

Elimination of Delay-Free Loops in Discrete-Time Models of Nonlinear Acoustic Systems

Gianpaolo Borin, Giovanni De Poli, *Member, IEEE*, and Davide Rocchesso, *Associate Member, IEEE*

Abstract—Nonlinear acoustic systems are often described by means of nonlinear maps acting as instantaneous constraints on the solutions of a system of linear differential equations. This description leads to discrete-time models exhibiting noncomputable loops. We present a solution to this computability problem by means of geometrical transformation of the nonlinearities and algebraic transformation of the time-dependent equations. The proposed solution leads to stable and accurate simulations even at relatively low sampling rates.

Index Terms—Acoustic system modeling, Kirchhoff digital domain, noncomputable loops, nonlinear circuits, wave digital domain.

I. INTRODUCTION

AS physical modeling has become one of the most important techniques for sound synthesis, some structural problems which limit the accuracy of simulations have arisen. As an example, consider the case of a nonlinear exciter (e.g., the reed) described as a nonlinear map relating two dual Kirchhoff variables (e.g., pressure versus volume velocity), and a resonator (e.g., the bore) described by means of digital waveguides [1]–[3]. Fig. 1 shows that the signal flowgraph resulting from the connection of exciter and resonator exhibits a noncomputable delay-free loop due to the conversion between Kirchhoff variables and wave variables. More severe computability problems appear when simulating dynamic exciters, since the linear equations used to describe the system dynamics are tightly coupled with some instantaneous nonlinear map [4]–[6].

In this paper, a general method is proposed for solving computability problems in nonlinear acoustic systems. The method transforms the linear equations of the system in order to highlight all noncomputable loops involving nonlinear maps, and then operates a geometrical transformation on the nonlinearities in order to cut the instantaneous dependencies. In general, it would be possible to use iterative procedures for solving implicit nonlinear relations. However, we propose a noniterative solution based on a precomputed table storing the transformed nonlin-

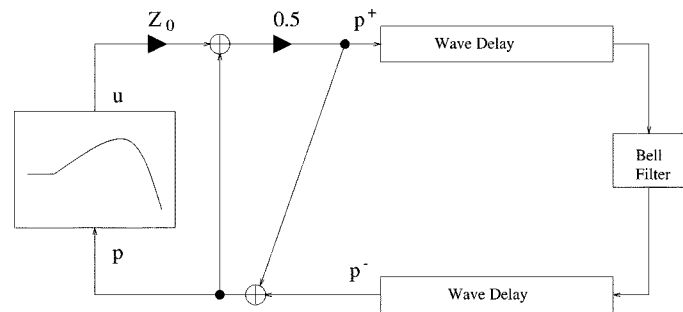


Fig. 1. Connection of a static reed model with a waveguide bore model.

earities. Therefore, an efficient realization of discrete-time systems simulating nonlinear acoustic systems can be readily implemented on digital signal processors.

Other noniterative methods have been proposed in the literature of wave digital filters for solving circuits having nonlinear elements without [7] or with memory [8]–[10]. These other approaches assume using wave variables rather than Kirchhoff variables to describe every distributed or lumped component of the equivalent circuit. Every circuit component is treated as a scattering element having a reference impedance, and different components are connected by adaptors.

The new method presented herein operates on nonlinearities described in terms of familiar and more easily accessible Kirchhoff variables, and it is therefore called the K method. Moreover, while the literature of wave digital filters only deals with nonlinear bipoles with or without memory, the K method can manage nonlinear functions of any set of Kirchhoff variables found in the circuit.

The K method is somewhat resemblant of other methods for removing delay-free loops from digital filters [11], [12], since it partitions the signal flowgraph into a subgraph having no delay-free loops and another subgraph having delay-free loops. However, while those other methods do only deal with linear filters, the K method allows arbitrary nonlinearities to be inserted anywhere in the circuit.

In Section II, we give a general treatment of the K method. In Section III, we validate the K method by applying it to a reference well-known nonlinear circuit, thus also demonstrating that the method can be used for simulating nonacoustic systems. In Section IV, we show how the K method improves the simulation of hammer-string interaction in sound synthesis of piano tones. In Section V, we show how the interaction of a dynamic reed with a bore implemented as a digital waveguide can be solved with the proposed method.

Manuscript received June 3, 1999; revised September 24, 1999. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Dennis R. Morgan.

G. Borin and G. De Poli are with the Centro di Sonologia Computazionale, Dipartimento di Elettronica e Informatica, Università degli Studi di Padova, 35131 Padova, Italy (e-mail: borin@dei.unipd.it; depoli@dei.unipd.it).

D. Rocchesso is with the Dipartimento Scientifico e Tecnologico, Università degli Studi di Verona, 37134 Verona, Italy (e-mail: rocchesso@sci.univr.it; <http://www.sci.univr.it/~rocchesso>).

Publisher Item Identifier S 1063-6676(00)06921-2.

II. K-METHOD

In this section, we present the general formulation of the K method and we illustrate the assumptions and conditions for its applicability.

A. System Decomposition

The underlying assumption for applying the K method is that the continuous-time system can be decomposed into a nonlinear instantaneous multiple-input multiple-output (MIMO) map $\mathbf{f}(\cdot)$ and a linear filter $\mathbf{L}(s)$ containing all the system dynamics. Even more generally, the nonlinear map might be expressed in the form of an implicit relationship $\mathbf{g}(\mathbf{x}, \mathbf{y}) = \mathbf{0}$, in which case we don't need to find an explicit $\mathbf{f}(\cdot)$ in order to apply the method. However, for ease of presentation, the structure we assume to be working with is depicted in Fig. 2, where we have introduced an exogenous vector \mathbf{u} which can be interpreted as a set of modulating signals in many practical cases. Namely, in a sound synthesis model, \mathbf{u} should contain the signals generated by user's gestures. Moreover, a signal vector \mathbf{y}_E is assumed to be extracted from the linear block as an output from the system. The reference structure we are adopting does not seem to be restrictive for acoustic modeling, because the linear part can be arranged to include the dynamics of nonlinear exciters as well as linear resonating elements.

The main driving idea for the derivation of the K method is to isolate any delay-free path connecting any element of \mathbf{y} with any element of \mathbf{x} . Then, geometrical transformations are applied to the system in order to get rid of those paths.

B. System Equations

To explain the K method, we assume that the continuous-time linear dynamics is being discretized by means of Euler backward differencing

$$s = \alpha \cdot (1 - z^{-1}) \quad (1)$$

where s is the Laplace complex variable and $\alpha = F_s$ is the sampling rate. Other, more effective, discretization techniques can be used and only affect the transformation matrices,¹ while the overall algorithm is maintained.

The continuous-time MIMO feedback system of Fig. 2 can be expressed in state-variable form as

$$\dot{\mathbf{w}} = \mathbf{A}\mathbf{w} + \mathbf{B}\mathbf{u} + \mathbf{C}\mathbf{y} \quad (2)$$

$$\mathbf{y} = \mathbf{f}(\mathbf{D}\mathbf{w} + \mathbf{E}\mathbf{u} + \mathbf{F}\mathbf{y}) \triangleq \mathbf{f}(\mathbf{x}) \quad (3)$$

$$\mathbf{y}_E = \mathbf{A}_E\mathbf{w} + \mathbf{B}_E\mathbf{u} + \mathbf{C}_E\mathbf{y} \quad (4)$$

where $\mathbf{w} \in \mathcal{R}^P$ is the state vector, and $\mathbf{x} \in \mathcal{R}^N$, $\mathbf{u} \in \mathcal{R}^L$, $\mathbf{y} \in \mathcal{R}^M$. The output variable \mathbf{y}_E and the matrices \mathbf{A}_E , \mathbf{B}_E , and \mathbf{C}_E

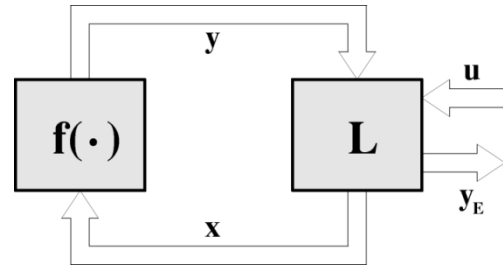


Fig. 2. System block decomposition.

will not be hereafter considered because they are not involved in any delay-free loop, and therefore the computation of (4) is straightforward. In particular, if \mathbf{y}_E contains wave variables to be fed into a waveguide model [2], and other wave variables are returned via \mathbf{u} , it is assumed that there is at least a delay element (or a waveguide section which is implemented with a couple of unit delays) in any path connecting \mathbf{y}_E to \mathbf{u} on the right-hand side of Fig. 2. This assumption allows decoupling the system of Fig. 2 from any other blocks it might be connected with.

The system dynamics is all contained in (2), which is discretized by applying (1) to its Laplace transform

$$(\alpha\mathbf{I} - \mathbf{A})\mathbf{W}(z) = z^{-1}\alpha\mathbf{W}(z) + \mathbf{B}\mathbf{U}(z) + \mathbf{C}\mathbf{Y}(z). \quad (5)$$

Here, $\mathbf{U}(z)$, $\mathbf{Y}(z)$, and $\mathbf{W}(z)$ are the z -transforms of the input, output, and state vector signals, respectively.

As long as the matrix

$$\mathbf{H} \triangleq (\alpha\mathbf{I} - \mathbf{A})^{-1} \quad (6)$$

exists (i.e., α is not an eigenvalue of \mathbf{A}), (5) can be rewritten in the discrete-time variable n as

$$\mathbf{w}(n) = \mathbf{H}\mathbf{C}\mathbf{y}(n) + \mathbf{H}(\mathbf{B}\mathbf{u}(n) + \alpha\mathbf{w}(n-1)). \quad (7)$$

From (3) and (7), the system of equations governing the nonlinear feedback scheme can be written as

$$\mathbf{p}(n) = (\mathbf{D}\mathbf{H}\mathbf{B} + \mathbf{E})\mathbf{u}(n) + \alpha\mathbf{D}\mathbf{H}\mathbf{w}(n-1) \quad (8)$$

$$\mathbf{y}(n) = \mathbf{f}(\mathbf{K}\mathbf{y}(n) + \mathbf{p}(n)) \quad (9)$$

where

$$\mathbf{K} = (\mathbf{D}\mathbf{H}\mathbf{C} + \mathbf{F}) \in \mathcal{R}^{N \times M} \quad (10)$$

is the weighting matrix for the delay-free path connecting \mathbf{y} to \mathbf{x} , and $\mathbf{p}(n) \in \mathcal{R}^N$ is the contribution of previously-computed and known (exogenous) terms.

Equation (9) is valid for other discretization techniques as well. For example, if the bilinear transformation $s = \alpha((1 - z^{-1})/(1 + z^{-1}))$ is used instead of Euler differencing, the matrix

¹Actually, as explained in Section II-D, a change of the discretization technique might induce a change in the procedure for setting the initial conditions.

TABLE I
SIMULATION STEPS NEEDED FOR
PRODUCING ONE OUTPUT SAMPLE FROM THE SYSTEM OF FIG. 2

Variable	Computation
$\bar{\mathbf{p}}(n)$	$= \mathbf{H}\mathbf{B}\mathbf{u}(n) + \alpha\mathbf{H}\mathbf{w}(n-1)$
$\mathbf{p}(n)$	$= \mathbf{D}\bar{\mathbf{p}}(n) + \mathbf{E}\mathbf{u}(n)$
$\mathbf{y}(n)$	$= \mathbf{h}(\mathbf{p}(n))$
$\mathbf{w}(n)$	$= \mathbf{H}\mathbf{C}\mathbf{y}(n) + \bar{\mathbf{p}}(n)$

\mathbf{K} is structurally unchanged,² and the only difference is found in the expression of $\mathbf{p}(n)$ that has to be rewritten as

$$\mathbf{p}(n) = (\mathbf{D}\mathbf{H}\mathbf{B} + \mathbf{E})\mathbf{u}(n) + \mathbf{D}\mathbf{H}(\alpha\mathbf{I} + \mathbf{A})\mathbf{w}(n-1) + \mathbf{D}\mathbf{H}\mathbf{B}\mathbf{u}(n-1) + \mathbf{D}\mathbf{H}\mathbf{C}\mathbf{y}(n-1). \quad (11)$$

C. Elimination of Delay-Free Loops

The formulation of (8) and (9) has isolated the implicit term $\mathbf{K}\mathbf{y}$ in the argument of the instantaneous nonlinear map \mathbf{f} . In order to solve the computability problem due to delay-free loops in (9) we have to transform the vector map

$$\mathbf{y} = \mathbf{f}(\mathbf{K}\mathbf{y} + \mathbf{p}) = \mathbf{f}(\mathbf{x}); \mathcal{R}^N \rightarrow \mathcal{R}^M \quad (12)$$

into an explicit form. Such a transformation is possible if the assumptions of the *implicit mapping theorem* (see, e.g., [13, p. 17]) are satisfied. In our multivariable formulation, the implicit mapping theorem reads as

given the vector function $\mathbf{g}(\mathbf{x}, \mathbf{y}): \mathcal{R}^N \times \mathcal{R}^M \rightarrow \mathcal{R}^M$, continuous with its derivatives, if there is a point $P_0 = (\mathbf{x}_0, \mathbf{y}_0)$, where $\mathbf{g}(\mathbf{x}_0, \mathbf{y}_0) = \mathbf{0}$ and the Jacobian determinant

$$\left. \frac{\partial \mathbf{g}}{\partial \mathbf{y}} \right|_{P_0} \triangleq \det \begin{bmatrix} \frac{\partial g_1}{\partial y_1} & \cdots & \frac{\partial g_1}{\partial y_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_M}{\partial y_1} & \cdots & \frac{\partial g_M}{\partial y_M} \end{bmatrix} \Big|_{P_0} \neq 0 \quad (13)$$

then there exists a vector function $\mathbf{h}(\mathbf{x}): \mathcal{R}^N \rightarrow \mathcal{R}^M$, continuous with its derivatives in an open neighborhood $A \subset \mathcal{R}^N$ containing \mathbf{x}_0 , such that

$$\mathbf{h}(\mathbf{x}_0) = \mathbf{y}_0 \quad (14)$$

and

$$\mathbf{g}(\mathbf{x}, \mathbf{h}(\mathbf{x})) = \mathbf{0} \quad \forall \mathbf{x} \in A. \quad (15)$$

Applying a little matrix algebra to our function $\mathbf{g}(\mathbf{x}, \mathbf{y}) = \mathbf{f}(\mathbf{x}) - \mathbf{y}$, the condition dictated by the implicit function theorem for the existence of a local explicit form of the vector map (12) reads as

$$\det \left| \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \mathbf{K} - \mathbf{I} \right| \neq 0. \quad (16)$$

In other words, if the matrix $(\partial \mathbf{f} / \partial \mathbf{x})\mathbf{K} - \mathbf{I}$ is not singular in P_0 , the multivalued function \mathbf{f} can be made locally explicit. If

²This is evident if we write $s = \alpha((1 - z^{-1}) / (1 + z^{-1})) = \alpha(1 - (2 / (1 + z^{-1})))z^{-1}$, where the instantaneous term is identical to the case of backward differencing.

this condition holds for any P_0 , \mathbf{f} can be made globally explicit, i.e., it is possible to construct a function $\mathbf{h}(\cdot)$ such that $\mathbf{y} = \mathbf{h}(\mathbf{p})$. In the more general case of a nonlinear map implicitly defined by $\mathbf{g}(\mathbf{x}, \mathbf{y}) = \mathbf{0}$, the condition for finding an explicit $\mathbf{y} = \mathbf{h}(\mathbf{p})$ is $\det |(\partial \mathbf{g} / \partial \mathbf{x})\mathbf{K} - (\partial \mathbf{g} / \partial \mathbf{y})| \neq 0$.

The identification and separation of instantaneous contributions of the output of \mathbf{f} to its input, expressed in (9), induces a geometric transformation that is a multidimensional shear

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{K} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{p} \\ \mathbf{y} \end{bmatrix} \triangleq \mathbf{S} \begin{bmatrix} \mathbf{p} \\ \mathbf{y} \end{bmatrix} \quad (17)$$

relating the two spaces (\mathbf{p}, \mathbf{y}) and (\mathbf{x}, \mathbf{y}) , so that the new nonlinear function $\mathbf{h}(\mathbf{p})$ can be computed by point-to-point transformation of $\mathbf{f}(\mathbf{x})$. The inverse transformation is, of course

$$\begin{bmatrix} \mathbf{p} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & -\mathbf{K} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \triangleq \mathbf{S}^{-1} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}. \quad (18)$$

So, the major burden left to the implementer would be that of properly shearing the nonlinearity, a task that can often be accomplished off line.

When it is neither possible nor convenient to find an explicit $\mathbf{y} = \mathbf{h}(\mathbf{p})$, it might still be possible to solve for the other vector variable \mathbf{x} just by using the inverse of the linear system \mathbf{L} , when it exists [14].

From a computational viewpoint, it is advantageous to introduce a vector variable $\bar{\mathbf{p}}(n)$, that is an intermediate product of the computation of $\mathbf{p}(n)$. With the proper definition of $\bar{\mathbf{p}}(n)$, the system of Fig. 2 can be simulated (in the case of backward differencing) by the four steps reported in Table I. After all the variables have been properly initialized, the simulation can proceed by iteration of the four steps of Table I.

D. Setting Initial Conditions

As far as initial conditions are concerned, we see that the procedure is quite dependent on the discretization method that is chosen. For example, if we adopt the backward difference scheme, we note that $\bar{\mathbf{p}}$ does not depend on \mathbf{y} , so that the knowledge of \mathbf{w} at time -1 is sufficient³ to start the simulation algorithm with proper initial conditions. On the other hand, if we adopt the bilinear transformation scheme, we notice that $\bar{\mathbf{p}}(0)$ does depend on $\mathbf{y}(-1)$, and so we must somehow guess the initial value of the output variable in order to start the simulation.

In this second situation, we can solve (3) with $\mathbf{w}(-1)$ and $\mathbf{u}(-1)$. Since this usually must be done once in the algorithm lifetime, any suitable iterative method can be used. However, it should be noted that in many practical cases we can assume that the system starts from a situation in which both the state and the output variables are known (and quite often are zero). Obviously, in these cases there is no need to solve the nonlinear equation (3).

As a concluding remark, we can compare the initial condition setting procedure of the K method with the one required by wave methods [10]. In our case, we have to determine the initial value for \mathbf{y} , but this step is often unnecessary if we suitably choose the discretization method or the starting point. With wave methods,

³We assume here that the first simulation output sample will be produced at time 0.

the problem is that of expressing the initial values of wave variables given the Kirchhoff counterparts, and this always requires the solution of a linear system of equations.

E. A Special Case: The Linear Map

The K method can still come useful when the function (3) degenerates in a linear map, as in

$$\mathbf{y} = \mathbf{S}\mathbf{x} + \mathbf{T}; \mathcal{R}^N \rightarrow \mathcal{R}^M. \quad (19)$$

In this case, instead of solving the whole system, we can apply the K method by just finding the closed form solution for the linear version of (12)

$$\mathbf{y} = (\mathbf{I} - \mathbf{S}\mathbf{K})^{-1}\mathbf{S}\mathbf{p} + (\mathbf{I} - \mathbf{S}\mathbf{K})^{-1}\mathbf{T} \triangleq \hat{\mathbf{S}}\mathbf{p} + \hat{\mathbf{T}} \quad (20)$$

and then use (20) in the algorithm shown in Table I.

This method becomes convenient when considering a piecewise linear map, as in the case of the elastic force law of a linear piano hammer (see Section IV).

III. VALIDATION BY SIMULATION OF THE CHUA–FELDERHOFF CIRCUIT

In order to show that the K method gives consistent results in practical cases, we need an application example with the following characteristics:

- it is a simple dynamic nonlinear model;
- its behavior has been extensively studied in the literature;
- its behavior is sensible to discretization.

The Chua–Felderhoff circuit [8], [10], depicted in Fig. 3, has all the characteristics that we seek, thus being a good testbed for validating the K method. The circuit is parameterized as follows:

$$L = 100 \mu\text{H} \quad R = 180 \Omega \quad e(t) = e_0 \sin(2\pi f_0 t) \quad (21)$$

with $e_0 = 3.57$ V in our specific example. The nonlinear capacitor has the voltage-charge characteristic

$$v(q) = \frac{1}{2v_0 C_0} \left(q^2 + q \sqrt{q^2 + 4C_0^2 v_0^2} \right) \quad (22)$$

where $C_0 = 80$ pF and $v_0 = 0.6$ V.

The equations of the circuit are easily derived as

$$\begin{cases} \dot{q} = i \\ \dot{i} = -\frac{R}{L}i + \frac{1}{L}e - \frac{1}{L}v \\ v = f(q) \end{cases} \quad (23)$$

where the meaning of symbols is clear from Fig. 3 and the dotted notation is used for time derivatives. The matrices of (2) and (3) are found

$$\mathbf{w} = \begin{bmatrix} q \\ i \end{bmatrix}; \quad \mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{R}{L} \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix}; \quad (24)$$

$$\mathbf{C} = \begin{bmatrix} 0 \\ -\frac{1}{L} \end{bmatrix}; \quad \mathbf{u} = e; \quad \mathbf{y} = v; \quad (25)$$

$$\mathbf{D} = [1 \ 0]; \quad \mathbf{E} = 0; \quad \mathbf{F} = 0. \quad (26)$$

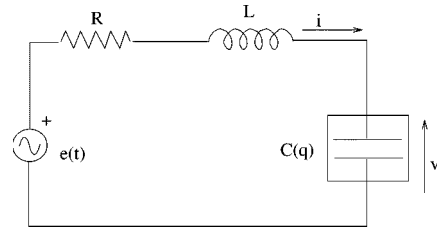


Fig. 3. Chua–Felderhoff circuit.

The matrices \mathbf{K} and \mathbf{p} of (10) and (8) are, respectively,

$$\mathbf{K} = -\frac{1}{\alpha(R + \alpha L)} \quad (27)$$

and

$$\mathbf{p}(n) = \frac{1}{\alpha(R + \alpha L)} \mathbf{u}(n) + \left[1 \quad \frac{L}{R + \alpha L} \right] \mathbf{w}(n-1). \quad (28)$$

If the bilinear transformation is used, (28) is superseded by

$$\begin{aligned} \mathbf{p}(n) = & \frac{1}{\alpha(R + \alpha L)} (\mathbf{u}(n) + \mathbf{u}(n-1)) \\ & + \left[1 \quad \frac{2\alpha L}{\alpha(R + \alpha L)} \right] \mathbf{w}(n-1) \\ & + \frac{1}{R + \alpha L} \mathbf{y}(n-1). \end{aligned} \quad (29)$$

Since in this example the nonlinear mapping (22) is one-to-one, the shearing transformation is simply

$$\begin{bmatrix} p \\ v \end{bmatrix} = \begin{bmatrix} 1 & \frac{1}{\alpha(R + \alpha L)} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} q \\ v \end{bmatrix} \quad (30)$$

and the condition for the existence of an explicit map is

$$-\frac{\partial f}{\partial q} \frac{1}{\alpha(R + \alpha L)} \neq 1. \quad (31)$$

Since (31) is valid for all values of q , $f(\cdot)$ can be made globally explicit. The explicit function $v = \mathbf{h}(\mathbf{p})$ can be tabulated by linear transformation of the points (q, v) . In this case, it is also possible to work out a closed form expression for the explicit function.

In Fig. 4, we show the steady-state phase portrait, obtained when a simulation is run with $F_s = 32f_0$ and $f_0 = 1/(2\pi\sqrt{LC_0}) \simeq 1.78$ MHz. The results are “close quantitatively and similar qualitatively”⁴ to those obtained with wave methods [8], [10] and the subharmonic oscillation is preserved under different sampling rates.

IV. HAMMER–STRING INTERACTION

For sound synthesis of piano tones, a hammer model based on a mass and a nonlinear spring has been extensively studied [5], [16]. The continuous-time equations for such a model are

$$d \triangleq y_h - y \quad (32)$$

$$f = \begin{cases} kd^r, & d \geq 0 \\ 0, & d < 0 \end{cases} \quad (33)$$

$$f = -m_h \ddot{y}_h, \quad (34)$$

⁴Quotation from [15, p. 82].

where the meaning and typical values of the symbols can be read in Table II.

In a model of hammer–string interaction, the connection between the hammer model (exciter) and the string model (resonator) has to be specified [4]. The hammer, in its mathematical description of (33) and (34), is a lumped element. The string is usually modeled as a distributed structure such as a digital waveguide, where discrete-time velocity waves are propagated. In the case of an ideal string, having wave impedance Z_0 , the relation between transverse force and velocity is governed by [16]

$$\dot{y} = \frac{1}{2Z_0} f + v_i \quad (35)$$

where v_i is the sum of the velocity waves coming from the two halves of the string.

For convenience, let us write the resulting equations of the hammer–string system during contact in terms of the felt compression d

$$\begin{cases} \dot{d} = v_h - \frac{1}{2Z_0} f - v_i \\ \dot{v}_h = \ddot{y}_h = -\frac{1}{m_h} f \\ f = kd^\gamma. \end{cases} \quad (36)$$

Whichever method we decide to use in order to translate these equations into discrete-time form, we obtain an implicit system relating the n th sample of the force $f(n)$ and the n th sample of the felt compression $d(n)$.

This implicit relationship can be made explicit by assuming that $f(n) \approx f(n-1)$, thus inserting a fictitious delay element in a delay-free path. Although this trick has been extensively used in the literature, Fig. 5 shows that the insertion of a fictitious delay element has severe consequences on the simulation of high-pitched notes at normal audio sampling rates. On the other hand, the K method rearranges the equations in such a way that instantaneous dependencies across the nonlinearity are dropped. In this way, artificial instabilities are avoided and a reliable force signal is reproduced (see dashed line in Fig. 5), resulting in a much more natural sound. In fact, while the small bumps in the dashed line of Fig. 5 correspond to reflections from the string ends, the force spikes in solid line have no correspondence in physical measurements [17] and they are responsible for a buzzy character of sound.

Starting from (36), the K method gives the following matrices:

$$\mathbf{w} = \begin{bmatrix} d \\ v_h \end{bmatrix}; \quad \mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}; \quad (37)$$

$$\mathbf{C} = \begin{bmatrix} -\frac{1}{2Z_0} \\ -\frac{1}{m_h} \end{bmatrix}; \quad \mathbf{u} = v_i; \quad \mathbf{y} = f; \quad (38)$$

$$\mathbf{D} = [1 \ 0]; \quad \mathbf{E} = 0; \quad \mathbf{F} = 0. \quad (39)$$

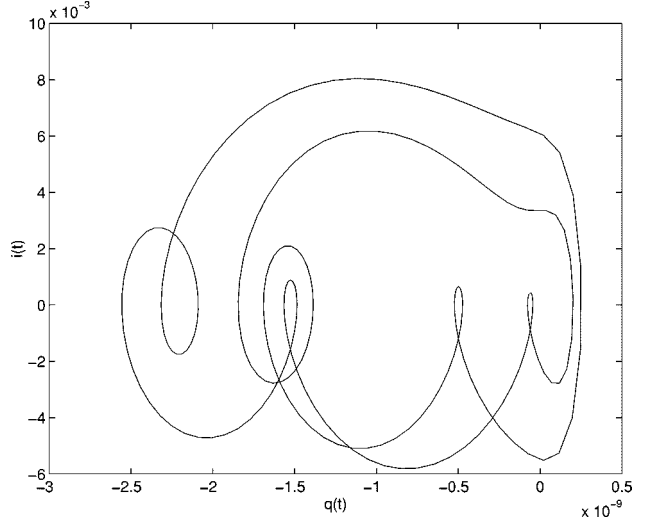


Fig. 4. Steady-state phase portrait of the Chua–Felderhoff circuit.

TABLE II
SYMBOLS USED IN THE HAMMER–STRING MODEL

Quantity	Symbol	Value
Hammer Mass	m_h	0.0066 kg
Felt Nonlinear Exponent	γ	3
Coefficient of Felt Stiffness	k	$2 \cdot 10^{11}$ (mks)
Hammer Velocity	v_h	m/s
String Wave Impedance	Z_0	2.88 kg/s
Interaction Force	f	N
Transverse String Velocity	v	m/s
Felt Compression	d	m
Sampling Rate	F_s	Hz

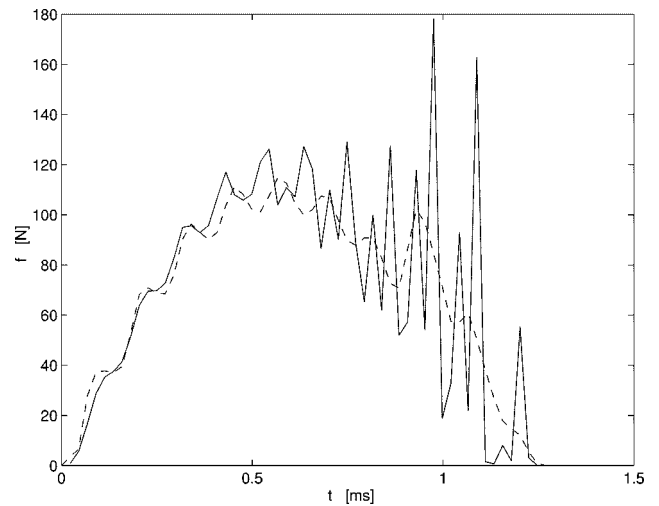


Fig. 5. Time evolution of the hammer force for a C6 (fundamental at 1046.5 Hz) with $F_s = 44\,100$ Hz, $v_h = 6.8$ m/s (*fortissimo*). Dashed line: delay-free loop resolution by the K method. Solid line: delay-free loop resolution by insertion of fictitious delay element. The bilinear transformation is used in both cases.

In this case, the matrices \mathbf{K} and \mathbf{p} of (10) and (8) are, indeed, the scalars

$$\mathbf{K} = -\frac{1}{\alpha} \left(\frac{1}{2Z_0} + \frac{1}{\alpha m_h} \right) \quad (40)$$

and

$$\mathbf{p}(n) = \left[1 \quad \frac{1}{\alpha} \right] \mathbf{w}(n-1) - \frac{1}{\alpha} \mathbf{u}(n). \quad (41)$$

It is also easy to verify that if the bilinear transformation is used, \mathbf{p} becomes

$$\begin{aligned} \mathbf{p}(n) = & \left[1 \quad \frac{2}{\alpha} \right] \mathbf{w}(n-1) - \frac{1}{\alpha} (\mathbf{u}(n) - \mathbf{u}(n-1)) \\ & + \frac{1}{\alpha} \left(\frac{1}{2Z_0} + \frac{1}{\alpha m_h} \right) f(n-1) \end{aligned} \quad (42)$$

while matrix \mathbf{K} is left untouched. In both cases, the condition for the existence of an explicit map is

$$-\frac{\partial f}{\partial d} \frac{1}{\alpha} \left(\frac{1}{2Z_0} + \frac{1}{\alpha m_h} \right) \neq 1. \quad (43)$$

Since (43) is valid for all values of d , $f(\cdot)$ is again globally explicitable, and our explicit function $f = \mathbf{h}(\mathbf{p})$ can be tabulated by linear transformation of the points (d, f) . In some simple cases, namely for $\gamma = 2$ and $\gamma = 3$ it is possible to solve $f(\cdot)$ in order to get the closed form expression for $\mathbf{h}(\cdot)$.

V. REED-BORE INTERACTION

In this section we give another example of application of the K method in musical acoustics, for an instrument characterized by a sustained interaction between exciter and resonator. Namely, we derive the matrices for the dynamic reed coupled to a bore. The model here presented differs from the over-simplified static reed model of Fig. 1 for the presence of reed dynamics which is tightly interwoven with the nonlinear characteristics.

According to Schumacher [18] and Keefe [19], [6], a sufficiently accurate model of the dynamics of a reed is a second order damped oscillator

$$\Delta p \triangleq p - P_0 = \mu_r (\ddot{x} + g_r \dot{x} + \omega_r (x - H)) \quad (44)$$

where the meaning and typical values of the symbols can be read in Table III. In (44), the effect of the Bernoulli force on the reed is neglected.

The pressure drop at the reed opening is nonlinearly related to the volume flow u [19, eq. (73)] via

$$\Delta p = -A^{-1.5} \text{sign}(u_f) |u_f|^{1.5} / x^2 \quad (45)$$

where the flow through the slit is given by

$$u_f = u + S_r \dot{x}. \quad (46)$$

Equation (45) can be summarized by expressing the mouthpiece acoustic pressure as a nonlinear function of u_f and x

$$p = \mathbf{f}(u_f, x) + P_0 = \mathbf{f} \left(\begin{bmatrix} u_f \\ x \end{bmatrix} \right) + P_0. \quad (47)$$

TABLE III
SYMBOLS USED IN THE REED-BORE MODEL

Quantity	Symbol	Value
Reed mass/area	μ_r	0.0231 kg/m ²
Effective Reed Area	S_r	1.46 · 10 ⁻⁴ m ²
Reed Resonance Freq.	ω_0	23250 rad/s
Squared Reed Res. Freq.	ω_r	23250 ² s ⁻²
Reed Damping	g_r	3000 s ⁻¹
Tip Opening	H	0.4 · 10 ⁻³ m
Amplitude Parameter	A	0.0797 (mks)
Mouth Pressure	P_0	N/m ²
Mouthpiece Pressure	p	N/m ²
Mouthpiece Volume Flow	u	m ³ /s
Flow through the Slit	u_f	m ³ /s
Reed Displacement	x	m
Bore Impedance	Z_0	2290133.5 kg/m ⁴ s
Sampling Rate	F_s	Hz

There are two fundamental simplifications behind (45) and (46): 1) the air inductance is neglected and 2) the effective reed area is assumed to be constant.

There is a third fundamental component in any model of the reed. Namely, the connection between the reed (exciter) and the bore (resonator) has to be specified [4]. We notice that the reed model, in its mathematical description of (44) and (47), is a lumped element which is coupled with a distributed structure such as a digital waveguide model of the bore, where discrete-time pressure waves are propagated. In the case of a lossless cylindrical tube having wave impedance Z_0 , the resulting relation between volume flow and pressure is

$$u = \frac{1}{Z_0} (p - 2p^-) \quad (48)$$

where p^- is the pressure wave coming from the waveguide resonator.

Equations (44), (47), and (48) can be translated into the symbols of Section II. The vector quantities of (2) are set to

$$\mathbf{w} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}; \quad \mathbf{y} = \Delta p = p - P_0; \quad \mathbf{u} = \begin{bmatrix} H \\ P_0 \\ p^- \end{bmatrix}. \quad (49)$$

The vector \mathbf{u} contains three kinds of variables: H is a parameter that is likely to remain fixed during simulation; P_0 is a slowly varying modulating signal used to control the model; p^- is the wave signal coming from the bore model.

The matrices used by the K method are easily computed from (44), (47), and (48)

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -\omega_r & -g_r \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ \omega_r & 0 & 0 \end{bmatrix}; \quad (50)$$

$$\mathbf{C} = \begin{bmatrix} 0 \\ \frac{1}{\mu_r} \end{bmatrix}; \quad \mathbf{D} = \begin{bmatrix} 0 & S_r \\ 1 & 0 \end{bmatrix}; \quad (51)$$

$$\mathbf{E} = \begin{bmatrix} 0 & \frac{1}{Z_0} & -\frac{2}{Z_0} \\ 0 & 0 & 0 \end{bmatrix}; \quad \mathbf{F} = \begin{bmatrix} \frac{1}{Z_0} \\ 0 \end{bmatrix}. \quad (52)$$

The matrix \mathbf{K} turns out to be

$$\mathbf{K} = \begin{bmatrix} \frac{\alpha S_r}{\mu_r(\omega_r + \alpha^2 + g_r \alpha)} + \frac{1}{Z_0} \\ \frac{1}{\mu_r(\omega_r + \alpha^2 + g_r \alpha)} \end{bmatrix} \quad (53)$$

and it does not contain any of the control variables of \mathbf{u} , thus indicating that the shear transformation (17) of the two-variable function $\mathbf{f}(\cdot)$ can be precomputed. Such transformation can be explicitly rewritten as

$$\begin{bmatrix} u_f \\ x \\ \Delta p \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{K} \\ \mathbf{0} & 1 \end{bmatrix} \cdot \begin{bmatrix} p_1 \\ p_2 \\ \Delta p \end{bmatrix} \quad (54)$$

where p_i is the i th component of the vector of previously-computed and known terms \mathbf{p} , which is given by (8) if backward differencing is used for discretization or by (11) if the bilinear transformation is used. Thanks to this shear transformation, we can turn (47) into the explicit

$$p = \mathbf{h}(p_1, p_2) + P_0 = \mathbf{h}\left(\begin{bmatrix} p_1 \\ p_2 \end{bmatrix}\right) + P_0. \quad (55)$$

Fig. 6 shows the nonlinear function (45), and Fig. 7 shows its sheared version computed by point-to-point transformation. The components of vector \mathbf{p} have been computed by the transformation (54). Their explicit form is

$$p_1 = u_f - \frac{\alpha S_r}{\mu_r(\omega_r + \alpha^2 + g_r \alpha)} \Delta p - \frac{1}{Z_0} \Delta p \quad (56)$$

$$p_2 = x - \frac{1}{\mu_r(\omega_r + \alpha^2 + g_r \alpha)} \Delta p. \quad (57)$$

The shear transformation is easily visualized by comparison of the contour plots on the floor of Figs. 6 and 7.

At every step of simulation, the incoming pressure wave and the mouth pressure P_0 are used to calculate the vector \mathbf{p} . Then, the sheared function $\mathbf{h}(\mathbf{p})$ can be used to give the value of the outgoing pressure wave p^+ at the same sampling instant

$$p^+ = \mathbf{h}(\mathbf{p}) + P_0 - p^-. \quad (58)$$

A. Practical Issues for High-Dimensional Systems

The reed-bore interaction model might be as well expressed by means of a three-variable function whose domain contains

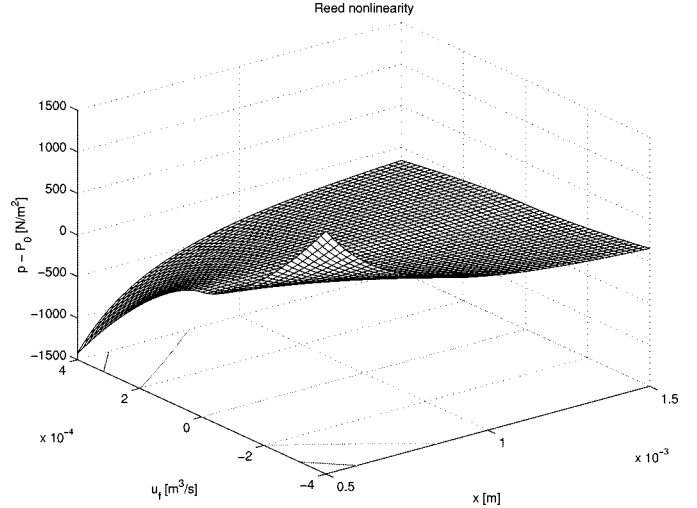


Fig. 6. Acoustic pressure as a nonlinear function of reed displacement and flow through the slit.

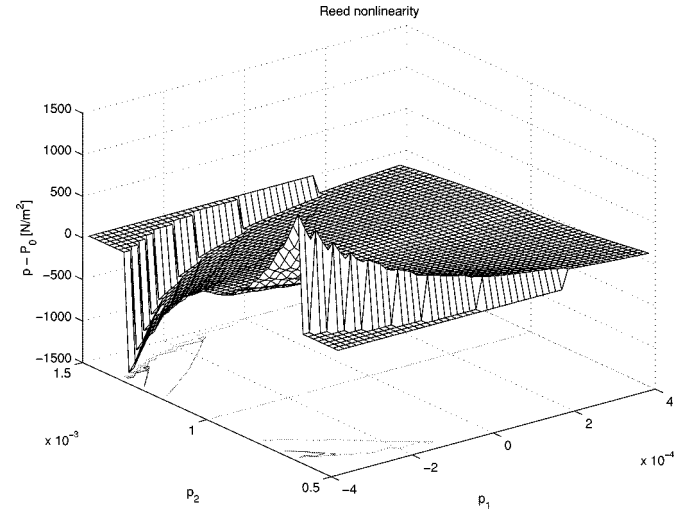


Fig. 7. Acoustic pressure as a sheared function of variables p_1 and p_2 .

the state variables x and \dot{x} , and the volume flow u [20]. The realization here presented has a reduced domain dimensionality, thus allowing an easier computation and display of the sheared function.⁵ However, for other acoustic systems it might be impossible to reduce the domain dimensionality to less than three. For such systems, it would be easier to perform a simulation where the nonlinearity is computed by means of an iterative method (e.g., Newton). As a byproduct of this simulation, a uniform sampling of the space of variables in their range of variability can be produced, and a sheared table directly computed when it is needed for hard real-time implementation. As a matter of fact, even the direct implementation of the iterative solution can run quite fast on modern superscalar architectures due to fact that the iteration is all comprised in a tight loop.

In these cases, the nonlinearity should be replaced by an implicit equation such as

$$\tilde{g}(p^+, p_1, p_2, \dots, p_m) = 0 \quad (59)$$

⁵We are thankful to F. Avanzini for pointing out this clearer formulation.

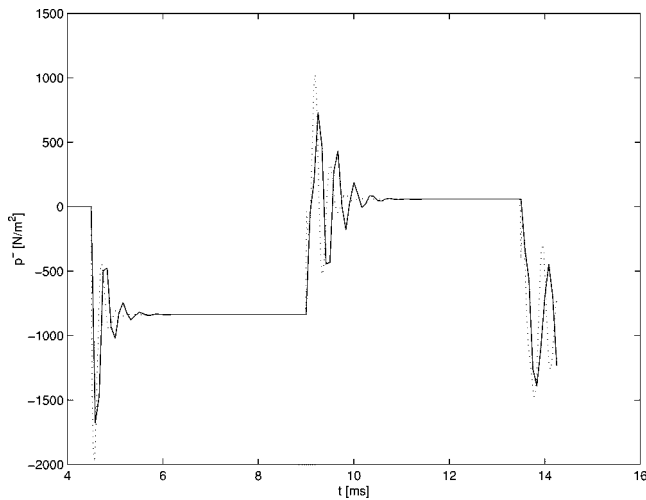


Fig. 8. Time evolution of the incoming pressure at the mouthpiece obtained with the bilinear transformation. The sampling rate is set to $F_s = 12\,000$ Hz (solid line) and $F_s = 44\,100$ Hz (dotted line). The bore is an ideal lossless cylinder and the reflection at the bell is approximated with a multiplication by -0.95 . The mouth pressure signal is a step function ($P_0 = 2100$ N/m²).

and the Newton iterative method can be summarized by the formula

$$p_{k+1}^+ = p_k^+ - \frac{\tilde{g}_k}{\tilde{g}'_k} \quad (60)$$

where k is the iteration counter and the derivative is computed with respect to p^+ .

As it was in the case of the hammer-string simulation, the K method allows to obtain a qualitatively correct behavior of the signals even when a very low sampling rate is used in the model, and regardless of the discretization technique that is chosen. It is clear that by increasing the sampling rate, and using the bilinear transformation (possibly with some control on the degree of frequency warping) instead of backward differencing, it is possible to achieve more accurate results. However, the important point here is that, as long as the frequency components generated by the nonlinearities do not alias, the simulation degrades gracefully and even a very economical implementation can be used reliably without severe impairment in timbre quality. For instance, Fig. 8 shows the incoming pressure waveform at the mouthpiece obtained when using the K method with the bilinear transformation at two different sampling rates. The qualitative behavior is similar in the two waveforms and differences in the ripples are due to the frequency warping introduced by the bilinear transformation.

VI. CONCLUSION

We have proposed a general method for solving computability problems in systems containing linear and nonlinear components. The method has been validated on the Chua-Felderhoff circuit whose nonlinear dynamic behavior is sensitive to the discretization technique. Two important examples in musical acoustics have been presented: the hammer-string interaction shows how the K method avoids the

artifacts introduced by other nonexact solutions; the bore-reed interaction is an example where all the matrices introduced in the general formulation are nontrivial.

REFERENCES

- [1] J. O. Smith, "Efficient simulation of the reed-bore and bow-string mechanisms," in *Proc. Int. Computer Music Conf.*, The Hague, The Netherlands, 1986. Also available in [3], pp. 275–280.
- [2] —, *Principles of Digital Waveguide Models of Musical Instruments*, M. Kahrs and K. Brandenburg, Eds. Norwell, MA: Kluwer, 1998, pp. 417–466.
- [3] —, "Music applications of digital waveguides," CCRMA, Stanford University, Stanford, CA, Rep. STAN-M-39, 1987.
- [4] G. Borin, G. De Poli, and A. Sarti, *Sound Synthesis by Dynamic Systems Interaction*, D. Baggi, Ed. Los Alamitos, CA: IEEE Comput. Soc. Press, 1992, pp. 139–160.
- [5] D. E. Hall, "Piano string excitation VI: Nonlinear modeling," *J. Acoust. Soc. Amer.*, vol. 92, pp. 95–105, July 1992.
- [6] D. H. Keefe, "Physical modeling of wind instruments," *Comput. Music J.*, vol. 16, pp. 57–73, Winter 1992.
- [7] K. Meerkötter and R. Scholtz, "Digital simulation of nonlinear circuits by wave digital filter principles," in *Proc. IEEE Int. Symp. Circuits Syst.*, Portland, OR, May 1989, pp. 720–723.
- [8] T. Felderhoff, "Simulation of nonlinear circuits with period doubling and chaotic behavior by wave digital filter principles," *IEEE Trans. Circuits Syst.*, vol. 41, pp. 485–491, July 1994.
- [9] A. Sarti and G. De Poli, "Generalized adaptors with memory for nonlinear wave digital structures," in *Proc. Eur. Signal Processing Conf.*, Trieste, Italy, Sept. 1996, pp. 1941–1944.
- [10] —, "Toward nonlinear wave digital filters," *IEEE Trans. Signal Processing*, vol. 47, pp. 1654–1668, June 1999.
- [11] A. Härmä, "Implementation of recursive filters having delay free loops," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Seattle, WA, 1998, pp. 1261–1264.
- [12] J. Szczupak and S. K. Mitra, "Detection, location, and removal of delay-free loops in digital filter configurations," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 23, pp. 558–562, Dec. 1975.
- [13] S. Lang, *Differential Manifolds*. New York: Springer-Verlag, 1985.
- [14] T. Kailath, *Linear Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1980.
- [15] L. O. Chua, "Nonlinear circuits," *IEEE Trans. Circuits Syst.*, vol. CAS-31, pp. 69–97, Jan. 1984.
- [16] G. Borin and G. De Poli, "A hysteretic hammer-string interaction model for physical model synthesis," in *Proc. Nordic Acoustical Meeting*, Helsinki, Finland, June 1996, pp. 399–406.
- [17] D. E. Hall, *The Hammer and the String*, A. Askenfelt, Ed. Stockholm, Sweden: Royal Swedish Academy of Music, 1990, pp. 59–72.
- [18] R. T. Schumacher, "Ab Initio calculations of the oscillations of a clarinet," *Acustica*, vol. 48, no. 2, pp. 71–85, 1981.
- [19] D. H. Keefe, "On sound production in reed-driven wind instruments," *System. Musicol. Progr. School Music*, Univ. Washington, Seattle, Tech. Rep. 9003, 1990.
- [20] G. Borin, G. De Poli, and D. Rocchesso, "Elimination of delay-free loops in discrete-time models of nonlinear acoustic systems," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, Mahonk, NY, 1997, pp. 12.1.1–4.



Gianpaolo Borin received the Laurea degree in electronic engineering from the University of Padova, Italy, in 1990 with a thesis on sound synthesis by physical modeling.

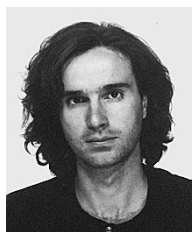
Since then, he has been doing research at the Centro di Sonologia Computazionale, University of Padova. He has also been working both as a Unix Professional Developer and as a Consultant Researcher for Generalmusic, S.p.A. He is coauthor of a Generalmusic's U.S. patent for digital piano postprocessing methods. His current research interests include algorithms and methods for the efficient implementation of physical models of musical instruments and tools for real-time sound synthesis.



Giovanni De Poli (S'71–M'87) received the Laurea degree in electronic engineering from the University of Padova, Italy.

He is currently an Associate Professor of computer science with the Department of Electronics and Informatics, University of Padova. His research interests are in algorithms for sound synthesis, representation of musical information and knowledge, and human–machine interaction. He is co-editor of the books *Representations of Music Signals* (Cambridge, MA: MIT Press, 1991) and *Musical*

Signal Processing (Lisse, The Netherlands: Swets & Zeitlinger, 1996).



Davide Rocchesso (S'93–A'96) received the Laurea degree in electronic engineering from the University of Padova, Italy, in 1992, and the Ph.D. degree from the same university in 1996. His Ph.D. research involved the design of structures and algorithms based on feedback delay networks for sound processing applications.

In 1994–1995, he was a Visiting Scholar at the Center for Computer Research in Music and Acoustics, Stanford University, Stanford, CA.

Since 1991, he has been collaborating with the Centro di Sonologia Computazionale, University of Padova, as a Researcher and a Live-Electronic Designer. Since March 1998, he has been with the Dipartimento Scientifico e Tecnologico, University of Verona, Verona, Italy, as an Assistant Professor. His main interests are in sound processing, physical modeling, sound reverberation and spatialization, multimedia systems.