1.0 Problem Description

Emulating reverberation algorithmically can be complex process, as there are a large number of parameters that need to be set in order to create a realistic reverb. Although the number of adjustable parameters is seen as an advantage for experienced users, it can be overwhelming for new users to understand, which leads them to simply using a preset, or potentially creating an unrealistic reverb, as they don’t actually understand how the signal is being modulated.

Therefore, by simplifying the parameters to allow the user to process a reverb, by choosing the size of the room they wish to emulate and selecting a dry/wet value in the Graphic User Interface (GUI) that has been designed (MoorerGUI.m). The modulated output signal that is heard, and graphically depicted, creates an interest into what is occurring to the signal.

This emulation has been designed using Moorer’s Reverberation structure, based off a number of sequential delay effects that will be discussed, and has been implemented into a user-friendly GUI for users to be able to easily emulate an effective reverb. They can further use this as a comparison when experimenting with an advanced reverb application, to successfully learn what each parameter alters.

2.0 Specifications

Moorer’s reverberator structure involves a number of modulated delay effects such as a tapped delay line, low pass comb filter, and all-pass filter, each requiring a number of variables.

Tapped delay lines are delays that have multiple reading points defined from the number of taps and the delay value in samples. These six tapped delays are summed together and are used to emulate early reflections as would occur in a real room.

The low-pass comb filters are delayed signals that cause a variation in amplitude that result in the signal looking like a comb, creating a metallic coloration characteristic. They are used to increase the density of the delays. The gain factor is implanted as a low pass filter to simulate air absorption and create a smooth decay.

The all-pass filters are similar to a comb filter, but also have a feed forward component in order to create a smoother frequency response, by filling in the frequency dips as created using a comb filter, essentially creating a flat frequency
response. These also contribute to the density increase of the reverb in order to simulate late reverberation.

The structure for the emulation is processed in two stages; a tapped delay line; and six parallel low pass comb filters, which is processed through three all-pass filter.

The output is then a sum of the tapped delay lines, the output of the all-pass filter and the dry signal, dependent on the gain factor of the dry to wet signal, as can be seen in the signal flow in section 6.0.

By altering the variables of these functions, it will return a different sounding environment, based on early reflections, late reflections and the density of the reflections.

3.0 Implementation

In order to emulate a simplified reverberation using Moorer’s structure, a number of predetermined parameters need to be set, for the user to be only required to determine the size of the room and the dry/wet variable.

For each of the different sized rooms, a set of predetermined values were determined, that best suited to the target room size.

The tapped delay line uses six taps of different delays and gains, in order to emulate the early reflections from the walls of the room. It also provides the first reference to the size of the room. The larger the delay values, the larger the room. The gain also decreases, as the later delays are usually not as prominent as the first, due to the absorption of materials. The tap delays increases in steps as the size of the room increases.

The comb filters each have a different delay value that is calculated by creating mutually co-prime numbers. These numbers are used, as they don’t cause the reflections to be multiples of each other, and therefore minimise the colouration created by flutter echoes. These values are calculated by multiplying two different prime numbers for each delay value, increasing in steps with the room size. Each of the gain values, including the low pass filter, remained the same, for each of the room types. The values set are based off the equations that are described below.

The all-pass filters have the same gain value, and the suggested delay values, which remain the same for each room type.
4.0 Evaluation

Moorer’s simplified reverberator, which has been described above, is processed through a simple GUI and only requires the user to determine the dry/wet value and select the room size, which will process the predetermined signal (dryspeech.wav) and write the output based to the current folder, play the output wave and display the output graphically.

The function works as described to produce an output environment defined by the user. The function is processed using the equations described below and implementing them into the structure shown in section 6.0.

The tapped delay line function is implemented by:

\[ y(n) = b_0 x(n) + b_{M1} x(n - M_1) + b_{M2} x(n - M_2) + b_{M3} x(n - M_3) \]

Where \( x \) refers to the input, \( M_1-3 \) refers to the first three delay values, \( b \) refers to the gain of the output of each tapped delay and \( y \) refers to the output.

The comb filter functions were implemented by:

\[ y(n) = b_o x(n) - a_M y(n-M) \]

The gain is calculated using the below equations:

\[ b_o = \text{the gain on the filter} \]
\[ a_M = b_o \left(1 - b_o \right) \]

Where \( x \) is the input signal, \( b \) and \( a \) is the gain, \( M \) is the delay in samples, and \( y \) is the output.
The all pass filters are implemented by:

\[
y(n) = b_0 x(n) + x(n - M) - a_M y(n - M)
\]

Where \(x\) is the input signal, \(M\) is the delay value, \(a\) and \(b\) is the gain, and \(y\) is the output signal.

Through using a combination of these functions that have been explained, and following the signal flow shown on the following page, it makes it possible to develop Moorer’s reverberator.

5.0 Performance

Its performance may be assessed by providing the DSP application to a number of participants who don’t have experience in DSP. By determining if this function would make it easier for them to emulate reverberation, and contribute to them learning about what’s involved in creating an effective realistic reverb.

It would be done by using this emulation as a base, and experimenting with a more advanced application, in order to create the same sounding reverberation for each room type. This will enable them to understand what’s involved in emulating a realistic reverb, instead of just selecting a preset on a plug in, as majority of people that don’t understand do, and don’t disregard what each of the parameters do.
6.0 Moorer's Reverberation Structure

7.0 Bibliography

