The University of Texas at Austin
Dept. of Electrical and Computer Engineering
Midterm #1

Date: March 11, 2016 Course: EE 445S Evans

Name: The Little Prince
Last, First

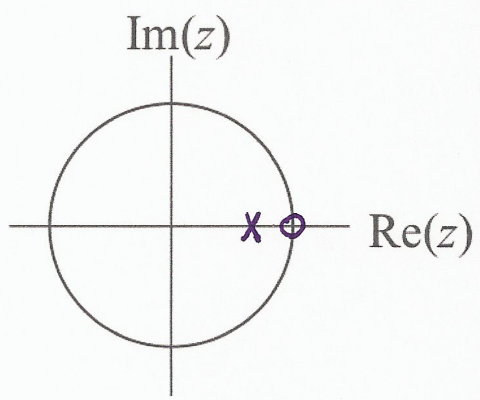
- The exam is scheduled to last 50 minutes.
- Open books and open notes. You may refer to your homework assignments and the homework solution sets.
- Calculators are allowed.
- You may use any standalone computer system, i.e. one that is not connected to a network. ***Please disable all wireless connections on your computer system(s).***
- Please turn off all cell phones.
- No headphones allowed.
- All work should be performed on the quiz itself. If more space is needed, then use the backs of the pages.
- **Fully justify your answers.** If you decide to quote text from a source, please give the quote, page number and source citation.

Problem	*Point Value*	*Your score*	*Topic*
1	28		Filter Design & Analysis
2	24		BIBO Stability
3	24		Upsampling
4	24		Potpourri
Total	100		

Problem 1.1 *Filter Design & Analysis.* 28 points.

Design and analyze a first-order discrete-time infinite impulse response filter to remove DC, which is a discrete-time frequency of 0 rad/sample.

(a) Place the pole and zero on the pole-zero diagram below. *4 points* HW 3.1(c)



IIR notch filter to remove DC and keep as much of the other frequencies as possible.

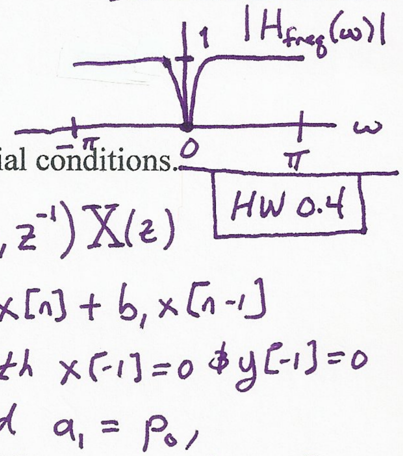
(b) Give numeric values of the pole and zero in part (a). Why did you choose these values? *4 points* HW 3.1(c)

Pole is at $p_0 = 0.9$. Zero is at $z_0 = 1$.
A zero at $z_0 = 1$ will zero out the frequency at $\omega_0 = 0$ rad/sample because $z_0 = e^{j\omega_0} = e^{j0} = 1$. A pole at the same angle will pass as much of the other frequencies as possible.

(c) Give a formula for the discrete-time frequency response and draw the magnitude response. *6 points.* HW 2.1

$$H(z) = C \, \frac{1 - z_0 z^{-1}}{1 - p_0 z^{-1}}.$$ With $|p_0| < 1$, system is bounded-input bounded-output stable.

$$H_{freq}(\omega) = C \, \frac{1 - z_0 e^{-j\omega}}{1 - p_0 e^{-j\omega}}.$$ Normalize frequency response at $\omega = \pi$: $C = \frac{1.9}{2}$.



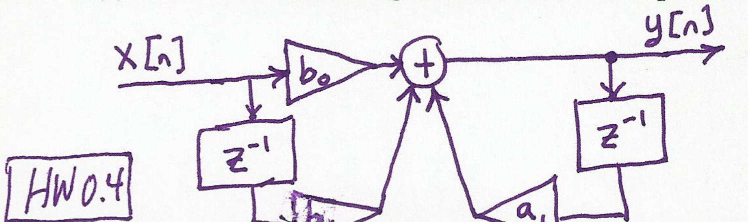
(d) Give the difference equation relating output $y[n]$ and input $x[n]$ including the initial conditions. *4 points* HW 0.4

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1}}{1 - a_1 z^{-1}} \Rightarrow (1 - a_1 z^{-1}) Y(z) = (b_0 + b_1 z^{-1}) X(z)$$

$$y[n] = a_1 y[n-1] + b_0 x[n] + b_1 x[n-1]$$

with $x[-1] = 0$ & $y[-1] = 0$ and $a_1 = p_0$, $b_0 = C$, $b_1 = -C z_0$

(e) Draw the block diagram for the filter. *4 points.* HW 0.4



(f) Why is removing the DC offset (average value) important in speech and audio systems? *6 points.* HW 1.3

When speech/audio is sampled, signal has little or no DC value. Humans cannot hear DC — normal hearing is 20 Hz to 20 kHz. Content from 0 Hz to 20 Hz would be noise/interference. (1) Removing DC offset improves signal-to-noise ratio; (2) For playback, speech/audio goes through D/A converter and needs to be converted to an integer format — DC offset uses many bits. DC removal improves quality; and (3) DC removal prevents clipping of amplitude when going through D/A conversion.

Problem 1.2 *Stability.* 24 points.

For a discrete-time linear time-invariant system with impulse response $h[n]$, the system is bounded-input bounded-output (BIBO) stable if and only if

$$\sum_{n=-\infty}^{\infty} |h[n]| < \infty$$

HW 2.1

(a) Using the above definition, prove that a discrete-time finite impulse response (FIR) filter is always BIBO stable. *12 points.*

An FIR filter has a finite number of non-zero coefficients.

$$\sum_{n=0}^{N-1} |h[n]| \leq N B < \infty$$

where $|h[n]| \leq B < \infty$ and N is number of FIR coefficients.

In other words, a finite sum of finite values must be finite.

HW 0.2

HW 0.4

HW 1.1

HW 2.1

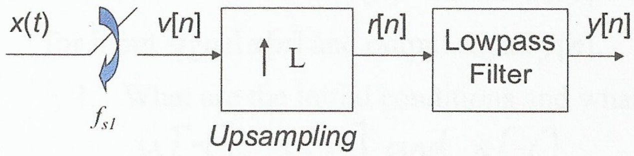
(b) Give an example of a BIBO unstable linear time-invariant system, an application that uses the BIBO unstable system, and how the application uses the BIBO unstable system. *12 points.*

Example #1: Sinusoidal generator (problem 1.4 on this midterm). Sinusoidal amplitude modulation multiplies a baseband signal by a sinusoidal signal to shift the center frequency of the baseband signal to the (carrier) frequency of the sinusoid.

Example #2: Discrete-time integrator – $y[n] = y[n-1] + x[n]$ with $y[-1] = 0$. Can be used to keep a running sum of temperature or signal power (i.e. signal amplitude squared). Running sum of signal power is total signal energy. Can also be used to create audio effects.

Problem 1.3 *Upsampling.* 24 points.

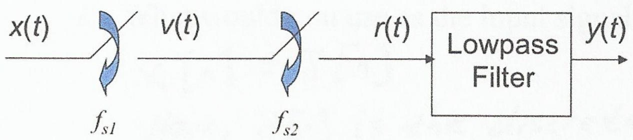
Consider the following block diagram to sample an audio signal $x(t)$ at f_{s1} = 32 kHz to produce a discrete-time signal $v[n]$ and then change the sampling rate to f_{s2} = 96 kHz via the discrete-time operation of upsampling by 3 (i.e. $L = 3$) followed by a discrete-time lowpass filter to produce $y[n]$.

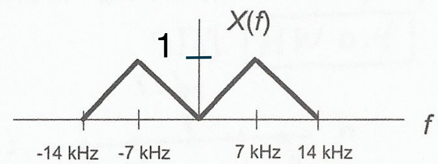


Note: Upsampler does not change amplitude.

Note: Continuous-time equivalent to upsampler should have been lowpass filter (fstop = 1/2 fs1) followed by sampling at fs2.

The continuous-time equivalent to the above block diagram is

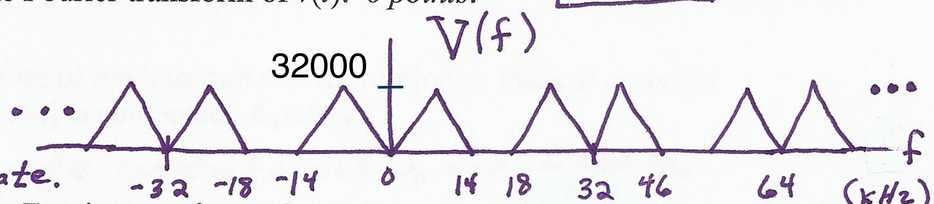




Using $X(f)$ given above to the right, which is the continuous-time Fourier transform of $x(t)$, please complete the following analysis:

(a) Draw $V(f)$, which is the continuous-time Fourier transform of $v(t)$. *6 points.* $\boxed{\text{HW 0.3}}$ $\boxed{\text{HW 2.2}}$

Sampling causes a periodic replication of the input spectrum at offsets that are multiples of the sampling rate.



(b) Draw $R(f)$, which is the continuous-time Fourier transform of $r(t)$. *6 points.* $\boxed{\text{HW 0.3}}$

R(f) will contain V(f) plus versions of V(f) shifted left and right by multiples of 96 kHz. R(f) will be equal to V(f) except that the amplitude will be scaled by 96000.

(c) Give the passband and stopband frequencies in Hz for the continuous-time lowpass filter to use to recover $x(t)$ from $r(t)$. *4 points.* $\boxed{\text{HW 0.2}}$ $\boxed{\text{HW 2.2}}$

f_{pass} = 14 KHz

f_{stop} = 16 KHz ← or 15.4 KHz to hit 10% rolloff exactly.

(d) Give the passband and stopband frequencies in rad/sample for the discrete-time lowpass filter in the upper block diagram to recover $v[n]$ from $r[n]$. *4 points.* $\boxed{\text{HW 3.1}}$

$\omega_{pass} = 2\pi \frac{14 \text{ KHz}}{96 \text{ KHz}} = 2\pi \frac{f_{pass}}{f_{s2}}$

$\omega_{stop} = 2\pi \frac{16 \text{ KHz}}{96 \text{ KHz}} = 2\pi \frac{f_{stop}}{f_{s2}}$

(e) For the discrete-time lowpass filter, would you advocate to use a finite impulse response (FIR) filter or an infinite impulse response (IIR) filter? Why? *4 points.* $\boxed{\text{HW 3.3(d)}}$

For audio processing, both linear phase and low group delay (~10 ms) matter.

FIR filter: linear phase over all frequencies but high group delay.

IIR filter: near linear phase in passbands and low group delay.

Advocate IIR filter design. See Fall 2015 Midterm #1 Problem 1.4(b).

Problem 1.4. *Potpourri.* 24 points.

(a) In lab #2, you implemented a cosine generator on the digital signal processing board in lab using a causal linear time-invariant filter with the difference equation

$$y[n] = (2 \cos \omega_0)\, y[n-1] - y[n-2] + x[n] - (\cos \omega_0)\, x[n-1]$$

for input signal $x[n]$ and output signal $y[n]$.

1. What are the initial conditions and what should their values be? *3 points.* HW 0.4

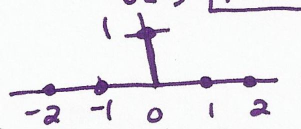
   $y[-1]$, $y[-2]$ and $x[-1]$.

   All should be set to zero to satisfy linear time-invariant properties.

2. What would you use as the input signal $x[n]$? *3 points.* $\delta[n]$ HW 0.4

   $x[n] = \delta[n]$

   Here, $\delta[n]$ is the discrete-time impulse:



3. Give the output signal $y[n]$ that is a solution to the above difference equation for the input signal $x[n]$ given in part 2 above? *3 points.* HW 0.4

   $y[n] = \cos(\omega_0 n)\, u[n]$

4. Describe an efficient implementation of the interrupt service routine so that $\cos \omega_0$ is not computed every time a sample of $y[n]$ is computed. *6 points.*

   Approach #1: Pre-compute constants $\cos \omega_0$ and $2 \cos \omega_0$.

   Approach #2: Compute $\cos \omega_0$ and $2 \cos \omega_0$ on first ISR call only.

   Approach #3: Use a DMA approach to compute a buffer of values.

   Approach #4: Use a lookup table to pre-compute 1 period of cosine.

(b) In lab #3, you implemented discrete-time infinite impulse response (IIR) filters in 32-bit IEEE floating-point data and coefficients on the digital signal processor board. What distortion could the **continuous-time output signal** of the digital-to-analog converter have if you were to implement the discrete-time IIR filter as a cascade of biquads but not implement the gain for each biquad? HW 3.3
9 points.

The IIR filter output is in 32-bit IEEE floating point, which needs to be converted to a 16-bit short integer (signed) to send to the digital-to-analog (D/A) converter.

The conversion to 16-bit short integer assumes that the floating-point number is in the range of $[-1, 1]$, e.g.

short16value = (short int)(float32value * 32768);

Case #1: Biquad gain product < 1. Floating point values will go out of range $[-1, 1]$ and clipping will occur.

Case #2: Biquad gain product > 1. Floating point values will be small compared to 1 in absolute value. Loss of quality/range.