

## Coefficients of Cascaded Discrete-Time Systems

In this article, we'll show how to compute the coefficients that result when you cascade discrete-time systems. With the coefficients in hand, it's then easy to compute the time or frequency response. The computation presented here can also be used to find coefficients of mixed discrete-time and continuous-time systems, by using a discrete time model of the continuous-time portion [1].

Consider the cascade of two systems,  $H_1(z)$  and  $H_2(z)$ , where

$$H_1(z) = K_1 \frac{b_0 + b_1 z^{-1} + \dots + b_N z^{-N}}{1 + a_1 z^{-1} + \dots + a_N z^{-N}} = K_1 \frac{b_0 z^N + b_1 z^{N-1} + \dots + b_N}{z^N + a_1 z^{N-1} + \dots + a_N}$$

$$H_2(z) = K_2 \frac{d_0 + d_1 z^{-1} + \dots + d_M z^{-M}}{1 + c_1 z^{-1} + \dots + c_M z^{-M}} = K_2 \frac{d_0 z^M + d_1 z^{M-1} + \dots + d_M}{z^M + c_1 z^{M-1} + \dots + c_M}$$

$H_1(z)$  and  $H_2(z)$  are assumed to have the same sample rate.  $H(z) = H_1(z) \cdot H_2(z)$  can be found by multiplying-out the numerator and denominator polynomials of  $H_1$  and  $H_2$ . This can be done by hand, but it turns out there is a rule for multiplying polynomials [2]. The rule is as follows: Given two polynomials with coefficients  $b = [b_0 \ b_1 \ \dots \ b_n]$  and  $d = [d_0 \ d_1 \ \dots \ d_n]$ , the coefficients of the product of the polynomials are given by the convolution of  $b$  and  $d$ . Thus  $H(z) = H_1(z) \cdot H_2(z)$  has coefficients:

$$\text{numerator coeffs} = b \circledast d \quad (1)$$

$$\text{denominator coeffs} = a \circledast c \quad (2)$$

where  $\circledast$  indicates convolution. The overall gain constant is  $K_1 \cdot K_2$ . For FIR systems, the denominator is unity, so the coefficients of the cascaded systems are  $b \circledast d$  (which is perhaps no surprise).

### Example

Here we'll cascade 2<sup>nd</sup>-order lowpass and band-reject IIR filters with transfer functions  $H_1(z)$  and  $H_2(z)$  as follows:

$$H_1(z) = K_1 \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

where  $H_1(z)$  is lowpass with

$$\begin{aligned} a &= [1 \ .3695 \ .1958]; \\ b &= [1 \ 2 \ 1]; \\ K_1 &= .3913 \end{aligned}$$

$$H_2(z) = K_2 \frac{d_0 + d_1 z^{-1} + d_2 z^{-2}}{1 + c_1 z^{-1} + c_2 z^{-2}}$$

where  $H_2(z)$  is band-reject with

```
c= [1 -1.1074 .8841];
d= [1 -1.1756 1];
K2= .942
```

The frequency responses of the two filters are computed using Matlab as follows:

```
fs= 100; % Hz sample frequency
[h1,f]= freqz(K1*b,a,2048,fs);
H1= 20*log10(abs(h1)); % dB magnitude response
[h2,f]= freqz(K2*d,c,2048,fs);
H2= 20*log10(abs(h2)); % dB magnitude response
```

The responses are plotted in Figure 1. The cascade coefficients are computed using Equations 1 and 2:

```
bb= conv(b,d)
aa= conv(a,c)
K= K1*K2
```

This gives coefficients:

```
aa = 1.0000 -0.7379 0.6707 0.1098 0.1731
bb = 1.0000 0.8244 -0.3512 0.8244 1.0000
K = 0.3686
```

The length of aa is 5, so the cascade transfer function is 4<sup>th</sup> order. The transfer function is:

$$H(z) = K \frac{bb_0 + bb_1 z^{-1} + bb_2 z^{-2} + bb_3 z^{-3} + bb_4 z^{-4}}{1 + aa_1 z^{-1} + aa_2 z^{-2} + aa_3 z^{-3} + aa_4 z^{-4}}$$

(I apologize for the ungainly notation). The frequency response is:

```
[h,f]= freqz(K*bb,aa,2048,fs);
H= 20*log10(abs(h)); % dB magnitude response
```

The cascaded response is shown in the lower plot of Figure 1.

In case you are interested, the lowpass and band-reject coefficients were calculated using the following Matlab code:

```
fs= 100;
fc= 30;
[b,a]= butter(2,2*fc/fs) % lowpass coeffs
f0= 15;
fU= 16;
[d,c]= br_synth1(1,f0,fU,fs) % band-reject coeffs
```

Here, `br_synth1` is a Matlab function developed in an earlier post [3].

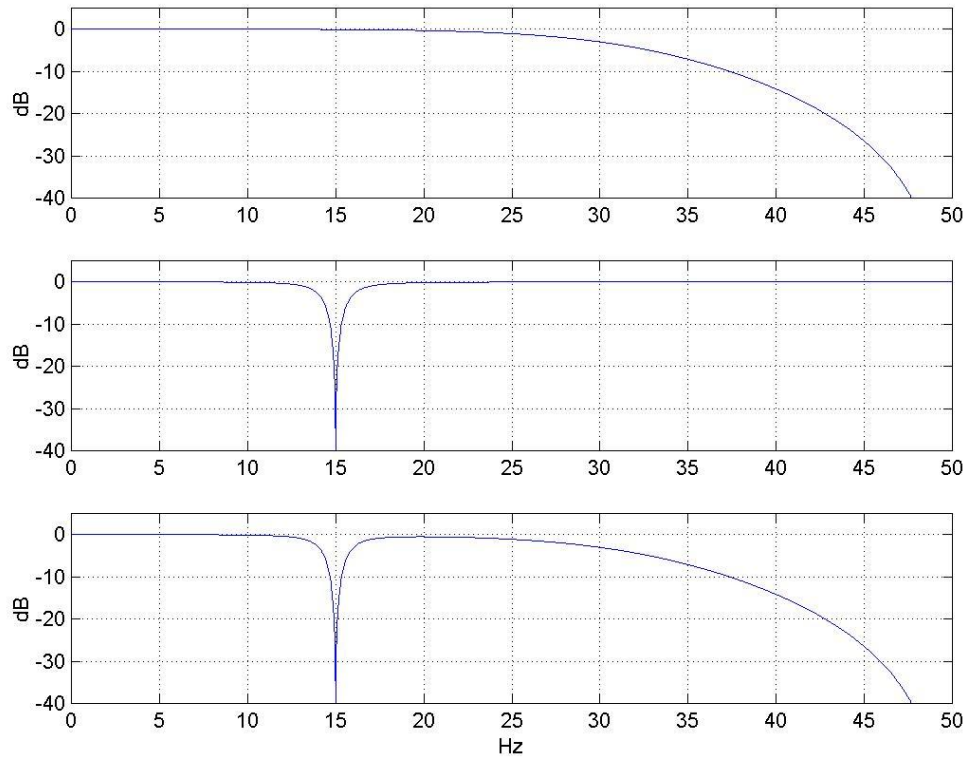


Figure 1. Magnitude response of cascaded lowpass and Band-reject filters.

top: Lowpass response  $|H_1|$     middle: Band-reject response  $|H_2|$     bottom: cascade response

## Appendix Convolution Formulas

The output  $y[n]$  of an FIR filter is the discrete convolution of its coefficients  $b$  with an input  $x[n]$ . We call each of these terms a “sequence”. Discrete convolution is defined as [4]:

$$y[n] = \sum_{k=-\infty}^{\infty} b[k] \cdot x[n - k]$$

$$= \sum_{k=0}^N b[k] \cdot x[n - k]$$

where  $N = \text{length}(b) - 1$  and indices of  $x$  that are negative or greater than  $\text{length}(x) - 1$  are ignored.

Cascading two FIR filters uses the same convolution formula. However, the two sequences are the coefficients of the filters. When multiplying polynomials, the two sequences are the polynomial coefficients. In these cases, since we are not dealing with an input signal  $x$ , we replace  $x$  with another symbol. Thus the convolution of two discrete sequences  $b$  and  $d$  is:

$$y[n] = \sum_{k=0}^N b[k] \cdot d[n - k]$$

Where, again,  $N = \text{length}(b) - 1$ . Convolution is commutative, so we can also write the above equation as:

$$y[n] = \sum_{k=0}^M d[k] \cdot b[n - k]$$

where  $M = \text{length}(d) - 1$ .

## References

1. Robertson, Neil, <https://www.dsprelated.com/showarticle/1055.php>
2. Smith, Julius O., Mathematics of the Discrete Fourier Transform (DFT), [https://www.dsprelated.com/freebooks/mdft/Polynomial\\_Multiplication.html](https://www.dsprelated.com/freebooks/mdft/Polynomial_Multiplication.html)
3. Robertson, Neil, <https://www.dsprelated.com/showarticle/1131.php>
4. Lyons, Richard G., Understanding Digital Signal Processing, 2<sup>nd</sup> Ed., Prentice Hall, 2004, section 5.9.1.