Pole Optimization of IIR Filters using Backpropagation

Kristóf Horváth, Balázs Bank
Budapest University of Technology and Economics
Department of Measurement and Information Systems
Budapest, Hungary
Email: {hkristof, bank}@mit.bme.hu

***Abstract*—Audio signal processing is a field where specialized techniques are used to account for the characteristics of hearing. In filter design the resulting transfer function need to follow the specification on an approximately logarithmic frequency scale, which can be done via methods such as frequency warping or fixed-pole parallel filters. Although these IIR filter design techniques are proven in practice, they do not produce optimal pole sets for the given specification. In this paper we present the first experiments of using a gradient-based pole optimization framework implemented in TensorFlow by realizing the IIR filter as a recurrent neural network (RNN). The method can improve the pole set of a filter compared to the initial pole set, resulting in a smaller approximation error. The proposed method is demonstrated using four example filter specifications.**

***Index Terms*—audio filter design, RNN, IIR filter**

I. Introduction

In audio filtering, infinite impulse response (IIR) filters are commonly used [1], where logarithmic frequency resolution is highly desired, to approximate the characteristics of hearing. In order to achieve this, several structures were developed including warped filters [2] and second-order fixed-pole parallel filters [3].

Warped IIR filters are derived from a direct-form IIR structure by substituting allpass sections into the delay elements [2]. The resulting structure has an additional parameter λ, called warping coefficient. In the design process, the specification is first transformed according to λ and then the filter coefficients are set using traditional methods such as Prony's method or the Steiglitz-McBride algorithm. Figure 1 shows the frequency mapping of the warping transformation. In essence the warping coefficient controls the frequency resolution of the filter by making specific parts of the specification more dominant in the warped frequency domain.

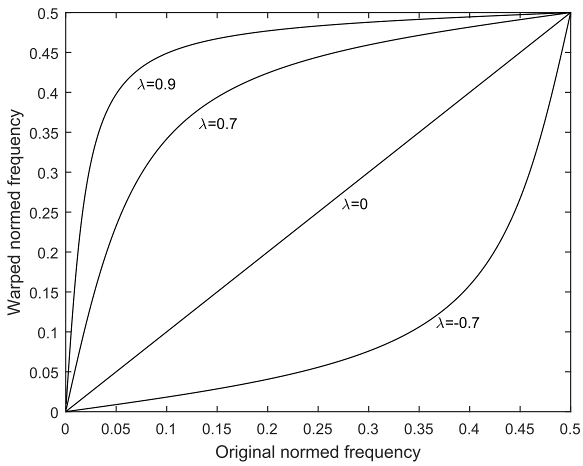

Fig. 1. Frequency transformation of warping.

In fixed-pole parallel second-order filters the frequency resolution is controlled by the setting the poles appropriately. After that, the filter response becomes linear in the numerator parameters and thus they can be estimated using the least squares (LS) method [4]. In their simplest form, fixed-pole parallel filters have predetermined pole sets which control their frequency resolution – for example poles uniformly distributed on logarithmic frequency scale will result in logarithmic frequency resolution. In addition, several more sophisticated pole positioning strategies have been developed which offer better modeling accuracy at the expense of a somewhat more complicated filter design procedure [5], [6].

In this paper we investigate whether the performance of fixed-pole parallel filters can be improved by optimizing their pole set using the backpropagation algorithm.

II. IIR filters as recurrent neural networks

A recurrent neural network (RNNs) is a class of artificial neural networks, which is often used in natural language processing. Contrary to the commonly used feedforward neural network topologies, RNNs have internal memory (state) and assume that the input has one temporal dimension, which can be arbitrarily long. The input is therefore processed along the time dimension. There are many commonly used nonlinear RNN structures such as long short-term memory (LSTM), gated recurrent unit (GRU), Elman network, etc.

In essence all linear, time invariant IIR systems can be considered as a special case of RNNs. To illustrate this, let's consider the Elman network [7], a simple RNN structure for language processing:

$$\mathbf{h}[n] = \sigma_h(W_h\mathbf{h}[n-1] + U_h\mathbf{x}[n] + \mathbf{b}_h), \quad (1)$$

$$\mathbf{y}[n] = \sigma_y(W_y\mathbf{h}[n] + U_y\mathbf{x}[n] + \mathbf{b}_y), \quad (2)$$

DOI 10.3311/MINISY2022-011

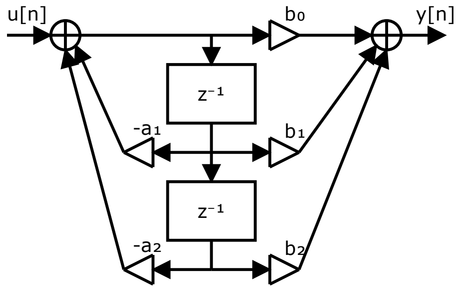

Fig. 2. IIR direct-form 2 second-order section.

where the vectors **x, y, h** are the layer inputs, outputs and hidden states, respectively. The matrices W_h, W_y, U_h, U_y and vectors $\mathbf{b}_h, \mathbf{b}_y$ are the trainable weights of the network, n is the temporal dimension, while σ_h, σ_y refer to activation functions, which are usually nonlinear functions. By setting the bias vectors to $\mathbf{b}_y = \mathbf{b}_\mathbf{h} = 0$, and removing the nonlinearities, the resulting equations are equivalent to the state-space representation of IIR filters:

$$\mathbf{h}[n] = W_h\mathbf{h}[n-1] + U_h\mathbf{x}[n], \quad (3)$$

$$\mathbf{y}[n] = W_y\mathbf{h}[n] + U_y\mathbf{x}[n]. \quad (4)$$

By representing audio filters as RNNs, the tools and frameworks for training neural networks become accessible for filter design [8]. When using mean square error (MSE) as cost function, the training process will converge to the least squares solution.

The state space representation of IIR filters preserve the filter structure, therefore it's important to specify the format of the matrices and vectors in Equations 3-4. By restricting which matrix elements can be trained the linear dependencies between variables can be eliminated.

The state space representation of a second-order IIR direct-form 2 section, as seen in Figure 2, is the following:

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}[n+1] = \begin{pmatrix} -a_1 & -a_2 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}[n] + \begin{pmatrix} 1 \\ 0 \end{pmatrix}u[n], \quad (5)$$

$$y[n] = \begin{pmatrix} b_1 - b_0a_1 & b_2 - b_0a_2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}[n] + b_0u[n]. \quad (6)$$

Note that contrary to the notation used in neural networks, in filter structures the (hidden) state is denoted by $\mathbf{x}$, while the input is denoted by u. Thus the state-space IIR direct form representation in Equations (5)-(6) corresponds to the regular transfer function:

$$H(z) = \frac{b_0 + b_1z^{-1} + b_2z^{-2}}{1 + a_1z^{-1} + a_2z^{-2}}. \quad (7)$$

RNNs are trained using backpropagation through time (BPTT), which is a gradient-based technique. A training step consists of two parts: forward propagation and backpropagation. In the former the network outputs and the cost function (also called loss function) are calculated by unrolling the mathematical operations used by the network for each time point. In the second step the derivatives of the loss function are calculated for all network weights. The network weights are updated using the analytically computed gradients such that the loss would decrease in each iteration.

III. Proposed method for pole optimization

Training the filter using backpropagation is equivalent to a gradient descent method with analytically computed gradients. Contrary to Prony and Steiglitz-McBride methods [9] though, the optimization problem in this case is nonconvex, assuming both the poles and zeros are tuned. This can cause problems such as the tendency to stuck in a local minimum as well as being prone to instability.

Iterative optimization methods, such as backpropagation, are sensitive to the initial network weights. Incorrect setting of the initial values can lead to slow convergence or getting stuck in a local optimum. Therefore we suggest that the initial network weights should be designed using the Steiglitz-McBride algorithm.

Because in Prony and Steiglitz-McBride methods designing the poles of the filter provide the biggest challenge, we suggest that only the poles of the filter should be trained using backpropagation and after a few epochs the zeros should be updated via least squares (LS) in one step. This training cycle should be repeated a few times.

Considering the previous points, the algorithm to design audio filters is the following:

1) **Warp the specification.** As a first step the filter specification is transformed to the warped frequency domain, where the filter design is performed [2].
2) **Design IIR filter using Steiglitz-McBride algorithm.** The Steiglitz-McBride method provides the initial values that are close to the optimum.
3) **Convert filter to parallel structure.** In order to use the previously designed poles and zeros, the coefficients of the direct-form representation are converted to parallel second-order representation using partial fraction expansion [10].
4) **Optimize poles with backpropagation.** The previously computed coefficients are set as the network weights of a filter represented as an RNN. In this structure only the coefficients related to the poles are trained with MSE as cost function.
5) **Design zeros for optimized poles with LS.** After the poles are optimized, the zeros are adjusted using a one-step least squares method based on Moore-Penrose pseudoinverse [4].
6) **Repeat steps 4-5.** In each iteration the remaining MSE is lowered with diminishing returns.
7) **Dewarp the second-order denominators.** In order to implement the filter, the coefficients designed in the warped frequency domain need to be transformed back. Because the dewarping would insert an additional zero to the second-order sections and thus would increase the computational demand, only the denominator is dewarped. For direct-form second-order sections the trans-

formation is performed using the following equations [11]:

$$a_1' = \frac{(1+\lambda^2)a_1 - 2\lambda a_0 - 2\lambda a_2}{1 - \lambda a_1 + \lambda^2 a_2}, \quad (8)$$

$$a_2' = \frac{a_2 - \lambda a_1 + \lambda^2 a_0}{1 - \lambda a_1 + \lambda^2 a_2}, \quad (9)$$

where the dewarped coefficients are denoted by prime.

8) **Design zeros with optimized poles.** Using the optimized pole set, the zeros of the parallel filter are designed with the usual least squares method [4], similarly to step 5.

IV. Ensuring stability

Recurrent neural networks often suffer from two major issues during training: the vanishing and the exploding gradients problem. The former is the result of unrolling nonlinear activation functions in time and as such it it is not relevant for IIR filters represented as RNNs. The exploding gradients problem, however, can be encountered when the poles of the filter are moved outside the unit circle, resulting in an unbounded growth at the output, which leads to high error values. One way to circumvent this issue is to use small learning rate and carefully initialize the coefficients [8].

In order to ensure that the poles would not become unstable during training, we have added a regularizer such that if a pole is moved outside of the unit circle in a training step, the regularizer puts it back to the circle while keeping its frequency intact.

Deriving the formulas for a conditional regularizer is hard for an arbitrarily high degree IIR filter, but for a second-order IIR direct-form section it is easy. Here we show the equations used by our implementation. If

$$4a_2 > a_1^2 \quad (10)$$

then the section has a conjugate complex pole pair. In this case the pole radius is computed as

$$R_c = \sqrt{a_2}. \quad (11)$$

If $R_c > 1$ then the pole is outside the unit circle and must be corrected to avoid instability. Optionally this condition can be tightened to avoid instability after coefficient quantization in the implementation. Thus the condition for correction is $R_c > 1-\epsilon$ where ϵ is a small number that limits the maximum amplification of the pole. The correction is done using these formulas:

$$a_1 := \frac{a_1}{R_c + \epsilon}, \quad (12)$$

$$a_2 := \frac{a_2}{(R_c + \epsilon)^2}. \quad (13)$$

If the condition in Equation (10) is false then the section has real poles. In this case the two poles and their radii are the following:

$$p_{1,2} = \frac{1}{2}\left(a_1 \pm \sqrt{a_1^2 - 4a_2}\right), \quad (14)$$

$$R_{1,2} = |p_{1,2}|. \quad (15)$$

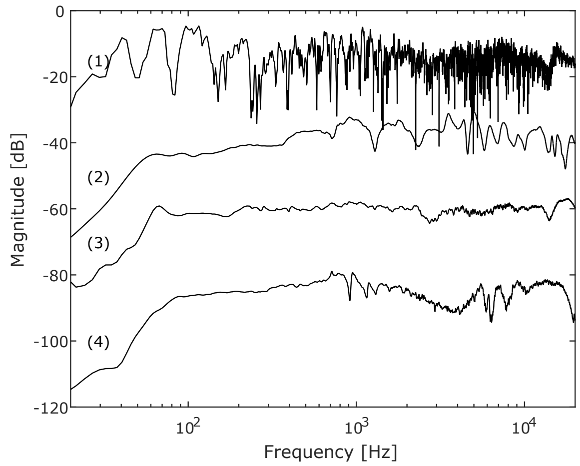

Fig. 3. Magnitude plots of example transfer functions. (1) is a room response, (2)-(4) are loudspeaker responses. The plots have been shifted in order to fit the figure.

System	Prony	Steiglitz-McBride	Proposed method	Gain
1.	$1.49 \cdot 10^{-5}$	$1.15 \cdot 10^{-5}$	$1.05 \cdot 10^{-5}$	9%
2.	$1.19 \cdot 10^{-6}$	$6.82 \cdot 10^{-7}$	$6.00 \cdot 10^{-7}$	12%
3.	$1.97 \cdot 10^{-6}$	$1.47 \cdot 10^{-6}$	$1.24 \cdot 10^{-6}$	16%
4.	$5.48 \cdot 10^{-7}$	$2.19 \cdot 10^{-7}$	$1.78 \cdot 10^{-7}$	19%

TABLE I
Mean square error (MSE) losses of different design methods on the example specifications. The values are calculated in the warped domain. Additionally, the reduction of the MSE error compared to the Seiglitz-McBride method is also shown.

If either of the pole radii are larger than 1, the pole regularizer moves them back to the unit circle. Suppose $R_1 > 1 - \epsilon$, the correction formulas are the following:

$$a_1 := a_1 - p_1 + \frac{p_1}{R_1 + \epsilon}, \quad (16)$$

$$a_2 := \frac{a_2}{R_1 + \epsilon}. \quad (17)$$

The formulas are similar for the case of $R_2 > 1 - \epsilon$.

Note that the zeros of the filter have no direct effect on the stability. The only way a zero can contribute to instability is when it covers a pole that is outside of the unit circle – in this case internal overflow can happen. Restricting the pole movements will eliminate this problem.

V. Experiments

To demonstrate the proposed method, we have designed second-order parallel filters with 4 different specifications, shown in Figure 3: one room and three loudspeaker impulse responses. In this paper we refer to these transfer functions by their numbers.

In our experiments, we have designed parallel filters with 10 second-order sections, altogether 20 poles and zeros. The warping coefficient was $\lambda = 0.9$, the learning rate during backpropagation was 10^{-4}, each pole optimization had 15 epochs and 800 steps per epoch. For the learning rate scheduler

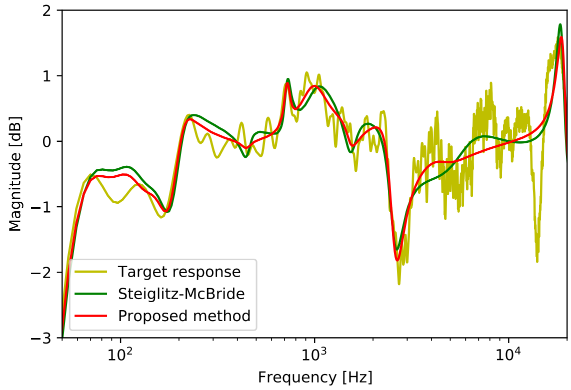

Fig. 4. Magnitude plots of system (3) and the filter responses designed by Steiglitz-McBride method and the proposed method.

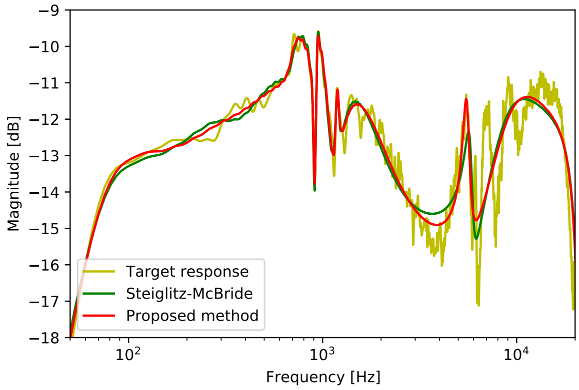

Fig. 5. Magnitude plots of system (4) and the filter responses designed by Steiglitz-McBride method and the proposed method.

we have used Adam [12] with default moments. The whole design process had 5 optimization cycles (steps 4-5 in the algorithm). The scripts were implemented using Python 3.6 and Tensorflow 2.1.0.

The achieved mean square error (MSE) values of the design process, which has been the targets of the optimization process in the warped domain, can be found in Table I. For reference we have added the results of traditional methods such as Prony and Steiglitz-McBride. Note that because the impulse responses are decaying over time and the mean is computed for $N = 1000$ samples, their mean squared value is small, leading to very small error values for all cases. However, this does not mean that this error is negligible, or would be comparable to an error coming from coefficient quantization, for example.

Accordingly, it is not the actual value of the MSE that describes the improvement due to optimization, but the relative reduction of the MSE compared to previous methods. It can be seen that the proposed method can produce coefficients that fit the example specifications with 9-19% lower remaining mean square error compared to traditional methods.

The magnitude plots of the specification and the designed filters can be found in Figures 4-5. It can be seen that the filter designed by Steiglitz-McBride method is improved by the proposed method in the full frequency range.

It should be mentioned that we have found that backpropagation is particularly sensitive to the learning rate. By setting too large learning rates the design process does not converge and therefore does not result in a usable filter. However, when setting the learning rates to too small values the poles barely get shifted and thus the method practically keeps the initial values. Finding a correct learning rate is a process of trial and error.

VI. Conclusion and Future Work

In this paper we have proposed a method for improving the poles of parallel filters using backpropagation. The results show that the method can produce filters that have lower mean square errors compared to the ones based on the original pole set designed by the Steiglitz-McBride method.

Future work includes using the proposed method for equalization, not just for modelling. Since the gradient calculation and the convergence rate is dependent on the structure of the second-order section, different implementation structures should be evaluated as well.

The use of backpropagation opens up the possibility for different cost functions, which can lead to audio filter design methods where the transformation (e.g. warping) is embedded in the loss function.

References

[1] V. Välimäki and J. D. Reiss, "All about audio equalization: Solutions and frontiers," *Applied Sciences*, vol. 6, no. 5, 2016, art. no. 129, doi: https://doi.org/10.3390/app6050129.

[2] A. Härmä, M. Karjalainen, L. Savioja, V. Välimäki, U. K. Laine, and J. Huopaniemi, "Frequency-warped signal processing for audio applications," *J. Audio Eng. Soc.*, vol. 48, no. 11, pp. 1011–1031, Nov. 2000.

[3] B. Bank, "Audio equalization with fixed-pole parallel filters: An efficient alternative to complex smoothing," *J. Audio Eng. Soc.*, vol. 61, no. 1/2, pp. 39–49, Jan. 2013.

[4] ——, "Perceptually motivated audio equalization using fixed-pole parallel second-order filters," *IEEE Signal Process. Lett.*, vol. 15, pp. 477–480, 2008.

[5] ——, "Loudspeaker and room equalization using parallel filters: Comparison of pole positioning strategies," in *Proc. 51st AES Conf. on Loudspeakers and Headphones*, Helsinki, Finland, Aug. 2013.

[6] E. Maestre, G. P. Scavone, and J. O. Smith, "Design of recursive digital filters in parallel form by linearly constrained pole optimization," *IEEE Signal Process. Lett.*, vol. 23, no. 11, pp. 1547–1550, Nov. 2016, doi: https://doi.org/10.1109/LSP.2016.2605626.

[7] J. L. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[8] B. Kuznetsov, J. D. Parker, and F. Esqueda, "Differentiable IIR filters for machine learning applications," in *Proc. Int. Conf. Digital Audio Effects (eDAFx-20)*, 2020, pp. 297–303.

[9] K. Steiglitz and L. E. McBride, "A technique for the indentification of linear systems," *IEEE Trans. Autom. Control*, vol. AC-10, pp. 461–464, Oct. 1965.

[10] A. V. Oppenheim, R. W. Schafer, and J. R. Bruck, *Discrete-Time Signal Processing*. Englewood Cliffs, New Jersey, USA: Prentice-Hall, 1975.

[11] K. Horváth, "Rounding effects in audio filters," Master's thesis, Budapest University of Technology and Economics, Budapest, Hungary, Dec. 2015, in Hungarian.

[12] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.