The Z Transform in MATLAB

Brian Frost-LaPlante

Spring 2018

This document is to serve as a lesson on the Z transform in MATLAB for students already familiar with the Z transform as a mathematical object. Please refer to the crash course on the Z transform document if you are not comfortable with this.

The Z Transform as a Rational Function

The Z transform, in practice, is often representable as a function that is rational in z. We have already discussed the treatment of polynomials in MATLAB as vectors, so the following should not be surprising:

Suppose you are looking to represent the Z transform $H(z) = \frac{z}{(z-.5)(z-2)}$ in MATLAB. You would simply set up the numerator and denominator polynomials as *row vectors*. You can then do all sorts of nice things with them! If you use column vectors instead, most Z transform functions will interpret these inputs as poles and zeros. This is an important distinction to keep in mind. As such, I can represent the above function in two ways:

```
b = [0 1 0]; % numerator polynomial
a = [1 -2.5 1];   % denominator polynomial
z = [0]; % zeros of H
p = [.5; 2];  % poles of H
```

Most Z transform functions in MATLAB will interpret the pairs of inputs (a, b) and (p, z) in the same way. Here are some examples of Z transform functions:

zplane(b,a), with no output arguments, simply plots the poles and zeros of a transfer function. Test it out on the above vectors and see that you will get the same answers. The figures it creates are quite nice! You can subplot and title them like any other figure. You can also store its output in a vector to get the poles and zeros of a rational transfer function.
impz(b,a,k), with no output arguments, plots the first k samples of the impulse response in the discrete time domain. You can write *[h t] = impz(b,a,k)* to get the impulse response h as a vector. The vector t will be equivalent to $0 : k$.
y = filter(b,a,x) returns the output signal y of the system described by b and a with input x.

You should know that Z this "filter" command is equivalent to convolution! The MATLAB function ***conv*** will convolve two vectors. Should you actually care to, you should try out all of these functions on the transfer function above. It is worth it, as you will have to be comfortable with this machinery in your ECE classes in the future.