OPEN ACCESS Freely available online

The Mathematics of Mixing

MICHAEL TERRELL, *AES Member,* **ANDREW SIMPSON, AND MARK SANDLER,** *AES Fellow*
(michael.terrell@qmul.ac.uk)

Queen Mary University of London, London, UK

Mixing is a quintessential optimization problem. Given control of several component tracks, a balance must be struck that reflects a trade-off between engineering methods, artistic objectives, and auditory perceptual constraints. Formally, this balance can be thought of as the optimal solution to a system of mathematical equations that describe the relationships between the component tracks within a mix. Hence, the nature of these equations defines the process by which solutions may be arrived at. As perception is strongly nonlinear, an analytical solution to this set of equations is not possible and so a search must be conducted. Here, taking loudness as an example, we develop an optimization theory treatment of the problem of mixing, complete with case studies to illustrate how auditory perception can complicate the process, not least due to masking-related interactions.

0 INTRODUCTION

Humans have been solving optimization problems in music for several centuries at least. In earlier times, the conductor controlled the sound balance between musicians of the orchestra. More recently, the mixing engineer performs a similar role by electronic means. In both cases, the problem can be thought of as a set of mathematical equations with equivalent control parameters. In both cases, the solution is validated subjectively, reflecting perceptual and artistic objectives.

If this cognitive process (mixing) were purely analytical, the solution could be instantaneously arrived at (calculated) given sufficient processing (thinking) time. However, in practice, mixing embodies an iterative search process typical of Numerical Optimization Theory [1,2]. Therefore, numerical optimization theory provides an ideal framework for investigation into the process.

When balancing the various musical components, perhaps the most important subjective judgment criterion is loudness. Loudness provides a proxy to salience; the loudest component is typically perceived as the most salient. Hence, a critical control parameter is acoustic intensity, which the conductor controls through instructions to the musicians and the mixing engineer controls through adjustments to his mixing interface.

Recent work focused on automating the process of mixing has sought to replicate the human process of optimization using two principle components: (i) metric models to describe the objective and (ii) algorithms that manipulate control parameters on a mixing device to realize the objective. The majority of work uses linear models, where the sounds within a mix do not interact with each other [3,4, 5,6, 7,8, 9], though recent efforts have sought to include more sophisticated auditory models [10,11]. This provides for a simple, analytical solution that is computationally trivial. However, auditory theory tells us that many aspects of sound perception are nonlinear, and experience confirms that balancing music is a non-trivial problem. The complexity of our auditory system goes some way to explaining the iterative nature of mixing. For example, if the loudness of a sound was linearly related to its intensity we could solve a linear least square problem to identify the changes in intensity required to give a certain balance (relative loudness). However, because loudness is nonlinearly related to intensity, real-world loudness balancing is a nonlinear least-square problem, which must therefore be solved iteratively (Fig. 1).

The main contribution of this article is a formal treatment of the mixing process, centered on the role of auditory nonlinearity in the numerical optimization. Taking as example the objective of a predefined loudness balance, we develop the necessary theory to relate the human process to the mathematical framework. Along the way, we illustrate and realize the theory through computational analysis featuring an auditory model that incorporates both nonlinearity and masking interactions. Our computational case studies examine the nonlinear iterative nature of mixing, demonstrate how masking adds complexity to the problem, and illustrate how mixing for different sound systems and listening conditions is a problem of best-fit. More generally, we provide an intuitive tutorial on applied optimization for those in the field and through this we offer some insight into the human process of mixing.

1 MIX MATHEMATICS

In this section we define the model used to estimate loudness metrics of our mix and explain the optimization algorithm by which target metrics can be realized. In the

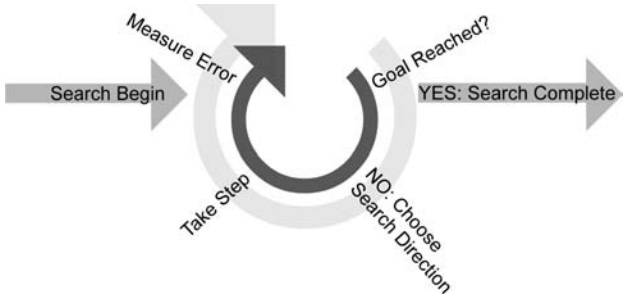


Fig. 1. Schematic diagram illustrating the iterative search process of numerical optimization.

following equations we represent scalar values by lower case variables (e.g., g_1 is the scalar gain applied to track 1), vectors by lower-case bold-face variables (e.g., $\mathbf{g}$ is a vector of scalar gain values), and matrices by upper-case bold-face variables (e.g., $\mathbf{E}$ is the spectro-temporal matrix describing auditory excitation).

1.1 Using Loudness to Describe the Mix

Mixing begins from an arbitrary starting point and adjustments are made after listening. The primary task is to set a good balance in loudness between component sounds by adjusting the intensity of the respective component tracks. A mix consists of n sources that have been pre-recorded onto n tracks. Track i therefore contains the unscaled audio signal from source i, given by $\mathbf{a_i}$. The mixing system can make changes to the intensity level of each track by applying gain, and these mix-state parameters are stored in the gain vector $\mathbf{g}$ (Eq. (1)). We refer to this gain vector, by convention, as the "fader gain" vector as it represents the positions of faders on a mixing console. For each of the n tracks, the respective acoustic signal (with units of pressure, Pa) reaching the listeners ears is $\mathbf{p_i}$, given by Eq. (2), where g_i is element i of the fader gain vector $\mathbf{g}$, and $\mathbf{h}$ is the impulse response of the reproduction system and room.

$$\mathbf{g} = \begin{bmatrix} g_1 & g_2 & g_3 & \dots & g_n \end{bmatrix}^T \tag{1}$$

$$\mathbf{p_i} = (g_i \mathbf{a_i}) * \mathbf{h} \tag{2}$$

Having represented the faders of the mixing console and the component acoustic signals, it remains to represent the listener (mixing engineer) who will guide the mixing process according to his perception. We use an excitation-pattern loudness model [12,13, 14] as a substitute for the listener to guide the intensity adjustments made to the mix (for a tutorial on using this model see [15]). The first stage of the loudness model applies a linear filter representing the effects of outer and middle ear. Next, the incoming signal is decomposed through "auditory filters" to represent the frequency-to-place transformation and frequency-tuning of the basilar membrane. We used a time-domain *gammatone* filter to decompose the signals [16], rather than the frequency domain approach given in [12]. This results in an instantaneous measurement of energy within a discrete number of frequency bands, each representing fibers of the auditory nerve, which together constitute the excitation pattern $\mathbf{E_i}$. Hence, the excitation pattern is a spectro-temporal matrix. We denote this operation by the function f as shown in Eq. (3).

$$\mathbf{E_i} = f\,(\mathbf{p_i}) \tag{3}$$

For a sound heard in isolation, the intensity represented in the excitation pattern matrix is converted into perceptual units in the form of a loudness time-series, $\mathbf{s_i}$, by compressing the signal in each frequency band (to reflect the action of the cochlea), integrating across frequency, and smoothing to reflect the time-response of the auditory system. We denote this by the function c as shown in Eq. (4).

$$\mathbf{s_i} = c\,(\mathbf{E_i}) \tag{4}$$

The combination of auditory filters and compression results in energetic (simultaneous) masking. When multiple sounds are heard concurrently Eq. (4) is adapted to account for the masking resulting from the respective excitation of the competing sounds, as given by Eq. (5). This partial attribution of excitation between two concurrent sounds is known as partial loudness. In such competing scenarios, if the magnitude of one excitation pattern is increased, it will reduce the partial loudness of the other.

$$\mathbf{s_i} = c\left[\mathbf{E_i},\, f\left(\sum_{j=1,\, j\neq i}^{n} \mathbf{p_j}\right)\right] \tag{5}$$

For each component track, the loudness time-series $\mathbf{s_i}$ is then converted into a single, scalar loudness measure l_i, which reflects the overall loudness impression over the duration of the measurement period; and that we refer to simply as the loudness. We denote this operation by the function μ because it represents some form of averaging of the loudness time function (Eq. (6)). A number of suggestions for this function have been made in the literature, including the peak [17], mean [13], and thresholded mean [18]. For the purposes of estimating the overall loudness relationships between the components of the mix, and hence describing the mix, the partial loudness of each component track is consolidated into the vector $\mathbf{l}$ (Eq. (7)).

$$l_i = \mu\,(\mathbf{s_i}) \tag{6}$$

$$\mathbf{l} = \begin{bmatrix} l_1 & l_2 & l_3 & \dots & l_n \end{bmatrix}^T \tag{7}$$

From the absolute loudness of each component track, ratios may be computed that describe the relationships in a way that is invariant with level. We define the loudness balance of a track, b_i, as its loudness relative to the mean loudness of all tracks, as shown in Eq. (8). This gives a loudness balance vector $\mathbf{b}$ (Eq. (9)), which we express in decibels to conform with engineering practice such that a difference of 3 dB means that one sound is twice as loud as another.

$$b_i = 10 \log_{10}\left(\frac{l_i}{\frac{1}{n}\sum_{j=1}^{n} l_j}\right) \tag{8}$$

$$\mathbf{b} = \begin{bmatrix} b_1 & b_2 & b_3 & \dots & b_n \end{bmatrix}^T \tag{9}$$

Given a description based on loudness ratios, it is clear that the same balance could be produced for a very quiet or loud mix. This is problematic because it means that the number of possible solutions is infinite. So in order to make our mix description unique, we add a further constraint; the loudness of the mix as a whole. This is evaluated using Eqs. (2) to (6), substituting $\mathbf{a_m}$ for $\mathbf{a_i}$ in Eq. (2), where $\mathbf{a_m}$ is the summation of all audio signals (Eq. (10)). We therefore describe our mix using the loudness balance vector b, and the loudness of the whole mix, l_m.

$$\mathbf{a_m} = \sum_{i=1}^{n} \mathbf{a_i} \tag{10}$$

1.2 Solving the Mix

We use our loudness model to identify the fader (gain) settings required to produce a mix with a predefined balance and mix loudness. The nonlinearity in the loudness model means that an analytical solution is not possible, hence numerical optimization must be used. If we begin with arbitrary (random) fader gain settings, the loudness metrics of our mix will differ from the target objective (unless we are very lucky!). We represent these differences as errors (signed), for each track, denoted by e_i (Eq. (11)), and the error in mix loudness is denoted by e_m (Eq. (12)), where the subscript t identifies the target metric. These are consolidated into the overall error vector $\mathbf{e}$ (Eq. (13)).

$$e_i = b_i - b_{t_i} \tag{11}$$

$$e_m = l_m - l_{t_m} \tag{12}$$

$$\mathbf{e} = \begin{bmatrix} e_1 & e_2 & e_3 & \dots & e_n & e_m \end{bmatrix}^T \tag{13}$$

Each element of $\mathbf{e}$ is a function of the fader gain vector $\mathbf{g}$, so the total error, $e_T(\mathbf{g})$, can be expressed as the sum of squares (i.e., the sum of squared errors) of $\mathbf{e}$ (Eq. (14)). The minimum of e_T, i.e., the point where our mix is closest to the target mix, can be found iteratively using a nonlinear least squares algorithm, which has the general form given in Eq. (15); where q is the iteration index. $\triangle_{\mathbf{g}}$ is the search direction in the fader gain vector where the error is decreasing most rapidly, and the step size, $0 < \lambda_q \leq 1$, determines how far along that search direction we move during a given iteration.

$$e_T(\mathbf{g}) = \sum_{i=1}^{n} e_i^2(\mathbf{g}) = \mathbf{e}(\mathbf{g})^T \mathbf{e}(\mathbf{g}) \tag{14}$$

$$\mathbf{g}_{q+1} = \mathbf{g}_q + \lambda_q \triangle_{\mathbf{g}_q} \tag{15}$$

The search direction is typically found using a numerical method derived using a Taylor series expansion, e.g., Newton's method (second order expansion). We choose to use the Gauss-Newton method, which is based on Newton's method but which uses an approximation to the second derivative in the expansion and reduces computation time for problems solved numerically (see [1,2] for details on numerical methods). The Gauss-Newton method produces a system of equations known as the *normal equations*, given in Eq. (16), which can be rearranged to give $\triangle_{\mathbf{g}}$ as in Eq. (17), where: $\mathbf{J_q}$ is the Jacobian matrix at iteration q, and $\left(\mathbf{J}^T\mathbf{J}\right)^{-1}\mathbf{J}^T$ is the *Moore-Penrose* pseudo inverse of $\mathbf{J}$; the inverse of a non-square matrix [19]. The Jacobian matrix contains the partial derivatives of each component of the error function with respect to each parameter, as shown in Eq. (18).

$$\left(\mathbf{J_q}^T\mathbf{J_q}\right) \triangle_{\mathbf{g}_q} = -\mathbf{J_q}^T\mathbf{e_q}. \tag{16}$$

$$\triangle_{\mathbf{g}_q} = -\left(\mathbf{J_q}^T\mathbf{J_q}\right)^{-1}\mathbf{J_q}^T\mathbf{e_q}, \tag{17}$$

$$\mathbf{J} = \begin{bmatrix} \frac{\partial e_1}{\partial g_1} & \frac{\partial e_1}{\partial g_2} & \dots & \frac{\partial e_1}{\partial g_n} \\ \frac{\partial e_2}{\partial g_1} & \frac{\partial e_2}{\partial g_2} & \dots & \frac{\partial e_2}{\partial g_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial e_n}{\partial g_1} & \frac{\partial e_n}{\partial g_2} & \dots & \frac{\partial e_n}{\partial g_n} \\ \frac{\partial e_m}{\partial g_1} & \frac{\partial e_m}{\partial g_2} & \dots & \frac{\partial e_m}{\partial g_n} \end{bmatrix} \tag{18}$$

Eq. (17) has the same form as a linear least-squares problem, i.e., $\mathbf{Ax} = \mathbf{b} \rightarrow \mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$, where $\mathbf{A}$ is non-square. Each iteration of the Gauss-Newton method can therefore be viewed as a linear least-square problem, where the search vector $\triangle_{\mathbf{g}}$ is the estimated change in gain that will minimize the error. The actual change in the error will depend on how accurately the normal equations describe the behavior of our system (of equations). By using the Gauss-Newton method we have already truncated the Taylor series expansion at second order terms and have used an approximation to the second derivative. If these approximations introduce no error, e.g., if the system of equations are in fact linear, then the minimum would be found (calculated) in a single iteration; but if the errors are large, it may take many iterations to converge. Depending on the nature of the search space and the approximations discussed above, it is possible that the search vector will produce gain changes that overshoot the minimum and which may even increase the error. The step length parameter λ_q may be used to mitigate this effect by giving a shorter, more cautious step toward the minimum; in effect, damping the changes in our parameters. While this may avoid potential overshoots (and accompanying oscillations), it will inevitably lead to an increase in the number of iterations before convergence. When implementing an optimization algorithm a fixed or a variable step length may be used. For the latter, the optimal step length at each iteration is found using a second, small-scale optimization algorithm, e.g., a direct search method [1,2]; but this inevitably increases computational demands.

1.3 Summary

In this section we have outlined the two key components of an automatic mixing system: a model to describe our chosen mix metrics (loudness balance and mix loudness) in terms of the control parameters of our mixing system (fader gain), and an optimization algorithm that will find the fader gain values to produce our target mix metrics.

The model is a general excitation pattern loudness model, which includes the nonlinearities of the human auditory system. The algorithm uses the Gauss-Newton method, which uses derivative information at each iteration to estimate the changes in gain that will minimize the error. This process is repeated iteratively until the minimum of the error function is found and, hence, our mix metrics are as close to the target mix as possible.

2 CASE STUDY: MIX OPTIMIZATION IN ACTION

We demonstrate mix optimization using an excerpt from an 8-track recording of a rock band. The band included voice, lead guitar, piano, bass guitar, kick drum, snare drum, hi-hats, and crash cymbal; and the audio signal for each was stored on a separate track. To simplify this example we assumed that the frequency response of the reproduction system was perfectly flat and invariant with level (i.e., linear); we modelled the impulse response **h** as a scalar calibration constant h, converting the audio signals—which in the digital domain are typically represented using values between –1 and 1—into an acoustic signal with units of pressure. We set $h = 1$, so an audio signal with a peak of 1 produced an acoustic signal with peak pressure of 1 Pa (94 dB SPL). Each audio signal was initially peak normalized and had a starting track fader gain of 0 dB, which corresponded to a peak level of 94 dB SPL. Our target loudness balance was set to provide equal loudness for all tracks (Eq. (19)), and the target mix loudness was set to 30 sones ($l_{t_m} = 30$), which is equivalent to a 1 kHz tone with an RMS level of approximately 90 dB SPL. We used the procedure outlined in Section 1.1 to evaluate loudness time-series, **s**, and set the function $\mu(\mathbf{s})$ as the mean of that time series.

$$\mathbf{b_t} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{19}$$

Assuming that the process does not begin with the optimal solution, the first step in the optimization algorithm requires evaluation of the search direction at the starting position (Fig. 1). **J** cannot be evaluated analytically and so must be estimated using a numerical, finite difference method, e.g., the forward difference (Eq. (20)) or central difference (Eq. (21)), both of which perturb the current gain vector by a finite amount, δ_{g_k} and calculate the change in the error. The latter is a more accurate approximation but requires an additional function evaluation for each element of **J**, and to increase the speed of our algorithm we selected the forward difference method.

$$\frac{\partial e_i}{\partial g_k} \approx \frac{e_i(\mathbf{g} + \delta_{g_k}) - e_i(\mathbf{g})}{\delta_{g_k}} \tag{20}$$

$$\frac{\partial e_i}{\partial g_k} \approx \frac{e_i(\mathbf{g} + \delta_{g_k}) - e_i(\mathbf{g} - \delta_{g_k})}{2\delta_{g_k}} \tag{21}$$

The target loudness balance is shown in Eq. (19), the loudness balance at our starting point (iteration 0) is shown in Eq. (22), and the overall mix loudness $l_{m_{q=0}} = 15.0$ sone. The starting error is shown in Eq. (23), and the corresponding Jacobian in Eq. (24).

$$\mathbf{b_{q=0}} = \begin{bmatrix} -5.9 & 8.4 & -13.1 & -9.0 & -27.3 & -13.7 & -3.8 & -8.1 \end{bmatrix}^T \tag{22}$$

$$\mathbf{e_{q=1}} = \begin{bmatrix} -5.9 & 8.4 & -13.1 & -9.0 & -27.3 & -13.7 & -3.8 & -8.1 & 15.0 \end{bmatrix}^T \tag{23}$$

$$\mathbf{J_{q=1}} = \begin{bmatrix} \mathbf{0.45} & -0.37 & -0.01 & -0.03 & -0.00 & 0.00 & -0.03 & -0.00 \\ -0.01 & \mathbf{0.05} & -0.00 & -0.01 & 0.00 & -0.00 & -0.01 & -0.01 \\ -0.03 & -0.46 & \mathbf{0.58} & -0.08 & -0.01 & 0.01 & -0.00 & -0.01 \\ -0.04 & -0.41 & -0.13 & \mathbf{0.65} & -0.05 & -0.01 & -0.01 & -0.00 \\ -0.03 & -0.44 & -0.13 & -0.52 & \mathbf{1.18} & 0.01 & -0.03 & -0.00 \\ -0.02 & -0.46 & -0.03 & -0.09 & 0.00 & \mathbf{0.69} & -0.10 & -0.00 \\ -0.02 & -0.22 & 0.01 & 0.01 & 0.00 & -0.03 & \mathbf{0.28} & -0.01 \\ 0.00 & -0.28 & 0.01 & 0.01 & 0.00 & 0.00 & -0.03 & \mathbf{0.29} \\ 0.07 & 0.79 & 0.04 & 0.09 & 0.02 & 0.03 & 0.08 & 0.03 \end{bmatrix} \tag{24}$$

The search direction, $\triangle_{\mathbf{g}}$ can then be found using Eq. (17) and is shown in Eq. (25) for a step size $\lambda_{q=0} = 1$. This process is repeated, with **J**, **e** and $\triangle_{\mathbf{g}}$ is re-calculated at each iteration, until a predefined tolerance has been reached (that in our case was $e_T = 0.01$). This represents the limit at which errors resulting from such small changes in fader gain settings are imperceptible. Fig. 2**a** shows the evolution in the total error over time, and Fig. 2**b** shows the fader gain values. The error drops rapidly (note the logarithmic y-axis) and by the 4th iteration is below our tolerance threshold. The rapid reduction in error, and the absence of oscillations in the gain parameters, suggest that the approximations we made in deriving the normal equations do not introduce substantial errors, and that in this case it is safe to use the maximum step size (though this may not be guaranteed for any song).

$$\triangle_{\mathbf{g_1}} = \begin{bmatrix} -2.8 & -20.4 & 7.1 & 3.7 & 17.8 & 7.0 & -0.8 & 8.0 \end{bmatrix}^T \tag{25}$$

2.1 Why Mix Loudness Is Important

In defining the mix metrics in Section 1.1 we included the mix loudness as well as the loudness balance to ensure that our mix had a unique description; but how does this relate to our optimization algorithm? As already discussed, each iteration can be viewed as a linear least-squares problem of the form $\mathbf{Ax} = \mathbf{b}$, which represents a system of simultaneous equations. The number of rows in **A** is the number of equations, and the number of columns is the number of parameters. If there are more independent equations than parameters the system is *over-determined*, and the solution will reflect the best fit, i.e., the solution with the least-square error. In this example, **J** has 9 rows and 8 columns, so we would expect the system to be *over-determined*; however, the fact that a given loudness balance can theoretically be achieved for any overall mix loudness suggests that there is linear dependency between the corresponding rows of **J** (rows 1 to 8). The rank of a matrix tells us how many independent rows there are, and $rank(\mathbf{J}) = 8$; hence though we

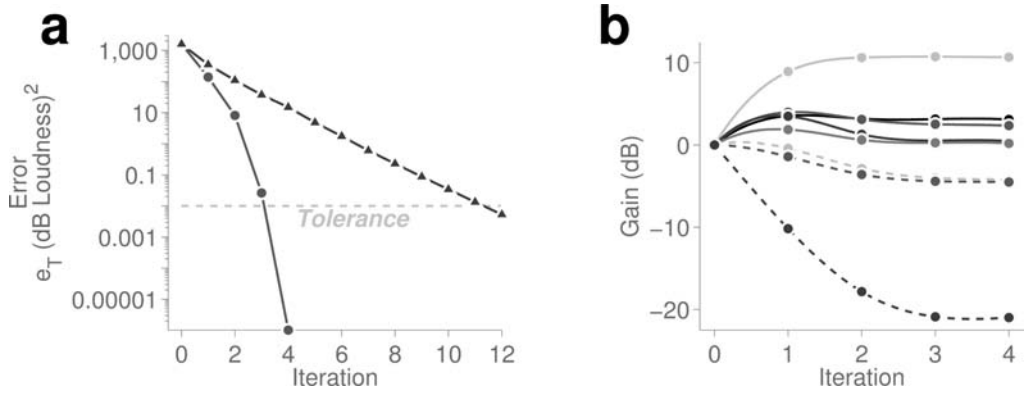


Fig. 2. Illustration of the converging search, showing values at each iteration in the optimization algorithm. **a** plots the total error e_T (circles where masking interactions are included, triangles where masking interactions are excluded), **b** plots the fader gain values for each track.

have 9 equations, we only have 8 *independent* equations. If we didn't include the mix loudness within the optimization algorithm, we would have a square matrix of 8 equations, but only 7 would be independent. **J** would therefore be rank deficient, and the system of equations it describes are *under-determined* so a unique solution would not exist. A square, rank deficient matrix (also known as a singular matrix) cannot be inverted and is analogous to dividing a scalar by zero. However, the approximations we made in using finite difference methods means our matrix is not perfectly singular, so it can be inverted, but will result in very large gain changes, which in the scalar analogy can be thought of as division by zero plus a small offset error. If we rerun the algorithm with this reduced set of equations our initial search vector is that shown in Eq. (26); adding 14,000 dB of gain to each track is clearly not going to yield a very useful mix!

$$\triangle_{\mathbf{g_{q=0}}} = [1.38 \;\; 1.39 \;\; 1.43 \;\; 1.41 \;\; 1.37 \;\; 1.37 \;\; 1.31 \;\; 1.37]^T \times 10^4 \tag{26}$$

2.2 Summary

This case study demonstrates the process of automatic mixing and provides some crude insight into how humans perform this task. From a random starting point we calculate the Jacobian and the error, which we then use to evaluate the search vector $\triangle_{\mathbf{g}}$, which is an estimate of the gain changes needed to minimize the error. Due to the nonlinearities in the model, this estimate will not be exact, so the process must be repeated until convergence is reached. We have also highlighted the fact that we must include the mix loudness constraint if we want to obtain a usable mix; and more generally, that our system of equations used to describe the mix allows for a unique solution to be found.

3 WHAT MAKES MIXING DIFFICULT: MASKING AND THE SENSITIVITY MATRIX

Each element of **J** describes the rate of change in one component (track) of the error, with respect to the change in one fader gain parameter, i.e., $J_{i,k} = \frac{\partial e_i}{\partial g_k}$. For now we consider $i = 1 \ldots n$, where the error relates to the loudness balance of the mix (for $i = n + 1$ the error relates to the mix loudness). The elements of **J** can be expanded as shown in Eq. (27), in which the term $\frac{\partial b_{t_i}}{\partial g_k} = 0$, because the target balance does not change with gain.

$$J_{i,k} = \frac{\partial e_i}{\partial g_k} = \frac{\partial b_i}{\partial g_k} - \frac{\partial b_{t_i}}{\partial g_k} = \frac{\partial b_i}{\partial g_k} \tag{27}$$

The loudness balance was defined in Eq. (8) in terms of the loudness of each track and is repeated here for convenience. Elements of **J** can therefore be expressed in terms of loudness as in Eq. (29).

$$b_i = 10 \log_{10} \left(\frac{l_i}{\frac{1}{n} \sum_{j=1}^{n} l_j} \right) \tag{28}$$

$$J_{i,k} = \frac{\partial b_i}{\partial g_k} = \frac{\partial}{\partial g_k} \left[10 \log_{10} \left(\frac{l_i}{\frac{1}{n} \sum_{j=1}^{n} l_j} \right) \right] \tag{29}$$

The logarithm with base 10 is converted to the natural logarithm using $\log_a(x) = \frac{\log_e(x)}{\log_e(a)}$ and the numerator and denominator are separated (Eq. (30)).

$$\begin{aligned} J_{i,k} &= \frac{\partial b_i}{\partial g_k} \\ &= \frac{10}{\log_e(10)} \frac{\partial}{\partial g_k} \left[\log_e(l_i) - \log_e \left(\frac{1}{n} \sum_{j=1}^{n} l_j \right) \right] \end{aligned} \tag{30}$$

The two logarithmic functions are differentiated using $f'(\log_e(x)) = \frac{f'(x)}{f(x)}$, where $S_{i,k} = \frac{\partial l_i}{\partial g_k}$ (Eq. (31)), and the $\frac{1}{n}$ terms are cancelled (Eq. (32)).

$$J_{i,k} = \frac{\partial b_i}{\partial g_k} = \frac{10}{\log_e(10)} \left[\frac{S_{i,k}}{l_i} - \frac{\frac{1}{n} \sum_{j=1}^{n} S_{j,k}}{\frac{1}{n} \sum_{j=1}^{n} l_j} \right] \tag{31}$$

$$J_{i,k} = \frac{\partial b_i}{\partial g_k} = \frac{10}{\log_e(10)} \left[\frac{S_{i,k}}{l_i} - \frac{\sum_{j=1}^{n} S_{j,k}}{\sum_{j=1}^{n} l_j} \right] \tag{32}$$

For $i = n + 1$, the error directly relates to the overall mix loudness, so the final row of **J** can be expressed using Eq. (33).

$$J_{n+1,k} = \frac{\partial e_m}{\partial g_k} = \frac{\partial l_m}{\partial g_k} - \frac{\partial l_{t_m}}{\partial g_k} = \frac{\partial l_m}{\partial g_k} \tag{33}$$

The matrix **S** has the same form as **J** and is referred to as the *sensitivity matrix* because it shows how sensitive the loudness of each track and the overall mix are to changes in fader gain (Eq. (34)). The sensitivity matrix at the start point of the previous case study is shown in Eq. (35). The diagonal terms in the top 8 rows of the matrix (highlighted in bold) show how the loudness of a track is related to changes to its fader gain. Adding gain will increase the intensity of the excitation, which in turn will increase loudness, so the diagonal elements of **S** are positive. The off-diagonal terms show how the loudness of a track interacts with changes in the fader gain applied to other tracks. These interactions must be due to masking, because the only mechanism by which l_i can be effected by g_k is through a change in the competing excitation pattern in Eq. (5). An increase of energy in the competing excitation pattern will cause more energetic masking, so l_i will decrease and the off-diagonal elements will be negative.

$$\mathbf{S} = \begin{bmatrix} \frac{\partial l_1}{\partial g_1} & \frac{\partial l_1}{\partial g_2} & \cdots & \frac{\partial l_1}{\partial g_n} \\ \frac{\partial l_2}{\partial g_1} & \frac{\partial l_2}{\partial g_2} & \cdots & \frac{\partial l_2}{\partial g_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial l_n}{\partial g_1} & \frac{\partial l_n}{\partial g_2} & \cdots & \frac{\partial l_n}{\partial g_n} \\ \frac{\partial l_m}{\partial g_1} & \frac{\partial l_m}{\partial g_2} & \cdots & \frac{\partial l_m}{\partial g_n} \end{bmatrix} \tag{34}$$

$$\mathbf{S}_{\mathbf{q=1}} = \begin{bmatrix} \mathbf{0.09} & -0.05 & -0.00 & -0.01 & -0.00 & -0.00 & -0.01 & -0.00 \\ -0.12 & \mathbf{1.08} & -0.07 & -0.10 & -0.01 & -0.06 & -0.09 & -0.03 \\ -0.00 & -0.01 & \mathbf{0.02} & -0.00 & -0.00 & -0.00 & -0.00 & -0.00 \\ -0.00 & -0.03 & -0.01 & \mathbf{0.07} & -0.01 & -0.00 & -0.00 & -0.00 \\ -0.00 & -0.00 & -0.00 & -0.00 & \mathbf{0.00} & -0.00 & -0.00 & -0.00 \\ -0.00 & -0.01 & -0.00 & -0.00 & -0.00 & \mathbf{0.02} & -0.00 & -0.00 \\ -0.01 & -0.03 & -0.00 & -0.00 & -0.00 & -0.01 & \mathbf{0.09} & -0.00 \\ -0.00 & -0.02 & -0.00 & -0.00 & -0.00 & -0.00 & -0.00 & \mathbf{0.04} \\ 0.07 & 0.79 & 0.04 & 0.09 & 0.02 & 0.03 & 0.08 & 0.03 \end{bmatrix} \tag{35}$$

3.1 Mixing Without Masking

If we were to assume that there were no masking interactions (off-diagonal elements of **S** are zero), then the diagonal elements of **J** could be simplified to Eq. (36) and the off-diagonal terms to Eq. (37). Adding fader gain to track k would then increase the attribution of balance to track k and decrease it for the other tracks by *equal amounts*. This is a critical point because it would mean that changing the fader gain of a track would *not* affect the loudness balance *between* the other tracks, i.e., the seven instruments within a mix could be balanced, and upon setting the vocal gain, the balance between the instruments would *not* change. This independence between tracks would make the process of mixing simpler, but as shown in Eq. (35) there *are* masking interactions, so changing the fader gain of a track *will* affect the balance between the other tracks. This can been understood numerically by inspecting **J** (Eq. (24)) and **S** (Eq. (35)). $S_{1,2} = -0.05$, and $S_{1,3} = -0.00$, which means that changes in g_1 have a far bigger effect on l_2 compared to l_3, i.e., there are strong masking interactions between tracks 1 and 2, but not 1 and 3. This is mirrored in the Jacobian, with $J_{1,2} = -0.37$ and $J_{1,3} = -0.01$. This means that adding 1 dB of fader gain to track 1 will reduce the balance of track 2 by 0.37 dB, but will hardly change the balance of track 3, i.e., the loudness of track 2 drops by 0.37 dB relative to track 3 if I add 1 dB gain to track 1. Understanding these interactions offers some insight into the human mixing process.

$$J_{i=k,k} = \frac{\partial b_i}{\partial g_k} = \frac{10}{\log_e(10)} \left[\frac{S_{k,k}}{l_k} - \frac{S_{k,k}}{\sum_{j=1}^{n} l_j} \right] \tag{36}$$

$$J_{i\neq k,k} = \frac{\partial b_i}{\partial g_k} = \frac{10}{\log_e(10)} \left[-\frac{S_{k,k}}{\sum_{j=1}^{n} l_j} \right] \tag{37}$$

3.2 Summary: Man vs Machine

Intuitively, a difficulty for a human mixing engineer (or conductor) would be to maintain the above information/matrices for the many possible channels involved in a typical mixing scenario. A second problem would be to adjust multiple fader gains simultaneously. To assist with this problem, the practice of subgrouping has emerged, where like-channels are combined, according to their own independent sub-mix, and controlled with a single fader movement. However, the masking interactions illustrated above mean that the balance within a sub-mix will be altered when subsequent tracks are added to the mix, or when their gain is modified, which in turn will necessitate further corrections to obtain the prior sub-mix balance. As a result, the human mixing process is inherently iterative. With an automated mix algorithm the interaction terms can be evaluated and stored, offering advantages over the human process. If these terms are ignored (i.e., forgotten or impossible to compute) then the algorithm will behave more like a human and will require more iterations to converge. Fig. 2**a** illustrates this point by showing the progression of the mix optimization from the previous case study compared to the same process but with off-diagonal elements in the top eight rows of **S** set to zero. The algorithm still converges to the same solution, but it takes three times as many iterations to get there.

4 ONE MIX FITS ALL: MIXING FOR MULTIPLE LISTENING CONDITIONS

The roles of mixing engineer and conductor share a further difficulty—the requirement that the balance should be optimal for multiple listening conditions at the same time. In the orchestral hall, the conductor attempts to optimize the balance of the orchestra for reception at seats from the front-row to the back-row and even balcony. In the recording studio, the mixing engineer attempts to optimize the mix for *any* possible listening condition, from the kitchen

Table 1.

Repro. System	Tgt. Mix Loud. (sones)	1kHz Calib. (dBSPL/dBFS)	Mix Error (dB sone)^2
Studio	30	104	4.7
Kitchen Radio	10	82	4.2
Headphones	30	101	8.4
Auditorium	40	114	8.9

radio to the dance-hall PA system, and must face the problem of each playback system featuring a "volume" control. In this section we illustrate how this extends the optimization problem to one of "best fit" and how, in the worst case, the best fit can amount to a poor compromise.

amount. With n tracks there will be n independent equations per system (in our case study there were 8), which gives $(n \times r)$ in total. Our track gain controls are common across all reproduction systems, but we have an additional master gain control for each, giving $(n + r)$ parameters in total.

$$\mathbf{J} = \left[\begin{array}{cccccccc}
\frac{\partial e_{1_1}}{\partial g_1} & \frac{\partial e_{1_1}}{\partial g_2} & \cdots & \frac{\partial e_{1_1}}{\partial g_n} & \frac{\partial e_{1_1}}{\partial g_{m_1}} & 0 & \cdots & 0 \\
\vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\
\frac{\partial e_{n_1}}{\partial g_1} & \frac{\partial e_{n_1}}{\partial g_2} & \cdots & \frac{\partial e_{n_1}}{\partial g_n} & \frac{\partial e_{n_1}}{\partial g_{m_1}} & 0 & \cdots & 0 \\
\frac{\partial e_{m_1}}{\partial g_1} & \frac{\partial e_{m_1}}{\partial g_2} & \cdots & \frac{\partial e_{m_1}}{\partial g_n} & \frac{\partial e_{m_1}}{\partial g_{m_1}} & 0 & \cdots & 0 \\
\hline
\frac{\partial e_{1_2}}{\partial g_1} & \frac{\partial e_{1_2}}{\partial g_2} & \cdots & \frac{\partial e_{1_2}}{\partial g_n} & 0 & \frac{\partial e_{1_2}}{\partial g_{m_2}} & \cdots & 0 \\
\vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\
\frac{\partial e_{n_2}}{\partial g_1} & \frac{\partial e_{n_2}}{\partial g_2} & \cdots & \frac{\partial e_{n_2}}{\partial g_n} & 0 & \frac{\partial e_{n_2}}{\partial g_{m_2}} & \cdots & 0 \\
\frac{\partial e_{m_2}}{\partial g_1} & \frac{\partial e_{m_2}}{\partial g_2} & \cdots & \frac{\partial e_{m_2}}{\partial g_n} & 0 & \frac{\partial e_{m_2}}{\partial g_{m_2}} & \cdots & 0 \\
\vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\
\hline
\vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\
\frac{\partial e_{1_r}}{\partial g_1} & \frac{\partial e_{1_r}}{\partial g_2} & \cdots & \frac{\partial e_{1_r}}{\partial g_n} & 0 & 0 & \cdots & \frac{\partial e_{1_r}}{\partial g_{m_2}} \\
\vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\
\frac{\partial e_{n_r}}{\partial g_1} & \frac{\partial e_{n_r}}{\partial g_2} & \cdots & \frac{\partial e_{n_r}}{\partial g_n} & 0 & 0 & \cdots & \frac{\partial e_{n_r}}{\partial g_{m_r}} \\
\frac{\partial e_{m_r}}{\partial g_1} & \frac{\partial e_{m_r}}{\partial g_2} & \cdots & \frac{\partial e_{m_r}}{\partial g_n} & 0 & 0 & \cdots & \frac{\partial e_{m_r}}{\partial g_{m_r}}
\end{array}\right], \mathbf{e} = \left[\begin{array}{c}
e_{1_1} \\ \vdots \\ e_{n_1} \\ e_{m_1} \\ \hline e_{1_2} \\ \vdots \\ e_{n_2} \\ e_{m_2} \\ \vdots \\ \hline \vdots \\ e_{1_r} \\ \vdots \\ e_{n_r} \\ e_{m_r}
\end{array}\right], \mathbf{g} = \left[\begin{array}{c}
g_1 \\ g_2 \\ \vdots \\ g_n \\ g_{m_1} \\ g_{m_2} \\ \vdots \\ g_{m_r}
\end{array}\right] \tag{38}$$

The metrics used to describe our mix are the loudness balance and overall mix loudness. The loudness model used to evaluate these metrics operates on the acoustic signals, which are evaluated by applying fader gain, and convolving the audio signals with the impulse response of the reproduction system and room (Eq. (2)); which includes scaling of signal level in units of pressure. The subsequent nonlinearities in the model mean that the impulse response may have a strong effect on the mix metrics, even if the fader gain controls are kept constant. This is a common problem for mixing engineers who want their mix to sound "the same" in any listening environment and at any listening level, and their solution is to listen to a mix on multiple systems and to correct a mix with further adjustments if necessary. This is, in effect, a further layer of iteration between reproduction systems.

This situation can be understood by considering our system of equations. We assume that a mix is to be reproduced on r systems and that our mix metrics should be recreated on each. To account for the fact that each system will have its own volume control, we add a master gain control for each system that scales each track in the mix by an equal

This means that when $r > 1$ we will have more equations than parameters, and that the ratio between equations and parameters will increase with r. As a result, it will be impossible to produce a mix with the same metrics on multiple reproduction systems.

4.1 True Best-fit for Multiple Conditions

The algorithm we have presented is well suited to provide true best-fits for mixes on multiple reproduction systems by expanding our equations as shown in Eq. (38). Each block of **J** reflects error derivatives for different listening conditions. Columns 1 to n contain the loudness balance derivatives, and there is a column per condition to account for an individual mix loudness, and a device-specific volume control. The error is of a similar form, with one block per condition, and the gain vector contains the existing fader controls and a master (volume) control for each condition.

We demonstrate this idea of best-fit mixing using four parallel listening scenarios, each with different acoustic conditions. The first scenario is an idealized recording studio featuring a flat frequency response at a moderate

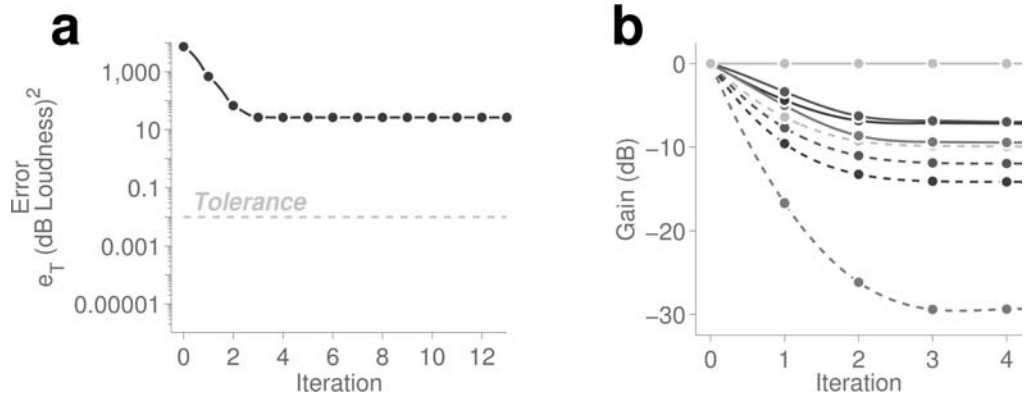


Fig. 3. Mixing for multiple conditions: illustration of a converging search with residual error. **a** plots mix error, **b** plots fader gain vs time.

listening level. The second scenario is a typical home-listening scenario featuring a kitchen radio at low listening level, and the third is a set of KRK-6400 headphones at moderate listening level. The fourth is a typical concert scenario in a large auditorium with high listening level. In each of the four scenarios the target was equal loudness across tracks, but in each case the target mix loudness was set separately as shown in Table 1. The mixes were solved using the expanded matrices (Eq. (38)), and the total error across all mixes is plotted in Fig. 3**a**. This shows that at the third iteration the error reaches its minimum value, but that this minimum is well above our tolerance, i.e., it is impossible to provide the target loudness balance to all mixes at the same time. The fader gain values for each track are plotted in Fig 3**b**. In this example we normalized the track gain values to have a maximum value of 0 dBFS to remove redundancy caused by the inclusion of a master gain parameter for each mix (identical mixes can be obtained by re-attributing master gain to all tracks and vice versa). The master gain values output by the optimization algorithm were used to determine the reproduction system calibration, which incorporates the presence of a volume control. These are shown in Table 1 and are stated as the ratio of dBSPL to dBFS for a frequency of 1 kHz, i.e., for the studio the loudspeaker volume is set so that a 1 kHz tone with an peak signal level of 0 dBFS will give an acoustic signal with a peak sound level of 104 dBSPL.

4.2 Summary

The orchestral conductor is free to walk the hall during rehearsal to sample the different listening scenarios of audience positions in the hall and perhaps take mental note of the possible best-fit parameters with relation to his podium experience. This best-fit problem seems reasonable since (often by design) the acoustic conditions do not vary wildly with location in the hall. In contrast, for the practicing mixing engineer the problem of producing a best-fit mix across multiple conditions is acute. Firstly because there is no simple means to store and compare his perception of mix features from one scenario to another, and secondly because the possible scenarios are unknown and, even if they were known, they would be practically impossible to sample in person. The algorithm presented here provides a means to find the true best fit and offers advantages over the current human process since it can simulate a listener able to listen to multiple scenarios simultaneously. However, we have also illustrated that the best-fit solution does not necessarily represent a *good* fit if the competing scenarios feature large differences in their acoustic conditions. The only solution in this case would be to produce custom mixes for each reproduction system (e.g., [20]).

5 DISCUSSION

In this article we have treated mixing as a numerical optimization problem. Using an auditory model, we have demonstrated how numerical optimization can be used to pose and solve a mix problem. We have highlighted the interplay between artistic objectives, perceptual constraints, and engineering methods. Taking loudness as example, we have shown that the nonlinearity in the perceptual model leads to complex behavior, but that it can be overcome by careful choice of optimization strategies and parameters. We have illustrated the problem of best-fit, and through our case studies we have provided some insight into the human process of mixing. While the optimization-theory approach offers several advantages over the human process, much work remains before the theory can be fully realized. In particular, the approach places great emphasis on the auditory model and, hence, is likely to identify weak points in auditory theory that had not previously arisen in less artistic contexts.

6 ACKNOWLEDGMENTS

The authors would like to thank the Engineering and Physical Sciences Research Council (EPSRC) UK for funding this research.

7 REFERENCES

[1] M. A. Wolfe, *Numerical Methods for Unconstrained Optimization (Van Nostrand Reinhold Company Ltd. 1978)*, pages 53–73.

[2] P. E. Gill and W. Murray, *Numerical Methods for Constrained Optimization (Academic Press Inc. (London) Ltd. 1974)*, pages 21–28.

[3] E. Perez-Gonzales and J. D. Reiss, "Automatic Mixing: Live Downmixing Stereo Panner," *Proc. DAFX 10th Int. Conf.* (2007).

[4] E. Perez-Gonzales and J. D. Reiss, "Automatic Equalization of Multichannel Audio Using Cross-Adaptive Methods," *presented at the 127th Convention of the Audio Engineering Society* (October 2009), convention paper 7830.

[5] E. Perez-Gonzales and J. D. Reiss "Automatic Gain and Fader Control for Live Mixing," *IEEE Workshop App. Sig. Proc. Audio and Acoust.* (October 2009).

[6] D. Barchiesi and J. D. Reiss, "Reverse Engineering of a Mix," *J. Audio Eng. Soc.*, vol. 58, pp. 563–576 (2010 July/Aug.).

[7] M. J. Terrell and J. D. Reiss, "Automatic Monitor Mixing for Live Musical Performance," *J. Audio Eng. Soc.*, vol. 57, pp. 927–936 (2009 Nov.).

[8] M. J. Terrell and M. Sandler, "An Offline, Automatic Mixing Method for Live Music, Incorporating Multiple Sources, Loudspeakers, and Room Effects," *Computer Music J.*, vol. 36, no. 2, pp. 37–54 (Summer 2012).

[9] M. J. Terrell, J. D. Reiss, and M. Sandler, "Automatic Noise Gate Settings for Drum Recordings Containing Bleed from Secondary Sources," *EURASIP* (2010).

[10] D. Ward, J. D. Reiss, and C. Athwal, "Multitrack Mixing Using a Model of Loudness and Partial Loudness," *presented at the 133rd Convention of the Audio Engineering Society* (2012 October), convention paper 8693.

[11] M. J. Terrell, A. J. R. Simpson, and M. B. Sandler, "A Perceptual Audio Mixing Device," *presented at the 134th Convention of the Audio Engineering Society* (2013 May), convention paper 8840.

[12] B. C. J. Moore, B. R. Glasberg, and T. Baer, "A Model for the Prediction of Thresholds, Loudness, and Partial Loudness," *J. Audio Eng. Soc.*, vol. 45, pp. 224–240 (1997 Apr.).

[13] B. R Glasberg and B. C. J. Moore, "A Model of Loudness Applicable to Time-Varying Sounds," *J. Audio Eng. Soc.*, vol. 50, pp. 331–342 (2002 May).

[14] B. R. Glasberg and B. C. J. Moore, "Development and Evaluation of a Model for Predicting the Audibility of Time-Varying Sounds in the Presence of Background Sounds," *J. Audio Eng. Soc.*, vol. 53, pp. 906–918 (2005 Oct.).

[15] A. J. R. Simpson, M. J. Terrell, and J. D. Reiss, "A Practical Step-by-Step Guide to the Time-Varying Loudness Model of Moore, Glasberg, and Baer (1997; 2002)," *presented at the 134th Convention of the Audio Engineering Society* (2013 May), convention paper 8873.

[16] P. I. M Johannesma, "The Pre-Response Stimulus Ensemble of Neurons in the Cochlear Nucleus," *Symp. Hear. Theory*, pp. 58–69 (1972).

[17] E. Zwicker, "Procedure for Calculating Loudness of Temporally Variable Sounds," *J. Acoust. Soc. Am.*, vol. 62, no. 3, pp. 675–682 (1977).

[18] S. Kuwano and S. Namba, "Continuous Judgment of Level-Fluctuating Sounds and the Relationship between Overall Loudness and Instantaneous Loudness," *Psychological Research*, vol. 47, pp. 27–37 (1985).

[19] G. H. Golub and C. F. Van Loan, *Matrix Computations* (The Johns Hopkins University Press, Baltimore, MD, 1996).

[20] M. J. Terrell, A. J. R. Simpson, and M. Sandler, "Sounds Not Signals: A Perceptual Audio Format," *presented at the 132nd Convention of the Audio Engineering Society* (2012 Apr.), e-Brief 52.

THE AUTHORS


Michael Terrell


Andrew Simpson


Mark Sandler

Michael Terrell was born in 1980. He received M.Eng. and Ph.D. degrees in aeronautical engineering from the University of Bristol, Bristol, UK in 2002 and 2006 respectively; and a Ph.D. degree in music technology from Queen Mary University of London, London, UK in 2012, specializing in the fields of music perception and music production. Since completing his recent research degree, Michael has worked as a research assistant and lecturer in audio and music production. He is currently working to commercialize his research via his start-up company Mix Elephant (www.mixelephant.com).

●

Andrew Simpson was born in 1978. He has an M.Sc. in advanced music production from Glamorgan University and is currently completing a Ph.D. in psychoacoustics at Queen Mary University of London, specializing in central auditory processing.

●

Mark Sandler was born in 1955. He received the B.Sc. and Ph.D. degrees from the University of Essex, Essex, UK, in 1978 and 1984, respectively. He is a Professor of Signal Processing at Queen Mary University of London, London, UK, and Head of the School of Electronic Engineering and Computer Science, where he founded the Centre for Digital Music. He has published over 400 papers in journals and conferences. Mark is a Fellow of the Institute of Engineering and Technology (FIET), a Fellow of the Audio Engineering Society (FAES), a Fellow of the British Computer Society (FBCS), and a Fellow of the Institution of Electronic and Electrical Engineers (FIEEE).