ECE4703 Midterm Exam

Your Name: SOLUTION Your box #: ____________

November 20, 2008

Tips:

- Look over all of the questions before starting.
- Budget your time to allow yourself enough time to work on each question.
- Write neatly and show your work!
- This exam is worth a total of 200 points.
- Attach your "cheat sheet" to the exam when you hand it in.

problem 1	problem 2	problem 3	problem 4	problem 5	total midterm exam score
20 points	20 points	40 points	70 points	50 points	200 points

1. 20 points. Suppose you have the analog signal

$$x(t) = a\cos(2\pi t)\cos(5\pi t)$$

defined on $-\infty < t < \infty$ with $a > 0$.

(a) 10 points. What is the minimum sampling frequency (in Hz) needed to sample this signal without aliasing?

$\cos x \cdot \cos y = \frac{1}{2}\cos(x-y) + \frac{1}{2}\cos(x+y)$

so $x(t) = \frac{a}{2}\cos(3\pi t) + \frac{a}{2}\cos(7\pi t)$

max frequency is 7π rad/sec $\Rightarrow$ 3.5 Hz

Hence minimum sampling frequency is 7Hz.

(b) 10 points. Suppose this signal is sampled by an ADC with a full-scale input voltage of $6.0V_{pp}$ (like the AIC23 codec you have been using in the lab assignments). What value should a be to achieve 90% full-scale range of the analog signal at the input of the ADC?

90% full-scale range is $0.9 \times 6.0V_{pp} = 5.4V_{pp}$

Hence $a = 2.7$.

2. 20 points. Suppose you run the following C code on the TMS320C6713:

```
short u,v,w;
u = 20000;
v = u*2;
w = v*2;
```

What are the values of v and w after this code runs?

$u * 2 = 40000$ which is too large for the short datatype.
The C6713 will wrap around: $v = 40000 - 2^{16} = -25536$.

$v * 2 = -51072$, also too large for the short data type.
The C6713 will wrap around: $w = -51072 + 2^{16} = 14464$

3. 40 points. After you graduate from WPI, you find yourself working for a company designing real-time DSP algorithms. Your boss asks you to design a bandpass filter with a particular frequency response. You design a floating-point FIR filter and confirm in `fdatool` that it will achieve the desired frequency response. When you implement it on the DSP, however, you discover that the filter is not running in real-time. List as many things that you can think of that you might be able to do to fix this problem.

Some possibilities:

- convert filter to fixed point
- convert filter to IIR (usually less calculations)
- try optimizing code (optimizing compiler or hand optimization)
- decrease the sampling frequency
- increase the clock rate of the DSP (buy a faster DSP)

4. 70 points total. Suppose you are given a linear time invariant IIR filter with the transfer function

$$\begin{aligned} H(z) &= \frac{Y(z)}{X(z)} \\ &= \frac{0.6(1-z^{-1})^2(1-2z^{-1})^2}{(1-(0.7+0.7j)z^{-1})(1-(0.7-0.7j)z^{-1})(1-(0.4+0.9j)z^{-1})(1-(0.4-0.9j)z^{-1})} \\ &= \frac{0.6-3.6z^{-1}+7.8z^{-2}-7.2z^{-3}+2.4z^{-4}}{1-2.2z^{-1}+3.07z^{-2}-2.142z^{-3}+0.9506z^{-4}} \end{aligned}$$

Note that $j = \sqrt{-1}$.

(a) 10 points. Is $H(z)$ stable? Why or why not?

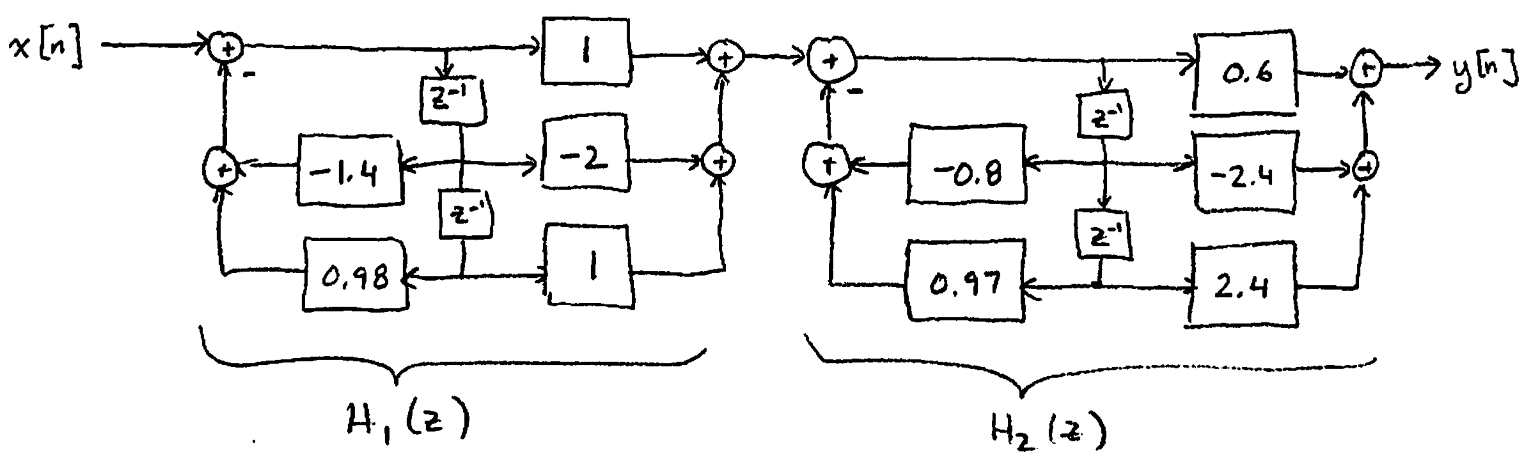
yes: magnitude of poles at $0.7 \pm 0.7j$ is ~ 0.99
magnitude of poles at $0.4 \pm 0.9j$ is ~ 0.985
magnitudes of all poles strictly less than 1
$\Rightarrow$ stable.

Continued...

(b) 40 points. Draw a "Direct Form II – Second Order Sections" realization of $H(z)$ assuming infinite precision filter coefficients (each second order section should have *real-valued* coefficients). Draw neatly and label everything accurately for full credit.

$$H_1(z) = \frac{1 - 2z^{-1} + z^{-2}}{1 - 1.4z^{-1} + 0.98z^{-2}}$$

$$H_2(z) = \frac{0.6(1 - 4z^{-1} + 4z^{-2})}{1 - 0.8z^{-1} + 0.97z^{-2}} = \frac{0.6 - 2.4z^{-1} + 2.4z^{-2}}{1 - 0.8z^{-1} + 0.97z^{-2}}$$



(c) 20 points. Suppose your Direct Form II – Second Order Sections filter is to be implemented as a fixed-point filter with all filter coefficients stored as 16-bit short signed integers. How many fractional bits should you specify for your filter coefficients to minimize the quantization error?

Largest coefficient is in $H_2(z)$: ± 2.4. Need two non-fractional bits to represent these coefficients without overflow. This would leave 13 fractional bits.

4. 50 points. Without using any math, discuss how fixed-point FIR and IIR filtering differs from floating-point FIR and IIR filtering. What are the basic principles of fixed-point filtering that should be followed to ensure good performance?

The main differences are that

i) you have to worry about overflow in fixed-point filters (less dynamic range)

ii) you have to approximate the actual filter by quantizing the coefficients

iii) you have to throw away precision during your processing

The basic principle of fixed-point filtering is much like the basic principle of getting the maximum SNR from your ADC. You want to use the full range of the datatype in each calculation (filter coefficients, results of products, results of sums, ...) but you must avoid overflow. If you are too conservative, i.e. avoiding overflow but throwing away too much precision, your filter response will be inaccurate. Your fixed-point filter design achieves the best performance when the calculations are close to causing overflow, but never actually do overflow.