Reverse Engineering of a Mix*

DANIELE BARCHIESI AND JOSHUA REISS

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

It is shown how to reverse engineer the parameters that, starting from a multitrack recording, can produce a given mix. Linear effects and dynamic processors, which comprise all the effects commonly used in the mixing and mastering stages, are considered. Two different techniques based on least-squares optimization are described. Starting from a multitrack recording and a target mix, which is obtained by applying effects to each of its channels, impulse responses and gain envelopes are calculated, which can be used to estimate gains, delays, filters, panning settings, and combinations of these processors; or to estimate time-varying gain envelopes produced by dynamic effects, such as compressors and expanders. Theoretical and experimental results show that, given some assumptions about the nature of the processing originally applied, the proposed techniques are able to precisely and efficiently retrieve the mixing parameters.

0 INTRODUCTION

0.1 Problem Definition and Applications

In a music production the released product is often far more than the recording of a music performance. It is the result of the work of musicians and of mixing and mastering engineers. Although mixing and mastering often overlap, the basic difference between them is that the former acts on single channels or instruments and is performed before the latter, which acts on groups of channels or instruments or on whole mixes. Both mixing and mastering involve the use of many audio effects in order to process the recorded signals for technical or artistic purposes [1], [2].

A basic classification of audio effects distinguishes between linear and nonlinear signal processors. Gains, delays, stereo panners, and filters (which are, indeed, a combination of gains and delays) belong to the former category, whereas dynamic effects, such as compressors or expanders, and other particular effects, such as distortion modules, belong to the latter.

In this paper we will consider linear effects and dynamic processors, which comprise all the effects commonly used in the mixing and mastering stages, and we will show how to reverse engineer the parameters that, starting from a multitrack recording, can produce a given mix. In particular we will describe two different techniques, which can be used to estimate gains, delays, filters, panning settings (and combinations of these processors) or time-varying gain envelopes produced by dynamic effects.

Nowadays digital audio and enhanced recording and signal processing techniques allow to produce recordings with a higher fidelity than in the past, and this, of course, reflects in a consumer's improved listening experience. For this reason many old analog recordings are being converted into digital formats and remastered, where the remastering usually consists in improving their quality by means of signal processing techniques (for instance, denoising, click and hum removal, and so on) and performing again the mixing and/or mastering processes.

At this stage it would be very helpful to have information about what effects have been applied originally and how, in order to produce a remastered edition that sounds *better*, and not *different* from the original record. In other words, it would be useful to have a recording of the engineer's performance. Unfortunately this is often not possible because, before people started using computers and digital effects, there was not an effective way to store information about the mixing and mastering parameters. The techniques presented in this paper may contribute to fill this gap, allowing—to some extent—the reverse engineering of a mix.

Besides remastering, an interesting new trend in the music industry is the release of records along with their raw multitrack recordings. This allows users to create their own mixes and to perform custom processing on the audio material (a couple of examples, among the most famous artists, can be found by visiting the Web sites of Nine Inch Nails and Radiohead.[1] An MPEG format called spatial audio object coding (SAOC) [3] is currently under standardization and will rule the storage and transmission of multiple objects (instruments and/or speech tracks) that can be spatially placed and remixed during reproduction. In this scenario reverse engineering of the mixed version is a valuable learning tool that can show how professional engineers have mixed and mastered the record.

0.2 Background

There exist very few scientific publications regarding the reverse engineering of a mix. A quite wide literature

*Manuscript received 2009 October 26; revised 2010 May 10.

[1] http://remix.nin.com and http://www.radioheadremix.com/

deals with related topics, such as the estimation of effects parameters [4] or the automatic adjustment of the parameters of an effect based on a target [5], [6]. There are several commercial products [7], [8] that implement this second principle, suggesting an equalization curve that matches a target frequency response, which is a problem that has been extensively addressed in the fields of adaptive filtering and systems equalization [9]. Although these tools can be useful in some situations, they do not really tackle the problem of finding parameters applied to a mix, because they act on single tracks or whole sessions and, therefore, are not able to distinguish different parameters applied to different channels or instruments.

The adress algorithm [10], although mainly designed for source separation, is able to retrieve the panning parameters of a mix and is a good example of parameter estimation acting on a multitrack recording. However, to the best of the authors' knowledge the only scientific publication dealing explicitly with the reverse engineering problem is a paper by Kolasinski [11]. The goal of his algorithm is to find the gains originally applied to a multitrack recording using a genetic optimization. This is a very powerful technique, which combines a random search with rules inspired by evolutionary processes and has been employed successfully to solve difficult optimization problems. However, this method requires a huge computational cost and produces less accurate results as the number of tracks increases.

In Section 1 we will describe an alternative approach, which allows us to retrieve gain parameters exactly, even for a large number of tracks, and which requires much less computational time. This technique will then be extended to the estimation of any linear time-invariant effect applied to each channel of the multitrack recording. Section 2 will describe the estimation of dynamic effects, and Section 3 will deal with the evaluation of the proposed techniques. Finally we will draw our conclusions in Section 4, along with plans for further research. The appendixes will show some theoretical results of the estimation of linear time-invariant systems.

1 LINEAR TIME-INVARIANT SYSTEM ESTIMATION

1.1 Least-Squares Solution

The basic principle behind all the techniques described in this paper is to represent the multitrack recording and the final mix (which, from now on, will be referred as the target mix) as vectors in a high-dimensional Hilbert space. This will allow us to view and solve the estimation of mixing parameters using geometric methods.

In the simplest mixing scenario we can assume that the target $t(n)$ is generated applying different gains α_k to the various channels $x_k(n)$ of the multitrack recording,

$$t(n) = \sum_{k=1}^{K} \alpha_k x_k(n). \tag{1}$$

This linear combination can be written in matrix notation as

$$t = \boldsymbol{X}\alpha$$

where $\boldsymbol{X}$ is the matrix whose columns contain the tracks x_k. If we consider the column space of $\boldsymbol{X}$, which is the subspace generated by any linear combination of the input tracks, then we can project the vector representing the target mix into that space and find a set of optimal coefficients $\hat{\alpha}_k$ that minimize the Euclidean distance $\hat{\alpha} = \min \|t - \boldsymbol{X}\alpha\|$ between target and estimated mix [12]. This $\hat{\alpha}$ can be done using the least-squares formula

$$\hat{\alpha} = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}t. \tag{2}$$

It follows from Eq. (1) that the target mix belongs to the column space of $\boldsymbol{X}$. Therefore the least-squares solution is able to retrieve exactly the original gains as long as the tracks x_k are linearly independent, which is an assumption that will be discussed in Section 1.2. This technique is much less computationally expensive than a heuristic optimization procedure such as the genetic algorithm. Moreover it produces exact results even with a large number of input tracks.

A more complex mixing model can be designed if we allow each input track to be processed by a linear time-invariant (LTI) system. One of the basic principles of digital signal processing states that every LTI system is uniquely defined by its impulse response and that its output y can be calculated as the convolution of the input signal x with the impulse response h,

$$y(n) = (x * h)(n).$$

If we express this relation in matrix notation, considering a Pth-order LTI system, then we obtain the following:

$$y = \boldsymbol{X}_P h$$

where $\boldsymbol{X}_P$ is the matrix whose columns contain shifted versions of the input signal up to the time index P.

Since we are considering a multitrack recording, the new mixing model can be written as

$$t = \boldsymbol{X}_{K,P}\alpha \tag{3}$$

where the matrix $\boldsymbol{X}_{K,P}$ contains shifted versions of all the input tracks and the vector α contains the coefficients of different Pth-order LTI systems applied to each channel. Once again, the optimization of α can be solved projecting the target mix into the space generated by any linear combination of the shifted input tracks using the least-squares formula, Eq. (2).

This technique can be used to estimate the impulse response produced by all the audio effects that fall into the category of LTI systems, which includes gains, delays, stereo panners, and equalization filters. However, we cannot make prior assumptions on the nature of the processing originally applied and, in particular, about the length of the impulse responses that generated the target mix. If the multitrack recording was mixed using FIR systems, then it is possible to increase the estimation order P until the target belongs to the column space of $\boldsymbol{X}_{K,P}$, obtaining a theoretically exact solution. On the other hand, if any IIR filter has been employed, then we would need to estimate an infinite number of coefficients in order to

obtain an exact solution. Appendix 1 describes the relation between the original impulse responses and an upper bound of the estimation error, providing a sufficient condition for the success of the least-squares method.

It is practical to implement the least-squares problem described by Eq. (3) on a frame-by-frame basis, since this leads to a smaller computational load. Moreover, by restricting the estimation to small windows, the assumption on the time invariance of the filters used during the mixing stage is no longer a strict requirement, since it is needed only during the time interval defined by each window. As a practical example, if we consider an eight-track recording and an estimation order $P = 500$, the minimum window size that results in a determined system of equations is $8 \times 500 = 4000$ samples, which corresponds to about 90 ms at the standard CD sample rate of 44.1 kHz.

More details on this topic will be discussed in Section 2.1, where we will treat the gain envelope produced by dynamic effects as a time-varying system to be estimated with a frame-by-frame technique.

In theory the least-squares method could be employed to estimate the impulse response of convolutional reverberators, since these effects belong to the category of FIR linear processors. However, the duration of a realistic impulse response can easily reach several seconds, which would lead to an optimization problem that is too large for the processing power of the current computers.

1.2 Linear Independence of Input Tracks and Undetermined Systems

The linear mixing model described in Section 1.1 assumes that the input tracks contained in the multitrack recording (and their delayed versions employed in the estimation of equalization filters) are linearly independent. This means that none of those signals can be expressed as a linear combination of the others or, more intuitively, that the tracks do not contain submixes of the various channels (or of their filtered versions). Even when two instruments are harmonically or rhythmically correlated, as is the case for backing vocal tracks, the resulting signals will be linearly independent. If some of the tracks contain leakage from different sources, which often happens in a multichannel recording of a drum kit, the tracks will still be linearly independent, as discussed in Appendix 3.

Among the common practices employed during the mixing process, we can mention the use of auxiliary buses and the additional processing of the master channels [1]. In the former case one or several tracks are routed to an auxiliary bus, where they are transformed by audio effects and then added to the final mix, while in the latter additional processing is applied to the left and right master channels after the tracks have been mixed. In those cases the estimation of mixing parameters is not unique.

This fact is immediately obvious if we consider a simple example, which can be extended to more complex processing chains. Suppose that one of the tracks is routed to an auxiliary bus and multiplied by a gain g_{aux} before being added to the final mix that contains the same signal amplified by the gain g_{chn} applied in its channel strip. Then there is an infinite choice of settings that will produce the target mix, that is, the set of parameters for which $g_{\text{chn}} + g_{\text{aux}}$ is constant. Whenever we are given the auxiliary buses or the master channels as part of the multitrack recording, we can still use the proposed algorithm, but the least-squares estimation will be applied to a set of tracks that are no longer linearly independent. The resulting system of equations will be undetermined and, consistently, there will be an infinite set of valid parameters that can produce the target mix. Still, the least-squares approach leads to a desirable solution in that it will choose the parameters with the smaller l_2 norm [13], avoiding unrealistic gains or filter coefficients.

1.3 From Impulse Responses to Mixing Parameters

Fig. 1 is the flowchart of a basic stereo mixing console. The input tracks $x_1, \ldots, x_K$ are processed through the gains $g_1, \ldots, g_K$ and the delays $d_1, \ldots, d_K$. The resulting signals are then equalized and placed in a particular position of the stereo field through the panning gains p_{L1}, p_{R1}, $\ldots$, $p_{\text{L}K}$, $p_{\text{R}K}$. The composition of all the processors contained in the dashed boxes in the image can be treated as a single linear time-invariant system and estimated independently for left and right channels with the technique described in Section 1.1. The resulting impulse responses can then be applied to each track in order to reproduce the left and right channels of the target mix.

However, if there is the need to distinguish between equalization parameters, delays, and gains, it is possible to separate from the estimated impulse response the contribution of the equalization filter, including some assumptions on the filter itself. Regarding the delays, we can assume that the equalization introduces the minimum possible delay, and therefore discard all the initial zero coefficients of the

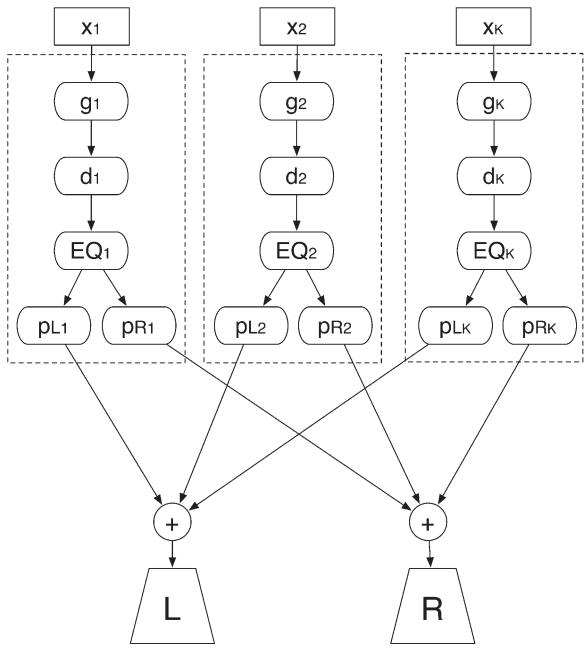

Fig. 1. Flowchart of a basic mixing console.

estimated impulse. Gain and equalization can be distinguished assuming that the filter does not affect the norm of the signal, and thus the gain α_k of a given track will be

$$\alpha_k = \frac{\|y_k\|}{\|x_k\|} \tag{4}$$

where y represents the output of the system and the norm can be the Euclidean norm $\|x\|_2$ or the infinity norm $\|x\|_\infty = \max_n |x(n)|$ if we assume that the equalization does not affect the peak value of the input signal.

As can be noted in Fig. 1, every track in the left or right channel is processed by two different gains. The first, identified by the symbol g_k, is used to balance the contribution of a particular instrument in the mix, whereas the second, p_{Lk} or p_{Rk}, is used for the panning. Consider now the parameter estimation of an arbitrary track and omit the indexes for clarity of notation. The gains α_L and α_R derived from Eq. (4) are actually the product of g with the panning gains p_L or p_R. It is possible to separate these two components, assuming a particular panning law.

One of the simplest and most used laws is the equal power panning law, which constrains the panning gains to follow the relations

$$p_L^2 + p_R^2 = 1$$
$$p_L = \cos(\theta), \qquad p_R = \sin(\theta)$$

where $\theta \in [0, \pi/2]$ is the angle in the stereo field. Therefore for a given track we can determine the first gain g by computing the following:

$$\begin{aligned}\sqrt{\alpha_L^2 + \alpha_R^2} &= \sqrt{g^2 p_L^2 + g^2 p_R^2} \\ &= \sqrt{g^2 (p_L^2 + p_R^2)} \\ &= g.\end{aligned}$$

We can then divide the gains α_L and α_R by this value and retrieve the panning angle θ by inverting the panning law,

$$\theta = \arccos(p_L) = \arcsin(p_R), \qquad \theta \in [0, \pi/2].$$

Clearly the mixing model described in this section is a basic one, and it is possible to consider more complex processing chains including, for example, auxiliary buses or additional effects in the master channels. In this case the algorithm will return one valid solution for each channel, as discussed in Section 1.2. Whether to apply the solution directly or divide the global impulse responses into smaller subsystems, which reflect a particular mixing model, is a task that is outside the scope of this work.

2 DYNAMIC EFFECTS ESTIMATION

Dynamic effects form a category of nonlinear signal processors whose objective is to modify the dynamic range of the input signal. Compressors, limiters, expanders, and noise gates are the most common effects that belong to this category and are widely used for technical or artistic reasons [1]. The shared aspect of all dynamic effects is that they apply a time-varying gain to the input signal based on a measurement of the signal level.

Fig. 2 shows a basic model of a dynamic effect. The input signal $x(n)$ is fed into a level measurement module whose output goes to a gain computer. Based on the type and parameters of the effect, the gain computer outputs an envelope function. This signal is then filtered to produce the time-varying gain $e(n)$, which is multiplied by the input signal to produce the output $y(n)$.

Although dynamic effects do not belong to the category of LTI processors, it is possible to tackle the estimation of time-varying gain envelopes $e_k(n)$ for each input track, using the technique described in Section 1.1 on a frame-by-frame basis.

2.1 Frame-Based Polynomial Gain Estimation

One first attempt at the estimation of gain envelopes was to perform the least-squares optimization of the gain parameters described by Eq. (2) on small windows, assuming that the gains were constant within those regions. Unfortunately this approach does not work because the error introduced where the actual envelopes are not constant leads to a noisy and unreliable estimation.

For this reason a new model has been designed where the envelopes are allowed to follow polynomial trajectories within each window, which is described by

$$t(n) = \sum_{k=1}^{K} \left[\sum_{p=0}^{P} \alpha_{k,p} l(n)^p \right] x_k(n). \tag{5}$$

Here the target mix t in each small window is expressed by the sum of the input tracks x_k multiplied by a polynomial envelope of order P, which is the function contained in the square brackets. $l(n)$ represents a linear function that goes from 0 to 1 during the time interval defined by each window.

Once again, we can express this linear model in matrix notation, obtaining a mixing model that is similar to the one defined by Eq. (3), except that now the matrix $\boldsymbol{X}_{K,P}$ does not contains shifted version of the input tracks but the multiplication between the input channels $x_k(n)$ and the polynomial functions $l(n)^p$. At this point we can estimate

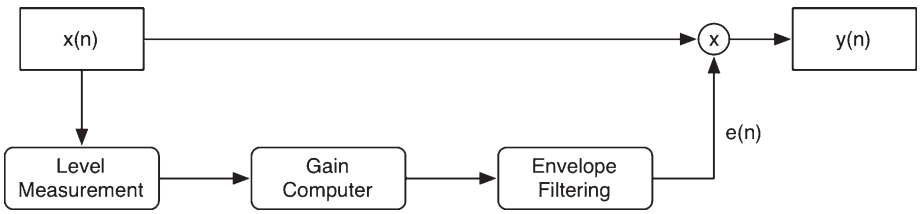

Fig. 2. Model of a dynamic effect.

the optimal polynomial coefficients $\hat{\alpha}_{k,p}$ using the least-squares formula, Eq. (2), and define different gain envelopes e_k for each track x_k,

$$e_k(n) = \sum_{p=0}^{P} \hat{\alpha}_{k,p} l(n)^p.$$

2.2 Polynomial Estimation and Envelope Smoothness

The two critical parameters that must be controlled for the envelope estimation are the polynomial order and the length of the window used in the frame-based algorithm. Adjusting these two variables involves a tradeoff between model complexity and number of variables to be estimated.

On one hand we would like to choose short windows because the envelopes are in general low-frequency functions, which are likely to be correctly approximated by polynomial functions in small regions. On the other hand we would like to choose an order P that is high enough to be able to describe complex trajectories. Unfortunately these two objectives are contradictory in that, as we increase the estimation order P, we also need to increase the window length to ensure that the least-squares algorithm solves an (over)determined system of equations. As a result the estimation will be successful if the target envelopes are smooth enough to be correctly described by polynomial functions of a given order within each window.

There is not a simple way of describing the smoothness of the envelope produced by a dynamic processor, because this will depend on the particular implementation of the effect. As will be shown in Section 3.2, we empirically found that a polynomial order between 2 and 6, with a window length chosen to be four times the total number of estimation variables, provides good results, considering different dynamic effect models.

Table 1. Mixing parameters for four-track test recording.

Track	Gain (dB)	Delay (samples)	Equalization
Drums	−6	0	Free-hand FIR 3
Guitar	0	30	Low-pass IIR 4
Bass	−6	50	None
Percussion	0	0	None

3 EVALUATION

3.1 Evaluation of LTI System Estimation

In order to evaluate the LTI algorithm, we first mixed a four-track test recording using different LTI processors for each channel. We then compared their impulse responses with the ones estimated using our method.

Table 1 shows the mixing parameters applied to each track. The recording is sampled at the standard CD quality (44.1 kHz/16 bit) and is 30 seconds long. Figs. 3 and 4

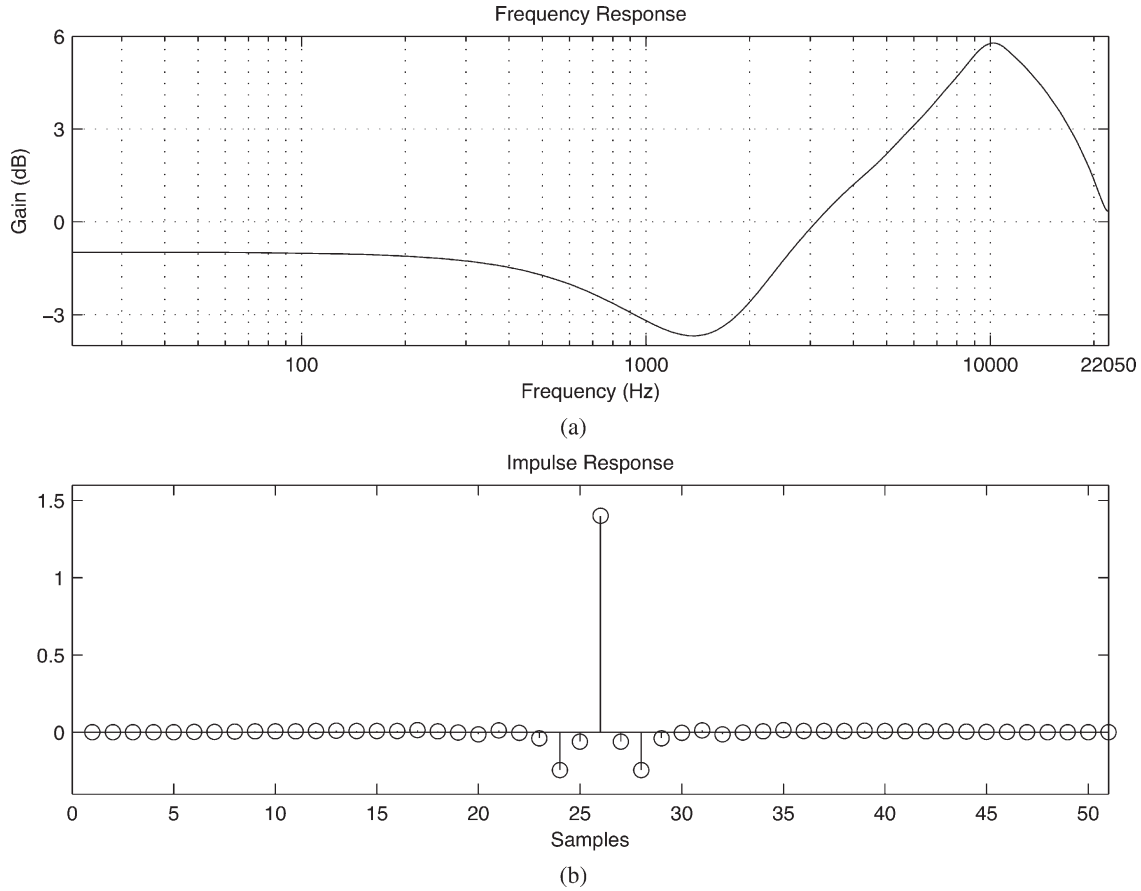

Fig. 3. Drums equalization filter. (a) Frequency response. (b) Impulse response.

show the frequency and impulse response of the filters used in the test. The equalization applied to the drums track is obtained using a 50th-order FIR filter whose frequency response is an interpolation between fixed gains defined at 100, 1000, and 10 000 Hz. The second filter used for the equalization of the guitar track is a second-order low-pass IIR filter with cutoff frequency at 1 kHz.

Fig. 5 depicts the impulse responses retrieved for each track choosing an estimation order $P = 100$. As can be seen, the impulse responses of the tracks that had been equalized are scaled and shifted versions of the original responses, where the scaling depends on the gain applied to the track and the shift depends on the delay. The impulse responses estimated for the channels that had not been equalized are a delta function, scaled and delayed appropriately.

3.2 Compression Envelope Estimation

The estimation of dynamic effects envelopes has been evaluated mixing an eight-track test recording with a

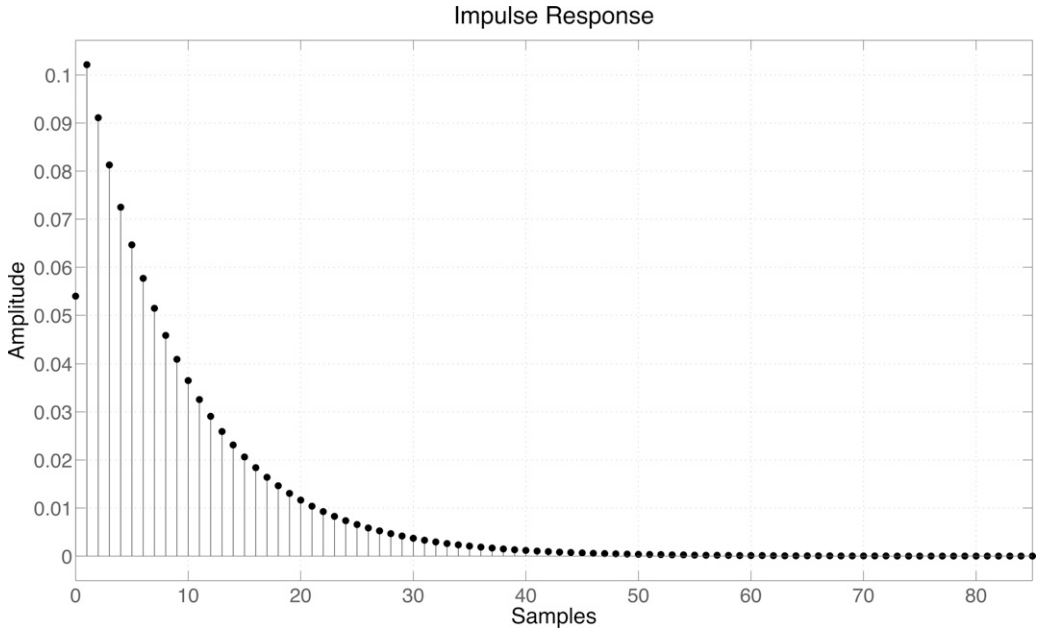

Fig. 4. Impulse response of guitar equalization filter.

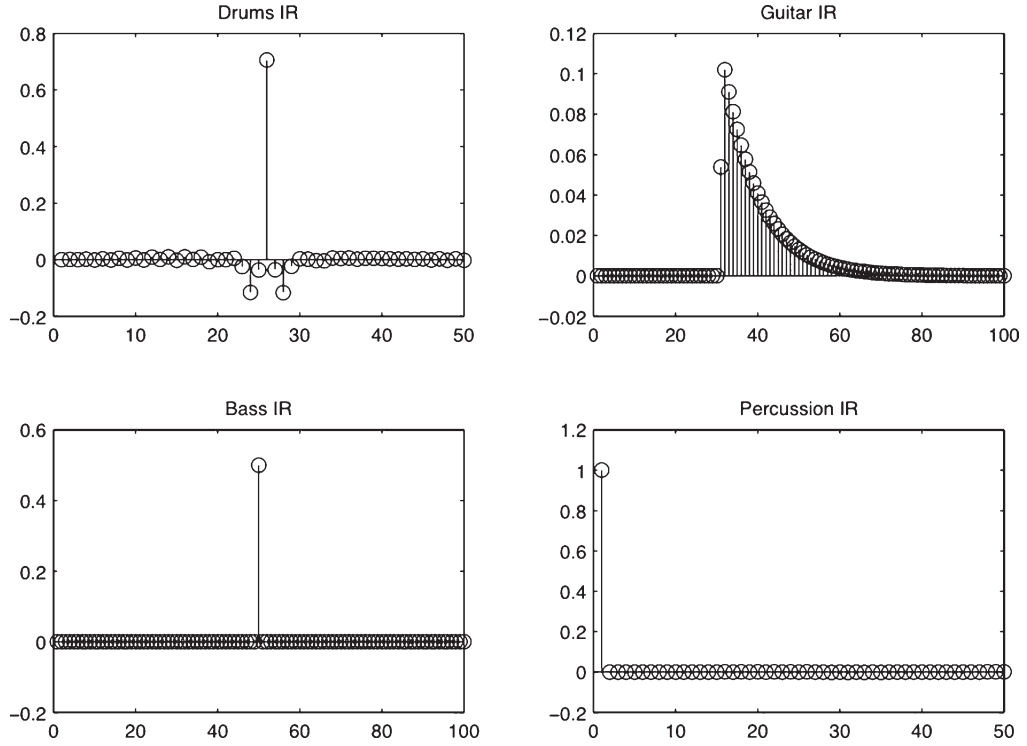

Fig. 5. Multitrack estimation of linear time-invariant systems.

dynamic compressor applied to each channel. The specifics of the recording are the same as for the one used in the previous evaluation.

A dynamic compressor generates a gain envelope based on a measurement of the rms value of the input signal, which is defined as

$$\mathrm{rms}(n) = \sqrt{\sum_{m=-M/2}^{M/2-1} \frac{x^2(n-m)}{M}}. \quad (6)$$

However, in the implementation of real-world effects this measurement is often approximated by filtering the squared input signal with a first-order low-pass IIR filter and taking the square root of the output [14],

$$\mathrm{rms}^2(n) = \alpha x^2(n) + (1-\alpha)\mathrm{rms}^2(n-1) \quad (7)$$

where α is the time constant of the filter. The same first-order IIR low-pass filter is then applied to the time-varying gain produced by the gain computer (see Fig. 2) as a smoothing filter, but this time using two different time constants during the attack and the release portions of the input $x(n)$.

According to the literature [4], [14], [15] there are various ways of choosing the relation between the user-defined time constants of the effect (which can be specified in ms) and the coefficient α. Moreover different implementations may use only one of the two filters placed before and after the gain computer, or may use the time-varying filter with attack and release time constants for both the rms measurement and the gain envelope smoothing.

Depending on all these variables, the resulting compression envelope may be more or less smooth, and therefore its estimation can be more or less successful, as described in Section 2.2. For this reason we decided to test three different compressor models whose four standard parameters (threshold, ratio, attack, and release) were set randomly for each track, choosing within ranges of typical values. All models follow the scheme depicted in Fig. 2, but each one has a different rms level measurement.

- **Model A** The rms is computed using Eqn. (6) on a frame-by-frame basis. The windows are overlapping by 50% of their length, and linear interpolation is used between consecutive frames.
- **Model B** The rms is approximated by Eq. (7). The parameter α is fixed and computed using a time constant of 100 ms.
- **Model C** The rms is computed filtering the square of the input signal with the same time-variant IIR used to process the compression envelope after the gain computer. The filter is followed by a square-root calculation.

Once the tracks had been mixed, we proceeded with the estimation of the compression envelopes choosing a polynomial order $P = 4$ and a window whose length was chosen to be four times the total number of estimation parameters: $4 \times 4 \times 8 = 128$ samples.

Figs. 6, 7, and 8 show the results for the three different models. The black and gray lines represent the true and estimated envelopes, respectively.

As can be seen, the algorithm is able to retrieve the correct envelopes in most of the regions where only the black lines are visible. There are some areas in tracks 6, 7, and 8 where the estimation is wrong, but this is due to the fact that the channels do not contain any signal in those regions, and therefore this does not affect the accuracy of the method. However, Fig. 7 shows some small estimation errors in most tracks. This is because the compression model B produces the least smooth envelopes, and the fourth-order polynomial used in the estimation is not able to follow accurately the gain trajectories in each window. These errors may be reduced by trying other values of polynomial order and window length or processing the estimated envelopes with a smoothing filter.

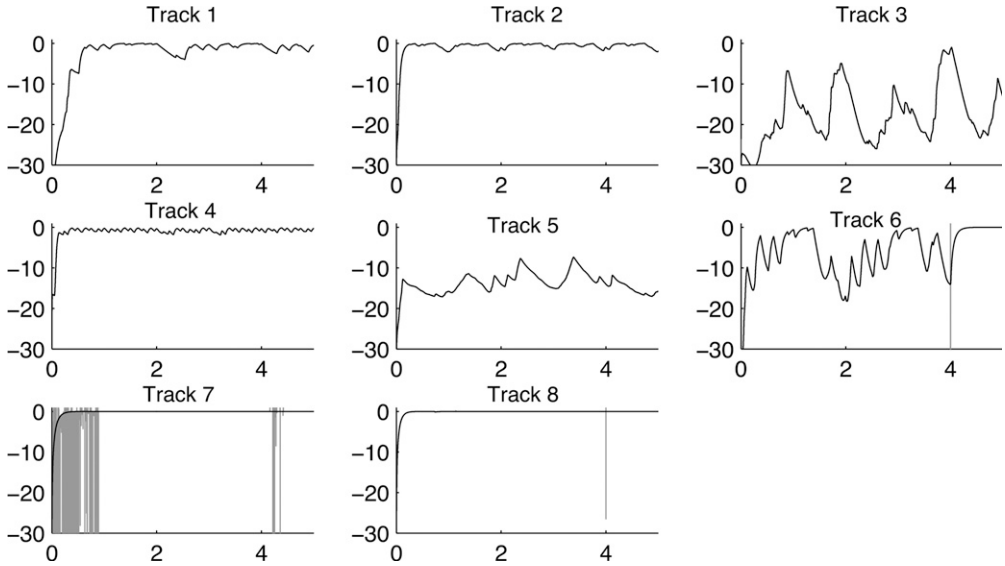

Fig. 6. Estimation of compression envelopes, model A.

3.3 Reverse Engineering Demonstration Software

We have developed a demonstration software that can be used to test our proposed algorithms and which can be downloaded freely from http://www.isophonics.net/content/reverse-engineering-mix.

Fig. 9 shows the main GUI of the program. The main panel in the upper part of the interface is a basic stereo mixer. The user can load up to eight channels of a multitrack recording and create a custom mix adding effects such as delay, equalization filters, compressors, gains, and pan controls. The equalization is obtained using 128th-order FIR filters designed to have a free-hand frequency response similar to the one shown in Fig. 3. The dynamic compressors are implemented using model C, described in Section 3.2. After having set all the mixing parameters it is possible to create the target mix using the button in the lower left panel. Alternatively the target mix can be loaded from an external file choosing the mode option in the same panel.

The lower right portion of the GUI contains the controls for the estimation of mixing parameters. Choosing from the mode option and typing the estimation order, it is possible to test both the LTI systems and the dynamic effects estimation. Once the optimization is completed, the user can view the target and estimated equalization pressing the *Show EQ* button in each channel of the mixer or the target and estimated envelopes pressing the *Show*

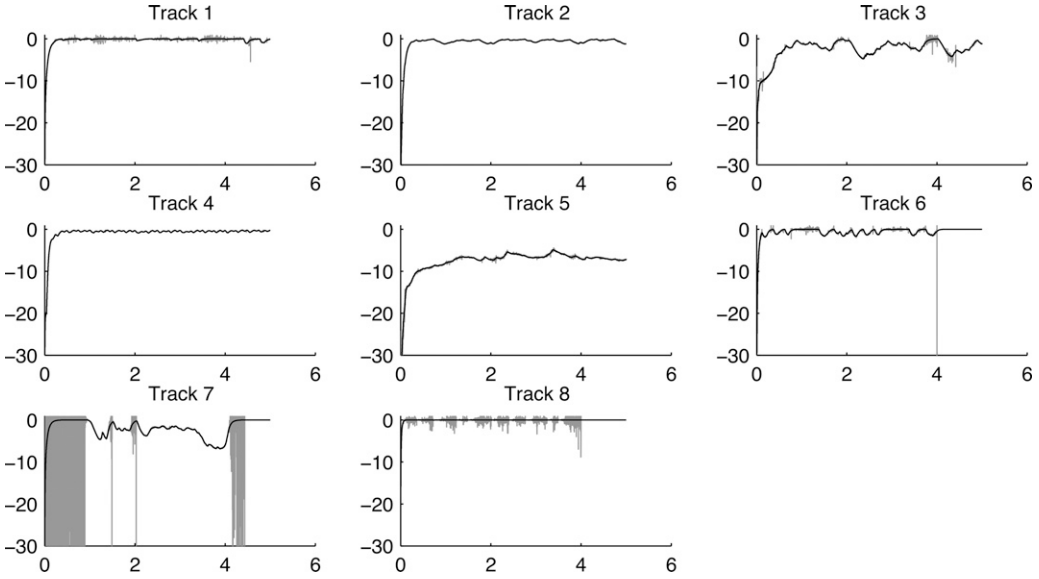

Fig. 7. Estimation of compression envelopes, model B.

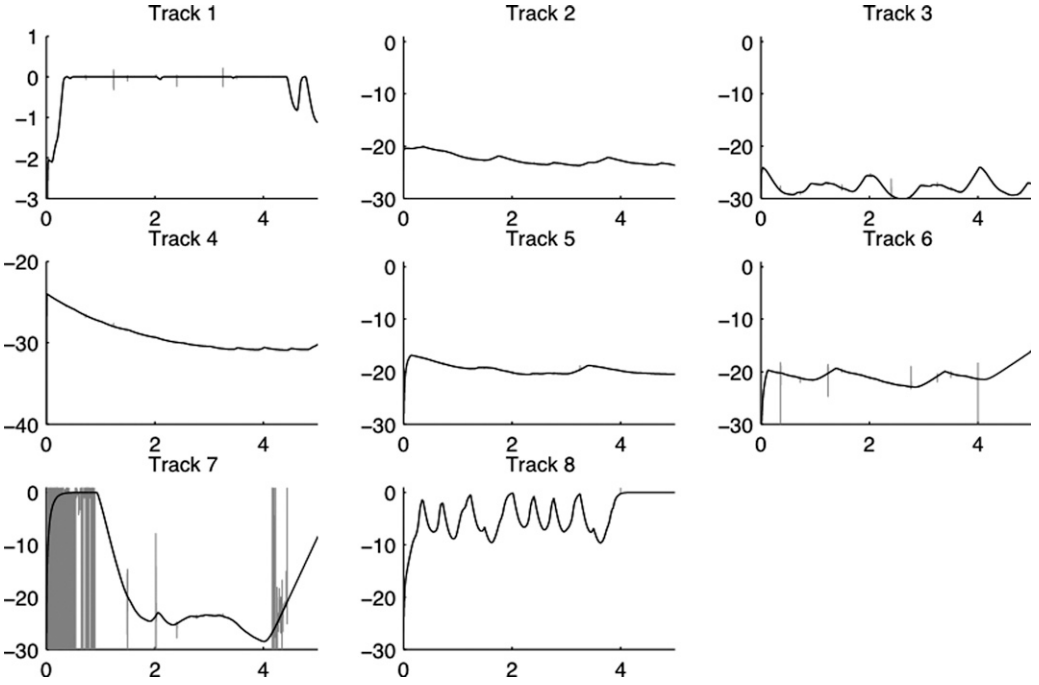

Fig. 8. Estimation of compression envelopes, model C.

ENV buttons. More details on the functionalities of the software can be found on the Web, along with a short video, which shows how to use it to reproduce results analogous to the ones presented in this paper.

3.3.1 Real-World Example

In order to test the proposed estimation algorithms in a real-world situation, we mixed a six-track recording using the Apple Logic Pro software. The signals are sampled at the standard CD quality and are available as part of the downloadable demonstrative application. Fig. 10 depicts a screen shot of the mixer panel. Once the mixed version had been exported, we proceeded with the parameter estimation loading the mix into the reverse engineering demonstration software. We set the estimation order to 512 and ran the LTI systems algorithm. When the algorithm terminates, the mean normalized error for the left and right channels is shown in the GUI,

$$\bar{\varepsilon} = \frac{1}{2}\left(\frac{\|t_L - e_L\|}{\|t_L\|} + \frac{\|t_R - e_R\|}{\|t_R\|}\right)$$

where t_L and t_R are the target mixes in the left and right channel, respectively, and e_L and e_R are the estimated mixes. In our experiment the error resulted in $\bar{\varepsilon} \approx 5.42 \times 10^{-4}$.

Figs. 11 and 12 show the estimated frequency response of the drums and guitar equalizers, which had been processed with the Channel EQ in Logic to produce the target mix. As can be seen, the frequency response of the two filters has been correctly identified. There is an offset in the global gain of the equalizers, which is due to the difference between the gain set in the logic channel strips, the attenuation caused by the panning, and the parameters retrieved by the reverse engineering demonstration software. However, this does not affect the accuracy of the solution since adding the various attenuations results in the same global gain.

A noisy estimation can be observed in correspondence with the very high frequencies of the retrieved equalizer responses. We believe that this is due to the quantization noise introduced when exporting the mix. An intuitive explanation of this fact is that the least-squares algorithm tries to adjust the high-frequency components of the signals in order to match the quantization noise.

4 CONCLUSIONS AND FURTHER RESEARCH

4.1 Current Achievements

In this paper we proposed two algorithms based on a least-squares optimization that can be used for reverse engineering a mix. The evaluation of our techniques shows that, given the raw multitrack recording and the final or target mix, it is possible to estimate the parameters of a wide range of different effects, including linear time-invariant processors (gains, delays, stereo panners, and filters) and dynamic effects.

The theory behind the optimization process is based on the definition of linear mixing models and on the simple principle of projection in a vectorial space. Therefore the

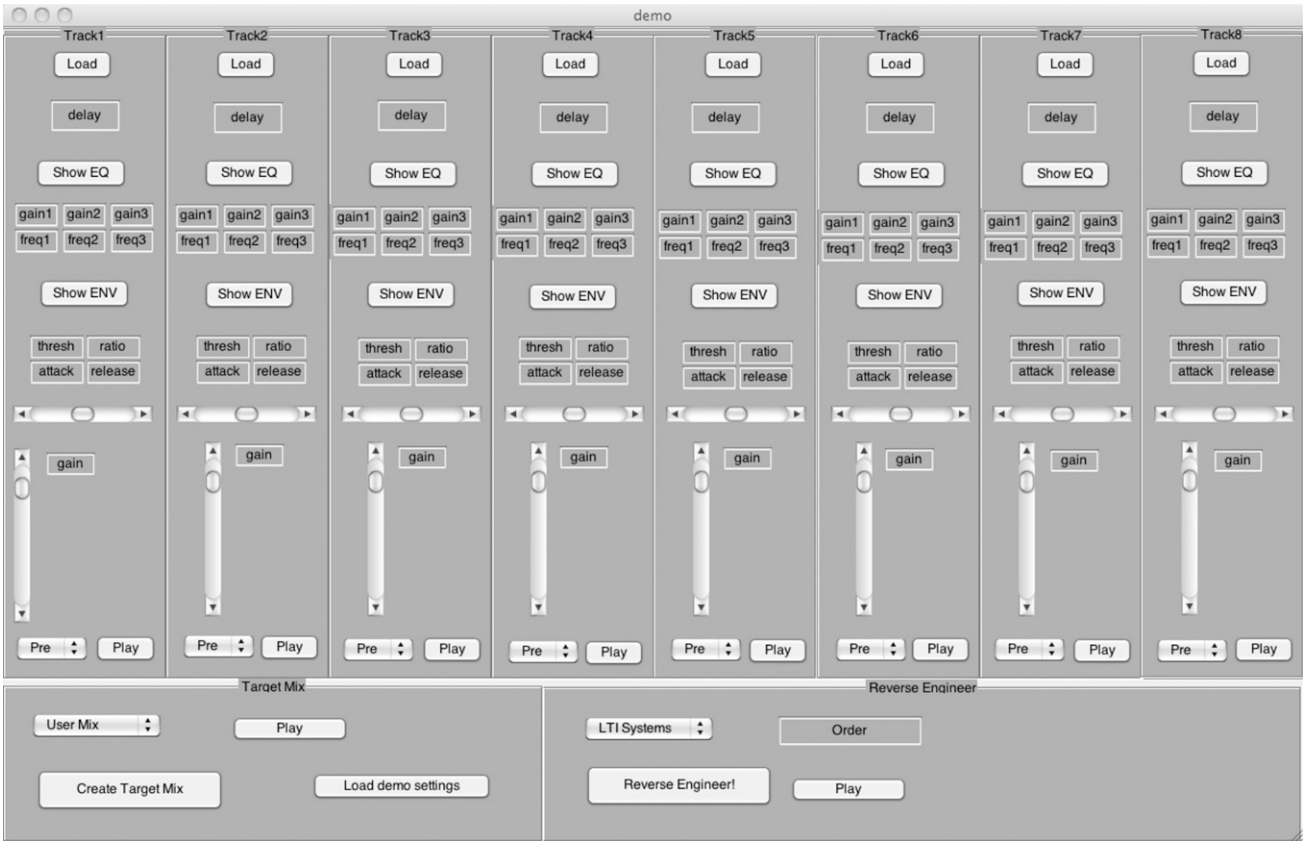

Fig. 9. Main GUI of reverse engineering demonstration software.

estimation requires a very small computational cost if compared with heuristc optimization algorithms. Moreover the retrieved functions are impulse responses and gain envelopes, which are general parameters that do not require any knowledge about the implementation of the original effects.

4.2 Further Research

The proposed system allows one to retrieve the impulse responses of linear effects or the gain envelopes produced by dynamic processors if only one of the two categories of effects has been used in the mix. In order to tackle this problem, one approach is to exploit the fact that the number of parameters of most dynamic effects is very small if compared to the number of variables required for the estimation of the envelopes. (For instance, the typical parameters of a compressor are threshold, ratio, attack, and release.) If we consider a particular compressor model it is possible to define the envelope as a function of the

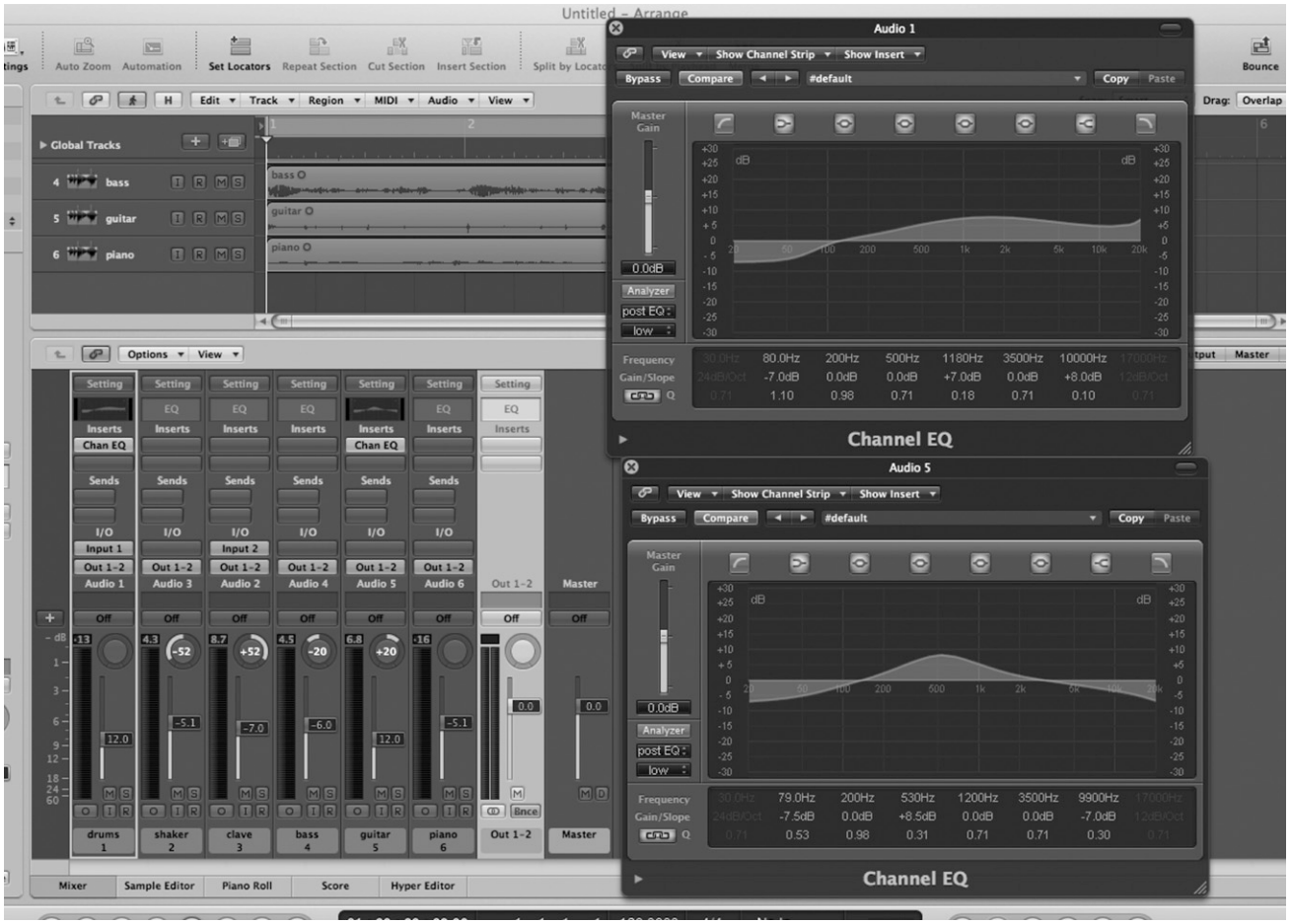

Fig. 10. Screenshot of mixer panel used to generate target mix.

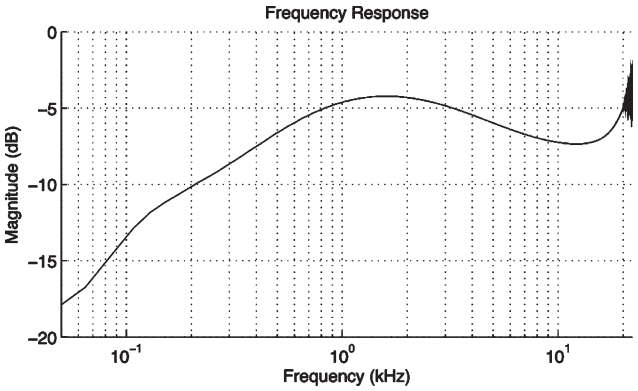

Fig. 11. Estimated frequency response of drums equalizer.

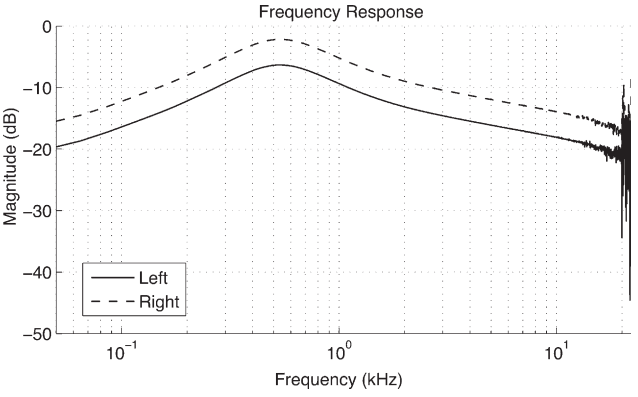

Fig. 12. Estimated frequency response of guitar equalizer (right and left channels).

parameters mentioned and perform a joined optimization of linear effects and compression over large windows of the signal. However, since dynamic effects are nonlinear, this optimization cannot be performed using a simple least-squares approach. Preliminary results show that, even considering a single track and one of the compressor models described in Section 3.2, it is still an open problem how to retrieve the compression parameters.

Another strategy that can be investigated is to perform a time–frequency analysis of the target mix. Since dynamic effects and filters are used to modify the signals in the time and frequency domains, it may be possible to separate their contributions and perform two separate estimations.

Another direction for further research regards the improvement of the proposed algorithms and, in particular, of the LTI systems estimation. As described in Appendix 1, the convergence of the algorithm depends on the target impulse response and on the time delays considered in the optimization. The present technique takes into account the first P coefficients of each FIR filter, which leads to an optimal solution only if the original filters are minimum phase. It may be possible to improve the robustness of the algorithm by finding an optimal set of delays using a matching pursuit type algorithm [16].

Finally one of the main disadvantages of the simple least-squares approach is that it is sensitive to noise. We observed this problem in the LTI estimation described in Section 3.3.1, where the signal was corrupted by very low quantization noise. This may be solved by employing a regularized least-squares method that enforces a constraint on the smoothness of the estimated frequency responses.

REFERENCES

[1] R. Izhaki, *Mixing Audio: Concepts, Practices and Tools*, 1st ed. (Focal Press, Oxford, UK, 2008).

[2] B. Katz, *Mastering Audio: The Art and the Science* (Focal Press, 2002).

[3] J. Breebaart, J. Engdegård, C. Falch, O. Hellmuth, J. Hilpert, A. Hoelzer, J. Koppens, W. Oomen, B. Resch, E. Schuijers, and L. Terentiev, "Spatial Audio Object Coding (SAOC)—The Upcoming MPEG Standard on Parametric Object Based Audio Coding," presented at the 124th Convention of the Audio Engineering Society, (Abstracts) www.aes.org/events/124/124thWrapUp.pdf, (2008 May), convention paper 7377.

[4] U. Simmer, D. Schmidt, and J. Bitzer, "Parameter Estimation of Dynamic Range Compressors: Models, Procedures, and Test Signals, presented at the 120th Convention of the Audio Engineering Society, *J. Audio Eng. Soc.* (Abstracts) vol. 54, p. 736 (2006 July/Aug.), convention paper 6849.

[5] S. Heise, M. Hlathy, and J. Loviscach, "Automatic Adjustment of Off-the-Shelf Reverberation Effects," presented at the 126th Convention of the Audio Engineering Society, (Abstracts) www.aes.org/events/126/126thWrapUp.pdf, (2009 May), convention paper 7758.

[6] D. Reed, "A Perceptual Assistant to Do Sound Equalization," *Proc. 5th Int. Conf.* on *Intelligent User Interfaces* (New Orleans, LA, 2000 Jan.), pp. 212–218.

[7] S. T. Pope and A. Kouznetsov, "Expert Mastering Assistant," Tech. Document. (University of California and FASTLab, Inc., Santa Barbara, CA, 2008 Sept.), p. 11.

[8] TC Electronic Inc., *Assimilator Manual* (2002); http://www.tcelectronic.com/assimilatordownloads.asp.

[9] O. Kirkeby and P. A. Nelson, "Digital Filter Design for Inversion Problems in Sound Reproduction," *J. Audio Eng. Soc.*, vol. 47, pp. 583–595 (1999 July/Aug.).

[10] D. Barry, B. Lawlor, and E. Coyle, "Real-Time Sound Source Separation: Azimuth Discrimination and Resynthesis," presented at the 117th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 53, p. 104 (2005 Jan./Feb.), Convention Paper 6258.

[11] B. Kolasinski, "A Framework for Automatic Mixing Using Timbral Similarity Measures and Genetic Optimization," presented at the 124th Convention of the Audio Engineering Society, (Abstracts) www.aes.org/events/124/124thWrapUp.pdf, (2008 May), convention paper 7496.

[12] D. Barchiesi and J. Reiss, "Automatic Target Mixing Using Least-Squares Optimization of Gain and Equalization Settings," in *Proc. 12th Int. Conf. on Digital Audio Effects (DAFx '09)*, vol. 1 (2009 Sept.), pp. 7–14.

[13] S. Boyd and L. Vandenberghe, *Convex Optimization* (Cambridge University Press, New York, 2004).

[14] U. Zolzer, *DAFx: Digital Audio Effects* (John Wiley & Sons, Chichester, UK, 2002).

[15] G. W. McNally, "Dynamic Range Control of Digital Audio Signals," *J. Audio Eng. Soc.*, vol. 32, pp. 316–327 (1984 May).

[16] P. S. K. Y. C. Pati and R. Rezaiifar, "Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition," in *Conf. Re., 27th Asilomar Conference on Signals, Systems and Computers*, vol. 1 (1993 Nov.), pp. 40–44.

APPENDIX 1
CONVERGENCE OF THE LTI SYSTEMS ESTIMATION

Let $t(n) \in \mathbb{R}^N$ be the target mix and

$$\lambda = \text{span}\{x_k(n-p)\}, \qquad k = 1, \ldots, K, \quad p \in \Lambda$$

be the subspace generated by any linear combination of the input tracks x_k delayed by p samples (where Λ represents an arbitrary set of delays considered during the estimation).

In general the target mix t can be expressed as a linear combination of infinite elements,

$$t(n) = \sum_{k=1}^{K} \sum_{p=-\infty}^{+\infty} \alpha_{k,p} x_k(n-p)$$

$$= \sum_{k=1}^{K} \left[\sum_{p \in \Lambda} \alpha_{k,p} x_k(n-p) + \sum_{p \notin \Lambda} \alpha_{k,p} x_k(n-p) \right].$$

Every vector $v \in \mathbb{R}^N$ can be written as the sum of its projection on the subspace $P_\lambda(v)$ and a component

orthogonal to the subspace $\perp_\lambda(v)$. Therefore the target mix can be written as

$$t = P_\lambda(t) + \perp_\lambda(t).$$

The estimation error J is the squared norm of the difference between target and estimated mix,

$$\begin{aligned} J &= \|t - P_\lambda(t)\|^2 \\ &= \|\perp_\lambda(t)\|^2. \end{aligned} \tag{8}$$

The orthogonal component of the target mix is

$$\begin{aligned} \perp_\lambda(t) &= \perp_\lambda \left\{ \sum_{k=1}^{K} \left[\sum_{p \in \Lambda} \alpha_{k,p} x_k(n-p) + \sum_{p \notin \Lambda} \alpha_{k,p} x_k(n-p) \right] \right\} \\ &= \sum_{k=1}^{K} \left\{ \sum_{p \in \Lambda} \alpha_{k,p} \perp_\lambda [x_k(n-p)] \right. \\ &\quad \left. + \sum_{p \notin \Lambda} \alpha_{k,p} \perp_\lambda [x_k(n-p)] \right\}. \end{aligned}$$

The orthogonal component of vectors belonging to the subspace λ is zero, so the previous equation reduces to

$$\perp_\lambda(t) = \sum_{k=1}^{K} \sum_{p \notin \Lambda} \alpha_{k,p} \perp_\lambda [x_k(n-p)].$$

Substituting into Eq. (8) leads to

$$\begin{aligned} J &= \left\| \sum_{k=1}^{K} \sum_{p \notin \Lambda} \alpha_{k,p} \perp_\lambda [x_k(n-p)] \right\|^2 \\ &\leq \sum_{k=1}^{K} \sum_{p \notin \Lambda} \alpha_{k,p}^2 \| \perp_\lambda [x_k(n-p)] \|^2. \end{aligned} \tag{9}$$

The squared norm in Eq. (9) is bounded by

$$B = \max_{k,p} \| \perp_\lambda [x_k(n-p)] \|^2.$$

Therefore the total error J will be

$$J \leq B \sum_{k=1}^{K} \sum_{p \notin \Lambda} \alpha_{k,p}^2.$$

This result shows that the estimation error is bounded by the energy of the impulse response in the region that is not taken into account during the optimization. For example, if the subspace λ is generated by the set $\{x_k\ (n - p)\}$, where $p = 0, \ldots, P$, the estimation will produce a small error only if the energy of the impulse responses applied in the target mix drops to zero after the Pth sample.

APPENDIX 2
EQUIVALENCE OF LEAST-SQUARES SOLUTION IN THE TIME AND FREQUENCY DOMAINS

The method described so far estimates LTI systems finding the optimal impulse responses in the time domain. However, if our goal is to retrieve the equalization curve that has been used to process a given input channel, considering the distance in the frequency domain is a much more meaningful metric for the optimization algorithm. In fact this is not an issue because the least-squares solution is identical with orthogonal transforms.

Let $\boldsymbol{F}$ be the matrix whose rows contain the Fourier basis. Consider now the mixing model [Eq. (3)] in the Fourier domain,

$$\boldsymbol{F}t = \boldsymbol{F}(\boldsymbol{X}\alpha)$$

where we omitted the subscripts K, P for clarity of notation. The least-squares solution of this model can be written as

$$\begin{aligned} \hat{\alpha} &= [(\boldsymbol{FX})^{\mathrm{H}}(\boldsymbol{FX})]^{-1}(\boldsymbol{FX})^{\mathrm{H}}\boldsymbol{F}t \\ &= (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{F}^{\mathrm{H}}\boldsymbol{FX})^{-1}\boldsymbol{X}\boldsymbol{F}^{\mathrm{H}}\boldsymbol{F}t \end{aligned}$$

where the operator $(\cdot)^{\mathrm{H}}$ indicates the complex conjugate or Hermitian of its argument. Since $\boldsymbol{F}$ is an orthogonal matrix, the last equation reduces to

$$\hat{\alpha} = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}t$$

which is the least-squares solution in the time domain.

APPENDIX 3
LINEAR INDEPENDENCE OF MICROPHONE RECORDINGS IN THE PRESENCE OF INTERFERING SOURCES

Consider the drum kit depicted in Fig. 13, which consists of J sources $\{s_j\}_{j=1}^{J}$ recorded using J microphones, producing the tracks $\{x_k\}_{k=1}^{J}$. Each of the signals captured by the microphones will contain a linear combination of the sources,

$$x_k(n) = \sum_{j=1}^{J} g_{jk} s_j(n - \Delta_{jk})$$

where Δ_{jk} and g_{jk} are the delay and the attenuation due to the distance between the jth source and the kth microphone.

In matrix form, stacking the (appropriately delayed) sources in the columns of matrix $\boldsymbol{S}$, the previous equation can be written as

$$\boldsymbol{X} = \boldsymbol{SG} \tag{10}$$

where $\boldsymbol{X}$ contains the microphone signals in each of its columns and $\boldsymbol{G}$ will be referred to as the mixing matrix. The matrix $\boldsymbol{S}$ contains linearly independent columns for the reason explained in Section 1.2. Therefore its rank will be equal to J^2. In order to prove the linear independence of the recorded tracks x_k, we must show that the matrix $\boldsymbol{X}$ has rank J. For the properties of ranks we have that a sufficient condition for the linear independence of the observed signals x_k is given by the mixing matrix $\boldsymbol{G}$ being full rank.

Let us denote $s_{jk} = s_j(n - \Delta_{jk})$ as the signal produced by the jth source and delayed according to the distance

between s_j and the microphone x_k. Then we can explicitly write Eq. (10) as

$$\begin{bmatrix} | & | & & | & & | & | & & | \\ s_{00} & s_{10} & \cdots & s_{J0} & \cdots & s_{0J} & s_{1J} & \cdots & s_{JJ} \\ | & | & & | & & | & | & & | \end{bmatrix} \begin{bmatrix} g_{00} \\ g_{10} \\ \vdots \\ g_{J0} \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ g_{0J} \\ g_{1J} \\ \vdots \\ g_{JJ} \end{bmatrix} = \begin{bmatrix} | & & | \\ x_1 & \cdots & x_J \\ | & & | \end{bmatrix}.$$

Denoting each column of the matrix $\boldsymbol{G}$ by g_k, we can observe that those vectors are sparse with disjoint support. As a consequence the inner product $\langle g_j, g_k \rangle$ is zero for all $j \neq k$, and the columns of the mixing matrix are mutually orthogonal. This ensures that matrix $\boldsymbol{G}$ is full rank, and that the observations x_k are linearly independent.

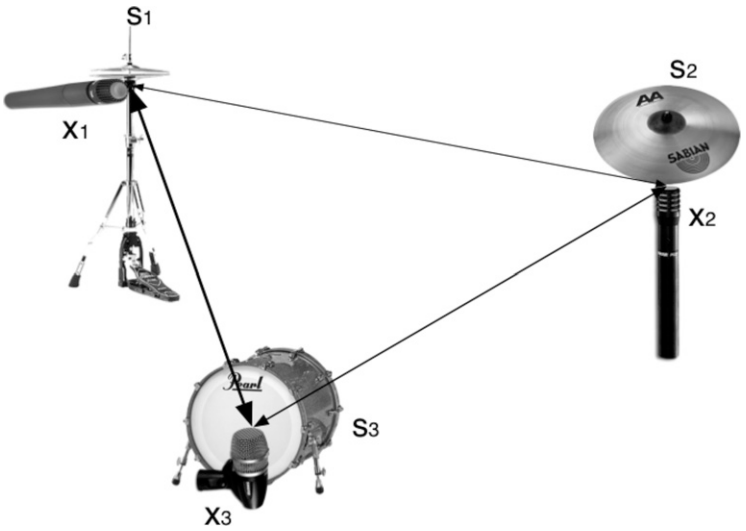

Fig. 13. Typical setup of a multichannel drum recording.

THE AUTHORS

D. Barchiesi

J. Reiss

Daniele Barchiesi was born in Desenzano del Garda, Italy, in 1985. He joined the Centre for Digital Music at Queen Mary University of London, UK, in 2008, where he received an M.Sc. degree in electronic engineering in 2009. He is currently pursuing a Ph.D. degree, working on sparse representations for blind deconvolution problems. His main research interests include signal processing and optimization for audio applications. In his spare time he enjoys singing and playing the piano.

●

Josh Reiss received a Ph.D. degree in physics from the Georgia Institute of Technology, specializing in analysis of nonlinear systems.

He is presently a senior lecturer with the Centre for Digital Music at Queen Mary University of London, UK. He made the transition to audio and musical signal processing through his work on sigma–delta modulators, which led to patents and a nomination for a best paper award from the IEEE. He has investigated music retrieval systems, time scaling and pitch shifting techniques, polyphonic music transcription, loudspeaker design, automatic mixing for live sound, and digital audio effects. His primary focus of research, which ties together many of these topics, is on the use of state-of-the-art signal processing techniques for professional sound engineering.

Dr. Reiss has published over 80 scientific papers and serves on several steering and technical committees. As coordinator of the EASAIER project, he led an international consortium of seven partners working to improve access to sound archives in museums, libraries, and cultural heritage institutions. He is cochair of the AES Technical Committee on High-Resolution Audio. He was program chair of ISMIR2005. In 2007 he was general chair of the 31st AES Conference, "New Directions in High-Resolution Audio," and in 2009 he was general secretary of the 35th AES International Conference, "Audio for Games." He was also chair of the recent 128th AES Convention in London.