Audio Engineering Society

Convention Paper 10016

Presented at the 144th Convention
2018 May 23 – 26, Milan, Italy

Improving the frequency response magnitude and phase of analogue-matched digital filters

John Flynn[1,2] and Joshua D. Reiss[2]

[1]*Balance Mastering, London, UK.*
[2]*Queen Mary, University of London, UK.*

Correspondence should be addressed to John Flynn (`john@balancemastering.com`)

ABSTRACT

Current closed-form IIR methods for approximating an analogue prototype filter in the discrete-domain do not match frequency response phase. The frequency sampling method can match phase, but requires extremely long filter lengths (and corresponding latency) to perform well at low frequencies. We propose a method for discretising an analogue prototype that does not succumb to these issues. Contrary to the IIR methods, it accurately approximates the phase, as well as the magnitude response. The proposed method exhibits good low frequency resolution using much smaller filter lengths than design by frequency sampling.

1 Introduction.

Creating a discrete-time digital filter that approximates a continuous-time analogue prototype is a common filter design problem. Some applications require an approximation that is highly accurate. For example, professional audio engineers value many analogue filters for their aesthetic sound quality [1], [2]. Fields of audio work such as mixing, mastering and forensics have produced commercial digital tools that are extremely accurate models of analogue designs. Elsewhere, telecommunications builds on continuous-time signal theory, where many problems are already solved by analogue filters [3]. Creating a discrete-time filter that approximates a continuous-time filter to the required degree avoids designing directly in the digital domain, an area where research is still relatively young [4].

Traditional design methods for continuous-time matched discrete-time filters include the bilinear transform, frequency sampling, impulse invariance and direct frequency domain methods [5]. These methods are not without drawbacks. The bilinear transform suffers poor magnitude response at high frequencies. Frequency sampling requires very long filter lengths to capture low frequency curves. The impulse invariance method exhibits aliasing in the frequency domain. Direct frequency domain methods do not guarantee filter stability and can be computationally complex compared to simpler methods [6]. These methods can perform well for static filter designs. However, in applications where parameters are controllable in realtime, guaranteed stability is a requirement for all possible parameter settings.

In Section 2, we discuss strengths and weaknesses of these methods, particularly frequency sampling and the bilinear transform. In Section 3, we introduce our method for analogue-matched digital filter design. Section 4 gives two examples of the method in use, a

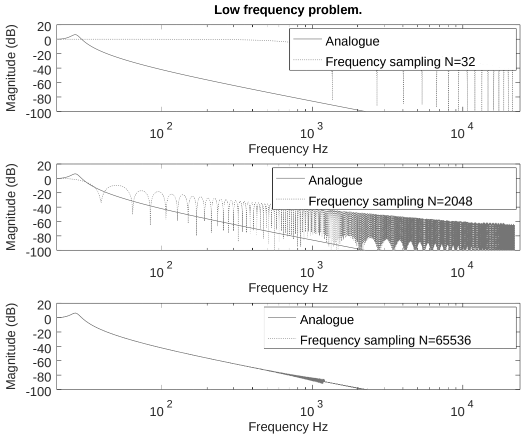

Fig. 1: Long filter lengths are required to capture low frequency curves when frequency sampling. The example analogue prototype has cutoff $f_0 = 20Hz$, $Q = 2$, $fs = 44100Hz$. To match the prototype here requires an extreme filter length of 65536 samples (latency is 743ms).

variable-Q lowpass filter and an elliptic filter. Section 5 discusses why the method works in more detail and Section 6 summarises the work, suggesting situations where it might be preferred over existing methods.

2 Current methods and motivation.

The frequency sampling method [5], [6] begins with the uniform sampling of a desired frequency response. An inverse discrete Fourier transform (IDFT) of these samples calculates the filter coefficients. The technique is simple to perform and can match both magnitude and phase response to a high degree of accuracy by choosing longer filter lengths. However, due to the uniform spacing of the DFT, extremely long filter lengths are required if the continuous-time curve extends into low frequencies. Figure 1 shows a low frequency case that requires a filter length of 65536 to adequately approximate the continuous-time prototype.

Extremely long filters add computational complexity and operational latency. Either of these factors may be unacceptable in practice. For example, if the phase is non-minimum, the latency of the third filter in Figure 1 would be $N/2 = 32768$ samples, which is unacceptable for many applications.

The impulse invariance method [5] preserves the impulse response of the continuous-time filter. However, since discrete-time sampling is inherently periodic, aliasing in the frequency domain occurs. If the passband of the continuous-time prototype extends into high frequencies the aliasing will be non-negligible. This distorts the frequency response of the discrete-time filter and limits the applicability of impulse invariance as a general solution. Likewise, direct frequency domain methods are also limited as general solutions, since they do not necessarily guarantee stable filters [6], [7].

The bilinear transform [5] is an infinite impulse response (IIR) solution that guarantees stability by mapping a stable continuous-time filter to a stable discrete-time filter. The solution is elegant and computationally simple. However, the continuous-time magnitude response approximation at high frequencies is poor. It also suffers from frequency response shape distortion due to its nonlinear mapping. This can be part compensated by "pre-warping" the continuous-time filter, but distortions away from specified critical points remain (See [8]'s discussion of Q/bandwidth).

Closed-form IIR solutions [9], [10], [11], [12] follow on from Orfanidis' improvement on the bilinear transform [13]. We will focus on the solutions put forward by [9], [10]. These solutions approximate continuous-time magnitude response fully up to Nyquist, whereas the bilinear transform drops sharply to zero as shown in Figure 2 (upper panel). While these methods have good magnitude behaviour, Figure 2 (lower panel) shows they do not match prototype phase response. The bilinear transform is a closer match in phase but the magnitude deviation at high frequencies is often unacceptable.

A full discussion of current methods in the context of audio equalisation can be found in [14]. In our brief outline, we have paid particular attention to the strengths and weaknesses of the frequency sampling and closed form IIR methods, which provide the motivation for our solution.

3 Method outline.

The method is similar to [10] in that it begins with a matched-z transformation of the continuous-time prototype. After which, the methods diverge. [10] performs a one/two-zero curve fit to yield the final filter, whereas we perform a more general, frequency sampling curve approximation.

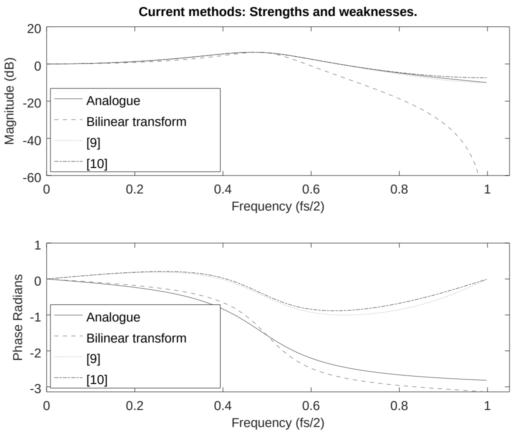

Fig. 2: Current methods and the phase problem. Upper panel shows magnitude responses of analogue prototype vs. current methods for cutoff $0.5f_s/2, Q = 2$. The bilinear transform (BLT) exhibits a drop to zero in high frequency and a visibly distorted peak shape. Lower panel shows phase responses. The BLT matches phase reasonably well but the [9], [10] designs do not match the analogue prototype's phase.

Let s, z be the continuous/discrete-time complex variables respectively, and let Ω, ω be continuous/discrete-time radian frequencies related by $\Omega = \omega/T_d$, where T_d is the design sampling period in seconds. We are given a general continuous-time prototype filter comprised of P poles located at ψ_l and Q zeros located at ζ_m in the s-domain

$$H_a(s) = \frac{\sum_{m=0}^{Q} \beta_m s^m}{1 + \sum_{l=1}^{P} \alpha_l s^l} = \beta_0 \frac{(s-\zeta_1)(s-\zeta_2)...(s-\zeta_Q)}{(s-\psi_1)(s-\psi_2)...(s-\psi_P)}$$

with frequency response $H_a(j\Omega)$. We seek to create a discrete-time filter $H_d(z)$ with frequency response $H_d(e^{j\omega})$, that approximates the prototype's frequency response $H_a(j\Omega)$ over the approximation interval $\Omega \in [-\pi/T_d, \pi/T_d]$, and meets the following design criteria:

1. Approximate magnitude response more closely than mentioned methods (over $[-\pi/T_d, \pi/T_d]$).
2. Approximate phase response more closely than mentioned methods (over $[-\pi/T_d, \pi/T_d]$).
3. Increased approximation accuracy with increasing filter length N (similar to frequency sampling).
4. Expected behaviour at low frequencies with low orders (contrary to frequency sampling).
5. Lower filter order/operating latency than frequency sampling method (for equivalent approximation accuracy).
6. Guaranteed stable, given a stable continuous-time prototype filter.

For this method, the zeros and poles of the prototype must lie in the sampling interval, that is $|\mathrm{Im}\{\zeta_m\}|, |\mathrm{Im}\{\psi_l\}| < \pi/T_d$. Regarding criterion 6 we assume that the prototype filter is stable, that is $\mathrm{Re}\{\psi_l\} < 0$.

First transform the continuous-time prototype filter $H_a(s)$ to the discrete-time filter $H_{mz}(z)$ by matched-z [5]. Each s-domain pole $s = \psi_l$ gets mapped to a z-domain pole $z = e^{\psi_l T_d}$ and each s-domain zero $s = \zeta_m$ gets mapped to a z-domain zero $z = e^{\zeta_m T_d}$.

$$H_{mz}(z) = \frac{\sum_{m=0}^{Q} b_m z^{-m}}{1 + \sum_{l=1}^{P} a_l z^{-l}}$$

$$= \beta_0 \frac{(1-e^{\zeta_1 T_d} z^{-1})(1-e^{\zeta_2 T_d} z^{-1})...(1-e^{\zeta_Q T_d} z^{-1})}{(1-e^{\psi_1 T_d} z^{-1})(1-e^{\psi_2 T_d} z^{-1})...(1-e^{\psi_P T_d} z^{-1})}$$

Due to frequency domain aliasing, it is unlikely that the frequency response of $H_{mz}(z)$ will be a good approximation of the desired prototype frequency response. To examine how different $H_{mz}(z)$ is from the prototype, find the frequency responses of $H_{mz}(z)$ and of the prototype $H_a(s)$. Evaluating the continuous-time prototype filter along the $j\Omega$ axis we get the frequency response

$$H_a(j\Omega) = \beta_0 \frac{(j\Omega-\zeta_1)(j\Omega-\zeta_2)...(j\Omega-\zeta_Q)}{(j\Omega-\psi_1)(j\Omega-\psi_2)...(j\Omega-\psi_P)}$$

Evaluating the matched-z transformed, discrete-time filter along the unit circle $e^{j\omega}$, with $\omega = \Omega T_d$, gives the frequency response

$$H_{mz}(e^{j\omega}) = \beta_0 \frac{(1-e^{\zeta_1 T_d} e^{-j\omega})...(1-e^{\zeta_Q T_d} e^{-j\omega})}{(1-e^{\psi_1 T_d} e^{-j\omega})...(1-e^{\psi_P T_d} e^{-j\omega})}$$

We are interested in how much the desired response $H_a(j\Omega)$ differs from the matched-z transformed response $H_{mz}(e^{j\omega})$, which can be expressed as the ratio

$$H_{\text{diff}}(\Omega) = \frac{H_a(j\Omega)}{H_{mz}(e^{j\omega})}$$

From here, the problem is reduced to finding a filter $H_{\text{diff}}(z)$ that approximates the ratio of responses $H_{\text{diff}}(\Omega)$. On finding $H_{\text{diff}}(z)$, we create our final discrete-time filter by series cascade: $H_d(z) = H_{\text{diff}}(z) \cdot H_{mz}(z)$, which approximates prototype $H_a(s)$ in frequency response. We are free to choose any filter design method to create $H_{\text{diff}}(z)$. However, in the interest of fulfilling design criterion 3 from above, we proceed with the frequency sampling method. We can then use an increasingly dense set of frequency sampling points to achieve higher approximation accuracy.

Our choice of uniform frequency sampling points is based on Rabiner/Gold's "Type C" scheme [5]. Since the scheme uses an odd number of points N starting at $\omega_k = 0$, it does not sample at the Nyquist frequency $\omega_k = \pi$. This decreases the frequency response oscillation between samples, a common problem when using frequency sampling. For our purposes we sample $H_{\text{diff}}(\Omega)$ at the following uniformly spaced frequencies

$$\omega_k = \frac{2\pi k}{N}, \quad k = 0, 1, ..., \frac{N-1}{2}, \quad N \text{ odd.}$$

Note that we only sample the positive frequency points, $\omega_k \in [0, \pi)$. We conjugate the positive frequency samples to get the negative frequency samples as follows

$$H_{\text{diff}}[k] = \begin{cases} H_{\text{diff}}(\Omega)^*|_{\Omega = \frac{-\omega_k}{T_d}}, & k = -\frac{N-1}{2}, ..., -2, -1 \\ H_{\text{diff}}(\Omega)|_{\Omega = \frac{\omega_k}{T_d}}, & k = 0, 1, ..., \frac{N-1}{2} \end{cases}$$

The samples can be converted to a filter by the frequency sampling interpolation formula [5] or, more commonly by the z-transform of the inverse DFT $H_{\text{diff}}(z) = Z\{\text{IDFT}\{H_{\text{diff}}[k]\}\}$. Finally, we cascade the two discrete-time filters in series to reach our goal

$$H_d(z) = H_{\text{diff}}(z) \cdot H_{mz}(z)$$

To summarise, we have designed a discrete-time filter $H_d(z)$ with frequency response $H_d(e^{j\omega})$, that approximates the prototype's frequency response $H_a(j\Omega)$ over the approximation interval $\Omega \in [-\pi/T_d, \pi/T_d]$.

4 Examples

This section demonstrates the method as performed on two example prototype filters, a variable-Q lowpass and an 8th order elliptic. It is useful to examine the variable-Q lowpass as this is the prototype used by [9], [10]. The elliptic example shows how the method can extend to prototypes having multiple poles/zeros.

4.1 Variable-Q lowpass filter.

The first example is a variable bandwidth/Q second order Butterworth-based filter, commonly used in audio applications. We begin with the transfer function of this analogue filter

$$H_a(s) = \frac{1}{\frac{1}{\Omega_c^2}s^2 + \frac{1}{Q\Omega_c}s + 1}$$

with $\Omega_c, Q \in \Re$, where Ω_c is the cutoff frequency and Q is the quality factor Q. This variable-Q filter is a Butterworth filter when quality factor is fixed to $Q = 1/\sqrt{2}$, that is, its magnitude response is maximally flat at this Q setting.

We seek to approximate the frequency response of this prototype with a digital discrete-time filter. In factored form

$$H_a(s) = \frac{1}{(s - \psi_1)(s - \psi_1^*)}$$

where the conjugate poles ψ_1, ψ_1^* in terms of the parameters are given by the quadratic formula

$$\psi_1, \psi_1^* = -\frac{\Omega_c}{2Q} \pm \Omega_c \sqrt{\left(\frac{1}{2Q}\right)^2 - 1}$$

We perform the matched-z transformation by mapping prototype poles $s = \psi_n$ to digital poles $z = e^{\psi_n T_d}$ to give

$$H_{mz}(z) = \frac{1}{(1 - e^{\psi_1 T_d} z^{-1})(1 - e^{\psi_1^* T_d} z^{-1})}$$

For matched-z, the poles of the prototype must lie in the sampling interval, that is $|\text{Im}\{\psi_1\}| < \pi/T_d$. In our lowpass example this constraint is always satisfied when cutoff Ω_c is in the interval $[0, \pi/T_d]$. The constraint is also satisfied for some $\Omega_c > \pi/T_d$, i.e. the filter cutoff may be higher than Nyquist, particularly when Q is low.

Then we evaluate the frequency responses of the prototype and the matched-z transformed filter.

$$H_a(j\Omega) = \frac{1}{(j\Omega - \psi_1)(j\Omega - \psi_1^*)}$$

$$H_{mz}(e^{j\omega}) = \frac{1}{(1 - e^{\psi_1 T_d} e^{-j\omega})(1 - e^{\psi_1^* T_d} e^{-j\omega})}$$

We form an expression of how $H_{mz}(z)$ deviates from the analogue frequency response as a ratio of magnitudes

$$H_{\text{diff}}(\Omega) = \frac{H_a(j\Omega)}{H_{mz}(e^{j\omega})} = \frac{(1 - e^{\psi_1 T_d} e^{-j\omega})(1 - e^{\psi_1^* T_d} e^{-j\omega})}{(j\Omega - \psi_1)(j\Omega - \psi_1^*)}$$

Choose filter order N and sample $H_{\text{diff}}(\Omega)$ to get the complex sequence

$$H_{\text{diff}}[k] = \begin{cases} H_{\text{diff}}(\Omega)^*|_{\Omega = \frac{-\omega_k}{T_d}}, & k = -\frac{N-1}{2}, ..., -2, -1 \\ H_{\text{diff}}(\Omega)|_{\Omega = \frac{\omega_k}{T_d}}, & k = 0, 1, ..., \frac{N-1}{2} \end{cases}$$

Take the z-transform of the inverse DFT to create $H_d(z)$ from $H_{\text{diff}}(\Omega)$

$$H_{\text{diff}}(z) = \mathrm{Z}\{\mathrm{IDFT}\{H_{\text{diff}}[k]\}\}$$

Then cascade $H_{mz}(z)$ and $H_{\text{diff}}(z)$ in series to create the final analogue matched digital filter

$$H_d(z) = H_{mz}(z) \cdot H_{\text{diff}}(z)$$

To increase average match accuracy further, before performing the inverse DFT we may linearly interpolate the phase of the positive and negative frequency samples closest to Nyquist. The trade-off with this manipulation is a slight increase in local maximum match accuracy close to Nyquist. For many applications average is more important than maximum approximation accuracy, or the frequency area close to Nyquist is non-critical.

Now that we have designed our analogue-matched digital filter, let us compare the frequency responses against current closed-form IIR methods. Figure 3 shows the method using the same example as in Figure 1. For magnitude response, our method outperforms [9], [10] with choice of filter order $N = 63$. More important is the phase response, which is an accurate approximation of the analogue prototype, a large improvement over the minimum phase approach of [9], [10].

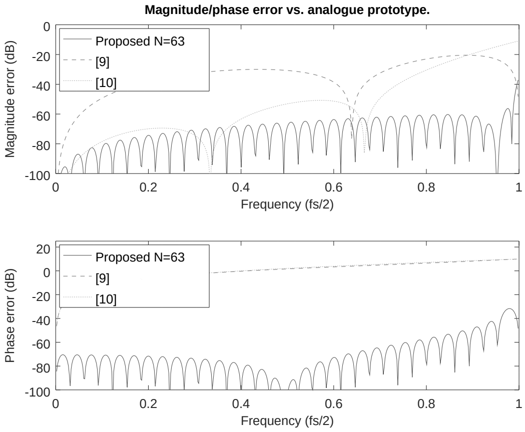

Fig. 3: Error against analogue prototype. Using filter length N=63, our method outperforms [9], [10]. Particularly, the phase response is a good approximation of the analogue prototype, where [9], [10] is not (graph traces coincide).

Figure 4 compares against the frequency sampling method, which can approximate the analogue phase response. Our solution outperforms the frequency sampling method in magnitude and has similar phase performance even with the relatively short filter order $N = 63$. In particular, the order of the frequency sampling filter is $N = 65536$, over 8000 times longer than our filter.

Figure 5 shows how increasing filter order N increases prototype match accuracy. A lower order of $N = 5$ performs similarly in magnitude response to [9], but with good phase approximation. The previous example of $N = 63$ is also shown. If extreme accuracy is required, a modest filter order of $N = 511$ gives very good results, giving a magnitude and phase response error on the order of approximately -100dB. If $N = 511$ seems like a large filter order, recall that for acceptable low frequency performance, the frequency sampling filter would be $N = 65536$.

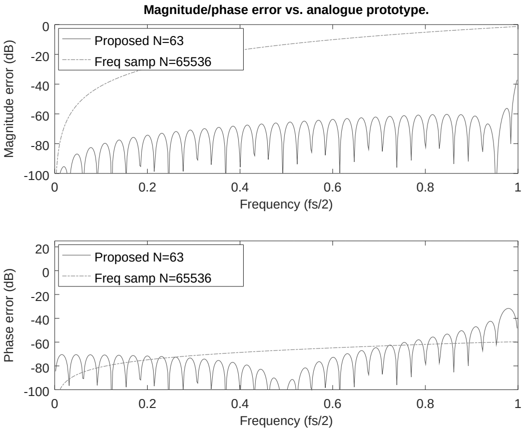

Fig. 4: Proposed method vs. frequency sampling. The proposed method greatly outperforms frequency sampling in magnitude response, and achieves similar phase response accuracy using N=63. Filter order savings are over 8000x compared with frequency sampling N=65536.

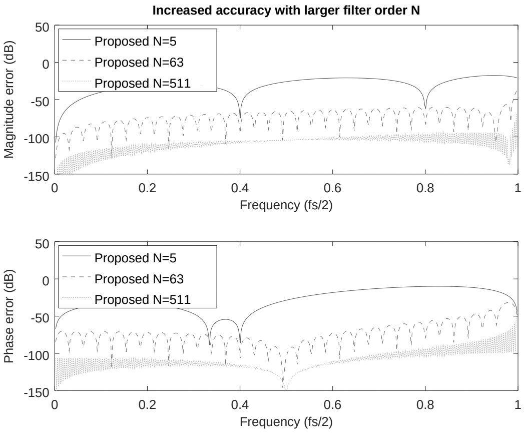

Fig. 5: Effect of varying filter order N for proposed method. Increasing filter order N increases match accuracy.

4.2 8th order elliptic filter.

Elliptic filters are typically designed by approximation of a desired magnitude tolerance scheme. However, here we transform a continuous-time elliptic prototype filter, to demonstrate that this method works for multiple poles/zeros.

We begin with an 8th order continuous-time prototype that has a unity cutoff frequency, a passband ripple of 1dB, and a stopband attenuation of 45.75dB. Designed in Matlab by the approach outlined in [6] with a continuous time transfer function of

$$H_a(s) = \beta_0 \frac{\prod_{m=1}^{8}(s-\zeta_m)}{\prod_{l=1}^{8}(s-\psi_l)}$$

consisting of gain $\beta_0 = 0.0051583$ and poles/zeros at the following s-plane locations

$$\zeta = \pm 3.139j, \pm 1.3305j, \pm 1.0926j, \pm 1.0418j$$
$$\psi = \begin{cases} -0.28490+0.35968j, -0.12557+0.81014j, \\ -0.03748+0.96087j, -0.00763+0.99977j, \\ -0.28490-0.35968j, -0.12557-0.81014j, \\ -0.03748-0.96087j, -0.00763-0.99977j \end{cases}$$

Note that the zero positions of largest magnitude are at $\Omega \approx \pi$. Transforming by matched-z to the discrete domain, with a design sampling period of $T_d = 1$ second gives

$$H_{mz}(z) = \beta_0 \frac{\prod_{m=1}^{8} 1 - e^{\zeta_m} z^{-1}}{\prod_{l=1}^{8} 1 - e^{\psi_l} z^{-1}}$$

The zeros at $\Omega \approx \pi$ in the s-domain get mapped to $\omega \approx e^{j\pi}$ in the z-domain. A prototype zero position of larger frequency magnitude would lead to aliasing, where the positive continuous frequency zeros are mapped to negative discrete frequencies, and vice-versa. For this example, we are just within the aliasing limit and the matched-z transform will be valid for our purposes. As before, create the ratio of frequency responses $H_{\text{diff}}(\Omega) = H_a(j\Omega)/H_{mz}(e^{j\omega})$, then we frequency sample $H_{\text{diff}}(\Omega)$ to get $H_{\text{diff}}[k]$.

We create $H_{\text{diff}}(z) = \mathrm{Z}\{\mathrm{IDFT}\{H_{\text{diff}}[k]\}\}$, which defines the second stage of the final filter series cascade $H_d(z) = H_{mz}(z) \cdot H_{\text{diff}}(z)$. This elliptic example was

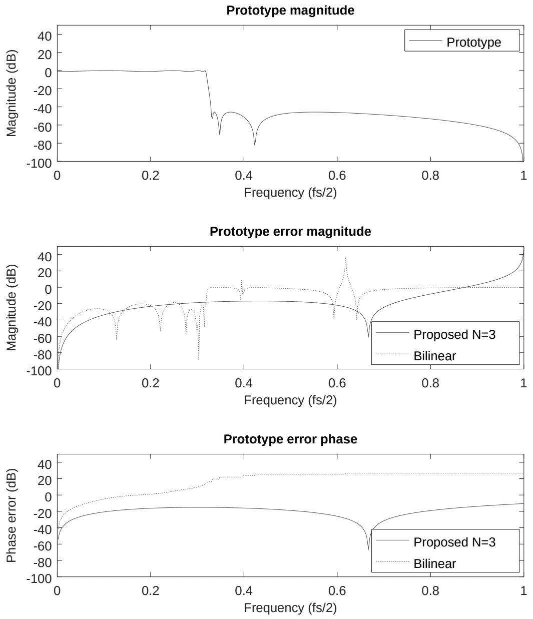

Fig. 6: Error in decibels showing how similar the digital filters are to the 8th order elliptic analogue prototype of unity radian cutoff frequency. Prototype magnitude response shown above.

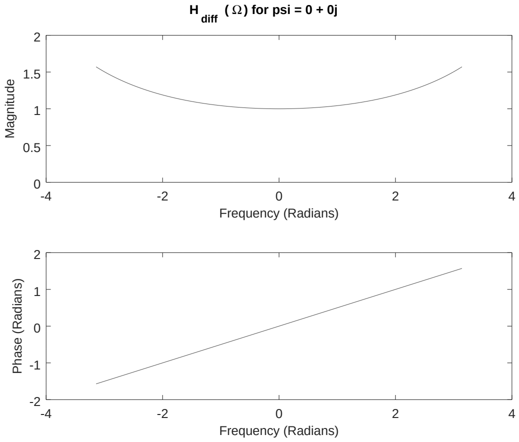

Fig. 7: Ratio of frequency responses $H_{\text{diff}}(\Omega)$ is smooth in magnitude and phase for $\zeta_1 = 0 + 0j$. The relative smoothness of this response eases approximation by frequency sampling using a low order N.

chosen to highlight an issue. When a prototype zero or pole approaches $\psi_l, \zeta_m = 0 \pm j\pi/T_d$ then $H_{\text{diff}}(\Omega)$ approaches infinity at $\Omega = \pm\pi/T_d$. We may mitigate this by choosing a low filter order N such as $N = 3$.

Figure 6 shows the decibel error against the prototype filter, with the bilinear transform method as a comparison. In this instance $N = 3$ performs well, particularly in phase response. In magnitude response there is a deviation approaching Nyquist, which should be considered by the designer. In many applications this deviation is acceptable since it occurs in the stopband.

5 Discussing $H_{\text{diff}}(\Omega)$

The method is made possible by frequency sampling the ratio of responses $H_{\text{diff}}(\Omega)$ successfully, using a low filter order. To understand this, we can examine the shape of $H_{\text{diff}}(\Omega)$ in the single pole/zero case. Higher order filters are serial cascades of the first order example. Consider a continuous-time filter that has a single finite, complex zero

$$H_a(s) = s - \zeta_1$$

Subject to the matched-z constraint: $|\text{Im}\{\zeta_1\}| < \pi/T_d$. (This example works equally well for a pole, where we would have to obey the stability constraint $\text{Re}\{\psi_l\} < 0$). Working through the process we transform to the discrete-domain by matched-z and calculate the ratio of frequency responses

$$H_{mz}(z) = 1 - e^{\zeta_1 T_d} z^{-1}$$
$$H_{\text{diff}}(\Omega) = \frac{j\Omega - \zeta_1}{1 - e^{\zeta_1 T_d} e^{-j\omega}}$$

The ratio of frequency responses $H_{\text{diff}}(\Omega)$ is smooth in the approximation interval $\Omega \in [-\pi/T_d, \pi/T_d]$. We can view an example of this by setting the design sampling period $T_d = 1$, setting zero $\zeta_1 = 0 + 0j$, and plotting the frequency response as shown in Figure 7. Plotting at different zero locations yields different, but similarly smooth responses. The relative smoothness helps explain why $H_{\text{diff}}(\Omega)$ can be approximated well by low-order frequency sampling.

The smoothness in itself does not guarantee a good approximation. However, if the frequency response of $H_{\text{diff}}(\Omega)$ is smoother than the frequency response of the prototype, it follows that we can frequency sample $H_{\text{diff}}(\Omega)$ using a lower order than frequency sampling the prototype directly. In our experiments, the order is significantly lower than direct frequency sampling of the prototype, especially in low frequency cases as we have seen in Figure 1, where the order is 8192 times lower.

6 Conclusion.

We have described a method for creating a discrete-time filter that accurately matches the magnitude and phase response of a continuous-time prototype. Table 1 informally summarises how the proposed method fits into the framework of discussed methods.

	Bilinear	[9], [10]	Proposed	Fr. samp
Magnitude	poor	very good	variable	variable
Phase	good	min only	variable	variable
Low freqs	very good	very good	very good	poor
Computation	very fast	fast	medium	slow

Table 1: Comparing proposed vs. discussed methods.

The bilinear transform is still a viable technique in applications where high frequency response approximation is not a requirement. It is computationally simple, as are the [9], [10] methods. However, [9], [10] are minimum phase solutions, so the phase response does not match that of the continuous-time prototype. If analogue phase response is not a requirement then [9] or [10] may be chosen over the proposed method, provided the magnitude response does not require extreme accuracy, in which case the proposed method should be preferred.

The frequency sampling method approximates low frequency curves poorly as demonstrated in Figure 1. Our method solves this problem by first performing a matched-z transform. The designer should be aware of the extra cost incurred by this initial matched-z transform. In most applications this cost will be acceptable, and outweighed by the order savings that occur from frequency sampling a smoother response $H_{\mathrm{diff}}(\Omega)$, as discussed in Section 5.

To summarise, the proposed method falls between closed-form IIR methods and design by frequency sampling. Similar to the frequency sampling method, it approximates the frequency response of the prototype increasingly well with higher filter order N. If analogue phase response matching is required, or if highly accurate magnitude response matching is required, the proposed method should be preferred over the mentioned closed-form IIR methods. In most applications, the proposed method can replace the frequency sampling method.

References.

[1] G. Massenburg, “Parametric equalization,” *Audio Engineering Society Convention 42, Los Angeles, USA.* 1972.

[2] R. Lagadec, D. Weiss, and R. Greutmann, “High quality analog filters for digital audio,” *Audio Engineering Society Convention 67, New York, USA.* 1980.

[3] S. Butterworth, “On the theory of filter amplifiers,” *Wireless Engineer*, vol. 7, no. 6, pp. 536–541, 1930.

[4] J. D. Reiss, “Design of audio parametric equalizer filters directly in the digital domain,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 6, pp. 1843–1848, 2011.

[5] L. R. Rabiner and B. Gold, *Theory and application of digital signal processing*. Englewood Cliffs NJ, Prentice-Hall, 1975.

[6] T. W. Parks and C. S. Burrus, *Digital filter design*. Wiley, 1987.

[7] J. Smith, *Introduction to digital filters with audio applications*. W3K Publishing, 2012.

[8] R. Bristow-Johnson, “The equivalence of various methods of computing biquad coefficients for audio parametric equalizers,” *Audio Engineering Society Convention 97, San Francisco, USA.* 1994.

[9] M. Massberg, “Digital low-pass filter design with analog-matched magnitude response,” *Audio Engineering Society Convention 131, New York, USA.* 2011.

[10] D. W. Gunness and O. S. Chauhan, “Optimizing the magnitude response of matched z-transform filters (‘mzti’) for loudspeaker equalization,” *Audio Engineering Society Convention 32, Hillerod, Denmark.* 2007.

[11] T. Schmidt and J. Bitzer, “Digital equalization filter: New solution to the frequency response near nyquist and evaluation by listening tests,” *Audio Engineering Society Convention 128, London, UK.* 2010.

[12] M. A. Al-Alaoui, “Improving the magnitude responses of digital filters for loudspeaker equalization,”

Journal of the Audio Engineering Society, vol. 58, no. 12, pp. 1064–1082, 2011.

[13] S. J. Orfanidis, “Digital parametric equalizer design with prescribed nyquist-frequency gain,” *Journal of the Audio Engineering Society*, vol. 45, no. 6, pp. 444–455, 1997.

[14] V. Välimäki and J. D. Reiss, “All about audio equalization: Solutions and frontiers,” *Applied Sciences*, vol. 6, no. 5, p. 129, 2016.