EXACT DISCRETE-TIME REALIZATION OF A DOLBY B ENCODING/DECODING ARCHITECTURE

Federico Avanzini

Dipartimento di Ing. dell'Informazione
Università di Padova
Via Gradenigo 6/A, Padova 35131, Italy
avanzini@dei.unipd.it

Federico Fontana

Dipartimento di Informatica
Università di Verona
Strada le Grazie 15, Verona 37134, Italy
federico.fontana@univr.it

ABSTRACT

An algebraic technique which computes nonlinear, delay-free digital filter networks is applied to model the Dolby B in the discrete-time. The model preserves the topology of the analog system, and imports the characteristics of the nonlinear processing blocks which are responsible of the peculiar functioning of Dolby B. The resulting numerical system exhibits qualitatively similar dynamic behavior and performance – full compliance with the Dolby B specifications would be achieved by deriving, from comprehensive data sheets of the system, accurate discrete-time models of the analog processing blocks. Results demonstrate that the computation converges if proper iterative methods are employed.

1. INTRODUCTION

The history of audio effects design traces back to the world of analog circuits. It was not long after the advent of digital architectures that scientists considered the possibility to reproduce in the digital domain the analog and electro-acoustic mechanisms the early audio effects were based upon.

Converting a continuous-time process into a sequence of discrete computations inevitably introduces approximations, which in some cases can generate intolerable problems, like heavy artifacts in the system response, or even instability of the discrete-time model. The *delay-free loop problem* [1, sec. 6.1.3] refers to the presence in a network of feedback paths that are not computable, meaning with this that the computation cannot be executed sequentially due to the lack of pure delays along the loop. This problem can appear in particular during conversion to the digital domain of analog filter networks, or even in digital-to-digital domain transformations (such as frequency-warping mappings [2]).

If the network is linear, various techniques can be used to convert a continuous-time system into an equivalent numerical one, working either in the time or in the Laplace domains. As an example, wave methods [3] and transfer function models [4] have been widely applied to the numerical simulation of acoustic systems. Moreover, a linear network can be always rearranged into a new one in which delay-free paths are solved by composing the filters belonging to them into bigger linear structures that "embed" the loop [1]. Nevertheless there are cases where this rearrangement is deprecated (e.g., situations in which the access to the filter parameters becomes too complicated after the rearrangement). Furthermore, the elimination of a delay-free path implies that all the branches belonging to it cannot be used any longer as input/output points where to inject/extract the signal to/from the system: this point is particularly relevant in the design of virtual musical instruments by physical modeling.

When nonlinearities exist in the continuous-time system, however, the discretization procedure must preserve stability and must ensure a precise simulation of the nonlinear characteristic. Moreover, if a nonlinearity is part of a delay-free path there is no general procedure to rearrange the loop to realize a new linear structure in which to embed the delay-free path.

A technique to compute linear delay-free paths without topology rearrangement was proposed in [5]. It was applied to warped IIR filter computation [2] and to magnitude-complementary parametric equalizers [6], and generalized to linear filter networks with arbitrary delay-free path configurations [7]. It was then extended to networks containing nonlinear blocks [8, 9]. The technique assumes that each linear and nonlinear block has been already individually modeled in the discrete time domain.

In this paper we analyze the Dolby B codec as a challenging example of analog system that includes nonlinear feedback loops. We show that the system can be realized *exactly* in the discrete-time domain by employing the numerical technique presented in [8, 9]. Sections 2 and 3 review the functioning of the analog system and of existing discrete-time realizations. Sections 4 and 5 discuss the exact realization proposed in this paper and analyze the convergence of the iterative schemes used to solve the nonlinear digital network. Finally, results from numerical simulations of the system are presented in Section 6.

2. THE DOLBY B CODEC

At the mid of the sixties Dolby Laboratories introduced a family of noise reduction systems that had a strong impact on the industry of consumer audio which, at that time, was experiencing the explosion of the compact cassette [10].

Dolby noise reduction systems gave the most successful solution to the problem of noise floor generated by the tape, clearly audible in the mid and high frequencies unless masked by a moderately loud audio message, and, hence, preventing music listening at Hi-Fi standards. While convincingly reducing noise floor, the Dolby coding paradigm did not heavily mask the audio content. In this way music from a Dolby encoded cassette could still be listened to even by a normal tape player [11]. Due to this versatility, Dolby systems marked a commercial advantage against other noise reduction architectures such as DBX, Tel Com by Telefunken and DNL by Philips.

Among the many noise reduction systems licensed by Dolby Laboratories [10], the B architecture was one of the most successful. As its predecessor, the A system, Dolby B is based on the

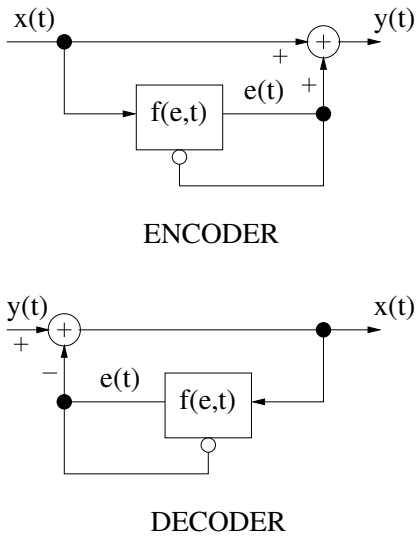

Figure 1: *Dolby B codec.*

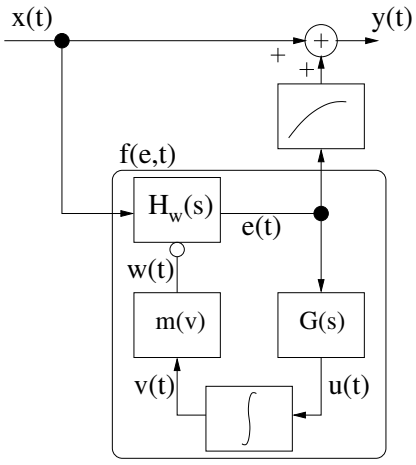

Figure 2: *Dolby B encoder.*

idea that tape noise floor has to be canceled only when the dynamics of the audio signal is too low to mask it, otherwise the music message can be left untouched. For this reason, before recording the signal on the tape, the encoder emphasizes the mid and high frequency range under low dynamics conditions. Coherently, the decoder de-emphasizes the same frequencies during audio tape reproduction under similar dynamic conditions. The overall result is a proportional decrease of the cassette noise.

The Dolby B codec is shown in Figure 1. According to this scheme, the encoded signal $y(t)$ is obtained by summing, to the unencoded signal $x(t)$, a filtered version $e(t)$ of the same unencoded signal:

$$y(t) = x(t) + e(t) = \{1 + f(e,t)\} * x(t)\,. \qquad (1)$$

In Eq. (1) we recognize the filter f to be nonlinear and time-varying: both characteristics are mandatory if we want to process the signal depending on its dynamics. More in detail, we see that the encoded message depends on the characteristics of the signal captured at the time-varying filter output (the more obvious choice of reading the dynamics before encoding was implemented by Dolby A, then abandoned). In particular, the higher the amplitude of e, the smoother the high-pass characteristic of f. In the limit case when e has a very pronounced dynamics, then f becomes nearly transparent, i.e., $f(e,t) \approx 1$.

Conversely, and assuming the inverse transfer characteristic of $1 + f(e,t)$ to be stable, the decoder realizes the following nonlinear transfer function against the noisy version $\tilde{y}(t)$ of the encoded signal:

$$\tilde{x}(t) = \tilde{y}(t) - \tilde{e}(t) = \tilde{y}(t) - f(\tilde{e},t) * \tilde{x}(t)\,. \qquad (2)$$

In Eq. (2) we have neglected transmission delay. Noticeably, if no noise or whatever audio artifact superimposes to $y(t)$, i.e., if $\tilde{y}(t) = y(t)$, then decoding is error-free, i.e., $\tilde{x}(t) = x(t)$. Of course noise-free transmission is unrealistic: should it exist, then noise reduction systems would have turn out to be unnecessary.

3. DIGITAL REALIZATIONS OF DOLBY B

Figure 1 immediately shows that implementing a Dolby B encoder in the digital domain is difficult due to the existence of a delay-free path connecting the time-varying filter output to its control input.

A more detailed inspection of the filter (see the schematic in Figure 2, disclosed by Dolby Labs [10]) reveals that control is realized by passing the signal $e(t)$ through a smooth high-pass linear filter $G(s)$ yielding a signal $u(t)$, and, then, through an envelope follower. This stage rectifies the audio message into a control signal $v(t)$ that instantaneously drives, by means of the output w from the map $m(v)$, the high-pass characteristic of a time-varying linear filter $H_w(s)$ whose output, depending on the value taken by w, is finally added to the original signal to form the encoded message.

In this explanation (but not in Figure 2) we have omitted the presence of a compressor, located between the time-varying filter and the adder, whose role is to remove from $e(t)$ overshoots arising when the input suddenly switches from very low to very high dynamics. In this case, a short but audible time window occurs in which the mid and high frequencies in $x(t)$ are mistakenly amplified until the system responds to the high dynamics by positioning the time-varying filter $H_w(s)$ to almost transparent behavior. This compressor, hence, prevents such high-frequency boosts to come out from the system by filtering them out from $e(t)$ until its amplitude falls within the correct dynamic range. Otherwise, e.g. for medium and low dynamics, the compressor is transparent.

In spite of this fairly complicate network, a digital encoder can be (at least in principle) realized by assuming that $v(t)$ is enough slowly varying to allow for the inclusion of a fictitious unit delay between the envelope follower and the map. Since the signal rectification performed by the envelope follower indeed results in a slowly varying output, then a computational scheme for a discrete-time Dolby B encoder can be figured out which, starting from an initial state plus an initial value for w (e.g. those resulting from a null input) computes e known x and, then, the new state and samples for all the block outputs, including y.

It is clear that this computing procedure requires that every analog processing block in Figure 2 has been somehow converted to the digital domain, possibly taking the unavoidable distortions introduced by the presence of the fictitious delay into account.

The digital conversion of the decoder is a much more com-

plicate matter. If one reports the internal structure of $f(e,t)$ (see Figure 2) into the block scheme of the dolby B decoder (see Figure 1), then a double feedback can be immediately noticed: one at the subtraction point $e(t) - y(t)$ and one, already explained, at the input control of the time-varying filter. The former, in particular, cannot be translated into a corresponding discrete time feedback loop by including a unit delay, since the audio signal flowing along it is by no means slowly varying. Furthermore, both feedbacks include nonlinear blocks in between.

Alternative structures aimed at realizing a real-time Dolby B decoder in the digital domain have been successfully engineered by Philips [12] and STMicroelectronics [13, 14]. Such structures include a fictitious time delay to account for the instantaneous feedback at the time-varying filter control, and rearrange the other feedback loop into an alternative, feed-forward topology. As a consequence, most of the parameters characterizing the circuitry of the Dolby B analog blocks must be carefully adapted for the resulting digital decoder to match the specification requirements imposed by Dolby Labs.

In particular, the discrete-time Dolby B proposed by STMicroelectronics [13, 14] contains careful transpositions of all analog stages into corresponding digital blocks. The transfer characteristic $X(s)/Y(s)$ is modeled by a time-varying feed-forward digital filter whose transfer function is given by

$$\frac{X(z)}{Y(z)} = \frac{1}{1 + H_w(z)}\,. \tag{3}$$

The filter $G(z)$, digital counterpart of $G(s)$, is driven by a discrete-time model of the transfer characteristic $E(s)/Y(s)$:

$$\frac{E(z)}{Y(z)} = \frac{H_w(z)}{1 + H_w(z)} = 1 - \frac{X(z)}{Y(z)}\,. \tag{4}$$

In this way a single time-varying digital filter, whose transfer function is given by (3), is used both to compute the system output and, via Eq. (4), to feed the filter $G(z)$. Obviously, such a realization requires to re-design the nonlinear map $m(v)$ properly.

4. EXACT TRANSPOSITION OF THE ENCODING/DECODING NETWORK

In [8] we have presented an algebraic technique which allows to compute every nonlinear filter network, regardless of the existence of delay-free paths located in between processing blocks. The same technique also promises to add insight on the inherent stability properties of a nonlinear system structured, like Dolby B, as an interconnection of input/output blocks [9].

In order to exploit this technique to realize a Dolby B architecture exactly, we will develop a 'mock-up' of the system which, in particular, preserves all the nonlinearities and topological details of the analog structure. However, we will avoid to carefully transpose into discrete time the transfer characteristics of the Dolby B analog blocks. Instead, we will rely on simplified digital versions of the same blocks without sacrificing in generality.

As a result, we have realized a digital "Dolby B-like" system working at 44100 Hz which, in spite of its resemblance to the real Dolby B both in structure and performance, does not comply with the requirements of Dolby Labs due to discrepancies between the transfer characteristics of the analog blocks forming the original system and their digital transpositions. As to the question whether a Dolby B codec can be *exactly* realized in discrete time, our answer is 'yes' as far as every block of the analog architecture is exactly translated into the digital domain using proper transformation methods [15, 1].

In the remainder of this section we adopt the notation used in [8, 9]: linear blocks are defined through their transfer functions H_i and nonlinear blocks are defined through their nonlinear characteristics f_i, with $i = 1, 2, \ldots$. Inputs and outputs are denoted as x_{Li}, y_{Li} (linear blocks) and x_{Ni}, y_{Ni} (nonlinear blocks). We employ the following digital blocks (refer to Figure 2 for nomenclature of the continuous-time blocks):

- The time-varying filter $H_w(s)$ is replaced by a digital high-frequency shelving filter [1]

$$\begin{aligned} H_w(z) &= H_1(z) + wH_2(z) \\ &= \frac{1}{2}\{1 - A(z)\} + \frac{w}{2}\{1 + A(z)\} \end{aligned} \tag{5}$$

  with

$$A(z) = \frac{\alpha - z^{-1}}{1 - \alpha z^{-1}} \quad \text{and} \quad \alpha = -0.9\,. \tag{6}$$

  In this filter the coefficient w controls the high-frequency gain. We will consider this gain to be a time-varying parameter $w(nT)$.

- The high-pass filter $G(s)$ in the feedback control loop is replaced with a digital equivalent [14]:

$$H_3(z) = \frac{U(z)}{E(z)} = \frac{b_0 - b_1 z^{-1}}{1 - a_1 z^{-1}}\,, \tag{7}$$

  with $b_0 = 0.55$, $b_1 = -0.46$, $a_1 = 0.014$;

- The envelope follower $\int$ is replaced by a digital equivalent that rectifies the signal $u(nT)$ according to the nonlinear function [16]:

$$\begin{aligned} v(nT) &= f_1(u(nT), v(nT - T)) \\ &= \{1 - b(nT)\}|u(nT)| + b(nT)v(nT - T) \end{aligned} \tag{8}$$

  with

$$b(nT) = \begin{cases} b_{up} & |u(nT)| > v(nT - T) \\ b_{down} & \text{otherwise} \end{cases} \tag{9}$$

  in which $b_{up} = 0.995$ and $b_{down} = 0.9998$;

- The control on the time-varying shelving filter is realized as a nonlinear map that defines the gain $w(nT)$ as a function of the envelope $v := y_{N1}$ and multiplies w by the output y_{L2} of H_2:

$$f_2(y_{N1}(nT), y_{L2}(nT)) = y_{L2}(nT) \cdot w(y_{N1}(nT)), \tag{10}$$

  with

$$w(y_{N1}) = \begin{cases} a - b(y_{N1})^e & 0 \le y_{N1} \le 3 \\ 1 + c/(1 - dy_{N1}) & \text{otherwise} \end{cases} \tag{11}$$

  This equation was determined empirically by observing that w must have high values for small values of the envelope, while $w \to 1$ (the high-pass filter becomes transparent) for high values of the envelope. Parameters in Eq. (11) were determined by interpolating over two values and by requiring w to be $\mathcal{C}^{(1)}$ at $y_{N1} = 1$. The chosen values are $a = 7.8,\ b = 1.4 \cdot 10^{-14},\ e = 30,\ c = 0.57,\ d = 2.86$.

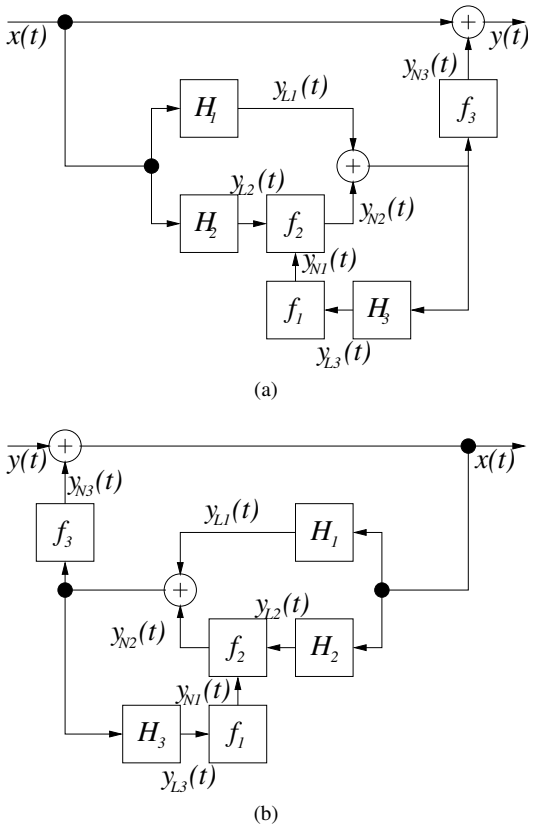

Figure 3: *Digital Dolby B-like system; (a) encoder, (b) decoder.*

- The overshoot characteristics realizes the following nonlinear compression function:

$$f_3(e(nT)) = \tag{12}$$

$$\begin{cases} e(nT)\,, & |e(nT)| \le T_1 \\ T_1 + \frac{T_2-T_1}{T_3-T_1}\{e(nT)-T_1\}\,, & T_1 < |e(nT)| \le T_3 \\ T_2\,, & |e(nT)| > T_3 \end{cases}$$

with $T_1 = 80$, $T_2 = 100$, $T_3 = 200$. We give reason of such parameters by considering that our system is designed to work in the dynamic range $(-10, 30)$ dB. Hence, the amplitude of $e(t)$ is considered acceptable until staying within 40 dB, i.e., absolute value equal to 100, otherwise an overshoot caused by the time-varying filter is detected.

Using the characteristic (6) for the time-varying filter guarantees that the transfer function $1 + H_w(z)$ is minimum-phase for every choice of w. This fact in practice ensures the stability of the decoder, where the output depends on the input according to the relation explained by (3) .

In summary, both the encoder and the decoder comprise three linear blocks H_i and three nonlinear blocks f_i. The update of linear blocks is written in matrix form as

$$\boldsymbol{y}_L[n] = \boldsymbol{B}\boldsymbol{x}_L[n] + \boldsymbol{q}[n], \tag{13}$$

where the vectors $\boldsymbol{x}_L$ and $\boldsymbol{y}_L$ collect inputs and outputs of blocks, $q_i[n] = b_{1,i}x_{Li}[n-1] + a_{1,i}y_{Li}[n-1]$ collect contributions of past components, and $\boldsymbol{B}$ is a diagonal matrix containing the linear coefficients $b_{0,i}$.

The update of nonlinear blocks is written in matrix form as

$$\boldsymbol{y}_N[n] = \boldsymbol{f}\big(\boldsymbol{x}_N[n], \boldsymbol{p}[n]\big), \tag{14}$$

where the vectors $\boldsymbol{x}_N$, $\boldsymbol{y}_N$, collect inputs and outputs of the nonlinear blocks, while p_i contains the contribution of historical components in the functions. According to equations (9,11,12), f_1 has a non-null historical component $p_1[n] = v(nT-T)$, while f_2 and f_3 are algebraic nonlinearities.

The topology of the network and the external inputs to each block are specified by the equation:

$$\left|\begin{array}{c}\boldsymbol{x}_N\\ \boldsymbol{x}_L\end{array}\right| = \boldsymbol{C}\left|\begin{array}{c}\boldsymbol{y}_N\\ \boldsymbol{y}_L\end{array}\right| + \left|\begin{array}{c}\boldsymbol{u}_N\\ \boldsymbol{u}_L\end{array}\right|, \qquad \boldsymbol{C} = \left|\begin{array}{cc}\boldsymbol{C}_{NN} & \boldsymbol{C}_{NL}\\ \boldsymbol{C}_{LN} & \boldsymbol{C}_{LL}\end{array}\right|, \tag{15}$$

where $\boldsymbol{u}_{N,L}$ represent external inputs to the nonlinear and linear blocks, respectively. The topology matrix $\boldsymbol{C}$ specifies connections between block inputs and outputs, and the four sub-matrices $\boldsymbol{C}_{NN,NL,LN,LL}$ account for nonlinear-to-nonlinear, linear-to-nonlinear, nonlinear-to-linear, and linear-to-linear connections, respectively. The encoder and the decoder systems differ only on the matrix $\boldsymbol{C}$ specifying the topology:

$$\boldsymbol{C} = \left|\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & c_{5,3} & 0 & 0 & 0 \\ 0 & 0 & c_{6,3} & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{array}\right|, \tag{16}$$

where $c_{5,3} = c_{6,3} = 0$ for the encoder, and $c_{5,3} = c_{6,3} = -1$ for the decoder. Note that in both cases there is no direct connection between the input and the output of the same block.

In summary, the encoder and the the decoder satisfy all the hypotheses for the applicability of the procedure described in [9]. The digital systems representing the encoder and the decoder are represented in Figure 4(a) and 4(b), respectively.

5. ANALYSIS OF CONVERGENCE

We have shown in previous works [8, 9] that straightforward algebra leads to the following equations for the system inputs:

$$\boldsymbol{x}_L = \boldsymbol{F}_{LL}^{-1}\boldsymbol{C}_{LN}\boldsymbol{f}\big(\boldsymbol{x}_N, \boldsymbol{p}\big) + \boldsymbol{F}_{LL}^{-1}(\boldsymbol{C}_{LL}\boldsymbol{q} + \boldsymbol{u}_L), \tag{17a}$$

$$\boldsymbol{x}_N = \boldsymbol{W}_1\boldsymbol{f}\big(\boldsymbol{x}_N, \boldsymbol{p}\big) + \boldsymbol{W}_2\boldsymbol{q} + \boldsymbol{W}_3\boldsymbol{u}_L + \boldsymbol{u}_N, \tag{17b}$$

where the matrices $\boldsymbol{F}_{LL}$, $\boldsymbol{W}_i$ are defined from $\boldsymbol{B}$ and $\boldsymbol{C}$ (see [9] for details). From equations (14) and (17b), one can write

$$\boldsymbol{y}_N[n] = \boldsymbol{f}\big(\boldsymbol{W}_1\boldsymbol{y}_N[n] + \tilde{\boldsymbol{x}}_N[n], \boldsymbol{p}\big), \tag{18}$$

where $\tilde{\boldsymbol{x}}_N[n] = \boldsymbol{W}_2\boldsymbol{q}[n] + \boldsymbol{W}_3\boldsymbol{u}_L[n] + \boldsymbol{u}_N[n]$ collects the contribution of known quantities to the input $\boldsymbol{x}_N$.

Note that the only unknown in (18) is $\boldsymbol{y}_N[n]$. The presence of delay-free paths in the network causes $\boldsymbol{W}_1$ to be non-null, and consequently Eq. (18) defines $\boldsymbol{y}_N[n]$ implicitly. The network is computable if $\boldsymbol{y}_N[n]$ can be computed from (18). More precisely, the computation can be decomposed into the following steps:

1. $\boldsymbol{x}_N[n]$ and $\boldsymbol{y}_N[n]$ are computed from (17b) and (18) using external inputs $\boldsymbol{u}[n]$ and historical components $\boldsymbol{p}[n]$, $\boldsymbol{q}[n]$;

2. $\boldsymbol{x}_L[n]$ and $\boldsymbol{y}_L[n]$ are computed from (17a) and (13), respectively;
3. $\boldsymbol{p}[n+1]$ and $\boldsymbol{q}[n+1]$ are computed from known variables; in particular $\boldsymbol{q}[n+1]$ can be computed by feeding each filter with a null signal [5]. No computation is needed if the filters are realized in transposed direct form [1, 7].

In [8] we discussed the use of Newton-Raphson (NR) iteration for the solution of step 1. in the computational scheme outlined above. The NR algorithm [17] searches a local zero of the function

$$\boldsymbol{f}\big(\boldsymbol{W}_1\boldsymbol{y}_N + \tilde{\boldsymbol{x}}_N, \boldsymbol{p}\big) - \boldsymbol{y}_N. \tag{19}$$

In [9] we have proposed an approach based on fixed-point (FP) iteration [17]. In this case we try to solve the fixed-point problem $\boldsymbol{y}_N = \boldsymbol{g}_{\boldsymbol{p}}(\boldsymbol{y}_N)$, where the function $\boldsymbol{g}_{\boldsymbol{p}}$ has been defined as

$$\boldsymbol{g}_{\boldsymbol{p}}(\boldsymbol{y}_N) = \boldsymbol{f}\big(\boldsymbol{W}_1\boldsymbol{y}_N + \tilde{\boldsymbol{x}}_N, \boldsymbol{p}\big). \tag{20}$$

FP iteration is preferable over NR iteration in terms of efficiency and ease of implementation. However, convergence of FP iteration is ensured only if the nonlinear function $\boldsymbol{g}_{\boldsymbol{p}}$ given in equation (20) satisfies more restrictive hypothesis. Namely, $\boldsymbol{g}_{\boldsymbol{p}}$ must be a *contraction*, i.e. it possesses a Lipschitz constant $0 \leq L_{g_p} < 1$ such that $\|\boldsymbol{g}_{\boldsymbol{p}}(\boldsymbol{y}) - \boldsymbol{g}_{\boldsymbol{p}}(\boldsymbol{y}^*)\| \leq L_{g_p}\|\boldsymbol{y} - \boldsymbol{y}^*\|$. We have shown [9] that L_{g_p} can be estimated from above as $L_{g_p} \leq L_{f_p}\|\boldsymbol{W}_1\|$, where L_{f_p} is a Lipschitz constant for $\boldsymbol{f}(\cdot, \boldsymbol{p})$. In the remainder of this section we refine this analysis and apply it to the Dolby B. Specifically we show that FP iteration can be applied to the encoder topology but not to the decoder topology.

We restrict our analysis to the second component g_2 of $\boldsymbol{g}_p$, as it is easy to show that this is the critical one. The function g_2 is $\mathcal{C}^{(1)}$, since $g_2(\boldsymbol{y}_N) = f_2(\boldsymbol{W}_1\boldsymbol{y}_N + \tilde{\boldsymbol{x}}_N)$. In order to estimate whether g_2 is a contraction, it suffices to estimate its derivatives $\partial g_2/\partial y_{Ni}$ $(i = 1, 2, 3)$ [9]. If the condition

$$\sup_{\boldsymbol{y}_N \in Y} \left| \frac{\partial g_2}{\partial y_{Ni}} \right| > 1 \tag{21}$$

holds for some i in a given range Y, then g_2 is locally not a contraction in Y and FP iteration is not convergent. The derivatives are computed as

$$\frac{\partial g_2}{\partial y_{Ni}} = \sum_j \frac{\partial f_2}{\partial x_{Nj}} \frac{\partial x_{Nj}}{\partial y_{Ni}} = \sum_j \frac{\partial f_2}{\partial x_{Nj}} [\boldsymbol{W}_1]_{j,i}. \tag{22}$$

Substituting values of the coefficients $[\boldsymbol{W}_1]_{j,i}$ for the encoder and the decoder yelds

$$\frac{\partial g_2}{\partial \boldsymbol{y}_N} = \begin{cases} \left[y_{L1} \cdot \dfrac{\partial w}{\partial v},\ 0,\ 0 \right] & \text{(encoder)} \\ \left[y_{L1} \cdot \dfrac{\partial w}{\partial v},\ 0,\ 0.95 \cdot w(v) \right] & \text{(decoder)} \end{cases} \tag{23}$$

Note in particular that the third component of this derivative is not null for the decoder: this is a consequence of the decoder topology, in which the output from the compressor f_3 influences the input to the time-varying high-frequency gain f_2 through the main feedback loop. Unfortunately $w(v)$ is well above 1 in a neighborhood of the origin, therefore $\partial g_2/\partial y_{N3}$ is locally greater than 1 for the decoder and FP iteration cannot be applied safely. On the contrary, $\boldsymbol{g}_{\boldsymbol{p}}$ is globally a contraction for the encoder, and FP iteration can be applied in this case.

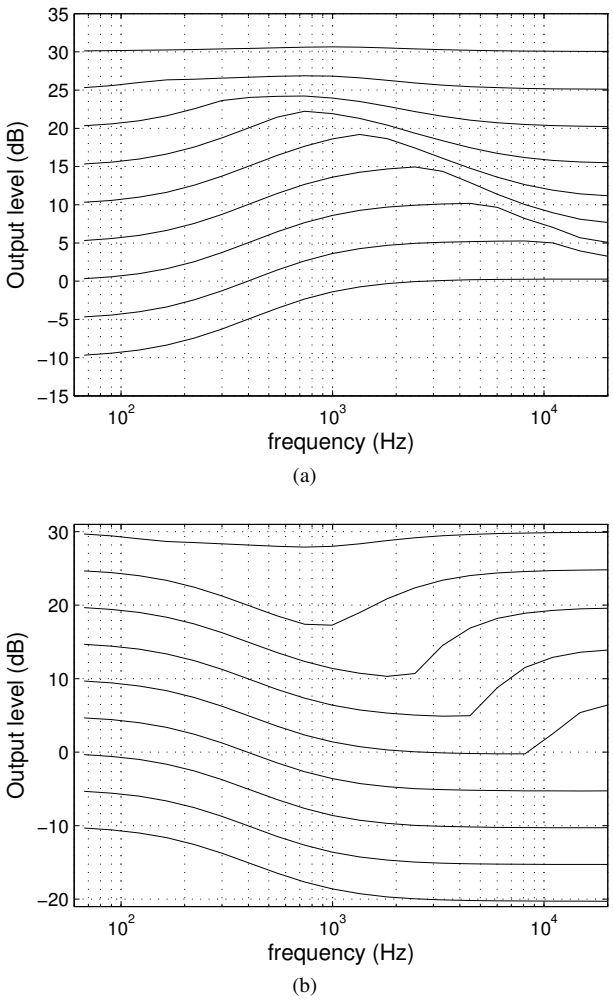

Figure 4: *Digital system response; (a) encoder, (b) decoder.*

6. NUMERICAL SIMULATIONS

The Dolby B-like codec was implemented as a set of Octave/Matlab functions. We first experimentally verified the correctness of the analysis reported in the previous section: numerical simulations show that FP iteration converges when the system topology is the one given in figure 4(a), while this is not the case for the topology of figure 4(b) in accordance to our analytical study on $\boldsymbol{g}_{\boldsymbol{p}}$.

We then tested the response of the encoder and the decoder by feeding both systems with a 9×20 matrix of sinusoidal inputs $x(t)$, containing 9 equally spaced input levels between -10 and 30 dB, and 20 exponentially spaced frequencies between 20 and 20000 Hz. In light of the results presented above, NR iteration was used for the solution of both systems.

For each input sinusoid, the average number of NR iterations per sample was computed. Results show that both the encoder and the decoder require in average ~ 2 NR iterations per sample. The average iteration number, not surprisingly, increases with the frequency of the input sinusoid, with a maximum of 2.25 for the encoder and 2.38 for the decoder. The same number decreases for higher input levels, reaching a minimum of 1.21 for the encoder

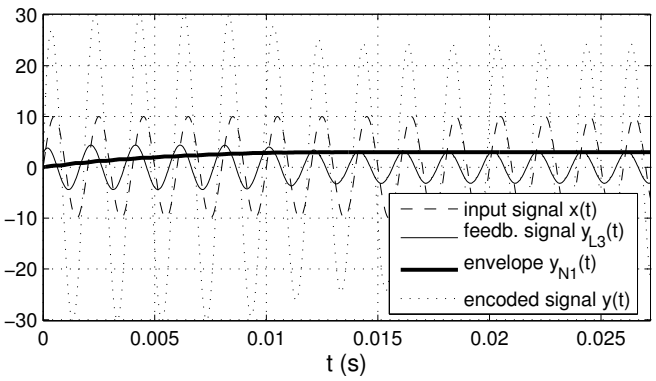

Figure 5: *Example of an encoded sinusoid.*

and 1.28 for the decoder: this is also a predictable effect, since the lower the input level is, the higher is the time-varying gain w.

The responses of the two systems are plotted in figure 5(a) and 5(b), respectively. For each element of the input matrix, the frequency and amplitude of the resulting output sinusoid determined the interpolation point which the responses of figure 4 intersect. Note in particular that for very low input levels the encoder provides a maximum of about 10 dB boost above 4000 Hz, which is qualitatively in accordance with the specification requirements imposed by Dolby Labs [10]. Note also that the systems are transparent to high input signal levels.

Figure 5 provides an example of the encoding process on a test sinusoid at 20 dB and 500 Hz. The encoded sinusoid is boosted, but as the envelope signal y_{N1} rises up the gain of the shelving filter is lowered and the output level is consequently reduced.

Numerical simulations made by cascading the decoder after the encoder show that the decoded sinusoids are exact reconstructions of the input signals. The presence of a very short and small transient encoding/decoding error, lasting less than five samples and invisible in Figure 5, disappears as soon as the initial input discontinuity arising when the null signal turns into a sinusoid is forgotten by the digital codec.

7. CONCLUSIONS

An exact realization of a Dolby B encoding/decoding architecture was made possible by implementing a previously known algebraic procedure that allows to compute nonlinear filter networks containing delay-free paths. The use of this procedure sheds light on some computational aspects of the Dolby B: specifically, the analytical results reported in section 5 provide a quantitative link between the inherent structure of the system and the robustness of the computational procedure.

Ongoing work is focusing on investigating the properties of the computational procedure further, and specifically on the relations between the topology of a generic nonlinear numerical system (17a,17b) and its computational behavior.

8. REFERENCES

[1] S. K. Mitra, *Digital Signal Processing. A computer-Based Approach*. New York: McGraw-Hill, 1998.

[2] A. Härmä, M. Karjalainen, L. Savioja, V. Välimäki, U. K. Laine, and J. Huopaniemi, "Frequency-warped signal processing for audio applications," *J. Audio Eng. Soc.*, vol. 48, no. 11, pp. 1011–1031, Nov. 2000.

[3] S. Bilbao, *Wave and Scattering Methods for the Numerical Integration of Partial Differential Equations*. New York: John Wiley & Sons., 2004.

[4] L. Trautmann and R. Rabenstein, *Digital Sound Synthesis by Physical Modeling Using the Functional Transformation Method*. New York: Kluwer Academic/Plenum Publishers, 2003.

[5] A. Härmä, "Implementation of frequency-warped recursive filters," *EURASIP Signal Processing*, vol. 80, no. 3, pp. 543–548, Mar. 2000.

[6] F. Fontana and M. Karjalainen, "A digital bandpass/bandstop complementary equalization filter with independent tuning characteristics," *IEEE Sig. Proc. Letters*, vol. 10, no. 4, pp. 88–91, Apr. 2003.

[7] F. Fontana, "Computation of linear filter networks containing delay-free loops, with an application to the waveguide mesh," *IEEE Trans. Speech and Audio Proc.*, vol. 11, no. 6, pp. 774–782, Nov. 2003.

[8] F. Fontana, F. Avanzini, and D. Rocchesso, "Computation of nonlinear filter networks containing delay-free paths," in *Proc. Int. Conf. on Digital Audio Effects (DAFx-04),* Naples, Italy, Oct. 2004, pp. 113–118.

[9] F. Avanzini, F. Fontana, and D. Rocchesso, "Efficient computation of nonlinear filter networks with delay-free loops and applications to physically-based sound models," in *Proc. 4th Int. Workshop on Multidimensional Systems (NDS 2005)*, Wuppertal, July 2005, pp. 110–115. [Online]. Available: http://www.dei.unipd.it/~avanzini

[10] Dolby Laboratories Inc., "San Francisco, CA," Retrieved June 29th, 2006, [Online] http://www.dolby.com.

[11] Audiotools, "Noise Reduction Systems," Retrieved June 29th, 2006, [Online] http://audiotools.com/noise.html.

[12] U. Sauvagerd, "Circuit arrangement for influencing the frequency response of a digital audio signal," Oct. 1999, uS Patent US5974156.

[13] F. Fontana and M. Bricchi, "Process for noise reduction, particularly for audio systems, device and computer program product therefore," US Patent US2003004591, Jan. 2003.

[14] M. Bricchi and F. Fontana, "A process for noise reduction, particularly for audio systems, device and computer program product therefore," EU Patent EP1271772, Jan. 2003.

[15] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1989.

[16] P. Dutilleux and U. Zölzer, "Nonlinear processing," in *DAFX – Digital Audio Effects*, U. Zölzer, Ed. New York: J. Wiley & Sons, 2002, pp. 93–136.

[17] J. D. Lambert, *Numerical Methods for Ordinary Differential Systems*. New York: John Wiley & Sons, 1991.