Lab #4: Z-transforms, Transfer Functions, Pole-Zero Plots, and Impulse Responses

PURPOSE: To better understand the Z transform sum and its associated ROC. In particular, numerical evaluation of the Z transform sum for different values of z will emphasize that the Z transform sum does not converge outside the ROC. Students will use the Matlab function, **`zplane`**, to make pole zero plots of several rational Z-transforms. Interpretation of a Z transform, $H(z)$, as the transfer function of a causal system whose impulse response is $h[n]$ will allow the student to understand that Matlab's **`filter`** function can be used to find the inverse Z transform, $h[n]$, for $n = 0, ..., n_{\text{max}}$. The student will also explore the relationship between time-domain signal characteristics and pole locations of the Z transform.

BACKGROUND: Let $h[n]$ denote a causal signal with ***rational*** Z transform $H(z)$ and region of convergence $|z| > |r|$. Then:

$$H(z) = \sum_{n=0}^{\infty} h[n]z^{-n} \tag{1}$$

and for $|z| > |r|$, $H(z)$ converges to a closed-form expression of the form:

$$H_{\text{cf}}(z) = \frac{b_0 + b_1 z^{-1} + b_M z^{-M}}{a_0 + a_1 z^{-1} + a_N z^{-N}} \tag{2}$$

As discussed in class, $H_{\text{cf}}(z)$ has interpretation as the transfer function of a causal discrete-time LTI filter with impulse response $h[n]$. Furthermore, the LCCDE which relates the filter's output sequence, $y[n]$, to the filter's input sequence, $x[n]$, is easily shown to be:

$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{k=0}^{M} b_k x[n-k] \tag{3}$$

Since $h[n]$ is the zero-state response of the system when $x[n] = \delta[n]$, the impulse response, $h[n]$, $n = 0, ..., n_{\text{max}}$ can be found numerically using Matlab's filter function as follows:

```
n = 0:nmax
Kdel = (n == 0)*1;
b = [b0,  b1,  ...,  bM]
a = [a0,  a1,  ...,  aN]
h = filter(b,a,Kdel)
```

with appropriate values substituted in the code above for n_{max}, b_0, b_1, $...b_M$, a_0, a_1, $...a_N$.

In addition, provided you have a closed-form expression for $H(z)$ in the form of (2), you can generate a pole-zero plot of $H(z)$ using the Matlab function **`zplane`** as follows:

```
zplane(b,a)
```

where the vectors **`b`** and **`a`** are as defined above.

EXERCISES:

1. Let $h[n] = \beta^n u[n]$ where $\beta = -0.8$. Then, by definition:

$$H(z) = \sum_{n=0}^{\infty} \beta^n z^{-n} \tag{4}$$

a) For $|z| > |\beta| = 0.8$, $H(z)$ converges to the following closed-form expression:

$$H_{\text{cf}}(z) = \frac{1}{1 - \beta z^{-1}} = \frac{1}{1 + 0.8z^{-1}} = \frac{z}{z + 0.8} \tag{5}$$

Evaluate the closed-form expression in (5) at the following values of z:

i. $z = 1$

ii. $z = -0.81$

iii. $z = -0.81 + j0.01$

iv. $z = -0.7$

For which values of z above do you expect $H(z)$ to agree with $H_{\text{cf}}(z)$?

b) As described in the BACKGROUND section, $h[n]$ is the impulse response of a causal LTI system with transfer function $H_{\text{cf}}(z)$. Use Matlab's **`filter`** function to find and stem $h[n]$ vs. n, for $n = 0, \ldots, 50$. Note that the values of $h[n]$ will be quite small for $n \geq 50$.
Hint: try **`h = filter(b,a,Kdel)`** with **`b`**, **`a`**, and **`Kdel`** defined as suggested in the BACKGROUND section.

c) Following the instructions in the BACKGROUND section, use the Matlab function **`zplane`** to make a pole-zero plot of $H(z)$. (Note: a pole-zero plot of $H(z)$ really refers to a pole-zero plot of $H_{\text{cf}}(z)$.)

d) The Z-transform of any causal sequence is given by the sum in (1). Provided the sum converges, it is reasonable to assume that a good approximation to $H(z)$ will be provided by $H_{\text{num}}(z)$, a truncated version of the sum in (1), as shown below.

$$H_{\text{num}}(z) = \sum_{n=0}^{n_{\max}} h[n] z^{-n} \tag{6}$$

Since the sum in (6) has a finite number of terms ($n_{\max} < \infty$), it can be evaluated numerically. Hence, we will refer to $H_{\text{num}}(z)$ as a numerical approximation of $H(z)$. Create a file named **ZTnum_eval.m** with the following lines of code.

```
function Hnum = ZTnum_eval(h,zval)
%h is a row vector containing the values h[n], n = 0,...,nmax
%zval is the value of z at which Hnum(z) is to be evaluated
%Hnum is Hnum(zval)
nmax = length(h) – 1;
n = 0:nmax;
z_to_the_minus_n = zval .^ (–n);
summand = h .* z_to_the_minus_n;
Hnum = sum(summand);
```

With the vector **h** as defined in part (b), use the function you just created to evaluate $H_{num}(z)$ at the same values of z for which you evaluated $H_{cf}(z)$ in part (a) above. For example, $H_{num}(1)$, can be found as: **`ZTnum_eval(h,1)`**.
Make a table to compare the values of $H_{num}(z)$ and $H_{cf}(z)$ at each value of z requested in part (a). Explain any differences.

Hint: Referring to equation (6), what is the value of n_{max} associated with the values of $H_{num}(z)$ in your table? For which values of z will $H_{num}(z)$ converge to $H_{cf}(z)$ in the limit as $n_{max} \to \infty$. In order to shed some light on any unexpected significant differences between $H_{num}(z)$ and $H_{cf}(z)$, you may want to add a few additional command lines to the function **`ZTnum_eval`** that you created above so as to generate a plot of **`abs(summand)`** vs. **`n`** for the various values of z.

2. As described in the background section, the coefficient vectors, **`b`** and **`a`**:
   `b = [`$b_0, \quad b_1, \quad \ldots, \quad b_M$**`]`**
   `a = [`$a_0, \quad a_1, \quad \ldots, \quad a_N$**`]`**
   can be used to specify an LTI system whose input-output relationship is given by the LCCDE in (3) and whose transfer function, $H(z)$, is given by (2). For each pair of coefficient vectors below: write out the associated transfer function $H(z)$; find the poles and zeros of $H(z)$; make a pole-zero plot of $H(z)$ and a stem plot of the system's impulse response, $h[n]$, for $n = 0, \ldots., 15$. In your lab report, discuss the relationship between the pole locations of $H(z)$ and the nature of $h[n]$.

   a)
   ```
   b1 = [1]
   a1 = [1, -0.8]
   ```
   b)
   ```
   b2 = [0, 0.8]
   a2 = conv(a1,a1)
   ```
   c)
   ```
   b3 = [0, 1]
   a3 = conv([1, 1] , [1, 1])
   ```
   d)
   ```
   F0 = 1/6
   r4 = 0.8
   c4 = [1, -r4*exp(j*2*pi*F0)]
   b4 = [0, sin(2*pi*F0)]
   a4 = conv(c4,conj(c4))
   ```
   e)
   ```
   r5 = 1;
   c5 = [1, -r5*exp(j*2*pi*F0)]
   b5 = [0, sin(2*pi*F0)]
   a5 = conv(c5,conj(c5))
   ```
   f)
   ```
   b6 = r5*sin(2*pi*F0)*conv([0,1], conv([1,r5],[1,-r5]))
   a6 = conv(a5,a5)
   ```