Reverber

VCarrara

Definition

Reverberation is a natural process where sound reflects on walls, surfaces and objects inside a room, and persists even after the original sound has faded. The reflections make the sound to decay, as well as to change its frequency spectra[1]. Small rooms create highly damped reverberation, whereas large rooms as gymnasium or caves sustain the reverberation for longer times.

The reverberation can be artificially produced using analog (plate, spring or Bucket Brigade Devices), or digital technologies[2].

Reverber

In a digital Reverber the audio reflections are simulated by delay lines, either by Feedforward Delay, Tapped Delay Lines, or Feedback Delay Networks. Whichever be the alternative, the multiple delayed signals must be filtered to change the spectrum and to avoid excessive presence of high frequencies on the output.

Some tests were made with well known Reverber algorithms, like the Freeverb, a single feedback delay line and a Feedback Delay Network (FDN). Best results come from Freeverb; however it is very process time consuming, and requires several large buffers to store the delay lines from the all-pass filters. Probably Freeverb can run on microcontrollers as single effect, but there is no evidence that it can share processing time in a multi-effect pedal. The single feedback delay line was already modeled in the Feedback Delay effect, but it lacks the typical reverberation of large rooms, due the missing filters. On the other hand, FDN produces good reverberations with large echo density, and requires modesty computation resources. Therefore FDN Reverber was a natural choice.

The selected FDN architecture was suggested by JOS[3] in his homepage, which is presented in Figure 1. There is just one input for all lines of the FDN, and the output lines are combined linearly. Instead of having multiple feedback lines, the FDN Reverber mixes them linearly by means of a matrix product, which increases reflections density. In order to reduce instability and the high frequency harmonics introduced by the feedback lines, the feedback gains g_i are replaced by low-pass filters. The input gains b_i provides the necessary adjustment of the reverberation amplitude as function of the delay times. As well as the feedback gains, the output c_i gains can also be replaced by a spectral band equalizer in order to achieve control over the reverberation process. Other modification introduced by JOS in the Reverber design is to mix the FDN output to the dry signal (input) weighted by a d gain, as can be seen in Figure 1.

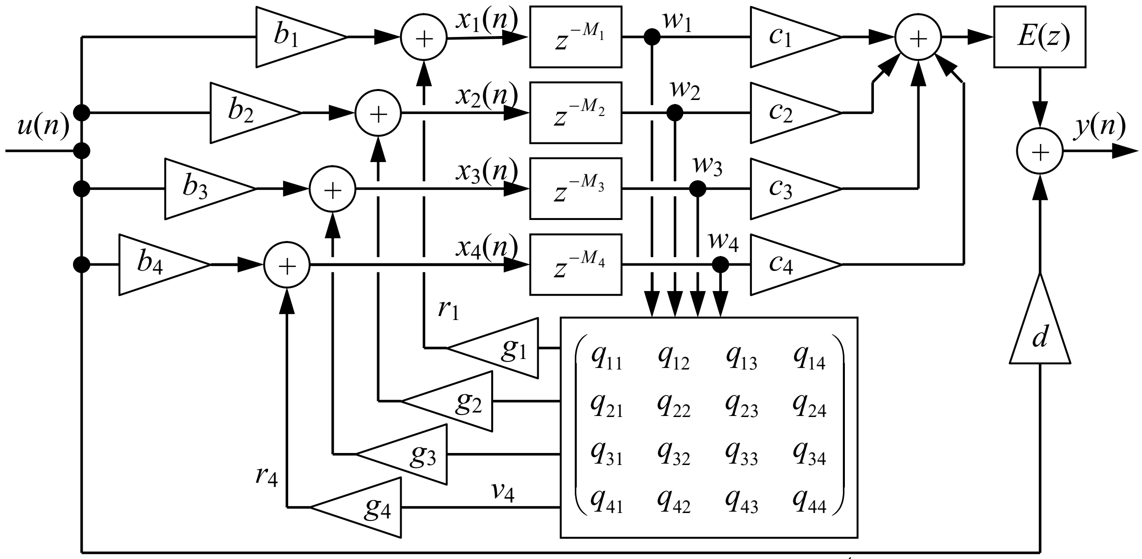

Fig. 1 – FDN Reverber (Adapted from JOS[4])

The difference equation of the feedback loop can be formulated with

$$\begin{pmatrix} x_1(n) \\ x_2(n) \\ x_3(n) \\ x_4(n) \end{pmatrix} = b \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} u(n) + \begin{pmatrix} g_1 & 0 & 0 & 0 \\ 0 & g_2 & 0 & 0 \\ 0 & 0 & g_3 & 0 \\ 0 & 0 & 0 & g_4 \end{pmatrix} \begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ q_{41} & q_{42} & q_{43} & q_{44} \end{pmatrix} \begin{pmatrix} x_1(n-M_1) \\ x_2(n-M_2) \\ x_3(n-M_3) \\ x_4(n-M_4) \end{pmatrix},$$

and the transfer function of this FDN is given by

$$\mathbf{X}(z) = b\,\mathbf{u}(n) + \mathbf{G}\,\mathbf{Q}\,\mathbf{D}(z)\,\mathbf{X}(z)$$

in which

$$\mathbf{D}(z) = \begin{pmatrix} z^{-M_1} & 0 & 0 & 0 \\ 0 & z^{-M_2} & 0 & 0 \\ 0 & 0 & z^{-M_3} & 0 \\ 0 & 0 & 0 & z^{-M_4} \end{pmatrix}.$$

Feedback matrix

Julios Orion Smith III[4] emphasizes that the feedback matrix A is simpler when its dimension is a multiple of 2 (2, 4, 8, 16, ...) because the coefficients are unitary and, therefore, it is not necessary to multiply the inputs by non integer coefficients. It was adopted then 4 FDN lines for the Reverber presented here. Moreover, this matrix must be orthogonal, such as, for example, a Hadamard matrix[4]. JOS also points out that the delayed signals can be replaced by a comb filters, such as FBCF, for example. The matrices suggested by JOS are:

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

which is stable if g is constant. One of the several Hadamard feedback matrix that can be used in FDN Reverber is given by

$$H_4 = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}.$$

Yet another possibility is the Householder matrix, whose main diagonal elements are positive, while the elements off diagonal are negative:

$$A_4 = \frac{1}{2}\begin{pmatrix} 1 & -1 & -1 & -1 \\ -1 & 1 & -1 & -1 \\ -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & 1 \end{pmatrix}$$

On tests the Hadamard H_4 matrix got a better reverberation than the others, but similar to the standard A matrix.

Delay time

High reverberation density can be achieved when all the time delays M_i are mutually primes. Of course the delay times are related to the size of a reverberation room; the larger the room, the longer the reverberation time. JOS presents some useful relations to compute the M_i delays as function of the mean free path of the sound for a specific room size. Anyway, good reverberations were achieved in tests with regular delay times ranging from 5 to 15 milliseconds, or with number of samples between 650 up to 2000. Since the reverberation suffers from phasing for low M_i and echoes when high values are used, it was adopted a fixed values for the M_i set: 653, 859, 1303 and 1987.

Low Pass Filter

As commented before, JOS states that the g_i gains provide the necessary damping of the reverberation. However, as the damping time depends on audio frequency, he suggests that g_i be replaced by a Low-Pass Filter $H(z)$ in the feedback delay line. In order to achieve the same damping time for different delay times, the low pass filter must also be function of the delay time, and therefore

$$H_i(z) = \frac{g_i}{1 - p_i z^{-1}},$$

which corresponds to the difference equation

$$r_i(n) = p_i\, r_i(n-1) + g_i\, v_i(n)\,,$$

such that

$$p_i = \frac{R_0^{M_i} - R_\pi^{M_i}}{R_0^{M_i} + R_\pi^{M_i}}$$

and

$$g_i = \frac{2\, R_0^{M_i}\, R_\pi^{M_i}}{R_0^{M_i} + R_\pi^{M_i}}$$

in which the constants $R_0^{M_i}$ and $R_\pi^{M_i}$ depend on the delay M_i:

$$R_0^{M_i} = 10^{-3M_i\,T/t_{60}(0)}$$

and

$$R_\pi^{M_i} = 10^{-3M_i\,T/t_{60}(\pi/T)}\,,$$

where $t_{60}(\omega)$ is the time the signal amplitude takes to decrease 60dB from unity at angular rate (frequency) ω. Thus $t_{60}(0)$ means a the damping time for null frequency, and π/T is the frequency at half of the sampling rate T, which is normally 22 kHz. Reasonable values for these times result with $t_{60}(0)$ twenty times greater than $t_{60}(\pi/T)$, or $t_{60}(0) = 20\ t_{60}(\pi/T)$. By making

$$\alpha = \frac{t_{60}(\pi/T)}{t_{60}(0)}\,.$$

and adopting $\alpha = 1/20$, the only parameter to be adjusted will be $t_{60}(0)$, which can vary from 0.1 second (almost no reverberation) up to 10 seconds (large reverberation).

Tonal corrector

JOS also says that $E(z)$, shown in Figure 1, is a low-order filter, called tonal corrector, that equalizes the total energy as function of the frequency band. The tonal corrector block diagram is shown in Figure 2, whose transfer function is given by

$$E(z) = \frac{1 - b\,z^{-1}}{1 - b}\,,$$

in which

$$b = \frac{1-\beta}{1+\beta}.$$

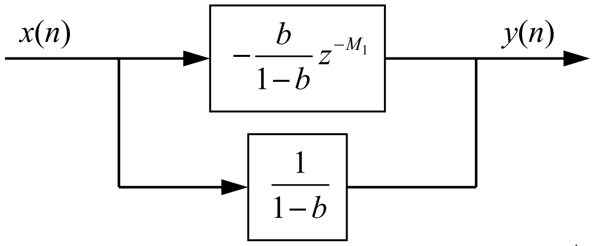

Fig. 2 – Tonal corretor (Adapted from JOS[4])

and such that $\beta = \alpha$. However, with $\alpha = 1/20$ (see above) the reverberation still sounds a little bit metallic, indicating the presence of high frequencies on output. Better results were obtained if $\beta = 10\alpha$. The difference equation of the tonal corrector is

$$(1-b)\,y(n) = x(n) - b\,x(n-1),$$

where $x(n)$ is the input to the tonal corrector and $y(n)$ its output.

Algorithm

The algorithm for the FDN Reverber is therefore

1) Create circular buffers x_i with dimension $M = \max(M_i)$ ($i = 1, \ldots, 4$) to store the delay lines, and initialize x_i with zeros. Make $M_i = 653$, 859, 1303, and 1987, for sampling rate $f = 44100$, and $T = 1/f$. Make $\alpha = 1/20$, $b_i = 1$ and $c_i = 1$, for $i = 1, \ldots, 4$.

2) Given the decay time $t_{60}(0)$ and the feedforward gain d, compute

$$t_{60}(\pi / T) = \alpha\, t_{60}(0),$$

   and the four values of

$$R_0^{M_i} = 10^{-3M_i T / t_{60}(0)},$$

   and

$$R_\pi^{M_i} = 10^{-3M_i T / t_{60}(\pi/T)}.$$

3) Now compute also the four values of

$$p_i = \frac{R_0^{M_i} - R_\pi^{M_i}}{R_0^{M_i} + R_\pi^{M_i}}$$

   and

$$g_i = \frac{2\, R_0^{M_i}\, R_\pi^{M_i}}{R_0^{M_i} + R_\pi^{M_i}}$$

4) Make $\beta = 10\alpha$ and compute

$$b = \frac{1-\beta}{1+\beta}.$$

5) Now implement the delay lines:

$$w_i(n) = x_i(n - M_i),$$

and the matrix product

$$\begin{aligned} v_1(n) &= w_1(n) + w_2(n) + w_3(n) + w_4(n) \\ v_2(n) &= w_1(n) - w_2(n) + w_3(n) - w_4(n) \\ v_3(n) &= w_1(n) + w_2(n) - w_3(n) - w_4(n) \\ v_4(n) &= w_1(n) - w_2(n) - w_3(n) + w_4(n) \end{aligned},$$

6) Apply the Low Pass Filter H_i:

$$r_i(n) = p_i\, r_i(n-1) + 0.5\, g_i\, v_i(n),$$

7) Store the value of $x_i(n)$ in the buffer

$$x_i(n) = b_i\, u(n) + r_i(n),$$

8) Apply the c gain and the tonal corrector

$$s(n) = c\,[w_1(n) + w_2(n) + w_3(n) + w_4(n)],$$

$$\upsilon(n) = \frac{1}{1-b}\, s(n) - \frac{b}{1-b}\, s(n-1),$$

9) Add the dry signal to the output

$$y(n) = \upsilon(n) + d\, u(n),$$

References

[1] Wikipedia. Reverberation. Available at: <https://en.wikipedia.org/wiki/Reverberation>, 2023.

[2] Wikipedia. Reverb effect. Available at: <https://en.wikipedia.org/wiki/Reverb_effect>, 2023.

[3] Smith III, J. O. CCRMA Home Page. Center for Computer Research in Music and Acoustics (CCRMA), Stanford University, Stanford (CA). Available at: <https://ccrma.stanford.edu/~jos/Welcome.html>, 2023.

[4] Smith III, J. O. FDN Reverberation. Center for Computer Research in Music and Acoustics (CCRMA), Stanford University, Stanford (CA). Available at: <https://ccrma.stanford.edu/~jos/pasp/FDN_Reverberation.html>, 2023.