Multi-band Reverberation

By Craig Szwec: 440497433
Introduction

Reverb is commonly used in audio production to create a sense of space. In a naturally reverberant environment the reverb of a sound is created by the room in which that sound was played. Another method of reverberation can be achieved by creating an artificial reverb. Many different reverbs have been created that try to mimic the sound of a reverberant room, however digital signal processing allows for reverberant effects to be created that can not exist in nature. A good example of this type of unnatural reverb is the multi-band reverb.

A multi-band reverb is a reverb that may be applied over frequency bands with different reverberant parameters on each band being processed. While this may seem like a trivial difference to a single reverb acting across all frequency bands in the same way, this type of multi-band reverb can produce many interesting effects that can not be realized using conventional reverberant methods.

Function

Creating the reverb

The reverb model that was chosen to be used for this project was the Schroeder reverb consisting of tapped delay, comb and allpass filters. This model was chosen as it is reliable and effective at creating natural sounding reverberation without coloration due to comb filtering. (Schroeder, 1961)
**Splitting the signal**

A series of filters must be applied to the signal to split it into different frequency bands before being processed by the `schroeder` function. Three separate bands were chosen, a low, mid and high and by using the inbuilt function `fdesign` a filter was created for each band.

**Low Pass**

A number of parameters had to be chosen when designing the low pass filter and these were specified by the `fdesign.lowpass` function that was used. First the passband frequency of the filter had to be chosen, after some consideration this point was chosen to be around 500Hz as, according to many eq charts such as the Recommended Equalization Frequencies chart (Dennis, 1998), below this point is where much of the unwanted ‘boxiness’ of sounds lie that one would not like to have a large amount of reverberation applied to. This point was moved slightly higher so as to account for crossover between the lowpass and bandpass filters. The stopband frequency was then chosen, this was placed slightly higher than the passband frequency so that when combined with the 60dB stopband attenuation, a relatively sharp rolloff was created. Finally a passband ripple of 1 was applied to ensure minimum coloration of the signal from the filter.

![Magnitude response of lowpass filter](image)

By Craig Szwec: 440497433
**Band Pass**

The Bandpass filter was slightly harder to design due to there being more parameters to deal with. In a band pass filter there are 2 pass band frequencies, 2 stopband frequencies as well as 2 stopband attenuations. In addition to this these variables have to be carefully chosen to ensure that they correctly crossover with both the lowpass and highpass filters. It is easier to design the low and high pass filters first and then to implement a bandpass filter by first contemplating the correct settings and then adjusting these by using the `fvtool` function to view the response of the filter compared to the other filters.

```matlab
fvtool(FBP,FHP);
```

Inbuilt function used to view filter response

This line of script will compare the response of the bandpass filter (FBP), shown in blue, with the response of the highpass filter (FHP), shown in green.

![Comparison between band and high pass filters](image_url)
High Pass

%For highpass filter

SB_HP = 7000; %sets the stopband frequency for the highpass filter
PB_HP = 8000; %sets the passband frequency for the highpass filter
SBA_HP = 60; %sets the stopband attenuation for the highpass filter
PBR_HP = 1; %sets the passband ripple for the highpass filter

DHP = fdesign.highpass('Fst,Fp,Ast,Ap', SB_HP, PB_HP, SBA_HP, PBR_HP, fs);
%creates highpass filter

FHP = design(DHP, 'FIR');
%Implements filter design as object FBP

Script implementing highpass filter.

It can be seen that unlike the lowpass and bandpass filters, an FIR filter was implemented in the design of the highpass filter instead of an IIR. The reason that this was used is that upon experimenting with the two filter types the shape of the FIR seemed more appropriate for use as the highpass. This was largely due to the shallower rolloff created for the FIR when compared to the same filter implemented in IIR. This shape allows for a larger crossover between the mid and high frequencies. There is a small amount of ripple in the bandpass frequencies, however this will be unnoticeable in the signal as it does not move beyond +/-0.5dB.

By Craig Szwec: 440497433
Recombining the signals

When recombining the low, mid and high reverberated signals it is important to remember that all signals must be of the same length in order for them to be summed together. As three different reverberations were used each signal (low, mid and high) ended up as different lengths. To fix this seemingly problematic situation zero padding was used on the signals to make them all the same length.

First the length of each signal, reverb_low, reverb_mid and reverb_high was found and placed in a matrix (L). The length of the longest signal was then found using \( \text{max}(L) \) and placed in the variable \( \text{max\_length} \). With this information it was then possible to find the number of zeros needed to be added to each of the signals by subtracting \( \text{max\_length} \) from each signal in the matrix L. Finally the zeros were added on to each signal to make them all the same length.

\[
L = \begin{bmatrix}
\text{length(reverb\_low)} & \text{length(reverb\_mid)} & \text{length(reverb\_high)}
\end{bmatrix};
\]
%creates a matrix for the signal lengths

\[
\text{max\_length} = \text{max}(L);%finds the longest signal and call this max\_length
\]

\[
L(2,1) = \text{max\_length} - L(1,1);%finds how many zeros needed to be added to reverb\_low signal
\]

\[
L(2,2) = \text{max\_length} - L(1,2);%finds how many zeros needed to be added to reverb\_mid signal
\]

\[
L(2,3) = \text{max\_length} - L(1,3);%finds how many zeros needed to be added to reverb\_high signal
\]

\[
\text{reverb\_low} = \begin{bmatrix}
\text{reverb\_low} & \text{zeros}(L(2,1),1)
\end{bmatrix};%pads low band with zeros
\]

\[
\text{reverb\_mid} = \begin{bmatrix}
\text{reverb\_mid} & \text{zeros}(L(2,2),1)
\end{bmatrix};%pads mid band with zeros
\]

\[
\text{reverb\_high} = \begin{bmatrix}
\text{reverb\_high} & \text{zeros}(L(2,3),1)
\end{bmatrix};%pads high band with zeros
\]

Script to make signals the same length

After this was implemented and the resulting signals were all the same length simple summation could be used to add the three signals together into one multi-reverb signal. This signal was then normalized and the signal was written into a wave file using the \texttt{wavwrite} function.

By Craig Szwec: 440497433
Further Optimization
This script was intended to show that the concept of multi-band reverberation was achievable in the Matlab environment. It is by no means optimized to its full potential.

One main area for further optimization is the filters used to split the signal into the three frequency bands. The specific crossover frequencies could be adjusted to achieve a more optimized frequency distribution as well as improving the stability of each filter. Another interesting concept would be to allow the user to adjust each individual filter so that they can have a fully customizable filter section. This could be implemented in a GUI that could graphically reflect how the filters were working.

It should also be noted that the phase response and step response of the filters used was not taken into account in the design. This could have implications on practical use, however these effects would be minimal as reverberation is by its very nature a delay based effect and so phase change would have little effect on the output signal.

Another area that could be vastly improved is the way in which the inputs for the Schroeder reverbs have been altered for each frequency band. When making this function no particular theory was followed in altering these parameters apart from reducing the amount of reverb in the low frequencies and slightly increasing the high reverberation as in general the effects of low frequencies being reverberated is less desirable than that of higher frequency reverberation. If further research was undertaken into this area this function could be vastly improved both in efficiency and in the aesthetic result obtained from this function.
**Bibliography**


Schroeder, M. R. 1961


By Craig Szwec: 440497433