Multiverb

Warren L. G. Koontz

January 30, 2014

Abstract

Multiverb is an audio signal processing algorithm that provides a reverberation effect to a digital audio signal. The algorithm is implemented as two C++ classes called Multiverb and DelayLine. These classes are used in turn to create LV2 and VST plug-ins for Audacity and other digital audio systems.

1 Theory

We begin with an overview of the theory behind Multiverb. For more detail please see [1] and [2].

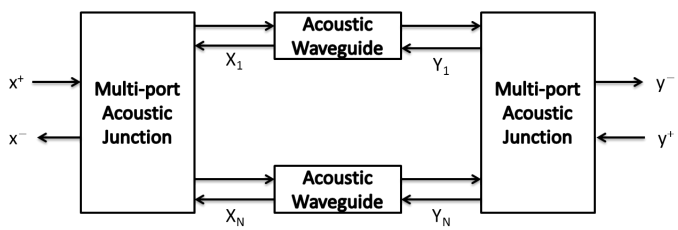

Figure 1: Multiverb Reverberation Model

Figure 1 illustrates the structure of Multiverb. which consists of two multi-port acoustic junctions connect by a bank of parallel wave-guides. Each wave-guide is modeled by a pair of damped delay lines denoted eastbound (left ot right) and westbound (right to left). The wave-guides introduce delay and frequency-dependent attenuation to the acoustic signal. There are N wave-guides and each junction has $N + 1$ ports, one for each waveguide and one for signal input and output. The signals are in the form of plane acoustic waves.

This inputs to the system are acoustic waves x^+ entering the left junction and y^+ entering at the right junction. The outputs are acoustic waves x^- exiting the left junction and y^- exiting the right junction. There are also vectors of acoustic waves entering the left junction from the wave-guides ($\mathbf{X}^+$), entering the wave-guides from the left junction ($\mathbf{X}^-$), entering the right junction from the wave-guides ($\mathbf{Y}^+$) and entering the wave-guides from the right junction ($\mathbf{Y}^-$).

The waves are related by the junction scattering matrices as follows:

$$\begin{aligned}\mathbf{X}^- &= \mathbf{A}\mathbf{X}^+ + \mathbf{B}x^+\\ x^- &= \mathbf{C}\mathbf{X}^+\\ \mathbf{Y}^- &= \mathbf{A}\mathbf{Y}^+ + \mathbf{B}y^+\\ y^- &= \mathbf{C}\mathbf{Y}^+\end{aligned}$$

where

$$\begin{aligned}\mathbf{A} &= \frac{1}{N}\mathbf{O}_N - \mathbf{I}_N\\ \mathbf{B} &= \begin{bmatrix}1 & 1 & \dots & 1\end{bmatrix}^T\\ \mathbf{C} &= \begin{bmatrix}1 & 1 & \dots & 1\end{bmatrix}/N\end{aligned}$$

$\mathbf{O}_N$ is a $N \times N$ matrix of ones, $\mathbf{I}_N$ is a $N \times N$ identity matrix and $\mathbf{B}$ and $\mathbf{C}$ are column and row N-vectors, respectively.

$\mathbf{Y}^+$ and $\mathbf{X}^-$ are linked by the eastbound delay lines; $\mathbf{X}^+$ and $\mathbf{Y}^-$ by the westbound. This linkage is expressed in a set of difference equations as follows:

$$\begin{aligned}Y_n^+ &= d_n Y_{n-1}^+ + g_n X_{n-D_n}^-\\ X_n^+ &= d_n X_{n-1}^+ + g_n Y_{n-D_n}^-\end{aligned}$$

where d_n, g_n and D_n are the damping, gain and sample delay of the n^{th} waveguide. The sample delays are prime numbers distributed over a range with a fixed upper limit. Given the sample delay, the damping and gain coefficients are determined from the low-frequency and high-frequency reverberation times (user-set parameters) by solving the following equations:

$$\begin{aligned}\frac{g_n}{1-d_n} &= 10^{-3D_n/(f_s T_{60}^{\text{LF}})}\\ \frac{g_n}{1+d_n} &= 10^{-3D_n/(f_s T_{60}^{\text{HF}})}\end{aligned}$$

where f_s is the sampling rate. Figure 2 is a signal flow diagram of the system. The notation $\mathbf{H}(z)$ represents the z transform of the waveguide difference equations.

Figure 3 shows a stereo reverb system incorporating multiverb and a stereo mixer copied from the freeverb reverberation system. The inputs to the stereo mixer are the left and right dry signals (u_L and u_R) and the left and right wet signals (y^- and x^-). The stereo outputs (v_L and v_R) are related to the inputs as follows:

$$\begin{bmatrix}v_L\\ v_R\end{bmatrix} = g_W \begin{bmatrix}\frac{1+s}{2} & \frac{1-s}{2}\\ \frac{1-s}{2} & \frac{1+s}{2}\end{bmatrix}\begin{bmatrix}y^-\\ x^-\end{bmatrix} + g_D\begin{bmatrix}u_L\\ u_R\end{bmatrix}$$

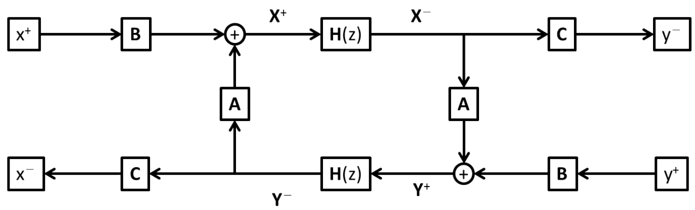

Figure 2: Multiverb Signal Flow Diagram

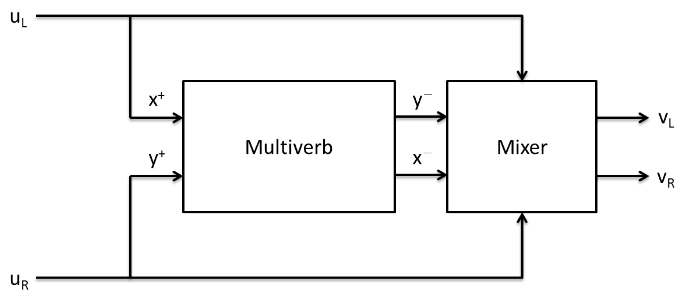

Figure 3: Multiverb Stereo System

where g_W and g_D are the wet and dry gains, respectively and s is called the "stereo separation factor", The stereo separation s is in the range $[0, 1]$.

In the case of a monaural signal, the right channel input y^+ and output x^- are not used and the monaural output is simply given by

$$v_L = g_W y^- + g_D u_L$$

Table 1 summarizes the parameters for this algorithm:

N	number of wave-guides (a.k.a. the order)
$D_{\max}$	sample delay upper limit
T_{60}^{LF}	low frequency reverberation time
T_{60}^{HF}	high frequency reverberation time
f_s	sampling rate
g_W	wet signal gain
g_D	dry signal gain
s	stereo separation

Table 1: Multiverb Parameters

2 C++ Implementation

The algorithm is implemented using two C++ classes: DelayLine and Multiverb. DelayLine basically implements a buffer and Multiverb uses DelayLine objects to implement the main algorithm.

2.1 DelayLine

This class models one direction of propagation in an acoustic waveguide. The input and output are sequences of samples of an audio signal. The output y and the input x are related by the difference equation

$$y_n = x_{n-D}$$

where D (an integer) is the sample delay. The DelayLine class includes the following methods:

void create(int D)
: Create and initialize (to all zeros) a delay line with sample delay D.

void push(float x)
: Enter new input sample (x), overwriting oldest sample and advancing pointer.

float get(void)
: Get current output sample. Returns sample value.

void clear(void)
: Reset delay line to all zeros.

2.2 Multiverb

This class models a reverberation mechanism consisting of two multi-port acoustic junctions connected by a bank of parallel wave-guides. The Multiverb class includes the following methods:

void SetX(float p)
: Methods to set given Multiverb parameters (e.g., low frequency reverberation time).

float GetX(void)
: Methods to retrieve given Multiverb parameters.

void Update(void)
: Update derived Multiverb parameters.

void Reset(void)
: Clear all delay lines and initialize state variables.

ClockProcess(float* S0, float* S1)
: Process one stereo pair of samples.

2.3 Additional Code

The Multiverb class is designed to work with toolkits for both LV2 and VST plug-ins. These toolkits provide code to complete dynamic library files that work as plug-ins for audio systems that support these types of plug-ins.

References

[1] Warren Koontz. Multiport acoustic models with applications in audio signal processing. *J. Audio Eng. Soc*, 61(10):727–736, 2013.

[2] Warren L. Koontz. Artificial reverberation using multi-port acoustic elements. *Proceedings of Meetings on Acoustics*, 20(1):–, 2013.