Music 420
Winter 2005-2006
Homework #3
Phase Delay, Artificial Reverberation
140 points
Due in two weeks (09/02/2006)

Note for STK part of this Assignment.

For this Homework, all the Matlab part and theory questions related to lab problems are to be submited on PAPER (including Matlab functions/scripts/figures) with the rest of the theory problems.

STK/C++ Lab assignments parts are to be submitted electronically at
http://coursework.stanford.edu.
For each assignment, submit a single archive (`zip` or `tar.gz`) file containing all code/figures/output for the assignment. For each problem in the assignment, create a sub-directory in the archive named `hwX_pY`, where `X` is the homework number, and `Y` is the problem number, and place all code/figures/output for that problem in that sub-directory.

Starter code MyJCRev.tar.gz[1] and FDNJot4LP.tar.gz[2] and soundfiles[3] for this assignment may be downloaded from the class homework Web directory.[4] To uncompress the `.tar.gz` files, first save them into your STK `myproj` directory (typically `~/stk/stk/myproj`). Then use the following command in a terminal at the same directory level:

```
tar -zxf filename.tar.tz
```

The directory <`filename`> should then be created.

1. (25 pts) **Phase and Group Delay**

   (a) For the following simple filters, sketch the phase delay. Also sketch the group delay if it differs from the phase delay.

      i. (3 pts) $H(z) = z^{-M}$

[1] http://ccrma.stanford.edu/~jos/hw420/hw3/givenCode/MyJCRev.tar.gz
[2] http://ccrma.stanford.edu/~jos/hw420/hw3/givenCode/FDNJot4LP.tar.gz
[3] http://ccrma.stanford.edu/~jos/hw420/hw3/hw3sounds/
[4] http://ccrma.stanford.edu/~jos/hw420/index.html

ii. (3 pts) $H(z) = 1 + z^{-1}$

iii. (3 pts) $H(z) = 1 + 1.05z^{-2}$

iv. (3 pts) $H(z) = 1 + z^{-2}$

v. (3 pts) $H(z) = 1 + 0.95z^{-2}$

vi. (3 pts) $H(z) = \frac{0.95+z^{-2}}{1+0.95z^{-2}}$

(b) (4 pts) What is the maximum absolute value of the phase for the last 4 filters?

(c) (3 pts) Relate the filter in 1(a)v to the filter in 1(a)vi, regarding the phase delay.

2. (5 pts) **Schroeder Allpass:** Show that a single section of the Schroeder reverb of Fig. 2.4 of the text (p. 55) is allpass.

3. (20 pts) The STK module `MyJCRev` will implement a Schroeder-type reverberator.[5] However, the reverberator output is fed into only two delay lines, for stereo output. The left and right channels correspond to `OutA` and `OutB`, respectively, in the reverberator diagram. Assume a sampling rate of 22050 Hz.

   Complete the `tick` method in the `MyJCRev.cpp` file. (Since the allpass sections are to be built using the `Rev` modules from the previous homework, you may also need to make changes to the instantiation of the `Rev` objects in the `MyJCRev` constructor.)

   Turn in your code for the `tick` method. Also plot the impulse response of the reverberator's left channel response on a linear amplitude scale and on a log magnitude (dB) scale.

4. (a) (30 pts) The STK module `FDNJot4LP` will implement an order 4 single-input, single-output feedback delay network with a one-pole lowpass filter

$$H_i(z) = g_i \frac{1 - a_i}{1 - a_i z^{-1}}, \quad i = 1, 2, 3, 4$$

cascaded with each of the four delay lines.

The basic order 4 SISO FDN has $B^T = [1, 1, 1, 1]$, $C^T = [1, 1, 1, 1]$, and feedback matrix $A = A_4$, where

$$A_N \triangleq g \left(I_N - \frac{2}{N} u_N u_N^T \right)$$

where I_N denotes the $N \times N$ identity matrix and $u_N = [1, \ldots, 1]^T$. (It may be helpful to refer to the third-order example in Figure 1.26[6] of the text.) When $g = 1$, A_N is a so-called *Householder reflection* about the vector u_N in N-space. Note that A_N is orthogonal ($A_N^T A_N = I_N$) when $g = \pm 1$ (as is any real rotation matrix). It is actually more efficient to implement the Householder reflection as written above than to calculate and implement the resulting four-by-four matrix.

[5] http://ccrma.stanford.edu/~jos/pasp/Schroeder_Reverberator_called_JCRev.html

[6] http://ccrma.stanford.edu/~jos/pasp/Single_Input_Single_Output_SISO_FDN.html

Complete the constructor and the `tick` method in `FDNJot4LP.cpp`. Use Jot's formulas to compute g_i and a_i.

Turn in your code for the constructor and the `tick` method. Test `FDNJot4LP.cpp` with an impulse input for the case $g = 1$, $t_{60}(0) = 2.0$, $t_{60}(\pi/T) = 1.0$, and M_i set to four delay line lengths in `MyJCRev.cpp` $(113, 337, 1051, 4799)$. Plot the impulse response on a linear amplitude scale and on a log magnitude (dB) scale.

(b) (10 pts) $t_{60}(\pi/T)$ must be constrained to what range of values for the filters $H_i(z)$ to be usable (stable)?

5. (20 pts) Complete indicated sections in the Matlab files `edcAndPlot.m` and `edrAndPlot.m` in order to calculate the Energy Decay Curve (EDC) and Energy Decay Relief (EDR) of a signal. Turn in the new code you wrote, and plot the EDC and EDR for the impulse response of an actual acoustical space and for the impulse responses of the reverberators you built in problems 3 and 4.

6. (10 pts) Sonically compare the "real" reverberator and the two artificial reverberators. Listen to their impulse responses and the results of processing an anechoic recording with them, and describe the differences that you note. (You will need to use the Matlab function `conv` to process the soundfile with the recorded cathedral reverberation. And it will take a while to compute!) Compare the reverberators visually, considering their impulse responses, EDCs, and EDRs.

7. (15 pts) Analyze the order 4 FDN reverberator impulse response from problem 4 by calculating its EDR and spectrogram (in dB units). Using the EDR, estimate the decay times near dc and near half the sampling rate and compare them to the values desired at design time. Do the same, using data from the spectrogram. To measure the t_{60}s you will need to complete the Matlab code provided (`estimateT60.m`), which will fit a straight line to log magnitude data across time frames for different frequency bins.

   Turn in the new code you wrote and the measured t_{60}s (four in total).