EFFICIENT COMPUTATION OF NONLINEAR FILTER NETWORKS WITH DELAY-FREE LOOPS AND APPLICATIONS TO PHYSICALLY-BASED SOUND MODELS

Federico Avanzini

Dept. of Information Engin.
University of Padova
Via Gradenigo 6/A,
35131 - Padova, Italy
avanzini@dei.unipd.it

Federico Fontana, Davide Rocchesso

Dept. of Computer Science
University of Verona,
Strada Le Grazie, 15
37134 - Verona, Italy
{fontana,rocchesso}@sci.univr.it

ABSTRACT

The paper presents a general procedure for the computation of filter networks made of linear filters and non-linear non-algebraic (dynamic) elements. The method is developed in the Kirchhoff domain and applies to cases where the network contains an arbitrary number of delay-free paths that involve nonlinear elements. Compared to existing techniques the method does not require a rearrangement of the network structure, instead it subdivides the network into computational substructures specified by appropriate matrices related to the network topology. Sufficient conditions are discussed for the applicability of the method, and results are provided that relate performance of the method to the properties of the nonlinear elements and to the network topology. The last part of the paper discusses applications of the method to the simulation of acoustic systems, including multidimensional wave propagation by means of waveguide-mesh techniques.

1. INTRODUCTION

The *delay-free loop problem* [1, sec. 6.1.3] refers to the presence in a network of feedback paths that are not computable, meaning with this that the computation cannot be executed sequentially due to the lack of pure delays along the loop. This problem can appear in particular during conversion to the digital domain of analog filter networks, or even in digital-to-digital domain transformations (such as frequency-warping mappings).

If the network is linear, various techniques can be used to convert a continuous-time system into an equivalent numerical one, working either in the time or in the Laplace domains. As an example, wave methods [2] and transfer function models [3] have been widely applied to the numerical simulation of acoustic systems. Moreover, a linear network can be always rearranged into a new one in which delay-free paths are solved by composing the filters belonging to them into bigger linear structures that "embed" the loop [1]. Nevertheless there are cases where this rearrangement is deprecated (e.g., situations in which the access to the filter parameters becomes too complicated after the rearrangement). Furthermore, the elimination of a delay-free path implies that all the branches belonging to it cannot be used any longer as input/output points where to inject/extract the signal to/from the system: this point is particularly relevant in the design of virtual musical instruments by physical modeling.

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

This paper presents a general method that enables to model a network of one-dimensional nonlinear and linear blocks, even in presence of delay-free paths. The network structure introduced in section 2 assumes that each block has been already modeled in the discrete-time domain. Moreover, we do not address stability issues. Section 3 discusses the case when nonlinear blocks are part of a delay-free loop, and provide a procedure to compute those loops without rearranging them into a different topology, thus preserving their original position in the network in terms of input/output mutual relations. Strategies for the efficient computation on nonlinear blocks are addressed in section 4. Finally in section 5 we present a few applications of the procedure to the simulation of acoustic systems, that are currently under development.

2. STRUCTURE OF THE NETWORK

A technique to compute linear delay-free paths without topology rearrangement was proposed in [4]. It was applied to warped IIR filter computation [5], and to magnitude-complementary parametric equalizers [6], and generalized to linear filter networks with arbitrary delay-free path configurations [7]. It was then extended to networks containing nonlinear blocks [8].

2.1. Linear and nonlinear blocks

The network structure addressed in [8] and in the remainder of this paper comprises m linear and nonlinear blocks. Inputs and outputs to and from each block are indicated with x and y variables, respectively. The $m_L \leq m$ linear blocks are defined as

$$y_i[n] = b_i x_i[n] + q_i[n], \qquad i = 1, \ldots, m_L \quad \text{(1a)}$$

$$q_i[n] = \sum_{k=1}^{Z_i} b_{k,i} x_i[n-k] + \sum_{k=1}^{P_i} a_{k,i} y[n-k], \quad \text{(1b)}$$

where we have assumed the ith block to have a transfer function $H_i(z) = \sum_{k=0}^{Z_i} b_{k,i} z^{-k} / (1 - \sum_{k=1}^{P_i} a_{k,i} z^{-k})$, have defined an *historical* component q_i that collects all past components, and have defined $b_i := b_{0,i}$.

Similarly, the $m_N = m - m_L$ nonlinear blocks are specified by their discrete-time transfer characteristic:

$$y_i[n] = f_i(x_i[n], p_i[n]), \quad i = m_L + 1, \ldots, m \quad \text{(2a)}$$

$$p_i[n] = p_i(x_i[n-1], y_i[n-1], p_i[n-1], \ldots). \quad \text{(2b)}$$

Equations (2) imply that nonlinear blocks respect two hypotheses: first, every block admits the existence of a transfer function f_i in the form (2a); second, the blocks represent non-algebraic (dynamic) elements in which p_i contains the contribution of historical components in the function. The nonlinearity can then be evaluated for past input and output values, thus obtaining a new function in the single variable x_i. In practice this class of nonlinear functions is sufficiently expressive for a wide range of audio applications [9, 10, 11].

2.2. Connections

The input x_i to the ith (linear or nonlinear) block is assumed to be a linear combination of R_i outputs from other blocks, possibly with the addition of an external input signal u_i to the same block:

$$x_i[n] = \sum_{k=1}^{R_i} y_{i_k}[n] + u_i[n], \quad i = 1, \ldots, m. \quad (3)$$

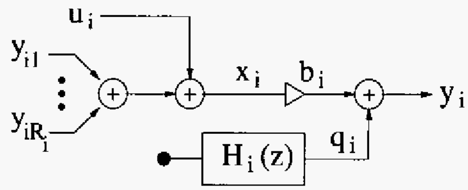

Figure 1: *Structure of a linear block. The output y_i is a superposition of instantaneous and historical components x_i, q_i. The input is a linear combination of outputs from other branches and the external input u_i.*

Additionally we assume that $i_k \neq i$ for $k = 1, \ldots, R_i$ and for any i. In other words, no direct connection between y_i and x_i is allowed. Note that this requirement can be always satisfied by inserting a "dummy" linear block $y_i = x_i$ (i.e., $H_i(z) \equiv 1$) in the network.

Equations (3) define the network topology. Figure 1 depicts the input-output configuration for a linear block: the historical component q_i can be computed by feeding the filter with a null value [4]. The situation is more complicated for the nonlinear blocks, since the scheme depicted in figure 1 becomes computable as long as p_i is known by (2b).

3. SOLUTION OF THE SYSTEM

3.1. Matrix formulation

Equations (1a,2a,3) can be rewritten in matrix form as

$$\begin{aligned} \boldsymbol{y}_N[n] &= \boldsymbol{f}(\boldsymbol{x}_N[n], \boldsymbol{p}[n]), & \text{(4a)} \\ \boldsymbol{y}_L[n] &= \boldsymbol{B}\boldsymbol{x}_L[n] + \boldsymbol{q}[n], & \text{(4b)} \\ \boldsymbol{x}[n] &= \boldsymbol{C}\boldsymbol{y}[n] + \boldsymbol{u}[n], & \text{(4c)} \end{aligned}$$

where column vectors $\boldsymbol{x}_{N,L}$, $\boldsymbol{y}_{N,L}$, $\boldsymbol{p}$, $\boldsymbol{q}$, $\boldsymbol{u}$ collect the corresponding components, and $\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{f}$ are defined as

$$\boldsymbol{y} = \begin{vmatrix} \boldsymbol{y}_N \\ \boldsymbol{y}_L \end{vmatrix}, \qquad \boldsymbol{x} = \begin{vmatrix} \boldsymbol{x}_N \\ \boldsymbol{x}_L \end{vmatrix}$$

$$\boldsymbol{f}(\boldsymbol{x}_N, \boldsymbol{p}) = \begin{vmatrix} f_1(x_1, p_1) \\ \cdots \\ f_{m_N}(x_{m_N}, p_{m_N}) \end{vmatrix}.$$

$\boldsymbol{B}$ is a diagonal matrix containing the linear coefficients $b_1, \ldots, b_{m_L}$ and $\boldsymbol{C}$ accounts for connections: $c_{ij} = 1$ if the output from the jth block is connected to the input to the ith block, otherwise $c_{ij} = 0$. The assumption of no direct paths between y_i and x_i in (3) translates into the property $c_{ii} = 0$ for $i = 1, \ldots, m$.

The matrix $\boldsymbol{C}$ can be split into four sub-matrices $\boldsymbol{C}_{NN,NL,LN,LL}$ that account for nonlinear-to-nonlinear,

linear-to-nonlinear, nonlinear-to-linear, and linear-to-linear connections, respectively:

$$\begin{vmatrix} \boldsymbol{x}_N \\ \boldsymbol{x}_L \end{vmatrix} = \begin{vmatrix} \boldsymbol{C}_{NN} & \boldsymbol{C}_{NL} \\ \boldsymbol{C}_{LN} & \boldsymbol{C}_{LL} \end{vmatrix} \begin{vmatrix} \boldsymbol{y}_N \\ \boldsymbol{y}_L \end{vmatrix} + \begin{vmatrix} \boldsymbol{u}_N \\ \boldsymbol{u}_L \end{vmatrix} , \quad (5)$$

where we have also split the vector $\boldsymbol{u}$ into $\boldsymbol{u}_N$ and $\boldsymbol{u}_L$.

We have shown in [8] that substitution of (4) in (5), plus some straightforward algebra, leads to the following expression for the linear-block inputs:

$$\boldsymbol{x}_L = \boldsymbol{F}_{LL}^{-1}\boldsymbol{C}_{LN}\boldsymbol{f}(\boldsymbol{x}_N, \boldsymbol{p}) + \boldsymbol{F}_{LL}^{-1}(\boldsymbol{C}_{LL}\boldsymbol{q} + \boldsymbol{u}_L), \quad (6)$$

where we have defined the matrix $\boldsymbol{F}_{LL} := \boldsymbol{I} - \boldsymbol{C}_{LL}\boldsymbol{B}$ and have assumed that it is invertible. This assumption is not restrictive, since it can be shown that $\boldsymbol{F}_{LL}^{-1}$ exists if the linear part of the network is causal [7].

A few more algebraic manipulations lead to the following expression for the nonlinear-block inputs:

$$\boldsymbol{x}_N = \boldsymbol{W}_1\boldsymbol{f}(\boldsymbol{x}_N, \boldsymbol{p}) + \boldsymbol{W}_2\boldsymbol{q} + \boldsymbol{W}_3\boldsymbol{u}_L + \boldsymbol{u}_N, \quad (7)$$

where the $\boldsymbol{W}_i$ matrices are defined as follows:

$$\begin{aligned} \boldsymbol{W}_3 &= \boldsymbol{C}_{NL}\boldsymbol{B}\boldsymbol{F}_{LL}^{-1} && \text{(8a)} \\ \boldsymbol{W}_1 &= \boldsymbol{W}_3\boldsymbol{C}_{LN} + \boldsymbol{C}_{NN} && \text{(8b)} \\ \boldsymbol{W}_2 &= \boldsymbol{W}_3\boldsymbol{C}_{LL} + \boldsymbol{C}_{NL} && \text{(8c)} \end{aligned}$$

Note that the only unknown in (7) is $\boldsymbol{x}_N[n]$. Note also that if the network contains a delay-free computational loop and $\boldsymbol{W}_1 \neq 0$, then (7) defines $\boldsymbol{x}_N[n]$ implicitly.

3.2. Computation of the network

Equation (7) defines the inputs $\boldsymbol{x}_N[n]$ to the nonlinear blocks in terms of known quantities: the historical components $\boldsymbol{q}[n]$ and the external inputs $\boldsymbol{u}_L[n], \boldsymbol{u}_N[n]$. In addition the matrix $\boldsymbol{W}_1$ isolates the instantaneous dependence of $\boldsymbol{x}_N[n]$ on $\boldsymbol{y}_N[n]$.

From equations (4a) and (7), one can write

$$\boldsymbol{y}_N[n] = \boldsymbol{f}(\boldsymbol{W}_1\boldsymbol{y}_N[n] + \tilde{\boldsymbol{x}}_N[n], \boldsymbol{p}), \quad (9)$$

where $\tilde{\boldsymbol{x}}_N[n] = \boldsymbol{W}_2\boldsymbol{q}[n] + \boldsymbol{W}_3\boldsymbol{u}_L[n] + \boldsymbol{u}_N[n]$ is again a historical component, since it collects the contribution of known quantities to the input $\boldsymbol{x}_N$. Note that the only unknown in (9) is $\boldsymbol{y}_N[n]$. Similarly to equation (7), if the network contains a delay-free loop and $\boldsymbol{W}_1 \neq 0$ then (9) defines $\boldsymbol{y}_N[n]$ implicitly.

We have shown in [8] that the network is computable if $\boldsymbol{y}_N[n]$ can be computed from (9). More precisely, the computation can be decomposed into the following steps (refer also to figure 2):

1. $\boldsymbol{x}_N[n]$ and $\boldsymbol{y}_N[n]$ are computed from (7) and (9) using external inputs $\boldsymbol{u}[n]$ and historical components $\boldsymbol{p}[n], \boldsymbol{q}[n]$;
2. $\boldsymbol{x}_L[n]$ is computed from (6) and $\boldsymbol{y}_L[n]$ is computed from (4b);
3. $\boldsymbol{p}[n+1]$ is computed from (2b) using known variables (as already mentioned, we do not investigate particular forms that this equation takes);
4. $\boldsymbol{q}[n+1]$ is computed from (1b) or, equivalently, by feeding each filter with a null signal [4]. Note that no computation is needed if the filters are realized in transposed direct form [1, 7].

4. COMPUTATION OF NONLINEAR BLOCKS

We remark once more that in order to apply the computational scheme outlined above one must first compute the outputs $\boldsymbol{y}_N[n]$ from the nonlinear blocks.

4.1. Newton-Raphson iteration

A strategy for solving equation (9) was proposed in [8] and amounts to applying the Newton-Raphson (NR) method to find $\boldsymbol{y}_N[n]$ iteratively. This has some similarity to what Borin *et al.* [12] have proposed. Note however that, unlike the formulation given in [12], the nonlinearity in (9) has memory due to the presence of the historical components $\boldsymbol{p}$. The NR algorithm [13] search a local zero of the function

$$\boldsymbol{g}_{\boldsymbol{p}}(\boldsymbol{y}_N) = \boldsymbol{f}(\boldsymbol{W}_1\boldsymbol{y}_N + \tilde{\boldsymbol{x}}_N, \boldsymbol{p}) - \boldsymbol{y}_N \quad (10)$$

A pseudocode description of the algorithm looks like the one given in figure 3, where $\boldsymbol{J}_k = \left[\frac{\partial(\boldsymbol{g}_{\boldsymbol{p}})_i}{\partial(\boldsymbol{y}_N)_j}\right]_{\boldsymbol{y}_{Nk}}$ is the jacobian of $\boldsymbol{g}_{\boldsymbol{p}}$ evaluated in $\boldsymbol{y}_{Nk}$.

4.2. Fixed-point iteration

We propose here a different approach to the solution of equation (9), which is based on fixed-point (FP) iteration [13]. We recall that a pseudocode description of the algorithm looks like the one given in figure 3(b), where this time the function $\boldsymbol{g}_{\boldsymbol{p}}$ has been defined as

$$\boldsymbol{g}_{\boldsymbol{p}}(\boldsymbol{y}_N) = \boldsymbol{f}(\boldsymbol{W}_1\boldsymbol{y}_N + \tilde{\boldsymbol{x}}_N, \boldsymbol{p}). \quad (11)$$

The problem in using FP iteration is that convergence is ensured only if the nonlinear function $\boldsymbol{g}_{\boldsymbol{p}}$ satisfies more restrictive hypothesis. Namely, $\boldsymbol{g}_{\boldsymbol{p}}$ must possess a "small" Lipschitz constant:

$$\|\boldsymbol{g}_{\boldsymbol{p}}(\boldsymbol{y}) - \boldsymbol{g}_{\boldsymbol{p}}(\boldsymbol{y}^*)\| \leq M_p\|\boldsymbol{y} - \boldsymbol{y}^*\|, \quad (12)$$

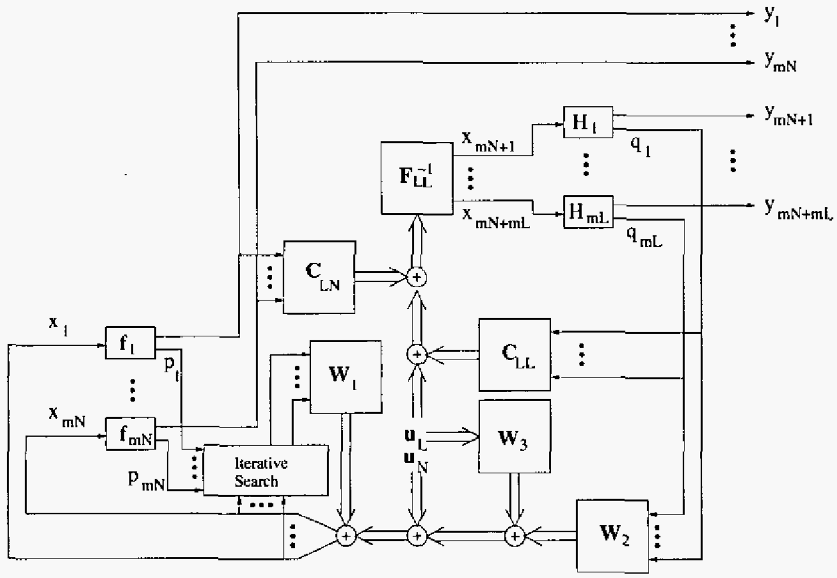

Figure 2: *Schematic of the proposed method.*

with $0 \leq M_p < 1$. We then look for an estimate of M. If $f(\cdot, p)$ has a Lipschitz constant $\bar{M}_p$, then

$$\begin{aligned} &\|g_p(y) - g_p(y*)\| = \\ &\|f(W_1 y + \tilde{x}_N, p) - f(W_1 y^* + \tilde{x}_N, p)\| \leq \\ &\bar{M}_p \|W_1(y - y^*)\| \leq \bar{M}_p \|W_1\| \cdot \|y - y^*\|. \end{aligned} \quad (13)$$

Therefore g_p has a Lipschitz constant $M_p = \bar{M}_p \|W_1\|$. Moreover, recalling equations (8a,8b), and assuming that $C_{NN} = 0$ in (8b)[1] one can write

$$M_p \leq \bar{M}_p \|C_{NL}\| \cdot \|B\| \cdot \|F_{LL}^{-1}\| \cdot \|C_{LN}\|. \quad (14)$$

Providing a quantitative estimate for M from (14) is not trivial, and specifically $\|F_{LL}^{-1}\|$ is not easily estimated. Here we only note that the property

$$M_p \to 0 \quad \text{for} \quad \max_i b_i \to 0 \quad (15)$$

holds, because $\|B\| \to 0$, and moreover $\|F_{LL}^{-1}\| \to 1$ because F_{LL}^{-1} is a perturbation of the identity matrix.

4.3. Discussion

Property (14) has a straightforward interpretation: FP iteration improves as the weights of the instantaneous contributions decrease. If the filters $H_i(z)$ have been obtained by discretizing a continuous-time system with sampling rate F_s, then in general $b_i = \mathcal{O}(F_s^{-n})$, where n depends on the order of the discretization method. Therefore property (14) ensures that FP iteration can always be used if the sampling rate is large enough.

Using FP rather than NR iteration is clearly advantageous in many respects. First, NR is far more expensive computationally: it requires at each iteration one evaluation of g_p together with its $m_N \times m_N$ jacobian, plus inversion of the jacobian, while FP only requires one evaluation of g_p per iteration. Second, NR is a local method and therefore convergence is not ensured *a priori*, while FP iteration converges globally on the interval where condition (14) holds.

5. APPLICATIONS

5.1. Modal synthesis

There has recently been growing interest in modeling vibrations of an elastic medium (e.g., a string) under large amplitude conditions [14, 15], where tension modulation effects as well as coupling between transverse and longitudinal modes must be considered. In particular, it is known that if one neglects transverse-to-longitudinal coupling effects, then transverse motion in a single polarization with tension modulation can be described by a nonlinear PDE introduced by Carrier, which admits a modal representation:

$$\ddot{x}_j = -g\dot{x}_j - (k_{0j} + k_{1j}\|x\|)x_j, \quad j = 1 \ldots \infty, \quad (16)$$

[1] This assumption is not restrictive: if $C_{NN} \neq 0$ one can find an equivalent network with $C_{NN} = 0$, by inserting "dummy" linear elements $y_i = x_i$ between nonlinear-to-nonlinear connections.

```
y_N0 = y_N[n - 1];  k = 0
while (err > Errmax)
      Compute  g_p(y_Nk) from Eq.  (10)
      Compute  y_N(k+1) = y_Nk - J_k^{-1} · g_p(y_Nk)
      Compute  err = abs(y_N(k+1) - y_Nk)
      k = k + 1
end
y_N[n] = y_Nk
```

(a)

```
y_N0 = y_N[n - 1];  k = 0
while (err > Errmax)
      Compute  y_N(k+1) = g_p(y_Nk)
      Compute  err = abs(y_N(k+1) - y_Nk)
      k = k + 1
end
y_N[n] = y_Nk
```

(b)

Figure 3: *Pseudocode realizations of (a) NR iteration and (b) FP iteration. The function g_p is defined as in equations (10) and (11), respectively.*

where $\|x\|$ is the L^2-norm of the modal displacement vector (see [16] for details). Such a system is well suited to be realized in the discrete-time domain using the technique described here. Linear blocks represent the modes as 2nd order resonators, while suitable nonlinear blocks account for the tension modulation effect represented by the term $\|x\|$. Additionally, nonlinear excitation mechanisms, such as striking or bowing, can be naturally embedded into the computational structure, by adding appropriate nonlinear blocks [17].

5.2. Digital Waveguides

Digital Waveguide Networks [18, 2] model ideal n-D wave propagation along a medium, and can be seen as linear filter networks in which adjacent scattering nodes are arranged in a grid and connected to each other via bidirectional unitary delay lines. At each time step, incoming signals at a node are instantaneously scattered to outgoing signals, that reach adjacent nodes at the next time step.

It is known that 2- and 3-D waveguide mesh structures introduce a dispersion error which causes resonance misplacement. A way of attenuating this error amounts to transforming pure delays into frequency-dependent phase shifts ($z^{-1} = A(\tilde{z})$, where A is an allpass filter) in a way that the new filter blocks introduce the desired delay to any frequency component. Such a "warped" version of the 2-D triangular Waveguide Mesh (TWM) has been shown [19] to improve accuracy considerably.

In [7] the technique described here has been applied to the computation of the warped TWM. The system addressed in [7] is a special case of (4) in that it does not include nonlinear elements and delay-free loops occur between linear blocks, however nonlinear elements can be included in the system in order to account for many effects. As an example, kettledrum simulation has been obtained by coupling a kettle model with a TWM membrane model, by means of loaded scattering junctions [20] that include a spring-mass system accounting for the frequency-varying air density which aligns resonances into an harmonic series in kettledrums [3]. Such a model can be considerably improved by substituting the simple (linear) air density function previously used in those models with a more realistic, nonlinear law.

6. CONCLUSION

We have presented a general procedure for the computation of a class of networks composed by nonlinear and linear blocks that satisfy weak hypotheses. The proposed solution does not require any network rearrangement and preserves the original topology. Each computational block can therefore be modeled independently. Then, provided that a connection topology is specified, the global computational structure for the complete system is constructed automatically using the procedure given in section 2.

We have focused on the nonlinear blocks, and proposed a scheme based on fixed-point iteration that allows efficient computation of the nonlinearities, thus showing that the network can be computed even in the presence of delay-free paths that involve nonlinear blocks. When applied to physical models of acoustic systems, the proposed procedure allows for a highly modular formulation in which each computational block has a clear physical meaning.

In the context of this research, the most desirable feature of this methodology would be to establish general criteria to guarantee the preservation of the structural properties of the system while moving from the analog to the digital domain, in a way that at the end of such a translation each filter block has a clear physical meaning. We are currently working on extending the scope of the method in this direction.

7. REFERENCES

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

[2] S. Bilbao, *Wave and Scattering Methods for the Numerical Integration of Partial Differential Equations*, Ph.D. thesis, CCRMA, Stanford University, Stanford (CA), May 2001, available at http://ccrma-www.stanford.edu/~bilbao/.

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

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

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

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

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

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

[9] G. Borin and G. De Poli, "A Hysteretic Hammer-String Interaction Model for Physical Model Synthesis," in *Proc. Nordic Acoustical Meeting (NAM-96)*, Helsinki, Finland, June 1996, pp. 399–406.

[10] F. Avanzini and D. Rocchesso, "Efficiency, accuracy, and stability issues in discrete time simulations of single reed instruments," *J. of the Acoustical Society of America*, vol. 111, no. 5, pp. 2293–2301, May 2002.

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

[12] G. Borin, G. De Poli, and D. Rocchesso, "Elimination of delay-free loops in discrete-time models of nonlinear acoustic systems," *IEEE Trans. on Speech and Audio Processing*, vol. 8, no. 5, pp. 597–605, 2000.

[13] J. D. Lambert, *Numerical Methods for Ordinary Differential Systems*, John Wiley & Sons, 1991.

[14] T. Tolonen, V. Välimäki, and M. Karjalainen, "Modeling of Tension Modulation Nonlinearity in plucked strings," *IEEE Trans. on Speech and Audio Processing*, vol. 8, no. 3, pp. 300–310, May 2000.

[15] S. Bilbao, "Modal-type synthesis techniques for nonlinear strings with an energy conservation property," in *Proc. Conf. on Digital Audio Effects (DAFX-04)*, Naples, Italy, 2004, pp. 119–124.

[16] R. W. Dickey, "Infinite systems of nonlinear oscillation equations with linear damping," *SIAM J. Appl. Math.*, vol. 19, no. 1, pp. 208–214, July 1970.

[17] F. Avanzini, M. Rath, D. Rocchesso, and L. Ottaviani, "Low-level models: resonators, interactions, surface textures," in *The Sounding Object*, D. Rocchesso and F. Fontana, Eds., pp. 137–172. Edizioni Mondo Estremo, Florence, Italy, 2003.

[18] S. A. Van Duyne and J. O. Smith, "Physical modeling with the 2-D digital waveguide mesh," in *Proc. Int. Computer Music Conf.*, Tokyo, Japan, 1993, ICMA, pp. 40–47.

[19] L. Savioja and V. Välimäki, "Reduction of the dispersion error in the triangular digital waveguide mesh using frequency warping," *IEEE Signal Processing Letters*, vol. 6, no. 3, pp. 58–60, Mar. 1999.

[20] F. Fontana and D. Rocchesso, "Physical modeling of membranes for percussion instruments," *Acustica*, vol. 83, no. 1, pp. 529–542, Jan. 1998.