EE-3220 LABORATORY

Week 6

Poles and Zeros – Impact on the Transfer Function and Transient Response

Goal – Investigate the impact of pole and zero locations on the system transfer function and the transient (impulse & step) response of an LTI system.

Materials – Laptop computer with MATLAB

Background

By convention, filter coefficients are stored in row vectors in MATLAB working from the current time to the past. For example:

```
b = 0.05 * [1 1]; % FIR/MA (moving average) coefficients, [b0 b1]
a = [1 -0.9]; % IIR/AR (autoregressive) coefficients, [a0 a1]
```

MATLAB can calculate the steady state frequency response for you from these values:

```
figure, freqz(b,a)
```

Unfortunately, freqz is not subplot-friendly, so it should be given its own figure when asked to directly generate the frequency response plot. (Try it in a subplot to see how it fails.) **`doc freqz`** to learn more about this function. There are “freqz” functions in both the “DSP System Toolbox” and the “Signal Processing Toolbox.” Your MATLAB install probably has both. Look at the help contents (left pane) to make sure you’re viewing help for the “Signal Processing Toolbox.” If you’re viewing the wrong one, there will be a link to “signal/freqz” at the top of the window that will take you to the needed documentation.

Let’s create a new figure window to contain the next 3 plots:

```
figure
```

And, let’s ask MATLAB to illustrate the location of the poles and zeros:

```
subplot(2,1,1), zplane(b,a)
```

`doc zplane` for more details.

MATLAB can calculate the location of the zeros and poles for us. They are the roots of the numerator and denominator polynomials in the z-transform. By convention, roots are stored in column vectors in MATLAB.

```
z = roots(b);
p = roots(a);
```

The following will give the same pole-zero plot as above:

```
zplane(z,p)
```

MATLAB knows you’re providing coefficients when you give it row inputs; otherwise it knows you are providing roots.

We can move from the roots back to the polynomials:

```
b2 = poly(z);
a2 = poly(p);
```

`roots` and **`poly`** are inverses, but the roots are the same if the whole polynomial is scaled by a constant, so **`poly(roots(x))`** may disagree with **`x`** by a constant gain factor.

MATLAB can also calculate the impulse and step responses given the coefficients of the difference equation. Consult the documentation for more information on these functions.

```
subplot(2,2,3), impz(b,a)
subplot(2,2,4), stepz(b,a)
```

Overview

Given the following systems, assemble a table that characterizes

1. the type of system (e.g., high-pass or low-pass),
2. the location of the pole(s) and zero(s),
3. the final values of the unit impulse and step responses of the system based on system response graphs,
4. the final value of the step response by evaluating the system z-transform for a steady state response (thus DTFT: z = $e^{j\omega}$) given a DC input (thus ω=0) (Calculate by hand or in MATLAB. **`polyval`** can help solve this general problem, but is probably overkill for this assignment.),
5. and the time after which the magnitude of the impulse response permanently falls below 0.1 of its maximum magnitude. Ignore the large initial value of some transient responses (which are due to b_1) that will appear as an outlier from an exponential decay in some of the plots.

Show your work by including MATLAB code, clearly labeled graphs, etc., after your summary table.

$$H_1(z) = \frac{0.05(1+\ z^{-1})}{1-0.9z^{-1}} \qquad H_5(z) = \frac{0.95(1-\ z^{-1})}{1-0.9z^{-1}}$$

$$H_2(z) = \frac{0.95(1+\ z^{-1})}{1+0.9z^{-1}} \qquad H_6(z) = \frac{0.05(1-\ z^{-1})}{1+0.9z^{-1}}$$

$$H_3(z) = \frac{0.45(1+\ z^{-1})}{1-0.1z^{-1}} \qquad H_7(z) = \frac{0.55(1-\ z^{-1})}{1-0.1z^{-1}}$$

$$H_4(z) = \frac{0.55(1+\ z^{-1})}{1+0.1z^{-1}} \qquad H_8(z) = \frac{0.45(1-\ z^{-1})}{1+0.1z^{-1}}$$

Questions

What feature(s) of the transfer function poles and zeros correspond to

1. a low-pass response?
2. being nearly linear phase?
3. a slow decay in the impulse response?