ECE 714 - FALL 2016 - Assignment #6 - Decimation and Interpolation

MATLAB Issues

1. The DFT (or FFT which gives the same result) of a given finite length signal is proportional to the length N of the sequence of the signal that is analyzed, since the DFT sums over 0 to N-1. If you double N, the sum will be over twice as many values and as long as the signal's spectral properties are consistent over time (the signal is *stationary*) each term of the DFT will come out approximately twice as large. As a result, if you want to directly compare the frequency spectra of signals of different length, it is appropriate to normalize the DFT result by dividing by N. In other words, if xx is the signal of interest, rather than plotting the magnitude of the DFT using abs(fft(xx)) , you should plot the *normalized magnitude* of the DFT using abs(fft(xx))/length(xx). For all of the frequency spectra below, plot the *normalized magnitude* as indicated.

Assignment #6

Do the following in a single MATLAB .m file, with one section (cell) for each of the numbered parts of the assignment (see item #15 in the MATLAB information above). Integrate the results in a Word document and save it as a PDF file. Upload your .pdf document and your MATLAB .m file to the corresponding assignment page on Canvas before the date and time indicated on Canvas.

Note: You can make your Word document more compact (and better organized) by creating tables of appropriate dimension in Word and then pasting your MATLAB plots into the individual table elements. Then you can better control how the plots are positioned on the page.

The MATLAB Assignment

1. In this assignment, you will practice down-sampling (decimation) and up-sampling (interpolation) using the data in the waveform audio speech6.wav (available on Canvas with the assignment). You can then read in the file using audioread(). The waveform is speech recorded at 32000 samples per second. Load the waveform into the MATLAB variable named x32. Play the waveform using sound(). Adjust the sound system volume (and tone if available) to get clear sounding speech.

2. Plot the entire signal x32 as a function of time. The time axis should be calibrated in seconds. The magnitude axis should be adjusted to the range -1.2 to 1.2. It is important to use this scaling for direct visual comparison of waveforms throughout this assignment. Label the axes correctly, etc.

3. Plot the normalized magnitude of the estimated frequency spectrum of the signal x32 over the range 0 to fs. Use a linear scale for magnitude (the standard plot() function). The frequency axis should be calibrated in Hz. The magnitude axis should be adjusted to the range 0 to 0.01. It is important to use this scaling for direct visual comparison of frequency spectra throughout this assignment. Label the axes correctly, etc.

4. We are going to down-sample x32(n) by a factor of 4. Create the down-sampled signal x32d8 by selecting every 4th sample of x32: (HINT: "x32d8 = x32(1:4:length(x32))"). Play the waveform using sound(). Remember that the sampling rate for x32d8 is 32000/4 = 8000 samples per second. How does this sound relative to the original audio signal?

5. Plot the entire signal x32d8 as a function of time. The time axis should be calibrated in seconds. The magnitude axis should be adjusted to the range -1.2 to 1.2. It is important to use this scaling for direct visual comparison of waveforms throughout this assignment. Label the axes correctly, etc.

6. Plot the normalized magnitude of the estimated frequency spectrum of the signal x32d8 over the range 0 to fs. Use a linear scale for magnitude (the standard plot() function). The frequency axis should be calibrated in Hz. The magnitude axis should be adjusted to the range 0 to 0.01. It is important to use this scaling for direct visual comparison of frequency spectra throughout this assignment. Label the axes correctly, etc.

7. In step 4 we "forgot" to use an anti-aliasing filter first before down-sampling by 4. We know that to do this properly we need to first filter x32 using a low pass filter with cut-off at *w* = pi/4 (f = fslow/2=4000 Hz), and then form a new sequence at the reduced rate by taking every 4th value of the filtered sequence. A linear phase FIR filter is a good choice here (and in many sampling/reconstruction applications) since we would like to avoid phase distortion of the waveform. The MATLAB "fir1(N, Wn)" function designs FIR filters using the windowed ideal impulse response method, very similar to what you did explicitly in assignment #4. Use the "fir1()" function to design a 65 coefficient (64th order) linear phase low-pass filter with cut-off at *w* = pi/4 (hint: in "fir1(N, Wn)", N is the filter order (64 in this case), and Wn is the desired low pass cut-off frequency as a fraction of the Nyquist frequency (Wn = 1/4 in this case)). Plot the sequence of FIR filter coefficients B in a stem plot.

8. Plot the frequency response of the FIR filter from part 7 using "freqz()".

9. Create the signal x32f by filtering x32 using the FIR filter designed in part 7. You can use the MATLAB "filter()" function for this. Play the waveform using sound().

10. Plot the normalized magnitude of the estimated frequency spectrum of the signal x32f over the range 0 to fs. Use a linear scale for magnitude (the standard plot() function). The frequency axis should be calibrated in Hz. The magnitude axis should be adjusted to the range 0 to 0.01. It is important to use this scaling for direct visual comparison of frequency spectra throughout this assignment. Label the axes correctly, etc.

11. Down-sample x32f by a factor of 4 to create x32fd8 (similar to step 4). Play the waveform again using sound(). Remember that the sampling rate for x32fd8 is 32000/4 = 8000 samples per second. How does this sound relative to the original audio signal?

12. Plot the entire signal x32fd8 as a function of time. The time axis should be calibrated in seconds. Remember that the sampling rate for x32fd8 is 32000/4 = 8000 samples per second. The magnitude axis should be adjusted to the range -1.2 to 1.2. It is important to use this scaling for direct visual comparison of waveforms throughout this assignment. Label the axes correctly, etc.

13. Plot the normalized magnitude of the estimated frequency spectrum of the signal x32fd8 over the range 0 to fs. Remember that the sampling rate for x32fd8 is 32000/4 = 8000 samples per second. Use a linear scale for magnitude (the standard plot() function). The frequency axis should be calibrated in Hz. The magnitude axis should be adjusted to the range 0 to 0.01. It is important to use this scaling for direct visual comparison of frequency spectra throughout this assignment. Label the axes correctly, etc.

14. Up-sample x32fd8 by a factor of 4 to create x32fd8u32 (HINT: " x32fd8u32(4*(1:length(x32fd8))) = x32fd8(1:length(x32fd8));" MATLAB will automatically insert 0's in the undefined vector elements). Play the waveform using sound(). Remember that the sampling rate for x32fd8u32 is now 8000*4 = 32000 samples per second. How does this sound relative to the original audio signal?

15. Plot the entire signal x32fd8u32 as a function of time. The time axis should be calibrated in seconds. Remember that the sampling rate for x32fd8u32 is 8000*4 = 32000 samples per second. The magnitude axis should be adjusted to the range -1.2 to 1.2. It is important to use this scaling for direct visual comparison of waveforms throughout this assignment. Label the axes correctly, etc.

16. Plot the normalized magnitude of the estimated frequency spectrum of the signal x32fd8u32 over the range 0 to fs. Remember that the sampling rate for x32fd8u32 is 8000*4 = 32000 samples per second. Use a linear scale for magnitude (the standard plot() function). The frequency axis should be calibrated in Hz. The magnitude axis should be adjusted to the range 0 to 0.01. It is important to use this scaling for direct visual comparison of frequency spectra throughout this assignment. Label the axes correctly, etc.

17. We can eliminate the unwanted spectral images (caused by up-sampling by 4) by filtering x32fd8u32 with a low pass filter with cut-off at *w* = pi/4 (f = fslow/2=4000 Hz). We already designed such a filter in step 7. Create the signal x32fd8u32f by filtering x32fd8u32 using the FIR filter designed in part 7. You can use the MATLAB "filter()" function for this. *Multiply the filter output by 4 to restore the correct magnitude* (we need a low pass filter with a gain of 4). Play the waveform using sound(). How does this sound relative to the original audio signal?

18. Plot the entire signal x32fd8u32f as a function of time. The time axis should be calibrated in seconds. The magnitude axis should be adjusted to the range -1.2 to 1.2. It is important to use this scaling for direct visual comparison of waveforms throughout this assignment. Label the axes correctly, etc.

19. Plot the normalized magnitude of the estimated frequency spectrum of the signal x32fd8u32f over the range 0 to fs. Use a linear scale for magnitude (the standard plot() function). The frequency axis should be calibrated in Hz. The magnitude axis should be adjusted to the range 0 to 0.01. It is important to use this scaling for direct visual comparison of frequency spectra throughout this assignment. Label the axes correctly, etc.

20. When you are done, you will have the signals:

- x32 - the original speech signal at 32000 samples per second
- x32d8 - the speech signal down-sampled to 8000 samples per second without filtering (incorrect)
- x32f - the speech signal filtered with an anti-aliasing filter before down-sampling (correct)
- x32fd8 - the speech signal filtered with an anti-aliasing filter and then down-sampled to 8000 samples per second (correct)
- x32fd8u32 - the speech signal up-sampled back to 32000 samples per second with no filtering (correct but not complete)
- x32fd8u32f - the speech signal up-sampled back to 32000 samples per second filtered with an anti-aliasing filter (correct)

You can then listen to these audio signals again from the MATLAB command line using the sound() function. Listen to and compare the audio quality of the different waveforms to help in preparing your comments for the previous parts of the assignment. You don't have to make any special comments here.