Artificial Reverberation: An Algorithmic Approximation of a Physical Model

Kevin Murning and Kelsey Kaplan
EEE4114F
Project Report

***Abstract*—In this paper we investigate the use of algorithmic implementations of artificial reverberation to approximate the behaviour of an existing physical model, namely a recorded impulse response of a real world reverberant environment. A series of metrics are developed in order to evaluate the performance of the algorithms when tested against the physical model, and finally, existing algorithms will be modified based on the results of the comparison in an attempt improve algorithmic performance.**

I. Introduction

Artificial reverberation is the practice of simulating the change in character of an audible signal as observed by a listener when sound waves are propagated through an acoustic environment. The perception of reverberation by a listener occurs naturally when a sound source is positioned in a reverberant room and projected outwards, effectively being filtered and delayed by reflections off of and absorption by structural elements of the environment. At a high level, this can be modelled as a simple point to point system, where the signal propagated by the source is processed by an appropriate transfer function modelling the room, which outputs a reverberant sound. However, a model that has a closer approximation to reality is that of multiple sources (for example, a band) as perceived by at least two receivers (the human ears). Each receiver experiences differing delays and filters which change dependent on the position of the receiver in physical space. It is theoretically possible to implement this model, but given the fact that most reverberation effects need to be processed in real time, its practical usefulness is limited by the prohibitively expensive computation required in its implementation [1].

The problem of artificially generating this effect is formulated as the ability to use an algorithm to synthesize reverberation more economically, whilst still maintaining the perceptual effects of sound in an acoustic space. This requires a movement from physical modelling to perceptual modelling - a process of identifying what is actually audible to the human ear and developing metrics to evaluate such. Once identified, these perceptual elements can then be used to parameterize reverberation, and thus allow a designer to tune an algorithmic implementation to approximate the perceptual effects of a physical model.

II. Problem Specification

The following report addresses the problem of approximating a physical model of reverberation using perceptual modelling. Our task is to evaluate a series of canonical algorithmic reverberation implementations and their effectiveness in approximating a series of desired perceptual parameters. This will be evaluated using a number of metrics developed to quantify the perceptual aspects of reverberation. In the sections to follow, we will implement and test a simple physical model, followed by some classical reverberation algorithms. Based on our findings, we will attempt to combine existing methods to improve upon some of the perceptual shortcomings of the existing implementations.

III. Literature Review

An exact representation of the reverberation of an acoustic environment can be obtained via transfer function modelling [1]. If we consider a room to be an LTI system, ignoring the effects of sound absorption in air, the reverberation of a space can be represented as an impulse response. The sound source and the acoustical pressures at the listener's eardrums are representative of the inputs and outputs of the LTI system [2]. However, a transfer function model developed from this framework is not generalizable to all points in the space. The movement of a listener from one point to another will change the perceived impulse response, and thus the consideration of points of reference in this approximation is important [1]. To obtain an impulse response, impulsive sound sources such as a balloon pop or pistol shot can be recorded. Many data sets of real room impulse responses exist [3]. Direct convolution with one of these impulse responses can yield highly realistic results, however, this does not solve the problem of computational complexity [1].

The perceptual aspects of reverberation can be segmented into early reflections and late reverberation [4]. As a sound wave propagates through space, the listener will first hear the direct sound, followed by early echoes stemming from reflections off of the closest surfaces. This generally lasts approximately 100ms [4]. Finally as the number and directions of reflections increase, a dense collection of decaying echoes will be perceived [2]. Since early and late echoes have different physical and perceptual qualities, they are generally dealt with separately [2]. The early reflections and late reverberation components of an impulse response are shown in figure 1.

Early reflections are characterized as a set of attenuating and delayed impulses. When modelling these reflections one must take into account the effects of sound absorption as well as the ratio of direct to reverberated sound energy [1]. They are

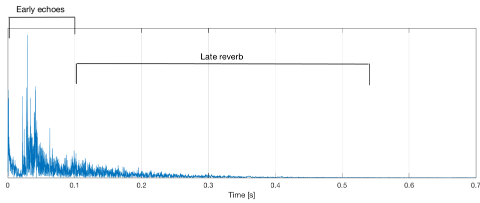

Fig. 1: Early and Late Reverberation Distinction

generally implemented using tapped delay lines and lowpass filters to mimic room and air absorption [4].

Late reverb is described as a diffuse sound field, characterised by its decay time and frequency response envelope [2]. The echo density of late echoes, as well as having a flat amplitude frequency response is imperative to the quality of the late reverberation [5]. In comparison to the relatively sparse collection of echoes in the early reflections, the echos present in late reverberation are so densely populated that it is best to characterise them using some form of statistical distribution. This is a further step that can be taken to reduce the computational complexity of exact reverb modelling [1]. Schroeder suggested in his early paper on algorithmic reverb that one could use a white noise impulse response to characterise late reverberation [5].

There exists a number of useful evaluation criteria for determining the perceptual quality of algorithmic reverb. These can be conveniently linked to the concepts of early and late reverberation as outlined above. It can be shown that for typical acoustic environments, echo density increases as t^2, where t is time [6]. Therefore, after some time the echo density will be so great that it can be modelled as some uniformly sampled stochastic process without loss of audio fidelity [1]. Echo density is quantifiable and can be used to separate and evaluate early and late reflections. Similarly it can be shown that the number of resonant modes in any given frequency band increases as frequency squared [1]. This means that above some frequency, the modes are so dense that they are perceptually equal to a random statistically generated frequency response. Resonances are so densely packed during the late reverberation stage that their separation is inaudible. This is implemented in practice using feedback comb-filters [1].

Another useful metric is the reverberation time, otherwise known as t_{60}. This is the time that the energy of the impulse response takes to decay to $-60dB$. Schroeder introduced the energy decay curve (EDC) as a means of evaluating t_{60}. However, perceptual studies have shown [1] that the most realistic sounding reverberation algorithms present different t_{60} values in different frequency bands. Smith introduced the energy decay relief (EDR) curve [1] which provides a useful metric with which one can tune t_{60} in differing frequency bands to achieve desired perceptual results. Echo density, EDC, EDR and t_{60} are further elaborated upon in section V.

The classical artificial reverberation implementation was proposed by Schroeder in 1961 [5]. These reverberators consisted of a series connection of several all pass filters followed by a parallel bank of feedback comb filters all summed using a mixing matrix [4]. Allpass filters provide *colourless* high-density echoes in the late impulse response. This is used to approximate the perceptual aspects as previously described. Furthermore the parallel comb-filter bank is intended to give a fluctuation in the reverberator frequency response that mimics the perceptual quality of the real room frequency response.

Schroeder's algorithm was improved upon by Moorer in 1979 [7]. His contribution included an implementation of an additional preprocessing block consisting of tapped delay lines to emulate the early reflections. Furthermore, he added lowpass filters to the feedback paths of the feedback comb filters to simulate the high frequency roll off that occurs in natural reverberation [8].

The use of Feedback Delay Networks (FDN) was proposed by Jot in 1992 [9] as a further embellishment on Schroeder's initial design. In the FDN, each delay block is fed back into itself, and then fed into further delay blocks. This topology is conceptually inspired by the movements of reflections in a room. Reflections off walls may proceed to themselves reflect off the ceiling or floors, resulting in further feedback paths. This is simulated by the crossover feedback in the FDN [8]. All of these algorithms will be discussed in detail in section VI.

IV. Data

A real world impulse response from the Echothief Impulse Response data set [10] was used and is shown in figure 2. In order to obtain an idealised physical reverberation response, this was convolved with an anechoic (echoless) voice recording, seen in figure 3. This is what is commonly referred to as convolutional reverb [4]. The algorithmic reverberation methods were implemented on the same anechoic sound.

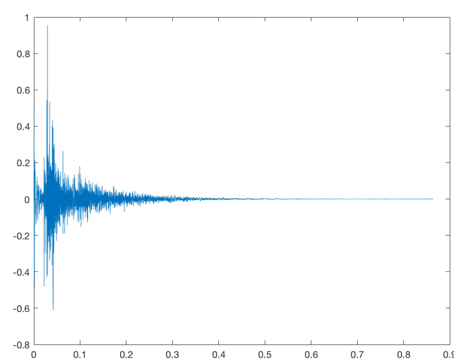

Fig. 2: Impulse Response of Real Room

V. Evaluation Criteria

The performance of the algorithmic reverberation implementations will be compared to the convolutional

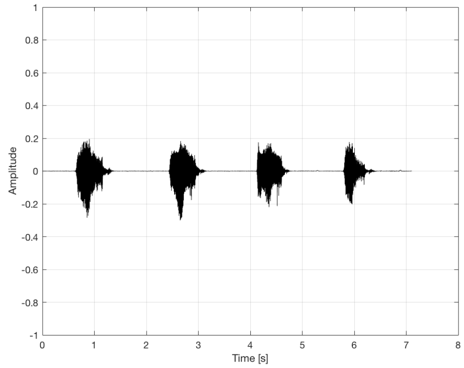

Fig. 3: Anechoic Audio Recording

reverberation implementation. This comparison will hopefully yield results that allow us to modify an algorithm to optimize performance. This evaluation will be performed using the metrics described in section III. Numerous other metrics exist, but for our purposes these are the most relevant, as they relate most directly to the perceptual aspects of reverberation. A detailed explanation of the metrics follows:

- EDC - the energy decay curve is used to find the reverberation time t_{60}. It describes the total signal energy remaining in the reverberator impulse response after time t and is defined as the *tail integral* of the squared impulse response at time t:

$$EDC(t) = \int_t^\infty h^2(\tau)d\tau$$

For the room impulse response, the EDC was obtained by first applying the Hilbert transform to the signal, to get its analytic representation, which aids in 'smoothing' the signal. The Schroeder Integration method shown above was then applied to get its envelope, since it gives a smoother decay curve [11], before it was converted to decibels (see Matlab script in Appendix). This process is illustrated in figure 4.

- $t_{60}(f)$ - The reverberation time is defined as the time taken for the signal to decay to $-60dB$, as this is when sound becomes inaudible to the human ear.
This was found using the EDC, and validated using a statistical approach that applied a curve-fitting function to find the approximate time at $-60dB$ (see Matlab script in Appendix). For our chosen real room impulse response, the reverberation time was found to be $0.6511s$.

- EDR - the energy decay relief defines the EDC for different frequency bands, since different frequencies decay at different rates, where k indicates the frequency bin and m the time-frame.

$$EDR(t_n, f_k) = \int_m = n^M |H^2(m,k)|$$

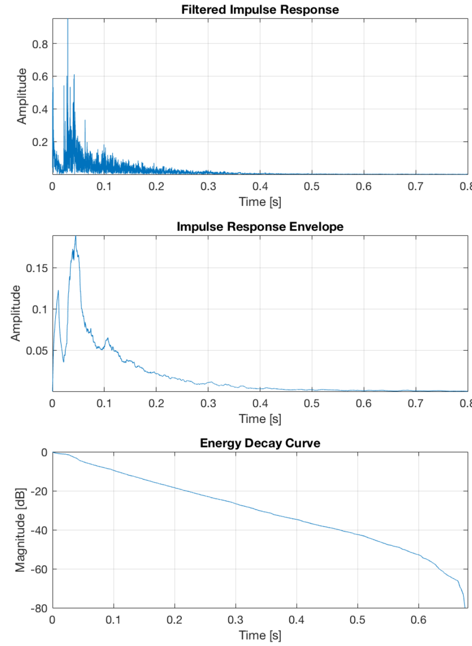

Fig. 4: Energy Decay Curve of Room Impulse Response

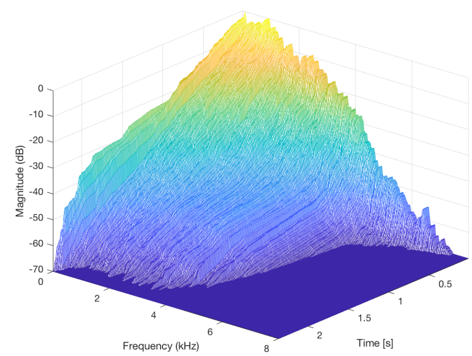

Fig. 5: Energy Decay Relief of Room Impulse Response

We can see in the EDR in figure 5 that the higher frequencies decay much faster than the low frequencies. This is in part due to the fact that high frequencies are more easily absorbed by the surrounding materials in the room.

- *Echo Density Profile* - measures the density of reflections by assuming that after a period of time, the echoes are so dense that the signal "taps" are equivalent to a Gaussian distribution. Following the method proposed by Abel and Huang [12], a sliding window over the signal is used to count the "taps" that are outside one standard deviation of the window. In the beginning, when reflections are sparse, the echo density will be low, and later, when echo density should be high, should tend towards 1. It can be defined by the following formula:

$$\eta = \frac{1}{erfc(\frac{1}{\sqrt{2}})} \sum_{\tau=t-\delta}^{t+\delta} 1\{|h(\tau) > \sigma|\}$$

η represents echo density. $\frac{1}{erfc(\frac{1}{\sqrt{2}})}$ is the expected fraction of "taps" lying outside a standard deviation for a Gaussian distribution, which is used to normalize the equation so that the result sits between 0 and 1. $1\{...\}$ returns 1 if the impulse response 'tap' is outside the standard deviation σ or 0 otherwise. $2\delta+1$ is the window length. The echo density profile of the real room impulse response is shown in figure 6.

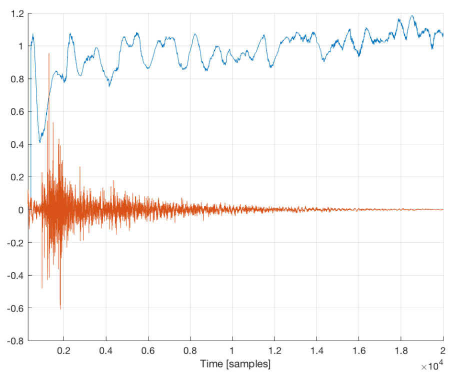

Fig. 6: Impulse Response and Echo Density of Room Impulse Response

A sufficiently smooth exponential decay and frequency response is desired to prevent an unnatural sounding "flutter" or "beating" in the reverberation. This is generally achieved if there is a high enough echo density, defined by Schroeder as 1000 echoes or more per second in the late reverb [4].

VI. Method

As previously mentioned, in order to establish a simplified physical model the anechoic recording was first convolved with the room's impulse response. This proceeded to act as golden measure for comparison with the algorithmic reverberation methods. The three classical reverberation algorithms described in section III were then implemented, and their results of their performance obtained and analysed.

A. Convolutional Reverb

The original anechoic sound signal is shown in figure 7, along with the result of convolved reverberation implementation.

B. The Signal Processing Building Blocks of Algorithmic Reverb

Schroeder was one of the first to provide discrete time processing algorithms for artificial reverberation [2]. Most reverberation algorithms today are composed of networks of

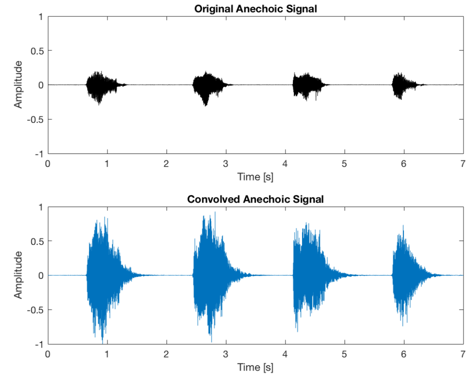

Fig. 7: Convolutional Reverberation Implementation

his proposals for modified comb and allpass filters, as well as additions of tapped delay lines and feedback delay networks. This subsection will detail these building blocks.

1) Tapped Delay Lines: Tapped delay lines (TDL) are an efficient way of extracting multiple echoes from a singular source signal. A TDL is defined as a delay line with a number of "taps", each of which extracts the delayed signal at a different point in the delay line. These taps are then summed to form the output signal of the TDL. An example of this is shown in figure 8.

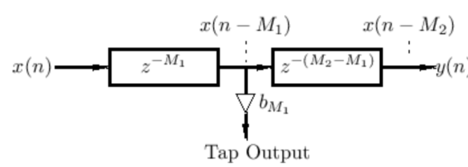

Fig. 8: Tapped Delay Line Block Diagram [1]

2) Comb Filters: Comb filters are a common building block in all audio signal processing. Both feedforward and feedback comb filters exist. Schroeder's implementation made use of Feedback Comb Filters (FBCF), and thus for our purposes it is sufficient to only elaborate upon these. Figure 9 shows a feedback comb filter block diagram.

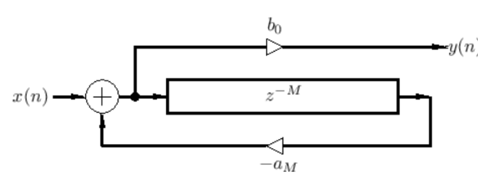

Fig. 9: Feedback Comb Filter Block Diagram [1]

.

A FBCF is a special case of an IIR filter and can be considered as a model of a series of echoes exponentially decaying and uniformly spaced in time [4]. For this reason these filters are very useful in generating the echoes and decay of acoustic reverb.

3) All Pass Filters: Due to the the frequency response of comb filters not being flat, Schroeder proposed combing a feedback comb filter with a feedforward comb filter, giving us the allpass filter (APF). The magnitude of the frequency response of the APF is 1 at each frequency, however, the phase can be arbitrary and thus different delays can be applied at each frequency. A flat frequency response with a number of delays is useful in the pursuit of a “colourless” reverb. A block diagram of the APF is shown in figure 10.

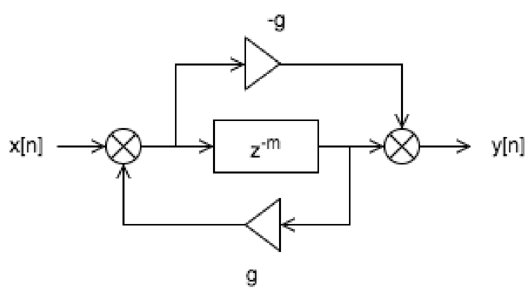

Fig. 10: Allpass Filter Block diagram

The impulse response of an APF is shown in figure 11. It is evident from this that it would be useful in creating multiple delays. APFs are generally used as a building block to construct more complex filtering networks to achieve realistic sounding reverberation. Figure 12 is an example, showing the result of cascading three allpass filters that each have different delay lengths and a gain of 0.7.

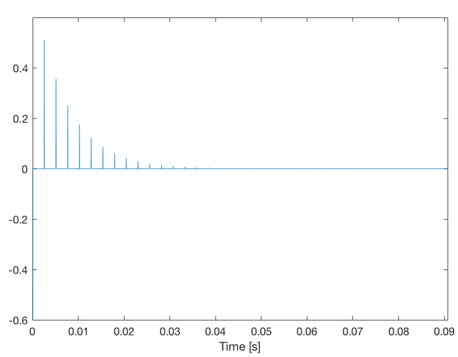

Fig. 11: Allpass Filter Response

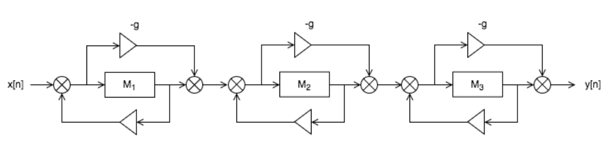

Fig. 12: Series of Cascaded Allpass Filters

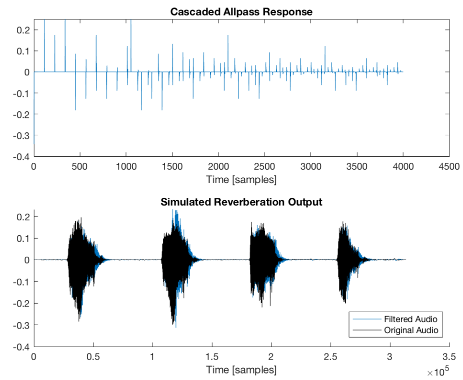

Fig. 13: Cascaded Allpass Filter Response and Reverberation Effect on Anechoic Sound

If we were to apply the cascaded APF with the anechoic sound, it can be seen in figure 13 that the effect on the audio signal is not very useful on its own in implementing a reverb, as it barely differs from its original state. One would expect to see some form of a reverb tail.

4) Feedback Delay Networks: A Feedback Delay Network is a vector FBCF, obtained by replacing the delay line with a diagonal delay matrix [1]. A block diagram of this is shown in figure 14.

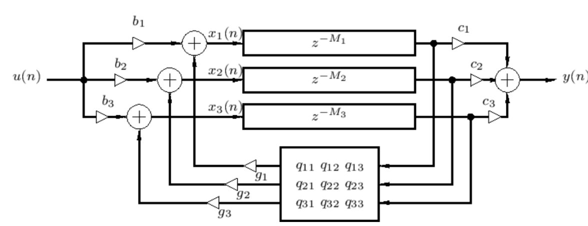

Fig. 14: SISO Feedback Delay Network Block Diagram [4]

The choice of feedback matrix and delay lengths have great effect on the quality of the reverb’s impulse response. As previously discussed, an ideal late reverb impulse response should represent some exponentially decaying noise. The design of the FDN feedback matrices is based on developing a “lossless prototype”, a reverberant impulse response with infinite decay, and ensuring that smooth noise generation is eventually reached [1]. Many Different feedback matrices have been developed. In our implementation we will make use of the “Stautner-Puckette” matrix [8].

C. Reverberation Algorithms

All the following reverberation algorithms make use of some combination of the building blocks discussed in the previous section.

1) Schroeder’s Algorithm: The earliest and most straightforward algorithmic implementation uses a combination of feedback comb filters in parallel followed by two allpass filters in series. The block diagram for this implementation is shown in figure 15. The values for gain and delay lengths were chosen in accordance with Schroeder's recommendations. The APFs have gains of 0.7 and delay line lengths that are mutually prime and approximately equal to $M_i T = \frac{100ms}{3^i}$ for $i = 0, 1, 2, 3, ...$ [4]. This produces a series of impulse responses with decay times that are a third of the previous one. Since the FBCFs are included to provide variation in the frequency response, the delay line lengths can have somewhat arbitrarily chosen lengths, so long as they remain mutually prime [4], and the gains were selected to provide the desired decay time.

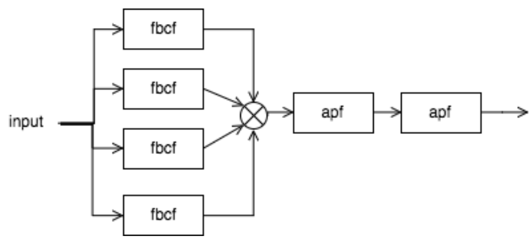

Fig. 15: Schroeder Implementation Block Diagram

The resulting impulse response and EDR graph is seen in figures 16 and 17. Using the statistical method, the t_{60} was found to be $0.6472s$, which is confirmed in the EDC plot in figure 27.

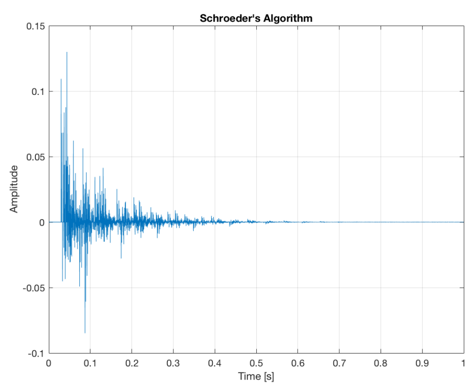

Fig. 16: Schroeder Algorithm Impulse Response

2) Moorer's Algorithm: As described in section III, Moorer's algorithm improves on Schroeder's by adding a tapped delay line in series as well the addition of a lowpass filter in each comb filter's feedback path [2]. The block diagram of the algorithm is shown in figure 18 and an expanded diagram of the series TDL is shown in figure 19. The TDL functions add to the effect of early reflections. This can be clearly seen in the impulse response shown in figure 20 when compared to Schroeder's response in 16. The t_{60} for Moorer's algorithm was found to be $0.67s$, which is slightly higher than Schroeder's.

3) Feedback Delay Networks: Gerzon [13] showed that cross coupling of comb filters could yield improvements in

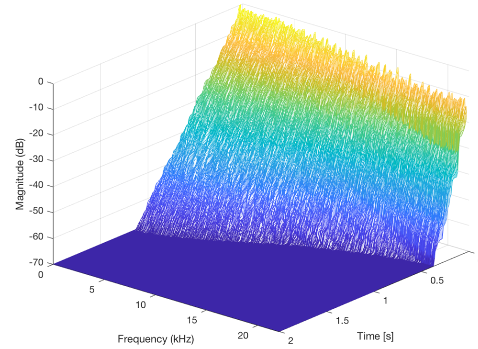

Fig. 17: EDR of Schroeder Implementation

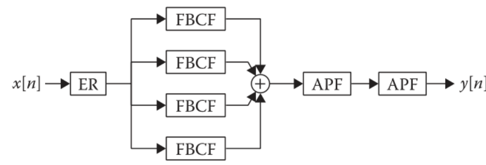

Fig. 18: Moorer's Algorithm Block Diagram

perceptual performance. To achieve this he suggested the use of an "orthogonal matrix feedback reverberation unit" around a parallel bank of delay lines [1]. For the purposes of illustration, if we consider an FDN with two delay lines shown in figure 22, an expanded diagram of its behaviour is shown in figure 23. This gives clear graphical representation of the cross coupling Gerzon described. Due to the complex feedback present in higher order networks, stability needs to be evaluated. Stautner and Puckette [14] proposed a feedback matrix for a fourth order FDN that adheres to a set of stability conditions of their determination. Our implementation of an FDN reverberator makes use of four delay lines and Stautner and Puckette's suggested feedback matrix.

$$A = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 1 & 1 & 0 \\ -1 & 0 & 0 & -1 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \end{pmatrix} \tag{1}$$

The delay line lengths were adapted from Schroeder's original suggested values. That is, chosen to be mutually prime [1]. The resulting impulse response and EDR graph of our FDN implementation is shown in figures 24 and 25. The t_{60} was found to be $0.68s$.

VII. Results & Analysis

Graphs 26, 27 and 28 plot the EDC, echo density profile and reverberant effect on the audio sample for all the implemented algorithms alongside the room's impulse response for easy comparison.

Whilst the reverberation time of Schroeder's implementation is most closely matched to that of the room, the shortfalls of

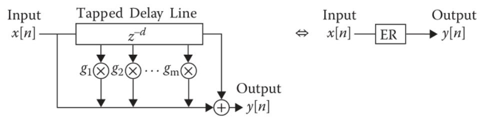

Fig. 19: Moorer's Algorithm Tapped Delay Line

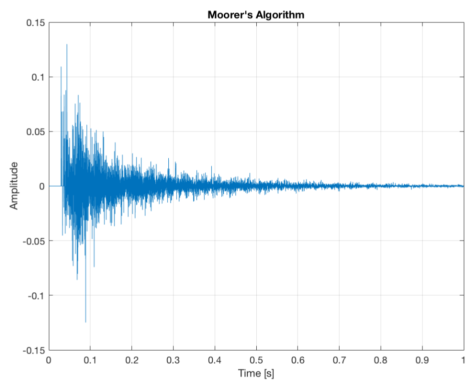

Fig. 20: Moorer's Algorithm Impulse Response

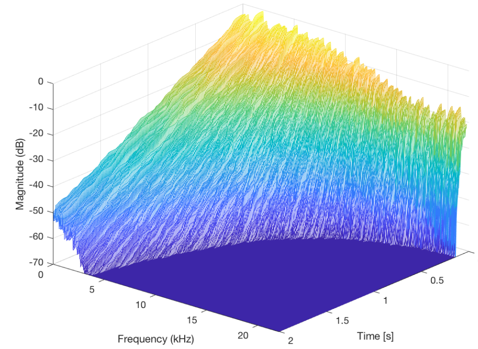

Fig. 21: EDR of Moorer Implementation

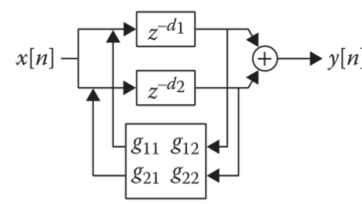

Fig. 22: FDN with two delay lines [8]

Schroeder's algorithm lie in its low echo density, seen in figure 28. It is far too sparse, with an echo profile that never reaches one and does not increase much with time. As can be seen in figure 17, the lack of contrast in reverberation time between the high and low frequencies is also problematic as this as this has been shown to result in a "metallic" sounding reverb [8]. To emulate the perceptual quality of the room in question, an EDR closer to the one of the real room impulse response as seen in figure 5 will be required.

The density of the early reflections is clearly greater in Moorer's algorithm than in Schroeder's. There is further evidence of this in figure 28, where the echo density of Moorer's algorithm is clearly higher during the period of early reflections. This is a step closer to approximating the echo density plot of the real room impulse response as seen in figure 6.

The EDR seen in figure 21 shows an improvement in the spread of reverberation times for high and low frequencies when compared to that of Schroeder's implementation. This can be interpreted as the successful use of low pass filters in the feedback paths to simulate the high frequency roll off of natural acoustic environments. The reverberation time of lower frequencies has improved, whilst higher frequencies decay faster, which is a more realistic reverberation response and removes the unnatural metallic sounds of Schroeder's algorithm.

The FDN network implementation with the Stautner-Puckette feedback matrix has the longest reverberation time, which can clearly be seen in 26 and 27. In the context of replicating the real room, this is highly undesirable. However, its echo density profile displays the most desirable characteristics, with a smoother increase that reaches 1 at the end of the response. The FDN Also displays a good EDR as shown in figure 25. Low frequencies have a much longer decay time than high frequency components.

All of the algorithms display the intended noise-like response at the end of their echo density profiles, as can be seen in figure 28. The FDN Network has the most perceptually desirable characteristics as they have been developed throughout this paper. However, our implementation of the network falls short of the desired real room impulse response decay time. In the following section, steps will be taken to attempt to improve the performance of the FDN implementation in relation to our chosen real room impulse response.

VIII. Development

The FDN algorithm can be modified in a number of ways, however, the element with the most variation in the literature seems to be the values in the feedback matrix [1]. Changing the matrix mostly has an effect on the characteristics of late reverberation. Thus with our goal in mind of decreasing the decay time of the current FDN matrix to more closely reflect that of the real room impulse response, this seems to be the most appropriate course of action. Depending on the choice of matrix, delay times and frequency spread (as per EDR) can be changed. Furthermore, the stability of the system can change dramatically and in the worst case begin to oscillate. Besides the Stautner-Puckette matrix described in section VI-C3, a number of feedback matrix choices have been defined in the literature [8]. For this investigation we will consider the

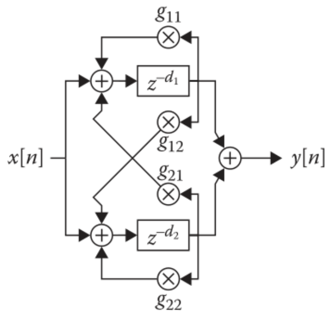

Fig. 23: Expanded Two Delay Line FDN [8]

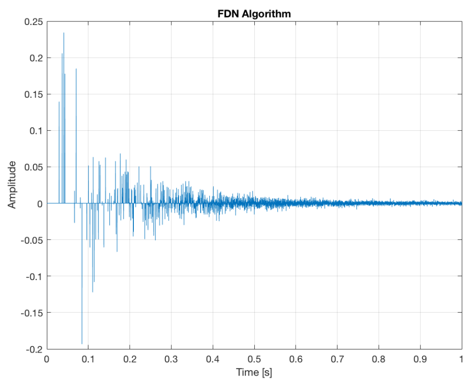

Fig. 24: FDN Impulse Response

performance of the Householder matrix and the Hadamard matrix [1], followed by some of our own experimentation. The Hadamard matrix is defined as

$$A = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \tag{2}$$

And the Householder is defined as

$$A = \frac{1}{2}\begin{pmatrix} 1 & -1 & -1 & -1 \\ -1 & 1 & -1 & -1 \\ -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & 1 \end{pmatrix} \tag{3}$$

Our experimentation involved changing the signs of various elements of the matrix as well as experimenting with their values. This was mostly a process of trial and error or tuning. During this processes we discovered that the choice of values in this matrix can very easily cause extreme oscillations and distortion, rendering the anechoic signal unidentifiable.

Figure 30 shows the results of measuring the reverberation times of the various feedback matrix implementations. There is very little reduction in decay time displayed by our custom matrix, whereas both the Householder and the Hadamard matrix display a marked improvement. There is further evidence of this on inspection of the EDR plots in figures 31 and 25. Thus both the Householder and Hadamard have been successful in achieving a closer approximation to the real room impulse response in question, however, it is evident in figure 30 that the Householder matrix achieves the best approximation, with a very close correlation in the later decay time. On the other hand, our custom implementation is shown to be ineffective in achieving our goal. As can be seen in figure 32, there is a good decay from low to high frequencies, but the decay time in the low frequencies is far too long. Further evidence of this can be seen in figure 34, where it is evident in the custom algorithm that oscillations are present in the late reverb.

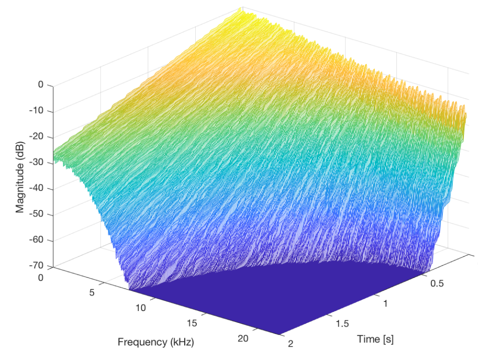

Fig. 25: EDR of FDN Implementation

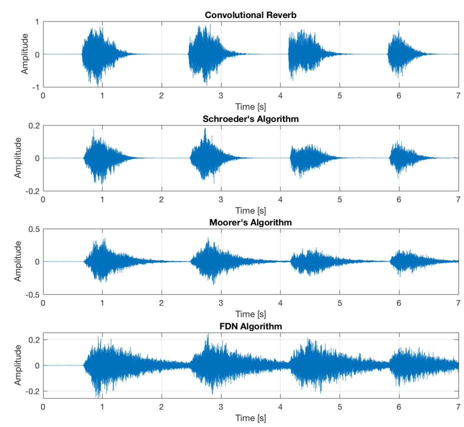

Fig. 26: Simulated Reverberation on Anechoic Sound

In figure 29 it is clear that the behaviour of the implementations all display similar echo density characteristics. Furthermore, it can be seen in figure 33 that the early reflections components are barely changed between the different implementations. This result is inline with the hypothesis that modifications to the FDN feedback matrix will only have a marked effect on the late reverberation components of the impulse response.

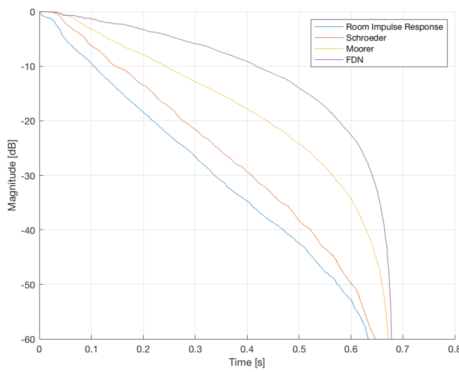

Fig. 27: EDC for All Reverberation Algorithms

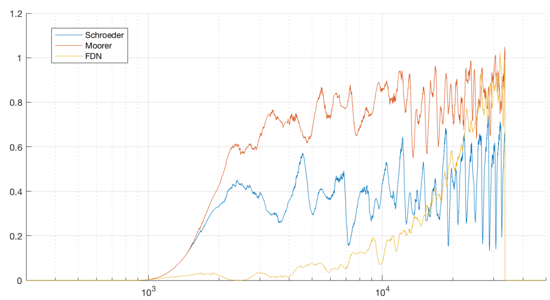

Fig. 28: Echo Density for All Reverberation Algorithms

Due to the fact that the Householder matrix displays the best approximation to the real room impulse response decay time as well as having a pleasing echo density profile, it is our clear choice for feedback matrix implementation in this case. However, as can be seen when comparing the EDR of the Householder (figure 31) with the EDR of the real room impulse response (figure 5), the decay of the low frequency components in the Householder implementation is insufficiently short. This could be an area of improvement in later implementations. Further low pass filtering in comb filter feedback paths could yield more please results.

In order to test the hypothesis that the Householder Feedback matrix has the best approximation to the real room impulse response, the anechoic signal was processed by both the convolutional implementation (seen in figure 7) and all of the FDN implementations. Correlation was performed between the output of convolutional reverb and each respective FDN implementation. The results are shown in table I. This confirms our findings that the Householder feedback matrix is the best implementation for our purposes.

IX. Conclusion

Emulating a physical reverberation model using an algorithmic approach presents an interesting signal processing challenge. The concept of perceptual modelling has allowed us to greatly reduce the complexity of this task. Our findings have shown that in order to emulate the reverberant effects of our chosen room, the use of a Feedback Delay Network with the Householder feedback matrix has been shown to be the most successful. However, this does not hold true to all room impulse responses. Different rooms will display differing perceptual metrics, and different algorithmic models may prove to be more successful. Although the Householder is the best implementation we have found, there is much room for improvement in certain aspects of its performance. The main shortcoming can be seen in the Householder EDR curve, and steps can be taken in later iterations to improve this. This could include the possibility of using higher order FDNs or adding more filtering in the comb-filter feedback paths. Perhaps a future study could use machine learning techniques based on the echo-thief dataset [10] to determine optimum FDN feedback matrix coefficients based on the perceptual metrics of each room. On a purely qualitative basis, the results of these algorithms sound quite convincing. Wav files containing the results as well as all code used in this paper are available in our github repository [15].

Algorithm	Correlation Coefficient
Stautner & Puckette	0.0460
Householder	0.0600
Hadamard	0.0299
Custom	0.0453

TABLE I: Correlation Between Reverberated Audio using Convolution and FDN Algorithms

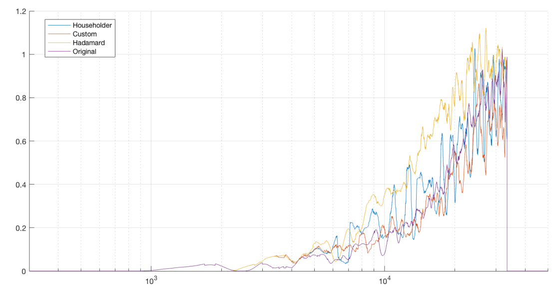

Fig. 29: Echo Density of FDN Algorithms

References

[1] J. O. Smith, *Physical Audio Signal Processing*. `http://ccrma.stanford.edu/~jos/pasp/`, accessed 20/04/19, online book, 2010 edition.

[2] W. Gardner, *Reverberation Algorithms*. MIT Media Laboratory, 2002.

[3] I. Szoke, M. Skacel, L. Mosner, J. Paliesek, and J. H. Cernocky, "Building and Evaluation of a Real Room Impulse Response Dataset," *arXiv:1811.06795 [eess]*, Nov. 2018, arXiv: 1811.06795. [Online]. Available: http://arxiv.org/abs/1811.06795

[4] J. O. Smith, "Artificial reverberation and spatialization," MUS420/EE367A Lecture 3, Center for Computer Research in Music and Acoustics, Department of Music, Stanford University, 2018.

[5] M. R. Schroeder, "Natural Sounding Artificial Reverberation," *Journal of the Audio Engineering Society*, vol. 10, no. 3, Jul. 1962.

[6] A. D. Pierce, *Acoustics*. American Institute of Physics, for the Acoustical Society of America, 1989.

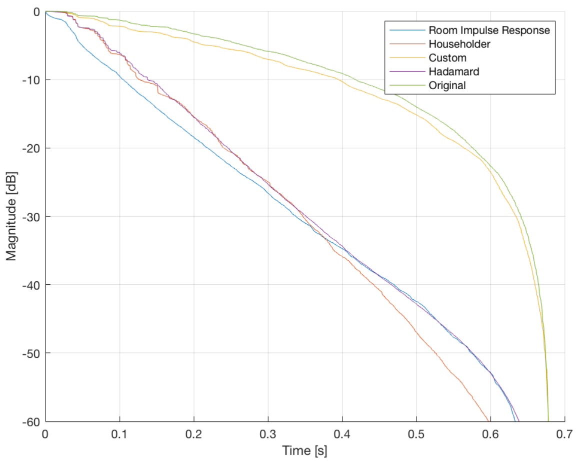

Fig. 30: EDC of FDN Algorithms

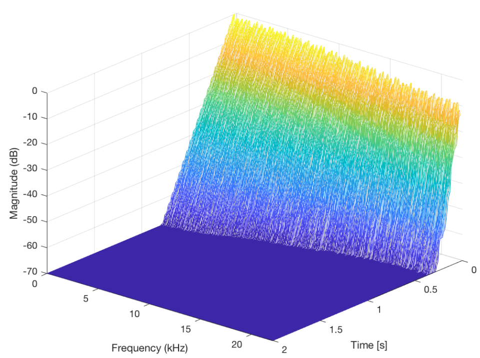

Fig. 31: EDR of Householder FDN Algorithm

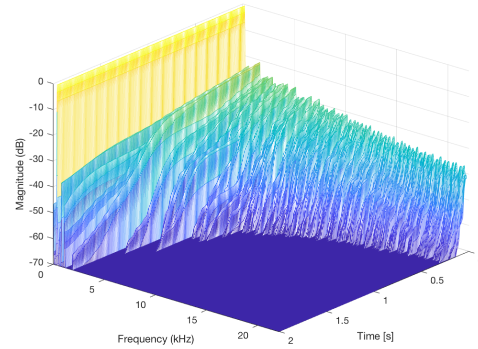

Fig. 32: EDR of Custom FDN Algorithm

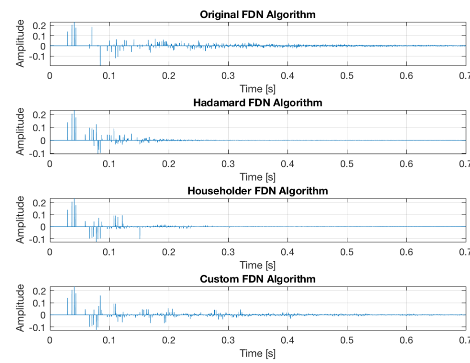

Fig. 33: Impulse Responses of FDN Algorithms

[7] J. A. Moorer, "About this reverberation business," *Computer Music Journal*, vol. 3, no. 2, pp. 13–28, 1979. [Online]. Available: http://www.jstor.org/stable/3680280

[8] E. Tarr, *Hack Audio: An Introduction to Computer Programming and Digital Signal Processing in MATLAB*, 1st ed. New York, NY : Routledge, 2019. | Series: Audio Engineering Society presents: Routledge, 2018.

[9] J.-M. Jot, "An analysis/synthesis approach to real-time artificial reverberation," *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, vol. 2, pp. 221–224, 01 1992.

[10] "Echo thief," http://www.echothief.com.

[11] M. R. Schroeder, "New Method of Measuring Reverberation Time," 1964.

[12] P. H. J. Abel, "A simple, robust measure of reverberation echo density," Convention Paper, Audio Engineering Society, 2006.

[13] M. A. Gerzon, "Synthetic stereo reverberation," *Studio Sound*, vol. 13, 1971.

[14] J. Stautner and M. Puckette, "Designing multi-channel reverberators," *Computer Music Journal*, vol. 6, p. 52, 09 1983.

[15] "Eee4114f project github repository." [Online]. Available: https://github.com/murning/DSP-Project-Reverb

APPENDIX

```
%
% function to get the EDC
% and the RT
%
% EDC: energy decay curve
% RT: reverberation time
%

function [RT, EDC] = edc(ir)

fs = 1/44100;

% smoothing with hilbert transform
irHilbert = abs(hilbert(ir));

% schroeder integral
td = 30e3; % value chosen to match envelope
EDC(td:-1:1)=(cumsum(irHilbert(td:-1:1))
/sum(irHilbert(1:td)));
EDC = 10*log(EDC);

% getting RT
dt=1/fs;
t = 0:dt:(length(EDC)*dt)-dt;
slope = polyfit(t,EDC,1);
slope = slope(1);
RT = -60/slope(1);

end
```

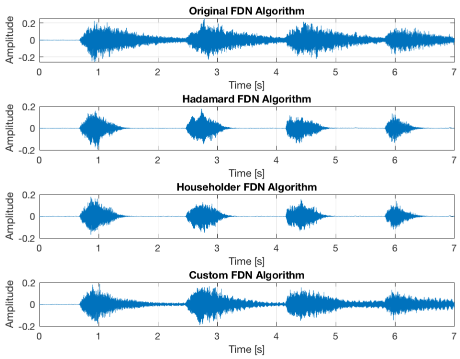

Fig. 34: Simulated Reverberation Responses of FDN Algorithms on Anechoic Sound

```
%
% function to implement three cascaded
% schroeder allpass filters
%
%

function [y] = allpass(signal)

% recommended gain
g = 0.7;

% recommended delay line lengths
M = [113 337 1051];

b = [-g zeros(1,M(1)-1) 1];
a = [1 zeros(1,M(1)-1) -g];

x = [1 zeros(1, 4000)]; % to apply filter to impulse
% x2 = signal; % to apply filter to signal

y = x;
for n = 1:length(M),
  b = [-g zeros(1,M(n)-1) 1];
  a = [1 zeros(1,M(n)-1) -g];
  y = filter(b, a, y);
end

end
```