ELIMINATION OF TRANSIENTS IN TIME-VARYING ALLPASS FRACTIONAL DELAY FILTERS WITH APPLICATION TO DIGITAL WAVEGUIDE MODELING

Vesa Välimäki*, Timo I. Laakso†, and Jonathan Mackenzie†

*Helsinki University of Technology, Laboratory of Acoustics and Audio Signal Processing
Otakaari 5A, FIN-02150 Espoo, Finland

†University of Westminster, School of Electronic and Manufacturing Systems Engineering
115 New Cavendish St., London W1M 8JS, U.K.

vesa.valimaki@hut.fi, timol@westminster.ac.uk, mackenj1@westminster.ac.uk

ABSTRACT: This paper considers discrete-time allpass filters that implement a time-varying fractional delay. These recursive filters are desirable from the point of view of implementational efficiency and flat magnitude response, but they are prone to transient effects when their parameters are changed. A state variable update approach to this problem is reviewed, the basic idea of which is to modify also the state of the filter when coefficients are changed so that the filter enters a new state smoothly without transient attacks. This method is adapted to give a new and simple practical method for eliminating the transients. The effectiveness of the new technique is verified by applying it to a digital waveguide model of a vibrating string.

1. INTRODUCTION

A fractional delay (FD) filter is a device for implementing a noninteger delay, a process equivalent to bandlimited interpolation between samples. FD filters are of fundamental importance in digital waveguide models of musical instruments. They are used for fine-tuning the length of delay lines that simulate a vibrating string or an acoustic tube, which is necessary to achieve the desired pitch exactly. Fractional delay filters are also used in several other areas of digital signal processing where accurate time delays are required, e.g., in speech coding, in time delay estimation, in timing adjustment of digital modems, and in asynchronous sample rate conversion.

A fractional delay can be approximated by recursive (IIR, infinite impulse response) or nonrecursive (FIR, finite impulse response) digital filters. However, a fixed fractional delay rarely solves a DSP problem: in many applications it is essential to be able to tune the value of the delay continuously in real time. This paper concentrates on *time-varying* FD filters, which are needed in practical applications such as waveguide modeling.

It is straightforward to exploit FIR filters in a time-varying situation. This is because they do not have transient or other problems when the filter coefficients are modified (i.e., the desired delay is changed). This is, of course, true only when the FIR filter has been implemented in a correct way so that the input samples that are needed for computing the output of the filter after the change of the delay value are available. Special care is needed with FIR filters when the order of the filter is changed or whole unit delays are added to or subtracted from the system. An additional advantage of nonrecursive filters is that they are stable in all (time-invariant and time-varying) situations.

A weakness of FIR filters is that in general a rather high-order filter is needed to meet the same requirements as a recursive filter. FIR-type fractional delay filters have an additional disadvantage of not having allpass-type frequency response, which is a basic property of a delay element: an FIR FD filter boosts or attenuates some frequencies at the same time when it tries to delay them. This can be harmful when the FD filter is used inside a delay line loop of a digital waveguide model. If the magnitude response of the FD filter exceeds unity at some frequencies, the model may become unstable (i.e., the overall loop gain may exceed unity). On the other hand, if the magnitude response of the FD filter is less than one, this in effect adds to the losses in the waveguide that is being simulated. This changes the sound quality of the waveguide model.

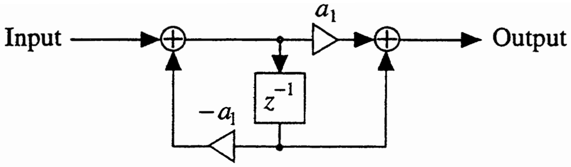

Fig. 1. *Direct form II realization of a first-order discrete-time allpass filter.*

In this paper we concentrate on design and implementation of a variable-length delay line using allpass FD filters. The main motivation for the use of an allpass filter for realizing a fractional delay is that its gain is always unity at all frequencies, just like that of an ideal delay element. Furthermore, a satisfactory delay approximation is usually obtained using a low-order allpass filter. In Section 2 we focus on the allpass FD filter that is a *maximally-flat group delay approximation* of the ideal fractional delay (Laakso *et al.* 1992, 1994). This design is especially useful because the filter coefficients can be expressed in closed form and the delay approximation is very good at low frequencies.

A major problem with recursive filters—including the allpass FD filter—is that when the parameters of the filter are changed during operation, this causes a transient at the output, similar to a response to abrupt changes in the input signal. In audio applications, the transients are particularly annoying as they are audible as disturbing clicks. In Section 3 we briefly discuss the known techniques for eliminating such transients in a recursive digital filter. Then we present a new efficient technique to update the state variables of the maximally-flat allpass FD filter in a time-varying case to suppress transients.

In Section 4, we test the allpass FD filter with a transient eliminator in digital waveguide simulation of a vibrating string. Comparison with an allpass FD filter without the eliminator shows that the performance of the proposed technique is audibly superior, and that practically all the transients can be eliminated with only slightly more complex implementation.

2. ALLPASS FRACTIONAL DELAY FILTERS

An allpass filter is well-suited for FD approximation because its magnitude response is equal to unity by definition. The transfer function $A(z)$ of an Nth-order discrete-time allpass filter is

$$A(z) = \frac{z^{-N}D(z^{-1})}{D(z)} = \frac{a_N + a_{N-1}z^{-1}+\ldots+a_1 z^{-(N-1)} + z^{-N}}{1 + a_1 z^{-1}+\ldots+a_{N-1}z^{-(N-1)} + a_N z^{-N}} \tag{1}$$

where $D(z)$ is the denominator polynomial and the numerator polynomial is its mirrored version. Figure 1 shows the direct form II realization of a first-order allpass filter.

2.1. Maximally-Flat Group Delay Design

A simple method for the design of FD allpass filters has been introduced by Laakso *et al.* (1992, 1994) and Välimäki (1994). This method is based on the maximally-flat (MF) group delay all-pole filter design developed by Thiran (1971). A remarkable feature of this design is that the filter coefficients can be given in closed form as

$$a_k = (-1)^k \binom{N}{k} \prod_{n=0}^{N} \frac{D-N+n}{D-N+k+n} \tag{2}$$

where N is the order of the allpass filter, D is the desired delay, and $k = 0, 1, 2, \ldots, N$. The coefficient of the first-order ($N = 1$) filter is

$$a_1 = \frac{1-D}{1+D} \tag{3}$$

This same formula for the coefficient of a first-order allpass filter was obtained by Jaffe and Smith (1983) using another approach. The coefficients for the second-order ($N = 2$) MF allpass filter are given by

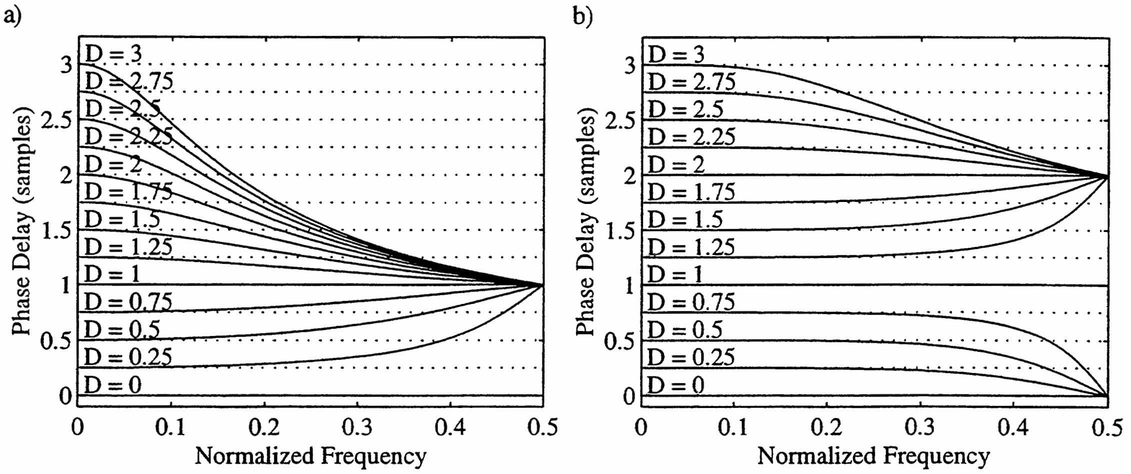

Fig. 2. *Phase delay (solid curves) of a) the first-order and b) the second-order MF allpass FD filter for several values of delay parameter D. The dotted lines show the nominal phase delay for each case.*

$$a_1 = -2\frac{D-2}{D+1} \quad \text{and} \quad a_2 = \frac{(D-1)(D-2)}{(D+1)(D+2)} \tag{4}$$

Figure 2 shows the phase delay $\tau_{\mathrm{p}}(\omega) = -\Theta(\omega)/\omega$ of these allpass FD filters for several values of D.

3. ELIMINATION OF TRANSIENTS IN ALLPASS FILTERS

The recursive nature of IIR filters means that abrupt changes in either the filter input, internal states (delay variables) or coefficients cause disturbances to the future states and transients in the output values of the filter. These disturbances depend on the size of the filter's impulse response so that they are in principle infinitely long in the IIR case. In practice, however, they decay exponentially and can be treated as having finite length. In any case, these transients are usually long enough to cause disturbing clicks in audio applications and they are typically the most serious problem in the implementation of time-varying recursive filters.

In spite of their importance, there exists only a handful of research reports on such transients. A most general approach was presented by Zetterberg and Zhang (1988) who used a state-space formulation of the recursive filter. Their main result was that, assuming a stationary input signal, every change in filter coefficients should be accompanied by an appropriate change in internal state (delay) variables. This guarantees that the filter switches directly from one state to another without any transient response. This method can thus completely eliminate the transients, but it does require that all past input signal values to the filter are known. This method is unfortunately complex and inflexible and thus impractical as such, but provides a good starting point for approximate but more efficient algorithms.

Another technique to reduce the transients is to use a set of intermediate filter coefficients for one sampling cycle after the previous and before the next coefficients (Rabenstein 1988). This technique is approximate and based on assumptions of a white noise-like input signal. In our experience it is not always suitable for audio applications as it may still result in audible glitches.

A simple method is to change the filter coefficients gradually using interpolation between original and desired final coefficients. Experience shows, however, that in order to suppress the transients down to an inaudible level requires very small interpolation steps. In applications of sound synthesis this results in glissando-like sounds which are not always desired.

In this paper we build on the formulation of Zetterberg and Zhang. The basic idea in our own method has been to find a practical way to update the state variables of the allpass filter when the filter coefficients are changed. The main goal is to obtain a solution for transient suppression that gives an acceptable auditory performance at a reasonable implementation complexity.

3.1. Transients Related to Time-Varying Delay

When the desired delay D is changed, the recursive filter enters a transient state. It takes some time before the filter reaches the steady state, and during this time the output signal contains disturbances. The source of the transient problem is in the state variables of the recursive filter.

A direct form II allpass filter can be expressed in the state-variable form as

$$\begin{aligned} \mathbf{v}(n+1) &= \mathbf{F}\mathbf{v}(n) + \mathbf{q}x(n) \\ y(n) &= \mathbf{g}^T\mathbf{v}(n) + g_0 x(n) \end{aligned} \tag{5a}$$

where $x(n)$ and $y(n)$ are the input and output signal of the filter, respectively, and the column vector $\mathbf{v}(n)$ contains the state variables

$$\mathbf{v}(n) = \begin{bmatrix} v_1(n) & v_2(n) & \cdots & v_N(n) \end{bmatrix}^T \tag{5b}$$

and

$$\mathbf{F} = \begin{bmatrix} -a_1 & -a_2 & \cdots & -a_{N-1} & -a_N \\ 1 & 0 & & 0 & 0 \\ 0 & 1 & & & \vdots \\ \vdots & & \ddots & 0 & 0 \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \mathbf{g} = \begin{bmatrix} a_{N-1} \\ a_{N-2} \\ \vdots \\ a_1 \\ 1 \end{bmatrix}, \quad g_0 = a_N \tag{5c}$$

The following derivation is adapted from Zetterberg and Zhang (1988). Let us express the state variable vector as a function of the input signal $x(n)$ and coefficient matrices when the coefficients have been changed at the Cth sample:

$$\mathbf{v}(n) = \begin{cases} \sum_{k=0}^{n-1} \mathbf{F}_1^{n-1-k}\mathbf{q}x(k) + \mathbf{F}_1^n\mathbf{v}(0), \quad 0 < n \le C & (6a) \\ \sum_{k=C}^{n-1} \mathbf{F}_2^{n-1-k}\mathbf{q}x(k) + \mathbf{F}_2^{n-C}\mathbf{v}(C), \quad n > C & (6b) \end{cases}$$

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 time C, respectively. After the coefficients have been changed, the state vector can be expressed as [substituting (6a) into (6b)]

$$\mathbf{v}(n) = \sum_{k=C}^{n-1} \mathbf{F}_2^{n-k-1}\mathbf{q}x(k) + \mathbf{F}_2^{n-C} \sum_{k=0}^{C-1} \mathbf{F}_1^{C-1-k}\mathbf{q}x(k) + \mathbf{F}_1^{n-C}\mathbf{F}_2^{C}\mathbf{v}(0) \tag{7}$$

Here the last term is the contribution of the initial conditions to the state of the filter. We assume that this initial transient has died out so that the last term can be neglected. In addition, grouping the terms differently result in the expression:

$$\mathbf{v}(n) = \sum_{k=0}^{n-1} \mathbf{F}_2^{n-k-1}\mathbf{q}x(k) + \mathbf{F}_2^{n-C}\Delta\mathbf{v}(C) \text{ where } \Delta\mathbf{v}(C) = \sum_{k=0}^{C-1}\left[\left(\mathbf{F}_1^{C-k-1} - \mathbf{F}_2^{C-k-1}\right)\mathbf{q}\right]x(k) \tag{8}$$

The first term in Eq. (8) is the response of the filter to the input after the parameters have changed, and the second term represents the transient due to coefficient change. A way to completely eliminate the transient caused by change of coefficients is *to subtract the term* $\Delta\mathbf{v}(C)$ *from the state vector at time* C (Zetterberg and Zhang 1988).

If it is assumed that the initial transient of the filter has died out, it is equivalent to give the state vector the values $\mathbf{v}_2(C)$ that they would have, were the coefficient matrix $\mathbf{F}_2$ used from the beginning. Then the state vector can be replaced by the following vector

$$\mathbf{v}_2(C) = \sum_{k=0}^{C-1} \mathbf{F}_2^{C-k-1}\mathbf{q}x(k) \tag{9}$$

3.2. A Novel and Efficient Method for Transient Elimination

It is clearly impractical to compute the state vector for a given coefficient set starting from time 0. A key observation is that the filter is affected by the past input values only in proportion to the magnitude of its impulse response. Thus, when we know how long the effective impulse response of the filter is (i.e., how many values of the impulse response are observably nonzero for the application), we can estimate how many past input values need to be taken into account in updating the state variables. Denoting this number of samples by B we can replace expression (9) by

$$\tilde{\mathbf{v}}_2(C) = \sum_{k=C-B}^{C-1} \mathbf{F}_2^{C-k-1}\mathbf{q}x(k) = \sum_{k=1}^{B} \mathbf{F}_2^{k-1}\mathbf{q}x(C-k) \qquad (10)$$

Consider application of this technique to a first-order MF allpass FD filter. As discussed above, the choice of parameter B, i.e., the number of input samples to be taken into account in computing the state vector, depends on how long the transient is and how well it is desired to suppress it. As a measure for the length of the impulse response we define *effective length* as the number of samples which are larger in amplitude than some *threshold* h_{tr}. After this the amplitude can be considered to be negligible for our application. Figure 3 gives two examples of the impulse response of a first-order maximally-flat group delay allpass filter. In this scale the response seems to die out within five or six samples in both cases.

Figure 4 shows the effective length of the impulse response of the a) first-order and b) second-order MF allpass filter as a function of delay D. Here the effective length corresponds to the number of samples whose amplitude is larger than $h_{tr} = 0.01$. After this the amplitude is considered to be small enough so that it is not disturbing. It appears that the length of the impulse response is shortest around $D = N$. Comparison with Fig. 2 shows that the MF allpass filter is also most accurate when its impulse response is short. Thus it may be suggested that the first-order allpass filter is used on the range $0.5 < D \leq 1.5$ and

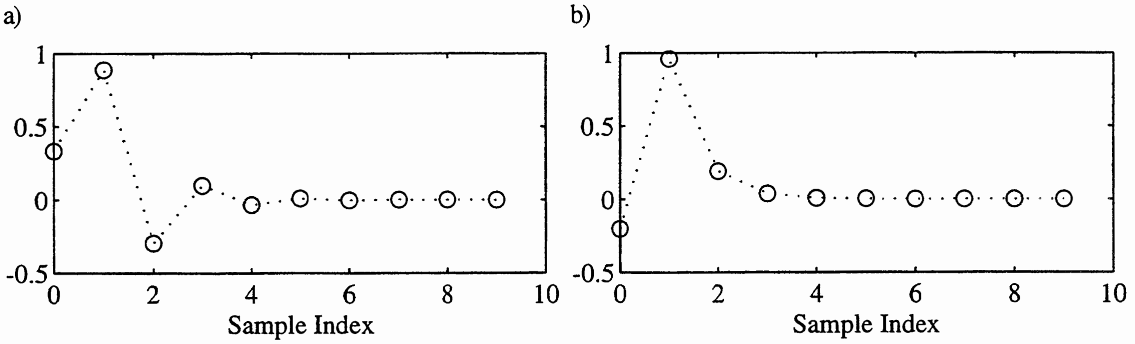

Fig. 3. *The first 10 samples of the impulse response of the first-order allpass filter for delay values a) D = 0.5 and b) D = 1.5.*

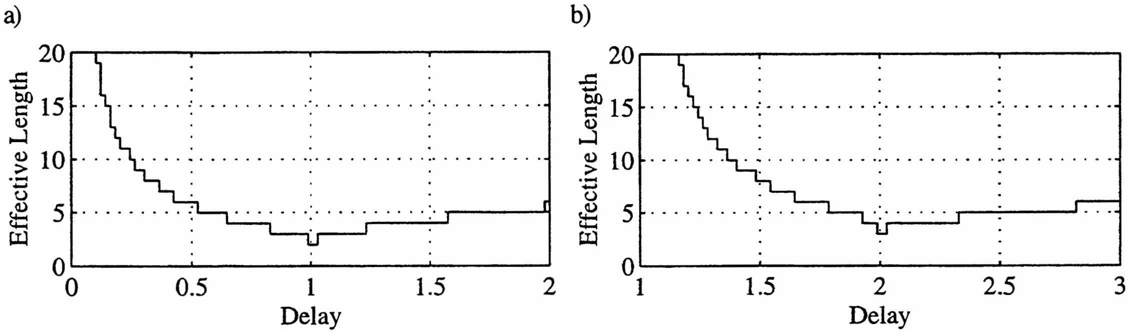

Fig. 4. *The effective length of the impulse response of the a) first-order and b) second-order MF allpass filter as a function delay parameter D. In this figure the effective length is the number of samples whose amplitude is greater than 0.01.*

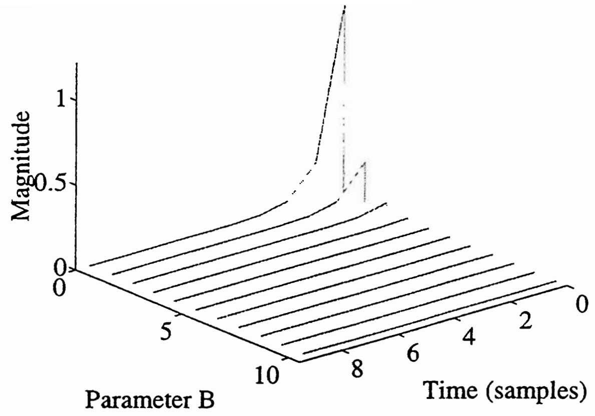

Fig. 5. *The absolute value of the first 10 samples of the transient signal as a function of parameter B. Sinusoidal input signal, first-order allpass FD filter, the coefficients changed such that the approximated nominal delay of the filter steps from 1.5 down to 0.5.*

the second-order filter on the range $1.5 < D \leq 2.5$. In the first-order case it seems to be sufficient to use $B = 6$ in the transient elimination method and $B = 8$ for the second-order filter.

Figure 5 shows the transient as a function of parameter B, assuming that the elimination is carried out using Eq. (10). In this case the input signal is a sine wave and the delay D has been abruptly changed from 1.5 to 0.5 which can be considered the worst case. In the case $B = 0$ the coefficients are changed but no attempt is made to suppress the transients, i.e., the state variables are not modified (nor cleared). When B is one, the transient eliminator is started one sample before the change. When B is increased, it can be seen that the magnitude of the transient gets smaller until it soon becomes negligible. This verifies the effectiveness of our technique and confirms the assumption that only a few past input values are needed for sufficient cancellation.

3.3. Implementation of the New Transient Elimination Method

Let us study the implementation of the variable FD allpass filter in more detail. Figure 6 shows the block diagram of the combination of an allpass filter $A(z)$ and a transient eliminator. Assuming that we use the direct form II structure of the allpass filter, only its recursive part $1/D(z)$ is needed for computing the new state variables. The elimination method works as follows: The input signal is fed to both systems. When the delay parameter D changes, the coefficients of the eliminator are updated immediately. However, the coefficients of the allpass filter are updated only B samples after the change of D, and at the same time the state variables are copied from the transient eliminator to the allpass filter. Note that the output signal of the transient eliminator is not used at all—only its state variables are needed.

The implementation technique of Fig. 6 thus requires that the coefficient update is deferred by B sample cycles and during that time the eliminator is executed in parallel with the allpass FD filter. If only one eliminator is used, it is then possible to change the fractional delay every Bth sample at most. This lag can be compensated if the next delay value D is known beforehand at least B samples before the coefficients change. Then it is not necessary to defer the coefficient update, but instead the eliminator can be started B samples before the actual time of coefficient change.

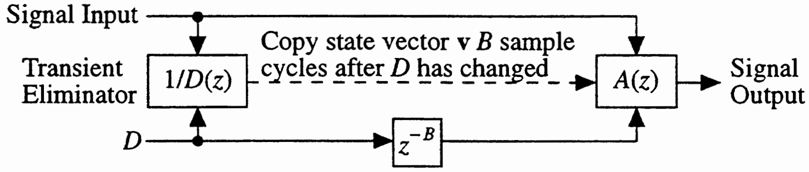

Fig. 6. *A configuration of the allpass FD filter with a transient eliminator. A copy of the recursive part of the allpass filter acts as an eliminator. Its state variables are copied to the allpass filter B samples after the delay parameter D has changed.*

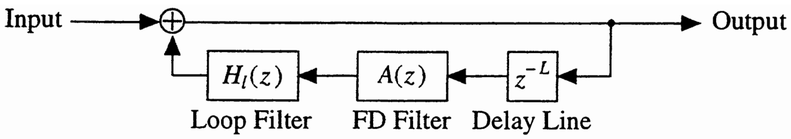

Fig. 7. *A block diagram of a simple waveguide string model. The delay of the feedback loop is determined by the delay line and the fractional delay filter. The loop filter is a low-pass filter that simulates the losses.*

3.4. Delayless Transient Elimination

If immediate coefficient update and transient suppression is required, the above technique is not feasible. However, it can be modified to enable instantaneous update by keeping the last B input samples in a buffer memory and executing the necessary computations in a single sample interval by running the eliminator at B times the sampling rate of the overall system. This naturally requires more computing power than the previous method. It is, however, our experience that, when the transients are eliminated, a coefficient change rate of once every 6-10 samples is usually quite sufficient.

4. APPLICATION TO DIGITAL WAVEGUIDE MODELING

Digital waveguide modeling (Smith 1992) is an audio DSP methodology that is widely used for sound synthesis based on physical models of musical instruments. In digital waveguide models, the wave propagation is simulated using delay lines. The length of the delay line corresponds to the propagation delay of sound waves in the medium. In physical models of string and wind instruments, the length of the delay line determines the pitch of the musical sounds. It is not accurate enough to represent this delay as an integer multiple of the sampling interval. Usually a fractional delay filter is used to correct the difference between the desired delay length and the nearest multiple of the sample interval.

Figure 7 depicts the block diagram of a simple waveguide string model (a generalization of the Karplus–Strong model, see, e.g., Jaffe and Smith (1983) or Välimäki *et al.* (1995)). The length of the string is determined by a delay line and a fractional delay filter. The losses have been lumped into a single lowpass filter (called the loop filter) which can be a simple one-pole digital filter.

Let us look into the details of modeling the variable delay block which is our main concern here. As fractional delay filters (FIR or allpass) can provide good approximation only in a finite delay range, the usual solution is to vary the delay of the FD filter only within 1 sample interval and add unit delay elements as needed. This introduces a further problem for transient elimination, namely the case when one unit delay is added to or deleted from the system. This can be done (and has been done), but the details are however beyond this paper. This problem has earlier been discussed by Smith (1991).

To test the transient elimination technique, the model in Fig. 7 has been implemented with a time-varying delay to generate a glissando tone. In this example the transfer function of the one-pole loop filter is $H_l(z) = 0.965 / (1 - 0.03z^{-1})$ and the input signal is a Hamming pulse. The delay increases from 20 to 30 units in steps of one half a delay every 25 output samples. Consequently the generated tone falls in pitch. Figure 8a shows the generated signal without transient elimination. The circles indicate the points at which the delay is increased and the filter parameters are changed. The unwanted transients can clearly be seen. Contrast this with Fig. 8b which shows the same glissando tone, but with the transient eliminator in action. An obvious improvement in the quality of output can be seen.

5. CONCLUSIONS

A novel and efficient transient elimination technique for time-varying fractional delay allpass filters has been presented. The technique is based on a method of state variable update at the time of the filter coefficient change. It was shown that only a few previous values of the input signal are required as input to a parallel filter to eliminate the transients so that they become inaudible. This technique can be used to improve the sound quality in digital waveguide models of musical instruments.

The proposed technique can also be used for suppressing transients of other time-varying IIR filters, since it does not in principle assume that the filter is an allpass one. We believe that this method has several potential applications in the field of audio signal processing.

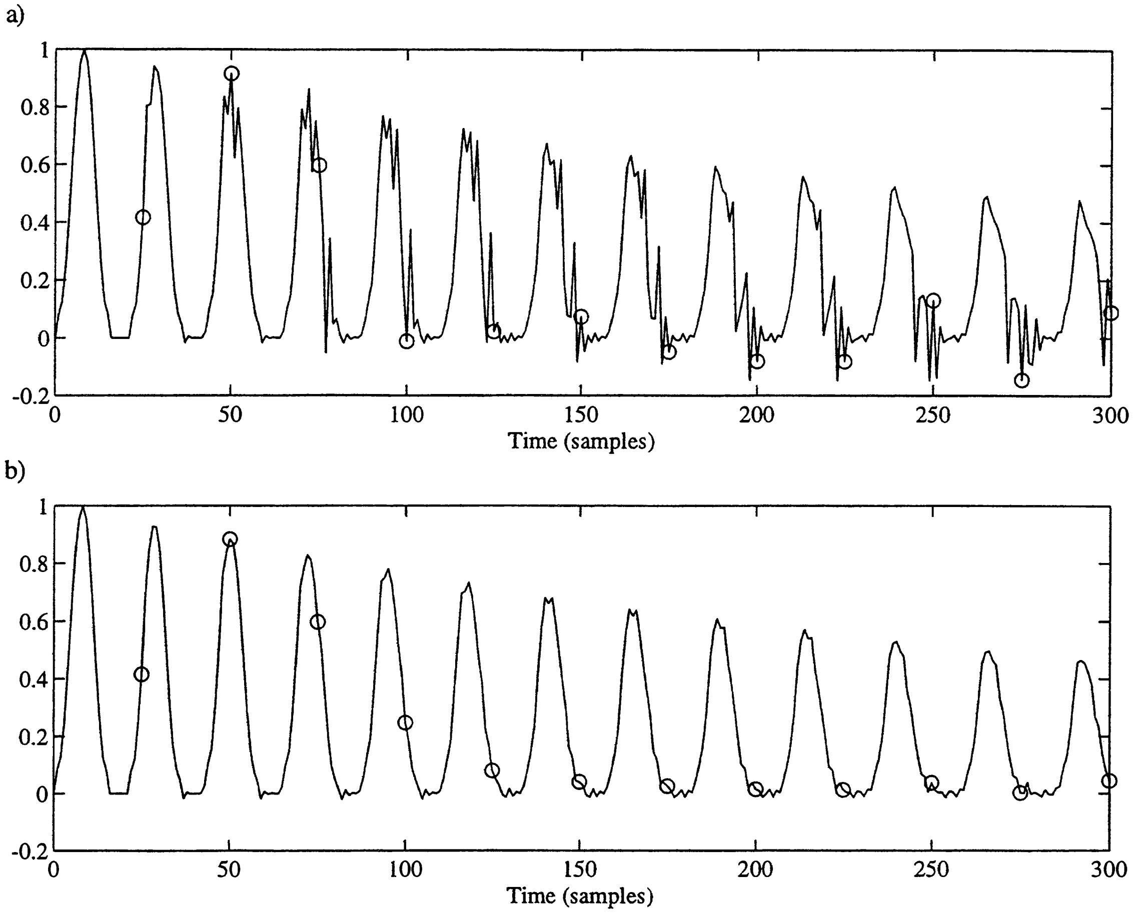

Fig. 8. *Output signals for the model in Fig. 7 with a stepped increase in the loop delay. a) The result without transient elimination and b) with the transient eliminator.*

REFERENCES

Jaffe, D., and J. O. Smith. 1983. "Extensions of the Karplus-Strong plucked string algorithm." *Computer Music Journal* 7(2): 56–69. Reprinted in Curtis Roads (ed.) 1989. *The Music Machine*. Cambridge, Massachusetts: MIT Press, pp. 481–494.

Laakso, T. I., V. Välimäki, M. Karjalainen, and U. K. Laine. 1992. "Real-time implementation techniques for a continuously variable digital delay in modeling musical instruments." *Proceedings of the International Computer Music Conference*, San Jose, CA, pp. 140–141.

Laakso, T. I., V. Välimäki, M. Karjalainen, and U. K. Laine. 1994. *Crushing the Delay—Tools for Fractional Delay Filter Design*. Espoo, Finland: Helsinki University of Technology, Laboratory of Acoustics and Audio Signal Processing, Report no. 35. Submitted to the *IEEE Signal Processing Magazine*.

Rabenstein, R. 1988. "Minimization of transient signals in recursive time-varying digital filters." *Circuits, Systems, and Signal Processing* 7(3): 345–359.

Smith, J. O. 1991. "J. O. Smith III comments on Sullivan Karplus-Strong article." *C. M. J.* 15(2): 10–11.

Smith, J. O. 1992. "Physical modeling using digital waveguides." *Computer Music Journal* 16(4): 75–87.

Thiran, J.-P. 1971. "Recursive digital filters with maximally flat group delay." *IEEE Trans. Circuit Theory* 18(6): pp. 659–664.

Välimäki, V. 1994. *Fractional Delay Waveguide Modeling of Acoustic Tubes*. Espoo, Finland: Helsinki University of Technology, Laboratory of Acoustics and Audio Signal Processing, Report no. 34.

Välimäki, V., J. Huopaniemi, M. Karjalainen, and Z. Jánosy. 1995. "Physical modeling of plucked string instruments with application to real-time sound synthesis." Presented at the *AES 98th Convention*, Paris, France, preprint no. 3956.

Zetterberg, L. H., and Q. Zhang. 1988. "Elimination of transients in adaptive filters with application to speech coding." *Signal Processing* 15(4): 419–428.