



Audio Engineering Society

Convention Paper

Presented at the 129th Convention
2010 Nov 7–10 San Francisco, CA

The papers at this Convention have been selected on the basis of a submitted abstract and extended precis that have been peer reviewed by at least two qualified anonymous reviewers. This convention paper has been reproduced from the author's advance manuscript, without editing, corrections, or consideration by the Review Board. The AES takes no responsibility for the contents. Additional papers may be obtained by sending request and remittance to Audio Engineering Society, 60 East 42nd Street, New York, New York 10165-2520, USA; also see www.aes.org. All rights reserved. Reproduction of this paper, or any portion thereof, is not permitted without direct permission from the Journal of the Audio Engineering Society.

An Infinite Impulse Response (IIR) Hilbert Transformer Filter Design Technique for Audio

Dan Harris¹, Edgar Berdahl², and Jonathan S. Abel³

¹ Sennheiser Research Laboratory, Palo Alto, CA 94306, USA
daniel.harris@sennheiser.com

² CCRMA, Stanford University, Stanford, CA, 94305, USA
eberdahl@ccrma.stanford.edu

³ CCRMA, Stanford University, Stanford, CA, 94305, USA
jonathan.abel@comcast.net

ABSTRACT

Hilbert Transformers have found many signal processing applications, from single-sideband communication systems to audio effects. IIR implementations are attractive for computational efficiency. In this paper, we present a complete design procedure for an efficient infinite impulse response (IIR) Hilbert transformer filter. We start from a half-band filter design, and show how the poles move as the half-band filter is transformed into summed all-pass filters and then into a Hilbert transformer filter. The design technique is based entirely on pole locations, and creates a numerically robust filter in cascaded first-order allpass form.

1. INTRODUCTION

Hilbert transformers are widely used in signal processing applications. Accurate frequency-domain algorithms for designing Hilbert transforms exist, however these algorithms necessitate expensive FFT computations and block-based processing. Efficient implementations can be realized using IIR filters.

Filter design techniques for designing FIR and IIR Hilbert transformers are well documented in the literature [1], [5], [6]. This paper will focus on IIR designs, and will present a complete design technique. This design technique differs from previous work as it is concerned with only pole locations. The final design produces coefficients for first-order allpass sections, which are desirable for their numerical properties.

1.1. Hilbert Transform Overview

The goal of a Hilbert transform filter is generally to create an analytic signal, a complex signal which has zero negative frequencies, or to create a pair of quadrature-phase signals, which are identical but for a 90 degree phase difference between them. Digital filters achieve this by creating a parallel network including a real path and an imaginary path. These two paths are in quadrature phase, and when summed together, create an analytic signal as follows:

$$x_{analytic} = x_{REAL} + jx_{IMAG}. \quad (1)$$

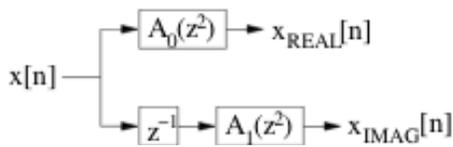


Figure 1: Structure for IIR Hilbert Transformer

In the case of the IIR Hilbert transformer, which is the focus of this paper, it is known that the real path can be implemented as two allpass filters as shown in Figure 2. The design of these allpass filters is the primary problem when designing Hilbert transform filters.

1.2. Design Strategy

Starting with three filter specifications values, the lower frequency and upper frequency limits between which the Hilbert transform phase quadrature should be accurate, and the stopband attenuation which determines the amount the negative frequencies shall be suppressed, the general strategy for designing IIR Hilbert transform filters involves three basic steps:

1. Design an IIR lowpass half-band filter in parallel-allpass form
2. Apply the appropriate transformation of the pole locations to convert the filter into a Hilbert transformer
3. Apply frequency warping to achieve the desired frequency range

As mentioned, the proposed design technique will follow these three steps using pole locations only, and present the result in the structure of cascaded first-order allpass sections.

2. HALF-BAND FILTER DESIGN

There are a number of techniques for designing IIR half-band filters, e.g. [2],[3],[4] and [5], and below, we outline one method.

2.1. Half-Band Filter Specifications

Half-band filters must satisfy three basic requirements:

- The frequency symmetry condition
- The attenuation condition
- The filter order condition

The frequency symmetry condition states that the sum of the passband edge frequency ω_p and the stopband edge frequency ω_s is equal to π ,

$$\omega_p + \omega_s = \pi. \quad (2)$$

The attenuation condition states that the passband ripple δ_1 and the stopband ripple δ_2 of the *squared* magnitude response must be equal:

$$\delta_1 = \delta_2 \quad (3)$$

In terms of the magnitude response, the ripples are related by

$$\delta_s^2 = 4\delta_p(1 - \delta_p). \quad (4)$$

The filter order condition simply states that the filter order must be odd. With these three requirements, a lowpass half-band elliptical filter with transfer function $H_{LP}(z)$ can be designed.

Typical filter design specifications include the passband edge frequency, the stopband edge frequency, the passband ripple, and the stopband attenuation. Due to the halfband constraints outlined above, the passband and stopband ripples are directly related, as are the passband and stopband edge frequencies, and so in fact only one edge frequency and one ripple parameter are

required to fully specify a halfband filter, and by extension a Hilbert transform filter.

2.2. The Parallel-Allpass Form for Half-Band Filters

It turns out that it is possible to realize $H_{LP}(z)$ in the format shown in Figure 2 [5], where $A_0(z^2)$ and $A_1(z^2)$ represent cascades of first-order allpass sections.

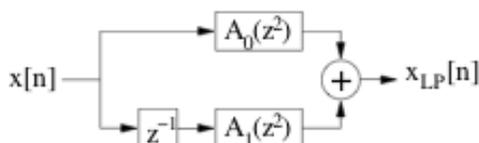


Figure 2: Lowpass filter realized with allpass decomposition

Expressed mathematically,

$$H_{LP}(z) = A_0(z^2) + z^{-1}A_1(z^2). \quad (5)$$

The poles of $A_0(z^2)$ and $A_1(z^2)$, denoted α'_k , can be calculated directly. This process is outlined in [1], and is repeated in the Appendix for completeness. The poles of $A_0(z^2)$ and $A_1(z^2)$, denoted α_k , are given by

$$\alpha_k = \pm j\sqrt{\alpha'_k}. \quad (6)$$

These poles can then be distributed between $A_0(z^2)$ and $A_1(z^2)$ using the pole-interlacing property [1],[5]. This pole-interlacing concept is illustrated in Figure 4, where the blue x's represent the poles of $A_0(z^2)$, or α_{0k} , and the red dots represent the poles of $z^{-1}A_1(z^2)$, notated as α_{1k} . The half-band filter shown in Figure 3 has a passband edge frequency of 8kHz at a sample rate of 40kHz, with a stopband attenuation of 60dB.

2.3. Robust Implementation

It is possible to implement $A_0(z^2)$ and $A_1(z^2)$ using cascaded first-order allpass filter blocks. The transfer function of a first-order allpass filter with allpass coefficient a is as follows:

$$\frac{a + z^{-1}}{1 + az^{-1}}. \quad (7)$$

It has a pole at $-a$ and a zero at $-1/a$. Thus, the cascaded first-order allpass implementation of $A_0(z^2)$ and $A_1(z^2)$

can be defined by the allpass coefficients a_k and thus by the poles α_k , where

$$\begin{aligned} \alpha_{0k} &= -\alpha_{0k} \\ \alpha_{1k} &= -\alpha_{1k} \end{aligned} \quad (8)$$

A structurally passive implementation of the first-order allpass filter is shown in Figure 3. It is passive as long as $|a| < 1$. This structure only requires one memory element, and is robust to coefficient quantization.¹

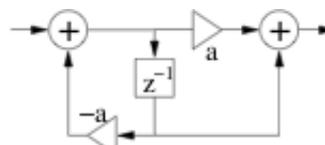


Figure 3: First-order allpass filter implementation incorporating only one sample of memory

The rest of the filter design is achieved by simply transforming the locations of these poles, and hence the values of the first-order allpass coefficients.

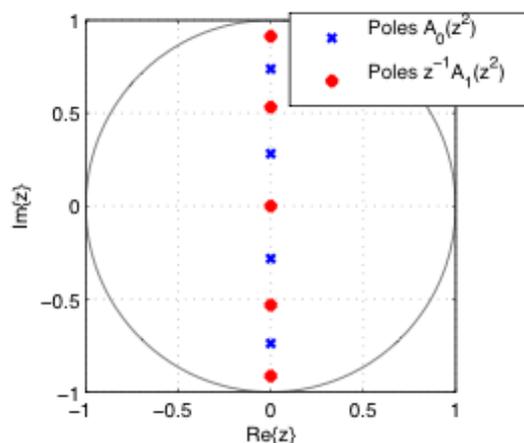


Figure 4: Lowpass half-band filter realized with allpass decomposition

3. CONVERTING TO A HILBERT TRANSFORM FILTER

As mentioned, Hilbert transformers using IIR filters can be implemented in the structure shown in Figure 1, with a real path and an imaginary path. These filters can be created by performing the transformation $z \rightarrow -jz$ on the two half-band filter allpass filters $A_0(z^2)$ and

¹ An even more numerically robust structure can be created using two memory elements, where the zero is applied first, followed by the pole.

$A_1(z^2)$ [5]. This has the same effect as rotating the poles of $A_0(z^2)$ and $A_1(z^2)$, which lie on the imaginary axis, 90° to lie on the real axis, as shown in Figure 5. This can be achieved directly on the pole locations as

$$\begin{aligned} \alpha_{Rk} &= j\alpha_{0k} \\ \alpha_{Ik} &= j\alpha_{1k} \end{aligned} \tag{9}$$

The interlacing of the poles between the two parallel paths remains the same. It must be noted, however, that when performing the transformation in this way, in order to suppress negative frequencies in the final filter implementation, the imaginary path must be multiplied by a factor of -1 if $(N - 1)/2$, with N being the filter order, is odd.

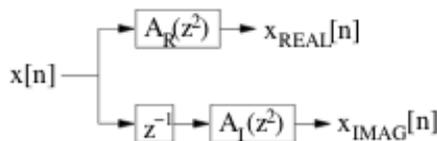


Figure 5: IIR Hilbert Transformer realized with parallel allpass decomposition.

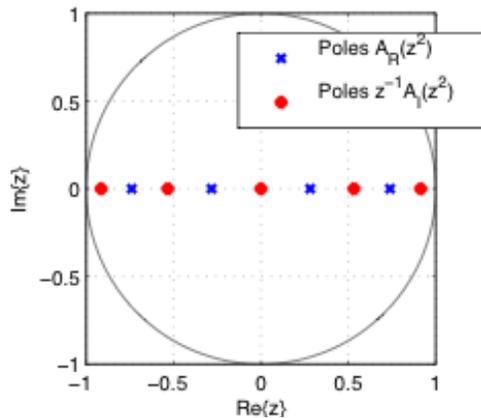


Figure 6: Poles of Hilbert transformer

This has the effect of shifting the frequency response magnitude of the half-band filter by $\pi/2$. Additionally, it has the desired effect that the phase of the two parallel paths differ by $\pi/2$. The frequency response magnitude of a 9th-order Hilbert transformer is shown in Figure 7.

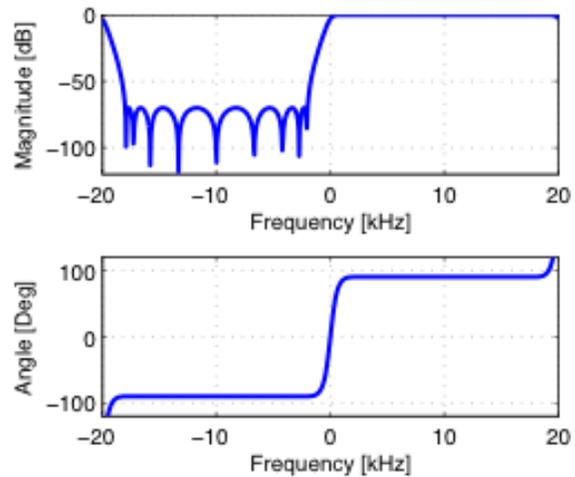


Figure 7: Frequency response of 9th-order Hilbert transformer. Note the suppression of the negative frequencies.

A more detailed look at the phase response of the example filter shows the frequency range in which the Hilbert transformer truly has the desired 90° phase shift. This is shown in Figure 8. It can be seen that the phase shift leaves its approximated region at 2kHz and 18kHz. In fact, it can be shown that the lower frequency limit at which the Hilbert transform’s phase quadrature will be valid is $\pi/2 - \omega_p$, while the upper frequency limit is $\pi/2 + \omega_p$. This corresponds directly to the 2kHz to 18kHz range seen in Figure 8, since the sampling rate was 40kHz.

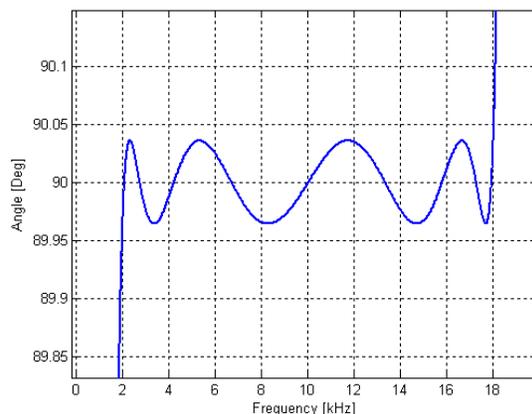


Figure 8: Magnified phase response view

4. FREQUENCY WARPING

Many applications require that Hilbert transforms work accurately in frequency regions whose limits come close to DC. This is particularly true of audio applications, where the frequency region of interest can reach as low as 20Hz. In the absence of further processing, the proposed design technique would require a very high order to achieve such performance. Frequency warping [5][7] can be used to mitigate this limitation, particularly when there are frequency regions for which having accurate phase quadrature are less important. For example, accuracy in the high frequency region can be sacrificed in order to achieve accuracy closer to DC.

Consider that it is desired to create a Hilbert transformer who's phase is approximately 90 degrees between frequencies ω_1 and ω_2 . From [5], it is known that it is necessary to design a "pre-warped" half-band filter with passband and stopband edge frequencies

$$\begin{aligned}\omega_p &= 2 \tan^{-1} \left[\left(\frac{\tan(\frac{\omega_1}{2})}{\tan(\frac{\omega_2}{2})} \right)^{\frac{1}{2}} \right] \\ \omega_s &= 2 \tan^{-1} \left[\left(\frac{\tan(\frac{\omega_2}{2})}{\tan(\frac{\omega_1}{2})} \right)^{\frac{1}{2}} \right]\end{aligned}\quad (10)$$

Once such a half-band filter has been designed, using the techniques outline above and in the Appendix, and it has been converted to a Hilbert filter using the techniques of Section 3, frequency warping can be implemented by applying the following first-order conformal map:

$$z^{-1} \rightarrow \frac{k_B + z^{-1}}{1 + z^{-1}k_B} \quad (11)$$

where

$$k_B = \frac{\beta - 1}{\beta + 1} \quad (12)$$

and

$$\beta = \left[\tan\left(\frac{\omega_1}{2}\right) \tan\left(\frac{\omega_2}{2}\right) \right]^{\frac{1}{2}}. \quad (13)$$

This transformation is order preserving, so that each pole and zero is separately transformed. Figure 9 shows

the new pole locations as a result of the frequency warping. It should be noted that the transformation of (11) must also be applied to the single sample delay shown in the imaginary path. The frequency-warped Hilbert transformer therefor does not have a pole at the origin.

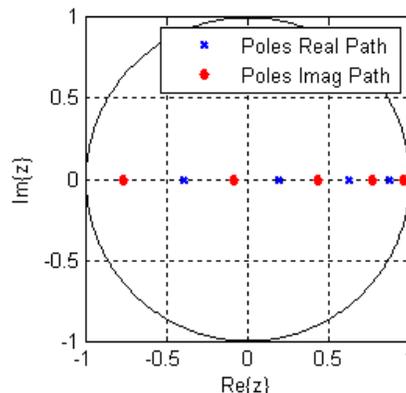


Figure 9: Poles of frequency-warped Hilbert transformer

It can be shown that performing the transformation of (11) on a first-order allpass filter results itself in an allpass filter. Thus, the frequency warping can easily be applied to the cascaded first-order allpass filters by replacing the allpass coefficient in each section as follows

$$a'_{Rn} = \frac{a_{Rn} + k_B}{a_{Rn}k_B + 1} \quad (14)$$

for each section n in the real path, and

$$a'_{In} = \frac{a_{In} + k_B}{a_{In}k_B + 1} \quad (15)$$

for each section n in the imaginary path, where

$$\begin{aligned}a_{Rn} &= -\alpha_{Rn} \\ a_{In} &= -\alpha_{In}\end{aligned}\quad (16)$$

5. EXAMPLE WITH HIGHER ORDER

A third example is now presented. The specifications for the filter are as follows:

$$\begin{aligned} \omega_1 &= 200 \text{ Hz} \\ \omega_2 &= 15 \text{ kHz} \\ A_{st} &= 80 \text{ dB} \\ f_s &= 40 \text{ kHz} \end{aligned} \tag{17}$$

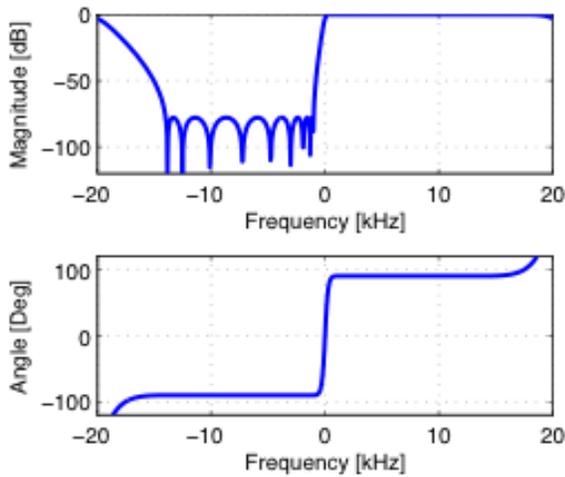


Figure 10: Frequency response of frequency-warped Hilbert transformer

The order of this filter is 13. It can be seen in Figure 12 that the frequency response magnitude rolls off much more steeply near DC, thereby more effectively eliminating negative frequencies, while near Nyquist rolling off much more gradually. Similarly, it can be seen that the phase difference between the imaginary and real paths maintains its 90° (or -270°) approximation closer to DC, but further from the Nyquist limit.

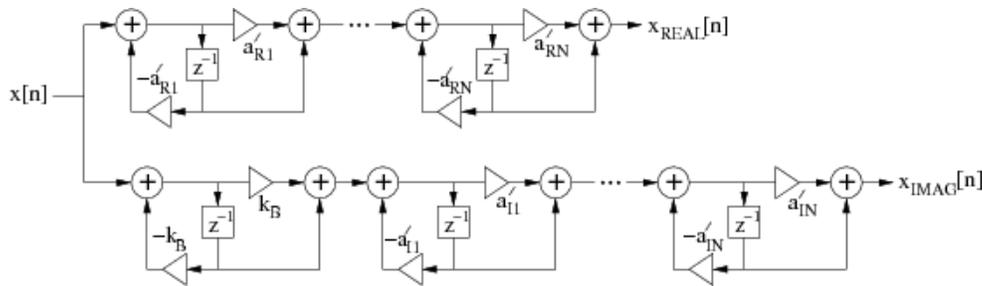


Figure 11: Robust allpass structure with frequency warping

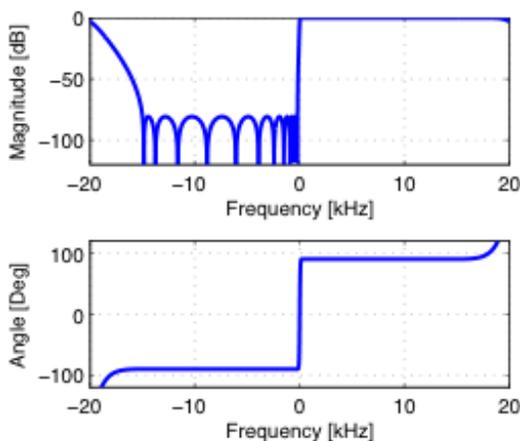


Figure 12: Frequency response for 13th order filter

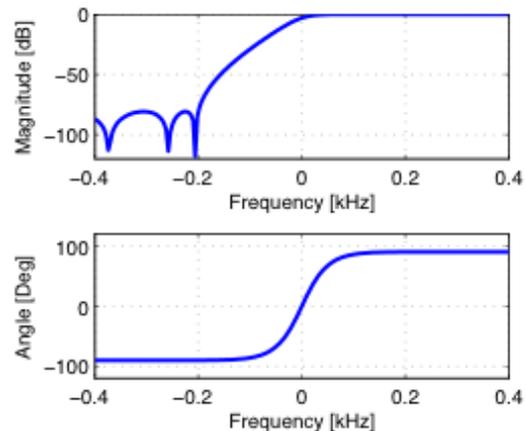


Figure 13: Frequency response for 13th order Hilbert filter zoomed in horizontally near DC

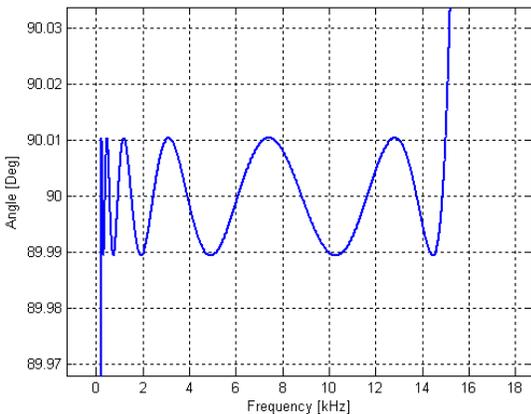


Figure 14: Phase response of 13th order Hilbert filter zoomed vertically to show ripple

6. SUMMARY

The final design process can then be summarized as follows:

1. Select the upper and lower frequency limits, ω_1 and ω_2 , and the stopband attenuation desired for the Hilbert filter
2. Calculate the pre-warped half-band filter specifications using (10).
3. Calculate the poles for a parallel-allpass structure half-band filter using the Appendix section and (6), and apply the pole-interlacing property to divide the poles between the two parallel paths.
4. Rotate the poles by 90° using (9) to convert to a Hilbert transform.
5. Apply the frequency warping using (11)-(16) to transform the pole locations and calculate the final allpass coefficients
6. Implement the filter using the calculated allpass coefficients using the structure shown in Figure 11.

7. Multiply the imaginary path by a factor of -1 if $(N-1)/2$ is odd.

7. ACKNOWLEDGEMENTS

We would like to extend our thanks to Julius O. Smith III and the AES staff for their assistance.

8. REFERENCES

- [1] Sanjit Mitra, *Digital Signal Processing*, McGraw-Hill, 3rd edition, New York, NY, 2005.
- [2] P. Vaidyanathan, P. Regalia, and S. Mitra, "Design of Doubly-Complementary IIR Filters Using a Single Complex Allpass Filter, With Multirate Applications," *IEEE Transactions on Circuits and Systems*, vol. 34, no. 4 (1987 April).
- [3] Miroslav Lutovac and Ljiljana Mili, "Half-Band IIR Filter Design Using MATLAB," *Conference TELEFOR, Yugoslavia*, pp. 329-32, 2000 November.
- [4] Otto Nyari, Tibor Szakall, and Peter Odry, "IIR Half-band Filter Design with TMS320VC33 DSP," *SISY 3rd Serbian-Hungarian Joint Symposium on Intelligent Systems, Subotica, Serbia*, 2005 August 31-September 1.
- [5] Sanjit Mitra and James Kaiser, "Design of Hilbert Transformers," in *Handbook for Digital Signal Processing*, John Wiley and Sons, Hoboken, NJ, pp. 909-931, 1993.
- [6] Andrew Reilly, Gordon Frazer, and Boualem Boashash, "Analytic Signal Generation—Tips and Tricks," *IEEE Transactions on Signal Processing*, vol. 42, no. 11, pp. 3241-3245, 1994 November.
- [7] Julius O Smith, Jonathan Abel, "Bark and ERB Bilinear Transforms", *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 6, 1999 November.

9. APPENDIX

9.1. Calculation of Poles of Half-Band Filters

This section provides the equations for finding the poles of $A_0(z)$ and $A_1(z)$. These equations were presented in [1], and repeated here (with very slight adaptations) for the sake of completeness.

Consider a half-band filter specified by the passband edge frequency, ω_p (in radians) and the stopband attenuation in dB, A_{st} . Calculate the stopband edge frequency

$$\omega_s = \pi - \omega_p \quad (18)$$

and the stopband ripple

$$\delta_s = 10^{\frac{-A_s}{20}}. \quad (19)$$

Then perform the following calculations:

$$r = \frac{\tan\left(\frac{\omega_p}{2}\right)}{\tan\left(\frac{\omega_s}{2}\right)} \quad (20)$$

$$r' = \sqrt{1 - r^2} \quad (21)$$

$$q_0 = \frac{1 - \sqrt{r'}}{2(1 + \sqrt{r'})} \quad (22)$$

$$q = q_0 + 2q_0^5 + 15q_0^9 + 150q_0^{13} \quad (23)$$

and

$$D = \left(\frac{1 - \delta_s^2}{\delta_s^2} \right)^2. \quad (24)$$

From this, the required order of the filter can be estimated as the smallest *odd* integer satisfying

$$N \geq \frac{\log_{10}(16D)}{\log_{10}(1/q)} \quad (25)$$

From here, the poles α'_k for $k = 1 \dots (N - 1)/2$ of the allpass filter sections can be calculated as follows:

$$\lambda_k = \frac{2q^{0.25} \sum_{i=0}^{\infty} (-1)^i q^{i(i+1)} \sin\left(\frac{(2i+1)k\pi}{N}\right)}{1 + 2 \sum_{i=0}^{\infty} (-1)^i q^{i(i+1)} \cos\left(\frac{2ik\pi}{N}\right)} \quad (26)$$

$$b_k = \sqrt{(1 - r\lambda_k^2) \left(1 - \frac{\lambda_k^2}{r}\right)} \quad (27)$$

$$c_k = \frac{2b_k}{1 + \lambda_k^2} \quad (28)$$

and then

$$\alpha'_{k-1} = \frac{2 - c_k}{2 + c_k}. \quad (29)$$

As described in [1], the infinite sum in (24) generally converges after about 5 or 6 iterations. The code used to generate the figures for this paper uses 7 iterations.