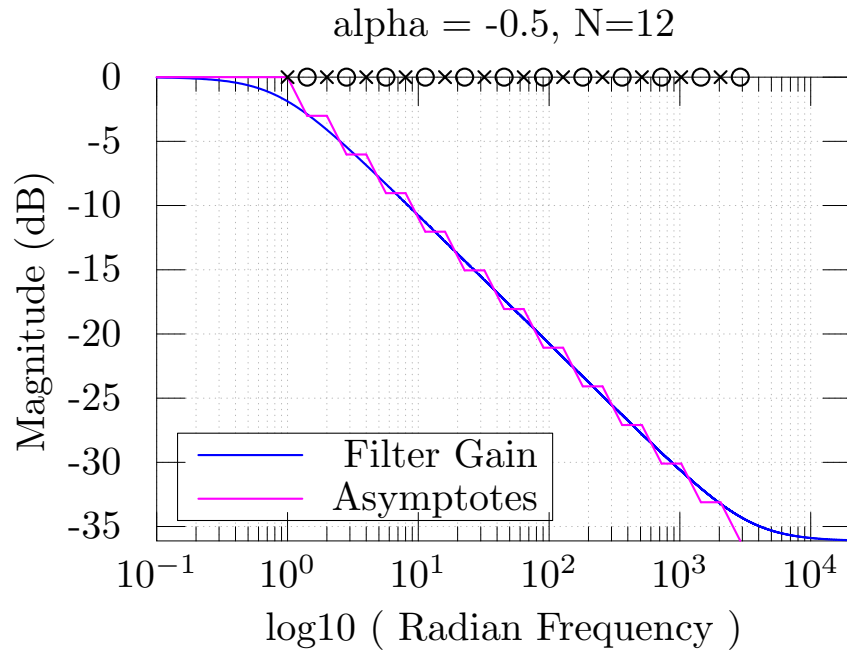


Spectral Tilt Filters*

JULIUS ORION SMITH III and HARRISON FREEMAN SMITH

*Center for Computer Research in Music and Acoustics (CCRMA)
Department of Music, Stanford University, Stanford, California 94305 USA*



Abstract

We derive closed-form expressions for the poles and zeros of approximate fractional integrator/differentiator filters, which correspond to spectral roll-off filters having any desired log-log slope to a controllable degree of accuracy over any bandwidth. The filters can be described as a uniform exponential distribution of poles along the negative-real axis of the s plane, with zeros interleaving them. Arbitrary spectral slopes are obtained by sliding the array of zeros relative to the array of poles, where each array maintains periodic spacing on a log scale. The nature of the slope approximation is close to Chebyshev optimal in the interior of the pole-zero array, approaching conjectured Chebyshev optimality over all frequencies in the limit as the order approaches infinity. Practical designs can arbitrarily approach the equal-ripple approximation by enlarging the pole-zero array band beyond the desired frequency band. The spectral roll-off slope can be robustly modulated in real time by varying only the zeros controlled by one slope parameter. Software implementations are provided in matlab and FAUST.

*Adapted from arXiv.org publication arXiv:1606.06154 [cs.CE].

1 Introduction

The notion of a *fractional derivative* or *integral* is naturally defined in terms of the integration and differentiation theorems for Laplace/Fourier transforms. Let $X(s)$ denote the *bilateral Laplace transform* of $x(t)$:

$$X(s) \triangleq \mathcal{L}_s\{x\} \triangleq \int_{-\infty}^{\infty} x(t)e^{-st} dt,$$

where $s = \sigma + j\omega$ is a complex variable, t typically denotes time in seconds, ‘ \triangleq ’ means “equals by definition,” and we assume $x(t)$ and all of its integrals and derivatives are absolutely integrable and approach zero as $t \rightarrow \pm\infty$. Then the *differentiation theorem* for bilateral Laplace transforms states that

$$\mathcal{L}_s\{x^{(1)}\} \triangleq \mathcal{L}_s\left\{\frac{dx(t)}{dt}\right\} = sX(s)$$

where $x^{(n)}(t)$ denotes the n th derivative of $x(t)$ with respect to t .¹ The proof is quickly derived using integration by parts.²

The *integration theorem* for Laplace transforms follows as a corollary:

$$\mathcal{L}_s\{x^{(-1)}\} \triangleq \mathcal{L}_s\left\{\int_{-\infty}^t x(\tau) d\tau\right\} = \frac{1}{s}X(s)$$

The Laplace transform specializes to the Fourier transform along the $s = j\omega$ axis in the complex plane, where $\omega = 2\pi f$ is *radian frequency* (radians per second), while f denotes frequency in Hz (cycles per second).

1.1 Fractional Derivatives and Integrals

Since the N th successive time derivative/integral of $x(t)$ Laplace-transforms to $s^{\pm N}X(s)$, it follows that the *fractional derivative* of order $\alpha \in (0, 1)$, denoted by $x^{(\alpha)}$, should correspond to the Laplace transform $s^\alpha X(s)$, while the *fractional integral* of order $\alpha \in (0, 1)$ corresponds to $s^{-\alpha}X(s)$. We thus let $x^{(\alpha)}$, $\alpha \in \mathbb{R}$, denote both fractional integrals ($\alpha < 0$) and derivatives ($\alpha > 0$), possibly including both an integer and fractional part. However, since integer $\alpha = \pm N$ corresponds to a repeated ordinary derivative or integral, we will develop methods for $\alpha \in (-1, 1)$ and implement any integer part in the usual way. Expressing j as $e^{j\pi/2}$ so that j^α can be defined as $e^{j\alpha\pi/2}$, we obtain the corresponding Laplace and Fourier transforms for fractional integration or differentiation as

$$\boxed{x^{(\alpha)}(t) \longleftrightarrow s^\alpha X(s) \xrightarrow{s=j\omega} e^{j\alpha\frac{\pi}{2}} \omega^\alpha X(j\omega)}. \quad (1)$$

Thus, a fractional integral or derivative of order α corresponds to a phase shift by $\alpha\pi/2$ and a spectral “tilt” by ω^α . For $\alpha = 1$, we obtain the frequency-response $j\omega$ of a differentiator, and for $\alpha = -1$, the integrator frequency-response $-j/\omega$ is obtained, as required.

¹In engineering, it is more common to use the *unilateral Laplace transform*, in which the integral traverses $[0, \infty)$, and the differentiation theorem becomes $\mathcal{L}_s\{x^{(1)}\} = sX(s) - x(0)$.

²<https://ccrma.stanford.edu/~jos/filters/Differentiation.html>

For integer $\alpha = N \in \mathbb{Z}$, $N \geq 0$, we have, from the *convolution theorem* for the *unilateral* Laplace transform applied to *causal* functions $x(t)$,³

$$\begin{aligned} s^{-N} X(s) &\longleftrightarrow \frac{u(t) t^N}{(N-1)!} * x(t) \\ &= \frac{1}{(N-1)!} \int_0^t x(\tau) (t-\tau)^{N-1} d\tau. \end{aligned} \quad (2)$$

where ‘*’ denotes convolution, and $u(t)$ denotes the Heaviside *unit step* function:

$$u(t) \triangleq \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases}$$

This form was evidently developed originally as *Cauchy’s repeated integral formula*.⁴

$$\begin{aligned} x^{(-N)}(t) &\triangleq \int_a^t \int_a^{\tau_1} \cdots \int_a^{\tau_{N-1}} x(\tau_N) d\tau_N d\tau_{N-1} \cdots d\tau_1 \\ &= \frac{1}{(N-1)!} \int_a^t (t-\tau)^{N-1} x(\tau) d\tau \end{aligned}$$

where a is any finite real number such that $x(t) = 0, \forall t < a$.

The generalization of s^{-N} to s^α for $\alpha \in (-1, 1)$ is quite natural. There is also no problem extending $(t-\tau)^{N-1}$ to $(t-\tau)^{-\alpha-1}$ in Eq. (2), and the lower limit of integration a can be extended (for all practical purposes) as far as needed to the left to encompass the support of $x(t)$. The last piece is generalizing $(N-1)!$ to $(-\alpha-1)!$, which is provided by the *gamma function*⁵

$$\Gamma(t) \triangleq \int_0^\infty \tau^{t-1} e^{-\tau} d\tau$$

which, for positive integers N , becomes $(N-1)$ factorial, *i.e.*,

$$\Gamma(N) = (N-1)!$$

for $N = 1, 2, 3, \dots$. We thus obtain the expression for fractional integrodifferentiation in the time domain as the convolution integral

$$x^{(\alpha)}(t) \triangleq \frac{u(t) t^{-\alpha}}{\Gamma(-\alpha)} * x(t) = \frac{1}{\Gamma(-\alpha)} \int_0^t \frac{x(\tau)}{(t-\tau)^{\alpha+1}} d\tau.$$

This is known as the *Riemann-Liouville differintegral*,⁶ more commonly stated closer to the following form:

$$x^{(\alpha)}(t) = \frac{1}{\Gamma(-\alpha)} \int_a^t \frac{x(\tau)}{(t-\tau)^{\alpha+1}} d\tau,$$

³A function of time $f(t)$ is said to be *causal* if $f(t) = 0$ for all $t < 0$.

⁴https://en.wikipedia.org/wiki/Cauchy_formula_for_repeated_integration

⁵https://en.wikipedia.org/wiki/Gamma_function

⁶<https://en.wikipedia.org/wiki/Differintegral>

where a is an arbitrary fixed base point, and α is any complex number with $\text{re}\{\alpha\} < 0$.⁷

The topic of fractional differentiation and integration falls within the well studied subject of *fractional calculus*⁸ [1]. We will adopt the term “differintegral” from that literature.

1.2 Filter Interpretation

As derived in obtaining Eq. (1) above, every fractional differintegral corresponds to a linear time-invariant *filter* having frequency-response

$$H_\alpha(j\omega) \triangleq e^{j\alpha\frac{\pi}{2}} \omega^\alpha. \quad (3)$$

Since this frequency response is not a rational polynomial in $j\omega$ for non-integer α , there is no exact realization as a finite-order filter [4]. We must therefore settle for a finite-order approximation obtained using a truncated series expansion or *filter design* technique [2, 3]. Many filter-design methods are available in MATLAB,⁹ and several basic design methods, such as `invfreqz`, are also available in the free, open-source, GNU Octave distribution.¹⁰ As far as we know, all filter-based approximations to date have been carried out along these lines.

1.3 Exponentially Distributed Real Pole-Zero Pairs

In contrast to exact integral forms or general-purpose filter designs for fractional differintegrals, we will develop them in *closed form* as *exponentially distributed pole-zero pairs*. (On a log scale, the poles and zeros are *uniformly* spaced.) It appears that such filters approach the Chebyshev optimal approximation (in terms of log-log *slope error*) for any α as the pole-zero density and span along the negative real axis increase.

Any needed integer part $\pm M$ for α can be trivially provided using M zeros or poles at/near the origin $s = 0$ of the complex plane.

The proposed filter structure is furthermore robust for *time-varying* α , because the poles are fixed, and only the zeros need to slide left or right along the negative real axis of the s plane as α is changed. On a log scale, the spacing of the zeros does not change as they are slid left and right.

The fractional order α sets the spacing of the zeros array relative to the poles array along the negative-real axis. For $\alpha = 1/2$, the zeros lie on the midpoints between the poles. For $\alpha = 0$, the array of zeros slides to right so as to cancel all of the poles, leaving the trivial filter $H_0(s) = 1$, as desired. At $\alpha = -1$, all poles are canceled except the first to the left along the negative real axis, leaving a normal integrator $H_{-1}(s) = 1/s$, as desired. At $\alpha = +1$, all poles are canceled except the last, and one zero is exposed near $s = 0$, yielding a normal “leaky differentiator with high-frequency leveling” $H_1(s) = p_N(s + \epsilon)/(s + p_N) \approx j\omega$ for frequencies interior to the interval $\omega \in [\epsilon, |p_N|]$. In the limit as the number of poles N goes to infinity, and as $\epsilon \rightarrow 0$, we obtain the ideal differentiator $H_1(s) = s$, as desired.

⁷Our definition of α is the negative of its usual definition in fractional calculus. We keep it so that α can refer to the *slope* of the Bode magnitude plot, instead of the negative slope. Note also that we allow $\alpha > 0$, but only consider $\alpha \in \mathbb{R}$.

⁸<http://nrich.maths.org/1369>

⁹<https://www.mathworks.com/>

¹⁰<http://www.gnu.org/software/octave/>

1.4 Importance in Audio Signal Processing

In audio signal processing, we often need a spectral shaping filter having a particular *roll-off*, usually specified in decibels (dB) per octave over the audio band. For example, it can be desirable to arbitrarily set the *slope* of the log-magnitude response versus log frequency between the two transition frequencies of a *shelf filter* [4].

A more classical example is the synthesis of *pink noise* from white noise, which requires a filter rolling off -3 dB per octave. Pink noise is also called “ $1/f$ noise”, referring to the roll-off of the *power spectral density* of the noise, which requires a filter for white-noise having a magnitude response proportional to $1/\sqrt{f}$. Many natural processes have been found to be well modeled by $1/f$ noise, such as amplitude fluctuations in classical music, sun spots, the distribution of galaxies, transistor flicker noise, flood levels of the river Nile, and more [5].¹¹

The ideal filter for synthesizing $1/f$ noise from white noise has transfer function

$$H_{-\frac{1}{2}}(s) = \frac{1}{\sqrt{s}} \xrightarrow{s=j\omega} \frac{1}{\sqrt{j}\sqrt{\omega}} = e^{-j\frac{\pi}{4}} \omega^{-\frac{1}{2}},$$

corresponding to $\alpha = -1/2$ in Eq. (3). Since the filter phase is arbitrary when filtering white noise, the filter-design problem can be formulated to match only the power frequency response $|H_{-1/2}(j\omega)|^2 = 1/\omega$ (hence the name “ $1/f$ filters”), thereby obtaining a distribution of poles and zeros yielding a frequency response proportional to $1/\sqrt{\omega}$ for frequencies $\omega = 2\pi f$ in some finite range $f \in [f_{\min}, f_{\max}]$. For audio, we ideally choose $f_{\min} \approx 20$ Hz and $f_{\max} \approx 20$ kHz. Such designs can be found on the Web¹² and in the FAUST distribution.¹³ There are also interesting “Voss-McCartney algorithms” which are essentially sums of white-noise processes that are sampled-and-held at various rates.¹⁴

1.5 Summary of Results

In this paper we derive closed-form expressions for the poles and zeros of spectral roll-off filters having any desired slope to a controllable degree of accuracy. The accuracy desired and the bandwidth over which the approximation holds dictate the order of the filter required, but the basic structure of the filter never varies. The poles and zeros are all real, and they alternate, with exponentially increasing spacing (uniformly spaced on a log scale). A simple initial derivation can be based on *Bode Plot* analysis, as described in the next section.

1.6 Outline of the Remainder

We first review Bode magnitude plots, and then design f^α filters accordingly. We then evaluate the quality of the approximation, and develop a practical design algorithm. Discrete-time filter design using the bilinear transform with frequency prewarping is discussed. Finally, software implementations are given in the matlab and FAUST languages.

¹¹http://123.physics.ucdavis.edu/week_3_files/voss-clarke.pdf

¹²https://ccrma.stanford.edu/~jos/sasp/Example_Synthesis_1_F_Noise.html

¹³faust.grame.fr

¹⁴<http://www.firstpr.com.au/dsp/pink-noise/>

2 Bode Plots

A *Bode Plot* of a filter frequency response $H(j\omega)$ separately graphs the log-magnitude and phase versus log-frequency.¹⁵ We are only concerned here with log-magnitude plots, and will omit consideration of the Bode phase plot, which happen to behave as expected naturally. The usual choice of log-magnitude units is *decibels* (dB) $20 \log_{10} [|H(j\omega)|/R]$ (relative to an arbitrary reference, such as $R = 1$), and the log-frequency axis is typically either in *octaves* ($\propto \log_2(\omega)$) or *decades* ($\propto \log_{10}(\omega)$). Thus, a single pole is said to give a roll-off of -6 dB per octave or -20 dB per decade. Octaves are typical in audio signal processing while decades are typical in the field of automatic control.

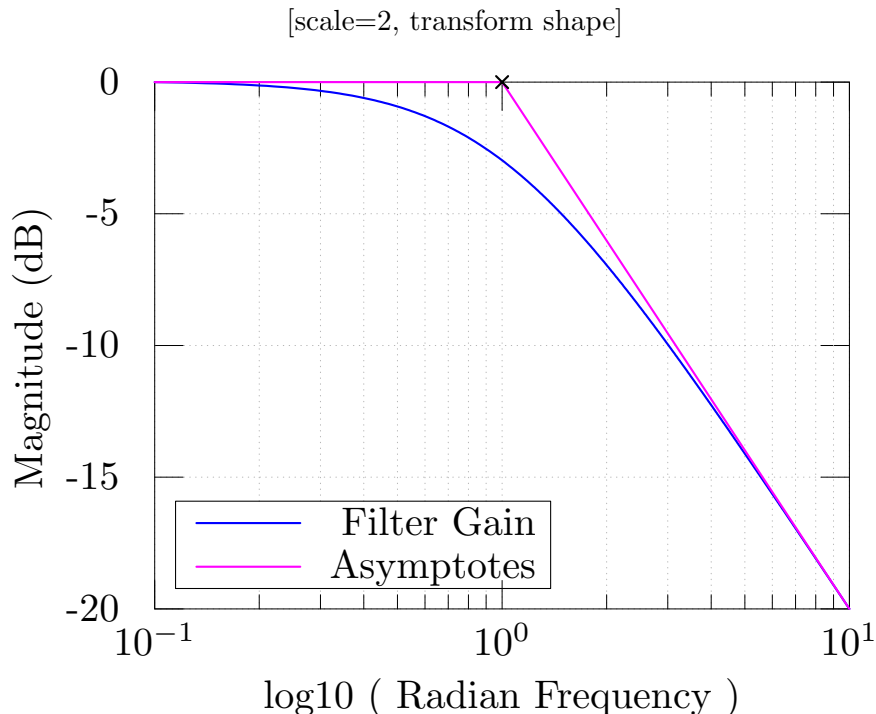


Figure 1: Bode plot and its asymptotes (“stick diagram”) for a single pole at $s = -1$, *i.e.*, the transfer function $H(s) = 1/(s + 1)$.

Figure 1 illustrates the Bode plot and its associated “stick diagram” (comprised of asymptotic gains) for a single pole at $s = -1$. We see that the response is flat for low frequencies, drops to -3 dB at the break frequency $\omega = 1$, and approaches the -20 dB per decade asymptote, reaching the asymptote quite well by one decade above the break frequency at $\omega = 10$.

For a general filter transfer function having N poles p_n and M zeros z_m

$$H(s) = g \frac{\prod_{m=1}^M (s - z_m)}{\prod_{n=1}^N (s - p_n)} \quad (4)$$

the Bode plot can be expressed as

$$\tilde{B}(\tilde{\omega}) = g_1 \log_{b_1} H \left(j b_2^{\log_{b_2}(\omega)} \right) = \tilde{G}(\tilde{\omega}) + j\Theta(\tilde{\omega}),$$

¹⁵https://en.wikipedia.org/wiki/Bode_plot

where $g_1 = 20$, $b_1 = 10$, $\tilde{\omega} = \log_{b_2}(\omega)$, and b_2 is typically 2 or 10. For mathematical simplicity, however, we'll choose instead $g_1 = 1$ and $b_1 = b_2 = e$, giving

$$\tilde{B}(\tilde{\omega}) = \ln H(je^{\tilde{\omega}}) = \tilde{G}(\tilde{\omega}) + j\Theta(\tilde{\omega}),$$

where $\tilde{\omega} = \ln(\omega)$. In this choice of units, N integrators $H(s) = 1/s^N$ give a magnitude roll-off of $-N$ “nepers per neper”, while M differentiators $H(s) = s^M$ give a slope of $+M$ in the Bode magnitude plot

$$\tilde{G}(\tilde{\omega}) = \text{re}\{\tilde{B}(\tilde{\omega})\} = \text{re}\{\ln H(je^{\tilde{\omega}})\}. \quad (5)$$

Our problem is to find poles and zeros of $H(s)$ to minimize some norm of the error

$$\left\| W(\tilde{\omega}) \left[\tilde{G}'(\tilde{\omega}) - \alpha \right] \right\|$$

where $\tilde{G}'(\tilde{\omega}) = d\tilde{G}(\tilde{\omega})/d\tilde{\omega}$ denotes the derivative of $\tilde{G}(\tilde{\omega})$ with respect to $\tilde{\omega}$, α is the desired slope of the log-magnitude frequency-response \tilde{G}' versus log frequency $\tilde{\omega}$, and $W(\tilde{\omega})$ denotes a real, nonnegative weighting function.

As a specific example, for the Chebyshev norm and a uniform weighting $W(\tilde{\omega}) \equiv 1$ between frequencies $\tilde{\omega}_1$ and $\tilde{\omega}_2$, we have

$$\min_H \left\{ \max_{\tilde{\omega} \in [\tilde{\omega}_1, \tilde{\omega}_2]} \left| \tilde{G}'(\tilde{\omega}) - \alpha \right| \right\}.$$

That is, we wish to minimize the worst-case deviation between the desired slope α and the achieved slope $\tilde{G}'(\tilde{\omega})$ over a specific (audio) band $[\tilde{\omega}_1, \tilde{\omega}_2]$.

3 Stick Diagram Design

When designing a filter with a prescribed magnitude response by the Bode “stick diagram” method, we think in terms of poles and zeros “breaking” at certain frequencies. For example, in the term

$$H_n(j\omega) = \frac{-p_n}{j\omega - p_n}$$

which is scaled to have unity gain at $\omega = 0$, the pole is said to “break” at frequency $\omega = |p_n|$. This is easily seen to be the -3 dB point of the term, since

$$H_n(jp_n) = \frac{-p_n}{j(\pm p_n) - p_n} = \frac{1}{1 \pm j}$$

which has magnitude $1/\sqrt{2} \approx -3$ dB. Thus, the gain of the term is approximately constant out to $\omega = |p_n|$, where it reaches its -3 dB, or “half power” frequency, followed by its asymptotic roll-off of -6 dB per octave. A zero term $H_m(j\omega) = (j\omega - z_m)/(-z_m) = 1 - j\omega/z_m$ similarly starts out flat, reaches magnitude-gain $+3$ dB at its break frequency $\omega = |z_m|$, and asymptotically approaches $+6$ dB per octave for $\omega \gg |z_m|$.

The Bode design procedure is then to start at dc ($\omega = 0$) and map out break-frequencies for poles and zeros so as to follow the desired response as closely as desired. Since this tool is commonly

applied in control-system design, there is also usually consideration for the phase plot as well, which has similarly simple behavior.¹⁶

The basic Bode “stick diagram” consists only of straight line segments, each having slope given by some integer number of nepers per neper (or integer multiple of ± 6 dB/octave, etc.), with the knowledge that the actual response will be a smoothed version of the stick diagram, traversing the ± 3 dB points at line-segment intersections corresponding to isolated breaking zeros and poles, respectively.

4 Approximating Arbitrary Slopes

To approximate arbitrary slopes α , we may alternate poles and zeros so that the average slope of the stick diagram equals α .

For example, to achieve a slope of $\alpha = -1/2$ (“half an integrator”), we may start with a pole near $p_1 = -2\pi f_1$, where f_1 is our lowest frequency of interest (nominally 20–40 Hz for audio), which causes the slope to approach -1 . Then, half an octave to the right, *e.g.*, we can locate a zero to cancel the pole’s roll-off, pushing the slope from -1 back toward 0. Continuing in this way, we may locate a pole at each octave point $f_1 2^k$, $k = 0, 1, 2, \dots$, with zeros interlacing at $f_1 2^{k+1/2}$, in order to achieve an *average slope* of $\alpha = -1/2$.

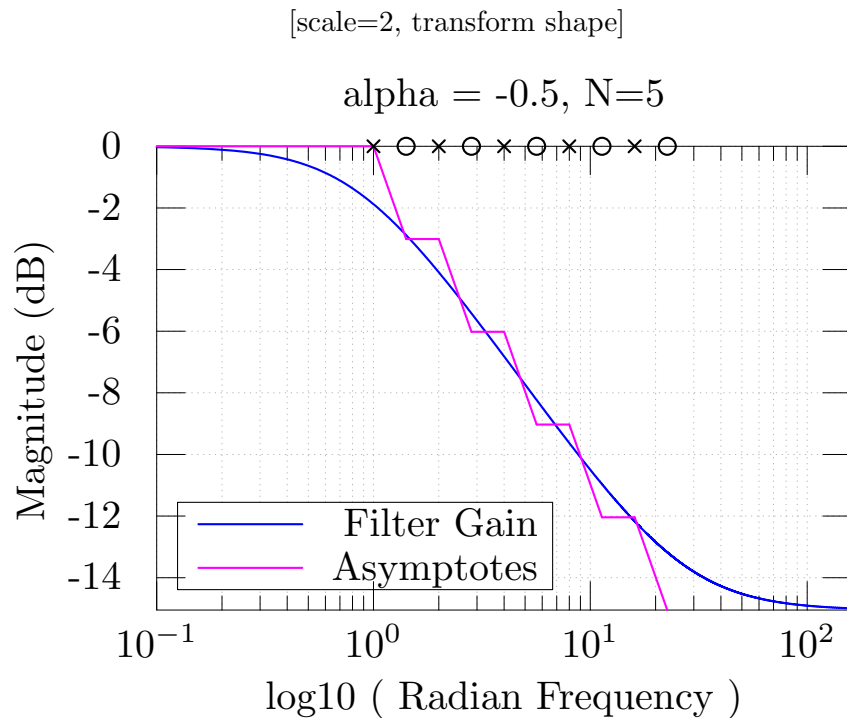


Figure 2: Bode plot and its asymptotes (“stick diagram”) for $N = 5$ poles and zeros, arranged to approximate a $1/f$ power response.

Figure 2 shows the Bode plot and the corresponding stick diagram for $N = 5$ poles located

¹⁶https://en.wikipedia.org/wiki/Phase_margin

to give breakpoints distributed along octaves starting with $\omega = 1$. That is, the poles are at $s = p_n = -r^n$, for $r = 2$ and $n = 0, 2, \dots, N - 1$. To approximate a $1/f$ power-response having slope $\alpha = -1/2$ nepers per neper, we place N zeros at $s = z_m = p_m r^{-\alpha}$, $m = 0, 2, \dots, N - 1$, *i.e.*, shifted half an octave toward higher frequency, interlacing the poles. We see that the response is flat for frequencies below the first break-frequency $\omega = 1$ as before, but the gain drops by less than 2 dB at $\omega = 1$ due primarily to the influence of the upcoming first zero at $\omega = \sqrt{2}$. Between the pole-zero frequencies $\omega = 1$ and $\omega = r^{N-1-\alpha}$, the Bode plot smoothly interpolates the stick diagram which alternates between slopes of 0 and -20 dB per decade, as the poles and zeros break in alternation, yielding an average roll-off of -10 dB per decade, as desired. After the last zero breaks, the response levels off to a final slope of 0. Alternatively, the last zero could be omitted to have a final -20 dB per decade slope, etc. Additionally, we plot the pole symbol ‘X’ and zero symbol ‘O’ along the upper horizontal axis at their corresponding break frequencies. This plot suggests that we may be able to choose N , r , and p_1 to achieve any desired accuracy over any finite band.

More generally, for any desired slope $\alpha \in (-1, 1)$, we place the k th zero on the negative-real axis of the complex s plane at $s = z_k = p_k r^{-\alpha}$, $k = 0, 1, 2, \dots, N - 1$, where $p_k = p_1 r^k$ denotes the k th pole, exponentially distributed along the negative-real axis with spacing ratio $r = p_{k+1}/p_k$, starting at $p_1 = -2\pi f_1$.

Note that $\alpha = 0$ cancels all of the poles with zeros, yielding a constant magnitude frequency response, while $\alpha = -1$ cancels all poles except the first p_1 , leaving a slope of -1 nepers per neper (an integrator) for $\omega \gg \omega_1$. When $\alpha > 0$, the pole-zero sequence starts out from the origin of the s plane with a zero, thereby giving a net positive slope to the Bode magnitude plot. In particular, at $\alpha = 1$, all of the poles are canceled by zeros, leaving behind a single zero at $s = p_1/r$, yielding a slope of $+1$ (differentiator) for $\omega \gg \omega_1/r$. Between these extremes, the poles and zeros interlace asymmetrically to approximate any desired slope α .

As examples of other slopes, Fig. 3 shows a Bode plot analogous to Fig. 2 for the case $\alpha = +1$ (“half a differentiator”), and Fig. 4 shows $\alpha = -0.2$, showing the resulting asymmetric pole-zero layout on a log-frequency scale.

To reduce the maximum error, the interlacing pole-zero pattern can be made more dense, *e.g.*, by placing poles every half octave, or third octave, etc. As an example, Fig. 5 shows the improvement obtained over Fig. 2 by increasing the order from $N = 5$ to 12.

It is not necessary for the slope α of the spectral roll-off to be restricted between -1 and 1 neper per neper. Any number of poles or zeros can be used in the low-frequency region to establish any integer part for the slope, or some number of the regularly spaced poles (zeros) can be skipped before the partial cancellation array of zeros (poles) begins. The subsequent interlacing poles and zeros then only need to set the fractional value of the slope above f_1 .

Figure 6 shows the quality of approximation to the log-magnitude *slope* for $\alpha = -1/2$ and $N = 20$ pole-zero sections. The poles are arbitrarily located at $s = -e^n$, for $n = 0, 1, 2, \dots, N - 1$, yielding log-break-frequencies at $\ln(\omega_n) = n$.

Figure 7 on page 14 shows the log-magnitude slope *error* for the example of Fig. 6. We see that the error is large toward the edges of the pole-zero range. This suggests that we may define a new parameter K which is the number of pole-zero sections to *skip* at the beginning and end of the pole-zero array so that the error is much smaller over the band of interest. Figure 6 marks the band defined by $K = 3$ using a second pair of vertical lines having the same color (see legend).

Figure 8 shows the same slope error as Fig. 7 but for $K = 3$. Zoomed in like this, the error

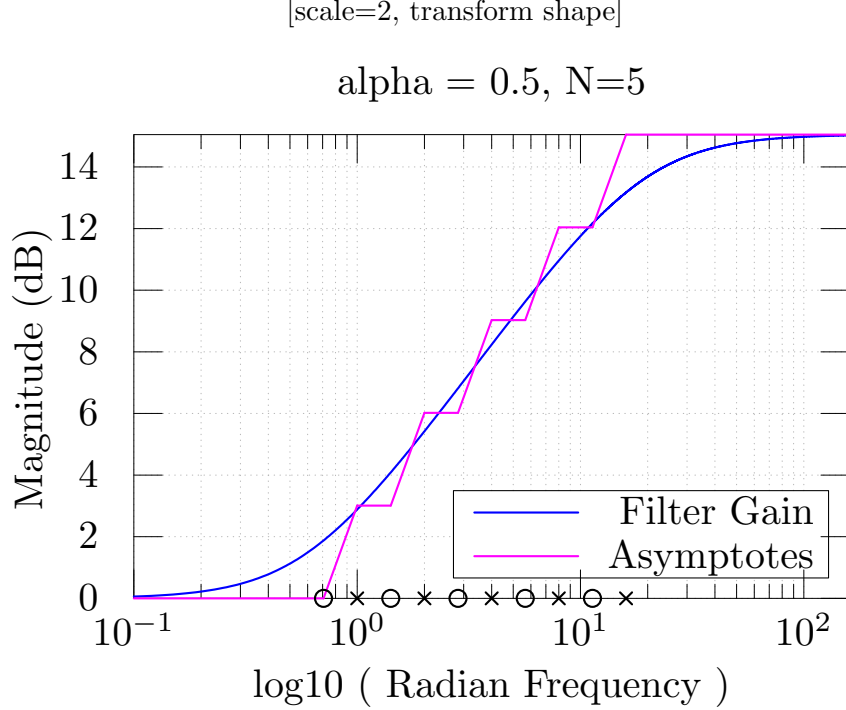


Figure 3: Bode plot and its asymptotes for $N = 5$ poles and zeros, arranged to approximate “half a differentiator” $|H_{1/2}(j\omega)| = \sqrt{\omega}$.

plot strongly suggests that, for an infinite array of pole-zero pairs, the optimal Chebyshev slope approximation is obtained in the limit.

5 Mathematical Approximation

As in Eq. (5), our Bode magnitude plot can be expressed as the real part of

$$\tilde{B}(\tilde{\omega}) = \tilde{H}(je^{\tilde{\omega}}) = \ln H(je^{\tilde{\omega}}) \quad (6)$$

and our problem is to find poles and zeros of $H(s)$ such that

$$\tilde{G}(\tilde{\omega}) \rightarrow \alpha \tilde{\omega}.$$

for $\alpha \in (-1, 1)$. It is convenient to *differentiate* with respect to $\tilde{\omega}$ and formulate the problem as

$$\frac{d}{d\tilde{\omega}} \tilde{G}(\tilde{\omega}) \rightarrow \alpha.$$

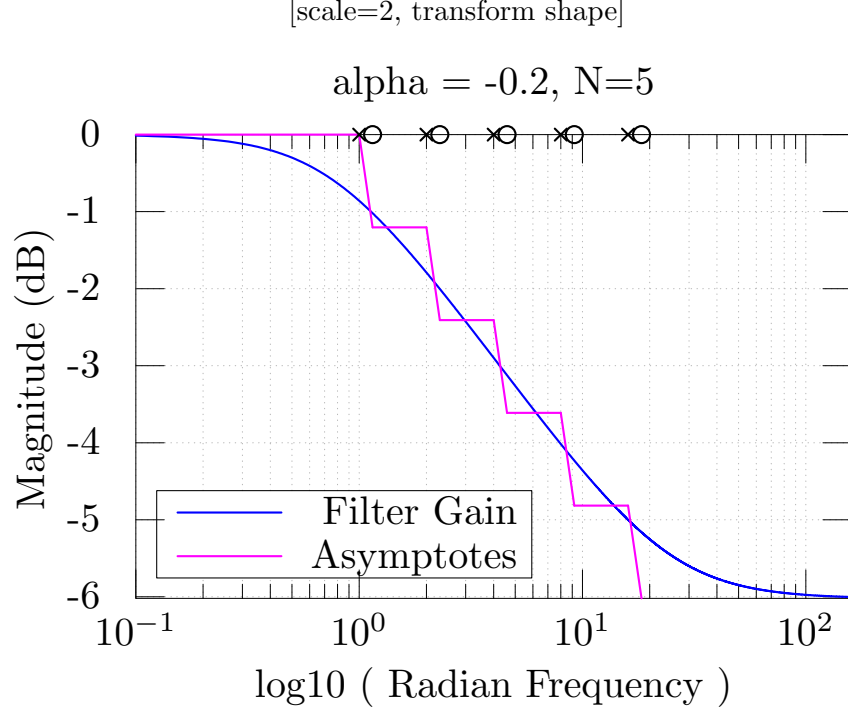


Figure 4: Bode plot and its asymptotes for $N = 5$ poles and zeros, arranged to yield a Bode plot with slope $\alpha = 0.2$ over a chosen frequency range: $|H_{-0.2}(j\omega)| = \omega^{-0.2}$.

Substituting the definition of $H(s)$ from Eq. (4) into Eq. (6) yields

$$\begin{aligned}
 \tilde{B}(\tilde{\omega}) &= \tilde{H}(je^{\tilde{\omega}}) \\
 &= \ln(g) + \left[\sum_m \ln(j\omega - z_m) - \sum_n \ln(j\omega - p_n) \right] \Big|_{\omega=e^{\tilde{\omega}}} \\
 &= \ln(g) + \sum_m \ln(je^{\tilde{\omega}} - e^{ztransform m}) - \sum_n \ln(je^{\tilde{\omega}} - e^{\tilde{p}_n})
 \end{aligned}$$

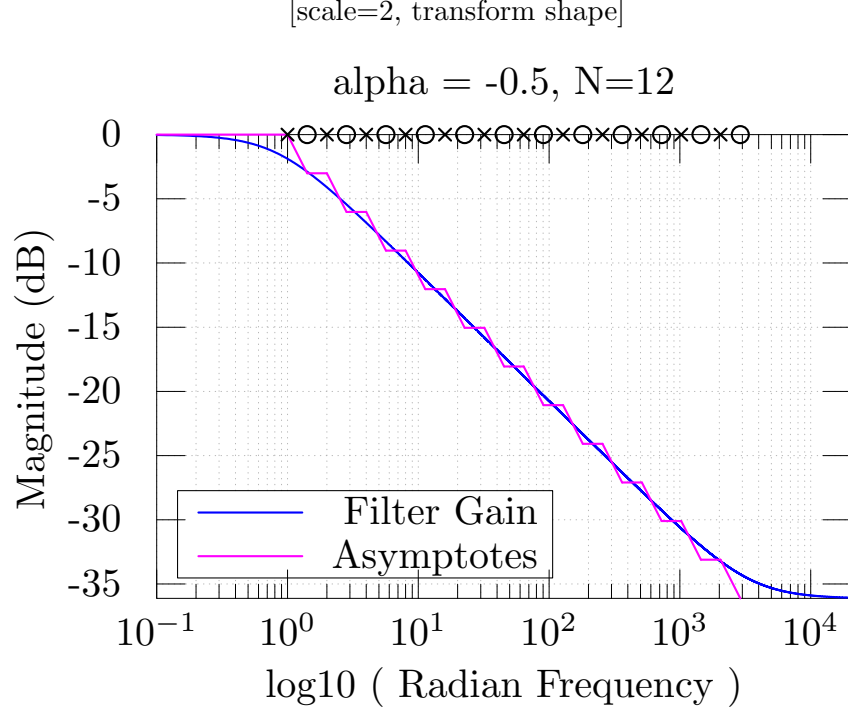


Figure 5: Bode plot and its asymptotes (“stick diagram”) for $N = 12$ poles and zeros, arranged to approximate a $1/f$ filter response.

and differentiating with respect to $\tilde{\omega}$ yields

$$\begin{aligned}
\tilde{B}'(\tilde{\omega}) &= \tilde{H}'(je^{\tilde{\omega}}) \\
&= \sum_m \frac{je^{\tilde{\omega}}}{je^{\tilde{\omega}} - e^{ztransform_m}} - \sum_n \frac{je^{\tilde{\omega}}}{je^{\tilde{\omega}} - e^{\tilde{p}_n}} \\
&= \sum_m \frac{1}{1 + je^{(ztransform_m - \tilde{\omega})}} - \sum_n \frac{1}{1 + je^{(\tilde{p}_n - \tilde{\omega})}} \\
&= \sum_m \frac{1}{1 + je^{\ln(z_m/\omega)}} - \sum_n \frac{1}{1 + je^{\ln(p_n/\omega)}} \\
&= \sum_m \frac{1}{1 + j(z_m/\omega)} - \sum_n \frac{1}{1 + j(p_n/\omega)} \\
&= \boxed{\sum_m \frac{j\omega}{j\omega - z_m} - \sum_n \frac{j\omega}{j\omega - p_n}} \\
&= \sum_m \frac{je^{\tilde{\omega}}}{je^{\tilde{\omega}} - e^{ztransform_m}} - \sum_n \frac{je^{\tilde{\omega}}}{je^{\tilde{\omega}} - e^{\tilde{p}_n}}
\end{aligned}$$

[scale=2, transform shape]

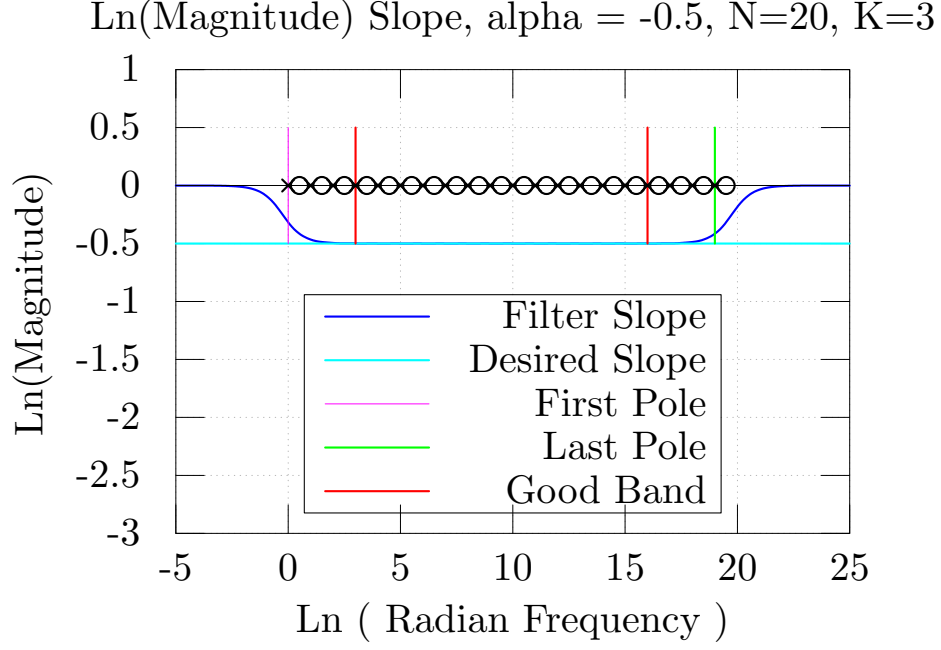


Figure 6: Slope-matching performance for 20 exponentially distributed pole-zero pairs approximating a $1/f$ filter response ($\alpha = -1/2$), with the first and last three poles placed outside the band of interest (“good band”).

The analysis so far has carried along both log-magnitude and phase, since it was no extra work to do so. Taking now the real part to look only at log-magnitude gives

$$\begin{aligned}
 \tilde{G}'(\tilde{\omega}) &= \text{re}\left\{ \tilde{H}'(je^{\tilde{\omega}}) \right\} \\
 &= \text{re}\left\{ \sum_m \frac{j\omega}{j\omega - z_m} - \sum_n \frac{j\omega}{j\omega - p_n} \right\} \\
 &= \text{re}\left\{ \sum_m \frac{j\omega(-j\omega - z_m)}{|j\omega - z_m|^2} - \sum_n \frac{j\omega(-j\omega - p_n)}{|j\omega - p_n|^2} \right\} \\
 &= \text{re}\left\{ \sum_m \frac{\omega^2 - jz_m\omega}{\omega^2 + z_m^2} - \sum_n \frac{\omega^2 - jp_n\omega}{\omega^2 + p_n^2} \right\} \\
 &= \sum_m \frac{\omega^2}{\omega^2 + z_m^2} - \sum_n \frac{\omega^2}{\omega^2 + p_n^2} \tag{7}
 \end{aligned}$$

$$\begin{aligned}
 &= \sum_m \frac{e^{2\tilde{\omega}}}{e^{2\tilde{\omega}} + e^{2z\text{transform}_m}} - \sum_n \frac{e^{2\tilde{\omega}}}{e^{2\tilde{\omega}} + e^{2\tilde{p}_n}} \\
 &= \sum_m \frac{1}{1 + e^{2(z\text{transform}_m - \tilde{\omega})}} - \sum_n \frac{1}{1 + e^{2(\tilde{p}_n - \tilde{\omega})}}. \tag{8}
 \end{aligned}$$

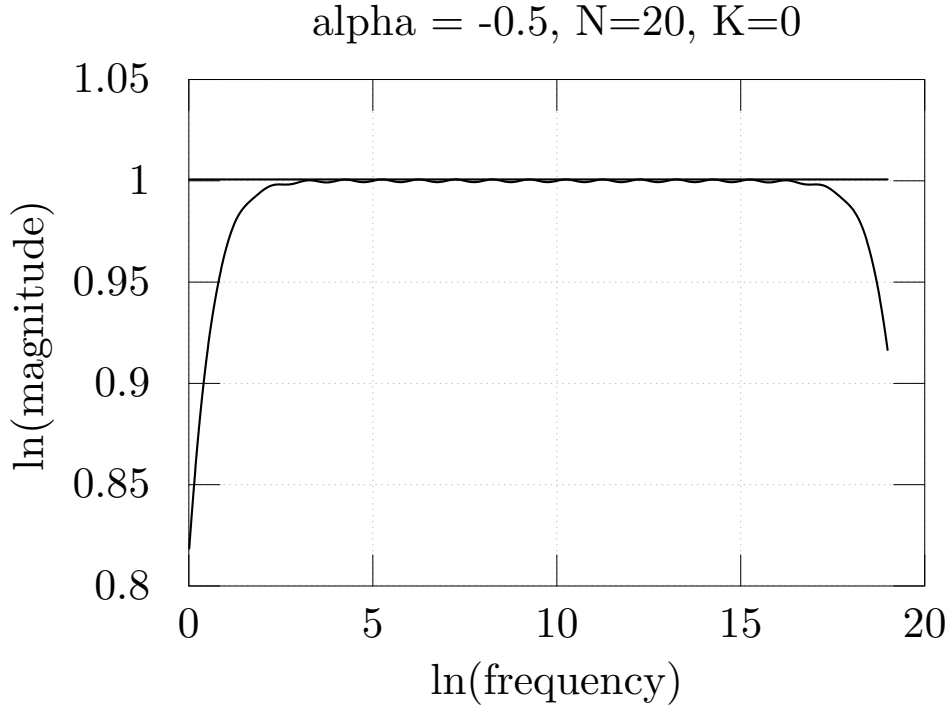


Figure 7: Slope error when pole-zero pairs are used only across the band of interest.

where, thanks to the squaring of all terms in Eq. (7), we may consider only the absolute values of the frequency ω , zeros z_m , and poles p_n :

$$\begin{aligned}\tilde{\omega} &= \ln(|\omega|) \\ ztransform_m &= \ln(|z_m|) \\ \tilde{p}_n &= \ln(|p_n|)\end{aligned}$$

Thus, the normalized basic building block of the Bode log-magnitude plot is given by

$$\tilde{G}'_{z_m}(\ln \omega) = \frac{\omega^2}{\omega^2 + z_m^2}$$

for a zero z_m , and

$$\tilde{G}'_{p_n}(\omega) = -\frac{\omega^2}{\omega^2 + p_n^2}$$

for a pole p_n .

Let b denote either a pole or a zero (the “break frequency” in rad/s due to the pole or zero):

$$\tilde{G}'_b(\ln \omega) = \pm \frac{\omega^2}{\omega^2 + b^2}$$

For $|\omega| \ll b$, the slope is zero, while for $|\omega| \gg b$, it approaches a slope of +1 for a zero and -1 for a pole. At $|\omega| = b$, the slope is $\pm 1/2$.

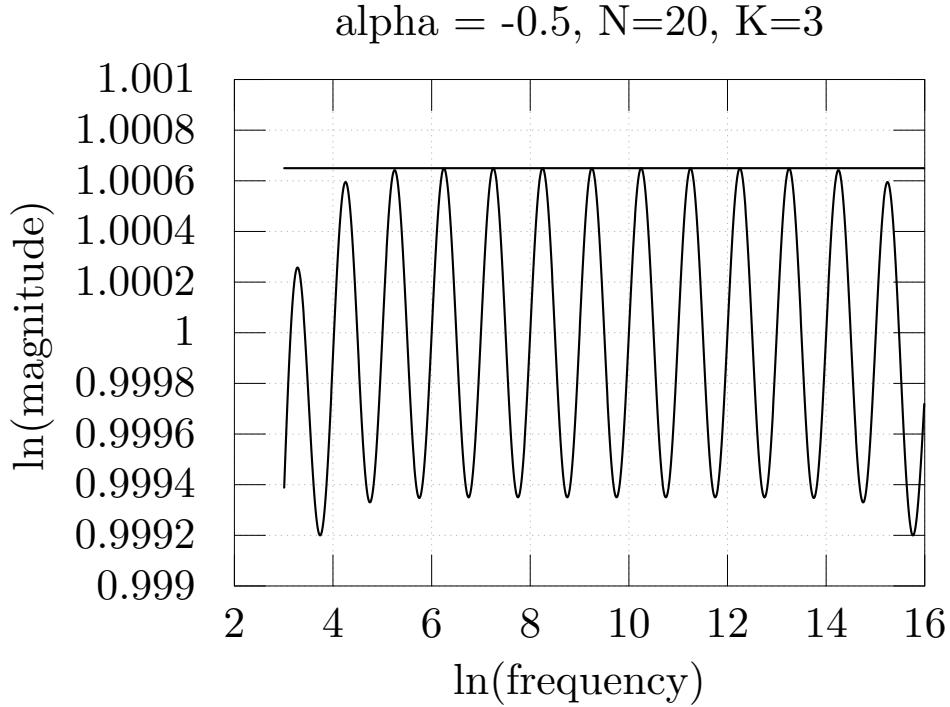


Figure 8: Slope error when $K = 3$ pole-zero pairs are placed both before and after the band of interest.

In log-frequency units, our basic slope building-block can be written

$$\tilde{G}'_b(\tilde{\omega}) = \pm \frac{e^{2\tilde{\omega}}}{e^{2\tilde{\omega}} + e^{2\tilde{b}}} = \pm \frac{1}{1 + e^{-2(\tilde{\omega}-\tilde{b})}}.$$

We may *normalize* this building block to the case $b = 0$

$$\tilde{G}'_0(\tilde{\omega}) = \pm \frac{1}{1 + e^{-2\tilde{\omega}}}$$

so that $\tilde{G}'_b(\tilde{\omega}) = \tilde{G}'_0(\tilde{\omega} - \tilde{b})$. That is, the slope of our desired log-magnitude frequency response is given by sums and differences of arbitrary shifts of \tilde{G}'_0 .

The “stick diagram” approximation of our slope basis function is given by

$$\tilde{G}'_0(\tilde{\omega}) \approx \pm u(\tilde{\omega}),$$

where $u(t)$ denotes the Heaviside unit step function, which steps from 0 to 1 as t goes from negative to positive. Thus, each zero $ztransform_n$ of the transfer function approximately contributes a positive-going step $u(\tilde{\omega} - ztransform_n)$ to the sum of terms making up the overall slope, while each pole \tilde{p}_m contributes $-u(\tilde{\omega} - \tilde{p}_m)$

5.1 Pulse Train View of the Approximation

It is well known that a rectangular pulse $\square_\tau(t)$ of width τ can be synthesized from two unit steps $u(t)$ as

$$\square_\tau(t) = u(t) - u(t - \tau).$$

We can furthermore create a periodic pulse train with period t_0 by means of periodic replication of \square_τ :

$$\square\square(t; \tau, t_0) = \sum_{n=0}^{\infty} \square_\tau(t - nt_0)$$

The mean of this periodic rectangular-pulse train is $\alpha = \tau/t_0$, adjustable between 0 and 1. Note that α is also the *duty cycle* of the pulse train (ratio of pulse width τ to period t_0).

6 Pole-Zero Placement Algorithms

Applying this ‘‘Bode thinking’’ to the fractional slope problem, we may

1. choose an exponential pole spacing $\Delta_p = |\tilde{p}_{k+1}| - |\tilde{p}_k|$ that will eventually determine our filter order, and
2. set the pole-to-zero spacing

$$\Delta_z = |ztransform_k| - |\tilde{p}_k|$$

so as to achieve the desired duty cycle

$$\alpha = -\frac{\Delta_z}{\Delta_p},$$

where the minus sign appears because we wish $z_k < p_k < 0$ which gives a negative slope, thereby effectively choosing $-\square\square(\tilde{\omega}; \Delta_z, \Delta_p)$ as our stick-diagram pulse train.

We thus choose Δ_p arbitrarily according to how many poles we can afford (or to yield a sufficiently small error), and set

$$\begin{aligned} \Delta_z &= -\alpha\Delta_p = -\alpha \cdot (|\tilde{p}_{k+1}| - |\tilde{p}_k|) \\ \Rightarrow \quad ztransform_k &= \tilde{p}_k - \alpha\Delta_p. \end{aligned}$$

For example, choosing octave spacing for the poles corresponds to $\Delta_p = \ln(2)$. Defining

$$r = \frac{p_{k+1}}{p_k},$$

we may write

$$p_k = p_0 r^k, \quad k \in \mathbb{Z}$$

where $p_0 < 0$ may be arbitrarily chosen on the negative-real axis, and

$$z_k = p_k r^{-\alpha}.$$

We may now assemble the complete transfer function as

$$H_\alpha(s; r, p_0) = \prod_{k=-\infty}^{\infty} \frac{s - p_0 r^{k-\alpha}}{s - p_0 r^k}.$$

We conjecture that $\lim_{r \rightarrow 1} H_\alpha(j\omega; r, p_0) = e^{j\alpha \frac{\pi}{2}} \omega^\alpha$, *i.e.*,

$$\boxed{\lim_{r \rightarrow 1} \prod_{k=-\infty}^{\infty} \frac{j\omega - p_0 r^{k-\alpha}}{j\omega - p_0 r^k} = e^{j\alpha \frac{\pi}{2}} \omega^\alpha} \quad (9)$$

for any real $p_0 < 0$, *i.e.*, that the approximation becomes exact in the limit as the pole-zero density goes to infinity in this configuration.

6.1 Chebyshev Optimality Argument

The log-magnitude approximation error in Eq. (9) is a periodic function of $\ln(\omega)$ and is therefore an “equal-ripple” oscillating error, as required for Chebyshev optimality. Furthermore, each pole-zero pair corresponds to one cycle of this oscillation, with each pole creating a specific negative error-peak, and each zero a specific positive error-peak. Thus, the correspondence between error-peaks and poles-and-zeros is bijective. Considering the degrees of freedom in the magnitude-response of a collection of poles and zeros, there can be no other error peaks. Therefore, the Chebyshev optimal configuration must be as found, provided that the basic Chebyshev approximation theorems hold in this extended setting.

6.2 Truncated Pole-Zero Pairs

In practice, we must choose a specific frequency interval $[f_{\min}, f_{\max}]$ outside of which we do not require a f^α response. We then have a trade-off between the number of poles N required and the approximation error between f_{\min} and f_{\max} . We may choose an error tolerance and determine the number of poles required, or simply minimize the error for a given number of poles.

6.3 Specifying Filter Order and Band of Interest

It is straightforward to solve for the pole-placement ratio r and first pole-frequency f_1 given the desired filter order N and the band of interest $[f_{\min}, f_{\max}]$:

1. Specify the desired number of poles N and approximation frequency-range $[f_{\min}, f_{\max}]$ Hz.
2. Determine the first pole frequency f_1 and pole ratio $r = f_{k+1}/f_k$ by solving

$$\begin{bmatrix} 1 & K \\ 1 & N - K - 1 \end{bmatrix} \begin{bmatrix} \tilde{f}_1 \\ \tilde{r} \end{bmatrix} = \begin{bmatrix} \tilde{f}_{\min} \\ \tilde{f}_{\max} \end{bmatrix}$$

for \tilde{f}_1 and \tilde{r} , where \tilde{x} denotes $\ln(x)$. The integer K is the number of poles occurring before and after the frequency range $[f_{\min}, f_{\max}]$. Numerical experiments indicate that $K = 3$ is a cost-effective choice, with higher values yielding somewhat smaller error in the desired frequency range $[f_{\min}, f_{\max}]$.

3. Then the n th pole is given by

$$p_n = -2\pi f_1 r^{n-1}$$

for $n = 1, 2, \dots, N$. The m th zero is

$$z_m = p_m r^{-\alpha}$$

where $\alpha \in [-1, 1]$ is the desired slope of the Bode magnitude plot (in \ln/\ln units).

As mentioned above, any integer part of the slope can be obtained by preceding the pole-zero array with the desired number of poles or zeros. While we could interchange the roles of poles and zeros to change the sign of the slope, it is better in practice to leave the poles fixed, and only vary the zeros to modulate the slope. Zeros can be dynamically modulated more robustly than poles, so real-time modulation of the spectral slope is best carried out by sliding the array of zeros uniformly to the left and right along the negative real log axis in the s plane. Note that as the array of zeros crosses the array of poles, an integer part is incremented or decremented in the slope α . Thus, there is no practical restriction to $\alpha \in (-1, 1)$.

7 Digitization

The filters considered up to now are all for continuous-time processing. We look now at the effects of *digitization* on the accuracy of filter slope.

The *bilinear transform*¹⁷ digitizes a filter by means of the substitution

$$s = c \frac{1 - z^{-1}}{1 + z^{-1}} \Leftrightarrow z = \frac{1 + s/c}{1 - s/c} \quad (10)$$

where c is some positive real constant. That is, given a continuous-time transfer function $H_a(s)$, we apply the bilinear transform by defining

$$H_d(z) = H_a\left(c \frac{1 - z^{-1}}{1 + z^{-1}}\right)$$

where the “ d ” subscript denotes “digital,” and “ a ” denotes “analog.”

Denoting the continuous-time radian-frequency axis by ω_a and the discrete-time radian-frequency axis by ω_d , we find the frequency-warping of the bilinear transform to be

$$j\omega_a = c \frac{1 - e^{-j\omega_d T}}{1 + e^{-j\omega_d T}} = jc \tan\left(\frac{\omega_d T}{2}\right),$$

where T denotes the discrete-time sampling interval in seconds. Thus, we may interpret c as a *frequency-scaling* constant. At low frequencies, $\tan(x) \approx x$, so that $\omega_a \approx c\omega_d T/2$ at low frequencies, leading to the typical choice of $c = 2/T = 2f_s$, where f_s denotes the sampling rate in Hz. However, c can be chosen to map exactly any particular interior frequency $\omega_a \in (0, f_s/2)$.

In our problem, we have complete control over the pole frequencies. We may therefore *prewarp* the pole locations so that they map to an exact geometric progression. We will therefore choose c

¹⁷https://ccrma.stanford.edu/~jos/pasp/Bilinear_Transformation.html

in the bilinear transform to exactly map the break-frequency f_1 of our first pole (or zero if starting out with a zero):

$$c = \frac{2\pi f_1}{\tan(\pi f_1 T)}$$

Next, we alter our s -plane pole frequencies f_k for $k = 2, 3, \dots$ to compensate for the frequency-warping of the bilinear transform:

$$\hat{f}_k = c \frac{\tan(\pi f_k T)}{2\pi} = f_1 \frac{\tan(\pi f_k T)}{\tan(\pi f_1 T)}$$

where \hat{f} denotes the *prewarped* version of f . Since the digital bandwidth is limited to half the sampling rate $f_s = 1/T$, we limit the number of digital-filter poles to N for which $f_{N+1} \leq f_s/2$ while $f_{N+2} > f_s/2$. This choice leaves a full pole-separation interval for the last zero to traverse as α traverses $[-1, 1]$.

One characteristic of the bilinear transform is that it maps any poles at infinity in the s plane to $z = -1$ in the z plane. This means we should either (1) choose the number of poles and zeros to be equal, so that outside the band of interest the response levels off to a constant magnitude, or (2) have more poles than zeros in the s -plane filter so that the resulting zeros at infinity will map harmlessly to $z = -1$ in the digital filter. Poles at infinity, such as in the ideal differentiator $H(s) = s$, give an unstable digital filter under the normal bilinear transform.

8 Faust Implementation

In the latest FAUST branches, the function `spectral_tilt` is defined in `filters.lib`. The example `examples/filtering/spectralTilt.dsp` invokes `dm.spectra_tilt_demo(N)` which is defined in `demos.lib`. This in turn invokes `fi.spectral_tilt` in `filters.lib` having the following API:

```
// USAGE:
// _ : spectral_tilt(N,f0,bw,alpha) : _
// where
//   N = desired integer filter order
//       (fixed at compile time)
//   f0 = lower frequency limit for
//       desired roll-off band
//   bw = bandwidth of desired roll-off
// alpha = slope of roll-off desired
//         in nepers per neper
//         (ln mag / ln radian freq)
```

9 Summary

We have derived closed-form approximate fractional integrator/differentiator filters $|H(j\omega)| = \omega^\alpha$ as exponentially distributed real pole-zero pairs. The approximation error can be made arbitrarily small by reducing the spacing of the pole-zero pairs and by extending them across a larger log-frequency band than what is being used. The poles are uniformly spaced along the negative-real log-axis of the s plane. The zeros interleave the poles and are spaced identically, but the relative

spacing between the array of poles and the array of zeros is $\alpha \in (-1, 1)$. Arbitrary spectral slopes may be obtained dynamically by sliding the array of zeros relative to the array of poles, without altering their internal spacing. The nature of the log-magnitude slope approximation approaches Chebyshev optimality in the interior of the pole-zero array, approaching conjectured Chebyshev optimality over all frequencies in the limit as the order approaches infinity. Software implementations were provided in matlab and FAUST.

10 Future Work

We conjecture that in the case of an infinite pole-zero array, the optimal Chebyshev f^α filter is obtained. Therefore, it should be possible to get closer to the Chebyshev approximation for finite pole-zero intervals by addressing the “edge effects” due to array truncation. It is of practical interest to find compensation strategies for these edge effects. Otherwise, iterative methods can be used as usual to convert the truncated pole-zero array toward the optimal Chebyshev approximation.

References

- [1] K. Oldham and J. Spanier, *The Fractional Calculus*, New York: Academic Press, 1974.
- [2] T. W. Parks and C. S. Burrus, *Digital Filter Design*, New York: John Wiley and Sons, Inc., June 1987, contains FORTRAN software listings.
- [3] J. O. Smith, *Techniques for Digital Filter Design and System Identification with Application to the Violin*, PhD thesis, Elec. Engineering Dept., Stanford University (CCRMA), June 1983, CCRMA Technical Report STAN-M-14, <https://ccrma.stanford.edu/STANM/stanms/stanm14/>.
- [4] J. O. Smith, *Introduction to Digital Filters with Audio Applications*, <https://ccrma.stanford.edu/~jos/filters/>, Sept. 2007, online book.
- [5] R. F. Voss and J. Clarke, “‘1/f noise’ in music: Music from 1/f noise,” *Journal of the Acoustical Society of America*, vol. 63, pp. 258–263, Jan. 1978.

Appendix: Matlab Software for Bode-Plot Figures

For all the matlab scripts used to generate the figures in this paper, see the matlab subdirectory of this paper.

The following matlab was used to generate Figures 2 through 8:

```
if !exist('dopause'), dopause=0; end

linewidth = 2; % need 4 or more for EPS, 2 fine for TIKZ
bpline = '-b'; % Bode magnitude plot
asline = '--m'; % Bode magnitude plot asymptotes
                % (dashed/dotted lines not available in tikz from (MacPorts) Octave right now,
                % so carefully choose colors to be distinguishable as gray levels)

% Bode magnitude plot for N-pole, N-zero filter approximating f^alpha

%N = 5; % Number of poles
N = 12; % Number of poles
```

```

alphas = [-1/2, 1/2, -0.2]; nalphas = length(alphas);
r = 2; % set closer to 1 for increased accuracy
for ia = 1:nalphas
    alpha = alphas(ia);
    mz = r .^ (-alpha+[0:N-1]); % minus the zeros
    mp = r .^ [0:N-1]; % minus the poles
    wa = [0.1*min(mp):0.01:10*max(mp)]; % radian frequency axis
    Ba = poly(mz) % filter transfer-function numerator B(s)
    Aa = poly(mp) % filter transfer-function denominator A(s)
    gainScale = Aa(end)/Ba(end); % Scale for unity dc gain (s plane)
    Ba = Ba * gainScale;
    Ha = freqs(Ba,Aa,wa);
    pdata = db(Ha); % filter magnitude frequency response in dB
    semilogx(wa,pdata,bpline,'linewidth',linewidth); grid; % Bode plot
    hold on % now draw the asymptotes:
    curDB = 0;
    pw = wa(1);
    slopeNeg = alpha<0;
    if slopeNeg, slope=-6.02; else slope=6.02; end
    for i=1:N
        if slopeNeg, bw=mp(i); else bw=mz(i); end % Head to next break frequency
        semilogx([pw bw],[curDB curDB],asline,'linewidth',linewidth);
        pw = bw;
        % continue to next zero/pole at broken slope:
        if slopeNeg, bw = mz(i); else bw = mp(i); end
        newDB = curDB + slope*log2(bw/pw); % pole i breaks
        semilogx([pw bw],[curDB newDB],asline,'linewidth',linewidth);
        curDB = newDB;
        pw = bw;
    end
    semilogx([pw wa(end)],[curDB curDB],asline,'linewidth',linewidth);

% Plot poles and zeros swung around to jw axis:
semilogx(mz,zeros(1,length(mz)),'ok','linewidth',linewidth);
semilogx(mp,zeros(1,length(mp)),'xk','linewidth',linewidth);
axis tight;
xlabel('log10 ( Radian Frequency )');
ylabel('Magnitude (dB)');
% title('Bode Plot and Stick Diagram');
title(sprintf('alpha = %0.1f, N=%d',alpha,N));
loc = 'southwest';
if slope>0, loc = 'southeast'; end
legend('Filter Gain','Asymptotes','location',loc);
hold off
if dopause, disp('PAUSING'); pause; end
fname = bode_filename('bode-pink',alpha,N);
print(['./tikz/',fname,'.tikz'],'-dtikz'); % for inclusion in LaTeX
end % for ia = 1:nalphas

```