Audio Engineering Society

Conference Paper

Presented at the Conference on Sound Field Control

2016 July 18–20, Guildford, UK

AmbiFreeverb 2 - Development of a 3D Ambisonic Reverb with Spatial Warping and Variable Scattering

Bruce Wiggins[1] and Mark Dring [1]

[1] *College of Engineering and Technology, University of Derby, UK*

Correspondence should be addressed to Bruce Wiggins (b.j.wiggins@derby.ac.uk)

ABSTRACT

In this paper, the development of a three dimensional Ambisonic reverb based on the open source Freeverb algorithm will be presented and discussed. This model is then extended to include processing in over-specified A-format, rather than B-format, variable scattering between channels along with controls for warping the distribution of the reflections to implement a reverb that is able to react to the source position in a spatially coherent way with an acoustical analysis of its performance.

1 Introduction

Ambisonics, as a spatial audio format, has been around since the 1970s, being pioneered by Michael Gerzon [1] as an improvement to the, then, state of the art system, Quadraphonics [2]. Work was carried out in the years that followed on music production techniques and equipment using Ambisonics, including the SoundField microphone which was able to the record four channels required for 1st order Ambisonic reproduction (for example see [3]). However, the tools needed to process and effect Ambisonic signals was rarely available or utilised. The advent of Steinberg's Virtual Studio Technology (VST) in 1996 [4] allowed for digital, software only, effects and processors to be created to be run on compatible host systems (initially Cubase VST, but others soon followed) making the creation of tools allowing for Ambisonic music production a more viable possibility. Many Digital Audio Workstations (DAWs) lacked the flexibility in channel routing needed to easily implement the Ambisonic mixing paradigm, but in 2005 work started on the software Reaper [5] which allowed for much more flexibility and control between channels within audio tracks, up to a limit of 64 channels per track.

Development of the WigWare suite of Ambisonic plugins began in 2006 with, initially, Ambisonic decoders for regular and the ITU 775 loudspeaker array [6], but this was soon expanded to include Ambisonic panners and, by 2008, a four channel Ambisonic Reverb (as no Ambisonic reverbs were available at this point) and higher order panners and decoders along with tutorial videos on their use in Reaper on YouTube [8] allowing the wider community to benefit from this work [7].

In 2014 an updated version of the Ambisonic Reverb (AmbiFreeverb 2) was released adding to the functionality of the earlier AmbiFreeverb from 2008.

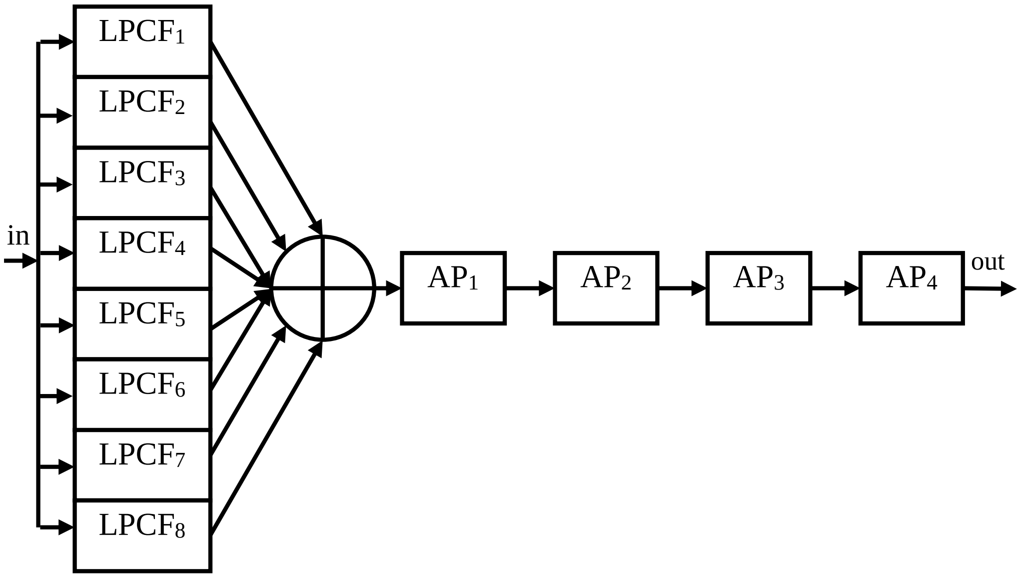

Figure 1 - Block diagram of one freeverb channel

2 Freeverb

The developed Ambisonic reverb was based on an extension of the excellent Freeverb, famously developed by 'Jezar at Dreampoint' in C++ as a VST plug-in. This plug-in uses four Schroeder all-pass filters and eight parallel Schroeder-Moorer filtered feedback comb filters for each channel (as shown in Figure 1) with the delay times being altered for each. Two of these channels were used, with altered time delays, creating the decorrelated audio.

3 Freeverb Components

3.1 Comb Filter

One of the two main echo generation blocks in Freeverb is based on a simple recursive comb filter taking the output after the delay line (so no impulse at 0 samples) as shown in (1).

$$CF(z) = \frac{z^{-r}}{1-fz^{-r}} \tag{1}$$

This will have an impulse and frequency response as shown in Figure 2 and Figure 3 where r represents the time between echoes and f the gain applied to the feedback.

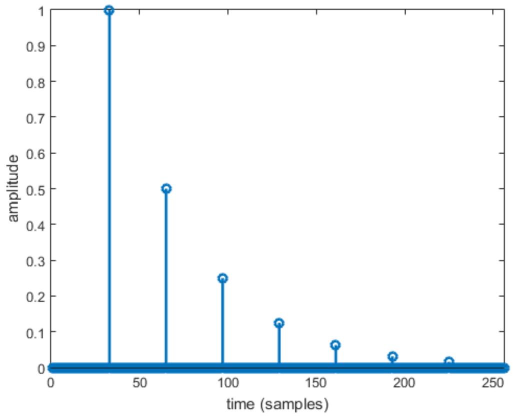

Figure 2 - Impulse response of a comb filter with 32 sample delay and a feedback gain of 0.5

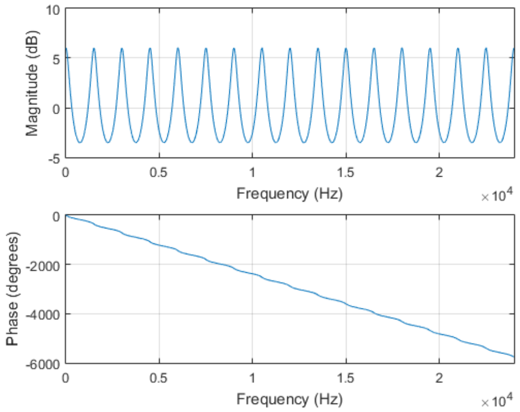

Figure 3 - Frequency and phase response of a comb filter with 32 sample delay and a feedback gain of 0.5

3.2 Acoustic Absorption Filter

This comb filter is then augmented with a simple one-pole acoustic absorption filter to help simulate the high frequency attenuation found in real reverberation and has a transfer function as shown in (2). The parameter *d* controls the amount of high frequency roll-off as shown in Figure 4.

$$LP(z) = \frac{1-d}{1-dz^{-1}} \tag{2}$$

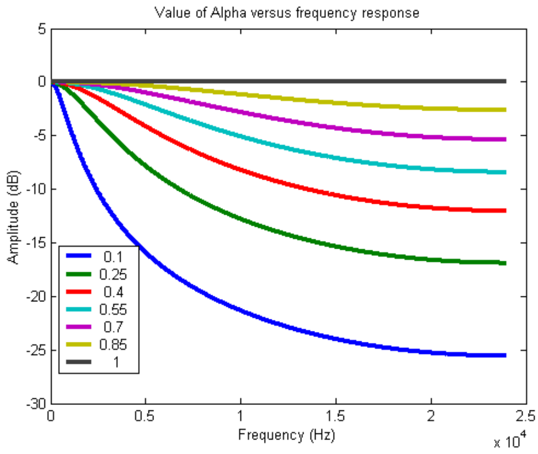

Figure 4 – Frequency response of the one-pole filter with varying values of (1-d).

3.3 Low Pass Feedback Comb Filter

These two transfer functions are then combined so the delayed samples in the comb filter have the one-pole low pass applied giving a transfer function as shown in (3) and an impulse and frequency response as shown in Figure 5 and Figure 6 using a delay of 32 samples, an *f* coefficient of 0.75 and a *d* coefficient of 0.25. Notice the time smearing created by the LP filter.

$$LPCF(z) = \frac{z^{-r}}{1-f\frac{1-d}{1-dz^{-1}}z^{-r}} \tag{3}$$

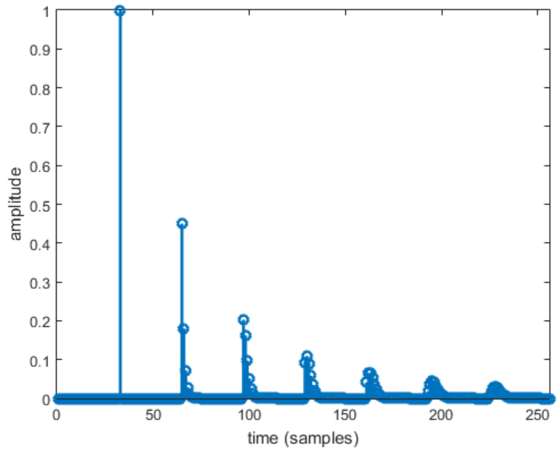

Figure 5 - Impulse response of a low pass feedback comb filter with 32 sample delay, feedback gain of 0.75 and damping coefficient of 0.25

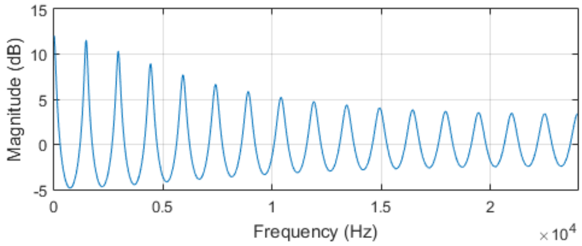

Figure 6 - Frequency response of a low pass feedback comb filter

3.4 All-pass Approximation Filter

As discussed in [9] the all-pass filter implemented in Freeverb doesn't actually exhibit an all-pass response. The transfer function for the all-pass implemented in Freeverb is shown in (4).

$$AP(z) = \frac{-1+(1+g)z^{-N}}{1-gz^{-N}} \tag{4}$$

An impulse and frequency/phase response of this filter using a delay (N) of 32 samples and gain (g) of 0.5 is shown in Figure 7 and Figure 8.

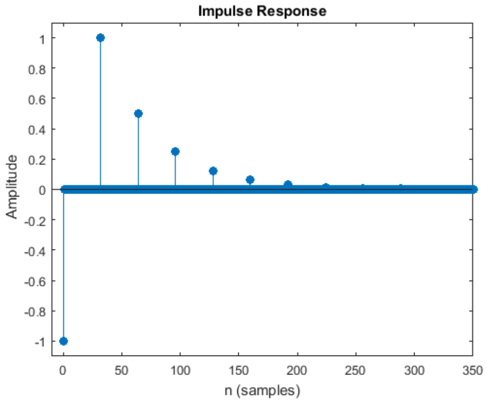

Figure 7 - Impulse response of the all-pass approximation used in Freeverb

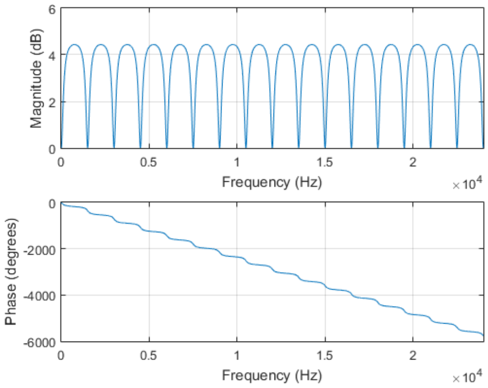

Figure 8 - Frequency and phase response of the all-pass filter approximation used in Freeverb

In the original Freeverb algorithm, the delay lines lengths of one channel were chosen empirically by the original developers, and then the delays for the 2nd channel used the same delay times, but positively offset by 0.5668ms.

```
Comb Filter Delay Times (ms)
25.31, 26.94, 28.96, 30.75, 32.24, 33.81,
35.31, 36.67
All-pass Filter Delay Times (ms)
12.61, 10.00, 7.73, 5.10
```

Figure 10 - Delay times used in Freeverb

The gain of the all-pass filters are fixed at 0.5, and the gain of the feedback comb filters is variable between 0.7 and 0.98 with the damping filter coefficient, d, being variable between 0 and 0.4. The plug-in also has a wet and dry gain variable and a spread control which allows for a mix between a mono and stereo gain (i.e. between fully correlated at both outputs and using the two decorrelated channels at full spread) with the input to the two decorrelated channels being a mix of the incoming Left and Right channels. Assuming outL and outR are the outputs from the two decorrelated reverb channels, the actual left and right outputs are calculated as shown in (5) [9] where *wet1 = spread* and *wet2 = 1-spread*, *dry* is the dry gain and *wet* is the wet gain.

$$\begin{bmatrix} outputL \\ outputR \end{bmatrix} = dry \begin{bmatrix} inputL \\ inputR \end{bmatrix} + wet \begin{bmatrix} wet1 & wet2 \\ wet2 & wet1 \end{bmatrix} \begin{bmatrix} outL \\ outR \end{bmatrix} \tag{5}$$

4 AmbiFreeverb

The original AmbiFreeverb extended this algorithm so instead of two decorrelated channels, there were four (one for each of the 1st order B-Format channels) which had the delay times offset by -0.4668ms for channel 3 (Y) and -0.96356 for channel 4 (Z). These were, again, empirically derived. The input was taken from the sum of the sound field's contents (the W channel), but the spread control now functioned slightly differently as shown in (6) where *wet1 = 2 – spread* and *wet2 = spread*, essentially trading the

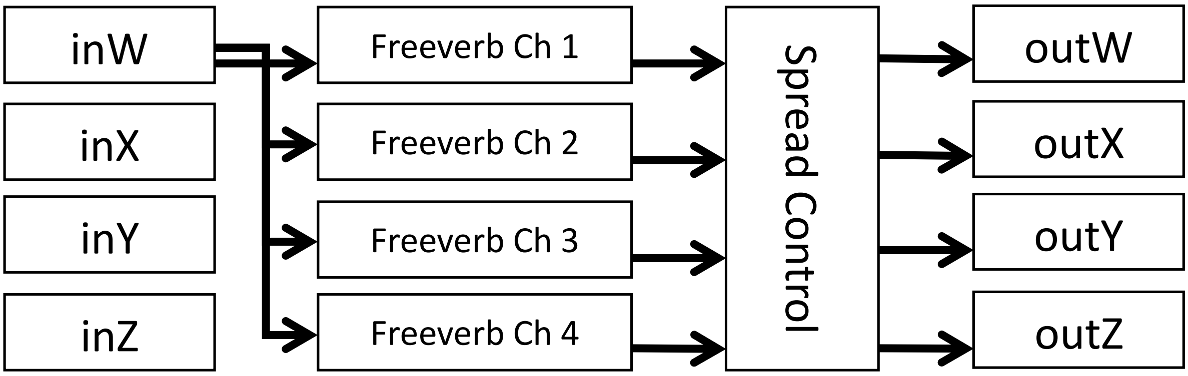

Figure 9 - Freeverb channel processing in AmbiFreeverb

directional components of the reverb off against the mono, W channel.

$$\begin{bmatrix} outputW \\ outputX \\ outputY \\ outputZ \end{bmatrix} = dry \begin{bmatrix} inputW \\ inputX \\ inputY \\ inputZ \end{bmatrix} + wet \begin{bmatrix} wet1 \times outW \\ wet2 \times outX \\ wet2 \times outY \\ wet2 \times outZ \end{bmatrix} \quad (6)$$

The fully *wet* version of AmbiFreeverb (i.e. dry set to 0 and wet set to 1) is shown in Figure 9.

This simple, essentially mono in, B-format out reverb worked well, but there were a few attributes which could be improved.

1. The reverb wasn't spatially aware, in that the panned position of the input made no difference on the output of the reverb.
2. There was no control of the spatial distribution of the reverberation, it was always equally everywhere.

In order to augment the algorithm, the reverberation was redesigned to apply the decorrelated channels to A-format, rather than B-format (as demonstrated in [11]). This meant essentially decoding the B-format to 4 channels that represented a response from a particular direction, containing a linear combination of the spherical harmonics (mirroring the tetrahedral arrangement of the capsules in the SoundField Microphone [10]) as shown in (7).

$$\begin{bmatrix} FrontLeftUp \\ FrontRightDown \\ BackLeftDown \\ BackRightUp \end{bmatrix} = 0.5 \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} W \\ X \\ Y \\ Z \end{bmatrix} \quad (7)$$

Transforming the B-format to A-format in this way allows the reverb channels to be applied directly to each one meaning the resulting reverberation will change depending on the direction of the original source. However, the side-effect of this arrangement is that the reverb will become more focussed in the direction of the source. To counter this, a scatter matrix as shown in [11] can be incorporated in the algorithm. The scattering matrix used in [11] is shown in (8).

$$0.5 \times \begin{bmatrix} 1 & -1 & -1 & -1 \\ -1 & 1 & -1 & -1 \\ -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & 1 \end{bmatrix} \quad (8)$$

This was adapted to include a variable (sc) in the range of 0 to 1 to allow for the amount of scattering to be controlled (9).

$$0.5 \times \begin{bmatrix} 2-sc & -sc & -sc & -sc \\ -sc & 2-sc & -sc & -sc \\ -sc & -sc & 2-sc & -sc \\ -sc & -sc & -sc & 2-sc \end{bmatrix} \quad (9)$$

The scatter matrix was placed after each iteration of the all-pass approximation filter across all four channels of the reverb engine which allowed for a transition between reverb concentrated in the direction of the sound source, to a more even, less directionally biased reverb.

Controls were also added for the spatial warping of the reverb in order to allow for the illusion of a listener positioned to the side of a room, or near an obstacle, for example. A number of transforms, to be carried out in B-format, have been proposed by various authors, with some good examples shown in [12]. However, although it is possible to apply a transform in a direction, it can be problematic when applying the transform more arbitrarily. For example, applying a shift in the X direction or the Y direction is straight forward, but applying both of these transforms equally, doesn't apply the resultant shift. In order to carry this out, a B-format rotation to the direction required needs implementing, followed by a shift in, for example, the X axis, followed by a reverse rotation to put the sound field back in the correct orientation. For this plug-in, a simple balance control was tested to allow for shifting in the Left-Right (Y), Front-Back (X) and Up-Down (Z) axis. A-format can be a more intuitive domain to affect the sound field. For example, taking the horizontal only case, if the front left quadrant was to be attenuated, the B-format could be transformed into directional A-format, the channel in the front left quadrant could be attenuated, and then the signal transformed back into B-format again, as shown in Figure 12.

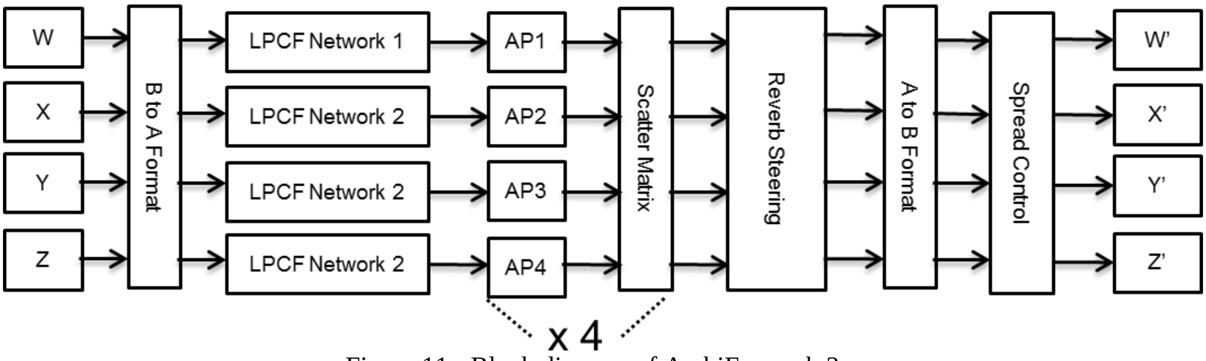

Figure 11 - Block diagram of AmbiFreeverb 2

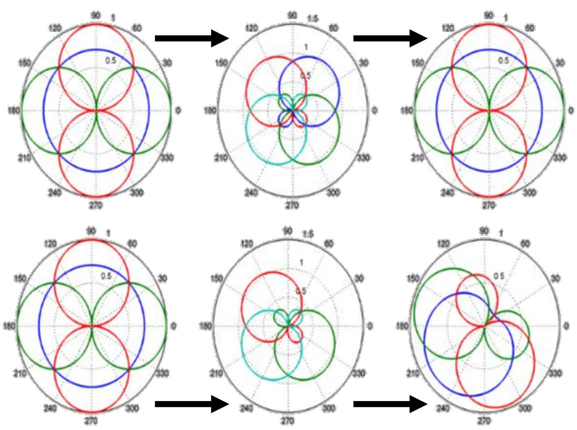

Figure 12 - B -> A -> B format, with and without front-left gain attenuation

The B-format signal is transformed to A-format, but using 8 channels, with the signals pointing at the corners of a cube. This is implemented by multiplying the B-format signal by the matrix shown in (10). This, over-specified, A-Format is necessary in order to allow for changes in each axis independently and in a straight forward manner.

$$\begin{array}{c} FLU \\ FRU \\ BLU \\ BRU \\ FLD \\ FRD \\ BLD \\ BRD \end{array} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & -1 \end{bmatrix} \times 0.5 \qquad (10)$$

where F = Front, B = Back, L = Left, R = Right, U = Up, D = Down

Each of the signals now comprises of an equal front/back, or left/right, or up/down component. If these signals are then multiplied by a gain coefficient of how much up/down, left/right or up/down signal is wanted, this will shift the sound field in this direction.

For example, the Front Left Up signal is multiplied by the desired front gain, left gain and up gain. Similarly, the Back Right Down signal is multiplied by the desired back gain, right gain and down gain with these gains derived using balance controls in the plug-in.

The final block diagram of AmbiFreeverb2 is shown in Figure 11.

5 Analysis

The variables available in AmbiFreeverb 2 are shown in Figure 13 .

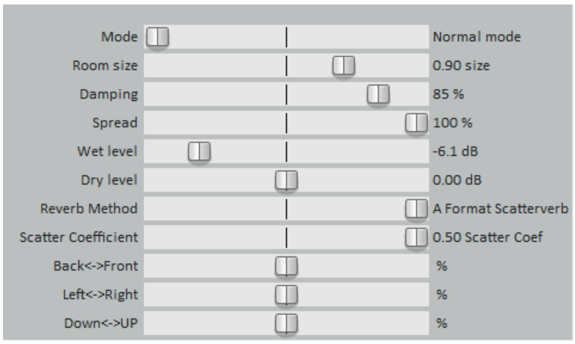

Figure 13 - AmbiFreeverb 2 parameters as shown in Reaper

In order to quantify and analyse the performance of the reverb, impulse response data will be obtained from the reverb under various parameter settings. The incoming impulse used to drive the plug-in can be set to represent an impulse from a direction as specified by the B-Format encoding equations (11). The reverb is then decoded to a 2D square virtual speaker array for analysis.

$$W[n] = \frac{1}{\sqrt{2}}\delta[n]$$
$$X[n] = \mathrm{r.}\cos(\theta)\cos(\phi)\,\delta[n]$$
$$Y[n] = \mathrm{r.}\sin(\theta)\cos(\phi)\,\delta[n] \quad (11)$$
$$Z[n] = \mathrm{r.}\sin(\phi)\,\delta[n]$$

Where r is how directional the impulse is (0->1), θ is the azimuth and ϕ is the elevation angle.

All testing is carried out with the wet gain at 0dB and the dry gain at -∞ dB.

5.1 Room Size

Altering the room size parameter affects the feedback coefficient in the low pass comb filters, and a value of 0 -> 1 linearly maps to a feedback coefficient of 0.7 -> 0.98. This, rather than increasing the room size, per se, actually alters the absorption coefficient of the walls, but the resulting rise of the RT60 makes the simulated room sound larger. The graphs in Figure 14 show the RT60 for each of the four outputs of a square decode with a room size parameter setting of 1, followed by bar charts representing the RT60 as the parameter is changed. Note that the each channel's delay lines length offsets can be observed with slightly shorter, or longer, RT60 times accordingly in these graphs. The damping parameter was set at 0.5 and the incoming impulse was omnidirectional (W only).

5.2 Damping Parameter

The damping parameter alters the low frequency cut-off of the low pass filter in the eight comb filters per channel. A value of 0 to 1 linearly maps to a value of 0 to 0.4 with each feedback loop of the comb filter being more low pass filtered than the last one, causing the low frequency damping to increase with time (as shown in Figure 16). Figure 15 shows the RT60 of a square decode, with room size at 0.5, with the damping changing from 0 to 1 in steps of one third. A low damping value gives the reverb a noise like quality that sounds unrealistic.

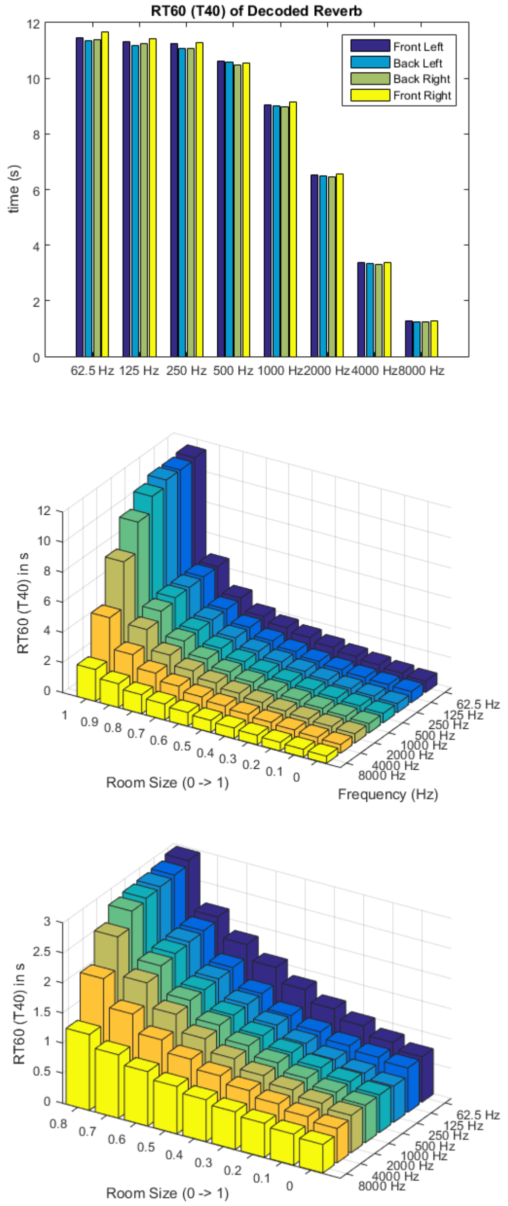

Figure 14 - RT60 times for AmbiFreeverb at various settings of Room Size (2nd bar graph is a rescaled version of the 1st for clarity – missing off the room size 1.0 and 0.9 settings).

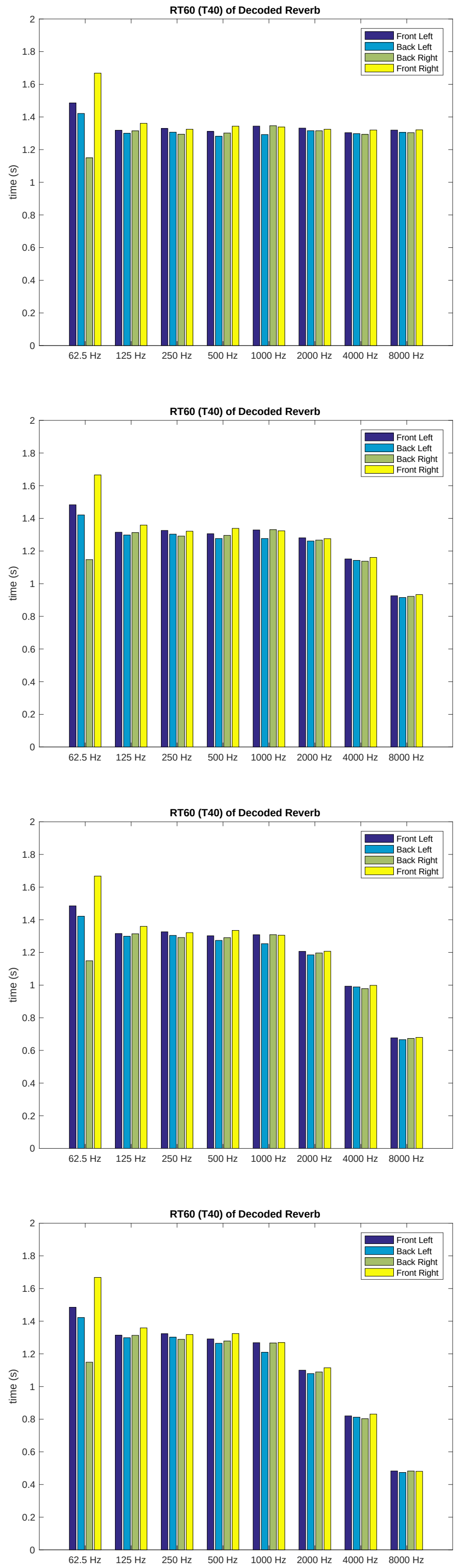

Figure 15 - RT60 of a square decode altering the damping coefficient from 0 to 1

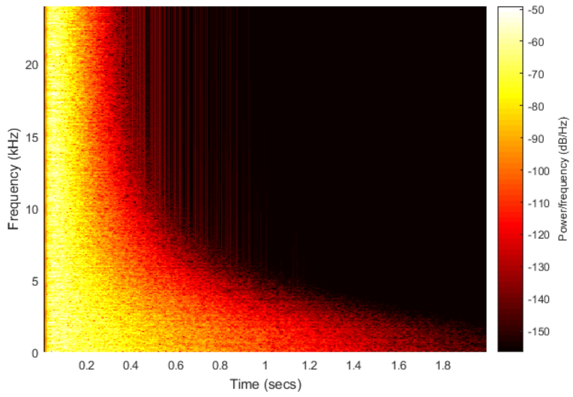

Figure 16 - Spectrogram of one channel of reverb showing how the low pass filtering increases with time.

5.3 Scattering Coefficient

The scattering coefficient, at a value of 0, results in the reverberation to be concentrated in the direction of the incoming source or, at a value of 1, to being maximally scattering between the channels by reflecting the reverb about each axis after each pass of the all-pass filters. Looking at the sound power in each channel, compared to the maximum, inputting an impulse from an angle of 45 degrees (front left) and adjusting the scatter coefficient from 0 to 1, we can see in Figure 17 that the quadrant opposite the source will rise quickest. Interestingly, at a coefficient value of 0.5, the resultant W signal will be 0 to then rise back to a level comparable to the X, Y and Z components when the scatter coefficient is 1 (originally, the plug-in was configured to use either 0 or 1).

The effect of the scattering coefficient can also be seen if the impulse is panned from 0 to 360 degrees using a scattering coefficient of 0 and 1. At a setting of 0 (no scattering), the channels show up to around 6dB of difference towards the location of the impulse and with a scatter coefficient of 1, this difference is reduced to around 1dB. If no difference at all is wanted, then AmbiFreeverb 2 can be set to work in the same mode as the original AmbiFreeverb (the 'mode' parameter) which derives each channels reverb only from the W, omni-directional, feed.

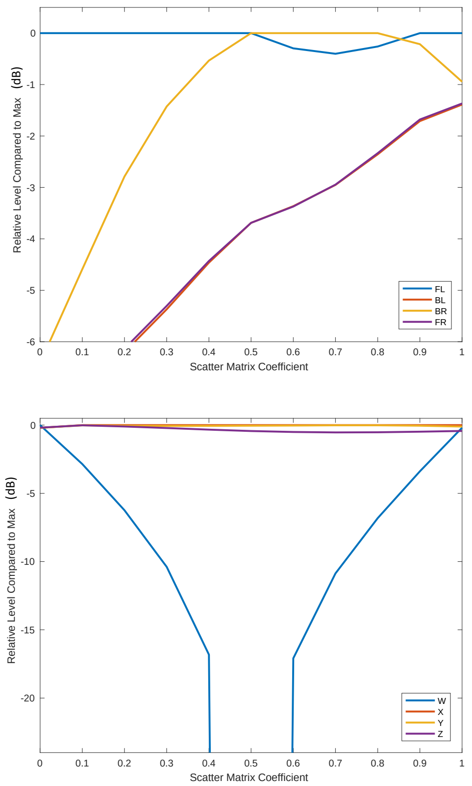

Figure 17 - The effect of the scattering matrix coefficient on a square decoders' outputs and the B-format channels

5.4 Sound Field Balance Controls

The sound field balance controls allow the level of the reverberation to be adjusted in each of the three axis. For example, feeding an omni-directional impulse (so all channel outputs are the same sound power) and altering the Left/Right balance the outputs of the square decode and the B-Format channels can be observed. Figure 19 shows the separation possible with this control at around 10 dB between the two left feeds and the two right feeds. The relative channel levels when altering both the Left/Right and Front/Back balance (between 0 and 1) simultaneously are shown in Figure 20 (i.e. changing the balance from back right to front left).

The controls included in the reverb allow for a variety of effects to be achieved. For example, if the balance was pushed to the left and the scattering was left at 0 it's possible to simulate a space with a live end and a dead end as the reverb is doubly attenuated (by the lack of scattering and the balance control) on the right side of the field and, as the reverb is positionally aware, this will occur depending on where the source is panned. Similarly, if the scatter coefficient is turned to 1, the reverb will sound more equally wherever the source is in the sound field, but will still be louder towards the left side (in this example), which could be a simple simulation of being in an offset position in a room.

Informal feedback on the sound of both AmbiFreeverb plug-ins has been extremely positive and the use of it, more recently, has expanded due to YouTube's newly released Spatial Audio feature which uses Ambisonics for the audio, decoded to binaural on Android phones taking into account the orientation of the phone/head in real time (for example, it features heavily in [13] and also in [14]). For more details on YouTube's Ambisonics to Binaural implementation, see [15].

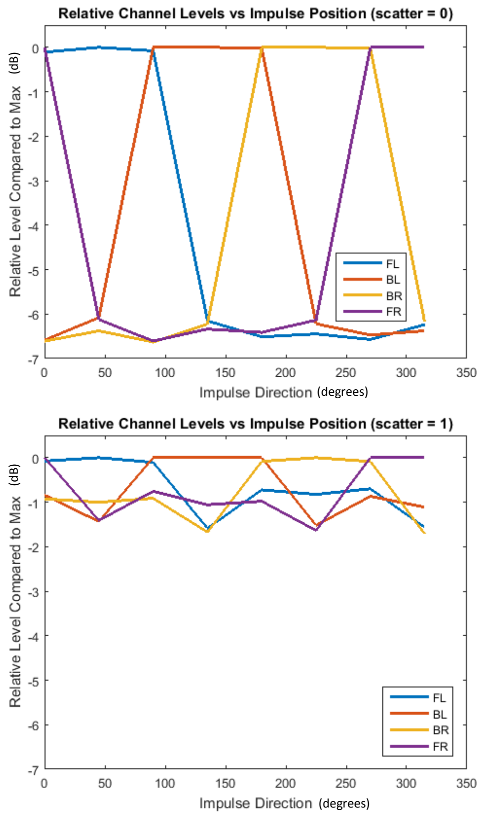

Figure 18 - Scattering coefficients of 0 and 1 when an impulse is panned from 0 to 360 degrees

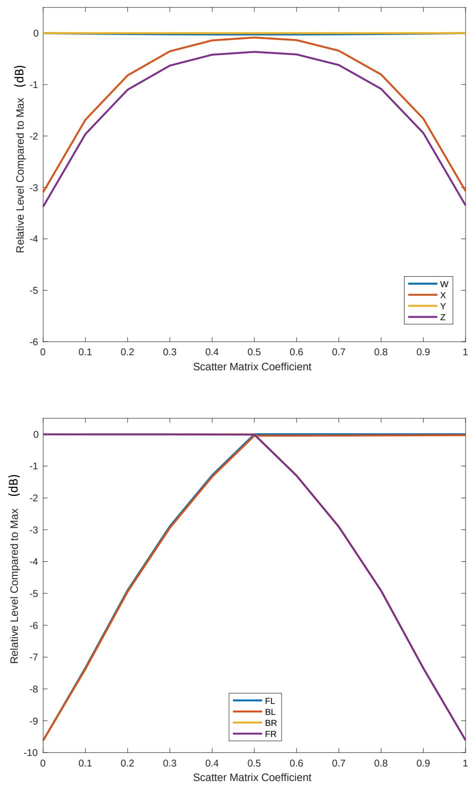

Figure 19 - B Format and square decoder channel outputs when later the left/right balance

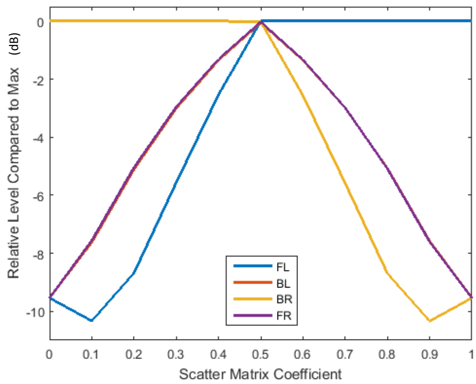

Figure 20 - Square decoder channel outputs altering balance from back right to front left

6 Conclusions/Further Work

A B-format, Ambisonic, reverb algorithm, based on Freeverb, has been described and its features analysed demonstrating some of the benefits of working in Ambisonic domain. AmbiFreeverb 1 was originally developed as no other B-format, 3D reverb VST plug-ins were available at the time (2008), and this paper describes some of the improvements to this earlier work in the scattering matrix, spatial balance controls, and processing in A-format allowing for more a more flexible reverb that changes depending on the position of the sound source. In terms of further work, there are a number of features that will be developed and investigated further. Addition of a pre-delay and also a room size parameter that affects the delay times (as opposed to the reflection coefficient), and the moving of the reverberation to a higher order model (more channels) will be carried out (particularly as YouTube and other VR formats move to support Higher Order Ambisonics). An early reflection model based on the image source method and further investigation into other scatter matrices will also be carried out.

7 References

1. Gerzon, M. A. (1974a) Sound Reproduction Systems. Patent No. 1494751.
2. Gerzon, M. A. (1974b) What's wrong with Quadraphonics. Available at: http://audiosignal.co.uk/Resources/What_is_wrong_with_quadraphonics_A4.pdf [Accessed 26 May 2016]
3. Elen, R. (1983). Ambisonic Mixing – an Introduction. [online] Ambisonic.net. Available at: http://www.ambisonic.net/ambimix.html [Accessed 17 May 2016].
4. Soundonsound.com. (1996). *Steinberg Cubase VST*. [online] Available at: http://www.soundonsound.com/sos/1996_articles/jul96/steinbergcubase3.html [Accessed 23 May 2016].
5. Cockos Inc. (2005) *Reaper | About*. Retrieved: May, 2016 from http://reaper.fm/.
6. ITU-R BS. 775 (1994) Multichannel Stereophonic Sound System With and Without Accompanying Picture
7. Wiggins, B. (2014). Improved Spatial Audio from Ambisonic Surround Sound Software. Impact Case Study submitted to the Research Excellence Framework (REF) 2014 [online] Available at: http://impact.ref.ac.uk/CaseStudies/CaseStudy.aspx?Id=9508# [Accessed May 2016]
8. Wiggins (2008). Ambisonics Audio Production using Reaper and WigWare. [online] Available at: http://tinyurl.com/WigWareVids
9. Smith, J.O. Freeverb - Physical Audio Signal Processing,

http://ccrma.stanford.edu/~jos/pasp/Freeverb.html , online book, 2010 edition, accessed May 2016.

10. Wiggins, B., Spenceley, T. (2009) Distance coding and performance of the mark 5 and st350 SoundField microphones and their suitability for Ambisonic reproduction. Reproduced Sound 25 – Proceeding of the Institute of Acoustics, Vol 31, Pt 4.
11. Anderson, J., Costello, S. (2009) Adapting Artificial Reverberation Architectures for B-Format Signal Processing. Ambisonic Symposium 2009, June 25-27, Graz.
12. Anderson, J. (2009) Introducing…the Ambisonic Toolkit. Ambisonic Symposium 2009, June 25-27, Graz.
13. Leusink, A. (2016). *Tomas Doncker & True Groove All Stars "The Mess We Made" - LIVE! in 360º video and spatial audio*. [online] YouTube. Available at: https://www.youtube.com/watch?v=wmViONWyGSA [Accessed 20 May 2016].
14. Michael, B., Phil, B., Marc, B. and Stuart, P. (2016). *The Shape Of My Thoughts*. [online] YouTube. Available at: https://www.youtube.com/watch?v=i3TjIiaKDVU [Accessed 20 May 2016].
15. Wiggins, B. (2016). *YouTube 360 VR Ambisonics Teardown!*. [online] The Blog of Bruce. Available at: http://www.brucewiggins.co.uk/?p=700#content [Accessed 20 May 2016].