New Mexico State University
Klipsch School of Electrical Engineering

EE395 - Introduction to Digital Signal Processing

Fall 2010
Final Exam Part 1

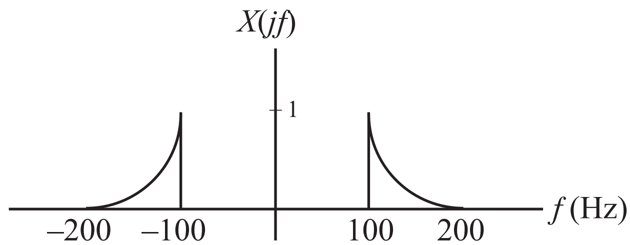
Name: ____________________

Solve any three Problems.
Circle below which three problems you want graded.

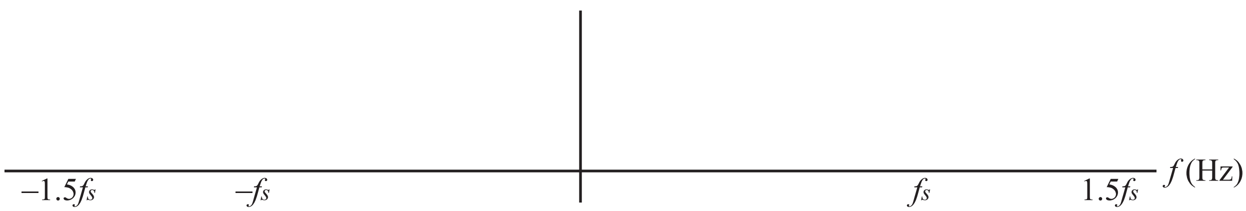
Prob. 1	/ 16.7 points
Prob. 2	/ 16.7 points
Prob. 3	/ 16.7 points
Prob. 4	/ 16.7 points
Prob. 5	/ 16.7 points
Total	/ 50 points

Prob. 1

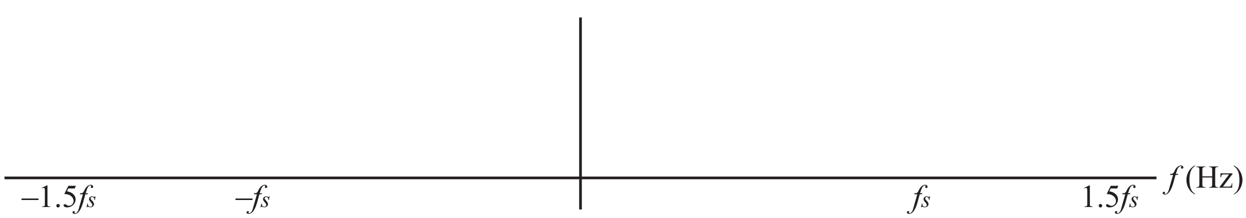
Let $x(t)$ be a continuous-time signal with a spectrum $X(jf)$ shown below. Suppose $x(t)$ is sampled at a rate f_s samples/s. For the given f_s, sketch the spectrum of the sampled signal for $-1.5f_s \leq f \leq 1.5f_s$. Be sure to label important edge frequencies.



(a) $f_s = 150$. Will there be aliasing (overlap in spectral replicas) (YES / NO)?

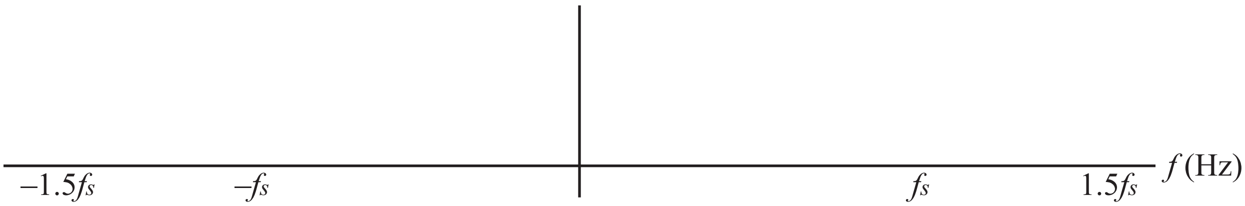


(b) $f_s = 250$. Will there be aliasing (overlap in spectral replicas) (YES / NO)?

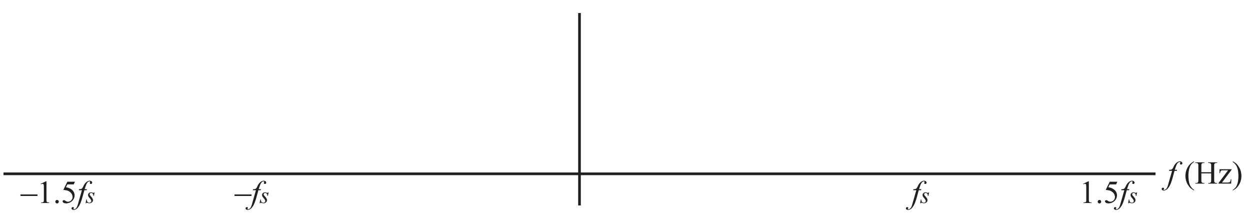


Prob. 1 (cont.)

(c) $f_s = 350$. Will there be aliasing (overlap in spectral replicas) (YES / NO)?



(d) $f_s = 450$. Will there be aliasing (overlap in spectral replicas) (YES / NO)?



Prob. 2

The linear, constant-coefficient difference equation (LCCDE) of a digital filter is given by

$$y(n) \quad = \quad -\frac{1}{4}y(n-1) + \frac{1}{8}y(n-2) + x(n). \tag{2.1}$$

Assume the filter is *causal* and at-rest, i.e. $y(-1) = y(-2) = 0$.

(a) Draw the Direct-Form I structure of the digital filter.

(b) Draw the Direct-Form II structure of the digital filter.

(c) Determine the transfer function $H(z)$ and Region of Convergence (ROC).

Prob. 2 (cont.)

(d) Is the filter *stable*? (YES / NO)

(e) Determine the impulse response $h(n)$ of the filter by inverse-transforming $H(z)$. Your answer should be in a closed-form expression giving all values from $-\infty \leq n \leq \infty$.

Prob. 3

Let $x_1(n) = e^{j2\pi nk/N}$ and $x_2(n) = \cos(2\pi nk/N)$ where $0 \leq n \leq N-1$ and k is an integer that determines the frequency of the signal. Assume $0 \leq k \leq N-1$. Let $X_1(m),\ X_2(m)$ be the N-point DFT of $x_1(n),\ x_2(n)$, respectively.

(a) Use the DFT equation to show that $X_1(k) = N$.

(b) Use the DFT equation to show that $X_1(m) = 0$ for $m \neq k$.

(c) For $k = 3$ and $N = 8$, sketch $X_1(m)$ for $0 \leq m \leq 7$.

Prob. 3 (cont.)

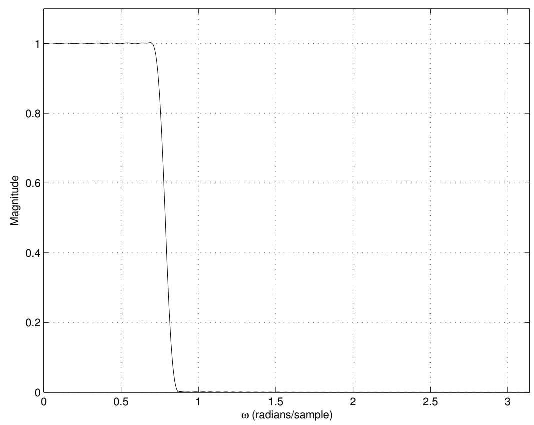
(d) Use the DFT equation to show that $X_2(k) = N/2$.

(e) Use the DFT equation to show that $X_2(m) = 0$ for $m \neq k$.

(f) For $k = 3$ and $N = 8$, sketch $X_2(m)$ for $0 \leq m \leq 7$.

Prob. 4

Let $h(n)$ be the real coefficients of a length N (odd), lowpass, FIR filter with cutoff frequency $\omega_c = \pi/4$ rads/sample that has the magnitude response below.



Suppose we create a new filter through multiplication with a cosine wave, i.e. modulation

$$g(n) = h(n)\cos(\omega_0 n)\,. \tag{4.1}$$

For each of the following shift frequencies ω_0, carefully sketch the magnitude response of $g(n)$ and note the passband gain.

(a) $\omega_0 = \pi$

Prob. 4 (cont.)

(b) $\omega_0 = \pi/4$

(c) $\omega_0 = 3\pi/4$

(d) $\omega_0 = \pi/8$

Prob. 5

The signal $x_{\text{old}}(n)$ was sampled at $f_s = 10,000$ samples/s but should have been sampled at $f_s = 12,000$ samples/s. The following MATLAB code implements the sample rate converter shown in Figure 10-7(b).

```
x_up = upsamp(x_old,M); % upsample
h = fir_wind(64,wc); % design the filter with length N and cutoff wc rads/sample
x_interp = filter(h,1,x_up); % filter
x_new = downsamp(x_interp,D); % downsample
```

Determine (and explain if necessary) your choices for M, D, and ω_c (rads/sample) . In order to receive full credit, your choices for M and D should be the minimum allowable values. You may assume that the filter has enough coefficients for the given signal.

New Mexico State University
Klipsch School of Electrical Engineering

EE395 - Introduction to Digital Signal Processing

Fall 2010
Final Exam Part 2

(Solution due in GA160G 5:00pm Thu., Dec. 9, 2010)

"The attached solution is due entirely to my own, individual efforts. I have not discussed this project with any other student nor have I consulted with anyone other than (possibly) the instructor of this course in creating these solutions."

Signature: ______________________________ Date: ____________________

Prob. 1	/ 15 points
Prob. 2	/ 10 points
Prob. 3	/ 10 points
Prob. 4	/ 15 points
Total	/ 50 points

Directions

For each problem, please write a MATLAB code which produces the required plots. Name your main codes prob1.m, prob2.m, prob3.m, and prob4.m. Please submit the *signed* cover sheet of this exam and a ***hardcopy of your solutions (plots and main codes)***. In addition, please send (before the due date) an email to `pdeleon@nmsu.edu` with a .zip file archive containing the two main codes and a directory called "DSP_toolkit" containing your tools. You will receive an email confirmation upon receipt of the .zip file.

Assistance

- You are free to email any questions to Prof. De Leon during the project period. Email may include a request to examine code.
- Office hours for the week of Dec. 6–10 are on a walk-in basis. However, if you wish to schedule an appointment please email `pdeleon@nmsu.edu` your request.

Prob. 1

In this problem, we will simulate the use of oversampling to reduce A/D quantization noise (we simulate because we do not have easy access to analog signals, A/D converters with adjustable resolution, and appropriate test and measurement equipment). In particular we will verify the statement that "...oversampling by a factor of 4 (and filtering), we gain a single bit's worth of quantization noise reduction." For this question, please use requantize.m (from Homework #9) and resample.m (from MATLAB's Signal Processing Toolbox).

Code the following simulation steps

1. Construct a reference 440 Hz tone x_{ref} (use cosingen.m) with $f_s = 1000$ Hz, $B = 4$ bits resolution (use requantize.m), and 100 ms duration.
2. Construct three 440 Hz tones x_1, x_2, x_3 with $f_{s,1} = 4000$, $f_{s,2} = 16000$, $f_{s,3} = 64000$ Hz and $B_1 = 3$, $B_2 = 2$, $B_3 = 1$ bits resolution, respectively.
3. Resample x_1, x_2, x_3 to $f_s = 1000$.
4. Normalize all signals, i.e. $x = x./\max(\text{abs}(x))$ so that any scaling effects from filtering are (roughly) eliminated.

(a) Individually plot (use plotcsig.m) x_{ref} and x_1, x_2, x_3 after Steps 2 and 3. Be sure to point out the 2^B levels in each plot after Step 2. Comment on the quantization noise present after Step 2 and the accuracy compared to the reference after Step 3.

(b) Measure the maximum error for each of the three tones defined as

$$e_{\max,l} \quad = \quad \max |x_{\text{ref}} - x_l|, \;\; l = 1, 2, 3. \tag{1.1}$$

You should trim input-on/input-off filter transients from x_l before computing the error.

(c) Describe what this simulation tells you and in particular whether the statement in the first paragraph is true.

Prob. 2

In this problem, we will examine amplitude modulation for music synthesis and use the Hilbert transform to recover the envelope from the signal.

(a) Code the adsr_gen.m tool and do both examples in the *DSP Software Toolkit*. Plot the ADSR envelope and modulated 440 Hz tone. Comment on why amplitude modulation of a tone might be a reasonable start toward synthesis of a musical instrument.

(b) Design a length 129 Hilbert filter. Plot the magnitude and phase responses of the filter.

(c) Let $x_r(n)$ be the modulated tone from (a). Compute $x_c(n)$ as in Figure 9-12 using your length 129 Hilbert filter from (b) and a $(129 - 1)/2 = 64$ sample delay. Estimate and plot the envelope with $E = |x_c(n)|$. Compare to (a) and comment.

Prob. 3

Let $x(n)$, $0 \leq n \leq 1023$ be a 440 Hz tone at $f_s = 1000$ Hz. Let $X(m)$ be the FFT of $x(n)$.

(a) Compute the inverse of $X(m)$ using Method 1 shown in Figure 13-14. Call this signal $x_1(n)$. Measure the mean-squared error (MSE) for Method 1 defined as

$$e_1 = \frac{1}{1024} \sum_{n=0}^{1023} [x(n) - x_1(n)]^2. \tag{3.1}$$

(b) Compute the inverse of $X(m)$ using Method 2 shown in Figure 13-15. Measure the MSE for Method 2.

Prob. 4

In this problem, we will verify the zero-phase filtering technique on p. 519 Figure 13-31(a). For time reversal, you may wish to use flipud.m (flip up/down).

(a) Design a 2nd order Butterworth lowpass filter with cutoff frequency $\omega_c = 3\pi/4$ rads/sample. Plot the magnitude and phase responses (the independent variable is ω rads/sample). Determine the magnitude and phase response values at $\omega_0 = 1$ rad/sample using the dtft.m tool.

(b) *Phase delay* in samples is computed as $(2\pi/\omega)\phi$ where ϕ is the phase angle. Determine the phase delay in samples of the filter at $\omega_0 = 1$ rad/sample.

(c) Generate a 100 sample tone (use cosingen.m) with frequency $\omega_0 = 1$ rad/sample, i.e. let $f_s = 2\pi$ and $f = 1$. In Figure 13-31(a), let $x(n)$ be the tone and the IIR filter be the filter from (a). Plot $x(n)$, the signal at node A, and $y(n)$. Also list the sample values of each signal in the range $45 \leq n \leq 55$.

(d) Use the plots and values of $x(n)$ and the signal at node A to verify (roughly) the expected phase delay value from (a). Use the plots of $x(n)$ and $y(n)$ to verify the zero-phase filtering. Note that amplitude differences should be consistent with the magnitude response while delays should be consistent with the phase response/phase delay.