LAB REPORT 1

HEAD SHADOWING FILTER FOR 3D AUDIO WITH HEADPHONES

Zhibo Song (Boey) 311114059(SID)

Digital Audio Systems, DESC9115, Semester 1 2012
Graduate Program in Audio and Acoustics
Faculty of Architecture, Design and Planning, The University of Sydney

Introduction

There are several methods to achieve 3D sound in the headphone listening environment now. However, for virtual sound sources, engineers want simply create the localization information by digital process. Through making appropriate time delays and spectral characteristics, there are different methods to simulate natural spatial cues now. In this report we will show a solution of using the Matlab functions to create filters which can simulate those cues.

mathematical specification

As mentioned before, there are different spatial cues of 3D sound. Here we only focus on the head-shadowing effects. The approach shown below comes from Brown and Duda, they model the structural properties of the system pinna-head-torso. The model can be divided intro three parts:

- Head Shadow and ITD
- Shoulder Echo
- Pinna Reflections

First，they use a first-order continuous-time system, which is a pole-zero couple in the Laplace complex plane as an approximation of the head shadow effect.

$$s_z = \frac{\omega_0}{\alpha(\theta)} \quad (1)$$

$$s_p = -2\omega_0 \quad (2)$$

Where $\omega_0$is related to the effective radius a of the head and the speed of sound c by

$$\omega_0 = \frac{c}{a} \quad (3)$$

The position of the zero varies with the azimuth angle θ according to the function is

$$\alpha(\theta) = 1.05 + 0.95cos\left(\frac{\theta}{150^\circ}180^\circ\right) \quad (4)$$

Here, notice that the value 1.05, 0.95 and 150 are used to make approximation to the frequency response curves. The pole-zero couple can be directly translated into a stable IIR digital filter by bilinear transformation, and the resulting filter is

$$H_{hs} = \frac{(\omega_0+\alpha F_s)+(\omega_0+\alpha F_s)z^{-1}}{(\omega_0+F_s)+(\omega_0+F_s)z^{-1}} \quad (5)$$

The ITD can be obtained by means of a first-order allpass filter whose group delay in seconds in the following function of the azimuth angle θ

$$\tau_h(\theta)\begin{cases} -\frac{a}{c}cos\theta \quad if\ 0 \le |\theta| \le \frac{\pi}{2} \\ \frac{a}{c}\left(|\theta| - \frac{\pi}{2}\right) \quad if\ \frac{\pi}{2} \le |\theta| \le \pi \end{cases} \quad (6)$$

Matlab code

The Matlab code comes from DAFx, chapter 6, the function called hsfilter.

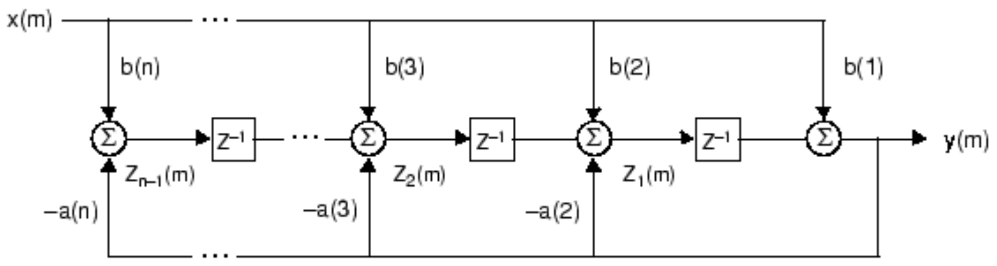
This filter will do 3 main operations.

1.define the position of the zero which varies with the azimuth angle

2.Create the IIR filter for a head shadowing effect

3.Obtain the ITD cues using a first-order allpass filter

From the code we can see that the author have made some changes on the formulas so they can suit the requirement of Matlab operations. The first thing is the function y=filter(b,a,X), from the Matlab help we can point out that The filter function is implemented as a direct form II transposed structure.



where n-1 is the filter order, which handles both FIR and IIR filters [1], na is the feedback filter order, and nb is the feedforward filter order. The input-output description of this filtering operation in the z-transform domain is a rational transfer function,

$$Y(z) = \frac{b(1) + b(2)z^{-1} + \ldots + b(nb+1)z^{-nb}}{1 + a(2)z^{-1} + \ldots + a(na+1)z^{-na}} X(z)$$

We also know that y = filter(b,a
,X) filters operate the data in vector X with the filter described by numerator coefficient vector b and denominator coefficient vector a. So the author follows the format and creates matrix of coefficient. Below are the details of this transition.

```
B = [(alfa+w0/Fs)/(1+w0/Fs),(-alfa+w0/Fs)/(1+w0/Fs)];
    % numerator of transfer Function H, notice that compare with the fomula (5) shown in the report,here the two parts of this
    % numerator are divided by w0+Fs, also have been translated into a
    % matrix, which will suit the format of Matlab y=filter(b,a,X) function
A = [1, -(1-w0/Fs)/(1+w0/Fs)];
    % denominator of transfer Funtion, with the same operations as above
if (abs(theta) < 90) %transfer the assumption from polar coordinates to cartesian coordinate
  gdelay = -Fs/w0*(cos(theta*pi/180) - 1);
```

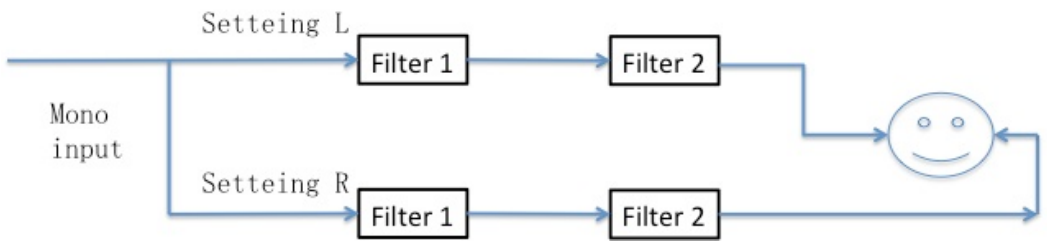
```
else
  gdelay = Fs/w0*((abs(theta) - 90)*pi/180 + 1);
end    %this group delay provide the ITD
a = (1 - gdelay)/(1 + gdelay);
        % allpass filter coefficient
out_magn = filter(B, A, input); % the first filter is a stable IIR digital filter
output = filter([a, 1],[1, a], out_magn); % the second filter is the group delay
```

Application

In order to apply this function, first we need to find a wave file which sounds as a single sound source. The source I used is a violin sound, I changed it into mono format. To see if the function do have effect, I made a group of results presenting the sounds come from different azimuth angles. The 3 settings are 20 degree, 45 degree, 80degree. One thing need to notice here is that to a given input sound,we need to apply the function twice, for the left and right ears with opposite values of argument theta. When get the two results, we need to feed one of them to the left ears and the other to the right ear. To achieve that, we can create the stereo sound as creating a 2 column matrix.

Processing figure and wave file's name

The input wave file is called 'so.wav', there are 3 output files, called 'so20.wav', 'so45.wav' and 'so 80.wav'. The function file called hsfilter.m, there is another file shows the setting process and the application details called 'application hsfilter'.



REFERENCES

[1] Francis Rumsey (2001) *Spatial Audio*, chapter 3

[2] Udo Zölzer, John Wiley & Sons, (2002), *Digital Audio Effects,* chapter 6

[3] Durand R. Begault (2000) *3-d Sound for Virtual Reality and Multimedia*, chapter 2&4

[4] Wiley (2007) *spatial sudio processing*, chapter 7

[5] C. Phillip Brown and Richard O. Duda, *Fellow, IEEE* (1998)*A Structural Model for Binaural Sound Synthesis*