ECE 3210 Final Exam

Week of: December 7, 2020

Instructor:
Eric Gibbons
ericgibbons@weber.edu
801-626-6861

You have 15 hours for 6 problems.

- Show enough (neat) work in the clear spaces on this exam to convince us that you derived, not guessed, your answers.
- Put your final answers in the boxes at the bottom of the page.
- Please submit all of your relevant Python code.
- Please scan your work as a single, legible PDF.

This test is open notes, open book, and open calculator/Python. Consulting with any third party is considered cheating.

Problem	Score	Possible Points
1		25
2		20
3		20
4		20
5		20
6		20
Total score		125

1 Short answer

(a) Determine if the following systems that take some input $x(t)$ and return some $y(t)$ are linear and/or time-invariant.

(a) $y[k] = \log(x[k])$

Circle one:	linear	nonlinear
Circle one:	time-invariant	time-variant

(b) $y(t) = x(2t)$

Circle one:	linear	nonlinear
Circle one:	time-invariant	time-variant

(b) Determine if the following systems are stable

(a) $H(s) = \frac{s^3+5s^2+s}{s^5-6s^4-2s^3+96s+136}$

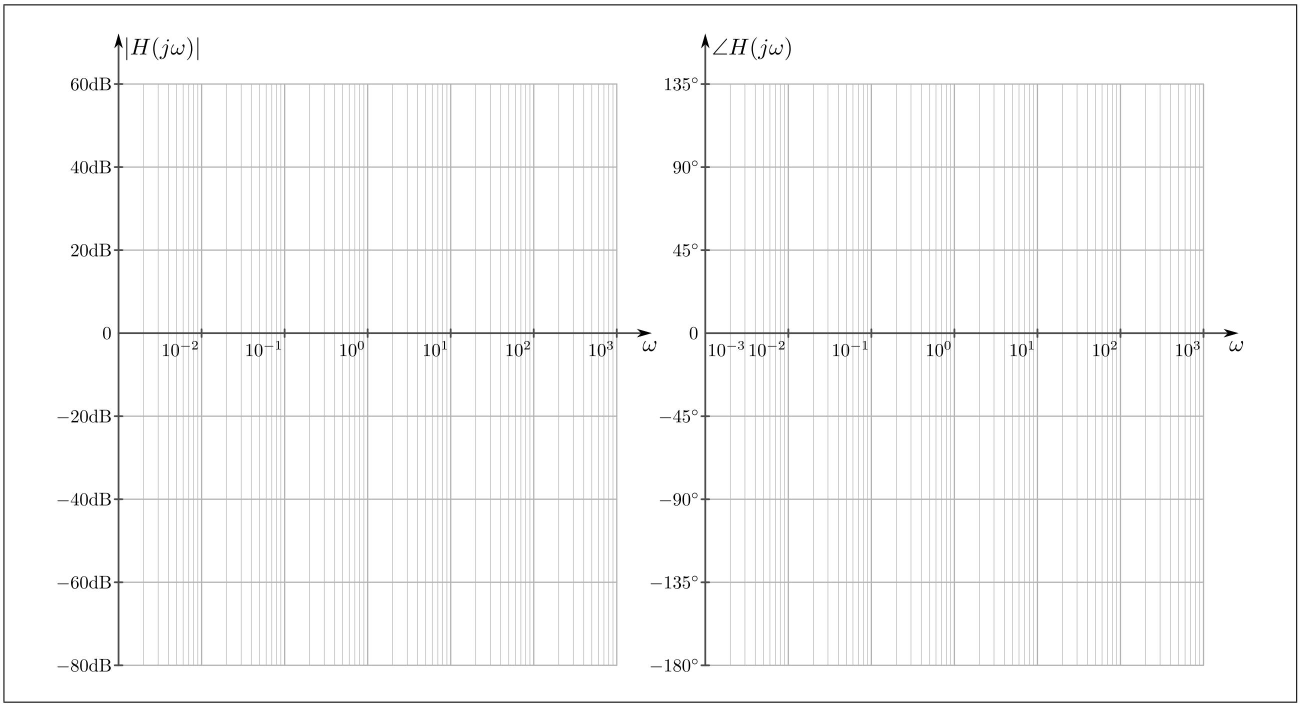
Circle one:	stable	marginally stable	unstable

(b) $H[z] = \frac{1}{z^4+1.28z^2+0.4096}$

Circle one:	stable	marginally stable	unstable

(c) Sketch the Bode plot given the following transfer function

$$H(s) = 100\frac{s+1}{s^2+110s+1000}.$$



2 Fourier series

For each signal, find its fundamental frequency ω_0, and the non-zero Fourier coefficients of the Fourier series. For example, if

$$f(t) = 1 + \cos(2\pi t)$$

the answer would be $\omega_0 = 2\pi$ and $D_0 = 1$, $D_{-1} = D_1 = \frac{1}{2}$.

(a) $f(t) = \sin(2\pi t) + \cos(4\pi t)$

$\omega_0 =$	The D_n's are:

(b) $f(t) = \cos(2\pi t) + \cos(3\pi t)$

$\omega_0 =$	The D_n's are:

(c) $f(t) = \cos(2\pi t)\cos(3\pi t)$

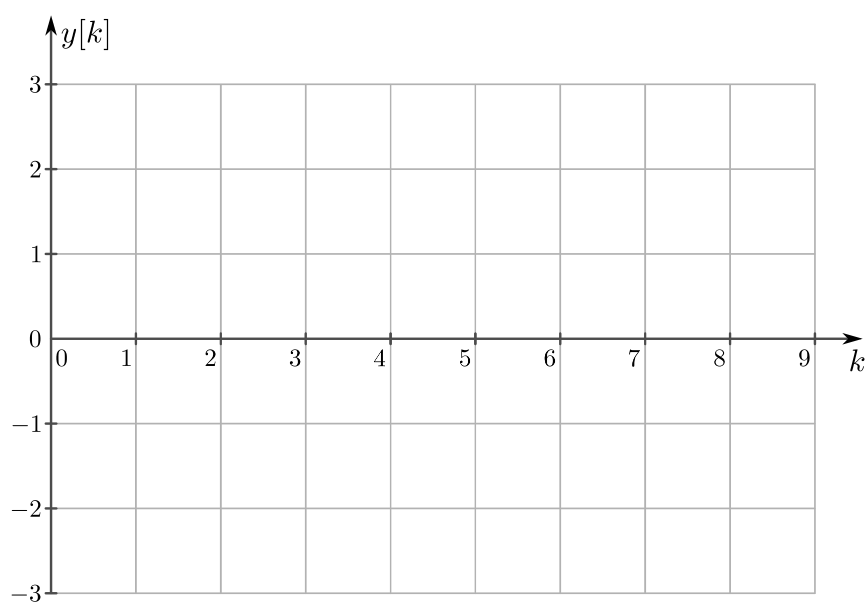
$\omega_0 =$	The D_n's are:

(d) $f(t) = \cos(2t) + \cos(3\pi t)$

$\omega_0 =$	The D_n's are:

3 Convolution

(a) Sketch the convolution $\cos(\pi k)u[k] * u[k]$ for time points $k \in [0, \quad 9]$.



(b) Find the convolution $y(t) = (u(t-1) - u(t-3)) * e^{-3t}u(t)$. You are MUST use the integration approach (i.e., you can't use the Laplace Transform—though you might want to check your work using it)

$y(t) =$

4 Integration and summation

(a) Evaluate the integral

$$\int_{-\infty}^{\infty} \mathrm{sinc}(t)dt.$$

You must show sufficient work to justify your answer.

$\int_{-\infty}^{\infty} \mathrm{sinc}(t)dt =$

(b) Evaluate the summation

$$\sum_{n=-\infty}^{\infty} \operatorname{sinc}^2\left(\frac{n}{2}\right)\cos(\pi n).$$

You must show sufficient work to justify your answer.

$\sum_{n=-\infty}^{\infty} \operatorname{sinc}^2\left(\frac{n}{2}\right)\cos(\pi n) =$

5 Filter design

Suppose we are interested in designing a *high-pass* filter. We will want this filter to be build with the following specifications.

- Passband gain to be above -2dB ($\hat{G}_p = -2$dB) for $f \in [1.5\text{kHz}, \quad \infty)$
- Stopband gain not to exceed $\hat{G}_s = -20$dB for $f \in [0\text{Hz}, \quad 500\text{Hz}]$

You are welcome (indeed, you are encouraged) to use Scipy for your design. You might find the `scipy.signal.butter( )` and `scipy.signal.buttord( )` functions to be helpful. The usage should be pretty similar to what we did in Lab 8. Please include all of your relevant Python code in the script `filter_design.py`

(a) What is the filter order n?

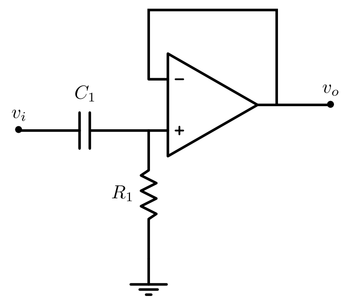
$n =$

(b) Determine the transfer function $H(s)$ as a cascade of 2$^{\text{nd}}$ and 1$^{\text{st}}$ transfer functions. You can use a mixture of Python and/or pencil-and-paper work here.

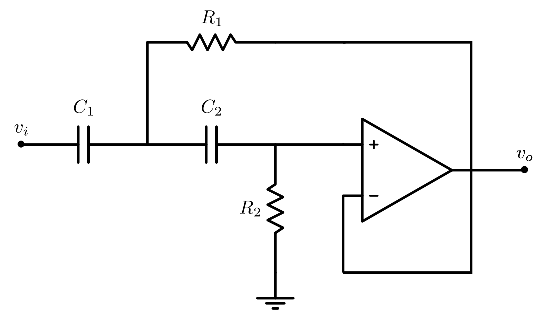
$H(s) =$

(c) Now suppose we want to build this circuit in hardware. We go to the ET 133-C lab and find that we have two op amps, a lot of resistors of any size you need, and a bunch of 500nF capacitors. Draw an appropriate filter that matches the transfer function you designed earlier.

Hint: It will be easiest to incorporate the following circuit topologies for your stages. You might need to use multiple filter stages cascaded together.



First order high pass filter



Second order high pass filter

Your work, continued...

6 Difference equation

Consider the following difference equation

$y[k+2]-y[k+1]-y[k]=f[k+1]+f[k]$.

It's initial conditions are $y[0]=1$ and $y[1]=1$. The input is $f[k]=\cos(\pi k)u[k]$.

(a) Using the Z-transform, please solve for the zero-input solution $y_{\text{zi}}[k]$.

$y_{\text{zi}}[k]=$

(b) Using the Z-transform, please solve for the zero-state solution $y_{\text{zs}}[k]$.

$y_{\text{zs}}[k] =$

(c) Using the previous two parts, please solve for the total solution $y[k]$.

$y[k] =$

(d) Using Python, write a script called `difference_equation.py`. This script should plot the *numerical* solution of this difference equation on a stem plot for $k \in [0,9]$. Also, plot your closed form solution for $y[k]$ in the previous part as stem plot for $k \in [0,9]$. (For your own sanity check, make sure both plots are the same.)

(e) This is a difference equation that yields a famous sequence of integers. Looking at the first few values of $y[k]$, what is this sequence?