Model-Based Inversion of Dynamic Range Compression

Stanislaw Gorlow, *Student Member, IEEE*, and Joshua D. Reiss, *Member, IEEE*

***Abstract*—In this work it is shown how a dynamic nonlinear time-variant operator, such as a dynamic range compressor, can be inverted using an explicit signal model. By knowing the model parameters that were used for compression one is able to recover the original uncompressed signal from a "broadcast" signal with high numerical accuracy and very low computational complexity. A compressor-decompressor scheme is worked out and described in detail. The approach is evaluated on real-world audio material with great success.**

***Index Terms*—Dynamic range compression, inversion, model-based, reverse audio engineering.**

I. Introduction

SOUND or audio engineering is an established discipline employed in many areas that are part of our everyday life without us taking notice of it. But not many know how the audio was produced. If we take sound recording and reproduction or broadcasting as an example, we may imagine that a prerecorded signal from an acoustic source is altered by an audio engineer in such a way that it corresponds to certain criteria when played back. The number of these criteria may be large and usually depends on the context. In general, the said alteration of the input signal is a sequence of numerous forward transformations, the reversibility of which is of little or no interest. But what if one wished to do exactly this, that is to reverse the transformation chain, and what is more, in a systematic and repeatable manner?

The research objective of *reverse audio engineering* is twofold: to identify the transformation parameters given the input and the output signals, as in [1], and to regain the input signal that goes with the output signal given the transformation parameters. In both cases, an explicit signal model is mandatory. The latter case might seem trivial, but only if the applied transformation is linear and orthogonal and as such perfectly invertible. Yet the forward transform is often neither linear nor invertible. This is the case for *dynamic range compression* (DRC), which is commonly described by a *dynamic nonlinear time-variant* system. The classical linear time-invariant (LTI) system theory does not apply here, so a tailored solution to the problem at hand must be found instead. At this point, we also like to highlight the fact that neither Volterra nor Wiener model approaches [2]–[4] offer a solution, and neither do describing functions [5], [6]. These are useful tools when identifying a *time-invariant* or a *slowly varying* nonlinear system or analyzing the *limit cycle* behavior of a feedback system with a static nonlinearity.

A method to invert dynamics compression is described in [7], but it requires an instantaneous gain value to be transmitted for each sample of the compressed signal. To provide a means to control the data rate, the gain signal is subsampled and also entropy coded. This approach is highly inefficient as it does not rely on a gain model and is extremely generic.

On the other hand, transmitting the uncompressed signal in conjunction with a few typical compression parameters like *threshold*, *ratio*, *attack*, and *release* would require a much smaller capacity and yield the best possible signal quality with regard to any thinkable measure. A more realistic scenario is when the uncompressed signal is not available on the consumer side. This is usually the case for studio music recordings and broadcast material where the listener is offered a signal that is meant to sound "good" to everyone. However, the loudness war [8] has resulted in *over-compressed* audio material. Over-compression makes a song lose its artistic features like excitingness or liveliness and desensitizes the ear thanks to a louder volume. There is a need to restore the original signal's dynamic range and to experience audio free of compression.

In addition to the normalization of the program's loudness level, the Dolby solution [9], [10] also includes *dynamic range expansion*. The expansion parameters that help reproduce the original program's dynamic range are tuned on the broadcaster side and transmitted as metadata together with the broadcast signal. This is a very convenient solution for broadcasters, not least because the metadata is quite compact. Dynamic range expansion is yet another forward transformation rather than a true inversion.

Evidently, none of the previous approaches satisfy the reverse engineering objective of this work. The goal of the present work, hence, is to *invert* dynamic range compression, which is a vital element not only in broadcasting but also in mastering. The paper is organized as follows. Section II provides a brief introduction to dynamic range compression and presents the compressor model upon which our considerations are based. The data model, the formulation of the problem, and the pursued approach are described next in Section III. The inversion

Manuscript received December 05, 2012; revised February 28, 2013; accepted February 28, 2013. Date of publication March 15, 2013; date of current version March 29, 2013. This work was supported in part by the "Agence Nationale de la Recherche" within the scope of the DReaM project (ANR-09-CORD-006) as well as the laboratory with which the first author is affiliated (see below) as part of the "mobilité juniors" program. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Woon-Seng Gan.

S. Gorlow is with the Computer Science Research Laboratory of Bordeaux (LaBRI), CNRS, Bordeaux 1 University, 33405 Talence Cedex, France (e-mail: stanislaw.gorlow@labri.fr).

J. D. Reiss is with the Centre for Digital Music (C4DM), Queen Mary, University of London, London E1 4NS, U.K. (e-mail: josh.reiss@eecs.qmul.ac.uk).

Digital Object Identifier 10.1109/TASL.2013.2253099

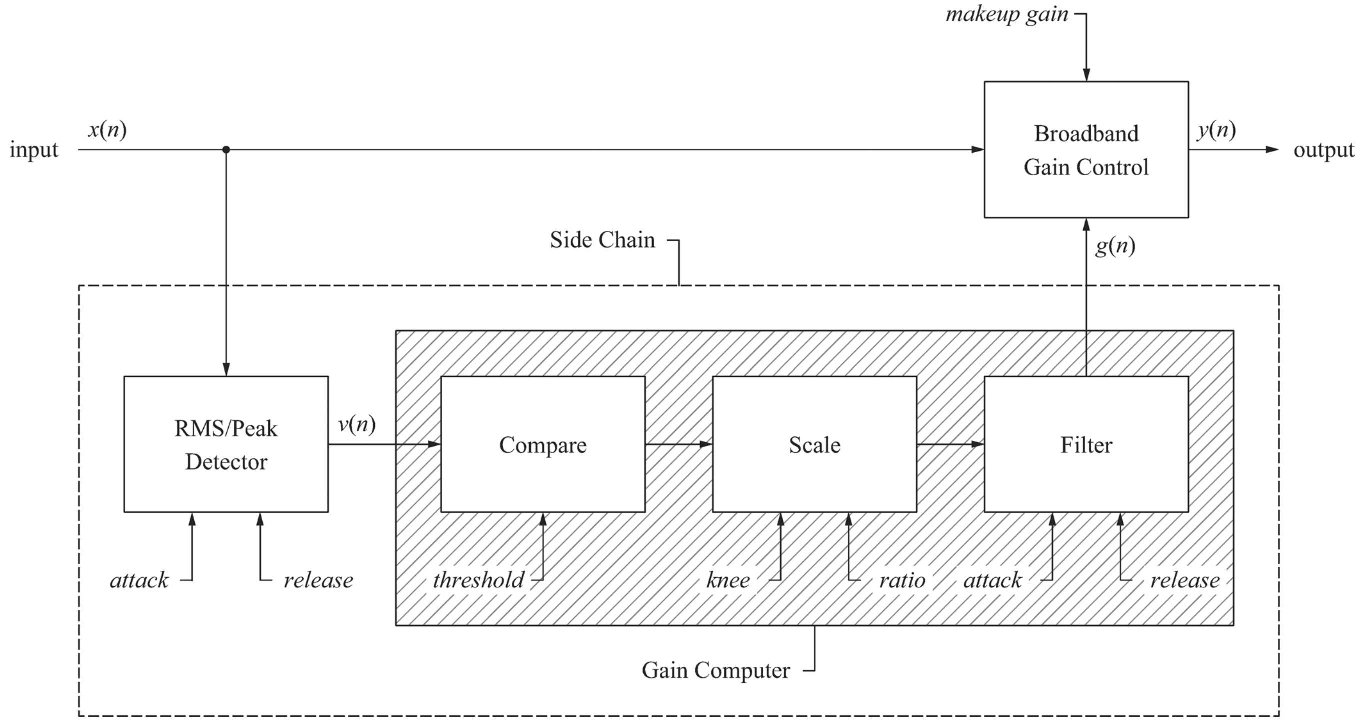

Fig. 1. Basic broadband compressor model (feed forward).

is discussed in detail in Section IV. Section V illustrates how an integral step of the inversion procedure, namely the search for the zero-crossing of a non-linear function, can be solved in an iterative manner by means of linearization. Some other compressor features are discussed in Section VI. The complete algorithm is given in the form of pseudocode in Section VII and its performance is evaluated for different compressor settings in Section VIII. Conclusions are drawn in Section IX, where some directions for future work are mentioned.

II. Dynamic Range Compression

Dynamic range compression or simply "compression" is a sound processing technique that attenuates loud sounds and/or amplifies quiet sounds, which in consequence leads to a reduction of an audio signal's dynamic range. The latter is defined as the difference between the loudest and quietest sound measured in decibel. In the following, we will use the word "compression" having "downward" compression in mind, though the discussed approach is likewise applicable to "upward" compression. Downward compressing means attenuating sounds above a certain threshold while leaving sounds below the threshold unchanged. A sound engineer might use a compressor to reduce the dynamic range of source material for purposes of aesthetics, intelligibility, recording or broadcast limitations.

Fig. 1 illustrates the *basic* compressor model from ([11], ch. 2) amended by a switchable RMS/peak detector in the side chain making it compatible with the compressor/limiter model from ([12], p. 106). We will hereafter restrict our considerations to this basic model, as the purpose of the present work is to demonstrate a general approach rather than a solution to a specific problem. First, the input signal is split and a copy is sent to the side chain. The detector then calculates the magnitude or level of the sidechain signal using the root mean square (RMS) or peak as a measure for how loud a sound is ([12], p. 107). The detector's temporal behavior is controlled by the attack and release parameters. The sound level is compared with the threshold level and, for the case it exceeds the threshold, a scale factor is calculated which corresponds to the ratio of input level to output level. The knee parameter determines how quick the compression ratio is reached. At the end of the side chain, the scale factor is fed to a smoothing filter that yields the gain. The response of the filter is controlled by another set of attack and release parameters. Finally, the gain control applies the smoothed gain to the input signal and adds a fixed amount of makeup gain to bring the output signal to a desired level. Such a broadband compressor operates on the input signal's full bandwidth, treating all frequencies from zero through the highest frequency equally. A detailed overview of all sidechain controls of a basic gain computer is given in ([11], ch. 3), e.g.,

III. Data Model, Problem Formulation, and Proposed Solution

A. Data Model and Problem Formulation

The employed data model is based on the compressor from Fig. 1. The following simplifications are additionally made: the knee parameter ("hard" knee) and the makeup gain (fixed at 0 dB) are ignored. The compressor is defined as a single-input single-output (SISO) system, that is both the input and the output are single-channel signals. What follows is a description of each block by means of a dedicated function.

The RMS/peak detector as well as the gain computer build upon a first-order (one-pole) lowpass filter. The sound level or envelope $v(n)$ of the input signal $x(n)$ is obtained by

$$\begin{aligned} \tilde{x}(n) &= \beta|x(n)|^p + \bar{\beta}\tilde{x}(n-1) \\ v(n) &= \sqrt[p]{\tilde{x}(n)} \end{aligned} \quad \text{with } p \in \{1, 2\}, \tag{1}$$

where $p = 2$ represents an RMS detector, and $p = 1$ a peak detector. The *non-zero* smoothing factor β, $0 < \beta \leqslant 1$, $\bar{\beta} = 1 - \beta$, may take on different values, $\beta_{\rm att}$ or $\beta_{\rm rel}$, depending on whether the detector is in the attack or release phase. The condition for the level detector to enter the attack phase and to choose $\beta_{\rm att}$ over $\beta_{\rm rel}$ is

$$|x(n)| > v(n-1). \tag{2}$$

A formula that converts a time constant τ into a smoothing factor is given in ([12], p. 109), so e.g.,

$$\beta = 1 - \exp\left[-2.2/(f_s \cdot \tau_v)\right],$$

where f_s is the sampling frequency. The static nonlinearity in the gain computer is usually modeled in the logarithmic domain as a continuous piecewise linear function:

$$F(n) = \begin{cases} -S \cdot [V(n) - L] & \text{if } V(n) > L \\ 0 & \text{otherwise} \end{cases}, \tag{3}$$

where S is the *slope*, $V(n) = 20\log_{10} v(n)$, and L is the threshold in decibel. The slope is further derived from the desired compression ratio R according to

$$S = 1 - \frac{1}{R}. \tag{4}$$

Equation (3) is equivalently expressed in the linear domain as

$$f(n) = \begin{cases} \kappa v^{-S}(n) & \text{if } v(n) > l \\ 1 & \text{otherwise} \end{cases}, \tag{5}$$

where $l = 10^{L/20}$, $\kappa = l^S$, and f is the linear scale factor before filtering. The smoothed gain g is then calculated as the exponentially-weighted moving average,

$$g(n) = \gamma f(n) + \bar{\gamma} g(n-1) \quad \text{with } \gamma \in \{\gamma_{\rm att}, \gamma_{\rm rel}\}, \tag{6}$$

where the decision for the gain computer to choose the attack smoothing factor $\gamma_{\rm att}$ instead of $\gamma_{\rm rel}$ is subject to

$$f(n) < g(n-1). \tag{7}$$

The output signal is finally obtained by multiplying the above gain with the input signal:

$$y(n) = g(n) \cdot x(n). \tag{8}$$

Due to the fact that the gain g is strictly positive, $0 < g \leqslant 1$, it follows that

$$\operatorname{sgn}(y) = \operatorname{sgn}(x), \tag{9}$$

where sgn is the signum or sign function. In consequence, it is convenient to factorize the input signal as a product of the sign and the modulus according to

$$x(n) = \operatorname{sgn}(x) \cdot |x(n)| \tag{10}$$

The problem at hand is formulated in the following manner: Given the compressed signal $y(n)$ and the model parameters

$$\boldsymbol{\theta} = [L \quad R \quad p \quad \beta_{\rm att} \quad \beta_{\rm rel} \quad \gamma_{\rm att} \quad \gamma_{\rm rel}],$$

recover the modulus of the original signal $|x(n)|$ from $|y(n)|$ based on $\boldsymbol{\theta}$. For a more intuitive use, the smoothing factors β and γ may be replaced by the time constants τ_v and τ_g. The meaning of each parameter is listed below.

L	The threshold in dB
R	The compression ratio $\mathrm{dB_{in}}$: $\mathrm{dB_{out}}$
p	The detector type (RMS or peak)
$\tau_{v,\rm att}$	The attack time of the envelope filter in ms
$\tau_{v,\rm rel}$	The release time of the envelope filter in ms
$\tau_{g,\rm att}$	The attack time of the gain filter in ms
$\tau_{g,\rm rel}$	The release time of the gain filter in ms

B. Proposed Solution

The output of the side chain, that is the gain of $|x(n)|$, given $\boldsymbol{\theta}$, $\tilde{x}(n-1)$, and $g(n-1)$, may be written as

$$g(n) = G[|x(n)| \mid \boldsymbol{\theta}, \tilde{x}(n-1), g(n-1)]. \tag{11}$$

In (11), G denotes a *nonlinear dynamic* operator that maps the modulus of the input signal $|x(n)|$ onto a sequence of instantaneous gain values $g(n)$ according to the compressor model represented by $\boldsymbol{\theta}$. Using (11), (8) can be solved for $|x(n)|$ yielding

$$|x(n)| = G^{-1}[g(n) \mid \boldsymbol{\theta}, \tilde{x}(n-1), g(n-1)] \cdot |y(n)|$$

subject to invertibility of G. In order to solve the above equation one requires the knowledge of $g(n)$, which is unavailable. However, since g is a function of $|x|$, we can express $|y|$ as a function of *one* independent variable $|x|$, and in that manner we obtain an equation with a single unknown:

$$|y(n)| = H[|x(n)| \mid \boldsymbol{\theta}, \tilde{x}(n-1), g(n-1)], \tag{12}$$

where H represents the entire compressor. If H is invertible, i.e., *bijective* for all n, $|x(n)|$ can be obtained from $|y(n)|$ by

$$|x(n)| = \begin{cases} H^{-1}[|y(n)| \mid \boldsymbol{\theta}, \ldots] & \text{if } v(n) > l \\ |y(n)| & \text{otherwise} \end{cases}. \tag{13}$$

And yet, since $v(n)$ is unknown, the condition for applying decompression must be predicted from $y(n)$, $\tilde{x}(n-1)$, and $g(n-1)$, and therefore needs the condition for toggling between the attack and release phases. Depending on the quality of the prediction, the recovered modulus $|z(n)|$ may differ somewhat at transition points from the original modulus $|x(n)|$, so that in the end

$$x(n) \approx \operatorname{sgn}(y) \cdot |z(n)| = z(n). \tag{14}$$

In the next section it is shown how such an *inverse compressor* or *decompressor* is derived.

IV. Inversion of Dynamic Range Compression

A. Characteristic Function

For simplicity, we choose the instantaneous envelope value $v(n)$ instead of $|x(n)|$ as the independent variable in (12). The relation between the two items is given by (1). From (6) and (8), when $v(n) > l$,

$$\begin{aligned} |y(n)| &= [\gamma f(n) + \bar{\gamma} g(n-1)] \cdot |x(n)| && (15)\\ &\overset{(5)}{=} [\gamma \kappa v^{-S}(n) + \bar{\gamma} g(n-1)] \cdot |x(n)|. && (16) \end{aligned}$$

From (1),

$$\begin{aligned} |y(n)| = {} & [\gamma \kappa v^{-S}(n) + \bar{\gamma} g(n-1)] \\ & \cdot \sqrt[p]{[v^p(n) - \bar{\beta}\tilde{x}(n-1)]/\beta}, \qquad (17) \end{aligned}$$

or equivalently (note that $\beta \neq 0$ by definition)

$$\begin{aligned} \beta |y(n)|^p = {} & [\gamma \kappa v^{-S}(n) + \bar{\gamma} g(n-1)]^p \\ & \cdot [v^p(n) - \bar{\beta}\tilde{x}(n-1)]. \qquad (18) \end{aligned}$$

Moreover, (18) has a unique solution if G and also H are invertible. Moving the expression on the left-hand side over to the right-hand side, we may define

$$\begin{aligned} \zeta_p(v) \triangleq {} & [\gamma \kappa v^{-S}(n) + \bar{\gamma} g(n-1)]^p \\ & \cdot [v^p(n) - \bar{\beta}\tilde{x}(n-1)] - \beta |y(n)|^p, \qquad (19) \end{aligned}$$

which shall be termed the *characteristic function*. The *root* or *zero-crossing* of $\zeta_p(v)$ hence represents the sought-after envelope value $v(n)$. Once $v(n)$ is found (see Section V), the current values of $\tilde{x}$, $|x|$, and g are updated as per

$$\begin{aligned} \tilde{x}(n) &= v^p(n) \\ |x(n)| &= \sqrt[p]{[\tilde{x}(n) - \bar{\beta}\tilde{x}(n-1)]/\beta} \\ g(n) &= |y(n)|/|x(n)| \qquad (20) \end{aligned}$$

and the decompressed sample is then calculated as

$$x(n) = \operatorname{sgn}(y) \cdot |x(n)|. \qquad (21)$$

B. Attack-Release Phase Toggle

1) Envelope Smoothing: In case a peak detector is in use, β takes on two different values. The condition for the attack phase is then given by (2) and is equivalent to

$$|x(n)|^p > \tilde{x}(n-1). \qquad (22)$$

Assuming that the past value of $\tilde{x}$ is known at time n, what is needed to be done is to express the unknown $|x|$ in terms of $|y|$ such that the above equation still holds true. If γ is rather small, $\gamma \leqslant 0.1 \ll 1$, or equivalently if τ_g is sufficiently large, $\tau_g \geqslant 0.5$ ms at 44.1-kHz sampling, the term $\gamma f(n)$ in (15) is negligible, so it approximates (15) as

$$|y(n)| \approx g(n-1) \cdot |x(n)|. \qquad (23)$$

Solving (23) for $|x(n)|$ and plugging the result into (22), we obtain

$$\left[\frac{|y(n)|}{g(n-1)}\right]^p > \tilde{x}(n-1). \qquad (24)$$

If (24) holds true, the detector is assumed to be in the attack phase.

2) Gain Smoothing: Just like the peak detector, the gain smoothing filter may be in either the attack or release phase. The *necessary* condition for the attack phase in (7) may also be formulated as

$$v(n) > \left[\frac{\kappa}{g(n-1)}\right]^{1/S} \quad \text{with } v(n) > l. \qquad (25)$$

But since the current envelope value is unknown, we need to substitute $v(n)$ in the above inequality by something that is known. With this in mind, (15) is rewritten as

$$\begin{aligned} |y(n)| &= \left[\gamma \frac{f(n)}{g(n-1)} + \bar{\gamma}\right] g(n-1) \cdot |x(n)| \\ &= \left[1 - \gamma\left(1 - \frac{f(n)}{g(n-1)}\right)\right] g(n-1) \cdot |x(n)|. \end{aligned} \qquad (26)$$

Provided that $f(n) < g(n-1)$, and due to the fact that $0 < \gamma \leqslant 1$, the expression in square brackets in (26) is smaller than one, and thus during attack

$$|y(n)| < g(n-1) \cdot |x(n)|. \qquad (27)$$

Substituting $|x(n)|$ by $\sqrt[p]{[v^p(n) - \bar{\beta}\tilde{x}(n-1)]/\beta}$ using (20), and solving (27) for $v(n)$ results in

$$v(n) > \sqrt[p]{\beta \left[\frac{|y(n)|}{g(n-1)}\right]^p + \bar{\beta}\tilde{x}(n-1)}. \qquad (28)$$

If $v(n)$ in (25) is substituted by the expression on the right-hand side of (28), (25) still holds true, so the following *sufficient* condition is used to predict the attack phase of the gain filter:

$$\sqrt[p]{\beta \left[\frac{|y(n)|}{g(n-1)}\right]^p + \bar{\beta}\tilde{x}(n-1)} > \left[\frac{\kappa}{g(n-1)}\right]^{1/S}. \qquad (29)$$

Note that the values of all variables are known whenever (29) is evaluated.

C. Envelope Predictor

An instantaneous estimate of the envelope value $v(n)$ is required not only to predict when compression is active, formally $v(n) > l$ according to (5), but also to initialize the iterative search algorithm in Section V. Resorting once more to (15) it can be noted that in the opposite case where $v(n) \leqslant l$, $f(n) = 1$, and so

$$|x(n)| = \frac{|y(n)|}{\gamma + \bar{\gamma} g(n-1)}. \tag{30}$$

The sound level of the input signal at time n is therefore

$$v(n) = \sqrt[p]{\beta \left[\frac{|y(n)|}{\gamma + \bar{\gamma} g(n-1)} \right]^p + \bar{\beta} \tilde{x}(n-1)}, \tag{31}$$

which must be greater than the threshold for compression to set in, whereas β and γ are selected based on (24) and (29), respectively.

D. Error Analysis

Consider $|x(n)|$ being estimated from $|y(n)|$ according to

$$|\hat{x}(n)| = \frac{|y(n)|}{g(n-1)}. \tag{32}$$

The normalized error is then

$$\begin{aligned} \hat{e}(n) &= \frac{|\hat{x}(n)| - |x(n)|}{|y(n)|} \qquad (33) \\ &= \left[\frac{|y(n)|}{g(n-1)} - \frac{|y(n)|}{g(n)} \right] \Big/ |y(n)| \\ &= \frac{g(n) - g(n-1)}{g(n) \cdot g(n-1)}. \qquad (34) \end{aligned}$$

As $g(n), g(n-1) > 0$, $\hat{e}(n) < 0$ during attack and $\hat{e}(n) \geqslant 0$ during release, respectively. The instantaneous gain $g(n)$ can also be expressed as

$$g(n) = \gamma \sum_{m=0}^{N} \bar{\gamma}^m f(n-m), \tag{35}$$

where N is the runtime in samples. Using (35) in (34), the magnitude of the error is given by

$$\begin{aligned} |\hat{e}(n)| &= \frac{\left| \sum_{m=0}^{N} \bar{\gamma}^m [f(n-m) - f(n-m-1)] \right|}{\gamma \sum_{i,j=0}^{N} \bar{\gamma}^{i+j} f(n-i) f(n-j-1)} \qquad (36) \\ &\leqslant \frac{\sum_{m=0}^{N} \bar{\gamma}^m |f(n-m) - f(n-m-1)|}{\gamma \sum_{i,j=0}^{N} \bar{\gamma}^{i+j} f(n-i) f(n-j-1)}. \qquad (37) \end{aligned}$$

For $\gamma = 1$, (36) becomes

$$|\hat{e}(n)|_{\gamma=1} = \frac{|f(n) - f(n-1)|}{f(n) \cdot f(n-1)}, \tag{38}$$

whereas for $\gamma \to 0$, (37) converges to infinity:

$$\begin{aligned} |\hat{e}(n)|_{\gamma \to 0} &\leqslant \frac{1}{\gamma} \frac{\sum_{m=0}^{N} \overbrace{|f(n-m) - f(n-m-1)|}^{>0 \text{ during compression}}}{\sum_{i,j=0}^{N} f(n-i) f(n-j-1)} \\ &\to \infty. \qquad (39) \end{aligned}$$

So, the error is smaller for large γ or short τ_g. The smallest possible error is for $\gamma = 1$, which then again depends on the current and the previous value of f. The error accumulates if $\gamma < 1$ with N. The difference between consecutive f-values is signal dependent. The signal envelope $v(n)$ fluctuates less and is thus smoother for smaller β or longer τ_v. $f(n)$ is also more stable when the compression ratio R is low. For $R = 1$, $f(n)$ is perfectly constant. The threshold L has a negative impact on error propagation. The lower L the more the error depends on N, since more samples are compressed with different f-values. The RMS detector stabilizes the envelope more than the peak detector, which also reduces the error. Furthermore, since usually $\tau_{\text{att}} < \tau_{\text{rel}}$, the error due to β is smaller during release whereas the error due to γ is smaller during attack. Finally, the error is expected to be larger at transition points between quiet to loud signal passages.

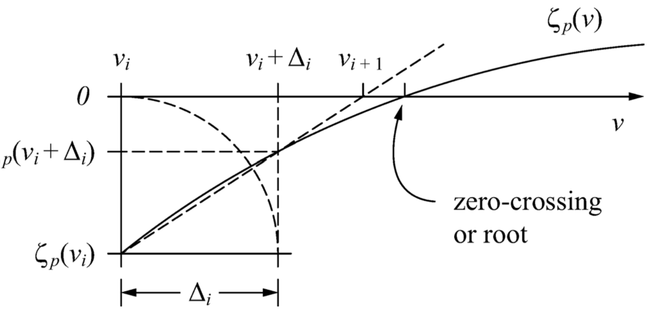

Fig. 2. Graphical illustration for the iterative search for the zero-crossing.

The above error may cause a decision in favor of a wrong smoothing factor β in (24), like β_{att} instead of β_{rel} e.g., The decision error from (24) then propagates to (29). Given that $\beta_{\text{att}} > \beta_{\text{rel}}$, the error due to (32) is accentuated by (24) with the consequence that (29) is less reliable than (24). The total error in (29) thus scales with $|\beta_{\text{att}} - \beta_{\text{rel}}|$. In regard to (31), reliability of the envelope's estimate is subject to validity of (24) and (29). A better estimate is obtained when the sound level detector and the gain filter are both in either the attack or release phase. Here too, the estimation error increases with $|\beta_{\text{att}} - \beta_{\text{rel}}|$ and also with $|\gamma_{\text{att}} - \gamma_{\text{rel}}|$.

V. Numerical Solution of the Characteristic Function

An approximate solution to the characteristic function can be found, e.g., by means of *linearization*. The estimate from (31) may moreover serve as a starting point for an iterative search of an optimum:

$$v_{\text{init}} = \sqrt[p]{\beta \left[\frac{|y(n)|}{\gamma + \bar{\gamma} g(n-1)} \right]^p + \bar{\beta} \tilde{x}(n-1)}.$$

The criterion for optimality is further chosen as the deviation of the characteristic function from zero, initialized to

$$\Delta_{\text{init}} = |\zeta_p(v_{\text{init}})|. \tag{40}$$

Thereupon, (19) may be approximated at a given point using the equation of a straight line, $\zeta = m \cdot v + c$, where m is the slope and c is the ζ-intercept. The zero-crossing is characterized by the equation

$$\frac{\zeta_p(v_i + \Delta_i) - \zeta_p(v_i)}{\Delta_i} \cdot v + \zeta_p(v_i) = 0, \tag{41}$$

as shown in Fig. 2. The new estimate of the optimal v is found as

$$v_{i+1} = v_i - \frac{\Delta_i \cdot \zeta_p(v_i)}{\zeta_p(v_i + \Delta_i) - \zeta_p(v_i)}. \tag{42}$$

If v_{i+1} is less optimal than v_i, the iteration is stopped and v_i is the final estimate. The iteration is also stopped if Δ_{i+1} is smaller than some ϵ. In the latter case, v_{i+1} has the optimal value with respect to the chosen criterion. Otherwise, v_i is set to v_{i+1} and Δ_i is set to Δ_{i+1} after every step and the procedure is repeated until v_{i+1} has converged to a more optimal value. The proposed method is a special form of the secant method with a single initial value v_{init}.

VI. General Remarks

A. Stereo Linking

When dealing with stereo signals, one might want to apply the same amount of gain reduction to both channels to prevent image shifting. This is achieved through *stereo linking*. One way is to calculate the required amount of gain reduction for each channel independently and then apply the larger amount to both channels. The question which arises in this context is which of the two channels was the gain derived from. To give an answer resolving the dilemma of ambiguity, one solution would be to signal which of the channels carries the applied gain. One could then decompress the marked sample and use its gain for the other channel. Although very simple to implement, this approach provokes an additional data rate of 44.1 kbps at 44.1-kHz sampling. A rate-efficient alternative that comes with a higher computational cost is realized in the following way. First, one decompresses both the left and the right channel independently and in so doing one obtains two estimates $z_l(n)$ and $z_r(n)$, where subscript l shall denote the left channel and subscript r the right channel, respectively. In a second step, one calculates the compressed values of $z_l(n)$ and $z_r(n)$ and selects the channel for which $H[z(n)] = y(n)$ holds true. In a final step, one updates the remaining variables using the gain of the selected channel.

B. Lookahead

A compressor with a *look-ahead* function, i.e., with a delay in the main signal path as in ([12], p. 106), uses past input samples as weighted output samples. Now that some future input samples are required to invert the process—which are unavailable, the inversion is rendered impossible. $g(n)$ and $x(n)$ must thus be *in sync* for the approach to be applied.

C. Clipping and Limiting

Another point worth mentioning is that "hard" clipping and "brick-wall" limiting are special cases of compression with the attack time set to zero and the compression ratio set to $\infty : 1$. The static nonlinearity F in that particular case is a one-to-many mapping, which by definition is *noninvertible*.

VII. The Algorithm

The complete algorithm is divided into three parts, each of them given as pseudocode below. Algorithm 1 outlines the compressor that corresponds to the model from Sections II–III. Algorithm 2 illustrates the decompressor described in Section IV, and the iterative search from Section V is finally summarized in Algorithm 3. The parameter f_s represents the sampling frequency in kHz.

Algorithm 1 **The compressor**

```
function COMP($x_n, \boldsymbol{\theta}, f_s$)
    $\tilde{x}_n \leftarrow 0$
    $g_n \leftarrow 1$
    for $n \leftarrow 1, N$ do
        if $|x_n|^p > \tilde{x}_n$ then
            $\beta \leftarrow 1 - \exp\left[-2.2/(f_s \cdot \tau_{v,\text{att}})\right]$
        else
            $\beta \leftarrow 1 - \exp\left[-2.2/(f_s \cdot \tau_{v,\text{rel}})\right]$
        end if
        $\tilde{x}_n \leftarrow \beta |x_n|^p + \bar{\beta} \tilde{x}_n$
        $v_n \leftarrow \sqrt[p]{\tilde{x}_n}$
        if $v_n > l$ then
            $f_n \leftarrow \kappa v_n^{-S}$
        else
            $f_n \leftarrow 1$
        end if
        if $f_n < g_n$ then
            $\gamma \leftarrow 1 - \exp\left[-2.2/(f_s \cdot \tau_{g,\text{att}})\right]$
        else
            $\gamma \leftarrow 1 - \exp\left[-2.2/(f_s \cdot \tau_{g,\text{rel}})\right]$
        end if
        $g_n \leftarrow \gamma f_n + \bar{\gamma} g_n$
        $y_n \leftarrow g_n x_n$
    end for
    return $y_n$
end function
```

VIII. Performance Evaluation

A. Performance Metrics

To evaluate the inverse approach, the following quantities are measured: the root-mean-square error (RMSE),

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{n=1}^{N} \left[z(n) - x(n)\right]^2}, \tag{43}$$

given in decibel relative to full scale (dBFS), the perceptual similarity between the original and decompressed signal, and the execution time of the decompressor relative to real time (RT). Furthermore, we present the percentage of compressed samples, the mean number of iterations until convergence per compressed sample, the error rate of the attack-release toggle for the gain smoothing filter, and finally the error rate of the envelope predictor. The perceptual similarity is assessed by PEMO-Q [13], [14] with PSM_t as metric. The simulations are run in MATLAB on an Intel Core i5-520M CPU.

Algorithm 2 The decompressor

```
function DECOMP($y_n, \boldsymbol{\theta}, \epsilon, f_s$)
    $\tilde{x}_n \leftarrow 0$
    $g_n \leftarrow 1$
    for $n \leftarrow 1, N$ do
        if $|y_n| > \sqrt[p]{\tilde{x}_n} \cdot g_n$ then
            $\beta \leftarrow 1 - \exp[-2.2/(f_s \cdot \tau_{v,\mathrm{att}})]$
        else
            $\beta \leftarrow 1 - \exp[-2.2/(f_s \cdot \tau_{v,\mathrm{rel}})]$
        end if
        if $|y_n| > \sqrt[p]{[(\kappa/g_n)^{p/S} - \bar{\beta}\tilde{x}_n]/\beta} \cdot g_n$ then
            $\gamma \leftarrow 1 - \exp[-2.2/(f_s \cdot \tau_{g,\mathrm{att}})]$
        else
            $\gamma \leftarrow 1 - \exp[-2.2/(f_s \cdot \tau_{g,\mathrm{rel}})]$
        end if
        if $|y_n| > \sqrt[p]{(l^p - \bar{\beta}\tilde{x}_n)/\beta} \cdot (\gamma + \bar{\gamma}g_n)$ then
            $v_n \leftarrow \sqrt[p]{\beta[|y_n|/(\gamma + \bar{\gamma}g_n)]^p + \bar{\beta}\tilde{x}_n}$
            $v_\mathrm{o} \leftarrow$ CHARFZERO($v_n, \epsilon$)
            $|x_n| \leftarrow \sqrt[p]{(v_\mathrm{o}^p - \bar{\beta}\tilde{x}_n)/\beta}$
            $\tilde{x}_n \leftarrow v_\mathrm{o}^p$
            $g_n \leftarrow |y_n|/|x_n|$
        else
            $g_n \leftarrow \gamma + \bar{\gamma}g_n$
            $|x_n| \leftarrow |y_n|/g_n$
            $\tilde{x}_n \leftarrow \beta|x_n|^p + \bar{\beta}\tilde{x}_n$
        end if
        $x_n \leftarrow \mathrm{sgn}(y_n) \cdot |x_n|$
    end for
    return $x_n$
end function
```

Algorithm 3 The iterative search for the zero-crossing

```
function CHARFZERO($v_n, \epsilon$)
    $v_i \leftarrow v_n$
    repeat
        $\Delta_i \leftarrow |\zeta_p(v_i)|$
        $v_i \leftarrow v_i - \Delta_i \cdot \zeta_p(v_i)/[\zeta_p(v_i + \Delta_i) - \zeta_p(v_i)]$
        if $|\zeta_p(v_i)| > \Delta_i$ then
            return $v_n$
        end if
        $v_n \leftarrow v_i$
    until $|\zeta_p(v_i)| < \epsilon$
    return $v_i$
end function
```

B. Computational Results

Fig. 3 shows the inverse output signal $z(n)$ for a synthetic input signal $x(n)$ using an RMS detector. The inverse signal is obtained from the compressed signal $y(n)$ with an error of -129 dBFS. It is visually indistinguishable from the original signal $x(n)$. Due to the fact that the signal envelope is constant most of the time, the error is noticeable only around transition points—which are few. The decompressor's performance is further evaluated for some commercial compressor presets. The used audio material consists of 12 items covering speech, sung voice, music, and jingles. All items are normalized to -16 LKFS [15]. The ϵ-value in the break condition of Algorithm 3 is set to $1 \cdot 10^{-12}$. A detailed overview of compressor settings and performance figures is given in Tables I–II. The presented results suggest that the decompressed signal is perceptually indistinguishable from the original—the PSM_t-value is flawless. This was also confirmed by the authors through informal listening tests.

As can be seen from Table II, the largest inversion error is associated with setting E and the smallest with setting B. For all five settings, the error is larger when an RMS detector is in use. This is partly due to the fact that $\zeta_2(v)$ has a stronger curvature in comparison to $\zeta_1(v)$. By defining the distance in (40) as $\Delta \triangleq \sqrt[p]{|\zeta_p(v)|}$, it is possible to attain a smaller error for an RMS detector at the cost of a slightly longer runtime. In most cases, the envelope predictor works more reliably as compared to the toggle switch between attack and release. It can also be observed that the choice of time constants seems to have little impact on decompressor's accuracy. The major parameters that affect the decompressor's performance are L and R, while the threshold is evidently the predominant one: the RMSE strongly correlates with the threshold level.

Figs. 4–5 show the inversion error as a function of various time constants. These are in the range of typical attack and release times for a limiter (peak) or compressor (RMS) ([12], pp. 109–110). It can be observed that the inversion accuracy depends on the release time of the peak detector and not so much on its attack time for both the envelope and the gain filter, see Figs. 4, 5(b). For the envelope filter, all error curves exhibit a local dip around a release time of 0.5 s. The error increases steeply below that bound but moderately with larger values. In the proximity of 5 s, the error converges to -130 dBFS. With regard to the gain filter, the error behaves in a reverse manner. The curves in Fig. 5(b) exhibit a local peak around 0.5 s with a value of -180 dBFS. It can further be observed in Fig. 4(a) that the curve for $\tau_{v,\mathrm{rel}} = 1$ ms has a dip where $\tau_{v,\mathrm{att}}$ is close to 1 ms, i.e., where $|\beta_\mathrm{att} - \beta_\mathrm{rel}|$ is minimal. This is also true for Fig. 4(c) and (d): the lowest error is where the attack and release times are identical. As a general rule, the error that is due to the attack-release switch is smaller for the gain filter in Fig. 5.

Looking at Fig. 6 one can see that the error decreases with threshold and increases with compression ratio. At a ratio of 10:1 and beyond, the RMSE scales almost exclusively with the threshold. The lower the threshold, the stronger the error propagates between decompressed samples, which leads to a larger

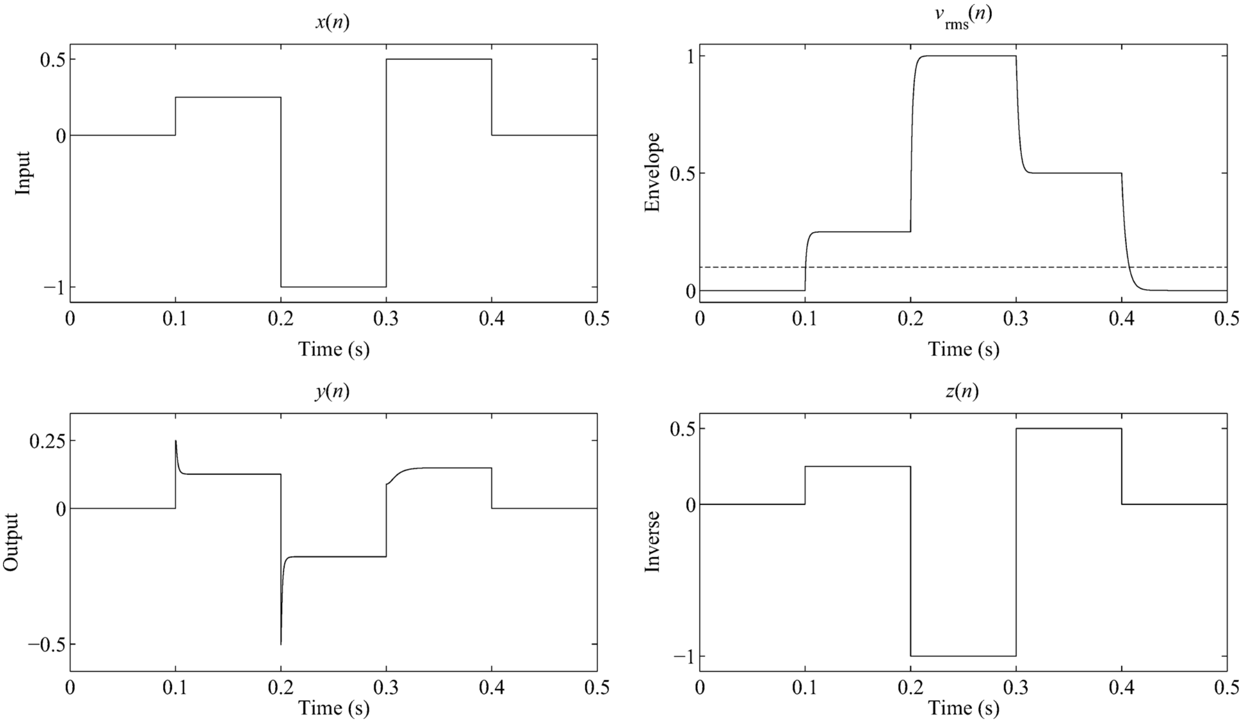

Fig. 3. An illustrative example using an RMS amplitude detector with τ_v set to 5 ms, a threshold of -20 dBFS (dashed line in the upper right corner), a compression ratio of 4:1, and τ_g set to 1.6 ms for attack and 17 ms for release, respectively. The RMSE is -129 dBFS.

TABLE I
SELECTED COMPRESSOR SETTINGS

Parameter	Description	A	B	C	D	E
L (dBFS)	Threshold	-32.0	-19.9	-24.4	-26.3	-38.0
R (dB_{in} : dB_{out})	Ratio	$3.0:1$	$1.8:1$	$3.2:1$	$7.3:1$	$4.9:1$
$\tau_{v,\text{att}}$ (ms)	Envelope attack	5.0	5.0	5.0	5.0	5.0
$\tau_{v,\text{rel}}$ (ms)	Envelope release					
$\tau_{g,\text{att}}$ (ms)	Gain attack	13.0	11.0	5.8	9.0	13.1
$\tau_{g,\text{rel}}$ (ms)	Gain release	435	49	112	705	257

TABLE II
PERFORMANCE FIGURES OBTAINED FOR VARIOUS AUDIO MATERIAL (12 ITEMS)

	A		B		C		D		E	
	Peak	RMS	Peak	RMS	Peak	RMS	Peak	RMS	Peak	RMS
RMSE (dBFS)	-74.4	-71.2	-97.2	-93.7	-81.0	-77.8	-76.3	-69.5	-63.2	-53.8
PSM_t (PEMO-Q)	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Execution time (RT)	0.54	0.53	0.40	0.44	0.47	0.49	0.48	0.50	0.54	0.54
Compression rate (%)	78.7	80.8	38.5	50.7	61.8	67.3	67.6	71.8	85.2	86.4
Iterations per sample (#)	1.04	1.02	1.00	1.01	1.07	1.06	1.05	1.03	1.09	1.04
Attack-release error rate (%)	0.05	0.09	0.01	0.01	0.02	0.04	0.01	0.03	0.14	0.51
State error rate (%)	0.02	0.03	0.01	0.01	0.01	0.02	0.02	0.03	0.03	0.05

RMSE value. The RMS detector further augments the error because it stabilizes the envelope $v(n)$ more than the peak detector. Clearly, the threshold level has the highest impact on the decompressor's accuracy.

IX. CONCLUSION AND OUTLOOK

This work examines the problem of finding an inverse to a nonlinear dynamic operator such as a digital compressor. The proposed approach is characterized by the fact that it uses an explicit signal model to solve the problem. To find the "dry" or uncompressed signal with high accuracy, it is sufficient to know the model parameters. The parameters can e.g., be sent together with the "wet" or compressed signal in the form of metadata as is the case with Dolby Volume and ReplayGain [16]. A new bitstream format is not mandatory, since many digital audio standards, like WAV or MP3, provide means to tag the audio con-

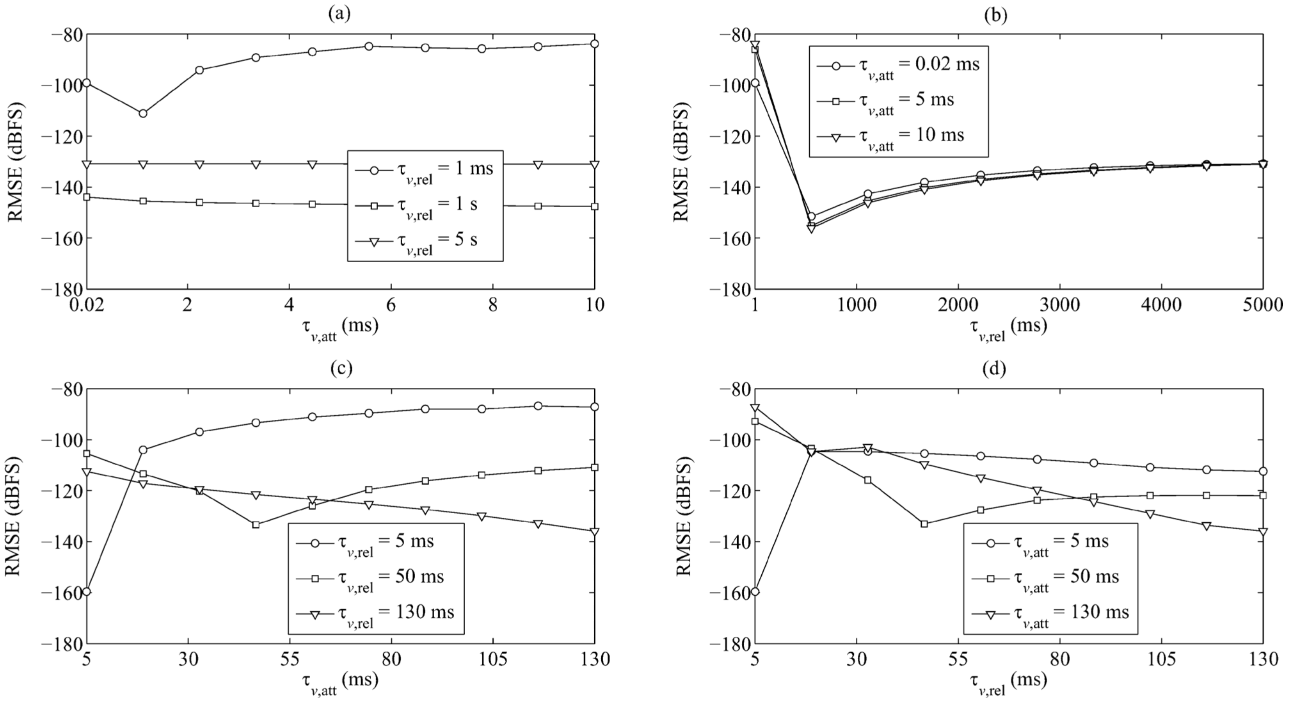

Fig. 4. RMSE as a function of typical attack and release times using a peak (upper row) or an RMS amplitude detector (lower row). In the left column, the attack time of the envelope filter is varied while the release time is held constant. The right column shows the reverse case. The time constants of the gain filter are fixed at zero. In all four cases, threshold and ratio are fixed at −32 dBFS and 4:1, respectively.

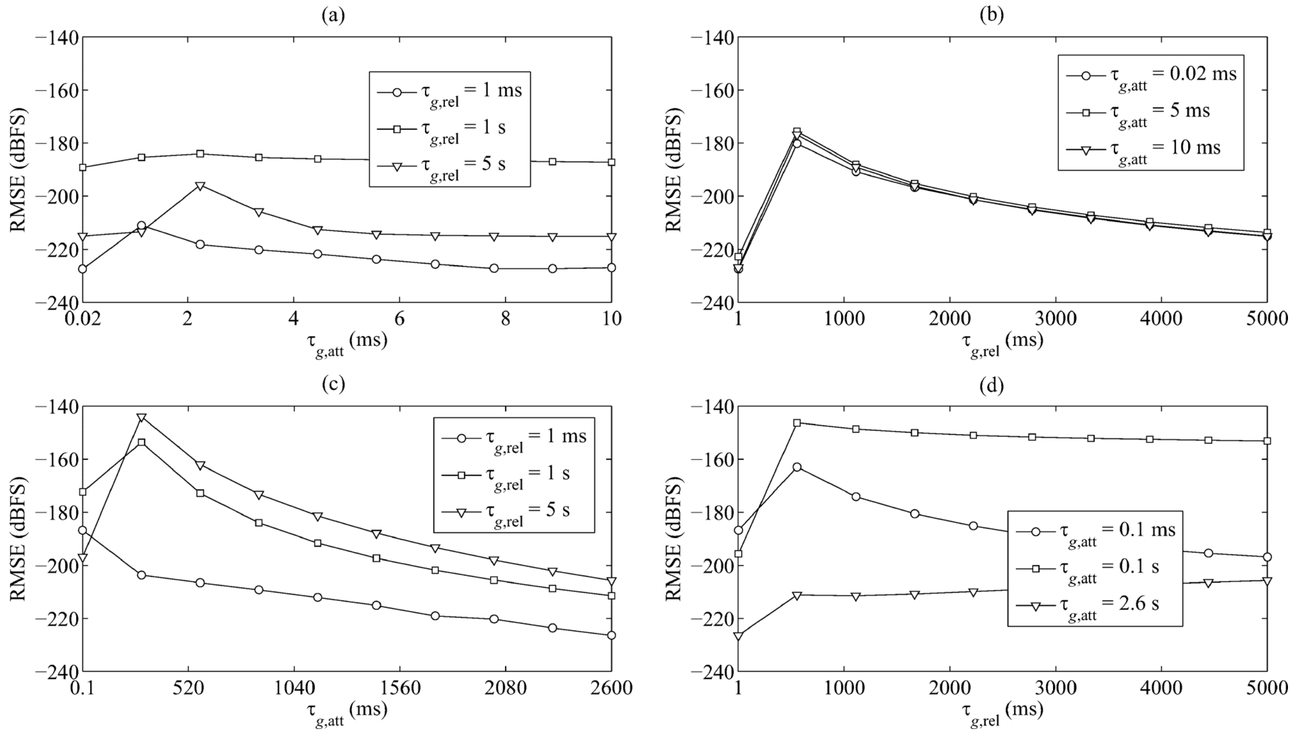

Fig. 5. RMSE as a function of typical attack and release times using a peak (upper row) or an RMS amplitude detector (lower row). In the left column, the attack time of the gain filter is varied while the release time is held constant. The right column shows the reverse case. The time constants of the envelope filter are fixed at zero. In all four cases, threshold and ratio are fixed at −32 dBFS and 4:1, respectively.

tent with "ancillary" data. With the help of the metadata, one can then reverse the compression applied after mixing or before broadcast. This allows the end user to have control over the amount of compression, which may be preferred because the sound engineer has no control over the playback environment or the listener's individual taste.

When the compressor parameters are unavailable, they can possibly be estimated from the compressed signal. This may

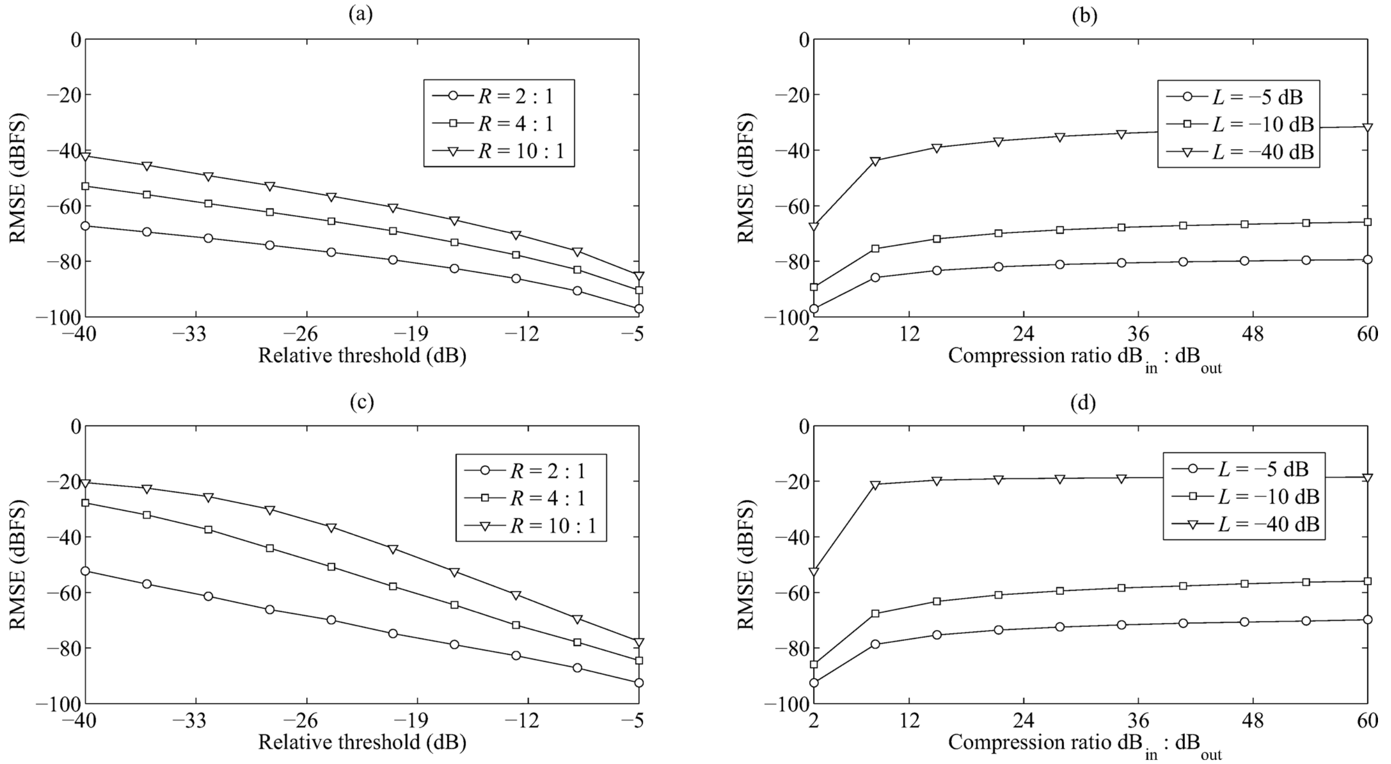

Fig. 6. RMSE as a function of threshold relative to the signal's average loudness level (left column) and compression ratio (right column) using a peak (upper row) or an RMS amplitude detector (lower row). The time constants are: $\tau_v = 5$ ms, $\tau_{g,\mathrm{att}} = 20$ ms, and $\tau_{g,\mathrm{rel}} = 1$ s.

thus be a direction for future work. Another direction would be to apply the approach to more sophisticated models that include a "soft" knee, parallel and multiband compression, or perform gain smoothing in the logarithmic domain, see [11], [12], [17], [18] and references therein.

In conclusion, we want to draw the reader's attention to the fact that the presented figures suggest that the decompressor is realtime capable which can pave the way for exciting new applications. One such application could be the restoration of dynamics in over-compressed audio or else the accentuation of transient components, see [19]–[21], by an adaptively tuned decompressor that has *no prior knowledge* of the compressor parameters.

Acknowledgment

This work was carried out in part at the Centre for Digital Music (C4DM), Queen Mary, University of London.

References

[1] D. Barchiesi and J. Reiss, "Reverse engineering of a mix," *J. Audio Eng. Soc.*, vol. 58, pp. 563–576, 2010.

[2] T. Ogunfunmi, *Adaptive Nonlinear System Identification: The Volterra and Wiener Model Approaches*. New York, NY, USA: Springer Science+Business Media, 2007, ch. 3.

[3] Y. Avargel and I. Cohen, "Adaptive nonlinear system identification in the short-time Fourier transform domain," *IEEE Trans. Signal Process.*, vol. 57, no. 10, pp. 3891–3904, Oct. 2009.

[4] Y. Avargel and I. Cohen, "Modeling and identification of nonlinear systems in the short-time Fourier transform domain," *IEEE Trans. Signal Process.*, vol. 58, no. 1, pp. 291–304, Jan. 2010.

[5] A. Gelb and W. E. Vander Velde, *Multiple-Input Describing Functions and Nonlinear System Design*. New York, NY, USA: McGraw-Hill, 1968, ch. 1.

[6] P. W. J. M. Nuij, O. H. Bosgra, and M. Steinbuch, "Higher-order sinusoidal input describing functions for the analysis of non-linear systems with harmonic responses," *Mech. Syst. Signal Process.*, vol. 20, pp. 1883–1904, 2006.

[7] B. Lachaise and L. Daudet, "Inverting dynamics compression with minimal side information," in *Proc. DAFx*, 2008, pp. 1–6.

[8] E. Vickers, "The loudness war: Background, speculation and recommendations," in *Proc. AES Conv. 129*, Nov. 2010.

[9] Dolby Digital and Dolby Volume Provide a Comprehensive Loudness Solution, Dolby Laboratories, 2007.

[10] Broadcast Loudness Issues: The Comprehensive Dolby Approach, Dolby Laboratories, 2011.

[11] R. Jeffs, S. Holden, and D. Bohn, Dynamics processor—Technology & Application Tips, Rane Corporation, 2005.

[12] U. Zölzer, *DAFX: Digital Audio Effects*, 2nd ed. Chichester, West Sussex, U.K.: Wiley, 2011, ch. 4, The Atrium, Southern Gate, PO19 8SQ.

[13] R. Huber and B. Kollmeier, "PEMO-Q—A new method for objective audio quality assessment using a model of auditory perception," *IEEE Trans. Audio Speech Lang. Process.*, vol. 14, no. 6, pp. 1902–1911, Nov. 2006.

[14] HörTech gGmbH, PEMO-Q [Online]. Available: http://www.hoertech.de/web_en/produkte/pemo-q.shtml, version 1.3

[15] ITU-R, Algorithms to Measure Audio Programme Loudness and True-Peak Audio Level, Mar. 2011, rec. ITU-R BS.1770-2.

[16] Hydrogenaudio, ReplayGain [Online]. Available: http://wiki.hydrogenaudio.org/index.php?title=ReplayGain, Feb. 2013

[17] J. C. Schmidt and J. C. Rutledge, "Multichannel dynamic range compression for music signals," in *Proc. IEEE ICASSP*, 1996, vol. 2, pp. 1013–1016.

[18] D. Giannoulis, M. Massberg, and J. D. Reiss, "Digital dynamic range compressor design—A tutorial and analysis," *J. Audio Eng. Soc.*, vol. 60, pp. 399–408, 2012.

[19] M. M. Goodwin and C. Avendano, "Frequency-domain algorithms for audio signal enhancement based on transient modification," *J. Audio Eng. Soc.*, vol. 54, pp. 827–840, 2006.

[20] M. Walsh, E. Stein, and J.-M. Jot, "Adaptive dynamics enhancement," in *Proc. AES Conv. 130*, May 2011.

[21] M. Zaunschirm, J. D. Reiss, and A. Klapuri, "A sub-band approach to modification of musical transients," *Comput. Music J.*, vol. 36, pp. 23–36, 2012.

Stanislaw Gorlow (S'11) received the B.Eng. degree in information technology from Georg Simon Ohm University of Applied Sciences Nuremberg, Germany, in 2007 and the M.Sc. degree in electrical engineering and information technology from Ilmenau University of Technology, Germany, in 2010. He is currently pursuing a Ph.D. degree in the computer science research laboratory of Bordeaux (LaBRI) at Bordeaux 1 University, France.

He spent his military service as staff duty soldier with the German Army in Amberg between 2001–2002. During the period between 2005–2007 he went from being an intern to an employee at Dolby Germany, Nuremberg. In 2009, in parallel to his studies he also assisted in the Communications Research Laboratory (CRL) at the Institute for Information Technology in Ilmenau. His research interests are in digital speech and audio signal processing and related fields.

Mr. Gorlow is a member of the Audio Engineering Society (AES) and the European Association for Signal Processing (EURASIP).

Joshua D. Reiss (M'07) received the Ph.D. degree in physics from the Georgia Institute of Technology, Atlanta, USA, specializing in analysis of nonlinear systems.

He is a Senior Lecturer with the Centre for Digital Music, Queen Mary, University of London, 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 multichannel and real-time audio signal processing, 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 the above topics, is on the use of state-of-the-art signal processing techniques for professional sound engineering.