3 Band EQ

Introduction

Digital Audio Augmentation

Within digital audio, filters hold a large proportion of importance (Watkinson, 1994, pp.71–91). More generally, filters are used in the process of analogue to digital conversions, digital to analogue conversion, compression systems and digital signal processes such as equalising. The impulse response of a filter allows us to analyse the parts of the filter which are not uniformly affected (Watkinson, 1994, pp.71). This creates an ideal vs actual implementation of filters which can be used to augment audio data.

Principles of IIR Filters

An infinite impulse response filter not only relies on the input and past input but, can also rely on the past output of the system (Tan, 2008, pp.304). The output of a digital filter system can be shown in equation ? below.

Equation ?. The Relationship of a Digital Filter and Digital Signal

Where is the digital filter system response, is the Discrete Fourier Transform of the input signal . The Discrete Fourier Transform of a signal can be found using equation ? below.

Equation ?. The Discrete Fourier Transform of a Signal (Tan, 2008, pp.92)

For representation of filter signal flow diagrams, the Z transform is used commonly. This is to highlight the characteristics of a filter, as well as provide suitable block diagrams and difference equations. Equation ? is the formula for Z transform.

Equation ?. The Z Transform Equation (Tan, 2008, pp.135)

Where is the Z transform of the input  and is a complex variable. From this, we can equate a system response of an IIR filter represented in the Z-domain, shown in equation ?. This can further be written in the discrete domain with a difference equation shown in equation ?.

Equation ?. A Digital IIR Filter Transfer Function (Tan, 2008, pp.303)

Equation ?. The Difference Equation of a Digital IIR Filter (Tan, 2008, pp.303)

Where and are the numerator and denominator coefficients and M & N are the number of zeros (number of values for which is infinite) and poles (number of values for which is infinite) respectively.

Designing an IIR Filter

For this implementation, a second order IIR filter is used. A block diagram for this is shown in figure ?.? below.

Diagram, schematic

Description automatically generated

Figure ?.?. A Second Order IIR Filter Block Diagram

This filter utilises two previous signals. From this, a digital filter can be designed by deciding upon specific factors of the filter and choosing a filter type. An analogue prototype can then be used to create a high pass, low pass and band pass filter.

Analogue Modelling

To create a digital filter, an approach is to create the analogue counterpart and transform into a digital filter via a series of steps, shown in figure ?.? below.

Diagram

Description automatically generated

Figure ?.?. The Design Method of a Filter (Tan, 2008, pp.305)

To calculate the parameters shown in figure ?.? below, the specifications are generated in the analogue filter, before being frequency warped and transformed.

Chart

Description automatically generated

Figure ?.?. The Specifications of the Lowpass and Bandpass Filter (Tan, 2008, pp.326)

The attenuation of the passband and stopband are calculated and placed into the second order Butterworth filter (figure ?.?).

Table

Description automatically generated with low confidence

Figure ?.? The Conversion from Analogue to Lowpass Prototype (Tan, 2008, pp.325)

Text

Description automatically generated

Figure ?.? The Butterworth Lowpass Prototypes. N = Order, Hp(s) = Prototype. (Tan, 2008, pp.323)

The advantages of this design is not moving into the frequency domain via the Discrete Fourier Transform (Equation ?). This is less cost on a system as Bilinear transformation allows for the movement into the Z domain. Figure ?.? is the magnitude and phase response of a bilinear transformed band stop filter, designed by finding the coefficients and respectively. Full code for this MATLAB based diagram is in Appendix 1.

Chart, line chart

Description automatically generated

Figure ?.? The Magnitude and Phase Response of a Digital Band Stop Filter

From this, the peaking and shelving filters of the equaliser can be designed. Higher order filters allow for more control of the attenuation of the transition band. Using the computational power of MATLAB to design higher order filters allows for more powerful control of the equaliser.

Peaking & Shelving Filters

Peaking and shelving filters are used to boost or cut a frequency range specified by the parameters of the IIR filter specifications. Shelving filters boost or cute the low or high frequency bands with the parameters the cut off frequency and gain . Peaking filters boost or cut mid frequency bands with the parameters centre frequency , bandwidth and gain (Zolzer, 2011, pp.61). These filters are typically used in series but are controlled independently. Figure ?.? highlights filter coefficients for 2nd order shelving filter and figure ?.? are the filter coefficients for the 2nd order peaking filter.

Graphical user interface, text, table

Description automatically generated

Figure ?.?. Filter Coefficients for 2nd Order Shelving Filters (Zolzer, 2011, pp.64)

Table

Description automatically generated

Figure ?.?. Filter Coefficients for 2nd Order Peaking Filters (Zolzer, 2011, pp.66)

From the above figures, the values of , and can be defined with equations ?,? and ? below.

Equation ?. Frequency Parameter for Cutting or Boosting of the Filters

Equation ?. The Gain Parameter for the Filters

Equation ?. The Quality Factor of the Peaking Filter

For a three-band tone control, two shelving filters and a peaking filter for the mids is utilised. According to the assignment brief, the cut off frequency for the high and low filters is introduced, alongside the bandwidth of the mid peaking bandwidth moving. The filter coefficients are calculated based on user inputs for the corresponding parameters in the MATLAB code.

Equaliser

Equalisation covers a broad class of filtering effects, with different requirements suiting different implementations (Reiss and McPherson, 2015, pp.89). Tone control equalisers give a way of affecting equalising filters directly, with controls for cut-off frequencies and gains. An equaliser will utilise a combination of several filters such as the low pass, high pass and band pass filter to attenuate desired frequencies. Shelving and peaking filters are used to cut or boost a frequency range.

There are also graphic equalisers and parametric equalisers, which use the same theory with adjusted controls based on application. Graphic equalisers are a tool for precise adjustments to the gain of frequency regions. In contrast to a tone control, which may use three knobs for tone control, the graphic equaliser can use up to thirty controls to affect the frequency response (Reiss and McPherson, 2015, pp.94). Parametric equalisers are the most flexible, with control given over the gain, centre frequency and Q (Bandwidth) values (Reiss and McPherson, 2015, pp.98).

These equalisers all use the same foundation of filtering. With controls of those changing to fit the need of the digital audio engineer. The focus of this technical report is a three-band tone control equaliser.

A three-band equaliser uses one low shelving filter, one mid peaking filter and one high shelving filter, implemented in a cascading, parallel or hybrid network (Zolzer, 2011, pp.61). For the implementation of the report, the cascading equaliser is discussed.

Cascading Equaliser

A cascading equaliser sends an inputted signal through a series of filters, such as the three band filters outlined previously, to produce an equalised output signal (Zolzer, 2011, pp.61). Figure ?.? highlights a series of connected shelving and peaking filters, used in a cascading fashion.

Diagram

Description automatically generated

(Figure ?.?.) A Series of Connected Filters (Zolzer, 2011, pp.61)

The overall frequency response of a cascade equaliser is therefore the product of each individual filter responses. This can be shown with equation ? (Valimaki and Liski, 2017).

Equation ?. The Frequency Response of a Cascading Equaliser (Valimaki and Liski, 2017).

Where is a gain factor, are the frequency responses of equalising filters (m = 1,2,3,…M), is the frequency and is the sample interval.

Loudness

For this context, loudness has a different meaning to volume or amplitude. The human ear perceives the loudness of different frequencies at varying levels (Reiss and McPherson, 2015, pp.93). Figure ?.?. below highlights a plot of the human auditory response.

Diagram

Description automatically generated

(Figure ?.?.) Realm of Human Auditory Response (Somers, n.d.)

This means that low frequencies would need to be boosted with respect to the midrange to be perceived as equally loud (Reiss and McPherson, 2015, pp.93). The inclusion of a loudness switch within an equaliser design allows for this boost to happen via a weighted filter defined by standards set out by that Acoustical Society of America (2016). ‘A’ weighting is the most commonly used, which is applied within the created three band filter.

Methodology

The methodology focuses on the implementation of a three band equaliser in MATLAB as MATLAB scripts and functions, as well as discussing the completed MATLAB Application. The implemented MATLAB script (EqMain) works as a prototype file for the MATLAB application (ThreeBandFilterApp), working with the same parameters for each portion, but with further control and an outputted frequency response for files. Both the script and application do not work in real time, requiring a user to allow for computation time. Future designs would work with creating a buffer to store data to supersede this.

The created audio equaliser is made with a cascade of filters, with the product of the filters used to change the values of the filter parameters. It is implemented with the aforementioned 2nd order shelving and peaking filters, using two shelving and one peaking filter. The filter coefficients were calculated using figures ?.? and ?.? above.

Equaliser Control

The equaliser is designed to calculate each individual magnitude response separately and then multiply them together for the product of the equaliser’s magnitude response. The inbuilt ‘freqz’ function is used in this instance using the filter coefficients as inputs, calculating the overall equaliser response by multiplying all filter responses. Using the inbuilt ‘invfreqz’ function finds the filter coefficients of the equalising filter. These coefficients can then be placed into the filter() function alongside the input signal to create the equalised output signal.

Convolving the individual magnitude responses would lead to the same output, with each convolution leading to the coefficients for the filter() function directly. Figure ?.? is a flowchart for the MATLAB script.

Diagram

Description automatically generated

Figure ?.?. A Flowchart for Audio Equalisation for MATLAB Script

Parameters

As stated for the assignment, the parameters for control for this equaliser is three gain controls for the low, mid and high values, alongside cutoff frequencies for the low and high shelving filters. These frequencies also correspond to the bandwidth of the mid peaking filter. Finally, an ‘a’ weighting loudness curve is implemented with a choice of on or off (boolean).

For the MATLAB script, the user is given the choice of these controls within the console of MATLAB, shown in figure ?.?. There are also presets given to a user to ease of application. The application has a graphic user interface which can be used to change values (figure ?.?).

Graphical user interface, text, application, email

Description automatically generated

Figure ?.?. The User Inputs for MATLAB Script Equaliser

Graphical user interface

Description automatically generated

Figure ?.?. The Equaliser Application Graphic User Interface

Once audio is played, the controls are disabled until the audio stops either manually or finishes the track. This is to reflect the non real time application of the filters.