BELA Implementation of a Time-Variant Reverberation Algorithm for Reverberation Enhancement Syetem

Student Name: Yinghao Ma, Student ID Number: 200230638

Abstract

The reverberation algorithm is usually an LTI system. The room in the concert hall does not change, so the response does not change over time. However, this can lead to colouration and instability due to feedback caused by the close proximity of the microphones and speakers. Some time-varying systems can solve this problem. This paper implements a new time-varying variable reverberation algorithm on bela, based on a combination of delay-line, lowpass filter and comb filter and feedback, for reverberation enhancement systems. Such systems can often be used in electroacoustically enhanced rehearsal rooms. This particular application is briefly outlined, and other possible applications are discussed while the shortcomings of the experimental approach are analysed.

1 Introduction & Background

An electroacoustic enhancement system, also known as a reverberation enhancement system (RES), is designed to manipulate the sound field in a given space. This is implemented by using various components such as microphones, loudspeakers, and electronic circuits. These systems are commonly employed in concert halls and multipurpose halls where the acoustic design is inadequate or where the acoustical properties of the hall need to be altered. This is particularly important in halls that do not possess enough natural reverberant energy. Svensson [1] has thoroughly explained the theoretical principles of reverberation enhancement systems. Furthermore, Kleiner and Svensson [2] have presented a review of commercial enhancement systems.

Figure 1 illustrates the transfer functions that are fundamental to all electroacoustic reverberation enhancement systems. The actual transfer functions can vary significantly depending on the system's specific implementation and the room's acoustics. Typically, the reverberation algorithms used, such as the one described in [3], are linear time-invariant (LTI) systems. As a result, changes to the concert hall's physical properties do not occur over time, and the response remains constant. A significant issue with these systems is the presence of acoustic feedback, which can lead to $H_{LM}(\omega) \times G_{ML}(\omega)$, the feedback problems. Furthermore, the use of transducers with non-flat frequency responses, microphone placement (which is often far from sound sources for practical reasons), and the complexity of the reverberation algorithm can exacerbate these issues.

In some cases, using electroacoustic reverberation enhancement systems may result in colouration and instability, which can be caused by the proximity of microphones and loudspeakers. However, this issue can be addressed by utilizing time-variant systems. When a multi-channel RES is used in a room, it creates a multi-feedback system. There is a limit to the open-loop gain, which, if exceeded, causes uncontrollable positive feedback between the microphones and the loudspeakers. This limit is known

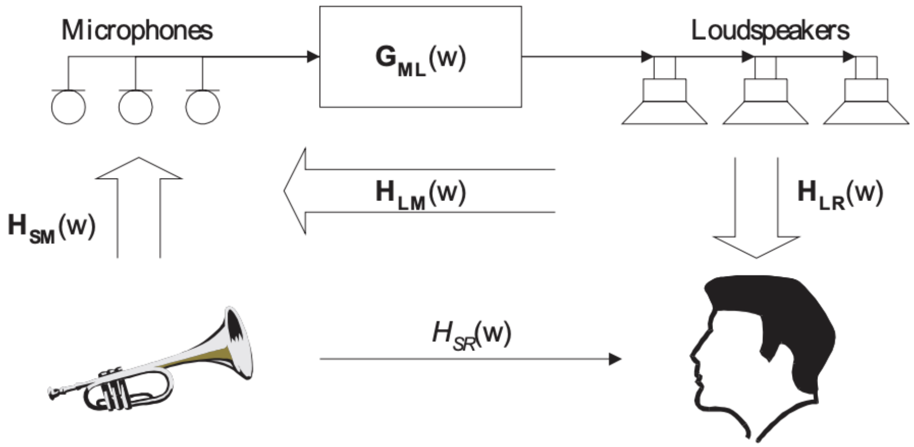


Figure 1: The transfer functions present in all RESs [1]

as the gain before instability (GBI). As the system approaches instability, the frequencies at which the feedback loop transfer function reaches its maximum values can produce changes in timbre or ringing tones, leading to discernible colouration even at levels 12 dB below the point of instability. Typically, the peaks in the transfer function are about 10 dB higher than the mean magnitude [4]. This means the system can become unstable at levels 10 dB lower than the wide-band average gain, which would suggest [1].

There are two approaches to addressing the feedback problem associated with electroacoustic reverberation enhancement systems. First, equalization can be used to reduce colouration. However, this approach does not eliminate the sharp peaks in the loop transfer function, which are heard as ringing tones and indicate potential instability. Second, time-varying algorithms can be employed to reduce the feedback issue more effectively. Time-varying algorithms continuously alter the loop transfer function, preventing the emergence of self-generating sustained peaks. Time variance can be achieved through amplitude, delay, phase modulation, or frequency shifting. Among these modulation methods, phase modulation is the most efficient, although it may be challenging to implement [5]. If the system transfer function $G_{ML}(\omega)$ includes a reverberation algorithm, time variance can also be integrated directly into the algorithm.

The application of such a time-variant reverberation algorithm has proven successful in a prototype of an electro-acoustically enhanced rehearsal room [6]. In this type of room, the stage of a concert hall is replicated, with an anechoic wall in place of the audience area (refer to Fig. 2). The anechoic wall is equipped with a reverberation enhancement system that generates the reverberant response that would typically return from the hall to the stage. There are numerous advantages to this system, such as requiring less space than an entire concert hall, early reflections that correspond to those of a real stage, a sound that is nearly identical to that of a real concert hall, a reasonable sound pressure level in the room, and the ability to adjust the room's reverberation time. Ultimately, this rehearsal room is a cost-effective solution for a symphony orchestra that cannot afford to conduct all of its rehearsals in a real concert hall.

The paper describes the implementation of a time-variant algorithm on BELA, which is a variation

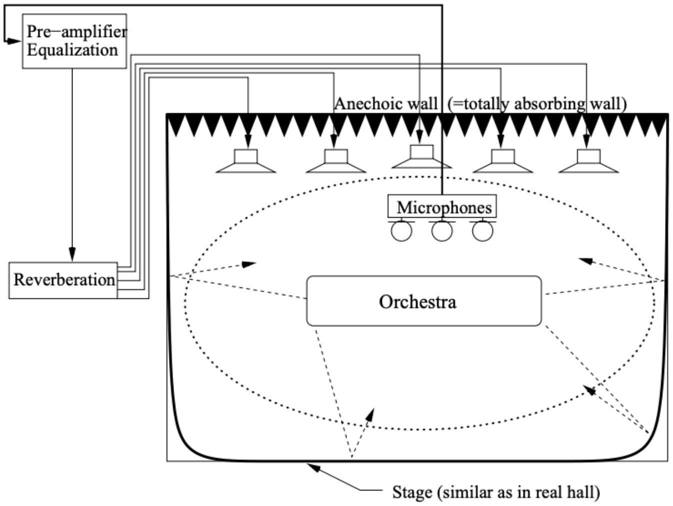

Pre–amplifier
Equalization
Reverberation
Anechoic wall (=totally absorbing wall)
Microphones
Orchestra
Stage (similar as in real hall)

Figure 2: The example application: Rehearsal room for symphony orchestra [6].

of the feedback delay network algorithm [7]. The algorithm consists of a comb all-pass filter at each delay line, with the feedback coefficient of the filter modulated by a few Hertz modulation frequencies. This modulation changes the group delay of each delay line, leading to a frequency shift of resonant frequencies. However, the shift is not uniform across all frequencies. If each delay line in the FDN has different modulation frequencies, there is no audible pitch shift [8]. This approach has proven successful in creating a physically small but acoustically large rehearsal space for a symphony orchestra [9].

2 Design & Implementation

2.1 Button

To start with, my plan is to add a button that allows me to switch between the original sound and the sound with added reverberation. This implementation will be similar to the one used in the step-sequencer project.

2.2 Circulat Buffer

Following that, I decided to focus on implementing a specific branch that included a delay line, a low-pass filter, and a comb all-pass filter. Once I had successfully implemented this branch, I replicated the same process for all the other branches. For the delay line, I utilized the circular buffer technique that we learned in week 5.

2.3 Low-Pass Filter

After studying the given information, I proceeded to apply the low-pass filter $H(z)$ depicted in figure 3. The results of the frequency response have been showcased in figure 4. The upper subfigure displays

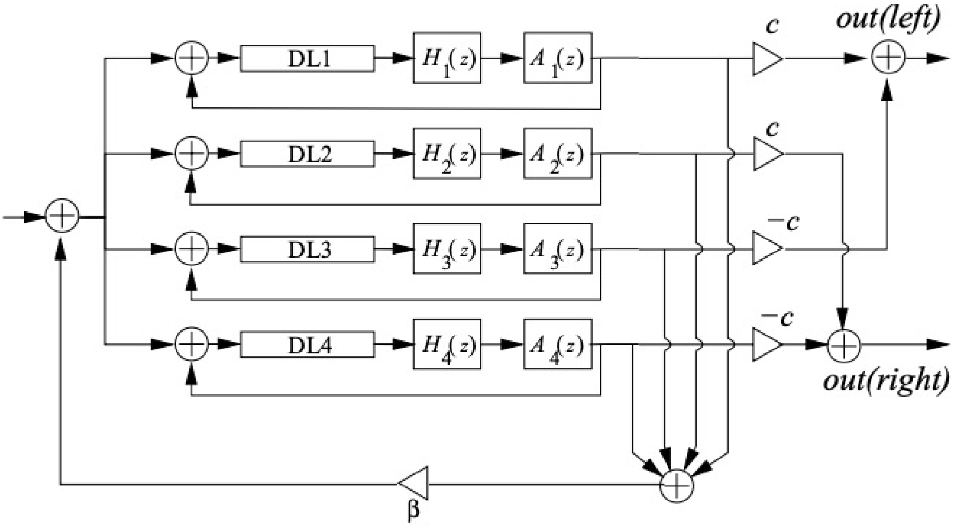
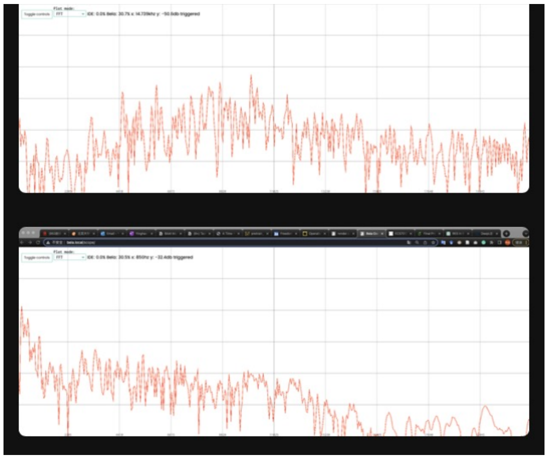


Figure 3: The applied reverberation algorithm, containing four channels [3].



IDE: 0.0% Bela: 30.7% x: 14.729khz y: -50.6db triggered

IDE: 0.0% Bela: 30.5% x: 850hz y: -32.4db triggered

Figure 4: The FFT energy response before and after applying the low-pass filter.

the response before applying the low-pass filter, while the lower subfigure illustrates the response after applying the low-pass filter. Upon observing the last column, it can be noticed that the amplitude values are consistently greater than the last row before the low-pass filter was applied, but lower than the last row after applying the low-pass filter. This observation confirms that the implementation of the low-pass filter has been carried out correctly.

The code is as follows.

```
Buffer_lpf[n] = (1/(1+b)) * (Buffer_tmp[n] + b*Buffer_lpf[n-1]);
```

The amplitude is set to $(1/(1+b))$ instead of $k \times (1-b)$ in case of the cutoff or the instability of the system.

2.4 Comb All-Pass Filter

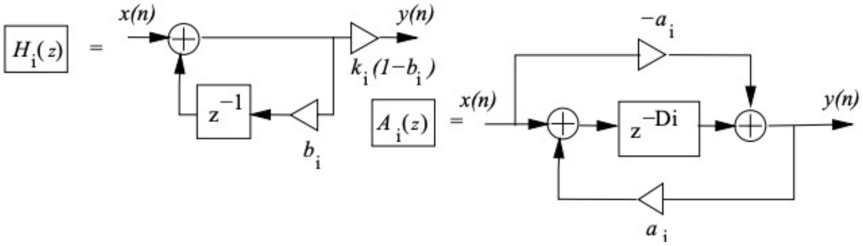


Figure 5: The low pass filter and comb all-pass filter in the reverberation algorithm [3].

In the field of signal processing, a comb filter (also known as a comb filter) superimposes a signal on its delayed counterpart to produce phase cancellation. The frequency response of a comb filter consists of a series of regularly distributed peaks, which look similar to a comb [1].

The i^{th} comb filter used in the algorithm is described in figure 5. A discrete-time system satisfies the following equation:

$$y[n] = -a_i x[n] + \cdot x[n-\tau] + a_i \cdot y[n-\tau] \tag{1}$$

The pseudo-code is like this

```
Buffer_out[n] = Buffer_lpf[n] + a * arBuffer_out[n] - a * Buffer_lpf[n];
```

After studying the given information, I proceeded to apply the low-pass filter $H(z)$ depicted in figure 3. The results of the frequency response have been showcased in figure 4. The upper subfigure displays the response before applying the low-pass filter, while the lower subfigure illustrates the response after applying the low-pass filter. Upon observing the last column, it can be noticed that the amplitude values are consistently greater than the last row before the low-pass filter was applied, but lower than the last row after applying the low-pass filter. This observation confirms that the implementation of the low-pass filter has been carried out correctly.

```
float a = 0.9 * gSineOscillator.process();
```

[1] https://en.wikipedia.org/wiki/Comb_filter

2.5 Implementation of Each Branch

My objective was to apply identical filters for each of the branches. To achieve this, I had to configure the delay-line, feedback amplitude, comb filter parameter, and low pass filter for each individual branch. Although I initially attempted to implement this in an object-oriented manner, I encountered several bugs due to my lack of experience in writing object-oriented code in C++. Consequently, I adopted a more direct approach where I faced the process head-on. I created 20 circular buffers and implemented each branch separately.

```
sum = beta * (Buffer[n-1] + Buffer3[n-1] + Buffer2[n-1] + Buffer4[n-1]);
Buffer_in[n] = in + sum ;
```

Furthermore, I configured the global feedback and linearly combined the outputs of different branches. Finally, to ensure causality, I modified the code by using the index $n-1$ instead of n.

2.6 MicroPhnoe Issues

I did not get the time to implement the microphone issues, but it is worth trying, as the algorithm for reverberation enhancement can only work with multiple microphones. And will leave it to the following work.

3 Evaluation

Initially, I assumed that the drum set utilized in the demo had pre-existing reverb, which was not an ideal scenario for showcasing the input samples. However, upon further investigation, I discovered an error in the output. The output consisted of four channels, and I had previously assigned the left audio to the first two channels and the right audio to the last two. Upon modifying the modulation frequency of the right channel, I noticed no discernible difference through my earphones. However, I realized that my earphones could only play the first two channels, and the output waveform had indeed changed. Therefore, I reassigned the left audio to the first channel and the right audio to the second channel. Listening to the output through earphones created the illusion of the pre-recorded audio having been recorded in a small room, with the further reverberation version creating a broader sound direction akin to a larger room.

Additionally, noticeable distortion in the high frequencies persists even after adjusting the delay-line and modulation values for the comb filters. Thus, finding the optimal set of hyperparameters for the algorithm requires more careful consideration.

4 Conclusion

This report focuses on implementing a time-variant reverberation algorithm to address feedback issues in reverberation enhancement systems. Additionally, the algorithm can be used in various applications, such as recording music in a small room with poor acoustics. However, the results of the implementation are not promising due to inherent weaknesses in the algorithm and the lack of guidance for setting hyperparameters. Furthermore, further testing is necessary to determine the audibility of the modulation and its effect on different frequency bands.

5 Acknowledge

I want to express my gratitude to Adan for helping me with the microphone issue, although I encountered difficulties with the distortion problem and did not have the opportunity to investigate the real-world microphone further. Additionally, I would like to thank Yazhou Li for providing valuable suggestions on my final project proposal. Lastly, I am thankful to Chin-Yun Yu and Christopher Mitcheltree for their helpful suggestions on debugging.

I'm not sure the Harvard Style for citations. So I use the ISMIR style.

References

[1] P. U. Svensson, "On reverberation enhancement in auditoria." 1996.

[2] M. Kleiner and P. Svensson, "Review of active systems in room acoustics and electroacoustics," in *INTER-NOISE and NOISE-CON Congress and Conference Proceedings*, vol. 1995, no. 5. Institute of Noise Control Engineering, 1995, pp. 39–56.

[3] J. Huopaniemi, V. Valimaki, M. Karjalainen, and R. Vaananen, "Efficient and parametric reverberator for room acoustics modeling," in *Proceedings: International Computer Music Conference 1997, Thessaloniki, Hellas, 25-30 september 1997.* The International Computer Music Association, 1997, pp. 200–203.

[4] T. Lokki and J. Hiipakka, "A time-variant reverberation algorithm for reverberation enhancement systems," in *Proceedings of COST G-6 Conference on Digital Audio Effects (DAFX-01), Limerick, Ireland*, 2001, pp. 28–32.

[5] J. L. Nielsen and U. P. Svensson, "Performance of some linear time-varying systems in control of acoustic feedback," *The Journal of the Acoustical Society of America*, vol. 106, no. 1, pp. 240–254, 1999.

[6] L. Tapio, N. Janne, and L. Tarja, "An electro-acoustic enhancement system for rehearsal rooms."

[7] J.-M. Jot and A. Chaigne, "Digital delay networks for designing artificial reverberators," in *Audio Engineering Society Convention 90.* Audio Engineering Society, 1991.

[8] U. Zölzer, X. Amatriain, D. Arfib, J. Bonada, G. De Poli, P. Dutilleux, G. Evangelista, F. Keiler, A. Loscos, D. Rocchesso *et al.*, *DAFX-Digital audio effects.* John Wiley & Sons, 2002.

[9] T. Lokki, J. Pätynen, T. Peltonen, and O. Salmensaari, "A rehearsal hall with virtual acoustics for symphony orchestras," in *Audio Engineering Society Convention 126.* Audio Engineering Society, 2009.