FACULTY OF ARCHITECTURE, DESIGN & PLANNING

ASSIGNMENT COVER SHEET

POSTGRADUATE

FAMILY NAME: EDSER **GIVEN NAME:** CARL SCOTT

SID: 430488700 **TELEPHONE NO:** 0409028990

UNIT COORDINATOR: BILL MARTENS

TUTOR: LUIS ALEJANDRO MIRANDA JOFRE

UNIT OF STUDY (Code and name): DESC9115 – DIGITAL AUDIO SYSTEMS

COURSE: MASTERS OF DESIGN SCIENCE (AUDIO AND ACOUSTICS

TITLE OF ASSIGNMENT: WRITTEN REVIEW 2

DATE DUE: 7th JUNE 2013 **DATE SUBMITTED:** 7th JUNE 2013

This cover sheet must be attached to Assignment

THE UNIVERSITY OF SYDNEY

Written Review 2

Carl Edser
Student # - 430488700

DESC9115 – Digital Audio Systems
William Martens
Assessment Task 4
Due Semester 1, Friday 7th June, 2013

Table of Contents

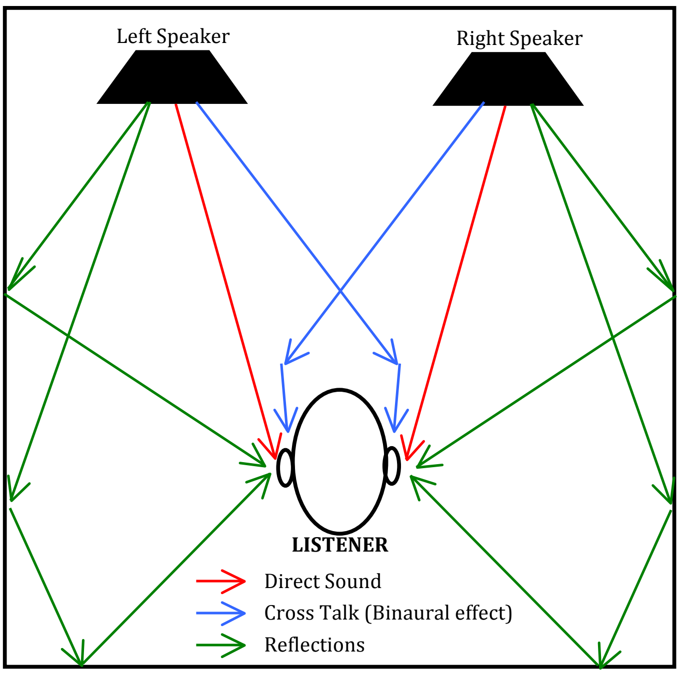
1. Introduction 3
1.1 What is Reverberation? 3
1.2 DSP Reverberation Models 3
1.2.1 Early Reverberation Models 4
1.2.2 Algorithmic Reverberation 4
2. Convolution Reverberation 5
2.1 Convolution 5
2.2 The Convolution Equation 5
2.2.1 Simple Convolution Example using Integer Values 6
2.3 Obtaining The Environment's Impulse Response 7
3. Working Convolution Model 7
4. Conclusion 9
5. Reference List 10

1. Introduction

1.1 What is Reverberation?

Reverberation is best described as the persistence of sound in a space, originating from a source. It exists when sound propagates and interacts with the physical and geometric conditions of the environment (Rocchesso, 2002, p.137). When an impulse is produced, say a clap for example, it may be reflected, absorbed or transmitted by any material or surface surrounding its point of origin. This singular, sharp impulse of broadband noise, generalised by the delta function $\delta[x]$, excites the room with reflections and ambience, forming what we hear as reverb.

In the natural listening environment, many of the early reflections will convolve not long after the direct signal, or continue to interact with other surroundings until the sound has decayed or been fully absorbed (Giesbrecht et al, 2009, p.3). A reverberant model is comprised of an initial impulse, pre-delay, early reflections and decay time or RT_{60}. As seen in *Figure 1b*, it is possible to construct a reverberant model from a magnitude of impulse responses and discrete delay lines. With the reflections travelling a further distance, and each surface presenting its own relative absorption coefficient, a reduction in amplitude will result. RT_{60} is an expression commonly used amongst audio and acoustic engineers and simply refers to the time it takes for the reverberant field to decay 60dB from the original source amplitude.



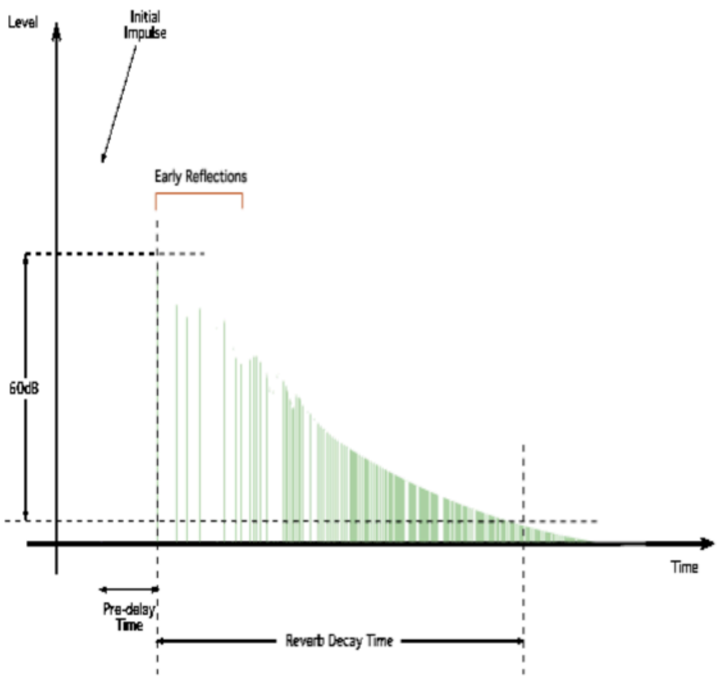


Figure 1a (left): Direct sound path and how the acoustics may *hypothetically* interact with the surrounding environment (Beringer, 2011, p.2)

Figure 1b (right): Impulse response of a room demonstrating the pre-delay, early reflections and RT60 required for the creation of a reverberant space (Senior, 2000)

1.2 Reverberation Models

1.2.1 Early Reverberation Models

Before digital signal processing somewhat simplified the recreation of artificial reverberation, it was generated and captured via natural acoustics, or mechanical methods respectively. Although natural acoustics is reasonably self-explanatory, many performance spaces, such as the Sydney Opera House, were designed with reverberation attributes in mind, to enhance the sonic qualities of the performance within.

Mechanical devices used transducers or other electromagnetically induced principles to vibrate or alternate the current of an input signal in a material such as a metal plate or spring (see *Figure 2*). One or several pickup transducers were then used to capture the reverberated signal generated by these materials. The reverberant signal generated would be re-amplified, before being combined with the original signal. The behaviour of these mechanical devices was dependent on the geometry, material, electrical/conductive properties and frequency response patterns.

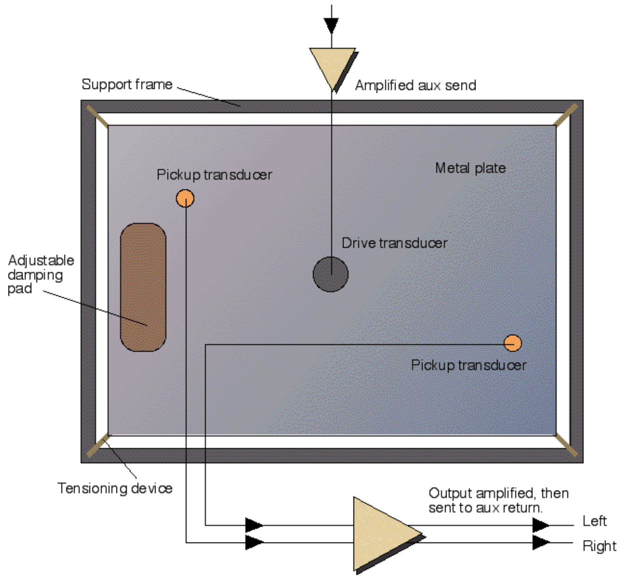


Figure 2: Signal flow and general components of a plate reverb unit (Sound On Sound, 2001)

In digital signal processing, it has become possible to simulate any reverberant environment through algorithmic or convolution calculations. Although algorithmic calculations of a reverberant image allow for more versatile and simplified CPU processing, they lack the specific spatialisation cues associated with convolution models (Stewart & Murphy, 2007, p.1). As a result, convolution presents a more 'natural' sounding simulation of the reverberant environment. A variety of algorithms have been developed for the recreation of artificial reverberation. However, the primary focus of this document will be to discuss the methods behind the development and implementation of reverberation that takes advantage of FFT (Fast Fourier Transform) convolution.

1.2.2 Algorithmic Reverberation

Algorithmic, or filterbank reverberation hypothetically models an artificial, acoustic space. Simulation of the early reflections, compounding delays and varied frequency decay of the artificial environment become adjustable parameters within the DSP, unlike that of most convolution reverberation devices. The ability to adjust the aforementioned parameters mean that an algorithmic reverb unit can become more versatile than convolution types when applied to the input signal, allowing the user to model a desired space.

2. Convolution Reverberation

2.1 Convolution

Convolution is a mathematical procedure in which two signals are combined to form a third signal (Smith, 1997, p.107), or as Rocchesso states, a generic signal processing operation like addition or multiplication (2002, p. 48). It is the single most important technique of digital signal processing and is required for the construction of convolution reverb.

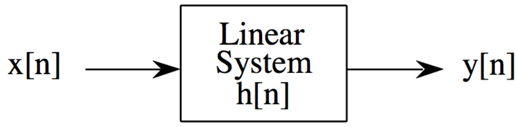
Convolution reverb is created from a series of impulse responses (IR) captured from the selected reverberant space (Duesenberry, 2005) and is convolved with the input signal. Once the 'blueprint' of that reverberant space has been captured, it is then possible to emulate an audio source as being heard in the same environment from which the IR was taken. For example, if an impulse response of the Sydney Opera House were captured, it would be possible to simulate any recording, from any studio, home or otherwise, to sound as though it were performed in that environment. Existing software that currently offers this type of simulation is the Audio Ease, AltiVerb plug-in, commonly used with production software such as Avid's Pro Tools (see *figure 2*).

Figure 3: Audio Ease - AltiVerb 7 (ProAudioBoutique, 2012)

2.2 The Convolution Equation

In most linear systems of convolution, *x[n]* and *y[n]* are terms respectively given to the input and output signals of a DSP. If the input remains unchanged by the DSP, it would be fair to say that *x[n] = y[n]*. As previously mentioned, generating convolution reverb requires the input to be modified by a series of impulse responses, for which *h[n]* would be used. Therefore, convolution in the time domain is often represented mathematically by the formula;

$$y[n] = x[n] * h[n]$$



where * means ‘convolved with,’ and not ‘multiplied by’ as used in general mathematic equations (Smith, 1997, p.109).

2.2.1 Simple Convolution Example using Integer Values

A simplistic way of understanding convolution is by using a string of integer values to represent the input, impulse response and output signals. The following example has used both Smith’s (1997, p.112) and Mitra’s images (2007, p.72) for guidance.

If a sound or input signal is represented by the vector;

```
x[n] = [1, 6, 5, 4]
```

and we have an impulse response for h[n] of;

```
h[n] = [3, 4, 2]
```

the first operation of convolution would resemble;

```
x[n] * h[1] = [1, 6, 5, 4] * [3] = [3, 18, 15, 12]
```

the signal x[n] is delayed by one sample for the second step of convolution;

```
x[n-1] = [0, 1, 6, 5, 4]
```

and multiplied;

```
x[n] * h[2] = [0, 1, 6, 5, 4] * [4] = [0, 4, 24, 20, 16]
```

likewise for the third step of convolution (x[n] has already been delayed);

```
x[n] * h[3] = [0, 0, 1, 6, 5, 4] * [2] = [0, 0, 2, 12, 10, 8]
```

As there are no more impulse response values remaining, the three vectors are aligned;

```
[3, 18, 15, 12]

[0,  4, 24, 20, 16]

[0,  0,  2, 12, 10, 8]
```

with zeroes being postpended in the first and second rows;

```
[3, 18, 15, 12,  0, 0]

[0,  4, 24, 20, 16, 0]

[0,  0,  2, 12, 10, 8]
```

Finally, all columns are added together, convolving the input signal x[n] with the impulse response h[n], obtaining an output value for y[n] of;

```
[3, 22, 41, 44, 26, 8]
```

2.3 Obtaining the Environment's Impulse Response

In order to convolve the input signal with the desired reverberant environment, the impulse response of the chosen space must first be captured. This is effectively achieved by generating a short burst of white noise (starter pistol/clap) within the environment and capturing the space with a microphone that presents a linear response (flat frequency response). Ultimately, a sweeping sine wave, played and captured through a linear speaker/microphone system, provides greater detail and accuracy of the reverberant environment, as well as better signal-to-noise ratios. However, this requires the room's response to be deconvolved from the impulse response, using significantly more processing power.

The recorded impulse response of white noise or a swept signal creates nothing more than an audio file, which can then be convolved with an input signal using software such as Matlab. This provides the system with a series of discrete impulses that are not only relevant to the reverberant characteristics of the space, but also the position of pickup source within it. It is important here to note that the reverberant characteristics captured will vary with different positioning of the pickup source, so multiple 'blueprints' of the space may be ideal to record if versatility in the design of the DSP is required.

3. Working Convolution Model

McGovern's function 'fconv.m' has been implemented and modified to provide the reader with a visual representation of convolution reverb using Matlab software (2004). All of the graphs represented on the following page are shown in the time domain.
In the first graph, we can see the waveform of a basic drum pattern (mono). This was simply created and bounced to a mono .WAV file using Logic Pro, which contains a dry and otherwise unprocessed signal.

The second graph has utilised McGovern's own impulse response recording of a cathedral (location unknown) that can be convolved with any mono input. A note here is that a stereo file could be also used if Matlab is told to read the audio input as a mono file. This is simply done by using;

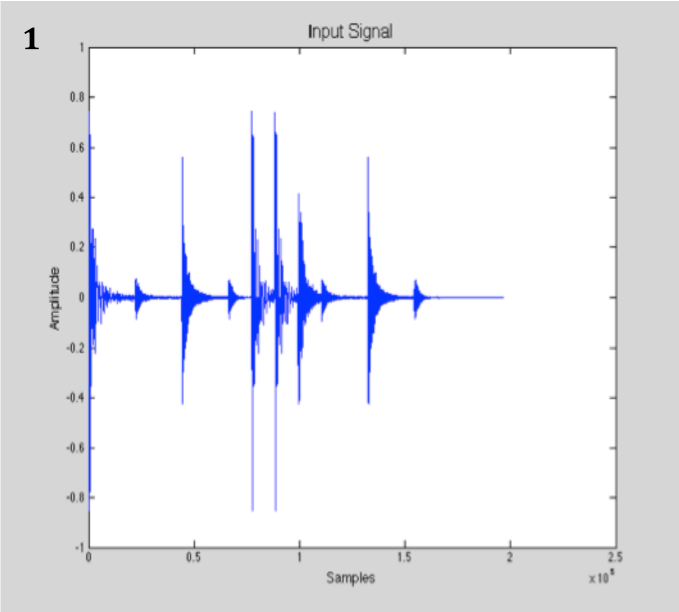
```
[x,fs] = datain(:,1).
```

It should also be noted that the impulse response in the second graph has been taken over $3x10^4$ samples on the x-axis, unlike $2.5x10^5$ for the other graphs.

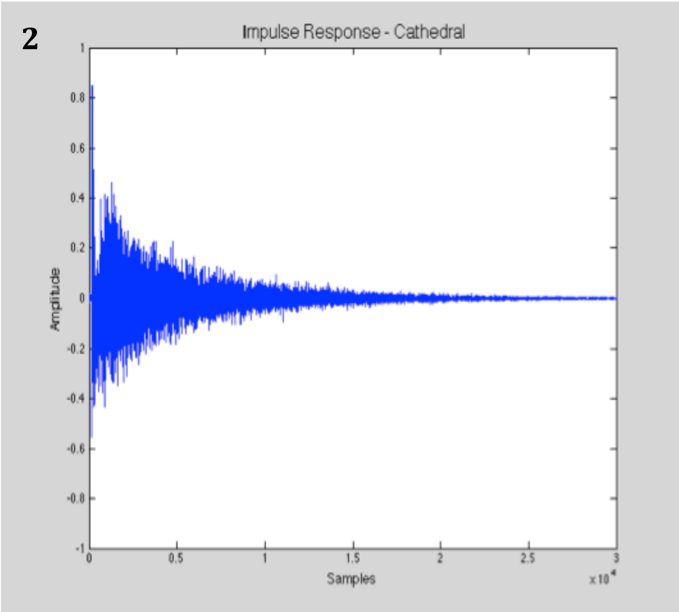
Graph 3 has been obtained after the convolution function was applied to the input signal. This shows the formation of reverberant tail as a result of the input being convolved with the impulse response.

The relationship between the input and reverb created from the impulse response is clearly evident in the final graph where Graph 1 and 3 have been layered on each other.

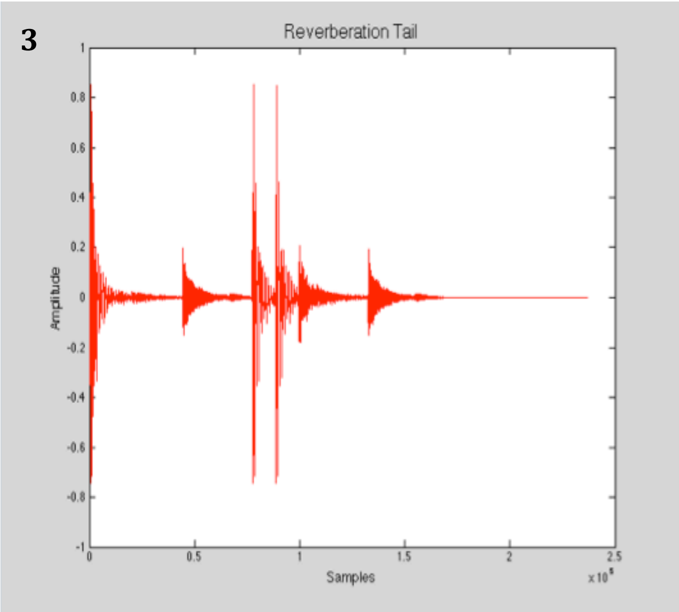
1



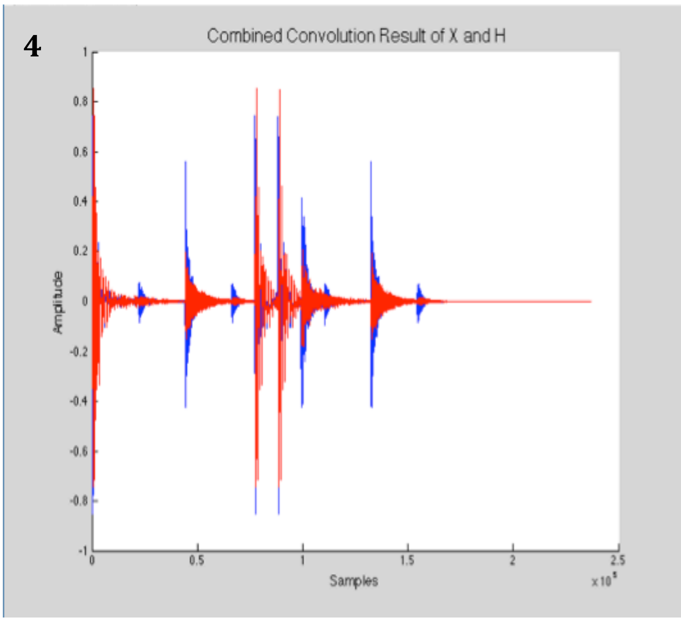
2



3



4



4. Conclusion

Reverberation is an integral part of compositional performance and has been for an extensive period. Its transition and development in the digital realm has allowed audio engineers the ability to simulate an environment, without the necessity to record or perform in the desired environment. Although some may prefer the authenticity of a real environment, the ability to take the impulse response from a desired space and convolve it with an input signal has significantly reduced production costs, as well as saving time. The example from 2.2.1 is a concise approach for understanding the convolution process, as vector lengths for even a short segment of audio can be quite extensive (the input in the 'Working Convolution Model' section alone contains roughly $2x10^5$ samples and an IR of $3x10^4$ samples).

5. Reference List

Beringer, P 2011, *Spatial Sound, Reverberation and Acoustic Emulation in Digital Audio Systems* [online image, PDF], viewed 30th May 2013, University of Sydney, Camperdown, Sydney, Australia. http://web.arch.usyd.edu.au/~wmar0109/DESC9115/old_stuff/reviews_2011/PBeringer%20SID%20200217730.pdf

Duesenberry, J 2005, *Sound Design Workshop: Convolution Reverb and Beyond - Broadband Audio Files as Impulse-Response Sources*. Electronic Musician, 21(4), pp. 76-76, NewBay Media, New York, N.Y, United States.

Giesbrecht, H, McFarland, W & Perry, T 2009, *ELEC 407, DSP Project: Algorithmic Reverberation - Combining Moorer's reverberator with simulated room IR reflection modeling*, University of Victoria, Saanich, Canada.

McGovern, S 2004, *A Model For Room Acoustics*, viewed 6th June 2013, Stevens Institute of Technology, New Jersey, United States.
http://www.sgm-audio.com/research/rir/rir.html

Mitra, S 2001, Digital Signal Processing, A Computer Based Approach, McGraw-Hill, 2nd Edition, California, United States.

ProAudioBoutique 2012, *Audio Ease Altiverb 7*, [online image] viewed 3rd June 2013, http://proaudioboutique.com/product/audio-ease-altiverb-7/

Rocchesso, D 2002, *DAFX – Digital Audio Effects*, John Wiley & Sons, Ltd, West Sussex, England.

Senior M, 2000, *Sound on Sound; Reverb – Frequently Asked Questions*, [online image] viewed 3rd June 2013, http://www.soundonsound.com/sos/may00/articles/reverb.htm

Smith, SW 1997, *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, San Diego, California.

Sound On Sound 2001, Understanding & Emulating Vintage Effects: Sound Techniques [online image], viewed 6th June 2013 Media House, Cambridge, United Kingdom.

Stewart, R & Murphy D 2007, 'A Hybrid Artificial Reverberation Algorithm,' *Audio Engineering Society Convention Paper: 7021*, 122nd Convention, Vienna, Austria.