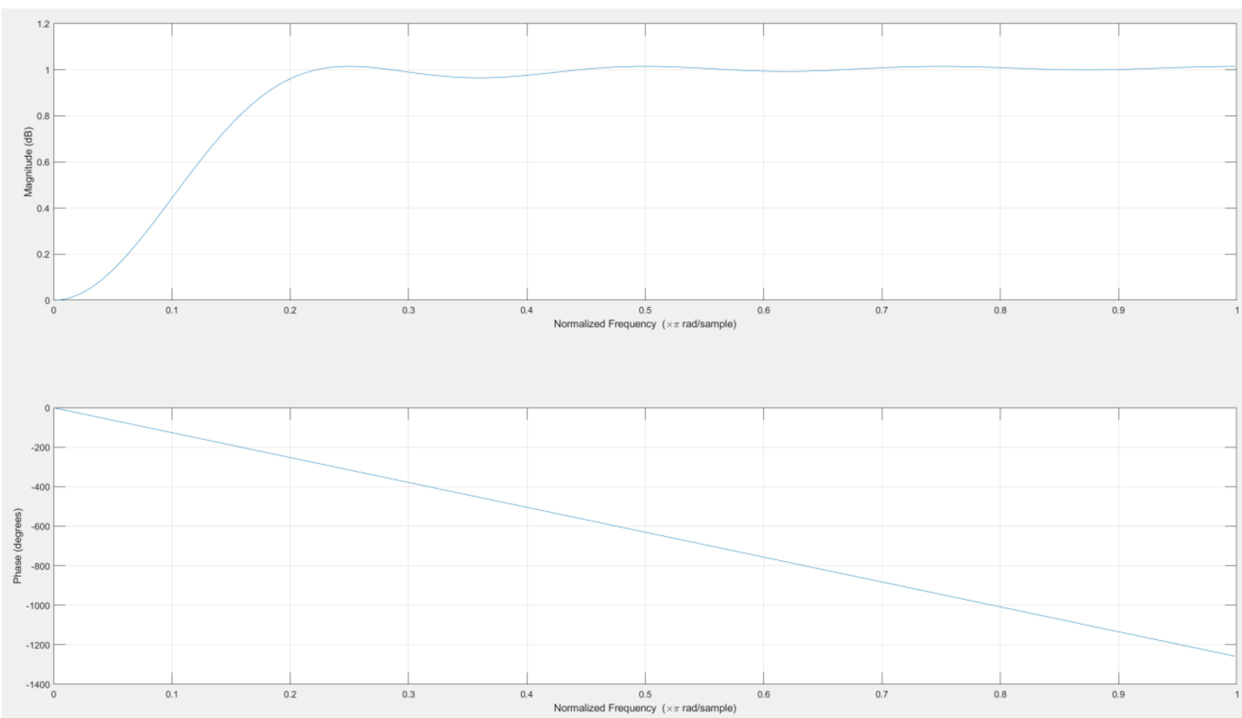


Fig.11e Filter Magnitude and Frequency Response with input $(G_L, G_{B1}, G_{B2}, G_{B3}, G_H) = (0,0,0,0,1)$ dB

Appendix III

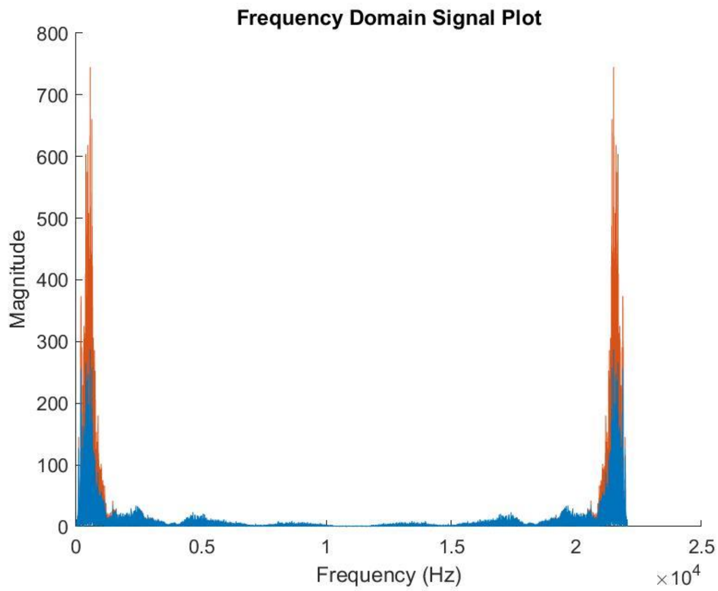


Fig. 12a Filtered signal (orange) and original signal (blue) plot of *sample1.wave* in linear scale with parameters: $f_s = 22050Hz$, $f_L = 300Hz$, $f_{B1} = 900Hz$, $f_{B2} = 1500Hz$, $f_H = 2500Hz$, $G_L = 0dB$, $G_1 = 10dB$, $G_2 = 0dB$, $G_3 = 0dB$, and $G_H = 0dB$

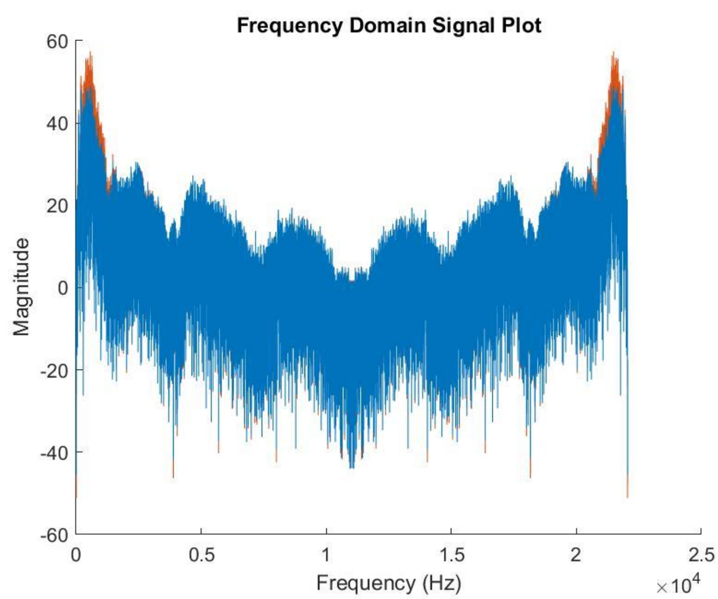


Fig. 12b Filtered signal (orange) and original signal (blue) plot of *sample1.wave* in dB scale with parameters: $f_s = 22050Hz$, $f_L = 300Hz$, $f_{B1} = 900Hz$, $f_{B2} = 1500Hz$, $f_H = 2500Hz$, $G_L = 0dB$, $G_1 = 10dB$, $G_2 = 0dB$, $G_3 = 0dB$, and $G_H = 0dB$

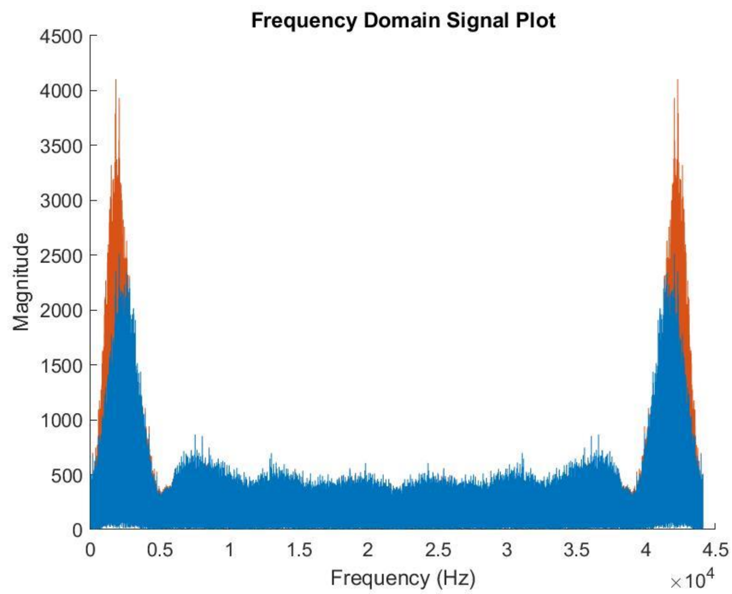


Fig. 13a Filtered signal (orange) and original signal (blue) plot of *sample2.wave* in linear scale with parameters: $f_s = 44100Hz$, $f_L = 300Hz$, $f_{B1} = 900Hz$, $f_{B2} = 1500Hz$, $f_H = 2500Hz$, $G_L = 0dB$, $G_1 = 0dB$, $G_2 = 10dB$, $G_3 = 0dB$, and $G_H = 0dB$

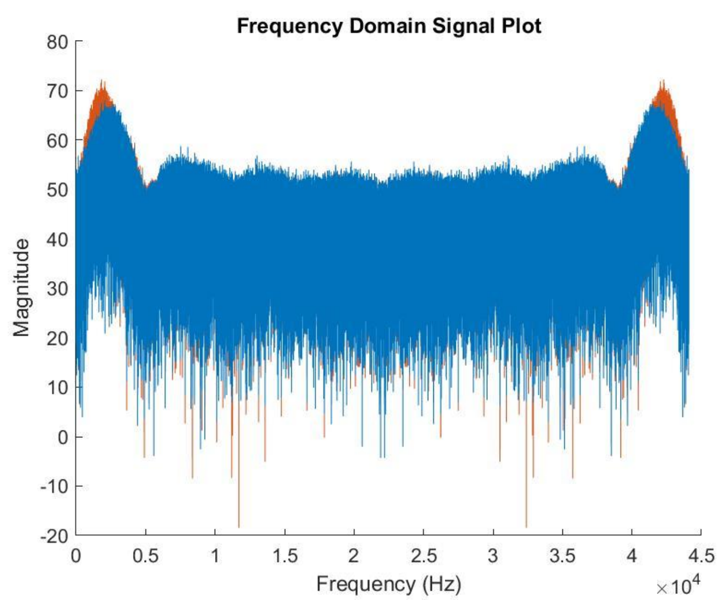


Fig. 13b Filtered signal (orange) and original signal (blue) plot of *sample2.wave* in dB scale with parameters: $f_s = 44100Hz$, $f_L = 300Hz$, $f_{B1} = 900Hz$, $f_{B2} = 1500Hz$, $f_H = 2500Hz$, $G_L = 0dB$, $G_1 = 0dB$, $G_2 = 10dB$, $G_3 = 0dB$, and $G_H = 0dB$

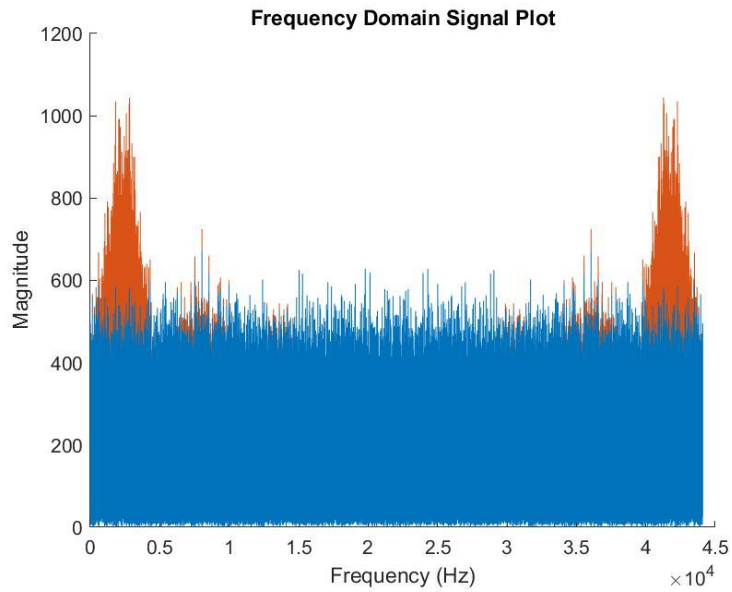


Fig. 14a Filtered signal (orange) and original signal (blue) plot of *sample3.wave* in linear scale with parameters: $f_s = 44100Hz$, $f_L = 300Hz$, $f_{B1} = 900Hz$, $f_{B2} = 1500Hz$, $f_H = 2500Hz$, $G_L = 0dB$, $G_1 = 0dB$, $G_2 = 0dB$, $G_3 = 10dB$, and $G_H = 0dB$

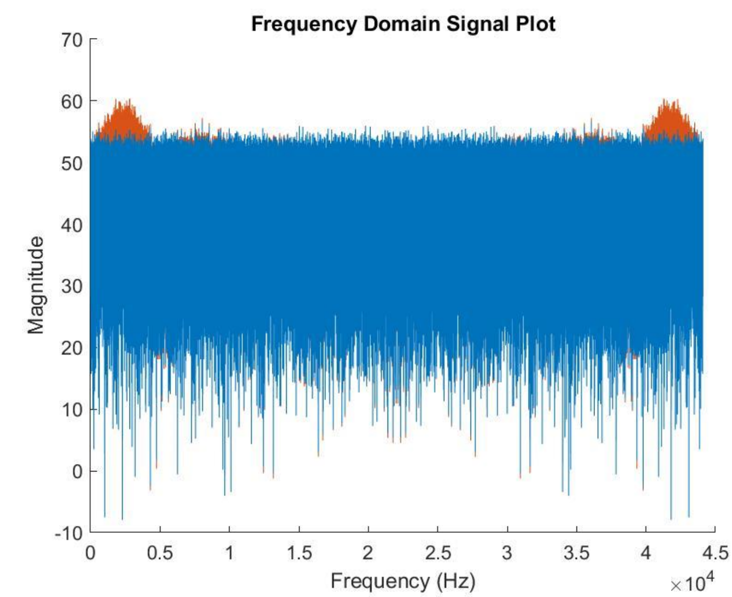


Fig. 14b Filtered signal (orange) and original signal (blue) plot of *sample3.wave* in dB scale with parameters: $f_s = 44100Hz$, $f_L = 300Hz$, $f_{B1} = 900Hz$, $f_{B2} = 1500Hz$, $f_H = 2500Hz$, $G_L = 0dB$, $G_1 = 0dB$, $G_2 = 0dB$, $G_3 = 10dB$, and $G_H = 0dB$

Appendix IV

```
%% clear all the previous data and record
clear;
clc;
syms z
% input file name
[a,Fs] = audioread('whitesample.wav');
white=wgn(length(a),1,0);
filename = 'whitenoise.wav';
% audiowrite(filename,test2,Fs);
audiowrite(filename,white,Fs);
[x,Fs]=audioread('whitenoise.wav');

system=inputparam(300,900,1500,2500,0,0,0,0,1,Fs);
%      inputparam(FLC,FB1,FB2,FHC,GL,GB1,GB2,GB3,GH,FS)
%results=simplify(system); % designed filter based on pass band cutoff and target gain
[fzn, fzd] = numden(system);
% extract the coefficient of the transfer function
fzc = sym2poly(fzn);
fzb = sym2poly(fzd);
%fvtool(fzc,fzb,'Fs',Fs)
freqz(fzc,fzb)
%% time domain convolution with impulse response conv with original signal
% get the impulse response based on the design parameters
h=impz(fzc,fzb);
% calculate the convolution of the system wit
fill=conv(x,h);
y=fill(length(h):length(fill)); %
NFFT=length(y);
FIL=fft(y,NFFT);
F = ((0:1/NFFT:1-1/NFFT)*Fs).';
magnitudeY = abs(FIL);

%% draw the filtered signal via filter function (Method 2)
% test2=filter(fzc,fzb,x);
% NFFT=length(test2);
% TEST=fft(test2,NFFT);
% F = ((0:1/NFFT:1-1/NFFT)*Fs).';
% TE = abs(TEST);
% plot(F,TE)
% grid on
% hold on

%% plot original signal before filtering
NFFT=length(x)
X=fft(x,NFFT);
F = ((0:1/NFFT:1-1/NFFT)*Fs).';
magnitudeX = abs(X);        % Magnitude of the FFT
%phaseX = unwrap(angle(X));  % Phase of the FFT
%% plot the  in freqeuncy domain
figure
plot(F,magnitudeX)
xlim([0 10000])
grid on
figure
%xlim([0 20000])
plot(F,magnitudeY)
xlim([0 10000])
grid on
```

```
%% plot in log scale:
% figure
% set(gca, 'XScale', 'log')
% xlim([0 20000])
% hold on
% plot(F,20*log10(magnitudeY))

%% plot and check the ratio of the gain after the filter
% figure
% H=magnitudeY./magnitudeX;
% ax=1:1:length(H);
% plot(ax,H)
% xlim([0 2000])
%% write into a new file
% filename = 'filteredsignal1.wav';
% % audiowrite(filename,test2,Fs);
% audiowrite(filename,fil1,Fs);
%% support function
function mainfunction=inputparam(FLC,FB1,FB2,FHC,GL,GB1,GB2,GB3,GH,FS)
syms z
L1=solveLength(FLC,FS)
B1=solveLength(FB1,FS)
B11=solveLength(FB2,FS)
H1=solveLength(FHC,FS)
mainfunction=Equalizer(L1,B1,B11,H1,GL,GB1,GB2,GB3,GH);
end

function result1=Drrs(L1,L2)
syms z
  result1=(1-z^(-L1))/(1-z^(-1))*(1-z^(-L2))/(1-z^(-1));
end

function y=getodd(x)
y = 2*floor(x/2)+1;
end

function bandLength=solveLength(fc,fs)
bandLength=getodd(1/(fc*2/fs));
%bandLength=getodd(1/(fc/fs));
end

function result2=Equalizer(L1,B1,B11,H1,GL,GB1,GB2,GB3,GH)
syms z
L2=getodd(L1/sqrt(2))
H2=getodd(H1/sqrt(2))
B2=getodd(B1/sqrt(2))
B22=getodd(B11/sqrt(2))
kH=1/(H1*H2);
kL=1/(L1*L2);
k1=1/(B1*B2);
k2=1/(B11*B22);
% change the log scale gain input to linear scale
GL=10^(GL/20);
GB1=10^(GB1/20);
GB2=10^(GB2/20);
GB3=10^(GB3/20);
GH=10^(GH/20);
result2=GH*z^(-[L1+L2+B1+B2+B11+B22]/2+1)*[z^(-[H1+H2]/2+1)-kH*Drrs(H1,H2)]...
        +GB3*[kH*z^(-[L1+L2+B1+B2+B11+B22]/2+1)*Drrs(H1,H2)-k2*z^(-[H1+H2+B1+B2+L1+L2]/2+1)*Drrs(B11,B22)]...
        +GB2*[k2*z^(-[H1+H2+B1+B2+L1+L2]/2+1)*Drrs(B11,B22)-k1*z^(-[H1+H2+L1+L2+B11+B22]/2+1)*Drrs(B1,B2)]...
        +GB1*[k1*z^(-[H1+H2+L1+L2+B11+B22]/2+1)*Drrs(B1,B2)-kL*z^(-[H1+H2+B1+B2+B11+B22]/2+1)*Drrs(L1,L2)]...
        +GL*kL*z^(-[H1+H2+B1+B2+B11+B22]/2+1)*Drrs(L1,L2);
end
```

Fig. 15 MATLAB code for the five-band DRRS tone control system