

## COMPUTATION OF NONLINEAR FILTER NETWORKS CONTAINING DELAY-FREE PATHS

*Federico Fontana, Davide Rocchesso*

Dept. of Computer Science  
University of Verona  
Strada le Grazie 15, 37134 Verona, Italy  
fontana@sci.univr.it  
davide.rocchesso@univr.it

*Federico Avanzini*

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

### ABSTRACT

A method for solving filter networks made of linear and nonlinear filters is presented. The method is valid independently of the presence of delay-free paths in the network, provided that the nonlinearities in the system respect certain (weak) hypotheses verified by a wide class of real components: in particular, that the contribution to the output due to the memory of the nonlinear blocks can be extracted from each nonlinearity separately. The method translates into a general procedure for computing the filter network, hence it can serve as a testbed for offline testing of complex audio systems and as a starting point toward further code optimizations aimed at achieving real time.

### 1. INTRODUCTION

The history of audio effects design traces back to the world of analog circuits, perhaps even before [1, 2]. It was not long after the digital architectures had appeared, that scientists considered the possibility to reproduce in the digital domain the analog and electro-acoustic mechanisms the early audio effects were based upon: at first with the aim of improving their quality by modeling them through digital filters; later on, with the goal of simulating these effects on digital equipment ranging from specialized hardware available in the studio to more common PC's, audio cards and other consumer electronic devices.

The conversion into a sequence of discrete computations of a continuous-time process, such as the one realized by an analog electronic, electro-acoustic, fluid-dynamic or mechanical system, must move through a number of steps each one potentially introducing approximations that progressively shift the model away from the original system. Sometimes such approximations introduce negligible effects, if any at all. Conversely, there are cases in which they generate intolerable problems, often resulting in heavy artifacts affecting the system response or, even, preventing the model to work correctly.

The theory of digital signal processing suggests how a discrete-time system can be realized in the form of a network of filters, suitable for implementation on a real-time processor. In the most simple case these filters are linear, furthermore they all run at one single sampling frequency over networks that do not contain delay-free paths [3].

Various methods taken from the numerical analysis allow to represent continuous-time systems in terms of networks made of individual interconnected processing blocks [4, 5], that have a di-

rect counterpart in a corresponding linear digital filter network. Alternatively, analog transfer characteristics of linear elasto-mechanic and fluid-dynamic systems can be computed in the Laplace domain [6]. From there, several techniques can be invoked to transform the analog characteristics into corresponding discrete-time transfer functions and, finally, into a digital filter network [7, 8].

In principle the discretization of an analog system does not prevent the final digital filter network to contain delay-free paths. In the linear case such a network can be always rearranged into a new filter network in which the delay-free paths have been formally solved by composing the filters belonging to them into bigger linear structures that "embed" the loop [3]. Nevertheless there are cases where this rearrangement is deprecated. These include 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 problem is particularly relevant in the design of virtual musical instruments by physical modeling [9, 10].

The analog-to-digital conversion becomes even more complicated when one or more nonlinearities exist in the continuous-time system. Linearizing the system before the conversion is often forbidden since the nonlinearity adds unique features, impossible to reproduce with a linear behavior. As long as nonlinearities are involved in the conversion new problems come into play, the most important of which is certainly the preservation of the system stability along with a precise simulation of the nonlinear characteristic [11, 12, 13, 14]. One further complication arises if a nonlinearity is part of a delay-free path: in this case there is no general procedure to rearrange the transfer functions of the loop to come up with a new linear structure in which to embed the delay-free path.

Moved by the idea of defining a general method for the representation of nonlinear filter networks, we have pointed out a strategy that in principle enables to model a network of (one-dimensional, although the method can be extended to the multi-dimensional case) nonlinear and linear filter realizations, even in presence of delay-free paths. Here we will assume that each filter block—both linear and nonlinear—has been already precisely modeled, and it is ready to be included in the network.

We also implicitly assume the stability of the resulting system. This assumption, because the method does not deal with the analog-to-digital conversion problem and the related stability issues [19, 4, 14]. Instead, it focuses on the way linear and nonlinear

filters can be connected each other from a purely structural view-point, *preserving their original position in the network in terms of input/output mutual relations*. In particular we will deal with the case when nonlinear blocks are part of a delay-free loop, and provide a method to compute those loops without rearranging them into a different topology.

## 2. FORMAL SOLUTION OF NONLINEAR DELAY-FREE PATHS

A technique to compute simple linear delay-free paths without rearranging the loopback topology has been first proposed by Härmä [15]. This technique has been successfully employed for the computation of warped IIR filters [16] and magnitude-complementary parametric equalizers [17], and it has been generalized to linear filter networks containing an arbitrary delay-free path configuration [18]. Here we will extend the same technique to networks containing nonlinear blocks.

We need to:

1. rearrange every linear and nonlinear transfer function, in order to separate the contribution of the *instantaneous* from the *historical* component in the function;
2. define additional equations accounting for the connections between blocks;
3. evaluate the historical component in every (both linear and nonlinear) block;
4. substitute the linear constraints into the nonlinear equations. Hence, if possible, solve the nonlinearities (that is, compute the signal output from every nonlinear block);
5. compute, if possible, the output from every linear block;
6. correctly update the historical components in each block.

In the following we address each point in detail.

### 2.1. Rearrangement of the transfer characteristics

The network is made of  $m_L$  linear filters, here expressed using their transfer function

$$H_i(z) = \frac{\sum_{k=0}^{Z_i} b_{k,i} z^{-k}}{1 - \sum_{k=1}^{P_i} a_{k,i} z^{-k}}, \quad i = 1, \dots, m_L, \quad (1)$$

and  $m_N$  nonlinear blocks, specified by writing their transfer characteristic in the discrete time from the input  $x_i$  to the output  $y_i$ :

$$y_i[n] = f_i(x_i[n], p_i[n]) \quad , \quad i = 1, \dots, m_N. \quad (2)$$

In writing (2), it is implicitly assumed that the nonlinear blocks respect two hypotheses:

- the output  $y_i$  can be made explicit. More weakly, that every nonlinearity admits the existence of a transfer function  $f_i$  in the form expressed by (2) [19];
- $p_i$  contains only the contribution of the historical components in the function in a way that we can evaluate the nonlinearity for past input and output values, thus obtaining a new function in the single variable  $x_i$ :

$$\begin{aligned} y_i[n] &= f_i(x_i[n], p_i[n]) \quad , \quad i = 1, \dots, m_N \\ p_i[n] &= p_i(x_i[n-1], y_i[n-1], p_i[n-1], \dots) \end{aligned} \quad (3)$$

In practice this class of nonlinear functions is sufficiently expressive for a wide range of audio applications [12, 14, 20].

On the other hand, in the linear case the historical component  $q_i$  can be immediately found out in every transfer function by gathering all past components together in the time-domain version of (1):

$$y_i[n] = b_i x_i[n] + q_i[n] \quad , \quad i = 1, \dots, m_L \quad (4)$$

$$q_i[n] = \sum_{k=1}^{Z_i} b_{k,i} x_i[n-k] + \sum_{k=1}^{P_i} a_{k,i} y_i[n-k] \quad (5)$$

in which the coefficient  $b_{0,i}$  has been simply denoted as  $b_i$ .

### 2.2. Connections between blocks

Without loss of generality we can assume that the whole network forms one single delay-free loop. Alternatively we can extract from the filter network a number of subgraphs forming individual delay-free subnetworks, then treat each of them separately using the method presented below, and finally feed the remaining blocks in the network with the output from such subnetworks<sup>1</sup> [21].

For each of the  $m_L + m_N$  block inputs we consider the  $R_i$  outputs from other blocks that feed that input, possibly with the addition of an external signal  $u_i$  directed to the same input:

$$x_i[n] = \sum_{k=1}^{R_i} y_{i_k}[n] + u_i[n] \quad , \quad i = 1, \dots, m_L + m_N. \quad (6)$$

Note that the paths that connect  $y_i$  and  $x_i$  directly must be treated as separate network branches. In other words, the condition

$$i_k \neq i \quad , \quad k = 1, \dots, R_i \quad (7)$$

must be satisfied for any  $i$ . For this purpose, such branches must result in corresponding equations that are a particular case of (4):  $y_i = x_i$ .

Equations (6) establish the correction between all the blocks. Figure 1 depicts the situation in the case of a branch formed by a linear block: in this case the historical component is computed simply by feeding the filter with a null value [15].

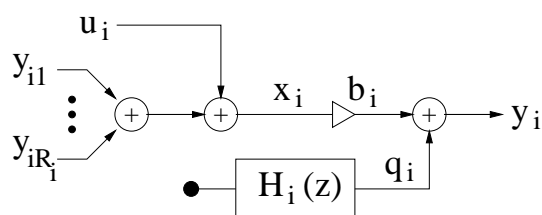


Figure 1: Structure of a linear filter branch. The output  $y_i$  is obtained as a superposition of the instantaneous component  $x_i$ , plus the historical component  $q_i$ . The input is the result of summing the outputs  $y_{i1}, \dots, y_{iR_i}$  from other branches plus one (possibly null) external input  $u_i$ .

For the nonlinear blocks the situation is slightly different, in that the scheme depicted in Figure 1 becomes computable as long as  $p_i$  is known by (3).

<sup>1</sup>Although in the pure linear case the proposed method works for any network topology [18].

### 2.3. Solution of the system

Once the previous algebra has been carried out we come up with the following equations:

$$\begin{cases} y_i[n] = f_i(x_i[n], p_i[n]), & i = 1, \dots, m_N \\ y_i[n] = b_i x_i[n] + q_i[n], & i = m_N + 1, \dots, m_N + m_L \\ x_i[n] = \sum_{k=1}^{R_i} y_{i_k}[n] + u_i[n], & i = 1, \dots, m_N + m_L \end{cases}$$

Such equations can be rewritten in matrix form as

$$\mathbf{y}_N[n] = \mathbf{f}(\mathbf{x}_N[n], \mathbf{p}[n]), \quad (8a)$$

$$\mathbf{y}_L[n] = \mathbf{B}\mathbf{x}_L[n] + \mathbf{q}[n], \quad (8b)$$

$$\mathbf{x}[n] = \mathbf{C}\mathbf{y}[n] + \mathbf{u}[n], \quad (8c)$$

in which the column vectors  $\mathbf{x}_N$ ,  $\mathbf{y}_N$ ,  $\mathbf{x}_L$ ,  $\mathbf{y}_L$ ,  $\mathbf{p}$ ,  $\mathbf{q}$ ,  $\mathbf{u}$  collect the respective signal components and  $\mathbf{y}$ ,  $\mathbf{x}$ ,  $\mathbf{f}$  are defined as

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_N \\ \mathbf{y}_L \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} \mathbf{x}_N \\ \mathbf{x}_L \end{bmatrix}, \quad \mathbf{f}(\mathbf{x}_N, \mathbf{p}) = \begin{bmatrix} f_1(x_1, p_1) \\ \dots \\ f_{m_N}(x_{m_N}, p_{m_N}) \end{bmatrix}$$

Furthermore  $\mathbf{B}$  is a diagonal matrix containing the linear coefficients  $b_1, \dots, b_{m_L}$  and  $\mathbf{C}$  accounts for the connections between blocks: if the element  $c_{ij}$  is equal to one then the  $j$ th output block is connected to the  $i$ th input block, otherwise it is equal to zero. Note that the property (7) translates into the fact that every diagonal element in  $\mathbf{C}$  is equal to zero.

In particular,  $\mathbf{C}$  can be seen as composed of four sub-matrices respectively accounting for the connections from ( $\mathbf{C}_{NN}$ ) nonlinear to nonlinear, ( $\mathbf{C}_{LL}$ ) linear to nonlinear, ( $\mathbf{C}_{LN}$ ) nonlinear to linear, and finally ( $\mathbf{C}_{LL}$ ) linear to linear blocks. For this reason in (8c) we also split the vector  $\mathbf{u}$  into  $\mathbf{u}_N$  and  $\mathbf{u}_L$ :

$$\begin{bmatrix} \mathbf{x}_N \\ \mathbf{x}_L \end{bmatrix} = \begin{bmatrix} \mathbf{C}_{NN} & \mathbf{C}_{NL} \\ \mathbf{C}_{LN} & \mathbf{C}_{LL} \end{bmatrix} \begin{bmatrix} \mathbf{y}_N \\ \mathbf{y}_L \end{bmatrix} + \begin{bmatrix} \mathbf{u}_N \\ \mathbf{u}_L \end{bmatrix}. \quad (9)$$

Substitution of equations (8a) and (8b) in (9) gives

$$\mathbf{x}_N = \mathbf{C}_{NN}\mathbf{f}(\mathbf{x}_N, \mathbf{p}) + \mathbf{C}_{NL}(\mathbf{B}\mathbf{x}_L + \mathbf{q}) + \mathbf{u}_N \quad (10a)$$

$$\mathbf{x}_L = \mathbf{C}_{LN}\mathbf{f}(\mathbf{x}_N, \mathbf{p}) + \mathbf{C}_{LL}(\mathbf{B}\mathbf{x}_L + \mathbf{q}) + \mathbf{u}_L. \quad (10b)$$

Let  $\mathbf{I}$  denote the identity matrix, then in (10a) and (10b) we can move all terms to the left-hand side except for the linear historical components and the external inputs, respectively:

$$\mathbf{x}_N - \mathbf{C}_{NN}\mathbf{f}(\mathbf{x}_N, \mathbf{p}) - \mathbf{C}_{NL}\mathbf{B}\mathbf{x}_L = \mathbf{C}_{NL}\mathbf{q} + \mathbf{u}_N, \quad (11a)$$

$$-\mathbf{C}_{LN}\mathbf{f}(\mathbf{x}_N, \mathbf{p}) + (\mathbf{I} - \mathbf{C}_{LL}\mathbf{B})\mathbf{x}_L = \mathbf{C}_{LL}\mathbf{q} + \mathbf{u}_L. \quad (11b)$$

If the matrix  $\mathbf{I} - \mathbf{C}_{LL}\mathbf{B} \triangleq \mathbf{F}_{LL}$  is invertible<sup>2</sup> then we can isolate  $\mathbf{x}_L$  in (11b):

$$\mathbf{x}_L = \mathbf{F}_{LL}^{-1}\mathbf{C}_{LN}\mathbf{f}(\mathbf{x}_N, \mathbf{p}) + \mathbf{F}_{LL}^{-1}(\mathbf{C}_{LL}\mathbf{q} + \mathbf{u}_L), \quad (12)$$

and substitute this formula inside (11a) in a way that  $\mathbf{x}_L$  is eliminated and  $\mathbf{x}_N$  remains the only unknown vector in that equation. After some algebraic manipulation (11a) is in fact carried out as:

$$\mathbf{x}_N = \mathbf{W}_1\mathbf{f}(\mathbf{x}_N, \mathbf{p}) + \mathbf{W}_2\mathbf{q} + \mathbf{W}_3\mathbf{u}_L + \mathbf{u}_N \quad (13)$$

<sup>2</sup>It can be shown that the inverse matrix exists if the linear part of the network respects the hypothesis of being causal [18].

with

$$\mathbf{W}_3 = \mathbf{C}_{NL}\mathbf{B}\mathbf{F}_{LL}^{-1} \quad (14a)$$

$$\mathbf{W}_1 = \mathbf{W}_3\mathbf{C}_{LN} + \mathbf{C}_{NN} \quad (14b)$$

$$\mathbf{W}_2 = \mathbf{W}_3\mathbf{C}_{LL} + \mathbf{C}_{NL} \quad (14c)$$

### 2.4. Solving nonlinearities

Equation (13) defines the inputs  $\mathbf{x}_N[n]$  to the nonlinear blocks in terms of known quantities: the historical components  $\mathbf{q}[n]$  and the external inputs  $\mathbf{u}_L[n]$ ,  $\mathbf{u}_N[n]$ . In addition the matrix  $\mathbf{W}_1$  isolates the instantaneous dependence of  $\mathbf{x}_N[n]$  on  $\mathbf{y}_N[n]$ . From (8a) and (13), the nonlinear equations can be written as

$$\mathbf{y}_N[n] = \mathbf{f}(\mathbf{W}_1\mathbf{y}_N[n] + \mathbf{W}_2\mathbf{q}[n] + \mathbf{W}_3\mathbf{u}_L[n] + \mathbf{u}_N[n], \mathbf{p}[n]). \quad (15)$$

Recall that  $\mathbf{p}$  contains only the contributions of historical components (see equation (3)). Therefore at each time step  $\mathbf{y}_n$  is the only unknown in (15) and  $\mathbf{p}$  parametrizes the function  $\mathbf{f}$ :

$$\mathbf{y}_N[n] = \tilde{\mathbf{f}}_{\mathbf{p}}(\mathbf{W}_1\mathbf{y}_N[n] + \tilde{\mathbf{x}}_N[n]), \quad (16)$$

where  $\tilde{\mathbf{f}}_{\mathbf{p}}(\cdot) = \mathbf{f}(\cdot, \mathbf{p}[n])$  and  $\tilde{\mathbf{x}}_N[n] = \mathbf{W}_2\mathbf{q}[n] + \mathbf{W}_3\mathbf{u}_L[n] + \mathbf{u}_N[n]$  collects the contribution of known quantities to the input  $\mathbf{x}_N$ .

Equation (16) provides a formulation which has strong similarities with that developed in [19]. According to it, the nonlinear function  $\tilde{\mathbf{f}}_{\mathbf{p}}$  defines implicitly the dependence of  $\mathbf{y}_N[n]$  on  $\tilde{\mathbf{x}}_N[n]$ . In addition to the formulation given in [19], the nonlinearity is here parametrized by the historical components  $\mathbf{p}$ .

It was shown in [19] that, under appropriate conditions for the matrix  $\mathbf{W}_1$ , the implicit dependence  $\mathbf{y}_N(\tilde{\mathbf{x}}_N)$  admits a global representation, and can therefore be precomputed and stored in a look-up table for efficient implementation. However, the efficiency of a table look-up drops dramatically with increasing dimensionality of the input. Furthermore, in our case the input to the table comprises not only  $\tilde{\mathbf{x}}_N$ , but also the historical components  $\mathbf{p}$ .

An alternative choice to table look-up is iterative search: at each time step the value  $\mathbf{y}_N[n]$  is found by searching a local zero of the function

$$\mathbf{g}[\mathbf{y}_N] = \tilde{\mathbf{f}}_{\mathbf{p}}(\mathbf{W}_1\mathbf{y}_N + \tilde{\mathbf{x}}_N) - \mathbf{y}_N \quad (17)$$

The Newton-Raphson algorithm operates the search in this way:

```


$$\mathbf{y}_{N0} = \mathbf{y}_N[n-1]$$


$$k = 1$$

while (err > Errmax)
  Compute  $\mathbf{g}(\mathbf{y}_{Nk})$  from Eq. (17)
  Compute  $\mathbf{y}_{N(k+1)} = \mathbf{y}_{Nk} - \mathbf{J}_k^{-1} \cdot \mathbf{g}(\mathbf{y}_{Nk})$ 
  Compute err = abs( $\mathbf{y}_{N(k+1)} - \mathbf{y}_{Nk}$ )
   $k = k + 1$ 
end

$$\mathbf{y}_N[n] = \mathbf{y}_{Nk}$$


```

where  $\mathbf{J}_k = \left[ \frac{\partial \mathbf{g}_i}{\partial (\mathbf{y}_N)_j} \right]_{\mathbf{y}_{Nk}}$  is the Jacobian of  $\mathbf{g}$  evaluated in  $\mathbf{y}_{Nk}$ .



therefore described in the continuous-time domain as:

$$\ddot{y}_j^{(r)} + g_j \dot{y}_j^{(r)} + [\omega_{0j}^2 + \omega_{1j}^2 \cdot |y_j^{(r)}|^{\alpha_j}] y_j^{(r)} = \frac{1}{m_j} f_{tot}^{(r)}, \quad (18)$$

where  $y_j^{(r)}$  is the displacement of the  $j$ th mode of the resonator and  $f_{tot}^{(r)}$  represents the net sum of forces that drive the resonator. The parameter  $g_j$  is the modal damping coefficient, and  $1/m_j$  controls the “inertial” properties of the mode ( $m_j$  has the dimension of a mass). Note that a nonlinear correction  $\omega_{1j}^2 \cdot |y_j^{(r)}|^{\alpha_j}$  has been added to the center frequency  $\omega_{0j}$ .

The effect of this correction is to increase the instantaneous center frequency with increasing displacement<sup>3</sup>.

Equation (18) can be rewritten as

$$\ddot{y}_j^{(r)} + g_j \dot{y}_j^{(r)} + \omega_{0j}^2 y_j^{(r)} = \frac{1}{m_j} [f_{tot}^{(r)} - m_j \omega_{1j}^2 \cdot \text{sgn}(y_j^{(r)}) |y_j^{(r)}|^{\alpha_j + 1}], \quad (19)$$

where a second order linear oscillator is recognized on the left-hand side, and the nonlinear correction is provided as an input to the oscillator on the right-hand side.

### 3.2. Contact

Given two nonlinear modal resonators ( $r = 1, 2$ ) as described above, an impulsive contact (collision) between them is modeled by assuming that the force acting on each mode is given by

$$f_{tot}^{(r)} = f_{ext}^{(r)} + (-1)^r f_c, \quad (20)$$

where  $f_c$  represents the contact force that occurs during collision. This is generated using the nonlinear model discussed in [22]:

$$f_c(y_c, \dot{y}_c) = \begin{cases} k y_c^\alpha + \lambda y_c^\alpha \cdot \dot{y}_c & y_c > 0, \\ 0 & y_c \leq 0, \end{cases} \quad (21)$$

where  $y_c = y^{(1)} - y^{(2)}$  is the inter-penetration of the two objects and  $y^{(1)}$ ,  $y^{(2)}$  are the object displacements, obtained as linear combinations of the respective modal displacements (see [22]).

### 3.3. Discrete-time system

The simple mechanical system described above can be transposed into the discrete-time domain using the formulation outlined in Section 2. The number  $m_L$  of linear blocks equals the total number of linear oscillators (19). The number  $m_N$  of nonlinear blocks equals the number of non-null correction terms  $\omega_{1j}$  plus one, i.e. the contact force (21).

#### Nonlinear blocks

The inputs to the nonlinear blocks are defined as follows:

$$\begin{aligned} x_i &:= y_j^{(r)}, \quad (i = 1, \dots, m_N - 1, \quad j : \omega_{1j} \neq 0), \\ x_{m_N} &:= y_c, \end{aligned} \quad (22)$$

The external inputs  $u_L$  are zero. The historical components  $p$  are all zero except for the last one:

$$p_i = 0 \quad (i = 1, \dots, m_N - 1). \quad (23)$$

<sup>3</sup>Note however that no attempt is made here to provide a quantitative physical description of the nonlinear correction. We are only interested in its qualitative effect, and its implications on the computational structure.

In order to compute the contact force (21), the inter-penetration velocity  $\dot{y}_c$  is also needed. This is estimated in the discrete-time domain using the bilinear transformation:

$$\begin{aligned} \dot{y}_c[n] &= 2F_s(y_c[n] - y_c[n-1]) - \dot{y}_c[n-1] \\ &= 2F_s x_{m_N}[n] + p_{m_N}[n], \end{aligned} \quad (24)$$

where the historical component  $p_{m_N}[n]$  is defined as

$$\begin{aligned} p_{m_N}[n] &:= 2F_s x_{m_N}[n-1] + \dot{y}_c[n-1] \\ &= 4F_s x_{m_N}[n-1] - p_{m_N}[n-1] \end{aligned} \quad (25)$$

The nonlinear characteristics are then given as

$$f_i(x_i, p_i) := -m_j \omega_{1j}^2 \cdot (x_i)^{\alpha_j + 1}, \quad (26a)$$

$$(i = 1, \dots, m_N - 1, \quad j : \omega_{1j} \neq 0),$$

$$f_{m_N}(x_{m_N}, p_{m_N}) := f_c(x_{m_N}, 2F_s x_{m_N} + p_{m_N}). \quad (26b)$$

#### Linear blocks

The inputs to the linear blocks are defined as follows:

$$\begin{aligned} x_i &:= y_{i-m_N} + (-1)^r y_{m_N}, \\ &\quad (i = m_N + 1, \dots, 2m_N - 1), \\ x_i &:= (-1)^r y_{m_N} \\ &\quad (i = 2m_N, \dots, m_N + m_L), \end{aligned} \quad (27)$$

with  $r (= 1, 2)$  such that the  $i$ th input belongs to resonator  $r$ . The external inputs  $u_L$  are simply

$$u_i := f_{ext}^{(r)}, \quad (i = m_N + 1, \dots, m_N + m_L) \quad (28)$$

with  $r (= 1, 2)$  such that the  $i$ th input belongs to resonator  $r$ .

Correspondingly, the outputs from the linear blocks are the modal displacements:

$$y_i := y_j^{(r)} \quad (i = m_N + 1, \dots, m_N + m_L, \quad j = 1, \dots, m_L). \quad (29)$$

In order to compute the transfer functions  $H_i(z)$  between the forces  $x_i$  and the modal displacements  $y_i$  ( $i = m_N + 1, \dots, m_N + m_L$ ,  $j = 1, \dots, m_L$ ), equation (19) is discretized with the bilinear transformation. It can be seen that the filter coefficients  $a_{k,i}$  and  $b_{k,i}$  defined in (1) are given by

$$\begin{aligned} a_{1,i} &= \frac{\omega_{0j}^2 - 4F_s^2}{2\Delta_j}, \quad a_{2,i} = \frac{\omega_{0j}^2 + 4F_s^2}{2\Delta_j} - 1, \\ b_{0,i} &= b_{2,i} = \frac{1}{4m_j}, \quad b_{1,i} = \frac{1}{2m_j}, \\ &\quad (i = m_N + 1, \dots, m_N + m_L, \quad j = 1, \dots, m_L). \end{aligned} \quad (30)$$

where the quantity  $\Delta_j$  is given by  $\Delta_j = F_s^2 + g_j F_s / 2 + [\omega_{0j}]^2 / 4$ .

### 3.4. Summary of the example

The above discussion has shown that the mechanical system under consideration can be assembled using three kind of computational blocks.

- The second-order filter whose coefficients are given in (30) is instantiated as many times as the number of modeled resonating modes.
- The nonlinear memoryless block (26a) is instantiated as many times as the number of non-null correction terms  $\omega_{1,j}$ .



- The nonlinear block with memory (26b) is instantiated once and accounts for the contact force.

The computational blocks are then connected according to the topology specified by equations (22) and (27): note that these equations define implicitly the connection matrix  $C$  given in (9). Finally, the computation of the resulting system is performed according to the procedure summarized in Section 2.

#### 4. CONCLUSION

We have presented a general procedure that enables to model a generic network of nonlinear and linear computational blocks that satisfy weak hypotheses.

It has been shown that the network can be solved even in the presence of delay-free paths that involve nonlinear blocks. Moreover, the proposed solution does not require any rearrangement of the network and preserves the original topology of the network, in terms of mutual input/output relations between blocks.

The example application provided in Section 3 has shown that the proposed procedure allows for a highly modular formulation of the system. Each computational block has a clear physical meaning: in the example the linear blocks are second order oscillators that represent modes of resonating objects, while nonlinear blocks are forces.

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.

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.

#### 5. REFERENCES

- [1] Hammond Organ Company, Franklin Park, IL, *Hammond B-3 Service Manual*.
- [2] B. Blesser, "An interdisciplinary synthesis of reverberation viewpoints," *J. of the Audio Engineering Society*, vol. 49, no. 10, 2001.
- [3] S. K. Mitra, *Digital Signal Processing. A computer-Based Approach*, McGraw-Hill, New York, 1998.
- [4] 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/>.
- [5] L. Trautmann and R. Rabenstein, "Digital sound synthesis based on transfer function models," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, 1999, pp. 83–86.
- [6] C. Cadoz, A. Luciani, and J.-L. Florens, "CORDIS-ANIMA: A Modeling and Simulation System for Sound Synthesis - The General Formalism," *Computer Music Journal*, vol. 17, no. 1, pp. 19–29, Spring 1993.
- [7] A. C. Constantinides, "Spectral transformations for digital filters," *Proc. IEE*, vol. 117, pp. 1585–1590, 1970.
- [8] A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1989.
- [9] J. O. Smith, "Physical modeling using digital waveguides," *Computer Music Journal*, vol. 16, no. 4, pp. 74–91, Winter 1992.
- [10] G. De Poli and D. Rocchesso, "Physically-based sound modeling," *Organised Sound*, vol. 3, no. 1, 1998.
- [11] S. A. Van Duyne and J. O. Smith, "A simplified approach to modeling dispersion caused by stiffness in strings and plates," in *Proc. Int. Computer Music Conf.*, Aarhus, Denmark, Sept. 1994, ICMA, pp. 407–410.
- [12] 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.
- [13] F. Avanzini and D. Rocchesso, "Modeling collision sounds: Non-linear contact force," in *Proc. Conf. on Digital Audio Effects (DAFX-01)*, Limerick, Ireland, Dec. 2001, pp. 61–66.
- [14] 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.
- [15] A. Härmä, "Implementation of frequency-warped recursive filters," *EURASIP Signal Processing*, vol. 80, no. 3, pp. 543–548, Mar. 2000.
- [16] 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.
- [17] 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.
- [18] 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.
- [19] 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.
- [20] 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.
- [21] J. Szczupak and S. K. Mitra, "Detection, location, and removal of delay-free loops in digital filter configurations," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. ASSP-23, no. 6, pp. 558–562, 1975.
- [22] 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.