A CLIPPING PREVENTION METHOD FOR ALL-PASS DIGITAL FILTERS WITH TIME-VARYING COEFFICIENTS

Federico Fontana

Dipartimento di Scienze Matematiche, Informatiche e Fisiche
Università degli Studi di Udine
33100 Udine, Italy
federico.fontana@uniud.it

Silvia Pasin

Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano
20133 Milan, Italy
silvia.pasin@mail.polimi.it

Alberto Bernardini

Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano
20133 Milan, Italy
alberto.bernardini@polimi.it

Stefano D'Angelo

Orastron srl
Via Pasquale Voso, 20/D
84043 Agropoli (SA), Italy
stefano.dangelo@orastron.com

ABSTRACT

A clipping prevention method is proposed for first- and second-order all-pass filters with time-varying coefficients. Unlike conventional anti-clipping or declipping approaches, the method operates directly on the coefficient dynamics and does not rely on assumptions about internal energy evolution, by just asking that the input signal is not already clipping. The core idea is to control the deviation between the output of the time-varying filter and that of an equivalent static all-pass structure with constant coefficients. By adaptively limiting this deviation at runtime, the output is constrained below a prescribed clipping threshold (typically unit magnitude). The method is active only during short transients where clipping would occur, after which the coefficients are released to reach their target values. This preserves the integrity of the input signal and the numerical properties of the all-pass filter. Experimental results confirm the expected behavior even in scenarios where energy-preserving all-pass structures exceed the clipping threshold, suggesting the proposed approach as a practical solution for robust dynamic filter implementations with limited additional computational cost, suitable especially for embedded digital audio processing hardware.

1. INTRODUCTION

Dynamic range limitation methods encompass a broad class of systems and algorithms aimed at controlling the magnitude of digital audio signals at runtime [1]. Some approaches model dynamic range limitation from an acoustic perspective, taking into account perceptual and auditory aspects [2]. Others instead focus on digital signal processing (DSP) techniques for directly constraining signal amplitude. The resulting solutions range from traditional filter-based methods [3] to more recent neural network approaches [4, 5].

Among the issues that can significantly degrade audio signals when their magnitude is not properly controlled along the DSP chain, clipping is a critical artifact that requires specific attention. It occurs when an otherwise undistorted discrete-time signal exceeds the amplitude range supported by one or more processing blocks. Its effects on sound quality can be disruptive and have been systematically studied across different types of audio signals [6]. The perceptual impact of clipping depends on the limitation strategy adopted by the processing blocks when handling out-of-range values [7].

Two main approaches are commonly used to avoid or mitigate clipping: prevention and detection. Prevention methods can be applied when a system, in addition to its standard DSP operations, is able to anticipate signal values that would exceed the allowable output range; in such cases, compression techniques can be employed to smoothly limit the signal [8]. Detection methods, instead, operate after clipping has occurred. Laguna and Lerch replace short clipped segments using time-domain spline interpolation, and longer regions using signal components reconstructed from spectral information in neighboring windows [9]. Defraene *et al.* proposed a compressed sensing approach informed by perceptual auditory models [10], while Esquéda *et al.* developed a spectral method targeting aliasing components introduced by clipping [11]. A recent survey shows that these methods generally rely on stochastic and/or energy-based models of audio signals [12], which are used to define and minimize the error between the original and the reconstructed signal after clipping.

A class of methods more suitable for real-time DSP applications exploits the *sparsity* of specific signal properties. This concept has been studied in a broader context by Gaultier *et al.*, together with a family of algorithms built upon it [13]. In this setting, sparsity can be observed in the magnitude–phase relationship of signals: local phase modifications around a clipping region can redistribute spectral components, effectively smoothing peaks below the clipping threshold with minimal perceptual impact. This property has been leveraged by Parker *et al.* [3] and Schlecht *et al.* [14], achieving peak attenuation of up to 5 dB. In both cases, all-pass filters are used to introduce controlled phase shifts in the signal components.

The present work also focuses on all-pass structures, as elements that are capable of spreading an energy peak in particular when it comes out from themselves. When such filters are employed in dynamic contexts, rapid parameter variations in fact may lead to abrupt changes in the coefficients, causing the output signal to exceed the clipping threshold. Rather than detecting clipping after it occurs, the proposed approach checks whether updated coefficient values would lead to clipping and, if so, determines via straightforward algebra the maximum allowable variation that keeps the output below the threshold. This results in a non-heuristic prevention strategy that modifies the temporal evolution of the coefficients while preserving the structural properties of the system. The method can also be interpreted as a controlled slowdown of the system dynamics, effectively spreading coefficient variations over time. All-pass filters are chosen as target structures for two main reasons. First, their well-known versatility enables their use in a wide range of parametric filter designs [15]. Second, the least possible number of coefficient values (that is, N values for an Nth-order structure) allows clipping prevention in first- and second-order sections to be formulated in a tractable algebraic form. Moreover, formulating the method for both first- and second-order sections enables its application to all-pass filters of arbitrary order via standard decomposition into cascaded sections [16].

Section 2 presents the proposed method by extending recent preliminary work [17], and details the resulting procedures for both first- and second-order cases. Section 3 then investigates two scenarios for each filter order, showing how the method prevents the output from clipping above threshold under abrupt coefficient variations, in the presence of an input signal lying within the same threshold. Finally, Section 4 discusses the results, including considerations on computational cost and numerical precision, and Section 5 concludes the paper.

2. METHOD

A first-order all-pass filter with variable coefficient c at time step n realizes the following recurrence equation, producing the output y for an input x:

$$y[n] = c(n)\big\{x[n] - y[n-1]\big\} + x[n-1], \tag{1}$$

in which we denote discrete-time dependence in signals using square brackets, and in filter coefficients using round brackets.

Let M denote a magnitude threshold that signals must not trespass, so that $|x[n]| \leq M$ for all n, and assume that no clipping occurred at the previous step for the output, i.e., $|y[n-1]| \leq M$. The goal is to ensure that no clipping occurs at the current step, meaning $|y[n]| \leq M$, even in the presence of an unsuitable value of the coefficient $c(n)$.

Clipping at step n can be calculated directly from (1). If clipping occurs, it can be determined whether it is caused by the coefficient update by considering the output $w[n]$ of a *static* counterpart, i.e., an identical all-pass structure with constant coefficient:

$$w[n] = c(n-1)\big\{x[n] - y[n-1]\big\} + x[n-1]. \tag{2}$$

If $w[n]$ does not clip, the contribution of the coefficient variation to the clipping can be expressed through the signal difference $\tau[n] = y[n] - w[n]$, which depends on the coefficient increment $\Delta c(n) = c(n) - c(n-1)$. Subtracting (2) from (1) yields

$$\tau[n] = \Delta c(n)\big\{x[n] - y[n-1]\big\}, \tag{3}$$

from which the non-clipping condition $|y[n]| = |w[n] + \tau[n]| \leq M$ can be expressed in terms of $\Delta c(n)$ as

$$\big|w[n] + \Delta c(n)\big\{x[n] - y[n-1]\big\}\big| \leq M. \tag{4}$$

Algorithm 1 First-order clipping prevention procedure

1: compute $w[n]$ from (2)
2: $\Delta c(n) \leftarrow c(n) - c(n-1)$
3: $d \leftarrow x[n] - y[n-1]$
4: **if** $d == 0$ **then**
5: skip to line 15
6: **end if**
7: evaluate the sign of d
8: **if** $d > 0$ **then**
9: compute $\widehat{\Delta c}(n)$ from (6)
10: **end if**
11: **if** $d < 0$ **then**
12: compute $\widehat{\Delta c}(n)$ from (7)
13: **end if**
14: $c(n) \leftarrow c(n-1) + \widehat{\Delta c}(n)$
15: compute $y[n]$ from (1)

Solving (4) for $\Delta c(n)$ yields lower and upper bounds, whose order depends on the sign of $x[n] - y[n-1]$:

$$-\frac{M + w[n]}{x[n] - y[n-1]} \lesseqqgtr \Delta c(n) \lesseqqgtr \frac{M - w[n]}{x[n] - y[n-1]}, \tag{5}$$

in which the inequality symbol $\leq$ holds if $x[n] - y[n-1] > 0$, and reverses otherwise. The coefficient increment $\Delta c(n)$ is then constrained within these bounds, yielding the modified coefficient $\widehat{c}(n) = c(n-1) + \widehat{\Delta c}(n)$, where

$$\begin{aligned} &\widehat{\Delta c}(n) = \\ &\min\Big\{\max\Big\{-\frac{M + w[n]}{x[n] - y[n-1]}, \Delta c(n)\Big\}, \frac{M - w[n]}{x[n] - y[n-1]}\Big\} \\ &\text{if } \; x[n] - y[n-1] > 0 \end{aligned} \tag{6}$$

or

$$\begin{aligned} &\widehat{\Delta c}(n) = \\ &\min\Big\{\max\Big\{\frac{M - w[n]}{x[n] - y[n-1]}, \Delta c(n)\Big\}, -\frac{M + w[n]}{x[n] - y[n-1]}\Big\} \\ &\text{if } \; x[n] - y[n-1] < 0 \; . \end{aligned} \tag{7}$$

A procedure can be defined based on the previous expressions to prevent clipping in the filter. Rather than activating it only when clipping is detected—by evaluating (1) before computing the output—we adopt a *first-order clipping prevention procedure* that operates at every time step. This choice ensures constant processing time and is particularly suitable for DSP architectures—typically embedded systems—where clipping may represent a critical issue.

The procedure is summarized as in Algorithm 1. Note that in the special case $x[n] = y[n-1]$ the procedure does not need to be called, since from (3) it descends $\tau[n] = 0$. In this case at time step n the all-pass block reduces to a unit delay, identically to what would happen if $c(n) = 0$, and consequently the output $y[n] = x[n-1]$ will not clip during this step holding the initial hypothesis $|x| \leq M$.

2.1. Second-Order Structure

A similar method can be formulated to prevent clipping in a second-order all-pass filter with time-varying coefficients c_0 and c_1. Starting from the input–output relationship

$$\begin{aligned} y[n] \quad = \quad & c_0(n)\big\{x[n]-y[n-2]\big\} \\ & + c_1(n)\big\{x[n-1]-y[n-1]\big\} + x[n-2], \end{aligned} \tag{8}$$

the same derivation as in the first-order case can be carried out. With a natural definition of coefficient increments $\Delta c_i(n) = c_i(n) - c_i(n-1)$, $i = 0, 1$, the following condition is obtained:

$$\begin{aligned} &\big|w[n] + \Delta c_0(n)\big\{x[n]-y[n-2]\big\} \\ &\quad + \Delta c_1(n)\big\{x[n-1]-y[n-1]\big\}\big| \leq M \end{aligned} \tag{9}$$

in which, similarly to the first-order case, $w[n]$ denotes the output of a static counterpart of the second-order structure:

$$\begin{aligned} w[n] \quad = \quad & c_0(n-1)\big\{x[n]-y[n-2]\big\} \\ & + c_1(n-1)\big\{x[n-1]-y[n-1]\big\} + x[n-2]. \end{aligned} \tag{10}$$

Clipping may occur due to changes in either of the all-pass coefficients. In this case, the second-order formulation does not introduce additional complications as (9) structurally reduces to (4) if $\Delta c_0(n) = 0$ or $\Delta c_1(n) = 0$.

However, clipping can also arise from simultaneous variations of both coefficients [17]. This situation, for instance, occurs when the second-order all-pass filter is embedded in a peaking or notch parametric equalizer structure [15]. Since (9) does not yield a unique solution pair $(\widehat{\Delta c_0}, \widehat{\Delta c_1})$, it is convenient—especially in DSP applications where efficiency is a key constraint—to distribute it according to a "clipping prevention budget" that can be computed easily. A simple approach proposed here is to determine $\widehat{\Delta c_0}$ and $\widehat{\Delta c_1}$ such that each term contributes at most to half of the bound, i.e,

$$\begin{array}{rcl} \big|w[n]/2 + \Delta c_0(n)\big\{x[n]-y[n-2]\big\}\big| & \leq & M/2 \\ \big|w[n]/2 + \Delta c_1(n)\big\{x[n-1]-y[n-1]\big\}\big| & \leq & M/2 \end{array}, \tag{11}$$

which, due to the known triangular inequality $|u+v| \leq |u| + |v|$, implies (9). From this, by applying the same derivation we did in the first-order case using the terms in (11), we obtain

$$\begin{aligned} &\widehat{\Delta c_0}(n) = \\ &\min\Big\{\max\big\{-\frac{1}{2}\frac{M+w[n]}{x[n]-y[n-2]}, \Delta c_0(n)\big\}, \\ &\frac{1}{2}\frac{M-w[n]}{x[n]-y[n-2]}\Big\} \text{ if } x[n]-y[n-2] > 0 \end{aligned} \tag{12}$$

or

$$\begin{aligned} &\widehat{\Delta c_0}(n) = \\ &\min\Big\{\max\big\{\frac{1}{2}\frac{M-w[n]}{x[n]-y[n-2]}, \Delta c_0(n)\big\}, \\ &-\frac{1}{2}\frac{M+w[n]}{x[n]-y[n-2]}\Big\} \text{ if } x[n]-y[n-2] < 0 \ , \end{aligned} \tag{13}$$

and

$$\begin{aligned} &\widehat{\Delta c_1}(n) = \\ &\min\Big\{\max\big\{-\frac{1}{2}\frac{M+w[n]}{x[n-1]-y[n-1]}, \Delta c_1(n)\big\}, \\ &\frac{1}{2}\frac{M-w[n]}{x[n-1]-y[n-1]}\Big\} \text{ if } x[n-1]-y[n-1] > 0 \ . \end{aligned} \tag{14}$$

Algorithm 2 Second-order clipping prevention procedure

1: compute $w[n]$ from (10)
2: $\Delta c_0(n) \leftarrow c_0(n) - c_0(n-1)$
3: $\Delta c_1(n) \leftarrow c_1(n) - c_1(n-1)$
4: $d_0 \leftarrow x[n] - y[n-2]$
5: $d_1 \leftarrow x[n-1] - y[n-1]$
6: **if** $(d_0 == 0)$ & & $(\mathrm{d}_1 == 0)$ **then**
7:     skip to line 29
8: **end if**
9: **if** $d_0 \,!\!= 0$ **then**
10:     evaluate the sign of d_0
11:     **if** $d_0 > 0$ **then**
12:         compute $\widehat{\Delta c_0}(n)$ from (12)
13:     **end if**
14:     **if** $d_0 < 0$ **then**
15:         compute $\widehat{\Delta c_0}(n)$ from (13)
16:     **end if**
17: **end if**
18: **if** $d_1 \,!\!= 0$ **then**
19:     evaluate the sign of d_1
20:     **if** $d_1 > 0$ **then**
21:         compute $\widehat{\Delta c_1}(n)$ from (14)
22:     **end if**
23:     **if** $d_1 < 0$ **then**
24:         compute $\widehat{\Delta c_1}(n)$ from (15)
25:     **end if**
26: **end if**
27: $c_0(n) \leftarrow c_0(n-1) + \widehat{\Delta c_0}(n)$
28: $c_1(n) \leftarrow c_1(n-1) + \widehat{\Delta c_1}(n)$
29: compute $y[n]$ from (8)

or

$$\begin{aligned} &\widehat{\Delta c_1}(n) = \\ &\min\Big\{\max\big\{\frac{1}{2}\frac{M-w[n]}{x[n-1]-y[n-1]}, \Delta c_1(n)\big\}, \\ &-\frac{1}{2}\frac{M+w[n]}{x[n-1]-y[n-1]}\Big\} \text{ if } x[n-1]-y[n-1] < 0 \ . \end{aligned} \tag{15}$$

The resulting *second-order clipping prevention procedure* is structurally analogous to the first-order case, however making use of $w[n]/2$ and $M/2$ while computing $\widehat{\Delta c_0}(n)$ and $\widehat{\Delta c_1}(n)$ respectively from (12)-(13) and (14)-(15), as well as conditioning the execution on $x[n] \neq y[n-2]$ and $x[n-1] \neq y[n-1]$, respectively. At each time step the procedure implements Algorithm 2. As in the first-order case, if both the previous conditions are not checked then the procedure does not need to be called: the all-pass block in this case corresponds to the cascade of two delay units, hence $|y[n] = x[n-2]| \leq M$ by hypothesis.

3. CASE STUDY

Fig. 1 compares the output of a standard all-pass filter without clipping protection to that modified by implementing the proposed prevention procedure, where, for simplicity, M is set to 1. A Wave Digital Filter (WDF)-based energy-preserving all-pass structure is introduced for comparison. Although not strictly anti-clipping, this structure is a reference candidate since it is specifically designed to avoid instability during runtime variations of the coeffi-

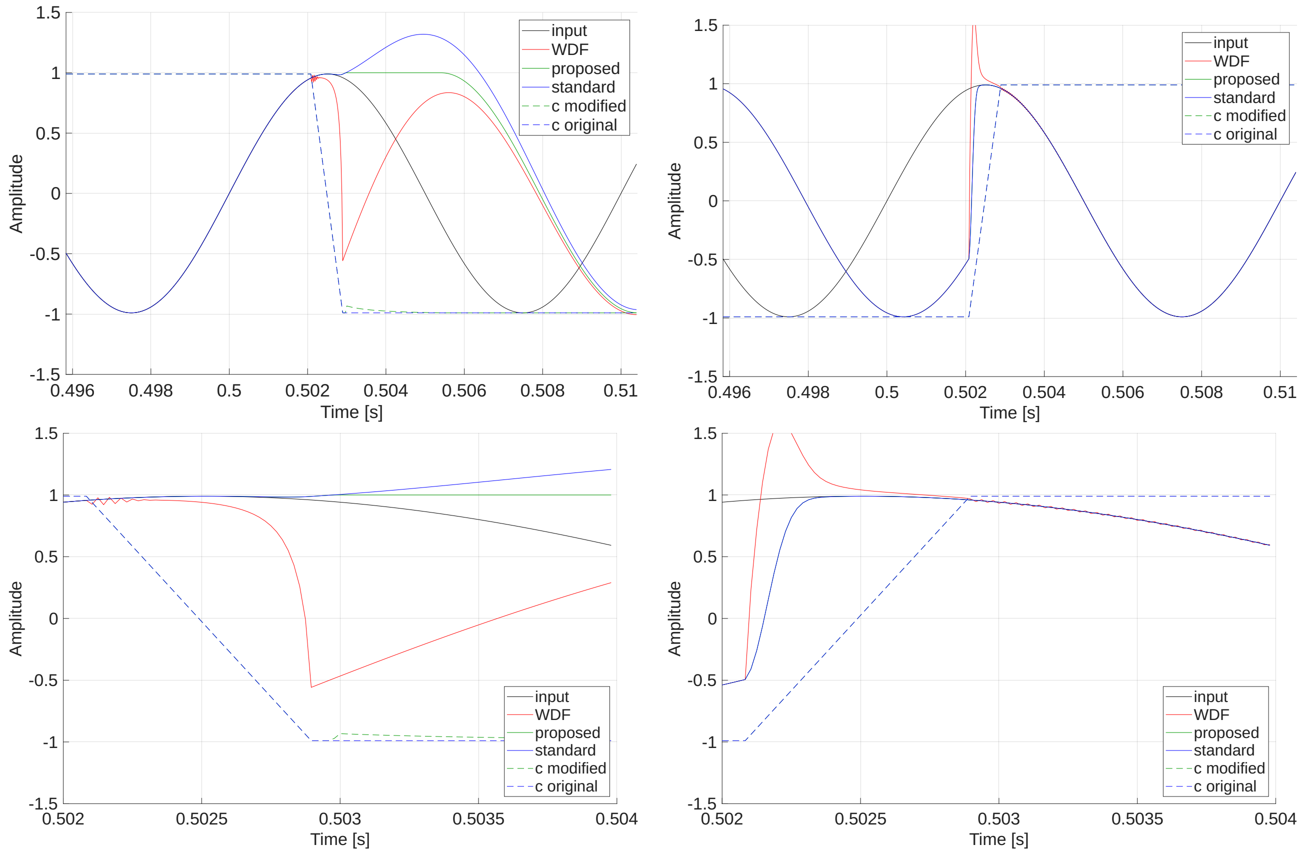

Figure 1: Plots (above) and x-axis zoomed versions (below) of the response to a 100 Hz sinusoidal input at 48 kHz with digital amplitude .99 (black line) of an all-pass block realized with: no protection from clipping (blue line); energy-preserving structure (red line); the proposed clipping prevention (green line). All-pass coefficient value changing between $+.99$ and $-.99$ (left) and the other way round (right). The coefficient values without (blue color) and with (green color) clipping prevention are plotted in dashed line. Clipping magnitude threshold M set to 1.

cients. Equality between the input and output energy is obtained through the following relationship [18]:

$$y[n] = c(n)x[n] + \sqrt{\frac{1-c^2(n)}{1-c^2(n-1)}}\{x[n-1] - c(n-1)y[n-1]\}. \tag{16}$$

All structures operate at $F_s = 48$ kHz and process a 100 Hz sinusoidal input with amplitude 0.99. Around $t \approx 0.5$ s, corresponding to a positive peak of the input signal, the all-pass coefficient c is linearly varied from $+0.99$ to -0.99 (left plot) or the other way round (right plot) over approximately 1 ms.

It can be observed that the former variation causes the standard structure to clip. In contrast, when the prevention procedure is applied, clipping is avoided through a suitable adaptation of the coefficient over time (dashed lines). The energy-preserving structure follows a different dynamic behavior, and does not clip in this case.

Conversely, if the coefficient is varied in the opposite direction then the energy-preserving structure does clip, while the standard realization does not. In this situation the prevention procedure is not activated, as indicated by the trajectory of the corresponding all-pass coefficient.

3.1. Second-Order Structure

Fig. 2 compares the output of a second-order all-pass block without clipping protection to that of the same block implementing the prevention procedure, with M again set to 1. As in the previous case, the WDF-based all-pass filter is also tested using the second-order structure in [18], whose port coefficients $M_0(n)$, $M_1(n)$, and $M_2(n)$ are derived from $c_0(n)$ and $c_1(n)$ as

$$M_{1,2}(n) = \frac{M_0(n)}{2}\,\frac{1+c_0(n) \pm c_1(n)}{1-c_0(n)}, \tag{17}$$

where the coefficient M_0, associated with the free port, can be chosen arbitrarily.

The same setup used for Fig. 1 is also considered in this case. The all-pass coefficients c_0 and c_1 are linearly varied from 0.98 to -0.98 and from -1.96 to 0.01 respectively (left plot), or in the opposite direction (right plot). Some similarities with the first-order case can be observed, particularly with the former variation of the coefficients. However, the latter causes the standard structure to clip. The resulting artifact requires a relatively large, impulsive correction of the filter coefficients in order to prevent it from clipping. Furthermore, this correction is beneficial for smoothing the subsequent signal values around the expected output.

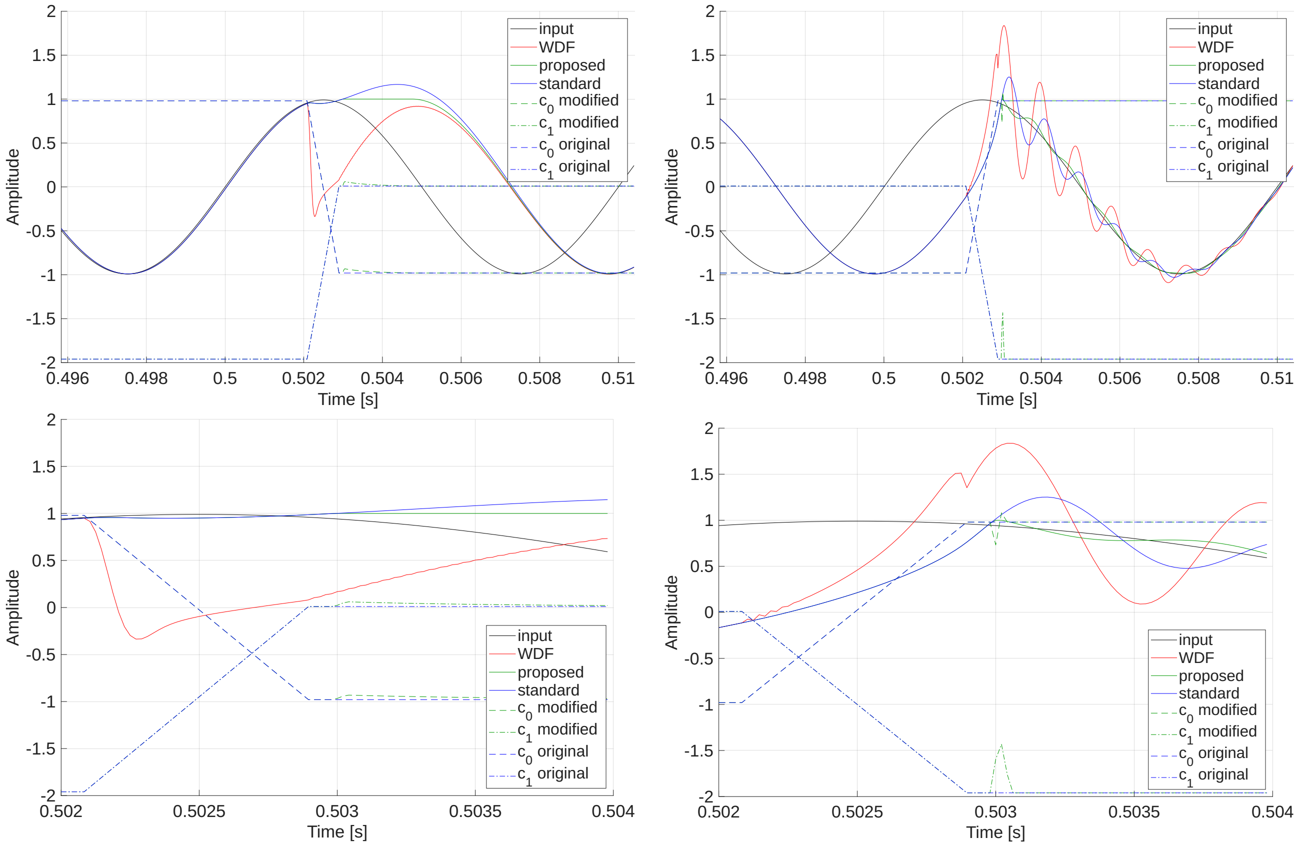

Figure 2: Plots (above) and x-axis zoomed versions (below) of the response to a 100 Hz sinusoidal input at 48 kHz with amplitude .99 (black line) of a second-order all-pass block realized with: no protection from clipping (blue line); energy-preserving structure (red line); the proposed clipping prevention (green line). All-pass coefficient values (c_0, c_1) changing between $([.98, -.98], [-1.96, .01])$ (left plot) and the other way round (right plot). The coefficient values without (blue color) and with (green color) clipping prevention are plotted in dashed (c_0) and dash-dotted (c_1) line. Clipping magnitude threshold M set to 1.

The second-order case has been studied also using a music excerpt as input, setting a clipping threshold $M = .99$. Fig. 3 (left) shows the effect of changing the filter coefficients between $([-.98, .98], [.01, -1.96],)$ as in Fig. 2 (right) starting at $t \approx 6.5$ s. After clipping prevention, the output of the proposed structure is generally closer to that of the energy-preserving filter. The latter clips again at the very end of the observation window, for $t \approx 6.534$ s, where the proposed algorithm prevents from that instead as shown by the little glitch modifying the trajectory of c_1. At the same time, the output from the standard structure follows a slightly different trajectory and does not end up beyond the clipping threshold. Fig. 3 (right) shows corresponding tracks in the Audacity sound editor, after saving the respective four signals (input, WDF, proposed, standard in top-down order) as WAV files: a careful inspection around $t \approx 6.5$ s confirms that the output of the proposed structure does not clip, as opposed to the other two outputs.

4. DISCUSSION

By construction, the proposed procedure temporarily steers the all-pass coefficients away from the trajectory dictated by the system control at runtime. This intervention is typically limited to a short transient (on the order of a few milliseconds) and, in many cases, introduces only minor deviations from the intended coefficient values. A closer inspection of the coefficient trajectories in Figs. 1 and 2 (dashed lines) reveals that the main clipping event in the unprotected filter can occur as soon as, or even shortly *after* the coefficients have stabilized to the target values. This fact highlights an important aspect: clipping does not necessarily occur instantaneously during the coefficients variation, but may also be part of the transient following the update. As a consequence, the proposed procedure does not simply act as a mechanism to slow down coefficient variations. Instead, it enforces coefficient settings that prevent clipping whenever it would occur, regardless of whether parameter updates are ongoing—recall e.g. what happens in the music excerpt example at $t \approx 6.534$ s.

The proposed procedure can be safely applied to all-pass structures in Direct Form I [16]. In fact, these structures implement the filter state by explicitly using past input and output values. The latter are guaranteed by the procedure itself not to be affected by clipping errors. Conversely, the realization of linear recombinations of the all-pass input–output relationships (1) and (8), leading to alternative forms, may cause internal clipping since the algorithm in its current implementation does not explicitly operate on the filter state. An extensive investigation on how the proposed

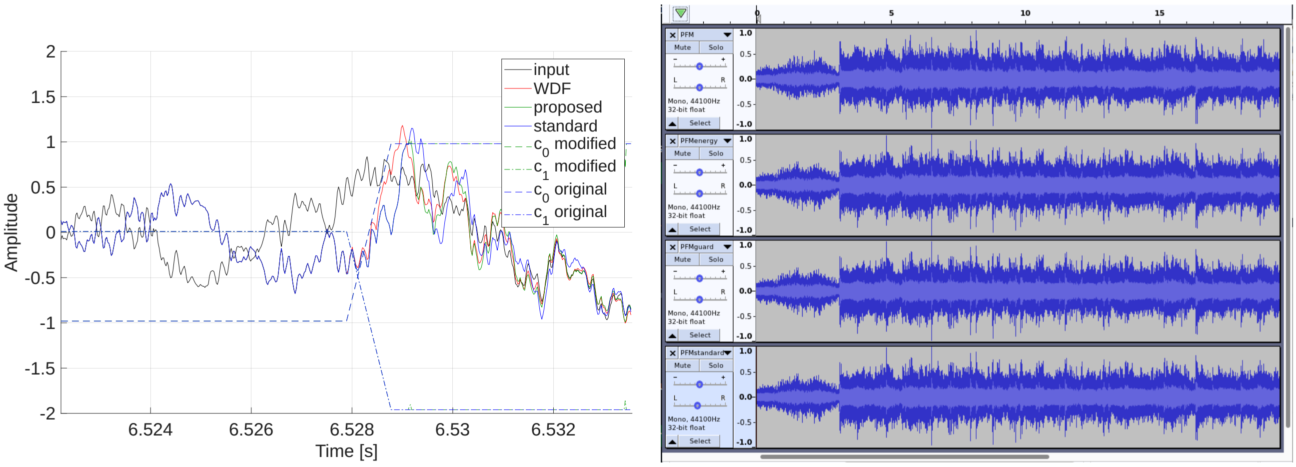

Figure 3: Left: Response to a music excerpt at 44.1 kHz (black line) of a second-order all-pass block realized with: no protection from clipping (blue line); energy-preserving structure (red line); the proposed clipping prevention (green line). All-pass coefficient values (c_0, c_1) changing between $([-.98, .98], [.01, -1.96],)$. The coefficient values before (blue color) and after (green color) clipping prevention are plotted in dashed (c_0) and dash-dotted (c_1) line. Clipping magnitude threshold M is set to .99. Right: the same responses shown as tracks in the Audacity sound editor after saving them as WAV files.

algorithm may be adapted to all-pass structures in their various forms [16], and the consequent computational cost assessment, is left for future research.

Provided that the original sequence of coefficients $c(n)$ was not intended to lead to an unstable first-order all-pass filter, i.e., $|c(n)| < 1$ for all n, one may wonder whether the modified sequence $\widehat{c}(n)$ still leads to a stable structure. Holding $|x[n]| \leq M$ for all n then the answer is positive in the sense of BIBO stability [16]. In fact, (4) ensures that the response to x is y such that $|y[n]| \leq M$ for all n. This implies BIBO stability with the modified sequence. The same happens by conditioning with (9) in the second-order case.

Concerning numerical behavior on architectures working with different precision, consider for simplicity the case $M = 1$, which is common in embedded DSP boards. In this setting, all quantities involved in the procedure either remain within, or can be constrained to, a magnitude below 2. The latter situation may arise when evaluating (6)-(7), (12)-(13), or (14)-(15), particularly under extreme coefficient transitions between opposite boundary values within a single time step. In particular, the effect on the output of the coefficient increment (or increments, in the second-order case) Δc is proportional to its/their multiplying factor(s), see (4) and (11). On DSP boards where divide operations can be inaccurate or are not even part of the instructions set, thus preventing from computing (6)-(7) or (12)-(15) respectively in the first- or second-order case, a simple algorithm can be implemented: if (4) or (9) signal that clipping would occur using the current all-pass coefficient(s) respectively in the first- or second-order case, then i) determine the sign of the multiplying factor(s) and ii) search $\widehat{\Delta c}$ by iteratively updating its/their value(s) across a (e.g. bisection) search of the null value of a function $g(n)$, obtained by properly rearranging (4) or (11) depending on the order of the all-pass. For instance in the first-order case with positive-signed factor $x[n] - y[n-1]$ and difference $M - w[n]$ to match:

$$g^{(i)}(n) = \{M - w[n]\} - \widehat{\Delta c^{(i)}}(n)\{x[n] - y[n-1]\}, \quad (18)$$

in which the apex $^{(i)}$ labels the ith iteration of the zero search depending on the numerical scheme elected for the search.

On the other hand, if a computationally more powerful architecture is available then the simple heuristics (11) leading to an even split of the "clipping prevention budget" may be substituted by a more accurate budget distribution. In this regard, the three components $w[n]$, $\Delta c_0(n)\{x[n] - y[n-2]\}$ and $\Delta c_1(n)\{x[n-1] - y[n-1]\}$ can be assessed separately, and more elaborate decisions regarding correcting $\Delta c_0(n)$ and $\Delta c_1(n)$ can be made in view of the M boundary. Peculiarly, in our 2nd-order case study the two latter components are largely symmetric (see Fig. 4), hence they have almost identical magnitude before the clipping prevention starts to operate: this symmetry likely occurs because both x and y are sinusoidal, and both coefficients c_0 and c_1 change with the same rate until the algorithm takes control over them. So, the specific case study speaks in favor of (11) as a well-balanced compromise between accuracy and computational efficiency.

It is also of interest to assess the computational overhead introduced by the clipping prevention procedure. Table 1 reports average processing times (in milliseconds) over 10 runs on a 4.7 GHz Intel Core i7 laptop running Ubuntu Linux of the standard (Direct Form I), energy-preserving, and proposed structures. Measurements were made on MATLAB for a one-second long input using the functions `tic` and `toc`, and on C++ implementations compiled with `gcc -O3 -ffast-math` for a ten-second long input. Especially in the compiled application the proposed method does not introduce a significant increase in computational cost, remaining proportional to that of the standard and energy-preserving structures per computational time factors amounting to less than four.

5. CONCLUSIONS

An algebraic method has been developed to prevent clipping in first- and second-order all-pass structures. Owing to its procedural simplicity, low computational cost, and modest arithmetic precision requirements, it is well suited for embedded audio DSP archi-

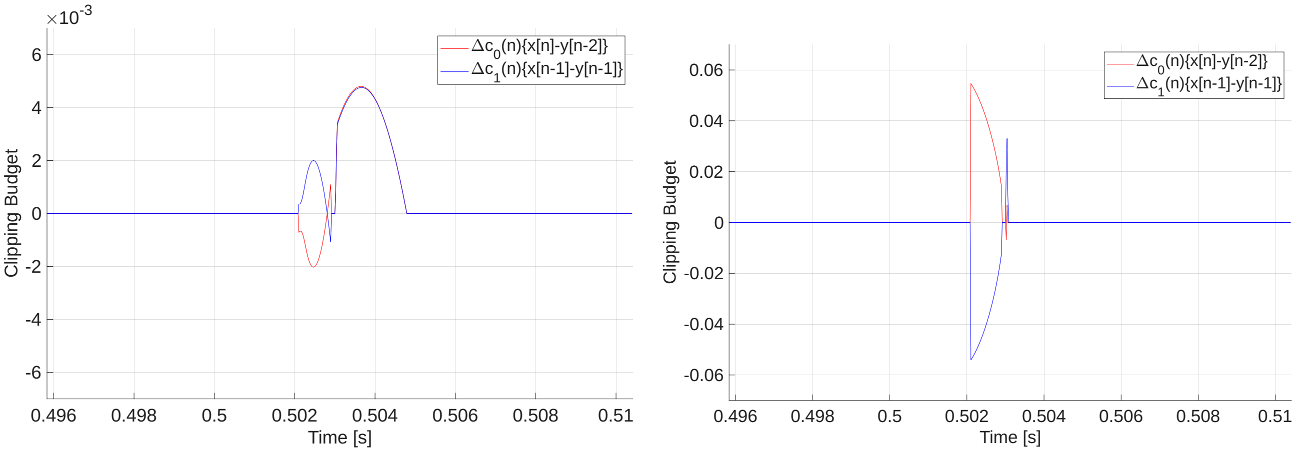

Figure 4: Impact of the components $\Delta c_0(n)\{x[n] - y[n-2]\}$ (red line) and $\Delta c_1(n)\{x[n-1] - y[n-1]\}$ (blue line) responsible of clipping in the 2nd-order case study (Fig. 2). All-pass coefficient values (c_0, c_1) changing between $([-1.96, .01], [.98, -.98])$ (left plot) and the other way round (right plot).

Table 1: Average processing times (ms) of audio inputs at 48 kHz on MATLAB (1s-long input) and compiled C++ (10s-long input) environments

environment	MATLAB		C++	
input size @48 kHz	1 s		10 s	
all-pass order	1st	2nd	1st	2nd
	processing time (ms)			
standard	9.2	9.5	1.1	1.5
energy-preserving	10.2	40.2	1.7	2.2
proposed	14.7	33.7	3.8	4.9

tectures, where real-time systems such as digital audio effects may be exposed to clipping due to rapid parameter variations. In this vein, future work may extend the proposed approach to energy-preserving structures. Due to the availability of various realizations for (16), each meeting specific system requirements [19], the joint adoption of clipping prevention and energy preservation may lead to a class of all-pass filters provided with excellent numerical behavior and stability.

6. REFERENCES

[1] G. W. McNally, "Dynamic range control of digital audio signals," *J. of the Audio Engineering Society*, vol. 32, no. 5, pp. 316–327, 1984.

[2] D. Giannoulis, M. Massberg, and J. D. Reiss, "Digital dynamic range compressor design—a tutorial and analysis," *J. of the Audio Engineering Society*, vol. 60, no. 6, pp. 399–408, 2012.

[3] J. Parker and V. Välimäki, "Linear dynamic range reduction of musical audio using an allpass filter chain," *IEEE Signal Processing Letters*, vol. 20, no. 7, pp. 669–672, 2013.

[4] S. I. Mimilakis, K. Drossos, T. Virtanen, and G. Schuller, "Deep neural networks for dynamic range compression in mastering applications," in *Audio Engineering Society Convention*, vol. 140, 2016.

[5] A. Wright and V. Valimaki, "Grey-box modelling of dynamic range compression," in *Proc. of the 25th International Conference on Digital Audio Effects (DAFx20in22)*, 2022, pp. 304–311.

[6] A. Prodeus and M. Didkovska, "Audio signals clipping detection using kurtosis and its transforms." *Int. J. Comput.*, vol. 19, no. 3, pp. 411–417, 2020.

[7] B. Defraene, T. van Waterschoot, H. J. Ferreau, M. Diehl, and M. Moonen, "Real-time perception-based clipping of audio signals using convex optimization," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 20, no. 10, pp. 2657–2671, 2012.

[8] P. Hämäläinen, "Smoothing of the control signal without clipped output in digital peak limiters," in *Proc. of the 5th International Conference on Digital Audio Effects (DAFx-02)*, Hamburg, Germany, Sep. 26–28, 2002, pp. 195–198.

[9] C. Laguna and A. Lerch, "An efficient algorithm for clipping detection and declipping audio," in *AES 141st Convention*, vol. 2, Los Angeles, CA, Sep. 29–Oct. 1, 2016.

[10] B. Defraene, N. Mansour, S. De Hertogh, T. van Waterschoot, M. Diehl, and M. Moonen, "Declipping of audio signals using perceptual compressed sensing," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 21, no. 12, pp. 2627–2637, 2013.

[11] F. Esqueda, S. Bilbao, and V. Välimäki, "Aliasing reduction in clipped signals," *IEEE Trans. on Signal Processing*, vol. 64, no. 20, pp. 5255–5267, 2016.

[12] P. Záviška, P. Rajmic, A. Ozerov, and L. Rencker, "A survey and an extensive evaluation of popular audio declipping methods," *IEEE J. of Selected Topics in Signal Processing*, vol. 15, no. 1, pp. 5–24, 2021.

[13] C. Gaultier, S. Kitić, R. Gribonval, and N. Bertin, "Sparsity-based audio declipping methods: Selected overview, new algorithms, and large-scale evaluation," *IEEE/ACM Trans. on*

Audio, Speech, and Language Processing, vol. 29, pp. 1174–1187, 2021.

[14] S. J. Schlecht, L. Fierro, V. Välimäki, and J. Backman, "Audio peak reduction using a synced allpass filter," in *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2022)*, 2022, pp. 1006–1010.

[15] P. Regalia, S. Mitra, and P. Vaidyanathan, "The digital all-pass filter: a versatile signal processing building block," *Proc. of the IEEE*, vol. 76, no. 1, pp. 19–37, 1988.

[16] S. K. Mitra, *Digital signal processing: a computer-based approach*. McGraw-Hill Higher Education, 2001.

[17] S. Pasin, "Clipping prevention in time-varying parametric equalizers based on first-order allpass filters," Master's thesis, Politecnico di Milano, Milano, Italy, 2025. [Online]. Available: https://www.politesi.polimi.it/handle/10589/240671

[18] S. Bilbao, "Time-varying generalizations of all-pass filters," *IEEE Signal Processing Letters*, vol. 12, no. 5, pp. 376–379, 2005.

[19] K. J. Werner, "Energy-preserving time-varying Schroeder allpass filters," in *Proc. of the 23th International Conference on Digital Audio Effects (DAFx2020)*, Vienna, Austria, Sep. 2020-21, pp. 242–249.