Belmont University, Curb College of Entertainment and Music Business

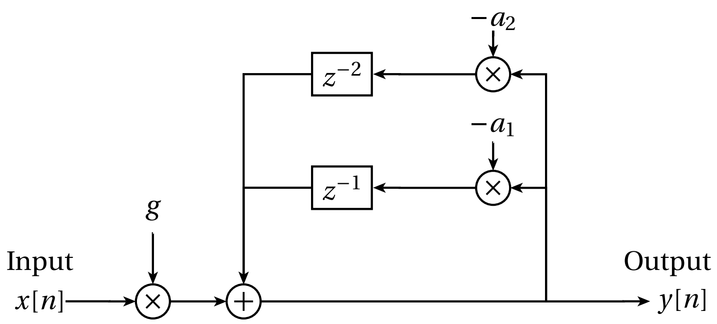
AET 5420 Final Exam

Name : ____________________________

April 19, 2021

1 System Design and Analysis

Given the following system in Fig. 1.1, analyze its performance with the frequency response, pole-zero plot, and step response.



$g = 1,\ a_1 = -1.4142,\ a_2 = 1$

Figure 1.1: Block Diagram of Second-Order System

Write the difference equation for this block diagram:

Rearrange the difference equation so all terms related to y are on the left-hand side and all terms related to x are on the right-hand side:

Take the $\mathbb{Z}$ - transform:

Re-write the result in the form of a transfer function $\frac{Y[z]}{X[z]} = H[z]$:

Draw the magnitude response for this system (use Matlab):

Find the roots of the denominator, as the solutions to the characteristic equations for z (use Matlab):

Assuming the roots are in catersian coordinates (Re + Im), convert one of the roots to polar coordinates (Radius & Angle).

Draw a Pole-Zero plot of the transfer function on the unit circle in the complex plane.

2 WRITING MATLAB CODE

Write the following code in MATLAB. When completed, create a new compressed zip folder comprised of all the MATLAB files. Name the zip file: XXXX_AET5420_Final.zip, where XXXX is your last name. Then email the zip file to: eric.tarr@belmont.edu. Confirm the receipt of your file before leaving.

2.1 SELF-OSCILLATING SYNTHESIS

There are many methods in audio which can be used to synthesize a signal. Previously, we have used trigonometic functions (e.g. sine, cosine, etc.) to synthesize basic test signals.

Another approach is to use a system with memory to synthesize a pure-tone signal through self-oscillation. This approach has the advantage of being very efficient to calculate. A block diagram for an oscillator is shown in Fig. 2.1.

This is a second-order system with two samples of feed-back delay. Therefore, it has two poles from the roots in the denominator of the transfer function.

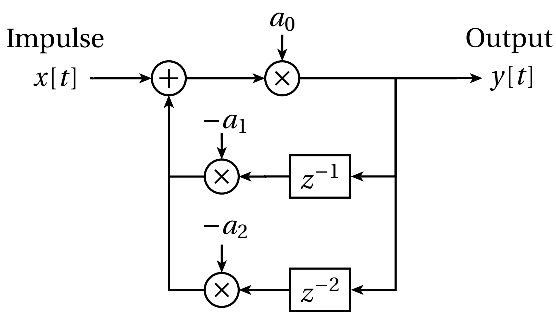


Figure 2.1: Oscillator Block Diagram

The premise of this approach is based on the stability of a system. Recall, there are three types of behavior for a system with memory. If a system is ***stable***, then the impulse response decays to zero over time. If a system is ***unstable***, then the impulse response approaches infinity over time.

If a system is ***marginally stable***, then the impulse response goes on forever without approaching zero or infinity. **This is the behavior we can utilize for synthesis.**

A system is marginally stable when its ***poles*** are located on the unit circle. Therefore, the magnitude of each pole is 1. The angle of the pole represents the frequency at which the oscillator resonates, or rings. Consider the system which produces the following pole-zero plot in the z-plane.

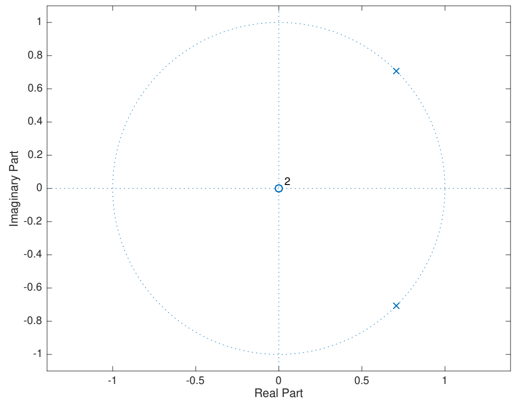


Figure 2.2: Pole-zero Plot for Oscillator

In this case, one pole has an angle of $\frac{\pi}{4}$ and one pole has an angle of $\frac{-\pi}{4}$. When presented with an impulse input signal, this system will self-oscillate indefinitely. The system will produce an output signal with a frequency at: $\frac{Nyquist}{4}$. By changing the angle of the poles, an output signal with a different frequency can be produced.

Don't forget, the poles must be a pair, symmetric over the real axis.

2.1.1 Problem

Create and save a **function** (m-file) in MATLAB that performs the following steps:

- Name the function: oscillator.m
- The function should have the following input/output variables:
  - *freq* : desired frequency in Hz
  - *lenSec* : length in seconds
  - *Fs* : sampling rate
  - *out* : synthesized signal
- Use the following steps within the function to synthesize the output signal
  - Convert the desired frequency in Hz to an angle (relative to Nyquist = π)
  - Convert the radius and angle from polar coordinates to cartesian coordinates
    * One pole should have a positive imaginary part, and the other negative
  - You have now found the roots of the denominator of the system's transfer function
  - Expand the roots to a polynomial expression for the denominator
    * At this point you have determined the coefficients: a = [a0, a1, a2];

    - Assume: b = 1;
  - Take the impulse response of the system for the input length of time
    - Use this as the output signal
  - Normalize the amplitude of the output signal

Use the test script, *oscTest.m* to ensure proper performance of your function. Remember to add comments to your code to explain what each command is accomplishing. For this problem, you will submit the function file: **oscillator.m** and the test script.