J. T. Colone and J. Reiss, "Reverse Engineering
a Nonlinear Mix of a Multitrack Recording"
J. Audio Eng. Soc., vol. 71, no. 9, pp. 586–595, (2023 September).
DOI: https://doi.org/10.17743/jaes.2022.0105

Reverse Engineering a Nonlinear Mix of a Multitrack Recording

JOSEPH T COLONE, *AES Student Member,* **AND JOSHUA REISS,** *AES Fellow*
(j.t.colonel@qmul.ac.uk) (joshua.reiss@qmul.ac.uk)

Centre for Digital Music, Queen Mary University of London, London, UK

In the field of intelligent audio production, neural networks have been trained to automatically mix a multitrack to a stereo mixdown. Although these algorithms contain latent models of mix engineering, there is still a lack of approaches that explicitly model the decisions a mix engineer makes while mixing. In this work, a method to retrieve the parameters used to create a multitrack mix using only raw tracks and the stereo mixdown is presented. This method is able to model a multitrack mix using gain, panning, equalization, dynamic range compression, distortion, delay, and reverb with the aid of greybox differentiable digital signal processing modules. This method allows for a fully interpretable representation of the mixing signal chain by explicitly modeling the audio effects one may expect in a typical engineer's mixing chain. The modeling capacities of several different mixing chains are measured using both objective and subjective measures on a dataset of student mixes. Results show that the full signal chain performs best on objective measures and that there is no statistically significant difference between the participants' perception of the full mixing chain and reference mixes.

0 INTRODUCTION

The process of mixing a multitrack recording to a mixdown is time-intensive, artistic in nature, and difficult to perfect, and it requires expert knowledge. With the spread of cheap computing and digital audio workstations, many amateur mix engineers and bedroom producers find themselves presented with a multitude of professional-grade software tools for mixing, but with little guidance. There is a gap in delivering this expert knowledge to amateurs, one that may be filled by methods for modeling mixing behavior.

From an engineering and machine learning perspective, the task of transforming a multitrack to a mixdown is highly complex because it includes a combination of linear, nonlinear, time-invariant, and time-varying transforms, referred to as audio effects. Furthermore, the choice of which effects to apply and their parameter settings depends not only on the content of each track in the multitrack but also genre and style considerations.

The question remains, though, of how to model a professional mixer's behavior and style. Over the past decade, much research has been published to address portions of this problem, such as interrogations of mixing "best practices and common sense" [1]. For example, an examination of the literature suggests that rules like panning the elements of a drum kit according to an audience's perspective are inaccurate and undesired [2]. Furthermore, much work has been published on algorithms for autonomous mixing. These include black box algorithms to mix a song from a set of stems into a final mixdown [3], algorithms devoted to properly grouping and panning percussive stems of a song [4], and plugins that aid users by providing a map of semantic descriptors to effect settings [2].

The field of music information retrieval has found some applications in this task as well. In [5], the authors perform music information retrieval feature extraction on a dataset of mixdowns and analyze how these features vary across mixes. Although there is some disagreement in the literature regarding how this approach may generalize [6], the core idea remains illuminating—if a model of mix engineering behavior exists in mixdowns, how best can it be extracted?In other words, what tools can be used to construct the field of "mix information retrieval?"

One such approach, called "reverse engineering a mix," was proposed in [7]. Given a multitrack and mixdown, this technique can match the effects used to mix the multitrack, with certain limitations. For example, this method does not attempt to match any reverb that may be used in a mixdown. Moreover, the paper presents separate methods for matching linear processing and nonlinear processing with no explicit method for combining the two into a full mixing chain. This approach was expanded upon in [8] to include reverb in the linear processing using differentiable

digital signal processing (DDSP) but did not attempt to model nonlinear processing.

This work extends the approach of [8] with the inclusion of nonlinear processing in the form of memoryless distortion and dynamic range compression. The memoryless distortion effect is adapted from [9], and the dynamic range compression model is adapted from [10]. Thus, this method is able to model a multitrack mix using gain, panning, equalization, dynamic range compression, distortion, delay, and reverb with the aid of greybox DDSP modules. The simultaneous optimization of linear and nonlinear effects represents an improvement to the original method shown in [7], which separately optimized linear and nonlinear processing, and its revision in [8], which only modeled linear processing. Furthermore, this method allows for a fully interpretable representation of the mixing signal chain by explicitly modeling the audio effects one may expect in a typical engineer's mixing chain.

1 BACKGROUND

1.1 Differentiable Digital Signal Processing

The term DDSP was proposed by [11], in which common DSP modules are manually implemented in a differentiable framework such as Tensorflow or Pytorch. This auto-differentiated regime allows for these modules to be implemented in or controlled by neural networks because of their ability to backpropagate gradients. In [11], a neural network was proposed that used differentiable harmonic oscillators, noise filtered through finite impulse response (FIR) equalisers (EQs), and convolutional reverb to synthesize audio. Since then, many individual audio effects have been implemented in a differentiable manner.

In [12], the author trains a neural network to estimate the parameters of a parametric EQ given a magnitude response as input. This is made possible because the author implemented the calculation of a magnitude response from these parameters in a differentiable framework. This approach is generalized to estimate the coefficients of an arbitrary cascade of biquads in [13].

In [14], the authors present a differentiable parametric reverberation algorithm. Here, the authors implement a feedback delay network using similar differentiable operations in [12, 13].

To model memoryless distortion effects, the authors of [9] propose a differentiable Wiener-Hammerstein (W-H) model with a learnable waveshaping nonlinearity. The method proposed in [15] can model a larger family of distortion effects by cascading hyperbolic tangent nonlinearities with hyper-conditioned differentiable biquads.

Dynamic range compressors (DRCs) have also been implemented using DDSP. In [16], the authors implement a feedforward DRC with a single pole level detector filter. The authors of [10] also implement a feedforward DRC but use approximate moving average filters to approximate attack and release ballistics.

DDSP has also found usage in tasks outside of individual effect modelling. Such tasks include audio synthesis [17–19], singing voice synthesis [20, 21], and style transfer of audio effects [22].

1.2 Reverse Engineering a Mix

In [7], a method for reverse engineering a mix is presented, which combines separate modeling of the linear and nonlinear processing used to create a stereo mixdown. This method takes as input both a multitrack and mixdown and outputs parameters that describes how the multitrack was mixed to the mixdown.

The nonlinear processing is estimated using a frame-based approach, in which DRC is modeled using time-varying polynomial gain envelopes of a fixed order over a fixed window and hop length. The coefficients of these polynomials across the multitrack can be solved for using a least-squares estimate in the time domain on a frame-by-frame basis. The smoothness of these polynomials on adjacent frames is considered as well.

A time domain least-squares approach is used to model linear processing as well, including gain, panning, delay, and EQ. This method theoretically holds when estimating a convolutional reverb impulse response, but the length of these impulse responses makes a least-squares estimate impractical.

A linear mixing chain is constructed in [8] using DDSP that can model gain, panning, delay, EQ, and reverb. This method uses gradient descent to simultaneously update the parameters of each module in the effect chain. SEC. 2 will present a thorough accounting of the methods used in [8] because this work will combine the linear processing chain of [8] with the memoryless distortion of [9] and the DRC of [10].

2 METHODS

2.1 Formal Problem Statement

Let $y(n)$ represent a target mixdown, and let $\hat{y(n)}$ represent the mixdown produced by some mixing chain characterized by a set of parameters θ. The goal is to find values θ that correspond to parameter settings in a mixing chain that will minimize $||y(n) - \hat{y}(n)||$, in which $\|\cdot\|$ denotes some cost function.

2.2 System Overview

All raw tracks and mixdowns are sampled at 44.1 kHz. The signal processing chain applied to each input raw track is as follows: Dry Input → Gain → FIR EQ → DRC & Wet/Dry Mix → Distortion & Wet/Dry Mix → Pan → Reverb & Wet/Dry Mix → Sum with other stems. To drive each module, a set of parameters Θ_{module} are estimated. Refer to Fig. 1 for a block diagram of the proposed system.

This mixing chain extends the "stereo reverb bus" mixing chain presented in [8] by adding a memoryless distortion effect and a DRC. Therefore, SECS. 2.3, 2.4, and 2.7 are restatements of the module formulations in [8]; SEC. 2.5 is an elaboration on the module presented in [10]; and SEC. 2.6 is a restatement of the module shown in [9].

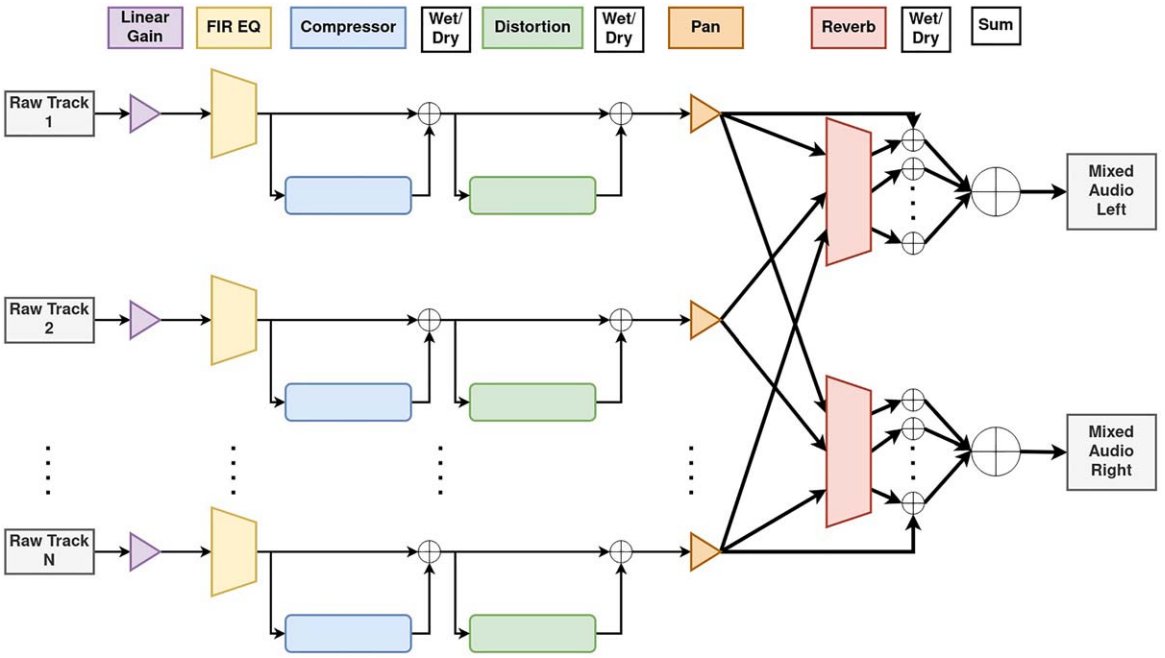

Fig. 1. Signal chain of full reverse-engineering system.

2.3 Gain and Panning

The gain module is a simple linear gain, formulated as

$$\Theta_{Gain} \times x(n). \tag{1}$$

Note that these gains can go negative, which corresponds with applying a polarity inversion to the raw track.

The panning module utilizes a linear panning law and is formulated as

$$\begin{aligned} \text{Pan}_L &= \big(0.5 + (0.5 \times \tanh(\Theta_{Pan}))\big) \times x(n) \\ \text{Pan}_R &= \big(1 - \text{Pan}_L\big) \times x(n) \end{aligned}, \tag{2}$$

where $\times$ denotes pointwise multiplication and $\tanh(\cdot)$ denotes the hyperbolic tangent function

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{3}$$

The panning module applies a gain of Pan_L to the signal before sending it to the left channel and a gain of Pan_R to the signal before sending it to the right channel.

The choice of which panning law to use is somewhat arbitrary, because both the linear panning law [23, 3] and equal power laws such as the cosine panning law [4] have found usage and success within intelligent music production. In this case, the linear law was chosen for simplicity.

The gain and panning modules are initialized using the least-squares estimate outlined in [7]. In the event of a polarity mismatch between the estimated left and right channel gains, the polarity of the larger gain is chosen for initialization.

2.4 "Graphic" Equalizer

EQing is performed by multiplying an input signal's short-time Fourier Transform (STFT) magnitude response with a user-specified curve in the frequency domain [11]. In this work, a 1,025-point frequency transfer curve Θ_{EQ} is used. This corresponds to a FIR EQ with 2,048 taps in its impulse response. Given a raw track $x(n)$, the EQ module can be written as

$$\text{EQ}(x(n)|\Theta_{EQ}) = \text{ISTFT}\big(\text{STFT}(x(n)) \times \Theta_{EQ}\big), \tag{4}$$

where ISTFT refers to the inverse short-time Fourier transform and $\times$ refers to pointwise multiplication.

In this work, the EQ is modeled after a ten-band FIR graphical EQ [24], which can be characterized using a ten-dimensional $\Theta_{EQ\,gains}$. The ten values specify the gain of each octave band filters, which are centered at 30; 60; 125; 250; 500; 1,000; 2,000; 4,000; 8,000; and 16,000 Hz, respectively. Shelving filters are used for frequencies below 30 Hz and above 16,000 Hz that match the attenuation specified at the lowest and highest octave band, respectively.

The following procedure is used to calculate the 1,025 dimensional Θ_{EQ} that will approximate a ten-band FIR graphical EQ. First, a ten-dimensional $\Theta_{EQ\,gains}$ is generated. Then, these values are transformed via

$$\Theta_{EQ\ gains} \leftarrow 1 - \sigma(\Theta_{EQ\ gains}), \tag{5}$$

where σ denotes the sigmoid function

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \tag{6}$$

The values in the transformed $\Theta_{EQ\,gains}$ range from (0,1) because of the bounds of the sigmoid function.

Finally, a piecewise linear frequency transfer curve Θ_{EQ} is constructed using linear interpolation between the octave band attenuations specified by $\Theta_{EQ\,gains}$. Thus, the EQ module's frequency transfer curve is bounded from (0,1) at all points. The estimated values are initialized with random uniform noise from [–1,1], which initializes the octave band gains from –6 to –1 dB.

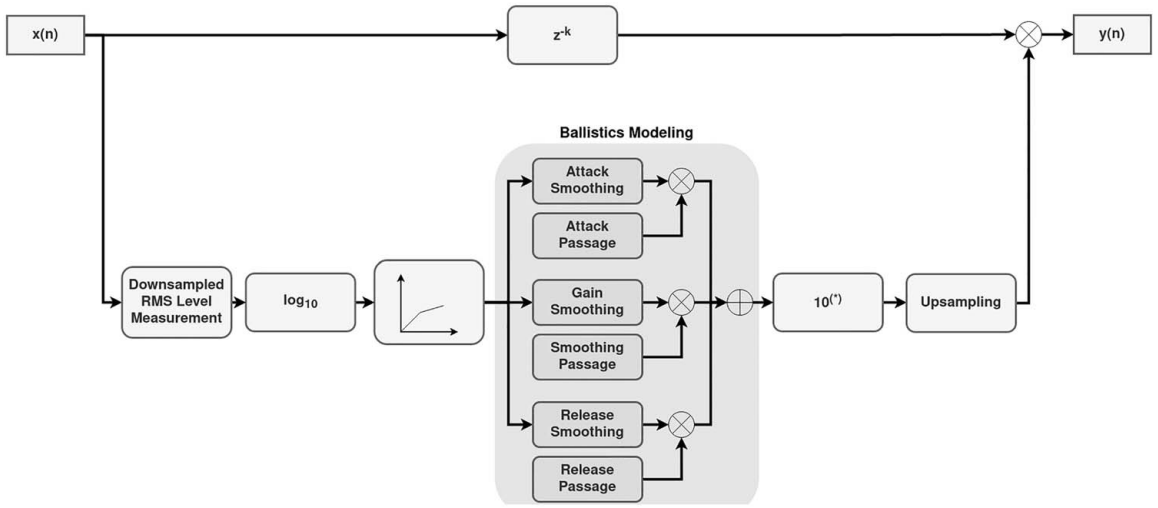

Fig. 2. Block diagram of DDSP DRC.

2.5 Dynamic Range Compressor

Refer to Fig. 2 for a block diagram of the proposed system. Given a fixed length audio signal sampled at 44.1 kHz and values for threshold, compression ratio, knee-width, makeup gain, attack time, and release time, the following procedure is used to apply dynamic range compression.

First, an RMS level measurement is calculated using a 5-ms window and hop size of 0.22 ms and converted to decibels. This generates a loudness curve measured at 4,410 frames per second.

Then, attack and release passages are estimated by finding when this loudness curve crosses the threshold value. Attack passages are calculated by convolving a rectangular window of length τ_{at} with the rising edge of the input signal passing the threshold, and they release passages by convolving a rectangular window of length τ_{rt} with the falling edge. The length of these rectangular windows corresponds to the attack and release times calculated in frames. These passages finally interfere with one another when they overlap so that the DRC is not simultaneously set to attack and release. Finally, gain smoothing passages are calculated by finding the portions of the loudness curve both above the threshold and outside of attack passages. Thus, three masks are produced that are the length of the signal's loudness curve corresponding to attack passages, release passages, and smoothing passages.

Afterward, a compression characteristic is calculated using the threshold, ratio, and knee width for the duration of the signal. This compression characteristic curve is then subtracted from the original signal's loudness curve in order to produce an attenuation curve. Note that this curve measures the decibel attenuation per frame that produces the characteristic curve when applied to the original signal.

Given a time constant τ in frames, an approximate moving average filter with support $[0, N]$ and scaling factor γ takes the form

$$h(x) = \frac{1}{\sum_0^N (\tanh(\gamma * \mathrm{relu}(\tau - x))} \tanh(\gamma * \mathrm{relu}(\tau - x)), \tag{7}$$

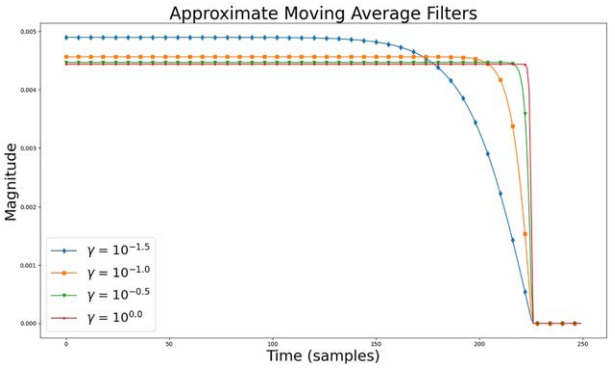

Fig. 3. Plots of approximate moving average filters with support [0,225] and varying γ.

where $\tanh(x)$ refers to the hyperbolic tangent function and $\mathrm{relu}(x)$ refers to the rectified linear unit. In this work, a scaling factor of 0.1 was chosen. Whereas a scaling factor larger than 0.1 would make $h(x)$ more closely approximate a moving average filter, it was experimentally found that the 0.1 scaling factor provides a decent approximation while allowing for gradients to backpropagate through the system. See Fig. 3 for plots of several approximate moving average filter with support [0,225] and varying γ.

Three approximate moving average filters are calculated using τ_{at}, τ_{rt}, and τ_{st}, corresponding to the attack action, release action, and gain smoothing action of the DRC. These three filters are convolved in parallel with the attenuation curve, windowed according to the attack/release/smoothing passages mentioned above, and then summed. Afterward, the makeup gain is applied. Finally, the smoothed attenuation curve is converted from decibels to a linear scale, up-sampled from 4,410 frames per second to the original sampling rate using linear interpolation, delayed by 5 ms to simulate a lag in level measurement, and applied to the original audio sample via multiplication.

The DRC parameters must be initialized such that each contributes to the compression applied to the dry signal.

Otherwise, these parameters will not update during optimization. Furthermore, "reasonable" parameters should be chosen to avoid portions of the loss surface very far from expected DRC parameters. As such, the threshold value is initialized close to the mean value of the dry signal's downsampled RMS level curve, the ratio initialized close to 2.0, the knee-width initialized close to 2 dB, makeup gain initialized just above 0 dB, τ_{at} and τ_{st} initialized close to 45 frames (about 10 ms), and τ_{rt} initialized close to 450 frames (about 100 ms).

2.6 Waveshaping Distortion

The distortion module is formulated as a differentiable W-H model. The typical W-H model consists of a linear block, nonlinear block, and linear block cascaded in series. The W-H models in this work are time-invariant, formulated using 20-band "graphic" equalizer pre-emphasis and de-emphasis filters for the linear blocks and a parameterized "PowTanh" waveshaping function as the nonlinearity.

$$f(x) = a_1 \tanh(x) + a_2 \tanh(x^2) + \ldots + a_{n-1} \tanh(x^{n-1}) + a_n \tanh(x^n). \quad (8)$$

As a weighted sum of tanh functions, the PowTanh family exhibits saturation toward $\pm\infty$. However, as a sum of even and odd functions the PowTanh can model both symmetric and asymmetric distortions. Note that $a_c \tanh(0^c) = 0$, meaning the waveshaper introduces no DC offset. As such, no DC offset is included in this parameterization. For stability, a_1 is initialized close to 1, and a_c is initialized close to 0 otherwise. During optimization, $f(x)$ is normalized such that $\max(|f(x)|) = 1$.

2.7 Reverb

Similar to the EQ module, the reverb module also performs convolution with a given impulse response via multiplication in the frequency domain. Instead of estimating a frequency transfer curve, however, the reverb module directly estimates an impulse response Θ_{IR}.

With this stereo reverb bus architecture, a wet/dry mix is produced by performing a weighted sum between the input stem and the stem with reverb applied. Thus, the module estimates Θ_{IR} for the reverb's impulse response and $\Theta_{W/D}$ for the module's wet/dry mix. The module's output is formulated as

$$\text{Reverb} = x(n) + \sigma(\Theta_{W/D}) \times (x(n) * \Theta_{IR}), \quad (9)$$

where $\sigma()$ denotes the sigmoid function, $\times$ denotes multiplication, and * denotes convolution. $\Theta_{W/D}$ is initialized with uniform random noise from $[-0.3, 0.3]$, which corresponds to a range of -7 to -5 dB, and Θ_{IR} is initialized with mean 0, variance 10^{-6} random Gaussian noise.

3 EXPERIMENT

3.1 Cost Function and Optimization Procedure

Given a randomly initialized signal chain, target mixdown $y(n)$, raw tracks $x_i(n)$ and estimated mixdown $\hat{y}(n)$, gradient descent can be used to minimize $||y(n) - \hat{y}(n)||$ by updating the module parameters Θ, in which $\|\cdot\|$ denotes some norm used as a cost function.

In this work, a multi-scale spectrogram (MSS) loss is used as the cost function $\|\cdot\|$ [11], which was inspired by the multi-resolution spectral amplitude distance demonstrated in [25]. As the name implies, MSS compute manuscripts a norm by measuring the distance between the spectrograms of two audio signals with varying STFT window sizes and performing a weighted sum of these differences. Although mean absolute error (MAE) in the time domain is often used in audio applications and is cheaper to compute than MSS loss, the latter was chosen because it ignores the phase differences between the target and estimated signals, which mimics human perception [26]. The resolutions for the spectrograms used are 32,768; 2,048; 512; and 128 samples. At a 44.1-kHz sampling rate, these correspond to windows of size 750, 50, 12, and 3 ms. An L1 loss is computed on these spectrograms, which is the absolute value of the difference between the spectrograms reduced across both the frame and frequency dimensions.

For a given mixdown, four separate reverse-engineering procedures are used that form an ablation study for the proposed system.

3.1.1 Full System

The first method utilizes the full mixing chain in Fig 1 and will be referred to as the "full system." Once parameters are initialized, the gradient descent with Adam optimizer and initial learning rate 10^{-4} first updates the gain, EQ, DRC, and panning parameters while bypassing the distortion and reverb modules for 40,000 iterations or until early stopping is reached. Afterward, distortion is introduced to the mixing chain and jointly updated with gain, EQ, DRC, and panning parameters for another 40,000 iterations or until early stopping is reached with learning rate 10^{-4}. Finally, reverb is introduced to the chain and all parameters are updated with a learning rate of 10^{-5} for another 40,000 iterations or until early stopping is reached.

3.1.2 DRC-EQ-Reverb System

The second method excludes the use of the distortion module and will be referred to as the "DRC-EQ-Reverb" system. To compensate, the gain, EQ, DRC, and panning parameters are allowed to update for 80,000 iterations with learning rate 10^{-4} or until early stopping is reached. The reverb module is then introduced, the learning rate dropped to 10^{-5}, and the gradient descent proceeds for 40,000 or until early stopping is reached.

3.1.3 EQ-Reverb System

The third method excludes both distortion and DRC, thus producing a linear mixdown similar to that of [8]. This system will be referred to as the "EQ-Reverb" system. The gain, EQ, and pan parameters are allowed to update for 80,000 iterations with a learning rate 10^{-4} or until early stopping is reached. Afterward, the reverb module is introduced, the learning rate dropped to 10^{-5}, and descent

Table 1. Multiscale spectrogram losses for reverse-engineered mixes. Rows in bold denote the top two performing reverse-engineered mixes using the full system for each song.

Mix	Full system	DRC-EQ-Reverb	EQ-Reverb	Gain Mix
00_A	1.2793	1.2840	1.2864	1.8987
01_A	1.8406	1.8683	1.9056	5.6233
02_A	1.5722	1.6723	1.7533	3.7127
03_A	**0.6690**	**0.7307**	**0.7295**	**1.3044**
04_A	1.5292	1.5734	1.5596	2.1578
05_A	1.1569	1.1815	1.2052	2.0023
08_A	**0.9358**	**0.9818**	**0.9945**	**1.5779**
17_A	1.9104	1.9148	1.9086	4.1754
06_B	**1.1986**	**1.3081**	**1.2858**	**2.8000**
07_B	2.7731	2.8395	2.8708	7.1260
09_B	1.9183	1.9733	1.9644	3.7378
10_B	5.9941	6.153	6.3025	11.0797
11_B	5.5800	5.9868	6.6175	9.5553
13_B	1.8684	2.0345	2.2830	4.2610
14_B	**1.5434**	**1.5575**	**1.5564**	**3.1416**
15_B	5.0719	5.2030	5.2575	8.6782
16_C	1.0712	1.0871	1.0842	1.8942
18_C	0.7481	0.8480	0.9614	2.1836
19_C	**0.6245**	**0.6469**	**0.6564**	**1.5719**
20_C	**0.5101**	**0.6606**	**0.6784**	**1.6957**
21_C	1.0388	1.2630	1.2925	2.4537
22_C	2.0082	2.0421	2.0868	4.5832
23_C	1.2846	1.3053	1.3028	2.0972

proceeds for another 40,000 iterations or until early stopping is reached.

3.1.4 Gain Mix System

The final method, which acts as the baseline for the study, uses only gain and panning. This mix is calculated using the least-squares estimate (LSE) method presented in [7] and is referred to as the "gain mix."

3.2 Dataset

The multitracks used to evaluate the system were taken from the Cambridge Multitracks dataset [27]. Three multitracks performed by the alt-rock band Woodfire for their Weird Fear EP were chosen because the multitracks were all recorded in the same studio, by the same engineer, under similar circumstances. The song "Animals" was recorded to 15 tracks, the song "Haunted House" to 14 tracks, and the song "Wealthy in Time" to 13 tracks.

These time-aligned multitracks were given to student mix engineers at Queen Mary University of London to mix as part of their coursework, with each student responsible for producing one mixdown of the multitrack they were assigned. Students were given 2 h to produce a mixdown of a song's first verse and chorus using the digital audio workstation of their choice and any plugins or automation they saw fit. Students were instructed to not alter the composition of any of the multitracks but were allowed to mute any elements in the mix. Eight students produced mixdowns of "Animals," another eight students produced mixdowns of "Haunted House," and seven students produced mixdowns of "Wealthy in Time."

There were 10 s from each song's chorus chosen for the reverse-engineering task. Thus, the system was tested against 23 mixdowns, 10 s in length, across three multitracks. Ultimately, 92 mixdowns were reverse engineered.

4 RESULTS

4.1 Objective Results

The final MSS loss calculated for each reverse-engineered mixdown can be found in Table 1. Mixes XX_A are mixes of the song "Animals," XX_B are mixes of the song "Haunted House," XX_C are mixes of "Wealthy in Time." Across all mixdowns, the full system performed best, and the gain mix performed worst. A total of 65% of the DRC-EQ-Reverb mixes outperformed the EQ-Reverb mixes on this objective measure. For the song "Animals," the full system performed best on the "03_A" and "08_A" mixdowns; for the song "Haunted House," the full system performed best on the "06_B" and "14_B" mixdowns; for the song "Wealthy in Time," the full system performed best on the "19_C" and "20_C" mixdowns.

4.2 Subjective Results

The top two performing reverse-engineered mixes of each multitrack were chosen for a listener evaluation using the webMUSHRA framework [28]. These mixdowns can be listened to at https://jtcolonel.github.io/NonlinRevEng/. All reference and reverse-engineered mixdowns were normalized to –24.0 LUFS-I for the listening test.

Although MUSHRA listening tests were originally formulated to assess "audio quality," [29] describes how they can be adapted to answer a researcher's specific question about the outputs of their audio algorithm. This includes providing participants with several listening examples to compare to a reference, including an anchor and the hid-

Table 2. Results of pairwise comparison of mixdown architecture on perceptual similarity rating across multitracks, with Bonferroni Correction.

	Reference Mix	Full Mix	DRC-EQ-Rev Mix	Linear Mix	Gain Mix
Reference Mix	·	o	*	*	*
Full Mix	o	·	*	*	*
DRC-EQ-Rev Mix	*	*	·	o	o
Linear Mix	*	*	o	·	o
Gain Mix	*	*	o	o	·

o = $p > 0.05$; * = $p < 0.001$; · = no comparison.

den reference. Furthermore, the expectations of what the participant is meant to be listening for should be clearly described before the test is taken.

Thus, participants in the listening test described below were presented with a reference mix and five stimuli, the four reverse-engineered mixdowns plus a hidden reference, across six reference mixes. Participants were asked to rate each stimuli according to how closely it matched the reference, with 0 representing a poor match and 1 representing a perfect match:

> You will be provided with a reference mixdown at the top of the page. The task is then to rate the stimuli below based on how closely they match the reference. When evaluating how close two mixdowns are, one should consider how the individual elements of the multitrack are balanced in each of the mixdowns. This balance may include how loud elements are compared to one another, how these elements are spread in the stereo field, the tone of each element, the dynamic characteristics of each element, and how the mixdown coheres as a whole. Adjust the sliders for each example to rate the closeness, and use the whole scale when possible. A perfect score should constitute a mixdown that exactly matches the reference.

A total of eight participants took part in the study, with an average age of 35 years and standard deviation of 7.08. Five participants identified as men, and three as women. Seven participants reported having at least 4 years of experience with music production or audio engineering, and one participant reported no experience. No participants reported any diagnosed hearing impairments. Box and whisker plots of the participants' ratings are presented in Fig. 4.

The analysis that follows is adapted from the perceptual study presented in [30]. The null hypothesis is that the perceptual evaluation scores are from the same distribution. A one-way ANOVA, with Bonferroni correction, shows for all stimuli that the effect each reverse-engineering method had on user perception was statistically significant.

With the null hypothesis rejected, a post-hoc Tukey pairwise comparison, with Bonferroni correction to reduce the chance of type I errors, was used. Table 2 shows the results of these pairwise comparisons for all architectures used.

The pairwise comparisons demonstrate that across the six selected mixdowns, the perception of the full system's reverse-engineered mixes do not differ significantly from the reference, and those of all other reverse-engineering methods do differ significantly from the reference. These

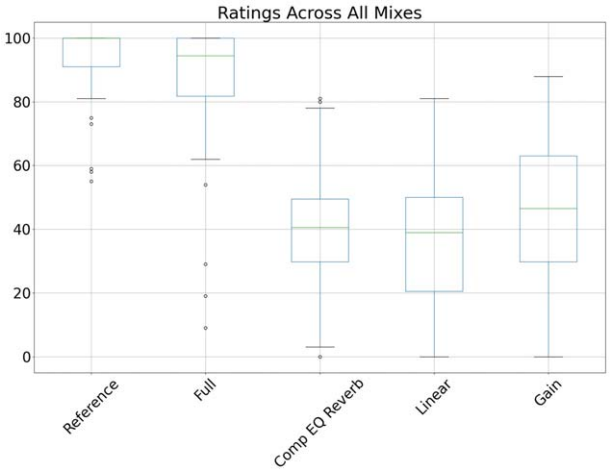

Fig. 4. Box and whisker plots of all participant ratings across all mixes.

results hold for all but two mixdowns: 06_B and 19_C. In the case of 06_B, the perception of all reverse-engineered mixes differ significantly from the reference. In the case of 19_C, the perception of both the full system and gain system's reverse-engineered mixes do not differ significantly from the reference.

5 DISCUSSION

The results of both the objective evaluation and listening test suggest that the full reverse-engineering mixing chain outperforms all other mixing chains. In terms of objective evaluation, the DRC-EQ-Reverb mix slightly outperforms the EQ-Reverb mix, with the gain mix performing the poorest. However, in terms of subjective evaluation, the gain mix had a higher average rating than both the DRC-EQ-Reverb mix and linear mix.

The listening test results for mix 06_B demonstrate that the MSS loss measure does not necessarily measure perceptual closeness—even though the full system's reverse-engineered 14_B mix had a greater MSS cost than that for 06_B, listener's rated the 14_B closer to its reference than 06_B. This contributes to a larger discussion in the literature regarding the need for more perceptually relevant cost functions for use in audio tasks. Differentiable mesostructural approaches like that shown in [31] or contrastive learning approaches like [32] may be more appropriate for this reverse-engineering task.

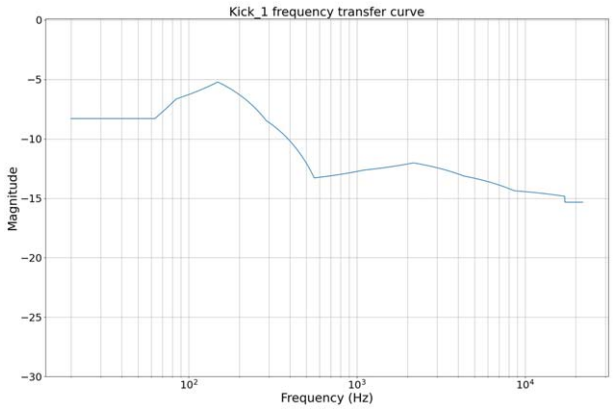

Fig. 5. Learned EQ of kick drum raw track in mix 20_C.

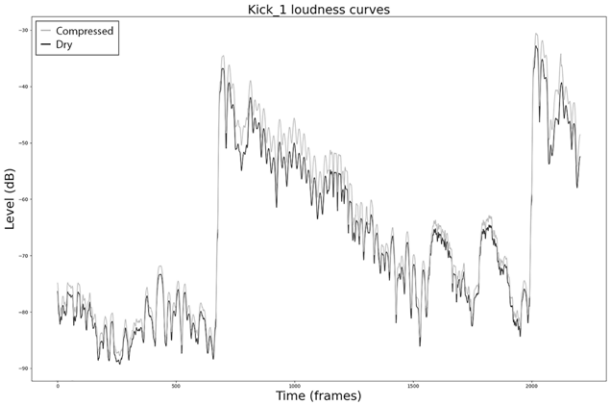

Fig. 6. Dry and compressed level curves of kick drum raw track in mix 20_C.

As well as providing mixdowns, students were asked to comment on the mixes of their peers. These comments provide potential explanations for what type of mixes the system performed poorly on. For example, across 16 comments on mix 10_B, the word "creative" appears in five. One evaluation mentions that the "singer seems like he is on a completely different stage from the band," and another mentions that "the reverb/delay on the vocal was not quite fit." This suggests that the reverb and delay used on the vocal is probably distinct than that used on the rest of the mix, which the full mixing chain is not equipped to handle. This may explain why mix 10_B performed worst on the MSS objective measure.

Students were not asked to submit their digital audio workstation session; students were only asked to submit their mixdowns. This means that direct comparison between reverse-engineered effects and the actual processing used by the students is not possible. However, for the reverse-engineered mixes that do reach perceptual tolerance, the stems of the mixes can be bounced individually and compared with the students' comments. For example, 11 of 17 students mention that the vocals in the mix are too low when commenting on mix 08_A. After bouncing the stems using the mixing chain learned by the full system, the vocals measure –29.1 LUFS-I, compared to the full mix minus vocals, which measures –23.8 LUFS-I. [1] found that listeners prefer when vocals sit between –2 and 0 LU compared to the rest of the mix.

For mix 20_C, several students commented that the kick drum is boomy and too prominent. The kick stem measures –25.6 LUFS-I, and the rest of the mix measures –26.0 LUFS-I. [1] found that the main element of the mix ought to be within –2 to 0 LU of the rest of the mix, so the relative loudness of the kick suggests that it will draw focus.

When observing the reverse-engineered mix's individual effects as seen in Fig. 5, the EQ on the kick has a slight boost between 60 and 300 Hz. In terms of spectrum, the descriptor "boomy" is typically applied to elements within the range of 20–250 Hz [33], with [34] specifically suggesting the range 60–250 Hz. Fig. 7 shows log frequency spectrograms of the raw kick track and processed kick stem. Here, it can be seen that the reverse-engineered effects processing increases the density of spectral energy below 200 Hz, which may explain the "boomy" comments. Furthermore, the learned compression has a threshold of –65.6 dB (the kick raw track has an average loudness of –65.3 dB), compression ratio of 5.56, makeup gain of 1.5 dB, and wet/dry of –3 dB dry/–3 dB wet. Although this is not a heavily compressed kick, these settings do indicate that the dynamic range of the kick drum has been compressed. The loudness curves of the dry and wet kick signals can be seen in Fig. 6.

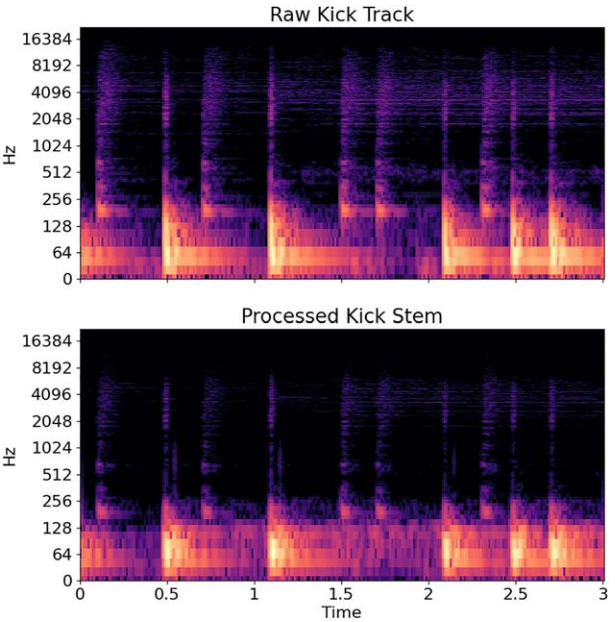

Fig. 7. Log frequency spectrogram of the raw kick track and processed kick stem for mix 20_C.

6 CONCLUSION

A method for reverse engineering a nonlinear multitrack mix with no automation has been proposed. This mixing chain contains differentiable greybox modules for gain, panning, EQ, delay, reverb, distortion, and dynamic range compression. Both objective and subjective results demon-

strate that the full signal chain outperforms a signal chain without distortion, a signal chain with only linear processing, and a signal chain with only gain and panning.

There are several directions future work can take. In the future a larger, more involved listening test could help better understand the differences between the mixdowns of each reverse-engineering system mentioned. Also, an evaluation of the legibility of learned parameters by mixing engineers would help to bolster claims of the proposed system's interpretability and spur further development of the presented modules.

Another direction is modeling automation, which is frequently used by mixing engineers. This could be realized using frame-by-frame approximations of mixing parameters or some other control scheme. Yet another direction would be a method for learning what mixing chain may best suit a mix, rather than fixing the chain shown in Fig 1. This could entail identifying which effects have been applied to the raw tracks in the mixdown, similar to the work of [35].

This reverse-engineering work may also aid in numerically characterizing mixing engineers' behavior by analyzing and extracting mix parameters from a corpus of professional mixes. This corpus could then be used to improve objective measures of multitrack mixes for perceptual correlation to avoid issues such as those encountered when objectively measuring mixdowns [6].

7 ACKNOWLEDGMENT

This work has been partially funded by the research division of the Yamaha Corporation.

8 REFERENCES

[1] P. Pestana, *Automatic Mixing Systems Using Adaptive Audio Effects*, Ph.D. thesis, Universidade Catolica Portuguesa, Lisbon, Portugal (2013 May).

[2] B. De Man, *Towards a Better Understanding of Mix Engineering*, Ph.D. thesis, Queen Mary University of London, London, UK (2017 Jan.).

[3] C. J. Steinmetz, *Learning to Mix With Neural Audio Effects in the Waveform Domain*, Master's thesis, Universitat Pompeu Fabra, Barcelona, Spain (2020 Sep.).

[4] E. Perez Gonzalez and J. Reiss, "A Real-Time Semiautonomous Audio Panning System for Music Mixing," *EURASIP J. Adv. Signal Process.*, vol. 2010, paper 436895 (2010 Dec.).

[5] A. Wilson and B. Fazenda, "Variation in Multitrack Mixes: Analysis of Low-Level Audio Signal Features," *J. Audio Eng. Soc.*, vol. 64, no. 7/8, pp. 466–473 (2016 Jul./Aug.).

[6] J. Colonel and J. D. Reiss, "Exploring Preference for Multitrack Mixes Using Statistical Analysis of MIR and Textual Features," presented at the *147th Convention of the Audio Engineering Society* (2019 Oct.), e-Brief 526.

[7] D. Barchiesi and J. Reiss, "Reverse Engineering of a Mix," *J. Audio Eng. Soc.*, vol. 58, no. 7/8, pp. 563–576 (2010 Jul./Aug.).

[8] J. T. Colonel and J. Reiss, "Reverse Engineering of a Recording Mix With Differentiable Digital Signal Processing," *J. Acoust. Soc. Am.*, vol. 150, no. 1, pp. 608–619 (2021 Jul.).

[9] J. T. Colonel, M. Comunità, and J. Reiss, "Reverse Engineering Memoryless Distortion Effects With Differentiable Waveshapers," presented at the *153rd Convention of the Audio Engineering Society* (2022 Oct.), paper 10626.

[10] J. T. Colonel, M. Comunità, and J. Reiss, "Approximating Ballistics in a Differentiable Dynamic Range Compressor," presented at the *153rd Convention of the Audio Engineering Society* (2022 Oct.), paper 33.

[11] J. Engel, L. Hantrakul, C. Gu, A. Roberts, "DDSP: Differentiable Digital Signal Processing," in *Proceedings of the International Conference on Learning Representations* (Addis Ababa, Ethiopia) (2020 Apr.).

[12] S. Nercessian, "Neural Parametric Equalizer Matching Using Differentiable Biquads," in *Proceedings of the 23rd International Conference on Digital Audio Effects (DAFx)*, pp. 265–272 (Vienna, Austria) (2020 Sep.).

[13] J. T. Colonel, C. J. Steinmetz, M. Michelen, and J. D. Reiss, "Direct Design of Biquad Filter Cascades With Deep Learning by Sampling Random Polynomials," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3104–3108 (Singapore) (2022 May).

[14] S. Lee, H.-S. Choi, and K. Lee, "Differentiable Artificial Reverberation," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 30, pp. 2541–2556 (2022 Jul.).

[15] S. Nercessian, A. Sarroff, and K. J. Werner, "Lightweight and Interpretable Neural Modeling of an Audio Distortion Effect Using Hyperconditioned Differentiable Biquads," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 890–894 (Toronto, Canada) (2021 Jun.).

[16] A. Wright and V. Välimäki, "Grey-Box Modelling of Dynamic Range Compression," in *Proceedings of the 25th International Conference on Digital Audio Effects (DAFx)*, pp. 304–311 (Vienna, Austria) (2022 Sep.).

[17] B. Hayes, C. Saitis, and G. Fazekas, "Neural Waveshaping Synthesis," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference (ISMIR)*, pp. 254–261 (Online) (2021 Nov.).

[18] S. Shan, L. Hantrakul, J. Chen, M. Avent, and D. Trevelyan, "Differentiable Wavetable Synthesis," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4598–4602 (Singapore) (2022 May).

[19] F. Caspe, A. McPherson, and M. Sandler, "DDX7: Differentiable FM Synthesis of Musical Instrument Sounds," *arXiv preprint arXiv:2208.06169* (2022 Aug.).

[20] S. Nercessian, "End-to-End Zero-Shot Voice Conversion Using a DDSP Vocoder," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 1–5 (New Paltz, NY) (2021 Oct.). https://doi.org/10.1109/WASPAA52581.2021.9632754.

[21] S. Nercessian, "Differentiable WORLD Synthesizer-Based Neural Vocoder With Application

To End-To-End Audio Style Transfer," *arXiv preprint arXiv:2208.07282* (2022 Aug.).

[22] C. J. Steinmetz, N. J. Bryan, and J. D. Reiss, "Style Transfer of Audio Effects with Differentiable Signal Processing," *arXiv preprint arXiv:2207.08759* (2022 Jul.).

[23] E. P. Gonzalez and J. D. Reiss, "Automatic Mixing: Live Downmixing Stereo Panner," in *Proceedings of the 7th International Conference on Digital Audio Effects (DAFx)*, pp. 63–68 (Bordeaux, France) (2007 Sep.).

[24] V. Välimäki and J. D. Reiss, "All About Audio Equalization: Solutions and Frontiers," *Appl. Sci.*, vol. 6, no. 5, paper 129 (2016 May). https://doi.org/10.3390/app6050129.

[25] X. Wang, S. Takaki, and J. Yamagishi, "Neural Source-Filter Waveform Models for Statistical Parametric Speech Synthesis," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 28, pp. 402–415 (2019 Dec.). https://doi.org/10.1109/TASLP.2019.2956145.

[26] T. Chi, P. Ru, and S. A. Shamma, "Multiresolution Spectrotemporal Analysis of Complex Sounds," *J. Acoust. Soc. Am.*, vol. 118, no. 2, pp. 887–906 (2005 Aug.).

[27] M. Senior, *Mixing Secrets for the Small Studio* (Taylor & Francis, New York, NY, 2011).

[28] M. Schoeffler, S. Bartoschek, F.-R. Stöter, et al., "webMUSHRA—A Comprehensive Framework for Web-Based Listening Tests," *J. Open Res. Softw.*, vol. 6, no. 1, paper 8 (2018 Feb.).

[29] B. De Man, R. Stables, and J. D. Reiss, *Intelligent Music Production* (Focal Press, Waltham, MA, 2019).

[30] D. Moffat and J. D. Reiss, "Perceptual Evaluation of Synthesized Sound Effects," *ACM Trans. Appl. Percept.*, vol. 15, no. 2, paper 13 (2018 Apr.).

[31] C. Vahidi, H. Han, C. Wang, et al., "Mesostructures: Beyond Spectrogram Loss in Differentiable Time-Frequency Analysis," *arXiv preprint arXiv:2301.10183* (2023 Jan.).

[32] P. Manocha, Z. Jin, R. Zhang, and A. Finkelstein, "CDPAM: Contrastive Learning for Perceptual Audio Similarity," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 196–200 (Toronto, Canada) (2021 Jun.).

[33] B. De Man and J. D. Reiss, "Analysis of Peer Reviews in Music Production," *J. Art Record Prod.*, vol. 10 (2015 Jul.).

[34] B. Owsinski, *The Mixing Engineer's Handbook* (Cengage Learning, Boston, MA, 2013), 3rd ed.

[35] J. Koo, M. A. Martinez-Ramirez, W.-H. Liao, et al., "Music Mixing Style Transfer: A Contrastive Learning Approach to Disentangle Audio Effects," *arXiv preprint arXiv:2211.02247* (2022 Nov.).

THE AUTHORS

Joseph T. Colonel

Joshua Reiss

Joseph T. Colonel is a final-year Ph.D. student in the Centre for Digital Music at Queen Mary University of London. His work focuses on applying machine learning and neural networks to music production behavior modeling. He received his Bachelor's and Master's degrees in electrical engineering from the Cooper Union in New York City, developing neural network audio effects for timbre interpolation and synthesis. He has also worked as an intern at Yamaha, working on singing voice synthesis, and Soundwide, working on intelligent audio production.

•

Josh Reiss is Professor of Audio Engineering with the Centre for Digital Music at Queen Mary University of London. He has published more than 200 scientific papers (including over 50 in premier journals and six best paper awards) and co-authored two books. His research has been featured in dozens of original articles and interviews on TV, on radio, and in the press. He is a Fellow and currently President of the Audio Engineering Society and chair of their Publications Policy Committee. He co-founded the highly successful spin-out company LandR and recently co-founded Tonz and Nemisindo, also based on his team's research. He maintains a popular blog, YouTube channel, and Twitter feed for scientific education and dissemination of research activities.