

## Get your Z's: The Z-transform and Transfer Functions

This tech note concerns a 3-step chain of math ideas starting at the Z-transform, moving to the Convolution Theorem, and finishing with the grand final Transfer Function.

### 1 Z-transform

The z-transform is a cute little math tool/trick. Let's say we have a  $N$  samples of a digital signal  $x[n]$  for  $n = 0, 1, 2, \dots, N - 1$ . In physical terms  $x[n]$  could be recordings of pressure waves, temperature, music, voltage, you name it.

Now define the z-transform of  $x[n]$  as follows:

$$Z \{x[n]\} = X(z) = \sum_{n=0}^{N-1} x[n]z^{-n} \quad (1)$$

where  $z = a + jb$  is a number in the complex plane.

We can write the z-transform long-hand like so:

$$X(z) = x[0]z^{-0} + x[1]z^{-1} + x[2]z^{-2} + \dots + x[N - 1]z^{-(N-1)} \quad (2)$$

One hint of why the z-transform is useful in practice: It is just a discrete Fourier transform (DFT) in disguise!

If we let  $z = e^{j\omega_m T}$ , then  $z^{-n} = e^{-j\omega_m T n} = e^{-j\omega_m t_n} = s_m[n]$ . Here, we have:

- $T$  is the sampling period
- $t_n$  is the time at which the  $n$ th sample is acquired
- $\omega_m = m\omega_o = m\frac{2\pi}{NT} = m\Omega_o$  is the angular frequency for a sequence  $N$  elements long
- $s_m[n]$  is the  $m$ th Fourier basis function written as complex exponential  $e^{-j\omega_m t_n}$

So the z-transform can thus be written:

$$X(z) = X(m) = \sum_{n=0}^{N-1} x[n]s_m[n]$$

which is exactly the DFT we previously saw! Notice we are using  $m$  to index frequency.

## 1.1 Real-world example of Z-transform: The moving mean filter

Let's take an early look at what's fun and good about the z-transform. Let's take a sequence that should look very familiar to you:  $b[n] = \{1/M, 1/M, \dots, 1/M\}$ , where  $n = 0, 1, 2, \dots, M - 1$ . Of course this is just the coefficients for a moving average filter with a window size of  $M$ .

The z-transform of  $b[n]$  is:

$$B(z) = \sum_{n=0}^{M-1} \frac{1}{M} z^{-n} \quad (3)$$

$$= \frac{1}{M} \sum_{n=0}^{M-1} z^{-n} \quad (4)$$

$$= \frac{1}{M} (1 + z^{-1} + z^{-2} + \dots + z^{-(M-1)}) \quad (5)$$

This is just a geometric series in  $z^{-1}$ , so we can recall our good old days in high school math, or when- or where-ver you learned the formula for summing a geometric series to write:

$$B(z) = \frac{1}{M} \left( \frac{1 - z^{-M}}{1 - z^{-1}} \right) \quad (6)$$

$$B(e^{j\omega_m T}) = \frac{1}{M} \left( \frac{1 - e^{-j\omega_m T M}}{1 - e^{-j\omega_m T}} \right) \quad (7)$$

$$(8)$$

If we take the magnitude  $|B(e^{j\omega_m T})|$ , after some algebra and trig IDs we obtain:

$$|B(e^{j\omega_m T})| = \frac{1}{M} \frac{\sin(M\omega_m T/2)}{\sin(\omega_m T/2)} \quad (9)$$

$$= \frac{1}{M} \frac{\sin(Mm\pi/N)}{\sin(m\pi/N)} \quad (10)$$

Eqn 9 is a modest hot mess of variables that merits review and further explanation of how to interpret this math madness.

Firstly, the big picture:  $|B(e^{j\omega_m T})|$  expresses the **magnitude response** of a moving average filter to an input sequence oscillating at angular frequency  $\omega_m$ . The sub- $m$  index denotes the  $m$ th discrete frequency that we can analyze:  $\omega_m = m \frac{2\pi}{NT} = m\omega_o$ , where  $N$  is the number of samples in an input sequence to which we apply the moving mean filter, and  $T$  is the sampling period of our input data sequence. (The sampling frequency is just  $f_s = 1/T$ ).

The plot of this function for  $m = 1, 2, \dots, N - 1$  looks like like a sinc function (Figure 1). The magnitude response is clearly greater at lower frequencies and smaller at higher frequencies. This analysis with the z-transform shows the moving mean filter acts just like a low-pass filter!!!

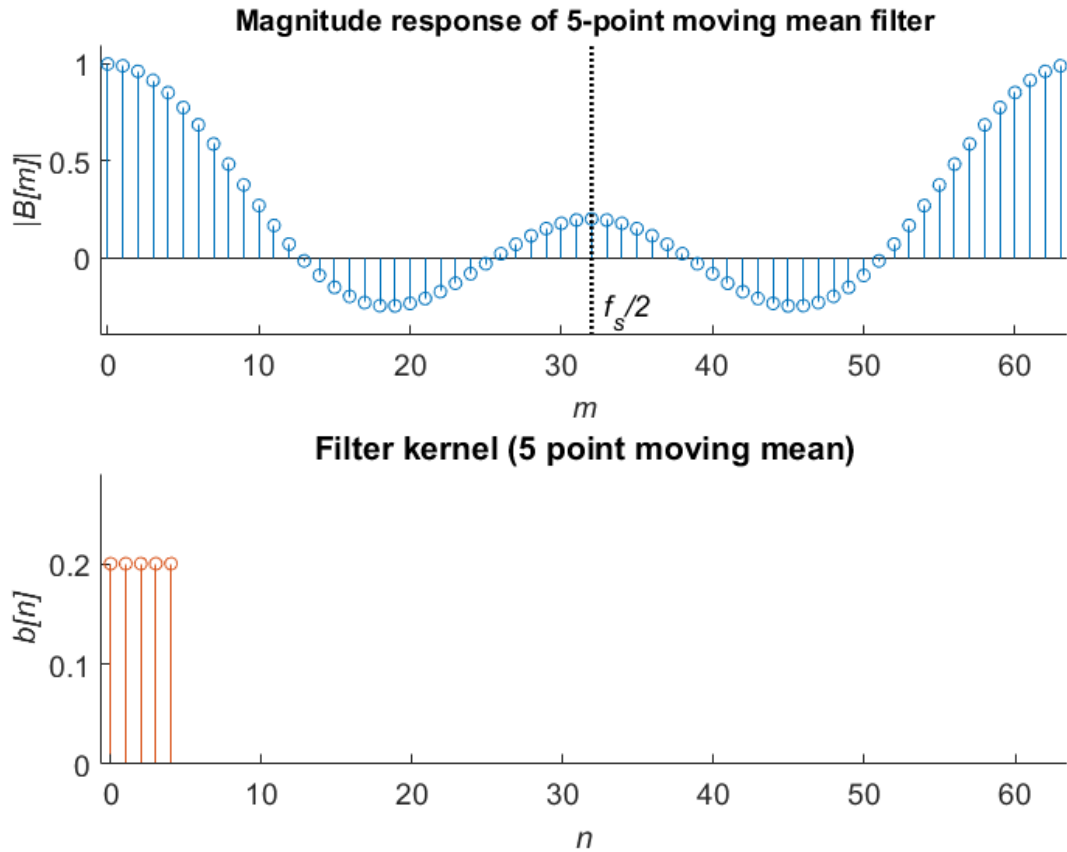


Figure 1: Bottom: filter coefficients (labeled  $b[n]$ ) for a  $M = 5$  point moving mean filter. Top: Magnitude response of the filter vs. frequency integer  $m$ . The physical frequency is given by for each integer  $m$  corresponds to  $\omega_m = m\omega_o = m\frac{2\pi}{NT}$ , where  $T$  is the sampling period ( $= 1/f_s$ ), and  $N$  is the length of the input data sequence to which the filter is applied (in this case  $N = 64$ ). Note the symmetry about the  $m = N/2 = 32$ , or equivalently around  $f_s/2$ .

## 2 Convolution Theorem

The Convolution Theorem is another useful math trick that finds way more practical use than it has any right to on the surface. In essence, this theorem says the z-transform of a convolution in the time domain is just a product of z-transforms:

$$b[n] * x[n] \Leftrightarrow B(z)X(z) \quad (11)$$

We'll see in just a few moments that the Convolution Theorem gives us the all-important transfer function. But first things first—to the math-mobile!

Let's take a digital sequence that we'll call  $u[k]$  which is computed as the time-domain convolution of some coefficients  $b[n]$  a digital signal  $x[k]$  (real world physical quantity).

We will assume  $b[n]$  has  $N$  elements ( $n = 0, 1, 2, \dots, N - 1$ ), and  $x[k]$  is of finite-length ( $k = 0, 1, 2, \dots, K - 1$ ), where  $k$  is the integer time index. Note also, that  $u[k]$  also has  $K$  elements, the same number of elements as  $x[k]$ .

In fancy math formula the convolution is just:

$$u[k] = \sum_{n=0}^{N-1} b[n]x[k-n] \quad (12)$$

Note there are  $N$  terms in the convolution because  $b[n]$  has  $N$  terms.

Now let's find the z-transform of  $u[k]$ :

$$U(z) = \sum_{k=0}^{K-1} u[k]z^{-k} = \sum_{k=0}^{K-1} \left( \sum_{n=0}^{N-1} b[n]x[k-n] \right) z^{-k} \quad (13)$$

where we have simply sub'd in Eqn 12 into the RHS of Eqn 13.

Now let's play a little math trick multiplying the RHS by a special form of  $1 = z^{-n}z^n$  to get:

$$U(z) = \sum_{k=0}^{K-1} \sum_{n=0}^{N-1} b[n]x[k-n]z^{-k}z^{-n}z^n \quad (14)$$

$$= \sum_{k=0}^{K-1} \sum_{n=0}^{N-1} b[n]z^{-n}x[k-n]z^{-(k-n)} \quad (15)$$

$$= \underbrace{\sum_{n=0}^{N-1} b[n]z^{-n}}_{B(z)} \sum_{k=0}^{K-1} x[k-n]z^{-(k-n)} \quad (16)$$

$$(17)$$

In the penultimate step, we just rearranged terms in a clever way. In the last step, the sum on  $k$  “slides” through because the terms that are indexed only by  $n$  act like a constant. We can

already see the term on the left is  $B(z)$ , the  $z$ -transform of the coefficients  $b[n]$ . The right term needs just a little massaging and reindexing as the time-index is  $k - n$  but the sum is over  $k$ .

Making the substitution  $l = k - n$ , the summation limits would be from  $l = 0 - n = -n$  to  $l = (K - 1) - n = L - 1$ . The lower limit is negative valued, e.g. it uses  $x[-n]$  in the first term of the sum. However, we assume the sequence starts at discrete time 0, and for “negative time”  $x[k < 0] = 0$ . So, the lower index of the sum can thus be set to  $l = 0$ . Thus, we can write at long last:

$$U(z) = \underbrace{\sum_{n=0}^{N-1} b[n]z^{-n}}_{B(z)} \sum_{l=0}^{L-1} x[k-n]z^{-(k-n)} \quad (18)$$

$$= \underbrace{\sum_{n=0}^{N-1} b[n]z^{-n}}_{B(z)} \underbrace{\sum_{l=0}^{L-1} x[l]z^{-l}}_{X(z)} \quad (19)$$

$$U(z) = B(z)X(z) \quad (20)$$

Badabing-badaboom, we have shown the convolution theorem. Q.E.D., hooray, do a happy dance cuz that was a lot of careful work.

To sum up, the convolution theorem says: ***A convolution sum in the time domain is equivalently a product of  $z$ -transforms***, as was foretold by the math prophets in AD 314. Yeah, if that’s not abstract, I don’t know what is. But this does turn out to be imminently useful

### 3 Transfer Function

All right, now the moment we’ve all been waiting for, the headline act is just about to come out on to stage. Ladies and gentlemen, please give your warmest welcome to the Transfer Function—woot woot!

Remember back at the ranch we said we can create a filtered digital signal like so:

$$y[n] = \sum_{k=0}^{N-1} b[k]x[n-k] - \sum_{l=1}^{L-1} a[l]y[n-l] \quad (21)$$

Let’s do a little re-arranging of Eqn 12 to see where not one, but two(!) convolution sums are hiding Let’s let  $a[l = 0] = 1$ , then we can write:

$$a[0]y[n] + \sum_{l=1}^{L-1} a[l] y[n-l] = \sum_{k=0}^{N-1} b[k] x[n-k] \quad (22)$$

The left most term we can just slip into the sum over  $l$  and update the indices to include  $l = 0$ :

$$\sum_{l=0}^{L-1} a[l] y[n-l] = \sum_{k=0}^{N-1} b[k] x[n-k] \quad (23)$$

Here note that  $n$  is the integer time index, whereas  $l$  and  $k$  are the time-shift indices.

Now take a good long look at Eqn 23 and see if you can spot the convolutions. Yep. Good. You got it! The LHS is one convolution  $a[n] * y[n]$ , where  $*$  means convolution. And the RHS is another convolution  $b[n] * x[n]$ .

Now think back to a galaxy far, far away, where the Convolution Theorem reigned supreme, wherein the Benevolent Math Emperor carved into the solid titanium walls:

$$A(z)Y(z) = B(z)X(z) \quad (24)$$

This can be quickly rearranged to read:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{B(z)}{A(z)} \quad (25)$$

How do we interpret this deceptively simple Eqn 25. Well, the  $Y/X$  term looks like the ratio of the output over the input. Kinda like that whole  $V_{out}/V_{in}$  **transfer function** thing from Circuits (don't run away screaming just yet!).

In fact, we have just figured out how to write the digital signal processing equivalent. Indeed, we term  $H(z)$  the transfer function for the system. Input a sequence of number  $x[n]$ , the coefficients in  $b[n]$  and  $a[n]$  “shape” it into the output sequence  $y[n]$ . Similarly, the function  $H(z)$  tells us the input-output relationship for the system, just in a different mathematical way.

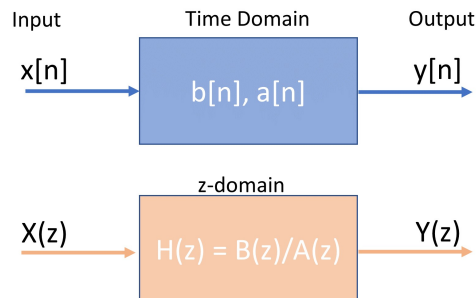


Figure 2: Graphical Representation of Transfer Function

So, the punchline, we can now write out Eqn 25 more explicitly (and remember  $a[0] = 1$ ):

$$H(z) = \frac{b[0] + b[1]z^{-1} + b[2]z^{-2} + \dots + b[N-1]z^{-(N-1)}}{1 + a[1]z^{-1} + a[2]z^{-2} + \dots + a[L-1]z^{-(L-1)}} \quad (26)$$

What is this madness you say? Well, firstly check out Matlab documentation on the filter function and scroll all the way to the bottom under “More About/Rational Transfer Function”: <https://www.mathworks.com/help/matlab/ref/filter.html> Look familiar? Now you know where this comes from; that’s powerful stuff!

With this formalism in hand, the goal now is to choose appropriate coefficient for  $a$  and  $b$  to implement the desired filter type (e.g low, high, band pass).

Second, let’s get a good graphic to help us illustrate interpret  $H(z) = B(z)/A(z)$ . The check out Figure 3, which illustrates the terms in the sum for  $B(e^{j\omega_m T})$ . That is, we are using  $z = e^{j\omega_m T} = e^{j\Omega_m}$ .

This complex number has a magnitude of 1, and a phase shift corresponding to digital frequency  $\Omega_m = \omega_m T$ . So, each successive term  $z^{-n}$  shifts backwards in term (clockwise) in phase by an amount  $\phi_n = -\Omega_m n$ . The successive phase shift is shown in Fig 3, left. Just how much does each successive term phase shift? Well, that depends on  $\Omega_m = \omega_m T = m\omega_o T = m\frac{2\pi}{NT}T = m\frac{2\pi}{N} = 2\pi\frac{m}{N}$ .

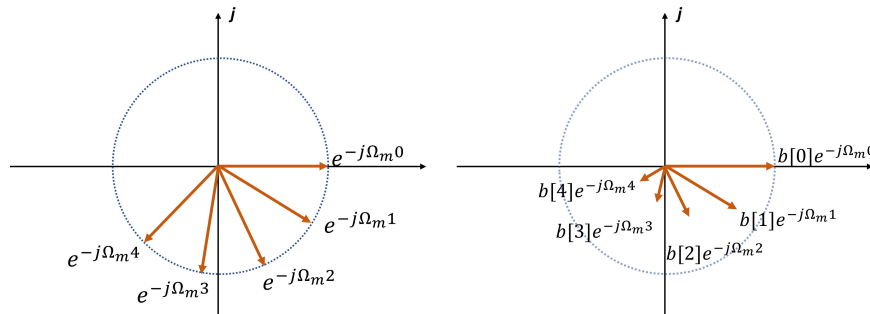


Figure 3: Visualization of the terms summed in  $B(z)$ . Left: unit circle showing phase shifts corresponding to the terms  $z^{-n} = e^{-j\Omega_m n}$ . In this example, the phase shift  $\Omega_m$  is about 36 degrees per sample, and we have 5 total terms in the sum ( $n = 0, 1, 2, 3, 4$ ). Right: Scaling the phase shifted vectors according to  $b[n]$  to visualize the sum in  $B(z)$ . Notice how each vector in the sum changes magnitude, but the phase shift relation remains the same.

In this example, the phase shift  $\Omega_m$  is about 36 degrees per sample (about 0.61 rad/sample) for 5 successive terms ( $n = 0, 1, 2, 3, 4$ ). The 0.61 rad/sample phase shift would result, for instance, for the case  $m = 1$  and a 10 element sequence for  $b[n]$ . What would this picture look like for  $m = 0$  (hint: all vectors drawn would line up on the real axis). Therefore, what would the sum look like? Now, what does the sum look like for the vectors as drawn? Now, what if you plot the magnitude of the resultant vector for all  $m = 0, 1, 2, \dots, N$ ?

### 3.1 Connection to analog circuits

You might quickly see a close connection between the digital filter transfer function in Eqn 26 and stuff you've seen and know and loved from analog circuits.

For example, for a straight-up single-stage RC low-pass filter, we would write the transfer function like so

$$H_{low}(\omega) = \frac{1}{1 + j\omega RC}$$

To clean this up just a bit, we can let  $s = j\omega$  and then rewrite this analog RC circuit transfer function in terms of  $s$ :

$$H_1(s) = \frac{1}{1 + RC s}$$

This almost looks like the digital filter for filter setting just 2 coefficients:  $b[0] = 1$  and  $a[1] = RC$ . So that should be a hint from the analog world that there are analogous games to play in digital filter land.

We could of course also do something similar with a 2nd order filter where we would have the transfer function:

$$H_2(s) = \frac{1}{1 + RC s} \frac{1}{1 + RC s} = \frac{1}{1 + 2RC s + (RC)^2 s^2}$$

which suggests a 2nd order digital filter just needs one more coefficient  $a[2]$ , by analogy.

### 3.2 Gain and phase response of digital filter

Also, as a parallel (pun intended) with analog circuits, you need to know that we can plot the frequency response (gain of the digital system) by computing the magnitude of the transfer function using  $z = e^{j\omega_m T}$ :

$$|H(e^{j\omega_m T})| = \frac{|B(e^{j\omega_m T})|}{|A(e^{j\omega_m T})|}$$

Recall that  $B(e^{j\omega_m T})$  is equivalent to the DFT of the coefficients  $b[n]$ ; similarly,  $A(e^{j\omega_m T})$  is the DFT of coefficients  $a[l]$ .

For example back at the ranch we imagined a moving mean filter with coefficients:  $a[0] = 1$  and  $a[l] = 0$  for  $l > 0$ ; and  $b[n] = 1/M$  for  $n = 1, 2, \dots, M - 1$ .

We can just compute the DFTs, divide element-wise and voila, we have our frequency response!

We can also plot the phase response (phase distortion) of the digital filter by computing:  $\phi_H = \phi_B - \phi_A$ .



We'll hopefully have a chance to talk good and proper about how to choose coefficients, but for now, check out some various filters and their frequency responses in Figure 4. These plots look very similar to ones we studied in Circuits, except notice the side lobes in the cutoff region. Those don't exist in the analog world; they are a fundamentally part of the digital world. Notice the fan favorite Butterworth filter. It has a maximally flat passband. However, it cuts off less steeply (less-ideal) than the others.

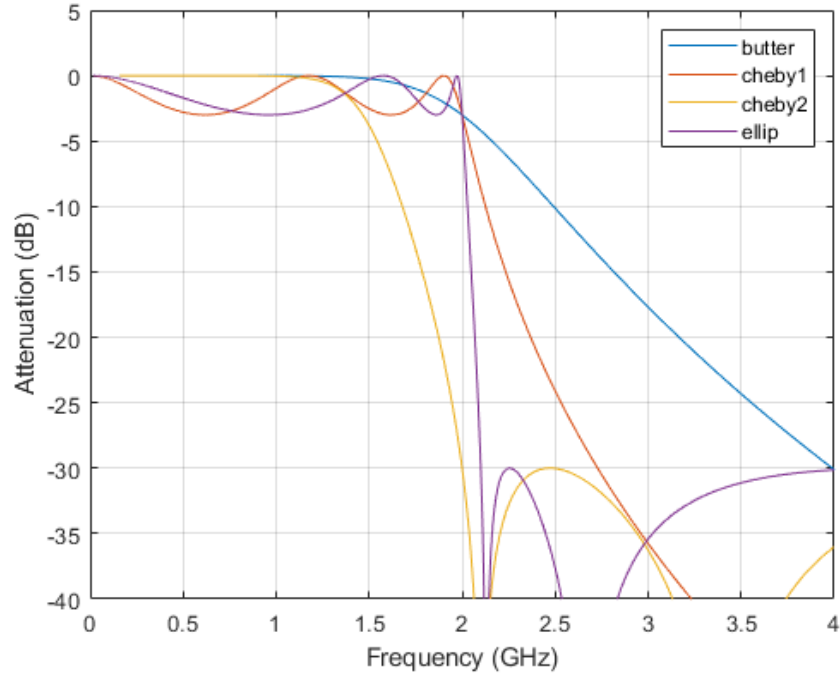


Figure 4: FIR low-pass filter examples. Cutoff frequency set for 2 GHz (fun with radio signals). Image credit: [urlhttps://www.mathworks.com/help/signal/ref/butter.html](https://www.mathworks.com/help/signal/ref/butter.html)

## 4 DFT and inverse DFT pairs

We've seen the Z-transform is essentially a DFT in disguise. For filter design, we sometimes need the inverse transform too. The DFT (Eqn 27) and its inverse (Eqn 28) always run around in pairs. Starting with the desired frequency response  $H[m]$  we can compute  $h[n]$  for each time index  $n = 0, 1, \dots, N$ , for an "N-tap" filter (i.e. the number of coefficients in the filter kernel). To do this, we apply Eqn 28.

$$H[m] = \sum_{n=0}^{N-1} h[n] e^{-j\omega_m t_n} \quad (27)$$

$$h[n] = \frac{1}{N} \sum_{m=0}^{N-1} H[m] e^{+j\omega_m t_n} \quad (28)$$