Music 420
Winter 2005-2006
Homework #2
Comb Filters, Lattice/Ladder Allpass Filters
125 points
Due in one week (01/26/2006)

Theory Problems

1. (5 pts) Find the poles of the recursive comb filter

$$y(n) = x(n) + y(n-8)$$

2. (5 pts) Find the poles of the recursive comb filter

$$y(n) = x(n) + 0.5y(n-8)$$

3. (5 pts) Plot the poles in the previous two problems on the same plot (you may want to refer to functions `pzmap` and `tf` in Matlab).

4. (5 pts) Let $H_1(z)$ denote the transfer function of the filter in problem 1, and let $H_2(z)$ denote that for problem 2. We can express $H_2(z)$ as a conformal mapping of $H_1(z)$, viz., $H_2(z) = H_1(\alpha z)$. Find α.

5. Given a filter's difference equation

$$y[n] = x[n] - y[n-3] - 0.25y[n-6]$$

   (a) (10 pts) Express the impulse response of the filter as a sum of N geometrically decaying sinusoids.
   (b) (5 pts) List explicitly the decay time-constant for each sinusoid.
   (c) (5 pts) Draw the system diagrams of the Direct Form II and Transposed Direct Form II implementations of the filter.

6. It is shown in the text that the nested allpass filter is equivalent to the 2-multiplier lattice filter shown in Fig. 1. This problem explores the relationship of this structure to the *Kelly Lochbaum ladder comb filter* depicted in Fig. 2.

   (a) (10 pts) Find the transfer function of the two-multiply filter in Fig. 1.
   (b) (10 pts) Find the transfer function of the four-multiply filter in Fig. 2.
   (c) (5 pts) Find the impulse response of the two-multiply filter in Fig. 1.

(d) (5 pts) Find the impulse response of the four-multiply filter in Fig. 2.

(e) (20 pts) Find a one-multiply implementation of the four-multiply filter in Fig. 2, and draw the implementation. [Hint: Write out the difference equation for each summer and factor out the coefficient in each of the two equations, and note that they multiply a common term.]

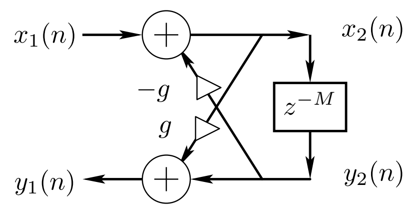

Figure 1: Two-multiply lattice filter.

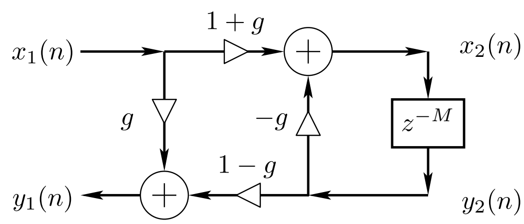

Figure 2: Four-multiply ladder filter.

Lab Assignments

Lab assignments are due one day after the theory portion of the homework (typically Friday afternoon at 5pm). Lab assignments 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.

1. (20 pts) Write an STK module called `Rev.cpp` which implements the two-multiply lattice filter shown in Fig. 1. The input arguments should be the two coefficients and the delay-line length. Write a small main program to test your module with an impulse input, and verify that you get the same numbers calculated by hand. Turn in your commented program listings and a print-out of your test results for the case $g = 0.5$ and $M = 10$ (include at least 35 output samples).

2. (20 pts) Write an STK module called `RevLP.cpp` which extends `Rev.cpp` to include the one-pole lowpass filter $y(n) = (1-a)x(n) + ay(n-1)$ *in the feedback loop only.* Thus, there should be one new argument a which can be set to 0 to obtain the former case (a useful intermediate test). Turn in your program listings and a print-out of your test results for the case $g = 0.7$, $M = 10$, and $a = 0.5$.