Carnegie Mellon **ADSP** Electrical & Computer ENGINEERING

Advanced Digital Signal Processing (18-792)
Fall Semester, 2008
Department of Electrical and Computer Engineering

ASSIGNMENT 6

Issued: 10/16/08
Due: 11/22/08

Note: This problem set is shorter than normal because of the mid-semester break.

Reading: This problem set discusses various topics related to multirate DSP, based primarily on the material in Secs. 3.0-3.3 of Lim and Oppenheim (LO), focussing on Sec. 3.3. You should also look over Secs. 3.4-3.7 in LO even though we did not directly discuss that material in class. The applications of multirate DSP described in Sec. 3.7 are interesting, and quite possibly relevant to work you may be doing for this problem set!.

Prob1em 6.1: Even though the input signal to a particular speech processing system has a bandwidth of 4 kHz, the signal had been sampled at a frequency of 32 kHz. You have been asked to design a system that decimates (downsamples) the incoming signal by a factor of 4, from the sampling rate of 32 kHz to 8 kHz..

(a) Determine the coefficients for the appropriate FIR lowpass filter using the window design technique, as discussed in 18-491 and in Sec. 7.2 in the text for DSP by Oppenheim, Shafer, and Buck (OSB) or equivalent sections of other texts. Assume that your filter will use a Hamming window and will have 36 coefficients. What are the values of these coefficients? (You may use MATLAB to assist you in this calculation, if you wish.

(b) Given the characteristics of your filter and the bandwidth of 4 kHz of the incoming signal do you believe that the system will accomplish the desired task without distorting the signal? Why or why not? Consider both the transition bandwidth and the stopband ripple of the filter you design.

(c) Draw a signal flow diagram of an efficient implementation of your system using polyphase filtering techniques. Be sure to specify the values of the filter coefficients.

Problem 6.2: Problem 4.53 in Oppenheim, Schafer, and Buck (OSB), the textbook for 18-491 used for several years. If you do not have a cop of OSB, I will post a copy of the problem on the Web by Friday afternoon.

MATLAB Problems

Problem C6.1:

Introduction: In this problem we will implement a system that will change the sampling rate of incoming speech from a nominal value of 16 kHz to 20 kHz. You can either use MATLAB or your own C code to implement the change in sampling rate. The task is to be accomplished by interpolating by a factor of 5 and then decimating using a factor 4, using the combined polyphase upsampling and downsampling approach illustrated in Fig. 3.25 of LO.

Before you begin:

For this problem we will make use of the usual "Welcome to DSP-I" utterance used in Problem Set 5.

Note: Turn in the MATLAB scripts that you used to obtain your answers in all of the following problems.

Please consider the following issues in developing your implementation:

- You should check your work by listening to the playback of the resampled waveform using MATLAB or any other convenient playback venue.
- You can design the lowpass filter for the implementation any way you wish, although, of course the computational efficiency will be affected by your filter design. You should strive to develop a system that is reasonably computationally efficient, but that produces a waveform that sounds decent when you play it back. I will play back some of your reconstructions that I consider to be particularly good and bad realizations in terms of quality and efficiency.

What you must turn in on paper:

- A block diagram and description of your system as you implement it. Be sure to state the critical choices you made in designing the lowpass filter and other aspects of the system, along with how you arrived at your decisions.
- A plot of the magnitude and phase of your lowpass filter. This is most easily accomplished using MATLAB.
- "Stem" plots of the original filter and the set of polyphase subfilters used in your implementation. Remember that you can pad your sample response by appending zeros if needed to make Q a multiple of L, as defined in OL Sec. 3.3.
- Hard copy of your MATLAB scripts.

What you will submit electronically:

- A sample of the resynthesized sound, in ASCII or .wav format, similar to the input file that you used. Label the files with your andrew id (or something like that) and the suffix "`6_1`". "`rms6_1.dat`" or `rms6_1s.wav`", for example, would be my filename, (but not yours!). Email your sound file to Kshitiz Kumar, ccing me, preferably as an attachment. Do **not** use Microsoft Outlook to send your mail.
- A copy of the MATLAB script used to realize your solutions.

Problem C6.2:

Lim and Oppenheim Sec. 3.6.3 describes how multirate techniques can be used to reduce the computational complexity associated with implementing lowpass filters with very tight specifications (and by extension bandpass and highpass filters as well).

Consider a linear-phase FIR lowpass filter with the following design specifications:

- Passband cutoff frequency: $\omega_p = 0.010\pi$
- Stopband cutoff frequency: $\omega_s = 0.015\pi$
- Passband ripple[1]: $-1\ \text{dB} \le |H(e^{j\omega})| \le 0\ \text{dB},\ |\omega| \le \omega_p$
- Stopband attenuation: $|H(e^{j\omega})| \le -70\ \text{dB},\ \omega_s \le |\omega| \le \pi$

According to the MATLAB routine `sptool`, implementing of this filter directly using the Parks-McClellan algorithm (*i.e.* the MATLAB function `firpm`) requires close to 1000 delays (MATLAB could not confirm this because the filter itself could not really be designed.)

(a) Following the general procedures outlined in LO Sec. 3.6.3, design a system that realizes the specifications above with a one-stage multirate filter implementation that minimizes the total number of multiplications in its processing by minimizing the length of its FIR filters. This would be accomplished by developing a system as in LO Fig. 3.36 that decimates the input by a factor of *M* and then interpolates the result by the same factor of *M*. The filters $h_1[n]$ and $h_2[n]$ depicted in LO Fig. 3.36 must be designed using the MATLAB function `firpm`. You must specify the value of *M* that you chose, along with the sample responses of the filters $h_1[n]$ and $h_2[n]$. Describe how you can verify that the overall system meets the performance specifications.

(b) Repeat your work in part (a), develop a two-stage multirate filter implementation that minimizes the total number of multiplications in its processing and meets the system specification described above. You must specify the two values of *M* that you chose, along with the sample responses of the filter used for decimation and interpolation at each stage. Describe how you can verify that the overall system meets the performance specifications.

(c) Develop a table similar to LO Table 3.1 that compares the performance of the direct-form, one-stage, and two-stage filter implementations.

What you must turn in on paper:

- Block diagrams and descriptions of the two systems that you implement. A plot of the magnitude

---

1. Note that the values of passband and stopband attenuation are given in dB. If *x* has the dimensions of amplitude or magnitude (as opposed to energy or power), the value of *x* expressed in dB would be $20\log_{10}(x)$

response of the overall system.

- Hard copy of your MATLAB scripts.

What you will submit electronically:

- A sample of the resynthesized sound, in ASCII or .wav format, similar to the input file that you used. Label the files with your andrew id (or something like that) and the suffix "`6_2`". `rms6_2.wav`", for example, would be my filename, (but not yours!). Email your sound file to Khsitiz Kumar, ccing me, preferably as an attachment. Do **not** use Microsoft Outlook to send your mail.

- An analysis of how much computation is required. Specifically, estimate the number of storage parameters and multiplys needed to represent each second of input speech.

- A copy of the MATLAB script used to realize your solutions.

SPECIAL 18-792 STUDENT COMPETITION!! FREE EXTRA CREDIT!!

Two prizes of 10 bonus points will be added to the grade for this problem set for the student or students who obtain the absolute lowest number of filter lengths in the design of the multirate lowpass filter in Problem C6.2. One prize will be awarded for the best solution in part (a) of Problem C6.2 and one prize will be awarded for the best solution in part (b). Candidate systems must meet design specifications in order to be eligible for bonus prize awards.