J-DSP Lab 2: The Z-Transform and Frequency Responses

Introduction

This lab exercise will cover the Z transform and the frequency response of digital filters. The goal of this exercise is to familiarize you with the utility of the Z transform in digital signal processing. The Z transform has a similar role in DSP as the Laplace transform has in circuit analysis:

a) It provides intuition in certain cases, e.g., pole location and filter stability,
b) It facilitates compact signal representations, e.g., certain deterministic infinite-length sequences can be represented by compact rational z-domain functions,
c) It allows us to compute signal outputs in source-system configurations in closed form, e.g., using partial functions to compute transient and steady state responses.
d) It associates intuitively with frequency domain representations and the Fourier transform

In this lab we use the ***Filter*** block of J-DSP to invert the Z transform of various signals. As we have seen in the previous lab, the ***Filter*** block in J-DSP can implement a filter transfer function of the following form

$$H(z) = \frac{\sum_{i=0}^{10} b_i z^{-i}}{1 + \sum_{j=1}^{10} a_j z^{-j}}$$

This is essentially realized as an I/P-O/P difference equation of the form

$$y(n) = \sum_{i=0}^{L} b_i x(n-i) - \sum_{i=1}^{M} a_i y(n-i)$$

The transfer function is associated with the impulse response and hence the output can also be written as

$$y(n) = x(n) * h(n)$$

Here, * denotes convolution; *x(n)* and *y(n)* are the input signal and output signal respectively. If the input to the ***Filter*** block is an impulse, the output of the ***Filter*** block will be *h(n)*, i.e., the inverse Z transform of the system function *H(z).* For all frequency responses use a Fast Fourier Transform (FFT) of size 256 by choosing “FFT size” 256 in the ***FFT*** block dialog window.

Problem 2-1: Exponential Sequences

We have seen that single pole IIR filters have exponential impulse responses. Design and simulate a digital filter that has the impulse response

$$h(n) = 0.9^n u(n)$$

(Hint: you can synthesize this by exciting an appropriate filter with a unit impulse. u(n) is the unit step signal and it simply implies that *n* >=0 ,i.e., *h(n)* is causal)

a) Give the simulation diagram and a transfer function: *H(z)* =? Corresponding to *h(n)* ?

b) Use J-DSP to plot the impulse response.

c) Use J-DSP to simulate and plot the frequency response:

Is this LPF or HPF?

Is the filter stable?

Write the poles and zeros of the system.

What would be the closest analog circuit to this first order filter (only for 2.1.3)?

d) Repeat from a) to c)

$$h(n) = (-0.8)^{n-1} u(n-1)$$

Note that u(n-1) simply denotes that the signal starts at sample one.

$$h(n) = (n+1)(0.9)^n u(n)$$

Hint: Remember the z-transform pair.

$$(n+1)p^n \Leftrightarrow \frac{z^2}{(z-p)^2}$$

Realize this as a filter using J-DSP

e) Given the transfer function for a causal system

$$H(\mathrm{z}) = \frac{1}{1 - 3\mathrm{z}^{-1} + 3z^{-2} + 2z^{-3}}$$

i) Does this transfer function represent a stable system?

Problem 2-2: Digital Oscillator

Digital oscillators can be designed by selecting the coefficients of the digital filter so that we get oscillatory behavior when the excitation is a unit impulse.

Design and simulate a digital oscillator for a sampling frequency of 8000 Hz and a sinusoidal frequency of 687 Hz. Note that the impulse response must be of the form:

$$h(n) = \cos(\Omega_0 n) u(n)$$

Use J-DSP to plot the frequency response and the poles and zeros.

Hint: find the frequency in radians

Problem 2-3: Canceling Sinusoidal Components

Filters can be designed to cancel sinusoids. Consider a system with the following impulse response:

$$h(n) = \delta(n) - 2\cos(\pi/4)\delta(n-1) + \delta(n-2)$$

The input signal to the system is

$$\begin{aligned} &x(n) = \sin(\Omega n)u(n) \\ &\text{with, a) } \Omega = \pi/2 \\ &\qquad\ \ \text{b) } \Omega = \pi/4 \end{aligned}$$

To generate the input signal, use the following parameters:

- Signal Type: sinusoid
- Amplitude: 1
- Time Shift: 0

For this problem, please address the following:

i) Plot the frequency response and observe poles and zeros.
ii) Simulate the convolution *y(n) = x(n)*h(n)* by programming *h(n)* as an FIR filter for the two sinusoidal inputs above. Use J-DSP and plot the output of the system.
iii) Note the behavior of this system with the two sinusoidal inputs above. When do we get cancellation?

Problem 2-4: Symmetric Impulse Response Produces Linear Phase

Linear phase filters have constant group delay and are important in some applications, i.e., high-end audio amplifiers and systems. Consider the following system with a symmetric impulse response:

$$H(z) = 1 + 0.5z^{-1} + 0.25z^{-2} + 0.25z^{-3} + 0.5z^{-4} + z^{-5}$$

a) Simulate the system in J-DSP and look at the phase response (use ***Freq Resp*** block).

b) Comment on the phase response (what is the slope?). What is the group delay of the filter?

c) Use pole-zero display and check for symmetries in the locations of zeros.

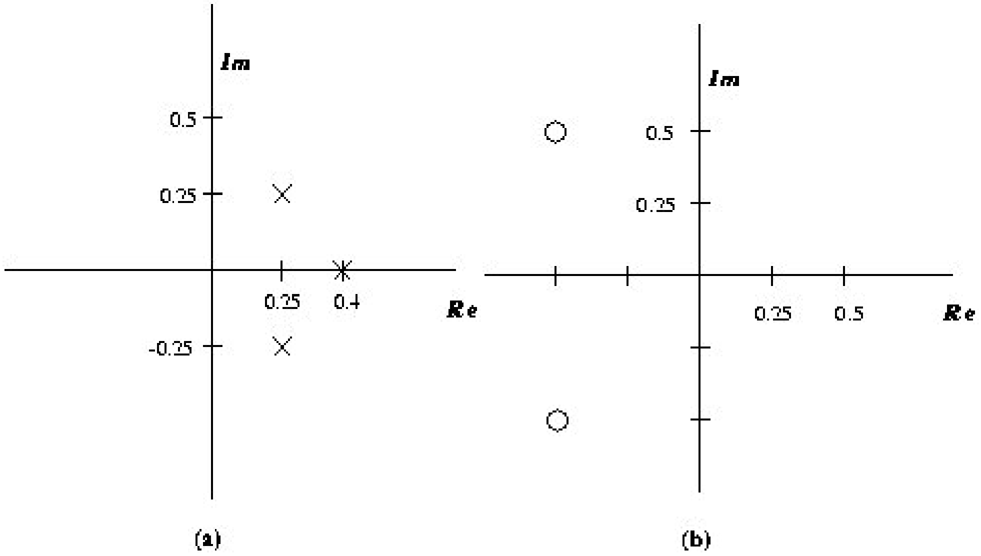
(Note that because of symmetry in h(n) we get symmetrical zeros in the z plane)

The group delay is the negative of the derivative of the phase with respect to the frequency. The group delay is measured in samples and can be determined from the slope of the phase.

Problem 2-5: Pole-Zero Plots

Let us now consider the role of poles and zeros in digital filters and their frequency response.

The figure below shows the pole-zero configuration of two linear time-invariant systems:



For each system,

a) Determine the coefficients of *H(z)* (use some simple algebra to form the filter coefficients)

b) Simulate and obtain the impulse response and frequency response of the two filters.

c) What are the effects of the poles and zeros on the frequency response?

Problem 2-6: Cascade and Parallel System Configurations

a) Consider the following system:

$$y(n) = \sum_{k=0}^{n} c^k d^{n-k}$$

Hint: Cascade configurations. Note that this is a convolution of two exponential signals

i) Implement the system using one ***Sig Gen***, two ***Filter***, two ***Coeff,*** blocks, and one ***Plot*** block for c = 0.5 and d = 0.25. Sketch the output of the system.

ii) Implement the system using one ***Sig Gen***, one ***Filter***, one ***Coeff.*** block, and one ***Plot*** block for c = 0.5 and d = 0.25

b) Consider the following system parallel configurations:

$$h(n) = [0.5^n + 2(-0.9)^n] \cdot u(n)$$

i) Implement the system using two ***Sig Gen***, two ***Filter***, two ***Coeff.*** blocks, one ***Mixer*** and one ***Plot*** block.

ii) Implement the system using one ***Sig Gen***, one ***Filter***, one ***Coeff.*** block, and one ***Plot*** block.