NEURAL PARAMETRIC EQUALIZER MATCHING USING DIFFERENTIABLE BIQUADS

Shahan Nercessian

iZotope, Inc.
Cambridge, MA, USA
shahan@izotope.com

ABSTRACT

This paper proposes a neural network for carrying out parametric equalizer (EQ) matching. The novelty of this neural network solution is that it can be optimized directly in the frequency domain by means of differentiable biquads, rather than relying solely on a loss on parameter values which does not correlate directly with the system output. We compare the performance of the proposed neural network approach with that of a baseline algorithm based on a convex relaxation of the problem. It is observed that the neural network can provide better matching than the baseline approach because it directly attempts to solve the non-convex problem. Moreover, we show that the same network trained with only a parameter loss is insufficient for the task, despite the fact that it matches underlying EQ parameters better than one trained with a combination of spectral and parameter losses.

1. INTRODUCTION

The equalizer (EQ) is an audio processor capable of selectively adjusting the loudness of specific frequencies [1]. It is a basic and important tool for the audio editor, which allows one to sculpt the tone of a sound or allow many elements to sit harmoniously in a mix. Arguably, the most popular form of EQ is the parametric EQ, largely due to its level of user control and low latency. A parametric EQ is characterized by a number of bands, whose type, frequency, gain, and quality factor (Q) can be specified by the user. In their most basic form, parametric EQs are implemented using a cascade of second-order, biquadratic filters (biquads), where each biquad corresponds to an EQ band that the user has control over [2]. Common filter types in parametric EQs include shelving, peaking, and high/low-pass filters.

EQ matching is the ability to (automatically) adjust EQ settings such that the spectral qualities of a reference track are transferred to some source material [3]. It can be used on a per-case basis, or to devise suitable criteria for automatic mixing [4] in some contexts. A common approach for EQ matching involves the computation of a desired magnitude frequency response by dividing the time-averaged spectrum of the reference by that of the source. The matching is carried out by multiplying spectral blocks of source material by the resulting magnitude frequency response using a fast Fourier transform (FFT). While providing an accurate matching, this method involves linear phase filtering, which can suffer from pre-ringing, and incurs the latency involved in FFT-based block processing. As such, a parametric EQ matching is much more desirable in many applications. Parametric EQ matching involves the automatic determination of band type, frequency, gain, and Q for each band in the EQ in such a way that it resembles the desired magnitude frequency response.

A method for parametric EQ matching was developed in [5], based on the observation that second-order peaking and shelving filters can be made nearly self-similar on a log magnitude scale with respect to peak and shelf gain changes. Band frequencies and Qs are first estimated to form a spectral basis matrix (note that for notational simplicity, we refer to both shelving slopes and peaking quality factors as Qs throughout this paper). The basis matrix is used to carry out a least-squares optimization to solve for band gains. Though it can be effective, the performance of this approach is dependent on the hand-tuned pre-estimation of frequency and Q values. Moreover, the performance can be limited due to the fact that it is a convex relaxation of the more general matching problem, as the pre-estimation of frequency and Q prior to the gain optimization does not ensure an optimal solution overall.

In [6], a similar approach to the method in [5] was used to design a graphic EQ, i.e. a constrained form of a parametric EQ with pre-determined type, frequency, and Q for each band. In fact, this is a specific instance of [5] where the initial estimation step is unneeded because spectral bases are determined a priori. Again, a least-squares optimization is used to estimate EQ band gains to match command gains, i.e. the desired magnitude frequency response evaluated at the center frequencies of the graphic EQ bands. This work was later extended in [7, 8] to make use of a neural network to infer EQ band gains. In this case, the neural network acts as a lightweight approximator of the closed form least-squares solution, where model training bootstraps the original algorithm. Though successfully retaining the performance of [6] at a lower computational cost, the disadvantage of this approach is that it is trained to minimize a loss on parameter values, rather than the magnitude frequency response itself. Therefore, one would expect limited generalizability of the approach to parametric EQs, where frequency and Q are no longer fixed. In fact, the success of the approach was likely due to the constrained nature of the graphic EQ matching problem, specifically the convex nature of the underlying gain optimization, and the higher correlation between parameters and their corresponding magnitude frequency responses relative to a more general parametric EQ matching.

In this paper, we propose a neural parametric EQ matching algorithm. The method provides a solution to the non-convex parametric EQ matching problem head-on without the need for any initial estimation of parameters via hand-tuned heuristics. Relative to other machine learning approaches, the main advantage of this model is that it is trained to optimize the spectral loss of its parameter predictions. We train a system of this sort by explicitly implementing biquads, specifically their coefficient formulae and frequency response evaluation, using differentiable operators that allow gradients to be back-propogated. This is motivated by a push

Table 1: *Biquad coefficient formulae for different filter types.*

Coefficient	Low shelf	High shelf	Peak
α	$\sin(\omega_0)\sqrt{(A^2+1)(1/q-1)+2A}$	$\sin(\omega_0)\sqrt{(A^2+1)(1/q-1)+2A}$	$\frac{\sin(\omega_0)}{2q}$
b_0	$A((A+1)-(A-1)\cos(\omega_0)+\alpha)$	$A((A+1)+(A-1)\cos(\omega_0)+\alpha)$	$1+\alpha*A$
b_1	$2A((A-1)-(A+1)\cos(\omega_0))$	$-2A((A-1)+(A+1)\cos(\omega_0))$	$-2\cos(\omega_0)$
b_2	$A((A+1)-(A-1)\cos(\omega_0)-\alpha)$	$A((A+1)+(A-1)\cos(\omega_0)-\alpha)$	$1-\alpha*A$
a_0	$(A+1)+(A-1)\cos(\omega_0)+\alpha$	$(A+1)-(A-1)\cos(\omega_0)+\alpha$	$1+\alpha/A$
a_1	$-2A((A-1)+(A+1)\cos(\omega_0))$	$2A((A-1)-(A+1)\cos(\omega_0))$	$-2\cos(\omega_0)$
a_2	$(A+1)+(A-1)\cos(\omega_0)-\alpha$	$(A+1)-(A-1)\cos(\omega_0)-\alpha$	$1-\alpha/A$

towards the implementation of differentiable audio processors in deep learning frameworks to enable end-to-end training [9]. As such, the method enjoys the ability to model non-linear functions by means of a neural network, while optimizing a loss that reflects directly the relevant output of the system. We would expect that such an approach would be more successful than a network trained to minimize a parameter loss, as such a loss would be less correlated to the direct output of the system [10]. While some prior works consider modeling audio effects using a purely black-box approach [11, 12], this paper serves as a step towards integrating the commonly used parametric EQ directly into neural networks.

The remainder of this paper is structured as follows: the construction of a parametric EQ using biquad filters and their related formulae are reviewed in Section 2. A baseline parametric EQ matching algorithm based on [5] and the proposed neural network approaches are outlined in Section 3. A comparison between the baseline algorithm and the proposed neural network solution is provided in Section 4. Finally, conclusions and allusions to future work are discussed in Section 5.

2. PARAMETRIC EQ USING BIQUADS

Biquads are a well-known and extensively studied class of infinite impulse response (IIR) filters [13]. A biquad implements the second-order difference equation

$$y[n] = \frac{1}{a_0}(b_0x[n] + b_1x[n-1] + b_2x[n-2] \\ -a_1y[n-1] - a_2y[n-2]) \tag{1}$$

where coefficients $\mathbf{b} = [b_0, b_1, b_2]$ and $\mathbf{a} = [a_0, a_1, a_2]$ are the feedforward and feedback gains of the filter, respectively. Often times, coefficients are normalized such that $a_0 = 1$. Moreover, common variations of the exact implementation of this (direct form 1) difference equation include the direct form 2 and transposed direct forms. The corresponding system function of a biquad is given as

$$H(z) = \frac{b_0 + b_1z^{-1} + b_2z^{-2}}{a_0 + a_1z^{-1} + a_2z^{-2}} \tag{2}$$

where the quadratic polynomials in the numerator and denominator affect the underlying poles and zeros of the system. Different types of filtering operations can be achieved via the placement of poles and zeros in specific ways.

The resulting frequency response of the filter can be evaluated at a digital frequency ω by evaluating equation (2) with $z = e^{j\omega}$. It is common to see this evaluation being performed over a vector of linearly-spaced digital frequencies Ω, i.e. a uniform sampling of the system function over the unit circle. We refer to this evaluation as the `freqz` operation, according to its name in MATLAB or Python/SciPy implementations. This vector Ω can be set to match the frequency axis of an underlying N-point FFT. Given real input, the FFT for even N is described completely by $N_b = N/2 + 1$ frequency bins, and accordingly, $\Omega = \pi \cdot \mathit{linspace}(0, 1, N_b)$.

There exist formulae for constructing biquad filters of a certain type which match specifications for a given center/cutoff frequency f in Hz, gain g in decibels (dB), and unitless Q/slope q [14]. In this paper, we restrict outselves to (low/high) shelving and peaking filters. For a given sampling rate f_s, we first define the terms ω_0 and A as

$$\omega_0 = 2\pi\frac{f}{f_s} \\ A = 10^{g/40} \tag{3}$$

Accordingly, filter coefficient formulae for the shelving and peaking filters used here are given in Table 1. We refer to the conversion of specified parameters f , g , q into their respective $\mathbf{b}$ and $\mathbf{a}$ coefficients as the `p2c` operation.

A K-band parametric EQ can be created by cascading K biquads in series, where each biquad acts as an EQ band with its own parameterization. In this paper, we impose the limitation that the parameter EQ consists of exactly one low shelf, one high shelf, and up to $K-2$ peaking filters. The composite system function of the parametric EQ is then given by

$$H_{eq}(z) = \prod_{k=0}^{K-1} H_k(z) \tag{4}$$

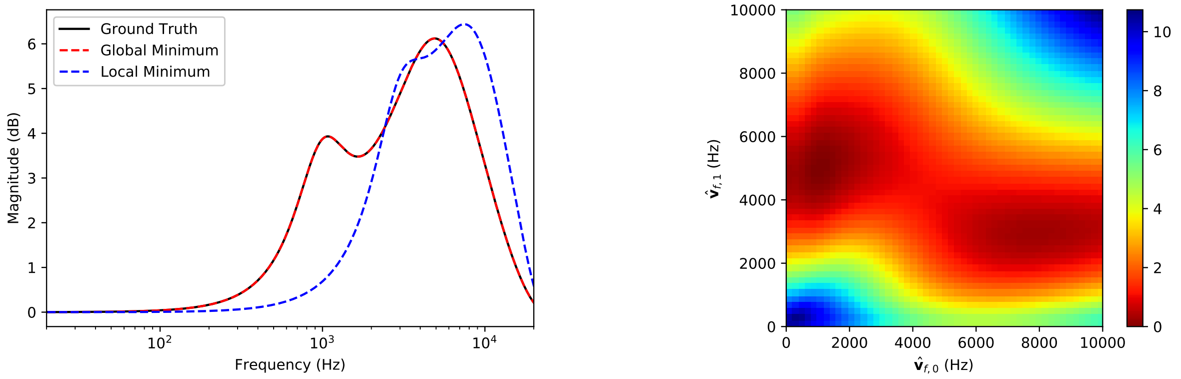
The magnitude frequency response of the parametric EQ can be evaluated over frequencies in Ω by

$$\left|H_{eq}(e^{j\Omega})\right| = \left|\prod_{k=0}^{K-1} H_k(e^{j\Omega})\right| \tag{5}$$

3. PARAMETRIC EQ MATCHING

3.1. Preliminaries

We compile the frequencies, gains, and Qs for bands comprising the parametric EQ into their respective K-dimensional vectors $\mathbf{v}_f$, $\mathbf{v}_g$, and $\mathbf{v}_q$. For example, the kth element of $\mathbf{v}_g$, denoted as $\mathbf{v}_{g,k}$, corresponds to the gain parameter of EQ band k. As such, the concatenation of the three vectors into a single $3K$-dimensional vector is a parameter vector $\mathbf{v} = [\mathbf{v}_f, \mathbf{v}_g, \mathbf{v}_q]$ which fully characterizes the settings of the EQ. Given a desired magnitude frequency response $\mathbf{x}$ (in dB), the goal of a parametric matching EQ algorithm



(a) (b)

Figure 1: *(a) Select frequency responses and (b) MSE loss as a function of frequency.*

is to estimate EQ parameters $\hat{\mathbf{v}} = [\hat{\mathbf{v}}_f, \hat{\mathbf{v}}_g, \hat{\mathbf{v}}_q]$, whose corresponding magnitude frequency response $\hat{\mathbf{x}}$ is most similar to $\mathbf{x}$.

It is instructive to verify the non-convexity of the parametric EQ matching problem. We can confirm this easily by means of a simple example using two peaking filters. We place the first band at $\mathbf{v}_{f,0} = 1\ KHz$ with $\mathbf{v}_{g,0} = 3\ dB$ and $\mathbf{v}_{q,0} = 1.0$, and the second band at $\mathbf{v}_{f,1} = 5\ KHz$ with $\mathbf{v}_{g,1} = 6\ dB$ and $\mathbf{v}_{q,1} = 0.5$. By setting the gains and Qs of these bands to different values, the bands form distinct spectral bases that are not separated by a permutation with respect to band frequency. We compute the frequency response $\mathbf{x}$ of the resulting impulse response, as illustrated in black in Figure 1a. Now, we can set $\hat{\mathbf{v}}_g = \mathbf{v}_g$, $\hat{\mathbf{v}}_q = \mathbf{v}_q$, and vary the EQ frequencies $\hat{\mathbf{v}}_f$. The difference between the desired magnitude frequency response (ground truth) and the deviated EQ magnitude frequency response is measured via the mean squared error (MSE), defined as

$$MSE(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{N_b} \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 \tag{6}$$

as illustrated in Figure 1b. We can clearly see two minima, implying that the loss surface is non-convex. It should be no surprise that the global minimum is achieved when $\hat{\mathbf{v}}_f = \mathbf{v}_f$, and corresponds exactly to the desired magnitude frequency response, as shown in red in Figure 1a. The corresponding magnitude frequency response of the non-global local minimum is shown in blue in Figure 1a.

3.2. Baseline algorithm

The baseline EQ matching algorithm is an implementation of the method described in [5]. The core observation for the algorithm is that peaking and shelving filters are approximately self-similar on the log magnitude scale with respect to gain changes. This is to say that

$$g \log \left| H_{f,\kappa,q}(e^{j\Omega}) \right| \approx \log \left| H_{f,g\kappa,q}(e^{j\Omega}) \right| \tag{7}$$

where κ is an arbitrary gain constant. The implication of equation (7) is that on a log scale, the magnitude frequency response of the parametric EQ can be approximately expressed as a linear combination of unit gain responses (i.e. $\kappa = 1\ dB$) given known $\mathbf{v}_f$ and $\mathbf{v}_q$. The combination weights are exactly the filter gains $\mathbf{v}_g$.

The self-similarity property enables an algorithm to approximate a desired magnitude frequency response using a parametric EQ. The frequencies and Qs of each EQ band are first estimated (it goes without saying that the effectiveness of the method is dependent on this estimation). With a number of EQ bands identified and frequency/Q estimates $\hat{\mathbf{v}}_f$ and $\hat{\mathbf{v}}_q$, the corresponding unit responses can be computed and stacked to form the basis matrix $\mathbf{B}$, defined as

$$\mathbf{B} = \begin{bmatrix} 20 \log \left| H_{\hat{\mathbf{v}}_{f,0},1,\hat{\mathbf{v}}_{q,0}}(e^{j\Omega}) \right| \\ \vdots \\ 20 \log \left| H_{\hat{\mathbf{v}}_{f,K-1},1,\hat{\mathbf{v}}_{q,K-1}}(e^{j\Omega}) \right| \end{bmatrix}^{\mathbf{T}} \tag{8}$$

Given $\mathbf{B}$, the gains which minimize equation (6) are determined by

$$\hat{\mathbf{v}}_g = (\mathbf{B}^{\mathbf{T}}\mathbf{B})^{-1}\mathbf{B}^{\mathbf{T}}\mathbf{x} \tag{9}$$

noting that the estimated gains $\hat{\mathbf{v}}_g$ are only optimal with respect to the estimated $\hat{\mathbf{v}}_f$ and $\hat{\mathbf{v}}_q$.

In the implementation used here, peaking filter center frequencies are determined by scanning the desired magnitude frequency response from left to right for local extrema, imposing that the next potential peaking filter be placed at least a third octave away from the previous one. Peaking filter Qs are estimated using the design equations outlined in [15]. Shelf cutoff frequencies are estimated as the closest frequencies which deviate from the frequency response evaluated at DC and Nyquist, respectively, by more than a factor of 0.5. Shelf slopes are set to a constant value of 0.75. These hand-tuned design decisions were based on analysis carried out during algorithm development, and deviate only slightly from the original method.

3.3. Neural network algorithm

The proposed approach infers $\hat{\mathbf{v}}$ and its corresponding $\hat{\mathbf{x}}$ using the neural network architecture depicted in Figure 2. In this work, the input to the model is the desired magnitude frequency response $\mathbf{x}$ in dB used "as is," though some transformation onto a logarithmic

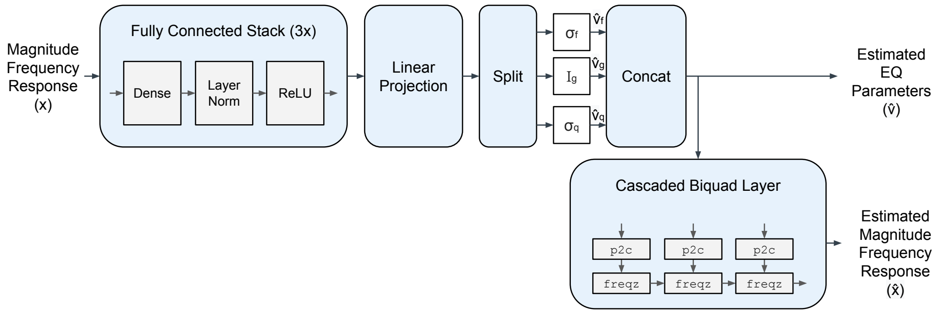

Magnitude Frequency Response (x)
Fully Connected Stack (3x)
Dense
Layer Norm
ReLU
Linear Projection
Split
σf
Ig
σq
v̂f
v̂g
v̂q
Concat
Estimated EQ Parameters (v̂)
Cascaded Biquad Layer
p2c
p2c
p2c
freqz
freqz
freqz
Estimated Magnitude Frequency Response (x̂)

Figure 2: *Parametric EQ matching neural network architecture.*

frequency scale could have also been considered. The model contains an EQ parameter inference network comprised of 3 fully connected stacks cascaded in series. Each stack consists of a 256-unit dense layer, layer normalization [16], and a ReLU activation. Note that this part of the network architecture is similar to the multilayer perceptron network in [9]. The output of the final stack is subjected to a dense layer whose output is of size $3K$. This output is split into 3 vectors of size K corresponding to the frequency, gain, and Q of the different EQ bands.

A subtlety in the design of such a network is that a portion of the output parameter space must/should be constrained. One must be mindful that adding constraints directly to network output layers by means of an activation function can eliminate gradients during model training, causing outputs to saturate at one of its extreme values. Accordingly, each parameter vector is subjected to its own type of activation. Considering the range of human hearing, and more importantly, that digital frequencies are cyclical around the Nyquist rate, the frequency vector is subjected to the activation $\sigma_f(\cdot)$, consisting of a sigmoid function and scaling to constrain its outputs to be in the range $f_{min} = 20\ Hz$ and $f_{max} = 20\ KHz$. Since there is no mathematical constraints on gains imposed by the biquad formulae, no constraint/activation is applied to the gain vector during training (indicated by the identity operator in Figure 2). During inference, however, they are clipped to $g_{min} = -10\ dB$ and $g_{max} = 10\ dB$, acknowledging that most EQs have some sort of maximum and minimum per-band gains associated with them, and that larger per-band gains in practical matching context are fairly uncommon. Values for Q must be non-negative, and for shelving filters, must also be ≤ 1. Accordingly, the Q vector is subjected to the activation $\sigma_q(\cdot)$, consisting of a sigmoid function and scaling to constrain its output to be in the range $q_{min} = 0.1$ and q_{max}, where $q_{max} = 1$ for shelving filters and $q_{max} = 3$ for peaking filters. The outputs of the respective activations are the 3 parameter vectors $\hat{\mathbf{v}}_f$, $\hat{\mathbf{v}}_g$, $\hat{\mathbf{v}}_q$, representing the frequencies, gains, and Q values for EQ bands, respectively. Finally, a concatenation of the resulting vectors forms the estimated parameter vector $\hat{\mathbf{v}}$.

One fundamental challenge in developing a useful neural parametric matching EQ algorithm is that it is instructive to relate inferred parameters to their resulting magnitude frequency response in a differentiable way. One way of achieving this is by explicitly constructing biquadratic RNN layers [17], which would enable back-propogation through time, and generating frequency responses of truncated impulse responses when passing them through these layers. Though this is certainly interesting, and may be useful for other deep learning applications which make use of parametric EQs, this is rather slow and computationally expensive in practice (this is generally the case for IIR filters). Moreover, it is unnecessary for the task at hand, assuming that there exists some reasonable means of estimating the desired magnitude frequency responses. Considering that the frequency response of cascaded biquads can be efficiently evaluated using the cascaded `freqz` expression in equation (5), we create differentiable biquads for this application by implementing the `p2c` and cascaded `freqz` operations using an automatic differentiation library (such as TensorFlow), making sure that the response is evaluated at the same frequencies as the underlying FFT used to the generate desired magnitude frequency responses. As such, the EQ frequency response evaluation operations form a layer that is actually a part of the model, enabling end-to-end training. Accordingly, the estimated desired magnitude frequency response $\hat{\mathbf{x}}$ can be determined by the estimated EQ parameters $\hat{\mathbf{v}}$, forming the outputs of the model for system training.

3.4. Loss function

The loss function of the network that is optimized during training is comprised of 3 terms, given by

$$L(\mathbf{x}, \hat{\mathbf{x}}, \mathbf{v}, \hat{\mathbf{v}}) = \alpha L_\alpha(\mathbf{x}, \hat{\mathbf{x}}) + \beta L_\beta(\mathbf{v}, \hat{\mathbf{v}}) + \gamma L_\gamma(\hat{\mathbf{v}}_g) \qquad (10)$$

where the weights α, β, and γ are network hyperparameters that balance between the different loss terms. The L_α term is the reconstruction error between the desired and predicted EQ magnitude frequency responses. Here, we simply use the MSE loss, but note that it is trivial to use any meaningful distance metric and include any form of perceptually-based frequency weighting scheme. The L_β term is the reconstruction error between the true and predicted EQ parameter values. Note that the inclusion of this term is only possible if data is generated via the sampling of EQ parameter values. The design of the curve generator used for generating training and validation data in this work is discussed further in Section 4. Here, the MSE loss is used after rescaling parameters into the range [0, 1] considering the values for f_{min}, f_{max}, g_{min}, g_{max}, q_{min}, and q_{max}, and is referred to as the MSE_v. Lastly, the L_γ term is an optional L_1 regularizer on the gain parameters of the network, which can be used to provide sparsity to the EQ match solution. Note that the training objective is fairly generalized, and can be made identical to that in [7, 8] when $\alpha = \gamma = 0$.

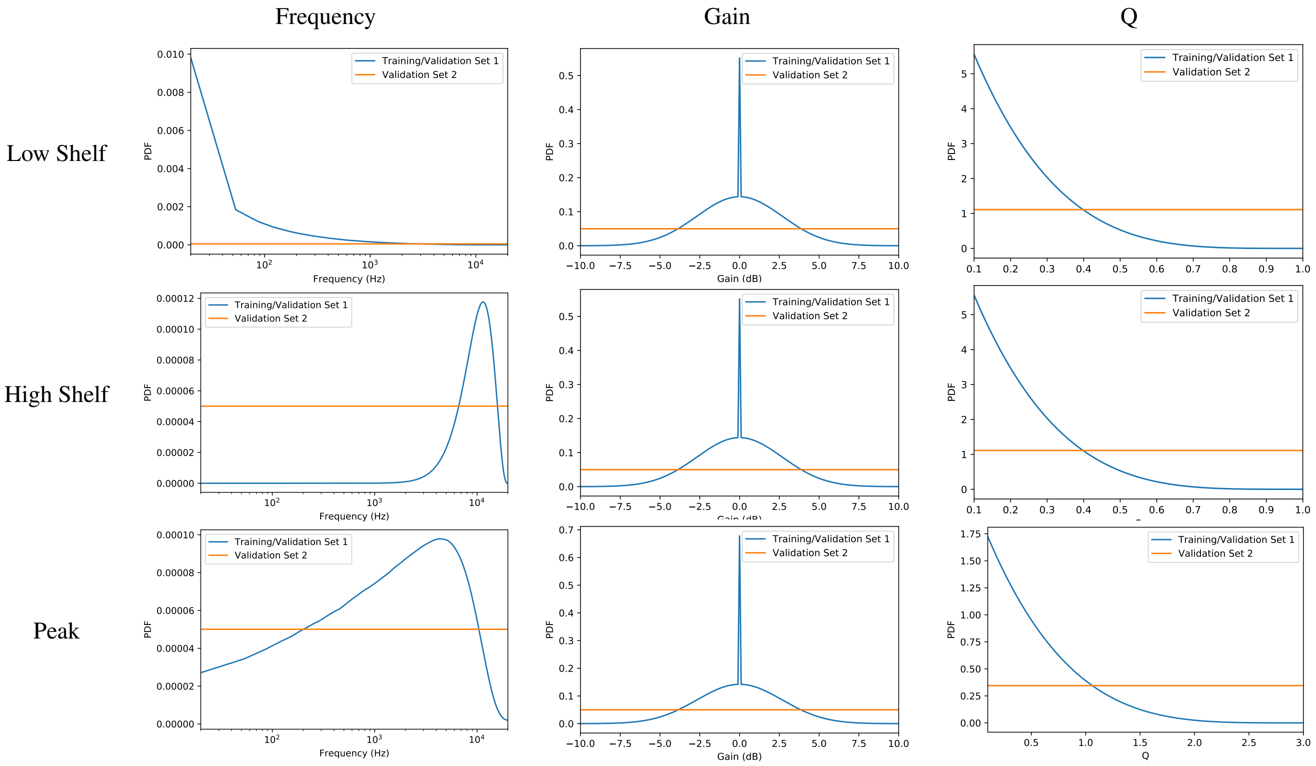


Figure 3: *PDFs used for sampling EQ parameters.*

4. EXPERIMENTAL RESULTS

4.1. Dataset description

Data is a critical component to the training and evaluation of deep learning algorithms. Here, magnitude frequency responses are generated via a random sampling of parametric EQ parameters, with $f_s = 48\ kHz$, $K = 12$, and $N = 4096$ ($N_b = 2049$), and networks are configured to match this setup. This allows for an expressive generation of magnitude frequency responses, and offers paired samples of desired magnitude frequency responses and their underlying EQ parameters. The distributions used for the random sampling of EQ parameters is dependent on band type, and the resulting probability distribution functions (PDFs) for all parameters and band types are summarized visually in Figure 3.

We chose training distributions, as shown in blue in Figure 3, to generate EQ curves that resemble the type that one would expect to see in practice. It is assumed that bands 0 and $K-1$ of the EQ are low and high shelving filters, respectively, while bands $k \in \{1, \ldots, K-2\}$ are peaking filters. Low shelf frequencies are predominantly concentrated in the low end, while high shelf frequencies span more uniformly across mid-high and high frequencies, and are respectively sampled by

$$\mathbf{v}_{f,0} = \lambda_{f,0} \cdot (f_{max} - f_{min}) + f_{min} \tag{11}$$

$$\mathbf{v}_{f,K-1} = \lambda_{f,K-1} \cdot (f_{max} - f_{min}) + f_{min} \tag{12}$$

where $\lambda_{f,0} \sim \text{Beta}(0.25, 5)$ and $\lambda_{f,K-1} \sim \text{Beta}(4, 5)$. Peaking band frequencies are sampled between low and high shelf frequencies, or $\mathbf{v}_{f,k} \sim \text{Uniform}(\mathbf{v}_{f,0}, \mathbf{v}_{f,K-1})$ for $k \in \{1, \ldots, K-2\}$. The resulting PDF is a function of random variables, visualized empirically in Figure 3. Peaking EQ bands are ordered by ascending frequency to minimize "confusion" of the parameter loss to permutations. The PDFs for gains are similar between band types. They are concentrated around 0 dB with some amount of variation, largely in the range $\pm 6dB$. Peaks in the PDF at 0 dB arise from the fact that bands are actively disabled with some probability. Accordingly,

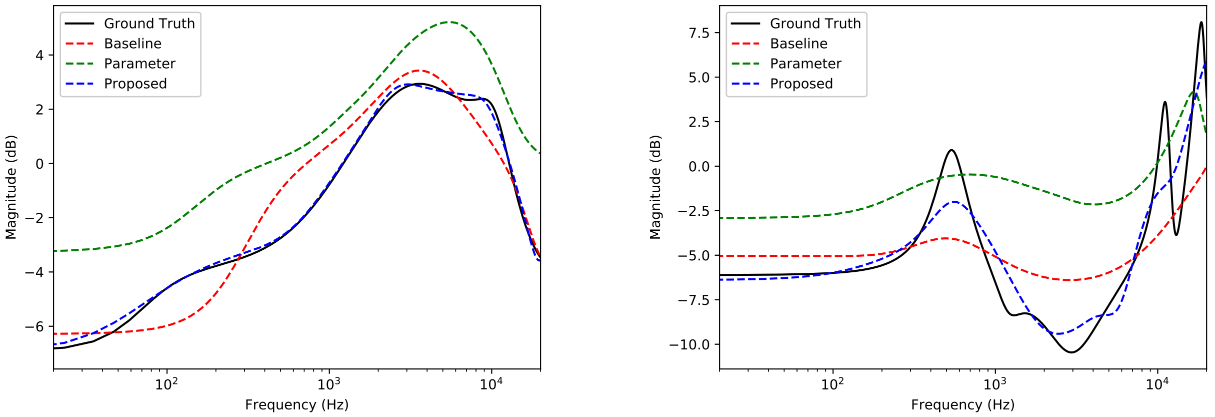
$$\mathbf{v}_{g,k} = \nu_{g,k} \cdot \lambda_{g,k} \cdot (g_{max} - g_{min}) + g_{min} \tag{13}$$

where $\nu_{g,k} \sim \text{Bernoulli}(p_k)$ with $p_k = 0.5$ for shelving filters and 0.333 for peaking filters, and $\lambda_{g,k} \sim \text{Beta}(5, 5)$. The PDFs for Qs are concentrated towards their lower range, as matching EQs are generally used more for tone shaping and less for surgical applications, and are sampled by

$$\mathbf{v}_{q,k} = \lambda_{q,k} \cdot (q_{max} - q_{min}) + q_{min} \tag{14}$$

with $\lambda_{q,k} \sim \text{Beta}(1, 5)$.

Two different validation sets for evaluating EQ matching algorithms were considered. *Validation set 1* is a set of 8192 new points generated from the training distribution that were not seen during training. *Validation set 2* is a similarly generated set of samples, except that the underlying distribution of the magnitude frequency response generator is changed dramatically by using uniform distributions for all parameters (spanning their full respective ranges), as visualized in orange in Figure 3. This introduces a large mismatch into the experiment used to analyze the generalization of different approaches. Indeed, the uniformly distributed parameters yield more complicated magnitude frequency responses that are far more challenging to all approaches that are compared here.



(a) (b)

Figure 4: *Parametric EQ matching examples from (a) validation set 1 and (b) validation set 2.*

4.2. Performance assessment

We quantatively compare the performance of different methods on both validation sets. We evaluated performance by measuring spectral similarity using both the MSE defined in equation (6) and the mean absolute error (MAE) defined as

$$MAE(\mathbf{x},\hat{\mathbf{x}}) = \frac{1}{N_b}\|\mathbf{x}-\hat{\mathbf{x}}\|_1 \tag{15}$$

The proposed network was trained with $\alpha = 1$, $\beta = 10^{-3}$, and $\gamma = 10^{-2}$, noting that a small, non-zero value for β can speed up training. Another network is trained solely using a parameter loss (i.e. $\alpha = 0$, $\beta = 1$, and $\gamma = 0$), effectively matching the training scenario in [7, 8]. Both networks were trained for 50000 steps using the Adam optimizer with a learning rate of 10^{-3} and a batch size of 16. The MSE_v is also reported for the neural networks to highlight the extent of correlations between spectral and parameter losses. We omit this computation for the baseline algorithm, as it selects a variable number of peaking filters for each sample.

Table 2 and 3 summarizes our performance evaluation. Figure 4 shows a few matching results on the two validation sets. We see that while the baseline approach can be somewhat effective, the proposed neural network approach with multiple losses visually and quantitatively provides closer matches to the ground truth magnitude frequency responses on both validation sets. We also see that while the network using only a parameter loss does capture the general shape of the desired magnitude frequency responses, its EQ matching is imprecise, and does not even offer performance comparable to the baseline approach. This is particularly interesting because its parameter loss is substantially lower than that of the proposed network, indicated by their MSE_v. This illustrates the necessity for end-to-end training that can only be achieved when the parametric EQ is itself part of the neural network.

Table 2: *Quantitative performance assessment on validation set 1.*

Method	$\mathrm{MSE}(\mathrm{x},\hat{\mathrm{x}})$	$\mathrm{MAE}(\mathrm{x},\hat{\mathrm{x}})$	$\mathrm{MSE_v}(\mathrm{v},\hat{\mathrm{v}})$
Baseline	0.1679	0.2457	–
Parameter	1.495	0.8710	**0.0275**
Proposed	**0.0782**	**0.1072**	0.0737

Table 3: *Quantitative performance assessment on validation set 2.*

Method	$\mathrm{MSE}(\mathrm{x},\hat{\mathrm{x}})$	$\mathrm{MAE}(\mathrm{x},\hat{\mathrm{x}})$	$\mathrm{MSE_v}(\mathrm{v},\hat{\mathrm{v}})$
Baseline	10.03	2.119	–
Parameter	25.29	3.881	**0.0776**
Proposed	**7.021**	**1.842**	0.1486

Lastly, we illustrate the effectiveness of the proposed approach on a real-world example. The source and reference material are recordings of the same guitar performance using a dynamic and large diaphragm condenser microphone, respectively. Critical-band smoothing [5, 18] is applied to both the time-averaged source and reference spectra $\mathbf{x}_s$ and $\mathbf{x}_r$, respectively. Specifically, we bidirectionally smooth [19] each spectrum using exponential moving average filters. Given an input spectrum $\mathbf{x}_{(\cdot)}$ to the smoothing procedure, we begin by initializing $\mathbf{x}_{(\cdot),sm} = \mathbf{x}_{(\cdot)}$. The first "forward" smoothing step (applied from left to right) is then defined by the recursive operation

$$\mathbf{x}_{(\cdot),sm}[i] = \mathbf{x}_{(\cdot),sm}[i-1] + \mu[i]\cdot(\mathbf{x}_{(\cdot),sm}[i] - \mathbf{x}_{(\cdot),sm}[i-1]) \tag{16}$$

where i is the frequency bin index. A second "backward" smoothing step is then applied recursively from right to left as

$$\mathbf{x}_{(\cdot),sm}[i] = \mathbf{x}_{(\cdot),sm}[i+1] + \mu[i]\cdot(\mathbf{x}_{(\cdot),sm}[i] - \mathbf{x}_{(\cdot),sm}[i+1]) \tag{17}$$

The filtering constant μ[i] is a function of the equivalent rectangular bandwidth of the corresponding frequency indexed at bin i. At a sampling rate f_s, an N-point FFT has a frequency resolution $\Delta f = f_s/N$, and accordingly, the smoothing constant is given by

$$\mu[i] = 1 - \exp\left[\frac{-\Delta f}{0.108(i\Delta f) + 24.7)}\right] \tag{18}$$

for $i \in \{0, \ldots, N_b - 1\}$. This filtering procedure is applied to each spectrum, yielding the smoothed, time-averaged source and reference spectra $\mathbf{x}_{s,sm}$ and $\mathbf{x}_{r,sm}$, respectively. Finally, the desired magnitude frequency response is estimated as $\mathbf{x} = \mathbf{x}_{r,sm}/\mathbf{x}_{s,sm}$.

Table 4: *Quantitative performance assessment on select examples.*

Method	Figure 4a			Figure 4b			Figure 5	
	$\mathrm{MSE}(\mathrm{x}, \hat{\mathrm{x}})$	$\mathrm{MAE}(\mathrm{x}, \hat{\mathrm{x}})$	$\mathrm{MSE}_{\mathrm{v}}(\mathrm{v}, \hat{\mathrm{v}})$	$\mathrm{MSE}(\mathrm{x}, \hat{\mathrm{x}})$	$\mathrm{MAE}(\mathrm{x}, \hat{\mathrm{x}})$	$\mathrm{MSE}_{\mathrm{v}}(\mathrm{v}, \hat{\mathrm{v}})$	$\mathrm{MSE}(\mathrm{x}, \hat{\mathrm{x}})$	$\mathrm{MAE}(\mathrm{x}, \hat{\mathrm{x}})$
Baseline	0.4394	0.4668	–	13.68	2.869	–	0.2077	0.3686
Parameter	8.044	2.692	**0.0072**	18.48	3.536	**0.1351**	1.775	0.9924
Proposed	**0.2762**	**0.2922**	0.0730	**5.703**	**1.899**	0.2048	**0.1035**	**0.2494**

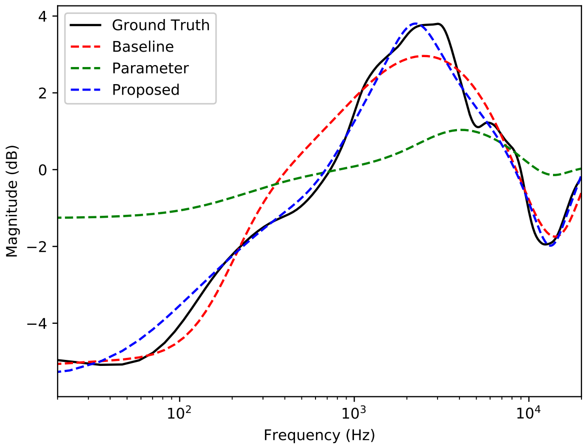


Figure 5: *Parametric EQ matching example on a real-world pair of samples.*

The results of the matching is illustrated in Figure 5. Quantitative evaluation for the examples in Figures 4 and 5 are shown in Table 4. It can be observed that the output of the proposed method is more similar to the desired magnitude frequency response, seen visually and indicated by their spectral distances. Again, the neural network trained solely on a parameter loss struggles with the matching, as its parameter estimation does not correlate directly to the resulting magnitude frequency response. Note that in this case, the underlying EQ parameter values do not exist as the desired magnitude frequency response was not synthetically created by a parametric EQ.

5. CONCLUSIONS

In this paper, a novel application of deep learning to the problem of parametric EQ matching was proposed. Central to the formulation of the neural network solution was the differentiable implementation of a parametric EQ using cascaded biquads, allowing for the optimization to occur directly in the frequency domain. The proposed solution outperformed a baseline approach based on a convex relaxation of the EQ matching problem. Moreover, it was shown that a simpler neural network optimization in the parameter space was insufficient for the problem at hand.

Future research will extend this work to classify different EQ band types during the matching process, as fixing the band types has remained a limitation imposed by different methods (including this one). Another interesting follow-on to this work would be to consider whether a neural network could learn a better notion of a desired magnitude frequency response than the division of source and reference spectra, in a similar fashion to the feature learning introduced in [20]. It could concievably learn to factor out performance-driven contributions (i.e. the potentially different pitches comprising the source and reference performances), lending itself closer to a true timbral matching.

6. ACKNOWLEDGEMENTS

The author would like to thank François Germain and all the anonymous reviewers for their invaluable comments while preparing this paper, which dramatically improved the quality of this work.

7. REFERENCES

[1] V. Välimäki and J. D. Reiss, “All about audio equalization: Solutions and frontiers,” *Applied Sciences*, vol. 6, no. 129/5, pp. 1–46, May 2016.

[2] 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, Aug. 2011.

[3] F. Germain, G. Mysore, and T. Fujioka, “Equalization matching of speech recordings in real-world environments,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Shanghai, China, 2016, pp. 609–613.

[4] B. De Man, J. D. Reiss, and R. Stables, “Ten years of automatic mixing,” in *Proceedings of the 3rd Workshop on Intelligent Music Production*, Salford, UK, Sept. 2017.

[5] J.S. Abel and D.P. Berners, “Filter design using second order peaking and shelving sections,” in *Proceedings of the International Computer Music Conferences*, Miami, FL, USA, Nov. 2004.

[6] J. Liski and V. Välimäki, “The quest for the best graphic equalizer,” in *Proceedings of the International Conference on Digital Audio Effects*, Edinburgh, UK, Sept. 2017, pp. 95–102.

[7] J. Rämö and V. Välimäki, “Neural third-octave graphic equalizer,” in *Proceedings of the International Conference on Digital Audio Effects*, Birmingham, UK, Sept. 2019, pp. 1–6.

[8] V. Välimäki and J. Rämö, “Neurally controlled graphic equalizer,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 2140–2149, Dec. 2019.

[9] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, “DDSP: Differentiable digital signal processing,” in *Proceedings of the International Conference on Learning Representations*, Addis Ababa, Ethiopia, Apr. 2020, pp. 26–30.

[10] P. Esling, N. Masuda, A. Bardet, R. Despres, and A. Chemla-Romeu-Santos, "Universal audio synthesizer control with normalizing flows," in *Proceedings of the International Conference on Digital Audio Effects*, Birmingham, UK, Sept. 2019.

[11] M. A. Martínez Ramírez and J. D. Reiss, "Modeling nonlinear audio effects with end-to-end deep neural networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 171–175.

[12] A. Wright, E.P. Damskägg, and V. Välimäki, "Real-time black-box modelling with recurrent neural networks," in *Proceedings of the International Conference on Digital Audio Effects*, Birmingham, UK, Sept. 2019.

[13] S. K. Mitra and J. F. Kaiser, Eds., *Handbook for Digital Signal Processing*, J. Wiley & Sons, New York, NY, USA, 1993.

[14] R. Bristow-Johnson, "RBJ Audio-EQ-Cookbook," Available at https://www.musicdsp.org/en/latest/Filters/197-rbj-audio-eq-cookbook.html, accessed March 08, 2020.

[15] J. S. Abel and D. P. Berners, "Discrete-time shelf filter design for analog modeling," in *Proceedings of Audio Engineering Society, Convention 115*, New York, NY, USA, Oct 2003.

[16] J. Ba, J. Kiros, and G. Hinton, "Layer normalization," *CoRR, abs/1607.06450*, July 2016.

[17] É. Bavu, A. Ramamonjy, H. Pujol, and A. Garcia, "TimeScaleNet: a multiresolution approach for raw audio recognition using learnable biquadratic iir filters and residual networks of depthwise separable one-dimensional atrous convolutions," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 220–235, May 2019.

[18] J. O. Smith III, Ed., *Techniques for Digital Filter Design and System Identification with Application to the Violin*, Stanford University, 1983, Ph.D. thesis.

[19] F. Gustafsson, "Determining the initial states in forward-backward filtering," *IEEE Transactions on Signal Processing*, vol. 44, pp. 988–992, 1996.

[20] D. Sheng and G. Fazekas, "A feature learning siamese model for intelligent control of the dynamic range compressor," in *Proceedings of the International Joint Conference on Neural Networks*, Budapest, Hungary, 2019.