Implementation of Parametric Spring Reverb

Tomas Gajarsky

I. Introduction

This mini-project is based on the spring reverberation model proposed by Välimäki et al. in [1]. The emulation technique applied by Välimäki produces a parametric audio effect that can reproduce the qualities of the physical spring reverb and allows manipulation by the user as well. The chirplike pulses typical for spring reverbs are implemented by a cascade of identical all-pass filters. The initial chirp is being repeated and gradually blurred by a modulated multitap delay line in a feedback loop.

This report is organized as follows. First, the difference equations are calculated from the impulse responses of individual blocks. The blocks are then connected to form the final model.

All of the parameters, impulse responses and block diagrams used in this report come from [1].

II. Low-Frequency Feedback Delay Structure

A. Interpolated stretched all-pass filters

This structure is used for producing low frequency chirps. The filter's stretching factor K is related to the transition frequency f_{C} of chirps by

$$K = \frac{f_{\mathrm{s}}}{2f_{\mathrm{C}}} \tag{1}$$

where f_{s} is the sampling frequency. The result of the fraction can be a rational number, therefore an interpolated delay line is applied in the stretched all-pass filters. The transfer function of one filter is

$$A_{low}(z) = \frac{a_1 + \frac{a_2+z^{-1}}{1+a_2z^{-1}}z^{-K_1}}{1 + a_1\frac{a_2+z^{-1}}{1+a_2z^{-1}}z^{-K_1}} \tag{2}$$

where $K_1 = round(K) - 1$, coefficient a_1 affects the shape of the chirp and coefficient a_2 determines the fractional delay

$$a_2 = \frac{1-d}{1+d} \tag{3}$$

where $d = K - K_1$.

The difference equation of one interpolated stretched all-pass filter is calculated from transfer function eq 2 using inverse Z-transform [2] as follows.

$$A_{low}(z) = \frac{\frac{a_1+a_1a_2z^{-1}+a_2z^{-K_1}+z^{-1}z^{-K_1}}{1+a_2z^{-1}}}{\frac{1+a_2z^{-1}+a_1a_2z^{-K_1}+a_1z^{-1}z^{-K_1}}{1+a_2z^{-1}}} \tag{4}$$

$$\frac{Y(z)}{X(z)} = \frac{a_1 + a_1a_2z^{-1} + a_2z^{-K_1} + z^{-1}z^{-K_1}}{1 + a_2z^{-1} + a_1a_2z^{-K_1} + a_1z^{-1}z^{-K_1}} \tag{5}$$

$$\begin{aligned} y[n] =& a_1x[n] + a_1a_2x[n-1] + \\ & a_2x[n-K_1] + x[n-1-K_1] - \\ & a_2y[n-1] - a_1a_2y[n-K_1] - \\ & a_1y[n-1-K_1]; \end{aligned} \tag{6}$$

The cascade of M_{low} of these filters is realized by feeding the output of the first filter to the input of the second one and so on.

B. The dc blocking high-pass filter

The real world spring reverb units lack low frequencies near dc. In order to model this feature a high-pass filter with the following transfer function is implemented.

$$H_{dc}(z) = \frac{1+a_{dc}}{2}\frac{1-z^{-1}}{1-a_{dc}z^{-1}} \tag{7}$$

The cut-off frequency f_{cutoff} determines the coefficient a_{dc} by

$$a_{dc} = tan\left(\frac{\pi}{4} - \frac{\pi f_{cutoff}}{f_s}\right) \tag{8}$$

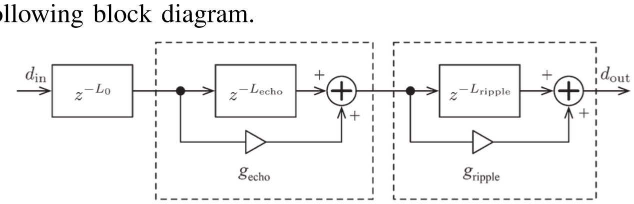
The difference equation of the high-pass filter is calculated from transfer function eq 7 using inverse Z-transform as follows.

$$\frac{Y(z)}{X(z)} = \frac{1 - z^{-1} + a_{dc} - a_{dc}z^{-1}}{2 - 2a_{dc}z^{-1}} \tag{9}$$

$$\begin{aligned} y[n] =& \frac{1}{2}(x[n] - x[n-1] + \\ & a_{dc}x[n] - a_{dc}x[n-1]) + \\ & a_{dc}y[n-1] \end{aligned} \tag{10}$$

C. Multi-tap delay line

The transfer function of multi-tap delay line is given by the following block diagram.



$$H_{md}(z) = z^{-L_0}(g_{echo} + z^{-L_{echo}})(g_{ripple} + z^{-L_{ripple}}) \tag{11}$$

The difference equation of the delay line is calculated from transfer function eq 11 using inverse Z-transform.

$$\begin{aligned} y[n] = &g_{echo} g_{ripple} x[n - L_0] + \\ &g_{echo} x[n - L_{ripple} - L_0] + \\ &g_{ripple} x[n - L_{echo} - L_0] + \\ &x[n - L_{echo} - L_{ripple} - L_0] \end{aligned} \tag{12}$$

where $L_{echo} = L/5$, $L_{ripple} = 2KN_{ripple}$, $N_{ripple} = 0.5$, $g_{ripple} = 0.1$, $g_{echo} = 0.1$ and $L_0 = L - L_{echo} - L_{ripple}$.

The total length of multi-tap delay line can be calculated from

$$L = T_D f_s - \tau(0) \tag{13}$$

where $T_D = 56ms$ and $\tau(0)$ is

$$\tau(0) = KM \frac{1 - a_1}{1 + a_1} \tag{14}$$

D. Delay line modulation

The process of delay length modulation produces blurring and uneven spreading of individual chirps. The chosen modulation signal is a white noise generated using the rand() function is filtered with a normalized leaky integrator defined by

$$H_{int}(z) = \frac{1 - a_{int}}{1 - a_{int} z^{-1}} \tag{15}$$

where $a_{int} = 0.93$. The difference equation of the normalized leaky integrator is calculated from transfer function eq 15 using inverse Z-transform as follows.

$$\frac{Y(z)}{X(z)} = \frac{1 - a_{int}}{1 - a_{int} z^{-1}} \tag{16}$$

$$y_{noise}[n] = x_{noise}[n] - a_{int} x_{noise}[n] + a_{int} y_{noise}[n-1] \tag{17}$$

The output signal of this filter is multiplicated by the modulation gain g_{mod}.

E. Equalizing the low frequency chirps

Välimäki et al. designed a stretched second-order IIR filter with following transfer function used for equalization of the low frequency chirps.

$$H_{eq}(z) = \frac{A_0}{2} \frac{1 - z^{-2K_{eq}}}{1 + a_{eq1} z^{-K_{eq}} + a_{eq2} z^{-2K_{eq}}} \tag{18}$$

where $a_{eq1} = -2R\cos 0$, $a_{eq2} = R^2$, $A_0 = 1 - R^2$ and $K_{eq} = floor(K)$. R is calculated from

$$R = 1 - \frac{\pi B K_{eq}}{f_s} \tag{19}$$

where $B = 130Hz$. $\cos 0$ can be obtained from

$$\cos 0 = \frac{1 + R^2}{2R} \cos\left(\frac{2\pi f_{peak} K_{eq}}{f_s}\right) \tag{20}$$

where $f_{peak} = 95Hz$. The difference equation of the stretched second-order IIR filter is calculated from transfer function eq 18 using inverse Z-transform as follows.

$$\frac{Y(z)}{X(z)} = \frac{A_0 - A_0 z^{-2K_{eq}}}{2 + 2a_{eq1} z^{-K_{eq}} + 2a_{eq2} z^{-2K_{eq}}} \tag{21}$$

$$\begin{aligned} y[n] = &\left(\frac{A_0}{2}\right)(x[n] - x[n - 2K_{eq}]) - \\ &a_{eq1} y[n - K_{eq}] - a_{eq2} y[n - 2K_{eq}] \end{aligned} \tag{22}$$

F. The output filter of low frequency structure

The low frequency chirps are suppressed by applying a tenth-order elliptic IIR low-pass filter with a cutoff frequency of 4750 Hz, 1-dB passband ripple, and 60-dB stopband suppression. The transfer function of this filter was not provided in [1]. Therefore the filter coefficients were obtained by designing this filter in Matlab [3] by using ellip() function as can be seen in the attached file filterDesign.pdf.

	a	**b**
1	-7.952953597940763	0.002664752967676
2	29.748452052047290	-0.013434509058057
3	-68.571433326253030	0.036183137660622
4	107.5493966631709	-0.065134704800960
5	-119.7162151903184	0.089024346522334
6	95.684751335115460	-0.097829543068792
7	-54.204321680605910	0.089024346522334
8	20.833974480614565	-0.065134704800960
9	-4.910863196724604	0.036183137660622
10	0.540083712167517	-0.013434509058057
11		0.002664752967676

TABLE I: The coefficients of the elliptic IIR low-pass filter

The difference equation of this filter was constructed from coefficients that can be seen in Table I.

$$\begin{aligned} y[n] = &b_1 x[n] + b_2 x[n-1] + b_3 x[n-2] + \\ &b_4 x[n-3] + b_5 x[n-4] + b_6 x[n-5] + \\ &b_5 x[n-6] + b_4 x[n-7] + b_3 x[n-8] + \\ &b_2 x[n-9] + b_1 x[n-10] - a_1 y[n-1] - \\ &a_2 y[n-2] - a_3 y[n-3] - a_4 y[n-4] - \\ &a_5 y[n-5] - a_6 y[n-6] - a_7 y[n-7] - \\ &a_8 y[n-8] - a_9 y[n-9] - a_{10} y[n-10] \end{aligned} \tag{23}$$

III. High-Frequency Feedback Delay Structure

A. First-order all-pass filter

The high chirps are modeled by using a cascade of first-order all-pass filters with transfer functions

$$A_{high}(z) = \left(\frac{a_{high} + z^{-1}}{1 + a_{high} z^{-1}}\right) \tag{24}$$

where $a_{high} = -0.6$. The difference equation calculated from the transfer function eq 24 is

$$y[n] = a_{high} x[n] + x[n-1] - a_{high} y[n-1]; \tag{25}$$

The cascade of M_{high} of these filters is realized similarly as in low-frequency structure II-A by feeding the output of the first filter to the input of the second one and so on until M_{high} number of the all-pass filters is used.

B. Delay line

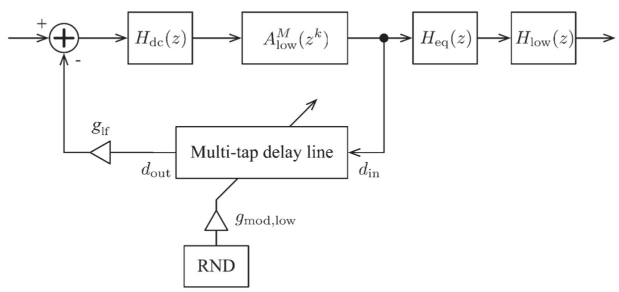
The simple delay line in high-frequency structure is defined by the difference equation

$$y[n] = x[n - L_{high}] \tag{26}$$

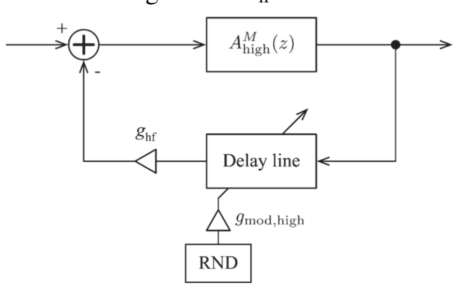
where $L_{high} = round(L/2.3)$. The length of delay line is modulated by the same filtered white noise as described in II-D.

IV. Connecting the parts

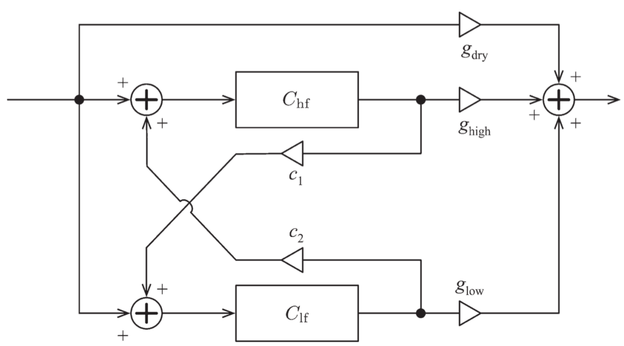
The block described in previous sections II and III were connected to bigger blocks C_{lf} and C_{hf} respectively according to following block diagrams by distributing partial signals as can be seen in the attached file parametricSpring.pdf. The same applies for connecting the blocks C_{lf} and C_{hf} to form the final parametric spring audio effect.



The block diagram of C_{lf} structure where $g_{lf} = -0.8$.



The block diagram of C_{hf} structure where $g_{hf} = -0.77$.



The block diagram of the parametric spring reverberation audio effect where the feedback loop gain from C_{hf} to C_{lf} block $c_1 = 0.1$, the feedback loop gain from C_{lf} to C_{hf} block $c_2 = 0$, the gain of the C_{lf} block $g_{low} = 0.95$, the gain of the C_{hf} block $g_{high} = g_{low}/1000$ and the gain of the input (dry) signal $g_{dry} = 1 - g_{low} - g_{high}$.

V. User interaction

The user can adjust the gain of the delay line modulation g_{mod}, the gain of the wet signal g_{low}, the time delay T_D and the transition frequency f_c which is set to 4300Hz by default.

VI. Matlab implementation

The model implemented in Matlab [3] calculates the output signal sample-by-sample. The code can be seen in attached file filterDesign.m. The real-time version was not implemented, however the computational time is quite high, therefore we assume that additional changes proposed in [4] would have to be done to the model in order to implement it as a real-time plugin.

References

[1] V. Välimäki, J. Parker, and J. Abel, “Parametric spring reverberation effect,” *JOURNAL OF THE AUDIO ENGINEERING SOCIETY*, vol. 58, no. 7/8, pp. 547–562, 2010.

[2] T. H. Park, *Introduction To Digital Signal Processing: Computer Musically Speaking*. River Edge, NJ, USA: World Scientific Publishing Co., Inc., 2009.

[3] MATLAB, *version R2017a*. The MathWorks Inc., 2017.

[4] J. Parker, “Efficient dispersion generation structures for spring reverb emulation,” *Eurasip Journal on Advances in Signal Processing*, vol. 2011, pp. 1–8, 2011. VK: 122815.