ECE-700 **Digital Signal Processing** **Winter 2007**

Homework #3 Jan. 17, 2007

HOMEWORK ASSIGNMENT #3

Due Wed. Jan. 24, 2007 (in class)

1. Polyphase Interpolation Filter Design: We have seen in the lecture that there are two different approaches to the design of a polyphase filter bank. In the indirect method, we design the master filter $h[n]$ and then decimate it. In the direct method, the polyphase filters are designed to have (approximately) constant group-delay and magnitude response. These two methods can be compared by looking at the magnitude and group-delays of the resulting polyphase banks, or by examining the magnitude response of the corresponding master filters.

   Assume interpolation factor $L = 15$, polyphase filter length $N = 11$ (i.e., master filter length LN), and input signal bandlimited to 0.8π radians.

   (a) What are the desired passbands and stopbands of $H(e^{j\omega})$ for the indirect method?

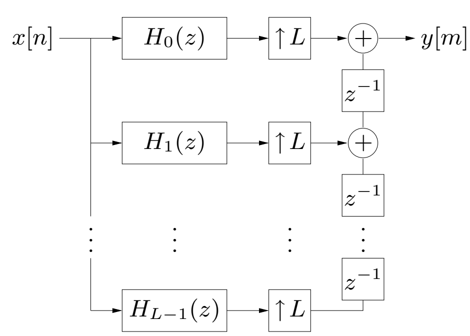
   (b) Using `firls`, design a length-LN linear-phase least-squares approximation to desired $h[n]$. Plot (i) the resulting DTFT magnitude response in dB, (ii) the polyphase filters' group delays, and (iii) the polyphase filters' DTFT magnitude responses in dB. Use a total of three plots.

   (c) Design an equiripple approximation to desired $h[n]$ via `firpm` and give the same plots as (b).

   (d) Using the Hamming-window method of homework 1.4c, design a length-N approximation to each polyphase filter $h_p[n]$, then construct the corresponding $h[n]$ (by interleaving). Finally, give the same plots as (b).

   (e) Based on your plots, which do you expect to yield the best interpolation performance?

2. MATLAB Polyphase Interpolator:



   (a) Generate length-1000 input signal $x[n]$ bandlimited to 0.8π radians, as in problem 2.5a.

   (b) Generate polyphase interpolated $y[m]$ using the structure above with $L = 15$ and the filters designed via the indirect least-squares approach of 3.1b. Demonstrate successful interpolation by plotting $x[n]$ for $n = [500 : 510]$ superimposed on the interpolates $y[m]$ (for the suitable range of m), as in 2.5d.

   (c) Repeat (b) for the indirect equiripple design of 3.1c.

(d) Repeat (b) for the direct windowed design of 3.1d.

(e) Is the interpolation performance consistent with your previous intuitions? What is going on?

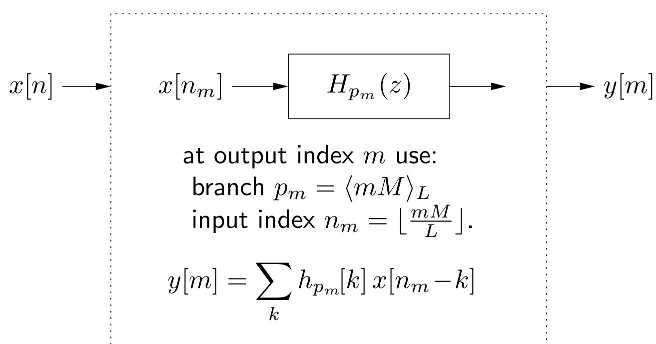
Hint: For debugging, recall the equivalence between polyphase and standard interpolation.

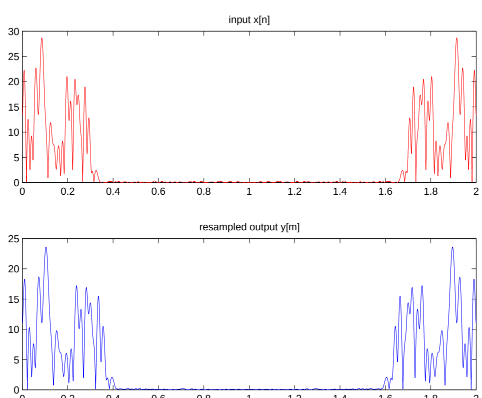
3. Rational Polyphase Resampling: Say $x_c(t)$ bandlimited to 300 kHz has been sampled at 1.75 MHz, and we want to resample the resulting signal to a rate of 1.44 MHz. In other words, $x[n] = x_c(nT_1)$ and $y[m] = x_c(mT_2)$ for $1/T_1 = 1.75$ MHz and $1/T_2 = 1.44$ MHz.

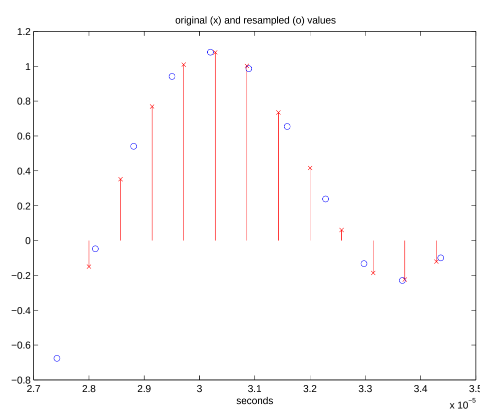
   (a) Generate length-150 $x[n]$ by appropriate filtering of `randn` output.

   (b) Describe the ideal master filter $|H(e^{j\omega})|$. Design a set of polyphase filters with 10 taps per filter (by a method of your choice) and plot the corresponding master filter's DTFT magnitude.

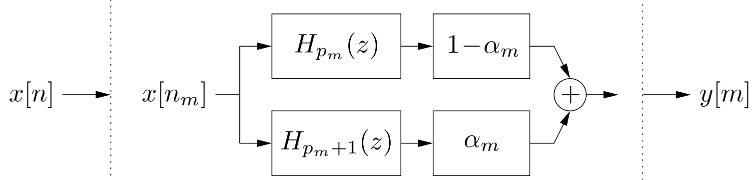
   (c) Implement the rational polyphase resampler (shown below). Plot the input and output DTFT magnitudes, then generate a time-domain plot which superimposes samples of the input and output over the range $t \in [28\mu s, 34\mu s]$. See example plots below. To line up the waveforms, remember to shift the output trace left by the delay τ caused by the resampling structure. (*Hint*: To determine τ, think about the effect of master filter delay in the equivalent standard (non-polyphase) resampler. Also, make sure that your implementation does not introduce unwanted delay; it should produce $y[0] = h[0]x[0]$.)







4. Arbitrary-Rate Polyphase Resampling: Here we repeat the resampling task in Problem 3 using the arbitrary-rate resampler illustrated below. We keep the resampling ratio $Q = \frac{144}{175}$, but implement the resampler using an $L = 10$ interpolation bank (rather than 144).



at output index m, use
branch $p_m = \lfloor \langle \frac{m}{Q} \rangle_1 L \rfloor$
input index $n_m = \lfloor \frac{m}{Q} \rfloor$
weight $\alpha_m = \langle \frac{mL}{Q} \rangle_1$.

$$y[m] = (1-\alpha_m) \sum_k h_{p_m}[k]\, x[n_m - k] + \alpha_m \sum_k h_{p_m+1}[k]\, x[n_m - k]$$

To implement H_{p_m+1} when $p_m = L-1$, we need the additional filter $H_L(z)$. This filter can be designed directly so that it approximates a group delay of $\frac{d-L}{L}$, or indirectly through decimation of a master filter with length $LN_p + 1$ (where N_p denotes polyphase filter length).

(a) Generate $x[n]$ as in problem 3a.

(b) Describe the ideal master filter $|H(e^{j\omega})|$. Design a set of polyphase filters with 10 taps per filter and plot the corresponding master filter's DTFT magnitude.

(c) Implement the arbitrary-rate resampler. As in problem 3c, plot the input and output DTFT magnitudes, then generate a time-domain plot which superimposes samples of the input and output over the range $t \in [28\mu s, 34\mu s)$. (*Hint*: This requires only minor modification of your code from 3c. Do not attempt this until you are sure that your previous code works correctly!)