Analysis of popular equalizers

Lauri Vapalahti
Aalto Universtiy
Master's Programme CCIS / AAT

lauri.vapalahti@aalto.fi

Abstract

Equalizers have many different algorithms and purposes which are not mentioned clearly publicly. This is why it is important to study how some of the most common EQs work and are those suitable for which use. Audacitys' EQ is accurate but it drawback is in effective gain control with single band. The structure is one large FIR filter using changeable length. Behringer Xenyx 1002B has smooth but inaccurate filtering. Filters are most likely 2nd order IIR filters in cascade form. Filters have lot of over-lap. Dolby audio for Acer laptops is a non-linear system which distorts low frequencies and the cuts the lowest frequencies. It is designed for compensating laptops poor speakers. Filtering is not accurate and pass/stop bands have a lot of ripple. Boss GE-7 is a good guitar EQ with 2nd order IIR filters in cascade. There is a lot of over-lap which has to be taken account when filtering. Ableton Lives default EQ is made with 4th order IIR or 8th order FIR filters in parallel structure so there is no over-lap. Fist methods and tools are presented, then EQ are measures with analyses. Scrips used for calculations are in appendix.

1 Introduction

Originally equalizer (EQ) were invented to correct the loss of high frequencies in long analog telephone lines [4]. Now days EQs are used in many fields of transmission and audio technologies. EQs can be used to achieve creative effects in music production, enhance certain frequencies for clearer understanding, compensate the effect of the listening room or to adjust the listening experience for individual listener. The most common types of equalizers used by music producers, audio engineers and consumers are graphic EQ and parametric EQ.

Graphic EQ consists usually 2-30 parameters which each controls the gain of certain frequency bands. All bands are bandpass/stop filters but the lowest and highest bands can also shelving filters. The bands are placed logarithmically across the humans' hearing range from 20 Hz to 20 kHz so that all band has has a common Q-factor.

$$Q = \frac{\omega_c}{B}, \tag{1}$$

where ω_c is the center angular frequency of the band and B is the band width of the band. Graphic EQs are user friendly because of the intuitive user interface. The drawback of graphical EQs can by the inference between bands' transition bands and high filter orders. The interference between bands can distort the wanted frequency response and high filter order produces longer latency and need more computing power in digital filter and more physical components in analog filters. [5]

Parametric EQs usually have 1-16 filters which center frequency (ω_c), gain (G), Q-factor (Q) and type can be adjusted. This allows accurate design for the EQs frequency response but the usage of a parametric EQ need more know-how compared to graphic EQ. That is why parametric EQs are used more in professional audio processing and almost all end-user EQs are graphic.

Because most of the user applications use graphic EQs which does not state the actual frequency response, it can not be said are those presentations of the responses accurate. This is why it is important to analyse some of the common EQs so those can be used correctly to achieve wanted effect. This is also interesting in academic point of view for product development.

This paper goes through how to get the frequency response from digital and analog EQs and how to analyse those. With these information the EQ implementations can be reverse engineered and user can choose right EQ for use and use it correctly. This paper includes only couple of EQ implementations for different usages: Audacity's default graphic EQ for filtering in free digital audio workstation (DAW), Behringer's mixer with three band graphic EQ for live music filtering, Dolby Audio's presets and graphic EQ for computers, Boss GE-7 guitar pedal graphic EQ for instrument filtering and Ableton Lives' default EQ for professional music production.

First this paper tells the used methods and how to analyse EQs. After that the frequency responses and analyses for each EQs are presented. Finally in conclusion the EQs are compared to each other so the right tool can be select for the task.

2 Measuring and analysing methods

This section section address the tools and measuring and analysis methods used to evaluate the equalizers. First Methods section describes mathematical practices how to get information from a system and which are used in this paper. Then tools what are used are mentioned in Tools section. Finally the method of analyse are covered.

2.1 Methods

Impulse response (IR) is when a unite impulse is fed through the system and the output is captured. Comparing the time difference between input and output the lag of the system can be calculated. The frequency response of the system can be calculated from the IR using discrete-time Fourier transformation (DTFT) which equation is

$$H(\omega) = \sum_{n=-\infty}^{\infty} h[n]e^{-i\omega n} \tag{2}$$

where h is the impulse response and ω is normalized angular frequency.

White noise can be also used to get the frequency response. White noise with constant power for all frequencies is fed to the systems and the frequency response is calculated from the output using equation [2].

Logarithmic sine-sweep also known as Farina's sweep is also a common method to achieve the frequency response. The equation for sweep is

$$x(t) = \sin\left(\frac{2\pi f_1 T}{\ln(f_2/f_1)}\left(e^{\frac{t}{T}\ln(f_2/f_1)} - 1\right)\right) \tag{3}$$

where f_1 is the starting frequency, f_2 is the end frequency and T is the length of the sweep in seconds. [3] The frequency response is achieved by taking DTFT from input signal x and output signal y and then taking their ratio

$$H(\omega) = \frac{FFT(y(t))}{FFT(x(t))} = \frac{Y(\omega)}{X(\omega)} \tag{4}$$

In this paper Farinas' sweep method was picked to receive the frequency responses from EQs because of its' effectiveness and easy implementation.

2.2 Tools

Matlab was used to generate the measuring signal and calculating the frequency responses. The signal used was five second long logarithmic sin-sweep (Farinas' sweep) starting from 20Hz and ending to 20kHz. Sampling frequency was 96kHz and signal was 16 bit WAV audio file format.

For hardware EQs audio was played and recorded through Focusrite Scarlett 2i4 2nd Gen audio interface [2]. For software EQs the filtering was done withing the DAWs (Audacity and Ableton) but for Dolby Audio for Acer computers the signal had to be played through computers own audio output and recorded with external audio interface.

2.3 What to look?

EQ patterns used in this paper were:

1. Zig-zag using maximum and minimum gains
2. One band to maximum gain (mid or around 1000 Hz)
3. All up

From these it's analysed:

1. How well command gains are met
This tells how accurately the user interface presents the real gains.

2. Is there over-lap between bands
From this it can be seen if the EQ uses parallel or cascaded filters and if the EQ uses some kind of compensating algorithm.

3. Length and shape of transition and pass/stop-band
This information can be used to estimate the order of the filters. It is also a common interest to know how much the amplitude varies in the band.

3 Measurements

This sections shows all the measure figures and information about the systems which are tested. All figures have logarithmic y-scale and frequencies scale from 20Hz to 20kHz. Ziz-zag patterns are all presented in own figures but all-up and mid-up patterns are drawn to the same figure with blue (all-up) and red (mid up).

3.1 Audacity

Audacity is a GPL-licensed cross-platform audio editing software. Audacity has a 32 band graphic EQ which length can be adjusted. All measurements are done with 4051 long filter which is recommended by Audacitys' manual. Each band can be adjusted between -20dB and +20dB.[1]

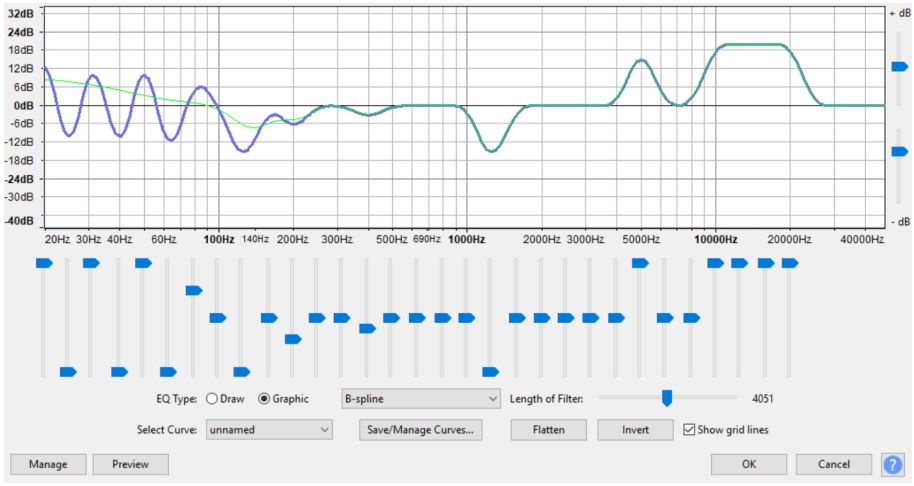

Figure 1: GUI of Audacitys' EQ

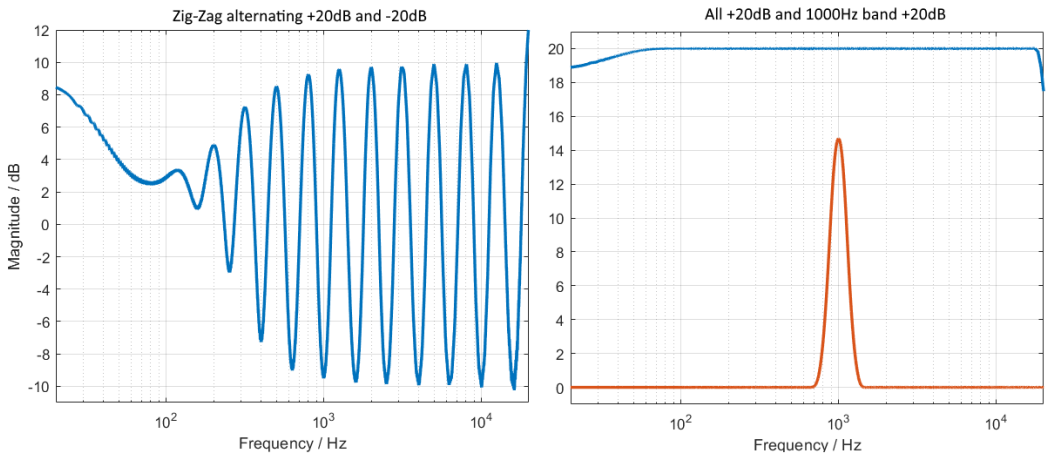

Figure 2: Measurements of Audacitys' EQ

Audacitys' EQ is designed to have flat response if gains are set to same command gains. From the zig-zag patterns on the left it can be seen that the filtering is not that accurate in low frequencies and the the command gain +20dB or -20dB is not met at any band due to overlap between neighbouring bands. A single band only gets to ±15dB. These drawbacks of the algorithms has be taken account when filtering in Audacity. The EQ seems to be cascade IIR structured or a single high order FIR filter.

3.2 Behringer Xenyx 1002B

Behringer is a six channel mixer which has three band EQ on each channel with low, mid and high controls with ±15dB gain control.

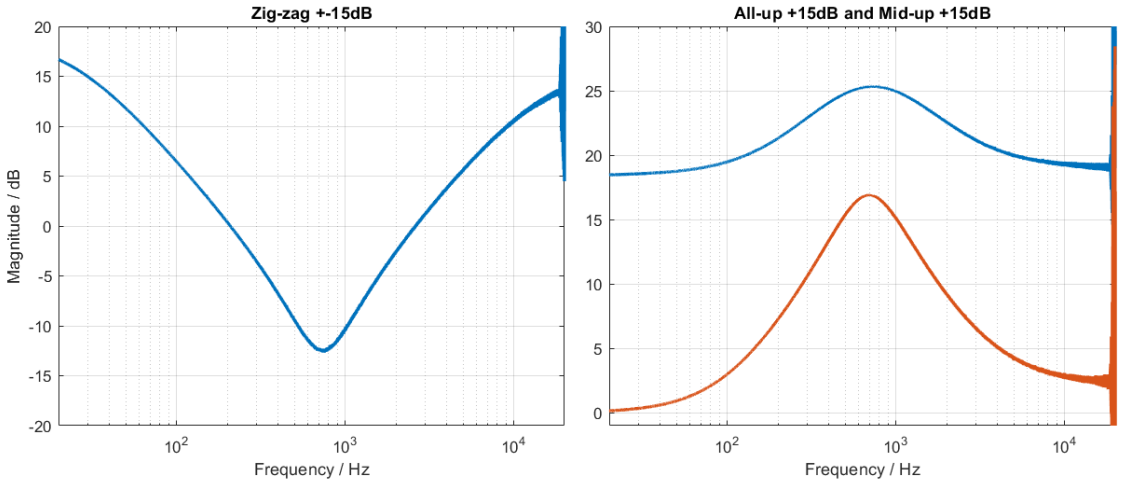

Figure 3: Measurements from Behringer Xenyx 1002B

The low and high bands work as shelving filters and mid band is wide bandfilter with center frequency at 700Hz. Command gains are not met so well. Mid channel error is +2dB when gain is set to +15dB and the response is not flat with maximum of +10dB error at 700Hz in all-up pattern. Maximum error with zig-zag pattern is +3dB at 700Hz

Filters are quite wide and round with long transition bands so he filter order can be estimated to be 2. There is also much overlap so the EQ seems to be made with cascade structure.

The Xenyx 1002B mixers' EQ is not the best solution for live mixing which need more accurate filters to reduce resonant frequencies of the room. The accuracy of the filters could be suitable for home use where EQ is used only to adjust the timbre of the sound.

3.3 Dolby Audio in Acer computers

Dolby has developed an audio enhancement program for Acer laptops and it has couple of presets and 20 band graphic EQ. Only the EQ is measured in this paper.

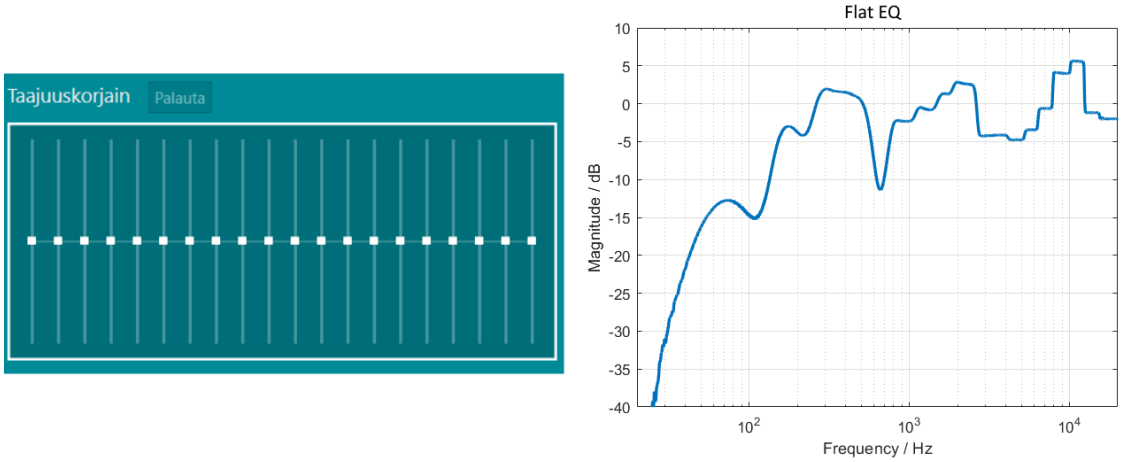

Figure 4: Dolby Audios' GUI and corresponding frequency response.

Dolbys audio enhancement for laptops is a non-linear process which is designed to compensate laptops poor speakers. Low frequencies are distorted to achieve louder bass and the lowest frequencies are cut away. This make Dolby audio difficult to compare with other EQs in this paper because the purposes are different. Zig-zag, all-up and mid-up spectrum's are compared with flat response from figure 4.

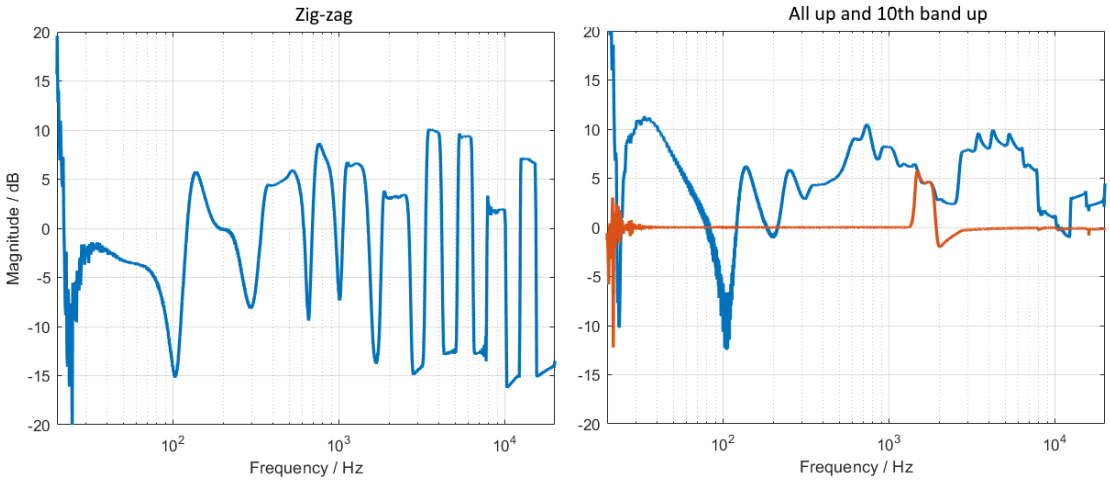

Figure 5: Zig-zag pattern on the left and all-up and 10th band up on the right.

Bands in Dolbys EQ seems to be able to control bands between -5dB and +5dB. Lower frequency bands are wider then higher bands which is not optimal because humans hearing is more accurate at low frequencies. The command gains are not met which might be because the software was designed to compensate poor laptop speakers.

Filters are designed with same orders which can be seem from that the filtering is more accurate at high frequencies where as low frequency filters are rounder. From short transitions bands and flat pass/stopbands the order can be estimated to be around 4. Overlap can also be detected within band so the structure is most likely cascade IIR.

3.4 Boss GE-7

Boss GE-7 is one oh the most common graphic EQ for electric guitars used between a guitar and a amplifier. GE-7 has 7 band and one common gain slider which can be adjusted between ±15dB. During the measurements the 800Hz band was discovered to be broken so that it did not affect the frequency response of the system.

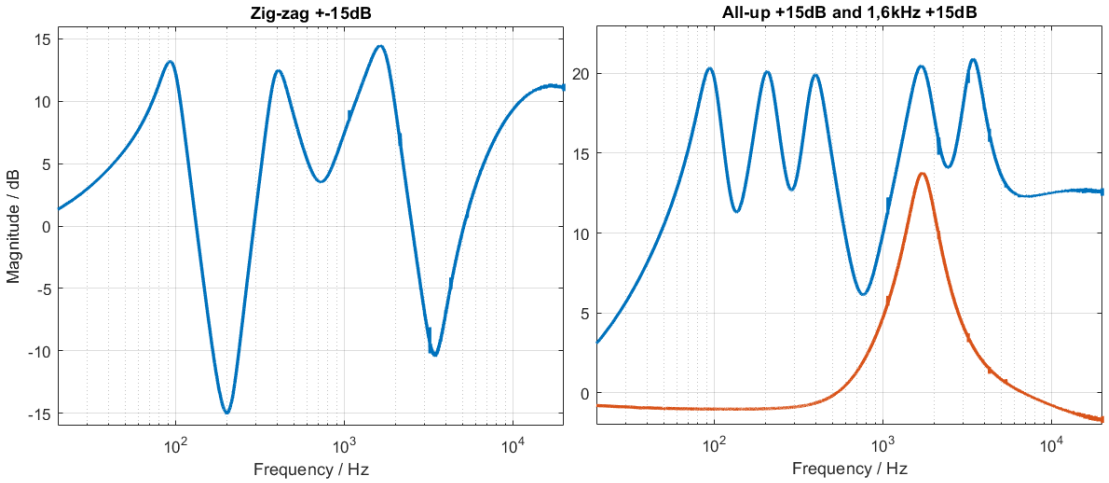

Figure 6: Measurements from Boss GE-7

All bands are bandfilters but the highest band has also shelving filter characteristics. The command gains are not met perfectly in any EQ patterns. Largest error is +6dB at ——Hz in all-up pattern. Red plot in right figure in 6 shows that single band does not achieve +15dB but +13dB.

All-up pattern has a large of variance at passband which indicates overlap within bands. From this the structure is cascade IIR. Looking at the shape of individual filters the order of the filters can be estimated to be 2.

3.5 Ableton Live default EQ

Ableton live is a commercial digital audio workstation. Ableton has a default three band semi-parametric EQ. In the EQ gains can be adjusted between $-\infty$ and +6dB. Cutoff frequencies of low and high bands can be adjusted but in these measurements these are kept at 350Hz and 2500Hz. The steepness of transition bands can also be adjusted to be 24dB/octave of 48dB/octave, but only 48dB/octave is used in the measurements.

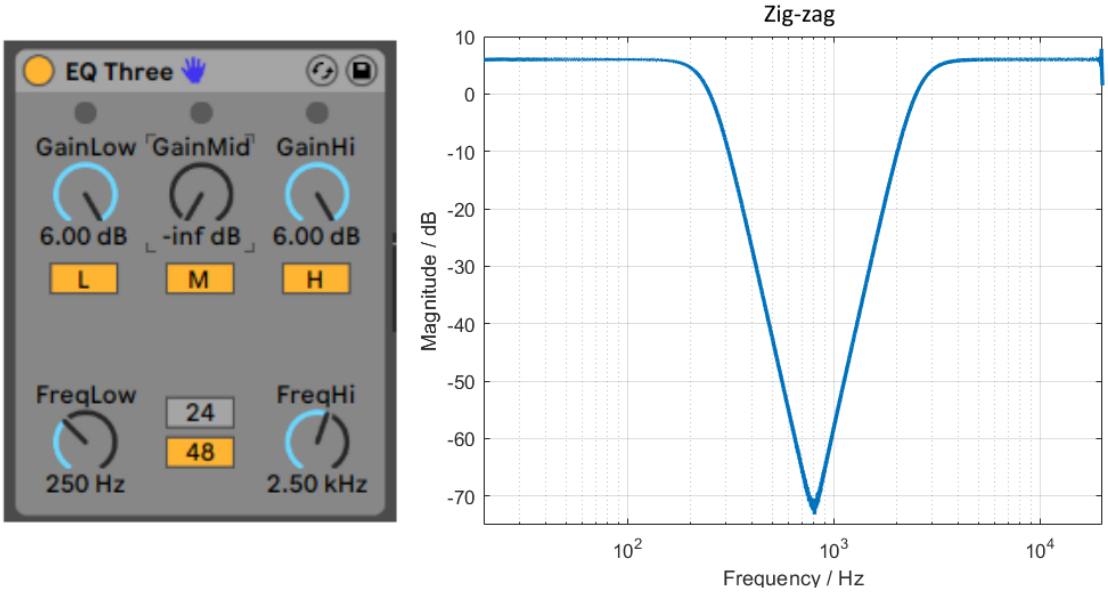

Figure 7: Abletons' GUI and zig-zag pattern.

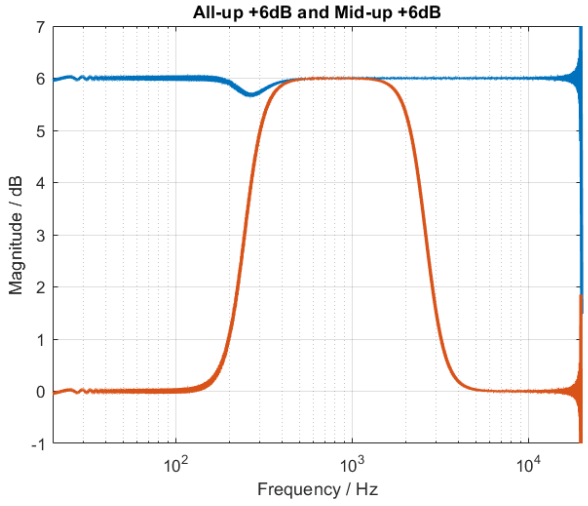

Figure 8: Abletons' all-up (blue) and mid-up (red) patterns.

Low and high bands work as shelving filters. Command gains are met well and pass/stop bands does not have ripple. Largest error of 0.3dB is in all-up pattern

between low and mid bands.

Filters have 46dB/octave transition band which indicates that filters are 4th order IIR or 8th order FIR filters. There is not much interference between band so structure is parallel FIR or IIR. [4]

4 Conclusion

There is many kind of equalizers in use and Equalizers can have widely different purposes and algorithms. Some equalizers have: larger amplification/attenuation, more accurate filtering, flatter pass/stop bands, shelving filters, real-time processing, more flexibility. So it is important to select correct tool to use in different situations.

Frequency response can be calculated in different ways. By using DTFT to impulse response or to filtered white noise or with Farinas' sweep method. Farinas' sweep method was discovered to be most robust way to calculate frequency responses during this study with available tools.

Farinas' sweep was made with 96kHz sampling frequency logarithmic sine-wave from 20Hz to 20kHz. The signal was then filtered in the softwares or fed through Focusrite Scarlett 2i4 2nd gen audio interface to hardwares and recorded with Audacity. Finally spectrum's were calculated with Matlab script in appendix.

Audacity has stable and accurate filtering over 200Hz bands, but over-lap within bands can lower the neighboring bands amplification. Ripple in pass/stop bands is nonexistent. The structure is most likely one large FIR filter with adjustable length.

Behringer three band EQ has smooth and quite accurate response. Bands have large over-lap with each other and thus there is a lot of overshoot. This EQ is mainly usable for creating artistic color to the sound.

Dolby audio has non-linear components in it. Filtering is not accurate and the bands can only be adjusted between $\pm$5dB. Filter are high order IIR filters with varying q-value. Dolby audio enhances the sound in laptops well but if clean signal is needed from a laptop then Dolby audio in best to shut down.

Boss GE-7 is handy seven band EQ for guitar filtering. The response is smooth, but overlapping filters affect the real gain. In all-up pattern ripple is 8dB which is a lot. Structure is 2nd order cascade IIR.

Ableton Live has accurate 3 band filtering. Command gains were met well and parallel IIR structure does not produce interference between bands. Abletons default EQ is well suited for simple filtering in audio production.

References

[1] Audacitys' EQ web-manual. https://manual.audacityteam.org/man/equalization.html. Accessed: 29.4.2019.

[2] Focusrate scarlett 2i4. https://focusrite.com/usb-audio-interface/scarlett/scarlett-2i4. Accessed: 29.4.2019.

[3] Farina, A. Advancements in impulse response measurements by sine sweeps. In *Audio Engineering Society Convention 122* (2007), Audio Engineering Society.

[4] Välimäki, V., and Reiss, J. All about audio equalization: Solutions and frontiers. *Applied Sciences 6*, 5 (2016), 129.

[5] Vapalahti, L., et al. Digitaaliset graafiset taajuuskorjaimet.

Appendix A Get a frequency response from an audio system

get_H(sig, G, fig, rajat, ref)
sig = Signal to be measured
G = gain difference in dB between sig and ref
fig = figure number
rajat = y-axis of the plot
ref = optional reference signal if other than Farina sweep.

ref is 5 seconds log Farinas sweep (20Hz->20kHz) if ref is not included in input arguments. Function returns 1 if there is no errors.

Lauri Vapalahti 2019

```
function temp = get_H(file, G, fig, rajat, reference)
    g = 10^(G/20);
    [y, fs] = audioread(file);   % Read system output and sampling frequency
    y = y(:,1);                  % Make signal mono

    switch nargin                % Use ref or generate a sweep
        case 5 % Use ref-signal
            x = audioread(reference);
            x = x(:,1);          % Make mono
        case 4 % Farinas sweep
            T  = 5;              % Legnth of the sweep in second
            t  = (0:T*fs-1)'/fs; % Time vector for the sweep
            f1 = 20;             % Starting frequency
            f2 = 20000;          % End frequency
            x  = g*0.5*sin(2*pi*f1*T./log(f2/f1)*(exp(t/T*log(f2/f1))-1));
    end

    if length(x) < length(y)     % Pad shorter signal with zeros
        x(length(x)+1:length(y)) = 0;
    elseif length(x) > length(y)
        y(length(y)+1:length(x)) = 0;
    end

    H = fft(y)./fft(x);

    % Plot
    H = H(1:floor(length(H)/2));
    m = length(H);
```

```
    w = (0:m-1)'/m*fs/2;
    H_mag = 20*log10(abs( H ));
    % H_mag = filter([1 1 1 1 1 1 1 1]/8,1,H_mag);  % Smoothen the plot if
    % needed
    figure(fig)
    semilogx(w, H_mag,'LineWidth',2)
    grid on
    axis([20 20000 rajat(1) rajat(2)])
    title(file)
    xlabel('Frequency / Hz')
    ylabel('Magnitude / dB')
    temp = 1;
end
```

Appendix B main.m

Generate a Farinas sweep

```
%fs = 96000;
% Sampling frequnecy %T = 5;
% Legnth of the sweep in second t = (0:T*fs-1)'/fs;
% Time vector for the sweep f1 = 20;
% Starting frequency f2 = 20000;
% End frequency x = 0.5*sin( 2*pi*f1*T./log(f2/f1) * ( exp(t/T*log(f2/f1))-1 ) );



%audiowrite('Sweep.wav',x, fs); audioinfo('Sweep.wav')



% get_H(sig, G, fig, rajat, ref)
%       sig = Signal to be measured
%       G   = gain difference in dB between sig and ref
%       fig = figure number
%       rajat = y-axis of the plot
%       ref = optional reference signal if other than Farina sweep
%
%       Returns 1 if there is no errors
```

Dolby on (flat EQ) vs. Dolby off

```
get_H('flat.wav', 0, 1, [-40 10], 'dolby_off.wav')
```

Dolby

```
rajat = [-20 20];
ref = 'flat.wav';

get_H('all_up.wav', 0, 11, rajat, ref);         % All-up
hold on
get_H('mid up.wav', 0, 11, rajat, ref);         % Mid-up
hold off
title('All-up and 10th band up')

get_H('zig.wav'   , 0,111, rajat, ref);         % Zig-zag
title('Zig-zag')
```

Audacity

```
rajat = [-1 21];

get_H('all+20_g-15.wav', -15, 2, rajat); hold on    % All-up
get_H('1000+20_g-10.wav', -10, 2, rajat); hold off  % Mid-up
title('All-up +20dB and 1kHz +20dB')

get_H('zig+20-20_g-10.wav', -10, 21, [-11 12]);     % Zig-zag
title('Zig-zag +-20dB')
```

Ableton

```
rajat = [-1 7];

get_H('all+6.wav', 0, 3, rajat); hold on            % All-up
get_H('mid+6_g0.wav', 0, 3, rajat); hold off        % Mid-up
title('All-up +6dB and Mid-up +6dB')

get_H('zig+6-inf+6_g0.wav', 0, 31, [-75 10]);       % Zig-zag
title('Zig-zag +6dB -inf +6dB')
```

Boss

```
get_H('Boss_flat.wav', -21.5, 5, [-5 10])
```

```
rajat = [-2 23];
ref = 'Boss_flat.wav';

get_H('Boss_all.wav', 0, 4, rajat, ref); hold on    % All-up
get_H('Boss_1600.wav', 0, 4, rajat, ref); hold off  % Mid-up
title('All-up +15dB and 1,6kHz +15dB')

get_H('Boss_zig.wav'   , 0,41, [-16 16], ref);      % Zig-zag
title('Zig-zag +-15dB')
```

Behringer

```
get_H('Beh_flat.wav', -26, 5, [-10 10])


rajat = [-1 30];
ref = 'Beh_flat.wav';

get_H('Beh_all.wav', 0, 5, rajat, ref); hold on     % All-up
get_H('Beh_mid_up.wav', 0, 5, rajat, ref); hold off % Mid-up
title('All-up +15dB and Mid-up +15dB')

get_H('Beh_zig.wav'   , 0, 51, [-20 20], ref);       % Zig-zag
title('Zig-zag +-15dB')
```