MultiBand Compressor

DAFX coursework 1

Clifford Moses Manasseh.

Introduction:

MultiBand compressors are a class of compressors that fall under the category of dynamics processing audio effects, They are useful in controlling the dynamic level of the audio signal .

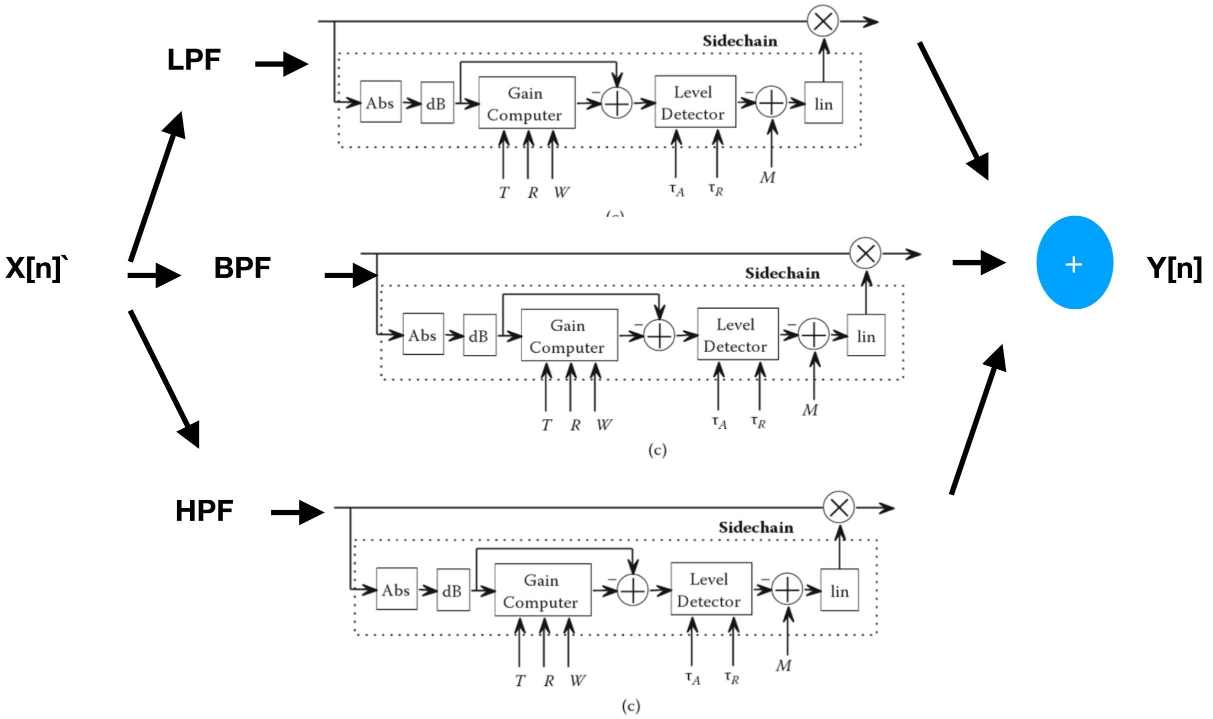
Compressors are used to control and modify the dynamic range of audio and also used in shaping the transients of the signal. A compressor on an audio signal can be used increase or decrease the difference between the softer and the louder parts of the audio and also shape how hard or soft transients hit and how fast or slow the signal fades off .

Although compressors can also be experimented with to produce other effects as too like positioning the elements in a mix.

The multi band compressor is an augmented compressor that provides flexibility to use multiple compressors in parallel with each one connected to a filter actin on different parts of the audio spectrum. The compressor for this course work contains three compressors coupled with a low pass and high pass filter and provides control over the low mid and high ranges of the audio signal individually.

Implementation:

For this course work, the multi band compressor was implemented using the JUCE and used example code from Reiss, Brecht de Man and Andrew McPherson from the DAFX textbook Chapter 6: Dynamics Processing.



Properties of a MultiBandCompressor:

Threshold: Defines at what level of the input signal the compressor should be activated. It the input signal is below the Threshold the audio signal is only affected by the makeup gain.

Attack Time: How fast the compressor starts affecting the input signal when the audio goes above the Threshold

Release Time : How fast the compressor stops affecting the input signal after the audio falls below the threshold value

Knee Width : Controls the strength of the ratio on the input signal. A harder knee width would make the compression ratio to act immediately on the signal and a softer knee width width would make the compression ration act gradually. In other words 'how smooth or star is the transition between the compression and the no compression point' (audio Effects: Theory, Implementation and Application By Joshua D. Reiss, Andrew McPherson. Pg;146)

MakeUp Gain : This is like a final gain stage for the compressor which can be used to Bring the signal level up or down after the compression has been applied. And this is implemented for each band of the multi compressor.

Input and Output Gain : Can be used to set the input and output gain of the audio signal.

Ratio: Ratio can be used to define how much compression will be applied on the signal. It defines how many parts in dB of the signal will be reduced for 1 dB of on increase in the signal level.

$$R = \frac{x_G - T}{y_G - T} \quad \text{for} \quad x_G > T$$

[1] Pg;145 fig 6.2

Xg is the input signal gain Yg is the output signal gain and T is the Threshold.

Compressor ON/OFF : used to activate or deactivate each compressor and its audio signal.

Low Band Frequency Cutoff: A filter cutoff for the low pass filter.
High Band Frequency Cutoff: Filter cut off for the High pass filter.

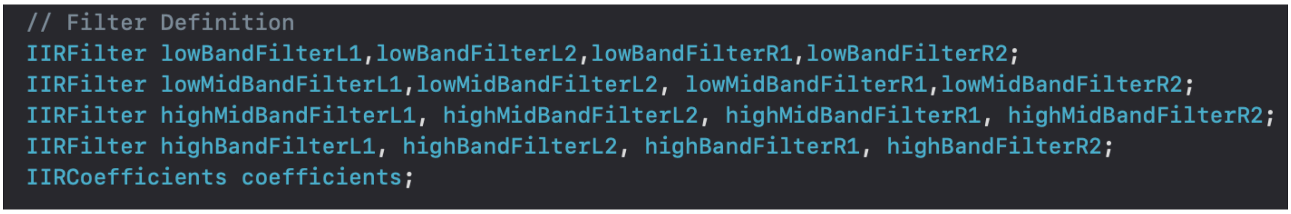
$$y_G = \begin{cases} x_G & x_G \le T \\ T + (x_G - T)/R & x_G > T \end{cases}$$

compression algorithm from [1] Pg;146 fig 6.3

The JUCE implementation is done in two parts inside the Plugin Processor and the Plugin Editor which define the DSP engine and the looks of the plugin respectively. All of the DSP is done inside the plugin Processor and the Knobs and Dials used to interact with the DSP are done in the plugin Editor.

Compressors work by splitting the audio into two signals. One signal is used to calculate a control voltage that is applies the appropriate gain reduction on the other signal which the compressed output signal. In this case the signals before reaching the compressor are copied into the inputs of three filters. Low pass , High pass, and a Band pass filter.

The filters were implemented using the IIR filter function provided by JUCE and the coefficients are defined by IIR coefficients function. Two filters are used in series for each band summing upto a 4th order filter applied on each band.



```
// Filter Definition
IIRFilter lowBandFilterL1,lowBandFilterL2,lowBandFilterR1,lowBandFilterR2;
IIRFilter lowMidBandFilterL1,lowMidBandFilterL2, lowMidBandFilterR1,lowMidBandFilterR2;
IIRFilter highMidBandFilterL1, highMidBandFilterL2, highMidBandFilterR1, highMidBandFilterR2;
IIRFilter highBandFilterL1, highBandFilterL2, highBandFilterR1, highBandFilterR2;
IIRCoefficients coefficients;
```

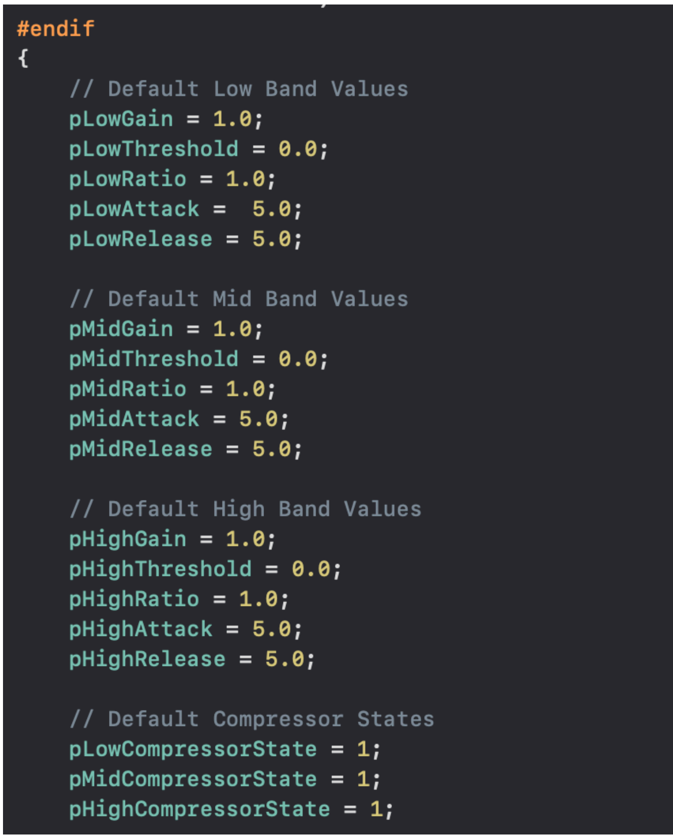
The low pass and high pass filter cut offs are used to determine the frequency response range of the mid band filter. all the parameters were initialised.

The audio processing is done in three different buffers the Low Output, Mid Output and High Output.

Gui Defentions and portions are done inside the Plugin Editor file using the predefined rotary knobs and button inside JUCE.

compressor.cpp

The compressor.cpp file provided by Reiss, Brecht de Man and Andrew McPherson provided all the maths that needed to be done on the samples inorder to bring the compression into effect. The code implements feed forward compression and uses peak sensing to apply gain reduction in the signal.
The below expression defines how the gain is calculated for different knee widths.
Xg Is the input signal gain
R is the Ratio and T is the Threshold
W is the knee width



```
#endif
{
    // Default Low Band Values
    pLowGain = 1.0;
    pLowThreshold = 0.0;
    pLowRatio = 1.0;
    pLowAttack =  5.0;
    pLowRelease = 5.0;

    // Default Mid Band Values
    pMidGain = 1.0;
    pMidThreshold = 0.0;
    pMidRatio = 1.0;
    pMidAttack = 5.0;
    pMidRelease = 5.0;

    // Default High Band Values
    pHighGain = 1.0;
    pHighThreshold = 0.0;
    pHighRatio = 1.0;
    pHighAttack = 5.0;
    pHighRelease = 5.0;

    // Default Compressor States
    pLowCompressorState = 1;
    pMidCompressorState = 1;
    pHighCompressorState = 1;
```

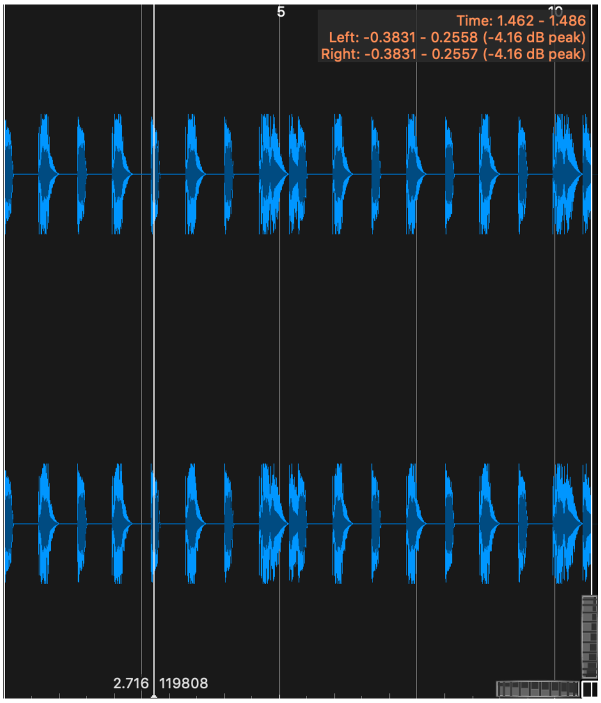
$$y_G = \begin{cases} x_G & 2(x_G - T) < -W \\ x_G + (1/R - 1)(x_G - T + W/2)^2/(2W) & 2|(x_G - T)| \leq W \\ T + (x_G - T)/R & 2(x_G - T) > W \end{cases} \qquad (6.4)$$

[1] Pg;145 fig 6.4

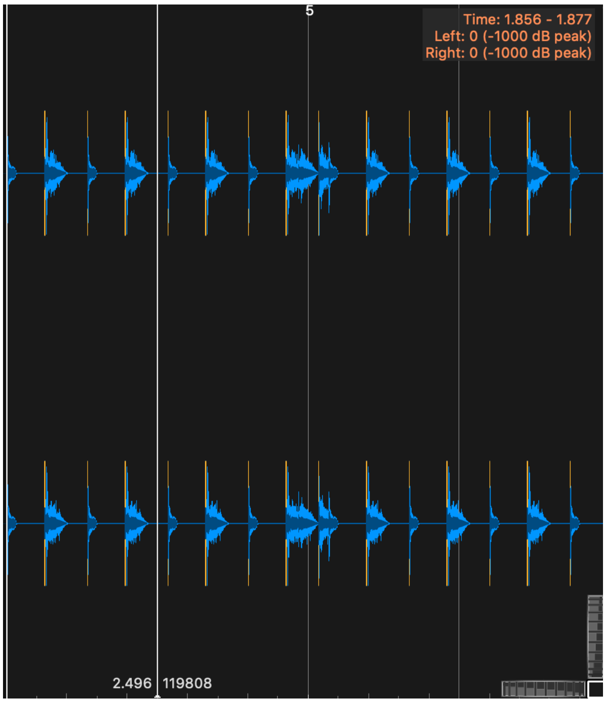
Evaluation:

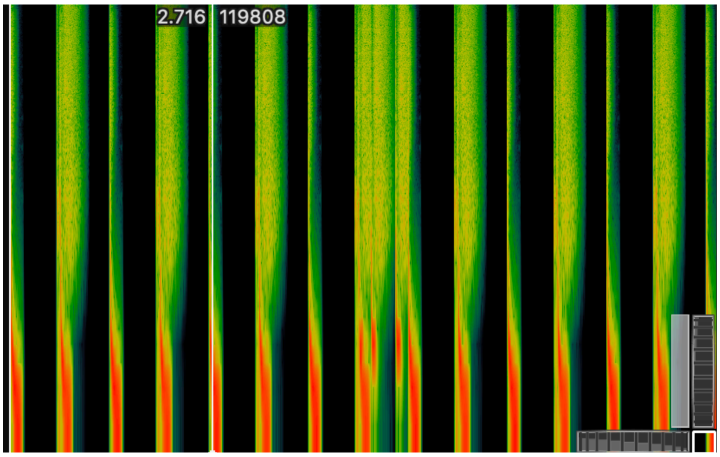
The VST plugin was hosted inside Ableton live and the audio signals were analysed using sonic visualiser, the images and values of the signals before and after compression are attached below. And the audio signals are attached in the zip file.

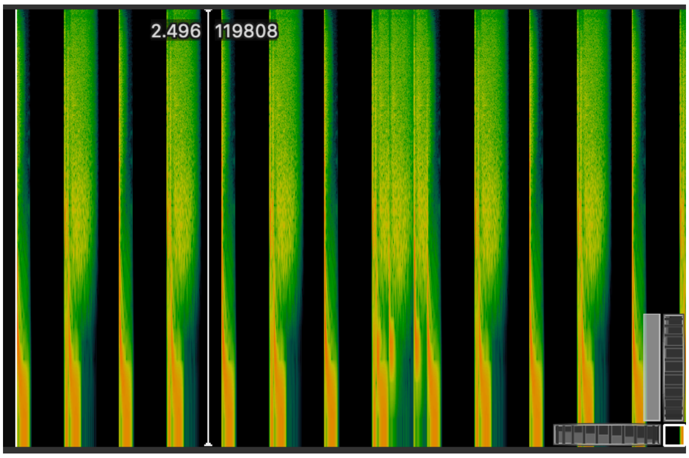
INPUT

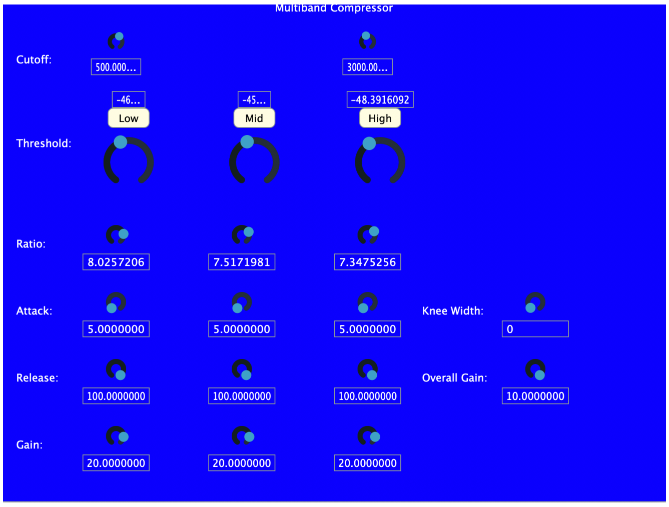

Time: 1.462 - 1.486
Left: -0.3831 - 0.2558 (-4.16 dB peak)
Right: -0.3831 - 0.2557 (-4.16 dB peak)

OUTPUT


Time: 1.856 - 1.877
Left: 0 (-1000 dB peak)
Right: 0 (-1000 dB peak)







Multiband Compressor

Cutoff:	500.000...		3000.00...		
	-46...	-45...	-48.3916092		
Threshold:	Low	Mid	High		
Ratio:	8.0257206	7.5171981	7.3475256		
Attack:	5.0000000	5.0000000	5.0000000	Knee Width:	0
Release:	100.0000000	100.0000000	100.0000000	Overall Gain:	10.0000000
Gain:	20.0000000	20.0000000	20.0000000		

The above results were obtained for the following compressor setting (as in image)

Low Threshold: -46
Mid Threshold: -45
High Threshold: -48

Low Ratio : 8.03

Mid Ratio: 7.52
High Ratio: 7.35

Release time was set to max and attack time was set to min and the knee width was set to minimum to ensure maximum effect of the control voltage on the output signal

A makeup gain of 20 db was applied to all the bands to increase the output signal loudness.

From the wave plot and the log spectrogram the signal has been considerably compressed which shows that the compressor works. In the wave plot a decrease In amplitude can be seen in all the bands.

But it can also be seen that the initial transient upto 5ms stays un affected by the compressor due to the minimum attack duration set to 5ms in the compressor settings. Even with a maximum release time of 100ms for this audio file the control voltage always went back to zero before the next transient hit. But around the 5 second mark it can be seen that the initial transients of consecutive snare hits have been rapidly affected by the release time. Lesser values were not experimented but it is known that smaller attack values might cause pops and clicks in the output audio signal.

Further evaluation was done by only activating one band to test each individual compressor the spectrograms and audio files are linked in the folder inside the zip file.

Conclusion: A working Multi band compressor was implemented and evaluated by referring to the Audio Effects: theory, Implementation and Application Text Book by Joshua D. Reiss, Andrew McPherson inside the JUCE framework.

References:

1. audio Effects: Theory, Implementation and Application By Joshua D. Reiss, Andrew McPherson.