$\{56, 166, 56\}$, whereas for the splitting of Example 2, a bank with filter lengths $\{L_0, L_1, L_2, L_3\} = \{163, 163, 163, 82\}$ was used. The maximum relative absolute reconstruction error when $x(n)$ is a rectangle signal was equal to $8.9E-04$ and $2.3E-03$ for Examples 1 and 2, respectively. The errors obtained with the IIR implementations are $2.1E-03$ and $2.2E-03$ for Examples 1 and 2, respectively. As to the complexity of the FIR analysis bank, if L is the length of the input signal, the mth subband samples are obtained with $L_m L/M_m$ mults. The cost to implement the analysis bank of Example 1 measured as multiplications per unit (MPU), i.e., the total cost divided by L, is equal to 55.6 MPU and 46 MPU in the FIR and IIR case, respectively, whereas the cost relative to Example 2 is 81.6 MPU and 64 MPU in the FIR and IIR case, respectively. Therefore, we have obtained some improvement in the computational cost at the expence of using a noncausal implementation. Additional computational saving, both for the FIR and IIR case, can be achieved by using the fact that the prototypes in different branches are the same. A third example is shown in [8] relative to a FIR bank with splitting $\{1/2, 1/4, 3/16, 1/16\}$ and implemented with 169.9 MPU. The same splitting, with comparable reconstruction characteristics, has been implemented with an IIR bank with orders $\mathcal{Q} = \{5, 4, 1, 1\}$ and a cost equal to 126 MPU (details of this example are not reported here for brevity's sake).

V. Conclusions

In this correspondence, the problem of designing pseudo-QMF nonuniform banks based on IIR filters has been addressed. The banks are achieved by means of the cosine modulation of different IIR prototypes, extending some results presented in [8] for the FIR case. Drawbacks of using IIR filters have also been pointed out: First, since linear-phase prototypes are needed, a noncausal implementation is necessary; second, the polyphase implementation of the filters imposes heavy constraints on the prototypes transfer functions. Nevertheless, some computational saving with respect to the FIR case [8] can be achieved.

References

[1] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[2] K. Nayebi, T. P. Barnwell, III, and M. J. T. Smith, "Nonuniform filter banks: A reconstruction and design theory," *IEEE Trans. Signal Processing*, vol. 41, pp. 1114–1127, Mar. 1993.

[3] J. Kovacevic and M. Vetterli, "Perfect reconstruction filter banks with rational sampling factors," *IEEE Trans. Signal Processing*, vol. 41, pp. 2047–2066, June 1993.

[4] B. Liu and L. T. Bruton, "The design of N-band nonuniform-band maximally decimated filter banks," in *Proc. 27th Asilomar Conf. Signals, Syst., Comput.*, 1993, pp. 1281–1284.

[5] T. Blu, "Lossless filter design in two-band rational: A new algorithm," in *Proc. Quatorzieme Colloque GRETSI*, Juan-Les-Pins, France, Sept. 1993, pp. 69–72.

[6] J. Princen, "The design of nonuniform modulated filterbanks," *IEEE Trans. Signal Processing*, vol. 43, pp. 2550–2560, Nov. 1995.

[7] T. Blu, "An iterated rational filter bank for audio coding," in *Proc. IEEE Symp. Time-Freq. Time-Scale Anal.*, Paris, France, June 1996, pp. 81–84.

[8] F. Argenti, B. Brogelli, and E. Del Re, "Design of pseudo-QMF banks with rational sampling factors using several prototype filters," *IEEE Trans. Signal Processing*, vol. 46, pp. 1709–1715, June 1998.

[9] R. D Koilpillai and P. P. Vaidyanathan, "Cosine-modulated FIR filter banks satisfying perfect reconstruction," *IEEE Trans. Signal Processing*, vol. 40, pp. 770–783, Apr. 1992.

[10] T. Q. Nguyen, T. I. Laakso, and T. E. Tuncer, "On perfect-reconstruction allpass-based cosine-modulated IIR filter banks," in *Proc. IEEE ISCAS*, 1994, pp. 33–36.

[11] C. D. Creusere and S. K. Mitra, "Efficient audio coding using perfect reconstruction noncausal IIR filter banks," *IEEE Trans. Speech Audio Processing*, vol. 4, pp. 115–123, Mar. 1996.

[12] P. P. Vaidianathan and T. Chen, "Role of anticausal inverses in multirate filter-banks—Part I: System-theoretic fundamentals," *IEEE Trans. Signal Processing*, vol. 43, pp. 1090–1102, May 1995.

[13] S. Pei and J. Shyu, "Design of 1-D and 2-D IIR eigenfilters," *IEEE Trans. Signal Processing*, vol. 42, pp. 962–966, Apr. 1994.

Suppression of Transients in Variable Recursive Digital Filters with a Novel and Efficient Cancellation Method

Vesa Välimäki and Timo I. Laakso

***Abstract*—A new method for suppressing transients in recursive infinite impulse response (IIR) digital filters is proposed. The technique is based on modifying the state (delay) variables of the filter when coefficients are changed so that the filter enters a new state smoothly without transient attacks, as originally proposed by Zetterberg and Zhang. In this correspondence, we modify the Zetterberg–Zhang algorithm to render it feasible for efficient implementation. We define a mean square error (MSE) measure for transients and determine the optimal transient suppresser to cancel the transients down to a desired level at the minimum complexity of implementation. The application of the method to all-pole and direct-form II (DF II) IIR filter sections is studied in detail. Time-varying recursive filtering with transient elimination is illustrated for tunable fractional delay filters and variable-bandwidth lowpass filters.**

***Index Terms*— Audio signal processing, recursive (IIR) filters, time-varying filters, transients, tunable filters.**

I. Introduction

Due to the recursive nature of IIR filters, abrupt changes in filter coefficients cause disturbances to values of internal state variables and, thus, result in transients at the filter output. These transients may cause trouble for practical applications, such as clicks in audio signals, and they are often the most critical problem in the implementation of tunable or time-varying recursive filters. Examples of filtering problems where transients occur are audio signal processing applications such as speech coding [1] and synthesis [2], vocal tract modeling [3], [4], model-based music synthesis [5]–[7], and equalization of audio signals [8]–[11].

Despite the importance of the problem, only a handful of research reports exist on strategies for suppressing transients in time-varying recursive filters. Six major approaches have been proposed for this task:

1) a cross-fading method [12];
2) gradual variation of coefficients using interpolation [8];

Manuscript received April 8, 1997; revised April 3, 1998. The associate editor coordinating the review of this paper and approving it for publication was Dr. Sawasd Tantaratana.

V. Välimäki is with the Laboratory of Acoustics and Audio Signal Processing, Helsinki University of Technology, Espoo, Finland (e-mail: Vesa.Valimaki@hut.fi).

T. Laakso is with the Laboratory of Telecommunications Technology, Helsinki University of Technology, Espoo, Finland (e-mail: Timo.Laakso@hut.fi).

Publisher Item Identifier S 1053-587X(98)08700-5.

3) intermediate coefficient matrix [13];
4) signal smoothing [14];
5) an input-switching method [2];
6) updating of the state vector [1].

The most general approach to transient suppression is the *state-variable update technique*, which was introduced by Zetterberg and Zhang [1]. Their main result was that, assuming a stationary input signal, every change in filter coefficients should be accompanied by an appropriate change in the internal state variables. This guarantees that the filter enters directly from one state to another without any transient response. The Zetterberg–Zhang (ZZ) method can completely eliminate the transients, but it does require that all the past input samples are known. This makes the approach impractical as such but provides a fruitful starting point for more efficient approximate algorithms. In this correspondence, we build on the ZZ method.

The motivation for our work is to find a practical way to update the state variables of a recursive filter when the filter coefficients are changed abruptly. We present a solution for transient suppression that gives an acceptable performance in terms of required MSE at the minimum implementation complexity. The new transient elimination technique introduced in this correspondence is general and can be applied to any recursive filter structure. We concentrate on the all-pole and direct-form II (DF II) structures since the transient elimination technique is best suited for structures where only the feedback part affects the state vector.

This correspondence is organized as follows. In Section II, we discuss transients in time-varying recursive filters, introduce the output-switching method, define the concept of transient, and show that the ZZ method is equivalent to the output-switching method where transients are not generated. In Section III we introduce modifications to the ZZ method that yield a new efficient parametric technique for suppressing the transient response at the desired level at minimum costs. Practical examples of transient elimination are given in Section IV in the context of tunable fractional delay allpass filters and variable-bandwidth lowpass filters. Finally, conclusions are drawn in Section V.

II. Time-Varying Recursive Filters and Transients

A. Output-Switching Method

Let us consider a recursive Nth-order digital filter with the transfer function

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + \cdots + b_N z^{-N}}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}} \tag{1}$$

where b_k and a_k are its numerator and denominator coefficients, respectively $(k = 0, 1, 2, \ldots, N)$. Assuming a causal implementation, the input–output relation of this filter may be expressed with the difference equation

$$y(n) = \sum_{k=0}^{N} b_k x(n-k) - \sum_{m=1}^{N} a_m y(n-m) \quad \text{for } n \le 0 \tag{2}$$

where $x(n)$ and $y(n)$ are the input and output signal of the filter, respectively. Unless otherwise stated, we assume that $x(n)$ and $y(n)$ are stationary over a long enough period.

In order to understand what the change of the filter characteristics means for the filter output, we consider a single change of the coefficient set at time index $n = n_c$. Ideally, the filter should instantly reach its *steady state*, and there would not be any disturbances in the output signal after the change. This can be achieved by running two

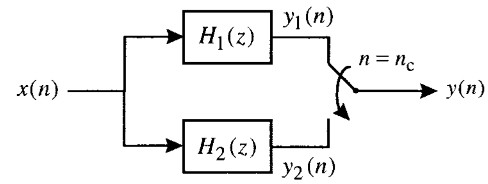

Fig. 1. Ideal transientless change in the characteristics of a recursive filter is implemented by switching the output of the filters. A single change of filter coefficients at time $n = n_c$ is considered.

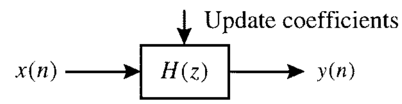

Fig. 2. Time-varying filter where the coefficients are changed during operation.

filters $H_1(z)$ and $H_2(z)$ in parallel, as shown in Fig. 1. The output signals of these two filters are

$$y_1(n) = x(n) * h_1(n) = \sum_{k=-\infty}^{\infty} x(k) h_1(n-k) \tag{3a}$$

$$y_2(n) = x(n) * h_2(n) = \sum_{k=-\infty}^{\infty} x(k) h_2(n-k) \tag{3b}$$

where the asterisk denotes discrete-time convolution. The output is switched at the time index $n = n_c > 0$, and the output of the system can be expressed as

$$y_{\text{id}}(n) = \begin{cases} y_1(n) & \text{for } 0 \le n < n_c \\ y_2(n) & \text{for } n \ge n_c. \end{cases} \tag{4}$$

We call this the *output-switching method* for implementing a time-varying filter, and it represents the ideal case where the change in the filter coefficients does not introduce any transients.

In a practical situation where multiple coefficient changes occur, realization of a time-varying filter using the output-switching method of Fig. 1 grows increasingly complex. For example, if 100 different coefficient sets are needed in a given digital filtering application, the transient-free system to be implemented requires as many as 100 filters running in parallel. This also implies that the filter coefficient sets must be known beforehand when the system is being implemented in the first place. This will destroy one of the main advantages of digital filters: the easy adaptation of coefficients.

B. The Transient

The output signal of the time-varying recursive filter depicted in Fig. 2 may be written as

$$y(n) = \begin{cases} y_1(n), & \text{for } 0 \le n < n_c \\ y_2(n) + y_{\text{t}}(n), & \text{for } n \ge n_c \end{cases} \tag{5}$$

where $y_1(n)$ and $y_2(n)$ are the steady-state responses of the filter before and after, respectively, the change in the coefficient set, and $y_t(n)$ is the *transient signal*. It is defined as the difference between the actual filter output and the ideal steady-state output signal, that is

$$y_{\text{t}}(n) = y(n) - y_{\text{id}}(n). \tag{6}$$

This fundamental relation follows from (4) and (5).

A recursive filter can be expressed in state-variable form as (see, e.g., [15])

$$\mathbf{v}(n+1) = \mathbf{F}\mathbf{v}(n) + \mathbf{q}x(n) \tag{7a}$$

$$y(n) = \mathbf{g}^{\text{T}}\mathbf{v}(n) + g_0 x(n). \tag{7b}$$

The dimensions and the values of the matrices and vectors used in (7a) and (7b) depend on the realization structure of the filter. Let us consider the time-varying filter presented in Fig. 2. According to (7a), the state-variable vector $\mathbf{v}(n)$ can be expressed as a function of the input signal $x(n)$ and coefficient matrices when the coefficients have been changed at time n_c [1]

$$\mathbf{v}(n) = \begin{cases} \mathbf{F}_1^n \mathbf{v}(0) + \sum_{k=0}^{n-1} \mathbf{F}_1^{n-k-1}\mathbf{q}x(k), & 0 < n \le n_c \\ \mathbf{F}_2^{n-n_c}\mathbf{v}(n_c) + \sum_{k=n_c}^{n-1} \mathbf{F}_2^{n-k-1}\mathbf{q}x(k), & n > n_c \end{cases} \tag{8a-b}$$

where $\mathbf{v}(0)$ is the initial state of the filter, and $\mathbf{F}_1$ and $\mathbf{F}_2$ are the coefficient matrices before and after the change of coefficients at time index $n = n_c$, respectively. In the following, we assume that $n_c \gg 0$ so that the *initial transient* $\mathbf{F}_1^n\mathbf{v}(0)$ has died out, and this term can be neglected in the further analysis. At the time of the coefficient change $(n = n_c)$, the state vector can be expressed as

$$\mathbf{v}(n_c) = \sum_{k=0}^{n_c-1} \mathbf{F}_1^{n_c-k-1}\mathbf{q}x(k) \tag{9}$$

and by substituting (9) into (8b), we obtain the state vector after the coefficients have been changed

$$\mathbf{v}(n) = \mathbf{F}_2^{n-n_c}\sum_{k=0}^{n_c-1} \mathbf{F}_1^{n_c-k-1}\mathbf{q}x(k) + \sum_{m=n_c}^{n-1}\mathbf{F}_2^{n-m-1}\mathbf{q}x(m) \qquad n > n_c. \tag{10}$$

This form can be elaborated to explicitly show the cause of the transient in the state vector

$$\mathbf{v}(n) = \mathbf{F}_2^{n-n_c}\Delta\mathbf{v}(n_c) + \sum_{k=0}^{n-1}\mathbf{F}_2^{n-k-1}\mathbf{q}x(k), \quad n > n_c \tag{11}$$

with

$$\Delta\mathbf{v}(n_c) = \sum_{k=0}^{n_c-1}\left(\mathbf{F}_1^{n_c-k-1} - \mathbf{F}_2^{n_c-k-1}\right)\mathbf{q}x(k). \tag{12}$$

The first term in (11) represents the *transient in the state vector* due to coefficient change, and the second term is the steady-state response of the filter to the input after the parameters have changed. The transient at the output of the filter can be expressed with the help of the above equations as

$$y_t(n) = y(n) - y_{id}(n) = \mathbf{g}_2^T\mathbf{F}_2^{n-n_c}\Delta\mathbf{v}(n_c) \quad \text{for } n > n_c \tag{13}$$

where $\mathbf{g}_2$ is the vector in the output equation (7b) after the change of coefficients.

C. Zetterberg–Zhang (ZZ) Method for Transient Elimination

As stated by Zetterberg and Zhang [1], one way to completely eliminate the transient caused by the change of coefficients is to *subtract* the term $\Delta\mathbf{v}(n_c)$ from the state vector at time n_c. The modified state vector (at time index $n = n_c$) may be written as

$$\begin{aligned}\mathbf{v}(n_c) - \Delta\mathbf{v}(n_c) &= \sum_{k=0}^{n_c-1}\mathbf{F}_1^{n_c-k-1}\mathbf{q}x(k) - \sum_{k=0}^{n_c-1}\left(\mathbf{F}_1^{n_c-k-1} - \mathbf{F}_2^{n_c-k-1}\right)\mathbf{q}x(k) \\ &= \sum_{k=0}^{n_c-1}\mathbf{F}_2^{n_c-k-1}\mathbf{q}x(k).\end{aligned} \tag{14}$$

This is the *Zetterberg–Zhang (ZZ) method* for the elimination of transients. It is seen that as a result of this correction, the state vector now contains the values corresponding to the steady state of the filter after the coefficient change. In other words, the mismatch in the state vector has been removed, and the transient has been completely eliminated. In fact, The ZZ method implements the output-switching method introduced in Section II-A. The drawbacks of the ZZ method are those discussed in Section II-A: high computational cost (since each filter set requires an actual filtering operation all the time) and inflexible implementation (since all the filter coefficient sets must be predetermined before the filtering may start). In the following, we propose modifications to the ZZ method and introduce a new and efficient method for transient elimination that does not have these problems.

III. Novel Method for Suppressing Transients

In this section, we present a new transient cancellation method that is based on the general idea of the ZZ method. Instead of requiring complete elimination, we define the desired accuracy of transient cancellation and devise a technique to design a transient eliminator that meets the requirements at minimum implementation costs.

A. Modifications to the ZZ Method

Equation (14) reveals that the ZZ method can equivalently be implemented by replacing the state vector with the following *transient cancellation vector* (TCV):

$$\mathbf{v}_{tc} = \sum_{k=0}^{n_c-1}\mathbf{F}_2^{n_c-k-1}\mathbf{q}x(k) \tag{15}$$

which simply contains the steady-state vector obtained when the coefficient matrix $\mathbf{F}_2$ is used from the beginning. Instead of computing the state vector, we suggest approximating $\mathbf{v}_{tc}$ with a *truncated sum* as [3], [17]

$$\hat{\mathbf{v}}_{tc}(N_a) = \sum_{k=n_c-N_a}^{n_c-1}\mathbf{F}_2^{n_c-k-1}\mathbf{q}x(k) = \sum_{k=1}^{N_a}\mathbf{F}_2^{k-1}\mathbf{q}x(n_c-k) \tag{16}$$

where N_a is called the *advance time*. It expresses the number of samples of the input signal that are used for computing the state vector in advance of the coefficient change. Now, the computation of the transient cancellation vector only takes finite time and need not be updated all the time in parallel with the filtering operation.

The use of a finite number of samples for computing $\hat{\mathbf{v}}_{tc}$ is motivated by the fact that the impulse response of a stable recursive filter decays exponentially and can thus be regarded as finite-length. Thus, the knowledge of the *effective length* of the impulse response (i.e., how many values of this impulse response are observably nonzero for the application) from the input to the state vector helps to estimate how many past input samples need to be taken into account in updating the transient cancellation vector. This principle may be applied to all discrete-time filter structures.

B. Allpole Filter

For simplicity, let us first consider an allpole filter with an impulse response $h_1(n)$. The output of this filter is given by (3a) and its state vector is

$$\mathbf{v}_1(n) = [v_1(n) \quad v_1(n-1) \quad \cdots \quad v_1(n-N+1)]^T \tag{17}$$

and the matrices and vectors used in the state-variable representation (8) are

$$\mathbf{F} = \begin{bmatrix} -a_1 & -a_2 & \cdots & -a_{N-1} & -a_N \\ 1 & 0 & & 0 & 0 \\ 0 & 1 & & & \vdots \\ \vdots & & \ddots & 0 & 0 \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\mathbf{g} = \begin{bmatrix} -a_1 \\ -a_2 \\ \vdots \\ -a_N \end{bmatrix}, \quad g_0 = b_0. \tag{18}$$

The coefficients of this filter are changed at the time index n_c, and the resulting new impulse response is denoted by $h_2(n)$. The output signal of the filter can then be written as in (3b). We now apply the new transient cancellation method introduced above, and thus, the state vector after the change is

$$\hat{\mathbf{v}}_{\rm tc}(n) = [\hat{v}_{\rm tc}(n) \quad \hat{v}_{\rm tc}(n-1) \quad \cdots \quad \hat{v}_{\rm tc}(n-N+1)]^{\rm T}. \quad (19)$$

A key observation in finding how to determine $N_{\rm a}$ is to understand how the contents of the state vector of an allpole filter are produced. The state vector contains the N latest output samples of the filter, that is

$$\mathbf{v}(n) = [y(n-1) \quad y(n-2) \quad \cdots \quad y(n-N)]^{\rm T}. \quad (20)$$

On the other hand, the output signal $y(n)$ is the convolution of the impulse response of the filter with the input signal [see (3a)]. Thus, in the case of an allpole filter, it is necessary to determine the *effective length* (EL) *of the impulse response* of the filter N_P to know how many past input samples effectively contribute to the first value of the state vector $v_1(n) = y(n-1)$. After N sample cycles, this value will disappear from the state vector. Thus, the advance time of the transient cancellation method may then be set equal to

$$N_{\rm a} = N_P + N \quad (21)$$

where N_P and N are the effective length of the impulse response and the order of the filter, respectively. This choice of $N_{\rm a}$ ensures that the updated state vector suffers sufficiently little from the truncation of the input signal, according to the same criterion that was used to determine N_P. The older input samples do not, in effect, contribute to the current state vector, and hence, it is not necessary to include them.

The effective length of an infinite impulse response can be determined using one of several techniques that have been reviewed in another paper [16]. We propose the use of a new energy-based method introduced in [16].

C. Determining N_a Using Energy-Based Criterion

The total energy of a causal, real impulse response is defined by

$$E = \sum_{n=0}^{\infty} [h(n)]^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} |H(e^{j\omega})|^2 \, d\omega. \quad (22)$$

The *energy-based effective length* of the impulse response is defined as the smallest non-negative integer time index N_P by which at least P% of the total energy of the impulse response has arrived [16]. The corresponding *accumulated energy* $E_{\rm A}(N_P)$ can be expressed as

$$E_{\rm A}(N_P) = \sum_{n=0}^{N_P} h^2(n) \geq E_P = \frac{P}{100}E. \quad (23)$$

Hence, we always require that $E_{\rm A}(N_P) \geq E_P$ since the effective length N_P must be an integer.

Computation of the energy-based EL of a filter thus requires solving two problems: 1) the total energy of the impulse response and 2) the time index N_P corresponding to the effective length. The total energy is easily solved by integration in the frequency domain [17], [18] or by using closed-form formulas that exist for low-order filters [17], [19]. The effective length may then be determined using a simple algorithm described in [16] or by using approximative formulas that are based on the analysis of first- and second-order subsections [16].

The *energy of the transient signal* $y_t(n)$ is defined by

$$E_{\rm t} = \sum_{n=0}^{\infty} [y_{\rm t}(n)]^2. \quad (24)$$

The proposed new transient elimination method effectively truncates the transient signal. We may determine the accumulated energy of the transient within $N_{\rm a}$ samples, that is

$$E_{\rm At}(N_{\rm a}) = \sum_{n=0}^{N_{\rm a}} [y_{\rm t}(n)]^2. \quad (25)$$

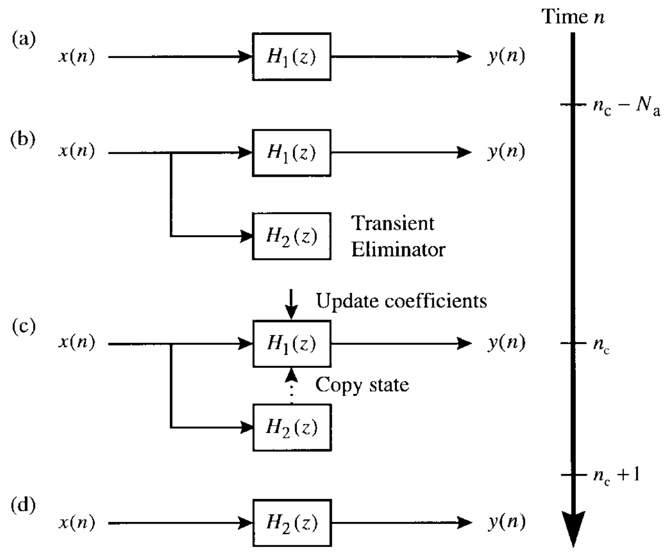

Fig. 3. Different phases of the new transient suppression scheme for a single coefficient change at time index $n = n_{\rm c}$. (a) Initially, coefficient set 1 is used in a conventional way. (b) $N_{\rm a}$ sample intervals before the change, a transient eliminator, which has the new coefficient values and zero initial state, is started in parallel with the signal filter. (c) At $n = N_{\rm a}$, the state (TCV) of the transient eliminator is copied to the signal filter. (d) Finally, coefficient set 2 is used to filter the input signal.

This is the amount of energy that is *excluded* for a given value of parameter $N_{\rm a}$. The problem in using this expression is that the actual form and the exact length of the transient depend on the filter's input signal. However, we would like to have a measure that is independent of the input. This can be done by utilizing (21); the length of the transient is equivalent to the length of the impulse response of the filter (with the new coefficient values) appended by the filter order N. Although we cannot say exactly how much of the transient energy is canceled in a particular case, according to our experiment, very close to P% is usually eliminated.

D. Suppression of Transients of a Time-Varying Direct-Form II Filter

Let us consider implementation of the transient elimination technique when the time-varying filter $H(z) = B(z)/A(z)$ is implemented using the direct-form (DF) II structure. For DF II realization, the column vector $\mathbf{v}(n)$ contains N state variables

$$\mathbf{v}(n) = [v_1(n) \quad v_2(n) \quad \cdots \quad v_N(n)]^{\rm T} \quad (26)$$

and

$$\mathbf{F} = \begin{bmatrix} -a_1 & -a_2 & \cdots & -a_{N-1} & -a_N \\ 1 & 0 & & 0 & 0 \\ 0 & 1 & & & \vdots \\ \vdots & & \ddots & 0 & 0 \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\mathbf{g} = \begin{bmatrix} b_1 - b_0 a_1 \\ b_2 - b_0 a_2 \\ \vdots \\ b_N - b_0 a_N \end{bmatrix}, \quad g_0 = b_0. \quad (27)$$

A key observation here is that the state of the DF II filter structure is affected by the past input values only in proportion to the magnitude

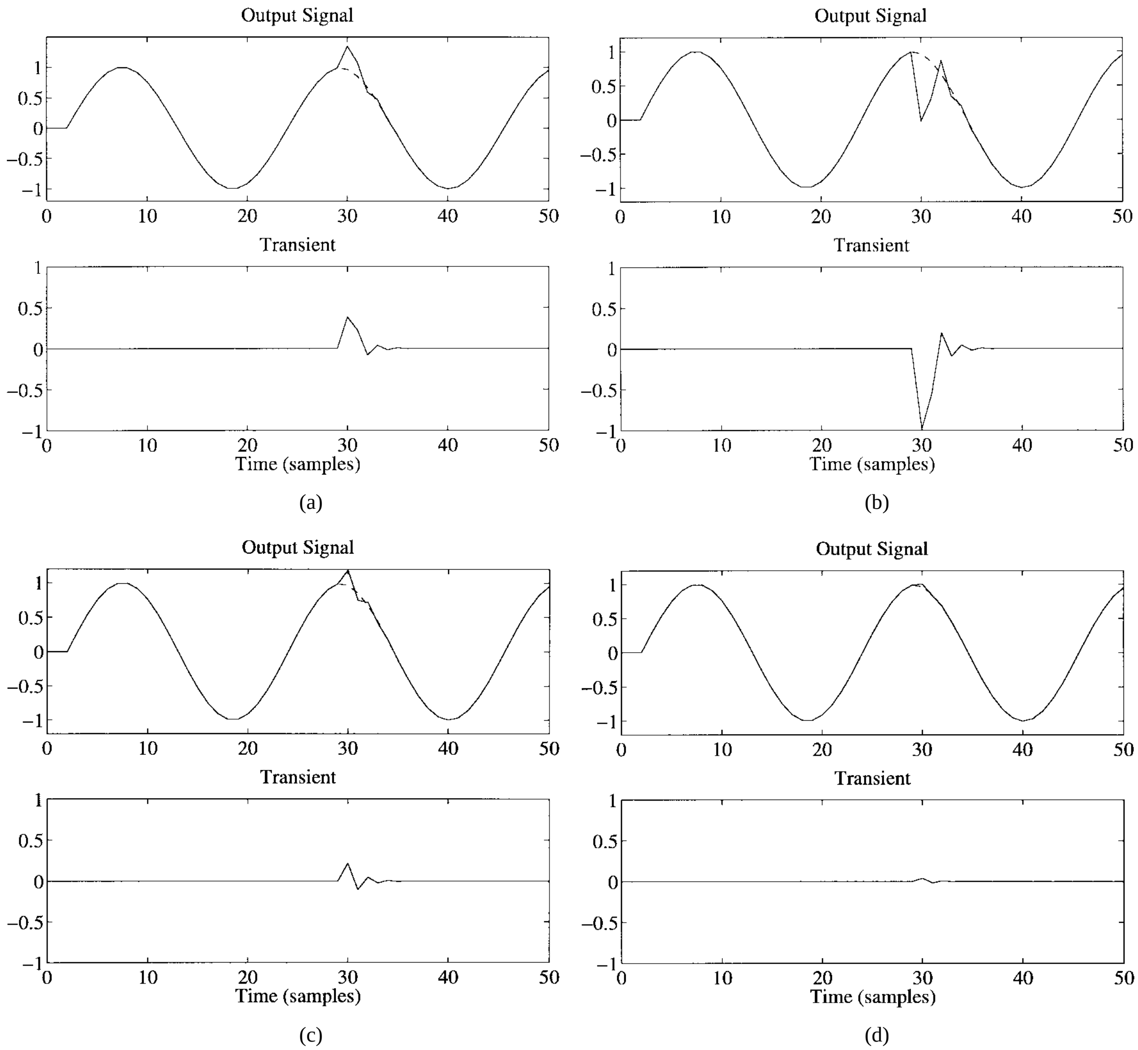

Fig. 4. Output signal of a second-order allpass filter when the delay is changed from 2 to 1.5 samples at time index $n = 30$ using ideal transient elimination (dashed line) and (a) no transient elimination. (b) Proposed transient suppresser with $N_\mathrm{a} = 0$ (clearing of state variables). (c) $N_\mathrm{a} = 2$. (d) $N_\mathrm{a} = 4$ (solid line). The lower figure displays the difference (transient) of the actual and ideal output signals in each case.

of the impulse response $h_r(n)$ of its recursive part, that is

$$h_r(n) = \begin{cases} 0, & n < 0 \\ 1, & n = 0 \\ \mathbf{q}^\mathrm{T}\mathbf{F}^{n-1}\mathbf{q}, & n > 0. \end{cases} \tag{28}$$

The feedforward coefficients do not contribute to the transient because vector $\mathbf{g}$ that contains these coefficients [see (27)] does not affect the state vector in (7a). The elements of the state vector of a DF II filter structure (at time $n = n_c$)

$$\mathbf{v}(n_c) = [v_1(n_c) \quad v_2(n_c) \quad \cdots \quad v_N(n_c)]^\mathrm{T} \tag{29}$$

can be rewritten by means of the impulse response $h_r(n)$ of the recursive part of the filter $H(z)$

$$v_m(n_c) = \sum_{k=0}^{n_c-1} h_\mathrm{r}(k)x(n_c - m - k) = y_r(n_c - m) \quad \text{for } m = 1, 2, \ldots, N \tag{30}$$

where $y_\mathrm{r}(n)$ is the output signal of the recursive part of the IIR filter.

The transient elimination method then works as depicted in Fig. 3; initially, the actual IIR filter $H_1(z)$—hereafter called the *signal filter*—processes the input signal [see Fig. 3(a)]. N_a samples before the coefficient change, the input signal $x(n)$ is fed into two systems: the signal filter $H_1(z)$ and the *transient eliminator* that has the new transfer function $H_2(z)$ as illustrated in Fig. 3(b). In the case of a DF II structure, the eliminator consists of the signal filter's recursive part $1/A(z)$ with the next coefficient set. At time $n = n_\mathrm{c}$, the coefficients of the signal filter are updated, and the state vector (TCV) is copied from the transient eliminator to the signal filter's state, as shown in Fig. 3(c). The transient eliminator is now turned off. Finally, the new coefficient set is used to filter the input signal [Fig. 3(d)]. As a result of this procedure, the transient phenomenon will be sufficiently suppressed if the value of parameter N_a is large enough.

The state variables of the transient eliminator are updated according to

$$v_{\mathrm{tc},k}(n) = v_{\mathrm{tc},k-1}(n-1) \quad \text{for } k = 2, 3, \ldots, N \tag{31}$$

$$v_{\mathrm{tc},1}(n_c) = x(n) - \sum_{k=1}^{N} a_k v_{\mathrm{tc},k}(n-1) \tag{32}$$

It is seen that (31) and (32) present a state-variable description of an *allpole filter* without output. The output signal is not used since the purpose of the transient eliminator is simply to provide the new state vector [transient cancellation vector (TCV)] for the signal filter, as is also shown in Fig. 3(b).

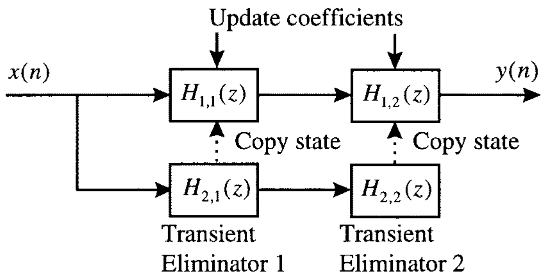

Fig. 5. Implementation of a transientless time-varying filter as a cascade of subfilters.

IV. Examples

We present two examples that illustrate the functioning of the new transient suppression method.

Example 1: We filter a low-frequency sine wave (0.0454 times the sampling frequency f_s) with a second-order allpass filter (direct-form II implementation) that approximates a constant group delay. Initially, the filter coefficients are $a_1 = 0$ and $a_2 = 0$ (corresponding to a constant group delay of 2 samples) and at time index 30, they are changed to values $a_1 = 0.400\,00$ and $a_2 = -0.028\,571$, which gives a maximally flat group delay approximation of a delay of 1.5 sample intervals. We present the output and transient signals of the filter in four cases:

a) without transient cancellation;
b) when the transient cancellation method is used with parameter values $N_{\rm a} = 0$ (equivalent to clearing the state variables before setting the new coefficient values);
c) $N_{\rm a} = 2$;
d) $N_{\rm a} = 4$.

These output signals are compared with the "ideal" or transientless output signal (dashed line in Fig. 4) that has been computed using the output-switching method by running two filters in parallel and changing the output at time $n = 30$ (see Fig. 1). The transient signal shown in the lower figure in each case is the difference of the output signals of the time-varying and the ideal filter.

Note that the case without transient elimination is not the worst case: the maximum amplitude (and residual energy) of the transient signal is larger in Fig. 4(b) than in Fig. 4(a). This confirms that the simple trick of clearing the state variables of the filter before changing the coefficients is not useful. In Fig. 4(c) $(N_{\rm a} = 2)$, the maximum amplitude of the transient has been slightly suppressed with respect to Fig. 4(a) (no suppression). Finally, Fig. 4(d) presents the output and transient signals with $N_{\rm a} = 4$. Now, the transient signal has been much suppressed. More suppression can be achieved by using a larger value for the advance time parameter $N_{\rm a}$.

Example 2: A sine wave $(0.125 f_s)$ is filtered with a fourth-order Butterworth lowpass filter, the cutoff frequency of which is changed from $0.25 f_s$ to $0.05 f_s$. In the former case, the amplitude of the sine wave is not affected, whereas in the latter case, it is attenuated to about 2% of its original value (attenuation of about −34 dB). The transfer function of the filter has been decomposed into a product of second-order transfer functions that are implemented in the DF II structure and cascaded (see Fig. 5). Fig. 6(a) shows the output signal and the transient without transient suppression. The ideal output signal (which is obtained with the output-switching method) is displayed with a dashed line. Fig. 6(b) presents the transient signal when the proposed suppression method is used. Both second-order sections have their individual transient eliminators, as shown in Fig. 5. In this example, we select $N_{\rm a}$ according to the largest pole radius of the two filters. The pole radii of the two filters after coefficient change are $r_1 = 0.7455$ and $r_2 = 0.8880$ and pole angles $\theta_1 = 0.1237$ rad and $\theta_2 = 0.2916$ rad, respectively. The computation of the cumulative energy of the impulse response tells that the 95%-energy length of the filter is 14 samples. The order of the filter is 2, and thus, according to (21), we choose $N_{\rm a} = 16$. The zeros of the transfer function do not contribute to the transient since they do not affect the state variables in the DF II structure. The energy of the transient without suppression [Fig. 6(a)] is 0.2957, and with suppression, it is 0.016 77, which corresponds to a suppression of 94.3% of the energy of the transient. This is very close to what we required. Remember that the effective length is only based on the information of the filter and not the input signal, and thus, the eliminated energy will not be exactly given by (25).

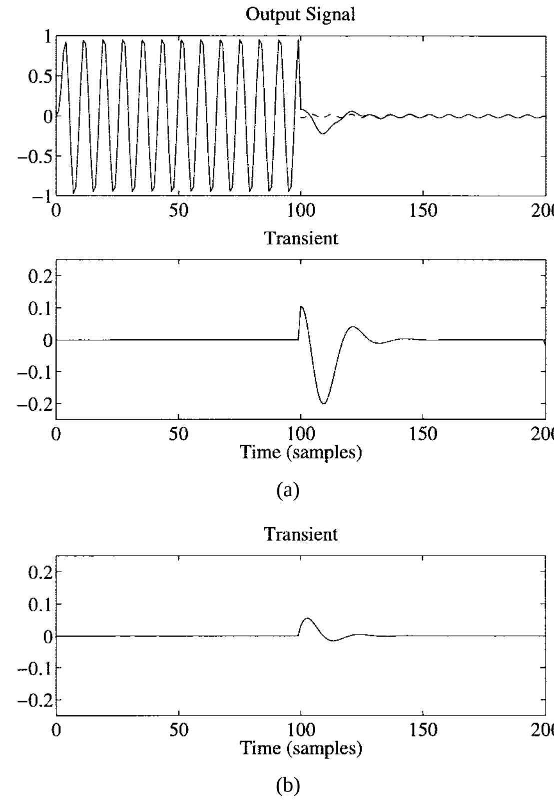

Fig. 6. Example of transient elimination when a cascade realization of a fourth-order lowpass filter is used. The input is a sine wave, and the cut-off frequency is changed at $n = 100$. (a) Output and transient waveforms are shown without suppression. (b) Transient when the proposed suppression method is used $(N_{\rm a} = 16)$. The ideal output signal is shown with a dashed line in Fig. 6(a).

V. Conclusion

A novel and efficient transient elimination technique for tunable recursive filters has been introduced. The technique is based on a method of state-variable update at the time of the filter coefficient change. It was shown that a finite number of samples is sufficient for computing new values for the state vector at the time of coefficient change. The advance time of the transient eliminator is determined such that it will result in required transient suppression accuracy at minimum implementation costs. The proposed transient elimination can be used with all kinds of recursive digital filters. However, it can be most efficiently implemented when used with an allpole or direct-form II filter structure. The new transient cancellation method presented in this paper has several potential applications, for example, in real-time audio signal processing where the properties of filters need to be changed while filtering a signal.

ACKNOWLEDGMENT

The authors wish to thank Dr. J. Mackenzie (King's College, London, U.K.) for his contribution in the initial phase of this study and Dr. J.-M. Jot (IRCAM, Paris, France), Prof. P. Alku (University of Turku, Finland), and Prof. M. Karjalainen (HUT, Espoo, Finland) for inspiring discussions on transients in digital filters and methods for their elimination.

REFERENCES

[1] L. H. Zetterberg and Q. Zhang, "Elimination of transients in adaptive filters with application to speech coding," *Signal Process.*, vol. 15, no. 4, pp. 419–428, 1988.

[2] W. Verhelst and P. Nilens, "A modified-superposition speech synthesizer and its applications," in *Proc. 1986 IEEE Int. Conf. Acoust., Speech, Signal Process.*, Tokyo, Japan, Apr. 7–11, 1986, vol. 3, pp. 2007–2010.

[3] V. Välimäki and M. Karjalainen, "Implementation of fractional delay waveguide models using allpass filters," in *Proc. 1995 IEEE Int. Conf. Acoust., Speech, Signal Process.*, Detroit, MI, May 9–12, 1995, vol. 2, pp. 1524–1527.

[4] V. Välimäki, "Discrete-time modeling of acoustic tubes using fractional delay filters," Ph.D. dissertation, Espoo, Finland, Helsinki Univ. Technol., Faculty Elect. Eng., Lab. Acoust. Audio Signal Process., Rep. 37, p. 193, Dec. 1995.

[5] J. O. Smith, "J. O. Smith III comments on Sullivan Karplus–Strong article," *Comput. Music J.*, vol. 15, no. 2, pp. 10–11, 1991.

[6] V. Välimäki, T. I. Laakso, and J. Mackenzie, "Elimination of transients in time-varying allpass fractional delay filters with application to digital waveguide modeling," in *Proc. 1995 Int. Comput. Music Conf.*, Banff, Alta., Canada, Sept. 3–7, 1995, pp. 327–334.

[7] V. Välimäki, J. Huopaniemi, M. Karjalainen, and Z. Jánosy, "Physical modeling of plucked string instruments with application to real-time sound synthesis," *J. Audio Eng. Soc.*, vol. 44, no. 5, pp. 331–353, May 1996.

[8] J. N. Mourjopoulos, E. D. Kyriakis-Bitzaros, and C. E. Goutis, "Theory and real-time implementation of time-varying digital audio filters," *J. Audio Eng. Soc.*, vol. 38, nos. 7–8, pp. 523–536, July/Aug. 1990.

[9] S. S. Nikolaidis, J. N. Mourjopoulos, and C. E. Goutis, "A dedicated processor for time-varying digital audio filters," *IEEE Trans. Circuits Syst. II*, vol. 40, pp. 452–455, July 1993.

[10] U. Zölzer, B. Redmer, and J. Bucholtz, "Strategies for switching digital audio filters," presented at the *95th AES Convention*, New York, NY, Oct. 7–10, 1993, preprint no. 3714.

[11] Y. Ding and D. Rossum, "Filter morphing of parametric equalizers and shelving filters for audio signal processing," *J. Audio Eng. Soc.*, vol. 43, no. 10, pp. 821–826, Oct. 1995.

[12] J.-M. Jot, V. Larcher, and O. Warusfel, "Digital signal processing issues in the context of binaural and transaural stereophony," presented at the *98th AES Convention*, Paris, France, Feb. 25–28, 1995, preprint no. 3980.

[13] R. Rabenstein, "Minimization of transient signals in recursive time-varying digital filters," *Circuits, Syst., Signal Process.*, vol. 7, no. 3, pp. 345–359, 1988.

[14] P. T. Harju, S. J. Ovaska, and V. Välimäki, "Delayless signal smoothing using a median and predictive filter hybrid," in *Proc. 1996 Int. Conf. Signal Process.*, Beijing, China, Oct. 14–18, 1996, pp. 87–90.

[15] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms and Applications*, 2nd ed. New York: Macmillan, 1992, pp. 510–526.

[16] T. I. Laakso and V. Välimäki, "Energy-based effective length of the impulse response of a recursive filter," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Seattle, WA, May 12–15, 1998, vol. 3, pp. 1253–1256.

[17] E. I. Jury, *Theory and Application of the z-Transform Method*. New York: Wiley, 1964, p. 298.

[18] J. Chun and N. K. Bose, "Fast evaluation of an integral occurring in digital filtering applications," *IEEE Trans. Signal Processing*, vol. 43, pp. 1982–1986, Aug. 1995.

[19] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975, p. 426.

Translation-Invariant Denoising Using Multiwavelets

Tien D. Bui and Guangyi Chen

***Abstract*— Translation invariant (TI) single wavelet denoising was developed by Coifman and Donoho, and they show that TI is better than non-TI single wavelet denoising. On the other hand, Strela *et al.* have found that non-TI multiwavelet denoising gives better results than non-TI single wavelets. In this correspondence, we extend Coifman and Donoho's TI single wavelet denoising scheme to multiwavelets. Experimental results show that TI multiwavelet denoising is better than the single case for soft thresholding.**

***Index Terms*—Denoising, multiwavelet, translation invariant, univariate and bivariate thresholding.**

I. INTRODUCTION

Over the past decade, wavelet transforms have received a lot of attention from researchers in many different areas. Both discrete and continuous wavelet transforms have shown great promise in such diverse fields as image compression, image denoising, signal processing, computer graphics, and pattern recognition to name only a few. In denoising, single orthogonal wavelets with a single-mother wavelet function have played an important role (see [2]–[4]). The pioneering work of Donoho and Johnstone [2], [3] can be summarized as follows. Let $g(t)$ be the noise-free signal and $f(t)$ the signal corrupted with white noise $z(t)$, i.e., $f(t) = g(t) + \sigma z(t)$, where $z(t)$ has a normal distribution $N(0, 1)$. Donoho and his coworkers proposed the following algorithm.

1) Discretize the continuous signal $f(t)$ into f_i, $i = 1, \cdots, n$ (e.g., via uniform sampling).
2) Transform the signal f_i into an orthogonal domain by discrete single wavelet transform.
3) Apply soft or hard thresholding to the resulting wavelet coefficients by using the threshold $\lambda = \sqrt{2\sigma^2 \log n}$.
4) Perform inverse discrete single wavelet transform to obtain the denoised signal.

The denoising is done only on the detail coefficients of the wavelet transform. It has been shown that this algorithm offers the advantages of smoothness and adaptation. However, as Coifman and Donoho pointed out, this algorithm exhibits visual artifacts: Gibbs phenomena in the neighborhood of discontinuities. Therefore, they propose in [1] a translation-invariant (TI) denoising scheme to suppress such artifacts by averaging over the denoised signals

Manuscript received September 23, 1997; revised May 11, 1998. This work was supported by Research Grants from the Natural Sciences and Engineering Research Council of Canada and by the Fonds pour la Formation de Chercheurs et l'Aide à la Recherche of Quèbec. The associate editor coordinating the review of this paper and approving it for publication was Dr. Konstantin Konstantinides.

The authors are with the Department of Computer Science, Concordia University, Montreal, P.Q., Canada H3G 1M8 (e-mail: bui@cs.concordia.ca).

Publisher Item Identifier S 1053-587X(98)08815-1.