McGill University

Schulich School of Music

Music Technology

MUMT 618

Project Report

Course Instructor: Dr. Gary P. Scavone

Student: Shayan Mozaffari

Table of Contents

Introduction

Nearly 60 years ago, Schroeder proposed the idea of using digital signal processing techniques to implement artificial reverberation. Since then, a lot of research has been done and a large amount of literature has been published. [1]

The study of the room acoustics, shows that late reverberation part of the room impulse response, resembles noise. In the late 90's, Rubak and Johansen used a randomly weighted impulse train with low impulse density as the impulse response of an artificial reverberator. They found out, that impulse densities between 2000 and 4000 spikes per second could result in perceptually rich and smooth reverberation. [2] [3]

In 2007, Karjalainen and Jarvelainen came up with the idea of using randomly signed jittered impulse train to implement artificial reverberation. As this noise is perceptually smoother than Gaussian noise, it is named velvet noise. They developed some perceptual studies about this noise and also proposed a practical algorithm to implement artificial reverberation using velvet noise. [4]

In this project, the aim is to study and implement some reverberation algorithms which utilize velvet noise. For this purpose, room acoustics and artificial reverberation are briefly reviewed. Then, the switched convolution reverberator and its implementation are discussed. It is followed by the explanation about theory and implementation of basic and advanced FVN algorithm which are both proposed by the same authors. At the end, velvet noise feedback delay network is briefly introduced.

All implementations of this project are in MATLAB. Not every details of algorithms were proposed in the papers. But the aim was to get a perceptually acceptable and consistent results. Perceptual conclusions are based on listening to output signals using iCon SX-4A studio monitors. Therefore, there might be some inaccuracies involved below 65 Hz.

Room Acoustics and Artificial Reverberation

Reverberation is consisted of multiple reflections of sound waves from surfaces and objects of a room. From signal processing point of view, room impulse response is consisted of direct sound, early reflections and late reverberation, as shown in figure 1.

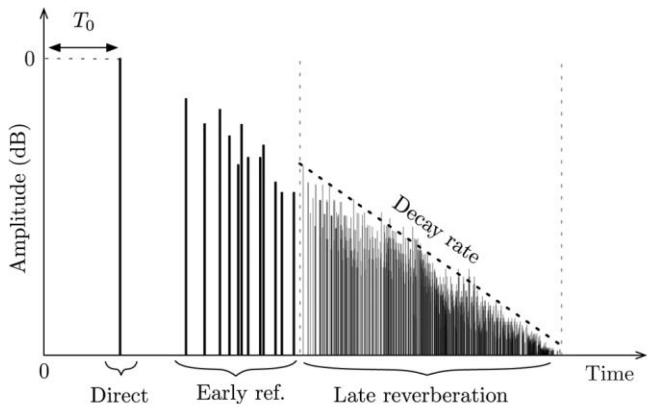

Figure 1: Example of generic room impulse response [1]

As it is apparent in this figure, Leith reverberation part resembles an exponentially decaying noise. As it is emphasized in figure 1, the amplitude of room impulse response in dB, decays linearly. Thus, absolute amplitude decays exponentially. It can also be shown, that absolute amplitude of frequency response of a room decays exponentially as frequency increases. Spectrogram of impulse response of the Memorial Church at Stanford University is shown in figure 2.

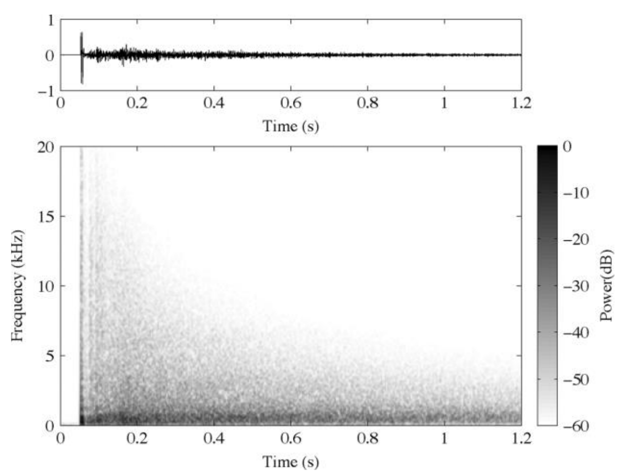

Figure 2: Spectrogram of impulse response of the Memorial Church at Stanford University [1]

The mentioned exponential decay in both domains, could be seen in this spectrogram. More precise analysis of room impulse response show that despite the decay of absolute amplitude, number of reflections grow as time increases. Number of reflections for late reverberation part is given by:

$$N_{refl}(t) = \frac{4\pi c^3 t^2}{V}$$

Where c is the speed of sound, t is time and V is the volume of the room. This quantity shows the number of reflections per second.

It is also shown that room impulse response is consisted of eigenmodes. There are a few high amplitude eigenmodes in low frequencies and as the absolute amplitude of these eigenmodes decreases, the number of them increases, as given by:

$$N_{mode}(f) = \frac{4\pi V f^2}{c^3}$$

Where f is frequency in Hz. This quantity shows the number of eigenmodes per Hz. [5] [4]

As observations and analysis show, it is logical to use shaped noise to implement late reverberation part of a room impulse response. Perceptual studies show that each reflection could not be heard. For example, Schreiber understood that for a bandwidth less than 2 kHz, impulse density of the noise should be equal to the bandwidth for a smooth sounding reverberation. But interesting results of his research is that for higher bandwidth, impulse density of 2 kHz results in a reverberation, with the same perceptual qualities. [6] [4]

Rubak and Johansen's research led to interesting results in artificial reverberation using digital signal processing techniques, as convolution with low impulse density signals is more computationally efficient compared to previous techniques. A few years later, Karjalainen and Jarvelainen introduced a sparse noise using the previous idea, and as it sounded smoother than other kinds of noise, they named it velvet noise. [2] [3] [4]

Velvet noise is only consisted of {0,1,-1}. Its formal definition is:

$$n[k] = \sum_{m=0}^{M} a[m]\delta[k - round\left(\frac{T_d}{T_s}(m + rnd(m))\right)]$$

$$where\ a[m] = \begin{cases} -1, & probability = 0.5 \\ 1, & probability = 0.5 \end{cases},\ rnd(m) \sim U[0,1]$$

$$, F_d = \frac{1}{T_d}\ is\ impulse\ density\ and\ F_s = \frac{1}{T_s}\ is\ sampling\ rate$$

It can also be seen as jittered impulse train. This jittering prevents periodicity. In fact, it is shown that the spectrum of this noise approaches a white spectrum as the length of the sequence increases. Perceptual studies of this noise show that this noise is perceived softer than Gaussian noise for impulse densities higher than 1500 impulses per second. The reason is found out to be that the low-pass filtered amplitude envelope of this noise is softer than Gaussian noise. [4]

A few approaches have been proposed for generating a sequence of this noise [4] [7] [8] [9]. In this project the classic approach proposed in [4] is used in `velvet(s,fd,fs)`. In this function, s is number of seconds, fd is the impulse density and fs is sampling frequency. An example of a generated velvet noise sequence and its spectrum using this function is shown in figure 3 and figure 4. The latter figure shows that the spectrum is fluctuating around a constant value (about 40 dBs).

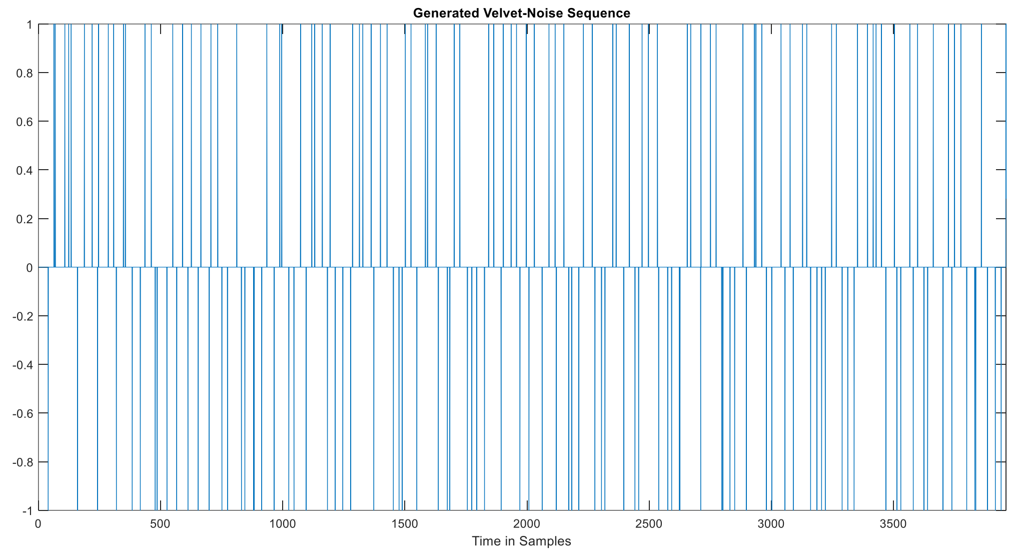

Figure 3: Part of a 10-second velvet noise sequence with impulse density of 2000 spikes/sec

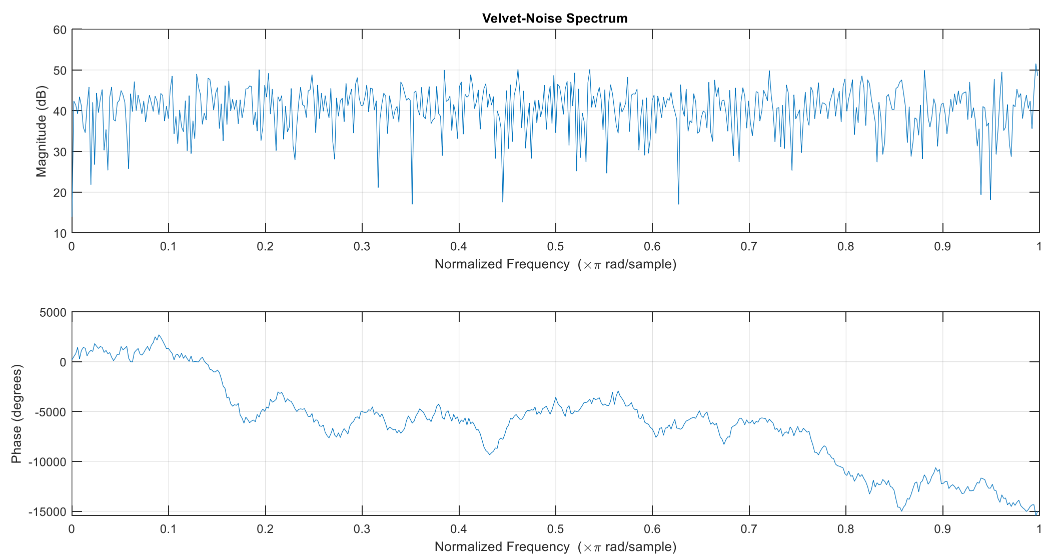

Figure 4: Spectrum of velvet-noise sequence presented in figure 3

Sound example of this velvet-noise sequence is also provided. Note that because of random jittering, length of the sequence might be slightly more than entered length which is negligible for long sequences. All velvet-noise sequences in this project are generated using this function.

The Switched Convolution Reverberator [4] [7]

Karjalainen and Jarvelainen utilized the structure shown in figure 5 to implement artificial reverberation using velvet noise.

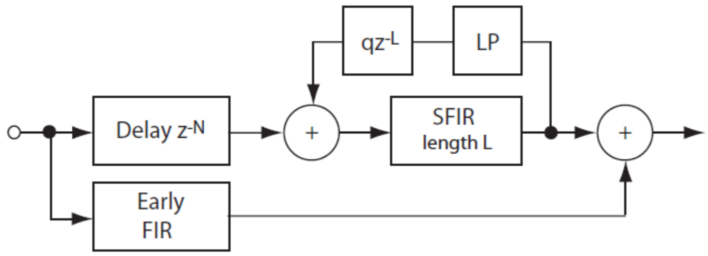

Figure 5: Basic structure proposed in [2] [3] and modified in [4]

It is also possible not to use a delay loop for artificial reverberation with sparse FIR velvet noise. But despite the fact that most of the samples of velvet noise sequence are zero and only 3-5% taps of a full-tap delay-line is required, convolution is still very computationally demanding and therefore a feedback loop is utilized.

The parameter q is tuned so that preferred T_{60} is obtained:

$$q = 10^{-\frac{3T_{Loop}}{T_{60}}}$$

Using this value, the following envelope function results:

$$e(t) = 10^{-\frac{3t}{T_{60}}}$$

Notice that in real world's environment, T_{60} is frequency dependent. Therefore, q will be a filter (q(z)).

Spectral coloration is the nature of a feedback loop. It was understood that short loops can cause spectral coloration and long loops introduce perceivable periodicities. [4] Thus, using velvet noise tap tables was the idea that Karjalainen and Jarvelainen came up with which is shown in figure 6.

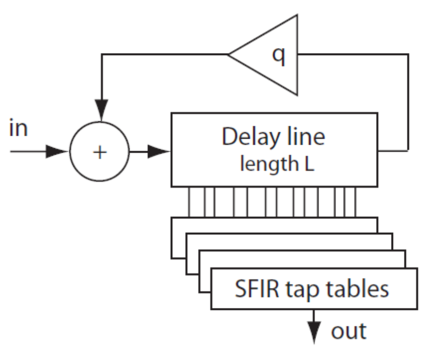

Figure 6: Time-varying structure proposed in [4]

Sharp changes between consecutive taps might cause some perceivable discontinuities. To overcome this difficulty, they have proposed square-root law interpolation. As we are working with noise sequences, interpolation should be done linearly in power, as shown in figure 7.

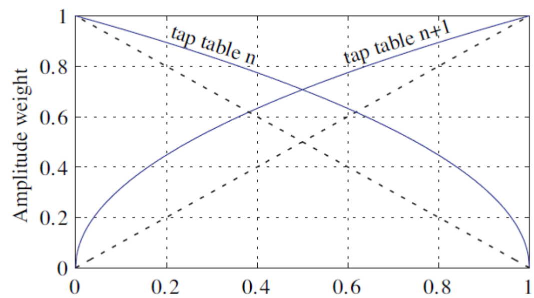

Figure 7: Square-root law interpolation [4]

A modified version of the Switched Convolution Reverberator was proposed in [7]. The idea is to use a steady sequence of velvet noise with some time-varying pulses in order to have a time-varying filter. A time-varying velvet-noise sequence during one crossover period (T_c) is shown in figure 8.

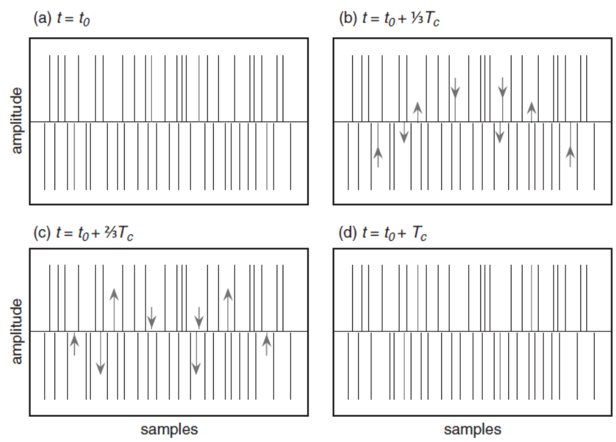

Figure 8: Time-varying velvet noise sequence over one crossover period [7]

The block diagram of their proposed reverberator is shown in figure 9.

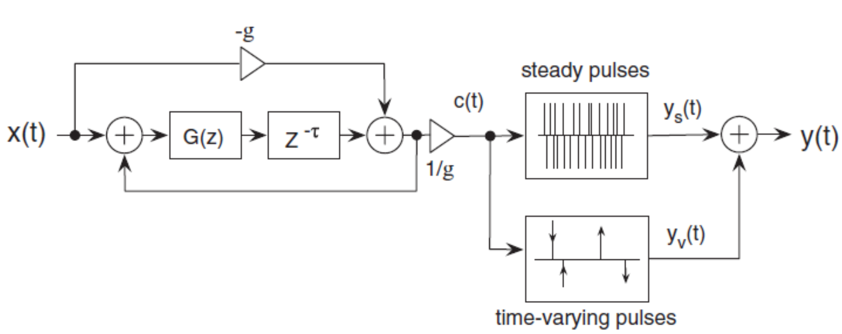

Figure 9: Switched Convolution Reverberator proposed in [7]

As it is apparent, a different feedback loop is used in this model. Moreover, the noise sequence is taken out of the loop. This looks more logical, as a long sequence of time-varying noise in a feedback loop can place the poles anywhere in the z-plane and cause unstability.

The time-varying part of the noise sequence is given by:

$$y_v(t) = \sum_{k=0}^{N-1} \{w_c(t) S_k c(t - p_k) + [1 - w_c(t)] R_k c(t - q_k)\}, 0 \le t \le T_c$$

Where $w_c(t) = \frac{t}{T_c}$ is the windowing function, $c(t)$ is the comb filter output, p_k and q_k are the randomly selected taps and S_k and R_k are random signs. These equations show that linear interpolation is used in this scheme, which does not look suitable for interpolation of random sequences.

In the implementation of this algorithm for this project, the aim was to merge suitable parts of mentioned structures. Thus, time-varying noise sequence was taken out of the loop to prevent unstability. Also, the square-root law interpolation was utilized for noise sequences. The loop structure proposed in [4] was used, but q was replaced by q(z) to realize frequency dependent T_{60}. Moreover, three Schroeder all pass filters have been used in order to reduce the peak power of impulses and also make a longer output sequence. All pass filters have coefficient of $\frac{\sqrt{5}-1}{2}$ which is reciprocal of golden ratio and the order of them are 1, 63 and 139 [8]. The block diagram of the implemented system is shown in figure 10.

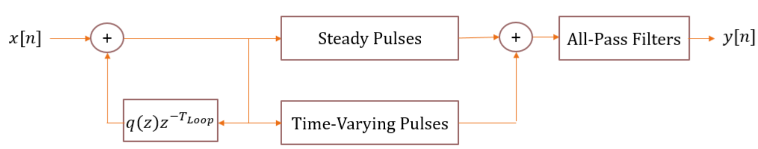

Figure 10: Block diagram of implemented switched convolution reverberator

q(z) filter is of order 20 (by default) and is designed using fir2 function using the input T_{60} values. The default filter and loop frequency response are shown in figure 10 and 11.

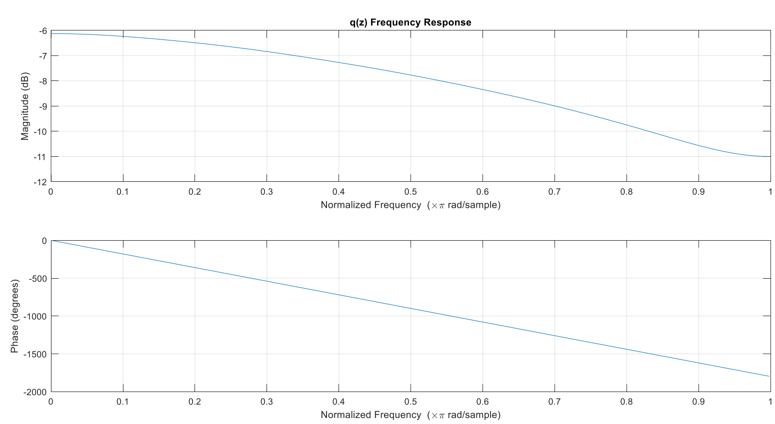

Figure 11: q(z) frequency response

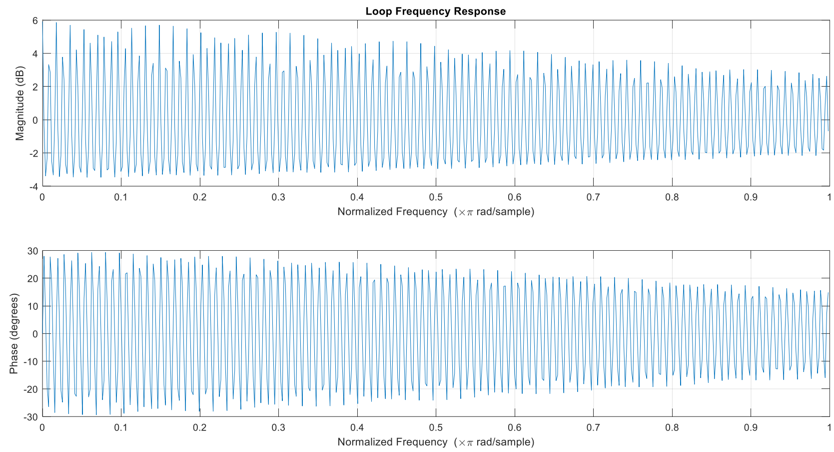

Figure 12: Loop frequency response

As the latter figure shows, the loop causes unwanted spectral coloration. But as there are a large number of spectral peaks, and the spectral envelope is a low pass filter, most likely based spectral coloration will not be perceived by our auditory system.

Modification of the noise sequence and convolution with the output of the loop, is done in a for loop. Two noise sequences are generated and changed every $N_c = f_s T_c$ samples; one of them at k=1 and one of them at k=$N_c/2$, where k is a counter counting from 1 to N_c. $h_1[n]$ contains steady pulses and $h_2[n,k]$ and $h_3[n,k]$ are time-varying velvet noise. h[n] is the final sequence which changes at each time step. The following equations describe this for loop:

$$\beta = \sqrt{\frac{|2k - N_c|}{N_c}}\ , \alpha = \sqrt{1-\beta^2}$$
$$h[n,k] = h_1[n] + \alpha h_2[n,k] + \beta h_3[n,k]$$
$$y[n] = h[n,k] * x[n]$$

But the output of this time-varying noise sequence is fed into mentioned all pass filters. At the end, there is a mixer to tune the amplitudes of direct sound, early reflections and late reverberation. Note that the early reflections are generated using first 100 milliseconds of a real room impulse response available at [10].

Basic FVN Algorithm [8]

FVN stands for filtered velvet noise. The theory behind this algorithm is quite simple. Unlike the previous algorithm, no feedback loop is utilized in this algorithm except for all pass filters. Block diagram of this algorithm is depicted in figure 13.

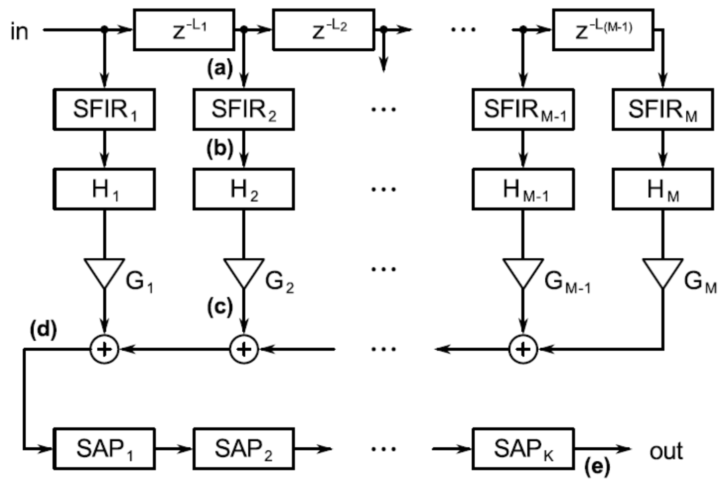

Figure 13: FVN algorithm block diagram [8]

In this algorithm, ideal impulse response is divided into segments and each segment has unique all pass filter and gain. As in reverberation, high frequency components fade out quicker than low frequency components, cut of frequency of filters is higher in first segments. Moreover, as mentioned before, impulse density of velvet noise sequences can decrease as the cutoff frequency decreases. This makes this algorithm computationally efficient. Furthermore, utilizing all pass filters allows us to use lower densities. An example of segmentation and required impulse density is shown in figure 14.

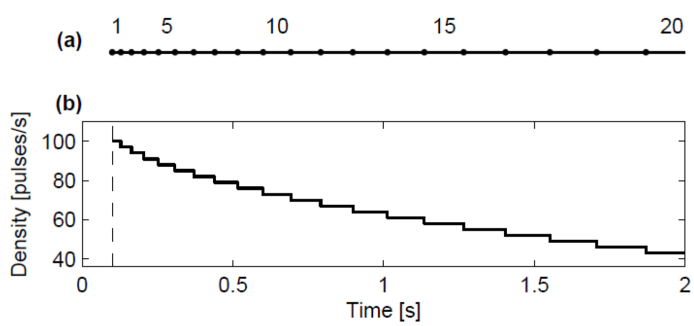

Figure 14: Segmentation and required impulse density [8]

This figure shows that segments become longer as time increases. Moreover, that decrease of impulse density is apparent.

To make the analyses more intuitive, an example of impulse response of different points of the system depicted in figure 13 is shown in figure 15.

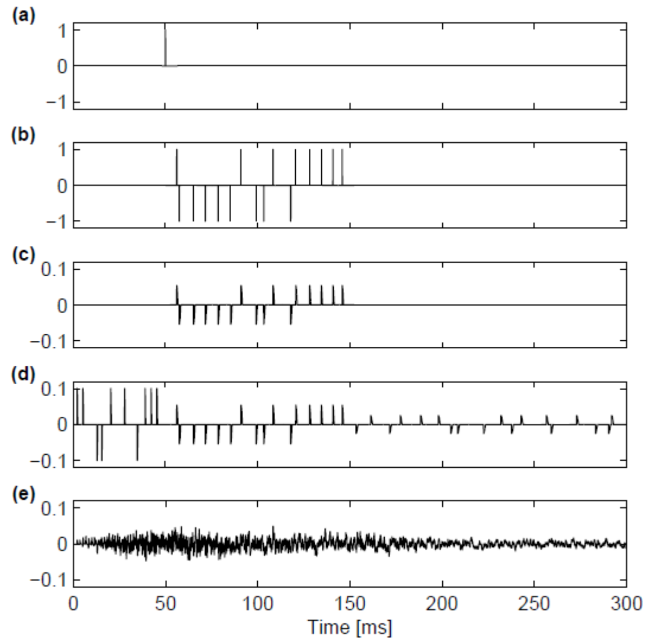

Figure 15: Impulse response of the system depicted in figure 13 [8]

This figure shows how this system is working. Low-pass filters utilized in this model are 20th order LPC filters. The magnitude response of these filters is shown in figure 15.

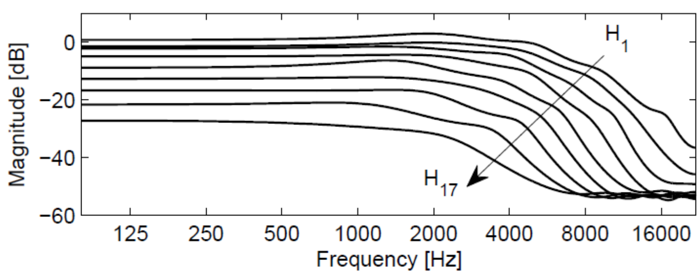

Figure 16: Magnitude response of low-pass filters

7 all pass filters have been utilized in this model. These filters are of orders 1, 64, 140, 209, 442, 555 and 630. Coefficient of all of these filters is $\frac{\sqrt{5}-1}{2}$ which is the reciprocal of the golden ratio and is utilized in order to take the best performance out of these filters.

The final sound, is taken from the mixer as shown in figure 17.

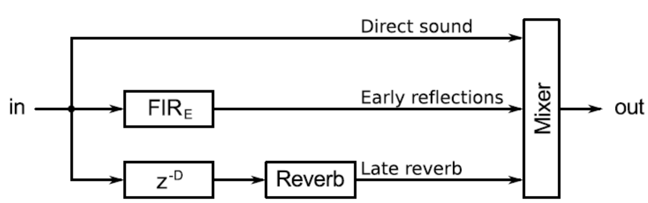

Figure 17: Mixer [8]

As there was no specific concert hall to simulate its impulse response using this model, I came up with using exponential changes in variables associated with this model. The first variable is length of segments. For a geometric sequence we know that:

$$b = a\frac{q^N - 1}{q - 1}$$

Where a is the first number of sequence, q is the ratio between two consecutive numbers of sequence and b is the sum of N consecutive numbers of the sequence. If we take b as the reverberation time the need and q as the ratio of two consecutive segments, we can find the length of the first segment as given below:

$$a = b\frac{q - 1}{q^N - 1}$$

Using this equation and geometric sequence idea, reverberation time of approximately b seconds is obtained. Furthermore, if they want to have approximately 60 dBs of decay in the last segment, we should use the following recursive equation for gains:

$$G_1 = 1\,, G_{i+1} = dG_i \ where\ d = 10^{\frac{-3}{N}}$$

Low-pass filters are designed using fir2 function. Although using LPC filters might result in a longer reverberation, but unfortunately, no more details is provided in the paper and therefore, simple FIR filters have been utilized. The **g** vector contains the gain values for frequencies saved in **f** vector. Default expressions of these vectors is:

$$\boldsymbol{f} = \{\mathbf{0}, \mathbf{1000}, \mathbf{3000}, \mathbf{5000}, \mathbf{8000}, \mathbf{15000}, \frac{\boldsymbol{f_s}}{\mathbf{2}}\}$$
$$\boldsymbol{g} = \{\mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{0.95}, \mathbf{0.9}, \mathbf{0.85}, \mathbf{0.8}\}$$

Each branch of the given block diagram in figure 13 is one iteration of a four loop. To modify the values of **g**, last four elements of this vector are updated in each iteration:

$$g_4^{i+1} = cg_4^i$$
$$g_5^{i+1} = c^2 g_5^i$$
$$g_6^{i+1} = c^3 g_6^i$$
$$g_7^{i+1} = c^4 g_7^i$$
$$where\ c = (\frac{1}{\sqrt{2}})^{\frac{1}{N}}$$

Where i shows the ith iteration. Using these values, the frequency of 5 kHz is approximately the 3dB cutoff frequency in the last branch (iteration). Moreover, the exponential decay of frequency components in short term impulse response of the system is preserved. Magnitude response of all of the all pass filters is shown in figure 18. This shows that filters designed by fir2 with the given input parameters approximately have exponential decay as frequency increases (linear in dB). Also this slope of these lines (in dB) is approximately multiplied by a constant in each iteration.

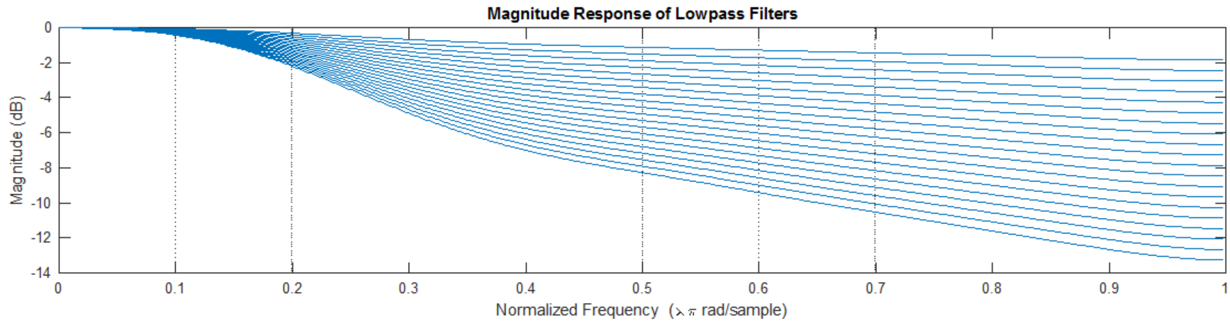

Figure 18: Magnitude response of all lowpass filters

After the resulting signal is passed through all pass filters, the mixer is placed to help us get the preferred sound.

Note that, like the previous algorithm, early reflections are implemented using the real room impulse response [10].

Advanced FVN Algorithm [9]

This algorithm was proposed by the same authors of previous algorithm. As expected, it has a lot of similarities with the basic FVN algorithm. The general block diagram of a reverberator using this algorithm is given that in figure in 19.

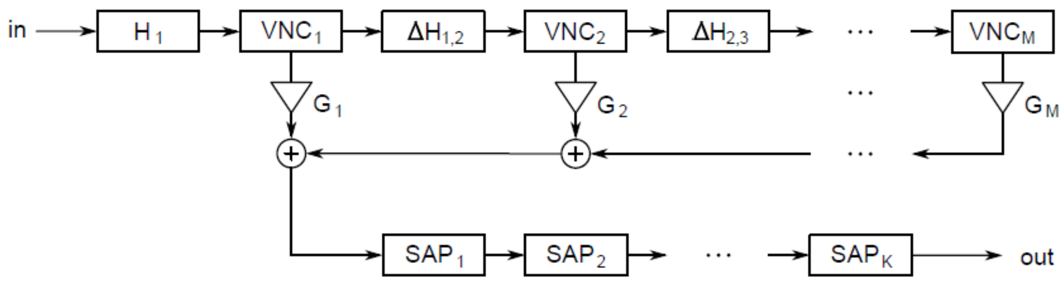

Figure 19: Block diagram of advanced FVN algorithm

There are two main differences between basic and advanced FVN algorithm. The first difference is that lowpass filters are replaced with a single lowpass filter and some differential coloration filters $\Delta H(z)$, in order to reduce the computational demands. Differential coloration filters are notch filters that are described the following equations:

$$\Delta H(z) = 1 + (V_0 - 1)\frac{1 - A(z)}{2}$$
$$A(z) = \frac{-\alpha + \beta(1-\alpha)z^{-1} + z^{-2}}{1 + \beta(1-\alpha)z^{-1} - \alpha z^{-2}}$$
$$where\ \alpha = \frac{tan(\pi\frac{f_b}{f_s}) - V_0}{tan(\pi\frac{f_b}{f_s}) + V_0}\ and\ \beta = -cos(2\pi\frac{f_c}{f_s})$$

Where f_b is the bandwidth of the notch filter, f_c is the center frequency, f_s is the sampling frequency and $1 > V_0 > 0$ determines the depth of the notch. For $V_0 = 0.5$, the depth is 6dBs. By replacing the definition of $A(z)$ into the equation of $\Delta H(z)$, we obtain:

$$\Delta H(z) = \frac{(1 + \frac{(V_0 - 1)(\alpha + 1)}{2}) + \beta(1-\alpha)z^{-1} - (\alpha + \frac{(V_0 - 1)(\alpha + 1)}{2})z^{-2}}{1 + \beta(1-\alpha)z^{-1} - \alpha z^{-2}}$$

Magnitude response of the lowpass filter, each differential filter and the resulting filter by placing all of these filters in series, is depicted in figure 20. There resulting filters are not as good as each lowpass filter in basic FVN algorithm. They do not show that linear decay in dBs as in basic FVN algorithm. But it can be shown that the computational demands are dramatically reduced.

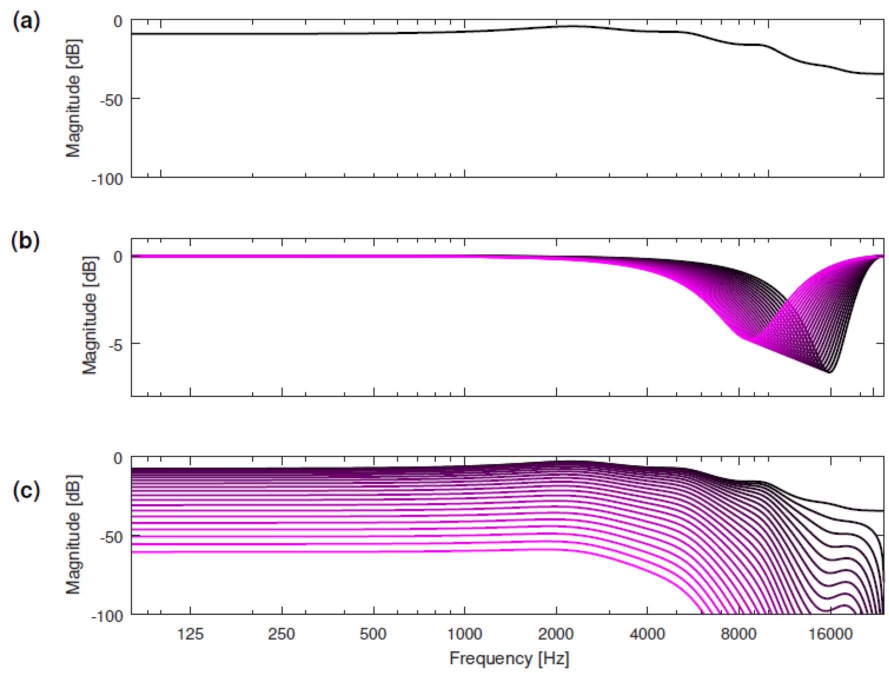

Figure 20: (a) Lowpass filter (b) Differential filters (c) Cascaded filters [9]

The second difference is in the segmentation of impulse response. In this algorithm, the room impulse response of desired concert hall is measured and segmentation is done based on STFT of it. New segment starts when the STFT of new segment is sufficiently different from the STFT of previous segment. An example of this method of segmentation which provided in the paper is shown in figure 21.

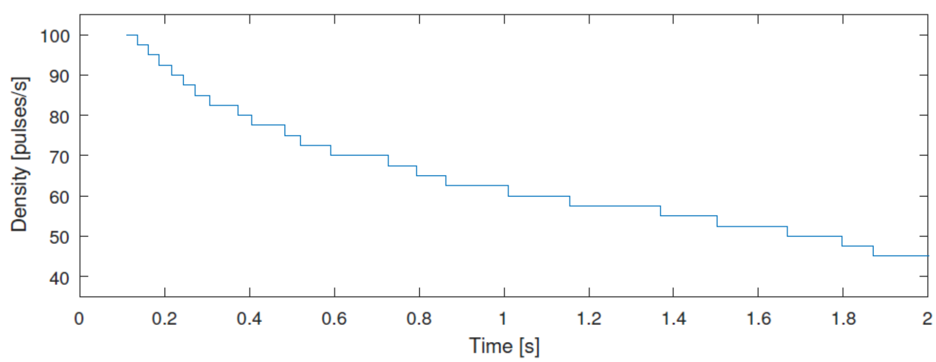

Figure 21: Segment lengths and densities

The implementation of these algorithm is pretty much like the basic FVN algorithm. As there was no desired reference room impulse response, length of the segments was modulated randomly. After the length of each segment is calculated, it is multiplied by a random variable which has a $U\sim[0.5,1.5]$ distribution. The center frequency of notch filters starts from 16 kHz and is multiplied by $\sqrt[N]{\frac{5}{16}}$. It means that the center frequency of the last notch filter is 5 kHz. Magnitude response of these filters are depicted in figure 22.

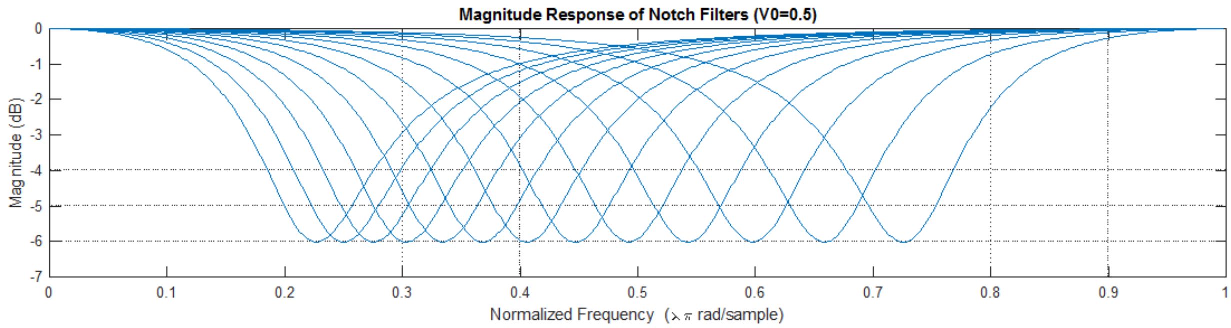

Figure 22: Magnitude response of differential (notch) filters

Other parts of these implementation are just as discussed in previous part.

Comparison and Conclusion

All of the mention algorithms have time-varying impulse response because of the velvet noise part. Therefore, impulse response of them is time variant; and so is their frequency response. But it might be helpful, to see what happens to an impulse when it is fed into the system as an input. To analyze the time and frequency properties of these systems, spectrograms of their impulse response is shown in figures 23 to 25. From now on, all of the analysis and comparisons are done using the wet output of the program (not the output of the mixer).

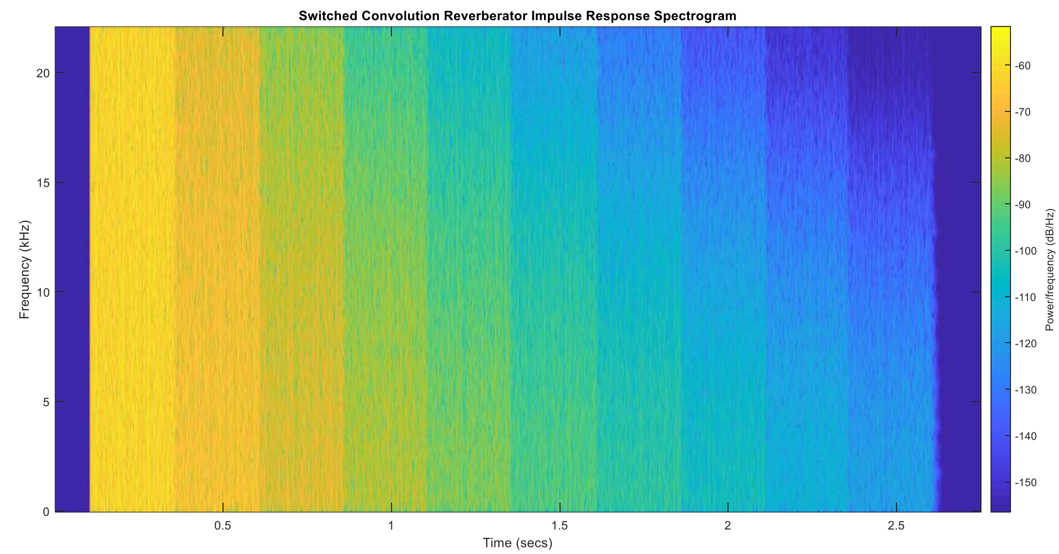

Figure 23: Spectrogram of impulse response of switched convolution reverberator

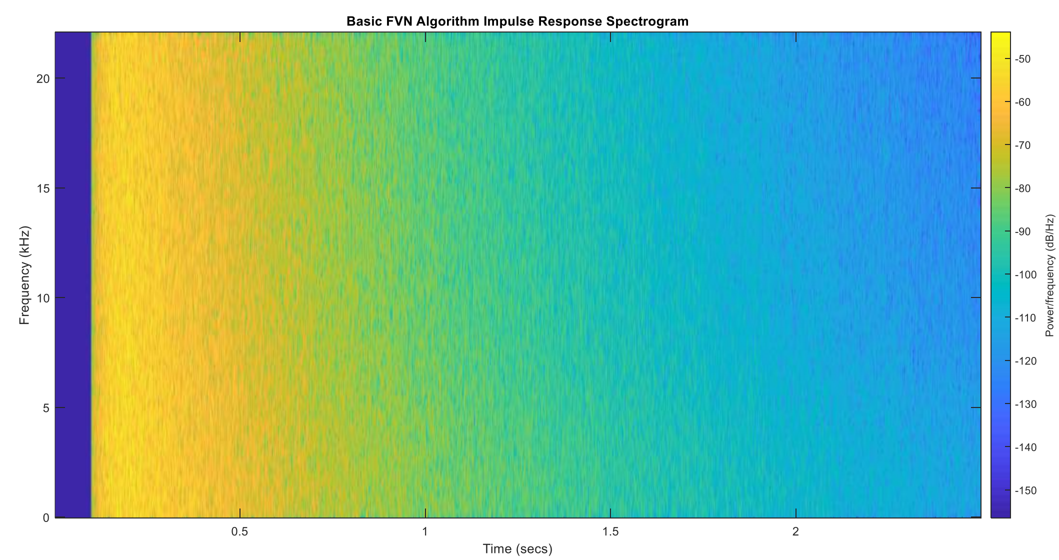

Figure 24: Spectrogram of impulse response of basic FVN algorithm

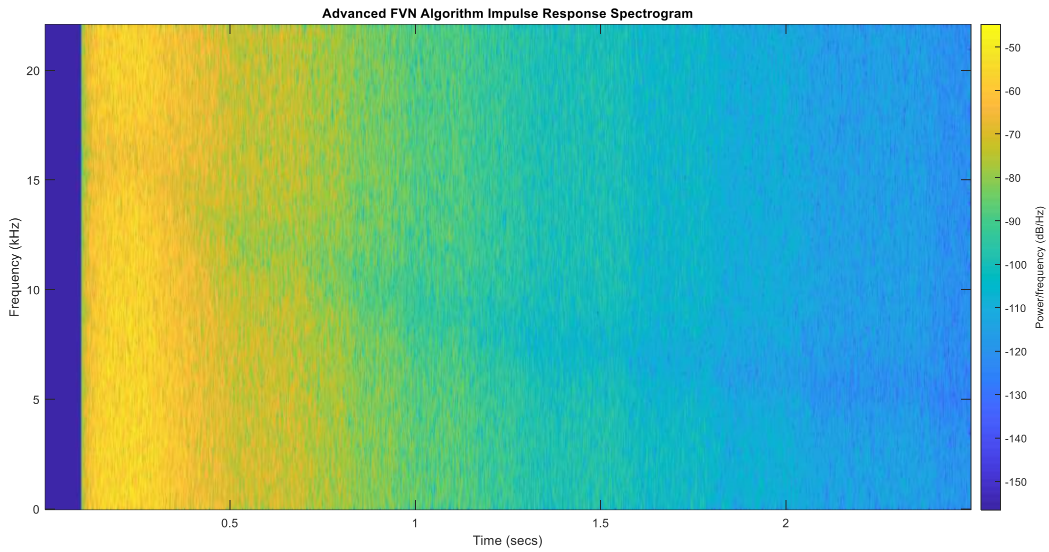

Figure 25: Spectrogram of impulse response of advanced FVN algorithm

As it is apparent in figure 23, some discontinuities in time domain is present in impulse response of switched convolution reverberator which could be because of a delay line in the feedback loop. At the same time, this reverberator shows better time-frequency properties, as higher frequencies decay faster. This property is present in all spectrograms but the latter two are approximately flat at all times steps.

To test the abilities of these reverberators more, four different inputs signals were fed into them. One of them is a steady sinewave, one is a short shout, one is a synth piano and the last one which is more realistic is 16 seconds of drum.

As predicted in [4], steady sinewaves are problematic in artificial reverberators, especially the ones with feedback loop. The Switched Convolution Reverberator does not also generate a good continuous output for the short shout. On the other hand, he produces a very warm reverberated sound for synth piano input and a decent output when fed with drum input. FVN algorithms do not provide us with a warm output then they are fed with the synth piano input, which could be caused because of their flat spectrum at every time step. Although, the produce better outputs when fed with steady sinewaves or short shout; because their impulse response is continuous. It should be noticed that synth piano and drum inputs, which are non-stationary signals, are closer to the real world; stationary sinewaves are not of that much of interest as non-stationary signals in the world of music. Therefore, although they are important in characterizing the system, they are not a big downside or an advantage for a musical system. But in the case of short shout, as it might happen in musical signals, the Switched Convolution Reverberator is not as good as FVN algorithms. Furthermore, some changes that were made during the implementation of these algorithms might have caused changes in the final results.

In the final conclusion, Switched Convolution Reverberator looks like the most natural sounding reverberator among these three. It has also a very unpredictable sound and also show some unwanted spectral coloration, at least more than the other two reverberators. All of them could be caused by the feedback loop of this reverberator. Unfortunately, because most of the reverberation process is done by the feedback loop, this reverberator has the least controllability. Among these three, every detail of filters is controllable in basic FVN algorithm. Therefore, it is the most controllable one. At the end, it should be pointed out that the advanced FVN method is very efficient computationally. More details could be found in [9].

All of the sound examples and codes are also provided. So that the reader can try repeating these experiments and analyze the results.

Velvet-Noise Feedback Delay Network [11]

This idea is proposed recently, in 2020. Unfortunately, because a lot of details were missing, it is not implemented. But a brief overview about this algorithm is considered. A block diagram of a conventional FDN is depicted in figure 26.

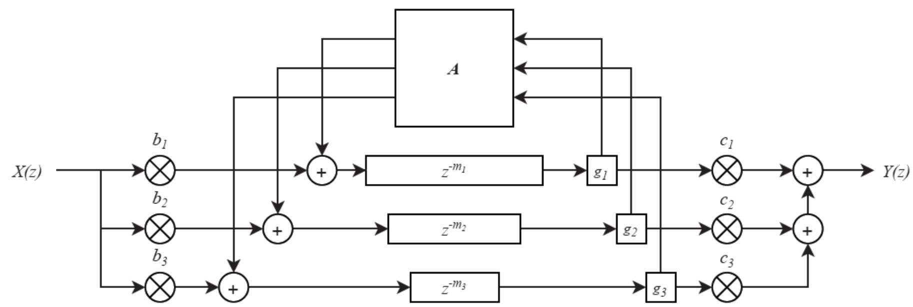

Figure 26: A conventional FDN

It can be shown that the transfer function of this system is given by:

$$H(z) = \boldsymbol{c}^T(\boldsymbol{D_m}(z)^{-1} - \boldsymbol{A})^{-1}\boldsymbol{b}$$

Where $\boldsymbol{c}$ and $\boldsymbol{b}$ are vectors that contain c and b coefficients and $\boldsymbol{D_m}(z)$ is a diagonal matrix that contains delay lines and potentially filters' transfer functions of each line.

The idea is to increase the echo density by using velvet noise sequences instead of $\boldsymbol{c}$ and $\boldsymbol{b}$ multipliers, as shown in figure 27.

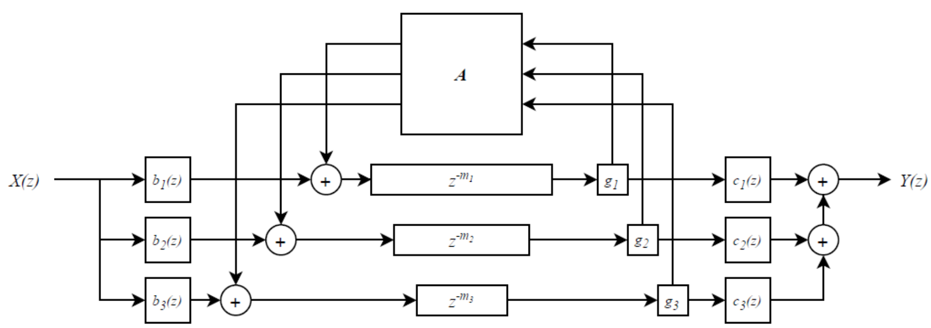

Figure 27: Proposed FDN in [11]

The transfer function of this system is given by:

$$H(z) = \boldsymbol{c}(\boldsymbol{z})^T(\boldsymbol{D_m}(z)^{-1} - \boldsymbol{A})^{-1}\boldsymbol{b}(\boldsymbol{z})$$

It can be shown that if multipliers at one end are replaced by velvet noise sequence with 2M impulses, echo density will be approximately multiplied by 2M; and If multipliers at one end are replaced by VNS with M impulses, echo density will be approximately multiplied by M^2. Notice that both methods require 2M impulses and have equivalent of computational cost. Therefore, for large value of M the latter idea is more logical.

References

[1] S. A. L. P. IEEE Trasncriptions On Audio, "Fifty Years of Artificial Reverberation," *Vesa Välimäki; Julian D. Parker; Lauri Savioja; Julius O. Smith; Jonathan S. Abel,* vol. 20, 2012.

[2] P. Rubak and E. G. Johansen, "Artificial Reverberation Based on a Psuedo-Random Impulse Response," in *AES 104th Convention*, Amsterdam, 1998.

[3] P. Rubak and L. G. Johansen, "Artificial Reverberation based on a Pseudo-random Impulse Response II," in *AES 106th Convention*, Munich, 1999.

[4] M. Karjalainen and H. Jarvelainen, "Reverberation Modeling Using Velvet Noise," *AES 30th International Conference,* 2007.

[5] A. D. Pierce, Acoustics - An Introduction to Its Physical Principles and Applications, 2019.

[6] L. Schreiber, "Was empfinden wir als gleichformiges Raushen?," *Frequenz,* vol. 14, 1960.

[7] K. S. Lee, J. S. Abel, V. Valimaki, T. Stilson and D. P. Berners, "The Switched Convolution Reverberator," *Audio Engineering Society,* vol. 60, 2012.

[8] B. Holm-Rasmussen, H.-M. Lehtonen and V. Valimaki, "A New Reverberator Based On Variable Sparsity Convolution," *Proc. of the 16th Int. Conference on Digital Audio Effects (DAFx-13),* 2013.

[9] V. Valimaki, B. Holm-Rasmussen, B. Alary and H.-M. Lehtonen, "Late Reverberation Synthesis Using Filtered Velvet Noise," *Applied Science,* 2017.

[10] [Online]. Available: http://research.spa.aalto.fi/publications/papers/applsci-fvn/.

[11] J. Fagerström, B. Alary, S. J. Schlecht and V. Välimäki, "Velvet-Noise Feedback Delay Network," *Proceedings of the 23rd International Conference on Digital Audio Effects (DAFx-2020),* 2020.