Lab 2, S0004E

Lp 4, 2018

Analysis of LTI Systems:

–Poles, Zeros, Coefficients, and Matlab–

James P. LeBlanc, 2004.
Revised 2005, 2008 & 2012.

Goal

This lab is intended to build understanding of the interrelations between discrete-time system transfer functions, frequency response, and Z-transforms.

Due: May 3, 23:59

Hand in a short report documenting your experience with the lab which includes appropriate plots (properly labeled) and a short discussion of each numbered point listed.

Poles, Zeros, Coefficients and Matlab

By way of example, we investigate how Matlab can be used for discrete-time systems. Consider the input/output difference equation,

$$y[n] = -0.7071y[n-1] - 0.25y[n-2] + x[n] - 1.3x[n-1] + 0.42x[n-2].$$

Taking its Z-transform yields,

$$\begin{aligned} Y(z) &= -0.7071z^{-1}Y(z) - 0.25z^{-2}Y(z) + X(z) - 1.3z^{-1}X(z) + 0.42z^{-2}X(z) \\ Y(z)(1 + 0.7071z^{-1} + 0.25z^{-2}) &= X(z)(1 - 1.3z^{-1} + 0.42z^{-2}). \end{aligned}$$

Thus the associated system's transfer function can be written as,

$$\begin{aligned} H(z) &= \frac{Y(z)}{X(z)} = \frac{1 - 1.3z^{-1} + 0.42z^{-2}}{1 + 0.7071z^{-1} + 0.25z^{-2}} = \frac{z^2 - 1.3z^1 + 0.42}{z^2 + 0.7071z^1 + 0.25} & (1) \\ &= \frac{(z-0.6)(z-0.7)}{(z-(-0.3536 + i0.2526))(z-(-0.3536 - i0.2526))} & (2) \end{aligned}$$

revealing two system poles at $z = 0.3536 \pm 0.3536i$ and two zeros at $z = 0.7$ and $z = 0.6$. Please note that if the difference equation has real-valued coefficients, then zeros and poles have occur as complex conjugate pairs, i.e., if $z = a+ib$ is a pole/zero then $z = a-ib$ has to be a pole/zero. Real-valued poles/zeros, such as $z = 0.6$, can however appear on their own. This is closely connected to the property that the spectrum of real-valued sequences have spectral symmetries.

In general, for causal systems, we can write the transfer function as a ratio of two polynomials in z. We can choose to use a form involving the polynomial's coefficients or polynomial's roots,

$$H(z) = \frac{\sum_{k=0}^{M-1} b_k z^{-k}}{\sum_{k=0}^{N-1} a_k z^{-k}} = \frac{\prod_{k=0}^{M-1} (z - c_k)}{\prod_{k=0}^{N-1} (z - d_k)}$$

Matlab provides useful tools to go back and forth from coefficient representation form and root representation form. In Matlab, to represent the polynomial

$$\phi(z) = z^3 + 0.2z^2 - 0.5$$

we construct a vector of its coefficients according to,

```
phi = [ 1 0.2 0 -0.5].
```

We can compute the roots of this polynomial by using the command `roots`,

```
roots(phi).
```

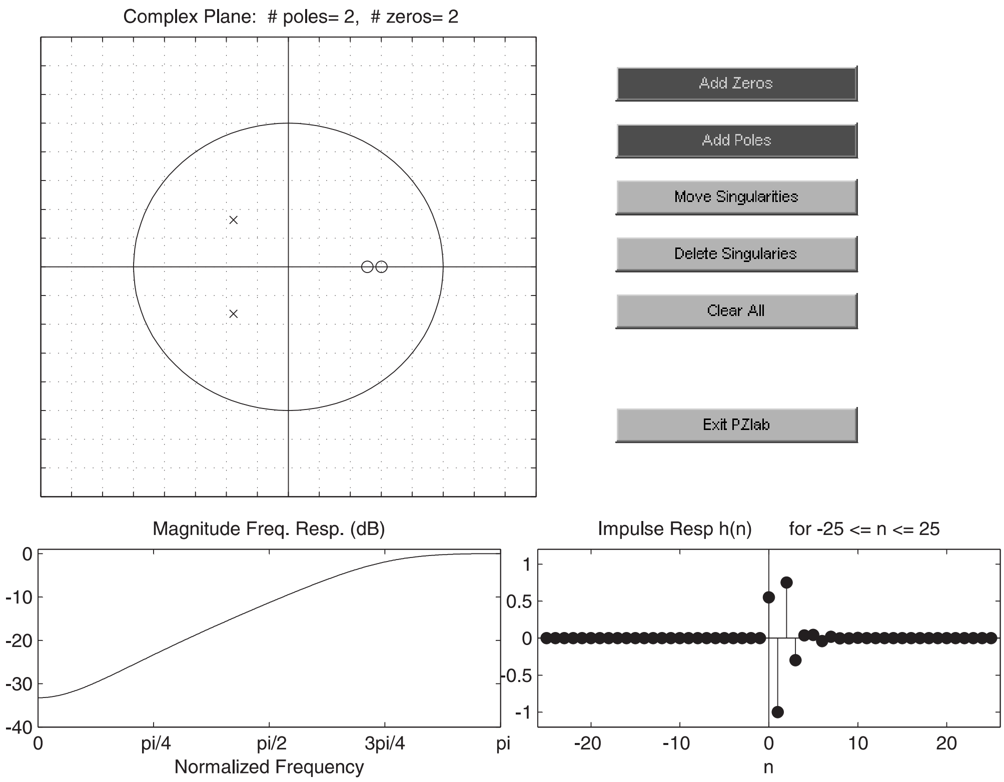
A vector containing the roots of our polynomial $\phi(z)$ will be returned. One can also go in the opposite direction, from roots to polynomials using the command `poly` For example, to construct a polynomial $\theta(z)$ with the roots at $z = 1$, $z = -1$, $z = 2$, we can type

```
theta = poly([ 1 -1 2])
```

See also some related commands `polyval, freqz, conv, filter` using the Matlab `help` facility.

Using PZlab

PZlab is a homebrew graphical user interface to simplify placing, and moving around transfer function singularities (i.e., poles and zeros) on the complex plane. Again, please note that since we only consider systems with real valued coefficients, and impulse responses, a complex pole/zero will be accompanied by its complex conjugate. With the graphical interface you can see the system's frequency response and impulse response associated with your chosen pole and zero locations. An example of the appearance for the PZlab interface for our earlier example is:



The interface allows the user to

- add zeros
- add poles
- move singularities (poles and zeros)
- delete singularities
- clear all singularities
- poles and zeros are saved upon exit, in the file name `PZoutput` in the vectors named `pz_poles` and `pz_zeros`.

Assignments

1. Analytically argue that if the coefficients of the difference equation are real, and $z = a + ib$ is a complex valued singularity, then its conjugate, $a - ib$ also has to be a singularity.

2. Open PZlab, and place a zero at $z = 0.5$. Is the resulting system causal? If not, add another singularity to make it causal. Hint: Both $(z - a)$ and $(1 - az^{-1})$ have a zero at $z = a$.

3. Move the zero at $z = 0.5$, progressively in steps towards the point $z = 1$. Describe the changes in the systems frequency response. Please note that since we only consider difference equations with real valued coefficients, the frequency response has symmetries that yield that it is sufficient to study the response for frequencies $0 \leq \omega \leq \pi$.

4. Move the zero at $z = 1$, progressively in steps counter clockwise around the unit circle. Again, describe the change in the systems frequency response. At what frequency does the null appear?

5. Clear all singularities, now repeat steps 2-4 with a pole instead of a zero.

6. Comment in general what happens to the frequency response with respect to the pole locations and the zero locations.

7. Try to design a causal notch filter that nulls out any input at $\omega = \frac{\pi}{4}$, by placing a root on the unit circle at the appropriate location. What is the corresponding input/output difference equation?

8. Improve your notch filter to make the notch narrower and steeper. Also, we'd like to have a very flat response everywhere else!
   Hint: try placing a couple of poles on both sides of your zero. Perhaps you'll need other singularities strategically located. Print out your resulting filter, and describe your efforts and results.

9. Assuming a sampling frequency of 44,100 Hz, ($\Omega_s = 2\pi \cdot 44.1 \cdot 10^3$ rad/s):

   (a) Use PZlab to design a filter that notches out the frequencies 500 Hz, 2000 Hz, 5000 Hz, but has nice peaks at 1000 Hz, and 8000 Hz.

   (b) Create the vector of associated coefficients from the saved pole and zero locations (located in the file `PZoutput`, created when exiting PZlab).

   (c) Use the command wavread, to read-in the wavfile `black.wav` (located on the course webpage), or any other audio clip of your choice. You may wish to read in only a few seconds of the file to reduce processing time (see `help wavread`).

   (d) Use the `filter` command, to process the sound with your designed system.

   (e) listen to you results for instance using the `sound` command. You may have to save your results as a wavfile, and then use whatever tool available to listen to the sound.