Magnitude squared method to solve a collection of arbitrary functions

Al Clark & Justin Johnson
Danville Signal Processing
2010 comp.dsp Conference

Magnitude Squared Response of H(z):

$$|H(e^{jw})|^2 = H(z)H(z^{-1})|_{z=e^{jw}}$$

Attributes:

- Matches frequency response at ω
- Ignores phase response
- Difficult to factor poles & zeros for high order polynomials

First Order Case:

$$H(z) = b_0 + b_1 z^{-1}$$
$$H(z)H(z^{-1}) = (b_0 + b_1 z^{-1})\ (b_0 + b_1 z)$$
$$H(z)H(z^{-1}) = b_0^2 + b_0 b_{1(} z^{-1} + z) + b_1^2$$
$$H(\omega)^2 = b_0^2 + 2\ b_0 b_1 \cos\omega + b_1^2$$

Pink Noise Filter

- Pink Noise is broadband noise where the power spectral density is inversely proportional to frequency
- Equal Energy per Octave
- Pink Noise = White Noise * (-3dB/octave filter)
- A.K.A 1/f Noise

Analog Design of Pink Noise Filter

- $H(s) = H_1(s)^* \ H_2(s)^* \ H_3(s)^* \ldots. \ H_n(s)$

- Where $H_n(s) = [a_n\ (s+b_n)]/\ [b_n(s+a_n)]$
  and $a_n\ /\ b_n$ is constant ratio

This is just a set of cascaded shelf filters where the poles and zeros are spaced logarithmically.

At low frequencies (below the first pole), the frequency response is flat (0dB)

At high frequencies (above the last pole), the frequency response falls at –6dB/oct (assuming 1 extra real pole)

The "pink" response is approximately –3dB/oct between the highest and lowest poles.

Digital Design of Pink Noise Filter

- $H_{pink}(z) = H_1(z) * H_2(z) * H_3(z) * \ldots. H_n(z)$

Similar to the Analog Method where real poles and zeros are spaced logarithmically.

Assume a single pole filter where

$H(z) = (1-a)/(1- az^{-1})$
$H(z)H(z^{-1}) = [(1-a)(1-a)]/[(1- az^{-1})(1- az)]$

Since $H(0) = 1$, $H(\omega)^2 = ½$

$2(1-2a+a^2) = 1-a(z^{-1}+z) + a^2$
$2-4a+2a^2 = 1-2a \cos \omega + a^2$
$a^2 + a(2 \cos \omega - 4) + 1 = 0$

$a = 2 - \cos \omega - \mathrm{sqrt}((\cos \omega - 3)(\cos \omega - 1))$

Create pole-zero pairs where ω_p/ω_z is a constant ratio:

$H_{pz}(z) = [(1-a_p)(1-a_z z^{-1})]/ [(1-a_z)(1-a_p z^{-1})$

The last stage omits the real zero

$H_p(z) = (1-a_p/(1-a_p z^{-1})$

8 pole – 7 zero example:

Sampling Rate: 204.8 kHz

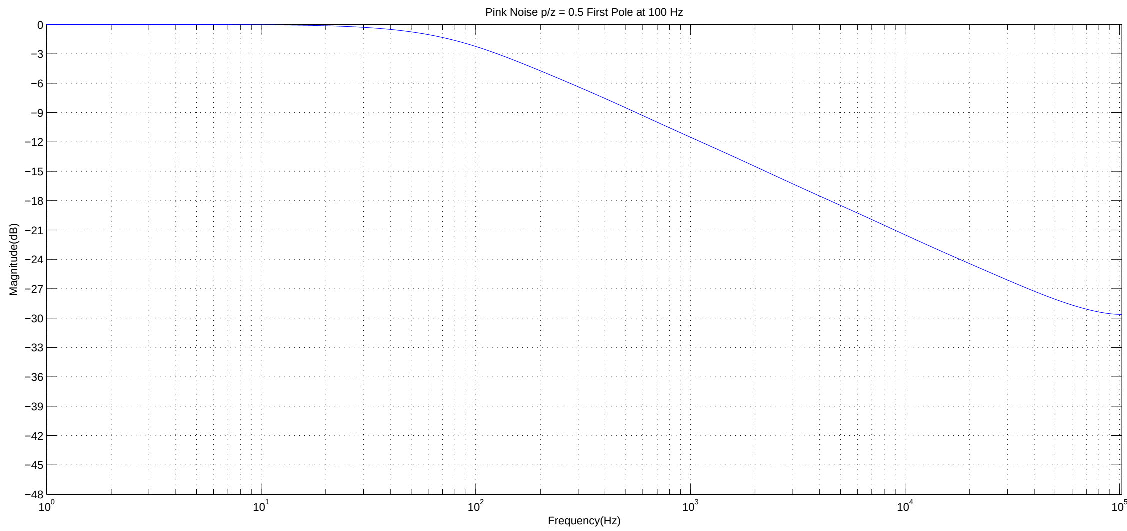
$\omega_p/\omega_z = 1/2$

Poles:

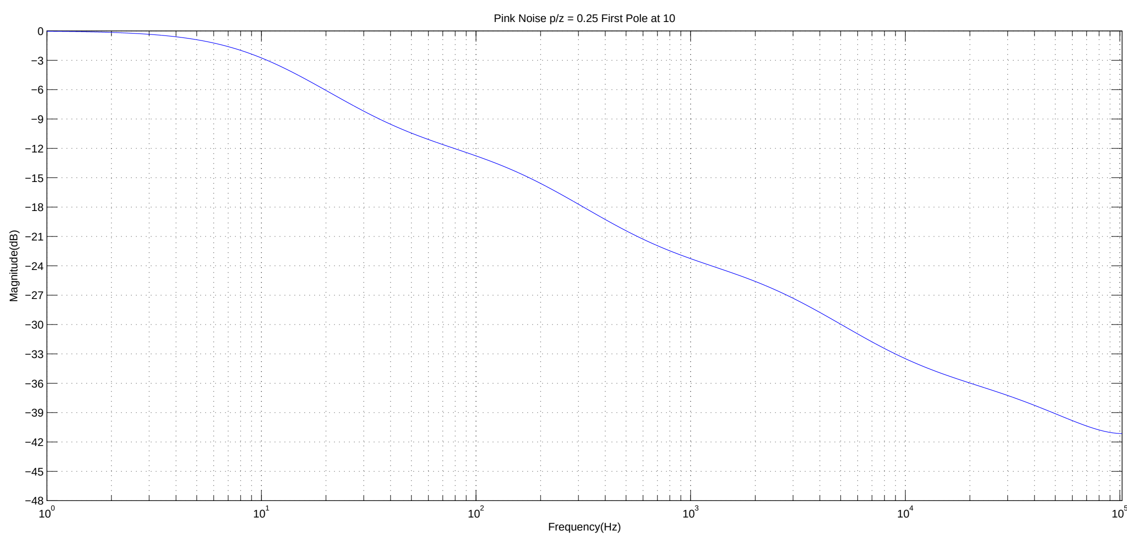
- ω_p = 6.25, 25, 100, 400,1.6k, 6.4k, 25.6k, 102.4k

Zeros:

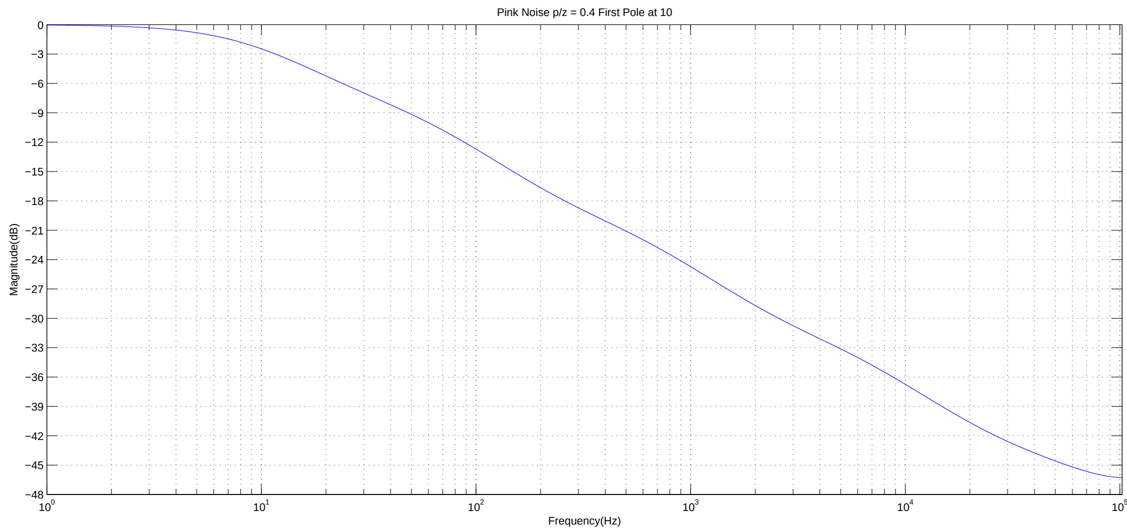
- ω_z = 12.5, 50, 200, 800, 3.2k, 12.8k, 51.2k



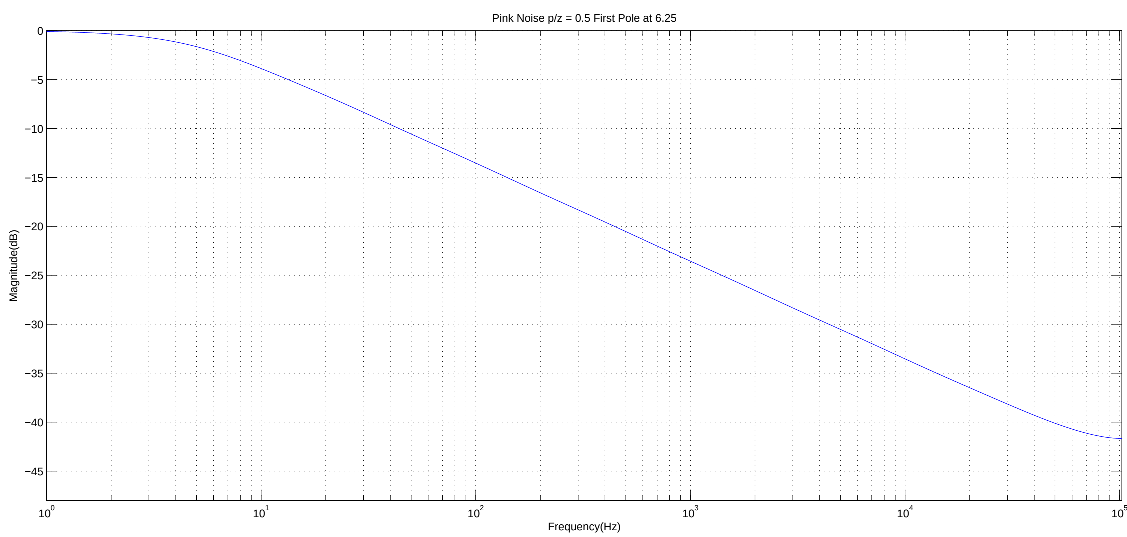
This filter starts at Nyquist (102.4k) and places 6 poles and 5 zeros descending on octave intervals. The lowest frequency pole is 100 Hz. The low frequency response is not very good, but the error is very good in the mid frequencies.



This filter starts at 10Hz (Fs = 204.8kHz) and places 4 poles and 3 zeros at 2 * octave intervals. The low frequency response extends to 10 Hz but there is noticeable ripple in the passband



This filter starts at 10Hz (Fs = 204.8kHz) and places 5 poles and 4 zeros at 2.5 intervals. The low frequency response extends to 10 Hz but there is still some noticeable ripple in the passband



This filter starts at Nyquist (102.4k) and places 8 poles and 7 zeros descending on octave intervals. The lowest frequency pole is 6.25 Hz. This is the filter that was described in the text example. The response is very good.

F	W	1-a	sum 1-a	a	-a	Ratio of p/z
100	0.003068	0.003063258	0.003063258	0.996936742	-0.996936742	0.5
200	0.006136	0.006117118	0.500768164	0.993882882	-0.993882882	
400	0.012272	0.012196702	0.00610772	0.987803298	-0.987803298	
800	0.024544	0.024243743	0.251929749	0.975756257	-0.975756257	
1600	0.049087	0.047892692	0.012065594	0.952107308	-0.952107308	
3200	0.098175	0.093438141	0.129129214	0.906561859	-0.906561859	
6400	0.19635	0.177758996	0.022953879	0.822241004	-0.822241004	
12800	0.392699	0.321416022	0.071414858	0.678583978	-0.678583978	
25600	0.785398	0.526602282	0.037607227	0.473397718	-0.473397718	
51200	1.570796	0.732050808	0.051372427	0.267949192	-0.267949192	
102400	3.141593	0.828427125	0.042558312	0.171572875	-0.171572875	
	0	1	0.042558312	0	0	
10	0.000307	0.000306749	0.000306749	0.999693251	-0.999693251	0.25
40	0.001227	0.001226432	0.25011509	0.998773568	-0.998773568	
160	0.004909	0.004896701	0.001224739	0.995103299	-0.995103299	
640	0.019635	0.019442825	0.062991807	0.980557175	-0.980557175	
2560	0.07854	0.075497454	0.004755721	0.924502546	-0.924502546	
10240	0.314159	0.267730532	0.017763088	0.732269468	-0.732269468	
40960	1.256637	0.672623802	0.011947876	0.327376198	-0.327376198	
0	0	1	0.011947876	0	0	

F	W	1-a	sum 1-a	a	-a	Ratio of p/z
10	0.000307	0.000306749	0.000306749	0.999693251	-0.999693251	0.4
25	0.000767	0.000766696	0.400092058	0.999233304	-0.999233304	
62.5	0.001917	0.001915638	0.000766432	0.998084362	-0.998084362	
250	0.00767	0.007640528	0.100311345	0.992359472	-0.992359472	
625	0.019175	0.018991517	0.001905065	0.981008483	-0.981008483	
2500	0.076699	0.073796649	0.025815056	0.926203351	-0.926203351	
6250	0.191748	0.174001878	0.004491868	0.825998122	-0.825998122	
25000	0.76699	0.518997864	0.008654888	0.481002136	-0.481002136	
62500	1.917476	0.775537718	0.006712192	0.224462282	-0.224462282	
0	0	1	0.006712192	0	0	
6.25	0.000192	0.000191729	0.000191729	0.999808271	-0.999808271	0.5
12.5	0.000383	0.000383422	0.500047941	0.999616578	-0.999616578	
25	0.000767	0.000766696	0.000766696	0.999233304	-0.999233304	
50	0.001534	0.001532805	0.500191821	0.998467195	-0.998467195	
100	0.003068	0.003063258	0.001531776	0.996936742	-0.996936742	
200	0.006136	0.006117118	0.250408089	0.993882882	-0.993882882	
400	0.012272	0.012196702	0.003054153	0.987803298	-0.987803298	
800	0.024544	0.024243743	0.125976952	0.975756257	-0.975756257	
1600	0.049087	0.047892692	0.006033375	0.952107308	-0.952107308	
3200	0.098175	0.093438141	0.064570798	0.906561859	-0.906561859	
6400	0.19635	0.177758996	0.01147804	0.822241004	-0.822241004	
12800	0.392699	0.321416022	0.035710853	0.678583978	-0.678583978	
25600	0.785398	0.526602282	0.018805416	0.473397718	-0.473397718	
51200	1.570796	0.732050808	0.025688677	0.267949192	-0.267949192	
102400	3.141593	0.828427125	0.029583839	0.171572875	-0.171572875	
	0	1	0.029583839	0	0	

Weighting Networks

Problem:

Many standard equalization networks and standard weighting curves are defined as analog transfer functions.

It is impossible to exactly match these functions in the digital domain.

Many of these networks include low pass filters with real poles that are too close to Nyquist.

The most popular mapping methodologies such as Bilinear Transform (BZT) and Impulse Invariant have frequency responses that deviate considerably from the standard analog transfer functions.

Examples of Weighting Networks & Equalization Filters defined as H(s):

Acoustics Standards:

- A,B,C Weighting Networks
- Two real poles at 12.2kHz

Audio Standards:

- RIAA Phono EQ
  (Real Pole at 75us, 2122Hz)

- FM Broadcast
  (Real Pole at 75us (North America)
  (Real Pole at 50us (Elsewhere)

Real Poles:

Analog:

$H(s) = \alpha/(s+\alpha)$ with $H(0) = 1$

Bilinear Transform (BZT):

$H(z) = b(1+z^{-1})/(1 - az^{-1})$ where
$a = (1-\alpha)/(1+\alpha), b = \alpha/(1+\alpha)$

Matched Z (MZT) (normalized):

$H(z) = (1-a)/(1 - az^{-1}), H(0) = 1$

Observation:

For the low pass case, the BZT places a zero at z=-1 (Nyquist). This causes the magnitude response to fall much faster than desired.

Matched Z (and Impulse Invariant) does not fall faster enough. A zero at the origin is "lazy". It does not affect that magnitude response.

Therefore, by simple intuition, it is apparent that a better match will exist if a single real zero is placed somewhere between z=0 and z=-1.

OTOH, The poles as predicted by the MZT are excellent choices.

Mapping $H(s) = \alpha/(s+\alpha)$ with MZT & Magnitude Squared

Matched Z of $H(s) = \alpha/(s+\alpha)$:

$H_{MZT}(z) = (1\text{-}a)/(1\text{-}az^{-1})$
where $a = e^{-\alpha/F_s}$

$|H_{MZT}(z)|^2 = (1\text{-}a)(1\text{-}a)/[(1\text{-}az^{-1})(1\text{-}az)]$
$|H_{MZT}(z)|^2 = (1\text{-}a)(1\text{-}a)/(1\text{-}2a\cos(\omega/F_s)+a^2)$

Determine $H(z) = H_{FIR}(z)/ H_{MZT}(z)$

We need to determine $H_{FIR}(z)$. If $H_{FIR}(z)$ is a single order filter then we can match at two different ω where

$k_w = k_S/k_{MZT} = |H_{MZT}(z)|^2$

$k_w = k_S/k_{MZT} = |H_{MZT}(z)|^2$

At $\omega = 0$, $k_S = 1$ & $k_{MZT} = 1$, therefore

$k_W = 1$

If $H_{FIR}(z) = (1\text{-}bz^{-1})/(1\text{-}b)$ then

$|H_{FIR}(z)|^2 = (1\text{-}2b\cos(\omega/F_s) + b^2)/[(1\text{-}b)(1\text{-}b)]$

Evaluate k_S & k_{MZT} at some arbitrary ω

$k_w = (1\text{-}2b\cos(\omega/F_s) + b^2)/[(1\text{-}b)(1\text{-}b)]$

Rearranging yields:

$b^2 - [2(\cos(\omega/F_s) - k_w)/(1 - k_w)]b + 1 = 0$

If $\beta = 2(\cos(\omega/F_s) - k_w)/(1 - k_w)$

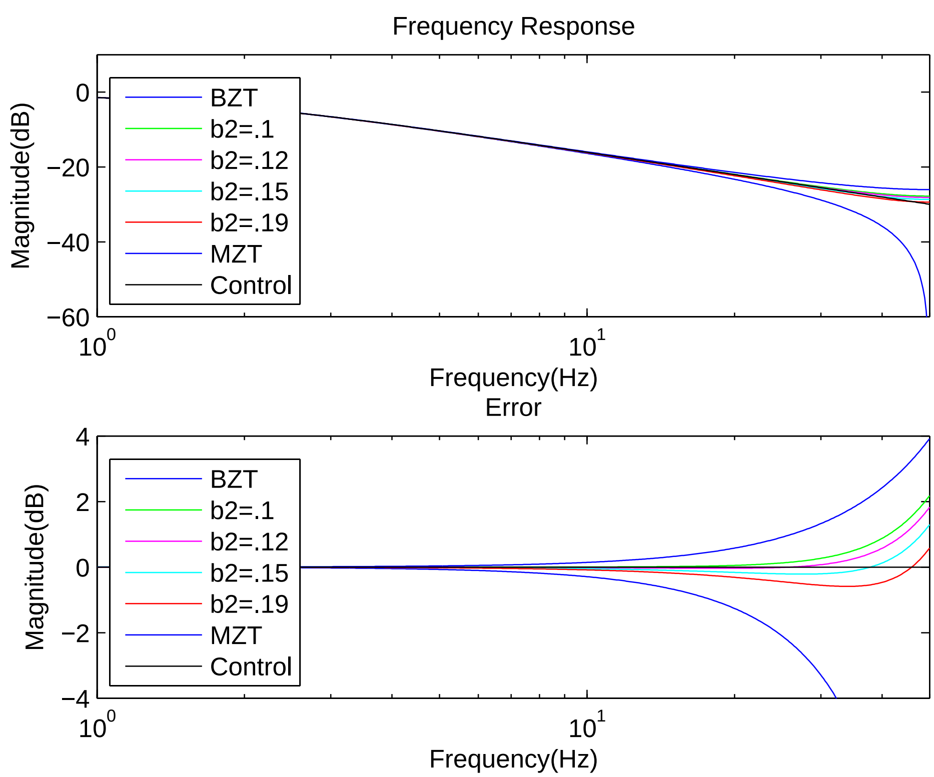
Then $b = (-\beta - \mathrm{sqrt}(\beta^2 - 4))/2$

Selecting ω:

Since the desired correction is needed most between ω_p and Nyquist, ω should be in this range.

Typically $\omega = 3\omega_p$ is a good choice provided it is not close to Nyquist

Typical values for b will be .10 to .15



Second Order Corrections:

If $H_{FIR}(z) = b_0 + b_1 z^{-1} + b_2 z^{-2}$ then

$$|H_{FIR}(z)|^2 = b_0^2 + b_1^2 + b_2^2 + 2b_0 b_1 \cos \omega + 2b_0 b_2 \cos 2\omega + 2b_1 b_2 \cos \omega$$

We now compute the filter by matching three points. This is difficult to factor for arbitrary ω.

Gunness & Chauhan found roots given

$$\omega_0 = 0\,, \omega_1 = \omega,\ \omega_2 = \pi - \omega$$

Their excellent paper focuses on biquadratic filter sections (bell filters) and many of the ideas in this paper benefited from their work.

Minimum phase solutions:

Given:

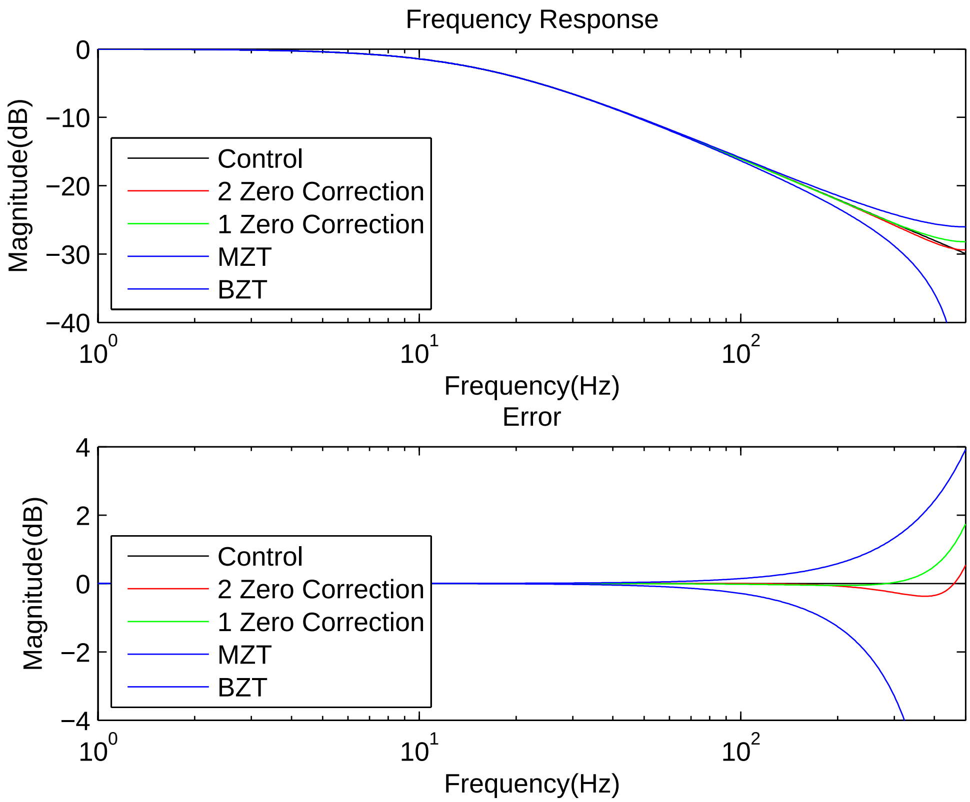
$$k_0 = H(0),\ k_1 = H(\omega),\ k_2 = H(\pi-\omega)$$

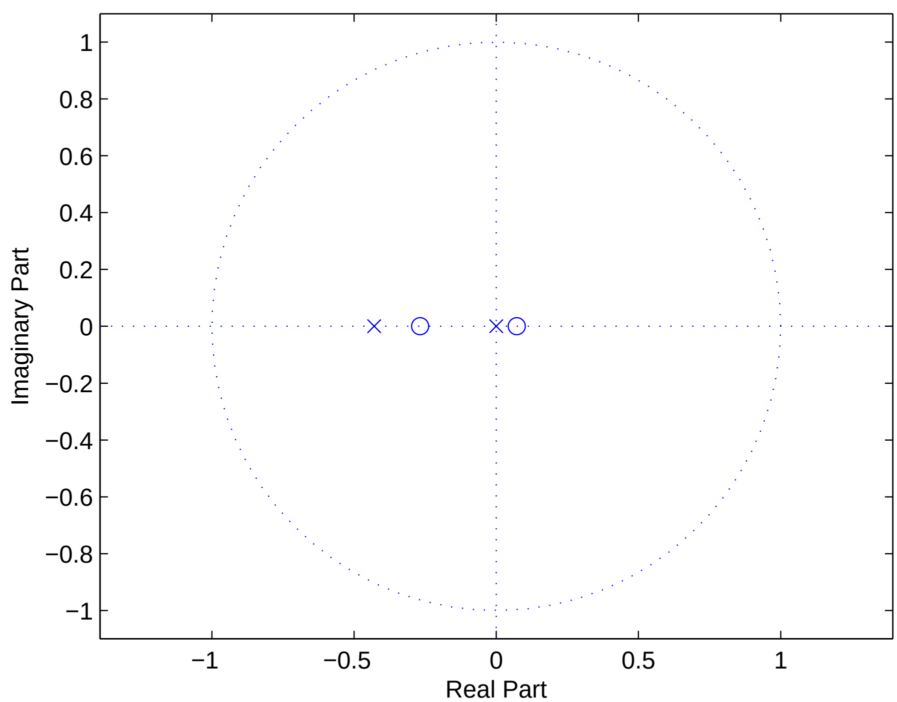
$$a = (k_1^2 - k_2^2)/4\cos\omega$$
$$\alpha = 2(1-\cos 2\omega)$$
$$\beta = -\alpha\,(k_0 - b_1)$$
$$\gamma = (k_0 - b_1)^2 + b_1^2 + 2(k_0 - b_1)\,b_1 \cos\omega - k_1^2$$

then

$$b_1 = [k_0 - \mathrm{sqrt}(k_0^2 - 4a)]/2$$
$$b_2 = [-\beta - \mathrm{sqrt}(\beta^2 - 4\alpha\gamma)]/2\alpha$$
$$b_0 = k_0 - b_1 - b_2$$

Note: In our examples, $k_0 = H(0) = 1$





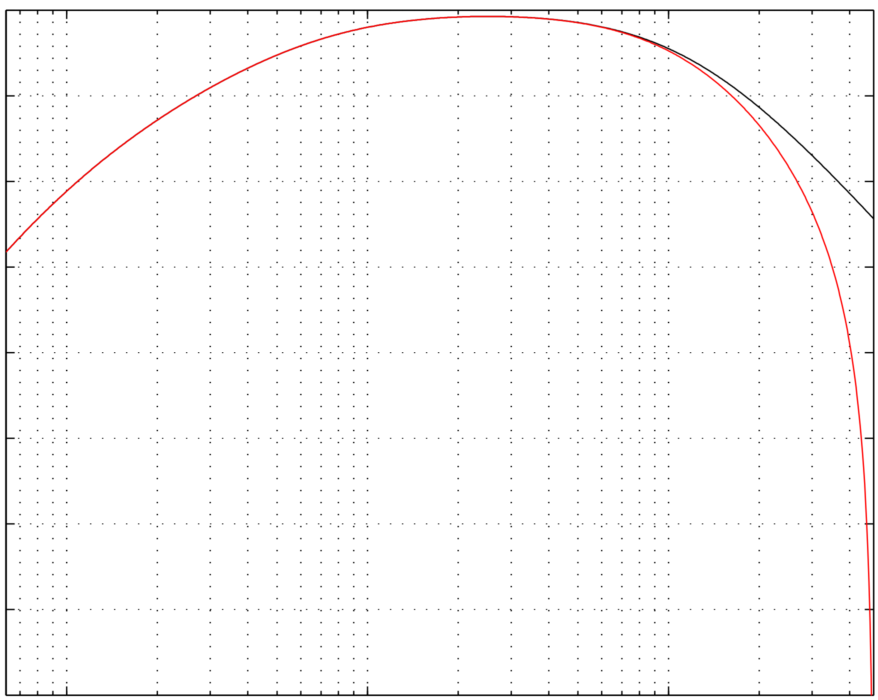
Magnitude squared method to solve a collection of arbitrary functions
Al Clark & Justin Johnson, Danville Signal Processing
2010 comp.dsp conference

A Weighting Network

$$H_A(s) = \omega_4^2 \, s^4/[(s+\omega_1)^2(s+\omega_2)(s+\omega_3)(s+\omega_4)^2]$$

$\omega_1 = 2\pi\,(20.598997) = 129.427315$
$\omega_2 = 2\pi\,(107.65265) = 676.401549$
$\omega_3 = 2\pi\,(737.8223) = 4636.12512$
$\omega_4 = 2\pi\,(12194.22) = 76618.5439$

* Does not include 2.0dB scaling at 1kHz



$H_A(s)$
BZT $F_S = 96000$

Magnitude squared method to solve a collection of arbitrary functions
Al Clark & Justin Johnson, Danville Signal Processing
2010 comp.dsp conference

BZT works well for the high pass sections but fails for the 12.2k poles

Therefore rewrite

$H_A(s) = H_{123}(s)\ H_4(s)$

$H_A(s) = [s^4/[(s+\omega_1)^2(s+\omega_2)(s+\omega_3)]\ *$
$\omega_4^2 /(s+\omega_4)^2$

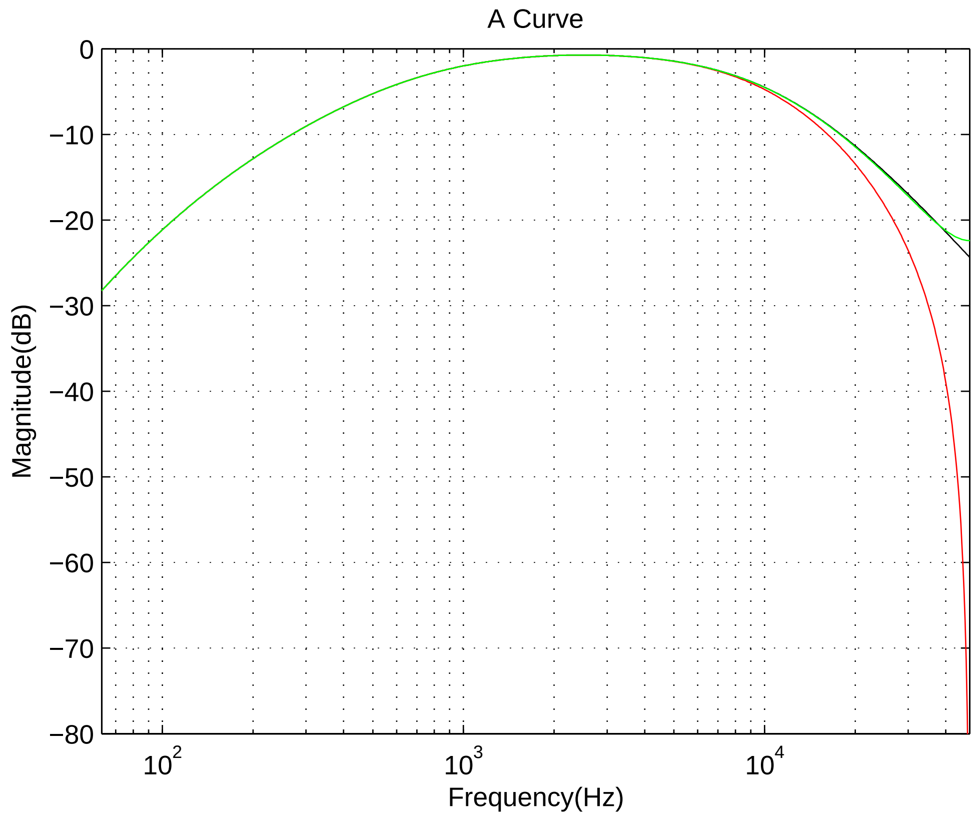
$H_{123}(s)$ -> $H_{BZT123}(z)$ Use BZT

$H_4(s) = [\omega_4 /(s+\omega_4)]\ *\ [\omega_4 /(s+\omega_4)]$

Use two second order $H_{FIR}(z)$ where

1st section uses $\omega_0 = 0, \omega_1 = \omega_p, \omega_2 = \pi - \omega_p$
2nd section uses $\omega_0 = 0, \omega_1 = \pi/3, \omega_2 = 2\pi/3$

* Using two different matching criteria spreads the error
* $\omega_0, \omega_1, \omega_2$ are not the A weight definitions
* ω_p = 76618.5439, the pole location



References:

[1] David Gunness & Ojas Chauhan, Optimizing the magnitude response of matched Z transform filters (MZTi) for Loudspeaker Equalization, Proceedings AES 32nd International Conference 21-23 Sept 2007

[2] Lawrence R. Rabiner & Bernard Gold, Theory and Application of Digital Signal Processing, 1975 Prentice-Hall

[3] Private Correspondence, Michael Arnao, Pink Noise 15 Jan 1997

[4] ANSI S1.4-1983, Specification for Sound Level Meters

Appendix:

```
function [b] = ZeroCorr(w,wp,Fs)
%[b] = ZeroCorr(w,wp,Fs)
%This function will return the z^-1 term to correct the pole matched at w.
%w is the function to put the zero at, wp is the frequency of the pole and
%Fs is the sampling frequency.
%The return is of the form
%
%H(z) = 1 + bz^-1

Ks = wp^2/(w^2+wp^2);
Kmzt = ((1-exp(-wp/Fs))^2)/(1-(2*exp(-wp/Fs)*cos(w/Fs))+exp(-2*wp/Fs));
K=Ks/Kmzt;%H(w)
beta = (2*(cos(w/Fs)-K))/(1-K);
b=(-beta - sqrt(beta^2-4))/2;
```

```
function [b0 b1 b2]=TwoZeroCorr(w,wp,Fs)
%TwoZeroCorr will return the coefficients of the zeros of a two zero
%corrected version of the BZT.  These zeros will be at w and pi-w.  The
%output is
%
% H(z) = b0 + b1 z^-1 + b2 z^-2
%

Ks = wp^2/(w^2+wp^2);
Kmzt = ((1-exp(-wp/Fs))^2)/(1-(2*exp(-wp/Fs)*cos(w/Fs))+exp(-2*wp/Fs));
K=Ks/Kmzt; %H(w)
Ks2 = wp^2/((pi*Fs-w)^2+wp^2);
Kmzt2 = ((1-exp(-wp/Fs))^2)/(1-(2*exp(-wp/Fs)*cos((pi*Fs-w)/Fs))+exp(-2*wp/Fs));
K2=Ks2/Kmzt2;%H(pi-w)
b1 = (1-sqrt(1-4*((K-K2)/4*cos(w/(Fs)))))/2;
alpha = 2*(1-cos(2*w/(Fs)));
beta = 2*(1-b1)*(cos(2*w/(Fs))-1);
gamma = (1-b1)^2+b1^2+2*(1-b1)*(b1)*cos(w/(Fs))-K;
b2 = (-beta-sqrt(beta^2-4*alpha*gamma))/(2*alpha);
b0 = 1-b1-b2;
```